Statistical Methods for Bioinformatics

XI

10 May 2016

Non-parametric methods

- Normal linear regression assumes e.g. normal distribution of errors.
  - Non-parametric covers techniques that do not rely on data belonging to any particular distribution. E.g. the Mann–Whitney U test for the hypothesis two samples are from the same population and is based on ranking your values. The test can be more powerful than a t-test on non-normal distributions .
- Polynomial expansions to fit a complex function still assume a single functional can generalize the predictor-response relationship.
  - Non-parametric methods make no (less) assumptions on the form of the functional

The simplest non-parametric regression

- A prediction for a value in a range is based on a **local weighted average** based on the nearby points.
- The function that defines the weights for the weighted average is dubbed a "kernel", e.g. a Gaussian kernel.
- The result is a smooth function
- package np in R

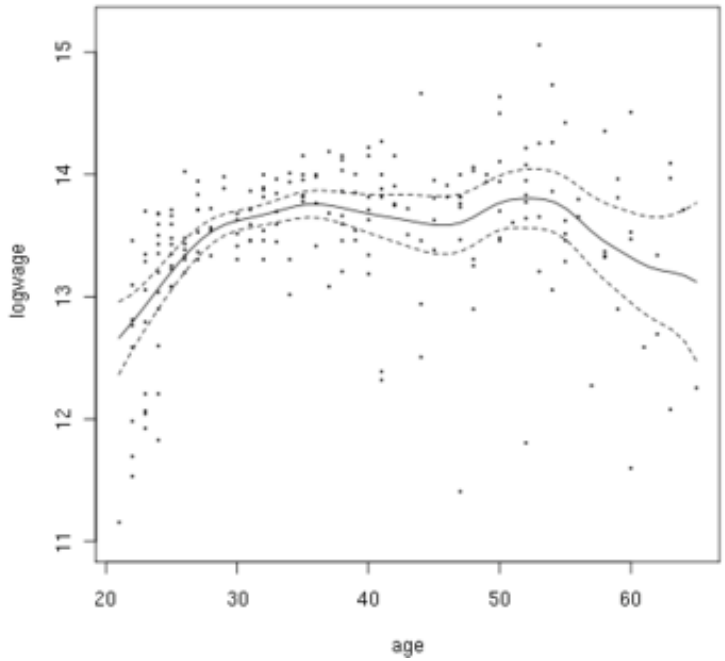
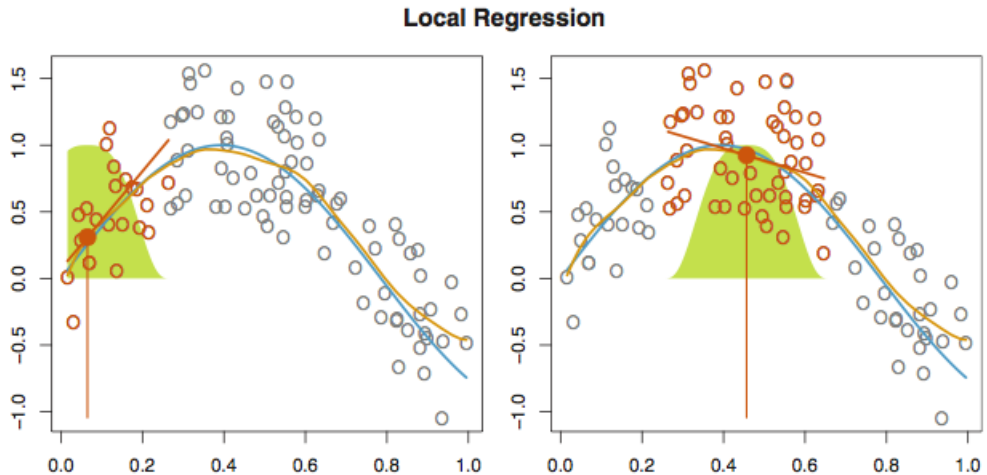


image from Wikipedia (http://en.wikipedia.org/wiki/Kernel_regression)

Local Regression; LOESS

- The relationship between predictor and response is now modelled with **local linear fits:** for a given x we generate a fit of the type $f(x) = \beta_0 + \beta_1 x$
- A weighted least squares fit is made for these simple linear models. The observations are sampled from around x and are weighted through a specified kernel function. The observations close to the value to be predicted are given most weight.
- LOESS/LOWESS stands for Locally Weighted Scatterplot Smoothing



Local Regression; LOESS

Algorithm 7.1 *Local Regression At* $X = x_0$

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0.

2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.

3. Fit a *weighted least squares regression* of the y_i on the x_i using the aforementioned weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2. \tag{7.14}$$

