Outline: Principal Components Analysis (F. Chiaromonte)

Introduction to Statistical Learning
Chapter 10 sections 1, 2 and 4 Lab 1

Principal Component Analysis (PCA); Some context:

Unsupervised (no response, no labels) dimension reduction technique:

1. Capture the main signals, patterns, structure in a multivariate data set without any prior target or specified predictive aim – Exploratory Data Analysis (EDA).
2. Represent the data in lower dimension, perhaps to visualized them.
3. De-noise the data by removing negligible variation.
4. Use low-dimensional/de-noised data in input for supervised analyses.

Note: if the purpose is to create low-dimensional representations relevant for other analyses (4), other approaches exist:

- Linear Discriminant Analysis (LDA) prior to classification
- Sufficient Dimension Reduction (SDR) prior to regression
- Multi-Dimensional Scaling (MDS) prior to clustering

… the strength of PCA is its "general purpose" nature.

Also connections to:

- Independent Components Analysis (extract independent, as opposed to uncorrelated, components)
- Factor Analysis (extract latent components in the framework of a probabilistic model for the mechanism generating the data.)

Units\Features	X_1	X_2	…	X_p
Unit 1	x_{11}	x_{12}		x_{1p}
Unit 2	x_{21}	x_{22}		x_{2p}
. . .				
Unit n	x_{n1}	x_{n2}		x_{np}

p features measured on n units:

- An n x p data matrix X
- A data cloud of n points in R^p .

Location is irrelevant to PCA; assume each feature is centered, mean 0.

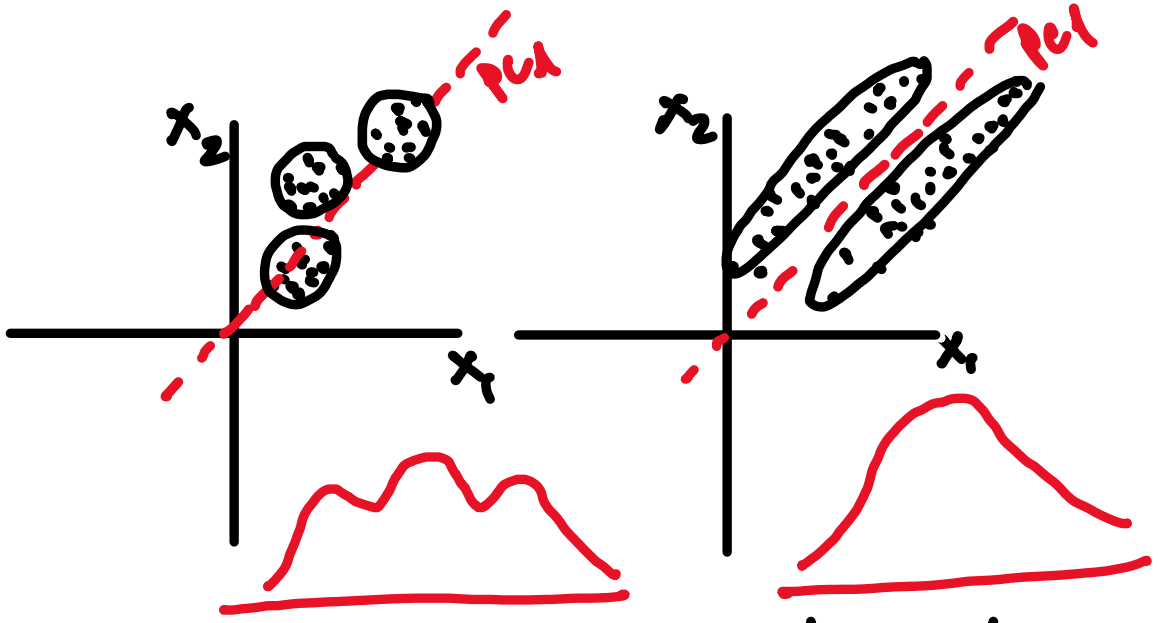
$\bar{X} = 0_p$ mean vector in R^p; cloud is centered/located at origin

$S = X'X$ (sample) p x p variance/covariance matrix

proportional to; divide by n or (n-1)

Find directions (and corresponding linear combinations) in R^p that are most interesting, in the sense of capturing variability of the data cloud.

