**Data Cleaning Performance Assessment**

Amanda Hartzler

Data Analytics, Western Governors University

D206: Data Cleaning

Dr. Middleton

Table of Contents

Part I: Research Question ............................................................................................................... 3

A ......................................................................................................................................... 3

B ......................................................................................................................................... 3

Part II: Data Cleaning Plan .......................................................................................................... 13

C1 ..................................................................................................................................... 13

C2 ..................................................................................................................................... 14

C3 ..................................................................................................................................... 16

C4 ..................................................................................................................................... 17

Part III: Data Cleaning ................................................................................................................. 20

D1 ..................................................................................................................................... 20

D2 ..................................................................................................................................... 23

D3 ..................................................................................................................................... 31

D4 ..................................................................................................................................... 32

D5 ..................................................................................................................................... 36

D6 ..................................................................................................................................... 36

D7 ..................................................................................................................................... 37

E1 ..................................................................................................................................... 38

E2 ..................................................................................................................................... 38

E3 ..................................................................................................................................... 39

Part IV: Supporting Documents ................................................................................................... 39

F ....................................................................................................................................... 39

G ....................................................................................................................................... 39

H ....................................................................................................................................... 40

**Part I: Research Question and Variables**

**A: Research Question**

Given the data provided in the Churn Dataset, can I determine which factors are most likely to increase a customer’s chance of leaving the company?

