PARIS PANTHÉON-ASSAS UNIVERSITÉ

Data Visualization

ADDA71

Principal Component Analysis (PCA)

for data visualization

Ilyes Jenhani

About your Instructor ?

- Ilyes Jenhani : « Responsable de la Majeur SE »
- 20 years of experience in Academia and in different countries
- 20 years of research in Machine Learning, Uncertainty in AI and AI4SE
  - Publications: 30+ papers
  - Citations: 700+
  - https://scholar.google.fr/citations?user=BBkQqFQAAAAJ&hl=fr
- ilyes.jenhani@efrei.fr

Organization

- Instructors: Mano Joseph MATHEW & Ilyes Jenhani
- Course and interactions → English & French (Preferably English)
- Platforms: Teams & Moodle
- Evaluation:

Group project (up to xxx students) – Ilyes	**60%**
TPs - Mano	**40%**

Schedule (Tentative) (2)

#	Class type	Content
1	CTP	Principal Component Analysis (PCA)
2	CTP	Correspondence Analysis (CA)
3	CTP	Multiple Correspondence Analysis (MCA)
4	PRJ	Project

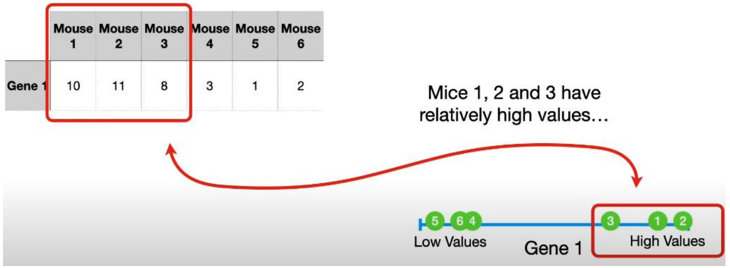
Introduction

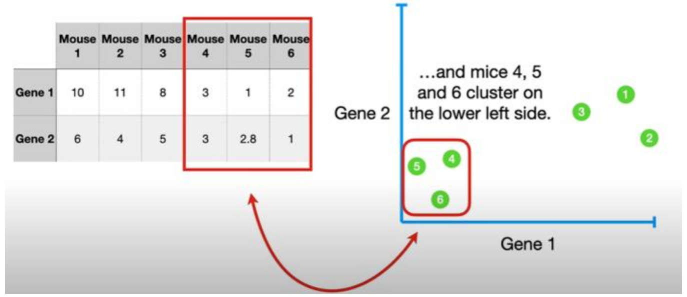
The Curse of Dimensionality

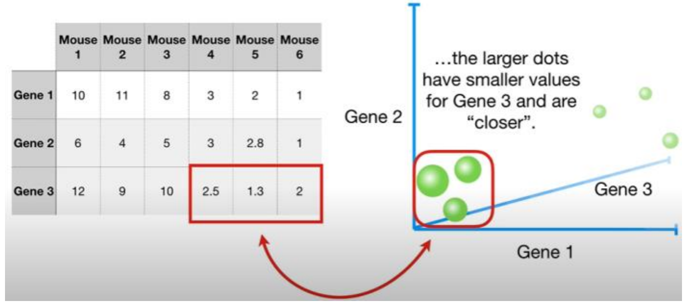
- In Machine Learning, the *Curse of Dimensionality* refers to the various challenges and complications that arise when analyzing data in high-dimensional spaces (often hundreds or thousands of *dimensions*).
- In the context of data analysis and machine learning, *dimensions* refer to the *features* or *attributes* of data.
  - For instance, if we consider a dataset of houses, the dimensions could include the house's price, size, number of bedrooms, location, and so on.
- Adding more features (or dimensions) increases the complexity of the dataset without necessarily increasing the amount of useful information.
- Causes many problems: data sparsity, increased computation, overfitting, *visualization challenges*, etc.

Dimensionality reduction

- In both Statistics and Machine Learning, the number of attributes/features/variables of a dataset is referred to as its **dimensionality**.







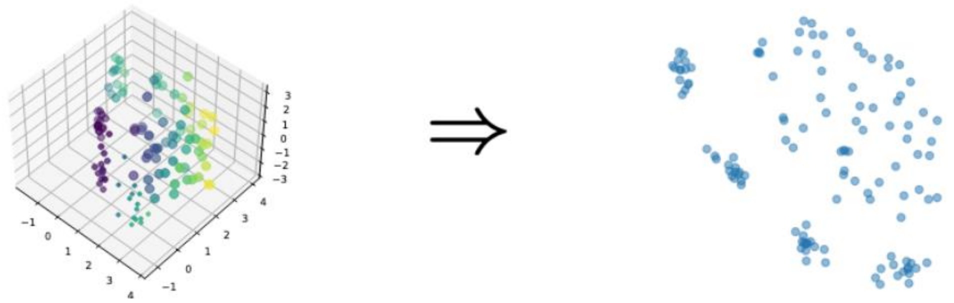
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1
Gene 3	12	9	10	2.5	1.3	2
Gene 4	5	7	6	2	4	7

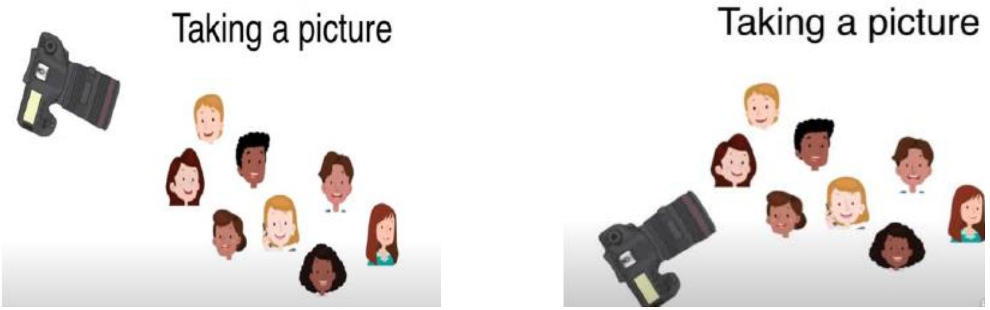
???

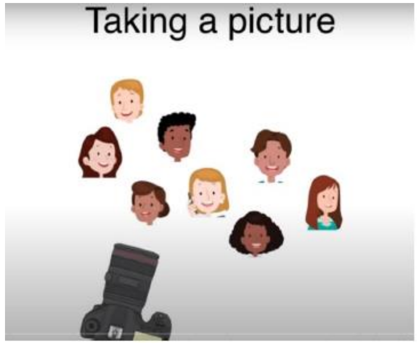
- Datasets with > 3 dimensions→ not easy to visualize !
- Solution : Dimensionality reduction.

Dimensionality reduction (2)

- Dimensionality reduction is the transformation of data from a high-dimensional space into a lower dimensional space so that the low-dimensional representation retains as much information as the original data.







Dimensionality reduction (3)

- There are several dimensionality reduction methods.

