BLACK FRIDAY SALES

2018

Totla, Rajshri

MET CS 544 FINAL PROJECT

12/16/2018

Contents

[Black Friday Sales 1](#_Toc532767383)

[About : This is a dataset of 550 000 observations about the Black Friday sales in a retail store, it contains different kinds of variables either numerical or categorical. It also contains some missing values. 1](#_Toc532767384)

[Project Requirements 3](#_Toc532767385)

[R-Packages Used 5](#_Toc532767386)

[ UsingR 5](#_Toc532767387)

[ sampling 5](#_Toc532767388)

[ stringr 5](#_Toc532767389)

[ tidyverse 5](#_Toc532767390)

[ stats 5](#_Toc532767391)

[ prob 5](#_Toc532767392)

[ dbplyr 6](#_Toc532767393)

[ dtplyr 6](#_Toc532767394)

[Executed R Code [Attached: Html Output File ] 7](#_Toc532767395)

# Black Friday Sales

### **About : This is a dataset of 550 000 observations about the Black Friday sales in a retail store, it contains different kinds of variables either numerical or categorical. It also contains some missing values.**

1. Downloaded data from the website [*https://www.kaggle.com/mehdidag/black-friday*](https://www.kaggle.com/mehdidag/black-friday)
2. Understood and explored data for its attributes and type.
3. Data we are using have following columns:

* *User\_ID (Numerical Variable)*
* *Product\_ID (Categorical Variable)*
* *Gender (Categorical Variable)*
* *Age (Categorical Variable, because it is in ranges)*
* *Occupation (Numerical Variable)*
* *City\_Category (Categorical Variable)*
* *Stay\_In\_Current\_City\_Years (Numerical Variable)*
* *Marital\_Status (Numerical Variable)*
* *Product\_Category\_1 (Numerical Variable)*
* *Product\_Category\_2 (Numerical Variable)*
* *Product\_Category\_3 (Numerical Variable)*
* *Purchase amount in dollars (Numerical Variable)*

1. **Data Gathering**

Convert the data into a format where it could be analyzed. We are not performing analysis on original data to maintain the integrity.

1. **Exploring Data**

Understand and explore data again. Check whether it is fit for analysis.

1. **Cleaning Data**

Data had some unwanted values or the data was not uniform. For example:

* Product Categories had NAs which we replaced with 0.
* Converted Age-range data into numerical values so that operations could be done on that. Earlier it was neither categorical nor numerical

1. **Final Formatting of Data**

Converted data into dataframes, vectors, etc to perform operations.

# Project Requirements

#### Do the analysis as in Module3 for at least one categorical variable and at least one numerical variable. Show appropriate plots for your data.

* Gender (categorical)
* Age (categorical)
* Purchase (numerical)

#### Do the analysis as in Module3 for at least one set of two or more variables. Show appropriate plots for your data.

* Gender, Product\_Category\_1, Product\_Category\_2, Product\_Category\_3 (4 Variables

#### Pick one variable with numerical data and examine the distribution of the data.

* Year in Current City

#### Draw various random samples of the data and show the applicability of the Central Limit Theorem for this variable.

* Purchase
* 5 Samples
* Size = 0.5%, 1%, 5%, 30%, 75% of total no of purchases

#### Show how various sampling methods can be used on your data. What are your conclusions if these samples are used instead of the whole dataset.

* Simple Random Sampling
* With Replacement
* Without Replacement
* Systematic Sampling
* Equal Probability
* Unequal Probability
* Stratified Sampling
* Cluster Sampling

#### Implementation of any feature(s) not mentioned in the specification

* String Function
* Rename string
* Tibble
* Average year a shopper stayed in the current city
* Average Purchase made in each City Category on the basis of Stay in the Current City
* Replacing NAs with 0s

# R-Packages Used

### UsingR

Using R for Introductory Statistics

### sampling

Functions for drawing and calibrating samples

### stringr

There are four main families of functions in stringr:

* Character manipulation: these functions allow you to manipulate individual characters within the strings in character vectors.
* Whitespace tools to add, remove, and manipulate whitespace.
* Locale sensitive operations whose operations will vary from locale to locale.
* Pattern matching functions. These recognise four engines of pattern description. The most common is regular expressions, but there are three other tools.

### tidyverse

The 'tidyverse' is a set of packages that work in harmony because they share common data representations and 'API' design.

Using it for TIBBLE in project.

### stats

For R statistical functions

### prob

A framework for performing elementary probability calculations on finite sample spaces, which may be represented by data frames or lists.

### dbplyr

This implements the data table back-end for 'dplyr' so that you can seamlessly use data table and 'dplyr' together.

### dtplyr

This implements the data table back-end for 'dplyr' so that you can seamlessly use data table and 'dplyr' together.

# Executed R Code [Attached: Html Output File ]

> original.data <- read.csv(file='E:/Practice Projects/R & Analytics Foundation/CS544Final\_totla/BlackFriday.csv', header=TRUE, sep=",")

>

> library(UsingR)

> library(sampling)

> library(stringr) # for STRING operations

> library(tidyverse) # to work with TIBBLE

> library(stats)

> library(prob)

>

> # DATA

> print("Black Friday Sales Data")

[1] "Black Friday Sales Data"

> head(original.data)

User\_ID Product\_ID Gender Age Occupation City\_Category

1 1000001 P00069042 F 0-17 10 A

2 1000001 P00248942 F 0-17 10 A

3 1000001 P00087842 F 0-17 10 A

4 1000001 P00085442 F 0-17 10 A

5 1000002 P00285442 M 55+ 16 C

6 1000003 P00193542 M 26-35 15 A

Stay\_In\_Current\_City\_Years Marital\_Status Product\_Category\_1 Product\_Category\_2

1 2 0 3 NA

2 2 0 1 6

3 2 0 12 NA

4 2 0 12 14

5 4+ 0 8 NA

6 3 0 1 2

Product\_Category\_3 Purchase

1 NA 8370

2 14 15200

3 NA 1422

4 NA 1057

5 NA 7969

6 NA 15227

>

> # COLUMNS NAME in data

> print("Columns we have in our data")

[1] "Columns we have in our data"

> names(original.data)

[1] "User\_ID" "Product\_ID"

[3] "Gender" "Age"

[5] "Occupation" "City\_Category"

[7] "Stay\_In\_Current\_City\_Years" "Marital\_Status"

[9] "Product\_Category\_1" "Product\_Category\_2"

[11] "Product\_Category\_3" "Purchase"

>

> attach(original.data)

The following objects are masked from original.data (pos = 3):

Age, City\_Category, Gender, Marital\_Status, Occupation,

Product\_Category\_1, Product\_Category\_2, Product\_Category\_3,

Product\_ID, Purchase, Stay\_In\_Current\_City\_Years, User\_ID

The following objects are masked from original.data (pos = 4):

Age, City\_Category, Gender, Marital\_Status, Occupation,

Product\_Category\_1, Product\_Category\_2, Product\_Category\_3,

Product\_ID, Purchase, Stay\_In\_Current\_City\_Years, User\_ID

The following objects are masked from original.data (pos = 5):

Age, City\_Category, Gender, Marital\_Status, Occupation,

Product\_Category\_1, Product\_Category\_2, Product\_Category\_3,

Product\_ID, Purchase, Stay\_In\_Current\_City\_Years, User\_ID

The following objects are masked from original.data (pos = 6):

Age, City\_Category, Gender, Marital\_Status, Occupation,

Product\_Category\_1, Product\_Category\_2, Product\_Category\_3,

Product\_ID, Purchase, Stay\_In\_Current\_City\_Years, User\_ID

The following objects are masked from original.data (pos = 7):

Age, City\_Category, Gender, Marital\_Status, Occupation,

Product\_Category\_1, Product\_Category\_2, Product\_Category\_3,

Product\_ID, Purchase, Stay\_In\_Current\_City\_Years, User\_ID

The following objects are masked from original.data (pos = 9):

Age, City\_Category, Gender, Marital\_Status, Occupation,

Product\_Category\_1, Product\_Category\_2, Product\_Category\_3,

Product\_ID, Purchase, Stay\_In\_Current\_City\_Years, User\_ID

>

> # CREATING DATA FOR ANALYSIS, PRESERVING ORIGINAL DATA

> # DATA GATHERING and CLEANING

> user.id = User\_ID

> product.id = Product\_ID

> gender = Gender

> age.range = Age

> occupation = Occupation

> city = City\_Category

>

> # convert data into numeric to perform calculations

> years.in.current.city = as.numeric(Stay\_In\_Current\_City\_Years)

