FML-Assignment 1

Chandima Attanayake

2024-09-08

01. Conduct Basic Descriptive Statistics

```
# Load necessary libraries

library(dplyr)     # For data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(psych)     # For psychological and descriptive statistics
library(readr)     # For efficient data import and export
library(tidyverse) # A collection of packages including dplyr, ggplot2, and more
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::%+%()   masks psych::%+%()
## x ggplot2::alpha() masks psych::alpha()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# Load the data
data <- read.csv("//Users/chandimaattanayake/Downloads/Churn.csv")

# View basic descriptive statistics


head(data)    # View the head of the data (first few rows)
```

```
##   CallFailures SubscriptionLength DataUsage VoiceMinutes CustomerSupportCalls
## 1           16                 11 4.1936070     4836.250                    2
## 2            4                  9 8.4093629     1694.926                    5
## 3            0                  8 0.6541119     4384.157                    3
## 4            9                  9 8.8331384     2609.912                    0
## 5            3                  8 7.2457588     2889.617                    3
## 6           17                  3 0.8084840     1206.698                    1
##   ContractType MonthlyCharges RoamingUsage Churn
## 1      Monthly       24.26866     2.600714     0
## 2      Monthly       82.48409     5.277427     1
## 3      Monthly       52.88977     3.170094     0
## 4      Monthly       32.25711     3.033796     0
## 5      Monthly       58.24018     8.905393     0
## 6      Monthly       31.97728     8.853842     1
```

```
tail(data)    # View the tail of the data (last few rows)
```

```
##      CallFailures SubscriptionLength DataUsage VoiceMinutes
## 995             9                 18 6.6938689    3231.4057
## 996            14                 13 0.3682914     220.6486
## 997            18                 18 3.3392797    1331.7691
## 998            11                  5 7.0439535    4889.2448
## 999            17                 14 3.9515069    2521.9432
## 1000           17                 10 0.2560617    2507.9017
##      CustomerSupportCalls ContractType MonthlyCharges RoamingUsage Churn
## 995                     4       Annual       21.92054     7.846210     1
## 996                     4       Annual       33.22040     8.031035     1
## 997                     4      Monthly       25.08640     9.778682     1
## 998                     4      Monthly       26.92161     8.391785     0
## 999                     5      Monthly       73.53721     7.639660     0
## 1000                    4      Monthly       60.24888     2.344890     0
```

```
dim(data)     # Check the dimensions of the dataset (number of rows and columns)
```

```
## [1] 1000    9
```

```
summary(data) # Get a summary of the dataset (min, max, median, etc.)
```

```
##   CallFailures    SubscriptionLength   DataUsage        VoiceMinutes
##  Min.   : 0.000   Min.   : 1.00      Min.   :0.04479   Min.   :   1.913
##  1st Qu.: 5.000   1st Qu.: 6.00      1st Qu.:2.45883   1st Qu.:1392.148
##  Median :10.000   Median :12.00      Median :5.07325   Median :2626.685
##  Mean   : 9.985   Mean   :12.09      Mean   :5.09635   Mean   :2564.964
##  3rd Qu.:16.000   3rd Qu.:18.00      3rd Qu.:7.82260   3rd Qu.:3712.721
##  Max.   :20.000   Max.   :24.00      Max.   :9.99831   Max.   :4998.703
##  CustomerSupportCalls ContractType       MonthlyCharges   RoamingUsage
##  Min.   :0.000        Length:1000        Min.   :20.07   Min.   :0.01345
##  1st Qu.:1.000        Class :character   1st Qu.:37.57   1st Qu.:2.32212
##  Median :2.000        Mode  :character   Median :56.91   Median :4.94221
##  Mean   :2.394                           Mean   :58.42   Mean   :4.95070
##  3rd Qu.:4.000                           3rd Qu.:77.45   3rd Qu.:7.44860
##  Max.   :5.000                           Max.   :99.96   Max.   :9.99680
```

```
##      Churn
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :1.000
##  Mean   :0.504
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
# Check the structure of the dataset (data types, number of factors, etc.)
str(data)
```

```
## 'data.frame':    1000 obs. of  9 variables:
##  $ CallFailures        : int  16 4 0 9 3 17 16 14 6 3 ...
##  $ SubscriptionLength  : int  11 9 8 9 8 3 8 10 10 2 ...
##  $ DataUsage           : num  4.194 8.409 0.654 8.833 7.246 ...
##  $ VoiceMinutes        : num  4836 1695 4384 2610 2890 ...
##  $ CustomerSupportCalls: int  2 5 3 0 3 1 1 4 3 1 ...
##  $ ContractType        : chr  "Monthly" "Monthly" "Monthly" "Monthly" ...
##  $ MonthlyCharges      : num  24.3 82.5 52.9 32.3 58.2 ...
##  $ RoamingUsage        : num  2.6 5.28 3.17 3.03 8.91 ...
##  $ Churn               : int  0 1 0 0 0 1 1 1 0 0 ...
```

```
# Calculate the mean for numerical variables (excluding NA values)
means <- sapply(data, function(x) if(is.numeric(x)) mean(x, na.rm=TRUE))

# Calculate the median for numerical variables
medians <- sapply(data, function(x) if(is.numeric(x)) median(x, na.rm=TRUE))

# Define a function to calculate the mode
get_mode <- function(x) {
  uniq_vals <- unique(x)
  uniq_vals[which.max(tabulate(match(x, uniq_vals)))]
}


modes <- sapply(data, get_mode) # Calculate the mode for each variable

missing_values <- sapply(data, function(x) sum(is.na(x))) # Check for any missing values in the dataset


# Print the results
print("Means:")
```

```
## [1] "Means:"
```

```
print(means)
```

```
## $CallFailures
## [1] 9.985
##
## $SubscriptionLength
## [1] 12.093
```

```
##
## $DataUsage
## [1] 5.096345
##
## $VoiceMinutes
## [1] 2564.964
##
## $CustomerSupportCalls
## [1] 2.394
##
## $ContractType
## NULL
##
## $MonthlyCharges
## [1] 58.41586
##
## $RoamingUsage
## [1] 4.950701
##
## $Churn
## [1] 0.504
```

```
print("Medians:")
```

```
## [1] "Medians:"
```

```
print(medians)
```

```
## $CallFailures
## [1] 10
##
## $SubscriptionLength
## [1] 12
##
## $DataUsage
## [1] 5.073246
##
## $VoiceMinutes
## [1] 2626.685
##
## $CustomerSupportCalls
## [1] 2
##
## $ContractType
## NULL
##
## $MonthlyCharges
## [1] 56.91449
##
## $RoamingUsage
## [1] 4.942205
##
## $Churn
## [1] 1
```

```r
print("Modes:")
```

```
## [1] "Modes:"
```

```r
print(modes)
```

```
##         CallFailures   SubscriptionLength            DataUsage
##                  "9"                 "12"   "4.19360703555867"
##         VoiceMinutes CustomerSupportCalls         ContractType
##   "4836.25042135827"                  "0"            "Monthly"
##       MonthlyCharges         RoamingUsage                Churn
##   "24.2686576396227"   "2.60071393335238"                  "1"
```

```r
print("Missing Values:")
```

```
## [1] "Missing Values:"
```

```r
print(missing_values)
```

```
##         CallFailures   SubscriptionLength            DataUsage
##                    0                    0                    0
##         VoiceMinutes CustomerSupportCalls         ContractType
##                    0                    0                    0
##       MonthlyCharges         RoamingUsage                Churn
##                    0                    0                    0
```

```r
knitr::opts_chunk$set(echo = TRUE)
```

