

# SME0820 - Modelos de Regressão e Aprendizado Supervisionado I - Trabalho I

Brenda da Silva Muniz 11811603      Francisco Rosa Dias de Miranda 4402962  
Heitor Carvalho Pinheiro 11833351      Mônica Amaral Novelli 11810453

Setembro 2021

Neste trabalho, nosso objetivo é ajustar um modelo de regressão linear simples ao conjunto de dados fornecido, utilizando linguagem R. Para esta tarefa, descreveremos cada etapa de nosso *pipeline*.

O dataset B.3 contém dados sobre o rendimento de Gasolina, em milhas, de 32 automóveis diferentes. Ajuste o modelo de regressão linear simples que relaciona o rendimento da gasolina (y) (Milhas por litro) e a cilindrada do motor (x1) (polegadas cúbicas). Use sempre Significância: 99%.

Primeiramente, vamos carregar os módulos utilizados nesta análise. Caso não possua algum dos pacotes, utilize o comando `install_packages("Nome_do_pacote")`.

```
library(tidyverse)
library(ggpubr)
library(corrplot)
library(DataExplorer)
library(GGally)
library(knitr)
library(data.table)
```

Com os pacotes carregados em nosso ambiente, lemos o arquivo `.csv` disponibilizado colocando-o na mesma pasta de nosso projeto. Vamos inspecionar o que foi carregado com auxílio do comando `head()`, que exibe as 5 primeiras observações.

```
dados <- read_csv("data-table-B3.csv", locale = locale(decimal_mark = ","))
```

```
head(dados) %>% kable(caption = "Cinco primeiras observações do dataset")
```

Table 1: Cinco primeiras observações do dataset

y	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
18.90	350	165	260	8.00	2.56	4	3	200.3	69.9	3910	1
17.00	350	170	275	8.50	2.56	4	3	199.6	72.9	3860	1
20.00	250	105	185	8.25	2.73	1	3	196.7	72.2	3510	1
18.25	351	143	255	8.00	3.00	2	3	199.9	74.0	3890	1
20.07	225	95	170	8.40	2.76	1	3	194.1	71.8	3365	0
11.20	440	215	330	8.20	2.88	4	3	184.5	69.0	4215	1

Vamos separar nossa base em treino e teste, onde guardaremos 4 observações para realizar a previsão mais tarde.

```
set.seed(42)
smp <- sample(32, 4)
treino <- dados[-smp,] %>% select(y, x1)
teste <- dados[smp,] %>% select(y, x1)
y <- treino$y
x1 <- treino$x1
n <- length(y)
```

## Parte a):

- Descrição do banco de dados
- Definição das variáveis
- Análise exploratória inicial

```
summary(dados %>% select(y,x1)) %>% kable(caption = "Sumário das variáveis utilizadas")
```

Table 2: Sumário das variáveis utilizadas

y	x1
Min. :11.20	Min. : 85.3
1st Qu.:16.48	1st Qu.:211.5
Median :19.30	Median :318.0
Mean :20.22	Mean :285.0
3rd Qu.:21.66	3rd Qu.:353.2
Max. :36.50	Max. :500.0

Com o comando **summary** verificamos as principais medidas descritivas para cada variável (feature) presente no nosso conjunto de dados. Temos 12 features e ajustaremos o modelo com base na feature x1.

## Dimensão dos dados

```
dim(dados) %>% kable(caption = "Dimensão dos dados")
```

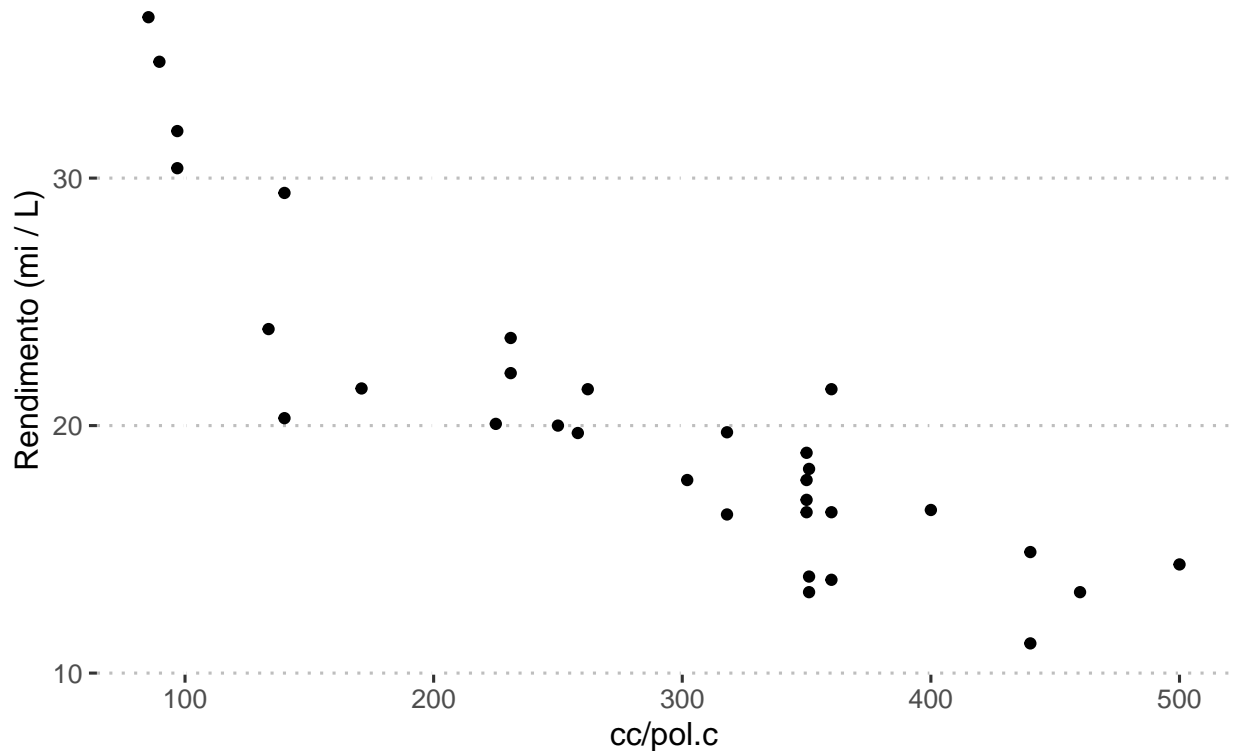
Table 3: Dimensão dos dados

x
32
12

## Análise Exploratória Básica

```
ggplot(dados, aes(x=dados$x1, y = dados$y)) + geom_point() + #geom_smooth(method = "lm") +
  ggtitle("Cilíndradas Vs Rendimento") + xlab("cc/pol.c") + ylab("Rendimento (mi / L)") +
  theme_pubclean() +
  theme(plot.title = element_text(size = 20, hjust = .5))
```

## Cilíndradas Vs Rendimento



A partir do gráfico de dispersão acima parece que existe uma relação linear entre as variáveis  $X_1$  e  $Y$ . Vamos verificar tal relação a partir do coeficiente de Pearson  $\rho$ .

