Module 5: Resampling

TMA4268 Statistical Learning V2023

Sara Martino, Department of Mathematical Sciences, NTNU

February 8 and 9, 2023

Acknowledgements

- A lot of this material stems from Mette Langaas and her TAs. I would like to thank Mette for the permission to use her material!
- Some of the figures and slides in this presentation are taken (or are inspired) from James et al. (2013).

Introduction

Learning material for this module

- James et al (2021): An Introduction to Statistical Learning, Chapter 5.
- All the material presented on these module slides.

Additional material for the interested reader: Chapter 7 (in particular 7.10) in Friedman et al (2001): Elements of Statistical learning.

What will you learn?

- What is model assessment and model selection?
- Ideal solution in a data rich situation.
- Cross-validation and what is best:
  - validation set
  - leave-one-out cross-validation (LOOCV)
  - k-fold CV
- Bootstrapping - how and why.

Performance of a learning method

- Our models are "good" when they can generalize.
- We want a learning method to perform well on new data (low test error).
- Inference and understanding of the true pattern (in contrast to overfitting)

This is important both for

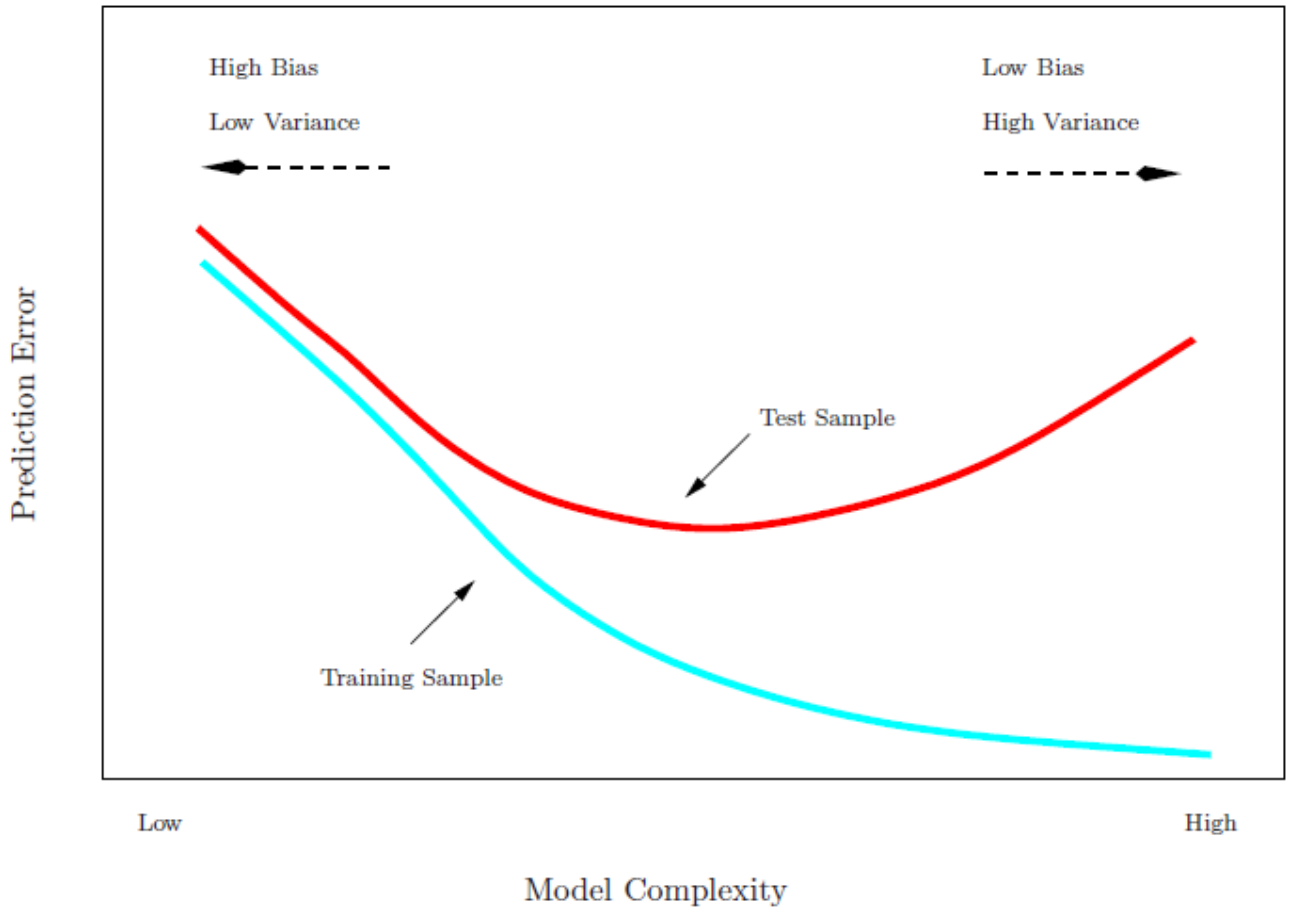
Model selection

Estimate the *performance* of different models to *choose the best model.*

Model assessment

Estimating the performance (prediction error) of the final model, on new data.

Training vs Test Error



Loss functions

In order to define how we measure error, we must first decide for a **loss function**. Here we use:

- *Mean squared error* (quadratic loss) for regression problems (continuous outcomes) $Y_i = f(x_i) + \varepsilon_i$, $i = 1, \ldots, n$:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2 \ .$$

- *Misclassification rate* (0/1 loss) for classification problems where we classify to the class with the highest probability $P(Y = j \mid x_0)$ for $j = 1, \ldots, K$:

$$\frac{1}{n} \sum_{i=1}^{n} \mathrm{I}(y_i \neq \hat{y}_i) \ .$$

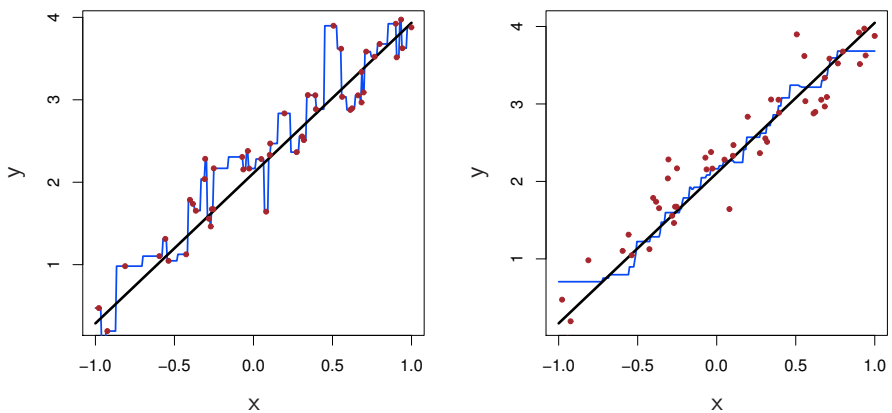
NB: These are just two of the possible loss functions, there are several model to be found in the literature

KNN regression (chapter 3.5 in course book)

- The KNN regression method provides a prediction at a value x_0 by finding the closest K points (Euclidean distance) and calculating the average of the observed y values at the points in the respective neighborhood $\mathcal{N}_0$

$$\hat{f}(x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} y_i \ .$$

Illustration: Linear regression with $K = 1$ (left) and $K = 9$ (right).

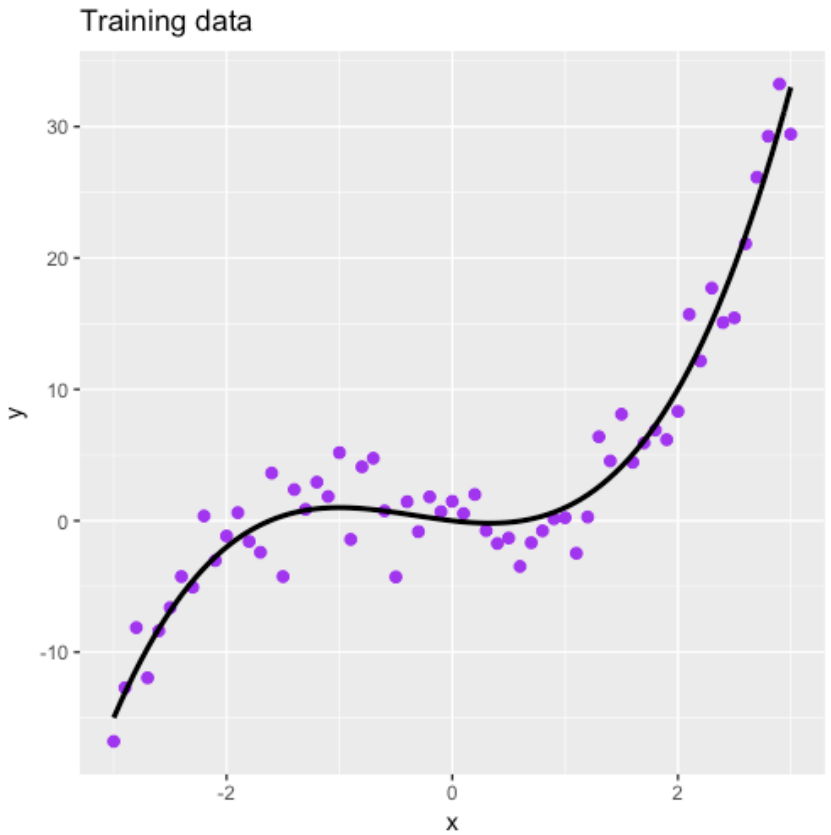


(Figure 3.17 from James et al. (2013)).

