Machine learning: PCA I

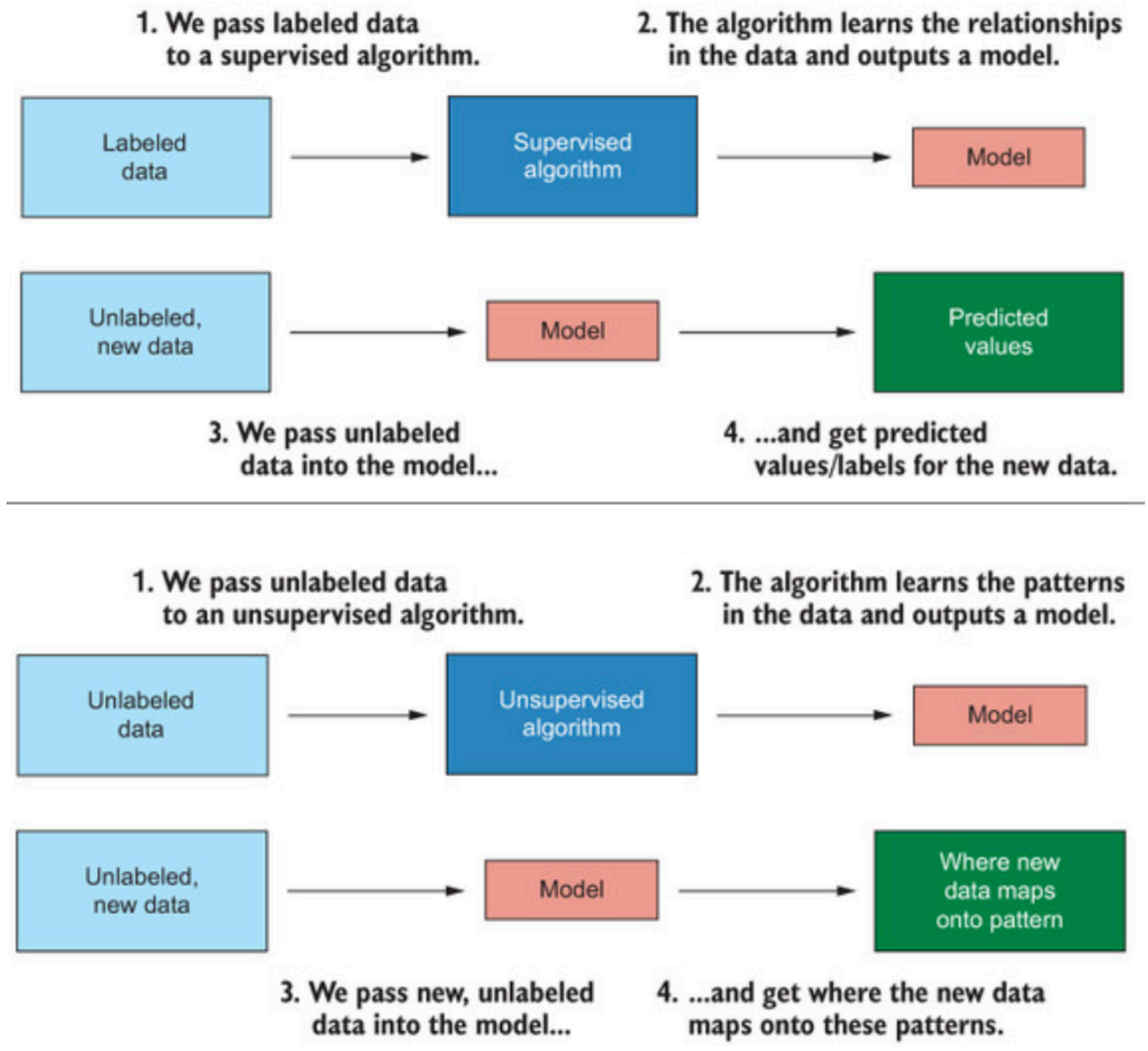
Alex Di Genova

27/06/2024

Machine learning algorithms

Classes

- Supervised
  - Classification
  - Regression
- Unsupervised
  - Dimension Reduction
  - Clustering
- Semi-supervised



Fitting a Linear Model in R

1. Data Preparation
2. Building the Model
3. Evaluating the Model

```
> glimpse(mtcars)
Rows: 32
Columns: 11
$ mpg  <dbl> 21.0, 21.0, 22.8, 21.4, 18.7,…
$ cyl  <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8,…
$ disp <dbl> 160.0, 160.0, 108.0, 258.0,…
$ hp   <dbl> 110, 110, 93, 110, 175, 105,…
$ drat <dbl> 3.90, 3.90, 3.85, 3.08, 3.15,…
$ wt   <dbl> 2.620, 2.875, 2.320, 3.215,…
$ qsec <dbl> 16.46, 17.02, 18.61, 19.44,…
$ vs   <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0…
$ am   <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,…
$ gear <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4,…
$ carb <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4,…
```

```
# Load necessary libraries
library(ggplot2)

# Load the data
data(mtcars)
head(mtcars)

# Fit a simple linear regression model
model <- lm(mpg ~ wt, data=mtcars)

# Summary of the model
summary(model)

# Plotting the data and the model
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point() +
  geom_smooth(method="lm", col="blue")
```

Fitting a Linear Model in R

Call:
lm(formula = mpg ~ wt, data = mtcars)