Feature Selection

Feature Extraction

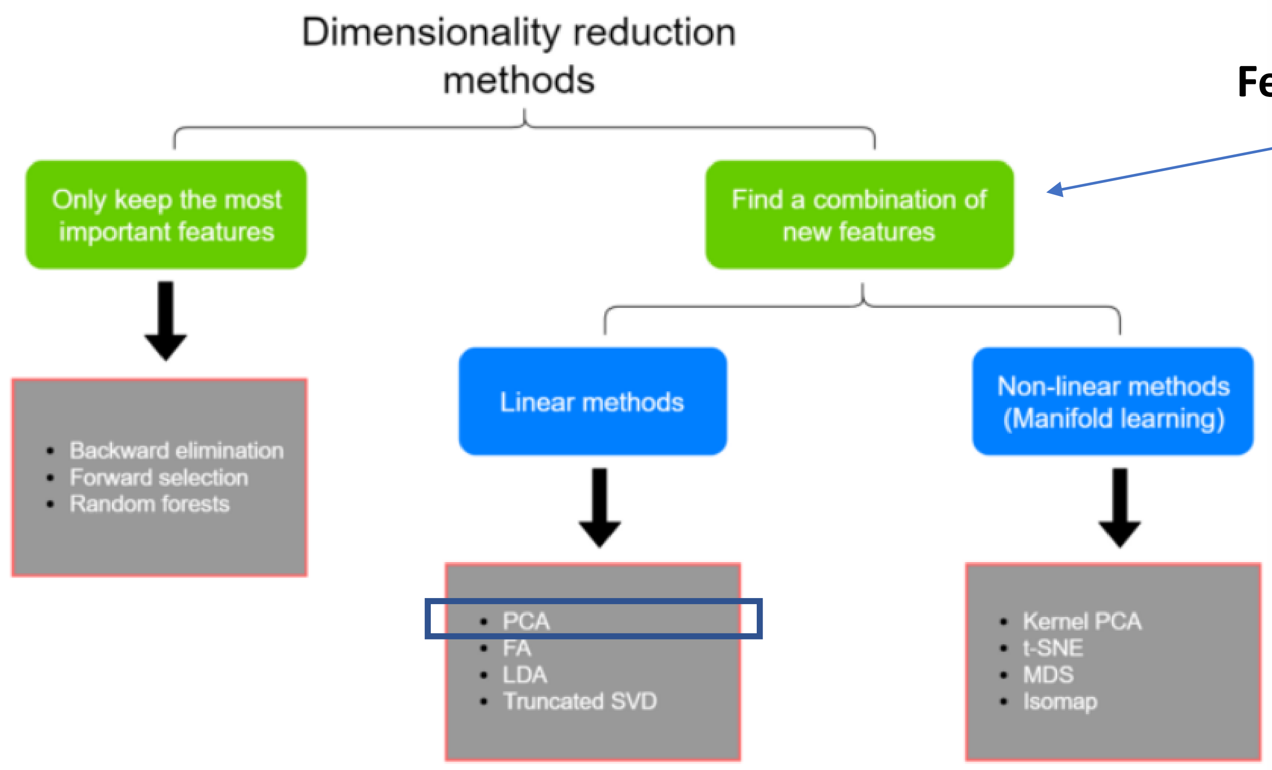
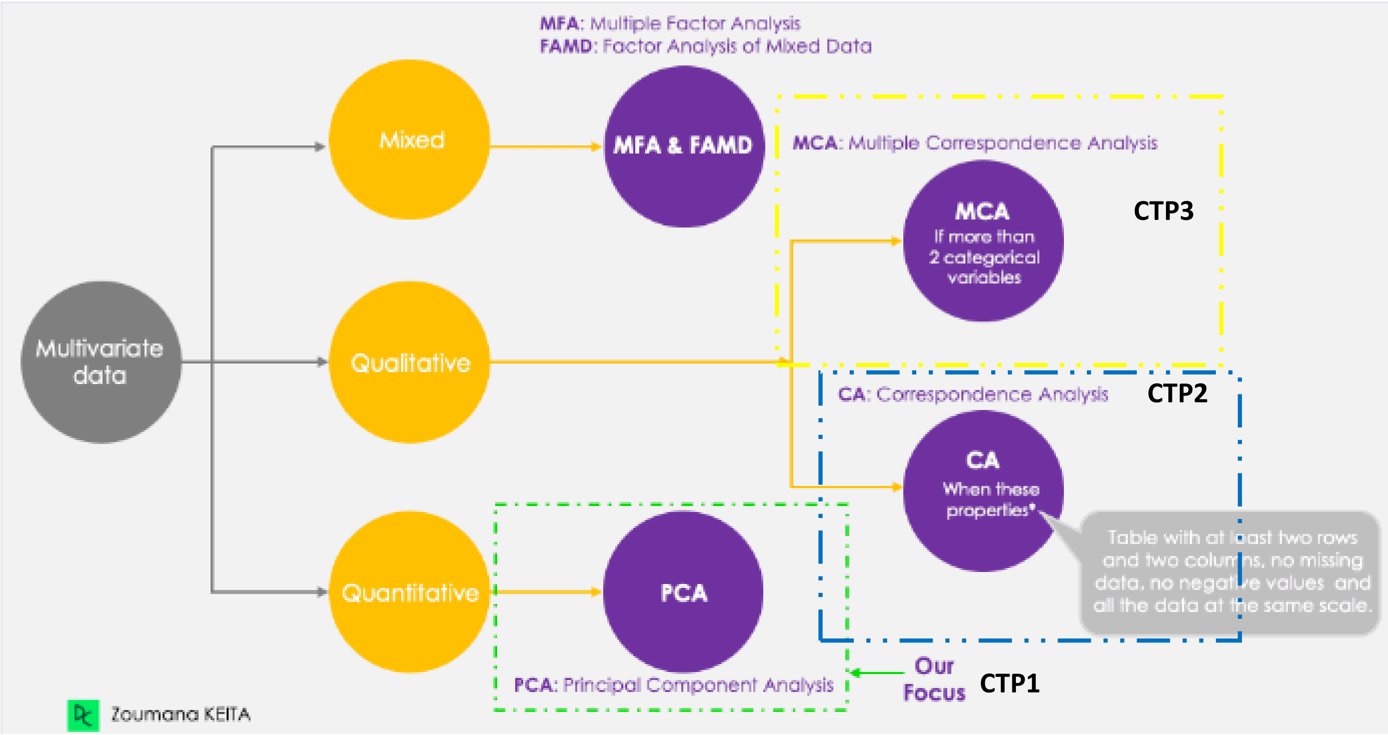


Image copyright: Rukshan Pramoditha

Dimensionality reduction methods

- Aim : summarize and visualize *multivariate* data

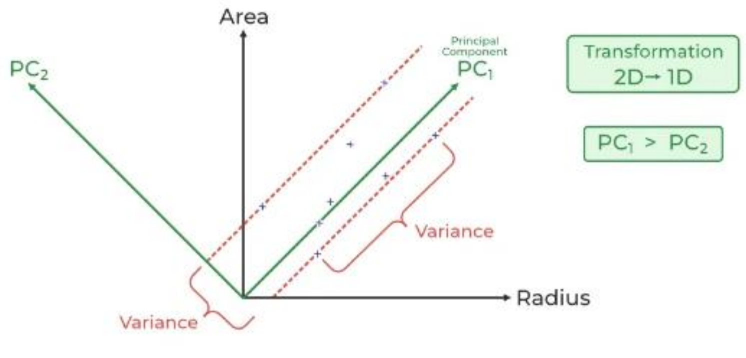


1.Principal Component Analysis (PCA)

What is PCA?

- Principal Component Analysis (PCA) is a *linear* dimensionality reduction technique widely used in data analysis, machine learning, and statistics.
- PCA identifies a set of orthogonal axes, called *principal components*, that capture the maximum variance in the data.
- The principal components are linear combinations of the original variables in the dataset and are ordered in decreasing order of importance.
- The total variance *captured* by ALL the principal components (cumulative inertia) is equal to the total variance in the original dataset (total inertia).

