Module 2, Part 1: Statistical Learning

TMA4268 Statistical Learning V2022

Sara Martino, Department of Mathematical Sciences, NTNU

January 17, 2022

Acknowledgements

- A lot of this material stems from Mette Langaas and her TAs (especiall Julia Debik). 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 @ISL.

Introduction

Learning material for this module

- James et al (2013): An Introduction to Statistical Learning. Chapter 2 (except 2.2.3).
- Additional material (in this module page) on random variables, covariance matrix and the multivariate normal distribution (known for students who have taken TMA4267 Linear statistical models).

What will you learn?

- Statistical learning and examples thereof
- Introduce relevant notation and terminology
- Prediction accuracy vs. model interpretability
- Bias-variance trade-off
- The basics of random vectors, covariance matrix and the multivariate normal distribution.

What is statistical learning?

- *Statistical learning* is the process of learning from data. We would like to
  - *draw conclusions* about the relations between the variables (*inference*) or
  - *find a predictive function* for new observations (*prediction*).
- Want to find structures in the data that help us to learn something about the real world.
- Plays a key role in many areas of science, finance and industry.
- A fundamental ingredient in the training of a modern data scientist.

Two variable types

Quantitative variables are variables from a continuous set, they have a numerical value.

- Examples: a person's weight, a company's income, the age of a building, the temperature outside, the level of precipitation etc.

Qualitative variables are variables from a discrete set with K different classes/labels/categories.

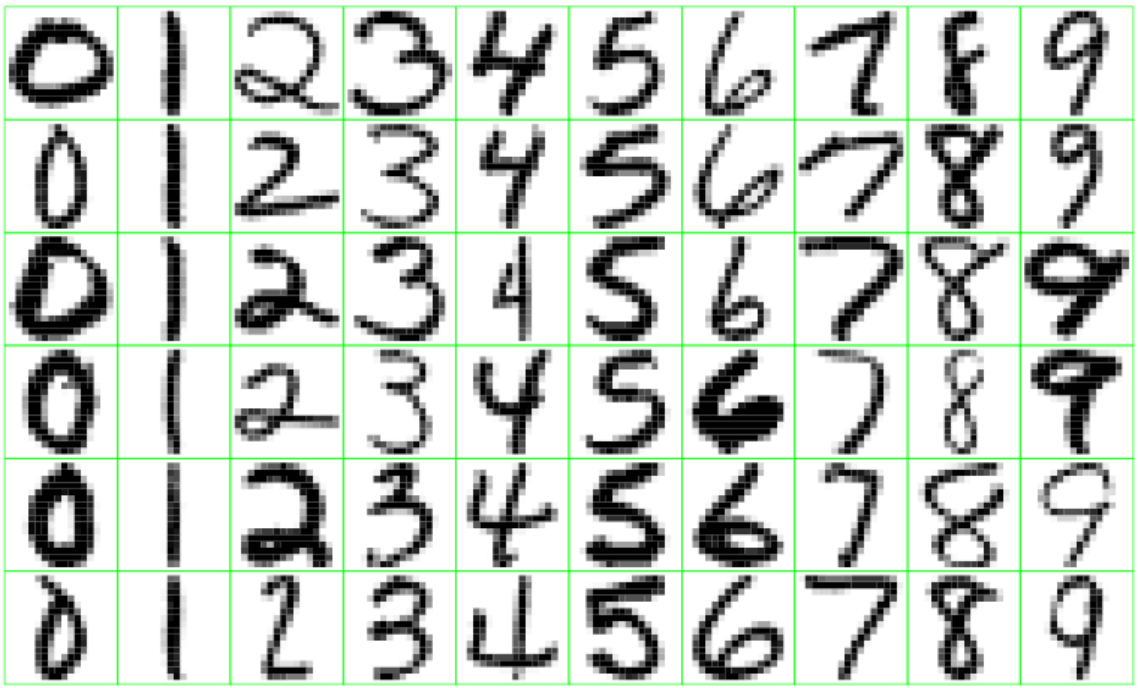
- Examples: type of fruit {apples, oranges, bananas, ...}, sex {male, female, other }, education level {none, low, medium, high}.
- Qualitative variables which have only two classes are called *binary* variables and are usually coded by 0 (no) and 1 (yes).

Examples of learning problems

- Predict the price of a stock 3 months from now, based on company performance measures and economic data. The response variable is quantitative (price). *Continuous outcome.*
- Spam detection for emails. *Binary outcome* (yes, no).
- Identification of risk factors for Prostate cancer. *Binary outcome* (yes, no).
- Estimating the risk of heart disease or heart attack, given knowledge about condition, behaviour, age, or demographic, diet and clinical measurements. *Binary outcome* (yes, no).
- Digit and image recognition. *Categorical outcome.*

Example 1: Handwritten digit recognition

- Aim: To identify the numbers in a handwritten ZIP code, from a digitized image.
- Classification problem with categorical response variable $\{0, 1, 2, \ldots, 9\}$.



Examples of handwritten digits from U.S. postal envelopes.
Image taken from https://web.stanford.edu/~hastie/ElemStatLearnII/

Example 2: Email classification (spam detection)

- Goal: to build a spam filter.
- This filter can based on the frequencies of words and characters in emails.

The table below shows the average percentage of words or characters in an email message, based on 4601 emails of which 1813 were classified as a spam.

	you	free	george	!	$	edu
not spam	1.27	0.07	1.27	0.11	0.01	0.29
spam	2.26	0.52	0.00	0.51	0.17	0.01

The Supervised Learning Problem

Starting point:

- Outcome measurement Y, also called dependent variable, response, target.
- Vector of p predictor measurements $X = (X_1, \ldots, X_p)$, also called inputs, regressors, covariates, features, independent variables.
- In the **regression problem**, Y is quantitative (e.g price, blood pressure).
- In the **classification problem**, Y takes values in a finite, unordered set (survived/died, digit 0-9, cancer class of tissue sample).
- We have training data $(x_1, y_1), \ldots, (x_N, y_N)$. These are observations (examples, instances) of these measurements.

Supervised learning and its objectives

Our data set (training set) consists of n measurement of the response variable Y and of p covariates x:

$$(y_1, x_{11}, x_{12}, \dots, x_{1p}), (y_2, x_{21}, \dots, x_{2p}), \dots, (y_n, x_{n1}, x_{n2}, \dots, x_{np}).$$

On the basis of the *training data* we would like to:

- **accurately predict** unseen test cases.
- **understand** which input affects the outcomes, and how.
- **assess the quality** of your predictions and inference.

The majority of problems studied in this course fall in the supervised learning category (exception: Module 10)

Notation and key statistical concepts

See notes and p. 9–12 in the course book.

The Unsupervised Learning Problem

- There is **no outcome variable** y, just a set of predictors (features) x_i measured on a set of samples.
- Objective is more fuzzy – find (hidden) patterns or groupings in the data - in order to *gain insight and understanding*. There is no *correct* answer.
- Difficult to know how well your are doing.

Examples in the course:

- Clustering (M10)
- Principal component analysis (M10)

Overall philosophy

- Important to understand the simpler methods first, in order to grasp the more sophisticated ones.

→ **Simpler methods often perform as well as fancier ones!**

- It is important to accurately *assess the performance of a method*, to know how well or how badly it is working.

Statistical Learning vs. Machine Learning

- Machine learning arose as a subfield of Artificial Intelligence.
- Statistical learning arose as a subfield of Statistics.
- There is much overlap – both fields focus on supervised and unsupervised problems:
  - Machine learning has a greater emphasis on large scale applications and prediction accuracy.
  - Statistical learning emphasizes models and their interpretability, and precision and uncertainty.
- The distinction has become more and more blurred, and there is a great deal of "cross-fertilization".
- Machine learning has the upper hand in Marketing!

