Practice / First Steps in Data Analytics

Dr. Nishat Mohammad

01/18/2024

Customer Transacation Data

```
# Question 1:
#Load the data set from the link above into a data frame. Inspect the data set using the str() function
custxndata <- as.data.frame (read.csv("customertxndata.csv"))
str(custxndata)
```

```
## 'data.frame':    22800 obs. of  5 variables:
##  $ numvisits: int  7 20 22 24 1 13 23 14 11 24 ...
##  $ numtxns  : int  0 1 1 2 0 1 2 1 1 2 ...
##  $ OS       : chr  "Android" "iOS" "iOS" "iOS" ...
##  $ gender   : chr  "Male" NA "Female" "Female" ...
##  $ rev      : num  0 577 850 1050 0 ...
```

```
# Get the Number of customers by using the number rows in the data
# I am not using the total gender (Male + Female) which would be 17,400 as the number of customers beca
customers <- nrow(custxndata)
data_vars <- colnames(custxndata)
```

The data reveals 22,800 customers with the following data: "numvisits", "numtxns", "OS", "gender", "rev"

```
##    numvisits        numtxns          OS               gender               rev
##  Min.   : 0.00   Min.   :0.000   Length:22800       Length:22800       Min.   :   0.0
##  1st Qu.: 6.00   1st Qu.:1.000   Class :character   Class :character   1st Qu.: 170.0
##  Median :12.00   Median :1.000   Mode  :character   Mode  :character   Median : 344.7
##  Mean   :12.49   Mean   :0.993                                         Mean   : 454.9
##  3rd Qu.:19.00   3rd Qu.:1.000                                         3rd Qu.: 576.9
##  Max.   :25.00   Max.   :2.000                                         Max.   :2000.0
##                  NA's   :1800
```

```
## # A tibble: 2 x 2
## # Groups:   gender [2]
##   gender     n
##   <chr>  <int>
## 1 Female  2670
## 2 Male   14730
```

The summative statistics are shown in the summary above, the number of males and females for customers who shared their gender are also shown in the table.

Data Analysis

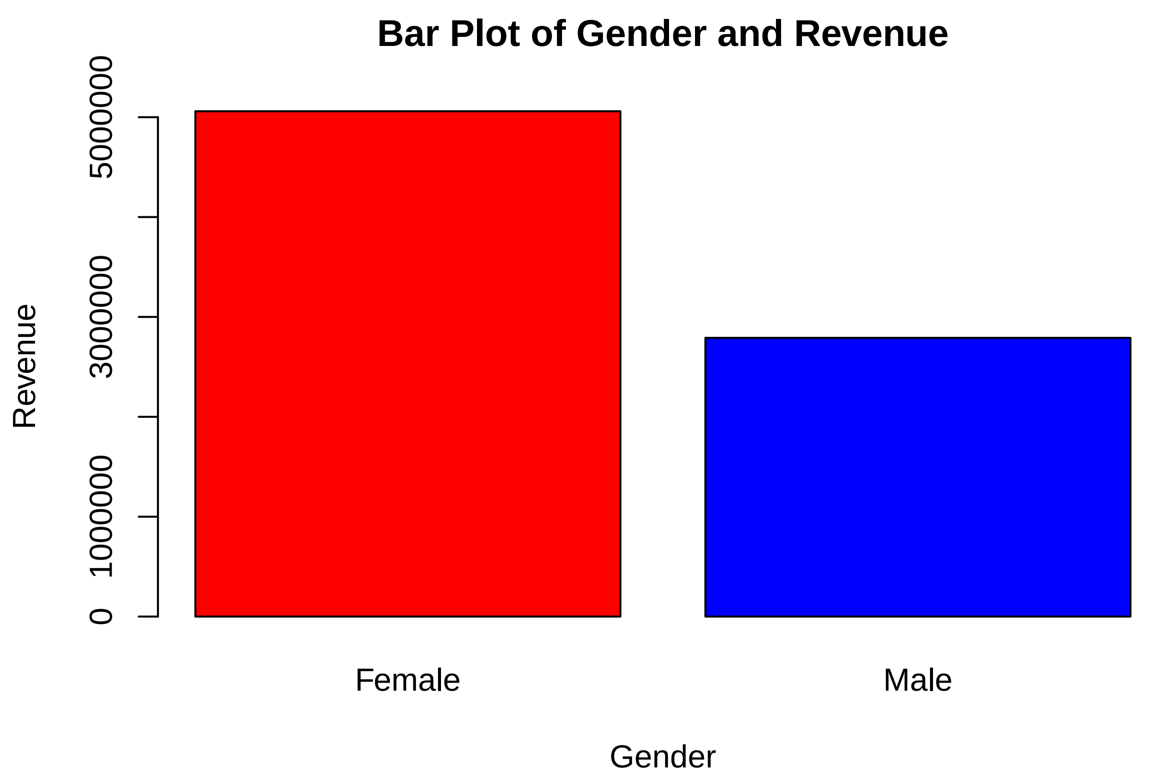
There were 22,800 customers and the mean number of visits per customer was 12.5. The median revenue was US$ 344.7 ($\sigma = 426$). Most of the visitors were "Male".

