



Universidade Presbiteriana
Mackenzie

Trilha de Aprendizagem 05

Regressão Linear Multivariada e Transformações de Variáveis

Mário Olímpio de Menezes



Conteúdo

I	Regressão Multivariada	
1	Regressão Linear Multivariada	5
	Análise Multivariada	5
	Regressão Linear Multivariada ou <i>Múltipla</i>	6
	Estudo de caso de regressão múltipla	9
	Representando Interações ou Efeito Moderador	14
	Diagnósticos da Regressão	16
	Regressão Múltipla com termo quadrático	20
II	Transformações de Variáveis	
2	Transformações de Variáveis	28
	Transformação na Variável Resposta	28
	Transformação na Variável Explicativa	39
	Finalizando	47
	Bibliografia	49



Regressão Multivariada



1. Regressão Linear Multivariada

Análise Multivariada

A técnica de Regressão Linear Multivariada, que é o foco desta Trilha, situa-se dentro do grande tópico das técnicas de Análise Multivariada.

A **Análise Multivariada** se refere a todas as técnicas que analisam simultaneamente múltiplas medidas de indivíduos ou objetos sob investigação. Assim, quaisquer análises simultâneas de mais do que duas variáveis pode ser considerada uma *análise multivariada* (HAIR JR et al. (2014)).

Dados multivariados surgem quando se medem várias variáveis para cada observação na amostra. A maioria dos conjuntos de dados coletados por pesquisadores em todas as áreas da ciência são multivariados (JOHNSON; WICHERN (1992)).

Nesta Trilha, além do tema da Regressão Linear Multivariada, também trataremos de Transformações de Variáveis e, no Material Complementar da Trilha, falamos de Seleção de Modelos. As Transformações de Variáveis são técnicas utilizadas quando se quer corrigir distorções encontradas na modelagem com relação às premissas estatísticas subjacentes ao método dos mínimos quadrados. Podemos fazer transformações na variável resposta ou transformações nas variáveis explicativas. Na seção sobre Seleção de Modelos abordaremos os critérios numéricos que nos permitem comparar modelos de regressão bem como algumas facilidades (bibliotecas e funções) que o **R** fornece para facilitar este processo de comparação e seleção de modelos. Apresentamos como Anexo desta Trilha, uma compilação sobre a criação de fórmulas no **R**; diversos tipos de fórmulas, seus significados e como podem ser utilizados são apresentados.

Regressão Linear Multivariada ou *Múltipla*

Quando temos mais do que uma variável preditora (explicativa), a regressão linear simples se transforma em **regressão linear multivariada**.

Assumamos que y_i representa o valor da variável resposta no i -ésimo indivíduo e que $x_{i1}, x_{i2}, x_{i3}, \dots, x_{iq}$ representam os valores individuais das q variáveis explicativas, com $i = 1, \dots, n$. O modelo de regressão linear multivariada (ou múltipla) é dado pela Equação (1):

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_q x_{iq} + \epsilon_i \quad (1)$$

O resíduo ou termo de erro ϵ_i , $i = 1, \dots, n$ são assumidos serem variáveis aleatórias independentes com uma distribuição normal, com média zero e variância constante σ^2 . Consequentemente, a distribuição da variável aleatória resposta y , também é normal, com um valor esperado dado pela combinação linear das variáveis explicativas:

$$E(y|x_1, \dots, x_q) = \beta_0 + \beta_1 x_1 + \dots + \beta_q x_q$$

e com variância σ^2 .

Os parâmetros do modelo β_k , $k = 1, \dots, q$ são os coeficientes da regressão; o coeficiente β_0 é a média global do modelo. Cada coeficiente representa a mudança esperada na variável resposta associada com uma mudança unitária na variável explicativa correspondente, quando as demais variáveis explicativas são mantidas constantes.

O relacionamento básico representado na regressão múltipla é a associação *linear* entre a variável dependente (métrica) e as variáveis independentes/explicativas (também métricas). O termo *linear* na regressão múltipla se aplica aos parâmetros da regressão (β_k , $k = 1, \dots, q$), não às variáveis explicativas ou à variável resposta. Consequentemente, modelos nos quais, por exemplo, o logaritmo de uma variável resposta é modelado em termos de funções quadráticas de algumas variáveis explicativas devem ser incluídos nesta classe de modelos *lineares*.



Outros exemplos onde temos variáveis explicativas não lineares em x mas lineares em β são:

- Uma regressão quadrática tem duas preditoras (X e X^2).
- A regressão cúbica tem três preditoras (X , X^2 , e X^3).
- Uma regressão polinomial é um caso especial de uma regressão múltipla.



Existem autores que tratam estes casos onde as variáveis explicativas tem relação não linear em x como *Regressão Não-Linear*.

Na Regressão Multivariada temos um problema adicional a ser tratado: o relacionamento entre as variáveis explicativas. Este problema é descrito como **Multicolinearidade** (HAIR JR et al. (2014)):

Multicolinearidade: A habilidade de uma variável independente **adicional** melhorar o modelo de regressão está relacionada não somente à sua correlação com a variável dependente, mas também às correlações da variável independente adicional com as outras variáveis independentes já presentes no modelo.

- **Colinearidade** é a associação, medida como correlação, entre duas variáveis independentes.
- **Multicolinearidade** se refere à correlação entre três ou mais variáveis independentes, (evidenciada quando uma é *regredida* em relação às outras).

O impacto da multicolinearidade é reduzir qualquer poder preditivo de uma variável independente única pela extensão a qual ela está associada com outra variável independente.

Conforme a colinearidade aumenta, a variância única explicada por cada variável independente diminui e o percentual de predição compartilhada aumenta. Como esta predição compartilhada somente conta uma vez, a predição total *aumenta* muito mais lentamente quando variáveis altamente correlacionadas são adicionadas ao modelo.

Para maximizar a predição de um dado número de variáveis independentes, devemos procurar aquelas que tenham baixa multicolinearidade com outras variáveis independentes mas que **também** tenham alta correlações com a variável dependente.

Predição com Regressão Múltipla

Um propósito fundamental da regressão múltipla é prever a variável dependente com um conjunto de variáveis independentes.

Fazendo isso, a regressão múltipla cumpre um de dois objetivos:

- O primeiro objetivo é maximizar o poder preditivo total das variáveis independentes, conforme representado na equação de regressão obtida (juntamente com os coeficientes)
 - Acurácia da predição é um ponto crítico para esta avaliação.
- O segundo objetivo é a comparação de dois ou mais conjuntos de variáveis independentes para avaliar o poder preditivo de cada equação de regressão, ou seja, encontrar o melhor

subconjunto de variáveis que resulta no melhor modelo.

Transformações de Variáveis

Um problema frequentemente encontrado na regressão múltipla é a incorporação de dados não-métricos, tais como gênero, ocupação, etc., na equação de regressão, ou seja, variáveis categóricas. Isso porque a regressão múltipla é limitada a dados métricos (numéricos).

Quando temos variáveis destes tipos (nominal ou ordinal), elas devem ser transformadas em variáveis numéricas utilizando um esquema de codificação; dentre os esquemas possíveis de transformação, há a codificação de zeros e uns, chamada de *dummy coding*. Assumindo que x_i seja um fator com k níveis, a submatriz de \mathbf{X} correspondente a x_i é uma matriz $n \times k$ de zeros e uns, onde o j -ésimo elemento na i -ésima linha é um quando x_{i1} estiver no j -ésimo nível.



Veja no Material Complementar da Trilha exemplos destes tipos de codificação, incluindo um feito no **R**.

Outro problema (ou restrição), é a inability de se representar diretamente relacionamentos não lineares das variáveis preditoras (independentes).

Uma alternativa para estas situações (relacionamentos não lineares) é a criação de novas variáveis através de transformações algébricas que eliminam os termos não lineares.

Outro uso para transformações de variáveis é para acertar violações de alguma das premissas (hipóteses) estatísticas.

Assim, temos duas razões básicas para transformarmos variáveis:

- Melhorar ou modificar o relacionamento entre as variáveis dependente e independentes (não linearidade ou violação de premissas estatísticas do método de mínimos quadrados).
- Habilitar o uso de variáveis não métricas na equação de regressão (*dummy coding*).

Veremos exemplos destas transformações ao longo do tema regressão múltipla e também regressão generalizada.

Estudo de caso de regressão múltipla

Vamos fazer um estudo de caso, um exemplo geral. Utilizaremos a base de dados `state.x77` que faz parte da instalação base do R. Queremos explorar o relacionamento entre a taxa de assassinatos de um estado e outras características do estado, incluindo população, grau de analfabetismo, renda média e níveis de frio (número médio de dias abaixo de zero).

A base de dados `state.x77` está contida em uma matriz e a função `lm()` requer que os dados estejam em um `data.frame`. Então precisamos acertar isso antes de prosseguir explorando os dados.

```
> states <- as.data.frame(
...       state.x77[,c("Murder", "Population", "Illiteracy", "Income", "Frost")]
...       )
```

Vamos explorar um pouco nossa base de dados para ganharmos mais *insights* sobre ela.

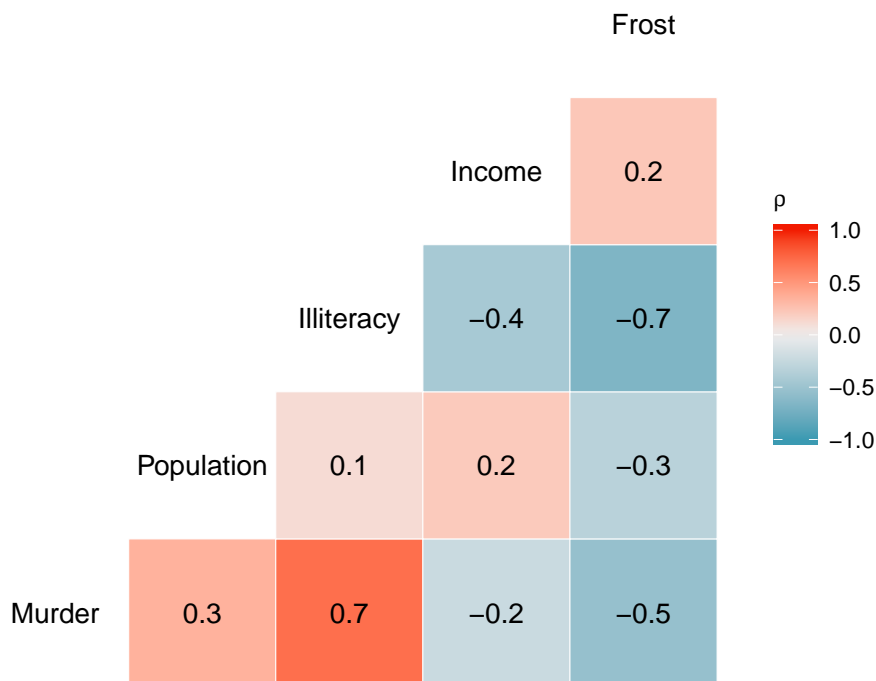
Explorando os dados

Um bom começo da regressão múltipla é examinar os relacionamentos entre as variáveis, duas de cada vez. A função `cor()` pode ser utilizada para fornecer uma *matriz* das correlações entre as variáveis.

O pacote `GGally` fornece algumas opções interessantes de visualização de dados. Por exemplo, a função `ggcorr` permite visualizar as correlações graficamente, com cores e rótulos mais indicativos, como mostrado na Figura 1.

```
> library(ggplot2)
> library(ggpubr)
> library(ggfortify)
> library(GGally)
> ggcorr(states, palette = "RdYlGn", name = bquote(rho),
...       label = TRUE, label_color = "black") +
...   labs( caption = "Fonte: Elaborado pelo autor") +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
```

Também podemos criar alguns gráficos especiais, do tipo pares *scatter plots* para inspecionarmos visualmente os relacionamentos.

Figura 1: Correlação entre variáveis da base state.x77

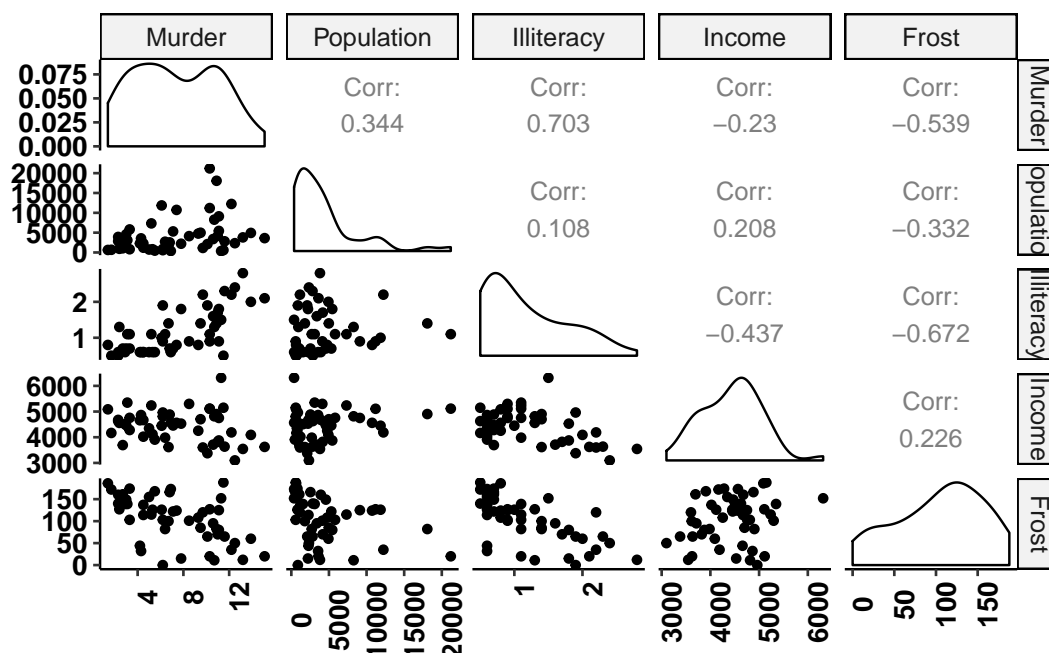
Fonte: Elaborado pelo autor

A função `ggpairs` do mesmo pacote `Ggally` permite esta visualização, agrupando no mesmo gráfico: curva de densidade, gráfico de dispersão (*scatter plots*) e a correlação entre as respectivas variáveis, como mostrado na Figura 2.

```
> ggpairs(states, columns = 1:ncol(states), title = "", axisLabels = "show") +
...   theme(axis.text.x = element_text(angle=90, vjust=0.5, size=10)) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
```

Do gráfico mostrado na Figura 2 podemos ver que a variável *Murder* (taxa de assassinato) possui uma distribuição bimodal, o que é um problema para o tipo de modelagem que estamos tratando. Além disso, vemos também que cada variável preditora tem alguma distorção em sua distribuição. A taxa de assassinato aumenta com a população e com o analfabetismo; e ela cai com o aumento da renda média e o número de dias frios. Ao mesmo tempo, observamos que os estados mais frios tem menores taxas de analfabetismo e população, mas tem renda média maior.

