

Aprendizagem de Máquina 1

Inteligência Artificial



AULA 07 – APRENDIZADO NÃO SUPERVISIONADO

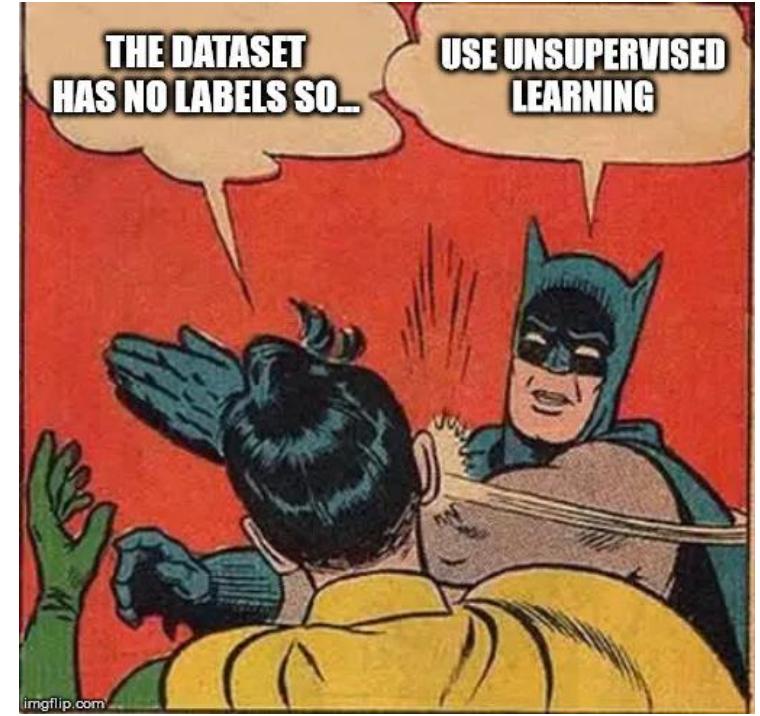
Larissa Driemeier
Thiago Martins

CRONOGRAMA

Data	Professor	Assunto
07/05	Larissa	Definição de aprendizado de máquina. Aprendizado supervisionado e não supervisionado. Regressão linear. Regressão polinomial.
14/05	Thiago	Exercícios de acompanhamento. Nota 01.
21/05	Larissa	Régressão Logística
28/05	Thiago	Exercícios de acompanhamento. Nota 02.
04/06	Larissa	Máquinas de vetores de suporte
11/06	Thiago	Exercícios de acompanhamento. Nota 03.
18/06	Larissa	Aprendizado não supervisionado
25/06	Thiago	Exercícios de acompanhamento. Nota 04.
02/07	Larissa	Redução de similaridade: análise de componentes principais (PCA) e suas variações.
16/07	Larissa/Thiago	Exercícios “Melhores Momentos”. Nota 05.



O QUE É APRENDIZADO NÃO SUPERVISIONADO OU AUTO SUPERVISIONADO?



Definição

Tipos de aprendizado não supervisionado

IMAGENET

1000 categorias

~1.2 milhões de imagens
com 256x256 pixels

Treinamento: 1000
imagens para cada
categoria (1 000 000
imagens)

Validação: 50k imagens

Teste: 150k imagens

0: tench, Tinca tinca



265:
toy
poodle

920: traffic
light, traffic
signal, stoplight



1: goldfish



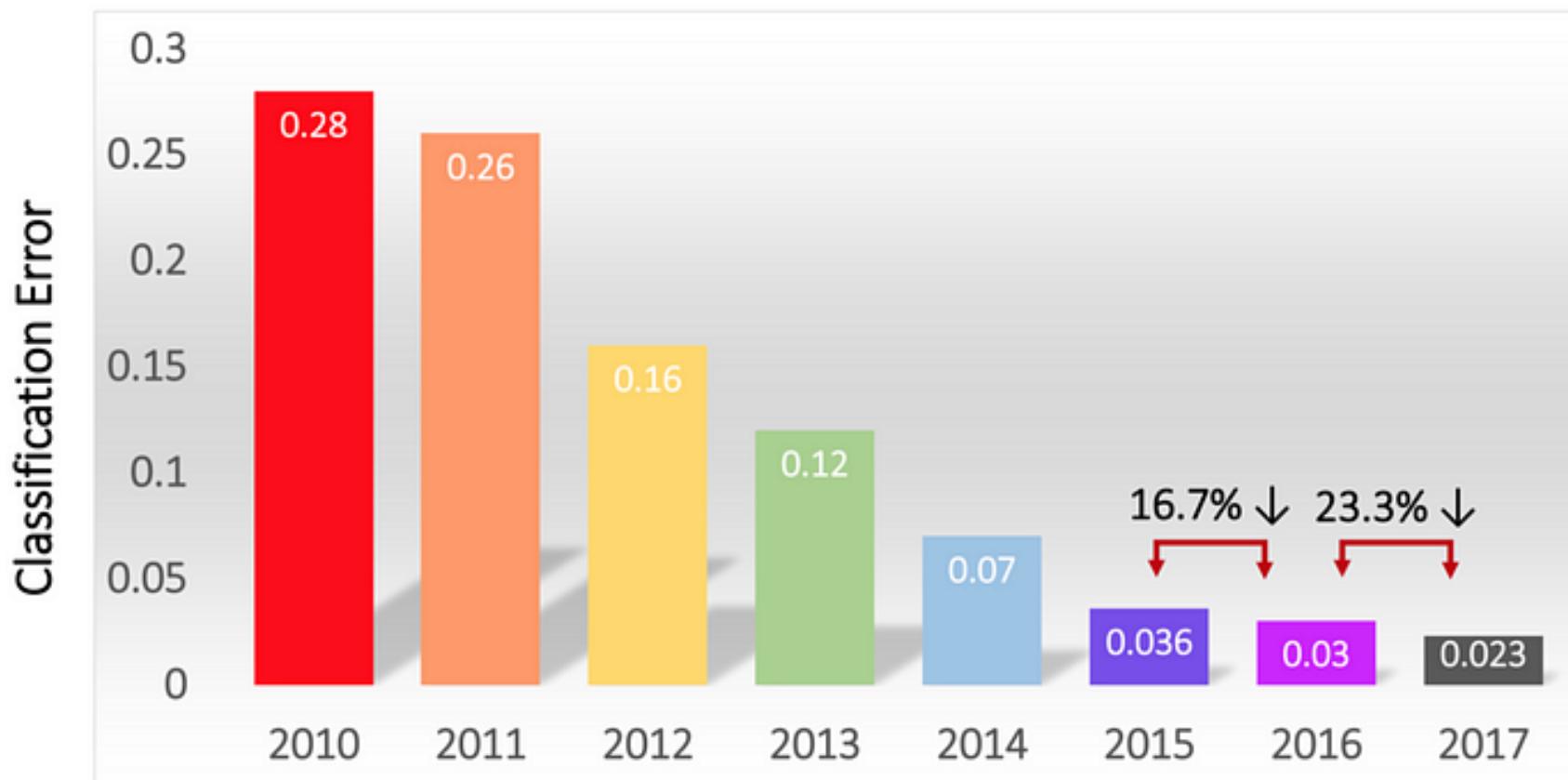
558: lute,
transverse
flute



999: toilet
tissue, toilet
paper,
bathroom
tissue

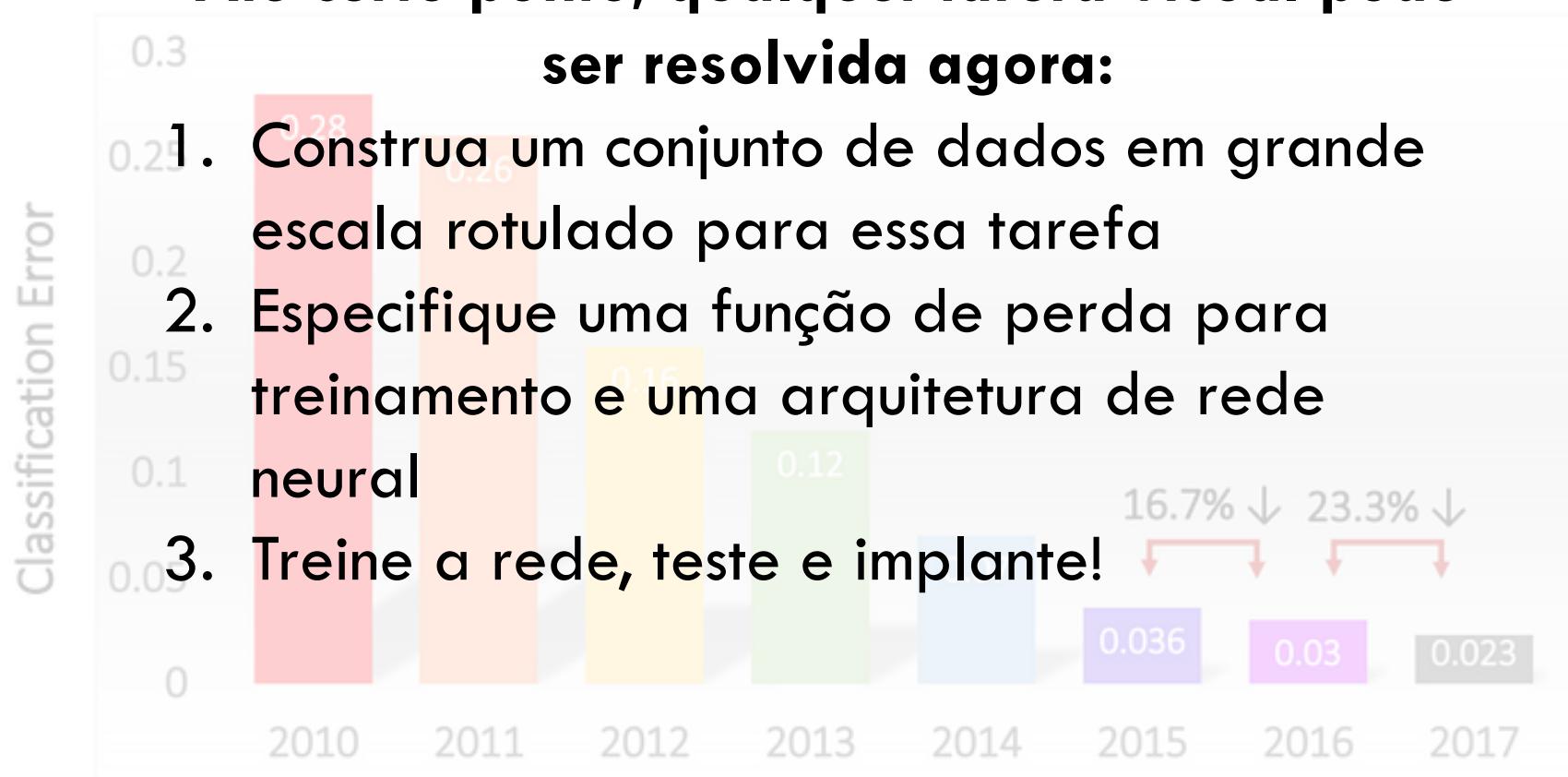


Classification Results (CLS)



Classification Results (CLS)

Até certo ponto, qualquer tarefa visual pode ser resolvida agora:



PORQUE USAR APRENDIZADO AUTO SUPERVISIONADO?

Custo de produção de um novo conjunto de dados para cada nova tarefa:

Restrições de Custo e Tempo: significativos custos financeiros e de tempo podem estar associados à criação de conjuntos de dados rotulados para cada nova tarefa, incluindo a necessidade de conhecimento especializado e trabalho manual. Destaca-se o problema da escalabilidade, especialmente para empresas que desejam implantar vários modelos de IA em diferentes domínios.

PORQUE USAR APRENDIZADO AUTO SUPERVISIONADO?

Custo de produção de um novo conjunto de dados para cada nova tarefa:

Restrições de Custo e Tempo: significativos custos financeiros e de tempo podem estar associados à criação de conjuntos de dados rotulados para cada nova tarefa, incluindo a necessidade de conhecimento especializado e trabalho manual. Destaca-se o problema da escalabilidade, especialmente para empresas que desejam implantar vários modelos de IA em diferentes domínios.

Áreas com escassez de supervisão:

Dados Médicos: escassez de dados médicos rotulados devido a preocupações com a privacidade, a necessidade de rotulagem por especialistas e as implicações éticas do uso de dados de pacientes.

Dados Legais e Financeiros: nos setores jurídico e financeiro, obter dados rotulados é desafiador devido à confidencialidade e à complexidade dos dados.

PORQUE USAR APRENDIZADO AUTO SUPERVISIONADO?

Custo de produção de um novo conjunto de dados para cada nova tarefa:

Restrições de Custo e Tempo: significativos custos financeiros e de tempo podem estar associados à criação de conjuntos de dados rotulados para cada nova tarefa, incluindo a necessidade de conhecimento especializado e trabalho manual. Destaca-se o problema da escalabilidade, especialmente para empresas que desejam implantar vários modelos de IA em diferentes domínios.

Áreas com escassez de supervisão:

Dados Médicos: escassez de dados médicos rotulados devido a preocupações com a privacidade, a necessidade de rotulagem por especialistas e as implicações éticas do uso de dados de pacientes.

Dados Legais e Financeiros: nos setores jurídico e financeiro, obter dados rotulados é desafiador devido à confidencialidade e à complexidade dos dados.

Disponibilidade inexplorada de um grande número de imagens/vídeos não rotulados:

Mídias Sociais e Plataformas Online: enorme volume de imagens e vídeos não rotulados disponíveis em plataformas como Facebook e YouTube, que representam uma rica fonte de dados para aprendizado não supervisionado.

Facebook: Mais de um bilhão de imagens são carregadas por dia, fornecendo um conjunto de dados vasto e diversificado.

YouTube: 300 horas de vídeo são carregadas a cada minuto, oferecendo extensos dados multimídia para treinamento de modelos.

Crowdsourcing e IoT: potencial de aproveitar dados de plataformas de crowdsourcing e dispositivos da Internet das Coisas (IoT) para reunir conjuntos de dados não rotulados em larga escala.

MOTIVAÇÃO COGNITIVA: COMO BEBÊS APRENDEM?



Bebês aprendem a reconhecer objetos e padrões sem supervisão explícita

Give a robot a label and you fees it for a second, teach a robot to label and you fees it for a lifetime.

Pierre Sermanet



APRENDIZADO NÃO SUPERVISIONADO

A aprendizagem não supervisionada é um conjunto de ferramentas estatísticas para cenários em que existe apenas um conjunto de recursos e nenhum rótulo.

Não podemos fazer previsões, uma vez que não existem respostas associadas a cada observação. Em vez disso, estamos interessados em encontrar uma forma interessante de visualizar dados ou em descobrir subgrupos de observações semelhantes.

Duas técnicas serão o foco desta aula: análise de componentes principais (PCA, do inglês, Principal Component Analysis) e agrupamento (clustering).

ANALOGIA DO BOLO DE YANN LECUN

■ "Pure" Reinforcement Learning (cherry)

- ▶ The machine predicts a scalar reward given once in a while.
- ▶ **A few bits for some samples**

■ Supervised Learning (icing)

- ▶ The machine predicts a category or a few numbers for each input
- ▶ Predicting human-supplied data
- ▶ **10→10,000 bits per sample**

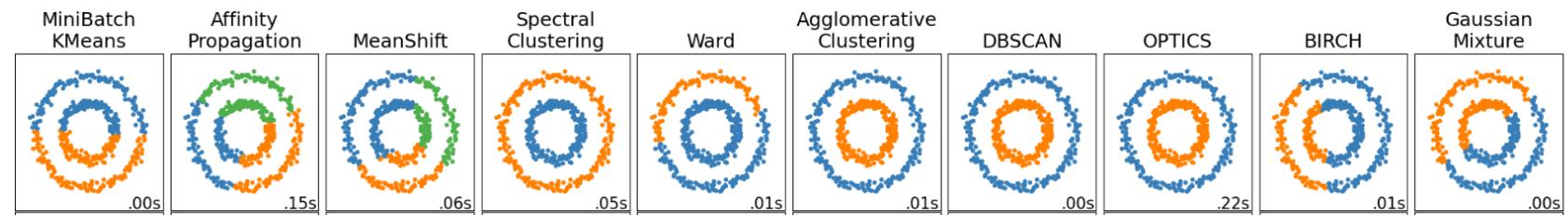
■ Unsupervised/Predictive Learning (cake)

- ▶ The machine predicts any part of its input for any observed part.
- ▶ Predicts future frames in videos
- ▶ **Millions of bits per sample**

■ (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)



Técnicas de aprendizado de máquina como o aprendizado supervisionado (que apenas prevê rótulos fornecidos por humanos) e o aprendizado por reforço (que apenas prevê uma função de valor) são muito restritas para criar máquinas inteligentes em nível humano. O aprendizado não supervisionado, no entanto, com seus milhões de bits de informação por amostra, pode ser usado para treinar máquinas altamente complexas sem supervisão humana.



ALGORITMOS DE CLUSTERIZAÇÃO

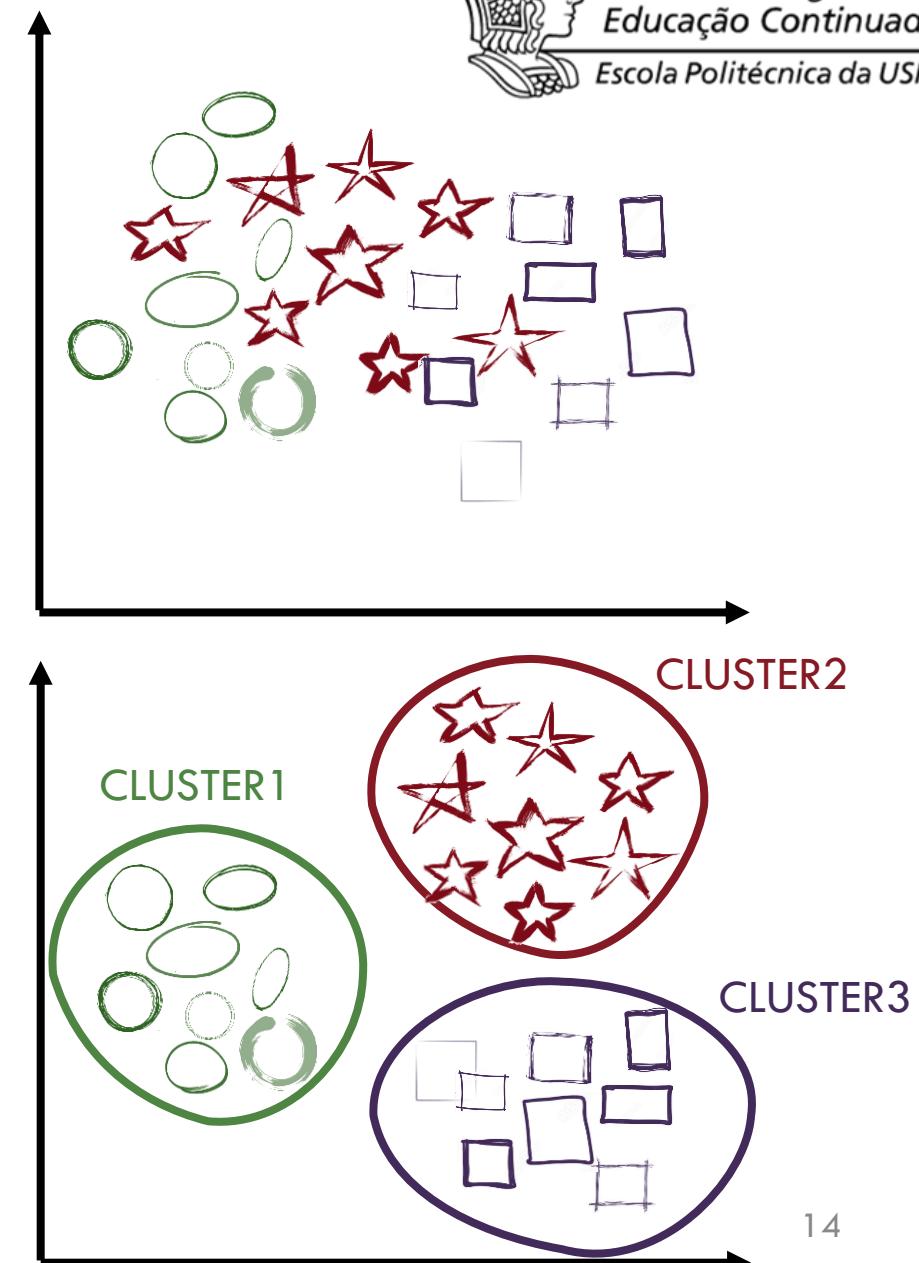
K-MEANS

```
from sklearn.cluster import KMeans
```

Um dos algoritmos mais populares e amplamente utilizados para tarefas de cluster é o k-means.

Um algoritmo de agrupamento K-means tenta agrupar itens semelhantes na forma de clusters. O número de grupos é representado por K.

O algoritmo iterativo baseado em centroide que cria clusters não sobrepostos.

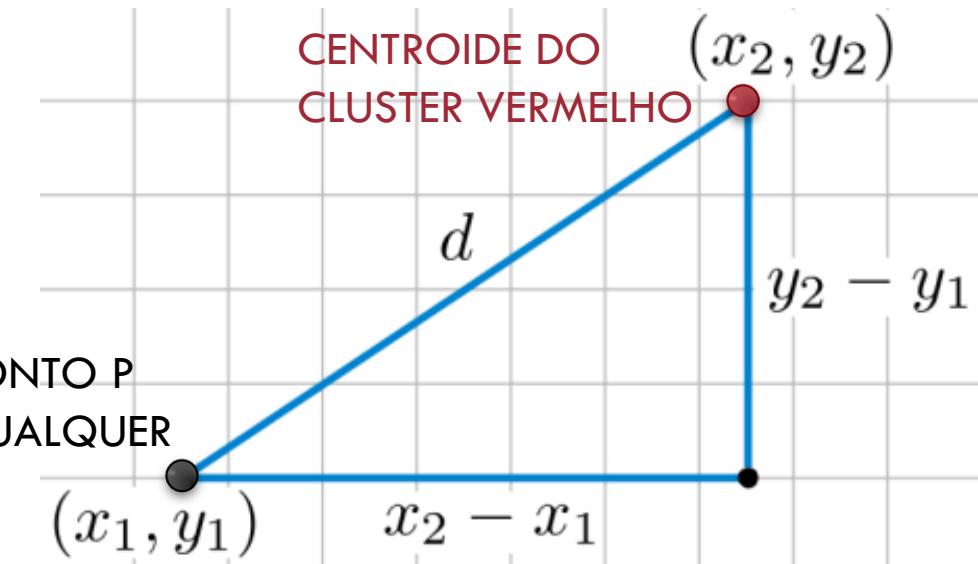


COMO FUNCIONA O ALGORITMO DE AGRUPAMENTO K-MEANS?

O algoritmo executa uma iteração inicial onde os centroides são escolhidos aleatoriamente.

A distância euclidiana d de cada ponto de dados a cada centroide é calculada e o ponto é alocado ao centroide de menor distância.

O novo centroide é calculado e o algoritmo começa outra iteração (calcula os centroides, calcula as distâncias, realoca pontos) e o processo continuará até que todos os agrupamentos tenham a variação mínima dentro do grupo.



$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

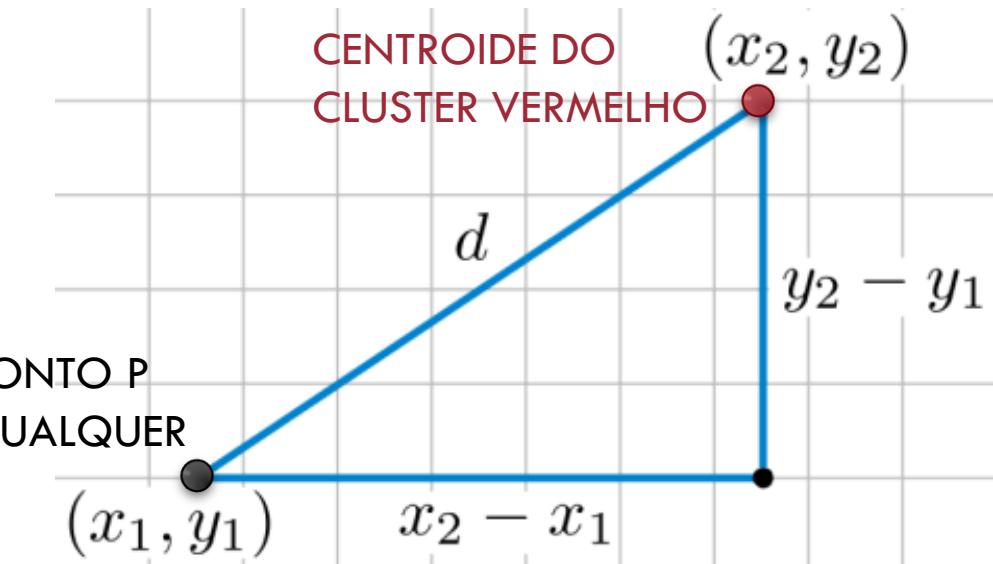
COMO FUNCIONA O ALGORITMO DE AGRUPAMENTO K-MEANS?



VAMOS PASSO A
PASSO EM UM
EXEMPLO?

O algoritmo executa uma iteração inicial com centroides aleatoriamente escolhidos. A distância d de cada ponto de dados aos centroides é calculada e o ponto é alocado ao centroide de menor distância.

Quando isso acontecer, o algoritmo executará outra iteração (calcula os centroides, calcula as distâncias, realoca pontos) e o processo continuará até que todos os agrupamentos tenham a variação mínima dentro do grupo.



$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

EXEMPLO – PARTE 1

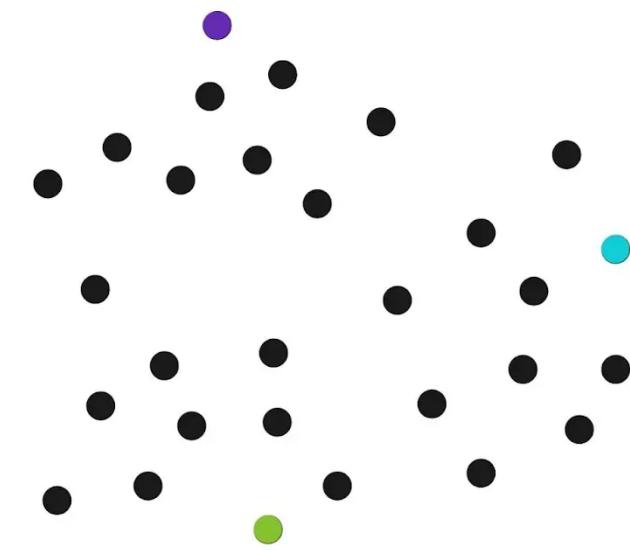
A partir dos dados, podemos identificar três clusters. Ao ajustar nosso modelo, podemos atribuir aleatoriamente $K = 3$. Isso simplesmente significa que estamos buscando dividir os pontos de dados em três agrupamentos.

Na iteração inicial, os K centroides são selecionados aleatoriamente



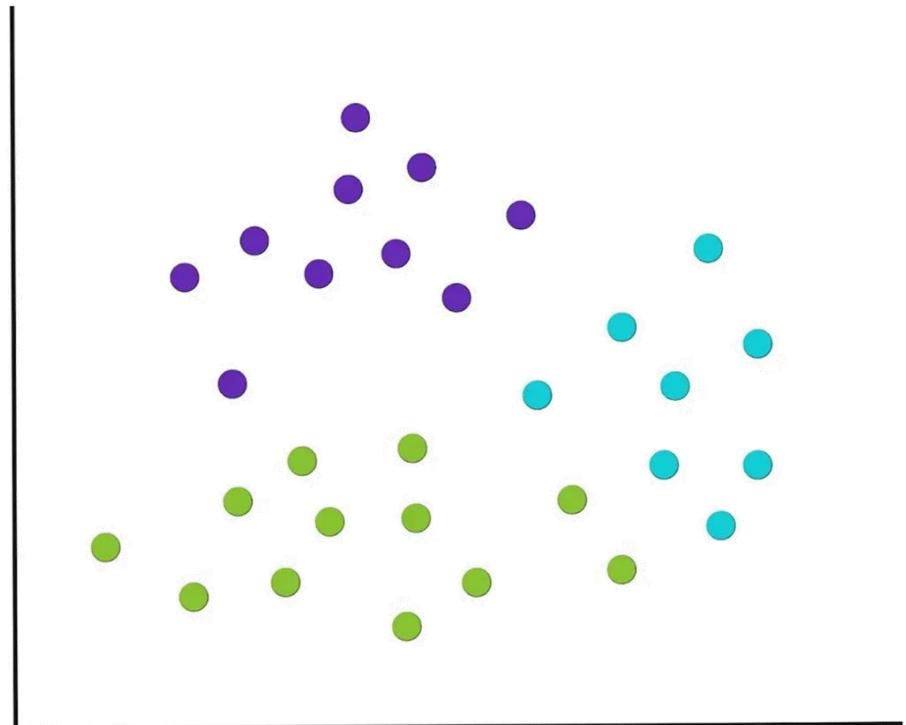
Mas eu posso não conseguir enxergar em quantos grupos dividir?

