#-------------------Assignment 10.2 ---------------------

# 1. Import Zip File

library(readxl)

setwd("E:/Data Analytics with RET/Assignment/")

AirQuality <- read\_xlsx(unzip("E:/Data Analytics with RET/Assignment/AirQualityUCI.zip"))

Air <- AirQuality

dim(Air)

str(Air)

View(Air)

# 2. Create Univariate for all the columns.

library(psych)

describe(Air)

# 3. Check for missing values in all columns.

Air[Air == -200] <- NA

View(Air)

library(VIM)

aggr(Air, col=c('navyblue','yellow'),

numbers=TRUE, sortVars=TRUE,

labels=names(Air), cex.axis=.7,

gap=3, ylab=c("Missing data","Pattern")) # graphical presentation of NAs

sapply(Air, function(x) sum(is.na(x))) # count of NAs

# Variable NMHC(GT) is having 90% of missing values.

# Hence, NMHC(GT) is not considered and omitted from the data frame

Air$`NMHC(GT)` <- NULL

#-----------------------------------------------------------------------------

# 4. Impute the missing values using appropriate methods.

names(Air)

Air$Date1 <- as.numeric(as.Date(Air$Date))

library(mice)

impute <- mice(Air[,-c(1,2)], m=5, maxit = 5, method = 'cart', seed = 100) # impute missing values

summary(impute)

complete <- complete(impute) # replaces the NAs with imputed values

str(complete)

sapply(complete, function(x) sum(is.na(x))) # check missing values

#-----------------------------------------------------------------------------

# 5. Create bi-variate analysis for all relationships

cor(complete) # values

pairs(complete) # graph

#-----------------------------------------------------------------------------

final <- complete

final$Date <- Air$Date

final$Time <- Air$Time

library(stringr)

final$Time1 <- sub(".+? ", "", final$Time)

final$datetime <- as.POSIXct(paste(final$Date, final$Time1), format="%Y-%m-%d %H:%M:%S")

View(final)

str(final)

#-----------------------------------------------------------------------------

# 6. Test relevant hypothesis for valid relations

t.test(final$`CO(GT)`, final$`PT08.S1(CO)`, paired = T)

t.test(final$`C6H6(GT)`, final$`PT08.S2(NMHC)`, paired = T)

t.test(final$`NOx(GT)`, final$`PT08.S3(NOx)`, paired = T)

mod <- lm(final$`CO(GT)`~final$Date1)

summary(mod)

mod <- lm(final$`CO(GT)`~final$T)

summary(mod)

mod <- lm(final$`CO(GT)`~final$RH)

summary(mod)

#-----------------------------------------------------------------------------

# 7. Create cross tabulations with derived variables

range(final$RH)

final <- within(final,

{

Tcat <- NA

Tcat[T<0] <- "Minus"

Tcat[T>=0 & T<=10] <- "Low"

Tcat[T>10 & T<=20] <- "Medium"

Tcat[T>20 & T<=30] <- "High"

Tcat[T>30] <- "Very High"

})

final <- within(final,

{

RHcat <- NA

RHcat[RH<20] <- "Very Low"

RHcat[RH>=20 & RH<=40] <- "Low"

RHcat[RH>40 & RH<=60] <- "Medium"

RHcat[RH>60 & RH<=80] <- "High"

RHcat[RH>80] <- "Very High"

})

mytable <- xtabs(`CO(GT)` ~ +Tcat +RHcat, data = final)

ftable(mytable) # print table

summary(mytable) # chi-square test of indepedence

mytable <- xtabs(`C6H6(GT)` ~ +Tcat +RHcat, data = final)

ftable(mytable) # print table

summary(mytable) # chi-square test of indepedence

mytable <- xtabs(`NOx(GT)` ~ +Tcat +RHcat, data = final)

ftable(mytable) # print table

summary(mytable) # chi-square test of indepedence

with(final, tapply(`NO2(GT)`, list(Tcat=Tcat, RHcat=RHcat), sd)) # using with()

with(final, tapply(`NO2(GT)`, list(Tcat=Tcat, RHcat=RHcat), mean))

#-----------------------------------------------------------------------------

# 8. Check for trends and patterns in time series.

library(xts)

timeseries <- xts(final$`CO(GT)`, final$datetime)

plot(timeseries)

summary(timeseries)

#-----------------------------------------------------------------------------

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

names(final)

library(dplyr)

polluted <- final%>%group\_by(Time)%>%

select(Time, `CO(GT)`, `C6H6(GT)`, `NO2(GT)`, `NOx(GT)` )%>%

summarise(CO = mean(`CO(GT)`), C6H6 = mean(`C6H6(GT)`), NO2 = mean(`NO2(GT)`), NOX =mean(`NOx(GT)`))%>%

polluted[c(which.max(polluted$CO),which.max(polluted$C6H6),which.max(polluted$NO2),which.max(polluted$NOX)),]

# 19:00:00 is the most polluted time of the day with CO, C6H6, NO2 & NOx

#-----------------------------------------------------------------------------