Eigenvalue = quantifies the explained variance of a principal component



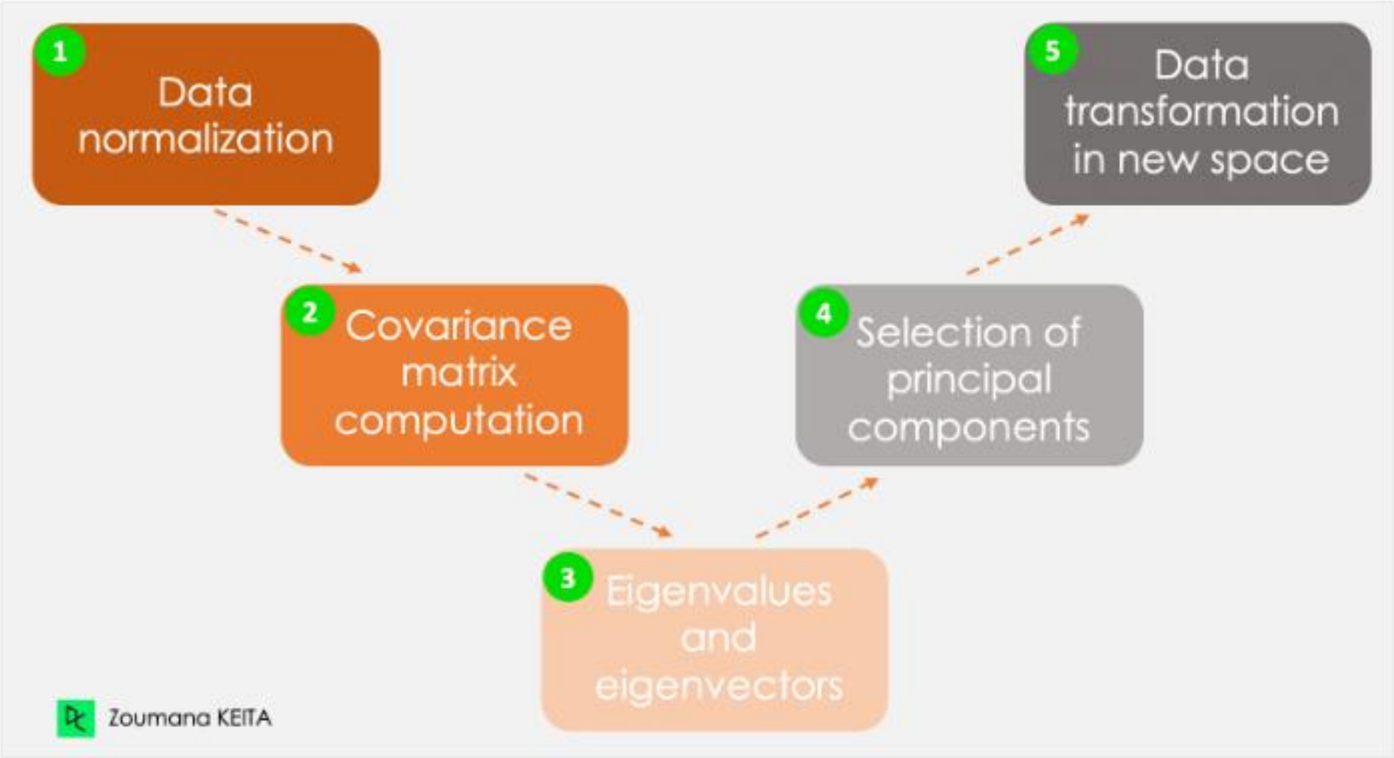
Principal Component Analysis

$$\text{Cumulative Inertia for the first } k \text{ components} = \sum_{i=1}^{k} \lambda_i$$

$$\text{Total Inertia} = \sum_{i=1}^{p} \lambda_i$$

where λ_i are the eigenvalues of the covariance matrix.

Steps of PCA



The five main steps for computing principal components

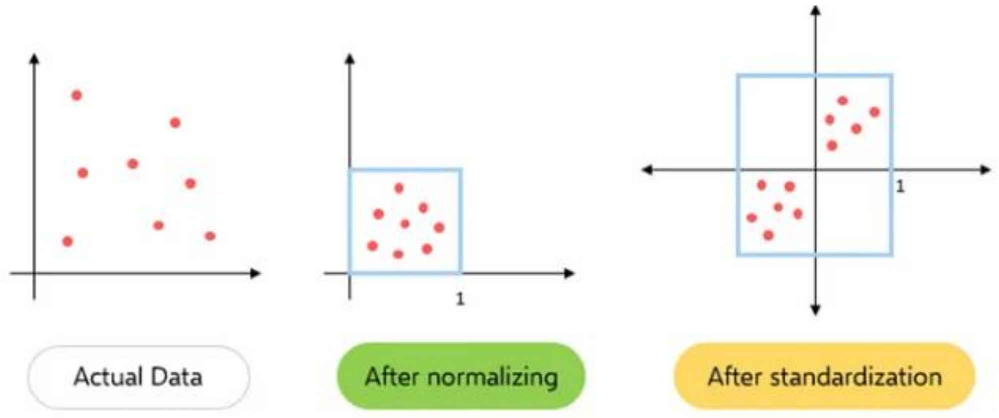
The mathematical details are covered in the ADIF72 : Mathematics for Data Scientists module.

Step 1: Data Standardization

- Let’s consider, for instance, the following information for a given client.
  - Monthly expenses: $300
  - Age: 27
  - Rating: 4.5
- This information has different scales and performing PCA using such data will lead to a biased result.
- This is where data standardization comes in.
- It ensures that each attribute has the same level of contribution, preventing one variable from dominating others.
- For each variable, standardization is done by subtracting its mean and dividing by its standard deviation

  Rescaling the data so that it has a mean = 0 and a standard deviation = 1.

  This process centers the data and ensures that all features contribute equally to the analysis.



Normalization

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Standardization

$$X' = \frac{\text{X} - \text{Mean}}{\text{Standard deviation}}$$

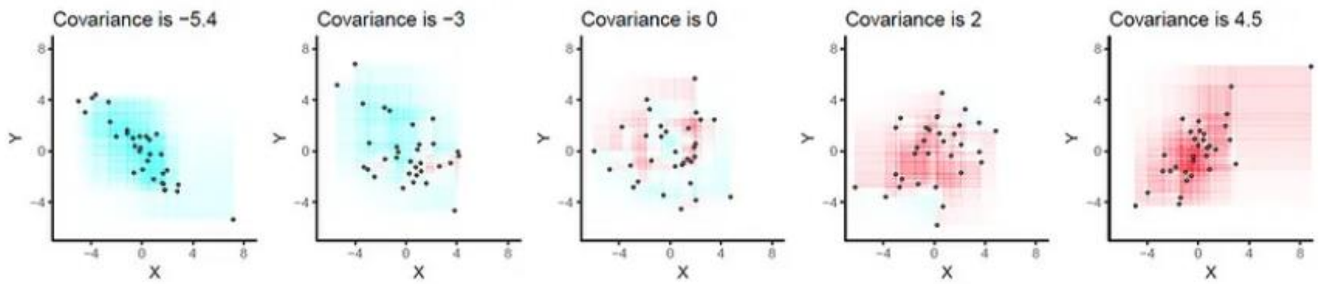
Step 2: Covariance matrix

- This step is about computing the **covariance matrix** from the standardized data.
- This is a symmetric matrix, and each element (i , j) corresponds to the covariance between variables (features) X_i and X_j.
- Covariance can be:
  - Positive: X_i and X_j increase (or decrease) together.
  - Negative: X_i decreases, X_j increases (or vice versa)
  - Null: no direct relationship.

