```r
#import parkinsons dataset
park <- read.csv("/Users/sesh/Desktop/Langara Spring/DANA Quant/Ass1/parkinsons.csv", colClasses = "numeric")

#dimension of the dataset
dim(park)

#missing_values for every column
na = colSums(is.na(park))
na

#sum of missing values
sum(is.na(park))

#range of variables in the dataset
summary(park)

#count of outliers
install.packages("dlookr")
library(dlookr)
outl <-diagnose_numeric(park)
View(outl)

#box-plots for outlier detection
qplot(x="",y= park$Jitter, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "jitter", xlab = "", main = "jitter box plot")
qplot(x="",y= park$Jitter.Abs, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "jitter.Abs", xlab = "", main = "jitter.Abs box plot")
qplot(x="",y= park$Jitter.RAP, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "jitter.RAP", xlab = "", main = "jitter.RAP box plot")
qplot(x="",y= park$Jitter.PPQ5, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "jitter.PPQ5", xlab = "", main = "jitter.PPQ5 box plot")
qplot(x="",y= park$Jitter.DDP, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "jitter.DDP", xlab = "", main = "jitter.DDP box plot")
qplot(x="",y= park$Shimmer, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "Shimmer", xlab = "", main = "Shimmer box plot")
qplot(x="",y= park$Shimmer.dB, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "Shimmer.dB", xlab = "", main = "Shimmer.dB box plot")
qplot(x="",y= park$Shimmer.APQ3, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "Shimmer.APQ3", xlab = "", main = "Shimmer.APQ3 box plot")
qplot(x="",y= park$Shimmer.APQ5, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "Shimmer.APQ5", xlab = "", main = "Shimmer.APQ5 box plot")
qplot(x="",y= park$Shimmer.APQ11, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "Shimmer.APQ11", xlab = "", main = "Shimmer.APQ11 box plot")
qplot(x="",y= park$Shimmer.DDA, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "Shimmer.DDA", xlab = "", main = "Shimmer.DDA box plot")
qplot(x="",y= park$NHR, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "NHR", xlab = "", main = "NHR box plot")
qplot(x="",y= park$HNR, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "HNR", xlab = "", main = "HNR box plot")
qplot(x="",y= park$RPDE, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "RPDE", xlab = "", main = "RPDE box plot")
qplot(x="",y= park$PPE, geom="boxplot", col = I("darkblue"), fill = I("lightblue"), ylab = "PPE", xlab = "", main = "PPE box plot")

#creating a duplicate version
parkclean <- park

#filling missing cells for subject
parkclean$subject <-ifelse(is.na(parkclean$subject) & parkclean$age == 60, 24,
                           (ifelse(is.na(parkclean$subject) & parkclean$age == 74 & parkclean$sex == 1 , 28,
                                   (ifelse(is.na(parkclean$subject) & parkclean$age == 61 & parkclean$sex == 0 , 42, parkclean$subject)))))
#checking the missing values
sum(is.na(parkclean))

#count of values where age is greater than 85
cnt <-length(which(parkclean$age>85))
cnt

#correcting age
parkclean$age <-ifelse(parkclean$age == 650 & parkclean$subject == 15, 65,
                       (ifelse(parkclean$age == 749 & parkclean$subject == 28, 74,
                               (ifelse(is.na(parkclean$age) & parkclean$subject == 10, 58 ,parkclean$age)))))

#checking the missing values and summary
sum(is.na(parkclean))
summary(parkclean)

#correcting negative time values
parkclean$test_time <-ifelse(parkclean$test_time< 0, parkclean$test_time*-1,parkclean$test_time)
parkclean$test_time_hr <-ifelse(parkclean$test_time_hr< 0, parkclean$test_time_hr*-1,parkclean$test_time_hr)
parkclean$test_time_min <-ifelse(parkclean$test_time_min< 0, parkclean$test_time_min*-1,parkclean$test_time_min)

#checking the time variables
summary(parkclean)

#count of negative values for jitter.ppq5 and shimmer.apq3
jitppq5 <-length(which(parkclean$Jitter.PPQ5 <0))
shimapq3 <-length(which(parkclean$Shimmer.APQ3 < 0))
