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

bla 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).

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)
```

```
## # A tibble: 6 x 12
##       y    x1    x2    x3    x4    x5    x6    x7    x8    x9   x10   x11
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  18.9   350   165   260  8     2.56     4     3  200.  69.9  3910     1
## 2  17     350   170   275  8.5   2.56     4     3  200.  72.9  3860     1
## 3  20     250   105   185  8.25  2.73     1     3  197.  72.2  3510     1
## 4  18.2   351   143   255  8     3        2     3  200.  74    3890     1
## 5  20.1   225    95   170  8.4   2.76     1     3  194.  71.8  3365     0
## 6  11.2   440   215   330  8.2   2.88     4     3  184.  69    4215     1
```

```
y <- dados$y
x1 <- dados$x1
n <- length(y)
```

Parte a):

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

```
summary(dados)
```

```
##        y               x1              x2              x3
##  Min.   :11.20   Min.   : 85.3   Min.   : 70.0   Min.   : 81.0
##  1st Qu.:16.48   1st Qu.:211.5   1st Qu.:102.8   1st Qu.:171.2
##  Median :19.30   Median :318.0   Median :141.5   Median :243.0
##  Mean   :20.22   Mean   :285.0   Mean   :136.9   Mean   :217.9
##  3rd Qu.:21.66   3rd Qu.:353.2   3rd Qu.:166.2   3rd Qu.:258.8
##  Max.   :36.50   Max.   :500.0   Max.   :223.0   Max.   :366.0
##                                                  NA's   :2
##        x4              x5              x6              x7
##  Min.   :7.600   Min.   :2.450   Min.   :1.000   Min.   :3.000
##  1st Qu.:8.000   1st Qu.:2.710   1st Qu.:2.000   1st Qu.:3.000
##  Median :8.250   Median :3.000   Median :2.000   Median :3.000
##  Mean   :8.281   Mean   :3.055   Mean   :2.594   Mean   :3.344
##  3rd Qu.:8.500   3rd Qu.:3.228   3rd Qu.:4.000   3rd Qu.:3.250
##  Max.   :9.000   Max.   :4.300   Max.   :4.000   Max.   :5.000
##
##        x8              x9             x10             x11
##  Min.   :155.7   Min.   :61.80   Min.   :1905   Min.   :0.0000
##  1st Qu.:175.2   1st Qu.:65.40   1st Qu.:2940   1st Qu.:0.0000
##  Median :195.7   Median :72.00   Median :3755   Median :1.0000
##  Mean   :192.0   Mean   :71.28   Mean   :3587   Mean   :0.7188
##  3rd Qu.:202.6   3rd Qu.:76.30   3rd Qu.:4215   3rd Qu.:1.0000
##  Max.   :231.0   Max.   :79.80   Max.   :5430   Max.   :1.0000
##
```

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**

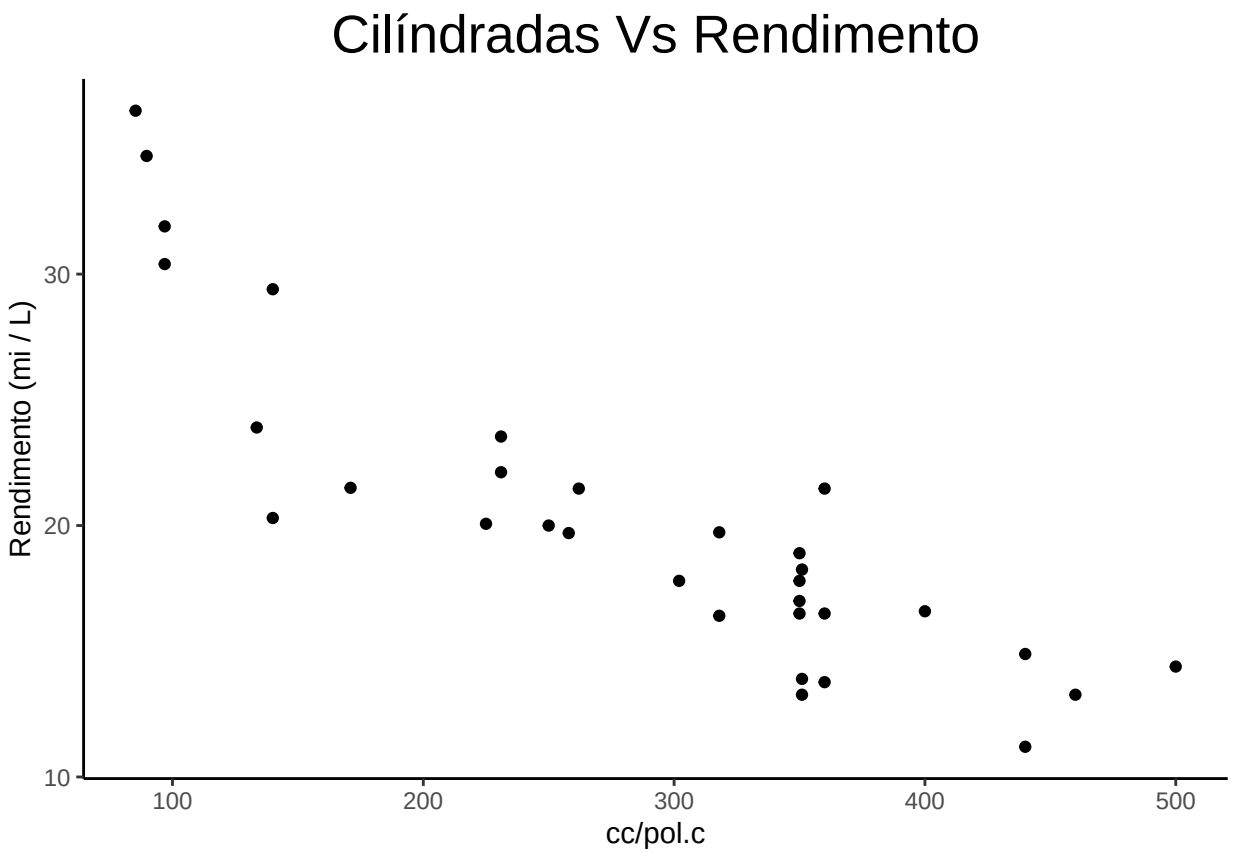
```
dim(dados)
```