$$cov(x1, x2) = \frac{\sum_{i=1}^{n} (x1_i - \bar{x1})(x2_i - \bar{x2})}{n-1}$$

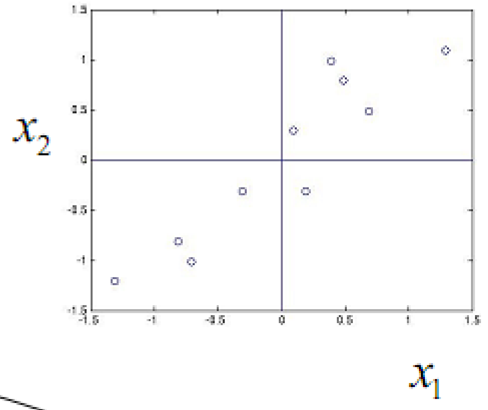
$$\begin{array}{cc} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} x \\ y \end{array} & \begin{bmatrix} var(x) & cov(x, y) \\ cov(x, y) & var(y) \end{bmatrix} \end{array}$$

$$\begin{array}{cc} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & \begin{bmatrix} var(x) & cov(x, y) & cov(x, z) \\ cov(x, y) & var(y) & cov(y, z) \\ cov(x, z) & cov(y, z) & var(z) \end{bmatrix} \end{array}$$



Step 3: Eigenvectors and Eigenvalues of the Covariance matrix

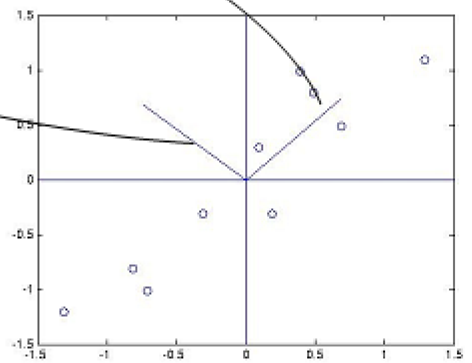
- The covariance matrix has eigenvectors



covariance matrix $C = \begin{bmatrix} .617 & .615 \\ .615 & .717 \end{bmatrix}$

eigenvectors $v_1 = \begin{bmatrix} -.735 \\ .678 \end{bmatrix}$ $v_2 = \begin{bmatrix} .678 \\ .735 \end{bmatrix}$

eigenvalues $\mu_1 = 0.049$ $\mu_2 = 1.284$



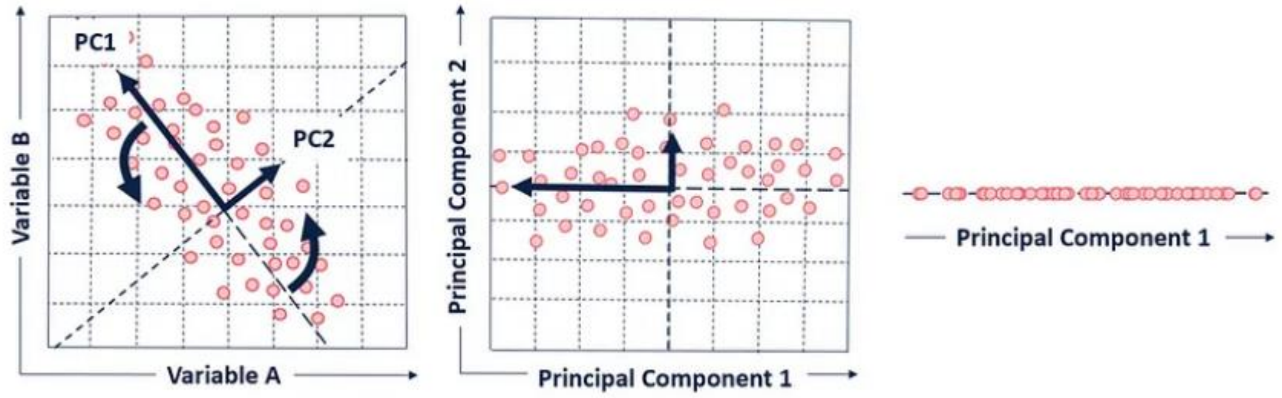
- Eigenvectors with larger eigenvalues correspond to directions in which the data varies more
- Finding the eigenvectors and eigenvalues of the covariance matrix for a set of data is called principal component analysis

Step 4: Selection of principal components

- There are as many pairs (eigenvector, eigenvalue) as the number of variables (features) in the data.
- Not all pairs are relevant.
- The eigenvector with the highest eigenvalue corresponds to the 1st PC (principal component). The 2nd PC is the eigenvector with the second highest eigenvalue, and so on.
- Different selection criteria:
  - Scree plot curve (Elbow)
  - Kaiser rule: pick PCs with eigenvalues of at least 1.
  - Proportion of variance plot: The selected PCs should be able to describe at least 80% of the variance

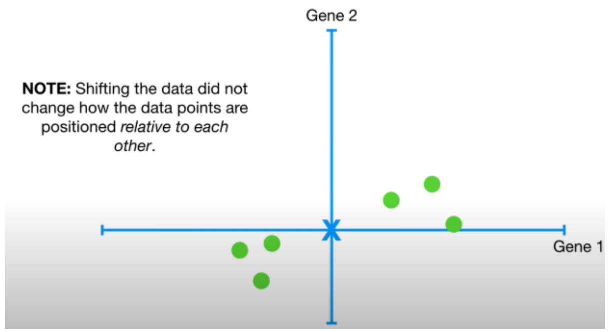
Step 5 - Data transformation in new dimensional space

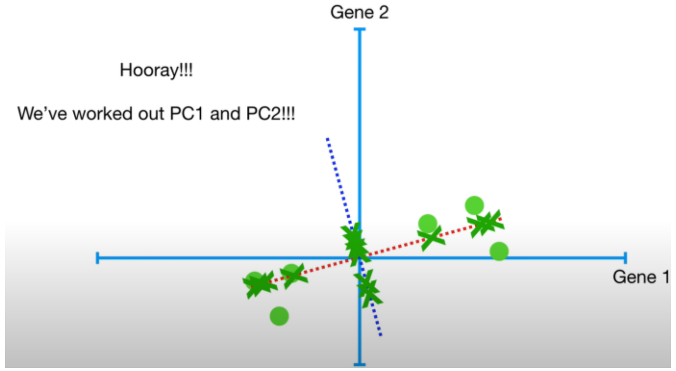
- This step involves re-orienting the original data onto a new subspace defined by the selected PCs.
- This reorientation is done by multiplying the original data by the previously computed eigenvectors.
- The principal components are linear combinations of the original variables (features), and they represent new axes along which the data can be projected.



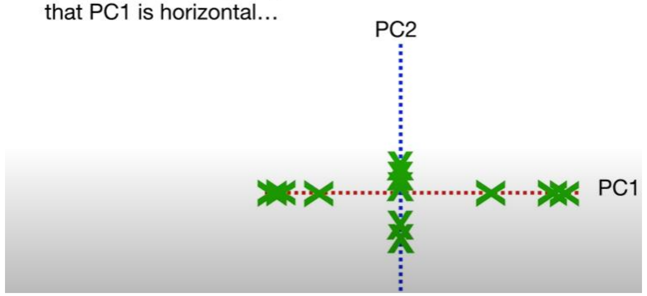
Summary

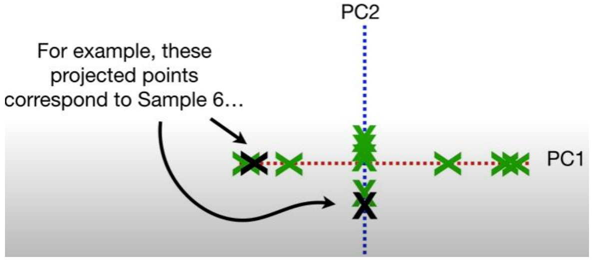
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	1	2
Gene 2	6	4	5	3	2.8	1