- There is a controversy and some scepticism against “too fancy’ ’ ML methods.
- Criticism: ML often re-invents existing methods and names them differently, but often without awareness of existing methods in statistics.
- Almost weekly new literature that delivers comparison. Often, the “simple” statistical methods “win”.

A tweet by one of the co-authors of the course book, coincidentially...

%

What is the aim in statistical learning?

We are talking about supervised methods now. Assume:

- we observe one *quantitative* response Y and p different predictors $x_1, x_2, ..., x_p$.
- We assume that there is a function f that relates the response and the predictor variables:

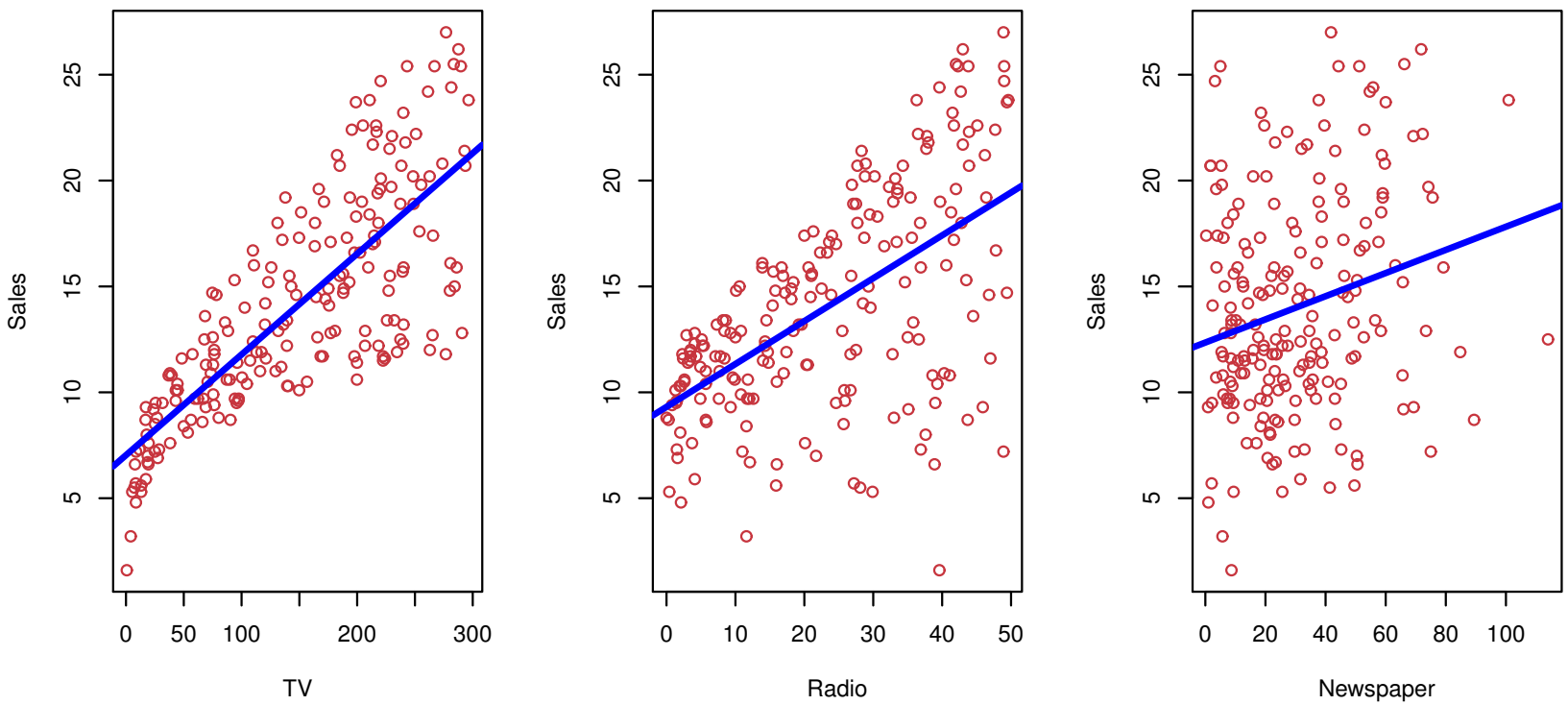
$$Y = f(x) + \varepsilon,$$

where ε is a random error term with mean 0 and independent of x.

{The aim is to estimate f.}

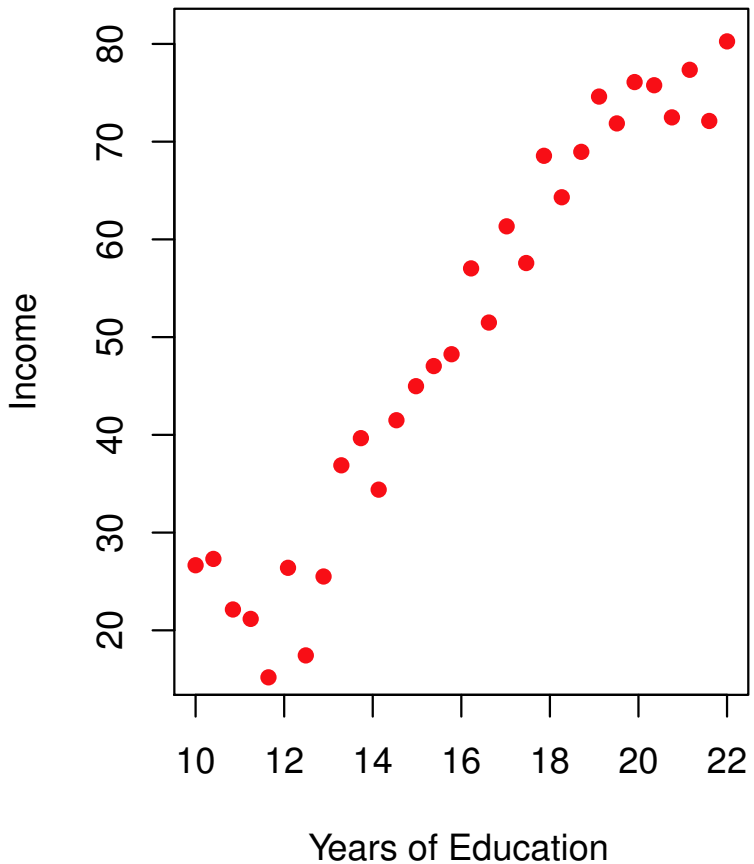
Example 1

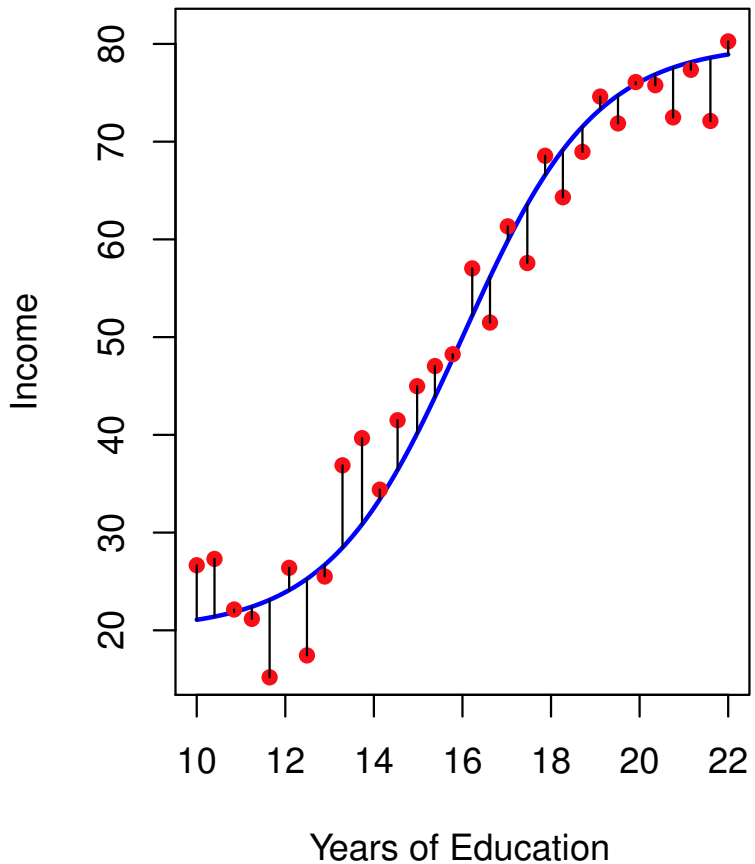
Sales of a product, given advertising budgets in different media.



