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)
library(papeR)
```

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 6 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

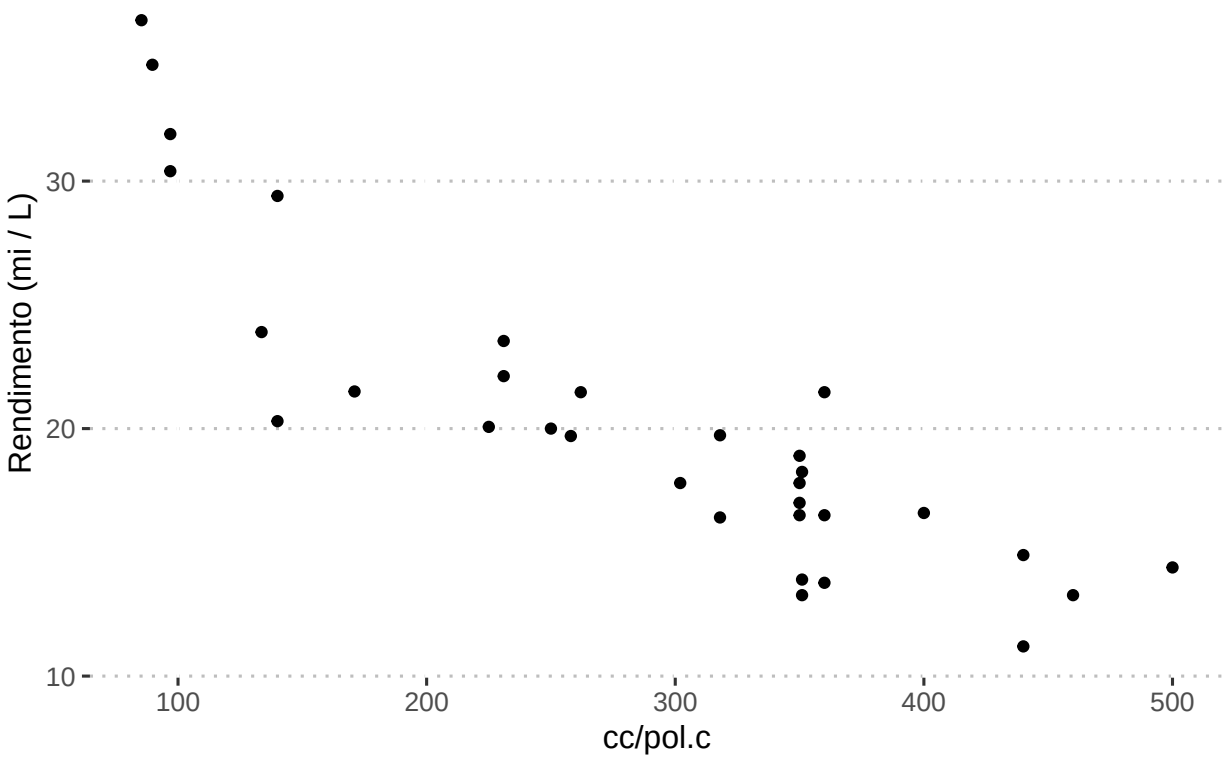
```
dim(dados) %>% kable(caption = "Dimensão dos dados")
```

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))
```



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 ρ.

