INFERÊNCIA ESTATÍSTICA E DATA MINING

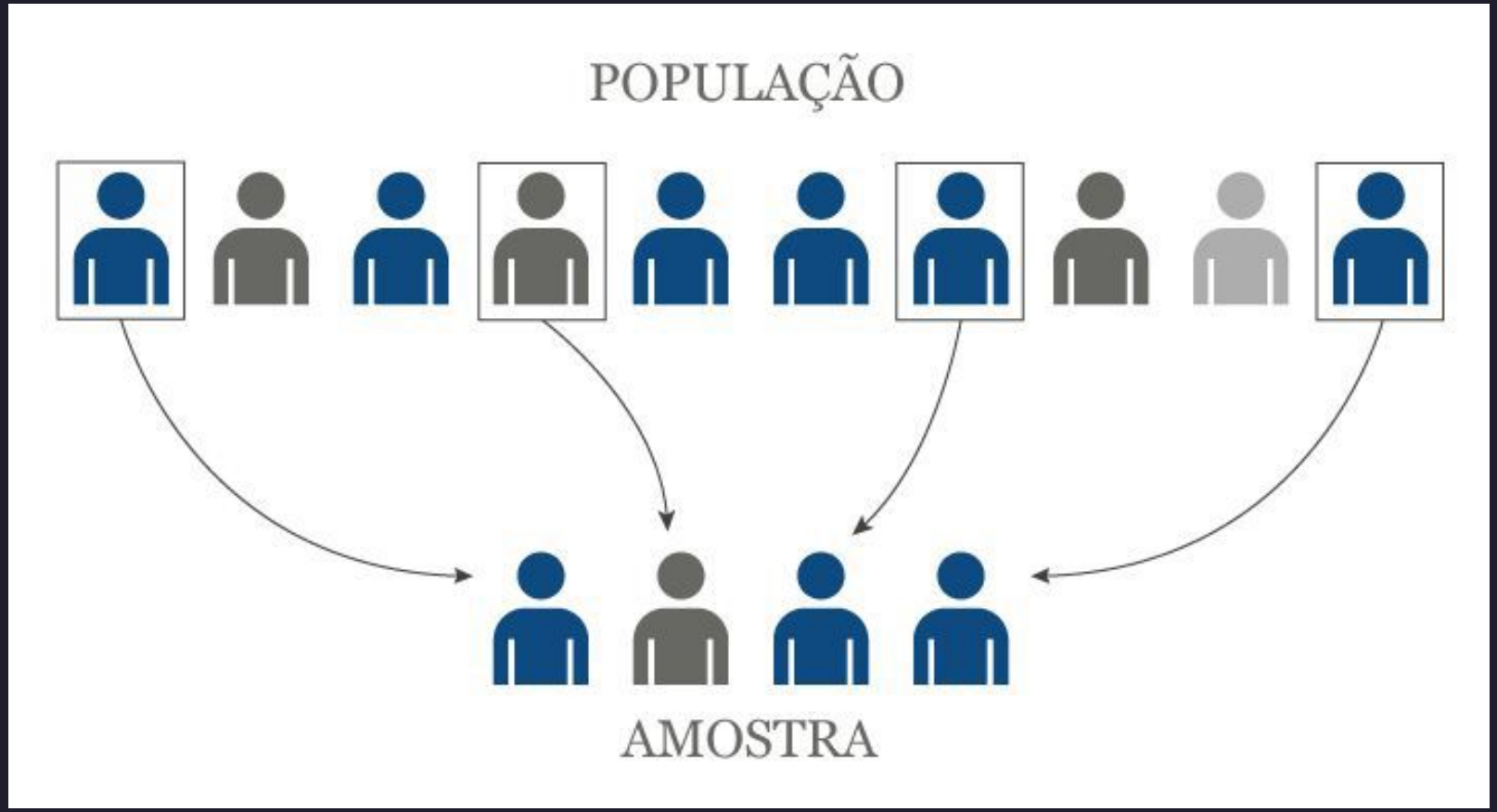
AULA 03

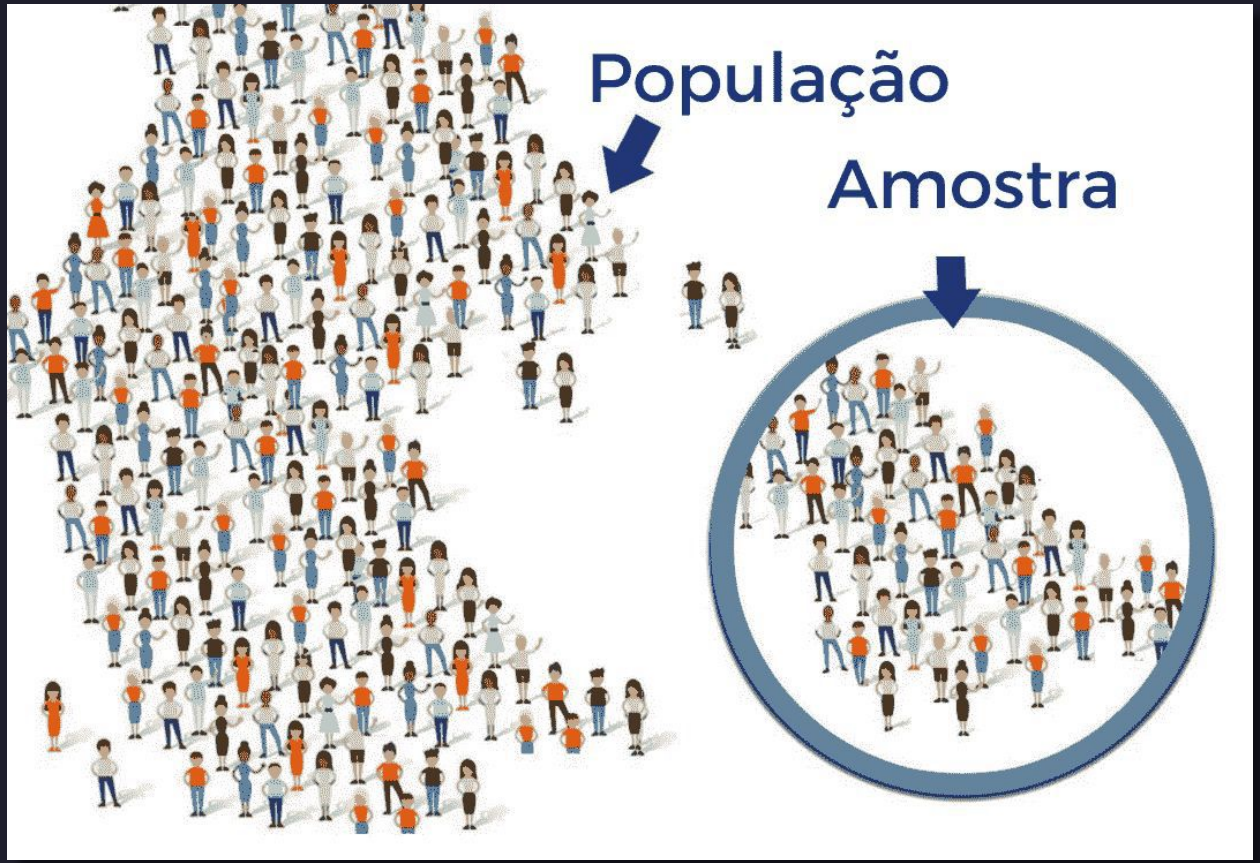
O QUE JÁ SABEMOS FAZER?

- Padronizar os dados
- Normalizar os dados

NOÇÕES SOBRE AMOSTRAGEM

QUAL A DIFERENÇA POPULAÇÃO E AMOSTRA?


POPULAÇÃO
AMOSTRA


População
Amostra

QUAL O TAMANHO IDEAL DE UMA AMOSTRA?

TAMANHO DA AMOSTRA - FÓRMULA DE YAMANE (1967)

A fórmula de Yamane é um método simples e amplamente utilizado para calcular o tamanho de amostra em pesquisas. Foi introduzida por Taro Yamane em 1967 e é especialmente útil para populações finitas.

- n = tamanho da amostra
- e = margem de erro (5%)
- N = tamanho da população

$$n = \frac{N}{1 + N(e)^2}$$

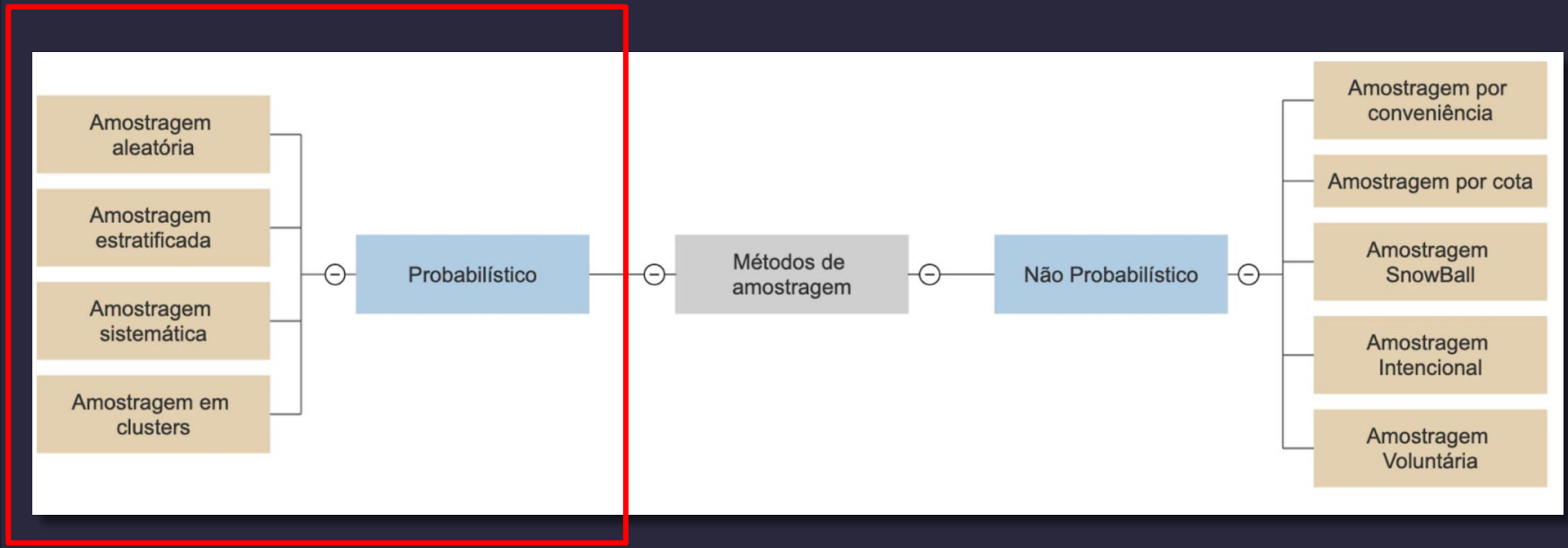
TAMANHO DA AMOSTRA - FÓRMULA DE YAMANE (1967)

```
def tamanhoAmostra(e,N):
    n = (N / (1 + (N*(e**2))))
    return (n)

e_ = 0.05
N_ = 4000

print(tamanhoAmostra(e_, N_))
```

