SME0820 - Modelos de Regressão e Aprendizado Supervisionado I

Trabalho 3 - Grupo 3

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

Dezembro 2021

Contents

Objetivo

Este trabalho tem como objetivo ajustar um modelo de regressão linear múltipla a um conjunto de dados.

Conjunto de dados

O dataset contém dados de um experimento para determinar **pressão**, **temperatura**, **fluxo de CO2**, **umidade** e **tamanho da partícula de amendoim** sob o **rendimento total de aceite por lote de amendoim**. [rendimento (y)].

Significância: 97%

```
dados <- read_csv("dados/data-table-B7.csv", locale = locale(decimal_mark = ","))
```

```
## Rows: 16 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (6): x1, x2, x3, x4, x5, y
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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

# Renomeando as colunas
names(dados) <- c("Pressao", "Temp", "FluxoCO2", "Umidade", "Tamanho", "y")

head(dados)
```

```
## # A tibble: 6 x 6
##   Pressao  Temp FluxoCO2 Umidade Tamanho     y
##     <dbl> <dbl>    <dbl>   <dbl>   <dbl> <dbl>
## 1     415    25        5      40    1.28    63
## 2     550    25        5      40    4.05    21
## 3     415    95        5      40    4.05    36
## 4     550    95        5      40    1.28    99
## 5     415    25       15      40    4.05    24
## 6     550    25       15      40    1.28    66
```

Temos cinco covariáveis quantitativas e a coluna y corresponde a nossa variável preditora que determina **o rendimento total de azeite por lote de amendoim**

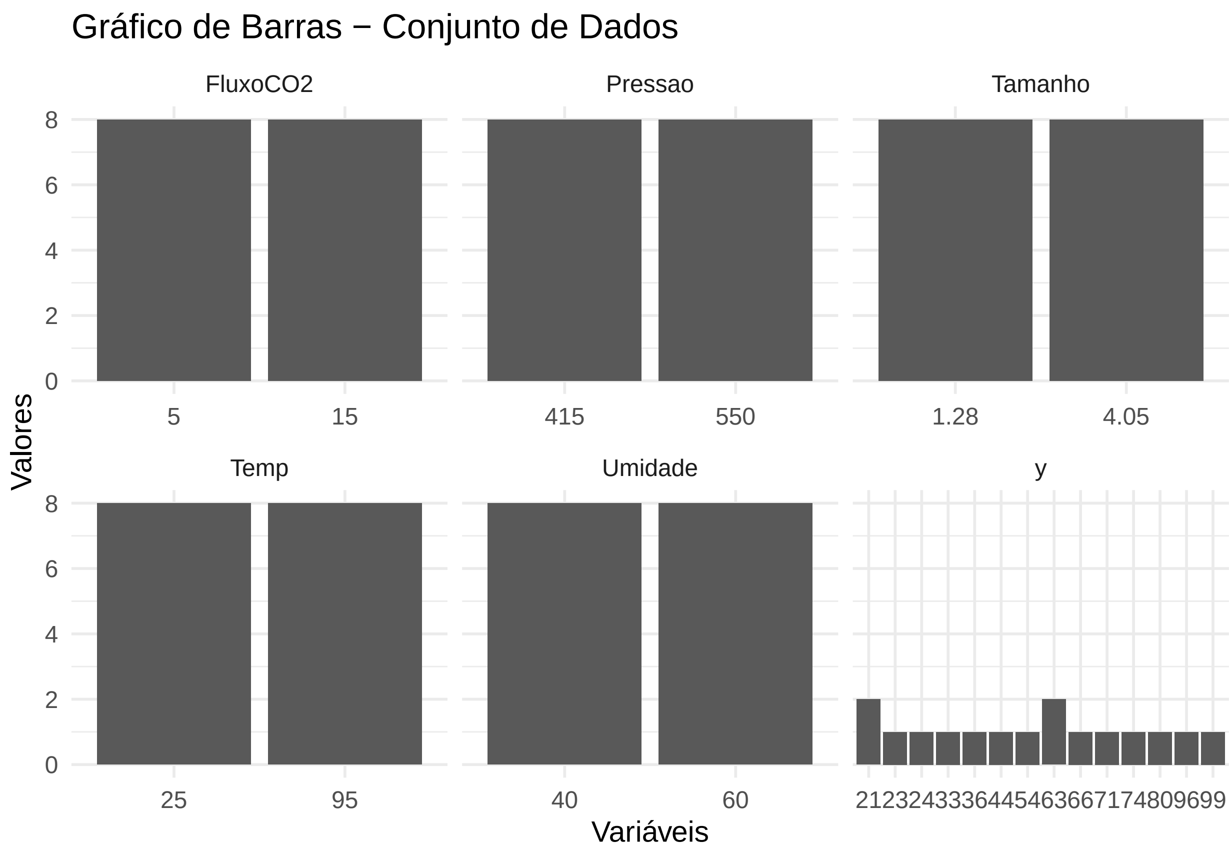
1. Análise Descritiva dos dados

- Y : Rendimento total de azeite por lote de amendoim*.
- X_1 : Pressão
- X_2 : Temperatura

- X_3 : Fluxo de CO2
- X_4 : Umidade
- X_5 : Tamanho

Gráficos de barras

```
dados %>%
  pivot_longer(cols = everything()) %>%
  ggplot() +
  geom_bar(aes(x = as_factor(value)), stat = "count") +
  facet_wrap(~name, scales = "free_x") +
  labs(
    x = "Variáveis",
    y = "Valores",
    title = "Gráfico de Barras - Conjunto de Dados"
  ) +
  theme_minimal()
```



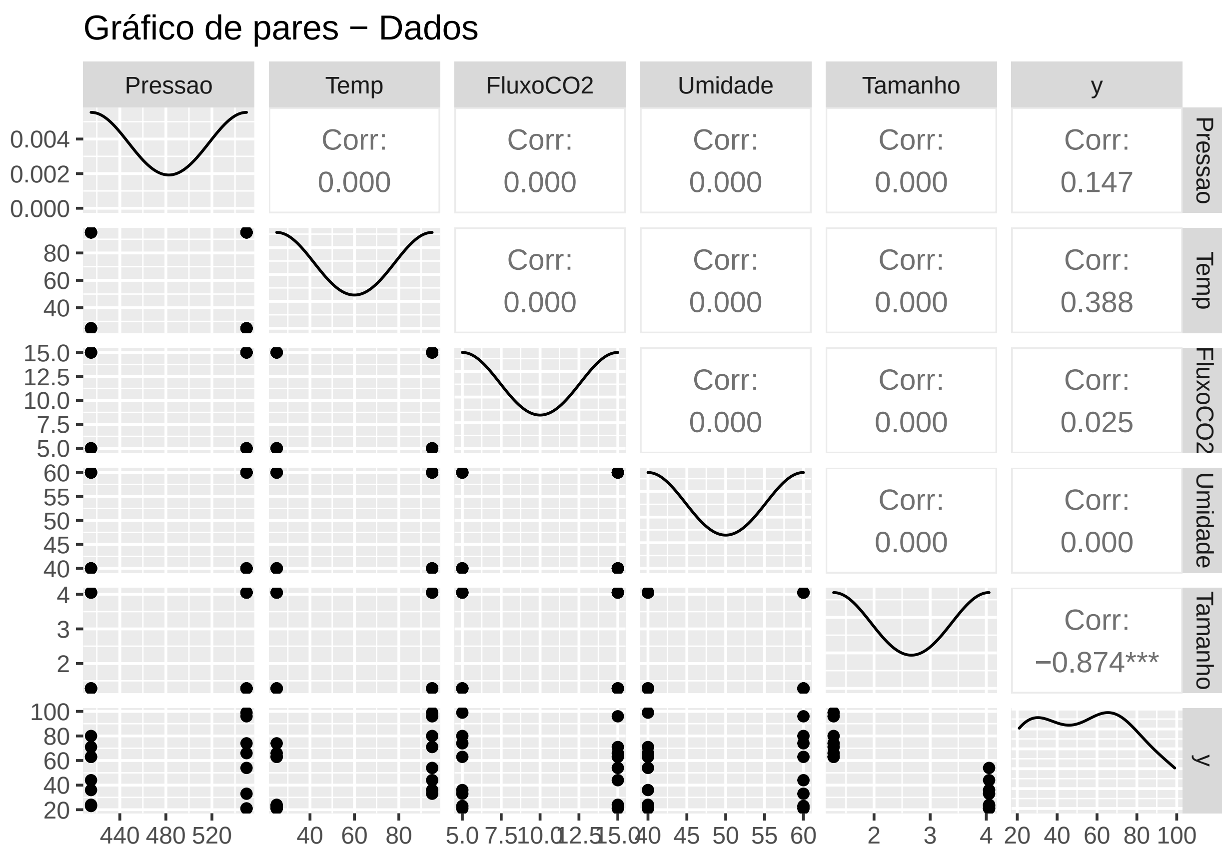
A partir dos gráficos de barras, podemos ver que nossas cinco covariáveis, apesar de serem quantitativas, assumem apenas dois valores, com a mesma proporção. A única variável que assume mais valores do que isso é y, que apararenta ter uma distribuição quase uniforme.

