**Table of Contents**

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

**A. Research Problem and Variable Description ........................ 3**

**Proposed Research Question ............................................. 3**

**Variables and Description ................................................ 3**

**Part II: Data Cleaning Plan………………………….4**

**Plan for Cleaning Data**

**Techniques and Steps for Data Quality Assessment .......... 4**

**Duplicates ............................................................... 4**

**Nulls ................................................................. 4**

**Outliers .............................................................. 4**

**Re-expressing Categorical Variables .......................... 4**

**Justification for Data Quality Assessment Approach ......... 4**

**Assessing Data**

**Duplicates ............................................................... 5**

**Missing Values ......................................................... 5**

**Outliers ............................................................... 5**

**Justification for Selected Programming Language and Libraries .... 5**

**Part III: Data Cleaning**

**Data-Cleaning Summary**

**Findings - Data Quality Issues ......................................... 6**

**Duplicates ............................................................... 6**

**Missing Values ......................................................... 6**

**Outliers ............................................................... 6**

**Justification of Methods ................................................ 7**

**Handling Missing Values ............................................. 7**

**Handling Outliers .................................................... 9**

**Summary of Work Performed ..........................................10**

**Limitations of Cleaning Data ........................................... 14**

**Detected and Mitigated Code .......................................... 15**

**Annotated Code ......................................................... 15**

**Cleaned CSV ........................................................... 15**

**Principal Component Analysis (PCA) ................................ 15**

**Variables Used for PCA ................................................. 15**

**Scree Plot and Eigenvalues ........................................... 15**

**Importance of PCA to Organizations ................................. 15**

**Part IV: Supporting Documents**

**References .................................................................. 16**

**Part I: Research Question**

A. Research Problem and Variable description

1. Proposed research question: “Do customer demographics affect customer churn?”

2. Variables and description:



**Part II: Data-Cleaning Plan**

**1.**

Duplicates

The .duplicates() method will be used to detect duplicates in the data set. The function .duplicates() on its own will search for complete duplicate rows in the data set. While this will detect full duplicated rows, I plan to also check specific columns for duplicates, such as name or customer\_id, independently. This will give a more complete picture of potential duplicate records where only a few values are different. While some duplicates on specific values may occur, they may not represent true duplicate records and will require comparing the complete rows. (<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6eedfad4-240e-4c5c-8eab-b058003d3e6b>)

Nulls

The .isna() method will be used to check for NULL values in the data set. .isna() will return a Boolean value for each data point, whether it is a NULL (True) or not (False). The method .isna() can be combine with .sum() to return the total count of NULL values per column. I also plan to use Missingno to give a graphical representation of missing values and to compare which rows are missing values. Missngno = help graphically reveal nulls in the data set as well, also helping to reveal in missing data has any relation to other missing data.

(<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=767749d2-ba19-4f94-bec8-b058017b2f5e>)

Outliers:

For finding outliers, I will use multiple approaches. Using the scipy method .zscore() will generate z-scores for numerical columns and show outliers (values with z-scores >= 3

or <= -3). I will also incorporate histograms through the Matplotlib .hist() function and box-plots from Seaborn to identify outliers graphically. I plan on using all three methods in conjunction (<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=19c24c56-0f37-408e-bb1f-b059002a77ac>)

[(Larose,](https://eds.p.ebscohost.com/eds/ebookviewer/ebook?sid=33470f2f-3e38-423c-8508-23f056395d74%40redis&ppid=pp_29&vid=0&format=EB) 2019)

Re-expressing Categorical Variables

I plan to keep categorical variables as they are (strings, Boolean, etc.) unless I find variables that are not consistent throughout the column (i.e. mixing Boolean with 1 and 0, or yes/no.)

**2. Assessing The Quality of the Data**

Duplicates

For duplicates, I am primarily checking for complete duplicate records. These would be most commonly duplicates created by mistake, perhaps by combining multiple data sets with overlapping rows. Next, it would be important to check for duplicates that only share some data. For example, you may have a customer from two different time periods listed twice for different periods. You would expect some data to be the same, such as customer\_id and even some demographic data, and other information to be different, like age and tenure. If two records are identified as being the same person, it would then need to be decided if the records should be merged or kept separate.