**Teste de Correlação de Pearson entre as variáveis  $X_1$  e  $Y$**

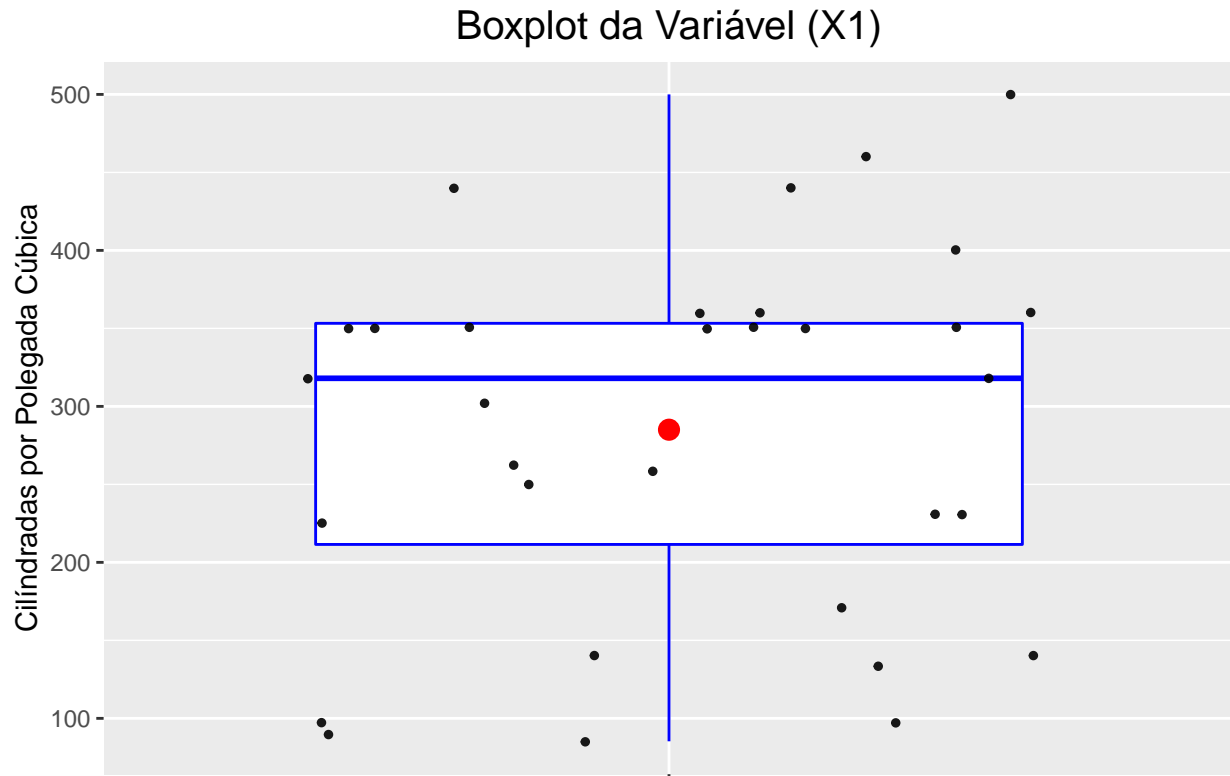
```
cor.test(dados$x1, dados$y)
```

```
##
## Pearson's product-moment correlation
##
## data:  dados$x1 and dados$y
## t = -10.086, df = 30, p-value = 3.743e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9395719 -0.7642987
## sample estimates:
##          cor
## -0.8787896
```

O teste de correlação de Person resultou em um  $\rho = -0.87$  o que indica uma forte correlação entre as variáveis. Nosso objetivo é estimar a influência de  $X_1$  sobre  $Y$ , ou seja, como  $Y$  varia em relação às variações em  $X_1$ .

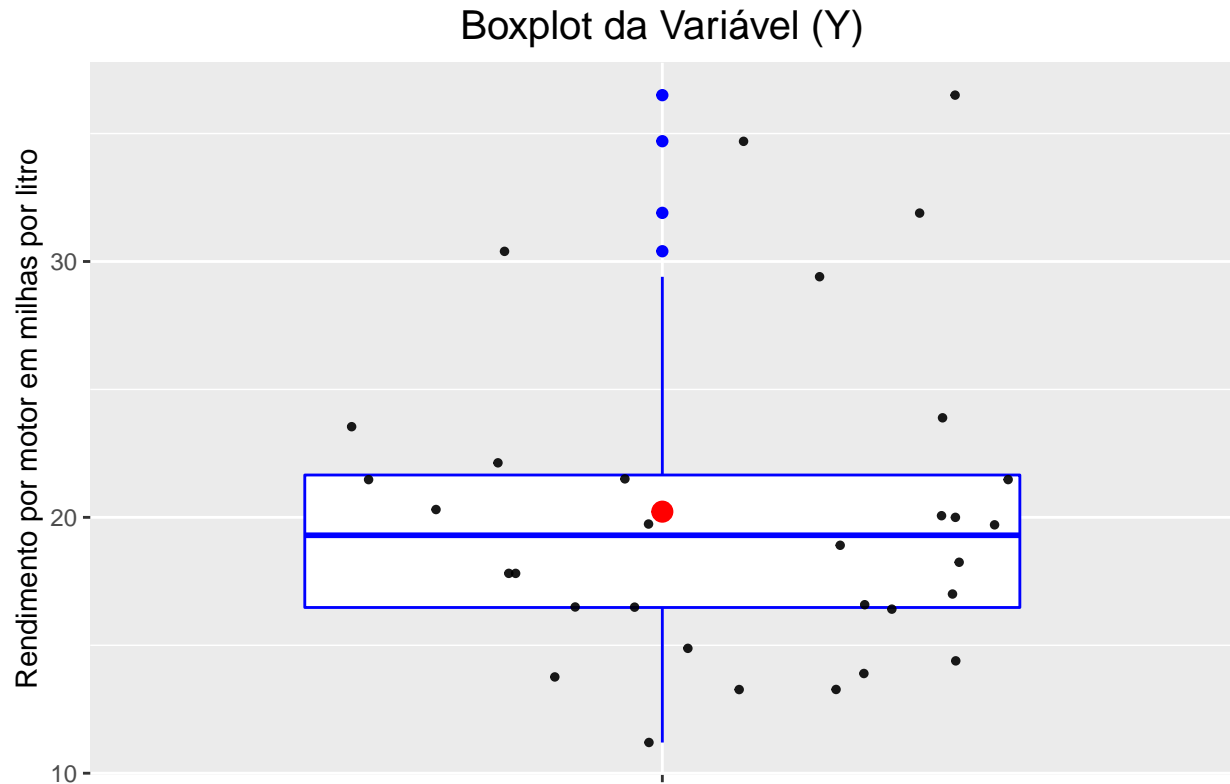
```
dados %>%
  ggplot(aes(x = "", y = dados$x1)) +
  geom_boxplot(color = "blue") +
```

```
stat_summary(fun = mean, geom = "point", shape = 20, size = 5, color = "red", fill = "red") +
geom_jitter(color="black", size=1, alpha=.9) +
theme(plot.title = element_text(size = 15, hjust = .5)) +
ggtitle("Boxplot da Variável (X1)") +
xlab("") + ylab("Cilíndradas por Polegada Cúbica")
```



Podemos perceber a partir do boxplot acima e da função summary que 50% dos carros tem menos de 318 cilíndradas.