02. Interpretation of Descriptive Statistics (in R Markdown)

Overview

Upon the descritptive analysis, following observations were noted.

- The dataset includes 1000 records with nine variables

call failure On average, customers experience about 10 call failures. The most common number of call failures reported is 9 with the mean and median both indicating a high frequency of issues.Therefore, the process related to this service should be revisted.

Subscription Length The average subscription length is approximately 12 months, which aligns with the median and mode. This indicates that most customers have a subscription period of around one year.

Data Usage Customers use an average of 5.10 GB of data per month. The mode of 4.19 GB suggests that a significant portion of customers use around this amount, even though the the average is slightly higher.

Voice Minutes The average voice usage is about 2565 minutes, however the highest one is 4836 minutes and this indicates a wide range in voice usage among customers.

Customer Support Calls On average, customers make approximately 2.39 support calls. However, majority of customers do not contact support at all, as evidenced by the mode being 0.

Monthly Charges The average monthly charge is about $58.42, with a median of $56.91. The mode of $24.27 suggests that this charge is common among customers, indicating possible tiered pricing.

Roaming Usage The average roaming usage is around 5 GB per month, with the most common usage being 2.60 GB. This suggests that while most customers use less roaming data, a few use significantly more.

Churn The churn rate is 50%, indicating that roughly half of the customers have churned. The mode of 1 highlights that churn is a common outcome for customers.

Missing Values There are no missing values and data set is ready for further analysis

03. Select and Compare Two or More Variables

```
# Load necessary libraries
library(dplyr)
library(ggplot2)

# Select two variables: Data Usage and Churn
selected_data <- dplyr::select(data, DataUsage, Churn)

