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 já trabalhado anteriormente, e também de encontrar conjuntos de dados que permitam aplicar técnicas de modelagem em regressão linear.

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)].

Iremos trabalhar com uma significância de 95%.

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

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

1. Análise Descritiva dos dados

Temos cinco covariáveis quantitativas em nosso dataset:

- Y : Rendimento total de aceite por lote de amendoim*.
- X_1 : Pressão
- X_2 : Temperatura
- X_3 : Fluxo de CO2
- X_4 : Umidade
- X_5 : Tamanho

e a coluna y corresponde a nossa variável preditora que determina **o rendimento total de aceite por lote de amendoim**.

Para obtermos um resumo dos dados, podemos utilizar as funções *glimpse* e *summary* que nos retornarão, respectivamente, os tipos de variáveis e o máximo de observações possíveis no espaço proposto na horizontal; e, medidas descritivas das nossas variáveis, sendo estas os valores mínimo e máximo das observações, o primeiro quantil, a mediana, a média e o terceiro quantil.

```
glimpse(dados)
```

```
## Rows: 16
## Columns: 6
## $ Pressao  <dbl> 415, 550, 415, 550, 415, 550, 415, 550, 415, 550, 415, 550, 4~
## $ Temp     <dbl> 25, 25, 95, 95, 25, 25, 95, 95, 25, 25, 95, 95, 25, 25, 95, 95
## $ FluxoCO2 <dbl> 5, 5, 5, 5, 15, 15, 15, 15, 5, 5, 5, 5, 15, 15, 15, 15
## $ Umidade  <dbl> 40, 40, 40, 40, 40, 40, 40, 40, 60, 60, 60, 60, 60, 60, 60, 60
## $ Tamanho  <dbl> 1.28, 4.05, 4.05, 1.28, 4.05, 1.28, 1.28, 4.05, 4.05, 1.28, 1~
## $ y        <dbl> 63, 21, 36, 99, 24, 66, 71, 54, 23, 74, 80, 33, 63, 21, 44, 96
```

Utilizaremos apenas as variáveis Temperatura e Tamanho para ajustar ao modelo neste trabalho, pois foi o modelo reduzido obtido na atividade prática anterior.

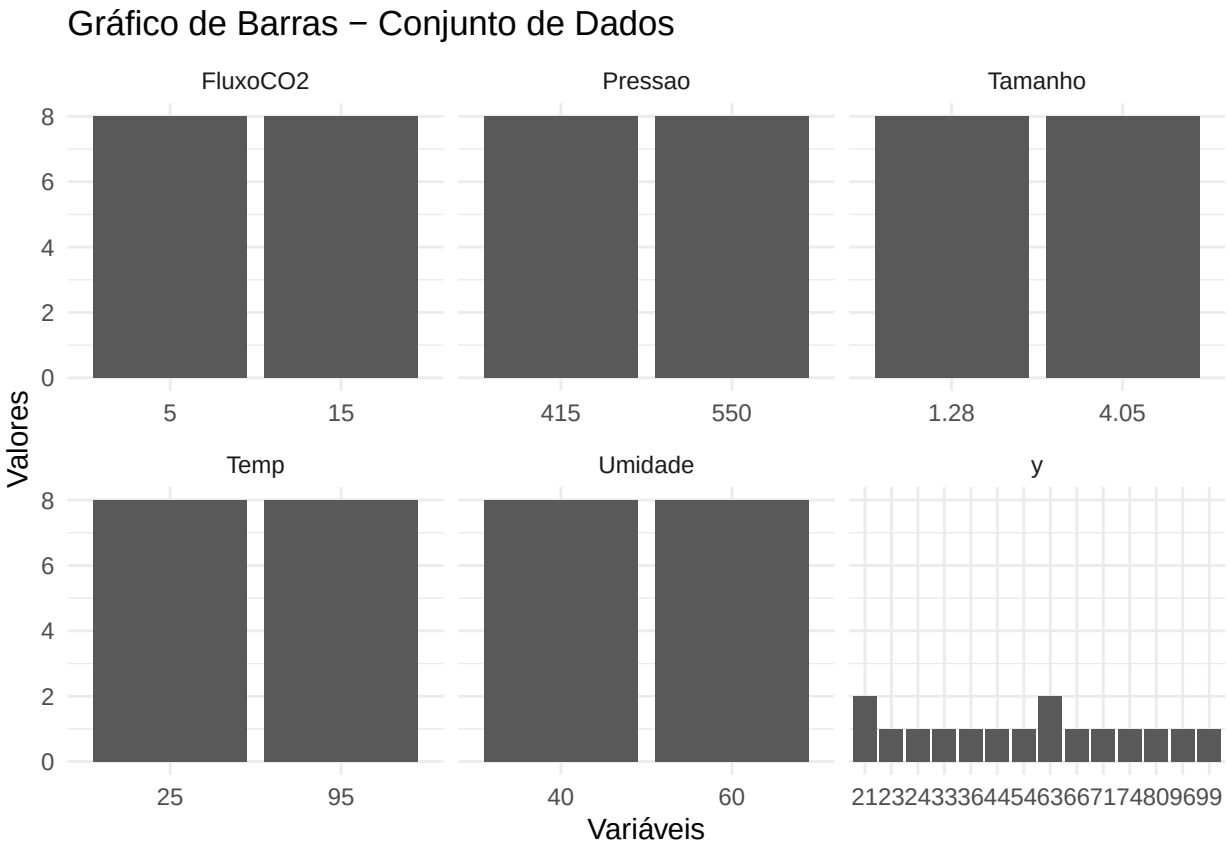
```
pander(summary(dados[,c(2,5,6)]), "Sumário das variáveis de interesse")
```

Table 1: Sumário das variáveis de interesse

Temp	Tamanho	y
Min. :25	Min. :1.280	Min. :21.00
1st Qu.:25	1st Qu.:1.280	1st Qu.:30.75
Median :60	Median :2.665	Median :58.50
Mean :60	Mean :2.665	Mean :54.25
3rd Qu.:95	3rd Qu.:4.050	3rd Qu.:71.75
Max. :95	Max. :4.050	Max. :99.00

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()
```



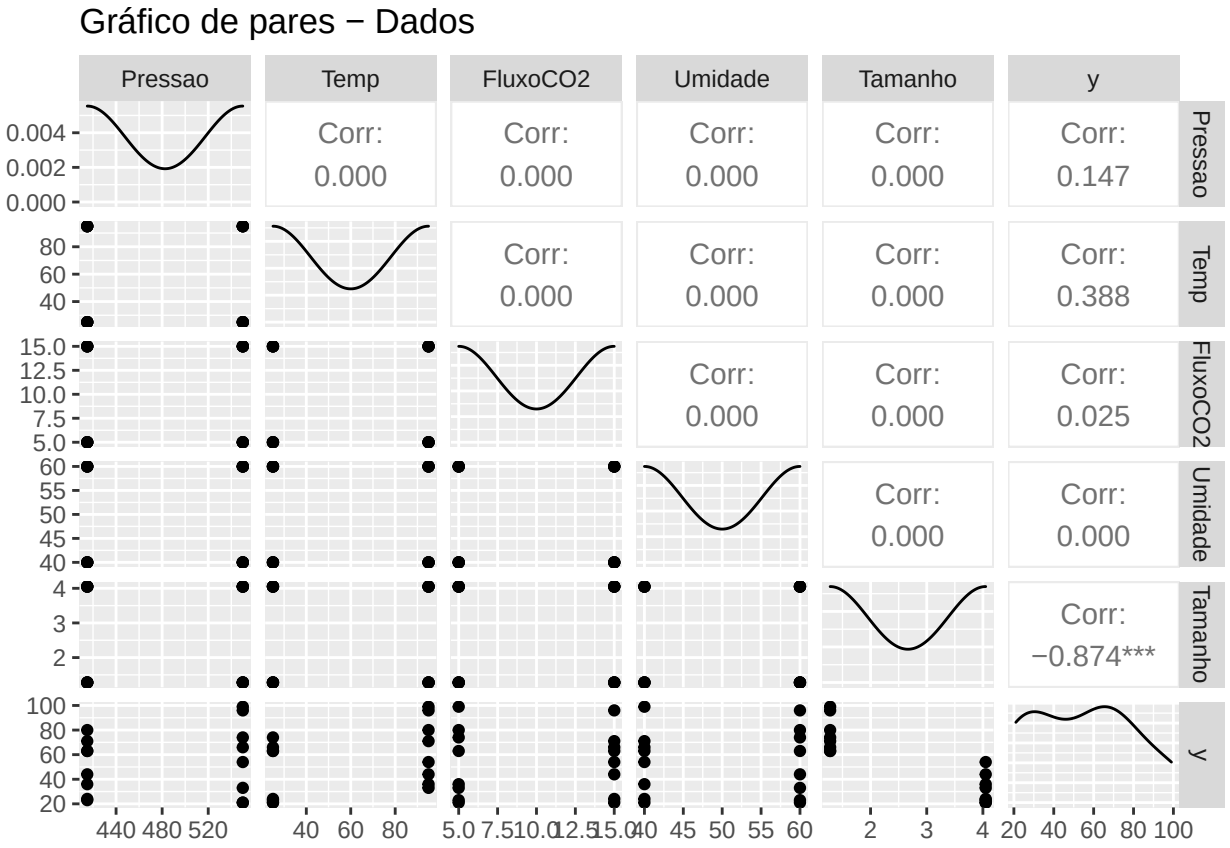
A partir dos gráficos de barras, podemos ver que nossas cinco covariáveis, apesar de serem quantitativas, assumem apenas dois valores, com a mesma proporção. A única variável que assume mais valores do que isso é y, que apararenta ter uma distribuição quase uniforme.

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.

Correlações

Podemos visualizando a correlação de Pearson entre as covariáveis e y por meio de um gráfico de pares: fazendo uso destas correlações dispostas, podemos construir uma matriz de gráficos para expor as relações entre as variáveis. Com isso, podemos visualizar as densidades de frequência na diagonal, gráficos de dispersão no painel triangular inferior e coeficientes das correlações no superior.

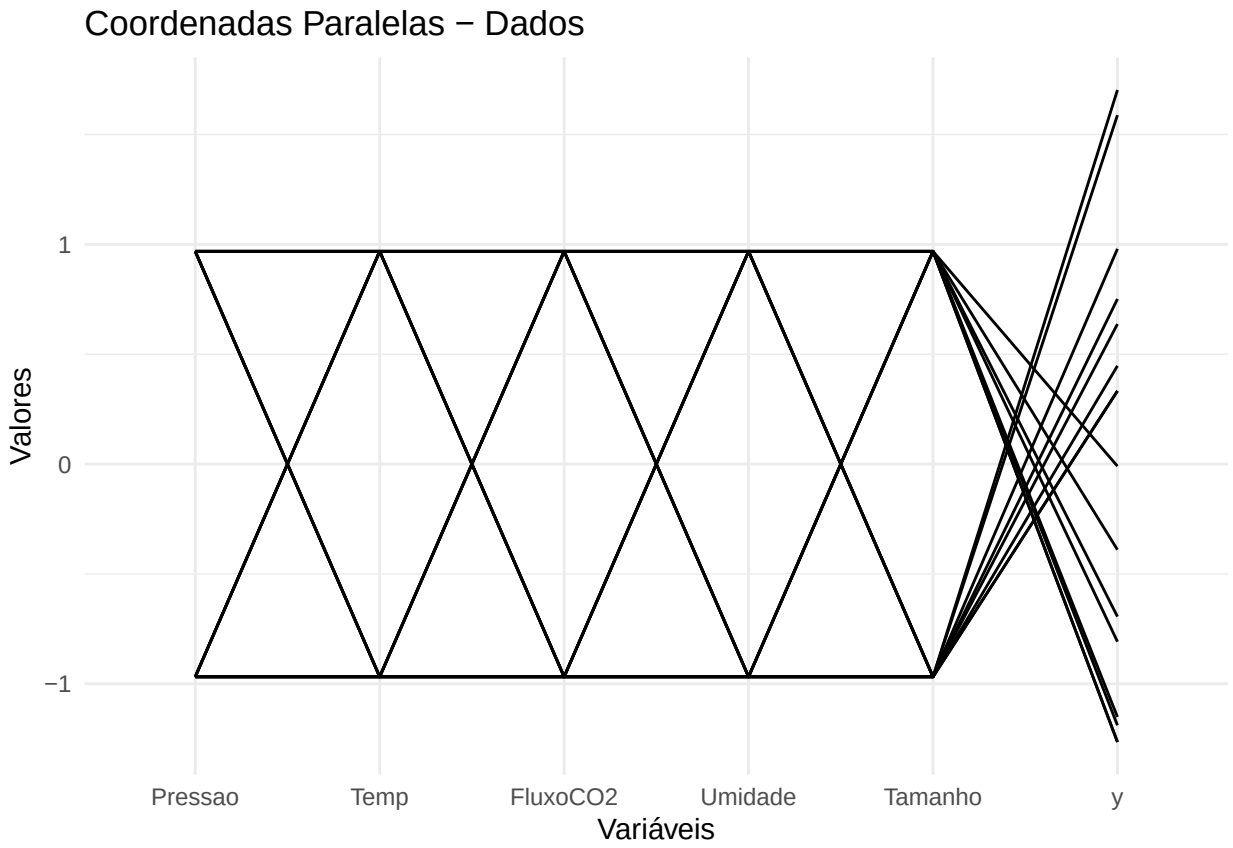
```
ggpairs(dados) + ggtitle("Gráfico de pares - Dados")
```



Observando o comportamento da correlação das nossas variávies e analisando os resultados dispostos, 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) - que apresenta uma correlação alta -, e x2 - que, apesar de apresentar uma correlação relativamente baixa, se torna significante devido ao cenário obtido.

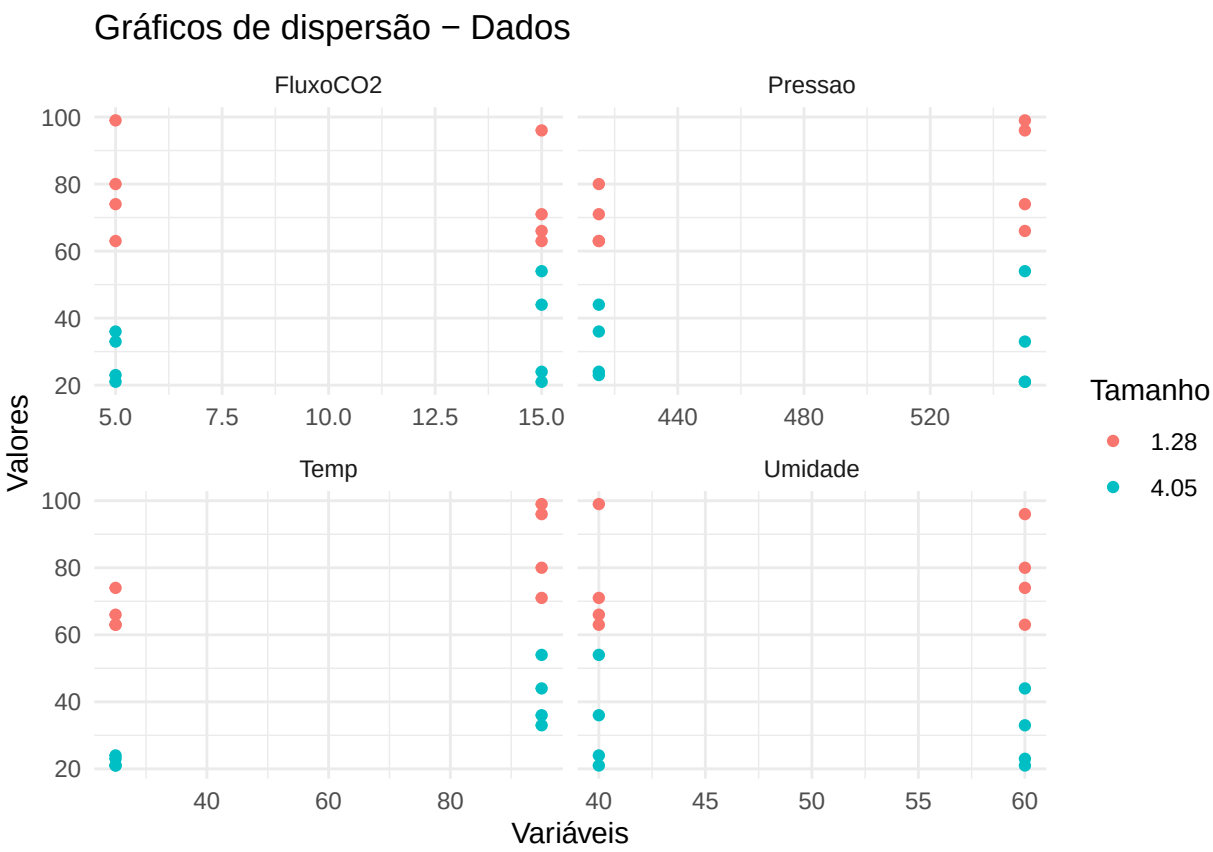
Essa ausência de correlação pode ser explicada pelo comportamento cruzado da maior parte das covariáveis, que também pode ser notado através do gráfico de coordenadas paralelas, repare o padrão em "X":