> marital.status = Marital\_Status

> product.category.1 = Product\_Category\_1

>

> print("Data Cleaning")

[1] "Data Cleaning"

> print("There are NAs in product category 2 & 3, to do operations we convert NAs to 0 so that opertaions can be performed")

[1] "There are NAs in product category 2 & 3, to do operations we convert NAs to 0 so that opertaions can be performed"

> # convert NA to 0 to perform calculations

> product.category.2 = Product\_Category\_2

> product.category.2[which(is.na(product.category.2))] = 0

>

> # convert NA to 0 to perform calculations

> product.category.3 = Product\_Category\_3

> product.category.3[which(is.na(product.category.3))] = 0

>

> purchase = Purchase

>

> #making dataframe of data

> blackfridaysales = data.frame(user.id, product.id, gender, occupation, city, years.in.current.city, marital.status, product.category.1, product.category.2, product.category.3, purchase)

>

> # EXPLORING THE DATA

>

> # table() will summarize data

> head(table(user.id))

user.id

1000001 1000002 1000003 1000004 1000005 1000006

34 76 29 13 106 46

> print("User Id: Not Unique, maps person to the particular purchase")

[1] "User Id: Not Unique, maps person to the particular purchase"

>

> head(table(product.id))

product.id

P00000142 P00000242 P00000342 P00000442 P00000542 P00000642

1130 371 238 92 146 512

> print("Product Id: Not Unique, tells how many purchases are made for a product")

[1] "Product Id: Not Unique, tells how many purchases are made for a product"

>

> table(gender)

gender

F M

132197 405380

> print("Gender have only two variables: F M")

[1] "Gender have only two variables: F M"

>

> table(age.range)

age.range

0-17 18-25 26-35 36-45 46-50 51-55 55+

14707 97634 214690 107499 44526 37618 20903

> print("Age is divided into 7 ranges, Here Age is Categorical Variable")

[1] "Age is divided into 7 ranges, Here Age is Categorical Variable"

>

> table(occupation)

occupation

0 1 2 3 4 5 6 7 8 9 10 11 12

68120 45971 25845 17366 70862 11985 19822 57806 1524 6153 12623 11338 30423

13 14 15 16 17 18 19 20

7548 26712 11812 24790 39090 6525 8352 32910

> print("There are 21 different occupation ranging from 0-21")

[1] "There are 21 different occupation ranging from 0-21"

>

> table(city)

city

A B C

144638 226493 166446

> print("Cities in which customers have lived is categorized into three categories: A B C")

[1] "Cities in which customers have lived is categorized into three categories: A B C"

>

> table(years.in.current.city)

years.in.current.city

1 2 3 4 5

72725 189192 99459 93312 82889

> print("People have lived in the current city for 0-5 years. Here 5 could mean atleast 5 years")

[1] "People have lived in the current city for 0-5 years. Here 5 could mean atleast 5 years"

>

> table(marital.status)

marital.status

0 1

317817 219760

> print("People have their marriage status marked as either 0 or 1")

[1] "People have their marriage status marked as either 0 or 1"

>

> table(product.category.1)

product.category.1

1 2 3 4 5 6 7 8 9 10 11

138353 23499 19849 11567 148592 20164 3668 112132 404 5032 23960

12 13 14 15 16 17 18

3875 5440 1500 6203 9697 567 3075

> print("Product Category 1: Ranges form 1-18")

[1] "Product Category 1: Ranges form 1-18"

>

> table(product.category.2)

product.category.2

0 2 3 4 5 6 7 8 9 10 11

166986 48481 2835 25225 25874 16251 615 63058 5591 2991 13945

12 13 14 15 16 17 18

5419 10369 54158 37317 42602 13130 2730

> print("Product Category 2: Ranges form 2-18")

[1] "Product Category 2: Ranges form 2-18"

> print("0 = NA for the product category, substituted this to make data clean")

[1] "0 = NA for the product category, substituted this to make data clean"

>

>

> table(product.category.3)

product.category.3

0 3 4 5 6 8 9 10 11 12 13

373299 600 1840 16380 4818 12384 11414 1698 1773 9094 5385

14 15 16 17 18

18121 27611 32148 16449 4563

> print("Product Category 3: Ranges form 3-18")

[1] "Product Category 3: Ranges form 3-18"

> print("0 = NA for the product category")

[1] "0 = NA for the product category"

>

> head(table(purchase),n=10)

purchase

185 186 187 188 189 190 191 192 193 194

4 4 4 6 2 4 3 4 3 1

> print("It is the amount people spended in $ for purchases. Not unique.")

[1] "It is the amount people spended in $ for purchases. Not unique."

>

>

|  |
| --- |
| > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # REQUIREMENT 1  > # Do the analysis as in Module3 for at least one categorical variable and at least one numerical variable. Show appropriate plots for your data.  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  >  >  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # ANALYSIS ON 1ST CATEGORICAL VARIABLE: Gender  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |
| |  | | --- | | > | |

>

>

> # keeping copy of data in a temporary variable

> temporary = gender

>

> gender = table(gender)

>

> # making labels for piechart

> gender.labels = c(" Female", " Male")

> gender.percent = round(gender/sum(gender)\*100)

> gender.labels = paste(gender.labels, gender.percent)

> gender.labels = paste(gender.labels, "%", sep="")

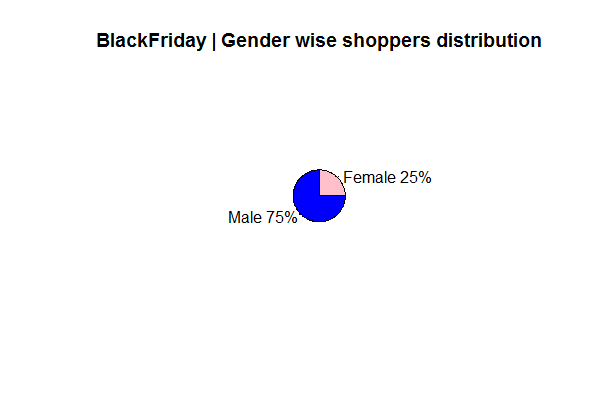
>

> print("Ploting a pie chart to check which gender shop more")

[1] "Ploting a pie chart to check which gender shop more"

> pie(gender, labels = gender.labels, col=c("Pink","Blue"), main="BlackFriday | Gender wise shoppers distribution")

>



> print("RESULT : We can conclude that Male(75%) shop more than Female(25%) by the pie chart")

[1] "RESULT : We can conclude that Male(75%) shop more than Female(25%) by the pie chart"

>

> # making the variable same as before

> gender = temporary

>

>

> # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>

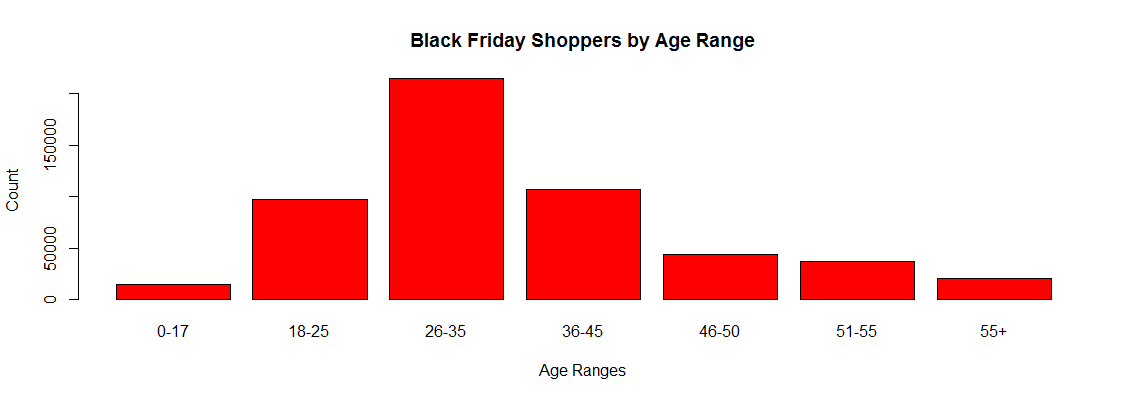
> # ANALYSIS ON CATEGORICAL VARIABLE: Age

> # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>

> barplot(table(age.range), col="Red", main="Black Friday Shoppers by Age Range", xlab="Age Ranges", ylab="Count")

>



> print("RESULT : People withing range of 26-35 shopped most")

[1] "RESULT : People withing range of 26-35 shopped most"