Outros gráficos que comparam as relações entre nossas variáveis é o gráfico de coordenadas paralelas e nossa matriz de correlação, ambos também explicitando a falta de correlação entre as covariáveis.

Correlação de Pearson entre as covariáveis e y

Fazendo uso das correlações, podemos dispor graficamente uma matriz de gráficos para expor as relações entre as variáveis, de modo que teremos densidades de frequência nas diagonais, gráficos de dispersão no painel triangular inferior e coeficientes das correlações no superior, de modo que o tamanho dos números é condicionado ao valor da correlação.

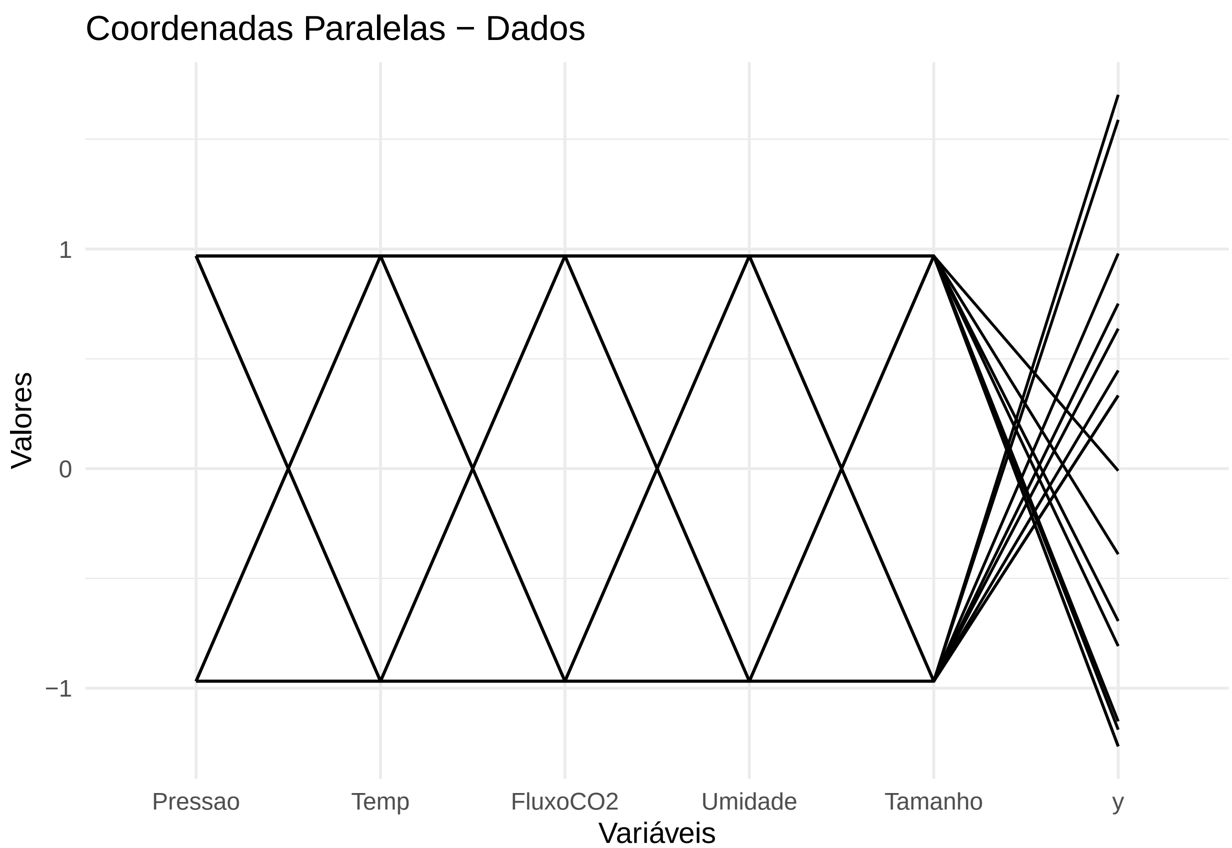
```
ggpairs(dados) + ggtitle("Gráfico de pares - Dados")
```



No gráfico de pares acima, podemos observar as correlações (ou ausência delas) de todas as covariáveis entre si e com a variável preditora. Analisando esses resultados, vemos que nenhuma das covariáveis se correlacionam entre si. Além disso, a maioria apresenta uma correlação muito baixa com a variável preditora - com exceção de x5 (Tamanho) com y.

