Saturdays.AI
Barcelona

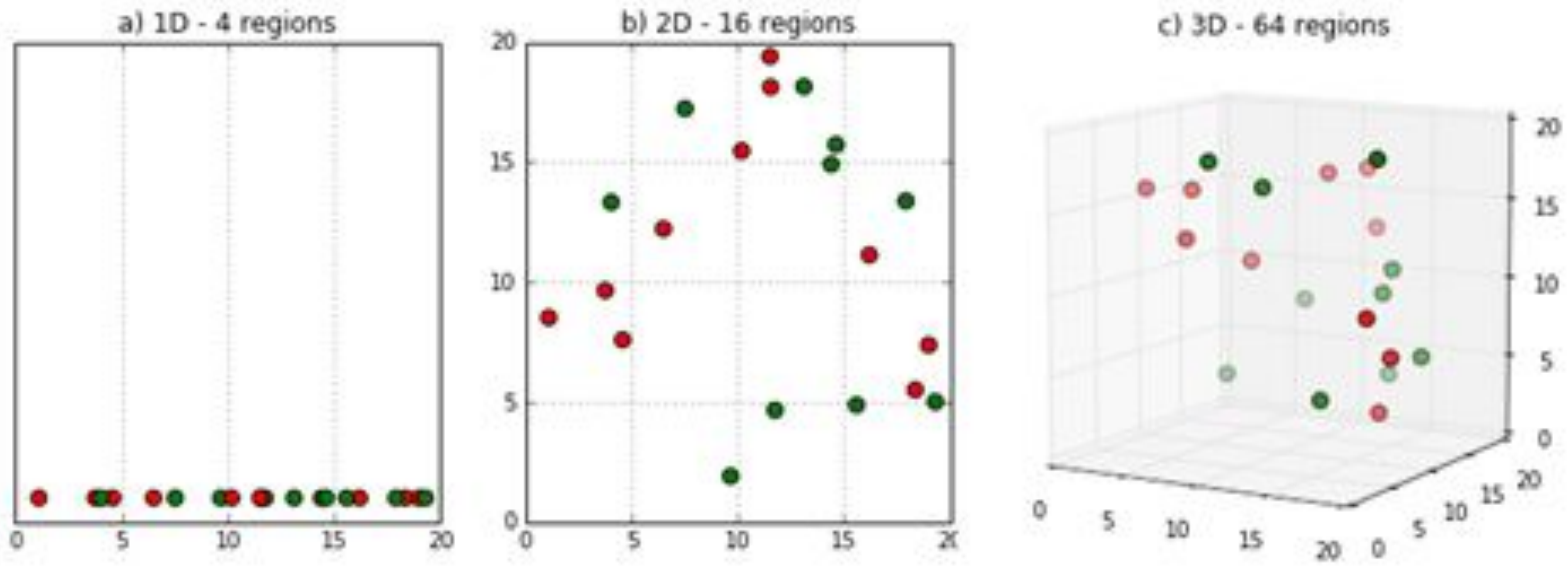
Dimensionality Reduction: PCA

by Saturdays AI

Get ready for the future AI!

Why dimensionality reduction?

When data has a high dimension (many features), it is extremely complex to process due to inconsistencies in the features, which increase the computational time processing and requires more evoluted EDA (Exploratory Data Analysis).