```
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 pontos vistos acima:

```
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 modelo reduzido que trabalharemos terá que conter apenas algumas delas.

2. Matriz Hat

Para criar a nossa matriz X, em que a primeira coluna corresponde a uma repetição de números 1, a segunda na covariável x1 e, a terceira, x2:

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

Definindo nossa matriz Y, que contém apenas uma coluna e é a da variável preditora y - o vetor resposta:

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

A partir disso, podemos construir nossa matriz HAT, em que:

$Hat: H = X(X^TX)^{-1}X^T$, onde h_{ii}: i-esimo elemento da diagonal de H; h_{ij}: elemento ij da matriz H.

```
H <- X %*% solve(t(X) %*% X) %*% t(X)
h <- diag(H)
pander(summary(h), "Sumário da diagonal da matriz hat obtida")
```

Table 2: Sumário da diagonal da matriz hat obtida

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

```
pander(cbind("$x_2$"= dados$Temp, "$x_5$"= dados$Tamanho, h), "Influência de cada um dos pontos")
```

Table 3: Influência de cada um dos pontos

x_2	x_5	h
25	1.28	0.1875
25	4.05	0.1875
95	4.05	0.1875
95	1.28	0.1875
25	4.05	0.1875
25	1.28	0.1875
95	1.28	0.1875
95	4.05	0.1875
25	4.05	0.1875
25	1.28	0.1875
95	1.28	0.1875
95	4.05	0.1875
25	1.28	0.1875
25	4.05	0.1875
95	4.05	0.1875
95	1.28	0.1875

Temos que:

$h_{ii} = \frac{1}{n} + \frac{(X_i - \overline{X})^2}{S_{XX}}$

Levando isso em consideração, h_{ii} deve atingir seu menor valor no ponto $\overline{X}$, se igualando a $\frac{1}{n}$.

Temos então que h_{ii} é uma medida de alavanca, que nos informa o quão distante do centro a observação está. Ou seja, quanto mais a observação se distancia de $\overline{X}$, mais h_{ii} cresce.

Com isso, é possível determinarmos possível outliers - uma vez que, se h_{ii} for relativamente maior do que os das demais observações, temos que ele provavelmente será um ponto influente e distante dos demais (e da média).

Analisando os resultados obtidos na tabela acima, e aplicando esses pontos, podemos observar que todos nossos pontos possuem a mesma influência, uma vez que h_{ii} se mantém constante para todas as observações. Logo, provavelmente também não contamos com outliers dentre as nossas observações.

Calculando $\frac{1}{n}$ e o comparando com nosso h_{ii}:

```
1/n
```

```
## [1] 0.0625
```

Como contamos com $h_{ii} = 0.1875$ para todos os valores da tabela, temos que os pontos se encontram com um afastamente uniforme dentre eles da média.

3. Análise de resíduos

Para verificar se os pressupostos básicos que precisamos assumir para ajustar o modelo de regressão linear estão sendo satisfeitos, podemos analisar os resíduos e procurar por padrões que não tenham sido modulados.

Ajuste do modelo

Primeiramente, vamos construir nosso modelo e utilizar a função *summary* para observarmos as principais medidas descritivas de nosso modelo de regressão linear.

```
fit <- lm(y ~ Temp + Tamanho, data = dados)
pander(fit, "Ajuste do modelo linear reduzido")
```

Table 4: Ajuste do modelo linear reduzido

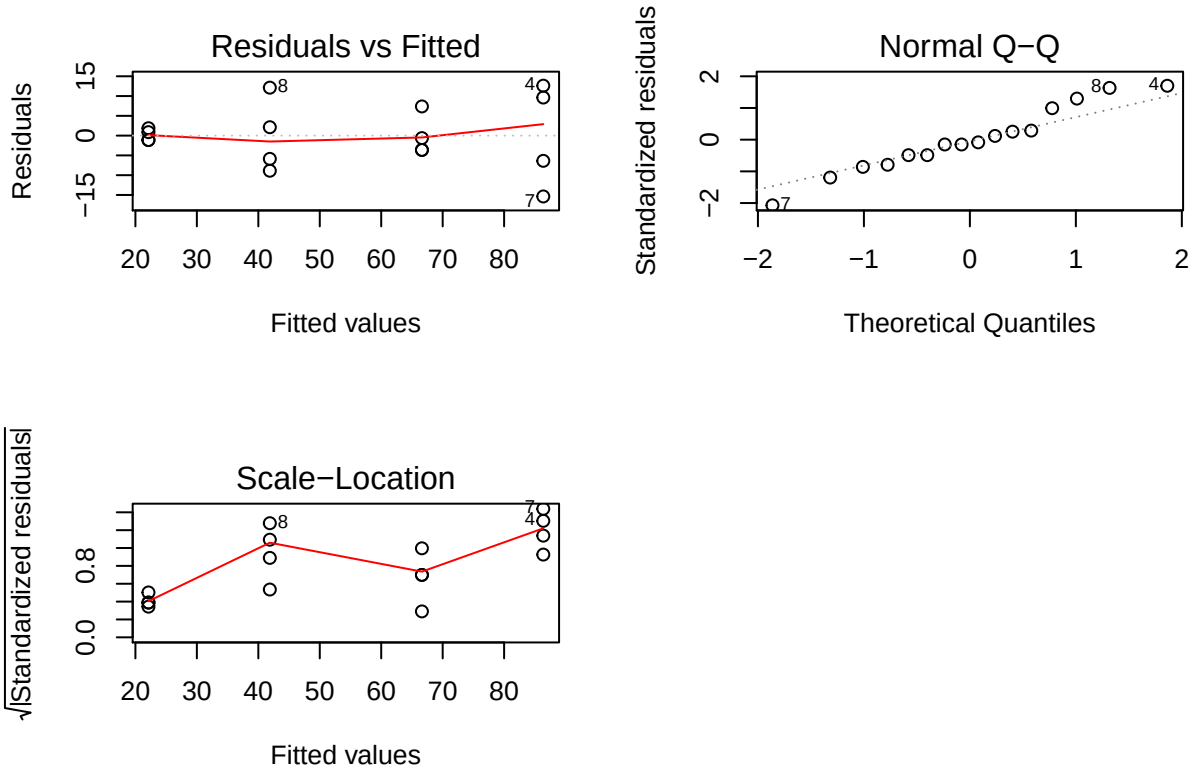
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	80.13	5.691	14.08	3.007e-09
Temp	0.2821	0.05883	4.796	0.0003492
Tamanho	-16.06	1.487	-10.81	7.263e-08

Note que o teste-t realizado automaticamente pelo programa já rejeita a hipótese nula de que as variáveis não sejam dignificativas ao modelo para um nível de significância de 5%.

Diagnóstico do modelo