K é um hiperparâmetro, um dado de entrada. Aprendemos mais adiante algumas técnicas para escolhermos o melhor valor de K



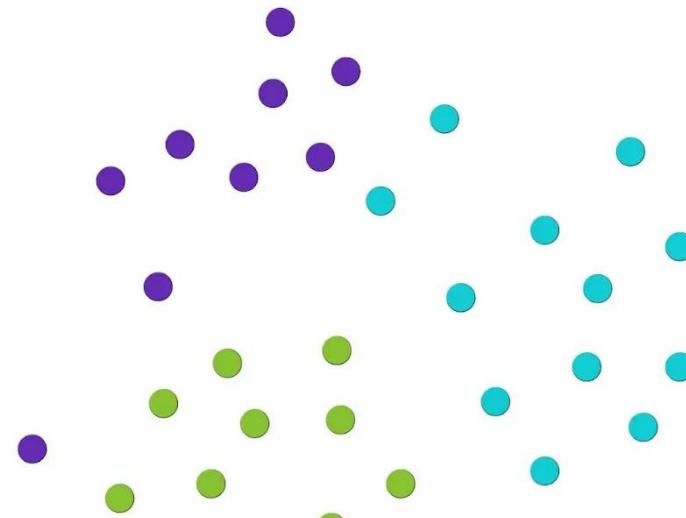
EXEMPLO – PARTE 2

Uma vez que os centroides foram selecionados, cada ponto de dado é atribuído ao centroide mais próximo, com base na distância euclidiana do ponto ao centroide mais próximo.



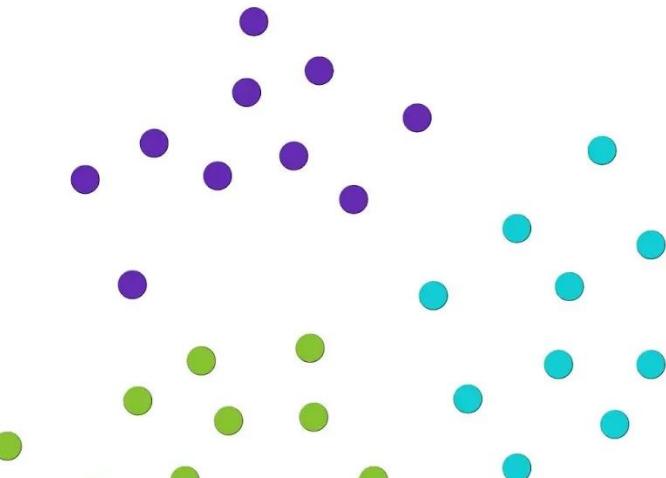
EXEMPLO – PARTE 3

Após a primeira rodada de agrupamentos, novos centroides são calculados e isso necessitará uma realocação dos pontos aos respectivos clusters.

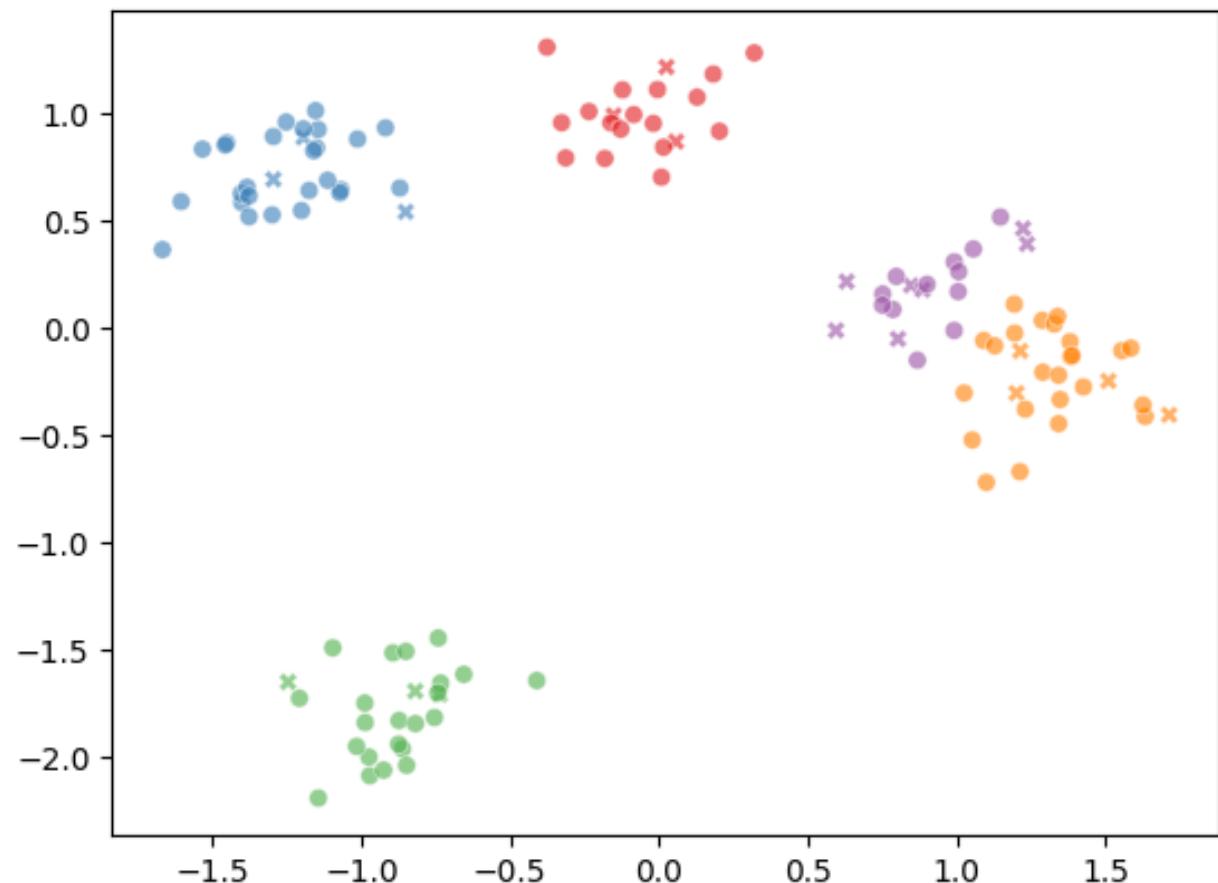
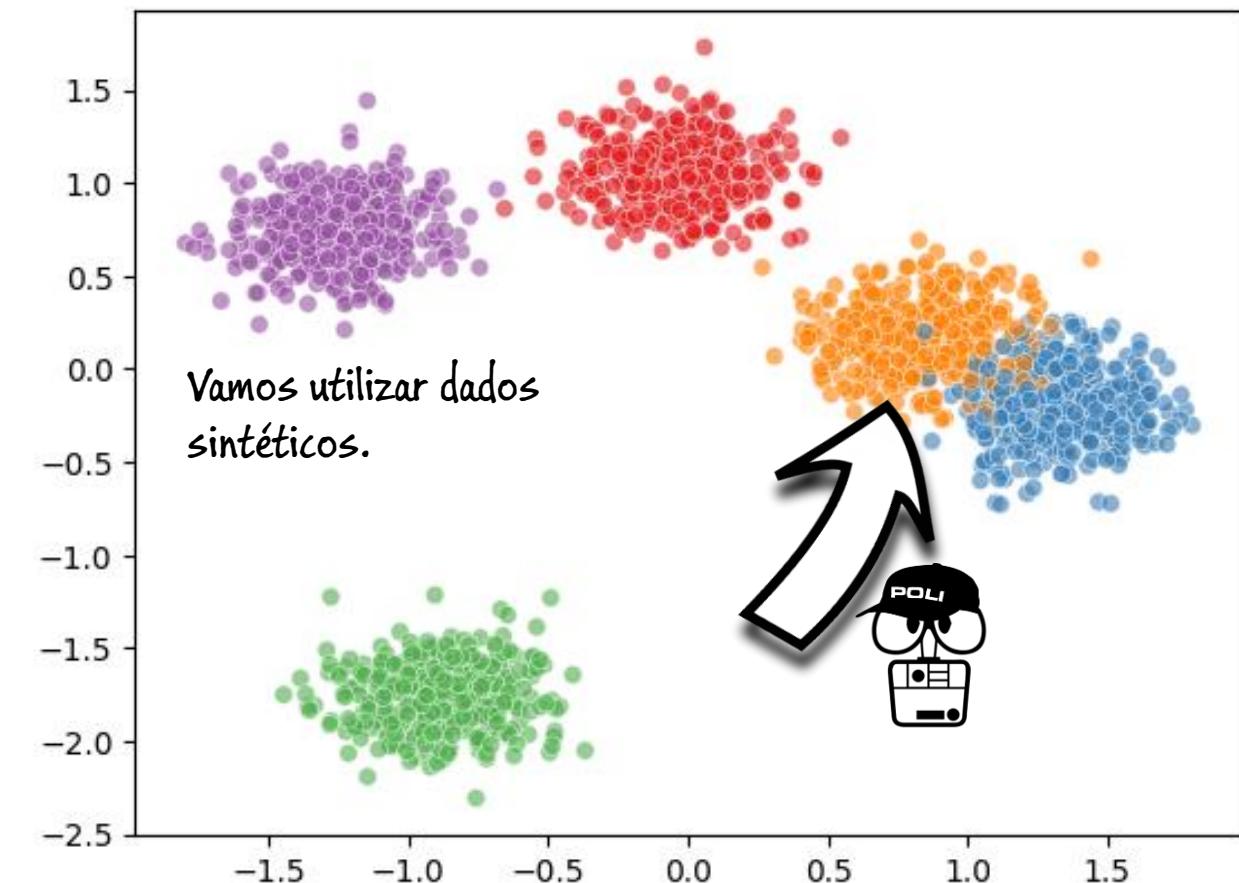


EXEMPLO – PARTE 4

O processo nos passos 2 e 3 é repetido até chegarmos a um ponto onde não há mais realocação dos pontos de dados ou alcançamos o número máximo de iterações.

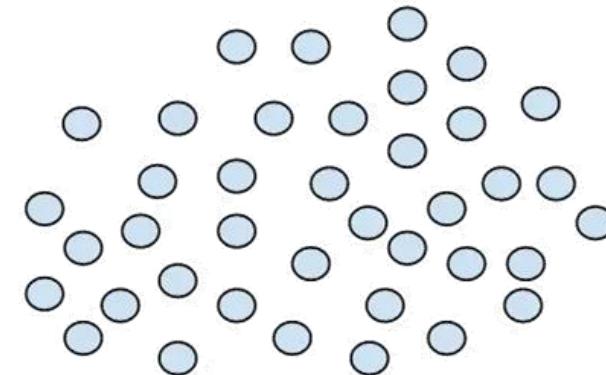


ALGORITMO DE K-MEANS NO NOTEBOOK



A ESCOLHA DE K

Os dados com os quais você trabalhará nem sempre terão demarcações distintas quando plotados. Aliás, muitas vezes, você lidará com dados de dimensões altas, que não podem ser plotados. Ou mesmo que sejam plotados, você não poderá determinar o número ótimo de agrupamentos.



Você consegue dizer o número de agrupamentos?
Não claramente. Então, como encontraremos o número ótimo de clusters nos quais os pontos de dados acima podem ser agrupados?

MÉTODO ELBOW (COTOVelo)

Esta abordagem usa as variações totais dentro de um cluster, também conhecidas como WCSS (Within Cluster Sum of Squares). O WCSS mede a distância euclidiana ao quadrado entre cada ponto e o centroide e calcula a diferença quadrada entre os dois. Daí o nome: soma dos quadrados dentro do cluster.

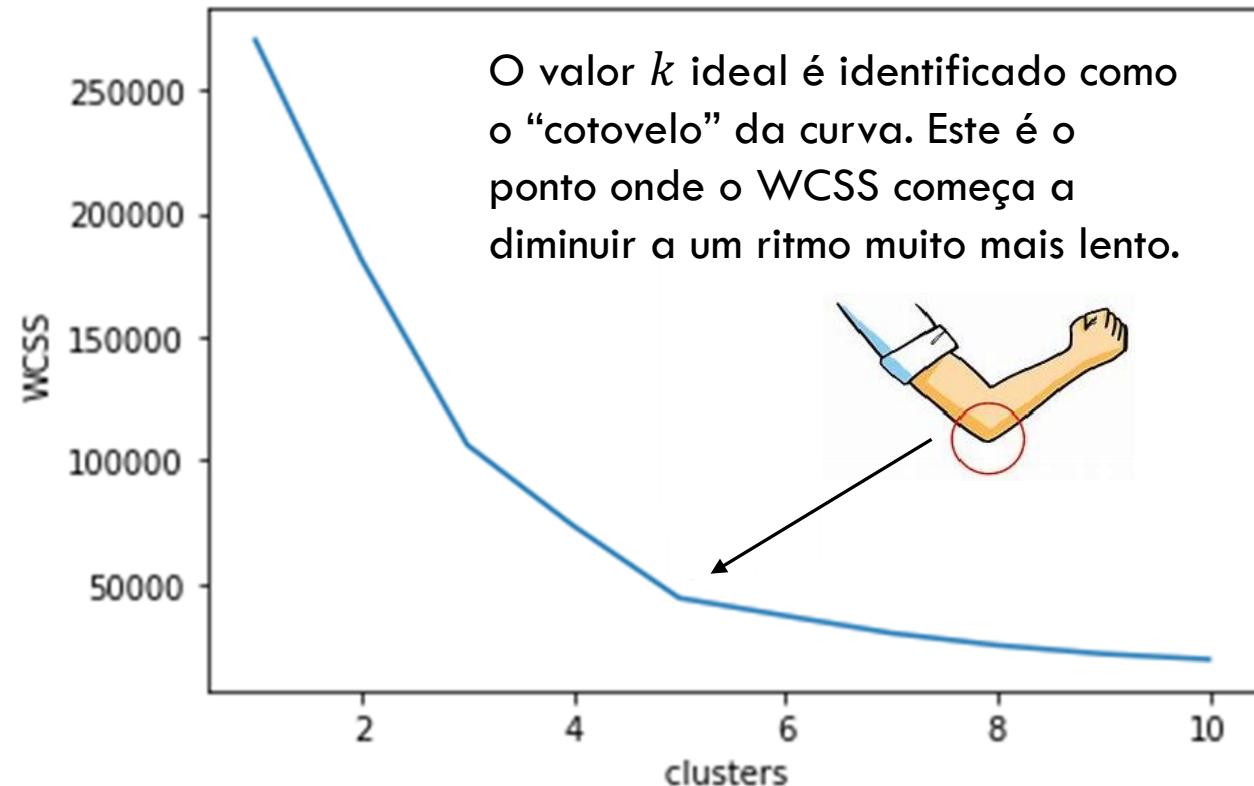
O objetivo é ter a variação mínima dentro dos clusters.



$$WCSS = \sum_{x \text{ em } \bullet} (x_j - x_{C_1})^2 + (y_j - y_{C_1})^2 + \\ \sum_{x \text{ em } \bullet} (x_j - x_{C_2})^2 + (y_j - y_{C_2})^2 + \\ \sum_{x \text{ em } \bullet} (x_j - x_{C_3})^2 + (y_j - y_{C_3})^2$$

PORQUE COTOVELO?

À medida que você aumenta o número de clusters, a soma das distâncias quadradas entre os pontos e seus centros de cluster (WCSS) continuará a diminuir. Isso ocorre porque você está basicamente dividindo os dados em grupos cada vez mais específicos.



Em essência, você está procurando o ponto em que adicionar mais clusters não melhora significativamente o ajuste dentro dos clusters.

MÉTODO DA SILHUETA (SILHOUETTE)

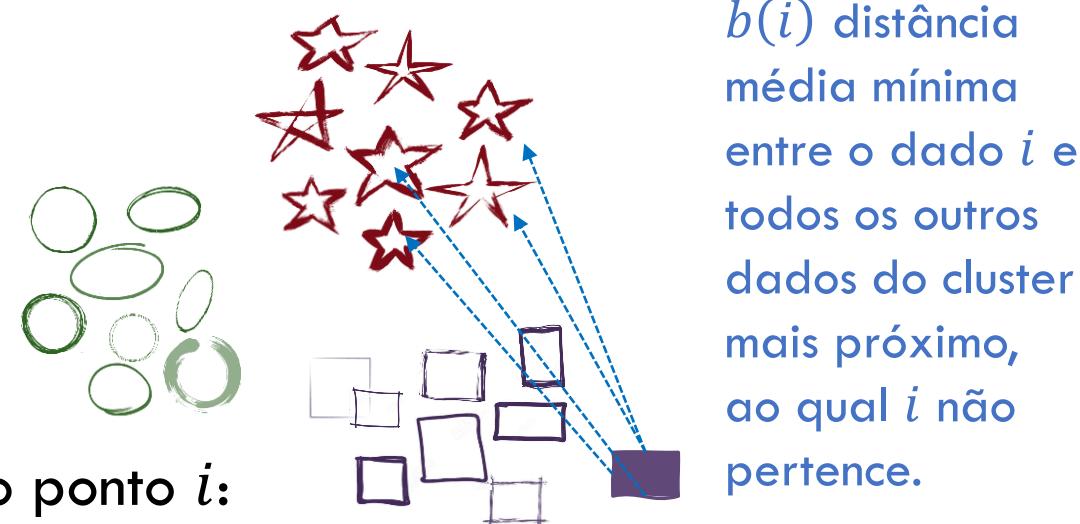
Este método mede similaridade e dissimilaridade. Ele quantifica a distância de um ponto para outros membros do seu cluster e também a distância para os membros em outros clusters.



$a(i)$ distância média entre o dado i e todos os outros dados do mesmo cluster.

Coeficiente de silhueta do ponto i :

$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]} \quad \text{Valores entre } [-1,1]$$

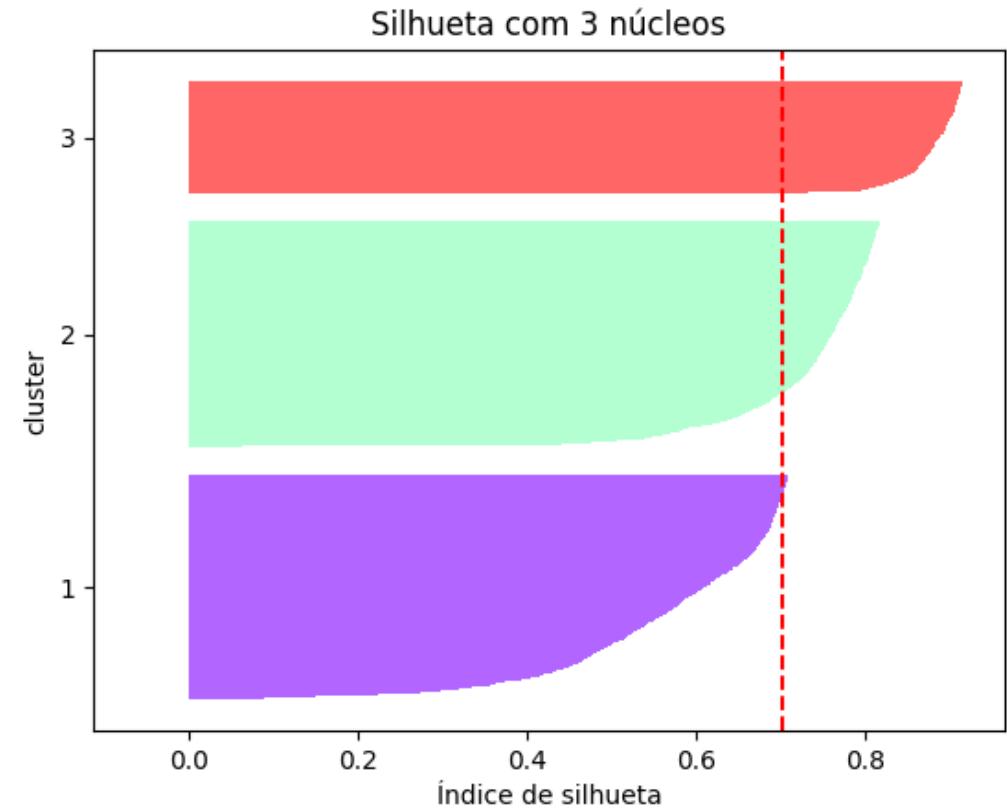
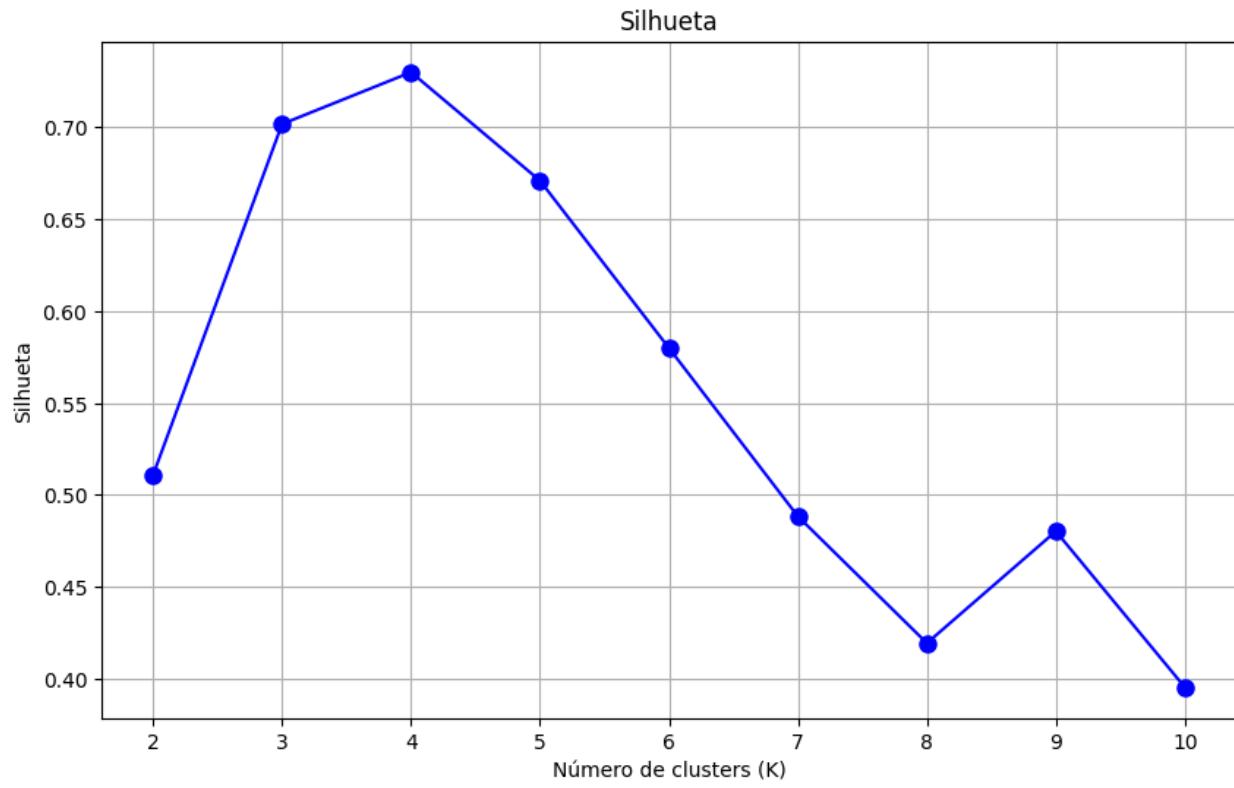


$b(i)$ distância média mínima entre o dado i e todos os outros dados do cluster mais próximo, ao qual i não pertence.

ALGORITMO PARA CÁLCULO DA SILHUETA

1. Define-se uma faixa de valores K começando com 2.
2. Para cada valor de K , calcula-se a similaridade do cluster, que é a distância média entre um ponto de dados e todos os outros membros do mesmo cluster.
3. Em seguida, a dissimilaridade do cluster é calculada ao se calcular a distância média entre um ponto de dados e todos os outros membros do cluster mais próximo.
4. O coeficiente de silhueta será a diferença entre o valor da similaridade do cluster e o valor da dissimilaridade do cluster, dividida pelo maior dos dois valores.
5. O K ótimo seria aquele com o maior coeficiente. Os valores desse coeficiente estão limitados na faixa de -1 a 1 .

NOTEBOOK



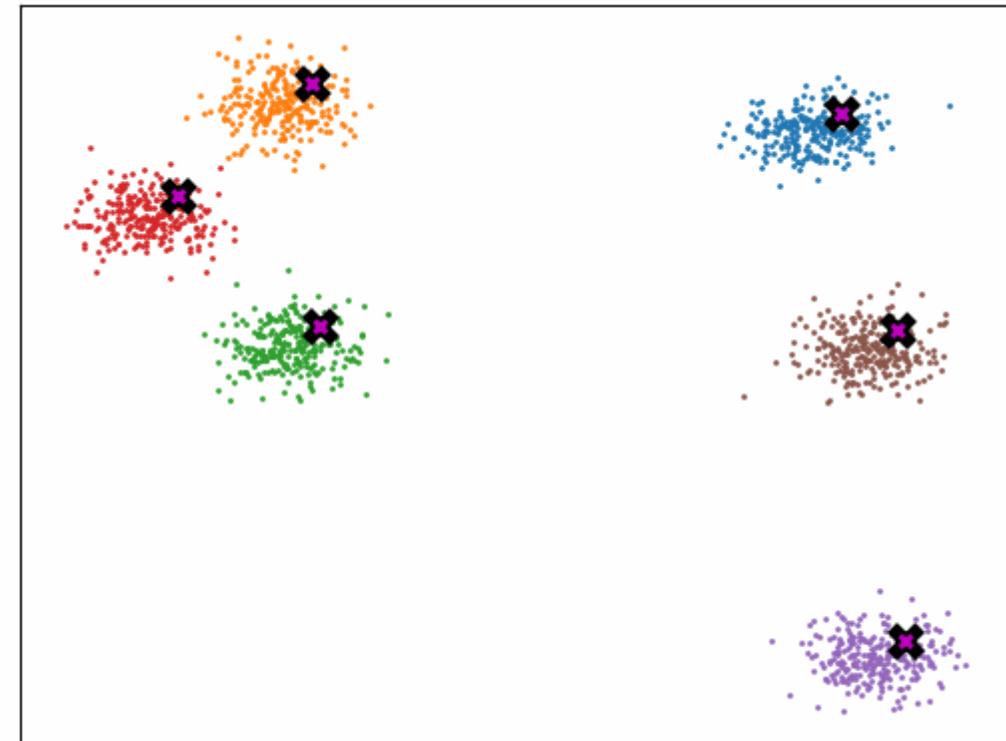
FATORES QUE IMPACTAM A EFICÁCIA DO MODELO

- 1. Número de clusters (K):** O número de clusters nos quais você deseja agrupar seus pontos de dados deve ser predefinido.
- 2. Valores/Seeds Iniciais:** A escolha dos centros iniciais do cluster pode ter um impacto na formação final do cluster. O algoritmo K-means não é determinístico. Isto significa que o resultado do agrupamento pode ser diferente cada vez que o algoritmo é executado, mesmo no mesmo conjunto de dados.
- 3. Outliers:** A formação de clusters é muito sensível à presença de outliers. Os outliers puxam o cluster para si, afetando assim sua formação ideal.
- 4. Medidas de distância:** o uso de diferentes medidas de distância (usadas para calcular a distância entre um ponto de dados e o centro do cluster) pode gerar clusters diferentes.
- 5.** O algoritmo K-Means não funciona com dados categóricos.
- 6.** O processo pode não convergir no determinado número de iterações. Você deve sempre verificar a convergência.

ALGORITMO DE MOVER PARA MÉDIA – MEANSHIFT (MSC)

```
from sklearn.cluster import MeanShift
```

1. Não requer a especificação do número de clusters. O próprio algoritmo determina automaticamente o número de clusters;
2. MSC também é baseado em centroides e atribui iterativamente cada ponto de dados a clusters;
3. O caso de uso mais comum para MSC são tarefas de segmentação de imagens;
4. É um método de agrupamento baseado em densidade que se concentra em encontrar as regiões de alta densidade e deslocar iterativamente os pontos de dados em direção à maior densidade de pontos.



COMO FUNCIONA O ALGORITMO?

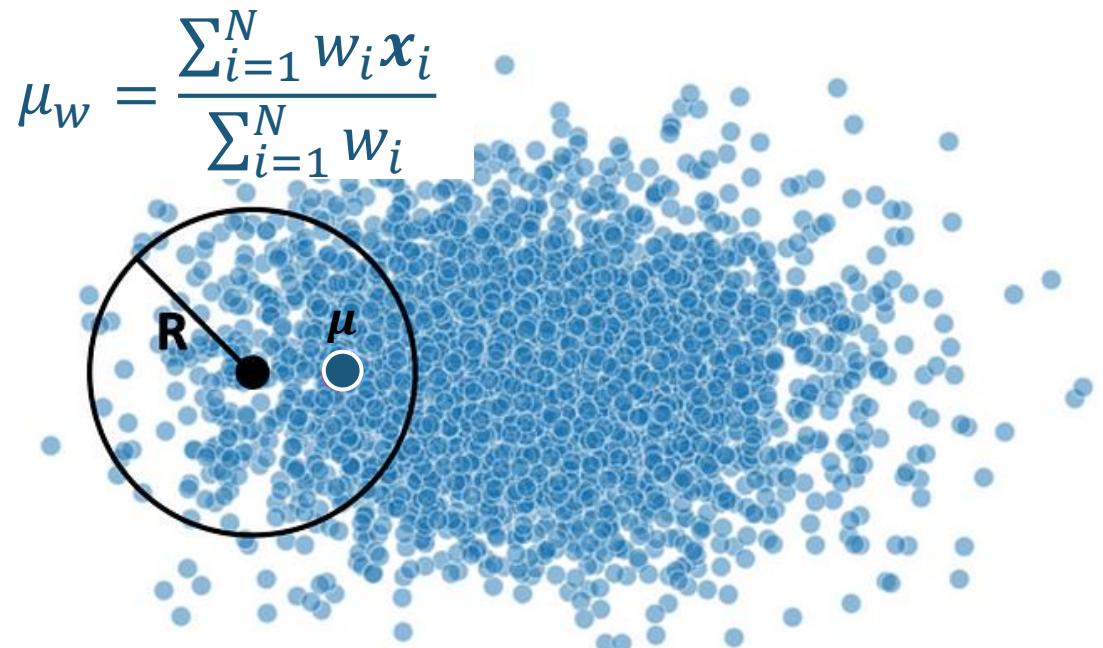
Estimativa de densidade do kernel: estima-se a função de densidade de probabilidade (PDF) subjacente dos pontos de dados. Isso normalmente é feito usando a estimativa de densidade do kernel, onde cada ponto de dados é representado por uma função do kernel centrada naquele ponto. A função kernel especifica o peso atribuído a cada ponto de dados no processo de estimativa de densidade.