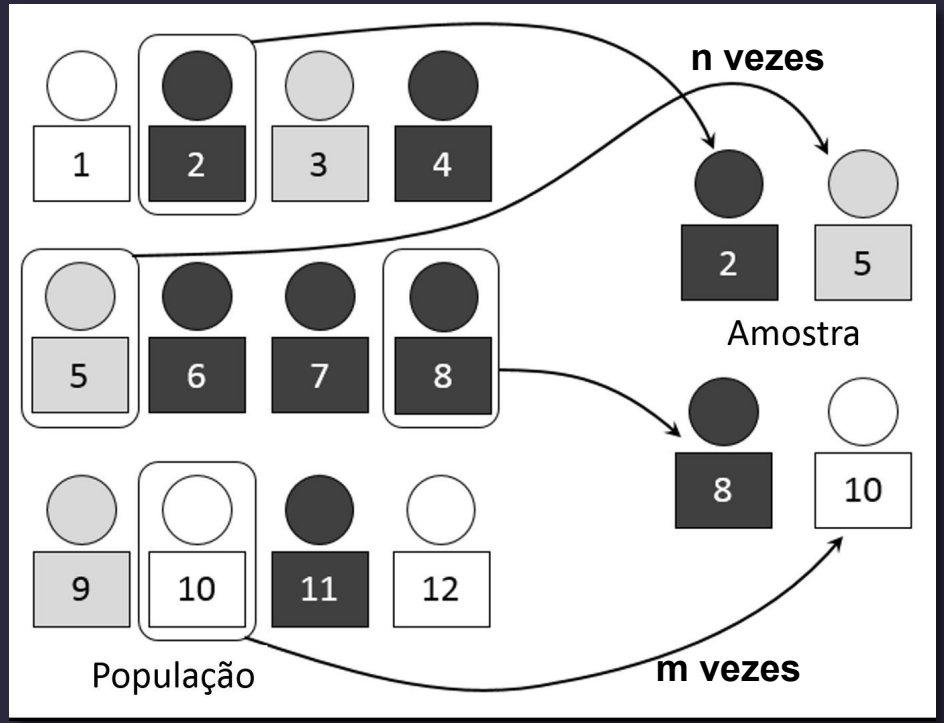
TIPOS DE AMOSTRAGEM



TIPOS DE AMOSTRAGEM PROBABILÍSTICA

- Amostragem aleatória simples:
  - Sem reposição;
  - Com reposição;
- Amostragem estratificada;
- Amostragem por conglomerados;
- Amostragem sistemática:
  - Estimador razão
  - Estimador regressão.

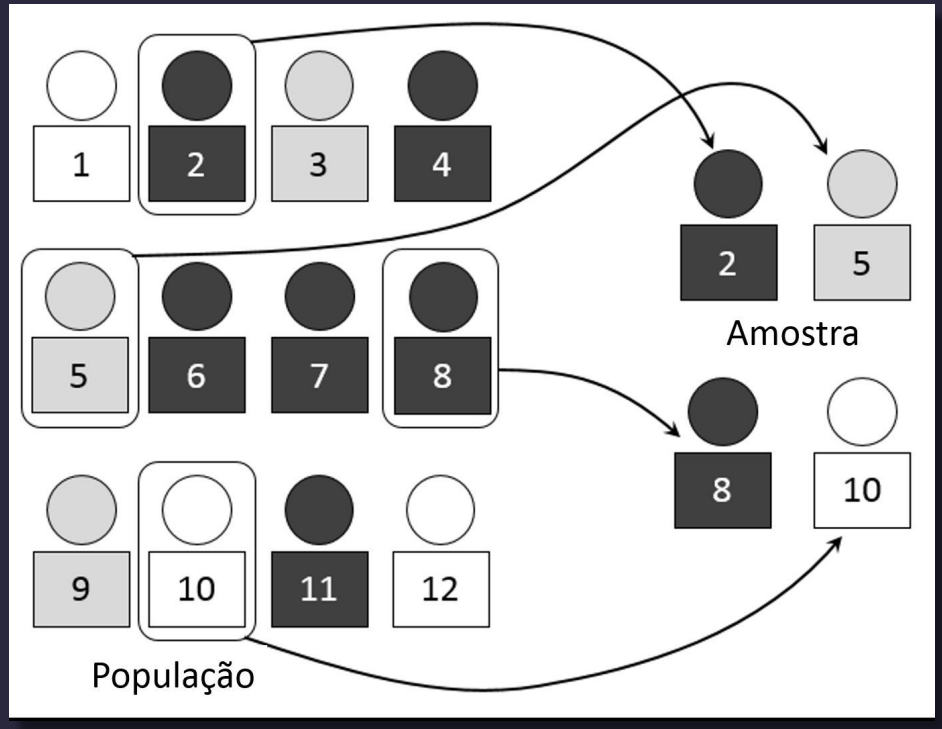
AMOSTRAGEM ALEATÓRIA SIMPLES SEM REPOSIÇÃO



AMOSTRAGEM ALEATÓRIA SIMPLES SEM REPOSIÇÃO

```
#amostra aleatória simples de 100 posições
amostra = df.sample(100)
```

AMOSTRAGEM ALEATÓRIA SIMPLES COM REPOSIÇÃO

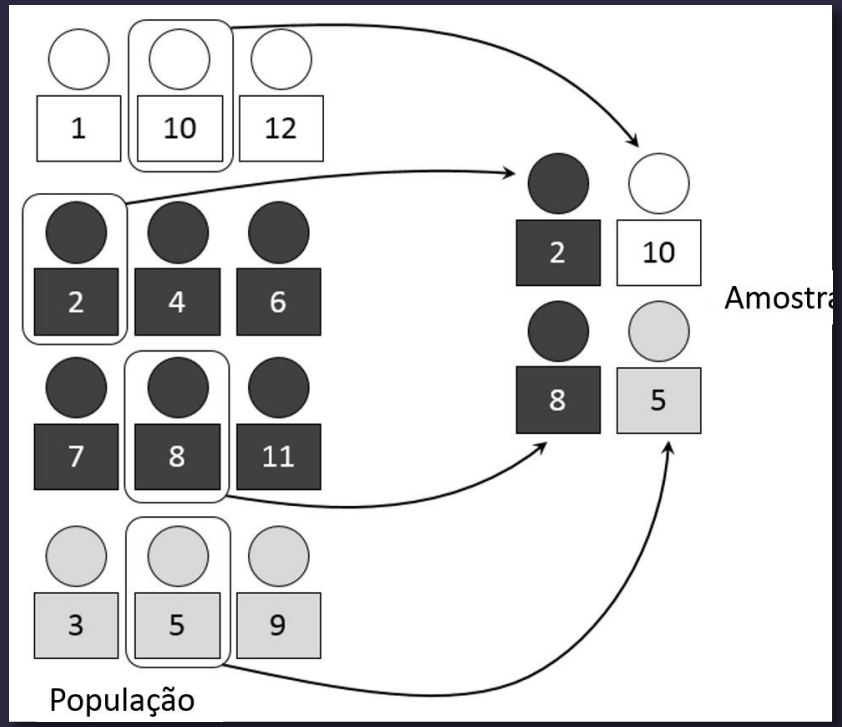


AMOSTRAGEM ALEATÓRIA SIMPLES COM REPOSIÇÃO

```
#amostra aleatória simples de 100 posições

amostra = df.sample(100, replace=True)
```

AMOSTRAGEM ESTRATIFICADA



AMOSTRAGEM ESTRATIFICADA

- Usuários rede social no estado A - 1 milhão
- Usuários rede social no estado B - 2 milhões
- Usuários rede social no estado C - 2 milhões

Quero calcular a média de seguidores dos usuários de uma rede social, observando os 3 estados. Como não ter um resultado tendencioso?

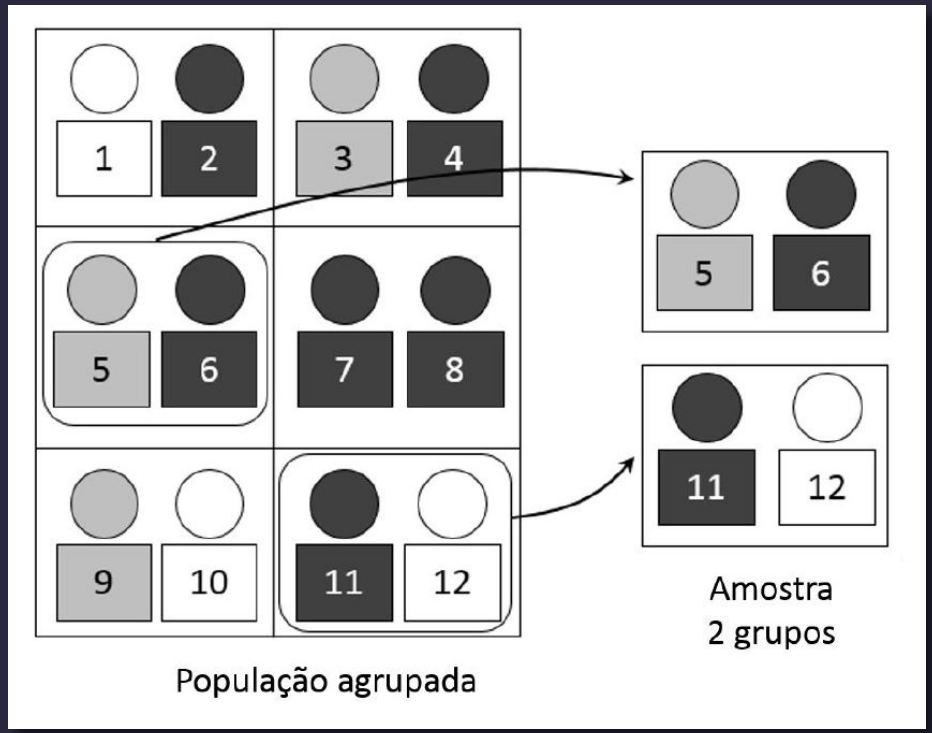
AMOSTRAGEM ESTRATIFICADA

```
from sklearn.model_selection import train_test_split

X = df.drop('cidade', axis=1)
y = df['cidade']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, stratify=y)
```

AMOSTRAGEM POR CONGLOMERADOS (CLUSTER)



AMOSTRAGEM POR CONGLOMERADOS (CLUSTER)

