Tampere University

Computational Diagnostics of Data

Data.ML-390

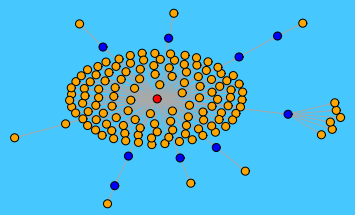
Week 4

Frank Emmert-Streib

Predictive Society and Data Analytics Lab

Tampere University, Finland

Contact: frank.emmert-streib@tuni.fi



Overall Schedule of Lectures

Effectively, 6 weeks!

1. Introduction to Data Science, general error measures and introduction to R
2. Statistical hypothesis testing & GO
3. High-throughput data from oligo arrays & classification methods
4. Resampling methods & Simulations
5. Linear regression analysis models
6. Survival analysis

Content of week 4

- Motivation
- Experimental design & reproducible research
  - General statistical models
  - Assessment of error measures
- Resampling methods
  - Cross validation (CV)
    - Hold-out set approach
    - Leave-one-out CV (aka Jackknife)
    - k-fold CV
    - Random resampling
  - Bootstrap
- Generalization
- Sampling from a distribution
- Standard error

Purpose of Resampling

Resampling methods mean to generate data

Content of week 4

- Motivation
- Experimental design & reproducible research
  - General statistical models
  - Assessment of error measures
- Resampling methods
  - Cross validation (CV)
    - Hold-out set approach
    - Leave-one-out CV (aka Jackknife)
    - k-fold CV
    - Random resampling
  - Bootstrap
- Generalization
- Sampling from a distribution
- Standard error

Theory

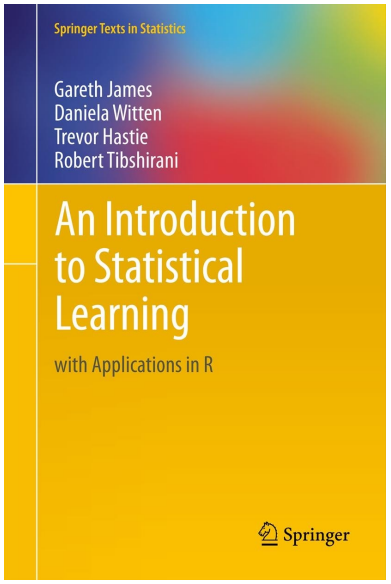
Part 1

Theory

Part 2

Basic idea

- In this lecture series we combine elements from ‘computational biology’ and ‘statistical learning’ to form ‘computational diagnostics’.

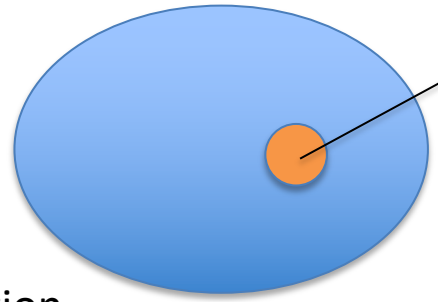


Recommended additional reading for this week

1. Motivation

Dealing with data = dealing with uncertainty

Estimating a model (including parameters):



Data (D; sample size n): cholesterol level of patients

apply a method (M)

Form a predictive model:

Population

$$y = M(x; \beta)$$

random variable (RV)!!!

data

parameter

Method M:

- Classification
- Linear regression

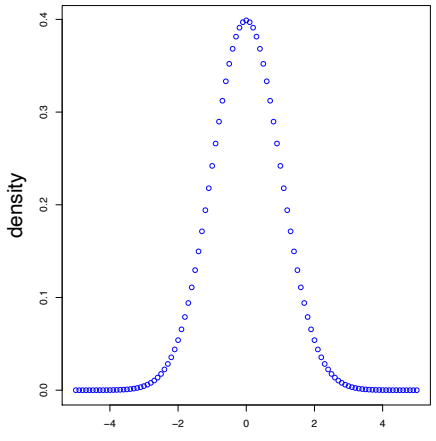
Dealing with uncertainty

Simple example: estimate the sample mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$x_i \sim f(x;\alpha)$$

$$D = \{x_1, \ldots, x_n\}$$



Sample two data sets (theoretical)

Perform two experiments (practical)

D_1 and D_2

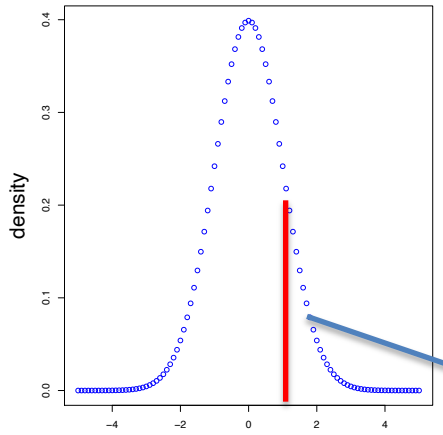
$$\bar{x}_{D_1} \neq \bar{x}_{D_2}$$

Recall: inferred vs derived

Dealing with uncertainty

- Typical to every data analysis problem.

- Analyzing one data set (no matter how large – its always finite) does not tell us everything we need to know.



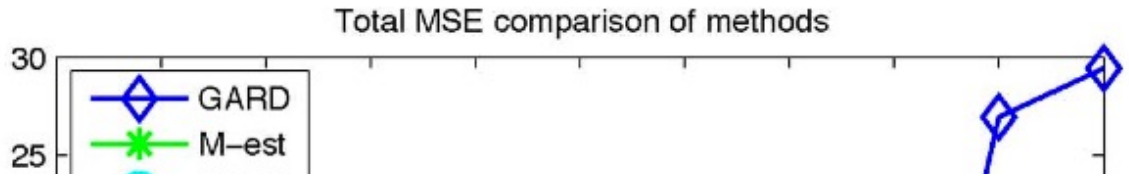
$$y = M(x;\beta)$$

Mean value
Support vector machine
Logistic regression
Linear regression

from one data set

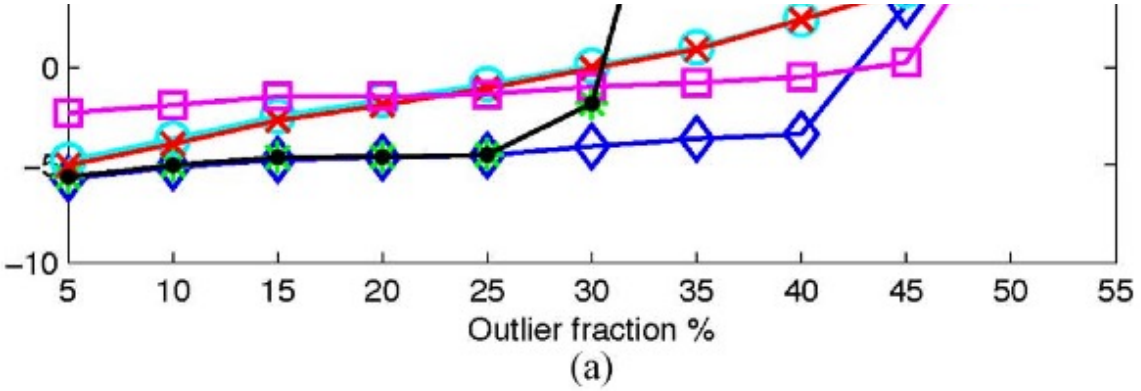
Example

MSE: mean squared error



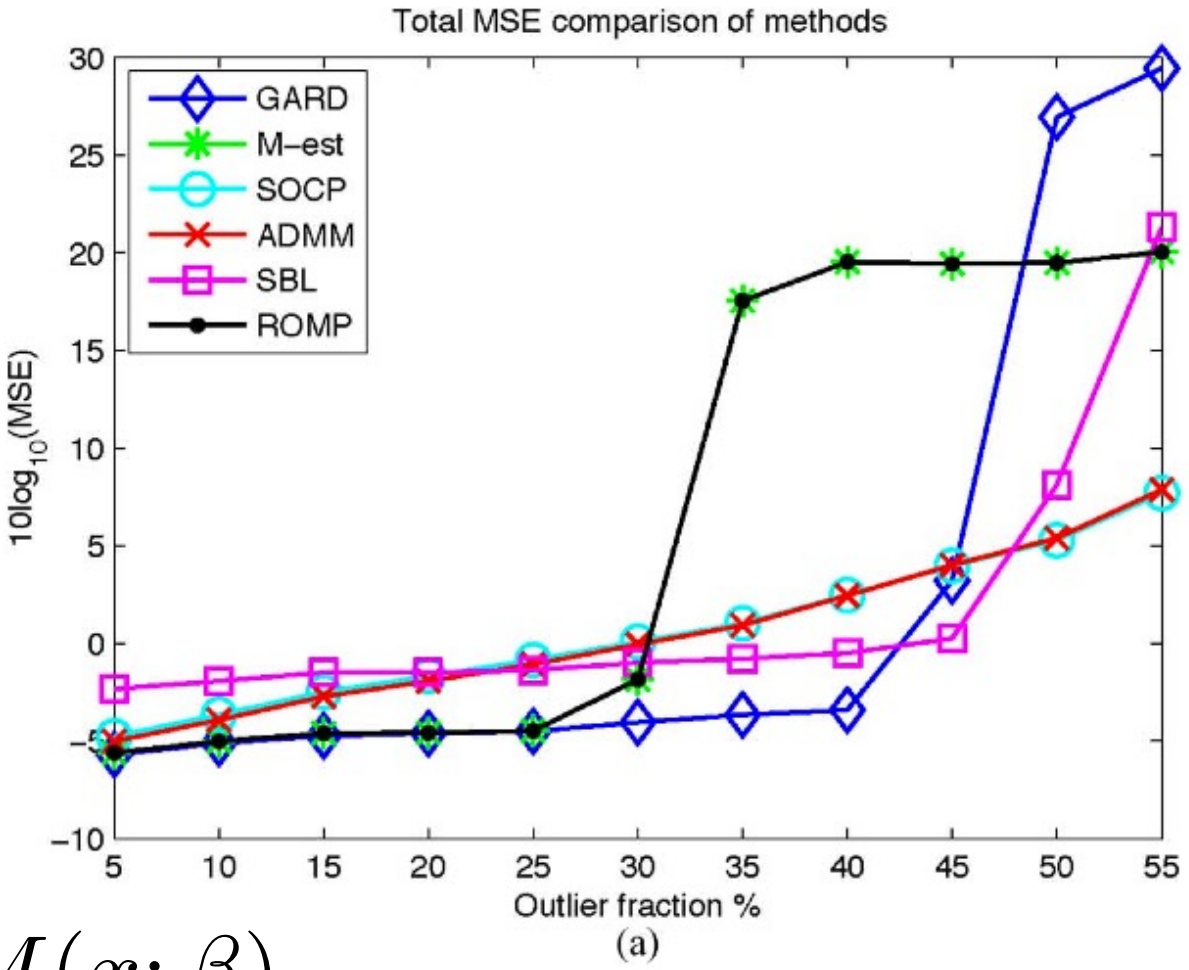
Robust Linear Regression Analysis— A Greedy Approach

George Papageorgiou, *Student Member, IEEE*, Pantelis Bouboulis, *Member, IEEE*, and Sergios Theodoridis, *Fellow, IEEE*



Example

MSE: mean squared error

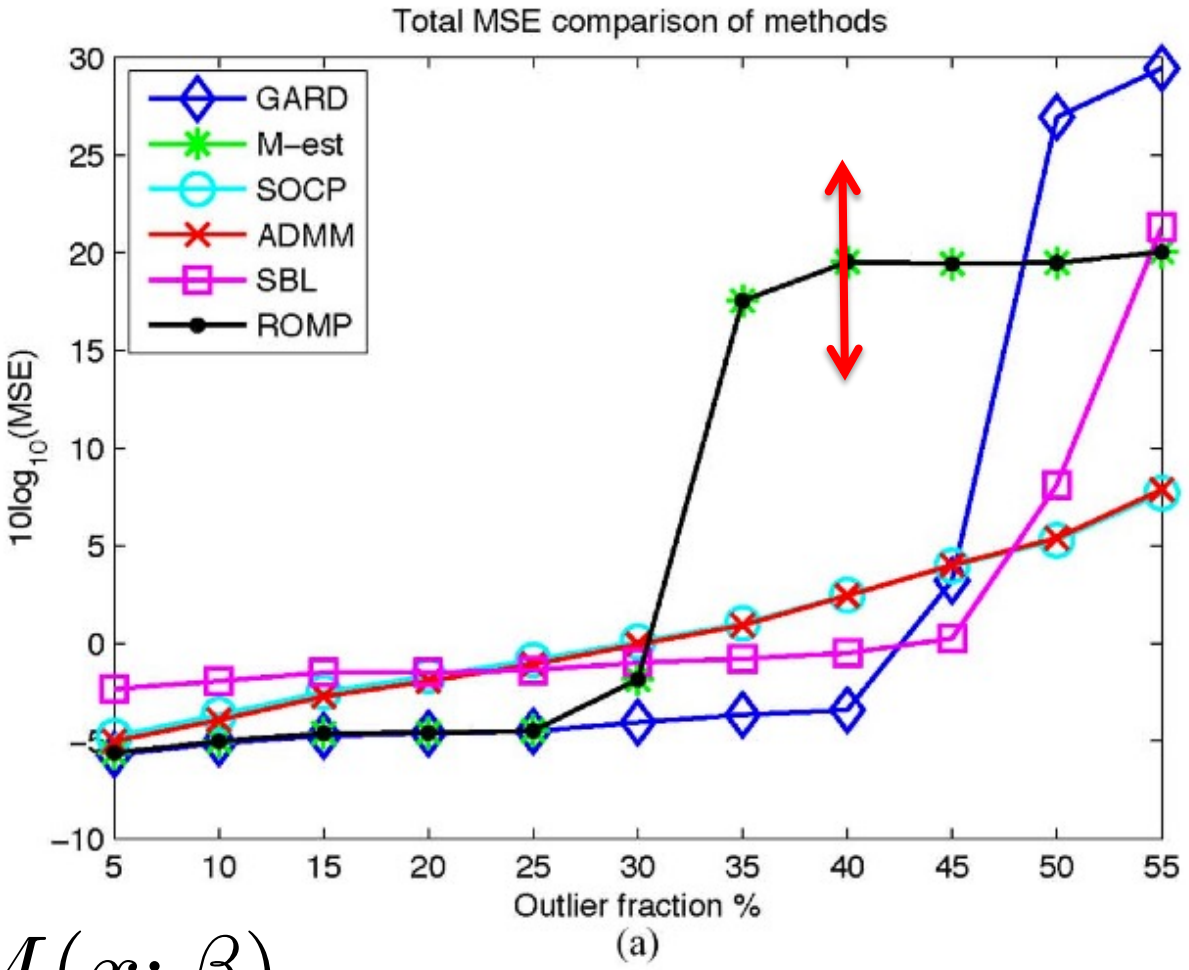


$$y = M(x; \beta)$$

The fact that y is a RV is not appreciated.

Example

MSE: mean squared error



Variability!

$$y = M(x; \beta)$$

Remark

- The resampling methods we discuss this week ‘use one data set’ to approximate characteristics of ‘many data sets’.

- This gives us information about the uncertainty in the data.

Applications:

1. Model assessment (assessment of errors)
2. Model selection

2. Experimental design & reproducible research

Experimental design

- General aspects for the planning of experiments, in order to be able to answer specific questions quantitatively.
- One important parameter of any experiment is sample size. (our focus)
- Other factors are: physiological conditions, perturbations of genes etc.
- Resampling methods can be used to assess the experimental design. For instance, by assessing estimates of errors. (our focus)

General setting

Generate data by an experiment:

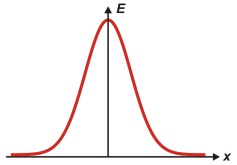
$$D = \{x_1, \ldots, x_n\}$$

$$x_i \sim f(x; \alpha)$$

Use the data to make a prediction:

parameter

$$M(x; \beta) \sim y$$



Prediction

random variable (RV)

M: Learning machine (statistical model, method)

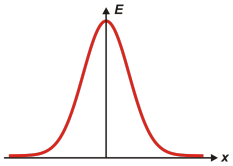
Classification, Regression, Hypothesis testing etc

General setting

Generate data by an experiment:

$$D = \{x_1, \ldots, x_n\}$$

Assessment of the predictions:



$$m(y) \sim E$$

Error measure (RV)

E: Error measure

Precision, sensitivity, AURUC etc

In Week 1

2.1. General statistical models

Nature of the problem

Using a statistical model is a two step process:

1. Training step: Learning the parameters of the model (estimating the values of its parameters).

$$M(x;\beta) \sim y$$

1. Testing step: Assessing (or testing) the predictive performance of the model.

   Called Model assessment.

Nature of the problem

Using a statistical model requires data:

1. In order to learn the parameters of the model we need data.

$$M(x;\beta) \sim y$$

Intuitively clear

2. In order to assess the model we need to know the truth.

Nature of the problem

Using a statistical model requires data:

1. In order to learn the parameters of the model we need data.

$$M(x;\beta) \sim y$$

Less clear

2. In order to assess the model we need data.

Consider two problems

Classification problem:

- Data points present in pairs (x_i, y_i = class label i)
- If using x_i to make a prediction about the class label, y_i provides information about the true class label.

Regression problem:

$$y \sim \sum_{k=1}^{m-1} \beta_k x^k$$

- Data points present in m-tuple $(x_i^1, x_i^2, \ldots, x_i^{m-1}, y_i)$
- If using $x_i^1, x_i^2, \ldots, x_i^{m-1}$ to make a prediction, y_i provides information about the true value.

Nature of the problem

Using a statistical model requires two types of data:

1. Training step: In order to learn the parameters of the model we need a **training data set**.

$$M(x;\beta) \sim y$$

2. Testing step: In order to assess the model we need a **test data set**.

Nature of the problem

When we perform a data analysis we only have one data set D available.

We do not have two data sets, one is called the training data set D_1 and the other the test data set D_2.

Q: How do we split D? ⟹ Resampling methods

Summary

- Using a statistical model is a two step process.
- We always require a training data set and a test data set.
- In order to obtain these we need to split our data set D into two parts.
- How to split, is answered by different resampling method.

Question

What happens if we do not split the data?

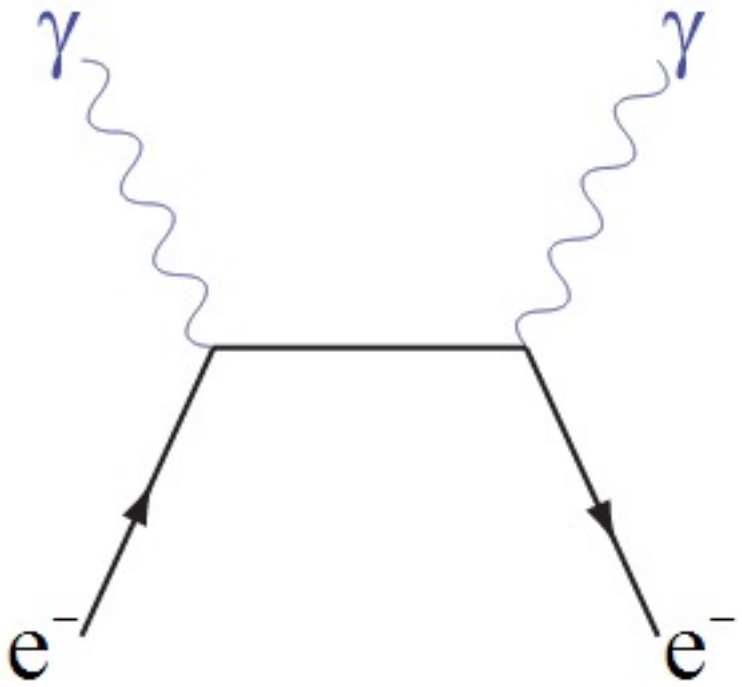
Use the same data for training and testing.

Extreme example

Training data set: (consists of one data point)

Q: What is the Feynman diagram of a Compton scattering?

Answer

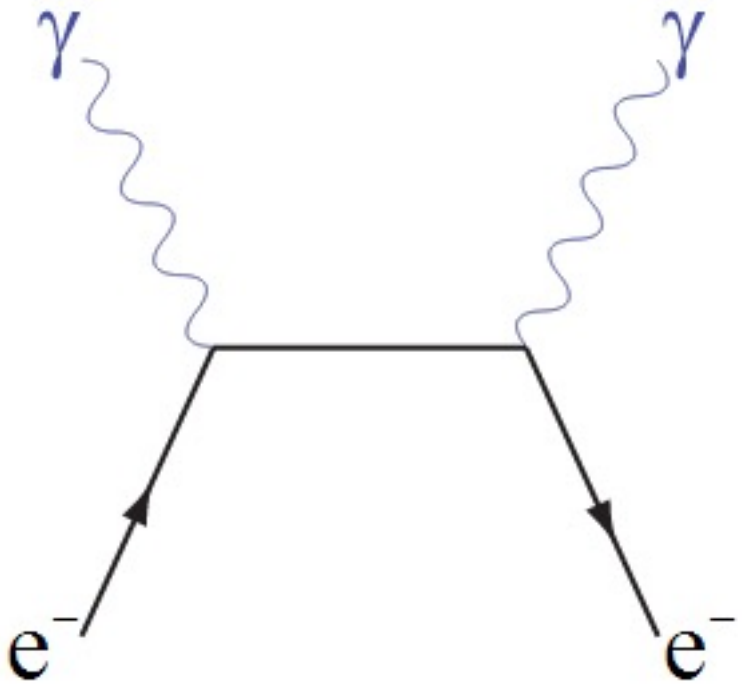


Extreme example

Test data set:

Q: What is the Feynman diagram of a Compton scattering?

Answer



Problem:

Generalization!

Assessment: Whatever error measure E one is using you would perform well.

Remarks

Depending on how we split the data, this effects the assessment of the model.

Remarks

When splitting a data set into a training data set D_1 and a test data set D_2, and using D_2 to assess the model, the resulting error E is a **sample error** not a population error.

Population error: Assessing the model by

- 'asking all possible questions'.
- using a data set with infinite many samples.

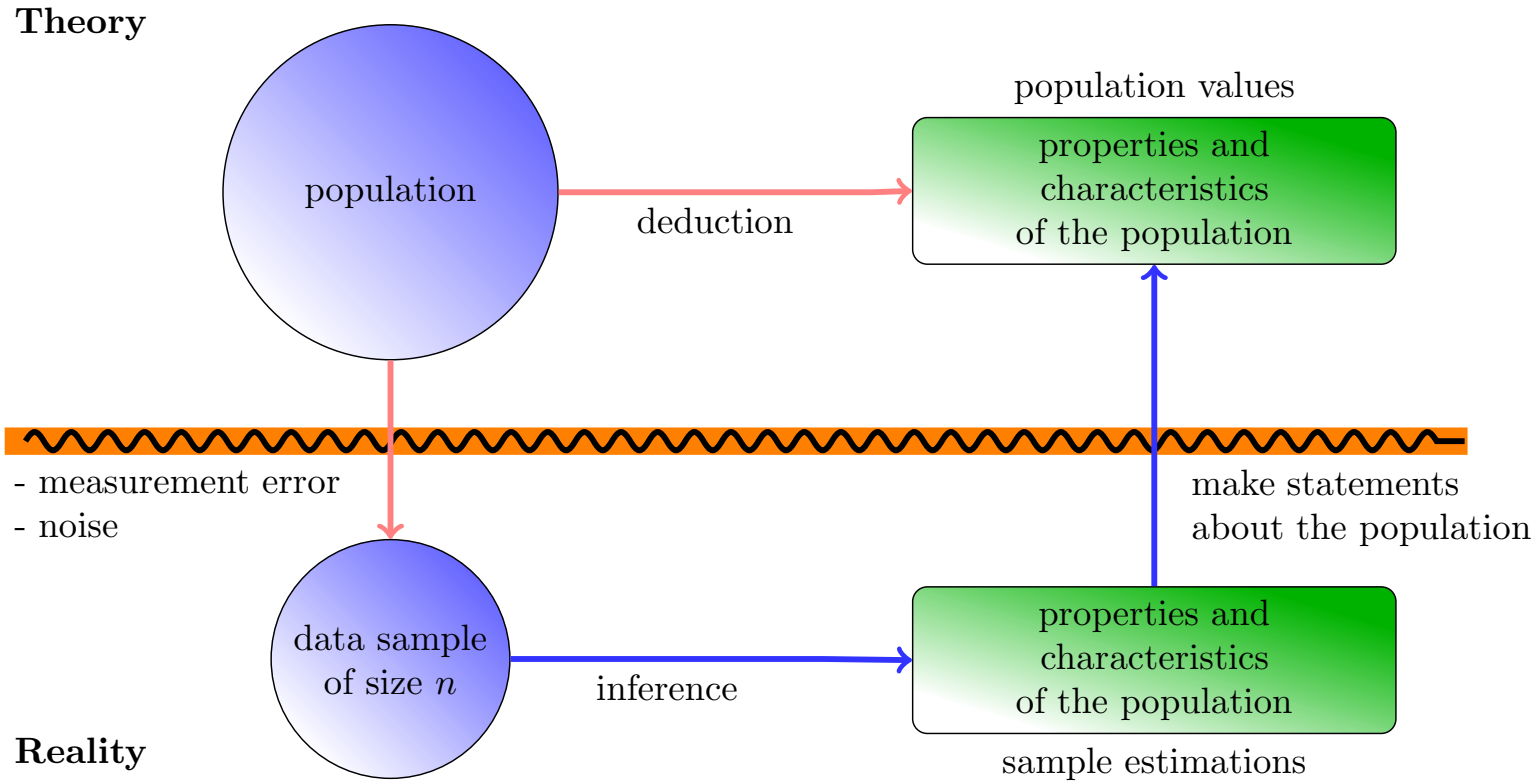
Most comprehensive assessment.

Remarks

Due to the difference between the sample error and the population error different ways of 'how to split' data exist in the form of resampling methods, because they have different estimation characteristics.

Different approaches have different advantages/disadvantages in certain situations.