What happens for K = number of data points?

Example

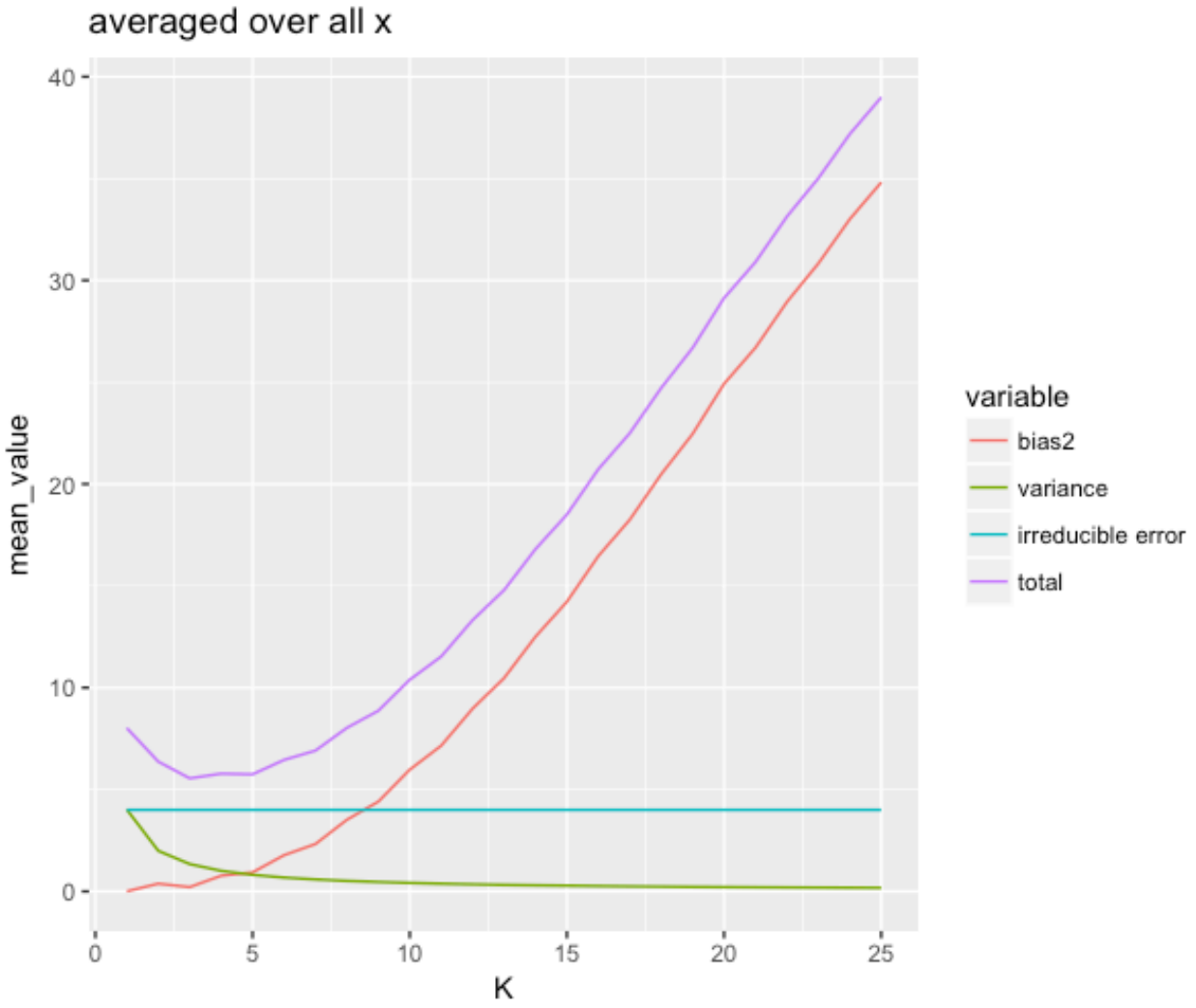
We aim to do *model selection* in KNN-regression, where true curve is $f(x) = -x + x^2 + x^3$ with $x \in [-3, 3]$. $n = 61$ for the training data.



- We have considered $K = 1, \ldots, 25$, and repeated the experiment $M = 1000$ times (that is, M versions of training and test set).

Remember: The bias-variance trade-off

For KNN: K small = high complexity; K large = low complexity.



The challenge

- In the above examples we knew the truth, so we could assess training and test error.
- In reality this is of course not the case.
- We need approaches that work with real data!

The data-rich situation (often unrealistic)

If we had a large amount of data we could divide our data into three parts:

- **Training set**: to fit the model
- **Validation set**: to select the best model (*model selection*)
- **Test set**: to assess how well the model fits on new independent data (*model assessment*)

Q: Before we had just training and test. Why do we need the additional validation set?
A: We have not discussed model selection before.

Q: Why can't we just use the training set for training, and then the test set both for model selection and for model evaluation?
A: We will be too optimistic if we report the error on the test set when we have already used the test set to choose the best model.

- If you have a lot of data – great – then you do not need Module 5.
- But, this is very seldom the case – so we will study other solutions based on efficient sample reuse with *resampling* data.
- An alternative strategy for model selection (using methods penalizing model complexity, e.g. AIC or lasso) is covered in Module 6.

We will look at *cross-validation* and the *bootstrap*.

Cross-validation (CV)

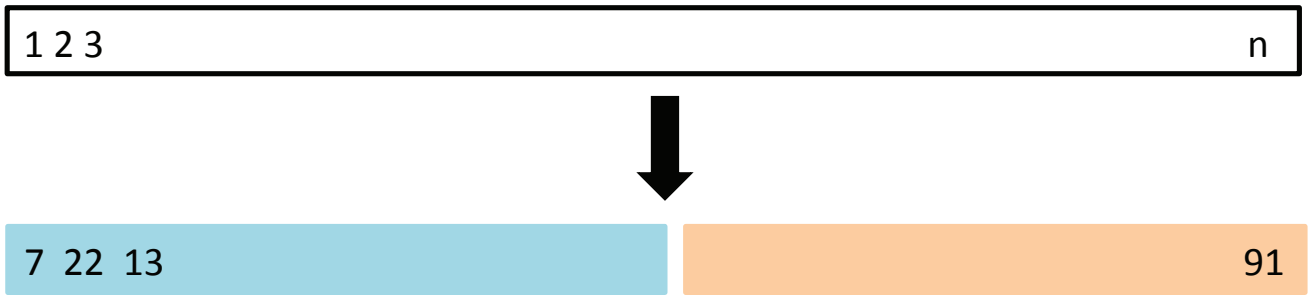
"Model selection" situation: We assume that test data is available (and has been put aside), and we want to use the rest of our data to find the model that performs "best", that is, *with lowest test error*.

This can be done by:

- the validation set approach (not strictly a *cross*-validation approach).
- leave one out cross-validation (LOOCV).
- k-fold cross-validation (CV), typically $k = 5$ or 10.