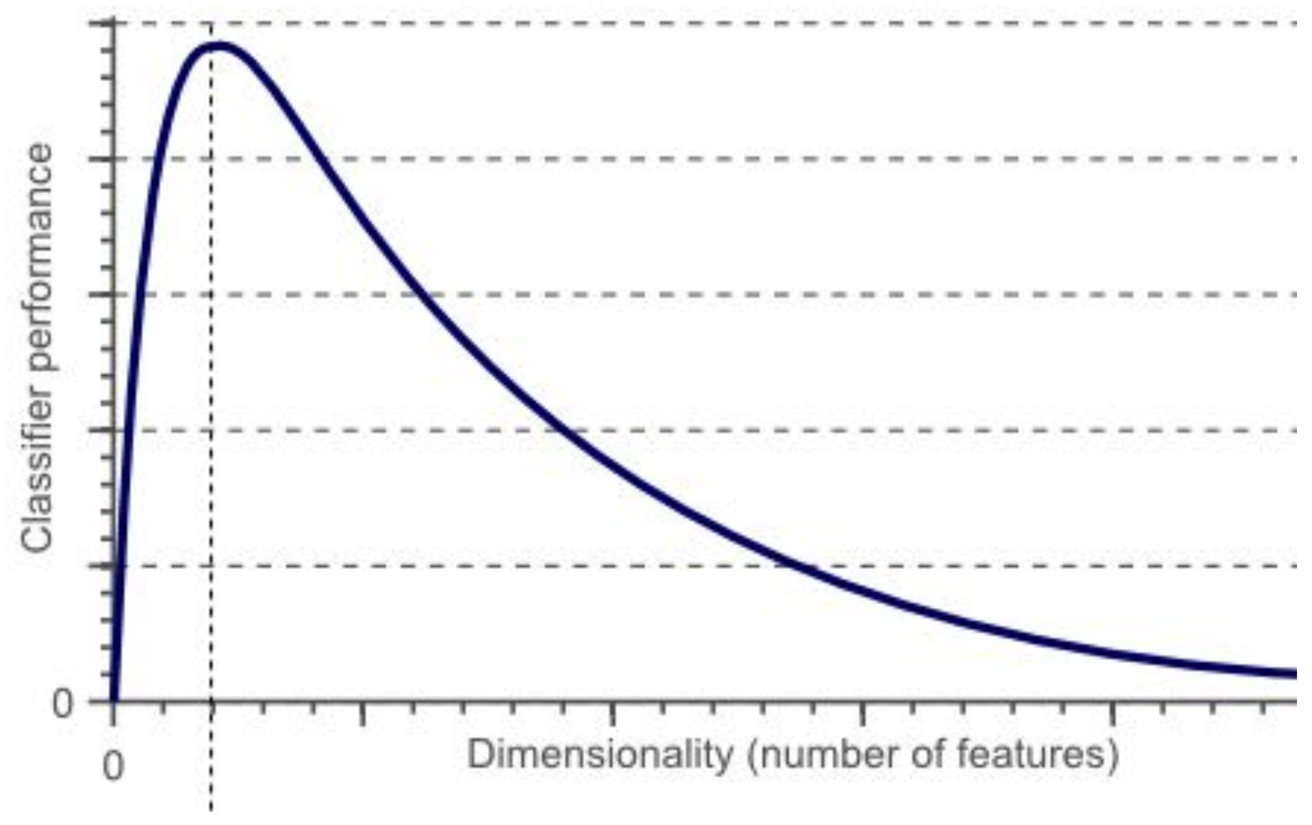


???

1D 2D 3D xD

Dimensionality Reduction, Overview

- **Goal:** reduce the number of features (dimensionality) by maximizing the explained variance, to obtaining a set of principal features.
- **How does it work?:** Transforming the data in the high-dimensional space to a space in fewer dimensions.
- **Advantages:**
  - Removes inconsistencies in the features
  - Highlight relevant features, not all features are relevant to our problem
  - Avoids overfitting due to strong correlations
  - Reduces computational time and space complexity
- **Disadvantages**
  - More difficult to explain the meaning
  - We fundamentally “miss” some dat



What is PCA?

Principal component analysis (PCA) is a dimensionality reduction technique that enables to identify correlations and patterns in a dataset so it can be transformed into a dataset of significant lower dimensions and keeping the most relevant information.

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

	principal component 1	principal component 2
0	-2.264542	0.505704
1	-2.086426	-0.655405
2	-2.367950	-0.318477
3	-2.304197	-0.575368
4	-2.388777	0.674767

What is PCA (math definition)?

Principal component analysis (PCA) is statistical procedure that uses an **orthogonal transformation** to convert a set of observations of possibly correlated variables into a set of values of **linearly uncorrelated variables** called principal components.

Definition source: https://en.wikipedia.org/wiki/Principal_component_analysis

PCA basics

Let’s practice with the Basic notebook: https://ja.cat/PCA_basics

PCA step by step

Standardize the data

Build the covariance matrix

Calculate the Eigenvectors and Eigenvalues

Compute Principal Components

Reduce the data dimensions

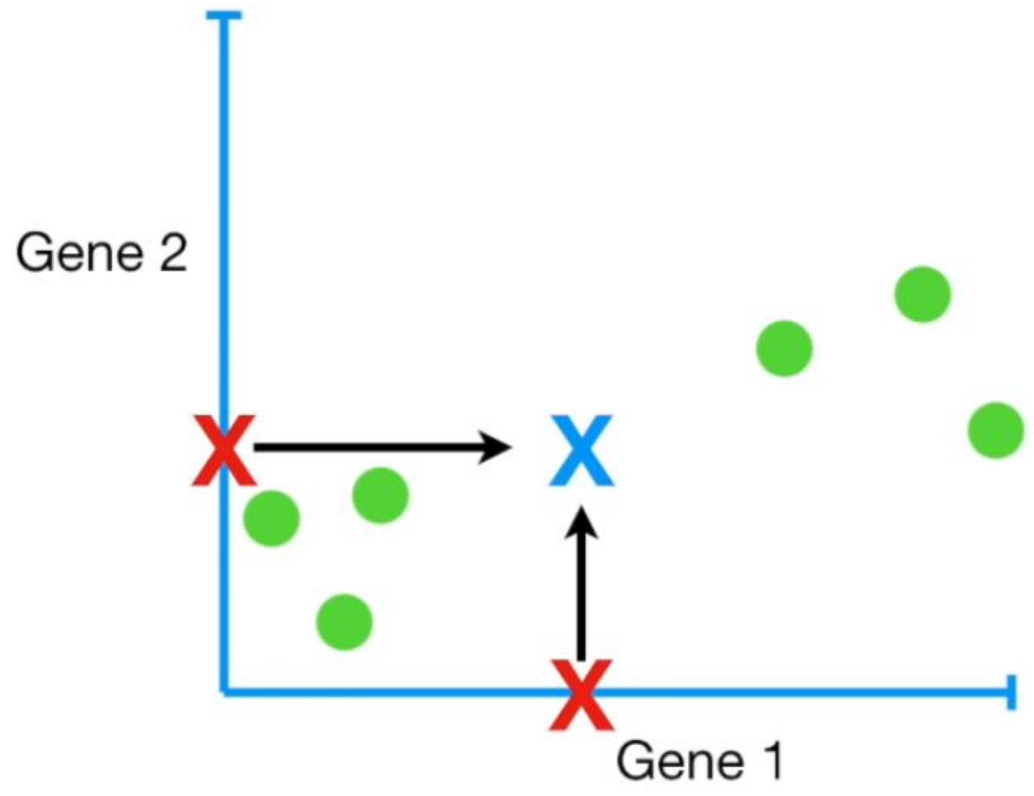
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