jitppq5
shimapq3

#removing negative jitter.ppq5 and shimmer.apq3
parkclean <- parkclean[parkclean$Jitter.PPQ5 >=0, ]
parkclean <- parkclean[parkclean$Shimmer.APQ3 >= 0, ]

#checking the dimensions of the clean dataset
dim(parkclean)

#count of outliers for jitter.ppq5 and shimmer.apq3
jitppq5 <-length(which(parkclean$Jitter.PPQ5 >8))
shimapq3 <-length(which(parkclean$Shimmer.APQ3 >5))
jitppq5
shimapq3

#outlier removing
parkclean <- parkclean[parkclean$Jitter.PPQ5< 8, ]
parkclean <- parkclean[parkclean$Shimmer.APQ3< 5, ]

#checking the dimensions of the clean dataset
dim(parkclean)

#corrrelation coefficient, 2d, 3d density plots and scatter plot of the three pairs
#1. jitter vs shimmer
library(ggpubr)
library(cowplot)
correlation_coeff<- cor(parkclean$Jitter,parkclean$Shimmer)
correlation_coeff

dens <-ggplot(parkclean, aes(x=Jitter, y=Shimmer) ) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour="white")

scat <-ggscatter(parkclean,x="Jitter", y="Shimmer")

plot_grid(dens,scat)

install.packages("plotly")
library(plotly)
library(MASS)
den3d <- kde2d(parkclean$Jitter, parkclean$Shimmer)
plot_ly(x=den3d$x, y=den3d$y, z=den3d$z) %>% add_surface()

#2. jitter vs PPE
correlation_coeff<- cor(parkclean$Jitter,parkclean$PPE)
correlation_coeff

dens <-ggplot(parkclean, aes(x=Jitter, y=PPE) ) +
```

```
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour="white")

scat <-ggscatter(parkclean,x="Jitter", y="PPE")

plot_grid(dens,scat)

den3d <- kde2d(parkclean$Jitter, parkclean$PPE)
plot_ly(x=den3d$x, y=den3d$y, z=den3d$z) %>% add_surface()

#3. jitter vs NHR
correlation_coeff<- cor(parkclean$Jitter,parkclean$NHR)
correlation_coeff

dens <-ggplot(parkclean, aes(x=Jitter, y=NHR) ) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour="white")

scat <-ggscatter(parkclean,x="Jitter", y="NHR")

plot_grid(dens,scat)

den3d <- kde2d(parkclean$Jitter, parkclean$NHR)
plot_ly(x=den3d$x, y=den3d$y, z=den3d$z) %>% add_surface()

#Creating duplicate of the clean version to store z score values.
parkTransform <- parkclean

#Applying Z score transformation on Total updrs, rpde, hnr and DFA
parkTransform$total_UPDRS_Zscore <- as.numeric(scale(parkclean$total_UPDRS))
parkTransform$RDPE_Zscore <- as.numeric(scale(parkclean$RPDE))
parkTransform$HNR_Zscore <- as.numeric(scale(parkclean$HNR))
parkTransform$DFA_Zscore <- as.numeric(scale(parkclean$DFA))
View(parkTransform)

#comaprison of distributions
hist(parkclean$DFA)
hist(parkTransform$DFA_Zscore)

#Applying Robust-Scalar transformation on Total updrs, rpde, hnr and PPE
m <-median(parkclean$total_UPDRS)
qrange <- IQR(parkclean$total_UPDRS)
parkTransform$total_UPDRS_robust <- (parkclean$total_UPDRS-m)/qrange

m <-median(parkclean$RPDE)
qrange <- IQR(parkclean$RPDE)
parkTransform$RPDE_robust <- (parkclean$RPDE-m)/qrange

m <-median(parkclean$HNR)
qrange <- IQR(parkclean$HNR)
parkTransform$HNR_robust <- (parkclean$HNR-m)/qrange

m <-median(parkclean$PPE)
qrange <- IQR(parkclean$PPE)
parkTransform$PPE_robust <- (parkclean$PPE-m)/qrange
View(parkTransform)

#comaprison of distributions
hist(parkclean$PPE)
hist(parkTransform$PPE_robust)

#Applying logarithmic transformation on Total updrs, Shimmer, PPE and Jitter Abs
parkTransform$total_UPDRS_logscore <- log10(parkclean$total_UPDRS+1)
parkTransform$Shimmer_logScore <- log10(parkclean$Shimmer+1)
parkTransform$PPE_logScore <- log10(parkclean$PPE+1)
parkTransform$Jitter.Abs_logScore <- log10(parkclean$Jitter.Abs+1)
View(parkTransform)

#comaprison of distributions
hist(parkclean$PPE)
hist(parkTransform$PPE_logScore)

#checking the variables significant to total_UPDRS
summary(lm(formula = total_UPDRS ~ ., data = parkclean))

#Linear regression model for unstandardized data
model1 <- lm(formula = total_UPDRS ~ Jitter.Abs + Shimmer.APQ5 + HNR + RPDE + DFA + PPE + age, data= parkclean)
summary(model1)

#plotting predictors against total_UPDRS
plot(x=predict(model1),