Fundamental problem



Convention

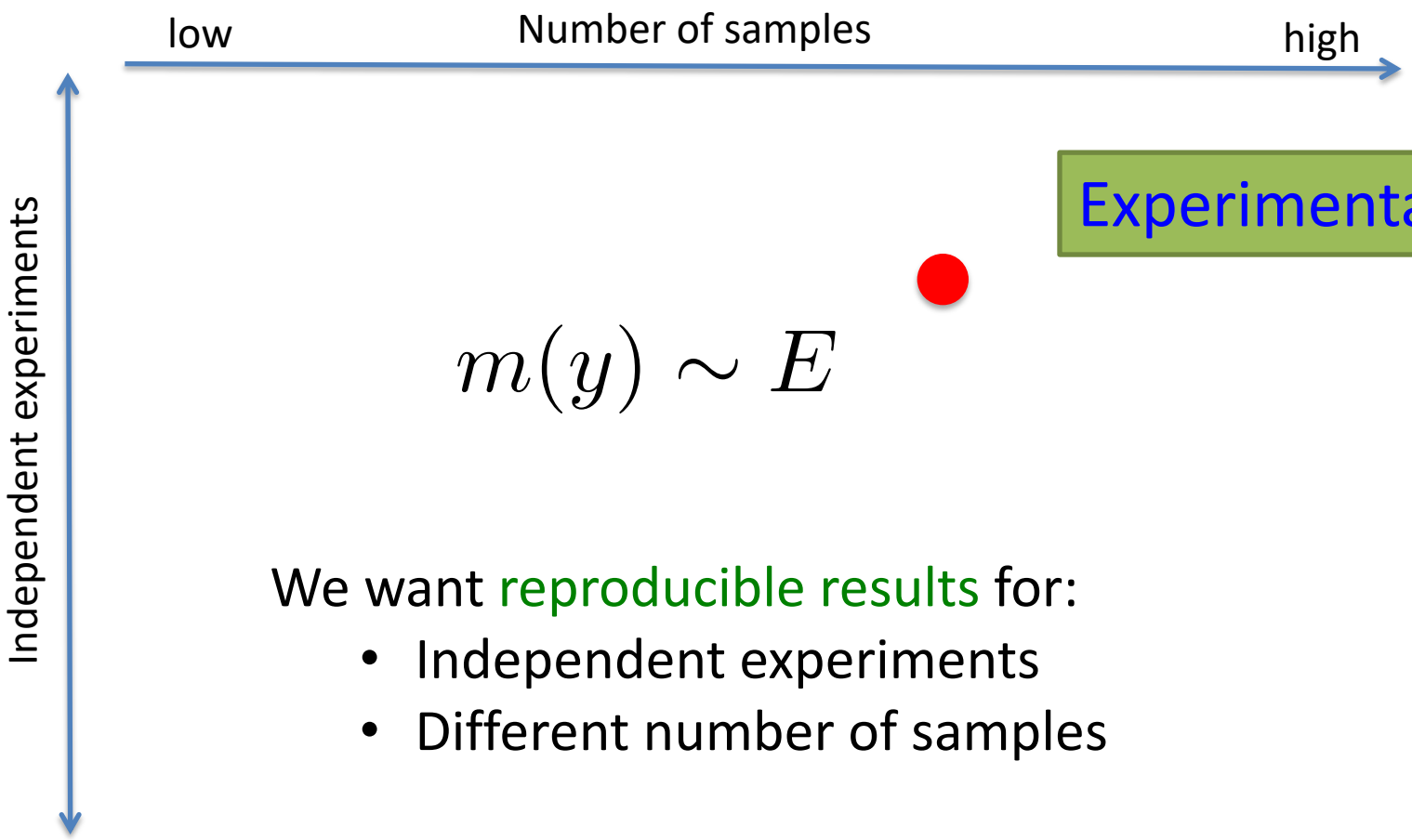
Important: When error measures (sensitivity, accuracy etc) are provided (publications) typically the word ‘sample’ is omitted (e.g. sample sensitivity).

Still, error measures are sample measures.

Now, we go a step back to understand the larger context (experimental design).

2.2. Assessment of error measures

Assessing error measures



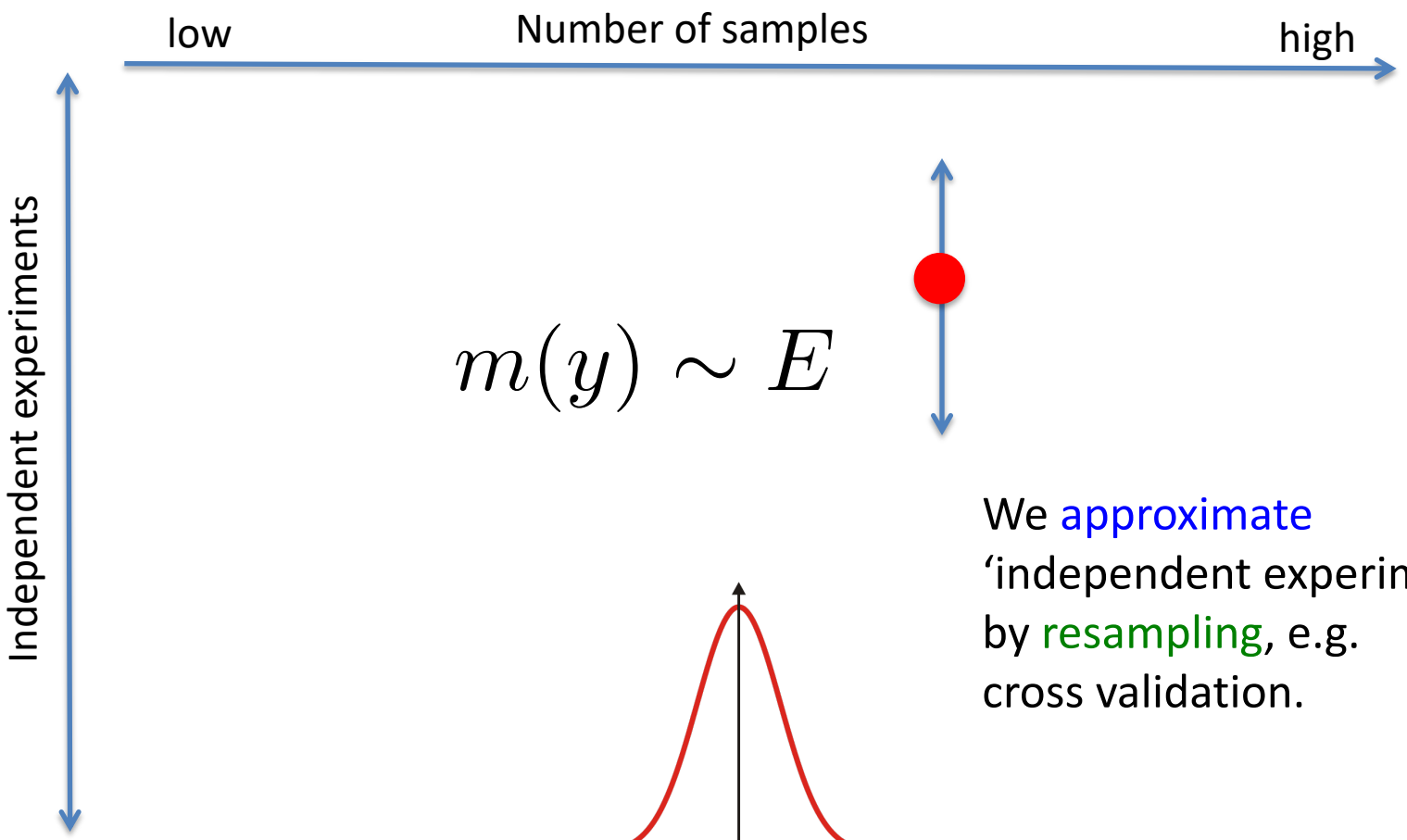
Experimental space

$$m(y) \sim E$$

We want reproducible results for:

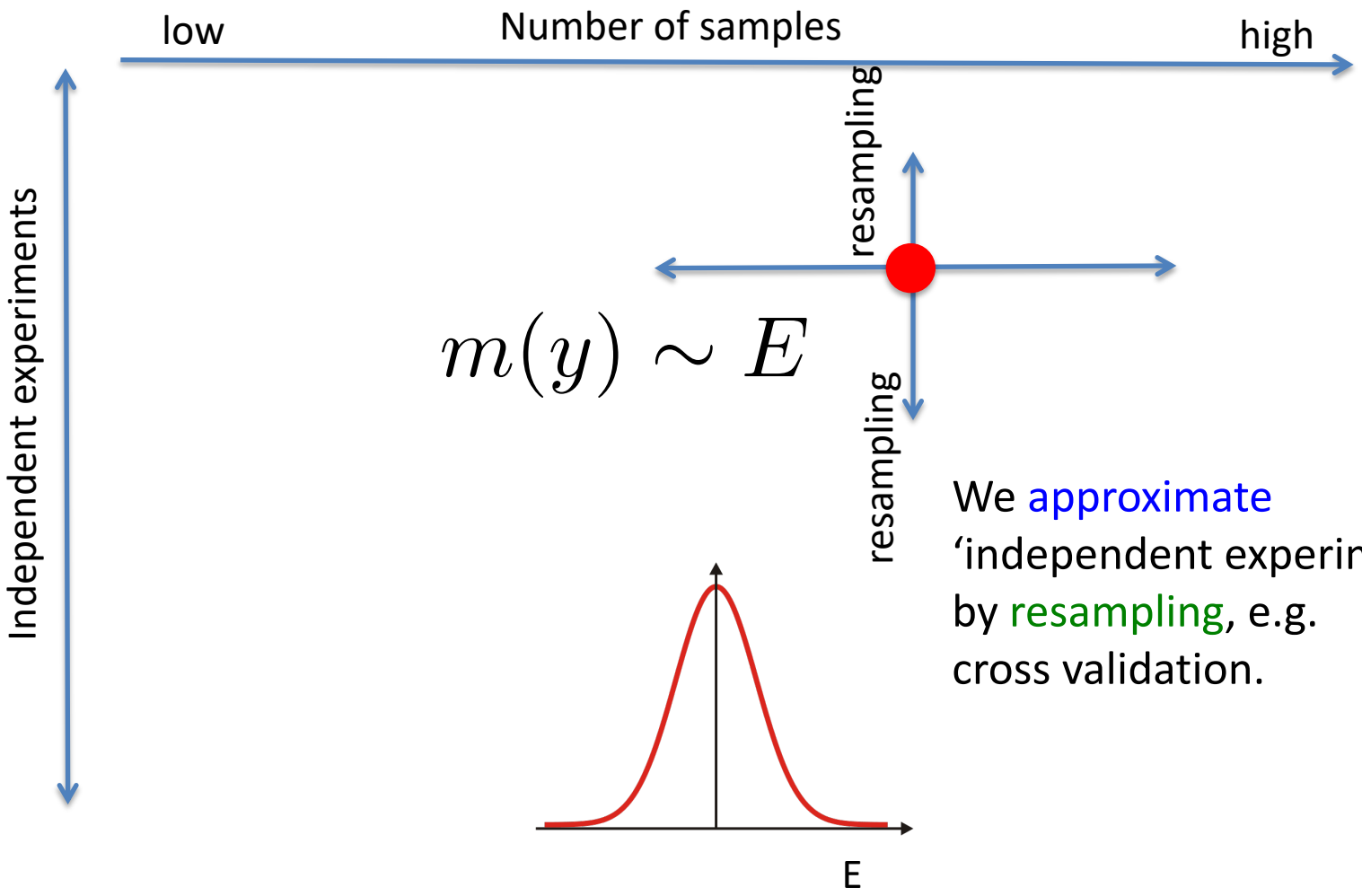
- Independent experiments
- Different number of samples

Assessing error measures



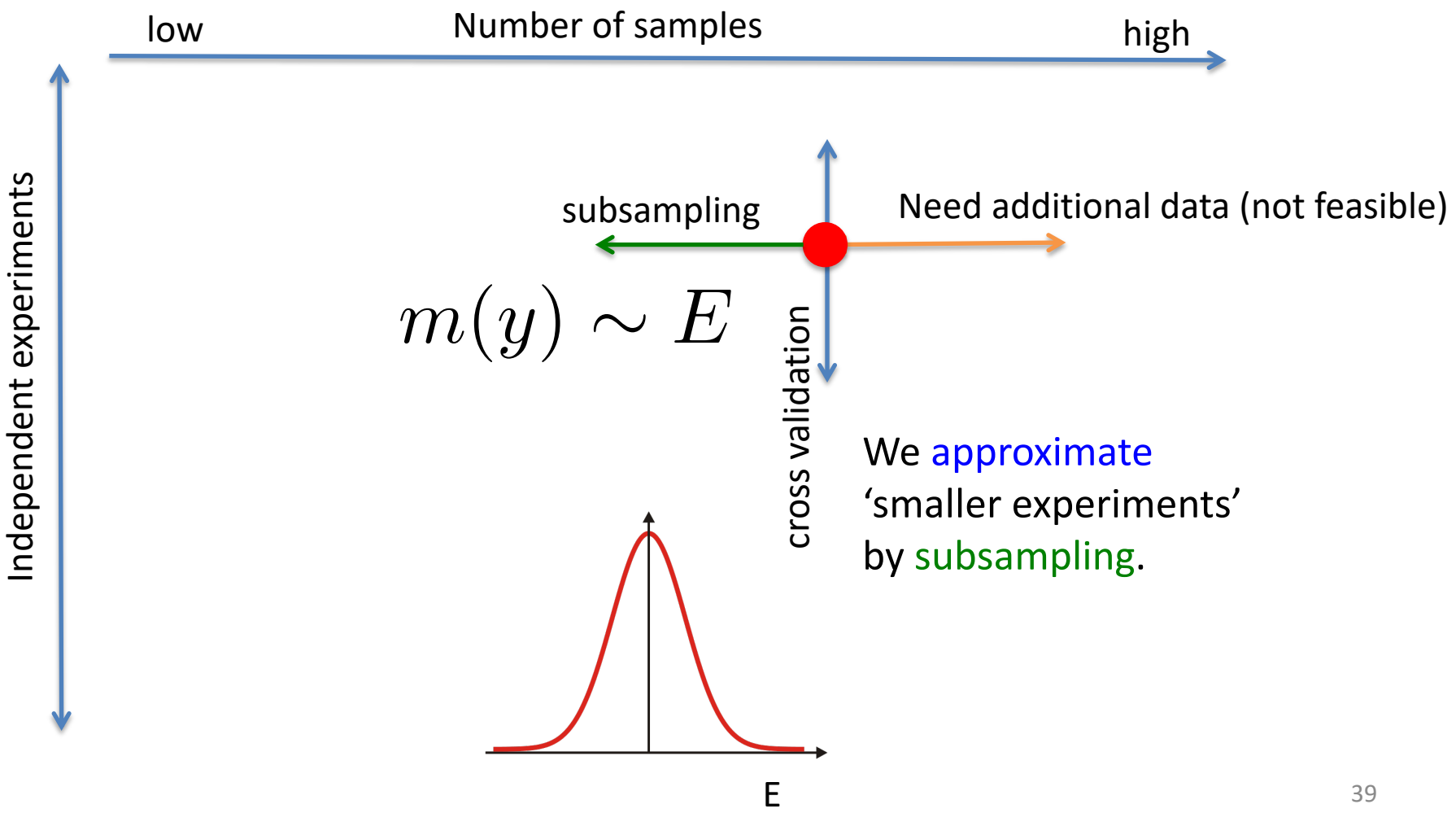
We approximate 'independent experiments' by resampling, e.g. cross validation.

Assessing error measures

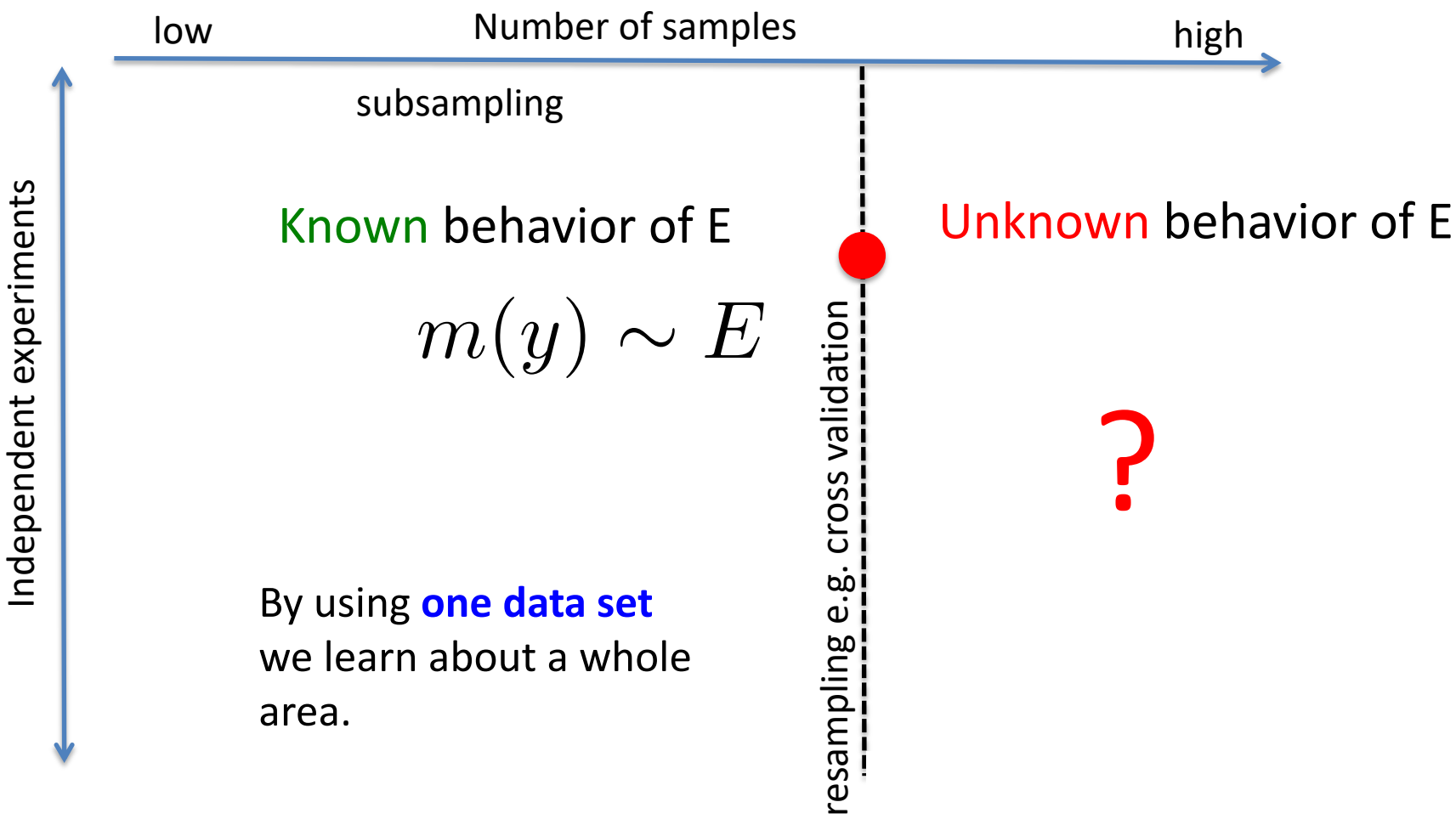


We approximate 'independent experiments' by resampling, e.g. cross validation.

Assessing error measures



Assessing error measures



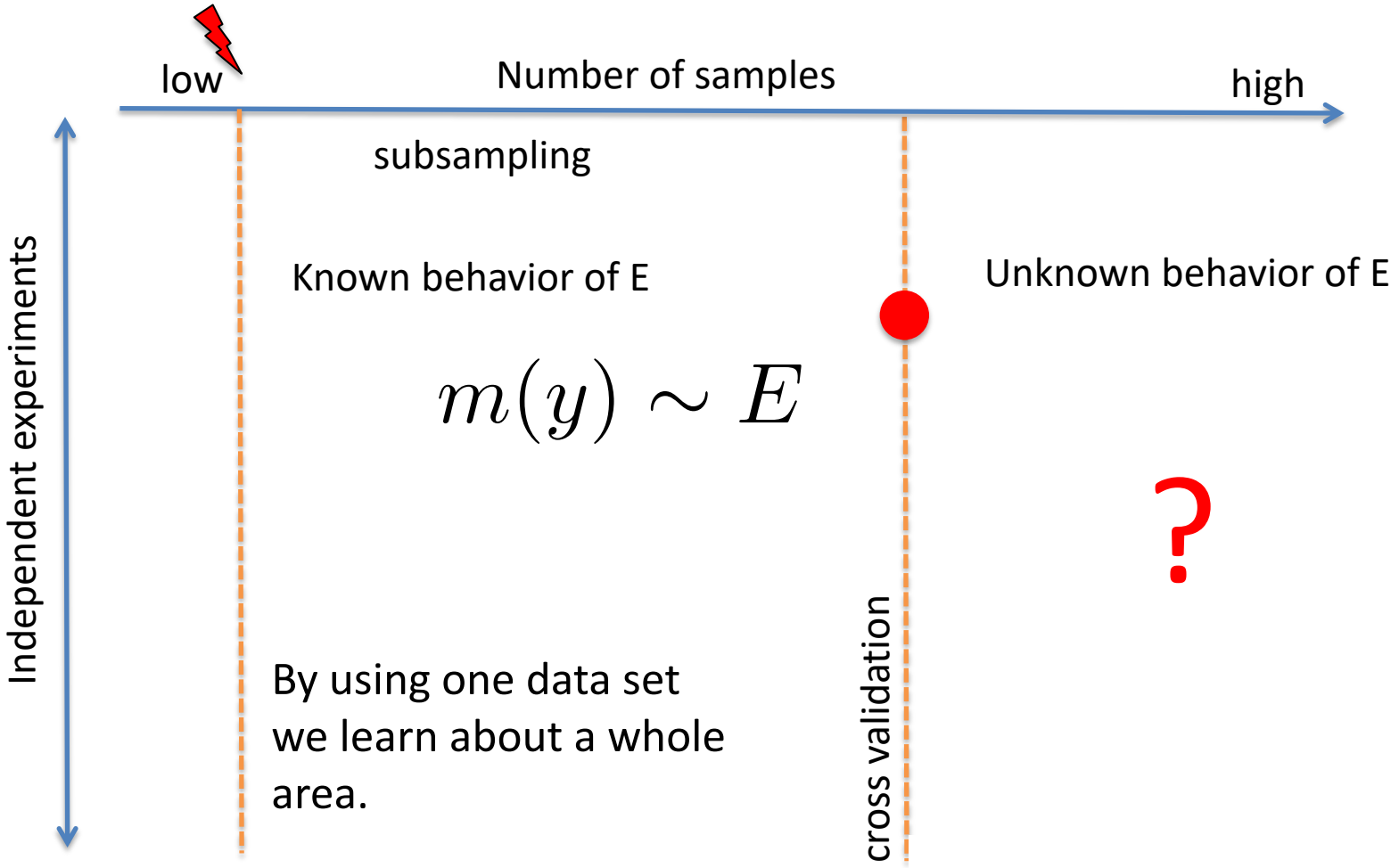
Modern Data analysis

A sensible data analysis of a sufficiently **large** dataset 'repeats' a similar analysis thousands of times.

In the era of Big Data, we have large sample sizes available.

Sample size matters

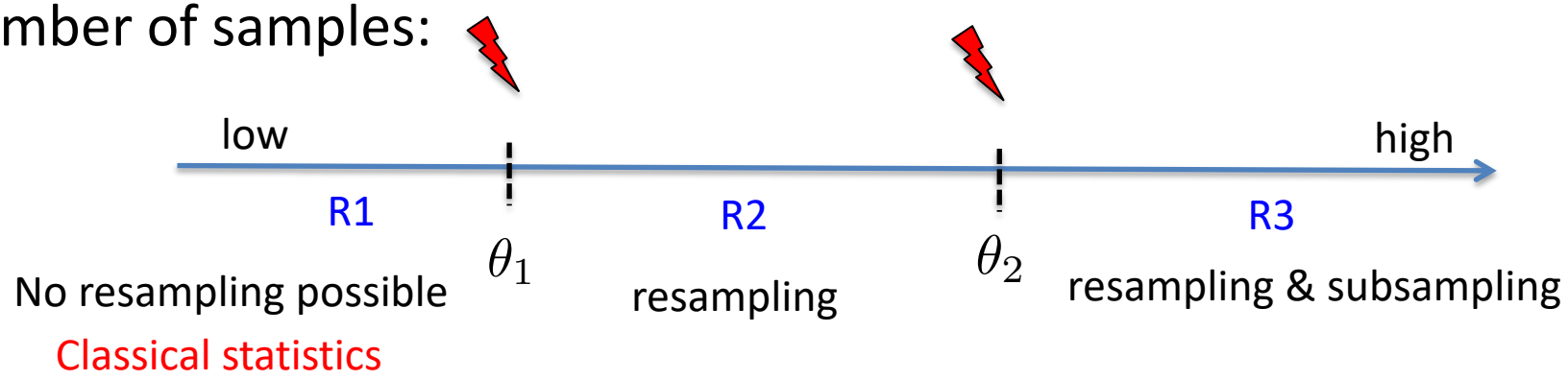
There is a sample size n_0 below which subsampling is not possible! (Experimental design)



Sample size matters

There are three regions (R1-R3) one needs to distinguish from each other.

