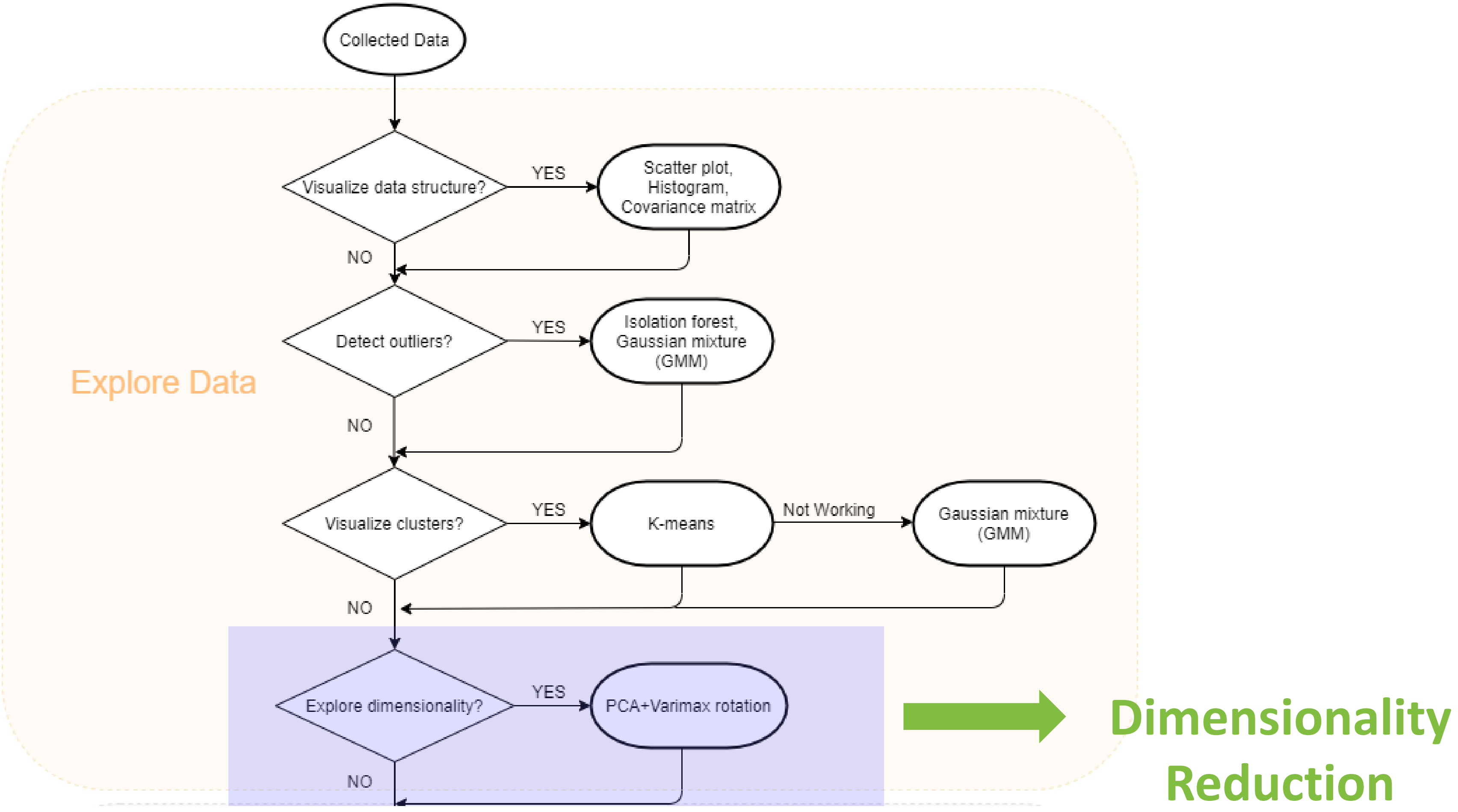
Dimensionality Reduction (PCA + Varimax Rotation)

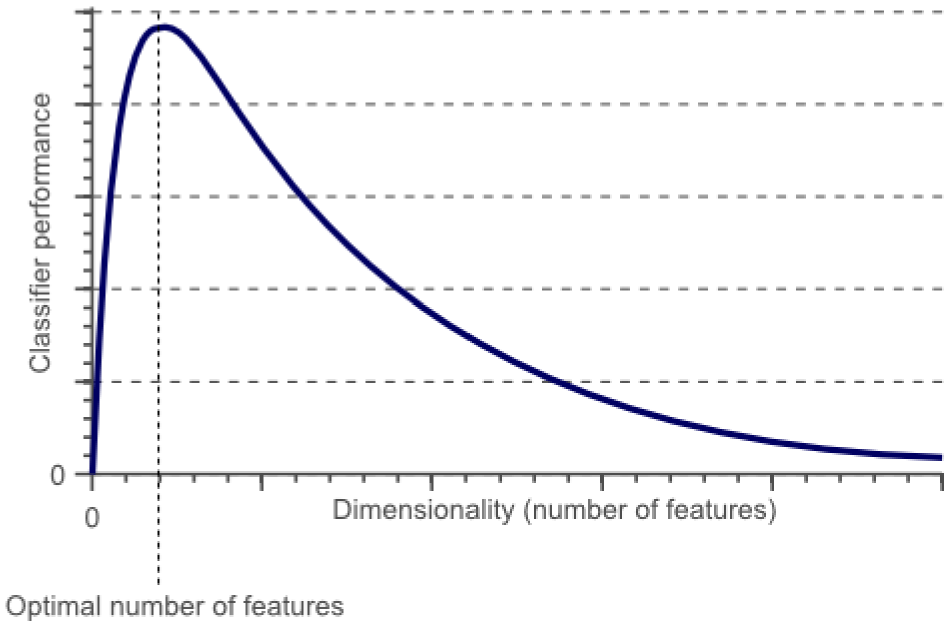
Xue Zong

Dimensionality Reduction



Motivation

The Curse of Dimensionality



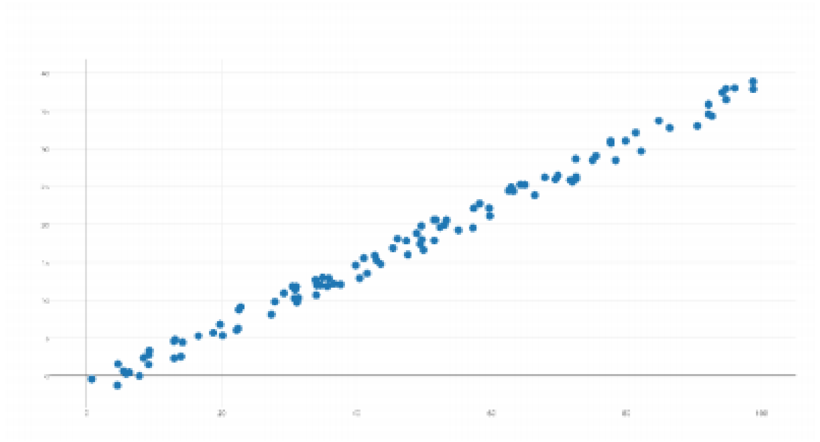
Dimensionality Reduction

- Data visualization
  Visualize data in 2D or 3D space
- Speed up ML algorithm
  Reduce computing time
- Compress data
  Save storage space.
- Remove redundant features and noise

