Oficina Prática de Ciência de Dados

(em Python)

Flavio Figueiredo - DCC/UFMG
@flaviovdf

Créditos e Referências

- Statistics for Hackers
  - Jake VanderPlas (vanderplas.com)
  - https://speakerdeck.com/pycon2016/jake-vanderplas-statistics-for-hackers
    http://christopherroach.com/articles/statistics-for-hackers

- Data 8: The Foundations of Data Science
  - Curso de Berkeley
  - http://data8.org/

- Practical Data Science
  - Carnegie Mellon
  - http://datasciencecourse.org

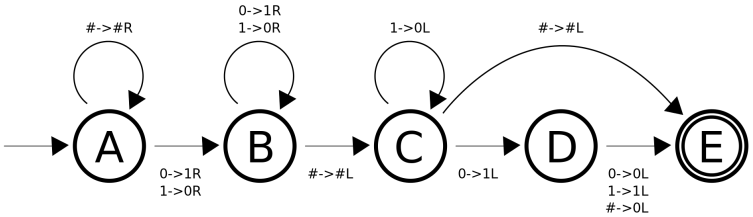
Ciência de Dados

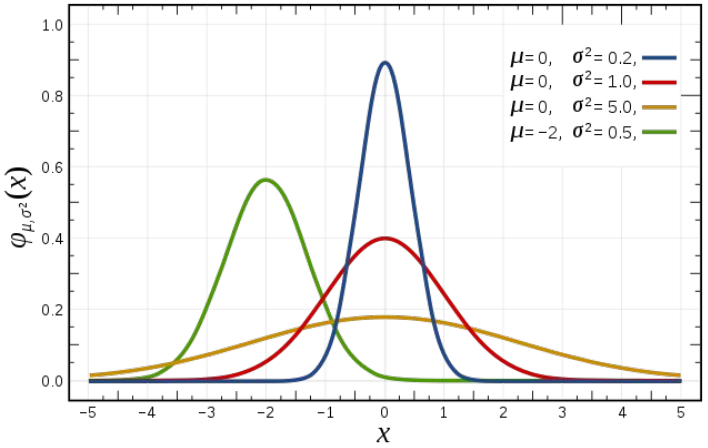
Ciência de Dados

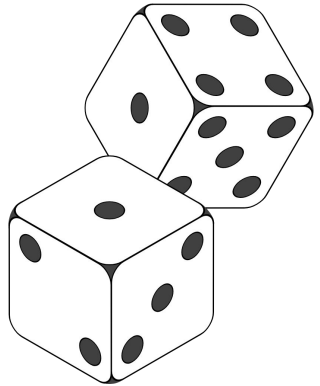
Aplicação de computação e estatística para entender fenômenos do mundo real.

Ciência de Dados

Aplicação de **computação** e **estatística** para entender fenômenos do **mundo real**.

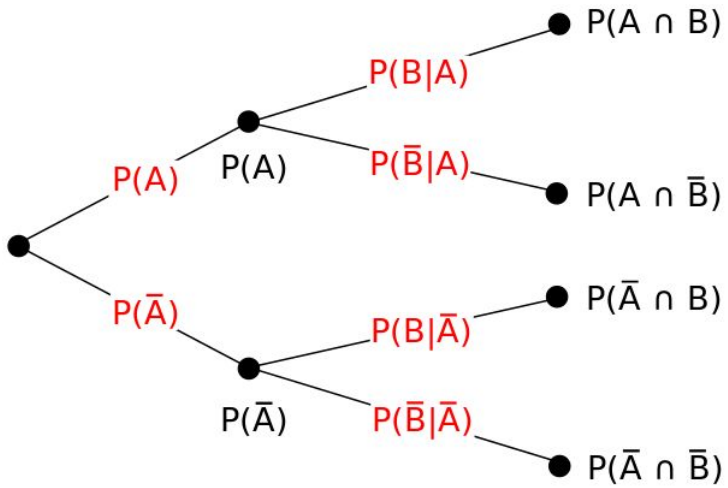






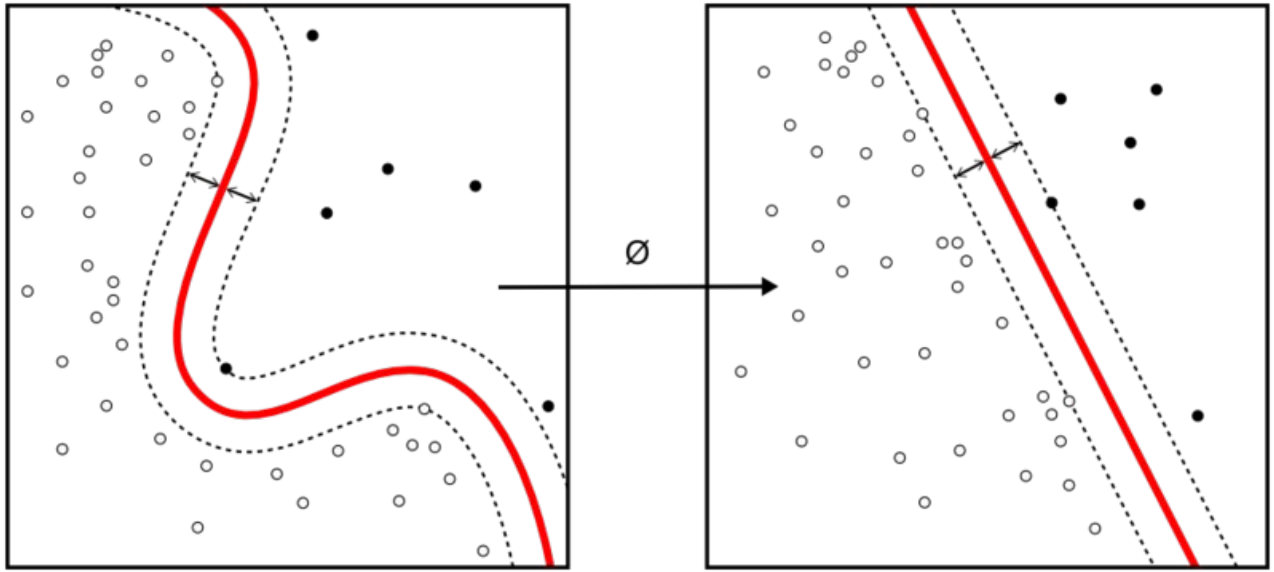
Probabilidade

Ciência de dados **não** é probabilidade linear. Usa probabilidade.



