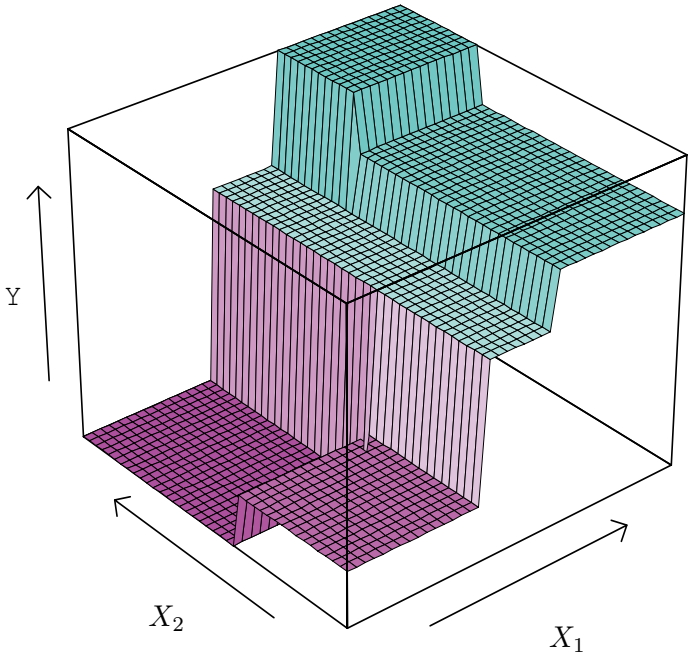
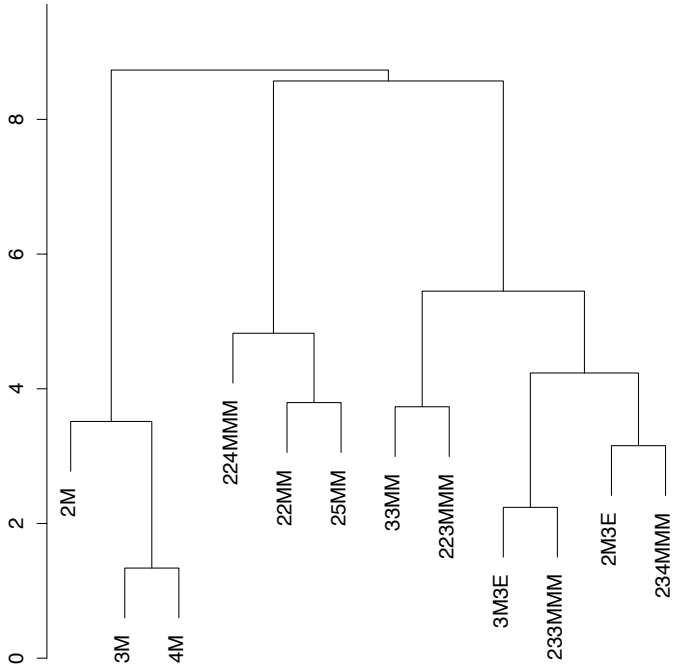
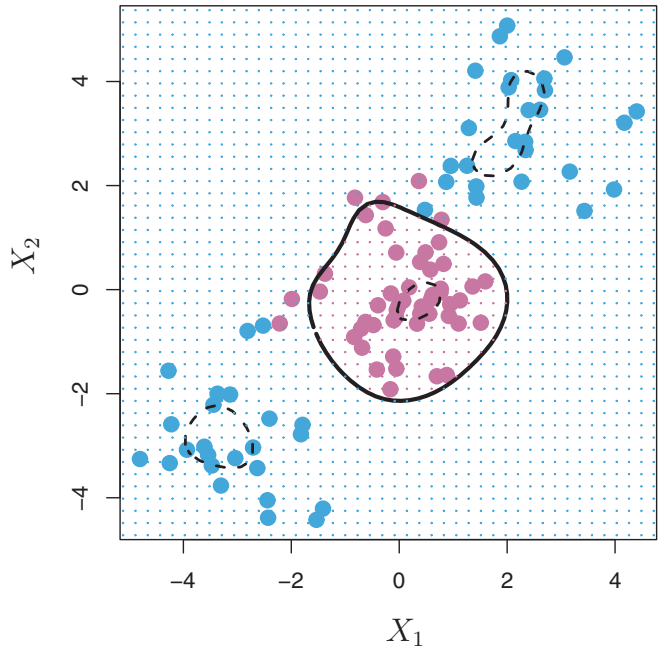
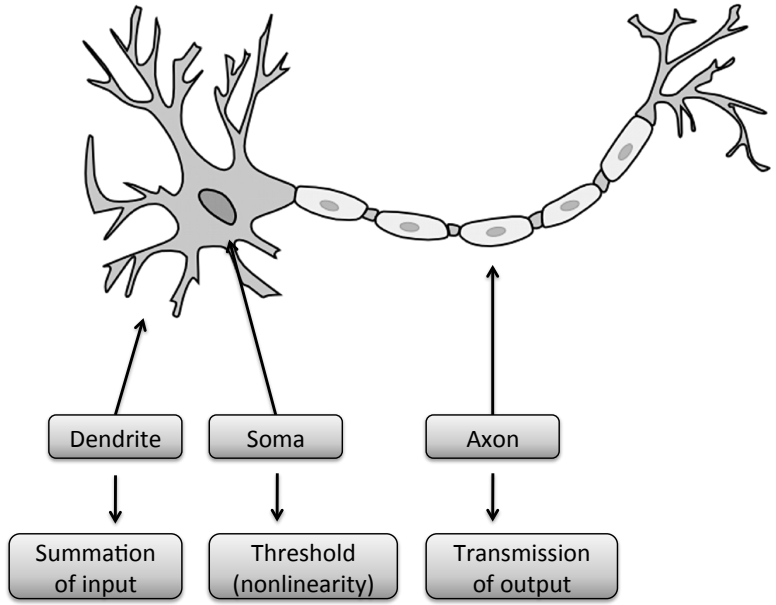


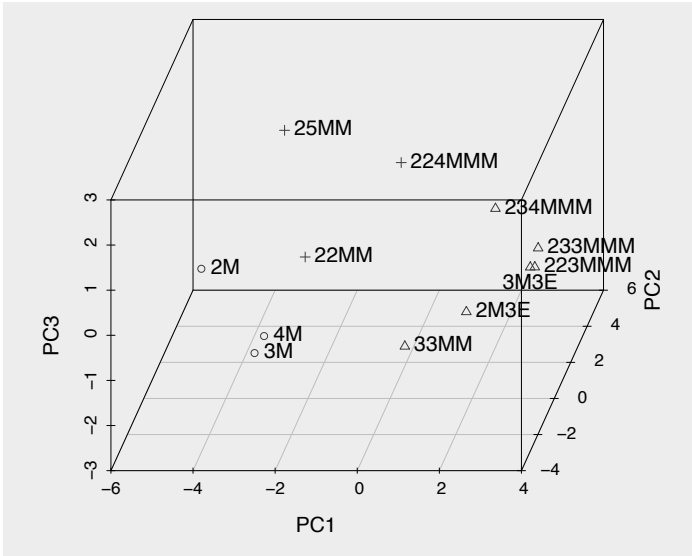
Machine Learning

By Phil Harrison
philip.harrison@farmbio.uu.se
Dept. of Pharmaceutical bioinformatics
Uppsala University



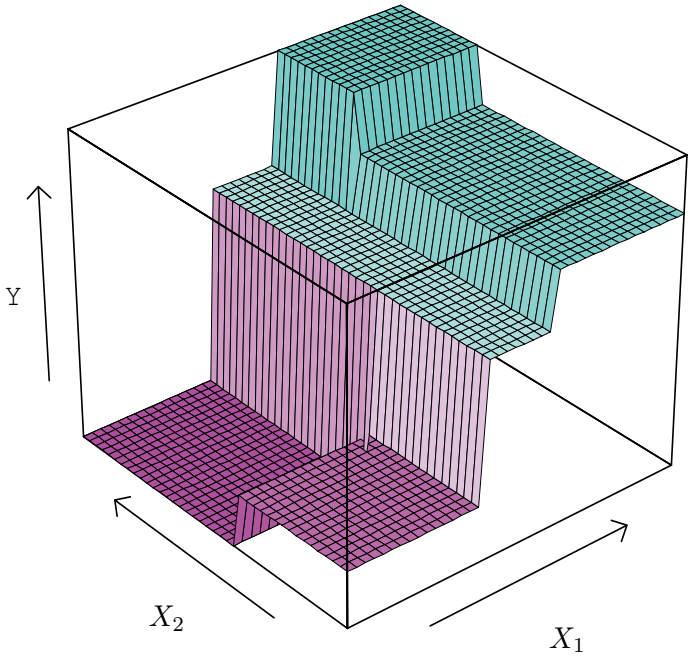


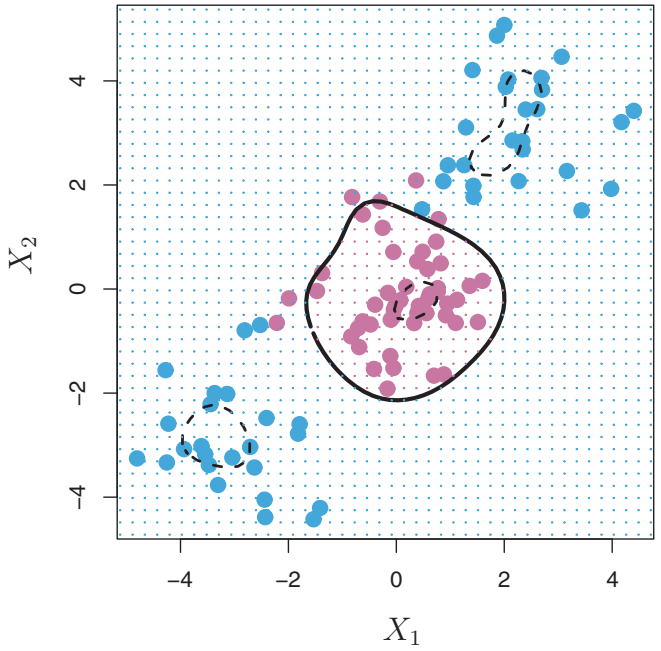


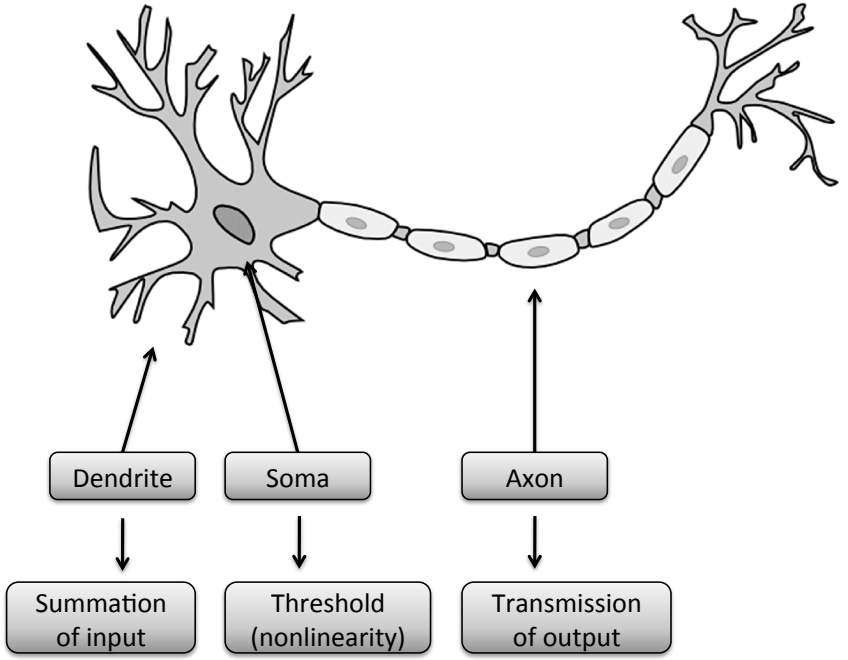


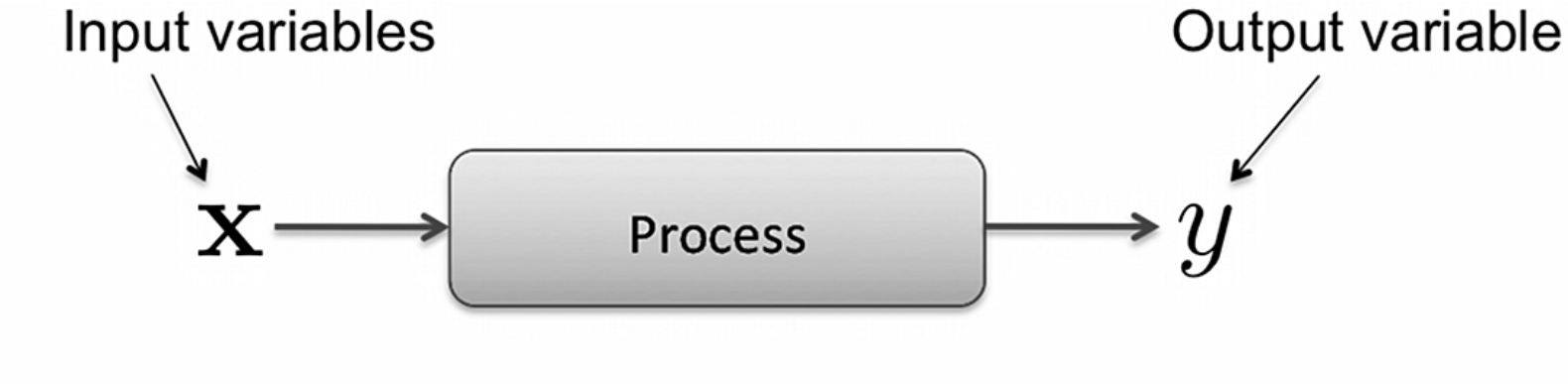
This lecture is based around chapter 11 ("Data analysis") from the course text book (Introduction to **Pharmaceutical Bioinformatics**)

Part 1: Supervised Machine Learning









Examples from pharmaceutical bioinformatics:

Input	Output
Compound & receptor physiochemical properties	Binding affinity
Microarray gene expression measurements	Risk of metastasis for a cancer patient
SNPs & physiochemical properties of drug	Treatment outcome
Physiochemical properties of a compound	Toxicity (LD_{50} in cell based systems)

Goals: prediction & inference

Input	Output
Compound & receptor physiochemical properties	Binding affinity
Microarray gene expression measurements	Risk of metastasis for a cancer patient
SNPs & physiochemical properties of drug	Treatment outcome
Physiochemical properties of a compound	Toxicity (LD_{50} in cell based systems)

Prediction	Inference
Will an untested compound bind?	Which properties important for binding?
Metastesis risk for a newly diagnosed patient	Which genes increase risk?
Efficacy of a given drug on a new patient	Which SNPs determine efficacy?
Toxicity of an untested compound	Which properties affect toxicity?

