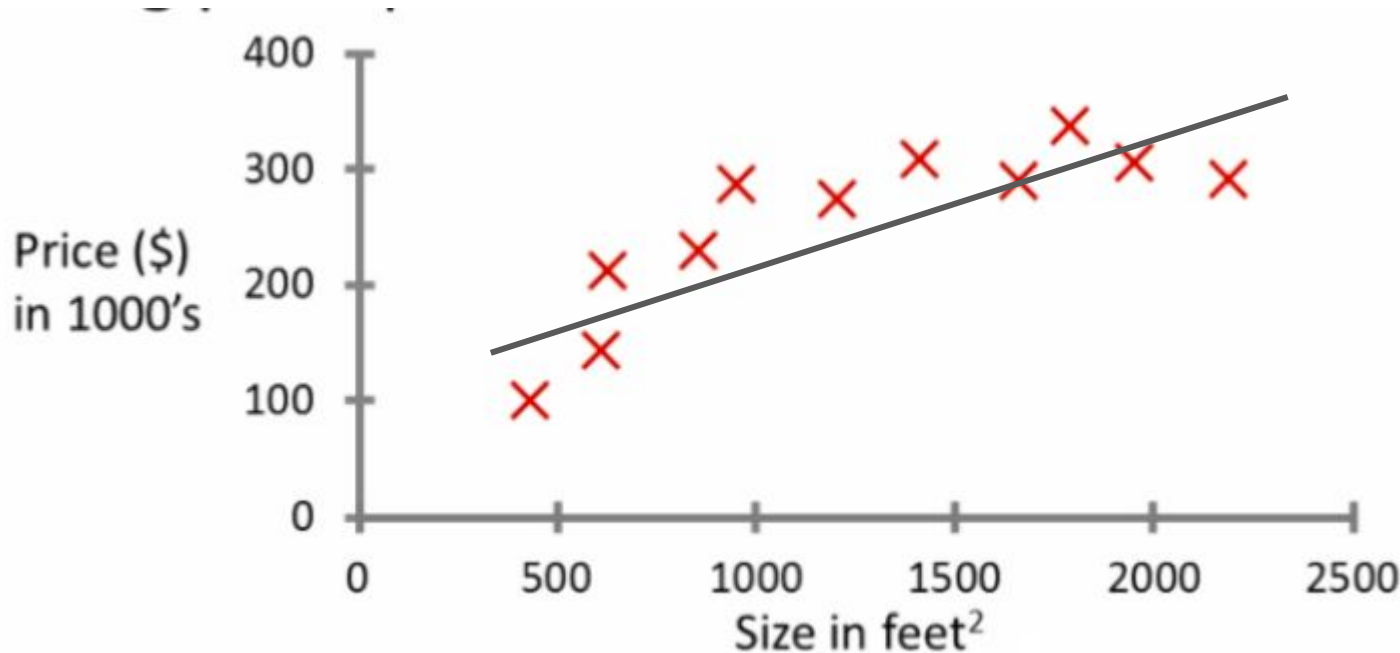




Regressão Linear Univariada

Curso Data Science - Jose Macedo

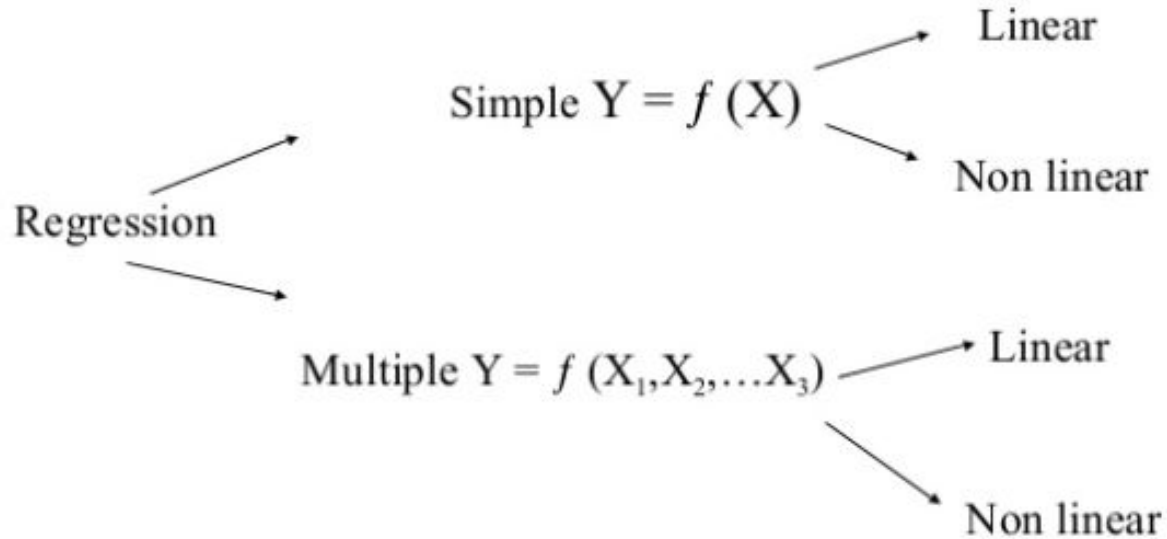
Motivação - Qual é o valor do apto para 700 feet²?



Motivação

- Regressão Linear é um dos modelos mais “simples” e rápido que existe.
- Existem muitas variações que buscam aprimorar os seus resultados, mas uma regressão linear simples pode gerar resultados bons dependendo do problema.
- Ao contrário do problema anterior onde tínhamos classes, na regressão linear queremos estimar valores reais
- O nosso modelo é uma representação da correlação das variáveis.
