D206 Data Cleaning

Scott Babcock WGU - MS, Data Analytics

Created: September 9 2025

Libraries Used

```
library(dplyr)
library(ggplot2)
library(naniar)
library(visdat)
library(factoextra)
library(rstatix)
```

Initial Data Load

```
# Load CSV file and assign it to a data frame, also create an original copy to refer back to
churn_df_orig <-read.csv("churn_raw_data.csv")
churn_df <- churn_df_orig
# Rename "Item" columns to more relevant names
churn_df <- churn_df %>%
  rename(
    timely_response = item1,
    timely_fix = item2,
    timely_replacement = item3,
    reliability = item4,
    options = item5,
    respectful_response = item6,
    courteous_exhange = item7,
    active_listening = item8 )
# Turn off scientific notation for labeling [In-text citation: (Bobbitt, 2021)]
options(scipen=999)
```

A1, Research Question

What customer factors influence the length of tenure?

A2, Variables

A table of variables with descriptions and examples was included in the submission but omitted from the code file.

B1, Plan

I used the R programming language to identify data quality issues and perform the data cleaning. To familiarize myself with the dataset, I used the str() function to profile the data, see all the data types, and get a preview. I wanted to ensure this aligned with my understanding of the data from the dictionary provided.

```
# Profile the data, see data types and initial observations
str(churn_df)
```

```
## 'data.frame':    10000 obs. of  52 variables:
##  $ X                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ CaseOrder         : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
##  $ Customer_id         : chr  "K409198" "S120509" "K191035" "D90850" ...
##  $ Interaction         : chr  "aa90260b-4141-4a24-8e36-b04ce1f4f77b" "fb76459f-c047-4a9d-8af9-e0f7d4a
##  $ City                : chr  "Point Baker" "West Branch" "Yamhill" "Del Mar" ...
##  $ State               : chr  "AK" "MI" "OR" "CA" ...
##  $ County              : chr  "Prince of Wales-Hyder" "Ogemaw" "Yamhill" "San Diego" ...
##  $ Zip                 : int  99927 48661 97148 92014 77461 31030 37847 73109 34771 45237 ...
##  $ Lat                 : num  56.3 44.3 45.4 33 29.4 ...
##  $ Lng                 : num  -133.4 -84.2 -123.2 -117.2 -95.8 ...
##  $ Population          : int  38 10446 3735 13863 11352 17701 2535 23144 17351 20193 ...
##  $ Area                : chr  "Urban" "Urban" "Urban" "Suburban" ...
##  $ Timezone            : chr  "America/Sitka" "America/Detroit" "America/Los_Angeles" "America/Los_An
##  $ Job                 : chr  "Environmental health practitioner" "Programmer, multimedia" "Chief Fin
##  $ Children            : int  NA 1 4 1 0 3 0 2 2 NA ...
##  $ Age                 : int  68 27 50 48 83 83 NA NA 49 86 ...
##  $ Education           : chr  "Master's Degree" "Regular High School Diploma" "Regular High School Di
##  $ Employment          : chr  "Part Time" "Retired" "Student" "Retired" ...
##  $ Income              : num  28562 21705 NA 18925 40074 ...
##  $ Marital             : chr  "Widowed" "Married" "Widowed" "Married" ...
##  $ Gender              : chr  "Male" "Female" "Female" "Male" ...
##  $ Churn               : chr  "No" "Yes" "No" "No" ...
##  $ Outage_sec_perweek  : num  6.97 12.01 10.25 15.21 8.96 ...
##  $ Email               : int  10 12 9 15 16 15 10 16 20 18 ...
##  $ Contacts            : int  0 0 0 2 2 3 0 0 2 1 ...
##  $ Yearly_equip_failure: int  1 1 1 0 1 1 1 0 3 0 ...
##  $ Techie              : chr  "No" "Yes" "Yes" "Yes" ...
##  $ Contract            : chr  "One year" "Month-to-month" "Two Year" "Two Year" ...
##  $ Port_modem          : chr  "Yes" "No" "Yes" "No" ...
##  $ Tablet              : chr  "Yes" "Yes" "No" "No" ...
##  $ InternetService     : chr  "Fiber Optic" "Fiber Optic" "DSL" "DSL" ...
##  $ Phone               : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ Multiple            : chr  "No" "Yes" "Yes" "No" ...
##  $ OnlineSecurity      : chr  "Yes" "Yes" "No" "Yes" ...
##  $ OnlineBackup        : chr  "Yes" "No" "No" "No" ...
##  $ DeviceProtection    : chr  "No" "No" "No" "No" ...
##  $ TechSupport         : chr  "No" "No" "No" "No" ...
##  $ StreamingTV         : chr  "No" "Yes" "No" "Yes" ...
##  $ StreamingMovies     : chr  "Yes" "Yes" "Yes" "No" ...
##  $ PaperlessBilling    : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ PaymentMethod       : chr  "Credit Card (automatic)" "Bank Transfer(automatic)" "Credit Card (auto
##  $ Tenure              : num  6.8 1.16 15.75 17.09 1.67 ...
##  $ MonthlyCharge       : num  171 243 159 120 151 ...
##  $ Bandwidth_GB_Year   : num  905 801 2055 2165 271 ...
##  $ timely_response     : int  5 3 4 4 4 3 6 2 5 2 ...
##  $ timely_fix          : int  5 4 4 4 4 3 5 2 4 2 ...
##  $ timely_replacement  : int  5 3 2 4 4 3 6 2 4 2 ...
##  $ reliability         : int  3 3 4 2 3 2 4 5 3 2 ...
##  $ options             : int  4 4 4 5 4 4 1 2 4 5 ...
##  $ respectful_response : int  4 3 3 4 4 3 5 3 3 2 ...
##  $ courteous_exhange   : int  3 4 3 3 4 3 5 4 4 3 ...
##  $ active_listening    : int  4 4 3 3 5 3 5 5 4 3 ...
```

My first detection step was to look for duplicate records. To identify duplicates, I used a combination of the sum() and duplicated() functions on the dataset to determine how many duplicates were present (Duplicates, n.d.). To be sure, I also used the same functions above on each unique identifier column to confirm no

duplicates were present. I did not identify any duplicates in the dataset, but had there been; I would have used the distinct() function to isolate the non-duplicated records.

```
# Detect Duplicate Records [In-text citation:(Getting Started with Duplicates, n.d.)]
sum(duplicated(churn_df))
```

```
## [1] 0
```

```
sum(duplicated(churn_df$CaseOrder))
```

```
## [1] 0
```

```
sum(duplicated(churn_df$Customer_id))
```

```
## [1] 0
```

```
sum(duplicated(churn_df$Interaction))
```