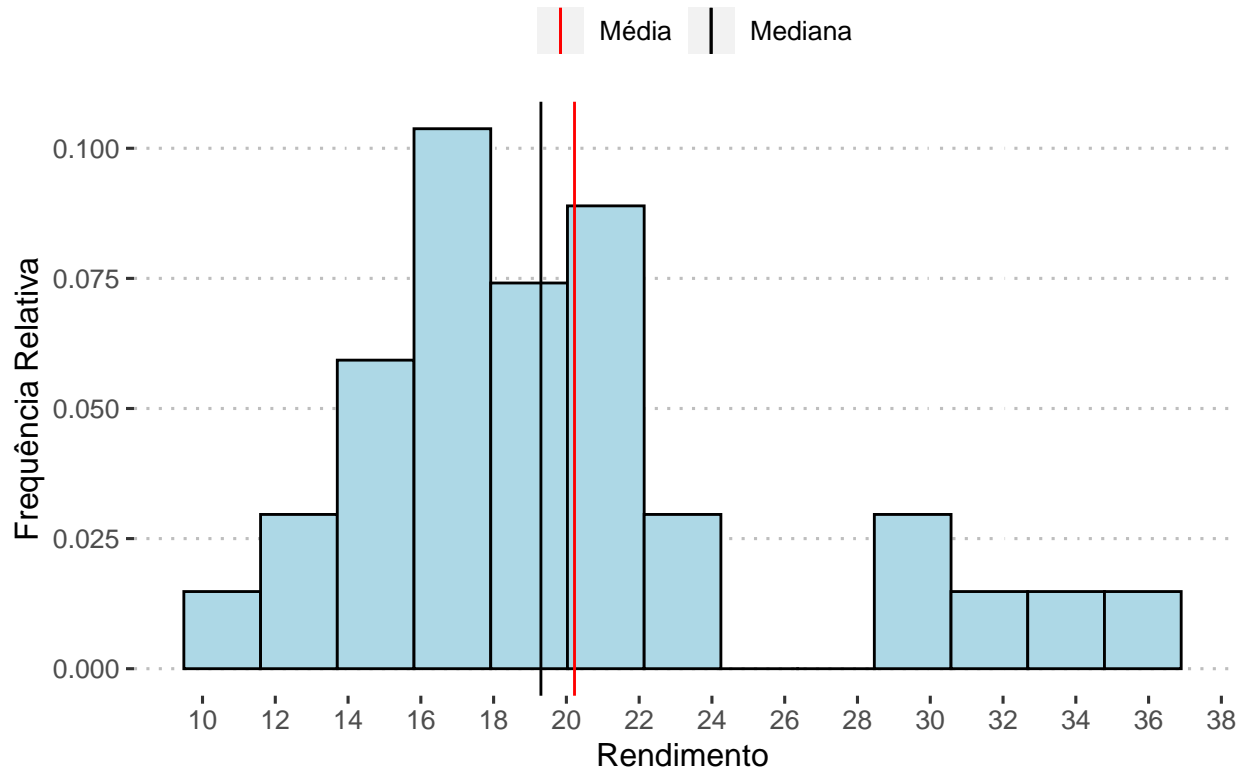
```
dados %>%
ggplot(aes(x = "", y = dados$y)) +
geom_boxplot(color = "blue") +
stat_summary(fun = mean, geom = "point", shape = 20, size = 5, color = "red", fill = "red") +
geom_jitter(color="black", size=1, alpha=.9) +
theme(plot.title = element_text(size = 15, hjust = .5)) +
ggtitle("Boxplot da Variável (Y)") +
xlab("") + ylab("Rendimento por motor em milhas por litro")
```



Pelo boxplot acima percebemos que os dados da variável y parecem seguir uma distribuição Normal - o que será confirmado com a análise de um histograma. Ainda, destaca-se quatro outliers, cujo rendimento é superior a 30 *mi/L*.

```
dados %>%
  ggplot(aes( x=dados$y)) +
    geom_histogram(aes(y=..density..), color = "black", fill = "lightblue", bins = 13) + # xlab("Rendim
    geom_vline(aes(xintercept=mean(dados$y), color = "Média"), linetype = "solid") +
    geom_vline(aes(xintercept=median(dados$y), color = "Mediana"), linetype = "solid") +
    labs(title = "Histograma da variável Y ( em milhas por L)") +
    scale_x_continuous("Rendimento", breaks = seq(10,38,2)) +
    scale_y_continuous("Frequência Relativa") +
    scale_color_manual(name=" ", values = c("red", "black")) +
    ylab("Frequência") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme_pubclean()
```

## Histograma da variável Y ( em milhas por L)



O histograma acima representa a distribuição da variável y. Fica claro, portanto, que os dados da variável resposta se assemelham a uma distribuição normal, não fossem os outliers.

```
dados %>%
  select(y,x1) %>%
  filter(y>30) %>%
  kable(caption = "Tabela com os 4 outliers presentes nos valores de Y")
```

