
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

**UNIVERSITY OF MACEDONIA**
**DEPARTMENT OF ECONOMICS**

**A Thesis presented for the degree of**
**Economics**

# Introduction to Structural Equation Modeling and Bayesian Networks in Statistics

**Vasileios I. Neokosmidis**
**Registration Number:** eco18068

**Supervisor:**

**Dimitrios Ioannidis**

**Co-Supervisor:**

**Theodoros Panagiotidis**

**THESSALONIKI, GREECE**
**AUGUST 2021**

# Acknowledgements

Special thanks to my family and loved ones for helping me throughout this thesis. Additionally, I wish to thank professor Ioannidis Dimitrios for accepting me to implement this bachelor thesis with him and the co-supervisor professor Panagiotidis Theodoros, who with his lessons gave me a supplementary material for this thesis. Finally, I am grateful to Mr. Athanasiadis Giannis for all his useful support in conducting this thesis.

# Contents

# Acronyms

| | |
|---|---|
| **AGFI** | Adjusted Goodness of Fit Index |
| **AIC** | Akaike Information Criterion |
| **AMOS** | Analysis of Moment Structures |
| **AVE** | Average Variance Extracted |
| **BDeu** | Bayesian Dirichlet equivalent uniform |
| **BIC** | Bayesian Information Criterion |
| **CFI** | Comparative Fit Index |
| **CPQ** | Conditional Probability Queries |
| **CR** | Composite Reliability |
| **DIC** | Deviance Information Criterion |
| **EQS** | Equations |
| **FIML** | Full Information Maximum Likelihood |
| **GFI** | Goodness of Fit Index |
| **HPD** | Highest Probability Density |
| **JASP** | Jeffrey's Amazing Statistics Program |
| **LISREL** | Linear Structural Relations |
| **MAE** | Mean Absolute Error |
| **MAP** | Maximum A Posteriori |
| **MAPE** | Mean Absolute Percentage Error |
| **MI** | Modification Indices |
| **ML** | Maximum Likelihood |
| **MPE** | Most Probable Explanation |
| **MSE** | Mean Squared Error |
| **NFI** | Normed Fit Index |
| **OLS** | Ordinary Least Squares |
| **PC** | Principal Component |
| **PCA** | Principal Component Analysis |
| **PGFI** | Parsimonious Goodness of Fit Index |
| **PNFI** | Parsimonious Normed Fit Index |
| **RAM** | Reticular Action Model |
| **RMR** | Root Mean Square Residual |
| **RMSEA** | Root Mean Square Error of Approximation |
| **RSME** | Root Square Mean Error |
| **SEM** | Structural Equation Modeling |
| **SVD** | Singular Value Decomposition |
| **TLI** | Tucker Lewis Index |
| **VIF** | Variance Inflation Factor |

# Abstract

In this thesis, theoretical background and applications of several statistical analyses of unsupervised learning are presented to conduct SEM. After SEM, the Bayesian SEM is introduced to approach the matter in a more probabilistic way. In the end, of every critical statistical concept, the corresponding R lab is conducted to demonstrate the usage and implement the theory. This paper includes 3 chapters: 1) General knowledge about data and data types, 2) Basic Concepts to conduct SEM, 3) Introduction to Bayesian Networks in Statistics.

The first chapter is dedicated in explaining the basic knowledge of the field of statistics. In fact, at the start of this chapter, the concept of variability is explained so that readers who are beginner at statistics can get familiar with data science. Consequently, one of the most important data categorization is mentioned. This is quantitative and qualitative data. On top of that, the theory and formulas of basic statistical measures such as 1) mode, 2) median, 3) mean, 4) variance, 5) standard deviation and 6) z-score are presented. After the completion of this chapter, the reader should be able to understand the basic concepts of statistics.

The second chapter is dedicated in the explicit analysis of unsupervised statistical methods which are essential to conduct SEM. At the start of this chapter, 1) a definition of SEM, 2) the history of SEM, and 3) R, the statistical software used during this thesis to conduct SEM, are briefly discussed. Two of the most important concepts in unsupervised learning are covariance and correlation. Thus, the rationale, formulas and applications of these two concepts are explicitly analyzed to prepare the reader for the upcoming statistical analyses. At this point, two of the most essential statistical techniques of SEM will be introduced to the reader. Namely, PCA and Cluster Analysis. At the start of the PCA section, basic concepts such as 1) eigenvectors and eigenvalues, 2) PCA loadings and 3) positive semi-definite matrices are theoretically explained. In the PCA lab, Principal Component Analysis is conducted with both correlation and covariance matrices as input with matrix and R approach to obtain the first (PC1) and second (PC2) principal component. Subsequently, PC1 and PC2 are used as input for the cluster analysis lab. Basic components of cluster analysis such as 1) silhouette method and 2) k-means clustering algorithm are explicitly analyzed. In the end of the second chapter, basic concepts in the SEM framework such as 1) endogeneity and exogeneity, 2) path analysis, 3) observed and latent variables and 4) graph theory and notation are analytically discussed. In the SEM lab, information obtained from the previous analyses of the second chapter regarding the variables are used to specify, identify, estimate and evaluate a SEM model. The final SEM model shows the interactions between the a) profile of an employee in a company, b) monthly income, c) quality of work, d) overall working years, e) number of companies worked, f) age, g) percentage of salary increase and h) rating of work performance. After the completion of this chapter, the reader should familiar with unsupervised analyses such as 1) Principal Component Analysis, 2) Cluster Analysis and ultimately 3) Structural Equation Modeling.

The third and final chapter of this thesis is dedicated in Bayesian statistics and more specifically the Bayesian approach in SEM modeling. At the beginning of the chapter, fundamental ideas of Bayesian statistics such as 1) non-informative and informative priors and

2) graph theory and notation are analyzed. Moving on to some more advanced topics in Bayesian statistics, the reader is introduced to concepts such as 1) maps, 2) d-separation, 3) essential graphical connections and 4) Markov blankets. In the last section of the theory of Bayesian Networks theory, the art of 1) exact inference and 2) approximate inference through Bayesian Networks is introduced to the reader. In the lab section, an BN example with R is conducted. In the BN lab, the probabilistic relationships between the a) age, b) quality of work, c) monthly income, d) number of companies worked, e) overall working years, f) years at company g) years in current role h) years since last promotion and i) years with current manager, of an employee are examined. First, the dataset is manipulated so that discrete BN analysis can take place. Then, the Directed Acyclic Graph (DAG) of the BN is specified and the parameters of the BN network are estimated with maximum likelihood or bayes estimation. After that, the fit of the DAG is evaluated through 1) Pearson's conditional independence tests 2) Bayesian Information Criterion (BIC) and 3) Bayesian Dirichlet equivalent uniform (BDeu) network scores. The last part of the analysis involves querying the BN to obtain the probability of combinations of events and evidences. This is done through either exact or approximate inference. Simple queries can be answered accurately through exact inference while more complex queries are approached through approximate inference. Additionally, R code for bar charts and dot plots is presented in the context of DAG and conditional probability distributions. Some of the most interesting queries which are given to the BN to answer are: 1) What is the probability of an old and experienced individual to perform low at his job and have a low monthly income compared to the probability of the same individual to have very high income and perform excellent at his job? and 2) What is the probability of an old individual who has worked in many different companies to be working 0 to 10 years at the same company compared to the same individual to be working 10 to 20 years?

More datasets and R source codes of this thesis can be found at: https://github.com/BillNeokosmidis/Intro-to-SEM-and-BN.git.

# Chapter 1

# General knowledge about data and data types

## 1.1 Statistics and Data

The science of statistics, accompanied by its methods of data analysis, is a key tool behind every science. In business world, statistics is taken heavily into account to study new products, predict sales, or to measure employee performance and eventually maximize the profit. In finance, statistics is used to study stock returns and investment opportunities. In medical science, statistics help to evaluate vaccines and new therapies by comparing them with older ones, and keeping track of patient history. Reasonably, most of the sciences that prosper society owe their growth to statistics. Statistics is able to answer any question that arises regardless of the scientific field. Learning the fundamental statistical concepts makes every scientist skeptical about his research findings and helps him filter the extensive volume of data information which surround him.

Subsequently, a definition of the term "statistics" is imperative. In the singular, the word statistic is simply a number calculated from data, but in the plural the term statistics is related to the way of thinking about data and a quantifying uncertainty. Statistics is the art and science of designing studies and analyzing the data from the product of those studies. Its ultimate goal is to translate ideas into data and data into knowledge improving the perception of the world around us (Agresti and Franklin (2018)).

Statistical problem solving is a process which involves four components.

1. Formulate a statistical question. Is the question correctly rendered? What does this research want to achieve? What question does the researcher want to answer? How many and which attributes is the researcher going to include in the study? These are some of the questions that all researchers should ask themselves. Generally in this step, it is described what is intended to be measured and if is accurately reflecting the purpose of the research.

2. Collect data. In gathering data, researchers deal with practical matters. The selection of the subjects that are suitable for the particular research, and from which the data will be collected. The number of subjects, which must be representative of the total population, known as sample. The method that is going to be used in order to collect the data from individuals and form the dataset. There are two methods that prevail, surveys or questionnaires and experiments. Additionally, researchers must make sure that subjects will give the same response when they are called to answer for the same study again. Invalid or unreliable data gathering render statistical analyses of the data meaningless and even possibly misleading.

3. Analyze data. This step includes the selection of the statistical analyses which depends on the nature of the problem. It requires proper fundamental knowledge of these statistical procedures, otherwise execution, interpretation, conclusions and reporting will be invalid.

4. Interpret results. This step includes interpretation of the results based on the outcome of the corresponding statistical analysis. This is another step in which good sense of statistics is crucial.

The information, which are gathered with surveys, questionnaires or experiments aiming to examine certain hypotheses are collectively called data. Data are obtained by multiple subjects. Each subject has a set of values on a specific number of items. Each item represents a unique characteristic or attribute (see **Variables** section). These items could be anything that has to do with individuals. For example, if the subject of interest are people, representative items could be their height, weight, marriage status etc. Data are collected and tested by the researcher via statistical procedures to.
Below, an example of complete research is demonstrated (**Example 1.1**).

**Example 1.1.** A political intervention could become motivation for examining the results of this decision. Assuming that a researcher decides to study the influence of an intervention such as a law. The data items of interest are the country's economic magnitudes. Thus, this data collection might consist of the following economic items (measures) for the individuals participating in the research: wage, debts, employment status, savings etc (Step 1). The researcher needs to collect data from an appropriate number of citizens forming a sample representative of the total population. The sample consists of people which is the type of subjects under study, in two time points: before and after the application of the law (Step 2). The researcher acquired with the proper statistical knowledge should run statistical tests comparing the results of the two investigations and calculate if there is a significant difference between them (Step 3). If the outcome of the test indicates the presence of significant difference between the two scenarios (before and after the application of the law) then there is indisputable evidence that the law influences the citizens' life financially. To complete the study, the researcher must interpret the results, derive into conclusions and report them to the state (Step 4).

## 1.2 Variables

There is an amount of variability in every single thing in life. Statistical methods provide ways to measure and better understand it. In a research environment, variation is a common phenomenon among the characteristics of the different and unique subjects. For instance, variability can be observed between different cars in color, brand, horsepower, gas consumption etc. Apart from subject variation, characteristics may differ by time as well. Applied in the previous example, the gas consumption can differ by both car and time since in most cases the older the car the more gas it consumes.

This variety of countless characteristics expressed through variables triggers peoples' interest to analyze and study them even further. The term "variable" comes from the word "vary" which predisposes about its interpretation. The essence of a variable is that it contains data values that constantly change. Agresti and Franklin (2018) gives a definition of a variable as the characteristic that can vary in value among subjects observed in a study. In the previous example, horsepower, gas consumption, and color form three variables because their values are differentiated from car to car. In real world experiments, variables are typically listed in the columns of a dataset, with the rows referring to the unique subjects

(Table 1.1). The preferable unit of measure of each variable is ordinarily based on the nature of the variables being studied. In this case for example, the variable gas consumption is measured in litres.

Table 1.1: An example of a dataset with 20 cars containing three variables.

| Observation | Horsepower | Gas consumption | Color |
|---|---|---|---|
| 1 | 405 | 40 | Red |
| 2 | 250 | 20 | Brown |
| 3 | 200 | 15 | Black |
| 4 | 300 | 22 | Yellow |
| 5 | 225 | 16 | Black |
| 6 | 400 | 38 | White |
| 7 | 150 | 13 | Red |
| 8 | 234 | 17 | Orange |
| 9 | 282 | 19 | Red |
| 10 | 291 | 21 | Black |
| 11 | 140 | 11 | Red |
| 12 | 198 | 14 | White |
| 13 | 211 | 18 | Yellow |
| 14 | 260 | 16 | Black |
| 15 | 170 | 20 | Brown |
| 16 | 205 | 16 | Red |
| 17 | 255 | 17 | Black |
| 18 | 420 | 39 | White |
| 19 | 100 | 8 | Black |
| 20 | 392 | 37 | Black |

## 1.3 Theoretical Definitions of Basic Statistical Measures

### 1.3.1 Mode

The mode is the observed value which has the highest frequency of appearance in a component. Mode is one of the measures of central tendency (Figure 1.3). Central tendency is a family of measures that aim to explain a component through a single central value. Statistical components such as datasets and variables can have no mode at all, one, two or multiple mode values. A component with one mode is called unimodal while a variable with two modes is called bimodal. Less popular but not non-existent are trimodal variables with three modes and multimodal with more than three. Mode is one of the most important statistical measures when examining qualitative data, especially in the nominal level of measurement (subsection 1.4.2).

For example, assuming a variable with the following values: 18, 19, 40, 40, 12, 1. Here the value of 40 occurs twice in the variable and therefore is the mode value of the whole variable as shown in Figure 1.1 at the left side and in the corresponding table at the right side. This variable is characterized as unimodal.

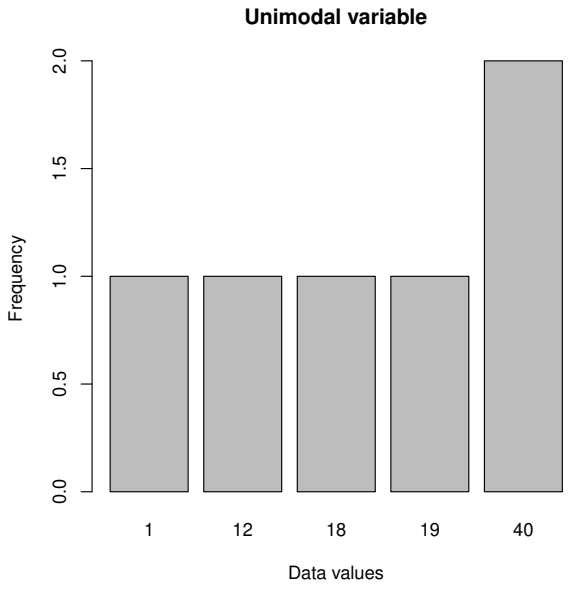



Table 1.2: Table of a unimodal variable.

| Data values |
|---|
| 1 |
| 12 |
| 18 |
| 19 |
| 40 |
| 40 |

Figure 1.1: Barplot of a unimodal variable.

Now, assuming another variable with the following values: 40, 40, 60, 60, 32, 14, 15 in Figure 1.2 at the right side and the corresponding table in the left side. In this example, both observed values 40 and 60 are both modes since they appear equal amount of times in the variable (two times). This variable is characterized as bimodal.

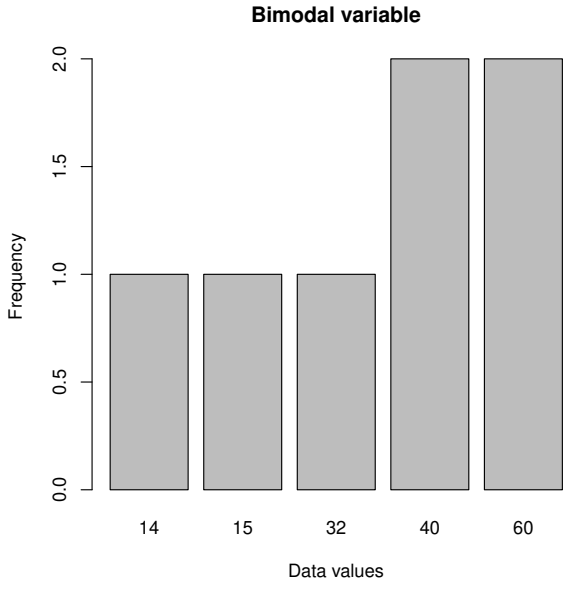



Table 1.3: Table of a bimodal variable.

| Data values |
|---|
| 14 |
| 15 |
| 32 |
| 40 |
| 40 |
| 60 |
| 60 |

Figure 1.2: Barplot of a bimodal variable.

### 1.3.2 Median

The median value is the middle value out of all the values of a component and its another measure of central tendency (Figure 1.3). For the median to be computed numbers must be sorted first, in ascending or descending order. The formula which calculates the median is

different each time depending in whether the component has odd or even number of values. In Equation 1.1 the formula is demonstrated, where N is the number of values.

$$Median = \begin{cases} (N+1)^{th}\,\text{data point}, & \text{When N is even.} \\ \dfrac{\frac{N^{th}}{2}\,\text{data point} + (\frac{N}{2}+1)\,\text{data point}}{2}, & \text{When N is odd.} \end{cases} \tag{1.1}$$

If the median is calculated from a component with odd number of values then it must be separated into two subsets with equal number of values, the lower half and the higher half. By using the Equation 1.1, both cases of datasets, with odd and even numbers, will be analyzed below. For example, assuming there is a dataset with odd number of values as following: 21, 23, 45, 32, 53, 64, 30. The ascending ordered dataset has the following form: 21, 23, 30, 32, 45, 53, 64. In this case the median value is the $4^{th}$ data point, the number 32, because it divines the dataset equally into two subsets with three values each. When the median is calculated from a component with even number of values the middle pair of values must be identified and divided by two to estimate the median value. For example, assuming there is a dataset with even number of values: 43, 56, 79, 33, 15, 19. The ascending ordered dataset has the following form: 15, 19, 33, 43, 56, 79. In this case, the median value is between the $3^{rd}$ value, 33, and the $4^{th}$ value, 43. By summing these two numbers and then diving them by 2 we get the median value which is: $(33 + 43)/2 = 38$. Median is one of the most important statistical measures when examining qualitative data in the ordinal level of measurement (subsection 1.4.2) and quantitative data in the interval and ratio level of measurement.

### 1.3.3 Mean

Mean, or expected value is a statistical measure which measures the average or central tendency (Figure 1.3) of a component and is the most important data characteristic in statistics. This measure when drawn from the entire population is called population mean ($\bar{x}$) and when is drawn from a sample of the population is called sample mean ($\mu$). It is calculated by using the set of values of a component, usually either variable or dataset. This measure is produced by the adding all the individual values of the component of interest and then diving them by their count. Mean is one of the most important statistical measures when examining quantitative data in the interval and ratio level of measurement.

In Equation 1.2, the formula of the population mean ($\mu$) is demonstrated, where $n$ is the number of values in a component of a population, and $x_n$ is the last value of the component with N number of values.

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n-1} = \frac{x_1 + x_2 + x_3 ... + x_N}{n-1} \tag{1.2}$$

In Equation 1.3, the formula of the sample mean ($\bar{x}$) is demonstrated, where $n$ is the number of values in a component of a sample, and $x_n$ is the last value of the component with n number of values.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n-1} = \frac{x_1 + x_2 + x_3 ... + x_n}{n-1} \tag{1.3}$$

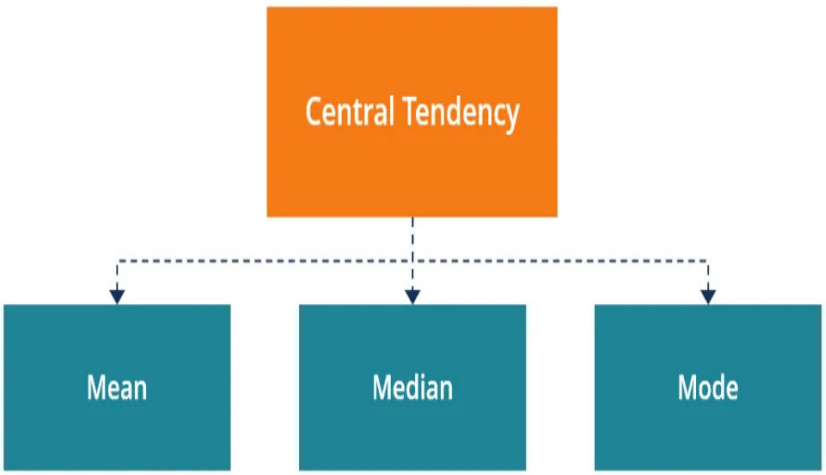



Figure 1.3: Statistical measures of central tendency.
(http://bit.ly/2ZjHibH)

### 1.3.4 Variance

The variance is a statistical measure of the spread or distance in average between values in a variable or a dataset and their mean. In the field of statistics, as the term might imply, this unique value measures variability from the mean. When the variance is drawn from the entire population is called population variance ($\sigma^2$) and when is drawn from a sample of the population is called sample variance ($s^2$). Essentially, the measurement of variance represents the mean distance of the data points of a component from the mean value. The first step to calculate this statistical measure is done by taking the differences between each value from the mean in the variable or dataset of interest. After that the differences are squared up so that they turn into positive numbers because they measure distance (negative number wouldn't make sense). The last step is to add up all the squared up differences and divide them by the count of the values.

In Equation 1.4, the formula of the population mean ($\sigma^2$) is demonstrated, where $n$ is the number of values in a component of a population, $x_n$ is the last value of the component with N number of values and $\mu$ is the mean of the population.

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1} = \frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + (x_3 - \mu)^2 ... + (x_N - \mu)^2}{n-1} \tag{1.4}$$

In Equation 1.5, the formula of the sample variance ($s^2$) is demonstrated, where $n$ is the number of values in a component of a sample, $x_n$ is the last value of the component with n number of values and $\bar{x}$ is the mean of the sample.

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 ... + (x_n - \bar{x})^2}{n-1} \tag{1.5}$$

### 1.3.5 Standard Deviation

Each unique subject along with the corresponding set of values for each variable is also called observation. Each observation deviates from the mean. The formula to calculate this deviation is $x - \bar{x}$ for sample observations and $x - \mu$ for population observations. If this

value is positive then the observation under study is above the mean, while a negative value indicates an observation below the mean. Therefore, from the definition of the mean as a concept, the sum of all the deviations of a component, variable or dataset, is always gonna be equal to zero (Agresti and Franklin (2018)). To overcome the issue of the zero value existence either absolute values or squared deviations are used to calculate both variance and standard deviation. However as it was demonstrated earlier, the variance formula uses squared values which can become very complex. For that very reason, the root of this formula is used and results in the standard deviation. The standard deviation is a statistical measure of the amount of variation or dispersion of a variable or dataset from the mean. As the name might imply, the measurement of standard deviation determines on average how far away are data points from the mean of the dataset or variable. If the standard deviation is drawn from a population is called population standard deviation ($\sigma$) and if it is drawn from a sample it is called sample standard deviation ($s$). The higher the standard deviation of a variable or a dataset the more distant are data values from the mean, and the opposite. It is calculated as the square root of the variance ($\sqrt{\sigma^2}$=$\sigma$). An empirical rule says that in most cases of variables or datasets, all the observations fall within three standard deviations from the mean (Agresti and Franklin (2018)). As mentioned earlier, the two statistical concepts are very closely related. Therefore, the calculation process is exactly the same, except here the variance formula is square rooted.

In Equation 1.6 and Equation 1.7, the formula of both population and sample standard deviation is demonstrated, respectively.

$$\boxed{\sigma = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n-1}} = \sqrt{\frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + (x_3 - \mu)^2 ... + (x_N - \mu)^2}{n-1}}} \tag{1.6}$$

$$\boxed{s = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 ... + (x_n - \bar{x})^2}{n-1}}} \tag{1.7}$$

### 1.3.6 Z-score

The Z-score for a value is a statistical measure that displays distance in terms of number of standard deviations that this value falls from the mean. The unit of measure of Z-scores is standard deviation. An empirical rule is that the z-score rarely falls off more than three deviations away from the mean (Agresti (2017)). For example, if $Z-score = 0$ the candidate value is equal to the mean, if its 1 then is 1 standard deviation above the mean, and if its -1 its 1 standard deviation below the mean. The Z-score is calculated by subtracting the mean score of the component of interest from the candidate value and diving the outcome with the standard deviation. In Equation 1.8, the Z-score formula is displayed where x is the value of interest, $\mu$ is the population mean and $\sigma$ the standard deviation of the population.

$$\boxed{Z = \frac{x - \mu}{\sigma}} \tag{1.8}$$

## 1.4 Data Types

As mentioned in previous section, each observation has its own value for each variable of the dataset. In the field of statistics the researcher is continually measuring or counting data values during a study or experiment. Even though data values vary between different observations, they all share the same measurement scale on each variable. Each variable takes on a set of values that form its measurement scale. Measurement scale is a tool that helps the researcher to measure the data values of a variable. Subsequently, it allows him to identify variation between data points of a variable. Measurement scales are also involved in understanding the deeper meaning of the variables and choosing the appropriate methods of statistical analysis. The preferable scale is ordinarily based on the purpose of the research. Some of the criteria include the nature of the variables being studied and the desired statistical procedures (Stevens et al. (1946)). The classification of variables in relation to their measurement scale is decided by the type of attributes or characteristics they represent. Variables, and by extension, data are classified into one of two measurement scales, non-metric (qualitative) or metric (quantitative). A variable can take on numbers as values such as the distance of a marathon runner in a day or be classified to categories, such as "yes" or "no", "Good"-"Intermediate"-"Bad". Practically, the researcher defines the measurement scale for each variable since for the computers data values are simply, numbers. Another distinction of data is by the the amount of their values into discrete and continuous.

### 1.4.1 Quantitative Data

Numerical variables, also referred to as quantitative, express a certain quantity, amount or range. As the names implies, the measurement scale of these variables consists of numerical values exclusively. These types of data are also called metric because they are used when differences are presented between subjects measured in amount or degree on a characteristic. Agresti and Franklin (2018) defines a variable as quantitative when its data values takes on numerical values that represent different magnitudes of the variable. He claims that numerical values must represent different magnitudes of the variable since they measure quantity. For this very reason, there are units of measure associated with this data. It would not make sense to assign IDs of people to the category of quantitative data because they do not vary in quantity. On the other hand, it would make sense to say that the average wage (variable) of 10 people (observations) is 5000$. The collection of quantitative data consists of numbers that represent real amounts that can be used in basic mathematical operations such as, addition, subtraction, division and multiplication. Quantitative variables can be either discrete or continuous. Reasonably, this type of variable is used to describe the attributes or properties, which an object or situation possesses, expressed in numbers. Consequently, numerical data is the most popular variable type participating in the majority of statistical analyses. The most critical features to describe in a numerical variable is the mean, and the spread or variance (Equation 1.9).
Below, some examples of numerical variables are demonstrated (**Example 1.2** and **Example 1.3**).

**Example 1.2.** The height of people is a numerical variable. Let's assume that a dataset contains only one variable, height with two observations and unit of measure is metres. In this example different heights variate in quantity and it would be appropriate to use numerical values to represent them. The first data value is 1.70m and the second is 1.80m. Basic mathematical operations can be applied with these numbers, for example addition, $1.70 + 1.80 = 3.5$m and multiplication, $1.70 \times 1.80 = 3.06$m. Statistical measures like mean and variance are possible to compute with the following formulas:

$$
\begin{aligned}
Mean &= \frac{(1.70 + 1.80)}{2} = 1.75 \\
Variance &= \frac{(1.70 - 1.75)^2 + (1.80 - 1.75)^2}{2} = 0.0025
\end{aligned}
\tag{1.9}
$$

**Example 1.3.** The temperature of countries is another numerical variable. Let's assume that a dataset contains only one variable, temperature with two observations and unit of measure in °C. The first data value is −5°C and the second is 15°C. As with the previous example basic mathematical operations can be applied with these numbers, for example subtraction, $-5 - 15 = -20$°C and division, $15/-5 = -3$°C. Mean and variance is possible to compute, the mean of these numbers is, 5°C and the variance is 100.

### 1.4.2 Qualitative Data

Categorical variables, also referred to as qualitative, do not involve mathematical content to their values but instead data are being categorized based on labels which describe an attribute. As the name implies, the measurement scale of these variables consists of multiple categories depending on the nature of each variable. These data are also called non-metric because they are used to render the presence or absence of an attribute or property. Agresti and Franklin (2018) defines a categorical variable when it contains data values that take on categorical values that correspond to one out of a set of distinct categories. For example, a person can be either male or female in gender, not both. He claims that the unique categories of qualitative data differ in quality not in quantity. Practically, values that represent each category are characters such as "a", "b", "c", "Male", "Female" or numbers such as "1", "2", "3" that do not have mathematical meaning, but instead they just name the categories.

Qualitative data are an important part of the majority of statistical analyses. They apply their own set of statistical methods which are different from those of quantitative data. The most crucial features for categorical variables include mode, frequencies and proportions. Mode is the category that appears most often in a dataset. Frequency for each category of a categorical variable is the corresponding number of observations that match the data value that represents the specific category. Proportions are the percentages that each category accounts for out of the whole. Researchers primarily present qualitative data with tablures and graphs. It is worth noting that, researcher can divide a numerical variable into multiple intervals creating the corresponding number of categories. Then, automatically, the numerical variable becomes categorical since those intervals can be presented as different categories. Below, some of examples of categorical variables are demonstrated (**Example 1.4** and **Example 1.5**).

**Example 1.4.** Diabetes is a categorical variable. Diabetes is divided into two types, type 1 and type 2. In order for researchers to study patients with diabetes they have to make a categorical variable. Diabetes's types are the attributes while numbers "1" and "2" represent those types.

**Example 1.5.** Peoples' Ethnicity is another categorical variable. Races of people vary and are divided into many categories, for example, "African", "American", "Asian", "European". Race is the attribute while the names represent each ethnicity.

## 1.5 Conclusions

In this chapter, the fundamentals of the field of statistics are mentioned. At the beginning of this chapter, the preciousness of data and statistics to the world is explicitly explained. More specifically, common application of statistics and the four steps in solving statistical problems: 1) Formulate a statistical question, 2) Collect data, 3) Analyze data and 4) Interpret results were introduced along with an example for better understanding. Next up, the definitions of basic statistical terms such as 1) data and 2) variables were given along with examples for better understanding. Then, the interpretations and formulas of fundamental statistical measures such as 1) mode, 2) median, 3) mean, 4) variance, 5) standard deviation and 6) z-score are demonstrated. In the end of the first chapter, two categorizations of data are presented: 1) qualitative and 2) quantitative data. As a reminder, quantitative data express a certain quantity, amount or range where the variables consist of numerical values exclusively. On the other hand, qualitative data do not involve mathematical content to their values, but instead data are being categorized based on labels which describe an attribute. In the context of qualitative data, variables consist of multiple categories depending on the nature of each variables. Each of the two types of data come with two examples for further familiarization with the concepts.

# Chapter 2

# Basic Concepts to conduct SEM

## 2.1 Definition of Structural Equation Modeling

Structural Equation Modeling (SEM) refers to a growing family of related procedures which demonstrates relations between observed and latent variables testing hypotheses made by the researcher. SEM consists of two models which apply these procedures, the measurement model and structural model. The measurement model define the way that sets of observed variables interact with each other forming constructs and how these constructs are related to each other always based on the hypothesis. Latent variables mentioned before refer to these exact constructs. Measurement model consists of various latent variable analysis techniques. The structural model specifies the paths between latent variables and observed variables that are not indexes to the latent constructs. Structural model consists of multiple regression models which explain these relations forming paths amongst variables. This is why this technique is called path analysis. Structural equation modeling combines the path analytic and latent variable techniques together and allow for regression models to analyze the relationships among both latent and observed variables. During SEM, measurement model is the first that is conducted for the simple reason that existing constructs must be specified since they are not directly observed by the dataset, but indirectly through the variables. Then, the possibility of a multivariate type of analysis such as the structural model becomes available. SEM is known with variant names. Causal Modeling due to the flexibility of the method which can test hypotheses with variables that are assumed to be connected in an cause-and-effect way and estimate cause effects. Latent Variable Modeling due to the ability to expand the analysis with the addition of latent variables, in contrast with the classical statistic method which measures and estimates exclusively observed variables. The primary data for the usage of SEM are covariance matrices which explains the alternative title as Covariance Structure Modeling (Hoyle (2012)). For example, an owner might want to run a research hypothesizing that a set of observed variables indicating mental health of employees (latent variable) along with age (observed variable) influences the production line and as a consequence the final product. A psychologist might want to test if a specific theory, that includes age and gender (observed variables) and anxiety or depression (latent variables), is verified.

Through his investigation the researcher applies these procedures following fixed steps each time. SEM uses regression models and the scientific method of hypothesis testing to display and analyze relationships between variables. The ordered steps that must be followed in SEM are five but some consider reporting the results a sixth step. As already mentioned, the five steps are model specification, model identification, model estimation, model testing, and model modification.

## 2.2 History of Structural Equation Modeling

SEM is a collection of related techniques that is continuously growing into multiple scientific fields. With that being said, the history of SEM cannot be classified in a specific timeline or origin. To discuss the history of SEM, the following four types of related models and their chronological order of development will be analyzed: Regression model, Path model, Exploratory and Confirmatory factor analysis, and Structural Equation Model.

Linear regression models are the first models that used a correlation coefficient and the Least Squares criterion to compute regression weights. Regression models owes its existence to a formula created by Karl Pearson referring to the correlation coefficient in 1896 that provided an index for the relation between two variables (Pearson (1938)). In 2006, Delucchi used regression analysis to predict student exam scores in statistics (dependent variable) from a series of collaborative learning group assignments (independent variables). The results provided some support for collaborative learning groups improving statistics of exam performance, although not for all tasks (Delucchi (2006)).

Factor analysis (FA) concept was captured in the early years of the 20th century usually credited to Charles Spearman. Charles Spearman realized that correlation coefficients can determine which items correlated or went together united as a set explaining some underlying factors. His basic idea was that if a set of items like that was correlated, then individual responses to the set of items could be summed to yield a score that would measure or define a construct. Spearman was the first to use the term factor analysis in defining a two-factor construct for a theory of general intelligence (Spearman (1927)). Thurstone (1935) further developed the basic idea of Spearman proposing more instruments (sets of items) that yielded observed scores from which constructs could be inferred developing factor models. Most of the aptitude, achievement, diagnostic tests, surveys and inventories in use today were created using factor analytic techniques. The idea of Confirmatory Factor Analysis is partially work of Howe (1955), T. W. Anderson and Rubin (1956), and D. Lawley (1958). In the 1960's, Karl Jöreskog fully developed the CFA method by testing whether a set of items defined a construct. Jöreskog completed his dissertation in 1963, published the first article on CFA in 1969, and subsequently helped develop the first CFA software program (Joereskog (1963); Jöreskog (1969)). Life total of factor analysis extends to more than 100 years and keeps creating measurement instruments in many academic disciplines. Nowadays, CFA uses observed variables derived sets of variables to test the existence of a theoretical construct. Goldberg (1990) used CFA to confirm the Big Five model of personality. His five-factor model consists of extraversion, agreeableness, conscientiousness, neuroticism, and intellect and was confirmed through the application of multiple indicator variables for each of the five hypothesized constructs.

Path model was originally developed by a biologist named Wright (Wright (1918),Wright (1921),Wright (1934)). Wright demonstrated how observed covariances could be related to the parameters of both direct and indirect effects among a set of observed variables. In the procedure, he showed how these effects could be estimated from sample data. Path models use correlation coefficients and multiple regression equations to model more complex relations amongst observed variables. The first application of path models was about animal behavior. Wright also invented path diagrams, or graphical representations of causal hypotheses that we still use to this day (Kline (2015)). Path analysis involves solving a set of simultaneous regression equations that theoretically establish the relations amongst the observed variables in the path model. Unfortunately, the technique of path analysis was forgotten until was subsequently introduced to the behavioral sciences. In the 1950s, econometricians brought to the surface as a form of simultaneous equation modeling (Wold (1954)). In the 1960s, sociologists (O. D. Duncan (1966); Blalock Jr (1961)) and others (Wolfle (2003)) rediscovered

it. Parkerson et al. (1984) conducted a path analysis to test Walberg's theoretical model of educational productivity for $5^{th}$ through $8^{th}$ grade students. The relations amongst the following variables were analyzed in a single model: home environment, peer group, media, ability, social environment, time on task, motivation, and instructional strategies. All of the hypothesized paths among those variables were shown to be statistically significant, providing support for the educational productivity path model. Wright also invented path diagrams, or graphical representations of causal hypotheses that we still use to this day.

Structural equation models combine the structural model (path analysis) and the measurement model (factor analysis) when establishing hypothesized relations amongst latent variables was initially known as the JKW model. The name of this model comes from the early development of SEM models that was integrated in the early 1970s in the work of basically three authors:Jöreskog (1969), Keesling (1972), and Wiley (1973).

The JKW model became popular under the name Linear Structural Relations Model (LISREL). The model became complicated and very quickly computers became an inseparable part of SEM leading to the develop of the first software program, LISREL, in 1973, just over 40 years ago. Jöreskog and van Thillo originally developed the LISREL software program at the Educational Testing Service (ETS) using a matrix command language that used Greek and matrix notation. LISREL is able to analyze models based on the JWK framework, now called SEM. The first publicly available version for mainframe computers, LISREL III, was published in 1974, and LISREL has been subsequently updated many times. By 1993, LISREL8 was released, introducing the SIMPLIS (SIMPle LISrel) command language in which equations were written using variable names, the dialog box interface using pull-down menus, point-and-click features and the path diagram mode which allows the user to draw a program to develop models. In 1999, the first interactive version of LISREL was released. LISREL9 has since been released with new features to address categorical and continuous variables. Karl Jöreskog was recognized by Cudeck et al. (2001), who edited a titular volume known as Festschrift, in honor of his contributions to the field of structural equation modeling. Their volume contains chapters by scholars who addressed the many topics, concerns, and applications in the field of structural equation modeling today, including milestones in measurement models, robustness, reliability, and fit assessment, repeated measurement designs, ordinal data, and interaction models.

LISREL was up to date but it was the only available program which could apply SEM. On top of that, LISREL and the necessary coding to conduct SEM were only known to a limited number of people and were available only on mainframe computers. Until 1993, the program syntax of any SEM modeler mostly used Greek and matrix notation. At that time, many researchers seek for help because of the complex programming and knowledge of the SEM syntax that was required. The introduction of personal computers into people's daily lives has changed reality. The fact that this technology was inexpensive but capable to handle the same analyses as mainframe computers along with the constantly evolving nature of SEM and the increased needs led to further development in the field of programs. Since the mid-1980s and 1990s the development of more statistical software programs including Mplus, R R-Studio, STATA was a phenomenon. User friendliness combined with the capabilities in modern SEM computer tools and other general statistical analyses are important features for every researcher. General statistical analyses of raw data such as correlations, means, missing data, outliers and solutions to deal with these sort of problems are key advantages. Also, most of the programs come with pre-installed datasets ready for import and output and visualization of theoretical models. New SEM software programs are particularly friendly to new researchers. Statistical softwares for personal computers with a graphical user interface are easier to use than their character-based predecessors and contain features similar to other Windows-based software packages, for example, pull-down menus, data spreadsheets and a

simple set of commands. Nowadays, as SEM becomes a must in the toolkit of any kind of researcher more and more computer programs are being created. In this thesis, R-Studio will be used to code, visualize and interpret the SEM results.

By the end of the $19^{th}$ century, the use of SEM techniques was a global event and expanded in many different scientific areas. Examples from this time include works about latent variable models of growth and change over time (T. Duncan et al. (1999)). B. Muthén (1984) describes methods for ordinal data further extended the range of application of SEM. Another major development concerned the convergence of SEM and techniques for multilevel modeling (B. O. Muthén (1994)). The field of structural equation modeling across all disciplines has expanded since 1994. Hershberger (2003) found that between 1994 and 2001 the number of journal articles concerned with SEM and journals publishing SEM research increased. SEM became a popular choice amongst multivariate methods and the journal Structural Equation Modeling became the primary source for technical developments in structural equation modeling, and continues so today. SEM research articles are now more prevalent than ever in professional journals of several different academic disciplines and behavioral sciences (medicine, psychology, business education, etc).

Since the 2000s, there has been a surge of interest in Bayesian methods in the behavioral sciences. Bayesian statistics are a set of methods for the orderly expression and revision of support for hypotheses as new evidence is gathered and combined with extant knowledge. Unlike traditional significance testing, which estimates the probabilities of data under null hypotheses, Bayesian methods can generate estimated probabilities of hypotheses, given the data. They also generally require the researcher to specify the exact forms of the distributions for hypothesized effects (parameters) both prior to synthesizing new data (prior distributions) and after (posterior distributions). Bayesian capabilities are available in some SEM computer tools, such as Amos, Mplus and R.

In summary, based on the history just reviewed, SEM is mature, well-studied set of techniques. With maturity comes the motivation to question the current state of the SEM method and to be open to new perspectives. SEM is a constantly evolving tool and all the community of scientists should be aware of its capabilities. Given the history, it is safe to say that SEM has come of age and with maturity should come awareness of one's limitations, the motivation to compensate for them, and openness to new perspectives. Life is a process of continual learning, so is causal modeling.

## 2.3 Structural Equation Modeling Software

In the mid-1980s, the expansion of SEM family of procedures throughout the centuries has led to nothing but a more subtle growth of computer programs with increasingly expanded capabilities to handle the statistical rigors which SEM methodology demands. This evolve of SEM in combination with the need for more comprehensive answers to investigations are the main reasons why statistical researchers are challenged to further develop SEM software programs. In 1974, original LISREL was the first statistical program which could, in someway, be worthy of expectations and demands of its time period. The ensuing years witnessed such a high demand of SEM that resulted in adaptation of LISREL to the needs. Today, there are several programs from which to choose, including Amos, EQS, R, JASP, Onyx, Mplus, advanced versions of LISREL and the list goes on. Each program is unique and offers features that others might not have available targeting for specific SEM applications. Some of the features are providing statistical analyses of raw data such as means, correlations, missing data conventions, routines for handling missing data and detecting outliers, generate the program's syntax, diagram the model and an environment for import and export of data and figures of a theoretical model. Also, many of the SEM software programs come with

sets of data and program examples that are clearly explained in their user guides. Some SEM software programs provide a pull-down menu with these capabilities, while others come included in a statistics package where they can be computed. SEM programs are either stand-alone or part of a statistical package, while most of them run in the Windows operating system (Schumacker and Lomax (2016)). Additionally, not all choices of computer tools to conduct SEM are free to use. The pricing of a SEM software varies depending on the purpose of its use, the site license arrangements and even whether one is a student or faculty member. Additionally, newer versions and updates which solve issues and provide even more advanced tools might also charged extra. Examples of free computer programs or procedures include Onyx, a graphical environment for creating and testing structural equation models and various SEM packages such as lavaan or sem for R, which is an open-source language and environment for statistical computing and graphics. Commercial options for SEM include Amos, EQS, LISREL, and Mplus, which are all free-standing programs that do not require a larger computing environment (Kline (2015)). Both the researchers' needs, preferences and complexity of the problem indicate which program is suitable to choose in order to conduct the corresponding SEM.

### 2.3.1 R

R is kind of a distant cousin of the commercial program Mplus. R was originally developed by Ihaka and Gentleman (1996) at the University of Auckland. The distant cousin of the commercial program Mplus, R, design purpose is to perform analyses through statistical applications expressed via a programming language. It is supported by Unix, Windows, Mac OS X and Linux operating system. R is a free stand-alone software collaborative and part of the GNU Project. As a result, issues are made known and tracked from the public meaning that multiple people are involved in development. The software is easy and intuitive to use plus it has all the required features for a researcher to conduct SEM. A big part of the program's popularity is due to the advanced thinking of approaching statistical procedures which uses a system of packages. The library of packages are equipped with functions that stand out for their flexible writing of script suitable to conduct statistical computing, data mining, and graphics. A basic R installation has about the same capabilities as commercial programs for general statistical analyses, but there are now thousands of freely available packages that further extend R's statistical repertoire. R does everything, SEM or otherwise therefore it would make sense to say the R can compete the commercial programs. It seems like R is an upgraded version of Mplus program. Between the SEM-specific tools, and other packages that can fill in the remaining functionality. R can do any application and model Mplus can, in many cases better and more efficiently. R has well surpassed Mplus in what it can offer in the SEM world.

The user interacts directly with the R source editor, an environment typing commands in the R language which are then interpreted and executed. Researchers create objects assigning them single or multiple values. An object is categorized by its value into four types: Logical, Numeric, Complex, Character.

- Logical type means that a logical value represents an object, TRUE or FALSE (T or F).
- Numeric type means that any real number represents an object. For example, 56, -42, 0, 2.3, e, pi etc.
- Complex type means that a complex expression represents an object. For example, 5+10i, 7i etc.

- Character type means that a sentence or a character expression closed into double quotes (" ") represent an object. For example, "hi", "24", "Bill" etc.

The above object types and therefore the creation of an object is achieved via the help of operators. R supports a complete set of standard operators which are mainly divided into four categories: Assignment operators, Mathematical operators, Relational operators and Logical operators.

- Assignment operators are the most important since they allow for values to register into objects using the registration arrow on both directions ($\leftarrow$ *and* $\rightarrow$). The value in the edge of the arrowhead is the object name while and the value at the start of the arrow is the value of the object. The users must be aware of certain inviolable rules while choosing object name:

  1. Names cannot start with a number. For example, 3hi is not valid, but hi3 is.
  2. R is case sensitive meaning that a name with a specific uppercase letter differs from the same name using the same lowercase letter. For example, hipeople and Hipeople represent two entirely different objects.
  3. Gaps between words are not allowed, instead the underscore (_) is used. For example, hi people is invalid, while hi_people is valid.
  4. There are some names that cannot be used because they are the names of fundamental functions in R. For example, if, else and for.
  5. Additionally, no special characters (#, $, %, () etc.) or operators ($+, -, /, *$,^ etc.) are allowed throughout the whole length of the name. A syntactically valid name consists of letters, numbers and the dot or underline characters and starts with a letter or the dot not followed by a number. Examples of valid names are object1$\leftarrow$ 24, object2$\leftarrow$"George", object3$\leftarrow$TRUE.