     y=parkclean$total_UPDRS,
     xlab='predicted value',
     ylab='total_UPDRS',
     main='Observed by Predicted for total_UPDRS')
abline(a=0, b=1)

#duplicate of the clean version created
parkModel2 <- parkclean

#Applying Z score transformation on Total updrs, rpde, hnr, DFA, Shimmer.APQ5, Jitter.Abs and PPE
parkModel2$total_UPDRS_Zscore <- as.numeric(scale(parkclean$total_UPDRS))
parkModel2$RDPE_Zscore <- as.numeric(scale(parkclean$RPDE))
parkModel2$HNR_Zscore <- as.numeric(scale(parkclean$HNR))
parkModel2$DFA_Zscore <- as.numeric(scale(parkclean$DFA))
parkModel2$Shimmer.APQ5_Zscore <- as.numeric(scale(parkclean$Shimmer.APQ5))
parkModel2$Jitter.Abs_Zscore <- as.numeric(scale(parkclean$Jitter.Abs))
parkModel2$PPE_Zscore <- as.numeric(scale(parkclean$PPE))
parkModel2$age_Zscore <- as.numeric(scale(parkclean$age))

#Linear regression model for Z score data
model2 <- lm(formula = total_UPDRS_Zscore ~ RDPE_Zscore + HNR_Zscore + DFA_Zscore + Shimmer.APQ5_Zscore + Jitter.Abs_Zscore + PPE_Zscore + age_Zscore, data= parkModel2)
summary(model2)

#plotting predictors against total_UPDRS_Zscore
plot(x=predict(model2),
     y=parkModel2$total_UPDRS_Zscore,
     xlab='predicted value',
     ylab='total_UPDRS_Zscore',
     main='Observed by Predicted for total_UPDRS_Zscore')
abline(a=0, b=1)

#duplicate of the clean version created
parkModel3 <- parkclean

#Applying robust scaler transformation on Total updrs, rpde, hnr, DFA, Shimmer.APQ5, Jitter.Abs and PPE
m <-median(parkclean$total_UPDRS)
qrange <- IQR(parkclean$total_UPDRS)
parkModel3$total_UPDRS_robust <- (parkclean$total_UPDRS-m)/qrange

m <-median(parkclean$RPDE)
qrange <- IQR(parkclean$RPDE)
parkModel3$RPDE_robust <- (parkclean$RPDE-m)/qrange
```

```r
m <-median(parkclean$HNR)
qrange <- IQR(parkclean$HNR)
parkModel3$HNR_robust <- (parkclean$HNR-m)/qrange

m <-median(parkclean$DFA)
qrange <- IQR(parkclean$DFA)
parkModel3$DFA_robust <- (parkclean$DFA-m)/qrange

m <-median(parkclean$Shimmer.APQ5)
qrange <- IQR(parkclean$Shimmer.APQ5)
parkModel3$Shimmer.APQ5_robust <- (parkclean$Shimmer.APQ5-m)/qrange

m <-median(parkclean$Jitter.Abs)
qrange <- IQR(parkclean$Jitter.Abs)
parkModel3$Jitter.Abs_robust <- (parkclean$Jitter.Abs-m)/qrange

m <-median(parkclean$PPE)
qrange <- IQR(parkclean$PPE)
parkModel3$PPE_robust <- (parkclean$PPE-m)/qrange

m <-median(parkclean$age)
qrange <- IQR(parkclean$age)
parkModel3$age_robust <- (parkclean$age-m)/qrange

#Linear regression model for robust scaler data
model3 <- lm(formula = total_UPDRS_robust ~ RPDE_robust + HNR_robust + DFA_robust + Shimmer.APQ5_robust + Jitter.Abs_robust + PPE_robust + age_robust, data= parkModel3)
summary(model3)

#plotting predictors against total_UPDRS_robust
plot(x=predict(model3),
     y=parkModel3$total_UPDRS_robust,
     xlab='predicted value',
     ylab='total_UPDRS_robust',
     main='Observed by Predicted for total_UPDRS_robust')
abline(a=0, b=1)

#duplicate of the clean version created
parkModel4 <- parkclean

#Applying logarithmic transformation on Total updrs, rpde, hnr, DFA, Shimmer.APQ5, Jitter.Abs and PPE
parkModel4$total_UPDRS_log <- log10(parkclean$total_UPDRS+1)
parkModel4$RDPE_log <- log10(parkclean$RPDE+1)
parkModel4$HNR_log <- log10(parkclean$HNR+1)
parkModel4$DFA_log <- log10(parkclean$DFA+1)
parkModel4$Shimmer.APQ5_log <- log10(parkclean$Shimmer.APQ5+1)
parkModel4$Jitter.Abs_log <- log10(parkclean$Jitter.Abs+1)
parkModel4$PPE_log <- log10(parkclean$PPE+1)
parkModel4$age_log <- log10(parkclean$age+1)

#Linear regression model for logarithmic transformation
model4 <- lm(formula = total_UPDRS_log ~ RDPE_log + HNR_log + DFA_log + Shimmer.APQ5_log + Jitter.Abs_log+ PPE_log + age_log, data= parkModel4)
summary(model4)

#plotting predictors against total_UPDRS_log
plot(x=predict(model4),
     y=parkModel4$total_UPDRS_log,
     xlab='predicted value',
     ylab='total_UPDRS_log',
     main='Observed by Predicted for total_UPDRS_log')
abline(a=0, b=1)
```