­

**Performance Assessment for D206: Data Cleaning**

Christopher Fischer - 011933891

College of Information Technology, Western Governors University

March 20, 2024

Performance Assessment for D206: Data Cleaning

This document contains the tasks and outputs required for the “NUM3 TASK 1: DATA CLEANING” assessment. All work is original to the author unless otherwise indicated by a citation.

# A - Research Question

Management is disheartened by the number of customers cancelling their services with our company. Clearly, work needs to be done to better understand which customers tend to cancel their service. The data analytics team has been provided with a CSV file containing many attributes about our customers including their demographics, their subscribed service options, their usage of our services, their experience with our support team, and their outcomes. Management wants to know *which characteristics of our customers tend to occur most often in those who decide to terminate their subscription to our services*.

# B – Data Dictionary

The following table details the raw input data to this study.

| Variable Name | Data Type | Description | Example Value |
| --- | --- | --- | --- |
| (unlabeled) | Categorical/Ordinal | A duplicate copy of the CaseOrder variable | 3814 |
| CaseOrder | Categorical/Ordinal | A sequence number assigned to each observation to retain the original ordering of the data | 3814 |
| Customer\_id | Categorical/Nominal | A unique ID for each customer | B800553 |
| Interaction | Categorical/Nominal | A unique system-generated ID (GUID) for each customer interaction | b4181fbe-b9f0-465c-8dd3-ad4fa6857842 |
| City | Categorical/Nominal | The city from the customer's address, taken from their billing record | Salt Lake City |
| State | Categorical/Nominal | The state from the customer's address, taken from their billing record | UT |
| County | Categorical/Nominal | The county from the customer's address, taken from their billing record | Salt Lake |
| Zip | Categorical/Nominal | The ZIP code from the customer's address, taken from their billing record | 84106 |
| Lat | Numerical/Continuous | The latitude coordinate of the customer's address, taken from their billing record | 40.70673 |
| Lng | Numerical/Continuous | The longitude coordinate of the customer's address, taken from their billing record | -111.8557 |
| Population | Numerical/Discrete | The population of the region within 1 mile of the customer's address, taken from census data | 36047 |
| Area | Categorical/Nominal | The type of area in which the customer's address is located | Suburban |
| Timezone | Categorical/Nominal | The customer's timezone reported when on-boarded | America/Denver |
| Job | Categorical/Nominal | The customer's job title reported when on-boarded | Public relations officer |
| Children | Numerical/Discrete | The number of children in the customer's family reported when on-boarded | 0 |
| Age | Numerical/Discrete | The customer's age reported when on-boarded | 54 |
| Education | Categorical/Ordinal | The customer's highest level of education reported when on-boarded | Bachelor's Degree |
| Employment | Categorical/Nominal | The customer's employment status reported when on-boarded | Unemployed |
| Income | Numerical/Discrete | The customer's income reported when on-boarded | NA |
| Marital | Categorical/Nominal | The customer's marital status reported when on-boarded | Separated |
| Gender | Categorical/Nominal | The customer's gender reported when on-boarded | Male |
| Churn | Categorical/Nominal | Yes/no indicating if the customer has cancelled their service in the last 30 days | Yes |
| Outage\_sec\_perweek | Numerical/Continuous | Average number of seconds service was unavailable each week within the customer's area | 11.05678 |
| Email | Numerical/Discrete | Number of emails we sent to the customer in the last 365 days | 9 |
| Contacts | Numerical/Discrete | Count of how many times the customer interacted with our support team | 0 |
| Yearly\_equip\_failure | Numerical/Discrete | Count of how many times the customer's equipment required reset or replacement in the last 365 days | 0 |
| Techie | Categorical/Nominal | Yes/no indicating if the customer believes themselves to be technically minded | Yes |
| Contract | Categorical/Nominal | The term of service the customer contracted (monthly, 1 year, 2 year) | Two Year |
| Port\_modem | Categorical/Nominal | Yes/no indicating if the customer received a portable modem | No |
| Tablet | Categorical/Nominal | Yes/no indicating if the customer reports owning a tablet computing device | No |
| InternetService | Categorical/Nominal | The customer's type of internet service | None |
| Phone | Categorical/Nominal | Yes/no indicating if the customer reports having phone service | Yes |
| Multiple | Categorical/Nominal | Yes/no indicating if the customer reports having more than one line of phone service | Yes |
| OnlineSecurity | Categorical/Nominal | Yes/no indicating if the customer subscribes to the optional security service | No |
| OnlineBackup | Categorical/Nominal | Yes/no indicating if the customer subscribes to the optional backup service | No |
| DeviceProtection | Categorical/Nominal | Yes/no indicating if the customer subscribes to the optional device protection service | Yes |
| TechSupport | Categorical/Nominal | Yes/no indicating if the customer subscribes to the optional technical support service | NA |
| StreamingTV | Categorical/Nominal | Yes/no indicating if the customer subscribes to the streaming television service | Yes |
| StreamingMovies | Categorical/Nominal | Yes/no indicating if the customer subscribes to the streaming movies service | Yes |
| PaperlessBilling | Categorical/Nominal | Yes/no indicating if the customer has chosen to receive billing statements electronically | No |
| PaymentMethod | Categorical/Nominal | The customer's selected method of payment | Mailed Check |
| Tenure | Numerical/Discrete | The number of months the customer has received service | 2.398402 |
| MonthlyCharge | Numerical/Continuous | The customer's average monthly billed amount | 219.8711 |
| Bandwidth\_GB\_Year | Numerical/Continuous | The customer's average amount of data consumed per year, in gigabytes | 840.4154 |
| item1 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of receiving timely responses | 2 |
| item2 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of receiving timely fixes | 2 |
| item3 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of receiving timely replacements | 3 |
| item4 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of service reliability | 3 |
| item5 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of service options | 5 |
| item6 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of receiving respectful responses | 1 |
| item7 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of having courteous exchanges with us | 1 |
| item8 | Categorical/Ordinal | The customer's reported importance (on a 1-8 scale) of seeing evidence that we are actively listening | 3 |

Table 1 - Input Data Dictionary

# C – Data Cleaning Plan

The following sections outline the team’s plan to evaluate the quality of the input data.

