Univariate EDA

Dynasty

2022-07-14

knitr::opts\_chunk$set(echo = TRUE)

# **Univariate EDA**

# Univariate Graphical Exploratory Data Analysis

## 1. Measures of Central Tendency

Before embarking on developing statistical models and generating predictions, it is essential to understand our data. This is typically done using conventional numerical and graphical methods.

We will be using the hills dataset, this dataset contains information on hill climbs made by various athletes

library(MASS)

## Warning: package 'MASS' was built under R version 4.1.3

head(hills)

## dist climb time  
## Greenmantle 2.5 650 16.083  
## Carnethy 6.0 2500 48.350  
## Craig Dunain 6.0 900 33.650  
## Ben Rha 7.5 800 45.600  
## Ben Lomond 8.0 3070 62.267  
## Goatfell 8.0 2866 73.217

Rows and columns

dim(hills)

## [1] 35 3

#### Mean Code Example 1.1

Find the mean of the distance covered by the athletes and assigning the mean to the variable athletes.dist.mean

athletes.dist.mean <- mean(hills$dist)  
athletes.dist.mean

## [1] 7.528571

The mean distance covered is 7.528571

#### Median Code Example 1.2

Finding the median which is the middle most value of the distance covered dist

athletes.dist.median <- median(hills$dist)  
athletes.dist.median

## [1] 6

The meadian is 6

#### Mode Code Example 1.3

Find the mode which is the value that has highest number of occurrences in a set of data.

Unfotunately, R does not have a standard in-built function to calculate mode so we have to build one, We create the mode function that will perform our mode operation for us

getmode <- function(v) {  
 uniqv <- unique(v)  
 uniqv[which.max(tabulate(match(v, uniqv)))]  
}

Now we Calculate the mode using out getmode() function

athletes.dist.mode <- getmode(hills$dist)  
athletes.dist.mode

## [1] 6

##### Let’s Challenge Ourselves

Will Find the mean, median, mode of the total evening calls given the following dataset

Dataset url = <http://bit.ly/CustomerSignatureforChurnAnalysis>

library(data.table)

## Warning: package 'data.table' was built under R version 4.1.2

churn <- fread('http://bit.ly/CustomerSignatureforChurnAnalysis')  
head(churn)

## recordID state account\_length area\_code international\_plan voice\_mail\_plan  
## 1: 1 HI 101 510 no no  
## 2: 2 MT 137 510 no no  
## 3: 3 OH 103 408 no yes  
## 4: 4 NM 99 415 no no  
## 5: 5 SC 108 415 no no  
## 6: 6 IA 117 415 no no  
## number\_vmail\_messages total\_day\_minutes total\_day\_calls total\_day\_charge  
## 1: 0 70.9 123 12.05  
## 2: 0 223.6 86 38.01  
## 3: 29 294.7 95 50.10  
## 4: 0 216.8 123 36.86  
## 5: 0 197.4 78 33.56  
## 6: 0 226.5 85 38.51  
## total\_eve\_minutes total\_eve\_calls total\_eve\_charge total\_night\_minutes  
## 1: 211.9 73 18.01 236.0  
## 2: 244.8 139 20.81 94.2  
## 3: 237.3 105 20.17 300.3  
## 4: 126.4 88 10.74 220.6  
## 5: 124.0 101 10.54 204.5  
## 6: 141.6 68 12.04 223.0  
## total\_night\_calls total\_night\_charge total\_intl\_minutes total\_intl\_calls  
## 1: 73 10.62 10.6 3  
## 2: 81 4.24 9.5 7  
## 3: 127 13.51 13.7 6  
## 4: 82 9.93 15.7 2  
## 5: 107 9.20 7.7 4  
## 6: 90 10.04 6.9 5  
## total\_intl\_charge number\_customer\_service\_calls churn customer\_id  
## 1: 2.86 3 no 23383607  
## 2: 2.57 0 no 22550362  
## 3: 3.70 1 no 59063354  
## 4: 4.24 1 no 25464504  
## 5: 2.08 2 no 691824  
## 6: 1.86 1 no 24456543

Let’s see number of rows and columns

dim(churn)

## [1] 12892 22

#### **Data cleaning**

Let’s do some data cleaning

**Checking for Missing values**

is.null(churn)

## [1] FALSE

We don’t have null values.