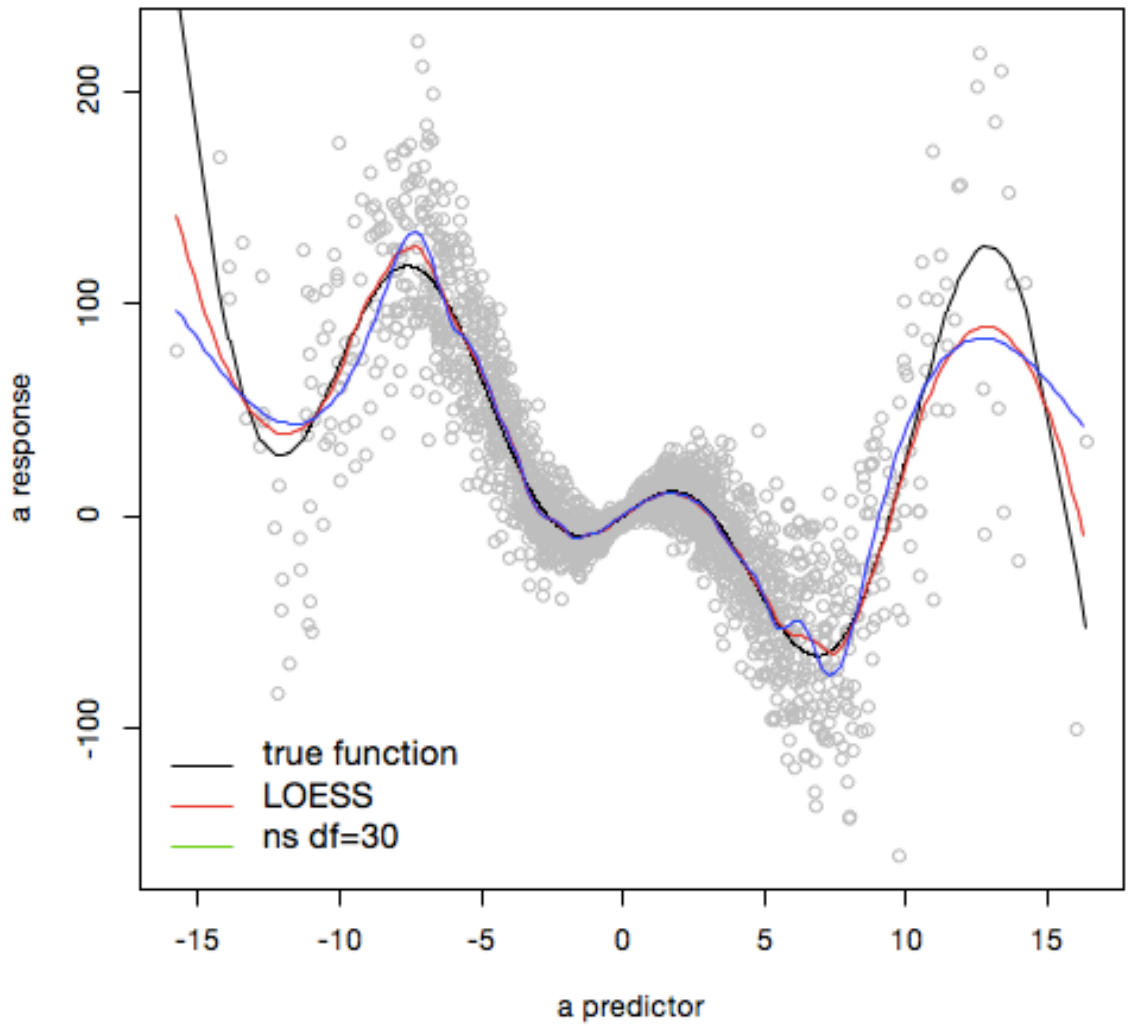
4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

Local Regression

- Choices:
  - The weighting function
    - a continuous, bounded, and symmetric real function
    - a running mean is known as the box kernel
    - a (truncated) Gaussian is a natural candidate
  - The weighting function comes with range parameter
    - e.g. the span, the fraction of the dataset considered by the kernel
  - Type of regression function
- Advantages: v. flexible fit
- Disadvantages:
  - Requires dense data to work well
  - No closed functional definition
    - a memory-based procedure

Local Regression vs Splines

- Which works better?



Specifics

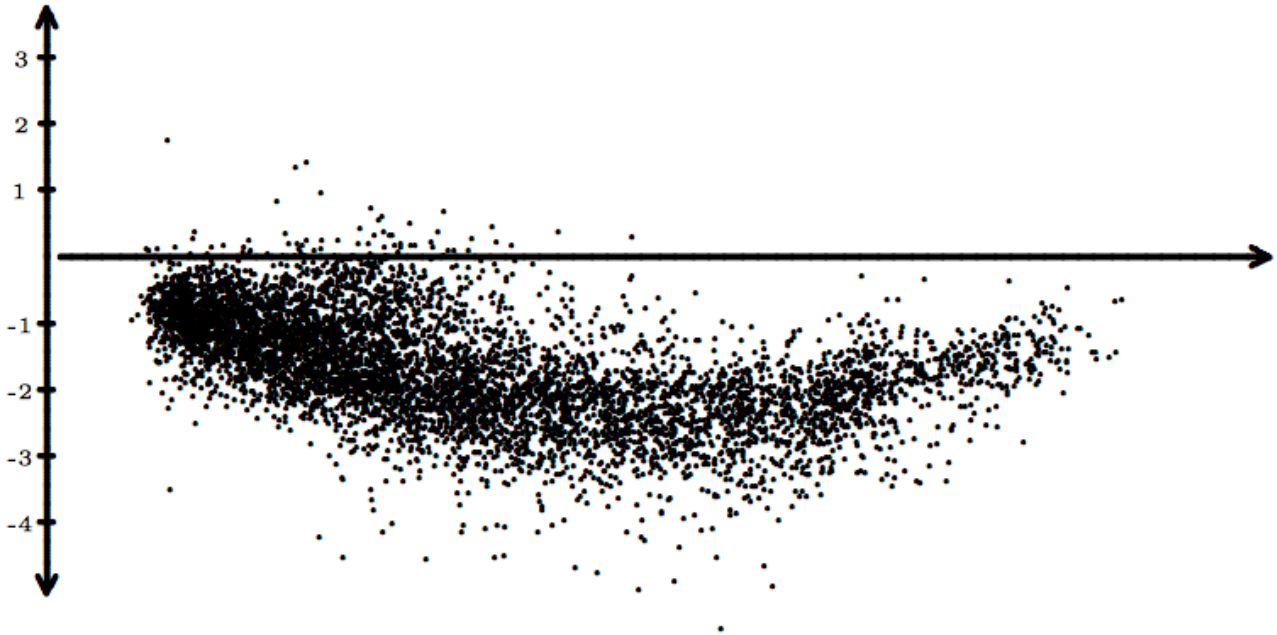
- Standard errors can be estimated for every point, however bootstrap estimates are often preferred
- The degrees of freedom used by the smoother can be estimated very similarly to how we did it for the smoothing spline. If we express the estimated function f as a function of the observations: $\hat{f} = Sy$, S is a matrix defined by our smoother and y are our observations. We can estimate the used degrees of freedom by $df = trace(S)$, the sum of the diagonal values of the matrix.