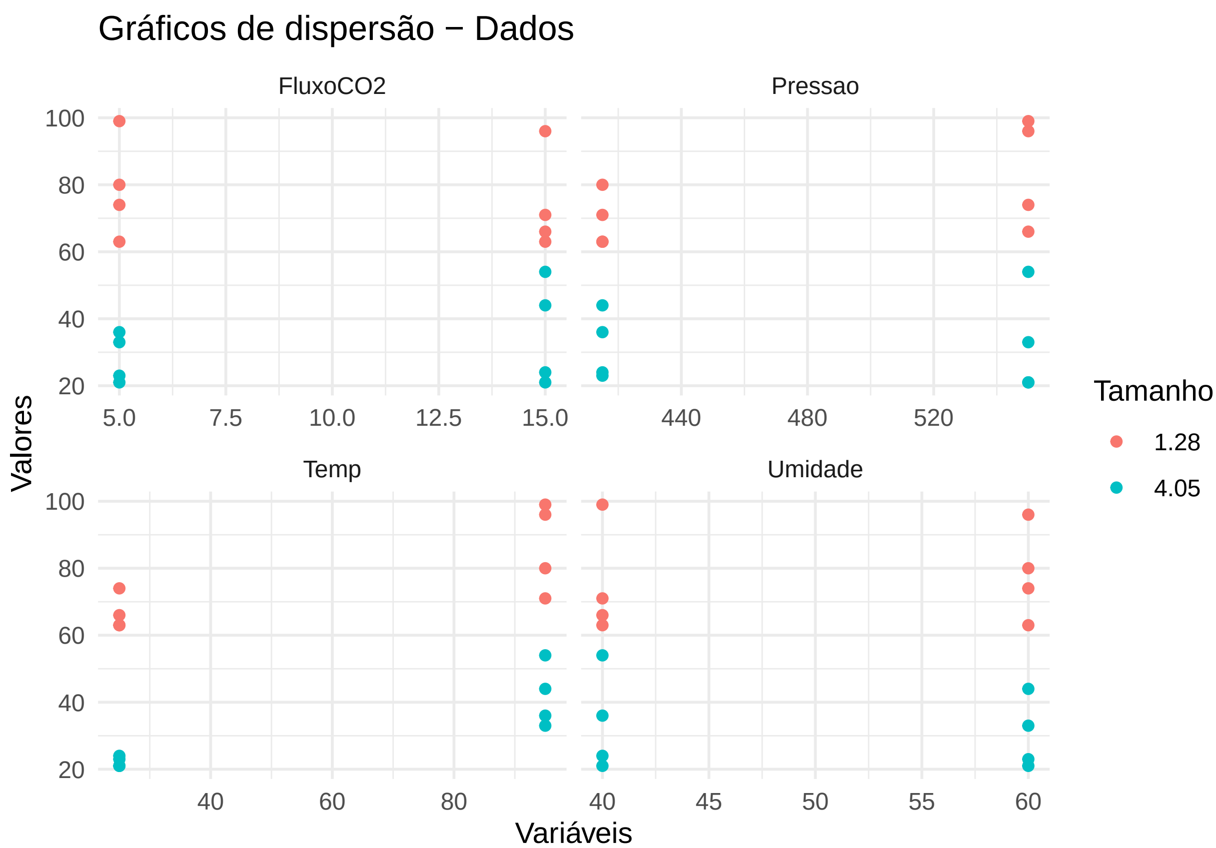
Essa ausência de correlação pode ser explicada pelo comportamento em "X" da maior parte das covariáveis, que também pode ser notado através do gráfico de coordenadas paralelas:

```
ggparcoord(dados) + labs(
  x = "Variáveis",
  y = "Valores",
  title = "Coordenadas Paralelas - Dados"
) +
  theme_minimal()
```



Definindo a covariável Tamanho como mapeamento para cor, podemos dispor outra versão dos gráficos de pares:

```
dados %>%
  pivot_longer(!c(Tamanho, y)) %>%
  ggplot(aes(y = y, color = as_factor(Tamanho))) +
  geom_point(aes(x = value)) +
  facet_wrap(~name, scales = "free_x") +
  labs(
    x = "Variáveis",
    y = "Valores",
    title = "Gráficos de dispersão - Dados",
    color = "Tamanho"
  ) +
  theme_minimal()
```



Note como a covariável Tamanho foi capaz de separar bem as variáveis no eixo y, enquanto que o mesmo feito não foi alcançado no eixo x. Temos aqui fortes indícios de independência entre as covariáveis, e o melhor modelo talvez não seja o que contenha todas elas, como veremos mais adiante.

2. Matriz Hat

- valores da matriz hat
- apresentar os valores e analisar
- determinar possiveis outliers e pontos de alavanca

```
X <- matrix(c(rep(1,n), dados$Temp, dados$Tamanho), ncol = 3, nrow = n, byrow = FALSE)
#X
```

```
Y <- matrix(dados$y, ncol = 1, nrow = length(dados$y))
#Y
```

```
H <- X %*% solve(t(X) %*% X) %*% t(X)
h <- diag(H)
summary(h)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1875  0.1875  0.1875  0.1875  0.1875  0.1875
```

3. Análise de resíduos

- ajuste do modelo
- resumo do modelo
- resíduos vs valores ajustados
- qqplot
- raiz de resituos estandartizados versus valores ajustados
- distancia de cook (residuos estandartizados vs pontos de alavanca)

```
# Modelo  reduzido do Ex 2

fit <- lm(y ~ Temp + Tamanho, data = dados)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ Temp + Tamanho, data = dados)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.375  -4.188  -0.875   3.438  12.625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  80.13461    5.69146  14.080 3.01e-09 ***
## Temp          0.28214    0.05883   4.796 0.000349 ***
## Tamanho     -16.06498    1.48659 -10.807 7.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.236 on 13 degrees of freedom
## Multiple R-squared:  0.9149, Adjusted R-squared:  0.9018
## F-statistic: 69.89 on 2 and 13 DF,  p-value: 1.107e-07
```

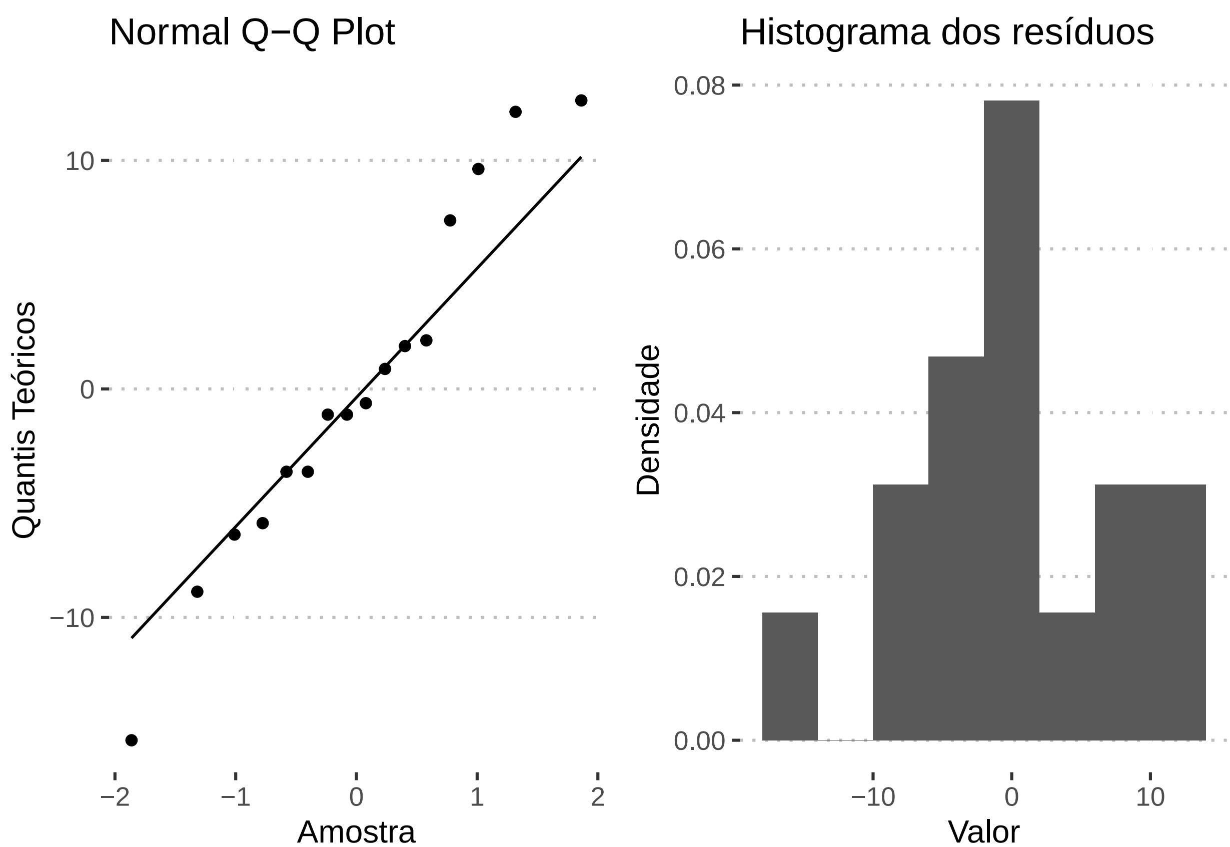
```
res <- fit$residuals

p <- ggplot(tibble(res), aes(sample = res)) +
  stat_qq() +
  stat_qq_line() +
  labs(
    x = "Amostra",
    y = "Quantis Teóricos",
    title = "Normal Q-Q Plot"
  ) +
  theme_pubclean()

q <- ggplot(tibble(res), aes(res)) +
  geom_histogram(aes(y = ..density..), binwidth = 4, stat = "bin") +
  labs(
    title = "Histograma dos resíduos",
    y = "Densidade",
    x = "Valor"
  ) +
```

```r
  theme_pubclean()

grid.arrange(p, q, ncol = 2)
```