## C1 – Methods of Detection and C2 - Reasons for Selection of Methods

A comprehensive data cleaning process involves looking at many measures of data quality. For this study, the following potential anomalies were screened for: duplicates (both entire observations and unique identifiers), missing values, value errors, outliers, candidates for re-expression of categorical variables. The team also examined the dataset for any other characteristics which might cause confusion later in the study.

### Duplicates

After a cursory review of output from the describe() method, the duplicated().sum() method was applied to the columns attribute of the data frame. This method would identify if any variables were duplicated in the data set. This method would permit duplicated columns to be removed from the data set. Removing extraneous columns would help streamline future cleaning and analysis processes. (Altacademy, 2024)

Next, the len(df)-len(df.drop\_duplicates()) method was used to screen for any observations containing entirely duplicated values. Additionally, as the *Customer\_id* and *Interaction* variables assert uniqueness, the same method was applied to each of these variables to detect if duplicate values were present. These methods provided a quick and concise means of identifying if any duplicate records or key values were present in the data set, warranting further analysis and remediation.

### Missing Values

To detect and evaluate any missing values, several methods were employed. First, the missingno.matrix() was plotted and viewed. This matrix provided a data-rich visualization of the extent of missing values across all variables in the data set. From this matrix, a hypothesis was formed that some missing values might be correlated with missing values in the *Age* variable. This hypothesis was tested by repeating the missingno.matrix() plot after sorting the data frame by *Age*. Another attempt to identify any correlations between missing values was performed using the missingno.heatmap() method. Lastly, in preparation for later cleaning activities, a separate data frame containing the variables with missing values was produced using the isnull().any() method as a filter on the data frame. Having these variables replicated to a separate data frame would permit trial and error cleaning attempts without the risk of damaging the original data set. The count of missing values for each variable was computed using the isnull().sum() method. (Middleton, Getting Started with Detecting and Treating Missing Values, 2023)

### Value Errors

Two sets of variables were checked for value errors: Yes/No variables and the Survey Response variables. For each set, an array of permitted values was used with the isin() method to filter the data frame for any variables in each subset which contained impermissible values. This method provided a quick means to visualize the quality of the values in these two sets of variables. Were any invalid values to be detected, further treatment could be planned and performed.

### Outliers

A search for potential outliers was conducted in several steps. First, z-scores were computed for each numerical variable using the mean()/std() method. Next, histograms for all the z-scores were plotted. Next, boxplots were created for any numeric variable which contained any values with a z-score > 3.0. Last, a count of observations containing values with z-scores > 3.0 was performed for each variable containing such values. Choosing these methods combined a statistical/mathematical analysis (z-scores and counts) with visual analysis (histograms and boxplots). By combining methods, the analyst had the best of all worlds for assessing the range and count of outlying values in the numerical variables. These are both required to select the most appropriate treatment method. (Middleton, Getting Started with Detecting and Treating Outliers, 2023)

### Re-expression Candidates

The unique values of each categorical variable were visualized using the value\_counts() method. Each list was assessed looking for opportunities to re-express the variable as ordinal integer values and/or into fewer values. For each candidate variable identified, a mapping of current values to new values was created which would be applied in the treatment phase. Re-expression of categorical variables as integers can make those variables available to further analysis later with algorithms for which textual data is unsuitable. (Larose & Larose, 2019)

## C3 – Language and Library Selection

The language selected for this data cleaning activity was Python. In its nearly 35 years, Python has become one of the most widely used general-purpose programming languages. Python is backed by a huge community of developers and a vast collection of libraries which extend the core capabilities of the language. Python is considered to use one of the most English-like syntaxes providing excellent readability. (Datacamp, 2022)

Another reason for selecting Python is that virtually all production data analytics projects in our company use Python. While R is an excellent choice for interactive analysis activities, Python is better suited for deployment on production servers as part of a data pipeline. (WGU Information Technology, n.d.)

The following table details the Python libraries used for this project.

|  |  |
| --- | --- |
| Library | Purpose |
| numpy | Provides many computational and array processing methods, including the where() method used in this project |
| pandas | Provides tools for processing tabular data sets |
| missingno | Provides methods for visualizing missing data in a data set, including the matrix() and heatmap() methods used in this project |
| sklearn.decomposition | Provides many machine-learning algorithms, including the PCA class used in this project |
| matplotlib.pyplot | Provides the tools needed to generate charts and graphs for visualizing data including histograms, boxplots, and scree plots used in this project |

Table 2 - Python Libraries in Use

## C4 – Detection Code

This code was used to perform the detection tests described in [C1 – Methods of Detection and C2 - Reasons for Selection of Methods](#_C1_–_Methods) above. Please see the Python notebook ChurnDataCleansing.ipynb submitted with this document for results of each code block.