**Checking for Duplicates**

churn\_duplicated <- churn[duplicated(churn),]  
churn\_duplicated

## recordID state account\_length area\_code international\_plan  
## 1: 2 MT 137 510 no  
## 2: 3 OH 103 408 no  
## 3: 4 NM 99 415 no  
## 4: 5 SC 108 415 no  
## 5: 6 IA 117 415 no  
## ---   
## 12886: 12888 MT 25 415 no  
## 12887: 12889 MT 113 415 no  
## 12888: 12890 ID 88 415 no  
## 12889: 12891 AK 120 415 no  
## 12890: 12892 UT 74 415 no  
## voice\_mail\_plan number\_vmail\_messages total\_day\_minutes total\_day\_calls  
## 1: no 0 223.6 86  
## 2: yes 29 294.7 95  
## 3: no 0 216.8 123  
## 4: no 0 197.4 78  
## 5: no 0 226.5 85  
## ---   
## 12886: no 0 134.3 98  
## 12887: no 0 215.9 93  
## 12888: yes 31 181.6 91  
## 12889: no 0 178.4 97  
## 12890: no 0 106.4 84  
## total\_day\_charge total\_eve\_minutes total\_eve\_calls total\_eve\_charge  
## 1: 38.01 244.8 139 20.81  
## 2: 50.10 237.3 105 20.17  
## 3: 36.86 126.4 88 10.74  
## 4: 33.56 124.0 101 10.54  
## 5: 38.51 141.6 68 12.04  
## ---   
## 12886: 22.83 202.3 109 17.20  
## 12887: 36.70 240.1 85 20.41  
## 12888: 30.87 213.2 120 18.12  
## 12889: 30.33 168.3 113 14.31  
## 12890: 18.09 140.2 104 11.92  
## total\_night\_minutes total\_night\_calls total\_night\_charge  
## 1: 94.2 81 4.24  
## 2: 300.3 127 13.51  
## 3: 220.6 82 9.93  
## 4: 204.5 107 9.20  
## 5: 223.0 90 10.04  
## ---   
## 12886: 195.9 100 8.82  
## 12887: 156.7 123 7.05  
## 12888: 207.8 104 9.35  
## 12889: 120.5 93 5.42  
## 12890: 90.9 81 4.09  
## total\_intl\_minutes total\_intl\_calls total\_intl\_charge  
## 1: 9.5 7 2.57  
## 2: 13.7 6 3.70  
## 3: 15.7 2 4.24  
## 4: 7.7 4 2.08  
## 5: 6.9 5 1.86  
## ---   
## 12886: 12.6 5 3.40  
## 12887: 4.9 5 1.32  
## 12888: 11.4 4 3.08  
## 12889: 9.3 9 2.51  
## 12890: 11.4 3 3.08  
## number\_customer\_service\_calls churn customer\_id  
## 1: 0 no 22550362  
## 2: 1 no 59063354  
## 3: 1 no 25464504  
## 4: 2 no 691824  
## 5: 1 no 24456543  
## ---   
## 12886: 2 no 3785730  
## 12887: 3 no 25171109  
## 12888: 1 no 12126991  
## 12889: 1 no 33084674  
## 12890: 1 no 28432623

unique(churn)