Mudança de pontos de dados: o algoritmo desloca iterativamente os pontos de dados para regiões de maior densidade. O vetor de deslocamento médio é calculado como a média ponderada das diferenças entre o ponto de dados e seus pontos vizinhos, onde os pesos são determinados pela função kernel.

Convergência e Identificação de Cluster: O algoritmo continua deslocando os pontos de dados até que a convergência seja alcançada. A convergência ocorre quando os vetores de deslocamento médio tornam-se muito pequenos ou insignificantes.

MEAN SHIFT

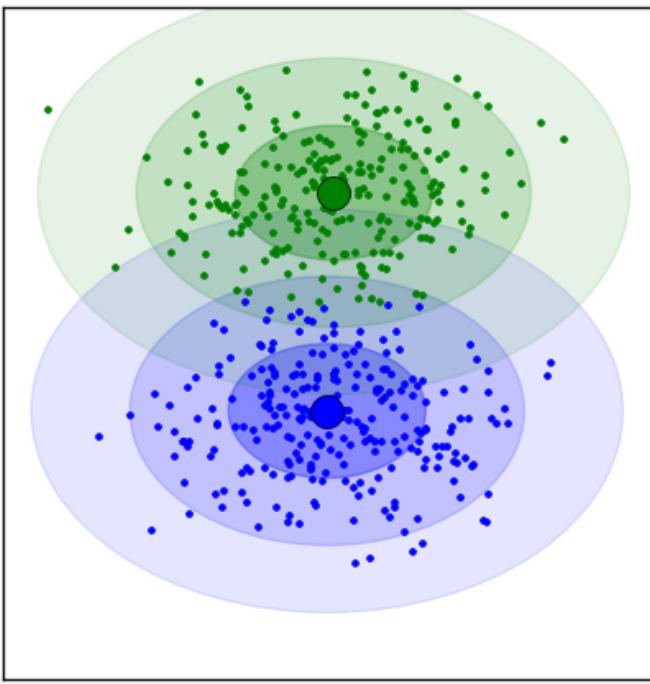
Porque média?



O ponto μ médio local que calculamos acima representa a posição com a maior densidade de pontos naquela área local específica. Note que o raio R do círculo preto é muito importante para definir a **região local**.

Na verdade, o raio, que é chamado de “largura de banda” (*bandwidth*), e o *kernel* utilizado são os principais hiperparâmetros no algoritmo de MSC.

Gaussian kernel

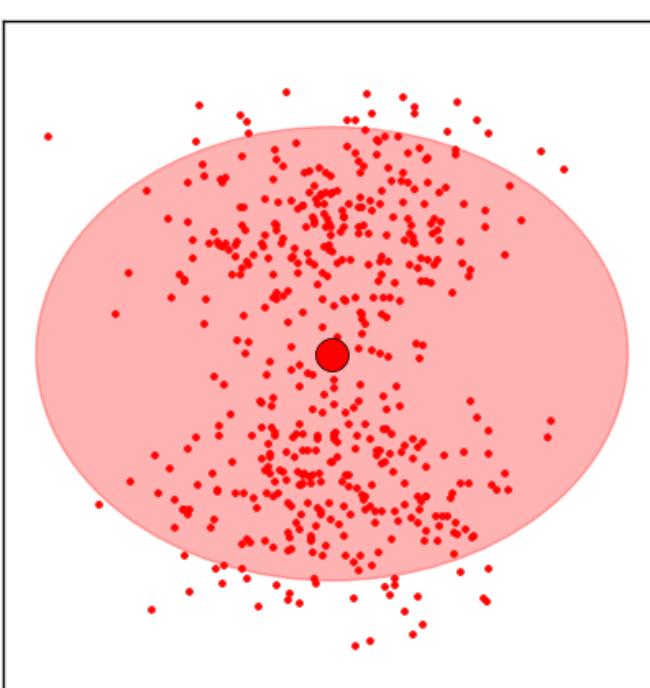


$$w_i(d) = \frac{1}{2\pi} e^{-\frac{d^2}{\sigma^2}}$$

O peso de cada ponto decai exponencialmente à medida que a distância do centro do kernel aumenta.

O kernel define os pesos.

Flat kernel



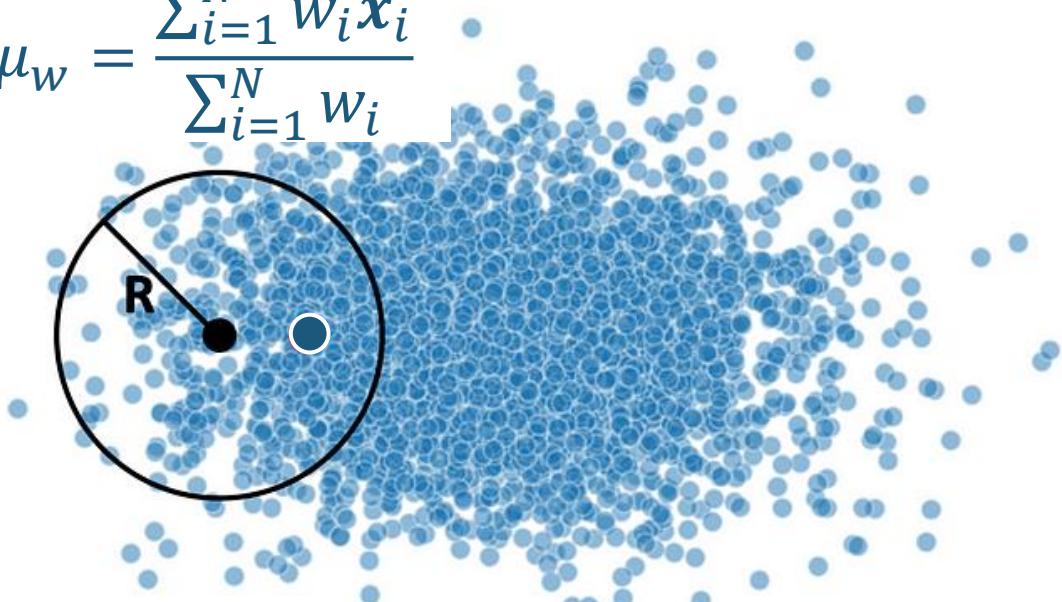
$$w_i(d) = \begin{cases} 1 & \text{se } d \leq R \\ 0 & \text{se } d > R \end{cases}$$

Todos os pontos dentro do raio R tem o mesmo peso unitário.

MEAN SHIFT

Porque média?

$$\mu_w = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i}$$

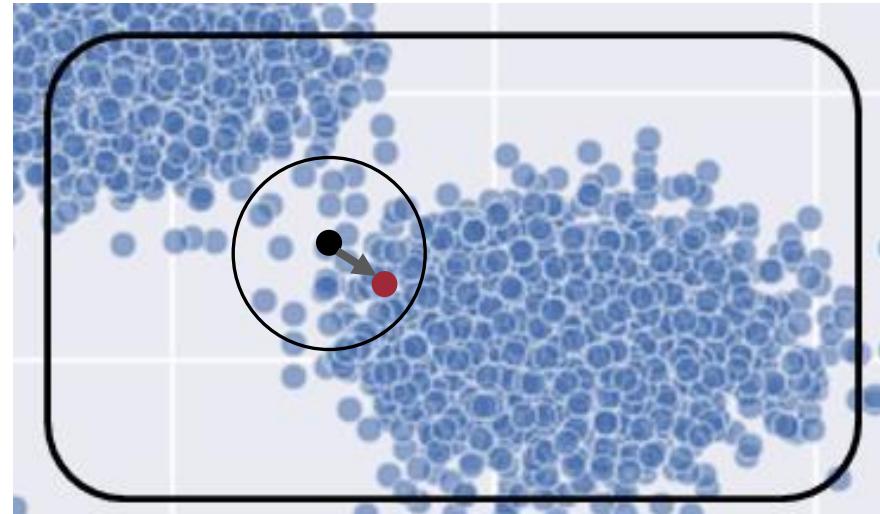
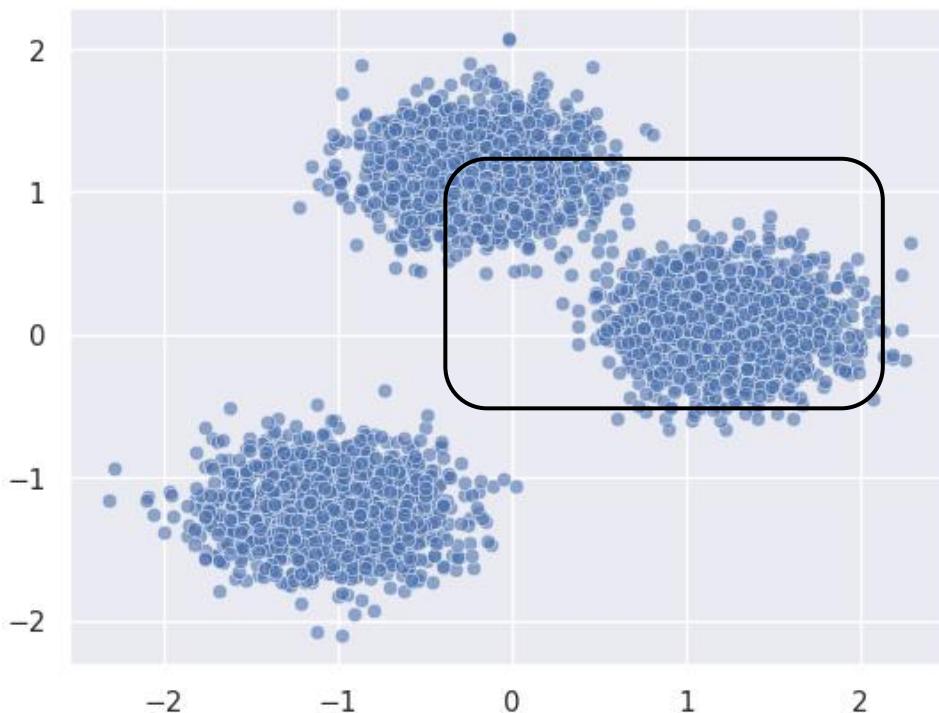


$$w_i(d) = \begin{cases} 1 & \text{se } d \leq R \\ 0 & \text{se } d > R \end{cases}$$

$$w_i(d) = e^{-\frac{d}{2\sigma^2}}$$

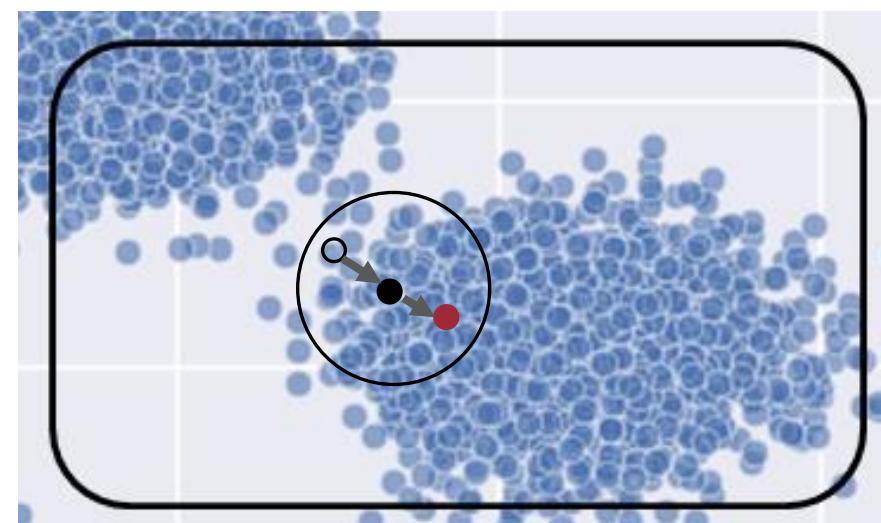
Porque shift?

Shift apenas descreve o processo de como rotulamos cada ponto de dados. Pontos semelhantes gradualmente “deslocam-se” para o centroide do cluster a que pertencem, para serem rotulados como tal.



- Ponto investigado
- Ponto médio local

Então o centro do círculo passa do ponto preto para o vermelho.



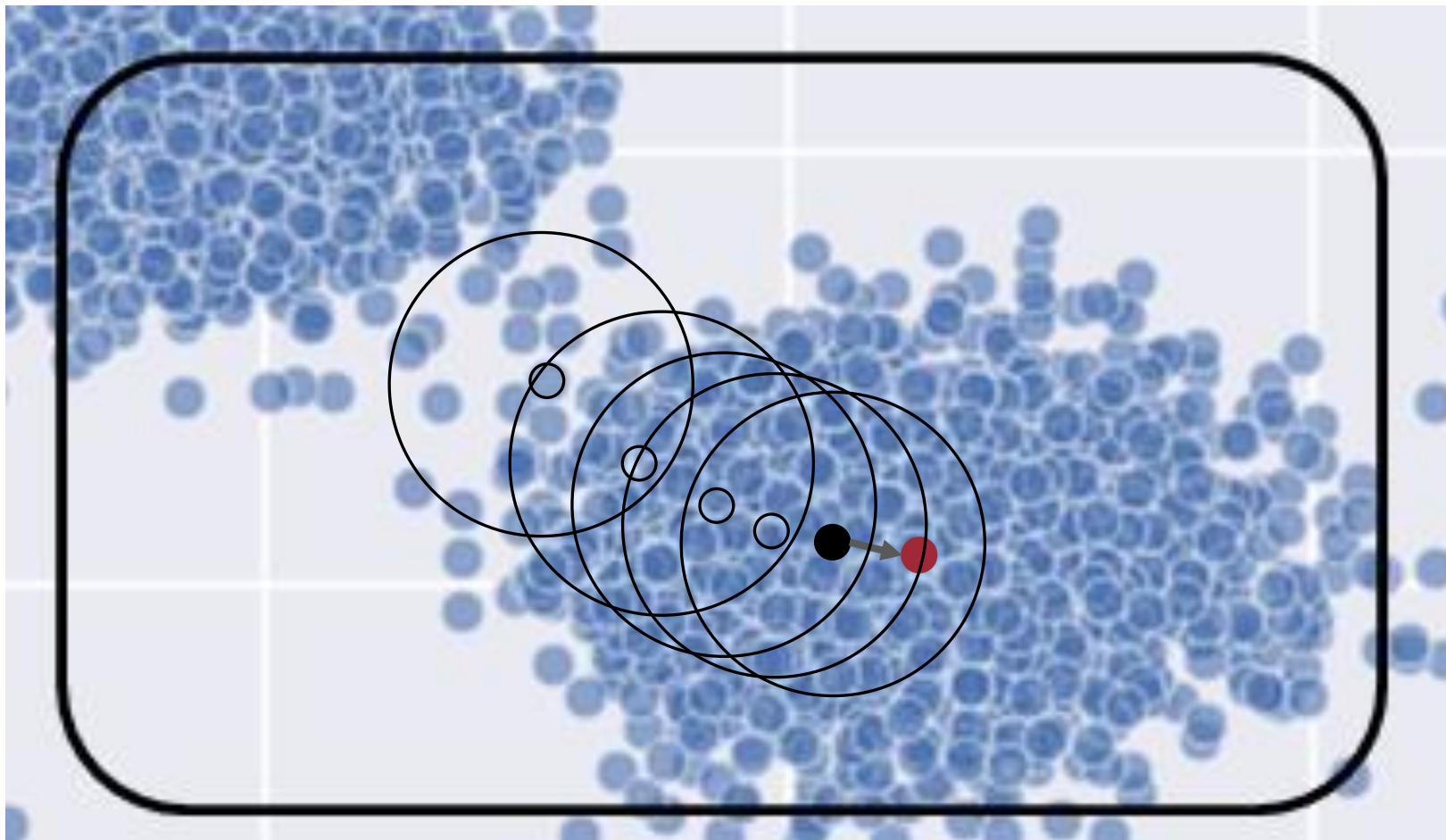
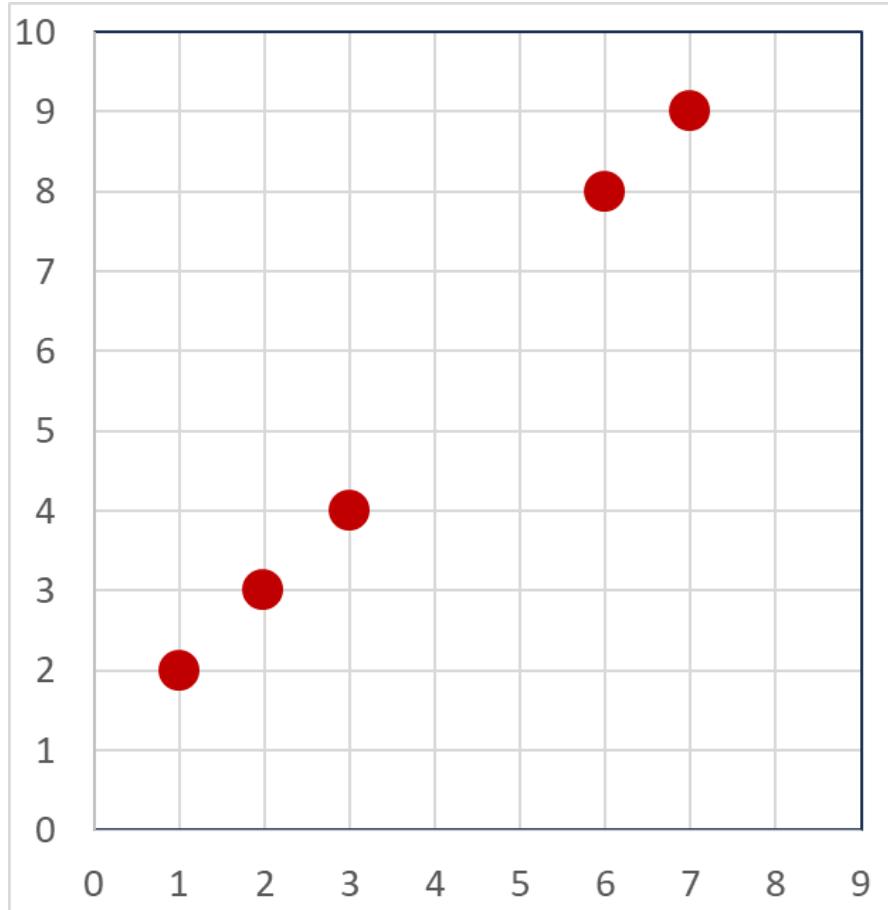


ILUSTRAÇÃO DO ALGORITMO

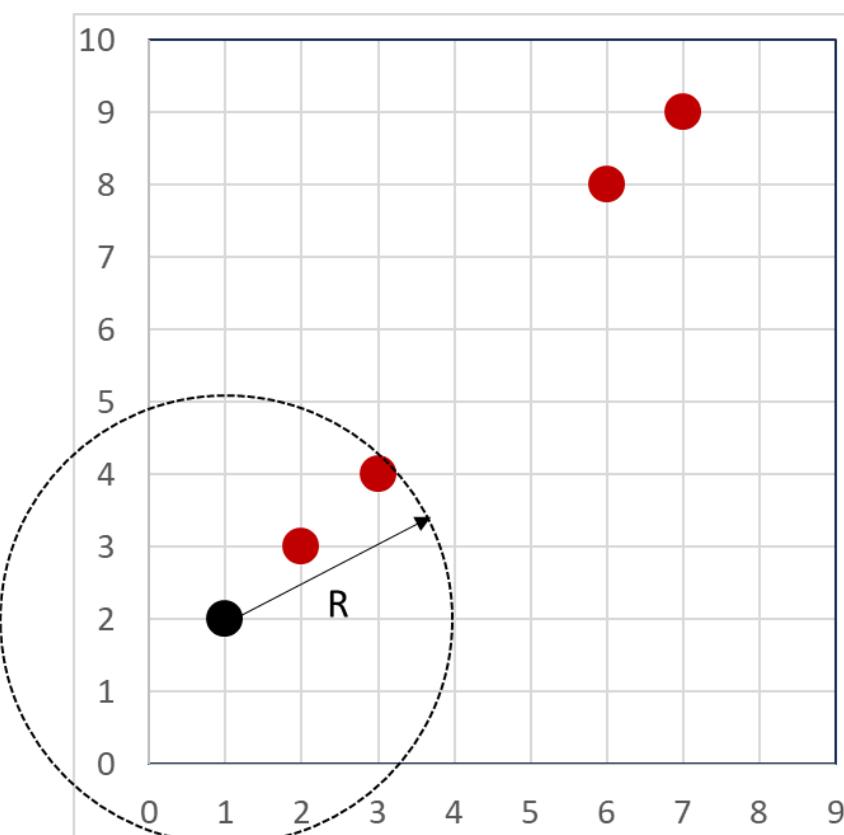


$$\mathbf{X} = \{(1, 2), (2, 3), (3, 4), (6, 8), (7, 9)\}$$

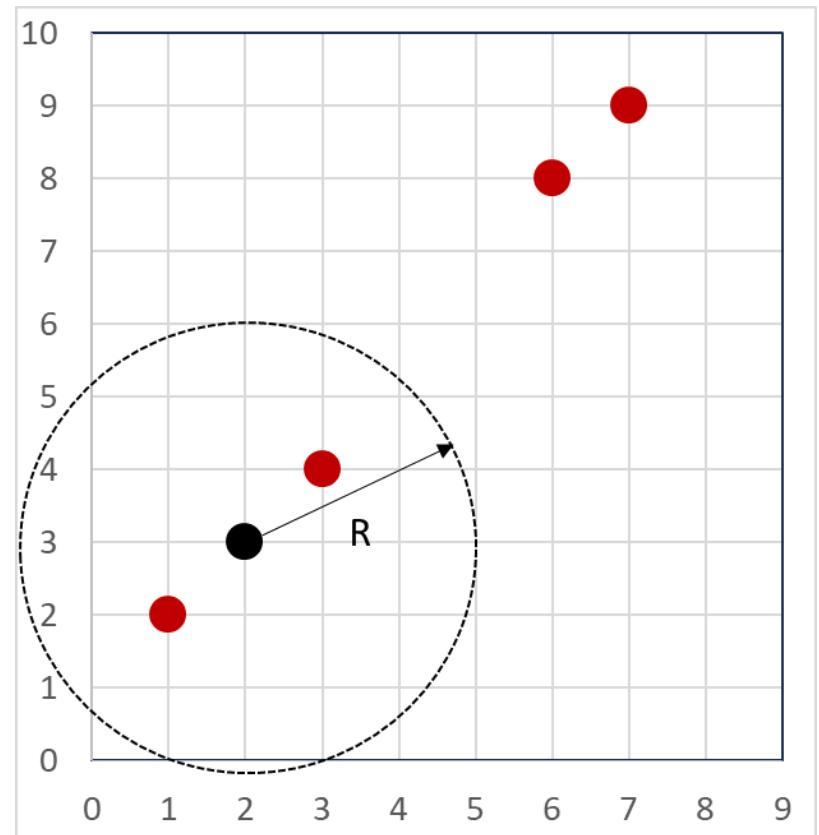
$$h = 3$$

$$\mu_w = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i}$$

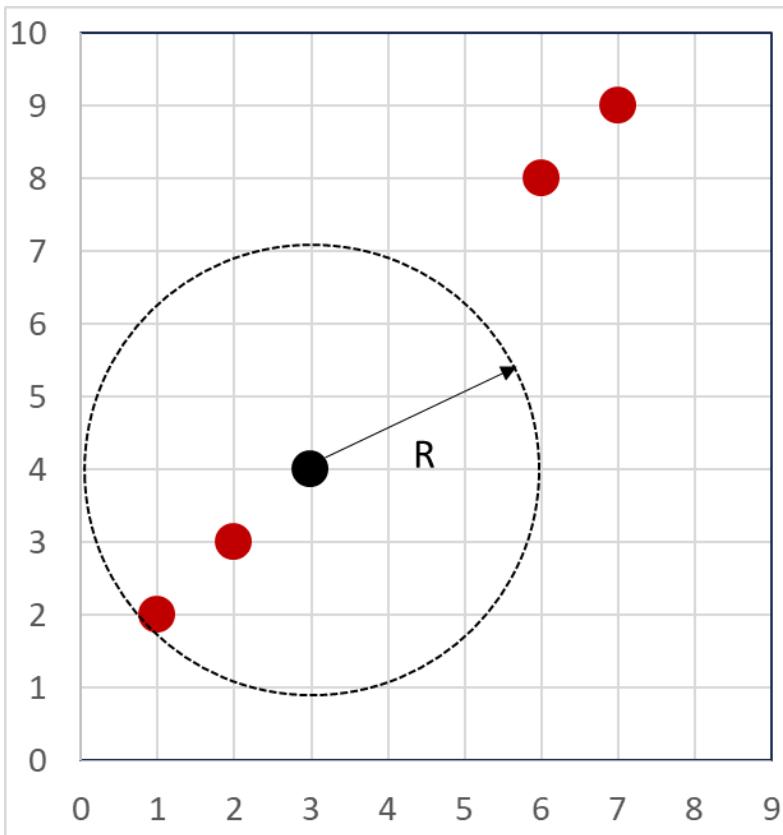
$$w_i(d) = \begin{cases} 1 & \text{se } d \leq R \\ 0 & \text{se } d > R \end{cases}$$



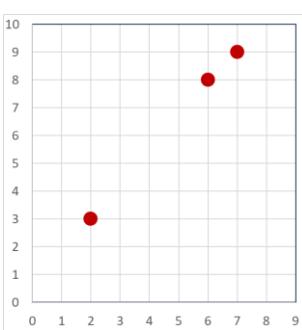
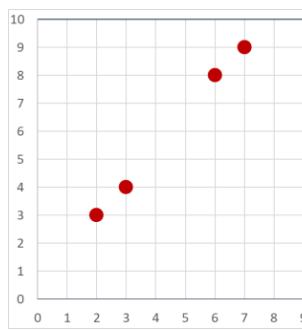
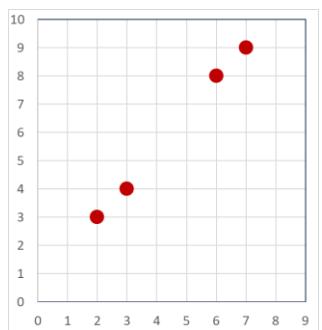
$$\left(\frac{1+2+3}{3}, \frac{2+3+4}{3} \right) = (2, 3)$$