#import needed libraries  
import numpy as np  
import pandas as pd  
import missingno as msno  
import matplotlib.pyplot as plt  
from sklearn.decomposition import PCA

#initialize arrays  
numerical\_variables = ['Age','Income','Children','Tenure','Bandwidth\_GB\_Year', \  
 'Outage\_sec\_perweek','Email','Contacts','Yearly\_equip\_failure','MonthlyCharge']  
categorical\_variables = ['Education','Employment','Marital','Gender','InternetService', \  
 'PaymentMethod']  
yes\_no\_variables = ['Churn','Techie','Port\_modem','Tablet','Phone','Multiple', \  
 'OnlineSecurity','OnlineBackup','DeviceProtection','TechSupport','StreamingTV', \  
 'StreamingMovies','PaperlessBilling']  
survey\_variables = ['item1','item2','item3','item4','item5','item6','item7','item8']  
survey\_answers = [1,2,3,4,5,6,7,8]

#load church\_missing\_data.csv  
raw\_data = pd.read\_csv('churn\_raw\_data.csv')

#Examine the data frame [In-Text Citation: (Patanam, 2015)]  
raw\_data.info()  
raw\_data.describe(include='all')

#Find the number of duplicated variable names  
#Formatting code [In-Text Citation:(Sanchhaya Education, Pvt Ltd, 2023)]  
print('Number of duplicated variables : {}'.format(raw\_data.columns.duplicated().sum()))

#Count duplicated rows [In-Text Citiation: (Zach, 2022)]  
#Formatting code [In-Text Citation:(Sanchhaya Education, Pvt Ltd, 2023)]  
print('Duplicate rows : {}'.format(len(raw\_data)-len(raw\_data.drop\_duplicates())))

#Count duplicate Customer\_id and Interaction values [In-Text Citiation: (Zach, 2022)]  
#Formatting code [In-Text Citation:(Sanchhaya Education, Pvt Ltd, 2023)]  
print('Duplicate Customer\_ids : {}'.format(len(raw\_data['Customer\_id']) - \  
 len(raw\_data['Customer\_id'].drop\_duplicates())))print('Duplicate Interactions : {}' \  
 .format(len(raw\_data['Interaction'])-len(raw\_data['Interaction'].drop\_duplicates())))

#Print the missing data matrix [In-Text Citation:(Middleton, Getting Started with Detecting and Treating Missing Values, 2023)]  
msno.matrix(raw\_data, fontsize = 12, labels=True)  
plt.title('Missing Data Matrix')  
plt.show()

#Sort the data by Age to see if there is a correlation of missingness between Age, Children, Income [In-Text Citation:(Middleton, Getting Started with Detecting and Treating Missing Values, 2023)]  
msno.matrix(raw\_data.sort\_values(by='Age'), fontsize = 12, labels=True)  
plt.title('Missing Data Matrix by Age')  
plt.show()

# Display a heatmap to show any correlation between missing columns  
msno.heatmap(raw\_data, fontsize = 12, labels=True)  
plt.title('Missing Data Heatmap')  
plt.show()

#List any features with missing values [In-Text Citation: (Uzunov, 2016)]  
missing\_data = raw\_data[raw\_data.columns[raw\_data.isnull().any()]].copy()  
print(len(missing\_data.columns)) # How many variables have missing data?  
print(missing\_data.isnull().sum()) # How many missing values in each variable?

#Check values in survey questions [In-Text Citation: (Guar, 2019)]  
print(raw\_data[survey\_variables][~raw\_data[survey\_variables].isin(survey\_answers)].count())

#Check values in yes/no variables [In-Text Citation: (Guar, 2019)]  
print(raw\_data[yes\_no\_variables][~raw\_data[yes\_no\_variables].isin(['Yes','No'])].count())

#Compute zscore for all numerical variables (other than demographics) [In-Text Citation:(Bathelt, 2017)]  
raw\_data\_z = (raw\_data[numerical\_variables] - raw\_data[numerical\_variables].mean()) / \  
 raw\_data[numerical\_variables].std(ddof=0)   
raw\_data\_z.hist(edgecolor='black', grid=False, figsize=(15,15))  
plt.show()

#Examine boxplots of variables with any values with zscore > 3 [In-Text Citation:(Varun, 2023)]  
outlier\_cols = raw\_data\_z.loc[: , (raw\_data\_z > 3.0).any()].columns  
raw\_data[outlier\_cols].plot(kind='box', subplots=True, layout=(3,3), sharex=False, \   
 sharey=False, figsize=(10, 15))  
plt.show()

#Count, min, max of outliers  
#Formatting code [In-Text Citation:(Sanchhaya Education, Pvt Ltd, 2023)]  
for col in outlier\_cols :  
 cnt = len(raw\_data\_z[raw\_data\_z[col]>3])  
 min, max = raw\_data[col].min(), raw\_data[col].max()  
 print('Likely outlier for {0:<20}\t Count: {1:7d} ({2:5.2%} of observations)\tMin: {3:>9.2f}\tMax: {4:>9.2f}'.format(col,cnt,cnt/10000,min,max))

#Evaluate categorical variables for re-expression  
for column in categorical\_variables:  
 print(raw\_data[column].value\_counts())

# D – Data Cleaning Process

The following sections detail the results of the data quality evaluation and the treatment methods applied to improve the overall quality of the input data set.