- Mathematical operators such as addition and multiplication ($+, *$), subtraction and division ($-, /$), exponentiation (^), modulus (i.e remainder from division) expressed with %%, integer division (%\%) which carries out math equations, expressions and result in a mathematical type of object (e.g. 0, -2, 5, 7, 56).

- Relational Operators such as less than ($<$), greater than ($>$), less than or equal to ($<=$), greater than or equal to ($>=$), equal to ($==$) and not equal to ($!=$) are used to compare values between objects or expressions of the same type and result in a logical type of object (T or F).

- Logical operators such as logical NOT (!), element-wise logical AND (&), Logical AND (&&), Element-wise logical OR (|) and Logical OR (||) are used to carry out Boolean operations such as AND, OR etc. and result in an logical type of object (T or F). Element-wise means that the symbols & and | produce results having length of the longer operand. But && and || examines only the first element of the operands resulting into a single length logical vector. R example of utilizing &, |, && and || along with the corresponding outputs is demonstrated below.

```
>x <- c(TRUE,FALSE,0,6)
>y <- c(FALSE,TRUE,FALSE,TRUE)
```

```
#Logical operator &
>x&y

Output:
FALSE FALSE FALSE  TRUE

#Logical operator &&
>x&&y

Output:
FALSE

#Logical operator |
>x|y

Output:
TRUE  TRUE FALSE  TRUE

#Logical operator ||
>x||y

Output:
TRUE
```

## 2.4 Correlation

When the data points of two variables are following a straight-line pattern, the variables are suspicious to be somewhat linearly related to each other. Correlation is a statistic which measures the strength of the association between two metric variables which can be positively, negatively or unrelated to each other in a linear way. The correlation measure is denoted by lowercase letter $r$. This point estimate, $r$, is also known as Pearson's $r$ named after Karl Pearson who originally proposed it in 1896.

As already mentioned, correlation examines two quantitative variables, $x$ and $y$. When positive correlation occurs between them, $r$ takes a positive value, indicating that an increase on variable $x$ results in an increase on variable $y$, and the other way around (an increase on variable $y$ results in an increase on variable $x$), and the two variables are said to be positively associated in a linear way. When negative correlation occurs, $r$ takes a negative value, indicating that an increase on variable $x$ results in a decrease on variable $y$, and the other way around (an increase on variable $y$ results in a decrease on variable $x$), and the two variables are said to be negatively associated in a linear way. When two variables are uncorrelated, r is very close to zero, $r = 0$, indicating that the two variables are neutrally related. Unlike regression, that is going to be examined later in this paper, correlation measure doesn't distinct between an independent and a dependent variable. Instead, the degree of the positive or negative nature of the two variables goes both ways. Mathematically, it is said that correlation is the same and is independent of the order of variables, $r_{xy} = r_{yx}$ (Agresti and Franklin (2018)).

As mentioned before, correlation refers to the amount or magnitude of the association between two variables, $x$ and $y$. As far as unit of measure goes, correlation is a standardized statistic. Its range values vary from $-1$ to 1. Therefore, the value of correlation index, $r$,

doesn't depend on the unit of measure of either of the two variables under study. For example, assume variable $x$ is measured in Euro currency and variable $y$ in Dollar currency, and the correlation structure is meant to be examined. If another researcher wants to examine the same $x$ and $y$ expressed in different units of measure, for instance, Yen and Lev, respectively, then the underlying relationship of the two variables represented by correlation $r$ wouldn't change. The closer $r$ is to its lower or upper boundary, $-1$ or 1, respectively, the closer the data points of the two variables tend to form a straight line indicating the presence of a strong linear association between them. Of course, it is implied that the correlation index $r$ has the extreme values of $-1$ and 1 when data points are aligned perfectly. The closer $r$ is to 0, the weaker the linear association.

In Figure 2.1 different values of $r_{xy}$ are displayed. Notice that when $r_{xy} = 1$ and $r_{xy} = -1$, the data points form a perfect straight line with positive and negative slope, indicating a perfect positive and negative linear relationship, respectively. When $r_{xy} = 0.8$ and $r_{xy} = -0.7$, the points tend to form a distinctive straight line with positive and negative slope, indicating a strong positive and negative linear relationship, respectively. When $r_{xy} = 0.4$, the data points tend to form a scattered positive straight line, indicating a medium power positive relationship. Finally, when $r_{xy} = 0$, the data points do not tend to form any straight line at all, but instead they are scattered all over the graph, indicating that there is not any kind of linear relationship between the two variables, $x$ and $y$ (Agresti and Franklin (2018)).

Next, the correlation formula of Pearson's $r$ is going to be demonstrated. In order for the researcher to obtain Pearson's $r_{xy}$ between $x$ and $y$ variables, the necessary calculation of statistics involve the z-scores of each variable, $z_{x_i}$, $z_{y_i}$, along with their product, $z_{x_i} z_{y_i}$, for each observation $i$ of a sample with $n$ size. Then, the researcher must sum up the products of z-scores of each individual $i$ up to $n$ observations, $\sum_{i=1}^{n} z_{x_i} z_{y_i}$, and find the average of the summation by diving this by the sample size, $n$, minus 1, $(\sum z_{x_i} z_{y_i})/(n-1)$. This formula is displayed in Equation 2.1, where $n$ is the number of observations, $\bar{x}$, $\bar{y}$ are the means of $x$, $y$, respectively, and $s_x$, $s_y$ are the standard deviations of $x$ and $y$, respectively.

$$r_{yx} = \frac{1}{n-1} \sum_{i=1}^{n} z_x z_y = \frac{1}{n-1} \sum_{i=1}^{n} \left( \frac{x - \bar{x}}{s_x} \right) \left( \frac{y - \bar{y}}{s_y} \right) \tag{2.1}$$

(Agresti and Franklin (2018))

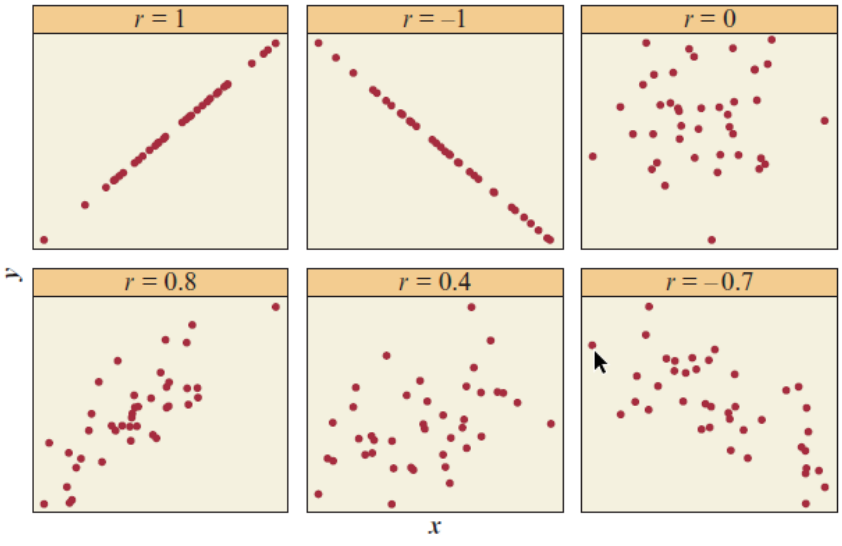



Figure 2.1: Scatterplot of the two variables, $x$ and $y$, at different levels of $r$, adopted from Agresti and Franklin (2018).

Additionally, note that the correlation between a variable and itself is always 1. Empirically, This can be proven through the present example. Assume that someone wanted to examine the correlation between $Taxes$ or $x_1$ and itself. From Equation 2.1, when $y = x = x_1$, then:

$$r_{x_1x_1} = \frac{\sum_{i=1}^{n}(z_{x_1}z_{x_1})}{n-1} = \frac{1}{n-1}\sum_{i=1}^{n}\left[\left(\frac{x_1-\bar{x}}{s_{x_1}}\right)\left(\frac{x_1-\bar{x_1}}{s_{x_1}}\right)\right] = \frac{\sum_{i=1}^{n}\left(\frac{x_1-\bar{x_1}}{s_{x_1}}\right)^2}{n-1} \tag{2.2}$$

The numerical example which is going to be used is obtained from a dataset which has 15 observations. The variables of interested are House distance from the sea, $Income$ and $Taxes$. The correlation matrix which contains the correlation index $r$ between each possible pair of these variables is demonstrated in Table 2.1. For simplicity, assume that $Income$, $Taxes$ and House distance from the sea are denoted with $y$, $x_1$ and $x_2$, respectively. As a reminder, the correlation measure does not distinct between predictors and outcome and thus has mirrored values in the off-diagonal positions of the matrix. From this matrix it can be seen that variables $Income$ and $Taxes$ have a correlation index, $r_{yx1} = 0.9725782$, indicating a very strong positive linear relationship. Variables $Taxes$ and House distance from the sea hold a very strong negative linear relationship since $r_{yx2} = -0.7472380$. Finally, the correlation between $Income$ and House distance from the sea is equal to $r_{x1x2} = -0.8447237$, indicating a very strong negative linear relationship.

Table 2.1: The $3x3$ correlation matrix formula.

| | Income | Taxes | House distance from the sea |
|---|---|---|---|
| Income | 1.00000000 | 0.97257820 | -0.8447237 |
| Taxes | 0.97257820 | 1.00000000 | 0.08014998 |
| House distance from the sea | -0.8447237 | 0.08014998 | 1.00000000 |

In the present example, the sample size is equal to 15 and squared sum of the product of the z-scores of $x_1$, $z_{x_1}^2$, is equal to 14. From Equation 2.2, $r_{x_1x_2} = 14/(15-1) = 1$. The same goes for $y$ and $x_2$ ($r_{yy} = 1, r_{x_2x_2} = 1$).

## 2.5 Covariance

Let's assume there are two observed continuous variables $x$ and $y$. Variables $x$ and $y$ have a mean and standard deviation expressed as $(\bar{x}, s_x)$ and $(\bar{y}, s_y)$, respectively. The covariance given by Equation 2.4 represents the sum of the cross products deviations of pairs of $x$ and $y$ scores from their respective means for each individual $i$, $(\sum_{i=1}^{n}[(x-\bar{x})(y-\bar{y})])$, divided by the sample size $n$ minus 1, known as covariance between $x$ and $y$, $Cov_{xy}$. Note that the covariance formula doesn't differ from the variance one in Equation 2.3. Variance shows how data points of a single variable $x$ variate from the mean. Covariance shows how data points of two variables $x$ and $y$ covariate from their respective means, $\bar{x}$ and $\bar{y}$. That's the difference between variance and covariance. Although, covariance seems like is capturing the magnitude of the linear association between $x$ and $y$ along with their variability measurements, in

reality the direction sign (+ or −) is what really worth interpreting and not the size of the value. As already mentioned in the correlation section, two variables, $x$ and $y$, can either have a linear positive or inverse - negative relationship between them. In the first case, an increase on $x$ results in an increase on $y$ and vice versa. In the first case, an increase on $x$ results in an decrease on $y$ and vice versa. Covariance is not a standardized measure which means that its values do not range from -1 to 1 but instead vary depending on the scales of the original scores. The fact that covariance is affected by the unit of measure of the variables is one of the main reasons why the mathematical sign of the value is the only meaningful point. Any real number can be used to express covariance. So for example, both covariance measures, $Cov_{xy} = 500$ and $Cov_{xz} = 1456378$ are reasonable covariance numbers and, regardless of their numerical value, they indicate a linear positive relationship without any further interpretation available in terms of relationship strength. Thus, the size of the index does not relate with the strength of the relationship between two variables.

$$\boxed{{s_x}^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{2.3}$$

$$\boxed{Cov_{xy} = \frac{\sum_{i=1}^{n}\left[(x_i - \bar{x})(y_i - \bar{y})\right]}{n-1}} \tag{2.4}$$

SEM analysis is done by using a mathematical entity called observed covariance matrix denoted by $S$. The covariance matrix is computed by using the observed values of the dataset under study. The analysis of covariance, and covariance matrices, aim to understand the covariance patterns among a set of observed variables while at the same time explain the most variance possible with the given SEM model. The covariance matrix by definition is symmetric meaning that $Cov_{xy} = Cov_{yx}$ which is expected since the formula itself is a product, $(x_i - \bar{x})(y_i - \bar{y}) = (y_i - \bar{y})(x_i - \bar{x})$ (subsection 2.9.1). On the main diagonal of the matrix the covariance values of the variables are equal to their variance values. In these positions of the matrix the covariance value is computed for the same variable. For example, the covariance for variable $x$, $Cov_{xx} = \sum_{i=1}^{n}[(x_i - \bar{x})(x_i - \bar{x})]/(n-1) = \sum_{i=1}^{n}[(x_i - \bar{x})^2]/(n-1) = s_x^2$ which is the formula in Equation 2.3. The rest of the matrix contains covariance values for each possible pair of the variables under study. The number of unique values which form the matrix is given by the formula $u(u+1)/2$, where $u$ is the number of observed variables of interest. The dimension of the matrix depends on the number of variables analyzed. For example the covariance analysis of three variables will result in a $3 \times 3$ covariance matrix, four variables $4 \times 4$, five variables $5 \times 5$ e.t.c. An example of a $3 \times 3$ covariance matrix is displayed in Table 2.2.

Table 2.2: The $3 \times 3$ covariance matrix S formula.

| | **x** | **y** | **z** |
|---|---|---|---|
| **x** | Cov(x,x)=$s_x^2$ | Cov(x,y) | Cov(x,z) |
| **y** | Cov(y,x) | Cov(y,y)=$s_y^2$ | Cov(y,z) |
| **z** | Cov(z,x) | Cov(z,y) | Cov(z,z)=$s_z^2$ |

# R code - Covariance matrix - Tab:2.2

```
#Covariance matrix - Tab:2.2
>continuousvariables<-cbind(dfv2$Income,dfv2$Taxes,
dfv2$‘House distance from the sea‘)
>print(continuousvariables)
>mycov<-data.frame(cov(continuousvariables))
>names(mycov)<-c("Income","Taxes","House distance from the sea")
>row.names(mycov)=c("Income","Taxes","House distance from the sea")
>mycov
```

As numerical example, assume there is the need for computing the covariance table from the numerical variables House distance from the sea, *Income* and *Taxes*. Since the number of variables is 3, the Table 2.3 will contain $3(3+1)/2 = 6$ unique values. The total number of values is 9. As mentioned already, the 3 diagonal values represent the variances for each variable of the corresponding row or column (they are the same). There are 3 values both below and above the diagonal which represent the covariance of the corresponding variables. Again, House distance from the sea, *Income* and *Taxes* are denoted with $x_2$, $x_1$ and $y$, respectively. As discussed before, as far as interpretation goes, one should notice that $Cov_{x_1x_2}$ and $Cov_{yx_2}$ are equal to $-11885.52$ and $-90922.86$, respectively. Therefore, $Cov_{x_1x_2}$ and $Cov_{yx_2}$ are negative numbers, indicating negative linear relationships between the corresponding variables. The rest of the covariances are positive, indicating that the corresponding pair of variables are positively related in a linear way.

Table 2.3: An example of covariance matrix of 3 variables Income, Taxes and House distance from the sea

| | Income | Taxes | House distance from the sea |
|---|---|---|---|
| Income | 1572571.4 | 226014.29 | -90922.86 |
| Taxes | 226014.3 | 34340.95 | -11885.524 |
| House distance from the sea | -90922.86 | -11885.52 | 7367.267 |

Since the covariance table is symmetric only 3 off-diagonal values, both below or above the diagonal, are unique. This is the reason why sometimes the covariance table is displayed omitting the values either below or above the diagonal (Table 2.4).

Table 2.4: An example of covariance matrix of 3 variables with the upper diagonal omitted.

| | Income | Taxes | House distance from the sea |
|---|---|---|---|
| Income | 1572571.4 | | |
| Taxes | 226014.3 | 34340.95 | |
| House distance from the sea | -90922.86 | -11885.52 | 7367.267 |

## 2.6 Relationship Between Correlation and Covariance

Correlation is related to covariance as a concept and as formula. The formula of correlation can also be computed by using the variances and unique covariances from the covariance matrix. This connection between variance, covariance and correlation is demonstrated Equation 2.5 for variables $x$ and $y$.

$$\boxed{r_{xy} = \frac{Cov_{xy}}{\sqrt{s_x^2 s_y^2}}} \tag{2.5}$$

The unique covariance terms from the covariance matrix in Table 2.4 found in the off-diagonal positions are going to be used to apply Equation 2.5. According to which each unique covariance of pair of variables is divided by the square root of the product of the variances of the corresponding two variables. Variances are found in the main diagonal. By applying this process for the three variables from Table 2.3 the results of the correlations are demonstrated below:
$r_{xx} = 1572571.4/1572571.4 = 1$
$r_{yy} = 34340.95/34340.95 = 1$
$r_{zz} = 346107142.7/346107142.7 = 1$
$r_{xy} = 226014.3/(1572571.4 * 34340.95)^{\frac{1}{2}} = 0,9725$
$r_{xz} = 1671266.4/(1572571.4 * 346107142.7)^{\frac{1}{2}} = 0,07163$
$r_{yz} = 276321.62/(34340.95 * 346107142.7)^{\frac{1}{2}} = 0,0801$

Notice that the denominator in Equation 2.5 contains the standard deviation of variables $x$ and $y$, $s_x$ and $s_y$. Below, both sides of Equation 2.5 are multiplied by the product of standard deviations of variables $x$ and $y$, $s_x s_y$, to derive the formula that connects covariance and correlation. In Equation 2.6, $Cov_{xy}$ is the covariance between $x$ and $y$, $r_{xy}$ is the Pearson correlation of variables $x$ and $y$, and $s_x$ and $s_y$ are the standard deviations of $x$ and $y$, respectively .

$$\boxed{Cov_{xy} = r_{xy} s_x s_y} \tag{2.6}$$

Additionally, notice that both the covariance formula in Equation 2.4 and the correlation one in Equation 2.1 are divided by the sample size minus 1, $n-1$. Additionally, notice that the numerator of the right side in Equation 2.1 is nothing but the covariance between the two variables $x$ and $y$, $\sum(x-\bar{x})(y-\bar{y})$, divided by their corresponding standard deviations, $s_x$ and $s_y$. Consequently, this is the reason why covariance is unit-dependent and correlation is not. Recall that covariance value is biased by the unit of measure while correlation is unbiased. This is a comparative advantage of correlation against covariance.

There are two matrices mainly used as input for SEM, the correlation and the covariance matrix. The basic SEM structure consists of multiple covariance matrices. SEM is more a covariance structure analysis than a correlation structure analysis. In later chapters, where a SEM example will be demonstrated, covariance matrices will be used as input. The computer programs usually when given a correlation matrix as an input they convert it into covariance matrix by default, unless it is specified otherwise. This is because most of the techniques included in the SEM family assume that variables are unstandardized, i.e. variables retain their original unit of measure. On the one hand, covariance $Cov_{xy}$ is biased by the unit of measure of the variables and therefore, Pearson's correlation $r_{xy}$ is normally preferred. But on the other hand, there are many cases in which $Cov_{xy}$ is used over Pearson's $r_{xy}$ because it contains more information. As far as SEM analysis goes, it is safer to analyze covariance matrices over a correlation matrices. As implied, correlation is just a special case of covariance which is also another reason why covariance matrices are more preferable than

correlation matrices in a SEM environment.

## 2.7 Supervised Learning

There are mainly two approaches for problem solving in the field of statistics, supervised and unsupervised learning. For example, assume the researcher collected $N$ size data for two variables $x$ and $y$ and decided to use the following linear model: $y = \alpha + \beta x$. In its simple form, linear regression model approach involves problems in which each observation of the input, $x_i$, corresponds to an output value, $y_i$. Supervised learning in practice attempts to predict future values of $y$ through a learning process done by feeding data, which contain both $x$ and $y$, into the model. Each value of the input variable, $x_i$, is fed into an artificial system, in this case $y = \alpha + \beta x$, which produces the corresponding output value, $y_i$. This learning process is done by a learning algorithm which has the authority to modify both the input and the output value based on the results of the difference between the actual and the predicted output values, $y_i$ and $\widehat{y_i}$, respectively. In general, the term "learning" refers to the estimation of the model's parameters, in this case $\alpha$ and $\beta$, such that the model agrees or fits the data at the best possible degree. The model is considered functional when for a specific value $x_i$ in the existing dataset it yields a corresponding $\widehat{y_i}$ value which is very close to the $y_i$ value in the existing dataset. Empirically, the goal is to minimize the difference between artificial and real outputs so that the model will be applicable to all sets of inputs of the real world (Hastie, Tibshirani, and Friedman (2009)). The researcher splits the dataset and forms two subsets, a $n$ size training set and a $m$ size testing set, where $n > m$ and, of course, $(n + m = N)$. The training set is used to learn or estimate the parameters of the model, which are then put into test, in the testing set. In the optimal scenario, the parameters yield a good fit in the testing set. SEM is not classified into the supervised learning because, in its largest part, is a family of techniques based on the correlation and causality between the variables without distinguishing between input and output. Apart from linear regression model, there are more modeling choices which operate under supervised learning such as logistic regression (James et al. (2013)).

## 2.8 Unsupervised Learning

The other category of statistical problems is called unsupervised learning. In contrast with the previous category, these kind of problems are more challenging. The reason is that for every observation $i$ a vector of measurement $x_i$ is observed but no corresponding particular response $y_i$. Thus, assuming, choosing and fitting a linear regression model is impossible since there is no particular dependent variable that is regressed. The term unsupervised refers to the absence of a response variable that can supervise the researcher's analysis. So since linear regression model is not feasible how is statistical analysis conducted in these situations? Instead of focusing the linearly relate on variable to other the trick here is to understand the relationships between the variables or between the observations. There is one statistical method called cluster analysis. The idea is to identify and classify particular groups or clusters of data into groups. For example, consider a dataset which includes multiple attributes of customers such as family income, zip code, and shopping habits. Suppose the researcher wanted to realize if customers were distinguished by the amount of money they spend. In this case, he wouldn't be able to run a linear regression because he doesn't have the dependent variable spending magnitude. If this information was available then the supervised analysis could kick in and solve the problem instantly. However, cluster analysis can help deal with this problem. The researcher will try to identify cluster patterns

among customers and separate them into different groups based on the existing variables. The researcher must have the ability to fully understand the essential meaning of the variables he has available. For example, in this case shopping habits is a variable somewhere close to spending magnitude increasing the probability to observe distinct groups with respect to it. Besides cluster analysis, SEM also helps solving issues on an unsupervised level. The reason is because SEM is the most popular multivariate technique worldwide providing with the maximum amount of information about the relationships between the observed and unobserved or latent variables without any of them being a well-defined response variable.

## 2.9 Principal Component Analysis

Broadly speaking, factor analysis can serve to reduce the dimensionality of data, or as an approach to modeling measurement error and understanding underlying constructs. Factor analysis is a very strong tool for the researcher since it gives him the ability to reduce the number of variables in his experiment while retaining as much information about the originals as possible making his model more practical and easier to analyze and understand. One of the most commonly used factor-analytic technique is Principal Components analysis (PCA).

For instance, a hypothesis could be that a researcher believes that variables X1 and X5 are taken into account under a common domain that is distinct from the one measured by variables X2 and X4. In SEM, it is relatively easy to specify a model where X1 and X5 are indicators of one factor, X2 and X4 are indicators of another factor. If the fit of the model just described to the data is poor, the hypothesis behind its specification will be rejected.

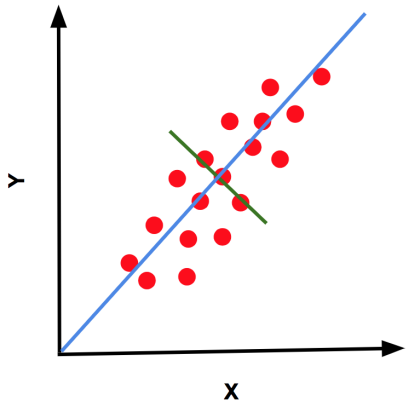



Figure 2.2: The blue and green line represent the first and second principal component, respectively.

Principal Components Analysis (PCA) is one of the most popular unsupervised technique which is mainly used for dimension reduction. Dimension reduction is referred to the process which results in a low dimensional set of features derived from a larger set of variables. The new variables are constructed as linear combinations of the original variables. These combinations are done in such a way that the variables derived are uncorrelated. The amount of new variables, or PC's, is equal to the number of variables in the original dataset. PCA is justified an unsupervised approach, since it involves only a set of features or variables $x_1, x_2, ..., x_n$ and no associated response variable, $y$. PCA structure involves creating the principal components in such way that most of the information is squeezed compressed or collapsed on the first PC's. The $1^{st}$ component contains the maximum information possible, then maximum remaining information is squeezed in the $2^{nd}$ PC and so on. The $1^{st}$ principal

component direction of the data is the one which the cases have the most variance. This is the line that captures most information of the data. Mathematically speaking, the $1^{st}$ principal component line is the one that maximizes the average of the squared distances from the projected points to the origin. The $2^{nd}$ PC is calculated with the restrictions of being uncorrelated with the $1^{st}$ and that it accounts for the next highest variance. The first and second component is demonstrated in Figure 2.2 (https://bit.ly/2UVNLuB).

### 2.9.1 Eigenvectors - Eigenvalues

Square matrices have 2 important measures: eigenvectors and eigenvalues. Essentially, eigenvectors and eigenvalues are nothing but a set of vectors and numbers, respectively. Combined they provide the eigen-decomposition of a matrix, which is the core structure of the specific matrix. More specifically, covariance, correlation, or cross-product matrices are involved while talking for eigen-decomposition. The utility of eigen-decomposition comes from the ability to find the maximum (or minimum) of these functions.

Eigenvectors and eigenvalues are also known as known as characteristic vectors and characteristic roots, or latent vectors and latent roots, respectively. Each eigenvector $u$ when is multiplied by the matrix $A$ under investigation the result is a vector proportionally equal to itself. Additionally, the coefficient of this proportionality is called eigenvalue and is denoted with $\lambda$. Consequently, among the many methods and definitions of the eigenvectors and eigenvalues, the most common that stands out is the following: eigenvector of a matrix $A$ is a vector $u$ that satisfies the equation in Equation 2.7, where $\lambda$ is a scalar, also known as the eigenvalue connected to the eigenvector (Abdi and Williams (2010)).

$$\boxed{Au = \lambda u} \tag{2.7}$$

The equation above can also be written as:

$$\boxed{Au - \lambda u = 0 \rightarrow u(A - \lambda) = 0 \rightarrow u(A - \lambda I) = 0}$$

From Equation 2.7, it can also be stated that matrix $A$ can only have an eigenvector $u$ only if the length of the specific vector is changed when it is multiplied by matrix $A$. As an instance, assume matrix $A$ is the one displayed in Equation 2.8.

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \tag{2.8}$$

The eigenvalues of matrix $A$ are:

$$u(A - \lambda I) = 0 \rightarrow \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = 0 \rightarrow \begin{bmatrix} 2 - \lambda & 3 \\ 2 & 1 - \lambda \end{bmatrix} = 0 \rightarrow \lambda^2 - 3\lambda - 4 = 0 \tag{2.9}$$

$$\lambda_1 = 4, \lambda_2 = -1$$

Assume the matrix which includes the parameters is named matrix $B$. To find the eigenvectors for each eigenvalue all someone has to do is to, first, input the eigenvalues into matrix $B$ to obtain it, and second, multiply this matrix by a $2 \times 1$ matrix $X$ with parameters $x_1$ and $x_2$ so that the following equation is justified:

$$\boxed{BX = 0} \tag{2.10}$$

In the example, according to Equation 2.10 above:

$$\begin{bmatrix} -2 & 3 \\ 2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \left[\begin{array}{cc|c} -2 & 3 & 0 \\ 2 & -3 & 0 \end{array}\right] \rightarrow \left[\begin{array}{cc|c} -2 & 3 & 0 \\ 0 & 0 & 0 \end{array}\right] \rightarrow -2x_1 + 3x_2 = 0 \tag{2.11}$$

For $x_2 = 2$, $x_1 = 3$, and thus the eigenvector $u_1$ when $\lambda_1 = 4$ is the following:

$$u_1 = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \tag{2.12}$$

Following the same procedure, when $\lambda_2 = -1$, the eigenvector $u_2$ is the following:

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{2.13}$$

Most of the times the eigenvectors are being normalized, this is displayed in Equation 2.14,where $u^T$ is the transpose of $u$.

$$u^T u = 1 \tag{2.14}$$

It is very common to place the eigenvectors of a matrix $A$ in another matrix $U$. Therefore, each eigenvector of $A$ is found in the columns of $U$. In a diagonal matrix $D$ the eigenvalues are found the diagonal positions of the matrix. Therefore Equation 2.7 can be written as following:

$$\boxed{AU = DU} \tag{2.15}$$

which is equivalent to:

$$\boxed{A = UDU^{-1}} \tag{2.16}$$

By applying Equation 2.16 in the previous example the following results are obtained in Equation 2.17.

$$UDU^{-1} = \begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ -4 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \tag{2.17}$$

The eigen-decomposition of matrix $A$ is formed from the eigenvectors and eigenvalues. Finally, note that not all matrices have an eigen-decomposition.

#### 2.9.1.1 Positive Semi-Definite Matrices

Another type of matrices which is occurred frequently in statistics are the positive semi definite matrices. These kind of matrices always have a particular eigen-decomposition structure. First things first, a matrix $A$ is called positive semi definite when it can obtained as the product of a matrix by its transpose. By the definition, positive semi definite are always symmetric. The most famous examples of positive semi definite matrices are cross-product, covariance and correlation. Mathematically, the definition can be expressed by Equation 2.18, where matrix $X$ is a real matrix (in the sense that it contains real numbers) and $X^T$ is the inverse of $X$. The most famous examples of positive semi definite matrices are cross-product, covariance and correlation (Abdi and Williams (2010)).

$$\boxed{A = XX^T} \tag{2.18}$$

One of the most important features of a positive semi definite matrix is that all its

eigenvalues are equal to zero or positive. In simple words, an eigenvalue of a positive semi definite matrix can never be a negative number. Additionally, the eigenvectors of such matrices are orthogonal when their eigenvalues are different.

At this point it is worth explaining when two eigenvectors are orthogonal, or equantively, two eigenvectors are uncorrelated. Two eigenvectors are orthogonal with each other when they are perpendicular. Mathematically, this means that their dot product is equal to zero. An example of a pair of eigenvectors ($B$ and $C$) which are orthogonal is displayed in Equation 2.19.

$$BC = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} = (1 \cdot 1) + (0 \cdot 3) + (-1 \cdot 1) = 0 \tag{2.19}$$

The property of the eigenvectors which correspond to different eigenvalues to be orthogonal allow to save all the eigenvectors in an orthogonal matrix. A matrix $A$ is orthogonal when the product with its transpose results in a identity matrix (Abdi and Williams (2010)). Mathematically, this is can be expressed by the following equation:

$$\boxed{AA^T = I} \tag{2.20}$$

A popular example of an orthogonal matrix is the one which has $2X2$ dimension, 0 in the diagonal positions and 1 in the off-diagonal. The proof of this matrix, assuming it is called $A$, being orthogonal is displayed below:

$$AA^T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} (0 \cdot 0) + (1 \cdot 1) & (0 \cdot 1) + (1 \cdot 0) \\ (1 \cdot 0) + (0 \cdot 1) & (1 \cdot 1) + (0 \cdot 0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \tag{2.21}$$

Therefore, the positive semi definite matrix $A$ can be written in the form in Equation 2.22, where $U$ is the matrix which has the normalized eigenvectors.

$$\boxed{A = UDU^T, \text{where } U^TU = I \text{ is true.}} \tag{2.22}$$

In summary, the calculation of the eigenvectors, $\bar{x_i}$, and their corresponding eigenvalues, $\lambda_i$, of any square $n \times n$ matrix by hand involves 5 steps:

1. Multiply the $n \times n$ identity matrix by the scalar λ.
2. Subtract the identity matrix multiple from the matrix A.
3. Find determinant of the matrix and the difference.
4. Solve for the values of λ that satisfy the equation det(A-λI)=0
5. Solve for the corresponding vector to each λ.

For example, assume the following $2 \times 2$ matrix $A$:

$$A = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \tag{2.23}$$

The goal is to compute the eigenvectors and eigenvalues of the matrix $A$. Of course this will be achieved through the application of the steps above:

1. First, $\lambda$ is multiplied by the identity matrix $I$:

$$\lambda I = \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \tag{2.24}$$

2. Subtract $\lambda I$ from the matrix A:

$$A - \lambda I = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix} \tag{2.25}$$

3. Determinant of $A - \lambda I$ matrix:

$$det(A - \lambda I) = det \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix} = (7-\lambda)(-1-\lambda) - (3)(3) = \lambda^2 - 6\lambda - 16 \tag{2.26}$$

4. Set this determinant equal to zero and solve for λ:

$$\lambda^2 - 6\lambda - 16 = 0 \rightarrow (\lambda - 8)(\lambda + 2) = 0 \rightarrow \lambda_1 = 8, \lambda_2 = -2 \tag{2.27}$$

   $\lambda = 8$ and $\lambda = -2$ are the eigenvalues of the matrix A.

5. To find the corresponding eigenvectors, first you input each of the $\lambda's$ into the $A - \lambda I$ matrix.

$$\begin{bmatrix} 7-8 & 3 \\ 3 & -1-8 \end{bmatrix} - \begin{bmatrix} -1 & 3 \\ 3 & -9 \end{bmatrix} \tag{2.28}$$

   By labeling the resulted $2 \times 2$ matrix, solve $B\bar{x} = 0$.

$$\begin{bmatrix} -1 & 3 \\ 3 & -9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \left[\begin{array}{cc|c} -1 & 3 & 0 \\ 3 & -9 & 0 \end{array}\right] \rightarrow \left[\begin{array}{cc|c} -1 & 3 & 0 \\ 0 & 0 & 0 \end{array}\right] \tag{2.29}$$

   Transitioning from matrix to algebraic formation, the resulted matrix can be written as:

$$-x_1 + 3x_2 = 0 \rightarrow 3x_2 = x_1 \tag{2.30}$$

   Picking a random value for $x_2 = 1$ gives that $x_1 = 3$. This is the eigenvector, $\bar{x_1}$, that corresponds to eigenvalue of 8 ($\lambda_1 = 8$).

   Repeating this step for $\lambda_2 = -2$:

$$\begin{bmatrix} 7-(-2) & 3 \\ 3 & -1-(-2) \end{bmatrix} - \begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix} \tag{2.31}$$

   By labeling the resulted $2 \times 2$ matrix, solve $B\bar{x} = 0$.

$$\begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \left[\begin{array}{cc|c} 9 & 3 & 0 \\ 3 & 1 & 0 \end{array}\right] \rightarrow \left[\begin{array}{cc|c} 0 & 0 & 0 \\ 3 & 1 & 0 \end{array}\right] \tag{2.32}$$

   Transitioning from matrix to algebraic formation, the resulted matrix can be written as:

$$3x_1 + x_2 = 0 \rightarrow x_2 = -3x_1 \tag{2.33}$$

   Picking a random value for $x_1 = 1$ gives that $x_2 = -3$. This is the eigenvector, $\bar{x_2}$, that corresponds to eigenvalue of -2 ($\lambda_2 = -2$).

6. There is an extra optional step in which the result is being confirmed. In order for the specific values to represent the eigenvector and eigenvalue of the specific square $2 \times 2$ A matrix, the following equation should be satisfied: $\mathrm{A}x_i = \lambda_i x_i$. Indeed, the results are being confirmed as shown below:

   For $\lambda_1 = 8$:

$$\mathrm{A}x_i = \lambda_i x_i \rightarrow \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 8 \begin{bmatrix} 3 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 24 & 8 \end{bmatrix} = \begin{bmatrix} 24 & 8 \end{bmatrix} \tag{2.34}$$

   For $\lambda_2 = -2$:

$$\mathrm{A}x_i = \lambda_i x_i \rightarrow \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -3 \end{bmatrix} = -2 \begin{bmatrix} 1 & -3 \end{bmatrix} \rightarrow - \begin{bmatrix} 2 & 6 \end{bmatrix} = - \begin{bmatrix} 2 & 6 \end{bmatrix} \tag{2.35}$$

### 2.9.2 Linear Algebra

Mathematically, the idea of principal component analysis is to summarize p-dimensional vectors into a q-dimensional subspace. The summary will be the projection of the original vectors on to q directions, or principal components distributed in the space. As already mentioned, the first principal component is found by projecting the data points which maximize the variance, the second principal component is the direction which maximizes variance among all directions orthogonal to the first. The $k^{th}$ component is the variance-maximizing direction orthogonal to the previous $k-1$ components. There are $p$ principal components in all (Shalizi (2013)).

As may already have been implied, in PCA the data table which consists of $I$ observations and $J$ variables is used as an input on a $I \times J$ matrix $X$. Of course, this matrix consists of $i \times j$ independent elements, $x_{ij}$ and has rank $L$ where $L \leq minI, J$.

The matrix $X$ has a specific form which follows the theory of singular value decomposition. This structure is presented in Equation 2.36, where where $P$ is the $I \times L$ matrix of left singular vectors, $Q$ is the $J \times L$ matrix of right singular vectors, and $\Delta$ is the diagonal matrix of singular values.

$$\boxed{X = P\Delta Q^T} \tag{2.36}$$

SVD is a generalized version of the eigen-decomposition. A rectangular matrix through SVD consists of 3 matrices: two orthogonal matrices (see later into this section) and one diagonal matrix.

In Equation 2.36, $P$ consists of the eigenvectors of the matrix $XX^T$ and its columns called left singular vectors of $X$ and $Q$ consists of the eigenvectors of the matrix $X^TX$ and its columns called right singular vectors of $X$. Finally, $\Delta$ is the diagonal matrix of the singular values. Its important to state that $\Delta = \Lambda^{\frac{1}{2}}$, where $\Lambda$ is the diagonal matrix of the eigenvalues of matrix $XX^T$ and of the matrix $X^TX$ (Abdi and Williams (2010)).

In a typical PCA the data table is always modified according to the type of PCA. As a first step, $X$ columns will be centered and the result will be a matrix where each column has mean equal to zero. Mathematically, that $X^T1 = 0$, where where 0 is a $J$ by 1 vector of zeros and 1 is an $I$ by 1 vector of ones. From this point on, the researcher has mainly two options. The first option is to divide each element of matrix $X$ by $\sqrt{I}$ making $X^TX$ a covariance matrix and, therefore, conducting a covariance PCA. The second option is to standardize each variable by diving it by the square root of the sum of all its squared elements making $X^TX$ a correlation matrix and conducting a correlation PCA.

The components of the PCA are derived with the help of SVD of the data table $X$. To

be more precise, by applying Equation 2.36, the $I \times L$ matrix of factor scores, denoted $F$, is calculated as following:

$$\boxed{F = P\Delta} \tag{2.37}$$

Matrix $Q$ can also be called loading matrix and contains the coefficients of the linear combinations which are used to calculate the factor scores. Matrix $F$ is also called projection matrix because when $X$ is multiplied by $Q$, it gives the values of the projections of the observations on the PC's. The proof to the latter is that when Equation 2.36 and Equation 2.37 are together taken into consideration then:

$$\boxed{F = P\Delta = P\Delta Q^T Q = XQ} \tag{2.38}$$

Matrix $X$ can also been read as the product of matrix $F$, which contains the factor scores, and the loading matrix $Q$. The decomposition in Equation 2.39 is called by the name bilinear decomposition of $X$:

$$\boxed{X = FQ^T} \tag{2.39}$$

The foundations of principal component analysis is linear algebra and in particular eigen-decomposition which was mentioned in subsection 2.9.1. PCA is a problem of mathematical optimization and the statistical properties of the eigen-decomposition is the core factor behind the solution. The goal in PCA is to obtain factor scores which: a) are linear combinations of the matrix $X$, b) retain the maximum amount of original information or variance, c) at the same time the sets of these factor scores are pairwise orthogonal. With PCA being an optimization problem, constraints must be taken into consideration. The main constraint of the problem is that the coefficients of the linear combinations must be finite. This means that the sum of squares of the coefficients of each linear combination to must be equal to unity. This amounts to defining the factor score matrix as demonstrated in Equation 2.40, where matrix $Q$ is the coefficient matrix of the linear combinations which the researcher is looking for (Abdi and Williams (2010)).

$$\boxed{F = XQ} \tag{2.40}$$

Mathematically the constraint is that matrix $F$ must be an orthogonal matrix (Equation 2.41) and that matrix $Q$ is orthogonal as well (Equation 2.42).

$$\boxed{F^T F = Q^T X^T XQ} \tag{2.41}$$

$$\boxed{Q^T Q = I} \tag{2.42}$$

Now that the problem has been explicitly explained all is left is the selection of the mathematical method which is going to be used. The answer to this question is Lagrangian multipliers. To be more specific the constraint in Equation 2.42 will be rewritten with a diagonal matrix $\Lambda$ of Lagrangian multipliers. The result is demonstrated in Equation 2.43.

$$\boxed{\Lambda(Q^T Q - I)} \tag{2.43}$$

The final Lagrange equation is formed as demonstrated in Equation 2.44. As a reminder, the expression `trace[]` gives the sum of the diagonal elements of a square matrix as an output.

$$\boxed{\mathscr{L} = trace[F^T F - \Lambda(Q^T Q - I)] = trace[Q^T X^T X Q - \Lambda(Q^T Q - I)]} \quad (2.44)$$

As the next step, in order to calculate the values of $Q$ which maximize $\mathscr{L}$, the derivate of $\mathscr{L}$ relative to $Q$ is taken and set to zero:

$$\frac{d\mathscr{L}}{dQ} = 0 \rightarrow 2X^T XQ - 2Q\Lambda = 0 \rightarrow X^T XQ = Q\Lambda \rightarrow X^T X = Q\Lambda Q^T \quad (2.45)$$

This is an eigen-decomposition problem since $\Lambda$ is diagonal. Automatically this makes $\Lambda$ the matrix which has the eigenvalues of the positive semi definite matrix $Q^T Q$ with descending order and $Q$ is the eigenvector matrix of $Q^T Q$ connected to $\Lambda$ (Abdi and Williams (2010)).

It is also true that the variance of the factors scores is equal to the eigenvalues. This can be proven since:

$$F^T F = Q^T X^T X Q = Q^T Q \Lambda Q^T Q = \Lambda \quad (2.46)$$

The fact that the trace of $X^T X$ is equal to the sum of the eigenvalues proves that the factor scores obtained by the first principal component contain the largest amount of the original variance as possible. This means that the factor scores obtained by the second principal component contain the maximum amount of information possible which is left undetected by the first PC. The same goes for the rest of the principal components (Abdi and Williams (2010)).

### 2.9.3 Contribution of a case to a PC

As already mentioned in previous section, the sum of the squared factor scores for a PC is equal to the eigenvalue which is connected to the specific PC. Consequently, how important a case $i$ is for a component $k$ depends on a measure obtained when diving the squared factor score of $i$, ${f^2}_{i,k}$, by the eigenvalue connected with the $k^{th}$ component, $\lambda_k$. This measure is called contribution of the observation $i$ to component $k$ and denoted $contr_{i,k}$ in Equation 2.47:

$$\boxed{contr_{i,k} = \frac{{f^2}_{i,k}}{\lambda_k}} \quad (2.47)$$

$contr_{i,k}$ is a standardized measure which means that it varies between 0 and 1 and, logically, the sum of the contributions of all observations is equal to 1. Of course, the larger the measure of the contribution, the more the observation contributes to the component. The cases with high value of the contribution index can help as indicators to identify the characteristics and label a component (Abdi and Williams (2010)).

### 2.9.4 Squared Cosine of a PC with a case

Another measure is the squared cosine which is an indication of how important is a component for a specific case. Mathematically, squared cosine shows how a component contributes to the squared distance of the case to the origin. The formula of the squared cosine, $cos^2_{i,k}$, is displayed in Equation 2.48, where ${d^2}_{i,g}$ is the squared distance of a specific case to the origin. PC's with a large value of $cos^2_{i,k}$ are part of a relatively large portion to the total distance and therefore these PC's are important for that case (Abdi and Williams (2010)).

$$\boxed{cos^2_{i,k} = \frac{{f^2}_{i,k}}{{d^2}_{i,g}}} \tag{2.48}$$

$cos^2$ is a useful tools when it comes to finding the components that are essential to interpret both active and supplementary cases.

### 2.9.5 PCA Loadings

The common information between a variable and a component is reflected by their correlation. While conducting PCA this correlation is called loading. The sum of the squared coefficients of correlation between a variable and all the components is equal to 1. Therefore, between the squared and the original loadings, the first are easier to interpret since they represent the proportion of the variance of the variables explained by the PC's. The content of matrix $Q$ are called loadings (Abdi and Williams (2010)).

## 2.10 Lab: PCA

### 2.10.1 Principal Component Analysis using prcomp with cor=T

For this example, the numerical continuous variables of the dataset used also in regression analysis will be used. To be more specific, the variables are demonstrated in Table 2.5.

