Practice / First Steps in Data Analytics

Prachi Sardana

## Question 1

Loading the customer data from url and structuring the customer data using str() function showing the transaction data:

customer\_data = read.csv(url("http://artificium.us/assignments/55.data-analytics/a-55-103/customertxndata.csv"))  
str(customer\_data)

## '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 ...

## Question 2

# Total revenue   
total\_revenue = sum(customer\_data$rev, na.rm = TRUE)  
print(total\_revenue)

## [1] 10372524

# Mean Number of visits   
num\_visits = mean(customer\_data$numvisits, na.rm = TRUE)  
print(num\_visits)

## [1] 12.48649

# Median revenue  
revenue = median(customer\_data$rev,na.rm = TRUE)  
print(revenue)

## [1] 344.6516

# standard deviation of revenue  
standard\_dev = sd(customer\_data$rev, na.rm = TRUE)  
print(standard\_dev)

## [1] 425.9884

# To find the most common gender   
  
# Created a vector to find the most common gender  
gender\_num <- c(customer\_data$gender)  
  
# Created the getmode function  
getmode <- function(gender\_num) {  
 uniqv <- unique(gender\_num)  
 uniqv[which.max(tabulate(match(gender\_num, uniqv)))]  
}  
  
# Calculate the mode using the user function.  
mode.gender <- getmode(gender\_num)  
print(mode.gender)

## [1] "Male"

The total revenue of all transactions is $ 10,372,524, while the median transaction amount is $ 344.65 with a standard deviation of 425.99. The average number of visits was 12. The majority of customers were “Male”.

## Question3

## Data analysis

There were 376 customers and the mean number of visits per customer was 3.4. The median revenue was US$ 35.8 (σ = 11.3). Most of the visitors were male.

# Number of Customers = Total Revenue / (Average Transaction Value \* Average Revenue per Visit)  
  
total\_revenue <- sum(customer\_data$rev)   
print(total\_revenue)

## [1] 10372524

# average revenue per visit = Total Revenue/ total number of visits  
  
total\_num\_visits <- sum(customer\_data$numvisits)  
print(total\_num\_visits)

## [1] 284692

avg\_revenue\_per\_visit <- total\_revenue/total\_num\_visits  
print(avg\_revenue\_per\_visit)

## [1] 36.43419

total\_transactions <- length(customer\_data$numtxns)  
print(total\_transactions)

## [1] 22800

average\_transac\_val <- total\_revenue/total\_transactions  
print(average\_transac\_val)

## [1] 454.9353

num\_customers <- total\_revenue/(average\_transac\_val\*avg\_revenue\_per\_visit)  
print(num\_customers)

## [1] 625.7858

mean\_visits\_per\_customer <- mean(table(customer\_data$numvisits))  
print(mean\_visits\_per\_customer)

## [1] 876.9231

std\_dev <- sd(customer\_data$rev)  
print(std\_dev)

## [1] 425.9884

# narrative   
# Created the getmode function  
getmode <- function(gender\_num) {  
 uniqv <- unique(gender\_num)  
 uniqv[which.max(tabulate(match(gender\_num, uniqv)))]  
}  
  
# Calculate the mode using the user function.  
mode.gender <- getmode(gender\_num)  
print(mode.gender)

## [1] "Male"

narrative <- sprintf("There were %f customers and the mean number of visits per customer was %f . The median revenue was US$ %f (σ = %f). Most of the visitors were %s.",num\_customers,mean\_visits\_per\_customer,revenue,std\_dev, mode.gender)  
  
  
  
cat(narrative)

## There were 625.785756 customers and the mean number of visits per customer was 876.923077 . The median revenue was US$ 344.651614 (σ = 425.988388). Most of the visitors were Male.