# Create a summary table for these variables
summary(selected_data)
```

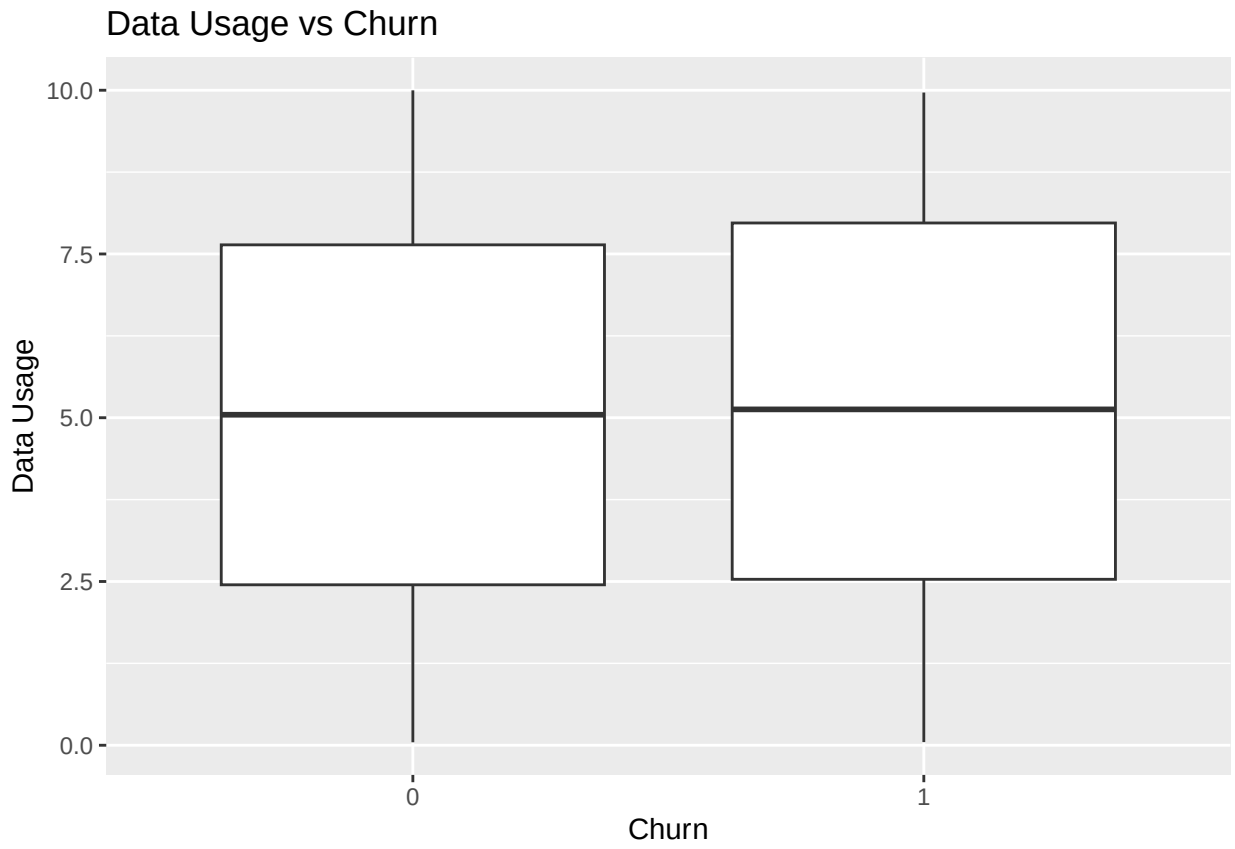
```
##    DataUsage           Churn
##  Min.   :0.04479   Min.   :0.000
##  1st Qu.:2.45883   1st Qu.:0.000
##  Median :5.07325   Median :1.000
##  Mean   :5.09635   Mean   :0.504
##  3rd Qu.:7.82260   3rd Qu.:1.000
##  Max.   :9.99831   Max.   :1.000
```

```
# Calculate the mean Data Usage for customers who churned vs those who did not
churned_vs_not <- selected_data %>% group_by(Churn) %>% summarize(mean_usage = mean(DataUsage, na.rm=TRU

# Display the summary table
print(churned_vs_not)
```

```
## # A tibble: 2 x 2
##   Churn mean_usage
##   <int>      <dbl>
## 1     0       5.04
## 2     1       5.15
```

```
# Create a simple visualization to compare
library(ggplot2)
ggplot(selected_data, aes(x = as.factor(Churn), y = DataUsage)) +
  geom_boxplot() +
  labs(x = "Churn", y = "Data Usage") +
  ggtitle("Data Usage vs Churn")
```




```
# Check the structure of the dataset to ensure CHURN exists
str(data)
```

```
## 'data.frame':    1000 obs. of  9 variables:
##  $ CallFailures        : int  16 4 0 9 3 17 16 14 6 3 ...
##  $ SubscriptionLength  : int  11 9 8 9 8 3 8 10 10 2 ...
##  $ DataUsage           : num  4.194 8.409 0.654 8.833 7.246 ...
##  $ VoiceMinutes        : num  4836 1695 4384 2610 2890 ...
##  $ CustomerSupportCalls: int  2 5 3 0 3 1 1 4 3 1 ...
##  $ ContractType        : chr  "Monthly" "Monthly" "Monthly" "Monthly" ...
##  $ MonthlyCharges      : num  24.3 82.5 52.9 32.3 58.2 ...
##  $ RoamingUsage        : num  2.6 5.28 3.17 3.03 8.91 ...
##  $ Churn               : int  0 1 0 0 0 1 1 1 0 0 ...
```

```
# Convert CHURN to a factor if it's not already one
data$Churn <- as.factor(data$Churn)

# Check if the conversion worked
str(data)
```

```
## 'data.frame':    1000 obs. of  9 variables:
##  $ CallFailures        : int  16 4 0 9 3 17 16 14 6 3 ...
##  $ SubscriptionLength  : int  11 9 8 9 8 3 8 10 10 2 ...
##  $ DataUsage           : num  4.194 8.409 0.654 8.833 7.246 ...
##  $ VoiceMinutes        : num  4836 1695 4384 2610 2890 ...
```