General purpose, in many applications (but not all) these are the most informative, capturing structure in the data.



Projecting on PC1 preserves the cluster structure

does not

Sequentially, find orthogonal directions (orthonormal vectors) maximizing the variance of the corresponding projections:

$$\text{for } m = 1, 2 \ldots p:$$

$$\max_{\phi_m} \text{var}(\phi_m^T X)$$

$$\text{subject to } \| \phi_m \| = \phi_m^T \phi_m = 1, \phi_m^T \phi_k = 0 \ (k = 1, 2 \ldots m-1)$$

Equivalent to taking the Eigen decomposition of S:

$$S = \sum_{m=1}^{p} \lambda_m \phi_m \phi_m^T$$

$$\lambda_1 \geq \ldots \geq \lambda_p \geq 0 \quad \text{(eigenvalues)}$$

eigenvalues are variances along the directions identified by eigenvectors; in non-increasing order.

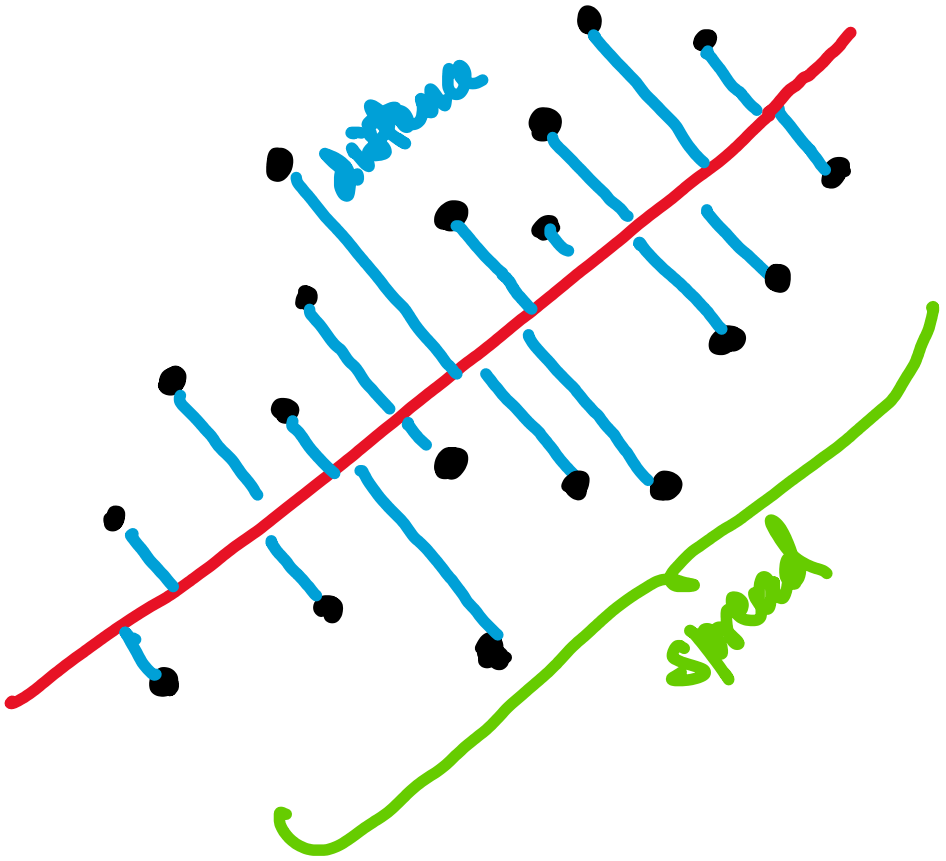
$$\| \phi_m \| = \phi_m^T \phi_m = 1, \phi_m^T \phi_k = 0 \ m, k = 1, 2 \ldots p \quad \text{(eigenvectors)}$$

For any given dimension *m*, identify the linear subspaces closest to the data:

$$\| P_{Span(\Phi_1 \ldots \Phi_m)} X \|^2 = \max$$

m-dimensional representation most likely to be useful in capturing structure, most informative (not always though!)

Note: here proper distance, least squares in regression uses "vertical" distance because one has a response to predict.