Teste de Correlação de Pearson entre as variáveis x_1 e y

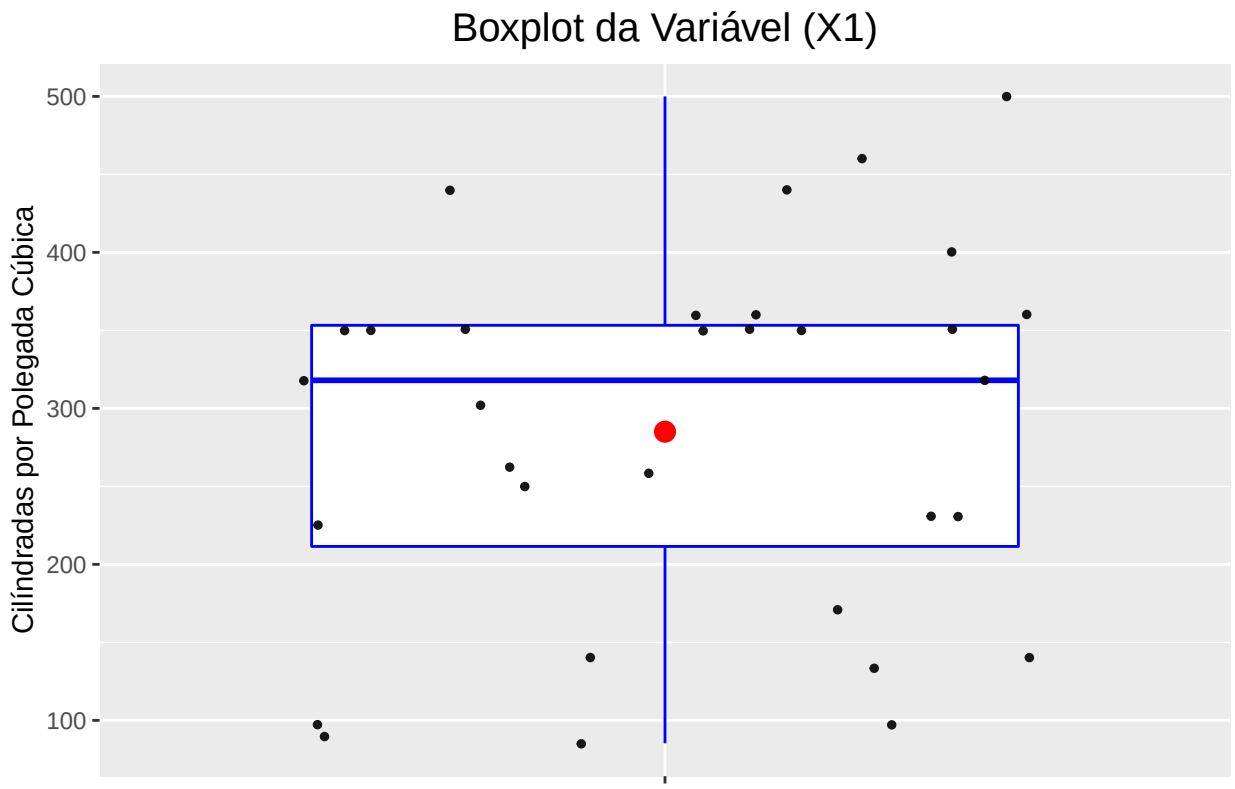
```
cor.test(dados$x1, dados$y)
```

```
##
##  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.

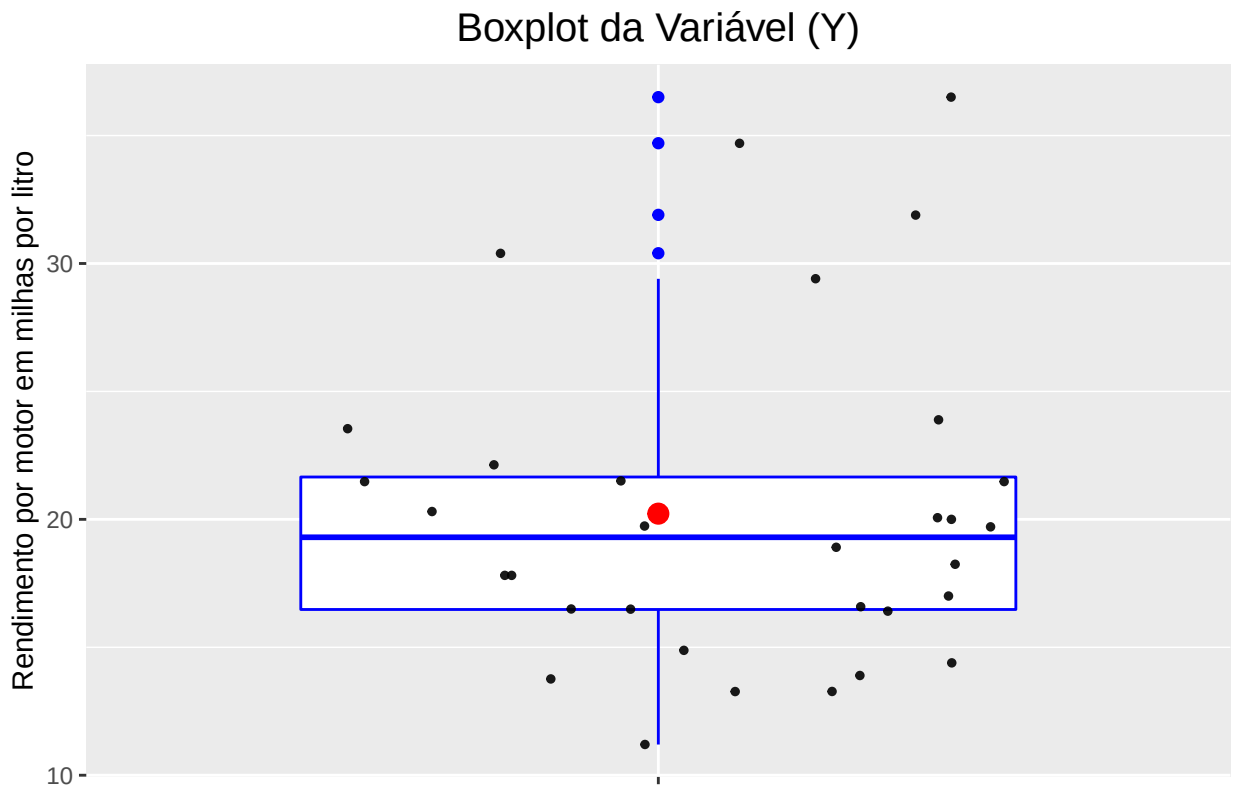
```
dados %>%
  ggplot(aes(x = "", y = dados$x1)) +
    geom_boxplot(color = "blue") +
```