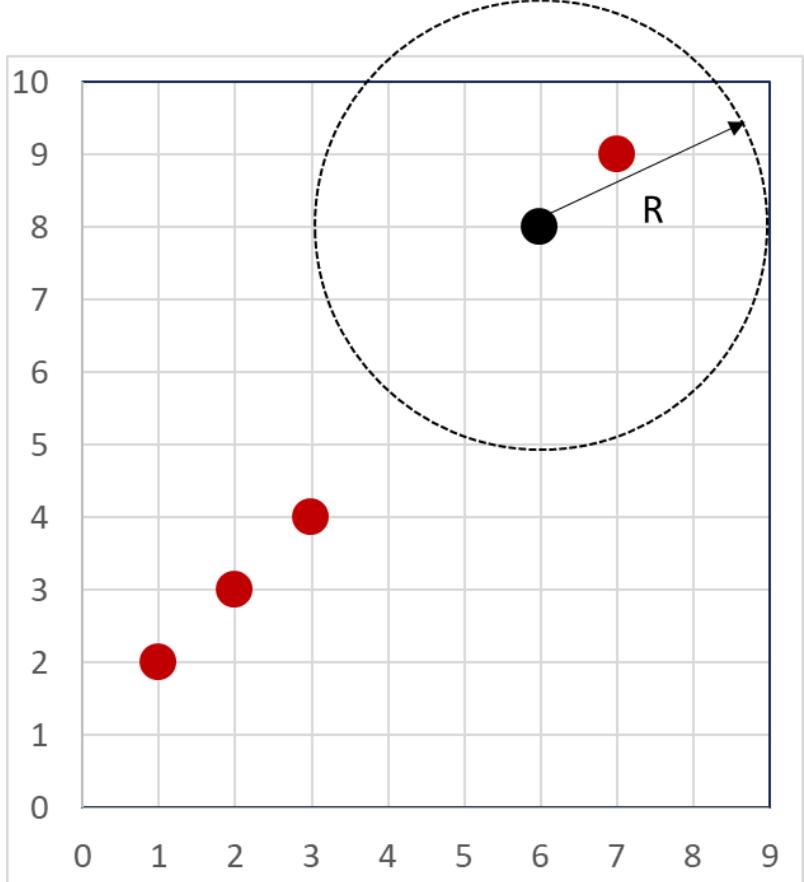


$$\left(\frac{1+2+3}{3}, \frac{2+3+4}{3} \right) = (2, 3)$$

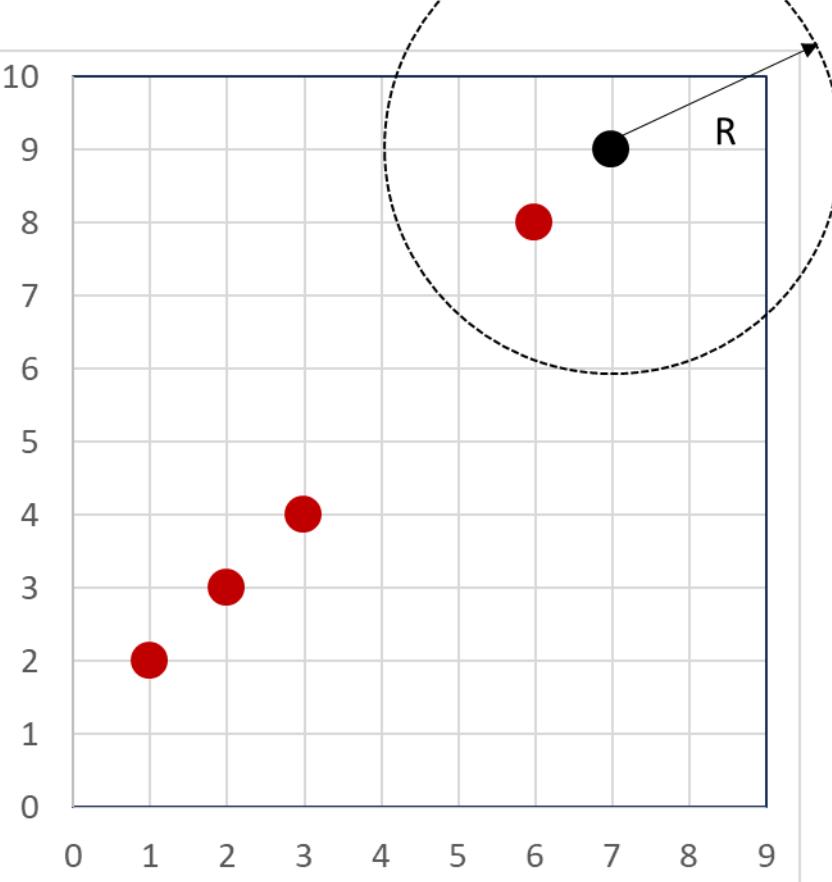
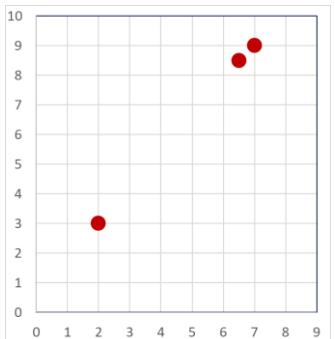


$$\left(\frac{1+2+3}{3}, \frac{2+3+4}{3} \right) = (2, 3)$$

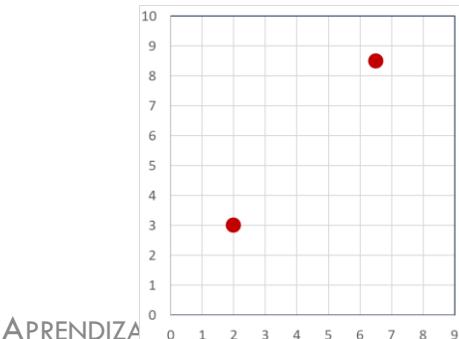




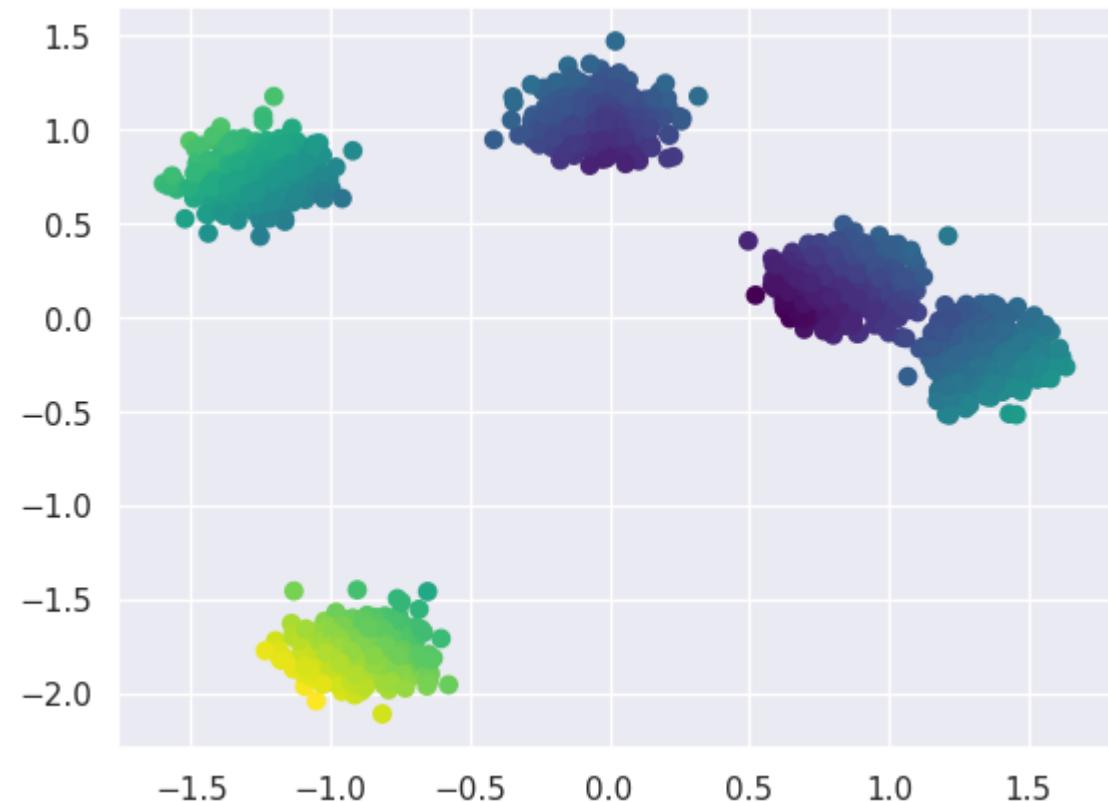
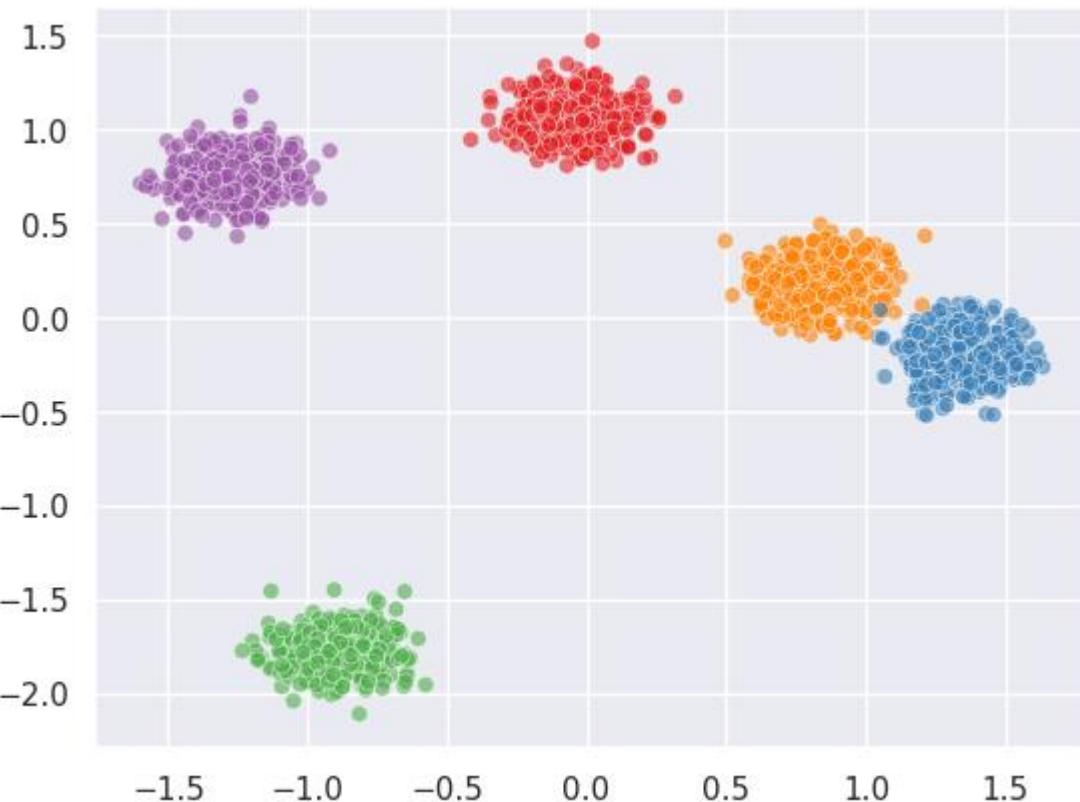
$$\left(\frac{6+7}{2}, \frac{8+9}{2} \right) = (6.5, 8.5)$$

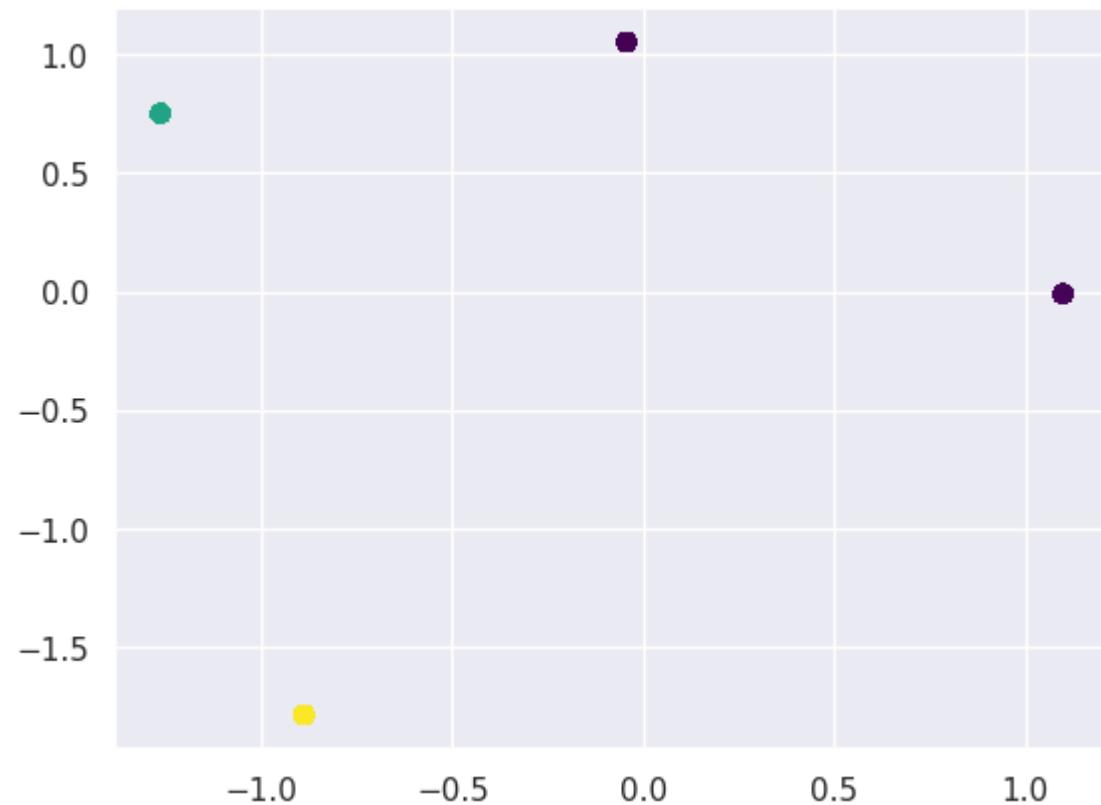
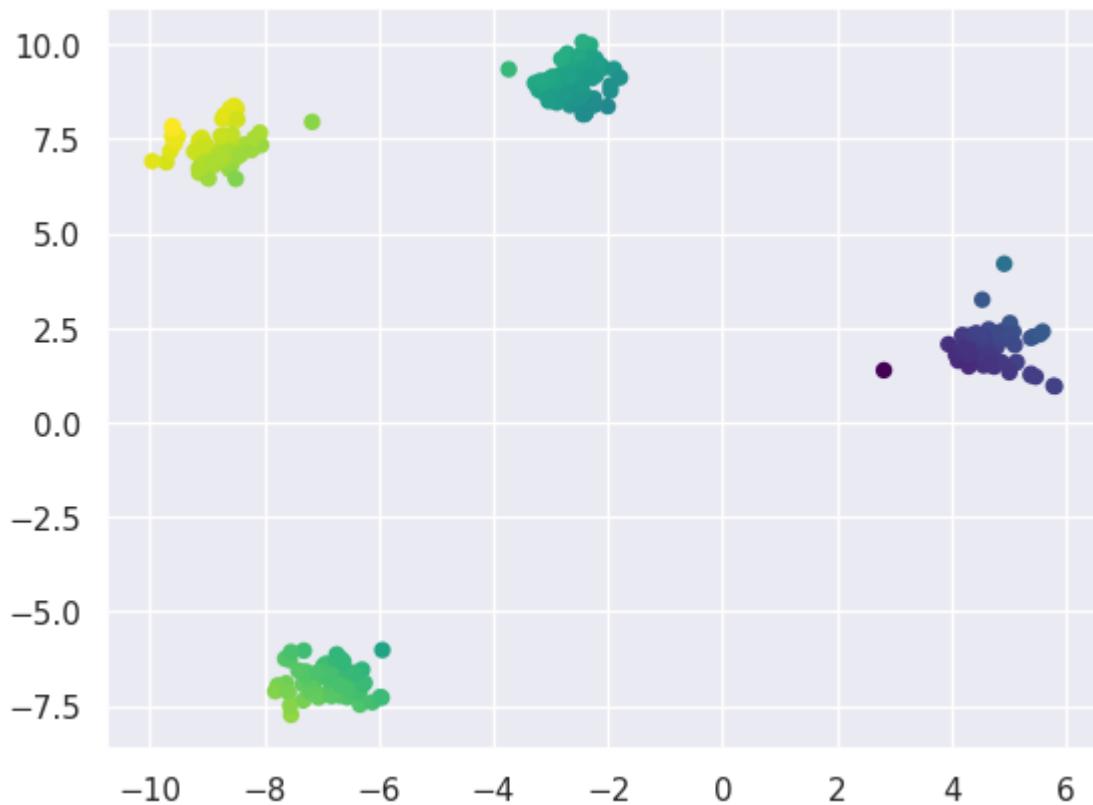


$$\left(\frac{6+7}{2}, \frac{8+9}{2} \right) = (6.5, 8.5)$$



- **Cluster 1:** (1, 2), (2, 3), (3, 4)
- **Cluster 2:** (6, 8), (7, 9)





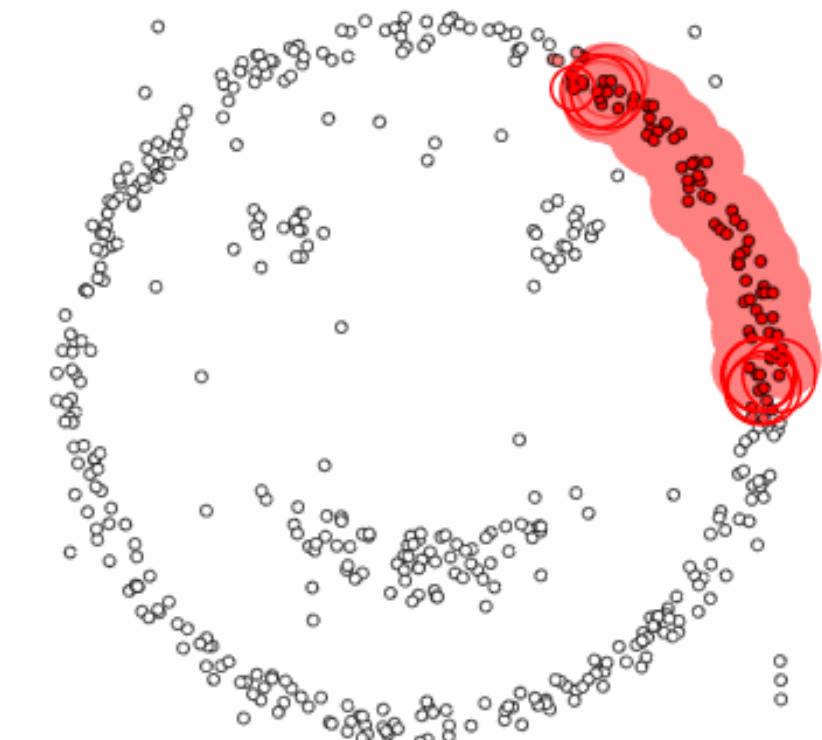
OUTROS ALGORITMOS DE CLUSTERIZAÇÃO

DBSCAN (Density-Based Spatial Clustering of Applications with Noise): trabalha com a premissa de que clusters são uma região de espaços densos separados por regiões de menor densidade. A maior vantagem deste algoritmo sobre os demais é que ele é robusto para outliers, o que significa que não os incluirá em nenhum cluster.

São usados dois parâmetros:

épsilon: raio do círculo a ser criado em torno de cada ponto de dados

minPoints: número mínimo de pontos de dados necessários dentro desse círculo para que esse ponto de dados seja classificado como um ponto central.



$$\varepsilon = 1$$

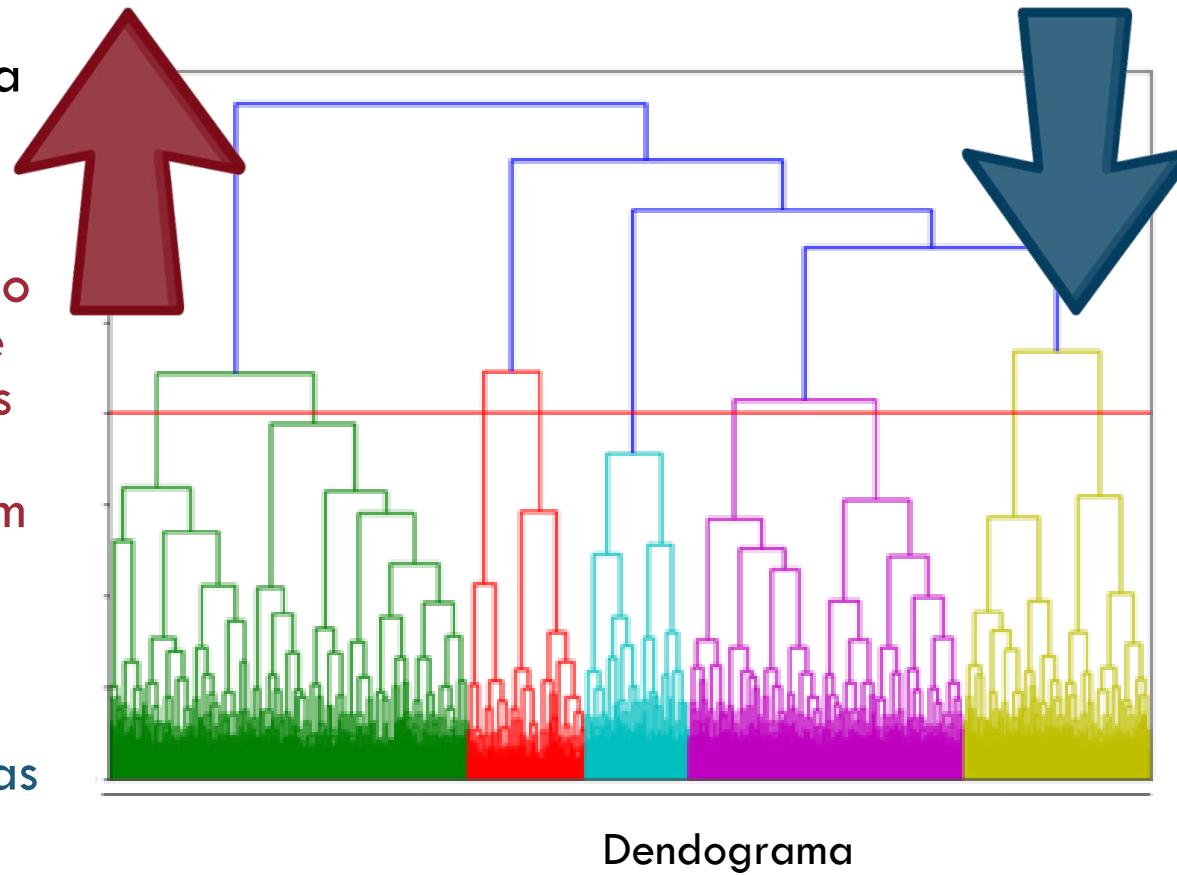
$$\text{minPoints} = 4$$

OUTROS ALGORITMOS DE CLUSTERIZAÇÃO

HC(Hierarchical-Clustering): constrói-se uma hierarquia de clusters.

Aglomerativo: abordagem “de baixo para cima” (bottom-up) onde cada observação é tratada como seu próprio cluster no início. Cria-se uma matriz de distância entre clusters e à medida que avançamos de baixo para cima, cada dois clusters que estão mais “próximos” são mesclados. Ao final, tem-se um único cluster.

Divisiva: Esta é uma abordagem “de cima para baixo” (top-down) onde todas as observações começam em um cluster e as divisões são realizadas recursivamente à medida que avançamos de cima para baixo.

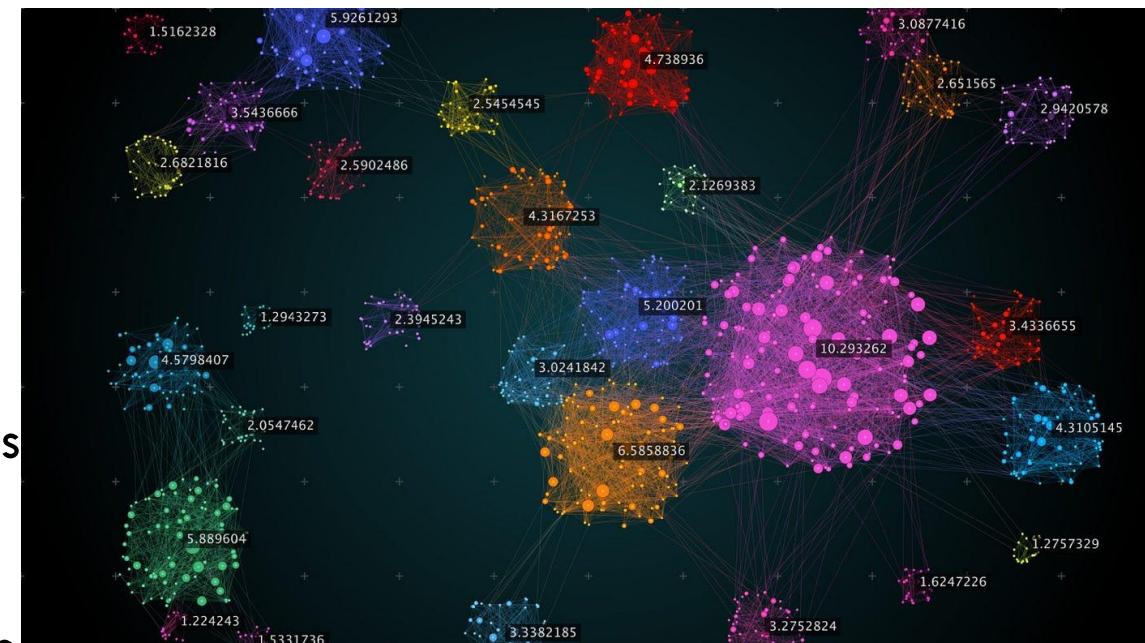


OUTROS ALGORITMOS DE CLUSTERIZAÇÃO

BIRCH (Balanced Iterative Hierarchical Based Clustering):

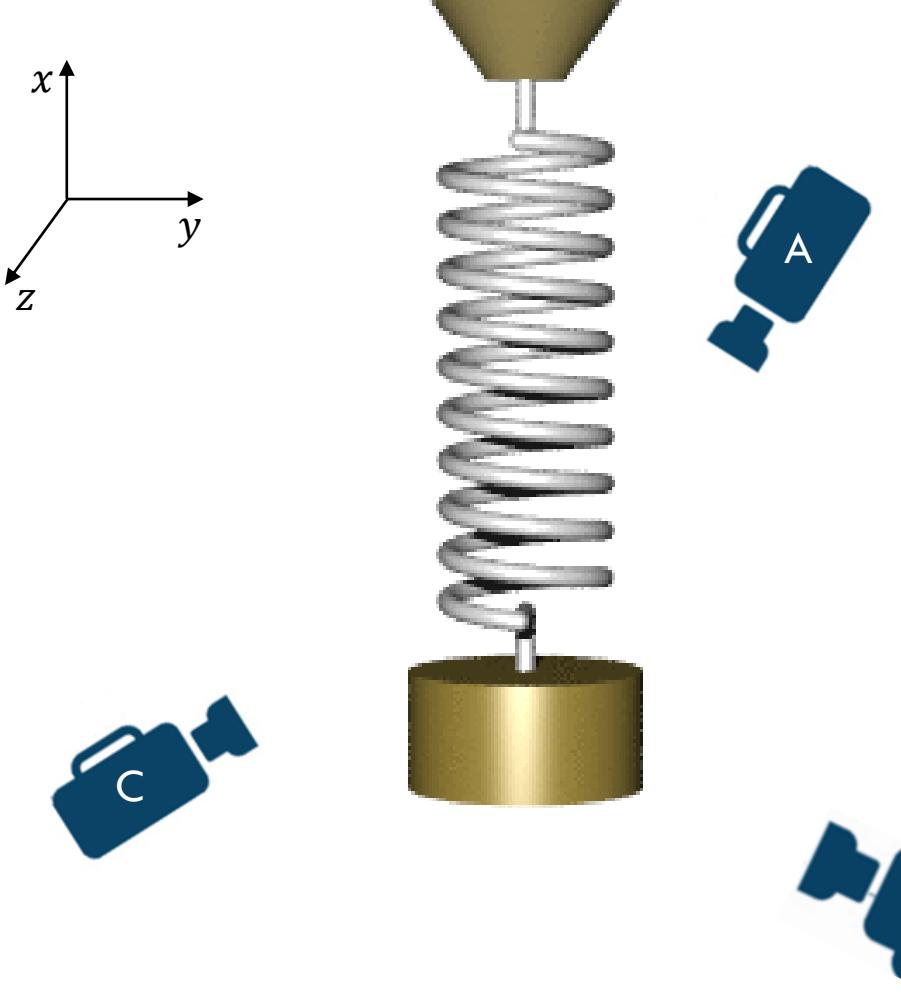
Ele é usado em conjuntos de dados muito grandes onde o K-Means não pode ser escalado na prática. O algoritmo BIRCH divide grandes dados em pequenos clusters e tenta reter o máximo de informações possível. Grupos menores são então agrupados para um resultado final, em vez de agrupar diretamente os grandes conjuntos de dados.

O BIRCH é frequentemente usado para complementar outros algoritmos de cluster, gerando um resumo das informações que os outros algoritmos de cluster podem utilizar.



Dendograma



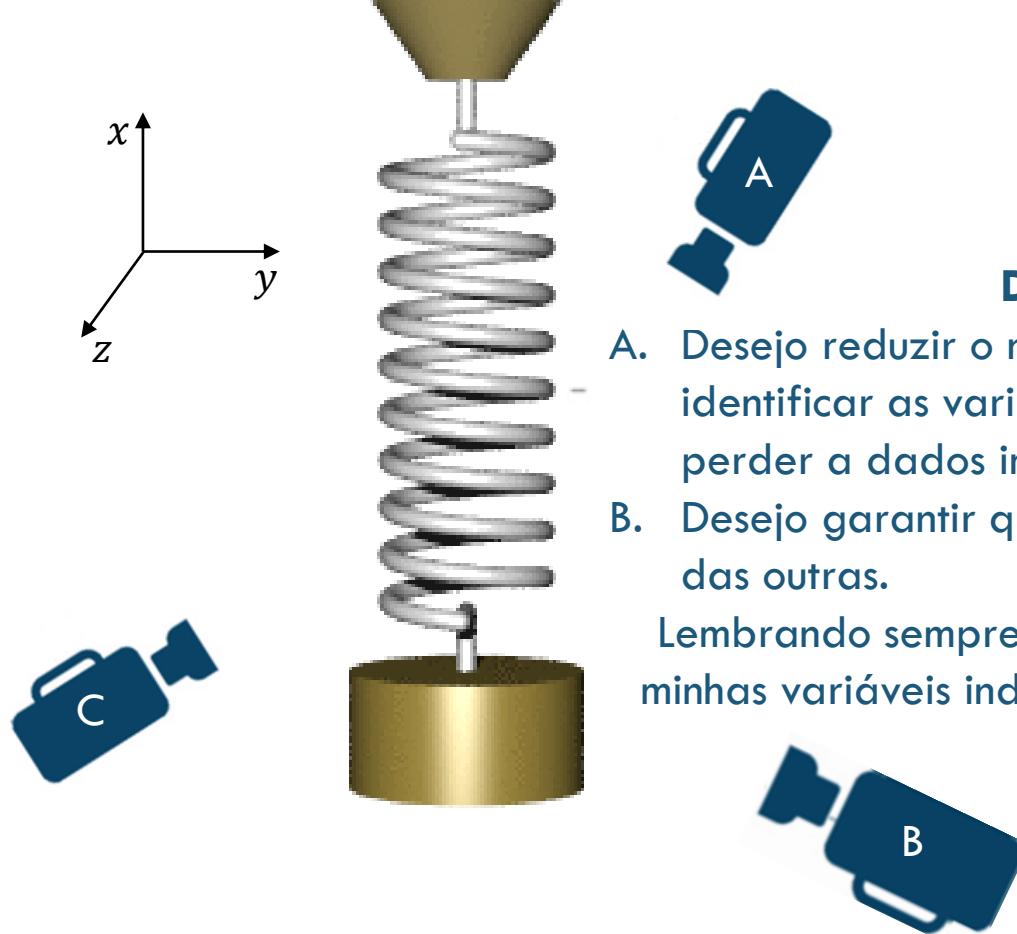


Decide-se medir a posição
da massa em um espaço
tridimensional

Cada câmera grava, digamos que a 200Hz, uma imagem indicando uma posição bidimensional da massa (uma projeção). Se gravamos com as câmeras por 2 minutos, teremos um imenso conjunto de dados 6D, em que cada câmera contribui com uma projeção bidimensional da posição da massa.