**B: Description of Dataset**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| CaseOrder | Quantitative/  Integer | Orders the original data. This variable can be used as a unique key for each row. | 1, 2, 3, 4, 5, ... |
| Customer\_id | Qualitative/  Object | Distinctive ID that identifies individual customers. ID begins with a letter and contains 5 or 6 numeric digits. | K409198,  S120509,  D90850,  F139569, ... |
| Interaction | Qualitative/  Object | Distinctive ID that identifies each interaction between a customer and technical support, a specific transaction, or a sign-up. | aa90260b-4141-4a24-8e36-b04ce1f4f77b |
| City | Qualitative/  Object | According to the billing statement, this is the customer’s city of residence. | Point Baker,  Yamhill,  Cincinnati, ... |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| State | Qualitative/  Object | According to the billing statement, this is the customer’s state of residence. | AK,  OR,  CA, ... |
| Country | Qualitative/  Object | According to the billing statement, this is the customer’s county of residence. | Ogemaw,  Yamhill,  Fort Bend, ... |
| Zip | Quantitative/  Integer | According to the billing statement, this is the customer’s zip code of residence. | 99927,  48661,  92014, ... |
| Lat | Quantitative/  Float | According to the billing statement, this is the latitude of the customer’s residence. | 56.251,  18.3041,  45.35589, .... |
| Lng | Quantitative/  Float | According to the billing statement, this is the longitude of the customer’s residence. | -133.37571,  -84.2408,  -117.24798, ... |
| Lng | Quantitative/  Float | According to the billing statement, this is the longitude of the customer’s residence. | -133.37571,  -84.2408,  -117.24798, ... |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| Population | Quantitative/  Integer | According to census data, this is the population density within a mile of the customer’s residence. | 38,  3735,  13863, ... |
| Area | Qualitative/  Object/ Categorical | According to census data, each residence’s location is categorized as rural, urban, or suburban. | Rural,  Urban,  Suburban |
| TimeZone | Qualitative/  Object/ Categorical | According to the customer’s sign-up information, this is the time zone for their residence. | America/Sitka,  America/Detroit,  America/Denver, ... |
| Job | Qualitative/  Object | According to customer self-reporting, this is the occupation of the customer or invoiced person. | Medical illustrator,  Immunologist,  Counsellor, ... |
| Children | Quantitative/  Float | According to customer self-reporting, this is the number of children in the customer’s household. | 1.0,  4.0,  0.0, ... |
| Age | Quantitative/  Float | According to customer self-reporting, this is their age in years. | 68.0,  27.0,  50.0, ... |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| Education | Qualitative/  Object/  Ordinal | According to customer self-reporting, this is the customer’s highest educational level. | Master’s Degree,  Associate’s Degree,  Bachelor’s Degree, ... |
| Employment | Qualitative/ Object/  Categorical | According to customer self-reporting, this is the employment status of each customer. | Part Time,  Retired,  Student, ... |
| Income | Quantitative/ Float | According to customer self-reporting, this is the customer’s annual income at the time of sign-up. | 28561.99  21704.77  18925.23, ... |
| Marital | Qualitative/  Object/ Categorical | According to customer self-reporting, this is the customer’s marital status at the time of sign-up. | Widowed,  Married,  Separated, ... |
| Gender | Qualitative/  Object/  Categorical | According to customer self-reporting, the customer identifies as male, female, or prefers not to say. | Male,  Female,  Prefer not to answer, ... |
| Churn | Qualitative/ Object/  Binary | If the customer discontinued service in the last month this column will be yes, otherwise no. | Yes,  No |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| Outage\_sec\_perweek | Quantitative/ Float | In the customer's neighborhood, this is the average number of seconds in a week the system has outages. | 6.972566093,  12.01454108,  10.24561565, ... |
| Email | Quantitative/ Integer | Including marketing and other correspondence, this is the number of emails sent to the customer in the past year. | 10,  12,  9, ... |
| Contacts | Quantitative/ Integer | Amount of times the customer has contacted customer support. | 0,  2,  3, ... |
| Yearly\_equip\_failure | Quantitative/ Integer | In the past year, this is the number of times the customer’s equipment failed and had to be reset or replaced. | 1,  0,  3, ... |
| Techie | Qualitative/  Binary | According to a self-reporting questionnaire, this variable is whether the customer considers themselves to be technically inclined. | Yes,  No |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| Contract | Qualitative/  Object/  Categorical | The contract term the customer has selected. | One year,  Month-to-month,  Two Year |
| Port\_modem | Qualitative/ Object/  Binary | Yes if the customer has a port modem, no otherwise. | Yes,  No |
| Tablet | Qualitative/ Object/  Binary | Yes if the customer has a tablet, no otherwise. | Yes,  No |
| InternetService | Qualitative/ Object/  Categorical | This is the internet service provider for the customer. | Fiber Optic,  DSL,  None |
| Phone | Qualitative/ Object/  Binary | Yes if the customer has a phone, no otherwise. | Yes,  No |
| Multiple | Qualitative/ Object/  Binary | Yes if the customer has multiple phone lines, no otherwise. | Yes,  No |
| OnlineSecurity | Qualitative/ Object/  Binary | Yes if the customer has an online security add-on, no otherwise. | Yes,  No |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| OnlineBackup | Qualitative/ Object/  Binary | Yes if the customer has an online backup add-on, no otherwise. | Yes,  No |
| DeviceProtection | Qualitative/ Object/  Binary | Yes if the customer has a device protection add-on, no otherwise. | Yes,  No |
| TechSupport | Qualitative/ Object/  Binary | Yes if the customer has a tech support add-on, no otherwise. | Yes,  No |
| StreamingTV | Qualitative/ Object/  Binary | Yes if the customer has streaming TV, no otherwise. | Yes,  No |
| StreamingMovies | Qualitative/ Object/  Binary | Yes if the customer has streaming movies, no otherwise. | Yes,  No |
| PaperlessBilling | Qualitative/ Object/  Binary | Yes if the customer has paperless billing, no otherwise. | Yes,  No |
| PaymentMethod | Qualitative/  Object/  Categorical | The method that the customer uses to pay. | Credit Card (automatic),  Mailed Check, ... |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| Tenure | Quantitative/  Float | Amount of time, in months, the customer has stayed with the provider. | 6.795512947,  1.156680997,  15.75414408, ... |
| MonthlyCharge | Quantitative/  Float | The average amount charged to each customer per month. | 171.4497621,  242.9480155,  159.4403984, ... |
| Bandwidth\_GB\_Year | Quantitative/  Float | Measured in GB, this is the average amount of data used in a year by the customer. | 904.5361102,  800.9827661,  2054.706961, ... |
| item1 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 1 was about timely response. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |
| item2 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 2 was about timely fixes. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| item3 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 3 was about timely replacements. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |
| item4 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 4 was about reliability. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |
| item5 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 5 was about options. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |

**B: Description of Dataset (continued)**

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example(s) |
| item6 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 6 was about respectful response. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |
| item7 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 7 was about courteous exchange. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |
| item8 | Quantitative/  Integer | Customer response to on survey ranking the importance of several factors. Item 8 was about evidence of active listening. (1 = most important, 8 = least important) | 1,  2,  3,  4, ... |

**Part II: Data Cleaning Plan**

**C1: Detection Methods**

Duplicates: To detect duplicates I used df.duplicated(). This came back false for all values.

Missing Values: To detect missing values I used df.isnull().sum(). This returned all variables in my dataset and the sum of all null values found in each specific variable. Additionally, I used missingno to visualize the missing values (Middleton, 2022a).

Outliers: To detect outliers I created a boxplot visualization for each quantitative variable (Middleton, 2022a).

