Universidade Federal do Paraná (UFPR)
Bacharelado em Informática Biomédica

Regressão

David Menotti
www.inf.ufpr.br/menotti/ci171-182

Hoje

- Regressão
  - Linear (e Múltipla)
  - Não-Linear (Exponencial / Logística)

Regressão

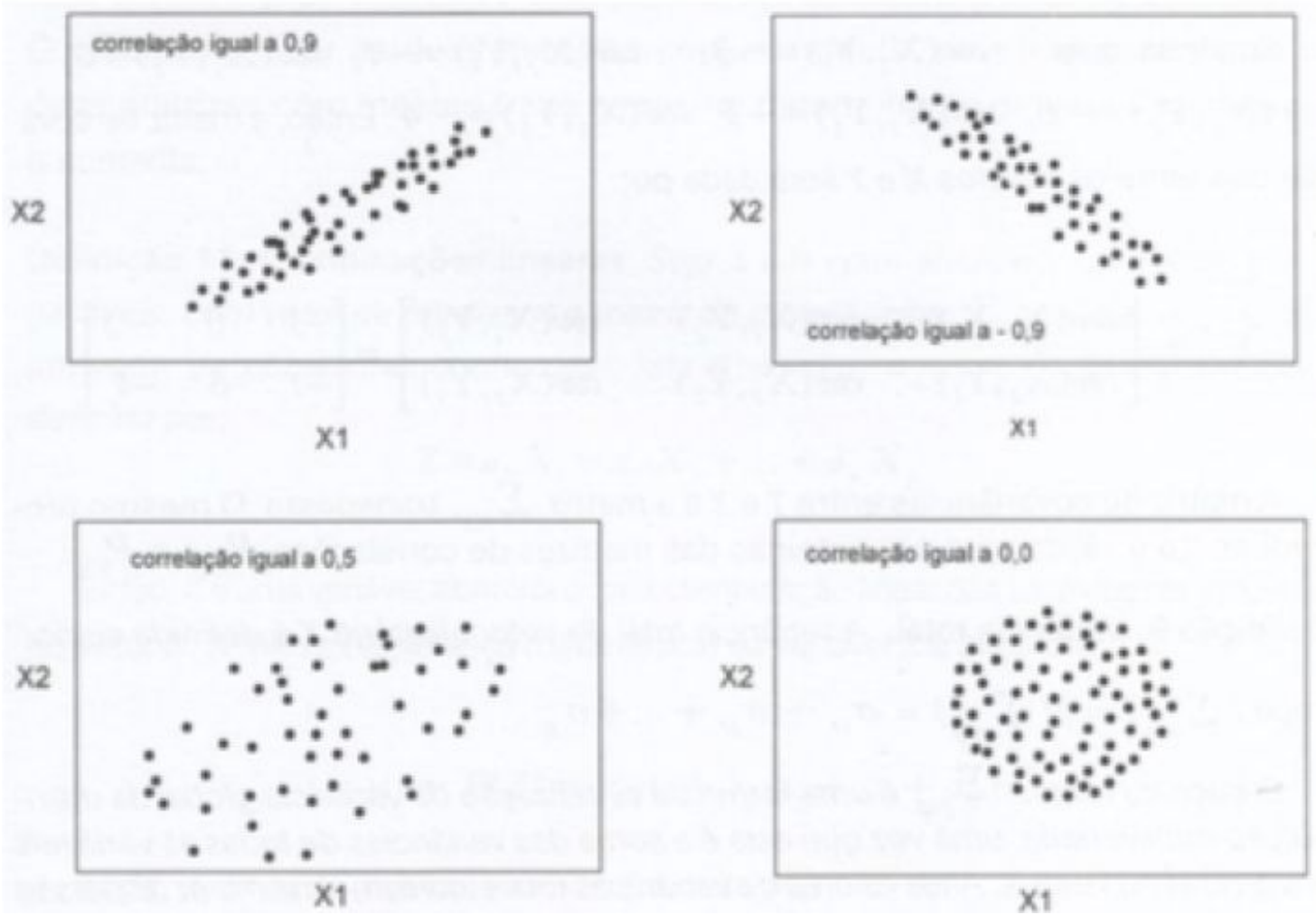
Agenda

- Regressão
  - Correlação
  - Análise de Regressão
    - Linear
    - Não-linear

Correlação

- Indica a força e a direção do relacionamento linear entre dois atributos.
- Trata-se de uma medida de **relação** entre dois atributos, embora correlação não implique **causalidade**
  - Duas variáveis podem estar altamente correlacionadas e não existir relação de causa e efeito entre elas.
- Em muitas aplicações duas ou mais variáveis estão relacionadas, sendo necessário explorar a natureza desta relação
  - Correlação muito próxima de 1 ou de -1 indica relação linear entre dois atributos.
  - Nesse caso é possível ajustar um modelo que expresse tal relação
  - Esse é o objetivo da análise de regressão.

Correlação


correlação igual a 0,9
X2
X1
correlação igual a - 0,9
X2
X1
correlação igual a 0,5
X2
X1
correlação igual a 0,0
X2
X1