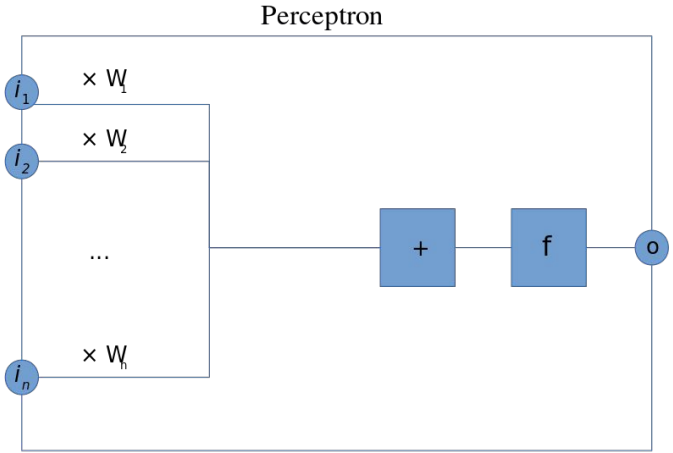
Álgebra Linear

Ciência de dados **não** é álgebra linear. Usa álgebra linear.

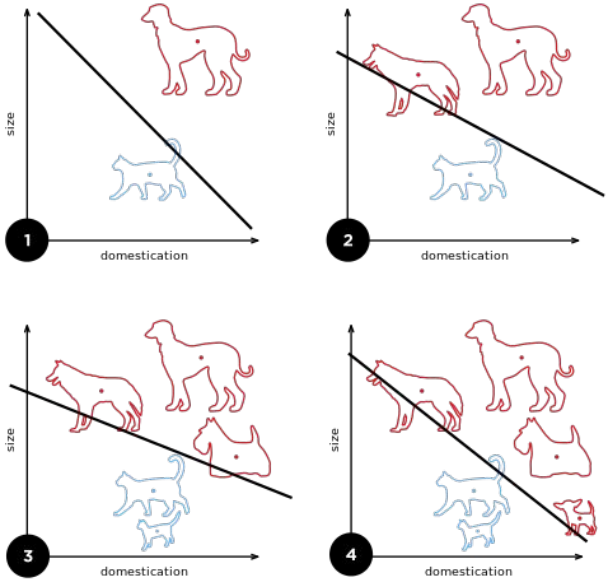


Aprendizado de Máquina

Ciência de dados **não** é aprendizado de máquina. Usa aprendizado de máquina.



$$o = f\left(\sum_{k=1}^{n} i_k \cdot W_k\right)$$



Inteligência Artificial

Ciência de dados **não** é inteligência artificial. Como também não usa :)

Ciência de Dados

- Data science → conhecimento sobre os dados/mundo
- Machine learning → previsões
- Artificial intelligence → ações

Estatística

Então é estatística?

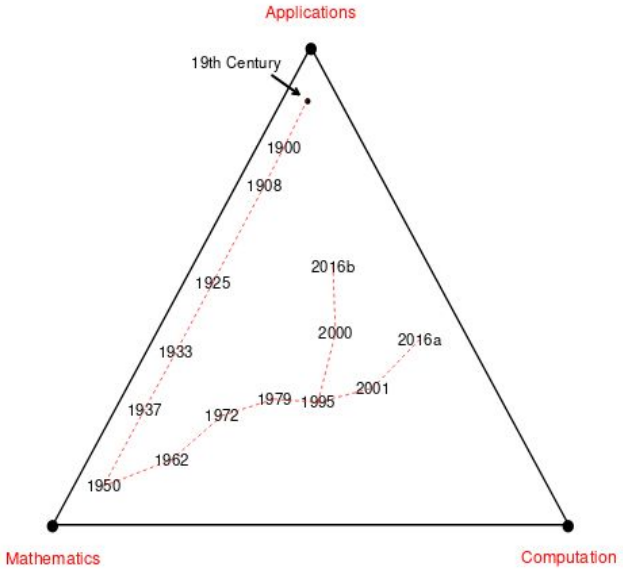
Estatística vs. Ciência de Dados

Diferença foi surgindo ao longo dos anos. . .

- Com o passar dos anos, foi ficando próxima da computação
  - Grandes massas de dados surgiram

- A computação aplicou o conhecimento estatístico para entender as mesmas
  - O pensamento computacional é chave!

Historicamente



Development of the statistics discipline since the end of the nineteenth century, as discussed in the text.

Receita

1. **Boa:** Computação
2. **Boa:** Estatística

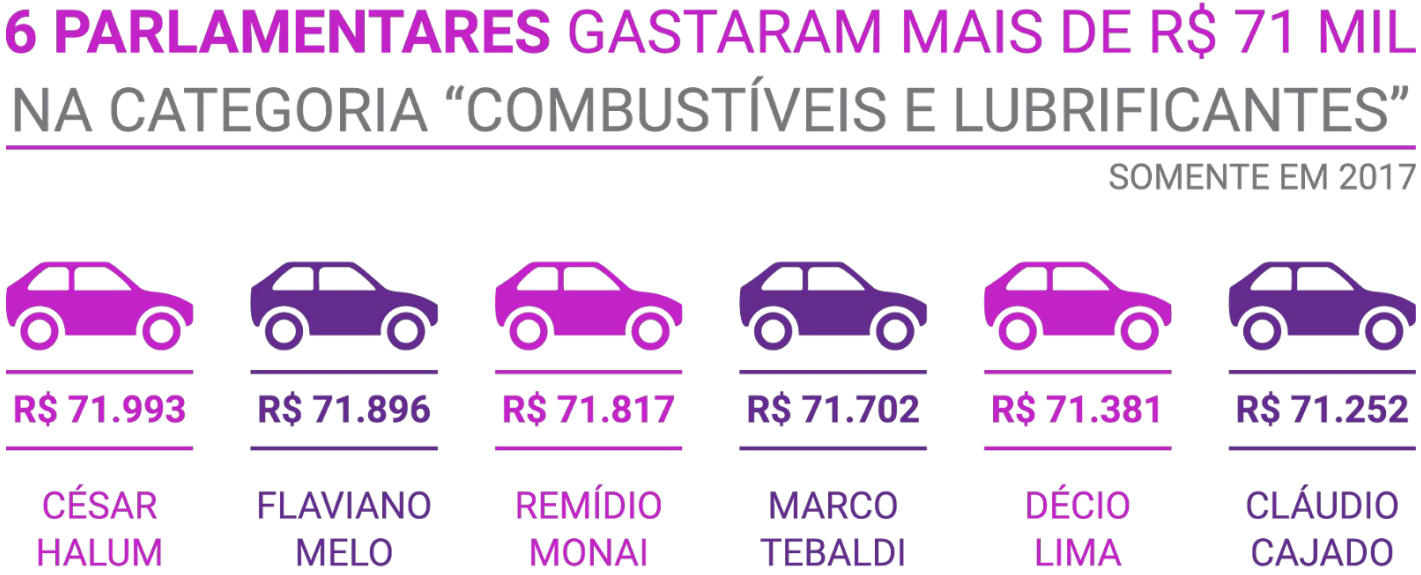
3. **Entendimento e uso:** Aprendizado de máquina

4. **Entendimento:** Probabilidade
5. **Entendimento:** Álgebra Linear

Principal: Ótimas Perguntas!

