DA5030.A1.Parpattedar

Shruti Parpattedar

January 21, 2019

Question 1

Loaded RStudio and RMarkdowns

Question 2

Loading given dataset and assigning columns names

```
library(readr)
data <- read_csv("customertxndata.csv", col_names = FALSE)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   X2 = col_double(),
##   X3 = col_character(),
##   X4 = col_character(),
##   X5 = col_double()
## )
```

```
names(data) <- c("Visits", "Tranx", "OS", "Gender", "Revenue")
```

Question 3

Calculating summative statistics: total transaction amount (revenue), mean number of visits, median revenue, standard deviation of revenue, most common gender.

```
total_revenue <- sum(data$Revenue, na.rm = TRUE)
mean_visits <- mean(data$Visits, na.rm = TRUE)
median_revenue <- median(data$Revenue, na.rm = TRUE)
sd_revenue <- sd(data$Revenue, na.rm = TRUE)
getMode <- function(x) {
  nas <- is.na(x)
  y <- x[-nas]
  uniq <- unique(y)
  uniq[which.max(tabulate(match(y, uniq)))]
}
gender_mode <- getMode(data$Gender)

output <- data.frame(total_revenue, mean_visits, median_revenue, sd_revenue, gender_mode)
output
```

```
##   total_revenue mean_visits median_revenue sd_revenue gender_mode
## 1      10372524    12.48649       344.6516   425.9884        Male
```

Question 4

Plotting a column graph for gender v/s revenue.

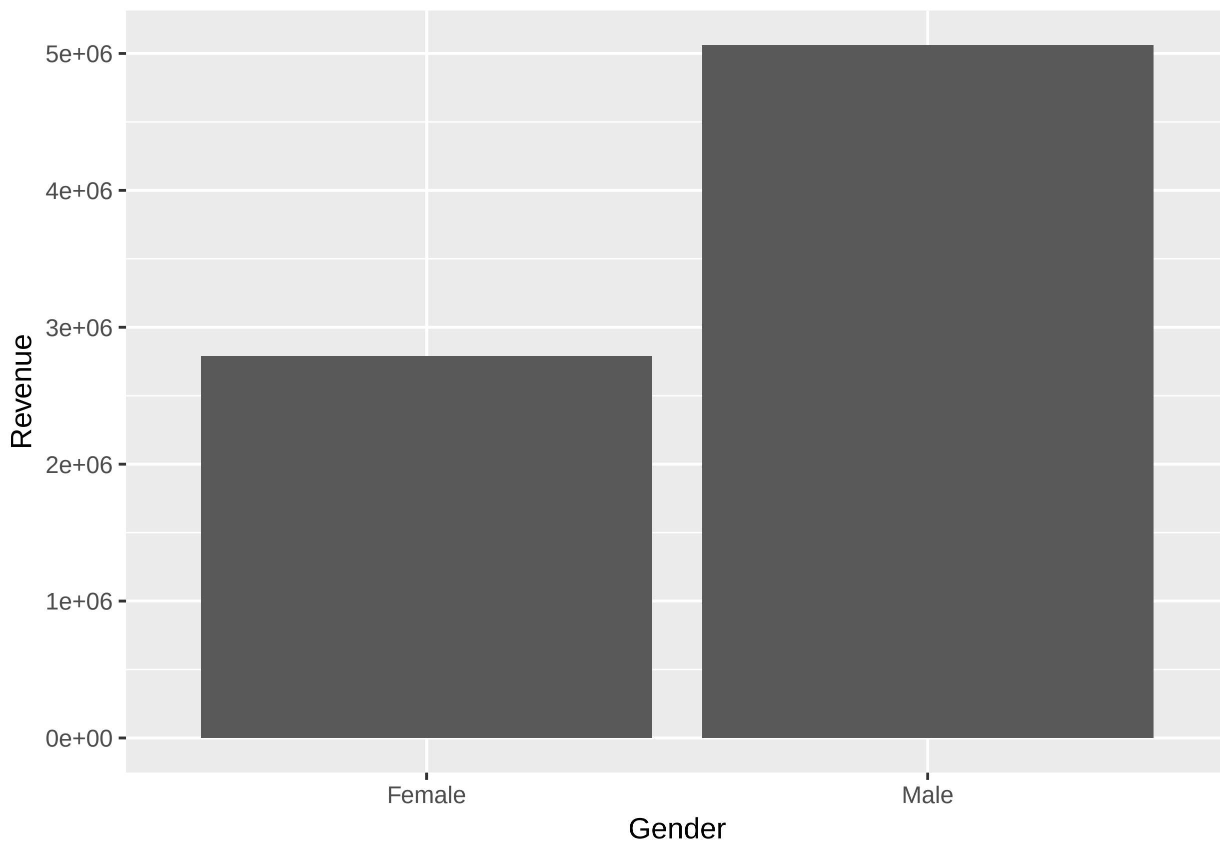
```r
library(magrittr)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------ tidyverse 1.2.1 --

## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v purrr   0.2.5     v forcats 0.3.0

## -- Conflicts --------------------------------------------------------- tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```r
data %>%
  select(Gender, Revenue) %>%
  filter(!is.na(Gender)) %>%
  ggplot(aes(Gender, Revenue)) + geom_col(na.rm = TRUE)
```



Question 5

Correlation between number of visits and revenue - The correlation obtained is ~0.739 which is high. So this means that there exists a positive correlation between the number of visits and the revenue earned. So, that means that there is a tendency for the revenue earned to increase with an increase in the number of visits.

```
cor(data$Visits, data$Revenue)
```

```
## [1] 0.7388448
```

Question 6

In order to find columns that contain missing values(NAs), I applied the unique function to each of the columns and observed the results for NA. By doing this, I found that the columns for Transactions and Gender contain NAs.