PC 1

minimizes the sum of squared distances

⇕ (equiv.)

maximizes the variance of the projected data

LOADINGS: coefficients expressing the m-th component in terms of the original features

$$\phi_{m1}, \quad \phi_{m2} \dots \phi_{mp}$$

(coordinates of each element of the new basis in terms of the original basis)

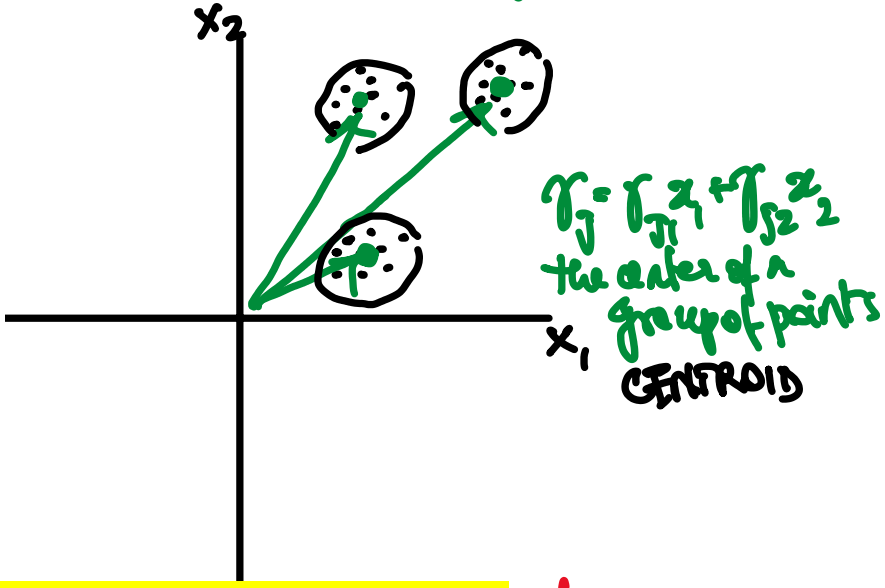
SCORES: values of the m-th component on the n units

$$z_{im} = \phi_{m1} x_{i1} + \phi_{m2} x_{i2} \dots + \phi_{mp} x_{ip} \quad , \quad i = 1, 2 \dots n$$

(coordinates of the n data points in terms of each element of the new basis)