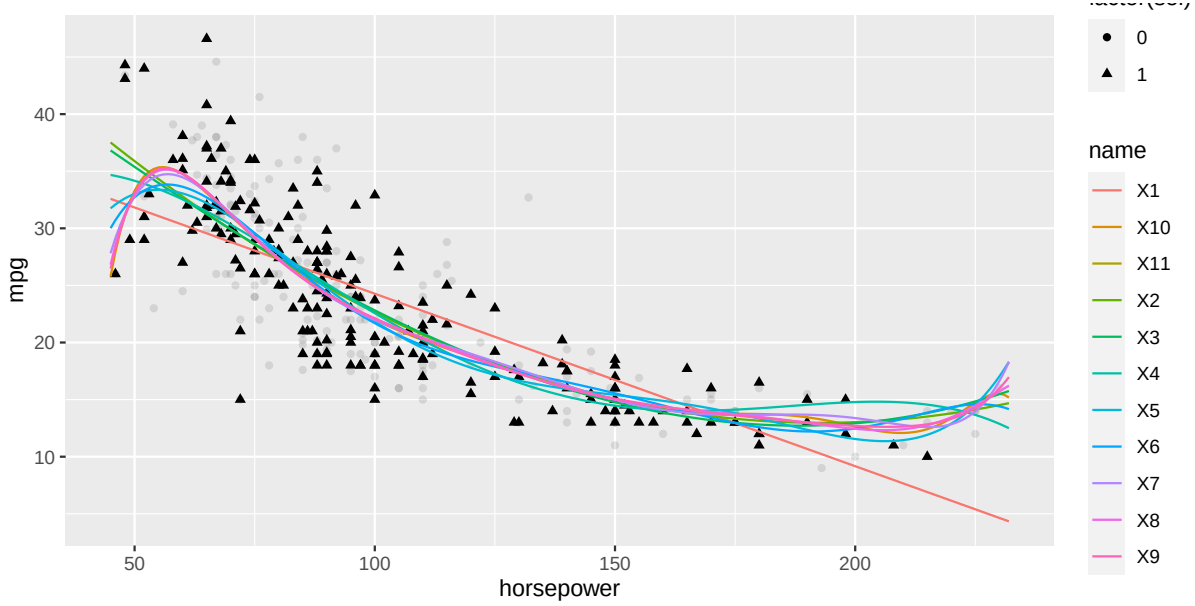
The validation set approach

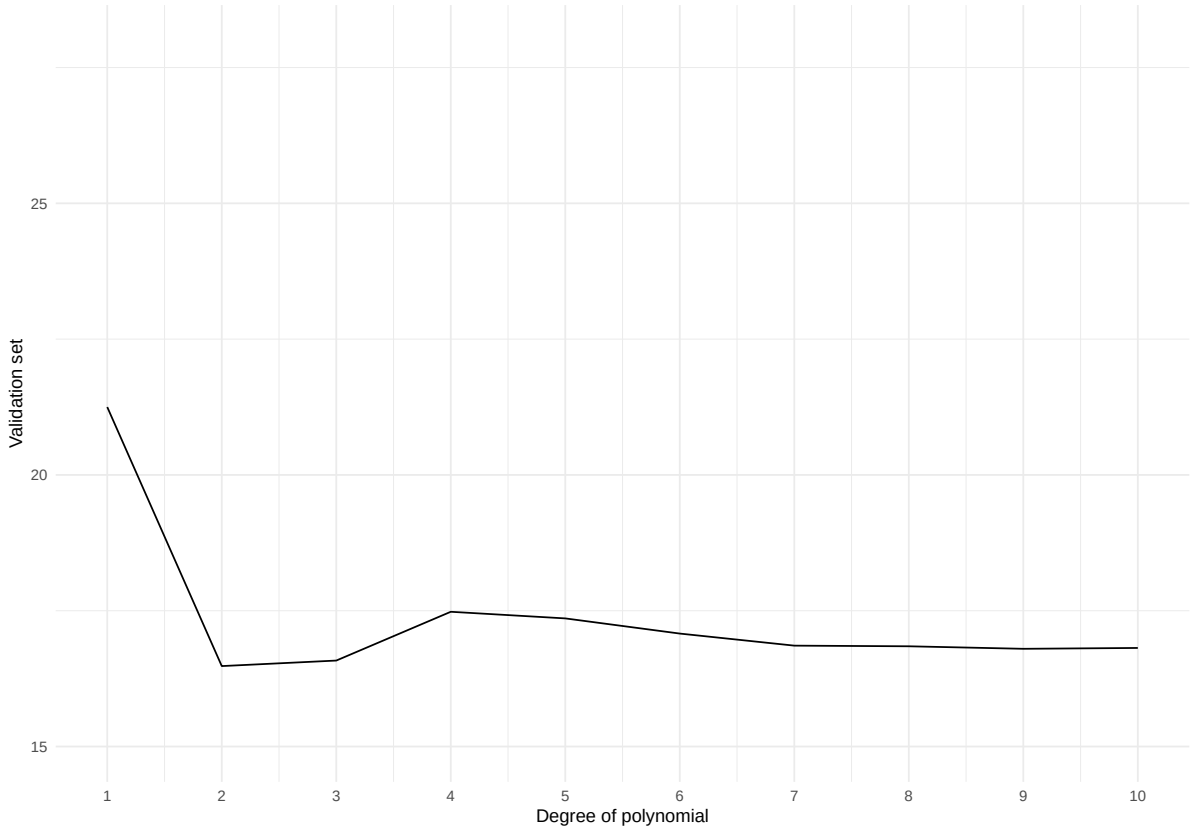
- Consider the case when you have a data set consisting of n observations.
- To fit a model and to evaluate its predictive performance you randomly divide the data set into two parts ($n/2$ sample size each):
  - a *training set* (to fit the model) and
  - a *validation set* (to make predictions of the response variable for the observations in the validation set)



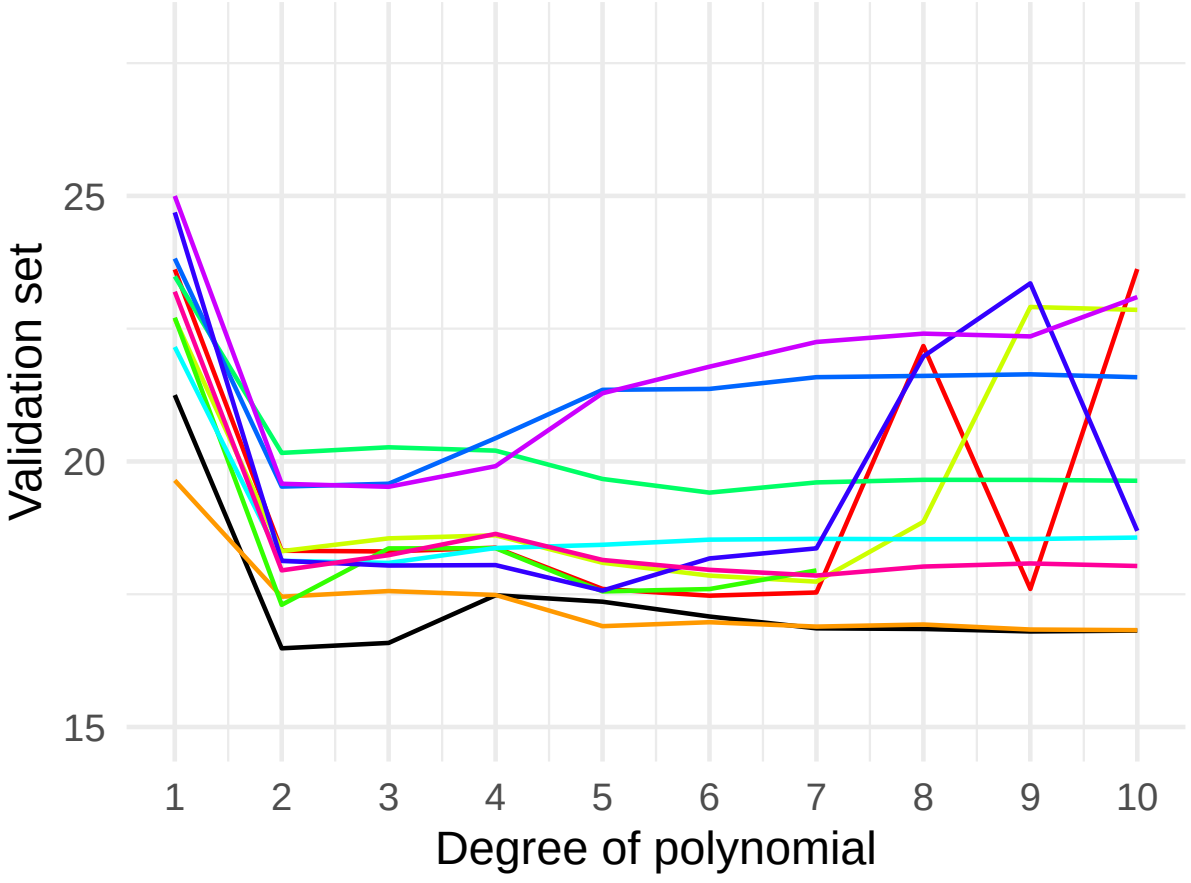
Example of validation set approach

Auto data set (library `ISLR`): predict `mpg` (miles pr gallon) using polynomial function of `horsepower` (of engine), $n = 392$. What do you see?




25
20
15
Validation set
1
2
3
4
5
6
7
8
9
10
Degree of polynomial

But what if we select another split into two parts? Many splits:



$\rightarrow$ No consensus which model really gives the lowest validation set MSE.

Drawbacks with the validation set approach

- *High variability* of validation set error due to dependency on the set of observation included in the training and validation set.
- *Smaller sample size* for model fit, as only half of the observations are in the training set. Therefore, the validation set error may tend to overestimate the error rate on new observations for a model that is fit on the full data set (the more data, the lower the error).

Better ideas?

Leave-one-out cross-validation (LOOCV)

Leave-one-out cross-validation (LOOCV) addresses the limitations of the validation set approach.