Re-expressing Categorical Variables: To detect categorical variables that could be re-expressed I looked at the provided data dictionary. Through the data dictionary, I was able to find columns with Yes/No binary values that I could use label encoding on, and I noticed that the values in the Education column could be ordered and re-expressed using ordinal encoding. (Middleton, 2022c)

**C2: Justification for Detection Methods**

|  |
| --- |
| Figure 1 |

Duplicates: I used the code df.duplicated() because it outputs an easy-to-read list with the Boolean values of either True or False (shown in Figure 1). If the row is the first instance of a duplicated value or a unique value then the output would be False, and if it was the second (or

more) instance of a value the output would be True.

|  |
| --- |
| Figure 2 |

|  |
| --- |
| Figure 3 |

Missing Values: First, to detect missing values I used df.isnull().sum() because it outputs an easy-to-read list of each variable and the sum of null values found in each column (shown in Figure 2). Then I created a visualization using the MissingNo package. I learned about this package from the second webinar taught by Dr. Middleton. The code I used was: msno.matrix(df, fontsize = 12, labels = True) (Middleton, 2022a). I used a missingno visualization because it allows me to easily see which variables have missing values (because they are not completely shaded) and because it allows me to visually compare which columns have more missing values (the less shaded means the variable has more missing value). This is shown in figure 3.

|  |
| --- |
| Figure 4 |

**C2: Justification for Detection Methods (continued)**

Outliers: I decided to create a boxplot for each variable to detect outliers because boxplots are simple to create and allow for easy detection of outliers (Middleton, 2022a). Figure 4 shows three of the boxplots created. It is clear that the first boxplot (Zip) does not have outliers because there are no points shown past the minimum and maximum “whiskers”, unlike the second (Lat) and third (Population) boxplots which do have points shown past these minimums and maximums.

Re-expressing Categorical Variables: I used two methods to re-express categorical variables, label encoding and ordinal encoding. Label encoding assigns a numeric value to each row based on alphabetical order. I used this on the variables that had Yes/No binary values. Label encoding assigned a 0 value to inputs of No and a 1 value to inputs of Yes. This works exactly how I want because 0 represents something that is not present and 1 represents something that is present (Middleton, 2022c). I then used ordinal encoding on the Education column. I used this because the Education variable has 12 unique values that can be ordered from least amount of education to most amount of education (Middleton, 2022c).

**C3: Programing Language**

I chose to do this project using Python and Jupyter Notebook. I made this decision because I was already very familiar with Python as a coding language, using the Jupyter Notebook environment.

Packages/Libraries I used for this project include:

* import pandas as pd: I used pandas because it allows me to easily upload a CSV file. Once uploaded pandas allows for easy data cleaning and visualizations.
* import numpy as np: I used numpy because it allows for easy statistical/mathematical calculations of my data. For example, finding a minimum or maximum.
* import matplotlib.pyplot as plt: I used matplotlib.pyplot because it allows me to create visual representations of my data. For example, I created histograms to determine the skew of many of my variables.
* import missingno as msno: I used missingno to create a visualization of the data that was missing from my dataset.
* import seaborn as sns: I used seaborn because it can create very user-friendly visualizations of data and has many different options to customize these visualizations.
* from sklearn.preprocessing import LabelEncoder: I used this during the re-expression of categorical variables process. It easily did label encoding.
* from sklearn.decomposition import PCA: I used this during the PCA process. It helped my find my principle components.

**C4: Detection Code**

\*see code / script attached\*

Duplicates:

df.duplicated()

Missing Values:

df.isnull().sum()

msno.matrix(df, fontsize = 12, labels = True) #(Middleton, 2022a)

Outliers:

caseorder\_boxplot = sns.boxplot(x = "CaseOrder", data = df)

plt.show()

Zip\_boxplot = sns.boxplot(x = "Zip", data = df)

plt.show()

Lat\_boxplot = sns.boxplot(x = "Lat", data = df)

plt.show()

Lng\_boxplot = sns.boxplot(x = "Lng", data = df)

plt.show()

Population\_boxplot = sns.boxplot(x = "Population", data = df)

plt.show()

Children\_boxplot = sns.boxplot(x = "Children", data = df)

plt.show()

Age\_boxplot = sns.boxplot(x = "Age", data = df)

**C4: Detection Code (continued)**

Outliers (continued):

plt.show()

Income\_boxplot = sns.boxplot(x = "Income", data = df)

plt.show()

Outage\_sec\_perweek\_boxplot = sns.boxplot(x = "Outage\_sec\_perweek", data = df)

plt.show()

Email\_boxplot = sns.boxplot(x = "Email", data = df)

plt.show()

Contancts\_boxplot = sns.boxplot(x = "Contacts", data = df)

plt.show()

Yearly\_equip\_failure\_boxplot = sns.boxplot(x = "Yearly\_equip\_failure", data = df)

plt.show()

Tenure\_boxplot = sns.boxplot(x = "Tenure", data = df)

plt.show()

MonthlyCharge\_boxplot = sns.boxplot(x = "MonthlyCharge", data = df)

plt.show()

Bandwidth\_GB\_Year\_boxplot = sns.boxplot(x = "Bandwidth\_GB\_Year", data = df)

plt.show()

item1\_boxplot = sns.boxplot(x = "item1", data = df)