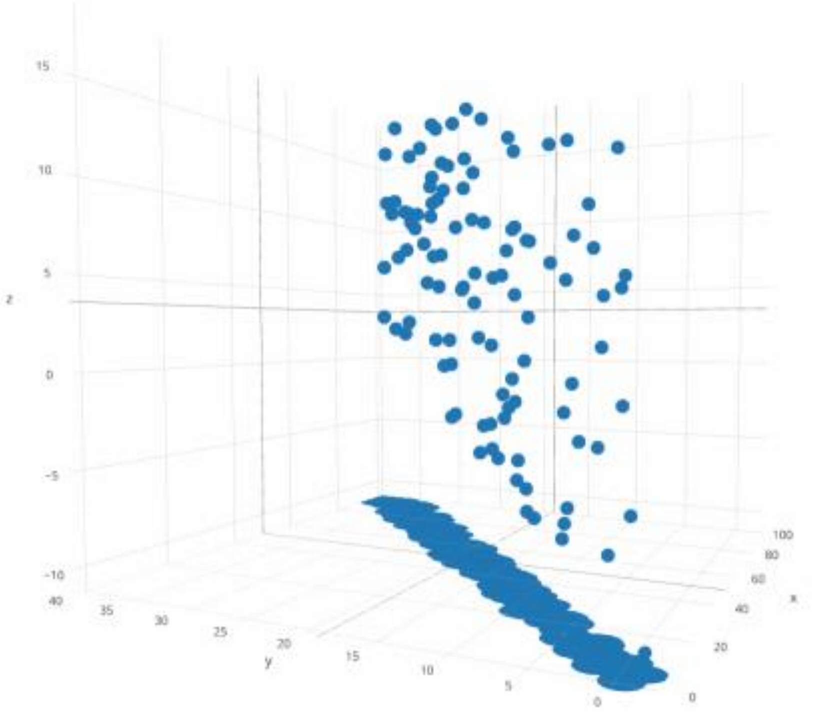
https://opendatascience.com/confronting-the-curse-of-dimensionality/