Figura 2: Visualização de Curva de Densidade, Gráfico de Dispersão e correlações das variáveis do conjunto de dados state.x77



Fonte: Elaborado pelo autor

Ajustando um Modelo de Regressão Linear Multivariada

Vamos utilizar a função `lm()` para fazer o ajuste multivariado

```
> fit <- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)
> summary(fit)
```

Call:

```
lm(formula = Murder ~ Population + Illiteracy + Income + Frost,
    data = states)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.7960	-1.6495	-0.0811	1.4815	7.6210

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.235e+00	3.866e+00	0.319	0.7510
Population	2.237e-04	9.052e-05	2.471	0.0173 *
Illiteracy	4.143e+00	8.744e-01	4.738	2.19e-05 ***

```

Income      6.442e-05  6.837e-04  0.094  0.9253
Frost       5.813e-04  1.005e-02  0.058  0.9541
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.535 on 45 degrees of freedom
Multiple R-squared:  0.567, Adjusted R-squared:  0.5285
F-statistic: 14.73 on 4 and 45 DF, p-value: 9.133e-08

```

Analizando o Modelo Multivariado Ajustado

Quando temos mais do que uma variável preditora, os coeficientes de regressão indicam o aumento na variável dependente para uma unidade de mudança na variável preditora, mantendo-se todas as outras variáveis predictoras constante:

- No nosso exemplo, **ainda com todas as variáveis no modelo**, o coeficiente de regressão para Illiteracy é 4.143, sugerindo que um aumento de 1% no analfabetismo está associado com um aumento de **4.143%** na taxa de assassinato.
 - O coeficiente é diferente de zero com nível de significância de 0.05: $P < 0.0001$
- Por outro lado, o coeficiente para Frost não é significativamente diferente de zero ($p = 0.954$) sugerindo que Frost e Murder não estão linearmente relacionadas quando se controla as outras variáveis predictoras.
- O mesmo ocorre com Income, isto é, não é significativamente diferente de zero ($p = 0.925$).
- Em conjunto, as variáveis predictoras respondem por 57% da variância na taxa de assassinato dos estados (*R-quadrado*); já o valor do *R-quadrado ajustado* é de 0.5285.

Depois de identificarmos variáveis que não são estatisticamente significantes, podemos **atualizar** nosso modelo, removendo-as. O processo de remoção deve ser feito uma variável de cada vez, com uma reavaliação do modelo a cada passo. Isto porque, ao se remover uma variável, as restantes serão afetadas, podendo passar a ter significância estatística ou deixando de ter.

Começamos com Frost:

```

> fit <- update(fit, . ~ . - Frost)
> summary(fit)

Call:
lm(formula = Murder ~ Population + Illiteracy + Income, data = states)

```

```

Residuals:
    Min       1Q   Median       3Q      Max
-4.7846 -1.6768 -0.0839  1.4783  7.6417

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.3402721   3.3694210   0.398  0.6926
Population    0.0002219   0.0000842   2.635  0.0114 *
Illiteracy    4.1109188   0.6706786   6.129 1.85e-07 ***
Income        0.0000644   0.0006762   0.095  0.9245
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.507 on 46 degrees of freedom
Multiple R-squared:  0.5669,    Adjusted R-squared:  0.5387
F-statistic: 20.07 on 3 and 46 DF,  p-value: 1.84e-08

```

Continuamos nossa **atualização** do modelo, removendo as variáveis que não tiveram significado estatístico. Income é a próxima candidata a remoção.

```

> fit <- update(fit, . ~ . - Income)
> summary(fit)

Call:
lm(formula = Murder ~ Population + Illiteracy, data = states)

Residuals:
    Min       1Q   Median       3Q      Max
-4.7652 -1.6561 -0.0898  1.4570  7.6758

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.652e+00   8.101e-01   2.039  0.04713 *
Population    2.242e-04   7.984e-05   2.808  0.00724 **
Illiteracy    4.081e+00   5.848e-01   6.978 8.83e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.481 on 47 degrees of freedom
Multiple R-squared:  0.5668,    Adjusted R-squared:  0.5484

```


F-statistic: 30.75 on 2 and 47 DF, p-value: 2.893e-09

Chegamos agora a um modelo que tem apenas duas variáveis explicativas; a equação do nosso modelo é:

$$\text{Murder} = 1.652 + 0.0002242 \times \text{Population} + 4.081 \times \text{Illiteracy}$$

Ou seja, o aumento de 1 ponto percentual na taxa de analfabetismo, implica no aumento de 4.081% na taxa de assassinato. Observe a pequena mudança em relação ao primeiro modelo!

O aumento da população tem um impacto bem menor sobre a taxa de assassinato; um aumento de cerca de 10.000 na População está associado a um aumento aproximado de 2% na taxa de assassinato.

Representando Interações ou Efeito Moderador

Em uma regressão linear multivariada, podemos descobrir que o modelo linear deve considerar **interações** entre as variáveis se quiser ser bem sucedido. Estas interações são também conhecidas como **efeito moderador**, que significa que o relacionamento entre um variável independente e a dependente é *afetado* por outra variável independente.

Para estudarmos este efeito, vamos utilizar a base de dados mtcars que faz parte da instalação base do **R**, através do pacote datasets. Veja o help desta base de dados para mais informações sobre ela (?mtcars)

Estamos interessados no impacto do peso e da potência dos automóveis na consumo de combustível (*mileage*). Vamos construir um modelo de regressão que inclua ambas as variáveis preditoras, juntamente com sua interação:

```
> fit.mpg <- lm(mpg ~ hp + wt + hp:wt, data=mtcars)
```

```
> summary(fit.mpg)
```

Call:

```
lm(formula = mpg ~ hp + wt + hp:wt, data = mtcars)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.0632	-1.6491	-0.7362	1.4211	4.5513

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.80842    3.60516   13.816 5.01e-14 ***
hp           -0.12010    0.02470   -4.863 4.04e-05 ***
wt           -8.21662    1.26971   -6.471 5.20e-07 ***
hp:wt         0.02785    0.00742    3.753 0.000811 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.153 on 28 degrees of freedom
Multiple R-squared:  0.8848,    Adjusted R-squared:  0.8724
F-statistic: 71.66 on 3 and 28 DF,  p-value: 2.981e-13

```

Podemos ver da coluna com os $\text{Pr}(>|t|)$ que a interação entre *horsepower* e *peso do carro* é significativa.



O que isso significa?

Uma interação significativa entre duas variáveis preditoras nos diz que o relacionamento entre uma variável preditora e a variável resposta depende do nível da outra preditora. Aqui, significa que o relacionamento entre *milhas por galão* e *horsepower* varia conforme o peso do carro.

O modelo de previsão de mpg é

$$\hat{mpg} = 49.81 - 0.12 \times hp - 8.22 \times wt + 0.03 \times hp \times wt$$

Para interpretar a interação, podemos colocar diversos valores de wt e simplificar a equação. Dentre os valores possíveis, utiliza-se a média (3.2) e também um desvio padrão para cima e um para baixo da média (2.2 e 4.2), respectivamente.

Para $wt = 2.2$, a equação simplifica para:

$$\begin{aligned}\hat{mpg} &= 49.81 - 0.12 \times hp - 8.22 \times (2.2) + 0.03 \times hp \times (2.2) \\ \hat{mpg} &= 31.41 - 0.06 \times hp\end{aligned}$$

Para $wt = 3.2$ a equação se torna: $\hat{mpg} = 23.37 - 0.03 \times hp$

Para $wt = 4.2$ a equação se torna: $\hat{mpg} = 15.33 - 0.003 \times hp$

Podemos ver que conforme o peso do carro aumenta (2.2, 3.2, 4.2), a mudança esperada em mpg por aumento unitário em hp diminui (0.06, 0.03, 0.003). Podemos visualizar as interações utilizando a função `effect()` do pacote `effects` juntamente com seu método `plot`. A função `effect()` retorna um objeto que contém os valores ajustados do modelo para cada valor da variável especificada pelo parâmetro `xlevels`. Veja o help da função para mais detalhes sobre sua utilização (`?effect`).

Para visualizarmos o gráfico dos efeitos, o formato da chamada da função é:

```
plot(effect(term, mod, , xlevels), multiline=TRUE)
```

Ao invés de utilizarmos esta função diretamente, vamos utilizar o objeto `eff` retornado e construir um `data.frame` para ser utilizado com o `ggplot2` e termos mais controle sobre o aspecto visual final do gráfico, conforme mostrado na Figura 3.

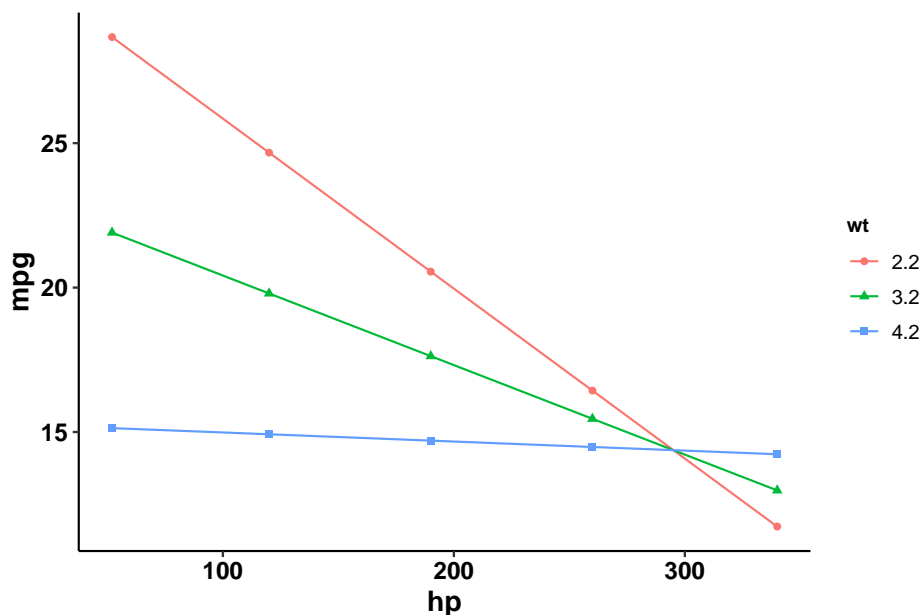
```
> library(effects)
> efeitos <- effect("hp:wt", fit.mpg, , list(wt=c(2.2,3.2,4.2)))
> df.ef <- as.data.frame(efeitos)
> df.ef$wt <- as.factor(df.ef$wt)
> ggplot(data = df.ef) +
...   geom_line(aes(y = fit, x = hp, shape = wt, color = wt)) +
...   geom_point(aes(x = hp, y = fit, shape = wt, color = wt)) +
...   ylab("mpg") + theme_pubr(legend = "right") + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
```

Podemos ver pelo gráfico da Figura 3 que conforme o peso do carro aumenta, o relacionamento entre *horsepower* e *milhas por galão* se enfraquece. Para $wt = 4.2$, a linha é praticamente horizontal, indicando que conforme wt aumenta, mpg não muda.

Diagnósticos da Regressão

Ajustar o modelo é somente parte da etapa de análise. Uma vez que ajustamos um modelo de regressão, precisamos avaliar se conseguimos atingir as hipóteses estatísticas subjacentes à nossa abordagem antes de pensarmos em intervalos de confiança.

Até aqui, utilizamos a função `summary()` para termos os parâmetros do modelo e um sumário das estatísticas. Infelizmente, como já vimos na Trilha de Regressão Linear Simples, **nada** na saída

Figura 3: Efeito da Interação hp:wt no consumo (mpg)

Fonte: Elaborado pelo autor

da função `summary(model)` nos diz se o nosso modelo é apropriado, ou seja, que ele satisfaz as hipóteses estatísticas subjacentes.

Nossa confiança nas inferências sobre os parâmetros da regressão dependem do grau em que conseguimos atender as hipóteses estatísticas do modelo de **minimos quadrados ordinários – OLS**.

!

Por que isso é importante?

- Irregularidades nos dados ou uma especificação errada dos relacionamentos entre as variáveis preditoras e a variável resposta pode nos levar a especificar um modelo amplamente impreciso.
 - Também podemos concluir que uma variável preditora e a variável resposta não estão relacionadas, quando na verdade, estão.
 - Ou o contrário!
-

Vamos começar nosso diagnóstico examinando os intervalos de confiança dos coeficientes; utilizaremos o modelo completo para uma análise *didática* apenas. Normalmente, podemos utilizar apenas os coeficientes do modelo já reduzido.