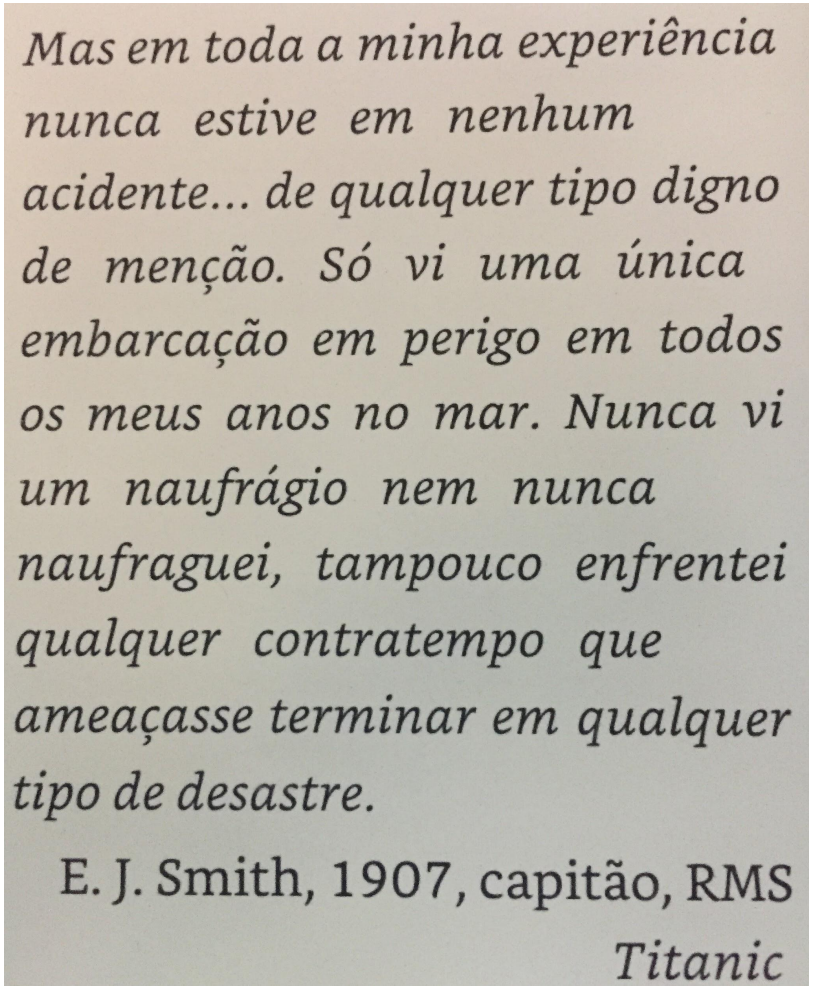
Análise da Regressão

Objetivo: Determinar o modelo que expressa esta relação (modelo de regressão) a qual é ajustada aos dados

- Permite construir um modelo matemático que representa dois atributos (x e y)
- y = f(x), em que f(.) é a função que relaciona x e y
- x é a variável independente da equação
- y é a variável dependente das variações de x

Análise da Regressão

- Esse modelo pode ser usado para predizer o valor de y para um dado valor de x
  - Realizar previsões sobre um comportamento futuro de algum fenômeno da realidade.
  - Extrapolar para o futuro relações de causa-efeito entre as variáveis, já observadas no passado.
  - Cuidado com a extrapolação

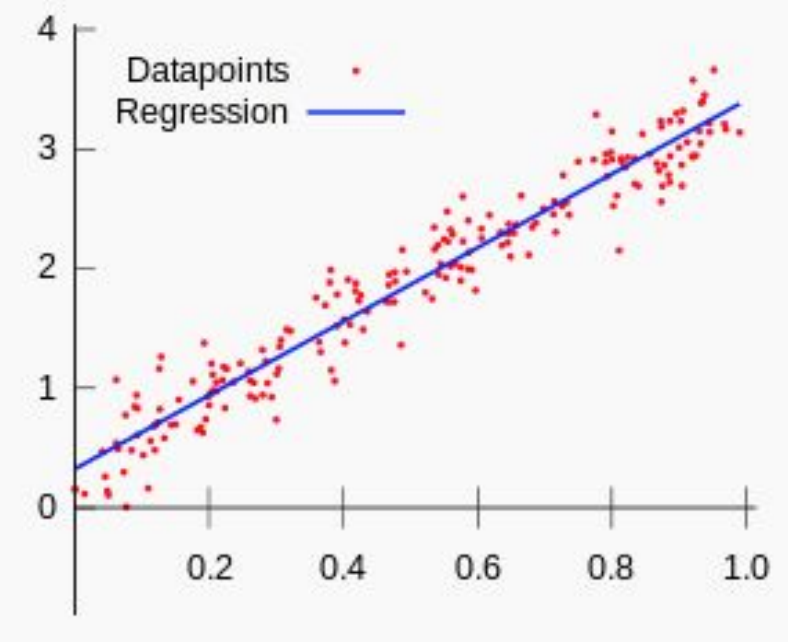


Análise da Regressão

- A análise de regressão compreende dois tipos básicos:
  - Linear
  - Linear Múltipla
  - Não Linear
    - Modelos exponenciais
    - Modelos logísticos

Regressão Linear

- Considera que a relação da resposta às variáveis é uma função linear de alguns parâmetros



- Modelos de regressão linear são frequentemente ajustados usando a abordagem dos mínimos quadrados.

Método dos Mínimos Quadrados

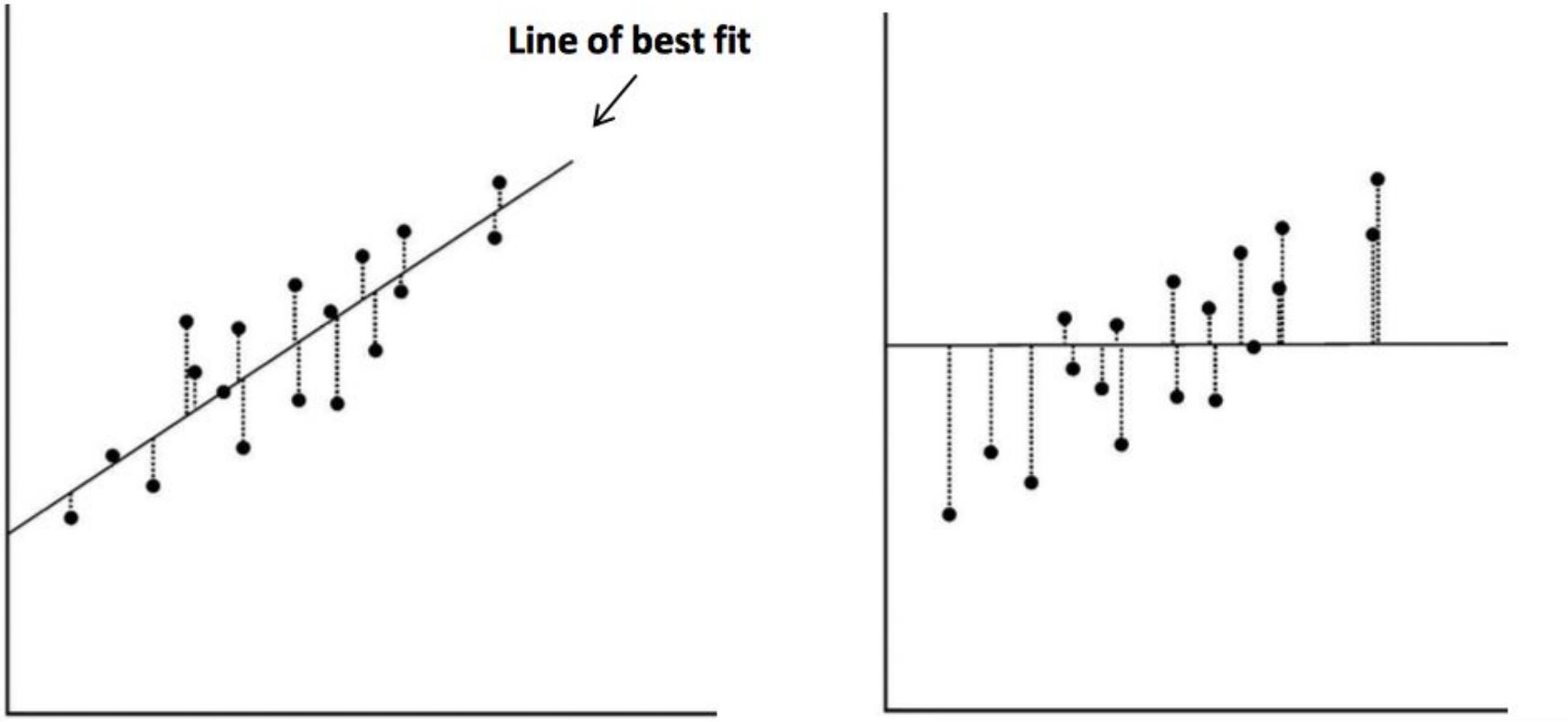
- Seja a equação da reta y = ax + b (ou y = α + β x) que modela a relação entre uma variável independente e uma variável dependente.
- Seja um conjunto de n pontos de dados conhecidos:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (n_x, y_n)\}$$