Table 4: Tabela com os 4 outliers presentes nos valores de Y

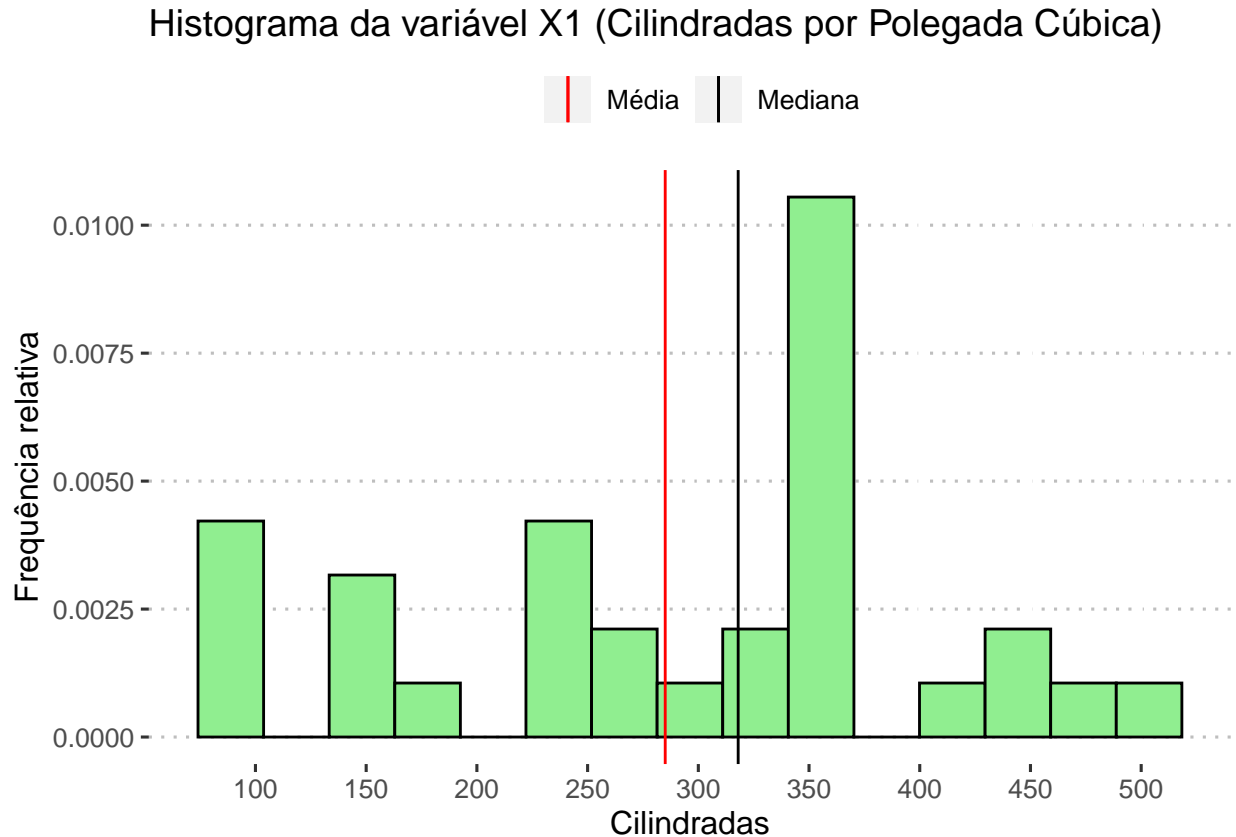
y	x1
34.7	89.7
30.4	96.9
36.5	85.3
31.9	96.9

A partir do histograma e da tabela acima, concluímos que os 4 outliers referem-se aos valores: 30.4, 31.9, 34.7, e 36.5.

Vamos verificar a distribuição da variável  $X_1$

```
dados %>%
  ggplot(aes( x=dados$x1)) +
  geom_histogram(aes(y=..density..), color = "black", fill = "lightgreen", bins = 15) + # xlab("Rendimento")
  geom_vline(aes(xintercept=mean(dados$x1), color = "Média"), linetype = "solid") +
```

```
geom_vline(aes(xintercept=median(dados$x1), color = "Mediana"), linetype = "solid") +
labs(title = "Histograma da variável X1 (Cilindradas por Polegada Cúbica)") +
scale_x_continuous("Cilindradas", breaks = c(100,150,200,250,300,350,400,450,500)) +
scale_y_continuous("Frequência relativa") +
scale_color_manual(name=" ", values = c("red", "black"))+
#limits = c(85,500,1)) +
ylab("Frequência") +
theme(plot.title = element_text(hjust = 0.5)) +
theme_pubclean()
```



O histograma da variável  $X_1$  não apresenta uma distribuição bem definida, entretanto, destaca-se maior frequência para os valores em torno de 350 cilindradas.

**Moda dos valores de  $X_1$**

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

```
mode_x1 = getmode(dados$x1)
print(mode_x1)
```

```
## [1] 350
```

De fato, a moda para os valores de  $X_1$  é 350.

## Parte b):

Consultar e descrever brevemente os conceitos Data splitting, cross validation, overfitting, underfitting, missing data, encoding data.

1. **Data Splitting:** Data Splitting ou também “divisão de dados” é uma abordagem para proteger dados confidenciais de acesso não autorizado, criptografando os dados e armazenando diferentes partes de um arquivo em servidores diferentes. Quando os dados divididos são acessados, as partes são recuperadas, combinadas e descriptografadas.
2. **Cross Validation:** Cross Validation ou também “validação cruzada” é uma técnica muito utilizada para avaliar o desempenho de modelos de aprendizado de máquina. Consiste, basicamente, em particionar os dados em conjuntos, onde um conjunto é utilizado para treino e outro para teste e avaliação do desempenho do modelo. A utilização correta da técnica tem altas chances de detectar se um modelo está sobreajustado aos seus dados de treinamento, ou seja, sofrendo overfitting. Vale ressaltar que existem vários métodos de aplicação da validação cruzada.
3. **Overfitting:** Overfitting ou também “Sobreajuste” consiste na situação em que o modelo se ajusta bem demais ao conjunto de treinamento. Ou seja, nos dados de treinamento, em geral, a acurácia do modelo é muito alta (e, quando há 100% de acurácia dizemos que o modelo “memorizou” os dados). Isso ocorre pois além de aprender os detalhes dos dados o modelo também aprende os ruídos, o que prejudica sua capacidade de generalização no conjunto de teste. Em geral, quanto maior a complexidade do modelo mais propenso ao Overfitting ele se torna.
4. **Underfitting:** Já o Underfitting, por outro lado, refere-se ao problema em que o modelo não é capaz de modelar o conjunto de treinamento e nem generalizar para dados nunca vistos. Em geral, a solução reside no aumento da complexidade do modelo ou a troca do algoritmo.
5. **Missing data:** Missing data, muitas vezes referido como missing values (com tradução literal: valores que faltam), é um conceito utilizado para quando alguma(s) observação(ões) no conjunto de dados está(ão) vazia(s), causando ambiguidade e falta de precisão para a análise do mesmo. Na análise multivariada, temos uma relação proporcional da quantidade de variáveis a serem relacionadas com a falta de rigor causada pelos missing values.
6. **Encoding data:** Encoding data (de tradução literal: dados codificados) é o nome dado para o processo de converter dados para um formato específico, assegurando sua transmissão e otimizando o modelo. Seu processo inverso - ou seja, a decodificação - refere-se a extrair as informações da forma convertida.