A grande questão: **como chegamos desse conjunto de dados a uma simples equação de x ?**

$$\vec{X} = \begin{bmatrix} x_A \\ y_A \\ x_B \\ y_B \\ x_C \\ y_C \end{bmatrix}$$



Devo usar PCA quando...

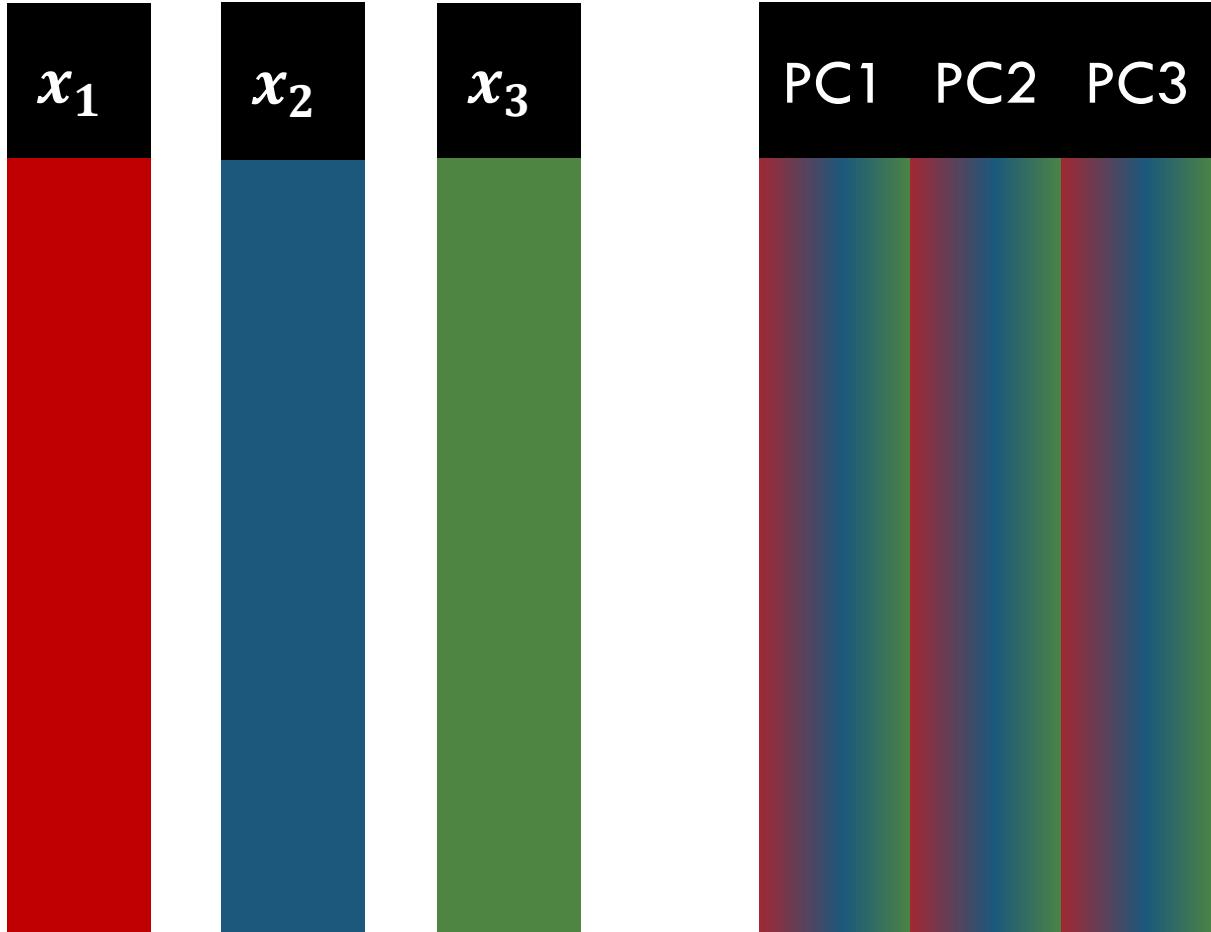
- A. Desejo reduzir o número de variáveis, mas não consigo identificar as variáveis que posso remover completamente sem perder a dados importantes do problema;
- B. Desejo garantir que as variáveis sejam independentes umas das outras.

Lembrando sempre que devo me sentir à vontade para tornar minhas variáveis independentes menos *interpretáveis fisicamente*.

Aqui, o objetivo explícito do PCA é determinar que **a dinâmica está no eixo x** .
Em outras palavras, o objetivo do PCA é determinar que o versor \hat{x} é a dimensão importante.

Principal Component Analysis

A ideia principal da **análise de componentes principais (PCA)** é encontrar padrões e correlações entre diferentes características do meu conjunto de dados de modo que este possa ser transformado em um conjunto de dados de dimensão significantemente menor sem perda de informação importante!



$$\begin{aligned}
 PC1 &= a_1x_1 + a_2x_2 + a_3x_3 \\
 PC2 &= b_1x_1 + b_2x_2 + b_3x_3 \\
 PC3 &= c_1x_1 + c_2x_2 + c_3x_3
 \end{aligned}$$

Número de variáveis
 =
 Número de componentes principais (PCs)

Dados observáveis			PC1	PC2	PC3
LDL	Pressão	Glicose			
169,00	185,00	143,00	255,10	160,99	215,66
154,00	194,00	247,00	222,35	256,75	258,99
185,00	179,00	319,00	221,00	325,91	305,38
183,00	192,00	129,00	276,45	149,79	218,88
205,00	210,00	390,00	244,50	395,80	359,05
192,00	210,00	394,00	231,80	398,22	353,78
151,00	173,00	369,00	173,40	368,65	308,38
150,00	150,00	393,00	149,25	389,49	312,21
199,00	196,00	270,00	258,60	282,76	295,15
145,00	141,00	353,00	148,00	351,25	287,96

PC1

$$0,9 \text{ } LDL + 0,75 \text{ } Pr - 0,25 \text{ } Gl$$

PC2

$$0,1 \text{ } LDL + 0,06 \text{ } Pr + 0,93 \text{ } Gl$$

PC3

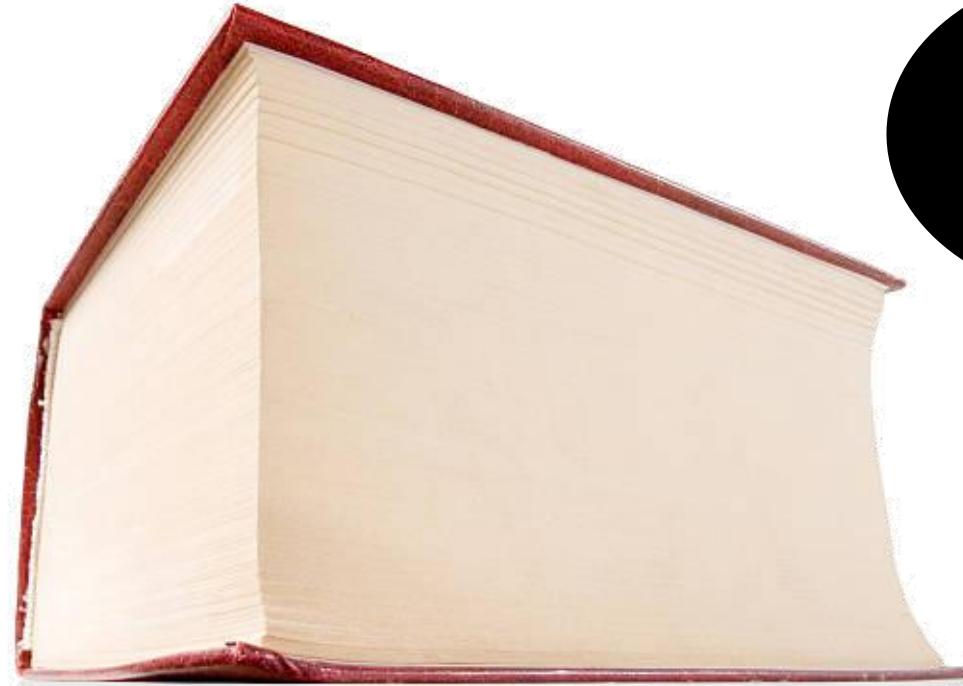
$$0,55 \text{ } LDL + 0,3 \text{ } Pr + 0,47 \text{ } Gl$$

$$PC1 = 0,9 \times 169 + 0,75 \times 185 - 0,25 \times 143 = 255,10$$

$$PC2 = 0,1 \times 169 + 0,06 \times 185 + 0,93 \times 143 = 160,99$$

$$PC3 = 0,55 \times 169 + 0,3 \times 185 + 0,47 \times 143 = 215,66$$

PENSE EM UM LIVRO GRANDE...



Só preciso das
ideias
principais...



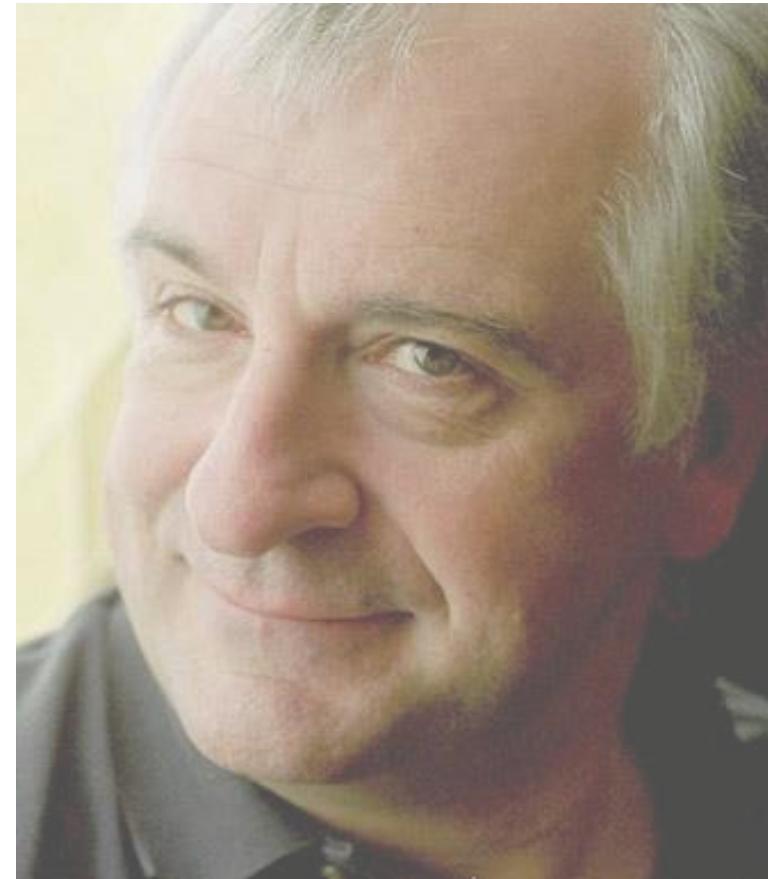
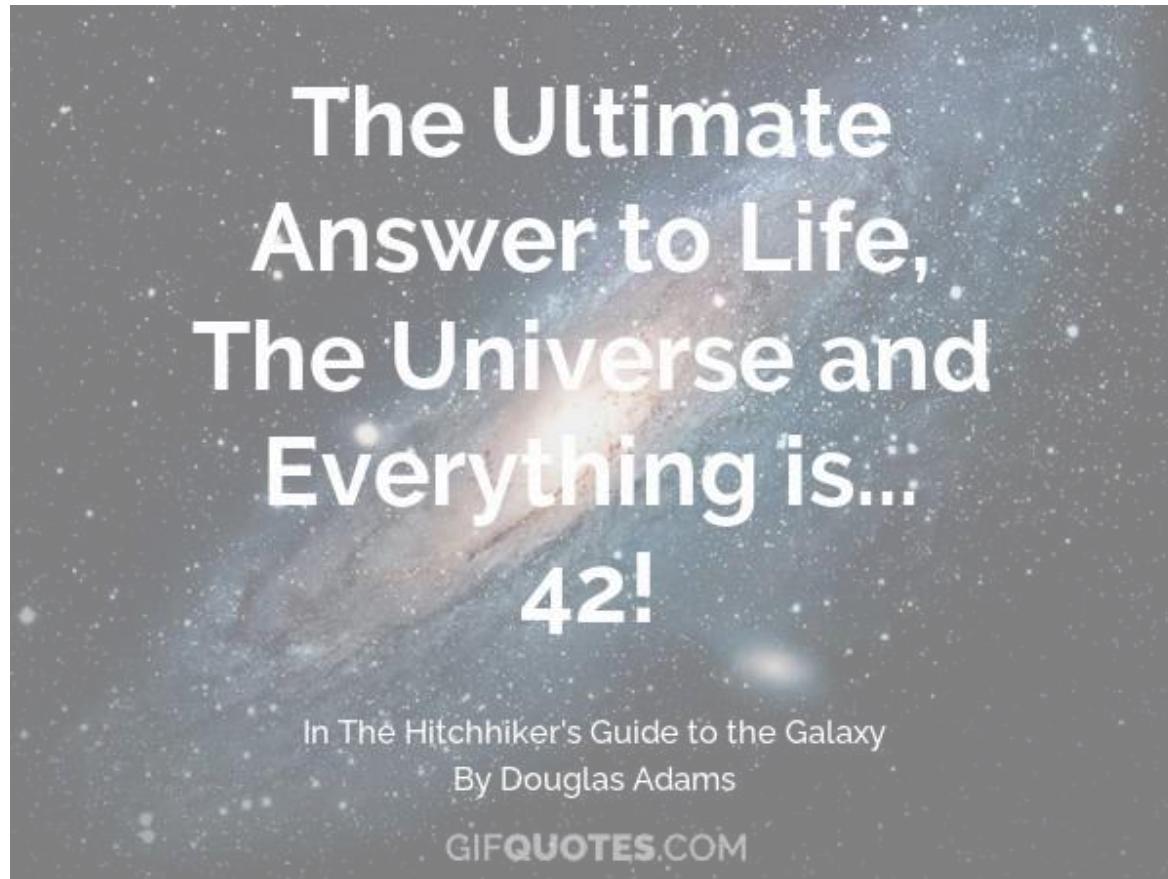
O PCA é extremamente útil ao trabalhar com conjuntos de dados que possuem muitas características. Embora ter mais dados seja sempre ótimo, às vezes eles têm tantas informações que teríamos um tempo de treinamento de modelo incrivelmente longo e a maldição da dimensionalidade começa a se tornar um problema.

PORTANTO...



PCA reduz a dimensionalidade a um conjunto de dados que **melhor expressa** as características do problema que estou analisando.

O QUE QUEREMOS DIZER COM “MELHOR EXPRESSA”?

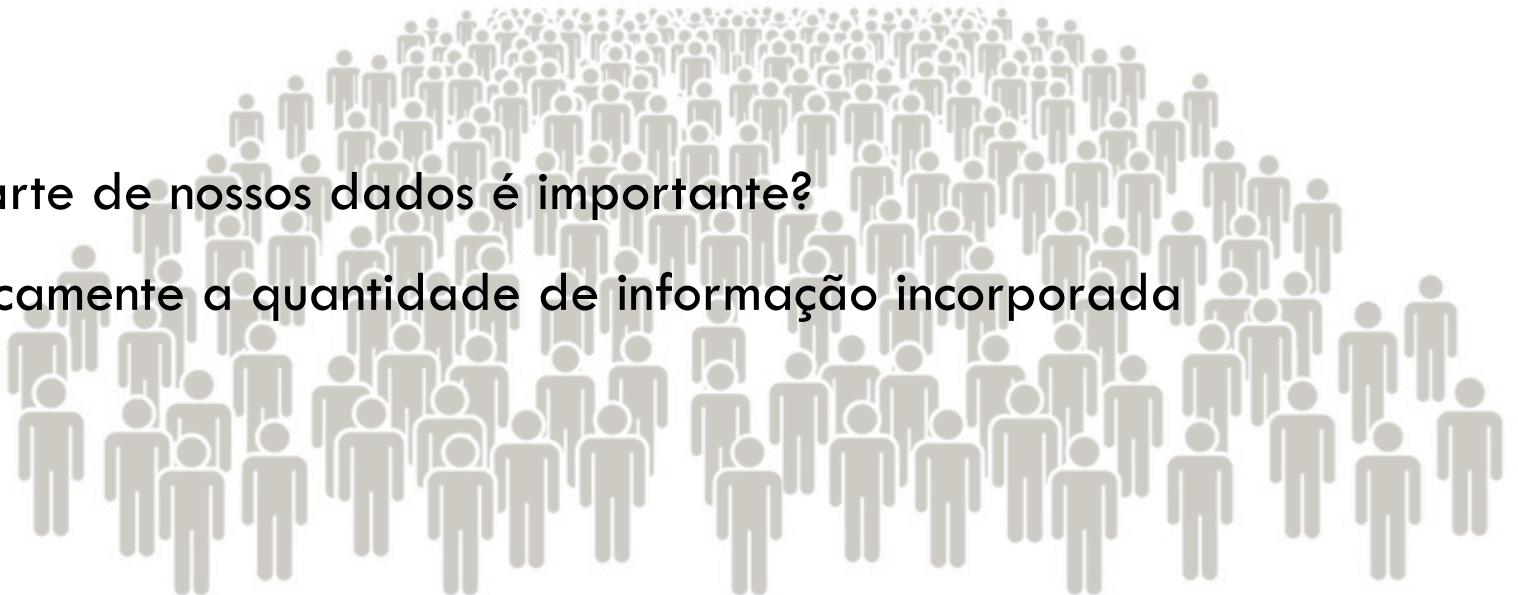


O PODER DA VARIÂNCIA

O PCA pode entender qual parte de nossos dados é importante?

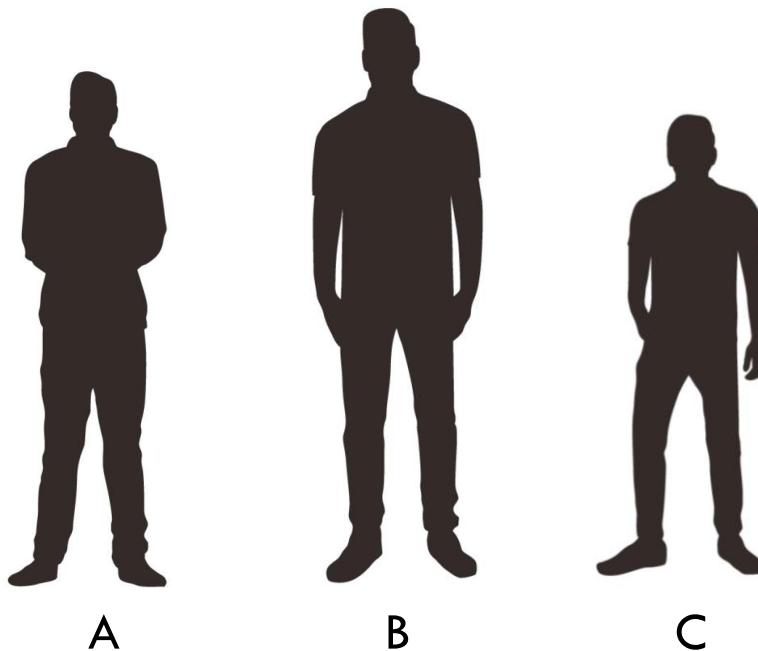
Podemos quantificar matematicamente a quantidade de informação incorporada nos dados?

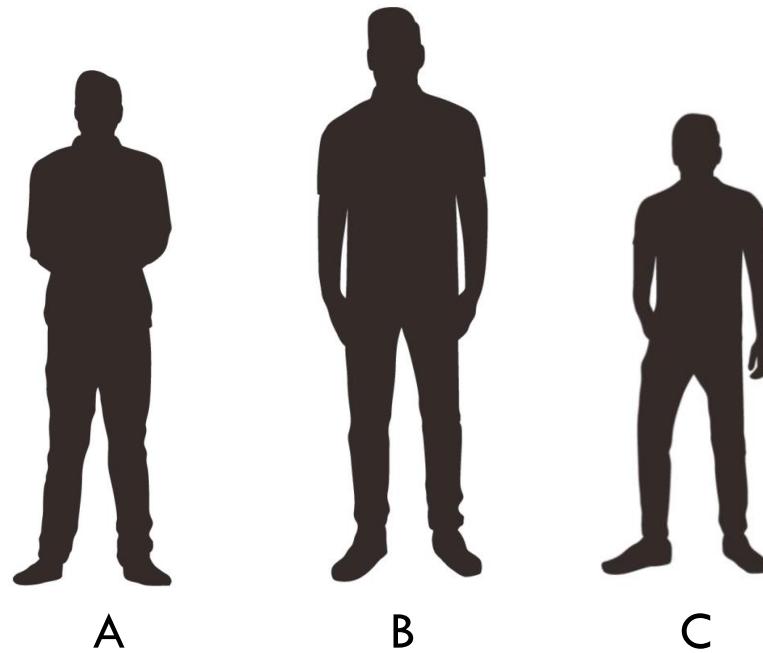
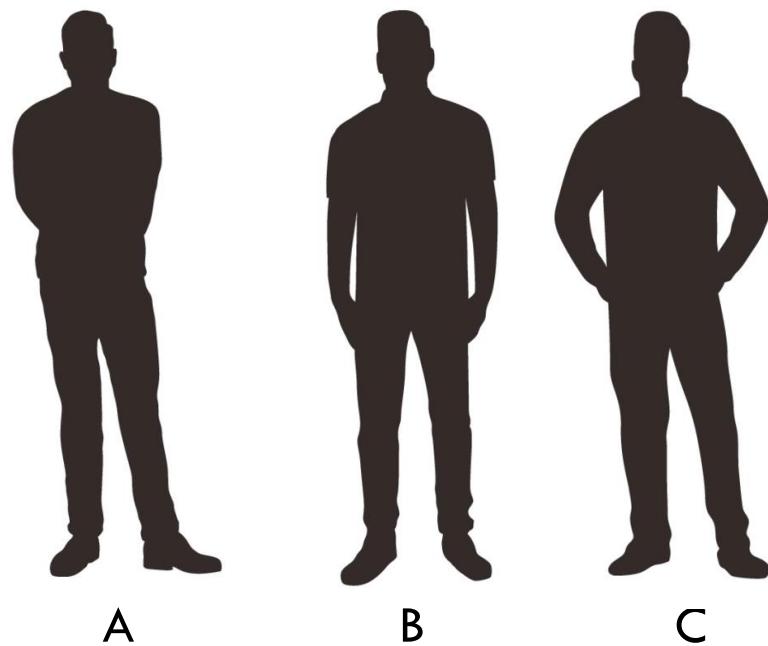
Bem, a variação pode.



Quanto maior a variação, mais informações. Vice-versa.

Pessoa	Altura [cm]
Alex	145
Eduardo	160
Felipe	185



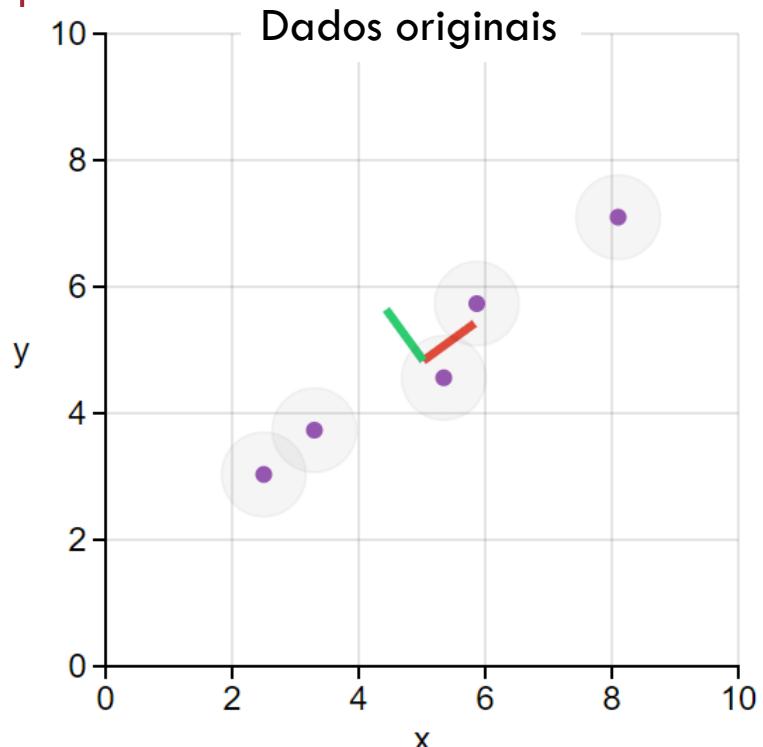


Pessoa	Altura [cm]
Daniel	183
Fernando	184
Felipe	185



Quanto maior a variação, mais fácil distinguir... Mais informação temos...

E O PCA...

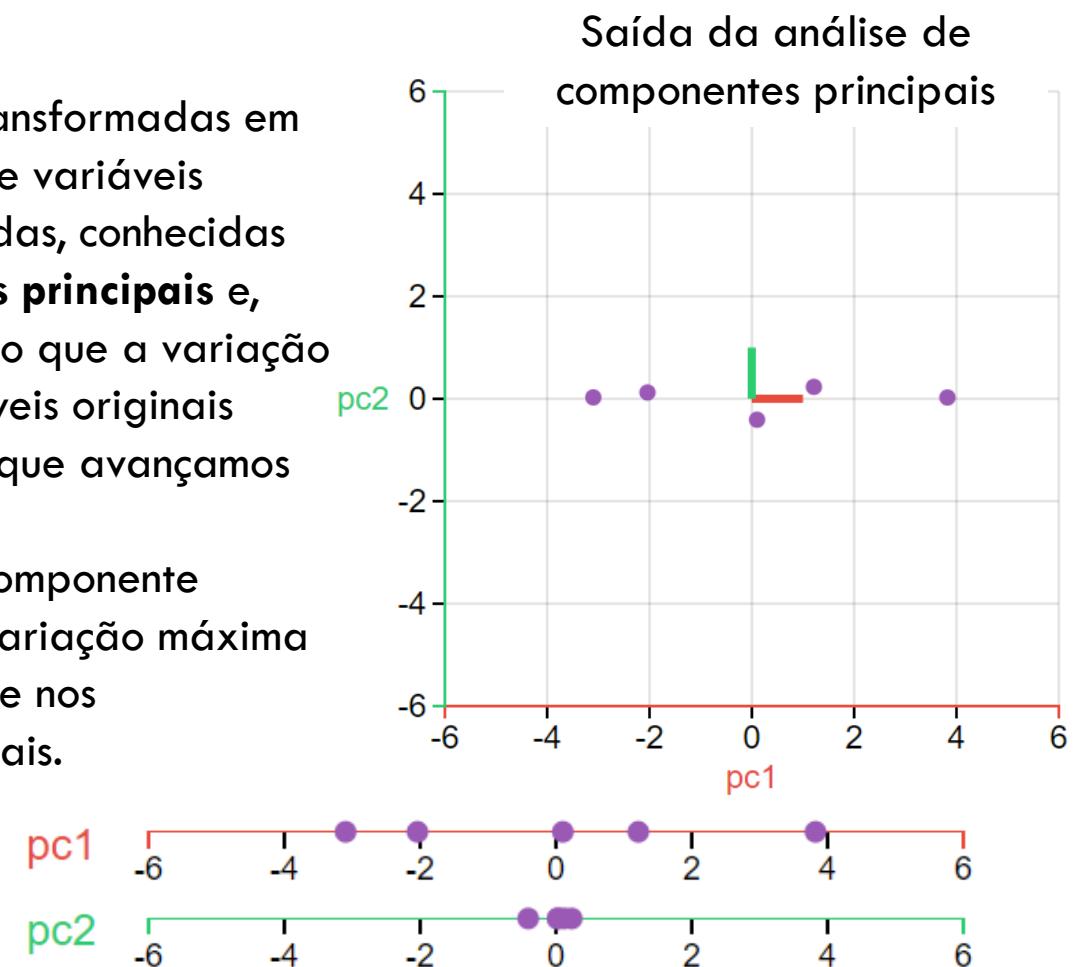


Fonte:

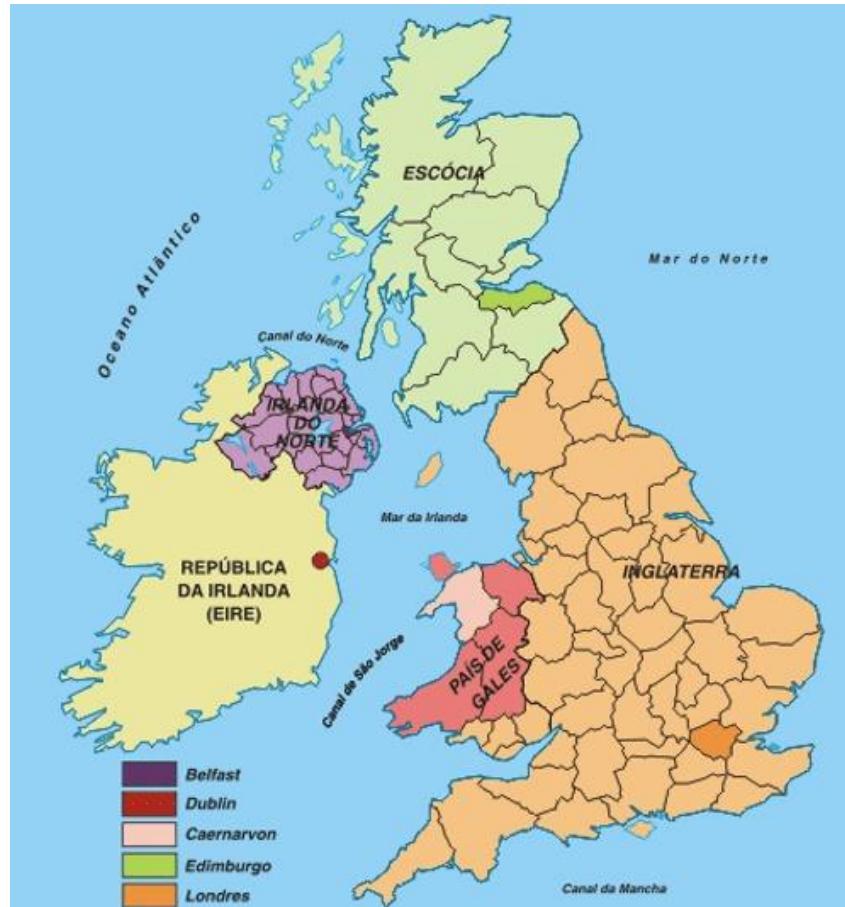
<https://setosa.io/ev/principal-component-analysis/>

As variáveis são transformadas em um novo conjunto de variáveis ortogonais ordenadas, conhecidas como **componentes principais** e, ordenadas de modo que a variação presente nas variáveis originais diminua à medida que avançamos na ordem.