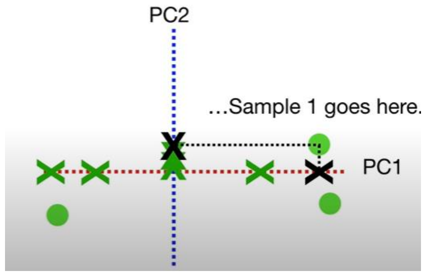




We simply rotate everything so that PC1 is horizontal…

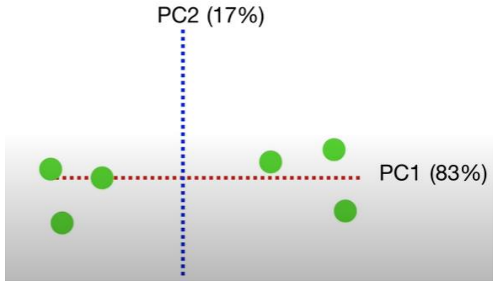




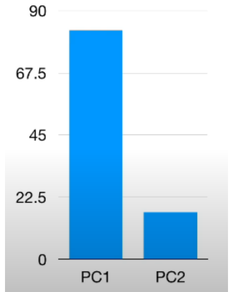


…

PC2 accounts for 17% of the variance

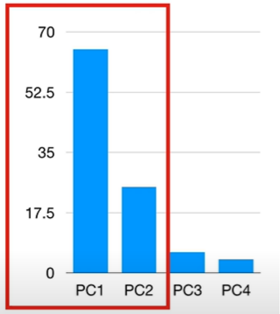


…then we use the projected points to find where the samples go in the PCA plot.



Scree plot

If we do the same for the 4 features



…in this case, PC1 and PC2 account for 90% of the variation, so we can just use those to draw a 2-dimensional PCA graph.

Interpretation of PCA results

- Principal components are the new axes or directions that maximize the variance in the dataset.
  - PC1 typically captures the most significant underlying patterns and trends in the data.

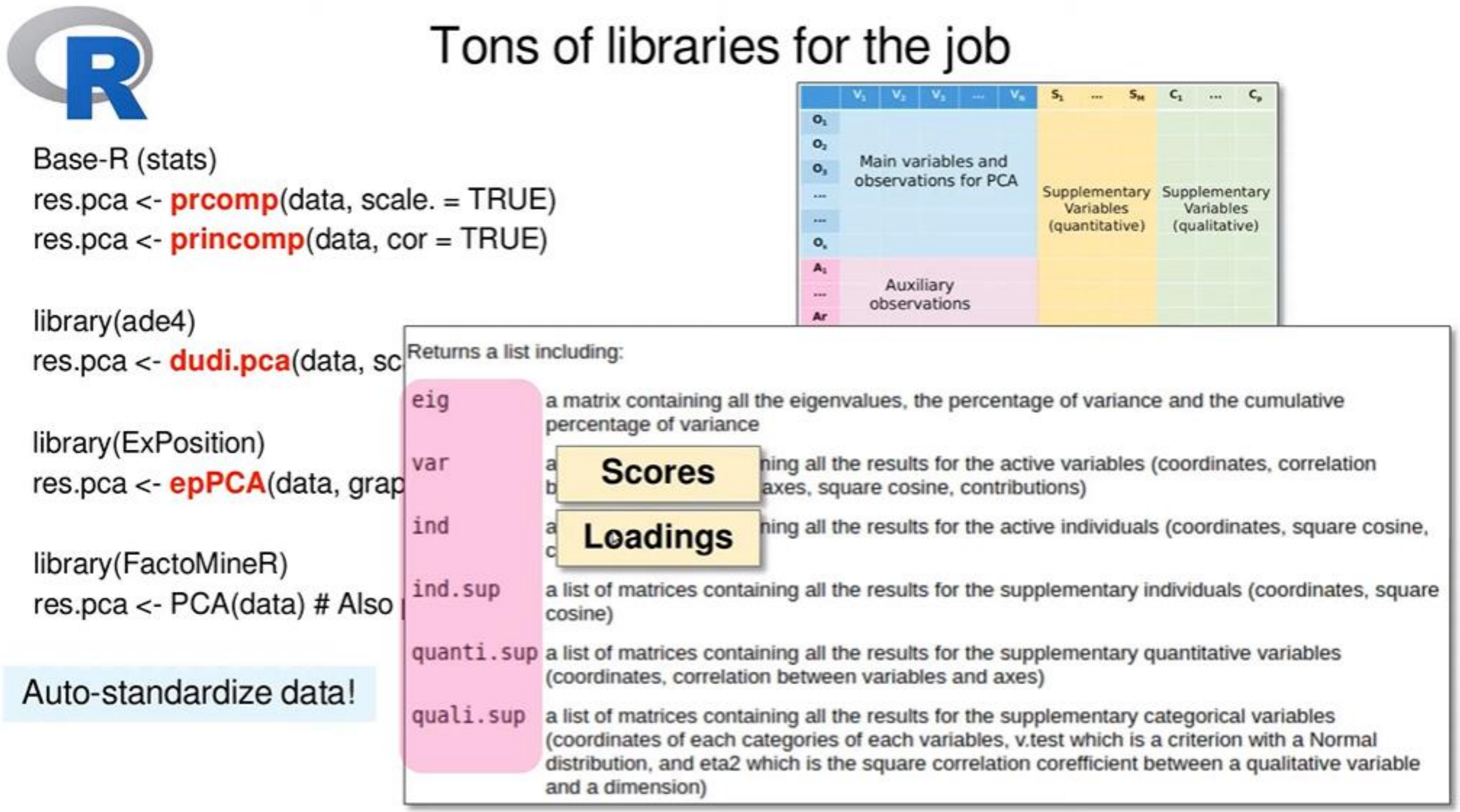
1. Explained variance: How to visualize and interpret eigenvalues ? Cumulative explained variance ?

2. Loading: How to visualize and analyze the contribution of each feature (variable) to each PC? i.e. how the original variable does correlate with the PC (positive, negative)?

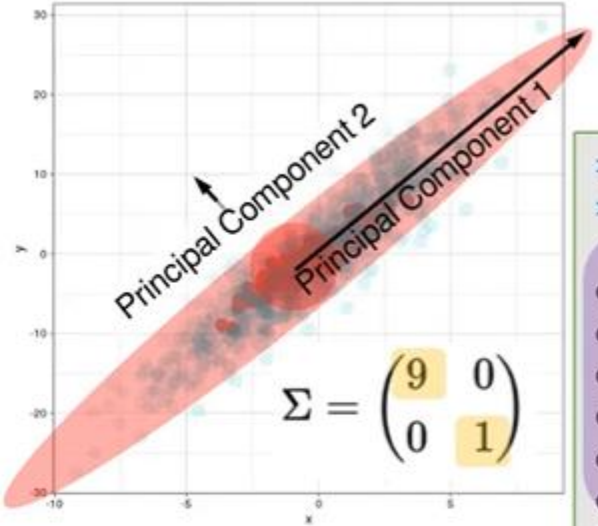
3. Scores: How to visualize and analyze the coordinates of the original data points (individuals) in the new reduced principal component space?