## Parte c):

1. Calcular  $S_{XX}$ ,  $S_{YY}$  e  $S_{XY}$

Calculando o valor de  $S_{xx}$

$$S_{XX} = \sum_{i=1}^n (x - \bar{x})^2$$

```
xbarra=mean(x1)
x1-xbarra
```

```
## [1] 67.62857 -32.37143 68.62857 157.62857 -51.37143 -20.37143
## [7] -192.67143 -185.47143 67.62857 -197.07143 -111.37143 -24.37143
## [13] -142.37143 19.62857 157.62857 67.62857 35.62857 -51.37143
## [19] 77.62857 117.62857 -185.47143 177.62857 -148.77143 35.62857
## [25] 68.62857 68.62857 77.62857 77.62857
```



```
(x1-xbarra)^2
```

```
## [1] 4573.6237 1047.9094 4709.8808 24846.7665 2639.0237 414.9951
## [7] 37122.2794 34399.6508 4573.6237 38837.1480 12403.5951 593.9665
## [13] 20269.6237 385.2808 24846.7665 4573.6237 1269.3951 2639.0237
## [19] 6026.1951 13836.4808 34399.6508 31551.9094 22132.9380 1269.3951
## [25] 4709.8808 4709.8808 6026.1951 6026.1951
```

```
Sxx=sum((x1-xbarra)^2)
```

Calculando o valor de  $S_{yy}$

$$S_{YY} = \sum_{i=1}^n (y - \bar{y})^2$$

```
ybarra=mean(y)
y-ybarra
```

```
## [1] -3.1564286 -0.1564286 -1.9064286 -8.9564286 1.9635714 1.3135714
## [7] 14.5435714 10.2435714 -3.6564286 16.3435714 1.3435714 -0.4564286
## [13] 0.1435714 -2.3564286 -5.2664286 -2.3564286 -3.7464286 3.3835714
## [19] 1.3135714 -3.5664286 11.7435714 -6.8864286 3.7435714 -0.4264286
## [25] -6.2564286 -6.8864286 -6.3864286 -3.6564286
```

```
(y-ybarra)^2
```

```
## [1] 9.96304133 0.02446990 3.63446990 80.21761276 3.85561276
## [6] 1.72546990 211.51546990 104.93075561 13.36946990 267.11232704
## [11] 1.80518418 0.20832704 0.02061276 5.55275561 27.73526990
## [16] 5.55275561 14.03572704 11.44855561 1.72546990 12.71941276
## [21] 137.91146990 47.42289847 14.01432704 0.18184133 39.14289847
## [26] 47.42289847 40.78646990 13.36946990
```

```
Syy=sum((y-ybarra)^2)
```

Calculando o valor de  $S_{xy}$

$$S_{XY} = \sum_{i=1}^n (x - \bar{x})(y - \bar{y})$$

```
Sxy=sum((x1-xbarra)*(y-ybarra))
cbind(Sxx,Syy,Sxy)
```

```
##           Sxx      Syy      Sxy
## [1,] 350834.9 1117.405 -17523.4
```

2. Ajustar um modelo de regressão linear simples, apresentar a estimativa de  $\beta_0, \beta_1$  e  $\sigma^2$  e fazer um gráfico com a reta ajustada

## Estimacao dos parametros

$$\beta_1 = S_{XY}/S_{XX}$$

Calculando o valor do coeficiente angular  $\beta_1$

```
b1_est <- Sxy/Sxx
```

Calculando o valor do intercepto  $\beta_0$

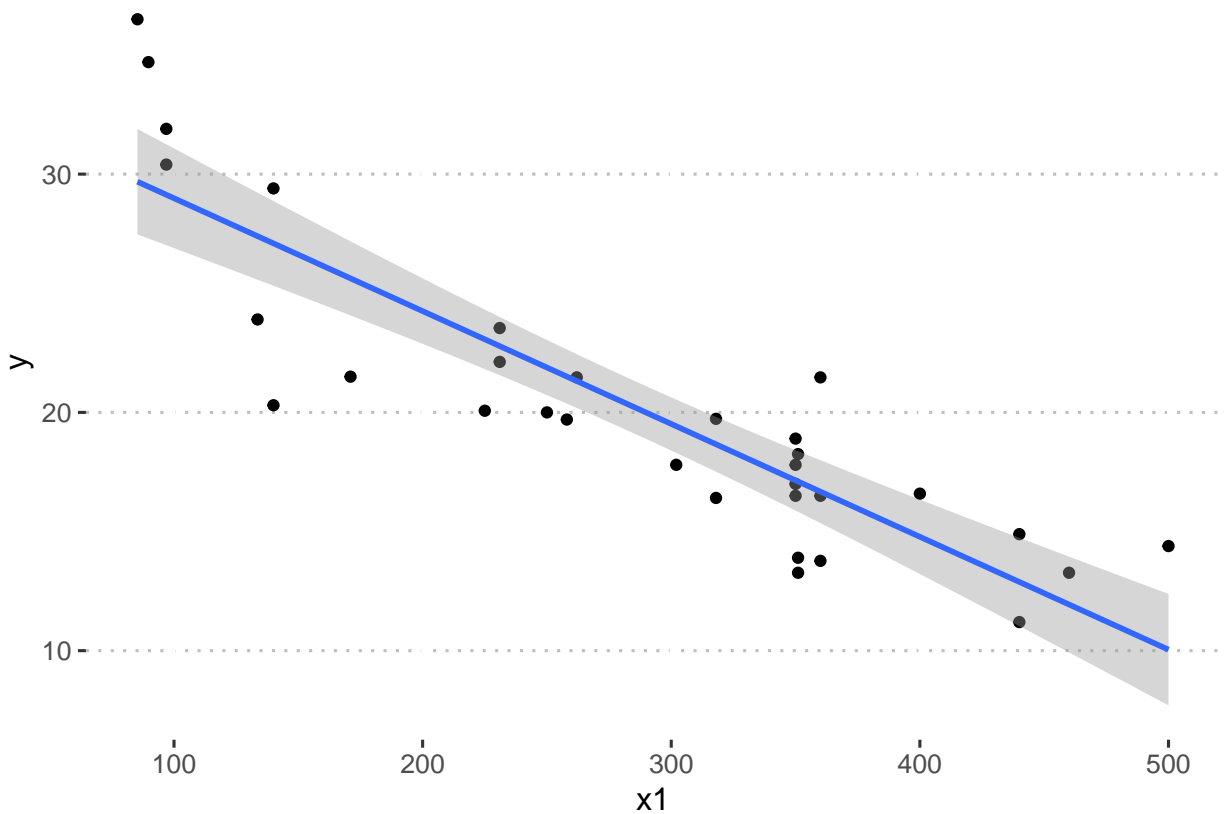
```
b0_est <- mean(y) - b1_est*mean(x1)
```

Calculando o estimador de  $\sigma^2$  não viesado.