## D1 – Findings of Detection Process

The detection code above found several anomalies. These are summarized in the following table:

| Test | Result | Treatment Required |
| --- | --- | --- |
| Check for duplicate columns | One duplicate column found; an unnamed column which duplicates the data found in *CaseOrder* | Y |
| Check for poorly named variables | The survey response variables (i*tem1…item8*) are poorly named and may add confusion to future analysis activities | Y |
| Check for duplicate records | No duplicate records found | N |
| Check for duplicate *Customer\_id* values | No duplicate values found | N |
| Check for duplicate *Interaction* values | No duplicate values found | N |
| Check for variables with missing values | Missing values found   |  |  |  | | --- | --- | --- | | Variable | Missing Count | Percent of Total | | Children | 2495 | 25.0% | | Age | 2475 | 24.8% | | Income | 2490 | 24.9% | | Techie | 2477 | 24.8% | | InternetService | 2129 | 21.3% | | Phone | 1026 | 10.3% | | TechSupport | 991 | 9.9% | | Tenure | 931 | 9.3% | | Bandwidth\_GB\_Year | 1021 | 10.2% | | Y |
| Check for correlation between missing values | No correlation found | N |
| Check for invalid values in survey questions | No invalid values found | N |
| Check for invalid values in yes/no variables | No invalid values found (other than missing values) | N |
| Check for outliers in numerical variables | Outliers found   |  |  |  |  | | --- | --- | --- | --- | | Variable | Outlier Count | Outlier Values | | | **Min** | **Max** | | Income | 110 | 740.66 | 258,900.70 | | Children | 144 | 0 | 10 | | Outage\_sec\_perweek | 491 | -1.35 | 47.05 | | Email | 3 | 1 | 23 | | Contacts | 165 | 0 | 7 | | Yearly\_equip\_failure | 94 | 0 | 6 | | MonthlyCharge | 3 | 77.51 | 315.88 | | Y |
| Check for categorical variables suitable for re-expression | Suitable variables identified   |  |  | | --- | --- | | Variable | Re-expression Method | | Churn | Yes: 1, No: 0 | | Techie | Yes: 1, No: 0 | | Port\_modem | Yes: 1, No: 0 | | Tablet | Yes: 1, No: 0 | | Phone | Yes: 1, No: 0 | | Multiple | Yes: 1, No: 0 | | OnlineSecurity | Yes: 1, No: 0 | | OnlineBackup | Yes: 1, No: 0 | | DeviceProtection | Yes: 1, No: 0 | | TechSupport | Yes: 1, No: 0 | | StreamingTV | Yes: 1, No: 0 | | StreamingMovies | Yes: 1, No: 0 | | PaperlessBilling | Yes: 1, No: 0 | | Education | Remap to 9 ordinal integers | | Employment | Remap to 3 ordinal integers | | Y |

Table 3 - Detection Results

## D2 – Treatment Methods

The following table outlines the treatment methods applied to each test indicated as requiring treatment in [Table 3](#Table3) above.

| Condition | Treatment Method | Reason |
| --- | --- | --- |
| Duplicate column detected | Drop the column from the data set | Leaving an extraneous, redundant column can lead to confusion later and increase processing times. (Altacademy, 2024) |
| Poorly named variables – *Item1...Item8* | Rename these columns in the data set to more meaningful names | Poorly named variables will lead to confusion and possibly incorrect results. (Koehrsen, 2019) |
| Fill missing values – *Children* | Simple univariate imputation using the median value | Median is selected due to there being a right skew distribution to the data. (Middleton, Getting Started with Detecting and Treating Missing Values, 2023) |
| Fill missing values – *Age* | Simple univariate imputation using the mean value, rounded to whole numbers | Mean is selected due to there being a uniform distribution to the data. (Middleton, Getting Started with Detecting and Treating Missing Values, 2023) |
| Fill missing values – *Income* | Simple univariate imputation using the median value | Median is selected due to there being a right skew distribution to the data. |
| Fill missing values – *Techie* | Simple univariate imputation using the mode value | Mode is selected because the variable is categorical. (Middleton, Getting Started with Detecting and Treating Missing Values, 2023) |
| Fill missing values – *InternetService* | Simple univariate imputation using the mode value | Mode is selected because the variable is categorical. |
| Fill missing values - *Phone* | Simple univariate imputation using the mode value | Mode is selected because the variable is categorical. |
| Fill missing values - *TechSupport* | Simple univariate imputation using the mode value | Mode is selected because the variable is categorical. |
| Fill missing values - *Tenure* | Simple univariate imputation using the median value | Median is selected due to there being a bimodal distribution to the data. (Middleton, Getting Started with Detecting and Treating Missing Values, 2023) |
| Fill missing values – *Bandwidth*\_*GB*\_*Year* | Simple univariate imputation using the median value | Median is selected due to there being a bimodal distribution to the data. |
| Treat outliers – *Income* | Retain outliers | There are 1.1% outlying values, all to the upper end. However, the max value of 258900.70 is not inconceivable for income. |
| Treat outliers – *Children* | Retain outliers | There are 1.4% outlying values with the outliers being between 8 and 10. While these may seem high, they are not implausible, particularly for older customers. |
| Treat outliers – *Outage\_sec\_perweek* | Replace values < 0 with 0; retain outliers at the upper end | It is not possible to have an outage which has an elapsed time of less than zero seconds; these values will be replaced with zero. The outliers at the upper band are all under 48 seconds, which is well within a reasonable range. |
| Treat outliers – *Email* | Retain outliers | There are just 0.03% outliers with a maximum value of 23. While high, it is conceivable that a customer having significant troubles might require multiple email messages to resolve the concern. |
| Treat outliers – *Contacts* | Retain outliers | There are 1.7% outlying values with a maximum of 7 contacts. As with email, it is entirely conceivable that a customer may need repetitive contacts to resolve a concern. |
| Treat outliers – *Yearly\_equip\_failure* | Retain outliers | There are 0.9% outlying values with a maximum of 6 failures. As we are trying to determine which customers cancel their service contracts, keeping as much data on equipment failure as possible may be helpful. |
| Treat outliers – *MonthlyCharge* | Retain outliers | There are 0.03% outlying values with a maximum of $315.88. While this may seem high, for customers who choose multiple service options, their bill could easily reach this level. |
| Re-expression – Yes/No variables | Remap:   * Yes -> 1 * No -> 0 | There are several techniques which allow the use of Boolean/bi-valued categorical variables. However, those techniques require re-expression of the variables into 0 and 1. (Kaplan & Pruim, 2023) |
| Re-expression – *Education* | Remap:   * No Schooling Completed -> 0 * Nursery School to 8th Grade -> 8 * 9th Grade to 12th Grade, No Diploma -> 9 * Regular High School Diploma  -> 12 * GED or Alternative Credential  -> 12 * Some College, 1 or More Years, No Degree -> 13 * Some College, Less than 1 Year -> 13 * Associate's Degree -> 14 * Bachelor's Degree -> 16 * Master's Degree -> 18 * Professional School Degree  -> 20 * Doctorate Degree -> 20 | Remapping a descriptive, categorical variable into integer values permits this data to be used in certain machine learning algorithms. However, this only applies to those variables whose values imply an ordering. This mapping attempts to provide not just an ordering of the values from lowest to highest, but a weighting of the various education levels as well. (Datacamp, 2023) |
| Re-expression – *Employment* | Remap:   * Full Time -> 2 * Part Time -> 1 * Retired -> 0 * Student -> 0 * Unemployed -> 0 | Remapping a descriptive, categorical variable into integer values permits this data to be used in certain machine learning algorithms. However, this only applies to those variables whose values imply an ordering. To achieve an ordinal arrangement, this mapping condenses all the “not working” values into zero for better analysis. |