```
> fit.coef <- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)
> confint(fit.coef)
```

	2.5 %	97.5 %
(Intercept)	-6.552191e+00	9.0213182149
Population	4.136397e-05	0.0004059867
Illiteracy	2.381799e+00	5.9038743192
Income	-1.312611e-03	0.0014414600
Frost	-1.966781e-02	0.0208304170

Os resultados sugerem que podemos estar 95% confiantes de que o intervalo [2.38, 5.90] contém a mudança verdadeira na taxa de assassinato para uma mudança de 1% na taxa de analfabetismo.

Adicionalmente, como o intervalo de confiança de Frost contém 0, podemos concluir que uma mudança na temperatura não está relacionado à taxa de assassinato, mantendo-se as outras variáveis constantes.

Mas nossa *fé* neste modelo deve ser tão forte quanto as evidências que temos sobre **se** nossos dados satisfazem as hipóteses estatísticas no modelo subjacente.

Vamos fazer um diagnóstico do nosso modelo verificando a *homocedasticidade* e também o comportamento dos resíduos com relação aos quantis teóricos, ou seja, o gráfico **QQ-plot**.

Ao invés de utilizarmos a função `plot()` do objeto retornado por `lm()` para obtermos os gráficos diagnósticos, vamos utilizar o `ggplot2` através da biblioteca `ggfortify`.

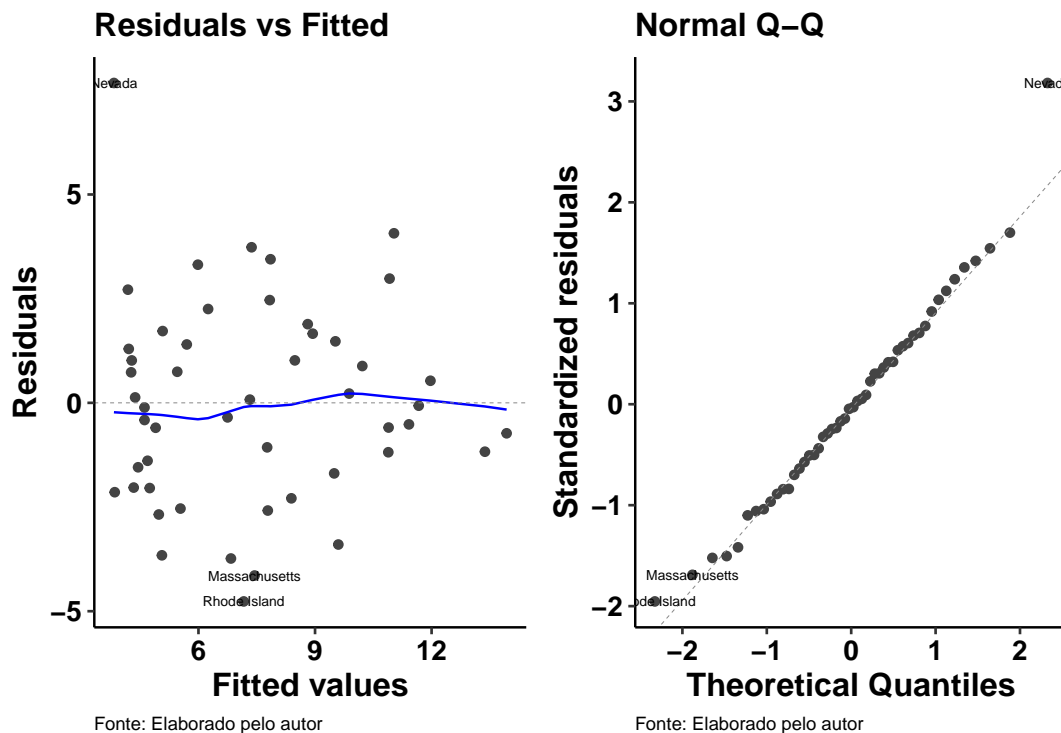
O `ggplot2` não consegue acessar alguns tipos de dados (na verdade, ele precisa sempre de um `data.frame`) tal como faz a função `plot` do sistema gráfico base; então, para conseguirmos obter os gráficos diagnósticos pelo `ggplot2` vamos utilizar a função `autoplot` do pacote `ggfortify` (`?autoplot`). Vamos selecionar apenas os dois primeiros gráficos, como mencionado acima. Os gráficos diagnósticos são mostrados na Figura 4.

```
> autoplot(fit, which = 1:2, ncol = 2, label.size = 2) + theme_pubr() +
... labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
... theme(plot.caption = element_text(hjust = 0, size= 8))
```

Teoricamente, os dois gráficos devem ter os seguintes comportamentos:

- O gráfico *Residuals vs Fitted* mostra os resíduos no eixo y contra os valores ajustados no eixo x. Não se deve observar estruturas ou padrões no gráfico. Os pontos devem se

Figura 4: Gráficos Diagnósticos do Modelo Ajustado



parecer como o céu à noite. É um problema se os pontos se espalham conforme os valores ajustados ficam maiores – como se fosse uma fatia de queijo.

- O gráfico **Normal Q-Q** (QQ-Plot) que deve ser uma linha reta se os erros são normalmente distribuídos. Se o gráfico tivesse a forma de um **S** ou de uma **banana** precisaríamos ajustar um modelo diferente.

Analisando os nossos gráficos diagnósticos acima temos:

- Os resíduos do nosso modelo tem um comportamento bem próximo do que se espera, isto é, não há um aumento dos resíduos com o aumento da variável dependente. Isto significa que nosso modelo apresenta *homocedasticidade* adequada.
- Os resíduos do nosso modelo também apresentam uma distribuição não muito divergente de uma distribuição normal, exceto pela parte inicial (4 ou 5 primeiros pontos), como pode ser observado do gráfico *Normal Q-Q*.
- Ademais, as observações destacadas (rotuladas) nos gráficos indicam possíveis problemas em termos de alavancagem e/ou outliers. No momento não abordaremos estes possíveis problemas.

Regressão Múltipla com termo quadrático

Vamos fazer mais um estudo de caso, demonstrando a utilização de termos quadráticos no modelo de regressão múltipla. Como já falamos, os coeficientes do modelo **permanecem** lineares, mas as variáveis preditoras (explicativas) podem ter outro relacionamento.

Vamos utilizar o conjunto de dados `women` da instalação base do **R**, que provê a altura e o peso para um conjunto de 15 mulheres com idades entre 30 a 39 anos. Queremos prever o peso a partir da altura, isto é, iniciaremos com um modelo de regressão linear simples do tipo `peso ~ altura`.

Iniciamos com uma exploração rápida dos dados.

```
> summary(women)
      height      weight
Min.   :58.0  Min.   :115.0
1st Qu.:61.5  1st Qu.:124.5
Median :65.0  Median :135.0
Mean   :65.0  Mean   :136.7
3rd Qu.:68.5  3rd Qu.:148.0
Max.   :72.0  Max.   :164.0
```

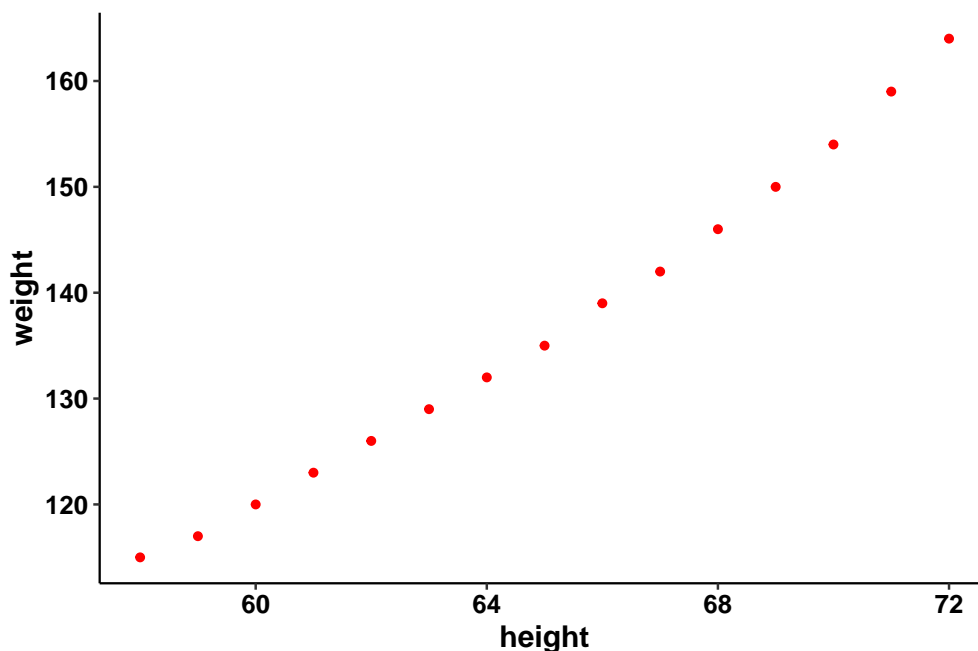
Uma inspeção na estrutura:

```
> str(women)
'data.frame':  15 obs. of  2 variables:
 $ height: num  58 59 60 61 62 63 64 65 66 67 ...
 $ weight: num 115 117 120 123 126 129 132 135 139 142 ...
```

E uma inspeção visual para uma primeira ideia sobre o relacionamento entre as duas variáveis de interesse, como mostrado na Figura 5.

```
> g1 <- ggplot(data = women) +
...   geom_point(aes(x = height, y = weight), color = "red") +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
> g1
```

Vamos então iniciar nossa modelagem com uma regressão linear simples do *peso* como uma

Figura 5: Relacionamento entre peso e altura das mulheres

Fonte: Elaborado pelo autor

função da *altura*, i.e., $\text{weight} \sim \text{height}$

```
> fit.w1 <- lm(weight ~ height, data=women)
```

```
> summary(fit.w1)
```

Call:

```
lm(formula = weight ~ height, data = women)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.7333	-1.1333	-0.3833	0.7417	3.1167

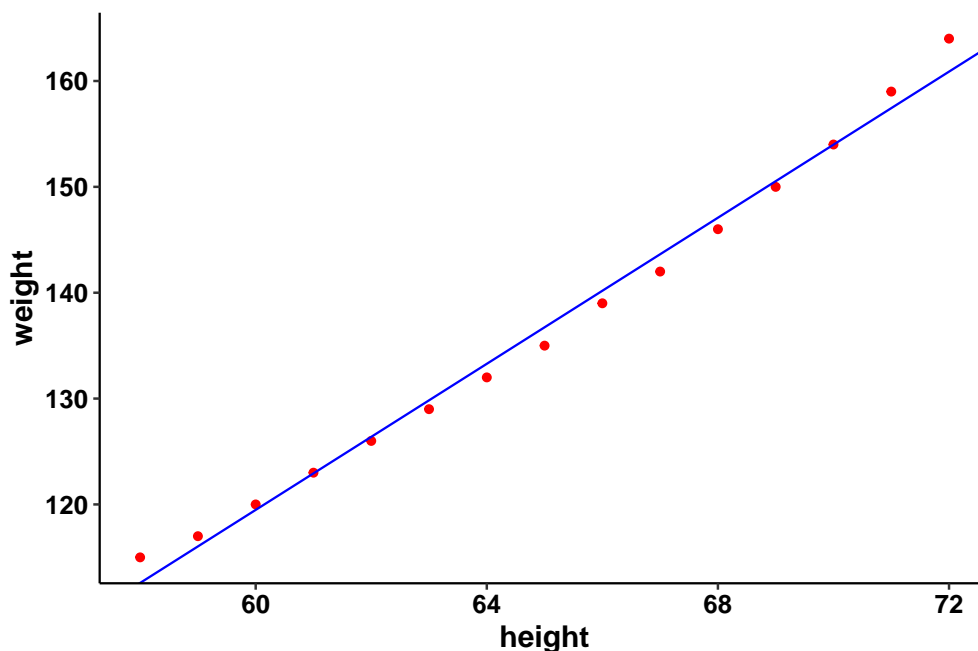
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-87.51667	5.93694	-14.74	1.71e-09 ***
height	3.45000	0.09114	37.85	1.09e-14 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.525 on 13 degrees of freedom

Multiple R-squared: 0.991, Adjusted R-squared: 0.9903

Figura 6: Peso como função da altura, com modelo linear

Fonte: Elaborado pelo autor

F-statistic: 1433 on 1 and 13 DF, p-value: 1.091e-14

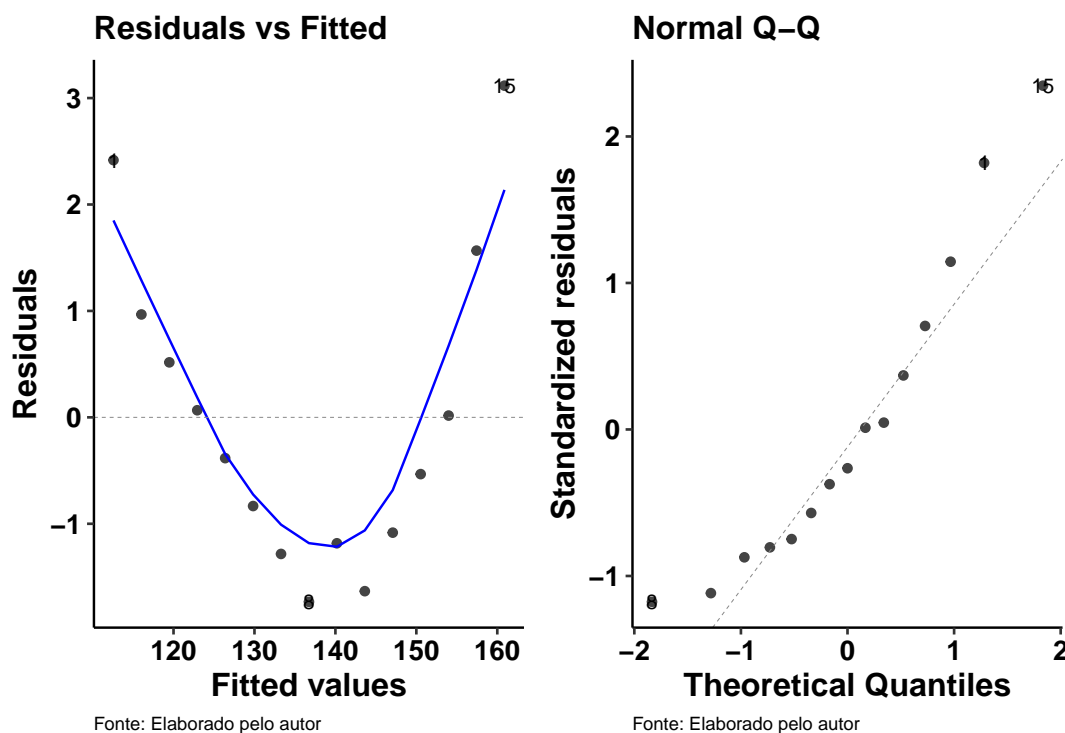
Analisando nosso modelo pelo sumário da regressão podemos ser levados a pensar que **já** temos um modelo ótimo, afinal, conseguimos explicar 99.03% da variância dos nossos dados; além disso, os dois coeficientes encontrados **tem** significância estatística a um nível de 5%.