Example 2

Income for given levels of education.





There are two main reasons for estimating f:

- **Prediction**
- **Inference**

Reason 1: Prediction

Aim: predict a response Y given new observations x of the covariates as accurately as possible.

Notation:

$$\hat{Y} = \hat{f}(x).$$

- $\hat{f}$: estimated f
- $\hat{Y}$ prediction for Y given x.
- We *do not really care* about the shape of f ("black box").
  $\rightarrow$ no interpretation of regression parameters when the aim is purely prediction!

There are two quantities which influence the accuracy of $\hat{Y}$ as a prediction of Y:

- The *reducible error* has to do with our estimate $\hat{f}$ of f. This error can be reduced by using the most *appropriate* statistical learning technique.
- The *irreducible error* comes from the error term ε and cannot be reduced by improving f. This is related to the unobserved quantities influencing the response and possibly the randomness of the situation.

For a given $\hat{f}$ and a set of predictors X which gives $\hat{Y} = \hat{f}(X)$, we have

$$\mathrm{E}[(Y - \hat{Y})^2] = \underbrace{(f(X) - \hat{f}(X))^2}_{reducible} + \underbrace{\mathrm{Var}(\epsilon)}_{irreducible}$$

Q: If there were a *deterministic* relationship between the response and a set of predictors, would there then be both reducible and irreducible error?

Reason 2: Inference

Aim: understand *how* the response variable is affected by the various predictors (covariates).

The *exact form* of $\hat{f}$ is of *main interest*.

- Which predictors are associated with the response?
- What is the relationship between the response and each predictor?
- Can the relationship be linear, or is a more complex model needed?

Estimating f

Overall idea:

- Using available *training data* $(x_1, y_n), \ldots (x_n, y_n)$ to estimate $\hat{f}$, such that $Y \approx \hat{f}(X)$ for any (X, Y) (also those that have not yet been observed).

Two main approaches:

- Parametric methods
- Non-parametric methods

Parametric methods

Assumption about the form or shape of the function f.
The multiple linear model (M3) is an example of a parametric method:

$$f(x) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \varepsilon \ ,$$

with $\varepsilon \sim N(0, \sigma^2)$.

The task simplifies to finding estimates of the $p + 1$ coefficients $\beta_0, \beta_1, .., \beta_p$. To do this we use the training data to fit the model, such that

$$Y \approx \hat{\beta}_0 + \hat{\beta}_1 x_1 + ... + \hat{\beta}_p x_p \ .$$

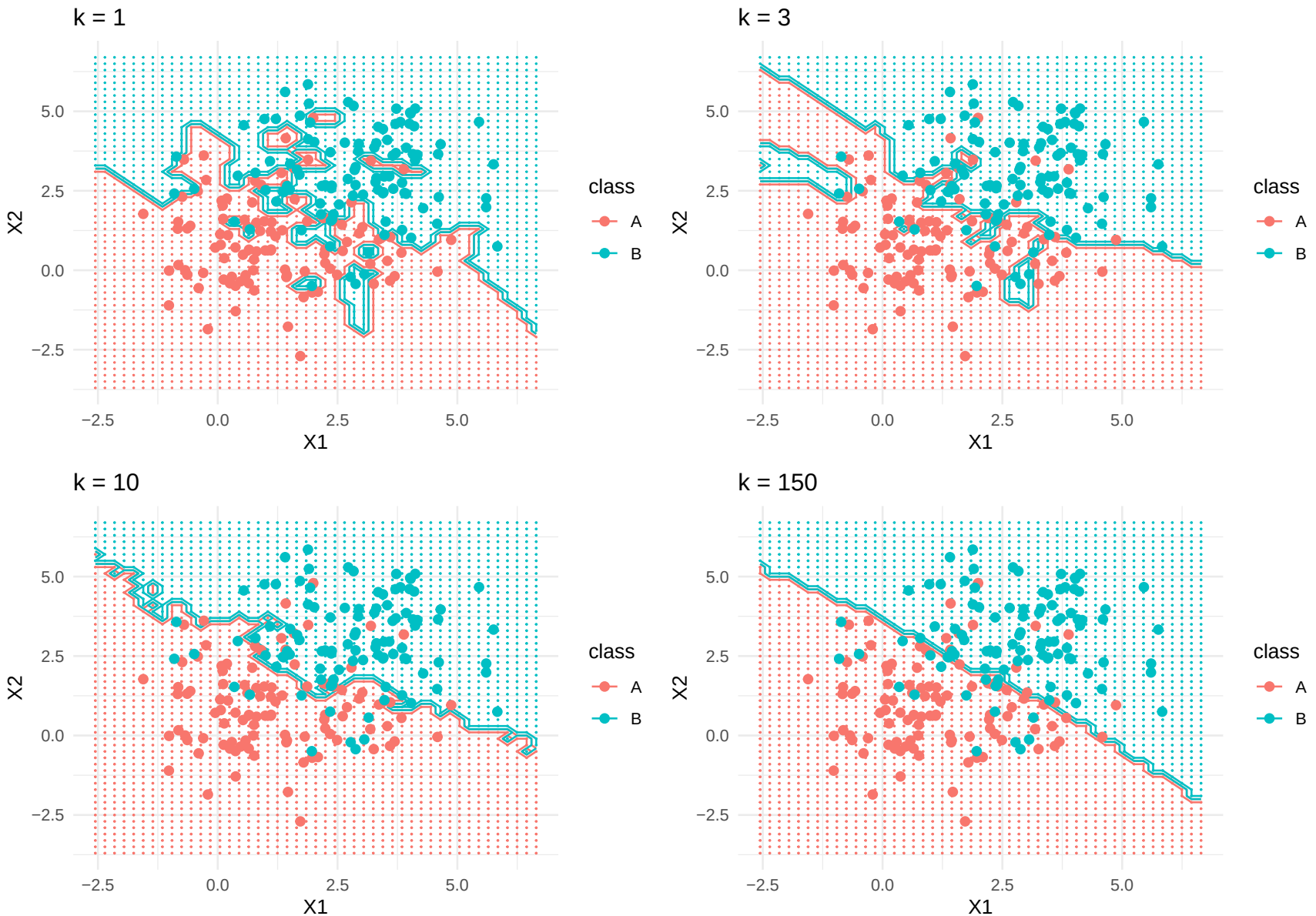
Fitting a parametric models is thus done in two steps:

1. Select a form for the function f.
2. Estimate the unknown parameters in f using the training set.

Non-parametric methods

- Non-parametric methods seek an estimate of f that gets close to the data points, but without making explicit assumptions about the form of the function f.
- Example: K-nearest neighbour (KNN) algorithm, used in classification. KNN predicts a class membership for a new observation by making a majority vote based on its K nearest neighbours. We will discuss the K-nearest neighbour algorithm in Module 4.

KNN example



Parametric methods

Advantages	Disadvantages
Simple to use and easy to understand	The function f is constrained to the specified form.
Requires little training data	The assumed function form of f will in general not match the true function, potentially giving a poor estimate.
Computationally cheap	Limited flexibility

Non-parametric methods

Advantages	Disadvantages
Flexible: a large number of functional forms can be fitted	Can overfit the data
No strong assumptions about the underlying function are made	Computationally more expensive as more parameters need to be estimated
Can often give good predictions	Much data are required to estimate (the complex) f.