Motivation

2D → 1D

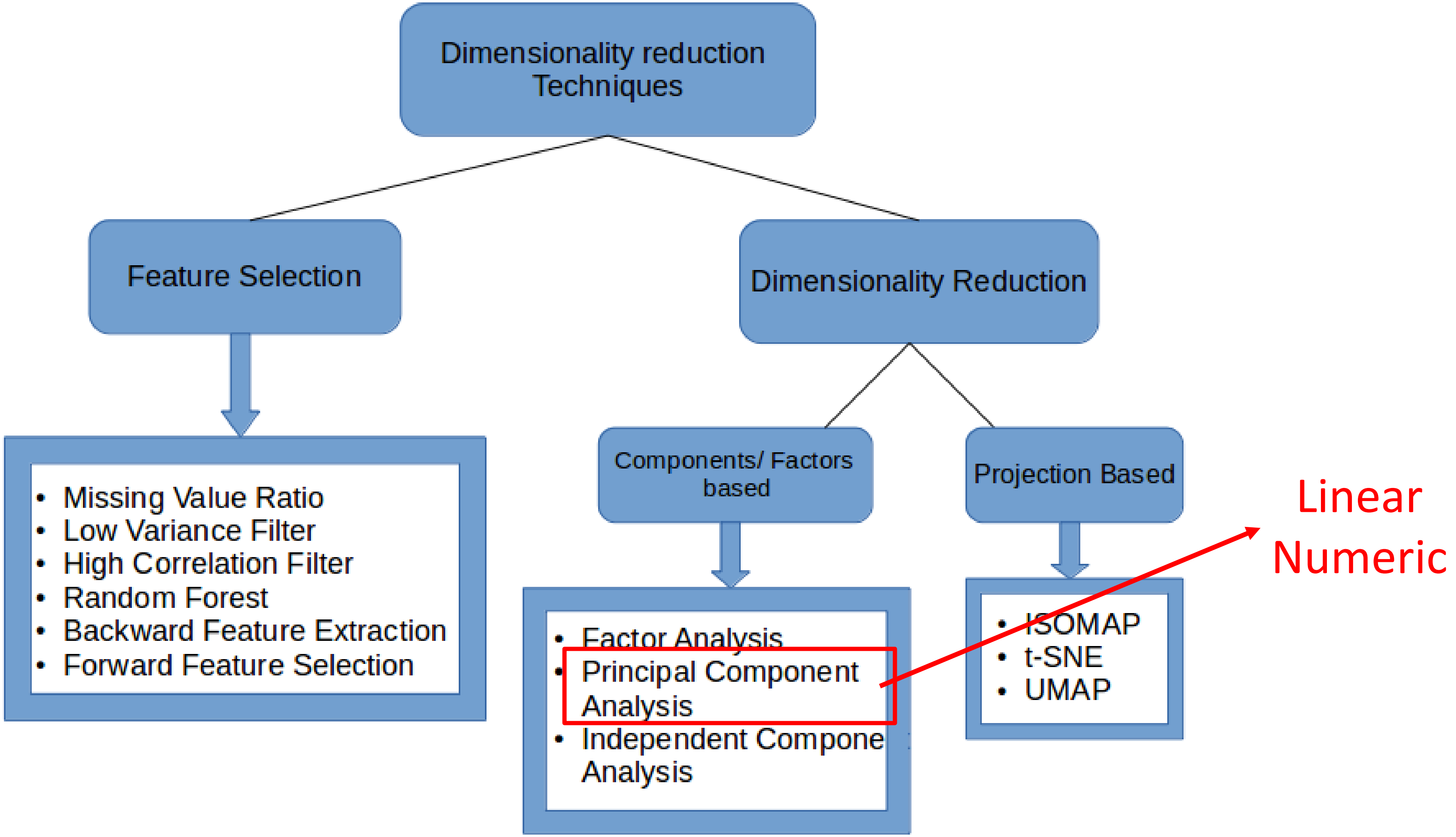


3D → 1D



Intrinsic dimensionality: 1

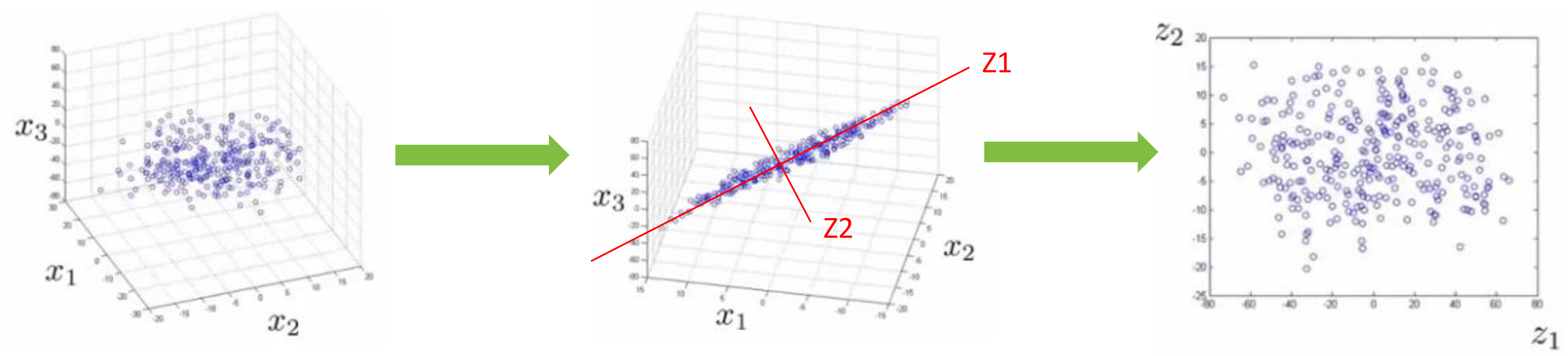
Common Methods to Reduce Dimensionality



https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/

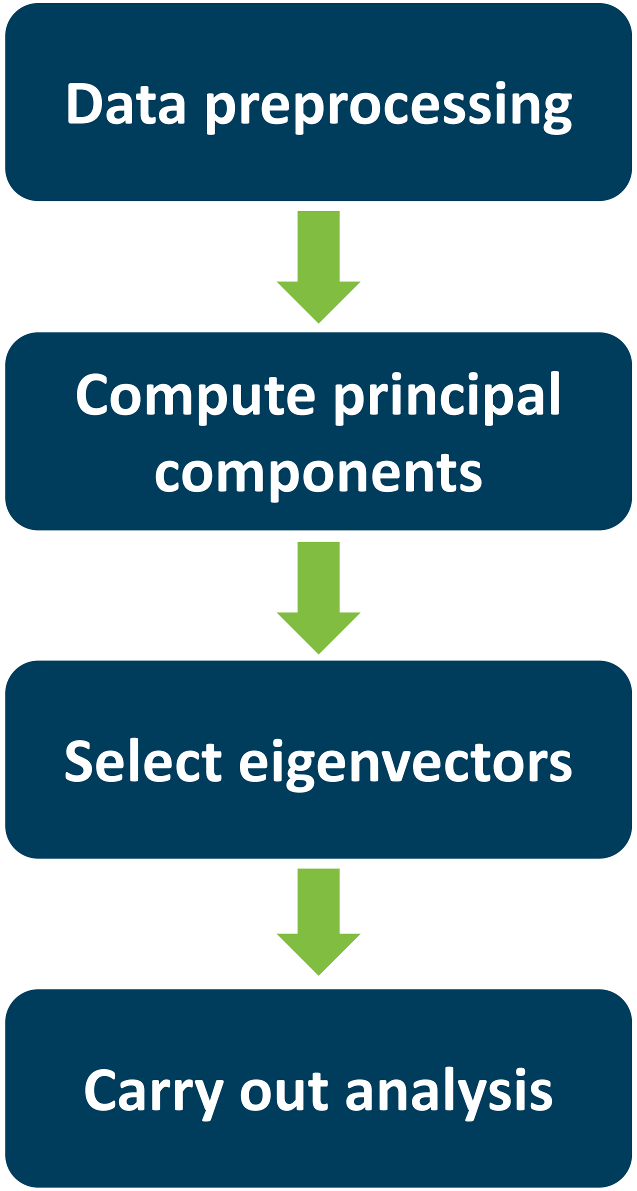
Principal Components Analysis (PCA)

- PCA is a technique for transforming a dataset consisting of a large number of possibly correlated variables from the original coordinate system into a new and more informative one [1]. It identifies the linear combinations of input variables that explain the **most variance.**
- Unsupervised dimensionality reduction technique and only suitable for numeric linear data.



[1] Ogunnaike, Babatunde A. *Random phenomena: fundamentals of probability and statistics for engineers*. CRC Press, 2011.

PCA Algorithm



- Data preprocessing: standardization

- Compute principal components:
  Compute covariance matrix
  Compute eigenvectors and eigenvalues of covariance matrix

- Carry out analysis
  Visualization/Clustering

Standardization

Data preprocessing

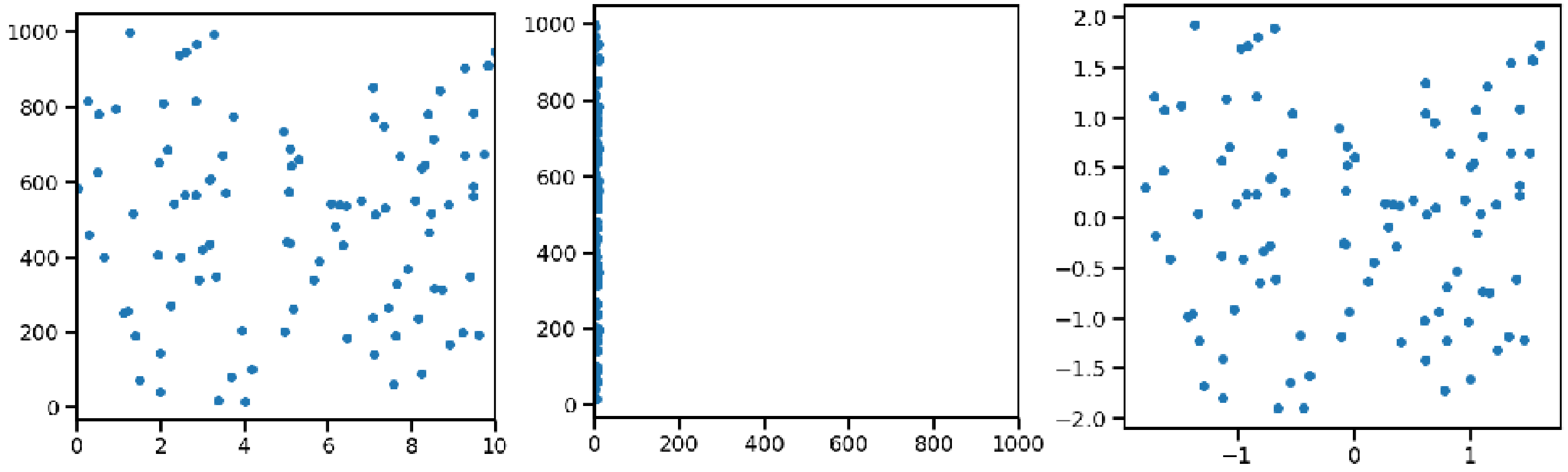
Compute principal components

Select eigenvectors

Carry out analysis

- Scale the data to remove units (technically optional, but nearly always necessary). Standardize the data.

- Mean-center and normalize the original data matrix



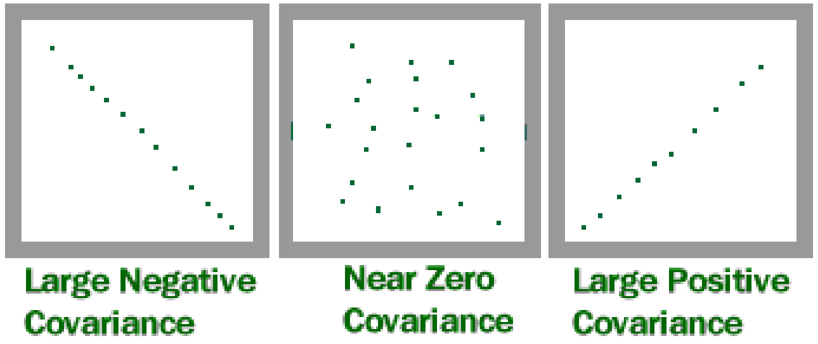
Compute Principal Components

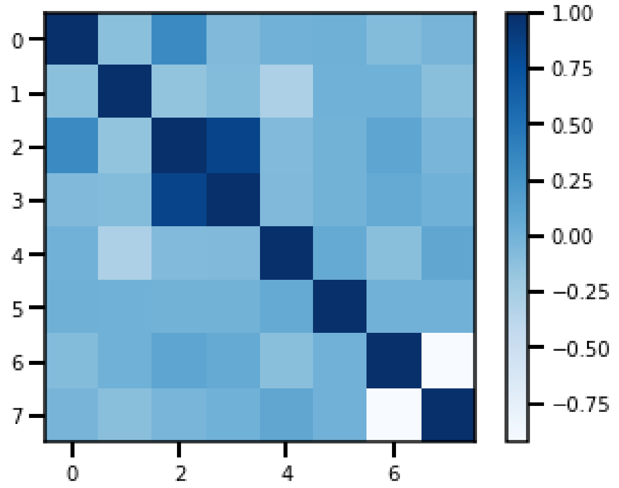
- Data preprocessing
- Compute principal components
- Select eigenvectors
- Carry out analysis

