Module 5: RESAMPLING

TMA4268 Statistical Learning V2018

Mette Langaas, Department of Mathematical Sciences, NTNU

week 7 2018 (Version 11.02.2018)

Learning material for this module:

- James et al (2013): An Introduction to Statistical Learning. Chapter 5.
- Classnotes 12.02.2018

Move to:

- Introduction
- Cross-validation and Recommended exercises on cross-validation
- Bootstrapping and Recommended exercises on bootstrapping
- Summing up
- Further reading
- Packages to install before knitting this R Markdown file

Introduction

What will you learn?

- What is model assessment and model selection?
- Ideal solution in a data rich situation.
- Validation set - LOOCV and k-fold CV - what is the best?
- Bootstrapping - how and why.
- Summing up
- The plan for the interactive lesson on Wednesday/Friday.

Focus: Generalization performance of learning method

- prediction capacity on independent test data
- inference and understanding

This is important both for

Model selection:

estimate the *performance* of different models (often different order of complexity within one model class) to *choose the best model*.

Model assessment:

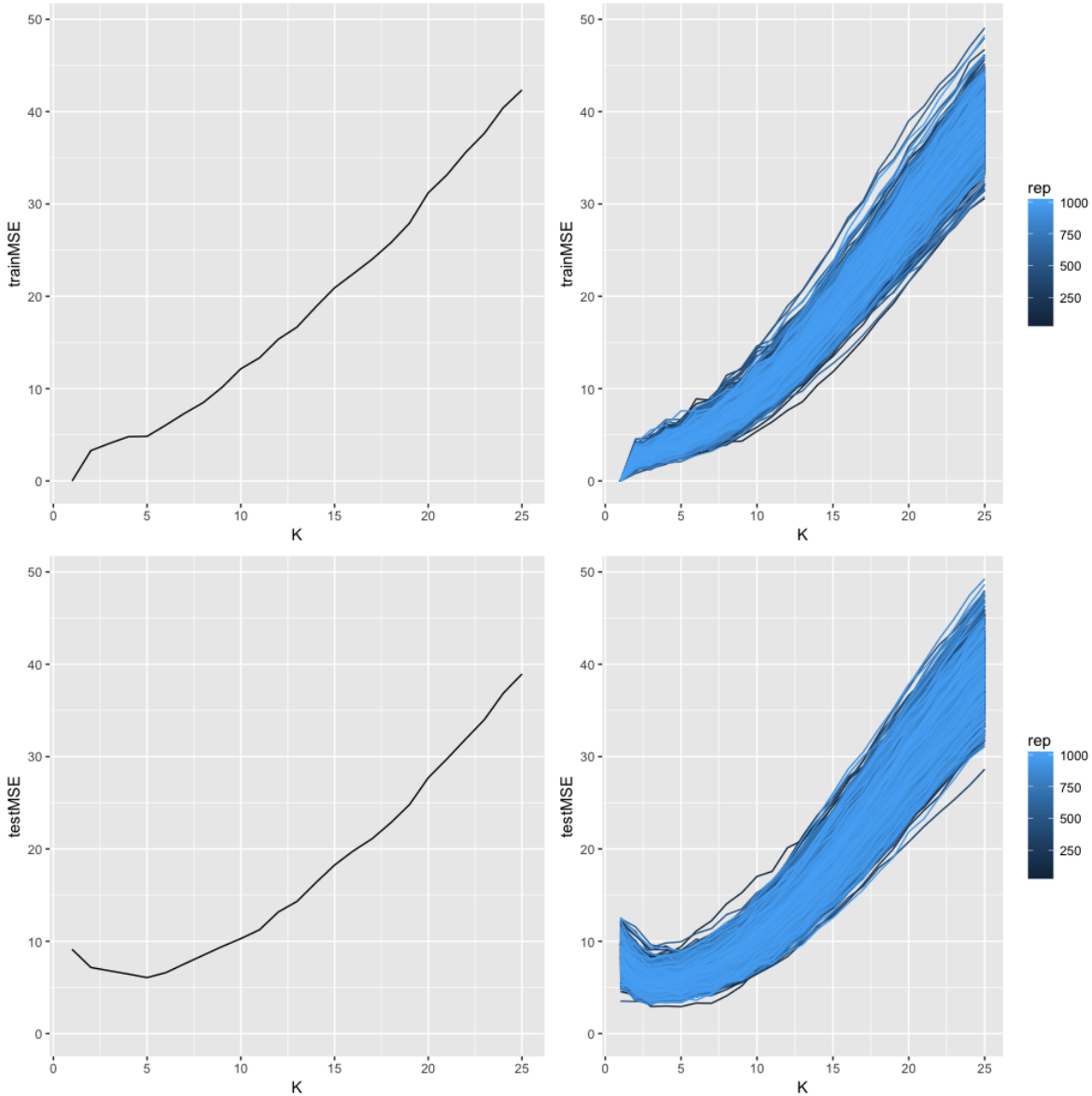
having chosen a final model, estimating its performance (prediction error) on new data.

Example from ComplsoryEx 1- Problem 1

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.

KNN-regression

$n = 61$ both for training and for test data (using same x-grid).



K small: high complexity (left) and K large: low complexity (right)