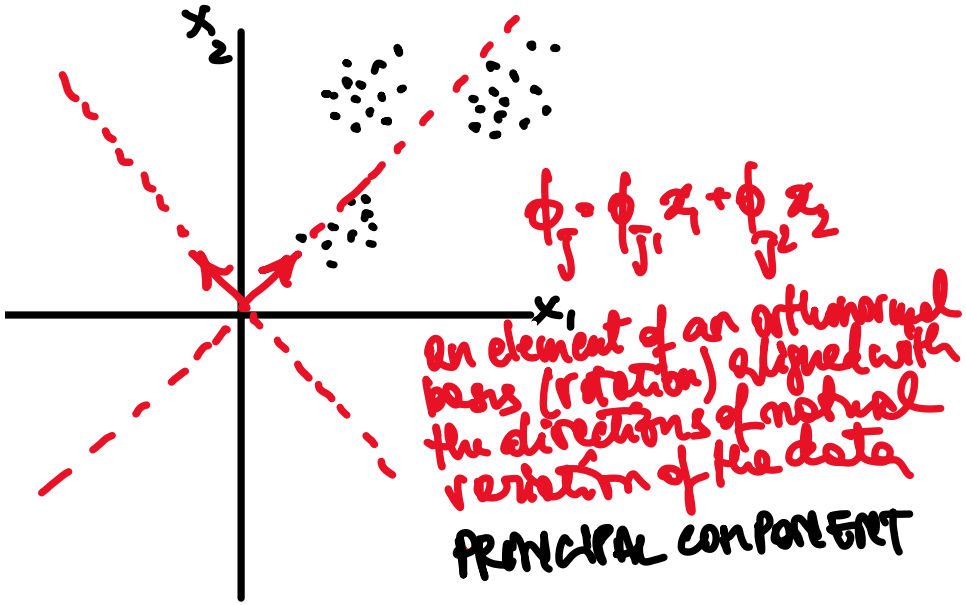
"PROTOTYPICAL" PATTERNS

$r_J, J=1,2,3$

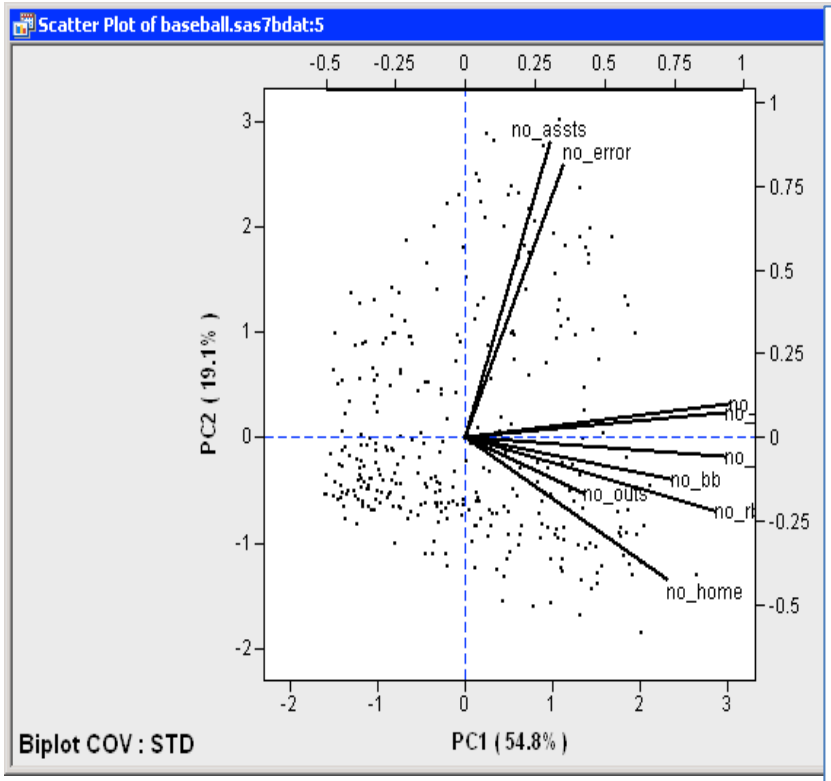


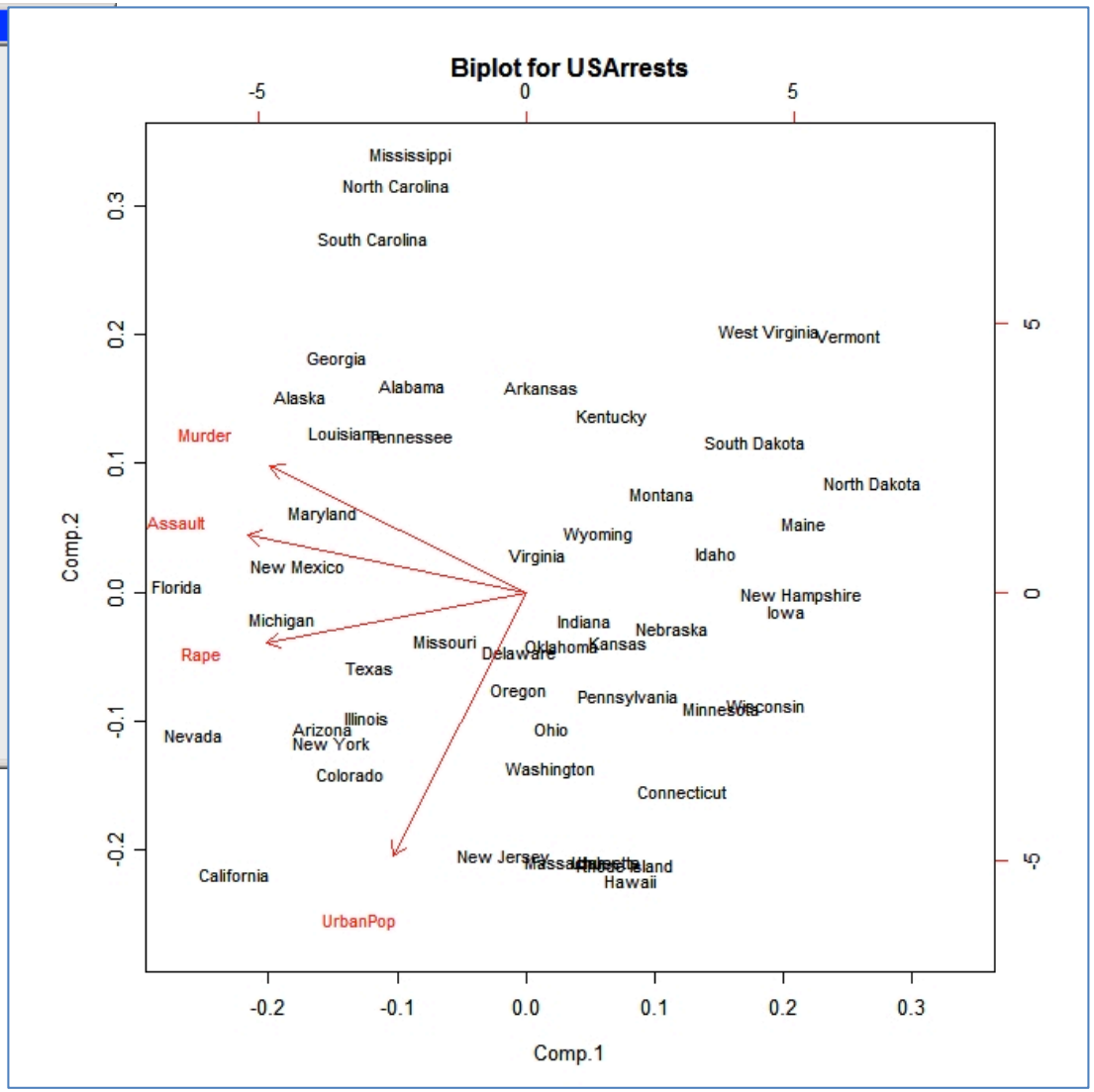
"COMPONENT" PATTERNS

$\phi_J, J=1,2$