Table 4- Treatment Methods

## D3 – Results of Treatment

The methods documented in section [C1 – Methods of Detection and C2 - Reasons for Selection of Methods](#_C1_–_Methods) were executed according to plan. The detection plan identified several data quality problems:

* One (1) redundant column
* Eight (8) variables with suboptimal naming
* Nine (9) variables with missing values
* Seven (7) variables with potential outlier values
* 15 categorical variables which would benefit from re-expression

Each of these issues was evaluated and treated according to the methods described in section [D2 – Treatment Methods](#_D2_–_Treatment). All methods were executed in a Jupyter Notebook. Jupyter provides an outstanding means of capturing code, documentation, and results all in one place. After completing the treatment plan:

* The redundant column has been removed from the data set

|  |  |
| --- | --- |
| Before    Figure 1 - df.info() before treatment | After    Figure 2 - df.info() after treatment |

* Survey response columns renamed for better clarity

|  |  |
| --- | --- |
| Before    Figure 3 - df.info() before treatment | After    Figure 4 - df.info() after treatment |

* All missing values have been treated

|  |  |
| --- | --- |
| Before    Figure 5 - msno.matrix() before treatment    Figure 6 - Count of missing values by variable before treatment | After    Figure 7 - msno.matrix() after treatment    Figure 8 - Count of missing values by variable after treatment |

* Treated columns have statistically similar distributions after treatment

|  |  |
| --- | --- |
| Before    Figure 9 - Histograms of numeric variables with missing values before treatment | After    Figure 10 - Histograms of numeric variables with missing values after treatment |

* Univariate statistics of treated columns have acceptable deviation after treatment  
    
  A screenshot of a computer screen

  Description automatically generated

Figure 11 - Summary of univariate statistics of treated variables before and after treatment

* One variable was successfully treated for outliers

|  |  |
| --- | --- |
| Before    Figure 12 - Boxplot of Outage\_sec\_perweek before treatment | After    Figure 13 - Boxplot of Outage\_sec\_perweek after treatment |

* All Yes/No variables re-expressed as integers

A screenshot of a computer

Description automatically generated

Figure 14 - df.info() after treatment

* Education variable re-expressed as an ordinal integer

A screenshot of a computer

Description automatically generated

Figure 15 value\_counts() after treatment

* Employment variable re-expressed as an ordinal integer

A close-up of a code

Description automatically generated

Figure 16 - value\_counts() after treatment

## D4 – Treatment Code

This code was used to perform the treatment activities described in [D2 – Treatment Methods](#_D2_–_Treatment) above. Please see the Python notebook ChurnDataCleansing.ipynb submitted with this document for results of each code block.

#Drop column zero as it is a copy of column 1 [In-Text Citation:(Chen, 2019)]  
raw\_data.drop(raw\_data.filter(regex="Unnamed"),axis=1, inplace=True)

#Copy the survey response variables and rename for better clarity; leave the original columns intact  
raw\_data[['Survey\_TimelyResponses', \  
 'Survey\_TimelyFixes', \  
 'Survey\_TimelyReplacements', \  
 'Survey\_Reliability', \  
 'Survey\_Options', \  
 'Survey\_Respectful', \  
 'Survey\_Courteous', \  
 'Survey\_ActiveListening']] = \  
 raw\_data[['item1', \  
 'item2', \  
 'item3', \  
 'item4', \  
 'item5', \  
 'item6', \  
 'item7', \  
 'item8']]