- O objetivo é encontrar os coeficientes a e b para o modelo que minimizem a medida de erro quadrático
  RSS (Residual Sum of Squares)

$$RSS = \sum_{i=1}^{n} (y - y_i)^2$$

Regressão Linear



Método dos Mínimos Quadrados

- Os parâmetros da reta podem ser estimados através do conjunto de dados D usando as seguintes equações:

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$b = \frac{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i y_i \sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

Método dos Mínimos Quadrados

Exemplo

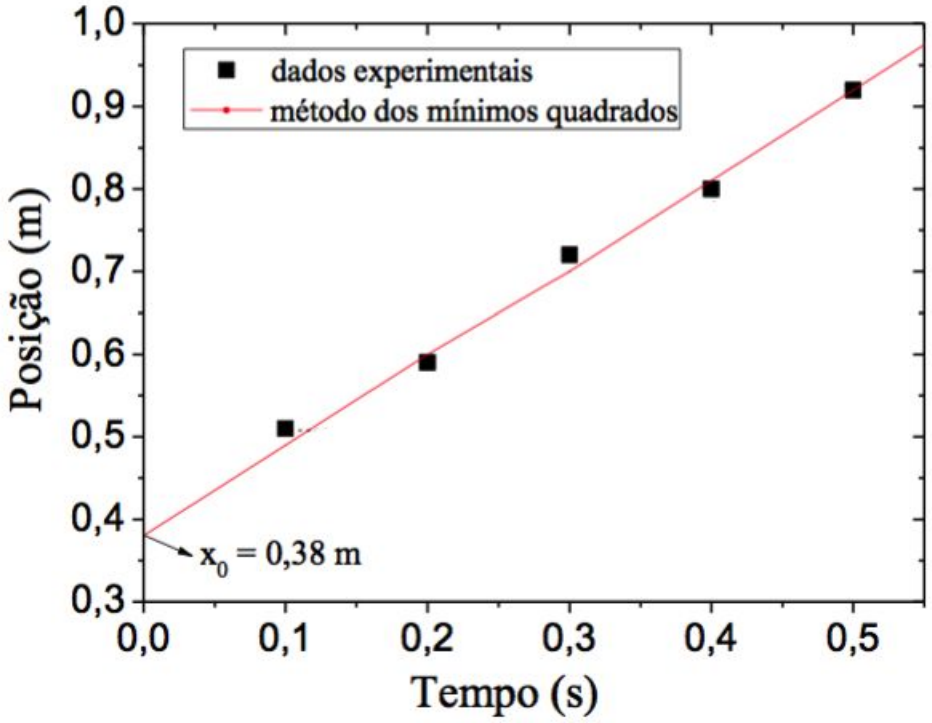
- Considere os dados da tabela abaixo. Encontre os coeficientes *a* e *b* e calcule a posição (y) no tempo x = 0,6

X - tempo (s)	Y - posição (m)
0,100	0,51
0,200	0,59
0,300	0,72
0,400	0,80
0,500	0,92

- Da tabela acima temos, $\sum x = 1.5$ $\sum xy = 1.17$ $\sum y = 3.54$ $\sum x^2 = 0.55$
- Usando as equações anteriores, temos a = 1,08 e b = 0,38
- Desta forma, temos o modelo de regressão y = 1,08x + 0,38

Método dos Mínimos Quadrados

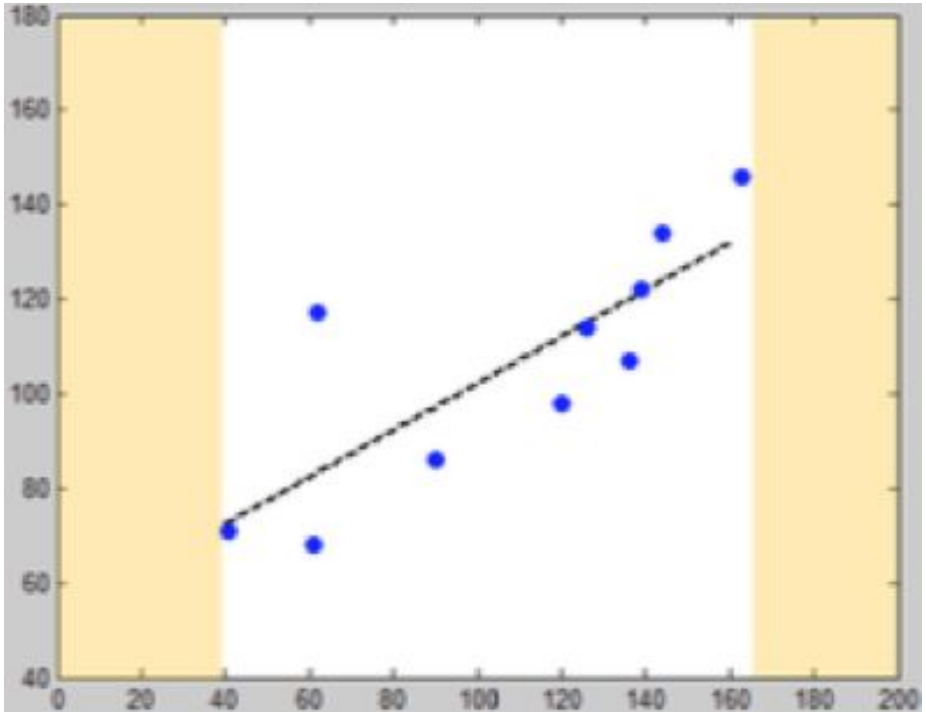
Exemplo



- Respondendo a questão anterior, para x = 0,6, y = 1,02

Método dos Mínimos Quadrados

- Modelos de regressão linear não costumam ser válidos para fins de **extrapolação**, ou seja, predizer um valor fora do domínio dos dados



Análise de resíduos

- Como avaliar a qualidade do modelo?
  - Os erros têm distribuição normal?
  - Existem "outliers'" no conjunto de dados?
  - O modelo gerado é adequado?