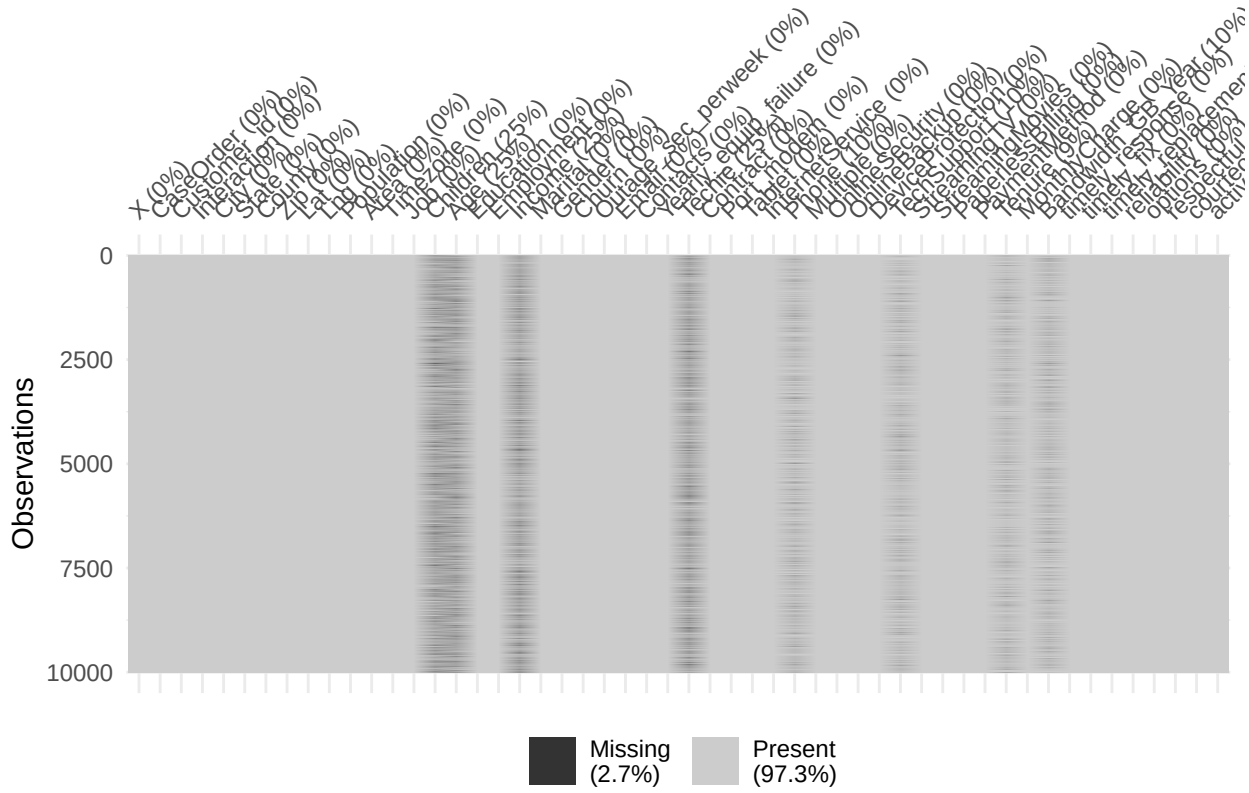
```
## [1] 0
```

Next, I focused on missing values. I used the miss_var_summary() function to list quantitative and qualitative variables with missing data, along with how many missing values were present and the percentage of each variable missing. I identified missing values in eight variables. I used the vis_miss() function to visualize the missing data (Tierney, n.d.).

```
# Detect Missing Values [In-text citation: (Tierney, n.d.)]
miss_var_summary(churn_df)
```

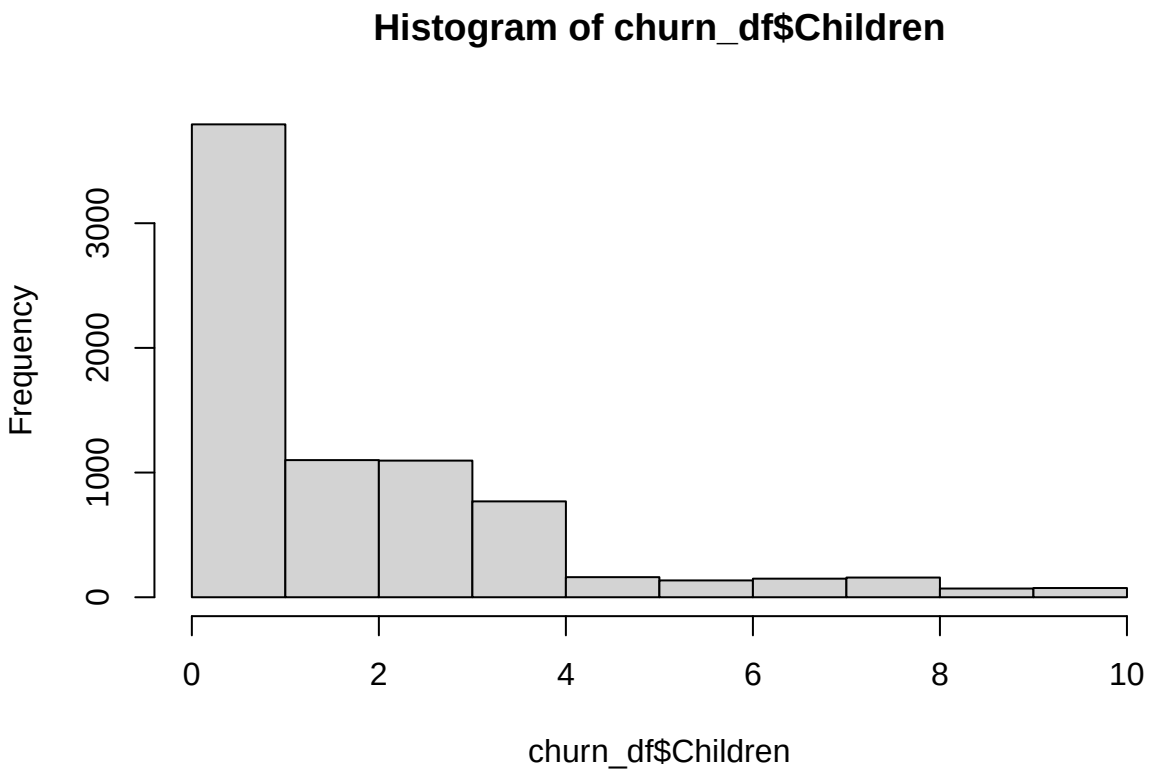
```
## # A tibble: 52 x 3
##    variable          n_miss pct_miss
##    <chr>              <int>    <num>
##  1 Children            2495    25.0
##  2 Income              2490    24.9
##  3 Techie              2477    24.8
##  4 Age                 2475    24.8
##  5 Phone               1026    10.3
##  6 Bandwidth_GB_Year   1021    10.2
##  7 TechSupport          991     9.91
##  8 Tenure               931     9.31
##  9 X                      0     0
## 10 CaseOrder              0     0
## # i 42 more rows
```

```
vis_miss(churn_df)
```

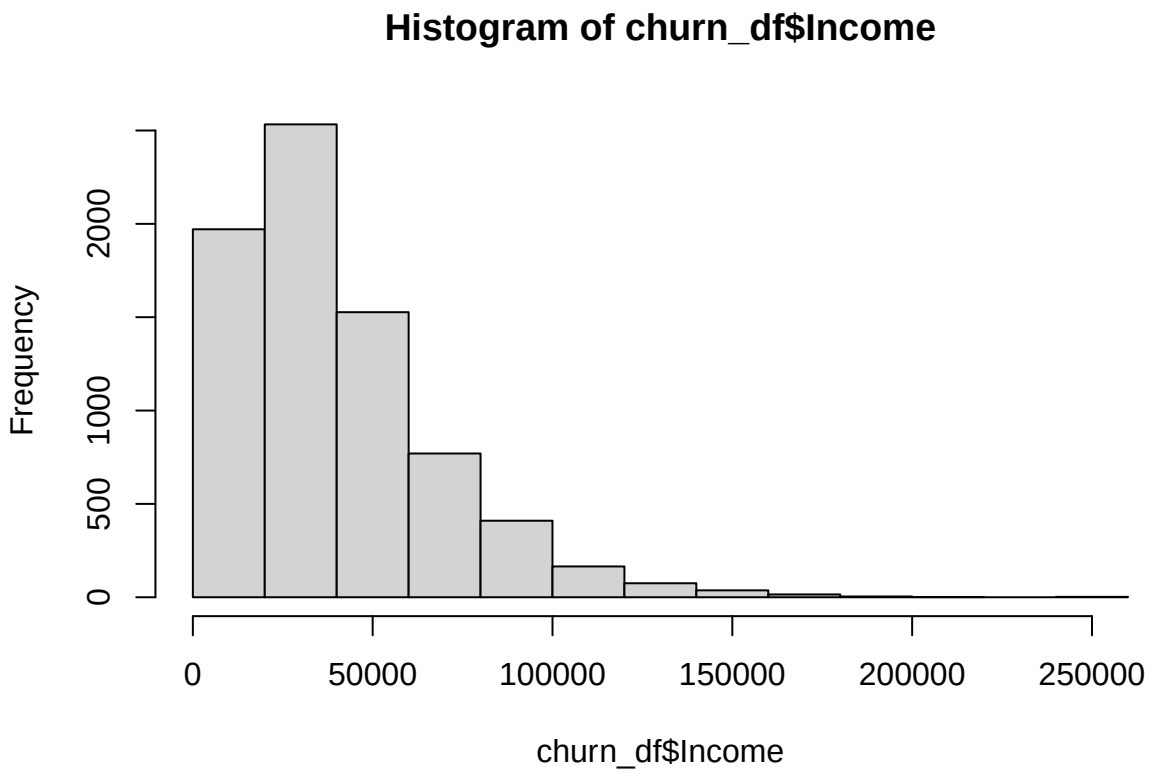


The subsequent detection step was to look for outliers in the data. I created a boxplot for each quantitative variable using the boxplot() function (Outliers, n.d.). After assessing the resulting plots, any values outside the fences on each end would be considered outliers. I detected outliers in 8 variables. There were several categorical variables in the dataset. I examined whether some variables were expressed as yes/no and others as 1/0. Using the unique() function on each of the yes/no variables, I could ensure no mixed values in the fields existed. Everything in the dataset appeared in yes/no format, so I did not feel that any variables would need to be re-expressed. Some survey column names did not provide much context, so I renamed them from 'item' to more applicable names.