```
## [1] 32 12
```

Análise Exploratória Básica

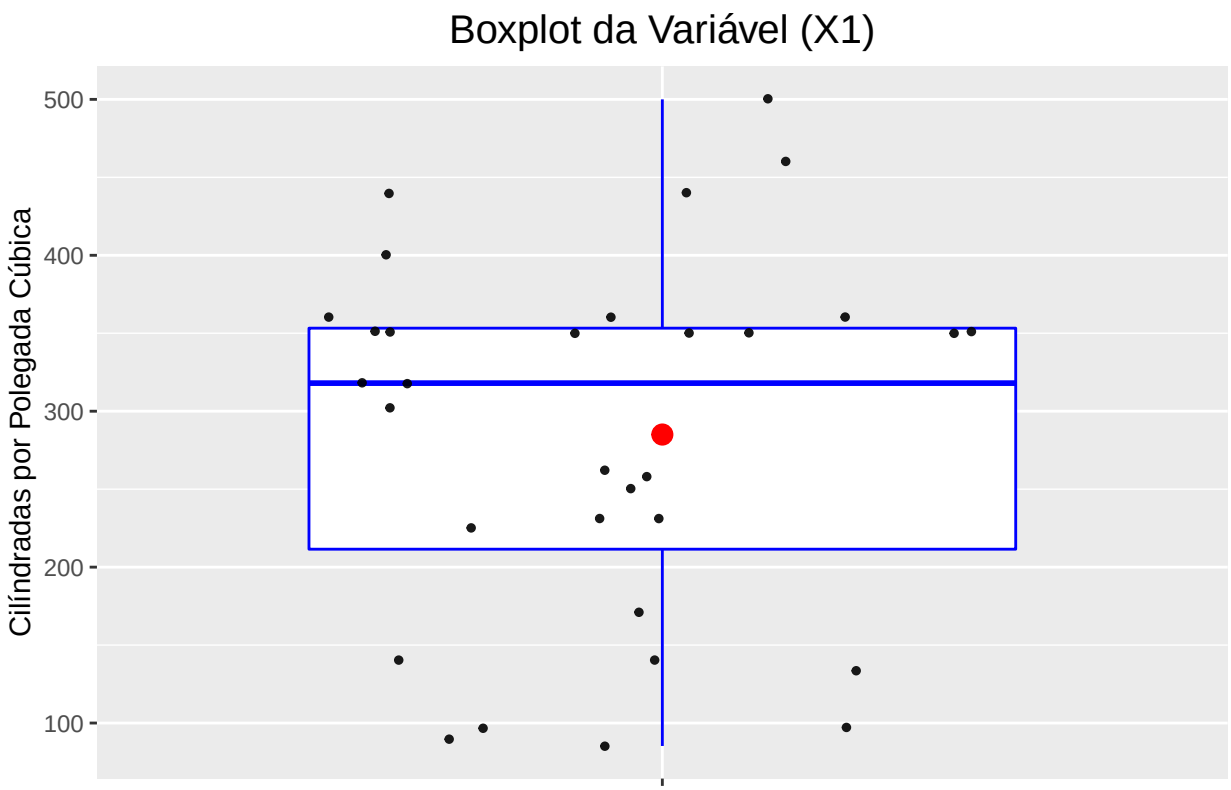
Gráfico da Dispersão entre x1 (cilindrada do motor) e y (rendimento da gasolina)

```
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_classic() +
  theme(plot.title = element_text(size = 20, hjust = .5))
```



Perceba que existe uma clara relação linear entre as duas variáveis, representada pela linha azul. Conforme as cilindradas do motor aumentam, o rendimento tende a diminuir.

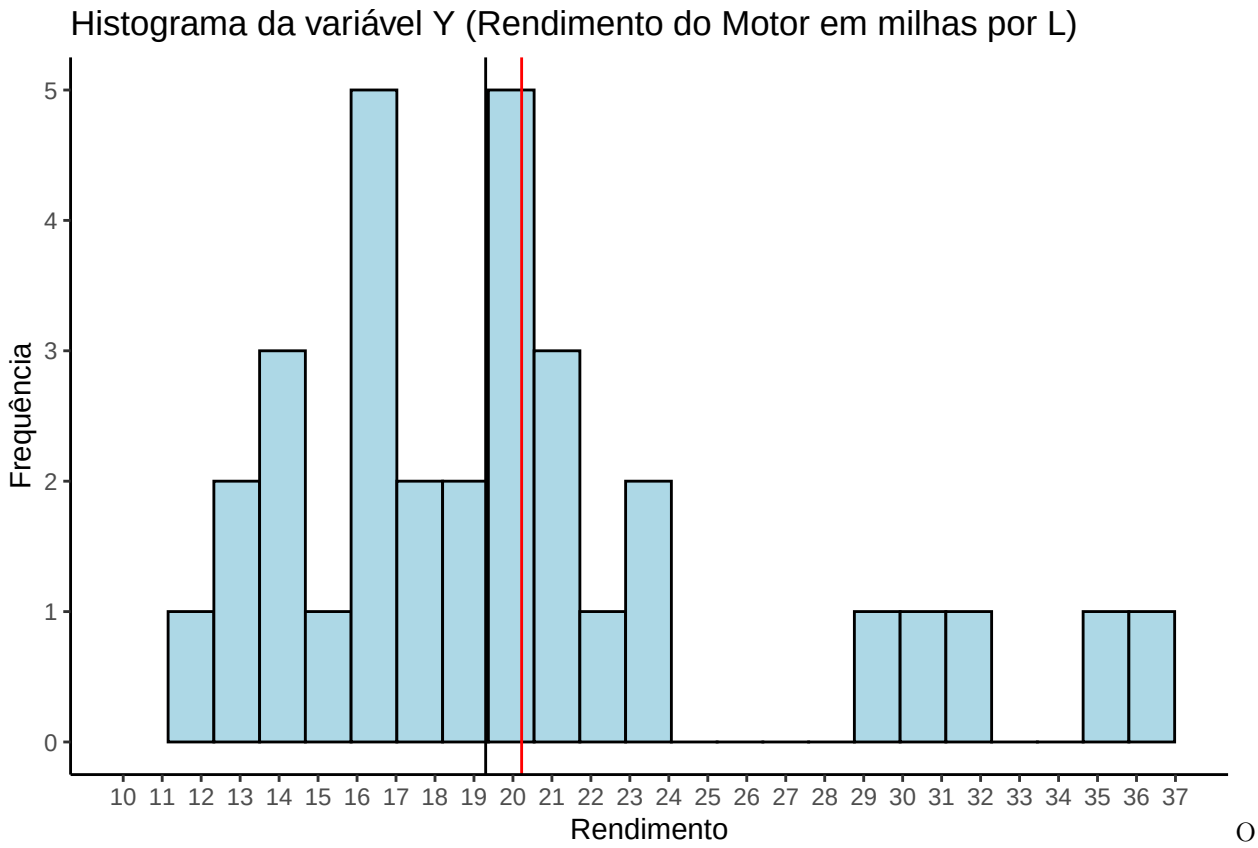
```
dados %>%
  ggplot(aes(x = "", y = dados$x1)) +
    geom_boxplot(color = "blue") +
    stat_summary(fun = mean, geom = "point", shape = 20, size = 5, color = "red", fill = "red") +
    geom_jitter(color="black", size=1, alpha=.9) +
    theme(plot.title = element_text(size = 15, hjust = .5)) +

