Relatório

Marcus Nunes

25 de outubro de 2017

1 Relatório

Vamos analisar o conjunto de dados cars, presente na memória do R. A partir de agora, cada trecho deste arquivo `.Rmd` que inicie com "` vai ser chamado de `chunk`. Cada `chunk` é um trecho de código do `R` que será executado e cujo output irá diretamente para o pdf final.

```
cars
```

```
##    speed dist
## 1      4    2
## 2      4   10
## 3      7    4
## 4      7   22
## 5      8   16
## 6      9   10
## 7     10   18
## 8     10   26
## 9     10   34
## 10    11   17
## 11    11   28
## 12    12   14
## 13    12   20
## 14    12   24
## 15    12   28
## 16    13   26
## 17    13   34
## 18    13   34
## 19    13   46
## 20    14   26
## 21    14   36
## 22    14   60
## 23    14   80
## 24    15   20
## 25    15   26
## 26    15   54
## 27    16   32
## 28    16   40
## 29    17   32
## 30    17   40
## 31    17   50
## 32    18   42
## 33    18   56
## 34    18   76
## 35    18   84
## 36    19   36
## 37    19   46
## 38    19   68
```

```
## 39    20   32
## 40    20   48
## 41    20   52
## 42    20   56
## 43    20   64
## 44    22   66
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
```

Este é um conjunto de dados com 50 linhas. Estas linhas ocupam muito espaço na página. Então, em vez de exibi-lo inteiro, vamos calcular algumas estatísticas a seu respeito.

```
summary(cars)
```