Prediction accuracy vs. interpretability

(we are warming up to the bias–variance trade–off)

Inflexible methods:

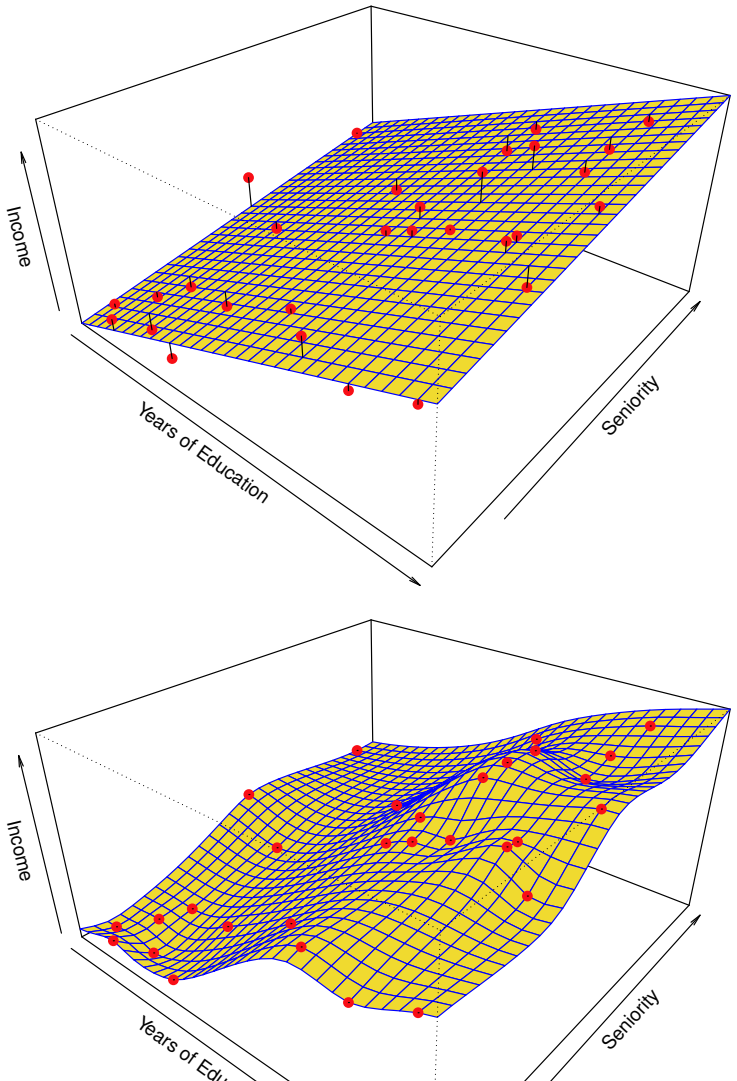
- Linear regression (M3)
- Linear discriminant analysis (M4)
- Subset selection and lasso (M6)

Flexible methods:

- KNN classification (M4), KNN regression, Smoothing splines (M7)
- Bagging and boosting (M8), support vector machines (M9)
- Neural networks (M11)

Why would I ever prefer an inflexible method?

Example: Prediction of icome from "Years of Education" and "Seniority"



Potential problems:

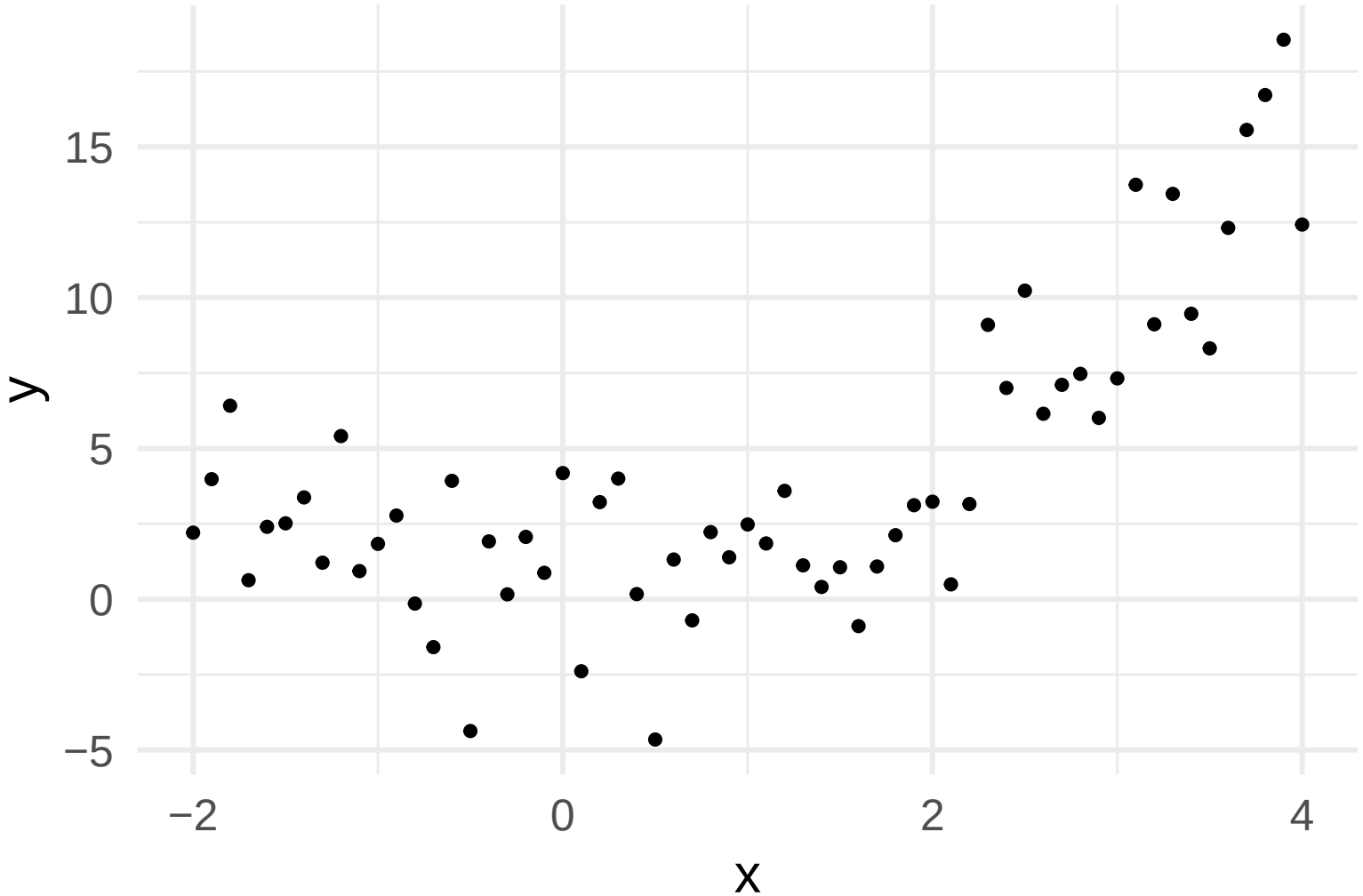
Overfitting occurs when the estimated function f is too closely fit to the observed data points.

Underfitting occurs when the estimated function f is too rigid to capture the underlying structure of the data.

We illustrate this by a toy example using polynomial regression.

Polynomial regression example (simulation)

Consider a covariate x observed between $x = -2, \ldots, 4$ and $n = 61$ observations.