Missing Values

The data set contains a diverse set of variables and treating nulls will be done in a way to preserve the distribution of the data whenever possible. This means, the first step for numeric variables will be to identify the distribution. For normal and uniform distribution, null values will be imputed using the mean. For skewed distributions, the median value will be imputed and for bimodal distribution, the mode will used for imputation.

For the categorical variables, a more nuanced approach will be used. Some of the self-reported questions are asked as “yes” or “no” questions. For these, I will evaluate whether a null value should represent a “no” or not confirming a question. Other data will be imputed in a way to best preserve the distribution.

Outliers

When checking the data set for outliers, I want to present a few different methods. Z-score is a good way to see numerically which values are distant from the mean, while histograms and boxplots can give a visual representation of distribution. Having access to all these methods will help to determine if the outliers need to be adjusted or not.

**3. Justify your selected programming language and any libraries and packages that will support the data-cleaning process.**

I chose to use Python for this project because it is an easily readable and deployable language that s common among data engineers and developers. I also used the following packages for specific purposes:

Pandas – Support the use of dataframes to manage rows and columns of data

Numpy – for advanced Math function in Python

Scipy – Use of advanced statistical analysis (mean, median, etc.)

Missingno – Identifying null values in a graphical display

Matplotlib – for Easy to use and effective visualizations

Seaborn – A package that can easily display a box and whisker plot to visualize distribution.

C. Data-cleaning Summary

1. Findings - Data Quality Issues

Duplicates

The data set was free from duplicated rows. Where there were duplicate values (city, zip code, age, etc.), it was clear that they were appropriate and did not represent any erroneously duplicated data. No adjustments were needed.

Missing Values

In total, there were 9 columns with null values. Age, Children, Income, Techie, InternetService, Phone, TechSupport, Tenure, and Bandwidth\_GB\_Year.

churn\_data\_null\_rows = churn\_data[['Age', 'Children' , 'Income', 'Techie', 'InternetService', 'Phone', 'TechSupport', 'Tenure', 'Bandwidth\_GB\_Year']]

churn\_data\_null\_rows.isna().sum()

Null Values by row:

Age 2475

Children 2495

Income 2490

Techie 2477

InternetService 2129

Phone 1026

TechSupport 991

Tenure 931

Bandwidth\_GB\_Year 1021

dtype: int64

Outliers

Outliers were detected after the null values had been treated. The variables with outliers were:

Population – 219 outliers with a range of 53,098 – 111,850

Income – 193 outliers with a range of 112,687.70 – 258,900.70

Children – 451 outliers with a range of 7 – 10

Outage\_sec\_perweek – 491 outliers with a range of 32.581260 - 47.049280

Email – 12 outliers with a range of 1-2 and 22-23

Contacts – 165 outliers with a range of 4 – 7

Yearly\_equip\_failure – 94 outliers with a range of 3 – 6

MonthlyCharge – 3 outliers with a range of 306.268000 – 315.878600

2. Justification of Methods

Missing Values

Age – A numeric variable with the age of each customer. The variable showed to have a uniform distribution. Since the distribution was uniform, the null values were imputed with the mean.

A blue graph with numbers

Description automatically generated

churn\_data['Age'].fillna(churn\_data['Age'].mean(), inplace=True)

Children – A numeric variable that has a right skewed distribution. Since the data was skewed, the null values were imputed using the median.

A blue graph with white background

Description automatically generated

churn\_data['Children'].fillna(churn\_data['Children'].median(), inplace=True)

Income – A numeric variable with right skewed distribution. Since it was skewed, the nulls were imputed using the median.