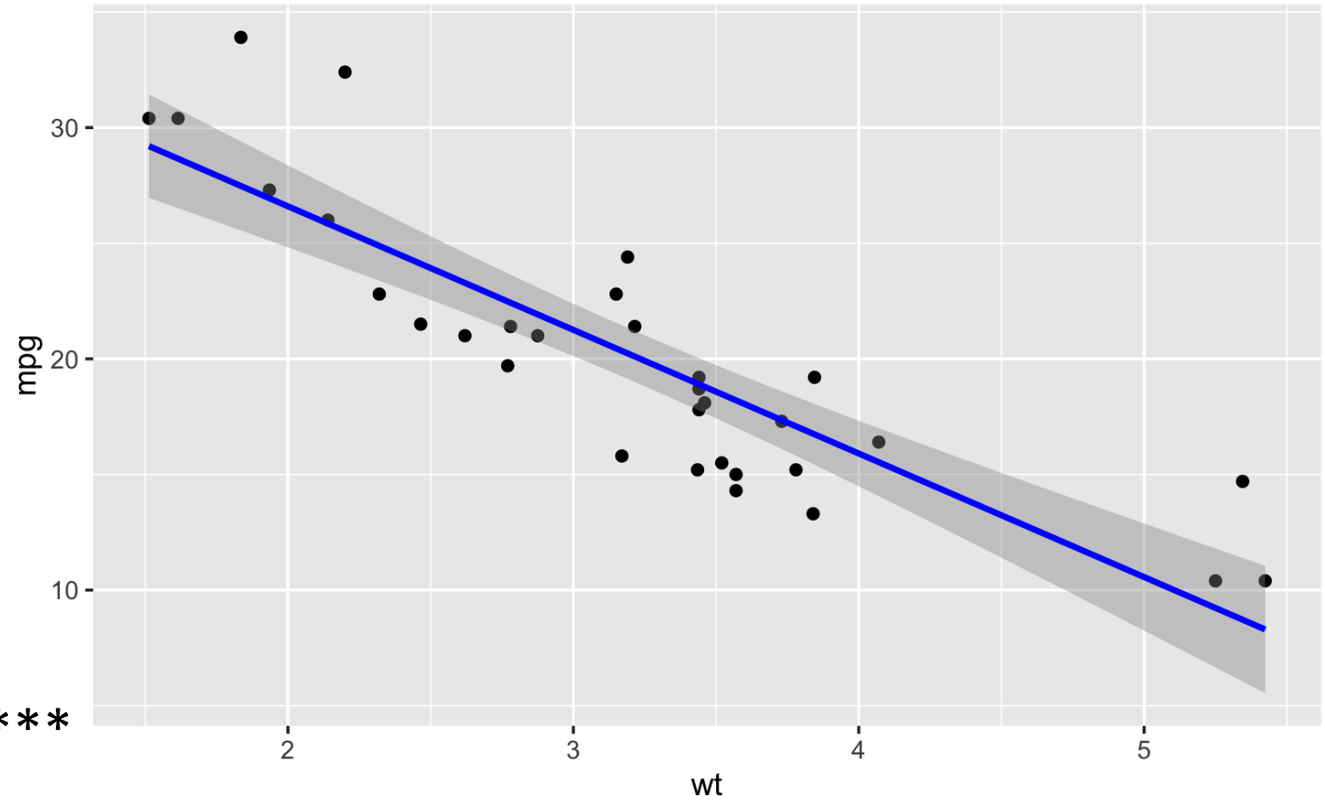
Residuals:
Min	1Q	Median	3Q	Max
-4.5432	-2.3647	-0.1252	1.4096	6.8727

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	37.2851	1.8776	19.858	< 2e-16	***
wt	-5.3445	0.5591	-9.559	1.29e-10	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared: 0.7528, Adjusted R-squared: 0.7446
F-statistic: 91.38 on 1 and 30 DF, p-value: 1.294e-10



Multiple Linear Regression in R

```
# Fit a multiple linear regression model
model_mult <- lm(mpg ~ wt + hp + qsec, data=mtcars)

# Summary of the model
summary(model_mult)

# Diagnostic plots
par(mfrow=c(2,2))
plot(model_mult)
```

```
Call:
lm(formula = mpg ~ wt + hp + qsec, data = mtcars)

Residuals:
   Min     1Q  Median     3Q    Max
-3.8591 -1.6418 -0.4636  1.1940  5.6092

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 27.61053    8.41993   3.279  0.00278 **
wt          -4.35880    0.75270  -5.791 3.22e-06 ***
hp          -0.01782    0.01498  -1.190  0.24418
qsec         0.51083    0.43922   1.163  0.25463
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.578 on 28 degrees of freedom
Multiple R-squared:  0.8348,     Adjusted R-squared:  0.8171
F-statistic: 47.15 on 3 and 28 DF,  p-value: 4.506e-11
```

How to select features?

- **Domain Knowledge**
- **Correlation Analysis**
  - cor_matrix <- cor(dataset) cor_matrix["target_variable",]
- **Variance Inflation Factor (VIF)**
  - Identify multicollinearity among features.
  - library(car) vif(model)
  - Features with a VIF > 5 or 10 should be removed from model.
- **Stepwise Selection**
  - Add or remove features sequentially based on statistical criteria (e.g., AIC, BIC).

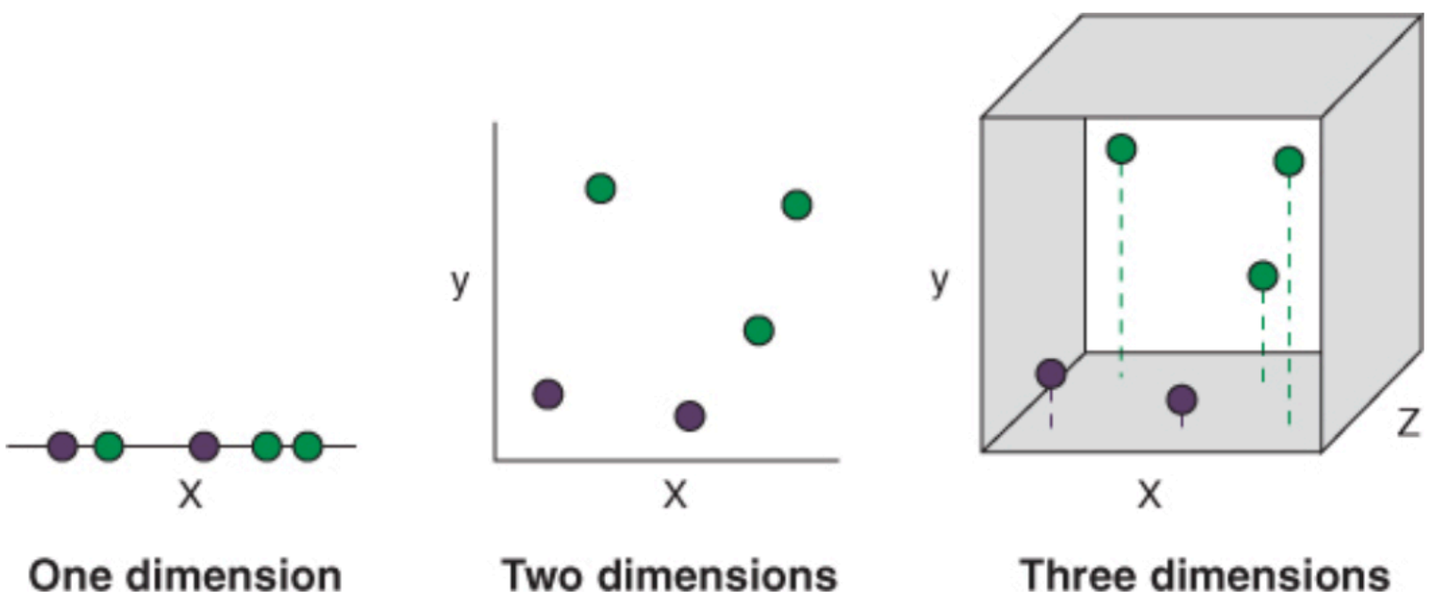
```
model <- lm(target_variable ~ ., data=dataset)

step_model <- stepAIC(model, direction="both")

summary(step_model)
```

- **Tree-Based Methods**
  - Feature importance can be derived from tree-based methods like Random Forests
- **PCA (Principal Component Analysis)**
  - Reduce dimensionality while retaining most of the variance in the data.
  - Use principal components as features instead of the original ones.