Idea:

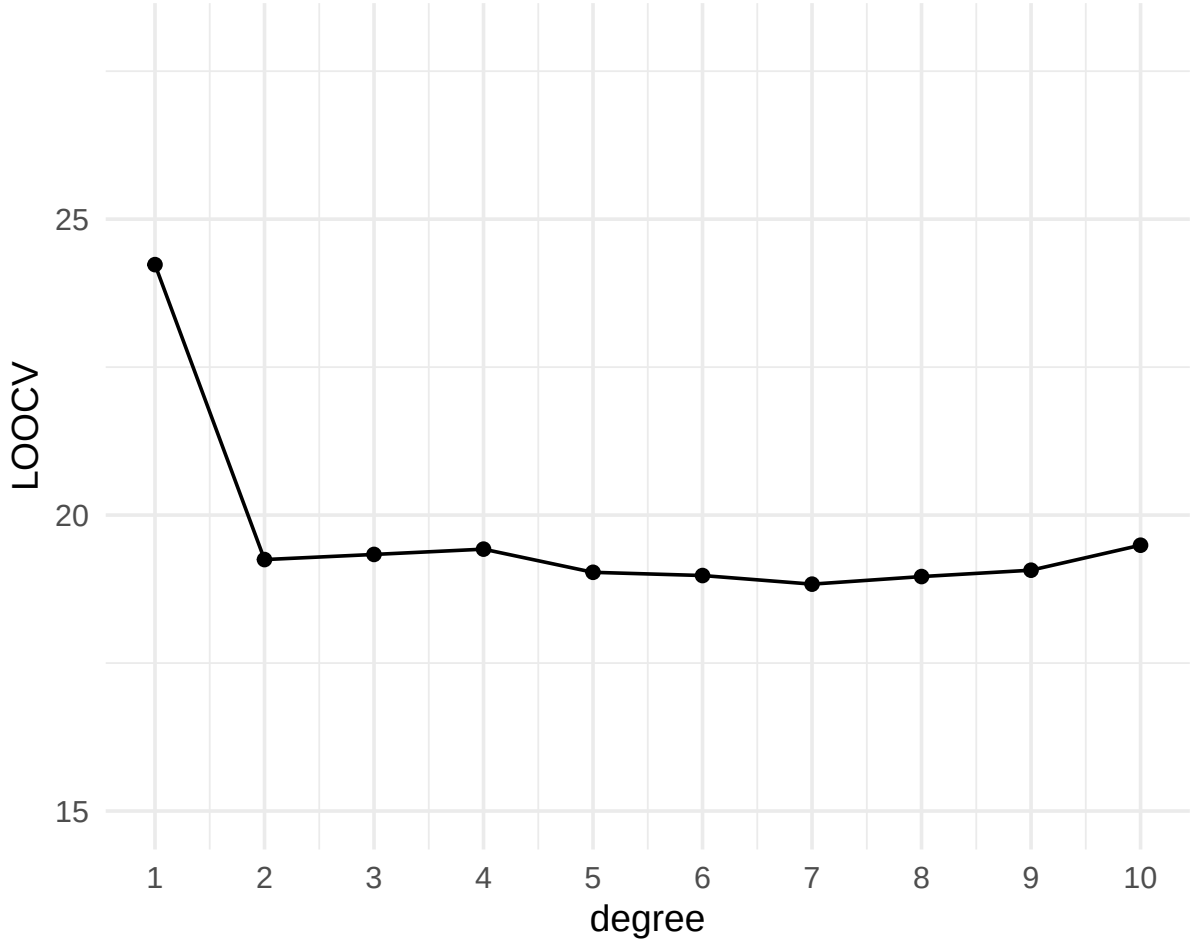
- Only **one observation at a time** is left out (test set size $n = 1$).
- The remaining $n - 1$ observations make up the training set.
- The procedure of model fitting is repeated n times, such that each of the n observations is left out once. In each step, we calculate

$$\text{MSE}_i = (y_i - \hat{y}_i)^2 \ .$$

- The **total prediction error** is the mean across these n models

$$\text{CV}_n = \frac{1}{n} \sum_{i=1}^{n} \text{MSE}_i \ .$$

Regression example: LOOCV



Issues with leave-one-out cross-validation

- Pros:
  - No randomness in training/validation splits!
  - Little bias, since nearly the whole data set used for training (compared to half for validation set approach).
- Cons:
  - Expensive to implement – need to fit n different models.
  - High variance since: two training sets only differ by one observation, thus estimates from each fold highly correlated, which can lead to high variance in their average*.

* Recall that

$$\mathrm{Var}(\sum_{i=1}^{n} a_i X_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \mathrm{Cov}(X_i, X_j)$$

$$= \sum_{i=1}^{n} a_i^2 \mathrm{Var}(X_i) + 2\sum_{i=2}^{n}\sum_{j=1}^{i-1} a_i a_j \mathrm{Cov}(X_i, X_j).$$

LOOCV for multiple linear regression

There is a nice shortcut for LOOCV in the case of linear regression:

$$\mathrm{CV}_n = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2 \quad ,$$

where h_i is the ith diagonal element (leverage) of the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$, and $\hat{y}_i$ is the ith fitted value from the original least squares fit.

$\rightarrow$ Need to fit the model only once!

k-fold cross-validation

To address the drawbacks of LOOCV, we can leave out not just one single observation in each iteration, but $1/k$-th of all data.

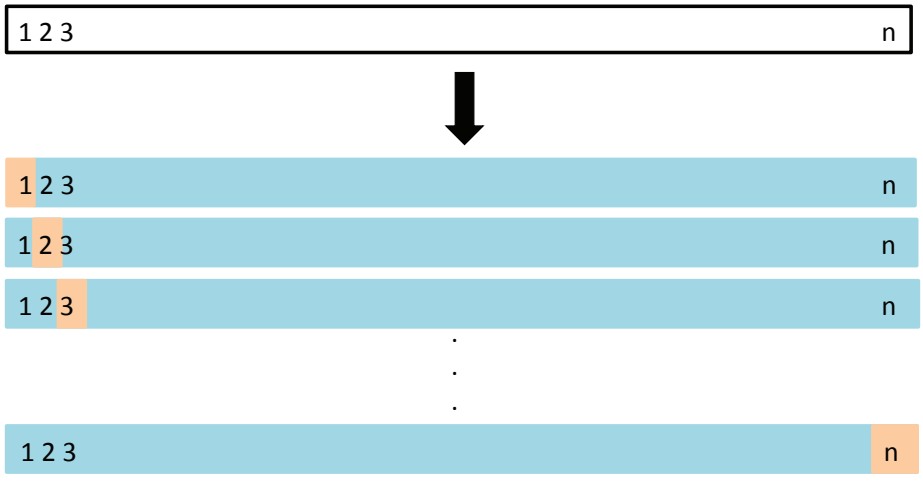
Procedure:

- Split the data into k (more or less) equal parts.
- Use $k-1$ parts to fit and the kth part to validate.
- Do this k times and leave out another part in each round.

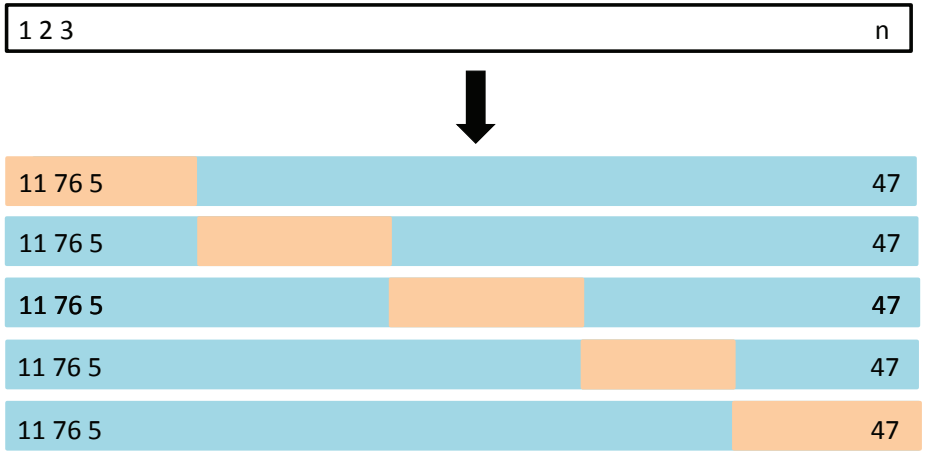
The MSE is then estimated in each of the k iterations $(\text{MSE}_1, \ldots, \text{MSE}_k)$. The k-fold CV is then a (weighted) avearge over the k MSEs.

Comparison of LOOCV and k-fold CV:

LOOCV:



k-fold:



Formally

- Indices of observations - divided into k folds: $C_1, C_2, \ldots, C_k$.
- n_j elements in fold j. If n is a multiple of k then $n_j = n/k$ for all folds.