- Mulheres cientistas de dados: 1000
- Homens cientistas de dados: 3000

Amostra 30% das mulheres atuantes na área de Ciência de Dados no Brasil

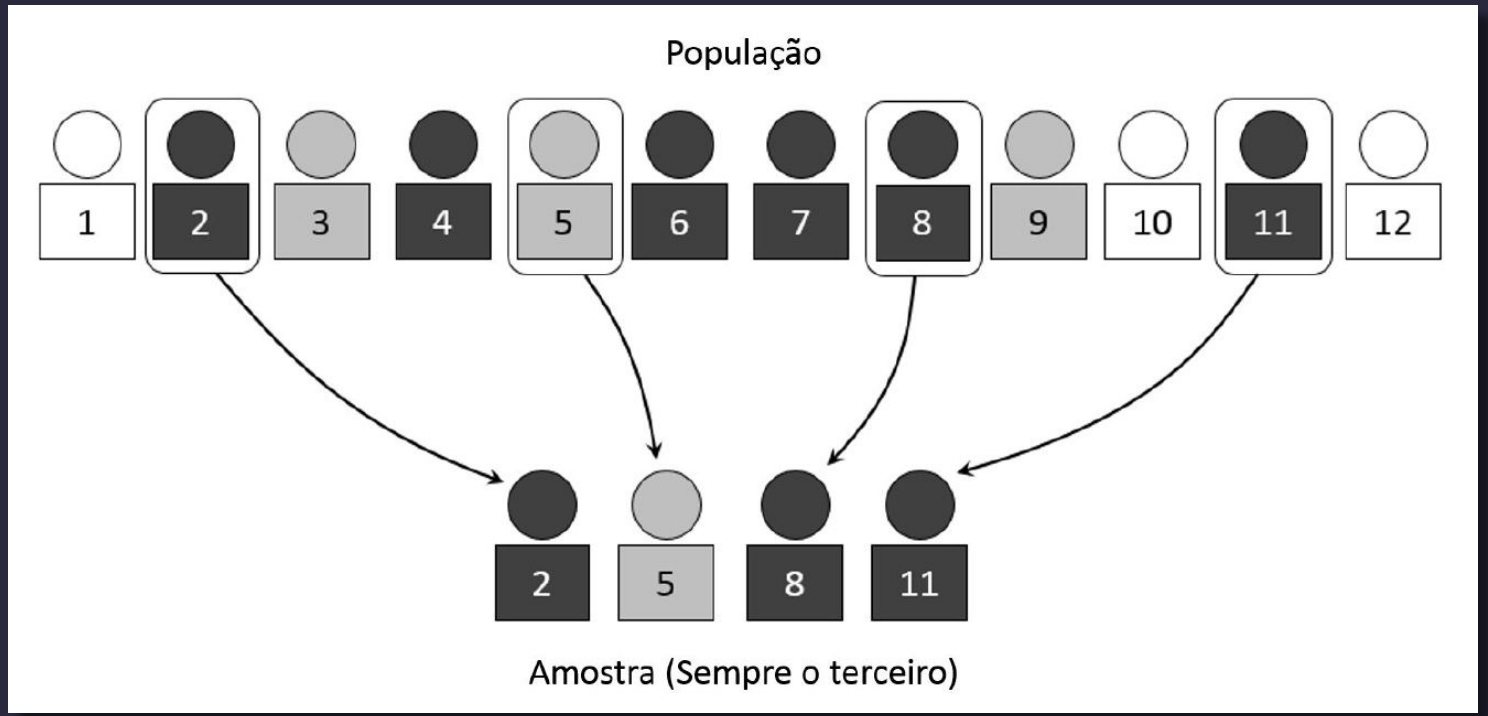
AMOSTRAGEM POR CONGLOMERADOS (CLUSTER)

```
import panda as pd

grupo = df.groupby('genero').apply(pd.DataFrame.sample,
frac=.3)

grupo[grupo.genero=='feminino']
```

AMOSTRAGEM SISTEMÁTICA

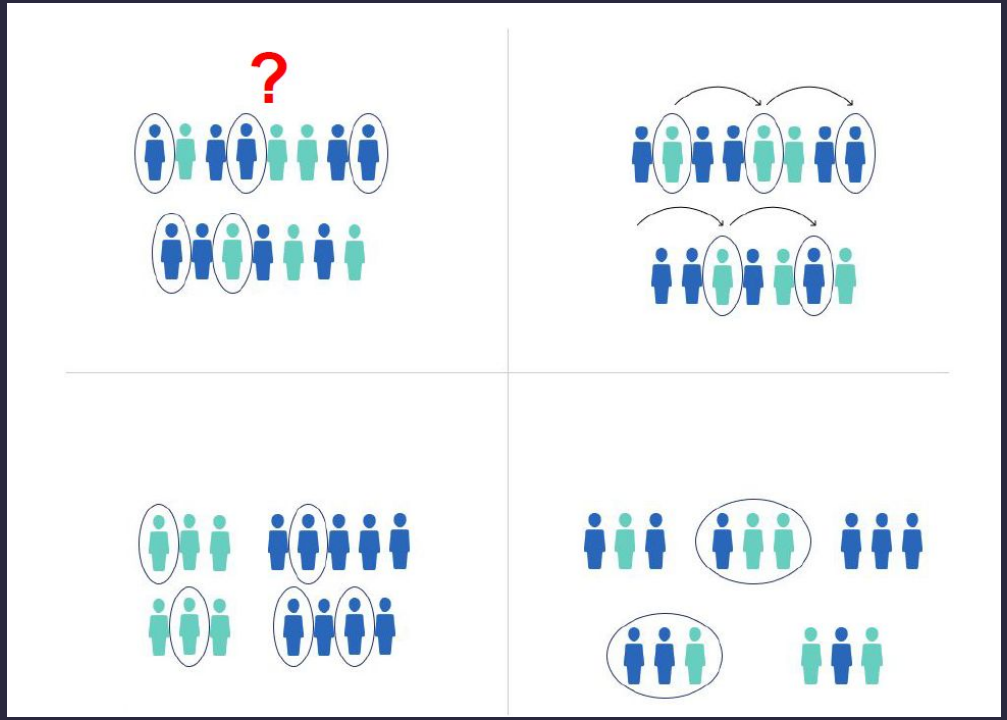


AMOSTRAGEM SISTEMÁTICA

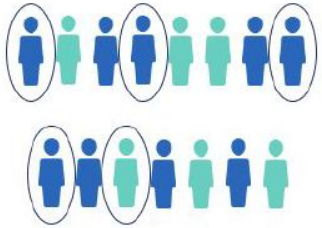
```
import numpy as np


# Gerando um array que inicia em 0
# e termina em 12 com um intervalo de 3:
indices = np.arange(0,12,3)
df.loc[indices]
```

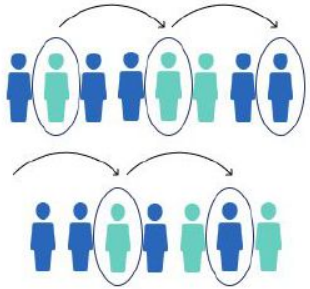
AMOSTRAGEM - RECAPITULANDO



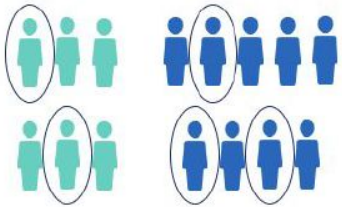
Amostragem aleatória



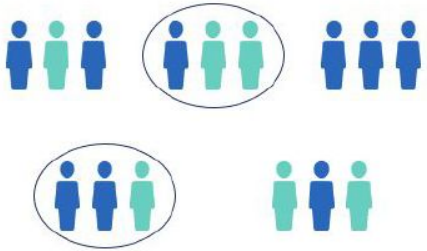
Amostragem sistemática



Amostragem estratificada



Amostragem por conglomerado

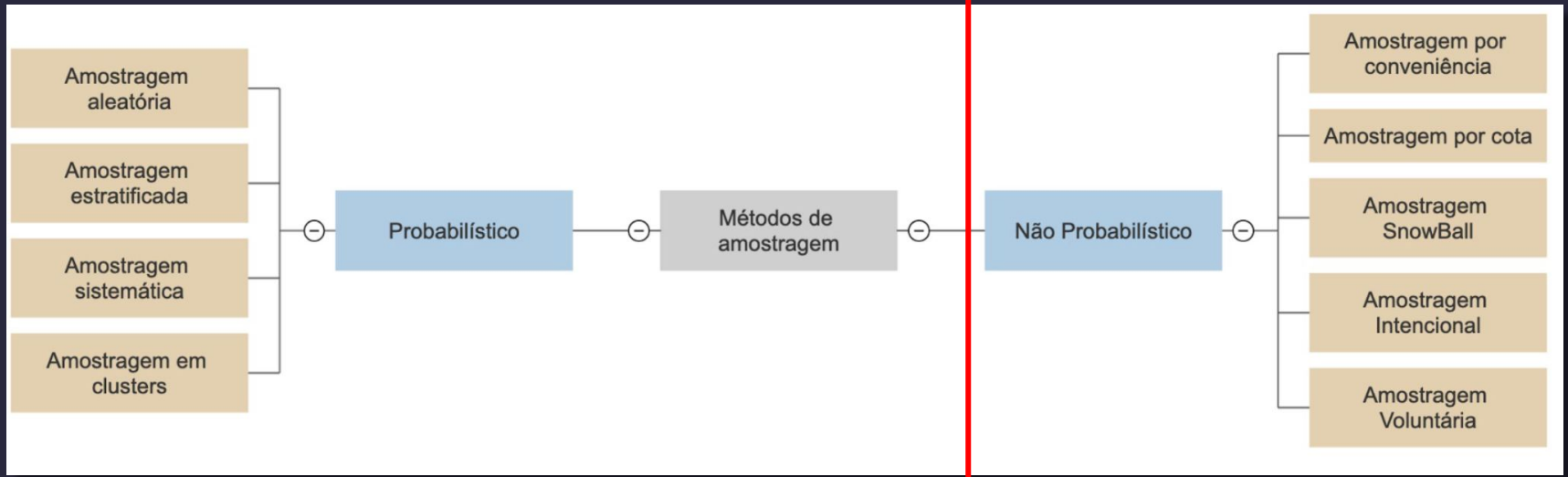


ATIVIDADE 11 - << DESAFIO >>

Crie as seguintes amostras, a partir dos dados sobre o PROUNI:

a) Amostra aleatória com repetição
b) Amostra aleatória sem repetição
c) Amostra estratificada (30% dos dados)

/TIPOS DE AMOSTRAGEM



TIPOS DE AMOSTRAGEM - NÃO PROBABILÍSTICA

- Amostragem por conveniência
- Amostragem por cota
- Amostragem snowball
- Amostragem Intencional
- Amostragem voluntária

AMOSTRAGEM - POR CONVENIÊNCIA

- Consiste em selecionar uma amostra da população que seja acessível;
- Representa uma maior facilidade operacional e baixo custo de amostragem
- Consequência: incapacidade de fazer afirmações gerais com rigor estatístico sobre a população.