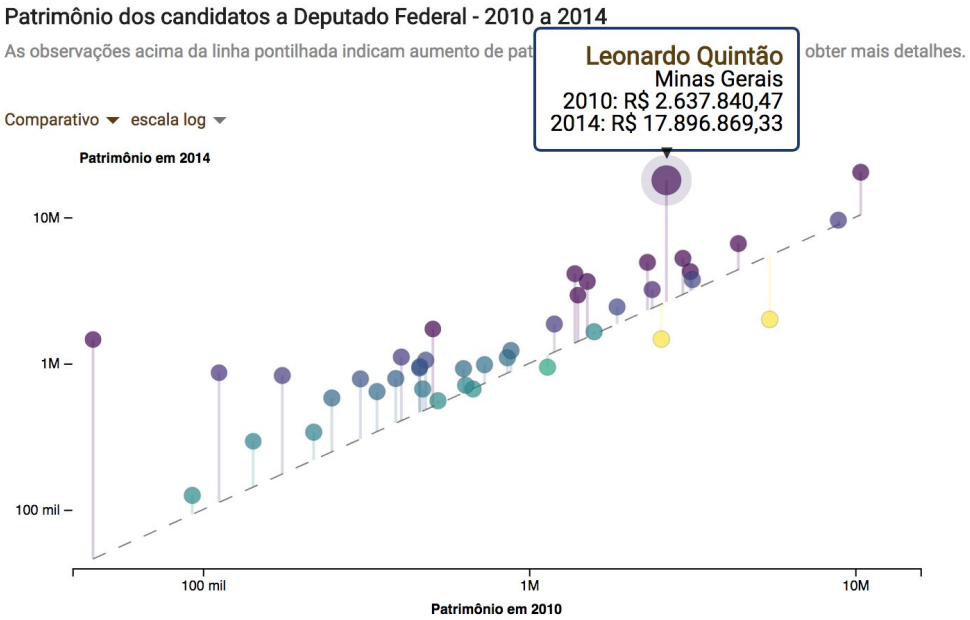
Serenata de Amor

https://serenata.ai/stories/



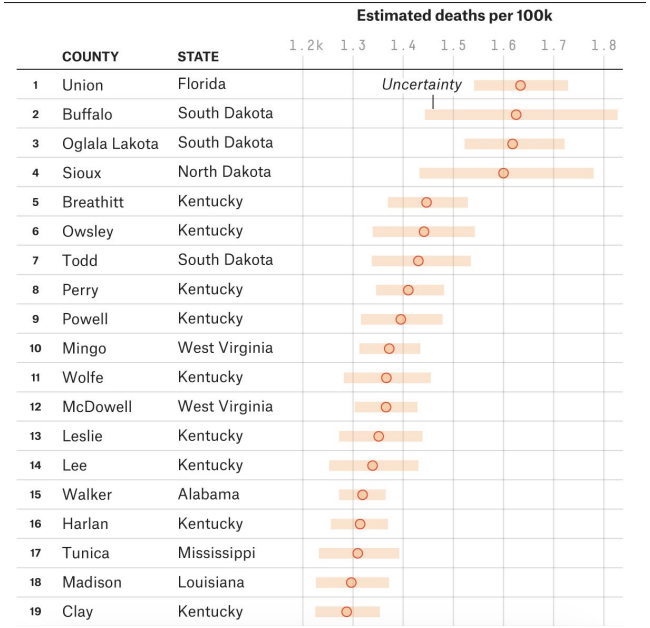
Capital dos Candidatos

http://www.capitaldoscandidatos.info/



Five Thirty Eight

https://projects.fivethirtyeight.com/mortality-rates-united-states/



	COUNTY	STATE
1	Union	Florida
2	Buffalo	South Dakota
3	Oglala Lakota	South Dakota
4	Sioux	North Dakota
5	Breathitt	Kentucky
6	Owsley	Kentucky
7	Todd	South Dakota
8	Perry	Kentucky
9	Powell	Kentucky
10	Mingo	West Virginia
11	Wolfe	Kentucky
12	McDowell	West Virginia
13	Leslie	Kentucky
14	Lee	Kentucky
15	Walker	Alabama
16	Harlan	Kentucky
17	Tunica	Mississippi
18	Madison	Louisiana
19	Clay	Kentucky

A Oficina

1. Entender o mínimo de uma linguagem de programação

2. Falar de alguns termos estatísticos

3. Usar o mínimo da computação para entender os termos
    a. De uma forma bem simplificada

4. **Por fim: Espero que tenham um pouco de entendimento de como as duas áreas se encontram**