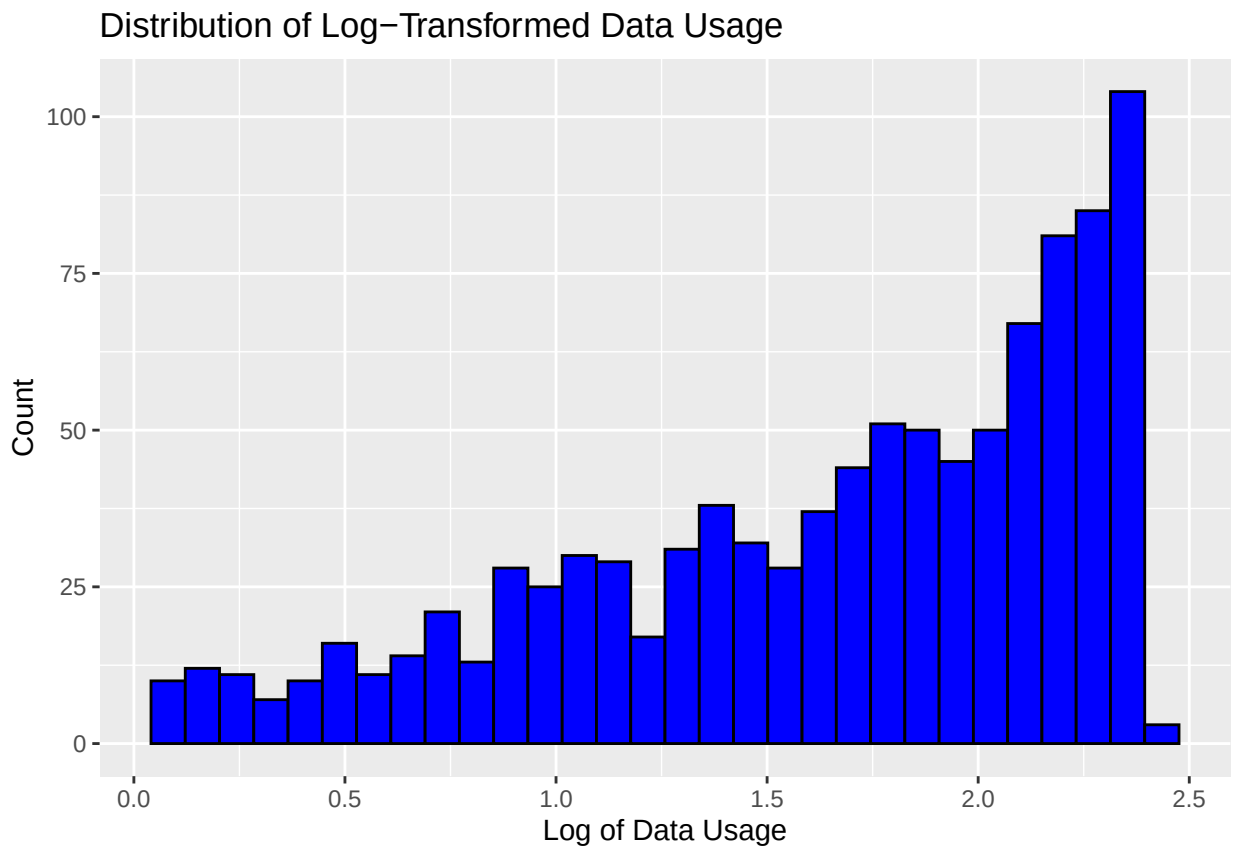
```
##  $ CustomerSupportCalls: int  2 5 3 0 3 1 1 4 3 1 ...
##  $ ContractType        : chr  "Monthly" "Monthly" "Monthly" "Monthly" ...
##  $ MonthlyCharges      : num  24.3 82.5 52.9 32.3 58.2 ...
##  $ RoamingUsage        : num  2.6 5.28 3.17 3.03 8.91 ...
##  $ Churn               : Factor w/ 2 levels "0","1": 1 2 1 1 1 2 2 2 1 1 ...
```