An Application of Non-Linear Models

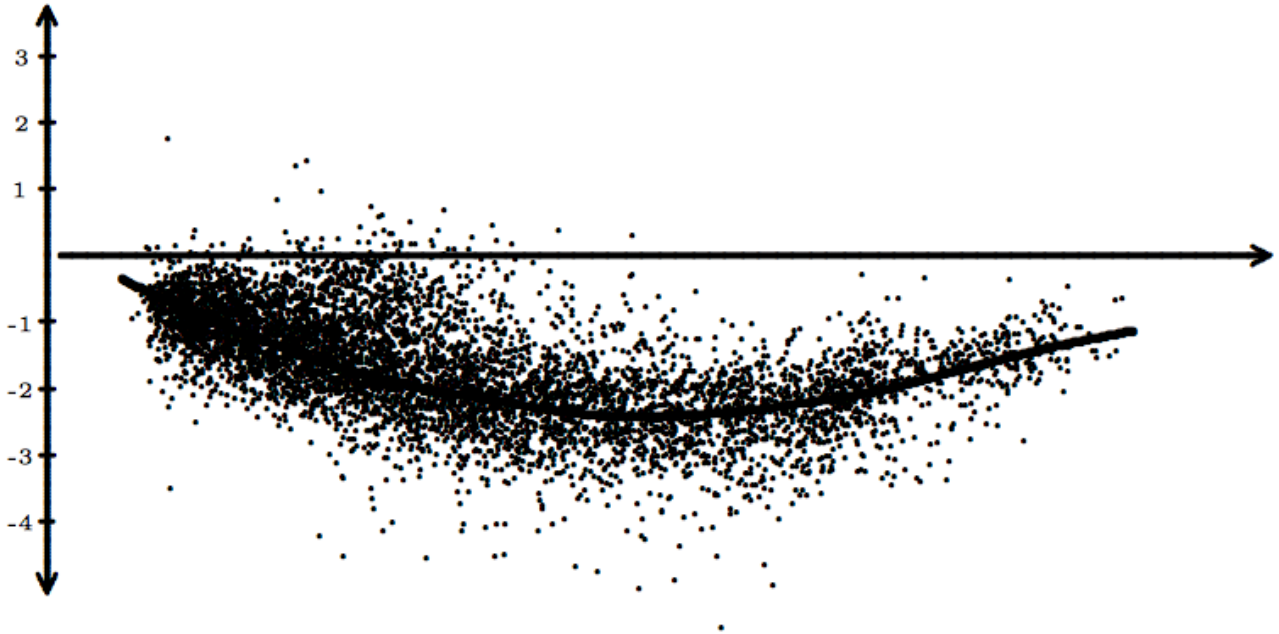
- One important use is to remove Systematic Experimental Bias from data; or calibration.
- An example: Spotted microarrays consist of spotted DNA samples in a regular pattern on a solid surface. Read out of relative abundances of mRNA by hybridization of cDNA tagged with a fluorescent dye. To compare two conditions, two dyes are used: e.g. Cy3 (green) and Cy5 (red).
- We are typically interested in the ratio between the signals as a measure of differential expression between conditions
- However the green dye often has a tendency to be stronger than the red dye. The magnitude of this effect varies from array to array. If we can measure this bias we can correct for it.
- A standard method of displaying microarray data that visualizes the spread between the two channels shows a G(g) as the Cy3 intensity for a gene g, and R(g) is the Cy5 intensity for g, and we plot M = log2(G(g)/R(g)) on the vertical axis, against A = (log2(G(g) + log2(R(g)))/2 on the horizontal axis

M versus A plot

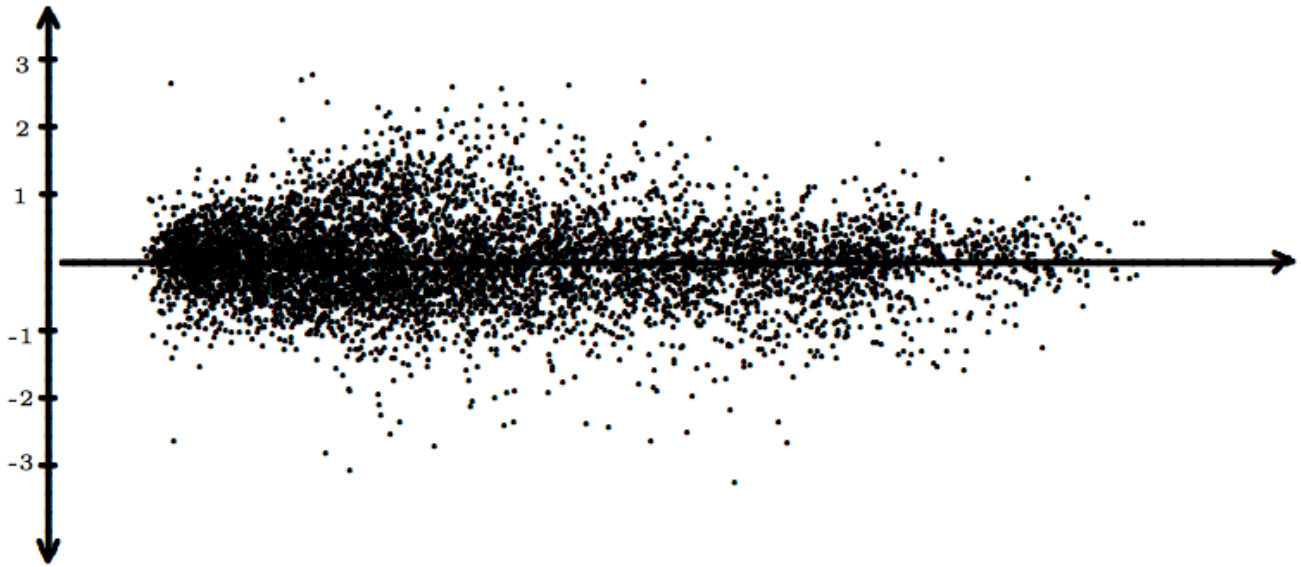
M is log fold (vertical axis), A is abundance (on the horizontal axis)