AMOSTRAGEM - POR COTAS

- Versão não-probabilística da amostra estratificada
- Etapas:
  - Segmentação
  - Definição do tamanho das cotas
  - Seleção de participantes

AMOSTRAGEM - *SNOWBALL*

- Amostragem bola de neve;
- Os indivíduos selecionados para serem estudados convidam novos participantes da sua rede de amigos e conhecidos;
- Usada em populações de baixa incidências e indivíduos de difícil acesso.
- Tipos:
  - Amostra linear
  - Amostra exponencial

AMOSTRAGEM INTENCIONAL

- A seleção é baseada no conhecimento sobre a população e o propósito do estudo;
- Conhecimento prévio da população.

AMOSTRAGEM VOLUNTÁRIA

- Ocorre quando o componente da população se oferece voluntariamente para fazer parte da amostra.

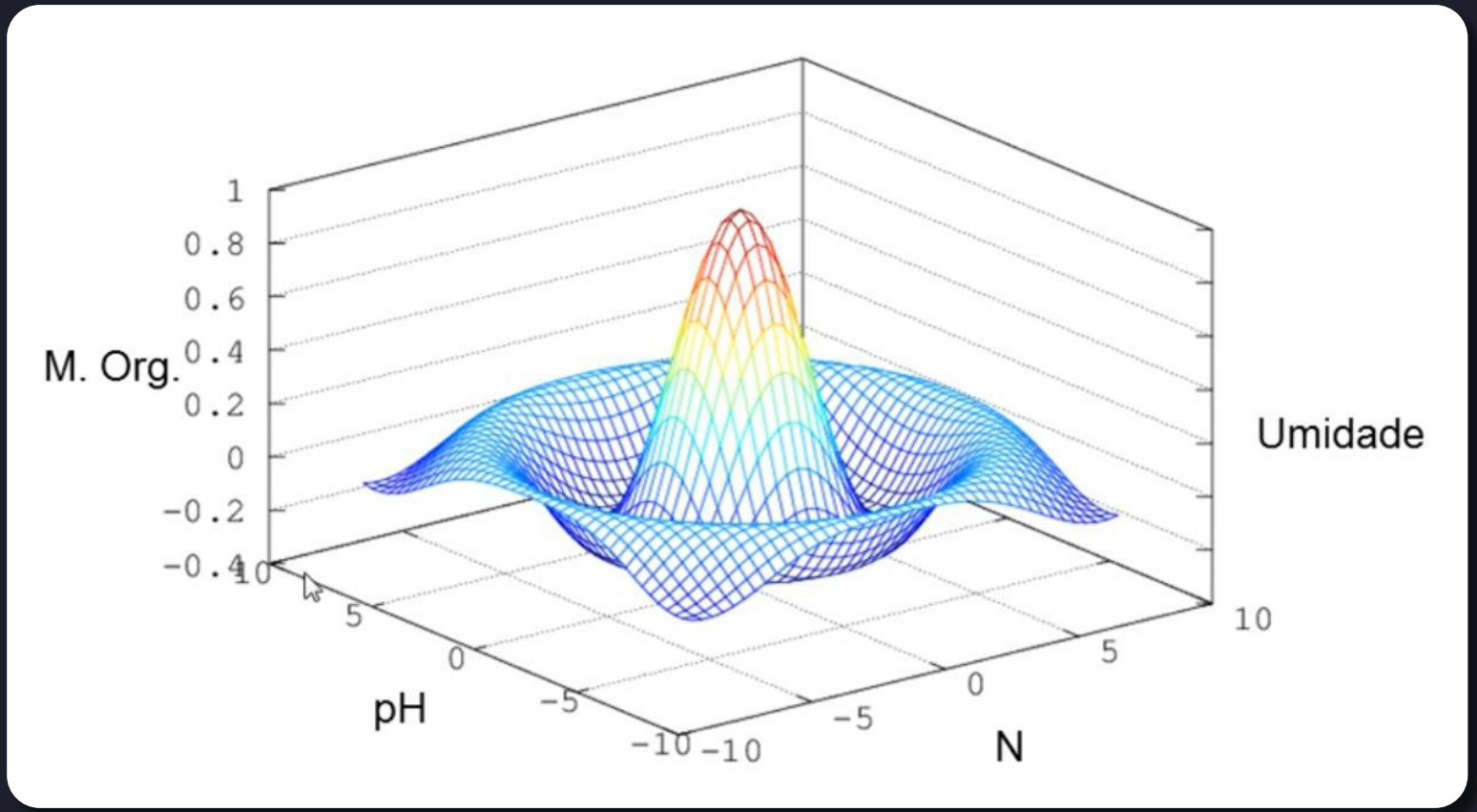
ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)

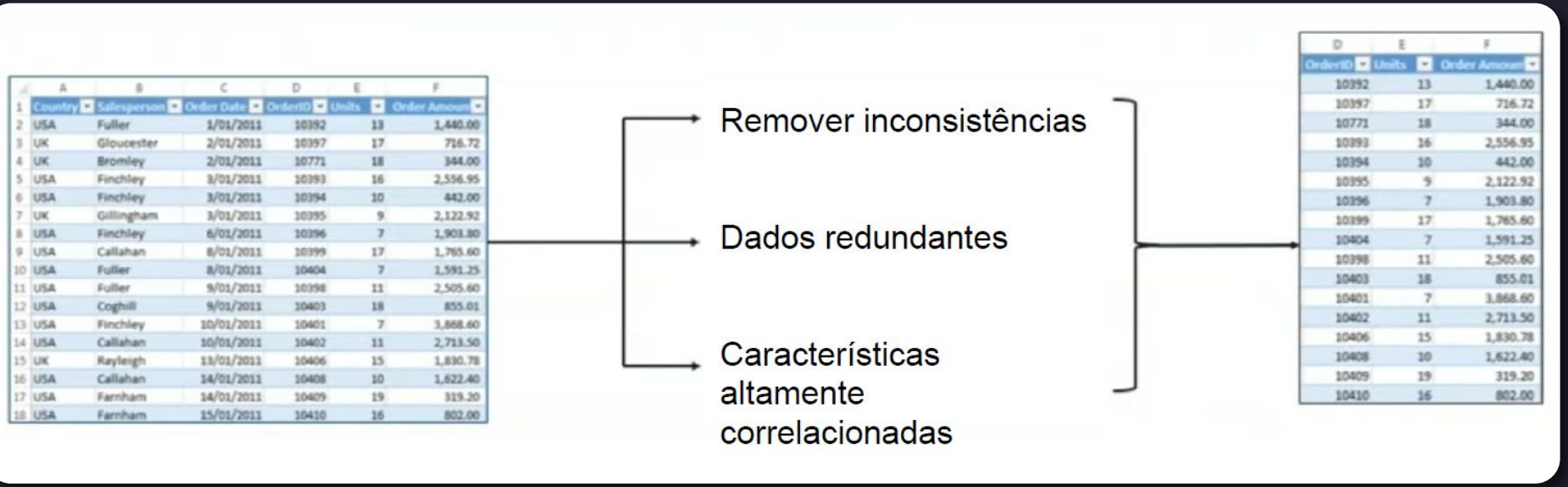
O QUE É PCA?

Técnica que usa princípios de álgebra linear para transformar variáveis, possivelmente correlacionadas, em um número menor de variáveis chamadas de **Componentes Principais.**

POR QUE USAR PCA?

- **Redução de Dimensionalidade:** Em conjuntos de dados com muitas variáveis, nem todas as variáveis são igualmente informativas. O PCA permite reduzir a dimensionalidade, retendo as características mais informativas.
- **Visualização:** Ao reduzir a dimensionalidade para 2 ou 3 componentes principais, os dados podem ser visualizados em um gráfico bidimensional ou tridimensional.
- **Redução de Ruído:** Ao manter apenas os componentes principais significativos, o PCA pode ajudar a filtrar o ruído.
- **Descorrelacionar Recursos:** Os componentes principais são ortogonais entre si, o que significa que eles são descorrelacionados.


M. Org.
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
10
5
0
-5
-10
pH
-10
-5
0
5
10
N
Umidade



	A	B	C	D	E	F
1	Country	Salesperson	Order Date	OrderID	Units	Order Amount
2	USA	Fuller	1/01/2011	10392	13	1,440.00
3	UK	Gloucester	2/01/2011	10397	17	716.72
4	UK	Bromley	2/01/2011	10771	18	344.00
5	USA	Finchley	3/01/2011	10393	16	2,556.95
6	USA	Finchley	3/01/2011	10394	10	442.00
7	UK	Gillingham	3/01/2011	10395	9	2,122.92
8	USA	Finchley	6/01/2011	10396	7	1,903.80
9	USA	Callahan	8/01/2011	10399	17	1,765.60
10	USA	Fuller	8/01/2011	10404	7	1,591.25
11	USA	Fuller	9/01/2011	10398	11	2,505.60
12	USA	Coghill	9/01/2011	10403	18	855.01
13	USA	Finchley	10/01/2011	10401	7	3,868.60
14	USA	Callahan	10/01/2011	10402	11	2,713.50
15	UK	Rayleigh	13/01/2011	10406	15	1,830.78
16	USA	Callahan	14/01/2011	10408	10	1,622.40
17	USA	Farnham	14/01/2011	10409	19	319.20
18	USA	Farnham	15/01/2011	10410	16	802.00

	D	E	F
1	OrderID	Units	Order Amount
2	10392	13	1,440.00
3	10397	17	716.72
4	10771	18	344.00
5	10393	16	2,556.95
6	10394	10	442.00
7	10395	9	2,122.92
8	10396	7	1,903.80
9	10399	17	1,765.60
10	10404	7	1,591.25
11	10398	11	2,505.60
12	10403	18	855.01
13	10401	7	3,868.60
14	10402	11	2,713.50
15	10406	15	1,830.78
16	10408	10	1,622.40
17	10409	19	319.20
18	10410	16	802.00

ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)

- Ajuda a extrair informações dos dados sem qualquer supervisão;
- Técnica de redução linear da dimensionalidade dos dados
- Objetivo: reduzir o número de variáveis significativas nos dados

COMO DIMINUIR A DIMENSIONALIDADE DOS DADOS?

Observando a variância dos dados de maneira maximizada.

