Andrew Fagundes

Department of Information Technology, Western Governors University

D206 Data Cleaning

Professor Eric Straw

April 15, 2024

Table of Contents

[Part I: Research Question and Variables 3](#_Toc164090074)

[A: Research Question 3](#_Toc164090075)

[B: Variable List and Data Types 3](#_Toc164090076)

[Andrew Fagundes 8](#_Toc164090077)

[C1: Methods/Functions Used 8](#_Toc164090078)

[C2: Reasoning 10](#_Toc164090079)

[C3: Programming Language 11](#_Toc164090080)

[C4: Code 12](#_Toc164090081)

[Part III: Data Cleaning (Treatment) 14](#_Toc164090082)

[D1: Detection Results 14](#_Toc164090083)

[D2: Treatment 16](#_Toc164090084)

[D3: Summarized Work 20](#_Toc164090085)

[D4: Code 21](#_Toc164090086)

[D5: Treated/Cleaned CSV 21](#_Toc164090087)

[D6: Advantages/Disadvantages of Treatments 22](#_Toc164090088)

[D7: Challenges in Data Analysis 23](#_Toc164090089)

[Part IV: Principal Component Analysis (PCA) 23](#_Toc164090090)

[E1: PCA Loadings 23](#_Toc164090091)

[E2: Which PCAs to Keep? 24](#_Toc164090092)

[E3: Benefits of PCA 26](#_Toc164090093)

[References (No third party code references used) 27](#_Toc164090094)

[Table of Figures 28](#_Toc164090095)

Part I: Research Question and Variables

## A: Research Question

For the medical data set, my research question is the following: “What contributing factors lead to increased hospital readmissions?” This question can help a hospital determine the main predictors for patient readmission so that a preventative approach can be taken to reduce readmission in the future. Even if some factors are uncontrollable, the hospital can use this data to focus on the factors they can control.

## B: Variable List and Data Types

The following table contains all variables included in the original dataset along with their individual data types and an example.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example |
| CaseOrder | Qualitative | Variable used to define order of cases | 0, 1, 2 |
| Customer\_id | Qualitative | An ID that defines a specific patient | Z919181 |
| Interaction | Qualitative | An ID related to patient transactions | d2450b70-0337-4406-bdbb-bc1037f1734c |
| UID | Qualitative | Unique ID related to patient transactions and other procedures | 176354c5eef714957d486009feabf195 |
| City | Qualitative | Patient’s city of residence | Marianna |
| State | Qualitative | Patient’s state of residence | FL |
| County | Qualitative | Patient’s county of residence | Jackson |
| Zip | Qualitative | Patient’s zipcode | 32446 |
| Lat | Quantitative | Latitude of patient’s billing address | 30.84513 |
| Lng | Quantitative | Longitude of patient’s billing address | -85.22907 |
| Population | Quantitative | Population within 1 mile radius of patient | 11303 |
| Area | Qualitative | Area type: rural, urban, suburban | Urban |
| Timezone | Qualitative | Time zone of patient residence | America/Chicago |
| Job | Qualitative | Job of patient or insurance holder | Community development worker |
| Children | Quantitative | Number of children in patient’s household | 3 |
| Age | Quantitative | Age of patient | 51 |
| Education | Qualitative | Highest degree earned by patient | Some College, 1 or More Years, No Degree |
| Employment | Qualitative | Employment status of patient | Full Time |
| Income | Quantitative | Annual income of patient | 46805.99 |
| Marital | Qualitative | Marital status of patient or insurance holder | Married |
| Gender | Qualitative | Customer self-identification of male, female, or nonbinary | Female |
| ReAdmis | Qualitative | Whether the patient was readmitted within one month of release | No |
| VitD\_levels | Quantitative | Patient’s vitamin D levels as measured in ng/mL | 18.99463952 |
| Doc\_visits | Quantitative | Number of times patient was visited in hospital by doctor | 4 |
| Full\_meals\_eat | Quantitative | Number of full meals eaten by patient in hospital (0=partial meals) | 2 |
| VitD\_supp | Qualitative | Number of times Vitamin D supplements were given to patient | 1 |
| Soft\_drink | Qualitative | Whether patient drinks 3 or more soft drinks in a day | No |
| Initial\_admin | Qualitative | Means by which the patient was admitted to the hospital (emergency, elective, observation) | Emergency Admission |
| HighBlood | Qualitative | Whether patient has high blood pressure (yes, no) | Yes |
| Stroke | Qualitative | Whether patient has had a stroke (yes, no) | No |
| Complication\_r | Qualitative | Level of patient’s complication risk assessed by doctor (high, medium, low) | High |
| Overweight | Qualitative | Whether patient is considered overweight based on age, gender, and height (yes, no) | 1 |
| Arthritis | Qualitative | Whether patient has arthritis (yes, no) | No |
| Diabetes | Qualitative | Whether patient has diabetes (yes,no) | No |
| Hyperlipidemia | Qualitative | Whether patient has hyperlipidemia (yes, no) | No |
| BackPain | Qualitative | Whether the patient has chronic back pain (yes, no) | Yes |
| Anxiety | Qualitative | Whether the patient has an anxiety disorder (yes, no) | 1 |
| Allergic\_rhini | Qualitative | Whether the patient has allergic rhinitis (yes, no) | Yes |
| Reflux\_esophag | Qualitative | Whether the patient has reflux esophagitis (yes, no) | No |
| Asthma | Qualitative | Whether patient has asthma (yes, no) | Yes |
| Services | Qualitative | Primary service patient received while hospitalized (blood work, intravenous, CT scan, MRI) | Intravenous |
| Initial\_days | Quantitative | Length of patient’s stay at hospital | 15.12956221 |
| TotalCharge | Quantitative | Average daily amount billed to patient | 4214.905346 |
| Additional\_charges | Quantitative | Average amount charged to patient for other procedures, treatments, medicines, anesthesiology, etc. | 17612.99812 |
| Item1 | Qualitative | Timely admission (1 = most important, 8 = least important) | 3 |
| Item2 | Qualitative | Timely treatment (1 = most important, 8 = least important) | 4 |
| Item3 | Qualitative | Timely visits (1 = most important, 8 = least important) | 3 |
| Item4 | Qualitative | Reliability (1 = most important, 8 = least important) | 4 |
| Item5 | Qualitative | Options (1 = most important, 8 = least important) | 4 |
| Item6 | Qualitative | Hours of treatment (1 = most important, 8 = least important) | 4 |
| Item7 | Qualitative | Courteous staff (1 = most important, 8 = least important) | 3 |
| Item8 | Qualitative | Evidence of active listening from doctor (1 = most important, 8 = least important) | 3 |
|  |  |  |  |