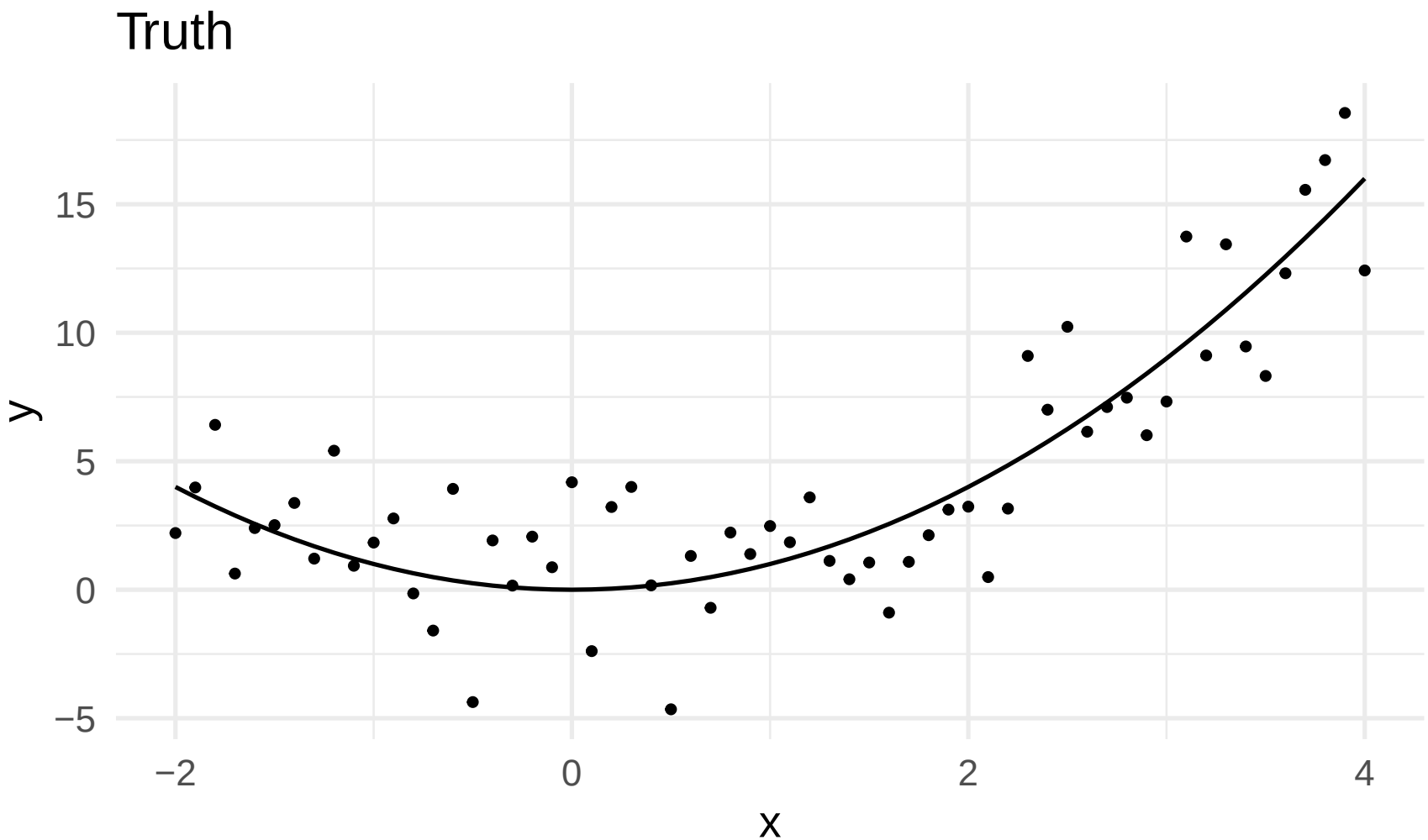


We impose a relationship between reponse Y and covariate x

$$Y = x^2 + \varepsilon$$

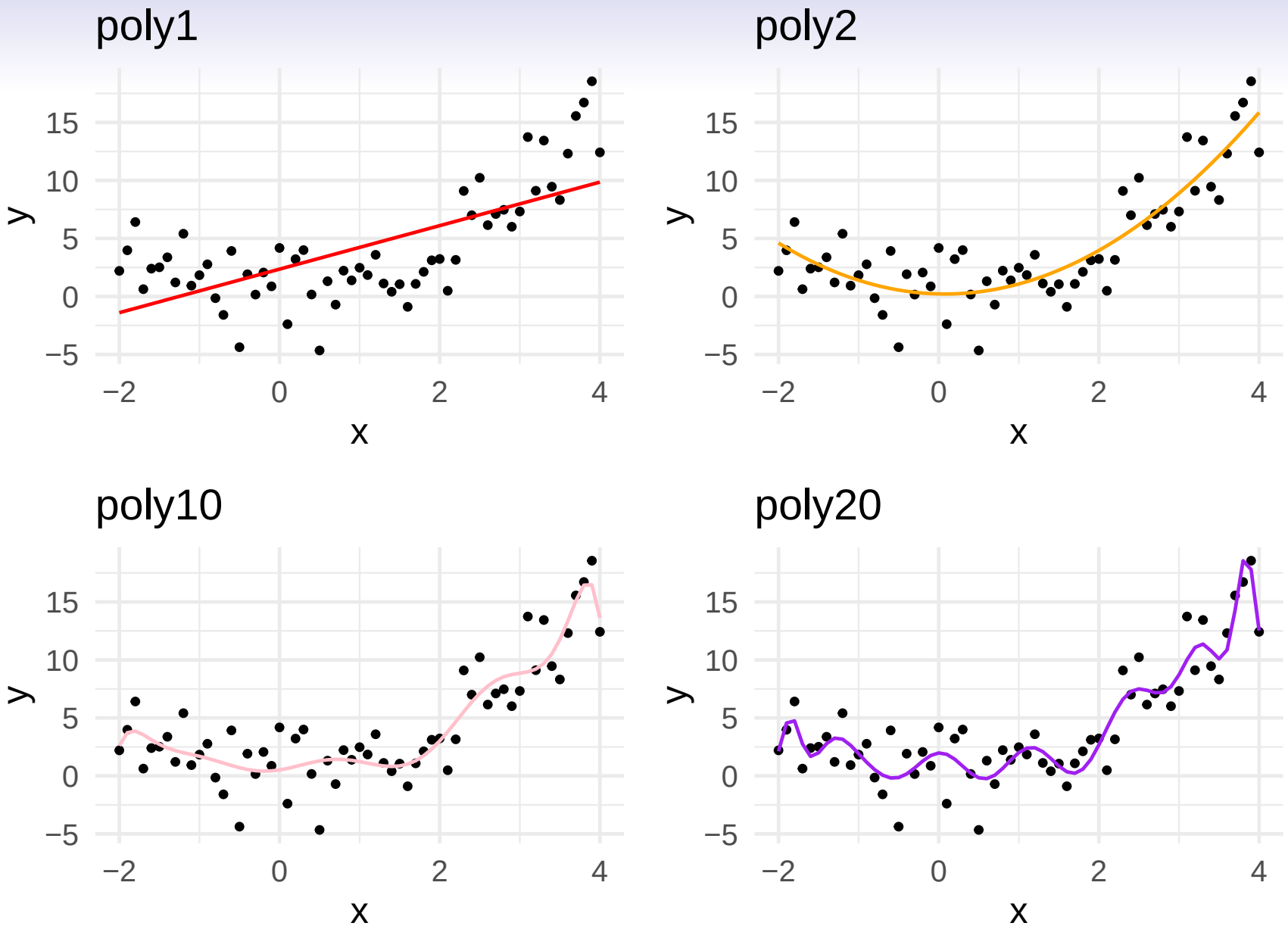
with error (noise) term $\varepsilon \sim N(0, \sigma^2)$ with $\sigma = 2$. It is a substitue for all the unobserved variables that are not in our equation, but that might influence Y.

- We call $Y = x^2$ the *truth*.
- ε is the *irreducible error*.


