Atividade Pratica 9 de Estatistica Aplicada

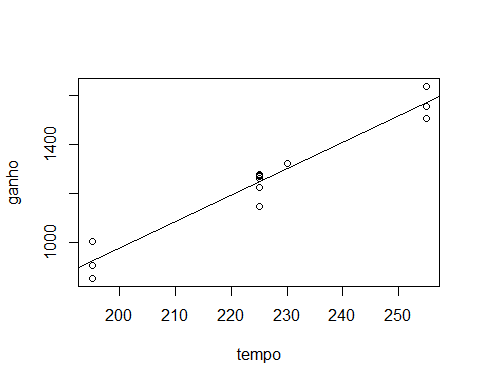
Andre Filipe Queiroz de Melo e Soares

28 de Novembro de 2018

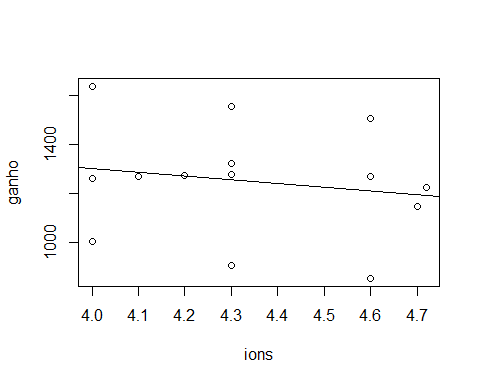
## QUESTAO 1

# A)

#QUESTAO 1  
  
  
#A)  
dados <- read.csv("tabela.csv", header=TRUE,sep=",",dec=",")  
attach(dados)  
  
graf = plot(tempo,ganho,pch = 1)  
abline(lm(ganho~tempo), col="black")



graf = plot(ions,ganho,pch = 1)  
abline(lm(ganho~ions), col="black")



# B)

dados <- read.csv("tabela.csv", header=TRUE,sep=",",dec=",")  
attach(dados)

## The following objects are masked from dados (pos = 3):  
##   
## ganho, ions, observacao, tempo

#B)  
  
correlacao = cor(x = tempo,y = ganho,method = "pearson")  
print(correlacao)

## [1] 0.9737924

correlacao = cor(x = ions,y = ganho,method = "pearson")  
print(correlacao)

## [1] -0.1802233

## C)

dados <- read.csv("tabela.csv", header=TRUE,sep=",",dec=",")  
attach(dados)

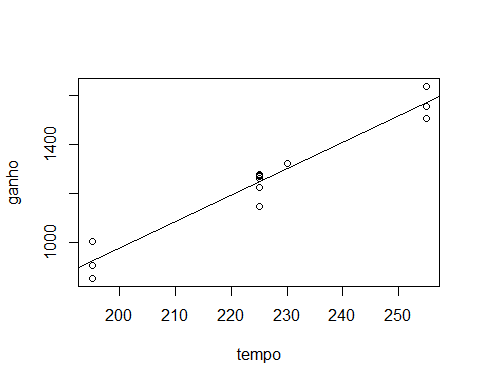
## The following objects are masked from dados (pos = 3):  
##   
## ganho, ions, observacao, tempo

## The following objects are masked from dados (pos = 4):  
##   
## ganho, ions, observacao, tempo

modelo1 <- lm(ganho~tempo)  
#ANOVA 1  
anova(modelo1)

## Analysis of Variance Table  
##   
## Response: ganho  
## Df Sum Sq Mean Sq F value Pr(>F)   
## tempo 1 630968 630968 219.98 4.421e-09 \*\*\*  
## Residuals 12 34419 2868   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(ganho~tempo)  
abline(modelo1)



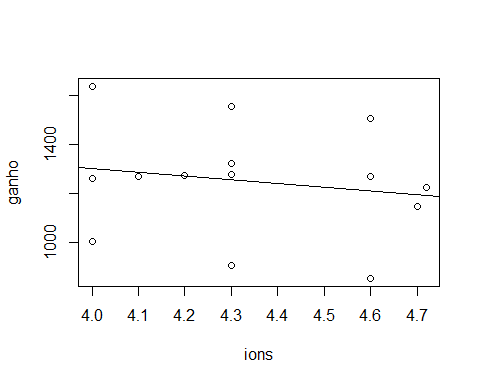
summary(modelo1)