A blue and white graph

Description automatically generated

churn\_data['Income'].fillna(churn\_data['Income'].median(), inplace=True)

Techie – This is a categorical variable that asks the customer to select “Yes” or “No”. For this variable, I assumed that any non-answer was equivalent to a negative answer and Imputed nulls with “No”.

churn\_data['Techie'] = churn\_data['Techie'].fillna('No')

Internet Service – This is a categorical variable with values of “Fiber Optic” or “DSL”. To preserve the distribution as best as possible, a backfill method was used to impute the nulls. Imputing with the Mode seemed to skew the data too much and backfill preserved the distribution better.

churn\_data['InternetService'].fillna(churn\_data['InternetService'].backfill())

Phone - This is a categorical variable that asks the customer to select “Yes” or “No”. For this variable, I assumed that any non-answer was equivalent to a negative answer and Imputed nulls with “No”.

churn\_data['Phone'] = churn\_data['Phone'].fillna('No')

TechSupport - This is a categorical variable that asks the customer to select “Yes” or “No”. For this variable, I assumed that any non-answer was equivalent to a negative answer and Imputed nulls with “No”.

churn\_data['TechSupport'] = churn\_data['TechSupport'].fillna('No')

Tenure – This variable is numeric and has a bi-modal distribution. Since it is bi-modal, I imputed the null values with the mode.

A graph with blue bars

Description automatically generated with medium confidence

churn\_data['Tenure'].fillna(churn\_data['Tenure'].mode()[0])

Bandwidth\_GB\_Year – This variable is numeric and has a bi-modal distribution. Since it is bi-modal, I imputed the null values with the mode.

A graph with blue bars

Description automatically generated with medium confidence

churn\_data['Bandwidth\_GB\_Year'].fillna(churn\_data['Bandwidth\_GB\_Year'].mode()[0])

Outliers

No corrections were made for outliers in the data set. Every variable that contained outliers was examined to assess the quality of the values. All the values that were outliers seemed to be within normal, expected ranges for the variable. Two examples of this, Income and Children had values that could be considered on the high side, but were still within a realistic range:

Children – 451 outliers with a range of 7 – 10 (children in the household)

Income – 193 outliers with a range of 112,687.70 – 258,900.70 (income per year)

The other variables followed a similar pattern (outliers with realistic values for the variable), and none have obviously erroneous values. In fact, these outliers may be important to preserve, depending on what further analysis will be done with the data set.

3. Summary of work performed

Cleaning this data required only two phases, dealing with missing values and identifying outliers. There were no duplicates in the data set and no categorical variables needed to be re-expressed numerically.

Missing values were imputed using the means describes above and were all successfully handled. Below is the result of “churn\_data.isna().sum()” function before and after the missing values were imputed:

**Before imputing nulls**

Unnamed: 0 0

CaseOrder 0

Customer\_id 0

Interaction 0

City 0

State 0

County 0

Zip 0

Lat 0

Lng 0

Population 0

Area 0

Timezone 0

Job 0

Children 2495

Age 2475

Education 0

Employment 0

Income 2490

Marital 0

Gender 0

Churn 0

Outage\_sec\_perweek 0

Email 0

Contacts 0

Yearly\_equip\_failure 0

Techie 2477

Contract 0

Port\_modem 0

Tablet 0

InternetService 2129

Phone 1026

Multiple 0

OnlineSecurity 0

OnlineBackup 0

DeviceProtection 0

TechSupport 991

StreamingTV 0

StreamingMovies 0

PaperlessBilling 0

PaymentMethod 0

Tenure 931

MonthlyCharge 0

Bandwidth\_GB\_Year 1021

item1 0

item2 0

item3 0

item4 0

item5 0

item6 0

item7 0

item8 0

**After imputing nulls**

Unnamed: 0 0

CaseOrder 0

Customer\_id 0

Interaction 0

City 0

State 0

County 0

Zip 0

Lat 0

Lng 0

Population 0

Area 0

Timezone 0

Job 0

Children 0

Age 0

Education 0

Employment 0

Income 0

Marital 0

Gender 0

Churn 0

Outage\_sec\_perweek 0

Email 0

Contacts 0

Yearly\_equip\_failure 0

Techie 0

Contract 0

Port\_modem 0

Tablet 0

InternetService 0

Phone 0

Multiple 0

OnlineSecurity 0

OnlineBackup 0

DeviceProtection 0

TechSupport 0

StreamingTV 0

StreamingMovies 0

PaperlessBilling 0

PaymentMethod 0

Tenure 0

MonthlyCharge 0

Bandwidth\_GB\_Year 0

item1 0

item2 0

item3 0

item4 0

item5 0

item6 0

item7 0

item8 0

population\_z 0

age\_z 0

income\_z 0

outage\_z 0

email\_z 0

contacts\_z 0

failure\_z 0

MonthlyCharge\_z 0

dtype: int64

Furthermore, the distribution of the imputed variables remained the same:

Age Before Imputation

A blue graph with numbers

Description automatically generated

Age After Imputation

A blue graph with numbers

Description automatically generated

Children Before Imputation

A blue graph with white background

Description automatically generated

Children After Imputation

A blue graph with numbers

Description automatically generated

Tenure Before imputation

A graph with blue bars

Description automatically generated with medium confidence

Tenure After Imputation

A graph with blue bars

Description automatically generated with medium confidence

Bandwidth\_GB\_Year Before Imputation

A graph with blue bars

Description automatically generated with medium confidence

Bandwidth\_GB\_Year After Imputation

A graph with blue bars

Description automatically generated with medium confidence

Since Outliers in this data set fell within reasonable and expected ranges, no variables with outliers had to be adjusted. The outliers themselves may be important to future analysis using the data set and are considered cleaned. No erroneous or problematic outliers were detected in the data set and none needed correction.

4. Limitations of cleaning data

The largest limitation of cleaning this data set came from the missing values in the self-reported variables. Because the answers could be left blank, there were some variables that had null values. Also, since the questions are self-reported on a survey, some of them could not be scraped from other sources. While these variables did not have an overabundance of nulls, if they were a specific component of a business question, they may make some of the rows unreliable.

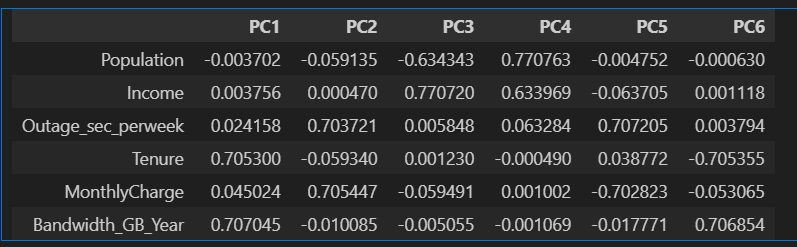
D. Detected and mitigated code

1. Annotated code – *See code attached.* “D206\_Jon\_Cooke\_Churn\_Data\_PA.ipynb”

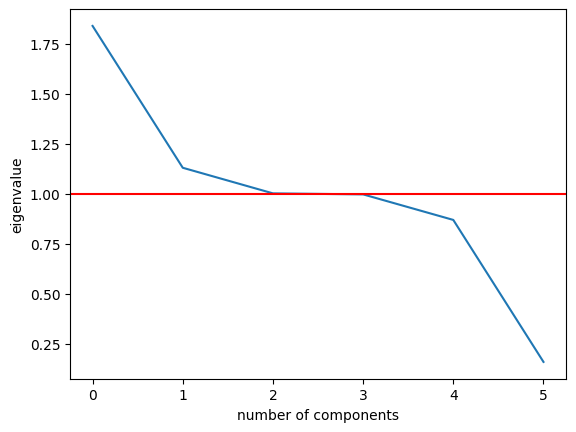
2. Cleaned CSV – *See attached* “D206\_Jon\_Cooke\_PA.csc”

E. Principal Component Analysis

1. There were six (6) variables used for the PCA: 'Population', 'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year'



2. According to the scree plot, PC1, PC2, PC3 and PC4 should be kept since their Eigenvalues are >= 1.



3. PCAs, like the one above, could be important to organizations for many reasons. A PCA will help reduce variables in a data set, making visualization and analysis easier and requires less storage space. The PCA process also filters out irrelevant values, reducing the required computing resources and increasing cost-effectiveness for an organization through efficiency.

References

**Lectures**

Duplicates

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6eedfad4-240e-4c5c-8eab-b058003d3e6b>

Missing Values

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=767749d2-ba19-4f94-bec8-b058017b2f5e>

Outliers

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=19c24c56-0f37-408e-bb1f-b059002a77ac>

PCA

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3bcc452f-fa35-43be-b69f-b05901356f95>

**Textbook**

Chantal D. Larose, & Daniel T. Larose. (2019). Data Science Using Python and R. Wiley.