- Hoje vamos utilizar apenas 1 variável de entrada e 1 de saída

Tipos de Regressão



O Modelo

- Podemos usar estatísticas sobre os dados de treinamento para estimar os coeficientes exigidos pelo modelo para fazer previsões em novos dados.
- A linha reta para um modelo de regressão linear simples pode ser escrita como:

$$y = b_0 + b_1 \times x$$

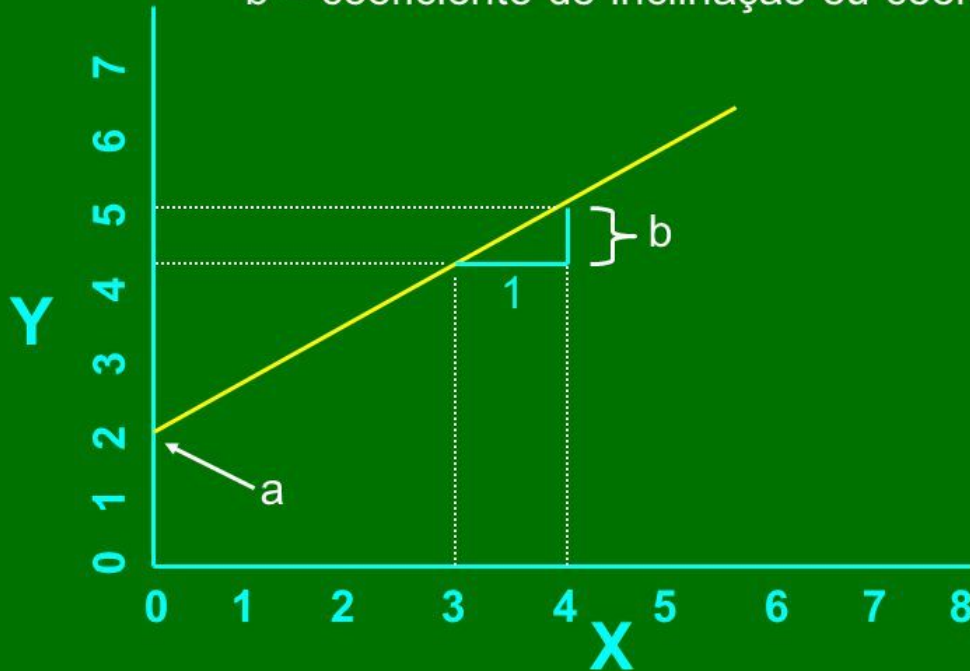
- Onde b_0 e b_1 são os coeficientes que devemos estimar a partir dos dados de treinamento.
- Uma vez que os coeficientes são conhecidos, podemos usar esta equação para estimar os valores de saída para y dado **novos exemplos** de entrada de x .