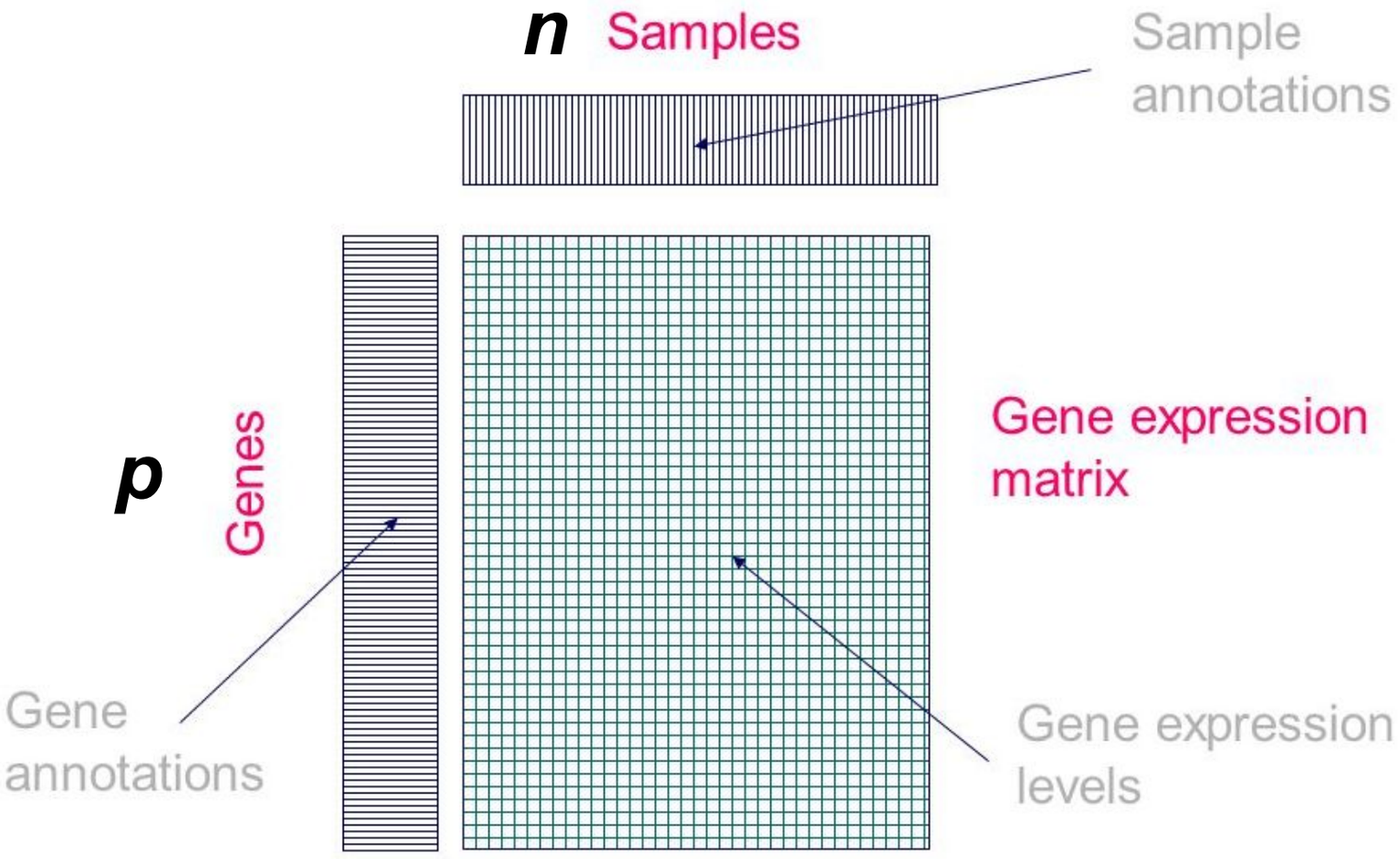
PCA: Principal Componets Analysis

Dimention Reduction

- Dimension reduction comprises a number of approaches that turn a set of (potentially many) variables into a smaller number of variables that retain as much of the original, multidimensional information as possible.
  - Making it easier to visualize a dataset with many variables.
  - Mitigating the curse of dimensionality.
  - Mitigating the effects of collinearity.



Matrix representation of data



$$X = \overbrace{\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots \\ x_{p1} & \cdots & x_{pn} \end{bmatrix}}^{n} \Big\} p$$

x_{ij} : value of variable i for individual j.

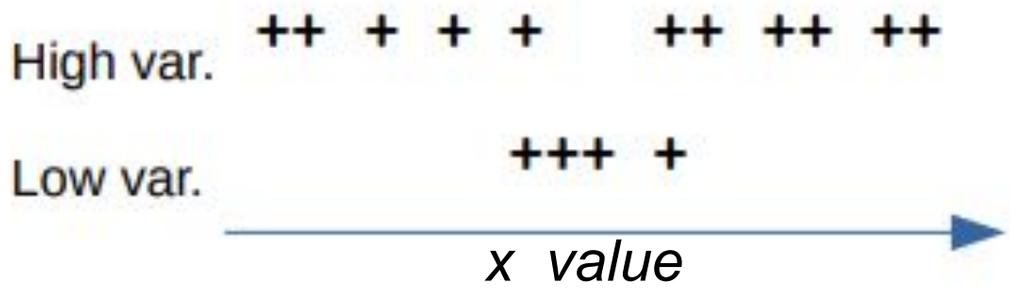
→ e.g. value of gene i for sample j

- Or transposed, n x p.

Basic vocabulary

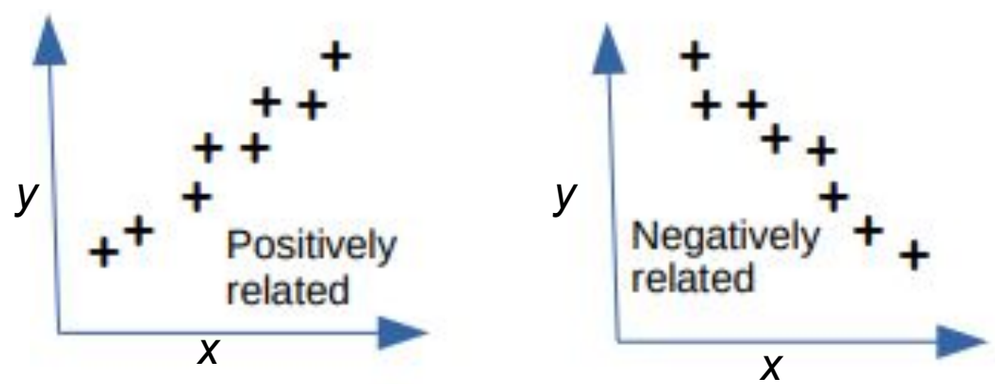
- **Variance:** indicator of spread for one variable *x.*

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad \text{with} \qquad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$