    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=dados$y)) +
    geom_histogram(color = "black", fill = "lightblue", bins = 24) + # xlab("Rendimento") + ylab("Frequ
    geom_vline(aes(xintercept=mean(dados$y)), color = "red", linetype = "solid") +
    geom_vline(aes(xintercept=median(dados$y)), color = "black", linetype = "solid") +
    labs(title = "Histograma da variável Y (Rendimento do Motor em milhas por L)") +
    scale_x_continuous("Rendimento", limits = c(10,37,1), breaks = c(10:37)) +
    ylab("Frequência") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme_classic()
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



O histograma acima representa a distribuição da variável y. As linhas verticais vermelha e preta representam, respectivamente, a média e a mediana.

```
dados %>% filter(dados$y > 29.4)
```

```
## # A tibble: 4 x 12
##       y    x1    x2    x3    x4    x5    x6    x7    x8    x9   x10   x11
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  34.7  89.7    70    81   8.2  3.9      2     4  156.  64    1905     0
## 2  30.4  96.9    75    83   9    4.3      2     5  165.  65    2320     0
## 3  36.5  85.3    80    83   8.5  3.89     2     4  161.  62.2  2009     0
## 4  31.9  96.9    75    83   9    4.3      2     5  165.  61.8  2275     0
```

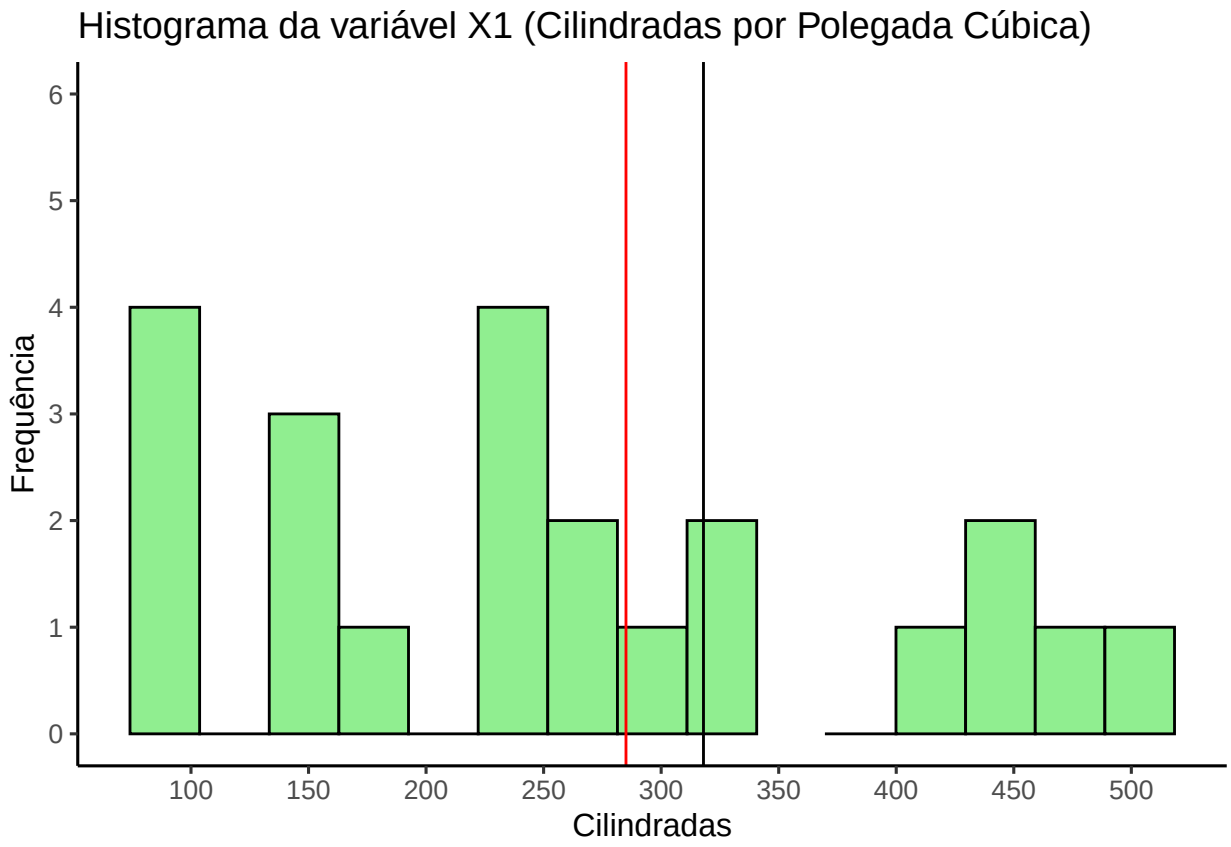
A partir do histogrma e da tabela acima, concluimos que os 4 outliers referem-se aos valores: 30.4, 31.9, 34.7, e 36.5

Vamos verificar a distribuição da variável x1

```
dados %>%
  ggplot(aes( x=dados$x1)) +
    geom_histogram(color = "black", fill = "lightgreen", bins = 15) + # xlab("Rendimento") + ylab("Freq
    geom_vline(aes(xintercept=mean(dados$x1)), color = "red", linetype = "solid") +
    geom_vline(aes(xintercept=median(dados$x1)), color = "black", 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", breaks = c(0:8), limits = c(0,6)) +
  #limits = c(85,500,1)) +
```

```
    ylab("Frequência") +
    theme(plot.title = element_text(hjust = 0.5)) +
    theme_classic2()
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```



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 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]   64.95625   64.95625  -35.04375   65.95625  -60.04375  154.95625
##  [7]  -54.04375  -23.04375 -195.34375 -188.14375   64.95625 -199.74375
## [13] -114.04375  -27.04375 -145.04375   16.95625  214.95625  154.95625
## [19]   64.95625   32.95625  -54.04375   74.95625  114.95625 -188.14375
## [25] -145.04375  174.95625 -151.44375   32.95625   65.95625   65.95625
## [31]   74.95625   74.95625
```

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

```
##  [1]  4219.3144  4219.3144  1228.0644  4350.2269  3605.2519 24011.4394
##  [7]  2920.7269   531.0144 38159.1807 35398.0707  4219.3144 39897.5657
## [13] 13005.9769   731.3644 21037.6894   287.5144 46206.1894 24011.4394
## [19]  4219.3144  1086.1144  2920.7269  5618.4394 13214.9394 35398.0707
## [25] 21037.6894 30609.6894 22935.2094  1086.1144  4350.2269  4350.2269
## [31]  5618.4394  5618.4394
```

```
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] -1.323125 -3.223125 -0.223125 -1.973125 -0.153125 -9.023125  1.896875
##  [8]  1.246875 14.476875 10.176875 -3.723125 16.276875  1.276875 -0.523125
## [15]  0.076875 -2.423125 -5.833125 -5.333125 -2.423125 -3.813125  3.316875
## [22]  1.246875 -3.633125 11.676875  9.176875 -6.953125  3.676875 -0.493125
## [29] -6.323125 -6.953125 -6.453125 -3.723125
```