Some important distinctions

- supervised vs unsupervised machine learning
- deterministic vs stochastic models
- parametric vs non-parametric methods
- regression vs classification
- linear vs nonlinear models
- static vs dynamic models

Supervised Learning

Regression

$$Y = g(X, \beta) + \varepsilon$$

e.g. modelling binding affinity between a receptor and a set of compounds

Classification

$$P(Y|X = \mathbf{x}) = g(X, \beta)$$

e.g. modelling whether or not a receptor binds to a set of compounds

Common loss functions

Regression

squared error loss

$$L(Y, g(X,\beta)) = (Y - g(X,\beta))^2$$

Classification

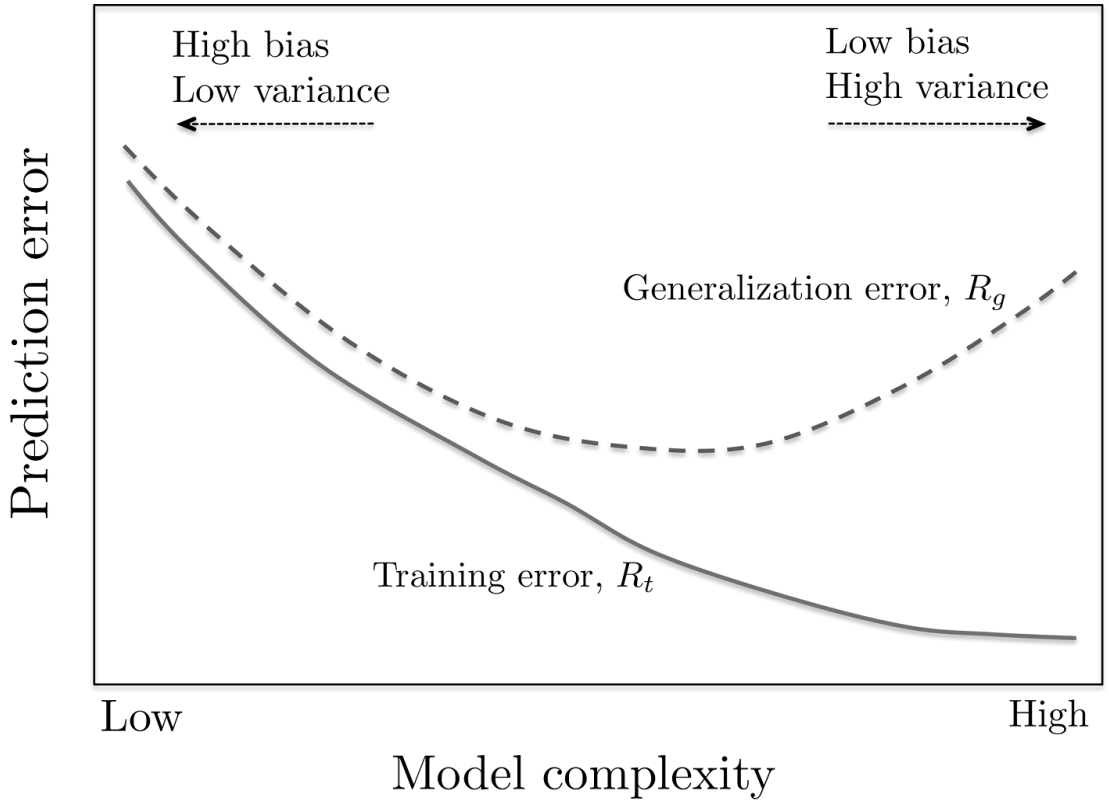
0-1 loss

$$L(Y, g(X,\beta)) = \begin{cases} 0 & \text{if } Y = g(X,\beta) \\ 1 & \text{if } Y \neq g(X,\beta) \end{cases}$$

Model choice & the bias-variance tradeoff

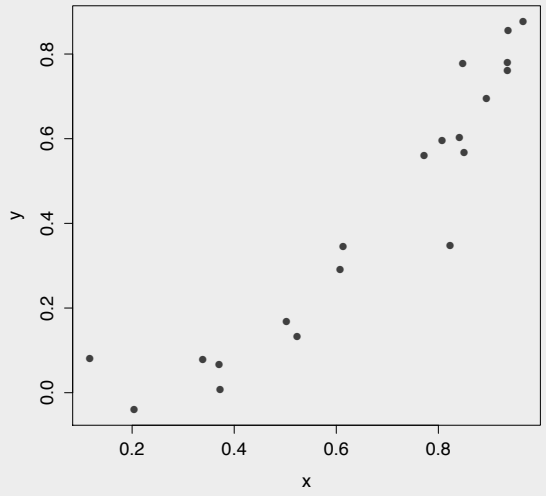
training error:

$$R_t = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{g}(\mathbf{x}_i))$$

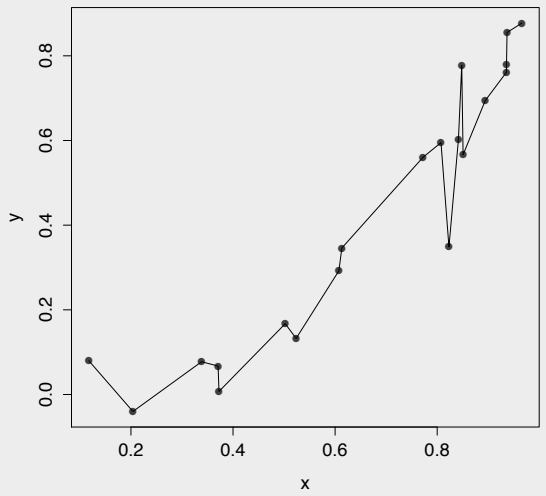


generalisation (test) error:

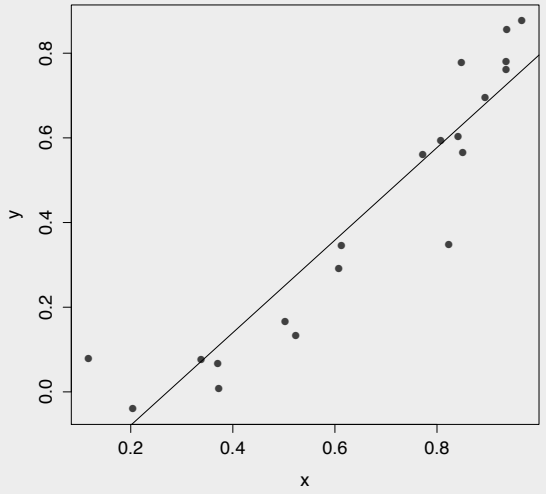
$$R_g = \mathrm{E}[(Y - \hat{g}(X))^2] = \int \mathrm{E}\left[(Y - \hat{g}(\mathbf{x}))^2 \mid X = \mathbf{x}\right] f_X(\mathbf{x}) d\mathbf{x}$$



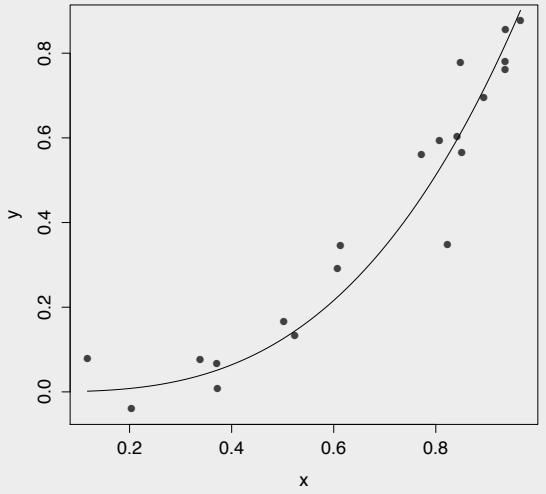
(a) Scatterplot of $\mathbf{x}$ versus $\mathbf{y}$.



(b) Join-the-dots model.



(c) Linear model.



(d) True model.

Simple example

20 observations from

$$Y = X^3 + \varepsilon$$

where

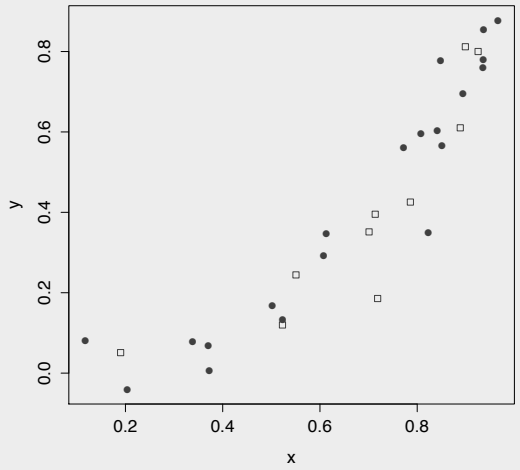
$$\varepsilon \sim N(0, 0.0064)$$

Get 10 **new** (test) observations and calculate

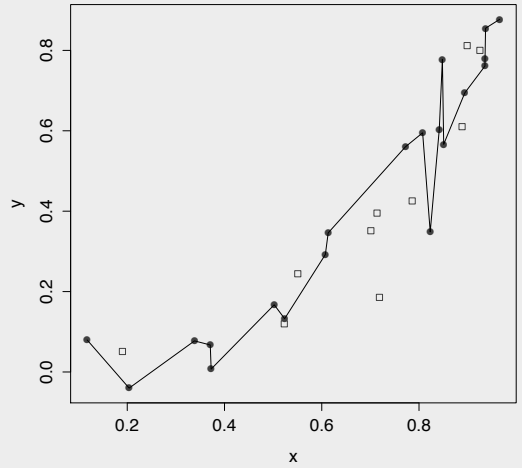
$$\hat{R}_g = \frac{1}{n}\sum_{i=1}^{10}(y_i^{new} - \hat{y}_i^{new})^2,$$

mean squared error (MSE)

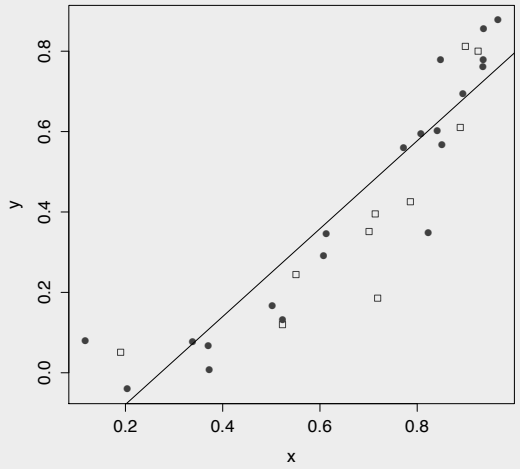
	R_t	$\hat{R}_g$
"Join-the-dots"	0	0.0181
Linear	0.0117	0.0206
True	0.0064	0.0063



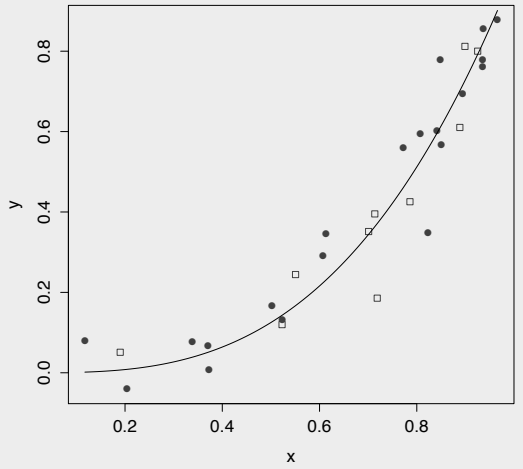
(a) Scatterplot of **x** versus **y**.



(b) Join-the-dots model.



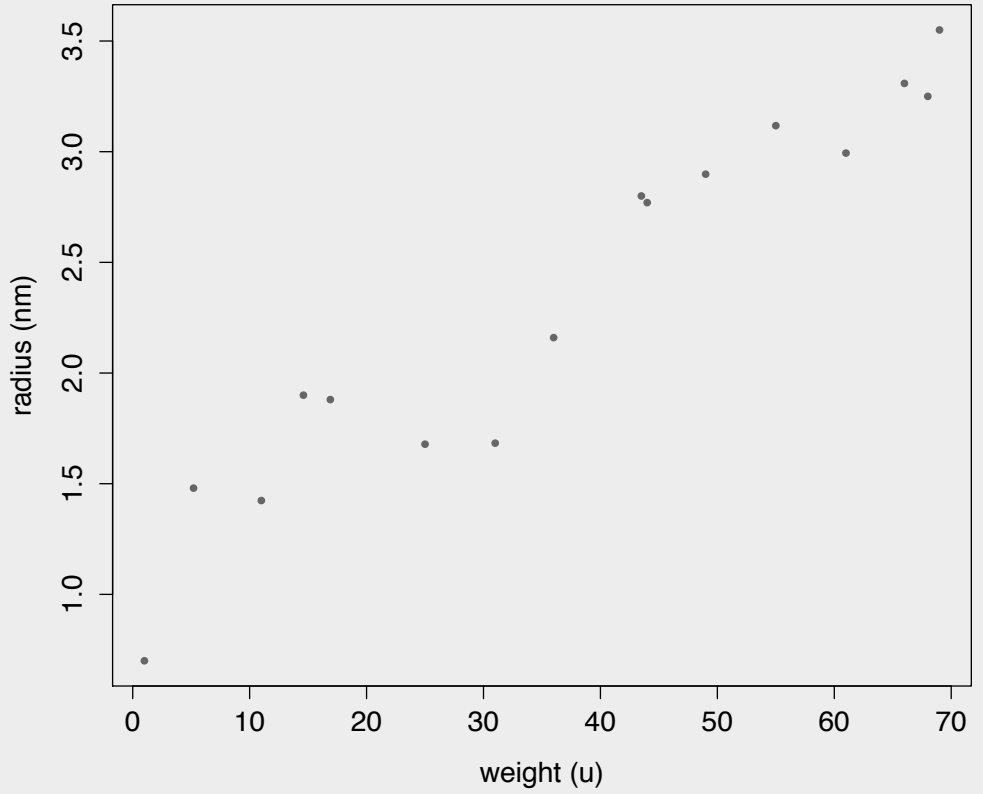
(c) Linear model.

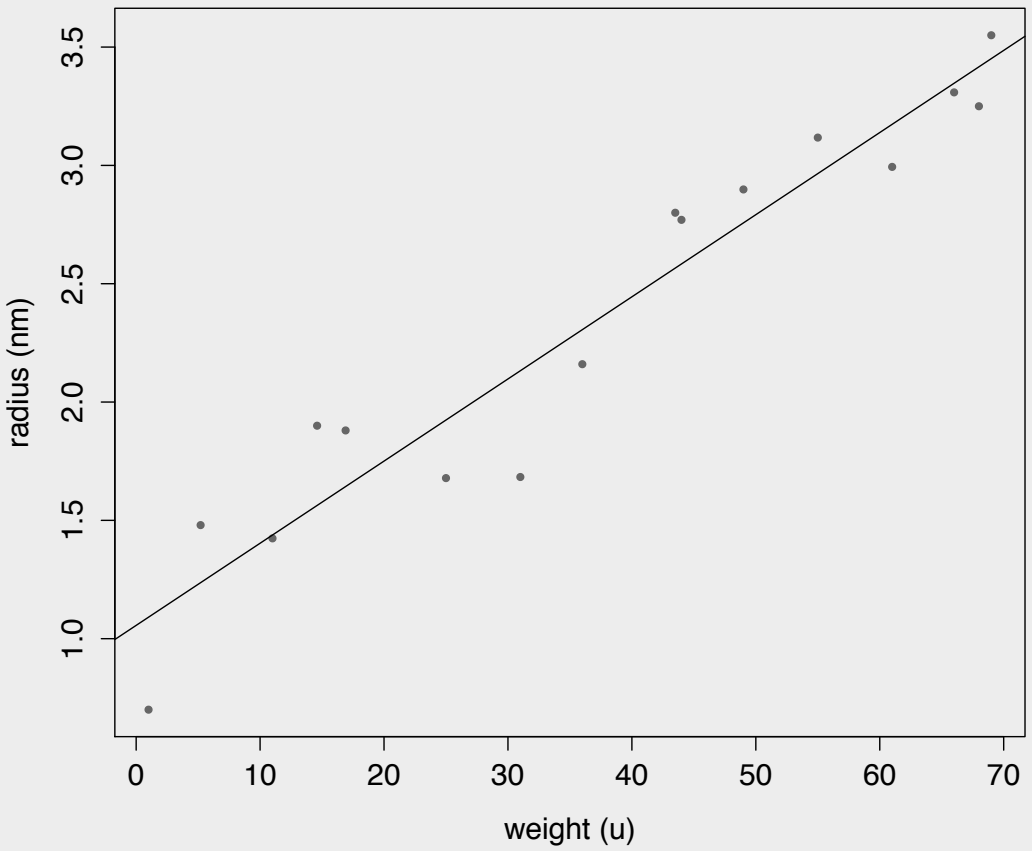


(d) True model.

Linear supervised methods: Linear regression

Protein	Weight (u)	Radius (nm)
Insulin	5.2	1.48
Lysozyme	14.6	1.90
Myoglobin	16.9	1.88
Lactoglobulin	36.0	2.16
Albumin	43.5	2.80
Bence Jones protein	44.0	2.77
Hemoglobin	68.0	3.25
Albumin	69.0	3.55
⋮	⋮	⋮





Ordinary least squares (OLS)

estimate the parameters of the model through minimising the residual sum of squares (RSS). In this case we only have an intercept (β_0) and slope (β_1).

```
radius = 1.056 + 0.034 x weight
```

To minimise RSS: differentiate with respect to β and set the first derivative equal to zero

$$RSS(\beta) = \sum_{i=1}^{n}(y_i - f((x)_i))^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$$

Problems with OLS

- variance and generalisation (test) error very large if have many variables (p) relative to number of observations (n)
  - this is often the case in pharmaceutical bioinformatics (in QSAR common to have 100s or 1000s of variables)
- if p > n cannot compute β