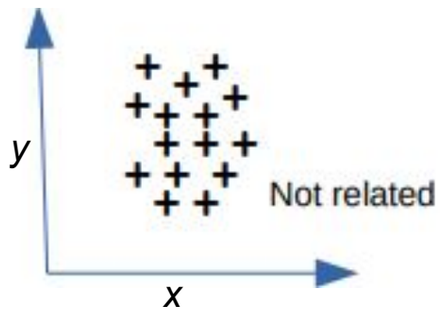
- **Covariance:** indicator of relationships for two variables *x* and *y.*

$$Cov(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n}$$



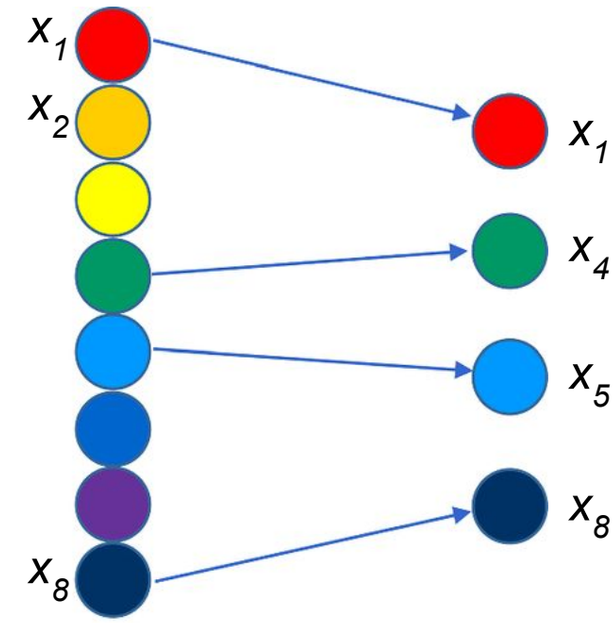
- **Correlation:** standardized covariance between -1 and 1

$$Cor(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \qquad \text{with} \qquad \sigma_X = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$$

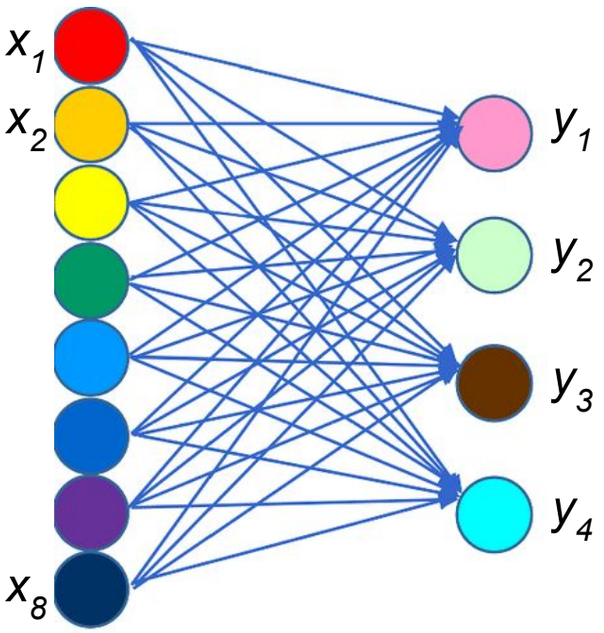


Feature selection vs extraction

- F. Selection: determine a smaller set of features minimizing (relevant) information loss.



- F. Extraction: combine the input features into another set of variables in a linear or non-linear way: $y_1 = \alpha_1 * x_1 + \alpha_2 * x_2 + \alpha_3 * x_3 + ...$

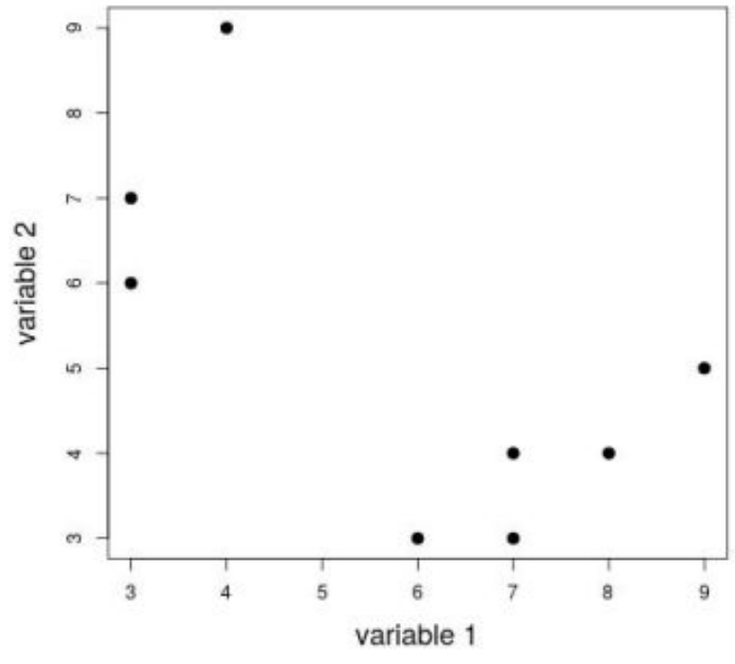


Dimensionality reduction

- Problem: ***n*** samples, ***p*** quantitative variables (e.g. peptides, proteins, metabolites, mRNA, …)

Visualize pairwise relations by scatter plots

***p*=2**

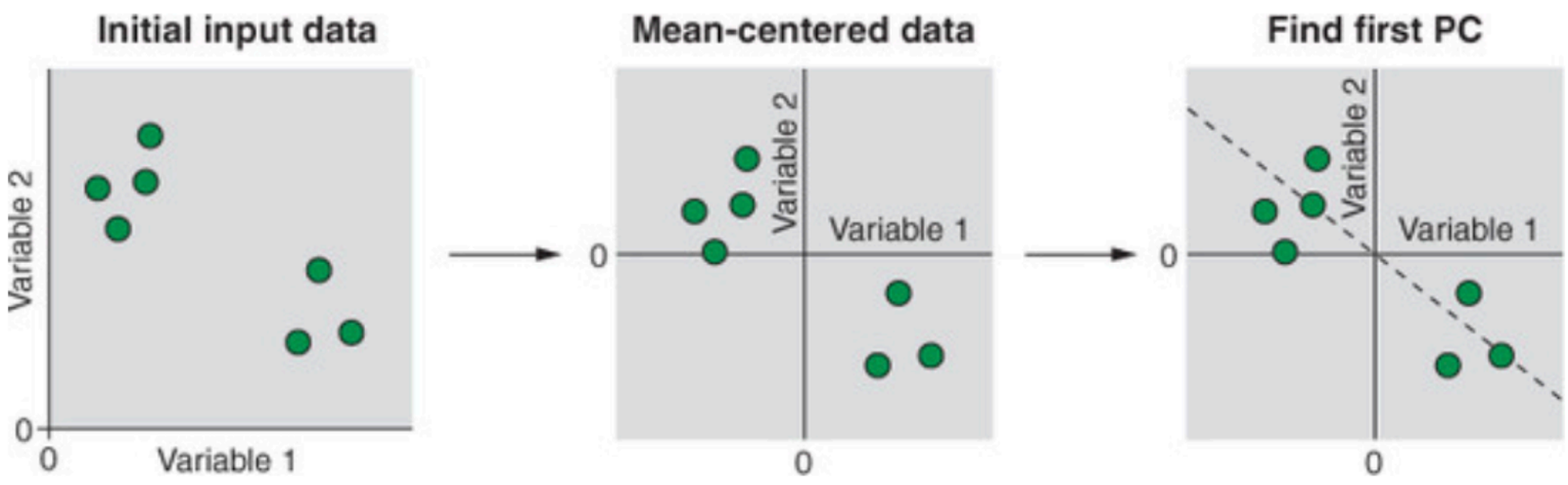


But when ***p*** is large ?