Truth
15
10
5
0
−5
y
−2
0
2
4
x

Try to fit a function to the observations *without* knowing the true relationship:

- **poly1**: Simple linear model of the form $\beta_0 + \beta_1 x$ fitted to the observations. *Underfits* the data.
- **poly2**: Quadratic polynomial fit to the data, of the form $\beta_0 + \beta_1 x + \beta_2 x^2$. This fits well.
- **poly10**: Polynomial of degree 10 fit of the form $\beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_{10} x^{10}$ *Overfits* the data.
- **poly20**: Polynomial of degree 10 fit of the form $\beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_{20} x^{20}$ *Overfits* the data.



The degree of the polynomial is a *flexibility parameter*.

We can now ask:

- Which of these models performs "best"?
- Is there *one* method that dominates all others?

Assessing model accuracy

No method dominates all others over all possible data sets.

- That is why we need to learn about many different methods.
- For a given data set we need to know how to decide which method produces the *best* results.
- We need to understand what *best* means.
- How close is the predicted response to the true response value?

Measuring the Quality of Fit

The quality of fit can be measures as the *Training MSE* (mean squared error), using the data that were used to estimate f:

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

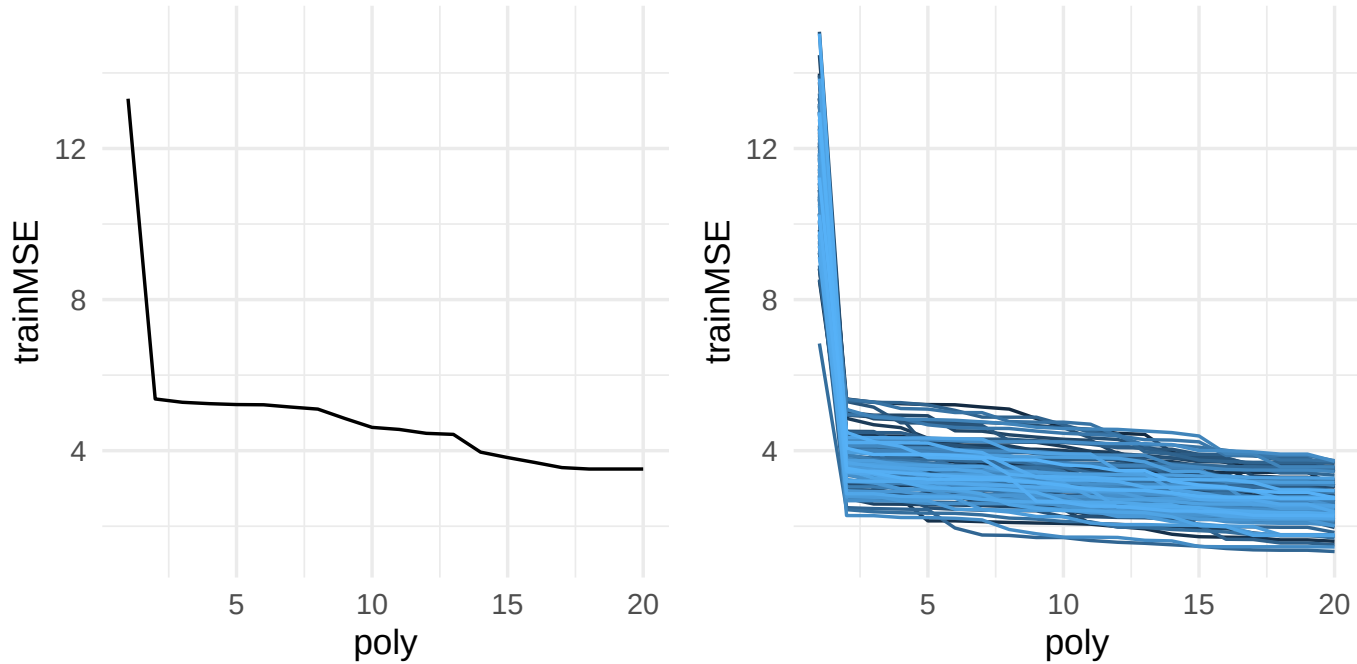
Why is the training MSE not the real measure of interest?

Why is the training MSE not the real measure of interest?

Examples:

- We don't want to predict last weeks stock price, we want to predict the stock price next week.
- We don't want to predict if a patient in the training data has diabetes (because we already know this), we want to predict if a new patient has diabetes.

Training error for the polynomial example



Left: one repetition, right: 100 repetitions of the training set.

Q: Based on the training MSE - which model fits the data the best?

Test MSE

- Simple solution: estimate $\hat{f}$ using the training data (maybe my minimizing the training MSE), but choose the *best* model using a separate *test set*.
- *Test MSE* for a set of n_0 test observations (x_{0j}, y_{0j}):

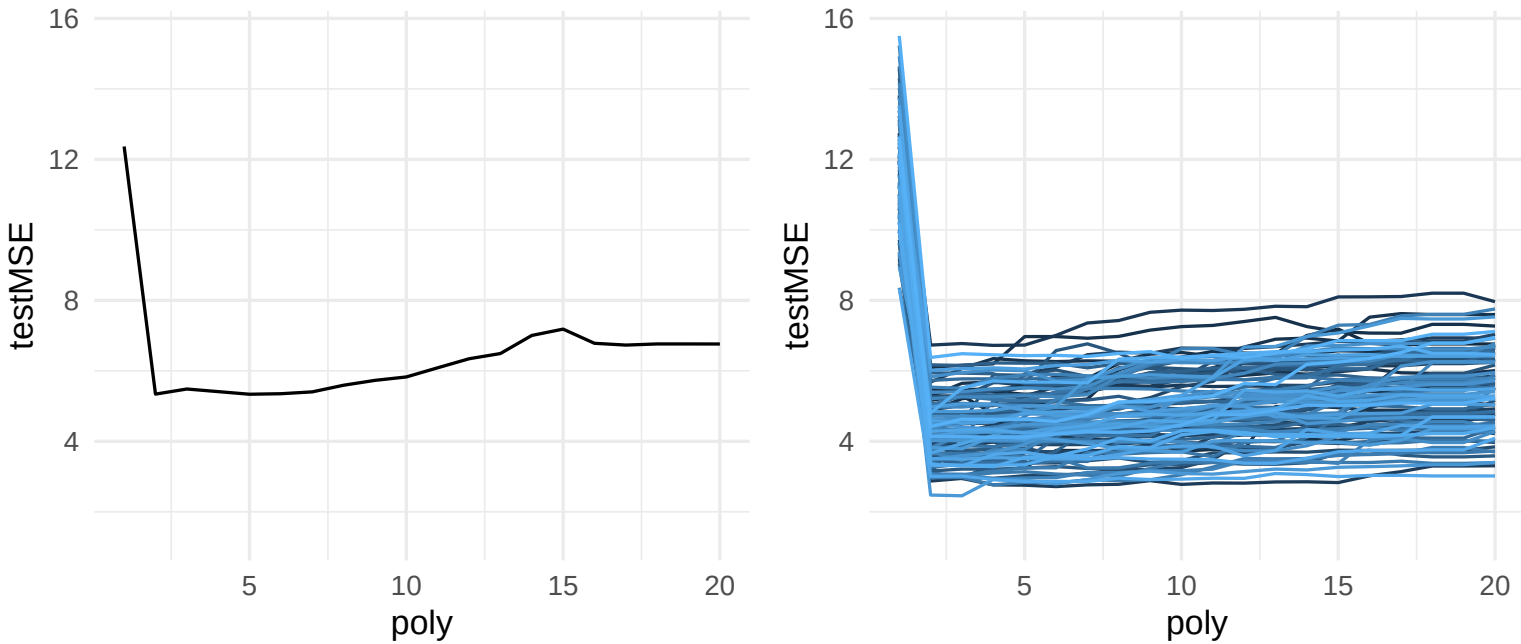
$$\text{MSE}_{\text{test}} = \frac{1}{n_0} \sum_{j=1}^{n_0} (y_{0j} - \hat{f}(x_{0j}))^2$$

- Alternative notation:

$$\text{Ave}(y_0 - \hat{f}(x_0))^2$$

(taking the average over all available test observations).