Table 2.5: The numerical continuous variables for PCA.

| Variables |
|---|
| Age |
| DailyRate |
| DistanceFromHome |
| HourlyRate |
| MonthlyIncome |
| MonthlyRate |
| NumCompaniesWorked |
| PercentSalaryHike |
| TotalWorkingYears |
| TrainingTimesLastYear |
| YearsAtCompany |
| YearsInCurrentRole |
| YearsSinceLastPromotion |
| YearsWithCurrManager |

The following commands in R separated numerical continuous variables from the rest:

```
# Selecting variables, removing EmployeeNumber no use in PCA
>pcanumdf<-numdf[,-15]

# Checking the variables are all in place.
>names(pcanumdf)
```

The following command in R performs principal component analysis, notice how scale is equal to true to use the standardized version of the dataset.

```
#Conducting PCA
>prcomp.pca <- prcomp(pcanumdf,scale=T)
>summary(prcomp.pca)
```

In order to decide which PC's are accounting for most of the information both numerical and visualization tools will be used. First, the command `summary(prcomp.pca)` will reveal the standard deviation, the proportion of the variance and the cumulative proportion of each of the principal components. The results are demonstrated below in Table 2.6. The first column is the standard deviation of each single one of the principal components, a measure of variability across each of the PC's. The second column is the proportion of variance, all the variability in the original data explained away by each PC. For example, as far as PC1 goes, about 33% of the variance is explained by it. Finally, cumulative proportion adds up all the explained variance up to a specific number of PC's. For example, the value of cumulative proportion in the $2^{nd}$ row shows the total variance up to the $2^{nd}$ PC. This table implies that PC1 and PC2 are potentially the ones that the researcher should retain because they account for a large proportion of the total variance.

Table 2.6: The measures which point out the importance of components.

| | Standard deviation | Proportion of Variance | Cumulative Proportion |
|---|---|---|---|
| PC1 | 2.0042 | 0.2869 | 0.2869 |
| PC2 | 1.2846 | 0.1179 | 0.4048 |
| PC3 | 1.03368 | 0.07632 | 0.48111 |
| PC4 | 1.02573 | 0.07515 | 0.55626 |
| PC5 | 1.0047 | 0.0721 | 0.6284 |
| PC6 | 0.99622 | 0.07089 | 0.69925 |
| PC7 | 0.9764 | 0.0681 | 0.7673 |
| PC8 | 0.95786 | 0.06554 | 0.83288 |
| PC9 | 0.8500 | 0.0516 | 0.8845 |
| PC10 | 0.72849 | 0.03791 | 0.92239 |
| PC11 | 0.68545 | 0.03356 | 0.95595 |
| PC12 | 0.53219 | 0.02023 | 0.97618 |
| PC13 | 0.43975 | 0.01381 | 0.99000 |
| PC14 | 0.3743 | 0.0100 | 1.00000 |

Researchers often make plot the eigenvalues in decreasing order. This plot is called scree plot. In a scree plot, the first principal component occurs where the highest eigenvalue takes place followed by the rest of the PC's with lower eigenvalues. Scree plots are also called elbow plots. The name is justified because the threshold for identifying the majority of the variation is where the elbow appears. It is often considered a rather objective but effective at the same time technique to distinct the appropriate number of principal components. This exact visualization tool, the scree plot, is demonstrated in Figure 2.3, according to which the first two principal components account for a reasonable amount of variance (Shalizi (2013)).

The command `attributes(prcomp.pca)` displays all the different outputs of the PCA. By using this command, the attributes are demonstrated below. `"sdev"` is the standard deviation is of each of the principal components, `"rotation"` displays the matrix which contains the weights or loadings of each variable related to every principal component, `"x"` is the matrix which contains the score of each observation. Finally, `"center"` and `"scale"` take the values FALSE or TRUE. For reasons considering the robust of PCA both `"center"` and `"scale"` in this example are equal to TRUE.

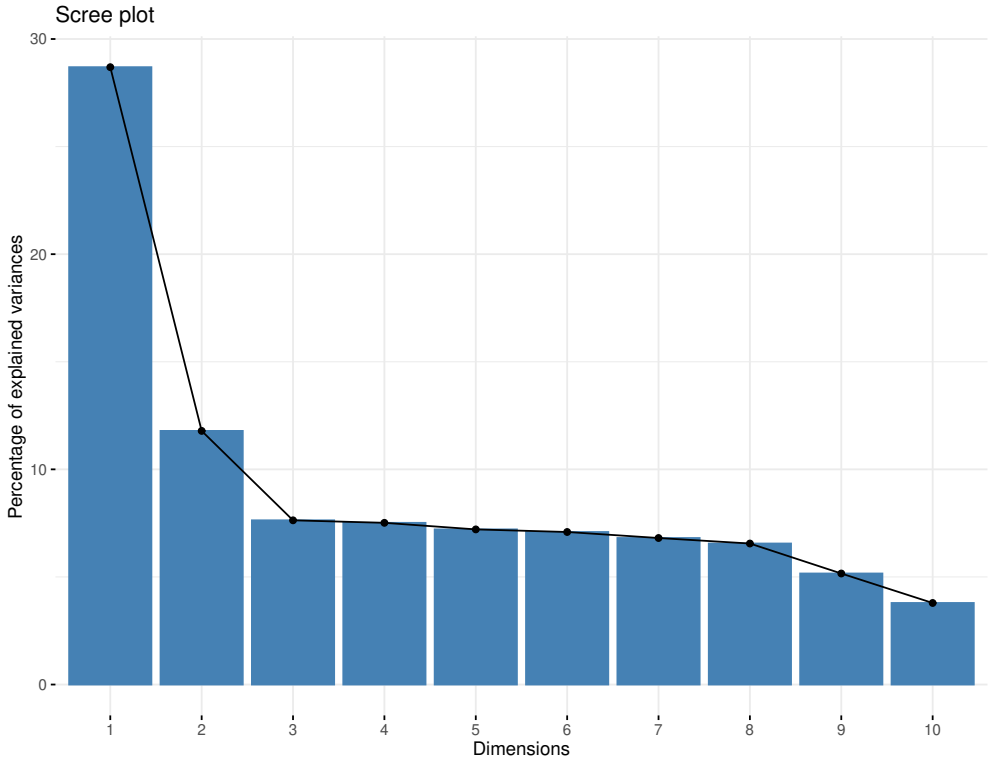



Figure 2.3: Scree plot of the PCA.

```
Call:
attributes(prcomp.pca)

Output:
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

Starting off with the rotation. Rotation command is producing the loadings or weight matrix w. The loadings represent the sign of the projection on to each component and its a percentage value. A table of the loadings can help examine the correlation between the variables and group them based on their sign. This allows for the researcher to uncover patterns between different groups of variables. In Table 2.7, the table is displaying the loadings of the numerical dataset. Notice that YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager, TotalWorkingYears, MonthlyIncome and Age are all positively projected on to PC1 with 0.443443529, 0.391353065, 0.344322397, 0.386171187, 0.415285665, 0.360622909, and 0.280157344, respectively. The rest of the variables do not project on to PCA1 at all.

To better interpret the conclusions of the loadings the biplot tool is frequently used. Essentially, what biplot does is plotting all the data points using PCA2 against PCA1 scores as coordinates. Along with the data points, the eigenvector of each variable of PCA1 and PCA2 is plotted, or also called projections of the original variables on to PCA1 and PCA2. A biplot of the PCA is demonstrated in Figure 2.4.

Table 2.7: The table of loadings of the numerical dataset regarding the first two PC's using prcomp.

| | PC1 | PC2 |
|---|---|---|
| Age | 0.280157344 | -0.472170158 |
| DailyRate | -0.006815197 | -0.077962430 |
| DistanceFromHome | 0.004812032 | 0.041564987 |
| HourlyRate | -0.011288550 | -0.062668026 |
| MonthlyIncome | 0.360622909 | -0.290395305 |
| MonthlyRate | 0.001123298 | -0.086158010 |
| NumCompaniesWorked | 0.030991906 | -0.560133264 |
| PercentSalaryHike | -0.015351368 | 0.004618486 |
| TotalWorkingYears | 0.415285665 | -0.318115831 |
| TrainingTimesLastYear | -0.010993402 | 0.092457674 |
| YearsAtCompany | 0.443443529 | 0.213079968 |
| YearsInCurrentRole | 0.391353065 | 0.279423881 |
| YearsSinceLastPromotion | 0.344322397 | 0.198658357 |
| YearsWithCurrManager | 0.386171187 | 0.295138965 |


Figure 2.4: The biplot of the two first components of the PCA.

Due to the large sample size of the dataset, the lines are indistinguishable and the plot is not so clear. So instead, in Figure 2.5, the same biplot is displayed but with the data points omitted and the variables retained. The way to read and understand the latter bitplot is as follows: a) Looking the plot x-axis wise, as an individual has higher value for PC1, the direction of the eigenvectors indicate if a variable is also increasing or decreasing, b) Looking the plot y-axis wise, as an observation has higher value for PC2, the direction of the arrows indicate if a variable is also increasing or decreasing. Adjusting to the current example, according to the latter biplot, it seems that as an individual has higher value for PC1, variables YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager, MonthlyIncome, TotalWorkingYears and Age are generally increasing. It is important to notice that the variables YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager are increasing with an increasing rate all towards the same direction which is an indication that they could group together and be represented by PCA1. In contrast variables MonthlyIncome, TotalWorkingYears and Age are increasing with descending rate. The variable NumCompaniesWorked stays constant while inviduals take higher values in PCA1. In a same fashion, looking at y-axis this time as PCA2 takes higher values, variables MonthlyIncome, TotalWorkingYears, NumCompaniesWorked and Age are decreasing while YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager are increasing with descending rate.

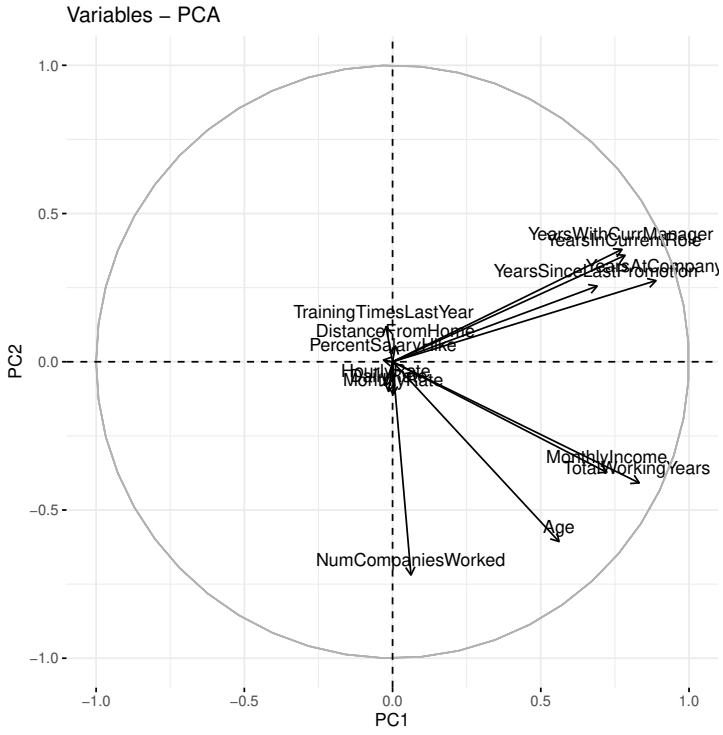



Figure 2.5: The biplot of the two first components of the PCA, but with the data points omitted for better visual result.

The code which produced the both the scree plot and the two biplots above is the following:

```
>fviz_eig(prcomp.pca) #Scree plot
#Biplot with the data points hidden
>fviz_pca_var(prcomp.pca,xlab = "PC1",ylab = "PC2")
#Biplot with the data points graphed
>fviz_pca_biplot(prcomp.pca,xlab = "PC1",ylab = "PC2")
```

As already mentioned, the eigenvalue of a principal component is associated with the amount of information out of the total explained by the specific PC. The larger the eigenvalue, the larger the contribution of the principal component to the total variance. In the example, the eigenvalues are demonstrated in Table 2.8. It is obvious that the first two PC's are accounting for most of the variance since its values are relatively larger than the rest of the eigenvalues.

Table 2.8: The table of eigenvalues or variance of each principal component using prcomp.

| | Eigenvalue (Variance) |
|---|---|
| PC1 | 4.0167738 |
| PC2 | 1.6502052 |
| PC3 | 1.0685042 |
| PC4 | 1.0521201 |
| PC5 | 1.0094108 |
| PC6 | 0.9924579 |
| PC7 | 0.9533497 |
| PC8 | 0.9174969 |
| PC9 | 0.7224654 |
| PC10 | 0.5306975 |
| PC11 | 0.4698454 |
| PC12 | 0.2832233 |
| PC13 | 0.1933840 |
| PC14 | 0.1400658 |
| Sum | 14.000000 |

The correlation between the variables and the principal components, or also called coordinates is another important PCA measure. Recall, that coordinates of PC1 and PC2 are the numbers used to plot the eigenvector of each variable in the biplot. In Table 2.9, the coordinates or correlation of the first two PC's with each of the 14 variables is displayed. The coordinates show the positive correlation of YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager with PC1 and PC2.

In Table 2.10, the contribution of each of the variables on to the fist two principal components is demonstrated. As was expected, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager all contribute highly in the first principal component. As far as the second PC is concerned, Age and NumCompaniesWorked are overwhelmingly contributing followed by MonthlyIncome, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager.

Finally, in Table 2.11, the squared cosine of the first two PC's with each of the variables is demonstrated. The results show that PC1 is of big importance for the variables: YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager, Age, MonthlyIncome and TotalWorkingYears. In the other hand, PC2 is of significance for Age, MonthlyIncome, NumCompaniesWorked, TotalWorkingYears, YearsInCurrentRole and YearsWithCurrManager. The results for PC2 indicate that not all variables which include the word Years are important for PC2. This is opposite to the results of contribution index where all the variables starting with the word "Years" were contributing to PC2.

Table 2.9: The table of coordinates of the first two principal components for each of the 14 variables.

| | PC1 | PC2 |
|---|---|---|
| Age | 0.561488284 | -0.606551267 |
| DailyRate | -0.013658943 | -0.100150782 |
| DistanceFromHome | 0.009644222 | 0.053394513 |
| HourlyRate | -0.022624389 | -0.080503543 |
| MonthlyIncome | 0.722756489 | -0.373042720 |
| MonthlyRate | 0.002251302 | -0.110678851 |
| NumCompaniesWorked | 0.062113639 | -0.719548948 |
| PercentSalaryHike | -0.030767043 | 0.005932922 |
| TotalWorkingYears | 0.832310987 | -0.408652595 |
| TrainingTimesLastYear | -0.022032856 | 0.118771418 |
| YearsAtCompany | 0.888744670 | 0.273723195 |
| YearsInCurrentRole | 0.784345533 | 0.358948795 |
| YearsSinceLastPromotion | 0.690087183 | 0.255197149 |
| YearsWithCurrManager | 0.773960070 | 0.379136440 |

Table 2.10: The table of the contribution of each variable on to the first two principal components.

| | PC1 | PC2 |
|---|---|---|
| Age | 7.849 | 22.294465785 |
| DailyRate | 0.005 | 0.607814044 |
| DistanceFromHome | 0.002 | 0.173 |
| HourlyRate | 0.013 | 0.393 |
| MonthlyIncome | 13.005 | 8.433 |
| MonthlyRate | 0.000 | 0.742 |
| NumCompaniesWorked | 0.096 | 31.375 |
| PercentSalaryHike | 0.024 | 0.002 |
| TotalWorkingYears | 17.246 | 10.120 |
| TrainingTimesLastYear | 0.012 | 0.855 |
| YearsAtCompany | 19.664 | 4.540 |
| YearsInCurrentRole | 15.316 | 7.808 |
| YearsSinceLastPromotion | 11.856 | 3.947 |
| YearsWithCurrManager | 14.913 | 8.711 |

Table 2.11: The table of the squared cosine of the first two PC's which shows their importance to each of the variables.

| | PC1 | PC2 |
|---|---|---|
| Age | 0.315 | 0.368 |
| DailyRate | 0.000 | 0.010 |
| DistanceFromHome | 0.000 | 0.003 |
| HourlyRate | 0.001 | 0.006 |
| MonthlyIncome | 0.522 | 0.139 |
| MonthlyRate | 0.000 | 0.012 |
| NumCompaniesWorked | 0.004 | 0.518 |
| PercentSalaryHike | 0.001 | 0.000 |
| TotalWorkingYears | 0.693 | 0.167 |
| TrainingTimesLastYear | 0.000 | 0.014 |
| YearsAtCompany | 0.790 | 0.075 |
| YearsInCurrentRole | 0.615 | 0.129 |
| YearsSinceLastPromotion | 0.476 | 0.065 |
| YearsWithCurrManager | 0.599 | 0.144 |

The results above were created by running the following code:

```
# Eigenvalues
>eig.val <- get_eigenvalue(prcomp.pca)

# Results for Variables
>prcomp.res.var <- get_pca_var(prcomp.pca)

# Coordinates
>prcomp.coord<-prcomp.pca$coord

# Contributions to the PC's
>prcomp.contrib<-prcomp.pca$contrib
>round(prcomp.contrib[,1:2],digits=3)

# Quality of representation
>prcomp.cos2<-prcomp.pca$cos2
>round(prcomp.cos2[,1:2],digits=3)
```

### 2.10.2 Principal Component Analysis using princomp with cor=T

There are several commands to conduct PCA and obtain the results of the technique. One of these commands is princomp. princomp is a command which requires two additional settings as input. This is `scores=TRUE` and `cor=TRUE`. The first command is required to obtain the scores as well in the output, the second command is used to use the correlation matrix instead of the covariance matrix.

The code which conducts PCA is demonstrated below.

```
>princomp.pca <- princomp(pcanumdf, scores=TRUE, cor=TRUE)
```

The following command reveals the available outputs from the PCA which are also displayed below.

```
Call:
attributes(princomp.pca)

Output:
"sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"   "call"
```

"n.obs" displays the number of observations for which PCA is conducted, "scores" contains the matrix of the PCA scores for each observation, "call" simply displays the PCA command which was executed, "loadings" displays the weight or loadings matrix for each variable and "sdev" is the standard deviation and its squared value is, of course, the variance of each principal component as well as the proportion of the variance and the cumulative proportion are demonstrated in Table 2.12. Notice how again the two first principal components account for a decent amount of the total variance (0.2869124+0.1178718 = 0.4047842). Finally, "center" and "scale", as before, are set TRUE for this PCA demonstration for robust results.

Table 2.12: The table of the standard deviation, the variance, the proportion of the variance and the cumulative proportion of the first two PC's using princomp.

| | Standard deviation | Variance | Proportion of the variance | Cumulative proportion |
|---|---|---|---|---|
| PC1 | 2.0041891 | 4.0167738 | 0.2869124 | 0.2869124 |
| PC2 | 1.2846031 | 1.6502052 | 0.1178718 | 0.4047842 |
| PC3 | 1.03368477 | 1.0685042 | 0.07632173 | 0.48110594 |
| PC4 | 1.02572905 | 1.0521201 | 0.07515143 | 0.55625738 |
| PC5 | 1.00469438 | 1.0094108 | 0.07210077 | 0.62835815 |
| PC6 | 0.99622180 | 0.9924579 | 0.07088985 | 0.69924800 |
| PC7 | 0.97639628 | 0.9533497 | 0.06809641 | 0.76734440 |
| PC8 | 0.9578606 | 0.9174969 | 0.0655355 | 0.8328799 |
| PC9 | 0.84997966 | 0.7224654 | 0.05160467 | 0.88448457 |
| PC10 | 0.72848988 | 0.5306975 | 0.03790696 | 0.92239154 |
| PC11 | 0.68545272 | 0.4698454 | 0.03356039 | 0.95595193 |
| PC12 | 0.53218724 | 0.2832233 | 0.02023023 | 0.97618216 |
| PC13 | 0.43975445 | 0.1933840 | 0.01381314 | 0.98999530 |
| PC14 | 0.3742537 | 0.1400658 | 0.0100047 | 1.0000000 |

The output above indicates that again 2 PC's are considered relatively more relevant than the rest of the principal components. These results are confirmed by the scree plot displayed in Figure 2.6.

The code which produced the results above is the following (continues on the next page):

```
>princomp.pca <- princomp(pcanumdf, scores=TRUE, cor=T)
>summary(princomp.pca)

# The attributes of princomp PCA
>attributes(princomp.pca)

# The standard deviation of each of the principal components
>princomp.sdev <- princomp.pca$sdev
```

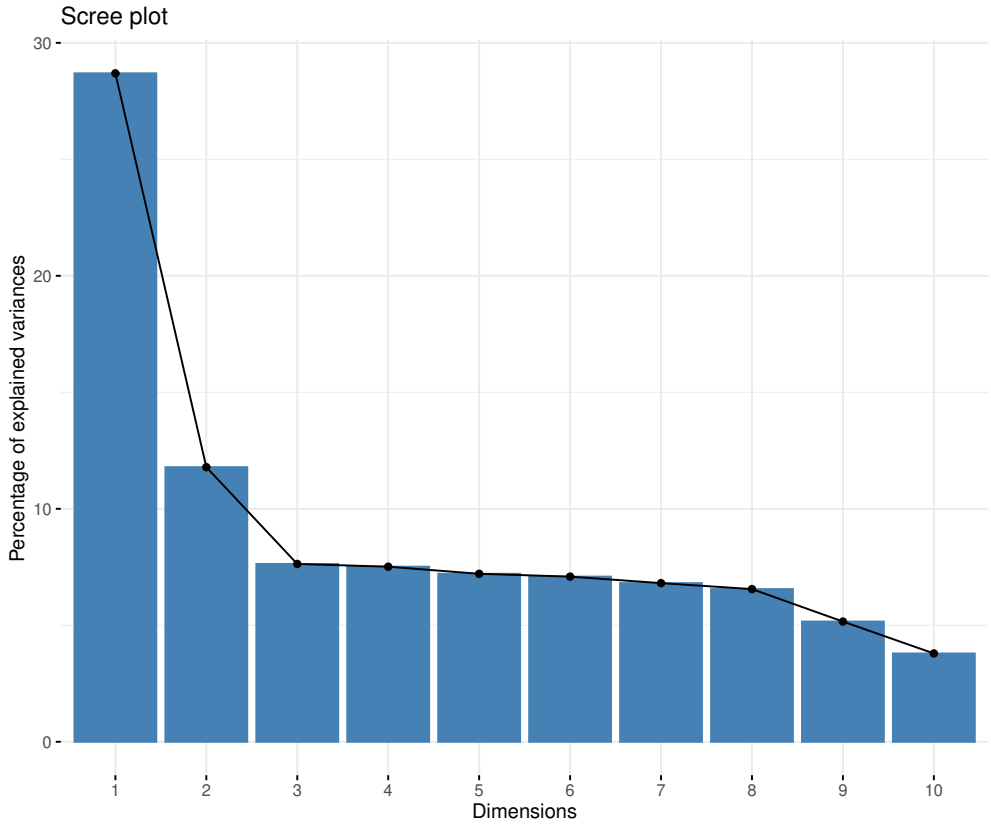



Figure 2.6: The scree plot or displayed eigenvalues in descending order using princomp.

```
# Variance of each of the principal components
>princomp.var <- princomp.sdev^2

# Proportion of variance of each of the principal components
>princomp.var.per <- round(princomp.var/sum(princomp.var)*100, digits = 2)

# The scree plot of the eigenvalues
>fviz_eig(princomp.pca)
```

The second attribute available as output by using the princomp command is the loadings which, as mentioned in the previous section, is nothing but the weight matrix w of the PCA. In practice, regarding PC1 for example, each data value of each variables of an observation is multiplied by these weights and their summation results in the score of the specific observation. Notice how the weights behave simiraly as when prcomp command was used to conduct PCA. This is mainly that YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager variables all have positive weights, quantitatively close to each and towards the same direction.

The biplot of the specific PCA using the princomp command lands similar results with the ones when using the prcomp command. This is that as PCA value raise for an observation generally YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager are increasing with descending rate. At the same time Age, TotalWorkingYears and MonthlyIncome are increasing rapidly which is normal since the weights pointed out so. The results are different as far as PC2 goes. In comparison with the results obtained from prcomp for PC2, as an observation increases in value, the variables YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager gen-

Table 2.13: The table of loadings of the numerical dataset regarding the first two PC's using princomp which show the direction of the eigenvectors and the direction of the variables.

| | PC1 | PC2 |
|---|---|---|
| Age | 0.28 | 0.47 |
| DailyRate | -0.01 | 0.08 |
| DistanceFromHome | 0.00 | -0.04 |
| HourlyRate | -0.01 | 0.06 |
| MonthlyIncome | 0.36 | 0.29 |
| MonthlyRate | 0.00 | 0.09 |
| NumCompaniesWorked | 0.03 | 0.56 |
| PercentSalaryHike | -0.02 | 0.00 |
| TotalWorkingYears | 0.42 | 0.32 |
| TrainingTimesLastYear | -0.01 | -0.09 |
| YearsAtCompany | 0.44 | -0.21 |
| YearsInCurrentRole | 0.39 | -0.28 |
| YearsSinceLastPromotion | 0.34 | -0.20 |
| YearsWithCurrManager | 0.39 | -0.30 |

erally decrease while Age, TotalWorkingYears, MonthlyIncome and NumCompaniesWorked increase. The biplot which lines up with these results in demonstrated in Figure 2.7.

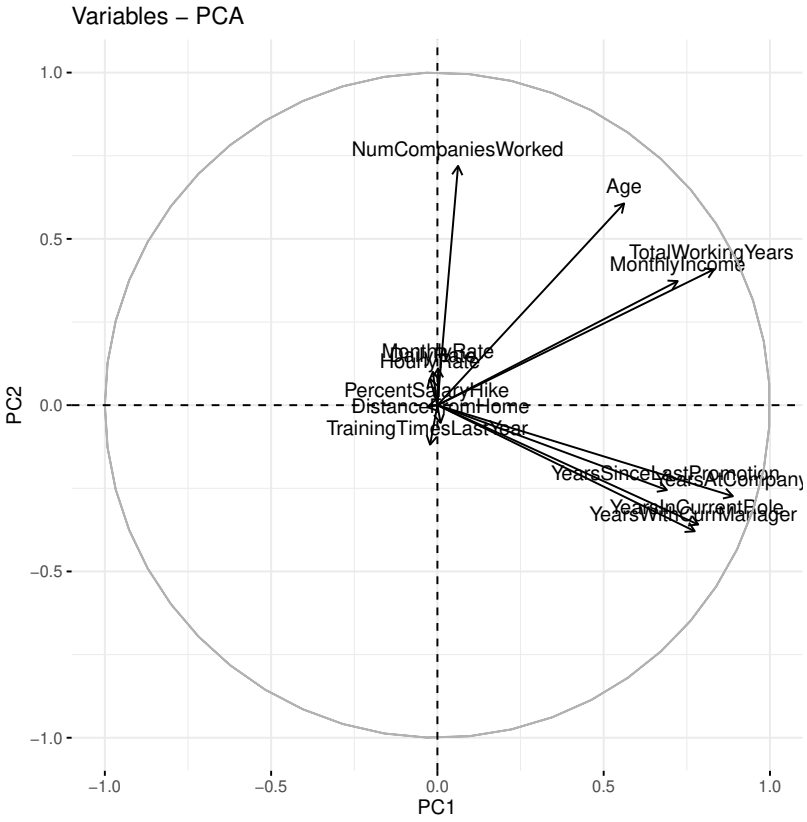



Figure 2.7: The biplot or displayed eigenvectors for each variable of PC1 and PC2 using princomp.

As already mentioned, the eigenvalue of a principal component is associated with the amount of information out of the total explained by the specific PC. The larger the eigen-

value, the larger the contribution of the principal component to the total variance. In the example, the eigenvalues are demonstrated in Table 2.14. It is obvious that the first two PC's are accounting for most of the variance since its values are relatively larger than the rest of the eigenvalues.

Table 2.14: The table of eigenvalues of each principal component.

| | Eigenvalue |
|---|---|
| PC1 | 4.0167738 |
| PC2 | 1.6502052 |
| PC3 | 1.0685042 |
| PC4 | 1.0521201 |
| PC5 | 1.0094108 |
| PC6 | 0.9924579 |
| PC7 | 0.9533497 |
| PC8 | 0.9174969 |
| PC9 | 0.7224654 |
| PC10 | 0.5306975 |
| PC11 | 0.4698454 |
| PC12 | 0.2832233 |
| PC13 | 0.1933840 |
| PC14 | 0.1400658 |
| Sum | 14.000000 |

The correlation between the variables and the principal components, or also called coordinates is another important PCA measure. Recall, that coordinates of PC1 and PC2 are the numbers used to plot the eigenvector of each variable in the biplot. In Table 2.15, the coordinates or correlation of the first two PC's with each of the 14 variables is displayed. The coordinates show the positive correlation of YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager with PC1 and PC2.

Table 2.15: The table of coordinates of the first two principal components for each of the 14 variables.

| | PC1 | PC2 |
|---|---|---|
| Age | 0.561488284 | -0.606551267 |
| DailyRate | -0.013658943 | -0.100150782 |
| DistanceFromHome | 0.009644222 | 0.053394513 |
| HourlyRate | -0.022624389 | -0.080503543 |
| MonthlyIncome | 0.722756489 | -0.373042720 |
| MonthlyRate | 0.002251302 | -0.110678851 |
| NumCompaniesWorked | 0.062113639 | -0.719548948 |
| PercentSalaryHike | -0.030767043 | 0.005932922 |
| TotalWorkingYears | 0.832310987 | -0.408652595 |
| TrainingTimesLastYear | -0.022032856 | 0.118771418 |
| YearsAtCompany | 0.888744670 | 0.273723195 |
| YearsInCurrentRole | 0.784345533 | 0.358948795 |
| YearsSinceLastPromotion | 0.690087183 | 0.255197149 |
| YearsWithCurrManager | 0.773960070 | 0.379136440 |

In Table 2.16, the contribution of each of the variables on to the fist two principal components is demonstrated. As was expected, YearsAtCompany, YearsInCurrentRole, YearsS-

inceLastPromotion and YearsWithCurrManager all contribute highly in the first principal component. As far as the second PC is concerned, Age and NumCompaniesWorked are overwhelmingly contributing followed by MonthlyIncome, TotalWorkingYears, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager.

Table 2.16: The table of the contribution of each variable on to the first two principal components.

| | PC1 | PC2 |
|---|---|---|
| Age | 7.85 | 22.29 |
| DailyRate | 0.00 | 0.61 |
| DistanceFromHome | 0.00 | 0.17 |
| HourlyRate | 0.01 | 0.39 |
| MonthlyIncome | 13.00 | 8.43 |
| MonthlyRate | 0.00 | 0.74 |
| NumCompaniesWorked | 0.10 | 31.37 |
| PercentSalaryHike | 0.02 | 0.00 |
| TotalWorkingYears | 17.25 | 10.12 |
| TrainingTimesLastYear | 0.01 | 0.85 |
| YearsAtCompany | 19.66 | 4.54 |
| YearsInCurrentRole | 15.32 | 7.81 |
| YearsSinceLastPromotion | 11.86 | 3.95 |
| YearsWithCurrManager | 14.91 | 8.71 |

In Table 2.17, the squared cosine of the first two PC's with each of the variables is demonstrated. The results show that PC1 is of big importance for the variables: YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager, Age, MonthlyIncome and TotalWorkingYears. In the other hand, PC2 is of significance for Age, MonthlyIncome, NumCompaniesWorked, TotalWorkingYears, YearsInCurrentRole and YearsWithCurrManager. The results for PC2 indicate that not all variables which include the word Years are important for PC2. This is opposite to the results of contribution index where all the variables starting with the word "Years" were contributing to PC2.

Table 2.17: The table of the squared cosine of the first two PC's which shows their importance to each of the variables.

| | PC1 | PC2 |
|---|---|---|
| Age | 0.32 | 0.37 |
| DailyRate | 0.00 | 0.01 |
| DistanceFromHome | 0.00 | 0.00 |
| HourlyRate | 0.00 | 0.01 |
| MonthlyIncome | 0.52 | 0.14 |
| MonthlyRate | 0.00 | 0.01 |
| NumCompaniesWorked | 0.00 | 0.52 |
| PercentSalaryHike | 0.00 | 0.00 |
| TotalWorkingYears | 0.69 | 0.17 |
| TrainingTimesLastYear | 0.00 | 0.01 |
| YearsAtCompany | 0.79 | 0.07 |
| YearsInCurrentRole | 0.62 | 0.13 |
| YearsSinceLastPromotion | 0.48 | 0.07 |
| YearsWithCurrManager | 0.60 | 0.14 |

According to Figure 2.8, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager are important variables to PC1.

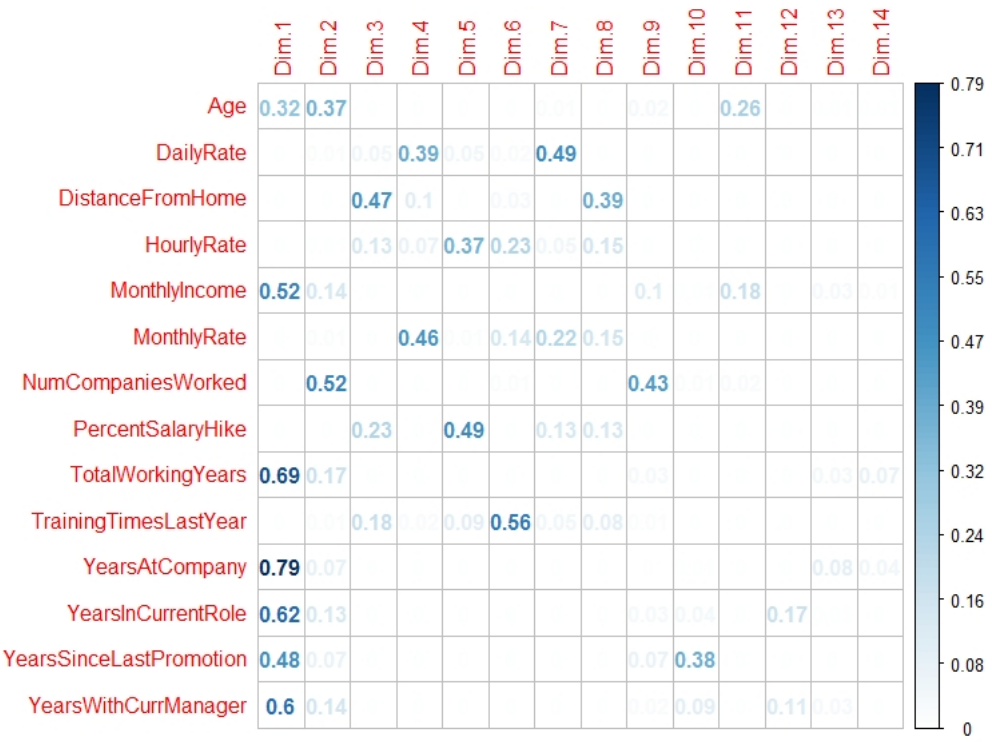



Figure 2.8: The correlation between the variables and the principal components through the squared cosine index.

According to Figure 2.9, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager contribute with the highest variance in PC1.

The results above were produced with the following lines of code:

```
#Results for Variables
>princomp.res.var<-get_pca_var(princomp.pca)

#Coordinates of the variables
>princomp.coord<-princomp.res.var$coord

#Contribution of variables to the PCs
>princomp.contrib<-princomp.res.var$contrib
round(princomp.contrib,digits=2)
>corrplot(princomp.contrib, is.corr=FALSE,method="number")