Number of samples:

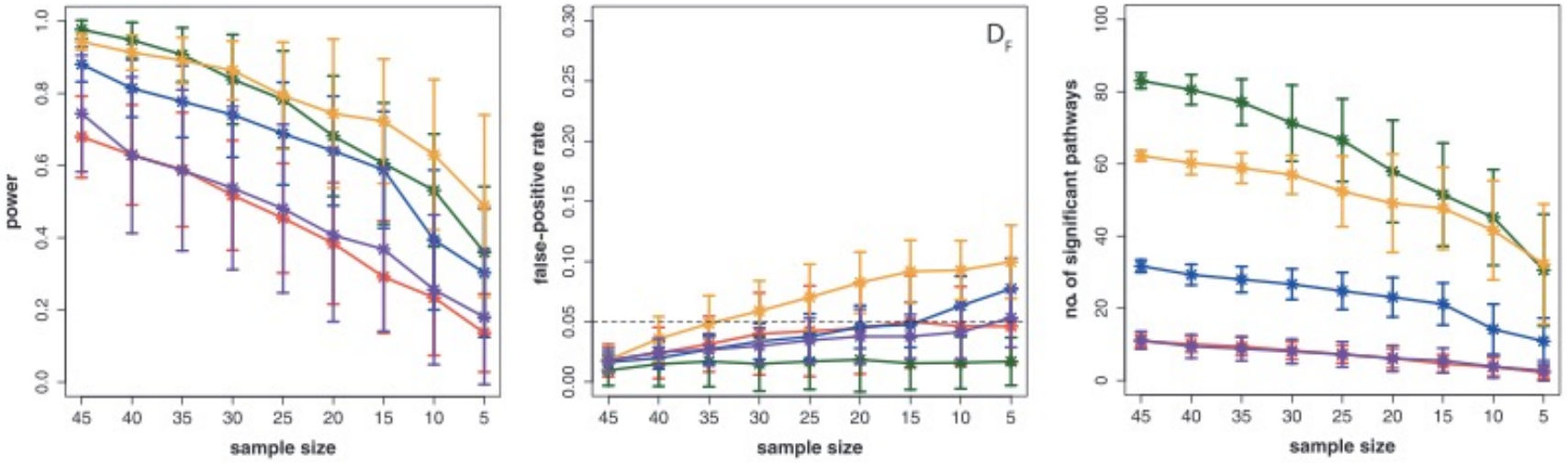


Analysis that can be performed given a certain data set D:

The value of θ_1, θ_2 depends on a given data set D and the method.

1. Example: Resampling + Subsampling

Three different error measures E – microarray data



Published online 6 February 2013

Nucleic Acids Research, 2013, Vol. 41, No. 7 ***e82***
doi:10.1093/nar/gkt054

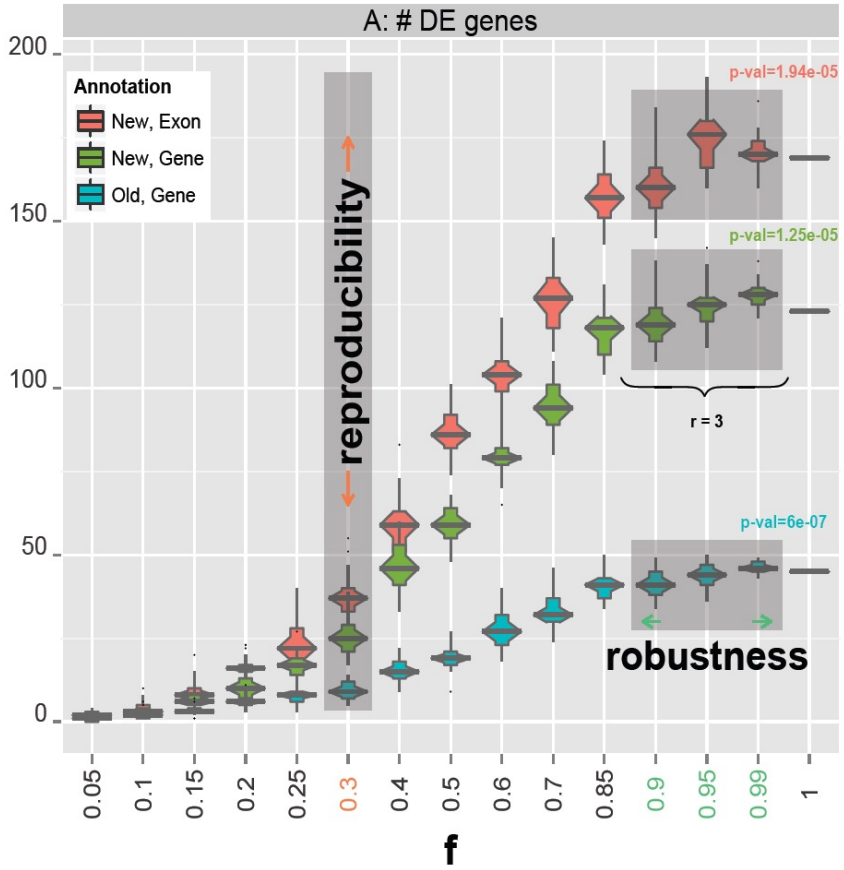
Ensuring the statistical soundness of competitive gene set approaches: gene filtering and genome-scale coverage are essential

Shailesh Tripathi[1], Galina V. Glazko[2] and Frank Emmert-Streib[1,*]

Total sample size: ~ 100 patients (50 per condition)

2. Example: Resampling + Subsampling

RNA-seq (NGS) data



BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–2

samExploreR: Exploring reproducibility and robustness of RNA-seq results based on SAM files

Alexey Stupnikov [1], Shailesh Tripathi [1,2], Ricardo de Matos Simoes [1], Darragh McArt [3], Manuel Salto-Tellez [3], Galina Glazko [4], Frank Emmert-Streib [1,5,6,*]

Total sample size:
~ 40 million (short reads)
sequencing depth

Data analysis

You need redundancy in your dataset in order to be able to 'leave some data out'.

- Cross validation
- Subsampling

That means, not all data (samples) are used to train/learn the model parameters.

3. Resampling methods

Purpose of resampling methods

Model assessment: The purpose of resampling methods is to assess results (e.g. error measures) that are obtained from the application of a method to a data set, based on the data set itself.

Whenever possible, resampling methods should be applied to every data analysis problem.

For reasons of simplicity, we will not do it during the course.

Remark

Resampling methods provide a practical approach to this problem:

- The resampling methods we discuss this week ‘use one data set’ to approximate characteristics of ‘many data sets’.

Before we move on, some exercises

1. Sample mean:

```
 8  N <- 3
 9  x <- rnorm(N, mean=0, sd=1)
10
```

Experiment 1

```
> x
[1] -0.6742971  0.1613641 -3.9820756
> mean(x)
[1] -1.498336
> sd(x)
[1] 2.191188
```

Before we move on, some exercises

1. Sample mean:

```
 8  N <- 3
 9  x <- rnorm(N, mean=0, sd=1)
10
```

Experiment 1

```
> x
[1] -0.6742971  0.1613641 -3.9820756
> mean(x)
[1] -1.498336
> sd(x)
[1] 2.191188
```

Lets repeat the same: Experiment 2

```
> x
[1] 0.4407241 1.4490309 1.2835662
> mean(x)
[1] 1.057774
> sd(x)
[1] 0.540747
```

We notice an uncertainty in the mean and the standard deviation.

Before we move on, some exercises

2. Sample mean (change sample size):

```
 8  N <- 1000
 9  x <- rnorm(N, mean=0, sd=1)
10
```

Experiment 1

```
> mean(x)
[1] -0.03534264
> sd(x)
[1] 0.9843959
>
```

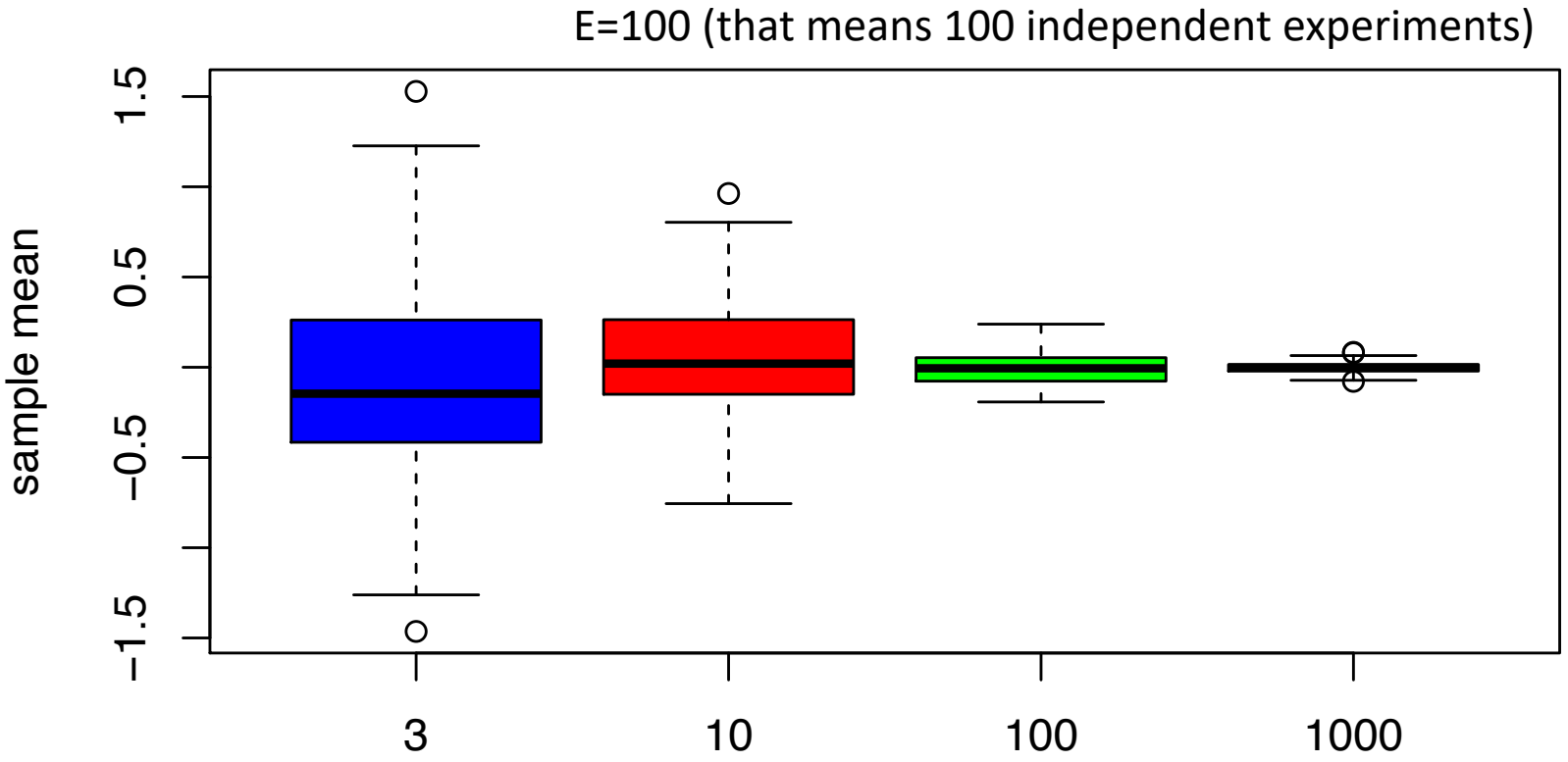
Lets repeat the same: Experiment 2

```
> mean(x)
[1] 0.01755184
> sd(x)
[1] 1.036585
>
```

We notice a **reduced** uncertainty in the mean and the standard deviation.

Understanding 'uncertainty'

What is a boxplot?



Here alternative meaning of y-axis: sample size

zero

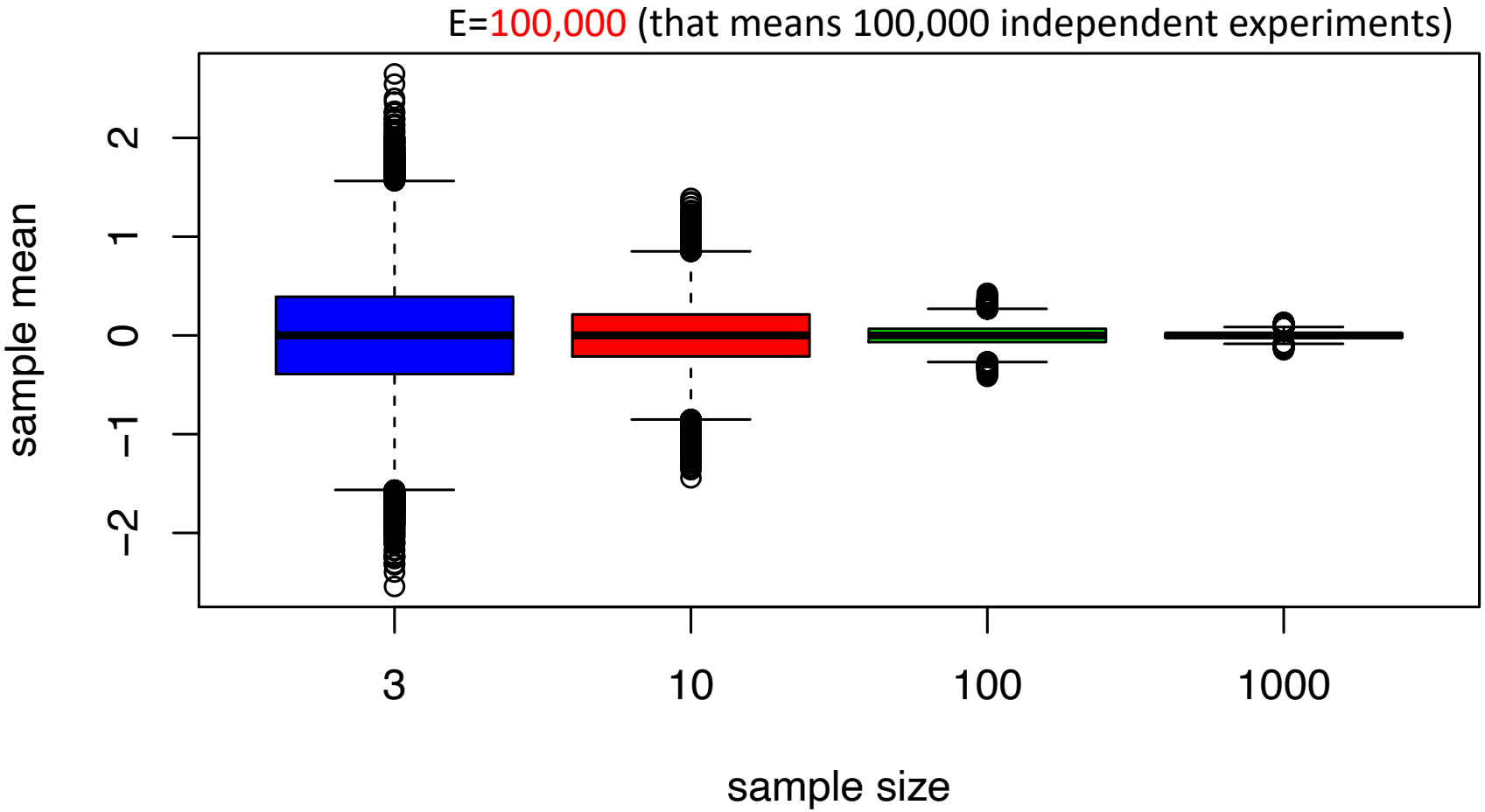
$$\text{distance: sample mean} = \text{sample mean} - \mu$$

R script

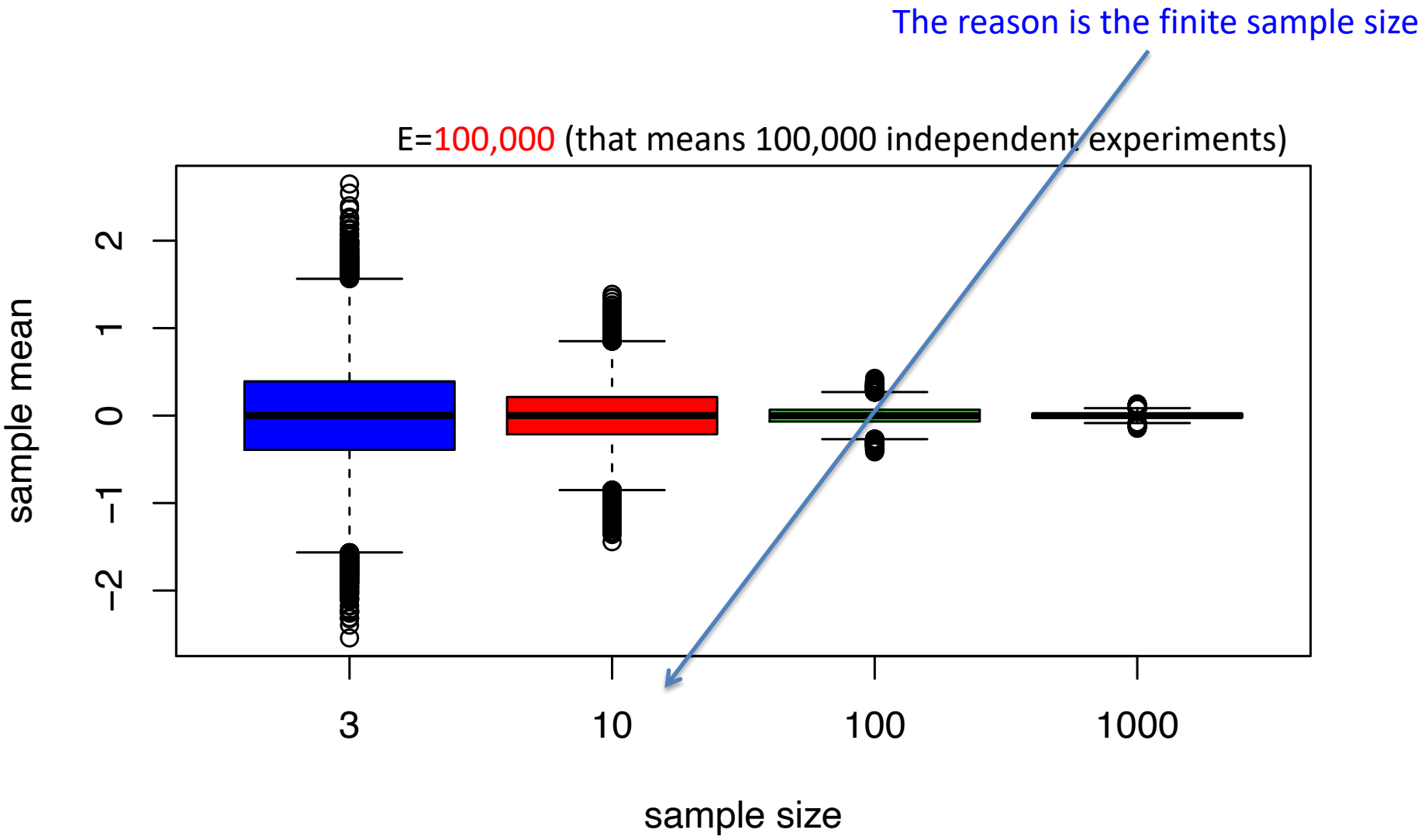
```
 7  E <- 100
 8  S <- c(3, 10, 100, 1000)
 9  xres <- matrix(0, nrow=4, ncol=E)
10
11  for(i in 1:4){
12      for(j in 1:E){
13          N <- S[i]
14          x <- rnorm(N, mean=0, sd=1)
15          xres[i,j] <- mean(x)
16      }
17  }
18
19  boxplot(t(xres), col=c("blue", "red", "green", "orange"),
20      names=S, xlab="sample size", ylab="sample mean")
21
22
23  dev.copy2pdf(file="boxS1.pdf", out.type = "pdf")
24
```

Understanding ‘uncertainty’

Why does it look the same as for E=100?



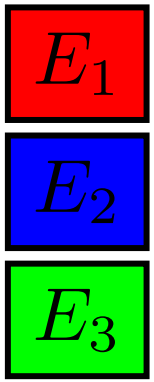
Understanding ‘uncertainty’



Understanding ‘resampling’

C

3 independent experiments



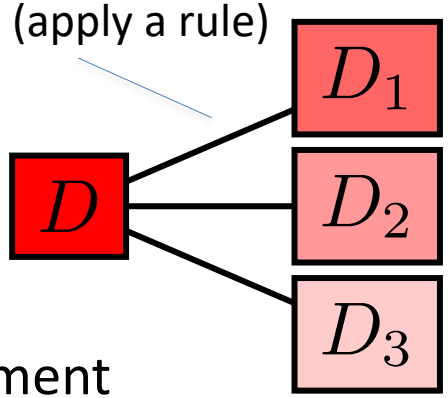
independent data

B

resampling (apply a rule)

A

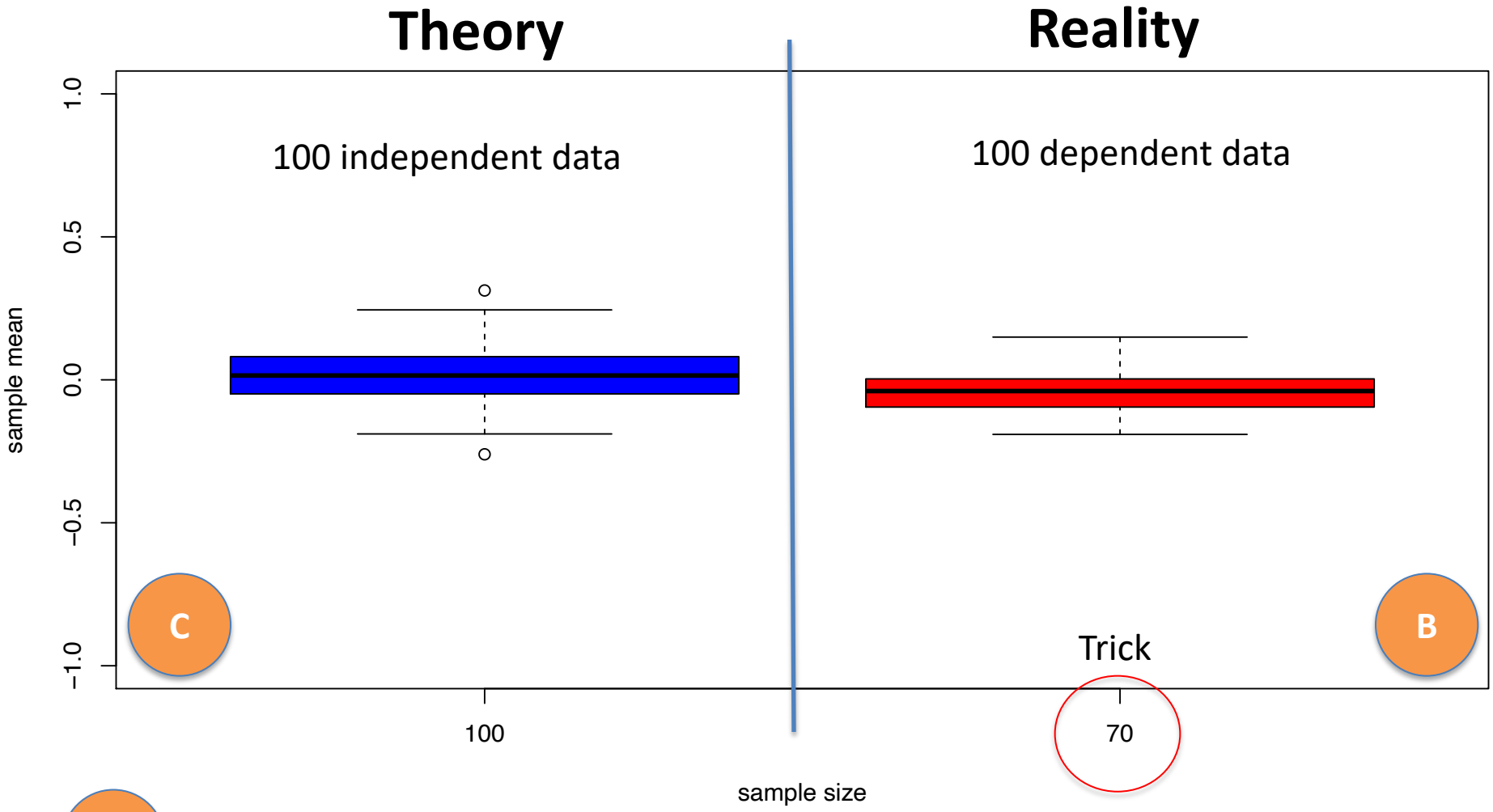
source data



dependent data

1 (independent) experiment

Understanding ‘resampling’



A

sample mean $= -0.053$ of the source data (n=100)

Reality

R script

```r
N <- 100
E <- 100
xres <- matrix(0, nrow=2, ncol=E)

for(j in 1:E){
    x <- rnorm(N, mean=0, sd=1)
    xres[1,j] <- mean(x)
}


x <- rnorm(N, mean=0, sd=1)
for(j in 1:E){
    xb <- sample(x, 0.7*N, replace=FALSE)
    xres[2,j] <- mean(xb)
}

boxplot(t(xres), col=c("blue", "red"), names=c(100, 70),
    xlab="sample size", ylab="sample mean", ylim=c(-1, 1))


dev.copy2pdf(file="boxS3.pdf", out.type = "pdf")
```

Different resampling methods

The following approaches are all resampling methods:

- Cross validation (CV)
  1. Hold-out set approach (educational)
  2. Leave-one-out CV (aka Jackknife)
  3. k-fold CV
  4. Random resampling
- Bootstrap

Problem of data splitting

In order to assess a statistical model we need training data and test data.

Ideal requirements:

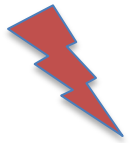
- Training data should be large as possible (the larger the better our estimates of the model's parameters)
- Test data should be as large as possible (the larger the better our estimates of the model's error)

Problem of data splitting

In order to assess a statistical model we need training data and test data.