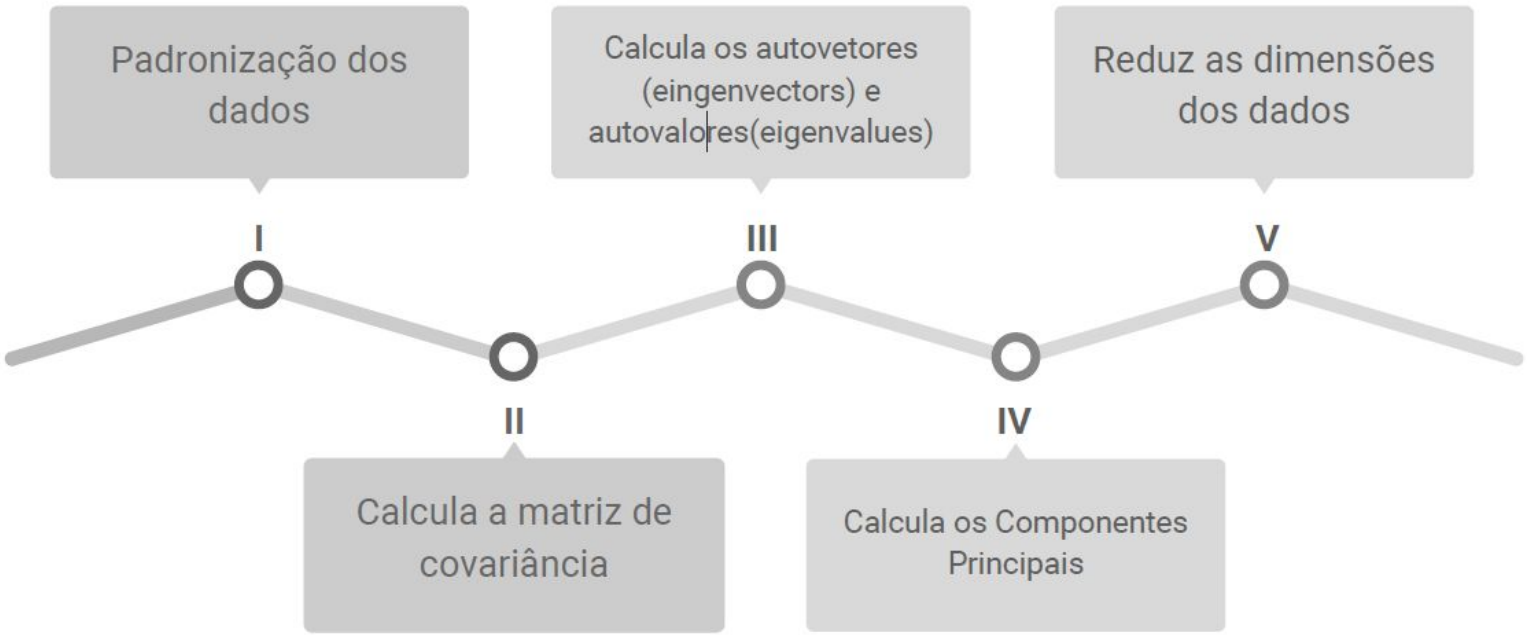
VARIÂNCIA

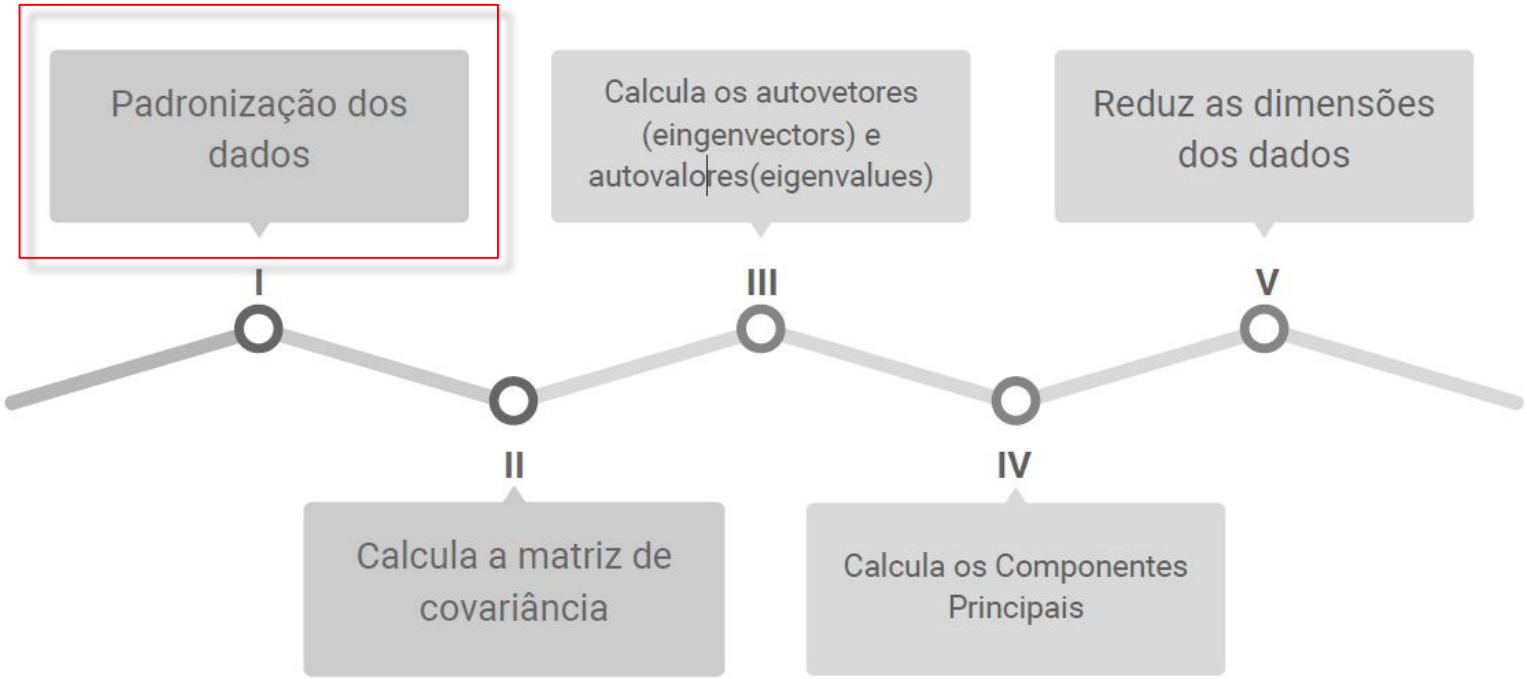
- É a tendência dos valores de uma variável mudar em cada medição.
- Variáveis quantitativas e categóricas podem apresentar variância de valores.

COVARIÂNCIA

- Maneira de verificar se duas variáveis estão associadas entre si.
  - Ou seja, se elas variam conjuntamente.
- Mudanças numa variável corresponde em mudanças em outra variável.

COMO APLICAR O PCA?


Padronização dos dados
I
Calcula a matriz de covariância
II
Calcula os autovetores (eingenvectors) e autovalores(eigenvalues)
III
Calcula os Componentes Principais
IV
Reduz as dimensões dos dados
V


Padronização dos dados
I
Calcula a matriz de covariância
II
Calcula os autovetores (eingenvectors) e autovalores(eigenvalues)
III
Calcula os Componentes Principais
IV
Reduz as dimensões dos dados
V

PADRONIZAÇÃO DOS DADOS

É importante padronizar os dados (média zero e variância unitária) antes do PCA, especialmente se as variáveis têm escalas diferentes.

Rating	# of downloads
5	1383
3	668
2	763
5	839
1	342

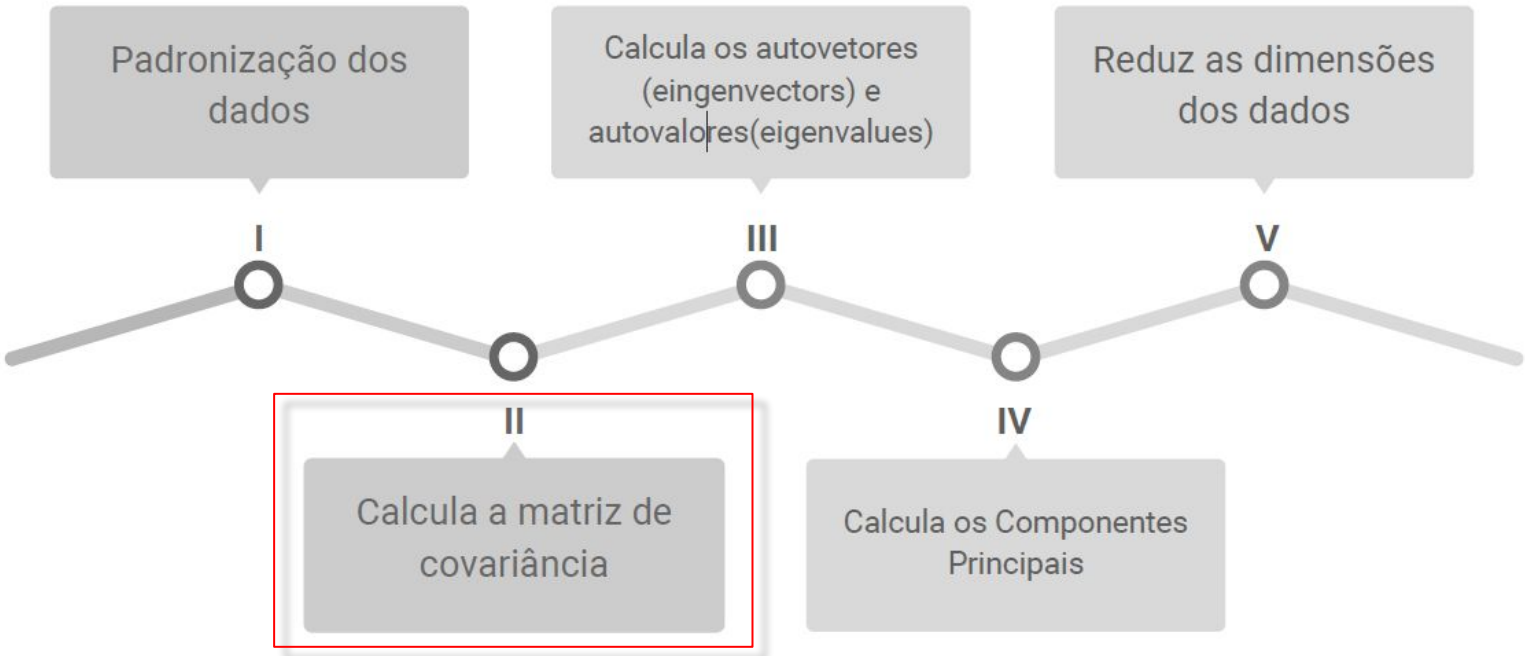
Rating feature ranges between 0-5

# of downloads ranges between 100-5000

PADRONIZAÇÃO DOS DADOS

```
from sklearn.preprocessing import StandardScaler

X_std = StandardScaler().fit_transform(X)
```


Padronização dos dados
I
Calcula a matriz de covariância
II
Calcula os autovetores (eingenvectors) e autovalores(eigenvalues)
III
Calcula os Componentes Principais
IV
Reduz as dimensões dos dados
V

MATRIZ DE COVARIÂNCIA

Calcule a matriz de covariância dos dados padronizados.

$$\begin{bmatrix} Cov(a, a) & Cov(a, b) \\ Cov(b, a) & Cov(b, b) \end{bmatrix}$$

A covariância sinaliza:

- O grau de dependência entre duas variáveis, que pode ser:
  - **Covariância negativa**: inversamente proporcionais;
  - **Covariância positiva**: diretamente proporcionais.