O Q-Q Plot nos mostra o quanto os resíduos estão distantes do esperado dado que os dados têm distribuição normal, enquanto que o histograma dos resíduos nos fornece aproximações a respeito da distribuição de ξ.

A partir dos gráficos acima, podemos notar que os resíduos não estão muito afastados dos quantis teóricos, embora sua distribuição seja ligeiramente assimétrica, conforme constatado no histograma. Podemos também utilizar o teste de Shapiro-Wilk para verificar a normalidade dos dados.

4. Testes nos resíduos

- resumo dos residuos
- análise dos testes de normalidade
- teste de homocedasticidade
- teste de multicolinearidade

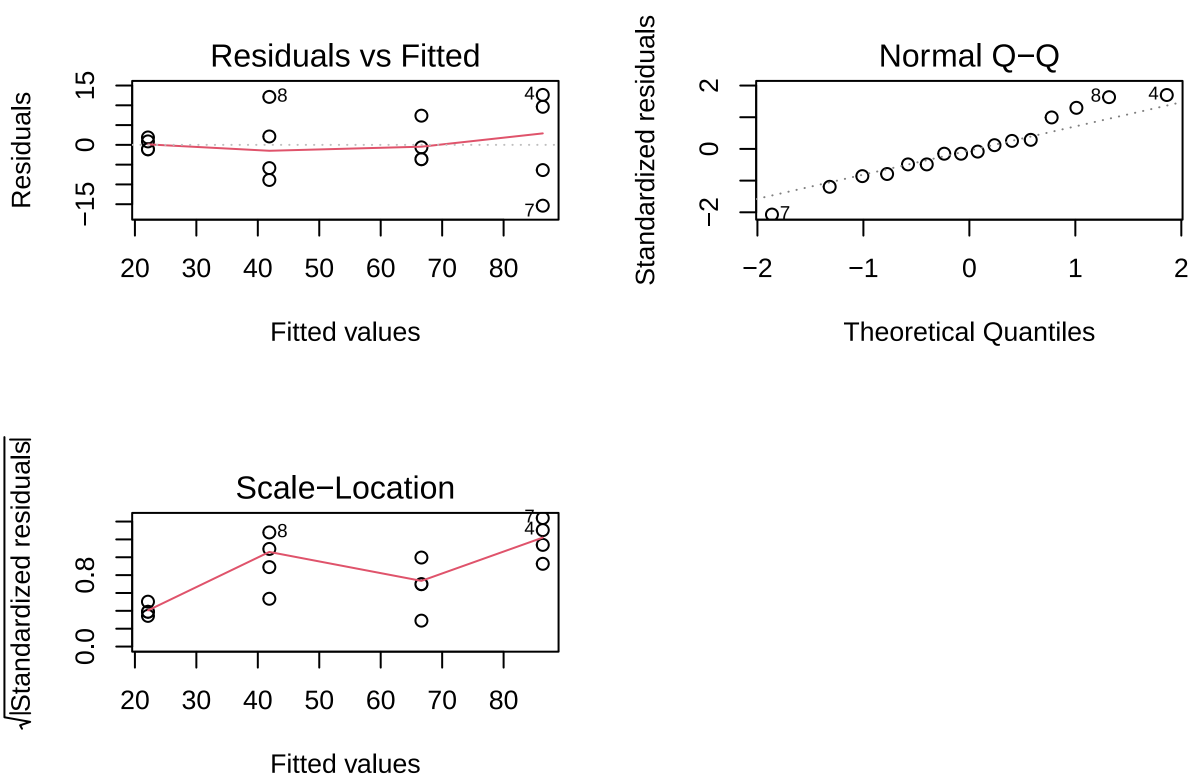
```r
summary(res)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -15.375  -4.188  -0.875   0.000   3.438  12.625
```

```r
#Análise gráfica
## Análise gráfica:
par(mfrow=c(2,2))

plot(fit)
```

```
## hat values (leverages) are all = 0.1875
##  and there are no factor predictors; no plot no. 5
```



O resumo dos resíduos nos indica que provavelmente devem existir outliers, como o valor 12.625, que se afasta muito da mediana e do 3º quantil.

Vale notar que o gráfico dos resíduos studentizados x valores ajustados indiac que não há homocesdaticidade nos dados, uma vez que a linha vermelha apresenta um padrão retangular. Iremos confirmar essa hipótese com o teste de Breusch Pagan em breve.

```r
pander(shapiro.test(res),
  style = "rmarkdown",
  caption = "Teste de normalidade Shapiro-Wilk para os resíduos"
)
```

Table 1: Teste de normalidade Shapiro-Wilk para os resíduos

Test statistic	P value
0.9658	0.7669

O teste acima confirma nossa suposição de que os resíduos têm distribuição Normal, pois, para um nível de significância de 97%, o valor-p obtido, 0,7669, não rejeita a hipótese nula, de normalidade dos dados.

Teste de Homoscedasticidade

```
#Teste de homoscedasticidade
bptest(fit)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit
## BP = 8.1436, df = 2, p-value = 0.01705
```

Dado que o p-valor para o teste de Breusch-Pagan é menor que 0.03, pode-se concluir que há heterocedasticidade nos dados, portanto rejeitamos a hipótese nula, H_0.

Teste de Multicolinearidade

Já verificamos anteriormente através dos gráficos de dispersão que não existe colinearidade entre a maioria das variáveis independentes. Podemos formalizar este resultado através da medida VIF

```
## Ausência de Multicolinearidade

vif(fit)
```

```
##    Temp Tamanho
##       1       1
```

Considerando o nosso modelo reduzido, uma vez quE $VIF = 1$ para as duas covariáveis, podemos concluir que não existe correlação alguma entre elas.