**C4: Detection Code (continued)**

Outliers (continued):

plt.show()

item2\_boxplot = sns.boxplot(x = "item2", data = df)

plt.show()

item3\_boxplot = sns.boxplot(x = "item3", data = df)

plt.show()

item4\_boxplot = sns.boxplot(x = "item4", data = df)

plt.show()

item5\_boxplot = sns.boxplot(x = "item5", data = df)

plt.show()

item6\_boxplot = sns.boxplot(x = "item6", data = df)

plt.show()

item7\_boxplot = sns.boxplot(x = "item7", data = df)

plt.show()

item8\_boxplot = sns.boxplot(x = "item8", data = df)

plt.show() #(Middleton, 2022a)

Re-expressing Categorical Variables:

I did not use code to detect which variables could be re-expressed. I looked at the original data set and the data dictionary provided to decide which variables to re-express.

**Part III: Data Cleaning**

**D1: Discussion of Findings**

Duplicates: I found that none of the rows in my data frame were duplicated.

Missing Values: I found that there were 8 variables with missing/null values.

* Children: 2495 missing/null values
* Age: 2475 missing/null values
* Income: 2490 missing/null values
* Techie: 2477 missing/null values
* Phone: 1026 missing/null values
* TechSupport: 991 missing/null values
* Tenure: 931 missing/null values
* Bandwidth\_GB\_Year: 1021 missing/null values

Outliers: I found that there were 18 variables with outliers.

* Lat: The maximum outlier was 70.64066, and the minimum outlier was 17.96612. There were many outliers on both extremes.
* Lng: There were many outliers past the minimum, but no outliers past the maximum. The smallest value in the dataset was -171.68815.
* Population: There were many outliers past the maximum, but no outliers past the minimum. The largest value in the dataset was 111850.0.

**D1: Discussion of Findings (continued)**

Outliers (continued):

* Children: There were 4 outliers past the maximum, but no outliers past the minimum. The largest value in the dataset was 10. The other three outliers were 7, 8, and 9.
* Income: There were many outliers past the maximum, but no outliers past the minimum. The largest value in the dataset was 258900.7.
* Outage\_sec\_perweek: The maximum outlier was 47.04928, and the minimum outlier was -1.348571. There were many outliers on both extremes.
* Email: The maximum outlier was 23, and the minimum outlier was 1. Other outliers were: 2, 3, 21, and 22.
* Contacts: There were 2 outliers past the maximum, but no outliers past the minimum. The largest value in the dataset was 7. The other outlier was 6.
* Yearly\_equip\_failure: There were 3 outliers past the maximum, but no outliers past the minimum. The largest value in the dataset was 6. The other outlies were 3 and 4.
* MonthlyCharge: There were a few outliers past the maximum, but no outliers past the minimum. The largest value in the dataset was 315.8786.
* item1: The maximum outlier was 7, and the minimum outlier was 1. The other outlier is 6.
* item2: The maximum outlier was 7, and the minimum outlier was 1. The other outlier is 6.

**D1: Discussion of Findings (continued)**

Outliers (continued):

* item3: The maximum outlier was 8, and the minimum outlier was 1. The other outliers were 6 and 7.
* item4: The maximum outlier was 7, and the minimum outlier was 1. The other outlier is 6.
* item5: The maximum outlier was 7, and the minimum outlier was 1. The other outlier is 6.
* item6: The maximum outlier was 8, and the minimum outlier was 1. The other outliers were 6 and 7.
* item7: The maximum outlier was 7, and the minimum outlier was 1. The other outlier is 6.
* item8: The maximum outlier was 8, and the minimum outlier was 1. The other outliers were 6 and 7.

Re-expressing Categorical Variables:

I found that there were 13 Yes/No value columns I could re-express using label encoding. These included: Churn, Techie, Port-modem, Tablet, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, and PaperlessBilling.

I also found that Education would be a good column to use ordinal encoding.

**D2: How and Why each Treatment Method was Used**

Duplicates: Because there were no duplicates, I did not have to use any treatment methods. However, I did want to double-check so I used the code: df.drop\_duplicates() and print(len(df)). This would have dropped any duplicates I may have missed and then returned the length of my new data frame. Since the data frame started with 10,000 rows and still had 10,000 rows after I ran this code I know there were no duplicates.

|  |
| --- |
| Figure 5 |

Missing Values: To treat missing values I first created histograms for the five numeric columns that had missing values (shown in Figure 5). This helped me determine the skew and distribution of each variable. I then used univariate statistical imputation to impute the data.

* Children: The children column is right-skewed; therefore, I used the median to impute the data (Middleton, 2022a). I used the median instead of the mean because the mean is influenced greatly by skewed data, while the median is not.
* Age: The age column has a fairly uniform distribution; therefore, I used the mean to impute the data (Middleton, 2022a). I used the mean instead of the median because this data was not significantly skewed one way.