Visualizing Revenue Disparity in Gender

```r
#Question 4:
#Create a bar (aka column) chart of gender (x-axis) versus revenue (y-axis). Omit missing values, i.e.,

Gender <- custxndata$gender
Revenue <- custxndata$rev
male_revenue <- sum(Revenue[which(Gender == "Male")])
female_revenue <- sum(Revenue[which(Gender == "Female")])

# Visualize with Barplot
barplot(c(male_revenue,female_revenue),
     main = "Bar Plot of Gender and Revenue",
     xlab = "Gender",ylab = "Revenue",
     names.arg = c("Female", "Male"),
     col = c("red","blue"))
```



The Bar Plot is titled Bar Plot of Gender and Revenue, with gender and revenue on the x and y axes respectively. males are in blue and females in red. The revenue of females is higher than males,female revenue is slightly above 500,000 and male revenue is close to 300,000.

Correlation Data

```
## Question 5:
#What is the Pearson Moment of Correlation between number of visits and revenue? Comment on the correla

Pcor_rev_visits <- cor(x=custxndata$numvisits, y=custxndata$rev, use = "complete.obs"
    , method = "pearson")
```

The Pearson correlation between number of visits and revenue is 0.739, It is positive and depicts that the two variables are directly proportional, increase in number of visits is commensurate with increase in revenue.

Missing Data Highlights and Analysis:

```
## Question 6:
#Which columns have missing data? How did you recognize them? How would you impute missing values? In y

# Recognize the columns with NA using colSums()
total_na<- colSums(is.na(custxndata))
total_na
```

```
## numvisits   numtxns        OS    gender       rev
##         0      1800         0      5400         0
```

```
gender_na <- length(which(is.na(custxndata$gender)))
txnnum_na <- length(which(is.na(custxndata$numtxns)))
# Recognize the columns with NA using summary()
custxndata_summary <- within(custxndata, {
  gender <- factor(gender, labels = c("F", "M"))
  customer_os <- factor(OS, labels = c("Android", "iOS"))
})
summary(custxndata_summary)
```

```
##    numvisits        numtxns            OS               gender          rev           customer_os
##  Min.   : 0.00   Min.   :0.000   Length:22800       F   : 2670   Min.   :   0.0   Android:16028
##  1st Qu.: 6.00   1st Qu.:1.000   Class :character   M   :14730   1st Qu.: 170.0   iOS    : 6772
##  Median :12.00   Median :1.000   Mode  :character   NA's: 5400   Median : 344.7
##  Mean   :12.49   Mean   :0.993                                   Mean   : 454.9
##  3rd Qu.:19.00   3rd Qu.:1.000                                   3rd Qu.: 576.9
##  Max.   :25.00   Max.   :2.000                                   Max.   :2000.0
##                  NA's   :1800
```

```
# Impute NA using na.omit
valid_data <- na.omit(custxndata)
summary(valid_data)
```

```
##    numvisits        numtxns             OS               gender               rev
##  Min.   : 0.00   Min.   :0.0000   Length:15600       Length:15600       Min.   :   0.0
##  1st Qu.: 6.00   1st Qu.:1.0000   Class :character   Class :character   1st Qu.: 140.0
##  Median :13.00   Median :1.0000   Mode  :character   Mode  :character   Median : 360.0
```

```
##  Mean   :12.59   Mean   :0.9914                                          Mean   : 465.5
##  3rd Qu.:19.00   3rd Qu.:1.0000                                          3rd Qu.: 600.0
##  Max.   :25.00   Max.   :2.0000                                          Max.   :2000.0
```

The missing data (NA) can be recognized using the summary() function or using is.na() in the colSums() function.
The total number of missing values are 7,200.
The Missing gender values are 5,400.
The Missing number of transactions are 1,800.