The bias-variance trade-off

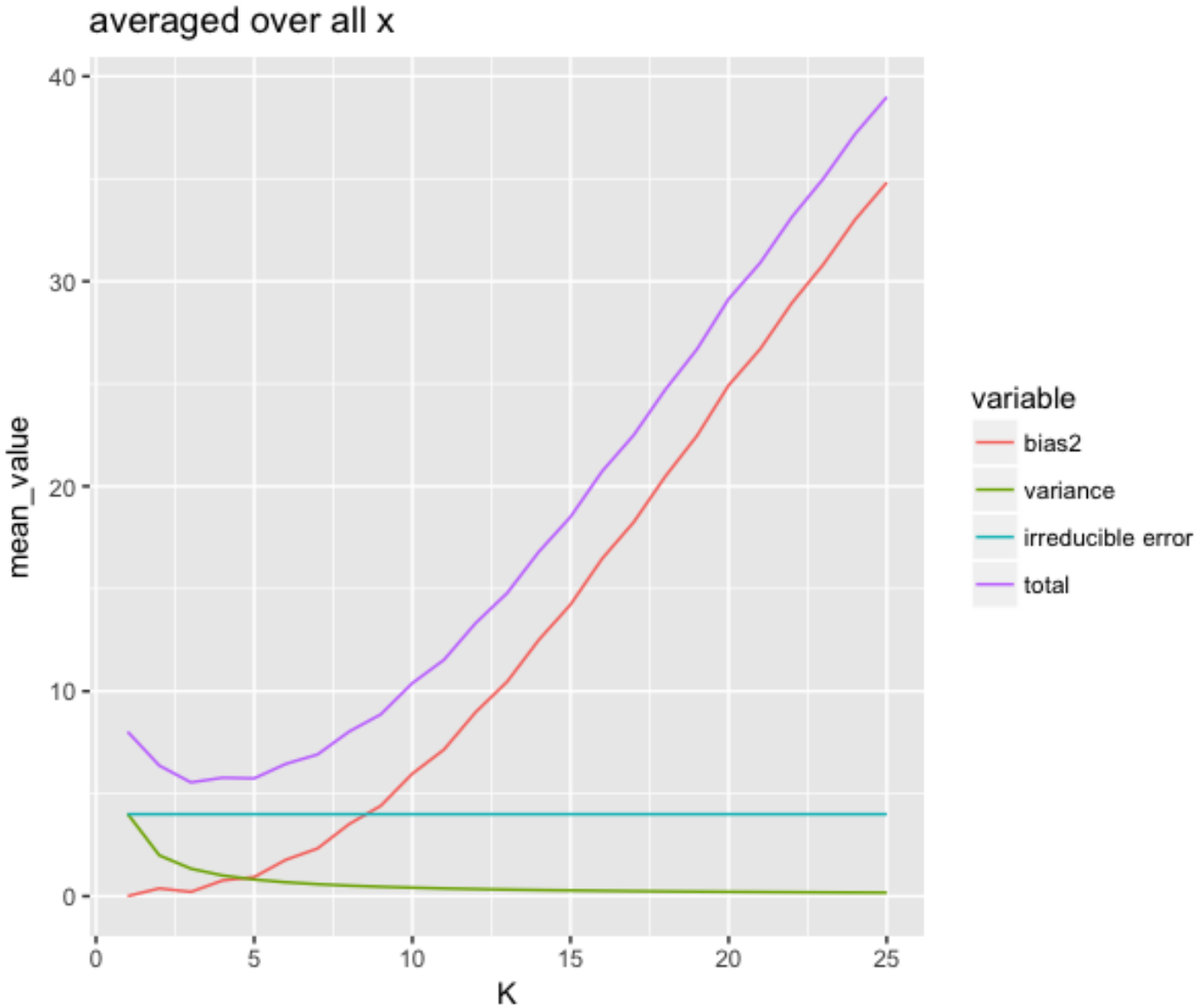


Figure 2: Regression problem: bias-variance traceoff

Loss functions - reminder - we will use

- Mean squared error (quadratic loss) for regression problems:

$$Y_i = f(\mathbf{x}_i) + \varepsilon_i \text{ for } i = 1, \ldots, n \text{ and MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(\mathbf{x}_i))^2$$

- 0/1 loss for classification problems:

$$P(Y = j \mid \mathbf{x}_0) \text{ for } j = 1, \ldots, K \text{ and } \frac{1}{n}\sum_{i=1}^{n} \mathrm{I}(y_i \neq \hat{y}_i)$$

The challenge

our example was based on simulated data, so I had unlimited access to data. Now, let us move to real data!

Data rich situation

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 (aka model selection)
- test set: to assess how well the model fits on new independent data (aka model assessment)

Q Why not enough with training and test?

If this is the case - great - then you do not need Module 5. But, this is very seldom the case - so we will study other solutions based on efficient cample resue with *resampling* data.

An alternative strategy for model selection (using methods penalizing model complexity) is covered in Module 6.

First we look at *crossvalidation*, then at *bootstrapping*.

Cross-validation (CV)

- the validation set approach
- leave one out cross validation (LOOCV)
- 5 and 10 fold crossvalidation (CV)
- selection bias - all elements of a model selection strategy need to be within the CV-loop
- recommended exercises

The validation set approach

Consider the case when you have a data set consisting of n observations.

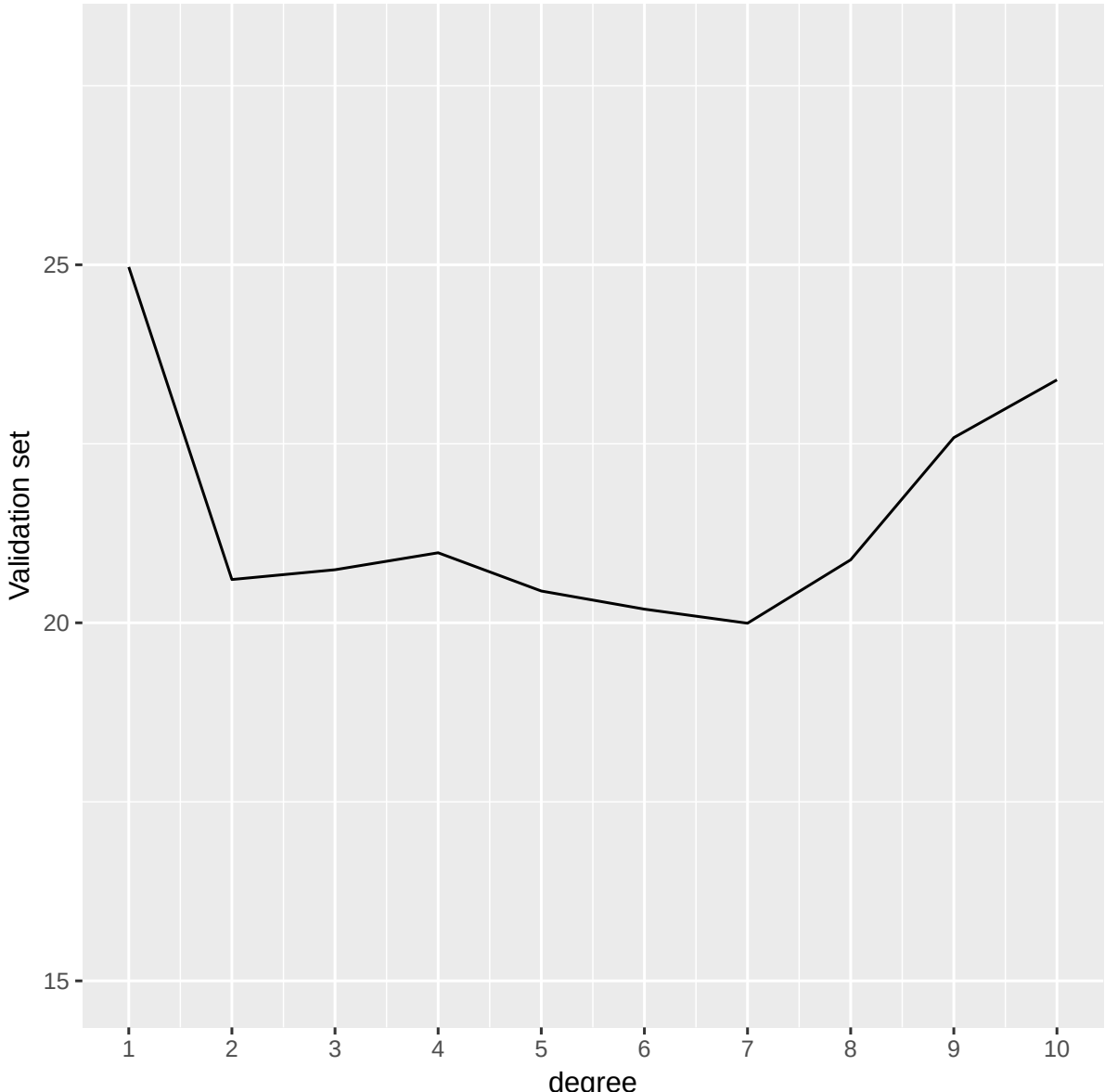
To fit a model and to test its predictive performance you randomly divide the data set into two parts ($n/2$ sample size each):