A equação da reta

$$Y = a + b \cdot X$$

a = intercepto

b = coeficiente de inclinação ou coeficiente angular



$$y_i = \alpha + \beta x_i$$

O Modelo

- Supondo que já possuímos β e α podemos fazer predições:

```
def predict(alpha, beta, x_i):  
    return beta * x_i + alpha
```

- Como escolhemos β e α ? Qualquer escolha que fizermos vai nos permitir calcular a predição para cada x_i . Como sabemos o valor real y_i podemos calcular o erro:

```
def error(alpha, beta, x_i, y_i):  
    """the error from predicting beta * x_i + alpha  
    when the actual value is y_i"""  
    return y_i - predict(alpha, beta, x_i)
```


O Modelo

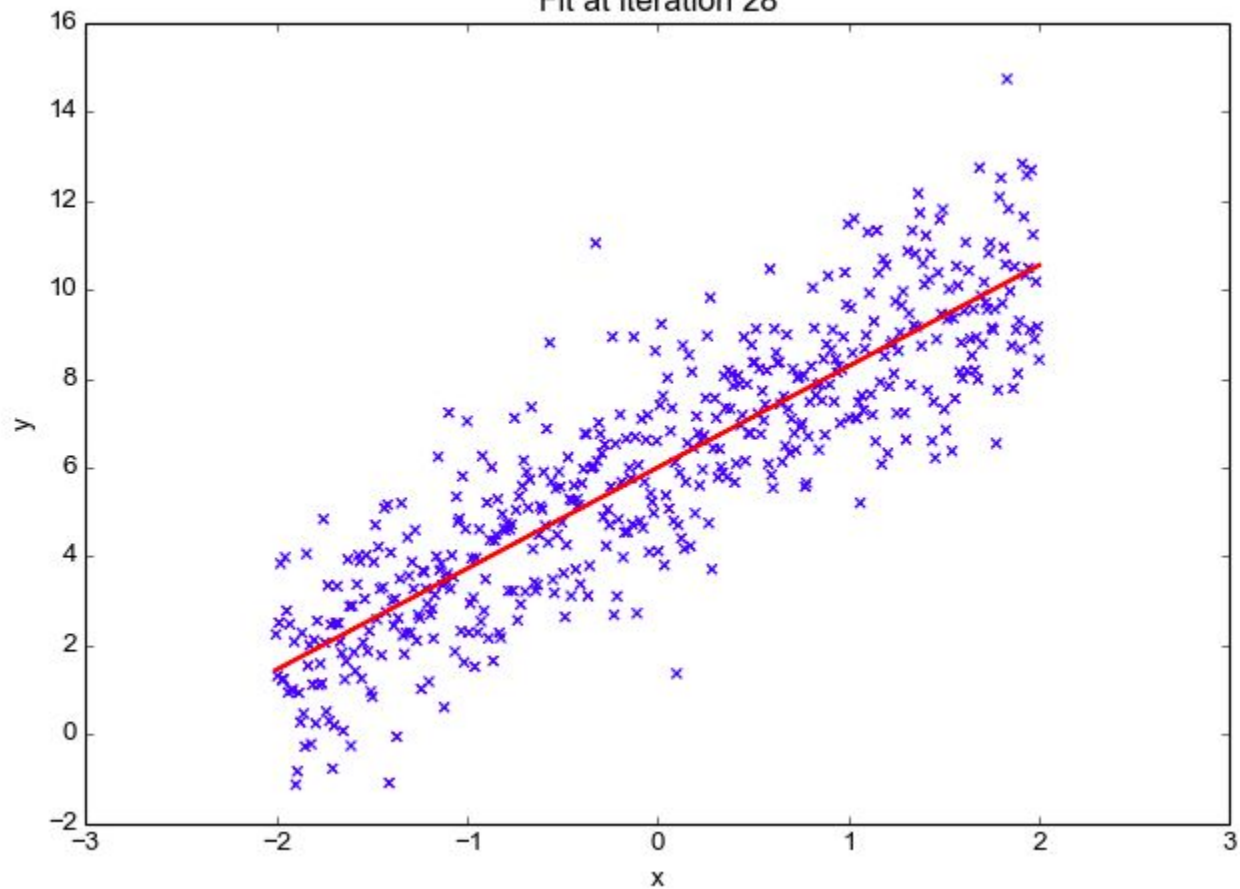
- Para os erros não se cancelarem, usamos os erros elevado ao quadrado

```
def sum_of_squared_errors(alpha, beta, x, y):  
    return sum(error(alpha, beta, x_i, y_i) ** 2  
               for x_i, y_i in zip(x, y))
```