Possible solutions

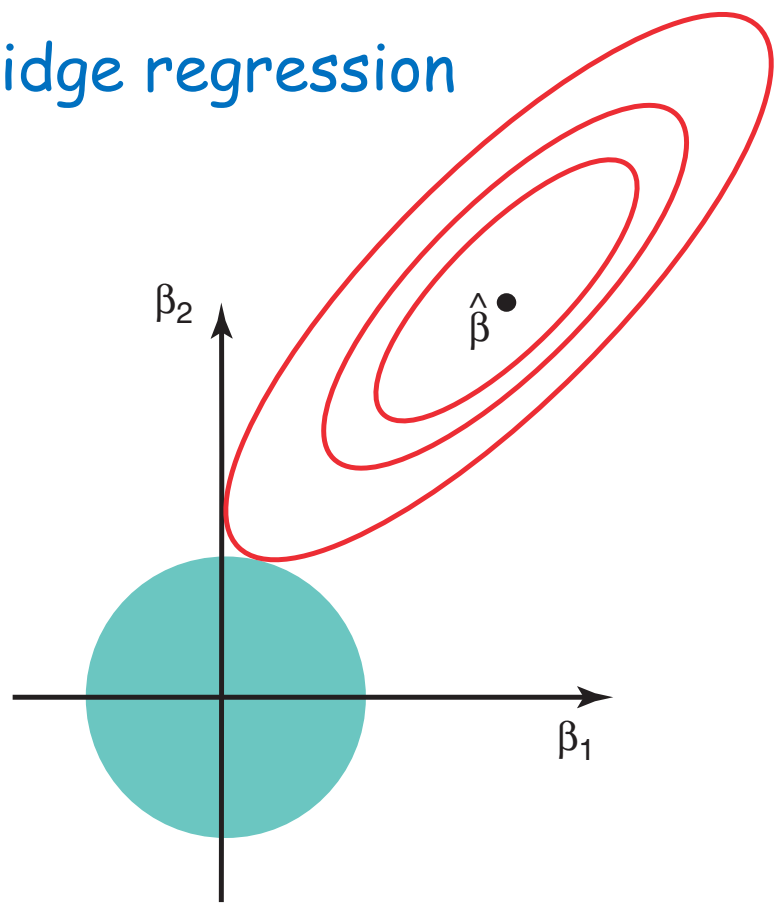
- variable selection
- shrinkage methods
  - ridge regression
  - lasso
- partial least squares (PLS)

$$PRSS(\beta) = RSS(\beta) + \lambda J(\beta)$$

Shrinkage methods

$$PRSS(\beta) = RSS(\beta) + \lambda J(\beta)$$

ridge regression



lasso

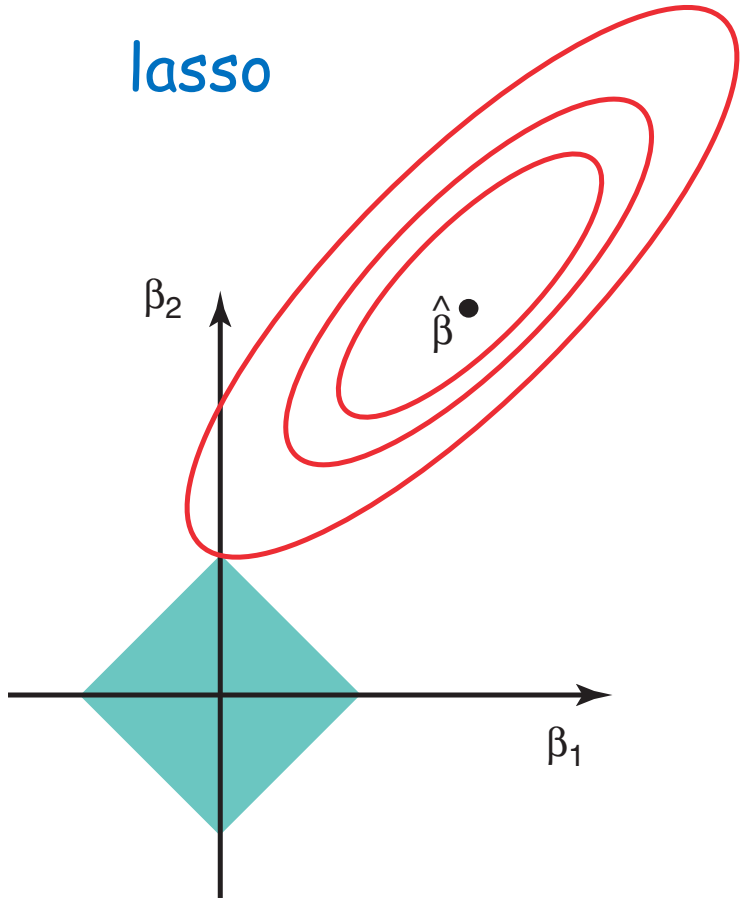


Figure from "An Introduction to Statistical Learning with Applications in R" (ISLR in later slides), page 222.

Partial least squares (PLS)

- variables often highly correlated
  - genes active in same biological pathway
  - molecular descriptors reflecting same quality (e.g. weight, surface area & radius)
- PLS is a dimension reduction method: identifies a small number of linear combinations (or directions) of variables ("features" in machine learning lingo) and uses these instead in the regression model
  - In PLS the response Y supervises the identification of the PLS directions (compare this with Principal Components Analysis (PCA) in tomorrow's lecture)

PLS method

1. standardise features (PLS not scale invariant)

2. first direction Z_1 derived from basic linear regression of Y against each feature separately (hence highest weight on features most strongly related to the response)

3. regress each feature on Z_1 and take residual

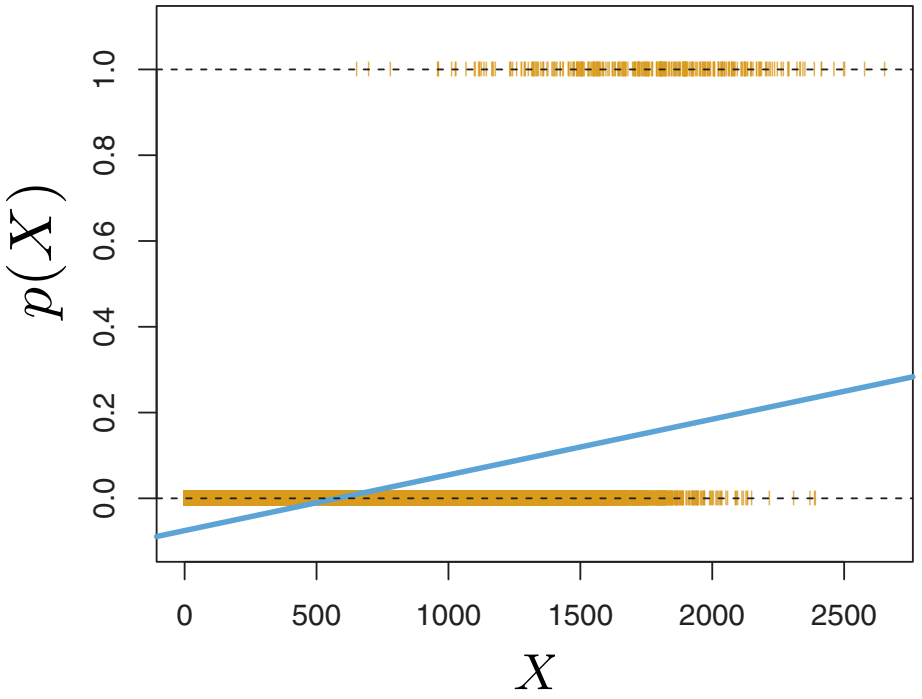
4. compute Z_2 as Z_1 based on these residuals (the *orthogonalised* data)

5. repeat M times to get $Z_1,\ldots,Z_M$ (where $M < p$)

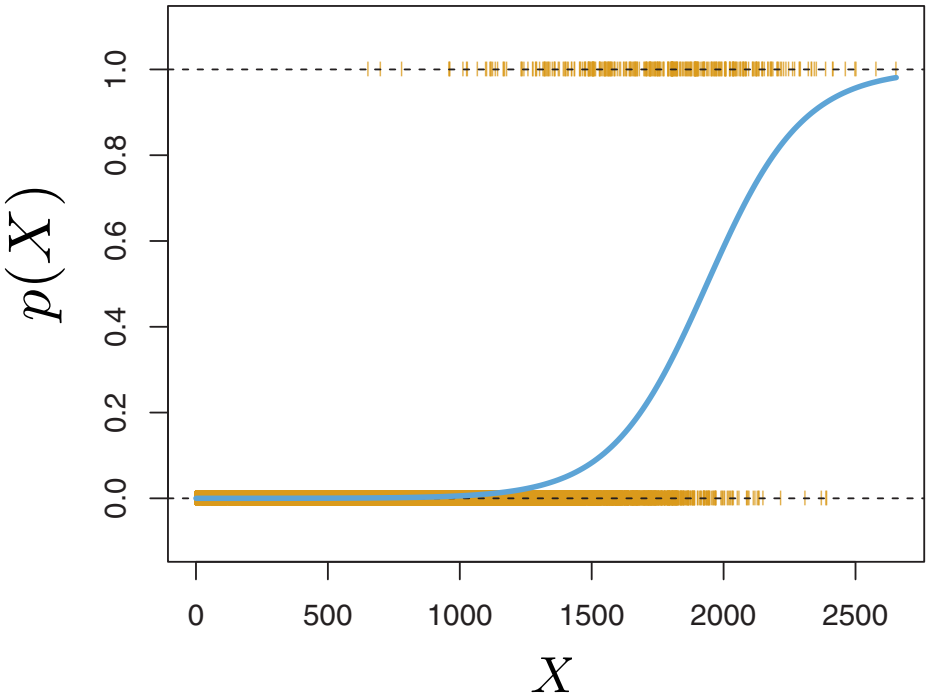
Linear classification: Logistic regression

$$p(X) = \Pr(Y = 1|X)$$

$$p(X) = \beta_0 + \beta_1 X$$



$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$



Figures modified from ISLR page 131.

Linear model assumptions

- linearity ☺
- observations are independent
- variance of errors independent of feature values
- no measurement errors

Nonlinear supervised methods

- Random forests (RFs)
  - an ensemble of decision trees
- Support Vector Machines (SVMs)
  - hyperplanes with maximal margins
- Artificial Neural Networks (ANNs)
  - simulates the structure of the brain
  - the topic of tomorrow's lecture

Decision trees for classification

Lipinski Rule of Five

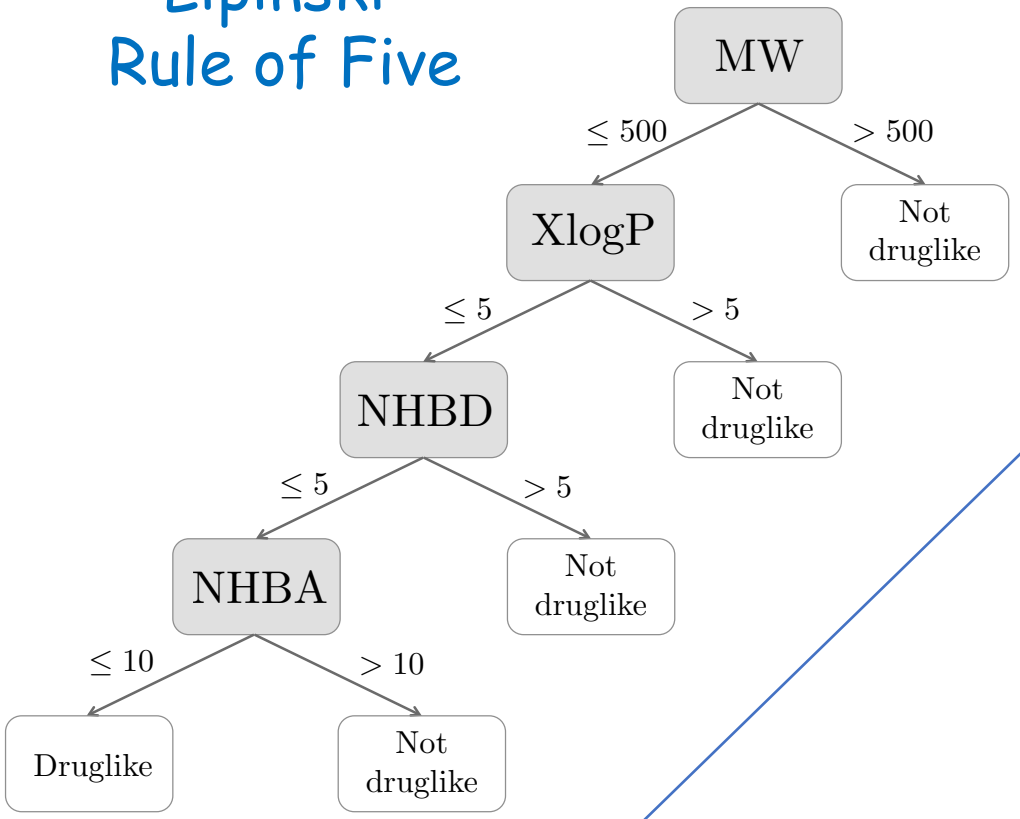
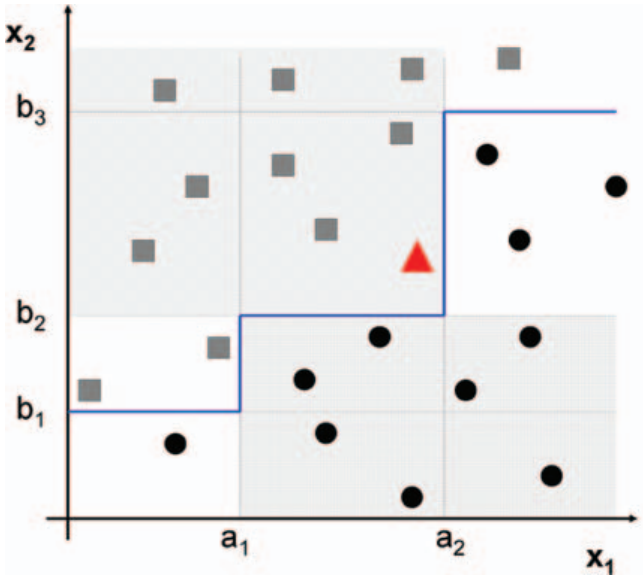


Illustration of a decision tree decision boundary



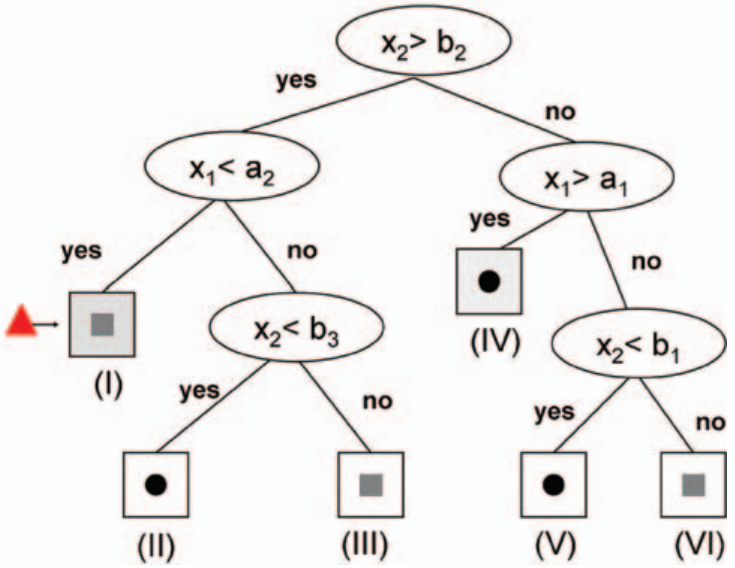
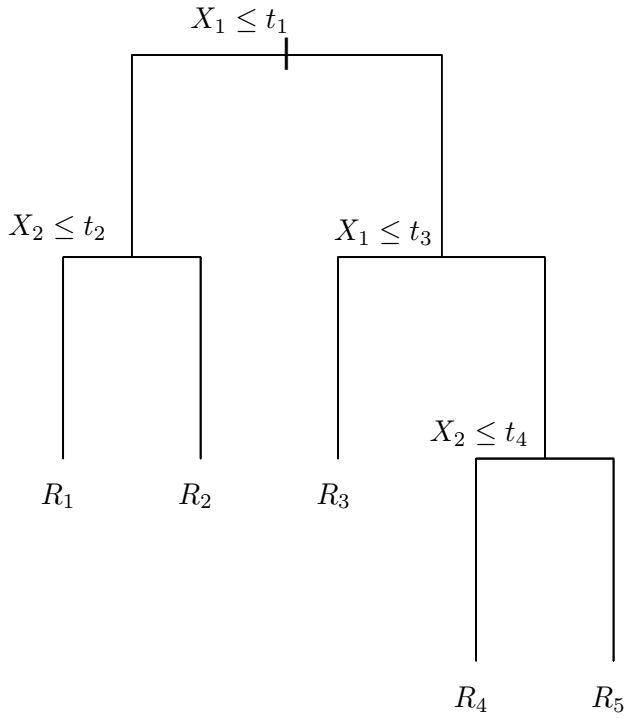
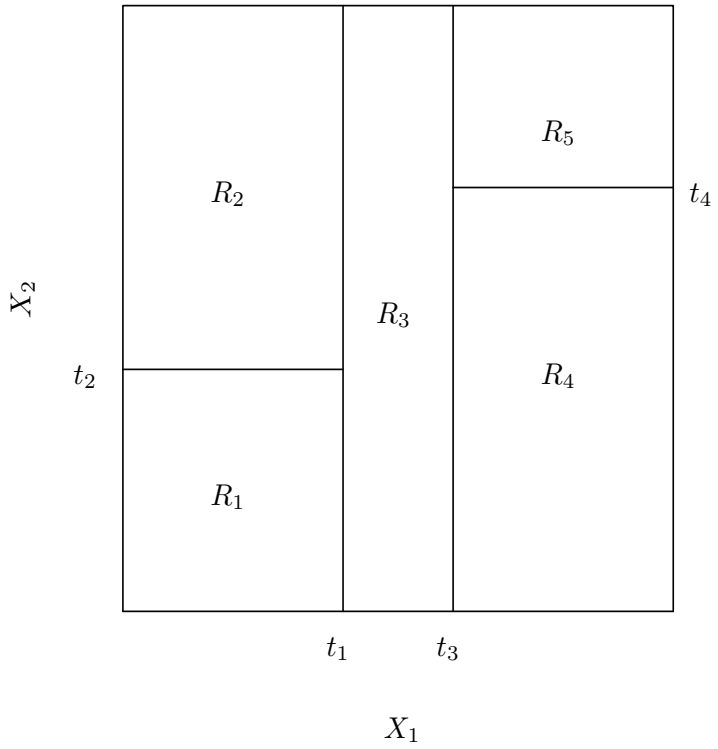
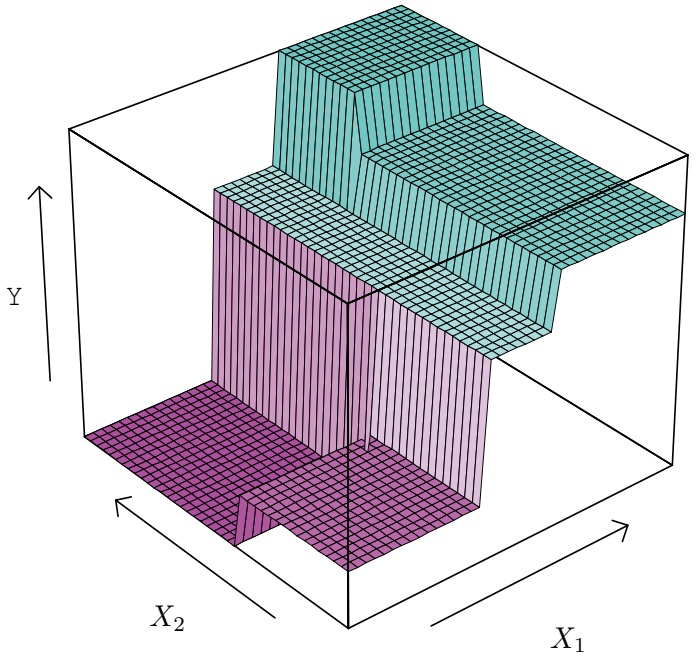