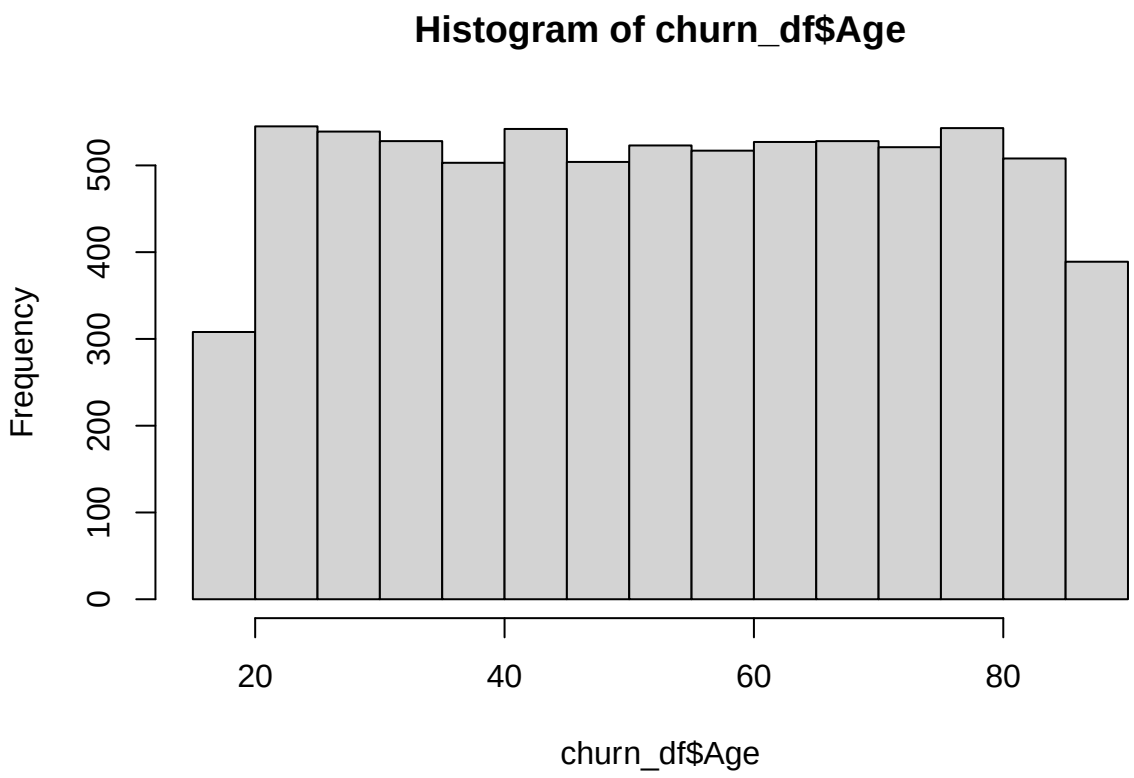
```
# Create histograms for the numeric variables with missing values
hist(churn_df$Children)
```



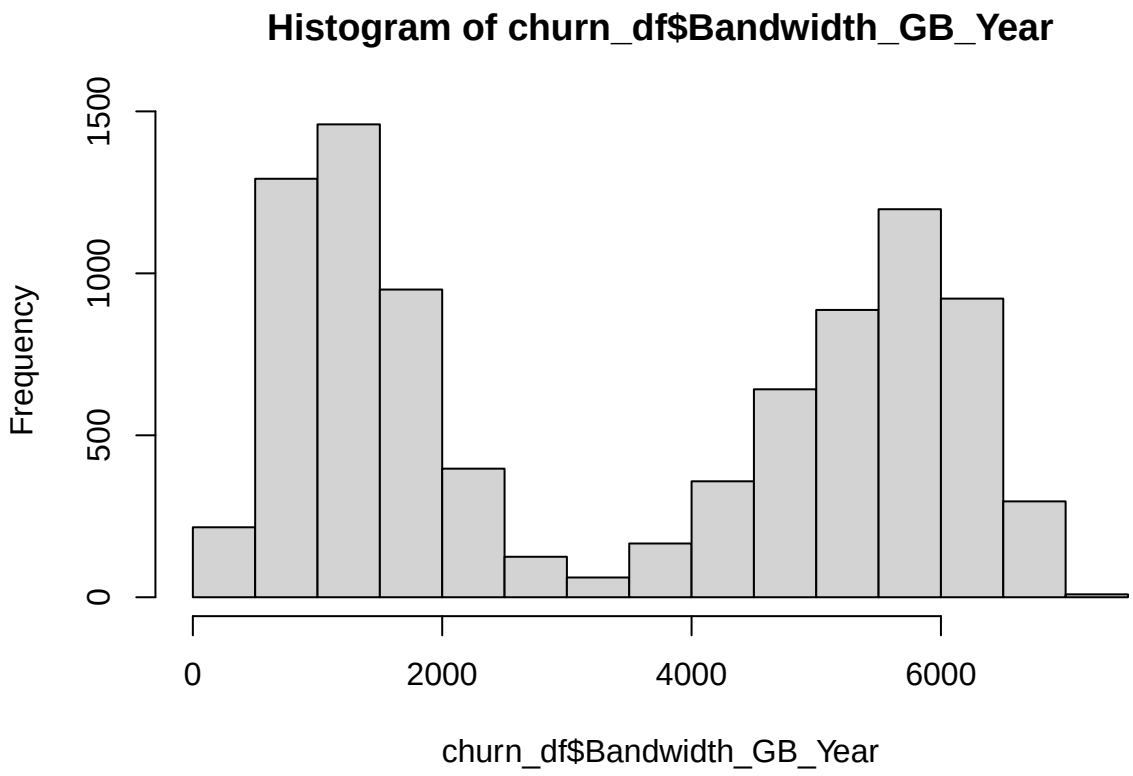

```
hist(churn_df$Income)
```



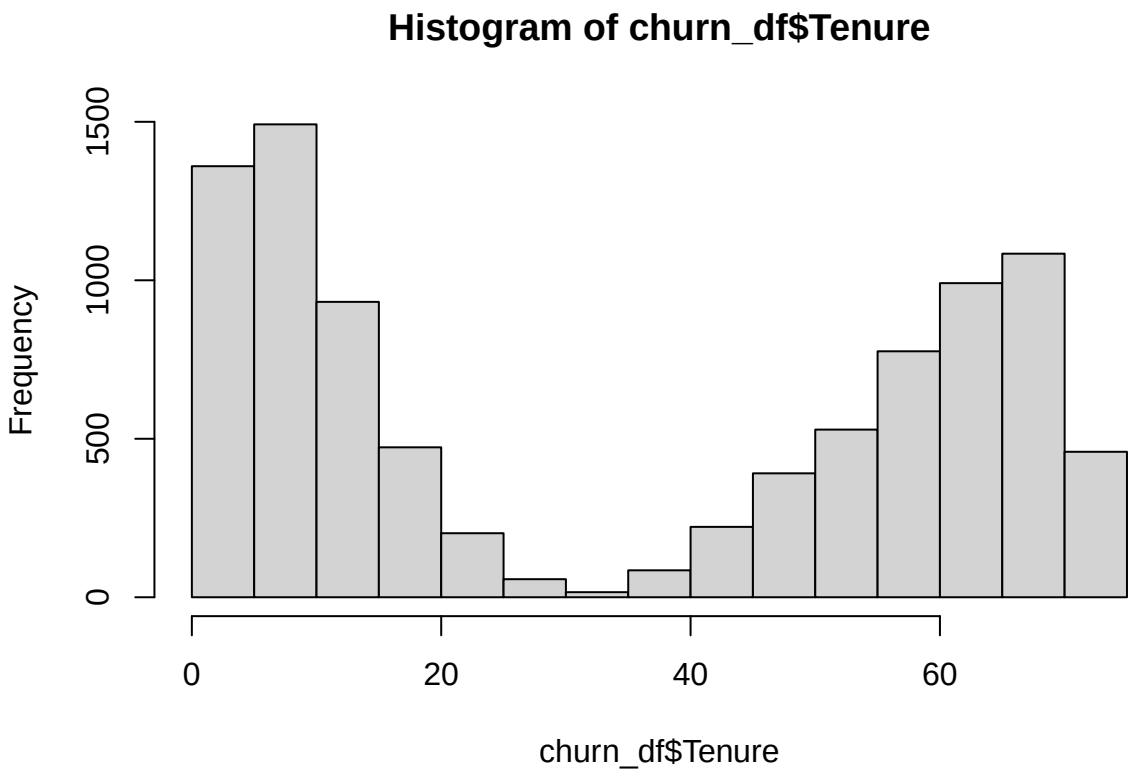

```
hist(churn_df$Age)
```




```
hist(churn_df$Bandwidth_GB_Year)
```




```
hist(churn_df$Tenure)
```



B2, Justification of approach

I took a reasonable and thorough approach to finding data inconsistencies. Regarding duplicates, one could manually scan the data for duplicate values, which is time-consuming and not feasible. The duplicated() function yields a true/false value for each record, which is also time-consuming to scan. Introducing the sum() function with the duplicated() function is the most efficient way to identify the duplicates.

