Exercise:

I. ARITHMETIC MEAN

a) Write suitable R code to compute the average of the following values.

12,7,3,4.2,18,2,54,-21,8,-5

b) Compute the mean after applying the trim option and removing 3 values from each

end.

c) Compute the mean of the following vector .

(12,7,3,4.2,18,2,54,-21,8,-5,NA)

#If there are missing values, then the mean function returns NA.

# Find mean dropping NA values.

#To drop the missing values from the calculation use na.rm = TRUE

code:

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

result.mean <- mean(x)

print(result.mean)

result.mean <- mean(x,trim = 0.3)

print(result.mean)

x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)

result.mean <- mean(x)

print(result.mean)

result.mean <- mean(x,na.rm = TRUE)

print(result.mean)

II.MEDIAN

Write suitable R code to compute the median of the following values.

12,7,3,4.2,18,2,54,-21,8,-5

code:

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

median.result <- median(x)

print(median.result)

III. MODE

Calculate the mode for the following numeric as well as character data set in R.

(2,1,2,3,1,2,3,4,1,5,5,3,2,3) , (&quot;o&quot;,&quot;it&quot;,&quot;the&quot;,&quot;it&quot;,&quot;it&quot;)

code:

getmode <- function(v) {

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

}

v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)

result <- getmode(v)

print(result)

charv <- c("o","it","the","it","it")

result <- getmode(charv)

print(result)

UNIVARIATE ANALYSIS IN R - MEASURES OF DISPERSION

Exercise: 4

Download mpg dataset which contains Fuel economy data from 1999 and 2008 for 38

popular models of car from the URL given below.

<https://vincentarelbundock.github.io/Rdatasets/datasets.html>

Answer the following queries

i) Find the car which gives maximum city miles per gallon

ii) Find the cars which gives minimum disp in compact and subcompact class

code:

mpg <- read.csv("mpg.csv")

max\_city\_mpg <- which.max(mpg$cty)

mpg[max\_city\_mpg, "model"]

compact <- mpg[mpg$class %in% c("compact", "subcompact"), ]

min\_disp\_compact <- which.min(compact$displ[compact$class == "compact"])

min\_disp\_subcompact <- which.min(compact$displ[compact$class == "subcompact"])

compact[min\_disp\_compact, "model"]

compact[min\_disp\_subcompact, "model"]

Exercise: 5

Use the same dataset as used in Exercise 4 and perform the following queries

i) Find the standard deviation of city milles per gallon

ii) Find the variance of highway milles per gallon

code:

mpg <- read.csv("mpg.csv")

sd\_city <- sd(mpg$cty)

cat("Standard deviation of city miles per gallon: ", sd\_city, "\n")

var\_highway <- var(mpg$hwy)

cat("Variance of highway miles per gallon: ", var\_highway, "\n")

Exercise 6

Use the same dataset and perform the following queries

i) Find the range of the disp in the data set mpg

ii) Find the Quartile of the disp in the data set mpg

iii) Find the IQR of the disp column in the data set mpg

code:

mpg <- read.csv("mpg.csv")

disp\_range <- range(mpg$disp)

cat("Range of disp: ", disp\_range, "\n")

disp\_quartiles <- quantile(mpg$disp)

cat("Quartiles of disp: ", disp\_quartiles, "\n")

disp\_iqr <- IQR(mpg$disp)

cat("Interquartile range of disp: ", disp\_iqr, "\n")

Exercise 7

#Install Library

library(e1071)

a. Find the skewness of city miles per mileage in the data set mpg ?