```
    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")
```



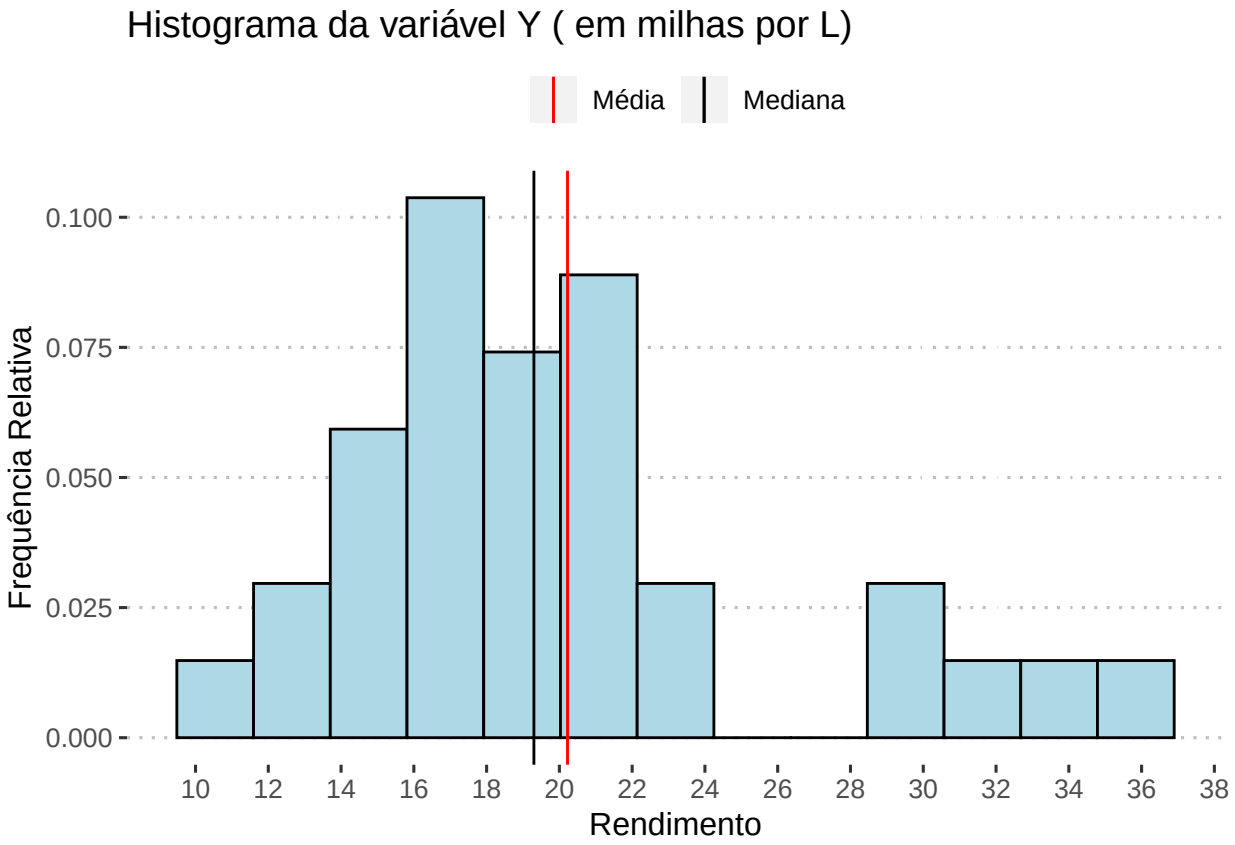
Podemos perceber a partir do boxplot acima e da função summary que 50% dos carros tem menos de 318 cilíndradas.