> print("While people in age-range 0-17 or 55+ shopped least and almost none compared to 26-35")

[1] "While people in age-range 0-17 or 55+ shopped least and almost none compared to 26-35"

> print("Also, overall people within age range 18-45 are the group which makes maximum population of shooping")

[1] "Also, overall people within age range 18-45 are the group which makes maximum population of shooping"

>

>

> # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>

> # ANALYSIS ON NUMERICAL VARIABLE: Purchase

> # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>

> print("Analysing 'Purchase' by barplot, histogram, boxplot ")

[1] "Analysing 'Purchase' by barplot, histogram, boxplot "

>

> # average purchase

> cat("Average dollars shoppers spent = ", mean(purchase))

Average dollars shoppers spent = 9334>

> # purchase range

> range.purchase = range(purchase)

> cat("Range of amount shoppers spent = ", range.purchase)

Range of amount shoppers spent = 185 23961

> # BARPLOT

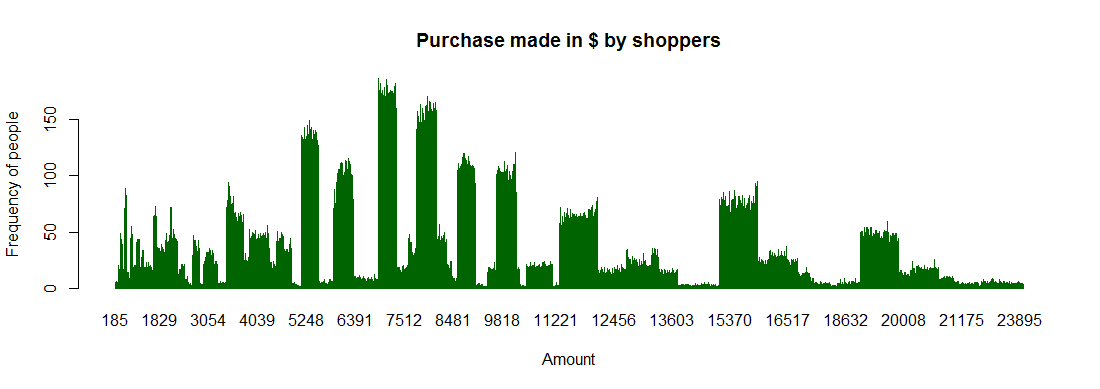
>

> print("We can see an overview how much amount is spent by people")

[1] "We can see an overview how much amount is spent by people"

> barplot(table(purchase), border = c("darkgreen"), main="Purchase made in $ by shoppers", xlab = "Amount", ylab="Frequency of people")

>

****

> print("RESULT : Hardly a shopper spend above $19000")

[1] "RESULT : Hardly a shopper spend above $19000"

> print("Shoppers mostly spent an amount of approximately 6800 or 8700 as they got highest peak in barplot")

[1] "Shoppers mostly spent an amount of approximately 6800 or 8700 as they got highest peak in barplot"

>

>

> # HISTOGRAMS

>

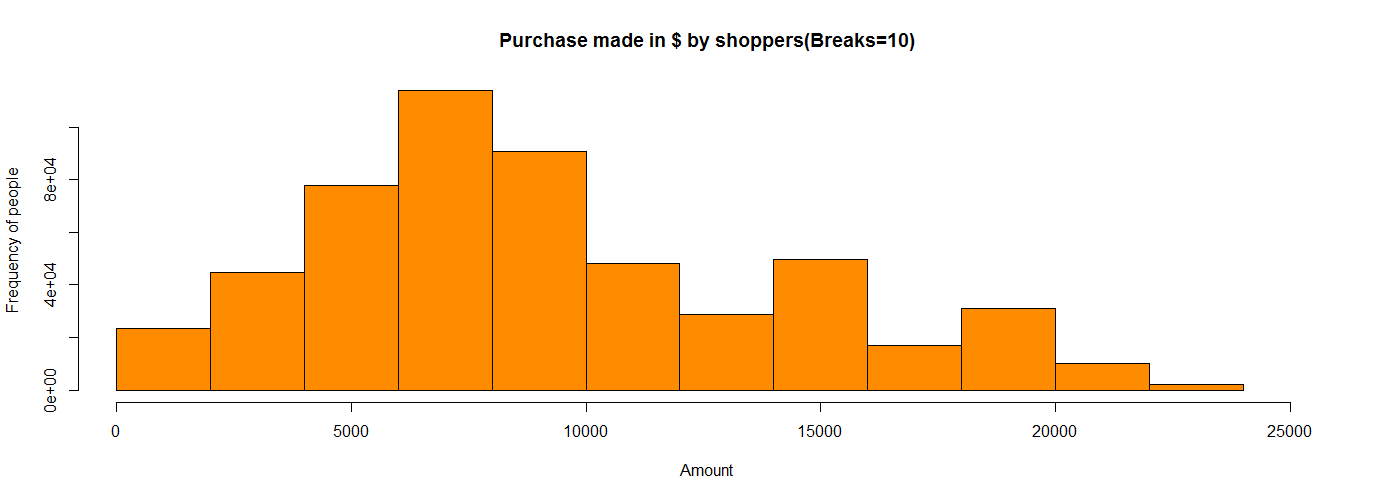
> print("We plot histogram with increasing breaks to analyse data")

[1] "We plot histogram with increasing breaks to analyse data"

>

> # breaks = 10

> hist(purchase,breaks=10, xlim=c(185,25000),col="darkorange", main="Purchase made in $ by shoppers(Breaks=10)", xlab="Amount", ylab="Frequency of people")

****

> print("Break=10 We see max data lies between 5000-10000, increase break to 20")

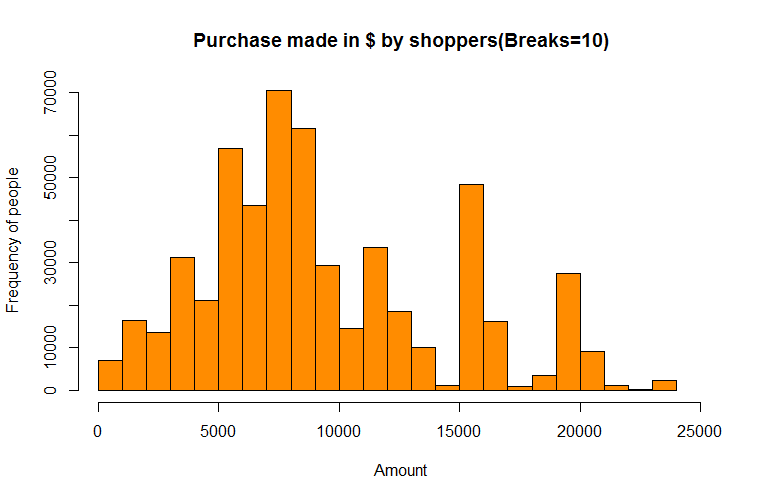
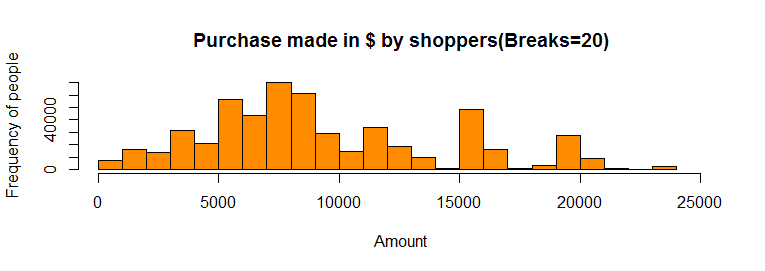
[1] "Break=10 We see max data lies between 5000-10000, increase break to 20"

>

> # breaks = 20

> hist(purchase,breaks=20, xlim=c(185,25000),col="darkorange", main="Purchase made in $ by shoppers(Breaks=20)", xlab="Amount",

ylab="Frequency of people")

****

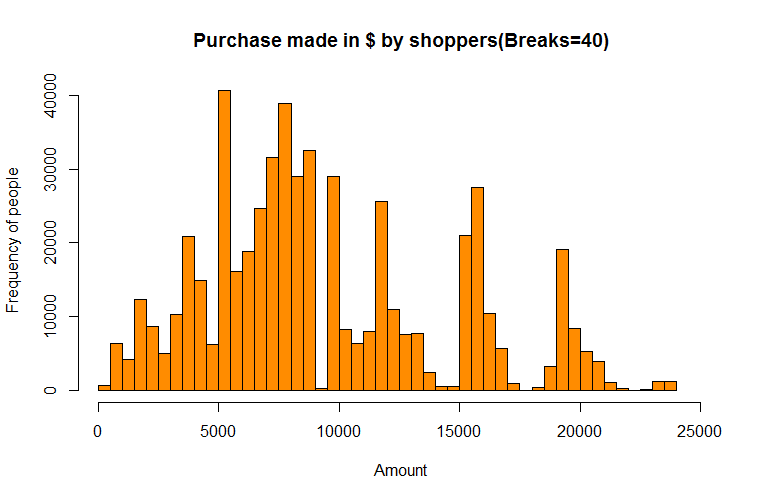
> print("We can see there are some figures which are not at all spent and good amount is spent near 15000 and b/w 5000-10000, increasing break=40")

