

PÓS-GRADUAÇÃO

**Linguagens de
programação para
ciência de dados
(Python com Spark)**



PÓS-GRADUAÇÃO

Análise estatística dos dados


Bloco 1

Marcelo Tavares de Lima





► Objetivos

- Apresentar os principais métodos estatísticos que podem ser aplicados na área de ciência de dados.
 - Apresentar comandos em linguagem Python para análise e processamento de dados com métodos estatísticos.
 - Apresentar métodos de organização de dados.
- 



► **Análise estatística de dados**

- Conhecimentos básicos em estatísticas e probabilidade são extremamente importantes para trabalhar na área de ciência de dados.
- Por meio da estatística podemos extrair informação dos dados para obter melhor compreensão das situações que representam.
- A linguagem de programação Python é considerada uma das melhores linguagens de programação que fornece suporte para trabalhar com análise estatística dos dados.



► Biblioteca NumPy

- O NumPy é uma poderosa biblioteca do Python que é usada, principalmente, para realizar cálculos em vetores (arrays) e matrizes multidimensionais.
- Fornece um conjunto de funções que ajudam os programadores a executar de forma simples cálculos numéricos.
- Exemplos de comandos podem ser encontrados na Leitura Fundamental.




► Biblioteca Pandas

- É uma das ferramentas mais importante à disposição dos cientistas e analistas de dados que trabalham, atualmente, em Python.
- Os dados manipulados no Pandas são, frequentemente, usados para trabalhar com análises estatísticas no SciPy, plotando funções do Matplotlib e algoritmos de aprendizado de máquina no scikit-learn.

(McKINNEY, 2011)



► Biblioteca SciPy

- É o pacote principal de rotinas científicas em Python, que se destina a operar de forma eficiente em matrizes NumPy, segundo Jones (2001).
 - Utiliza como base a biblioteca Numpy para lidar eficientemente com grandes quantidades de números de maneira eficiente.
 - Contém um conjunto de funções que trabalham com estatísticas, pesquisa operacional, otimização, processamento de sinais e imagens, solução de equações diferenciais, polinômios etc.
- 



► Biblioteca Matplotlib

- Principal ferramenta de visualização bidimensional (2D) de gráficos científicos em Python.
- Seu criador John Hunter usou, principalmente, o MATLAB para visualização científica, mas como começou a integrar dados de fontes diferentes usando Python, percebeu que precisava uma solução Python para visualização, segundo Unpingco (2016).



► Biblioteca Statsmodels

- É um módulo Python que fornece classes e funções para a estimativa de muitos modelos estatísticos, bem como para a realização de testes e a exploração de dados estatísticos.
- É possível usar fórmulas da linguagem de programação R junto com os *dataframe* da biblioteca Pandas para ajustar seus modelos, segundo Seabold (2010).



► **Análise estatística com Python**

- Para nossos propósitos como cientistas de dados, você deve pensar em probabilidade e estatística como uma maneira de quantificar a incerteza associado a eventos escolhidos em algum universo de eventos e aprender utilizar os métodos estatísticos para entender e extrair informações dos dados.



► **Análise estatística com Python**

- Medidas estatísticas básicas:
 - Média aritmética simples.
 - Mediana.
 - Moda.
 - Variância.
 - Desvio padrão.

► Análise estatística com Python

- Código para calcular média aritmética simples.
- Vale lembrar que alguns comandos não funcionam na versão Python 3.

```
# -*- coding: utf-8 -*-
```

```
import scipy
```

```
import numpy as np
```

```
lista_idades = np.array([20, 49, 41, 33, 25, 10, 29, 40])
```

```
media = scipy.mean(lista_idades)
```

```
print "A média das idades é: ", round(media, 3)
```

```
A = [[1, 3, 27], [3, 4, 6], [7, 6, 3], [3, 6, 8]]
```

```
print "\nA média da matriz A é:", round(scipy.mean(A),3)
```

```
print "\nMédia de cada coluna da matriz A\n",
```

```
scipy.mean(A, axis = 0)
```

```
print "\nMédia de cada linha da matriz A\n",
```

```
scipy.mean(A, axis = 1)
```

► Análise estatística com Python

- Código para calcular média aritmética simples (saída).