##   
## Call:  
## lm(formula = ganho ~ tempo)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -99.79 -20.39 17.74 25.71 81.80   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1181.1422 164.5143 -7.18 1.12e-05 \*\*\*  
## tempo 10.7864 0.7272 14.83 4.42e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 53.56 on 12 degrees of freedom  
## Multiple R-squared: 0.9483, Adjusted R-squared: 0.944   
## F-statistic: 220 on 1 and 12 DF, p-value: 4.421e-09

modelo2 <- lm(ganho~ions)  
#ANOVA 2  
anova(modelo2)

## Analysis of Variance Table  
##   
## Response: ganho  
## Df Sum Sq Mean Sq F value Pr(>F)  
## ions 1 21612 21612 0.4029 0.5375  
## Residuals 12 643775 53648

plot(ganho~ions)  
abline(modelo2)



summary(modelo2)

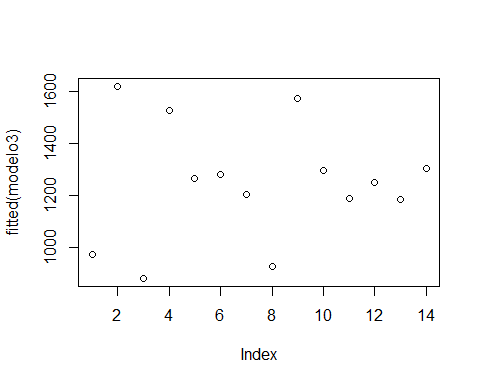
##   
## Call:  
## lm(formula = ganho ~ ions)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -357.07 -46.15 10.91 64.20 334.32   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1919.1 1056.6 1.816 0.0944 .  
## ions -154.4 243.2 -0.635 0.5375   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 231.6 on 12 degrees of freedom  
## Multiple R-squared: 0.03248, Adjusted R-squared: -0.04815   
## F-statistic: 0.4029 on 1 and 12 DF, p-value: 0.5375

modelo3 <- lm(ganho~tempo+ions)  
#ANOVA   
anova(modelo3)

## Analysis of Variance Table  
##   
## Response: ganho  
## Df Sum Sq Mean Sq F value Pr(>F)   
## tempo 1 630968 630968 517.14 1.342e-10 \*\*\*  
## ions 1 20998 20998 17.21 0.00162 \*\*   
## Residuals 11 13421 1220   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(fitted(modelo3))  
abline(modelo3)

## Warning in abline(modelo3): only using the first two of 3 regression  
## coefficients



summary(modelo3)

##   
## Call:  
## lm(formula = ganho ~ tempo + ions)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -44.584 -24.565 -3.266 22.330 63.201   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -520.0767 192.1071 -2.707 0.02039 \*   
## tempo 10.7812 0.4743 22.730 1.35e-10 \*\*\*  
## ions -152.1489 36.6754 -4.149 0.00162 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 34.93 on 11 degrees of freedom  
## Multiple R-squared: 0.9798, Adjusted R-squared: 0.9762   
## F-statistic: 267.2 on 2 and 11 DF, p-value: 4.742e-10

## D)

#Anova permite avaliar afirmações sobre as médias de populações. A análise visa, fundamentalmente,  
#verificar se existe uma diferença significativa. Quanto maior for R2, que   
#representa o quão bem o modelo prediz as respostas, maior capacidade de predizer tem o modelo .  
#Dos 3 modelos acima, o que tem maior valor é o 3º, tempo e dose de Íons. Logo,   
#o modelo que mais se ajusta aos dados foi o terceiro.

## QUESTAO 2

# A)

info <- read.csv("tabela2.csv", header=TRUE,sep=",",dec=".")  
attach(info)  
  
Y = cbind(Indice)  
Y

## Indice  
## [1,] 8.1  
## [2,] 6.8  
## [3,] 7.0  
## [4,] 7.4  
## [5,] 7.7  
## [6,] 7.5  
## [7,] 7.6  
## [8,] 8.0

X = cbind(rep(1,8), Num.Ataques, Duracao)  
X

## Num.Ataques Duracao  
## [1,] 1 5 118  
## [2,] 1 13 132  
## [3,] 1 20 119  
## [4,] 1 28 153  
## [5,] 1 41 91  
## [6,] 1 49 118  
## [7,] 1 61 132  
## [8,] 1 62 105

# B)

info <- read.csv("tabela2.csv", header=TRUE,sep=",",dec=".")  
attach(info)