Biplot: shows the scores for two specified components (e.g. 1st and 2nd; projection of the points on the first PCA plane) along with the loadings represented by arrows.





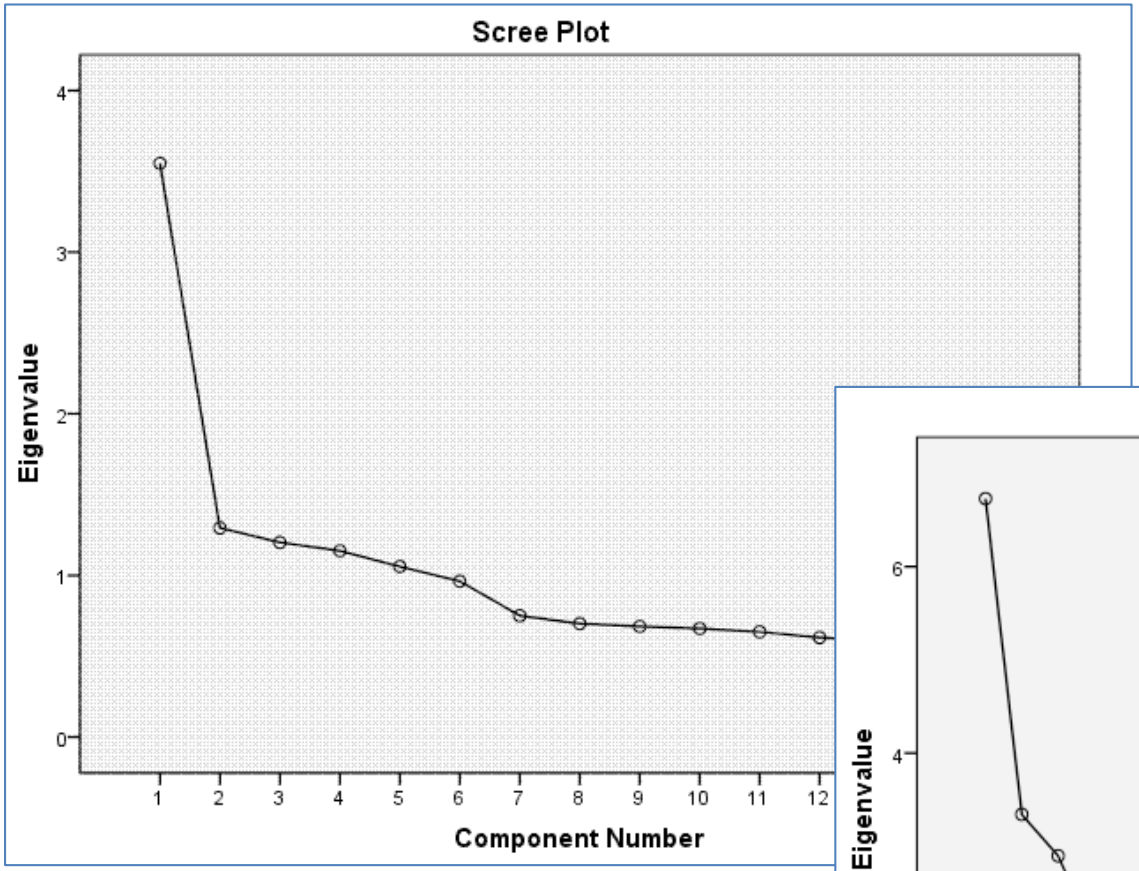
PERCENTAGE OF VARIANCE EXPLAINED (PVE) by the m-th component