Básico de Programação

Python

Língua franca de ciência de dados

Python

https://docs.scipy.org/doc/numpy-1.10.0/user/c-info.python-as-glue.html

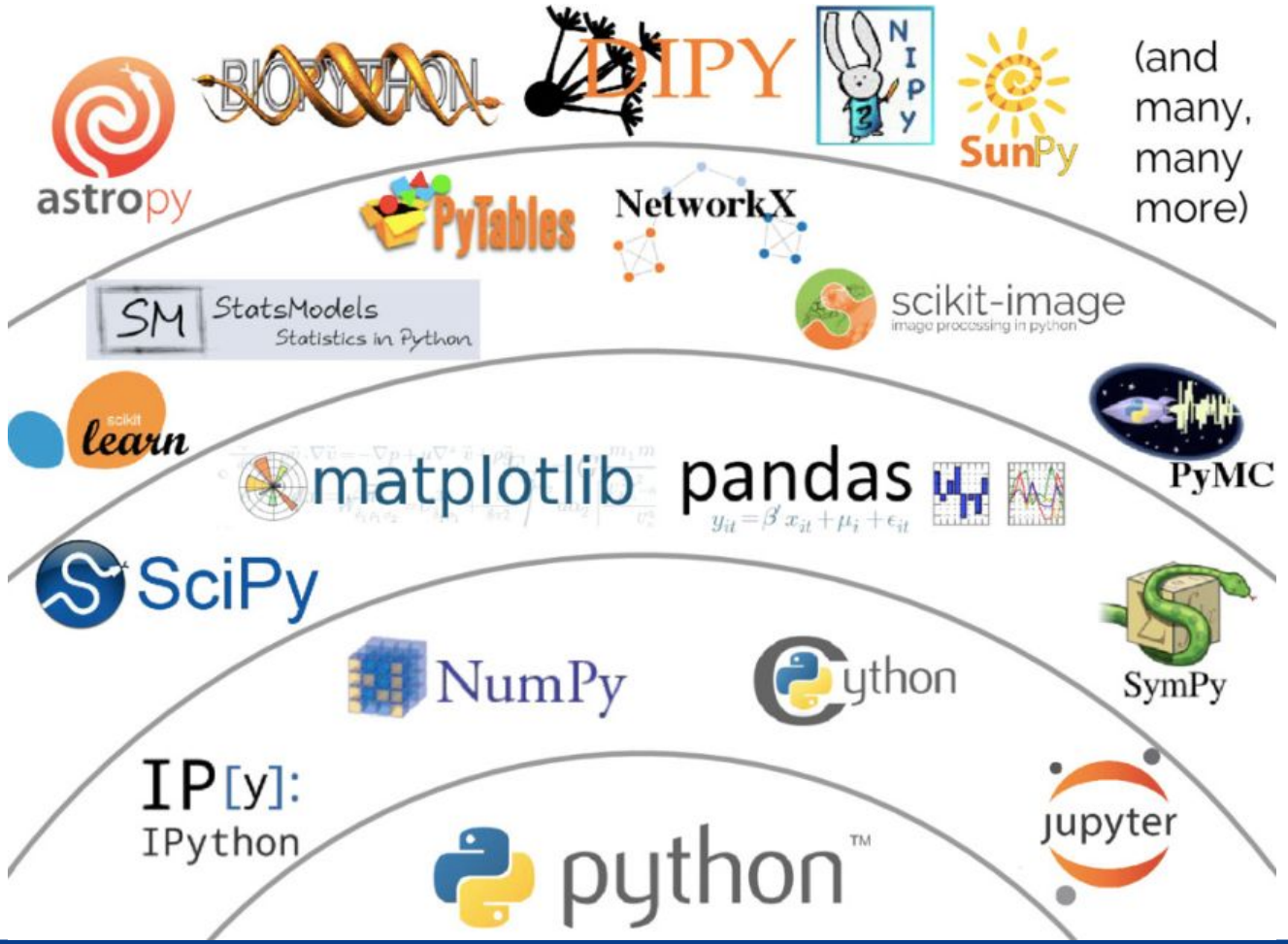
Python

https://docs.scipy.org/doc/numpy-1.10.0/user/c-info.python-as-glue.html

Alternativas:
- R/Julia/Matlab etc.

São de um uso mais específico.

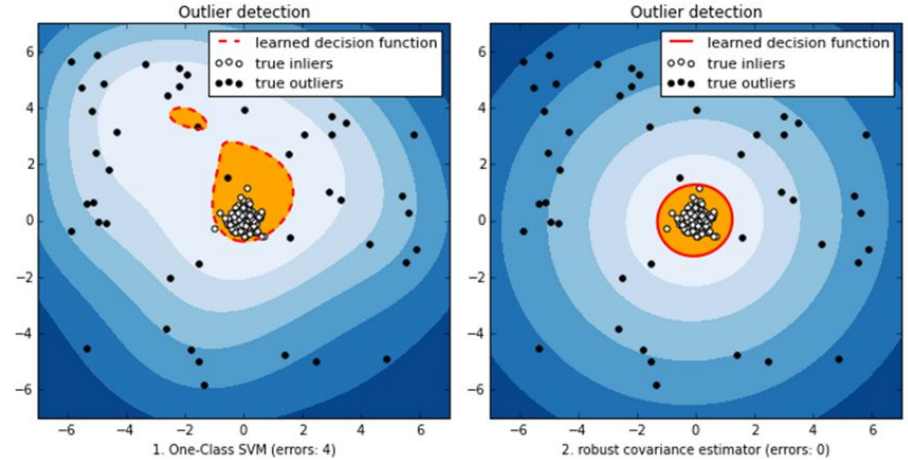
Python Data Science Stack



Jupyter Hands On


```
            [a.collections[0], b, c],
            ['learned decision function', 'true inliers', 'true outliers'],
            prop=matplotlib.font_manager.FontProperties(size=11))
        subplot.set_xlabel("%d. %s (errors: %d)" % (i + 1, clf_name, n_errors))
        subplot.set_xlim((-7, 7))
        subplot.set_ylim((-7, 7))
    plt.subplots_adjust(0.04, 0.1, 0.96, 0.94, 0.1, 0.26)

plt.show()
```



Laço for

```
for i in range(10)
```

Laço for

```
for i in range(100)
```

Laço for

```
for i in range(2)
```

Laço for

```
for i in range(2)
    jogue uma moeda para cima
    ou
    troque um valor do grupo a com o grupo b
```

Laço for

```
for i in range(2)
    jogue uma moeda para cima
    ou
    troque um valor do grupo a com o grupo b
    ou
    leia um valor do grupo
```

Se você sabe escrever um laço você consegue entender alguns conceitos da estatística

Moedas

- Vamos supor que você jogue uma moeda para cima 30 vezes
- A mesma cai em cara 22 vezes
- A moeda é justa?
  - Não viesada

Dois lados do argumento

A brincadeira é assumir que alguém está certo

- Depois mostramos que a chance de tal pessoa está certa é muito baixa

- Como?

A brincadeira é assumir que alguém está certo

- Depois mostramos que a chance de tal pessoa está certa é muito baixa
- Como?
  - Fazendo algo similar ao Batman
- Gerar uma hipótese nula

Moedas

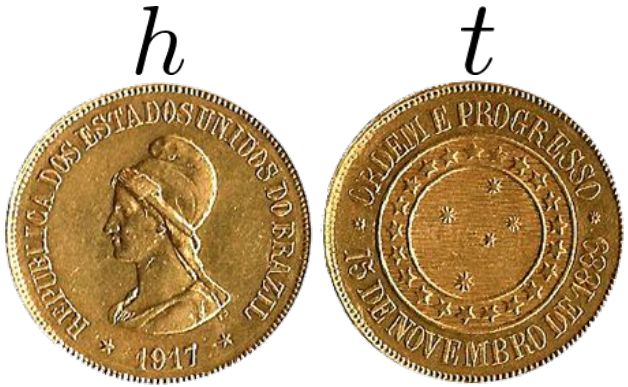
- Vamos supor que você jogue uma moeda para cima 30 vezes

- Qual é a probabilidade de sair 22 caras?
  - Vamos modelar o problema corretamente

Moedas

- Vamos supor que você jogue uma moeda para cima 30 vezes

- Qual é a probabilidade de sair 22 caras?
  - Vamos modelar o problema corretamente



$$\mathcal{S} = \{h, t\}$$

$$P(h) = \frac{1}{2}$$

$$P(t) = \frac{1}{2}$$

Moedas

- **Se for só 3 caras em 4 sorteios?**

$$hhhh \quad \underline{hthh} \quad \underline{thhh} \quad tthh$$
$$\underline{hhht} \quad htht \quad thht \quad ttht$$
$$\underline{hhth} \quad htth \quad thth \quad ttth$$
$$hhtt \quad httt \quad thtt \quad tttt$$

Podemos enumerar todos os casos. 25%

Binomial

- O mesmo pode ser modelado com uma distribuição binomial

$$Pr(k;n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

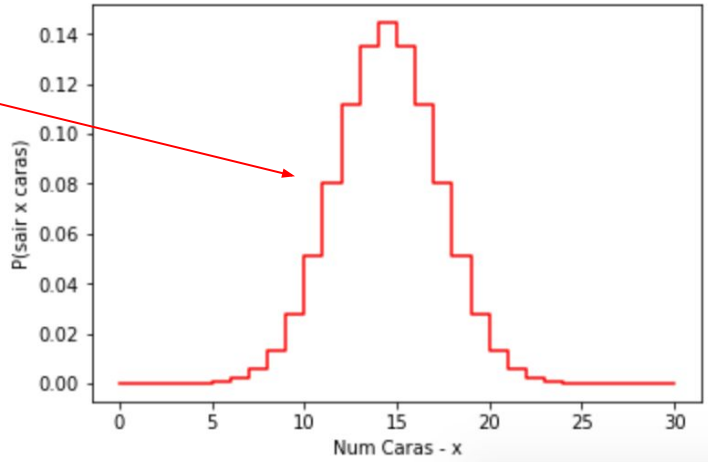
$$Pr(k;n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$Pr(k;n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