- Podemos responder essas perguntas analisando os resíduos, o qual é dado pela diferença entre o y_i e sua estimativa $\hat{y}_i$

$$e_i = y_i - \hat{y}_i$$

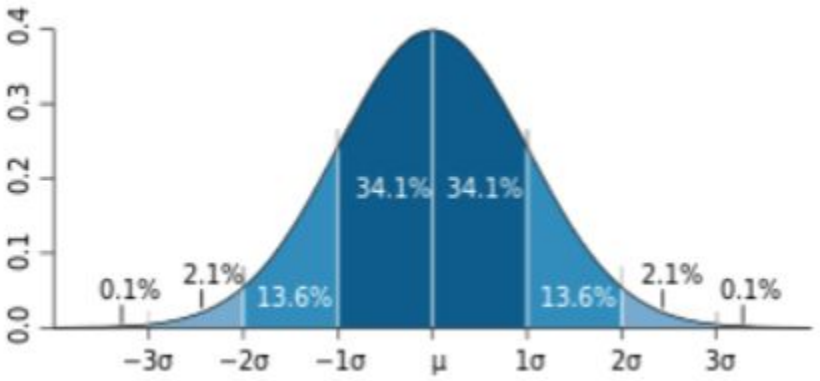
Análise de resíduos

Outliers

- Construir um histograma de frequência dos resíduos
- Normalizar de modo a ter média zero e desvio 1

$$Z = \frac{e-\mu}{\sigma}$$

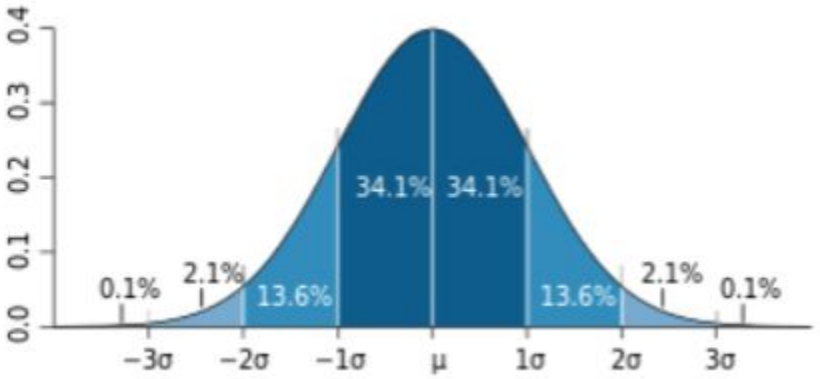
- O histograma de resíduos deve ser semelhante a uma normal.



Análise de resíduos

Outliers

- Se os erros tiverem uma distribuição normal,
  - Aproximadamente, 95% dos resíduos estarão no intervalo de um desvio padrão da média
- Caso contrário, deve existir a presença de "outlier",
  - ou seja, valores atípicos ao restante dos dados.
- O que fazer com os outliers?

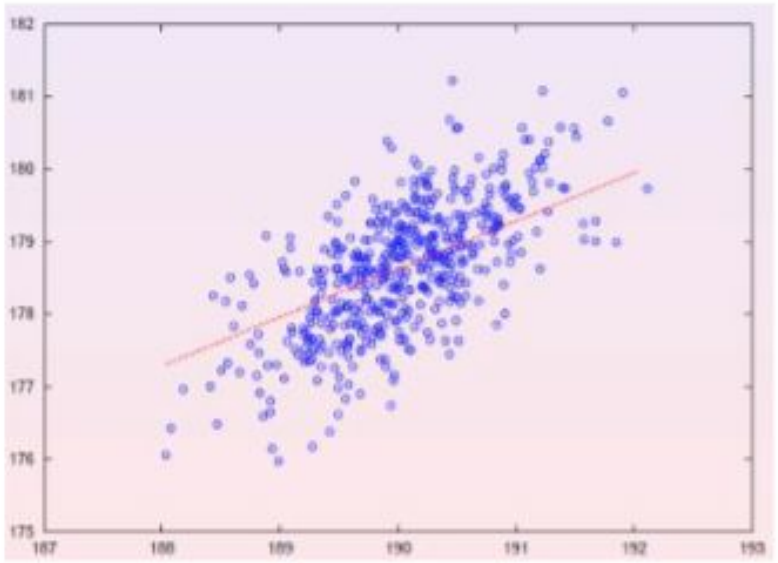


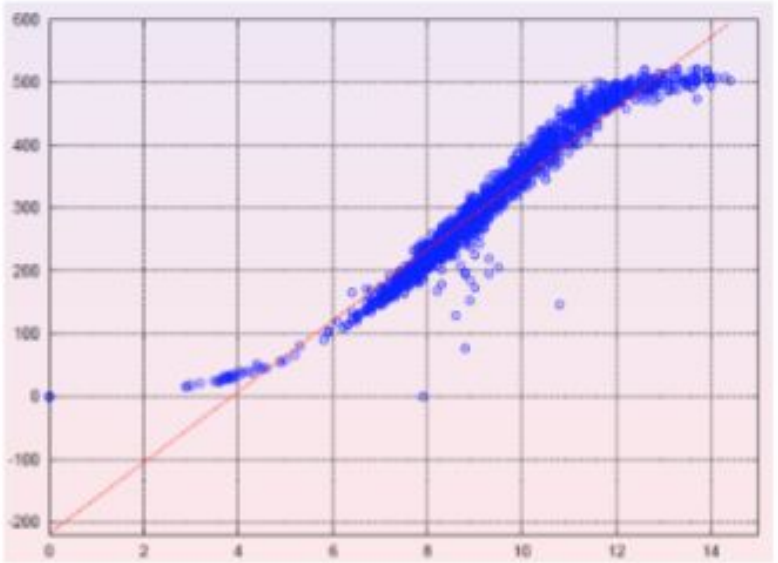
Coeficiente de Determinação (R^2)

- Uma forma de avaliar a qualidade de ajuste do modelo.
- Indica a quantidade de variabilidade dos dados que o modelo de regressão é capaz de explicar.
- Varia entre 0 e 1, indicando quanto o modelo consegue explicar os valores observados.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}$$

Coeficiente de Determinação (R^2)





$$R^2 = 0.44$$

$$R^2 = 0.93$$

Regressão Linear

Exercício

- Programa exemplo de regressão usando `scikit-learn`: **Linear Regression Example**¶

