**Escola Politécnica de Pernambuco**

*Especialização em Ciência de Dados e Analytics*

*Laís Ingrid Soares de Souza Vidoto*

**Fatores que influenciam o consumo de combustível**

**Estatística Computacional**

Recife

12/06/2019

base de dados

url-> http://archive.ics.uci.edu/ml/datasets/Auto+MPG

descrição→ é uma base responsável por realizar uma analise dos carros, em relação á seu consumo de gasolina

Atributos:

|  |  |  |  |
| --- | --- | --- | --- |
| Nome campo | Tipo | Descrição | Utilização |
| Mpg | contínuo | Consumo de combustível do ciclo urbano em milhas por galão | Será considerada |
| Cilindros | discreto com vários valores | Força mecânica que ajuda o deslocamento do carro | Será considerada |
| Deslocamento | contínuo | Mudança de um lugar para outro | Será considerada |
| **cavalo-vapor** | **contínuo** | **Força do motor** | **Não será considerada** |
| Peso | contínuo | Peso total do carro | Será considerada |
| Aceleração | contínuo | Velocidade máxima do carro | Será considerada |
| Ano modelo | discreto com vários valores | Ano de fabricação do modelo do carro | Será considerada |
| Origem | discreto com vários valores | Mesma montadora de origem | Será considerada |
| **Nome do carro** | **string** | **Como o carro será chamado** | **Não será considerada** |

Conclusões após análise dos dados

Referente a Interpretação de B

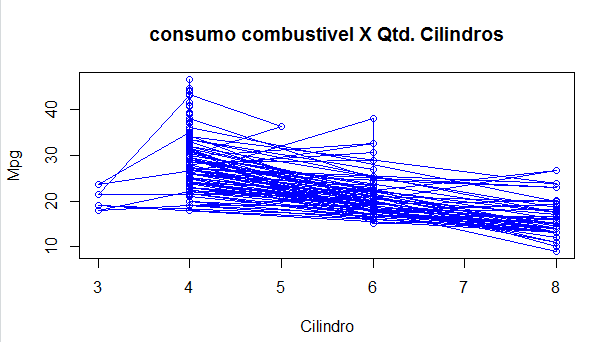
Aumentando a quantidade de cilindros(X),o consumo de gasolina mpg(Y) aumenta, em media, 2,60 por litro, e com 1 litro em X, Y será de 17,04, já que sua amplitude é de 16,78.

y = a + (b \* x)→ (16,78 + (0,26 \* 1) )→16,78 + (0,26)→17,04 litros

Maiores valores para x, são dos atributos: cilindros, aceleração e ano Modelo segundo á coleta de dados da api.

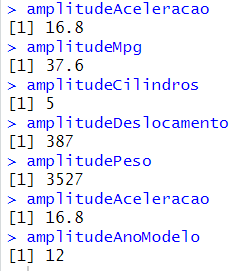
O Y é o atributo: Mpg consumo de gasolina

Gráfico de relação entre o consumo de combustível com á quantidade de Cilindros composta no carro.

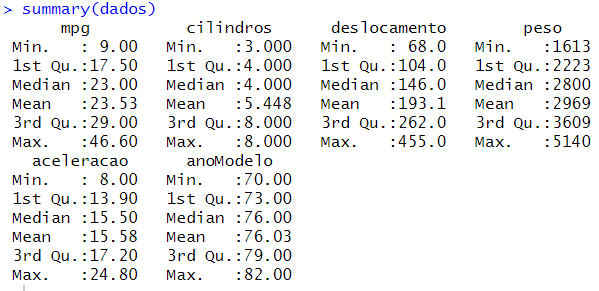


análises deverão ser feitas

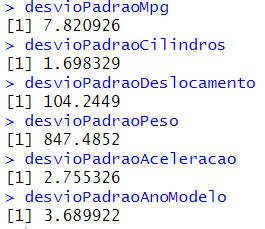
Identificar a amplitude de cada uma das variáveis;