Vamos visualizar nosso modelo juntamente com os pontos de dados, como mostrado na Figura 6.

```
> slope = as.numeric(fit.w1$coefficients[2])
> interc = as.numeric(fit.w1$coefficients[1])
> g1 <- ggplot(data = women) +
...   geom_point(aes(x = height, y = weight), color = "red") +
...   geom_abline(slope = slope, intercept = interc,
...               color = "blue", data = women) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
> g1
```

Pela figura acima, o modelo linear simples apesar de explicar um elevado percentual da variância

Figura 7: Gráficos Diagnósticos da Regressão - Base women



dos dados, parece não ser o que mais se adequa ao formato dos nossos dados.

Continuamos nossa avaliação do modelo, e como já falamos outras vezes, precisamos nos certificar que nosso modelo atende às premissas estatísticas do método de mínimos quadrados ordinários OLS.

Para isso, utilizamos novamente os gráficos diagnósticos dos resíduos e o QQ-Plot, como mostrado na Figura 7.

```
> autoplot(fit.w1, which = 1:2, ncol = 2, label.size = 3) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
```

O gráfico dos resíduos do nosso modelo claramente aponta para uma inapropriação de um modelo linear simples para descrever nossos dados. O gráfico apresenta um aspecto de **U**.

Da mesma forma, o gráfico QQ-Plot apresenta uma forma acentuada de *banana*, indicando também que os resíduos não seguem bem uma distribuição normal.

Como vimos na fase exploratória, nossos dados apresentam uma forma levemente curva; esta forma curva, aliado ao que observamos no formato do gráfico dos resíduos (**U**), nos leva à seguinte decisão: incluir um termo quadrático no nosso modelo, isto é, um termo do tipo X^2 para capturarmos este comportamento dos dados no modelo.

A inclusão do termo quadrático no modelo é feita com a função **I**. Esta função, que significa *as-is*, indica para o **R** que ele deve interpretar o termo do modelo tal como está escrito, neste caso, um termo quadrático height^2 . Isto é necessário pois em uma fórmula do **R**, a expressão height^2 indicaria uma interação da variável com ela mesma.



Veja a seção *Criando fórmulas no R* no Material Complementar da Trilha.

```
> fit.w2 <- lm(weight ~ height + I(height^2), data=women)
```

```
> summary(fit.w2)
```

Call:

```
lm(formula = weight ~ height + I(height^2), data = women)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.50941	-0.29611	-0.00941	0.28615	0.59706

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	261.87818	25.19677	10.393	2.36e-07 ***
height	-7.34832	0.77769	-9.449	6.58e-07 ***
I(height^2)	0.08306	0.00598	13.891	9.32e-09 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3841 on 12 degrees of freedom

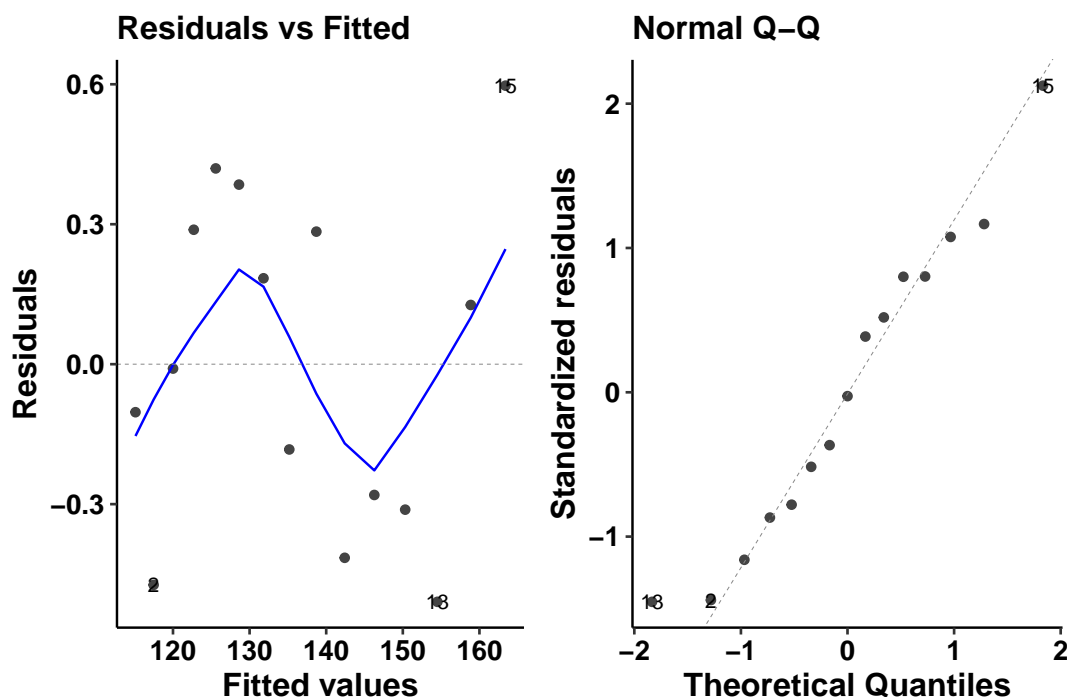
Multiple R-squared: 0.9995, Adjusted R-squared: 0.9994

F-statistic: 1.139e+04 on 2 and 12 DF, p-value: < 2.2e-16

Novamente, analisamos o sumário do nosso modelo e verificamos que temos significância estatística para todos os coeficientes, incluindo o termo quadrático e agora conseguimos explicar 99.94% da variância dos nossos dados!

Vamos então aos gráficos diagnósticos, como mostrados na Figura 8.

Figura 8: Gráficos Diagnósticos da Regressão, com termo quadrático - Base women



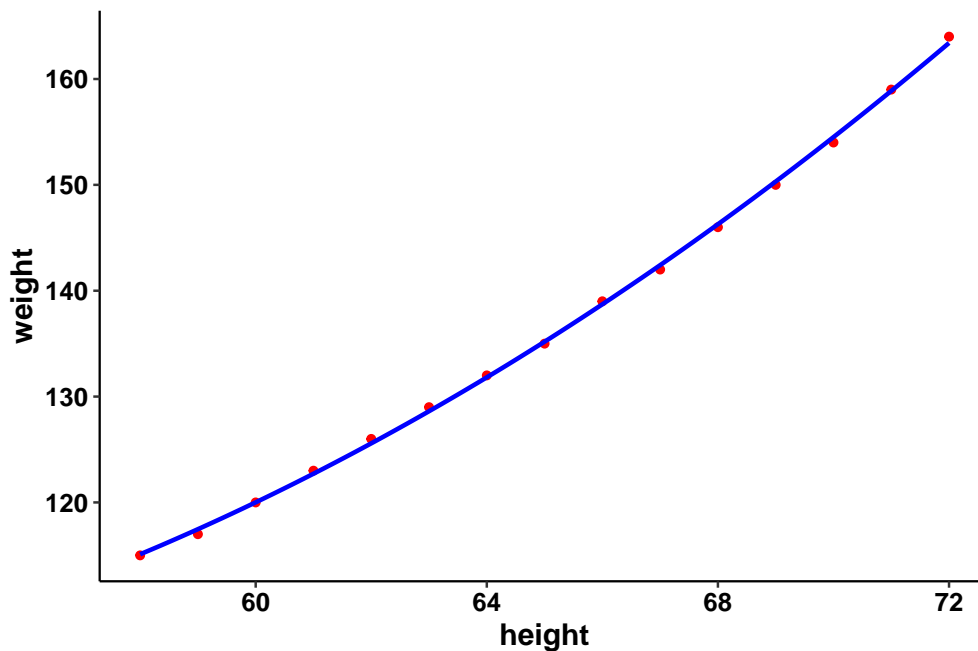
Fonte: Elaborado pelo autor

Fonte: Elaborado pelo autor

```
> autoplot(fit.w2, which = 1:2, ncol = 2, label.size = 3) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
```

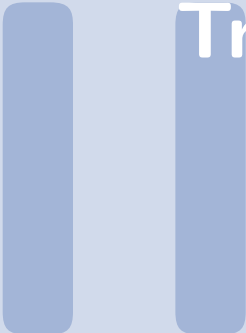
Analisando os gráficos vemos que a *homocedasticidade* dos resíduos agora está mais aceitável, ou seja, a variância se mantém dentro da mesma faixa ao longo dos valores ajustados. Da mesma forma, o gráfico QQ-Plot também mostra que os resíduos agora apresentam uma distribuição bem mais próxima de uma normal. Vamos examinar novamente o modelo ajustado, mostrado no gráfico da Figura 9.

```
> g2 <- ggplot(data = women, aes(x = height, y = weight)) +
...   geom_point(color = "red") +
...   geom_smooth( method = "lm", formula = y ~ x + I(x^2),
...               se = FALSE, color = "blue", data = women) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
> g2
```

Figura 9: Peso como função da altura, com modelo com termo linear quadrático

Fonte: Elaborado pelo autor

Como pode ser observado no gráfico acima, temos agora um ajuste muito mais preciso utilizando uma equação quadrática. Apesar de ser mínima, a forma quadrática dos dados exige um modelo também quadrático para que tenhamos um ajuste apropriado, seguindo as premissas básicas da modelagem com mínimos quadrados.



Transformações de Variáveis



2. Transformações de Variáveis

Quando os modelos não atendem as hipóteses de normalidade, linearidade, homocedasticidade, transformações de uma ou mais variáveis frequentemente tem bom resultado para melhorar ou corrigir a situação.

Quando o modelo viola as hipóteses de normalidade, a transformação tipicamente é realizada na variável resposta. Quando a hipótese de linearidade é violada, uma transformação nas variáveis explicativas pode ajudar.

Transformação na Variável Resposta

Transformação de potência (Y^λ)

As transformações típicas envolvem substituir a variável resposta Y por Y^λ . Valores típicos de λ e sua interpretação são dados no Quadro 1. Se Y é uma proporção, uma transformação *logit* [$\ln(Y/1 - Y)$] é frequentemente utilizada.

Quadro 1: Transformações típicas de variáveis e valores típicos de λ (KABACOFF (2015))

λ	-2	-1	-0.5	0	0.5	1	2
Transformação	$1/Y^2$	$1/Y$	$1/\sqrt{Y}$	$\log(Y)$	\sqrt{Y}	Nenhuma	Y^2

Fonte: Elaborado pelo próprio autor

Transformação Box-Cox

Uma transformação típica utilizada para acertos de normalidade da variável resposta é a Box-Cox, que tem este nome devido aos sobrenomes de seus proponentes (G. E. P. Box e D. R. Cox, respectivamente da University of Wisconsin e do Birkbeck College, University of London); esta transformação foi proposta por eles no artigo intitulado “An Analysis of Transformations” (BOX; COX (1964)).

A transformação de Box-Cox é dada pela expressão mostrada na Equação (2):

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{se } \lambda \neq 0; \\ \log y, & \text{se } \lambda = 0 \end{cases} \quad (2)$$

onde λ é um parâmetro a ser estimado dos dados; \log é o logaritmo neperiano.

Uma vez obtido o valor de λ , encontramos os valores dos dados transformados conforme a equação acima e utilizamos estes dados transformados para efetuar as análises.

Exemplo de Transformação na Variável Resposta

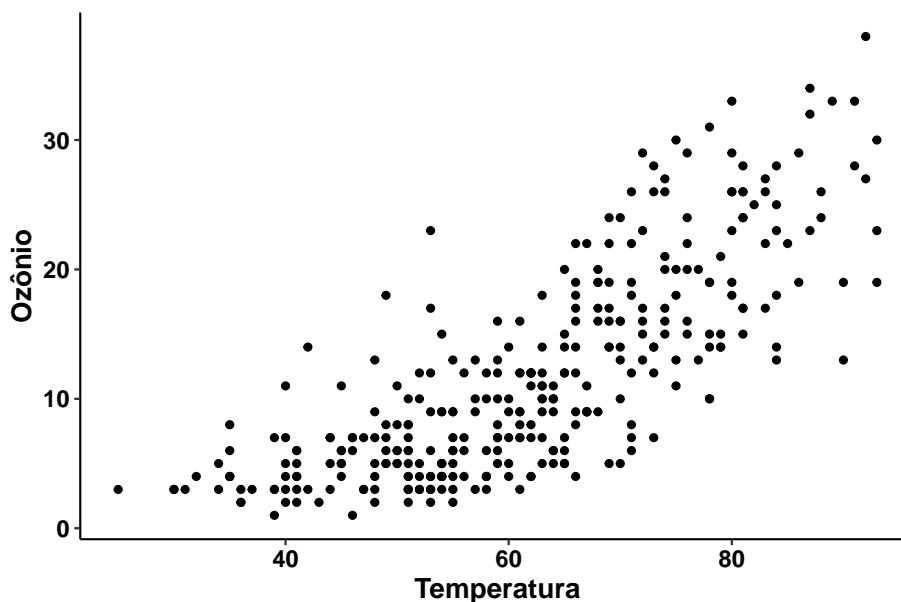
Para ilustrar este tipo de transformação vamos utilizar uma base de dados sobre medidas de ozônio que faz parte da biblioteca *faraway* do **R**. Caso não esteja instalada, a biblioteca *faraway* deve ser instalada com o comando:

```
> install.packages("faraway")
```

O conjunto de dados ozone tem 330 observações e 10 variáveis, entre elas a medida de ozônio e de temperatura. Para nosso propósito, utilizaremos apenas as variáveis ozônio e temperatura.