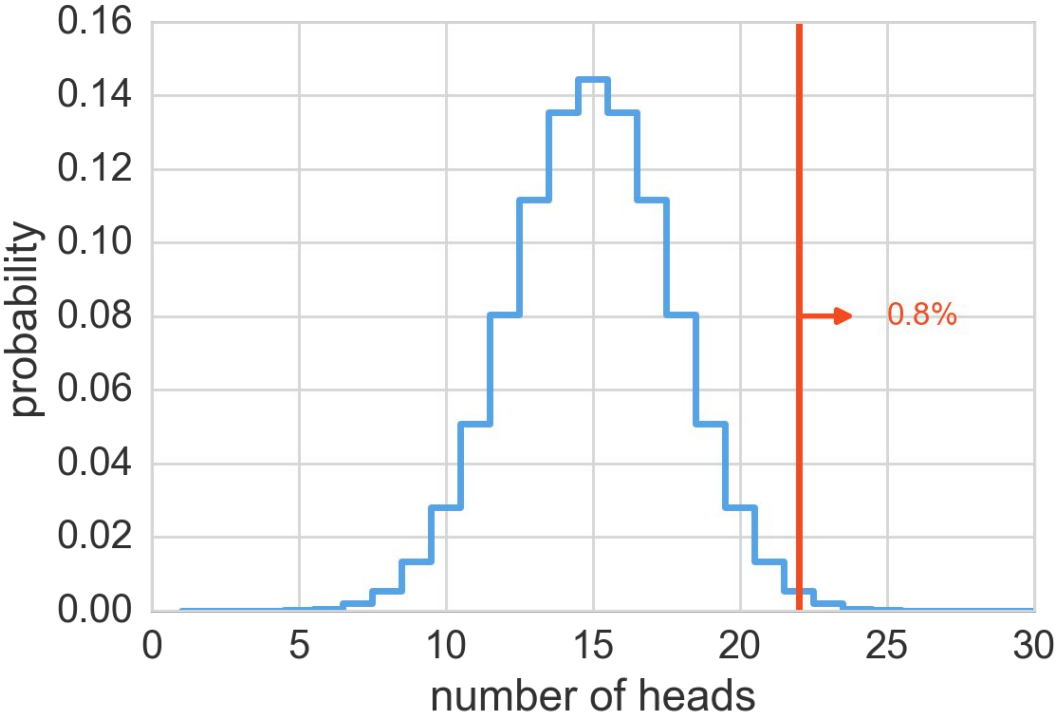
```
In [4]: p = 0.5 # probabilidade de heads/tails
        n = 30  # temos 33 jogadas
        x = np.arange(0, 31)
        prob_binom = ss.distributions.binom.pmf(x, n, p)
        plt.step(x, prob_binom, 'r-')
        plt.xlabel('Num Caras - x')
        plt.ylabel('P(sair x caras)')

Out[4]: <matplotlib.text.Text at 0x1123655c0>
```

Uma moeda justa!



Teoricamente: Probabilidade de valores >= 22



Abordando o problema através de laços

- Eu pessoalmente não sei a fórmula da binomial decorada
- Mas sei jogar uma moeda para cima
- Vamos fazer várias vezes!

Jogar moedas para cima é simples

```
In [8]: # Jogando uma única moeda
        np.random.randint(2)
```

```
Out[8]: 1
```

```
In [9]: # Jogando 30 moedas
        np.random.randint(2, size=30)
```

```
Out[9]: array([1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0,
               1, 1, 1, 0, 1, 1, 1])
```

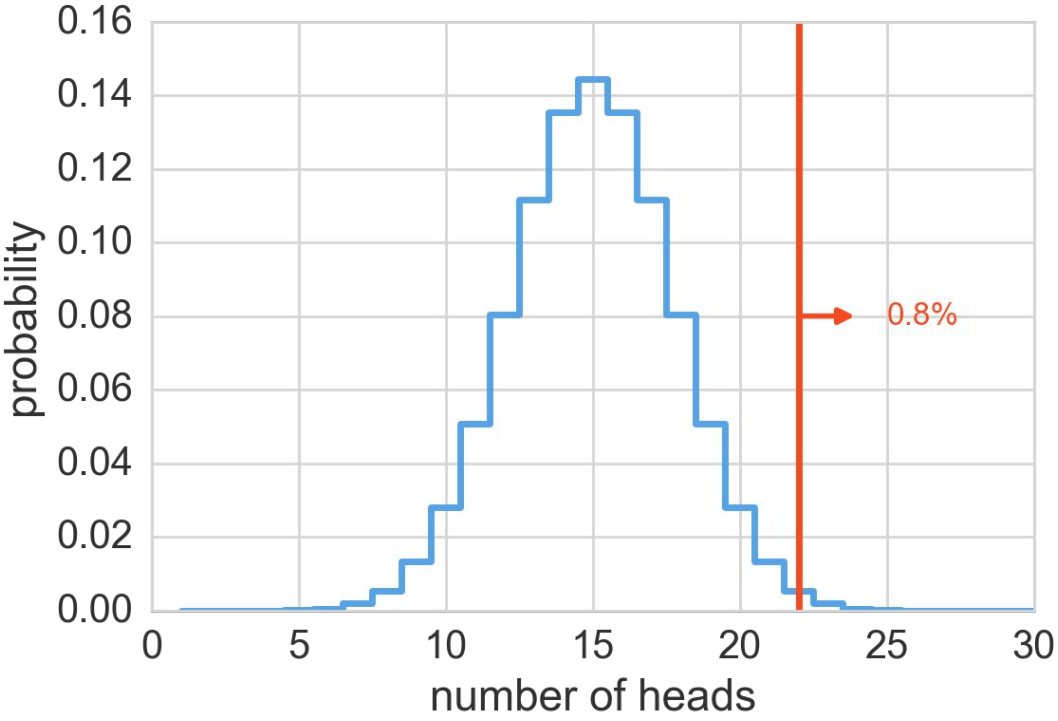
Vamos jogar um monte!

Na verdade, vamos jogar 30 moedas 10000x

In [10]:
```
NUM_SIMULACOES = 100000
resultados = 0
n = 30
for i in range(NUM_SIMULACOES):
    jogadas = np.random.randint(2, size=n) # joga 30 moedas para cima
    n_caras = (jogadas == 1).sum()         # conta quantas foram == 1, caras
    if n_caras >= 22:
        resultados += 1                    # quantas vezes vi >= 22 caras
print(resultados / NUM_SIMULACOES)
```

```
0.00833
```

Chegamos no mesmo valor



Simular não é difícil!
É computação.