- 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)

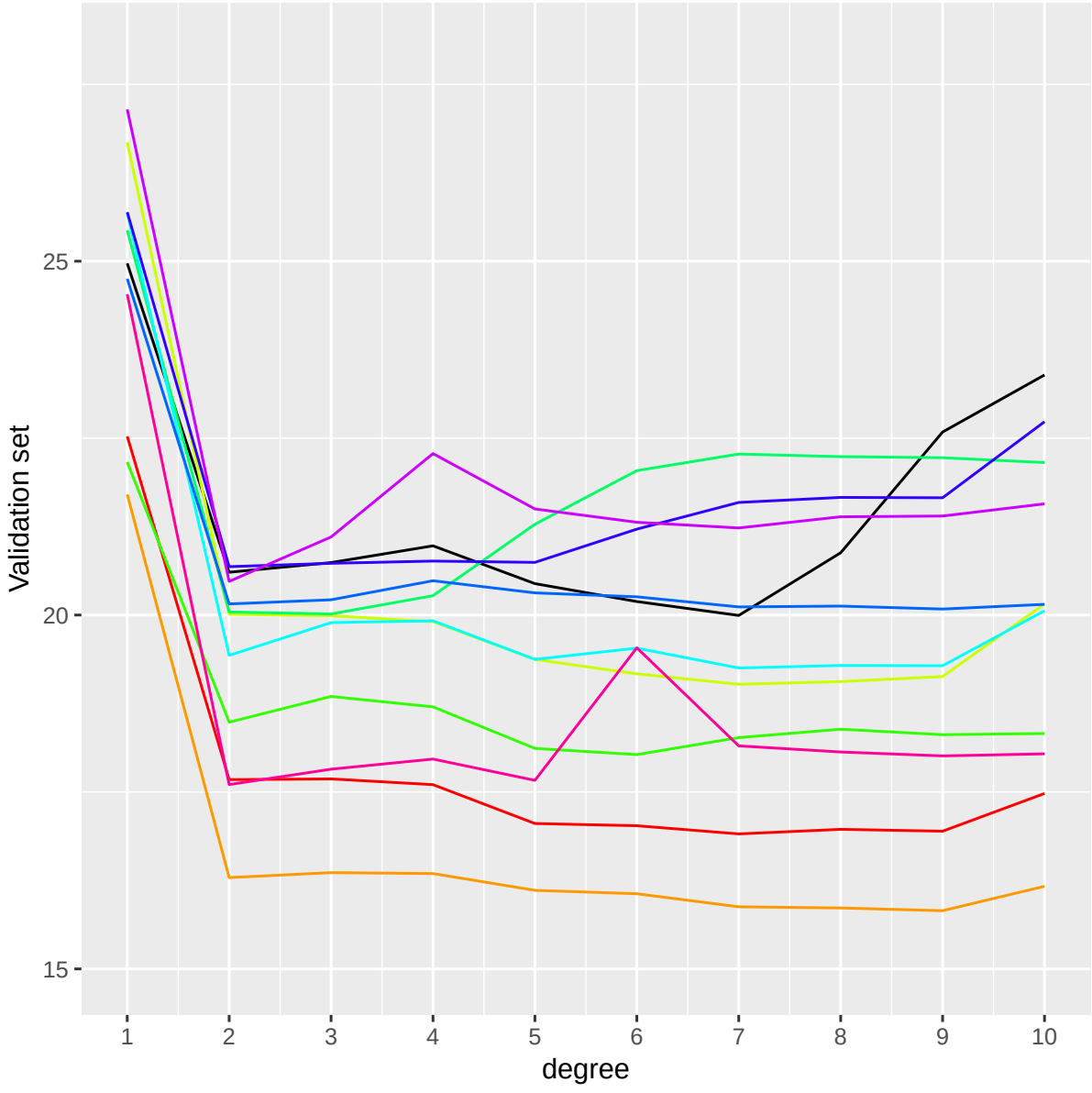
First: focus on model selection

Regression model selection example: validation set error

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



Regression example: validation set error for many random divisions



Drawbacks with the validation set approach

- high variability of validation set error (which we think of as estimate for test set error) - since this is dependent on which observation are included in the training and validation set
- smaller sample size for model fit - since not all observations can be in the training set
- the validation set error may tend to overestimate the test set error for a model that is fit on the full data set (because - the more data the lower error, and here our training set is half of our data set).