**D2: How and Why each Treatment Method was Used (continued)**

Missing Values (continued):

* Income: The income column is right-skewed; therefore, I used the median to impute the data (Middleton, 2022a). I used the median instead of the mean because the mean is influenced greatly by skewed data, while the median is not.
* Tenure: The tenure column has a bimodal distribution; therefore, I decided to use the median to impute the data (Middleton, 2022a). I used the median instead of the mean because the mean is influenced greatly by skewed data, while the median is not.
* Bandwidth\_GB\_Year: The Bandwidth\_GB\_Year column has a bimodal distribution; therefore, I decided to use the median to impute the data (Middleton, 2022a). I used the median instead of the mean because the mean is influenced greatly by skewed data, while the median is not.

I cleaned the Techie, Phone, and TechSupport columns using the mode (Middleton, 2022a). I used the mode to impute the data because these columns have text/object values. The median and mean are not valid to use on text values because there is no inherent order or numeric values for these columns. Using the mode replaces each missing value with the most common value for that column.

**D2: How and Why each Treatment Method was Used (continued)**

|  |
| --- |
| Figure 6 |

Outliers:

|  |
| --- |
| Figure 7 |

* Lat: Figure 6 shows the output boxplot that was created from the Lat column. I noticed that the Lat column (Latitude) had a lot of outliers, but I wanted to find if these outliers were mistakes, or if they were possible latitude values. I started by finding the maximum and minimum values for this dataset (70.64066 and 17.96612 respectively). I knew from the data dictionary pdf that all the customers reside in the United States, however, I did not know what latitude values were valid for the US. According to Bathman (2018) both the maximum and minimum latitude values in this dataset are possible for US states and territories. I decided to retain all outliers for the Lat value since it is likely these are all valid inputs.
* Lng: Figure 7 shows the output boxplot that was created from the Lng column. I noticed that the Lng column (longitude) had a lot of outliers, but I wanted to find if these outliers were mistakes, or if they were possible longitude values. I started by finding the maximum and minimum values for this dataset (-65.66785 and

-171.68815 respectively). I knew from the data dictionary pdf that all the customers reside in the United States, however, I did not know what longitude values were valid for the US. According to Bathman (2018) both the maximum and minimum longitude values in this dataset are possible for US states and territories. I decided to retain all outliers for the Lng value since it is likely these are all valid inputs.

**D2: How and Why each Treatment Method was Used (continued)**

|  |
| --- |
| Figure 8 |

Outliers (continued):

* Population: Figure 8 shows the output boxplot that was created from the Population column. I noticed that the Population column had a lot of outliers past the maximum, but I wanted to find if these outliers were mistakes, or if they were possible population values. I started by using df['Population'].describe() to find the maximum, minimum, mean, and median values. I then researched what the greatest population density per square mile is in the United States. I found that New York (the city with the largest population density) has a maximum population density of approximately 27,000 (*Planning-Population-NYC Population Facts – DCP*, n.d.). The largest value in the population column was 111,850 therefore this is likely a mistake. I decided to replace all values greater than 27,000 with the median value. I used the median because the mean would have been affected by the skewed data. To do this I used the code df['Population'] = np.where(df['Population'] >= 27000, 2931, df['Population']) (Singh, 2019).

**D2: How and Why each Treatment Method was Used (continued)**

|  |
| --- |
| Figure 9 |

Outliers (continued):

* Children: Figure 9 shows the output boxplot that was created from the Population column. I decided to retain all outliers from the children column because all inputs were valid. Although it is uncommon for an individual to have 10 children this is still within a plausible range for valid input.

|  |
| --- |
| Figure 10 |

* Income: Figure 10 shows the output boxplot that was created from the Income column. I first wanted to double-check the max amount so I used the code: df['Income'].max(). I found that the max income in this column was: $258,999.70. I decided to retain all outliers from the income column because all inputs were valid. Although it is less common for an individual to have an income at $258,999.70 this is still well within a plausible range for valid input.

|  |
| --- |
| Figure 11 |

* Outage\_sec\_perweek: Figure 11 shows the output boxplot that was created from the Outage\_sec\_perweek column. I first found the maximum (47.04928) and minimum

(-1.348571) values. Since this column is the seconds of outages per week negative values are more than likely a mistake. Therefore, I decided to replace all values less than 0 with the median value. I used the median because the mean would have been affected by the skewed data. To do this I used the code df['Outage\_sec\_perweek'] = np.where(df['Outage\_sec\_perweek'] < 0, 10.214231, df['Outage\_sec\_perweek']) (Singh, 2019).