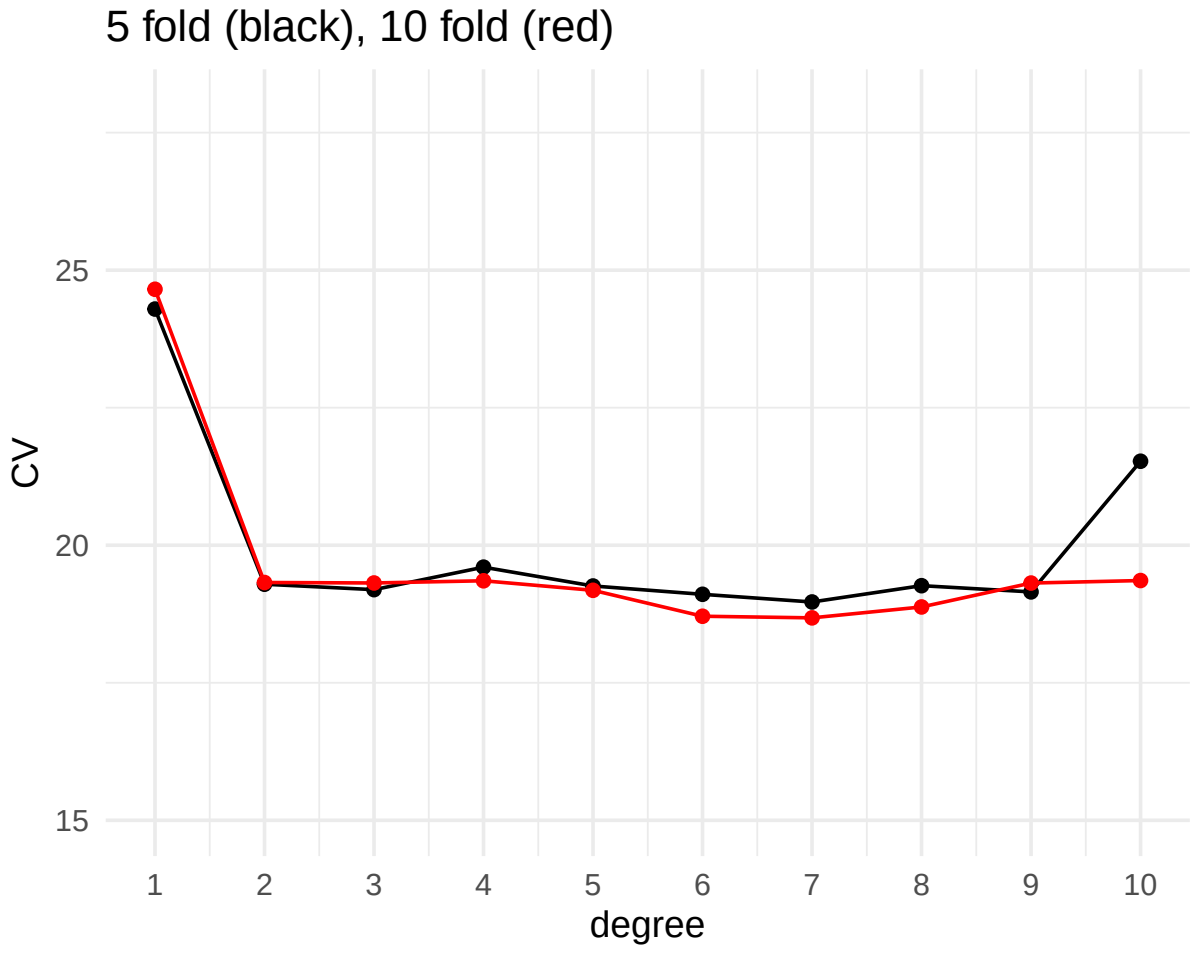
$$\text{MSE}_j = \frac{1}{n_j} \sum_{i \in C_j} (y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the fit for observation i obtained from the data with part j removed.

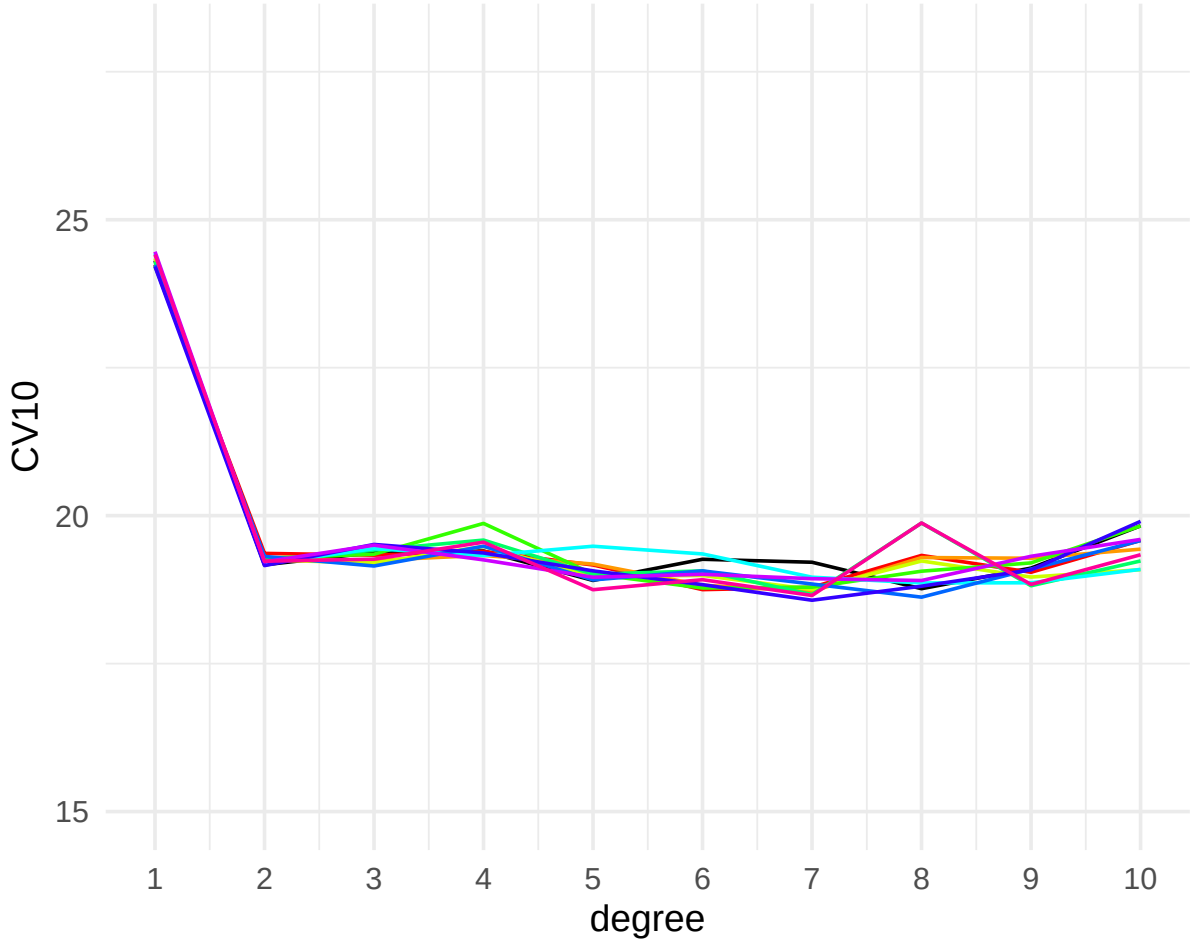
$$\text{CV}_k = \frac{1}{n} \sum_{j=1}^{k} n_j \text{MSE}_j$$

Observe: setting $k = n$ gives LOOCV.

Regression example: 5 and 10-fold cross-validation



10 reruns (different splits) of the 10-CV method - to see variability:



There still *is* variability, but *much less* than for validation set approach.

Issues with k-fold cross-validation

1. The *result may vary* according to how the folds are made, but the variation is in general lower than for the validation set approach.
2. Computational load lower with $k = 5$ or 10 than LOOCV.
3. The training set is $(k-1)/k$ times the size of the original data set - the estimate of the prediction error is biased upwards.
4. This bias is the smallest when $k = n$ (LOOCV), but we know that LOOCV has high variance.
5. Due to the *bias-variance-trade-off*, k-fold CV often gives more accurate estimates of the test error rate than does LOOCV.
   $\rightarrow$ $k = 5$ or $k = 10$ is used as a compromise.

Choosing the best model

- There is a model parameter (maybe K in KNN or the degree of the polynomial), say θ, involved to calculate CV_j, $j = 1, \ldots, k$
- Based on the CV vs θ-plot we can choose the model with *the smallest* CV_k as our best model.
- We then fit this model using the whole data set (not the test part, that is still kept away), and evaluate the performance on the test set.

One standard error rule:

Denote by $\mathrm{MSE}_j(\theta)$, $j = 1, \ldots, k$ the k parts of the MSE that together give the CV_k.

We can compute the sample standard deviation (standard error) of all $\mathrm{MSE}_j(\theta)$, $j = 1, \ldots, k$

$$\hat{\mathrm{SE}}(\mathrm{CV}_k(\theta)) = \sqrt{\sum_{j=1}^{k} (\mathrm{MSE}_j(\theta) - \overline{\mathrm{MSE}}(\theta))/(k-1)}$$

for each value of the complexity parameter θ.[1]

The *one standard error rule* is to choose the simplest model (*e.g.*, with lowest polynomial degree) within one standard error of the minimal error.