Figure 1: Variable List and Data Types

Andrew Fagundes

## C1: Methods/Functions Used

In this section, we will explore the data and determine the methods that will detect for duplicates, null values, outliers, and the re-expression of categorical variables. We begin by importing the Pandas and Numpy libraries. Following this, we import the CSV file as a data frame by using the following code: df = pd.read\_csv(‘medical\_raw\_data.csv’, na\_values = ‘NA’, index\_col = 0). Following this, I print the data frame as output and display all the columns for review. Once the data frame is saved, I used the “.info()” function to investigate the columns, non-null counts, and data types. Then, I review each variable’s values to determine if there is any data that looks incorrect, data that should be standardized, or data is being used in the proper format. To conduct this review, I start by checking the uniqueness constraints of columns CaseOrder, Customer\_id, Interaction, and UID to make sure they contain unique values. I do this by pulling out the columns I want to assess the uniqueness of and store them in a new data frame so I can use a for loop to quickly iterate through them and get the information I desire. These four variables all had 10,000 unique values. Then, I check for duplicate values in the data set with the following code: df.duplicated().value\_counts(). This function provides the output of “False 10000” since there are no duplicates and 10,000 is the total rows in the data set. To verify this, I compare the shape of the data set with the output. This is done with the .shape command, which illustrates the total number of rows and columns in the data set.

To discover the variables with null values, I use the df.isnull().sum() method. The summary illustrates that the variables Children, Age, Income, Soft\_drink, Overweight, Anxiety, and Initial\_days have null values. Once I know the variables with null values, I check the statistical summary of the quantitative variables with missing data and without by using the .describe() function. This provides me with the value count, mean, standard deviation, minimum, 25th percentile, 50th percentile, 75th percentile, and maximum values. After completing this, we visualize the null values using histograms to verify gaps in the bar value. Following this, I use the missingno’s matrix to create a missing data matrix and plt.title() and .show() to customize the matrix.