**D2: How and Why each Treatment Method was Used (continued)**

|  |
| --- |
| Figure 12 |

Outliers (continued):

|  |
| --- |
| Figure 13 |

* Email: Figure 12 shows the output boxplot that was created from the Email column. I decided to retain all outliers from the Email column because all outliers are in a plausible range for emails sent.
* Contacts: Figure 13 shows the output boxplot that was created from the Contacts column. I decided to retain all outliers from the Contacts column because all outliers are in a plausible range for contacts made.

|  |
| --- |
| Figure 14 |

* Yearly\_equip\_failure: Figure 14 shows the output boxplot that was created

from the Yearly\_euip\_failure column. I decided to retain all outliers from the Yearly\_equip\_failure column because all outliers are in a plausible range for the number of times the equipment failed and had to be repaired or replaced in a year.

**D2: How and Why each Treatment Method was Used (continued)**

|  |
| --- |
| Figure 15 |

Outliers (continued):

|  |
| --- |
| Figure 16 |

* MonthlyCharge: Figure 15 shows the output boxplot that was created from the MonthlyCharge column. I first found the maximum was 315.8786. I decided to retain all outliers from the MonthlyCharge column because all outliers are in a plausible range for the average monthly charge.
* item1: Figure 16 shows the output boxplot that was created from the item1 column. I decided to retain all outliers from the item1 column because the options of the survey were between 1 and 8.

|  |
| --- |
| Figure 17 |

* item2: Figure 17 shows the output boxplot that was created from the item2 column. I decided to retain all outliers from the item2 column because the options of the survey were between 1 and 8.

|  |
| --- |
| Figure 18 |

* item3: Figure 18 shows the output boxplot that was created from the item3 column. I decided to retain all outliers from the item3 column because the options of the survey were between 1 and 8.

**D2: How and Why each Treatment Method was Used (continued)**

|  |
| --- |
| Figure 19 |

Outliers (continued):

* item4: Figure 19 shows the output boxplot that was created from the item4 column. I decided to retain all outliers from the item4 column because the options of the survey were between 1 and 8.

|  |
| --- |
| Figure 20 |

* item5: Figure 20 shows the output boxplot that was created from the item5 column. I decided to retain all outliers from the item5 column because the options of the survey were between 1 and 8.

|  |
| --- |
| Figure 21 |

* item6: Figure 21 shows the output boxplot that was created from the item6 column. I decided to retain all outliers from the item6 column because the options of the survey were between 1 and 8.

|  |
| --- |
| Figure 22 |

* item7: Figure 22 shows the output boxplot that was created from the item7 column. I decided to retain all outliers from the item7 column because the options of the survey were between 1 and 8.

|  |
| --- |
| Figure 23 |

* item8: Figure 23 shows the output boxplot that was created from the item8 column. I decided to retain all outliers from the item8 column because the options of the survey were between 1 and 8.

**D2: How and Why each Treatment Method was Used (continued)**

Re-expressing Categorical Variables:

* Label Encoding: I used label encoding because many of the columns in this dataset had binary inputs of Yes/No. Label encoding assigns a numeric value to each row based on alphabetical order. Therefore, it assigned a 0 value to inputs of No and a 1 value to inputs of Yes. This works exactly how I want because 0 represents something that is not present and 1 represents something that is present (Middleton, 2022c).
* Ordinal Encoding: Ordinal encoding allows you to assign a numeric value to each unique value in a column based on the perceived real-world order. I used ordinal encoding on the Education column because the Education variable has 12 unique values that can be ordered from least amount of education to most amount of education (Middleton, 2022c).

|  |
| --- |
| Figure 24 |

**D3: Summary of Work**

The dataset is now clean and ready for analysis. There are no duplicates or missing values. Outliers have either been changed or have been retained and noted. A few categorical columns have also been re-expressed for analysis in the future.

Missing Values: Figure 24 shows the output after re-running the code: df.isnull().sum(). All values are zero, therefore there are no missing values in the dataset after my cleaning.

|  |
| --- |
| Figure 25 |
|  |
| Figure 26 |

**D3: Summary of Work (continued)**

Outliers: I retained all outliers except in the Population and Outage\_sec\_perweek. Figure 25 shows the new data in the population variable after I cleaned the non-possible outliers. Figure 26 shows the new data in the outage\_sec\_perweek variable after I cleaned the non-possible outliers.