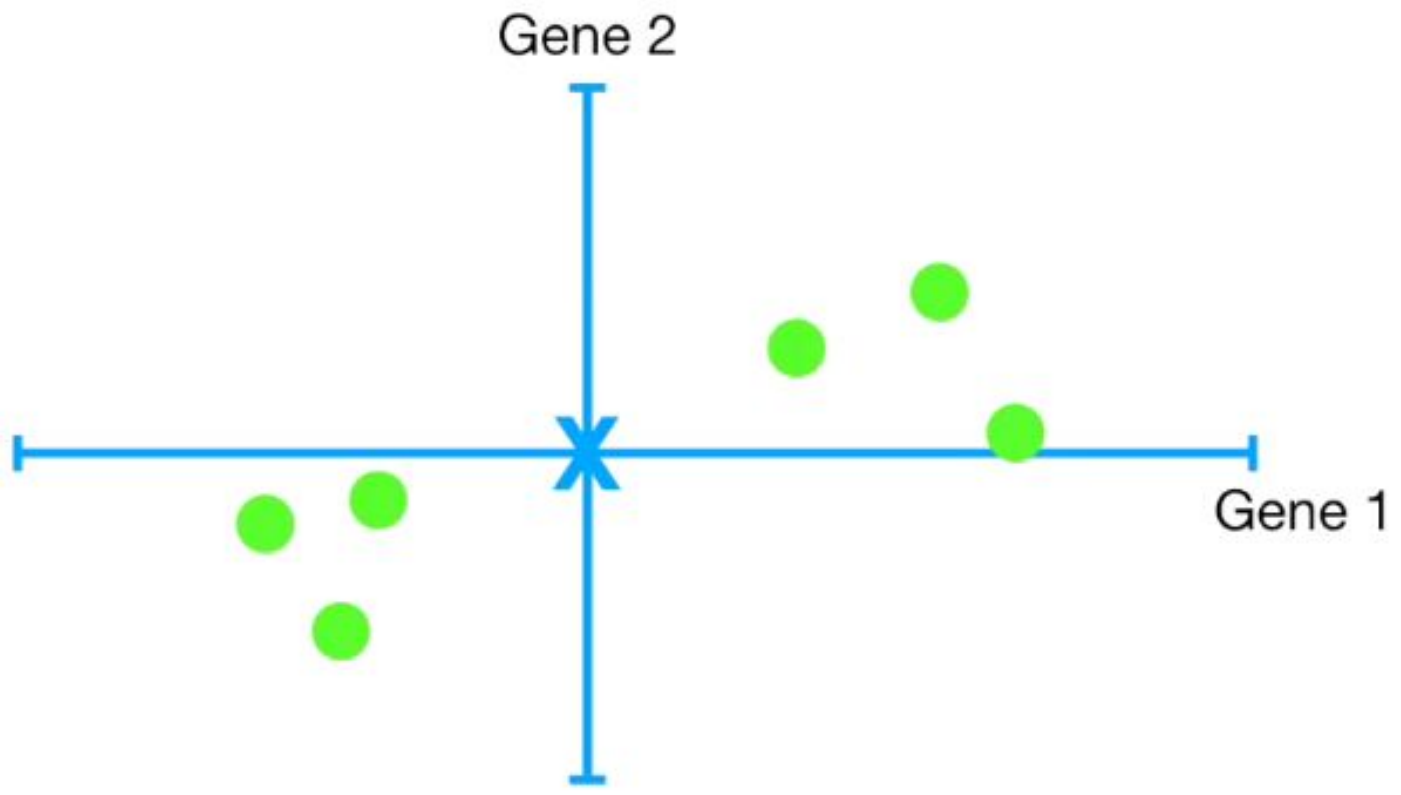
Definition source: https://en.wikipedia.org/wiki/Principal_component_analysis
Images source: StatQuest: PCA step by Step