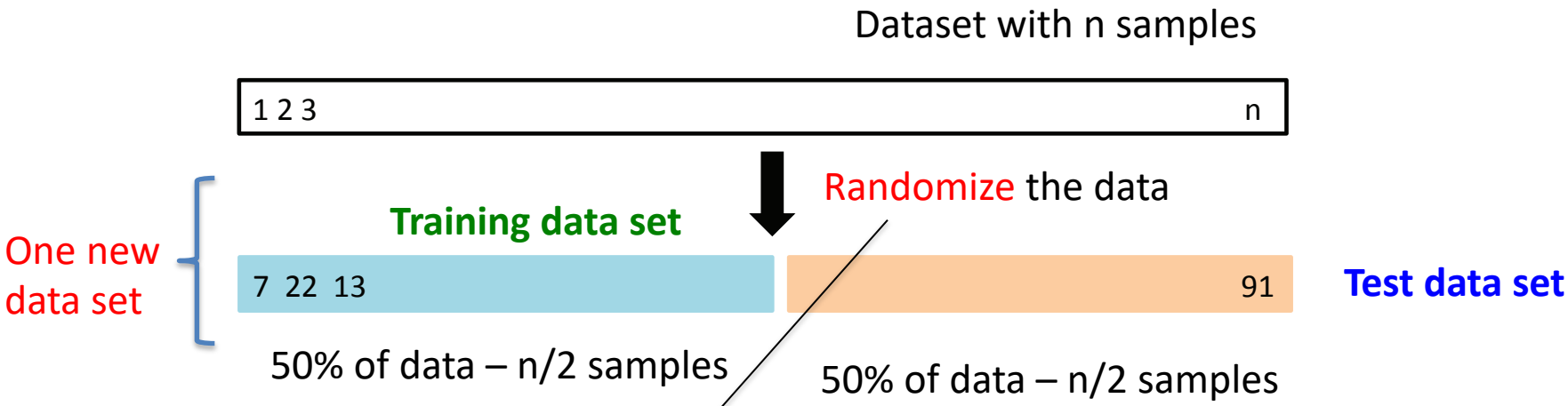
Ideal requirements:

- Training data should be large as possible (the larger the better our estimates of the model's parameters)
- Test data should be as large as possible (the larger the better our estimates of the model's error)



Compromise required

1. Hold-out set approach



How do we decide which samples are for the training set and the test set?

We assign the samples randomly, i.e.,

Prob(train) = ½ = Prob(test) for each sample

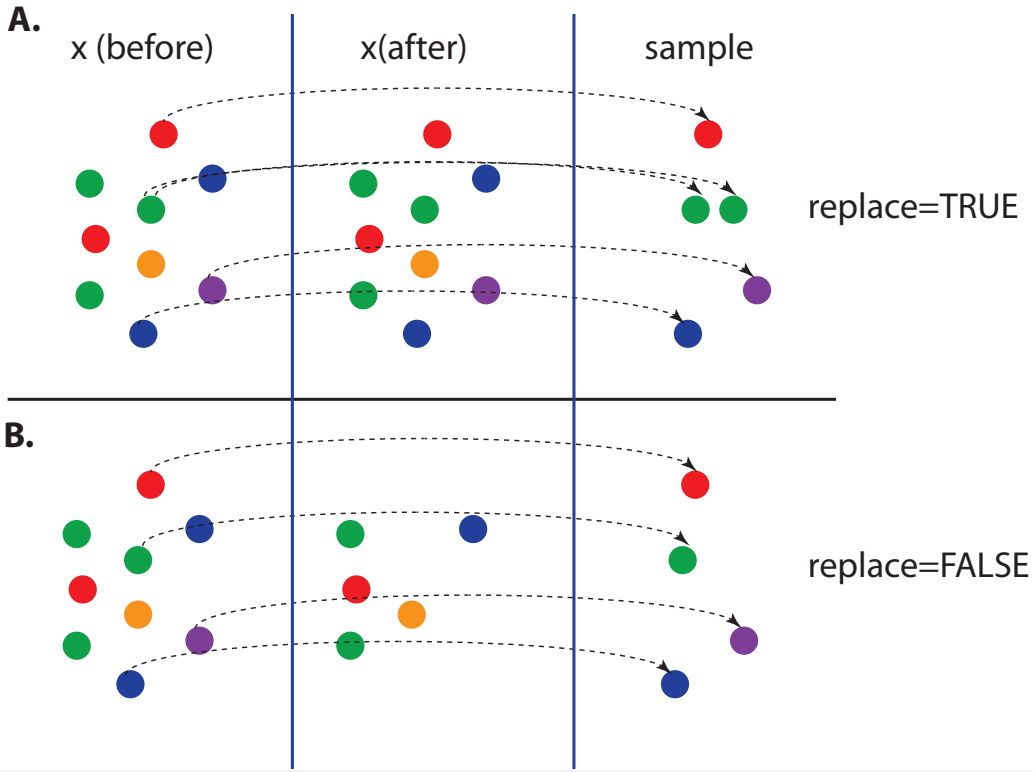
Hold-out set approach in R

```
n <- 100
m <- n/2
D <- 1:n  # the indices of our data points
D1 <- sample(D, m, replace=FALSE)
D2 <- setdiff(D, D1)
```

D1 and D2 contain the indices of the data points in the training and the test data set.

Sample function in R

sample:




```
> sample(1:5, 20, replace = TRUE)
 [1] 5 4 5 4 1 1 4 5 4 1 1 5 2 5 1 4 3 2 5 2
```

1. Hold-out set approach

The holdout method has two drawbacks

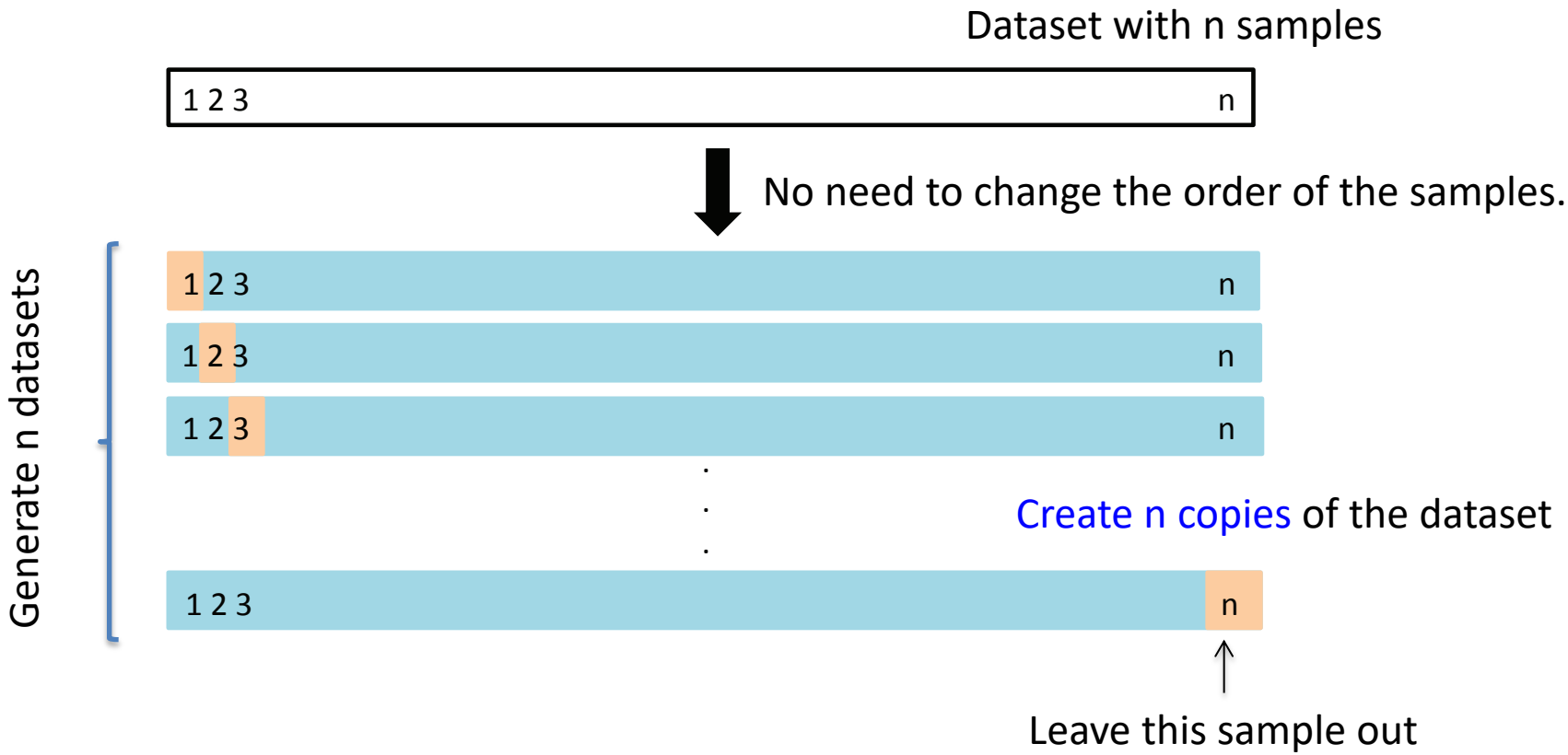
- For a sparse data set (small number of samples) we may not be able to afford the "luxury" of setting aside a large portion (50%) of the data for testing
- Since it is a single train-and-test approach, the holdout error estimate maybe be atypical if we happen to get an "unfortunate" split

1. Hold-out set approach

The limitations of the hold-out set can be overcome with different resampling methods at the expense of higher computational costs

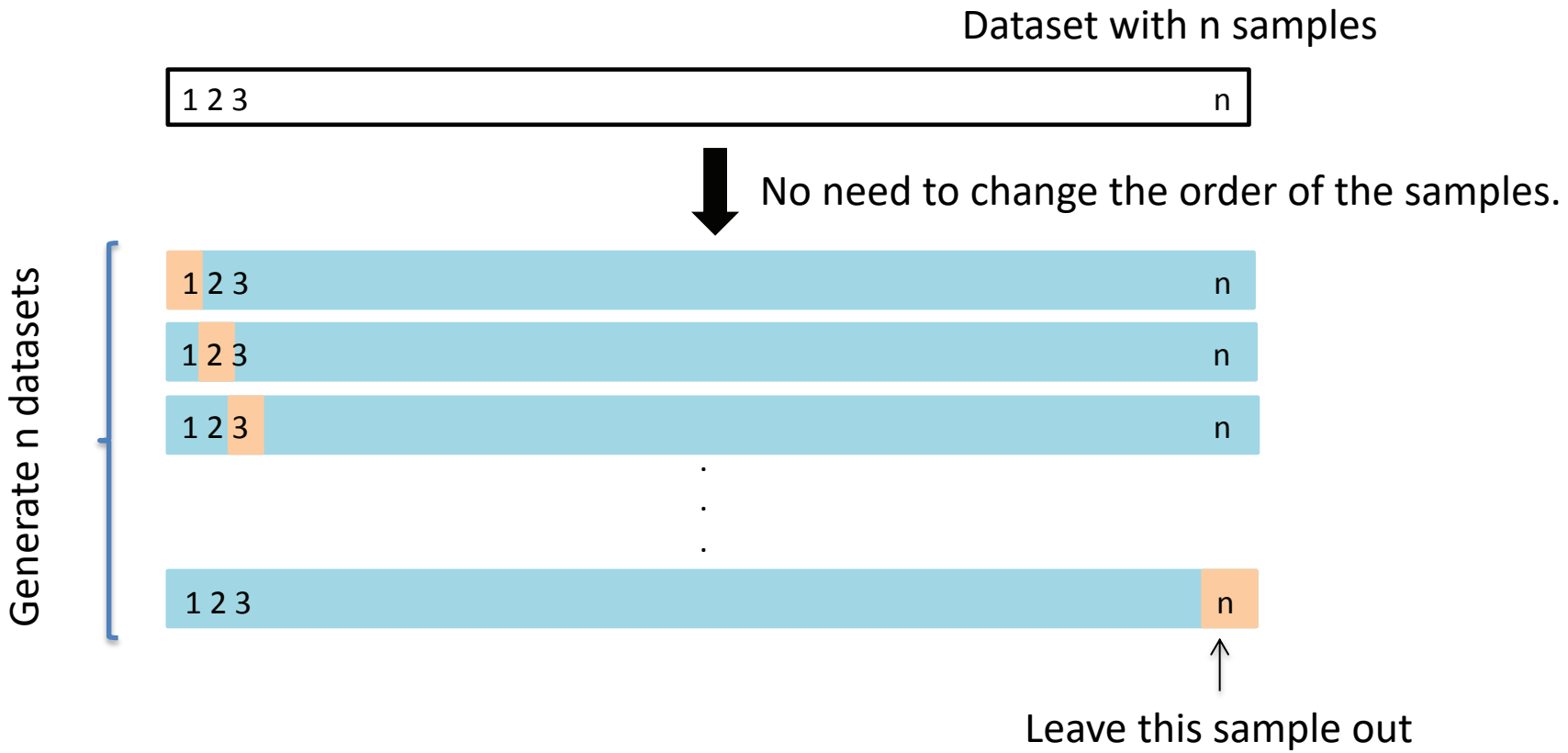
- Cross Validation
  - Leave-one-out Cross-Validation (aka Jackknife)
  - K-Fold Cross-Validation
  - Random resampling
- Bootstrap

2. Leave-one-out CV



Q: Why no need to change the order of the samples?

2. Leave-one-out CV



A: Every reordering of the samples leads to the exact same n datasets, just in different order.

2. Leave-one-out CV

Leave-one-out is the degenerate case of K-Fold Cross Validation, where K is chosen as the total number of samples (K=n)

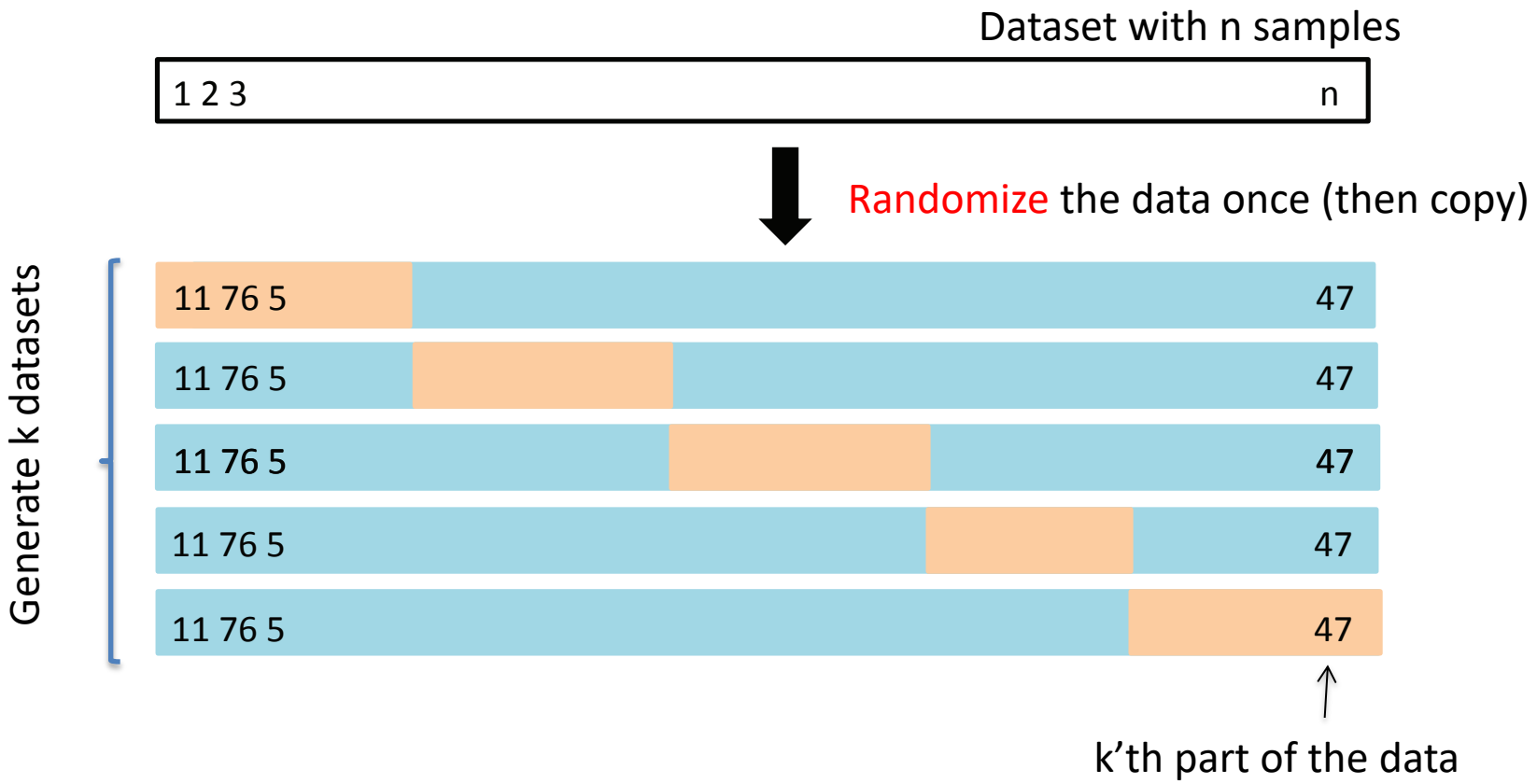
- For a dataset with n examples, perform n experiments
- For each experiment use n-1 data points for training and the remaining one data point for testing

2. Leave-one-out CV

The Leave-one-out CV method has two drawbacks

- The number of generated data sets is fixed (corresponds to n)
- A test data set consists of only one sample

3. K-fold CV



In this example, k=5.

3. K-fold CV

Create a K-fold partition of the the dataset

- For each of k experiments, use k-1 folds for training and the remaining fold for testing
- K-1 folds correspond to k-1/k * 100% of the data. For k=10, this is 90%.
- For k=n we obtain Leave-one-out CV.

Extensions

- Leave-2-out CV: $\binom{n}{2}$ different data sets (total number of)

$$\binom{100}{2} = 4950$$

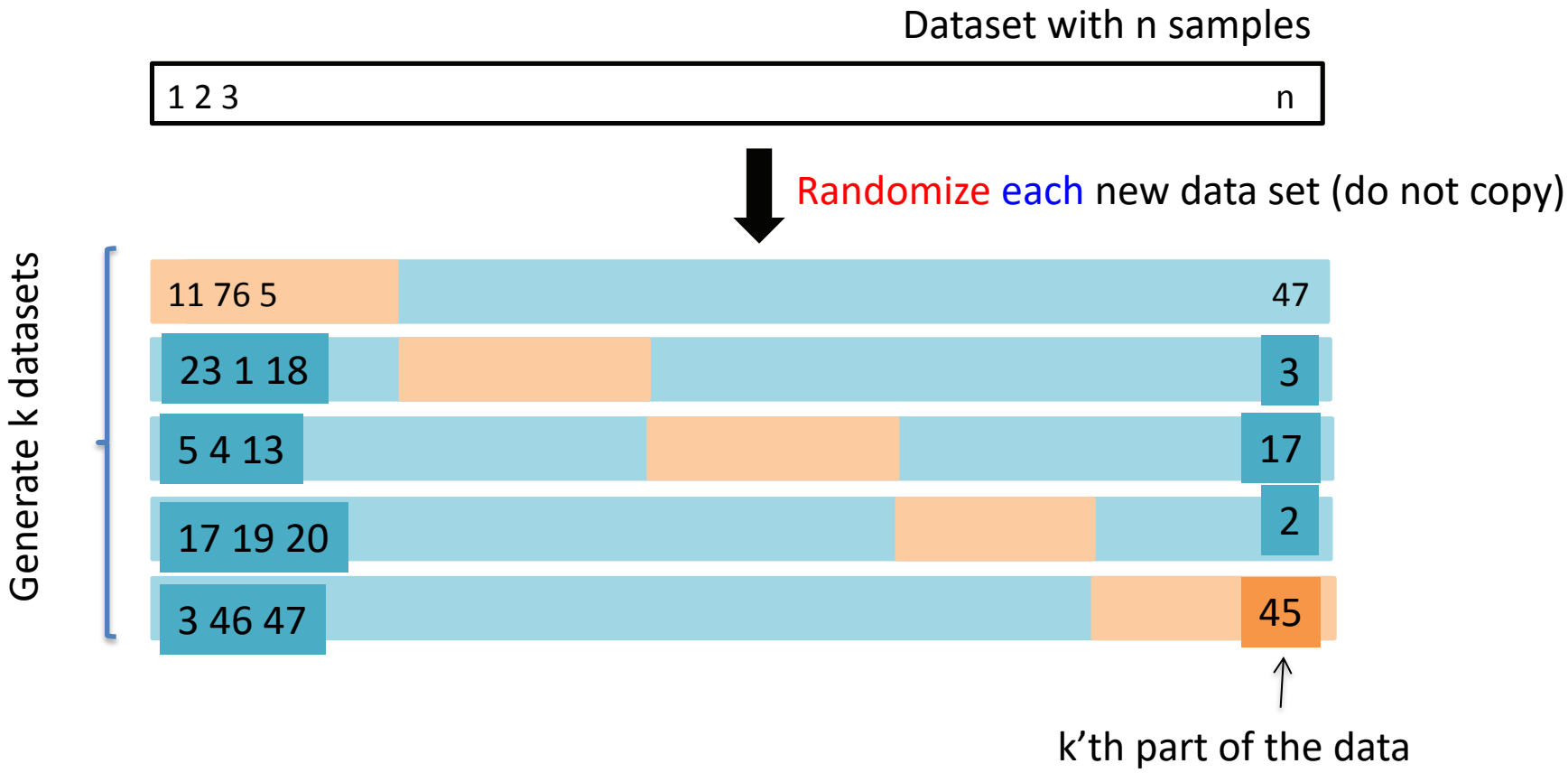
- Leave-m-out CV: $\binom{n}{m}$ different data sets (with m<n)

$$\binom{100}{10} = 10^{13}$$

Extensions

- Random resampling is an approximation of Leave-m-out CV that can be practically realized.

4. Random resampling



4. Random resampling

- Random resampling is similar to K-fold CV with the difference that each new data set is based on a new random sampling.
- K should be chosen higher than for K-fold CV.
- The disadvantage is that it can happen that not every data point is used for the training or testing of the model. For Leave-one-out CV and K-fold CV this cannot happen due to the non-randomization copying.

Summary

Assumption: One data set results in an estimated error of E (sensitivity, AUROC etc).

Summarizing estimates of errors:

- Hold-out set:
- Leave-one-out CV:
- K-fold CV:
- Random resampling

Mean errors!

$$E_{HOS} = E$$

$$E_{\mathrm{LOO}} = \frac{1}{n}\sum_{i=1}^{n} E_i$$

$$E_{\mathrm{KFCV}} = \frac{1}{k}\sum_{i=1}^{k} E'_i$$

$$E_{\mathrm{RES}} = \frac{1}{k}\sum_{i=1}^{k} E''_i$$

Summary

In practical applications the approaches

- Leave-one-out CV
- K-fold CV
- Random resampling

are commonly used.

For very small data sets one has to use

- Leave-one-out CV

because for k-fold CV, a small training data set may not allow to estimate the parameters of the statistical model.

Summary

The choice of the number of folds (k) depends on the data set.

In practical applications k=5 and k=10 are frequent choices.

Summary of this lecture

- Experimental design & reproducible research
  - Statistical model: Training data & Test data
  - Experimental space
- Resampling methods for assessing error measures
  - Cross validation (CV)
    - Hold-out set approach
    - Leave-one-out CV (aka Jackknife)
    - k-fold CV
    - Random resampling
- Emphasize was on understanding 'uncertainty'.

Part 2

Content of this lecture

- Generalization
- Resampling methods
  - Bootstrap
- Sampling from a distribution
- Standard error

} Formalization of ‘uncertainty’.

4. Generalization

Predictive statistical model*

General definition:

$$y = M(x; \beta)$$

prediction — predictor — parameter

Examples:

- Classification method (week 3)
- Regression model (week 6)

*: briefly called model

Learning a model

This means to estimate (fit, learn) parameters β of the predictive statistical model from training data (D_1) (by an algorithm).

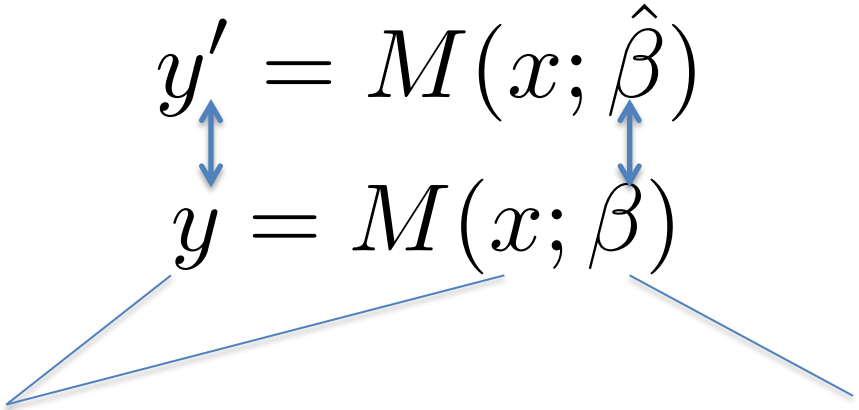
The estimated values are denoted by: $\hat{\beta}(D_1)$

These may be different from the true values: β

Resulting predictive statistical model: $y' = M(x; \hat{\beta})$

Errors

Due to the fact that the model parameters are estimated from data, there may be errors.



Error measure

We can define many different error measures by comparing the predicted value of the estimated model with the output of the true model.