MATRIZ DE COVARIÂNCIA

```
import numpy as np

# Forma 1
mean_vec = np.mean(X_std, axis=0)
cov_mat = (X_std - mean_vec).T.dot((X_std - mean_vec)) /
(X_std.shape[0]-1)
```

MATRIZ DE COVARIÂNCIA

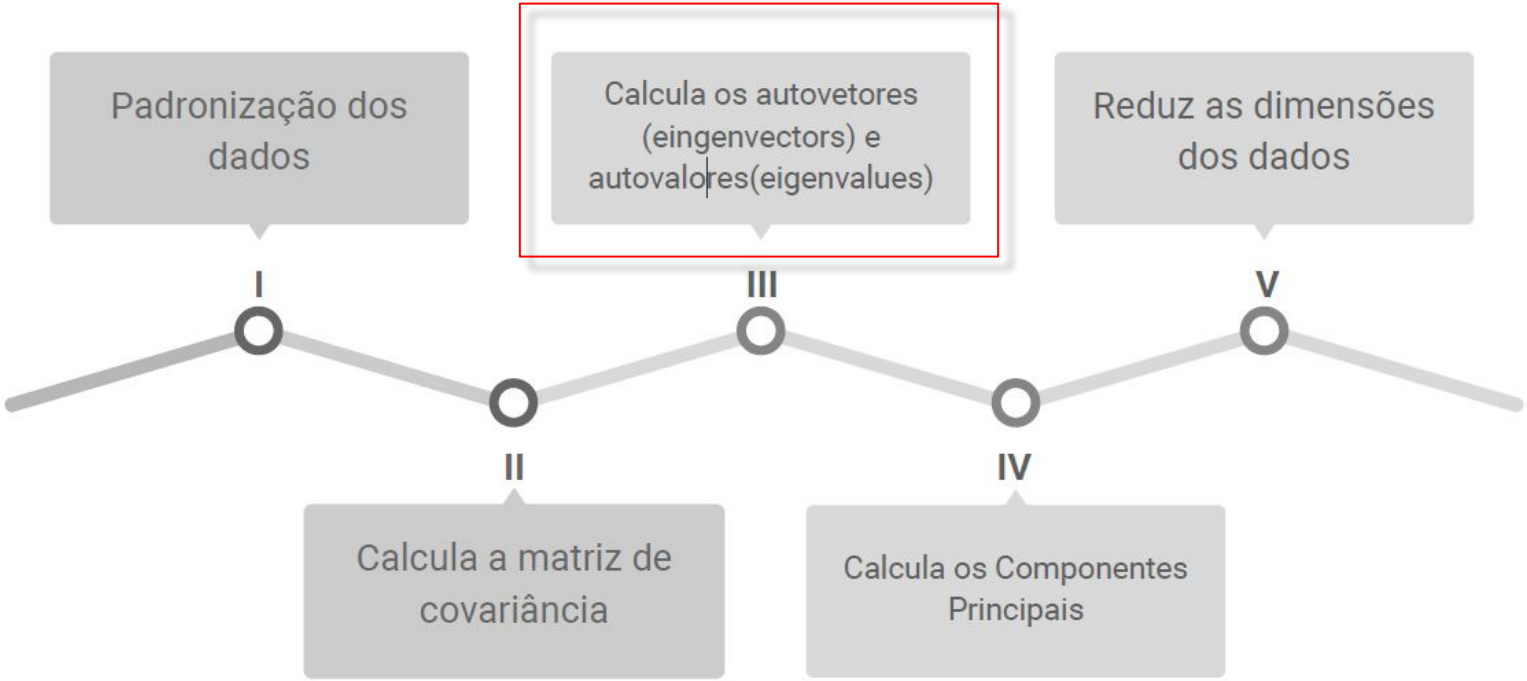
```
import numpy as np


# Forma 2
cov_mat = np.cov(X_std.T)
```

/MATRIZ DE COVARIÂNCIA

```
import numpy as np


# Forma 3
cor_mat1 = np.corrcoef(X_std.T)
```


Padronização dos dados
I
Calcula a matriz de covariância
II
Calcula os autovetores (eingenvectors) e autovalores(eigenvalues)
III
Calcula os Componentes Principais
IV
Reduz as dimensões dos dados
V

AUTOVETORES E AUTOVALORES

São construções matemáticas que devem ser computados da matriz de covariância para determinar os componentes principais dos dados.

- **Autovetores (eigenvectors):**
  - são esses vetores quando uma transformação linear é executada neles e suas direções não mudam;

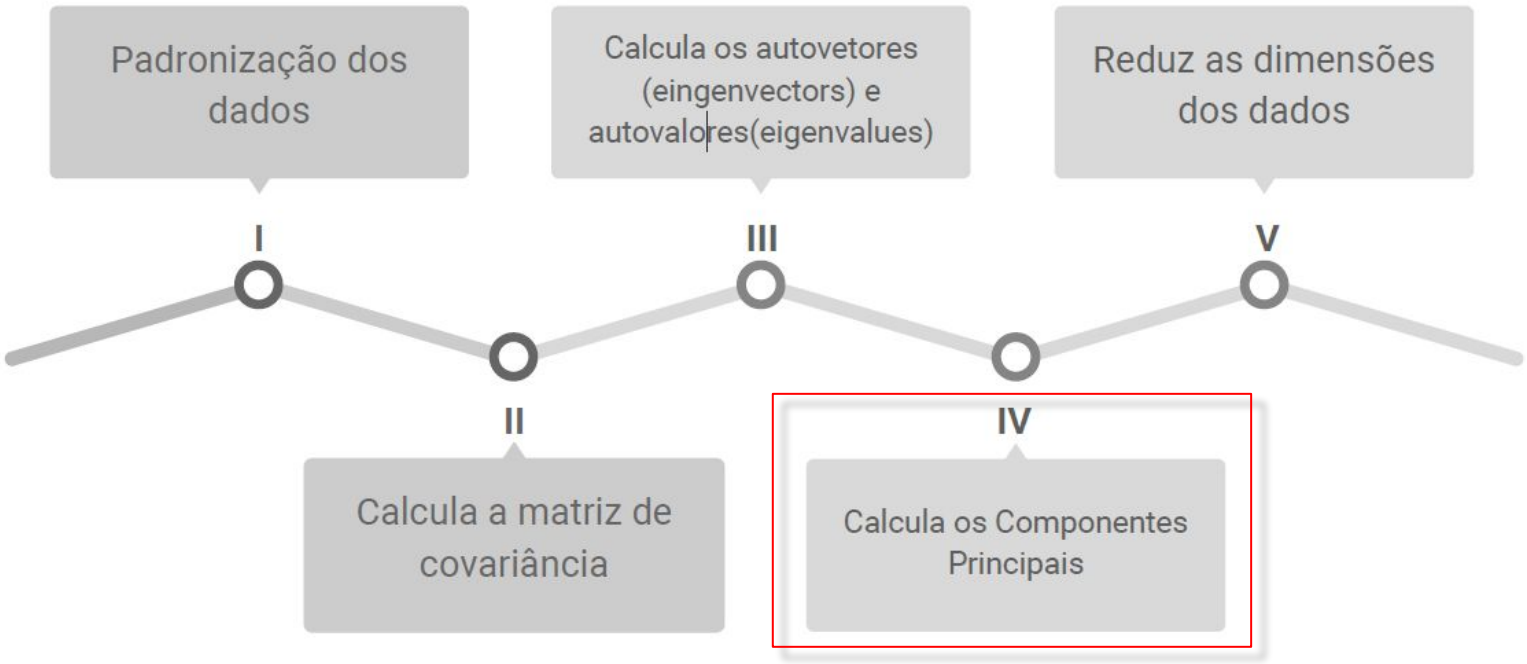
- **Autovalores (eigenvalues):**
  - são as escalas desse autovetores

AUTOVETORES E AUTOVALORES

```
import numpy as np

cov_mat = np.cov(X_std.T)
eig_vals, eig_vecs = np.linalg.eig(cov_mat)

print('Autovetores \n%s' %eig_vecs)
print('\nAutovalores \n%s' %eig_vals)
```

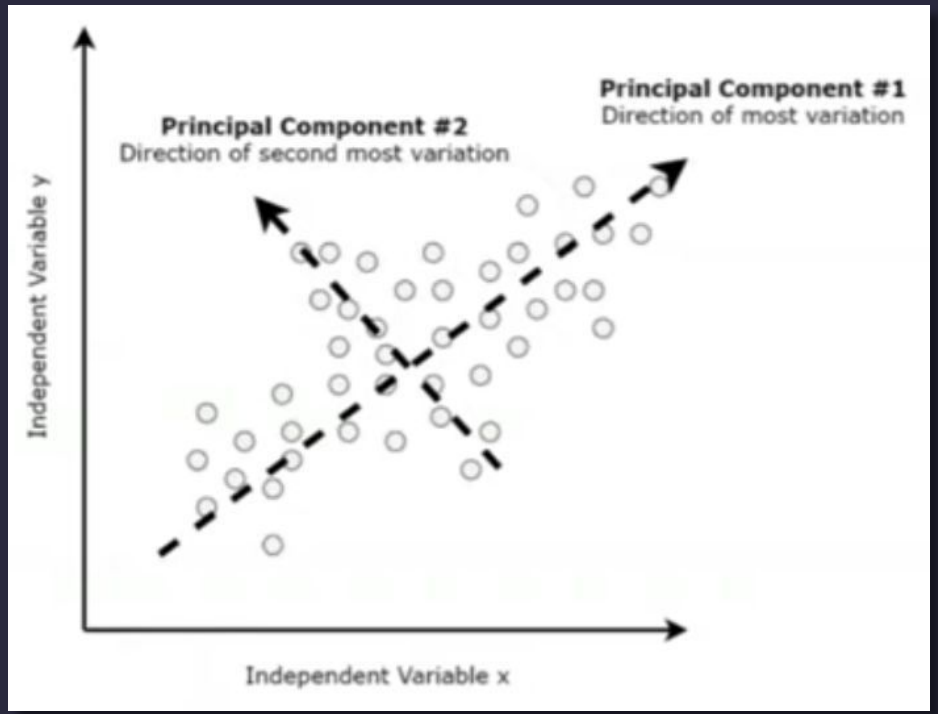

Padronização dos dados
I
Calcula a matriz de covariância
II
Calcula os autovetores (eingenvectors) e autovalores(eigenvalues)
III
Calcula os Componentes Principais
IV
Reduz as dimensões dos dados
V

CÁLCULO DOS PRINCIPAIS COMPONENTES

Agora os autovetores e autovalores são colocados em ordem decrescente para descobrir quais fatores são mais significantes.