PCA step by step

Standardize the data

	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1



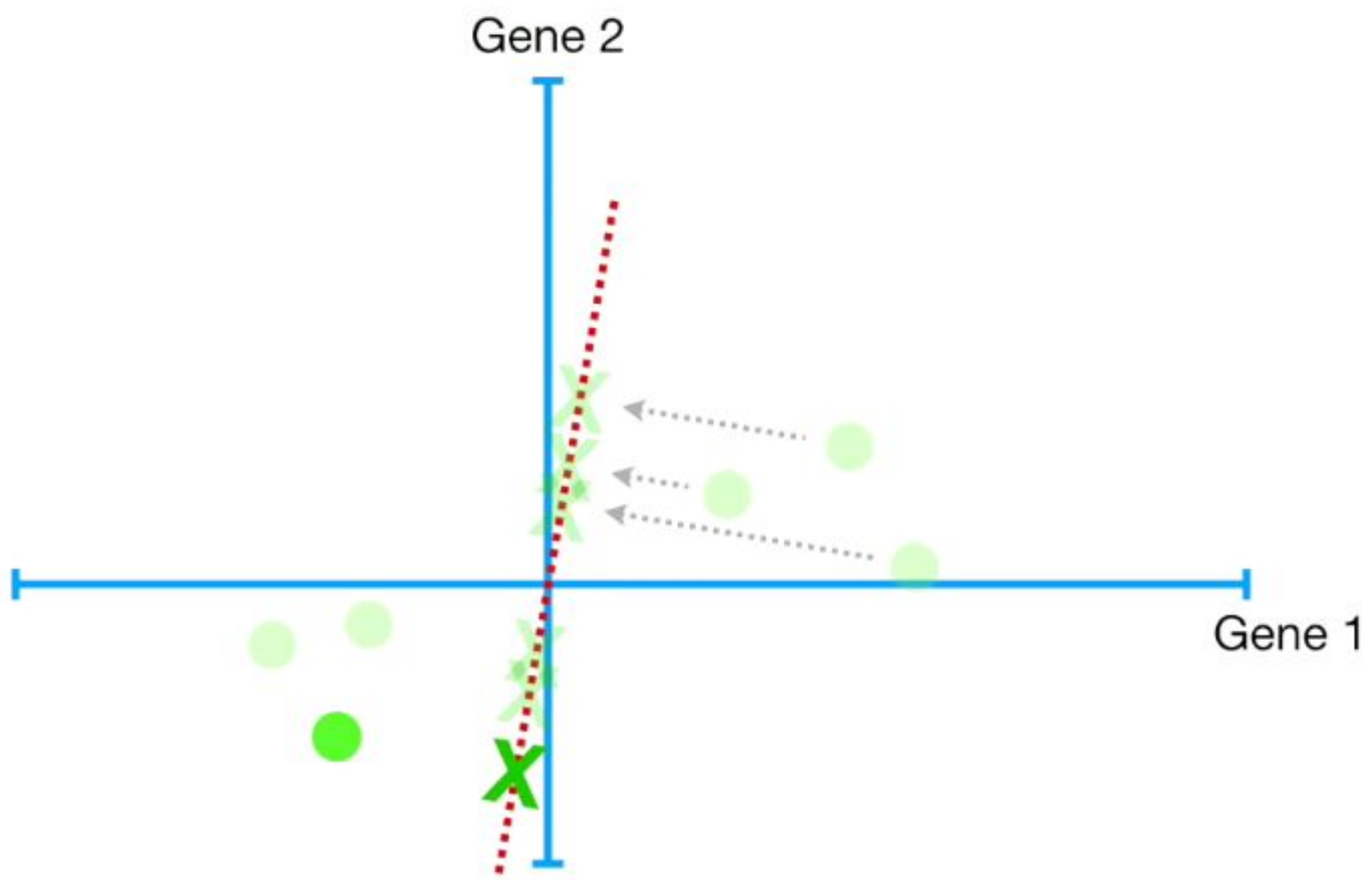


Images source: [StatQuest: PCA step by Step]