[1] "We can see there are some figures which are not at all spent and good amount is spent near 15000 and b/w 5000-10000, increasing break=40"

>

> # breaks = 40

> hist(purchase,breaks=40, xlim=c(185,25000),col="darkorange", main="Purchase made in $ by shoppers(Breaks=40)", xlab="Amount", ylab="Frequency of people")

****

> print("We can now clearly see how much figures people spent")

[1] "We can now clearly see how much figures people spent"

>

> print("RESULT : If a shopper is coming to black friday sale there are maximum chances, he would be spending on an average at least $5000")

[1] "RESULT : If a shopper is coming to black friday sale there are maximum chances, he would be spending on an average at least $5000"

> print("Maximum shoppers populayion lie across $5000 mark")

[1] "Maximum shoppers populayion lie across $5000 mark"

> print("Coincidence & Interesting to see a 0 frequency near 10,000, and mid of 15000-20000")

[1] "Coincidence & Interesting to see a 0 frequency near 10,000, and mid of 15000-20000"

> print("We may consider that people didn't spent in $9000 or $17000(avg of 15K & 20K) in sales")

[1] "We may consider that people didn't spent in $9000 or $17000(avg of 15K & 20K) in sales"

>

>

> # SUMMARY

>

> print("Summary for figures people spent")

[1] "Summary for figures people spent"

> summary(purchase)

Min. 1st Qu. Median Mean 3rd Qu. Max.

185 5866 8062 9334 12073 23961

>

> print("Min - Minimum Purchase Value")

[1] "Min - Minimum Purchase Value"

>

> print("1st Quantle - Middle number between the smallest number and the median of the data set")

[1] "1st Quantle - Middle number between the smallest number and the median of the data set"

> print("It tells the mid value of min and median purcahse value")

[1] "It tells the mid value of min and median purcahse value"

>

> print("Median - Middle value of all the values")

[1] "Median - Middle value of all the values"

>

> print("2nd Quantle - Middle number between the median and the largest number of the data set")

[1] "2nd Quantle - Middle number between the median and the largest number of the data set"

> print("It tells the mid value of min and median purcahse value")

[1] "It tells the mid value of min and median purcahse value"

>

> print("Max - Maximum Purchase Value")

[1] "Max - Maximum Purchase Value"

>

>

> # BOXPLOT

>

> print("Plotting a boxplot")

[1] "Plotting a boxplot"

> print("In box plot we can clearly see distribution of the amounts spended in black friday sales")

[1] "In box plot we can clearly see distribution of the amounts spended in black friday sales"

>

> f = fivenum(purchase)

> oulier = c(f[2]-1.5\*(f[4]-f[2]) , f[4]+1.5\*(f[4]-f[2]))

> boxplot(f,horizontal = TRUE, xaxt="n", xlab="Amount", col="yellow", main="Purchase made in $ by shoppers")

> axis(side = 1, at = f, labels=TRUE)

> text(f,srt=90, rep(1.2,5), adj=0,labels=c("Min", "Lower Hinge", "Mean","Upper Hinge", "Max"))

>

****

> print("RESULT : We can consider an average shopper will spend $5866-$12073 in black friday sales")

[1] "RESULT : We can consider an average shopper will spend $5866-$12073 in black friday sales"

|  |
| --- |
| > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # REQUIREMENT 2  > # Do the analysis as in Module3 for at least one set of two or more variables. Show appropriate plots for your data|
|  |
| |  | | --- | | > | |

> print("Analysing Multivariate Data")

[1] "Analysing Multivariate Data"

>

> # CHECK WHICH GENDER TOOK WHICH PRODUCT CATEGORY

> # THERE ARE 3 CATEGORIES

>

> # m male

> # f female

> # pc.1 product category 1

> # pc.2 product category 2

> # pc.3 product category 3

>

> # converting data to make data frame

>

> g = as.vector(as.character(gender))

> pc1 = as.vector(as.numeric(as.character(product.category.1)))

> pc2 = as.vector(as.numeric(as.character(product.category.2)))

> pc3 = as.vector(as.numeric(as.character(product.category.3)))

>

> # Checking all have equal rows to convert to data frame and confirming after cleaning the data

> # > NROW(x = pc2)

> # [1] 537577

> # > NROW(x = pc1)

> # [1] 537577

> # > NROW(x = pc3)

> # [1] 537577

> # > NROW(x = g)

> # [1] 537577

>

>

> # created a data frame for variables/data required (gender, product category 1,2,3)

>

> temp.data = data.frame(gender = g, product.category.1 = pc1, product.category.2 = pc2, product.category.3 = pc3)

> head(temp.data)

gender product.category.1 product.category.2 product.category.3

1 F 3 0 0

2 F 1 6 14

3 F 12 0 0

4 F 12 14 0

5 M 8 0 0

6 M 1 2 0

> g = temp.data$gender

> pc1 = temp.data$product.category.1

> pc2 = temp.data$product.category.2

> pc3 = temp.data$product.category.3

>

> m.pc.1 = sum(pc1[which( g == 'M')])

> m.pc.2 = sum(pc2[which( g == 'M')])

> m.pc.3 = sum(pc3[which( g == 'M')])

>

> f.pc.1 = sum(pc1[which( g == 'F')])

> f.pc.2 = sum(pc2[which( g == 'F')])

> f.pc.3 = sum(pc3[which( g == 'F')])

>

>

>

> # CREATING SUMMARIZED DATA/TABLE

>

> # table creation

> bidata = rbind(c(m.pc.1,f.pc.1),c(m.pc.2,f.pc.2),c(m.pc.3,f.pc.3))

>

>

> # naming columns and rows

>

> gender.names = c("male","female")

> product.category.names = c("product category 1","product category 2","product category 3")

> colnames(bidata) = gender.names

> rownames(bidata) = product.category.names

>

> dimnames(bidata) = list(ProductCategory = product.category.names,Gender = gender.names)

>

> print("Our summarized data for analysis is")

[1] "Our summarized data for analysis is"

> bidata

Gender

ProductCategory male female

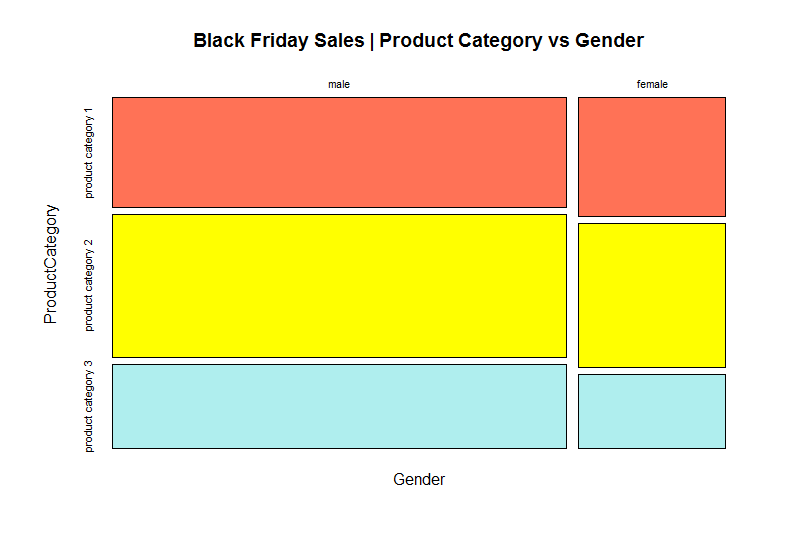
product category 1 2107063 739701

product category 2 2748194 899216

product category 3 1621487 459889

> mosaicplot(t(bidata),col=c("coral1","yellow","paleturquoise"),main="Black Friday Sales | Product Category vs Gender")

>

****

> print("RESULT :")

[1] "RESULT :"

> print("Overall there are more male shoppers cleary from plot")

[1] "Overall there are more male shoppers cleary from plot"