- O autovalor mais significativo é o componente principal 1 (PC1)
- O 2o autovalor mais significativo é o componente principal 2 (PC2)
- ….

/CÁLCULO DOS PRINCIPAIS COMPONENTES

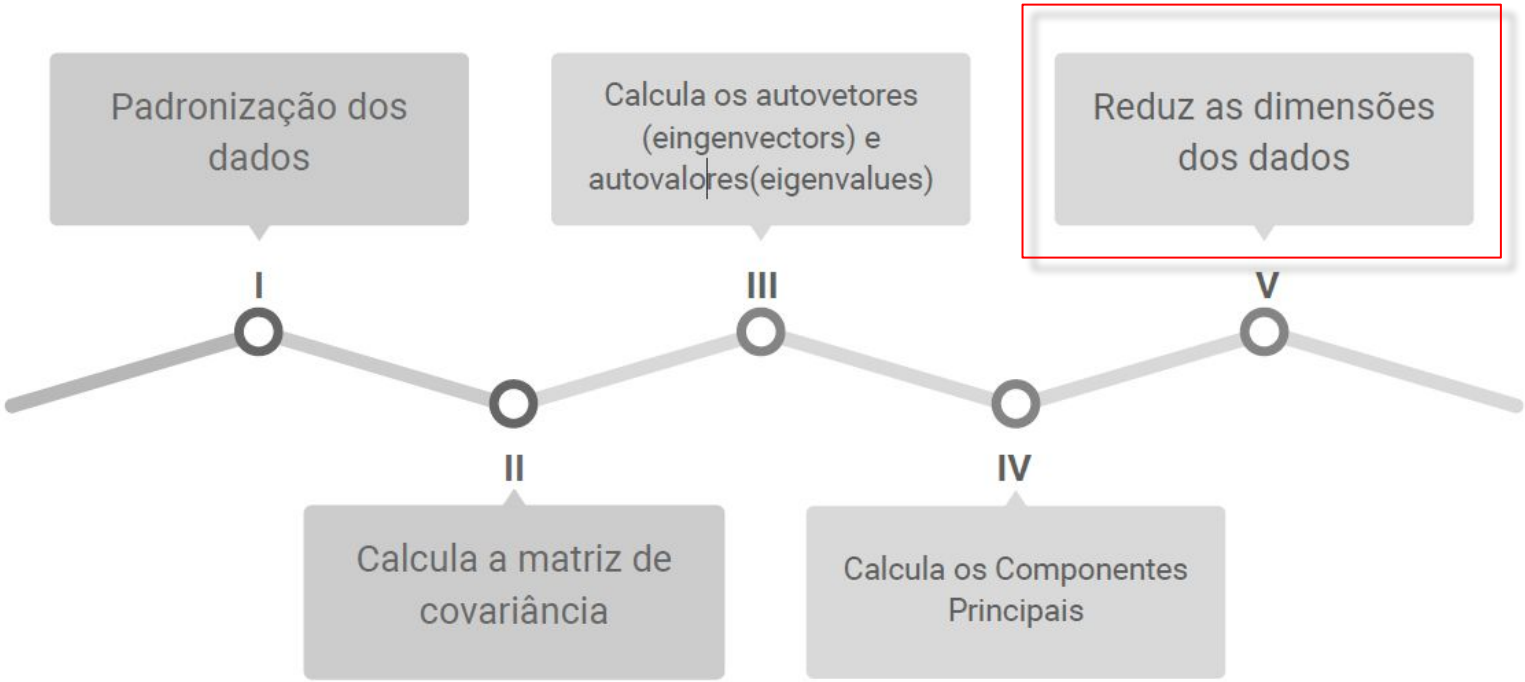


CÁLCULO DOS PRINCIPAIS COMPONENTES

```
from sklearn.decomposition import PCA

pca = PCA(n_components= 9)
principalComponents = pca.fit_transform(X_std)

print(pca.explained_variance_ratio_)
```


Padronização dos dados
I
Calcula a matriz de covariância
II
Calcula os autovetores (eingenvectors) e autovalores(eigenvalues)
III
Calcula os Componentes Principais
IV
Reduz as dimensões dos dados
V

REDUÇÃO DAS DIMENSÕES DOS DADOS

O último passo é o rearranjo dos dados originais com os componentes principais finais.

Representam a informação máxima e mais significativa dos dados.

movieId	title	genres
1	Toy Story	Adventure|Animation|Childre
2	Jumanji (1	Adventure|Children|Fantasy
3	Grumpier	Comedy|Romance
4	Waiting to	Comedy|Drama|Romance
5	Father of	Comedy
6	Heat (199	Action|Crime|Thriller
7	Sabrina (1	Comedy|Romance
8	Tom and H	Adventure|Children
9	Sudden D	Action
10	GoldenEy	Action|Adventure|Thriller
11	American	Comedy|Drama|Romance
12	Dracula: D	Comedy|Horror
13	Balto (199	Adventure|Animation|Childre
14	Nixon (19	Drama
15	Cutthroat	Action|Adventure|Romance
16	Casino (19	Crime|Drama
17	Sense and	Drama|Romance
18	Four Roon	Comedy

userId	movieId	rating	timestamp
1	1	5	8.47E+08
1	2	3	8.48E+08
1	10	3	8.48E+08
1	32	4	8.48E+08
1	34	4	8.48E+08
1	47	3	8.48E+08
1	50	4	8.48E+08
1	62	4	8.48E+08
1	150	4	8.47E+08
1	153	3	8.47E+08
1	160	3	8.48E+08
1	161	4	8.48E+08
1	165	4	8.47E+08
1	185	3	8.48E+08
1	208	3	8.48E+08
1	253	3	8.48E+08
1	265	5	8.48E+08

REDUÇÃO DAS DIMENSÕES DOS DADOS

```
model = PCA(n_components=9).fit(X_std)
X_pc = model.transform(X_std)


# número de componentes
n_pcs= model.components_.shape[0]


# recebe o índice das características mais importantes de
# cada componente (variável)
most_important = [np.abs(model.components_[i]).argmax() for
i in range(n_pcs)]
```

REDUÇÃO DAS DIMENSÕES DOS DADOS

```
initial_feature_names = X.columns


#recebendo os nomes
most_important_names =
[initial_feature_names[most_important[i]] for i in
range(n_pcs)]
dic = {'PC{}'.format(i): most_important_names[i] for i in
range(n_pcs)}
```

REDUÇÃO DAS DIMENSÕES DOS DADOS

```
# build dataframe
listagemPCAs = pd.DataFrame(dic.items())
listagemPCAs
```

TESTE DE HIPÓTESES

O QUE É UMA HIPÓTESE?

O QUE É UMA HIPÓTESE?

Conjectura

Suposição que se faz sobre algo, que pode ser verdadeira ou falsa, fundamentando-se em evidências incompletas ou pressentimentos;

EXEMPLOS DE HIPÓTESES

- A produtividade média de cana de açúcar no estado da PB é de 2500 kg/ha;
- A proporção de peças defeituosas em uma unidade de fabricação é de 10%;
- A propaganda produz efeito positivo nas vendas;
- Métodos de ensino diferentes produzem resultados diferentes de aprendizagem;

QUAL É A DIFERENÇA DE HIPÓTESE E DO PROBLEMA A SER RESOLVIDO PELA ANÁLISE DE DADOS?

QUAL É A DIFERENÇA DE HIPÓTESE E DO PROBLEMA A SER RESOLVIDO PELA ANÁLISE DE DADOS?

O problema é solucionado a partir da elaboração de diversas hipóteses.

QUAL É A DIFERENÇA DE HIPÓTESE E DO PROBLEMA A SER RESOLVIDO PELA ANÁLISE DE DADOS?

Problema:
Como aumentar as vendas do jogo God of War?

Hipóteses:

- Se dobrar a produção da mídia do jogo, as vendas irão aumentar 50%.
- Se baixar o preço pela metade, as vendas irão aumentar 50%.
- A propaganda produz efeito positivo nas vendas.
- Se promover um campeonato mundial, as vendas irão aumentar 100%.

ATIVIDADE 12 - << DESAFIO >>

Formule uma hipótese para os dados do PROUNI.

HIPÓTESE DE EXEMPLO

Problema:
Notas do prouni de instituições da Paraíba

Hipóteses:

A média de notas dos cursos de Bacharelado é igual a 593.89

Cursos de Bacharelado tem uma nota de prouni melhores que cursos de Licenciatura

Hipótese

≠

Hipótese Estatística

HIPÓTESE ESTATÍSTICA

Uma hipótese estatística, formalmente, é uma afirmação sobre alguma característica da população.

HIPÓTESE ESTATÍSTICA

Teste de hipótese é um procedimento estatístico capaz rejeitar ou não uma afirmação que representa uma igualdade sobre uma população (chamada de H0).

A decisão do teste é tomada com base na menor probabilidade tolerável de incorrer no erro tipo 1 (rejeitar H0, quando ela é verdadeira).

EXEMPLO A - CONTEXTUALIZAÇÃO

Imagine que houve uma votação sobre qual é o melhor jogo online de todos os tempos!

EXEMPLO A - CONTEXTUALIZAÇÃO

Imagine que os jogadores deram notas (0-10) para alguns jogos!

A - Counter Strike
B - League of Legends
C - Free Fire
D - FIFA

EXEMPLO A - CONTEXTUALIZAÇÃO

Problema:

Quero saber a popularidade do LOL entre seus jogadores (eles dariam notas para o jogo de 0 a 10)

user	nota	jogo
A	10	LOL
B	2	CS
...	...	...

EXEMPLO A

Hipótese: $\mu LOL = 8$

As notas do jogo LOL terão um comportamento parecido dos outros jogos. (média das notas de todos os jogos = 8)

Hipótese nula (H0): $\mu LOL \leq 8$

O jogo LOL terá uma média de notas **menor ou igual** do que 8.

Hipótese alternativa (H1): $\mu LOL > 8$

O jogo LOL terá uma média de notas **maior do que 8.**

EXEMPLO A

Hipótese nula (H0): □LOL ≤ 8

O jogo LOL terá uma média de notas **menor ou igual** do que 8.

Hipótese alternativa (H1): □LOL > 8

O jogo LOL terá uma média de notas **maior do que 8**.

Como averiguar essa hipótese nula?

EXEMPLO A

Hipótese nula (H0): $\mu_{LOL} \leq 8$

O jogo LOL terá uma média de notas **menor ou igual** do que 8.

Hipótese alternativa (H1): $\mu_{LOL} > 8$

O jogo LOL terá uma média de notas **maior do que 8**.