ANOVA Teste

Além disso, para determinar matematicamente se existe uma relação linear entre a variável resposta $\boldsymbol{Y}$ e qualquer as outras covariáveis $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k$, é possível utilizar o teste **ANOVA**. Nele, queremos testar:

H_0: Nenhuma das variáveis contribui significativamente ao modelo, versus:

H_a: Pelo menos uma das covariáveis contribui significativamente ao modelo.

```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Temp       1 1560.2  1560.2  23.003 0.0003492 ***
## Tamanho    1 7921.0  7921.0 116.782 7.263e-08 ***
## Residuals 13  881.7    67.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5. Resíduos Escalonados

```
(betas <- as.vector(fit$coefficients))
```

```
## [1]  80.1346055   0.2821429 -16.0649819
```

$$Y = 80.134 + 0.282x_2 - 16.065x_5$$

Interpretação dos coeficientes:

- β_0: Quando todos os x_i são iguais a zero, o valor esperado de y é de 80,134.
- β_2: Em média, para cada aumento de 1 ponto na Temperatura, esperamos um aumento de 0,282 em y, com todo o resto mantido constante.
- β_5: Em média, a cada aumento de 1 ponto no Tamanho, é esperado um descréscimo de 16,065 unidades em y, com todo o resto mantido constante.
- estimativas do modelo (betas)
- residuos e QMres
- residuos padronizados
- residuos studentizados internamente
- residuos studentizados externamente
- observações que possam ser remotas no espaço
- histograma delas e analisar

Os resíduos escalonados são úteis para encontrarmos outliers ou valores extremos.

```
# Resíduos

y_est <- as.vector(fit$fitted.values)
res <- fit$residuals

p <- 3 # número de parâmetros estimados

QMRes <- sum(res^2) / (n-p)
QMRes
```

```
## [1] 67.82692
```

Resíduo Padronizado

O Resíduo padronizado ajuda na detecção de uma observação ser potencial outlier.

```
res.padr <- res / sqrt( QMRes)
res.padr
```

```
##           1           2           3           4           5           6 
## -0.44015633 -0.13660024 -0.71335681  1.53295826  0.22766707 -0.07588902 
##           7           8           9          10          11          12 
## -1.86686996  1.47224704  0.10624463  0.89549047 -0.77406803 -1.07762412 
##          13          14          15          16 
## -0.44015633 -0.13660024  0.25802268  1.16869095
```

Aqui é preciso ter cuidado pois podemos superestimar a variância do resíduo.

Resíduo Studentizado Internamente

"Refinamento" do resíduo padronizado onde escalamos o resíduo pelo desvio-padrão 'exato' do i-ésimo resíduo e levamos em consideração onde o ponto da variável está no espaço.

```r
#  +++ pto => +hii => 1-hii -- => +++ res.int.st
res.int.st <- res / sqrt( QMRes * (1 - h))
res.int.st
```

```
##           1           2           3           4           5           6 
## -0.48830961 -0.15154436 -0.79139833  1.70066449  0.25257393 -0.08419131 
##           7           8           9          10          11          12 
## -2.07110626  1.63331144  0.11786784  0.99345747 -0.85875138 -1.19551662 
##          13          14          15          16 
## -0.48830961 -0.15154436  0.28625046  1.29654620
```

```r
# CURIOSIDADE
#obs
p/n
```

```
## [1] 0.1875
```

Resíduo Studentizado Externamente

Primeiro calculamos o **QMRes** do resíduo sem a i-ésima observação, com $i = 1, \ldots, n$ (cálculo das n variâncias sem a i-ésima observação, com $i = 1, \ldots, n$).

```r
S_i <- ( (n - p) * QMRes - res^2 / (1 - h)  ) / (n - p - 1)
S_i
```

```
##        1        2        3        4        5        6        7        8 
## 72.13141 73.34936 69.93910 57.13141 73.11859 73.43910 49.23397 58.40064 
##        9       10       11       12       13       14       15       16 
## 73.40064 67.90064 69.31090 65.40064 72.13141 73.34936 73.01603 63.97756
```

Se não tivermos nem uma observação influente, esperamos que res.int.st esteja próximo de res.ext.st. Se tivermos a i-ésima observação influente então esperamos que o i-ésimo res.ext.st seja maior em comparação com o i-ésimo res.int.st.

```r
res.ext.st <- res / sqrt( S_i * (1 - h))
res.ext.st
```

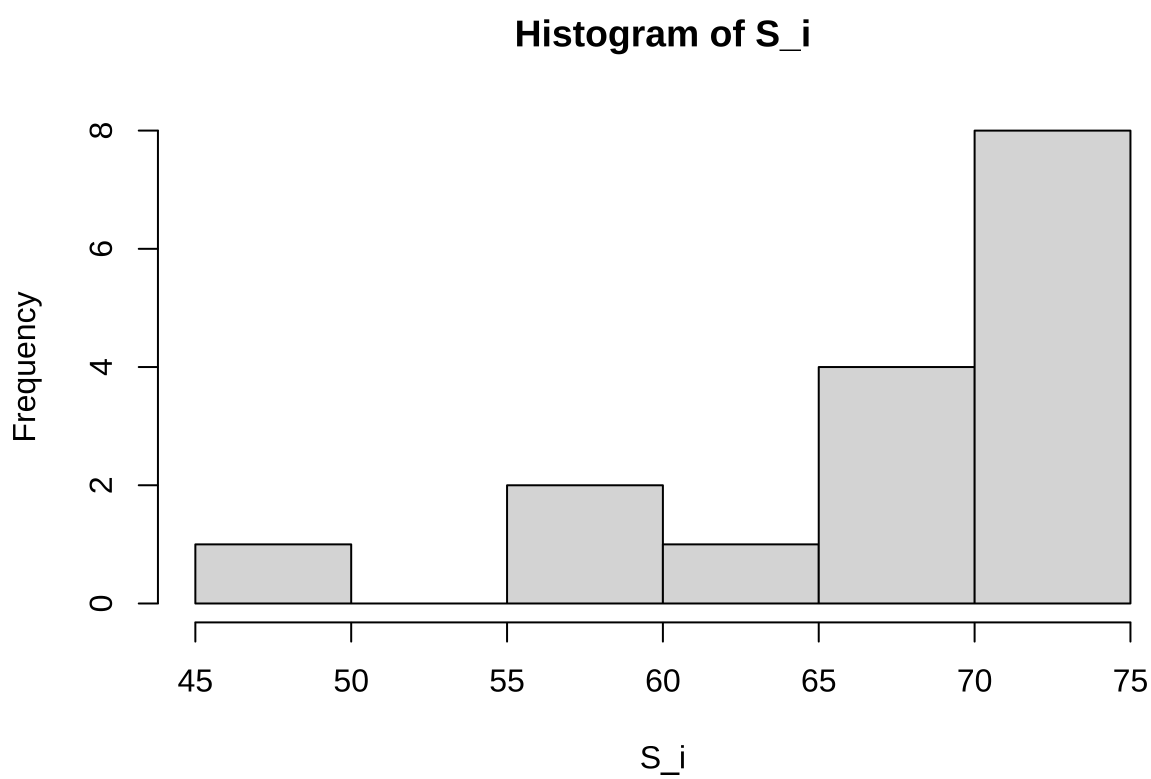
```
##           1           2           3           4           5           6
## -0.47351541 -0.14572789 -0.77935649  1.85302906  0.24326279 -0.08091046
##           7           8           9          10          11          12
## -2.43092182  1.76019691  0.11330431  0.99291804 -0.84950853 -1.21749076
##          13          14          15          16
## -0.47351541 -0.14572789  0.27589139  1.33498136
```

Vamos observar se há observações que podem ser remotas no espaço,

```r
sort(S_i)
```

```
##        7        4        8       16       12       10       11        3
## 49.23397 57.13141 58.40064 63.97756 65.40064 67.90064 69.31090 69.93910
##        1       13       15        5        2       14        9        6
## 72.13141 72.13141 73.01603 73.11859 73.34936 73.34936 73.40064 73.43910
```