```
> library(faraway)
> ozdata <- faraway::ozone[,c("O3", "temp")]
> names(ozdata) <- c("ozonio", "temperatura")
> head(ozdata)
  ozonio temperatura
1      3          40
2      5          45
```

Figura 10: Níveis de Ozônio como função da Temperatura



Fonte: Elaborado pelo autor

3	5	54
4	6	35
5	4	45
6	4	55

Após a leitura dos dados, vamos ver a relação entre as duas variáveis através de um gráfico de dispersão, como mostrado na Figura 10.

```
> library(ggplot2)
> library(ggpubr)
> library(ggfortify)
> g <- ggplot(data = ozdata, aes(x = temperatura, y = ozonio)) +
...   geom_point() + xlab("Temperatura") + ylab("Ozônio") +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
> g
```

Observe que o gráfico de dispersão mostrado na Figura 10, mostra uma forte relação crescente não linear entre as medidas de ozônio e temperatura. Também observamos que as medidas de ozônio apresentam aumento de variabilidade para valores crescentes de temperatura.

Diante disso, podemos levantar dúvidas se o ajuste do modelo de regressão linear simples com as variáveis na sua forma original é adequado neste caso. Vamos fazer este ajuste para evidenciar sua inadequação através da análise de resíduos.

```
> oz1 <- lm(ozonio ~ temperatura, ozdata)
```

```
> summary(oz1)
```

Call:

```
lm(formula = ozonio ~ temperatura, data = ozdata)
```

Residuals:

Min	1Q	Median	3Q	Max
-10.9939	-3.8202	-0.1796	3.1951	15.0112

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-14.93745	1.21247	-12.32	<2e-16 ***
temperatura	0.43257	0.01912	22.63	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.014 on 328 degrees of freedom

Multiple R-squared: 0.6095, Adjusted R-squared: 0.6083

F-statistic: 511.9 on 1 and 328 DF, p-value: < 2.2e-16

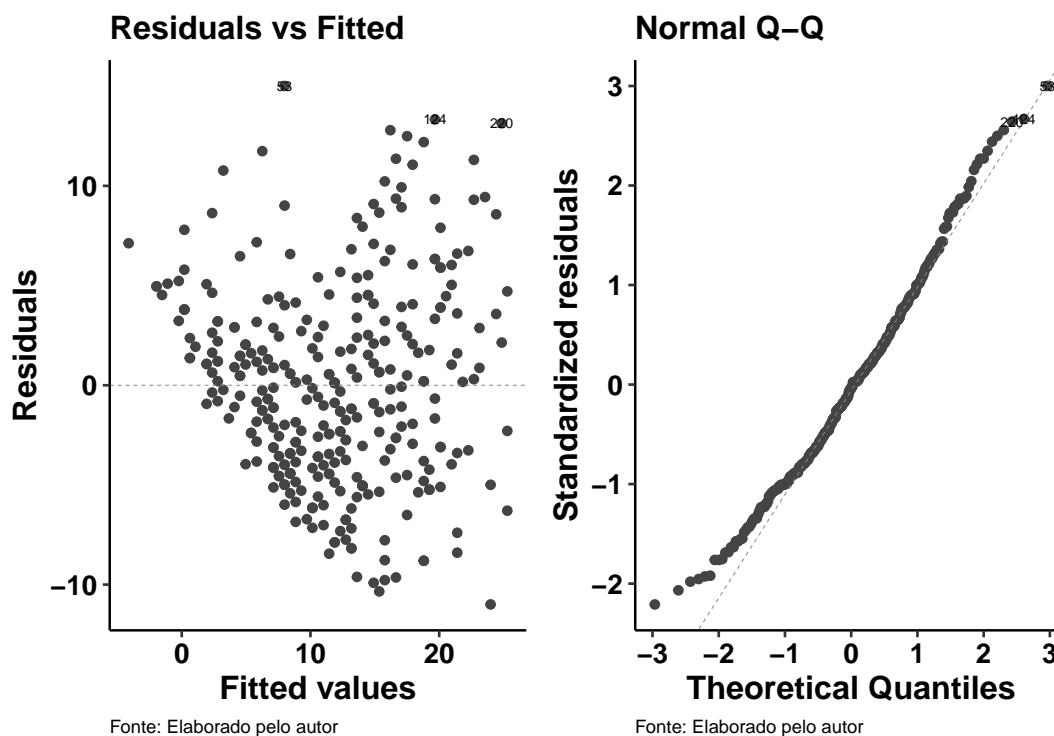
Teste de normalidade dos Resíduos

Uma das premissas estatísticas para a utilização do método dos mínimos quadrados ordinários (OLS) é que a variável resposta Y tenha uma distribuição normal em torno da média. O teste de Shapiro-Wilk utiliza o princípio da *hipótese nula* para verificar se uma amostra vem de uma população com distribuição normal. A hipótese nula deste teste é que a população é normalmente distribuída. Assim, se o p-value é **menor** do que o nível do alfa escolhido, a hipótese nula é **rejeitada** e há evidência de que os dados testados não são de uma população com distribuição normal.

```
> shapiro.test(residuals(oz1))
```

```
Shapiro-Wilk normality test
```

Figura 11: Gráficos Diagnósticos da Regressão



```
data: residuals(oz1)
W = 0.9856, p-value = 0.002235
```

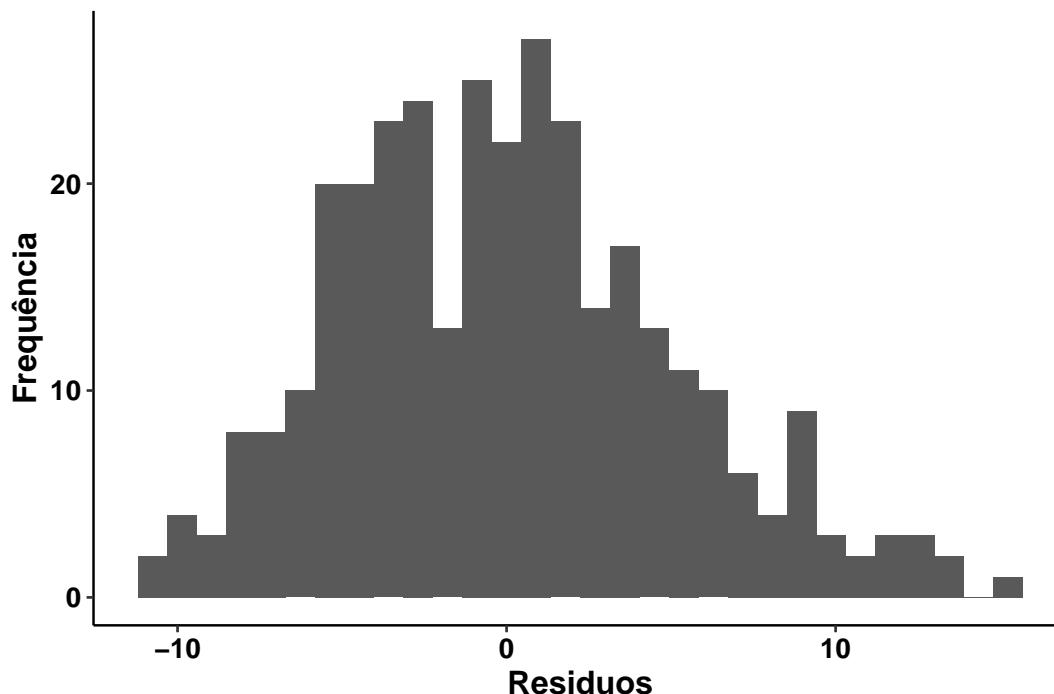
Observamos, pelo Teste de normalidade de Shapiro-Wilk, que a hipótese nula deve ser rejeitada, já que obtivemos um p-value de 0.0022.

Gráficos Diagnósticos

Continuamos nosso diagnóstico do modelo ajustado agora com a inspeção dos gráficos diagnósticos. Vamos inspecionar os seguintes gráficos: resíduos × valores ajustados, gráfico dos quantis teóricos × quantis dos resíduos (QQ-Plot), conforme mostrado na Figura 11.

```
> autoplot(oz1, which = 1:2, ncol = 2, label.size = 2, smooth.linetype = 0) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
```

Figura 12: Histograma dos resíduos



Fonte: Elaborado pelo autor

Observamos nos gráficos da Figura 11 que a variância dos erros não é constante (gráfico Resíduos x Valores Ajustados); há um aumento na variabilidade dos resíduos com o aumento do valor da variável resposta Y ; no gráfico QQ-plot observamos que há um desvio da normalidade (gráfico QQ-Plot), suposição que também é confirmada pelo Teste de normalidade de Shapiro-Wilk, cujo p-valor = 0.0022.

Outra premissa do método dos mínimos quadrados ordinários (OLS) é que os resíduos tenham uma distribuição normal. Vamos então fazer um histograma dos resíduos, conforme mostrado na Figura 12. Nesta figura observamos que a distribuição aparenta ser bimodal.

```
> df.hist <- data.frame(Resíduos = residuals(oz1), Ajustados = fitted(oz1))
> gh <- ggplot(data = df.hist, aes(x = Resíduos)) +
...   geom_histogram() + theme_pubr() +
...   labs_pubr() + ylab("Frequência") +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
> gh
```

A fim de solucionar os problemas de variância não-constante deve-se tentar realizar uma transfor-

mação na variável resposta Y .

Apesar de ser possível, em muitos casos, selecionar empiricamente a transformação adequada do tipo Y^λ , vamos utilizar a técnica da *Transformação Box-Cox*.

Utilizamos a função `boxcox` do pacote MASS para a determinação do parâmetro λ . A função `boxcox` é muito fácil de utilizar; especificamos a fórmula do modelo e, normalmente, as opções padrões cuidam de todo o resto.

O gráfico padrão da função `boxcox` tem como limites (-2,2). Como estamos interessados no ponto de máximo, fazemos um novo gráfico com um *zoom* na região de interesse, como mostrado na Figura 13:

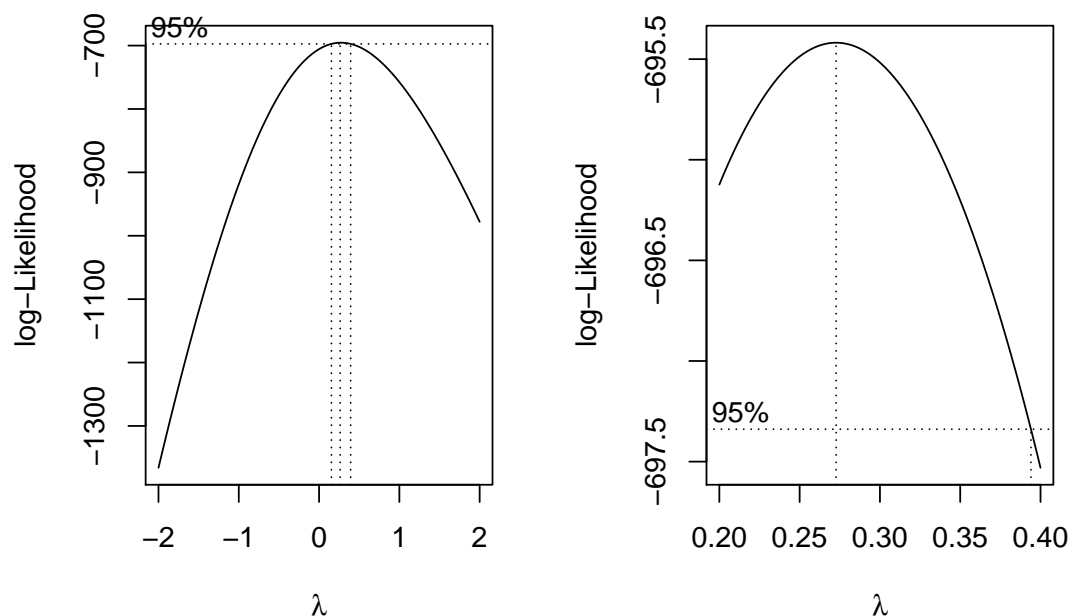
```
> require(MASS)
> library(dplyr)
> set.seed(123456)
> options(digits = 7)
> par(mfrow = c(1,2))
> boxcox(oz1, eps = 0.001)
> mtext("Fonte: Elaborado pelo autor", xpd = NA, cex = 0.7,
...      side = 1, line = 3.8, adj=-1)
> boxcox(oz1, lambda=seq(0.2, 0.4, by=0.01), eps = 0.001)

> par(mfrow = c(1,1))
```

Pelo gráfico verificamos que o máximo da verossimilhança foi atingido com aproximadamente $\lambda = 0,27$, com intervalo de confiança de 95% igual a $[0,15;0,39]$. Como esse intervalo não inclui o valor 1, há forte evidência da necessidade de transformação na variável resposta ozônio.

Para extrairmos o valor calculado de lambda, chamamos a função `boxcox` mas agora atribuindo o resultado a uma variável (objeto). O retorno da função é uma lista do vetor lambda e do perfil do vetor *log-likelihood* calculados; estes vetores são invisíveis quando os resultados são plotados. Estamos interessados no valor de lambda no máximo do *log-likelihood*.