For outlier detection, we use the data in all quantitative variables and use the Seaborn package to generate boxplots for all quantitative variables described. Each boxplot will illustrate if there are any outliers. Other outlier detection methods such as histograms and z-scores were also used. Histograms are used to determine the distribution of each variable. Since we don’t know if the outliers found are factual errors, we extract them, save as their own data frame, and then remove them from the original data frame. We will use the z-scores to extract all records whose z-score is greater than 3 or less than -3. To do this, we import the Scipy.stats function. Two variables, ‘Overweight’ and ‘Anxiety’, were excluded from this since they needed to be re-expressed. These were converted from 0’s and 1’s to Yes and No for data uniformity. Afterwards, data profiling would be used to determine that the values are acceptable.

To determine the re-expression of variables, I start by checking for incorrect datatypes using .info() on the data frame containing the medical data to get a summary of the columns and their data types. To check for categorical variables that need re-encoding, I use .value\_counts(). This returns the total number of rows for each option, which is optimal for scanning potential problems. There were two variables, ‘Overweight’ and ‘Anxiety’, that were using 0’s and 1’s to represent Yes and No in a categorical way, while there was one variable, ‘Soft\_drink’, that was using Yes and No. To treat the missing values for the three categorical variables, we will calculate the percentage of each category (0 and 1) and impute missing values with the highest percentage category (Tamboli 2021). The rest of the Boolean variables in the data set were expressed as Yes or No. The re-expression takes place so that all these variables represent the same categories. Children, Age, Income, and Initial\_days are quantitative variables while Overweight, Soft\_drink and Anxiety are qualitative as they expressed Yes and No. However, Soft\_drink is not re-expressed to represent Yes or No. To fix this, ordinal encoding is used to re-express “Yes” as 1 and “No” as 0 followed by filling in the missing values (Middleton, 2024). The ordinal encoding process involves replicating the variable by replacing the categorical values with numeric values. This established by setting up a dictionary for converting the categorical values to numeric values. Once the dictionary is created, we use the .replace(dict\_soft\_drink, inplace=True) function to replace the variable’s values and store in the existing data frame. Finally, we run .info() to verify that no missing values remain.

## C2: Reasoning

Based on the functions and methods used from the pandas library, it was possible to review the data set’s variables and determine whether there were any duplicated values, missing values and outliers. Along with pandas, packages were used such as seaborn and scikit to create the boxplots and PCA analysis.

The approach for detecting duplicates was used because it made it possible to observe the entire data set for duplicate values. All the columns along with null values were illustrated with this approach, which makes it easy to understand the big picture.

To find missing values, we use the ‘.isnull()’ function on the variables. If the results from each column contain any missing data, the column name is collected into a list. This function can be used for finding missing values for numeric, object and datetime objects.

Boxplots from the Seaborn package were used for outlier detection. Box plots provide self-explanatory charts with left and right whiskers and any dots that appear beyond these whiskers indicate outliers.

For the expression of variables, data uniformity is important to keep things consistent and clear. For the variables with 0/1 values, these needed to provide a Yes or No, which is why they were re-expressed from a numerical to a categorical datatype with Yes or No after the missing values were imputed.

The ‘.describe()’ function will provide a summary of quantitative data, which illustrates the data through statistical measures such as the mean, standard deviation, and min/max. The ‘.value\_counts()’ allows me to understand qualitative variables by understanding how many observations were collected per option and what the names of those options are.

## C3: Programming Language

I used Python and the packages Pandas, NumPy, Missingno, Matplotlib, Sklearn, and Seaborn. Numpy provides mathematical functions that may be necessary for transforming the data, and pandas provides a logical structure called a data frame, which allows us to store the CSV in a similar format to a spreadsheet with functions to transform or standardize the data as needed. Once the data is cleaned, a Principal Component Analysis will be performed, and this makes use of the PCA method from SKlearn to perform the PCA analysis. Seaborn is used to graph any scree plots used as part of this PCA.

This decision was made because Python is one of the most popular programming languages in data analytics. One reason I chose Python is because it is valuable for complicated statistical tests and communicating the findings with data visualization (https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html). Python code is typically easier to write and handles large data sets well. It is also intuitive and has a smoother learning curve compared to R, along with a variety of libraries and packages that help with coding. I wrote my code in Jupyter Notebook because it is easy to attach the notebook file for the assessment.