```
par(mfrow=c(2,2))
plot(fit)
```

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



1) Para a primeira plotagem, obtemos o gráfico dos resíduos comparados com os valores ajustados, onde é possível avaliar o pressuposto de linearidade. Nesse gráfico, podemos notar que a linha vermelha está muito próxima de estar completamente no eixo horizontal, uma vez que o balanceamento de valores é muito equilibrado. Ou seja, não temos nenhuma observação que influencia nosso ajuste muito fortemente - seja positivamente ou negativamente.

2) Já no segundo gráfico temos um QQ plot. Nele, podemos verificar se os resíduos apresentaram distribuição normal. Podemos ver que existe uma tendência a esta distribuição - principalmente nos pontosmais centrais. Entretanto, há um leve afastamento nas extremidades, e um falta de preenchimento de espaço no centro do gráfico, o que pode nos indicar que essa tendência não é tão forte.

3) No terceiro gráfico era esperado termos os resíduos estandardizados vs valores ajustados, que serviria para verificar a homocedasticidade. Entretanto, por conta dos valores da nossa matriz HAT serem uniformes, não é possível realizar tal plotagem.

4) Na última plotagem, podemos verificar caso existam dutliers e possíveis pontos influentes, reiterando o pressuposto no item 2 do trabalho. Como podemos ver, há uma influência alternada entre positiva e negativa, porém em aproximadamente mesmos graus de intensidade e distanciamento. Também podemos verificar a ausência de outliers, uma vez que, se houvesse, deveria haver uma linha pontilhada vermelha com pontos para fora desta.

4. Testes nos resíduos

Além de verificar nossas suposições graficamentes, é necessários embasá-las com o auxílio de testes estatísticos. Uma primeira suposição passível de ser verificada é a de normalidade dos resíduos. Primeiramente, olhemos para as medidas-resumo obtidas:

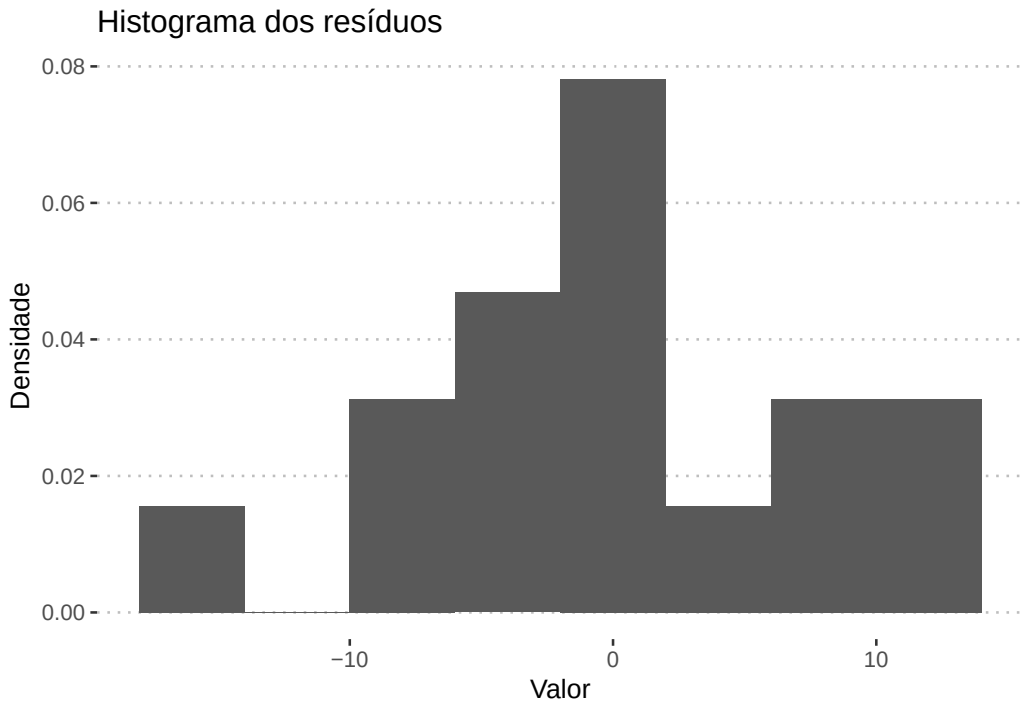
```r
res <- fit$residuals
pander(summary(res), "Sumário dos resíduos")
```

Table 5: Sumário dos resíduos

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-15.38	-4.188	-0.875	-1.665e-16	3.437	12.62

O resumo dos resíduos nos indica que provavelmente devem existir outliers, como o valor -15.38, que se afasta muito da mediana e do 3^o quantil. Podemos avaliar a suposição graficamente com o auxílio de um histograma:

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



Embora haja uma leve assimetria no gráfico, tal comportamento também pode ser em decorrência ao número reduzido de observações, mas, é difícil mensurar essa suposição sem a utilização de testes apropriados.

Teste de normalidade

O teste de Shapiro-wilk tem como hipótese nula que os dados têm distribuição normal, versus a ausência de normalidade. A tabela abaixo mostra os valores do teste obtidos para os resíduos:

```
pander(shapiro.test(res), "Teste de Shapiro-Wilk para os resíduos")
```

Table 6: Teste de Shapiro-Wilk para os resíduos

Test statistic	P value
0.9658	0.7669

Dessa forma, confirmamos nossa suposição de que os resíduos têm distribuição Normal, pois, para um nível de confiança de 95%, o valor-p obtido, 0,7669, não rejeita a hipótese nula, de normalidade dos dados.

Teste de Homoscedasticidade

Este teste que só funciona quando a distribuição é normal. Nele, testamos:

- H_0: há homocedasticidade na amostra, versus:
- H_1: não há homocedasticidade.

A tabela abaixo ilustra os valores obtidos com o teste.

```
pander(bptest(fit), "Teste de Breush-Pagan para o modelo ajustado")
```

Table 7: Teste de Breush-Pagan para o modelo ajustado

Test statistic	df	P value
8.144	2	0.01705 *

Dado que o p-valor para o teste de Breusch-Pagan é menor que 0.05, rejeitamos a hipótese nula, e pode-se concluir que há heterocedasticidade nos dados a um nível de significância de 5%.

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

```
pander(vif(fit),caption= "VIF das covariáveis do modelo reduzido")
```

Temp	Tamanho
1	1

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

Teste ANOVA

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.

```
pander(anova(fit), "Tabela ANOVA do modelo ajustado")
```

Table 9: Tabela ANOVA do modelo ajustado

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Temp	1	1560	1560	23	0.0003492
Tamanho	1	7921	7921	116.8	7.263e-08
Residuals	13	881.7	67.83	NA	NA

Neste caso, os dois p-valores obtidos rejeitam H_0 a um nível de significância de 5%. Dessa forma, ambas as covariáveis contribuem significativamente ao modelo.

5. Resíduos Escalonados

Definimos os resíduos como sendo

$$e_i = y_i - \hat{y}_i, \quad i = 1, ..., n.$$

Iniciamos essa sessão apresentando os coeficientes ajustados de nosso modelo.

```
pander(fit$coefficients, "Coeficientes do modelo ajustado")
```

(Intercept)	Temp	Tamanho
80.13	0.2821	-16.06

Dessa forma,

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

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.

É útil trabalharmos com o escalonamento dos resíduos para encontrarmos **outliers**, observações que estejam de alguma maneira separadas do resto dos dados.

Quadrado Médio dos Resíduos

Temos que $QM_{res} = \frac{1}{(n-p)} \sum_{i=1}^{n} e_i^2$, que será nosso critério para saber se a retirada de uma possível observação influente melhora ou piora nosso modelo.

```
QMRes <- anova(fit)$`Mean Sq`[3]
pander(cat("QMres: ", QMRes))
```

QMres: 67.82692

Resíduo Padronizado

Sendo a variância média dos resíduos estimada por QM_{res}, para torna-lá igual a 1 basta fazermos:

$$d_i = \frac{e_i}{\sqrt{QM_{res}}}, \quad i = 1, ..., n.$$

Consequentemente, valores grandes (como, digamos, $d_i > 2$) potencialmente indicam um **outlier**. Note que QM_{res} é apenas uma aproximação para a variância do i-ésimo resíduo, o que pode ocasionar em distorções em sua estimação.

```
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
```

Não observamos nenhum ponto influente em comparação com os demais com essa metodologia.

Resíduo Studentizado Internamente

Podemos refinar o método anterior escalonando o resíduo pelo desvio-padrão 'exato' do i-ésimo resíduo e levando em consideração onde o ponto da variável está no espaço.

Utilizando a matriz hat, podemos estimar a variância do i-ésimo resíduo como sendo:

$$Var(e_i) = \sigma^2(1 - h_{ii})$$

Onde h_{ii} é o i-ésimo elemento da diagonal da matriz Hat. Ainda mais, como essa é uma medida de **locação** do i-ésimo ponto com respeito a x, a variância de e_i depende de onde o ponto x_i está. Dessa forma:

$$r_i = \frac{e_i}{\sqrt{QM_{res}(1 - h_{ii})}}$$

```
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
```

A observação 7 obteve um valor alto aqui, fornecendo indícios que possa ser um ponto de interesse para nossa análise.

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 $i = 1, \ldots, n$).

```
p <- 3
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. Assim,

$$t_i = \frac{e_i}{\sqrt{S^2_{(i)}(1 - h_{ii})}}$$

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

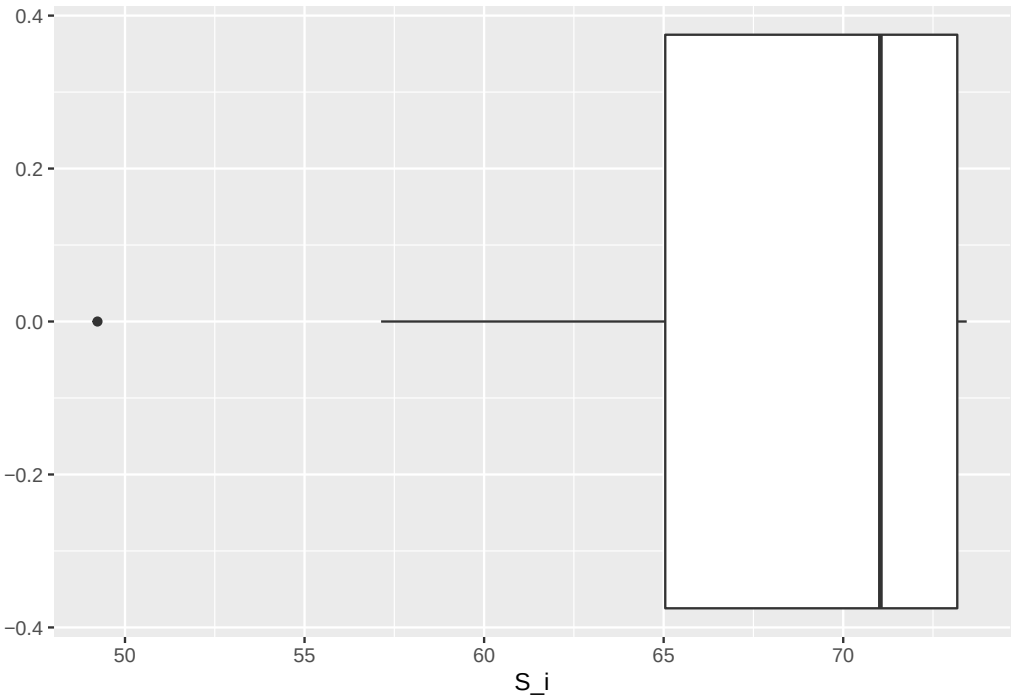
```
##           1           2           3           4           5           6
## -0.47351541 -0.14572789 -0.77935649  1.85302906  0.24326279 -0.08091046
##           7           8           9          10          11          12
## -2.43092182  1.76019691  0.11330431  0.99291804 -0.84950853 -1.21749076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

Novamente, observação 7 destoou das demais.

Observações remotas no espaço

Esses pontos podem potencialmente controlar algumas das propriedades do modelo de regressão.