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

```
##  [1] 1.750660e+00 1.038853e+01 4.978477e-02 3.893222e+00 2.344727e-02
##  [6] 8.141678e+01 3.598135e+00 1.554697e+00 2.095799e+02 1.035688e+02
## [11] 1.386166e+01 2.649367e+02 1.630410e+00 2.736598e-01 5.909766e-03
## [16] 5.871535e+00 3.402535e+01 2.844222e+01 5.871535e+00 1.453992e+01
## [21] 1.100166e+01 1.554697e+00 1.319960e+01 1.363494e+02 8.421503e+01
## [26] 4.834595e+01 1.351941e+01 2.431723e-01 3.998191e+01 4.834595e+01
## [31] 4.164282e+01 1.386166e+01
```

```
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,] 426103.3 1237.544 -20180.07
```

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

Estimacao dos parametros

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

Calculando o valor do coeficiente angular β_1

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

Calculando o valor do intercepto β_0

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

Calculando O estimador de sigma^2 nao viesado.

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

```r
# 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)
```

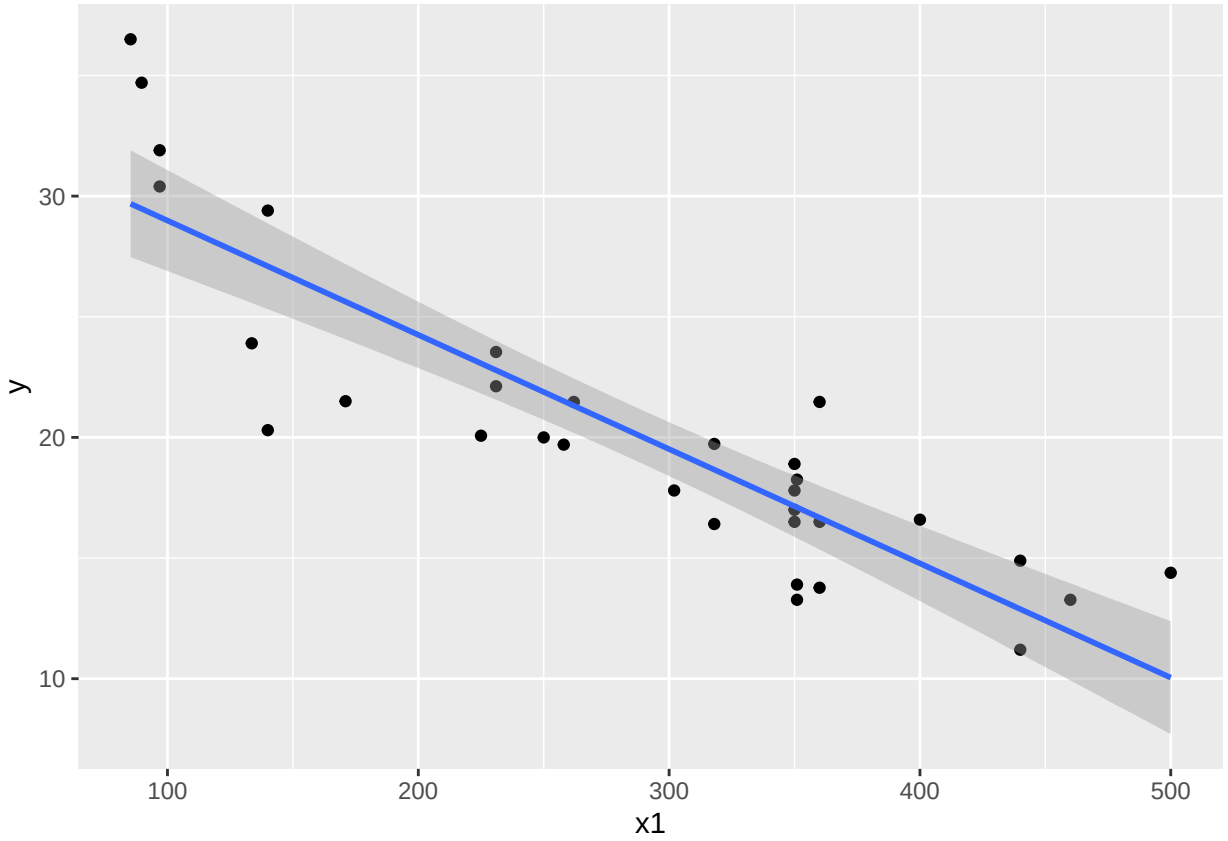
```r
fit <- lm(y~x1, data = dados)
#gráfico reta ajustada
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1, data = dados)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7923 -1.9752  0.0044  1.7677  6.8171
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.722677   1.443903   23.36  < 2e-16 ***
## x1          -0.047360   0.004695  -10.09 3.74e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.065 on 30 degrees of freedom
## Multiple R-squared:  0.7723, Adjusted R-squared:  0.7647
## F-statistic: 101.7 on 1 and 30 DF,  p-value: 3.743e-11
```

```r
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x1         1 955.72  955.72  101.74 3.743e-11 ***
## Residuals 30 281.82    9.39
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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




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

```
## [1] 33.7227
```

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

```
## [1] -0.0474
```

Consequentemente, a reta ajustada é:

$$\widehat{Y}_i = 1224.043 - 0.7769 * X_i$$

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
```