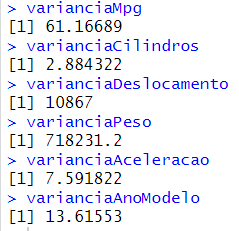


Calcular média, moda e mediana para cada uma das variáveis numéricas;

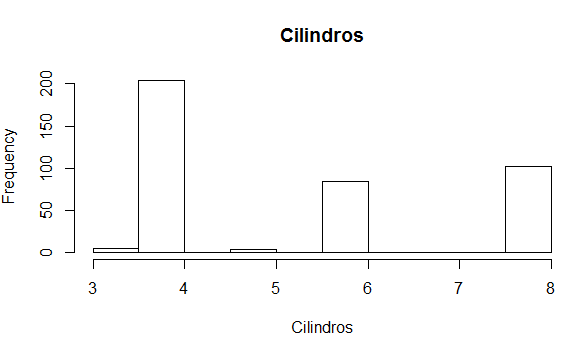


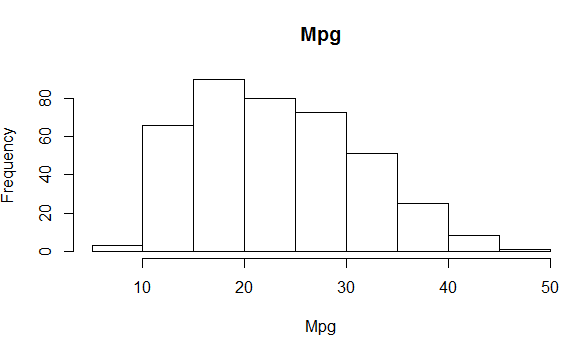
Calcular variância e desvio padrão para cada uma das variáveis;

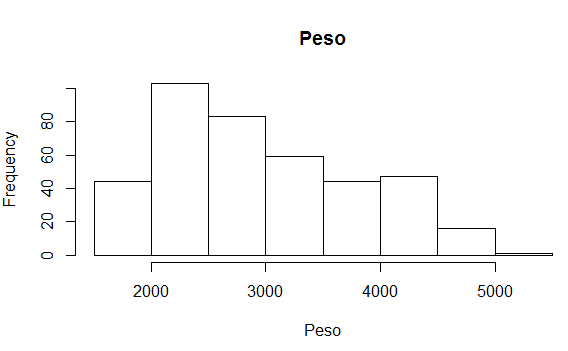


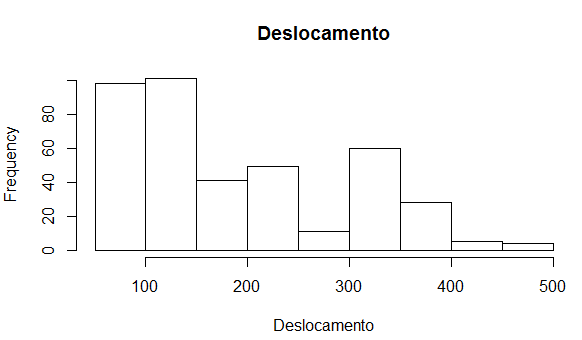


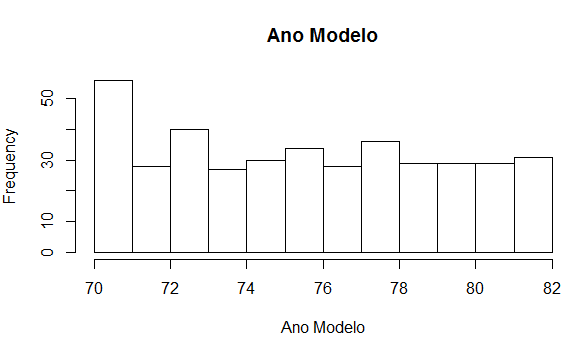
Plotar o melhor gráfico para cada uma das variáveis;