```
ggplot(tibble(S_i), aes(x = S_i)) + geom_boxplot()
```



A observação 7 foi a mais afastada, apresentando resíduo Studentizado igual a 49.23397. Vamos retirá-la do modelo e calcular o QM_{res}.

```
fit7 <- lm(y ~ Temp + Tamanho, subset = -7, data = dados)
pander(cat("QMres: ", anova(fit7)$`Mean Sq`[3]))
```

QMres: 49.23397

Como o quadrado médio dos resíduos diminuiu, concluímos que a retirada da observação 7 contribuiu para um melhor ajuste do modelo de regressão linear múltipla.

6. Comparações resíduos escalonados

```
nome_colunas <- c('$i$', '$e_i (1)$', '$d_i (2)$', '$r_i (3)$', '$h_{ii}$', '$t_i (4)$')
tab <- tibble(i= 1:16, res, res.padr, res.int.st, h, res.ext.st)
pander(tab, col.names = nome_colunas, "Resíduos escalonados obtidos")
```

Table 11: Resíduos escalonados obtidos

i	$e_i(1)$	$d_i(2)$	$r_i(3)$	h_{ii}	$t_i(4)$
1	-3.625	-0.4402	-0.4883	0.1875	-0.4735
2	-1.125	-0.1366	-0.1515	0.1875	-0.1457
3	-5.875	-0.7134	-0.7914	0.1875	-0.7794
4	12.62	1.533	1.701	0.1875	1.853
5	1.875	0.2277	0.2526	0.1875	0.2433
6	-0.625	-0.07589	-0.08419	0.1875	-0.08091
7	-15.38	-1.867	-2.071	0.1875	-2.431
8	12.12	1.472	1.633	0.1875	1.76

i	$e_i(1)$	$d_i(2)$	$r_i(3)$	h_{ii}	$t_i(4)$
9	0.875	0.1062	0.1179	0.1875	0.1133
10	7.375	0.8955	0.9935	0.1875	0.9929
11	-6.375	-0.7741	-0.8588	0.1875	-0.8495
12	-8.875	-1.078	-1.196	0.1875	-1.217
13	-3.625	-0.4402	-0.4883	0.1875	-0.4735
14	-1.125	-0.1366	-0.1515	0.1875	-0.1457
15	2.125	0.258	0.2863	0.1875	0.2759
16	9.625	1.169	1.297	0.1875	1.335

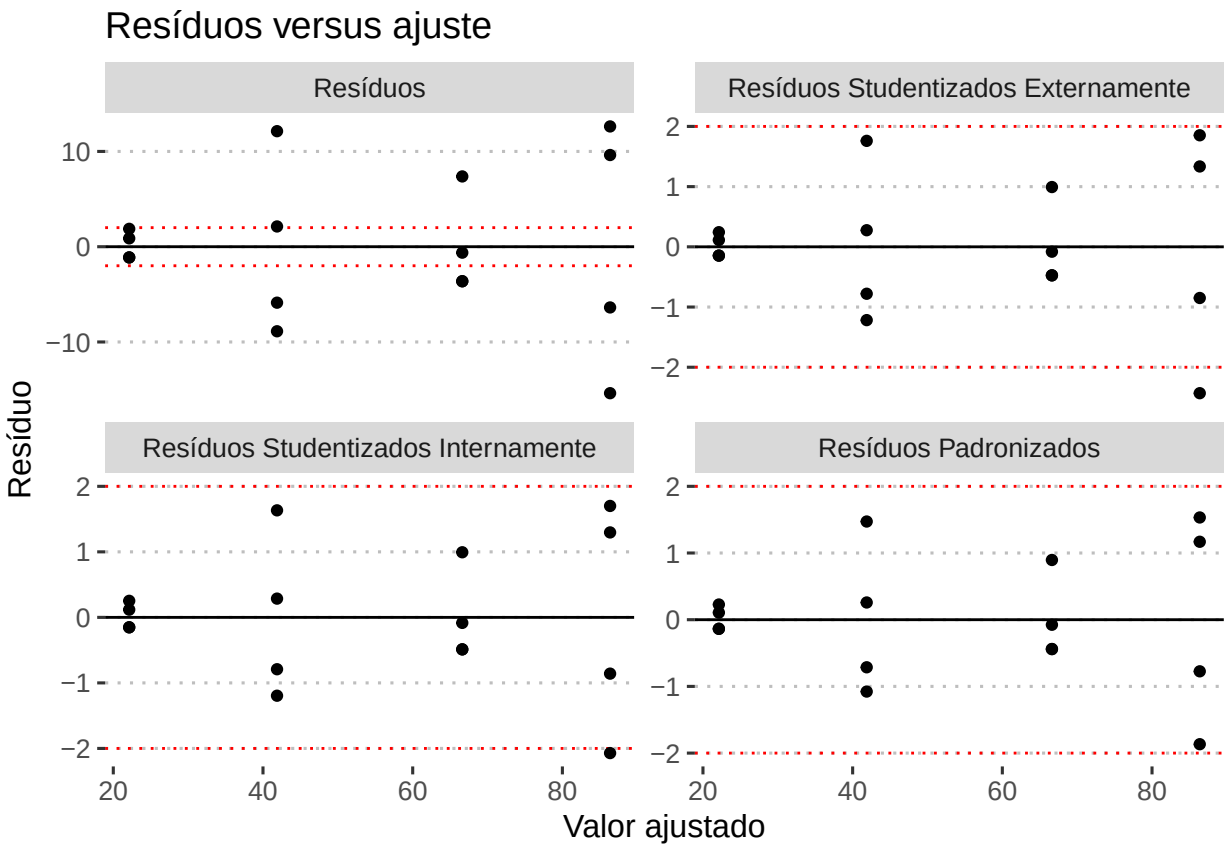
- (1) analisando os resíduos, vemos que $e_7 = -15.38$ é grande;
- (2) Resíduo padronizado: não temos nenhum $d_i > 2$;
- (3) Resíduo Studentizado: aqui r_7 é grande, indicando um ponto remoto influente;
- (4) Resíduo Studentizado Externamente: $t_i - t_7 > r_7$ e portanto o QM_{res} sem ele é menor do que com todas as observações.

7. Gráfico de Resíduos versus ajuste

Nessa sessão, avaliaremos os gráficos de cada um dos resíduos escalonados versus ajuste, obtidos no item anterior. Esse tipo de análise pode contribuir em exibir padrões não modulados pelo ajuste, sendo desejável a impressão de uma banda horizontal contendo os resíduos.

```
res_types <- as_labeller(c(
                          'res'= "Resíduos",
                          'res.ext.st' = "Resíduos Studentizados Externamente",
                          'res.int.st' = "Resíduos Studentizados Internamente",
                          'res.padr' = "Resíduos Padronizados"))
```

```
  fit_vs_val <- tab %>% bind_cols(fitted = fit$fitted.values) %>%
  dplyr::select(!c(i,h)) %>%
  pivot_longer(!fitted)

  fit_vs_val %>%
  ggplot() +
  geom_hline(yintercept = 0) +
  geom_hline(yintercept = -2, lty = 3, color = "red") +
  geom_hline(yintercept = 2, lty = 3, color = "red") +
  facet_wrap(~name, scales = "free_y", labeller = res_types) +
  geom_point(aes(x = fitted, y = value)) +
  labs(title = "Resíduos versus ajuste",
       x = "Valor ajustado",
       y = "Resíduo") +
  theme_pubclean()
```



Não foi observado nenhum padrão nos plots residuais, o que nos fornece indícios gráficos de uma certa dispersão aleatória dos resíduos, com os escalonamentos aparentando ter contribuído para diminuição da variância, como seria esperado. Apenas a observação 7 ficou fora das bandas de controle estipuladas no caso dos resíduos studentizados.

8. Transformações

Conforme observado através do resultado obtido na questão anterior, constatamos que nosso ajuste obteve algum sucesso dado que vários dos probelmas iniciais foram solucionados. No entanto, apesar da melhora ser visível, ainda é possível melhorar alguns pontos, tais como a presença de heterocedasticidade nos dados; e a constância da variância. Então, com a finalidade de comparar, como não há uma relação bem definida entre as covariáveis, optaremos por uma transformação na variável resposta.

Fazendo a tranformação na variável resposta

```
mod_logy <- lm(log(y) ~ Temp + Tamanho , data = dados)
pander(mod_logy)
```

Table 12: Fitting linear model: log(y) ~ Temp + Tamanho

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.374	0.1176	37.18	1.375e-14
Temp	0.006205	0.001216	5.103	0.0002026

	Estimate	Std. Error	t value	Pr(>|t|)
Tamanho	-0.3307	0.03073	-10.76	7.618e-08

Significância da Regressão

```
pander(anova(mod_logy))
```

Table 13: Analysis of Variance Table

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Temp	1	0.7546	0.7546	26.04	0.0002026
Tamanho	1	3.357	3.357	115.8	7.618e-08
Residuals	13	0.3767	0.02898	NA	NA

```
n <- length(dados$y)
Xmod_logy <- NULL
Xmod_logy <- matrix(c(rep(1,n),dados$Temp,dados$Tamanho), nrow=n, ncol=3 )
Hmod_logy <- Xmod_logy %*% solve(t(Xmod_logy) %*% Xmod_logy) %*% t(Xmod_logy)
hmod_logy <- diag(Hmod_logy)
pander(summary(hmod_logy), 'Sumário de $h_{ii}$')
```

Table 14: Sumário de h_{ii}

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

```
n <- length(dados$y)
betasmod_logy <- as.vector(mod_logy$coefficients)
b0_estmod_logy <- betasmod_logy[1]
b1_estmod_logy <- betasmod_logy[2]
b2_estmod_logy <- betasmod_logy[3]
y_estmod_logy <- as.vector(mod_logy$fitted.values)
resmod_logy <- log(dados$y) - y_estmod_logy
betasmod_logy
```

```
## [1]  4.374020674  0.006204987 -0.330702651
```

Interpretação dos coeficientes:

- β_0: Quando todos os x_i são iguais a zero, o valor esperado de y passou de 80.134 para 4.374.
- β_2: Em média, para cada aumento de 1 ponto na Temperatura, esperávamos um aumento de 0.282 em y, agora esperamos um de 0.006 com todo o resto mantido constante.
- β_5: Em média, a cada aumento de 1 ponto no Tamanho, era esperado um descréscimo de 16.065 unidades em y, agora esperamos um de -0.331 com todo o resto mantido constante.