## C4: Code

For full code, see code attached. This is in the following file: *‘D206 Data Cleaning FINAL.ipynb’*

A white rectangular object with blue text

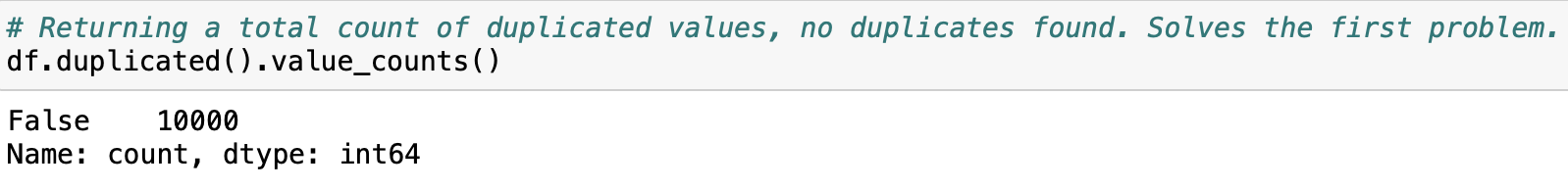
Description automatically generated

Figure 2: Shape of data frame and total duplicated values that exist.

A screenshot of a computer

Description automatically generated

Figure 3: Summary of variables with null values.

A screenshot of a computer code

Description automatically generated

Figure 4: Histograms of quantitative variables.

# Part III: Data Cleaning (Treatment)

## D1: Detection Results

The first issue solved were the duplicated values. Based on the output of the code, there were no duplicated values in the data set. The output was derived from the df.duplicated().value\_counts() function, which resulted in ‘False 10000’.

To determine if any missing values exist in the variables of the dataset, we use the following function, df.isnull().sum(). 7 columns have missing values: children (2588), age (2414), income (2464), soft\_drink (2467), overweight (982), anxiety (984) and initial\_days (1056). *A close-up of a bar code

Description automatically generated*

Figure 5: Missing Data Matrix from missingno.

Outliers were found in the data set. For the quantitative variables, 4 outliers were found in the Children variable, none were found in the Age and Initial\_days variables and the variable with the most outliers was Income. Income’s 75th percentile quartile is 46466.797500 and the outliers were approximately 2500 records. This was done through boxplots from the Seaborn package.

Children and Age are considered discrete quantitative variables because they can only be numbers. Income and Initial\_days are considered continuous quantitative variables because income can be whole numbers or contain decimals. Overweight, Soft\_drink and Anxiety are considered nominal qualitative variables because they are either yes or no. We run the following code to determine statistical info on quantitative variables with missing data: df[[‘Children’, ‘Age’, ‘Income’, ‘Initial\_days’]].describe(). This will show the total count, mean, standard deviations, minimum, maximum, and interquartile range.

Importing the seaborn package allows us to create histograms of the quantitative variables Children, Age, Income, Initial\_days to visually analyze their distribution. Once the package is imported, we plot the histograms. The following insights can be made with these graphs: both Children and Income are positively skewed to the right, Age is uniformly distributed, and Initial\_days has a bimodal distribution. For Income, Children, Initial\_days, and Age variables, we will treat missing values by imputation. Due to the skews of the distributions, we will use the median value for Income, Children, Initial\_days and the mean value for Age. The median value was used because the distributions were skewed to the right and bimodal. Meanwhile, the mean value was used for Age due to its uniform distribution. Once this is done, we will check summary statistics again to compare with previous summary. We will also plot histograms again to check for skewness.

Next, we focus on the remaining categorical variables: Overweight, Anxiety, and Soft\_drink. We detect amount of missing values for each variable with isnull().value\_counts(). After the missing data treatment phase, we then set to detect if any numerical columns only had 0/1’s as values. This indicated that they could be represented as Yes/No to become uniform with the rest of the Boolean columns. The variables (columns) ‘Overweight’ and ‘Anxiety’ were both found to only include 0/1’s with .value\_counts().

## D2: Treatment

For duplicate values, we used the df.duplicated().value\_counts() function to determine if there were any duplicates in the data set. This function worked best because it explored the entire set for total duplicate values. Since there were none, nothing needed to be done here.

For the missing values, imputation was performed. This was chosen because we do not want to reduce and damage the data set size and accuracy for all other rows. To determine the type of imputation, histograms are created to plot the distribution of a numeric variable’s values. This also checks for the skewness so we can determine which statistical value will replace the missing value. For the variables Income, Children, and Initial\_days, the median was used as the reference value because the distributions were skewed to the right for Income and Children along with Initial\_days illustrating a bimodal distribution. For Age, the mean was used as the reference value since it was uniformly distributed (Middleton 2024). These imputations were done with the .fillna().median() or .mean() functions (Tamboli 2021). After imputation is completed, we verify if the missing values are resolved and if the new distributions are in proper alignment.