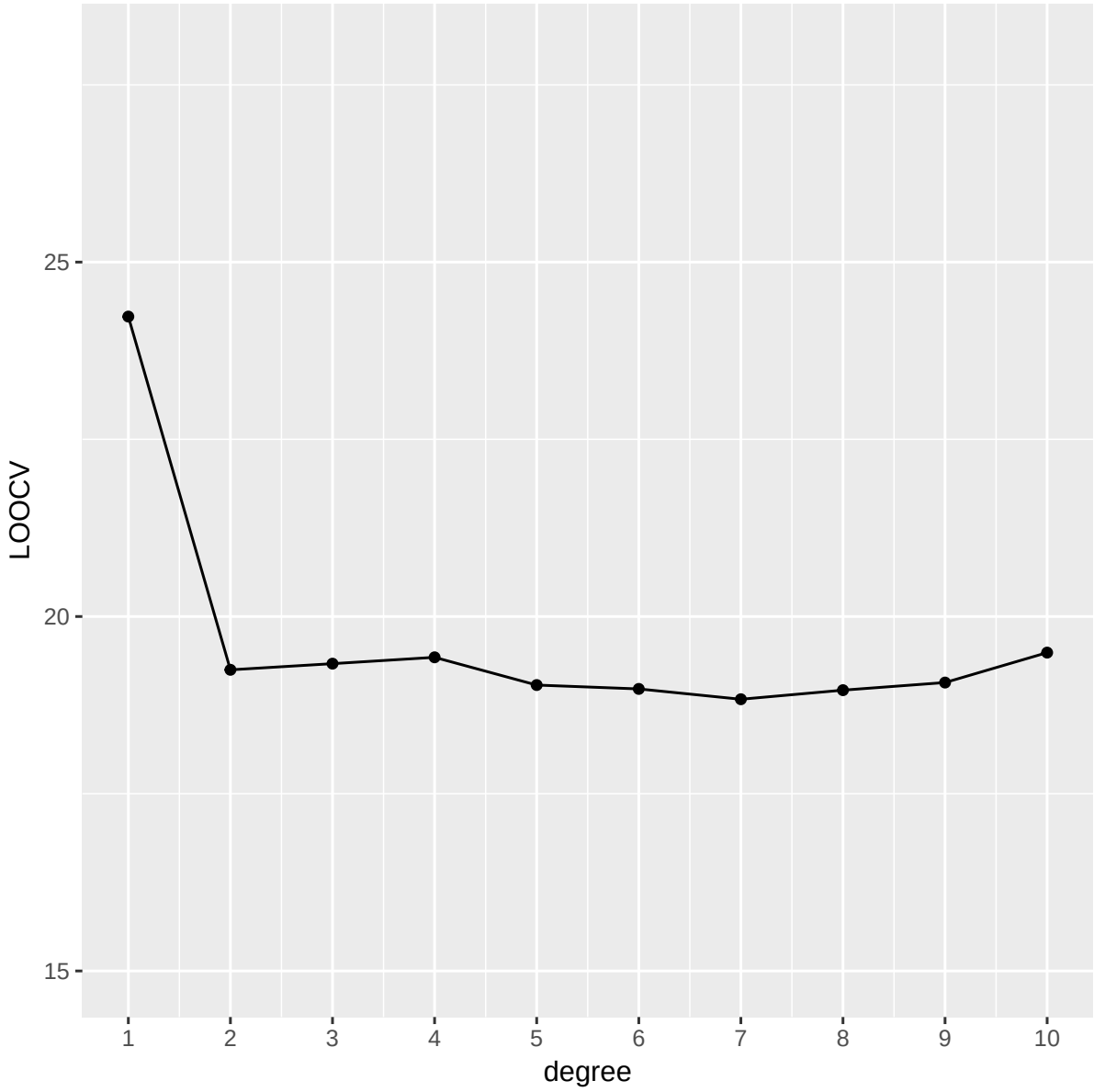
Leave-one-out cross-validation (LOOCV)

- If the data is very limited and the division of the data into two parts is unreasonable, leave-one-out cross-validation (LOOCV) can be used.
- In LOOCV one observation at a time is left out and makes up the test set.
- 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.
- The total prediction error is the mean across these n models.

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

$$CV_n = \frac{1}{n} \sum_{i=1}^{n} \text{MSE}_i$$

Regression example: LOOCV



```r
library(ISLR)
library(boot)
library(ggplot2)
set.seed(123)
n=dim(Auto)[1]
testMSEvec=NULL
start=Sys.time()
for (polydeg in 1:10)
  {
    glm.fit=glm(mpg~poly(horsepower,polydeg),data=Auto)
    glm.cv1=cv.glm(Auto, glm.fit,K=n)
    testMSEvec=c(testMSEvec,glm.cv1$delta[1])
}
stopp=Sys.time()
yrange=c(15,28)
plotdf=data.frame("testMSE"=testMSEvec,"degree"=1:10)
g0=ggplot(plotdf,aes(x=degree,y=testMSE))+geom_line()+geom_point()+scale_y_continuous(limits = yrange)+sca
g0
```

Issues with leave-one-out cross-validation

- Good:
  - no randomness in training/validation splits!
  - little bias since nearly the whole data set used for training (compared to half for validation set approach)
- Bad:
  - expensive to implement - need to fit n different models - however nice formula for linear model LOOCV - but not genereally so
  - high variance since: two training sets only differ by one observation - which makes estimates from each fold are highly correlated and this can lead to that their average can have high variance.

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

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

LOOCV for multiple linear regression

$$CV_n = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2$$

where h_i is the ith diagonal element (leverage) of the hat matrix $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.

k-fold cross-validation

Formally

- Indices of observations - divided into k folds: $C_1, C_2, \ldots, C_k$.
- n_k elements in each fold, if n is a multiple of k then $n_k = n/k$.

