**#Exercise on WHO Data ------1**

**#R functions**

**#Loading data**

**#Subsetting the data**

**#Data analysis using R**

**#Plot data (DOT, BOX, HIST)**

**#Handling missing data**

**#Exercise on USDA Data ------2**

**#exploring data**

**#Working with plots (DOT HIST BOX)**

**#Adding variables to dataset**

**#R functions**

>dataframe\_name=data.frame(array1,array2,array3)

> seq(1,10) # displays the value in sequence

[1] 1 2 3 4 5 6 7 8 9 10

>rbind(<dataframe1,dataframe2)

>array=c()

**#Loading data**

>getwd() #get the current working directory

>WHO=read.csv("WHO.csv")

>str(WHO) # to display the data from R

#factor is a categorical data that repeats

#numerical summary

>summary(WHO)

**#Subsetting the data**

>WHO\_Eup=subset(WHO, Region==”Europe”)

>str(WHO\_Eup)

>write.csv(WHO\_Eup, “WHO\_eup.csv”)

>ls() # to see what variables you have in R

>rm(WHO\_Eup) #WHO\_Eup has gone!

>ls() # now you should not see the WHO\_Eup data set

**#Data analysis using R**

>WHO$Under15 # this is how R recognizes a column of a data frame

>mean(WHO$Under15) # will give the mean, you can also use sd(),summart() functions on a column

>which.min(WHO$Under15) # returns the index of the row that has minimum value.

[1] 86

>WHO$Country[86] # access the row with the column

[1] Japan

>table(WHO$Region) #Group and sum the column

>tapply(WHO$Over60, WHO$Region, mean) #displays mean prder by region

**#Plot data**

>plot(WHO$GNI,WHO$Fertilityrate) # this will display a dot plot

>outliers=subset(WHO, GNI>10000 & FertilityRate>2.5) # Sub setting the outliers

>nrow(outliers) #displays rows

[1] 7

>outliers[c('Country','GNI','FertilityRate')] # create a vector and display data.

>hist(WHO$CellularSubscribers) # to plot a histogram

>boxplot(WHO$LifeExpectancy ~ WHO$Country, xlab=' ',ylab='Life Expectancy',Main='LifeExpectancy by Country') # to view the statistical range of the data.

**#Handling missing data**

tapply(WHO$LiteracyRate, WHO$Region, min, na.rm=’True’) # this is the way we can exclude missing data from datset.

**#######E X C E R S I Z E –2##################\***

**#####################################\***

**#exploring data**

> getwd() #to see the working directory

> USDA = read.csv('USDA.csv') #load data

> summary(USDA) # to show the summary of the numerical variables

> names(USDA) # to retrieve the column names of the data frame

> match(“CAVIAR”, USDA$Description) # to find the matching description and returns the Index

[1] 4154

>USDA$Sodium[4154] # to find the level of sodium in Caviar

> USDA$Sodium[match(“CAVIAR”, USDA$Description)] # this also works as above

**# Working with plots**

>plot(USDA$Protein, USDA$TotalFat) #This will display a dot plot

#the plot shows a triangular shape which means the food that has protein doesn’t have fat and vice versa.

> hist(USDA$VitaminC, xlab='Vitamin C (mg)', main='Histogramof Vitamin C Levels') **# show the frequency of VitaminC**

#Xlim will specify the X-axis area where the histogram zoom into. Breaks will give the cell breakupsfor original interval

> hist(USDA$VitaminC, xlab='Vitamin C (mg)', main='Histogramof Vitamin C Levels',xlim=c(0,100),breaks=100)

**# BOX Plots**

> boxplot(USDA$Sugar, main ='heading', ylab='sugur (g)')

**#Adding variables to dataset**

#the Records in the data frame, that have higher sodium value than mean.

>HighSodium = USDA$Sodium > mean(USDA$sodium , na.rm=True) # HighSodium is a vector of Boolean

> HighSodium = as.numeric(USDA$Sodium > mean(USDA$Sodium , na.rm='true')) # this will be a numeric 1=true and 0=false.

> str(USDA$HighSodium) # to display the values of vector.

> USDA$HighFat= as.numeric(USDA$TotalFat > mean(USDA$TotalFat , na.rm='true')) # adding column to data frame.

> USDA$HighSodium

#similarly add the below vectors as well.

>USDA$HighFat

>USDA$HighCarbohidrate

>USDA$HighProtein

>USDA$HighSoudium

#how many foods have higher than average sodium value and lower than average sodium values.

>table(USDA$HighSodium)

0 1

4884 2090

#show the number with high sodium and low fat and vice versa

>table(USDA$HighSodium, USDA$HighFat)

0 1

0 3529 1355

1 1378 712

>

# tapply function has 3 arguments, tapply(argument1, argument2, argument3)

# Group argumen1 by argument2 and apply argument 3

#find the mean iron values for high and low protein food

>tapply(USDA$Iron, USDA$highprotein, mean,na.rm=True)