## recordID state account\_length area\_code international\_plan  
## 1: 1 HI 101 510 no  
## 2: 2 MT 137 510 no  
## 3: 3 OH 103 408 no  
## 4: 4 NM 99 415 no  
## 5: 5 SC 108 415 no  
## ---   
## 12888: 12888 MT 25 415 no  
## 12889: 12889 MT 113 415 no  
## 12890: 12890 ID 88 415 no  
## 12891: 12891 AK 120 415 no  
## 12892: 12892 UT 74 415 no  
## voice\_mail\_plan number\_vmail\_messages total\_day\_minutes total\_day\_calls  
## 1: no 0 70.9 123  
## 2: no 0 223.6 86  
## 3: yes 29 294.7 95  
## 4: no 0 216.8 123  
## 5: no 0 197.4 78  
## ---   
## 12888: no 0 134.3 98  
## 12889: no 0 215.9 93  
## 12890: yes 31 181.6 91  
## 12891: no 0 178.4 97  
## 12892: no 0 106.4 84  
## total\_day\_charge total\_eve\_minutes total\_eve\_calls total\_eve\_charge  
## 1: 12.05 211.9 73 18.01  
## 2: 38.01 244.8 139 20.81  
## 3: 50.10 237.3 105 20.17  
## 4: 36.86 126.4 88 10.74  
## 5: 33.56 124.0 101 10.54  
## ---   
## 12888: 22.83 202.3 109 17.20  
## 12889: 36.70 240.1 85 20.41  
## 12890: 30.87 213.2 120 18.12  
## 12891: 30.33 168.3 113 14.31  
## 12892: 18.09 140.2 104 11.92  
## total\_night\_minutes total\_night\_calls total\_night\_charge  
## 1: 236.0 73 10.62  
## 2: 94.2 81 4.24  
## 3: 300.3 127 13.51  
## 4: 220.6 82 9.93  
## 5: 204.5 107 9.20  
## ---   
## 12888: 195.9 100 8.82  
## 12889: 156.7 123 7.05  
## 12890: 207.8 104 9.35  
## 12891: 120.5 93 5.42  
## 12892: 90.9 81 4.09  
## total\_intl\_minutes total\_intl\_calls total\_intl\_charge  
## 1: 10.6 3 2.86  
## 2: 9.5 7 2.57  
## 3: 13.7 6 3.70  
## 4: 15.7 2 4.24  
## 5: 7.7 4 2.08  
## ---   
## 12888: 12.6 5 3.40  
## 12889: 4.9 5 1.32  
## 12890: 11.4 4 3.08  
## 12891: 9.3 9 2.51  
## 12892: 11.4 3 3.08  
## number\_customer\_service\_calls churn customer\_id  
## 1: 3 no 23383607  
## 2: 0 no 22550362  
## 3: 1 no 59063354  
## 4: 1 no 25464504  
## 5: 2 no 691824  
## ---   
## 12888: 2 no 3785730  
## 12889: 3 no 25171109  
## 12890: 1 no 12126991  
## 12891: 1 no 33084674  
## 12892: 1 no 28432623

We have no duplicates

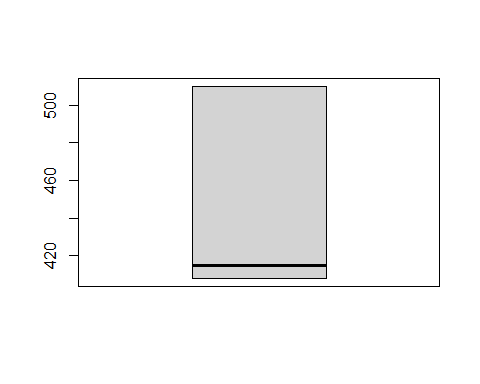
**Checking for Outliers**

library("ggplot2") # Load ggplot2

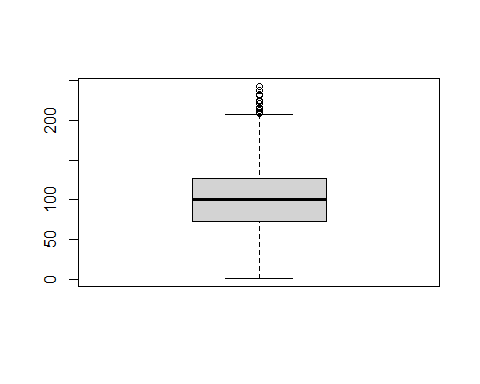
## Warning: package 'ggplot2' was built under R version 4.1.3