[1] Strictly speaking, this estimate is not quite valid. Why?

k-fold cross-validation in classification

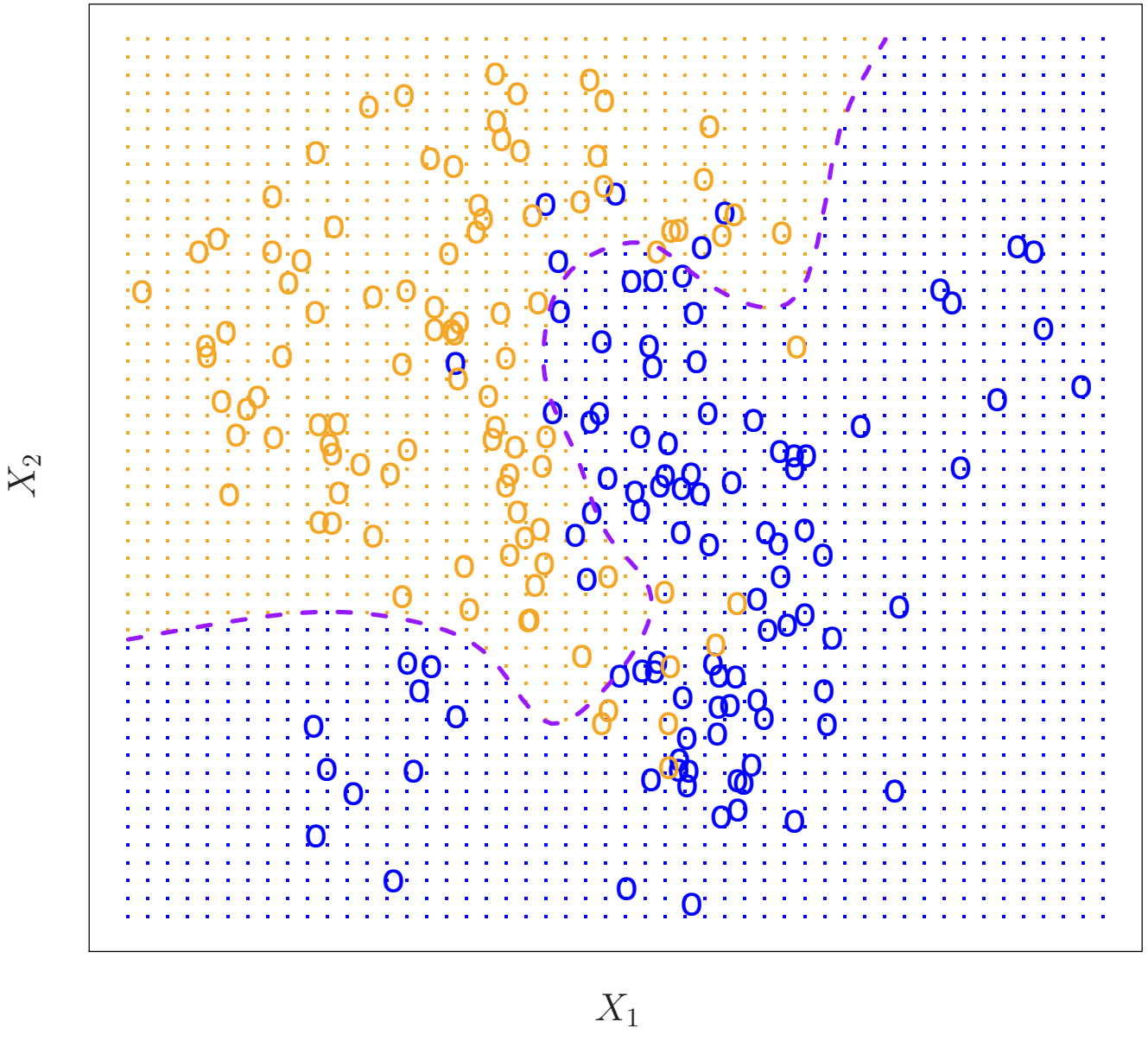
What do we need to change from our regression set-up?

- For LOOCV $\hat{y}_i$ is the fit for observation i obtained from the data with observations i removed, and $\text{Err}_i = I(y_i \neq \hat{y}_i)$. LOOCV is then

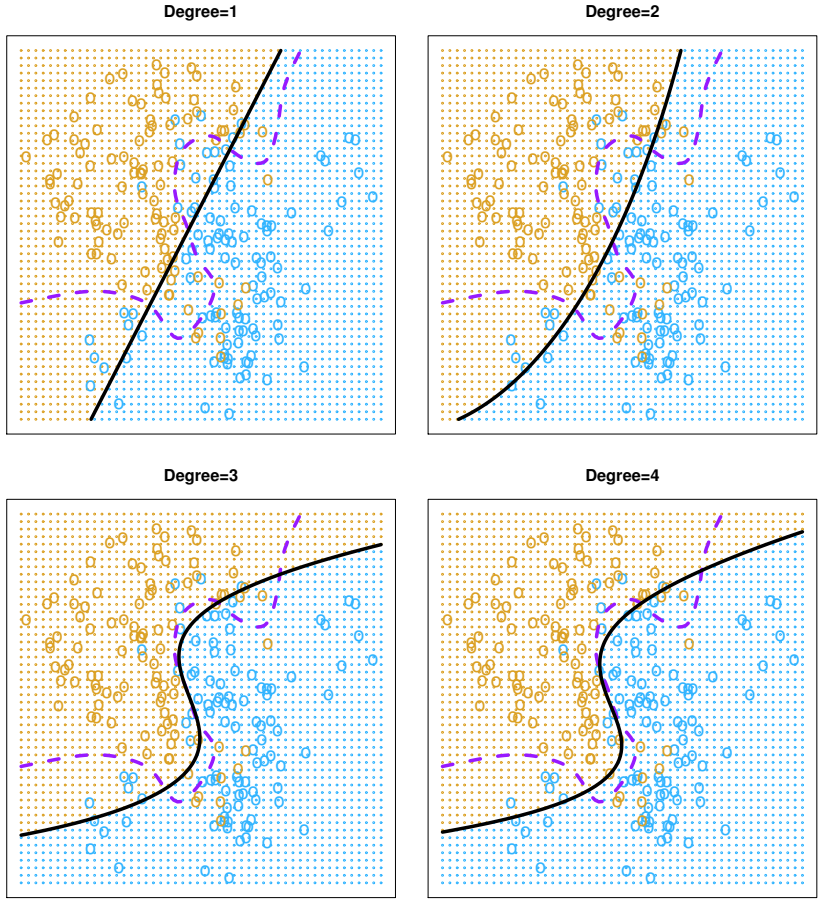
$$\text{CV}_n = \frac{1}{n} \sum_{i=1}^{n} \text{Err}_i$$

- The k-fold CV is defined analogously.
- Chapter 5.1.5 in the course book.

Let us look at a data set from Chapter 2:



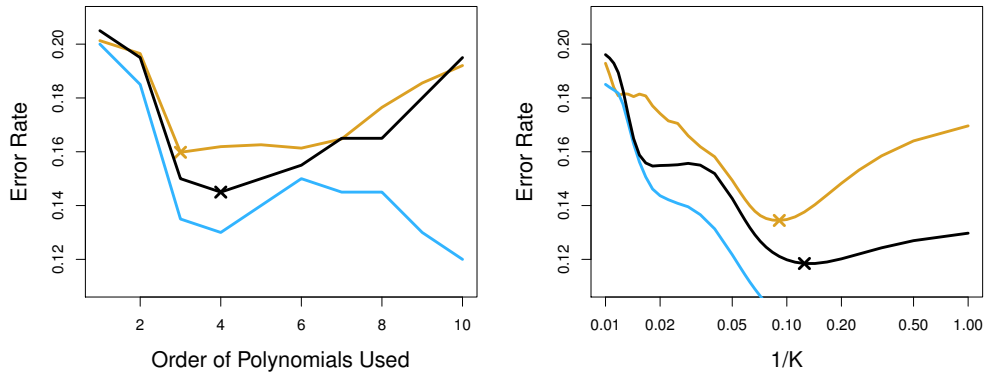
We can fit logistic regression and look at the decision boundary, depending on the degree of of the regression model:



- Error rates: 0.201, 0.197, 0.160, 0.162, respectively
- Bayes error rate: 0.133

Which degree of the logistic regression polynomial is best?

$\rightarrow$ Cross-validation!



- Orange: True test error.
- Black: 10-fold CV error.
- Blue: Training error.

Can we use CV for model assessment?

- Assume we already have our model (maybe found by using the methods from Module 6), so we want to perform model assessment based on all our data.
- Then we can use CV with all data (then the validation part is really the test part) and report on the model performance using the validation parts of the data as above.

Can we use CV both for model selection and model assessment?

- Not really: using the test set for both model selection and estimation tends to overfit the test data, and the bias will be underestimated.
- Solution: you can use two layers of CV - also called *nested CV*. See drawing in class.

The right and the wrong way to do cross-validation

Example from the online lecture by Hastie and Tibshirain (slide 17 for part 5):

- Two-class problem, would like to use a simple classification method.
- Many possible predictors (e.g., $p = 5000$) and not a big sample size (e.g., $n = 50$).

Strategy:

1. Calculate the correlation between the class label and each of the p predictors, and choose the $d = 25$ predictors that have the strongest correlation with the class label.
2. Fit our classifier (logistic regression) using only the $d = 25$ predictors.

How can we use cross-validation to produce an estimate of the performance of this classifier?

Q: Can we apply cross-validation only to step 2? Why (not)?

A: No, **step 1 is part of the training procedure** (the class labels have already been used) and must be part of the CV to give an honest estimate of the performance of the classifier.

- Wrong: Apply cross-validation in step 2.
- Right: Apply cross-validation to steps 1 and 2.

Note: We will see in the Recommended Exercises that doing the wrong thing can give a misclassification error approximately 0 - even if the "true" rate is 50%.

Selection bias in gene extraction on the basis of microarray gene-expression data

Article by Christophe Ambroise and Geoffrey J. McLachlan, PNAS 2002.