Example: Mean squared error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i'\right)^2$$

Method: Regression $y' = \hat{g}(x, D_{train}) \sim \sum_{i=0}^{d} \beta_i x^i$

Generalization error

- Loss function (quadratic loss):

$$L(y, \hat{g}(x, D_{train})) = \left(y - \hat{g}(x, D_{train})\right)^2$$

- *Sample* test error:

$$E_{test} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} L(y_i, \hat{g}(x_i, D_{train}))$$

- *Population* test error:

$$E_{test}(D_{train}, n_{train}) = \mathbb{E}_P \left[L(y, \hat{g}(x, D_{train})) \right].$$

- (Expected) Generalization error:

All possible test data

$$E_{test}(n_{train}) = \mathbb{E}_{D_{train}} \mathbb{E}_P \left[L(y, \hat{g}(x, D_{train})) \right].$$

All possible training data of size n-train

Model assessment vs Model selection

- Last lecture, we assumed one model and wanted to assess its prediction capabilities:
  - Model assessment

- Today, we assume many models and (1) want to select the best model and (2) assess its prediction capabilities:
  1. Model selection
  2. Model assessment

1. Example

- Linear polynomial regression: degree 4

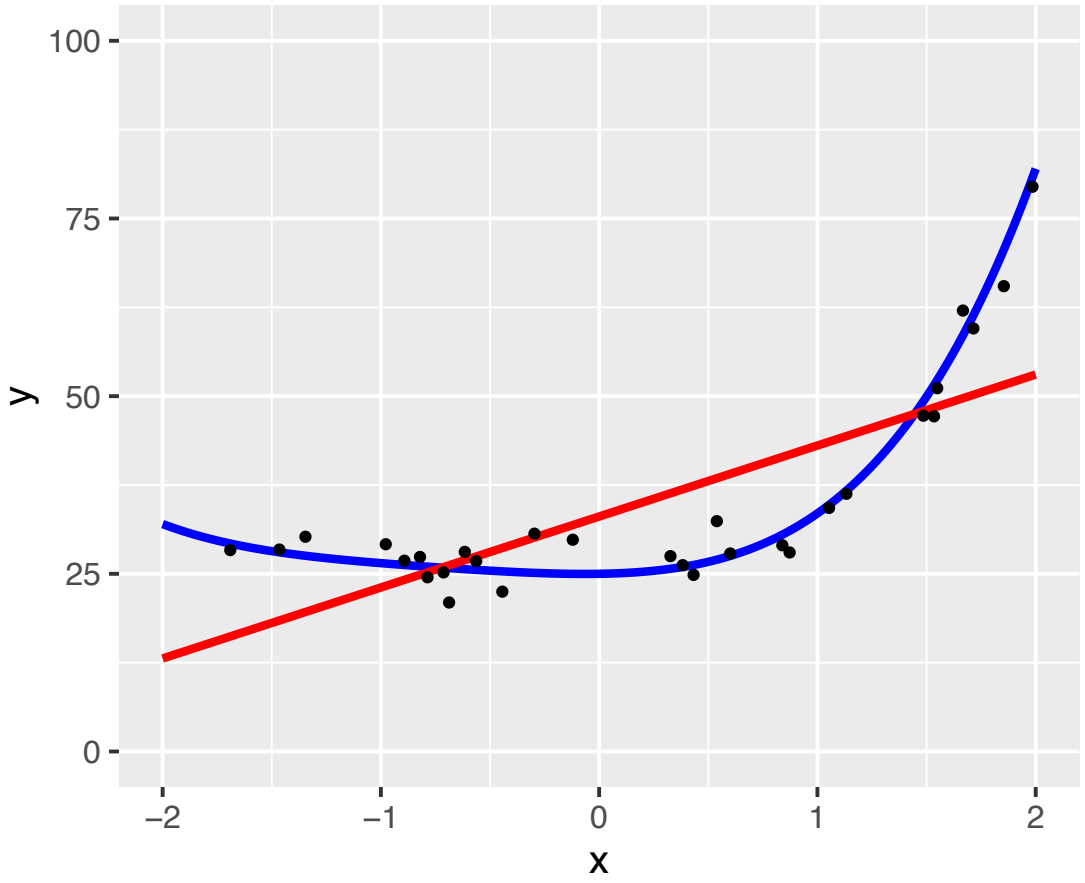
- True model:

$$f(x, \boldsymbol{\beta}') = 25 + 0.5x + 4x^2 + 3x^3 + x^4$$

- Regression model family: which d to use?

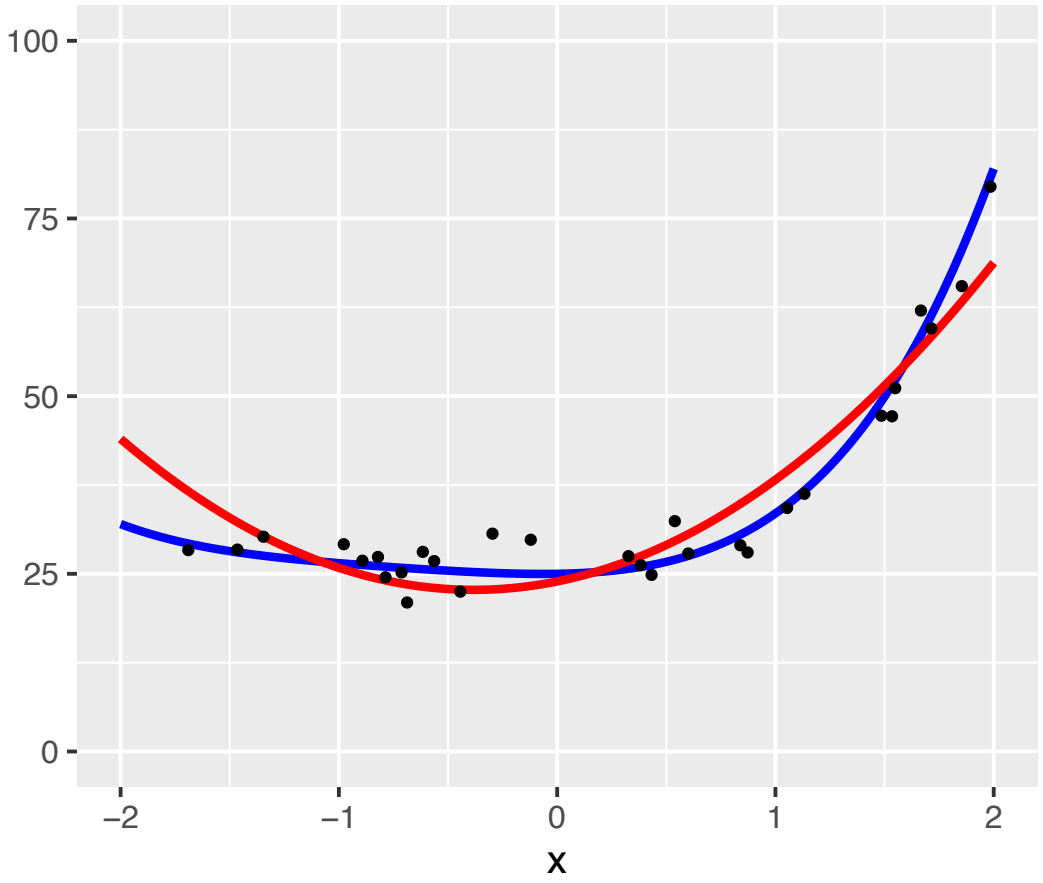
$$g(x, \boldsymbol{\beta}) = \sum_{i=0}^{d} \beta_i x^i = \beta_0 + \beta_1 x + \cdots + \beta_d x^d$$

Linear polynomial regression: fit degree 1



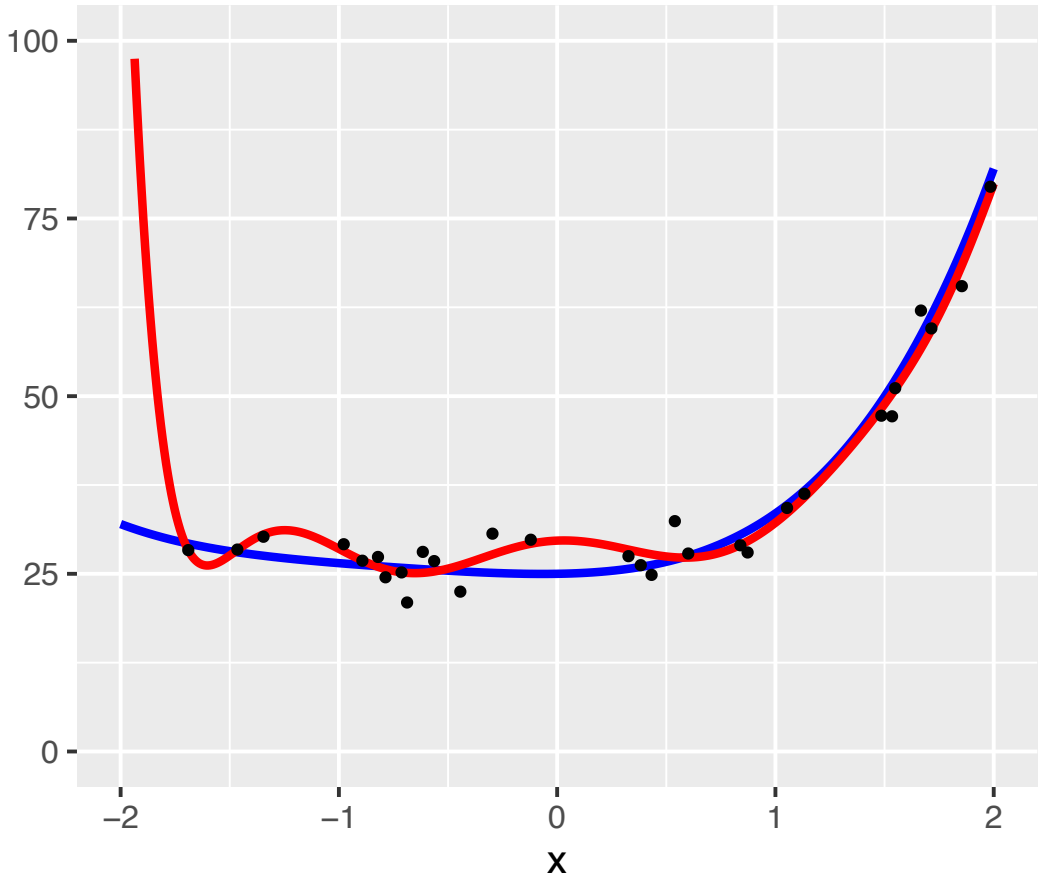
True model
Fitted model

Linear polynomial regression: fit degree 2



True model
Fitted model

Linear polynomial regression: fit degree 9



True model
Fitted model

Model complexity

Different models:

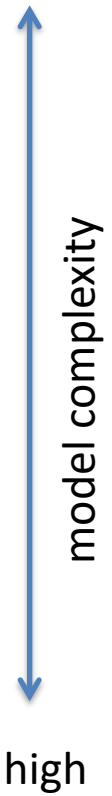
$$y \sim \beta_1 x$$

$$y \sim \beta_1 x + \beta_2 x^2$$

$$y \sim \sum_{k=1}^{3} \beta_k x^k$$

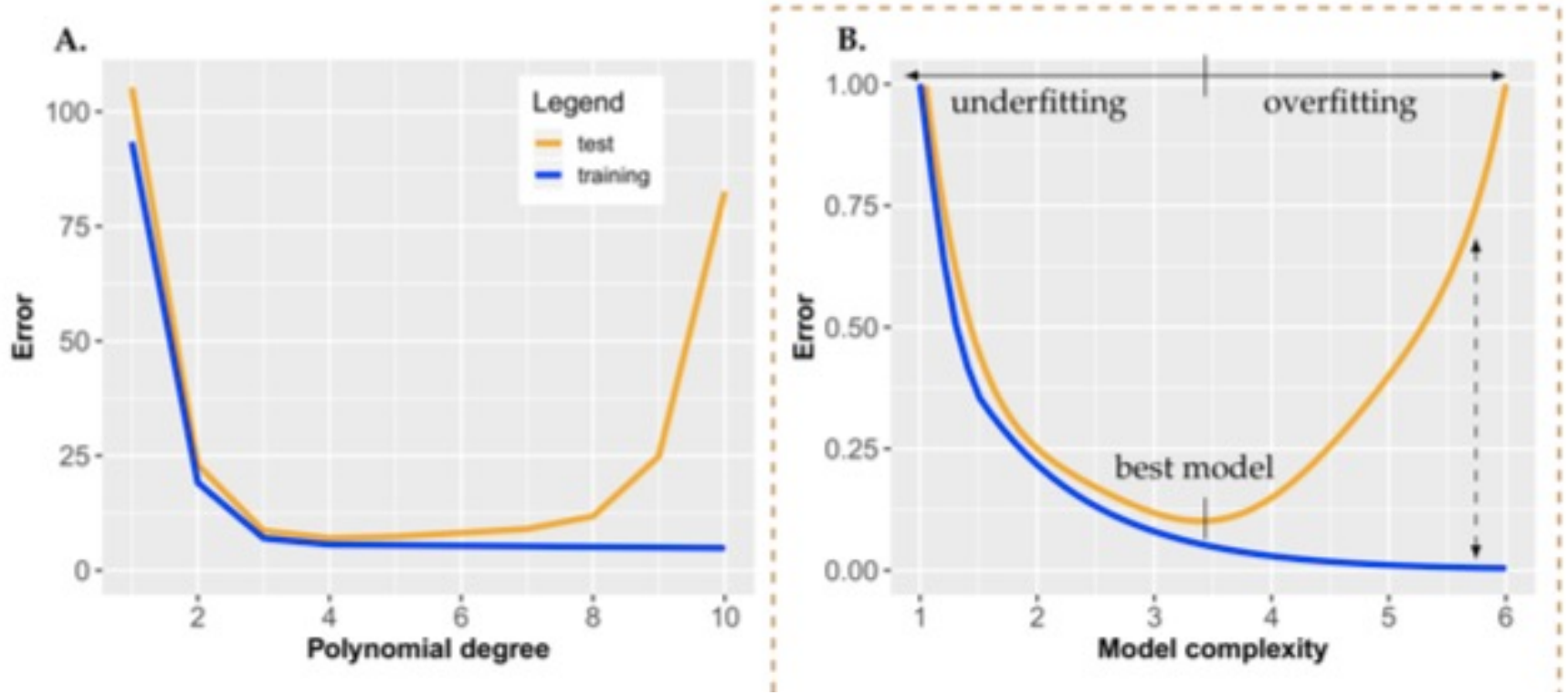
$$\dots$$

$$y \sim \sum_{k=1}^{8} \beta_k x^k$$



Results

Here we use large test data (avoid resampling), n-train is fixed.



Numerical results (linear polynomial regression)

Idealized results (for all models)

Overfitting

Definition 2 (model overfitting). *A model with complexity c is called* **overfitting** *if, for the test error of this model, the following holds:*

$$E_{test}(c) - E_{test}(c_{opt}) > 0 \quad \forall c > c_{opt} \tag{50}$$

with

$$c_{opt} = \arg\min_{c} \left\{ E_{test}(c) \right\} \tag{51}$$

$$E_{test}(c_{opt}) = \min_{c} \left\{ E_{test}(c) \right\} \tag{52}$$

Models are too complex.

Comparative measure.

Underfitting

Definition 3 (model underfitting). *A model with complexity c is called* **underfitting** *if, for the test error of this model, the following holds:*

$$E_{test}(c) - E_{test}(c_{opt}) > 0 \quad \forall c < c_{opt}. \tag{54}$$

Models are not complex enough (too simple).

Comparative measure.

Generalization

Definition 4 (generalization). *If a model with complexity c holds*

$$|E_{test}(c) - E_{train}(c)| < \delta \text{ with } \delta \in \mathbb{R}^+, \tag{56}$$

we say the model has good generalization capabilities.

With a small delta.

Possible values:

- Ideal: delta=0
- Real: delta ~ 0.01

Comparative measure.

Decompose the generalization error

- Bias-variance tradeoff (theory, because based on the generalization error):

All possible training data of size n-train

All possible test data

$$\mathbb{E}_D\,\mathbb{E}_{x,y}\left[\left(y-\hat{g}(x,D)\right)^2\right]=$$

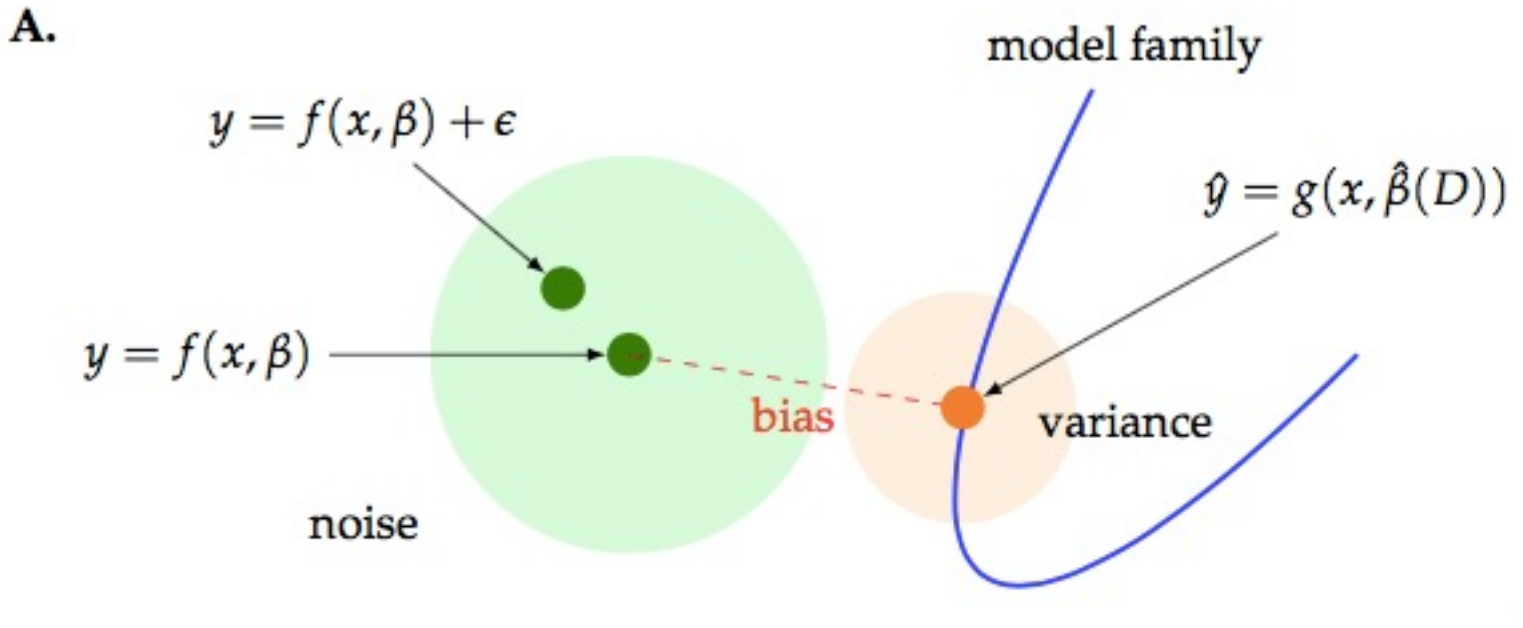
$$\mathbb{E}_{x,y}\left[\left(y-\bar{y}(x)\right)^2\right]+\mathbb{E}_x\,\mathbb{E}_D\left[\left(\mathbb{E}_D\left[\hat{g}(x,D)\right]-\hat{g}(x,D)\right)^2\right]+\mathbb{E}_x\left[\left(\bar{y}(x)-\mathbb{E}_D\left[\hat{g}(x,D)\right]\right)^2\right]$$

$$=\mathbb{E}_{x,y}\left[\left(y-\bar{y}(x)\right)^2\right]+\mathbb{E}_x\,\mathbb{E}_D\left[\left(\bar{g}(x)-\hat{g}(x,D)\right)^2\right]+\mathbb{E}_x\left[\left(\bar{y}(x)-\bar{g}(x)\right)^2\right]$$

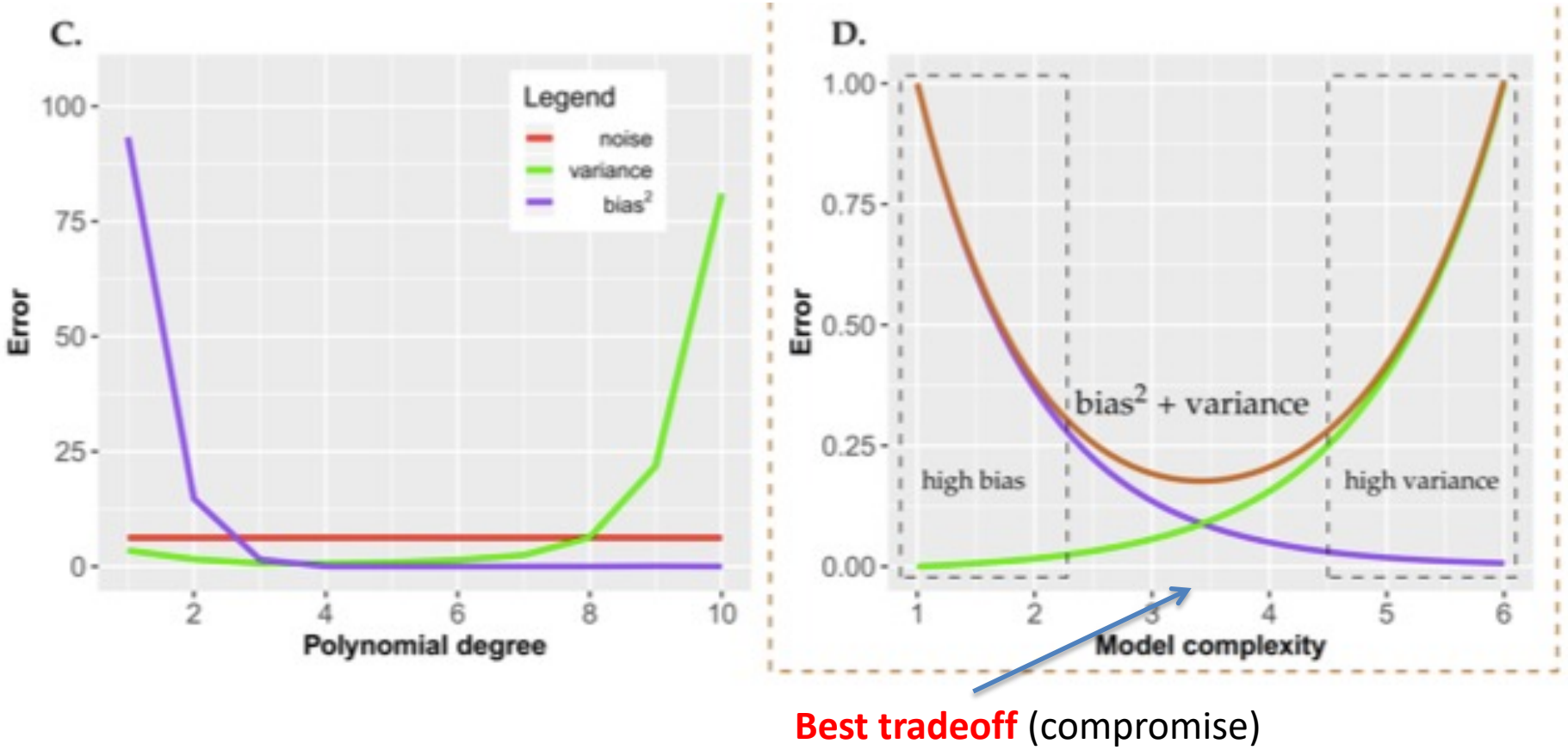
$$=\text{Noise}+\text{Variance}+\text{Bias}^2$$

Variance: model uncertainty
Bias: distance between true model and approximate model
Noise: random influences

Visualization of Bias vs variance



Bias-variance tradeoff



Best tradeoff (compromise)

Numerical results

Idealized results

Best model

We are interested in finding the simplest model 'c' that minimizes (tradeoff): $\text{bias}^2 + \text{variance}$

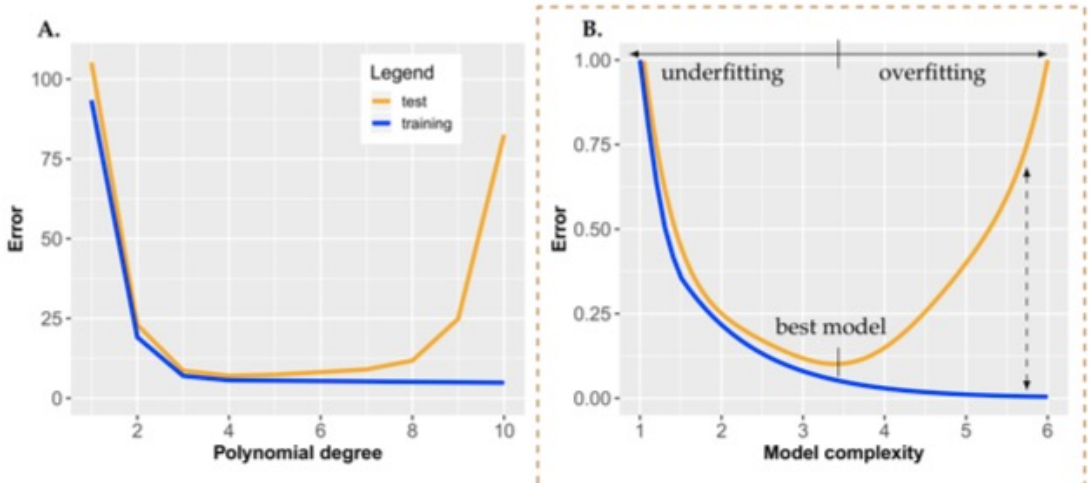
Here 'c' indicates the degree of the polynomial.

Practically, we approximate this by the minimum of the test error.

⟹ Best model

What is the error of the selected model?

- MSE



- For model selection, we use a training data set and a test data set.
- Problem: After we select a model we need to assess it (model assessment)

⇒ need a validation data set

Different categories of data

Data set 1 – training data: Used to learn the parameters of the model.

$$\hat{\beta}(D_1)$$

Data set 2 – test data: Used to assess the quality of the learned models.

$$MSE(D_2, D_1) = \frac{1}{n_2} \sum_{i=1}^{n_2} \left(y_i - y_i(D_1)'\right)^2$$

$$y' = M(x; \hat{\beta})$$

Data set 1

$$y = M(x; \beta)$$

Data set 2

Different categories of data

Data set 1- training data: Used to **learn the parameters** of the model.

Step 1

$$\hat{\beta}(D_1)$$

Data set 2 – test data: Used to **assess the quality of the learned models**.

Step 2

$$MSE(D_2, D_1) = \frac{1}{n_2} \sum_{i=1}^{n_2} \left(y_i - y_i(D_1)'\right)^2$$

$$y' = M(x; \hat{\beta})$$

Data set 1

$$y = M(x; \beta)$$

Data set 2

One additional data category

A new data set D_3 has been generated.