4. How to visualize and interpret both loadings of the variables and scores of the individuals using Biplots?

Tons of libraries (R and Python)



Explained variance - eigenvalues



Other criterion is limit cumulative variance to >70% (80%, 90%, ...)
In this example, the same "cutoff"

```
> res.pca <- PCA(data, scale.unit = TRUE, quali.sup = c(1,2,15), ncp = 5)
> res.pca$eig
        eigenvalue percentage of variance cumulative percentage of variance
comp 1  4.699109e+00           2.764181e+01                          27.64181
comp 2  2.492367e+00           1.466098e+01                          42.30280
comp 3  1.825143e+00           1.073613e+01                          53.03893
comp 4  1.419917e+00           8.352455e+00                          61.39139
comp 5  1.204090e+00           7.082883e+00                          68.47427
comp 6  1.118311e+00           6.578298e+00                          75.05257
comp 7  8.354839e-01           4.914611e+00                          79.96718
comp 8  7.301688e-01           4.295110e+00                          84.26229
comp 9  6.570788e-01           3.865169e+00                          88.12746
comp 10 5.745309e-01           3.379593e+00                          91.50705
comp 11 4.571703e-01           2.689237e+00                          94.19629
comp 12 3.734672e-01           2.196866e+00                          96.39315
comp 13 2.961649e-01           1.742147e+00                          98.13530
comp 14 1.571032e-01           9.241366e-01                          99.05944
comp 15 9.196537e-02           5.409728e-01                          99.60041
comp 16 6.793007e-02           3.995887e-01                         100.00000
comp 17 1.473409e-07           8.667111e-07                         100.00000
```

eigenvalue>1
indicates that PCs account for more variance than accounted by one of the original variables
(criterion for "cutoff")

Explained variance - eigenvalues

Loadings: contribution of each var to each PC

```
> head(res.pca$var$coord)
                                      Dim.1      Dim.2       Dim.3        Dim.4     Dim.5
Population                       0.02241625 -0.0374425  0.76590160  0.434057805 0.2096857
Area (sq. mi.)                   0.03952854  0.2205003  0.63573963  0.562567298 0.2313934
Pop. Density (per sq. mi.)       0.24163728  0.1077903 -0.21420881  0.003841362 0.2706660
Coastline (coast/area ratio)     0.11934469 -0.3834396 -0.40935817  0.363530409 0.2301244
Net migration                    0.15190368  0.5934147  0.12748704 -0.364475889 0.3185615
Infant mortality (per 1000 births) -0.92159072  0.1445784  0.04632528 -0.111024580 0.1380521
```

var: variables (not variance)

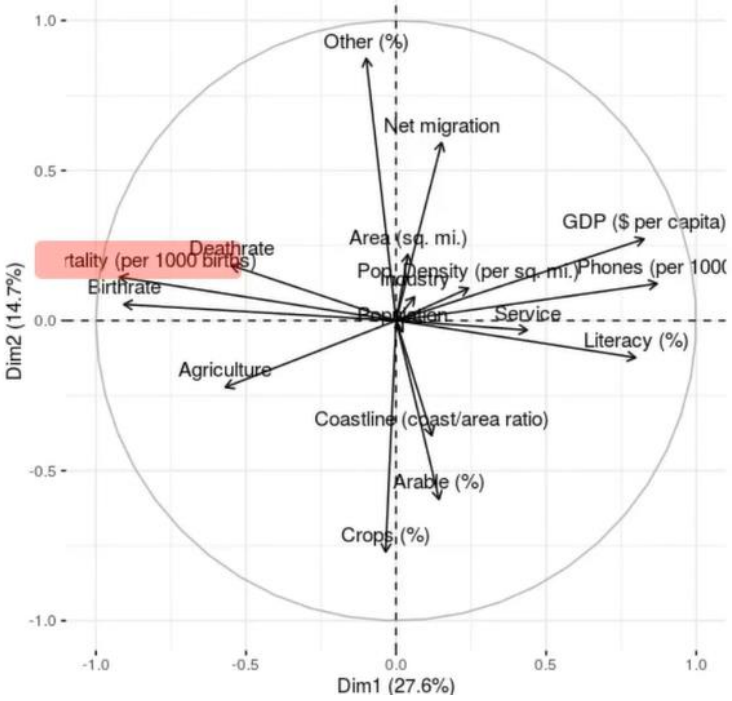
coord: coordinates (loadings)

Dim.$_i$: PC$_i$

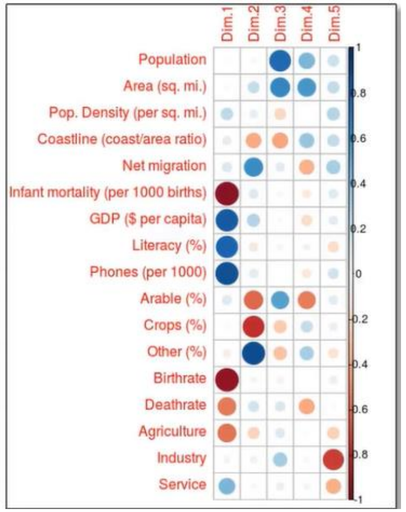
Contribution of “Population” to PC$_3$ is 76.59%

Correlation circle

```
> fviz_pca_var(res.pca)
```



Correlogram



The sign is arbitrary: doesn’t mean its bad !

Loadings: contribution of each var to each PC (2)

```
> head(res.pca$var$coord)
                                    Dim.1      Dim.2       Dim.3        Dim.4     Dim.5
Population                     0.02241625 -0.0374425  0.76590160  0.434057805 0.2096857
Area (sq. mi.)                 0.03952854  0.2205003  0.63573963  0.562567298 0.2313934
Pop. Density (per sq. mi.)     0.24163728  0.1077903 -0.21420881  0.003841362 0.2706660
Coastline (coast/area ratio)   0.11934469 -0.3834396 -0.40935817  0.363530409 0.2301244
Net migration                  0.15190368  0.5934147  0.12748704 -0.364475889 0.3185615
Infant mortality (per 1000 births) -0.92159072  0.1445784  0.04632528 -0.111024580 0.1380521
```

var: variables (not variance)

coord: coordinates (loadings)

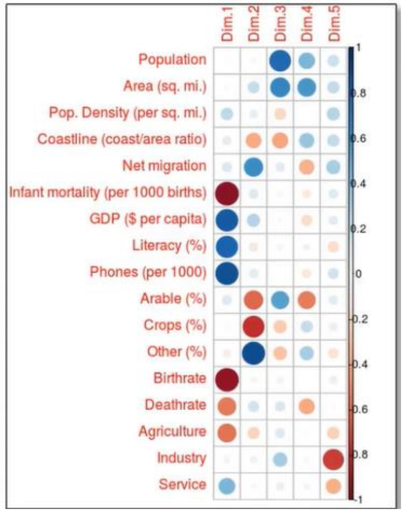
Dim.$_i$: PC$_i$

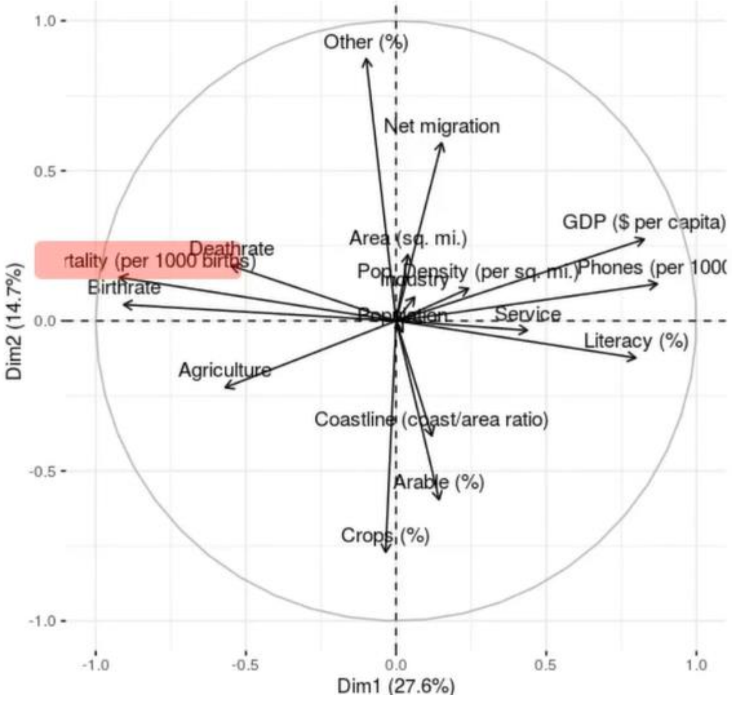
Contribution of "Population" to PC$_3$ is 76.59%

Correlation circle

```
> fviz_pca_var(res.pca)
```

Correlogram





The sign is arbitrary: doesn't mean its bad !