Figure from Tarca et al (2007) "Machine learning and its applications to biology"

Decision trees for regression







Figures from ISLR page 308

Growing a random forest

1. Draw n samples, with replacement, from the dataset and label dataset*

2. Fit a decision tree to dataset*
   for each split of the decision tree take a random sample of size q from the features (split can then only use one (the best) of those features)

3. Repeat (1) & (2) B times.

B = 500 common; as is
q = sqrt(p) – for classification
q = p/3 - for regression

Support Vector Machines (SVMs)

Illustration of a maximal margin hyperplane with *hard margins…*

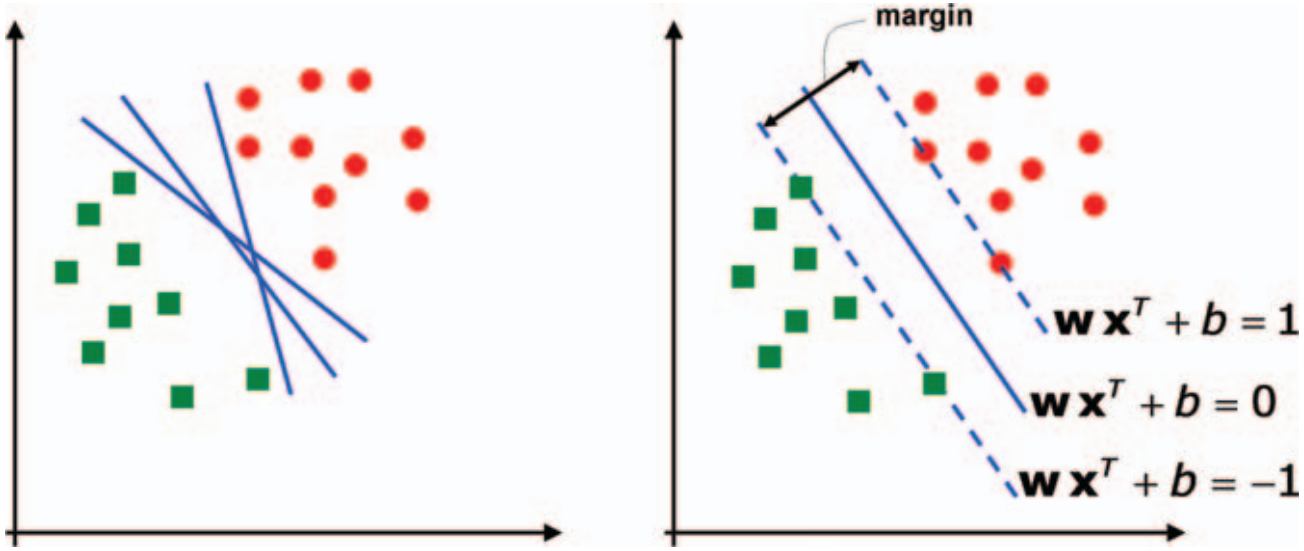
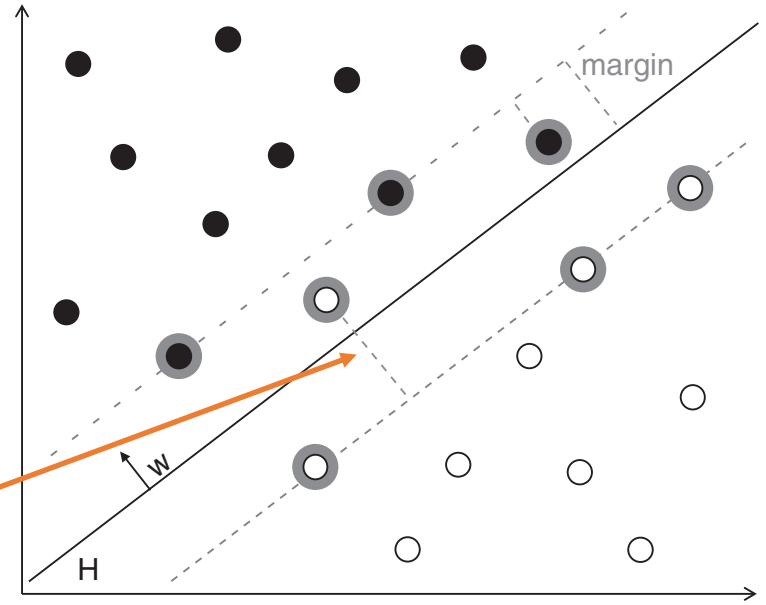


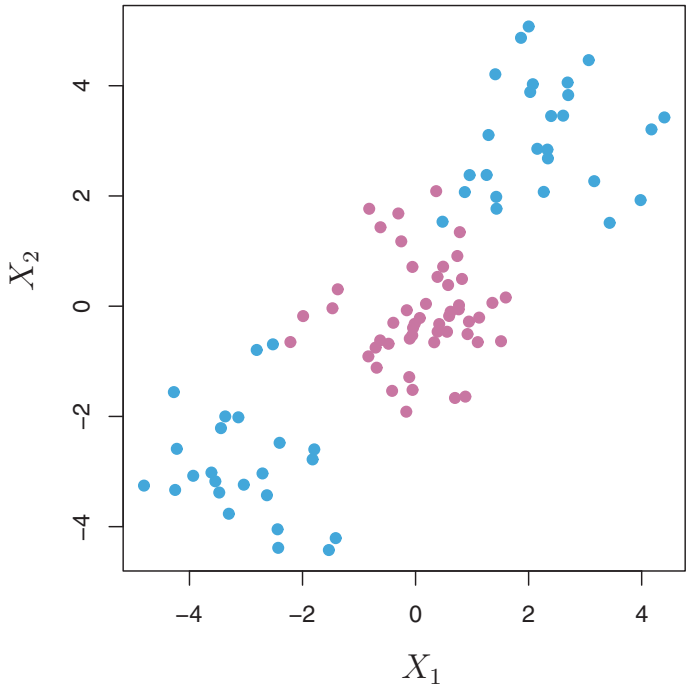
Figure from Tarca et al (2007)

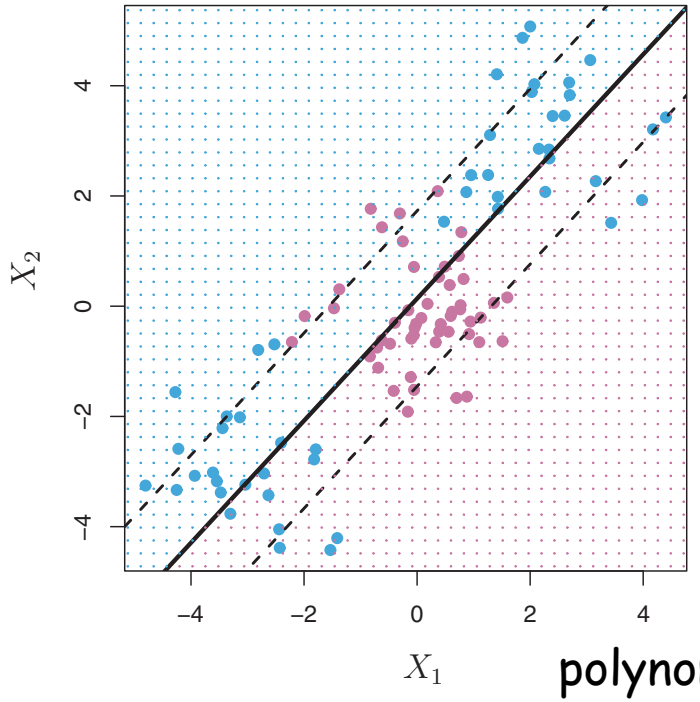
… and *soft margins*



a cost parameter, C, controls how big these errors (slacks) can get
large C = high penalty for errors

Figure from Heikamp & Bajorath (2014)
"Support vector machines for drug discovery"





support vector classifier seeks a linear boundary and performs poorly…

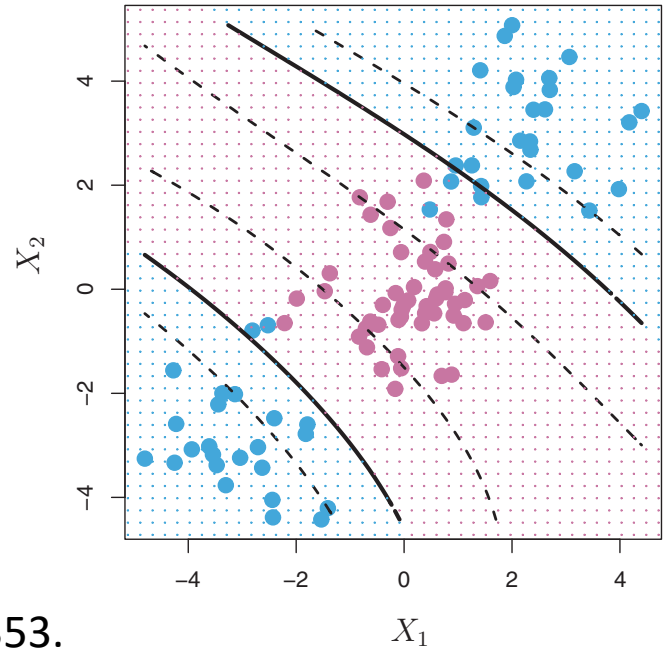
SOLUTION:
Apply the "kernel trick"!

Gaussian (AKA radial basis function) kernel has inverse width parameter γ

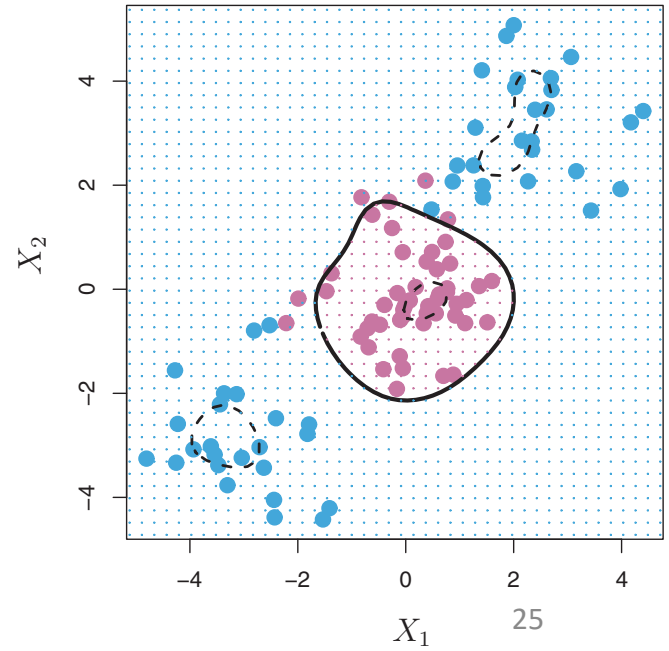
if γ too high risk overfitting
if γ too low get limited accuracy

γ and C are parameters that need to be tuned during model fitting (see later)

polynomial kernel (d=3)



Gaussian kernel



Figures from ISLR pages 349 and 353.

SVM for regression (AKA SVR)

As SVM for classification, SVR uses the concept of margin maximization and the kernel trick for non-linear boundaries (has equivalent slacks, thus C, and e.g. γ if using a Gaussian kernel, that need to be tuned)

Points lying on the edge or outside the ε-tube are the support vectors

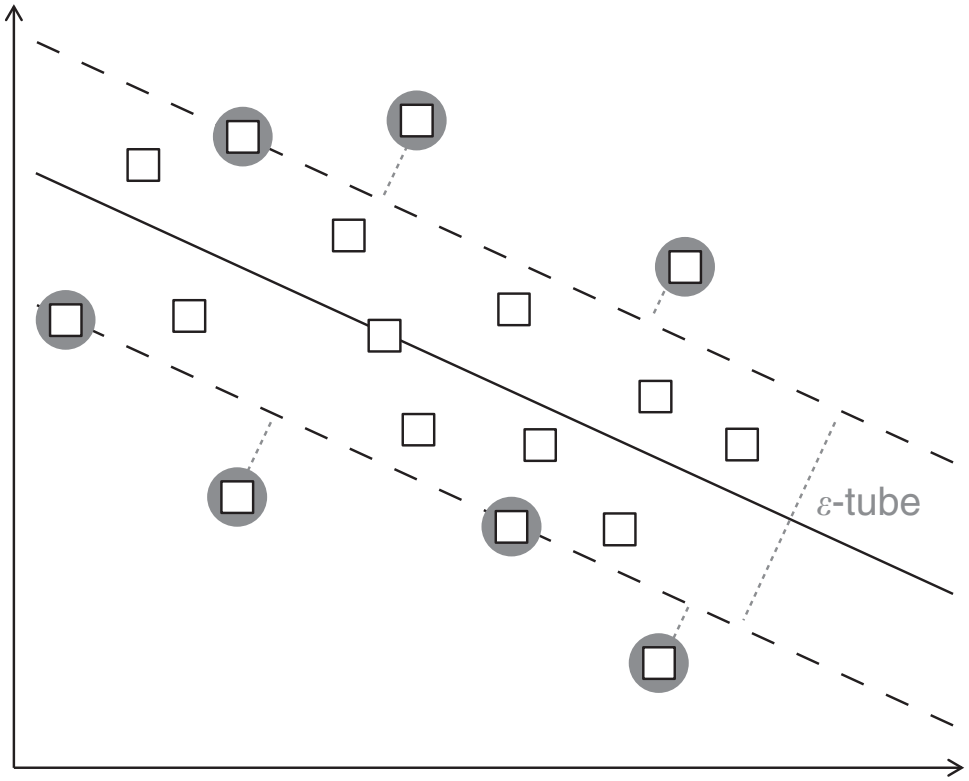
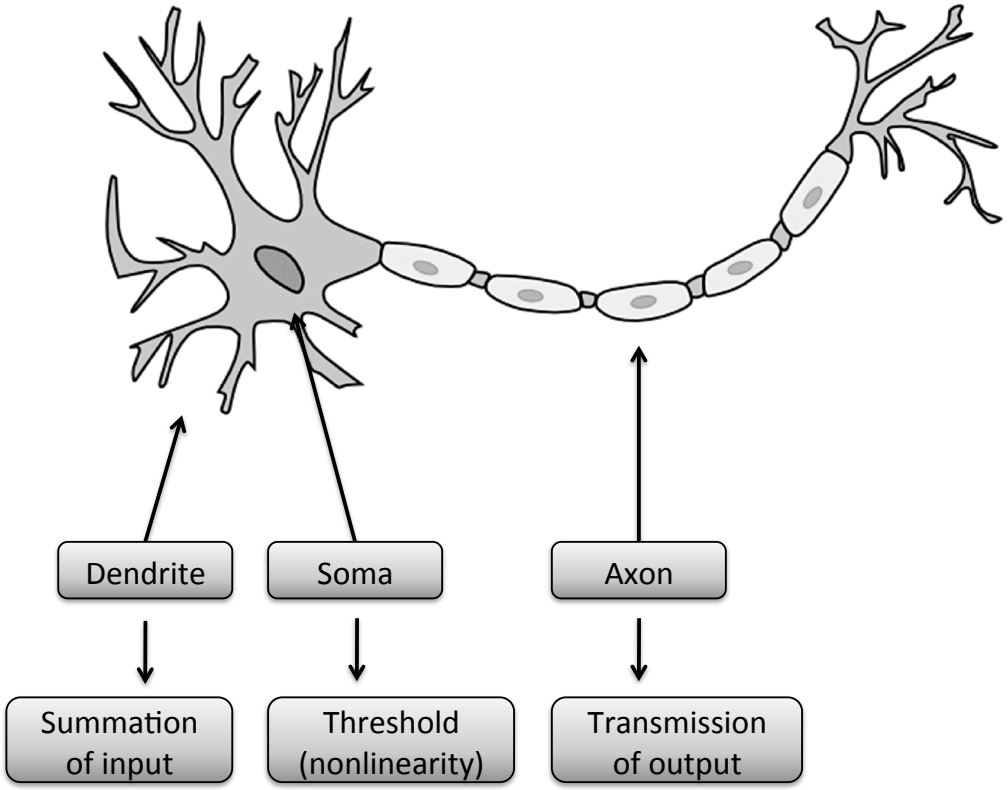


Figure from Heikamp & Bajorath (2014)

Artificial Neural Networks (ANNs)