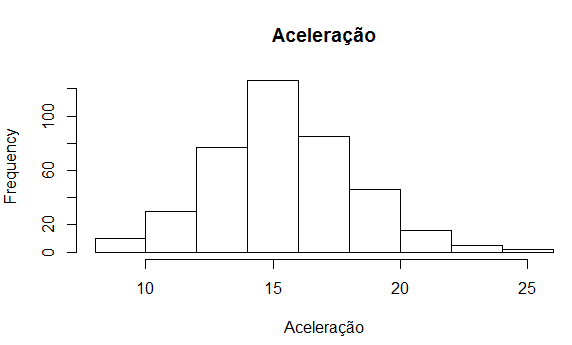




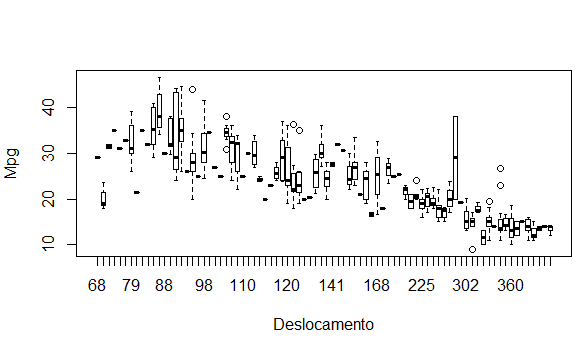
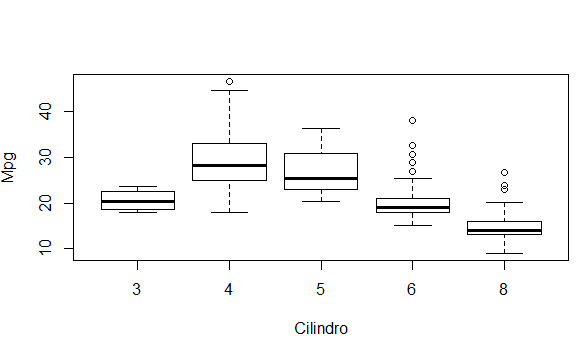


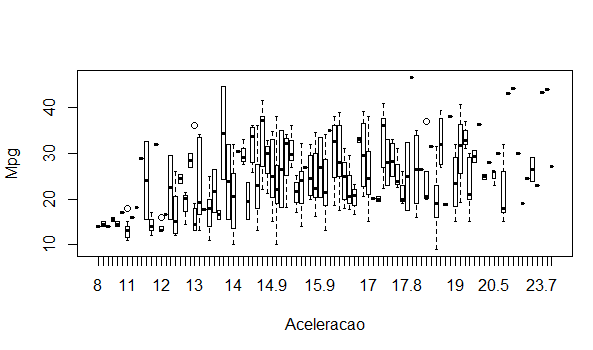
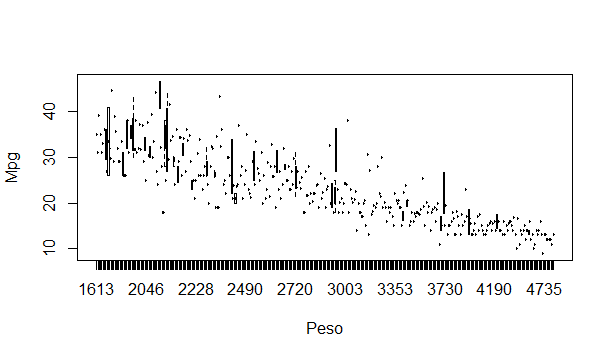


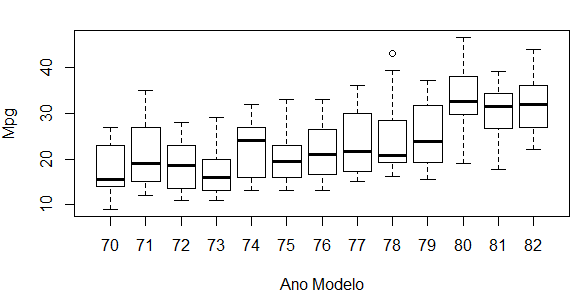




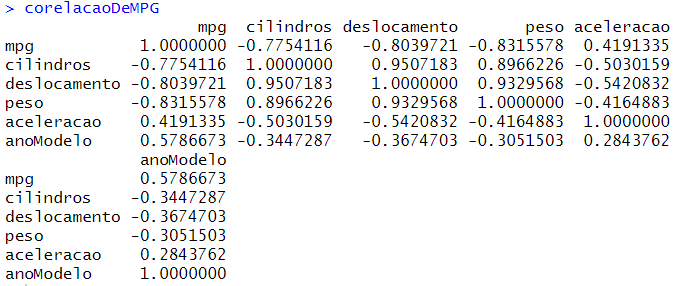
Plotar o boxplot para todas as variáveis quantitativas;

