

---

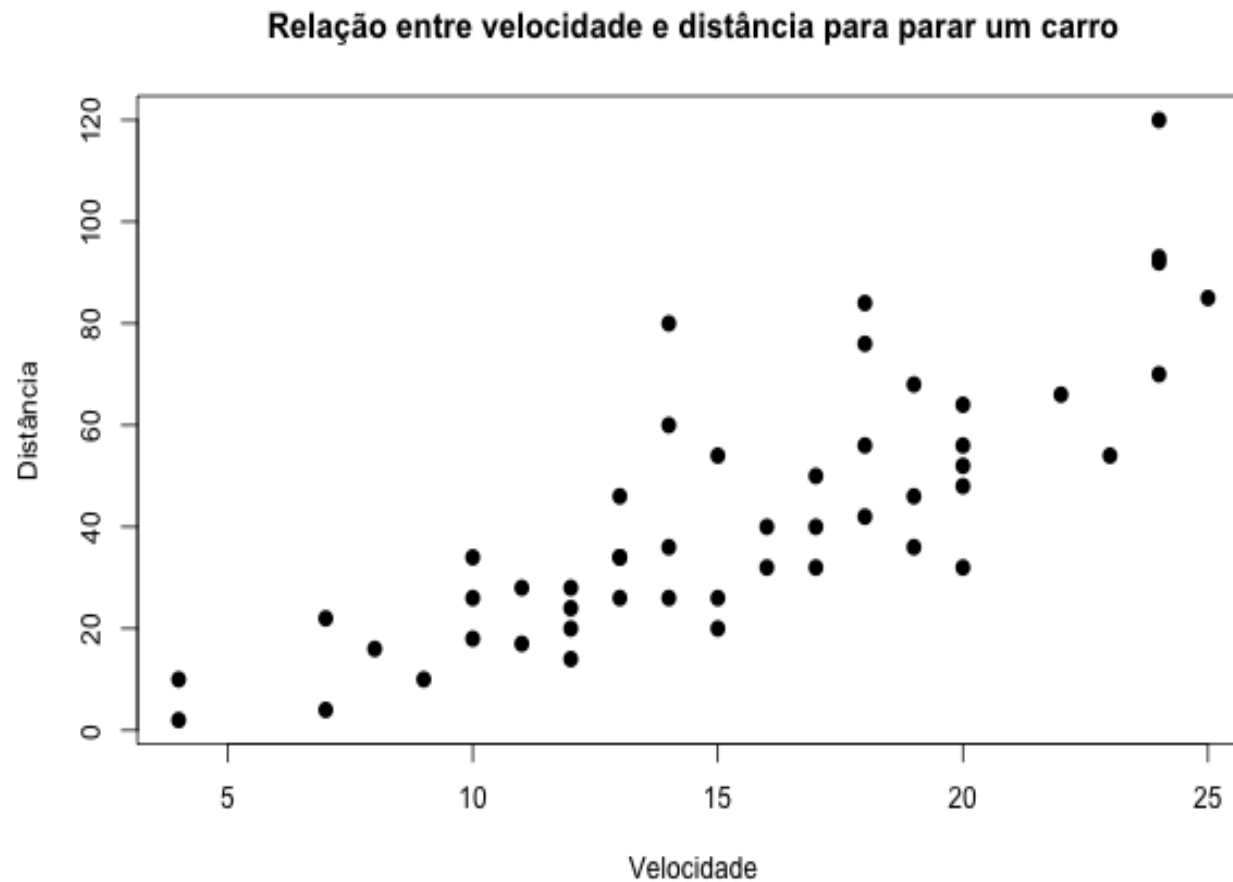
# Regressão Linear

Fabrício Barth

Maio de 2018

---

# Dados sobre carros



---

## Código para plotar o exemplo anterior

```
data(cars)
```

```
plot(cars$dist ~ cars$speed, pch=19, lwd=2,  
      xlab="Velocidade", ylab="Distância",  
      main="Relação entre velocidade e distância  
para parar um carro")
```

---

# Relações entre variáveis

- Será que existe relação entre a distância com que um carro consegue parar e a velocidade com que ele estava no momento da freada?
- Métodos de regressão tentam identificar se existe uma relação entre a variável dependente (o valor que precisa ser predito) e a variável independente.
- Distância = variável dependente
- Velocidade = variável independente

---

## Definindo linhas

- Uma linha pode ser definida na forma de
$$y = \alpha + \beta \times x$$
- onde  $x$  é a variável independente e  $y$  a variável dependente.
- $\beta$  indica quanto a linha cresce a cada incremento de  $x$ .
- A variável  $\alpha$  indica o valor de  $y$  quando  $x = 0$ .