Tal estimador é obtido através da soma do quadrado dos resíduos, definido pela variável  $QMres$ , de modo que:

```
# Soma do quadrado da regressão:  
SQreg <- b1_est*Sxy  
# Soma do quadrado total:  
SQtotal <- sum((y-mean(y))^2)  
# Diferença entre a soma do quadrado da regressão e a soma do quadrado total:  
SQres <- SQtotal - SQreg  
# Soma do quadrado dos resíduos:  
QMres <- SQres/(n-2)
```

```
dados %>% ggplot(aes(x= x1, y= y)) + geom_point() +  
  geom_smooth(method='lm', formula= y~x) +  
  theme_pubclean()
```



```
#Arredondando b0 e b1
round(b0_est, 4)
```

```
## [1] 34.2602
```

```
round(b1_est, 4)
```

```
## [1] -0.0499
```

Consequentemente, a reta ajustada é:

$$\hat{Y}_i = 1224.043 - 0.7769X_i$$

3. Calcule o valor dos  $\hat{Y}$  e o valor dos resíduos para seu modelo, resumo e histograma dos resíduos, e faça uma análise da distribuição destes.

O cálculo de  $\hat{Y}$  pode ser realizado utilizando o modelo de regressão linear simples, em que a variabilidade de interesse é dada em função de uma única covariável - no caso,  $x1$ . No R, podemos expressar  $\hat{Y}$  como sendo:

```
y_pred <- b0_est + b1_est*x1
```

Os resíduos se dão pelo desvio entre as observações e os valores preditos, sendo uma medida de variabilidade na variável resposta onde qualquer desvio relativo a suposição dos erros deveria aparecer. Analisá-los nos permite um discernimento maior em relação a quão adequado é o modelo. Fazendo uso do cálculo de  $y\_pred$  feito anteriormente, salvamos nossos resíduos em uma variável *res* abaixo.

```
res <- y - y_pred
```

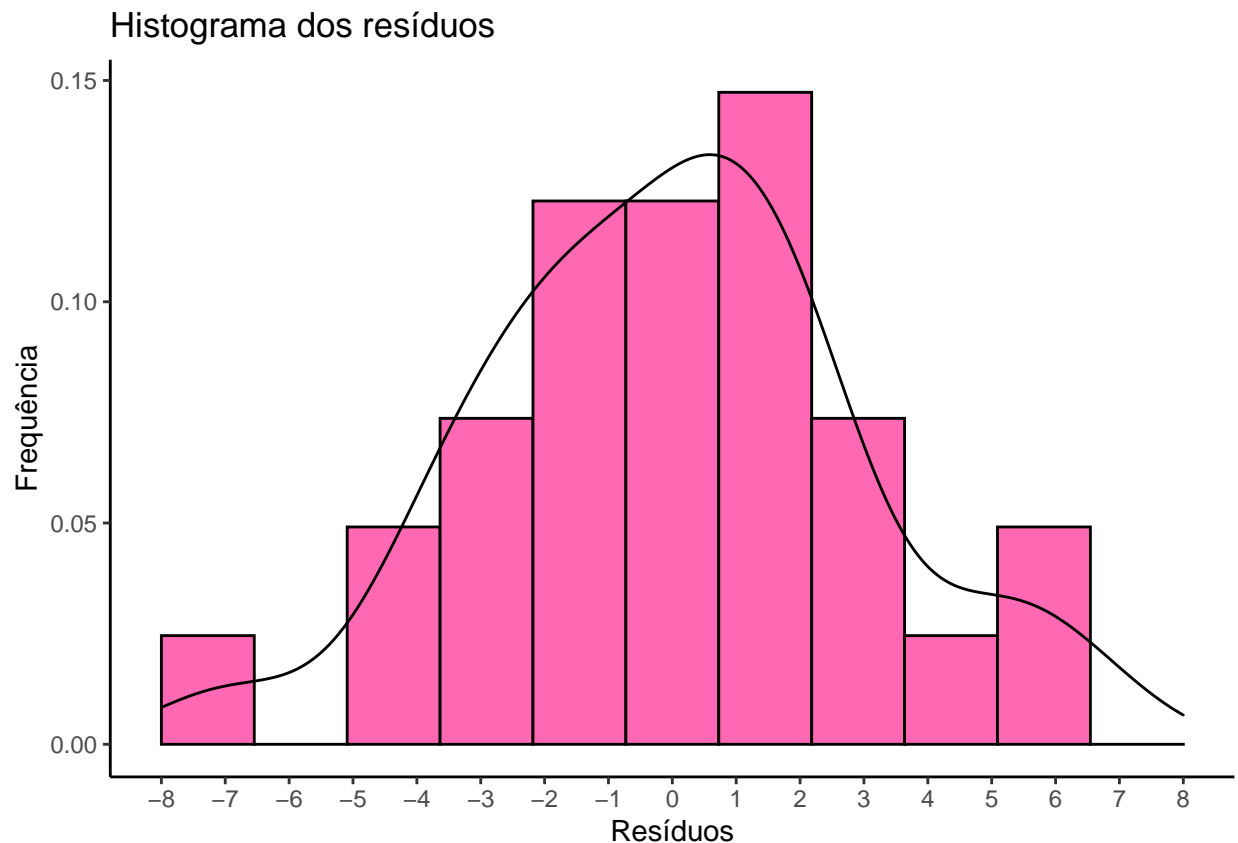
Utilizando o comando *summary*, podemos observar as principais medidas descritivas da variável, o que nos auxilia para a análise da mesma.

```
summary(res)
```

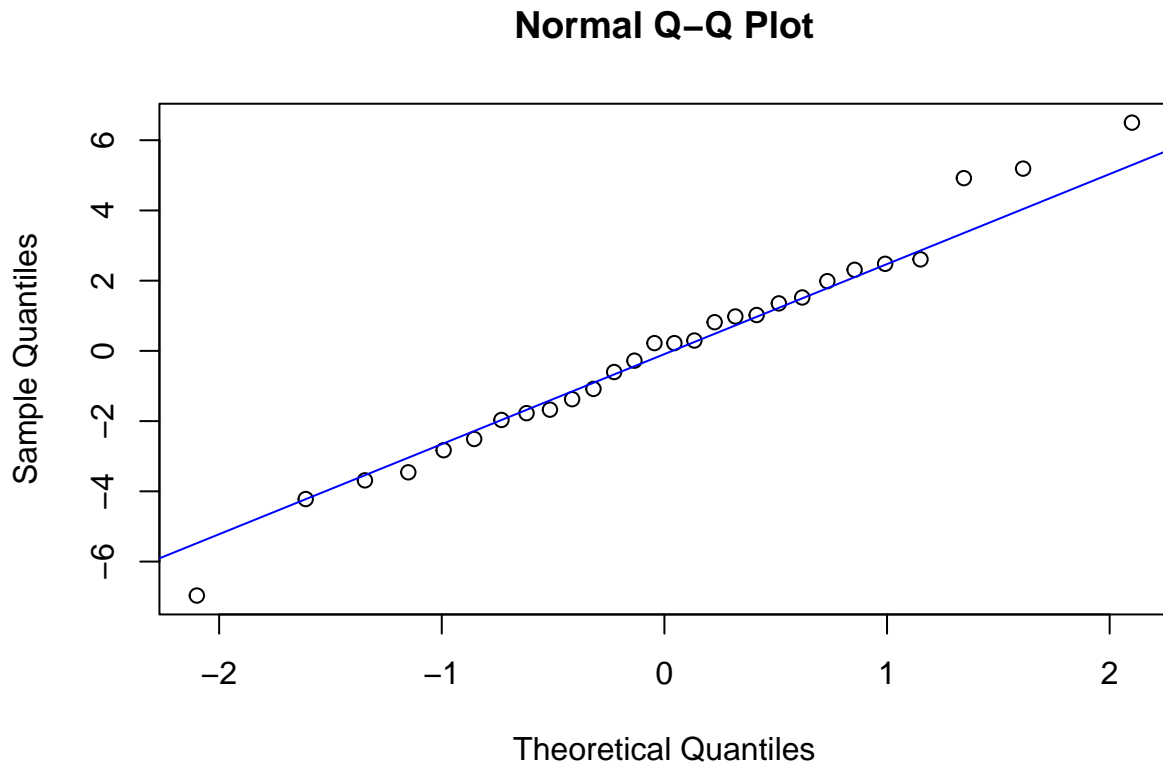
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -6.9676 -1.8217   0.2212   0.0000  1.6375   6.5003
```

Também podemos construir um histograma, que facilitará a visualização do comportamento dos resíduos.