#Quality of representation
>princomp.cos<-princomp.res.var$cos2
round(princomp.cos,digits=2)
>corrplot(princomp.cos, is.corr=FALSE)
```

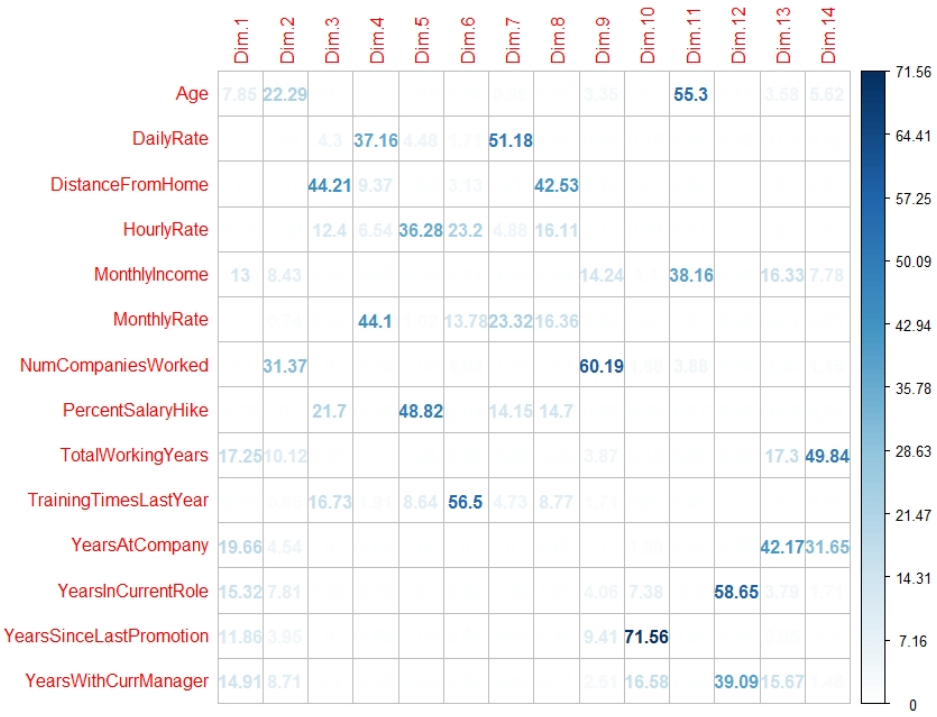



Figure 2.9: The correlation between the variables and the principal components through the contribution index.

### 2.10.3 Principal Component Analysis with cor=F

Most scientific community prefers the covariance table over the correlation table to conduct PCA. Therefore, for good measure, in this section the results of the PCA with covariance table will be demonstrated. The R command which is going to be used to perform PCA is princomp. The command which is used to perform PCA with the princomp is demonstrated below. Notice how the input `cor=F` indicates that the covariance matrix will be used instead of the correlation matrix. The command `summary(princomp.pca)` displays the standard deviation, the proportion of variance and the cumulative proportion.

Table 2.18: The table which consist of standard deviation, proportion of variance and cumulative proportion with the covariance matrix as input.

| | Standard deviation | Proportion of the variance | Cumulative proportion |
|---|---|---|---|
| PC1 | 7118.72 | 0.6947505 | 0.6947505 |
| PC2 | 4701.29 | 0.3030125 | 0.9977630 |
| PC3 | 403.14 | 0.00222 | 0.9999912 |
| PC4 | 20.31 | ≈0 | ≈100 |
| PC5 | 8.66 | ≈0 | ≈100 |
| PC6 | 8.09 | ≈0 | ≈100 |
| PC7 | 6.81 | ≈0 | ≈100 |
| PC8 | 3.65 | ≈0 | ≈100 |
| PC9 | 3.43 | ≈0 | ≈100 |
| PC10 | 2.44 | ≈0 | ≈100 |
| PC11 | 2.27 | ≈0 | ≈100 |
| PC12 | 2.01 | ≈0 | ≈100 |
| PC13 | 1.89 | ≈0 | ≈100 |
| PC14 | 1.28 | ≈0 | ≈100 |

The results are demonstrated in Table 2.18. According to Table 2.18, the first two principal components are responsible for almost 100% of the total variation. The scree plot in Figure 2.10 confirms the result. This is that the two first principal components account for more than 99% of the total variance. This outcome is remarkably good for a SEM analysis because usually these are the expected results.

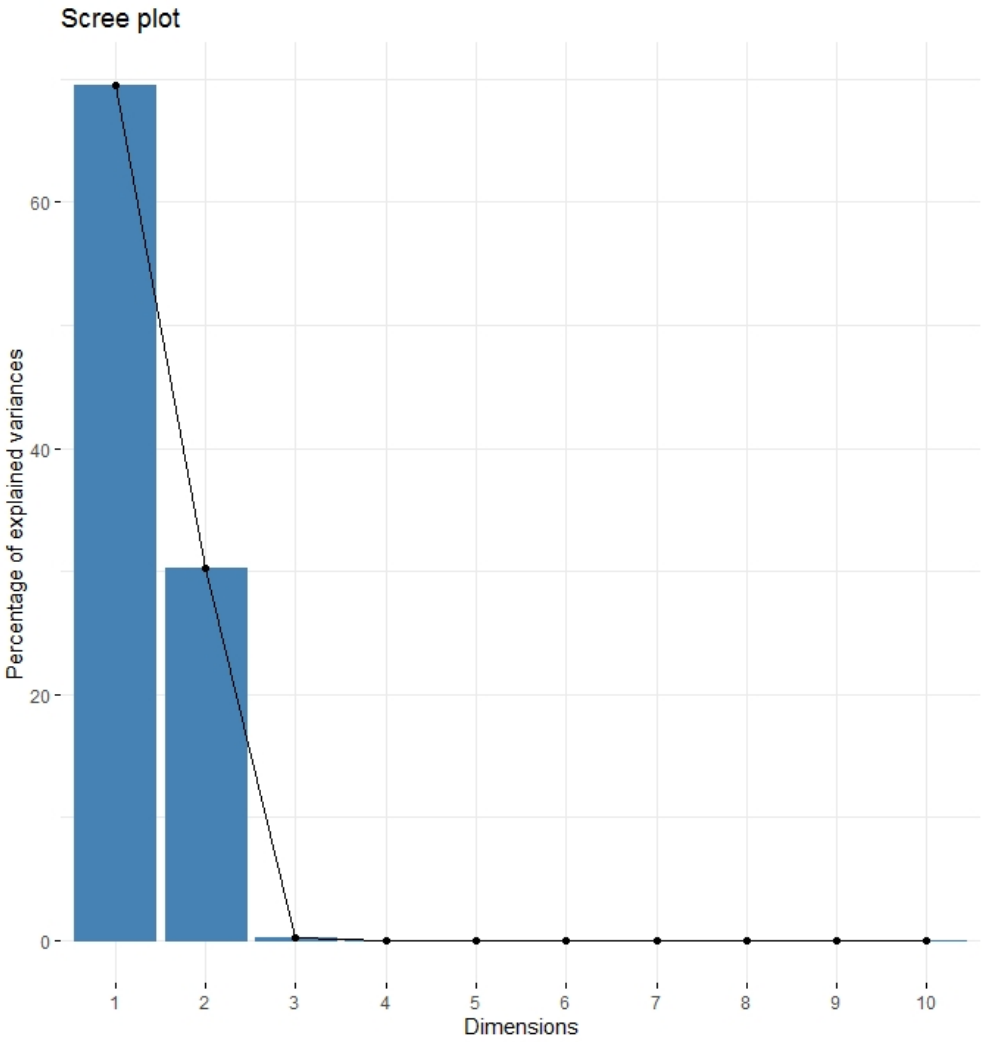



Figure 2.10: Scree plot of the PCA with covariance table as input.

## 2.11 Endogeneity and Exogeneity

The most important categorization in SEM is between exogenous and endogenous variables. The reasons why variables are exogenous is not presented in the structural models. In fact, the causes of an exogenous variable cannot be measured or are unknown (out of the model's reach). It is interesting that exogenous variables are allowed to vary and covary. The path diagram of a SEM analysis usually symbolizes the covariance between two exogenous variables with a curved line with two arrowheads that point in each one of the exogenous variables. On the other side, the causes which result in a variable being endogenous are explicitly analyzed and presented in the SEM model. This means that endogenous variables are not allowed to vary or covary.

In path diagrams endogenous variables can have direct effects pointing at them from both exogenous and endogenous variables. When one or more endogenous variables is specified

as a direct cause of another endogenous variables then it said that there is an indirect effect taking place. For example, if $Y_1$ and $Y_2$ are endogenous and $X$ is exogenous, then if the path diagram is specified in the following order: $X \rightarrow Y_1 \rightarrow Y_2$, then an indirect effect occurs. In this indirect effect, $Y_1$ is partially contributor of the effect of $X$ on $Y_2$. $Y_1$ is this case is also known as mediator. The indirect can be measured as the product of the path coefficients of all the direct effects which build the indirect effect as it will be demonstrated later on. Conceptually, the difference between the two categories of variables are not based just in theory. If a researcher states or thinks that a variable is exogenous, his statement doesn't hold in absolute fashion. There are several statistical disadvantages when exogenous variables is claimed to not caused by endogenous variables. The consequences are connected with the assumption of exogeneity. By definition, exogeneity takes place when the parameters of the conditional distribution of the endogenous variables given the exogenous variables are unrelated to those that describe the distributions of the exogenous variables by themselves.

All kind of disturbances of the endogenous variables in structural model are not related to the exogenous variables when exogeneity takes place. In simple words, the hidden causes of the endogenous variables are uncorrelated with all the exogenous variables. This assumption is known as pseudo-isolation and allows for the estimation of direct effects and disturbances, with constant omitted causes. In a SEM framework, exogeneity requirements include valid specification of the directionality and that there are no hidden causes which are related to the exogenous variables. These requirements imply that in order for a researcher to conduct SEM, he must have strong statistical fundamentals to specify the initial model correctly.

Assuming the following specification: $X \rightarrow Y$, endogeneity occurs when a candidate exogenous variable $X$ covaries with the $Y$ error term. This reveals two facts: a) exogeneity cannot hold and b) $X$, in reality, is not exogenous. If the direction of the causal relation between $X$ and $Y$ is misspecified, this can result in endogeneity. In case $Y$ indeed causes $X$ ($Y \rightarrow X$), then $X$ is definitely not exogenous. An bi-causal relationship between $X$ and $Y$ ($X \rightarrow Y$ and $Y \rightarrow X$) leads to similar results. An illustrative example will be demonstrated to prove the above facts. Assume that a country increases the number of police officers in order to reduce the crime rate. Additionally, the increase in crime actually kind of "forces" the country to hire more police, which means that the latter is not exogenous since the two variables affect each other in both directions.

Often researchers control for a variable randomly without having constructed a theory about the kind of relationships between the variables. Hence, variables most of the times are controlled without any justification or so ever. The most common justification for controlling a variable is that its effect has already occurred in a specific point in time in the past. The theoretical framework should be the the "compass" with which the researcher will make such decisions. Assume that a dataset has three variables where: $B$= Student background characteristics $S$ = school quality and $A$ = Academic achievement. The variables are assumed to cause one another based on theory and the variance is partitioning in four different ways corresponding to the four graphs of Figure 2.11.

As a reminder, a variable is called exogenous when its variability is assumed to be determined by causes outside the causal model under study. In simple words, it doesn't make sense to try to explain the variability or the relationships of an exogenous variable with other unique exogenous variables. In contrast, a variable is called endogenous when it can be explained by both endogenous and/or exogenous variables. In the first graph of Figure 2.11, $B$ and $S$ are assumed to be exogenous. This is obvious since they can covary as well. In the same graph, variable $A$ is considered endogenous since it is affected by two exogenous variables, $B$ and $S$. In the second graph of Figure 2.11, $B$ is exogenous since it doesn't get affected by any variable, but most likely by hidden causes. In the same graph, $S$ and $A$ are the endogenous variables since they both get affected by some variable. In the third graph,

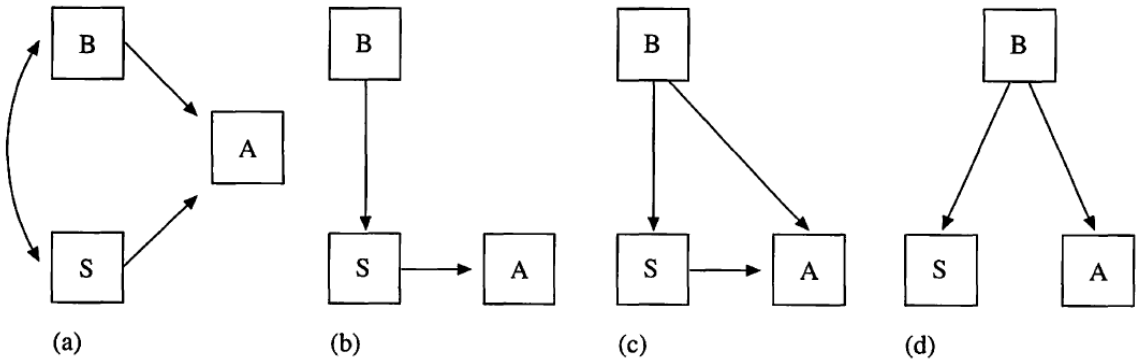



Figure 2.11: Four different path diagrams and variance partitioning for a dataset consisting of students.

$B$ is exogenous, while $S$ and $A$ are endogenous. Finally, in the fourth graph $B$ is considered exogenous and $S$ and $A$ are considered endogenous.

## 2.12 Path Analysis

The path diagram is of complementary nature for the analysis itself. However, it is good to demonstrate visually the hypothesized relationships between the different variables. In the previous section, a separation was made between exogenous and endogenous variables. In Figure 2.12, the distinction between the two kind of variables is clear. In the specific graph, variables 1 and 2 are exogenous and their correlation ($r_{12}$) is depicted with the double headed curved arrow. This is an indication that the specific researcher doesn't consider that one variable is causing the other by any means. Therefore, $r_{12}$ which represents the connection between variable 1 and 2 is unknown. In contrast, variables 3 and 4 of Figure 2.12 are endogenous because they are caused by other variables. All the variables are connected through lines with arrowheads. Variables in which an arrowhead ends up are considered to be caused (dependent) from the variables (independent) in which the corresponding line of the arrowhead begins. For example, in Figure 2.12, variable 3 (dependent) is considered to be caused by variables 1 and 2 (independents). It is worth noticing that the models analyzed in this section are recursive models. A model is called recursive when the causal flow is unidirectional. The reciprocal causation between variables is not allowed. For instance, in Figure 2.12, the variable 3 is caused partially from variable 2, but the opposite (that a variable 2 is caused by variable 3) is not possible. In such models, the endogenous variable is considered to be the dependent variable with a set of variables being the independent variables, always with respect to the other endogenous variables. For example, variables 1 and 2 are predictors on variable 3, the outcome, with respect to variable 4. It is worth noticing that, in this case, the causal flow is considered to be unidirectional. This is because in the science of statistics in general, the overall variance of a specific variable cannot be accounted and residuals reflect the effects of variables which are excluded from the model.

According to Figure 2.12, the residuals are $a$ and $b$, just like the residual term $e$ of the regression model.

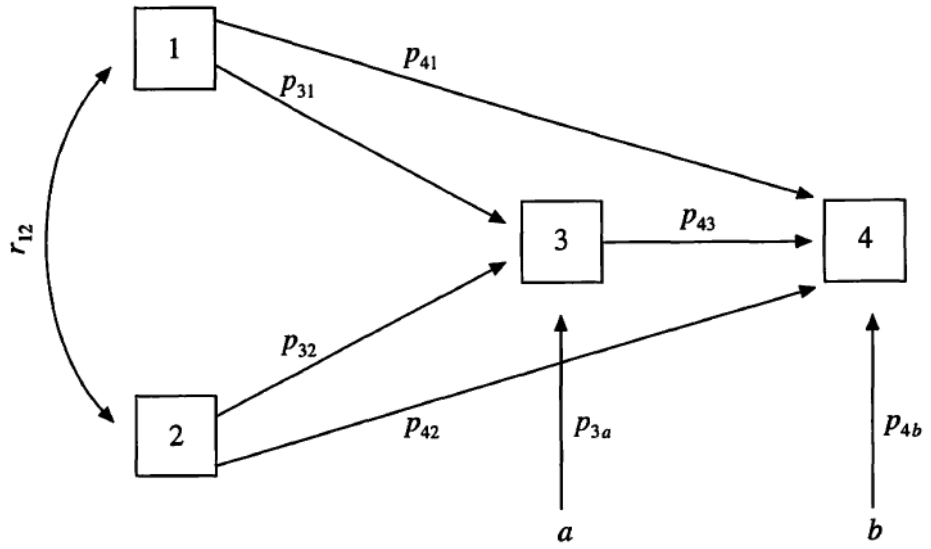



Figure 2.12: A path diagram with 4 variables, 7 path coefficients, 2 residuals and 1 correlation relationship.

### 2.12.1 Assumptions

The following assumption are necessary for implementing path analysis:

- The relationships between the variables are linear, additive, and causal. Therefore multiplicative, interaction and curvilinear relations are excluded.
- The correlation between a specific residuals and preceding variables in the model is not feasible. Visually, in Figure 2.12, it is true that a) the correlation between variables 1,2 and $a$, b) the correlation between variables 1,2,3 and $b$ is zero c) the correlation between the different residuals is zero (for Figure 2.12, that the correlation between $a$ and $b$ is zero).

  The facts above make clear that all relevant variables are in the model which is tested and, of course, that hidden or excluded variables are not correlated with the actual variables. The linear combination of exogenous and endogenous variables along with a residuals consist of an endogenous variable in the model. This is true for every endogenous variable.
- The system has one-way causal flow, reciprocal causation is forbidden.
- The scale of all variables is interval.
- The measurement of the variables is done without error.

### 2.12.2 Path Coefficients

Wright (1934) gave the definition of a path coefficient as follows: "A path coefficient is the fraction of the standard deviation of the dependent variable (with the appropriate sign) for which the designated factor is directly responsible, in the sense of the fraction which would be found if this factor varies to the same extent as in the observed data while all others are constant."

Briefly, a path coefficient is an indication of a variable's direct effect hypothesized as a variable's cause taken as an effect. In the notation of Figure 2.12, a path coefficient is

denoted with $p$ with two indices. The first index shows the effect and the second the cause. For instance, $p_{32}$ represents the direct effect of variable 2 on variable 3.

An equation which consists of the variables on which it is hypothesized to be dependent and an error term represents each of the endogenous variables. A path coefficients correspond to each of the endogenous variables which shows the magnitude of expected change in the dependent variable when the independent variable changes by a unit. The residual term represents the exogenous variables since there are other hidden variables which are considered dependent for such variables. The residuals are denoted with $e$ in Figure 2.13.

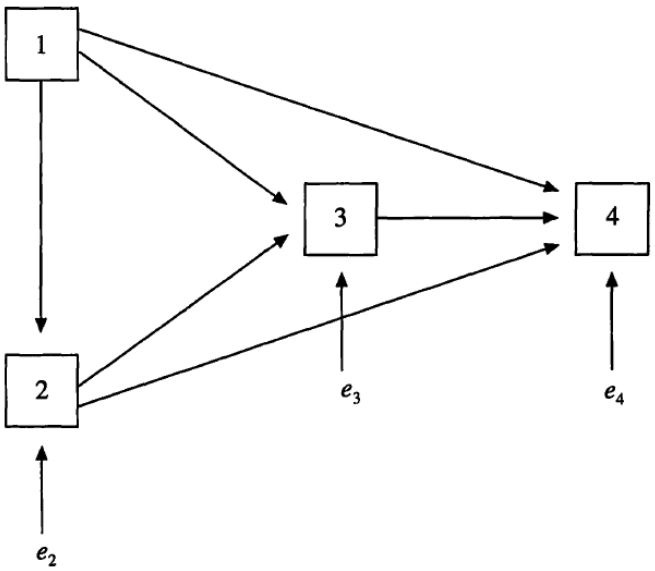



Figure 2.13: A path diagram with 4 variables and 3 residuals.

The equations which are displayed graphically in Figure 2.13 are demonstrated in Equation 2.49, where $e's$ are expressed in standard scores and are the hidden variables of the model (Pedhazur, Kerlinger, et al. (1982)).

$$z_1 = e_1 \tag{2.49}$$
$$z_2 = p_{21}z_1 + e_2 \tag{2.50}$$
$$z_3 = p_{31}z_1 + p_{32}z_2 + e_3 \tag{2.51}$$
$$z_4 = p_{41}z_1 + p_{42}z_2 + p_{43}z_3 + e_4 \tag{2.52}$$
$$\tag{2.53}$$

As it is demonstrated in Figure 2.13, variable 1 is exogenous and, therefore, is represented by $e_1$ ($e_1$ represents the variables out of the model which are affecting itself). Variable 1 and $e_2$ ($e_2$ represents the variables out of the model which affect variable 2). The rest of the equation are interpreted in a similar fashion. This set of equation is known as recursive system. In such systems a minimum of 50% of the path coefficients are zero. Therefore, the recursive system can be represented with matrix with lower triangular form, since the upper consists of values which are equal to zero. Mathematically, this is implies that $z_1 = e_1 + 0_{12}z_2 + 0_{13}z_3 + 0_{14}z_4$ for the first equation in Equation 2.49. The rest of the equation of Equation 2.49 are treated similarly.

Now, the model's path coefficients of Figure 2.13 will be calculated. In order to achieve such calculation the procedure is the following: Starting of with $p_{21}$ which represents the variable's 1 effect on variable 2. As reminder, from the previous section it is true that:

$$r_{12} = \frac{1}{N} \sum z_1 z_2 \tag{2.54}$$

The substitution of the second equation of Equation 2.49 for $z_2$ results in:

$$r_{12} = \frac{1}{N} \Sigma z_1 (p_{21} z_1 + e_2) = p_{21} \frac{\sum z_1 z_1}{N} + \frac{\sum z_1 e_2}{N} \tag{2.55}$$

The second element of the first product in Equation 2.55, $(\sum z_1 z_1)/N$ is equal to one since $(\sum z_1 z_1)/N = \sum {z_1}^2/N = 1$ or the standard scores' variance is equal to one. Additionally, it is assumed that variable 1 and $e_2$ have covariance equal to 0. With all that being said it is true that:

$$r_{12} = p_{21} \tag{2.56}$$

$\beta$ is the correlation coefficient in a simple regression framework. Accordingly, $r_{12} = \beta_{21} = p_{21}$. In simple words, the data helps to estimate $\beta_{21}$ (the path coefficient from variable 1 to variable 2) through estimating $r_{12}$.

A path coefficient has zero-order correlation if a variable is considered to be dependent on only one cause and residual. In case a variable is considered to be dependent on a larger number of independent causes the same is applied. In Figure 2.14, $X$ and $Z$ are considered to be independent causes of $Y$. Thus, is true that $p_{yx} = r_{yx}$ and $p_{yz} = r_{yz}$ (Pedhazur, Kerlinger, et al. (1982)).

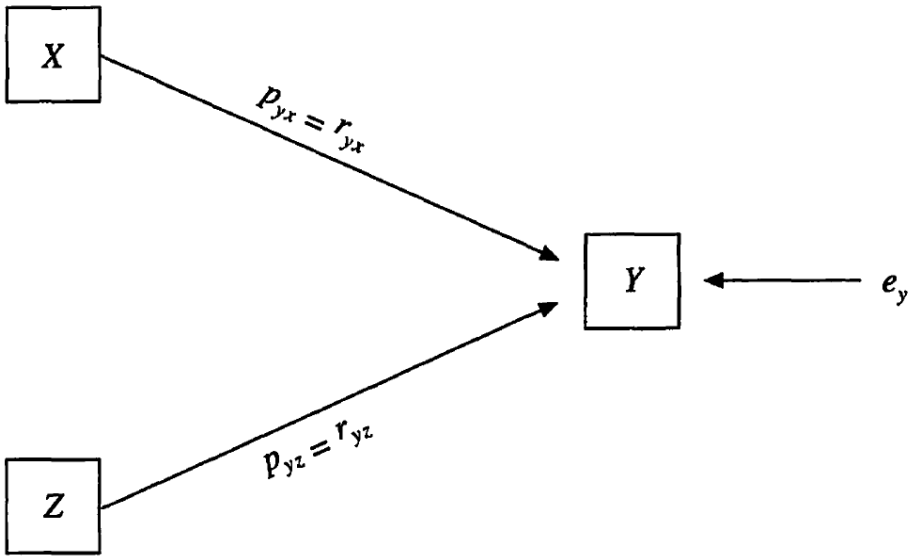



Figure 2.14: A graph where it shows that variables $X$ and $Z$ are independent causes of variable $Y$.

Back to the main discussion, in Figure 2.13, as already mentioned, behind $e_2$ represents the hidden or excluded variables in the system. Therefore, the direct calculation of the path coefficient from $e_2$ to variable 2 is not possible. The assumptions of zero correlation between $e_2$ and variable 1 alongside with the fact the all the variables are expressed with standard scores gives solution to the problem. Previously it is said that the path coefficient of each cause (when they are independent of one another) is equal to the zero-order correlation with the endogenous variable. The variable's 2 causes, independent of variable 1, are now represented by $e_2$. It is proven then, that the correlation and the path coefficient between $e_2$ and variable 2 the same exact number. Remember that the correlation between variable 2

and $e_2$ is equal to $\sqrt{1-r_{12}^2}$ (the path coefficient from $e_2$ to variable 2). It is worth noticing that a) variable's 2 variance accounted for by variable 1 is $r_{12}^2$ and b) that accounted for by $e_2$ is $1-r_{12}^2$ (the path coefficients' squares for variable 1 and $e_2$, respectively).

With the same logic, it can be proved that the path coefficient obtained from hidden variables to an endogenous variable,j , in a recursive system is equal to $\sqrt{1-R_{j.12...i}^2}$, where the endogenous variable's $j$ squared value of the multiple correlation is equal to $R_{j.12...i}^2$ and $1, 2..., i$ are the variables that affect $j$. For variable 2 of Figure 2.13, the above equation is reduced down to $R_{2.1}^2 = \sqrt{r_{21}^2}$ (Pedhazur, Kerlinger, et al. (1982)).

Variable 1 and variable which are not independent of each other are affecting variable 3 of Figure 2.13. Both variable 2 is dependent on variable 1 and $e_2$. Next up, below the calculation of the two paths $p_{31}$ and $p_{32}$ takes place. Due to the assumptions regarding $e's$, it is feasible to remove these terms from all the equations that follow, thereby simplifying the presentation.

$$r_{13} = \frac{1}{N}\sum z_1 z_3 \tag{2.57}$$

The substitution of the third equation of Equation 2.49 for $z_3$ results in:

$$r_{13} = \frac{1}{N}\Sigma z_1(p_{31}z_1 + p_{32}z_2) = p_{31}\frac{\sum {z_1}^2}{N} + p_{32}\frac{\sum z_1 z_2}{N} \rightarrow r_{13} = p_{32}r_{12} \tag{2.58}$$

The outcome of Equation 2.58 has two unknown path coefficients, $p_{31}$ and $p_{32}$, and the solution is not feasible. To somewhere approach the solution the construction of another equation with the same unknowns is necessary. The second equation is demonstrated in Equation 2.59.

$$r_{23} = \frac{1}{N}\sum z_2 z_3 \tag{2.59}$$

The substitution of the third equation of Equation 2.49 for $z_3$ results in:

$$r_{23} = \frac{1}{N}\Sigma z_2(p_{31}z_1 + p_{32}z_2) = p_{31}\frac{\sum z_2 z_1}{N} + p_{32}\frac{\sum {z_2}^2}{N} \rightarrow r_{23} = p_{31}r_{12} + p_{32} \tag{2.60}$$

Now there are two equation which have as their building blocks the path coefficients of the path towards variable 3. These equations are demonstrated in Equation 2.61.

$$r_{13} = p_{31} + p_{32}r_{12} \tag{2.61}$$
$$r_{23} = p_{31}r_{12} + p_{32} \tag{2.62}$$
$$\tag{2.63}$$

The two equations can be written in under the form of $\beta's$ solutions as following:

$$\beta_{31.2} + \beta_{32.1}r_{12} = r_{13} \tag{2.64}$$
$$\beta_{31.2}r_{12} + \beta_{32.1} = r_{23} \tag{2.65}$$
$$\tag{2.66}$$

Other than the different notation, Equation 2.61 and Equation 2.64 are the same. Therefore, the two solutions for path coefficients and for $\beta's$ is the same. As examined in previous section, the solution is given by the application of least squares on the regression of variable 3 on variables 1 and 2. The $\beta$ and the path coefficient is the same for a specific variable. In other words, $p_{31} = \beta_{31.2}$ and $p_{32} = \beta_{32.1}$, where $p_{31} \neq p_{13}$. Recall that it is not feasible to have both $p_{31}$ and $p_{13}$ because the models examined are recursive. This principle is applied to all path coefficients of such models. The causal model, as specified by the researcher, determines the calculation of the path coefficients. In Figure 2.13, variable 3 is affected by variable 1 and, therefore, $p_{31}$ is calculated (Pedhazur, Kerlinger, et al. (1982)).

Similarly the path coefficient from $e_3$ to variable 3 is equal to $\sqrt{1 - R^2_{3.12}}$. As far as variable 4 of Figure 2.13 is concerned, there are three path coefficients that need to be computed. These represent how variable 4 is affected by variables 1,2 and 3, respectively. In this framework, three equation are derived. The $1^{st}$ equation is the following:

$$r_{14} = \frac{1}{N} \sum z_1 z_4 \tag{2.67}$$

The substitution of the fourth equation of Equation 2.49 for $z_4$ results in:

$$r_{14} = \frac{1}{N} \Sigma z_1 (p_{41} z_1 + p_{42} z_2 + p_{43} z_3) = p_{41} \frac{\sum {z_1}^2}{N} + p_{42} \frac{\sum z_1 z_2}{N} + p_{43} \frac{\sum z_1 z_3}{N} \tag{2.68}$$

$$\rightarrow r_{14} = p_{41} + p_{42} r_{12} + p_{43} r_{13} \tag{2.69}$$

The two equations which are created are:

$$r_{24} = p_{41} r_{12} + p_{42} + p_{43} r_{23} \tag{2.70}$$

$$r_{34} = p_{41} r_{13} + p_{42} r_{23} + p_{43} \tag{2.71}$$

$$\tag{2.72}$$

In conclusion, when the assumptions referred in the previous section are fulfilled and the causal model consists of variables which are expressed in standard scores (z), the path coefficients are nothing more but the standard regression coefficients, $\beta's$ from the multiple regression framework. However, these two approaches should not be considered the exact same thing. This is because the outcome is regressed on all the predictors under study in a multiple regression framework. Contrary, the number of regression analysis is more than one in path analysis. In each step, the variables which are considered to affect an endogenous variable are regressing it. The path coefficients for the paths leading the particular set of independent variables to the dependent variable under consideration are the $\beta's$ computed. In Figure 2.13, three regression analyses are required to compute the path coefficients. The regression of variable 2 on variable 1 is used to compute the corresponding path ($p_{21}$) as shown in Equation 2.56. The regression of variable 3 on variables 1 and 2 is used to computed $p_{31}$ and $p_{32}$ as shown in Equation 2.58. The regression of variable 4 on variables 1,2 and 3 is used to compute $p_{41}$, $p_{42}$ and $p_{43}$ as shown in Equation 2.68. $\sqrt{1 - R^2_{4.123}}$ is the path coefficient from $e_4$ to variable 4 (Pedhazur, Kerlinger, et al. (1982)).

## 2.13 Observed and Latent Variables

### 2.13.1 Observed Variables

Observed variables are directly observed and measured from our data using tests, surveys, scales, and so on. In simple words, it is the data for which scores have been collected and assembled in a dataset. Researchers have instant access on observed variables. All observed variables in SEM can be ordered categorical or continuous. In SEM, observed variables can be grouped to define or infer the latent variable or construct as it will be defined later. Observed variables which are used to measure a latent construct indirectly are called indicators.

For example, the Wechsler-Intelligence Scale for Children-Revised (WISC-R) is an instrument which produces scores used to formalize the construct of the intelligence of children (latent variable). These scores could be from different measures of children such as the thought process speed, level of empathy and many more. The Dow Jones index is a very popular measure of the American corporate economy construct. Other indicator variables of this latent could be retail sales, export sales and gross national product. A latent variable which is highly related to health such as fitness could have blood pressure, exercise and diet as indicators. Researchers use multiple indicator variables to help them define a latent variable.

### 2.13.2 Latent Variables

Latent variables are not directly observed or measured but rather inferred from a set of indicators. These particular observed variables share a unique source of commonality in a certain degree which is a strong indication of a potential underlying latent variable. The variance of an indicator within latent variable models is a result of three components. Firstly, the degree in which the indicators is attributed to the factors. Secondly, a unique reliable factor which is paired to the particular indicator. Thirdly, the random error or unreliability. The distinction between the variance attributable to the factors, the unique factor of the indicators, and the random error allows the researcher to form the theoretical construct of interest. SEM specializes in constructing and estimating such models.

Researchers do not have instant access on latent variables. The name latent comes from the Latin word "lateo" meaning "lie hidden". Generally SEM as a family of techniques is specialized in latent variables which are continuous. To analyze latent variables which are categorical other statistical methods are used. Since latent variables can't be measured we synthesize a construct including all observed variables that are considered imperfect measures of the latent under study. Each relation of an observed variable with the latent is estimated and then, they are combined into a mathematical model which provides an underlying measure of the latent variable, a construct. Latent variables, or also called factors, are formed by the statistical scores obtained by the indicators.

In SEM latent variables correspond to constructs which are hypothetical. Practically, latent variable models display the relations between theoretical constructs and gathered data. These models consists of one or more hypothetical components which are intended to be explained by a set of indicators. A correct positioning of the items representative of the theory is everything in latent variable models (Borsboom (2008)).

Plenty of latent variable models have been developed in the past years depending on the data types of the variables to which they are applicable. In case both observed and latent variables are continuous, then the model is called a factor model (Derrick Norman Lawley and Maxwell (1971), Jöreskog (1971)). In case observed variables and latent variables are categorical and continuous respectively, then the model is known as Item Response Theory (Birnbaum (1968), Embretson and Reise (2000), Sijtsma and Molenaar (2002), Hambleton

and Swaminathan (2013)). In case both observed and latent variables are categorical, then the model is called latent class model (Pf and NW (1968), Goodman (1974)). Finally, if observed variables and latent variables are continuous and categorical respectively, then a mixture model (McLachlan, S. X. Lee, and Rathnayake (2019)). In Table 2.19, a summary of the most of famous latent models is demonstrated (Borsboom (2008)).

Table 2.19: A summary table of the variable types with the corresponding latent variable model.

| | Factor model | Latent class model | IRT model | Mixture model |
|---|---|---|---|---|
| Observed & Continuous | ✓ | | | ✓ |
| Observed & Categorical | | ✓ | ✓ | |
| Latent & Continuous | ✓ | | ✓ | |
| Latent & Categorical | | ✓ | | ✓ |

An example of a psychological construct is intelligence. There is no definitive measure of intelligence, but instead a wide range of observed variables could be used to, somewhere, approach intelligence such as memory capacity, ability to understand and thought processing speed. Mathematical models which consist of observed variables and aim to explain them in terms of latent variables are called latent variable models. Latent variable models form one of the essential parts of SEM, the measurement model.

## 2.14 Graph Theory and Notation in SEM

Graphical models are represented as a mathematical or statistical constructs connecting nodes (vertices) via edges (links). The nodes represent variables of interest in our dataset, and edges specify the relationships among them (Clark (n.d.)). In the science of statistics any model, including SEM, can be expressed as some form of a graphical model. Graphical models are sequences of adjacent edges that connects multiple two-variable pairs regardless of the directions of those edges. The graph represents all hypothesized connections, causal or non-causal, between any pair of variables. In SEM, there are restrictions in how two variables can be related. There is only one unique link, one coefficient, and variables are not allowed to have loop effect upon each other. Let's take for example two variables, a child's abusement and his parent's education, which are related to each other within a graphical model. These two variables have a unique coefficient which explains the relationship between them. It doesn't matter if the relationship is a simple association or a cause-and-effect, the point is that each of these coefficients is one of a kind.

There are 3 types of graphs:

- Directed graphs have all edges in the form of arrows that point away from a cause toward an effect. Such graphs imply a causal flow from the beginning of the path to the end. Directed acyclic graphs are the graphs where there are no feedback loops.

- Undirected graphs have no arrows merely denoting relations among the nodes. Such graphs convey statistical association, but not causation, between the variables at either end.

- Mixed graphs contain both directional and symmetric relationships.

  SEM is mainly captured by directed acyclic and mixed graphs.

Path diagram is a visual representation of the relationships among variables in a form of causal graphic model. SEM is one of the most widely used models when it comes to multivariate analysis. By nature, SEM consists of complex relationships which wouldn't be easy to understand without such visual objects. After selecting the appropriate variables for the study, researchers are responsible for the creation of the development of a path diagram. They call upon their personal observations, experience, research literature, common sense, and logic to carry out this task. The unique role of theory in constructing path diagram had been exposed by Bohrnstedt and Carter (1971), MacDonald (1977), Pedhazur, Kerlinger, et al. (1982), Bohrnstedt and Knoke (1982), Browne et al. (1993), Hetherington (2000).

1. Theory helps researchers to isolate groups of variable into a system of functional equations.

2. Sound theory reduces specification problem because researchers are able to identify relevant variables to be included in the model and irrelevant variable to be excluded, and the conditions under which a causal relationship is like to exist.

3. Theory not only guides researchers in specifying the logical causal ordering of variables into independent,intervening, and depended variables but also provides the general framework for investigating the nature of all relationships.

4. Theory plays an important prole in interpreting research findings because it is a primary frame of reference through which researchers understanding the contents and implications of their findings.

5. Theory guides researchers in determining how to assess the meaningfulness of a "weak" association and how to test for direction of influence and spuriousness, and the tenability of the model.Such decision rests not on the data,but rather on the theory that generated the causal model in the first place.

6. Theory may also provide us with a statement about the sign (- or +) and/or relative size of the direct effect of one construct on another.

   (Olobatuyi (2006))

Path diagrams, which originated by Wright (1921), is the one way to go for researchers who want to connect their projects with SEM community and is a necessary chapter that all publications involving SEM procedures must acquire. The adoption of this visual tool is one of the reasons for enthusiasm for structural modeling. Path diagrams provide researchers and theorists with unique advantages as stated by Bagozzi (1980), Heise (1969), J. G. Anderson (1973), Biddle and Marlin (1987), McClendon (1994).

1. Path diagrams make explicit the assumptions, variables, and hypothesized relationships in one's theory.

2. By clear definition of variables and operationalizations and the functional relationship among variables path diagrams add a certain degree of accuracy to one's theory and research effort.

3. In real life researchers are faced with social and physiological phenomena that involve many complex interactions and feed backs. Subsequently, the path diagram would constitute a useful method of understanding such systems of true relationships.

4. They provide a mechanism for constructing and testing internal adequacy of theories and measurements and also the degree of correspondence between theory and observation.

5. Path diagrams are visual representation of a complex argument. They may be lacking in precision but many researchers find that is a clearer and more efficient way of demarcating the relationships among multivariate data than an algebraic system of equations. Besides, they are more appealing to the statistically naive readers and authors than the length discussions or the presentation of tabular data.

   (Olobatuyi (2006))

As mentioned in previous sections, SEM includes cause-and-effect relationships among a variety sets of variables. In that sense researchers must be very careful in the construction of path diagrams. Bohrnstedt and Knoke (1982), Biddle and Marlin (1987), Land (1969), MacDonald (1977) and Heise (1975) state a couple of important points.

1. It is best to include all variables that are theoretically or empirically relevant the the study, and define them clearly because path models require more stringent theoretical specification than multiple regression models.

2. Each variable in the model should be represented by a brief acronym or symbol.

3. According to Heise (1975), conventionally, the following capital letters V,W,X,Y and Z are reserved for variables. The same letter maybe be used repeatedly for the variables but can be distinguished by attaching subscripts. For example if letter Y is used, all variables may be identified as Y1,Y2,Y3,Y4 etc. but what each acronym, symbol or letter signifies should be clearly defined. For instance, father's job (Y1), father's education (Y2), mother's job (Y3), mother's education (Y4).

   (Olobatuyi (2006))

The researcher apart from setting up his theoretical framework and choosing the appropriate variables to conduct path diagram of SEM, he must define some symbols to represent types of variables and relationships, errors and disturbances. In order for him to achieve that he needs to establish some conventions in the form of notations.

There are two types of variables, observed or measured variables and unobserved or latent variables. In this paper, the following notation will be used. Observed variables are represented by squares or rectangles (□), while latent variables by circles (○). The observed variables which are supposed to measure a latent variable are called indicators and are also represented by rectangles (□). The residual error variance terms which represents the error of the observed variables and the measurement error of the latent variables are denoted with arrows without a starting point ($\xrightarrow{x}$ □ and $\xrightarrow{x}$ ○, respectively). Thus, if the line has not starting point and ends in the same node, the $x$ coefficient indicates the amount of variance that is unexplained for the specific variable, also known as residual variance (for observed variables) or measurement error (for latent variables). Residual or error terms in the case of indicators, is equal to the unexplained variance of the factor that the corresponding indicator is supposed to measure. Some of this variance which is not explained is because of the random measurement error or score unreliability. Residual terms is another latent variable category in SEM, which can be associated with either observed variables or factors.

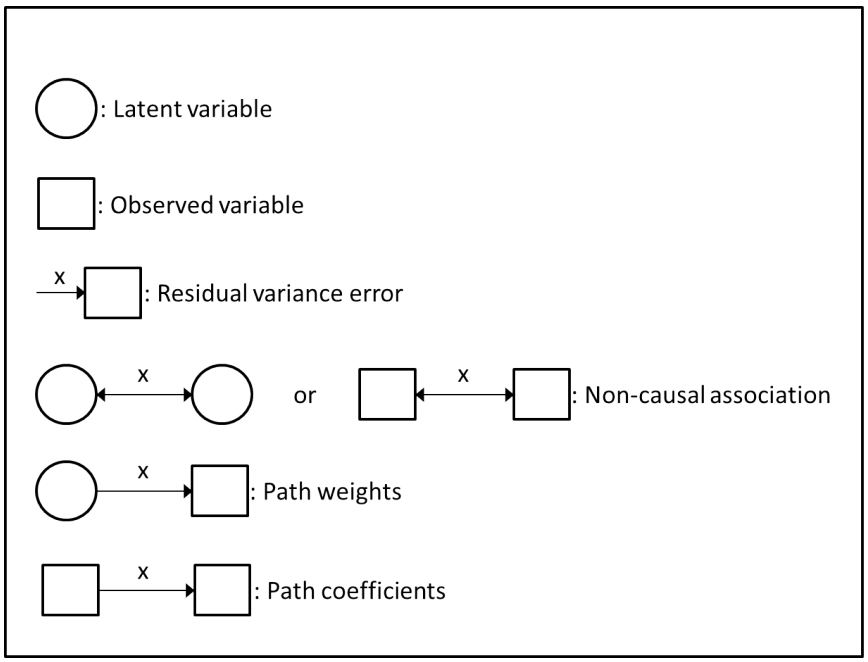



Figure 2.15: The symbolic conventions of the SEM path diagram used in this paper.

The relationships between latent and observed variables within a SEM model are represented with lines and arrowheads. A solid straight line between two variables indicates a direct causal direction where the arrowhead pointing from one variable to other shows the direction of causality ($\square \rightarrow \square$) (MacDonald (1977), Land (1969), Bohrnstedt and Knoke (1982), O. D. Duncan (1966), Biddle and Marlin (1987), MacCallum (1995)).

The following conventions are used in the path diagram:

- A directed (single-headed) arrow originating from an independent variable and ending at a dependent variable represents a direct causal effect of the independent variable (cause or predictor) on the dependent variable (effect or outcome). Each such arrow is labelled with a structural coefficient ($\square \xrightarrow{x} \square$). The absence of directional arrow from one variable to another implies the absence of direct effect. A variable in a structural equation model is referred to as "exogenous" if, and only if, it does not serve as an outcome variable in a given model. In a path diagram, this means there are no single-headed arrows pointing to it ($\square \xleftarrow{x} \square \xrightarrow{x} \square$). Otherwise, the variable is "endogenous" ($\square \xrightarrow{x} \square \xrightarrow{x} \square$).

- A bidirectional (double-headed) curved arrow represents a covariance, linking two variables or errors, that is not given causal interpretation ($\square \overset{x}{\leftrightarrow} \square$ or $\bigcirc \overset{x}{\leftrightarrow} \bigcirc$). Variables are assumed to be associated or correlated, but not causally related. This also suggests that such variable relations are influenced by other variables exogenous or external to the path model.

- A directed (single-headed) arrow originating from a latent variable and ends on an indicator ($\bigcirc \xrightarrow{x} \square$) represents a weight and a hypothetical relationship between the two variables.

- A directed (single-headed) arrow originating from an observed variable and ends on another observed variable ($\square \xrightarrow{x} \square$) represents a path coefficient and a hypothetical causal relation between the two variables.

(Hoyle (2012))

It is important to keep in mind that irrespective of the line type, all lines come along with a unique coefficient. In Figure 2.15, the SEM graphical notations which are going to be used in this paper are represented.

## 2.15 Lab: SEM

### 2.15.1 R Packages for SEM

In this section, R and its packages are tools that are going to be used in order to conduct SEM. Comprehensive R Archive Network (CRAN, at `http://www.cran.r-project.org`) is an up-to-date server hosts the base R software and has more than 3000 extension packages for defining and manipulating objects of various sorts. Both the R program in general and SEM packages for R in particular support object-oriented programming. Any package name followed by exclamation mark (!Package name) displays a window which provides info about the package via help guide. Also, vignette(Package name) shows the commands that are included into the package, leaving the parenthesis empty will result in displaying all the pre-installed R packages. This means that data, models, and analyses results can all be defined as classes with attributes and functions for manipulating class content. Researchers with no programming experience whatsoever may find working in R to be austere, but others should be able to adapt without great difficulty. The SEM packages described that follow next work only in batch mode processing.

In R, all packages are available for installation with various ways. Researcher can go to *Tools→Install Packages...* in the pull-down menu in the R software, type the package name in the *Packages* section and click *Install*. Alternatively, he can type and run *install.packages("Package Name")* in the R Console. His final option is to go to the window, click *Packages*, click *Install*, then type the package name in the *Packages* section and click *Install* (Figure 2.16). Regardless of the way, once the package is loaded then the researcher must activate it before the beginning of his analysis by typing and running *library("Package Name")* in the source editor. Any package name typed in the R console window followed by question mark (?Package name) displays another window which provides information and examples about the package via help guide. Also, vignette(Package name) shows the commands that are included into the package, leaving the parenthesis empty will result in displaying all the pre-installed R packages without discrimination.

Two Structural Equation modeling packages are dominating in every SEM analysis and are available in R, *sem* and *lavaan*. In conducting SEM, *sem* and *lavaan* require commands with arguments that specify the SEM model, parameter names, data file input, start values, optimization routine, and missing data. Throughout this section both packages will be frequently used so the way they function and interpret will be explicitly analyzed in later chapters. *sem* package has been around for many years. It is chronologically one of the first SEM packages for R created, constantly developed and updated by Fox 2012. Fox (2006) was the main representative of the package publishing an article in the *Structural Equation Modeling journal*. The *sem* package provides basic SEM techniques for analyzing and fitting for both SEM measurement and structural model. It contains the *sem* function fits observed and latent variable models by full-information maximum likelihood assuming multivariate-normal data or another method, and employs a path-centric interface for model specification (Hoyle (2012)). Additionally, *sem* package has capabilities for calculating robust standard errors and bootstrapping. A version of maximum likelihood estimation for incomplete raw data files is also available. Models are specified using McArdle–McDonald RAM notation which will be analyzed later on. *lavaan* (latent variable analysis) package created by Rosseel (2012) includes a collection of tools that can understand and estimate a wide family of SEM

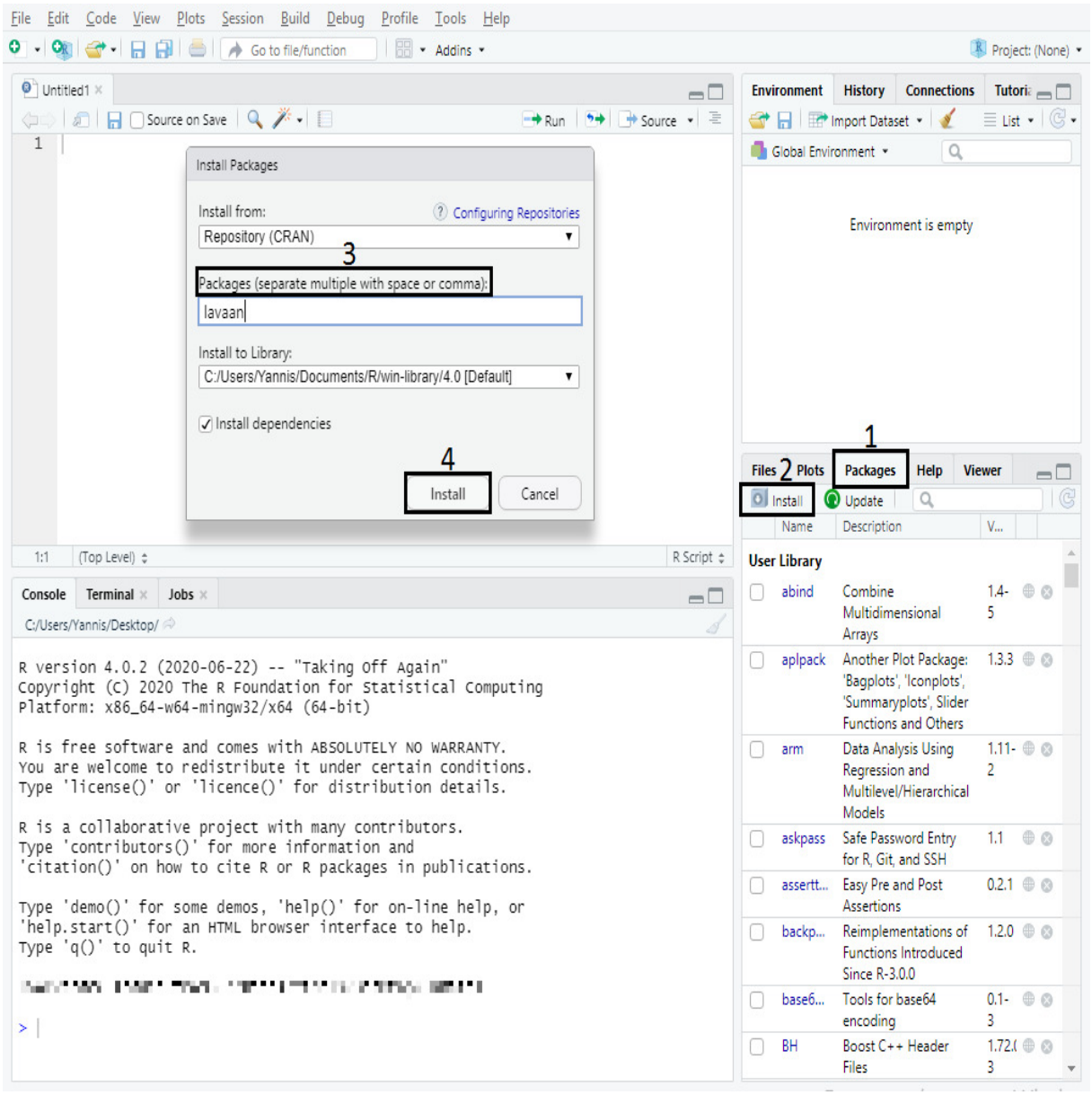



Figure 2.16: Step-by-Step picture for package installation. In this picture the *lavaan* package is installed.

latent variable models such factor analysis, structural equation, longitudinal, multilevel, latent class, item response, and missing data models (Skrondal and Rabe-Hesketh (2004),S.-Y. Lee (2007),B. O. Muthén (2002)). The package can analyze models with ordinal or continuous outcomes with severely non-normal distributions, and incomplete data files. There are also options for bootstrapping. Models are specified in a text and equations based language for defining regression models and measurement models. *lavaan* aims to be attractive while at the same time to meet the expectations of many applied researchers who have not used R before in the past and used to interact with commercial SEM softwares equipped with various modeling features. This package is a living proof that it is possible to have an open-source SEM program which can compete head to head with the commercial programs enabling direct access to the SEM code. Rosseel (2012) was the main representative of the package publishing an article about the *lavaan* package in the *Journal of Statistical Software*. Beaujean (2014) also gives examples of latent variable analyses using *lavaan*. The distinct feature in the *lavaan* package is the mimic argument which permits similarity to either the default Mplus or EQS program (Kline (2015)). This means that since the mimic option makes a smooth transition possible from lavaan to one of the major commercial programs, and back students who received initial instruction in SEM with lavaan should have little difficulty using other paid SEM programs in the future. You can obtain more information and examples for either package by typing ?sem or ?lavaan in the R console window. *lavaan* covers most of the SEM analysis and will be analyzed through examples.

*semTools* tests for measurement invariance in factor analysis. It can also estimate the power of certain types of SEM significance tests. The *semPlot* package (Epskamp and Stuber (2014)) generates path diagrams. It can also create model syntax for one program, such as *lavaan*, based on the output of another program, such as *sem* Kline (2015). More information about R, R installation, and R packages can be found at: `www.cran.r-project.org`.

Therefore, in order to conduct SEM in R certain packages are required. The packages are lavaan, semPlot and semTools. The sequence of R commands which are used to install (if you haven't already installed them) and load the packages is demonstrated below:

```
>install.packages("lavaan") #If you haven't installed already
>library(lavaan) #Loading the package

>install.packages("semPlot") #If you haven't installed already
library(semPlot) #Loading the package