M versus A LOESS fit



M versus A LOESS fit subtracted



Details

- When one may not assume that most of the genes are unchanged between the two conditions, applying this method may normalize out true biological differences.
- Another issue of normalization involves the spread of the M values across the array, which may depend on the array itself and not on the biology.
- In real experiments there are normally many biases and random effects.

High dimensionsality

- Should we fit non-linearly when p is large (and n<p)?

Generalized Additive Models

- Generalized Additive Models (GAMs) extend the Generalized Linear Model so that non-linear responses can be included, maintaining the additive form between components.

$$g(y_i) = \beta_0 + \sum_{j=1}^{p} \beta_j f_j(x_{ij}) + \varepsilon_i$$

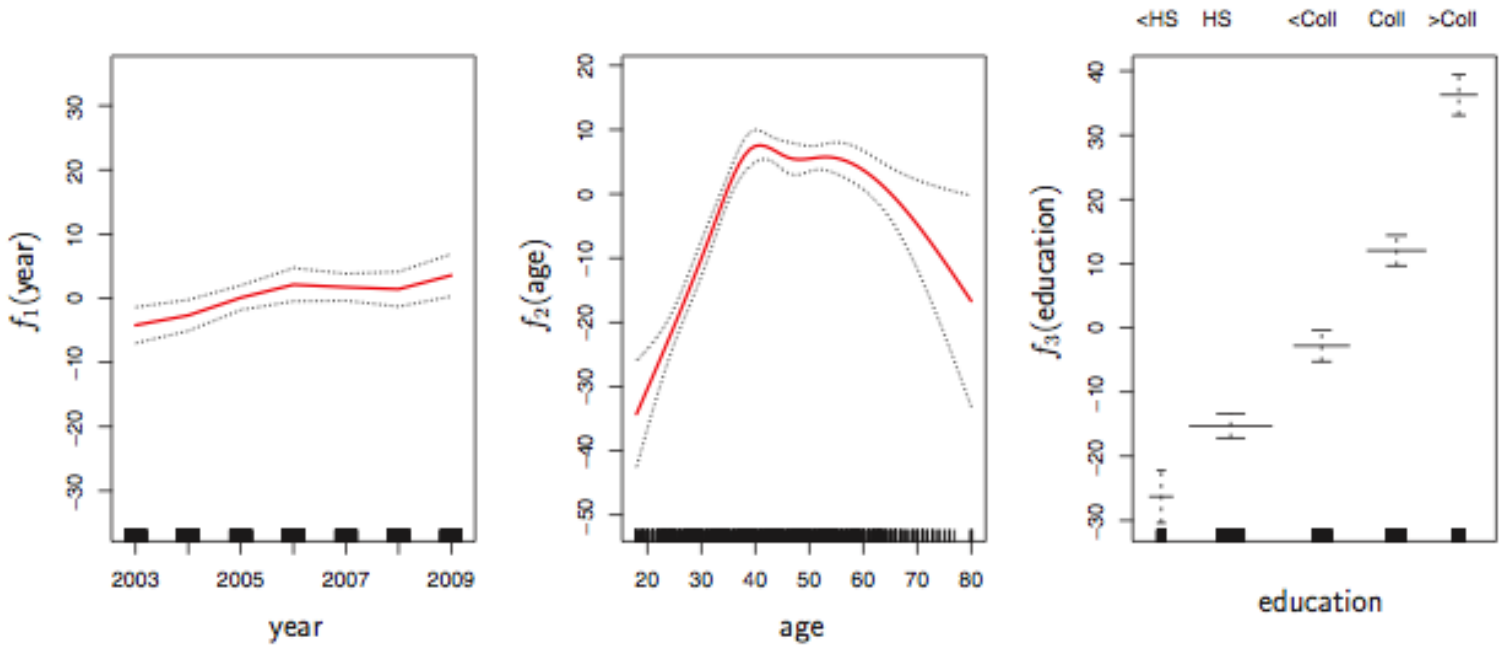
becomes

$$g(y_i) = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i$$

- For natural/regression splines the non-linear function can be represented as a normal set of basis functions and we can use a normal least squares approach and a general linear model!
- Other functionals push to alternative fitting procedures, as the backfitting procedure (exercise 11)

Generalized Additive Models

- Why the additive format?



For the Wage data, plots of the relationship between each feature and the response, wage, in the fitted model $wage = \beta_0 + f_1(year) + f_2(age) + f_3(education) + \varepsilon$. Each plot displays the fitted function and pointwise standard errors. The first two functions are natural splines in year and age, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable education.