Imputation of Missing Values:

Imputation can be down using na.omit() or rm.na() and removal validated using is.na() again or summary() again. We can replace with the mean or mode for the transactions and assign a neutral gender to the missing gender values.

```r
## Question 7:
#Impute missing transaction and gender values. Use the mean for transaction (rounded to the nearest who

# Impute NA in transactions with mean value
mean_txn_col <- round(mean(custxndata$numtxns, na.rm = TRUE), digits = 0)
custxndata$numtxns[is.na(custxndata$numtxns)] <- mean_txn_col

# Impute NA in gender with mode
gender_mode <- max(custxndata$gender, na.rm = TRUE)
custxndata$gender[is.na(custxndata$gender)] <- gender_mode
```

```r
#Calculating the descriptive statistics again
summary(custxndata)
```

```
##    numvisits        numtxns            OS                gender               rev        
##  Min.   : 0.00   Min.   :0.0000   Length:22800       Length:22800       Min.   :   0.0  
##  1st Qu.: 6.00   1st Qu.:1.0000   Class :character   Class :character   1st Qu.: 170.0  
##  Median :12.00   Median :1.0000   Mode  :character   Mode  :character   Median : 344.7  
##  Mean   :12.49   Mean   :0.9936                                          Mean   : 454.9  
##  3rd Qu.:19.00   3rd Qu.:1.0000                                          3rd Qu.: 576.9  
##  Max.   :25.00   Max.   :2.0000                                          Max.   :2000.0  
```

```r
# Get the total transaction amount/Revenue
new_total_revenue <- sum(custxndata$rev, na.rm = T)

# Get the mean number of visits
new_mean_visits <- mean(custxndata$numvisits, na.rm = T)

# Get the median Revenue
new_median_revenue <- median(custxndata$rev, na.rm = T)

# Get the sd of Revenue
new_sd_revenue <- sd(custxndata$rev, na.rm = T)

# Get the most common gender
help(unique)
```

```r
help(match)
valid_gender <- unique(custxndata$gender)
new_most_common_gender <- valid_gender[which.max(tabulate(match(custxndata$gender,valid_gender)))]
new_most_common_gender
```

```
## [1] "Male"
```

```r
# Validate by using dplyr
library(dplyr)
new_customer_gender <- custxndata%>%
  filter(!is.na(gender)) %>%
  group_by(gender) %>%
  count()
# Look at the customer_gender to check if Male is highest
new_customer_gender
```

```
## # A tibble: 2 x 2
## # Groups:   gender [2]
##   gender     n
##   <chr>  <int>
## 1 Female  2670
## 2 Male   20130
```

Data Analysis After Imputation

There mean number of visits per customer was 12.5.
The median revenue was US$ 344.7 ($\sigma = 426$).
Most of the visitors were "Male".

Training and Validating sets:

```r
## Question 8:
#Split the data set into two equally sized data sets where one can be used for training a model and the


samp_cus_data <- custxndata %>%
  mutate(cus_data_split = seq_len(nrow(custxndata)) %% 2)

# Get rows in data to check the split
total_rows <- nrow(samp_cus_data)

# Train data set with odd case
training_dataset <- samp_cus_data %>%
  filter(cus_data_split == 1)

# Check split
Training_rows <- nrow(training_dataset)

# Validate data set with even case
validation_dataset <- samp_cus_data %>%
```

```r
  filter(cus_data_split == 0)
# Check equal split
validation_rows <- nrow(validation_dataset)
```

Splitting Data The data set has been split into 2; training and validation sets. All the odd rows in the training set and all the even rows in the validation set. The values in the training set are 11,400 while the values in the validation set are 11,400.

```r
## Question 9:
#Calculate the mean revenue for the training and the validation data sets and compare them. Comment on

# mean revenue training set
mean_rev_training <- mean(training_dataset$rev)

# mean revenue validation set
mean_rev_validation_df <- mean(validation_dataset$rev)
```

Data Analysis After Split The mean revenue for the training set is US$ 449.61 and that for the validation set is US$ 460.26. The values are not so far apart, the validation set revenue mean is a bit higher than the training set revenue mean.

Training, Testing and Validation sets

```r
## Question 10:
#For many data mining and machine learning tasks, there are packages in R. Use the sample() function to

final_split_data <- custxndata
set.seed(77654)

samp <- sample(seq(1,3), size = nrow(final_split_data), replace = T, prob = c(0.6, 0.2, 0.2))

train_split <- final_split_data[samp == 1, ]
test_split <- final_split_data[samp == 2, ]
valid_split <- final_split_data[samp == 3,]

# Look at Sample distribution
traning_vals <- nrow(train_split)
testing_vals <- nrow(test_split)
validation_vals <- nrow(valid_split)
```

Splitted the data into three:
The training data set with 13,690.
The testing data set with 4,578.
The validation data set with 4,532.