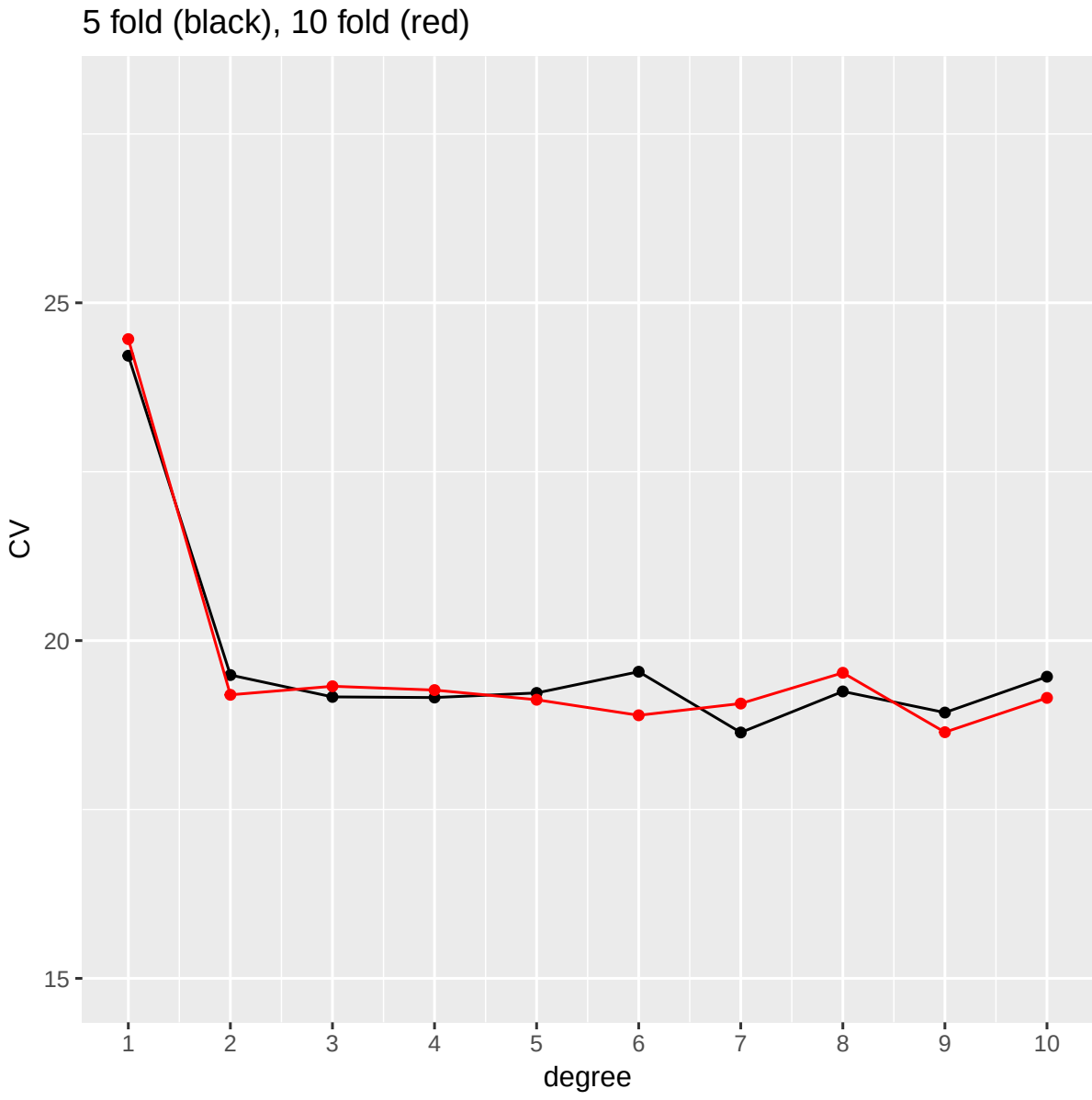
$$\text{MSE}_k = \frac{1}{n_k} \sum_{i \in C_k} (y_i - \hat{y}_i)^2$$

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

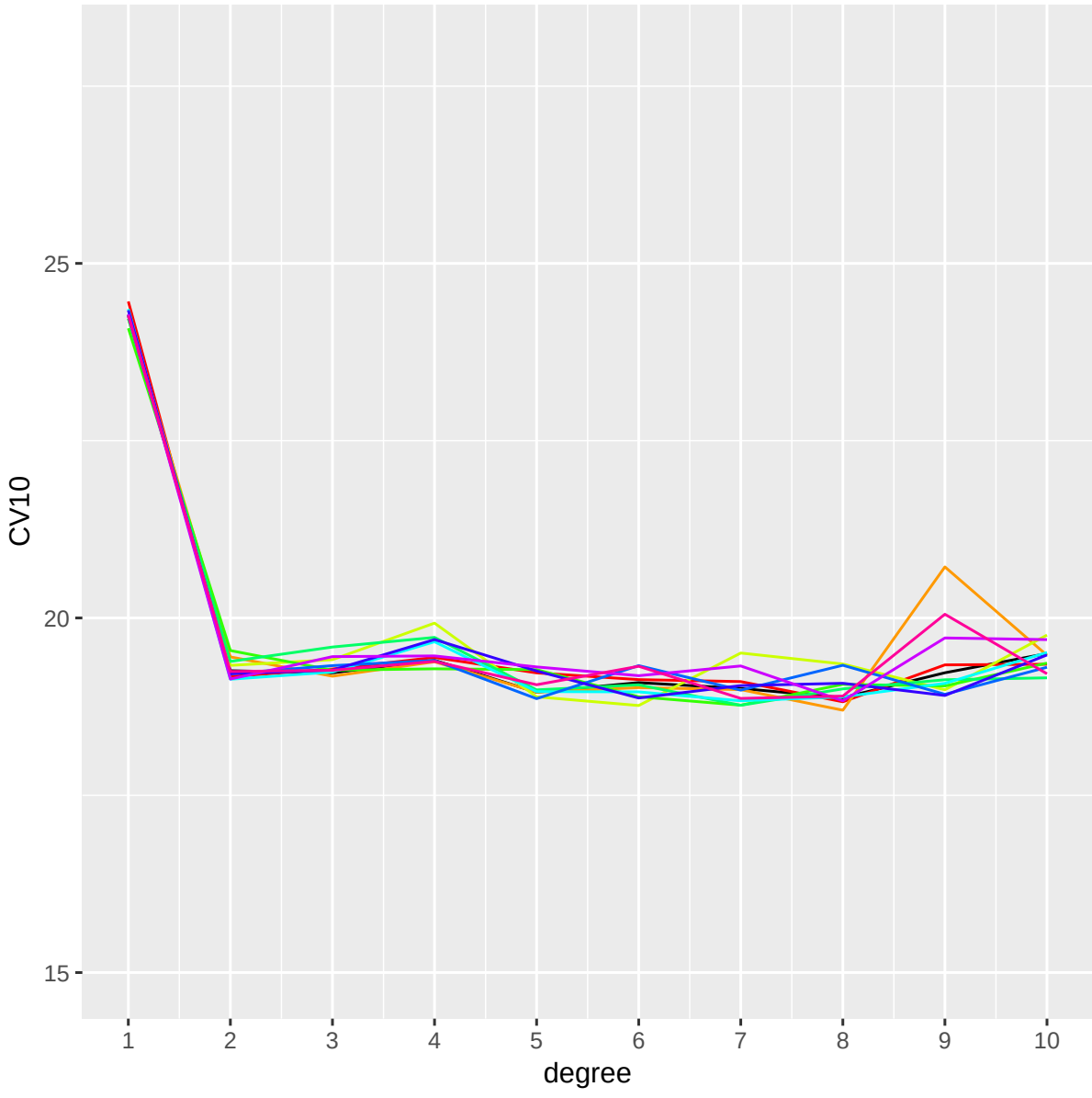
$$CV_k = \frac{1}{n} \sum_{i=j}^{k} n_k \text{MSE}_k$$

Observe: setting $k = n$ gives LOOCV.