GAM for Classification

A more general notation for part of the GAM formulation is

$$g(E(y)) = \beta_0 + \sum_{j=1}^{p} f_j(x_j)$$

where a link function g connects the predictions to a specified exponential error function distribution (e.g. Poisson, Gaussian, Binomial). Hence GAM's can also be used for classification problems:

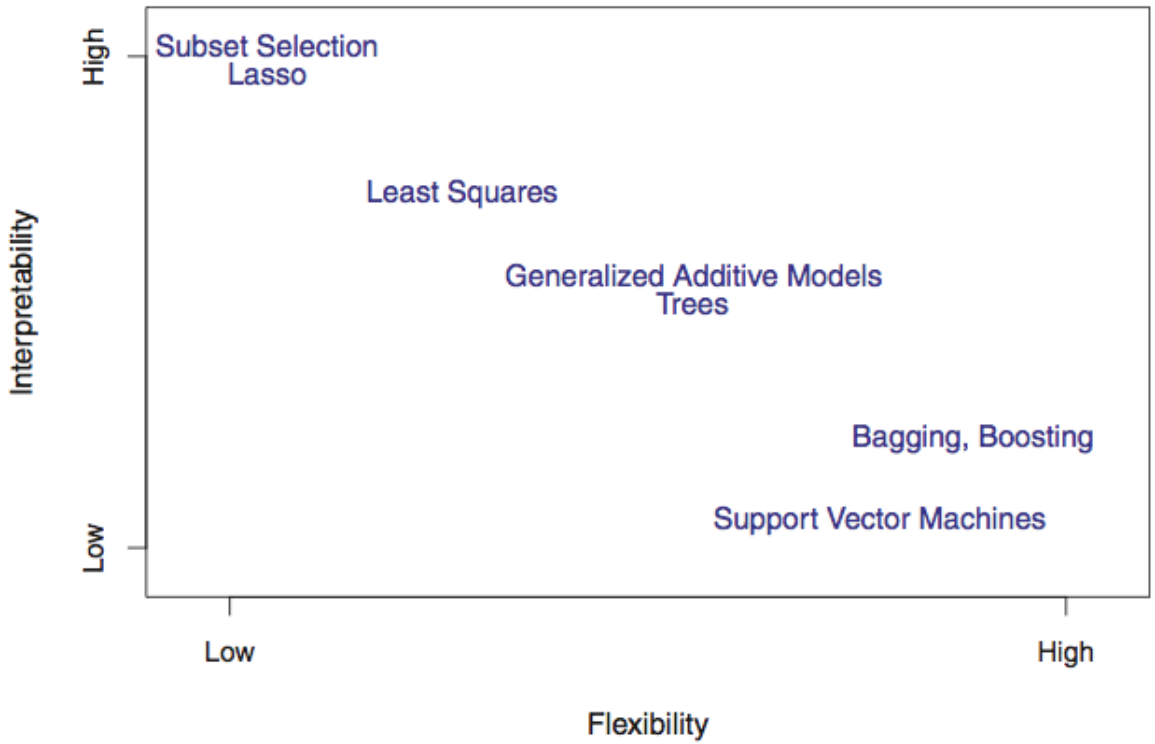
$$log(\frac{p(y_i)}{1 - p(y_i)}) = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i$$

Generalized Additive Models

- The GAM' allows flexible fits, with relaxed assumptions, to better represent relationships in the data. (lower bias)
- This comes at some loss of interpretablity.
  - Ease of understanding, summarization, communication
  - Parametrized methods give easy and simple predictions
- Overfitting is a serious problem!
  - Control degrees of freedom
  - Cross-validation
  - Compare GAM fits to GLM fits, is it really that much better?
- It is usually preferable to rely on a simple well understood model for predicting future cases, than on a complex model that is difficult to interpret and summarize.
- How about interactions between variables?

Even more flexible models

- A default GAM does not inherently incorporate interactions between variables, though they can be included.
- Another form of flexibility is to focus on interactions between variables.
- One can consider decision trees, Random Forests and SVM (etc)