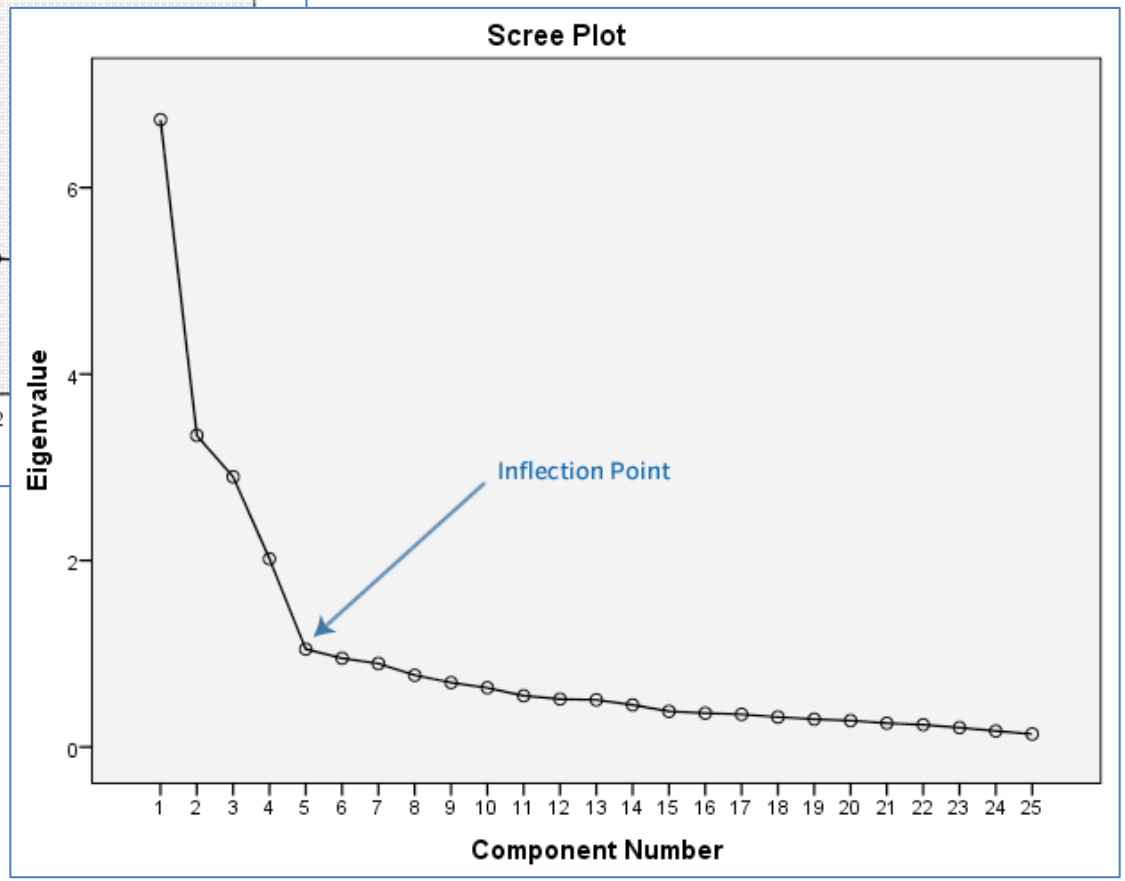
$$PVM_m = \frac{\lambda_m}{\sum_{k=1}^{p} \lambda_k} = \frac{\mathrm{var}(\phi_m^T X)}{\sum_{k=1}^{p} \mathrm{var}(\phi_k^T X)}$$