```r
hist(S_i)
```



6. Comparações resíduos escalonados

```r
nome_colunas <- c("i", "e_i", "d_i", "r_i", "h_ij", "t_i")
tab <- tibble(i= 1:16,res, res.ext.st, res.int.st,h,res.padr)
kable(tab, col.names = nome_colunas, format = "markdown")
```

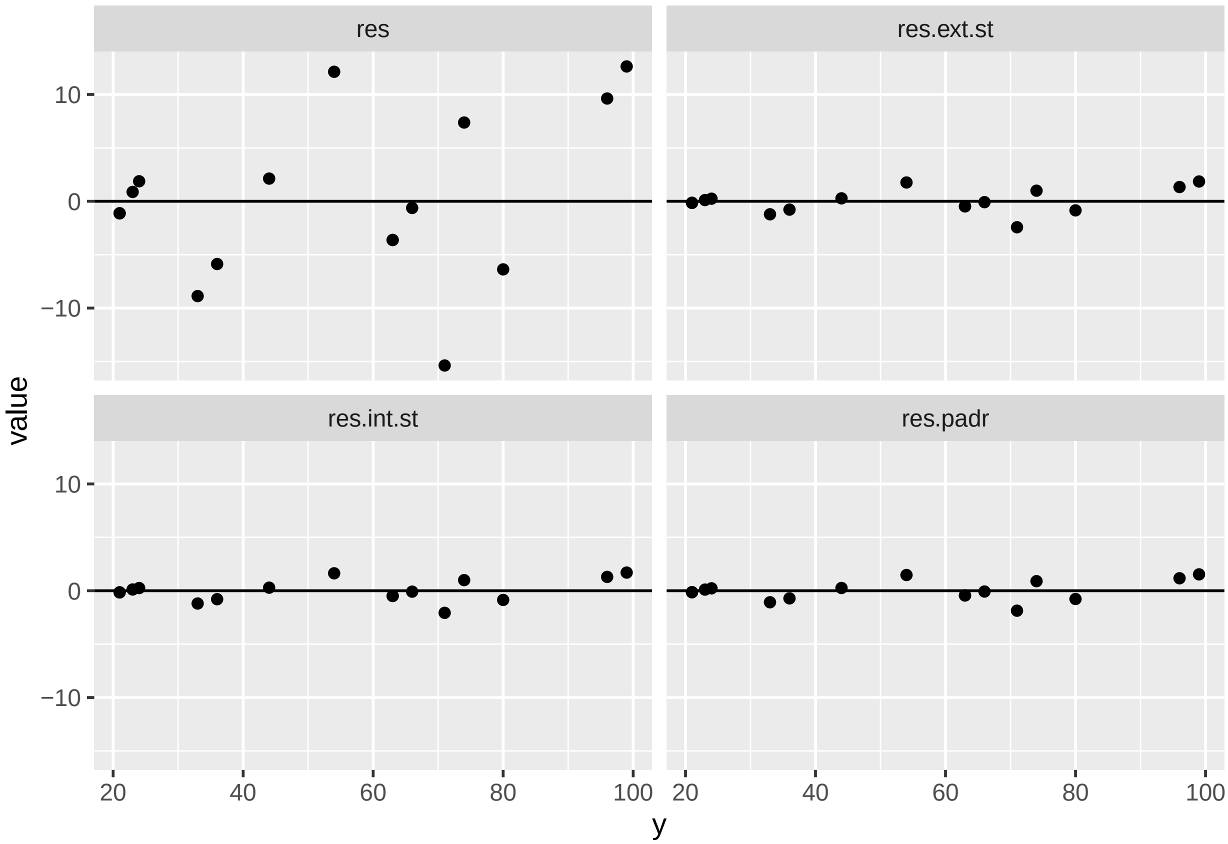
i	e_i	d_i	r_i	h_ij	t_i
1	-3.625	-0.4735154	-0.4883096	0.1875	-0.4401563
2	-1.125	-0.1457279	-0.1515444	0.1875	-0.1366002
3	-5.875	-0.7793565	-0.7913983	0.1875	-0.7133568
4	12.625	1.8530291	1.7006645	0.1875	1.5329583
5	1.875	0.2432628	0.2525739	0.1875	0.2276671
6	-0.625	-0.0809105	-0.0841913	0.1875	-0.0758890
7	-15.375	-2.4309218	-2.0711063	0.1875	-1.8668700
8	12.125	1.7601969	1.6333114	0.1875	1.4722470
9	0.875	0.1133043	0.1178678	0.1875	0.1062446
10	7.375	0.9929180	0.9934575	0.1875	0.8954905
11	-6.375	-0.8495085	-0.8587514	0.1875	-0.7740680
12	-8.875	-1.2174908	-1.1955166	0.1875	-1.0776241
13	-3.625	-0.4735154	-0.4883096	0.1875	-0.4401563
14	-1.125	-0.1457279	-0.1515444	0.1875	-0.1366002
15	2.125	0.2758914	0.2862505	0.1875	0.2580227
16	9.625	1.3349814	1.2965462	0.1875	1.1686909

- quadro comparativo com os resultados obtidos no item anterior
- análise de cada um dos resíduos calculados (aula 17)

7. Gráfico de Resíduos versus ajuste

- análise do gráfico para cada um dos residuos calculados no item 5 vs valores ajustados

```
tab %>% dplyr::select(!c(i,h)) %>%
mutate(y = dados$y) %>%
pivot_longer(!y) %>%
ggplot() +
geom_point(aes(x = y, y = value)) +
geom_hline(yintercept = 0) +
facet_wrap(~name)
```



8. Transformações

- proponha uma transformação para seu modelo que corrija possíveis problemas e compare
- refazer os itens de 1 a 7 com o novo modelo

9. Teste de Falta de ajuste

- proponha um caso ou exemplo onde seja necessario a aplicação do teste da falta de ajuste
- residuos vs valores ajustados
- mostre a falta de ajuste dos dados
- ajuste o modelo
- ANOVA
- SQEP
- SQFA
- análise do teste F_0

Criação de dataset artificial

Resolvemos utilizar um dataset que contém dados do comprimento da mandíbula de veados com relação à idade do animal.

```
dados2 <- read_table("jaws.txt")
```

```
##
```

```
## -- Column specification --------------------------------------------------------
## cols(
##   age = col_double(),
##   bone = col_double()
## )
```

```
#Ajustando um modelo linear
fit2 <- lm(dados2$bone ~ dados2$age, data = dados2)
summary(fit2)
```