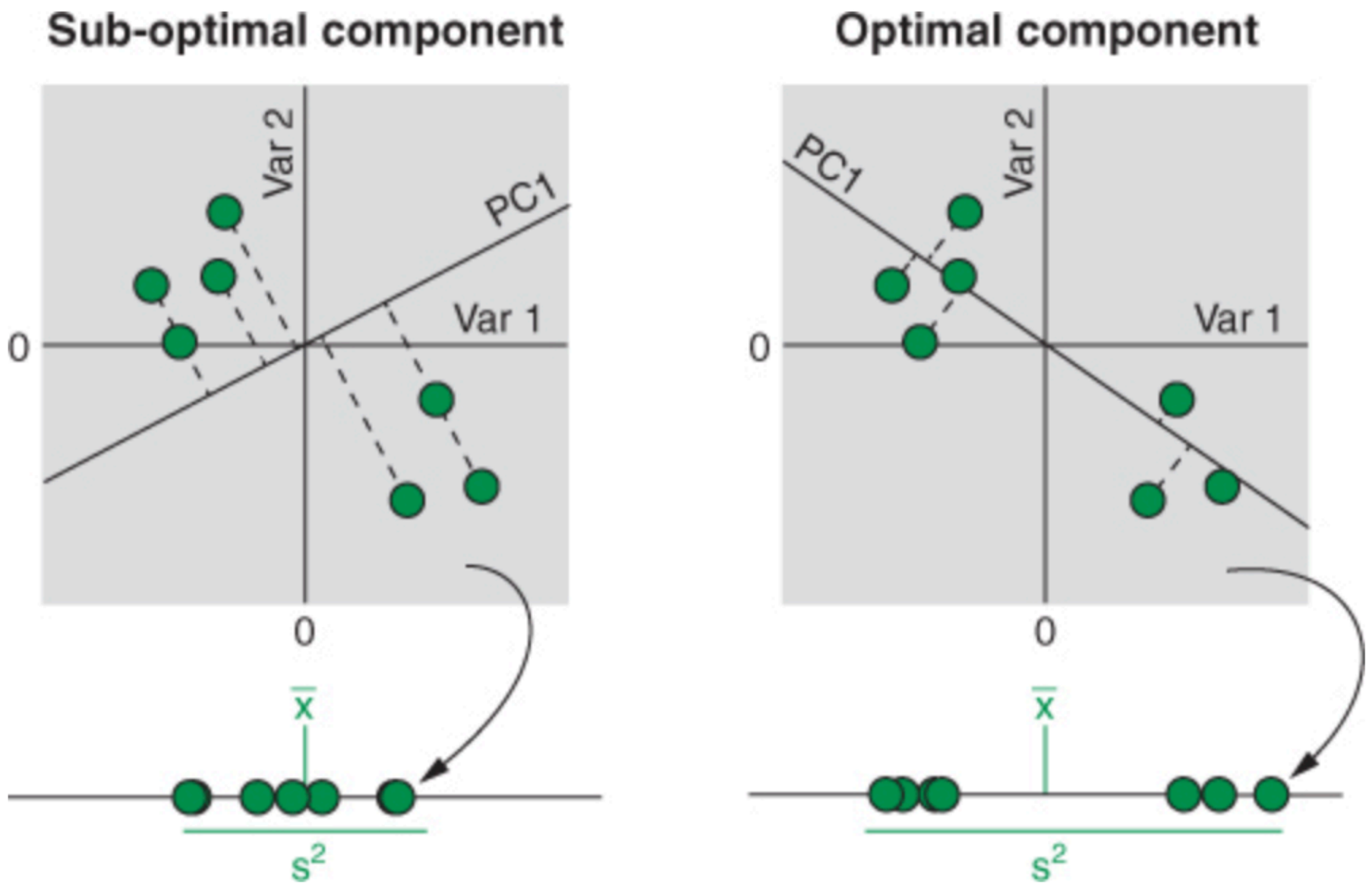
- **Need to reduce this large number of dimensions (*p*) to a smaller number of relevant variables, i.e, variables that carry most of the information (or variance) of a dataset and without redundancy.**

PCA

- Principal Component Analysis (PCA) is a method for reducing the dimensionality of datasets, increasing interpretability while minimizing information loss.
- It transforms the data into a new coordinate system.
- Creates new variables that are linear combinations of the original variables.
- The new variables explain most of the variation/information in the data