#Plot a histogram of all numeric columns with missing values  
missing\_data.hist(edgecolor='black', grid=False, figsize=(10,10))  
plt.show()

#Create a copy of untreated data for later comparison  
missing\_data\_treated = missing\_data.copy()

#Treat the missing values  
missing\_data\_treated['Age'] = missing\_data['Age'].fillna(missing\_data['Age'].mean().round())  
missing\_data\_treated['Income'] = missing\_data['Income'].fillna(missing\_data['Income'].median())  
missing\_data\_treated['Children'] = missing\_data['Children'].fillna(missing\_data['Children'].median())  
missing\_data\_treated['Tenure'] = missing\_data['Tenure'].fillna(missing\_data['Tenure'].median())  
missing\_data\_treated['Bandwidth\_GB\_Year'] = missing\_data['Bandwidth\_GB\_Year'].fillna(missing\_data['Bandwidth\_GB\_Year'].median())  
missing\_data\_treated['Phone'] = missing\_data['Phone'].fillna(missing\_data['Phone'].mode()[0])  
missing\_data\_treated['TechSupport'] = missing\_data['TechSupport'].fillna(missing\_data['TechSupport'].mode()[0])  
missing\_data\_treated['Techie'] = missing\_data['Techie'].fillna(missing\_data['Techie'].mode()[0])  
missing\_data\_treated['InternetService'] = missing\_data['InternetService'].fillna(missing\_data['InternetService'].mode()[0])

#Check mean and median of before and after treatment  
for column in ['Age','Income','Children','Tenure','Bandwidth\_GB\_Year']:  
 mean\_before = missing\_data[column].mean()  
 mean\_after = missing\_data\_treated[column].mean()  
 mean\_diff = (mean\_before - mean\_after) / mean\_before  
  
 median\_before = missing\_data[column].median()  
 median\_after = missing\_data\_treated[column].median()  
 median\_diff = (median\_before - median\_after) / median\_before  
  
 #Formatting code [In-Text Citation:(Sanchhaya Education, Pvt Ltd, 2023)]  
 print('\nVariable: {}'.format(column))  
 print('\tMean before: {0:7.2f}'.format(mean\_before))  
 print('\tMean after: {0:7.2f}'.format(mean\_after))  
 print('\tMean diff: {0:7.2%}'.format(mean\_diff))  
 print('\n\tMedian before: {0:7.2f}'.format(median\_before))  
 print('\tMedian after: {0:7.2f}'.format(median\_after))  
 print('\tMedian diff: {0:7.2%}'.format(median\_diff))

#Plot a histogram of all treated columns  
missing\_data\_treated.hist(edgecolor='black', grid=False, figsize=(10,10))  
plt.show()

#Replace missing value columns in original df with treated columns  
for column in missing\_data\_treated.columns:  
 raw\_data[column] = missing\_data\_treated[column]

#Replace any Outage\_sec\_perweek < 0 with 0  
raw\_data['Outage\_sec\_perweek'] = np.where(raw\_data['Outage\_sec\_perweek'] < 0, 0, raw\_data['Outage\_sec\_perweek'])

#Reexpress yes/no columns as numbers [In-Text Citation:(Eiler, 2017)]  
yesno\_dict = {'No': 0, 'Yes': 1}  
yes\_no\_int\_variables = []  
for column in yes\_no\_variables:  
 new\_col = column + '\_int'  
 yes\_no\_int\_variables.append(new\_col)  
 raw\_data[new\_col] = raw\_data[column].map(yesno\_dict)  
 print(raw\_data[new\_col].value\_counts())

#Re-express education into fewer numeric ordinal values [In-Text Citation:(Eiler, 2017)]  
edu\_dict = {"Regular High School Diploma" : 12, \  
 "Bachelor's Degree" : 16, \  
 "Some College, 1 or More Years, No Degree" : 13, \  
 "9th Grade to 12th Grade, No Diploma" : 9, \  
 "Master's Degree" : 18, \  
 "Associate's Degree" : 14, \  
 "Some College, Less than 1 Year" : 13, \  
 "Nursery School to 8th Grade" : 8, \  
 "GED or Alternative Credential" : 12, \  
 "Professional School Degree" : 20, \  
 "No Schooling Completed" : 0, \  
 "Doctorate Degree" : 20}  
raw\_data['Education\_int'] = raw\_data['Education'].map(edu\_dict)  
print(raw\_data['Education\_int'].value\_counts())

#Re-express employment into fewer numeric values [In-Text Citation:(Eiler, 2017)]  
emp\_dict = {"Full Time" : 2, \  
 "Part Time" : 1, \  
 "Retired" : 0, \  
 "Unemployed" : 0, \  
 "Student" : 0}  
raw\_data['Employment\_int'] = raw\_data['Employment'].map(emp\_dict)  
print(raw\_data['Employment\_int'].value\_counts())

## D5 – Resulting Data Set

This file contains an extract of the cleaned data set. It was also uploaded with this file.



## D6 – Disadvantages of the Selected Methods

As with most areas of data analytics, there are trade-offs to each data cleaning detection and treatment method. In this study, every attempt has been made to select methods appropriate to the data and the research question. Nevertheless, there are some disadvantages that should be noted.

In dealing with missing values, univariate substitutions were performed for each variable with missing values. While this is a simple method, any variability in the observed values due to positive correlation with other variables would be lost for the imputed values as those imputations do not consider the values of the correlated variables. (Donthi & Nehme, n.d.) This potential exists for the *tenure* and *bandwidth\_gb\_year* variables which exhibit a strong correlation.