We will focus on the numeric columns

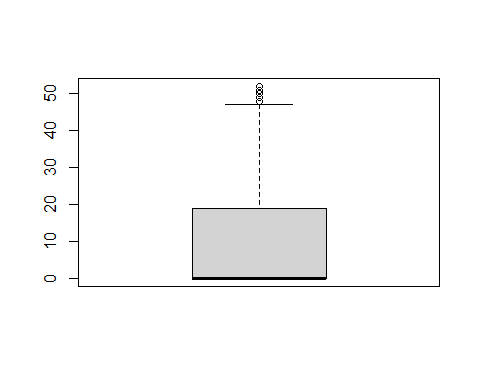
boxplot(churn$area\_code)



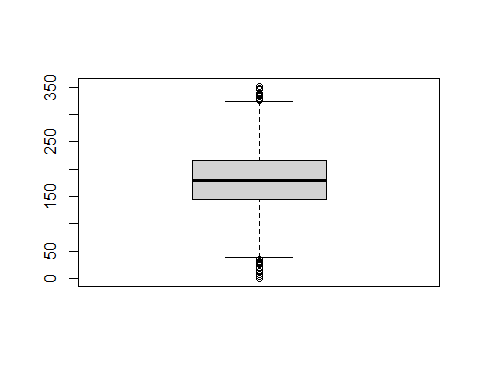
boxplot(churn$account\_length)



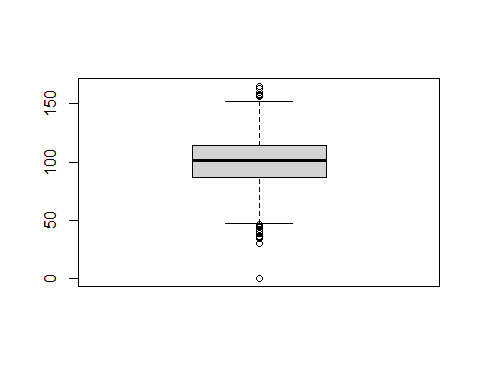
boxplot(churn$number\_vmail\_messages)



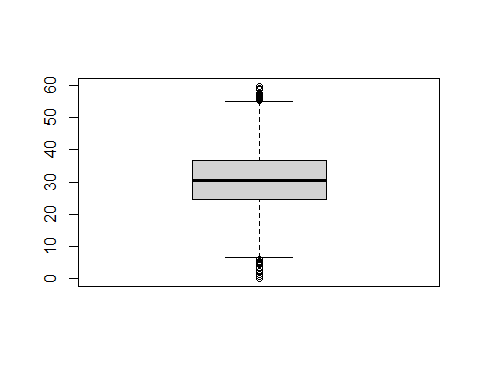
boxplot(churn$total\_day\_minutes)



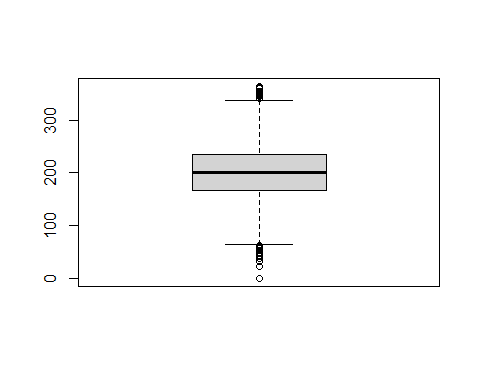
boxplot(churn$total\_day\_calls)



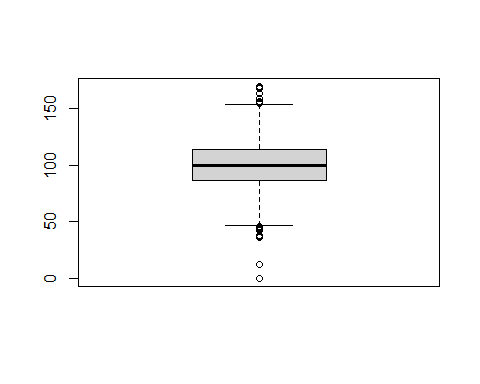
boxplot(churn$total\_day\_charge)