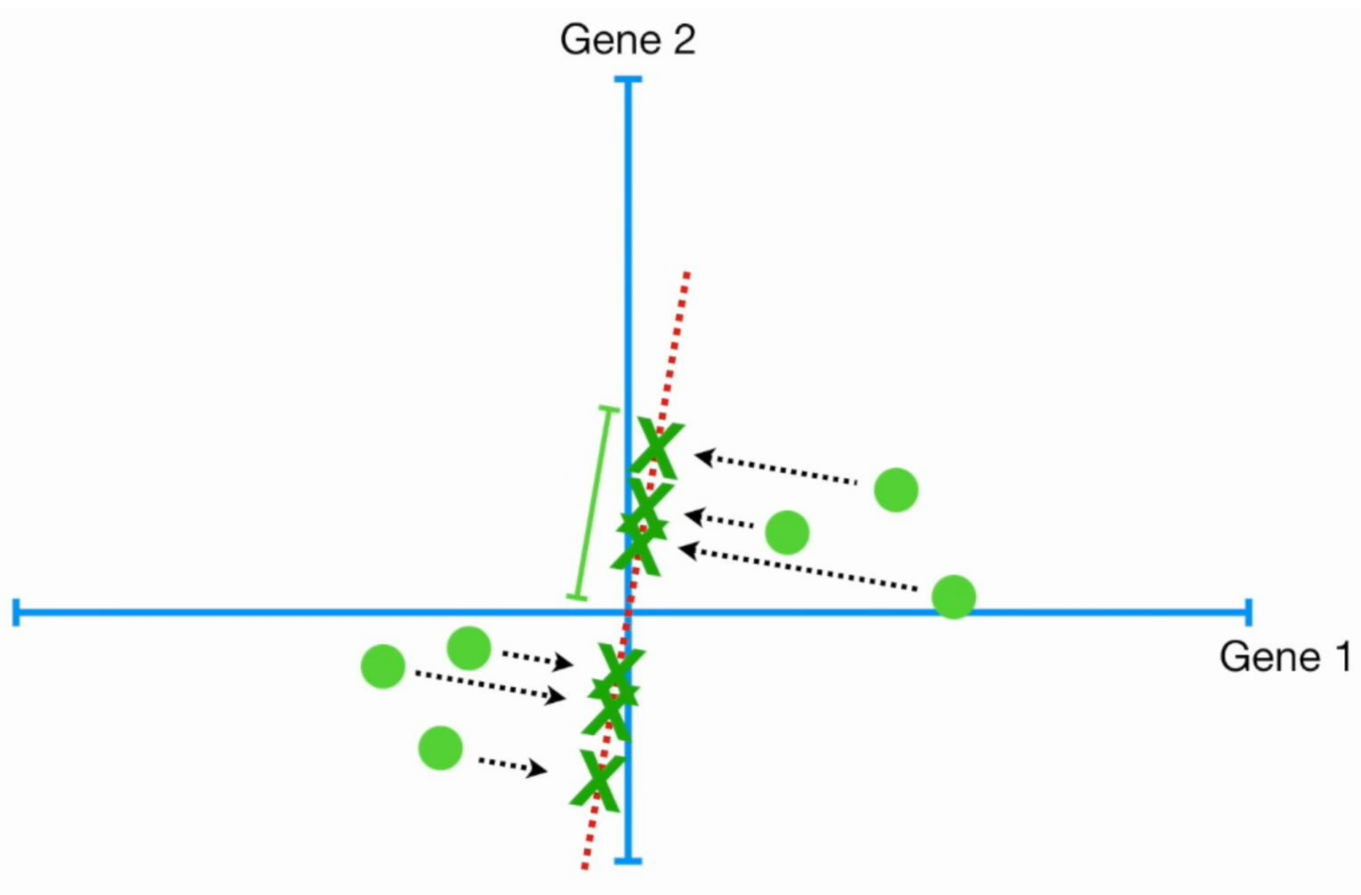
PCA step by step

Build the covariance matrix

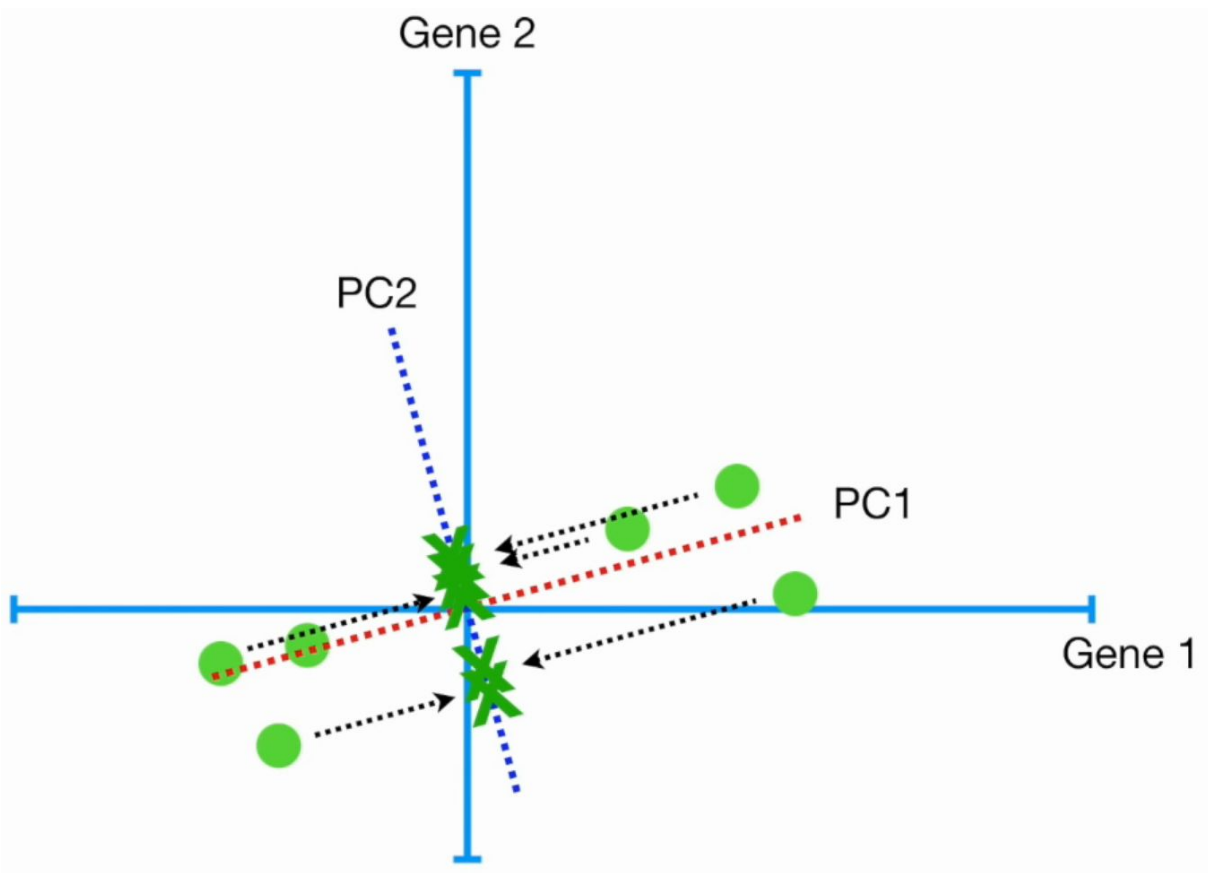
Calculate the Eigenvectors and Eigenvalues