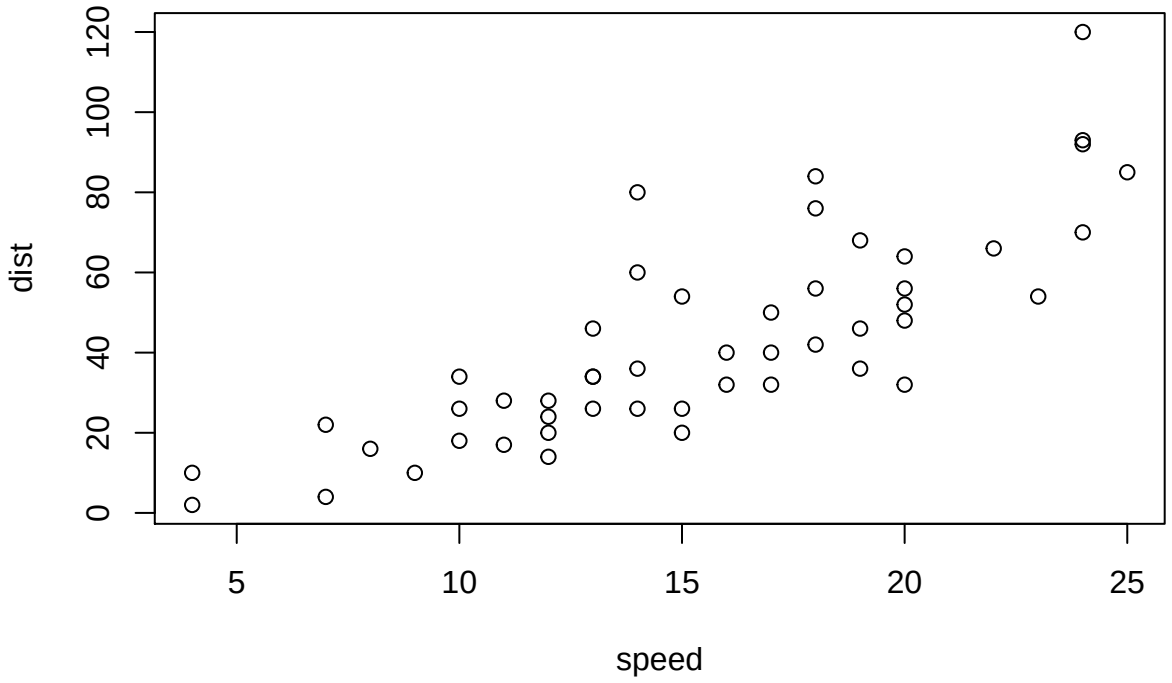
```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Interessante. Encontramos o *Resumo dos Cinco Números* do conjunto de dados `cars`. Aliás, perceba como aproveitei este pequeno parágrafo para demonstrar algumas ~~poçibilidades~~ possibilidades de formatação de texto utilizando R Markdown.

> É possível até mesmo escrever **citações** com a linguagem!
>
> *Nunes, M.A. (2017)*

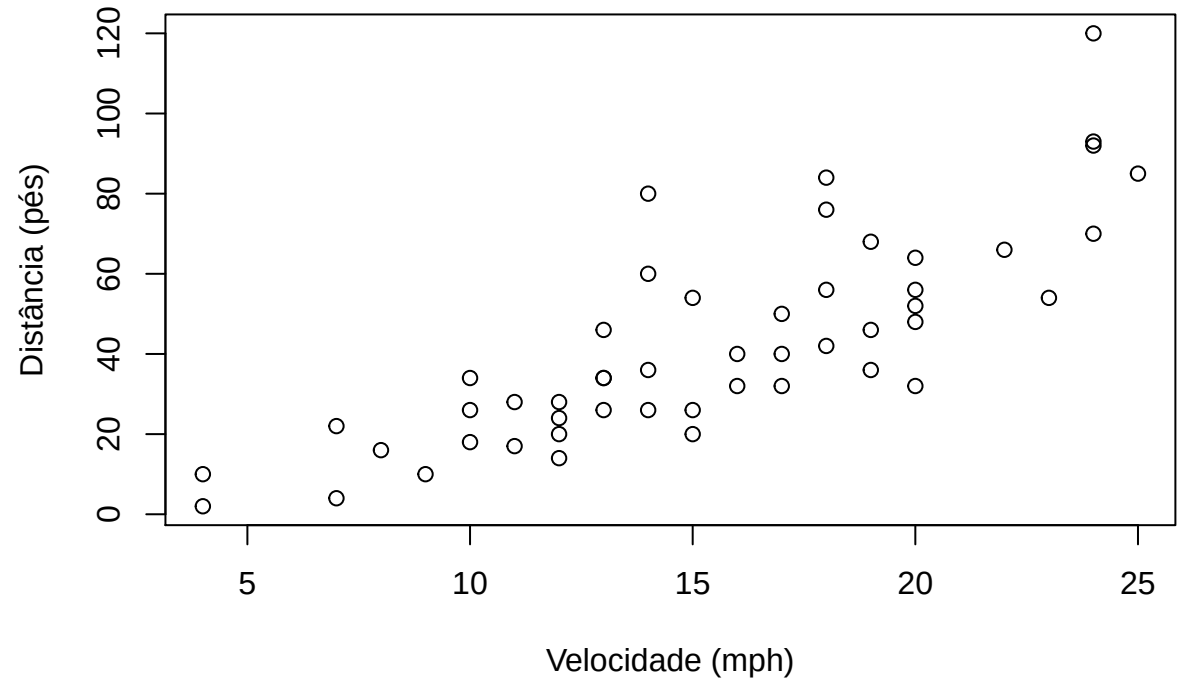
Mas vamos voltar à análise, pois é isto que importa aqui. Após nossas primeiras estatísticas descritivas a respeito destes dados, é interessante que façamos um gráfico relacionando distância e velocidade.

```
plot(dist ~ speed, data=cars)
```



Note que os chunks reproduzem exatamente aquilo que o código do R deve reproduzir. Portanto, precisamos identificar corretamente os eixos do gráfico:

```
plot(dist ~ speed, data=cars, xlab="Velocidade (mph)", ylab="Distância (pés)")
```



Note que este chunk exibe tanto o código quanto o resultado do gráfico. Se estivéssemos escrevendo um relatório para um cliente, a parte do código deveria ser eliminada. Felizmente, o `knitr` permite que coloquemos a opção `echo=FALSE` na definição do chunk e, assim, apenas o gráfico é produzido:

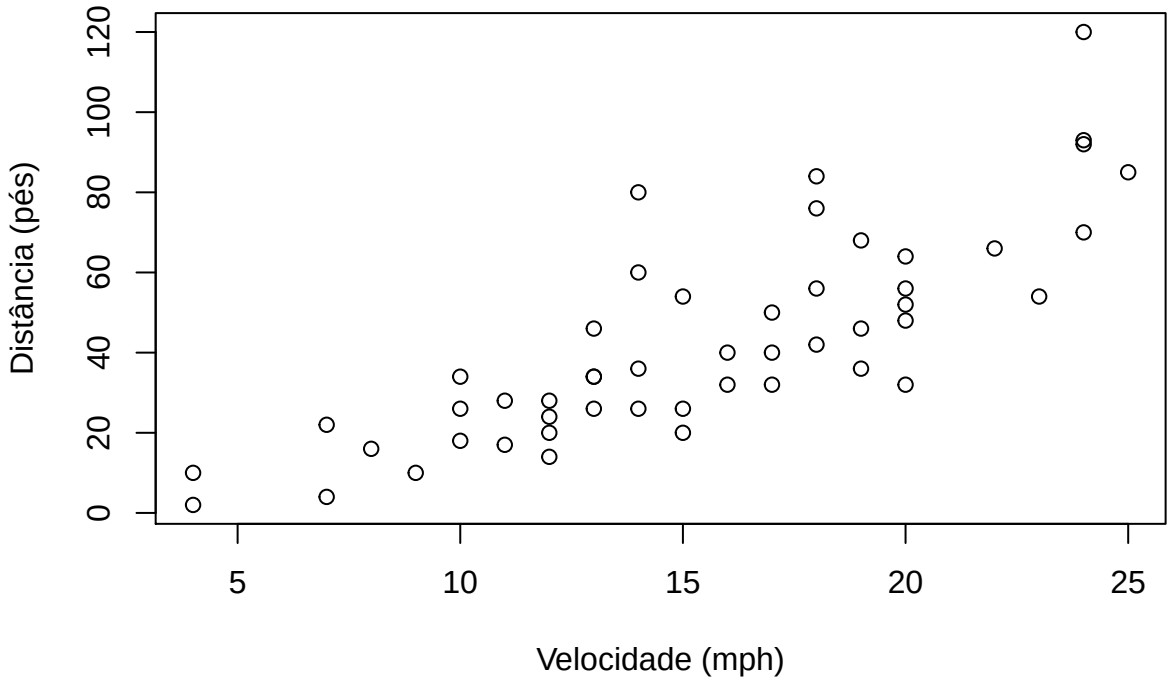
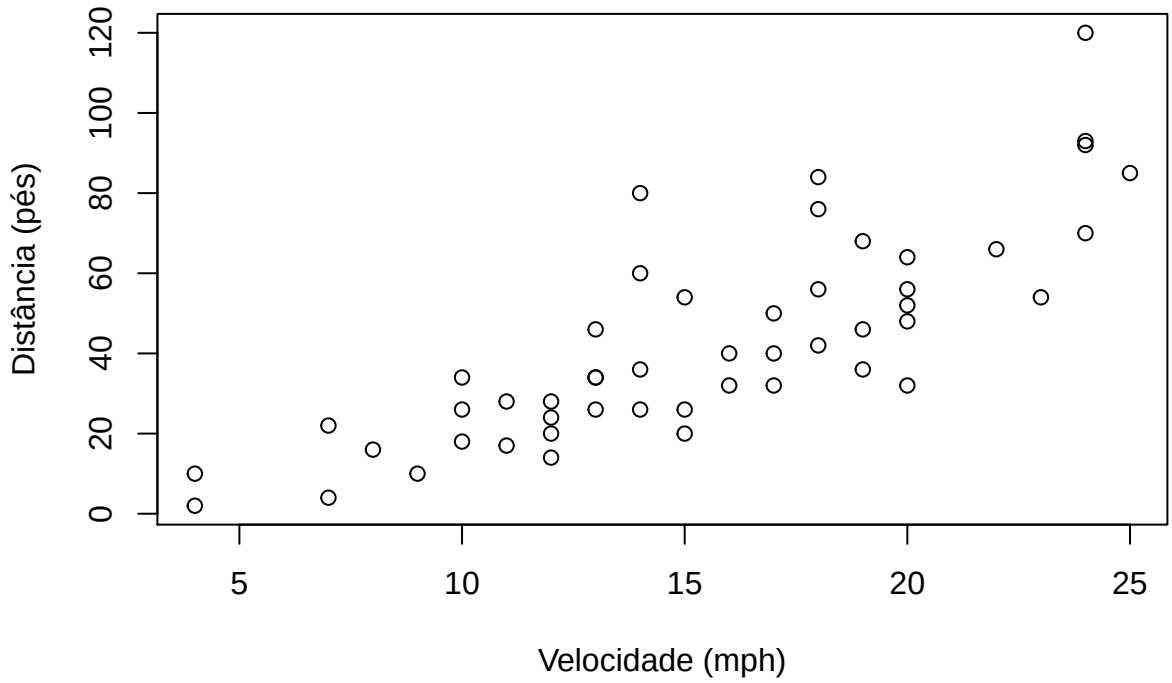


Figura 1: Gráfico de dispersão entre a distância necessária para frear completamente um carro (em pés) e sua velocidade (em milhas por hora).



Se colocarmos a opção `eval=FALSE`, apenas o código é exibido, sem que o gráfico seja plotado:

```
plot(dist ~ speed, data=cars, xlab="Velocidade (mph)", ylab="Distância (pés)")
```

Se quisermos, podemos colocar um label na Figura, para que ela possa ser referenciada posteriormente:

Veja como consigo referenciar a Figura 1 como se eu estivesse utilizando LaTeX. Eu posso inclusive fazê-la novamente em outra cor, para mostrar que a numeração é atualizada como no LaTeX tradicional:

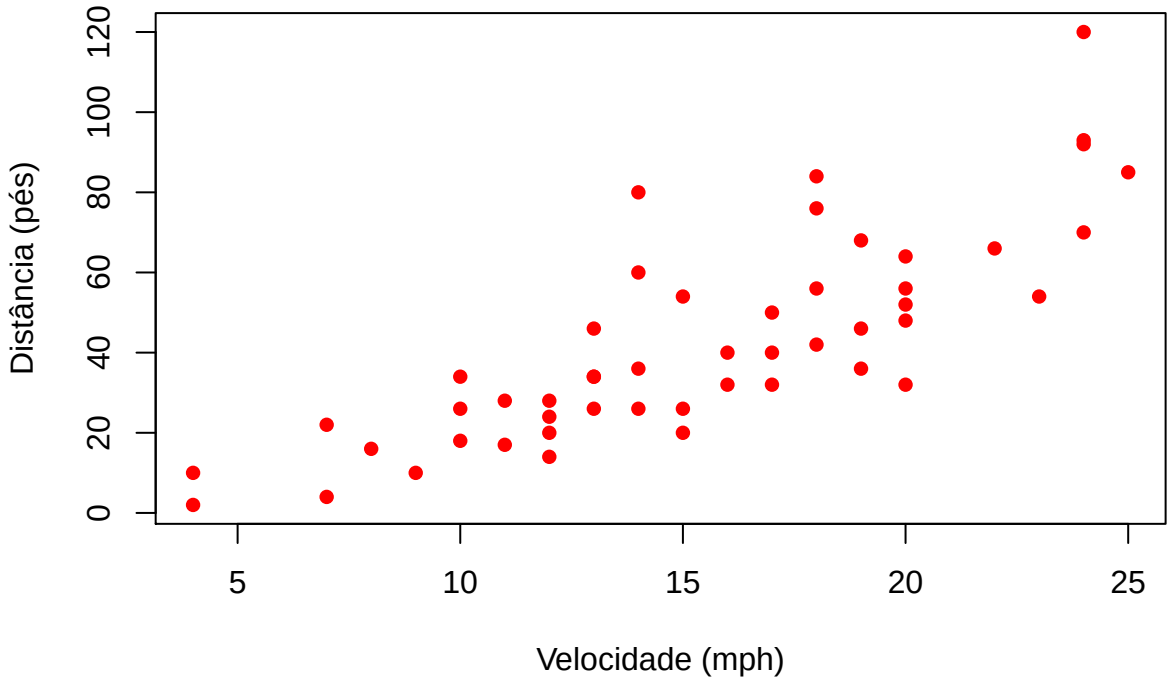


Figura 2: Gráfico de dispersão com os pontos em vermelho entre a distância necessária para frear completamente um carro (em pés) e sua velocidade (em milhas por hora).

Portanto, a Figura 2 mostra aquilo que foi prometido no parágrafo anterior. Note que estamos de fato utilizando LaTeX, pois a figura na página nnao ficou onde desejávamos!

(podemos consertar isto, mas este não é o escopo deste minicurso)

Além de gráficos, podemos ajustar modelos aos nossos dados. Vamos fazer uma regressão linear no conjunto de dados `cars`:

```
ajuste <- lm(dist ~ speed, data=cars)
summary(ajuste)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Este output seria interessante para nós, estatísticos. Mas e o usuário comum? Como proceder para que ele veja o resultado do ajuste que fizemos, sem utilizar o output padrão do `R`? Vamos usar o pacote `knitr` e a função `kable` para isto:

```
library(knitr)
kable(summary(ajuste)$coefficients, format="latex")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-17.579095	6.7584402	-2.601058	0.0123188
speed	3.932409	0.4155128	9.463990	0.0000000

Note que conseguimos fazer uma tabela, mas ela está feia. Esta tabela

1. tem linhas demais
2. está fora de centro
3. não tem título
4. tem muitos dígitos
5. as colunas não estão em português
6. mas olhe pelo lado bom: pelo menos aprendemos a fazer uma lista numerada no markdown

Podemos corrigir todos estes problemas de maneira razoavelmente simples. Em primeiro lugar, vamos escrever uma função que traduza o ajuste feito pela função `lm`:

```
lm_traducao <- function(ajuste){
  traducao <- summary(ajuste)$coefficients
  colnames(traducao) <- c("Estimativa", "Erro Padrão", "t", "p-valor")
  rownames(traducao) <- c("Intercepto", "Velocidade")
  return(traducao)
}

lm_traducao(ajuste)
```

```
##            Estimativa Erro Padrão         t      p-valor
## Intercepto -17.579095   6.7584402 -2.601058 1.231882e-02
## Velocidade   3.932409   0.4155128  9.463990 1.489836e-12
```

Pronto. Agora basta criarmos uma tabela da maneira que mais nos agrada:

```
library(kableExtra)
kable(lm_traducao(ajuste), format="latex", booktabs=TRUE,
      caption="\\label{CarsLM_4} Resultado do ajuste de um modelo linear
      aos dados do pacote \\texttt{cars}.", digits=4) %>%
  kable_styling(latex_options="hold_position")
```

Tabela 1: Resultado do ajuste de um modelo linear aos dados do pacote `cars`.

	Estimativa	Erro Padrão	t	p-valor
Intercepto	-17.5791	6.7584	-2.6011	0.0123
Velocidade	3.9324	0.4155	9.4640	0.0000

Portanto, a Tabela 1 ficou exatamente do jeito que imaginávamos. Agora podemos fazer o gráfico final da nossa análise, plotando os dados em conjunto com a reta ajustada. Este resultado se encontra na Figura 3.

```
plot(dist ~ speed, data=cars, xlab="Velocidade (mph)", ylab="Distância (pés)")
abline(ajuste)
```

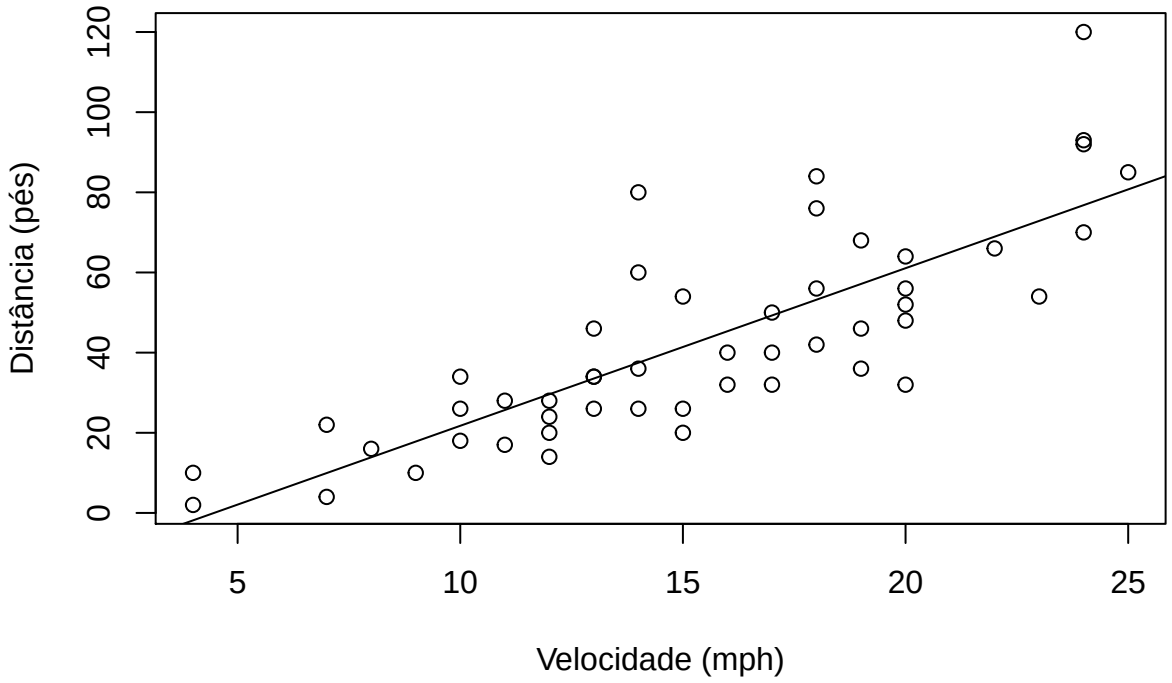


Figura 3: Resultado do ajuste linear entre a distância necessária para frear completamente um carro (em pés) e sua velocidade (em milhas por hora).

Caso a Figura 3 tenha ficado muito grande, podemos alterar as suas dimensões, como no caso da Figura 4.