```
> bx <- boxcox(oz1, lambda=seq(0.2, 0.4, by=0.01), plotit = FALSE, eps = 0.001)
> bx.df <- data.frame(x = bx$x, y = bx$y)
> bx2.df <- bx.df[with(bx.df, order(-bx.df$y)),]
> bx2.df[1,]
      x      y
8 0.27 -695.4193
```

Figura 13: Gráfico da função boxcox, para determinar o valor ótimo de lambda

Fonte: Elaborado pelo autor

```
> round(bx2.df[1, "x"], 4)
[1] 0.27
```

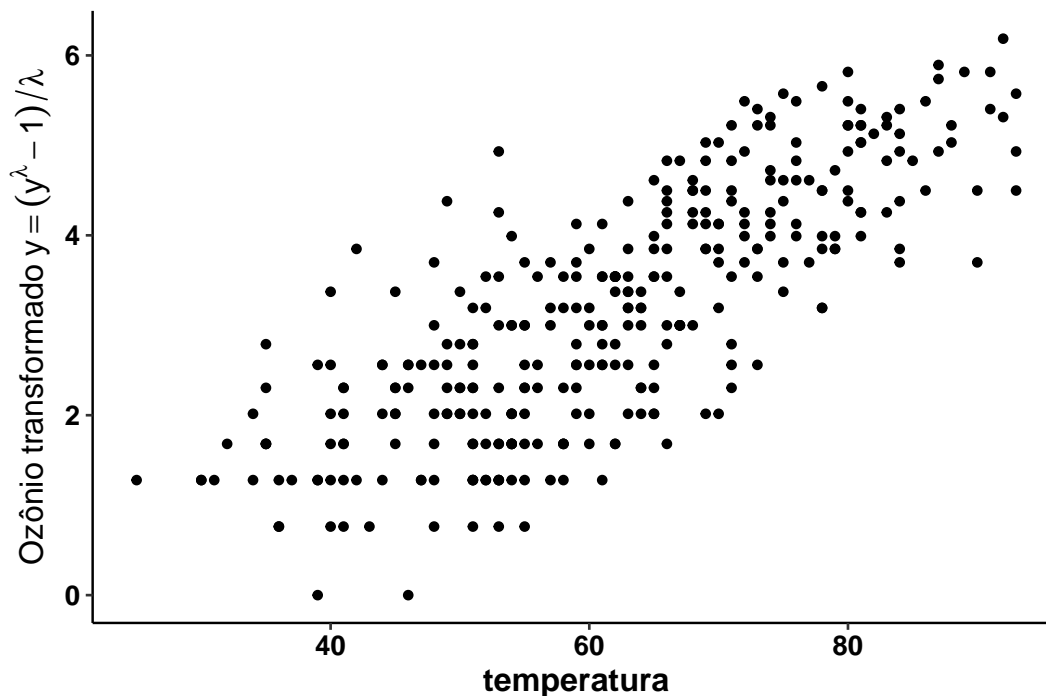
O valor calculado de λ pelo método da função boxcox é 0.27.

Assim, a transformação dos dados será dada por: $ozonio = (ozonio^{0,27} - 1)/0,27$.

- Se o intervalo de confiança contivesse o 0, provavelmente a transformação logarítmica dos dados poderia ser utilizada com bons resultados.

Sendo assim a nova variável transformada ozoniotrans deve ser inserida no nosso conjunto de dados, para que o novo modelo de regressão linear simples seja ajustado.

```
> lmbd <- round(bx2.df[1, "x"], 3)
> ozdatatrans <- mutate(ozdata, ozoniotrans = (ozonio^lmbd - 1)/lmbd)
> head(ozdatatrans)
  ozonio temperatura ozoniotrans
1      3          40    1.278930
2      5          45    2.015797
3      5          54    2.015797
4      6          35    2.304395
```


Figura 14: Níveis de Ozônio (após transformação) em função da temperatura

Fonte: Elaborado pelo autor

5	4	45	1.681380
6	4	55	1.681380

Com a variável resposta transformada, fazemos novamente o gráfico de dispersão, como mostrado na Figura 14:

```
> g <- ggplot(ozdatatrans, aes(x = temperatura, y = ozoniotrans)) +
...   geom_point() + ylab(expression(Ozônio~transformado~y==(y^lambda -1)/lambda)) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
> g
```

O gráfico mostrado na Figura 14 indica uma forte relação linear crescente entre as medidas de ozônio transformadas, via método de Box Cox, versus temperatura, com *variabilidade aproximadamente constante*, que era o nosso objetivo.

Podemos então ajustar novamente o modelo linear, agora utilizando a variável transformada:

```
> oz2 <- lm(ozoniotrans ~ temperatura, ozdatatrans)

> summary(oz2)

Call:
lm(formula = ozoniotrans ~ temperatura, data = ozdatatrans)

Residuals:
    Min       1Q   Median       3Q      Max
-1.99712 -0.56569  0.07148  0.56078  2.41671

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.408685    0.199642  -7.056 1.02e-11 ***
temperatura  0.074039    0.003148  23.520 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8256 on 328 degrees of freedom
Multiple R-squared:  0.6278,    Adjusted R-squared:  0.6266
F-statistic: 553.2 on 1 and 328 DF,  p-value: < 2.2e-16
```

Examinamos agora os gráficos diagnósticos do nosso modelo com a variável transformada, como mostrado na Figura 15:

```
> autoplot(oz2, which = 1:2, ncol = 2, label.size = 3, smooth.linetype = 0) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
```

Examinamos também o histograma dos resíduos (lembrando que devem seguir uma distribuição normal), como mostrado na Figura 16:

```
> df.hist <- data.frame(Residuos = residuals(oz2), Ajustados = fitted(oz2))
> gh <- ggplot(data = df.hist, aes(x = Residuos)) + geom_histogram() +
...   theme_pubr() + labs_pubr() + ylab("Frequência") +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
> gh
```

Figura 15: Gráficos Diagnósticos após transformação BoxCox

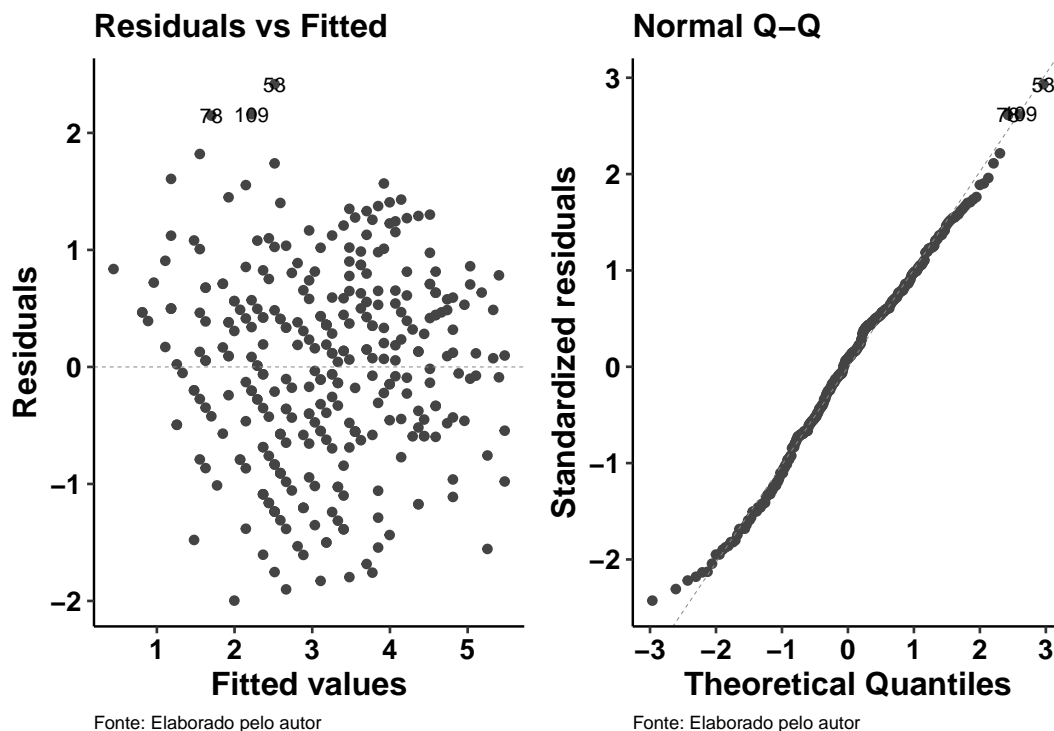
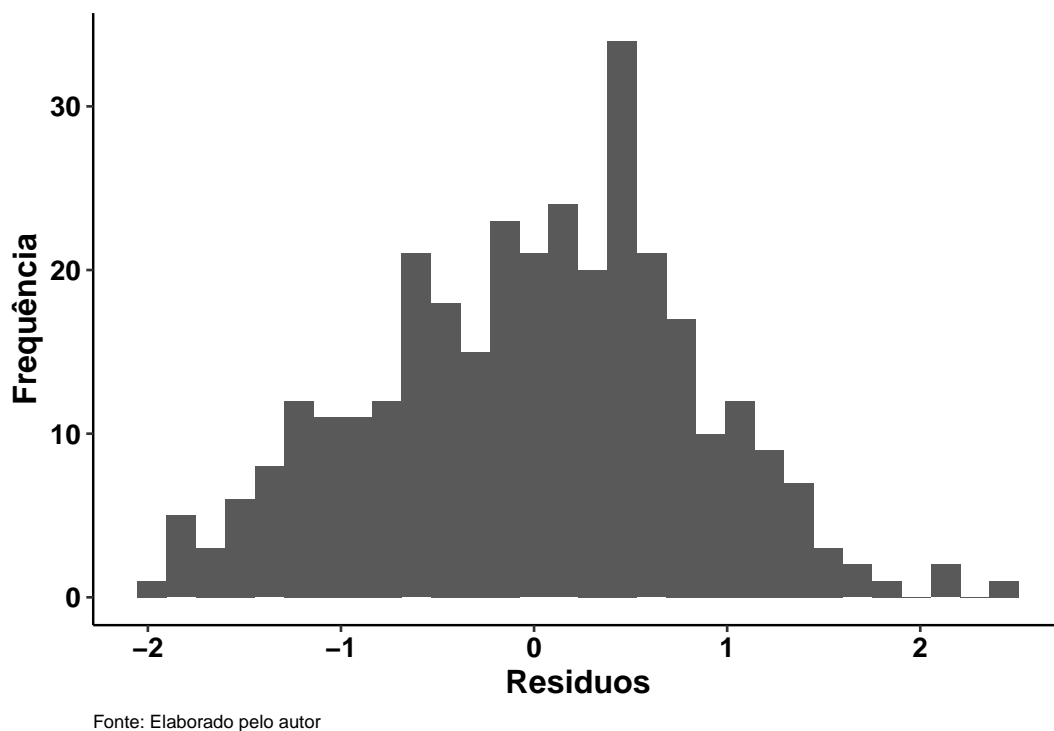


Figura 16: Histograma dos Resíduos (variável níveis de Ozônio transformada)



O novo ajuste (oz2), cuja equação da reta ajustada é dada por

$$\hat{Y} = -1,41 + 0,074X_i$$

tem um $R^2 - ajustado = 0,63$.

Pelos gráficos diagnósticos mostrados na Figura 15, observa-se que a suposição de normalidade é aceitável, bem como a homocedasticidade dos erros.

Realizamos o teste de Shapiro-Wilk para verificarmos o resultado da transformação Box-Cox.