Regression example: 5 and 10-fold cross-validation



```r
library(ISLR)
library(boot)
library(ggplot2)
set.seed(123)
n=dim(Auto)[1]
testMSEvec5=NULL
testMSEvec10=NULL
start=Sys.time()
for (polydeg in 1:10)
  {
    glm.fit=glm(mpg~poly(horsepower,polydeg),data=Auto)
    glm.cv5=cv.glm(Auto, glm.fit,K=5)
    glm.cv10=cv.glm(Auto, glm.fit,K=10)
    testMSEvec5=c(testMSEvec5,glm.cv5$delta[1])
    testMSEvec10=c(testMSEvec10,glm.cv10$delta[1])
}
stopp=Sys.time()
yrange=c(15,28)
plotdf=data.frame("testMSE5"=testMSEvec5,"degree"=1:10)
g0=ggplot(plotdf,aes(x=degree,y=testMSE5))+geom_line()+geom_point()+scale_y_continuous(limits = yrange)+sc
g0+geom_line(aes(y=testMSEvec10),colour="red")+geom_point(aes(y=testMSEvec10),colour="red")+ggtitle("5 fol
```


25
20
15
CV10
1
2
3
4
5
6
7
8
9
10
degree

Issues with k-fold cross-validation

1. As for the validation set, 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. The training set is (k-1)/k of the original data set - this will estimated of prediction error that are biased upwards.
3. This bias is the smalles when $k = n$ (LOOCV), but we know that LOOCV has high variance.
4. This is way often $k = 5$ or $k = 10$ is used as a compromise between 2 and 3.

k-fold cross-validation in classification

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

The right and the wrong way to do cross-validation

ISL book slides, page 17: model assessment.

- We have a two-class problem and would like to use a simple classification method, however,
- we have many possible predictors $p = 5000$ and not so big sample size $n = 50$.

We use this strategy to produce a classifier:

1. We calculate the correlation between the class label and each of the p predictors, and choose the $d = 25$ predictors that have the highest (absolute value) correlation with the class label. (We need to have $d < n$ to fit the logistic regression uniquely.)
2. Then we fit our classifier (here: 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 or why not?

Can we apply cross-validation only to step 2?

A: No, step 1 is part of the training procedure (the class labels are 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.

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: Direct quotation from the abstract of the article follows.

- In the context of cancer diagnosis and treatment, we consider the problem of constructing an accurate prediction rule on the basis of a relatively small number of tumor tissue samples of known type containing the expression data on very many (possibly thousands) genes.
- Recently, results have been presented in the literature suggesting that it is possible to construct a prediction rule from only a few genes such that it has a negligible prediction error rate.
- However, in these results the test error or the leave-one-out cross-validated error is calculated without allowance for the selection bias.

- There is no allowance because the rule is either tested on tissue samples that were used in the first instance to select the genes being used in the rule or because the cross-validation of the rule is not external to the selection process; that is, gene selection is not performed in training the rule at each stage of the crossvalidation process.
- We describe how in practice the selection bias can be assessed and corrected for by either performing a crossvalidation or applying the bootstrap external to the selection process.
- We recommend using 10-fold rather than leave-one-out cross-validation, and concerning the bootstrap, we suggest using the so-called .632 bootstrap error estimate designed to handle overfitted prediction rules.
- Using two published data sets, we demonstrate that when correction is made for the selection bias, the cross-validated error is no longer zero for a subset of only a few genes.

Recommended exercises on cross-validation

Problem 1: Explain how k-fold cross-validation is implemented + draw a figure, + specify algorithmically what is done, + and in particular how the "results" from each fold are aggregated, + relate to one example from regression (maybe is complexity wrt polynomials of increasing degree in multiple linear regression or K in KNN-regression?) + relate to one example from classification (maybe is complexity wrt polynomials of increasing degree in logistic regression or K in KNN-classification?)

Hint: the words "loss function", "fold", "training", "validation" are central.

Problem 2: What are the advantages and disadvantages of k-fold cross-validation relative to + the validation set approach + leave one out cross-validation (LOOCV) + what are recommended values for k, and why?

Hint: the words "bias", "variance" and "computational complexity" should be included.

Problem 3:. Selection bias and the "wrong way to do CV".

The task here is to devise an algorithm to "prove" that the wrong way is wrong and that the right way is right.

1. What are the steps of such an algorithm? Write down a suggestion. Hint: how do you generate data for predictors and class labels, how do you do the classification task, where is the CV in the correct way and wrong way inserted into your algorithm? Can you make a schematic drawing of the right and the wrong way? Hint: ISL book slides, page 20+21 - but you can do better?

2. One possible version of this is presented in the R-code below. Go through the code and explain what is done in each step, then run the code and observe if the results are in agreement with what you expected. Make changes to the R-code if you want to test out different strategies.

```
library(boot)
# GENERATE DATA
# reproducible
set.seed(4268)
n=50 #number of observations
p=5000 #number of predictors
d=25 #top correlated predictors chosen
kfold=10
#generating predictor data
xs=matrix(rnorm(n*p,0,4),ncol=p,nrow=n) #simple way to to t
dim(xs) # n times p
# generate class labels independent of predictors - so if c
ys=c(rep(0,n/2),rep(1,n/2)) #now really 50% of each
table(ys)
```

Problem 4: Trying out different versions of cross-validation with R. We will use a simulated data set (so that in the end the truth can be reviled and you can see how well you have done). To see differences we will look at a large data set and a small data set (in the number of observations), and focus on regression and on classification.

The Bootstrap

- flexible and powerful statistical tool that can be used to quantify *uncertainty* associated with an estimator or statistical learning method
- we will look at getting an estimate for the standard error of a sample median and of a regression coefficient
- in Module 8 - bootstrapping is the core of the *ensemble method* referred to at *bagging*=bootstrap aggregation,
- in TMA4300 Computation statistics - more on the bootstrap.

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 interesting in saying something about the population median, and to do that we calculate the sample median $\tilde{X}$. But, how accurate is $\tilde{X}$ as an estimator?