Orthogonal transformation feature 1



Orthogonal transformation feature 2

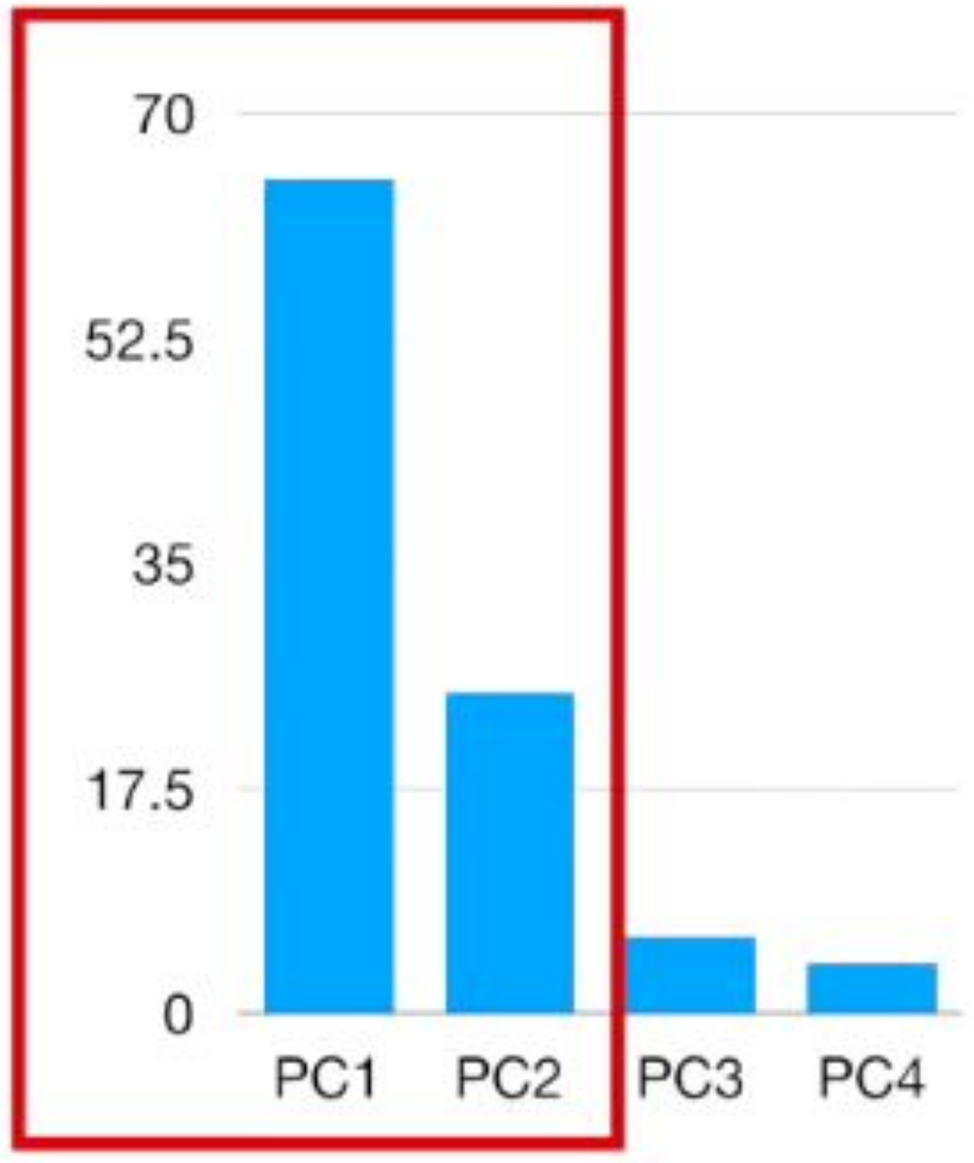


Images source: [StatQuest: PCA step by Step]