I like the miss_var_summary() the best for identifying missing values. It allows you to quickly diagnose all the variables with missing values in order of missingness. It provides each variable's count and percentage of missing values to see how pervasive the missing data is.

My approach to detecting outliers is where I could have gone in different directions. I have used boxplots for quite some time in data analysis, so I felt comfortable with them. It certainly takes some time to create and study the plots for each quantitative variable, but that would have been the case if calculating z-scores or histograms.

I wanted to ensure there was not a mix of yes/no and 1/0 variables. I knew I could quickly obtain the unique values in each variable using the unique() function. While there may be a more efficient way to get the same result, this was my best approach. Given that all the yes/no variables were consistent, I also felt okay not re-expressing them.

B3, Justification of programming language

The R programming language was created specifically for statistical analysis. In my experience using the language, the syntax is more straightforward than Python, and it has made sense to me. The visualization

capabilities are strong and easy to produce. There are many packages available to users that make data cleaning much more efficient. It was an excellent choice for the initial data cleaning and exploration stages.

I used the Tidyverse, Naniar, Rstatix, Visdat, and Factoextra libraries to assist with my data cleaning. Tidyverse is a broad collection of packages designed for data science (Tidyverse, n.d.). Within Tidyverse, I utilized the ggplot2 package for visualization purposes. Also, some functions in the Factoextra library require ggplot2 to be active. The Naniar library provides a wide array of functions to assist with identifying and visualizing missing data (Naniar, 2023). I enjoy the simplicity of the missingness functions in the Naniar library. I used the outlier functionality of the Rstatix library to easily parse the outliers into separate data frames and get the information I needed. I used the Factoextra library to perform the Principal Component Analysis, and Visdat assisted with the scree plot. It allowed me to perform PCA in one step.

C1, Findings

After checking for data inconsistencies, I identified some potential issues within the data. My first detection step was to check for duplicate data. I did not find any issues regarding duplicates. Had there been duplicates, I would have isolated the non-duplicated records using the distinct() function.

I then progressed to looking for missing values. Using the miss_var_summary() function, I could identify missing values in eight variables. The table below shows the magnitude of missingness within the dataset. Four fields have roughly ¼ of the values missing, with the other four hovering around 10%.

```
miss_var_summary(churn_df)
```

```
## # A tibble: 52 x 3
##    variable          n_miss pct_miss
##    <chr>              <int>    <num>
##  1 Children            2495    25.0
##  2 Income              2490    24.9
##  3 Techie              2477    24.8
##  4 Age                 2475    24.8
##  5 Phone               1026    10.3
##  6 Bandwidth_GB_Year   1021    10.2
##  7 TechSupport          991     9.91
##  8 Tenure               931     9.31
##  9 X                      0     0
## 10 CaseOrder              0     0
## # i 42 more rows
```

I detected outliers in eight of the quantitative variables using box plots. The 'Email' and 'Outage_sec_perweek' variables had outliers outside both fences. The remaining six variables only had outliers beyond the upper fences. I used the identify_outliers() function from the Rstatix package to isolate the outliers and determine how many were present. I could call the minimum and maximum of these outliers to get the ranges.

```
# For variables with apparent outliers, obtain the counts and range of values [In-text citation: (Ident
population_outlier <- identify_outliers(data = churn_df, variable = "Population")
  sum(population_outlier$is.outlier)
```

```
## [1] 937
```

```
  min(population_outlier$Population)
```

```
## [1] 31816
```

```
  max(population_outlier$Population)
```

```
## [1] 111850
```

```
children_outlier <- identify_outliers(data = churn_df, variable = "Children")
  sum(children_outlier$is.outlier)
```

```
## [1] 302
```

```
  min(children_outlier$Children)
```

```
## [1] 8
```

```
  max(children_outlier$Children)
```

```
## [1] 10
```

```
income_outlier <- identify_outliers(data = churn_df, variable = "Income")
  sum(income_outlier$is.outlier)
```

```
## [1] 249
```

```
  min(income_outlier$Income)
```

```
## [1] 104867.5
```

```
  max(income_outlier$Income)
```

```
## [1] 258900.7
```

```
outage_outlier <- identify_outliers(data = churn_df, variable = "Outage_sec_perweek")
  sum(outage_outlier$is.outlier)
```

```
## [1] 539
```

```
  min(outage_outlier$Outage_sec_perweek)
```

```
## [1] -1.348571
```

```
  max(outage_outlier$Outage_sec_perweek)
```

```
## [1] 47.04928
```

```
email_outlier <- identify_outliers(data = churn_df, variable = "Email")
  sum(email_outlier$is.outlier)
```

```
## [1] 38
```

```
  min(email_outlier$Email)
```

```
## [1] 1
```

```
  max(email_outlier$Email)
```

```
## [1] 23
```

```
contacts_outlier <- identify_outliers(data = churn_df, variable = "Contacts")
  sum(contacts_outlier$is.outlier)
```

```
## [1] 8
```

```
  min(contacts_outlier$Contacts)
```

```
## [1] 6
```

```
  max(contacts_outlier$Contacts)
```

```
## [1] 7
```

```
equip_outlier <- identify_outliers(data = churn_df, variable = "Yearly_equip_failure")
  sum(equip_outlier$is.outlier)
```

```
## [1] 94
```

```
  min(equip_outlier$Yearly_equip_failure)
```

```
## [1] 3
```

```
  max(equip_outlier$Yearly_equip_failure)
```

```
## [1] 6
```

```
monthly_charges_outlier <- identify_outliers(data = churn_df, variable = "MonthlyCharge")
  sum(monthly_charges_outlier$is.outlier)
```

```
## [1] 5
```

```
  min(monthly_charges_outlier$MonthlyCharge)
```

```
## [1] 298.173
```

```
  max(monthly_charges_outlier$MonthlyCharge)
```

```
## [1] 315.8786
```

C2, Justification of treatment

As stated, I did not identify duplicates, so no action was necessary. After detecting which variables had missing values, I created histograms for each numeric variable to view the distribution shape using the hist() function. For all the variables, I chose to use Univariate Imputation for the simplicity of the approach. Deletion was not a desirable choice due to the pervasiveness of missing data.