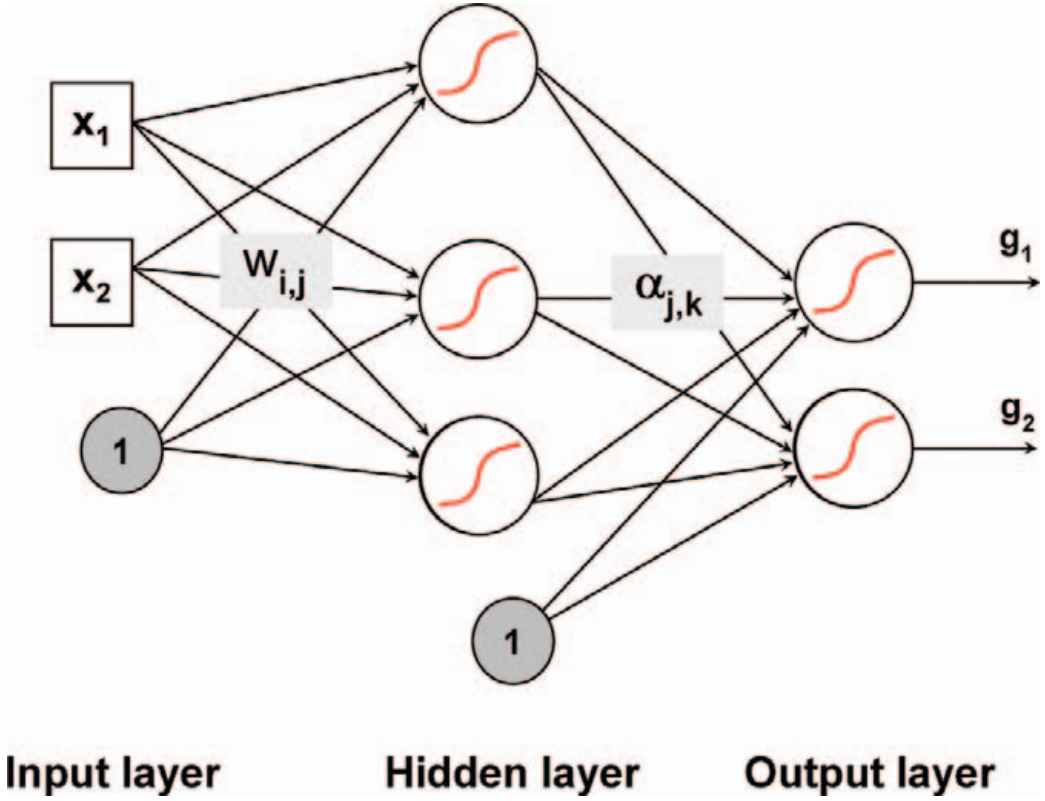
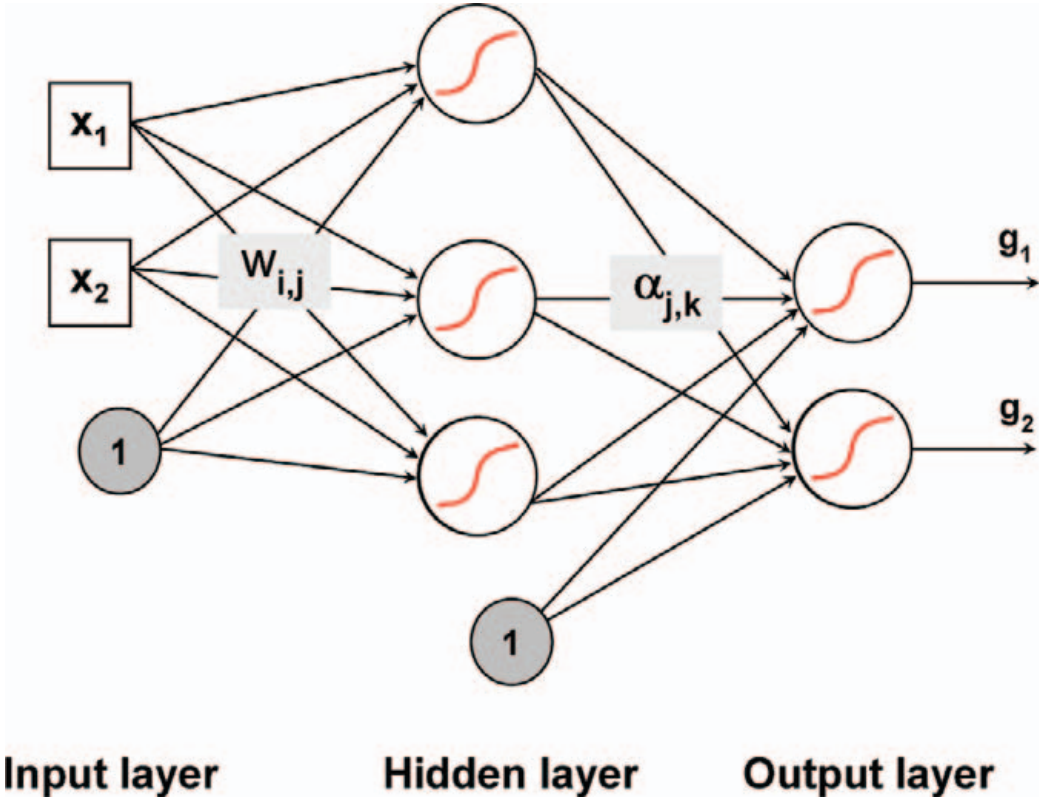


Figure from Tarca et al (2007)

Most common architecture used for classification problems



weights

$$g_k(\mathbf{x}) = \sigma[\sum_{j=1}^{J} \alpha_{j,k}\sigma(\sum_{i=1}^{p} x_i w_{i,j} + b_j^h) + b_k^o]$$

bias terms

$$\sigma(z) = \frac{1}{1 + exp(z)}$$

bias terms are like intercepts that shift the activation threshold of the neuron (or node)

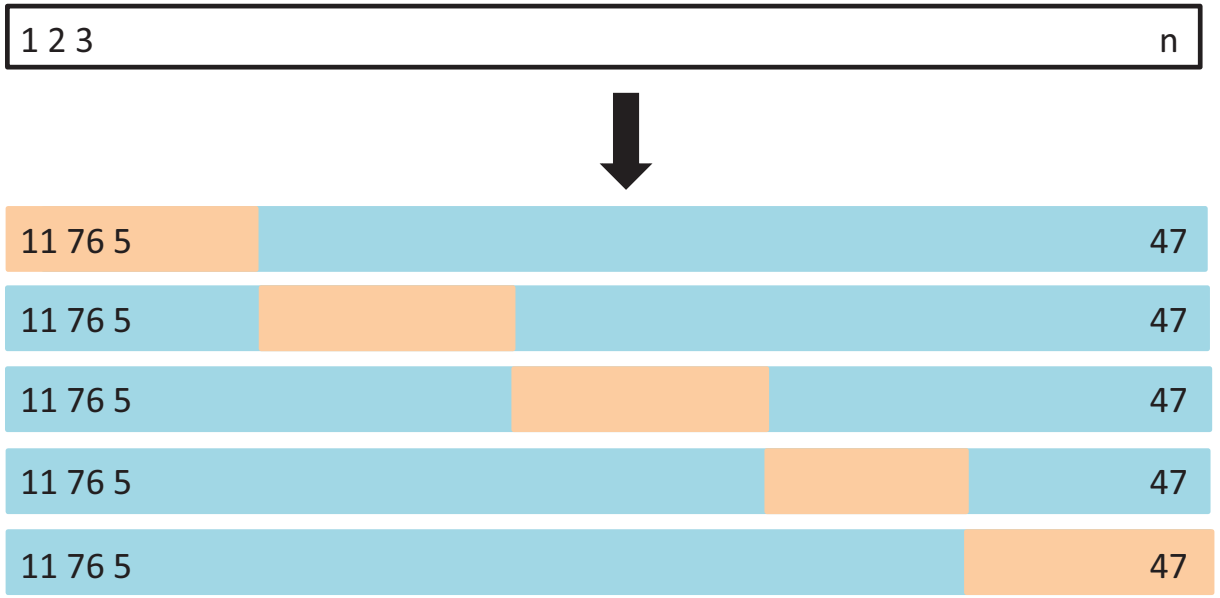
Model choice

- Information criteria
  - likelihood based
  - e.g. AIC or BIC
  - tradeoff between model fit and complexity

Sometimes difficult or impossible to specify likelihood

- Cross-validation (CV)
  - estimate generalisation error from data
  - leave-one-out cross validation (LOOCV)
  - k-fold CV (k = 5 or k = 10 are common choices)

E.g. 5-fold CV

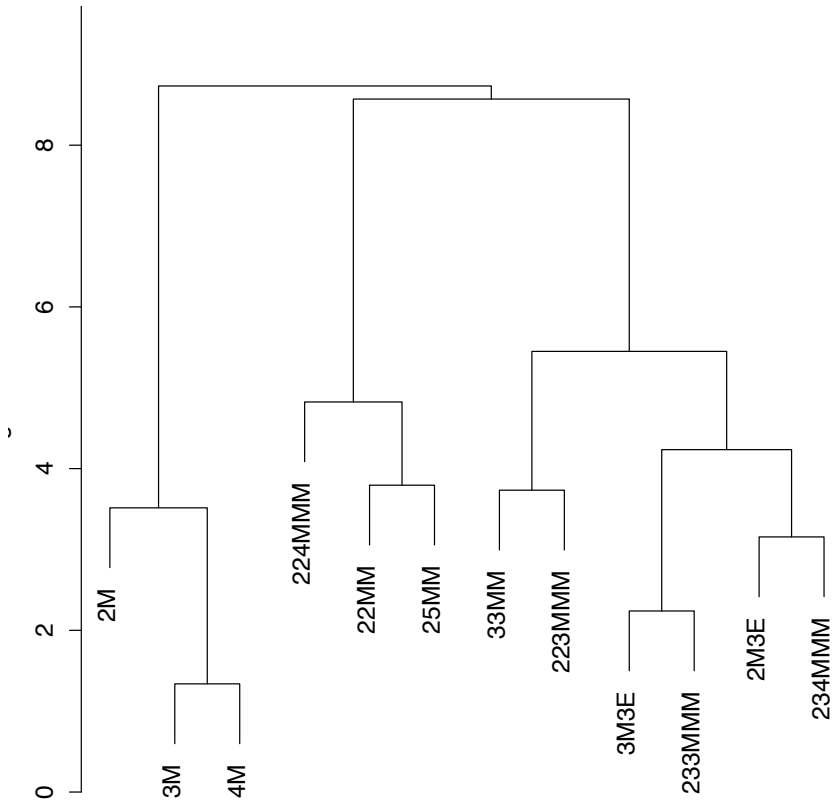


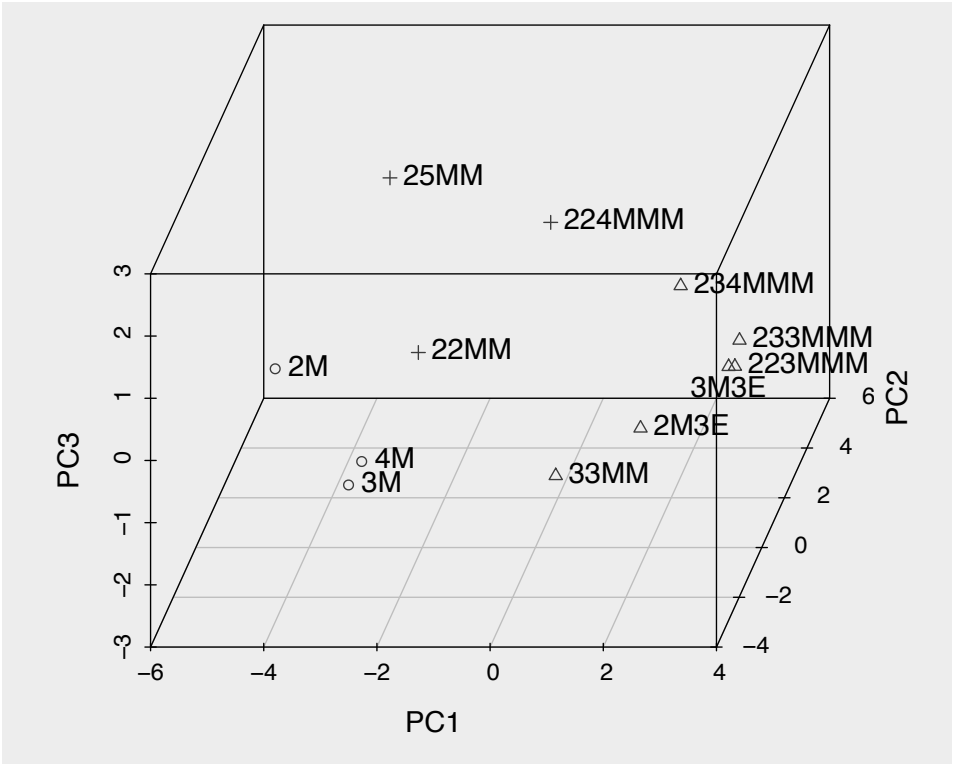
choice of parameter values (e.g. C & γ in the SVMs) is based on minimizing the generalisation error (see earlier slides) across the 5 (orange) held-out subsets

Model assessment

- to assess the model ideally we should keep out a fraction of the data for final testing (e.g. 25% of the data) and compute the generalisation error on this
- do not confuse this with the splitting made for CV purposes
- when not enough data to split into training and testing can do double-loop CV

Part 2: Unsupervised Machine Learning





Preamble

- **supervised machine learning:**
  - observations of features and associated responses
  - fit a model to these and then:
    1. make predictions for new observations
    2. infer relationship between response and features

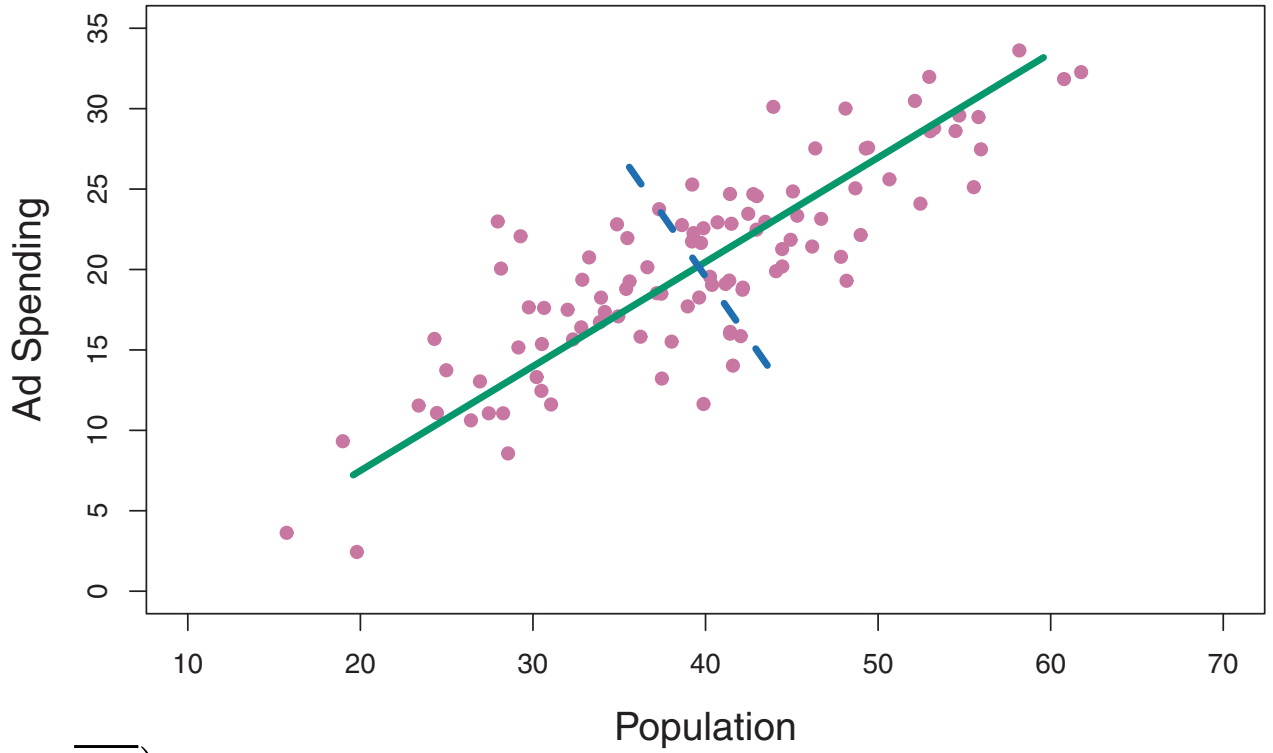
- **unsupervised machine learning:**
  - no response measurements
  - cannot make predictions
  - but perhaps we can:
    1. visualise this potentially high dimensional data
    2. reduce it's dimensions
    3. cluster it into groups

Topics covered

- dimension reduction
  - Principal Components Analysis (PCA)
- clustering
  - hierarchical clustering
  - k-means clustering

PCA

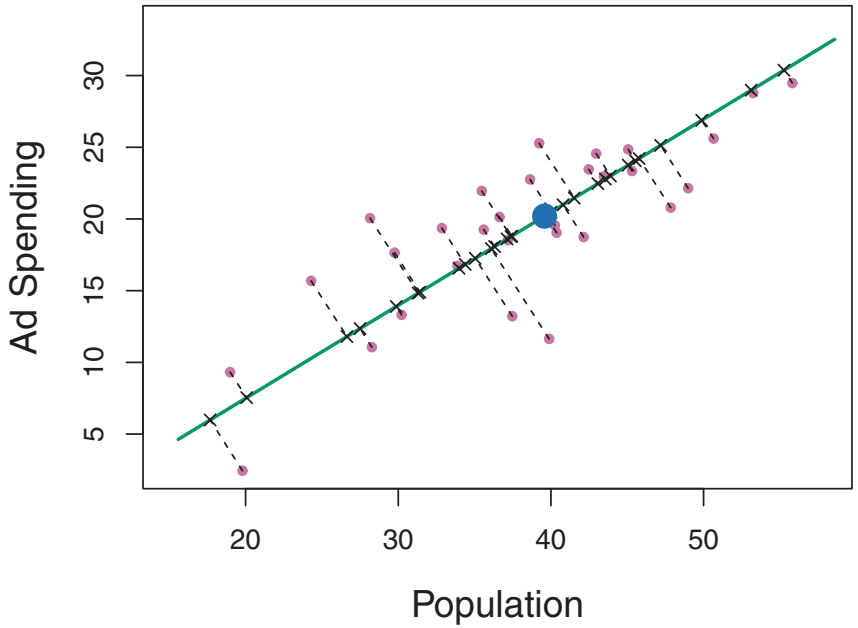
Example from ISLR p230-233:

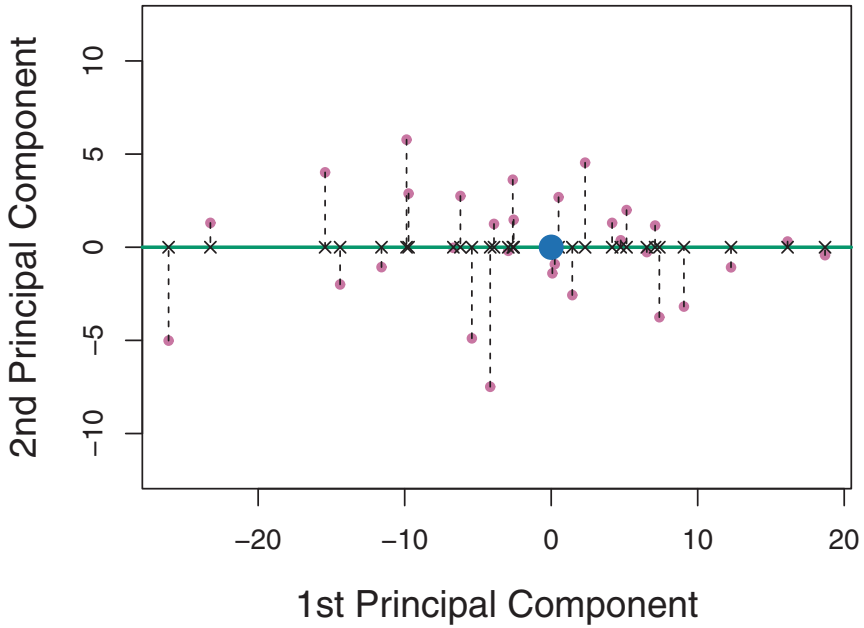


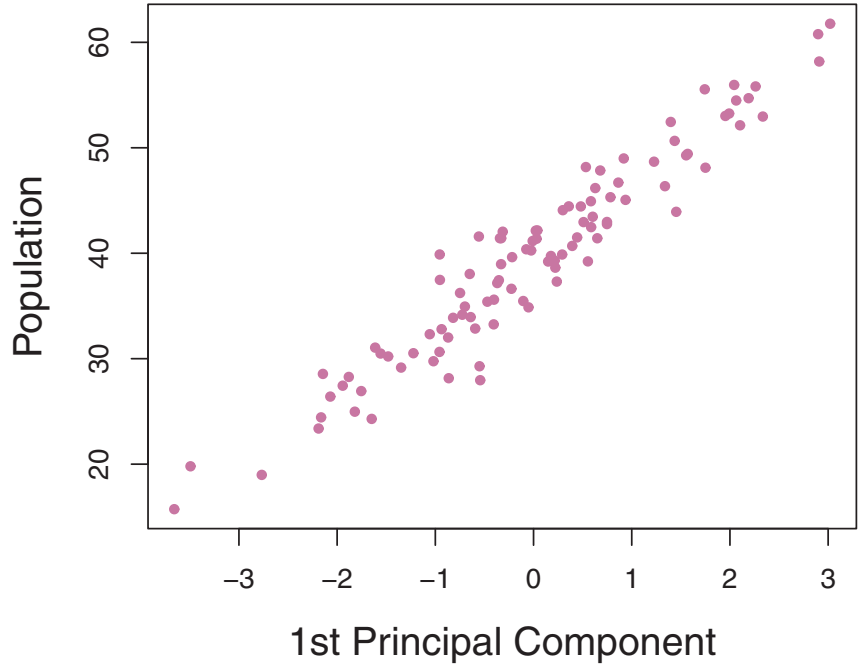
$$Z_1 = 0.839 \times (\texttt{pop} - \overline{\texttt{pop}}) + 0.544 \times (\texttt{ad} - \overline{\texttt{ad}})$$

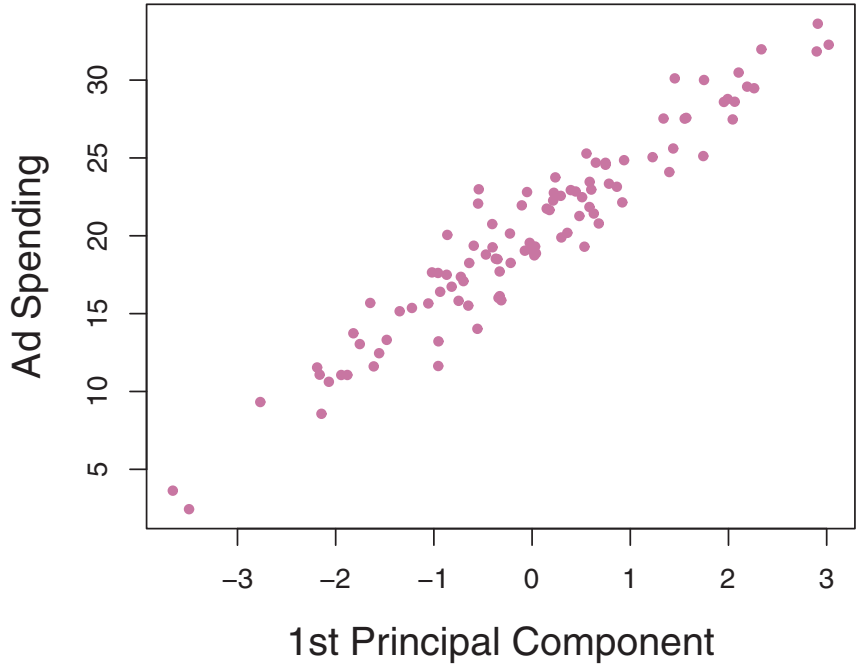
$\phi_{11} = 0.839$ and $\phi_{21} = 0.544$ are the principal component loadings

$\text{Var}(\phi_{11} \times (\texttt{pop} - \overline{\texttt{pop}}) + \phi_{21} \times (\texttt{ad} - \overline{\texttt{ad}}))$ is maximized such that $\phi_{11}^2 + \phi_{21}^2 = 1$

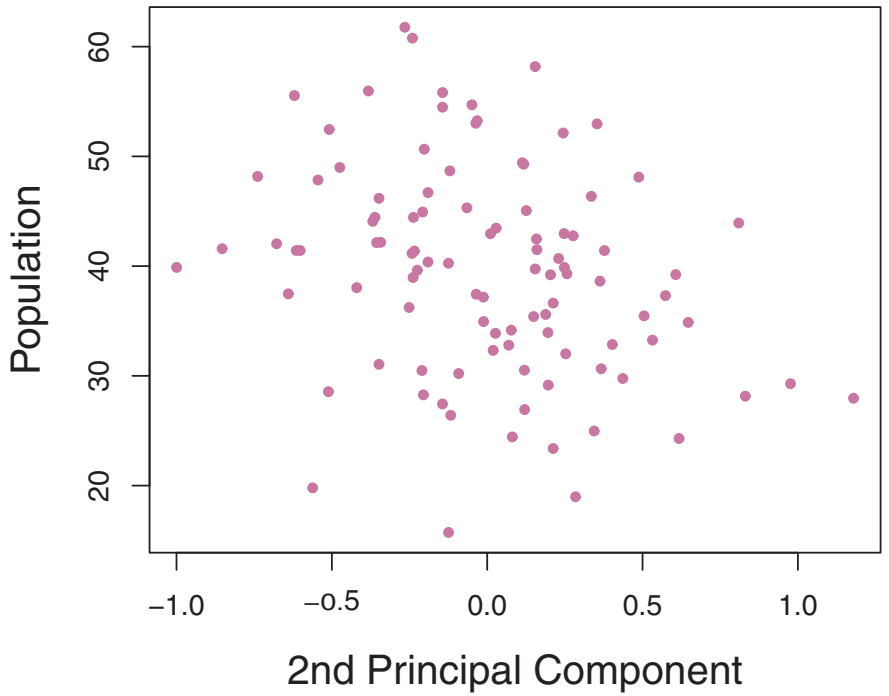

Ad Spending
Population
5
10
15
20
25
30
20
30
40
50

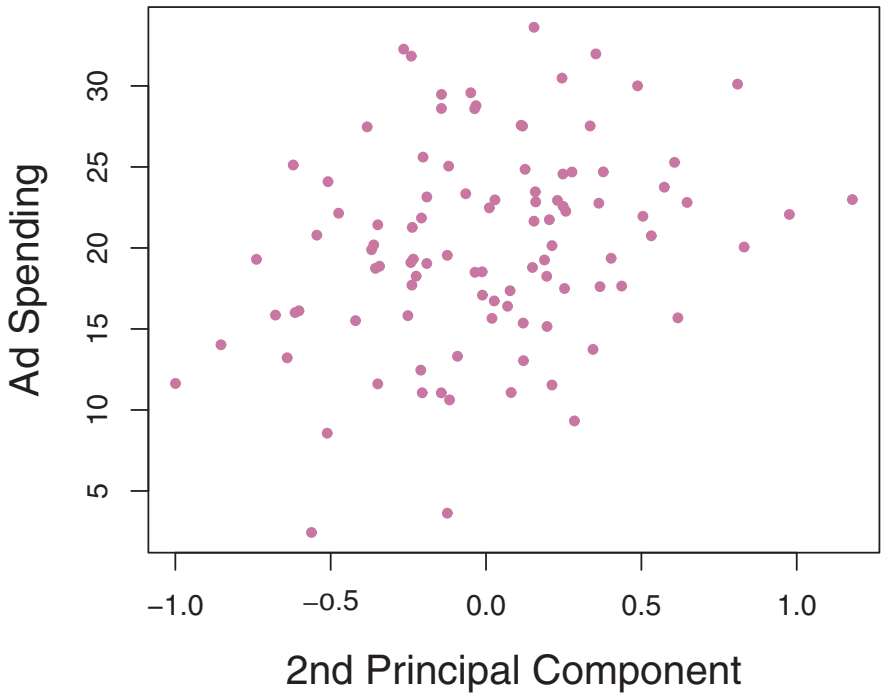

2nd Principal Component
1st Principal Component
−10
−5
0
5
10
−20
−10
0
10
20


Population
1st Principal Component
20
30
40
50
60
−3
−2
−1
0
1
2
3


Ad Spending
1st Principal Component
5
10
15
20
25
30
−3
−2
−1
0
1
2
3

$$Z_2 = 0.544 \times (\texttt{pop} - \overline{\texttt{pop}}) - 0.839 \times (\texttt{ad} - \overline{\texttt{ad}})$$

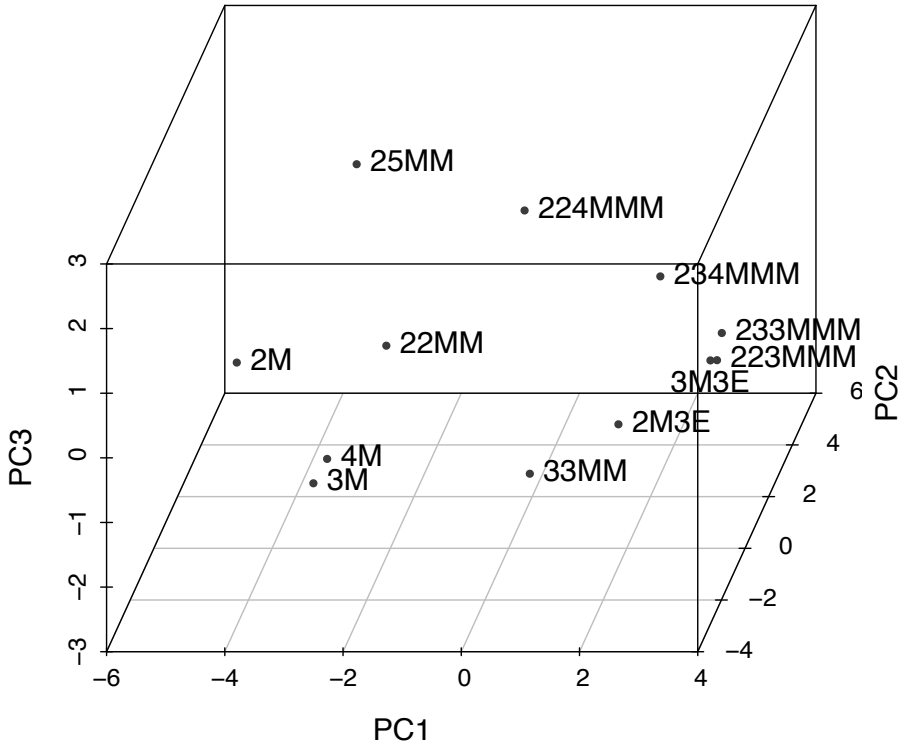




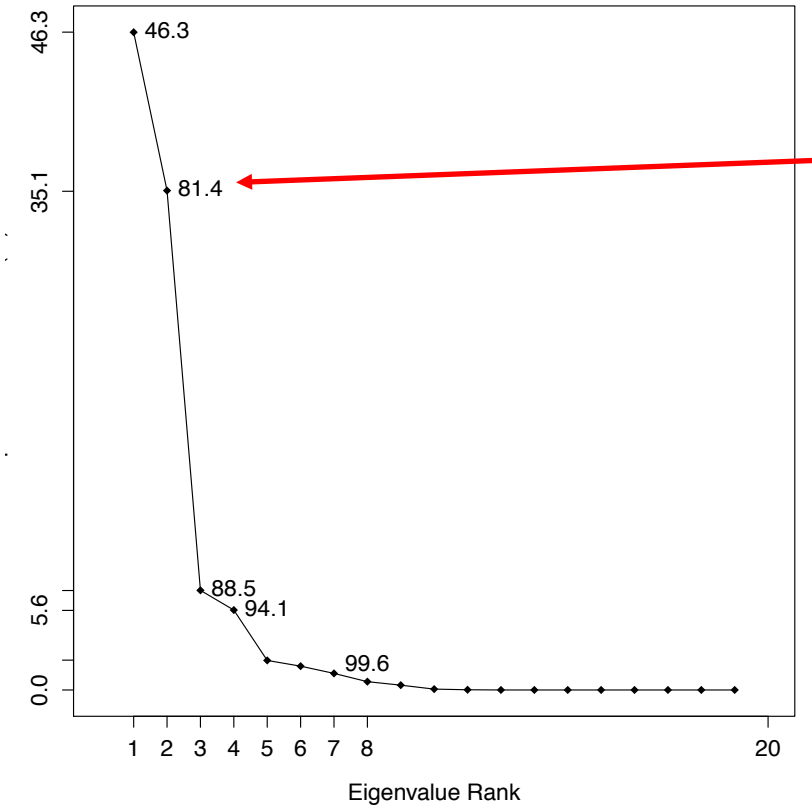
PCA example

PCA applied to 12 different N,N-dimethyl-2-Br-phenetyl-amines using 20 different descriptors (*e.g.* molecular weight and logP)

3D scatter plot



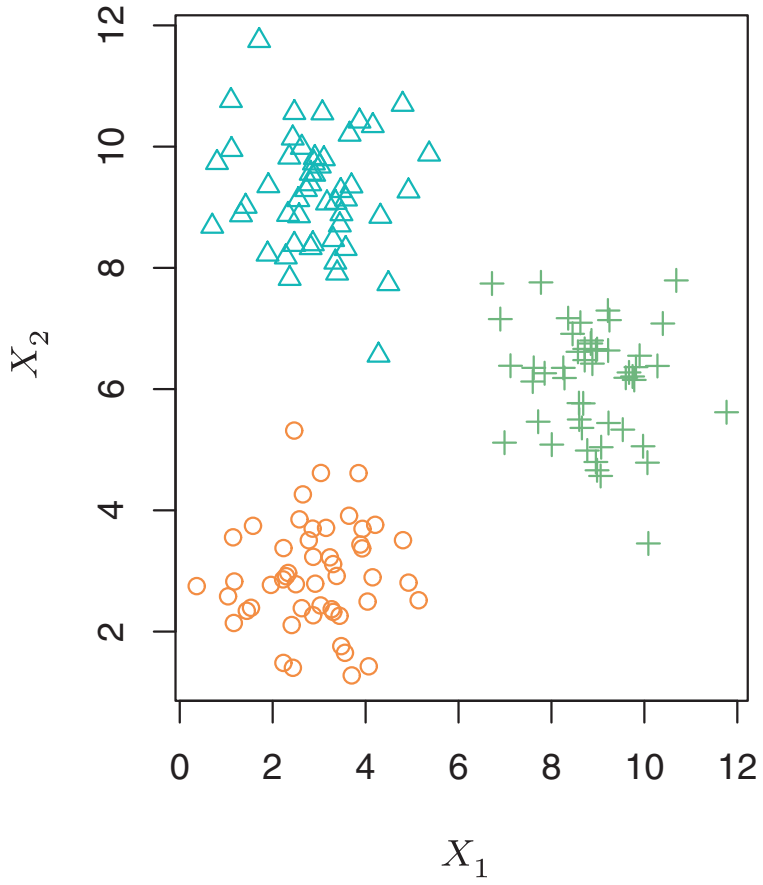
Screeplot



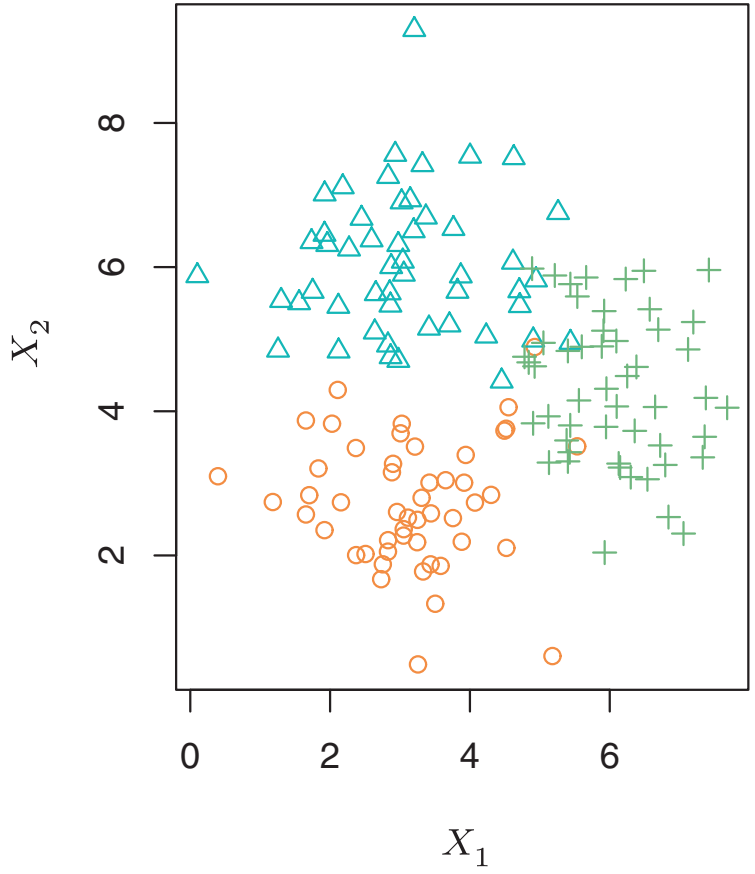
81.4% of variance explained by first two principal components

Clustering

well-separated



overlapping



Figures from ISLR page 27.