```r
p <- 3 # Número de Parâmetros Estimados
SQResmod_logy <- sum((resmod_logy)^2)
QMResmod_logy <- SQResmod_logy / (n-p)
SQResmod_logy
```

```
## [1] 0.3766866
```

```r
QMResmod_logy
```

```
## [1] 0.02897589
```

```r
pander(bptest(mod_logy))
```

Table 15: studentized Breusch-Pagan test: `mod_logy`

Test statistic	df	P value
2.114	2	0.3474

Vale ressaltar que mesmo este novo modelo tendo apresentado um aumento referente ao modelo inicial, o p-valor para o teste de Breusch-Pagan continua sendo menor que 0.05, novamente rejeitamos a hipótese nula, e pode-se concluir que continua havendo heterocedasticidade nos dados a um nível de significância de 5%

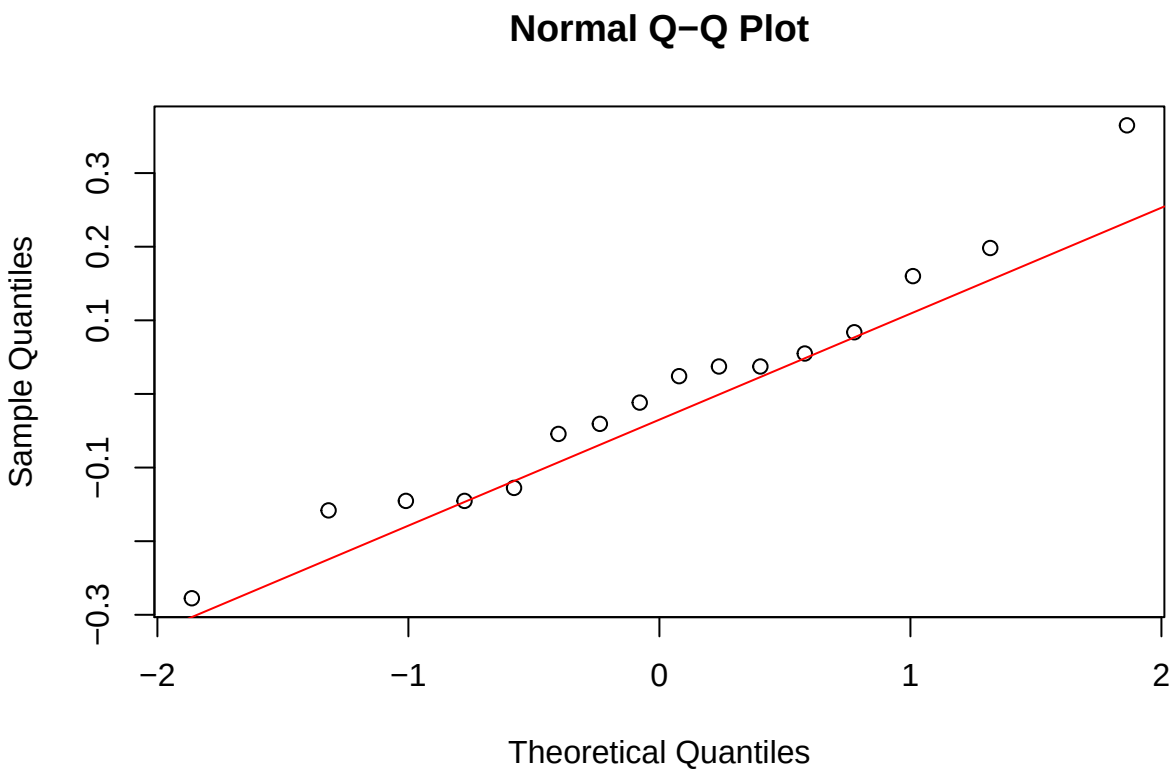
```r
pander(shapiro.test(mod_logy$residuals))
```

Table 16: Shapiro-Wilk normality test: `mod_logy$residuals`

Test statistic	P value
0.9696	0.8328

Neste novo modelo, houve um aumento significativo no p-valor, reafirmando que os dados realmente possuem distribuição normal.

```r
par(mfrow = c(1, 1))
qqnorm(mod_logy$residuals)
qqline(mod_logy$residuals, col= 'red')
```




```
res_mod_logy <- mod_logy$residuals
```

Resíduos

```
res.padr_mod_logy <- resmod_logy / sqrt( QMResmod_logy)
```

Resíduo Padronizado

```
res.int.st_mod_logy<-rstandard(mod_logy)
```

Resíduo Internamente Studentizado

```
res.ext.st_mod_logy <- rstudent(mod_logy)
```

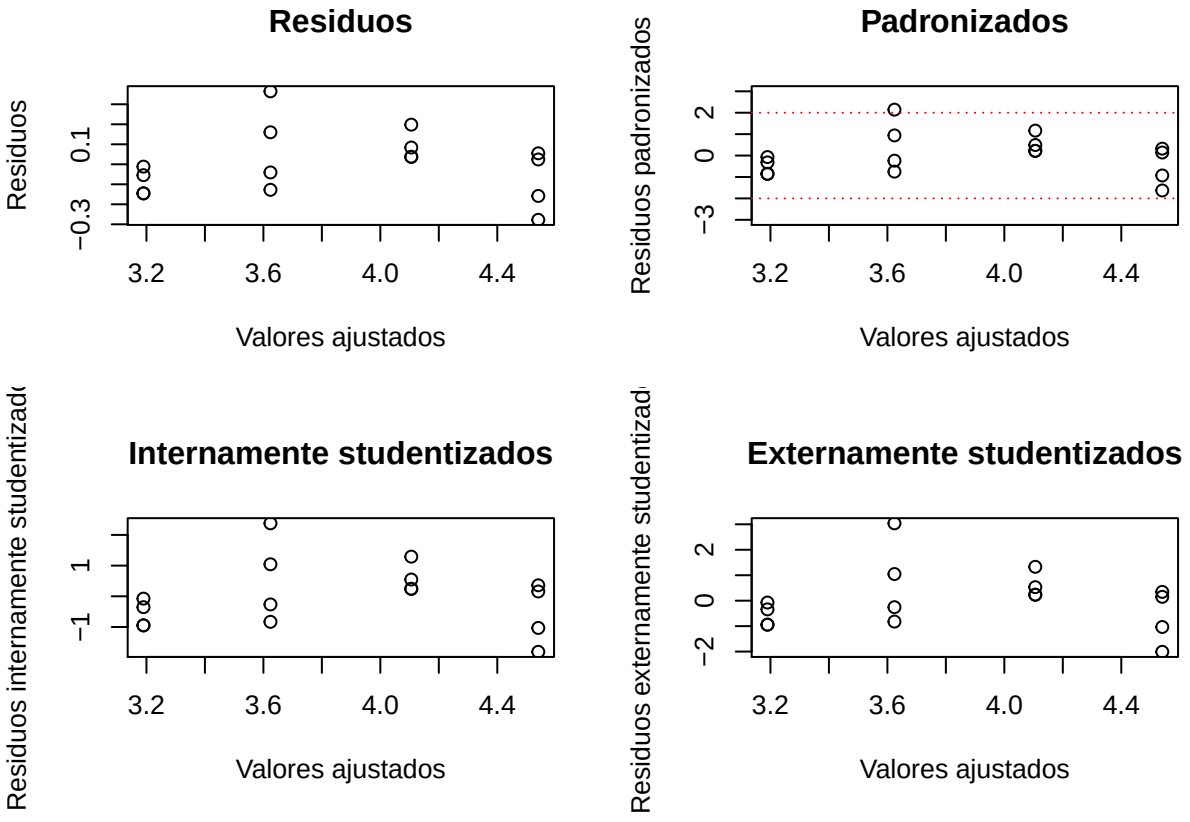
Resíduo Externamente Studentizado

```r
par(mfrow = c( 2, 2))
y_estmod_logy <- mod_logy$fitted.values
plot(res_mod_logy ~ y_estmod_logy,
     ylab = "Residuos", xlab = "Valores ajustados", main = "Residuos")
plot(res.padr_mod_logy ~ y_estmod_logy,
    ylab = "Residuos padronizados", xlab = "Valores ajustados", main = "Padronizados", ylim = c(-3, 3))
abline(h = c(-2, 2), col = 'red', lty = 3)
which(res.padr_mod_logy > 2)
```

Resíduos pelos Valores Ajustados

```
## [1] 8
```

```r
plot(res.int.st_mod_logy ~ y_estmod_logy, ylab = "Residuos internamente studentizados", xlab
 = "Valores ajustados", main = "Internamente studentizados")
plot(res.ext.st_mod_logy ~ y_estmod_logy, ylab = "Residuos externamente studentizados", xlab
 = "Valores ajustados", main = "Externamente studentizados")
```



Novamente não foi observado nenhum padrão nos plots residuais, o que reforça nossos indícios gráficos de uma certa dispersão aleatória dos resíduos. Porém, no ajuste realizado na questão anterior apenas a observação 7 encontrava-se fora das bandas de controle estipuladas no caso dos resíduos studentizados, já agora, através da visualização gráfica, o número de outliers parece ter aumentado.

Comparação dos Resíduos

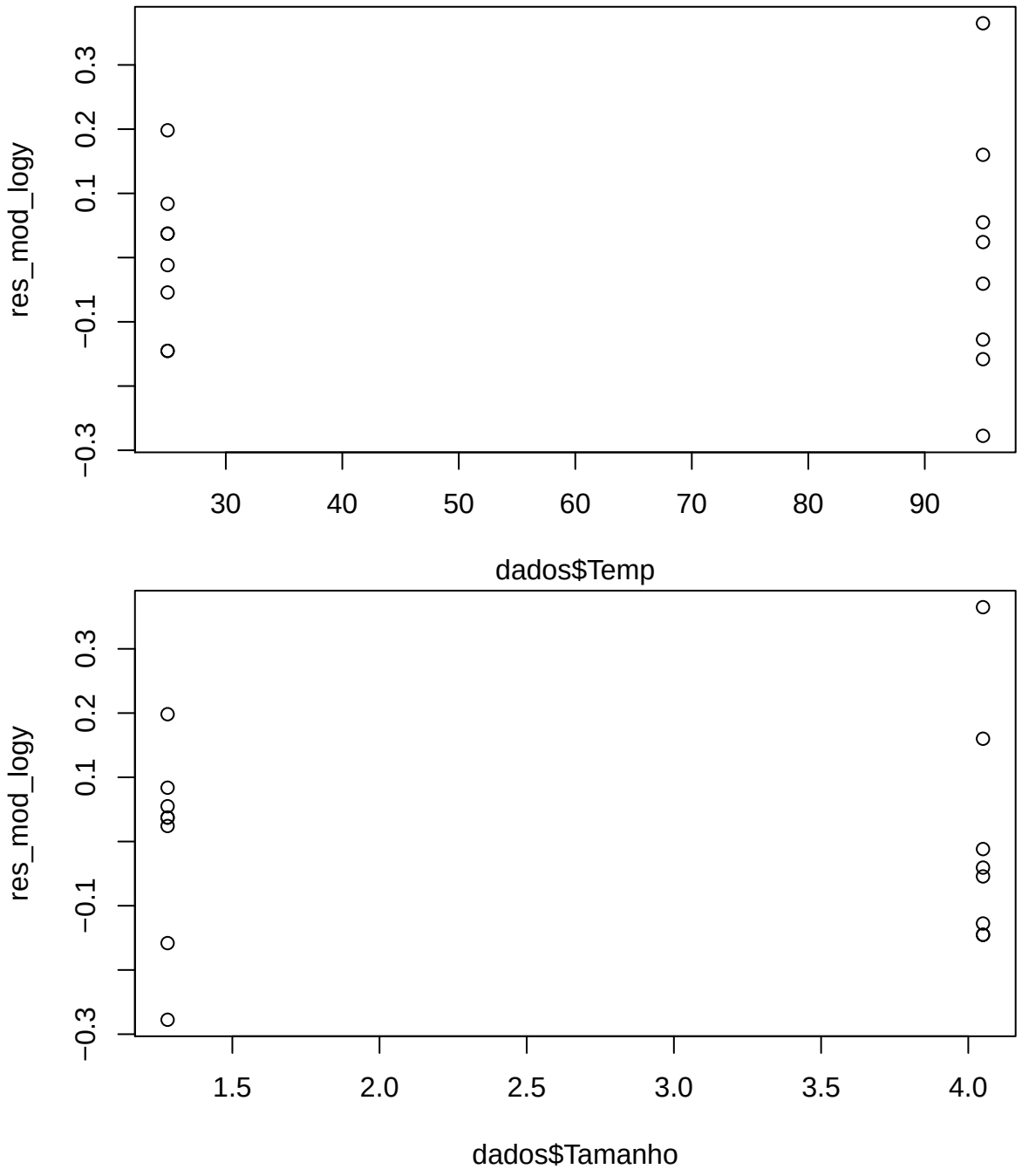
```
tab <- tibble(i= 1:16, res_mod_logy, res.padr_mod_logy, res.int.st_mod_logy, h, res.ext.st_mod_logy)

pander(tab, col.names = nome_colunas, "Resíduos escalonados obtidos")
```

Table 17: Resíduos escalonados obtidos

i	$e_i(1)$	$d_i(2)$	$r_i(3)$	h_{ii}	$t_i(4)$
1	0.03729	0.2191	0.243	0.1875	0.234
2	-0.1453	-0.8535	-0.9468	0.1875	-0.9428
3	-0.04063	-0.2387	-0.2648	0.1875	-0.2551
4	0.05492	0.3227	0.358	0.1875	0.3456
5	-0.01175	-0.069	-0.07655	0.1875	-0.07356
6	0.08381	0.4923	0.5462	0.1875	0.5309
7	-0.2775	-1.63	-1.809	0.1875	-2.009
8	0.3648	2.143	2.378	0.1875	3.039
9	-0.05431	-0.319	-0.3539	0.1875	-0.3417
10	0.1982	1.164	1.292	0.1875	1.329
11	-0.1582	-0.9292	-1.031	0.1875	-1.034
12	-0.1276	-0.7498	-0.8319	0.1875	-0.8214
13	0.03729	0.2191	0.243	0.1875	0.234
14	-0.1453	-0.8535	-0.9468	0.1875	-0.9428
15	0.16	0.9402	1.043	0.1875	1.047
16	0.02415	0.1419	0.1574	0.1875	0.1514

Os resíduos escalonados são úteis para identificarmos valores extremos, sendo os resíduos padronizados uma estimativa para a variância dos resíduos. Logo, como houve um aumento significativo destes, constatou-se um aumento no número de outliers.