> print("Product Category 2 being sold most")

[1] "Product Category 2 being sold most"

> print("Product category 3 sales are almost half of product category 2 in case of female shoppers")

[1] "Product category 3 sales are almost half of product category 2 in case of female shoppers"

>

> print("Total sales of every product category")

[1] "Total sales of every product category"

> margin.table(bidata,1)

ProductCategory

product category 1 product category 2 product category 3

2846764 3647410 2081376

>

> print("Total sales by gender")

[1] "Total sales by gender"

> margin.table(bidata,2)

Gender

male female

6476744 2098806

>

>

> # RESCALING DATA

>

> print("Rescaling data for better understanding")

[1] "Rescaling data for better understanding"

> rescale.bidata = round(bidata/100000)

>

> print("Rescaled Data")

[1] "Rescaled Data"

> rescale.bidata

Gender

ProductCategory male female

product category 1 21 7

product category 2 27 9

product category 3 16 5

>

> print("Original Data")

[1] "Original Data"

> bidata

Gender

ProductCategory male female

product category 1 2107063 739701

product category 2 2748194 899216

product category 3 1621487 459889

>

> print("Every value now onwards for bivariate data represent 100,000s (Million[M]) value")

[1] "Every value now onwards for bivariate data represent 100,000s (Million[M]) value"

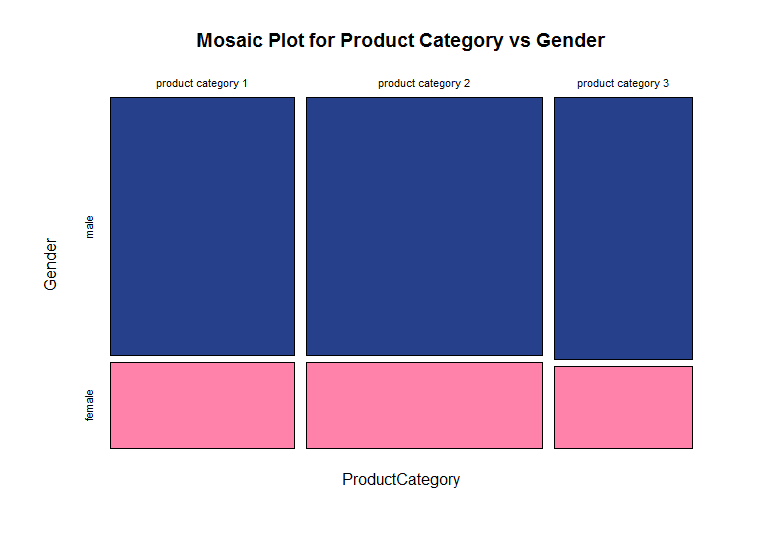
>

>

> # MOSAICPLOT

> mosaicplot(rescale.bidata,col=c("royalblue4","palevioletred1"),main="Mosaic Plot for Product Category vs Gender")

>

****

> print("Total Sales (in Millions(M))")

[1] "Total Sales (in Millions(M))"

> addmargins(rescale.bidata)

Gender

ProductCategory male female Sum

product category 1 21 7 28

product category 2 27 9 36

product category 3 16 5 21

Sum 64 21 85

> print("RESULT : Approximately,")

[1] "RESULT : Approximately,"

> print("Total there is sales of 85M products")

[1] "Total there is sales of 85M products"

> print("out of which product category 1,2,3 have 28M, 36M, 21M sales respectively")

[1] "out of which product category 1,2,3 have 28M, 36M, 21M sales respectively"

> print("and males brought 64M products while female 21M")

[1] "and males brought 64M products while female 21M"

>

> print("Percentage wise sales")

[1] "Percentage wise sales"

> x = prop.table(rescale.bidata)\*100

> round(x)

Gender

ProductCategory male female

product category 1 25 8

product category 2 32 11

product category 3 19 6

>

> print("Percentage wise sales for gender")

[1] "Percentage wise sales for gender"

> x = prop.table(rescale.bidata,1)\*100

> round(x)

Gender

ProductCategory male female

product category 1 75 25

product category 2 75 25

product category 3 76 24

> print("Notable thing is values of Male are around 75% and Female 25% we analysed earlier too")

[1] "Notable thing is values of Male are around 75% and Female 25% we analysed earlier too"

>

>

> print("Percentage wise sales for category")

[1] "Percentage wise sales for category"

> x = prop.table(rescale.bidata,2)\*100

> round(x)

Gender

ProductCategory male female

product category 1 33 33

product category 2 42 43

product category 3 25 24

>

> print("RESULT : Each gender have almost same contribution in every category")

[1] "RESULT : Each gender have almost same contribution in every category"

>

>

>

>

|  |
| --- |
| > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # REQUIREMENT 3  > # Pick one variable with numerical data and examine the distribution of the data.  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  >  >  >  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # ANALYSIS ON ONE NUMERICAL VARIABLE: Year in Current City  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |
| |  | | --- | |  | |

> temporary = years.in.current.city # keeping copy of data in a temporary variable

> years.in.current.city = table(years.in.current.city)

>

> # GEOMETRIC DITRIBUTION

>

> df.years.in.current.city = data.frame(years.in.current.city)

> df.years.in.current.city$probability = df.years.in.current.city$Freq/sum(df.years.in.current.city$Freq)

> options(digits = 2)

>

> print("Geometric Distribution")

[1] "Geometric Distribution"

> print("Probability that the person I picked have stayed 5 years in current city")

[1] "Probability that the person I picked have stayed 5 years in current city"

>

> print("First person")

[1] "First person"

> dgeom(0, prob=df.years.in.current.city$probability[5])\*100

[1] 20

>

> print("Second person")

[1] "Second person"

> dgeom(1, prob=df.years.in.current.city$probability[5])\*100

[1] 16

>

> # checking for 10 persons

> pmf.10 = dgeom(0:9, prob=df.years.in.current.city$probability[5])

> print("till 10 persons")

[1] "till 10 persons"

> person.stay.5.year = data.frame(

+ person.count = seq(1:10),

+ probability = pmf.10,

+ percentage = pmf.10 \* 100

+ )

> person.stay.5.year

person.count probability percentage

1 1 0.200 20.0

2 2 0.160 16.0

3 3 0.128 12.8

4 4 0.102 10.2

5 5 0.082 8.2

6 6 0.066 6.6

7 7 0.052 5.2

8 8 0.042 4.2

9 9 0.034 3.4

10 10 0.027 2.7

> # plotting graph

> plot(0:9,pmf.10,type="h",col="green",main="Choosed a person who spent 5 years in current city out of 10 trials",xlab="Person Count", ylab="PMF/ Probability",pch=16)

> abline(h=0)

>

> ****

>

> # BARPLOT

>

> years.in.current.city = round(years.in.current.city/1000)

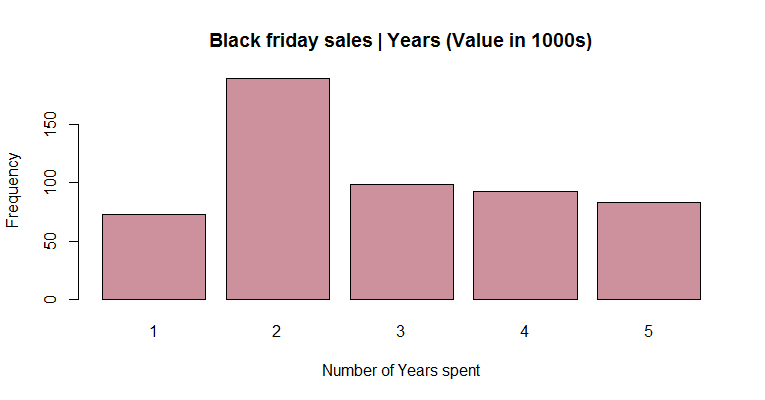
>

> print("We can see an overview which years.in.current.city people spent most")

[1] "We can see an overview which years.in.current.city people spent most"

> barplot(years.in.current.city,col="pink3",xlab="Number of Years spent",ylab="Frequency",main="Black friday sales | Years (Value in 1000s)")

>

****

> print("RESULT : Most of the people have spent at least 2 years in the city where the survey took place")

[1] "RESULT : Most of the people have spent at least 2 years in the city where the survey took place"

>

>

> # PIECHART

>

> years.in.current.city.labels = c("1 Year","2 Years","3 Years","4 Years","5 Years")

> years.in.current.city.percent = round(years.in.current.city/sum(years.in.current.city)\*100)

> years.in.current.city.labels = paste(years.in.current.city.labels, years.in.current.city.percent)