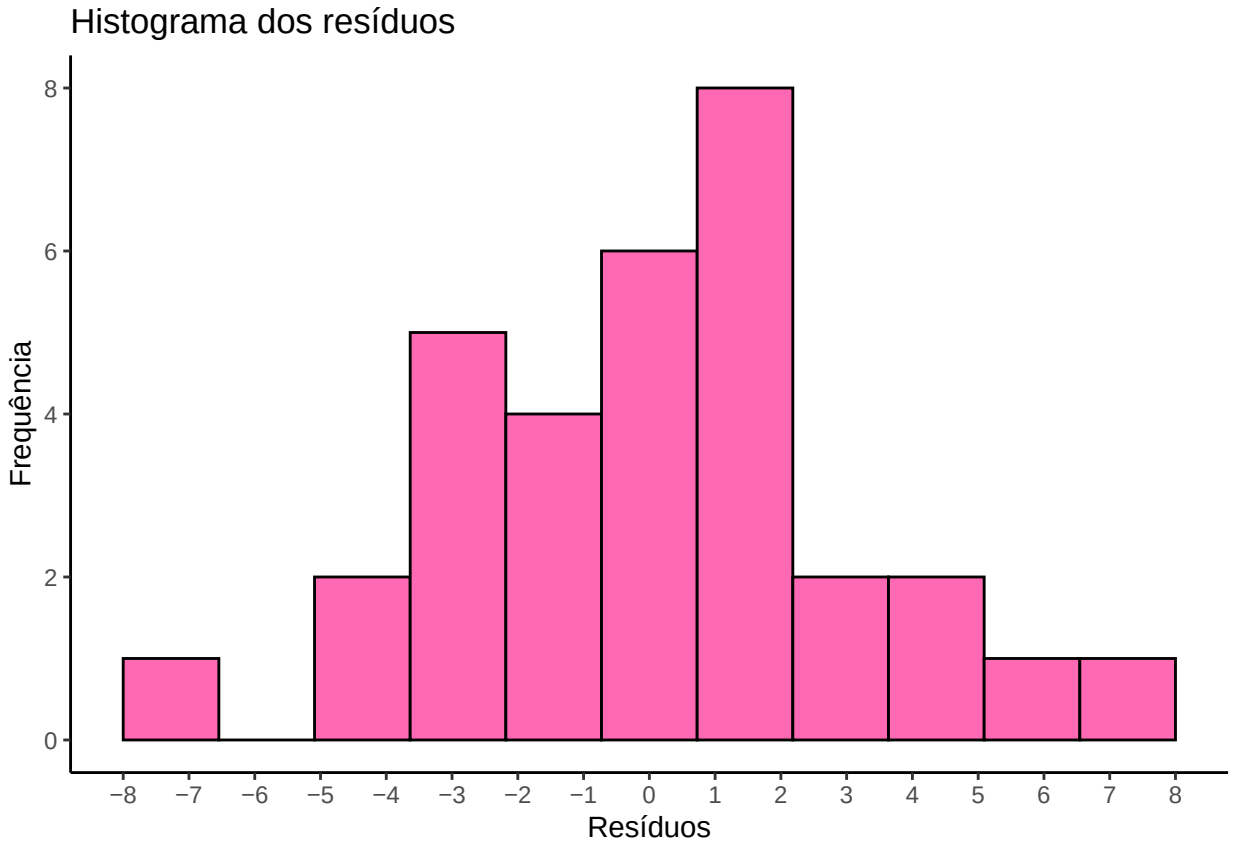
Utilizando o comando *summary*, podemos observar as principais medidas descritivas da variável, o que nos auxilia para a análise da mesma.

```
summary(res)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -6.792336 -1.975169  0.004354  0.000000  1.767670  6.817095
```

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

```
# Histograma
n_res <- length(res)

ggplot(dados, aes(x = res)) +
  ylab("Frequência") +
  geom_histogram(color = "black", fill = "#FF69B4", bins=12)+
  labs(title = "Histograma dos resíduos")+
  scale_x_continuous("Resíduos", limits = c(-8,8,1), breaks = c(-8:8))+
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic()
```



Desse modo, temos que a análise dos resíduos é ———

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 α= 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
```

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 constrção 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
```

Realizando o teste, temos que H0 é rejeitado, logo, β_0 é diferente de zero.

5. intervalos de confiança

```
confint(fit, level = 0.99)
```

```
##                   0.5 %      99.5 %
## (Intercept) 29.75195035 37.69340302
## x1          -0.06027187 -0.03444729
```

6. intervalos de predição

```
#predict(fit, newdata = new.dat, interval = 'prediction')
```

Galera daqui ppra baixo só colei o markdown do danilo

Calcule Sxx, Syy, Sxy.

Note que as expressões para Sxx, Syy e Sxy foram apresentadas na Aula 3 e são dadas por:

$Sxx = \sum_{i=1}^{n}(X_i - \overline{X})^2 = \sum_{i=1}^{n} X_i^2 - n\overline{X}^2$,

$Syy = \sum_{i=1}^{n}(Y_i - \overline{Y})^2 = \sum_{i=1}^{n} Y_i^2 - n\overline{Y}^2$,

$Sxy = \sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y}) = \sum_{i=1}^{n}(X_iY_i) - n\overline{XY} = \sum_{i=1}^{n}(X_i - \overline{X})Y_i$

Vamos cacular primeiramente $\overline{X}$, logo subtraimos ele da variavél, elevamos ao quadrado cada termo e sumarmos para obter.

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

```
##  [1]   64.95625   64.95625  -35.04375   65.95625  -60.04375  154.95625
##  [7]  -54.04375  -23.04375 -195.34375 -188.14375   64.95625 -199.74375
## [13] -114.04375  -27.04375 -145.04375   16.95625  214.95625  154.95625
## [19]   64.95625   32.95625  -54.04375   74.95625  114.95625 -188.14375
## [25] -145.04375  174.95625 -151.44375   32.95625   65.95625   65.95625
## [31]   74.95625   74.95625
```

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

```
##  [1]  4219.3144  4219.3144  1228.0644  4350.2269  3605.2519 24011.4394
##  [7]  2920.7269   531.0144 38159.1807 35398.0707  4219.3144 39897.5657
## [13] 13005.9769   731.3644 21037.6894   287.5144 46206.1894 24011.4394
## [19]  4219.3144  1086.1144  2920.7269  5618.4394 13214.9394 35398.0707
## [25] 21037.6894 30609.6894 22935.2094  1086.1144  4350.2269  4350.2269
## [31]  5618.4394  5618.4394
```

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

```
## [1] 426103.3
```

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

```
##  [1] -1.323125 -3.223125 -0.223125 -1.973125 -0.153125 -9.023125  1.896875
##  [8]  1.246875 14.476875 10.176875 -3.723125 16.276875  1.276875 -0.523125
## [15]  0.076875 -2.423125 -5.833125 -5.333125 -2.423125 -3.813125  3.316875
## [22]  1.246875 -3.633125 11.676875  9.176875 -6.953125  3.676875 -0.493125
## [29] -6.323125 -6.953125 -6.453125 -3.723125
```

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

```
##  [1] 1.750660e+00 1.038853e+01 4.978477e-02 3.893222e+00 2.344727e-02
##  [6] 8.141678e+01 3.598135e+00 1.554697e+00 2.095799e+02 1.035688e+02
## [11] 1.386166e+01 2.649367e+02 1.630410e+00 2.736598e-01 5.909766e-03
## [16] 5.871535e+00 3.402535e+01 2.844222e+01 5.871535e+00 1.453992e+01
## [21] 1.100166e+01 1.554697e+00 1.319960e+01 1.363494e+02 8.421503e+01
## [26] 4.834595e+01 1.351941e+01 2.431723e-01 3.998191e+01 4.834595e+01
## [31] 4.164282e+01 1.386166e+01
```