In the transaction column, there are 1800 NAs. This is approximately 7.9% of the total data. This is not an exceptionally high percentage so we could handle the missing data by imputing the average transactions in their place. Since, the percentage of imputed values is not very high the standard deviation would not vary too much.

In the Gender column, there are 5400 NAs. This is approximately 23.9% of the total data. This is a considerably high value so deletion would affect the statistics of the data. So, we could handle the missing data either by imputing the mode of the gender values or by creating a decision tree which woud predict the gender of the missing values.

```
unique(data$Visits)
```

```
##  [1]  7 20 22 24  1 13 23 14 11 17  2  8 18 16 25  3  0 19  6 10  9  5 21
## [24] 12 15  4
```

```
unique(data$Tranx)
```

```
## [1]  0  1  2 NA
```

```
unique(data$OS)
```

```
## [1] "Android" "iOS"
```

```
unique(data$Gender)
```

```
## [1] "Male"   NA       "Female"
```

```
unique(data$Revenue)
```

```
##   [1]    0.0000  576.8668  850.0000 1050.0000  460.0000 1850.0000  480.0000
##   [8]  110.0000 1950.0000  225.0000  344.6516 1300.0000  990.3062  405.2441
##  [15]  550.0000 1500.0000  330.0000  121.7745 1222.5214  676.7308  187.1906
##  [22]  360.0000  320.0000  340.0000  210.0000  450.0000  380.0000  420.0000
##  [29]  410.0000  925.0000  180.0000  725.0000  190.0000 1000.0000  296.2173
##  [36]  775.0000  260.0000  675.0000  150.0000  290.0000  200.0000  280.0000
##  [43]  300.0000  975.0000 1700.0000  230.8013  575.0000  825.0000  339.8281
##  [50]  430.0000  318.0227  750.0000  140.0000  143.5799  270.0000  900.0000
##  [57]  720.3415  500.0000  240.0000  230.0000  350.0000  130.0000  490.0000
##  [64]  470.0000  252.6066  600.0000  390.0000  275.0000  220.0000  160.0000
##  [71]  440.0000 1150.0000  274.4120  950.0000  400.0000  361.6334 1100.0000
##  [78]  310.0000  170.0000 1450.0000  427.0495  700.0000 1550.0000  120.0000
##  [85]  370.0000  650.0000  165.3852  383.4388 1600.0000  625.0000 1750.0000
##  [92] 1650.0000 2000.0000  742.1469  250.0000  800.0000 1200.0000  375.0000
```

```
##  [99] 1800.0000  525.0000  698.5362  475.0000  425.0000 1900.0000 1250.0000
## [106] 1350.0000  100.0000  875.0000  325.0000  785.7576 1400.0000  807.5629
## [113]  763.9522  654.9254  448.8548
```

```
data %>%
  select(Tranx) %>%
  filter(is.na(Tranx)) %>%
  summarise(Tranx_NAs = n())
```

```
## # A tibble: 1 x 1
##   Tranx_NAs
##       <int>
## 1      1800
```

```
data %>%
  select(Gender) %>%
  filter(is.na(Gender)) %>%
  summarise(Gender_NAs = n())
```

```
## # A tibble: 1 x 1
##   Gender_NAs
##        <int>
## 1       5400
```

Question 7

Imputing the missing values in the transaction column by the average rounded to the nearest whole number and those in the gender column by the mode (Male). I have used the function defined in Q3 for this purpose.

```
Tranx_NA <- is.na(data$Tranx)
Gender_NA <- is.na(data$Gender)
data$Tranx[Tranx_NA] <- round(mean(data$Tranx, na.rm = TRUE))
data$Gender[Gender_NA] <- getMode(data$Gender)
```

Question 8

Dividing the dataset into training and Validation datasets. Training - odd row numbers Validation - even row numbers

```
training <- data[seq(1, dim(data)[1], 2),]
validation <- data[seq(2, dim(data)[1], 2),]
```

Question 9

Comparing the mean revenue in the training and validation datasets. The difference between the means tells us that taking every alternate row for the datasets was not the ideal way to divide the dataset ie. the data is not evenly distributed in the two datasets.

```
train_mean <- mean(training$Revenue)
validation_mean <- mean(validation$Revenue)
train_mean
```

```
## [1] 449.6105
```

```
validation_mean
```

```
## [1] 460.26
```

Question 10

Generating sample sizes to creating subsets for training, testing and validation.

```
set.seed(77654)

sample <- sample.int(n = nrow(data), size = 0.6*nrow(data), replace = FALSE)
train <- data[sample,]

rem <- data[-sample,]
sample2 <- sample.int(n = nrow(rem), size = 0.2*nrow(data), replace = FALSE)
test <- data[sample2,]
validn <- data[-sample2,]
```