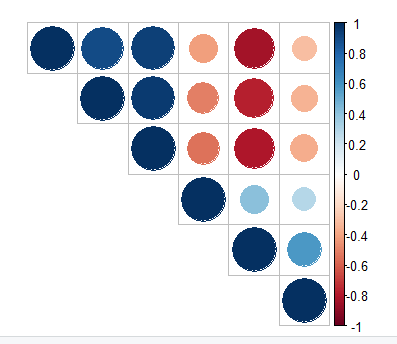




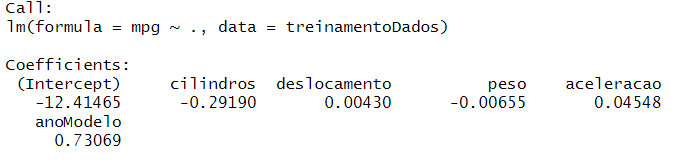


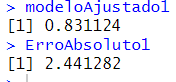
Calcular a correlação e plotar o gráfico de dispersão;

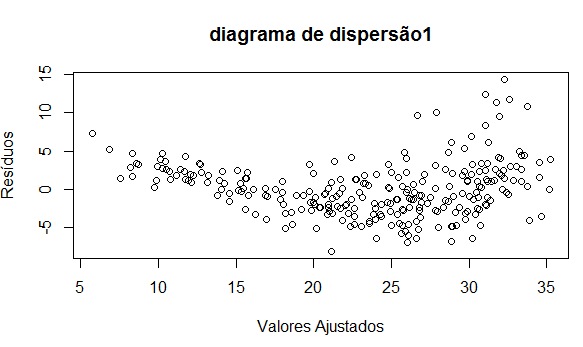
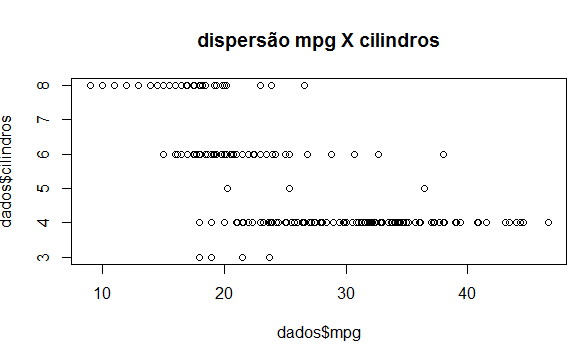


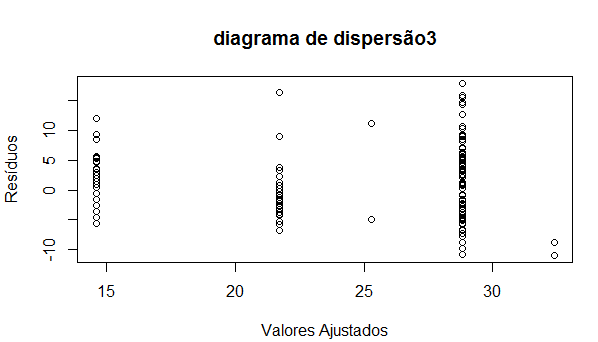


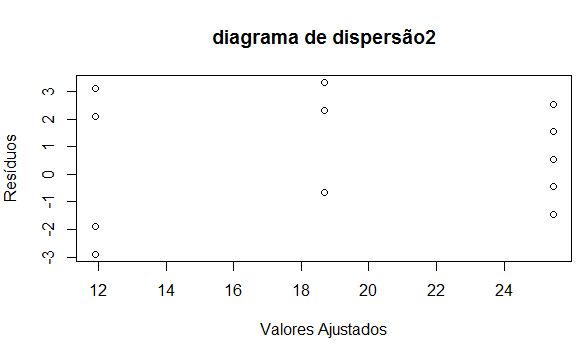
Formular um problema de regressão e análisa-lo como também discuti-lo;