For detecting outliers, z-score was the primary method employed in this study. However, z-score can yield erroneous results for variables which do not exhibit a normal distribution. (Shukla, 2022) In this case, a method like Inter-quartile Range (IQR) may be more appropriate. Nevertheless, it should be stated that no outlier detection method is perfect as we can seldom be 100% certain that an observation is truly outside an acceptable range. (Gorrie, 2046) The same concerns exist when replacing outlying values as mentioned above in imputing missing values. In the case of this study, a very conservative approach was selected in replacing outliers where only clearly invalid values were replaced while the rest of the candidate outliers were retained for various reasons discussed in [D2 – Treatment Methods](#_D2_–_Treatment).

In selecting which categorical variables to re-express as ordinals, the analyst must apply a good deal of domain knowledge to the given data set to select appropriate variables and replacement values. If a nominal variable is erroneously mistaken for an ordinal, invalid results may be produced by later machine learning algorithms.

## D7 – Potential Challenges for Further Analysis

While this data is much cleaner now having applied these detection and treatment methods, there are limitations which may impact later analysis. First, as previously mentioned, most potential outliers were retained. It is anticipated that making this decision will be helpful for further studies. However, if the interpretation of the outliers as being reasonable was incorrect, the output of models may be rendered inaccurate. Second, the demographic data supplied was not subjected to rigorous cross-referencing. Instead, the direction from management was to trust these data as-is. Should it later be discovered that the quality of the demographics is low, it may hamper later analysis. Lastly, the application of univariate imputation of missing values may prove to be overly simplistic, particularly for the *tenure* and *bandwidth\_gb\_year* variables. Should that prove to be the case, some rework would be needed to employ more robust imputation methods such as multivariate imputation by chained equations (MICE) or k-nearest neighbor (KNN).

# E – Principal Component Analysis

The following sections provide details about how a principal component analysis (PCA) was performed and recommendations for leveraging the results of this analysis.

## E1 – Variables Selected for PCA

These columns were selected for inclusion in the PCA model. They were chosen because of being numerical, non-categorical variables. One of the drawbacks to selecting these variables is that they exhibit relatively weak correlation. PCA works best with variables with stronger correlation. (Middleton, Getting Started with Principal Component Analysis (PCA), 2023)

* Age
* Bandwidth\_GB\_Year
* Children
* Contacts
* Email
* Income
* MonthlyCharge
* Outage\_sec\_perweek
* Tenure
* Yearly\_equip\_failure

After transforming the normalized data, the following loadings resulted:

A black screen with numbers

Description automatically generated

Figure 17 - PCA Loadings

## E2 – Recommended PCAs to be Retained

Here is a scree plot of the eigenvalues for the PCs shown above:

A graph with a line and a red line

Description automatically generated

Figure 18 - Scree plot of eigenvalues

Based on the Kaiser Rule which says that PCs with eigenvalues greater than or equal to one (1) are the most meaningful, PCs 1-4 are recommended to be retained and used for further analysis. (Middleton, Getting Started with Principal Component Analysis (PCA), 2023)

## E3 – Benefits of PCA

PCA, a form of unsupervised learning, can offer several benefits to this research project. First, many machine learning algorithms require exponentially more processing time and resources as the number of variables increases. PCA helps reduce the dimensionality of the data set, thereby reducing the computing power required for further studies. The *churn* data set contains 51 variables in its raw form. PCA could help reduce that to a more manageable study while still yielding actionable results. Second, PCA can help analysts and stakeholders make better sense of complex, multidimensional data similarly by reducing the number of dimensions being plotted.

# F – Recorded Code Review

A recording of the code review presentation was uploaded with this submission. For quick reference, that video may be found here: [Panopto Recording](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=a8ad947b-ded2-4675-9dc9-b13e00dd48ad)

# G – Third-Party Code References

Altacademy. (2024, 01 15). *How to drop column Pandas*. Retrieved 03 21, 2024, from https://www.altcademy.com/blog/how-to-drop-column-pandas/#:~:text=There%20are%20several%20reasons%20you%20might%20want%20to,removing%20unnecessary%20columns%20can%20help%20reduce%20the%20size.

Bathelt, J. (2017). *Pandas - Compute z-score for all columns*. Retrieved 03 18, 2024, from Stackoverflow: https://stackoverflow.com/questions/24761998/pandas-compute-z-score-for-all-columns

Chen, S. (2019). *How to get rid of "Unnamed: 0" column in a pandas DataFrame read in from CSV file*. Retrieved 03 21, 2024, from Stackoverflow: https://stackoverflow.com/questions/36519086/how-to-get-rid-of-unnamed-0-column-in-a-pandas-dataframe-read-in-from-csv-fil

Eiler, J. (2017). *Remap values in pandas column with a dict, preserve NaNs*. Retrieved 03 18, 2024, from Stackoverflow: https://stackoverflow.com/questions/20250771/remap-values-in-pandas-column-with-a-dict-preserve-nans

Gaur, A. (2019). *How to filter Pandas dataframe using 'in' and 'not in' like in SQL*. Retrieved 03 16, 2024, from Stackoverflow: https://stackoverflow.com/questions/19960077/how-to-filter-pandas-dataframe-using-in-and-not-in-like-in-sql

Middleton, D. K. (2023). *Getting Started with Detecting and Treating Missing Values.* Retrieved 03 15, 2024, from Getting Started with D206 Video Series: https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?isAscending=true&CT=1709401191194&OR=OWA%2DNT%2DMail&CID=88c680f2%2D1656%2D34a8%2D0895%2D79fa5f192bd8&sortField=LinkFilename&id=%2Fsites%2FDataS