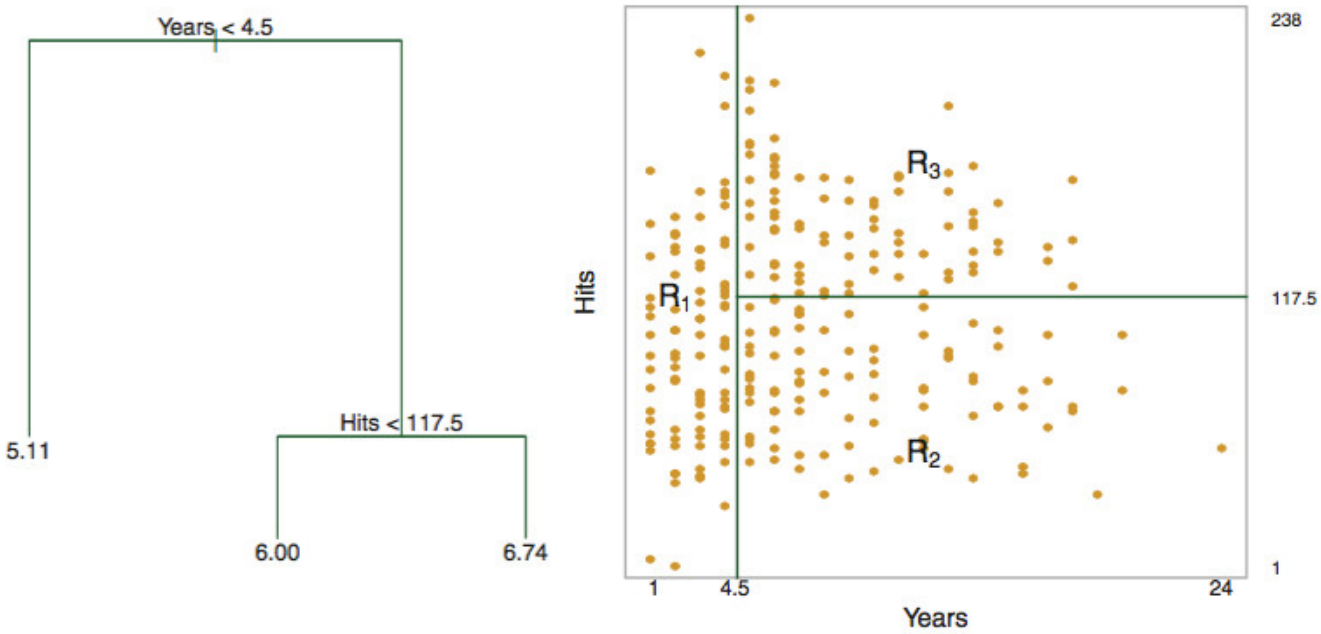


Trees are very broadly used

- Systematically structuring knowledge (Gene Ontology)
- Phylogenetic tree
- Many data structures
  - e.g. directory structure
- Decision trees as a procedure: e.g. in clinical practice

Tree-Based Methods

- Basic tree approaches are simple, and useful for interpretation
- Progressively stratifying or segmenting predictor space into regions.
- Readily exploit interactions between variables.



Building a tree

1. We divide the predictor space — that is, the set of possible values for X1,X2,...,Xp — into J distinct and non-overlapping regions, R1, R2, . . . , RJ .
2. For every observation that falls into the region Rj, we make the same prediction, which is simply the mean of the response values for the training observations in Rj.
3. The regions are high-dimensional rectangles/boxes
4. The goal is minimize RSS: $\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$

Building a tree

Procedure

1. Split the predictor space so that the biggest drop in RSS is achieved.
2. Then split one of two new spaces with the same criterion
3. Continue till some criterion is reached.

- This process can overfit the data if divisions continue till data scarcity
- Smaller trees tend to have less variance for a bit more bias.
- A strong limit on growth of the tree is often sub-optimal however
  - Stopping early may prevent finding v. good fits deeper in the tree.

Pruning a Tree

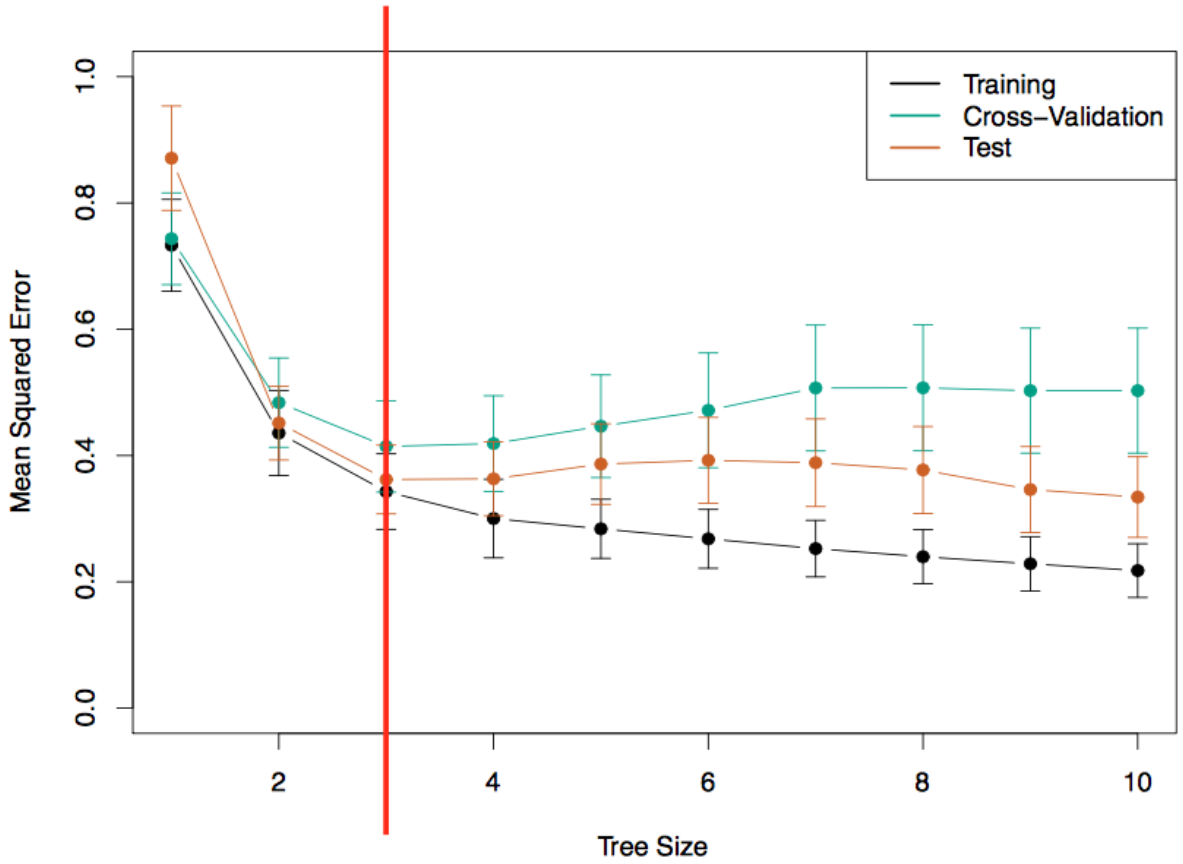
- Strategy of choice is to grow a tree and then prune it back.
- The branches that give the smallest drop in RSS for their number of splits are removed first. This is formalized as minimizing:

$$\sum_{m=1}^{|T|} \sum_{i:\ x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