```
##
## Call:
## lm(formula = dados2$bone ~ dados2$age, data = dados2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.259 -10.457   2.353  18.048  33.589
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   53.259      5.837   9.124 2.23e-12 ***
## dados2$age     1.642      0.201   8.166 6.97e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.3 on 52 degrees of freedom
## Multiple R-squared:  0.5619, Adjusted R-squared:  0.5534
## F-statistic: 66.68 on 1 and 52 DF,  p-value: 6.968e-11
```

Gráfico dos resíduos vs Valores Ajustados

10. Mínimos Quadrados Ponderados

- proponha um caso ou exemplo onde seja necessário a aplicação da técnica dos mínimos quadrados e faça a respectiva análise

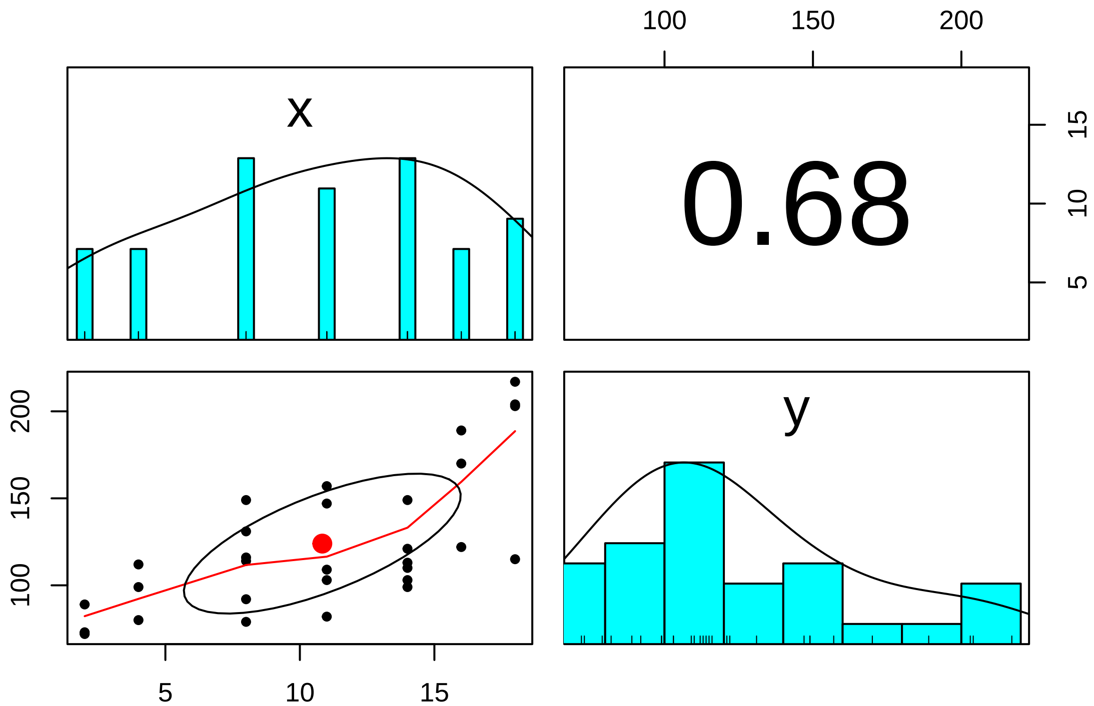
A Continuação será apresentado um exemplo simulado de falta de ajuste.

Exemplo: Um pesquisador no setor de vendas queria estudar a associação entre o faturamento mensal médio de vendas de lanches (Y) e a despesa por mês com divulção(X). Os dados referentes a 30 lanchonetes encontram-se abaixo:

```
x<-c(2,2,2,4,4.0,4,8,8,8,8,8,8,11,11,11,11,11,14,14,14,14,14,14,16,16,16,18,18,18,18)
y<-c(89,73,72,80,112,99,79,114,116,92,131,149,109,157,103,147,82,113,149,121,99,103,110,170,189,122,203
gasto_venda= data.frame(cbind(x,y))
```

Note que que estes dados indicam falta de ajuste a um modelo linear

```
pairs.panels(gasto_venda)
```

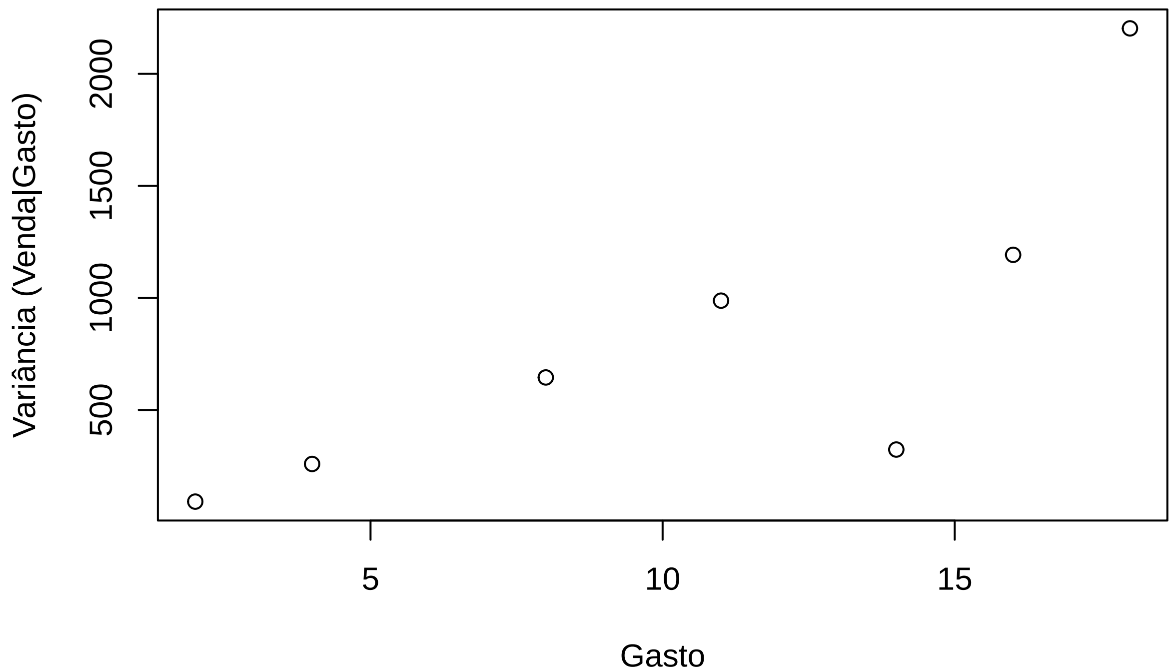


Portanto, devemos realizar o ajuste do modelo utilizando o Método de Mínimos Quadrados Ponderados. Para isso deve-se observar as estimativas do Erro Puro para cada nível de X, ou seja, os valores de Var(Y | X). Observe a função abaixo:

```
tapply(gasto_venda[,2],as.factor(gasto_venda[,1]),var)
```

```
##         2         4         8        11        14        16        18
##   91.0000  259.0000  645.1000  987.8000  323.3667 1192.3333 2202.9167
```

```
gasto <-c(2,4,8,11,14,16,18)
v<-c(91.0000,  259.0000,  645.1000,  987.8000,  323.3667, 1192.3333, 2202.9167)
plot (gasto,v,xlab="Gasto",ylab="Variância (Venda|Gasto)" )
```



Observa-se que Var(Venda | Gasto) é proporcional ao Gasto. Sendo assim, o peso W_i deve ser inversamente proporcional ao X_i.