- Each element in the **covariance matrix** R represents the covariance between two features.

$$R = X^T X$$

- Eigenvalues and eigenvectors are calculated via **singular value decomposition (SVD).**

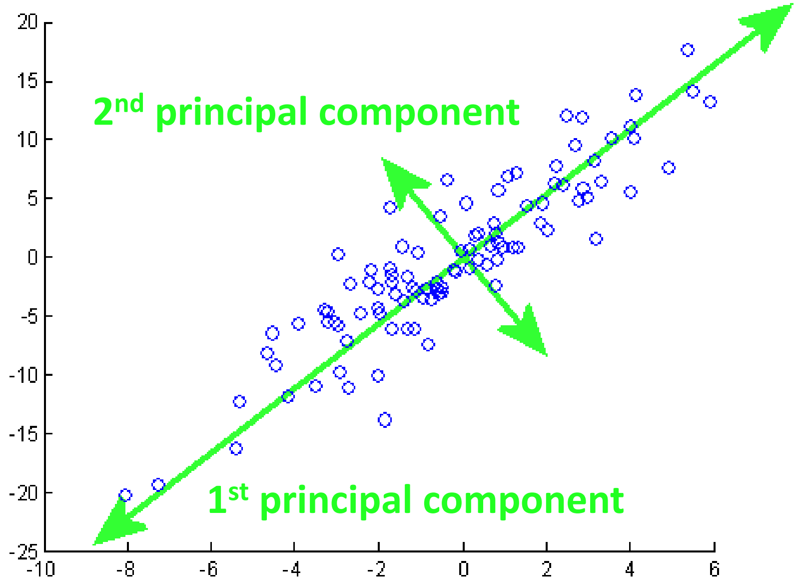




Compute Principal Components

- Data preprocessing
- Compute principal components
- Select eigenvectors
- Carry out analysis

- **Principal components** (loading vectors): eigenvectors of covariance matrix
- **Principal component scores**: project the data matrix onto the loading vectors



Scree Plot (Choose number of PCs)

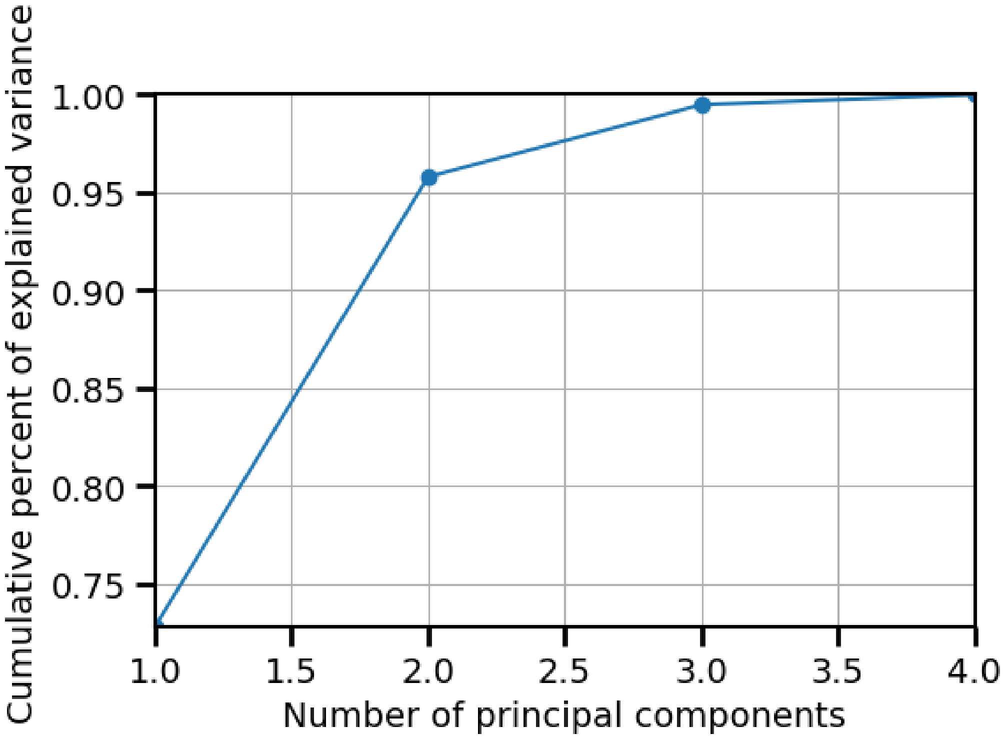
Data preprocessing

Compute principal components

Select eigenvectors

Carry out analysis

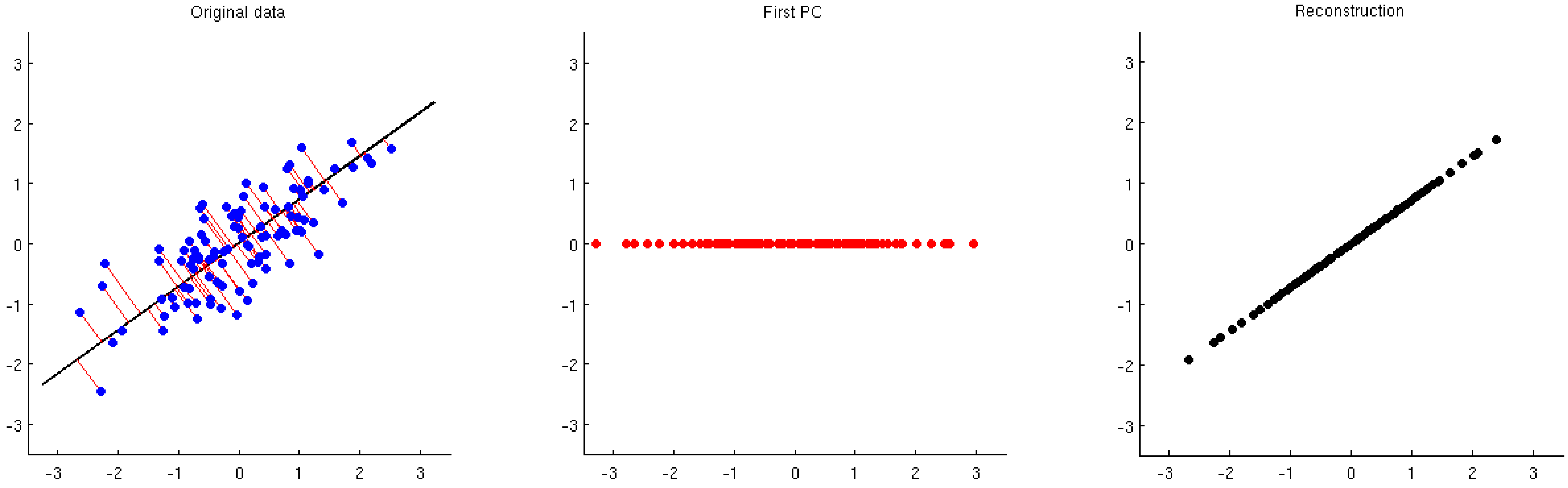
- Sort eigenvalues in descending order and choose the k eigenvectors that correspond to the k largest eigenvalues where k is the number of dimensions of the new feature subspace.
- Cumulative explained variance $\Gamma_{\mathrm{j}} = \frac{\sum_{i=1}^{j} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$



Threshold: 90% or 95%

With two principal components, the variance in the dataset can be explained by 95%.