>install.packages("semTools") #If you haven't installed already
>library(semTools) #Loading the package
```

### 2.15.2 The Company Dataset

### 2.15.3 Data Overview

And now in this section, a SEM example will be demonstrated with the help of all the information gathered from the data along the paper. In this section, variables which were used to conduct regression analysis, PCA and cluster analysis, in addition with some extra variables from the model will be used to construct the structural equation model.

However, to have the bigger picture as well, the full dataset is demonstrated in Table 2.20. This dataset contains 1470 observations and 35 variables. The dataset is about a company which has collected data from its employees. The variables demonstrated in Table 2.20 are manipulated towards the correct data type. The next step is to distinct which are appropriate for structural equation modeling.

There are many reasons why some of the variables must be removed from the SEM analysis:

1. There are variables which do not provide any additional value to the dataset and will be removed from the original model (see Table 2.20).

2. There are variables which are unordered categorical variables. The same variables are omitted because they require special treatment during SEM model estimation. Hence, these variables will be ignored to demonstrate a simplified SEM model. Only numerical and ordered categorical variables disguised as pseudo-numeric variables will be used in this example (see Table 2.21).

3. There are variables which have comparably low correlation with the rest of the dataset. SEM is a high correlation technique and, therefore, these variables will not eventually make it into the final SEM model (see Table 2.22).

All the variables which are not included into the SEM modeling and correspond to the $1^{st}$ justification as demonstrated in Table 2.20 and analyzed below:

- EmployeeCount (1): The variable EmployeeCount takes the value of 1 in every case. It describes the number of employees for which the case was recorded. This information obliviously is of no use for the analysis and will be removed.

- EmployeeNumber (1): The variable EmployeeNumber takes a unique value for each case. It displays a unique value of ID which corresponds to each case. Of course, this information is useless for the analysis (in R each row corresponds to a case).

- StandardHours (1): The variable StandardHours takes the value of 80 in every case and hence is not useful for the analysis.

- Over18 (1): The variable Over18 takes the value of "Yes" in every case and hence is not useful for the analysis. It shows if the individual is above 18 years old.

All the variables which are not included into the SEM modeling and correspond to the $2^{nd}$ justification as demonstrated in Table 2.21 and analyzed below (the list continues after the tables):

- Attrition (2): The variable Attrition takes the values 0 and 1 which correspond to "No" and "Yes". The variable shows if the individual suffers from attrition at his job.

- Department (2): The variable Department takes the values 1,2 and 3 which correspond to "Human Resources", "Research & Development" and "Sales", respectively. The variable shows the department of the company in which the individual works.

- EducationField (2): The variable EducationField takes the values 1,2,3,4,5 and 6 which correspond to "Human Resources", "Life Sciences", "Marketing", "Medical", "Technical Degree" and "Other", respectively. The variable shows the education field of expertise of the individual.

- Gender (2): The variable Gender takes the values 0 and 1 which correspond to "Female" and "Male", respectively. The variable represents the gender of the individual.

Table 2.20: The 35 variables of the original dataset. 4 variables are removed due to reason 1. The viable-meaningful dataset consists of 31 variables.

| Variable | Data type |
|---|---|
| Age | Numerical |
| Attrition | Unordered Categorical |
| BusinessTravel | Ordered Categorical |
| DailyRate | Numerical |
| Department | Unordered Categorical |
| DistanceFromHome | Numerical |
| Education | Ordered Categorical |
| EducationField | Unordered Categorical |
| EmployeeCount (Removed) | - |
| EmployeeNumber (Removed) | - |
| EnvironmentSatisfaction | Ordered Categorical |
| Gender | Unordered Categorical |
| HourlyRate | Numerical |
| JobInvolvement | Ordered Categorical |
| JobLevel | Ordered Categorical |
| JobRole | Unordered Categorical |
| JobSatisfaction | Ordered Categorical |
| MaritalStatus | Unordered Categorical |
| MonthlyIncome | Numerical |
| MonthlyRate | Numerical |
| NumCompaniesWorked | Numerical |
| Over18 (Removed) | - |
| OverTime | Unordered Categorical |
| PercentSalaryHike | Numerical |
| PerformanceRating | Ordered Categorical |
| RelationshipSatisfaction | Ordered Categorical |
| StandardHours (Removed) | - |
| StockOptionLevel | Unordered Categorical |
| TotalWorkingYears | Numerical |
| TrainingTimesLastYear | Numerical |
| WorkLifeBalance | Ordered Categorical |
| YearsAtCompany | Numerical |
| YearsInCurrentRole | Numerical |
| YearsSinceLastPromotion | Numerical |
| YearsWithCurrManager | Numerical |

Table 2.21: The 31 variables of the original dataset. 8 variables are removed due to reason 2. The dataset which will be used to construct the SEM model consists of 23 variables. The variables retained are of Numerical and Ordered Categorical type.

| Variable | Data type |
|---|---|
| Age | Numerical |
| Attrition (removed) | Unordered Categorical |
| BusinessTravel | Ordered Categorical |
| DailyRate | Numerical |
| Department (removed) | Unordered Categorical |
| DistanceFromHome | Numerical |
| Education | Ordered Categorical |
| EducationField (removed) | Unordered Categorical |
| EnvironmentSatisfaction | Ordered Categorical |
| Gender (removed) | Unordered Categorical |
| HourlyRate | Numerical |
| JobInvolvement | Ordered Categorical |
| JobLevel | Ordered Categorical |
| JobRole (removed) | Unordered Categorical |
| JobSatisfaction | Ordered Categorical |
| MaritalStatus (removed) | Unordered Categorical |
| MonthlyIncome | Numerical |
| MonthlyRate | Numerical |
| NumCompaniesWorked | Numerical |
| OverTime (removed) | Unordered Categorical |
| PercentSalaryHike | Numerical |
| PerformanceRating | Ordered Categorical |
| RelationshipSatisfaction | Ordered Categorical |
| StockOptionLevel (removed) | Unordered Categorical |
| TotalWorkingYears | Numerical |
| TrainingTimesLastYear | Numerical |
| WorkLifeBalance | Ordered Categorical |
| YearsAtCompany | Numerical |
| YearsInCurrentRole | Numerical |
| YearsSinceLastPromotion | Numerical |
| YearsWithCurrManager | Numerical |

Table 2.22: The 23 candidate variables dataset which will be used to construct a SEM model. 12 variables are removed due to reason 3. The variables which will be included into the final SEM model are 11.

| Variable | Data type |
|---|---|
| Age | Numerical |
| BusinessTravel (removed) | Ordered Categorical |
| DailyRate (removed) | Numerical |
| DistanceFromHome (removed) | Numerical |
| Education (removed) | Ordered Categorical |
| EnvironmentSatisfaction (removed) | Ordered Categorical |
| HourlyRate (removed) | Numerical |
| JobInvolvement (removed) | Ordered Categorical |
| JobLevel | Ordered Categorical |
| JobSatisfaction (removed) | Ordered Categorical |
| MonthlyIncome | Numerical |
| MonthlyRate (removed) | Numerical |
| NumCompaniesWorked | Numerical |
| PercentSalaryHike | Numerical |
| PerformanceRating | Ordered Categorical |
| RelationshipSatisfaction (removed) | Ordered Categorical |
| TotalWorkingYears | Numerical |
| TrainingTimesLastYear (removed) | Numerical |
| WorkLifeBalance (removed) | Ordered Categorical |
| YearsAtCompany | Numerical |
| YearsInCurrentRole | Numerical |
| YearsSinceLastPromotion | Numerical |
| YearsWithCurrManager | Numerical |

Table 2.23: The final 11 variables dataset that will be included into the final SEM model.

| Variable | Data type | |
|---|---|---|
| Age | Numerical | 41 49 37... |
| JobLevel | Ordered Categorical | 2 2 1... |
| MonthlyIncome | Numerical | 5993 5130 2090... |
| NumCompaniesWorked | Numerical | 8 1 6... |
| PercentSalaryHike | Numerical | 11 23 15... |
| PerformanceRating | Ordered Categorical | 3 4 3... |
| TotalWorkingYears | Numerical | 8 10 7... |
| YearsAtCompany | Numerical | 6 10 0... |
| YearsInCurrentRole | Numerical | 4 7 0... |
| YearsSinceLastPromotion | Numerical | 0 2 3... |
| YearsWithCurrManager | Numerical | 5 7 0... |

- JobRole (2): The variable JobRole takes the values 1,2,3,4,5,6,7,8 and 9 which correspond to "Healthcare Representative", "Human Resources", "Laboratory Technician", "Manager", "Manufacturing Director", "Research Director", "Research Scientist", "Sales Executive", "Sales Representative", respectively. The variable displays the role of the individual in the company.

- MaritalStatus (2): The variable Marital Status takes the values 1,2 and 3 which correspond to "Divorced", "Married" and "Single", respectively. The variable describes the relationship status of the individual.

- OverTime (2): The variable OverTime takes the values 0 and 1 which correspond to "No" and "Yes", respectively. The variable shows if the individual works overtime shifts in the company.

- StockOptionLevel (2): The variable StockOptionLevel takes the values 0,1,2 and 3..

Data structure and information of the final 11 variables which will be used in the SEM model is available below:

1. Age: The variable Age takes numerical variables. This variables shows the age of the individual.

2. JobLevel: The variable JobLevel takes the values 1, 2, 3, 4 and 5 which correspond to "Poor", "Fair", "Good", "Very good" and "Excellent", respectively. This variable shows the quality of the work of the individual in the company.

3. MonthlyIncome: The variable MonthlyIncome takes numerical values. The variable shows the monthly income of each individual.

4. NumCompaniesWorked: The variable NumCompaniesWorked takes numerical values. The variable shows the amount of companies in which the individual has worked.

5. PercentSalaryHike: The variable PercentSalaryHike takes numerical values. The variable shows the percentage of salary increase which the individual receives.

6. PerformanceRating: The variable PerformanceRating takes the values 1,2,3 and 4 which correspond to "Low", "Good", "Excellent" and "Outstanding", respectively. The variable shows the performance rate of the individual.

7. TotalWorkingYears: The variable TotalWorkingYears takes numerical values. The variable shows the total working years of the individual.

8. YearsAtCompany: The variable YearsAtCompany takes numerical values. The variable shows the amount of years of the individual in the company.

9. YearsInCurrentRole: The variable YearsInCurrentRole takes numerical values. The variable shows the amount of years of the individual in the current role of the company.

10. YearsSinceLastPromotion: The variable YearsSinceLastPromotion takes numerical values. The variable shows the amount of years of the individual since his/her last promotion in the company.

11. YearsWithCurrManager: The variable YearsWithCurrManager takes numerical values. The variable shows the amount of years of the individual with the current manager in the company.

### 2.15.4 Model Specification

The modeling of the relevant equations of the dataset puts a theoretical model into test. This theoretical is also based upon theory. In other words, research and investigation of the theory behind the variables is a prerequisite in order to model and graph the relationship between them. Hence, some variables might form latent variables which then can be used into the final SEM model.

In SEM, model specification is the first out of the four main steps. It is critical for the researcher to find the combination of the variables which can be justified both statistically and theoretically. The optimal model will yield good results both in representing the data, the relationship between the variables and the theory. The next steps will be all invalid without the correct specification of the model. In a way, path analysis doesn't provide specification of the relationships between the variables of a model. All that it does is estimate the relations based upon theory. That is the reason why the final SEM specification is very important.

Path analysis calculates the strength of the relations used a variance-covariance matrix as input. A variance-covariance matrix has the variance of each variable in its main diagonal and the covariance between them in the off-diagonal positions. The researcher has the option to use raw data, correlation matrix or covariance matrix as an input to conduct SEM. If a correlation matrix is used as input, then most statistical softwares will convert it in variance-covariance matrix. This is done by utilizing the mean and the standard deviations. When raw data are used as an input, by default the variance-covariance matrix. By default SEM uses the variance-covariance matrix. Path analysis allows for a simple association between any two variables to be decomposed into the compound paths which connect them. The amount and the type of the complex paths between any two variables are represented via a restricted model suggested by the researcher (Kapsali (2020)).

The model specification occurs when the researcher defines the relationships which supposed to exist or not exist between the latent and observed variables. Usually a latent variable is measured through multiple observed variables, but there exact number is not known. However, as the number of observed variables which are combined to explain a latent variable increases, the latter is more accurately represented from all its aspects.

The model specification includes the representation of the theoretical relationships between the variables. This representation is attributed as a model which consists of multiple equation. These equations define the paths and the parameters of the model. There are three kinds of parameters which are the variances, the covariances and the directional effects. The latter are presented by finite arrows with beginning and end. The directional effects which display the relations between observed and latent variables are called path coefficients. The path coefficients of the paths which connect observed variables with latent variables must be above 0.70 optimally. This is a good indication that the variables represent the latent variable successfully. Some scientists accept values above 0.4 as well. The values of the rest of the path coefficients vary from -1 to 1. Path coefficients which have value close to zero have almost no effect on the corresponding value. The parameters, are either set to zero and not estimated or left free to be estimated. In the case of the latent variable. one parameter of one of its components is set to zero to allow for the latent variable to vary.

Algebraically, every parameter of the model can be estimated by the variance-covariance matrix of the sample which is created by the variables. The number of unique values of the matrix must be equal to the number of parameters to be estimated for the model. The parameters must be estimated in such way so that the difference between the actual covariance matrix and the hypothesized covariance matrix is minimized. Mathematically, that $\Sigma(\theta) = S$, where $\Sigma$ is the hypothesized matrix of observed variables, $\theta$ is the parameters of the model to be estimated and $S$ is the covariance matrix of the observed variables of the

sample.

There are several ways to represent the matrices of the SEM models. Two of the most commonly used are LISREL and Reticular Action Model, also known as RAM. The RAM approach expresses the hypothesized covariance matrix through 3 matrices (Kapsali (2020)). The $A$, $S$ and $F$ matrix. Each of the component matrices of the RAM approach will be analyzed below.

The hypothesized matrix, $C$, is displayed in Equation 2.73:

$$C = \left[\begin{array}{c|ccccc} & x_1 & x_2 & \cdots & y_7 & y_8 \\ \hline x_1 & Var(x_1) & Cov(x_1,x_2) & \cdots & Cov(x_1,y_7) & Cov(x_1,y_8) \\ x_2 & Cov(x_2,x_1) & Var(x_2) & \cdots & Cov(x_2,y_7) & Cov(x_2,y_8) \\ \vdots & \vdots & & \ddots & & \vdots \\ y_7 & Cov(y_7,x_1) & Cov(y_7,x_2) & & Var(x_7) & Cov(y_7,y_8) \\ y_8 & Cov(y_8,x_1) & Cov(y_8,x_2) & \cdots & Cov(y_8,y_7) & Var(y_8) \end{array}\right] \tag{2.73}$$

$A$ is a non-symmetric square matrix which contains paths. The rows and columns of the matrix are equal to the number of the variables in the model. The variable that is in the beginning of a path is in the column and the variable in the row is in the end of the path. A variable which doesn't correspond to a path has value of zero in the matrix. An example of a $A$ matrix is demonstrated in Equation 2.74.

$$A = \left[\begin{array}{c|cc} & x & y \\ \hline x & 0 & 0 \\ y & \beta_1 & 0 \end{array}\right] \tag{2.74}$$

$S$ is a symmetrical matrix which contains the covariance or associations and the residual variances. Additionally, it is a square matrix with the same dimensions as $A$. In the positions of the main diagonal of the matrix $C$, the residual variances are located. If there are covariances among the variables which are not explained by the paths of matrix $A$, then these are the residual covariances. These covariances are located in the elements in the off-diagonal positions of the matrix. The residual variances are sub-cases of the residual covariances. Recall, that in covariances the anti-transposition property is true. This means that if the $i, j$ element of a matrix exists, then the $j, i$ must be filled as well. In case, the correlation matrix is used as an input those two values are the same. An example of the $S$ matrix based on the $A$ matrix is demonstrated in Equation 2.75.

$$S = \left[\begin{array}{c|cc} & x & y \\ \hline x & \sigma_x^2 & 0 \\ y & 0 & \sigma_{e_y}^2 \end{array}\right] \tag{2.75}$$

The $F$ matrix filters the observed variables and is a version of the identity matrix, $I$. This matrix has as many columns as the number of variables of the model and rows as the number of observed variables of the model. In this matrix $F$, there are two values, 0 and 1. A value of 1 is placed in the corresponding row and column of all the observed variables of the model. A value of 0 is placed in the rest of the positions of the matrix $F$. The final matrix $F$ has the number of rows and columns equal with the number of variables in the model. An example of the $F$ matrix is demonstrated Equation 2.76.

$$F = \left[\begin{array}{c|cc} & x & y \\ \hline x & 1 & 0 \\ y & 0 & 1 \end{array}\right] \tag{2.76}$$

The goal of the SEM procedure is to solve the free parameters in those matrices. $A$, $S$ and $F$ have the same number of rows and columns which correspond to the latent variables of the model.

The first step in every SEM project is to specify the model. In this step, the measurement and the structural model will be specified based on prior research and theory. Additionally, recall that SEM is a correlation/covariance technique and hence, works better with relatively high correlated variables. The specific 23 variables in Table 2.22 have large differences in scaling. In order to overcome this problem, in this example the correlation matrix will be used as an input for SEM modeling. The correlation matrix of these 23 variables is demonstrated in Figure 2.17.

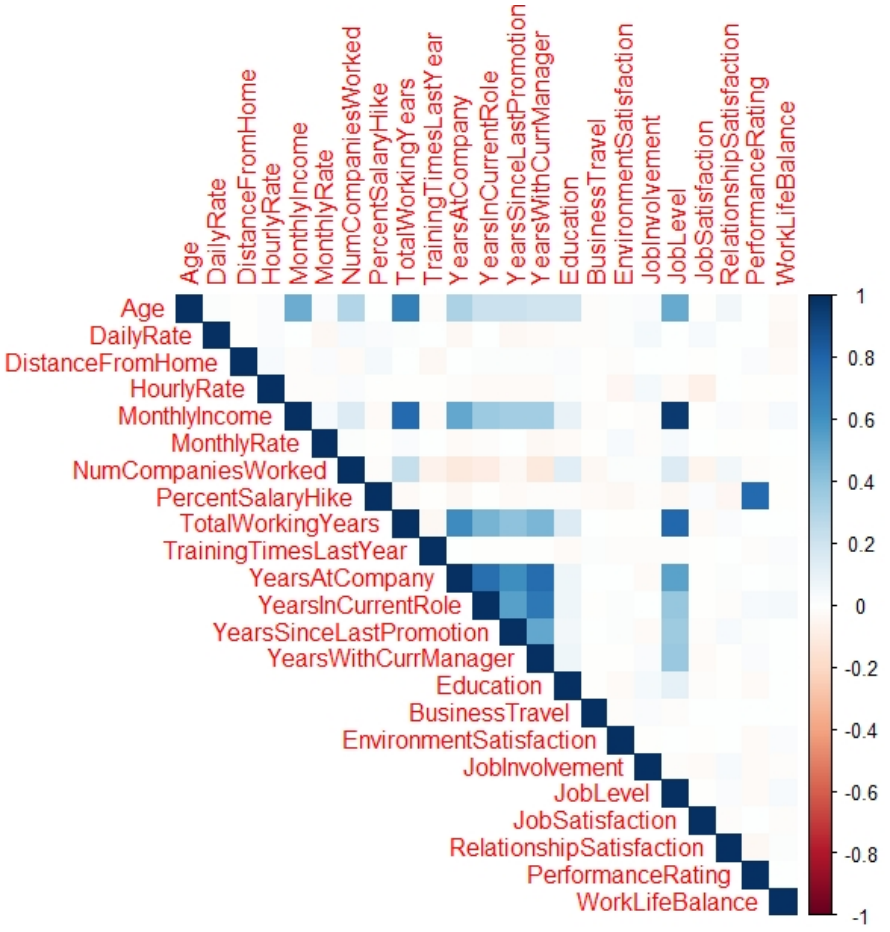



Figure 2.17: The correlation table of the candidate variables for the SEM model.

### 2.15.4.1 Measurement Model

Starting of with the measurement model, notice how the variables YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager seem to have strong correlation between them. Additionally, recall that in cluster analysis, these exact variables were dominant in terms of correlation. Finally, all these results suggest that it would be a good idea to make a latent variable with those variables. Notice how every variable measures the years inside of the company in many different scenarios (at company, in current role, since last promotion and with current manager). Therefore, the combination of those variables into a latent variable theoretically is correct. In order to make a latent variable in R, you input the desired latent name on the left side, the components on the right side, and an equal-circumflex symbol (=∼) in between. Of course, the name of variable must be

representative of the overall characteristic of its components. In this example, the latent variable will be named CY, which stands for Company Years.

The resulted plot of the latent model is displayed in Figure 2.18 and the estimation of the latent below it. The latent variable is graphed in a circle and the observed variable with a squared rectangle. Additionally, the names of the variables are sorted due to space capacity. Hence, Company Years, YearsAtCompany, YearsSinceLastPromotion, YearsInCurrentRole and YearsWithCurrManager are represented by CY, YAC, YSL, YIC and YWC, respectively. In this example, CY is the latent variable and YAC, YSL, YIC and YWC are the observed variables or components of the latent. The arrow which starts from a latent variable and ends on an observed variable represents the association between the two in the measurement model. For example, the arrow which connects CY and YAC represents the association between CY and YAC. The value which is above the arrow is the weight and shows the degree of association between the latent variable and the corresponding observed variable. For example, the weight between CY and YAC is equal to 0.92, which indicates that the two are highly associated. The dotted arrow is the value of the parameter which is 1 and not estimated. In this example, the dotted arrow is between CY and YSL and the weight is equal to 0.65. Finally, the arrows which have no starting point and just point in a variable are the measurement errors of the corresponding variables. For example, the measurement error of YearsAtCompany is equal to 0.16.

The interpretation of the output shows good results for the latent variable. The first note is that all the estimates are statistically significant with a p-value of 0. In the first section called `Latent Variables`, the weight of each of the arrows between the latent variable and an observed variable is displayed in the `Std.lv` and `Std.all` columns. In the fourth column the latent variable is the only one which is standardized, while in the fifth column both latent and observed variables are standardized. The latter solution is usually considered the "completely standardized solution". Notice how Company Years is highly associated with every single of its components with weights of 0.655, 0.835, 0.918 and 0.838. In the second section called `Variances`, the `Std.lv` and `Std.all` columns give the measurement errors. The measurement error represents the amount of the variance of the corresponding observed variable which is left unexplained. According to the output the results of the variables are 0.572 0.303 0.157 and 0.298. Finally, the $R^2$ values of the model is another good indication that the specific variables load pretty well into the latent variable.

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  CY =~
    YrsSncLstPrmtn    1.000                               0.654    0.655
    YearsInCrrntRl    1.275    0.047   27.223    0.000    0.835    0.835
    YearsAtCompany    1.402    0.049   28.838    0.000    0.918    0.918
    YersWthCrrMngr    1.280    0.047   27.291    0.000    0.837    0.838

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .YrsSncLstPrmtn    0.571    0.023   25.152    0.000    0.571    0.572
   .YearsInCrrntRl    0.303    0.015   20.526    0.000    0.303    0.303
   .YearsAtCompany    0.157    0.012   12.595    0.000    0.157    0.157
   .YersWthCrrMngr    0.298    0.015   20.356    0.000    0.298    0.298
    CY                0.428    0.031   13.631    0.000    1.000    1.000

R-Square:
                   Estimate
```

```
    YrsSncLstPrmtn    0.428
    YearsInCrrntRl    0.697
    YearsAtCompany    0.843
    YersWthCrrMngr    0.702
```

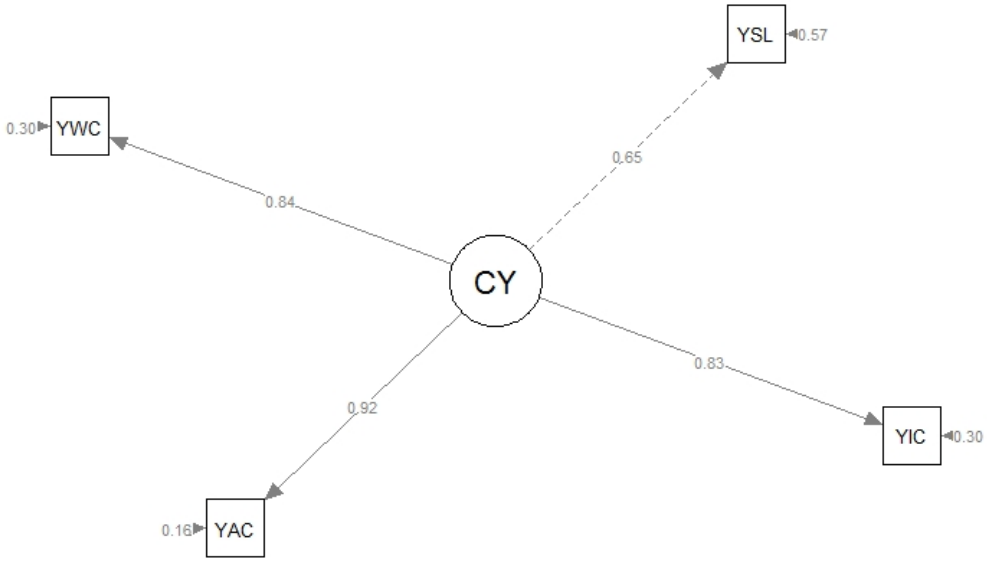



Figure 2.18: The measurement model.

The form of Figure 2.18 in terms of code is the following:

```
#The measurement model
    >sem_model<-'
  #Measurement Models
    ##Company Years
    CY =~ YearsSinceLastPromotion  + YearsInCurrentRole + YearsAtCompany
    + YearsWithCurrManager
'
#
>semfit<-sem(sem_model,sample.cov=data.cor,sample.nobs=1470)
>summary(semfit,standardized=TRUE,rsquare=TRUE)
```

#### 2.15.4.2 Regressions

The next step is to specify the statistically significant regressions in the SEM model. In general, a variable which is highly correlated with many variable is considered a good dependent variable. Notice how, in Figure 2.18, MonthlyIncome is highly correlated with TotalWorkingYears, JobLevel with a Pearson's correlation of 0.77 and 0.95, respectively. Hence, a

multiple linear regression is made where MonthlyIncome is regressed on TotalWorkingYears and JobLevel. Before adding the regression in the SEM model, it is recommended to confirm the relationships between the variables. For this reason, the linear model function, `lm`, is used first to take a look into the regression itself. The piece of code which performs the regression alongside with the corresponding output is the following:

```
#The multiple linear regression specification
>sem.reg1<-lm(MonthlyIncome~ JobLevel + TotalWorkingYears,data=numdf)
#The regression's output
>summary(sem.reg1)
```

The function `summary` reveals the regression's output which is demonstrated below. The results are successful for the regression. JobLevel and TotalWorkingYears are positively effecting MonthlyIncome with coefficients of 3788.378 and 46.082, respectively. These results are significant at the popular 0.05 threshold, with the p-values of the two predictors being 2e-16 and 4.34e-09, respectively. On top of that, the predictors jointly are significant to the model with a p-value of $2.2e-16 < 0.05$. Finally, the model yields a really good $R^2$ of 0.9053.

```
    Call:
lm(formula = MonthlyIncome ~ JobLevel + TotalWorkingYears, data = numdf)

Residuals:
    Min      1Q  Median      3Q     Max
-5425.2  -924.7    83.0   791.2  3917.5

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1835.862     80.019 -22.943  < 2e-16 ***
JobLevel           3788.378     54.843  69.077  < 2e-16 ***
TotalWorkingYears    46.082      7.802   5.906 4.34e-09 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1450 on 1467 degrees of freedom
Multiple R-squared:  0.9053,Adjusted R-squared:  0.9052
F-statistic:  7014 on 2 and 1467 DF,  p-value: < 2.2e-16
```

Additionally, TotalWorkingYears is highly correlated with JobLevel with a Pearson's correlation of 0.78. In addition to that, NumCompaniesWorked as a predictor of TotalWorkingYears is theoretically correct and will be examined, despite its low correlation index. Hence, a multiple linear regression is made where TotalWorkingYears is regressed on JobLevel and NumCompaniesWorked. The piece of code which performs the regression alongside with the corresponding output is the following:

```
#The multiple linear regression specification
>sem.reg2<-lm(TotalWorkingYears~JobLevel + NumCompaniesWorked,data=numdf)
#The regression's output
>summary(sem.reg2)
```

The function `summary` reveals the regression's output which is demonstrated below. The results are successful for the regression. JobLevel and NumCompaniesWorked are positively effecting TotalWorkingYears with coefficients of 5.36921 and 0.40115, respectively. These results are significant at the popular 0.05 threshold, with the p-values of the two predictors being 2e-16 and 2.39e-15, respectively. On top of that, the predictors jointly are significant to the model with a p-value of $2.2e-16 < 0.05$. Finally, the model yields a decent $R^2$ of 0.6281.

```
Call:
lm(formula = TotalWorkingYears ~ JobLevel + NumCompaniesWorked,
    data = numdf)

Residuals:
     Min       1Q   Median       3Q      Max
-12.7977  -3.8559  -0.4867   2.4821  24.3738

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.88254    0.28113  -3.139  0.00173 **
JobLevel            5.36921    0.11307  47.485  < 2e-16 ***
NumCompaniesWorked  0.40115    0.05011   8.006 2.39e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.748 on 1467 degrees of freedom
Multiple R-squared:  0.6281,Adjusted R-squared:  0.6276
F-statistic:  1239 on 2 and 1467 DF,  p-value: < 2.2e-16
```

There are few more adjustments that have to made in order for the SEM model to be consider more complete. This is the addition of the exogenous variables in the model. An exogenous variable is this example is Age because it out of the control or out of the system of the researcher. Hence, it cannot be predicted by any variable, but only be used as predictor. By examining the variable Age, it is important to notice that the two most higher correlation of this variable is with TotalWorkingYears and JobLevel. Theoretically, Age can effect TotalWorkingYears because the productiveness of a person is directly affected by its Age, it makes sense. Additionally, JobLevel is affected by Age for the same reason. With a closer look of Figure 2.17, one can notice a high Pearson's correlation between PercentSalaryHike and PerformanceRating (0.77). This implies that possibly the two variables are associated. However, after many tests, the direction of the causation is not clear. Hence, the covariance of the two variables will be included in the model. Finally, since the latent variable Company Years consists of YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion and YearsWithCurrManager, it is possible for it to be used as predictor on JobLevel. In theoretical basis, it makes sense for a latent variable which is characterised as years at company to act as a predictor for Job level which represents the quality of the work of the individual. The quality of the work of every individual in a company is only increased as its experience increases.

So, now its time for these 2 regressions and the adjustments above to be added into the SEM model. In order to add the regressions in R, the outcome is written in the left side and the predictors in the right side with a circumflex symbol ($\sim$) in between just like in classic linear model specification. The numbers in the lines which start and end in observed variables are called path coefficients and range from -1 to 1. In order for the exogenous

effects of Age to be included again ∼ is used with Age in the right side and the predictors in the left side. Finally, for the covariance to be added between PercentSalaryHike and PerformanceRating the two variables are written with a double ∼ symbol in the middle (∼∼) because their not a clear direction of the effect. In R, the SEM model has now the following final form in terms of graph and code:

```
>sem_model<-'
  #Measurement Models
    ##Company Years
    CY =~ YearsSinceLastPromotion  + YearsInCurrentRole + YearsAtCompany
    + YearsWithCurrManager

  #Regressions
  MonthlyIncome~ JobLevel + TotalWorkingYears
  TotalWorkingYears~ JobLevel + NumCompaniesWorked
  JobLevel~ CY

  #Exogenous effects
    ##Age
    TotalWorkingYears~ Age
    JobLevel~Age

  #Covariances
  PercentSalaryHike~~PerformanceRating

#Graphing the model
  semPaths(semfit,what="paths",whatLabels = "std"
         ,layout="spring"
         ,style="Lisrel"
         ,rotation=2,sizeLat2=10
         ,sizeLat=10
         ,sizeMan=4
         ,residScale=10
         ,font=2
         ,label.cex=1.3)
```

In Figure 2.19, the variable is the circle, CY is the latent variable and the rest of the variables which are on square rectangles are the observed variables. As already mentioned, YearsAtCompany, YearsSinceLastPromotion, YearsInCurrentRole and YearsWithCurrManager are the variables which load into CY. Additionally, notice that Age and NumCompaniesWorked are exogenous variables because there is no arrowhead towards them. The rest of the variables are endogenous. Notice that the weights of each of the 4 variables which are assumed to be explained by the latent variable CY load strongly. Their values are all above 0.7 which is a strong indication of a correct latent variable. The dotted line between the latent variable and YearsSinceLastPromotion is dotted. This is because the specific variable's parameter was chosen to be set to 1 for the rest of the variables to vary. Hence, the estimate of the specific variable is not available. Finally, the arrows which have no beginning on each of the variables represent the measurement error. Each of the endogenous variables come along with a value which represents the amount of variance which is not explained with the current model as it stands.

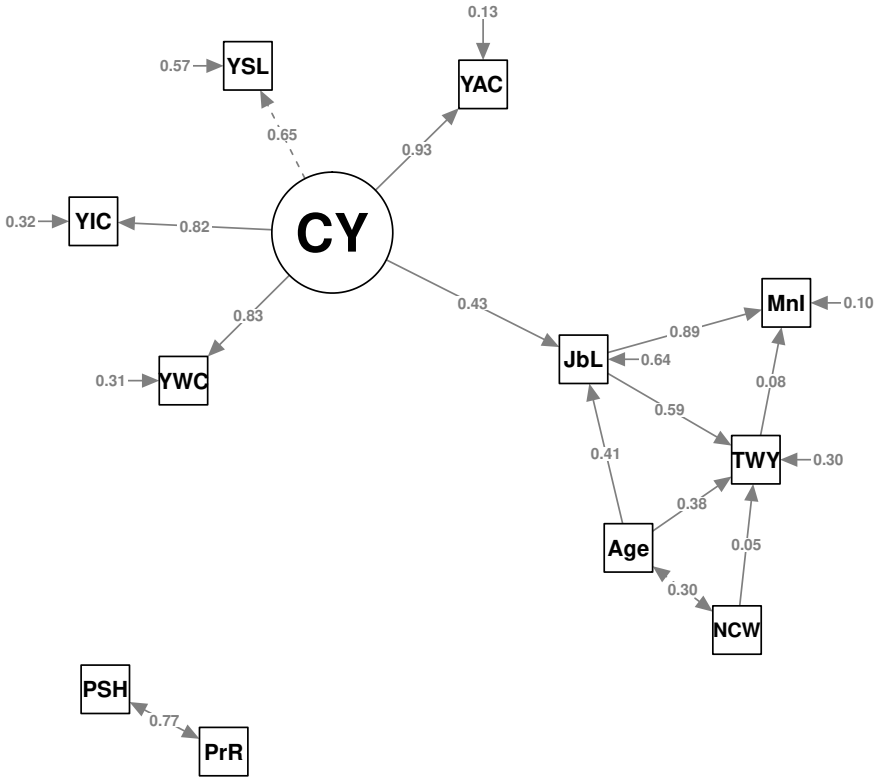



Figure 2.19: The graphical representation of the final SEM model.

To specify the model with the RAM approach, the three matrices, A, S and F will be demonstrated next using the command `semMatrixAlgebra`. The following lines of code will display each one of the three matrices.

```
#Matrix A of RAM approach
>semMatrixAlgebra(semfit, A)

#Matrix S of RAM approach
>semMatrixAlgebra(semfit, S)

#Matrix F of RAM approach
>semMatrixAlgebra(semfit, F)
```

Table 2.24: Matrix A of RAM model in R.

| | YSL | YIC | YAC | YWC | Mnl | TWY | JbL | NCW | Age | PSH | PrR | CY |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| YSL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| YIC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.26 |
| YAC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.42 |
| YWC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.26 |
| Mnl | 0 | 0 | 0 | 0 | 0 | 0.076 | 0.89 | 0 | 0 | 0 | 0 | 0 |
| TWY | 0 | 0 | 0 | 0 | 0 | 0 | 0.58 | 0.043 | 0.36 | 0 | 0 | 0 |
| JbL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.39 | 0 | 0 | 0.62 |
| NCW | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Age | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PSH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PrR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CY | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 2.25: Matrix S of RAM model in R.

| | YSL | YIC | YAC | YWC | Mnl | TWY | JbL | NCW | Age | PSH | PrR | CY |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| YSL | 0.57 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| YIC | 0 | 0.32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| YAC | 0 | 0 | 0.13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| YWC | 0 | 0 | 0 | 0.31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Mnl | 0 | 0 | 0 | 0 | 0.09 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TWY | 0 | 0 | 0 | 0 | 0 | 0.28 | 0 | 0 | 0 | 0 | 0 | 0 |
| JbL | 0 | 0 | 0 | 0 | 0 | 0 | 0.58 | 0 | 0 | 0 | 0 | 0 |
| NCW | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.99 | 0.29 | 0 | 0 | 0 |
| Age | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.29 | 0.99 | 0 | 0 | 0 |
| PSH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.99 | 0.77 | 0 |
| PrR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.77 | 0.99 | 0 |
| CY | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.42 |

### 2.15.5 Model Identification

In general, the identification is required because it tries to solve the problem of finding a unique estimate of the value of each parameter (Clark (n.d.)). An example is demonstrated

Table 2.26: Matrix F of RAM model in R.

| | YSL | YIC | YAC | YWC | Mnl | TWY | JbL | NCW | Age | PSH | PrR | CY |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| YSL | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| YIC | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| YAC | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| YWC | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Mnl | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TWY | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| JbL | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| NCW | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Age | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| PSH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| PrR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| CY | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

at Equation 2.77. The determination of a unique a and b solution is not feasible. This is because there are unlimited combinations of values of a and b which satisfy this equation. For example, a=0 and b=2, a=500 and b=-498 etc.

$$a + b = 2 \tag{2.77}$$

The next step after specifying the model is to actually identify it. In simple words, he must prove that the model's estimations are unique. A model's identification occurs before its estimation. By definition, an identified model is the one where the degrees of freedom is equal to or greater than 1. When degrees of freedom is equal to 0, then it said that the model is just-identified. In simple words, this means that all parameters are estimated. Such identification occurs when number of observations (in terms of variances) is equal to the number of parameters to estimate. In a just-identified model the model fit cannot be tested because there are no remaining degrees of freedom. A model is called under-identified when it has negative number of degrees of freedom. This is because more parameters are being estimated than the number of values in the covariance matrix. In such models, it is not feasible to find a unique estimate for each parameter. Aside from negative degrees of freedom, these models may also have problems with their structure. In order for a model to be viable it must be over-identified. An over-identified model specifies fewer paths or variable relations. When a model is over-identified, the parameters are available for estimation. Such models have positive degrees of freedom and plenty of information for the model is work with. Additionally, it allows for other model fit measures (Clark (n.d.)).

Hence, in order for a model to be viable and ready for estimation the number of known must be more than the number of the unknown information pieces. The number of unknown information pieces is equal to the number of parameters to estimate in the SEM model (variances, path coefficients, covariances, measurement errors). The degrees of freedom of the model is equal to the number of known pieces of information minus the number of unknown pieces of information.

In the example, according to the output, the known pieces of information are 66 because the number of observed variables in the model are 11. The model has 10 residual errors + 7 path coefficients + 3 weights (because the dotted line doesn't get estimated, it is used for scaling) + 1 covariance= 21 model parameters to be estimate. In simple words, there are 21 pieces of unknown information. The model has 42 degrees of freedom and is over-identified. Therefore, the model is viable and the model's parameter can be estimated.

The R code which demonstrates the over-identified model is below. The command is

called `sem`. The inputs are the specified SEM model, the correlation matrix and the number of observations. In this case, `sem_model`, `data.cor` and 1470, respectively. The output confirms that the model is over-identified with 42 degrees of freedom. The result is statistically significant with a Chi-square p-value of $0 < 0.05$.

```
  ##Model Identification ----
>semfit<-sem(sem_model,sample.cov=data.cor,sample.nobs=1470);semfit

###Output###
    lavaan 0.6-9 ended normally after 34 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        21

  Number of observations                          1470

Model Test User Model:

  Test statistic                               739.721
  Degrees of freedom                                42
  P-value (Chi-square)                           0.000
```

### 2.15.6 Model Estimation

After the model which is obtained from a large sample is specified and identified, the next step is to actually estimate the parameter of the model. After the model's parameter estimations. The goal of the estimation is to assign values for the free parameters of the model. Of course, as most of the model estimations, the procedure of estimating the population parameters is not done randomly. The estimation is done by minimizing the difference between the observed and the predicted variance-covariance matrix. The path models utilize matrix algebra to calculate the estimation of the parameters. Most of the matrix algebra consists of multiple constraints which must be taken into consideration. In case the constraints are not fulfilled, the statistical software display errors which are related with the matrices.

For the model estimation there are several methods available. It is up to the researcher's preference to decide which will be the estimation method. The estimation method then will estimate the parameters of the model and compare the observed and the hypothetical matrix. The most popular estimation methods out there are the following: Maximum Likelihood, Generalized Least Squares, Weighted Least Squares, Unweighted Least Squares, Ordinary Least Squares and the Full-information. The abbreviations for the methods above are ML (Maximum Likelihood), GLS (Generalized Least Squares), WLS (Weighted Least Squares), OLS (Ordinary Least Squares) and Raw ML, respectively. Both ML and GLS produce asymptotic unbiased parameter estimations and have similar properties. The only difference is that GLS has a least restrictive constraint of multidimensional normality which results in a value of $\chi^2$. As a result, the $\chi^2$ of the model fits better to the observed data. The WLS method calculates a weight matrix, based on the asymptotic variances and covariances of multi-space correlation upon estimating SEM models. The OLS method utilizes the sum of squared residuals and the magnitude of the difference between the observed and the hypothesized variance-covariance matrix. Finally, the raw ML is an asymptotic effective method of estimating models simultaneously with normally distributed errors. In the end, the most frequently used method to estimate SEM models is Maximum Likelihood. Most

statistical programs set ML as the default estimation method as it is more robust, effective and unbiased to various cases of models.

#### 2.15.6.1 Maximum Likelihood

The ML method is widely used for the estimation of multiple SEM models. It is a method which is "familiar" with instruments and hence can estimate non-recursive causal relations in path models. In simple words, the comparative advantage of ML is the effectiveness of the method to estimate models with latent variables. The name of the method, maximum likelihood, represents the principle which is behind the estimates of the parameters. The principle is the following: the estimates are the ones that maximize the likelihood that the observed variance-covariances were drawn from this population. Additionally, when estimating with ML the assumption of multivariate normality for the joint population distribution of the endogenous variables, given the exogenous variables is made. This is why the maximum likelihood is categorized as a normal theory method. Another strong advantage of ML is that is suitable in cases where the data are not normally distributed or when the sample size is small. The mechanism behind maximum likelihood is a method which utilizes recurrence. More specifically, ML takes an initial value which is repeatedly replaced for a better value. The process ends when the best possible values are found. In that case, it said that the model converges (Kapsali (2020)).

It is time now for the example SEM model to be estimated. This will be done with the ML method. In R, the command which allows the user to observe the model's estimation and more information is the `summary` of the model fit. Secondary inputs will be used to obtain more information about the model such as the $R^2$ values, the fit measures and the standardized solutions. The lines of code and its estimation output is displayed below:

```
#Model fit-identification
>semfit<-sem(sem_model,sample.cov=data.cor,sample.nobs=1470)

#Summary of the model fit-estimation
>summary(semfit,standardized=TRUE,rsquare=TRUE,fit.measures=TRUE)

Output:
lavaan 0.6-9 ended normally after 34 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        21

  Number of observations                          1470

Model Test User Model:

  Test statistic                               739.721
  Degrees of freedom                                42
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                             11623.624
  Degrees of freedom                                54
```

```
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.940
  Tucker-Lewis Index (TLI)                       0.922

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -13326.104
  Loglikelihood unrestricted model (H1)     -12956.243

  Akaike (AIC)                               26694.207
  Bayesian (BIC)                             26805.361
  Sample-size adjusted Bayesian (BIC)        26738.650

Root Mean Square Error of Approximation:

  RMSEA                                          0.106
  90 Percent confidence interval - lower         0.100
  90 Percent confidence interval - upper         0.113
  P-value RMSEA <= 0.05                          0.000

Standardized Root Mean Square Residual:

  SRMR                                           0.109

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  CY =~
    YrsSncLstPrmtn    1.000                               0.654    0.654
    YearsInCrrntRl    1.261    0.046   27.147    0.000    0.825    0.825
    YearsAtCompany    1.424    0.049   29.266    0.000    0.931    0.932
    YersWthCrrMngr    1.265    0.046   27.221    0.000    0.827    0.828

Regressions:
                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  MonthlyIncome ~
    JobLevel            0.891    0.013   69.940    0.000    0.891    0.888
    TotalWorkngYrs      0.076    0.013    6.007    0.000    0.076    0.076
  TotalWorkingYears ~
    JobLevel            0.589    0.016   37.090    0.000    0.589    0.586
    NumCompansWrkd      0.044    0.014    3.025    0.002    0.044    0.046
  JobLevel ~
```

```
    CY                0.628    0.037   16.763    0.000    0.411    0.431
  TotalWorkingYears ~
    Age               0.367    0.016   23.320    0.000    0.367    0.383
  JobLevel ~
    Age               0.395    0.020   19.579    0.000    0.395    0.414

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
PercentSalaryHike ~~
PerformancRtng      0.773    0.033   23.459    0.000    0.773    0.774

Variances:
                  Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .YrsSncLstPrmtn    0.571    0.023   25.319    0.000    0.571    0.572
   .YearsInCrrntRl    0.319    0.015   21.446    0.000    0.319    0.320
   .YearsAtCompany    0.132    0.012   11.229    0.000    0.132    0.132
   .YersWthCrrMngr    0.315    0.015   21.297    0.000    0.315    0.315
   .MonthlyIncome     0.095    0.003   27.111    0.000    0.095    0.104
   .TotalWorkngYrs    0.279    0.010   27.111    0.000    0.279    0.304
   .JobLevel          0.584    0.022   26.441    0.000    0.584    0.643
    PercentSalryHk    0.999    0.037   27.111    0.000    0.999    1.000
    PerformancRtng    0.999    0.037   27.111    0.000    0.999    1.000
    CY                0.428    0.031   13.673    0.000    1.000    1.000

R-Square:
                  Estimate
    YrsSncLstPrmtn    0.428
    YearsInCrrntRl    0.680
    YearsAtCompany    0.868
    YersWthCrrMngr    0.685
    MonthlyIncome     0.896
    TotalWorkngYrs    0.696
    JobLevel          0.357
```

In the top of the output the estimation method is displayed, which is of course ML. Then, the number of model parameters are displayed which is equal to 21. As a reminder, the number of model parameters section are the total number of free parameters of the model to estimate including residual variances, covariances, weights and path coefficients. Be careful, because the weight of the latent variable denoted with the dotted line is excluded from the parameter counting. Additionally, the number of observations is given, which in this case are 1470, the model's degrees of freedom which are 42 and the Chi-Square value which is equal to 739.721 (p-value$< 0.05$). The rest of the output up before the Parameter Estimates refers to model fit indices and will be analyzed later.

The main section of the output which is of great interest is the Parameter Estimates. The results for the measurement model which consists of the latent variable of the model, CY, are demonstrated in the Latent Variables section. In the first column, estimate, are the values of the parameters of the non-standardized path coefficients. Notice that the parameter value of YearsSinceLastPromotion is set to 1 and hence is not estimated. This is the reason why there is no output in the corresponding std errors, z-value and P($> |z|$) columns. In the std errors and z-value columns are the values of the standard errors and z-value, respectively. Recall, that z-value formula is the parameter estimate divided by its

standard error. The P($> |z|$) is the column which displays the statistical significance for each of the parameter estimates through p-value. Notice every single one of the p-values is below 0.05 and therefore are statistically significant. The two last columns contain the standardized values of the parameters. The std.lv column consists of the standard values of the latent variables only. The std.all column consists of the standard values of both observed and latent variables. Notice that the estimates are all positive and significant for the latent variable since their standardized value is above 0.7. The results are confirmed by the theory. After the results for the latent variables, the ones for the regressions and the residual variances take place. The columns are similar in every section of the parameter output.