## The following objects are masked from info (pos = 3):  
##   
## Duracao, Esquema, Indice, Num.Ataques

Y = cbind(Indice)  
Y

## Indice  
## [1,] 8.1  
## [2,] 6.8  
## [3,] 7.0  
## [4,] 7.4  
## [5,] 7.7  
## [6,] 7.5  
## [7,] 7.6  
## [8,] 8.0

X = cbind(rep(1,8), Num.Ataques, Duracao)  
X

## Num.Ataques Duracao  
## [1,] 1 5 118  
## [2,] 1 13 132  
## [3,] 1 20 119  
## [4,] 1 28 153  
## [5,] 1 41 91  
## [6,] 1 49 118  
## [7,] 1 61 132  
## [8,] 1 62 105

beta = solve(t(X) %\*% X) %\*% (t(X) %\*% Y)  
print(beta)

## Indice  
## 8.372602064  
## Num.Ataques 0.005095369  
## Duracao -0.008576885

# C)

variavel = lm(Indice ~ Num.Ataques + Duracao)  
print(variavel)

##   
## Call:  
## lm(formula = Indice ~ Num.Ataques + Duracao)  
##   
## Coefficients:  
## (Intercept) Num.Ataques Duracao   
## 8.372602 0.005095 -0.008577

#Reta pelo beta chapeu:   
 Y = 8.372602064 + 0.005095369 \* Num.Ataques - 0.008576885 \* Duracao   
 Y

## [1] 7.386006 7.306693 7.453860 7.203009 7.801016 7.610203 7.551271 7.787942

#Reta pelo lm():   
 Y = 8.372602 + 0.005095 \* Num.Ataques - 0.008577 \* Duracao  
 Y

## [1] 7.385991 7.306673 7.453839 7.202981 7.800990 7.610171 7.551233 7.787907

# D)

eruption.lm = lm(Num.Ataques ~ Duracao, data=faithful)  
summary(eruption.lm)$r.squared

## [1] 0.06330766

eruption.lm2 = lm(Num.Ataques ~ Indice, data=faithful)  
summary(eruption.lm2)$r.squared

## [1] 0.1131025

eruption.lm3 = lm(Indice ~ Duracao, data=faithful)  
summary(eruption.lm3)$r.squared

## [1] 0.1757661

#R², é uma medida de ajustamento de um modelo estatístico linear generalizado, em relação aos valores observados.   
#O R² varia entre 0 e 1, indicando, em percentagem, o quanto o modelo consegue explicar os   
#valores observados. Quanto maior o R², mais explicativo é o modelo, melhor ele se ajusta à amostra.

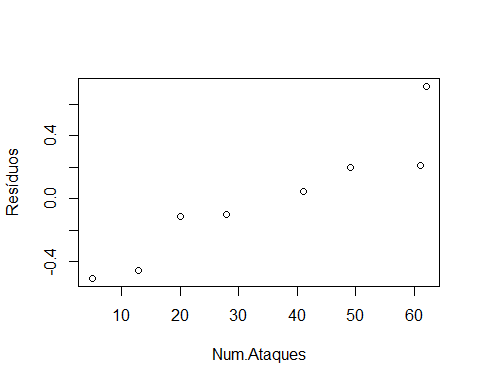
# E)

Previsao = 8.372602 + 0.005095 \* 25 - 0.008577 \* 100  
Previsao

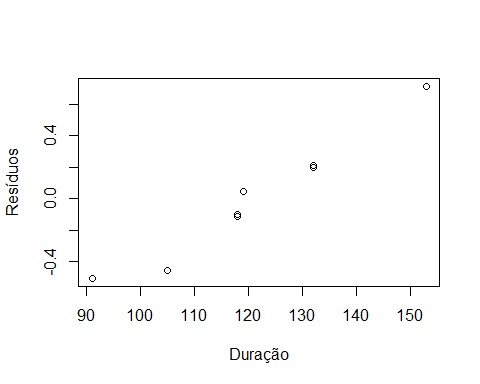
## [1] 7.642277

# F)

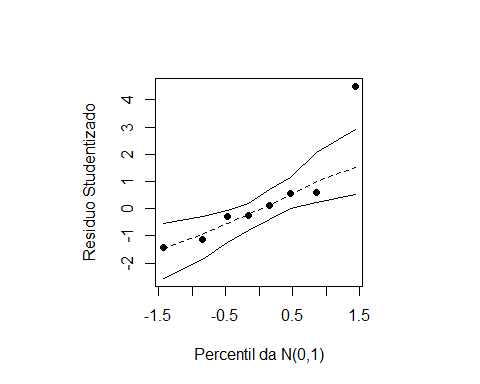
# Gráficos Q-Q Plot  
variavel = lm(Indice ~ Num.Ataques + Duracao)  
  
qqplot(Num.Ataques, residuals(variavel), xlab="Num.Ataques",ylab="Resíduos")



qqplot(Duracao, residuals(variavel), xlab="Duração",ylab="Resíduos")