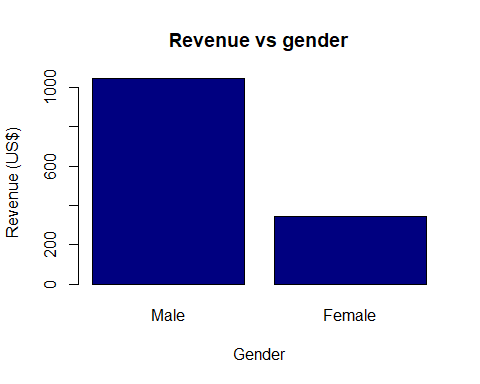
## Question 4

Create a bar (aka column) chart of gender (x-axis) versus revenue (y-axis). Omit missing values, i.e., where gender is NA or missing. Use the plot() function rather than functions from ggplot2. Show only the chart and not the code that generated it. Add markdown text to comment on what the chart means.

customer\_data = read.csv(url("http://artificium.us/assignments/55.data-analytics/a-55-103/customertxndata.csv"))  
str(customer\_data)

## '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 ...

customer\_data <- na.omit(customer\_data)  
  
customer\_data.agg <- aggregate(customer\_data$rev, list(customer\_data$gender), FUN=mean)  
  
barplot(customer\_data.agg$x,  
 main = "Revenue vs gender",  
 xlab = "Gender",  
 ylab = "Revenue (US$)",  
 names.arg = c("Male", "Female"),  
 col = "navy",  
 horiz = FALSE)



The bar chart indicates that in “male” customers the revenue was double compared to “female”.

## Question 5

# Number of visits  
x <- customer\_data$numvisits  
  
# Revenue   
y <- customer\_data$rev  
  
# correlation between number of visits and revenue of customer data  
cor\_xy <- cor.test(x, y, method=c("pearson"))

The pearson correlation coefficient is round(cor\_xy, 2) which indicates a moderate positive relationship between the number of visits and the revenue.

## Question 6

Which columns have missing data? How did you recognize them? How would you impute missing values? In your markdown, add comments on missing data and imputation strategies.

customer\_data = read.csv(url("http://artificium.us/assignments/55.data-analytics/a-55-103/customertxndata.csv"))  
#print(customer\_data)  
  
# To find out which coloumns have missing data, we can use is.na() function  
  
na.test <- function(customer\_data) {  
 missing\_counts <- sum(is.na(customer\_data))  
   
 if (missing\_counts > 0) {  
 print(names(customer\_data)[sapply(customer\_data, function(x) any(is.na(x)))])  
 } else {  
 print("No missing values found.")  
 }  
}  
  
na.test(customer\_data)

## [1] "numtxns" "gender"

# In order to impute values, the NA's can be replaced with 0. For example   
  
new\_customer\_gender <- replace(customer\_data$gender, is.na(customer\_data$gender),0)  
  
  
# Another imputation strategy is by replacing NA's with mean or median  
  
# Imputation with mean   
new\_customer\_transaction = replace(customer\_data$numtxns, is.na(customer\_data$numtxns), mean(customer\_data$numtxns, na.rm = TRUE))  
  
# Imputation with median  
new\_customer\_trans = replace(customer\_data$numtxns, is.na(customer\_data$numtxns), median(customer\_data$numtxns, na.rm = TRUE))

The data set has missing data for “gender” and “numtxns” by using *na.test* function specifically.There are different imputation strategies like replacing NAs with O or imputing with mean or median.

## Question7

Impute missing transaction and gender values. Use the mean for transaction (rounded to the nearest whole number) and the mode for gender. Recalculate the descriptive statistics of (2) and repeat the markdown of (3) with the new (computed) values. Comment in the markdown on the difference.