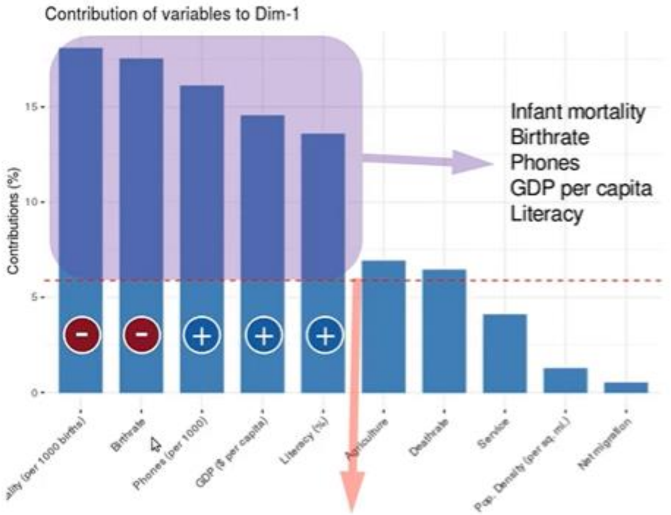
Loadings: contribution of each var to each PC (3)

```
> head(res.pca$var$coord)
                                    Dim.1      Dim.2       Dim.3        Dim.4     Dim.5
Population                     0.02241625 -0.0374425  0.76590160  0.434057805 0.2096857
Area (sq. mi.)                 0.03952854  0.2205003  0.63573963  0.562567298 0.2313934
Pop. Density (per sq. mi.)     0.24163728  0.1077903 -0.21420881  0.003841362 0.2706660
Coastline (coast/area ratio)   0.11934469 -0.3834396 -0.40935817  0.363530409 0.2301244
Net migration                  0.15190368  0.5934147  0.12748704 -0.364475889 0.3185615
Infant mortality (per 1000 births) -0.92159072  0.1445784  0.04632528 -0.111024580 0.1380521
```

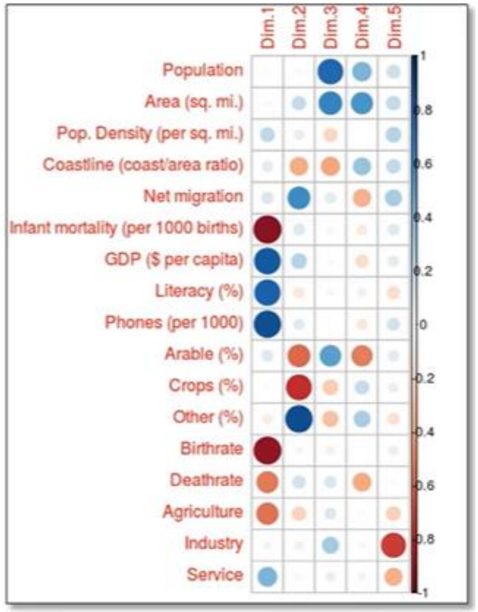
```
fviz_contrib(res.pca,choice = 'var',top=10)
fviz_contrib(res.pca,choice = 'var',top=10,axes = 2)
```

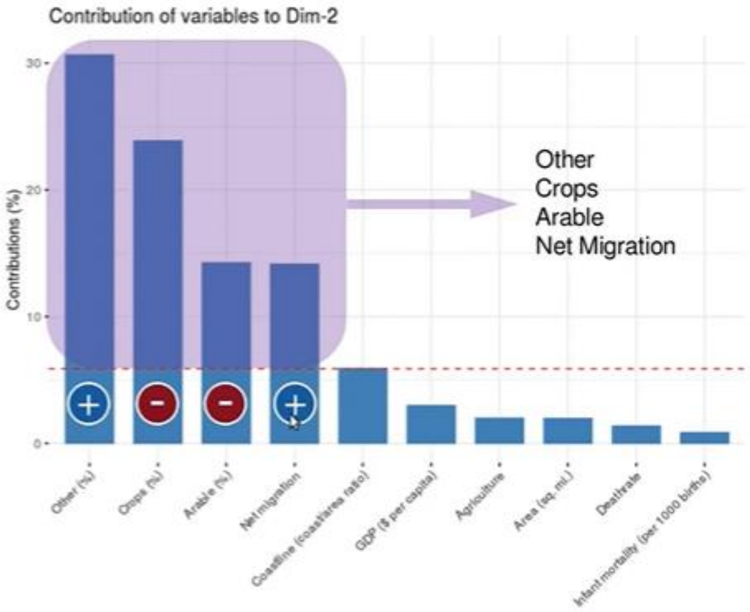
Top 10 var. Only PC_1

PC_1 and PC_2



1/Number of variables
(assuming uniform distribution)