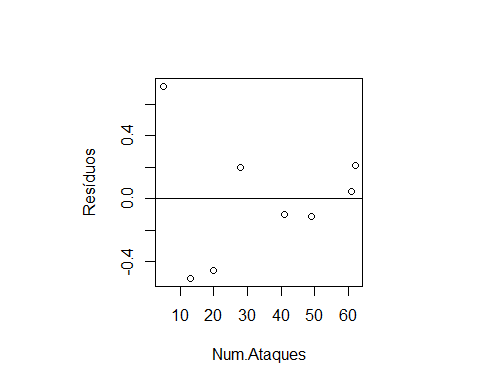
# Gráfico de Envelope  
fit.model <- variavel   
par(mfrow=c(1,1))  
X <- model.matrix(fit.model)  
n <- nrow(X)  
p <- ncol(X)  
H <- X%\*%solve(t(X)%\*%X)%\*%t(X)  
h <- diag(H)  
si <- lm.influence(fit.model)$sigma  
r <- resid(fit.model)  
tsi <- r/(si\*sqrt(1-h))  
ident <- diag(n)  
epsilon <- matrix(0,n,100)  
e <- matrix(0,n,100)  
e1 <- numeric(n)  
e2 <- numeric(n)  
for(i in 1:100){  
 epsilon[,i] <- rnorm(n,0,1)  
 e[,i] <- (ident - H)%\*%epsilon[,i]  
 u <- diag(ident - H)  
 e[,i] <- e[,i]/sqrt(u)  
 e[,i] <- sort(e[,i]) }  
for(i in 1:n){  
 eo <- sort(e[i,])  
 e1[i] <- (eo[2]+eo[3])/2  
 e2[i] <- (eo[97]+eo[98])/2 }  
med <- apply(e,1,mean)  
faixa <- range(tsi,e1,e2)  
par(pty="s")  
qqnorm(tsi,xlab="Percentil da N(0,1)",  
 ylab="Residuo Studentizado", ylim=faixa, pch=16, main="")  
par(new=TRUE)  
qqnorm(e1,axes=F,xlab="",ylab="",type="l",ylim=faixa,lty=1, main="")  
par(new=TRUE)  
qqnorm(e2,axes=F,xlab="",ylab="", type="l",ylim=faixa,lty=1, main="")  
par(new=TRUE)  
qqnorm(med,axes=F,xlab="",ylab="",type="l",ylim=faixa,lty=2, main="")



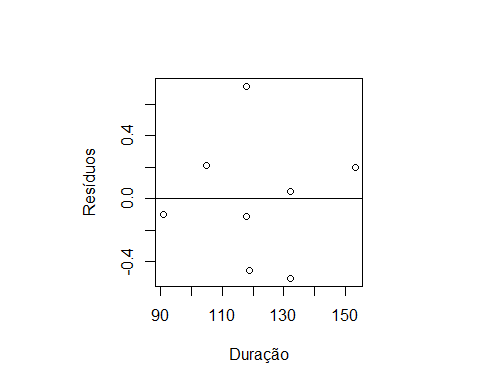
# Teste de Kolmogorv-Smirnov  
ks.test(residuals(variavel), "pnorm")

##   
## One-sample Kolmogorov-Smirnov test  
##   
## data: residuals(variavel)  
## D = 0.30619, p-value = 0.3657  
## alternative hypothesis: two-sided

#p-valor(0.3657) > 0.05 # logo os resíduos seguem distribuição normal.  
  
  
# Homocedasticidade  
plot(Num.Ataques, residuals(variavel),xlab="Num.Ataques",ylab="Resíduos")  
abline(h=0)



plot(Duracao, residuals(variavel),xlab="Duração",ylab="Resíduos")  
abline(h=0)



# Por causa do comportamento observado no gráfico , percebemos indícios que podem nos levar à rejeição de homocedasticidade do modelo.

```