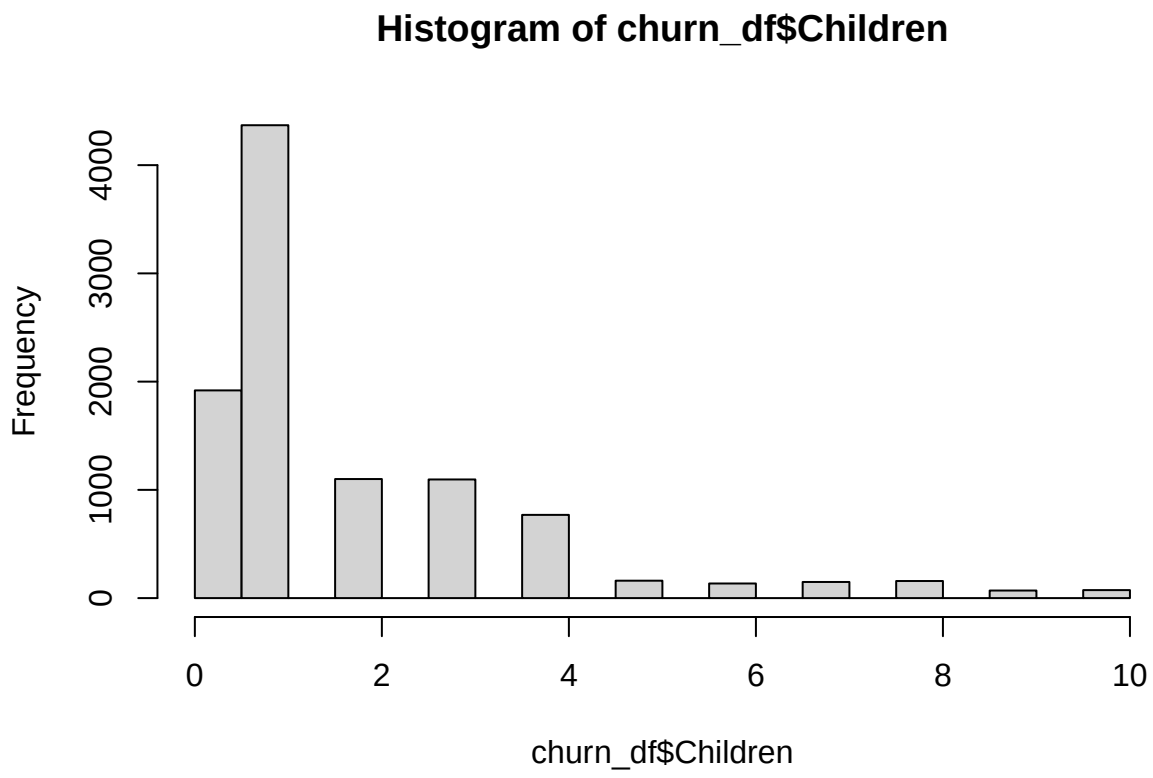
The 'Children' and 'Income' variables were both right-skewed. I used the median to impute the missing values for both. Using the mean would have given extreme values due to the skewness. The 'Age' variable had a uniform distribution, which led to me using the mean for imputation. The 'Bandwidth_GB_Year' and 'Tenure' variables had bi-modal distributions. I used the median for imputation. I felt that was a safer option than using the mode. Three categorical variables with missing values were Techie, Phone, and TechSupport. I used the mode for imputation in these cases (Missing Values, n.d.). Once imputed all the missing values, I again used the miss_var_summary() function to ensure no missing values remained.

```
# Treat the missing values for the numeric variables using Univariate Imputation [In-text citaiton: (Mi
churn_df$Children[is.na(churn_df$Children)] <- median(churn_df$Children, na.rm = TRUE)
churn_df$Income[is.na(churn_df$Income)] <- median(churn_df$Income, na.rm = TRUE)
churn_df$Age[is.na(churn_df$Age)] <- round(mean(churn_df$Age, na.rm = TRUE))
churn_df$Bandwidth_GB_Year[is.na(churn_df$Bandwidth_GB_Year)] <- median(churn_df$Bandwidth_GB_Year, na.
churn_df$Tenure[is.na(churn_df$Tenure)] <- median(churn_df$Tenure, na.rm = TRUE)

# Treat the missing values for the categorical variables using Univariate Imputation [In-text citation:
churn_df$Techie[is.na(churn_df$Techie)] <- (names(which.max(table(churn_df$Techie))))
churn_df$Phone[is.na(churn_df$Phone)] <- (names(which.max(table(churn_df$Phone))))
churn_df$TechSupport[is.na(churn_df$TechSupport)] <- (names(which.max(table(churn_df$TechSupport))))

# Verify that all missing values have been treated
miss_var_summary(churn_df)
```

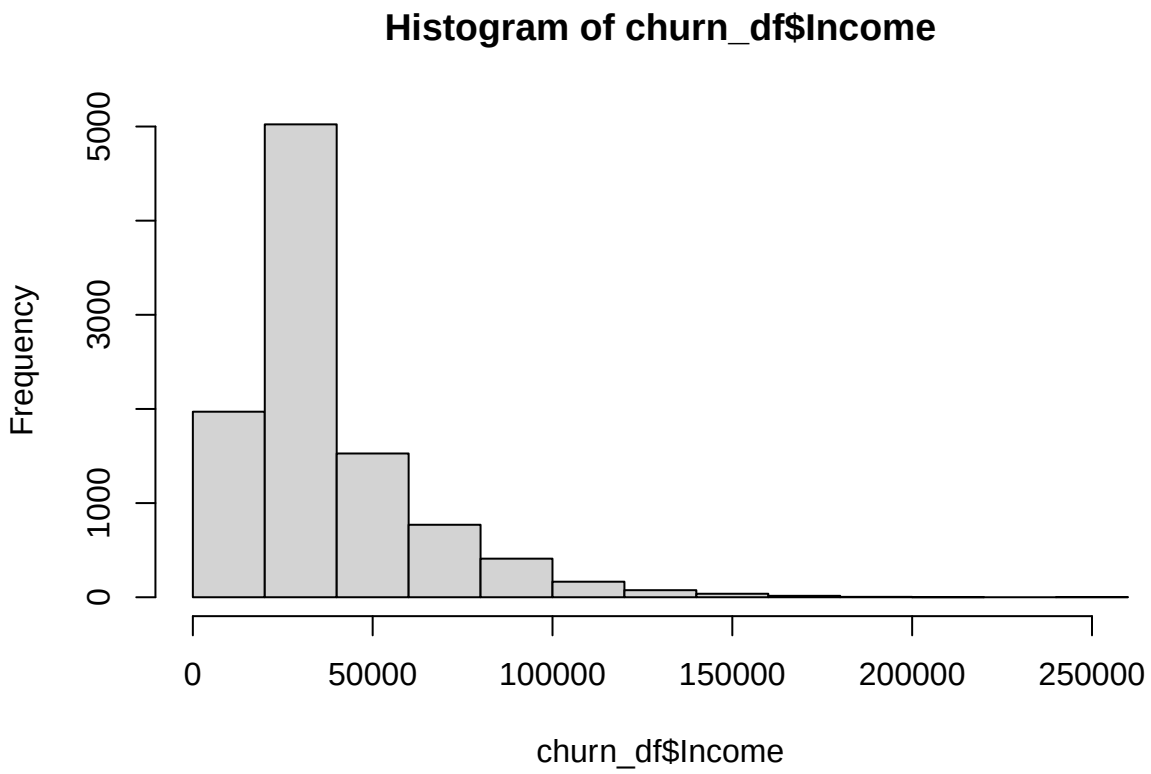
```
## # A tibble: 52 x 3
##    variable    n_miss pct_miss
##    <chr>        <int>    <num>
##  1 X                0        0
##  2 CaseOrder        0        0
##  3 Customer_id      0        0
##  4 Interaction      0        0
```

```
##  5 City             0        0
##  6 State            0        0
##  7 County           0        0
##  8 Zip              0        0
##  9 Lat              0        0
## 10 Lng              0        0
## # i 42 more rows
```

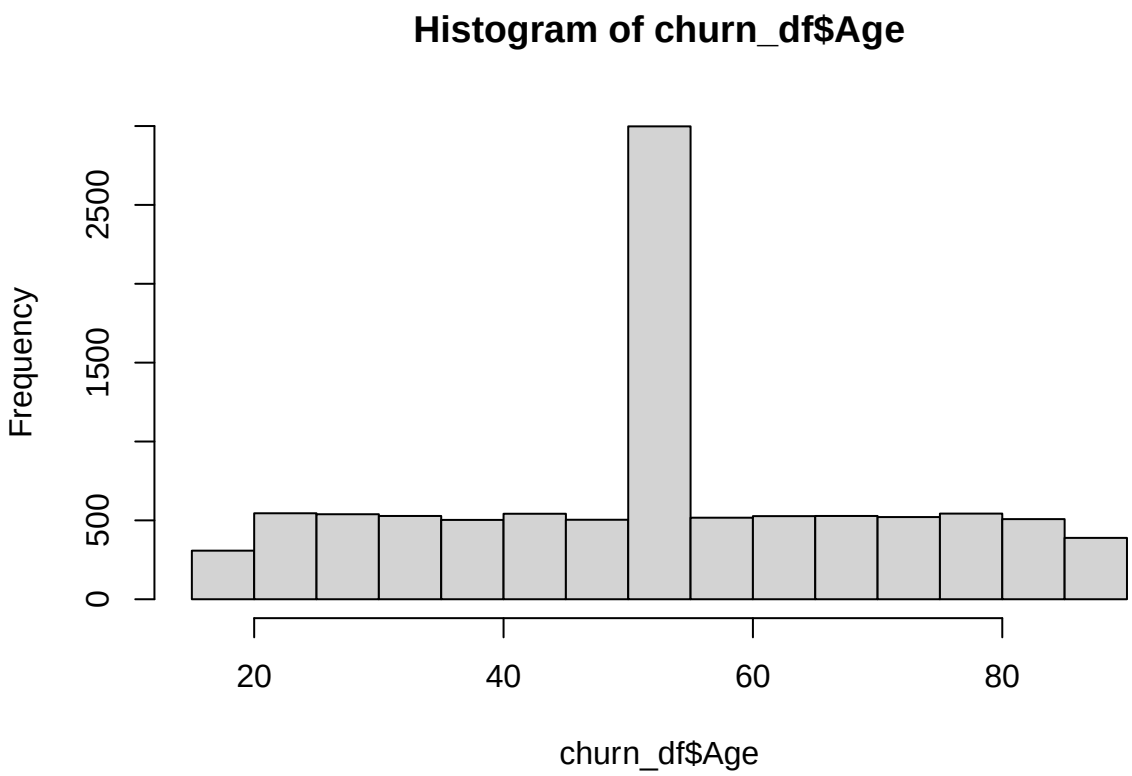
```r
# View updated histograms for the numeric variables after imputation
hist(churn_df$Children)
```