In the Regressions section, the regression of the model are displayed. In the first regression, MonthlyIncome is regressed on JobLevel and TotalWorkingYears. The JobLevel positively and strongly causes MonthlyIncome with a standardized value of 0.888. It makes sense theoretically, because as the quality of the work of an employee increases, his salary is expected to increase as well. On the other hand, MonthlyIncome is weakly caused by TotalWorkingYears with a standardized value of 0.076. Thus, it seems like a high number of total working years doesn't necessarily increase the salary of an employee. In the second regression, TotalWorkingYears is regressed on JobLevel and NumCompaniesWorked. The JobLevel positively and strongly causes TotalWorkingYears with a standardized value of 0.586. Thus, an individual that is working many years is more likely to be developing high level skills at his job. The NumCompaniesWorked variable weakly causes TotalWorkingYears with a standardized value of 0.046. Therefore, it looks that an individual who has worked in many different companies doesn't necessary work many years. As a final endogenous effect, the JobLevel is regressed on the latent variable, CY. It is found that years at company in multiple posts strongly and positively causes JobLevel with a standardized value of 0.431. Therefore, increases in the company's years, increase the productivity and effectiveness of an employee's job. The exogenous effect of Age on TotalWorkingYears is decent and positive with a standardized value of 0.383. This is justified theoretically, since as an individual gets older, he works more and hence has more working years overall.

The next section of the parameters output is called Covariances. In this section, are the variables which the researcher is not sure for the direction of the causality. In this example, PercentSalaryHike and PerformanceRating yield a covariance value of 0.774. In other words, the two variables are highly associated without the knowledge of direction. Thus, the results show that the percentage of salary increase is highly associated with the yielded performance score of an employee.

The next section of the output of the parameters is called Variances and represents the residual variances of each of the endogenous variables of the model. Notice that Age, PercentSalaryHike, PerformanceRating and CY which are exogenous (only have arrows coming out of them) have an estimate of 1. This makes sense, because they are not explained by any other component of the model and hence their residual variance is equal to 1. Besides that, YearsSinceLastPromotion looks like it is explained decently by the model with a standardized value of 0.571. YearsInCurrentRole are even better explained by the model with a standardized residual variance value of 0.320. YearsAtCompany is almost entirely explained by the model since its standardized residual variance is equal to 0.132. The same is true for YearsWithCurrentManager which has a value of 0.315. These 4 variables are components of the latent variable, CY. Therefore, it seems that the latent variable successfully explains most of the variance of those variables. MonthlyIncome is almost entirely explained by the model since it has a value standardized residual variance of 0.104. TotalWorkingYears are decently explained by the model with a value of 0.304. Finally, JobLevel has a standardized residual variance value of 0.643.

The last section called R-Square contains the $R^2$ values for each endogenous variables of the model. In other words, exogenous variables have a $R^2$ value of 0. The values of $R^2$ represent the percentage of variance which is accounted from every endogenous variable. Notice that all the endogenous have a decent $R^2$ value. YearsSinceLastPromotion and JobLevel have the lowest values (0.428 and 0.357). Notice that, the total variance of a variable is equal to the residual variance + $R^2$ value. For example, for JobLevel, $0.643 + 0.357 = 1$.

### 2.15.7 Model Evaluation

There are several model fit indices for model evaluation of a specified and identified model in SEM. In order for R to display the model fit indices the command is `fitmeasures` which takes the model fit as the main input. As secondary input, the model fit indices already analyzed will be specified. The output of the specific example is demonstrated in Table 2.27. The corresponding code is demonstrated below.

```
#For model evaluation
>model.eval<-fitmeasures(semfit,c('chisq','rmsea','gfi','agfi','rmr','nfi',
'tli','cfi','pgfi','pnfi'))
model.eval
```

Table 2.27: The most popular fit indices which are used in model evaluation alongside with their acceptable values.

| Model fit index | Result | Acceptable conditions |
|---|---|---|
| $\chi^2$ | p-value=0.00 | $< 0.05$ |
| RMSEA | 0.106 | $< 0.08$ |
| GFI | 0.921 | $> 0.90$ |
| AGFI | 0.875 | $> 0.90$ |
| RMR | 0.109 | $< 0.05$ |
| NFI | 0.936 | $\geq 0.95$ |
| TLI | 0.922 | $\geq 0.95$ |
| CFI | 0.940 | $\geq 0.95$ |
| PGFI | 0.586 | $> 0.50$ |
| PNFI | 0.728 | $> 0.50$ |

According to the results displayed on Table 2.27, Chi-Squared which is considered as the main model fit index is statistically significant with a p-value of 0.00 and a value of 739.721. The rest of the model fit indices which are GFI, AGFI, RMSEA and RMR are equal to 0.921, 0.875, 0.106 and 0.109. Hence, only GFI fulfills its acceptable condition because $0.921 > 0.9$, RMSEA and AGFI are relatively close to their acceptable condition with $0.106 > 0.08$ and $0.875 < 0.90$, respectively. RMR has a value of 0.109 which is more than double of its acceptable condition ($< 0.05$). The model comparison indices which are TLI, NFI and CFI are equal to 0.922, 0.936 and 0.940, respectively. Although, none of those fit indices fulfill their acceptable condition ($\geq 0.95$), despite their values being extremely close to it. Finally, model parsimony fit indices which are PGFI and PNFI, both fulfill their acceptable condition ($> 0.50$) because they are equal to 0.586 and 0.728, respectively. Overall, the model is average fitted. Therefore, the model will be modified in the next section to yield better results.

#### 2.15.7.1 Validity and Reliability

Validity refers to the ability of a statistical entity, for example of a questionnaire, to correctly measure its variables' measurements. Validity mainly refers to the measurement model of the SEM model. There are three types of validity which are required for every measurement model. The first is the convergent validity. Convergent validity refers to the statistical significance of all the components of a measurement model. The convergent validity is conducted through the Average Variance Extracted index, also known as AVE. The second one is called construct validity. Construct validity is achieved when the model fit indices score well. The third one is called discriminant validity. Discriminant validity is achieved when the measurement model contains no unnecessary components. The rule of thumb for discriminant validity is the following: each pair of latent exogenous structure variables to be smaller than 0,85. It is important to notice, that in order for a model to be completely valid, the three validities above must be examined jointly not separately.

Reliability refers to how reliable is the measurement model in measuring its components. Reliability is a critical requirement for every latent variable. There are three types of reliability for a measurement model. The first is called internal reliability, and is achieved when the Cronback coefficient α, also known as Cronback's α, is larger than 0.7. This coefficient can take values from -1 to 1. If this requirement is not fulfilled, the researcher must identify and remove the problematic components to increase the Cronback's α. The formula of Conback's α is demonstrated in Equation 2.78. Assume $X_i$ is the observed score of an item $i$, then $X = (X_1 + X_2 + \cdots + X_k)$ is the sum of all items in a the test, where $k$ is the number of items in a latent variable. Finally, $\sigma_i^2$ is the variance of $X_i$ and $\sigma_X^2$ is the variance of $X$ which consists of item variances and inter-item covariances (https://bit.ly/2V7SDNf).

$$\boxed{\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k}\sigma_i^2}{\sigma_X^2}\right)} \tag{2.78}$$

The second is called construct reliability. Construct reliability is the measure which represent the internal cohesion between the observed variables of a latent variable. To achieve construct reliability, the Composite Reliability index, also known as CR, must have a values larger than 0.7. This index is calculated through the weights of the components of a latent variable. The formula of CR is demonstrated in Equation 2.79, where $\lambda_i$ is the standardized loading of the $i_{th}$ indicator, $\sigma_{e_i}^2$ is the variance of the error term of the $i_{th}$ indicator and $k$ is the number of indicators.

$$\boxed{\text{CR} = \frac{(\sum_{i=1}^{k})^2}{(\sum_{i=1}^{k})^2 + \sum_{i}^{k}\sigma_{e_i}^2}} \tag{2.79}$$

The type of reliability is for AVE which represents the average percentage of covariance explained by the items of a structure variable to be larger than 0.5. The formula for the calculation of AVE is demonstrated in Equation 2.80, where $k$ is the number of items, $\lambda_i$ is the factor loading of an item $i$ and $\sigma_{e_i}^2$ is the variance of the error term $i$ (https://bit.ly/3x8kZnG).

$$\boxed{\text{AVE} = \frac{\sum_{i=1}^{k}\lambda_i^2}{\sum_{i=1}^{k}\lambda_i^2 + \sum_{i=1}^{k}\sigma_{e_i}^2}} \tag{2.80}$$

It is important for results of the validity and reliability to be examined jointly. As already mentioned every observed variable corresponds to a standardized parameter value. In the

optimal scenario, every standardized parameter value must be larger than 0.6. During the evaluation of the model if an observed variable does not cross the threshold, then its up to the researcher's judgement to remove it. Often lower than the optimum standardized values are retained in the final model for its better cohesiveness.

Now it is time to demonstrate the validity and the reliability tests in the example SEM model. The specification, identification and estimation of the measurement model is done by using the following lines of R code. In the first lines of code the latent variable is specified, then it is identified and finally estimated.

```
Measurement model Specification
>SEMfactor<-'
#Measurement model
    ##Company Years
    CY =~ YearsSinceLastPromotion  + YearsInCurrentRole
    + YearsAtCompany + YearsWithCurrManager
'
#The identification of the latent variable
>semfactor<-sem(SEMfactor,sample.cov=data.cor,sample.nobs=1470);semfactor

#The estimation of the latent variable
>summary(semfactor,standardized=TRUE,rsquare=TRUE,fit.measures=TRUE)
```

The estimation output of the measurement model is displayed below. Notice that the model is over-identified with 2 degrees of freedom. Additionally the model's CFI and TLI are 0.994 and 0.982, respectively (both above 0.95) and RMSEA is equal to 0.085 (very close to $< 0.08$). Hence, the model seems to fit well.

```
    lavaan 0.6-9 ended normally after 18 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         8

  Number of observations                          1470

Model Test User Model:

  Test statistic                                23.305
  Degrees of freedom                                 2
  P-value (Chi-square)                           0.000

Model Test Baseline Model:

  Test statistic                              3474.113
  Degrees of freedom                                 6
  P-value                                        0.000

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.994
  Tucker-Lewis Index (TLI)                       0.982
```

```
Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -6615.954
  Loglikelihood unrestricted model (H1)      -6604.301

  Akaike (AIC)                               13247.908
  Bayesian (BIC)                             13290.252
  Sample-size adjusted Bayesian (BIC)        13264.838

Root Mean Square Error of Approximation:

  RMSEA                                          0.085
  90 Percent confidence interval - lower         0.056
  90 Percent confidence interval - upper         0.118
  P-value RMSEA <= 0.05                          0.024

Standardized Root Mean Square Residual:

  SRMR                                           0.014

Parameter Estimates:

  Standard errors                             Standard
  Information                                 Expected
  Information saturated (h1) model          Structured

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  CY =~
    YrsSncLstPrmtn    1.000                               0.654    0.655
    YearsInCrrntRl    1.275    0.047   27.223    0.000    0.835    0.835
    YearsAtCompany    1.402    0.049   28.838    0.000    0.918    0.918
    YersWthCrrMngr    1.280    0.047   27.291    0.000    0.837    0.838

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .YrsSncLstPrmtn    0.571    0.023   25.152    0.000    0.571    0.572
   .YearsInCrrntRl    0.303    0.015   20.526    0.000    0.303    0.303
   .YearsAtCompany    0.157    0.012   12.595    0.000    0.157    0.157
   .YersWthCrrMngr    0.298    0.015   20.356    0.000    0.298    0.298
    CY                0.428    0.031   13.631    0.000    1.000    1.000

R-Square:
                   Estimate
    YrsSncLstPrmtn    0.428
    YearsInCrrntRl    0.697
    YearsAtCompany    0.843
    YersWthCrrMngr    0.702
```

According to the estimation output of the measurement model which is displayed above, the model is over-identified with 2 degrees of freedom. The convergent validity is fulfilled in the model because all the parameters values have a p-value of 0 as displayed in the $P(>|z|)$ column. The construct validity refers to the model fit indices to have the appropriate values. In order for the model fit indices to be displayed again the `fitmeasures` command will be used. The lines of code are demonstrated below. In the object `construct.val` contains the model's fit indices and the results are displayed in Table 2.28. According to Table 2.28, almost all the model fit indices are fulfilling their accepting conditions except parsimony fit indices (PGFI and PNFI). But, overall the model fits very well. The Discriminant validity is meaningless since there is only 1 factor in the measurement model.

```
#Fit indices of the measurement model
>construct.val<-fitmeasures(semfactor,c('chisq','rmsea','gfi','agfi','rmr',
'nfi','tli','cfi','pgfi','pnfi'))
>construct.val
```

Table 2.28: The fit indices of the measurement model of the final SEM model.

| Model fit index | Result | Acceptable conditions |
|---|---|---|
| $\chi^2$ | p-value=0.00 | $< 0.05$ |
| RMSEA | 0.085 | $< 0.08$ |
| GFI | 0.992 | $> 0.90$ |
| AGFI | 0.961 | $> 0.90$ |
| RMR | 0.014 | $< 0.05$ |
| NFI | 0.993 | $\geq 0.95$ |
| TLI | 0.982 | $\geq 0.95$ |
| CFI | 0.994 | $\geq 0.95$ |
| PGFI | 0.198 | $> 0.50$ |
| PNFI | 0.331 | $> 0.50$ |

Moving on the the types of reliability. The interval reliability requires for the Cronbach's alpha to be above 0.7, while the AVE must be above 0.5. In order for those two reliability indices to be displayed the command `reliability` and the fit of the measurement model is used as an input. The code and the output are displayed below. The first row corresponds to the Cronbach's alpha and is above 0.7 (0.88) and the last row corresponds to the AVE which is larger than 0.5 (0.66).

```
#Interval and AVE reliability
>reliability(semfactor)

#Output:
           CY
alpha  0.8828083
avevar 0.6674060
```

The last type of reliability is the construct reliability which is achieved when the CR index is above 0.7. The computation of the Composite Reliability index requires a few steps. First, the standardized solutions of the measurement model are obtained. Second, the standardized estimation of the measurement model are saved into an object. Third, the

residual variances are computed with the following formula: $1-SL^2$. Finally, the Composite Reliability index is obtained in the output. According to the output $CR = 0.8878275$ which is larger than 0.7.

```
        ###Construct reliability - Composite reliability
>SL<-standardizedSolution(semfactor)
>SL <- SL$est.std[SL$op == "=~"]
>SL

#Residual variances
>RE<-1 - SL^2

#CR
>CR<-sum(SL)^2 / (sum(SL)^2 + sum(RE));CR

#Output:
0.8878275
```

### 2.15.8 Model Modification

In SEM analysis is all about coming up with a theoretically correct model, collect the sample necessary and check if the data fit the model. However, when the hypotheses are tested a model may not fit the data. The last step in SEM is called model modification and occurs when the researcher judges that the models doesn't fit the data. The researcher must correctly modify the model in order to yield a better fit in the data. The researchers is consulted by residual values, modification indices and the estimation of the model (stat insignificant results, low loadings) for the necessary adjustments. It is of big importance for the changed to be supported both logically and theoretically (Kline (2015)). The addition of covariances is more preferable than the path deletion. The Modification Indices, also known as MI, categorize specific adjustments that can be made in descending order (from the best to the worst modification). In general, the most common modification is to add an error covariance term between the observed variables of a latent variable. Most of the times this one adjustment is enough to better fit the model. The addition of error covariances must come along with the proper justification. Some of them are same measurement scale or similar instrumentation (Schumacker and Lomax (2016)). Last but not least, there is the possibility for the model to simply not fit to the specific sample. In this case, another random sample may work better for the construction of the SEM model.

#### 2.15.8.1 Modification Indices

Every SEM statistical software provides modification indices. A modification index shows the magnitude of the reduction of the overall model-fit Chi-Square with the addition of a path or covariance in the SEM model. Additionally, a MI is calculated for every possible relationship which is not included in the current model. Based on literature, modification indices that are smaller than 4 are considered irrelevant and should not be taken into consideration (Hair et al. (2013)). Usually, the largest MI is selected and added to the model. Then, the analysis is repeated and the significance of the adjustment is evaluated for the SEM model. Adjusting the model by relying exclusively on modification indices is not recommended. Researchers should look at residual diagnostics for an adjustment suggested by a modification index and then act accordingly, if justified by theory. The place were the researcher should look

first are the error covariance terms. The reason behind this move is that they might be related with observed variable relations. The next thing which might be concerning is low factor loadings on the latent variables. Lastly, be careful because changing the paths of the model, changes the core theoretical basis of the hypothesized model (Schumacker and Lomax (2016)). According to Schumacker and Lomax (2016), the following suggestions are recommended in order for a researcher to publish his findings:

1. Researcher should review the literate in which his theoretical model is based.
2. Researcher should provide the statistical software which he used and the version.
3. Researcher should report the type of SEM analysis
4. Researcher should include the correlation matrix, the sample size, the means and the standard deviation of the variance in his publication.
5. Researcher should include the diagram of the theoretical model.
6. Researcher should report the interpretation of the results and give further insights regarding the model fit, validation and reliability indices.

For the example, to inspect the modification indices in R the `modificationindices` command is used with the model fit as input. The secondary input sort. =T is used to provide a more summarized output. Thus, the following lines of code provide the researcher with the modification indices. As already discussed previously, the researcher is only interested in the modifications with where $mi > 4$. Thus, the `subset` command will be used to isolate those modifications with $mi > 4$.

```
#Inspecting the modification indices to decide what to modify
>model_mod<-modificationindices(semfit,sort. = T)

#Isolating only modification with mi>4
>subset(model_mod,mi>4)

Output:
                       lhs op                  rhs       mi
56           MonthlyIncome ~~   PercentSalaryHike 5383.784
44          YearsAtCompany ~~       MonthlyIncome 1994.971
57           MonthlyIncome ~~   PerformanceRating 1960.369
64           MonthlyIncome  ~                  CY 1113.130
24      NumCompaniesWorked ~~                 Age  559.098
75      NumCompaniesWorked  ~                 Age  497.059
62               JobLevel ~~   PerformanceRating  239.513
27                     CY =~   PerformanceRating  207.534
45          YearsAtCompany ~~   TotalWorkingYears  136.761
23      NumCompaniesWorked ~~  NumCompaniesWorked   71.558
51    YearsWithCurrManager ~~            JobLevel   70.306
40      YearsInCurrentRole ~~            JobLevel   57.296
37      YearsInCurrentRole ~~ YearsWithCurrManager   48.885
60       TotalWorkingYears ~~   PerformanceRating   46.021
38      YearsInCurrentRole ~~       MonthlyIncome   38.000
49    YearsWithCurrManager ~~       MonthlyIncome   27.912
63           MonthlyIncome  ~  NumCompaniesWorked   22.850
```

```
36      YearsInCurrentRole ~~       YearsAtCompany   19.188
48          YearsAtCompany ~~    PerformanceRating   16.748
65           MonthlyIncome  ~                  Age   12.757
30 YearsSinceLastPromotion ~~ YearsWithCurrManager   11.422
31 YearsSinceLastPromotion ~~        MonthlyIncome   10.763
47          YearsAtCompany ~~    PercentSalaryHike    9.525
52    YearsWithCurrManager ~~    PercentSalaryHike    8.609
29 YearsSinceLastPromotion ~~       YearsAtCompany    5.751
61                JobLevel ~~    PercentSalaryHike    4.472
```

In the output, the lhs column stands for left hand side or outcome, rhs stands for right hand side or predictor and op stands for operation and defines the kind of modification that has to be made. When op is equal to $\sim\sim$, a covariance is suggested, when op is equal to $\sim$ a regression is suggested and when op is equal to $=\sim$, the right variable acts like an indicator for the latent variable in the left. Recall that, the model yielded decent results regarding model fit, validity and reliability indices. Additionally, the modifications suggested by the statistical program do not stand theoretically. In conclusion, all those results point out to the fact that the original model will be retained.

## 2.16 Discussion

In this section, i want to briefly render my personal view of the challenges, prerequisites and difficulty of SEM modeling. From the experienced gained from the SEM technique, i realized that SEM is not easy and obvious to conduct as most of the beginners believe. From model specification to model modification, the construction of a valid SEM model from scratch is something that depends on multiple factors. First of all, the researcher must be knowledgeable on both the basic concepts of statistics and SEM to even consider to apply this advanced technique. SEM is more complicated than simply assigning direction of causality to strongly correlated observed and latent structures and variables. In every test of a SEM model, the researcher must examine the changes on multiple measures to determine if the current model is better than the previous ones. Some of these measures include the fit indices, statistical significance of the estimates, the strength of the path coefficients and loadings, the residual error variances and the coefficients of determination of the endogenous observed variables. Based on my personal experience, apart from the statistical knowledge, SEM requires for the researcher to have deep knowledge about the structure of the dataset and the interpretation of the variables. Simply choosing highly correlated variables to specify and graph a SEM model might not be enough to yield statistically significant results and good fit to the data. Regardless of the correlation strength, the relation between variables which theoretically make sense to be associated in some sort of a way should always be tested during the specification of the SEM model. Thus, i have noticed that very useful skills necessary for every researcher are intuition and imagination. The researcher must explore his options, judge which relations make sense to exist, test those relations and construct the SEM model from there. Finally, i highly recommend for the researchers to not blindly consult the modification indices during the model modification phase. Recall, that modification indices are computed through mathematical formulas which aim to maximize the result of the Chi-Square ($\chi^2$) test. They are simply mathematical formulas which recommend paths based on $\chi^2$ and for that reason the researcher must use his own experience, knowledge and judgement to add and remove paths from his model. I wanted to share these personal insights on SEM modeling to try to make you understand the complexity of the specific technique and think outside-of-the-box.

## 2.17 Conclusions

In this chapter, the fundamental statistical techniques of unsupervised learning were explicitly analyzed. As a reminder, unsupervised learning is related with $i$ observations which correspond to the observation of a vector of measurement $x_i$ but without a particular response $y_i$. The term unsupervised refers to the absence of a response variable that can supervise the researcher's analysis. On the other hand, supervised learning involves problems in which each observation of the input, $x_i$, corresponds to an output value, $y_i$. At the start of the second chapter, a definition of SEM is given. Structural Equation Modeling refers to a growing family of related procedures which demonstrate relations between observed and latent variables testing hypotheses made by the researcher. After this section, a brief history of structural equation modeling is given. The main topics of discussion regarding the SEM history involves regression model, path model, exploratory and confirmatory factor analysis, and, finally, SEM. In the end of the first part of the second chapter, R is introduced as the main statistical software since it will be used constantly throughout this thesis. More specifically, the 1) advantages, 2) types of R objects, 3) assignment operators, 4) mathematical operators, 5) relational operators and 6) logical operators are explicitly analyzed in this section.

At the start of the second part of the second chapter, the formulas of correlation and covariance are explicitly explained and demonstrated through examples since they are essential statistical concepts in the upcoming unsupervised statistical analyses. Consequently, next up two of the most famous unsupervised techniques, PCA and Cluster Analysis will be theoretically explained and applied through an example. Before the actual PCA application, the mathematical entities of eigenvectors and eigenvalues are introduced because they are essential blocks of PCA. Then, the basic concepts of PCA were introduced algebraically. After the algebraic approach, the theory behind the basic parts of a PCA analysis such as contribution of a case to a PC, squared cosine of a PC with a case and PCA loadings was explicitly analyzed. In the R lab, PCA was conducted with both correlation and covariance matrices. PCA was conducted with matrices and computer functions. The example dataset is called Company dataset and includes data of 1470 employees which give answers to demographic and company-related questions and the following 14 numerical variables are obtained and used for the analysis: $Age$, $DailyRate$, $DistanceFromHome$, $HourlyRate$, $MonthlyIncome$, $MonthlyRate$, $NumCompaniesWorked$, $PercentSalaryHike$, $TotalWorkingYears$, $TrainingTimesLastYear$, $YearsAtCompany$, $YearsInCurrentRole$, $YearsSinceLastPromotion$ and $YearsWithCurrManager$. Some of the most noticeable results of the PCA analysis are the following:

1. In case the correlation matrix was used as input, the variables $YearsWithCurrentManager$, $YearsInCurrentRole$, $YearsSinceLastPromotion$ and $YearsAtCompany$ load strongly, positively and have the same direction onto the first PC. In simple words, these variables tend to behave similarly. The variables $MonthlyIncome$, $TotalWorkingYears$ and $Age$ load positively and have the same direction while $NumCompaniesWorked$ barely loads onto the first PC.

2. In case the correlation matrix was used as input, the variables $YearsWithCurrentManager$, $YearsInCurrentRole$, $YearsSinceLastPromotion$ and $YearsAtCompany$ load positively and have the same direction onto the second PC. The variables $MonthlyIncome$, $TotalWorkingYears$, $Age$ and $NumCompaniesWorked$ load negatively and have the same direction load onto the second PC.

After PCA was conducted, the first and second principal component, which are the most important, are extracted and used as input for Cluster Analysis in R. But before that,

the basic concepts and ideas of cluster analysis are presented. Cluster analysis is mainly conducted thanks to the Silhouettes approach which uses the two principal components to define the number of clusters and k-means algorithm which assigns each observation to a cluster. The theory behind both of these concepts of cluster analysis are explicitly analyzed in this point of the chapter. After the theoretical part of cluster analysis was covered, an example with the first and second principal component of the PCA analysis of the Company dataset as input took place. Silhouette method indicated that the optimal number of clusters was 3. The main characteristics of the variables in each cluster pointed out by their correlation are the following:

1. The first cluster consists of 688 observations. The variables which scored a relatively high correlation index in this cluster are $TotalWorkingYears$, $YearsAtCompany$, $YearsInCurrentRole$, $YearsWithCurrManager$ and $MonthlyIncome$. These variables are positively correlated and therefore jointly increase or decrease. Additionally, $TotalWorkingYears$ and $Age$ have a correlation index of 0.42 which indicates that as one of these variables moves towards one direction the other does the same.

2. The second cluster consists of 393 observations. In this cluster, $Age$ is negatively correlated with $NumCompaniesWorked$ and positively correlated with $TotalWorkingYear s$ and $MonthlyIncome$ (stronger positive correlation than cluster 1). Additionally, $TotalWorkingYears$ is negatively correlated with $NumCompaniesWorked$ and positively correlated with $MonthlyIncome$ and $Age$. Finally, $YearsAtCompany$, $YearsIn CurrentRole$ and $YearsWithCurrManager$ are positively correlated, but these results are similar to the ones obtained from analyzing cluster 1. This is because cluster 2 shares a reasonable amount of common observations with cluster 1.

3. The second cluster consists of 383 observations. According to the characteristics of this cluster, $MonthlyIncome$, $Age$, $TotalWorkingYears$ and $YearsAtCompany$ are positively correlated.

Finally, the information gathered from the results of PCA and Cluster Analysis were used to specify the SEM model. At the start of the last core part of this chapter, the theory behind basic concepts of SEM such as 1) path analysis, 2) endogeneity and exogeneity, 3) observed and latent variables and 4) graph theory are explicitly analyzed. After the core parts of SEM are analyzed, an lab example in R is presented. The example involves the Company dataset again. The specification of the SEM model of the Company dataset includes the following observed variables: $JobLevel$, $MonthlyIncome$, $TotalWorkingYears$, $Age$, $NumCompaniesWorked$, $PerformanceRating$, $PercentSalaryHike$, and a latent variable called Company Years which consists of $YearsInCurrentRole$, $YearsSinceLastPromotion$, $YearsAtCompany$ and $YearsWithCurrentManager$. The SEM model consists of 3 regressions: a) $MonthlyIncome \sim JobLevel + TotalWorkingYears$, b)$TotalWorkingYears \sim JobLevel+NumCompaniesWorked$ and c) $JobLevel \sim CY$, 2 exogenous effects: a)$TotalWo rkingYears \sim Age$ and b)$JobLevel \sim Age$ and 1 undirected association: $PercentSalaryHik e \sim\sim PerformanceRating$. This particular specification was created based on theory behind the variables and the consultation of modification indices. After the specification took place, the model was identified so that its parameters can be estimated. Then, the model fitness to the data was evaluated through the following indices: RMSEA, GFI, AGFI, RMR, NFI, TLI, CFI, PGFI, and PNFI. After the estimation of the parameters of the SEM model, the following results are displayed:

1. The joint effect of $YearsWithCurrentManager$, $YearsSinceLastPromotion$, $YearsA tCompany$ and $YearsInCurrentRole$ which is represented by the latent variable Company Years on $JobLevel$ is positive and relatively high with a path coefficient of 0.43.

Thus, it seems like as the years of an individual increase, the quality of his job gets better.

2. The effect of $JobLevel$ on $MonthlyIncome$ and $TotalWorkingYears$ is positive and pretty high with path coefficients of 0.89 and 0.59, respectively. Thus, it seems like as an individual gets better at his job, both his/her monthly income and working years of his/her lifetime increase.

3. The effect of $Age$ on $JobLevel$ and $TotalWorkingYears$ is positive and relatively high with path coefficients of 0.41 and 0.38, respectively. Thus, it seems like as an individual gets older, both the quality of his/her work and working years of his/her lifetime increase.

4. The effect of $NumCompaniesWorked$ on $TotalWorkingYears$ is positive and very low with a path coefficient of 0.05, respectively. Thus, it seems like as an individual works in many companies, his/her total working years increase.

5. The effect of $TotalWorkingYears$ on $MonthlyIncome$ is positive and very low with a path coefficient of 0.08, respectively. Thus, it seems like as the total working years of an individual increase, his/her monthly income increase as well.

6. The undirected association between $PerformanceRating$ and $PercentSalaryHike$ is positive and pretty high with a covariance term of 0.77. Thus, it seems like the rating of the performance of an individual at his/her work is positively and highly associated with his/her percentage of salary increase.

7. The unexplained variances, also known as residual variances, of the endogenous observed variables: $JobLevel$, $MonthlyIncome$, $TotalWorkingYears$, $YearsWithCurrentManager$, $YearsSinceLastPromotion$, $YearsAtCompany$ and $YearsInCurrentRole$ are equal to 0.64, 0.10, 0.30, 0.31, 0.57, 0.13 and 0.32, respectively. Thus, it seems like the explained variance of the variables $MonthlyIncome$, $TotalWorkingYears$, $YearsWithCurrentManager$, $YearsAtCompany$ and $YearsInCurrentRole$ is relatively high while $JobLevel$ and $YearsSinceLastPromotion$ are less explained by the current SEM model.

# Chapter 3

# Introduction to Bayesian Networks in Statistics

## 3.1 Connection between Bayesian Statistics and SEM

In this chapter, Bayesian Structural Equation Modeling will be introduced as an alternative to the classical SEM approaches. To understand the Bayesian SEM approach to its full extent, the demonstration of the application of Bayesian methodology to first generation SEM is essential. The result is a second generation Bayesian Structural Equation Modeling. Basic and advanced concepts of both first and second generation SEM will be demonstrated later. Bayesian methodology provides a coherent philosophical alternative to conventional SEM practice, regardless of whether models are first or second generation (Kaplan and Depaoli (2012)). In the sections following, an introduction to Bayesian ideas will be made, including, the Bayesian graph theory, Bayes' theorem, the nature of prior distributions, description of the posterior distribution and Bayesian model building. Then, examples of the application of Bayesian SEM will be demonstrated.

To begin with, history of SEM can be divided into the first and the second generation. The first SEM generation consists of topics such as confirmatory factor analysis and simultaneous equation modeling. Additionally, in the first SEM generation, the necessary remedies for handling nonstandard conditions of the data took place. Examples of such nonstandard problems are missing data, non-normal data and sample size. The second SEM generation consists of the merge of the first generation's models for continuous latent variables and the first generation's models for categorical latent variables. This merge happened due to the extension of finite mixture modeling to the SEM framework. These extensive SEM framework came along with elegant theory which allowed for critical applications. Examples of such applications are techniques for handling the evaluation of interventions with noncompliance (Jo and B. O. Muthén (2001)), discrete-time mixture survival models (B. Muthén and Masyn (2005)), and models for examining unique trajectories of growth in academic outcomes (Kaplan (2002)). At the same time of the development of the two SEM generations, the Bayesian methods were adapted for complex SEM models. B. Muthén and Asparouhov (2012) recently proved that Bayesian SEM shows signs of great flexibility. Complicated computer algorithms allowed for Bayesian logic to enter the SEM world.

## 3.2 The Fundamental Concept of Bayesian Statistics

In this section, the basic concepts of in Bayesian inference will be analyzed. Bayesian inference is a basic block into building the Bayesian SEM framework. Assume that $Y$ is

a random variable and takes realized value $y$. For example, $y$ could be the socioeconomic status of a person. In the SEM framework, $y$ could be vector-valued, such as items on an attitude survey. $Y$ suddenly becomes realized as $y$ when an individual answers to the survey. Therefore, $Y$ could be considering unobserved and represents the probability distribution of $Y$ that the researcher tries to find through the data values $y$. Additionally, assume that parameter $\theta$ is believed that represents the probability model. Parameter $\theta$ can take values of a scalar, such as the mean or the variance of a distribution, or a vector, such as the set of all SEM parameters. The main concept behind the Bayesian logic is to determine to probability of observing $y$ given unknown parameter $\theta$, denoted as $p(y|\theta)$. From a statistical point of view, the goal is to get estimates of the $\theta$ parameters given certain data. This is expressed as the likelihood of the parameters given the data, denoted as $L(\theta|y)$. In literature, it is common for the researcher to use the log-likelihood, denoted with $l(\theta|y)$ (Hoyle (2012)).

There is an important difference Bayesian and frequentist statistical inference. This is the nature of $\theta$. In the frequentist statistical inference, the $\theta$ is unknown but fixed. In Bayesian statistical inference, $\theta$ is random and represents a probability distribution of the uncertainty of the actual value of $\theta$. The joint probability of the parameters and the data can be modeled as a function of the conditional distribution of the data given the parameters, and the prior distribution of the parameters. This is because observed data $y$ and parameter $\theta$ are assumed to be random. Mathematically, this is represented in Equation 3.1:

$$\boxed{p(\theta, y) = p(y|\theta)p(\theta)} \tag{3.1}$$

But, according to the probability theory, the joint probabilities are symmetrical and hence:

$$\boxed{p(y|\theta)p(\theta) = p(\theta|y)p(y)} \tag{3.2}$$

and thus:

$$\boxed{p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)}} \tag{3.3}$$

In Equation 3.3, $p(\theta|y)$ is known as the posterior distribution of the $\theta$ parameters given $y$ data. According to Equation 3.3, $p(\theta|y)$ is equal to the data distribution, $p(y|\theta)$, multiplied by the prior distribution of the parameters $p(\theta)$. The result is divided by $p(y)$. Equation 3.3 is known as the Bayes' theorem. For discrete and continuous variables, the Bayes' theorem is demonstrated in Equation 3.4 and Equation 3.5, respectively.

$$\boxed{p(y) = \sum_{\theta} p(y|\theta)p(\theta)} \tag{3.4}$$

$$\boxed{p(y) = \int_{\theta} p(y|\theta)p(\theta)d\theta} \tag{3.5}$$

Notice that the denominator of Equation 3.3 does not contain any model parameters. Thus, the term can be omitted to get the non-normalized posterior distribution demonstrated in Equation 3.6.

$$\boxed{p(\theta|y) \propto p(y|\theta)p(\theta)} \tag{3.6}$$

In Equation 3.6, the term $p(y|\theta)$ can be written in the form of the unknown parameters $\theta$ for fixed values $y$. In that case, the term is the likelihood $L(\theta|y)$. Thus, Equation 3.6 can

be written as:

$$\boxed{p(\theta|y) \propto L(\theta|y)p(\theta)} \tag{3.7}$$

The essence of the Bayesian statistical inference and the difference from the frequentist is represented by Equation 3.6. In simple words, Equation 3.6 says that the uncertainty about the model's parameters expressed by the prior distribution $p(\theta)$, is is weighted by the observed data, $p(y|\theta)$, and results in an estimation of the parameters of the model represented by the posterior distribution, $p(\theta|y)$ (Hoyle (2012)).

## 3.3 Types of Priors

Bayesian inference is characterized by the prior distribution for the model parameters. The researcher is called to choose prior distributions for the parameters of a model. This task is considered difficult for the researcher. There are two types of priors: a) non-informative and b) informative priors. The difference between the two types is based on the amount of information that is considered to be prior to data collection and the degree of accuracy of the information.

### 3.3.1 Non-informative priors

There are cases where there are not enough information to draw posterior inferences. Considering the Bayesian approach, this lack of information is important to be considered into the model. In simple words, the quantification of the ignorance and cumulative understanding of a problem is of equal importance. The quantification of the ignorance is done by embodying a non-informative prior into the specification. The uniform distribution over some sensible range of values is the most popular non-informative prior distribution. However, the researcher must proceed to the step of the selection of the range of values over uniform distribution with caution. For example, a uniform $[-\infty, +\infty]$ would be invalid since it cannot be integrated to 1 as with any probability distributions. There is also the "Jeffrey's prior", which is suitable problems in uniform priors.

### 3.3.2 Informative priors

There are many cases where there are enough information about the distribution of a model parameter that it can be embodied into the prior distribution. These priors are called informative. There is one type of informative prior which is based in the so called "conjugate prior" distribution. When the "conjugate prior" distribution is combined with the likelihood function, yields a posterior distribution that is in the same distributional family as the prior distribution (Hoyle (2012)). A non-conjugate prior results in a complex posterior distribution. Nowadays, there are several methods for Bayesian statistics such as the Markov Chain Monte Carlo sampling which can efficiently deal with the problem of nonconjugacy.

## 3.4 Graph Theory and Notation in Bayesian Networks

Bayesian Networks are graphically represented by a $G = (V, A)$ space, where $V$ is a non-empty set of nodes or vertices and $A$ is a finite set of pairs of vertices, also called arcs or links (Nagarajan, Scutari, and Lèbre (2013)). Each $a = (u, v)$ arc is either an ordered or unordered pair of nodes connected by and incident on the arc and adjacent to each other. Because $u$ and $v$ are adjacent, they are also reasonably called neighbors . In case, $(u, v)$ is

an ordered pair, $u$ and $v$ are said to be the tail and head of the arc, respectively. In this case, the arc is said to be directed from $u$ to $v$ via an arrowhead towards $v$ ($u \rightarrow v$). In case, $(u, v)$ is an unordered pair, $u$ and $v$ are said to simply be associated on the arc without any further information. Such arcs are called undirected, denoted with $e \in E$ and displayed with a line without an arrowhead ($u - v$). According to the type of arcs, the graph adopt the corresponding name (directed or undirected graphs). When a graph consists of directed arcs only then, the graph is called directed and is denoted as $G = (V, A)$. Contrary, if a graph consists of undirected arcs only then, the graph is called undirected and is denoted as $G = (V, E)$. Finally, these is a last category of graphs called mixed graphs which consist of both directed and undirected graphs. Such graphs are denoted as $G = (V, A, E)$. In Figure 3.1, all three types of graphs are demonstrated.

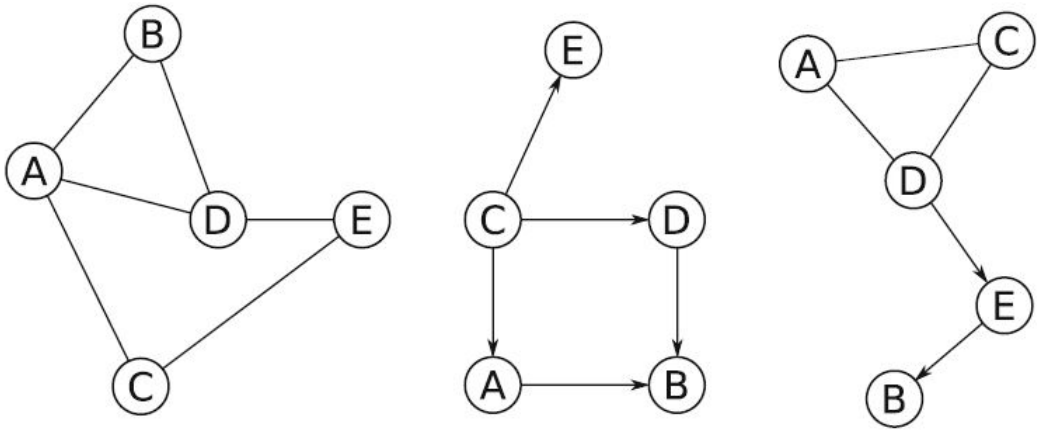



Figure 3.1: The undirected graph on the left, the directed graph on the center, and the mixed graph on the right, adopted from Nagarajan, Scutari, and Lèbre (2013, p. 2).

The notation and graph theory of Figure 3.1, reveals several interesting points.
For the undirected graph represented on the left of Figure 3.1:

- The set of nodes is $V = A, B, C, D, E$ and the set of edges is $E = \{(A - B), (A - C), (A - D), (B - D), (C - E), (D - E)\}$.
- The arcs are undirected. For example, $A - B$ and $B - A$ represent the same thing and define the same edge.
- $A$ and $B$ are adjacent because they are connected to each other.

  (Nagarajan, Scutari, and Lèbre (2013))

For the directed graph represented on the center of Figure 3.1:

- The set of nodes is $V = A, B, C, D, E$ and the set of arcs (not set of edges, since they have an arrowhead, a direction) is $A = \{(A \rightarrow B), (C \rightarrow A), (D \rightarrow B), (C \rightarrow D), (C \rightarrow E)\}$.
- The arcs are directed. For example, $(C \rightarrow A)$ and $(A \rightarrow C)$ define different arcs. Due to the acyclicity constraint derived from the graph theory, it is not feasible for both arcs to be graphed because one arc between each pair of nodes is the maximum that is allowed.
- $A$ and $B$ remain adjacent, since the arc $A \rightarrow B$ exists. In this example, $A$ is the tail and $B$ is the head. Additionally, it is said that there is an outgoing arc for $A$ and an

incoming arc for $B$.

(Nagarajan, Scutari, and Lèbre (2013))

For the mixed graph represented on the right of Figure 3.1:

- There is a mix of the set of edges $E = \{(A - C), (A - D), (C - D)\}$ and the set of arcs $A = \{(D \rightarrow E), (E \rightarrow B)\}$
- A directed and/or a mixed graph can construct an undirected graph. The researcher must simply replace all the directed arcs from the undirected ones. The resulted graph is called the skeleton or the underlying undirected graph of the initial graph.

  (Nagarajan, Scutari, and Lèbre (2013))

The arcs and the nodes are not placed randomly in the graph space. Instead, the positioning of the arcs and nodes is really important in the Bayesian graph theory and is known as the structure of the graph. Assuming that $u$ and $v$ vertices incident on each arc are distinct and that there is at most one arc between them so that $(u, v)$ uniquely identifies an arc. Of course, there are no loops that can occur when $u = v$ (Nagarajan, Scutari, and Lèbre (2013)).

A graph is said to be empty when it has no arcs. On the other hand, a graph is said to be saturated when all the nodes are connected with each other. In real-world applications, the Bayesian graph usually falls between the empty and saturated graph. Researcher often categorize the graphs into sparse or dense. Even though, the difference between the two is not clear, a graph is said to be sparse if $O(|E| + |A|) = O(|V)$.

There are several statistical properties of a Bayesian graph and most of them concern the paths. As already discussed in the SEM framework, a path is a sequence of arcs or edges connecting two nodes. The path are denoted with the sequence of vertices $(v_1, v_2, ..., v_n)$ incident on those arcs. Vertices $v_1, v_2, ..., v_n$ are connected via arcs. An assumption is made that such arcs are unique, which means that a path passes through each arc only once. In the graphs with direction, the assumption is made that all the arcs in a path have the same direction. Path where $v_1 = v_n$ are named cycles and are of big importance in the Bayesian network.

As already mentioned, the graph is the Bayesian framework is assumed to be acyclic. This property allows for a partial ordering of the nodes which is defined by the structure of the graph. This type of ordering is known as topological ordering and depends on the direction of the arcs. The definition of the topological ordering is the following: if a node $v_i$ precedes $v_j$, then an arc from $v_j$ to $v_i$ is not feasible. Thus, the first nodes are called root nodes and have no incoming arcs, while the last nodes are called leaf nodes and have one incoming arc at minimum and not a single outgoing one. Of course, $v_i$ precedes $v_j$ in the sequence of the ordered nodes when a path occurs from $v_i$ to $v_j$. Here, $v_i$ is also known as an ancestor of $v_j$, while $v_j$ is an descendant of $v_i$. Finally, it is said that $x_i$ is a parent of $v_j$ and $v_j$ is a child of $v_i$ when the corresponding path contains only one arc.

For example, take the $A$ node in the directed graph on the left picture of Figure 3.1. Notice how the adjacent nodes are either parents or children of $A$. This is the property of a node being neighbour to another. The whole neighborhood consists of the parents and the children. Additionally, notice that the parents of $A$ are also its ancestors because the topological ordering forces them to precede $A$. With the same logic, children of $A$ are also its descendants (Nagarajan, Scutari, and Lèbre (2013)).

The specific graph has the following topological ordering:

$$(\{F, G, H\}, \{C, B\}, \{A\}, \{D, E\}, \{L, K\}). \tag{3.8}$$

## 3.5 The Basic Definitions and Properties of Bayesian Networks

Graphical models in the Bayesian networks framework allow for synoptic representation of the probabilistic dependencies between a given set of random variables $X = X_1, X_2, ..., X_p$. An example of such graphical models is the directed acyclic graph $G = (V, A)$. Each random variable $X_i$ is assigned to a node $v_i \in V$.

### 3.5.1 Maps

The absence of a specific arc corresponds to a graphical separation, denoted as $\perp\!\!\!\perp_G$. The dependencies which occur between the variables are accurately represented by the probabilistic independence, denoted as $\perp\!\!\!\perp_P$. Pearl (2014) yielded a correspondence as an independency map.

The definition of an independency map (I-map) is the following: A graph $G$ is an independency map (I-map) of the probabilistic dependence structure $P$ of $X$ if there is a one-to-one correspondence between the random variables in $X$ and the nodes $V$ of $G$, such that for all disjoint subsets $A$, $B$, $C$ of

$$\boxed{A \perp\!\!\!\perp_P B|C \Longleftarrow A \perp\!\!\!\perp_G B|C} \tag{3.9}$$

Similarly, $G$ is a dependency map ($D - map$) of $P$ if $X$ for is true that

$$\boxed{A \perp\!\!\!\perp_P B|C \Longrightarrow A \perp\!\!\!\perp_G B|C} \tag{3.10}$$

$G$ is said to be a perfect map of $P$ if it is both a $D - map$ and an $I - map$. In this case, $P$ is said to be faithful to $G$. Thus that

$$\boxed{A \perp\!\!\!\perp_P B|C \Longleftrightarrow A \perp\!\!\!\perp_G B|C} \tag{3.11}$$

### 3.5.2 D-separation

The correspondence between the directed acyclic graph $G$ and the represented conditional independence relationships is clarified by the directed separation criterion (Pearl (2014)), also known as d-separation.