Data set 1 – training data: Used to learn the parameters of the model.

$$D = D_1 \cup D_2$$

$$\hat{\beta}(D_1)$$

Data set 2 – test data: Used to assess the quality of the learned models.

$$MSE(D_2, D_1) = \frac{1}{n_2} \sum_{i=1}^{n_2} \left(y_i - y_i(D_1)'\right)^2$$

Data set 3 – validation data: Used to assess the quality of the selected model.

$$MSE(D_3, D_1) = \frac{1}{n_3} \sum_{i=1}^{n_3} \left(y_i - y_i(D_1)'\right)^2$$

Practical realization

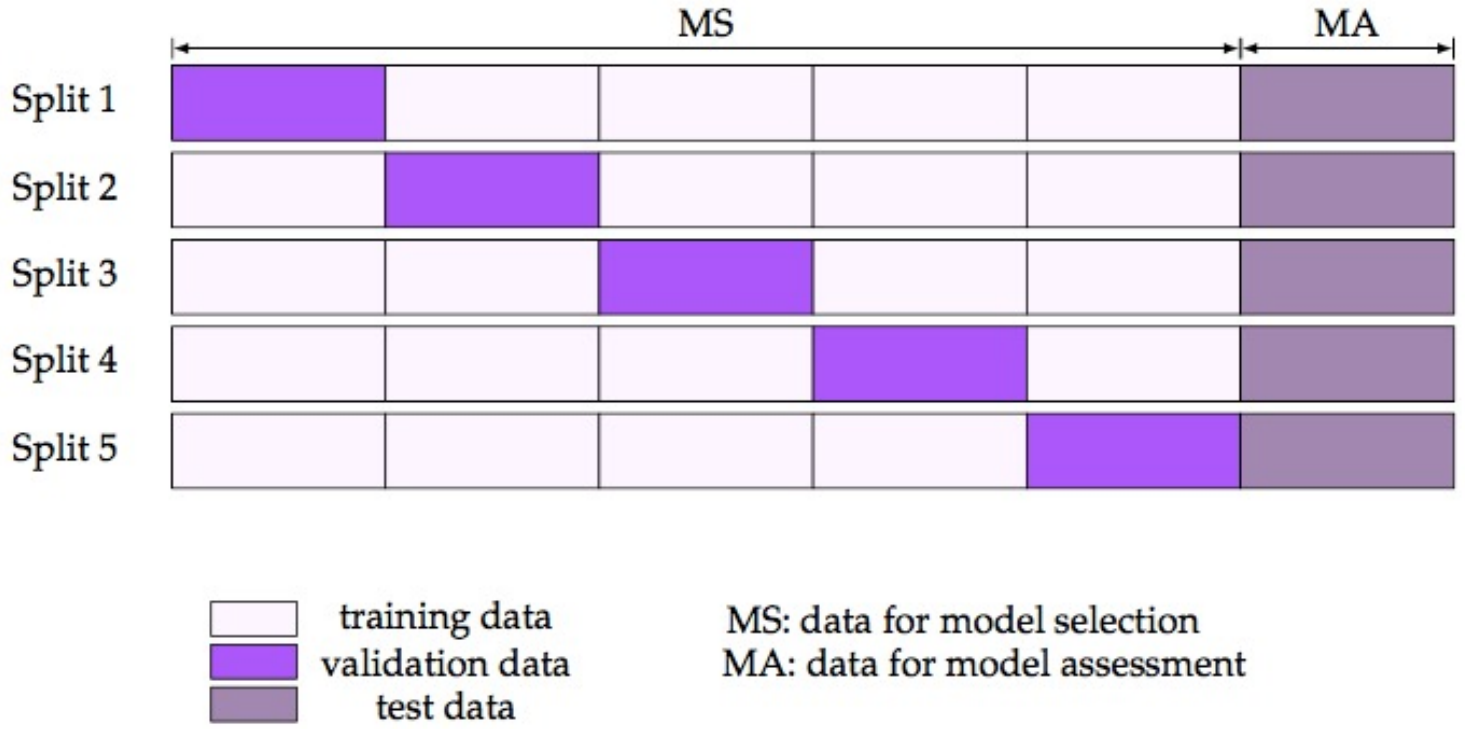


Figure 5. Visualization of cross-validation. Before splitting the data, the data points are randomized, but then kept fixed for all splits. Neglecting the column MA shows a standard five-fold cross validation. Consideration of the column MA shows a five-fold cross validation with holding-out of a test set.

Practical realization

Data are precious (expensive to generate).

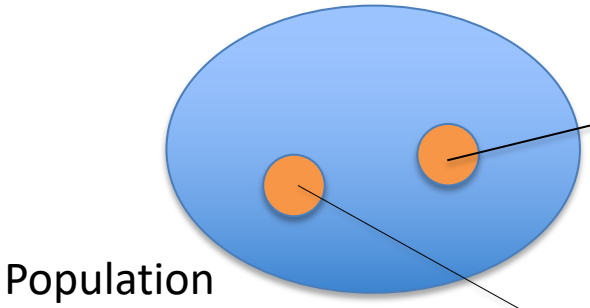
For this reason, usually, there are not (arbitrarily) many data sets available.

Approximation: Use one data set in a clever way.

 Resampling methods

Ultimate goal

Summarizing formulation:



$$D = D_1 \cup D_2$$

Sample size n_1 & n_2 is crucial.

We would like to achieve:

Sample error

$$MSE(D_3, D_1) \approx MSE(D_2, D_1)$$

for every data set D_3 (with n_3 sufficiently large).

Ideally for $n_3 \to \infty$ population error measures.

Remark

Reason for introducing validation data

Example:

- Students are learning at home

$$MSE(D_2, D_1) = \frac{1}{n_2} \sum_{i=1}^{n_2} \left(y_i - y_i(D_1)' \right)^2$$

- Students are writing an examination

$$MSE(D_3, D_1) = \frac{1}{n_3} \sum_{i=1}^{n_3} \left(y_i - y_i(D_1)' \right)^2$$

Remark

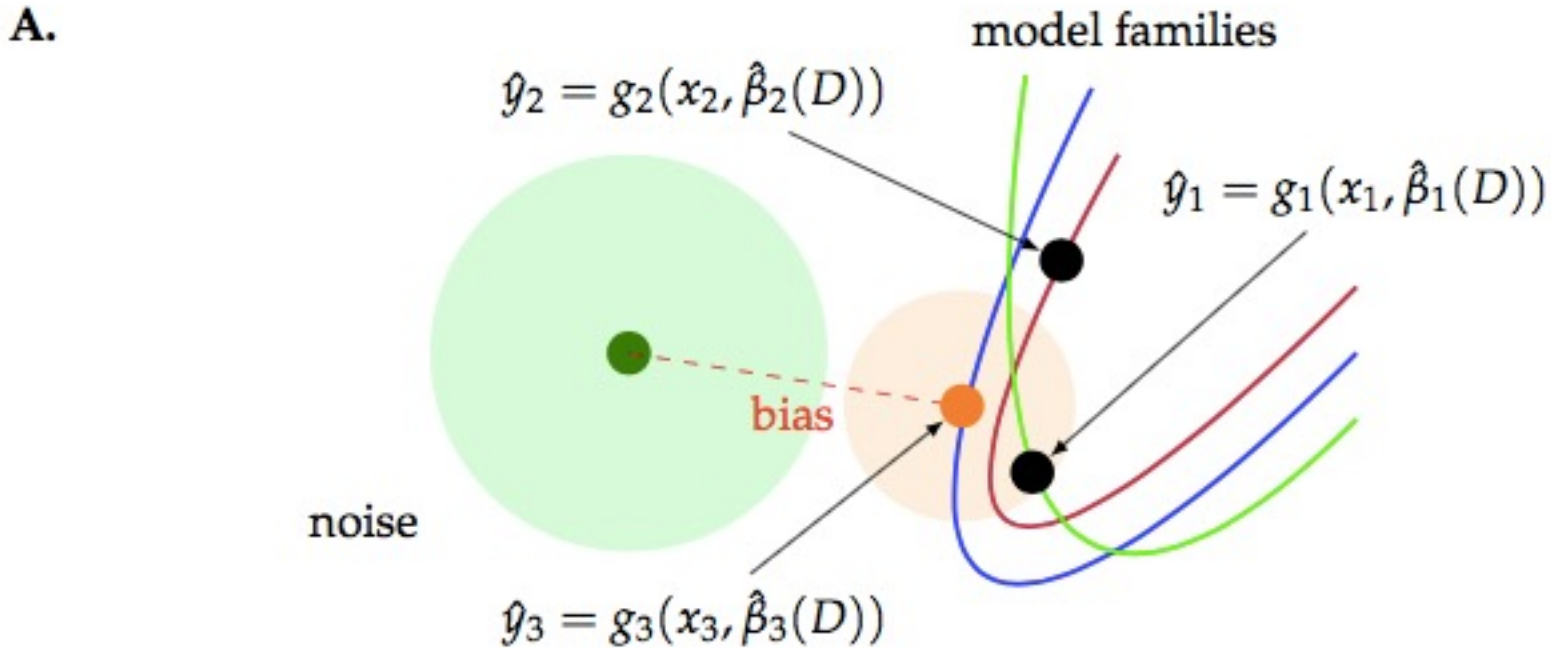
- We do not want to use the same data for two different decisions:
  - model selection
  - model assessment

Remark

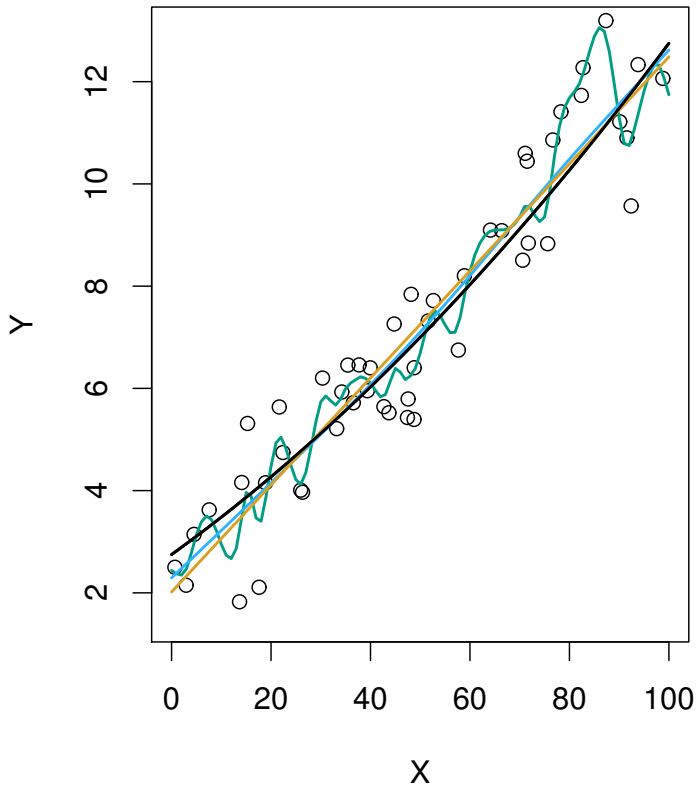
- If we have just one model and we want to do model assessment, we need:
  - Training data
  - Test data

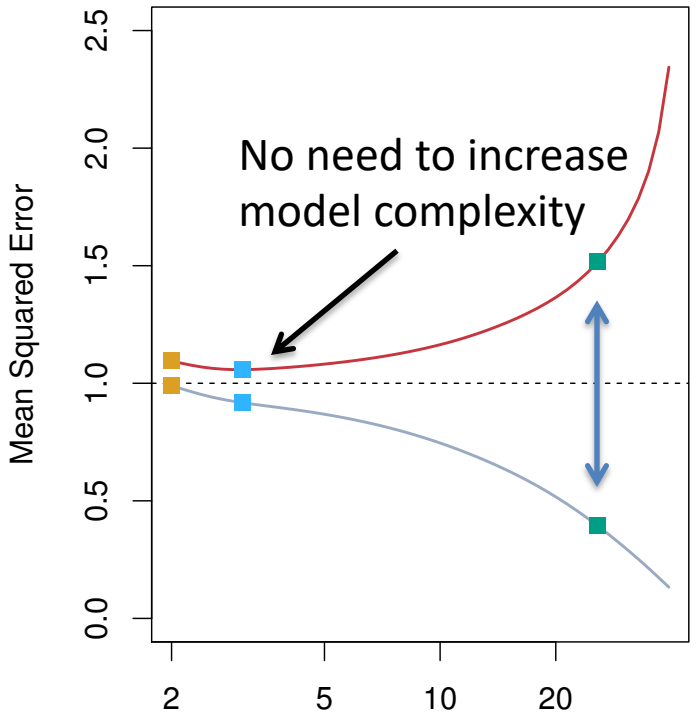
- If we have many models and we want to do model selection and model assessment of the best model, we need:
  - Training data
  - Test data
  - Validation data

Model selection



2. Example: Generalization





Different true model (compared to previous data)

Model selection

- Application of resampling methods for model selection

- Find the optimal degree d for a regression model.

3. Example

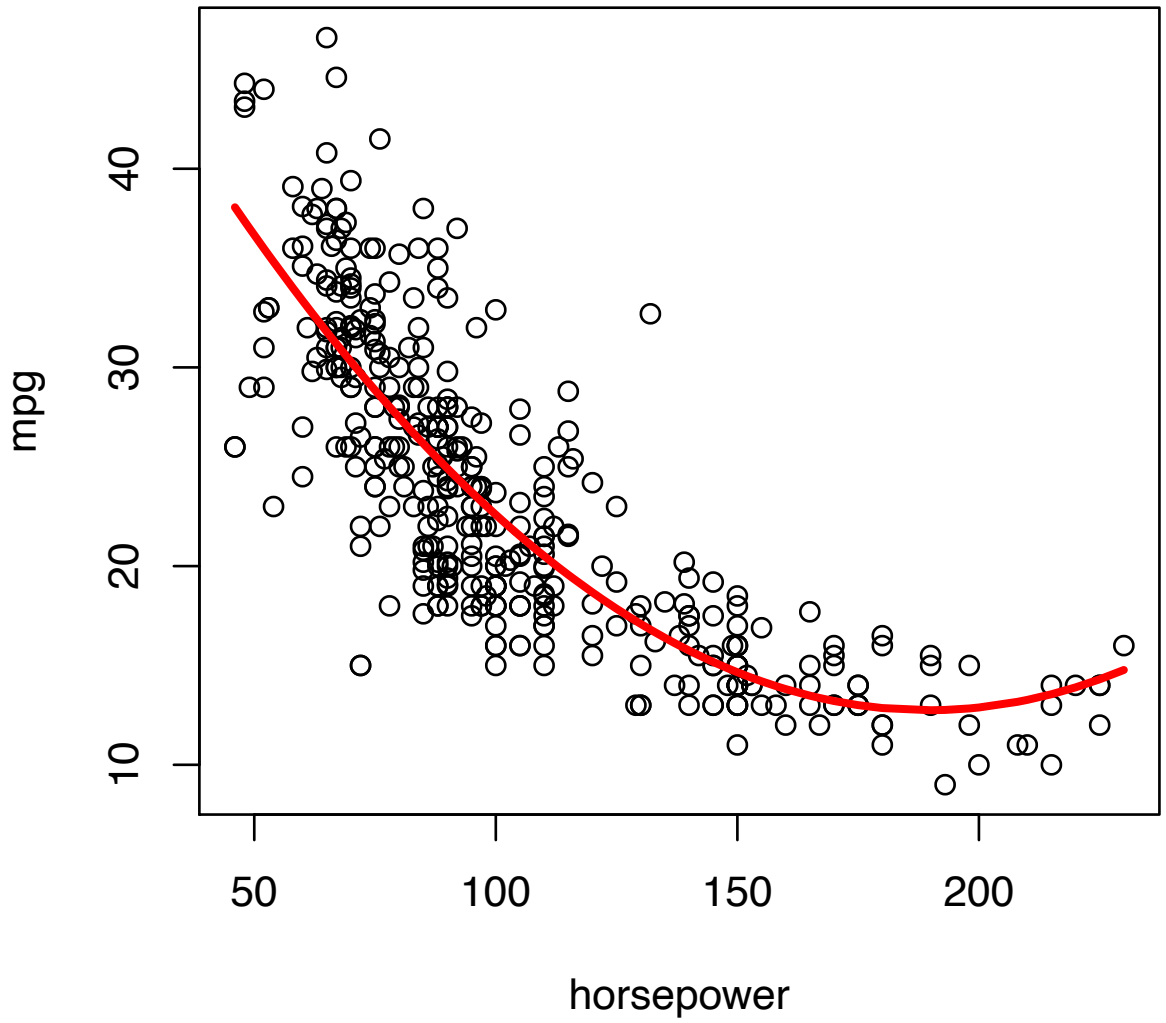
Car data: characteristics of different models

mpg	cylinders	displacemen	horsepower	weight	acceleration	year	origin	name
18	8	307	130	3504	12	70	1	chevrolet chevelle malibu
15	8	350	165	3693	11.5	70	1	buick skylark 320
18	8	318	150	3436	11	70	1	plymouth satellite
16	8	304	150	3433	12	70	1	amc rebel sst
17	8	302	140	3449	10.5	70	1	ford torino
15	8	429	198	4341	10	70	1	ford galaxie 500
14	8	454	220	4354	9	70	1	chevrolet impala
14	8	440	215	4312	8.5	70	1	plymouth fury iii
14	8	455	225	4425	10	70	1	pontiac catalina
15	8	390	190	3850	8.5	70	1	amc ambassador dpl
15	8	383	170	3563	10	70	1	dodge challenger se

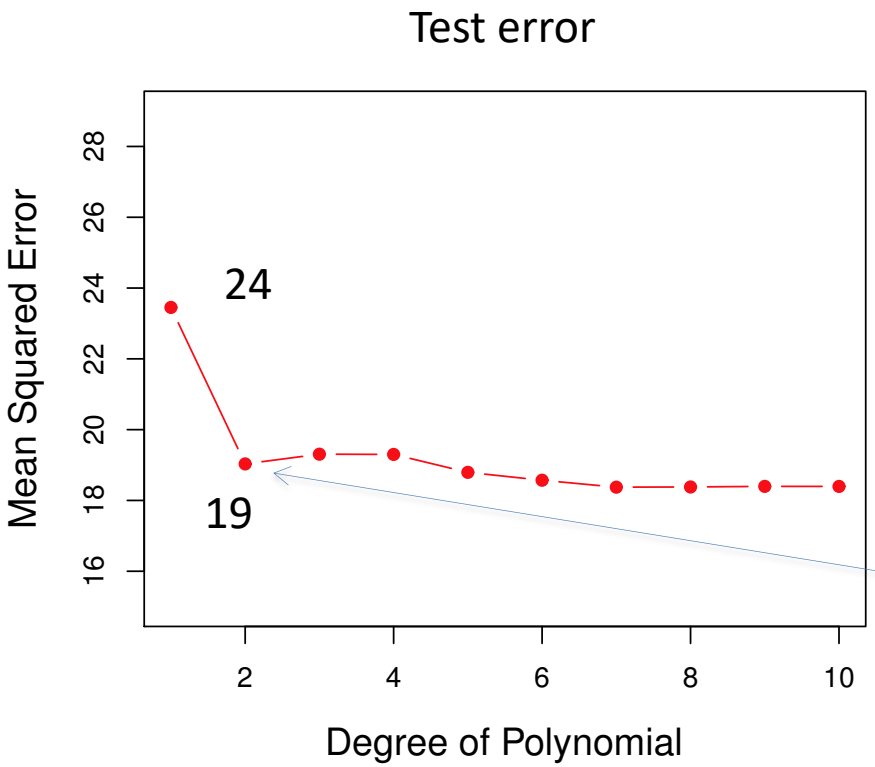
Y X

We focus on these

Regression model



Hold-out set approach

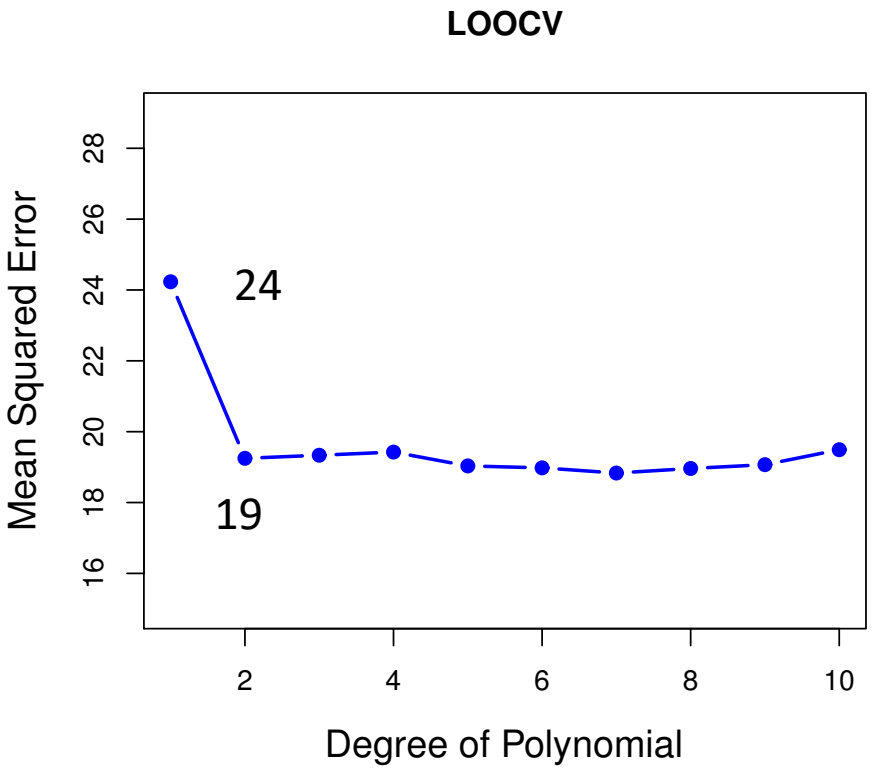


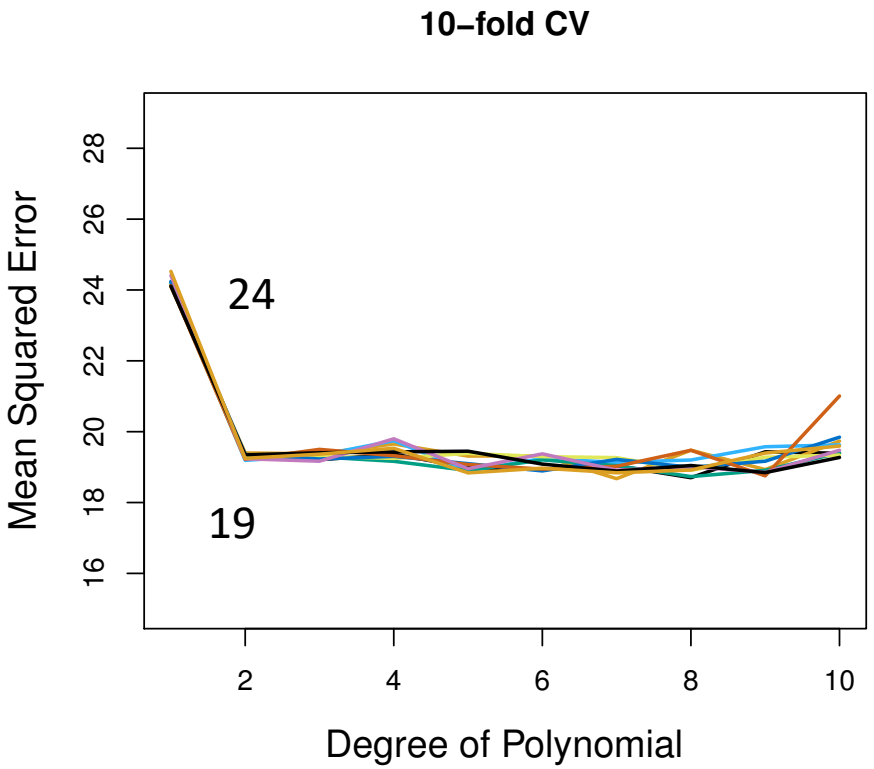
In order to avoid overfitting select the model with lowest complexity and reasonable MSE.

Simple model

Fit different models of increasing complexity.

LOOCV vs 10-fold CV





Repeat 10-fold CV 9 times

Remark: the different resampling methods lead to the same result

5. Bootstrap

Baron Münchhausen

The Surprising Adventures of Baron Munchausen by **Rudolf Erich Raspe** (1736 – 1794)

Bootstraping

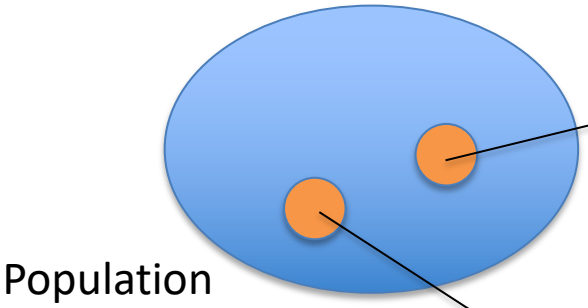
- Boostraping is a resampling method.
- It has been introduced by Brad Efron

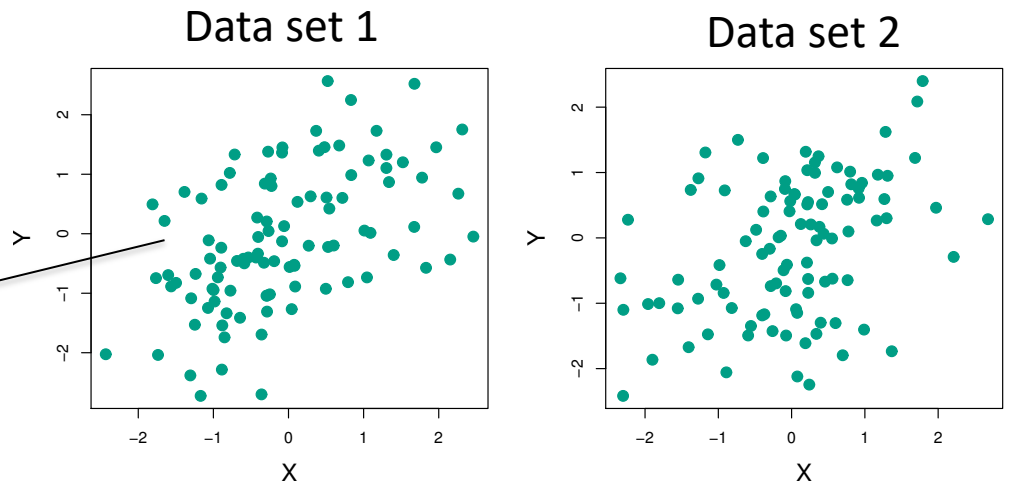
- Boostrap is used to quantify the uncertainty of a parameter (point estimator).
- Boostrap estimates the sampling distribution of a parameter.