---

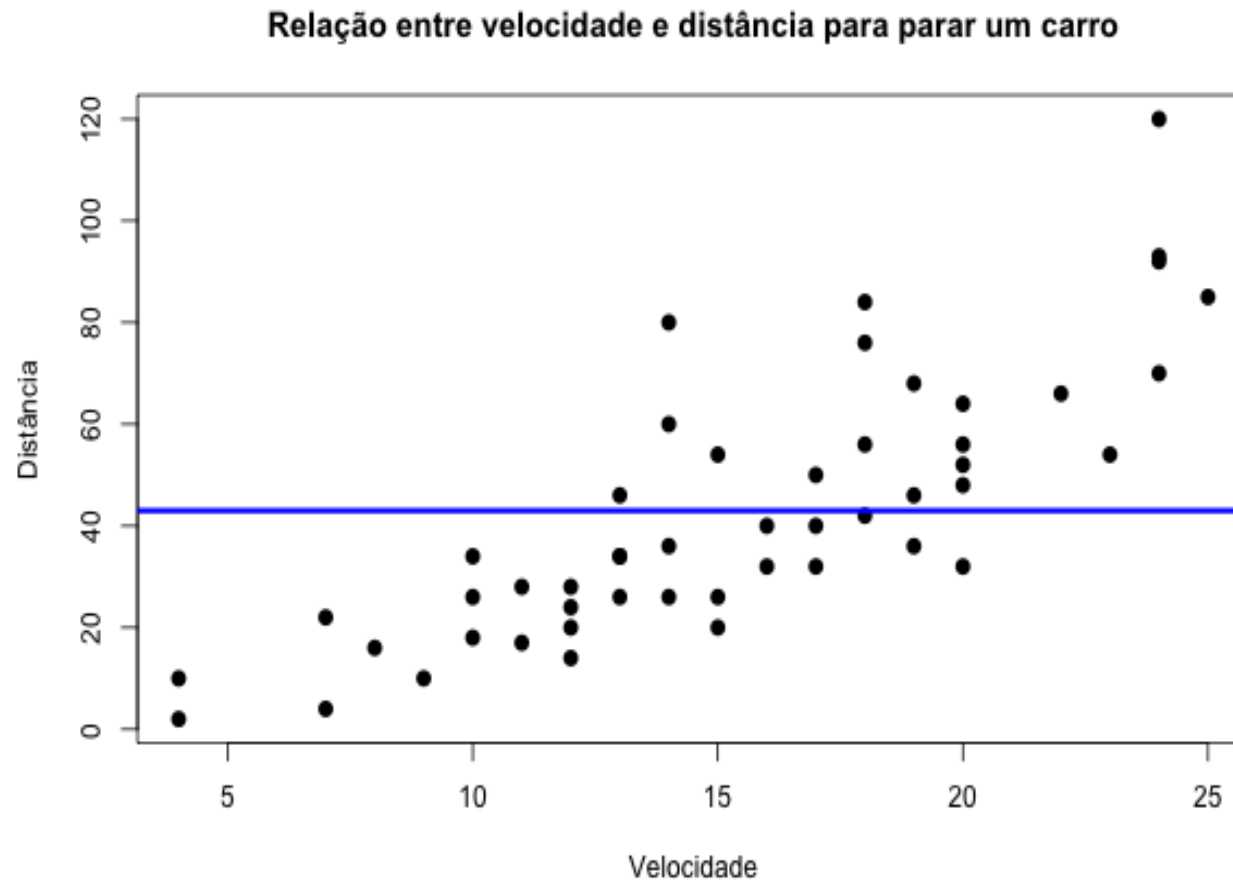
## Definindo modelos de regressão linear

- O objetivo de um algoritmo que cria este tipo de função é definir valores para  $\alpha$  e  $\beta$  de tal maneira que a linha consiga representar o conjunto de dados.
- Esta linha pode não representar o conjunto de dados perfeitamente. Portanto, é necessário calcular o erro de alguma forma.