```
# Histograma
n_res <- length(res)
normalDist<- rnorm(length(res), mean = mean(res), sd= sd(res))
dat<- data.frame(error = res, norm = normalDist)
ggplot(tibble(res), aes(x = res)) +
  ylab("Frequência") +
  geom_histogram(aes(y=..density..), color = "black", fill = "#FF69B4", bins=12, position="identity")+
  labs(title = "Histograma dos resíduos")+
  geom_density(data = dat) +
  scale_x_continuous("Resíduos", limits = c(-8,8,1), breaks = c(-8:8))+
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```



```
qqnorm(res)
qqline(res, col = "blue")
```



Desse modo, temos que a análise dos resíduos é —

#### 4. testes de hipótese para $\beta_0$ e $\beta_1$

Para realizarmos nossos testes de hipóteses, é necessário o estimador do parâmetro  $\sigma^2$  do nosso modelo, uma vez que ele não é dado. Tal estimador não viesado é obtido através da soma do quadrado dos resíduos, definido pela variável  $QMres$ , calculada no item 3 de modo que:

```
# Soma do quadrado da regressão:
SQreg <- b1_est*Sxy
# Soma do quadrado total:
SQtotal <- sum((y-mean(y))^2)
# Diferença entre a soma do quadrado da regressão e a soma do quadrado total:
SQres <- SQtotal - SQreg
# Soma do quadrado dos resíduos:
QMres <- SQres/(n-2)
```

A partir disso, podemos prosseguir com nossos testes de hipóteses para  $\beta_1$  e  $\beta_0$ , com decisão de rejeitar ou não  $H_0$ , uma vez que este representa o parâmetro se igualar a 0 estatisticamente caso não seja rejeitado, descrevendo a significância da contribuição do mesmo.

- Testagem se  $\beta_1 = 0$ :

$\beta_1$  possui distribuição Normal com média  $\beta_1$  e variância  $\sigma^2/S_{xx}$ , com isso, definimos:

```
dp_b1 <- (sqrt(QMres/Sxx))
t0_b1 <- b1_est/dp_b1
```

Pelo enunciado, é dado que  $\alpha = 5\%$ . Se  $H_0$  não for rejeitado, temos que  $\beta_1$  é estatisticamente igual a zero. A partir disso, definimos  $\alpha$  e dois quantis, de modo que  $t1$  é o quantil  $\frac{\alpha}{2}$  da distribuição  $t$  com grau de liberdade  $n - 2$ , enquanto  $t2$  é o quantil  $\frac{1-\alpha}{2}$  da distribuição  $t$  com grau de liberdade  $n - 2$ . Com esses dados, podemos construir nosso programa de decisão que retornará caso  $H_0$  seja rejeitado.

```
alpha <- 0.05
t1 <- qt(alpha/2,n-2)
t2 <- qt(1-alpha/2,n-2)
if(t0_b1 < t1 || t0_b1>t2){
  cat("Rejeita-se H0")
}
```

## Rejeita-se  $H_0$

Para  $\alpha = 1\%$ :

```
alpha <- 0.01
t1 <- qt(alpha/2,n-2)
t2 <- qt(1-alpha/2,n-2)
if(t0_b1 < t1 || t0_b1>t2){
  cat("Rejeita-se H0")
}
```

## Rejeita-se  $H_0$

Realizando o teste, temos que  $H_0$  é rejeitado, logo,  $\beta_1$  é diferente de zero.

- Testagem se  $\beta_0 = 0$ :

$\beta_0$  possui distribuição Normal com média  $\beta_0$  e variância  $\sigma^2((\frac{1}{n}) + \frac{\bar{X}}{S_{xx}})$ , com isso, definimos:

```
dp_b0 <- (sqrt( QMres * ( (1/n) + (mean(x1))^2/Sxx )))
t0_b0 <- b0_est/dp_b0
```

Em um processo semelhante à testagem de  $\beta_1$ , com  $\alpha = 5\%$  e os mesmos quantis, também é possível a construção de nossa função de decisão. Se  $H_0$  não for rejeitado, temos que  $\beta_0$  é estatisticamente igual a zero.

```
alpha <- 0.05
t1 <- qt(alpha/2,n-2)
t2 <- qt(1-alpha/2,n-2)
if(t0_b1 < t1 || t0_b1>t2){
  cat("Rejeita-se H0")
}
```

## Rejeita-se  $H_0$

Para  $\alpha = 0.01$

```
alpha <- 0.01
t1 <- qt(alpha/2,n-2)
t2 <- qt(1-alpha/2,n-2)
if(t0_b1 < t1 || t0_b1>t2){
  cat("Rejeita-se H0")
}
```

```
## Rejeita-se H0
```

Realizando o teste, temos que  $H_0$  é rejeitado, logo,  $\beta_0$  é diferente de zero.

5. intervalos de confiança

### *Intervalos de Confiança*

Intervalos de Confiança para  $(\beta_0, \beta_1, \sigma^2)$  e  $E(Y)$ .

*Calculando intervalo de Confiança para  $\beta_1$*

```
b1_min <- b1_est-t2*dp_b1
b1_max <- b1_est-t1*dp_b1
IC_b1_est <- cbind(b1_min, b1_max)
IC_b1_est
```

```
##          b1_min      b1_max
## [1,] -0.06426462 -0.03563078
```

(y) (Milhas por litro) e a (x1) (polegadas cúbicas)

**Interpretação:** Cada incremento em polegada cúbica na cilindrada do motor aumenta o consumo em milhas por litro em -0.0473, com uma margem de erro de aproximadamente 0.009 para mais ou para menos.