A graph of numbers and a number of children

Description automatically generatedA graph of a number of people

Description automatically generated with medium confidence

A graph of income

Description automatically generatedA graph of blue bars

Description automatically generated

Figure 6: Histograms before imputation.

A graph of numbers and a number of children

Description automatically generatedA graph of age and age

Description automatically generated

A graph of income and income

Description automatically generatedA graph of blue bars

Description automatically generated

Figure 7: Histogram after imputation.

Outliers were detected using seaborn boxplots. The columns Age and Initial\_days had no outliers while Children had 4 and Income had 2500. Since we do not know if the outliers are factual errors, we will extract the outliers, save them as their own data frame and then remove them from the original data frame. This is so that we can determine the z-scores of the variables. Finally, check the new data frame for any outliers. This demonstrates that the outliers were extracted successful to their own data frame. This shows that any outliers with a z-score lower than -3 and higher than 3 were checked for. Their sums were returned showing 0 for the sum of existing outliers.

A graph with a blue rectangle and black lines

Description automatically generatedA blue rectangular object with black lines

Description automatically generatedA graph with a bar and a line

Description automatically generated with medium confidenceA blue rectangular object with white lines

Description automatically generated

Figure 8: Boxplots after imputation.

A screenshot of a calculator

Description automatically generatedA screenshot of a calculator

Description automatically generated

Figure 9: First 10 Z-scores for Children and Income

A screenshot of a computer code

Description automatically generated

Figure 10: Extracting the outliers.

## D3: Summarized Work

The data has no duplicated values. The missing values were treated as well as the outliers. Outliers have been extracted to their own data frames so as not to completely remove these values from future analysis. Output of code shows no more existing outliers in the data frame based on z-scores. Two columns were re-expressed from 0/1s to Yes/No to address data uniformity issues.

A screenshot of a computer code

Description automatically generated

Figure 11: Summarized Output after Imputation

## D4: Code

Please attached code. The file is ‘D206 Data Cleaning FINAL.ipynb.’

## D5: Treated/Cleaned CSV

Cleaned data has been uploaded for the assessment. The attached file is Medical\_Data\_Treated\_Final.CSV.

## D6: Advantages/Disadvantages of Treatments

No duplicates were found so no limitations on this data cleaning step. The advantage of using the .duplicated() function made it easy to evaluate the entire data set for duplicates.

The methods used were based on the PowerPoints Dr. Middleton presented. Depending on the distributions of the histograms, different values were used to impute the missing data (Middleton 2024). The Univariate Imputation was used to replace the missing data since these include mean and median values. This could possibly distort the data and its distribution (Middleton 2024). For the variables Children, Income, Initial\_days, the median value was used for imputation since the distributions skewed right and bimodal (Tamboli 2021). Meanwhile, the variable Age exhibited a normal distribution with no outliers so the mean was used (Tamboli 2021). The advantage of using these methods is that the statistical properties of the variables were mostly retained since it would be beneficial for future data analysis. Imputation is used to replace missing data because removing data is not always feasible and can reduce the size of the data set, which can possibly bias the data set and lead to incorrect analysis (Middleton 2024).

It is important to handle missing data for multiple reasons. First, many machine learning algorithms fail if the data set contains missing values. Some algorithms like K-nearest neighbors and naïve Bayes support data with missing values. Second, you may end up building a biased machine learning model, leading to incorrect results if the missing values are not handled properly. Finally, missing data can lead to a lack of precision in the statistical analysis (Middleton 2024).

One disadvantage of using these values, instead of removing them or using other techniques, is that their histograms would be visually distorted. In the case of outliers, the extraction method was used – to establish data integrity, the outliers were removed and included in an individual data frame. By doing this, the data could be merged to include the original dataset for analysis if needed. Creating a separate data frame for the outliers also allowed us the opportunity to specifically inspect the outliers.

## D7: Challenges in Data Analysis

Imputation of the median and mean values may distort the data to some extent. This changes the overall statistics of the data and could introduce some degree of error, which can be accounted for. By removing the outliers, the extreme values were removed from the cleaned data. These outliers could be by error or true in nature, which is why it is important to include these in the analysis.

Another challenge is that it is tough to conduct data analysis if you don’t have the domain knowledge necessary to interpret the data. There are four ways to treat outliers: imputation, retention, exclusion, and removal. Without extensive domain knowledge, it makes it difficult to understand the relationships of the outliers with the variables along with which method to use for treating the outliers. This could also skew the results of our PCA analysis since PCA is sensitive to outliers.