# Imputing the missing transaction value with mean number of transaction in customer data  
  
new\_customer\_transaction = replace(customer\_data$numtxns, is.na(customer\_data$numtxns), mean(customer\_data$numtxns, na.rm = TRUE))  
round\_trans <- round(new\_customer\_transaction)  
#print(round\_trans)  
  
# Imputing the missing value of gender\_num with mode function   
getmode <- function(gender\_num) {  
 uniqv <- unique(gender\_num)  
 uniqv[which.max(tabulate(match(gender\_num, uniqv)))]  
}  
  
# Calling the getmode function   
mode.gender <- getmode(gender\_num)  
print(mode.gender)

## [1] "Male"

# Total revenue   
total\_rev = sum(customer\_data$rev, na.rm = TRUE)  
print(total\_rev)

## [1] 10372524

# Mean Number of visits   
num\_visits = mean(customer\_data$numvisits, na.rm = TRUE)  
print(num\_visits)

## [1] 12.48649

# Median revenue  
revenue = median(customer\_data$rev,na.rm = TRUE)  
print(revenue)

## [1] 344.6516

# standard deviation of revenue  
standard\_dev = sd(customer\_data$rev, na.rm = TRUE)  
print(standard\_dev)

## [1] 425.9884

Using the imputed values, we have a total revenue of all transactions of $ 10,372,524, while the median transaction amount is $ 344.65 with a standard deviation of 425.99. The average number of visits was 12. The majority of customers were “Male”.

## Question 8

Splitting data into training data that consist of odd numbered cases and validation test for even number cases where larger set is used for training the model.On the other hand, smaller set is used for validation of the model.

# Created a new data frame x   
x <- data.frame(customer\_data)  
#print(x)  
  
# Calculated the number of rows in x   
nrow <- nrow(x)  
  
# select only odds rows  
# Training set   
train\_Set <- x[seq(1,nrow,by = 2),]  
head(train\_Set,4)

## numvisits numtxns OS gender rev  
## 1 7 0 Android Male 0  
## 3 22 1 iOS Female 850  
## 5 1 0 Android Male 0  
## 7 23 2 iOS Male 1850

# Test set   
test\_set<-x[seq(2,nrow,by=2),] #select only rows by 10  
head(test\_set,4)

## numvisits numtxns OS gender rev  
## 2 20 1 iOS <NA> 576.8668  
## 4 24 2 iOS Female 1050.0000  
## 6 13 1 Android Male 460.0000  
## 8 14 1 Android Male 480.0000

## Question 9

training\_set\_mean <- mean(train\_Set$rev)  
#print(training\_set\_mean)  
  
test\_set\_mean <- mean(test\_set$rev)  
#print(test\_set\_mean)  
  
t\_test <- t.test(train\_Set$rev, test\_set$rev)  
print(t\_test)

##   
## Welch Two Sample t-test  
##   
## data: train\_Set$rev and test\_set$rev  
## t = -1.8875, df = 22790, p-value = 0.0591  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -21.70830 0.40925  
## sample estimates:  
## mean of x mean of y   
## 449.6105 460.2600

# mean of training data set is less than mean of test data set which indicates that the data is not proportionally distributed.  
# Based on paired t test shows that difference is not statistically significant since p value is = 0.0591

Calculating the mean revenue for the training 449.61and the validation data sets 460.26 assuming that mean of training and test data is same but comparing them using **t**-test shows that **p-value** > 0.05 Based on the result the mean is not statistically significant as there is a slight differences in mean

## Question 10

Used a random sample of 60% for training.Since the remaining data is 40% and we need 20% of test data and 20% of validation data which means we need to split the remaining data into 50% to get test data and the validation data.

# Set the seed so that same sample can be reproduced in the future  
set.seed(77654)  
  
# Split the 60% of data into training data set  
sample <- sample.int(n = nrow(customer\_data), size = floor(.60\*nrow(customer\_data)), replace = F)  
train <- customer\_data[sample, ]  
  
  
# Since the remaining data is 40% and we need 20% of test data and 20% of validation data which means we need to split the remaining data into 50% to get test data and the validation data.  
remaining\_data <-customer\_data[-sample,]  
sample <- sample.int(n = nrow(remaining\_data), size = floor(0.50\*nrow(remaining\_data)),replace = F)  
  
test <- remaining\_data[sample, ]  
  
validation\_data <- remaining\_data[-sample,]  
#print(train)  
#print(test)  
#print(validation\_data)