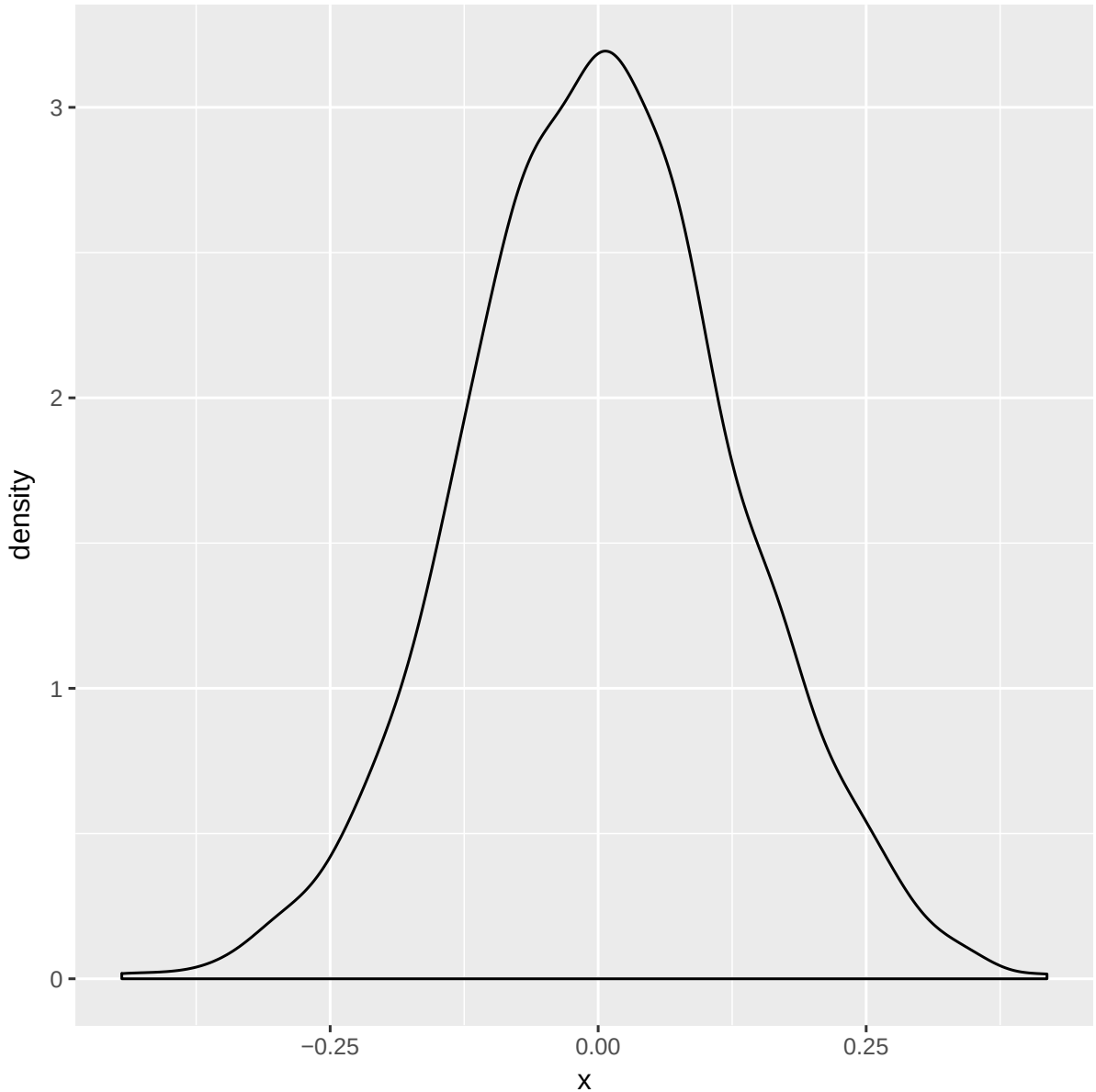
The bootstrap was introduced as a computer-based method to estimate the standard deviation of an estimator, for example our estimator $\tilde{X}$.

But, before we look at the bootstrap method, first we assume that we know F and can sample from F, and use simulations to answer our question.

```r
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)
# approximation for large samples
# (sd of median of standard normal)
1.253*1/sqrt(n)
```

```
## [1] 0.1259035
## [1] 0.1246782
```

```
ggplot(data=data.frame(x=estimator),aes(x=x))+
  geom_density()
```



Moving from simulation to bootstrapping

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* $X_1, X_2, \ldots, X_n$.

This means that our bootstrap sample consists of n members of $X_1, X_2, \ldots, X_n$ - some appearing 0 times, some 1, some 2, etc.

```r
set.seed(123)
n=101
original=rnorm(n)
median(original)
boot1=sample(x=original,size=n,replace=TRUE)
table(table(boot1))
n-sum(table(table(boot1)))
median(boot1)
```

```
## [1] 0.05300423
##
##  1  2  3  4  5
## 42 16  6  1  1
## [1] 35
## [1] -0.1388914
```

The bootstrap algorithm for estimating standard errors

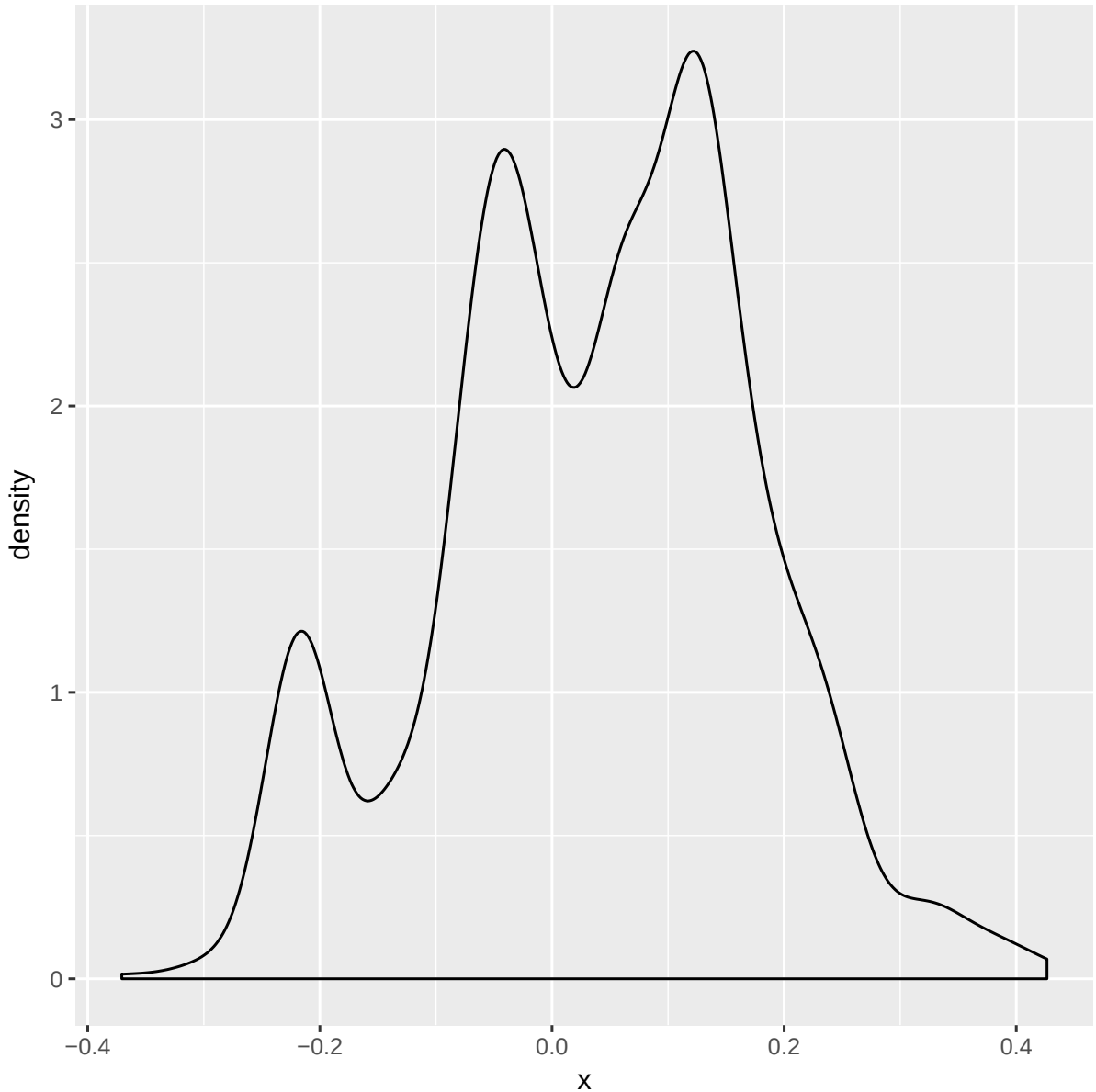
1. B bootstrap samples:
2. Evaluate statistic:
3. Estimate standard error by:

with for-loop in R

```
set.seed(123)
n=101
original=rnorm(n)
median(original)
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] 0.05300423
## [1] 0.1365856
```

```
ggplot(data=data.frame(x=estimator),aes(x=x))+
  geom_density()
```



using built in boot function from library boot

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

```
## [1] 0.05300423
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -2.30917 -0.50232   0.05300   0.08248   0.68864   2.18733
```

```r
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 -0.01413291    0.136591
```

With or without replacement?

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

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 $\text{Cov}(\beta) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$.

We will in the recommended exercises look at how to use bootstrapping to estimate the covariance of the estimator. Why is that "needed" if we already know the mathematical formula for the standard deviation?

We will not do this here - but our bootstrap samples can also be used to make confidence intervals for the regression coefficients or prediction intervals for new observations. This means that we do not have to rely on assuming that the error terms are normally

Bagging

Bagging is a special case of *ensemble methods*.

In Module 8 we will look at bagging, which is built on bootstrapping the the fact that it is possible to reduce the variance of a prediction by taking the average of many model fits.

- Assume that we have B different predictors $X_1, X_2, \ldots, X_B$. We have built them on B different bootstrap samples.
- All are predictors for some parameter μ and that all have some unknown variance σ^2.
- We then decide that we want to use all the predictors together - equally weighted - and make $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$, which we often use to predict μ.

We can therefore obtain a new model (our average of the individual models) that has a smaller variance than each of the individual model because

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

Since this averaged predictor has smaller variance than each of the predictors we would assum that this is a more accurate prediction. However, the interpretation of this bagged prediction might be harder then for the separate predictors.

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.

Recommended exercises on bootstrapping

Problem 1: We will calculate the probability that a given observation in our original sample is part of a bootstrap sample. This is useful for us to know in Module 8.

Our sample size is n.

1. We draw one observation from our sample. What is the probability of drawing observation x_i? And of not drawing observation x_i?
2. We make n independent drawing (with replacement). What is the probability of not drawing observation x_i in any of the n drawings? What is then the probability that x_i is in our bootstrap sample (that is, more than 0 times)?
3. When n is large $(1 - \frac{1}{n})^n = \frac{1}{\exp(1)}$. Use this to give a numerical value for the probability that a specific observation x_i is in our bootstrap sample.
4. Write a short R code chunk to check your result. (Hint: An example on how to this is on page 198 in our ISLR book.)

Problem 2: Explain with words and an algorithm how you would proceed to use bootstrapping to estimate the standard deviation of one of the regression parameters in multiple linear regression. Comment on which assumptions you make for your regression model.

Problem 3: Implement your algorithm from 2 both using for-loop and using the `boot` function. Hint: see page 195 of our ISLR book. Use our SLID data set and provide standard errors for the coefficient for age. Compare with the theoretical value $(\mathbf{X}^T\mathbf{X})^{-1}\hat{\sigma}^2$ that you find in the output from the regression model.

```
library(car)
library(boot)
SLID = na.omit(SLID)
n = dim(SLID)[1]
SLID.lm = glm(wages~., data = SLID)
summary(SLID.lm)$coeff[3,2]
```

Summing up

Plan for interactive lecture (Wednesday 10.15-12.00 and Friday 12.15-14.00)

1. Enter - participate in interactive lecture IL (with lecturer) or supervision of CompEx1 (TAs). If supervision only - allocated table - or participate in IL and take breaks to ask for supervision of CompEx1.
2. If IL - answer: name+study programme+year+"I found CompEx1 ok or difficult" Then groups are formed (by lecuturer) as homogeneous as possible (so, not random).
3. 10.15/12.15: Presentation round in groups: name+study programme+year+previous background in statistics+level of expertise in R+favorite hobby!
4. 10.20/12.20: Introduction to problems on cross-validation - work with problems 1-3 (4): Recommended exercises on cross-validation
5. 10.50/12.50: Summing up problems 1-2 (3) by lecturer.
6. 11.00/13.00: Break - with fruits.

7. Maybe new groups, maybe not (we evaluate if some of the groups do not work well). If changes, need new presentation round.
8. 11.15/13.15: Introduction to problems on bootstrapping - work with problems 1-3: Recommended exercises on bootstrapping
9. 11.40/13.40: Summing up problems 1-2 (3) by lecturer.
10. 11.45/13.45: Team Kahoot! (in ghostmode on Friday - to beat the Wednesday people).
11. 12.00/14.00: The end:-)

Further reading

- Videoes on YouTube by the authors of ISL, Chapter 5, and corresponding slides

R packages to install before knitting this R Markdown file

```
# packages to install before knitting this R Markdown file
# to knit the Rmd
install.packages("knitr")
install.packages("rmarkdown")
# cool layout for the Rmd
install.packages("prettydoc") # alternative to github
#plotting
install.packages("ggplot2") # cool plotting
install.packages("ggpubr") # for many ggplots
#datasets
install.packages("ElemStatLearn")
install.packages("ISLR")
# cross-validation and bootstrapping
install.packages("boot")
```