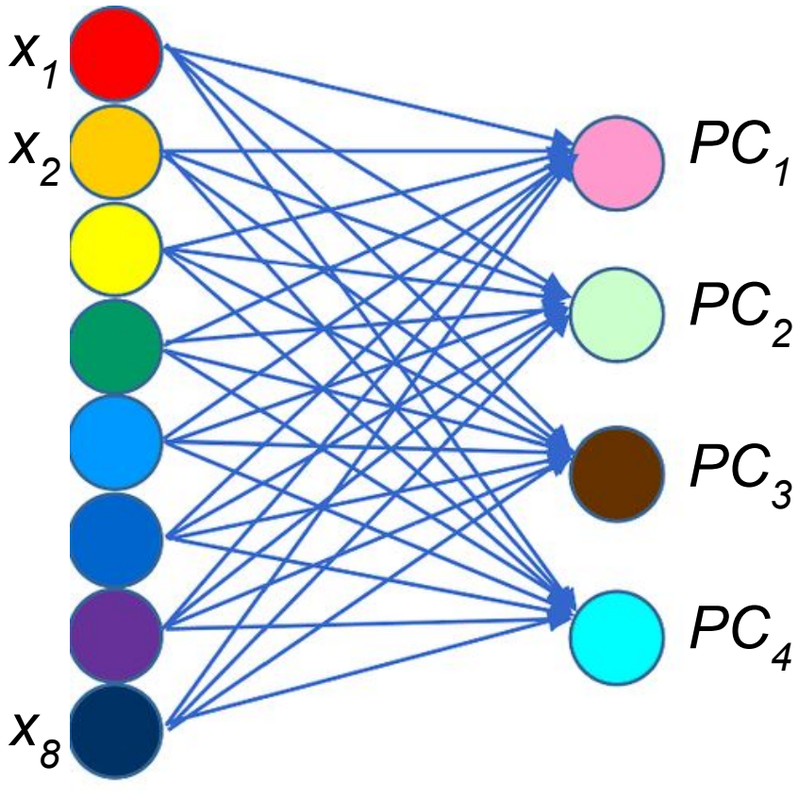


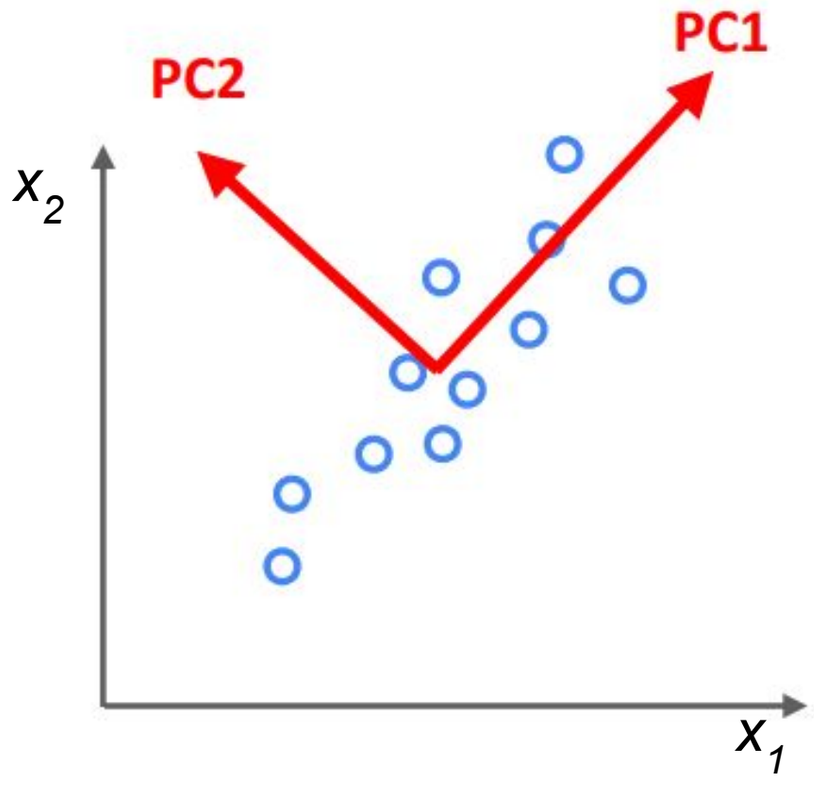
PCA



- PC1= 0.95 var1 + (-0.32) var2

PCA principle





$$PC_1 = \alpha_1 * x_1 + \alpha_2 * x_2 + \alpha_3 * x_3 + \ldots.$$

- Find orthogonal axes (Principal Components) on which one can project sample to obtain a comprehensible space of reduced dimension.

PCA Computing

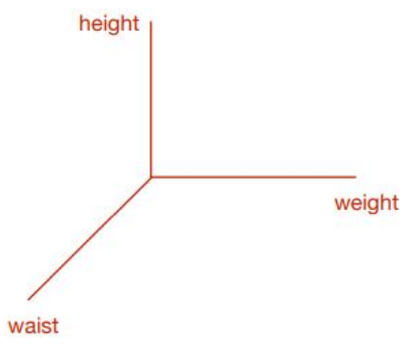
- 1. Standardize the data. $$z = \frac{x - \mu}{\sigma}$$
- 2. Compute the covariance matrix. $$\boldsymbol{A} \begin{bmatrix} var(x) & cov(x,y) & cov(x,z) \\ cov(x,y) & var(y) & cov(y,z) \\ cov(x,z) & cov(y,z) & var(z) \end{bmatrix}$$
- 3. Calculate the eigenvalues and eigenvectors.

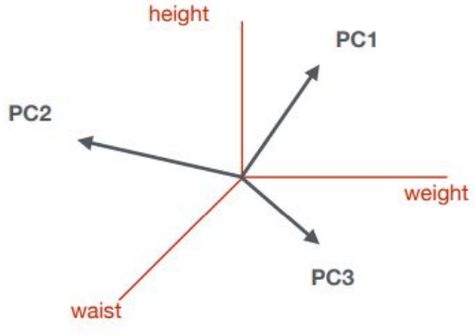
  $\rightarrow$ *solve* $|\mathbf{A}-\lambda\cdot\mathbf{I}| = 0$

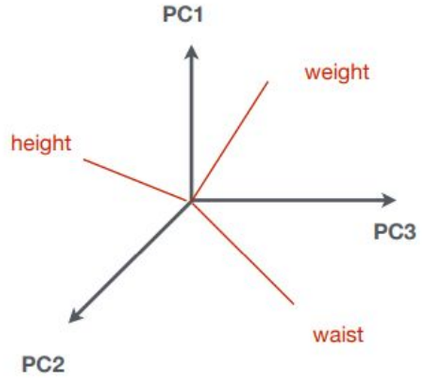
- 4. Sort eigenvalues and corresponding eigenvectors, and select k components.
- 5. Recast the data along the principal component axes