```
par(mfrow=c(1,1))
plot(res_mod_logy~dados$Temp+dados$Tamanho)
```



9. Teste de Falta de ajuste

Adaptamos um dataset que contém dados do comprimento da mandíbula de veados com relação à idade do animal, arredondando os valores das variáveis para a inclusão de réplicas, a fim de simularmos a falta de ajuste do modelo, e propor uma transformação que solucione o problema.

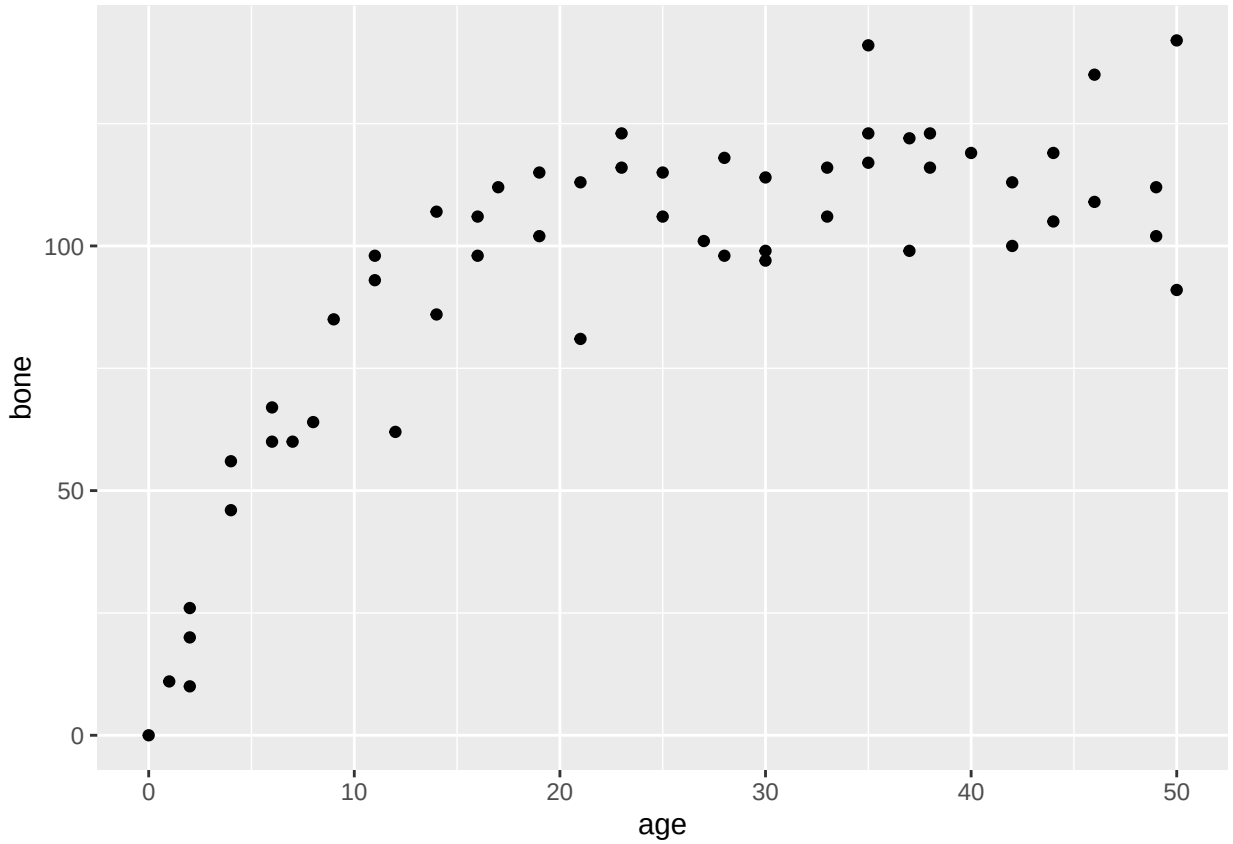
```
dados2 <- read_delim("dados/mandibulas.txt", ",")
#Ajustando um modelo linear
fit2 <- lm(bone ~ age, data = dados2)
pander(fit2)
```

Table 18: Fitting linear model: bone ~ age

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	53.06	5.741	9.243	1.467e-12
age	1.656	0.1982	8.359	3.467e-11

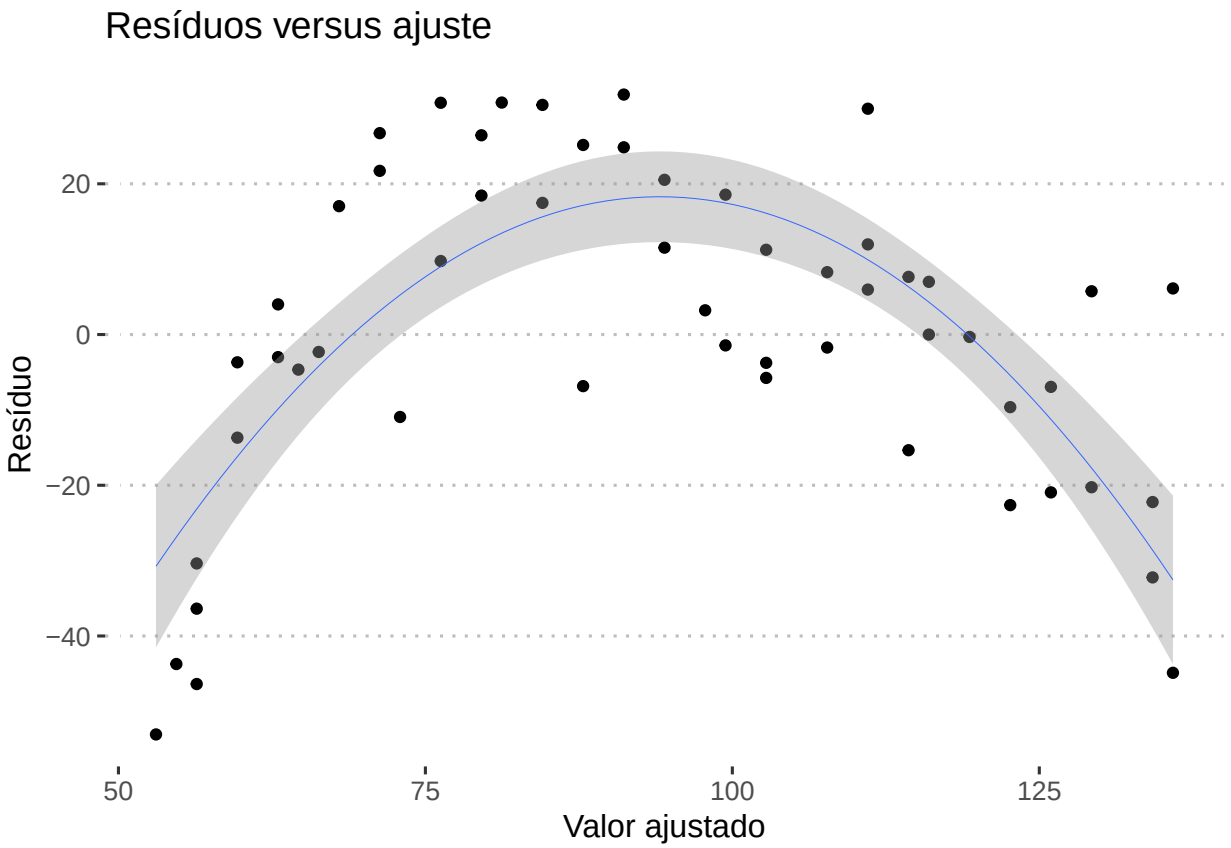
Com os dados carregados, primeiramente plotamos o gráfico de dispersão da Idade versus comprimento da mandíbula:

```
dados2 %>% ggplot() +
  geom_point(aes(x= age, y = bone))
```



Note um certo padrão não linear no gráfico acima, que reflete na falta de ajuste linear, explicitada no gráfico dos resíduos versus ajuste.

```
tibble(res = fit2$residuals, fit = fit2$fitted.values) %>%
  ggplot(aes(x = fit, y = res)) +
  geom_point() +
stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 0) +
  labs(title = "Resíduos versus ajuste",
       x = "Valor ajustado",
       y = "Resíduo") +
  theme_pubclean()
```



Nota-se um certo padrão quadrático no gráfico acima, o que nos sugere indícios da falta de ajuste do modelo.

ANOVA da falta de ajuste

```
pander(anova(fit2), "ANOVA da modelo ajustado")
```

Table 19: ANOVA da modelo ajustado

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
age	1	33908	33908	69.87	3.467e-11
Residuals	52	25237	485.3	NA	NA

A falta de ajuste pode ser quantificada através do teste ANOVA apropriado.

$H_0 : E[Y] = \beta_0 + \beta_1.X$, versus $H_1 : E[Y] \neq \beta_0 + \beta_1.X$