```
plot(dist ~ speed, data=cars, xlab="Velocidade (mph)", ylab="Distância (pés)")
abline(ajuste)
```

Quem já está acostumado com LaTeX percebe que o comportamento das figuras e tabelas do `knitr` é idêntica às figuras e tabelas do LaTeX: a posição de cada novo float é uma surpresa.

Para que este relatório fique completo, só falta a análise dos resíduos do ajuste realizado. A Figura 5 exibe este resultado.

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

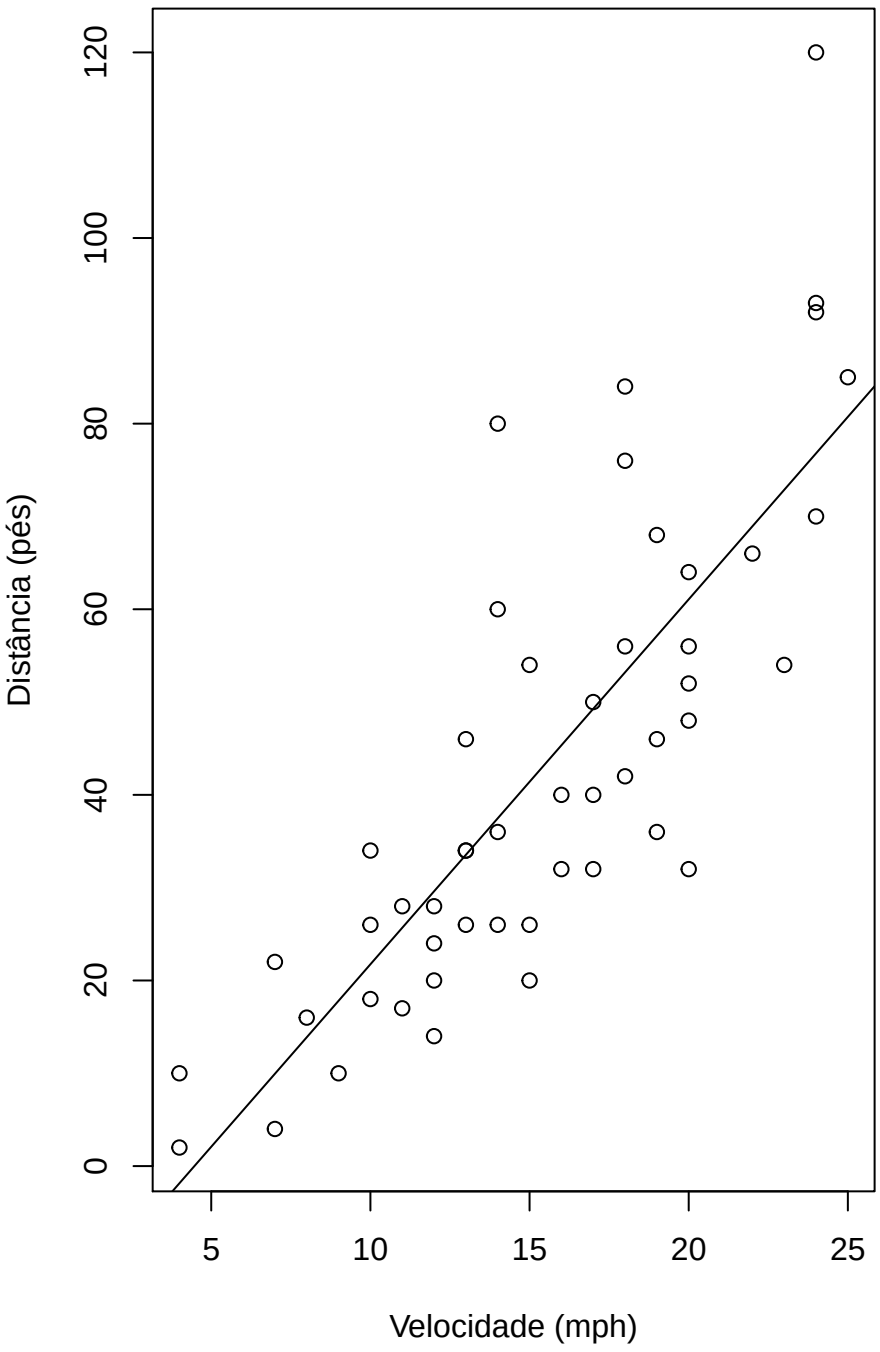


Figura 4: Resultado do ajuste linear entre a distância necessária para frear completamente um carro (em pés) e sua velocidade (em milhas por hora), com as dimensões alteradas.

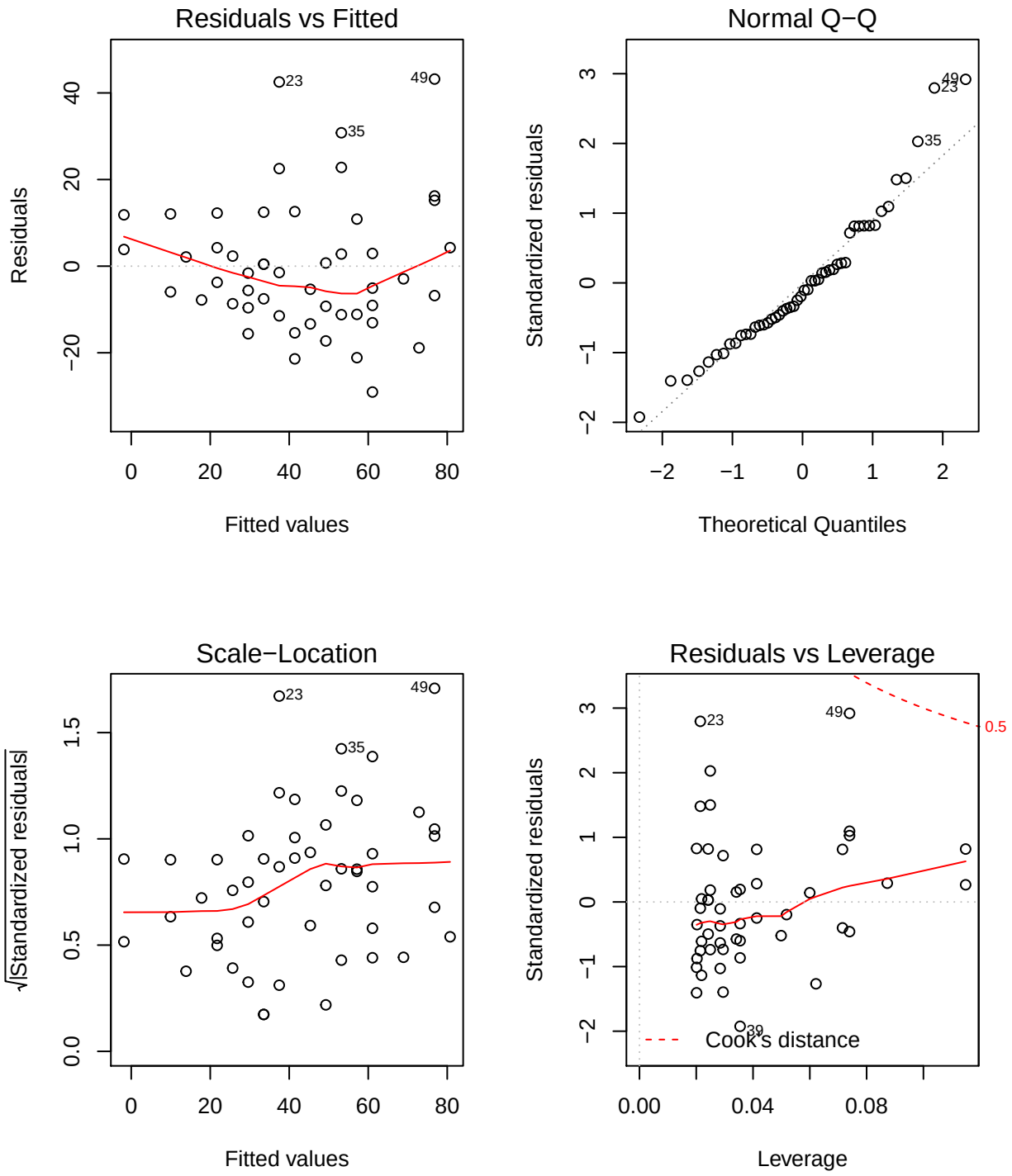


Figura 5: Gráficos de diagnóstico do ajuste linear entre a distância necessária para frear completamente um carro (em pés) e sua velocidade (em milhas por hora).

Note que, assim como no caso da tabela com o output do modelo de regressão, os gráficos de diagnóstico possuem os títulos e nomes dos eixos em inglês. Vou aproveitar os anos de experiência como instrutor de Matemática, Estatística e `R` e vou deixar registrada a seguinte maneira de lidar com este problema:

> A tradução destes gráficos fica como exercício para o leitor.