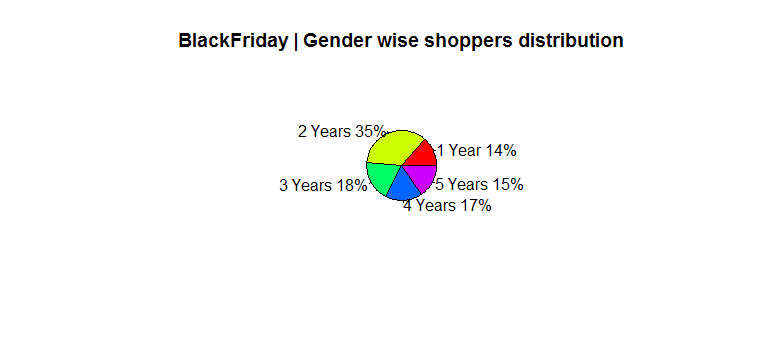
> years.in.current.city.labels = paste(years.in.current.city.labels, "%", sep="")

>

> print("Ploting a pie chart for years")

[1] "Ploting a pie chart for years"

> pie(years.in.current.city, labels = years.in.current.city.labels, col=rainbow(5), main="BlackFriday | Gender wise shoppers distribution")

****

> print("RESULT : From pie chart we can say that 50% of popuation have spend atmost 2 years and 50% more than 2 years")

[1] "RESULT : From pie chart we can say that 50% of popuation have spend atmost 2 years and 50% more than 2 years"

>

> # copying original data in variable

> occupation = temporary

|  |
| --- |
| >  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # REQUIREMENT 4  > # Draw various random samples of the data and show the applicability of the Central Limit Theorem for this variable.  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  >  >  >  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # CENTRAL LIMIT THEOREM  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  >  > print("Applying central limit theorem on Purchase Amount")  [1] "Applying central limit theorem on Purchase Amount"  >  > # total samples  > purchase.samples.count = nrow(table(purchase))  > cat("Total Samples: ", purchase.samples.count)  Total Samples: 17959>  > # bringing data back to original state  > temporary = purchase  >  > print("For easy calculation rescaling purchases")  [1] "For easy calculation rescaling purchases"  > purchase = round(purchase/1000)  >  > # calculating mean  > purchase.mean = mean(purchase)  > cat("Mean: ",purchase.mean)  Mean: 9.35>  > # calculating std. deviation  > purchase.sd = sd(purchase)  > cat("Standard Deviation: ",purchase.sd)  Standard Deviation: 4.99>  > # taking samples  >  > # SAMPLE 1  > print("Taking 0.5% of total sample")  [1] "Taking 0.5% of total sample"  > sample.size = round(purchase.samples.count\*0.005)  > cat("Sample size= ", sample.size)  Sample size= 90>  > sample1 = sample(purchase, size = sample.size, replace = TRUE)  > sample1.mean = mean(sample1)  > sample1.sd = sd(sample1)  >  > cat("Orginal Data Mean | Sample Mean = ", purchase.mean, " | ",sample1.mean)  Orginal Data Mean | Sample Mean = 9.35 | 9.38> cat("Orginal Data Std. Dev. | Sample Std. Dev. = ", purchase.sd, " | ",sample1.sd)  Orginal Data Std. Dev. | Sample Std. Dev. = 4.99 | 5.24>  > main = paste("0.5% of Shoppers | Sample size = ",sample.size)  > hist(sample1, prob = TRUE, breaks=15, xlim=c(0,60),col="darkorange", main = main)  >    >  > # SAMPLE 2  > print("Taking 1% of total sample")  [1] "Taking 1% of total sample"  > sample.size = round(purchase.samples.count\*0.01)  > cat("Sample size= ", sample.size)  Sample size= 180>  > sample2 = sample(purchase, size = sample.size, replace = TRUE)  > sample2.mean = mean(sample2)  > sample2.sd = sd(sample2)  >  > cat("Orginal Data Mean | Sample Mean = ", purchase.mean, " | ",sample2.mean)  Orginal Data Mean | Sample Mean = 9.35 | 9.73> cat("Orginal Data Std. Dev. | Sample Std. Dev. = ", purchase.sd, " | ",sample2.sd)  Orginal Data Std. Dev. | Sample Std. Dev. = 4.99 | 4.96>  > main = paste("1% of Shoppers | Sample size = ",sample.size)  > hist(sample2, prob = TRUE, breaks=15, xlim=c(0,60),col="darkorange", main = main)    >  > # SAMPLE 3  > print("Taking 5% of total sample")  [1] "Taking 5% of total sample"  > sample.size = round(purchase.samples.count\*0.05)  > cat("Sample size= ", sample.size)  Sample size= 898>  > sample3 = sample(purchase, size = sample.size, replace = TRUE)  > sample3.mean = mean(sample3)  > sample3.sd = sd(sample3)  >  > cat("Orginal Data Mean | Sample Mean = ", purchase.mean, " | ",sample3.mean)  Orginal Data Mean | Sample Mean = 9.35 | 9.2> cat("Orginal Data Std. Dev. | Sample Std. Dev. = ", purchase.sd, " | ",sample3.sd)  Orginal Data Std. Dev. | Sample Std. Dev. = 4.99 | 5.08>  > main = paste("5% of Shoppers | Sample size = ",sample.size)  > hist(sample3, prob = TRUE, breaks=15, xlim=c(0,60),col="darkorange", main = main)  >    >  > # SAMPLE 4  > print("Taking 30% of total sample")  [1] "Taking 30% of total sample"  > sample.size = round(purchase.samples.count\*0.3)  > cat("Sample size= ", sample.size)  Sample size= 5388>  > sample4 = sample(purchase, size = sample.size, replace = TRUE)  > sample4.mean = mean(sample4)  > sample4.sd = sd(sample4)  >  > cat("Orginal Data Mean | Sample Mean = ", purchase.mean, " | ",sample4.mean)  Orginal Data Mean | Sample Mean = 9.35 | 9.42> cat("Orginal Data Std. Dev. | Sample Std. Dev. = ", purchase.sd, " | ",sample4.sd)  Orginal Data Std. Dev. | Sample Std. Dev. = 4.99 | 5.05>  > main = paste("30% of Shoppers | Sample size = ",sample.size)  > hist(sample4, prob = TRUE, breaks=15, xlim=c(0,60),col="darkorange", main = main)  >  >  >  > # SAMPLE 5  > print("Taking 75% of total sample")  [1] "Taking 75% of total sample"  > sample.size = round(purchase.samples.count\*0.75)  > cat("Sample size= ", sample.size)  Sample size= 13469>  > sample5 = sample(purchase, size = sample.size, replace = TRUE)  > sample5.mean = mean(sample5)  > sample5.sd = sd(sample5)  >  > cat("Orginal Data Mean | Sample Mean = ", purchase.mean, " | ",sample5.mean)  Orginal Data Mean | Sample Mean = 9.35 | 9.36> cat("Orginal Data Std. Dev. | Sample Std. Dev. = ", purchase.sd, " | ",sample5.sd)  Orginal Data Std. Dev. | Sample Std. Dev. = 4.99 | 4.99>  > main = paste("75% of Shoppers | Sample size = ",sample.size)  > hist(sample5, prob = TRUE, breaks=15, xlim=c(0,60),col="darkorange", main = main)  >    > print("In all plots we saw the distribution is centered around original mean")  [1] "In all plots we saw the distribution is centered around original mean"  >  > options(digits = 3)  > print("Mean: Original, 0.5%, 1%, 5%, 30%, 75% of total no of purchases")  [1] "Mean: Original, 0.5%, 1%, 5%, 30%, 75% of total no of purchases"  > cat(purchase.mean, sample1.mean, sample2.mean, sample3.mean, sample4.mean, sample5.mean)  9.35 9.38 9.73 9.2 9.42 9.36>  > print("Std. Dev.: Original, 0.5%, 1%, 5%, 30%, 75% of total no of purchases")  [1] "Std. Dev.: Original, 0.5%, 1%, 5%, 30%, 75% of total no of purchases"  > cat(purchase.sd, sample1.sd, sample2.sd, sample3.sd, sample4.sd, sample5.sd)  4.99 5.235097 4.963104 5.082541 5.053998 4.987335>  > print("Therefore we can see that")  [1] "Therefore we can see that"  > print("If data is drawn from any distribution, then the distribution of the sample means has  + the shape of a normal distribution for a large sample. The mean of the sample mean distribution is equal to the  + mean of the parent data. The higher the sample size, the narrower the spread of the sample means.")  [1] "If data is drawn from any distribution, then the distribution of the sample means has\nthe shape of a normal distribution for a large sample. The mean of the sample mean distribution is equal to the\n mean of the parent data. The higher the sample size, the narrower the spread of the sample means."  >  > # copy the original data into variable again  > purchase = temporary  >  >  >  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # REQUIREMENT 5  > # Show how various sampling methods can be used on your data. What are your conclusions if these samples are used instead of the whole dataset.  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  >  >  >  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  > # SAMPLING : Stay in Current City  > # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  >  > # still using rescaled Purchase values  > sample.size = round(0.07 \* purchase.samples.count)#picking random sample size  > cat("Sample size = ", sample.size)  Sample size = 1257>  > # SIMPLE RANDOM SAMPLING  > print("SIMPLE RANDOM SAMPLING")  [1] "SIMPLE RANDOM SAMPLING"  >  > #with replacement  > print("With Replacement")  [1] "With Replacement"  > s = srswr(sample.size, nrow(blackfridaysales))  > rows = (1:nrow(blackfridaysales))[s!=0]  > rows = rep(rows,s[s!=0])  > sample.with.replace = blackfridaysales[rows,]  > print("For Example")  [1] "For Example"  > print("Years in Current City Original: ")  [1] "Years in Current City Original: "  > table(blackfridaysales$years.in.current.city)  1 2 3 4 5  72725 189192 99459 93312 82889  > print("Years in Current City Sampled Data")  [1] "Years in Current City Sampled Data"  > table(sample.with.replace$years.in.current.city)  1 2 3 4 5  159 462 240 195 201  >  > # Using Sample Data vs Original Data  > print("Sample Data")  [1] "Sample Data"  > print("Gender Count")  [1] "Gender Count"  > table(sample.with.replace$gender)  F M  313 944  > print("Percentage wise Gender Distribution")  [1] "Percentage wise Gender Distribution"  > prop.table(table(sample.with.replace$gender))\*100  F M  24.9 75.1  >  > print("Original Data")  [1] "Original Data"  > print("Percentage wise Gender Distribution")  [1] "Percentage wise Gender Distribution"  > prop.table(table(blackfridaysales$gender))\*100  F M  24.6 75.4  >  > print("RESULT : As we can see that distribution has changed in sample data but not much deflection wrt original")  [1] "RESULT : As we can see that distribution has changed in sample data but not much deflection wrt original"  >  >  > # without replacement  > print("Without Replacement")  [1] "Without Replacement"  > s = srswor(sample.size, nrow(blackfridaysales))  > rows = (1:nrow(blackfridaysales))[s!=0]  > rows = rep(rows,s[s!=0])  > sample.without.replace = blackfridaysales[rows,]  > print("For Example")  [1] "For Example"  > print("Years in Current City Original: ")  [1] "Years in Current City Original: "  > table(blackfridaysales$years.in.current.city)  1 2 3 4 5  72725 189192 99459 93312 82889  > print("Years in Current City Sampled Data")  [1] "Years in Current City Sampled Data"  > table(sample.without.replace$years.in.current.city)  1 2 3 4 5  182 443 244 199 189 |
|  |
| |  | | --- | |  | |