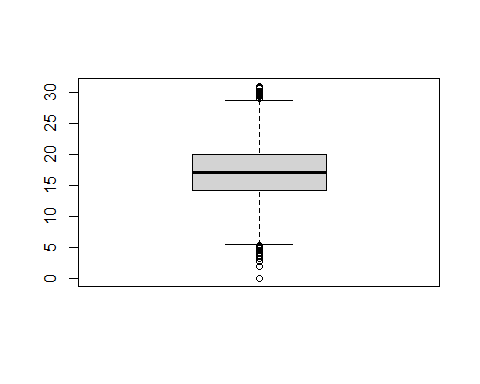
boxplot(churn$total\_eve\_minutes)



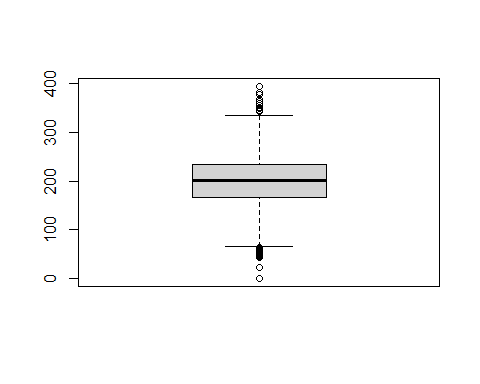
boxplot(churn$total\_eve\_calls)



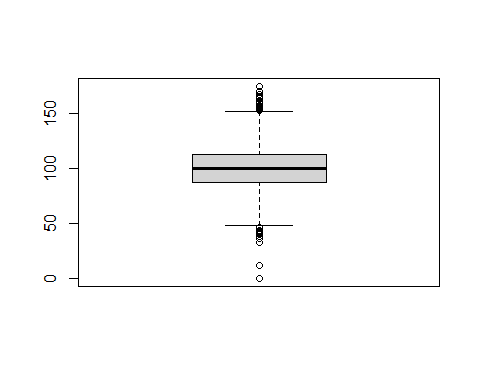
boxplot(churn$total\_eve\_charge)



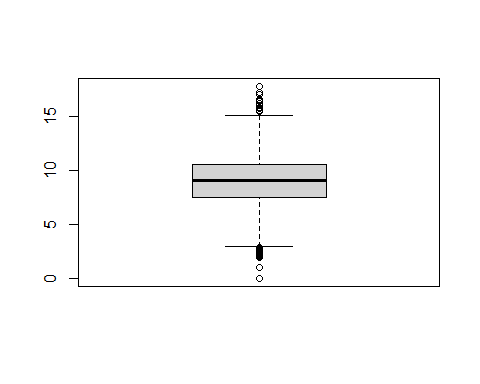
boxplot(churn$total\_night\_minutes)



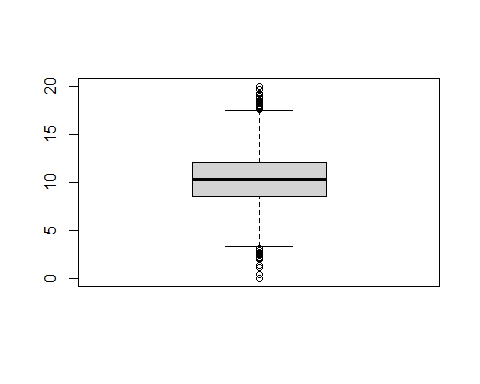
boxplot(churn$total\_night\_calls)



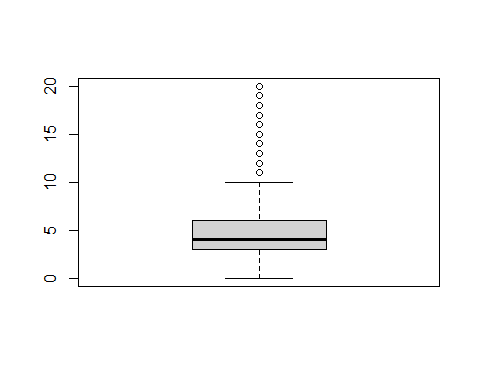
boxplot(churn$total\_night\_charge)