Also, **CUMULATIVE PVE** up to the m-th component

$$CPVM_m = \sum_{j=1}^{m} PVM_j = \frac{\sum_{j=1}^{m} \lambda_j}{\sum_{k=1}^{p} \lambda_k}$$

Scree plot: Show the variances (eigenvalues) or PVEs in non-increasing order.





How many components to consider?

$M \leq \min\{ p, n-1 \}$

Rule of thumb: look where the PVEs become very small and/or level off (stop decreasing substantially).

The book says this is necessarily ad-hoc because an unsupervised analysis does not have a prediction outcome that allows to select "tuning" parameters of the procedure through cross-validation.

But some less subjective approaches are available! Technically, assessing how many (tail) eigenvalues are not significantly > 0:

- Rigorous tests exist if the features are multivariate Gaussian
- Sampling variability of the eigenvalues can be gaged by bootstrapping (put bootstrap "bands" around the scree plot)
- Empirical null distribution of the eigenvalues can be simulated under appropriate assumptions
- Information-type criteria can be formed under appropriate assumptions, comprising a penalty for complexity.

Scaling: In some (but not all) circumstances, it is good to eliminate from consideration not just location but also units of measurement and/or magnitude of variation of the original features:

- Normalize each dividing by its standard deviation; $X \leftarrow \text{Diag}(s_j)^{-1/2} X$
- S becomes a correlation matrix (diagonal entries = 1; off diagonal entries in (-1,1)).

Note: unlike other analyses (e.g. regression) PCA is affected in non-trivial ways by scaling; the Eigen decompositions of $X'X$ and $X'DX$ (where D is diagonal) can be very different. Unless D is a multiple of the identity:

- Eigenvectors can change
- Eigenvalues can differ by more than just a scaling factor.

De-noised representation of the data:

based on a reconstruction in M components, take

$$x_{ij} \approx \sum_{m=1}^{M} z_{im}\phi_{jm1} \quad , \quad i = 1,2 \ldots n \, , \, j = 1,2 \ldots p$$

eliminates the “detail” (noise) in the directions that have been discarded; de-noise.

Some additions: **Multidimensional Scaling**

Given a matrix $D = \{d_{ij}\}$ containing the distances (dissimilarities) between each pair of n objects in a set, and a chosen number of dimensions, k, MDS places each object into a k-dimensional Euclidian space as to retain as much as possible the distances between objects. If k = *1, 2* or *3*, the resulting data cloud can be visualized.

Stress function

$$S_k(\boldsymbol{X}) = \sqrt{\frac{\sum_{i<j}(d(x_i; xj) - dij)^2}{\sum_{i<j} d(x_i; xj)^2}} \longrightarrow \boldsymbol{X}_k{}^* = argmin\ S_k(\boldsymbol{X})$$

The minimization can be performed numerically under a variety of specifications of the problem.

In the simplest, where D is a true Euclidian distance matrix, this reduces to an Eigen decomposition problem like in PCA. Form the scores in the Eigen-space spanned by the eigenvectors of the k largest eigenvalues of

$$A = \left\{-\frac{1}{2}d^2{}_{ij}\right\}$$

Centered Matrix

$$B = \{a_{ij} - ai_{\cdot}\_a_{\cdot j} + a_{\cdot\cdot}\}$$

Can pick a reasonable k using the eigenvalues.

Notes:

- The stress can be brought to 0 for k $\geq$ n-1.
- Non-uniqueness, e.g., any translation or rotation of the solution produces an equally good solution.