Test error for the polynomial example



Left: one repetition, right: 100 repetitions for the testMSE.

Questions:

Q1: Based on the test MSE - which model fits the data the best?

Q2: What if we do not have access to test data?

Q3: Can we instead just use the training data MSE to choose a model? A low training error should also give a low test error?

Q4: Important observation:

- The test error seems to have a minimum (U-shape) in between the extremes.
- The training error keeps going down.

Why?

The Bias-Variance trade-off

The U-shape is the result of *two competing properties* of statistical learning methods.

- Assume we have fitted a *regression* curve

$$Y = f(x) + \varepsilon$$

  to our training data $\{x_i, y_i\}$ for $i = 1, .., n$, and ε is an unobserved random variable that adds random, uncorrelated, mean-zero noise term with variance σ^2.[1]

- The training data was used to estimate $\hat{f}$.

[1] ε is a substitute for all the unobserved variables that influence Y.

- The *expected test mean squared error (MSE)* *at* x_0 (unseen test observation) is defined as:

$$\mathrm{E}[(y_0 - \hat{f}(x_0))^2] \; .$$

  This is the average test MSE that we would obtain if we repeatedly estimated f on different training sets.

- Compare this to the test MSE for the polynomical example ($\mathrm{MSE}_{\mathrm{test}}$): The average is simply replaced by the *theoretical version* (expected value).

Using that $y_0 = f(x_0) + \varepsilon$, this expected test MSE (at $X = x_0$) can be decomposed into three terms (board)

$$\mathrm{E}[(y_0 - \hat{f}(x_0))^2] =$$

...

$$= \underbrace{\mathrm{Var}(\varepsilon)}_{\text{Irreducible error}} + \underbrace{\mathrm{Var}(\hat{f}(x_0))}_{\text{Variance of prediction}} + \underbrace{\Big(f(x_0) - \mathrm{E}[\hat{f}(x_0)]\Big)^2}_{\text{Squared bias}}$$

$$\mathrm{E}[(y_0 - \hat{f}(x_0))^2] = \cdots = \mathrm{Var}(\varepsilon) + \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2$$

- *Irreducible error*. This term cannot be reduced regardless how well our statistical model fits the data.
- *Variance* of the prediction at $\hat{f}(x_0)$. Relates to the amount by which $\hat{f}(x_0)$ is expected to change for different training data. If the variance is high, there is large uncertainty associated with the prediction.
- *Squared bias*. The bias gives an estimate of how much the prediction differs from the true mean. If the bias is low the model gives a prediction which is close to the true value.

- Note:

$$\mathrm{E}[(y_0 - \hat{f}(x_0))^2]$$

  is the **expected test MSE**. We can think of this as the average test MSE we would obtain if we repeatedly estimated f using many training sets (as we did in our example), and then tested this estimate at x_0.

- However, if we also assume that X is a random variable, this is actually $\mathrm{E}[(Y - \hat{f}(x_0))^2 \mid X = x_0]$

- The **overall expected test MSE** can be computed by averaging the expected test MSE over all possible values of x_0 (averaging with respect to frequency in test set).

- Mathematically: $\mathrm{E}\{\mathrm{E}[(Y - \hat{f}(X))^2 \mid X]\}$ (by the law of total expectation / law of double expectations).

General rule

For more flexible models, the variance will *increase* and the bias will *decrease*.

This is called the *Bias-variance trade-off*.

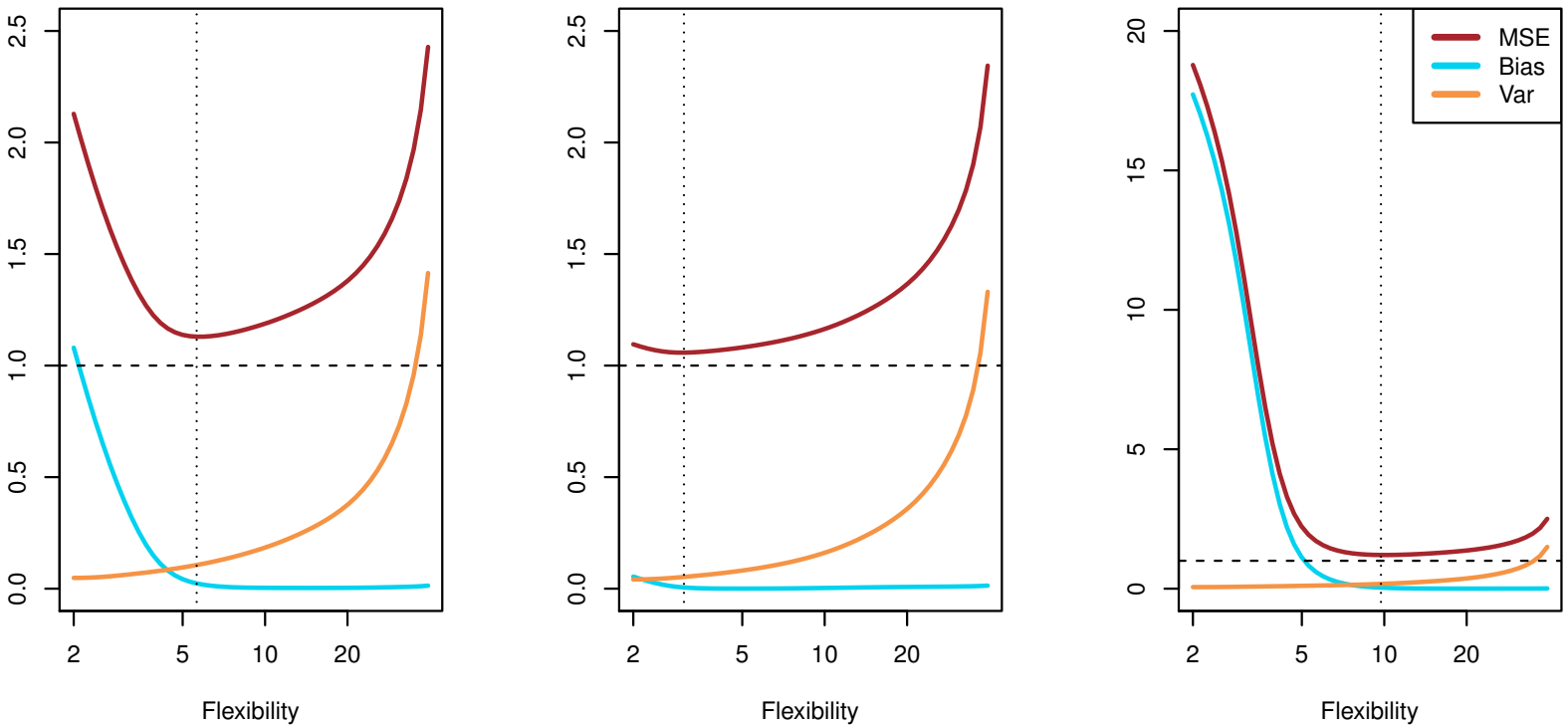
Choosing the best model

The aim is often to obtain **the most predictive model**.
Ingredients:

1. **Training set**: The observations used to fit the statistical model → Training error
2. **Test sample**: new observations which were not used when fitting the model → Test error

We have seen that

- Training error decreases for more complex/flexible models, but the test error has an **optimum**.
  → **Bias-Variance trade-off**