Middleton, D. K. (2023). *Getting Started with Principal Component Analysis (PCA).* Retrieved 03 20, 2024, from Getting Started with D206 Video Series: https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?isAscending=true&CT=1709401191194&OR=OWA%2DNT%2DMail&CID=88c680f2%2D1656%2D34a8%2D0895%2D79fa5f192bd8&sortField=LinkFilename&id=%2Fsites%2FDataS

Patanam, I. (2015). *Pandas 'describe' is not returning summary of all columns*. Retrieved 03 08, 2024, from Stackoverflow: https://stackoverflow.com/questions/24524104/pandas-describe-is-not-returning-summary-of-all-columns

Ray, S. (2020). *Show all colums of a pandas-dataframe in ".describe()"*. Retrieved 03 08, 2024, from Stack Overflow: https://stackoverflow.com/questions/64455605/show-all-colums-of-a-pandas-dataframe-in-describe

Sanchhaya Education, Pvt Ltd. (2023). *Python | Output Formatting*. Retrieved 03 16, 2024, from Geeksforgeeks: https://www.geeksforgeeks.org/python-output-formatting/

Uzunov, M. (2016). *How to find which columns contain any NaN value in Pandas dataframe*. Retrieved 03 08, 2024, from Stackoverflow: https://stackoverflow.com/questions/36226083/how-to-find-which-columns-contain-any-nan-value-in-pandas-dataframe

Varun. (2023). *Pandas: Select columns based on conditions in dataframe*. Retrieved 03 18, 2024, from thisPointer: https://thispointer.com/pandas-select-columns-based-on-conditions-in-dataframe/

Zach. (2022). How to Count Duplicates in Pandas (With Examples). Retrieved 03 09, 2024, from Statology: https://www.statology.org/pandas-count-duplicates/

# H – Referenced Works

Bathelt, J. (2017). *Pandas - Compute z-score for all columns*. Retrieved 03 18, 2024, from Stackoverflow: https://stackoverflow.com/questions/24761998/pandas-compute-z-score-for-all-columns

Datacamp. (2022). *Python vs R for Data Science: Which Should You Learn?* Retrieved 03 22, 2024, from Datacamp Blog: https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference

Datacamp. (2023, 02). *Handling Machine Learning Categorical Data with Python Tutorial*. Retrieved 03 22, 2024, from Datacamp: https://www.datacamp.com/tutorial/categorical-data

Donthi, S., & Nehme, A. (n.d.). *Dealing with Missing Data in Python*. Retrieved 03 23, 2024, from Datacamp: https://campus.datacamp.com/courses/dealing-with-missing-data-in-python/imputation-techniques

Gorrie, C. (2046, 03 29). *Three ways to detect outliers*. Retrieved 03 23, 2024, from Colin Gorrie's Data Story: https://colingorrie.github.io/outlier-detection.html

Kaplan, D. T., & Pruim, R. (2023). Statistical Modeling: A Fresh Approach. Self-published. Retrieved 03 22, 2024, from https://statistical-modeling.netlify.app/17-logistic

Koehrsen, W. (2019, 07 08). *Data Scientists: Your Variable Names Are Awful. Here’s How to Fix Them.* Retrieved 03 22, 2024, from Towards Data Science: https://towardsdatascience.com/data-scientists-your-variable-names-are-awful-heres-how-to-fix-them-89053d2855be

Larose, C. D., & Larose, D. T. (2019). *Data Science Using Python and R.* Hoboken, NJ, USA: Wiley. Retrieved 03 21, 2024, from https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2091371&site=eds-live&scope=site

Middleton, D. K. (2023). *Getting Started with Detecting and Treating Missing Values.* Retrieved 03 15, 2024, from Getting Started with D206 Video Series: https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?isAscending=true&CT=1709401191194&OR=OWA%2DNT%2DMail&CID=88c680f2%2D1656%2D34a8%2D0895%2D79fa5f192bd8&sortField=LinkFilename&id=%2Fsites%2FDataS

Middleton, D. K. (2023). *Getting Started with Detecting and Treating Outliers.* (W. C. IT, Ed.) Retrieved 03 08, 2024, from Getting Started with D206 Video Series: https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?isAscending=true&CT=1709401191194&OR=OWA%2DNT%2DMail&CID=88c680f2%2D1656%2D34a8%2D0895%2D79fa5f192bd8&sortField=LinkFilename&id=%2Fsites%2FDataS

Middleton, D. K. (2023). *Getting Started with Principal Component Analysis (PCA).* Retrieved 03 20, 2024, from Getting Started with D206 Video Series: https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?isAscending=true&CT=1709401191194&OR=OWA%2DNT%2DMail&CID=88c680f2%2D1656%2D34a8%2D0895%2D79fa5f192bd8&sortField=LinkFilename&id=%2Fsites%2FDataS

Schork, J. (n.d.). *Advantages & Disadvantages of Principal Component Analysis (PCA)*. Retrieved 03 23, 2024, from Statistics Globe: https://statisticsglobe.com/advantages-disadvantages-pca#advantages-of-pca

Shukla, S. (2022, 11 14). *Outliers Detection Using IQR, Z-score, LOF and DBSCAN*. Retrieved 03 23, 2024, from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2022/10/outliers-detection-using-iqr-z-score-lof-and-dbscan/

WGU Information Technology. (n.d.). *R or Python*. Retrieved 03 22, 2024, from https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html