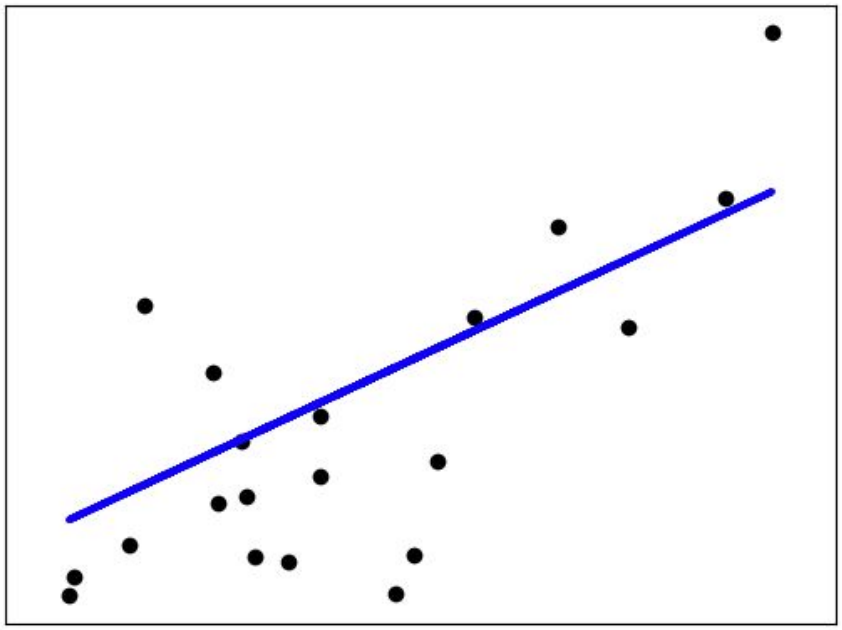
```
print(__doc__)


# Code source: Jaques Grobler
# License: BSD 3 clause


import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Load the diabetes dataset
diabetes = datasets.load_diabetes()


# Use only one feature
diabetes_X = diabetes.data[:, np.newaxis, 2]

# Split the data into training/testing sets
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]

# Split the targets into training/testing sets
diabetes_y_train = diabetes.target[:-20]
diabetes_y_test = diabetes.target[-20:]

# Create linear regression object
regr = linear_model.LinearRegression()
```

```
# Train the model using the training sets
regr.fit(diabetes_X_train, diabetes_y_train)

# Make predictions using the testing set
diabetes_y_pred = regr.predict(diabetes_X_test)

# The coefficients
print('Coefficients: \n', regr.coef_)
# The mean squared error
print("Mean squared error: %.2f"
      % mean_squared_error(diabetes_y_test, diabetes_y_pred))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % r2_score(diabetes_y_test, diabetes_y_pred))

# Plot outputs
plt.scatter(diabetes_X_test, diabetes_y_test,  color='black')
plt.plot(diabetes_X_test, diabetes_y_pred, color='blue', linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()
```

Regressão Linear

Exercício

The coefficients, the residual sum of squares and the variance score are also calculated.



Out:

```
Coefficients:
 [938.23786125]
Mean squared error: 2548.07
Variance score: 0.47
```

Regressão Linear Múltipla

- A regressão múltipla funciona de forma parecida com a regressão simples

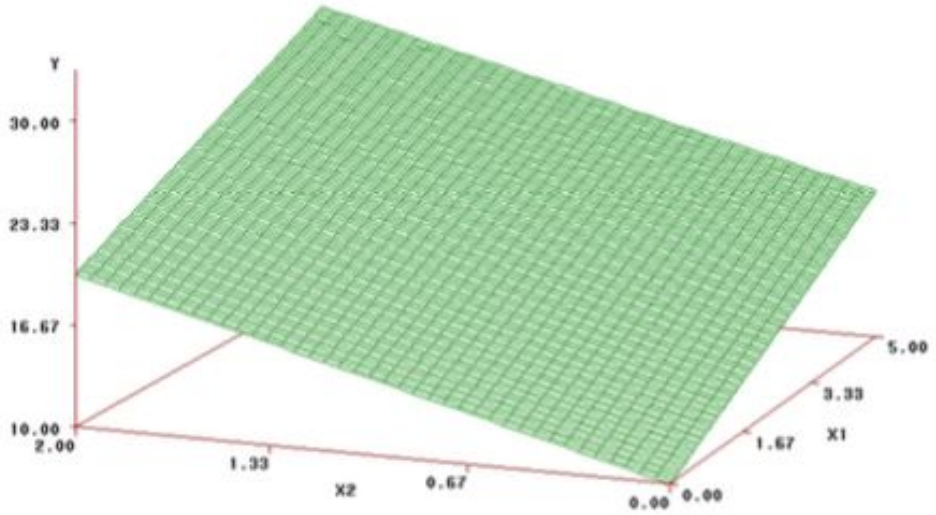
- Leva em consideração diversas variáveis de entrada x_i, $i = 1, ..., p$, influenciando ao mesmo tempo uma única variável de saída, y

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon$$

Regressão Linear Múltipla

Exemplo:

$$y = 10 + 2x_1 + 5x_2$$



- A função de regressão na regressão múltipla é chamada de superfície de resposta
- Descreve um hiperplano no espaço p-dimensional das variáveis de entrada x

Regressão Linear Múltipla

Como calcular a superfície de regressão?

- Usar o método dos mínimos quadrados como feito na regressão linear simples
- A diferença é que agora temos um elevado número de parâmetros na forma

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon$$

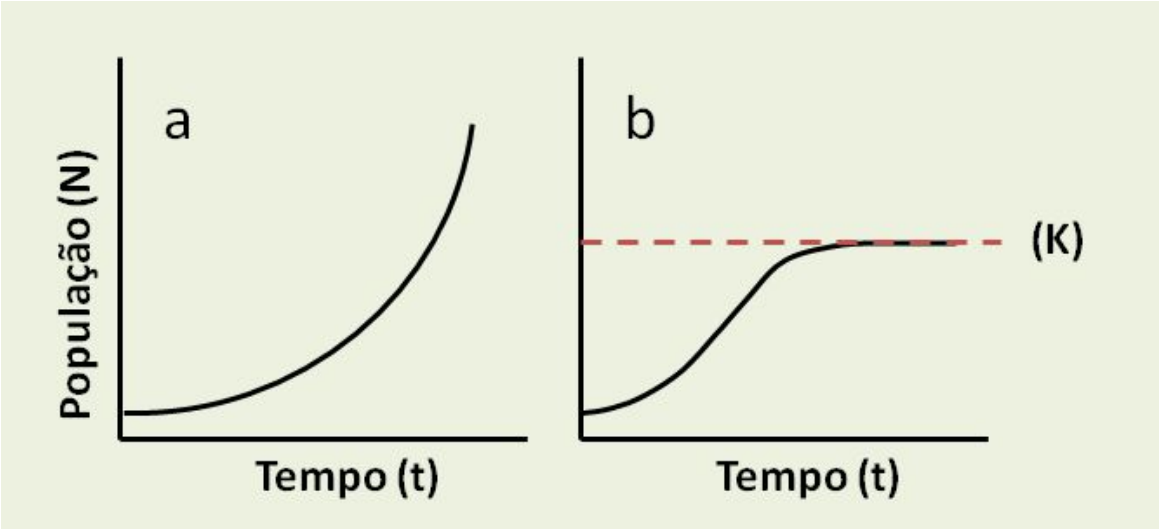
- **Solução**: Expressar as operações matemáticas utilizando notação matricial