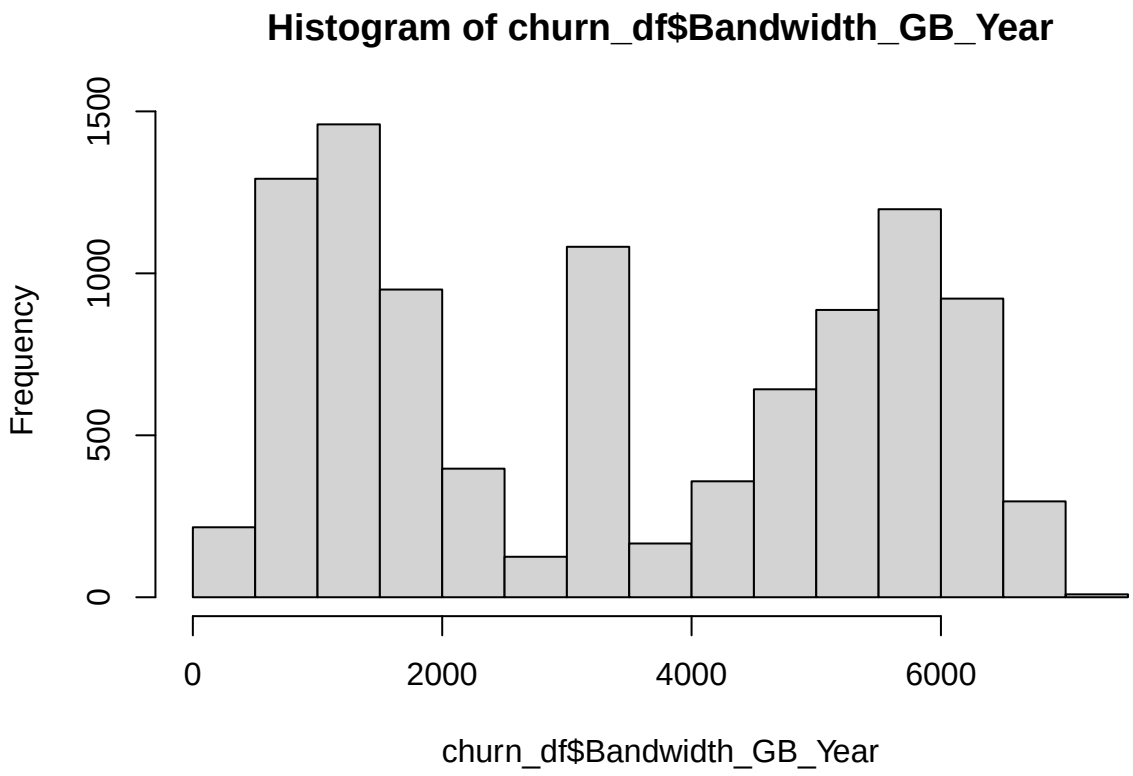

```r
hist(churn_df$Income)
```



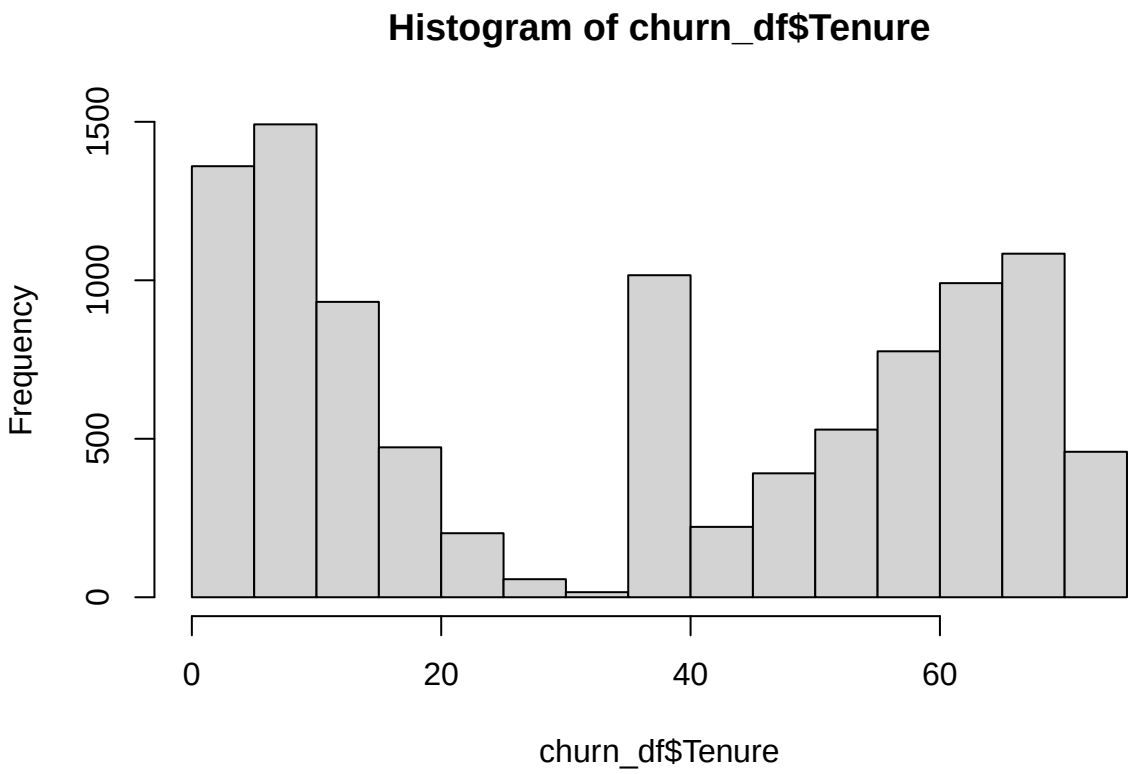

```
hist(churn_df$Age)
```




```
hist(churn_df$Bandwidth_GB_Year)
```




```
hist(churn_df$Tenure)
```



I chose varying treatment methods for the outliers I identified. There were certain variables that I felt the best approach was to retain the outliers. The 'MonthlyCharge' variable had five outliers ranging from $298 to $315. While that does seem high, I determined they were acceptable values given the nature of the services offered. The 'Yearly_equip_failure' variable had 94 outliers ranging from 3 to 6. This variable represents how many times equipment had to be replaced in the last year. Again, these numbers felt acceptable, considering we are dealing with technology. The 'Email' variable had 38 outliers with two different ranges. Below the lower fence, the range was 1 to 3; beyond the upper fence, it was 21 to 23. These seemed like acceptable values for emails going out to customers. The 'Income' variable had 759 outliers ranging from $78K to $259K. These all seem like acceptable values for income. In this case, I would be more likely to question some values deemed in-bounds. These were customer-reported values, potentially with errors or misunderstandings, such as weekly or monthly income instead of annual. The 'Children' variable had 451 outliers ranging from 7 to 10. Again, I felt these were acceptable values.