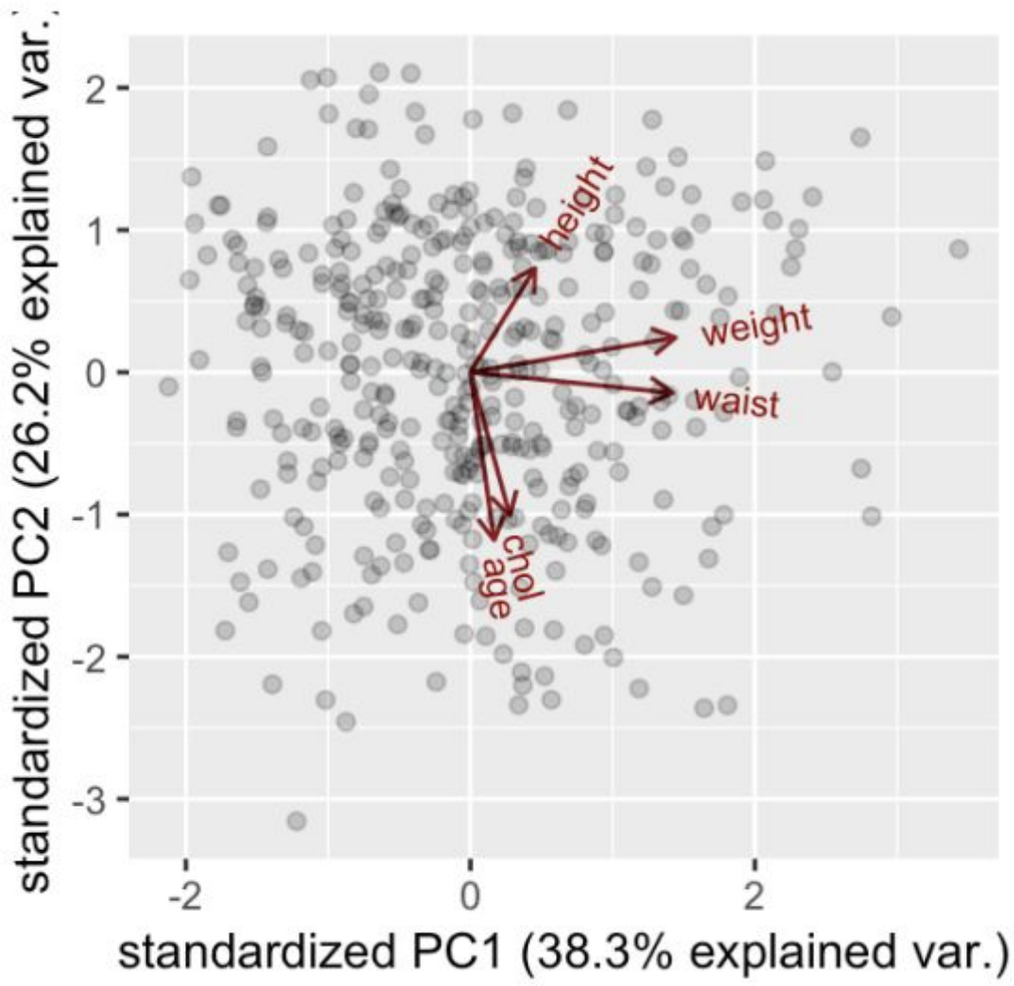
PCA Plots

- each dot is a sample
- new coordinate system (PC1, PC2…)
- red arrows = contribution of each initial variable (old coordinate system)
- several 2D (2 PCs) plots :
  - PC1/PC2
  - PC1/PC3
  - PC2/PC3





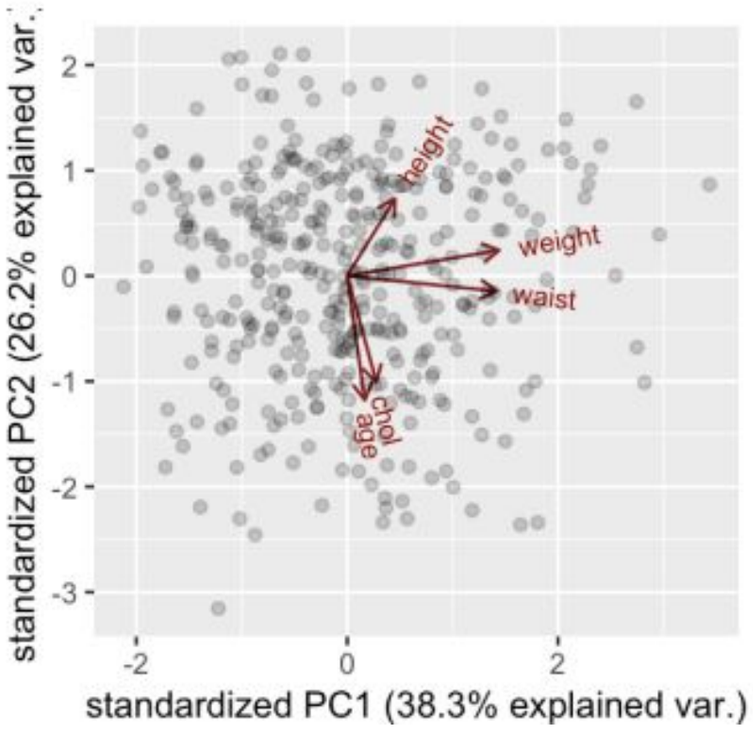


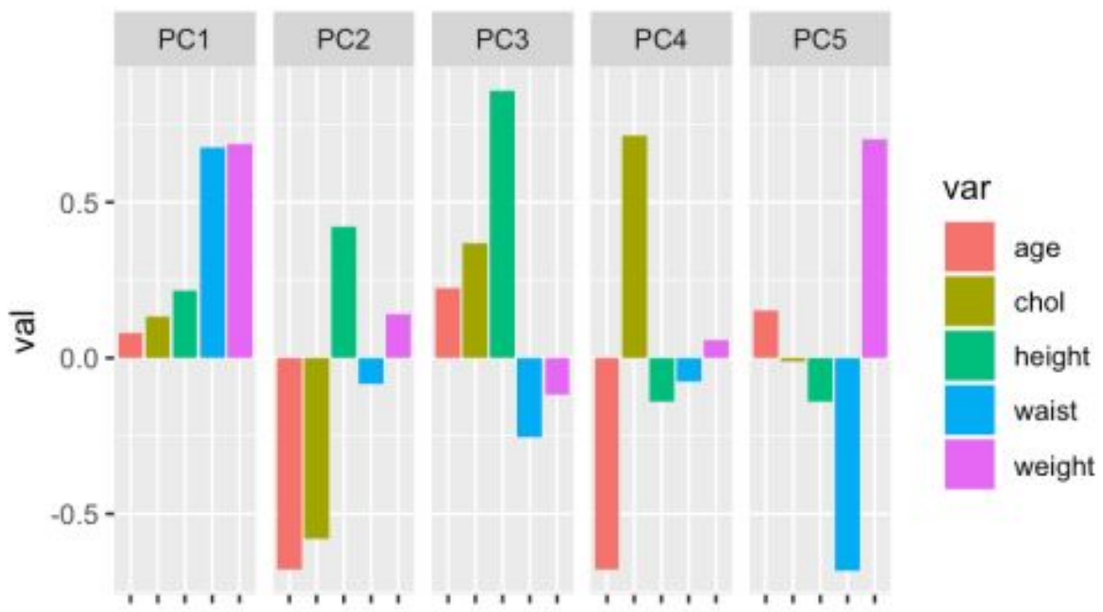


PCA components

- contribution of each initial variable to the *PC*: α, β, γ...are coefficients also called “loadings”
- some variables contribute in the same direction to some PCs (e.g. waist and height for PC1),
  but opposite to others (PC5)
- PC are orthogonal: no information redundancy between PCs.