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} \cdots & x_{p1} \\ 1 & x_{12} \cdots & x_{p2} \\ \cdots & \cdots \cdots & \cdots \\ 1 & x_{1p} \cdots & x_{pn} \end{pmatrix} * \begin{pmatrix} \beta_0 \\ \beta_1 \\ \cdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdots \\ \varepsilon_n \end{pmatrix}$$

- $y = X\beta + \epsilon$

Regressão Não Linear

- Em alguns casos o modelo linear pode não ser o mais adequado.
- Muitas aplicações biológicas são modeladas por meio de relações não lineares.
- Por exemplo, padrões de crescimento podem seguir modelos:
  - Exponenciais: onde a população aumenta sem limites
  - Logísticos: onde a população cresce rapidamente no início, desacelera e se mantém estável.



Método dos Mínimos Quadrados
Caso Exponencial

- O método dos mínimos quadrados pode ser facilmente adaptado para o caso exponencial, usando logaritmos neperianos.

- Nesse caso, $y' = ln(y)$

- A equação que modela a relação entre uma variável independente e uma variável dependente é dada por $y = be^{ax}$

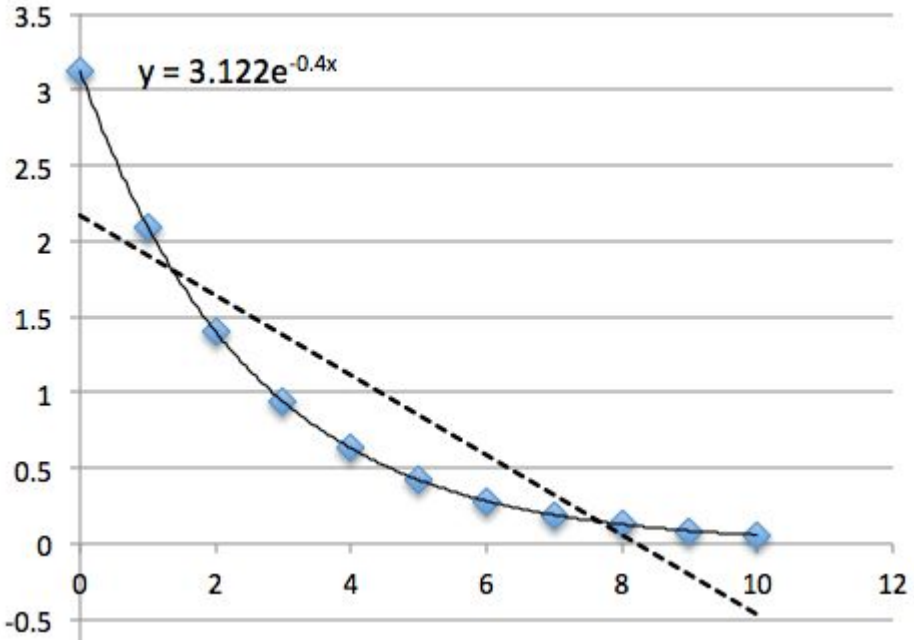
$$a = \frac{n\sum_{i=1}^{n} x_i y_i' - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i'}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$b = e^{\overline{y'} - a\overline{x}}$$

Método dos Mínimos Quadrados
Caso Exponencial

Exemplo

	x	y	ln(y) =y'	x2	x*y'
	0	3.12	1.1378	0	0
	1	2.091	0.7376	1	0.738
	2	1.402	0.3379	4	0.676
	3	0.94	-0.062	9	-0.186
	4	0.63	-0.462	16	-1.848
	5	0.422	-0.863	25	-4.314
	6	0.283	-1.262	36	-7.574
	7	0.19	-1.661	49	-11.63
	8	0.127	-2.064	64	-16.51
	9	0.085	-2.465	81	-22.19
	10	0.057	-2.865	100	-28.65
SUM	55	9.347	-9.49	385	-91.47
AVG	5		-0.863		-8.316



Regressão Logística

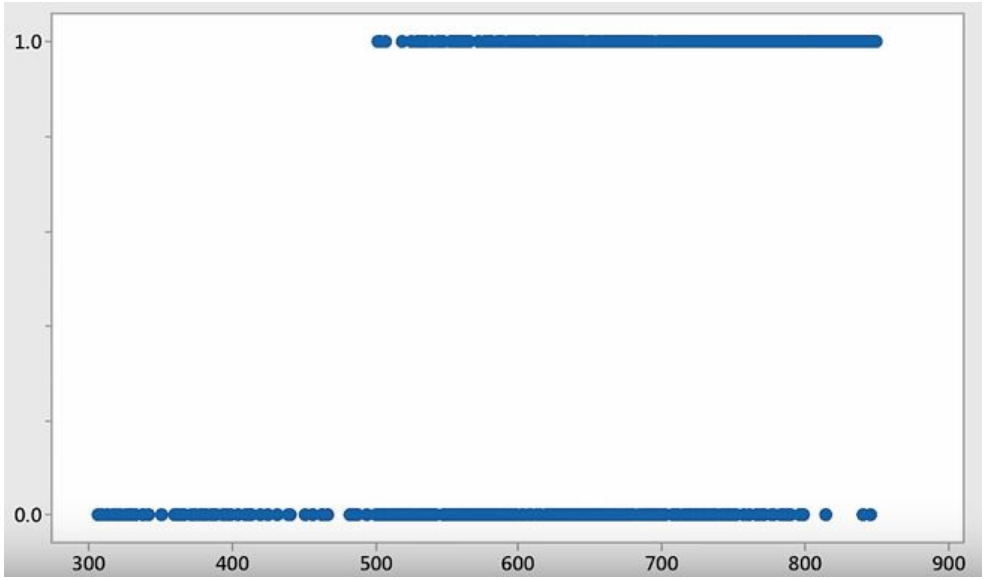
Exemplo

- Considere que alguém queira comprar uma casa e está na busca por um financiamento.
- Os bancos mantém para cada cliente um **score**, que varia de 300 a 850
- Suponha que um dado cliente tem um score de 720 e gostaria de saber qual é a probabilidade de ter seu crédito aprovado pela instituição financeira.
- Esse cliente encontrou ainda dados de 1000 clientes com seus respectivos escores, que tiveram seus pedidos aprovados ou não.
- Notem que nesse caso a variável y é dicotômica
  - Aprovado (1)
  - Negado (0)