4. Transform a Variable and Generate a Plot Using ggplot

```r
# Load necessary libraries
library(ggplot2)

# Transform the Data Usage variable using log transformation
data$log_dataUsage <- log(data$DataUsage + 1)  # Adding 1 to avoid log(0)

# Plot the transformed variable using ggplot
ggplot(data, aes(x = log_dataUsage)) +  # Changed to plot the transformed variable
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  labs(x = "Log of Data Usage", y = "Count") +
  ggtitle("Distribution of Log-Transformed Data Usage")
```




```r
# Check the structure of the dataset to ensure the variable is created
str(data)
```

```
## 'data.frame':    1000 obs. of  10 variables:
```

```
##  $ CallFailures        : int  16 4 0 9 3 17 16 14 6 3 ...
##  $ SubscriptionLength  : int  11 9 8 9 8 3 8 10 10 2 ...
##  $ DataUsage           : num  4.194 8.409 0.654 8.833 7.246 ...
##  $ VoiceMinutes        : num  4836 1695 4384 2610 2890 ...
##  $ CustomerSupportCalls: int  2 5 3 0 3 1 1 4 3 1 ...
##  $ ContractType        : chr  "Monthly" "Monthly" "Monthly" "Monthly" ...
##  $ MonthlyCharges      : num  24.3 82.5 52.9 32.3 58.2 ...
##  $ RoamingUsage        : num  2.6 5.28 3.17 3.03 8.91 ...
##  $ Churn               : Factor w/ 2 levels "0","1": 1 2 1 1 1 2 2 2 1 1 ...
##  $ log_dataUsage       : num  1.647 2.242 0.503 2.286 2.11 ...
```