Assim, o primeiro componente principal retém a variação máxima que estava presente nos componentes originais.



MUITAS VARIÁVEIS...

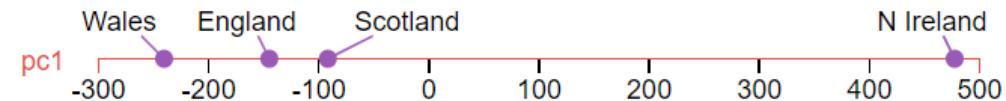


Fonte:

<https://setosa.io/ev/principal-component-analysis/>

A nova estrutura que visualizamos reflete um grande fato da geografia do mundo real: a Irlanda do Norte é o único dos quatro países que não estão na ilha da Grã-Bretanha.

67% da variação nos dados é contabilizada pela primeira componente principal:



97% da variação total é contabilizada pelas duas primeiras componentes principais:

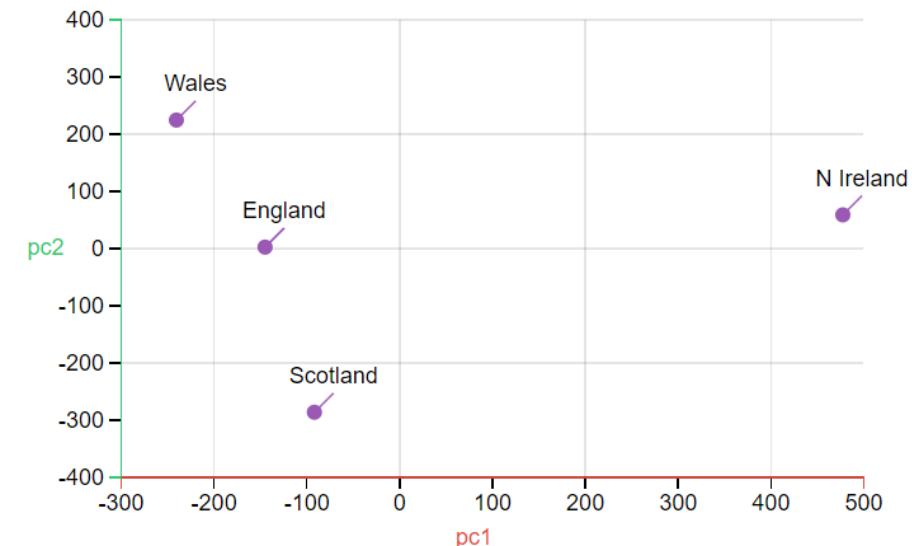


IMAGEM MULTICANAL OU MULTIESPECTRAL

A primeiro componente exibe (ou explica) 93,5% da variação de cena presente nos dados iniciais.

Dessa maneira, os dados iniciais de três canais foram reduzidos para dados de um canal, com uma perda em algum sentido de apenas 6,5% da variação da cena.

Lay, D.C.; Lay, S.R.; McDonald, J.J. *Linear Algebra and its applications*, 5th edition, Pearson Education, Inc.

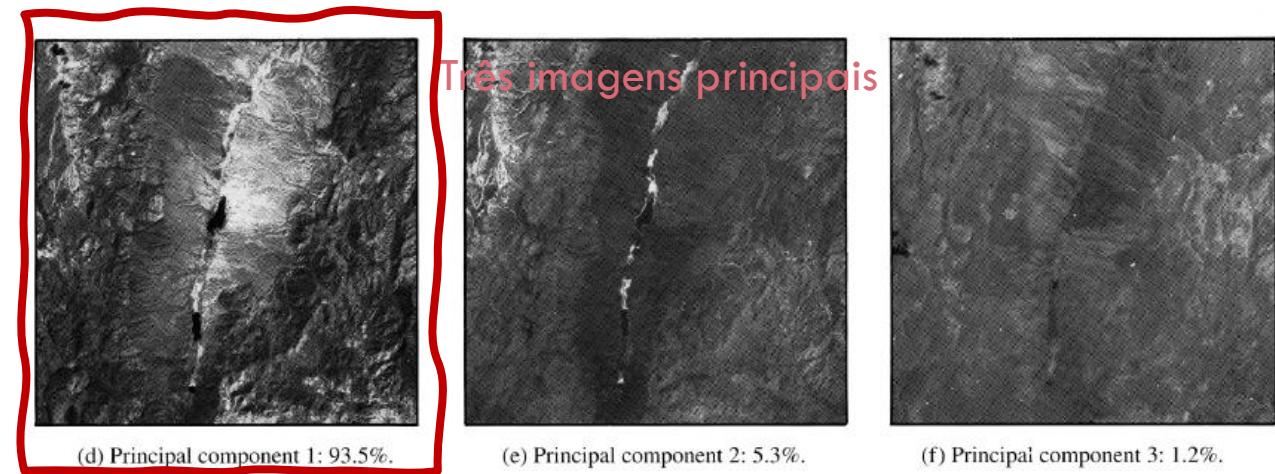
Railroad Valley



(a) Spectral band 1: Visible blue.

(b) Spectral band 4: Near infrared.

(c) Spectral band 7: Mid-infrared.



PERSPECTIVAS



NGC 1300



NGC 4594

Figuras extraídas de:

Principal Component Analysis, Leow Wee Kheng, National University of Singapore (NUS)



QUICK RECAP

Melhores momentos
do curso de
Estatística...

VARIÂNCIA , COVARIÂNCIA , CORRELAÇÃO

Basicamente, **variância** mede a variação de uma única variável aleatória (como a altura de uma pessoa em uma população), enquanto **covariância** é uma medida de quanto duas variáveis aleatórias variam juntas (como a altura e peso de uma pessoa em uma população). O **coeficiente de correlação** é uma quantidade adimensional.

$$s_1^2 = \sum_{i=1}^m \frac{(x_{1i} - \bar{x}_1)^2}{m - 1}$$

$$s_{12}^2 = \sum_{i=1}^m \frac{(x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)}{m - 1}$$

$$r_{12} = \frac{s_{12}^2}{s_1 s_2}$$

EXEMPLO

Os dados abaixo referem-se ao peso e altura de 4 pessoas. Monte a matriz X e calcule a matriz de covariância usando as fórmulas que aprendemos

Pessoa	Altura (cm)	Peso (kg)
1	149	48
2	155	52
3	163	57
4	177	68

$$s_{11}^2 = \sum_{i=1}^m \frac{(x_{1i} - \bar{x}_1)^2}{m-1} \quad s_{22}^2 = \sum_{i=1}^m \frac{(x_{2i} - \bar{x}_2)^2}{m-1}$$

$$s_{12}^2 = \frac{1}{m-1} \sum_{i=1}^m (x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)$$

Pessoa	Altura (cm)	Peso (kg)
1	149	48
2	155	52
3	163	57
4	177	68
Média	161	56,25

$$s_{11}^2 = \frac{\sum_{i=1}^m (x_{1i} - \bar{x}_1)^2}{m-1}$$

$$= \frac{(149 - 161)^2 + (155 - 161)^2 + (163 - 161)^2 + (177 - 161)^2}{3}$$

$$= \mathbf{146,67}$$

$$s_{22}^2 = \frac{\sum_{i=1}^m (x_{2i} - \bar{x}_2)^2}{m-1}$$

$$= \frac{(48 - 56,25)^2 + (52 - 56,25)^2 + (57 - 56,25)^2 + (68 - 56,25)^2}{3}$$

$$= \mathbf{74,92}$$

$$s_{12}^2 = \frac{1}{m-1} \sum_{i=1}^m (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) = \frac{1}{3} \sum_{i=1}^3 (x_{i1} - 161)(x_{i2} - 56,25)$$

$$= \frac{(48 - 56,25)(149 - 161) + (52 - 56,25)(155 - 161) + (57 - 56,25)(163 - 161) + (68 - 56,25)(177 - 161)}{3}$$

$$= \mathbf{104,67}$$

MATRIZ DE DADOS $X \in \mathbb{R}^{m \times n}$

$$\mathbf{X}_{m,n} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{m,n} \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \hat{\mathbf{x}}_1 & \hat{\mathbf{x}}_2 & \cdots & \hat{\mathbf{x}}_n \\ | & | & & | \end{pmatrix} = \begin{pmatrix} - & \mathbf{x}_1 & - \\ - & \mathbf{x}_2 & - \\ \vdots & & \vdots \\ - & \mathbf{x}_m & - \end{pmatrix}$$

cada coluna representa
 várias medições da mesma
 variável (tenho n variáveis)

cada linha é um conjunto de
 observações das variáveis
 (tenho m dados)

MATRIZ DE DADOS $X \in \mathbb{R}^{m \times n}$

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Vetores de características $\hat{\mathbf{x}}_j, \hat{\mathbf{x}}_k \in \mathbb{R}^n$ tal que,

$$\hat{\mathbf{x}}_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{mj} \end{bmatrix}, \quad \hat{\mathbf{x}}_k = \begin{bmatrix} x_{1k} \\ x_{2k} \\ \vdots \\ x_{mk} \end{bmatrix}$$

$$s_{jk} = \frac{1}{m-1} \sum_{i=1}^m (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$

As entradas diagonais da matriz S são as **variâncias** e as entradas fora da diagonal são as **covariâncias**.

NO EXEMPLO ANTERIOR...

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} = \begin{bmatrix} 146,67 & 104,67 \\ 104,67 & 74,92 \end{bmatrix}$$

Pode-se escrever a matriz de correlação,

$$S = \begin{bmatrix} 1 & \frac{104,67}{\sqrt{146,67}\sqrt{74,92}} \\ \frac{104,67}{\sqrt{146,67}\sqrt{74,92}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0,998512 \\ 0,998512 & 1 \end{bmatrix}$$

MATRIZ DE COVARIÂNCIA

$$\mathbf{S} = \frac{1}{m-1} \sum_{i=1}^m (\mathbf{x}_i - \bar{\mathbf{x}})^T (\mathbf{x}_i - \bar{\mathbf{x}})$$

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$

$$\mathbf{x}_i = (x_{i1} \quad x_{i2} \quad \cdots \quad x_{in}), \quad i = 1, \dots, m$$

$\bar{\mathbf{x}}^T = [\bar{x}_1 \quad \bar{x}_2 \quad \cdots \quad \bar{x}_n]$, de modo que $\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$ é a média da variável j .

MATRIZ DE COVARIÂNCIA

$$\mathbf{S} = \frac{1}{m - 1} [\mathbf{X}_c^T \mathbf{X}_c]$$

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$

$$\mathbf{X}_c = \mathbf{X} - \mathbf{1}_m \bar{\mathbf{x}}^T, \quad \bar{\mathbf{x}}^T = [\bar{x}_1 \quad \bar{x}_2 \quad \cdots \quad \bar{x}_n], \quad \mathbf{1}_m^T = [1 \quad 1 \quad \cdots \quad 1]_{1 \times m}$$

$\mathbf{S} \in \mathbb{R}^{n \times n}$ é uma matriz quadrada e simétrica.

VAMOS FAZER UMAS CONTINHAS PARA ENTENDER...

Pessoa	Altura (cm)	Peso (kg)
1	149	48
2	155	52
\bar{x}	152	50

$$s_{11}^2 = \sum_{i=1}^m \frac{(x_{1i} - \bar{x}_1)^2}{m-1} = \frac{(149 - 152)^2 + (155 - 152)^2}{1} = 18$$

$$s_{22}^2 = \sum_{i=1}^m \frac{(x_{2i} - \bar{x}_2)^2}{m-1} = \frac{(48 - 50)^2 + (52 - 50)^2}{1} = 8$$

$$s_{12}^2 = \frac{1}{m-1} \sum_{i=1}^m (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) = \frac{1}{1} \sum_{i=1}^3 (x_{i1} - 152)(x_{i2} - 50) = (149 - 152)(48 - 50) + (155 - 152)(52 - 50) = 12$$

```
#Reescrevendo o vetor X
X = np.array([[149, 48], [155, 52]])
#Covariância:
print('Numpy:')
print(np.cov(X.T), '\n')
```

```
Numpy:
[[18. 12.]
 [12. 8.]]
```

VAMOS FAZER UMAS CONTINHAS PARA ENTENDER...

Pessoa	Altura (cm)	Peso (kg)
1	149	48
2	155	52
\bar{x}	152	50

Numpy:

```
[ [18. 12.]
 [12. 8.]]
```

$$\mathbf{S} = \frac{1}{m-1} \sum_{i=1}^m (\mathbf{x}_i - \bar{\mathbf{x}})^T (\mathbf{x}_i - \bar{\mathbf{x}})$$

$$\bar{\mathbf{x}} = [152 \quad 50]$$

$$\mathbf{x}_1 = [149 \quad 48]$$

$$\mathbf{x}_2 = [155 \quad 52]$$

$$\mathbf{S} = \sum_{i=1}^m \frac{(\mathbf{x}_i - \bar{\mathbf{x}})^T (\mathbf{x}_i - \bar{\mathbf{x}})}{m-1}$$

$$= \{[149 \quad 48] - [152 \quad 50]\}^T \{[149 \quad 48] - [152 \quad 50]\} + \{[155 \quad 52] - [152 \quad 50]\}^T \{[155 \quad 52] - [152 \quad 50]\}$$

$$= \begin{bmatrix} -3 \\ -2 \end{bmatrix} [-3 \quad -2] + \begin{bmatrix} 3 \\ 2 \end{bmatrix} [3 \quad 2] = \begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix} + \begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix} = \begin{bmatrix} 18 & 12 \\ 12 & 8 \end{bmatrix}$$

VAMOS FAZER UMAS CONTINHAS PARA ENTENDER...

Pessoa	Altura (cm)	Peso (kg)
1	149	48
2	155	52
\bar{x}	152	50

Numpy:
 $\begin{bmatrix} [18. 12.] \\ [12. 8.] \end{bmatrix}$

$$\mathbf{S} = \frac{1}{m-1} [\mathbf{X}_c^T \mathbf{X}_c]$$

$$\mathbf{X}_c = \mathbf{X} - \mathbf{1}_m \bar{\mathbf{x}}^T,$$

$$\bar{\mathbf{x}}^T = [152 \quad 50]$$

$$\mathbf{1}_m^T = [1 \quad 1]$$

$$\mathbf{X} = \begin{bmatrix} 149 & 48 \\ 155 & 52 \end{bmatrix}$$

$$\mathbf{S} = \frac{1}{m-1} (\mathbf{X}_c^T \mathbf{X}_c)$$

$$\mathbf{X}_c = \begin{bmatrix} 149 & 48 \\ 155 & 52 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} [152 \quad 50] = \begin{bmatrix} 149 & 48 \\ 155 & 52 \end{bmatrix} - \begin{bmatrix} 152 & 50 \\ 152 & 50 \end{bmatrix} = \begin{bmatrix} -3 & -2 \\ 3 & 2 \end{bmatrix}$$

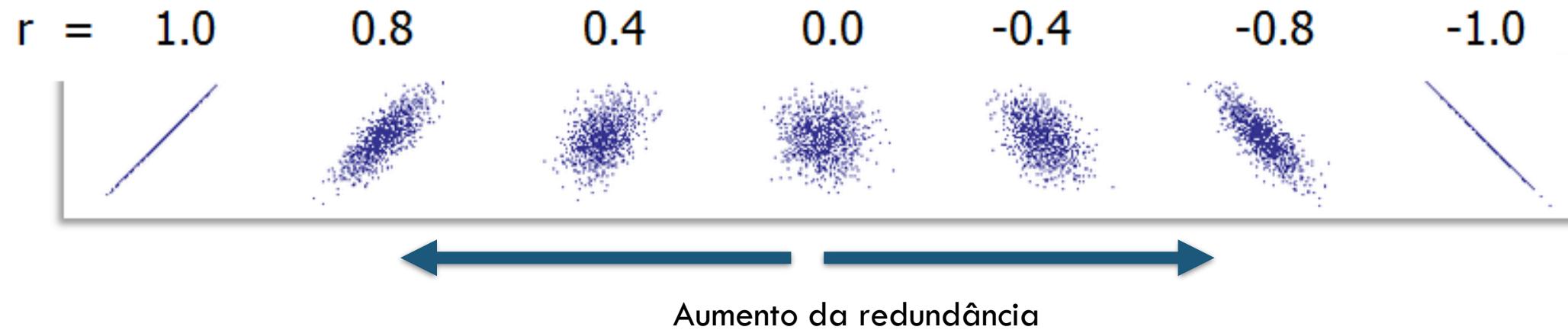
$$\mathbf{S} = \begin{bmatrix} -3 & 3 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} -3 & -2 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 18 & 12 \\ 12 & 8 \end{bmatrix}$$

FAÇA VOCÊ

Os dados abaixo referem-se ao peso e altura de 4 pessoas. Monte a matriz X e calcule a matriz de covariância usando as fórmulas que aprendemos

Pessoa	Altura (cm)	Peso (kg)	Idade (anos)
1	149	48	27
2	155	52	39
3	163	57	45
4	177	68	33

COVARIÂNCIA E REDUNDÂNCIA DE CARACTERÍSTICAS



Termos diagonais: variância

Valores grandes = sinal

Fora da diagonal: covariância

Valores grandes = alta redundância



OUTRA RECAP

Base de um vetor

LEMBREM-SE...

Matematicamente, para que um conjunto de vetores \mathbf{b}_i seja linearmente independente, em um espaço n -dimensional, a expressão

$$c_1\mathbf{b}_1 + c_2\mathbf{b}_2 + \dots + c_n\mathbf{b}_n = 0$$

deve ser possível apenas se todos os fatores lineares c_i forem 0.

Se um conjunto de vetores é LI, nenhum desses vetores pode ser expresso como uma combinação linear dos outros.

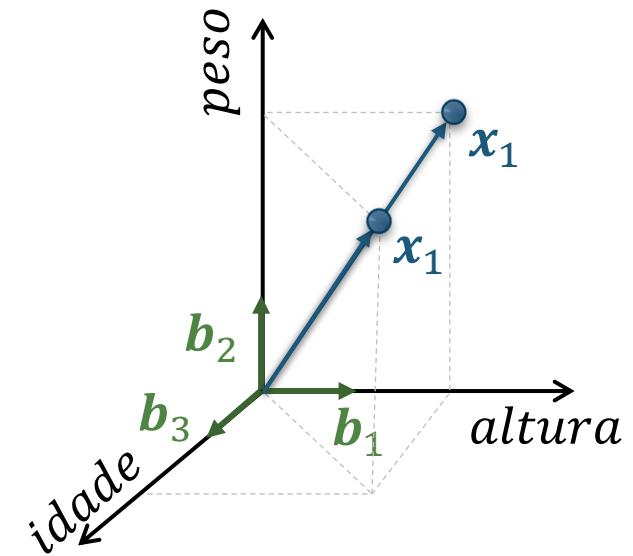
BASE

Em geral, cada amostra de dados é um vetor no espaço de dimensão n , onde n é o número de características da amostra. Equivalentemente, toda amostra é um vetor que se encontra em um espaço vetorial de dimensão n , representado por uma base ortonormal.

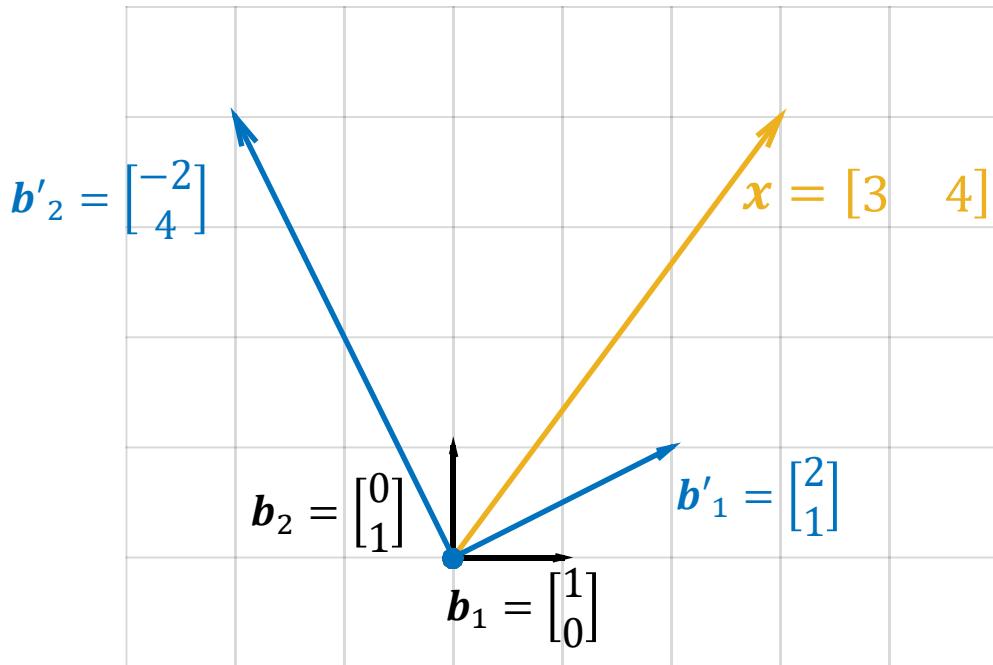
Todos os vetores de medição nesse espaço são uma combinação linear desse conjunto de vetores básicos de comprimento unitário. Uma escolha simples e direta de uma base B é a matriz de identidade I .

$$B = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = I$$

Cada coluna \mathbf{b}_i é uma base com n componentes.



MUDANÇA DE BASE



$$x = [3 \quad 4] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = z_1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + z_2 \begin{bmatrix} -2 \\ 4 \end{bmatrix}$$

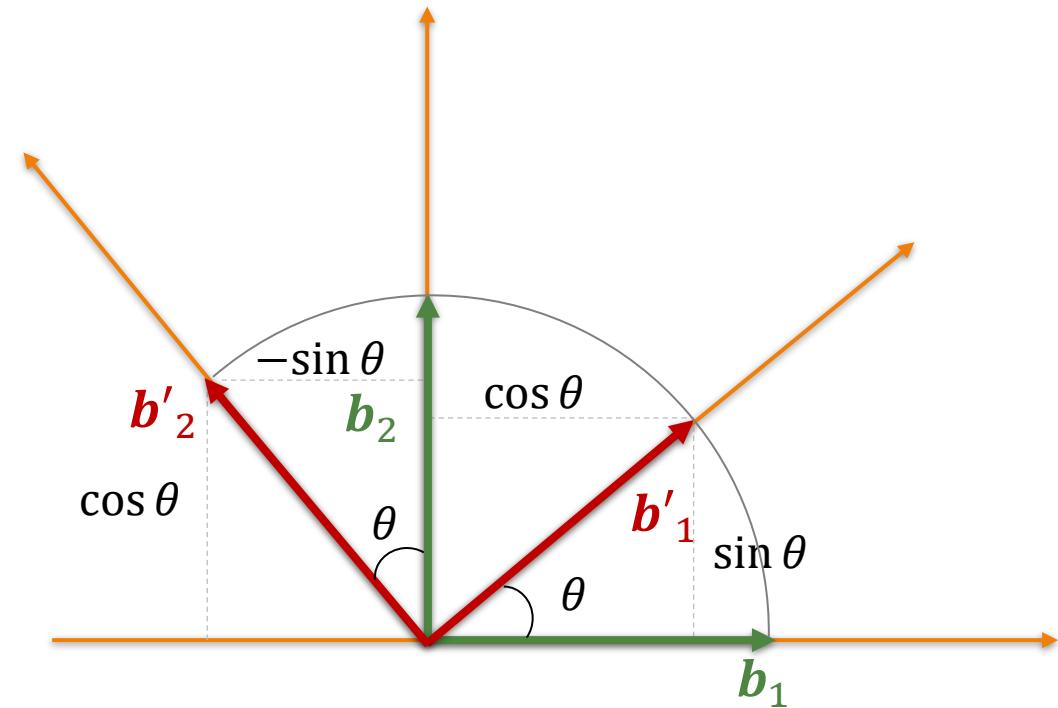
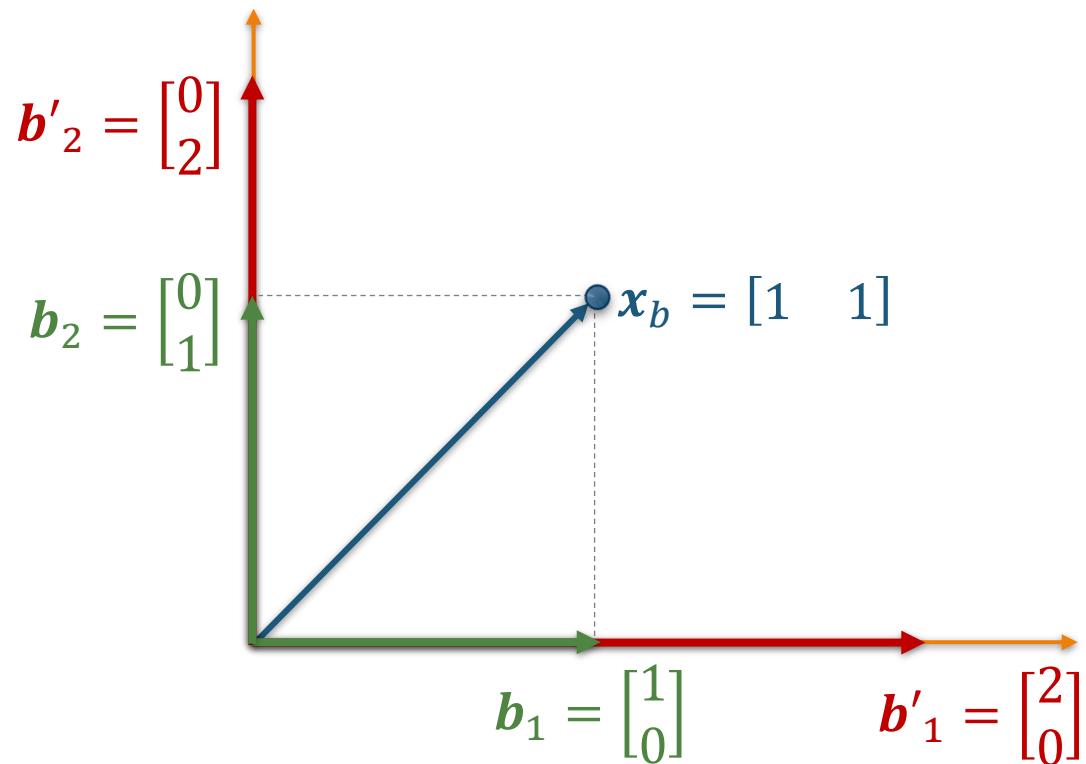
$$= [z_1 \quad z_2] \begin{bmatrix} 2 & 1 \\ -2 & 4 \end{bmatrix}$$

```
np.linalg.inv(np.array([[2,1],[-2,4]]))  
array([[ 0.4, -0.1],  
       [ 0.2,  0.2]])
```

$$[z_1 \quad z_2] = [3 \quad 4] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & -1 \\ 2 & 2 \end{bmatrix} \frac{1}{10} W$$

$$z = [z_1 \quad z_2] = [2 \quad 1/2]$$

EXEMPLO DO NOTEBOOK



DADOS

$X, Z \in \mathbb{R}^{n \times m}, W \in \mathbb{R}^{n \times n} \rightarrow W$ é a matriz que transforma X em $Z = XW$