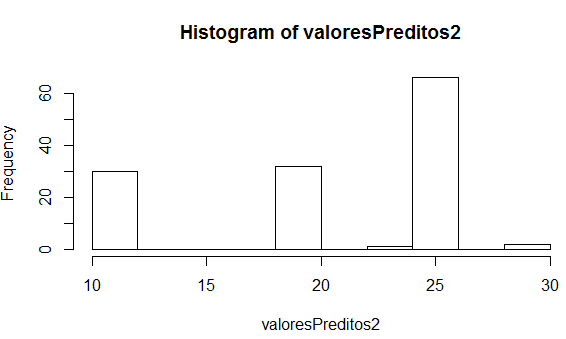
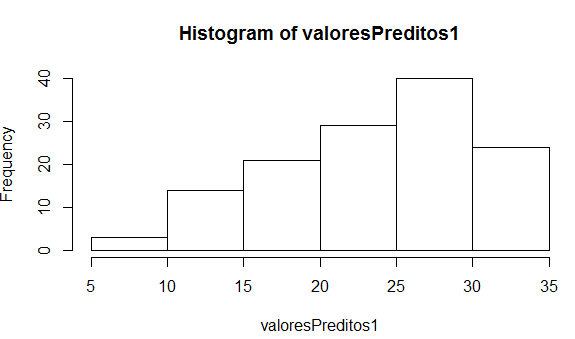


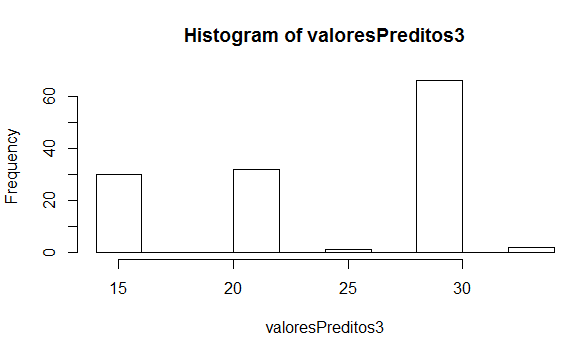




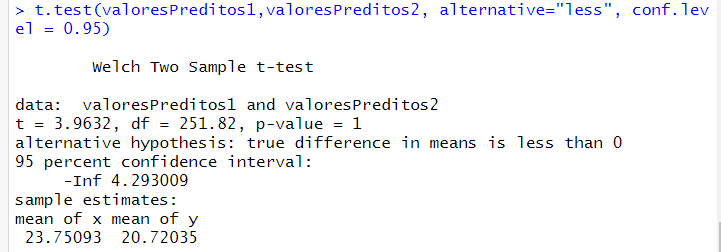


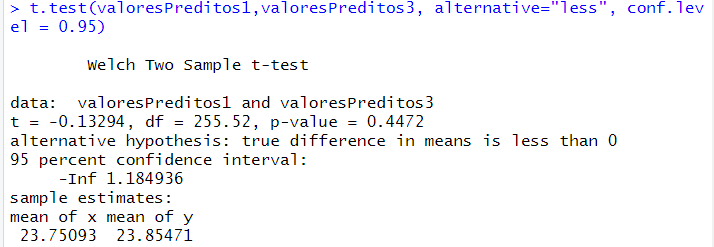
Realizar teste de normalidade usando boxplot, histograma e shapiro.test, discutir os resultados;