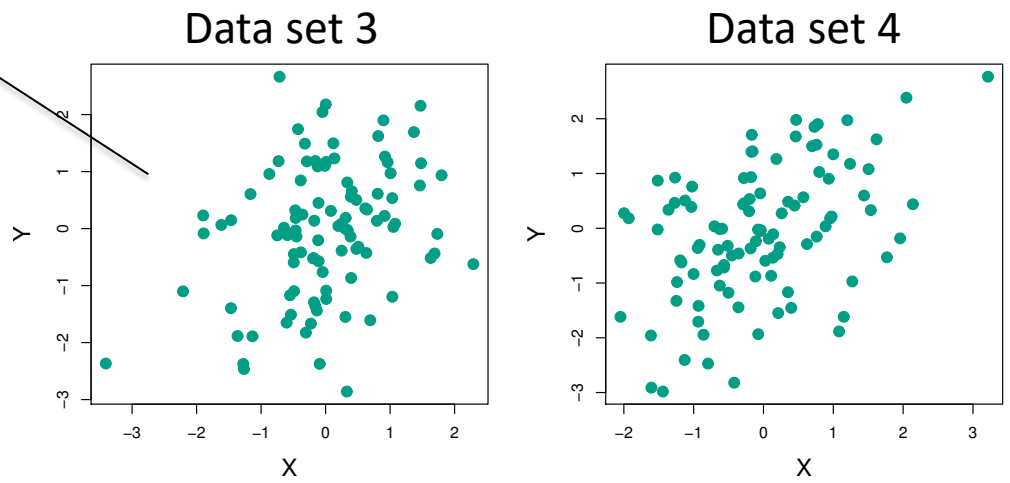
Simulated data – no Bootstrap!

Theoretical Situation

Sample infinite many
data sets from the population





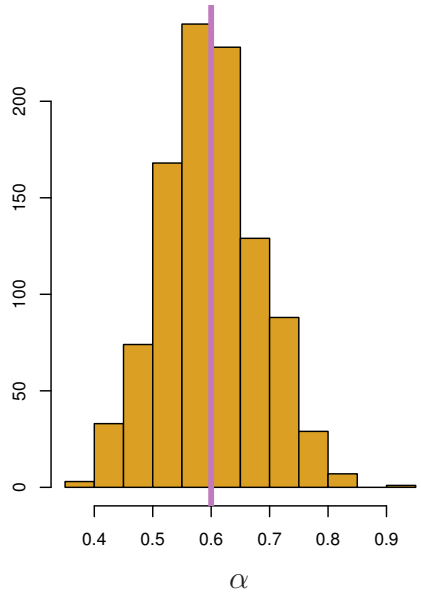


$$\hat{\alpha}_i = f(D_i)$$

Any function of the data.

Fixed (finite) sample size n.

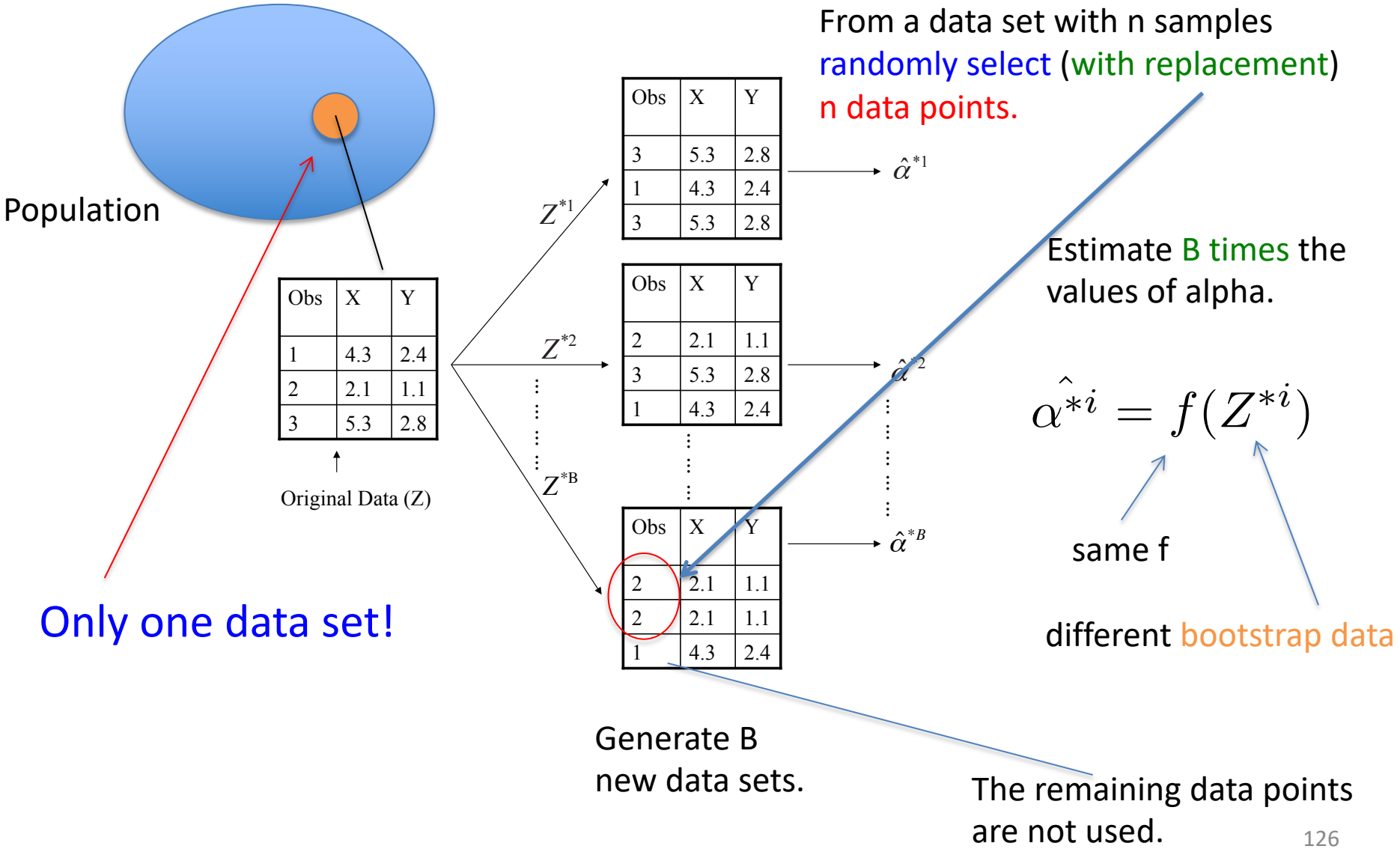
Distribution



$$\text{true value of } \alpha = 0.6$$

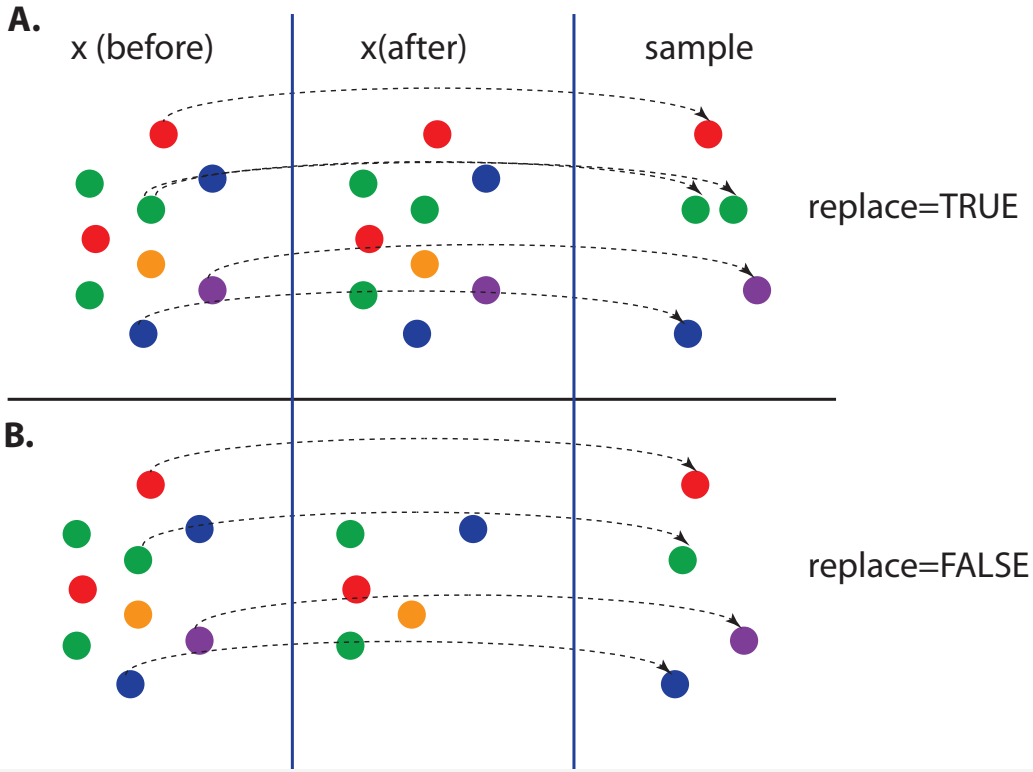
Population results for 1000 independent data sets

Bootstrap method



Sample function in R

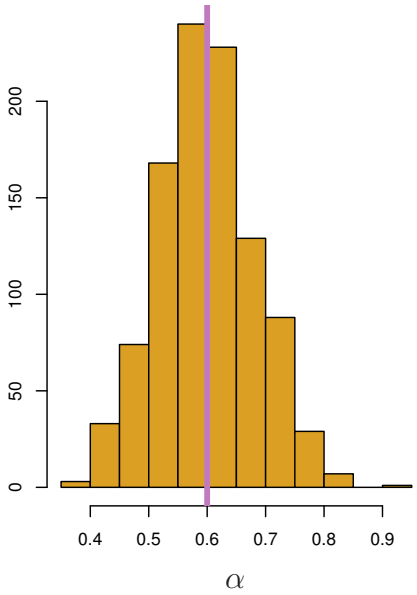
sample:




```
> sample(1:5, 5 , replace = TRUE)
```

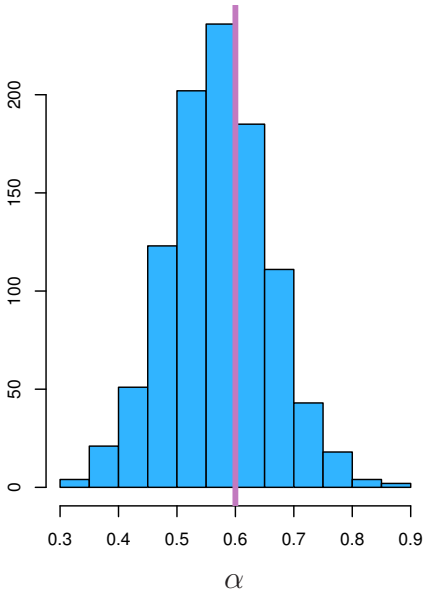
Comparison of Distributions

Theoretical results
1000 independent data sets

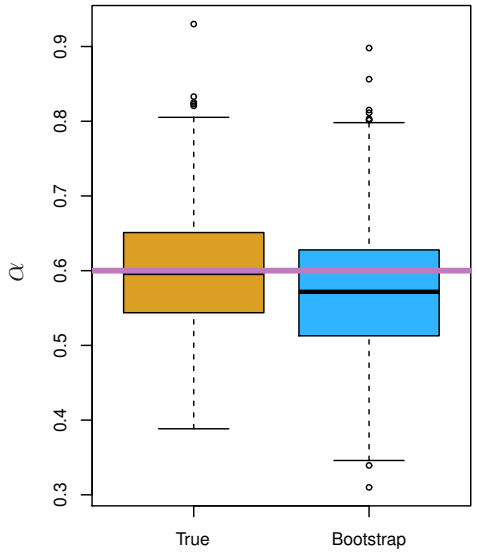


Population results

Practical results
1 data set



Bootstrap results



Remarks

- Bootstrap is resampling with replacement.
- Multiply occurring data points are used.
- If the sample size of the data set is of size n, then resample n data points (not unique!).
- The data set is not split, because we do not use the un-sampled data points. This is in contrast to, e.g., cross validation.

Remarks

- Bootstrapping is used for parameter estimation problems (see example), not for model selection (e.g. CV).

- Specifically, use Boostrap to quantify the uncertainty of model parameters.

- Reason: The Bootstrap samples are 'quite similar' to each other (in contrast to CV) and hence underestimate the test error.

6. Sampling from a distribution

Generating samples

How do we actually do this?

Generate samples by an experiment:

$$D = \{x_1, \ldots, x_n\}$$

Generate samples from a distribution:

$$x_i \sim f(x; \alpha)$$

statistical model for the experiment

random variable

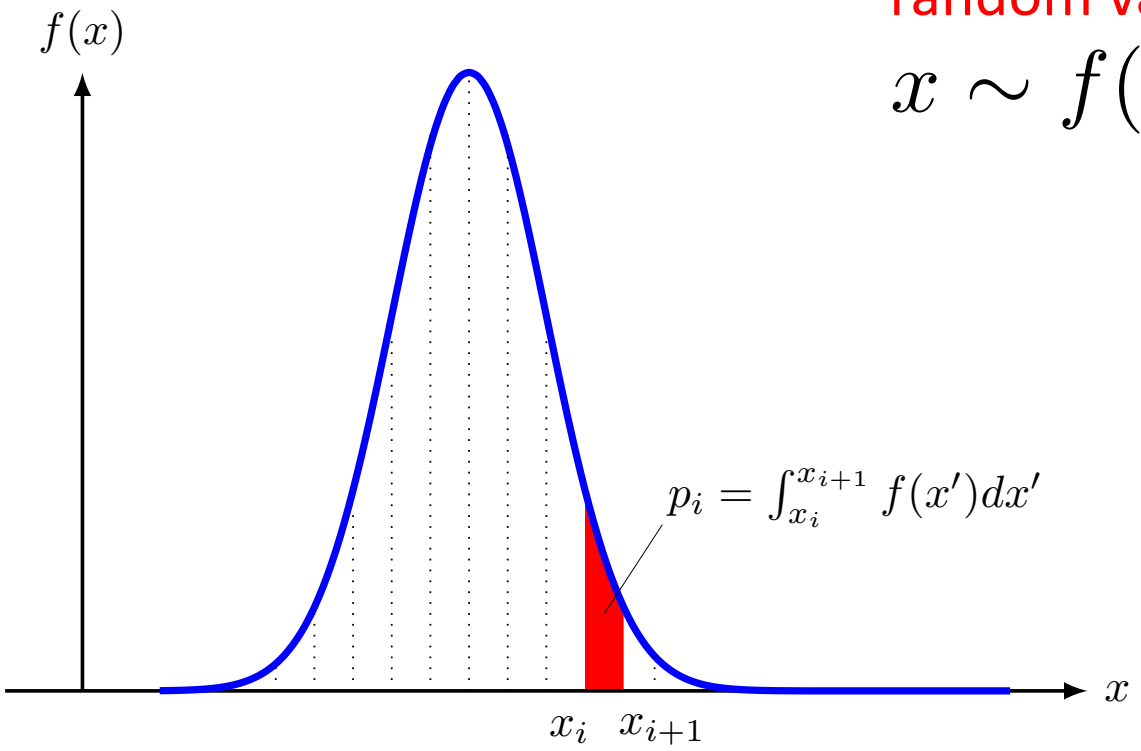
draw a sample

parameter

Drawing a sample

random variable

$$x \sim f(x;\alpha)$$



$$\Delta x = x_i - x_{i+1}$$

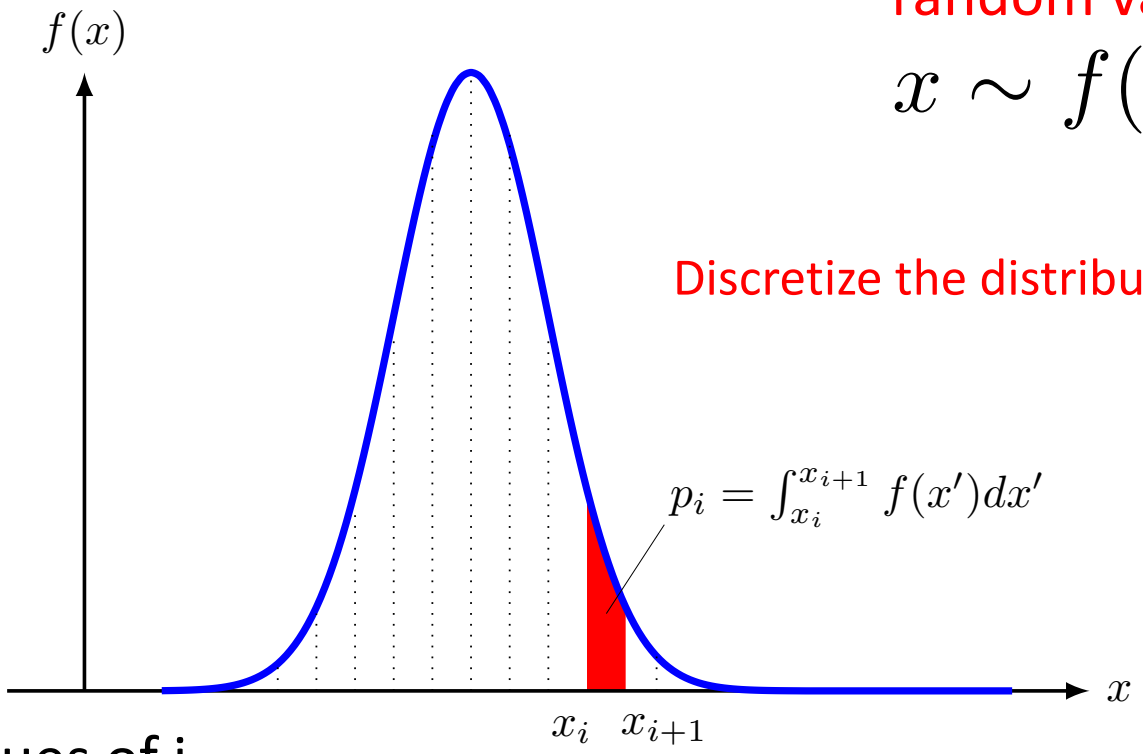
$$\lim_{i \to \infty} \Delta x \to 0$$

Probability density

Drawing a sample

random variable

$$x \sim f(x;\alpha)$$



Discretize the distribution

For finite values of i,
discrete prob. distribution:

$$P = \{p_1, p_2, \ldots, p_n\}$$

$$p_i = Prob(X = x_i')$$

$$x_i' = \frac{x_i + x_{i+1}}{2}$$

How to utilize p_i practically

Assume we have the following discrete probability distribution:

$$P = \{p_1, p_2, \ldots, p_n\} \qquad p_i = Prob(X = x_i')$$

Most simple way to utilize this in a physical sampling process: (without computer)

1. N: number of balls
2. Number of balls with label 'i': $N_i = N \times p_i$ rounding
3. Put all balls in an **urn**
4. Draw one ball randomly

R script – draw ball from urn

```
P <- c(0.1, 0.3, 0.2, 0.4)
n <- length(P)
N <- 100
Np <- N*P

coin.aux <- runif(1)  # auxilliary random number
coin <- N*coin.aux # draw one ball

label <- c()   # initialization of the label
flag <- 1
i <- 1
cs <- cumsum(Np)
while(flag){
    if(coin <= cs[i]){
        label <- i
        flag <- 0
    }
    i <- i + 1
}
print(c(label, coin))
```

R script – draw ball from urn

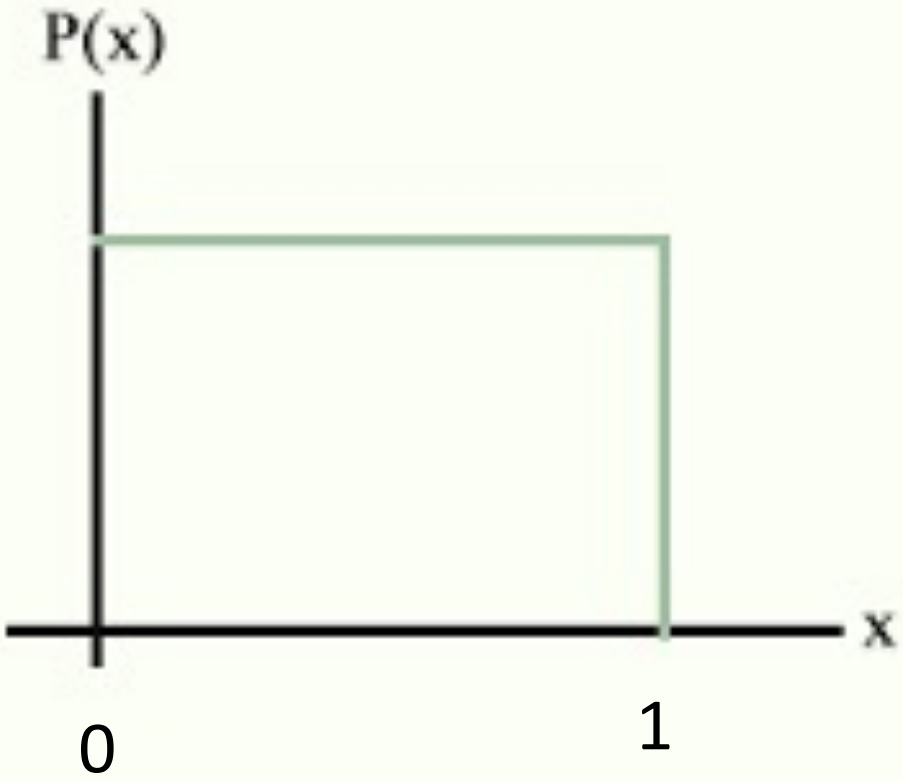
```r
P <- c(0.1, 0.3, 0.2, 0.4)
n <- length(P)
N <- 100
Np <- N*P

coin.aux <- runif(1)  # auxilliary random number
coin <- N*coin.aux # draw one ball

label <- c()   # initialization of the label
flag <- 1
i <- 1
cs <- cumsum(Np)
while(flag){
    if(coin <= cs[i]){
        label <- i
        flag <- 0
    }
    i <- i + 1
}
print(c(label, coin))
```

Just example numbers.

Sampling from an arbitrary distribution by utilizing random numbers from a uniform distribution.

The result we want.