```
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()
```



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

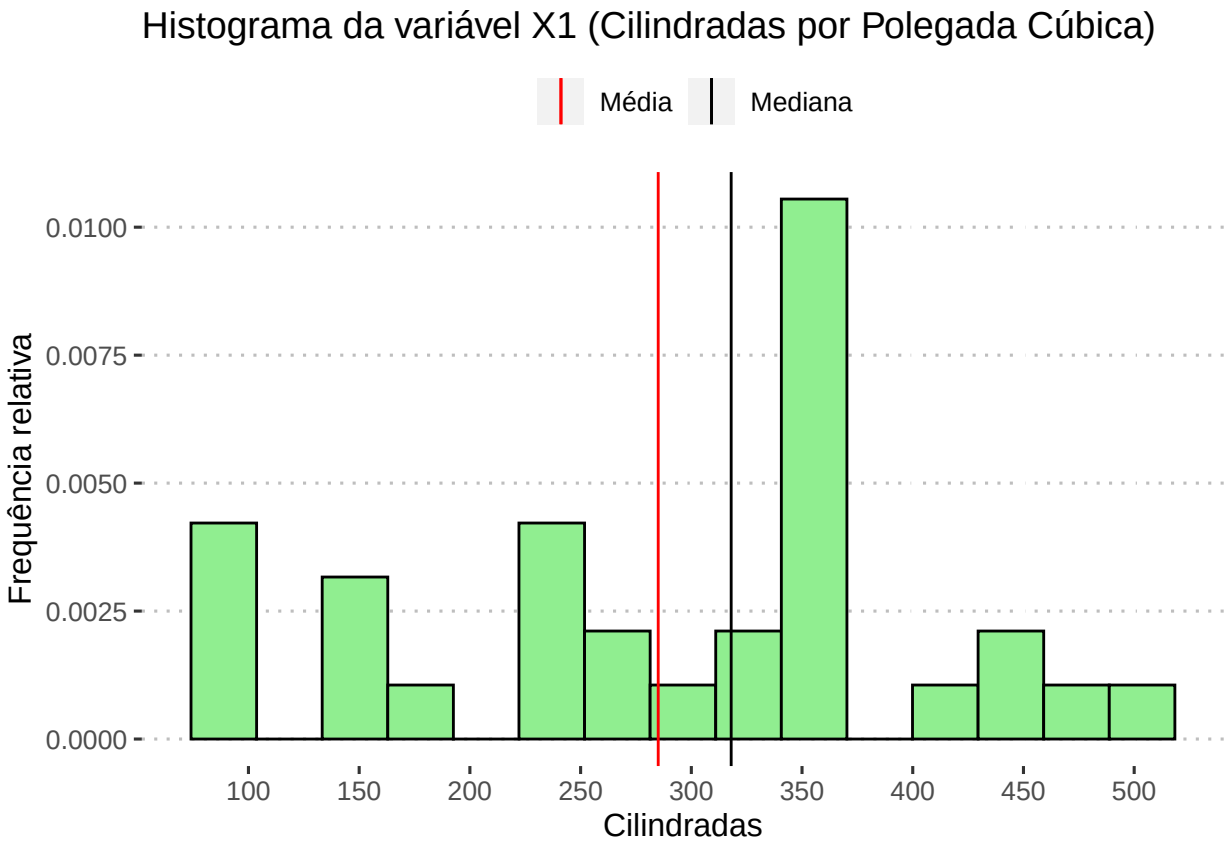
y	x1
34.7	89.7
30.4	96.9
36.5	85.3
31.9	96.9

A partir do histograma e da tabela acima, concluimos 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$x1)) +
    geom_histogram(aes(y=..density..), color = "black", fill = "lightgreen", bins = 15) + # xlab("Rendi
    geom_vline(aes(xintercept=mean(dados$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 cilíndradas.

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 de data splitting, cross validation, overfitting, underfitting, missing data e 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 ambuiguidade 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 β_0, β_1 e σ^2 e fazer um gráfico com a reta ajustada

Estimação dos parâmetros

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

Calculando o valor do coeficiente angular β_1

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

Calculando o valor do intercepto β_0

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

Calculando o estimador de σ^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)
```