The definition of D-separation will be given next. Assume there are three disjoint subsets of nodes, namely $A$, $B$ and $C$ in the directed acyclic graph $G$ (Nagarajan, Scutari, and Lèbre (2013)). In this case, $C$ d-separates $A$ from $B$, and is denoted as $A \perp\!\!\!\perp_G B|C$ only if for every sequence of arcs between a node contained in $A$ and a node in $B$, there is another node $v$ which fulfills two conditions:

1. Adjacent nodes in the path has two arcs pointing to $v$ and $C$ does not contain none if $v$ or its descendants.

2. $C$ contains $v$ and does not have converging arcs.

D-separation is followed by the Markov property of Bayesian networks. The latter allows for the product of conditional probability distributions, also known as the local distributions associated with each variable $X_i$, which results in the joint probability distribution of the random variables in $X$, also known as global distribution. In case there are discrete random variables, the factorization of the joint probability distribution $P_X$ is displayed in Equation 3.12, where the parents of $X_i$ are denoted as $\Pi_{X_i}$.

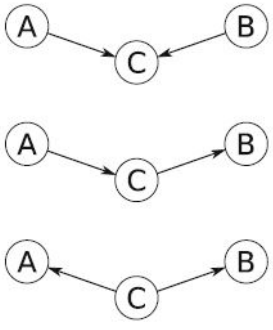
$$\boxed{P_X(X) = \prod_{i=1}^{p} P_{X_i}(X_i|\Pi_{X_i})} \tag{3.12}$$

In the case of continuous random variables:

$$\boxed{f_X(X) = \prod_{i=1}^{p} f_{X_i}(X_i|\Pi_{X_i})} \tag{3.13}$$

### 3.5.3 The Essential Connections

The essential connections as stated by Jensen and Nielsen (2007) are represented by the three possible combinations that one can do with two arcs and three nodes. There are 3 connections that one can do with these graphical objects: the convergent, serial and diverging connection. These connections are represented in Figure 3.2




$$\mathbf{A} \not\perp\!\!\!\perp_G \mathbf{B} \,|\, \mathbf{C}$$
$$\mathrm{P}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{P}(\mathbf{C} \,|\, \mathbf{A}, \mathbf{B})\, \mathrm{P}(\mathbf{A})\, \mathrm{P}(\mathbf{B})$$

$$\mathbf{A} \perp\!\!\!\perp_G \mathbf{B} \,|\, \mathbf{C} \Rightarrow \mathbf{A} \perp\!\!\!\perp_P \mathbf{B} \,|\, \mathbf{C}$$
$$\begin{aligned}\mathrm{P}(\mathbf{A}, \mathbf{B}, \mathbf{C}) &= \\ &= \mathrm{P}(\mathbf{B} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{C} \,|\, \mathbf{A})\, \mathrm{P}(\mathbf{A}) \\ &= \mathrm{P}(\mathbf{A} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{B} \,|\, \mathbf{C})\, \mathrm{P}(\mathbf{C})\end{aligned}$$

Figure 3.2: The three basic connections: convergent, serial and diverging from top to bottom. In the right side, the graphical separation, conditional independence and probability decomposition of the connections is presented, adopted from Nagarajan, Scutari, and Lèbre (2013, p. 15).

The convergent connection is also called v-structure because of the shape of the graph. As displayed in Figure 3.2, in the specific connection outgoing arcs from nodes $A$ and $B$ are received as incoming arcs for node $C$. Under these circumstances, the convergent connection violates the conditions of the d-separation (Nagarajan, Scutari, and Lèbre (2013)). Thus, it is fair to say that $A$ and $B$ are not d-separated by $C$. Consequently, $A$ and $B$ are not independent given $C$. Finally notice that $\Pi_A = \{\oslash\}$, $\Pi_B = \{\oslash\}$ and $\Pi_C = \{A, B\}$ and therefore:

$$\boxed{P(A, B, C) = P(C|A, B)P(A)P(B)} \tag{3.14}$$

From Equation 3.12 which introduces the Markov property, Equation 3.14 supports that $C$ depends on the joint distributions of $A$ and $B$. Thus, $A$ and $B$ are not conditionally independent given $C$.

In contrary, in both serial and diverging connections the conditions of the d-separations are fulfilled. Therefore, in this case, $A$ and $B$ are independent given $C$. More specifically, in the serial connection, $\Pi_A = \{\oslash\}$, $\Pi_B = \{C\}$ and $\Pi_C = \{A\}$ and therefore:

$$\boxed{P(A, B, C) = P(B|C)P(C|A)P(A)} \tag{3.15}$$

Finally, in the diverging connection, $\Pi_A = \{C\}$, $\Pi_B = \{C\}$ and $\Pi_C = \{\oslash\}$ and therefore:

$$\boxed{P(A,B,C) = P(A|C)P(B|C)P(C)} \tag{3.16}$$

### 3.5.4 Markov Blankets

Markov blanket is an essential quantity and is connected to the concept of Maps and D-separation. The set of nodes that completely d-separates a specific node from the rest of the Bayesian graph is represented by the Markov blanket. The definition of the Markov blanket of a node $A \in V$ is the minimal subset $S$ of $V$ so that:

$$\boxed{A \perp\!\!\!\perp_P V - S - A|S} \tag{3.17}$$

In simple words, the parents, children and all the other nodes sharing a child with a specific node $A$ consist the Markov blanket of the node $A$ (Hoyle (2012)).

Markov blankets make the comparison of Bayesian networks with undirected graphs more easy than ever. Directed acyclic graphs have the ability to transform into undirected graph by following the two steps below:

1. First, the researcher must connect the non-adjacent nodes in each v-structure with an undirected arc.

2. Second, ignore the other arcs' direction and replace the arcs with edges.

## 3.6 Bayesian Networks and Inference

### 3.6.1 Bayesian Inference and Statistics

The main goal of the field of statistics is to answer questions that the human eye cannot detect concerning a particular set of data. Under this prism, Bayesian networks utilize evidence which are discovered by the researcher known as inference. The questions set by the researcher are called queries. In the process of answering these questions according to the Bayesian logic, the probabilistic reasoning is essential.

Inference and Statistics are two different terms. Statistics finds its origin in statistical procedures which aim to summarize the information of a data set and perform inference with a probabilistic model. The Bayesian Network, is a tough concept to understand. It has the ability to propose a probabilistic model regarding a number of variables without any usage of data. For some people Bayesian Statistics in the form of a Bayesian Network have the following form: Parameters $\rightarrow$ Data. The multivariate node has a marginal distribution and is considered the prior (Parameters). The likelihood is the conditional distribution of the multivariate node (Data). In the end, the conditional distribution of the multivariate node (Parameters) for the actual (Data) is simply the posterior distribution. Essentially, Bayesian Network is a Bayesian statistical procedure.

When the researcher asks a query to a Bayesian Network, a Bayesian statistical approach is applied. The procedure of querying a Bayesian Network is done through fixing some of its nodes and updating the local probability distributions in this new context. If the researcher accepts to assimilate the fixed node(s) to observation(s), then the application can be considered into the Bayesian statistical framework (Scutari and Denis (2021)). Bayesian Networks have the ability to express any probabilistic model expressed in the statistical approach.

### 3.6.2 Algorithms for Exact and Approximate Inference

A basic problem when the researchers evaluates the queries is the estimation of the posterior probabilities. More specifically, queries which involve very small or very large networks are difficult to deal with. This is because is such cases, the probabilistic problems are challenging. On top of that, the computational complexity can even get to be exponential in the number of variables.

When one is referring to belief updating algorithms, he talks for either exact or approximate ones. Both of these categories of algorithm have the basic properties of Bayesian Networks implemented as those were discussed in previous sections. Of course, as already discussed, the goal is to utilize local distributions to reduce the dimensions.

For example, marginalization is expressed as demonstrated in Equation 3.18.

$$P(Q|E,G,\Theta) = \int P(X|E,G,\Theta)d(X/Q) = \tag{3.18}$$

$$= \int \left[\prod_{i=1}^{p} P(X_i|E,\Pi_{X_i},\Theta_{X_i})\right] d(X/Q) = \prod_{i:X_i\in Q} \int P(X_i|E,\Pi_{X_i},\Theta_{X_i})dX_i \tag{3.19}$$

The correspondence between d-separation and conditional independence plays an important part in the reduction of a problem's dimension. By the definition of d-separation, the d-separation between other variables from $Q$ and $E$ is equivalent in the former not able to influence the outcome of the query. Thus, it is fine for them to be removed from the calculation of the posterior probabilities.

Exact inference algorithms are nothing but constant repetition of the Bayes' theorem, but with local computations to get the exact values $P(Q|E,G,\Theta)$ or $f(Q|E,G,\Theta)$. However, such algorithms are only for simplistic networks such as trees and polytrees. Two of the most popular exact inference algorithms are variable elimination and junction trees. Their original development was targeting discrete Bayesian Networks, but later they were modified for continuous and mixed Bayesian Networks as well. The direct structure of the Bayesian network is utilizing in variable elimination. During variable elimination, the specification of the optimal sequence of operations on the local distributions and the way to cache intermediate results to avoid unnecessary computations take place (Nagarajan, Scutari, and Lèbre (2013)). First, the transformation of the the Bayesian network into a junction tree takes place to perform belief updates. Below the steps of the junction tree clustering algorithm is demonstrated.

1. Create the moral graph of the Bayesian network $B$
2. Break every cycle spanning 4 or more nodes into subcycles of exactly 3 nodes by adding arcs to the moral graph, thus obtaining a triangulated graph.
3. Identify the cliques of the triangulated graph (i.e., maximal subsets of nodes in which each element is adjacent to all the others).
4. Create a tree in which each clique is a node, and adjacent cliques are linked by arcs.
5. Use the parameters of the local distributions of $B$ to compute the parameter sets of the compound nodes of the junction tree.

   (Nagarajan, Scutari, and Lèbre (2013))

Monte Carlo simulations are utilized by approximate inference algorithms to obtain a sample from the local distributions. By doing that, the estimation of $P(Q|E,G,\Theta)$ or

$f(Q|E, G, \Theta)$ is feasible. Such algorithms involve the generation of large number of samples from $B$ and the estimation of the relevant conditional probabilities. The latter is done by weighting the samples that include both $E$ and $Q = q$ against the ones that include only $E$ (Nagarajan, Scutari, and Lèbre (2013)). Monte Carlo simulations is not the only approach to do random sampling and weighting. There are plenty of approximate inference algorithms. One of the most popular is rejection sampling. Additionally, the variety of weight functions can go from the uniform distribution to likelihood functions to various estimates of posterior probability. One of the most simplistic approaches is logic sampling.

The steps of the algorithm are demonstrated below and more information can be found in Korb and Nicholson (2010).

1. Order the variables $X$ according to the topological ordering implied by $G$, for example $X_{(1)} < X_{(2)} < ... < X_{(p)}$.
2. For an appropriately large number of samples $x^* = (x_1^*, ..., (x_p^*)$:
   (a) For $i = 1, ..., p$, generate $x_i^*$ from $X_{(i)}|\Pi_{Q_{(i)}}$,
   (b) if $x$ includes $E$, then set $n_E = n_E + 1$.
   (c) if $X$ includes both $Q = q$ and $E$, then set $n_{E,q} = n_{E,q} + 1$.
3. Estimate $P(Q|E, G, \Theta)$ with $n_{E,q}/n_E$.

   (Nagarajan, Scutari, and Lèbre (2013))

Logic sampling is nothing but a combination of rejection sampling and uniform weights. Logic sampling counts the proportion of generated samples including $E$ that also include $Q = q$. However, in case $P(E)$ is not large enough, then the algorithm does not operate well. The latter happens because most particles will be removed without participation in estimating $P(Q|E, G, \Theta)$. Nevertheless, logic sampling is very simple and can handle very complex specifications of $E$ and $Q$ for both MAP$(Q|E, B)$ and CPQ$(Q|E, B)$ (Nagarajan, Scutari, and Lèbre (2013)). On the other hand, approximate algorithms are more suitable for the estimation of small conditional probabilities such as $10^{-50}$ and do better on large Bayesian Networks. However, the researcher should be cautious because approximate algorithms assume a discrete nature of the network.

## 3.7 Lab: Bayesian Networks

### 3.7.1 R Packages for Bayesian Networks

As already mentioned when explicitly analyzing the idea behind R, R has an enormous amount of packages for almost every kind of statistical analysis. In the case of Bayesian Networks, R has some packages worth mentioning and analyzing. They are can be separated into two categories: the packages that exclusively help in learning parameters and inference and those which apply structure and parameter learning. The packages that will be analyzed in this section are: *bnlearn*, *deal*, *pcalg*, *catnet*, *gRbase*, *gRain* and *rbmn*. The packages *bnlearn*, *pcalg* and *catnet* are considered to be in the first category, while *gRbase* and *gRain* fall in the second (Scutari and Denis (2021)).

*bnlearn* consists of all three kinds of algorithms for structure learning (Constraint-Based, Score-Based and Hybrid) along with multiple tests and network scores. Additionally, *bnlearn* includes methods to learn the parameters of a Bayesian Network such as maximum likelihood and Bayesian estimation along with several inference techniques such as conditional

probability queries and prediction. The comparative advantage of *bnlearn* is the fact that it distinguishes the structure of a Bayesian Network from the corresponding local probability distributions. The distinction is so clear that *bnlearn* has two different classes of R objects for them.

*deal* help in structure and parameter learning by utilizing the Bayesian approach. The comparative advantage of this package is that it can manage Bayesian Networks with both discrete and continuous nodes via a conditional Gaussian distribution. The rules of this combination are: a) discrete parents determine the variances of continuous nodes and b) continuous nodes are not allowed to be parents of discrete ones. The learning process of the structure of the Bayesian Network is done via the hill-climbing greedy search algorithm as described in the corresponding section of the paper. The specific algorithm is executed with random restarts and the posterior density of the network as a score function evade local maxima.

*pcalg*, as the name implies, includes the PC algorithm and its particular development focuses on estimating and measuring causal effects. As with *deal*, *pcalg* can manage both discrete and continuous data. A very interesting feature of the package is the fact that it can handle the presence of latent variables on the Bayesian network. This is done by utilizing a specialized algorithm called Fast Causal Inference, also known as FCI (Spirtes et al. (2000)), which is based on the modification of the PC algorithm.

*catnet* package is specialized on discrete Bayesian Networks by utilizing classical frequentist methods. The learning of the structure is achieved with two steps:

1. Simulated annealing algorithm is used to learn the node ordering of the DAG from the data. If the researcher decides so, he can also specify a custom node ordering.

2. The maximum likelihood solution is extracted by performing a detailistic search among the network structures with the given node ordering. Finally, learning of the parameters and prediction are applied.

The *gRbase* and *gRain* are packages specialize in discrete Bayesian Networks. They are designed to manipulate the parameters of the network, on prediction, and on inference. Since they fall in the second category, they do not include any structure learning algorithm and therefore the specification of the Bayesian Network is assigned to the researcher.

*rbmn* is specialized in linear Gaussian Bayesian Networks. They derive joint and conditional distributions regarding subsets of nodes. As with *gRbase* and *gRain*, *rbmn* includes no parameter learning algorithm.

Table 3.1: The summary table of the features of each of the R packages regarding Bayesian Networks, adopted from Scutari and Denis (2021).

| | *bnlearn* | *catnet* | *deal* | *pcalg* | *gRbase* | *gRain* | *rbmn* |
|---|---|---|---|---|---|---|---|
| Discrete data | Yes | Yes | Yes | Yes | Yes | Yes | No |
| Continuous data | Yes | No | Yes | Yes | Yes | No | Yes |
| Mixed data | No | No | Yes | No | No | No | No |
| Constraint-based algthm | Yes | No | No | Yes | No | No | No |
| Score-based algthm | Yes | Yes | Yes | No | No | No | No |
| Hybrid algthm | Yes | No | No | No | No | No | No |
| Structure manipulation | Yes | Yes | No | No | Yes | No | No |
| Parameter estimation | Yes | Yes | Yes | Yes | No | No | Yes |
| Prediction | Yes | Yes | No | No | No | Yes | Yes |
| Approximate Inference | Yes | No | No | No | No | Yes | No |

### 3.7.2 The Company Dataset

The Company dataset consists of data obtained by employees of a company regarding both demographic and job characteristics. The same dataset was used as input in the previous chapter to conduct SEM. As a reminder, the final SEM model is demonstrated in Figure 3.3. The particular variables, paths and associations of the SEM model will be also used as input to construct the DAG of the Bayesian Network.

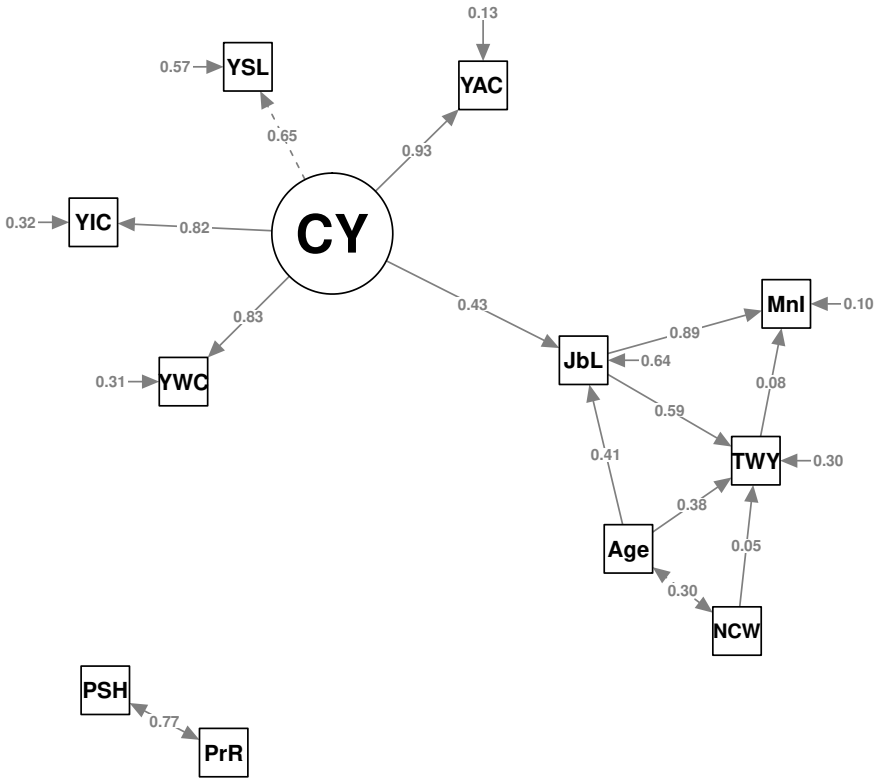



Figure 3.3: The graphical representation of the final SEM model which will be used in the Bayesian Networks.

#### 3.7.2.1 Data Preparation

The first step in every statistical analysis in R is to actually import and read the dataset. The specific dataset is in .csv format and thus, the `read.csv` function will be used. The `read.csv` function takes the location of the csv file as main input. It is important to notice that the specific function requires forward slashes in the location of the file. Additional arguments are `header` which states whether the first line of the dataset represents variable names and `sep` which indicates the symbol by which the data values of the dataset are separated. The dataset is saved in an R object called `df`.

The following lines of code successfully read and import the desired dataset.

```
#Reading the file in the working directory
>df<-read.csv("C:/Users/Yannis/Desktop/PAMAK_THESIS_R_CODES/
Attrition_project.csv",header = T,sep = ",")
```

The next step is to isolate and select the variables of interest according to the SEM model (Figure 3.3). According to the SEM model, the 9 variables which must be selected are: $Age$, $JobLevel$, $MonthlyIncome$, $NumCompaniesWorked$, $TotalWorkingYears$, $YearsAtCompany$, $YearsInCurrentRole$, $YearsSinceLastPromotion$, $YearsWithCurrentManager$. In order to isolate these variables from the dataset, the `select` function will be used from the *dplyr* package. The following lines of code successfully update the previous `df` according to the needs of the analysis.

```
 #Selecting the desired variables for the BN (from a SEM model)
>df<-df %>% select(Age,JobLevel,MonthlyIncome,NumCompaniesWorked,
TotalWorkingYears,YearsAtCompany,YearsInCurrentRole,YearsSinceLastPromotion,
YearsWithCurrManager)
```

The next step is to get a taste from the dataset by viewing some of its values and reviewing its structure. This task is achieved by using the `str` command. The lines of code are demonstrated below.

```
#The structure of the dataset
>str(df)

Output:
'data.frame': 1470 obs. of  9 variables:
 $ Age                     : int  41 49 37 33 27 32 59 30 38 36 ...
 $ JobLevel                : int  2 2 1 1 1 1 1 1 3 2 ...
 $ MonthlyIncome           : int  5993 5130 2090 2909 3468 3068 2670
 2693 9526 5237 ...
 $ NumCompaniesWorked      : int  8 1 6 1 9 0 4 1 0 6 ...
 $ TotalWorkingYears       : int  8 10 7 8 6 8 12 1 10 17 ...
 $ YearsAtCompany          : int  6 10 0 8 2 7 1 1 9 7 ...
 $ YearsInCurrentRole      : int  4 7 0 7 2 7 0 0 7 7 ...
 $ YearsSinceLastPromotion: int  0 1 0 3 2 3 0 0 1 7 ...
 $ YearsWithCurrManager    : int  5 7 0 0 2 6 0 0 8 7 ...
```

According to the output above, all the variables are of integer class. Now, each of the variables will be manipulated to become factors with discrete values. The reason why the numerical variables will be transformed into discrete factors is for the application of the discrete Bayesian Network. The R command which assigns the data values into categories is called `cut`. `cut` takes an input the variable, the intervals of each of the categories (with `breaks`) and the name of each of the categories (with `labels`). Finally, the `include.lowest=T` argument makes the right bracket of the first interval which corresponds to the first category closed ($[a, b]$). Then, for R to understand that the names correspond to actual categories the `factor` function is used. `factor` uses the name of the variable and levels as inputs.

1. The first variable is *Age*. The data values of *Age* will be putted into 3 categories. Namely those categories are young((20,40]), adult((40,60]) and old((60,80]). The structure of the variable is confirmed via the `str` command. The following lines of code successfully transform *Age* into discrete factor.

```
#Making the Categories
>df$Age<-cut(df$Age,breaks=c(0,20,40,60),labels = c("young",
"adult","old"),include.lowest=T)

#Making Age a factor
>df$Age<-factor(df$Age,levels =c("young","adult","old"))

#Structure of the variable
>str(df$Age)

Output:
Factor w/ 3 levels "young","adult",..: 3 3 2 2 2 2 3 2 2 2 ...
```

2. The second variable is *JobLevel*. The data values of *JobLevel* will be putted into 3 categories. Namely those categories are low([0,2]), good((2,4]), excellent((4,6]). Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform *JobLevel* into discrete factor.

```
#Making the Categories
>df$JobLevel<-cut(df$JobLevel,breaks=c(0,2,4,6),labels = c("low",
"good","excellent"),include.lowest=T)

#Making Age a factor
>df$JobLevel<-factor(df$JobLevel,levels =c("low","good","excellent"))

#Structure of the variable
>str(df$JobLevel)

Output:
Factor w/ 3 levels "low","good","excellent": 1 1 1 1 1 1 1 1 2 1 ...
```

3. The third variable is *MonthlyIncome*. The data values of *MonthlyIncome* will be putted into 4 categories. Namely those categories are low([0,5000]), medium((5000,1000 0]), high((10000,15000]), very high((15000,20000]). Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform *MonthlyIncome* into discrete factor.

```
#Making the Categories
>df$MonthlyIncome<-cut(df$MonthlyIncome,breaks=c(0,5000,10000,15000,
20000),labels = c("low","medium","high","very high"),include.lowest=T)

#Making Age a factor
>df$MonthlyIncome<-factor(df$MonthlyIncome,levels =c("low","medium",
"high","very high"))

#Structure of the variable
>str(df$MonthlyIncome)

Output:
Factor w/ 4 levels "low","medium",..: 2 2 1 1 1 1 1 1 2 2 ...
```

4. The fourth variable is $NumCompaniesWorked$. The data values of $NumCompaniesWorked$ will be putted into 5 categories. Namely those categories are [0,2],(2,4],(4,6],(6,8], (8,10]. Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform $NumCompaniesWorked$ into discrete factor.

```
#Making the Categories
>df$NumCompaniesWorked<-cut(df$NumCompaniesWorked,
breaks=c(0,2,4,6,8,10),labels =c("[0,2]","(2,4]","(4,6]","(6,8]",
"(8,10]"),include.lowest=T)

#Making Age a factor
>df$NumCompaniesWorked<-factor(df$NumCompaniesWorked,
levels =c("[0,2]","(2,4]","(4,6]","(6,8]","(8,10]"))

#Structure of the variable
>str(df$NumCompaniesWorked)
Output:
Factor w/ 5 levels "[0,2]","(2,4]",..: 4 1 3 1 5 1 2 1 1 3 ...
```

5. The fifth variable is $TotalWorkingYears$. The data values of $TotalWorkingYears$ will be putted into 4 categories. Namely those categories are [0,10],(10,20],(20,30],(30,40]. Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform $TotalWorkingYears$ into discrete factor.

```
#Making the Categories
>df$TotalWorkingYears<-cut(df$TotalWorkingYears,
breaks=c(0,10,20,30,40),labels =c("[0,10]","(10,20]",
"(20,30]","(30,40]"),include.lowest=T)

#Making Age a factor
>df$TotalWorkingYears<-factor(df$TotalWorkingYears,
levels =c("[0,10]","(10,20]","(20,30]","(30,40]"))

#Structure of the variable
>str(df$TotalWorkingYears)

Output:
Factor w/ 4 levels "[0,10]","(10,20]",..: 1 1 1 1 1 1 2 1 1 2 ...
```

6. The sixth variable is $YearsAtCompany$. The data values of $YearsAtCompany$ will be putted into 4 categories. Namely those categories are [0,10],(10,20],(20,30],(30,40]. Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform $YearsAtCompany$ into discrete factor.

```
#Making the Categories
>df$YearsAtCompany<-cut(df$YearsAtCompany,
breaks=c(0,10,20,30,40),labels =c("[0,10]","(10,20]","(20,30]",
"(30,40]"),include.lowest=T)
```

```
#Making Age a factor
>df$YearsAtCompany<-factor(df$YearsAtCompany,
levels =c("[0,10]","(10,20]","(20,30]","(30,40]"))

#Structure of the variable
>str(df$YearsAtCompany)

Output:
Factor w/ 4 levels "[0,10]","(10,20]",..: 1 1 1 1 1 1 1 1 1 1 ...
```

7. The seventh variable is $YearsInCurrentRole$. The data values of $YearsInCurrentRole$ will be putted into 4 categories. Namely those categories are [0,5],(5,10],(10,15],(15,20 ). Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform $YearsInCurrentRole$ into discrete factor.

```
#Making the Categories
>df$YearsInCurrentRole<-cut(df$YearsInCurrentRole,
breaks=c(0,5,10,15,20),labels = c("[0,5]","(5,10]","(10,15]",
"(15,20)"),include.lowest=T)

#Making YearsInCurrentRole a factor
>df$YearsInCurrentRole<-factor(df$YearsInCurrentRole,
levels =c("[0,5]","(5,10]","(10,15]","(15,20)"))

#Structure of the variable
>str(df$YearsInCurrentRole)

Output:
Factor w/ 4 levels "[0,5]","(5,10]",..: 1 2 1 2 1 2 1 1 2 2 ...
```

8. The eighth variable is $YearsSinceLastPromotion$. The data values of $YearsSinceLastPromotion$ will be putted into 3 categories. Namely those categories are [0,5],(5,10],(10, 15]. Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform $YearsSinceLastPromotion$ into discrete factor.

```
#Making the Categories
>df$YearsSinceLastPromotion<-cut(df$YearsSinceLastPromotion,
breaks=c(0,5,10,15),labels = c("[0,5]","(5,10]","(10,15]"),
include.lowest=T)

#Making YearsSinceLastPromotion a factor
>df$YearsSinceLastPromotion<-factor(df$YearsSinceLastPromotion,
levels =c("[0,5]","(5,10]","(10,15]"))

#Structure of the variable
>str(df$YearsSinceLastPromotion)

Output:
Factor w/ 3 levels "[0,5]","(5,10]",..: 1 1 1 1 1 1 1 1 1 2 ...
```

9. The last variable is $YearsWithCurrManager$. The data values of $YearsWithCurrManager$ will be putted into 4 categories. Namely those categories are [0,5],(5,10],(10,15],(15,20). Finally, the structure of the variable is confirmed via the `str` command. The following lines of code successfully transform $YearsWithCurrManager$ into discrete factor.

```
#Making the Categories
>df$YearsWithCurrManager<-cut(df$YearsWithCurrManager,
breaks=c(0,5,10,15,20),labels = c("[0,5]","(5,10]","(10,15]",
"(15,20)"),include.lowest=T)

#Making YearsWithCurrManager a factor
>df$YearsWithCurrManager<-factor(df$YearsWithCurrManager,
levels =c("[0,5]","(5,10]","(10,15]","(15,20)"))

#Structure of the variable
>str(df$YearsWithCurrManager)

Output:
Factor w/ 4 levels "[0,5]","(5,10]",..: 1 2 1 1 1 2 1 1 2 2 ...
```

Table 3.2: The 9 variables which will be used in the BN along with their categories.

| Variable | Categories |
|---|---|
| Age | young,adult,old |
| JobLevel | low,good,excellent |
| MonthlyIncome | low,medium,high,very high |
| NumCompaniesWorked | [0,2],(2,4],(4,6],(6,8],(8,10] |
| TotalWorkingYears | [0,10],(10,20],(20,30],(30,40] |
| YearsAtCompany | [0,10],(10,20],(20,30],(30,40] |
| YearsInCurrentRole | [0,5],(5,10],(10,15],(15,20) |
| YearsSinceLastPromotion | [0,5],(5,10],(10,15] |
| YearsWithCurrManager | [0,5],(5,10],(10,15],(15,20) |

#### 3.7.2.2 Building the DAG

In the starting of building the DAG model the names of the variables will be shorted. This is done for them to be more easily used through out the analysis. More specifically, $Age$, $JobLevel$, $MonthlyIncome$, $NumCompaniesWorked$, $TotalWorkingYears$, $YearsAtCompany$, $YearsInCurrentRole$, $YearsSinceLastPromotion$ and $YearsWithCurrentManager$ will be shorted as $Age$,$JbL$,$Mnl$,$NCW$,$TWY$,$YAC$,$YIC$,$YSL$ and $YWC$, respectively. Then, an empty BN graph will be created with the `empty.graph`. Finally, the empty BN is demonstrated in Figure 3.4 via the `plot(empty.net)` command. The tasks above are done with the following lines of code.

```
#Shorting the variable names
>names(df)[1:9]<-c("Age","JbL","Mnl","NCW","TWY","YAC","YIC","YSL","YWC")

#Empty Bayesian Network Graph
```

```
>empty.net <- empty.graph(nodes = c("Age","JbL","Mnl","NCW","TWY","YAC",
"YIC","YSL","YWC"))
>empty.net

Output:
  Random/Generated Bayesian network

  model:
   [Age][JbL][Mnl][NCW][TWY][YAC][YIC][YSL][YWC]
  nodes:                                 9
  arcs:                                  0
    undirected arcs:                     0
    directed arcs:                       0
  average markov blanket size:           0.00
  average neighbourhood size:            0.00
  average branching factor:              0.00

  generation algorithm:                  Empty

 #Plotting the empty BN
>plot(empty.net)
```

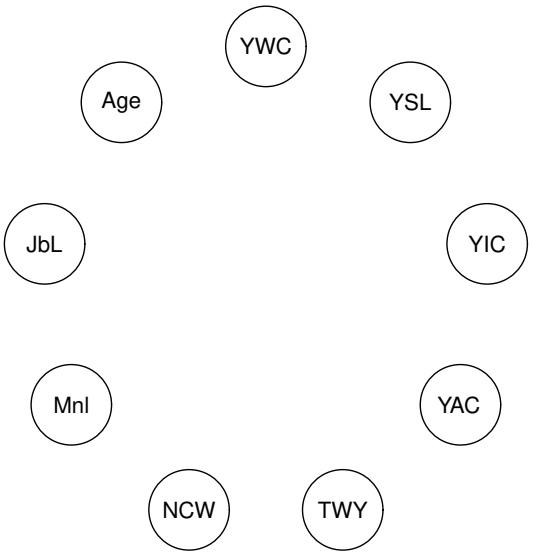



Figure 3.4: The empty BN of the Company dataset.

According to Figure 3.3, the arcs which must be specified are the following: $YWC \rightarrow JbL$, $YIC \rightarrow JbL$, $YSL \rightarrow JbL$, $YAC \rightarrow JbL$, $JbL \rightarrow Mnl$, $JbL \rightarrow TWY$, $Age \rightarrow JbL$, $Age \rightarrow TWY$, $TWY \rightarrow Mnl$ and $NCW \rightarrow TWY$. The `set.arc` function is used to specify the arcs. Finally, in Figure 3.5, the final DAG is demonstrated.

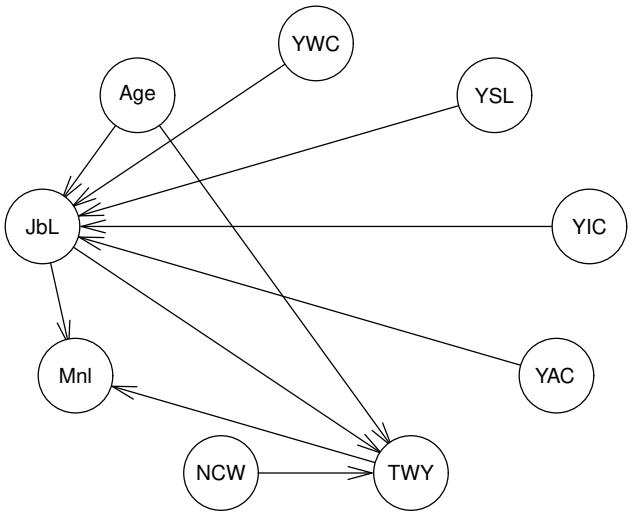



Figure 3.5: The final DAG of the Company dataset.

The following lines of code produce the DAG of the Company dataset:

```
#Arc from YearsWithCurrentManager to JobLevel
>DAG <- set.arc(DAG, from = "YWC", to = "JbL")

#Arc from YearsInCurrentRole to JobLevel
>DAG <- set.arc(DAG, from = "YIC", to = "JbL")

#Arc from YearsSinceLastPromotion to JobLevel
>DAG <- set.arc(DAG, from = "YSL", to = "JbL")

#Arc from YearsAtCompany to JobLevel
>DAG <- set.arc(DAG, from = "YAC", to = "JbL")

#Arc from JobLevel to MonthlyIncome
>DAG <- set.arc(DAG, from = "JbL", to = "Mnl")

#Arc from JobLevel to TotalWorkingYears
>DAG <- set.arc(DAG, from = "JbL", to = "TWY")

#Arc from Age to JobLevel
>DAG <- set.arc(DAG, from = "Age", to = "JbL")
```

```
#Arc from Age to TotalWorkingYears
>DAG <- set.arc(DAG, from = "Age", to = "TWY")

#Arc from TotalWorkingYears to MonthlyIncome
>DAG <- set.arc(DAG, from = "TWY", to = "Mnl")

#Arc from NumCompaniesWorked to TotalWorkingYears
>DAG <- set.arc(DAG, from = "NCW", to = "TWY")
>DAG

#Plotting the final DAG
>plot(DAG)
```

Additional information can be obtained from the DAG through `nodes`, `arcs` and `modelstring`. The output of `nodes` is the total nodes of the DAG. The output of `arcs` is the arcs between the nodes along with their starting and ending node. The output of `modelstring` is the DAG represented algebraically. The following lines of code implement the functions above and reveal their results.

```
#Nodes of Final DAG
>nodes(DAG)

Output:
[1] "Age" "JbL" "Mnl" "NCW" "TWY" "YAC" "YIC" "YSL" "YWC"

#Arcs of Final DAG
>arcs(DAG)

Output:
      from  to
 [1,] "YWC" "JbL"
 [2,] "YIC" "JbL"
 [3,] "YSL" "JbL"
 [4,] "YAC" "JbL"
 [5,] "JbL" "Mnl"
 [6,] "JbL" "TWY"
 [7,] "Age" "JbL"
 [8,] "Age" "TWY"
 [9,] "TWY" "Mnl"
[10,] "NCW" "TWY"

#Model String Representation of Final DAG
>modelstring(DAG)

Output:
[1] "[Age][NCW][YAC][YIC][YSL][YWC][JbL|Age:YAC:YIC:YSL:YWC]
[TWY|Age:JbL:NCW][Mnl|JbL:TWY]"
```

#### 3.7.2.3 Parameter Estimation

The `bn.fit` function will be used to compute the parameters of the local distribution from the observed sample. The two methods available for parameter estimation are maximum likelihood estimation and bayes. The latter method takes an additional argument, `iss`, known as imaginary sample size. `iss` decides the weight assigned to the prior distribution compared to the data when calculating the posterior. Posterior estimates are more robust than mle and result in BNs with better predictive power. Under this framework, bayes method is considered more preferable. Finally, the number of parameters of the BN is 1340 and the dataset has 1470 observations, thus the estimation is valid. The following lines of code implement `bn.fit` with both maximum likelihood estimation and bayes and display the number of parameters of the BN.

```
#Parameter Estimation of the BN with mle
>bn.estimation.mle <- bn.fit(DAG, data = df, method = "mle")

#Parameters of node Age
>bn.estimation.mle$Age

Output:

  Parameters of node Age (multinomial distribution)

Conditional probability table:
    children       young young adult       adult         old
 0.01904762  0.24353741  0.42108844  0.21904762  0.09727891

#Parameters of node NCW
>bn.estimation.mle$NCW

Output:

  Parameters of node NCW (multinomial distribution)

Conditional probability table:
      [0,2]      (2,4]      (4,6]      (6,8]     (8,10]
0.58775510 0.20272109 0.09047619 0.08367347 0.03537415

#Parameter Estimation of the BN with bayes
>bn.estimation.bayes <- bn.fit(DAG, data = df, method = "bayes",iss=10)

#Parameters of node Age
>bn.estimation.bayes$Age

Output:
  Parameters of node Age (multinomial distribution)

Conditional probability table:
    children       young young adult       adult         old
 0.02027027  0.24324324  0.41959459  0.21891892  0.09797297
```

```
#Parameters of node NCW
>bn.estimation.bayes$NCW

Output:
  Parameters of node NCW (multinomial distribution)

Conditional probability table:
      [0,2]      (2,4]      (4,6]      (6,8]     (8,10]
0.58513514 0.20270270 0.09121622 0.08445946 0.03648649

#They have the same number of parameters
>nparams(bn.estimation.bayes)==nparams(bn.estimation.mle)

Output:
[1] TRUE

#Number of Parameters of the Network
>nparams(bn.estimation.bayes)

Output:
[1] 1340
```

#### 3.7.2.4 Network Tests and Scores

The `arc.strength` function will be used to determine the strength of the probabilistic dependence corresponding to each arc by removing that particular arc from the graph and quantifying the change with some probabilistic criterion such as Pearson's $X^2$ or Mutual Information $MI$. In R, the argument `criterion` takes the values $x2$ and $mi$ for the two criteria, respectively. The conditional independence tests are about the independence of the ending node of an arc from the starting node of an arc, conditional on the remaining parents of the ending node. The null hypothesis is that the arc is not well-supported by the data. The reported column `strength` displays the p-value of each of the arcs. In R, the `score` function is used with DAG, dataset and type of index as inputs. Of course, the higher the BIC and BDe indices, the better for the network. Both `arc.strength` and `score` belong to the *bnlearn* package. The following lines of code reveal the conditional independence test of each of the node with $X^2$ and MI criterion and the network score with BIC and BDe.

```
#Conditional Independence Tests - Arc Strength
>arcs_power<-arc.strength(DAG, data = df, criterion = "mi")
>arcs_power

Output:
   from  to      strength
1   YWC JbL  1.000000e+00
2   YIC JbL  1.000000e+00
3   YSL JbL  1.000000e+00
4   YAC JbL  1.000000e+00
5   JbL Mnl 4.945093e-112
6   JbL TWY  1.302706e-80
7   Age JbL  1.000000e+00
```

```
8   Age TWY  5.939365e-12
9   TWY Mnl  1.272808e-32
10  NCW TWY  9.979279e-01

#Rounding up the strength
>arcs_strength<-arcs_power$strength
>round(arcs_strength,digits=4)

Output:
[1] 1.0000 1.0000 1.0000 1.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.9979



#BIC
>bnlearn::score(DAG, data = df,type = "bic")

Output:
[1] -14108.1

#BDE
>bnlearn::score(DAG, data = df,type = "bde", iss = 10)

Output:
[1] -9831.67
```

According to the output, the following arcs: $YWC \rightarrow JbL$, $YIC \rightarrow JbL$, $YSL \rightarrow JbL$, $YAC \rightarrow JbL$, $Age \rightarrow JbL$, $NCW \rightarrow TWY$ have p-values larger than 0.05 and are not well-supported by the data. Thus, they should be removed. Additionally, $BIC = -14108.1$ and $BDE = -9831.67$ are also considered low. Therefore, the model should be modified to improve those indices. In order for the arcs to be relevant to the data and the network scores to be improved, several actions must be taken. There are several algorithms such as hill-climbing which aim to find the DAG with the best network score. However, hill-climbing doesn't take into consideration the nature of the variables. More specifically, hill-climbing might indicate that a specific arc starts at an endogenous variable and ends on an exogenous variable (e.g. Age). Such specification would be theoretically unjustifiable. This is an existing problem with the hill-climbing algorithm. Thus, in this case, because of the limited number of variables and their nature, the improved model will be specified manually. The steps of the algorithm are the following:

1. Remove the insignificant arcs from the original model.

2. Repeat #ADDING, #TEST AND SCORES and #REMOVING until the network score cannot be improved AND all arcs are relatively supported by the data:

   (a) #ADDING: Add every possible combination of remaining arc pairs with respect to the basic rules of path analysis (loops, exogenous variables etc.) and theoretical justification of the candidate arc.

   (b) #TEST AND SCORES: if the p-value of the Conditional Independence Test is above 0.05 (p-value<=0.05) and network score is increased, then add the path. If p-value>0.05 or the network score is not increased remove the path with the next step.

(c) #REMOVING: Remove the candidate arc.

In R, the `drop.arc` function is used to remove an arc and the `set.arc` function to add an arc. The algorithm explained above, the paths removed and added to the network are demonstrated with the following lines of code.

```
#Creating a 2nd DAG
DAG2<-DAG

#Example of the algorithm with the "NCW"->"YAC" arc
#ADDING
>DAG2 <- set.arc(DAG2, from = "NCW", to = "YAC")
>plot(DAG2)

#TESTS AND SCORES
#Conditional Independence Test
>arcs_power<-arc.strength(DAG2, data = df, criterion = "mi");arcs_power
>arcs_strength<-arcs_power$strength
>round(arcs_strength,digits=4)

#BIC
>bnlearn::score(DAG2, data = df,type = "bic")

#BDE
>bnlearn::score(DAG2, data = df,type = "bde", iss = 10)

#REMOVING
>DAG2 <- drop.arc(DAG2, from = "NCW", to = "YAC")
>plot(DAG2)
```

For the example, after the algorithm is completed, the following paths: $NCW \rightarrow TWY$, $YWC \rightarrow JbL$ and $YAC \rightarrow JbL$ are removed and the paths $YAC \rightarrow YWC$, $YAC \rightarrow YSL$, $YAC \rightarrow YIC$, $Age \rightarrow YAC$, $YWC \rightarrow YIC$, $NCW \rightarrow YAC$. The following lines of code form the new DAG which is saved in a new R object called `DAG2`.

```
#Removing arc from NCW to TWY
>DAG2 <- drop.arc(DAG2, from = "NCW", to = "TWY")
>plot(DAG2)

#Removing arc from YWC to JbL
>DAG2 <- drop.arc(DAG2, from = "YWC", to = "JbL")
>plot(DAG2)

#Removing arc from YAC to JbL
>DAG2 <- drop.arc(DAG2, from = "YAC", to = "JbL")
>plot(DAG2)

#Adding arc from YAC to YWC
>DAG2 <- set.arc(DAG2, from = "YAC", to = "YWC")
>plot(DAG2)
```

```
#Adding arc from YAC to YSL
>DAG2 <- set.arc(DAG2, from = "YAC", to = "YSL")
>plot(DAG2)

#Adding arc from YAC to YIC
>DAG2 <- set.arc(DAG2, from = "YAC", to = "YIC")
>plot(DAG2)

#Adding arc from Age to YAC
>DAG2 <- set.arc(DAG2, from = "Age", to = "YAC")
>plot(DAG2)

#Adding arc from YWC to YIC
>DAG2 <- set.arc(DAG2, from = "YWC", to = "YIC")
>plot(DAG2)

#Adding arc from NCW to YAC
>DAG2 <- set.arc(DAG2, from = "NCW", to = "YAC")
>plot(DAG2)
```

It is now time to compare the new DAG with the modified DAG2. 50% of the arcs of the original DAG didn't even pass the conditional independence test. In the new DAG2, every arc is supported by the data ($NCW \rightarrow YAC$ with p-value of 0.051 is barely acceptable). Recall that, the original DAG yielded network scores of $BIC = -14108.1$ and $BDE = -9831.67$. However, the new DAG2 yielded network scores of $BIC = -9203.69$ and $BDE = -8662.618$. Thus, the BIC network score was significantly improved up by almost 30%.