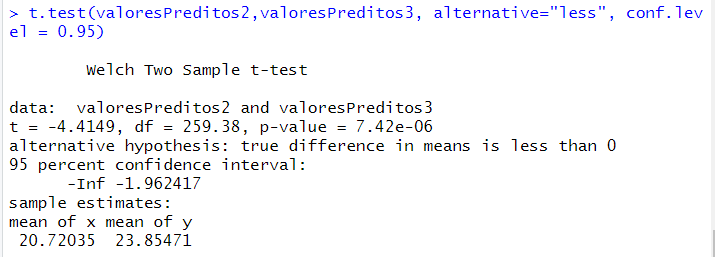


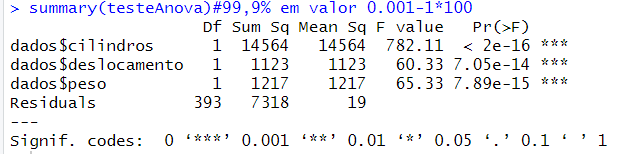


Formular teste de hipótese e Anova, discutir os resultados;









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

# Leitura de Arquivos txt

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

getwd()

setwd("C:/Users/UpePos/Desktop/Nova pasta/upe/05 - Estatistica Computacional")

leituraArquivo = read.table("auto-mpg.txt", header = TRUE)

colnames(leituraArquivo) = c("mpg", "cilindros", "deslocamento", "cavaloVapor", "peso", "aceleracao", "anoModelo","origem", "nomeCarro" )

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

# trazendo apenas as colunas que eu queiro

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

dados = cbind.data.frame(leituraArquivo$mpg,leituraArquivo$cilindro,leituraArquivo$deslocamento,leituraArquivo$peso,leituraArquivo$aceleracao,leituraArquivo$anoModelo)

colnames(dados) = c("mpg", "cilindros", "deslocamento", "peso", "aceleracao", "anoModelo")

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

# amplitude de cada variavel quantitativa

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

amplitudeMpg = ( range(dados$mpg)[2] - range(dados$mpg)[1] )

amplitudeCilindros = ( range(dados$cilindros)[2] - range(dados$cilindros)[1] )

amplitudeDeslocamento = ( range(dados$deslocamento)[2] - range(dados$deslocamento)[1] )

amplitudePeso = ( range(dados$peso)[2] - range(dados$peso)[1] )

amplitudeAceleracao = ( range(dados$aceleracao)[2] - range(dados$aceleracao)[1] )

amplitudeAnoModelo = ( range(dados$anoModelo)[2] - range(dados$anoModelo)[1] )

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

# min,max,mediana,media dos dados

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

summary(dados)

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

# variançia

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

var(dados)

varianciaMpg = var(dados$mpg)

varianciaCilindros = var(dados$cilindros)

varianciaDeslocamento = var(dados$deslocamento)

varianciaPeso = var(dados$peso)

varianciaAceleracao = var(dados$aceleracao)

varianciaAnoModelo = var(dados$anoModelo)

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

# desvio padrao

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

desvioPadraoMpg = sd(dados$mpg)

desvioPadraoCilindros = sd(dados$cilindros)

desvioPadraoDeslocamento = sd(dados$deslocamento)

desvioPadraoPeso = sd(dados$peso)

desvioPadraoAceleracao = sd(dados$aceleracao)

desvioPadraoAnoModelo = sd(dados$anoModelo)

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

# o melhor grafico para cada variavel quantitativa

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

plot(dados$cilindros, dados$mpg, main = "consumo combustivel X Qtd. Cilindros", type="o", col="blue",lwd=1, xlab="Cilindro", ylab="Mpg")

hist(dados$mpg, main="Mpg", xlab = "Mpg")

hist(dados$cilindros, main="Cilindros", xlab = "Cilindros")

hist(dados$deslocamento, main="Deslocamento", xlab = "Deslocamento")

hist(dados$peso, main="Peso", xlab = "Peso")

hist(dados$aceleracao, main="Aceleração", xlab = "Aceleração")

hist(dados$anoModelo, main="Ano Modelo", xlab = "Ano Modelo")

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

# graficos das variaveis quantitativas

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

boxplot(dados$mpg ~ dados$cilindros, xlab = "Cilindro", ylab = "Mpg")

boxplot(dados$mpg ~ dados$deslocamento, xlab = "Deslocamento", ylab = "Mpg")

boxplot(dados$mpg ~ dados$peso, xlab = "Peso", ylab = "Mpg")

boxplot(dados$mpg ~ dados$aceleracao, xlab = "Aceleracao", ylab = "Mpg")

boxplot(dados$mpg ~ dados$anoModelo, xlab = "Ano Modelo", ylab = "Mpg")