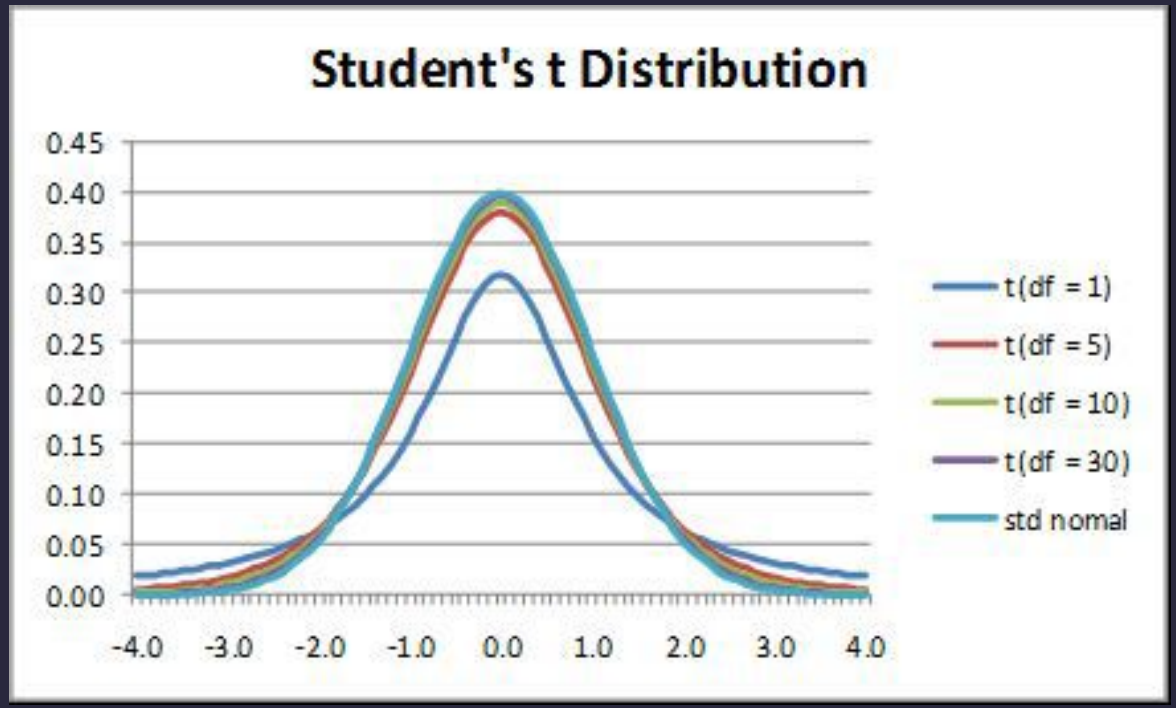
Como averiguar essa hipótese nula?

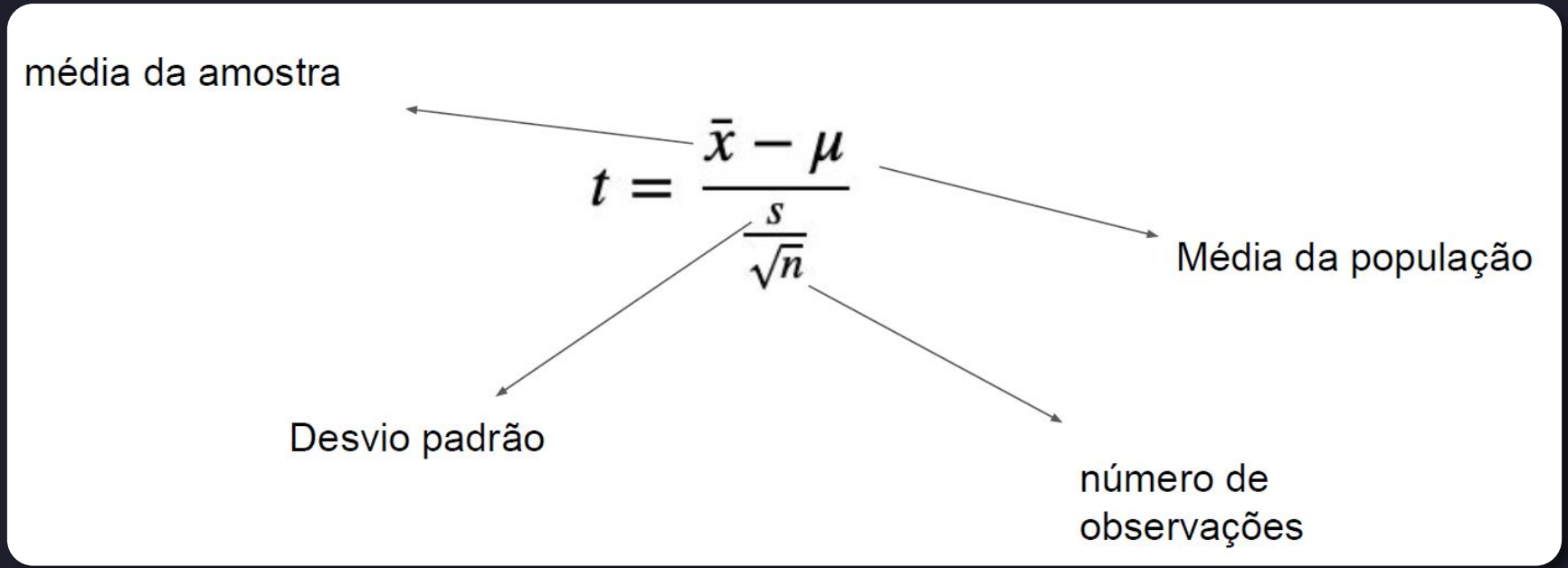
Teste t de Student (de uma amostra)

TESTE t DE STUDENT

- É um tipo de teste de hipótese útil na estatística quando é necessário comparar médias de dois grupos;
- É possível comparar uma média amostral com um valor hipotético ou com um valor alvo usando um teste t para uma amostra;
- Usado em amostras dependentes e independentes;

TESTE t DE STUDENT




média da amostra
$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$
Média da população
Desvio padrão
número de observações

TESTE t DE STUDENT

```
from scipy import stats

stats.shapiro(df['LOL'])
(0.989665354387, 0.9876567575985)
```

É normal!

TESTE t DE STUDENT

```
from scipy import stats
stats.shapiro(df['LOL'])
```

```
stats.stats.ttest_1samp(df['LOL'],8, axis=0, equal_var=True)
(0.905009388923645, 0.2484823909759521)
```

Estatística t

p-valor

TESTE t DE STUDENT

```
from scipy import stats
stats.shapiro(df['LOL'])
```

```
stats.stats.ttest_1samp(df['LOL'],8, axis=0, equal_var=True)
(0.905009388923645, 0.2484823909759521)
```

Estatística t

p-valor > 0.05 = NÃO rejeita hipótese nula
(o que você suspeitou estava correto)

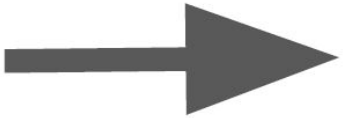
EXEMPLO B - CONTEXTUALIZAÇÃO

Imagine que estamos observando o tempo de participação, por seção, no jogo God of War 4.

EXEMPLO B

Hipótese:
Jogadores homens passam mais tempo jogando God of War ininterruptamente do que jogadoras mulheres

...	Gênero	Tempo
	fem	234
	masc	245
	fem	142



jogadorx_masc	jogadorx_fem
233	245
234	165
190	60

EXEMPLO B

Hipótese:

Jogadores homens passam mais tempo jogando God of War ininterruptamente do que jogadoras mulheres

Como você provaria assertivamente essa afirmação?

EXEMPLO B

Hipótese nula (H0):

$$\square_{HOMENS} \leq \square_{MULHERES}$$

Hipótese alternativa (H1):

$$\square_{HOMENS} > \square_{MULHERES}$$

EXEMPLO B

Hipótese nula (H0):

$$\square HOMENS \leq \square MULHERES$$

Hipótese alternativa (H1):

$$\square HOMENS > \square MULHERES$$

Como averiguar essa hipótese nula?

EXEMPLO B

Hipótese nula (H0):

$$\mu_{HOMENS} \leq \mu_{MULHERES}$$

Hipótese alternativa (H1):

$$\mu_{HOMENS} > \mu_{MULHERES}$$

Como averiguar essa hipótese nula?

Teste t de Student (de duas amostras)

EXEMPLO B

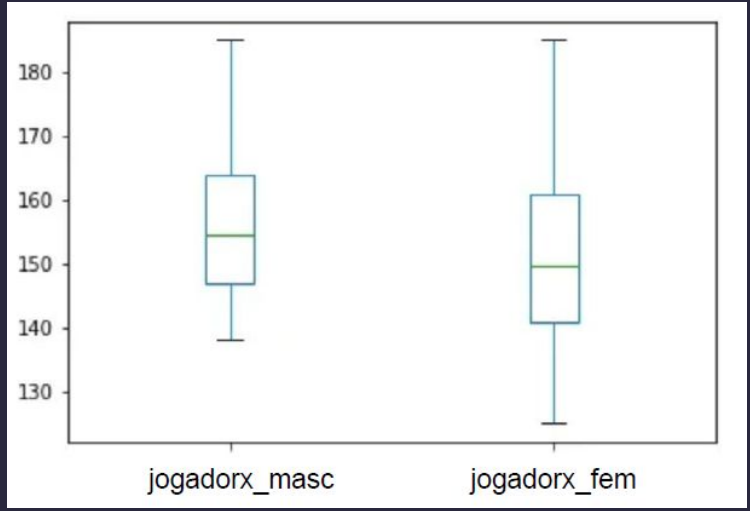
```
from scipy import stats
import matplotlib.pyplot as plt
df[['jogadorx_masc', 'jogadorx_fem']].describe()
```

	jogadorx_masc	jogadorx_fem
count	120.00	120.00
mean	156.450000	151.358333
std	11.389845	14.177622
min	138.000000	125.000000
25%	147.000000	140.750000
50%	154.500000	149.500000
75%	164.000000	161.000000
max	185.000000	185.000000

	jogadorx_masc	jogadorx_fem
count	120.00	120.00
mean	156.450000	151.358333
std	11.389845	14.177622
min	138.000000	125.000000
25%	147.000000	140.750000
50%	154.500000	149.500000
75%	164.000000	161.000000
max	185.000000	185.000000

EXEMPLO B

```
from scipy import stats
import matplotlib.pyplot as plt
df[['jogadorx_masc', 'jogadorx_fem']].plot(kind='box')
```



EXEMPLO B

```
from scipy import stats
import matplotlib.pyplot as plt

stats.shapiro(df['jogadorx_masc'])
(0.9926842451095581, 0.7841846942901611)

stats.shapiro(df['jogadorx_fem'])
(0.8384329442500341, 0.454324242455278)
```

É NORMAL!

EXEMPLO B

```
from scipy import stats
import matplotlib.pyplot as plt
stats.ttest_rel(df['jogadorx_masc'], df['jogadorx_fem'])

Ttest_relResult(statistic=3.3371870510833657,
pvalue=0.0011297914644840823)
```


The End

INFERÊNCIA ESTATÍSTICA E DATA MINING

AULA 03