```
> shapiro.test(residuals(oz2))
```

```
Shapiro-Wilk normality test
```

```
data: residuals(oz2)
```

```
W = 0.99325, p-value = 0.1456
```

Obtemos agora um p-value que cai na região de aceitação da hipótese nula, indicando que a transformação surtiu o efeito desejado.

Transformação na Variável Explicativa

Além de transformações na variável dependente (resposta), também é possível realizar transformações na variável explicativa; tais transformações são geralmente necessárias quando os aspectos de linearidades não estão sendo atendidos e não se quer introduzir termos não lineares de variáveis explicativas. O objetivo é a obtenção de um modelo estatístico mais adequado para a descrição dos dados. Vamos estudar um pequeno conjunto de dados neste exemplo.

Exemplo de Transformação na Variável Explicativa

Os dados fictícios que utilizaremos tratam do estudo que o gerente de Recursos Humanos de uma loja realizou para estimar o efeito do número de dias de treinamento (X) no desempenho em um teste simulado de vendas (Y) aplicado em seus vendedores. Os dados estão na Tabela 1.

Ao invés de fazermos a leitura dos dados de um arquivo, vamos colocá-los diretamente em um data.frame no R chamado treino_venda; os nomes das variáveis devem ser Tempo e Desempenho, respectivamente

Tabela 1: Tempo de Treinamento vs Desempenho no teste

Tempo de Treinamento	Desempenho
0.5	42.5
0.5	50.6
1	68.5
1	80.7
1.5	89
1.5	99.6
2	105.3
2	111.8
2.5	112.3
2.5	125.7

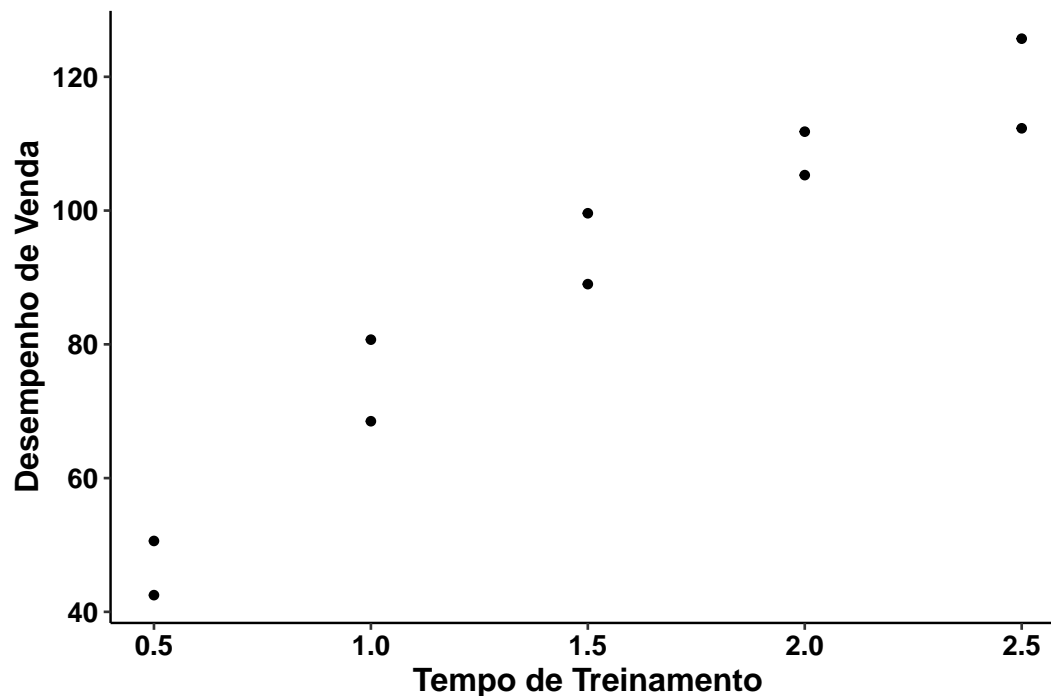
Fonte: Elaborado pelo autor

```
> treino_venda <- data.frame(Tempo=c(0.5,0.5,1,1,1.5,1.5,2,2,2.5,2.5),
...                           Desempenho=c(42.5,50.6,68.5,80.7,89,99.6,105.3,111.8,112.3,125.7))
> head(treino_venda)
  Tempo Desempenho
1  0.5      42.5
2  0.5      50.6
3  1.0      68.5
4  1.0      80.7
5  1.5      89.0
6  1.5      99.6
```

Vamos fazer uma primeira inspeção visual dos dados, através de um gráfico de dispersão, mostrado na Figura 17.

```
> g <- ggplot(treino_venda, aes(x=Tempo, y = Desempenho)) +
...   geom_point() + xlab("Tempo de Treinamento") +
...   theme_pubr() + ylab("Desempenho de Venda") + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
> g
```

O gráfico de dispersão da Figura 17 mostra uma relação levemente curvilínea entre X e Y, com variabilidade aproximadamente constante nos níveis de X. Desse modo, vamos tentar realizar uma transformação apenas em X. A escolha da transformação adequada aqui é feita apenas de forma empírica.

Figura 17: Desempenho de Venda vs Tempo de Treinamento

Fonte: Elaborado pelo autor

Baseando-se em padrões já conhecidos, vamos escolher a função raiz quadrada, ou seja, $X' = \sqrt{X}$. Entretanto, para fins comparativos, antes de realizar a regressão linear simples com a variável transformada X' , realizamos a regressão com a variável original X .

```
> ajuste1 <- lm(Desempenho ~ Tempo, data=treino_venda)
```

```
> summary(ajuste1)
```

Call:

```
lm(formula = Desempenho ~ Tempo, data = treino_venda)
```

Residuals:

Min	1Q	Median	3Q	Max
-12.0700	-2.2262	-0.3925	4.3187	11.0000

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	34.945	5.948	5.875	0.000372 ***
Tempo	35.770	3.587	9.973	8.66e-06 ***

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 8.02 on 8 degrees of freedom
```

```
Multiple R-squared:  0.9256,    Adjusted R-squared:  0.9163
```

```
F-statistic: 99.46 on 1 and 8 DF,  p-value: 8.66e-06
```

Iniciamos o diagnóstico pelo teste de Shapiro-Wilk.

```
> shapiro.test(residuals(ajuste1))
```

```
Shapiro-Wilk normality test
```

```
data:  residuals(ajuste1)
```

```
W = 0.94359, p-value = 0.5936
```

Pelo teste de Shapiro-Wilk, obtemos um p-value = 0.5936, que está na região de aceitação da hipótese nula, ou seja, resíduos com distribuição normal. Mas vamos prosseguir...

Continuamos nosso diagnóstico do modelo ajustado agora com a inspeção dos gráficos diagnósticos, como mostrado na Figura 18.

```
> autoplot(ajuste1, which = 1:2, ncol = 2, label.size = 3, smooth.linetype = 0) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
```

O histograma dos resíduos, mostrados na Figura 19.

```
> df.aj1 <- data.frame(residuos = residuals(ajuste1), ajustados = fitted(ajuste1))
> g <- ggplot(data = df.aj1, aes(x = residuos)) +
...   ylab("Frequência") + geom_histogram(binwidth = 4) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
> g
```

A análise dos resíduos na Figura 18 indica uma distribuição com tendência (parábola), que é diferente de um padrão aleatório que seria esperado, apesar do resultado do Teste de normalidade de Shapiro-Wilk, cujo P-valor é 0.5936 indicar que devemos aceitar a hipótese nula.

Figura 18: Gráficos Diagnósticos do Modelo Linear

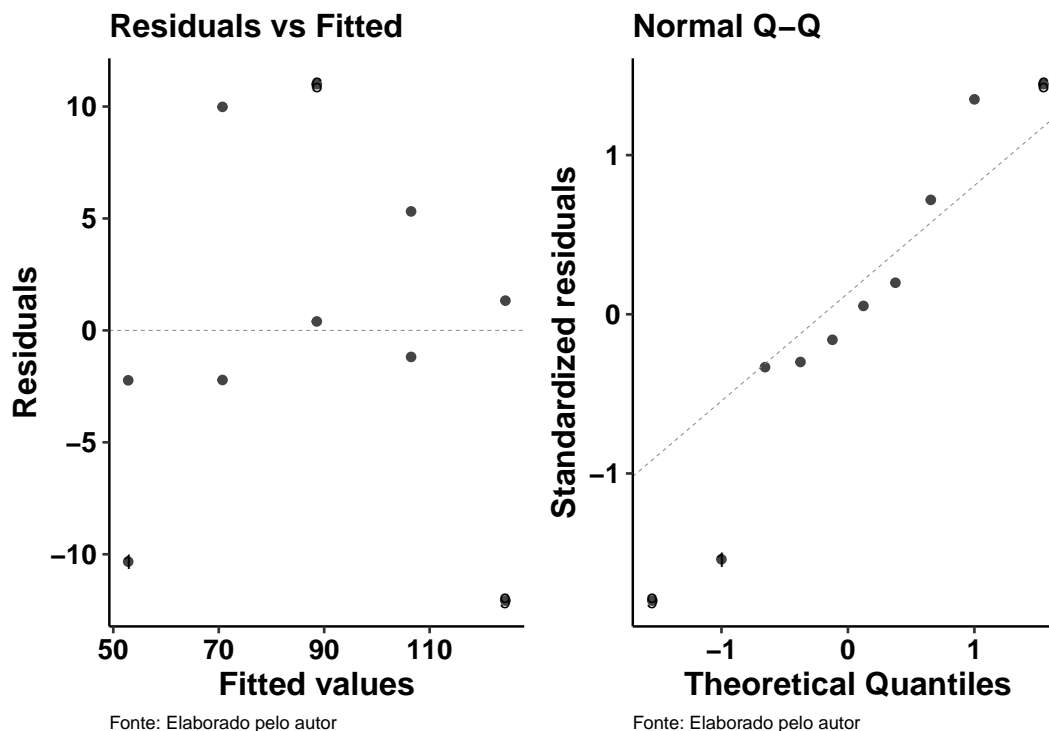


Figura 19: Histograma dos Resíduos do Modelo Linear

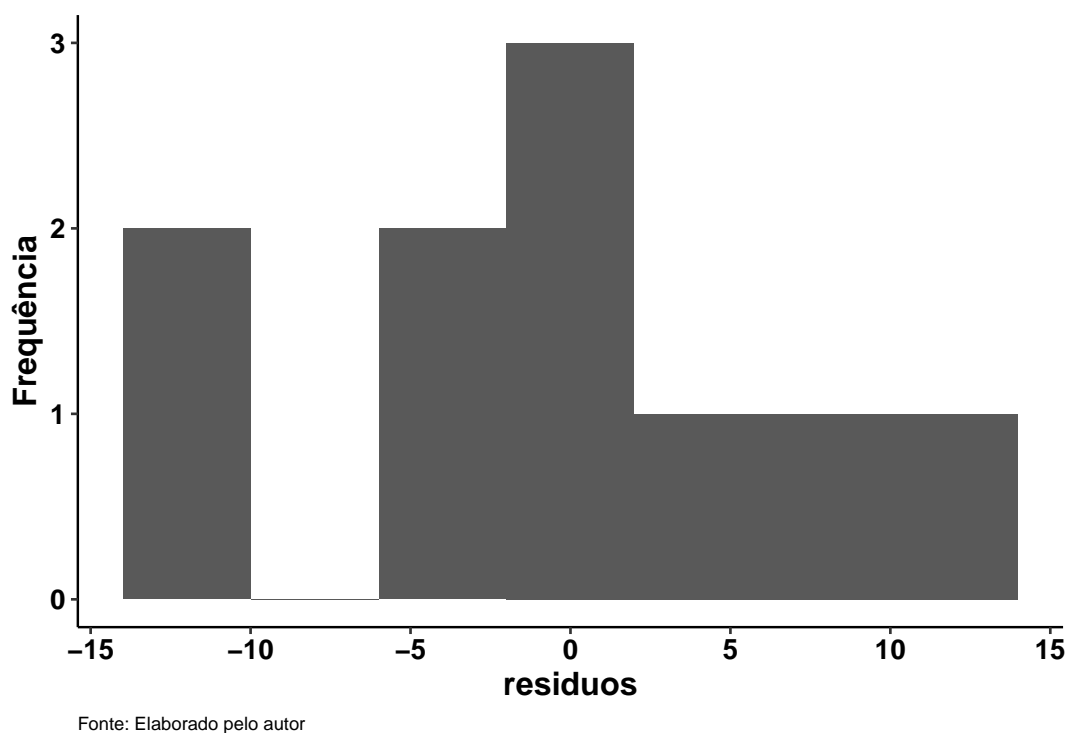
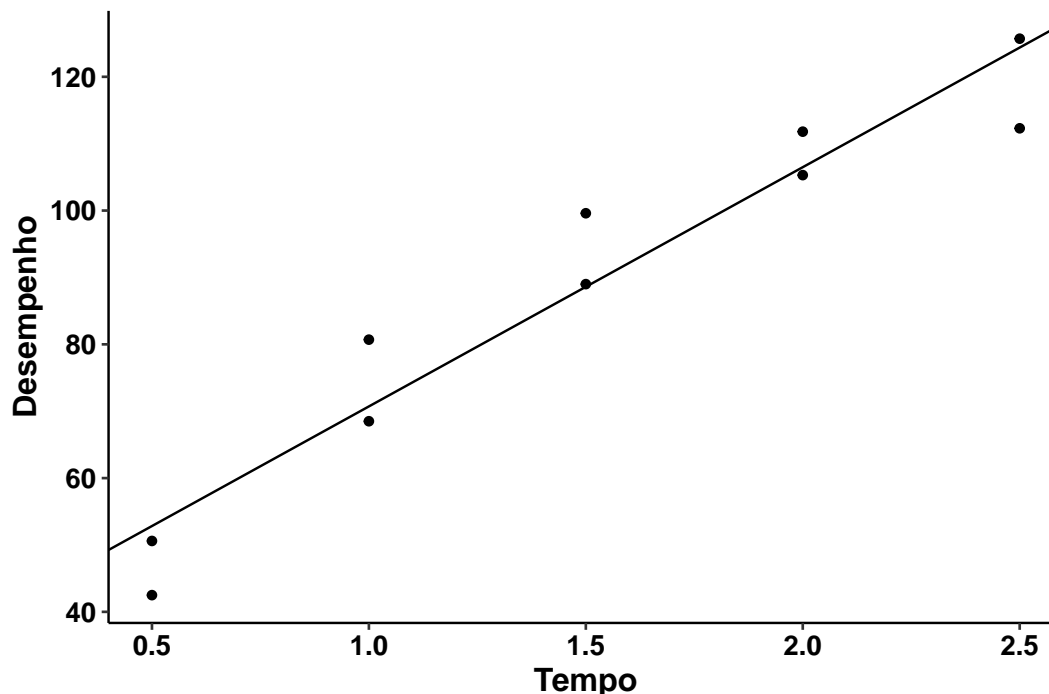


Figura 20: Modelo Linear nos dados de Desempenho de Venda vs Tempo de Treinamento

Fonte: Elaborado pelo autor

Para melhor visualizarmos o problema que enfrentamos neste caso, vamos plotar também a linha de regressão do nosso modelo, como mostrado na Figura 20.