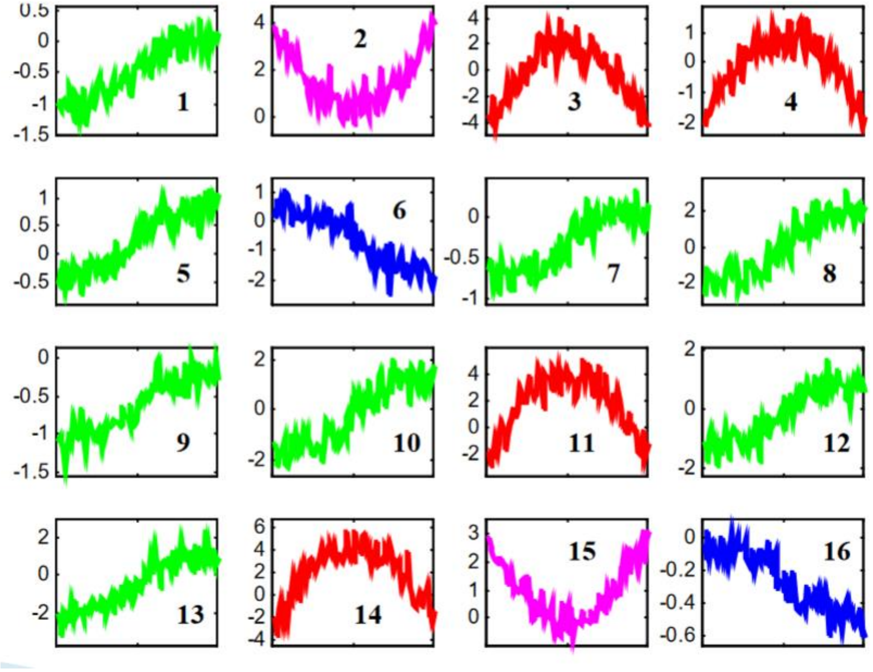
Reconstruction from Principal Components



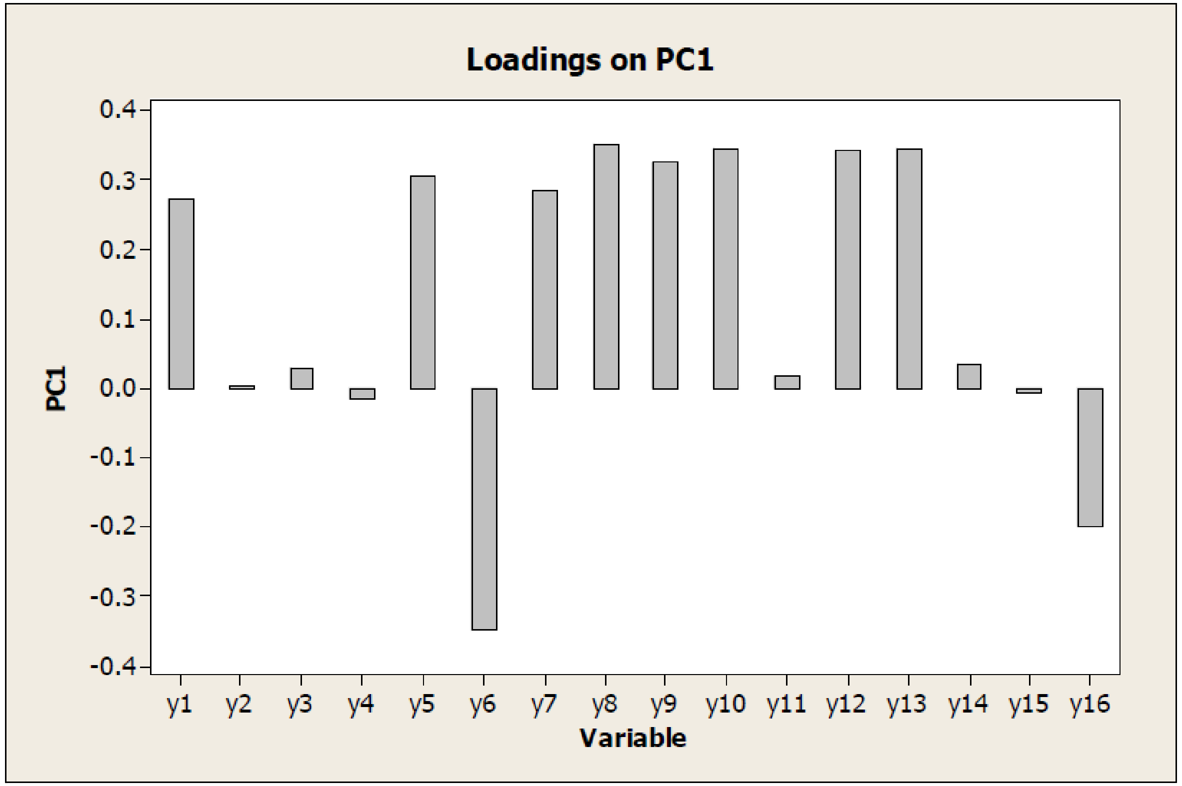
- Principal components scores
  $$Z = XV \ (n \times k \ matrix)$$
- Reconstruction
  $$\hat{X} = ZV^T \ (n \times p \ matrix)$$