Use qplot function and display the graph for the city miles per mileage column

b. Find the kurtosis of city miles per mileage in the data set mpg

Use qplot function and display the graph for the city miles per mileage column

code:

library(ggplot2)

data(mpg)

library(moments)

skewness(mpg$cty / mpg$hwy)

library(ggplot2)

qplot(mpg$cty / mpg$hwy, geom = "histogram", bins = 20, xlab = "City Miles per Gallon")

# Calculate the kurtosis of the "city" variable in the mpg dataset

library(moments)

kurtosis(mpg$cty / mpg$hwy)

library(ggplot2)

qplot(mpg$cty / mpg$hwy, geom = "histogram", bins = 20, xlab = "City Miles per Gallon")

Exercise: 8

Reference Status Gender TestNewOrFollowUp

1 KRXH Accepted Female Test1 New

2 KRPT Accepted Male Test1 New

3 FHRA Rejected Male Test2 New

4 CZKK Accepted Female Test3 New

5 CQTN Rejected Female Test1 New

6 PZXW Accepted Female Test4 Follow-up

7 SZRZ Rejected Male Test4 New

8 RMZE Rejected Female Test2 New

9 STNX Accepted Female Test3 New

10 TMDW Accepted Female Test1 New

i) Load the dataset and Create a data frame and name it as dataframe1

ii) Load the function for crosstab

code:

dataframe1 <- data.frame(

Reference = c("KRXH", "KRPT", "FHRA", "CZKK", "CQTN", "PZXW", "SZRZ", "RMZE", "STNX", "TMDW"),

Status = c("Accepted", "Accepted", "Rejected", "Accepted", "Rejected", "Accepted", "Rejected", "Rejected", "Accepted", "Accepted"),

Gender = c("Female", "Male", "Male", "Female", "Female", "Female", "Male", "Female", "Female", "Female"),

TestNewOrFollowUp = c("Test1", "Test1", "Test2", "Test3", "Test1", "Follow-up", "Test4", "Test2", "Test3", "Test1"))

library(gmodels)

Exercise: 9

i) Use Two Categorical Variables and Discover the relationships within a

dataset

ii) Next, using the xtabs() function, apply two variables from “dataframe1 “, to

create a table delineating the relationship between the “Reference”

category, and the “Status” category.

iii) Save the file in the name of dataframe2

code:

Load the ggplot2 library

library(ggplot2)

# Create the stacked bar chart

ggplot(mtcars, aes(x = factor(cyl), fill = factor(am))) +

geom\_bar()

# Create a subset of the mtcars dataset with the "Reference" and "Status" columns

dataframe1 <- data.frame(

Reference = c("A", "B", "A", "B", "A", "B"),

Status = c("C", "C", "D", "D", "C", "D")

)

# Create a table using xtabs() function

dataframe2 <- xtabs(~ Reference + Status, dataframe1)

dataframe2

write.csv(dataframe2, file = "dataframe2.csv")

Exercise: 10

Use the same data frame using three Categorical Variables create a Multi-Dimensional Table

Apply three variables from “dataframe1” to create a Multi-Dimensional Cross-Tabulation of

“Status“, “Gender“, and “Test“.

code:

# Load the Titanic dataset

data(Titanic)

# Create a subset of the data with three categorical variables

dataframe1 <- as.data.frame(Titanic)

dataframe1 <- subset(dataframe1, select = c(Survived, Class, Sex))

# Create a multi-dimensional table using xtabs() function

dataframe2 <- xtabs(~ Survived + Class + Sex, dataframe1)

dataframe2

# Create a sample dataframe with three categorical variables

dataframe1 <- data.frame(

Status = c("Positive", "Negative", "Positive", "Positive", "Negative", "Negative"),

Gender = c("Male", "Female", "Male", "Female", "Male", "Female"),

Test = c("Test1", "Test1", "Test2", "Test2", "Test1", "Test2")

)

# Create a multi-dimensional table using xtabs() function

dataframe2 <- xtabs(~ Status + Gender + Test, dataframe1)

dataframe2

Exercise: 11

Row Percentages

The R package “tigerstats” is required for the next two exercises.

1) Create an xtabs() formula that cross-tabulates “Status“, and “Test“.

2) Enclose the xtabs() formula in the tigerstats function, “rowPerc()” to display row

percentages for “Status” by “Test“.