Brincando com dados reais.
NBA Salaries

Até Agora

- Tudo é bem bacana, mas jogar moedas é simples
- Vamos pensar em outro caso.
- Salários da NBA

Dados

- O código tem um pouco de magia Pandas para filtrar
- Resumindo, estou apenas pegando os dois times de interesse

```
In [2]: df = pd.read_csv('data/nba_salaries.csv')
        df = df[df['TEAM'].isin(['Houston Rockets', 'Cleveland Cavaliers'])]
```

```
In [3]: df
```

Out[3]:

	PLAYER	POSITION	TEAM	SALARY
72	LeBron James	SF	Cleveland Cavaliers	22.970500
73	Kevin Love	PF	Cleveland Cavaliers	19.689000
74	Kyrie Irving	PG	Cleveland Cavaliers	16.407501
75	Tristan Thompson	C	Cleveland Cavaliers	14.260870
76	Brendan Haywood	C	Cleveland Cavaliers	10.522500
77	Iman Shumpert	SG	Cleveland Cavaliers	8.988765
78	Timofey Mozgov	C	Cleveland Cavaliers	4.950000
79	Mo Williams	PG	Cleveland Cavaliers	2.100000
80	Sasha Kaun	C	Cleveland Cavaliers	1.276000
81	Matthew Dellavedova	PG	Cleveland Cavaliers	1.147276
131	Dwight Howard	C	Houston Rockets	22.359364
132	James Harden	SG	Houston Rockets	15.756438
133	Ty Lawson	PG	Houston Rockets	12.404495

O salário médio do Cleveland é maior do que o salário médio do Houston?

Qual é o problema?

- Comparar dois salários médios.

$ 10.23M

$ 7.10M

A forma clássica de responder...

1. Compute o valor *t*

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s_{\bar{x}_1 - \bar{x}_2}}$$

A forma clássica de responder...

2. Compute *s*

$$s_{\bar{x}_1 - \bar{x}_2} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

A forma clássica de responder...

3. Determine o número de graus de liberdade *v*

$$\frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

A forma clássica de responder...

3. Determine o número de graus de liberdade *v*

$$\frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

A forma clássica de responder...

4. Determine uma significância.
    a. Essa é fácil, o professor sempre fala 5%

A forma clássica de responder...

5. Agora podemos estimar uma distribuição t-student

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Tem forma mais simples

6. Baixe uma tabelinha da Internet
    a. http://www.sjsu.edu/faculty/gerstman/StatPrimer/t-table.pdf

t Table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646

t Table

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$				
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025				
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05				
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71				
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303				
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182				
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776				
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571				
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447				
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365				
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
1											
1											
1											
2											
2											
2											
2											
2											
2											
2											
2											
2											
3											

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s_{\bar{x}_1 - \bar{x}_2}}$$

$$f(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

E ae...

Ninguém nem lembra do problema original.

E ae...

Ninguém nem lembra do problema original.

Estou exagerando, tem forma mais simples de fazer isso. Inclusive, o Scipy/R/Julia faz tudo isso com poucas linhas.

Nós vamos simular para aprender.

Qual era o problema?

- Comparar dois salários médios.

$ 10.23M

$ 7.10M

Qual é o problema?

- Comparar dois salários médios.

$ 9.12M

$ 9.10M

Qual é o problema?

- Comparar dois salários médios.

$ 10.1M

$ 8.9M

Qual é o problema?

- Comparar dois salários médios.

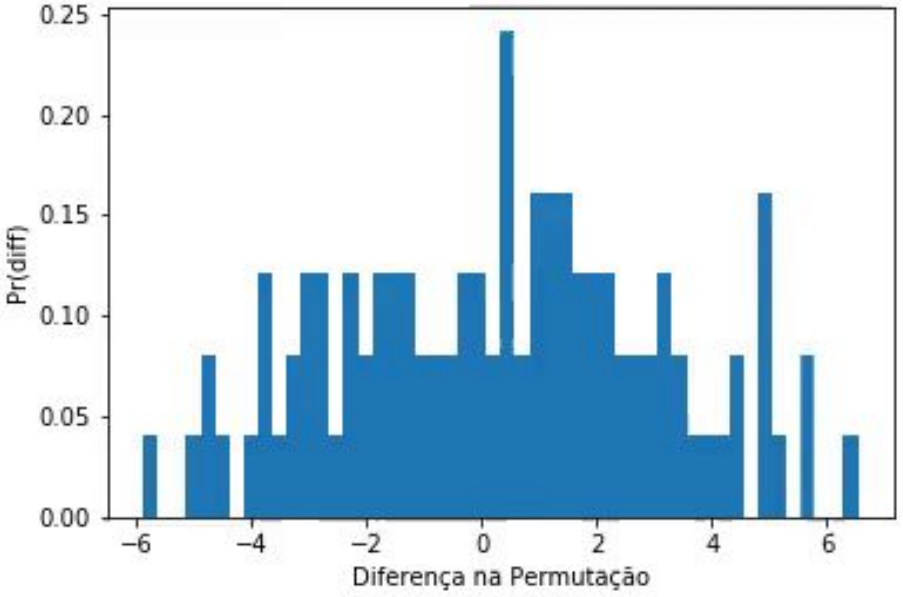
$ 9.14M

$ 9.9M

Repetindo muitas vezes...

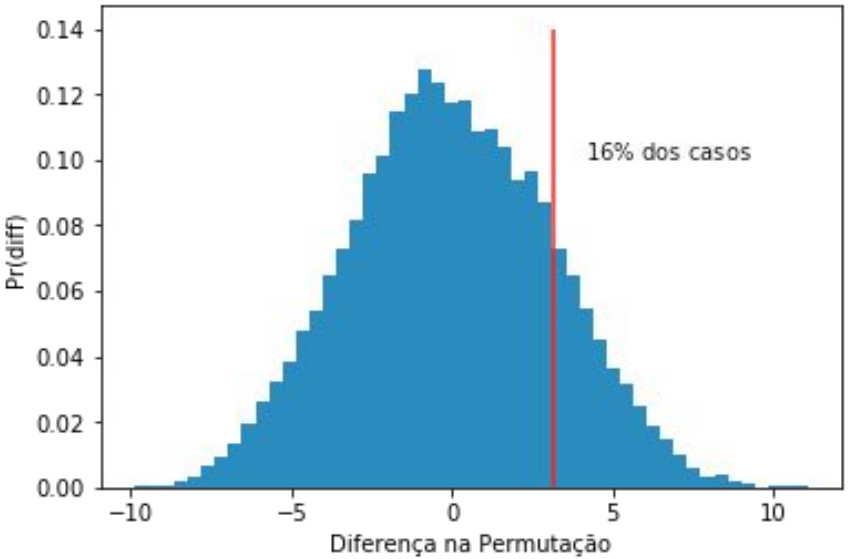
In [9]:
```
N = 10000
diferencas = np.zeros(N)
for i in range(N):
    np.random.shuffle(filtro.values)
    diff = df[~filtro]['SALARY'].mean() - df[filtro]['SALARY'].mean()
    diferencas[i] = diff
```

Repetindo muitas vezes...