A média das idades é: 30.875

A média da matriz A é: 6.417

Média de cada coluna da matriz A

[3.5 4.75 11.]

Média de cada linha da matriz A

*[10.33333333 4.33333333
5.33333333 5.66666667]*

► Análise estatística com Python

- Código para calcular mediana.

```
# -*- coding: utf-8 -*-  
import statistics  
import numpy as np  
  
A = [1, 3, 5, 7, 10, 15]  
  
print "A mediana do conjunto é: ",  
statistics.median(A)  
print "A mediana (baixa) do conjunto é: ",  
statistics.median_low(A)  
print "A mediana (alta) do conjunto é: ",  
statistics.median_high(A)
```



► Análise estatística com Python

- Código para calcular mediana (saída).
 - *A mediana do conjunto é: 6.0*
 - *A mediana (baixa) do conjunto é: 5*
 - *A mediana (alta) do conjunto é: 7*

► Análise estatística com Python

- Código para calcular moda.
- *# -*- coding: utf-8 -*-i*
- *import statistics*
- *import numpy as np*
- *A = [1, 3, 5, 5, 7, 10, 10, 10, 15]*
- *print "A moda do conjunto A é: ",
statistics.mode(A)*
- *B = [1, 3, 5, 5, 7, 10, 10, 15]*
- *print "A moda do conjunto do conjunto B: ",
statistics.mode(B)*
- *print "A moda da sequência de caracteres é: ",
statistics.multimode('AABBBBCCDDDEEEFFFGG'
)*

► Análise estatística com Python

- Código para calcular moda (saída).
- *A moda do conjunto A é: 10*
- *StatisticsError: no unique mode;
found 2 equally common values*
- *A moda da sequência de caracteres é:
['B', 'D', 'F']*

PÓS-GRADUAÇÃO

Análise estatística dos dados

Bloco 2

Marcelo Tavares de Lima





► **Análise estatística com Python (continuação)**

- Veremos comandos para calcular medidas de dispersão e elaboração de gráficos.

► Análise estatística com Python (continuação)

- Código para calcular desvio padrão.
- *import statistics*
- *import numpy as np*
- *idades = [10, 30, 55, 15, 17, 22, 38, 41, 15]*
- *print "O desvio padrão do conjunto de idades é: ", round(statistics.pstdev(idades), 3)*



► Análise estatística com Python (continuação)

- Código para calcular desvio padrão (saída).

o desvio padrão do conjunto de idades é: 14.189

► Análise estatística com Python (continuação)

- Código para calcular variância.

```
# -*- coding: utf-8 -*-
```

```
import statistics
```

```
import numpy as np
```

```
A = [0.0, 0.25, 0.25, 1.25, 1.5, 1.75, 2.75, 3.25]
```

```
print "A variância do conjunto de dados A é: ",  
round(statistics.variance(A),3)
```

► Análise estatística com Python (continuação)

- Código para calcular variância (saída).

A variância do conjunto de dados A é: 1.429



► Elaboração de gráficos: histograma

- Representação gráfica (barras verticais) da distribuição de frequências de um conjunto de dados.
- O primeiro passo para criar um histograma é sempre coletar dados. Utilizando Python é possível criar histogramas de diversas maneiras, são: Numpy, Pandas ou Matplotlib.

► Elaboração de gráficos: histograma

- Código para elaboração de Histograma.

```
# -*- coding: utf-8 -*-  
import numpy as np  
import matplotlib.mlab as mlab  
import matplotlib.pyplot as plt  
# mu = média da distribuição  
# sigma = desvio padrão da  
distribuição  
mu, sigma = 100, 15  
x = mu +  
sigma*np.random.randn(10000)  
#10.000 amostras
```

Continua...