>
> *Todo autor de livro matemático quando está com preguiça de trabalhar.*

Com tudo pronto e aprendido, podemos passar agora à nossa prática.

2 Exercícios

Os exercícios de análise de dados a seguir foram baseados no conjunto de dados `iris`, presente na memória do `R`. Basta digitar 'iris no console para ter acesso ao seu conteúdo.

Este conjunto de dados possui 150 linhas e 5 colunas. Ele possui 4 medidas a respeito de três espécies de flores, chamadas *Iris setosa*, *Iris versicolor* e *Iris virginica*. As 4 medidas disponíveis para estas flores são

- Sepal.Length: comprimento da sépala
- Sepal.Width: largura da sépala
- Petal.Length: comprimento da pétala
- Petal.Width: largura da pétala

Todas estas 4 medidas estão disponíveis para as 3 espécies de flores.

A não ser que seja especificado de outra forma no enunciado, as respostas dos exercícios devem conter o código de resolução da questão e seu output desejado.

Caso a solução de alguma questão não seja encontrada neste arquivo de exemplo, utilize o Google para procurar uma maneira de resolvê-la.

1. Utilizando o menu do RStudio, crie um novo markdown, com output em pdf, chamado `exercicios.Rmd`. Altere o cabeçalho deste arquivo, de modo que ele fique similar ao do arquivo `relatorio.Rmd`.