> # Using Sample Data vs Original Data

> print("Sample Data")

[1] "Sample Data"

> print("Gender Count")

[1] "Gender Count"

> table(sample.without.replace$gender)

F M

325 932

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(sample.without.replace$gender))\*100

F M

25.9 74.1

>

> print("Original Data")

[1] "Original Data"

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(blackfridaysales$gender))\*100

F M

24.6 75.4

>

> print("RESULT : As we can see that distribution has changed in sample data but not much deflection")

[1] "RESULT : As we can see that distribution has changed in sample data but not much deflection"

>

>

>

> # SYSTEMATIC SAMPLING

> print("SYSTEMATIC SAMPLING")

[1] "SYSTEMATIC SAMPLING"

>

> # Equal Probability

> print("Considering equal probabilities of every data")

[1] "Considering equal probabilities of every data"

>

> N = nrow(blackfridaysales)

> n = sample.size

> k = ceiling(N/n)

> r = sample(k,1)

> s = seq(r, by=k, length=n)

> sample.systematic = blackfridaysales[s,]

> print("For Example")

[1] "For Example"

> print("Years in Current City Original: ")

[1] "Years in Current City Original: "

> table(blackfridaysales$years.in.current.city)

1 2 3 4 5

72725 189192 99459 93312 82889

> print("Years in Current City Sampled Data")

[1] "Years in Current City Sampled Data"

> table(sample.systematic$years.in.current.city)

1 2 3 4 5

146 479 232 213 186

>

>

> # Using Sample Data vs Original Data

> print("Sample Data")

[1] "Sample Data"

> print("Gender Count")

[1] "Gender Count"

> table(sample.systematic$gender)

F M

319 937

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(sample.systematic$gender))\*100

F M

25.4 74.6

>

> print("Original Data")

[1] "Original Data"

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(blackfridaysales$gender))\*100

F M

24.6 75.4

>

> print("RESULT : As we can see that distribution has changed in sample data but not much deflection")

[1] "RESULT : As we can see that distribution has changed in sample data but not much deflection"

>

>

>

>

> # UNEQUAL PROBABILITIES

>

> print("Considering unequal probabilities for data")

[1] "Considering unequal probabilities for data"

>

> pik = inclusionprobabilities(blackfridaysales$years.in.current.city, sample.size)

> s= UPsystematic(pik)

> sample.systematic.unequal.prob = blackfridaysales[s!=0,]

> print("For Example")

[1] "For Example"

> print("Years in Current City Original: ")

[1] "Years in Current City Original: "

> table(blackfridaysales$years.in.current.city)

1 2 3 4 5

72725 189192 99459 93312 82889

> print("Years in Current City Sampled Data")

[1] "Years in Current City Sampled Data"

> table(sample.systematic.unequal.prob$years.in.current.city)

1 2 3 4 5

55 301 274 282 345

>

>

> # Using Sample Data vs Original Data

> print("Sample Data")

[1] "Sample Data"

> print("Gender Count")

[1] "Gender Count"

> table(sample.systematic.unequal.prob$gender)

F M

290 967

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(sample.systematic.unequal.prob$gender))\*100

F M

23.1 76.9

>

> print("Original Data")

[1] "Original Data"

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(blackfridaysales$gender))\*100

F M

24.6 75.4

>

> print("RESULT : As we can see that distribution has changed in sample data but not much deflection wrt original")

[1] "RESULT : As we can see that distribution has changed in sample data but not much deflection wrt original"

>

>

>

> # STRATIFIED SAMPLING

>

> temporary = blackfridaysales

> temporary.size = sample.size

>

> print("STRATIFIED SAMPLING")

[1] "STRATIFIED SAMPLING"

>

> sample.size = 10 #changin the sample size for this sampling to better understand

> cat("Sample size for startified sampling = ",sample.size)

Sample size for startified sampling = 10

> # ordering the data

> blackfridaysales = blackfridaysales[order(blackfridaysales$years.in.current.city),]

>

> freq <- table(blackfridaysales$years.in.current.city)

> size <- sample.size \* freq / sum(freq)

>

> st1 = strata(blackfridaysales, size = size , stratanames = c("years.in.current.city"), method = "srswr", description = TRUE)

Stratum 1

Population total and number of selected units: 72725 1.35

Stratum 2

Population total and number of selected units: 189192 3.52

Stratum 3

Population total and number of selected units: 99459 1.85

Stratum 4

Population total and number of selected units: 93312 1.74

Stratum 5

Population total and number of selected units: 82889 1.54

Number of strata 5

Total number of selected units 10

> sample = getdata(blackfridaysales,st1)

> sample

user.id product.id gender occupation city marital.status product.category.1

392408 1000392 P00182242 M 7 A 0 1

307820 1005440 P00053842 F 2 B 1 4

472137 1000757 P00076042 M 12 A 0 8

528752 1003491 P00190142 M 16 B 0 3

211499 1002649 P00282642 M 16 C 0 5

412093 1003469 P00090842 M 17 B 1 11

116467 1005984 P00117942 M 1 A 0 5

product.category.2 product.category.3 purchase years.in.current.city

392408 5 6 11563 1

307820 5 12 3555 2

472137 0 0 6078 2

528752 4 5 10720 2

211499 0 0 5411 3

412093 15 0 6153 4

116467 15 0 7184 5

ID\_unit Prob Stratum

392408 53268 1.86e-05 1

307820 181268 1.86e-05 2

472137 238755 1.86e-05 2

528752 258916 1.86e-05 2

211499 301004 1.86e-05 3

412093 433052 1.86e-05 4

116467 472690 1.86e-05 5

>

> # Using Sample Data vs Original Data

> print("Sample Data")