► Elaboração de gráficos: histograma

- Código para elaboração de Histograma.

Continuação...

```
# histograma
```

```
n, bins, patches = plt.hist(x, 50, normed=1, facecolor='green',  
alpha=0.8)
```

```
y = mlab.normpdf( bins, mu, sigma)
```

```
l = plt.plot(bins, y, 'r--', linewidth=1)
```

Continua...

► Elaboração de gráficos: histograma

- Código para elaboração de Histograma.

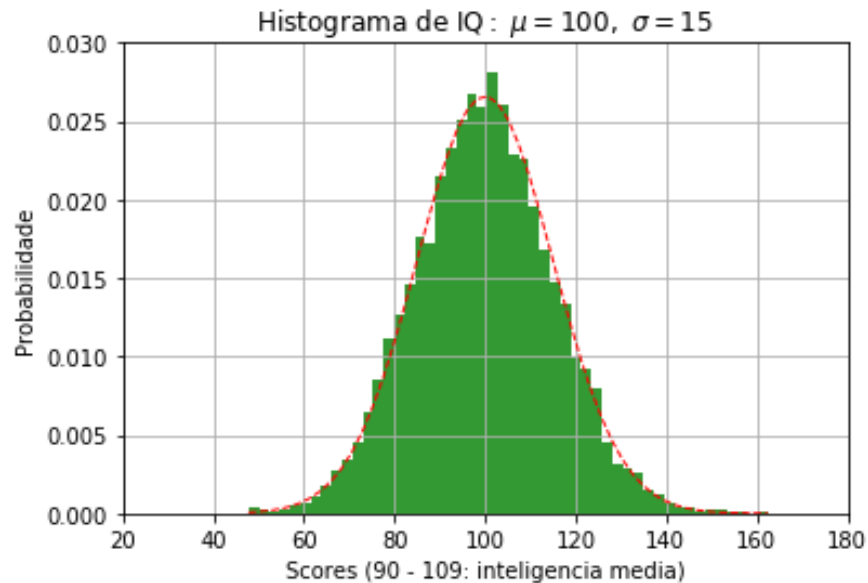
Continuação...

```
plt.xlabel('Scores (90 - 109: inteligencia media)')
plt.ylabel('Probabilidade')
plt.title(r'$\mathrm{Histograma\ de\ IQ:}\ \mu=100,\ \sigma=15$')
plt.axis([20, 180, 0, 0.03])
plt.grid(True)
plt.show()
```

► Elaboração de gráficos: histograma

- Saída para elaboração de histograma.

Figura 1 – Exemplo de histograma



Fonte: Pereira (2019)



► Correlação

- É uma análise descritiva que mede o grau de dependência entre duas variáveis, segundo Grus (2019) e Ben (2017).
- Os valores de correlação variam entre -1 e 1.
- Existem dois componentes principais de um valor de correlação:
 - Magnitude: quanto maior a magnitude (mais próxima de 1 ou -1), mais forte a correlação.
 - Sinal: se negativo, há uma correlação inversa. Se positivo, há uma correlação regular.



► Correlação

- **Correlação positiva** – na biblioteca Numpy, a função *corrcoef* que retorna uma matriz de correlações de x com x , x com y , y com x e y com y . Estamos interessados nos valores da correlação de x com y (então posição $(1, 0)$ ou $(0, 1)$).
- **Correlação negativa** - o que acontece com a correlação, se invertemos a correlação, de modo que um aumento em x resulte em uma diminuição em y ?



► Correlação

- **Sem correlação** – em alguns casos, pode não existir correlação entre as variáveis ou conjunto de dados. Em alguns casos, o valor da correlação será 0 (zero) ou um valor muito próximo de zero.