---

$$distancia = 42.3 + 0 \times velocidade$$

É uma função válida. Mas é uma função boa?



Erro de  
 $distancia = 42.3 + 0 \times velocidade$





---

## Determinando o valor de $\alpha$ e $\beta$ em uma regressão linear simples

- Para estimar os melhores valores para  $\alpha$  e  $\beta$  é utilizado método chamado de **ordinary least squares (OLS)**.
- Com este método, os valores de  $\alpha$  e  $\beta$  são escolhidos para minimizar a soma dos erros ao quadrado, ou seja, a distância vertical entre o valor predito e o valor real.

$$erro = \sum (y_i - \hat{y}_i)^2 \quad (1)$$

onde,  $y_i$  é o valor real e  $\hat{y}_i$  é o valor predito.

# Uma função com um erro menor



---

## Código em *R* para o slide anterior

```
model <- lm(dist ~ speed, data=cars)
plot(cars$dist ~ cars$speed, pch=19, lwd=2,
      xlab="Velocidade", ylab="Distância",
      main="Relação entre velocidade e distância
           para parar um carro")
abline(model, col="red", lwd=3)
```

---

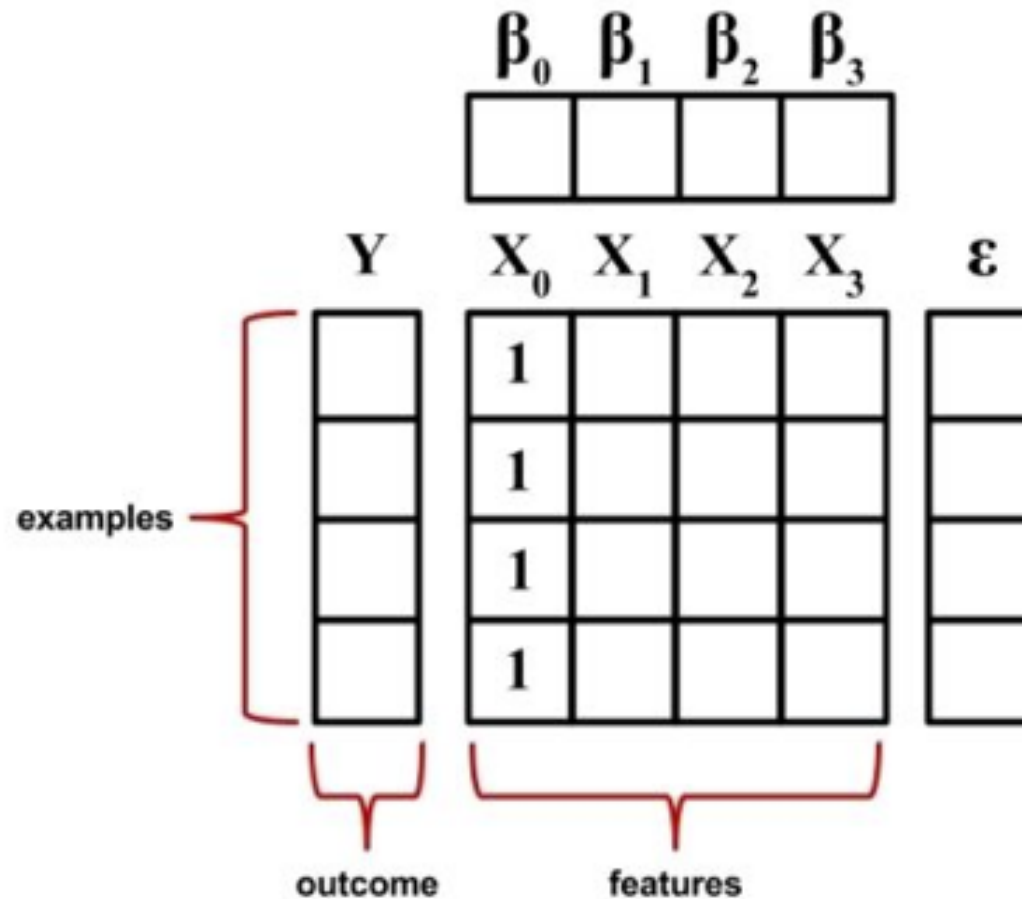
## Regressão linear múltipla

$$y = \alpha + \beta_1 \times x_1 + \beta_2 \times x_2 + \cdots + \beta_i \times x_i + e \quad (2)$$