**D4: Code to Treat Data**

\*see code / script attached\*

Duplicates:

df.drop\_duplicates()

print(len(df))

Missing Values:

df.hist(column=["Children", "Age", "Income", "Tenure", "Bandwidth\_GB\_Year"], figsize=(12,8))

df['Children'].fillna(df['Children'].median(), inplace = True)

print(df['Children'].isnull().sum())

plt.hist(df['Children'])

plt.show()

df['Age'].fillna(df['Age'].mean(), inplace = True)

**D4: Code to Treat Data (continued)**

Missing Values (continued):

print(df['Age'].isnull().sum())

plt.hist(df['Age'])

plt.show()

df['Income'].fillna(df['Income'].median(), inplace = True)

print(df['Income'].isnull().sum())

plt.hist(df['Income'])

plt.show()

df['Tenure'].fillna(df['Tenure'].median(), inplace = True)

print(df['Tenure'].isnull().sum())

plt.hist(df['Tenure'])

plt.show()

df['Bandwidth\_GB\_Year'].fillna(df['Bandwidth\_GB\_Year'].median(), inplace = True)

print(df['Bandwidth\_GB\_Year'].isnull().sum())

plt.hist(df['Bandwidth\_GB\_Year'])

plt.show()

df['Techie'] = df['Techie'].fillna(df['Techie'].mode()[0])

print(df['Techie'].isnull().sum())

df['Phone'] = df['Phone'].fillna(df['Phone'].mode()[0])

print(df['Phone'].isnull().sum())

df['TechSupport'] = df['TechSupport'].fillna(df['TechSupport'].mode()[0])

print(df['TechSupport'].isnull().sum())

df.isnull().sum()

**D4: Code to Treat Data (continued)**

Outliers:

print(df['Lat'].max())

print(df['Lat'].min())

print(df['Lng'].max())

print(df['Lng'].min())

print(df['Population'].describe())

df['Population'] = np.where(df['Population'] >= 27000, 2931, df['Population']) #(Singh, 2019)

print(df['Population'].describe())

Population\_boxplot = sns.boxplot(x = "Population", data = df).set\_title("Population")

plt.show()

print(df['Income'].max())

print(df['Outage\_sec\_perweek'].max())

print(df['Outage\_sec\_perweek'].min())

df['Outage\_sec\_perweek'].describe()

df['Outage\_sec\_perweek'] = np.where(df['Outage\_sec\_perweek'] < 0, 10.214231, df['Outage\_sec\_perweek']) #(Singh, 2019)

print(df['Outage\_sec\_perweek'].describe())

Outage\_sec\_perweek\_boxplot = sns.boxplot(x = "Outage\_sec\_perweek", data = df).set\_title("Outage\_sec\_perweek")

**D4: Code to Treat Data (continued)**

Outliers (continued):

plt.show()

print(df['MonthlyCharge'].max())

Re-expressing Categorical Variables:

labelencoder = LabelEncoder() #(By Great Learning Team - , 2022)

df['Churn'] = labelencoder.fit\_transform(df['Churn'])

print(df['Churn'].head())

df['Techie'] = labelencoder.fit\_transform(df['Techie'])

df['Port\_modem'] = labelencoder.fit\_transform(df['Port\_modem'])

df['Phone'] = labelencoder.fit\_transform(df['Phone'])

print(df['Education'].unique())

print(len(df['Education'].unique()))

df['Education\_numeric'] = df['Education'] #(Middleton, 2022c)

dict\_edu = {'Education\_numeric': {'No Schooling Completed': 0, 'Nursery School to 8th Grade': 1, '9th Grade to 12th Grade, No Diploma': 2, 'GED or Alternative Credential': 3, 'Regular High School Diploma': 4, 'Some College, Less than 1 Year': 5, 'Some College, 1 or More Years, No Degree': 6, "Associate's Degree": 7, 'Professional School Degree': 8, "Bachelor's Degree": 9, "Master's Degree": 10, 'Doctorate Degree': 11}}

df.replace(dict\_edu, inplace = True)

df['Education\_numeric'].unique()

**D5: CSV File**

See attached document (D206\_PA.csv)

**D6: Disadvantages of Cleaning/Treatment Methods**

Duplicates: In the context of the Telecommunications company there are no real disadvantages to dropping data. There is no valid reasoning for all the information to be identical in two rows. However, dropping duplicate values could be a disadvantage in some situations where both rows contain valid information. For example, if two students at a school have the same name. In this situation dropping one of the duplicates would be dropping valid information from the dataset.

Missing Values: I used univariate imputation to fill in all missing values. This means I imputed the values with either the mean, median, or mode from the variable. The biggest disadvantage of this method is that it can distort the distribution of the data. Although this is a good estimate of what the value may be, it is likely not 100% accurate.

Outliers: I used two methods for treating outliers the first was to retain outliers that were possible values for the variable. The disadvantage of this is that there are still outliers that can make future analysis more difficult and the dataset has a wider range of variation. The other method I used was replacement. I replaced values in both the population and the outage\_sec\_perweek columns. One disadvantage of the replacement is that it may distort the data. Because the data that I replaced is likely invalid, replacing it with the median is a good estimate of what the value might be, but it is likely not 100% accurate.

**D6: Disadvantages of Cleaning/Treatment Methods (continued)**

Re-expressing Categorical Variables: Using ordinal encoding to re-express categorical variables does not have many disadvantages because it does not change the data, it just adds a new column with more data. The only disadvantage of this is that it can make the data frame larger and hard to work with. Label encoding does change the column values so a disadvantage of this is that it may be harder for some to understand.