PÓS-GRADUAÇÃO

Teoria em prática


Bloco 3

Marcelo Tavares de Lima





► Análise de dados com Python

- Uma empresa de venda on-line (*e-commerce*) percebeu que suas vendas tiveram quedas em determinado período do ano.
 - Diante desse cenário, a empresa precisa descobrir as causas e tomar as devidas providências.
- 



► **Análise de dados com Python**

- Outras informações importantes que a empresa precisa saber são: quais são os meses em que as vendas tiveram as quedas, quais foram os meses mais produtivos, produtos mais vendidos e também as informações dos consumidores como: idade, sexo e localização.
- Utilizando os conceitos de estatística e probabilidades aprendidas nessa unidade, como você organizaria essas informações para a empresa?

Dica do professor

Bloco 4

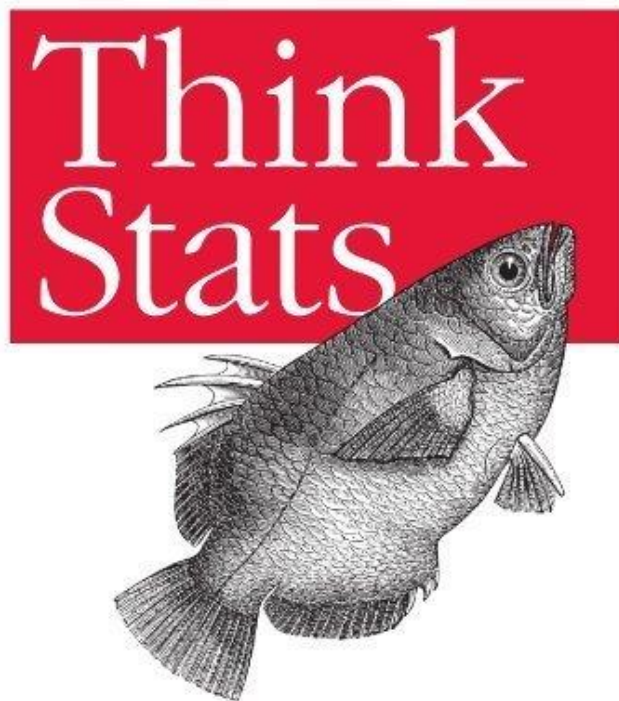
Marcelo Tavares de Lima



► Dica de livro

Figura 2 – Livro

Probability and Statistics for Programmers



O'REILLY®

Allen B. Downey

Think Stats: probability and statistics for programmers.

Autor: Allen B. Downey
(2011).

Editora: O'Reilly Media Inc.

Fonte: <https://www.amazon.com/Think-Stats-Allen-B-Downey/dp/1449307116>. Acesso em: 21 jan. 2020.



► Referências

BEN K. **Correlation in Python**. 2017. Disponível em: <http://benalexkeen.com/correlation-in-python/>. Acesso em: 16 jan. 2020.

DOWNEY, A. B. **Think stats**: probability and statistics for programmers. O'Reilly Media, Inc, 2011.

DUCHESNAY, E.; LÖFSTEDT, T.; FEKI Y. **Statistics and machine learning in Python**. 2018.

GRUS, J. **Data science from scratch**: first principles with Python. O'Reilly Media, 2019.

JONES, E.; OLIPHANT, T., PETERSON, P. **SciPy**: open source scientific tools for Python, 2001.





► Referências

McKINNEY, W., **Pandas**: a foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing, 2012.

PEREIRA, D. R. **Linguagens de programação para ciência de dados** (Python com Spark). Londrina: Editora e Distribuidora Educacional S.A., 2019

SEABOLD, S, JOSEF P. **Statsmodels**: econometric and statistical modeling with python. Proceedings of the 9th Python in Science Conference. 2010.

UNPINGCO, J. **Python for probability, statistics, and machine learning**. Springer International Publishing”, 2016.