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

```
## [1] 1237.544
```

```
Sxy=sum((x1-xbarra)*(y-ybarra))
Sxy
```

```
## [1] -20180.07
```

```
cbind(Sxx,Syy,Sxy)
```

```
##           Sxx      Syy       Sxy
## [1,] 426103.3 1237.544 -20180.07
```

##Ajuste do modelo de regressão linear simples e gráfico da reta ajustada

Note que na aula 3 temos feito passo a passo do ajuste do modelos mediante o método de minimos quadrados (ajustar um modelo significa mesmo estimar seus parâmetros). Nosso modelo liear simples é da forma:

$$Y = \beta_0 + \beta_1 X + \xi$$

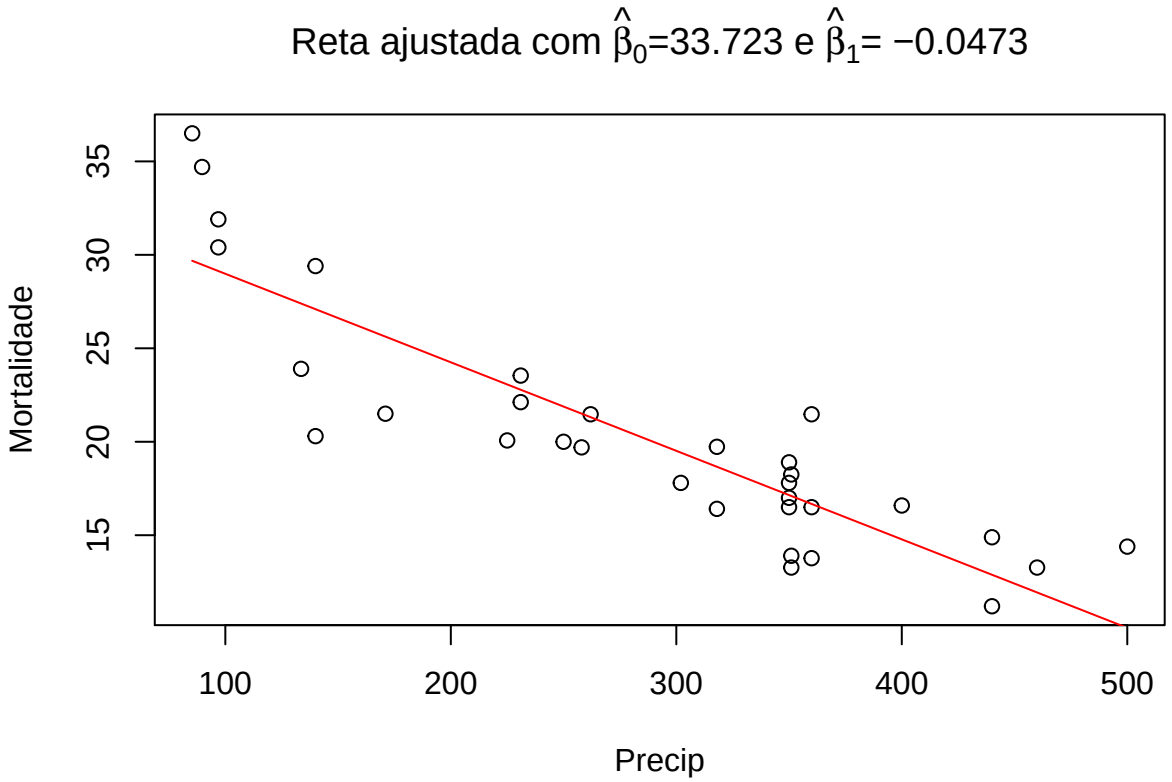
onde aplicando a técnica de mínimos quadrados teremos estimadores para nossos parâmetros dados pelas seguintes expressões: $\widehat{\beta_0} = \overline{Y} - \widehat{\beta_1}\overline{X}$
$\widehat{\beta_1} = \dfrac{Sxy}{Sxx}$. Portanto,

```
beta1est=(Sxy)/(Sxx)
beta0est=ybarra-(beta1est*xbarra)
cbind(beta0est,beta1est)
```

```
##      beta0est    beta1est
## [1,] 33.72268 -0.04735958
```

Gráfico Reta Ajustada

```
plot(x1,y , #ylim = c(750,1150), xlim = c(9,65),
     main = expression(paste("Reta ajustada com ",
                             hat(beta)[0],"=33.723",
                             " e ", hat(beta)[1],"= -0.0473")),
     xlab = "Precip", ylab = "Mortalidade")
curve(beta0est + beta1est*x, add = T, col = 'red')
```

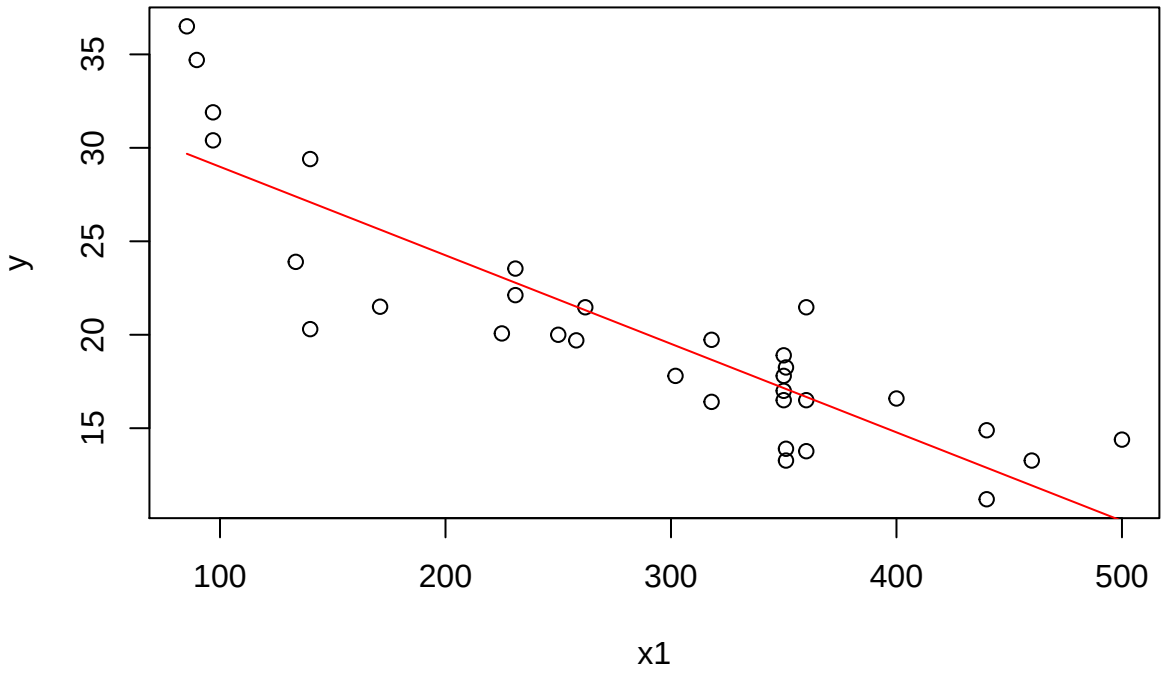


Consequentemente, a reta ajustada é

$$\widehat{Y}_i = 821.7546 + 3.174002 * X_i$$

gráfico simples da reta ajustada usando b0 e b1

```
plot(x1,y)
curve(b0_est + b1_est*x, add = T, col = 'red')
```



Estimando σ^2

O estimador de σ^2 não viesado eh obtido pelo quadrado médio do resíduo (QMres) apresentado passo a passo na aula 5. Para termos QMreg precisamos do SQres.