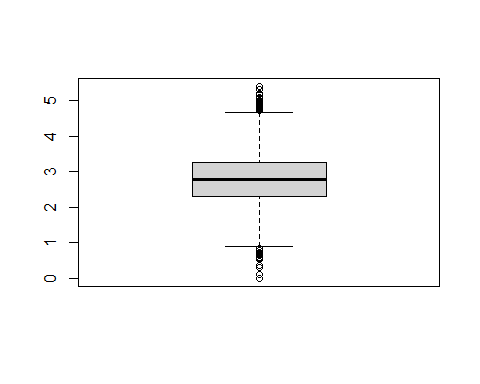
boxplot(churn$total\_intl\_minutes)



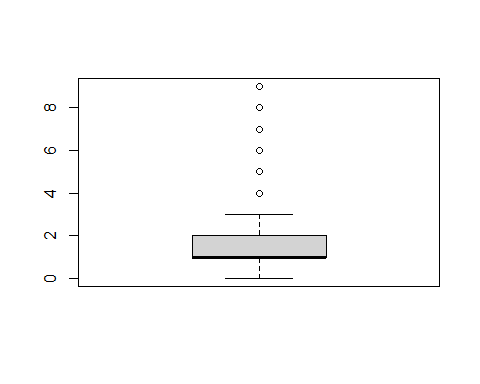
boxplot(churn$total\_intl\_calls)



boxplot(churn$total\_intl\_charge)



boxplot(churn$number\_customer\_service\_calls)



we have some columns with outliers

### **Questions**

**Find the minimum of total day calls**

churn.dist.min <- min(churn$total\_day\_calls)  
churn.dist.min

## [1] 0

**Find the maximum i.e. max() total day calls**

churn.dist.max <- max(churn$total\_day\_calls)  
churn.dist.max

## [1] 165

**Find the range i.e. range() of total day calls**

churn.dist.range <- range(churn$total\_day\_calls)  
churn.dist.range

## [1] 0 165

**Find the quantile of total day calls**

churn.dist.quantile <- quantile(churn$total\_day\_calls)  
churn.dist.quantile

## 0% 25% 50% 75% 100%   
## 0 87 101 114 165

**Find the variance of total day calls**

churn.dist.variance <- var(churn$total\_day\_calls)  
churn.dist.variance

## [1] 397.8691

**Find the standard deviation of total day calls**

churn.dist.sd <- sd(churn$total\_day\_calls)  
churn.dist.sd

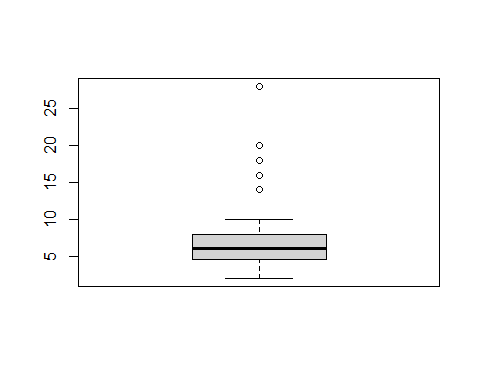
## [1] 19.94666

## **3. Univariate Graphical**

#### Box Plots Code Example 3.1

Lets create a boxplot graph for the distance using the boxplot() function

boxplot(hills$dist)



The box plot of an observation variable is a graphical representation based on its quartiles, as well as its smallest and largest values. It attempts to provide a visual shape of the data distribution.

#### Bar Graph Code Example 3.2

A bar graph of a qualitative data sample consists of vertical parallel bars that shows the frequency distribution graphically.