- A solução dos mínimos quadrados é escolher β e α que minimizem a soma dos erros quadráticos

```
def least_squares_fit(x, y):  
    """given training values for x and y,  
    find the least-squares values of alpha and beta"""  
    beta = correlation(x, y) * standard_deviation(y) / standard_deviation(x)  
    alpha = mean(y) - beta * mean(x)  
    return alpha, beta
```

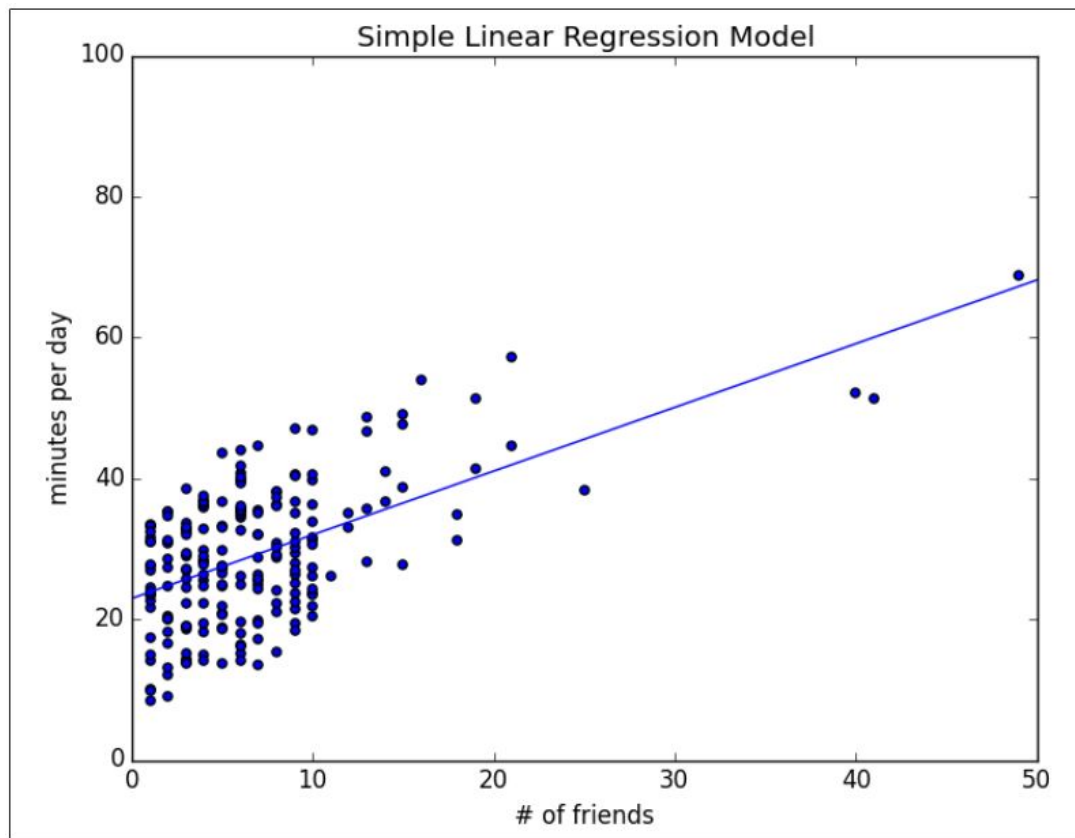
Fit at iteration 28



Modelo - Exemplo

-Isso nos dá $\beta = 0.903$


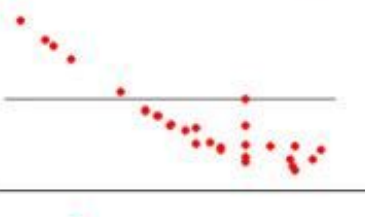

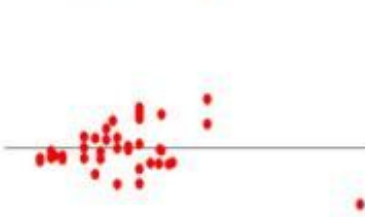
e $\alpha = 22.95$, e a seguinte reta:

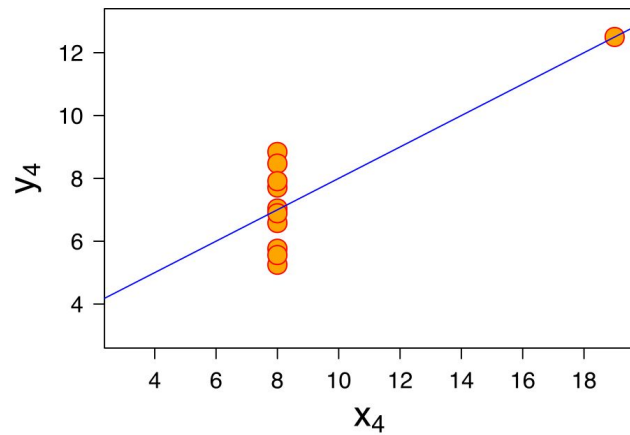
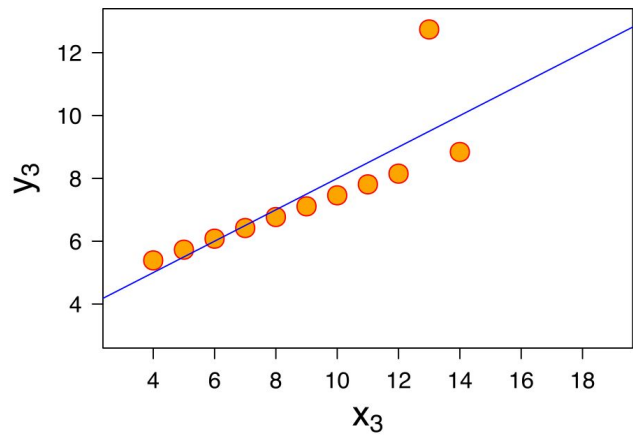
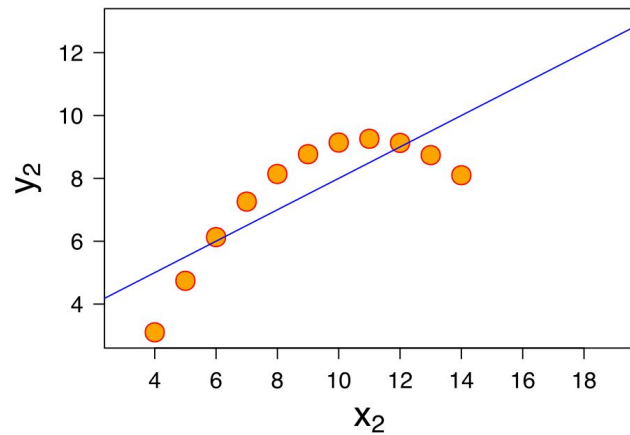
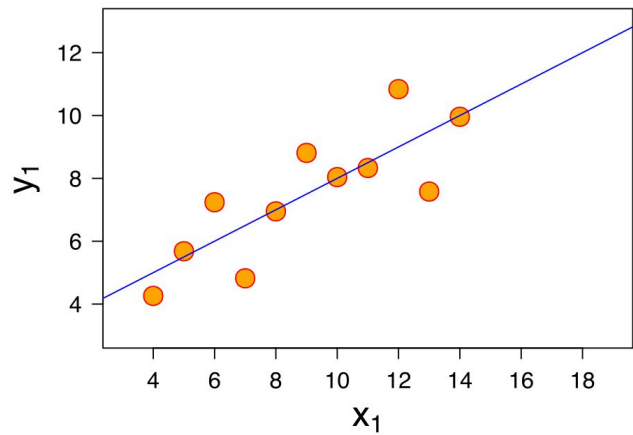