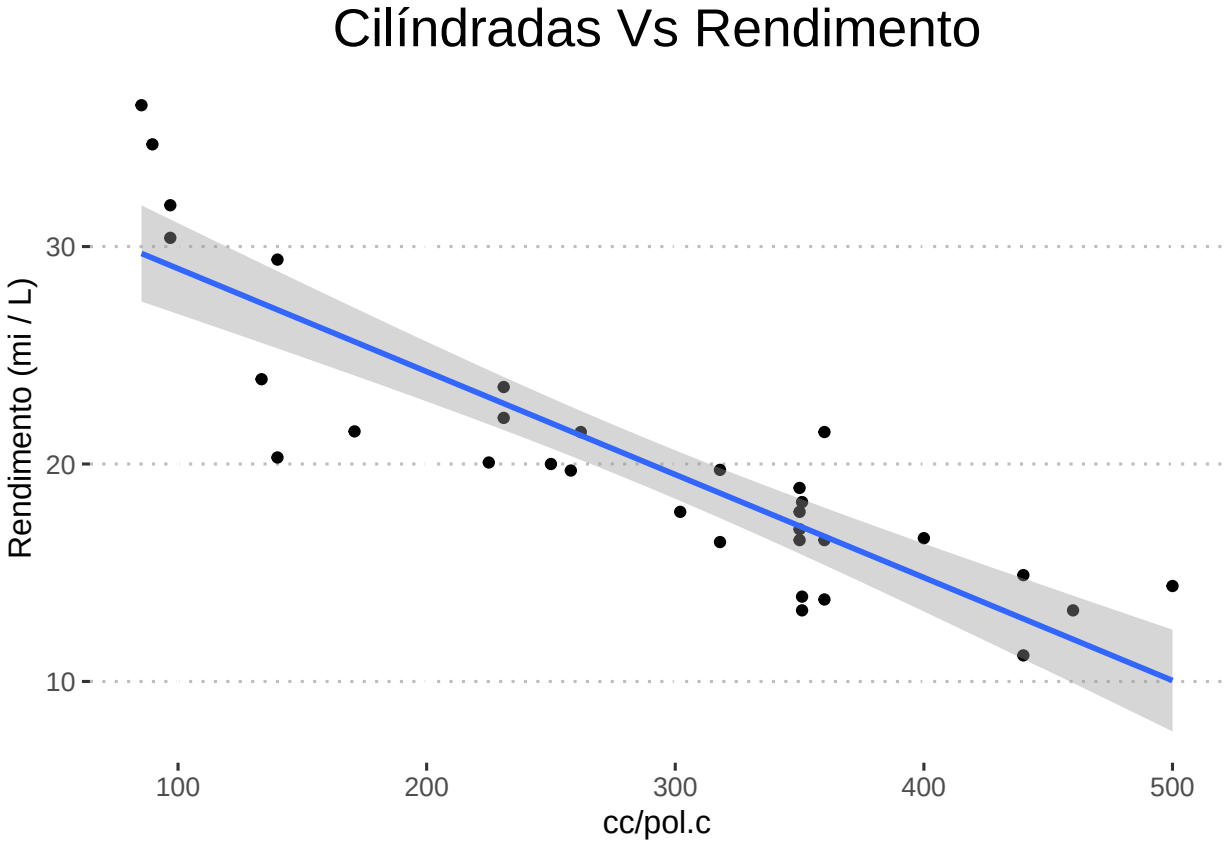
Calculando o Coeficiente de Determinação R^2 para realizar a interpretação gráfica

$$R^2 = SQ_{reg}/SQ_{total}$$

```
Coef_Det = SQreg/SQtotal
```

Gráfico de Dispersão entre x_1 (cilindrada do motor) e y (rendimento da gasolina) com a reta ajustada