# Part IV: Principal Component Analysis (PCA)

## E1: PCA Loadings

Only quantitative variables can be used for the PCA. The variables are: Lat, Lng, Population, Children, Age, Income, VitD\_level, Doc\_visits, Full\_meals\_eaten, VitD\_supp, Initial\_days, TotalCharge, and Additional\_charges. CaseOrder is left out since it is an index. Analyzing the PCs determines which combination of related variables are significantly associated depending on how close they are to 1 or -1. For PC1, the variables with the most significant correlations are TotalCharge, VitD\_levels, and Initial\_days. For PC2, the variables with the most significant correlations are Age and Additional\_charges. For PC3, the variables with the most significant correlations are Lat and Population. For PC4, the variables with the greatest correlations are Lng and VitD\_supp. For PC5, the variables with the greatest correlations are Full\_meals\_eaten, Lng, Income, and Initial\_days.

A screenshot of the PCA loadings is attached below:

A table of numbers and symbols

Description automatically generated

Figure 12: PCA Loadings

## E2: Which PCAs to Keep?

The PCA loadings that should be kept are PCs 1 through 6. The Kaiser rule is used to decide which principal components to keep. This rule states that if the eigenvalue for a principal component is equal to or greater than one, it should be kept. A scree plot is created to visualize this rule. To determine accuracy, I also printed out the eigenvalues to confirm the threshold of 1 or above was being met. These eigenvalues represent the amount of variance explained by each PC.

A graph with a line

Description automatically generated

Figure 13: Scree Plot

A number of a number

Description automatically generated with medium confidence

Figure 14: Eigenvalues

## E3: Benefits of PCA

PCA allows for dimensionality reduction. This is useful for data sets with many variables. This is done with the quantitative variables. PCA allows for analysis on 6 components that still explain more than 50% of the variance. Keeping more PCs would explain 90% of the variance, but at the cost of less dimensionality reduction.

Dimensionality reduction is beneficial to organizations because high dimensionality can be detrimental to the performance of algorithms used to analyze the data, causing them to run slowly and require significantly more resources (Biga Bid Ltd, 2023). Sacrificing a small amount of accuracy can be worth it if it saves time and dimensionality is reduced.

PCA is also beneficial because it can remove unnecessary noise in the data that could lead to poor analysis and its ability to provide an organization with uncorrelated data features (Biga Bid Ltd, 2023). The accuracy of classification models can be improved after PCA is performed on a data set. PCA also allows us to visualize data and allow for the inspection of clustering and classification algorithms (Biga Bid Ltd, 2023).

References (No third party code references used)

Biga Bid Media Ltd. (2023, February 8). What is Principal Component Analysis (PCA) & How to Use It?. BigaBid.

Middleton. (2024). Webinar for Missing Data. Retrieved from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=767749d2-ba19-4f94-bec8-b058017b2f5e

Middleton (2024). Webinar for Reexpression of Categorical Variables. Retrieved from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=bd7b8541-77ba-42e0-80a4-b059010bc790

Middleton (2024). Webinar for PCA. Retrieved from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3bcc452f-fa35-43be-b69f-b05901356f95

Tamboli, N. (Oct., 2021). All you need to know about different types of missing data values and how to handle it. Analytics Vidhya. Available at <https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/>

https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html

# Table of Figures

[Figure 1: Variable List and Data Types 7](#_Toc164027185)

[Figure 2: Shape of data frame and total duplicated values that exist 12](#_Toc164027186)

[Figure 3: Summary of variables with null values 13](#_Toc164027187)

[Figure 4: Histograms of quantitative variables 13](#_Toc164027188)

[Figure 5: Missing Data Matrix from missingno 14](#_Toc164027189)

[Figure 6: Histograms before imputation 17](#_Toc164027190)

[Figure 7: Histogram after imputation 18](#_Toc164027191)

[Figure 8: Boxplots after imputation 19](#_Toc164027192)

[Figure 9: First 10 Z-scores for Children and Income 20](#_Toc164027193)

[Figure 10: Extracting the outliers 20](#_Toc164027194)

[Figure 11: Summarized Output after Imputation 21](#_Toc164027195)

[Figure 12: PCA Loadings 24](#_Toc164027196)

[Figure 13: Scree Plot 25](#_Toc164027197)

[Figure 14: Eigenvalues 25](#_Toc164027198)