**D7: Future Challenges**

The biggest challenge for a future data analyst using this data would be that many columns have changed data from the cleaning of missing data and outliers. A future analyst would have to double-check that their finding were valid and were not significantly impacted by this changed data. A future analyst would also have to be aware of the outliers that were kept in this dataset. The outliers would make various applications more difficult in the future.

**E1: PCA**

Variables used for PCA: Zip, Lat, Lng, Population, Children, Age, Income, Outage\_sec\_perweek, Email, Contacts, Yearly\_equip\_failure, Tenure, MonthlyCharge, Bandwidth\_GB\_Year, item1, item2, item3, item4, item5, item6, item7, item8.

Figure 27 shows the PCA loadings matrix output.

|  |
| --- |
| Figure 27 |

**E2: Variables to keep after PCA**

|  |
| --- |
| Figure 28 |

Using the Kaiser Rule, PCs with eigenvalues greater than 1 should be kept (Middleton, 2022b). This means the first 9 PCs should be kept. The eigenvalue of a PC represents the amount of variance that is explained by that PC in the original data. The greater the eigenvalue, the more variation, and therefore it will tell us more about the data.

**E3: Benefit of PCA**

PCA allows for faster computation, which is better for machine learning. PCA also reduces redundancy. If there are variables that are highly correlated, they can be combined into one PC (Middleton, 2022b).

**Part IV: Supporting Documents**

**F: Panopto Recording**

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=d6655067-73dc-464a-a030-aea601411a2f>

**G: Third Party Code References**

By Great Learning Team -. (2022, May 5). *What is Label Encoding in Python | Great Learning*. GreatLearning Blog. https://www.mygreatlearning.com/blog/label-encoding-in-python/

Middleton, K. (2022a). *Getting Started with “Missing Data and Outliers”* [Slides]. My.Sharepoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FD206%2DMissingDataandOutliersPresentationSlides%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

Middleton, K. (2022b). *“Getting Started with PCA”* [Slides]. My.SharePoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FUnderstanding%20PCA%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

Middleton, K. (2022c). *Getting Started with "Re-expression of Categorical Variables* [Slides]. My.Sharepoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FRe%2DexpressionCategoricalVariablesPresentationSlides%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocumentF%2FD206&ga=1

Singh, D. (2019, October 22). *Cleaning up Data from Outliers*. PluralSight. https://www.pluralsight.com/guides/cleaning-up-data-from-outliers

**H: Content References**

Bathman, J. (2018). *The approximate max/min latitude and longitude for all states and major*

*territories*. GetHub Gist.

Middleton, K. (2022a). *Getting Started with “Missing Data and Outliers”* [Slides]. My.Sharepoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FD206%2DMissingDataandOutliersPresentationSlides%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

Middleton, K. (2022b). *“Getting Started with PCA”* [Slides]. My.SharePoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FUnderstanding%20PCA%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

Middleton, K. (2022c). *Getting Started with "Re-expression of Categorical Variables* [Slides]. My.Sharepoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FRe%2DexpressionCategoricalVariablesPresentationSlides%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

*Planning-Population-NYC Population Facts - DCP*. (n.d.). NYC Planning. https://www1.nyc.gov/site/planning/planning-level/nyc-population/population-facts.page#:%7E:text=About%201%20in%20every%2038,arrived%20in%202000%20or%20later.

# References

Bathman, J. (2018). *The approximate max/min latitude and longitude for all states and major*

*territories*. GetHub Gist.

https://gist.github.com/jakebathman/719e8416191ba14bb6e700fc2d5fccc5

By Great Learning Team -. (2022, May 5). *What is Label Encoding in Python | Great Learning*. GreatLearning Blog. https://www.mygreatlearning.com/blog/label-encoding-in-python/

Middleton, K. (2022a). *Getting Started with “Missing Data and Outliers”* [Slides]. My.Sharepoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FD206%2DMissingDataandOutliersPresentationSlides%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

Middleton, K. (2022b). *“Getting Started with PCA”* [Slides]. My.SharePoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FUnderstanding%20PCA%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

Middleton, K. (2022c). *Getting Started with "Re-expression of Categorical Variables* [Slides]. My.Sharepoint. https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona\_middleton\_wgu\_edu/\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FRe%2DexpressionCategoricalVariablesPresentationSlides%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1

*Planning-Population-NYC Population Facts - DCP*. (n.d.). NYC Planning. https://www1.nyc.gov/site/planning/planning-level/nyc-population/population-facts.page#:%7E:text=About%201%20in%20every%2038,arrived%20in%202000%20or%20later.

Singh, D. (2019, October 22). *Cleaning up Data from Outliers*. PluralSight. https://www.pluralsight.com/guides/cleaning-up-data-from-outliers