Podemos utilizar uma equação compactada:

$$Y = X \times \beta + e \quad (3)$$

# Regressão linear múltipla



---

## Regressão linear múltipla

Agora o objetivo é resolver  $\hat{\beta}$ :

$$\beta = (X^T X)^{-1} X^T Y \quad (4)$$

onde:

- $X^T$  é matriz transposta de  $X$ , e;
- $X^{-1}$  a matriz inversa de  $X$ .

---

## Implementação em R

```
reg <- function(x,y){  
  x <- as.matrix(x)  
  x <- cbind(Intercept = 1, x)  
  solve(t(x) %*% x) %*% t(x) %*% y  
}
```

onde: `solve()` retorna a matriz inversa, `t()` calcula a matriz transposta e `%*%` multiplica duas matrizes.

---

## Encontrando os coeficientes para o problema do carro

```
reg(y = cars$dist, x = cars$speed)
```

deve retornar os mesmos valores de coeficientes que

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



---

## Exemplo de regressão linear simples usando *lm*

```
> model <- lm(dist ~ speed, data=cars)
> model
```

Call:

```
lm(formula = dist ~ speed, data = cars)
```

Coefficients:

(Intercept)	speed
-17.579	3.932

---

## Exemplo de regressão linear múltipla

```
data(airquality)  
help(airquality)  
head(airquality)
```

---

## Exemplo de regressão linear múltipla

```
> head(airquality)
```

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6

---

## Exemplo de regressão linear múltipla

```
> modelAirQuality <- lm(Ozone ~ Solar.R + Wind +  
  Temp, data=airquality)  
> modelAirQuality
```

Call:

```
lm(formula = Ozone ~ Solar.R + Wind + Temp,  
    data = airquality)
```

Coefficients:

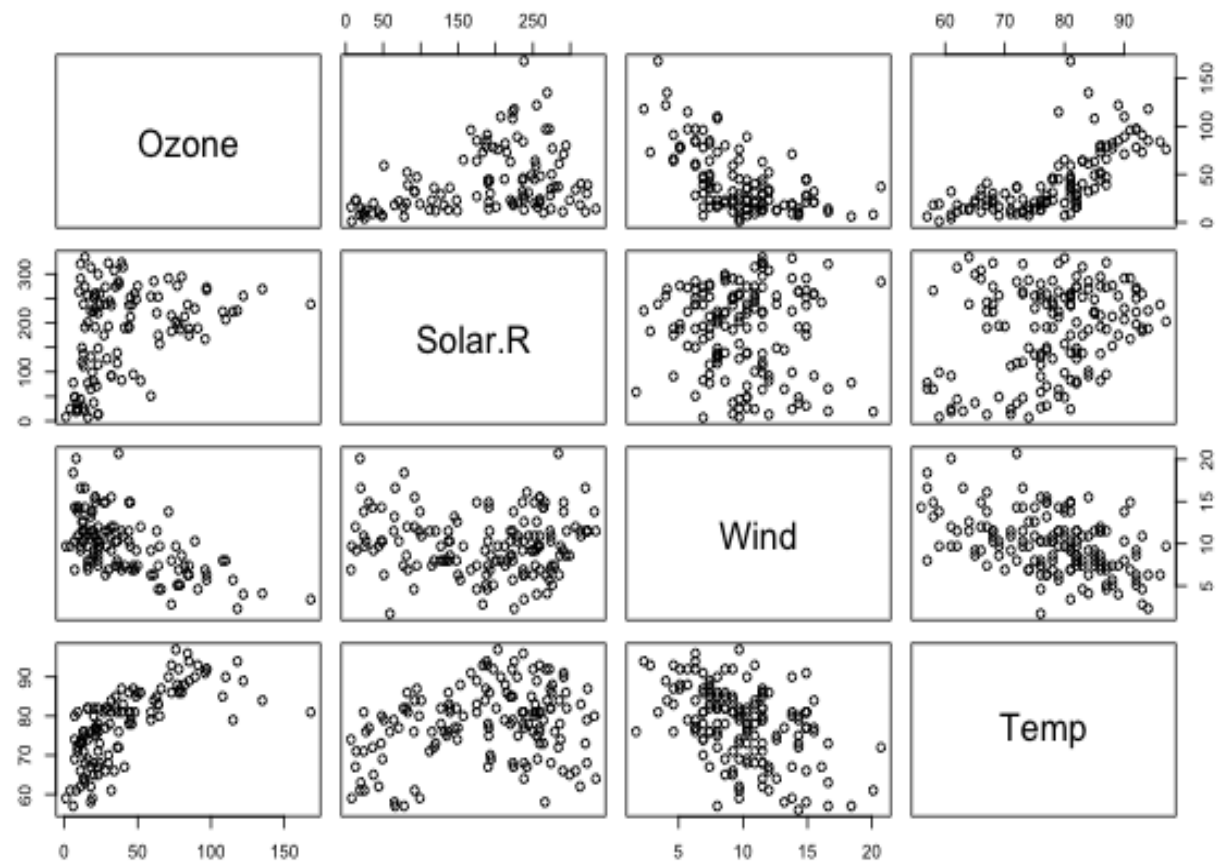
(Intercept)	Solar.R	Wind	Temp
-64.34208	0.05982	-3.33359	1.65209