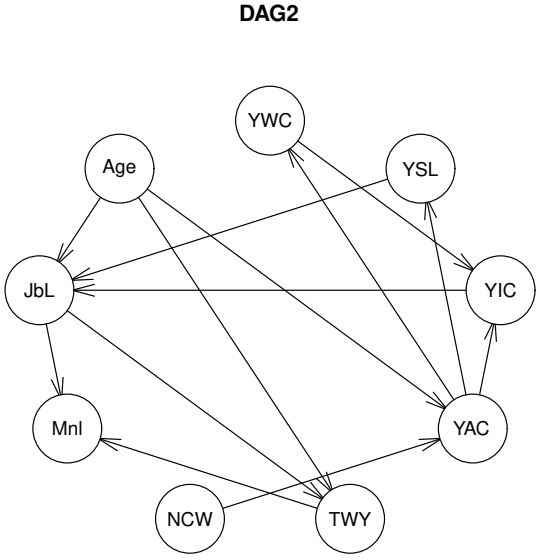



Figure 3.6: The new Directed Acyclic Graph (DAG2) of the Company dataset.

The following lines of code point out these results. The new DAG, DAG2, is demonstrated in Figure 3.6.

```
#The conditional Independence tests of DAG versus DAG2
#DAG
>arcs_power<-arc.strength(DAG, data = df, criterion = "mi")
>arcs_strength<-arcs_power$strength
>round(arcs_strength,digits=4)

Output:
 [1] 1.0000 1.0000 1.0000 1.0000 0.0000 0.0000 1.0000 0.0000
 0.0000 0.9979

#DAG2
>arcs_power<-arc.strength(DAG2, data = df, criterion = "mi")
>arcs_strength<-arcs_power$strength
>round(arcs_strength,digits=4)

Output:
 [1] 0.000 0.004 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.000 0.051

#The network scores of DAG versus DAG2

#BIC
>bnlearn::score(DAG, data = df,type = "bic")

Output:
[1] -14108.1

>bnlearn::score(DAG2, data = df,type = "bic")

Output:
[1] -9203.69

#BDE
>bnlearn::score(DAG, data = df,type = "bde", iss = 10)

Output:
[1] -9831.67

>bnlearn::score(DAG2, data = df,type = "bde", iss = 10)

Output:
[1] -8662.618
```

The estimation of the parameters of the new DAG2 again take place with the `bn.fit` function along with either maximum likelihood estimation or bayes estimation. As already mentioned, the bayes estimation will be preferred over the maximum likelihood estimation. In the code below, the bayes estimation of DAG2, parameters of $YSL$ and $YWC$ and number of parameters of DAG2 (through `nparams` function) are demonstrated. Notice how

the number of parameters of DAG2 is only 254, significantly lower than the original DAG's (1340).

```
#Parameter Estimation of the BN with bayes
>bn.estimation.bayes2 <- bn.fit(DAG2, data = df, method = "bayes",iss=10)

#Parameters of node YSL
>bn.estimation.bayes2$YSL

Output:
  Parameters of node YSL (multinomial distribution)

Conditional probability table:

         YAC
YSL             [0,10]      (10,20]      (20,30]      (30,40]
  [0,5]   0.9056937084 0.6511415525 0.3968253968 0.4234234234
  (5,10]  0.0936268515 0.1634703196 0.0920634921 0.1531531532
  (10,15] 0.0006794401 0.1853881279 0.5111111111 0.4234234234

#Parameters of node YWC
>bn.estimation.bayes2$YWC

Output:
  Parameters of node YWC (multinomial distribution)

Conditional probability table:

         YAC
YWC             [0,10]      (10,20]      (20,30]      (30,40]
  [0,5]   0.7644720750 0.0636986301 0.1071428571 0.0878378378
  (5,10]  0.2345087648 0.6280821918 0.6023809524 0.6283783784
  (10,15] 0.0005095801 0.2719178082 0.2404761905 0.1959459459
  (15,20) 0.0005095801 0.0363013699 0.0500000000 0.0878378378

#Number of Parameters of the Network
>nparams(bn.estimation.bayes2)

Output:
[1] 254
```

#### 3.7.2.5 Inference and Queries

Starting of with the exact inference. The *gRain* package is used to construct the junction tree which is necessary to reduce the time of calculation of the conditional probabilities. The `complile` function paired with the `as.grain` argument take the estimated parameters (in this case estimated with the bayes approach). The following line of code constructs the junction tree.

```
#Construction of the junction tree
>junction <- compile(as.grain(bn.estimation.bayes2))
```

Next up, the `setEvidence` function is used to put evidence into `junction`. The queries take place with the `querygrain` which contains the distribution of the desired nodes from `junction`. For example, one may wish to examine the Job Level variable of the adult people compared to the whole dataset. In this case, `querygrain` takes the $JbL$ node as input and `setEvidence` takes the `junction`, the node of interest ($Age$) and the category of interest ($adult$). The lines of code along with the results are displayed below.

```
#The Job Level of the whole dataset
>querygrain(junction, nodes = "JbL")$JbL



Output:
JbL
       low       good  excellent
0.71604353 0.22560846 0.05834801

#The Job Level of adult people
>age.adult <- setEvidence(junction, nodes = "Age", states = "adult")
>querygrain(age.adult, nodes = "JbL")$JbL

Output:
JbL
       low       good  excellent
0.83167462 0.15725407 0.01107131
```

It is interesting to notice that the probability of an individual to do his job poorly (low job level) is almost 12% more for adult people. Mathematically, the former probability is $Pr(JbL)$ and the latter probability is $Pr(JbL|Age = adult)$.

Another interesting query could be the Job Level of the people which state that they have very high Monthly Income. Theoretically, a person has very high monthly income when the quality of his job is good or in many cases excellent compared to the whole dataset. In this case, `querygrain` takes the $JbL$ node as input and `setEvidence` takes the `junction`, the node of interest ($Mnl$) and the category of interest (very high). The lines of code along with the results are displayed below.

```
#The Job Level of the whole dataset
>querygrain(junction, nodes = "JbL")$JbL

Output:
JbL
       low       good  excellent
0.71604353 0.22560846 0.05834801

#The Job Level of people with very high Monthly Income
>Mnl.very_high <- setEvidence(junction, nodes = "Mnl", states = "very high")
>querygrain(Mnl.very_high, nodes = "JbL")$JbL

Output:
JbL
        low        good   excellent
0.005286003 0.440542146 0.554171851
```

According to the results the theory is confirmed. The probability of a highly paid individual to also be professional or expert on his job (excellent job level) is almost 50% more compared to the whole dataset. Mathematically, the former probability is $Pr(JbL)$ and the latter probability is $Pr(JbL|Mnl = \text{very high})$. This example illustrates the power of the Bayesian Networks.

The queries which involve conditional probabilities can also be utilized to assess conditional independence. For example, consider the relationship between $Age$ and $Mnl$ given that $JbL = low$. Mathematically, the joint probability of $Age$ and $Mnl$ given that $JbL = low$ is expressed as $Pr(Age, Mnl|JbL = low)$. In this case, `querygrain` takes the nodes $Age$ and $Mnl$ as input and the argument `type=joint` to make R understand that is a joint probability. The `setEvidence` function takes the `junction`, the node of interest ($JbL$) and the category of interest ($low$). The lines of code along with the results are displayed below.

```
#The joint probability of Age and MonthlyIncome given that Job Level is low
>Age.Mnl_given_JbL.low <- setEvidence(junction, nodes = "JbL",
states = "low")
>rounding<-querygrain(Age.Mnl_given_JbL.low, nodes = c("Age","Mnl"),
type="joint")

#Rounding the results
>round(rounding,digits=5)

Output:
       Mnl
Age          low  medium    high very high
  young 0.01423 0.00482 0.00010   0.00010
  adult 0.54265 0.22589 0.00041   0.00041
  old   0.13692 0.07402 0.00023   0.00023
```

According to the results, the highest probability, given the low job level, for one to be adult with low monthly income is equal to 54%, while an adult with medium size monthly income is also likely to occur (22%). Overall, it seems like the probability of one being an adult given that the individual has low job level is high ($0.54265 + 0.22589 = 0.76854$).

Notice how the argument `type` of the `querygrain` function was set equal to `joint` for the joint probability of the nodes referred in the `nodes` argument ("Age" and "Mnl"). But one can also set `type=marginal`, which is the default choice is no type is specified, to get the marginal distribution of each node. Namely for "Age" and "Mnl", the code below reveals their marginal distribution.

```
#The marginal probability of Age and MonthlyIncome
>querygrain(Age.Mnl_given_JbL.low, nodes = c("Age", "Mnl"),
type = "marginal")

Output:

$Age
Age
     young      adult        old
0.01923653 0.76935375 0.21140971

$Mnl
```

```
Mnl
         low       medium         high    very high
0.6937968835 0.3047293462 0.0007368852 0.0007368852
```

The last choice for the researcher as input in the `type` argument is `conditional`. The `type=conditional` argument allows for `querygrain` to return the distribution of the first node in `nodes` conditional on the other nodes in `nodes` while also accounting for the given evidence. For example, let's examine the distribution of $NCW$ conditional on $YWC$ given that $YAC$ ranges from (10,20]. The following lines of code reveal the results.

```
#The conditional probability of NumCompaniesWorked and
YearsWithCurrentManager given YearsAtCompany=(10,20]
>NCW.YWC_given_YAC10to20<-setEvidence(junction, nodes = "YAC",
states = "(10,20]")
>querygrain(NCW.YWC_given_YAC10to20, nodes = c("NCW", "YWC"),
type = "conditional")

Output:
        YWC
NCW           [0,5]     (5,10]    (10,15]    (15,20)
  [0,2]  0.62508829 0.62508829 0.62508829 0.62508829
  (2,4]  0.15637753 0.15637753 0.15637753 0.15637753
  (4,6]  0.08064113 0.08064113 0.08064113 0.08064113
  (6,8]  0.09800199 0.09800199 0.09800199 0.09800199
  (8,10] 0.03989106 0.03989106 0.03989106 0.03989106
```

The probabilities in each column sum up to 1 because they are calculated conditional on the value $YWC$ assumes in that specific column. Additionally, notice how the conditional probabilities $Pr(NCW = [0,2]|YWC = ywc, YAC = (10,20])$ where $ywc \in \{[0,10],(10,20],(20,30],(30,40]\}$ are exactly the same no matter the values of $YWC$. The same is true for the rest of the levels of $NCW$. This happens because $NCW$ is independent from $YWC$ conditional on $YAC$. In other words, the knowledge of the number of companies worked cannot tell the researcher much about the years with current manager when the years at company are known. This is because $NCW$ and $YWC$ are d-separated by $YAC$. If it was to be examined if $A$ and $B$ are d-separated by $C$, the `dsep` command takes $x = A$, $y = B$ and $z = C$ and assesses the d-separation. Thus, the following lines of code prove that $NCW$ and $YWC$ are indeed d-separated by $YAC$.

```
#The probabilities are the same in each row because NCW and YWC are
d-separated by YAC
>dsep(bn.estimation.bayes2, x = "NCW", y = "YWC", z = "YAC")

Output:
[1] TRUE
```

On top of exact inference, approximate inference is also available. In discrete Bayesian Networks, approximate inference is implemented with rejection sampling. Rejection sampling basically generates random observation from the Bayesian Network. Then, the algorithm measures the ratio between the amount of observations which match the event of the probability of interest and the amount of observations matching the conditioning evidence.

`cpquery` function is used from the *bnlear*
*n* package to implement approximate inference. `cpquery` does nothing but returning the

probability of a particular event given certain evidence. For example, what is the probability of a young-low monthly income person given that he is doing his job poorly? The extra argument `n=10^6` refers to the massive increase to the number of random observations to produce a more accurate result. The following lines of code reveal the probability above.

```
#The probability of Age=adult and Mnl=low given JbL=low
>cpquery(bn.estimation.bayes2, event = (Age=='adult') & (Mnl=='low'),
evidence = (JbL=='low'),n=10^6)

Output:
[1] 0.542714
```

Notice that the number 0.542714 is very close to the result produced by the `querygrain` function which is $Pr(Age = adult, Mnl = low|JbL = low) = 0.54265$. Approximate inference can come handy when dealing with complex queries consisting of multiple conditions. For example, what is the probability of an individual to have low monthly income (Mnl=low) and to do bad at his job (JbL=low) given that his total working years range from 20 to 30 years (TWY=(20,30]) and he is old (Age=old) or that he works in the company for 20 to 30 years (YAC=(20,30]). Mathematically that $Pr(JbL = low, Mnl = low|\{Age = old, TWY = (20, 30]\} \cup \{YAC = (20, 30]\})$. The particular probability is displayed with the following lines of code.

```
#The probability of JbL=low and Mnl=low given Age=old and TWY=(20,30] or
YAC=(20,30]
>cpquery(bn.estimation.bayes2, event = (JbL == "low") & (Mnl == "low"),
evidence = ((Age == "old") & (TWY == "(20,30]")) | (YAC == "(20,30]"))

Output:
[1] 0.05016447
```

The resulted probability is equal to about 5%. Theoretically the result is justified. It is highly unlikely for an old (Age=old) and overall experienced (TWY=(20,30]) individual to perform bad at his job (JbL=low) and get a low monthly income (Mnl=low) because he has probably improved the quality of his work over the years. The same is true for an individual which is experienced in working with the particular company (YAC=(20,30]) regardless of his age.

On the other side, old (Age=old) and overall experienced (TWY=(20,30]) or company experienced (YAC=(20,30]) individuals have almost 25% more chances of being excellent at their job (JbL=excellent) and getting a very high monthly income (Mnl=very high). The particular probability is equal to $Pr(JbL = excellent, Mnl = \text{very high}|\{Age = old, TWY = (20, 30]\} \cup \{YAC = (20, 30]\}) = 0.297648$. The lines of code are demonstrated below.

```
#The probability of JbL=excellent and Mnl=very high given Age=old and
TWY=(20,30] or YAC=(20,30]
>cpquery(bn.estimation.bayes2, event = (JbL == "excellent") & (Mnl ==
"very high"),evidence = ((Age == "old") & (TWY == "(20,30]")) | (YAC ==
"(20,30]"))

Output:
[1] 0.297648
```

Another complex query that could be set as a question to the Bayesian network is the probability of an old (Age=old) individual who has worked in many companies (NCW=(6,8]) and has many total working years (TWY=(20,30]) to be at the particular company of the dataset for 0 to 10 years (YAC=[0,10]) compared to the probability of the same individual to be at the particular company for 10 to 20 years (YAC=(10,20]). An individual who has worked many years in different companies it would be sensible to be working only a small amount of years in each of them. Thus, it would be expected that $Pr(YAC = [0, 10]|Age = old, NCW = (6, 8], TWY = (20, 30]) < Pr(YAC = (10, 20]|Age = old, NCW = (6, 8], TWY = (20, 30])$. In other words, the probability of an individual with the particular characteristics to be working less in the company should be larger.

```
#The probability of YAC=[0,10] given Age=old and NCW=(6,8] and TWY=(20,30]
>cpquery(bn.estimation.bayes2, event = (YAC == "[0,10]"),evidence =
((Age == "old") & (NCW=="(6,8]") & (TWY == "(20,30]")))

Output:
[1] 0.6896552

#The probability of YAC=(10,20] given Age=old and NCW=(6,8] and TWY=(20,30]
>cpquery(bn.estimation.bayes2, event = (YAC == "(20,30]"),evidence =
((Age == "old") & (NCW=="(6,8]") & (TWY == "(20,30]")))

Output:
[1] 0.08536585
```

The results confirm the theory, $Pr(YAC = [0, 10]|Age = old, NCW = (6, 8], TWY = (20, 30]) = 0.6896552$ and $Pr(YAC = (10, 20]|Age = old, NCW = (6, 8], TWY = (20, 30]) = 0.08536585$. In conclusion, the difference between the probability of an individual with many working years in multiple companies to be working for 0-10 years to the specific company compared to 10-20 is more than 60%.

#### 3.7.2.6 Graphical Representations in the BN

One of the most advantageous points of a BN is the visual representations of the model. The DAG of a Bayesian Network can be customized through the `graphviz.plot` from the *Rgraphviz* package. `graphviz.plot` takes the `layout` argument which changes the positioning of the arcs and nodes of the BN graph. The `layout` argument takes multiple values such as dots, neato, twopi, circo and fdp. The example BN is graphed with the `layout=fdp` argument. The fdp layout draws undirected graphs using a spring model. It relies on a force-directed approach. The fdp model uses springs only between nodes connected with an edge, and an electrical repulsive force between all pairs of nodes. Also, it achieves a layout by minimizing the forces rather than the energy of the system. The following lines of code produce the DAG2 with `layout=fdp` as demonstrated in Figure 3.7.

```
#DAG2 with fdp layout
>graphviz.plot(DAG2,layout='fdp')
```

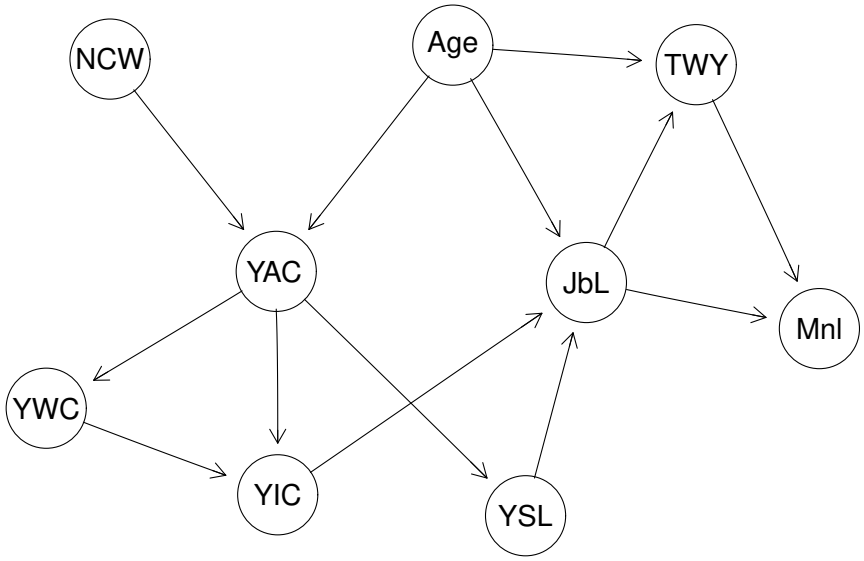



Figure 3.7: The customized Bayesian Network (DAG2) with fdp layout.

Now assume that one wanted to emphasize the path $NCW \rightarrow YAC \rightarrow YWC$ because $NCW$ and $YWC$ d-separated by $YAC$. Then, the researcher should first make all the nodes, arcs and labels to grey. In order to do that, all the nodes and arcs are listed to an R object called `grey_graph`. Additional arguments are `col = "grey"` and `textCol = "grey"` to customize the color of all the nodes, arcs and labels to grey. After that, `DAG2` and `grey_graph` (through the `highlight` argument) are used as input in the `graphviz.plot` function. The result is saved in `grey_DAG2`.

The following lines of code successfully create the `grey_graph`.

```
#Setting the color of all arcs, nodes and labels to grey
>grey_graph <- list(nodes = nodes(DAG2), arcs = arcs(DAG2), col = "grey",
textCol = "grey")

#Changing the color of all arcs and nodes of DAG2 to grey
>grey_DAG2<-graphviz.plot(DAG2,layout='fdp',highlight = grey_graph)
```

Then `grey_DAG2` is used as input in `nodeRenderInfo`, `nodeRenderInfo` and `renderGraph` functions from the *graph* package. The `nodeRenderInfo` function will be used to change the color of the arcs of the path of interest ($NCW \rightarrow YAC \rightarrow YWC$) through the `col` argument. The `nodeRenderInfo` will be used to change the color of the nodes and labels of interest ($NCW, YAC, YWC$) through the `col` and `textcol` arguments, respectively. Finally, `renderGraph` will be used to plot the final customized form of the DAG2 with the $NCW \rightarrow YAC \rightarrow YWC$ path black (Figure 3.8).

```
#Changing the color and width of arcs of the path NCW->YAC->YWC of DAG2 to
black
>edgeRenderInfo(grey_DAG2) <-list(col = c("NCW~YAC" = "black", "YAC~YWC" =
"black"))

#Changing the color of nodes NCW,YAC,YWC of DAG2 to black
```

```
>nodeRenderInfo(grey_DAG2) <-list(col = c("NCW" = "black", "YAC" =
"black","YWC"="black"),textCol = c("NCW" = "black", "YAC" = "black",
"YWC"="black"))

#Plotting the customized DAG2
>custom_DAG2<-grey_DAG2
>renderGraph(custom_DAG2)
```

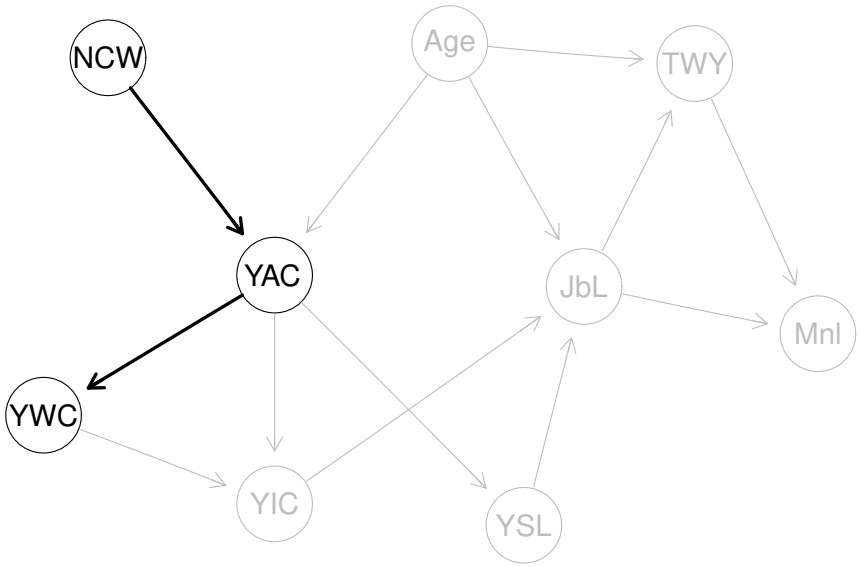



Figure 3.8: The final customized Bayesian Network (DAG2) with fdp layout and the path $NCW \rightarrow YAC \rightarrow YWC$ black colored.

Regarding the plotting of the conditional probability distributions, the `bn.fit.barchart` and `bn.fit.dotplot` functions from the *bnlearn* package are used to plot the bar charts and dot plots, respectively. For example, the bar plot and dot plot of the probability $Pr(YIC|YWC, YAC)$ are displayed in Figure 3.9 and Figure 3.10, respectively. Both `bn.fit.barchart` and `bn.fit.dotplot` functions take the estimation of a particular node ($YSL$) and produces the graph of the probability given its parents ($YWC, YAC$). In both Figure 3.9 and Figure 3.10, the values in the green boxes correspond to the levels of $YWC$ and the orange bar to the levels of $YAC$.

```
#The bar chart of the conditional probability of YearsInCurrentRole given
YearsWithCurrentManager and YearsAtCompany
>bn.fit.barchart(bn.estimation.bayes2$YIC, main = "YIC",xlab =
"Pr(YIC | YWC,YAC)", ylab = "")

#The dot plot of the conditional probability of YearsInCurrentRole
given YearsWithCurrentManager and YearsAtCompany
>bn.fit.dotplot(bn.estimation.bayes2$YIC, main = "YIC",xlab =
"Pr(YIC | YWC,YAC)", ylab = "")
```

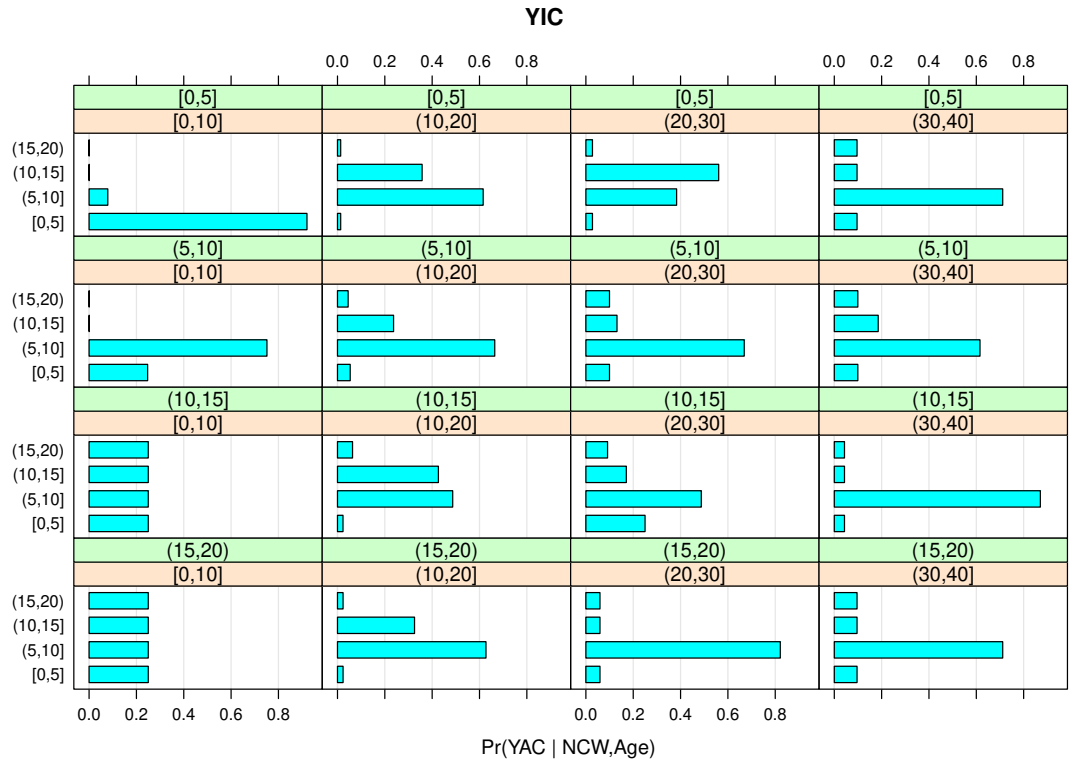



Figure 3.9: The bar chart of the conditional probability distribution of $Pr(YIC|YWC, YAC)$.

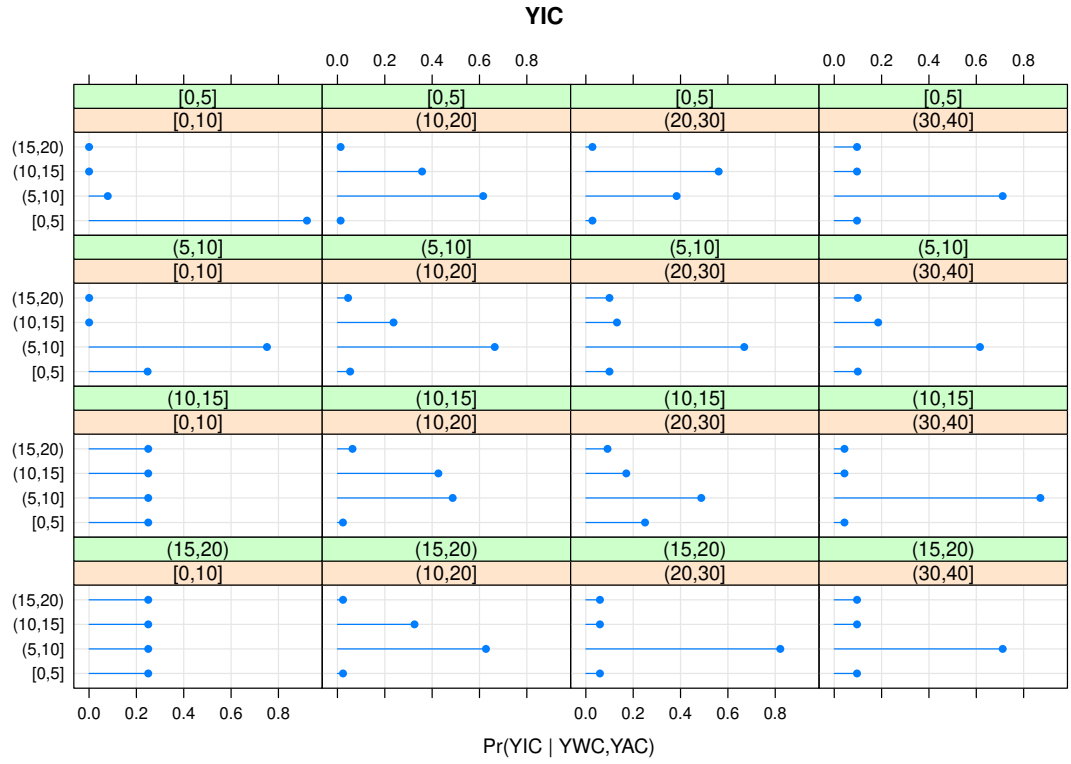



Figure 3.10: The dot plot of the conditional probability distribution of $Pr(YIC|YWC, YAC)$.

## 3.8 Conclusions

In this chapter, the theory behind Bayesian Networks was explicitly analyzed and an R example of discrete BN is demonstrated. At the beginning, it was stated that Bayesian SEM approach is more flexible than the classical SEM approach. The fundamental concept of Bayesian Networks is that the researcher chooses prior distributions for the parameters of a model through the non-informative and informative priors and calculates the posterior distributions. The researcher first specifies the nodes and arcs of the Directed Acyclic Graph based on theory or, in this case of this thesis, an already valid SEM model. Before the Bayesian Network model can be used for inference, the researcher evaluates it through conditional independence tests and network scores. Two of the most popular conditional independence tests are implemented through Mutual Information and Bayesian Information Criterion. At the same time, the network score of the Bayesian Network is calculated through Bayesian Information Criterion and Bayesian Dirichlet Equivalent. If it is judged that the model is incapable of fitting the data, based on the conditional independence tests and network scores, then the model is re-specified to yield a better fit. When it is verified that the model is specified, the estimation of the parameters take place with maximum likelihood or bayes estimation. Finally, the Bayesian Network is now ready to be used for inference. The inference is done through a set of queries which the BN is called to give answers to. Conditional Probability Tables are utilized to conduct exact or approximate inference. Exact inference uses a crafted tree called junction to give answers to the queries of interest faster and preciser. Exact inference can come handy when the researcher wants answers to simple queries. On the other hand, approximate inference is usually done through Monte Carlo simulations which generate random observations which can be used for to calculate the desired approximate estimates of the conditional probabilities. Monte Carlo simulations allow for very complex queries but lack of computational power. Approximate inference is often paired with likelihood weighting which generates random observations so that all of them match the evidence. Then, the algorithm weights them again, but according to the desired conditional probability pointed out from the query. Essentially likelihood weighting yields better results.

Regarding the lab in R, the Company dataset is used as input to construct a Bayesian Network based on the SEM specification. But network scores and conditional independence tests pointed in the following Bayesian Network expressed in string form: "$[Age][NCW][YAC|Age:NCW][YSL|YAC][YWC|YAC][YIC|YAC:YWC][JbL|Age:YIC:YSL][TWY|Age:JbL][Mnl|JbL:TWY]$", where $Age = Age$, $JbL = JobLevel$, $Mnl = MonthlyIncome$, $NCW = NumCompaniesWorked$, $TWY = TotalWorkingYears$, $YAC = YearsAtCompany$, $YIC = YearsInCurrentRole$, $YSL = YearsSinceLastPromotion$ and $YWC = YearsWithCurrentManager$. During the inference process, several queries were given into the network and some of the most remarkable conclusions are the following:

1. The marginal probability of an individual to have low job level among the people of the dataset, $Pr(JbL = low)$, is equal to 0.71604353. However, the marginal probability of an individual to have low job level ($Pr(JbL = low)$) given that he/she is an adult, $Pr(JbL = low|Age = adult)$ is equal to 0.83167462. In other words, there is almost 12% more chance for adult people to have low job level compared to the whole dataset. Additionally, the marginal probability of an individual to have good job level among the people of the dataset, $Pr(JbL = good)$, is equal to 0.22560846. This probability drops to 0.15725407 given that the individual is an adult ($Pr(JbL = good|Age = adult)$=0.15725407). In other words, there is about 7% less chance for adult to perform good at their job compared to the whole dataset.

2. The marginal probabilities of an individual having low, good and excellent job level given that his/her monthly income is very high ($Pr(Mnl = \text{very high})$) are 0.005286003, 0.440542146 and 0.554171851, respectively. Mathematically, $Pr(JbL = low|Mnl = \text{very high}) = 0.005286003$, $Pr(JbL = good|Mnl = \text{very high}) = 0.440542146$ and $Pr(JbL = excellent|Mnl = \text{very high}) = 0.554171851$. In other words, the cumulative probability of an individual being good or excellent to his job given that his monthly income is very high is equal to $0.440542146 + 0.554171851 = 0.994714$. Therefore, very good paid individual are in almost every case good or excellent at their job.

3. The probability of an individual having low job level ($Pr(JbL = low)$) and low monthly income ($Pr(Mnl = low)$) given that he/she is old ($Pr(Age = old)$) and his/her total working years range from 20 to 30 ($Pr(TWY = (20, 30])$) or his/her years at company range from 20 to 30 ($Pr(YAC = (20, 30])$) is equal to 0.04729149. Mathematically, $Pr(JbL = low, Mnl = low|\{Age = old, TWY = (20, 30]\} \cup \{YAC = (20, 30]\}) = 0.04729149$. On the other hand, the probability of an individual having excellent job level ($Pr(JbL = excellent)$) and very high monthly income ($Pr(Mnl = \text{very high})$) given that he/she is old ($Pr(Age = old)$) and his/her total working years range from 20 to 30 ($Pr(TWY = (20, 30])$) or his/her years at company range from 20 to 30 ($Pr(YAC = (20, 30])$) is equal to 0.2714171. Mathematically, $Pr(JbL = excellent, Mnl = \text{very high}|\{Age = old, TWY = (20, 30]\} \cup \{YAC = (20, 30]\}) = 0.2714171$. In other words, the probability of an individual old in age, with many working years in general and in the company, to be excellent at his job and getting a very high monthly income is about 23% higher than for the same individual to be bad at his job and getting a low monthly income.

4. The probability of an individual to be working 0 to 10 years at the company ($Pr(YAC = [0, 10])$) given that he/she is old ($Pr(Age = old)$), the number of companies he/she has worked already range from 6 to 8 ($Pr(NCW = (6, 8])$) and the total working years of his/her life range from 20 to 30 ($Pr(TWY = (20, 30])$), $Pr(YAC = [0, 10]|Age = old, NCW = (6, 8], TWY = (20, 30]) = 0.605835$. On the other hand, the probability of an individual to be working 10 to 20 years at the company ($Pr(YAC = (10, 20])$) given that he/she is old ($Pr(Age = old)$), the number of companies he/she has worked already range from 6 to 8 ($Pr(NCW = (6, 8])$) and the total working years of his/her life range from 20 to 30 ($Pr(TWY = (20, 30])$), $Pr(YAC = (10, 20]|Age = old, NCW = (6, 8], TWY = (20, 30]) = 0.159334$. In other words, the probability of an individual old in age, who has worked for many years and companies, to be for 0 to 10 years in the particular company is about 44% higher than for the same individual to be for 10 to 20 years in the particular company.

# Bibliography


Abdi, Hervé and Lynne J Williams (2010). "Principal component analysis". In: *Wiley interdisciplinary reviews: computational statistics* 2.4, pp. 433–459.

Agresti, Alan (2017). *Statistical methods for the social sciences (5th Edition)*. Pearson.

Agresti, Alan and Christine Franklin (2018). *Statistics the Art and Science of Learning from data*. Pearson Education Limited.

Anderson, James G (1973). "Causal models and social indicators: Toward the development of social systems models". In: *American Sociological Review*, pp. 285–301.

Anderson, Theodore W and Herman Rubin (1956). "Statistical inference in factor analysis". In: *Proceedings of the third Berkeley symposium on mathematical statistics and probability*. Vol. 5, pp. 111–150.

Bagozzi, Richard P (1980). *Causal models in marketing*. Wiley.

Beaujean, A Alexander (2014). *Latent variable modeling using R: A step-by-step guide*. Routledge.

Biddle, Bruce J and Marjorie M Marlin (1987). "Causality, confirmation, credulity, and structural equation modeling". In: *Child development*, pp. 4–17.

Birnbaum, A Lord (1968). "Some latent trait models and their use in inferring an examinee's ability". In: *Statistical theories of mental test scores*.

Blalock Jr, Hubert M (1961). "Correlation and causality: The multivariate case". In: *Social Forces* 39.3, pp. 246–251.

Bohrnstedt, George W and T Michael Carter (1971). "Robustness in regression analysis". In: *Sociological methodology* 3, pp. 118–146.

Bohrnstedt, George W and David Knoke (1982). *Statistics for social data analysis*. Tech. rep.

Borsboom, Denny (2008). "Latent variable theory". In:

Browne, Michael W et al. (1993). "Testing structural equation models". In:

Clark, Michael (n.d.). *Graphical & and Latent Variable Modeling*. URL: `https://m-clark.github.io/sem/`. (Accessed: 2020-12-25).

Cudeck, Robert et al. (2001). *Structural equation modeling: Present and future: A Festschrift in honor of Karl Jöreskog*. Scientific Software International.

Delucchi, Michael (2006). "The efficacy of collaborative learning groups in an undergraduate statistics course". In: *College Teaching* 54.2, pp. 244–248.

Duncan, Otis Dudley (1966). "Path analysis: Sociological examples". In: *American journal of Sociology* 74, pp. 119–137.

Duncan, T et al. (1999). "Introduction to Latent Variable Growth Curve Modeling. London". In:

Embretson, Susan E and Steven Paul Reise (2000). *Item response theory for psychologists. Maheah*.

Epskamp, Sacha and S Stuber (2014). *semPlot: Path diagrams and visual analysis of various SEM packages' output. R package version 1.0. 1*.

Fox, John (2006). "Teacher's corner: structural equation modeling with the sem package in R". In: *Structural equation modeling* 13.3, pp. 465–486.

Goldberg, Lewis R (1990). “An alternative" description of personality": the big-five factor structure.” In: *Journal of personality and social psychology* 59.6, p. 1216.

Goodman, Leo A (1974). “Exploratory latent structure analysis using both identifiable and unidentifiable models”. In: *Biometrika* 61.2, pp. 215–231.

Hair, J.F. et al. (2013). *Multivariate Data Analysis*. Always learning. Pearson Education Limited. ISBN: 9781292021904.

Hambleton, Ronald K and Hariharan Swaminathan (2013). *Item response theory: Principles and applications*. Springer Science & Business Media.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). “Overview of supervised learning”. In: *The elements of statistical learning*. Springer, pp. 9–41.

Heise, David R (1969). “Problems in path analysis and causal inference”. In: *Sociological methodology* 1, pp. 38–73.

Heise, David R (1975). *Causal analysis*. John Wiley & Sons.

Hershberger, Scott L (2003). “The growth of structural equation modeling: 1994-2001”. In: *Structural Equation Modeling* 10.1, pp. 35–46.

Hetherington, John (2000). “Role of theory and experimental design in multivariate analysis and mathematical modeling”. In: *Handbook of applied multivariate statistics and mathematical modeling*. Elsevier, pp. 37–63.

Howe, William Gerow (1955). *Some contributions to factor analysis*. Tech. rep. Oak Ridge National Lab., Tenn.

Hoyle, Rick H (2012). *Handbook of structural equation modeling*. Guilford press, pp. 1–70.

Ihaka, Ross and Robert Gentleman (1996). “R: a language for data analysis and graphics”. In: *Journal of computational and graphical statistics* 5.3, pp. 299–314.

James, Gareth et al. (2013). *An introduction to statistical learning*. Vol. 112. Springer.

Jensen, Finn V and Thomas Dyhre Nielsen (2007). *Bayesian networks and decision graphs*. Vol. 2. Springer.

Jo, Booil and Bengt O Muthén (2001). “Modeling of intervention effects with noncompliance: A latent variable approach for randomized trials”. In: *New developments and techniques in structural equation modeling*. Psychology Press, pp. 77–108.

Joereskog, Karl Gustav (1963). *Statistical estimation in factor analysis*. Almqvist & Wiksell.

Jöreskog, Karl G (1969). “A general approach to confirmatory maximum likelihood factor analysis”. In: *Psychometrika* 34.2, pp. 183–202.

Jöreskog, Karl G (1971). “Statistical analysis of sets of congeneric tests”. In: *Psychometrika* 36.2, pp. 109–133.

Kaplan, David (2002). “Methodological advances in the analysis of individual growth with relevance to education policy”. In: *Peabody Journal of Education* 77.4, pp. 189–215.

Kaplan, David and Sarah Depaoli (2012). “Bayesian structural equation modeling.” In:

Kapsali, Maria (2020). “Introduction To Structural Equation Modeling With Applications In Lisrel and R”. Diploma Thesis.

Keesling, J Ward (1972). “Maximum likelihood approaches to causal analysis”. In: *Ph. D. dissertation. Department of Education: University of Chicago*.

Kline, Rex B. (2015). *Principles and Practice of Structural Equation Modeling, Fourth Edition*. Methodology in the Social Sciences. Guilford Publications, pp. 7–100. ISBN: 9781462523351.

Korb, Kevin B and Ann E Nicholson (2010). *Bayesian artificial intelligence*. CRC press.

Land, Kenneth C (1969). “Principles of path analysis”. In: *Sociological methodology* 1, pp. 3–37.

Lawley, Derrick Norman and Albert Ernest Maxwell (1971). *Factor analysis as statistical method*. Tech. rep.

Lawley, DN (1958). "Estimation in factor analysis under various initial assumptions". In: *British journal of statistical Psychology* 11.1, pp. 1–12.

Lee, Sik-Yum (2007). *Handbook of Latent Variable and Related Models: Handbook of Computing and Statistics With Applications, Volume 1*. Elsevier Science & Technology.

MacCallum, Robert C (1995). "Model specification: Procedures, strategies, and related issues." In: pp. 16–36.

MacDonald, KI (1977). *Path analysis*. Vol. 2. New York: Wiley & Sons.

McClendon, JM (1994). *Multiple regression and causal analysis. Prospect Heights, IL*.

McLachlan, Geoffrey J, Sharon X Lee, and Suren I Rathnayake (2019). "Finite mixture models". In: *Annual review of statistics and its application* 6, pp. 355–378.

Muthén, Bengt (1984). "A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators". In: *Psychometrika* 49.1, pp. 115–132.

Muthén, Bengt and Tihomir Asparouhov (2012). "Bayesian structural equation modeling: a more flexible representation of substantive theory." In: *Psychological methods* 17.3, p. 313.

Muthén, Bengt and Katherine Masyn (2005). "Discrete-time survival mixture analysis". In: *Journal of Educational and Behavioral statistics* 30.1, pp. 27–58.

Muthén, Bengt O (1994). "Multilevel covariance structure analysis". In: *Sociological methods & research* 22.3, pp. 376–398.

Muthén, Bengt O (2002). "Beyond SEM: General latent variable modeling". In: *Behaviormetrika* 29, pp. 81–117.

Nagarajan, Radhakrishnan, Marco Scutari, and Sophie Lèbre (2013). "Bayesian networks in r". In: *Springer* 122, pp. 125–127.

Olobatuyi, Moses E (2006). *A user's guide to path analysis*. University Press of America, pp. 53–61.

Parkerson, Jo A et al. (1984). "Exploring causal models of education achievement." In: *Journal of Educational Psychology* 76.4, pp. 638–646.

Pearl, Judea (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Pearson, Egon Sharpe (1938). *Karl Pearson: An appreciation of some aspects of his life and work*. Cambridge University Press Cambridge.

Pedhazur, Elazar J, Fred Nichols Kerlinger, et al. (1982). *Multiple regression in behavioral research*. Holt, Rinehart, and Winston.

Pf, Lazarsfeld and Henry NW (1968). *Latent structure analysis*.

Rosseel, Yves (2012). "Lavaan: An R package for structural equation modeling and more. Version 0.5–12 (BETA)". In: *Journal of statistical software* 48.2, pp. 1–36.

Schumacker, R.E. and R.G. Lomax (2016). *A Beginner's Guide to Structural Equation Modeling*. Routledge. ISBN: 9781138811935.

Scutari, Marco and Jean-Baptiste Denis (2021). *Bayesian networks: with examples in R*. CRC press.

Shalizi, Cosma (2013). *Advanced data analysis from an elementary point of view*.

Sijtsma, Klaas and Ivo W Molenaar (2002). *Introduction to nonparametric item response theory*. Vol. 5. sage.

Skrondal, Anders and Sophia Rabe-Hesketh (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. Crc Press.

Spearman, Charles (1927). *The abilities of man*. Vol. 6. Macmillan New York.

Spirtes, Peter et al. (2000). *Causation, prediction, and search*. MIT press.

Stevens, Stanley Smith et al. (1946). "On the theory of scales of measurement". In: 103, pp. 677–680.

Thurstone, Louis Leon (1935). “The vectors of mind: Multiple-factor analysis for the isolation of primary traits.” In: pp. 226–231.

Wiley, David E (1973). “The identification problem for structural equation models with unmeasured variables”. In: *Structural equation models in the social sciences*, pp. 69–83.

Wold, Herman (1954). “Causality and econometrics”. In: *Econometrica: Journal of the Econometric Society* 22.2, pp. 162–177.

Wolfle, Lee M (2003). “The introduction of path analysis to the social sciences, and some emergent themes: An annotated bibliography”. In: *Structural Equation Modeling* 10.1, pp. 1–34.

Wright, Sewall (1918). “On the nature of size factors”. In: *Genetics* 3, pp. 367–374.

Wright, Sewall (1921). “Correlation and Causation. JouMal of”. In: *Agricultural Research* 20, pp. 557–585.

Wright, Sewall (1934). “The method of path coefficients”. In: *The annals of mathematical statistics* 5.3, pp. 161–215.