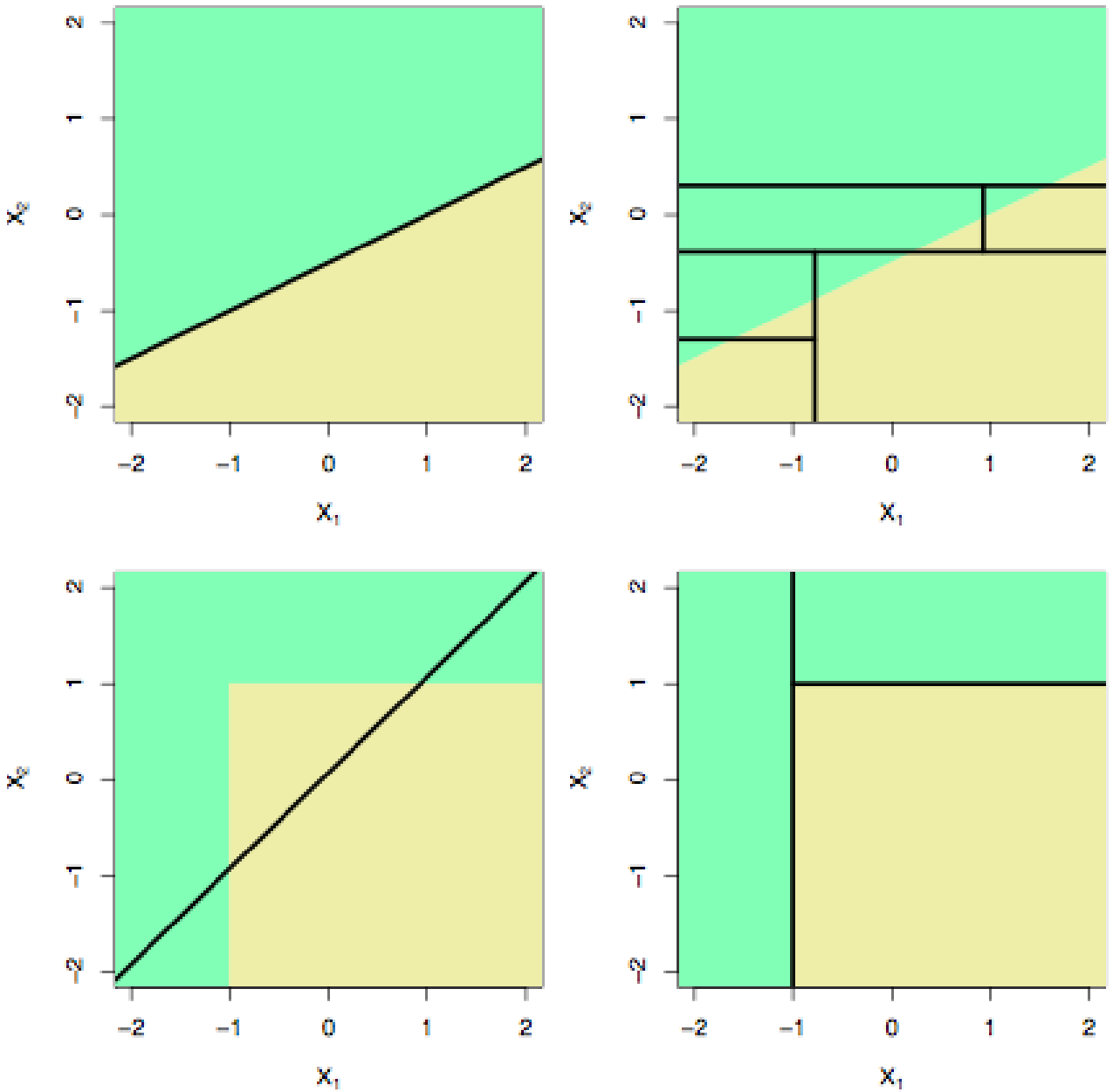
- $|T|$ represents the terminal node count
- α is a non-negative tuning parameter chosen with cross-validation

Example: Baseball Players' Salaries

The minimum cross validation error occurs at a tree size of 3



Trees vs Linear Model: classification example



Classification tree

- Same principle as regression tree
- Intuitive optimization function is to take for every box the most common class and take all examples not of this class as errors: $E = 1 - max_k(\hat{p}_{mk})$ with $\hat{p}_{mk}$ the proportion of observations in the m-th box of the k-th class.
- Above classification error is not very sensitive (many models have very similar scores) so we need something else
- Different cost function that measures purity of the nodes
  - Gini index $G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
  - cross-entropy $D = -\sum_{k=1}^{K} p_{mk} log(p_{mk})$

Tree recap

- Transparent and easy to understand method
  - Plotting and interpretation is easy
- Naturally incorporates interactions between variables
- Naturally incorporates qualitative predictors
- But they tend to not perform very well for most datasets