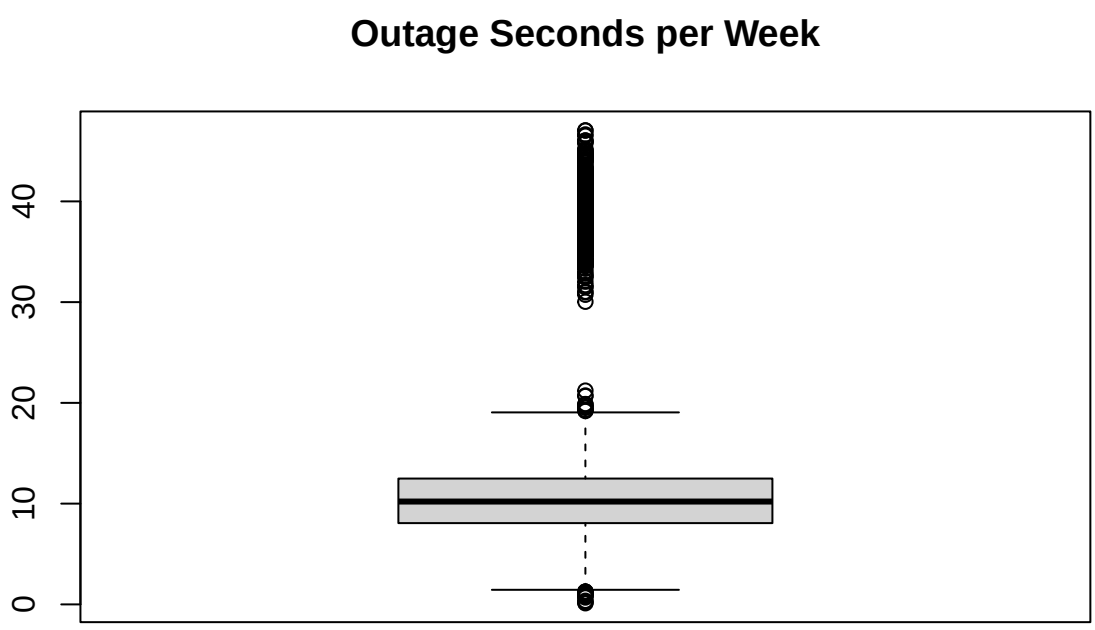
There were two variables for which I felt it was not okay to retain the outliers. In the 'Outage_sec_perweek' variable, there were outliers under the lower fence with negative values. Given that this variable represents the average outage time per week, it is impossible to have a negative time value. Therefore, these values needed to be treated. I used the median to impute the outage values. After treating the negative values, the number of outliers remaining was reduced to 528 observations with a lower range of 0.11 to 1.27 and an upper range of 19.19 to 47.05.

The 'Population' variable had 937 outliers ranging from 31,816 to 111,850. These values I felt were acceptable. However, upon investigating the data, I realized there was a subset with zero population. Zero population is unacceptable, considering the customer lives there at the very least. There were also zero values for large cities such as New York and Philadelphia. I used the median to impute population values in this instance. After imputation, the number of outliers increased by five observations. Imputing with the median tightened the fences, expanding the outlier range from 31,749 to 111,850.

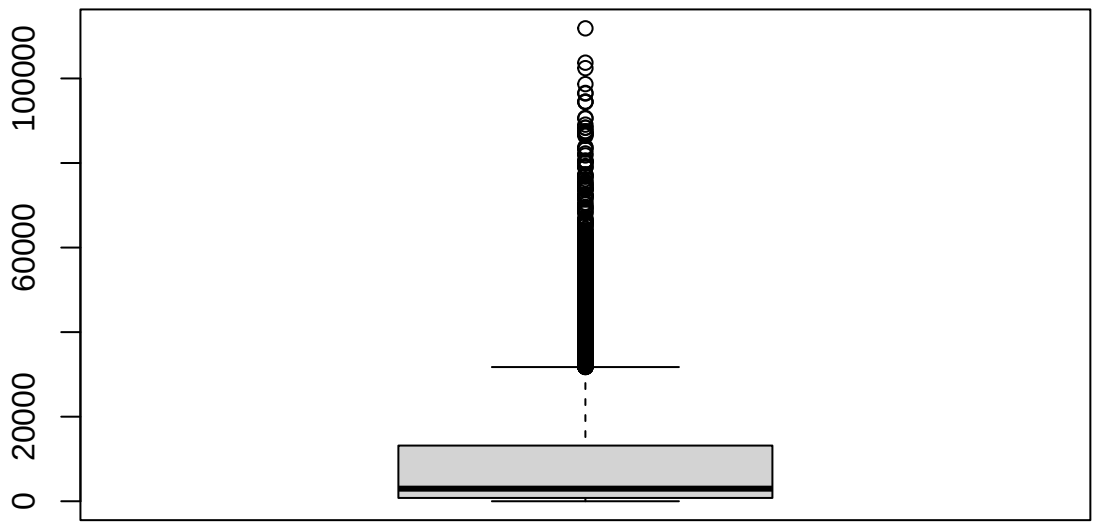
```r
# Treat non-acceptable outliers - replace outliers with NA and verify there are now missing values
churn_df$Outage_sec_perweek[churn_df$Outage_sec_perweek < 0] <- NA
churn_df$Population[churn_df$Population == 0] <- NA
miss_var_summary(churn_df)
```

```
## # A tibble: 52 x 3
##    variable           n_miss pct_miss
##    <chr>               <int>    <num>
##  1 Population             97     0.97
##  2 Outage_sec_perweek     11     0.11
##  3 X                       0     0
##  4 CaseOrder               0     0
##  5 Customer_id             0     0
##  6 Interaction             0     0
##  7 City                    0     0
##  8 State                   0     0
##  9 County                  0     0
## 10 Zip                     0     0
## # i 42 more rows
```

```r
# Impute missing values with Median
churn_df$Outage_sec_perweek[is.na(churn_df$Outage_sec_perweek)] <- median(churn_df$Outage_sec_perweek, 
churn_df$Population[is.na(churn_df$Population)] <- median(churn_df$Population, na.rm = TRUE)

# View Updated Box plots for each of the Imputed Outliers
boxplot(churn_df$Outage_sec_perweek, main = "Outage Seconds per Week")
```




```
boxplot(churn_df$Population, main = "Population")
```




```
# View Updated Counts and Ranges for Imputed Outliers
population_outlier_updated <- identify_outliers(data = churn_df, variable = "Population")
sum(population_outlier_updated$is.outlier)
```

```
## [1] 942
```

```
min(population_outlier_updated$Population)
```

```
## [1] 31749
```

```
max(population_outlier_updated$Population)
```

```
## [1] 111850
```

```
outage_outlier_updated <- identify_outliers(data = churn_df, variable = "Outage_sec_perweek")
sum(outage_outlier_updated$is.outlier)
```

```
## [1] 528
```

```
min(outage_outlier_updated$Outage_sec_perweek)
```

```
## [1] 0.1138212
```

```r
max(outage_outlier_updated$Outage_sec_perweek)
```

```
## [1] 47.04928
```

```r
min(churn_df_orig$Population)
```

```
## [1] 0
```

```r
min(churn_df$Population)
```

```
## [1] 2
```

C3, Summarize Outcome

My data plan consisted of recurring detection and treatment steps for each potential data quality issue. I took steps to identify duplicate records and did not find any. I detected missing values in eight variables and imputed the missing values with either the mean, median, or mode. I detected outliers in eight variables, six of which I determined were acceptable. For the remaining two variables, I imputed the outliers with the median.

After all my cleaning steps, the end-product is a data set free of duplicates and missing values. Since I chose to retain many outliers, outliers will still be present in the data. However, I can confirm that the values will make more sense after imputing the negative values in the 'Outage_sec_perweek' variable and imputing values equal to zero in the 'Population' column.

C4, Disadvantages of Methods

Every choice has a consequence. The same goes for the detection and treatment methods I used in the data set. When missing values are present, there can be adverse consequences when treated with Univariate Imputation. You are guessing what the missing values are, albeit in a mathematical way. This can artificially change the distribution of your data set, especially when there are many missing values. It is certainly possible the missing values would have been at either end of the original distribution.

There are many ways outliers can be detected in a data set. I chose to use the box plot method. While this is a solid approach, it may not be as precise as using z-scores to determine the outliers. The fact that I retained many outliers can also have potential limitations. Based on what I knew of the variables, they were acceptable values. There is a risk, however, that these were errant and thus true outliers. When I imputed the negative values for the 'Outage_sec_perweek' variable, there is again the issue of guessing the values. By imputing zero values for the population, I already know the new values are inaccurate. In a real-world scenario, I would have preferred the actual population values, as I imagine there would be a lookup table. However, imputing zero values with the median is still more accurate than showing zero values.

C5, Limitations

My initial research question was: what customer factors influence length of tenure? After all the data detection and treatment techniques used, we are left with a data set in a better state than we originally started with. However, as previously discussed, there are potential drawbacks to some of the steps taken. An analyst would need to be aware that many missing values were imputed. It could give a misleading conclusion when determining influential factors for length of tenure. The same would go for retaining outliers in multiple

variables. While the values seemed acceptable, given my knowledge of them, it is possible they should have been imputed or removed. Imputing the zero values in the 'Population' variable made it more accurate, but it still needs to be corrected. It would have been better to substitute the actual population values. These could all lead to inaccurate conclusions.