*Calculando intervalo de Confiança para  $\beta_0$*

```
b0_min <- b0_est-t2*dp_b0
b0_max <- b0_est-t1*dp_b0
IC_b0_est <- cbind(b0_min, b0_max)
IC_b0_est
```

```
##          b0_min    b0_max
## [1,] 29.91148 38.60898
```

*Calculando intervalo de confiança para  $\sigma^2$*

Lembrado que  $SQ_{res}/\sigma^2$  tem Distribuição qui-quadrado com  $(n - 2)$  G.L.

```
t1_sig <- qchisq(alpha/2, n-2)
t2_sig <- qchisq(1-alpha/2,n-2)
```

```
sig_min <- SQres/t2_sig
sig_max <- SQres/t1_sig
```

```
IC_sig_est <- cbind(sig_min, sig_max)
IC_sig_est
```

```
##          sig_min  sig_max
## [1,] 5.014542 21.69771
```

### Calculando intervalo de confiança para a esperança de y

(valor medio da variavel resposta para um valor particular da cov.,  $MIy|X_0$ ).

#### Lembrando:

1. O valor médio da variável resposta é dado um  $X_0$ .
2.  $\bar{Y}$  tem Distribuição Normal. com média  $\beta_0 + \beta_1 * \bar{X}$  e variância  $\sigma^2/n$ .
3.  $\beta_1$  tem Distribuição Normal com média  $\beta_1$  e variância  $\sigma^2/Sxx$ .
4. A Esperança de  $Y|X_0$  é Normal.
5. a Variância de  $MIy|X_0$  é  $\sigma^2 * (1/n + ((X_0 - \bar{X})^2)/Sxx)$ ,  $t_1 =$  quantil da dist.  $t(\alpha/2, n-2)$ ,  $t_2 =$  quantil da dist.  $t(1 - \alpha/2, n-2)$ ,  $\alpha = 0,05$ .

**Exemplo:** Nesse exemplo usaremos  $X_0$  como sendo o proprio  $\bar{X}$ .

```
X0 <- mean(x1) # poderia ser outro valor
v_medio <- (mean(y)+b1_est* (X0-mean(x1)))
auxiliar <- sqrt(QMres*(1/n + (X0 - mean(x1)) /Sxx ))
```

### Intervalo De Confiança

```
v_medio_min <- v_medio - t2*auxiliar
v_medio_max <- v_medio - t1*auxiliar
IC_v_medio <- cbind(v_medio_min, v_medio_max)
IC_v_medio
```

```
##          v_medio_min v_medio_max
## [1,]      18.55384      21.75902
```

E se quiséssemos prever a mortalidade baseado em um novo valor da variavel explicativa utilizada. Qual seria o intervalo que em 95% das vezes iria conter o verdadeiro valor predito considerando a nova informação de  $x_1$  ? (Ou seja, qual seria o Intervalo de Confiança para o valor predito de  $Y$  baseado no novo valor da variável  $x_1$  com 95% de confiança).

### Intervalo de predição

O intervalo de predição para até 5 valores diferentes de  $X_0$ .

#### Intervalos de predição

*Lembrando:*

$Y_0\_est = \beta_0\_est + \beta_1\_est * x_1\_novo$ .



$Y_0$  e  $Y_{0\_est}$  são independentes.

$t_1 =$  quantil da dist.  $t(\alpha/2, n - 2)$ .

$t_2 =$  quantil da dist.  $t(1 - \alpha/2, n - 2)$ .

$\alpha = 0,05$ .

### ***Exemplo***

```
x1_novo <- 12
#x1_novo <- c(12,20,48,51,57,62)
Y0_est <- b0_est + b1_est*x1_novo
auxiliar_y0 <- sqrt(QMres*(1+ 1/n + (x1_novo - mean(x1)) /Sxx ))
```

```
Y0_est_min <- Y0_est - t2*auxiliar_y0
Y0_est_max <- Y0_est - t1*auxiliar_y0
```

```
IC_Y0_est <- cbind(Y0_est_min, Y0_est_max)
IC_Y0_est
```

```
##      Y0_est_min Y0_est_max
## [1,]    25.03387    42.28785
```

### ***Análise de Variância***

A Análise de Variância com todos os valores (Graus de Liberdade, SQTotal, SQRes, SQReg, QMRes, QMReg e F).

#### ***ANOVA***

Teria outra forma de testarmos a significancia da regressão ? Sim! Outra forma seria pela Análise de Variância (ANOVA), nesse caso testariamos se  $\beta_1 = 0$ .

#### **Soma do quadrado da Regressão**

```
SQreg <- b1_est*Sxy
SQreg
```

```
## [1] 875.2534
```

#### **Soma do quadrado total**

```
SQttotal <- sum((y-mean(y))^2)
SQttotal
```

```
## [1] 1117.405
```

#### **Soma do quadrado do resíduo**

```
SQres <- sum((y-mean(y))^2) - b1_est*Sxy
SQres
```

```
## [1] 242.1516
```

```
QMreg <- SQreg
QMreg
```

```
## [1] 875.2534
```

Lembre-se que  $QMres$  eh o estimador de  $\sigma^2$  e  $QMres = SQres / (n-2)$

```
F_0 <- QMreg/QMres
F_0
```

```
## [1] 93.9766
```

Quantil da Distribuição F-Snedecor

```
f1 <- pf(F_0, df1 = 1, df2 = n-2, lower.tail = F)
f1
```

```
## [1] 4.031973e-10
```

```
if(F_0 > f1){
  cat("Rejeita-se H0")
}
```

```
## Rejeita-se H0
```

ou poderíamos ter calculado

```
f1.2 <- pf(t0_b1^2, df1 = 1, df2 = n-2, lower.tail = F)
```

```
if(t0_b1^2 > f1.2){
  cat("Rejeita-se H0")
}
```

```
## Rejeita-se H0
```

*Anova usando funções do R*

```
Anovamodel <- aov(y ~ x1, data = dados)
Anovamodel
```

```
## Call:
## aov(formula = y ~ x1, data = dados)
##
## Terms:
##              x1 Residuals
## Sum of Squares 955.7197 281.8244
## Deg. of Freedom      1      30
##
## Residual standard error: 3.064987
## Estimated effects may be unbalanced
```

```
summary(Anovamodel)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## x1              1  955.7    955.7   101.7 3.74e-11 ***
## Residuals      30  281.8      9.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### *Normalidade dos resíduos*

```
shapiro.test(resid(Anovamodel))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(Anovamodel)
## W = 0.98718, p-value = 0.961
```

A hipótese nula do Teste de Shapiro-Wilk é de que não há diferença entre a nossa distribuição dos dados e a distribuição normal. O valor-p maior do que 0.05 nos dá uma confiança estatística para afirmar que as distribuição dos nossos resíduos não difere da distribuição normal.

Dessa forma nossos dados satisfazem todas as premissas da ANOVA e portanto, o resultado da nossa ANOVA são válidos.