```
pander(anovaPE(fit2), "ANOVA da falta de ajuste do modelo linear")
```

Table 20: ANOVA da falta de ajuste do modelo linear

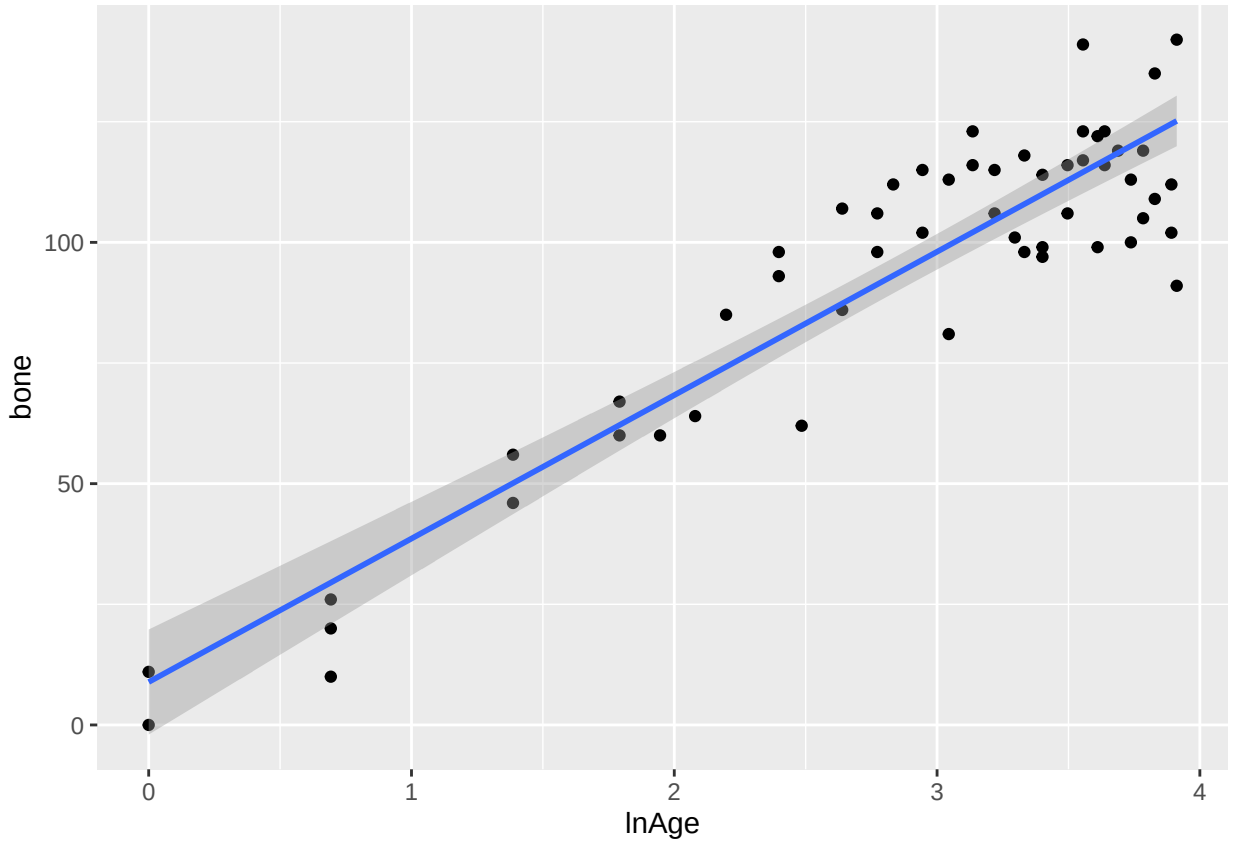
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
age	1	33908	33908	202.1	3.469e-13
Lack of Fit	28	21211	757.5	4.515	0.0001809
Pure Error	24	4026	167.8	NA	NA

Dessa maneira, ao nível de 5% de significância, temos evidências de que há falta de ajuste no modelo linear de regressão. Como alternativa, podemos efetuar uma transformação na variável `age`.

Transformação na variável dependente

Vamos efetuar a transformação $x = \ln(x)$ e analisar.

```
dados2 <- dados2 %>% mutate(lnAge = ifelse( age > 0,
                                    log(age),
                                    0))

dados2 %>% ggplot(aes(x= lnAge, y = bone)) +
  geom_point() +
  geom_smooth(formula = y~x, method = "lm")
```



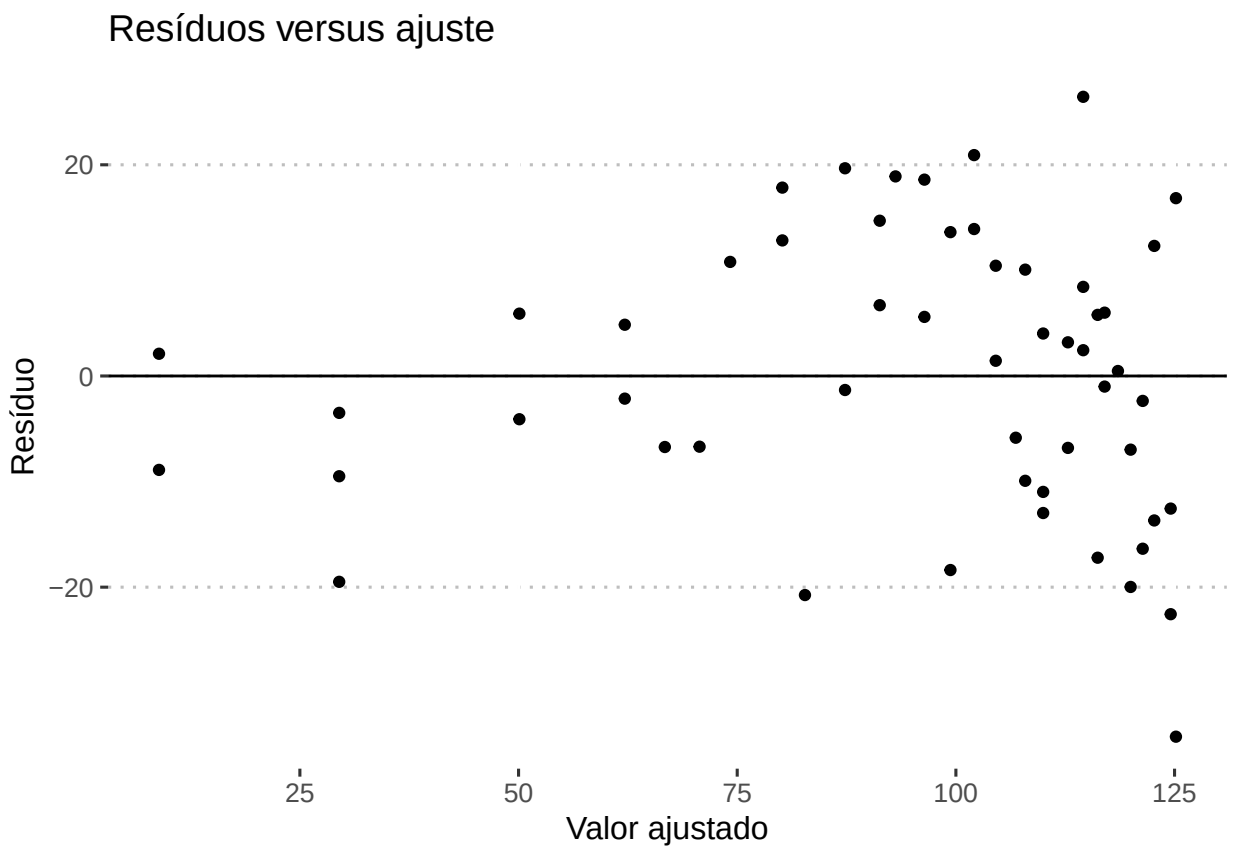
Aparentemente, a transformação contribuiu para a linearidade do modelo. Façamos agora o ajuste:

```
fit2_ln <- lm(bone ~ lnAge, data = dados2)

pander(fit2_ln)
```

Table 21: Fitting linear model: bone ~ lnAge

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	8.895	5.423	1.64	0.107
lnAge	29.72	1.784	16.66	1.689e-22

```
tibble(res = fit2_ln$residuals, fit = fit2_ln$fitted.values) %>%
  ggplot(aes(x = fit, y = res)) +
  geom_point() +
  geom_hline(yintercept = 0)+
  labs(title = "Resíduos versus ajuste",
       x = "Valor ajustado",
       y = "Resíduo") +
  theme_pubclean()
```



A transformação proposta aparentemente retirou o padrão indesejado de nossos dados. É necessário confirmarmos essa suposição através do mesmo teste realizado anteriormente.

```
pander(anovaPE(fit2_ln), "ANOVA da falta de ajuste do modelo linear transformado")
```

Table 22: ANOVA da falta de ajuste do modelo linear transformado

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
lnAge	1	49813	49813	304.7	1.629e-15
Lack of Fit	27	5246	194.3	1.188	0.3337
Pure Error	25	4087	163.5	NA	NA

Note que agora não rejeitamos H_0 a um nível de significância de 5%. Dessa forma, a transformação proposta foi capaz de solucionar a falta de ajuste do modelo.

Além disso, o SQEP e o SQFA são menores, sugerindo que o novo modelo é melhor em explicar a fonte de variabilidade dos dados.

10. Mínimos Quadrados Ponderados

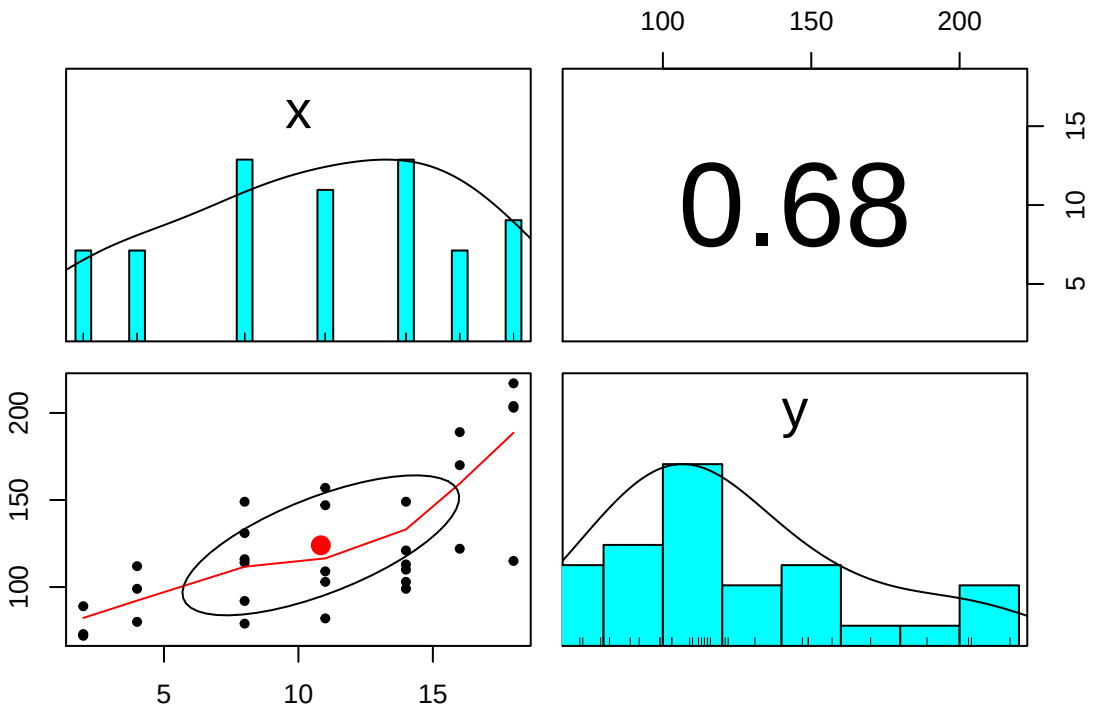
A Continuação será apresentado um exemplo simulado de falta de ajuste.

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,
     82,113,149,121,99,103,110,170,189,122,203,115,217,204)
gasto_venda= data.frame(cbind(x,y))
```

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

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

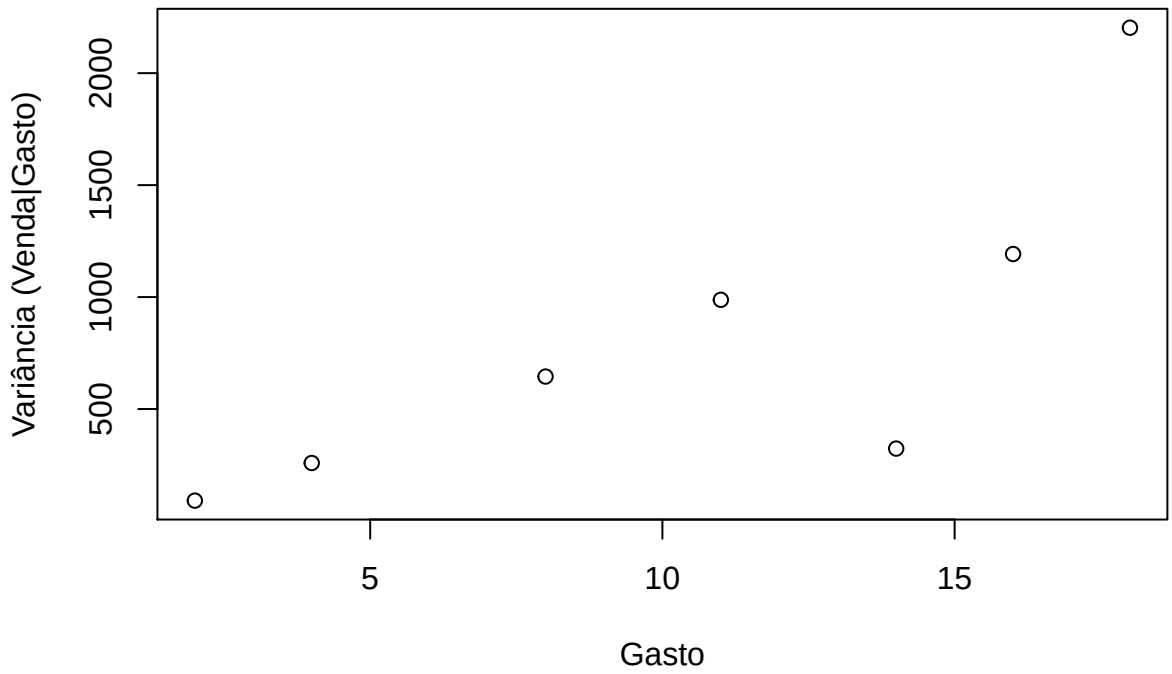


Portanto, devemos realizar o ajuste do modelo utilizando o Método de Mínimos Quadrados Ponderados. Para isso deve-se observar as estimativas do Erro Puro para cada nível de X, ou seja, os valores de Var(Y | X). Observe a função abaixo:

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

2	4	8	11	14	16	18
91	259	645.1	987.8	323.4	1192	2203

```
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 %>%
  pander()
```

gasto	wi
2	0.5
4	0.25
8	0.125
11	0.09091
14	0.07143
16	0.0625

gasto	wi
18	0.05556

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

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

Table 25: Fitting linear model: y ~ x

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	70.04	6.993	10.02	9.303e-11
x	4.978	0.8037	6.194	1.087e-06

Cálculo da Anova

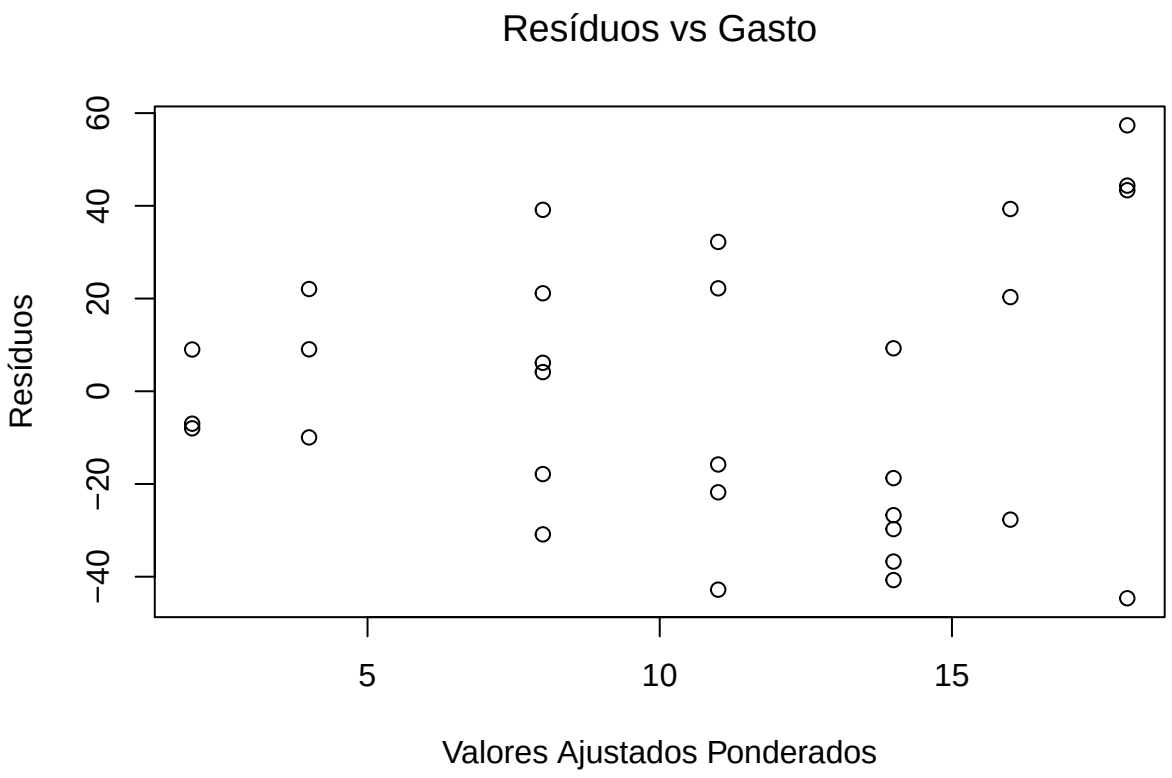
```
pander(anova(ajuste_ponderado))
```

Table 26: Analysis of Variance Table

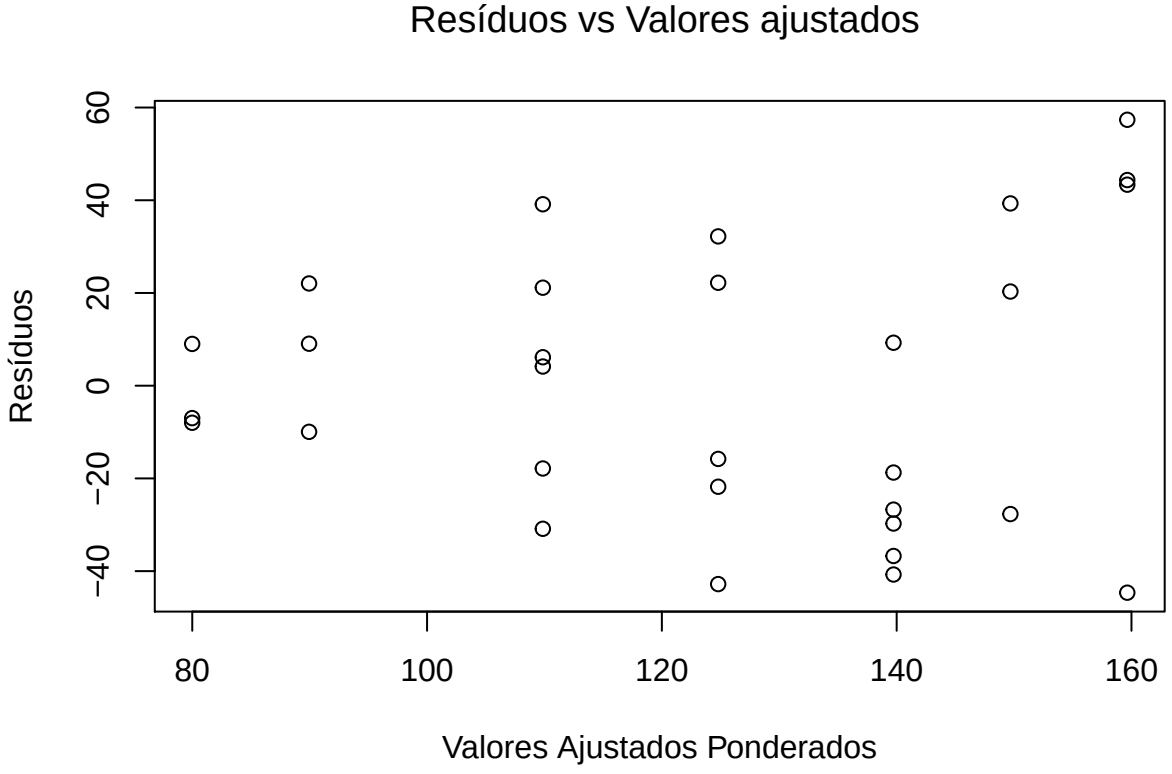
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x	1	2859	2859	38.37	1.087e-06
Residuals	28	2086	74.5	NA	NA

Gráficos para Análise dos Resíduos

```
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:

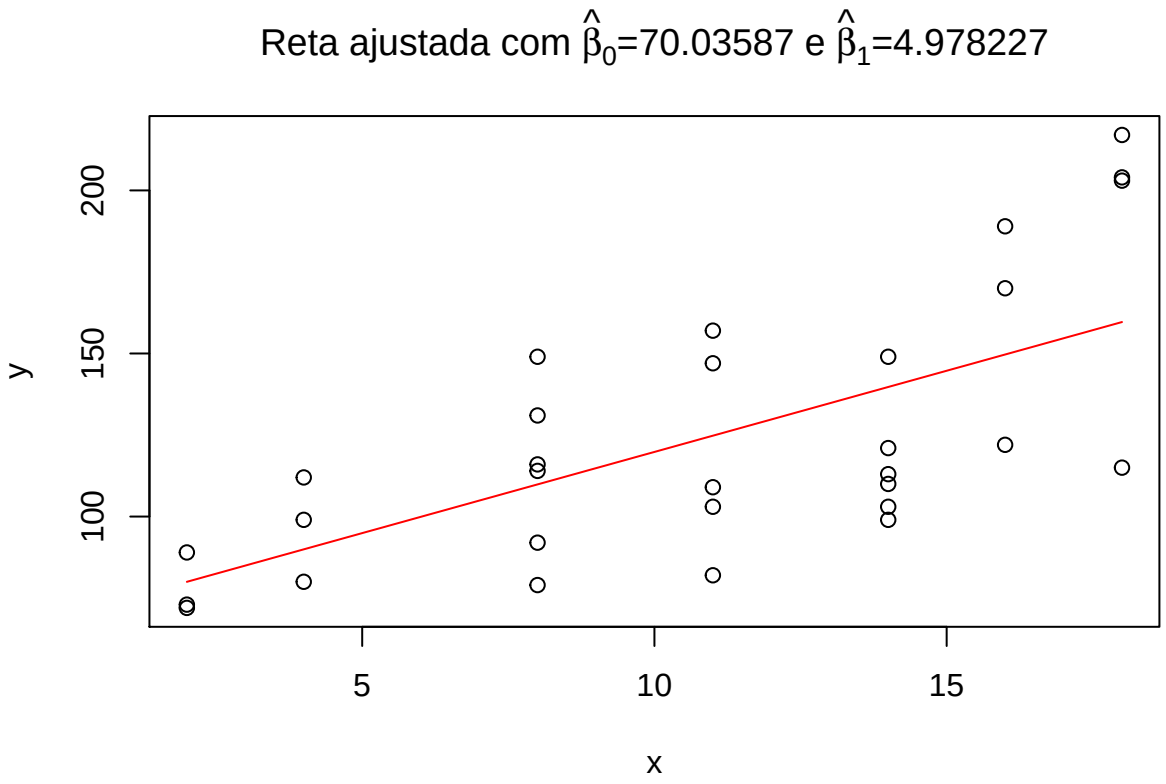
```
b0_est=ajuste_ponderado$coefficients[1]
b1_est=ajuste_ponderado$coefficients[2]
pander(ajuste_ponderado)
```

Table 27: Fitting linear model: y ~ x

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	70.04	6.993	10.02	9.303e-11
x	4.978	0.8037	6.194	1.087e-06

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')
```



Conclusão

Neste trabalho, pudemos continuar nossa análise do modelo de regressão linear múltipla obtido na atividade anterior, efetuar diagnósticos e confirmar nossas suposições a respeito das distribuição das quantidades de interesse envolvidas na modelagem.

O conjunto de dados analisado forneceu insights a respeito de como a independência de variáveis pode ocorrer, e como ela interfere na boa qualidade do ajuste, sendo necessário a utilização de uma grande gama de técnicas, desenvolvidas durante a disciplina.

Os últimos exercícios forneceram um aprendizado valioso em como encontrar, tratar dados para que satisfaçam as suposições necessárias para aplicar métodos como Falta de Ajuste do Modelo ou Mínimos Quadrados Ponderados. A oportunidade de desenvolver esses métodos através de uma abordagem prática foi muito valiosa aos integrantes.