2. Faça o resumo dos cinco números das varíaveis quantitativas do conjunto de dados. É possível perceber alguma assimetria olhando estes resultados?
3. Rode os comandos

```
library(GGally)
ggpairs(iris[, -5])
```

e interprete o resultado obtido.

4. Escolha as duas variáveis que se relacionam com maior intensidade e ajuste um modelo de regressão linear entre elas.
5. Exiba em uma tabela os resultados do modelo de regressão ajustado no passo anterior. Interprete o resultado obtido.
6. Crie um gráfico de dispersão relacionando estas duas variáveis. Pinte cada ponto com uma cor associada à sua espécie.
7. Faça um boxplot comparando as observações da variável `Sepal.Width` entre as três espécies de plantas.
8. Teste as hipóteses

$$H_0 : \mu_1 = \mu_2 = \mu_3$$
$$H_0 : \text{pelo menos um par } \mu_i \neq \mu_j, \text{ se } i \neq j$$

em que μ_i é a média da variável `Sepal.Width` para os grupos

- $i = 1$ (setosa)
- $i = 2$ (versicolor)
- $i = 3$ (virginica)

9. Interprete o resultado do comando anterior.
10. Utilize a função `TukeyHSD` para verificar quais diferenças entre as médias são diferentes de zero de maneira significante, com nível de confiança de 95%.