---

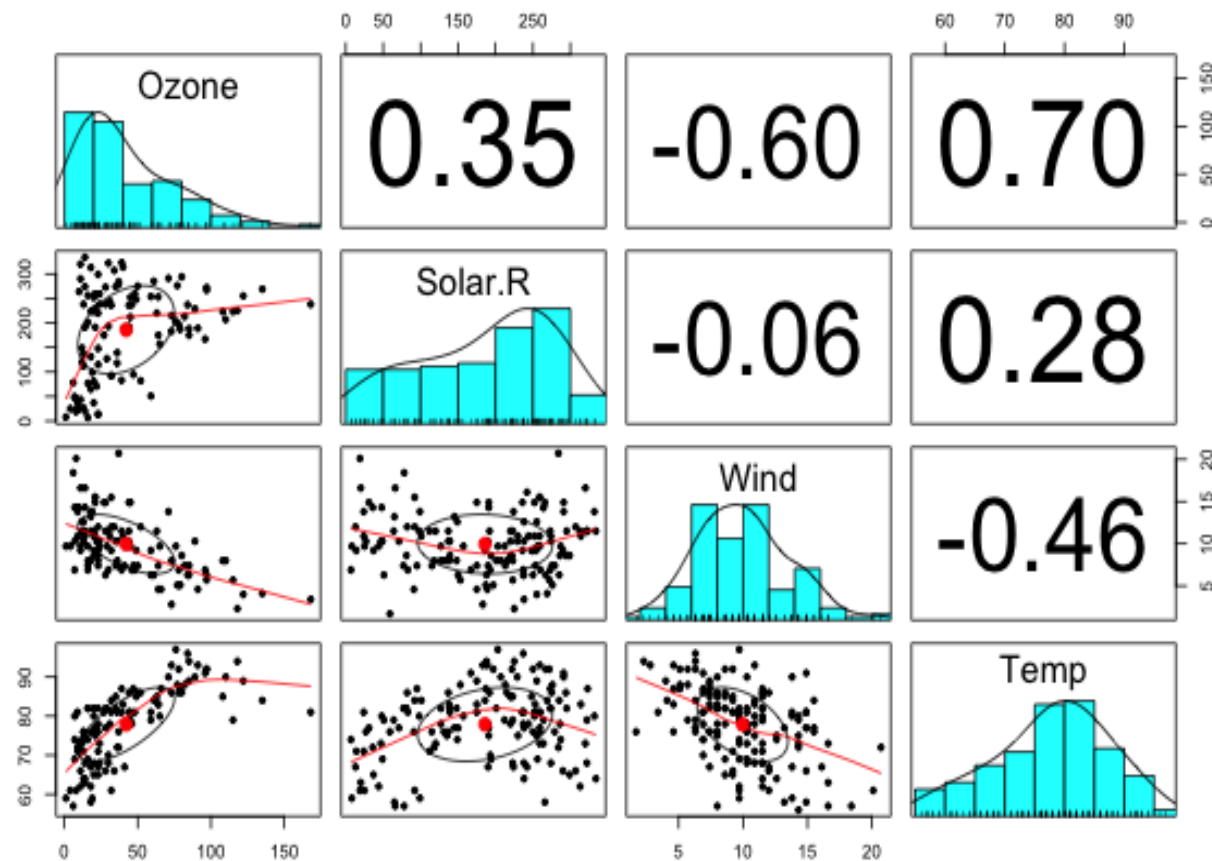
## Analizando os dados

```
png(filename=" ../figuras/pairsNY.png", width=600,
      height=400)
pairs(airquality[1:4])
dev.off()
library(psych)
png(filename=" ../figuras/pairsPanelNY.png",
      width=600, height=400)
pairs.panels(airquality[1:4])
dev.off()
```