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

# Correlação

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

library(corrplot)

corelacaoDeMPG = cor(dados)

corrplot(corelacaoDeMPG, method="circle",type = "upper", tl.pos = "td",tl.cex = 0.5,tl.col = 'brank',order = "hclust")

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

# partição dos dados em teste e treinamento

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

library(lattice)

library(ggplot2)

library(caret)

indice = createDataPartition(dados$mpg, p=2/3, list = FALSE)

treinamentoDados = dados[indice,]

testeDados = dados[-indice,]

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

# Modelo de Regressão 1 - comparação do mpg com todos os dados

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

modelo1 = lm(mpg ~ ., data = treinamentoDados)

valoresPreditos1 = predict(modelo1, newdata = data.frame(testeDados) )

modeloAjustado1 = R2(valoresPreditos1, testeDados$mpg)

ErroAbsoluto1 = MAE(valoresPreditos1, testeDados$mpg)

x=dados$cilindros[1]

Yestimado = modelo1$coefficients[1] + modelo1$coefficients[2]\*x #-> para deslocamento

plot(dados$mpg, dados$cilindros, main = "dispersão mpg X cilindros")

abline(modelo1)

plot(fitted(modelo1), main = "diagrama de dispersão1", residuals(modelo1), xlab="Valores Ajustados", ylab="Resíduos")

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

# Modelo de Regressão 2 - 3 maiores

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

modelo2 = lm(mpg ~ cilindros,aceleracao,anoModelo, data = treinamentoDados)

valoresPreditos2 = predict(modelo2, newdata = data.frame(testeDados) )

modeloAjustado2 = R2(valoresPreditos2, testeDados$mpg)

ErroAbsoluto2 = MAE(valoresPreditos2, testeDados$mpg)

plot(dados$mpg, dados$cilindros,main = "D")

abline(modelo2)

plot(fitted(modelo2), main = "diagrama de dispersão2", residuals(modelo2), xlab="Valores Ajustados", ylab="Resíduos")

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

# Modelo de Regressão 3 - o maior

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

modelo3 = lm(mpg ~ cilindros, data = treinamentoDados)

valoresPreditos3 = predict(modelo3, newdata = data.frame(testeDados) )

modeloAjustado3 = R2(valoresPreditos3, testeDados$mpg)

ErroAbsoluto3 = MAE(valoresPreditos3, testeDados$mpg)

plot(dados$mpg, dados$cilindros)

abline(modelo3)

plot(fitted(modelo3), main = "diagrama de dispersão3", residuals(modelo3), xlab="Valores Ajustados", ylab="Resíduos")

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

# teste de normalidade

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

shapiroNormalidadeY1 = shapiro.test(valoresPreditos1)

ks.test(valoresPreditos1, "pnorm", mean(valoresPreditos1), sd(valoresPreditos1))

hist(valoresPreditos1)

shapiroNormalidadeY2 = shapiro.test(valoresPreditos2)

ks.test(valoresPreditos2, "pnorm", mean(valoresPreditos2), sd(valoresPreditos2))

hist(valoresPreditos2)

shapiroNormalidadeY3 = shapiro.test(valoresPreditos3)

ks.test(valoresPreditos3, "pnorm", mean(valoresPreditos3), sd(valoresPreditos3))

hist(valoresPreditos3)

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

# teste de hipotese

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

t.test(valoresPreditos1,valoresPreditos2, alternative="less", conf.level = 0.95)

t.test(valoresPreditos1,valoresPreditos3, alternative="less", conf.level = 0.95)

t.test(valoresPreditos2,valoresPreditos3, alternative="less", conf.level = 0.95)

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

# teste de Anova

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

testeAnova = aov(dados$mpg ~ dados$cilindros + dados$deslocamento + dados$peso)

summary(testeAnova)#99,9% em valor 0.001-1\*100