Regressão Logística

Exemplo

creditScore	approved
655	0
692	0
681	0
663	1
688	1
693	1
699	0
699	1
683	1
698	0
655	1
703	0
704	1
745	1
702	1



- Note que a distribuição dos dados é diferente (binomial).
- Os modelos de regressão vistos até então não funcionam nesse caso.
- Solução: Regressão Logística

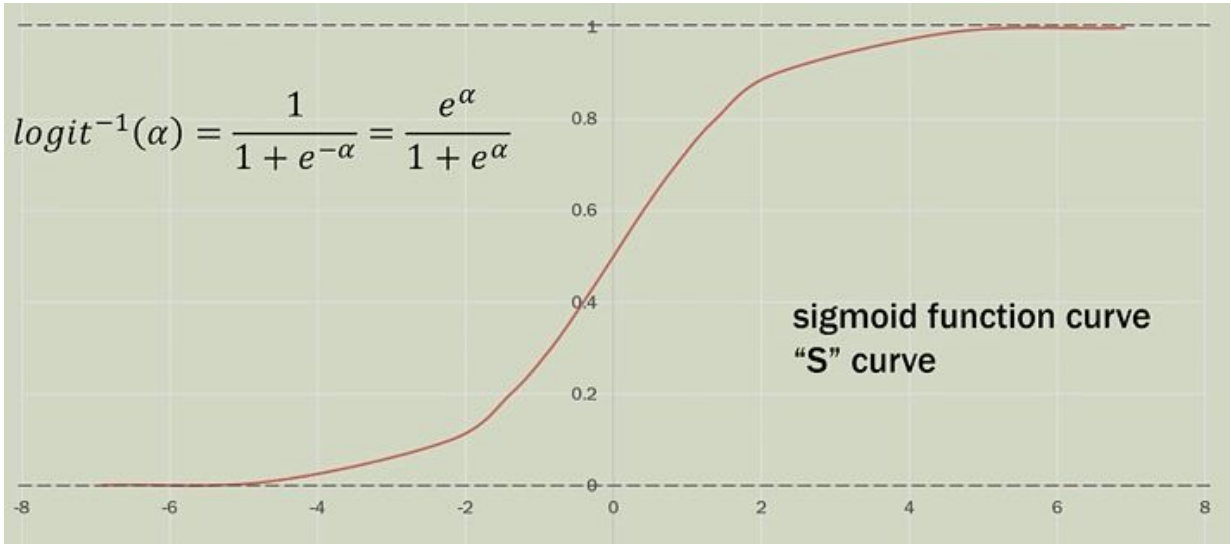
Regressão Logística

A regressão logística tem como objetivo:

- Modelar a probabilidade de um evento ocorrer dependendo dos valores das variáveis independentes.
- Estimar a probabilidade de um evento ocorrer (e também de não ocorrer) para uma dada observação.
- Pode ser usada como um classificador, atribuindo uma classe ao padrão de entrada.
  - No nosso exemplo, crédito aprovado ou reprovado.

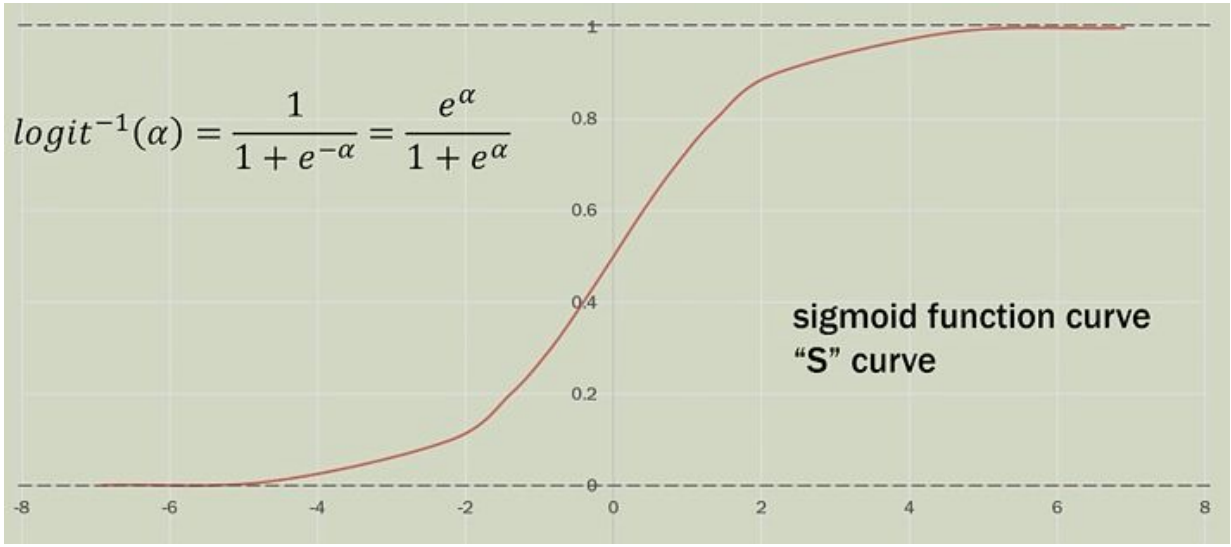
Regressão Logística

- A variável dependente na regressão logística segue a distribuição de Bernoulli.
- Distribuição discreta de espaço amostral {0,1} que tem probabilidade de sucesso p e falha q = 1 - p
- Na regressão logística estimamos p para qualquer combinação linear das variáveis independentes.