Conjunto de m observações da mesma variável.

$$X_{m,n} = \left(\begin{array}{c|cccc} x_{11} & x_{12} & \cdots & x_{1n} \\ \hline x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \ddots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{m,n} \end{array} \right) = \left(\begin{array}{cccc} | & | & & | \\ \hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_n \\ | & | & & | \end{array} \right) = \left(\begin{array}{ccc} - & \mathbf{x}_1 & - \\ - & \mathbf{x}_2 & - \\ \vdots & & \\ - & \mathbf{x}_m & - \end{array} \right)$$

$$Z_{m,n} = X_{m,n} W_{n,n} = \left(\begin{array}{cccc} \mathbf{x}_1 \mathbf{w}_1 & \mathbf{x}_1 \mathbf{w}_2 & \cdots & \mathbf{x}_1 \mathbf{w}_n \\ \mathbf{x}_2 \mathbf{w}_1 & \mathbf{x}_2 \mathbf{w}_2 & \cdots & \mathbf{x}_2 \mathbf{w}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m \mathbf{w}_1 & \mathbf{x}_m \mathbf{w}_2 & \cdots & \mathbf{x}_m \mathbf{w}_n \end{array} \right)$$

$$W_{n,n} = \left(\begin{array}{cccc} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_n \\ | & | & & | \end{array} \right)$$

PORTANTO...

$$\mathbf{Z}_{m,n} = \mathbf{X}_{m,n} \mathbf{W}_{n,n} = \begin{pmatrix} \mathbf{x}_1 \mathbf{w}_1 & \mathbf{x}_1 \mathbf{w}_2 & \cdots & \mathbf{x}_1 \mathbf{w}_n \\ \mathbf{x}_2 \mathbf{w}_1 & \mathbf{x}_2 \mathbf{w}_2 & \cdots & \mathbf{x}_2 \mathbf{w}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m \mathbf{w}_1 & \mathbf{x}_m \mathbf{w}_2 & \cdots & \mathbf{x}_m \mathbf{w}_n \end{pmatrix}$$

$$\mathbf{Z}_{m,n} = \begin{pmatrix} | & | & | \\ \hat{\mathbf{z}}_1 & \hat{\mathbf{z}}_2 & \cdots & \hat{\mathbf{z}}_n \\ | & | & | \end{pmatrix} = \begin{pmatrix} | & \mathbf{z}_1 & | \\ - & \mathbf{z}_2 & - \\ & \vdots & \\ - & \mathbf{z}_m & - \end{pmatrix}$$

$$\mathbf{z}_i = (\mathbf{x}_i \mathbf{w}_1 \quad \mathbf{x}_i \mathbf{w}_2 \quad \cdots \quad \mathbf{x}_i \mathbf{w}_n)$$

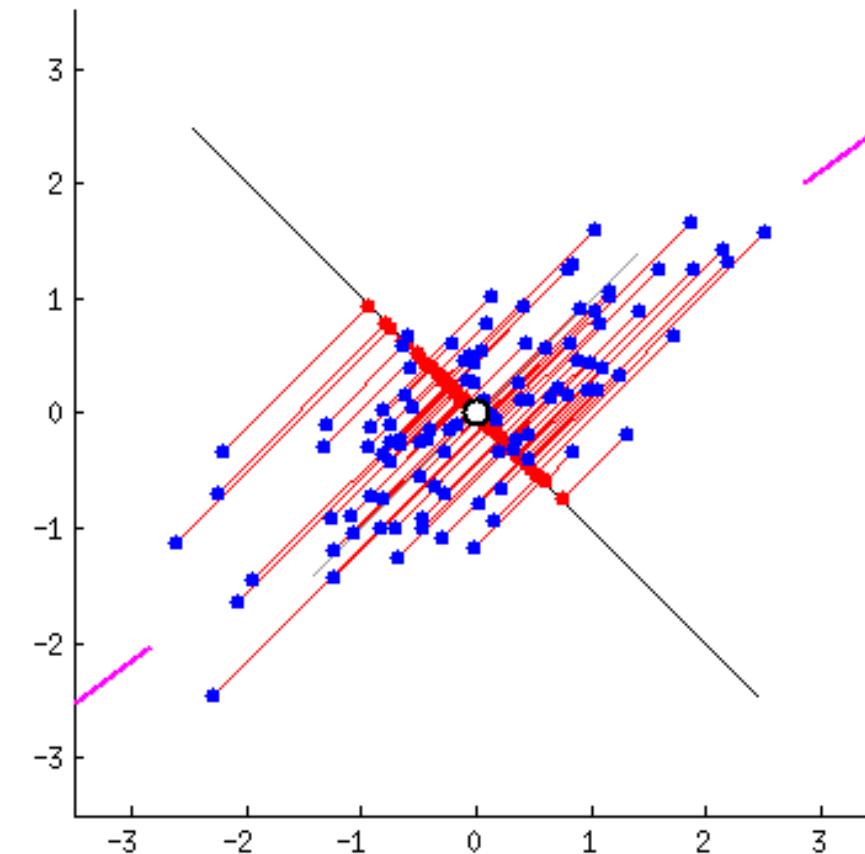
é a transformação da observação i . Ponto i no espaço é projetado nas novas direções $\mathbf{w}_i, i = 1, \dots, n$

$$\hat{\mathbf{z}}_i = \begin{pmatrix} \mathbf{x}_1 \mathbf{w}_i \\ \mathbf{x}_2 \mathbf{w}_i \\ \vdots \\ \mathbf{x}_m \mathbf{w}_i \end{pmatrix}$$

é a transformação dos m valores de uma determinada característica na direção \mathbf{w}_i

QUAL A PERGUNTA DO PCA?

Podemos agora declarar com mais precisão o que o PCA pergunta:





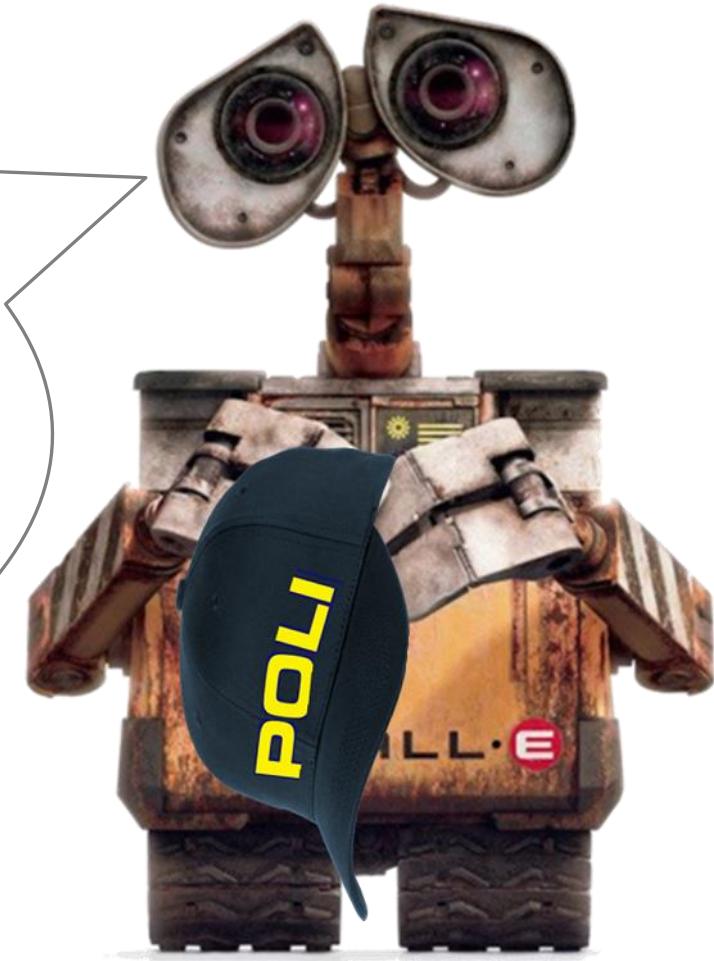
$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nn} \end{bmatrix}$$

Entendido!!!!

A melhor base equivale àquela em que as covariâncias de diferentes variáveis na matriz S devem ser tão próximas de zero quanto possível.

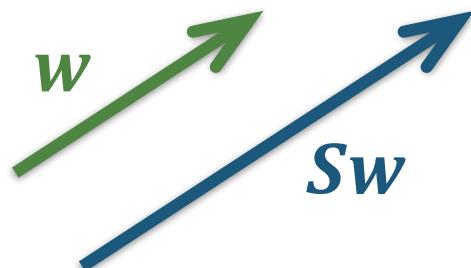
Por outro lado, variâncias grandes nos interessam, pois correspondem a dinâmicas interessantes no sistema (variâncias pequenas podem muito bem ser ruídos).

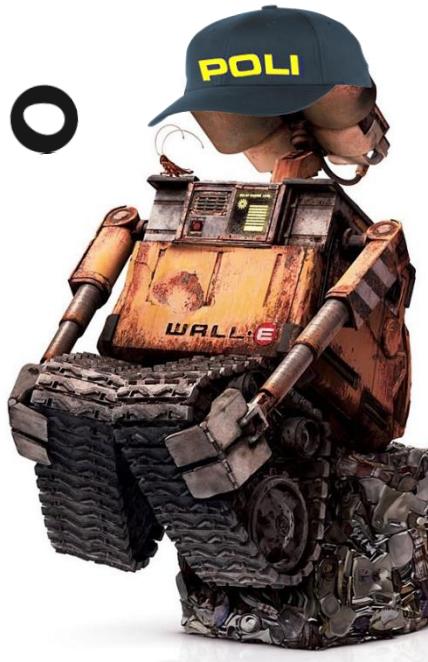
Devemos encontrar uma projeção dos dados em direções w_i que maximizem a variância dos dados originais.



**Isso nos
levará ao
óbvio!!!**

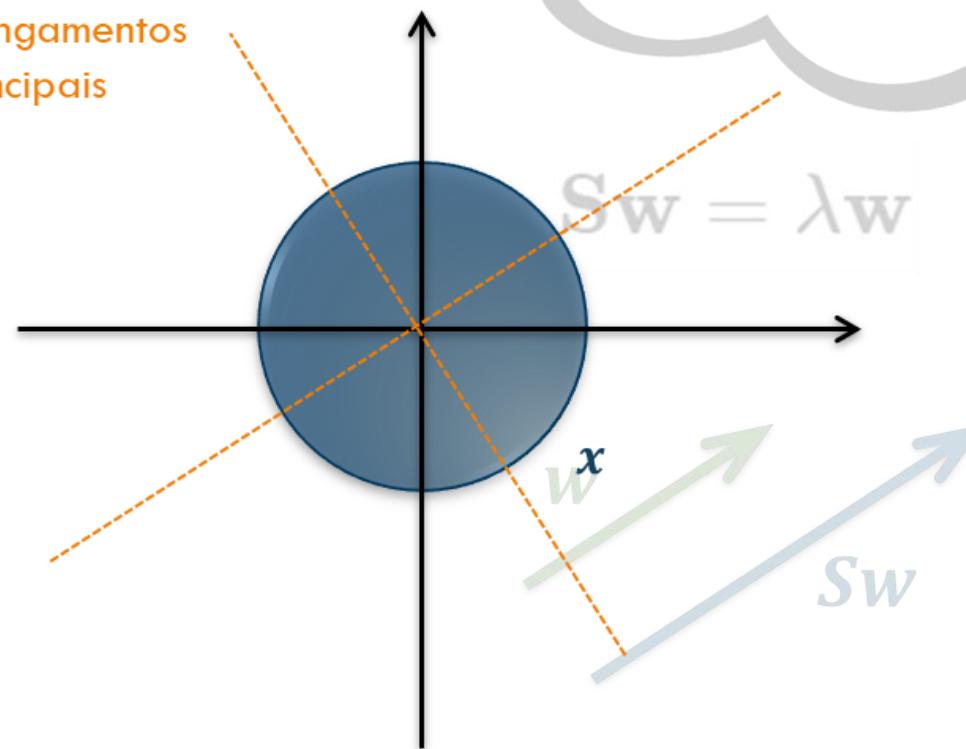
$$S_w = \lambda w$$


$$w$$
$$S_w$$

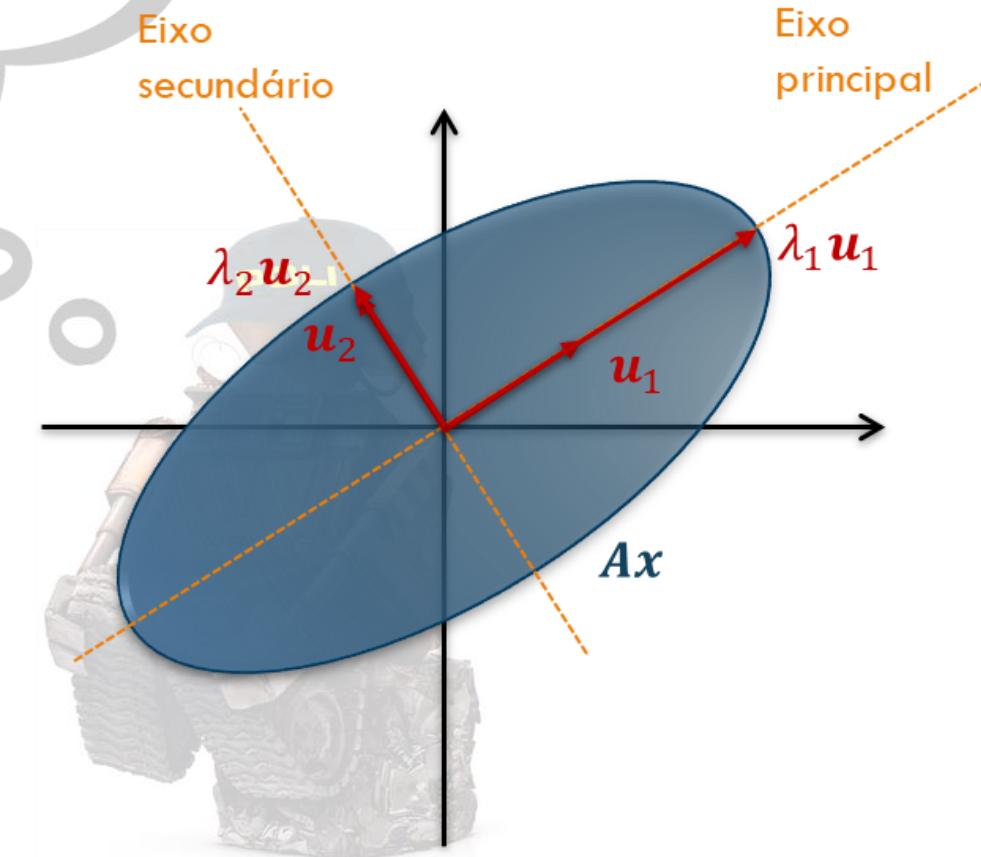


AUTOVALORES...

Direções de
alongamentos
principais



Isso nos
levará ao
óbvio!!!

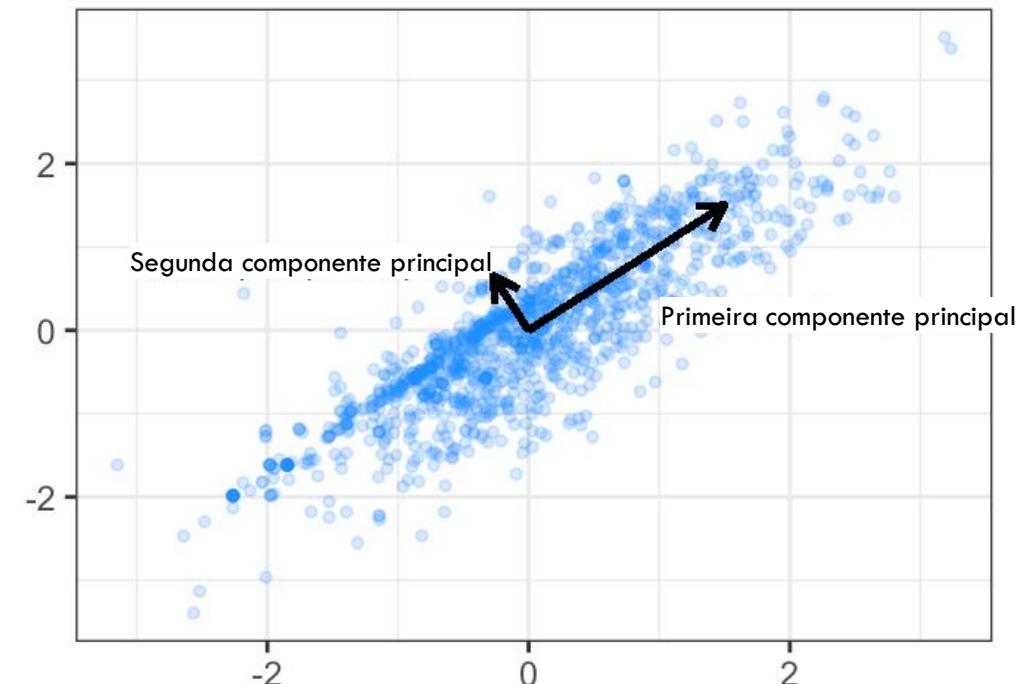


AUTOVALORES E AUTOVETORES DA MATRIZ DE COVARIÂNCIA

Os **autovetores** de uma matriz de covariância são chamados direções principais, pois correspondem às direções da variância máxima. As projeções dos dados nas direções principais são conhecidas como componentes principais.

Daí o nome **Análise de Componentes Principais**.

Quanto maior o autovalor, maior a quantidade de variação capturada por aquela componente principal. A variância capturada ao longo de cada CP pode ser calculada pela variância da projeção de X na direção principal.





```
class PCA:  
    def __init__(self, n_components):  
        self.n_components = n_components  
        self.components = None  
        self.mean = None  
        self.explained_variance = None
```

ALGORITMO

ETAPAS

Padronização
dos dados

$$z = x - \mu$$

```
def fit(self, X):  
    #1: Padronização dos dados (subtraia a média)  
    self.mean = np.mean(X, axis=0)  
    Xc = X - self.mean
```

ETAPAS

Padronização
dos dados

Computar a
matriz de
covariância

#2: Cálculo da matriz de covariância

```
cov_matrix = np.cov(Xc.T)
```

ETAPAS

Padronização
dos dados

Computar a
matriz de
covariância

Calcular
autovetores e
autovalores
da matriz

```
#3: Cálculo dos autovalores e autovetores  
eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)
```

```
#4: Ordenação dos autovalores e autovetores correspondentes  
sorted_indices = np.argsort(eigenvalues) [::-1]  
eigenvalues = eigenvalues[sorted_indices]  
eigenvectors = eigenvectors[:, sorted_indices]
```

ETAPAS

Padronização
dos dados

Computar a
matriz de
covariância

Calcular
autovetores e
autovalores
da matriz

Computar as
componentes
principais

```
# 5: Seleção das top n_components
self.components = eigenvectors[:, :self.n_components]
```

ETAPAS

Padronização
dos dados

Computar a
matriz de
covariância

Calcular
autovetores e
autovalores
da matriz

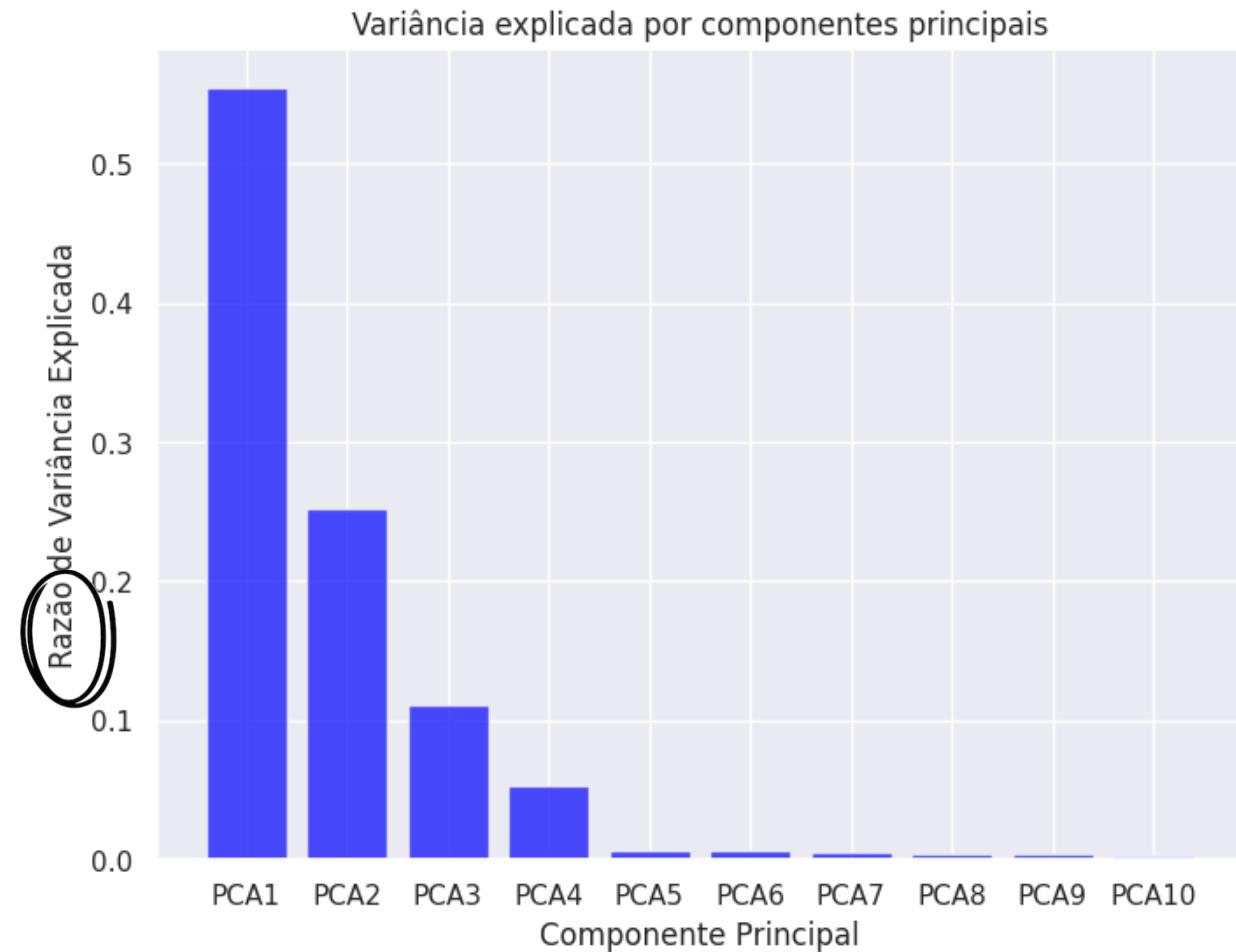
Computar as
componentes
principais

Reducir as
dimensões dos
dados

VARIÂNCIA EXPLICADA

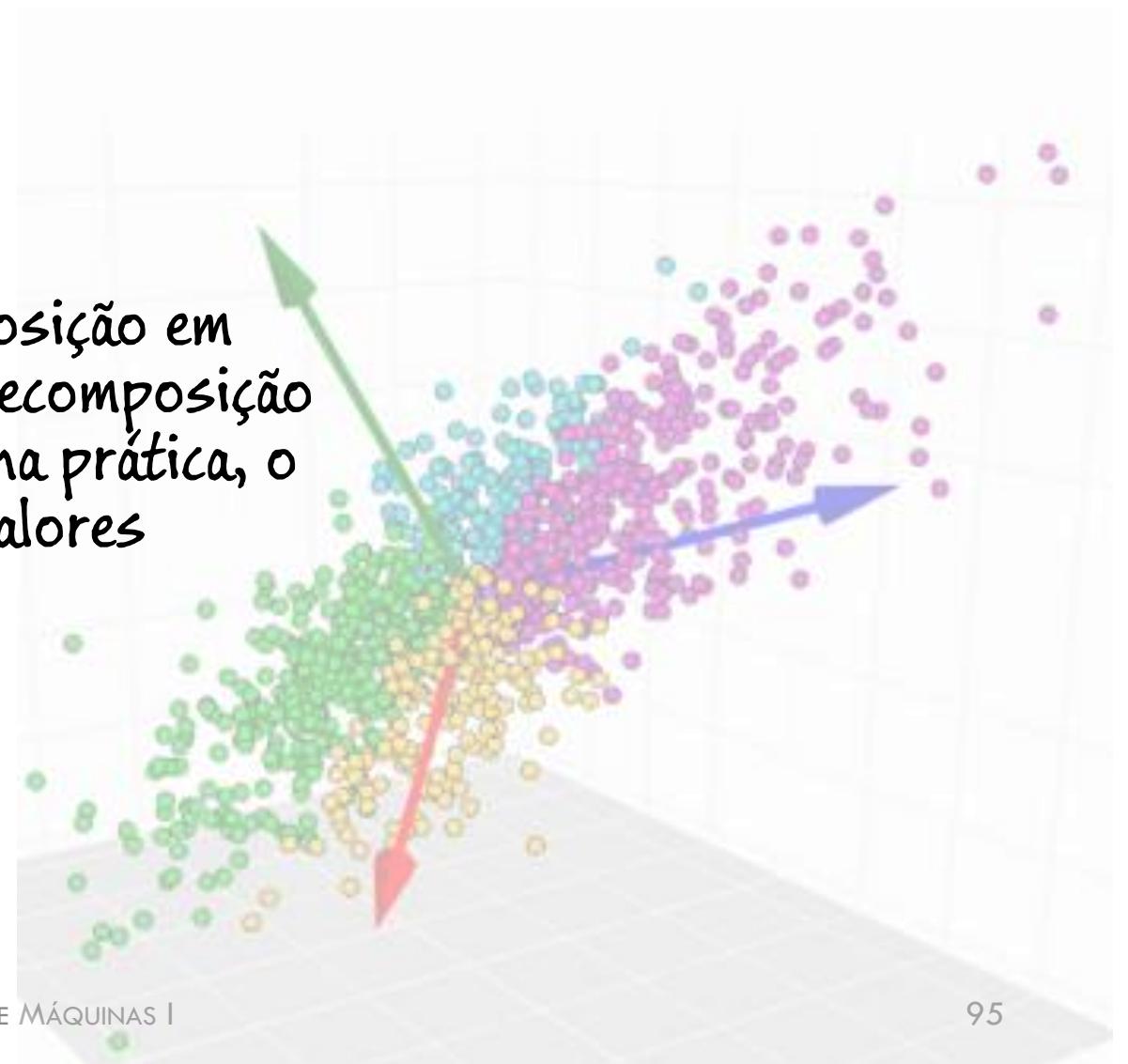
Os autovalores no PCA informam quanta variância pode ser explicada por seu autovetor associado. Portanto, o maior autovalor indica que a maior variância nos R dados foi observada na direção de seu autovetor. Consequentemente, se você juntar todos os autovetores, poderá explicar toda a variação na amostra de dados.

$$R(k) = \frac{\lambda_k}{\sum_{j=1}^n \lambda_j}$$



O PCA

O PCA, portanto, nada mais é que a decomposição em autovalores da matriz de covariância. Mas decomposição em autovalores pode ser bem ineficiente e, na prática, o PCA é calculado usando Decomposição em Valores Singulares (SVD, do inglês Singular Value Decomposition).