SQreg

```
SQreg <- beta1est*Sxy
SQreg
```

```
## [1] 955.7197
```

SQtotal

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

```
## [1] 1237.544
```

SQRes = SQtotal - SQreg

```
SQres <- SQtotal - SQreg
SQres
```

```
## [1] 281.8244
```

MQres

```
#estimador nao viesado de sigma^2
QMres <- SQres/(n-2)
SigmaQuadradoEst<-QMres
SigmaQuadradoEst
```

```
## [1] 9.394146
```

```
#Estimativas para Beta0, Beta1 e Sigma^2
cbind(beta0est, beta1est, SigmaQuadradoEst)
```

```
##      beta0est    beta1est SigmaQuadradoEst
## [1,] 33.72268 -0.04735958         9.394146
```

Resíduos

O valor dos $\widehat{Y}$'s e o valor dos resíduos para o seu modelo, faça um resumo e Histograma dos resíduos e faça análise da distribuição destes.

Para sabermos os resíduos do nosso modelo vamos calcular o valor predito primeiro.

```
# valor predito
y_pred <- beta0est + beta1est*x1
```

```r
# Ou ainda
y_pred <- mean(y) + beta1est*(x1 - mean(x1))
```

Resíduos

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

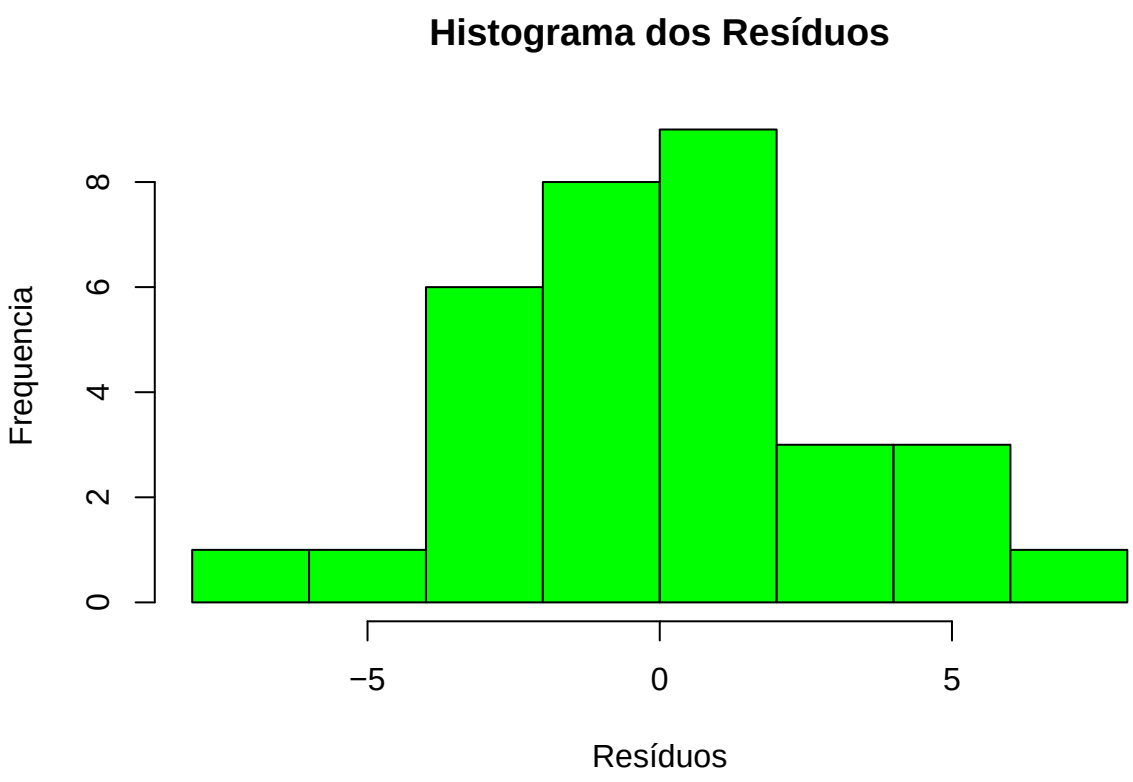
```
##  [1]  1.7531756 -0.1468244 -1.8827822  1.1505352 -2.9967717 -1.6844624
##  [7] -0.6626142  0.1555327  5.2254775  1.2664664 -0.6468244  6.8170953
## [13] -4.1241889 -1.8039056 -6.7923358 -1.6200842  4.3471123  2.0055376
## [19]  0.6531756 -2.2523309  0.7573858  4.7967714  1.8111545  2.7664664
## [25]  2.3076642  1.3327292 -3.4954371  1.0676691 -3.1994648 -3.8294648
## [31] -2.9032286 -0.1732286
```

```r
summary(res)
```