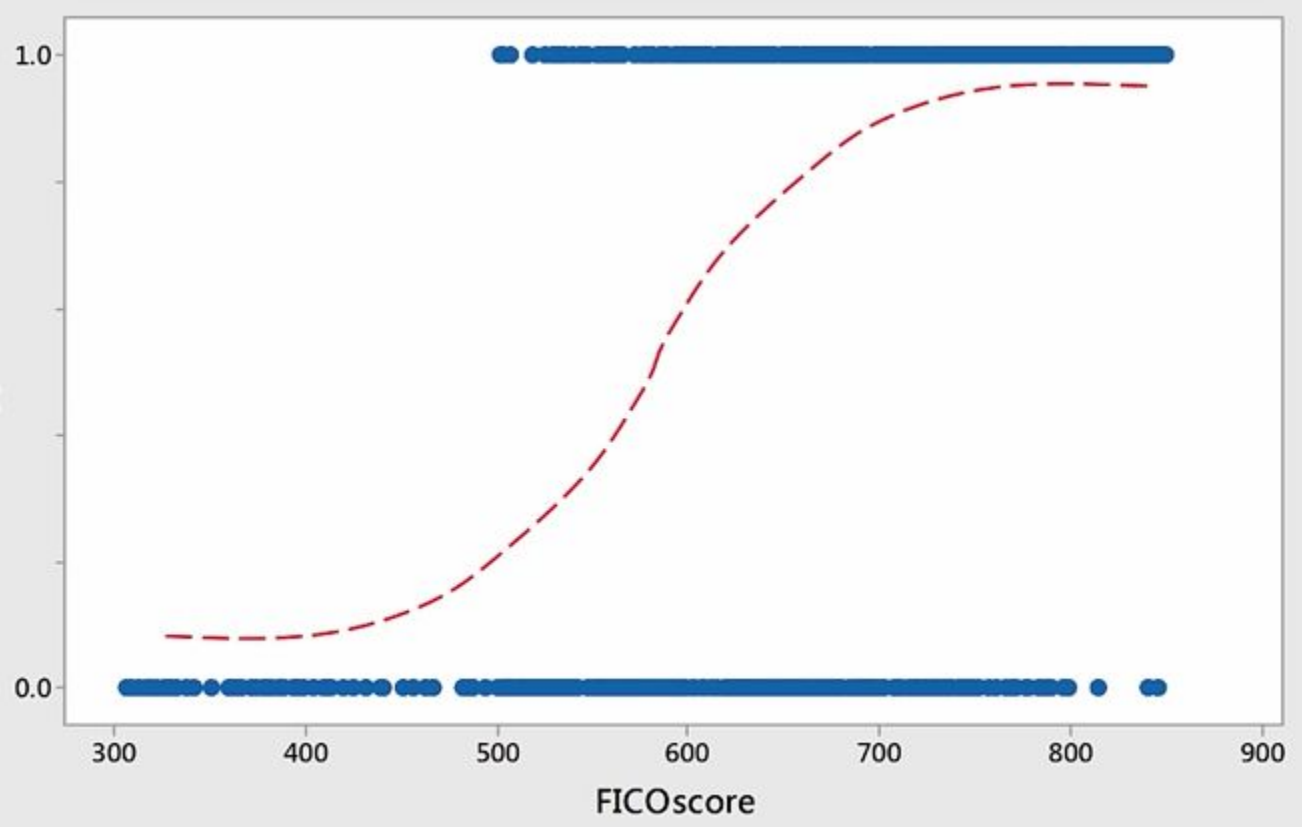


Regressão Logística

- Isso pode ser alcançado ajustando o seguinte modelo (função logística)
- MLE (*Maximum Likehood Estimation*) é usada para estimar os coeficiente do modelo.



Regressão Logística



- Esse modelo diz basicamente que a probabilidade de conseguir crédito sobe em função do *score*

Regressão Logística

Exercício

- Programa exemplo de regressão usando `scikit-learn`: **Logistic Function**

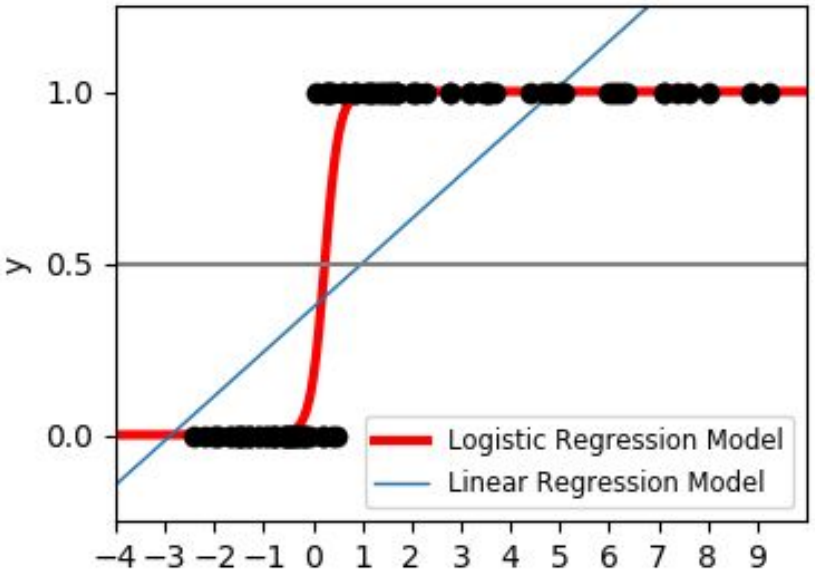
```
print(__doc__)


# Code source: Gael Varoquaux
# License: BSD 3 clause

import numpy as np
import matplotlib.pyplot as plt

from sklearn import linear_model

# this is our test set, it's just a straight line
# Gaussian noise
xmin, xmax = -5, 5
n_samples = 100
np.random.seed(0)
X = np.random.normal(size=n_samples)
y = (X > 0).astype(np.float)
X[X > 0] *= 4
X += .3 * np.random.normal(size=n_samples)

X = X[:, np.newaxis]
# run the classifier
clf = linear_model.LogisticRegression(C=1e5)
clf.fit(X, y)
```

```
# and plot the result
plt.figure(1, figsize=(4, 3))
plt.clf()
plt.scatter(X.ravel(), y, color='black', zorder=20)
X_test = np.linspace(-5, 10, 300)


def model(x):
    return 1 / (1 + np.exp(-x))
loss = model(X_test * clf.coef_ + clf.intercept_).ravel()
plt.plot(X_test, loss, color='red', linewidth=3)

ols = linear_model.LinearRegression()
ols.fit(X, y)
plt.plot(X_test, ols.coef_ * X_test + ols.intercept_, linewidth=1)
plt.axhline(.5, color='.5')

plt.ylabel('y')
plt.xlabel('X')
plt.xticks(range(-5, 10))
plt.yticks([0, 0.5, 1])
plt.ylim(-.25, 1.25)
plt.xlim(-4, 10)
plt.legend(('Logistic Regression Model', 'Linear Regression Model'),
           loc="lower right", fontsize='small')
plt.show()
```

Regressão Logística

Exercício

- Programa exemplo de regressão usando `scikit-learn`: **Logistic Function**



Referências

- Luiz E. S. Oliviera,
  Regressão,
  Notas de Aulas, DInf / UFPR, 2017.