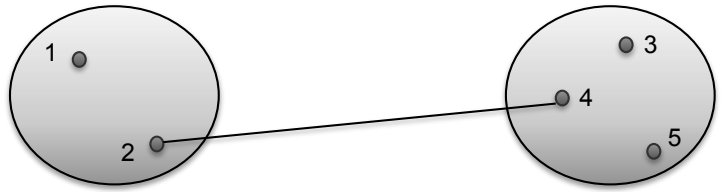
Hierarchical clustering

1. Begin with n observations and a measure (e.g. Euclidean distance) of all pairwise dissimilarities. Treat each observation as its own cluster.

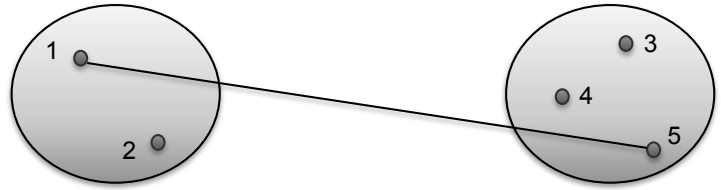
2. For i = n, n–1,…, 2:

   a. Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (i.e. most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates height in the dendrogram (see next slide) at which the fusion should be placed.

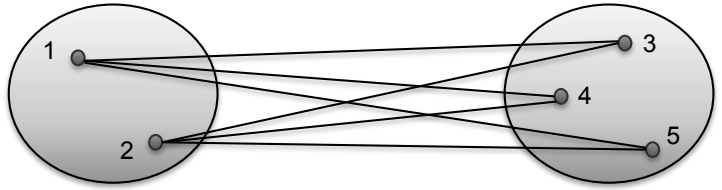
   b. Compute the new pairwise inter-cluster dissimilarities among the i – 1 remaining clusters.



(a) Single linkage.

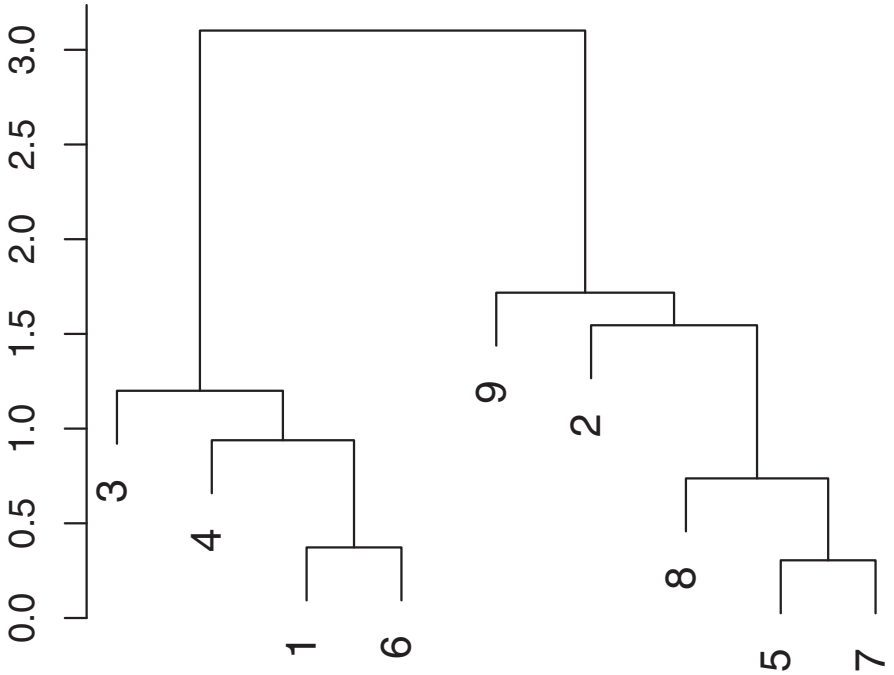


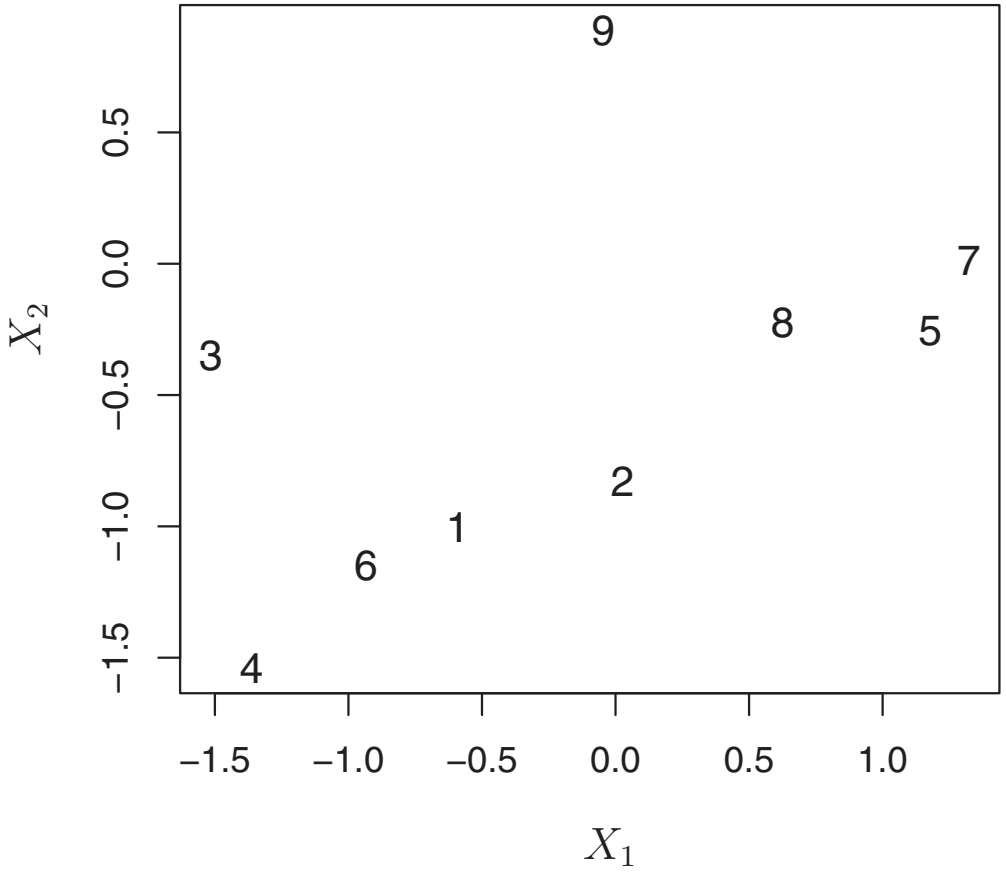
(b) Complete linkage.



(c) Average linkage.

Dendogram interpretation





Figures from ISLR page 393.

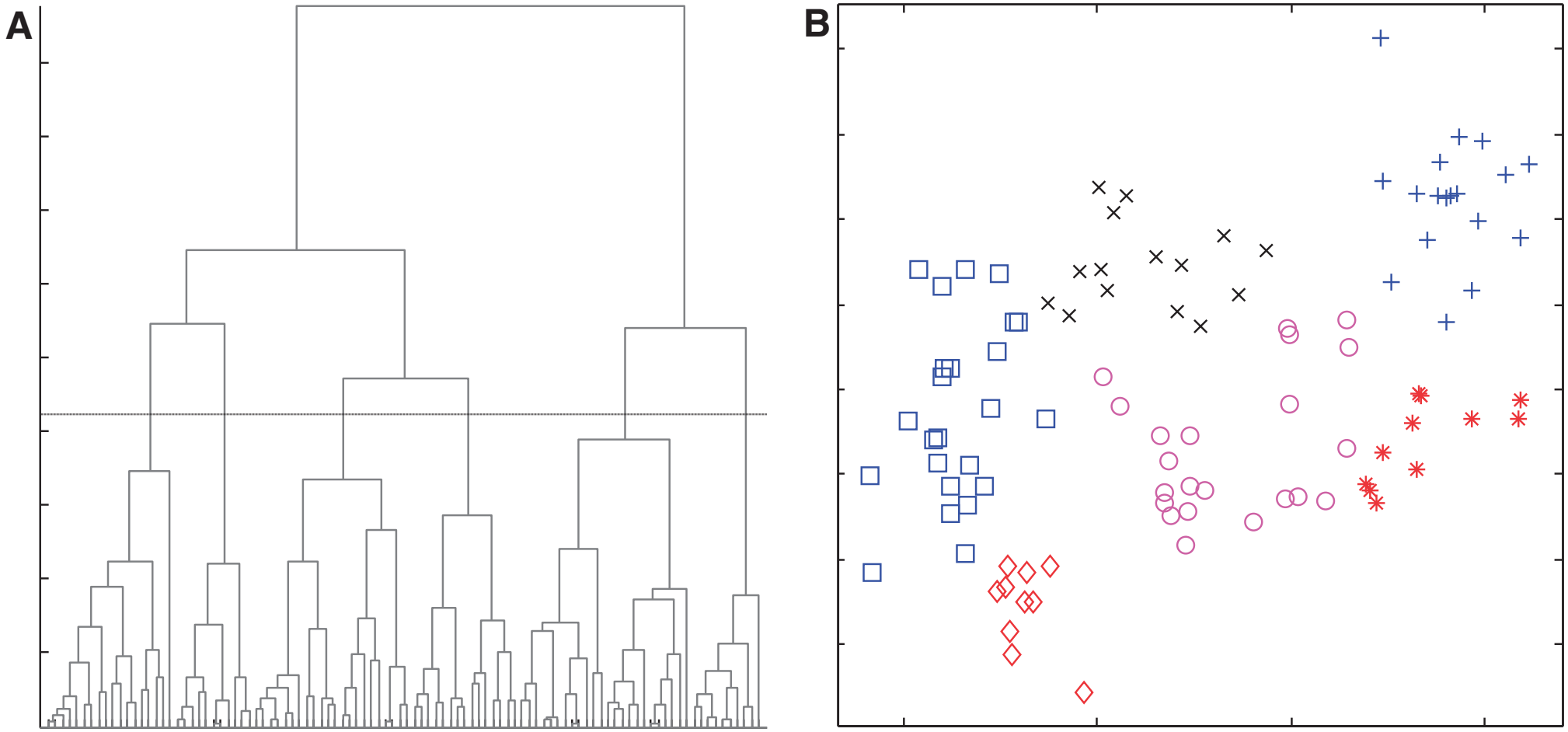


Figure from De Ridder et al (2013) “Pattern recognition in bioinformatics”.

Where to cut?

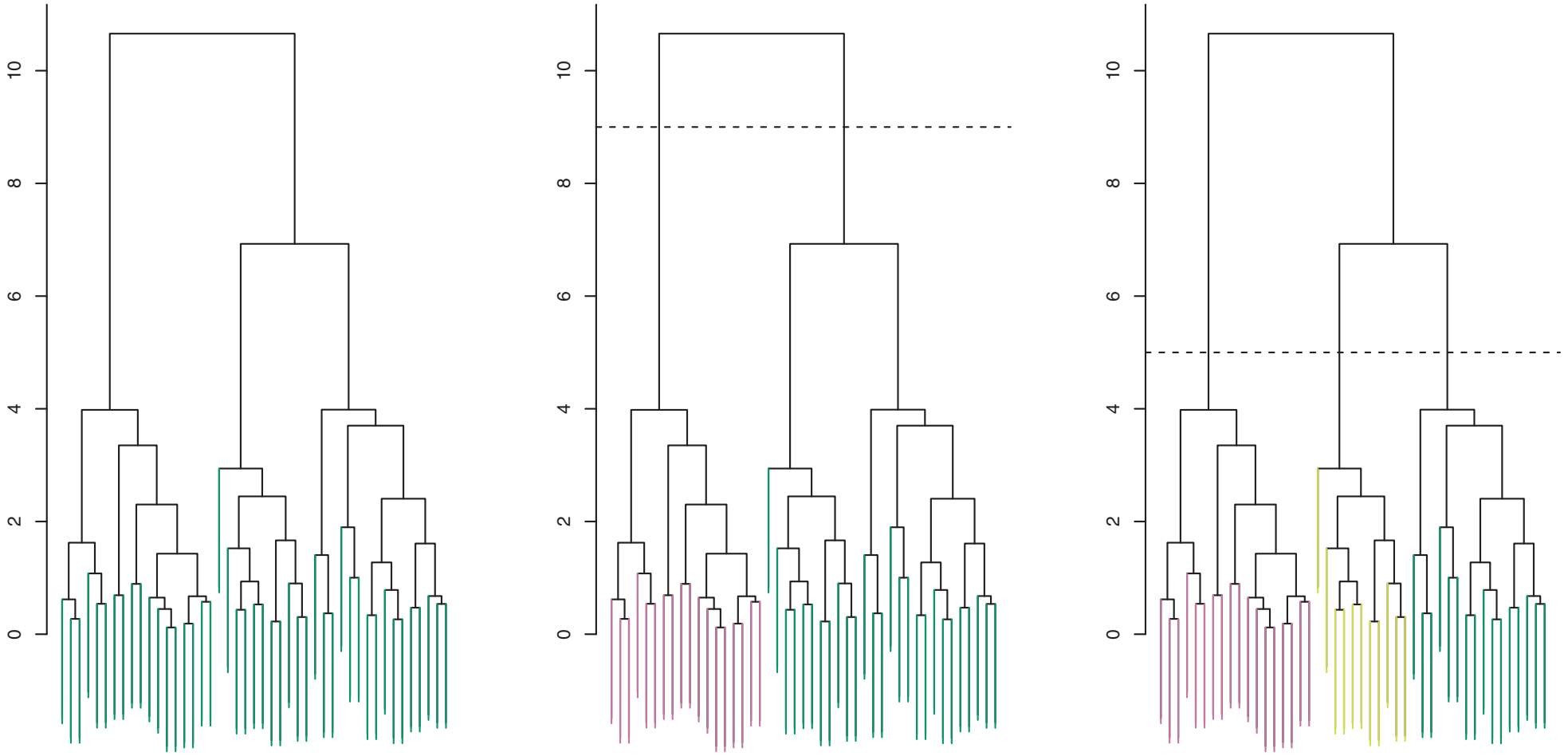
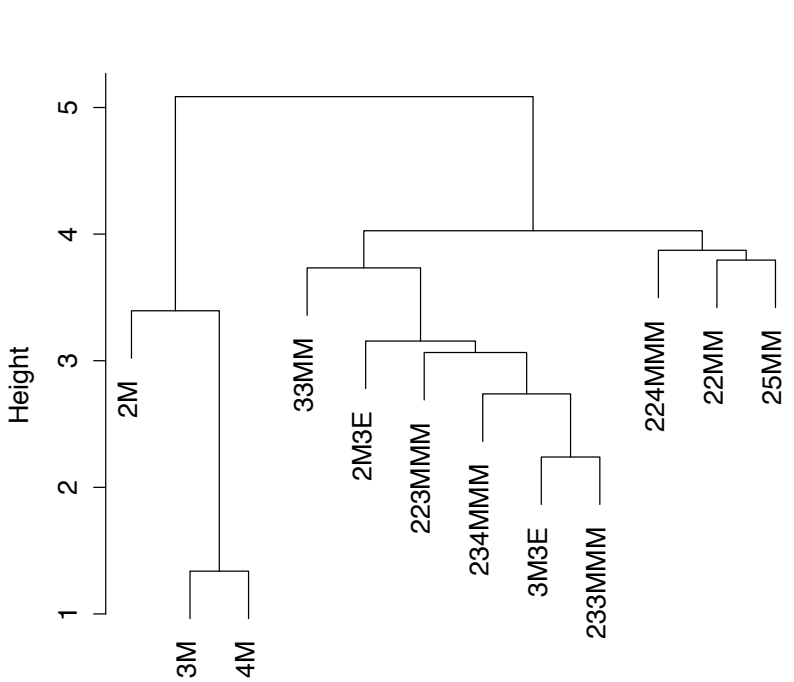
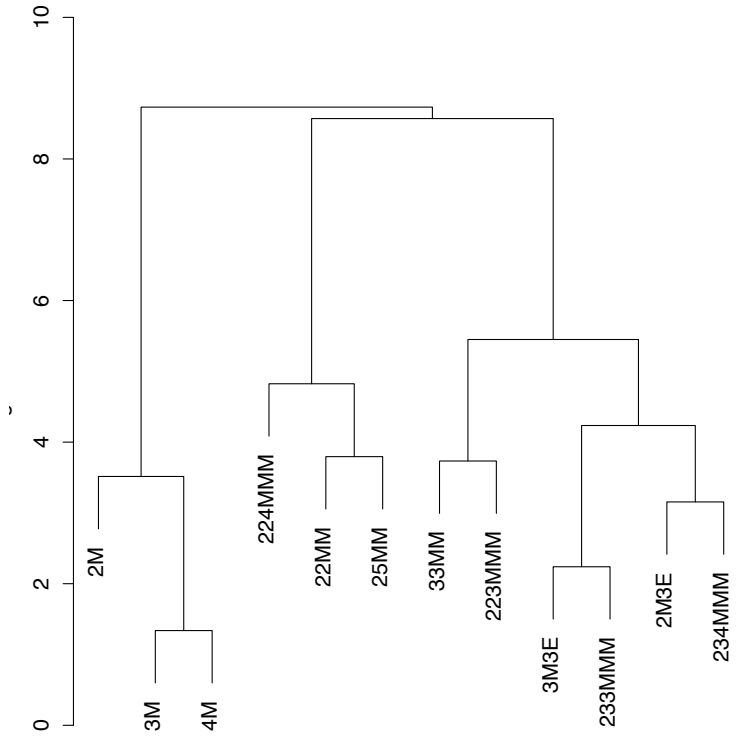