[1] "Sample Data"

> print("Gender Count")

[1] "Gender Count"

> table(sample$gender)

F M

1 6

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(sample$gender))\*100

F M

14.3 85.7

>

> print("Original Data")

[1] "Original Data"

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(blackfridaysales$gender))\*100

F M

24.6 75.4

>

> print("RESULT : As we can see that distribution has changed by large percentages")

[1] "RESULT : As we can see that distribution has changed by large percentages"

> print("With this distribution we may have much different results")

[1] "With this distribution we may have much different results"

>

> print("Taking another example of Average Purchase")

[1] "Taking another example of Average Purchase"

> print("Sample Data")

[1] "Sample Data"

> mean(sample$purchase)

[1] 7238

>

> print("Original Data")

[1] "Original Data"

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> mean(blackfridaysales$purchase)

[1] 9334

>

> print("RESULT : Mean is differing in 1000s using this sample")

[1] "RESULT : Mean is differing in 1000s using this sample"

> print("Thus much different interpretation if used this sample data")

[1] "Thus much different interpretation if used this sample data"

>

> # revert the data back to orginal

> blackfridaysales = temporary

> sample.size = temporary.size

>

> # CLUSTER SAMPLING

>

> temporary.size = sample.size

> sample.size = 2 # should be less than no of unique value of the column which is basis of sampling

>

> print("CLUSTER SAMPLING")

[1] "CLUSTER SAMPLING"

> cl = cluster(blackfridaysales, c("years.in.current.city"), size=sample.size, method="srswor")

> cl.sample = getdata(blackfridaysales, cl)

> cat("Cluster Sampling for Years in Current City with sample size= ",sample.size)

Cluster Sampling for Years in Current City with sample size= 2> table(cl.sample$years.in.current.city)

1 3

72725 99459

> # head(cl.sample)

> # tail(cl.sample)

>

> # Using Sample Data vs Original Data

> print("Sample Data")

[1] "Sample Data"

> print("Gender Count")

[1] "Gender Count"

> table(cl.sample$gender)

F M

40282 131902

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(cl.sample$gender))\*100

F M

23.4 76.6

>

> print("Original Data")

[1] "Original Data"

> print("Percentage wise Gender Distribution")

[1] "Percentage wise Gender Distribution"

> prop.table(table(blackfridaysales$gender))\*100

F M

24.6 75.4

>

> print("RESULT : As we can see that distribution has changed in sample data but not much deflection wrt original")

[1] "RESULT : As we can see that distribution has changed in sample data but not much deflection wrt original"

>

> # revert the data back to original

> sample.size = temporary.size

>

|  |
| --- |
| > print("RESULT: Thus of all sampling in this data SYSTEMATIC SAMPLING WITH EQUAL PROBABILITY & STRATIFIED SAMPLING samples get deflected most as compared to original data")  [1] "RESULT: Thus of all sampling in this data SYSTEMATIC SAMPLING WITH EQUAL PROBABILITY & STRATIFIED SAMPLING samples get deflected most as compared to original data"  > print("Rest of the sampling data interpretations remains similar to the original data")  [1] "Rest of the sampling data interpretations remains similar to the original data" |
|  |
| |  | | --- | | > | |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# REQUIREMENT 6

# Implementation of any feature(s) not mentioned in the specification

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# EXTRA

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# STRING FUNCTIONS

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> # Replacing city Names

>

> print("City Category currently is A,B,C which is vague and does not give clear idea of its meaning")

[1] "City Category currently is A,B,C which is vague and does not give clear idea of its meaning"

> print("So let's rename the category")

[1] "So let's rename the category"

> print("Originally the city categories are")

[1] "Originally the city categories are"

> names(table(city))

[1] "A" "B" "C"

> print("Renaming them to A: urban, B: town, C: rural")

[1] "Renaming them to A: urban, B: town, C: rural"

> temp = city

>

> # made a function to rename cities

>

> renameCity = function (data,this,with){

+ str\_replace\_all(data,this,with)

+ }

>

> city = renameCity(city, 'A', "urban")

> city = renameCity(city, 'B', "town")

> city = renameCity(city, 'C', "rural")

>

> print("Updated city is")

[1] "Updated city is"

> names(table(city))

[1] "rural" "town" "urban"

>

>

> # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> # TIBBLE

> # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>

> print("Creating tibble of data")

[1] "Creating tibble of data"

> bfs = as.tibble(blackfridaysales)

> bfs

# A tibble: 537,577 x 11

user.id product.id gender occupation city years.in.curren~ marital.status

*<int>* *<fct>* *<fct>* *<int>* *<fct>* *<dbl>* *<int>*

1 1000001 P00069042 F 10 A 3 0

2 1000001 P00248942 F 10 A 3 0

3 1000001 P00087842 F 10 A 3 0

4 1000001 P00085442 F 10 A 3 0

5 1000002 P00285442 M 16 C 5 0

6 1000003 P00193542 M 15 A 4 0

7 1000004 P00184942 M 7 B 3 1

8 1000004 P00346142 M 7 B 3 1

9 1000004 P0097242 M 7 B 3 1

10 1000005 P00274942 M 20 A 2 1

# ... with 537,567 more rows, and 4 more variables: product.category.1 *<int>*,

# product.category.2 *<dbl>*, product.category.3 *<dbl>*, purchase *<int>*

>

> # Observation

>

> print("Average year a shopper stayed in the current city")

[1] "Average year a shopper stayed in the current city"

> print("For A")

[1] "For A"

> dplyr::filter(bfs, city=="A") %>% summarise(mean(years.in.current.city))

# A tibble: 1 x 1

`mean(years.in.current.city)`

*<dbl>*

1 2.81

> print("For B")

[1] "For B"

> dplyr::filter(bfs, city=="B") %>% summarise(mean(years.in.current.city))

# A tibble: 1 x 1

`mean(years.in.current.city)`

*<dbl>*

1 2.88

> print("For C")

[1] "For C"

> dplyr::filter(bfs, city=="C") %>% summarise(mean(years.in.current.city))

# A tibble: 1 x 1

`mean(years.in.current.city)`

*<dbl>*

1 2.88

>

> # Observation

>

> print("Average Purchase made in each City Category on the basis of Stay in the Current City")

[1] "Average Purchase made in each City Category on the basis of Stay in the Current City"

> print("For A")

[1] "For A"

> dplyr::filter(bfs, city=="A") %>% group\_by(years.in.current.city) %>% summarise(mean(purchase))

# A tibble: 5 x 2

years.in.current.city `mean(purchase)`

*<dbl>* *<dbl>*

1 1 9029.

2 2 8928.

3 3 9015.

4 4 8957.

5 5 8880.

> print("For B")

[1] "For B"

> dplyr::filter(bfs, city=="B") %>% group\_by(years.in.current.city) %>% summarise(mean(purchase))

# A tibble: 5 x 2

years.in.current.city `mean(purchase)`

*<dbl>* *<dbl>*

1 1 8947.

2 2 9225.

3 3 9251.

4 4 9231.

5 5 9240.

> print("For C")

[1] "For C"

> dplyr::filter(bfs, city=="C") %>% group\_by(years.in.current.city) %>% summarise(mean(purchase))

# A tibble: 5 x 2

years.in.current.city `mean(purchase)`

*<dbl>* *<dbl>*

1 1 9900.

2 2 9768.

3 3 9900.

4 4 9893.

5 5 9856.

>

>

> # CONCLUSIONS

>

> print("We analyzed following points and could use them to increase our sales in next black friday")

[1] "We analyzed following points and could use them to increase our sales in next black friday"

> print("Number of Male Shoppers > Female Shoppers")

[1] "Number of Male Shoppers > Female Shoppers"

> print("Products in Product Category 2 sold most")

[1] "Products in Product Category 2 sold most"

> print("People generally spent over $5000 in sales")

[1] "People generally spent over $5000 in sales"

> print("People in age range 26-35 purchase most")

[1] "People in age range 26-35 purchase most"

> print("There are highest average sales in City Category ‘C’ as compared to other")

[1] "There are highest average sales in City Category ‘C’ as compared to other"

> print("Unequal Probability sampling technique could be used over this dataset for best results")

[1] "Unequal Probability sampling technique could be used over this dataset for best results"