Ensemble methods

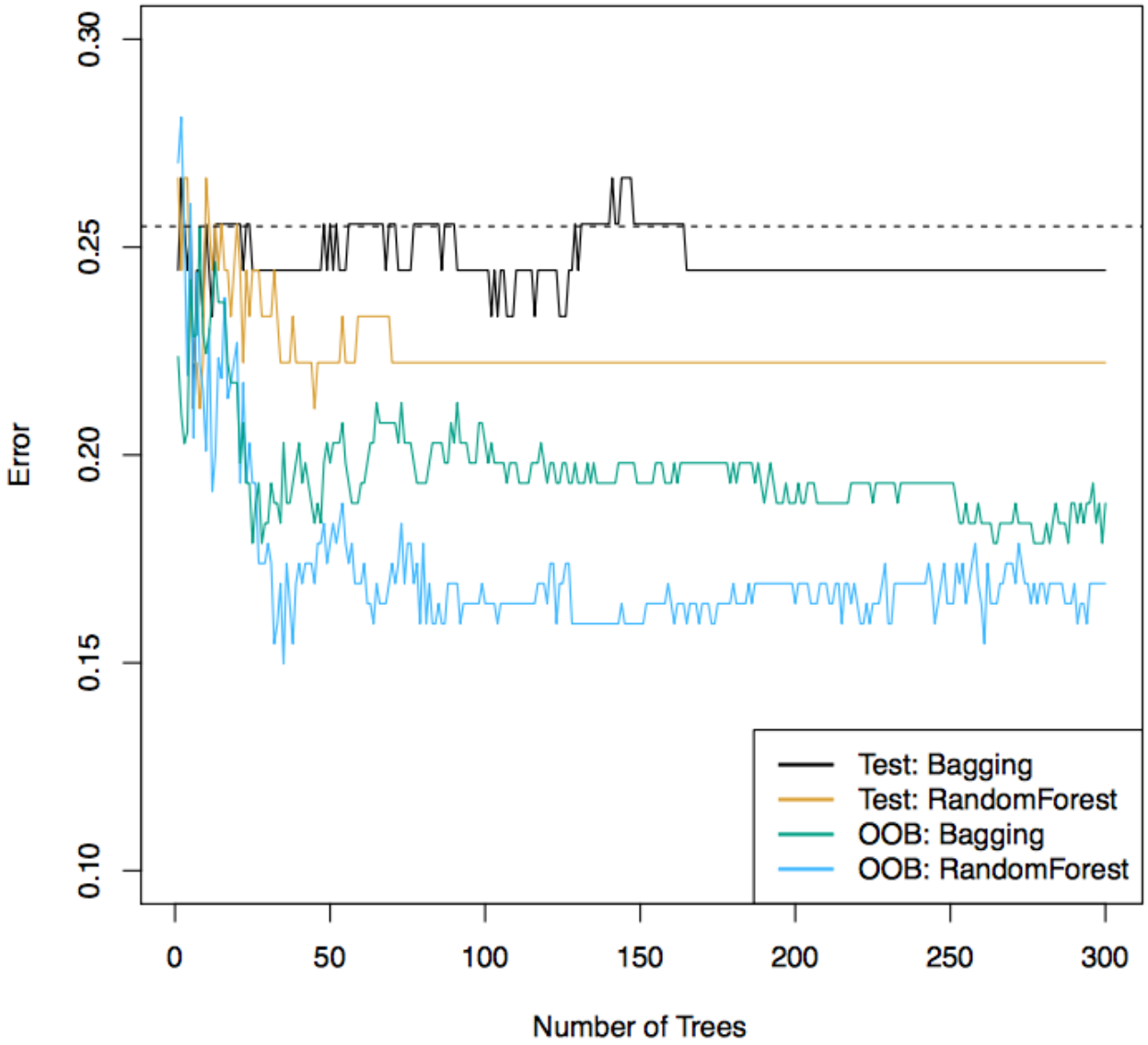
In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms.

- e.g. Bagging: a general-purpose procedure for reducing the variance of a statistical learning method; we introduce it here because it is particularly useful and frequently used in the context of decision trees.
- Important player was Leo Breiman (who proposed a.o. Random forests), a very creative man to advanced age.

Bagging

- Through bootstrap, resample your data repeatedly
- Learn different trees.
- Use this ensemble of trees to come to a single predictor:
  - average the predictions
  - majority vote for classification

The heart data



Out-of-Bag Error Estimation

- On average each bagged tree uses about two-thirds of the observations
- The remaining one-third can be used to evaluate performance (the out-of-bag OOB observations)
- The response for a given observation can be estimated using each of the trees for which it was not selected for learning. Average the prediction to estimate error.

Random forests

- Random forests improve over bagged trees by way of decorrelating the trees. This reduces the variance when we average the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- However each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors.
- A fresh selection of m predictors is taken at each split, and typically we choose $m \approx \sqrt{p}$ — that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors (4 out of the 13 for the Heart data).

Random Forests in the context of Bioinformatics

- A good and popular predictor.
- Works well with multiple correlated variables
- Suitable for high dimensional datasets.
- Large increase in predictive power at the cost of transparency.

E.g. for another Tree ensemble method:

Algorithm 8.2 *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{8.10}$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{8.11}$$
3. Output the boosted model,
   $$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{8.12}$$

Boosting

- Progressive learning
  - later trees focus on unexplained variation by weighting the data
- Again a very general **meta**-procedure that works beyond just trees

Boosting

Tuning features:

- Number of Trees in the ensemble (select with CV, overlearning can occur)
- Shrinkage parameter λ (speed of learning, value interacts with required number of trees)
- Depth of the individual trees (often a depth of 1 or 2)

Exercises

- Finish Lab chapter 7, and do the labs of chapter 8
- For next week exercises:
  - Chapter 7, exercise 10
  - For the vd Vijver dataset of class 3: Can you improve predictive performance with trees?
    - Evaluate performance for a classification tree, a bagging of classification trees, a random forest, and, facultatively, classification trees with boosting
    - Compare the variable importance plots for the simple Bagging and for Random Forests