Let’s Create a frequency distribution of the School variable using an R built-in database named painters

head(painters)

## Composition Drawing Colour Expression School  
## Da Udine 10 8 16 3 A  
## Da Vinci 15 16 4 14 A  
## Del Piombo 8 13 16 7 A  
## Del Sarto 12 16 9 8 A  
## Fr. Penni 0 15 8 0 A  
## Guilio Romano 15 16 4 14 A

dim(painters)

## [1] 54 5

First Fetch the school column

school <- painters$School

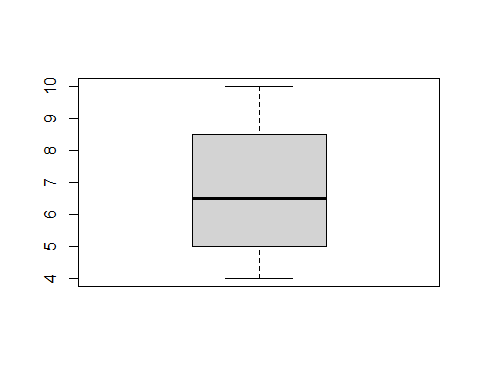
When we apply the table() function will compute the frequency distribution of the School variable

school\_frequency <- table(school)  
school\_frequency

## school  
## A B C D E F G H   
## 10 6 6 10 7 4 7 4

Now we apply the barplot function to produce its bar graph

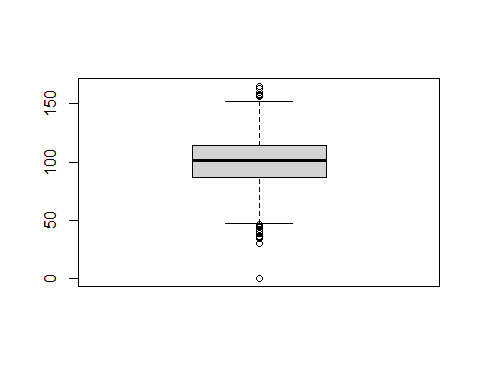
boxplot(school\_frequency)



##### **Challenge**

Now we challenge ourselves, will create a bar graph of the total day calls in the customer signature dataset

boxplot(churn$total\_day\_calls)



#### **Histogram Code Example 3.3**

A histogram shows the frequency distribution of a quantitative variable. The area of each bar is equal to the frequency of items found in each class.

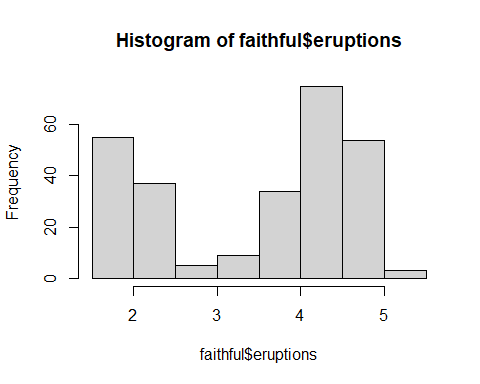
Will Create a histogram using the faithful dataset

head(faithful)

## eruptions waiting  
## 1 3.600 79  
## 2 1.800 54  
## 3 3.333 74  
## 4 2.283 62  
## 5 4.533 85  
## 6 2.883 55

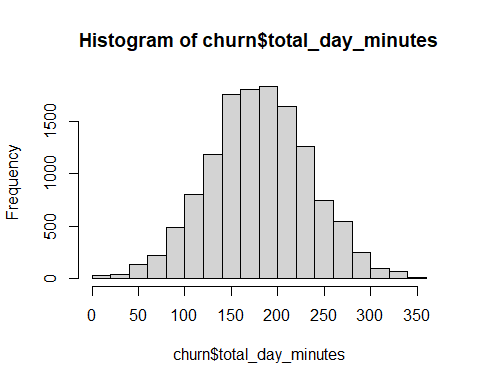
Then applying the hist() function to produce the histogram of the eruptions variable

hist(faithful$eruptions)