Figure from ISLR page 392.

Our N,N-dimethyl... example continued



(a) Single linkage.



(b) Complete linkage.

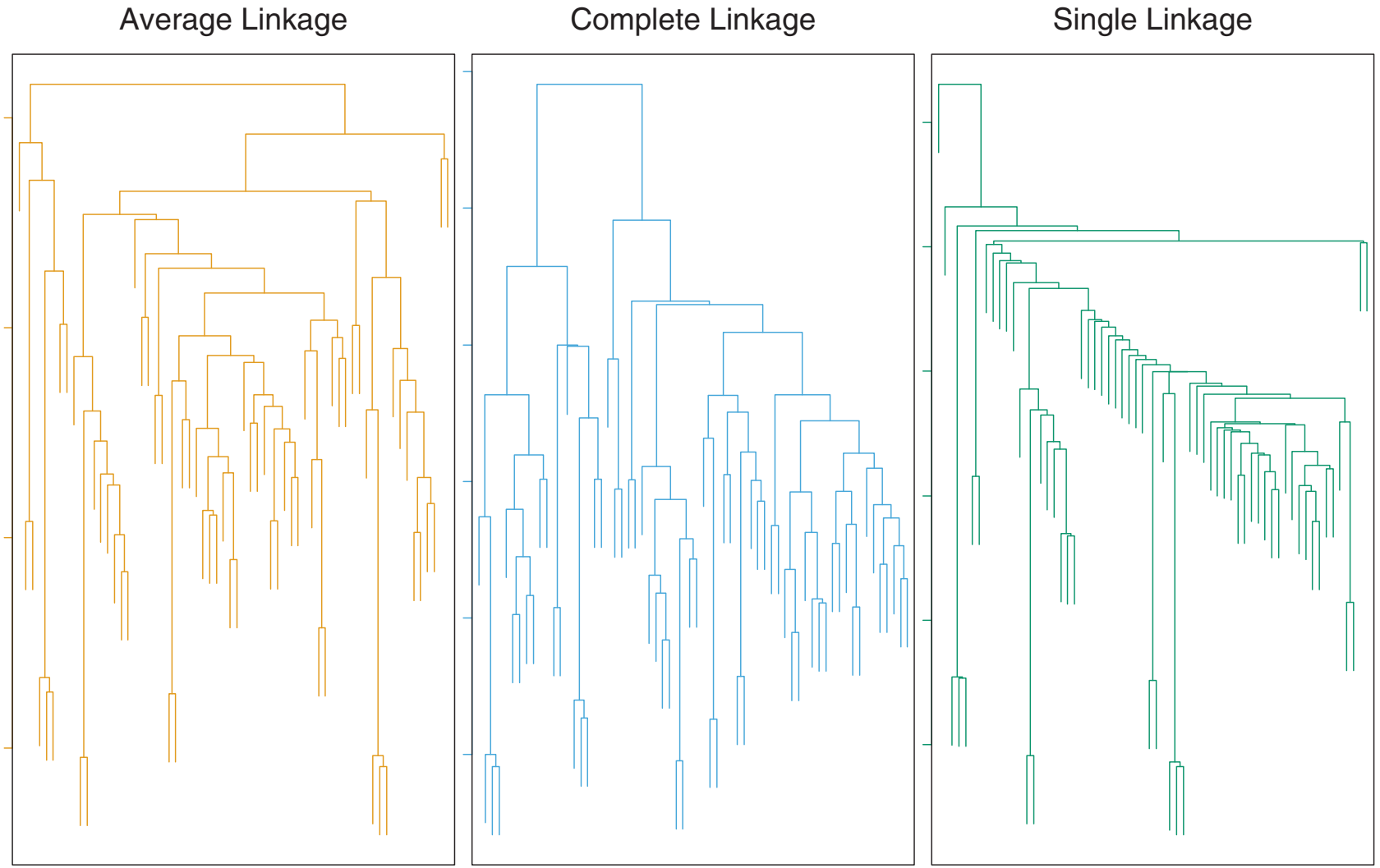


Figure from ISLR page 397.

K-means clustering

1. Randomly assign a number, from 1 to K, to each observation (initial cluster assignments)

2. Iterate until cluster assignments stop changing:

   a. For each of the K clusters, compute the cluster centroid.

   b. Assign each observation to the cluster whose centroid is closest (defined using Euclidean distance).

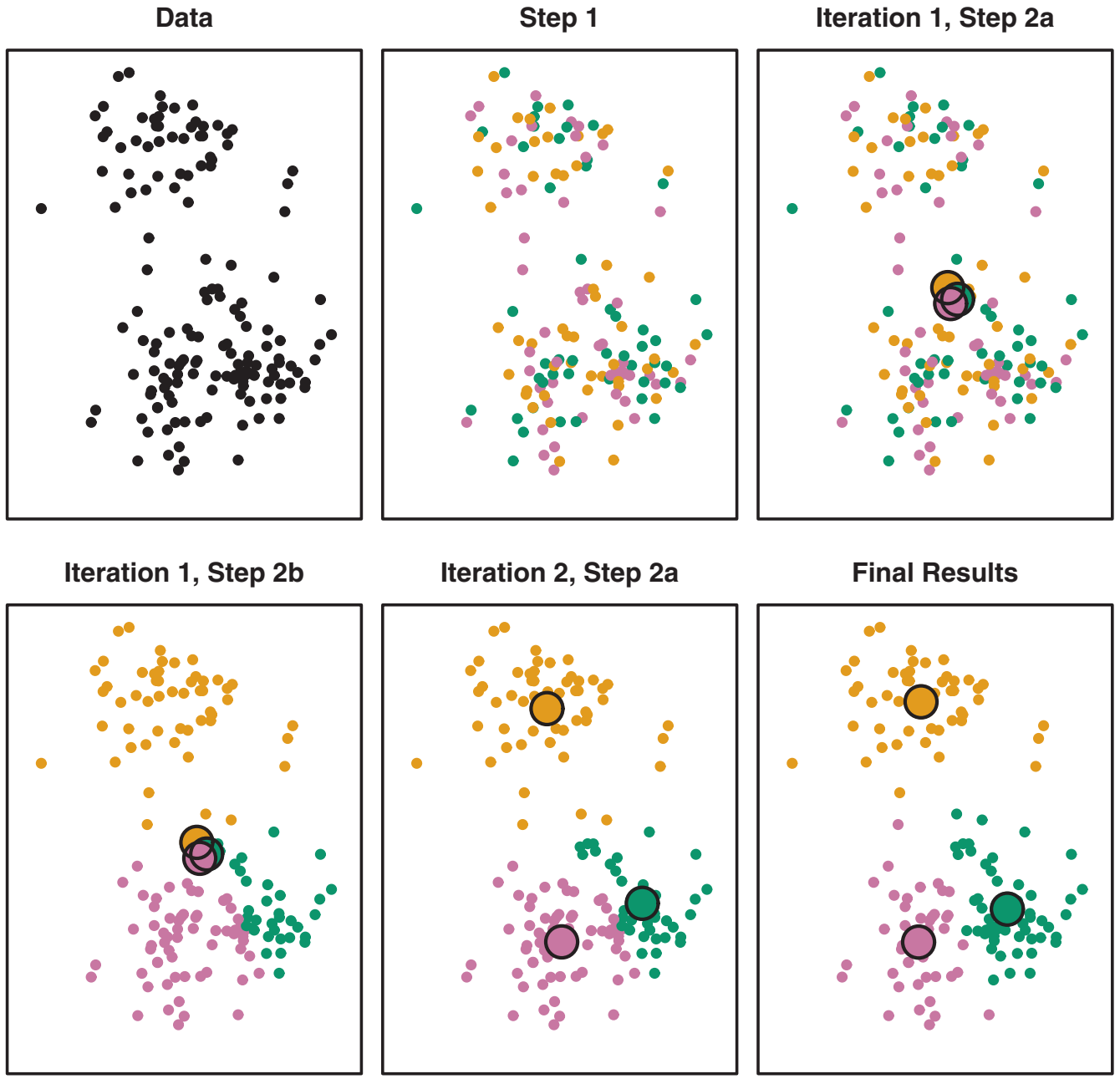
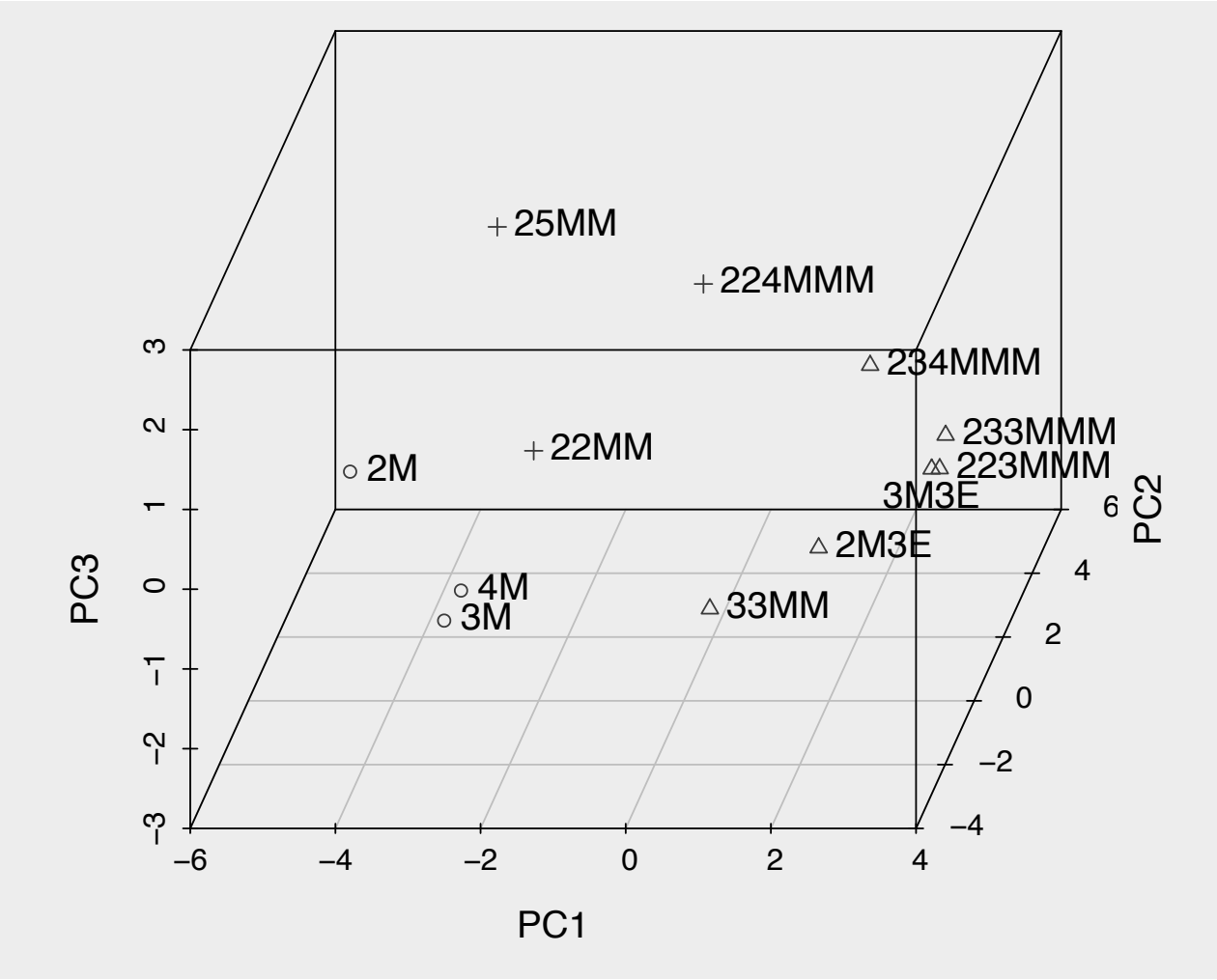
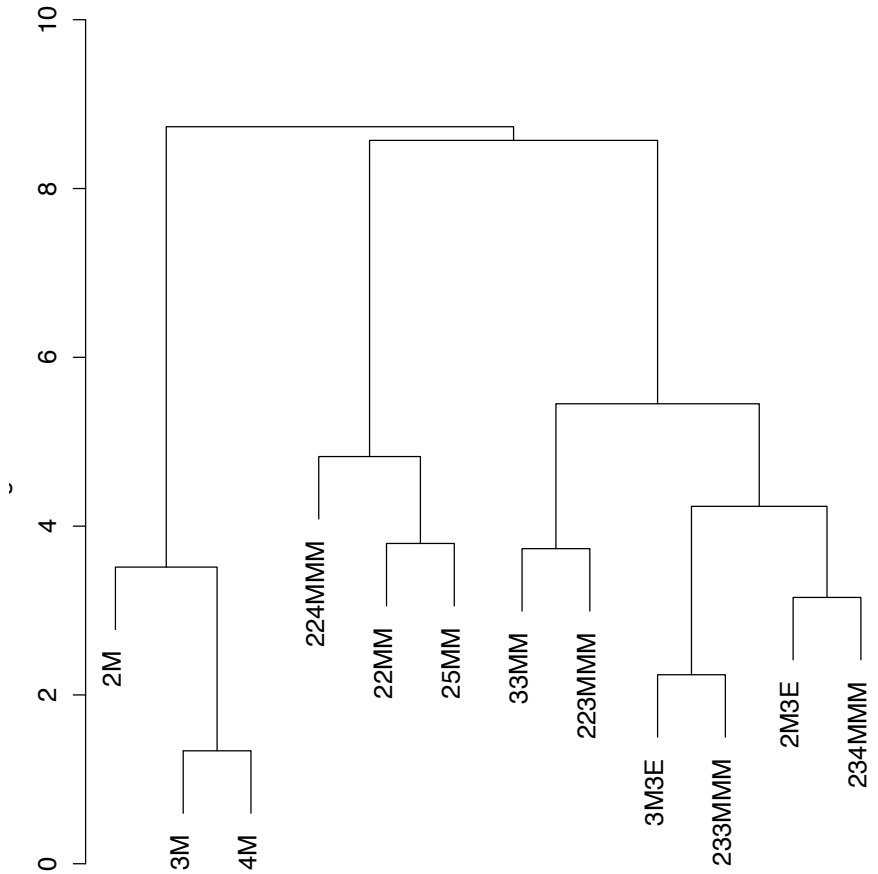


Figure from ISLR page 389.

PCA & K-means clustering for our N,N-dimethyl… example



and for hierarchical clustering with complete linkage



Some concluding remarks

- Principle components can be used as features in supervised learning (when response variables are available)
- K-means clustering is sensitive to the initial cluster assignments (it is important to make multiple runs of the algorithm and select the one that optimizes the objective function (see task in today's practical).
- Bottom-up or agglomerative hierarchical clustering is the most common, but a top-down or divisive version is also possible.
- Alternative distance measures exist for hierarchical clustering (such as correlation-based distance measures) which can be better suited for certain situations than the Euclidian distance.