DECOMPOSIÇÃO EM AUTOVALORES

Toda **matriz simétrica quadrada**, como nossa matriz de covariância S , pode ser decomposta em,

$$S = V \Lambda V^T$$

$$S = \begin{pmatrix} & & & | \\ v_1 & v_2 & \dots & v_n \\ & & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{pmatrix} \begin{pmatrix} - & v_1^T & - \\ - & v_2^T & - \\ \vdots & \vdots & \vdots \\ - & v_n^T & - \end{pmatrix}$$

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$$

onde v_i são os **autovetores** de S e λ_i são os **autovalores** de S

DE FORMA QUE...

$$\begin{aligned}
 \Lambda V^T &\rightarrow \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ \vdots & \vdots & \vdots \\ - & v_n^T & - \end{bmatrix} = \\
 &\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{bmatrix} = \\
 &\begin{bmatrix} \lambda_1 v_{11} & \lambda_1 v_{12} & \cdots & \lambda_1 v_{1n} \\ \lambda_2 v_{21} & \lambda_2 v_{22} & \cdots & \lambda_2 v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_n v_{n1} & \lambda_n v_{n2} & \cdots & \lambda_n v_{nn} \end{bmatrix} = \begin{bmatrix} \lambda_1 v_1^T \\ \lambda_2 v_2^T \\ \vdots \\ \lambda_n v_n^T \end{bmatrix}
 \end{aligned}$$

PORTANTO...

$$S = V\Lambda V^T$$

$$S = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n] \begin{bmatrix} \lambda_1 \mathbf{v}_1^T \\ \lambda_2 \mathbf{v}_2^T \\ \vdots \\ \lambda_n \mathbf{v}_n^T \end{bmatrix}$$



$$S = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \cdots + \lambda_n \mathbf{v}_n \mathbf{v}_n^T$$

Portanto, a matriz S pode ser decomposta em n matrizes $\mathbf{v}_i \mathbf{v}_i^T$ $n \times n$ ponderadas de λ_i

```
def svd_reconstruction_with_variance(matrix, variance_threshold=0.80):  
    if np.all(matrix-matrix.T==0) == False:  
        raise ValueError("Matriz deve ser simétrica")  
    V, Lambda, VT = np.linalg.svd(matrix)  
  
    total_variance = np.sum(Lambda)  
  
    variance_sum = 0  
    num_singular_values = 0  
    variances = []  
    for i in range(len(Lambda)):  
        variance_sum += Lambda[i]  
        variances.append(variance_sum / total_variance)  
        if variance_sum / total_variance >= variance_threshold:  
            num_singular_values = i + 1  
            break  
  
    V_reduced = V[:, :num_singular_values]  
    Lambda_reduced = np.diag(Lambda[:num_singular_values])  
    VT_reduced = VT[:num_singular_values, :]  
  
    reconstructed_matrix = np.dot(V_reduced, np.dot(Lambda_reduced, VT_reduced))  
    return reconstructed_matrix, num_singular_values, variances
```

SVD da matriz simétrica

Variância total

Número de valores singulares necessários para alcançar o limite mínimo de variância dos dados

Uso apenas dos principais valores singulares definidos em 'num_singular_values' para reconstruir a matriz

```
A = np.array([[4, 1],  
             [1, 3]])  
var_threshold = 0.6  
Ar, num_singular_values, variances = svd_reconstruction_with_variance(A, var_threshold)  
  
print("Matriz original:")  
print(A)  
print("\nMatriz reconstruída com {:.0f}% de variância:".format(var_threshold*100))  
print(Ar)  
print("\nNúmero de valores singulares usados:")  
print(num_singular_values)  
  
print("\nPorcentagem acumulada da variância total de cada valor singular utilizado:")  
print(variances)
```

```
A = np.array([[4, 1],
             [1, 3]])
var_threshold = 0.6
Ar, num_singular_values, variances = svd_reconstruction_with_variance(A, var_threshold)

print("Matriz original:")
print(A)
print("\nMatriz reconstruída com 60% de variância:")
print(Ar)
print("\nNúmero de valores singulares usados:")
print(num_singular_values)

print("\nPorcentagem acumulada da variância total de cada valor singular utilizado:")
print(variances)
```

```
Matriz original:
[[4 1]
 [1 3]]
Matriz reconstruída com 60% de variância:
[[3.34164079 2.06524758]
 [2.06524758 1.2763932 ]]
Número de valores singulares usados:
1
Porcentagem acumulada da variância total de cada valor singular utilizado:
[0.659719141249985]
```

PORTANTO, DEVEMOS USAR v_i PARA TRANSFORMAÇÃO DE BASE

Veja que, se transformamos o vetor de dados da seguinte forma:

$$\mathbf{Z} = \mathbf{X}_c \mathbf{V}$$

Basta multiplicar o conjunto de dados, após centralizado, pela matriz de autovetores encontrada na decomposição da matriz de covariância.

Lembram-se? **Variância máxima e mínima covariância...** Então vamos analisar a matriz de covariância da nova base,

$$S_z = \frac{1}{m-1} \mathbf{Z}^T \mathbf{Z}$$

COMO FICA A MATRIZ DE COVARIÂNCIA DEPOIS DA TRANSFORMAÇÃO?

$$\begin{aligned}
 \mathbf{S}_z &= \frac{1}{m-1} (\mathbf{X}_c \mathbf{V})^T (\mathbf{X}_c \mathbf{V}) \\
 &= \frac{1}{m-1} \mathbf{V}^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{V} \\
 &= \frac{1}{m-1} \mathbf{V}^T (\mathbf{X}_c^T \mathbf{X}_c) \mathbf{V} \\
 &= \mathbf{V}^T \mathbf{S} \mathbf{V}
 \end{aligned}$$

S_z possui máxima variância!!!

$$\mathbf{S}_z = \frac{1}{m-1} \mathbf{Z}^T \mathbf{Z}$$

$\mathbf{Z} = \mathbf{X}_c \mathbf{V}$

$$S = V \Lambda V^T \rightarrow$$

$$\begin{aligned}
 \mathbf{S}_z &= \mathbf{V}^T (\mathbf{V} \Lambda \mathbf{V}^T) \mathbf{V} \\
 &= (\mathbf{V}^T \mathbf{V}) \Lambda (\mathbf{V}^T \mathbf{V}) \\
 &= \Lambda
 \end{aligned}$$

Dessa forma, S_z é diagonal, e esse era outro objetivo do PCA!!!

SVD PARA MATRIZES NÃO SIMÉTRICAS

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T$$

onde U e V são matrizes ortogonais

$$V = \begin{pmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_n \\ | & | & & | \end{pmatrix}$$

$$U = \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_m \\ | & | & & | \end{pmatrix}$$

onde v_i são os autovetores de $A^T A$ e u_i são os autovetores de AA^T , e Σ é diagonal, cujos elementos diagonais da matriz são os valores singulares $\sigma_i = \sqrt{\lambda_i}$, computados em ordem decrescente $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$

PARTINDO DA DECOMPOSIÇÃO EM VALOR SINGULAR DE \mathbf{X}_c

$$\mathbf{X}_c = \boxed{\mathbf{U}} \boxed{\Sigma} \boxed{\mathbf{V}}^T$$

autovetor $\mathbf{S} = \frac{1}{m-1} \mathbf{V} \Lambda \mathbf{V}^T$ autovetores de $\mathbf{X}_c^T \mathbf{X}_c$

$$\mathbf{S} = \frac{1}{m-1} \mathbf{X}_c^T \mathbf{X}_c = \frac{1}{m-1} (\mathbf{U} \Sigma \mathbf{V}^T)^T (\mathbf{U} \Sigma \mathbf{V}^T) = \frac{1}{m-1} (\mathbf{V} \Sigma \mathbf{U}^T) (\mathbf{U} \Sigma \mathbf{V}^T)$$

$$\mathbf{S} = \frac{1}{m-1} \mathbf{X}_c^T \mathbf{X}_c = \frac{1}{m-1} \mathbf{V} \Sigma^2 \mathbf{V}^T = \mathbf{V} \Lambda \mathbf{V}^T$$

$$\Lambda = \frac{1}{m-1} \Sigma^2$$



$$\lambda_i = \frac{\sigma_i^2}{m-1}$$

ETAPAS

Padronização
dos dados: X_c

SVD da
matriz X_c

Computar as
componentes
principais

Reducir as
dimensões
dos dados

```
class PCA:  
    def __init__(self, n_components):  
        self.n_components = n_components  
        self.components = None  
        self.mean = None  
        self.explained_variance = None
```

ETAPAS

Padronização
dos dados: X_c

SVD da
matriz X_c

Computar as
componentes
principais

Reducir as
dimensões
dos dados

```
def fit(self, X):  
    self.mean = np.mean(X, axis=0)  
    Xc = X - self.mean
```

Depois de normalização da média (todas as características
DEVEM ter média zero) e, opcionalmente, escalonamento das
variáveis

ETAPAS

Padronização
dos dados: X_c

SVD da
matriz X_c

Computar as
componentes
principais

Reducir as
dimensões
dos dados

```
U, sigma, VT = np.linalg.svd(Xc)
self.components_ = VT.T[:, :self.n_components]
```

ETAPAS

Padronização
dos dados: X_c

SVD da
matriz X_c

Computar as
componentes
principais

Reducir as
dimensões
dos dados

```
def transform(self, X):  
    # 6: Projeção dos dados nas componentes selecionadas  
    Xc = X - self.mean  
    Zk = np.dot(Xc, self.components)  
    Xk = np.dot(Zk, self.components.T)  
    return Zk, Xk
```

REDUÇÃO

$$\mathbf{Z}_{m,k} = \mathbf{X}_{m,n} \mathbf{V}_{n,k} = \begin{pmatrix} \mathbf{x}_1 \mathbf{v}_1 & \mathbf{x}_1 \mathbf{v}_2 & \cdots & \mathbf{x}_1 \mathbf{v}_k \\ \mathbf{x}_2 \mathbf{v}_1 & \mathbf{x}_2 \mathbf{v}_2 & \cdots & \mathbf{x}_2 \mathbf{v}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_m \mathbf{v}_1 & \mathbf{x}_m \mathbf{v}_2 & \cdots & \mathbf{x}_m \mathbf{v}_k \end{pmatrix}$$

$$\mathbf{V}_{n,n} = \begin{pmatrix} | & | & & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_k & \cdots & \mathbf{v}_n \\ | & | & & | & & | \end{pmatrix}$$

$$\mathbf{Z}_{(m,k)} = \mathbf{X}_{(m,n)} \mathbf{V}_{(n,k)}$$

Selecionados os k primeiros Componentes Principais



```
class PCA:  
    def __init__(self, n_components):  
        self.n_components = n_components  
        self.components = None  
        self.mean = None  
        self.explained_variance = None  
  
    def fit(self, X):  
        #1: Padronização dos dados (subtração da média)  
        self.mean = np.mean(X, axis=0)  
        Xc = X - self.mean  
  
        #2: Cálculo das componentes principais  
        U, sigmai, VT = np.linalg.svd(Xc)  
        self.components = VT.T[:, :self.n_components]  
  
        #3: Cálculo da variância explicada  
        lambdai=sigmai**2/(len(U)-1)  
        total_variance = np.sum(lambdai)  
        self.explained_variance = lambdai[:self.n_components] / total_variance
```

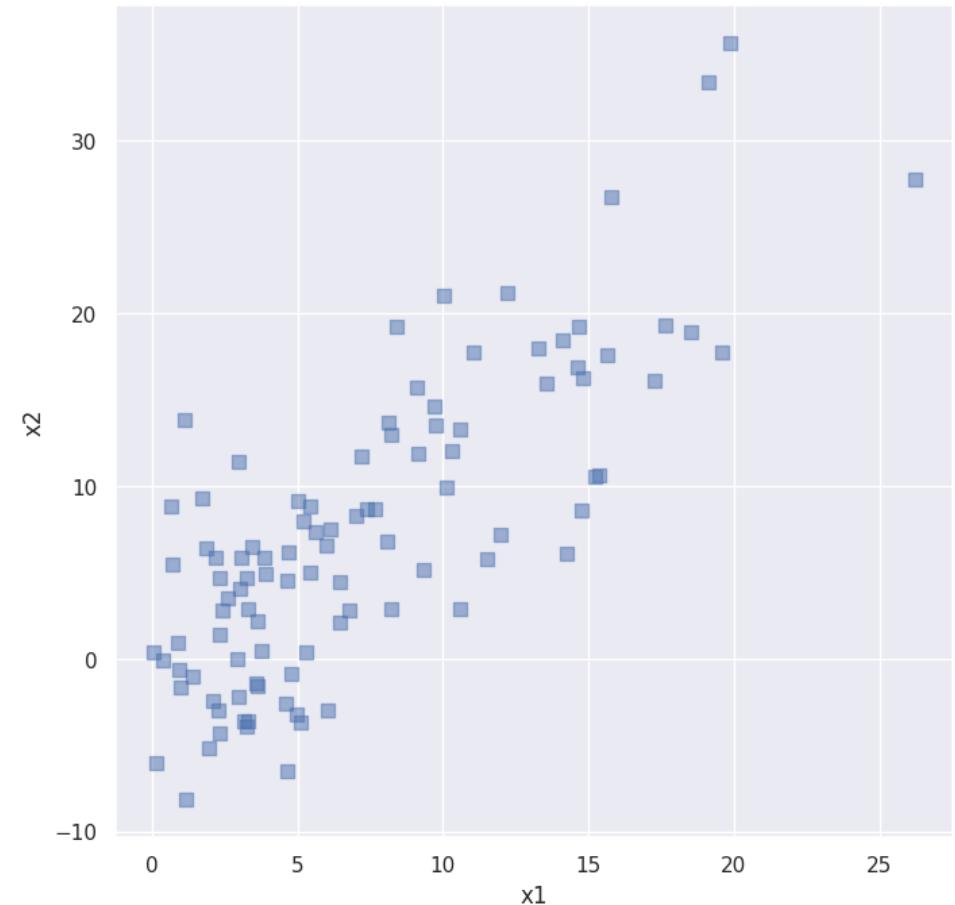


```
def transform(self, X):  
    # Projeção dos dados nas componentes selecionadas  
    Xc = X - self.mean  
    Zk = np.dot(Xc, self.components)  
    Xk = np.dot(Zk, self.components.T)  
    return Zk, Xk  
  
def inverse_transform(self, Zk):  
    # Reconstrução dos dados  
    return np.dot(Xc, self.components)  
  
def get(self):  
    return self.components, self.mean, self.explained_variance  
  
def plot_explained_variance(self):  
    # Criação de rótulos para cada componente principal  
    labels = [f'PCA{i+1}' for i in range(self.n_components)]  
  
    # Criação de um gráfico de barras para variação explicada  
    plt.figure(figsize=(8, 6))  
    plt.bar(range(1, self.n_components + 1), self.explained_variance,  
            alpha=0.7, align='center', color='blue', tick_label=labels)
```

EXEMPLO

Para matriz X ,

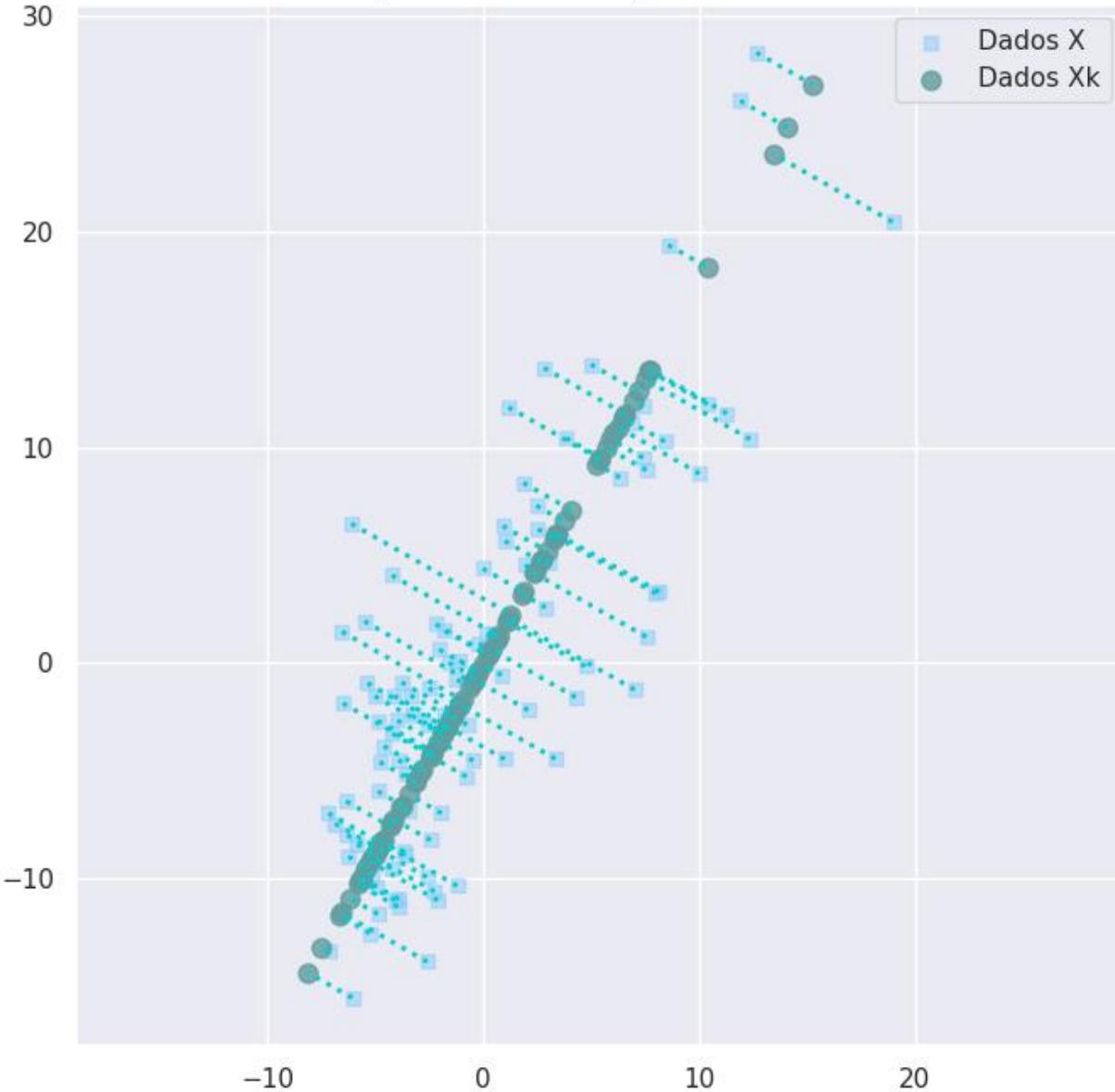
```
m = 100
n = 2
np.random.seed(42)
x1 = 10.*np.sqrt((np.random.normal(size=m))**2)
x2 = x1 + 5.8*(np.random.normal(size=m))
10.*np.sqrt((np.random.normal(size=m))**2)
X = np.stack((x1,x2),axis=1)
```



```
pca = PCA(n_components=2)
pca.fit(X)
cps, meanX, explained_variance = pca.get()
print("Variância explicada:\n", explained_variance)
pca.plot_explained_variance()
```



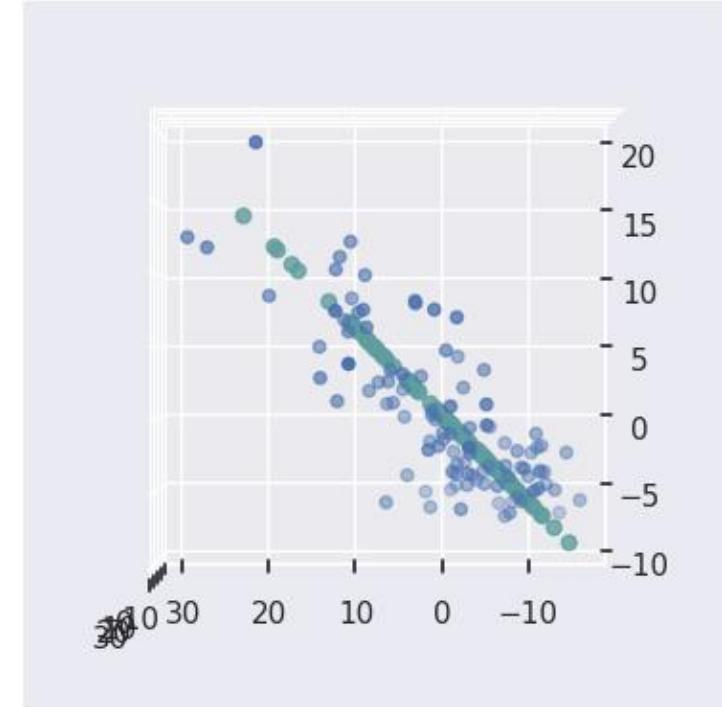
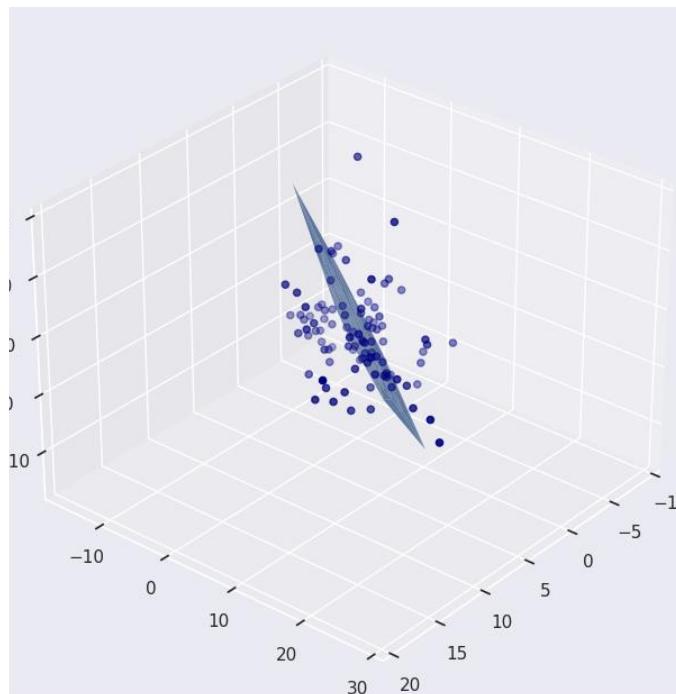
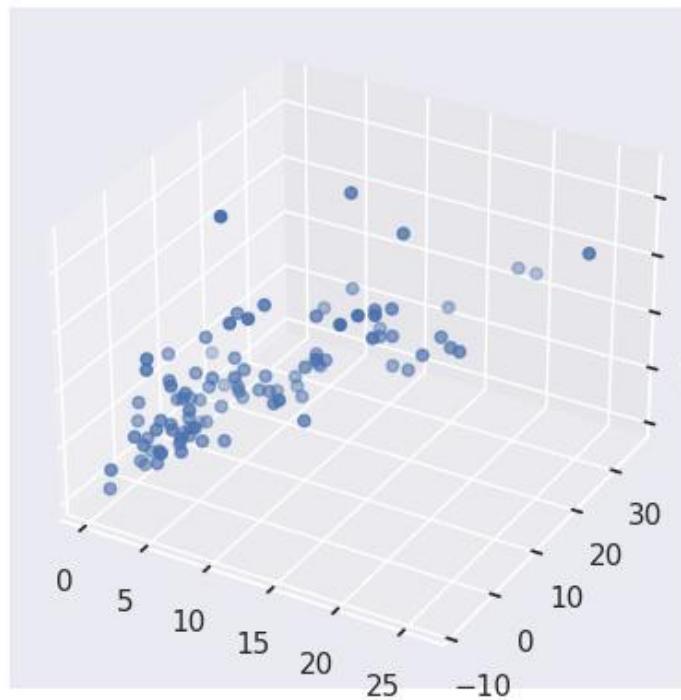
Visualização dos Dados Originais e Reconstituídos



```
pca = PCA(n_components=1)
pca.fit(X)
cps, meanX, explained_variance = pca.get()

Xc = X - meanX
Zk, Xk = pca.transform(X)
```

EXEMPLO 3D DO NOTEBOOK



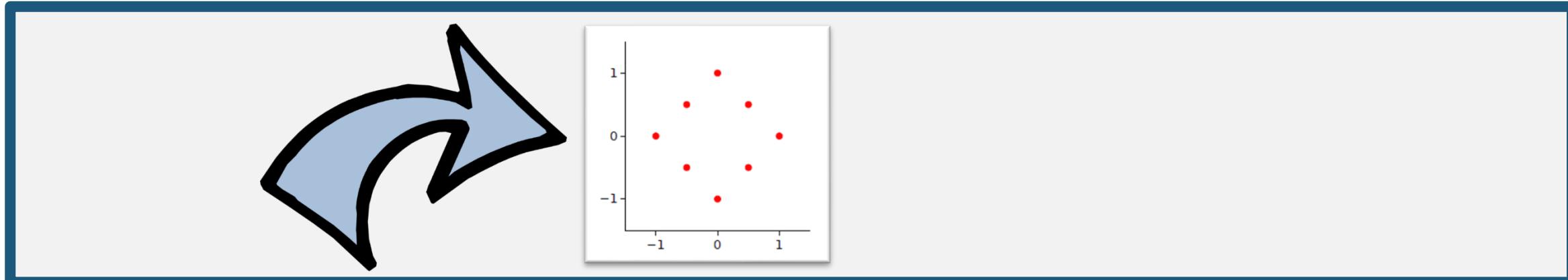
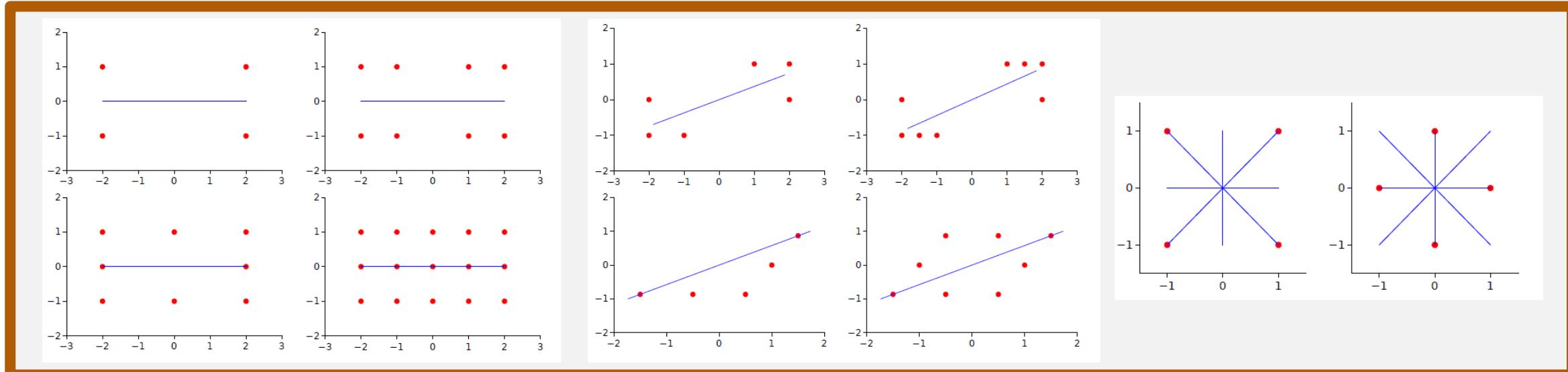


SEUS ESTUDOS

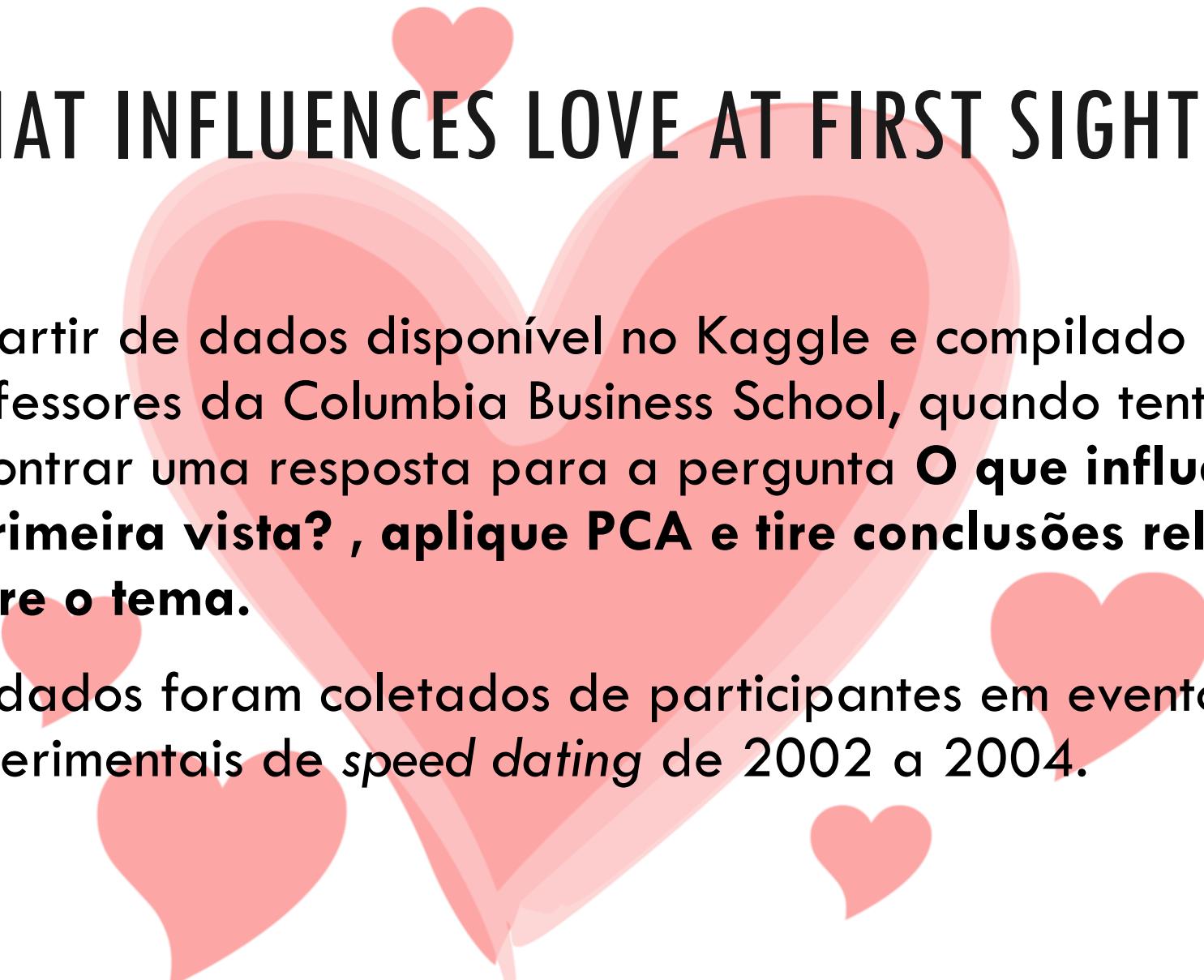
Explique as direções principais (em azul) das imagens

Parte I e construa as componentes principais da Parte II.

Figuras extraídas de: *Principal Component Analysis*, Leow Wee Kheng, National University of Singapore (NUS)



WHAT INFLUENCES LOVE AT FIRST SIGHT?



A partir de dados disponível no Kaggle e compilado por professores da Columbia Business School, quando tentavam encontrar uma resposta para a pergunta **O que influencia o amor à primeira vista? , aplique PCA e tire conclusões relevantes sobre o tema.**

Os dados foram coletados de participantes em eventos experimentais de speed dating de 2002 a 2004.



FINAL DA AULA

Próxima aula:
exercícios.