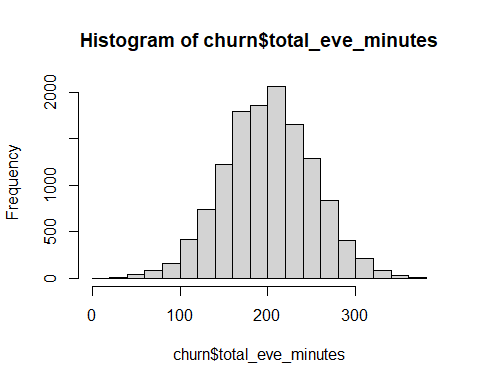
Now let’s Create a histogram of the total day minutes in the customer signature dataset

hist(churn$total\_day\_minutes)

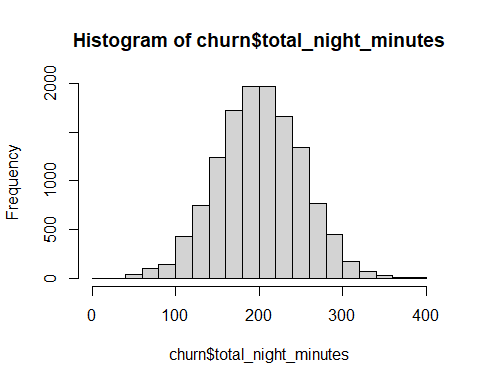


Let’s have more fun with the churn data set

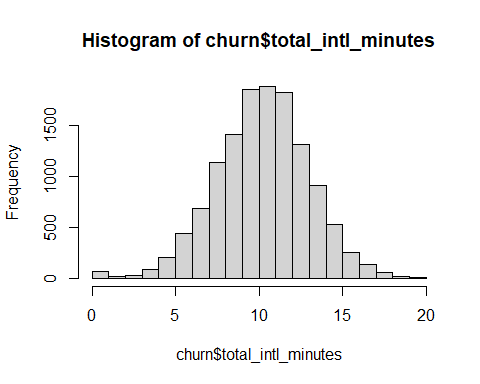
hist(churn$total\_eve\_minutes)



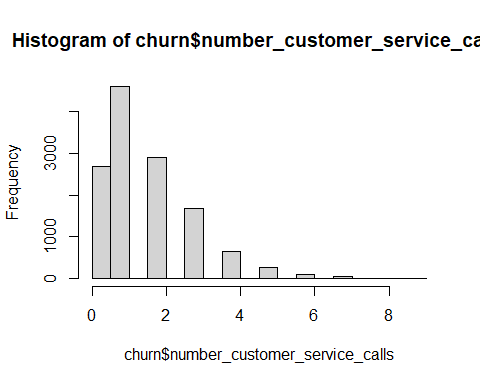
hist(churn$total\_night\_minutes)



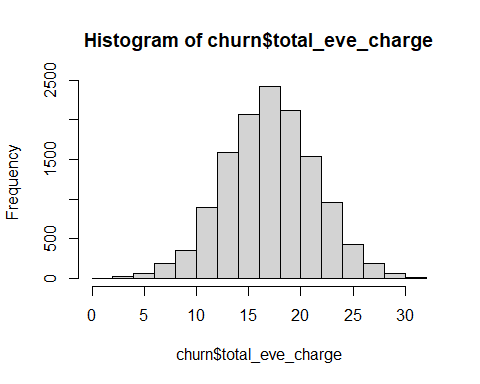
hist(churn$total\_intl\_minutes)



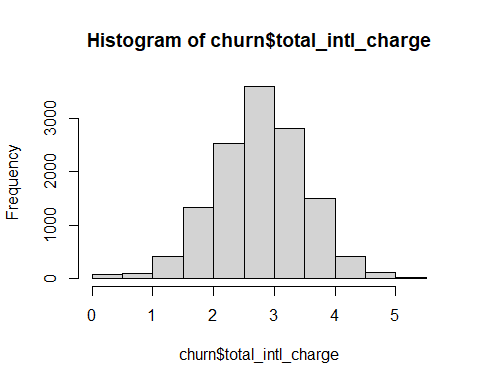
hist(churn$number\_customer\_service\_calls)



hist(churn$total\_eve\_charge)



hist(churn$total\_intl\_charge)



hist(churn$total\_night\_charge)