Loading Plots

Original dataset



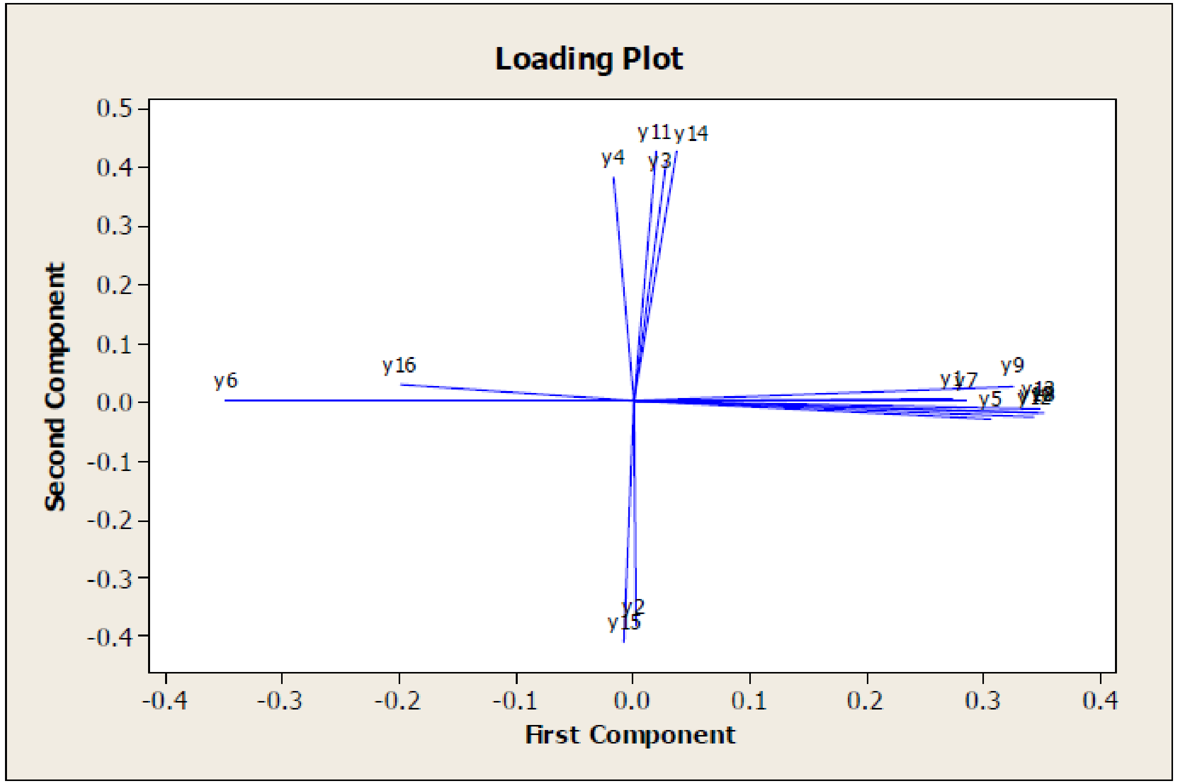
1D



Ogunnaike, Babatunde A. *Random phenomena: fundamentals of probability and statistics for engineers*. CRC Press, 2011.

Loading Plots

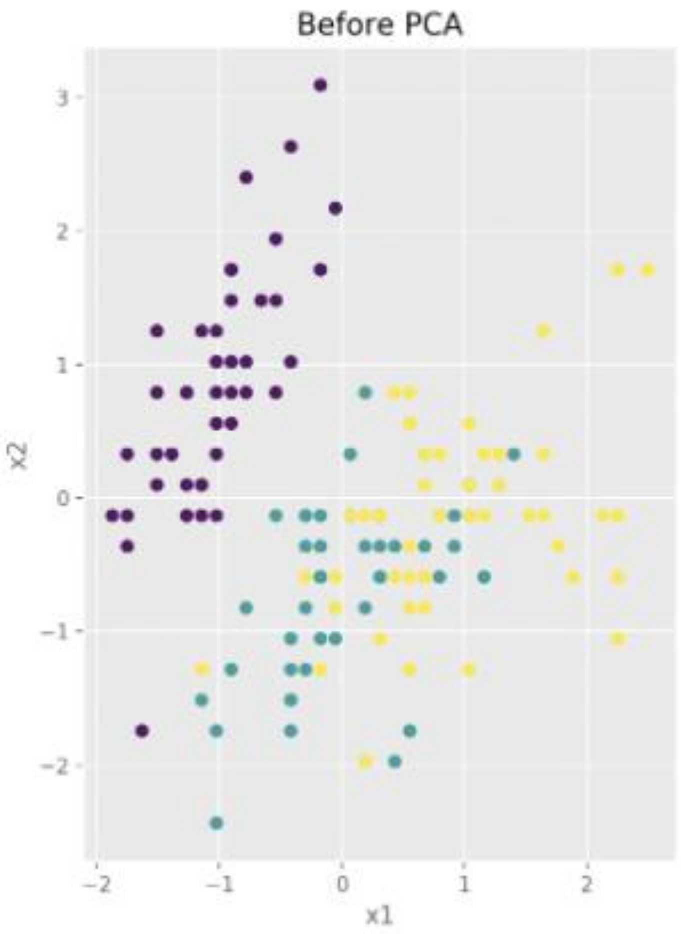
2D

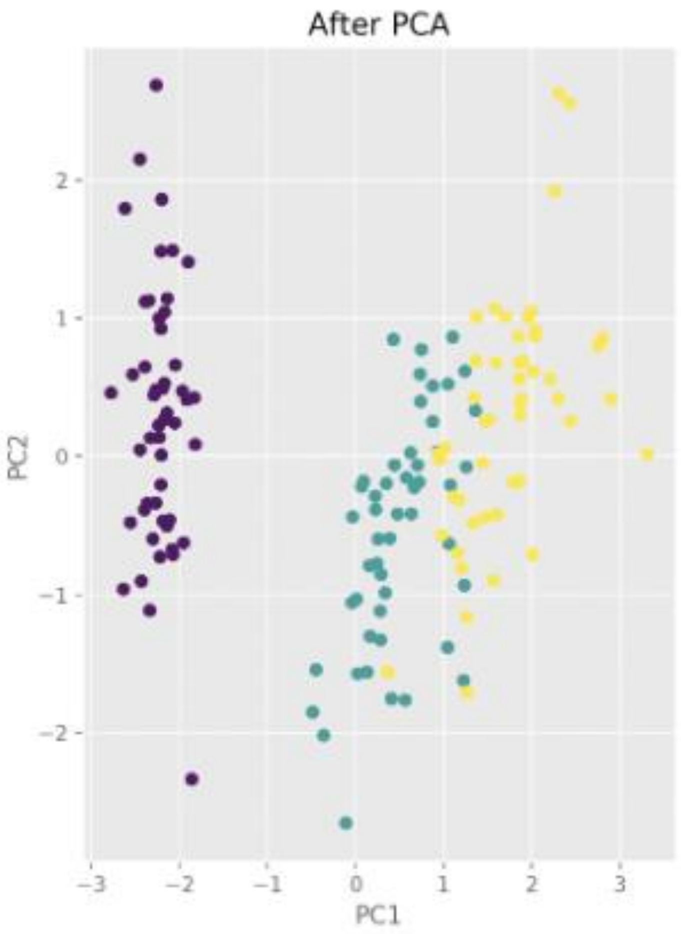


- Variables cluster together based on the nature of the component contributions
- Similar information as 1D loading plot
- Length represents how much contribution is made by one variable.
- X axis is the first component and Y axis is the second component.

