Import dataset from the following link: AirQuality Data Set

Perform the following written operations:

1. Read the file in Zip format and get it into R.

Ans:

>zip.url<-"http://archive.ics.uci.edu/ml/machine-learning-databases/00360/AirQualityUCI.zip"

>temp<-tempfile()

>download.file(zip.url,temp)

>data<-data.frame(read.csv(unz(temp,'AirQualityUCI.csv'),sep = ";"))

2. Create Univariate for all the columns.

Ans:

>summary(data) #gives us range, median, upper and lower quartiles

that divide the data into four equal sizes,

maximum value and minimum value for all Univariate columns.

>fivenum(data$ PT08.S1.CO.) # min, max, median, lower-hinge and upper-hinge values.

>describe(data$ PT08.S1.CO.) #Number of rows,Standard deviation,Trimmed mean

#Mean absolute deviation,Skewness,Kurtosis,Standard error

>stat.desc(data$ PT08.S1.CO.) #Variance,Coefficient of variation,Confidence interval for mean

3. Check for missing values in all columns.

Ans:

>data<-AirQualityUCI

>data[data=="-200"]<-NA

> which(is.na(data$Date))

> which(is.na(data$Time))

> which(is.na(data$CO.GT.))

> which(is.na(data$NMHC.GT.))

> which(is.na(data$PT08.S1.CO.))

> which(is.na(data$C6H6.GT.))

> which(is.na(data$PT08.S2.NMHC.))

> which(is.na(data$NOx.GT.))

> which(is.na(data$PT08.S4.NO2.))

> which(is.na(data$PT08.S5.O3.))

> which(is.na(data$T))

> which(is.na(data$RH))

> which(is.na(data$AH))

> which(is.na(data$X))

> which(is.na(data$X.1))

or

>sumarry(data)

4. Impute the missing values using appropriate methods.

Ans:

>data<-AirQualityUCI

>library(Hmisc)

>summary(data)

>data$CO.GT.<- with(data, impute(CO.GT., 'random'))

>data$PT08.S1.CO.<- with(data, impute(PT08.S1.CO., 'random'))

>data$NMHC.GT.<- with(data, impute(NMHC.GT., 'random'))

>data$PT08.S2.NMHC.<- with(data, impute(PT08.S2.NMHC., 'random'))

>data$NOx.GT.<- with(data, impute(NOx.GT., 'random'))

>data$PT08.S3.NOx.<- with(data, impute(PT08.S3.NOx., 'random'))

>data$NO2.GT.<- with(data, impute(NO2.GT., 'random'))

>data$PT08.S4.NO2.<- with(data, impute(PT08.S4.NO2., 'random'))

>data$PT08.S5.O3.<- with(data, impute(PT08.S5.O3., 'random'))

>data$T <- with(data, impute(T , 'random'))

>data$RH<- with(data, impute(RH, 'random'))

>data$AH<- with(data, impute(AH, 'random'))

>str(data1)

>summary(data)

OR

>library(caret)

>knnImputation(data, k = 10, scale = T, meth = "weighAvg", distData = NULL)

OR

>centralImputation(data)

>head(data,10)

5. Create bi-variate analysis for all relationships.

Ans:

>data<-AirQualityUCI

>summary(data)

>plot(data$NOx.GT.~data$PT08.S2.NMHC.)

>plot(data$PT08.S1.CO.~data$PT08.S3.NOx.)

>plot(data$NO2.GT.~data$PT08.S4.NO2.)

>plot(data$PT08.S5.O3.~data$T)

6. Test relevant hypothesis for valid relations.

Ans:

>data<-AirQualityUCI

>plot(data$PT08.S1.CO.,data$T)

>lm(formula=data$PT08.S3.NOx.~data$NOx.GT.)

>lm(formula = data$PT08.S1.CO.~data$T)

>lm(formula = data$NMHC.GT.~data$PT08.S2.NMHC.)

>plot(data$PT08.S5.O3.,data$NOx.GT.)

>lm(formula =data$PT08.S5.O3.~data$NOx.GT.)

>pnorm(1.49)

>pnorm(1.097)

>qnorm(0.9318879)

>qnorm(0.8636793)

>library(car)

>mod=lm(data$PT08.S5.O3. ~ data$NOx.GT.)

>summary(mod)

>predict(mod)

7. Create cross tabulations with derived variables.

Ans:

>View(data)

# 2-Way Frequency Table

>attach(data)

# A will be rows, B will be columns

>mytable <- table(A,B)

>mytable # print table

# A frequencies (summed over B)

>margin.table(mytable, 1)

# B frequencies (summed over A)

>margin.table(mytable, 2)

# cell percentages

>prop.table(mytable)

# row percentages

>prop.table(mytable, 1)

# column percentages

>prop.table(mytable, 2)

8. Check for trends and patterns in time series.

Ans:

>data<-AirQualityUCI

# frequency 4 => Quarterly Data

>ts (data, frequency = 4, start = c(1959, 2))

# freq 12 => Monthly data.

>ts (1:10, frequency = 12, start = 1990)

# Yearly Data

>ts (AirQualityUCI, start=c(2009), end=c(2014), frequency=1)

>ts (1:1000, frequency = 365, start = 1990)# freq 365 => daily data.

>tsAirqualityUCI <- EuStockMarkets[, 1]

#plot time series

>tsAirqualityUCI <- EuStockMarkets[, 1] # ts data

>decomposedRes <- decompose(tsAirqualityUCI, type="mult") # use type = "additive" for additive components

>plot (decomposedRes)

9. Find out the most polluted time of the day and the name of the chemical compound.

Ans:

>data<-AirQualityUCI

>df<-data[,-c(1,2,3,6,13,14,15,16,17)]

>df<-transform(df, sum=rowSums(df))

>head(df)

>names(df)

>data$sum<-df$sum

>head(data)

>data[which.max(data$sum),]

Date Time CO.GT. PT08.S1.CO. NMHC.GT. C6H6.GT. PT08.S2.NMHC. NOx.GT.

23/11/2004 20.00.00 11,5 1918 743 49,4 1958 1358

PT08.S3.NOx. NO2.GT. PT08.S4.NO2. PT08.S5.O3. T RH AH X X.1 sum

335 190 2477 2237 11,5 76,2 1,0324 NA NA 11216

most polluted day is 23/11/2004 and time is 20.00.00 and chemical compound is PT08.S5.O3.(indium oxide)