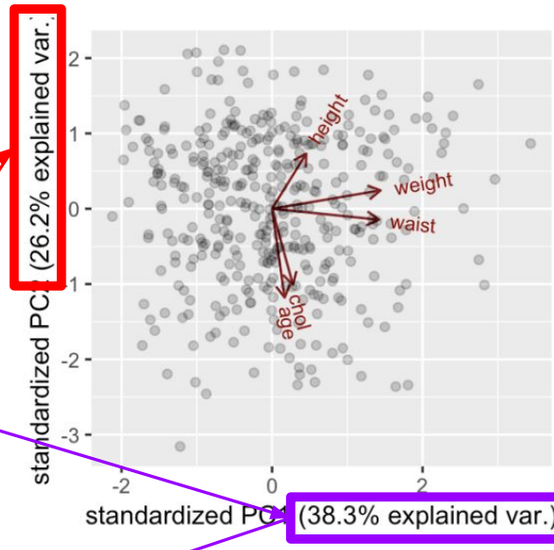
$$PC_i = \alpha_i \cdot \text{age} + \beta_i \cdot \text{chol} + \gamma_i \cdot \text{height} + \delta_i \cdot \text{waist} + \epsilon_i \cdot \text{weight}$$

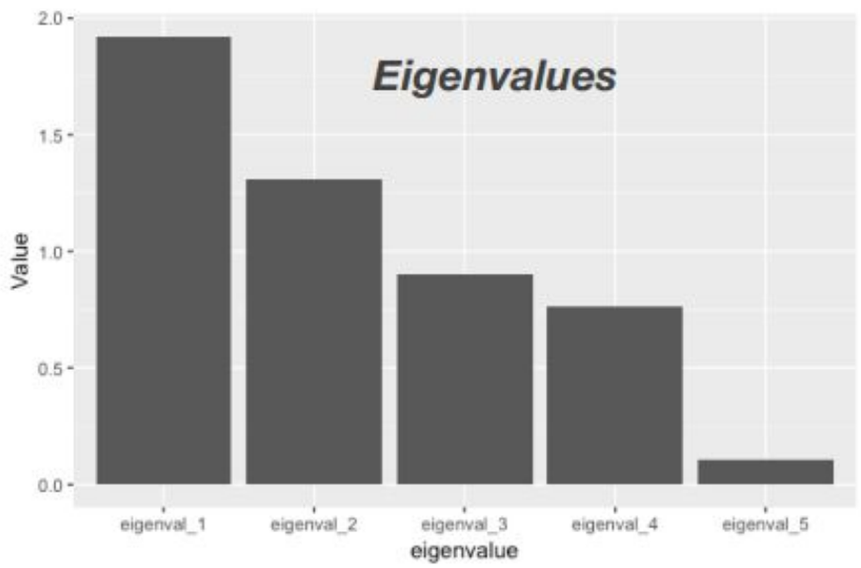


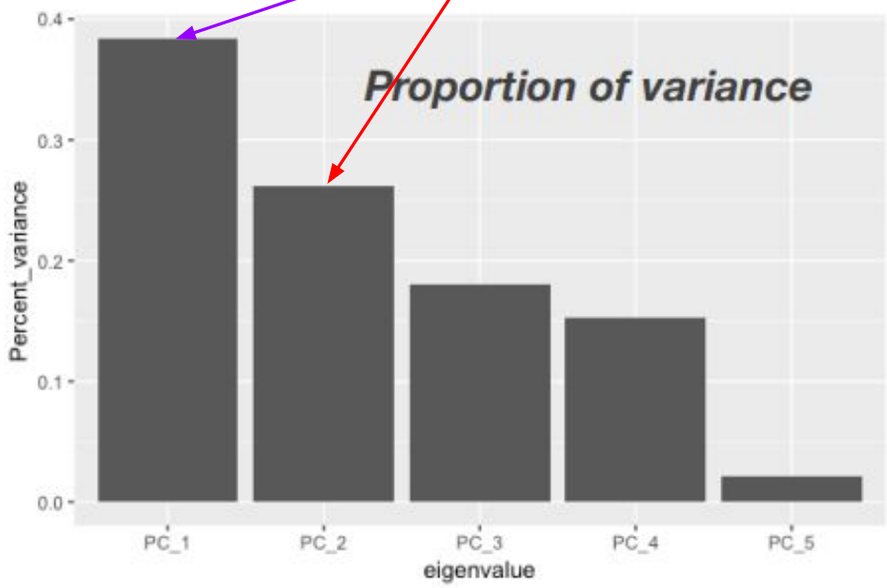


PCA components

- Each PC explains some part of the total variance of the dataset
- This amount is proportional to the corresponding eigenvalue
- PC are ordered by decreasing eigenvalue (hence explained variance)



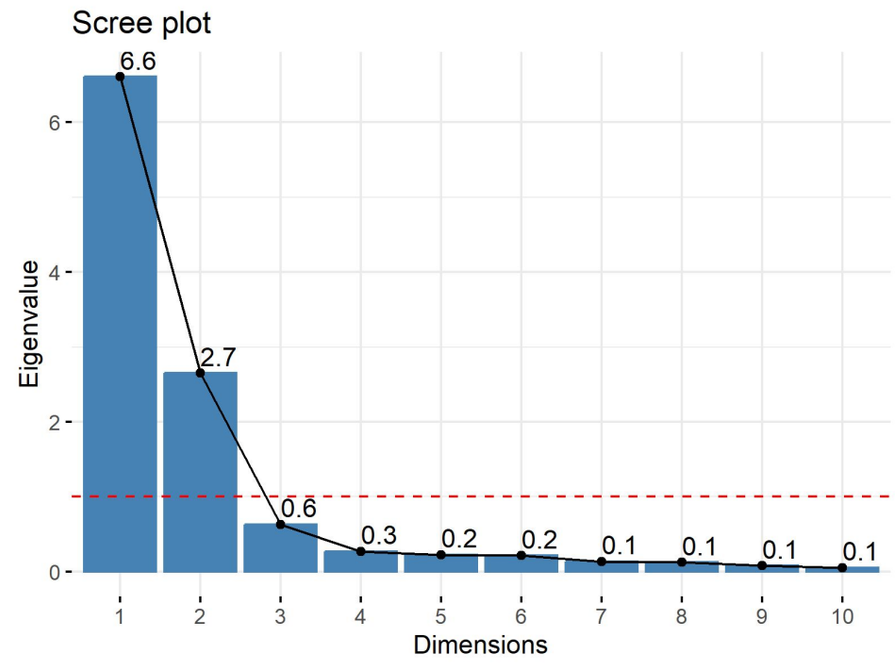




Considering PC1 & PC2 explains 63% of the total vairance

How many PCs?

- Several criteria to select the optimal subset of PC, without loosing too much information
- Proportion of total variance: keep PC such that the cumulative variance is above threshold
- Average eigenvalue criteria: keep PC which have eigenvalue larger than
  - mean eigenvalue (Kaiser rule) or
  - 70% of mean eigenvalue (Jottclife rule)



R packages for PCA

- Several functions from different packages are available in the *R software* for computing PCA:
- *prcomp*() and *princomp*() [built-in R *stats* package],
- *PCA*() [*FactoMineR* package],
- dudi.pca() [ade4 package],
- and epPCA() [ExPosition package]
- PCAtools [https://bioconductor.org/packages/release/bioc/vignettes/PCAtools/inst/doc/PCAtools.html]

A useful workflow:
Unsupervised -> Supervised Learning

- Often, start with unsupervised learning, which can:
  - Reduce number of dimensions
  - Provide insights into natural clusters
  - Generate more natural features
- Then run supervised learning methods!

Book:

Machine learning with R

Questions?

Practice!!!