Em algum momento estabiliza

- 16% dos casos são maiores do que a diferença real
- Isso é muito!
- Não existe diferença média entre os times
  - É aleatório!



Intervalos de Confiança

Problema

- Qual o salário médio de um jogador da NBA?

Problema

- Qual o salário médio de um jogador da NBA?
- Qual a incerteza de tal estimativa?

Intervalos de Confiança

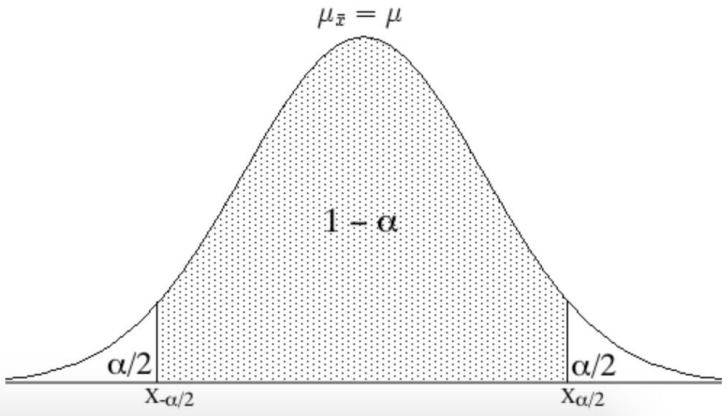
- Se você abrir um livro de estatística
  - Ou o Wikipedia

Intervalo de confiança para a média de uma população [editar | editar código-fonte]

Seja uma população de média μ e desvio padrão σ , da qual se toma amostras de n elementos. Cada uma das amostras tem média $\bar{x}$, sendo que a média de todas as amostras significativas coincide com a média da população $\mu_{\bar{x}} = \mu$.[44][45][46] Se o tamanho da amostra for suficientemente grande, a distribuição amostral segue praticamente uma distribuição normal (distribuição de Gauss) com média μ e desvio padrão $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$. Isto é representado como $\bar{X} \sim N(\mu, \frac{\sigma}{\sqrt{n}})$. Padronizando, tem–se $\frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} = Z \sim N(0,1)$.[47]

Com $Z \sim N(0,1)$, pode–se calcular um intervalo de confiança dentro do qual pode conter uma determinada porcentagem de observações. É possível encontrar Z_1 e Z_2, tal que $P(Z_1 \leq Z \leq 2) = 1 - \alpha$, em que $(1 - \alpha) \times 100$ é o porcentual desejado. Com μ, $P(\mu_1 \leq \mu \leq \mu_2) = 1 - \alpha$. Nesta distribuição normal, pode–se calcular o intervalo de confiança en que a população significativa apenas pode ser encontrada se uma amostra conhecida com média $\bar{x}$ tiver uma certa confiança. Normalmente, os valores entre 95% e 99% são comuns. Estes valores serão chamados de $1 - \alpha$. Isto exige o cálculo de $Z_{\frac{\alpha}{2}}$ ou do valor crítico junto com sua distribuição oposta $X_{\frac{-\alpha}{2}}$. [44][45][46]

Estes pontos definem a probabilidade do intervalo de tempo como mostra a figura a seguir.



Intervalos de Confiança

- Se você abrir um livro de estatística
  - Ou o Wikipedia
- Assumindo uma população
  - Jogadores da NBA
- Ao gerar amostras de tamanho S
  - S = 100
- Onde caiem 95% dos salários médios de diferentes amostras de tamanho S

Novamente

População

Amostra

Novamente

População

Amostra

Novamente

População

Amostra

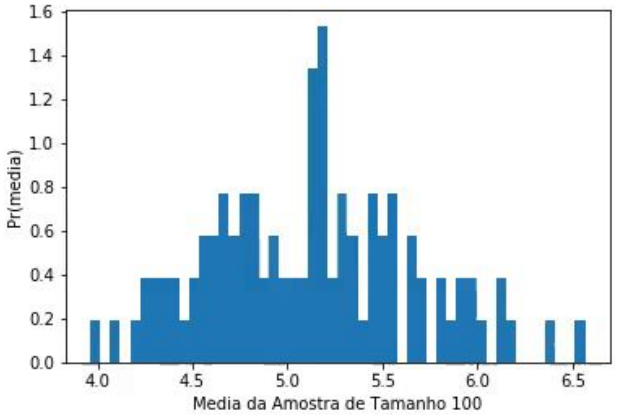
Novamente

População

Amostra

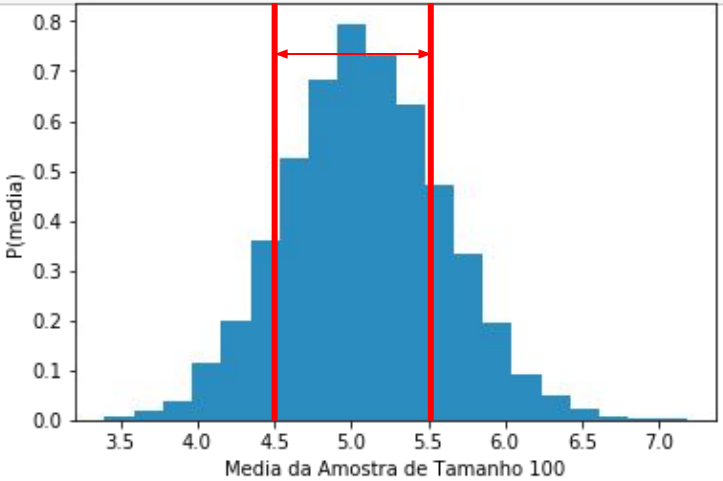
Novamente com um código simples

In [31]:
```
S = 100
N = 10000
values = np.zeros(N)
for i in range(N):
    sample = np.random.choice(df.index, replace=True, size=S) # Escolhe 100 elementos
    values[i] = df['SALARY'][sample].mean()
```



Novamente com um código simples

In [31]:
```
S = 100
N = 10000
values = np.zeros(N)
for i in range(N):
    sample = np.random.choice(df.index, replace=True, size=S) # Escolhe 100 elementos
    values[i] = df['SALARY'][sample].mean()
```



Bootstrap

1. Gera amostras com reposição
    a. Tamanho S
2. Computa métrica de interesse
3. Guarda e repete
    a. N vezes

Bootstrap Falha em Alguns Casos

- Funciona bem em alguns casos
  - Média e Mediana
- Falha em algumas distribuições
  - Leis de potência
- Dados temporais

Aprendizado

Se você entender isso eu fico feliz...