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



Interpretação gráfica: Conforme revelado no resultado do coeficiente de determinação, o qual mostra a proporção em que a variação do rendimento de gasolina é explicada pelas cilindradas do motor, cujo resultado foi de aproximadamente 0.7833, apresenta um valor próximo de 1, indicando assim uma boa correlação linear entre as variáveis dado que segundo esta análise, aproximadamente 78,33% da variação do rendimento de gasolina é explicada pelas cilindradas do motor.

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

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

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

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

Consequentemente, a reta ajustada é:

$$\widehat{Y}_i = 34.2602 - 0.0499X_i$$

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, x_1. 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 *summarize*, podemos observar as principais medidas descritivas da variável, o que nos auxilia para a análise da mesma.

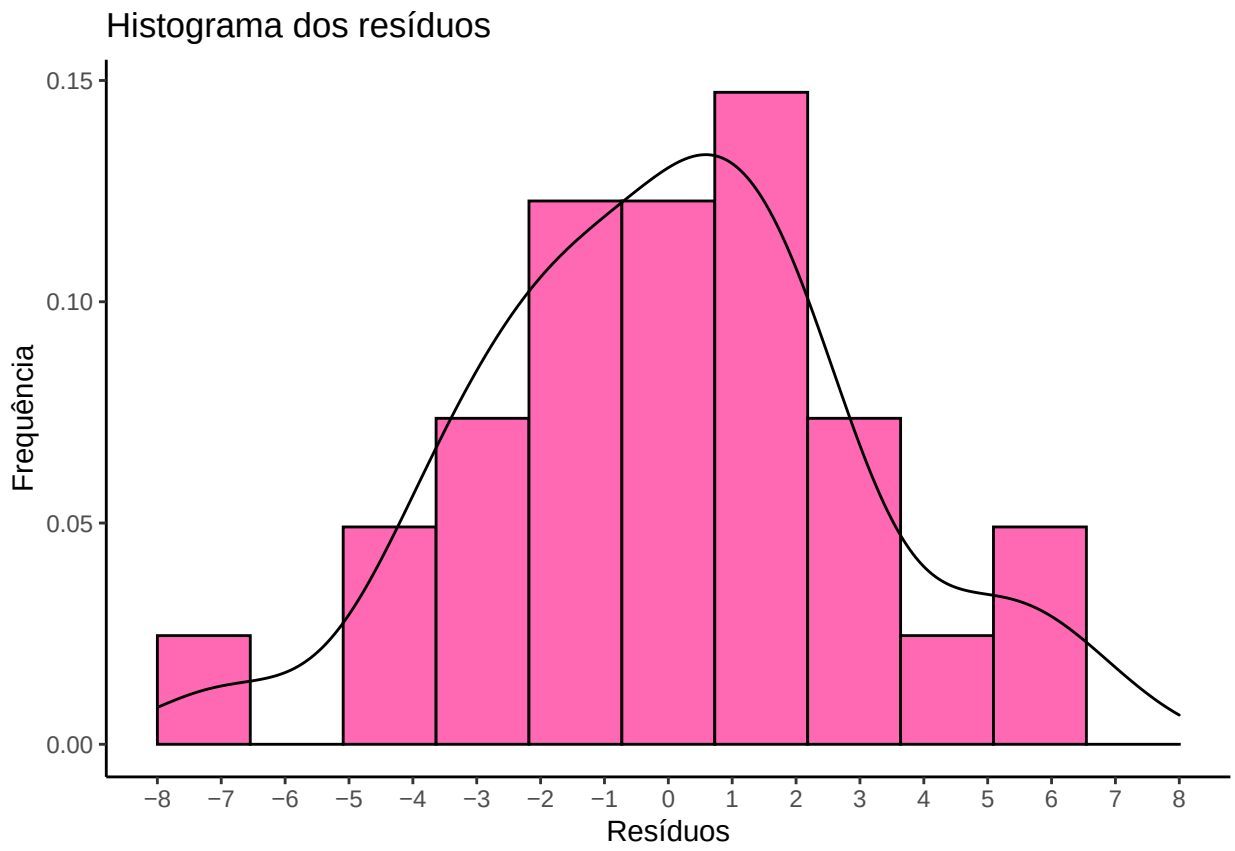
```
papeR::summarize(tibble(res)) %>% kable(caption = "Sumário dos resíduos")
```