# Install the tigerstats package

install.packages("tigerstats")

# Load the tigerstats package

library(tigerstats)

# Create a sample dataframe with two categorical variables

dataframe1 <- data.frame(

Status = c("Positive", "Negative", "Positive", "Positive", "Negative", "Negative"),

Test = c("Test1", "Test1", "Test2", "Test2", "Test1", "Test2")

)

# Create an xtabs() formula to cross-tabulate "Status" and "Test"

xtabs(~ Status + Test, dataframe1)

# Create an xtabs() formula to cross-tabulate "Status" and "Test"

table1 <- xtabs(~ Status + Test, dataframe1)

# Display row percentages using the rowPerc() function

rowPerc(table1)

Exercise 12

Column Percentages

1) Create an xtabs() formula that cross-tabulates “Status“, and “Test“.

2) Enclose the xtabs() formula in the tigerstats function, “colPerc()” to display row

percentages for “Status” by “Test“.

code:

# Install the tigerstats package

install.packages("tigerstats")

# Load the tigerstats package

library(tigerstats)

# Create a sample dataframe with two categorical variables

dataframe1 <- data.frame(

Status = c("Positive", "Negative", "Positive", "Positive", "Negative", "Negative"),

Test = c("Test1", "Test1", "Test2", "Test2", "Test1", "Test2")

)

# Create an xtabs() formula to cross-tabulate "Status" and "Test"

xtabs(~ Status + Test, dataframe1)

# Create an xtabs() formula to cross-tabulate "Status" and "Test"

table1 <- xtabs(~ Status + Test, dataframe1)

# Display column percentages using the colPerc() function

colPerc(table1)

13. Write a program for creating a pie-chart in R using the input vector(21,62,10,53). Provide labels for the chart as ‘London’, ‘New York’, ‘Singapore’, ‘Mumbai’. Add a title to the chart as ‘city pie-chart’ and add a legend at the top right corner of the chart.

# Define the input vector

data <- c(21, 62, 10, 53)

# Define the labels for the chart

labels <- c("London", "New York", "Singapore", "Mumbai")

# Create the pie chart with labels, title, and legend

pie(data, labels = labels, main = "City Pie-Chart")

legend("topright", legend = labels, fill = rainbow(length(labels)))

14. Create a 3D Pie Chart for the dataset “political Knowledge” with suitable

labels,colours and a legend at the top right corner of the chart.

code:

#Install and load the plotrix package

install.packages("plotrix")

library(plotrix)

# Example dataset

data <- c(10, 25, 15, 30, 20)

# Example labels

labels <- c("Label1", "Label2", "Label3", "Label4", "Labe#l5")

# Example colors for each slice

colors <- c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd")

# Create the 3D pie chart

pie3D(data, labels = labels, explode = 0.1, col = colors, main = "3D Pie Chart")

# Add a legend at the top right corner of the chart

15. Write a program for creating a bar chart using the vectors H=c(7,12,28,3,41) and

M=c(“mar”, “apr”, “may”, “jun”, “jul”). Add a title to the chart as “Revenue chart”.

code:

H <- c(7, 12, 28, 3, 41)

M <- c("mar", "apr", "may", "jun", "jul")

# create bar chart

barplot(H, names.arg = M, xlab = "Month", ylab = "Revenue", main = "Revenue chart")

16. Make a histogram for the “AirPassengers“dataset, start at 100 on the x-axis, and from

values 200 to 700, make the bins 200 wide

code:

# Load the AirPassengers dataset

data(AirPassengers)

# Create a histogram with custom bin widths and starting point

hist(AirPassengers,

xlim = c(100, 700),

breaks = seq(200, 700, by = 200),

main = "AirPassengers Histogram",

xlab = "Passenger Count")

17. Create a Boxplot graph for the relation between &quot;mpg&quot;(miles per galloon) and

&quot;cyl&quot;(number of Cylinders) for the dataset &quot;mtcars&quot; available in R Environment.

code:

# Load the mtcars dataset

data(mtcars)

# Create a boxplot of mpg vs cyl

boxplot(mpg ~ cyl, data = mtcars,

main = "Boxplot of Miles per Gallon by Number of Cylinders",

xlab = "Number of Cylinders",

ylab = "Miles per Gallon")