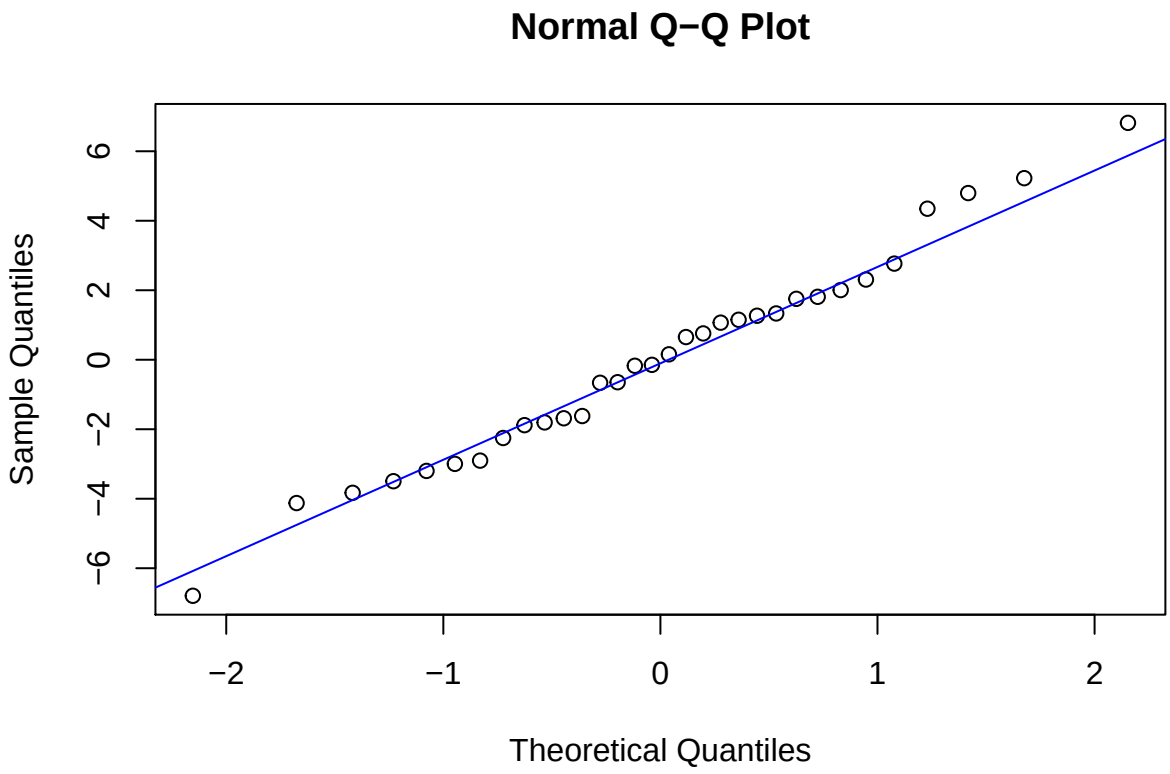
```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -6.792336 -1.975169  0.004354  0.000000  1.767670  6.817095
```

Histograma dos Resíduos

```r
hist(res, main = "Histograma dos Resíduos", col = "green",
    # xlim = c(-151, 130), ylim = c(0, 30),
     ylab = "Frequencia", xlab = "Resíduos")
```



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



Teste de Hipótese

Obteremos os teste de hipóteses para β_0 e β_1 com a decisão de rejeitar ou não H_0.

Sera que $\beta_0 = 0$ estatísticamente? E sera que $\beta_1 = 0$ estatísticamente?. Para isso precisamos calcular o estimador de σ^2 (Utilizando $\alpha = 5\%$).

Teste de Hipótese para β_1

Testando se $\beta_1 = 0$, Lembrando que β_1 estimado tem distribuição Normal com media β_1 e variância= (σ^2/Sxx).

Como nâo temos o valor de σ^2 temos que estima-lo.

$H_0 : \beta_1 = 0$ vs $H_1 : \beta_1$ não eh zero $(\beta_1 \neq 0)$.

```r
dp_b1 <- (sqrt(QMres/Sxx))
t0_b1 <- beta1est/dp_b1
```

Rejeitamos H_0 se t0_b1 $< t_1$ ou t0_b1 $> t_2$, em que t_1 eh o quantil $\alpha/2$ da Distribuição t com $n-2$ G.L. e t_2 eh o quantil $1-\alpha/2$ da Distribuição t com $n-2$ G.L. útilizando $\alpha = 5\%$.

```r
alpha <- 0.05
t1 <- qt(alpha/2,n-2)
t2 <- qt(1-alpha/2,n-2)
```

Regra de Decisão

```
if(t0_b1 < t1 || t0_b1>t2){
  cat("Rejeita-se H0")
}
```

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

Como podemos ver pelo teste, rejeitamos H_0. Ou seja, rejeitamos a hípotese que o valor do coeficiente $\beta_1 = 0$.

Teste de Hipótese para β_0

Testando se $\beta_0 = 0$. Lembrando que β_0 estimado tem Distribuição Normal com média β_0 e variância= $(\sigma^2 * ((1/n) + \overline{X}/Sxx))$ como não temos o valor de σ^2 temos que estima-lo.

$H_0 : \beta_0 = 0$ vs $H_1 : \beta_0$ não eh zero $(\beta_0 \neq 0)$

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

t0_b0 <- beta0est/dp_b0
```

Rejeitamos H_0 se t0_b0 $< t_1$ ou t0_b0 $> t_2$, em que t_1 eh o quantil $\alpha/2$ da Distribuição t com $(n-2)$ G.L. t_2 eh o quantil $1-\alpha/2$ da Distribuicao t com $(n-2)$ G.L. útilizando $\alpha = 5\%$

```
alpha <- 0.05
t1 <- qt(alpha/2,n-2)
t2 <- qt(1-alpha/2,n-2)
```

Regra de Decisão

```
if(t0_b0 < t1 || t0_b0>t2){
  cat("Rejeita-se H0")
}
```

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

Como podemos ver pelo teste, rejeitamos H_0. Ou seja, rejeitamos a hípotese que o valor do intercepto $\beta_0 = 0$.

vou editar a partir daqui

Intervalos de Confiança

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

Calculando intervalo de Confiança para β_1

```
b1_min <- beta1est-t2*dp_b1
b1_max <- beta1est-t1*dp_b1
IC_b1_est <- cbind(b1_min, b1_max)
IC_b1_est
```

```
##           b1_min      b1_max
## [1,] -0.05694883 -0.03777032
```

(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 β_0

```
b0_min <- beta0est-t2*dp_b0
b0_max <- beta0est-t1*dp_b0
IC_b0_est <- cbind(b0_min, b0_max)
IC_b0_est
```

```
##        b0_min   b0_max
## [1,] 30.77383 36.67152
```

Calculando intervalo de confiança para σ^2

Lembrado que $SQres/\sigma^2$ 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,] 5.998913 16.78448
```

Calculando intervalo de confiança para a esperança de y

(valor medio da variavel resposta para um valor particular da cov., MIy|X0).

Lembrando:

1. O valor médio da variavél 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. $t(1-\alpha/2, n-2)$, $\alpha = 0,05$.

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

```
X0 <- mean(x1) # poderia ser outro valor

v_medio <- (mean(y)+beta1est* (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,]    19.11658    21.32967
```

E se quisessemos predizer 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 variavèl 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 <- beta0est + beta1est*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,]   26.79975   39.50898
```

Análise de Variâcia

A Análise de Variâcia 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 <- beta1est*Sxy
SQreg
```

```
## [1] 955.7197
```

Soma do quadrado total

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

```
## [1] 1237.544
```

Soma do quadrado do resíduo

```
SQres <- sum((y-mean(y))^2) - beta1est*Sxy
SQres
```

```
## [1] 281.8244
```

```
QMreg <- SQreg
QMreg
```

```
## [1] 955.7197
```

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

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

```
## [1] 101.7357
```

Quantil da Distribuição F-Snedecor

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

```
## [1] 3.743041e-11
```

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

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

ou poderiamos ter calculado

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

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

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

Anova usando funções do R

```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
```

```r
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

```r
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.