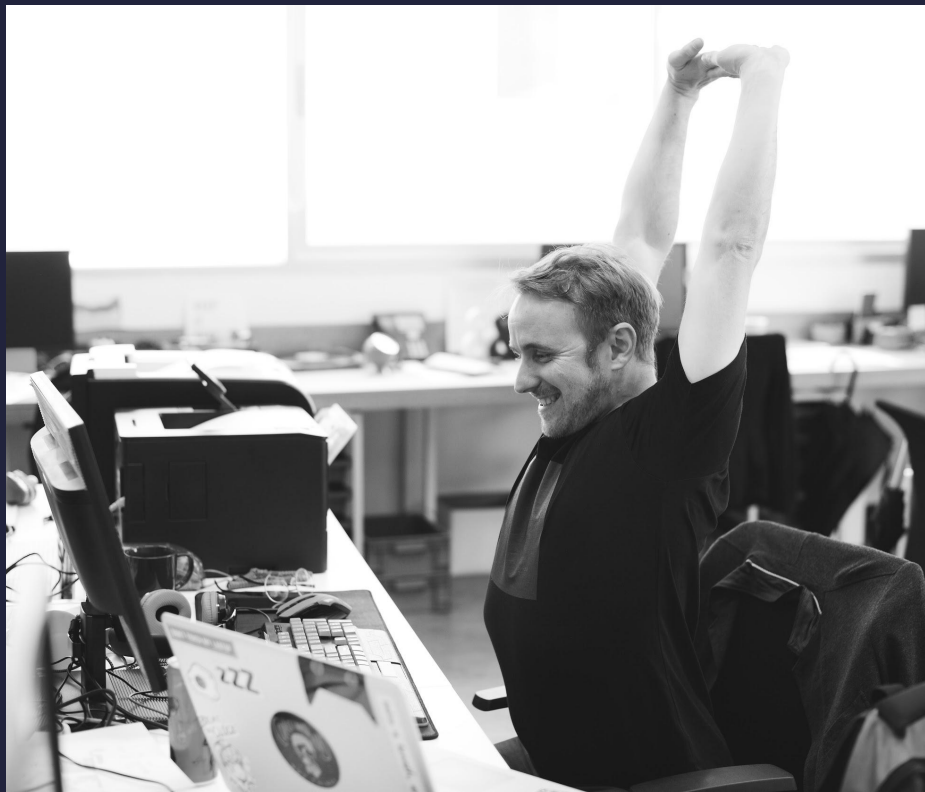
O Modelo

- Obviamente precisamos de uma maneira melhor de avaliar que apenas olhar um gráfico. Para isso usa-se o coeficiente de determinação ou R^2 , que nos dá uma idéia do quanto da variação foi capturada pelo modelo:

```
def total_sum_of_squares(y):  
    """the total squared variation of y_i's from their mean"""  
    return sum(v ** 2 for v in de_mean(y))  
  
def r_squared(alpha, beta, x, y):  
    """the fraction of variation in y captured by the model, which equals  
    1 - the fraction of variation in y not captured by the model"""  
  
    return 1.0 - (sum_of_squared_errors(alpha, beta, x, y) /  
                  total_sum_of_squares(y))  
  
r_squared(alpha, beta, num_friends_good, daily_minutes_good)    # 0.329
```

Problem	Example	Meaning
Uneven spread of residuals across fitted values		The model does not fit consistently at all values of X
Curvilinear pattern		The model may be missing a higher order term
Outlier (extreme Y value)		There may be an underlying special cause, such as data recording error
Influential value (extreme X value)		An observation has greater influence on the model compared to other observations





TUTORIAL

- ▶ Baixe os arquivos Jupyter Notebook;
- ▶ Abra o Tutorial de Regressão Linear