# Correlação entre atributos



# Correlação entre atributos



# Avaliando o modelo

```
> summary(modelAirQuality)
```

Call:

```
lm(formula = Ozone ~ Solar.R + Wind + Temp, data = airquality)
```

Residuals:

Min	1Q	Median	3Q	Max
-40.485	-14.219	-3.551	10.097	95.619

1

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	-64.34208	23.05472	-2.791	0.00623	**
Solar.R	0.05982	0.02319	2.580	0.01124	*
Wind	-3.33359	0.65441	-5.094	1.52e-06	***
Temp	1.65209	0.25353	6.516	2.42e-09	***

2

---

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

Residual standard error: 21.18 on 107 degrees of freedom  
(42 observations deleted due to missingness)

3

Multiple R-squared: 0.6059, Adjusted R-squared: 0.5948

F-statistic: 54.83 on 3 and 107 DF, p-value: < 2.2e-16



---

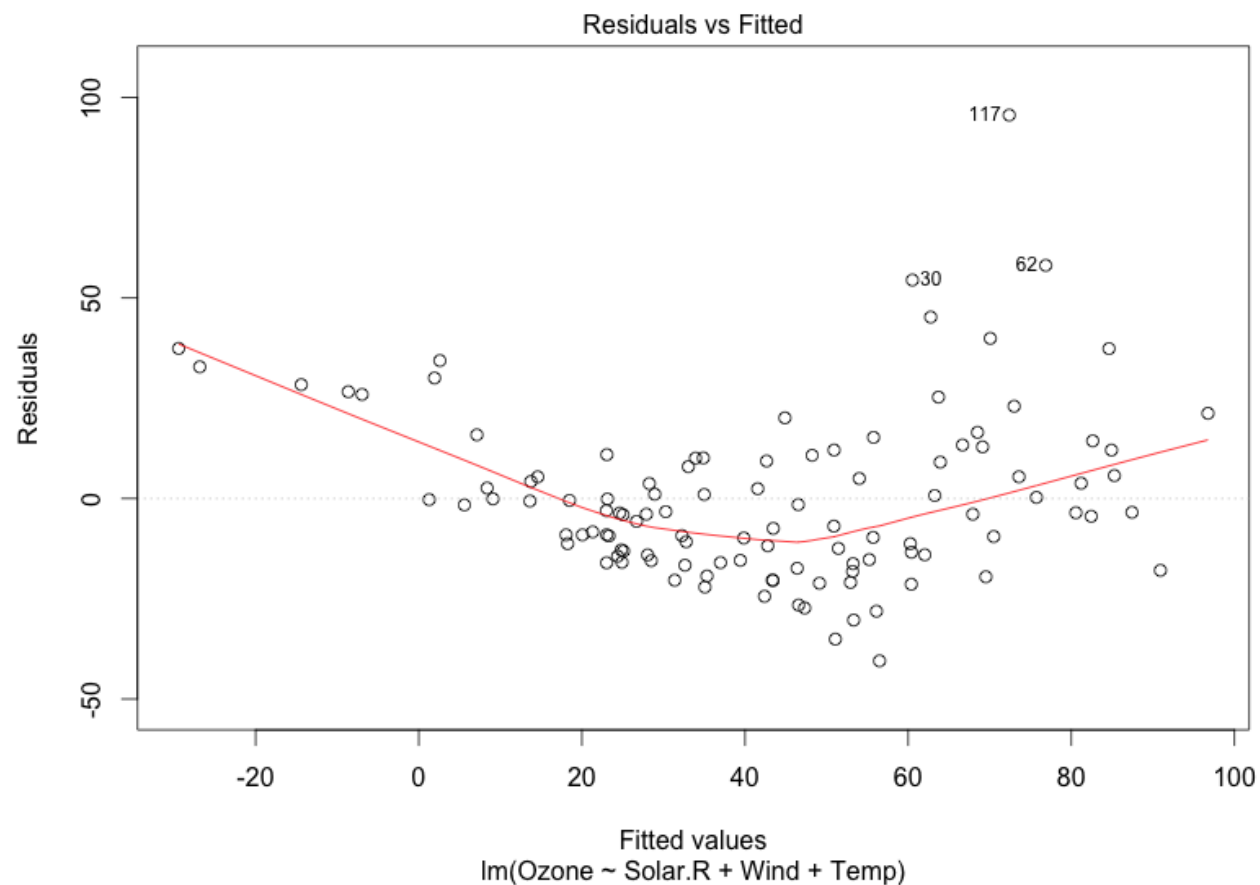
## Avaliando o modelo

1. Fornece um resumo sobre os erros do modelo. Um resíduo é igual ao valor verdadeiro menos o valor predito. O valor máximo do resíduo no problema anterior é 95.619. Isto significa que o modelo subestima o valor de Ozônio de pelo menos um dia em 95.619. Por outro lado, 50% dos erros ficam entre os valores do primeiro e terceiro quartis, ou seja, o modelo super-estima em 14.219 e sub-estima em 10.097.

- 
2. As estrelas (\* \* \*) indicam o poder de predição de cada atributo no modelo. Para fins práticos, considera-se que um atributo é estatisticamente significativo quando o nível de significância é menor ou igual a 0.05. Se o modelo possui poucos atributos estatisticamente significantes então deve-se considerar outros modelos para prever a variável de interesse.