See also this nice anecdote at about 7min 15 in the video.

The Bootstrap

- Flexible and powerful statistical tool that can be used to quantify *uncertainty* associated with an estimator or statistical learning method.
- Very popular to obtain standard errors or confidence intervals for a coefficient, when parametric theory does not provide it.
- We will look at getting an estimate for the standard error of a sample median and of a regression coefficient.

- The inventor: Bradley Efron in 1979 - see interview.
- The name? *To pull oneself up by one's bootstraps* from "The Surprising Adventures of Baron Munchausen" by Rudolph Erich Raspe:

  The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.

- **Idea: Use the data itself to get more information about a statistic (an estimator).**

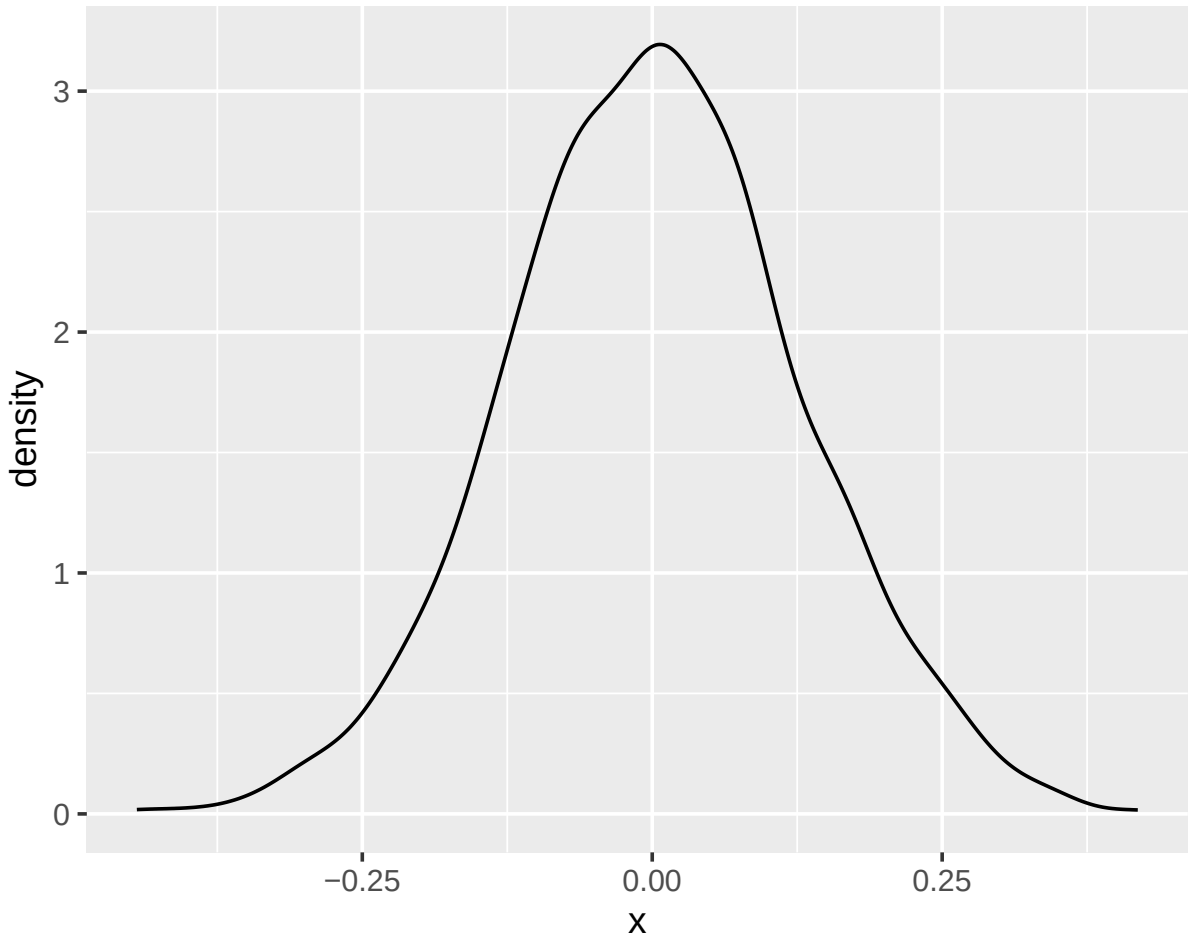
Example: the standard deviation of the sample median?

- Assume that we observe a random sample $X_1, X_2, \ldots, X_n$ from an unknown probability distribution f. We are interested in saying something about the population median, thus we calculate the sample median $\tilde{X}$.
  $\rightarrow$ Q: How accurate is $\tilde{X}$ as an estimator?
- If we would know our distribution F, we could sample from F, and use simulations to answer our question.
- However, without knowledge of the distribution, we cannot calculate the standard deviation of our estimator, thus $\mathrm{SD}(\tilde{X})$.
- That's where the bootstrap method comes into play.

Let's first assume we would know f, for example $X \sim N(0, 1)$. Then we can repeatedly take samples and calculate the standard deviation of all medians to obtain an estimate:

```
set.seed(123)
n = 101
B = 1000
estimator = rep(NA, B)
for (b in 1:B) {
    xs = rnorm(n)
    estimator[b] = median(xs)
}
sd(estimator)
```

```
## [1] 0.1259035
```


3
2
1
0
density
−0.25
0.00
0.25
x

Moving from simulation to bootstrapping (f unknown)

- The bootstrap method is using the observed data to estimate the *empirical distribution* $\hat{f}$, that is each observed value of x is given probability $1/n$.
- A *bootstrap sample* $X_1^*, X_2^*, \ldots, X_n^*$ is a random sample drawn from $\hat{f}$.
- A simple way to obtain the bootstrap sample is to *draw with replacement from* $X_1, X_2, \ldots, X_n$.
- **Note**: Our bootstrap sample consists of n members of $X_1, X_2, \ldots, X_n$ - some appearing more than once, other not appearing at all.

Compare the sample median

```
set.seed(123)
n = 101
original = rnorm(n)
median(original)
```

```
## [1] 0.05300423
```

to the median from **one bootstrap-sample**:

```
boot1 = sample(x = original, size = n, replace = TRUE)
median(boot1)
```

```
## [1] -0.02854676
```

However, drawing only *one* such sample does not help much.

The bootstrap algorithm for estimating standard errors

1. Drawing B bootstrap samples: drawn with replacement from the original data.
2. Evaluate the statistic of interest on *each of the B bootstrap samples* to get $\tilde{X}_b^*$ for the bth bootstrap sample.
3. Estimate squared standard error by

$$\sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\tilde{X}_b^* - \frac{1}{B}\sum_{b=1}^{B}\tilde{X}_b^*)^2} \ ,$$

   which is the empirical standard deviation from the B estimates $\tilde{X}_b^*$, $b = 1, \ldots, B$.

Illustration for the median example (with a for-loop in R)

Simulate data:

```
set.seed(123)
n = 101
original = rnorm(n)
median(original)
```