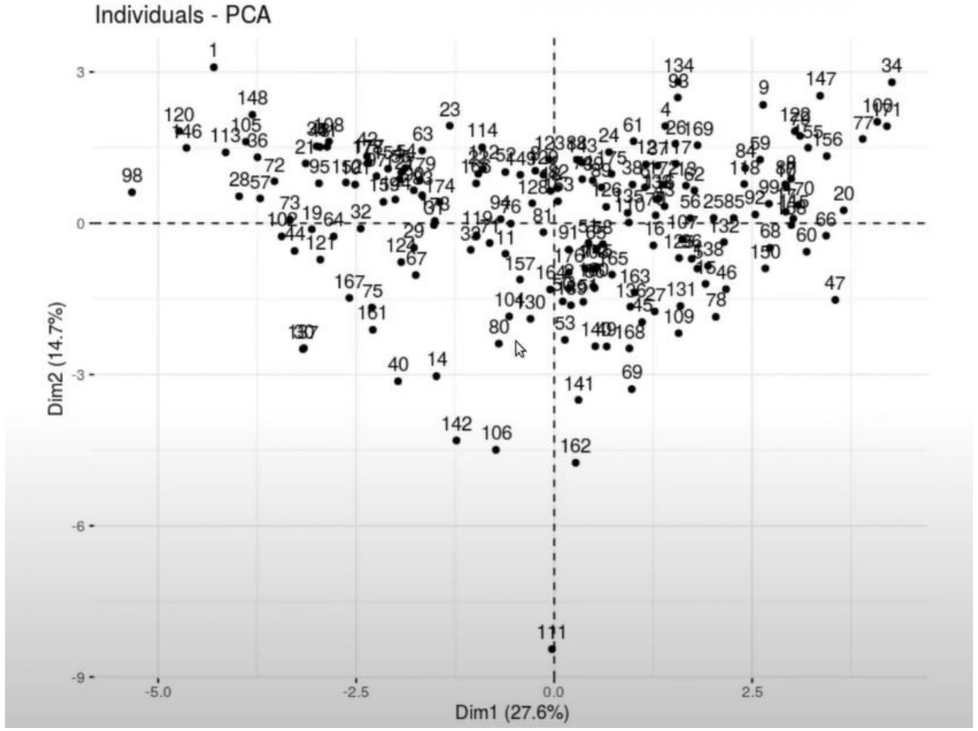




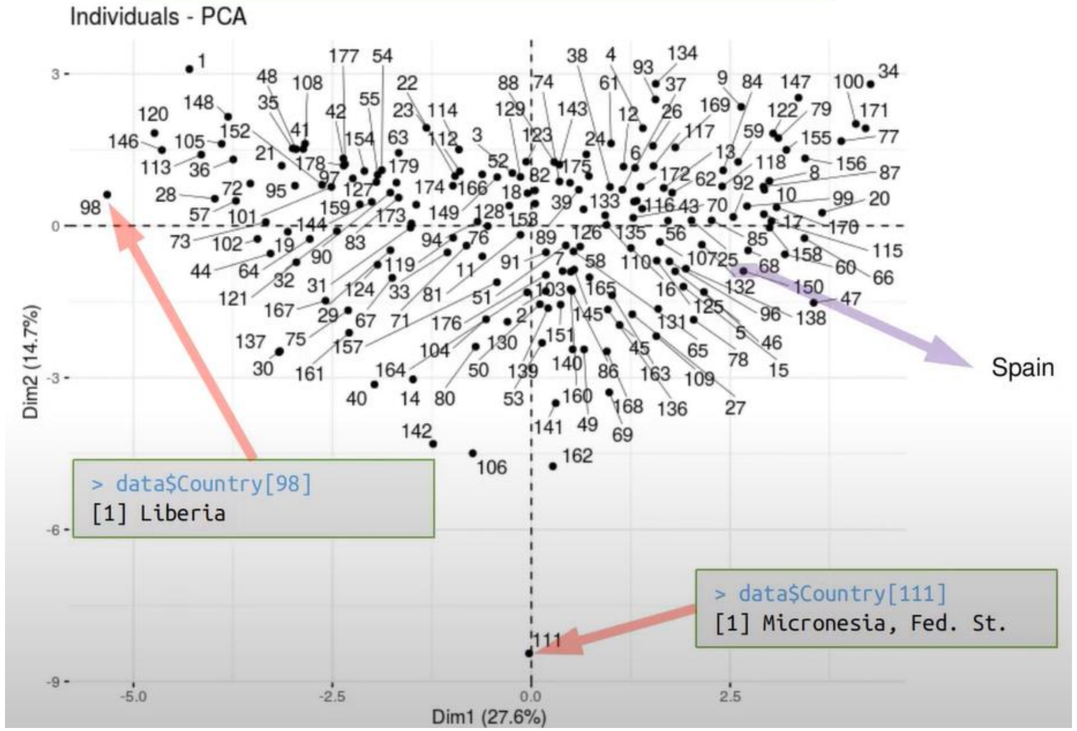
Scores: projection on the new space

Score plots individuals

```
fviz_pca_ind(res.pca,repel=FALSE)
```




```
fviz_pca_ind(res.pca,repel=FALSE)
fviz_pca_ind(res.pca,repel=TRUE)
```

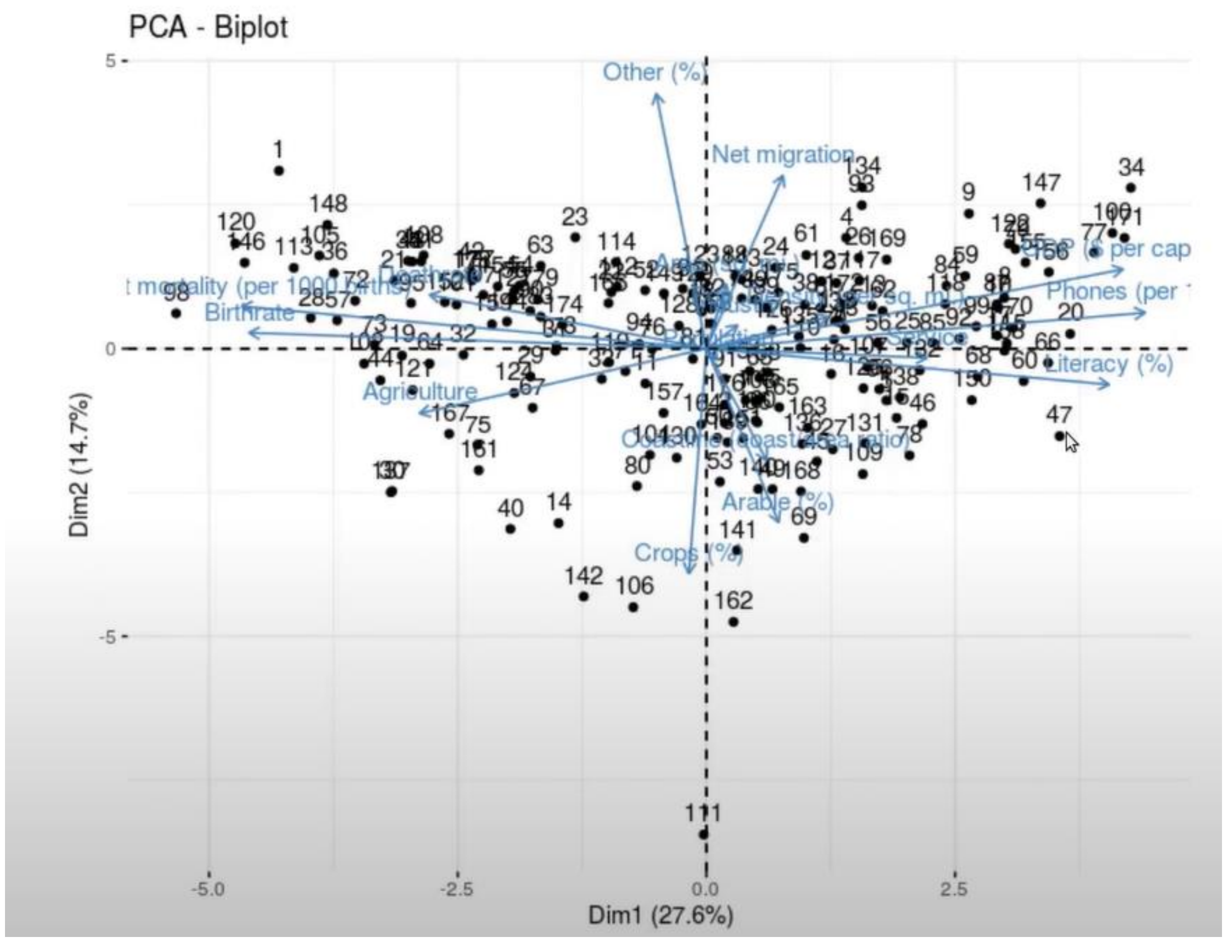


- Interpretation:
  - **Data clustering**: You can use scores to visualize and analyze clusters of data points in the reduced dimension space. For example, plotting data in the PC1 vs. PC2 space can reveal groups or clusters.
  - **Outliers**: Data points that lie far from others in the principal component space might be outliers.

Scores & loadings on the same graph

Biplots

```
fviz_pca_biplot(res.pca)
```



Prince

- Prince is a Python library for multivariate exploratory data analysis in Python.
- It includes a variety of methods for summarizing tabular data, including principal component analysis and correspondence analysis.
- Prince provides efficient implementations, using a scikit-learn API.
- Prince uses Altair for making charts.
- Prince GitHub
- Let’s start our Lab !

Useful links

- https://www.youtube.com/watch?v=kDbyBcDcC3I
- Prince python library: https://maxhalford.github.io/prince/