Table 5: Sumário dos resíduos

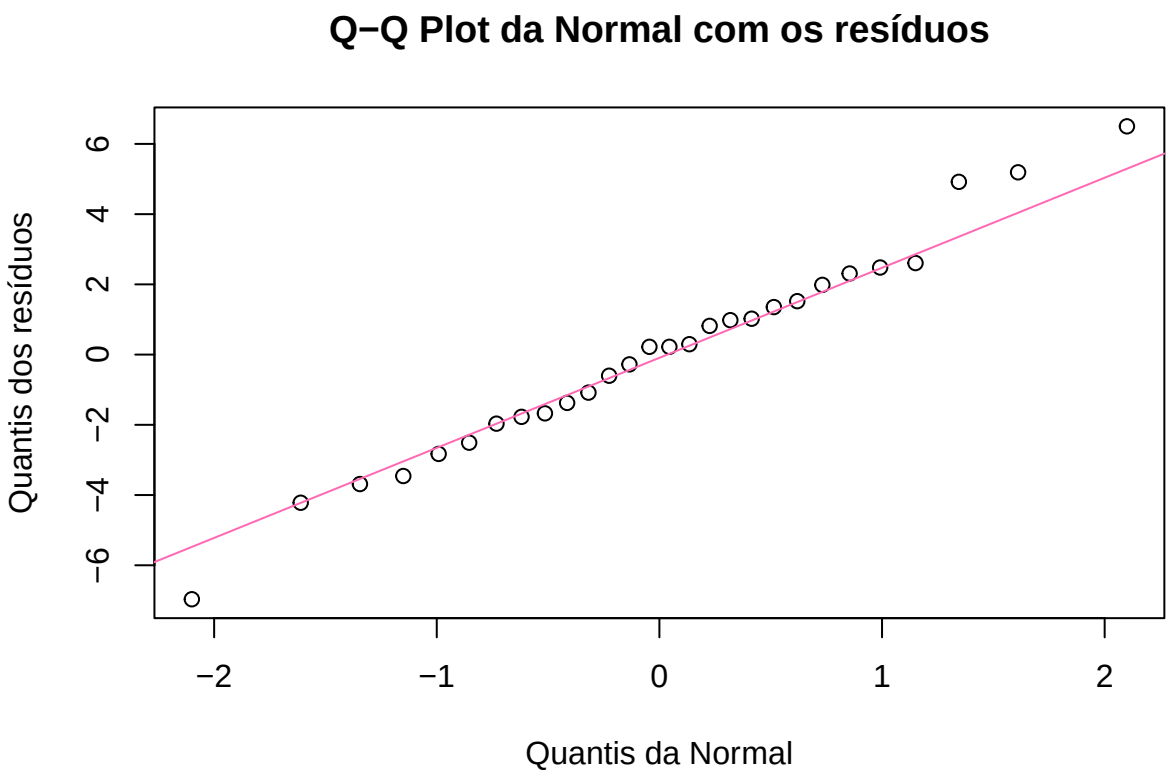
	N	Mean	SD	Min	Q1	Median	Q3	Max
res	28	0	2.99	-6.97	-1.87	0.22	1.75	6.5

Também podemos construir um histograma, assim como um gráfico quantil-quantil, 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()
```




```
# Q-Q
qqnorm(res, main = "Q-Q Plot da Normal com os resíduos", ylab =" Quantis dos resíduos ", xlab = " Quanti
qqline(res, col = "#FF69B4")
```



Desse modo, é possível observar que os resíduos tendem a se aproximar cada vez mais da normal. Posteriormente, a normalidade dos resíduos será abordada com mais precisão.

4. testes de hipotese para β_0 e β_1

Para realizarmos nossos testes de hipóteses, é necessário o estimador do parâmetro σ^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 β_1 e β_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 $\beta_1 = 0$:

β_1 possuí distribuição Normal com média β_1 e variância σ^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 H0 não for rejeitado, temos que β_1 é estatisticamente igual a zero. A partir disso, definimos α e dois quantis, de modo que *t1* é o quantil $\frac{\alpha}{2}$ da distribuicao t com grau de liberdade $n-2$, enquanto *t2* é o quantil $\frac{1-\alpha}{2}$ da distribuicao 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 H0
```