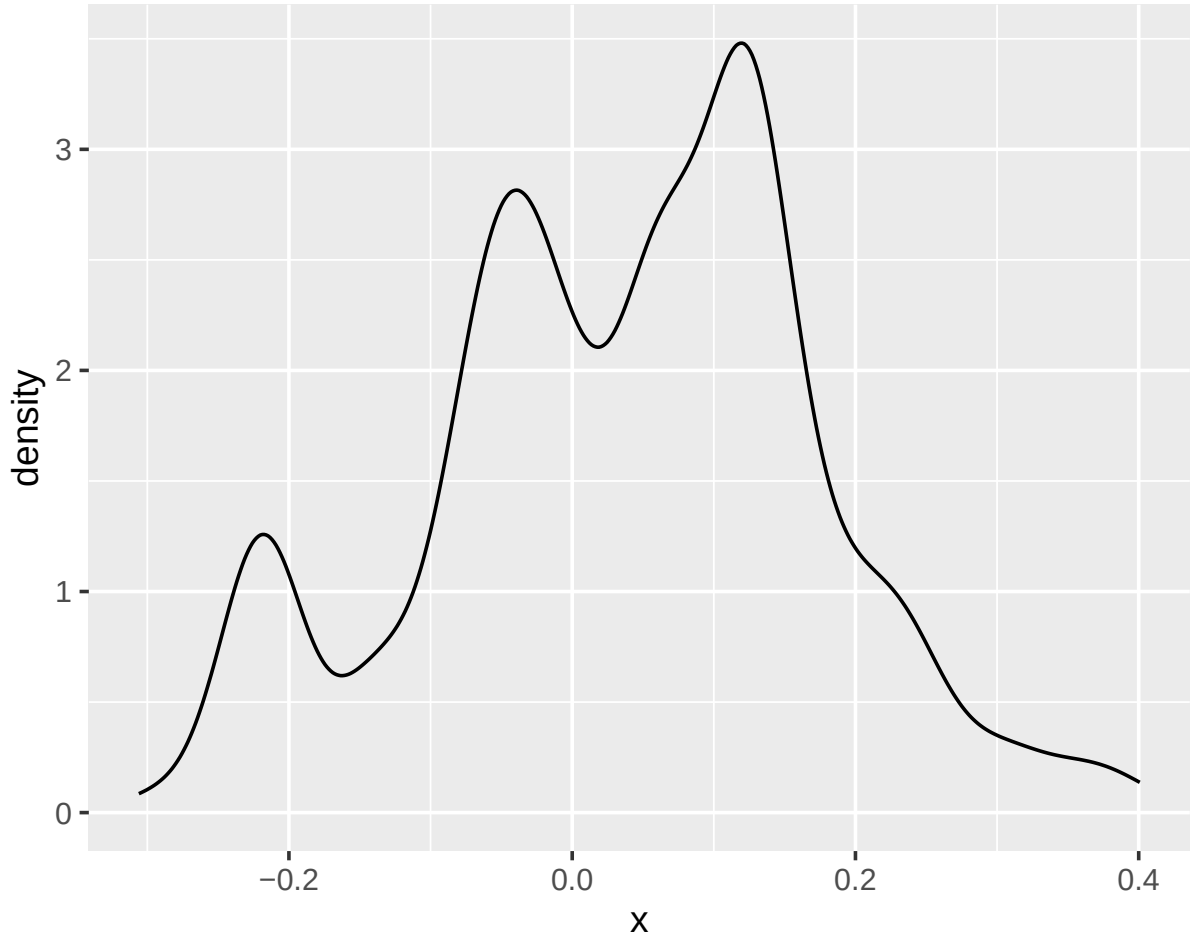
```
## [1] 0.05300423
```

Then take 1000 bootstrap samples and repeatedly estimate the median to get a standard deviation:

```
B = 1000
estimator = rep(NA, B)
for (b in 1:B) {
    thisboot = sample(x = original, size = n, replace = TRUE)
    estimator[b] = median(thisboot)
}
sd(estimator)
```

```
## [1] 0.1365448
```

The distribution of the 1000 sampled estimates:



Alternative: the built-in `boot` function from library `boot`

```
library(boot)
boot.median = function(data, index) return(median(data[index]))
B = 1000
boot(original, boot.median, R = B)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = original, statistic = boot.median, R = B)
##
##
## Bootstrap Statistics :
##       original      bias    std. error
## t1* 0.05300423 -0.01602688   0.1310411
```

With or without replacement?

In bootstrapping we sample *with replacement* from our observations.

Q: What if we instead sample *without replacement*?

Example: multiple linear regression

We assume, for observation i:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \varepsilon_i,$$

where $i = 1, 2, ..., n$. The model can be written in matrix form:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon.$$

The least squares estimator: $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ has
$\text{Cov}(\beta) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$.
In the recommended exercises we will look at how to use bootstrapping to estimate the covariance of the estimator.[2]

Note: Our bootstrap samples can also be used to make confidence intervals for the regression coefficients or prediction intervals for new observations.

[2]Why is that "needed" if we already know the mathematical formula for the standard deviation? Answer: not needed - but OK to look at an example where we know the "truth".

- Look at the lab in the course book Section 5.3.4.
- In particular: “Estimating the Accuracy of a Linear Regression Model”.
- Why do bootstrap standard errors and theoretical standard errors not always correspond? What does that mean?

A related method: Bagging

Bagging (*bootstrap aggregation*) is a special case of *ensemble methods*.

- In Module 8 we will look at bagging, which is built on bootstrapping and the fact that it is possible to reduce the variance of a prediction by taking the average of many model fits.
- Particularly useful for estimation methods with large variances (like regression trees).
- Idea:
  - Draw B bootstrap samples from your data and train the method for each sample b in order to get $\hat{f}^{\star b}(x)$.
  - Average over all predictions to obtain

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{\star b}(x) \ .$$

- Like this, we obtain a new model that has a smaller variance than each of the individual model. If the bootstrap samples were independent (which they are of course not), the variance (thus prediction error) would be reduced by

$$\mathrm{Var}(\bar{X}) = \frac{\sigma^2}{B} \ .$$

- In reality, the variance reduction is less. For a pairwise correlation ρ we would have $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$.

- Models that have poor prediction ability (as we may see can happen with regression and classification trees) might benefit greatly from bagging. More in Module 8.

Further reading

- Videoes on YouTube by the authors of ISL, Chapter 5, and corresponding slides
- Solutions to exercises in the book, chapter 5

Excursion: R Markdown and `knitr`

- You can work through the Bonus part about R Markdown in the R course https://digit.ntnu.no/courses/course-v1:NTNU+IMF001+2020/course/.
- You find a template for the exercise sheets on our course website https://wiki.math.ntnu.no/tma4268/2023v/subpage6. Do not forget to install the packages listed at the beginning of the exercise sheet.

How to render a document?

You do this by pressing `knit`.
Knitting is also done by: Ctrl+Shift+K (Windows) or Cmd+Shift+K (MacOS).

1) Creating documents with R Markdown starts with an .Rmd file that contains a combination of markdown (content with simple text formatting) and R code chunks.
2) The .Rmd file is fed to `knitr` which executes all of the R code chunks and creates a new markdown (.md) document which includes the R code and it's output.
3) The markdown file generated by `knitr` is then processed by `pandoc` which is responsible for creating a finished web page, PDF, MS Word document, slide show, handout, book, dashboard, package vignette or other format.

- More: About pandoc - the swiss army knife for file conversion
- NB: even if you write tex this is first translated to md and then via pandoc to pdf, so subtle tex stuff may be missed on the way.
- Do you get a separate window popping up, or is your output shown in the Viewer tab of one of the window panes? Go to RStudio-Tools-Global Options-RMarkdown and check what is your value of "show output preview in".

What output type do you want to produce?

- Just keep track of your own work: html_document
- For TMA4268 Compulsory exercise 1: we ask for a pdf-file (because that is easy to read and grade when you upload that to Blackboard)
- To produce a `pdf_document` RStudio (using pandoc) will call a latex-installation, so you need to have latex installed on your laptop to be able to produce a pdf-file.
- Toggle comment/uncomment with hashtag in YAML header `output` to make different options active, then press `knit`. Alterntively this can be done by calling the function `rmarkdown::render()` from your Console window.
- Optional: check that uncommenting `pdf_document` and commenting out `html_document` and pressing `knit` will give you a pdf-file.
- During rendering we use the location of the .Rmd file as the working directory, and the rendering is done in a *new session*.

Formatting your R Markdown file

Text, mathematics

- formatted with markdown
- mathematics (in latex) with formulas starting and ending with one `$` and equation with `$$`
- boldface with two stars and italic with one, new line with two spaces,
- For more, like sections, bulleted or numbered lists, tables, footnotes, rulers,... see https://github.com/rstudio/cheatsheets/raw/master/rmarkdown-2.0.pdf

Hands-on: go the the Compulsory exercise 1 template, and just write and press `knitr` to see!

Check how a nice formula using latex is generated for
$Y_i = \beta_0 + \beta_1 x_{i1} + \varepsilon_i$.

Code Chunks

- Chunks of embedded code. Each chunk begins with “‘{r} and ends with “‘
- Set of code chunk options - but I have mainly used these two:
  - `echo`: display the code in the chunk, TRUE or FALSE or selected lines, or maybe with an R-object (later)
  - `eval`: run code in the chunk, TRUE or FALSE
- Remember to include packages to be used within the chunk (only needed the first time in a document, if the chung is evaluated, `eval=TRUE`).
- Chunks can have (unique) names, may help when debugging.
- In chunks where figures are generated you can adjust the figure width/height and alignment, for example as

```
{r auto, echo=FALSE, fig.width=4, fig.height=3,fig.align =
"center",out.width='70%'}
```

Set-up chunk

- The set-up chunk is a code chunk that you add before you actually start to do the work.
- Smart things to add to the setup-chunk:

```
library(knitr)
knitr::opts_chunk$set(echo = TRUE, tidy = TRUE, message = FALSE, warning = FALSE,
    strip.white = TRUE, prompt = FALSE, cache = TRUE, size = "scriptsize",
    fig.width = 4, fig.height = 3)
```

Calling R outside of the code chunks

Use the ‘ r before and ‘ after an R command to integrate into the text. For example,

```
2 + 2
```

```
## [1] 4
```

is equal to 4.

This is what we have done in the YAML-header to include todays date on your submission:
‘ r format(Sys.time(), ‘%d %B, %Y’)‘ gives 07 February, 2024.

Problems

When I `knit` with output: pdf_document no pdf-file is produced. Why?

- html_document is more forgiving than pdf_document wrt tex-errors
- a tex-error is not easy to spot - log is terrible
- many students have problems here, and some just end up handing in html or Rmd for the projects

My solution

- first I render html_document and look for tex-errors and fix them
- then I render pdf_document, and include `keep_tex: yes` yaml option
- then I compile the tex in my favorite `texshop` and look for sensible log for errors,
- and then go back to the Rmd and fix the error.

Handing in Compulsory exercise 1

- I hope you have already joined a group from Bb (front page – users and groups – groups).
- Then under Compulsory exercises you will see "Hand in" is possible (this will come).
- Upload **both your Rmd and pdf-version of you R Markdown file** with your solutions to the exercise (based on the template).
- Scores and comments will be given on Bb.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning with Applications in r*. New York: Springer.