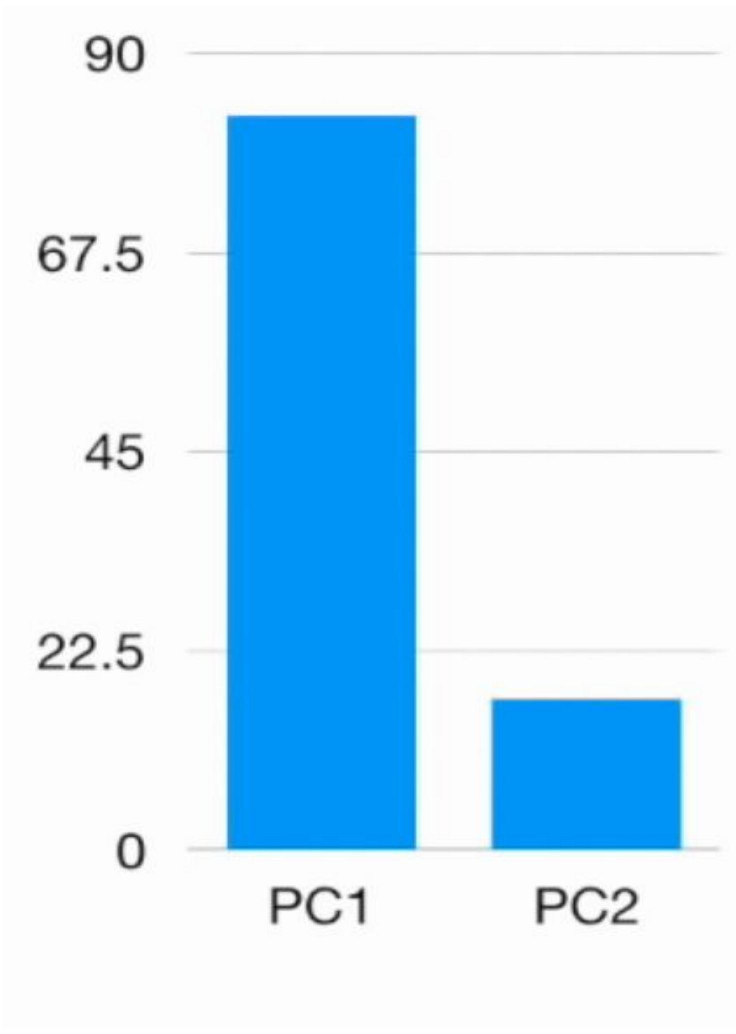
PCA step by step

Compute Principal Components
Reduce the data dimensions

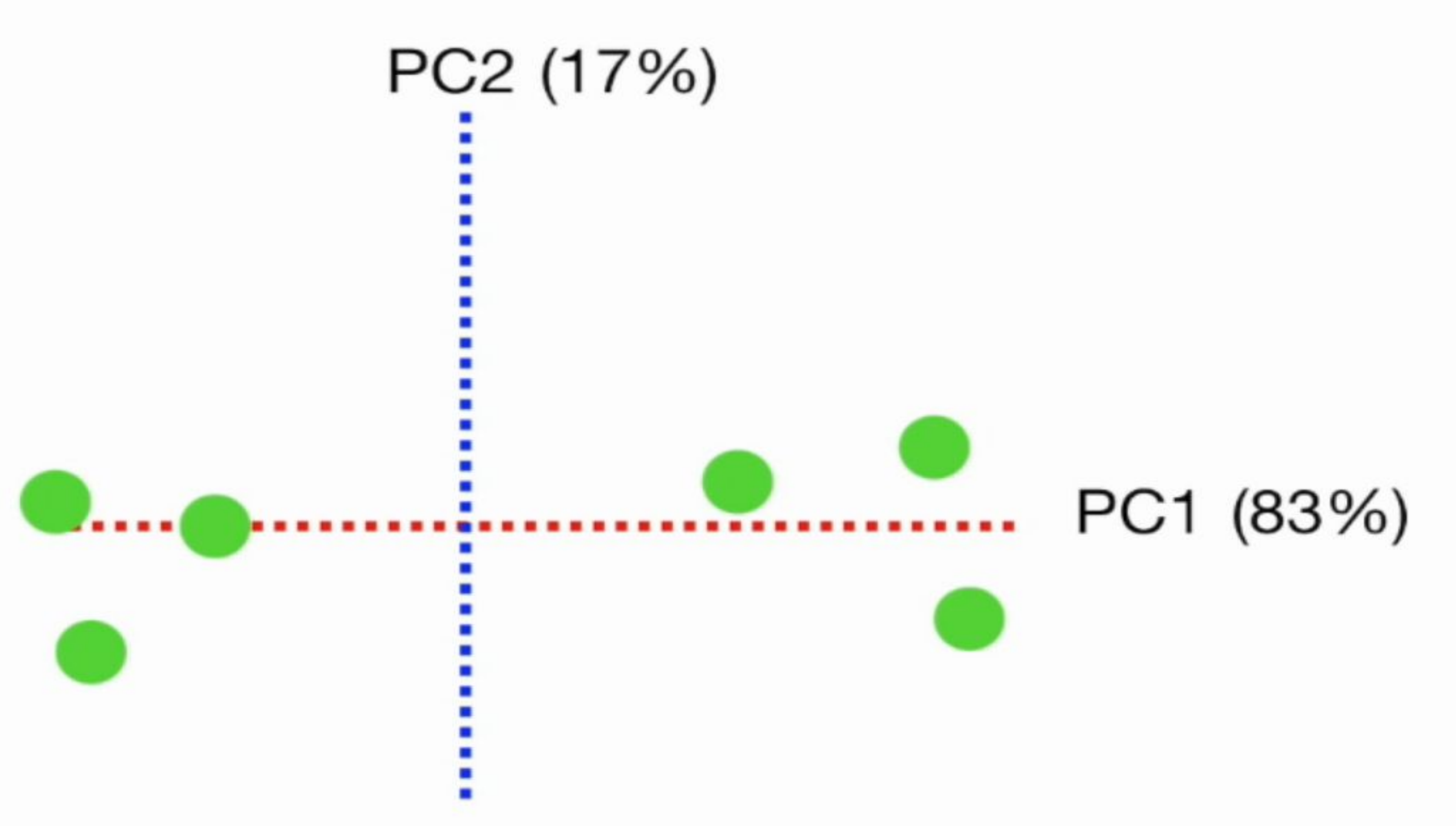
Compute Principal Components



Selected PC



Percentages of variation (%)s



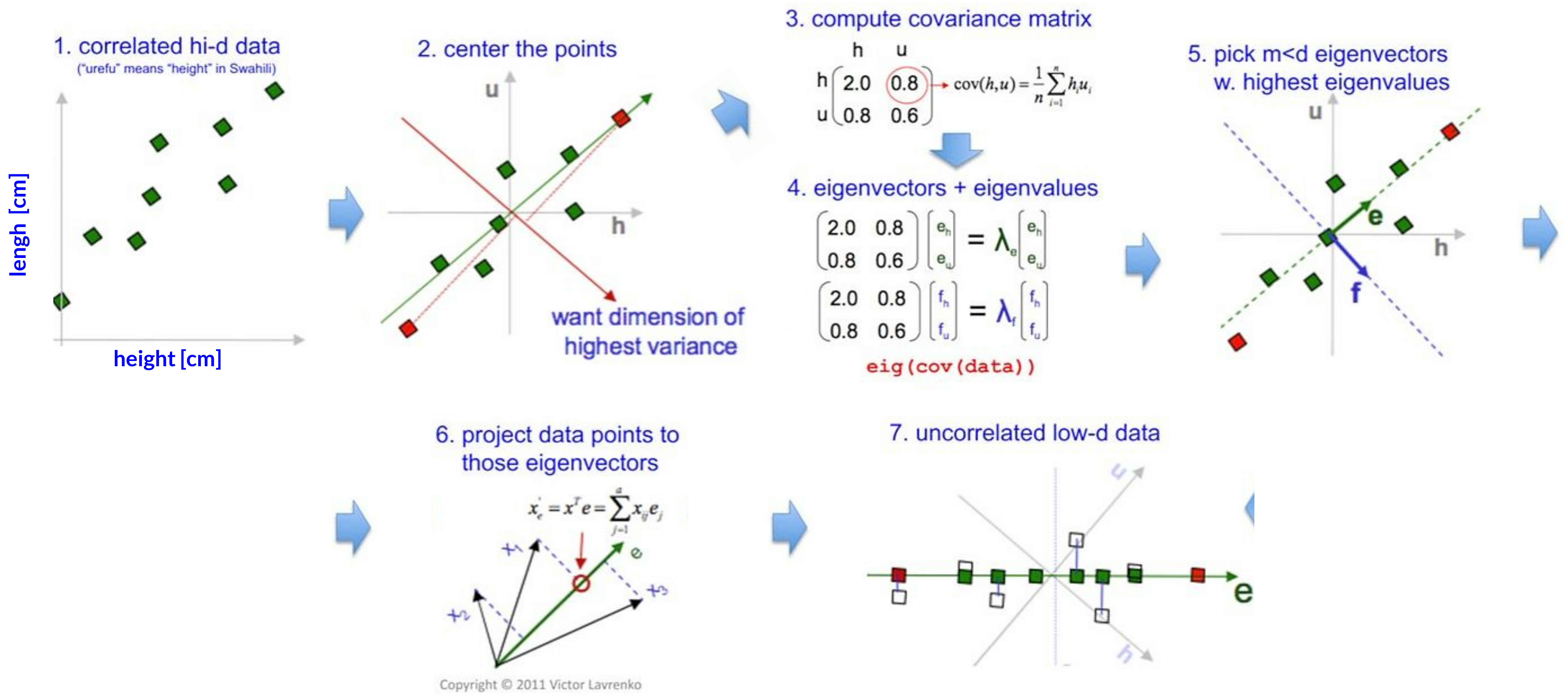
Images source: [StatQuest: PCA step by Step]

PCA step by step

Let’s practice with the step by step notebook: https://ja.cat/PCA_steps

PCA summary

Saturdays.AI
Barcelona



Source: [Dimensionality Reduction Using PCA]()

Sklearn PCA

```
from sklearn.decomposition import PCA
```

pca = PCA(…)

Arguments in PCA:

- **n_components** = number of components
- **svd_solver** = *# 'randomized'… (eigenvalues and eigenvectors)*
- **whiten** = True *#True or False → For pixels (0-255) and image processing*

pca.fit(data)

Attributes:

- pca.**explained_variance_ratio_** *# np.cumsum(pca.explained_variance_ratio_)*
- pca.**components_** *# coefficients of the linear transformation of the original data*
- pca.**n_components_** *# number of components*

data_pca = pca.transform(data) *# data coordinates using the principal components*