D1, CSV

See the attached CSV file with the cleaned data. The file name is "d206_babcock_churn_df_clean.csv".

D2, R Code

See the R Code attached. The file name is "d206_babcock_code.R".

E1, Performing PCA

I performed PCA on the cleaned data set. The variables I used for the analysis were as follows, which resulted in five principal components:

- Income
- Outage_sec_perweek
- Tenure
- MonthlyCharge
- Bandwidth_GB_year

Below is a screenshot of the PCA loadings matrix.

```
# Obtain numeric variables to use in PCA
churn_pca_data <- churn_df %>%
  select(
    Income,
    Outage_sec_perweek,
    Tenure,
    MonthlyCharge,
    Bandwidth_GB_Year
  )

# Perform PCA, ensuring to scale the data [In-text citation: (PCA, n.d.)]
churn_pca_results <- prcomp(churn_pca_data, center = TRUE, scale. = TRUE)

# View the PCA loadings [In-text citation: (PCA, n.d.)]
churn_pca_results$rotation
```

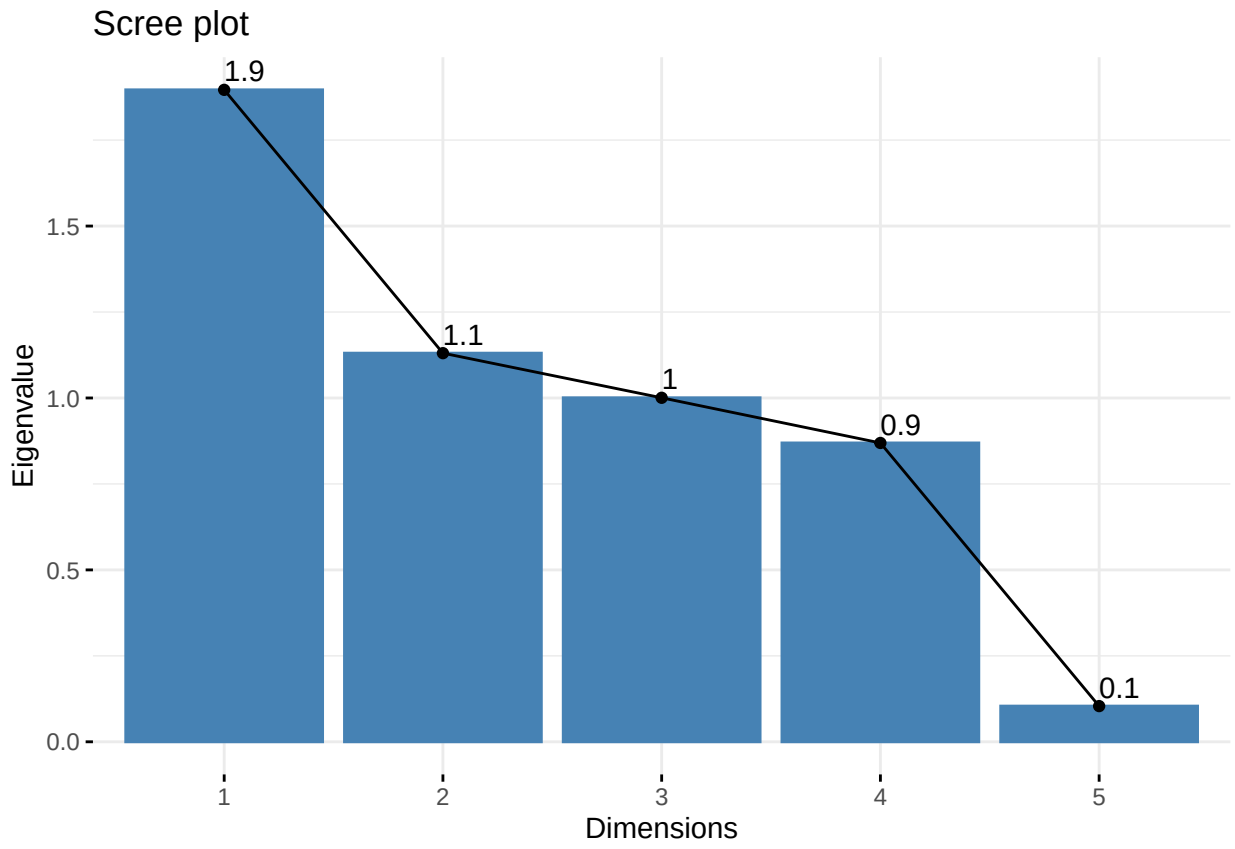
```
##                          PC1          PC2          PC3         PC4
## Income             0.00597915 -0.000375298  0.997848989  0.06527475
## Outage_sec_perweek 0.02171264 -0.705727423  0.045831330 -0.70666601
## Tenure             0.70537520  0.057602080 -0.001256366 -0.03595411
## MonthlyCharge      0.04537851 -0.706073597 -0.046517386  0.70351179
## Bandwidth_GB_Year  0.70702151  0.009525794 -0.005607056  0.01186685
##                           PC5
```

```
## Income              0.00084380736
## Outage_sec_perweek  0.00002006882
## Tenure             -0.70557320263
## MonthlyCharge      -0.04804339059
## Bandwidth_GB_Year   0.70700606500
```

E2, Justify PCA Selection

I selected these five variables because they were the only continuous numeric variables in the dataset. I also considered including 'Age' and 'Population' but ultimately decided to exclude them because they are discrete. Below is a screenshot of the resulting scree plot. Utilizing the Kaiser criterion, I would retain principal components 1, 2, and 3 as they had values greater than or equal to 1(PCA, n.d.).

```
# Create scree plot with eigenvalues [In-text citation: (PCA, n.d.)]
fviz_eig(churn_pca_results, choice = "eigenvalue", addlabels = TRUE)
```



E3, PCA Benefits

PCA can offer many benefits to organizations. Generally, it helps with dimension reduction, making machine-learning models more efficient and accurate. PCA will also help reduce noise in the data and make it easier to visualize data (Bigabid, n.d.). In our example, the Churn data set is extensive. It has 52 variables. Although I only used five continuous numeric variables for the PCA, I was able to reduce the variables of interest to 3 principal components.

F, Panopto Recording

I created a Panopto video recording that covered the program I used and the execution of the code used. The video link can was included in the submission.

G, Sources for Code

Bobbitt, Z. (July 30, 2021). How to turn off scientific notation in R (with examples). Statology. Retrieved September 6, 2024, from (https://www.statology.org/turn-off-scientific-notation-in-r/)

Identify univariate outliers using boxplot methods (n.d.). Rdocumentation. Retrieved September 2, 2024, from (https://www.rdocumentation.org/packages/rstatix/versions/0.7.2/topics/identify_outliers)

Tierney, N. (n.d.). Dealing with Missing Data in R [MOOC]. DataCamp. (https://app.datacamp.com/learn/courses/dealing-with-missing-data-in-r)

WGU College of Information Technology (n.d.). Getting Started with Detecting and Treating Missing Values [PowerPoint Slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Docu

WGU College of Information Technology (n.d.). Getting Started with Duplicates [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20

WGU College of Information Technology (n.d.). Getting Started with Principal Component Analysis (PCA) [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Docu

WGU College of Information Technology (n.d.). Introduction to Data Types, Distributions and Imputation [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Docu

H, Sources for Content

R packages for data science (n.d.). Tidyverse. Retrieved September 2, 2024, from (https://www.tidyverse.org)

Tierney, N., Cook, D. (2023). Naniar. Retrieved September 2, 2024, from (https://naniar.njtierney.com)

Tierney, N. (n.d.). Dealing with Missing Data in R [MOOC]. DataCamp. (https://app.datacamp.com/learn/courses/dealing-with-missing-data-in-r)

WGU College of Information Technology (n.d.). Getting Started with Detecting and Treating Missing Values [PowerPoint Slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Docu

WGU College of Information Technology (n.d.). Getting Started with Detecting and Treating Outliers [PowerPoint Slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Document

WGU College of Information Technology (n.d.). Getting Started with Duplicates [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20

WGU College of Information Technology (n.d.). Getting Started with Principal Component Analysis (PCA) [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Docu

WGU College of Information Technology (n.d.). Introduction to Data Types, Distributions and Imputation [PowerPoint slides]. Western Governors University. (https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Docu

What is Principal Component Analysis? (n.d.). Bigabid. Retrieved September 4, 2024, from (https://www.bigabid.com/what-is-pca-and-how-can-i-use-it/)