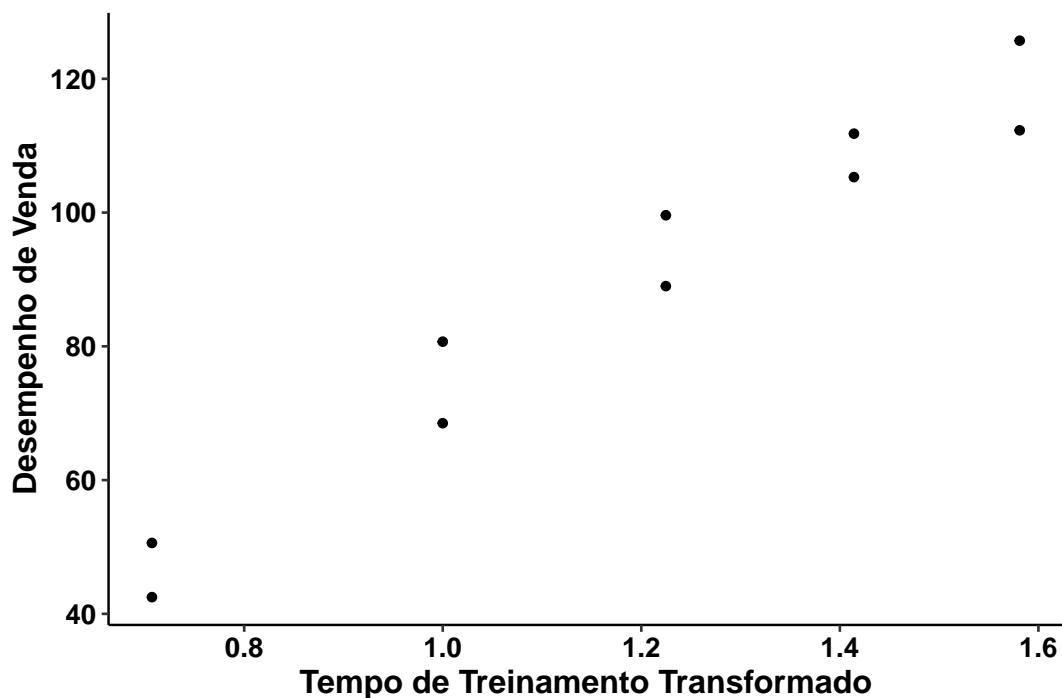
```
> it <- coef(ajuste1)[1]
> sl <- coef(ajuste1)[2]
> ggplot(data = treino_venda, aes(x = Tempo, y = Desempenho)) +
...   geom_point() + geom_abline(slope = sl, intercept = it) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))
```

Observamos no gráfico mostrado na Figura 20 que a reta de regressão não acompanha bem todos os pontos, evidenciando o aspecto curvilíneo da relação entre X e Y.

A fim de linearizar o modelo acima, sem modificar as condições de normalidade, vamos utilizar a transformação da variável explicativa pela função raiz quadrada, mostrada a seguir:

```
> treino_venda_trans = mutate(treino_venda, Tempotrans = sqrt(Tempo))
```

Figura 21: Desempenho de Venda vs Tempo de Treinamento - Transformado



Fonte: Elaborado pelo autor

Podemos então visualizar o gráfico dos dados transformados mostrado na Figura 21

```
> ggplot(data = treino_venda_trans, aes(x = Tempotrans, y = Desempenho)) +
...   geom_point() + theme_pubr() + labs_pubr() +
...   xlab("Tempo de Treinamento Transformado") + ylab("Desempenho de Venda") +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
```

Vamos então ajustar um novo modelo aos dados transformados.

```
> ajuste2 <- lm(Desempenho ~ Tempotrans, data=treino_venda_trans)
> summary(ajuste2)
```

Call:

```
lm(formula = Desempenho ~ Tempotrans, data = treino_venda_trans)
```

Residuals:

Min	1Q	Median	3Q	Max
-9.3221	-4.1884	-0.2367	4.1007	7.7200

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -10.328      7.892  -1.309   0.227
Tempotrans    83.453      6.444  12.951 1.2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.272 on 8 degrees of freedom
Multiple R-squared:  0.9545,    Adjusted R-squared:  0.9488
F-statistic: 167.7 on 1 and 8 DF,  p-value: 1.197e-06

```

Repetimos o teste de normalidade dos resíduos, para ver se nossa transformação não *bagunçou* com o que estava *OK*.

```

> shapiro.test(residuals(ajuste2))

      Shapiro-Wilk normality test

data:  residuals(ajuste2)
W = 0.94032, p-value = 0.5566

```

Continuamos com uma distribuição normal para os resíduos, com p-value de 0.5566.

Vamos então fazer uma inspeção visual no modelo plotando os dados e o modelo, como mostra a Figura 22.

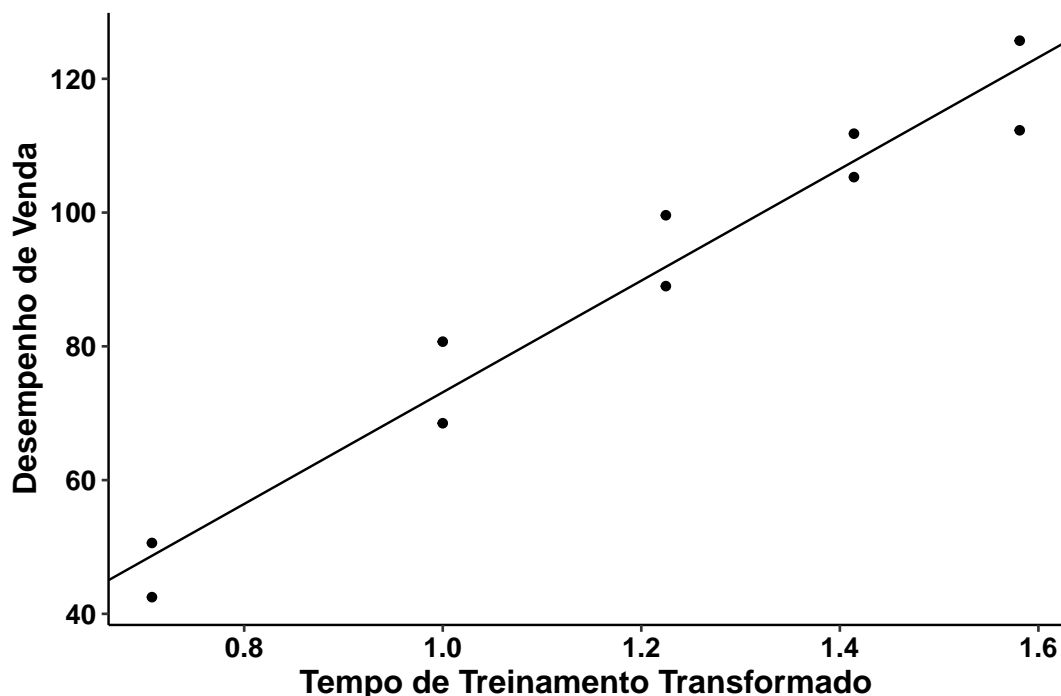
```

> it <- coef(ajuste2)[1]
> sl <- coef(ajuste2)[2]
> ggplot(data = treino_venda_trans, aes(x = Tempotrans, y = Desempenho)) +
...   geom_point() + geom_abline(slope = sl, intercept = it) +
...   xlab("Tempo de Treinamento Transformado") + theme_pubr() +
...   labs_pubr() + ylab("Desempenho de Venda") +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size = 8))

```

Vamos também analisar o modelo através dos gráficos diagnósticos, mostrados na Figura 23, resíduos vs valores ajustados e QQ-Plot.

Figura 22: Curva de Regressão do Modelo Linear com Variável Transformada



Fonte: Elaborado pelo autor

```
> autoplot(ajuste2, which = 1:2, ncol = 2, label.size = 3, smooth.linetype = 0) +
...   theme_pubr() + labs_pubr() +
...   labs(caption = "Fonte: Elaborado pelo autor") + labs_pubr() +
...   theme(plot.caption = element_text(hjust = 0, size= 8))
```

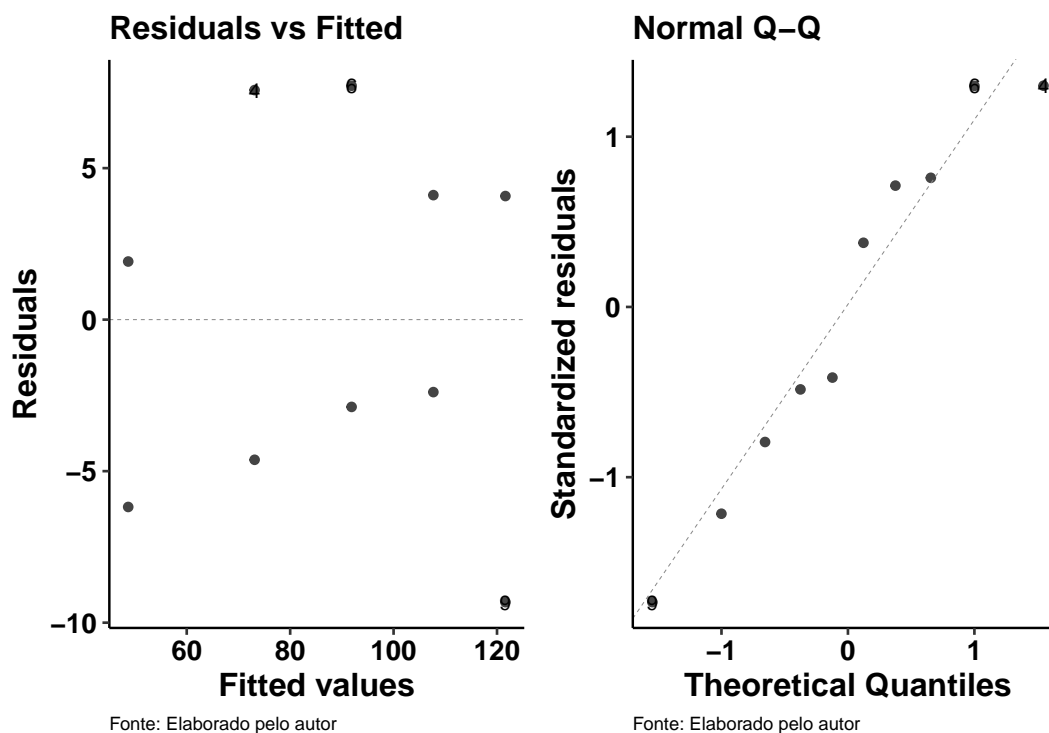
A equação da reta ajustada é dada por $\hat{Y} = -10.328 + 83.453X'_i$, com $R^2 - ajustado = 0,95$ (maior que o anterior). Observamos na Figura 22 que a reta de regressão agora acompanha bem todos os pontos, indicando que a linearidade entre X e Y foi alcançada. A análise dos resíduos mostrado nos gráficos da Figura 23 indica um bom ajuste do modelo, assim como o Teste de normalidade de Shapiro-Wilk, cujo P-valor é 0.5566.

Finalizando

Nesta Trilha abordamos a Regressão Linear Multivariada, Transformações de Variáveis e Técnicas de Seleção de Modelos (Material Complementar).

A Regressão Linear Multivariada é a técnica utilizada quando temos mais do que uma variável

Figura 23: Gráficos Diagnósticos do Modelo Linear com Variável Transformada



explicativa no nosso modelo. Vários aspectos relacionados à sua utilização foram abordados, incluindo a verificação de multicolinearidade, interação entre variáveis explicativas e as transformações de variáveis que eventualmente são necessárias. Uma restrição importante em nossa abordagem é que a variável resposta deve ser uma combinação *linear* das variáveis explicativas, embora estas possam aparecer como funções quadráticas, logarítmicas ou outras. Os seus **coeficientes** no entanto, devem ser lineares.

Quando nosso modelo não atende às hipóteses estatísticas subjacentes para a utilização do método dos mínimos quadrados, uma alternativa é realizar transformações de variáveis. Vimos nesta Trilha como transformar a variável resposta utilizando a abordagem de Box-Cox e também como transformar a variável explicativa com uma função simples (raiz quadrada). As transformações se mostraram eficientes ao restaurarem os aspectos necessários para que os modelos tivessem aderências às premissas estatísticas.

Por fim, no Material Complementar da Trilha abordamos alguns critérios numéricos objetivos para selecionarmos um *melhor* modelo entre modelos possíveis com as variáveis presentes nos dados. Os critérios abordados são construídos de modo a penalizar modelos com mais variáveis e mesmo poder de explicação. O R tem algumas funções que auxiliam na visualização dos testes para a seleção dos modelos, dentre as quais algumas foram abordadas nesta Trilha.



Bibliografia

BOX, G. E.; COX, D. R. An analysis of transformations. **Journal of the Royal Statistical Society: Series B (Methodological)**, v. 26, n. 2, p. 211–243, 1964.

HAIR JR, J. F. et al. **Multivariate Data Analysis**. 7th. ed. Harlow, Essex, UK: Pearson Education Ltd, 2014.

JOHNSON, R. A.; WICHERN, D. W. **Applied Multivariate Statistical Analysis**. 3rd. ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1992.