- 
3. O *Multiple R-squared* (também chamado como coeficiente de determinação) fornece uma medida de quão bem o modelo como um todo explica os valores da variável dependente. É similar ao coeficiente de correlação, onde quanto mais próximo de 1.0, melhor o modelo explica os dados. O valor de *Adjusted R-squared* penaliza modelos com um número maior de variáveis independentes.

# Avaliando o modelo de forma visual



---

## Exercício

Qual dos modelos abaixo consegue explicar melhor os dados?

```
modelAirQuality2 <- lm(Ozone ~ Wind + Temp,  
                        data=airquality)  
summary(modelAirQuality2)
```

```
modelAirQuality3 <- lm(Ozone ~ Solar.R + Wind + Temp,  
                        data=airquality)  
summary(modelAirQuality3)
```

```
modelAirQuality4 <- lm(Ozone ~ Solar.R + Wind*Temp,  
                        data=airquality)
```

---

```
summary(modelAirQuality4)
```

```
airquality$Solar2 <- airquality$Solar.R^2
```

```
airquality$Wind2 <- airquality$Wind^2
```

```
airquality$Temp2 <- airquality$Temp^2
```

```
modelAirQuality5 <- lm(Ozone ~ Solar.R + Wind + Temp +  
                        Solar2 + Wind2 + Temp2,  
                        data=airquality)
```

```
summary(modelAirQuality5)
```

```
modelAirQuality6 <- lm(Ozone ~ Wind + Temp + Wind2 +  
                        Temp2, data=airquality)
```

```
summary(modelAirQuality6)
```