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 H0
```

Realizando o teste, temos que H0 é rejeitado, logo, β_1 é diferente de zero.

- Testagem se $\beta_0 = 0$:

β_0 possuí distribuição Normal com média β_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 β_1, com α= 5% e os mesmos quantis, também é possível a construção de nossa função de decisão. Se H0 não for rejeitado, temos que β_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 H0
```

Para $\alpha = 0.01$

```r
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 H0 é rejeitado, logo, β_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 β_1

```r
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
```

Interpretação: Cada incremento em polegada cúbica na cilindrada do motor aumenta o consumo em milhas por litro em -0.0643, com uma margem de erro de aproximadamente 0.014 para mais ou para menos.

Calculando intervalo de Confiança para β_0

```r
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 σ^2

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

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

```r
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

Aqui, calculamos o valor médio da variável resposta para para um valor particular da covariável $\mu(y|x_0)$.

Lembrando:

1. O valor médio da variável resposta é dado um X_0.
2. $\overline{Y}$ tem Distribuição Normal. com média $\beta_0 + \beta_1 * \overline{X}$ e variância σ^2/n.
3. β_1 tem Distribuição Normal com média β_1 e variância σ^2/Sxx.
4. A Esperança de $Y|X_0$ é Normal.
5. a Variância de $MIy|X_0$ é $\sigma^2 * (1/n + ((X_0 - \overline{X}^2)/Sxx)$, t_1 = quantil da dist. $t(\alpha/2, n-2)$, t_2 = quantil da dist. $t(1-\alpha/2, n-2)$, $\alpha = 0,05$.

Usaremos X_0 como sendo o proprio $\overline{X}$.

```
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
```

Intervalo de predição

Se quisessemos predizer a mortalidade baseado em um novo valor da variavel explicativa utilizada. Qual seria o intervalo que em 99% das vezes iria conter o verdadeiro valor predito considerando a nova informação de x_1?

Calcularemos o intervalo de predição com 99% de confiança para a nossa amostra de teste.

Intervalos de predição

Lembrando:

Y_0_est = β_0_est + β_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. $t(1-\alpha/2, n-2)$.

$\alpha = 0,05$.

```
x1_novo <- teste$x1
#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 <- tibble(Y0_est_min, Y0_est_max, y = teste$y)
IC_Y0_est %>% kable(caption = "Intervalo de predição para amostra de teste")
```

Table 6: Intervalo de predição para amostra de teste

Y0_est_min	Y0_est_max	y
0.6535957	17.91917	14.39
14.3924791	31.65152	20.07
8.1475320	25.40954	18.90
18.6390434	35.89607	29.40

Como podemos ver, o valor de y real estava dentro do intervalo para as quatro observações.

Análise de Variâcia

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

ANOVA

Vamos testar a significancia da regressão através da 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

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

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

Soma do quadrado do resíduo

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

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

```r
QMreg <- SQreg
QMreg
```

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

Lembre-se que QMres eh o estimador de sigmaˆ2 e QMres=SQres/(n-2)

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

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

Quantil da Distribuição F-Snedecor

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

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

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

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

Usando funções do R

```r
fit <- lm(y~x1)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9676 -1.8217  0.2212  1.6375  6.5003
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.260233   1.565022  21.891  < 2e-16 ***
## x1          -0.049948   0.005152  -9.694 4.03e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.052 on 26 degrees of freedom
## Multiple R-squared:  0.7833, Adjusted R-squared:  0.775
## F-statistic: 93.98 on 1 and 26 DF,  p-value: 4.032e-10
```

Podemos ver que obtivemos coeficientes similares em nosso modelo.

```r
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 875.25  875.25  93.977 4.032e-10 ***
## Residuals 26 242.15    9.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Podemos notar que todos os resultados são similares aos já obtidos.

Normalidade dos resíduos

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

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 0.01 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, os resultados da nossa ANOVA são válidos, confirmando nossa premissa de que os resíduos possuem distribuição normal.