```
###################
# analysis.r      #
# Giammario Fulco #
###################

setwd ("~/Desktop/Homework/analysis")


library(MASS)
###### CREATING THE DATA FRAME FOR INPUT DATA #######


lvl<-c(-1,1)
dm<-expand.grid (A=lvl,B=lvl,C=lvl,D=lvl,E=lvl)
dm<-data.frame(StdOrder=c(1:nrow (dm)),dm)
dm<-data.frame(RunOrder=sample(nrow(dm)),dm)
dm<-dm[order(dm$RunOrder),]
data<-
c(40.606,0.003,0.003,0.003,41.304,0.004,0.006,0.004,40.25,47.286,42.002,0.002,0.004,0.01,47.303,0.004,0.002,0.01,0.004,0.003,0.005,0.003,0.002,45.935,0.009,0.003,0.004,0.005,0.003,0.003,0.003

df<-data.frame(dm,Data=data)
#write.table(df,"~/Desktop/Homework/analysis/data_.dat",row=F)

############################################################
## ANALYSIS                                               ##
############################################################

############# MODEL 1 ##################
df.lm<-lm(Data~A*B*C*D*E,d=df)

# Daniel ' s method : pl o t the e f f e c t s
effect<-as.vector(df.lm$effect)[2:length(df.lm$eff)]
qn<-qqnorm(effect,datax=T,ylab="EffectQuantiles")
qqline(effect,datax = T)
text(qn$x,qn$y,lab=names(df.lm$eff)[2:length(df.lm$eff)],pos=4)

# ############# MODEL 2 ################

# First model, after Daniel' s method
df.lm2<-lm(Data~(A+B+C+D+E), d=df)
anova (df.lm2)
hist(df.lm2$residuals,main="Histogram of residuals for LinearModel 2",xlab="Residuals")
shapiro.test(df.lm2$residuals) #<0.05 -> not normal
qqnorm(df.lm2$residuals)
qqline(df.lm2$residuals)
plot(jitter(df.lm2$fit),df.lm2$residuals,xlab="Fitted Values",ylab="Residuals",main="Fitted Value Pattern")

# ############# MODEL 3 ##################

df.lm3<-lm(Data~(A+B+C+D+E)^2,d=df)
anova(df.lm3)
hist(df.lm3$residuals,main="Histogram ofresiduals for Linear Model 3" , xlab = "Residuals")
shapiro.test(df.lm3$residuals)
qqnorm(df.lm3$residuals)
qqline(df.lm3$residuals)
plot(jitter(df.lm3$fit),df.lm3$residuals,xlab="Fitted Values" ,ylab="Residuals" , main=" Fitted Value Pattern")
stdres<-rstandard(df.lm3)
plot(stdres,ylim=c(-3,3), ylab="Standardized Residuals" , main="Standardized residuals distribution")
##1 sigma
abline(h=1, col="green")
abline(h=-1,col="green")
##2 sigma
abline(h=2,col="orange")
abline(h=-2,col="orange")
##3 sigma
abline(h=3,col="red")
abline(h=-3,col="red" )
plot(density(stdres),ylim=c(0,0.7),main="Standardized Residuals PDF" )
curve(dnorm(x),add=T,col="red")

########### MODEL 4 ###################
# Fourth model.
df.lm4<-lm(Data~((A+B+C+D+E)^3),d=df)
anova(df.lm4)
hist(df.lm4$residuals,main="Histogram of residuals for Linear Model 4" , xlab = "Residuals")
shapiro.test(df.lm4$residuals)
qqnorm(df.lm4$residuals)
qqline(df.lm4$residuals)
plot(jitter(df.lm4$fit),df.lm4$residuals,xlab="Fitted Values ",ylab="Residuals",main="Fitted Value Pattern")
stdres4<-rstandard(df.lm4)
plot(stdres4,ylim=c(-3,3),ylab="Standardized Residuals",main="Standardized residuals distribution")
#1 sigma
abline(h=1,col="green")
abline(h=-1,col="green")
#2 sigma
abline(h=2,col="orange")
abline(h=-2,col="orange")
#3 sigma
abline(h=3,col="red")
abline(h=-3,col="red")
plot(density(stdres4),ylim=c(0,0.4),main="Standardized Residuals PDF" )
curve(dnorm(x),add=T,col="red")


# BoxCox
w<-df[1:32,1]
m<-df[1:32,2]
j<-df[1:32,3]
t<-df[1:32,4]
g<-df[1:32,5]
y<-data
x<-(w+m+j+t+g)^3
#run the box-cox transformation
bc<-boxcox(y~x)
(lambda<-bc$x[which.max(bc$y)])
#final model
df.lm7<-lm(Data^(lambda)~(A+B+C+D+E)^3, d=df)
#NORMALITY CHECKs (MAC)
# Shapiro milk test to check the normality
hist(df.lm7$res, xlab="Residuals",main ="Hist of Residuals")
shapiro.test(df.lm7$res)
# add noise
plot(jitter(df.lm7$fitted),df.lm7$res)
#QQ plot to check the normality of the residuals is fair
qqnorm(df.lm7$residuals)
qqline(df.lm7$residuals)
#anova
anova(df.lm7)


########### TESTING ###################
# Fourth model without E factor.
df.lm5<-lm(Data~((A+B+C+D)^3),d=df)
anova(df.lm5)
shapiro.test(df.lm5$residuals)

# Third model whitout E factor.
df.lm6<-lm(Data~((A+B+C+D)^2),d=df)
anova(df.lm6)
shapiro.test(df.lm6$residuals)

########### INTERACTION PLOTS ####################

interaction.plot(df$A,df$B,data, xlab="Processor", ylab= "data",trace.label="Language")
interaction.plot(df$A,df$E,data, xlab="Processor", ylab= "data",trace.label="Datatype")
interaction.plot(df$B,df$E,data, xlab="Programming Language", ylab= "data",trace.label="Datatype")
```

```
interaction.plot(df$C,df$D,data, xlab="Length of the Vector", ylab= "data",trace.label="Dimensions")


########## CONTOUR PLOTS ###################
library(lattice)
CPU<-seq(-1,1,0.1)
Length<-seq(-1,1,0.1)
g <- expand.grid(A = CPU,B=df$B, C =Length, D = df$D,E=df$E)
g$data <- predict(df.lm7,g)

contourplot(data~A*C, data = g[g$B==1,], cuts = 10, region = T, xlab = "CPU", ylab = "Length", main = "Contourplot Matlab")
contourplot(data~A*C, data = g[g$B==-1,], cuts = 10, region = T, xlab = "CPU", ylab = "Length", main = "Contourplot C++")
```