```
wi <- c(1/2 ,1/4 ,1/8 ,1/11 ,1/14 ,1/16 ,1/18)
valores_peso= data.frame(cbind(gasto,wi))
valores_peso %>%
  knitr::kable()
```

gasto	wi
2	0.5000000
4	0.2500000
8	0.1250000
11	0.0909091
14	0.0714286
16	0.0625000
18	0.0555556

Abaixo encontra-se nosso comando R para o ajuste do modelo via Método de Mínimos Quadrados Ponderados e a respectiva saída do software com os coeficientes ajustados.

Cálculo do ajuste_ponderado

```r
ajuste_ponderado=lm(formula = y ~ x, weights = 1/x)
summary(ajuste_ponderado)
```

```
##
## Call:
## lm(formula = y ~ x, weights = 1/x)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -12.904  -6.834  -1.650   7.279  13.838
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.0359     6.9928  10.015 9.30e-11 ***
## x             4.9782     0.8037   6.194 1.09e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.632 on 28 degrees of freedom
## Multiple R-squared:  0.5781, Adjusted R-squared:  0.5631
## F-statistic: 38.37 on 1 and 28 DF,  p-value: 1.087e-06
```

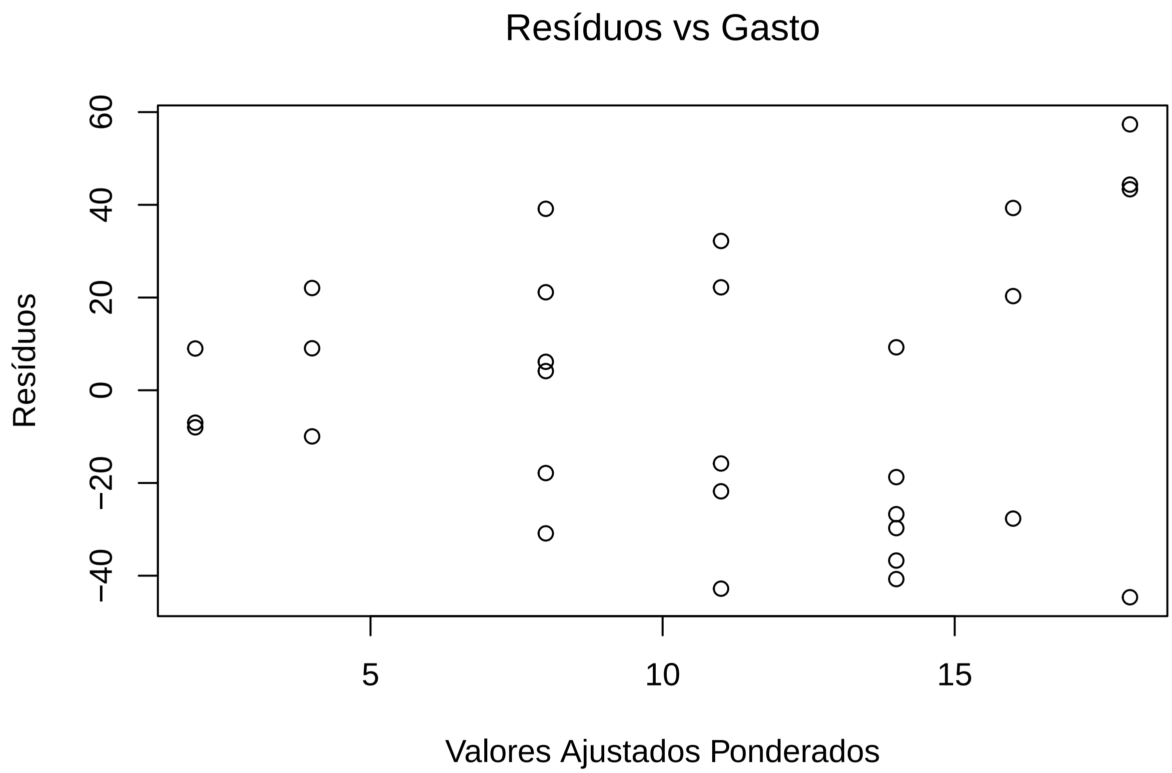
Cálculo da Anova

```r
anova(ajuste_ponderado)
```

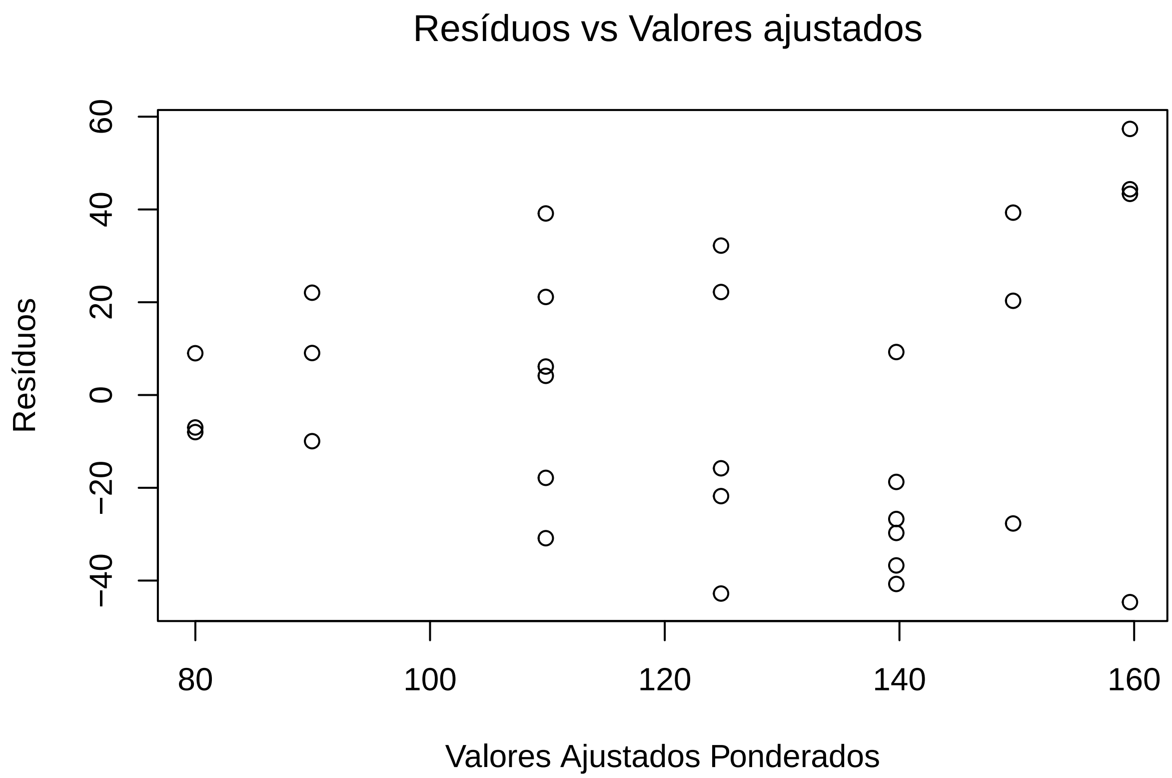
```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1 2858.7  2858.7   38.37 1.087e-06 ***
## Residuals 28 2086.1    74.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gráficos para Análise dos Resíduos

```r
plot(x,ajuste_ponderado$residuals,
     main = expression(paste("Resíduos vs Gasto")),
     xlab="Valores Ajustados Ponderados",ylab="Resíduos")
```




```
plot(ajuste_ponderado$fitted.values,ajuste_ponderado$residuals,
     main = expression(paste(" Resíduos vs Valores ajustados")),
     xlab="Valores Ajustados Ponderados",ylab="Resíduos")
```



As Figuras acima evidenciam que o problema da heterocedasticidade dos erros foi solucionado, pois nos dois gráficos os resíduos ponderados estão dispostos homogeneamente em torno de zero.

Note também, que os coeficientes (Betas) estimados seriam:

```
b0_est=ajuste_ponderado$coefficients[1]
b1_est=ajuste_ponderado$coefficients[2]
cbind(b0_est,b1_est)
```

```
##               b0_est   b1_est
## (Intercept) 70.03587 4.978227
```

Gráfico da reta ajustada aos dados

```
plot(x,y,
 main = expression(paste("Reta ajustada com ",
 hat(beta)[0],"=70.03587",
 " e ", hat(beta)[1],"=4.978227")),
 xlab = "x", ylab = "y")
curve(b0_est + b1_est*x, add = T, col = 'red')
```

Reta ajustada com $\hat{\beta}_0$=70.03587 e $\hat{\beta}_1$=4.978227