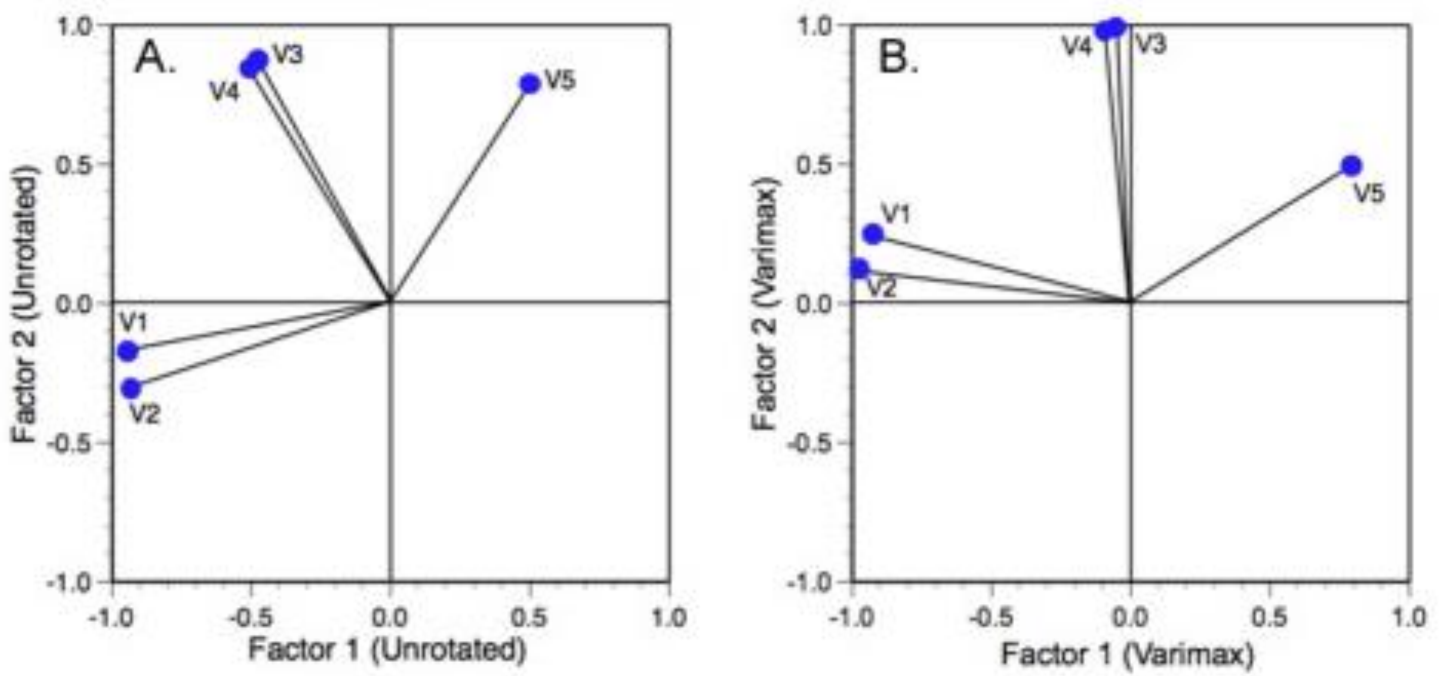
Ogunnaike, Babatunde A. *Random phenomena: fundamentals of probability and statistics for engineers*. CRC Press, 2011.

Visualization/Clustering





Varimax Rotation



- Cross-loading: one feature has significant weights across many principal components.
- Varimax rotation: Better interpretation of principal components. Rotation of PCA components to associate each component with one or a small number of factors.

https://online.stat.psu.edu/stat505/lesson/12/12.11

https://www.palass.org/publications/newsletter/palaeomath-101/palaeomath-part-6-factor-analysis

Practice Time!

Common Methods to Reduce Dimensionality

- Random forest

Random forest can tell us the **importance of each feature** present in the dataset. We can find the importance of each feature and keep the top most features, resulting in dimensionality reduction. It tends to bias toward variables that have more numbers of distinct values, i.e. **favor numeric variables** over binary/categorical values。

- t-SNE (t-Distributed Stochastic Neighbor Embedding)

Use a similarity measure like Euclidean distance to learn about discrepancies between pairs of data. The **local structures** of data are preserved which is not the case of PCA. It works well for visualization and also when the data is strongly **non-linear**.

- Factor analysis

It is best suited for situations when some variables are **highly correlated.** It divides the variables based on their correlation into different groups and represents each group with a factor. There are two methods performing factor analysis: EFA (Exploratory Factor Analysis) and CFA (Confirmatory Factor Analysis).

Common Methods to Reduce Dimensionality

- Linear dimensionality reduction methods

PCA: project data along the direction of increasing variance. The features with the maximum variance are the principal components.

Factor analysis

LDA: project data in a way that the class separability is maximized. Examples from same class are put closely together by the projection. Examples from different classes are placed far apart by the projection.

- Non-linear dimensionality reduction (manifold learning methods)

Multi-dimensional scaling (MDS)

Isometric feature mapping (Isomap)

Locally linear embedding (LLE)

t-SNE

Singular Value Decomposition (SVD)

Data preprocessing

Compute principal components

Select eigenvectors

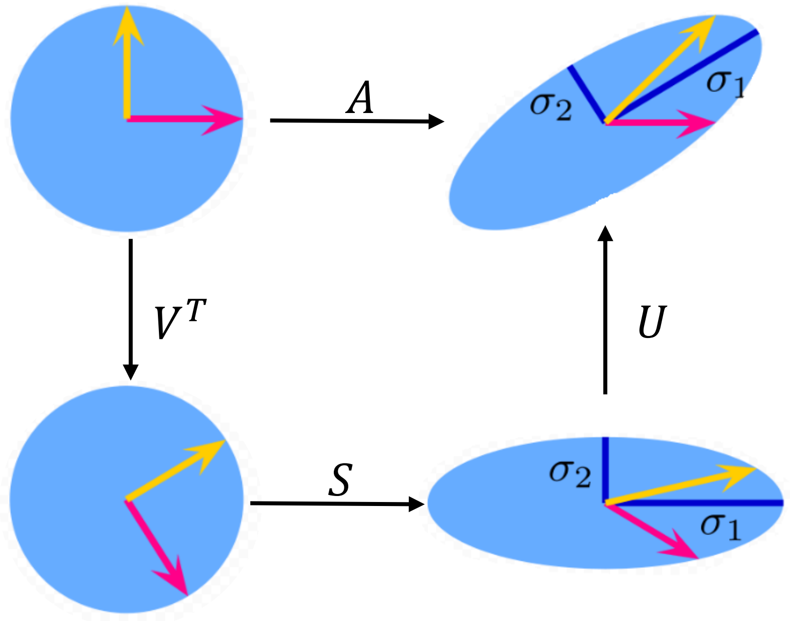
Carry out analysis

- SVD theorem:

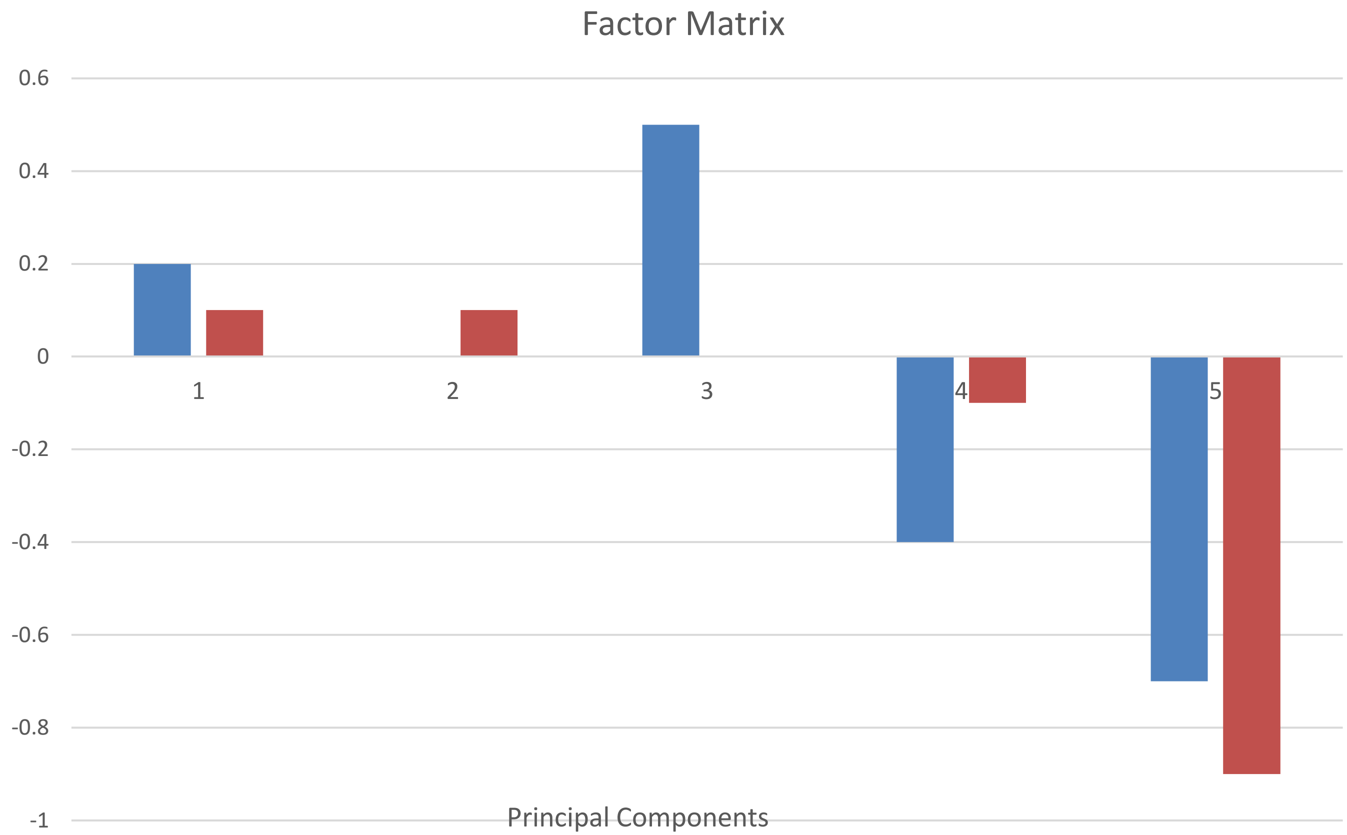
$$A_{n\times p} = U_{n\times n} S_{n\times p} V^T_{p\times p}$$

V: eigenvectors of $A^T A$
U: eigenvectors of AA^T
S: square roots of eigenvalues from $A^T A$ or AA^T

- The singular values are the diagonal entries of the S matrix and are arranged in descending order.



Varimax Rotation



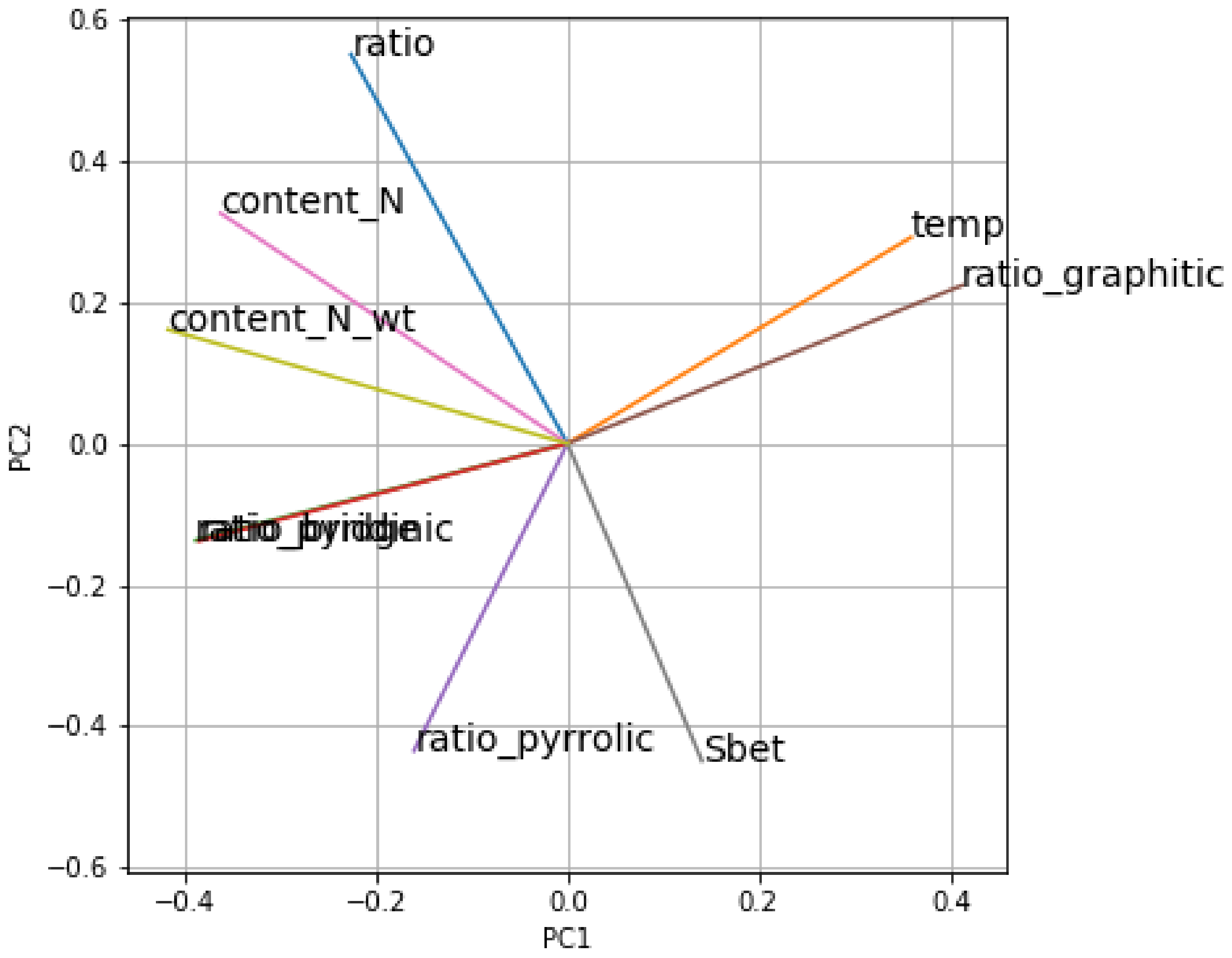

```
Factor matrix:
 [[ 0.2  0.2  0.3  0.3  0.3  0.4 -0.3  0.4  0.4  0.3]
 [ 0.  -0.4 -0.2 -0.1  0.6  0.5  0.5 -0.1 -0.  -0.1]
 [ 0.5 -0.1  0.2  0.5 -0.1 -0.3  0.4 -0.4  0.1  0.3]
 [-0.4 -0.7  0.5 -0.  -0.1 -0.2 -0.1  0.   0.3  0.1]
 [-0.7  0.4  0.1  0.5  0.1  0.1  0.2 -0.1 -0.2  0. ]]
 Number of cross-loadings: 20

Rotated factor matrix:
 [[ 0.1  0.   0.1  0.1  0.6  0.6  0.   0.4  0.2  0.1]
 [ 0.1  0.1  0.5  0.6  0.2  0.1 -0.1  0.2  0.4  0.4]
 [ 0.   0.2  0.3 -0.1 -0.   0.1 -0.7  0.5  0.3  0.2]
 [-0.1 -0.9  0.1 -0.3  0.1 -0.1  0.2 -0.2  0.1 -0.1]
 [-0.9 -0.1  0.1 -0.1 -0.1 -0.1 -0.  -0.1 -0.2 -0.2]]
 Number of cross_loadings: 13
```

https://pypi.org/project/advanced-pca/

Varimax Rotation: Steven's Example

Original PCA



PCA with Varimax Rotation