- Modelamos o mundo através de distribuições de probabilidade
- Porém, avaliar e entender tais distribuições pode ser complicado
- Simular as mesmas nem tanto

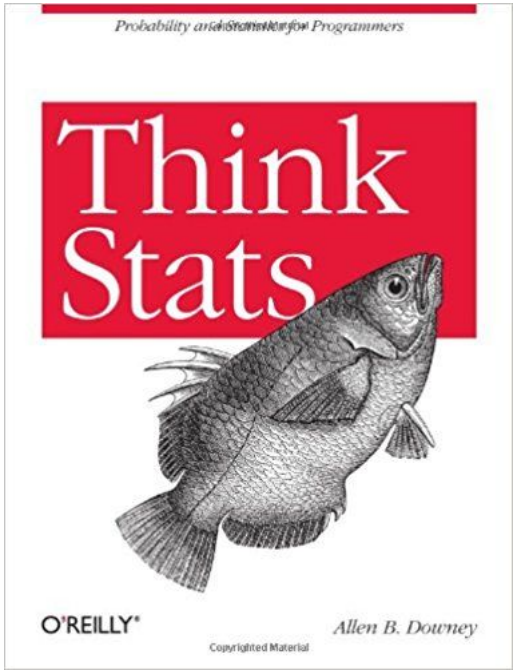
WARNING

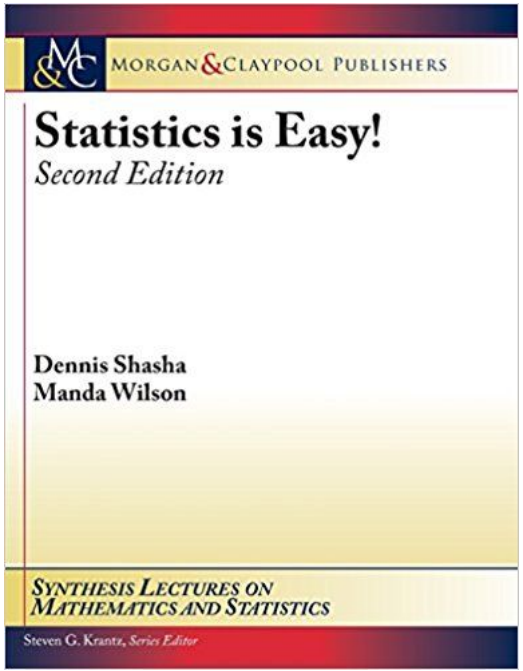
- A palestra aqui é para expor conceitos

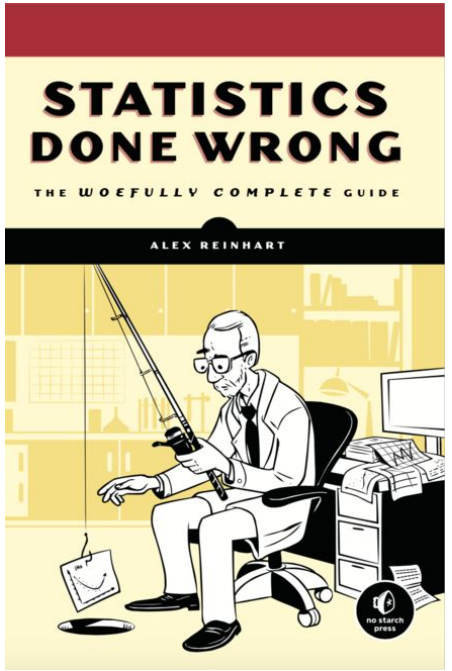
- Os métodos só funcionam com algumas premissas
  - Amostras representativas
  - Focamos em médias no geral
  - Dados bem comportados

Mais Referências (Livros)

Se você quer aprender conceitos por alto e não cometer erros

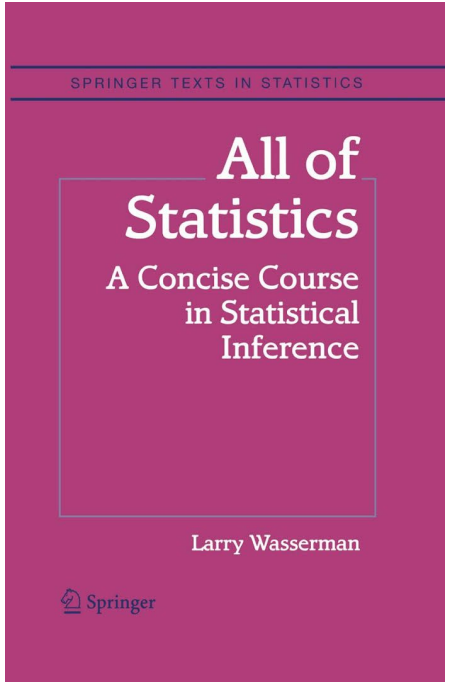


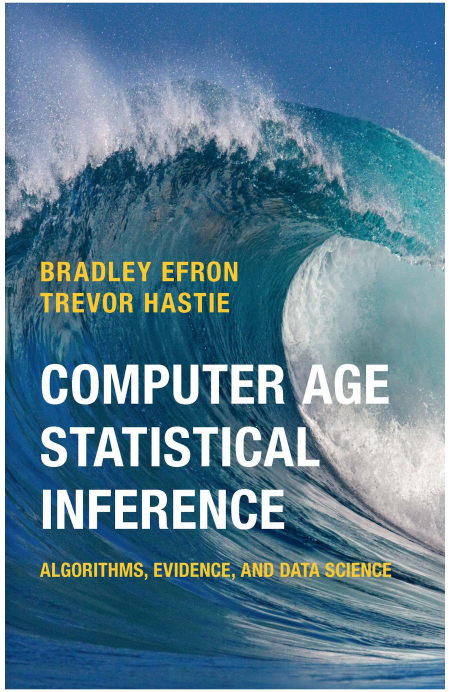


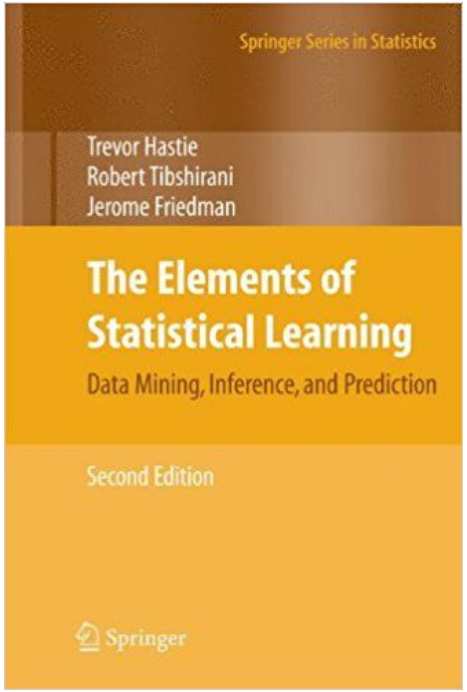


Referências (Livros "Melhores")

os 2 últimos estão disponíveis na web







Preciso de 6 livros?!

- Sendo bem sincero, só deu uma olhada nos dois primeiros
  - Think Stats e Statistics is Easy
- De qualquer forma:
  - Existe "uma moda" hoje em dia em resumir conceitos estatísticos com programas simples
  - A palestra apresentada tem culpa nesse aspecto
- O importante é que o aluno entenda os conceitos, possa aplicar
- Depois ele se aprofunde nos últimos livros

Obrigado!