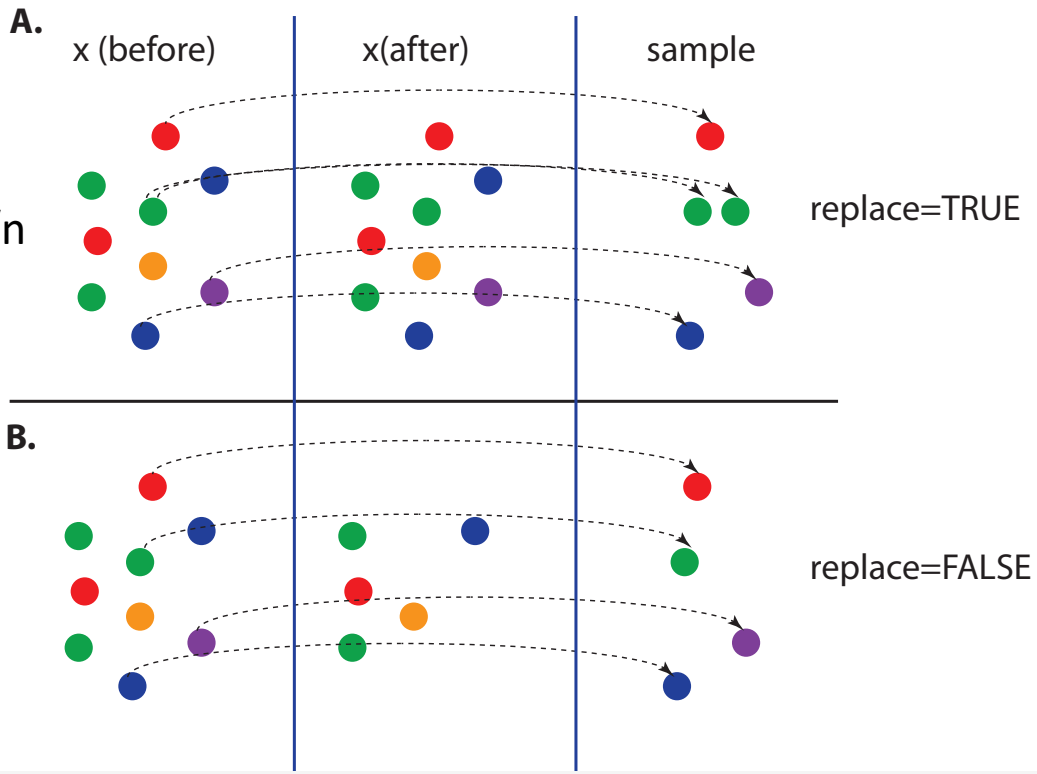
Uniform distribution



Alternative: Drawing samples with **sample**

sample:

Selection probability: 1/n for each ball

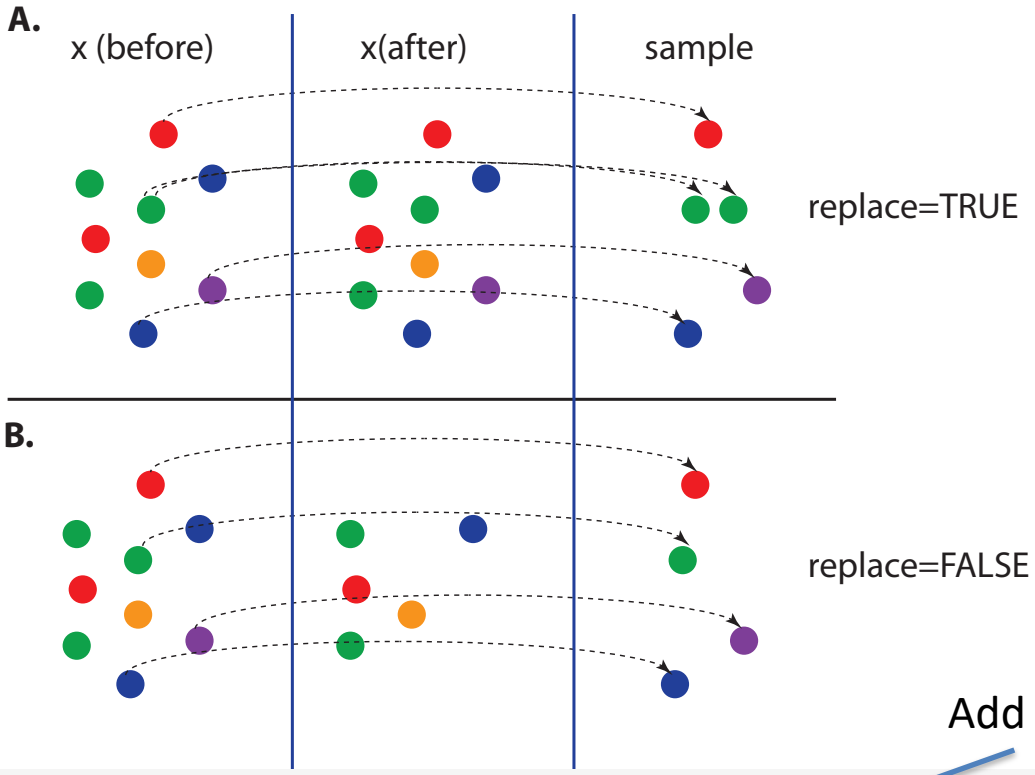



```
> sample(1:5, 20, replace = TRUE)
 [1] 5 4 5 4 1 1 4 5 4 1 1 5 2 5 1 4 3 2 5 2
```

Drawing samples with **sample**

sample:

Selection probability: p_i for each ball i



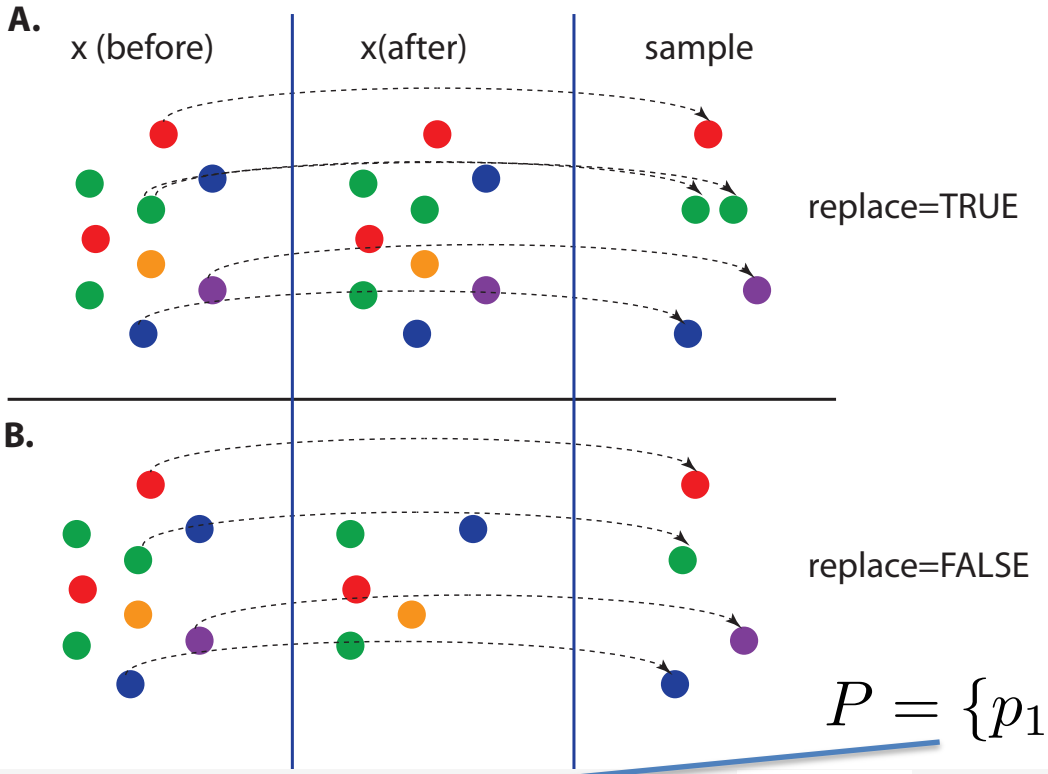
Add prob = P

```
> sample(1:5, 20, replace = TRUE)
 [1] 5 4 5 4 1 1 4 5 4 1 1 5 2 5 1 4 3 2 5 2
```

Drawing samples with **sample**

sample:

Selection probability: p_i for each ball i



$$P = \{p_1, \ldots, p_5\}$$

```
> sample(1:5, 20, replace = TRUE, prob = P)
 [1] 5 4 5 4 1 1 4 5 4 1 1 5 2 5 1 4 3 2 5 2
```

The order needs to correspond!

Summary

By using the ‘sample’ function we can draw (approximate) samples from arbitrary probability distributions.

$$x_i' = \frac{x_i + x_{i+1}}{2}$$

The quality of this approximation depends on the resolution (given by the number n) of the probability vector.

$$P = \{p_1, p_2, \ldots, p_n\}$$

Important probability distributions

Normal distribution (aka Gaussian distribution):

The density function of the normal distribution is defined by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad -\infty \le x \le \infty$$

This distribution depends on 2 parameters:

- Mean: μ
- Variance: σ^2

Normal distribution

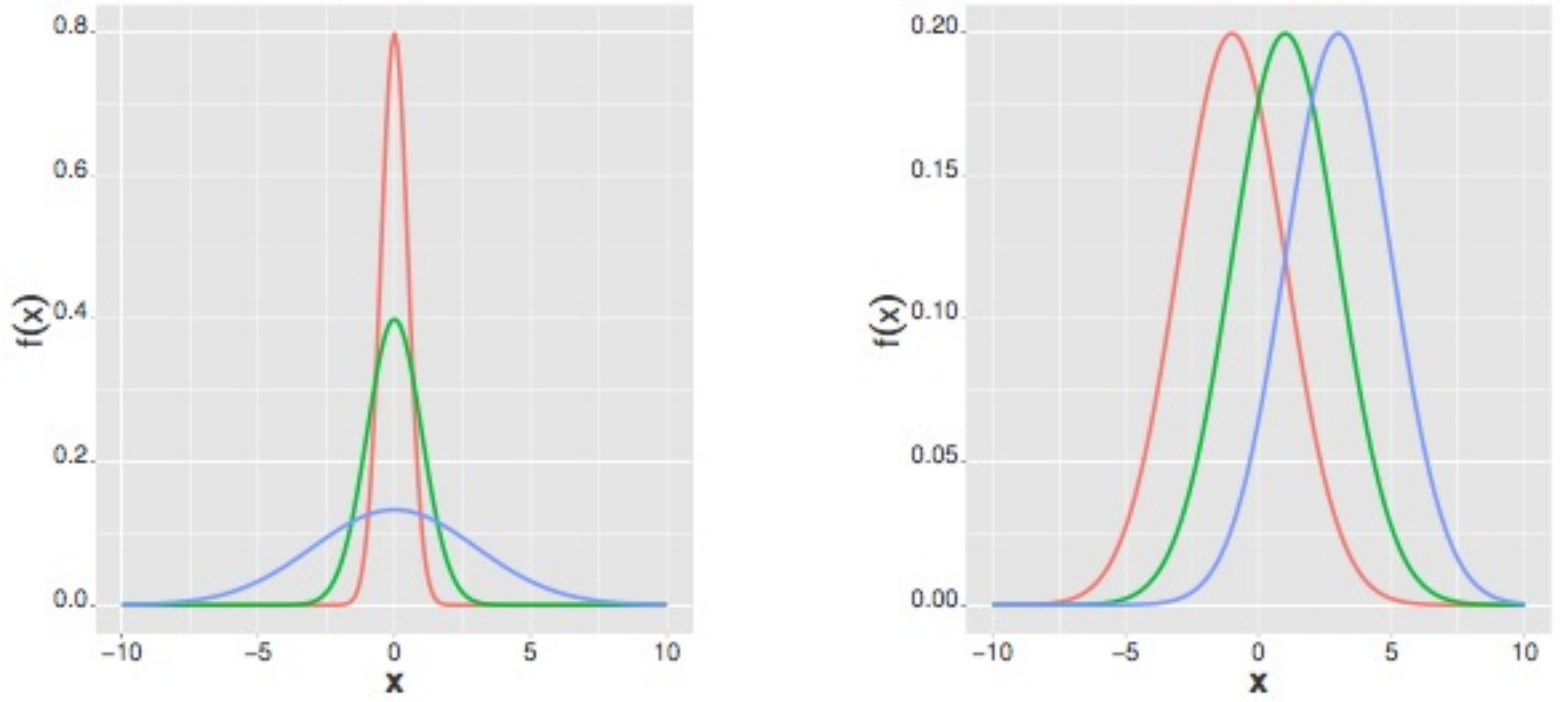


Figure 17.10: One-dimensional normal distribution. Left: Different values of $\sigma \in \{0.5, 1, 3\}$ for a constant mean of $\mu = 0$. Right: Different values of $\mu \in \{-1, 1, 3\}$ for a constant standard deviation of $\sigma = 2$.

Standard Normal distribution

An important special case of a normal distribution is the standard normal distribution defined by

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2}), \quad -\infty \leq x \leq \infty \qquad (17.58)$$

The standard normal distribution has a mean of 0 and a variance of 1.

Question:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}), \quad -\infty \leq x \leq \infty$$

Standard Normal

Z-transformation

Applications

The normal distribution plays an important role in:

- Statistical hypothesis testing
- Noise models (e.g. Regression models)
- Central limit theorem

The sample mean of a large number of independent random variables will be approximately normally distributed, regardless of the underlying distribution.

Alternative: R functions

```
Normal                  package:stats                  R Documentation

The Normal Distribution

Description:

     Density, distribution function, quantile function and random
     generation for the normal distribution with mean equal to 'mean'
     and standard deviation equal to 'sd'.

Usage:

     dnorm(x, mean = 0, sd = 1, log = FALSE)
     pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
     qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
     rnorm(n, mean = 0, sd = 1)

Arguments:

    x, q: vector of quantiles.

       p: vector of probabilities.
```

```r
n <- 10
rn <- rnorm(n, mean=0, sd=0.5)
```

Draw 10 random numbers.

Summary

Sampling random numbers from a discrete distribution in 3 different ways

1. Draw balls from an urn (no computer needed)
2. Use the sample function in R (approximation)
3. Use special R function for the distribution (exact - if it exists)(also for continuous distr.)

7. Standard error

The standard error allows us to quantify the 'uncertainty' of the sample mean.

There are various sources for uncertainty in data

1. Finite sample size
2. Variability of covariates
3. Measurement errors

Question regarding Standard deviation

Simple example: estimate the sample mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$x_i \sim f(x;\alpha)$$

$$D = \{x_1, \ldots, x_n\}$$

Question: Is the standard deviation of x_i and $\bar{x}$ the same?

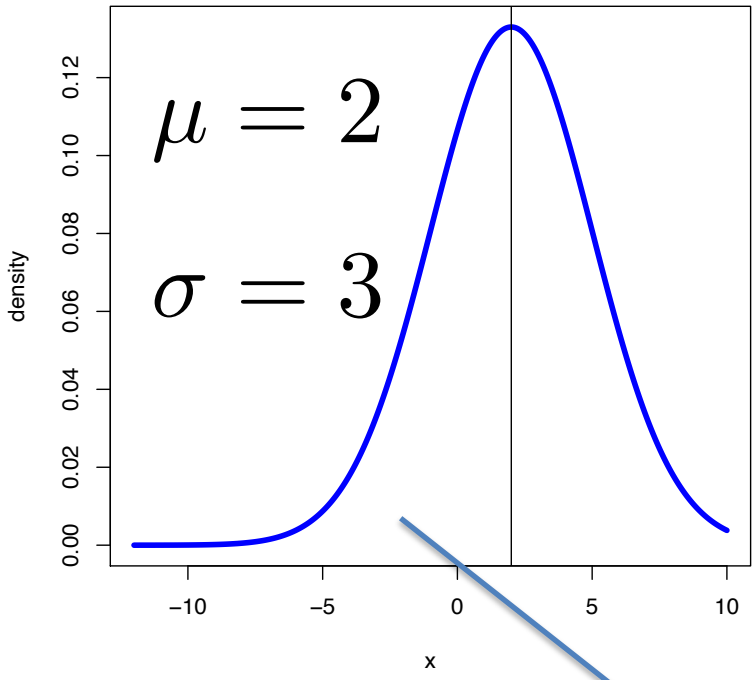
Problem

There is a difference in the standard deviation for:

1. Individual samples (individual data points): x_i

2. The sample mean: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Variability of the mean

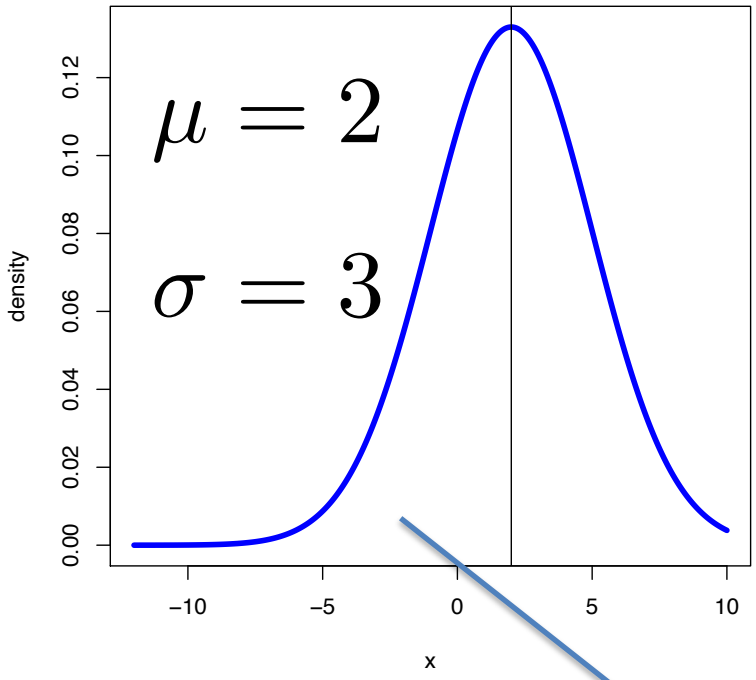


Sample mean:
m = 1.778

Sample standard deviation:
s = 3.481

Draw n = 10 samples.

Variability of the mean

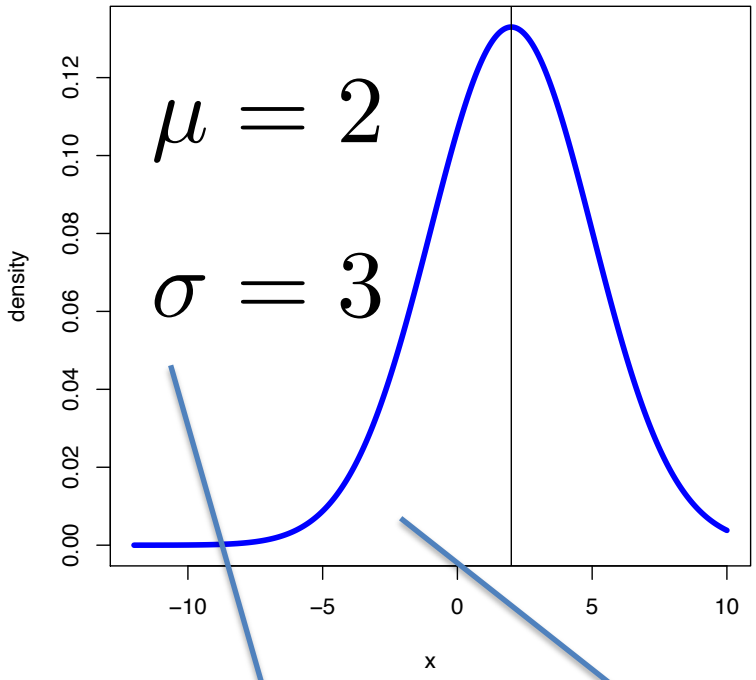


Sample mean:
m = 1.9995

Sample standard deviation:
s = 3.00073

Draw n = 10,000,000 samples.

Variability of the mean



Sample mean:
m = 1.9995

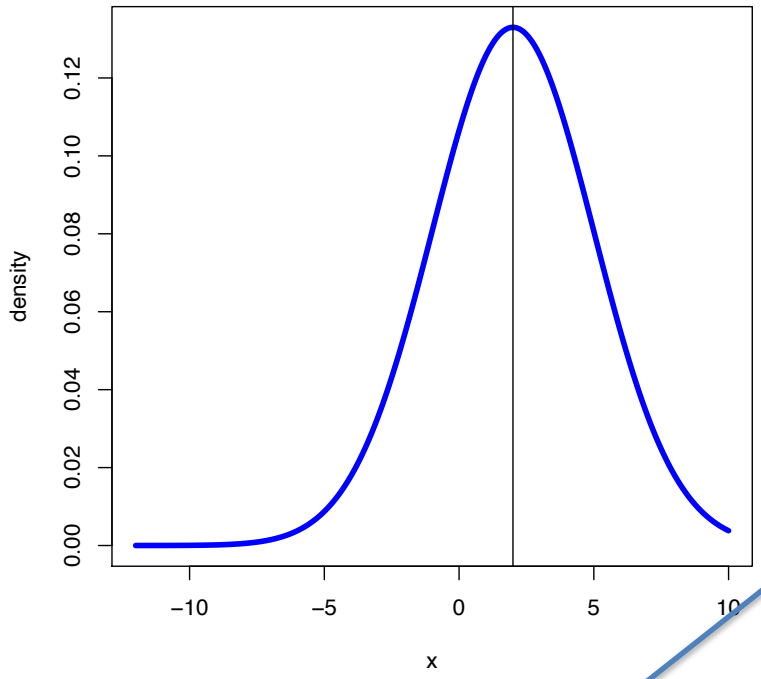
Sample standard deviation:
s = 3.00073

Not getting zero!
Q: Why?

Draw n = 10,000,000 samples.

The population standard deviation and the sample standard deviation refer to **individual** observations.

What means ‘individual’?



Probability density for x:

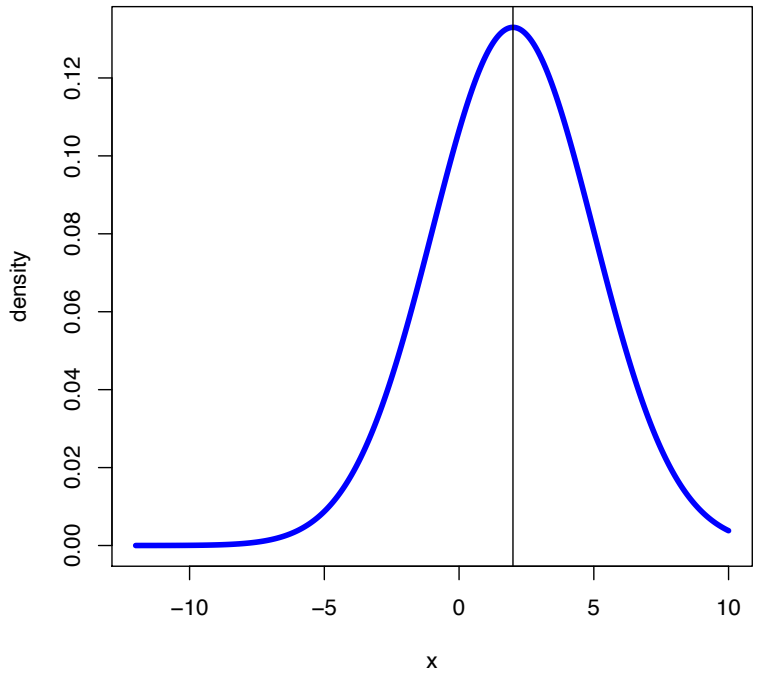
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}), \quad -\infty \le x \le \infty$$

Here x is a scalar value (one number).

X is an individual observation that is drawn from the population of all possible values.

The mean value and the standard deviation are only for individual observations. (Not for sums of these observations etc!)

What means ‘individual’?



Probability density for x:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}), \quad -\infty \le x \le \infty$$

The population mean value and the standard deviation are for individual observations only.

$$\mu = E[X] \qquad \sigma = \sqrt{E[(X-\mu)^2]}$$

However, we want to know the standard deviation of the sample mean?

Standard deviation of sample mean:

$$\sigma = \sqrt{E[(Y - \mu_Y)^2]}$$

Sample mean:

$$Y = \frac{\sum_{i=1}^{n} x_i}{n}$$

Standard deviation of individual sample:

$$\sigma = \sqrt{E[(X - \mu)^2]}$$

Mean of an individual sample:

$$\mu = E[x]$$

Visit to probability theory

Show that: $\mu_Y = \mu$

$$\mu_Y = E[Y] = E[\frac{\sum_{i=1}^{n} x_i}{n}]$$

$$= \frac{1}{n} E[\sum_{i=1}^{n} x_i] = \frac{1}{n} \sum_{i=1}^{n} E[x_i]$$

$$= \frac{1}{n} \sum_{i=1}^{n} E[x] = \frac{n}{n} E[x] = \mu$$

Remark

That means, the mean of the sample mean is the same as the mean of an individual sample.

$$\mu_{\bar{Y}} = \mu$$

Derivation (for variance)

$$\sigma_Y^2 = E[(Y - \mu_Y)^2] = E[(Y - E[Y])^2]$$

$$= E\Big[\Big(\frac{1}{n}\sum_i x_i - E[\frac{1}{n}\sum_i x_i]\Big)^2\Big]$$

$$= \frac{1}{n^2} E\Big[\Big(\sum_i x_i - E[\sum_i x_i]\Big)^2\Big]$$

$$= \frac{1}{n^2} E\Big[\Big(\sum_i x_i - \sum_i E[x]\Big)^2\Big]$$

$$= \frac{1}{n^2} E\Big[\Big(\sum_i x_i - \sum_i \mu\Big)^2\Big]$$

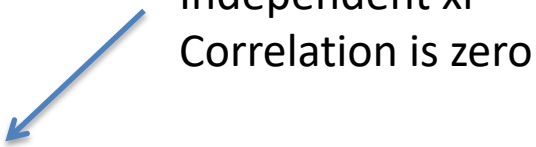
Derivation

$$= \frac{1}{n^2} E\left[\left(\sum_i (x_i - \mu)\right)^2\right]$$

$$= \frac{1}{n^2} E\left[\sum_i (x_i - \mu)^2\right]$$

$$= \frac{1}{n^2} \sum_i E[(x_i - \mu)^2]$$

$$= \frac{1}{n} E[(x - \mu)^2] = \frac{1}{n}\sigma^2$$



Derivation

Hence:

$$\sigma_Y^2 = \frac{\sigma^2}{n}$$

Variance

$$SE = \frac{s}{\sqrt{n}}$$

Standard deviation

Standard error

However, the standard deviation of the mean is not the same as the standard deviation of an individual sample.

The standard deviation of the mean is called the standard error of the mean:

$$SE = \frac{s}{\sqrt{n}}$$

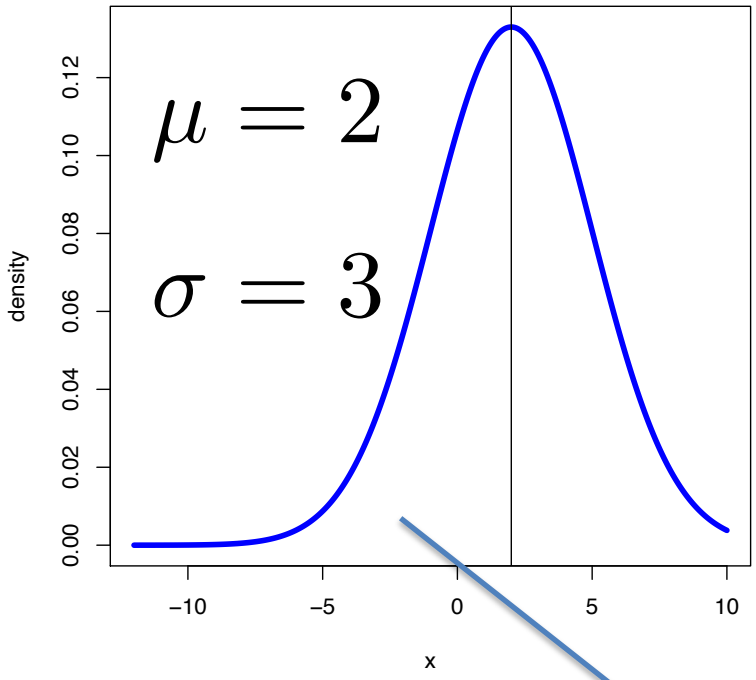
Sample estimator

n=1: SE=s
n infinity: SE=0

s: sample standard deviation

n: sample size

Variability of the mean



Sample mean:
m = 1.9995

Sample standard deviation:
s = 3.00073

Standard error of the mean:
SE = 0.0009 (for n=10, SE=1.10)

Draw n = 10,000,000 samples.

The population standard deviation and the sample standard deviation refer to individual observations.

Remark

- When one asks about the ‘variability’ (standard deviation) of the sample mean one asks about the standard error.

Summary of week 4

- Motivation
- Experimental design & reproducible research
- Resampling methods
  - Cross validation (CV)
    - Hold-out set approach
    - Leave-one-out CV (aka Jackknife)
    - k-fold CV
    - Random resampling
  - Bootstrap
- Generalization
- Sampling from a distribution
- Standard error

Additional reading

Mach. Learn. Knowl. Extr. **2019**, *1*(1), 521-551;
https://doi.org/10.3390/make1010032

Open Access | Feature Paper | Review

Evaluation of Regression Models: Model Assessment, Model Selection and Generalization Error

Frank Emmert-Streib [1,2,*] and Matthias Dehmer [3,4,5]

Can be downloaded from:

https://www.mdpi.com/2504-4990/1/1/32