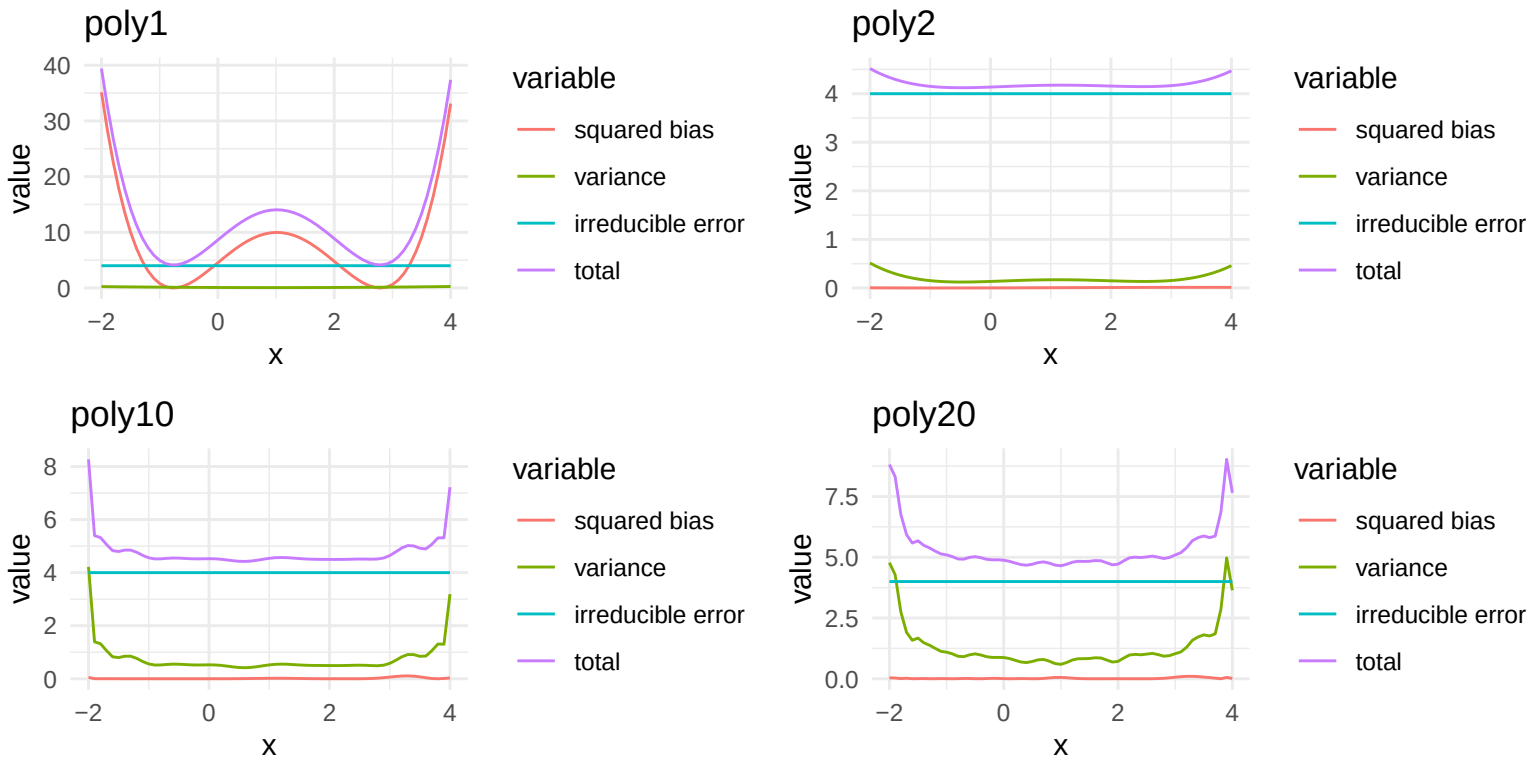


- Inflexible models: may lead to a poor fit (high bias).
- Flexible (complex) models: may overfit the data (high variance).
- Aim: find the optimum where the *test MSE is minimal*.

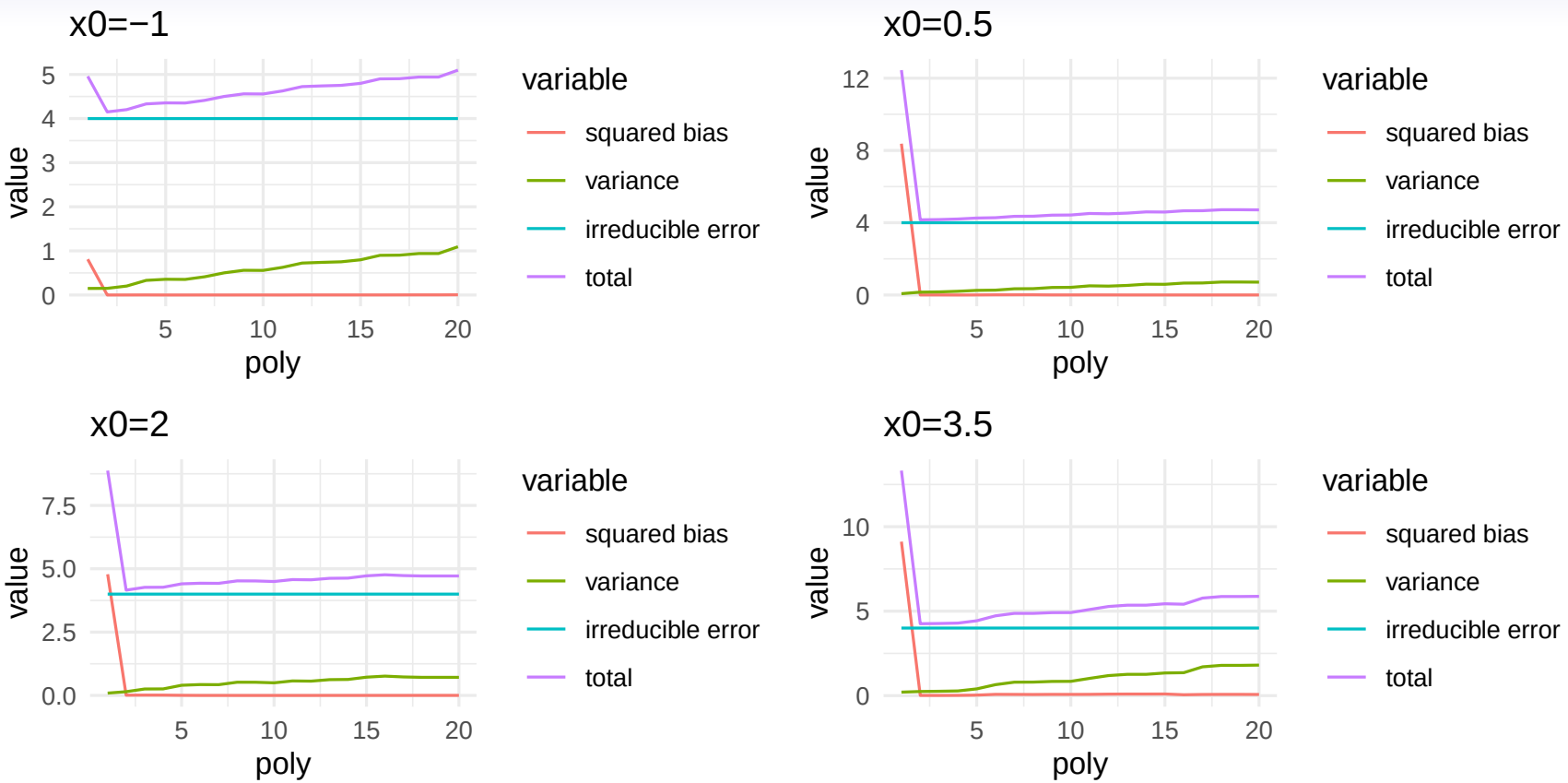
Polynomial example (cont.)

See recommended exercise 2.

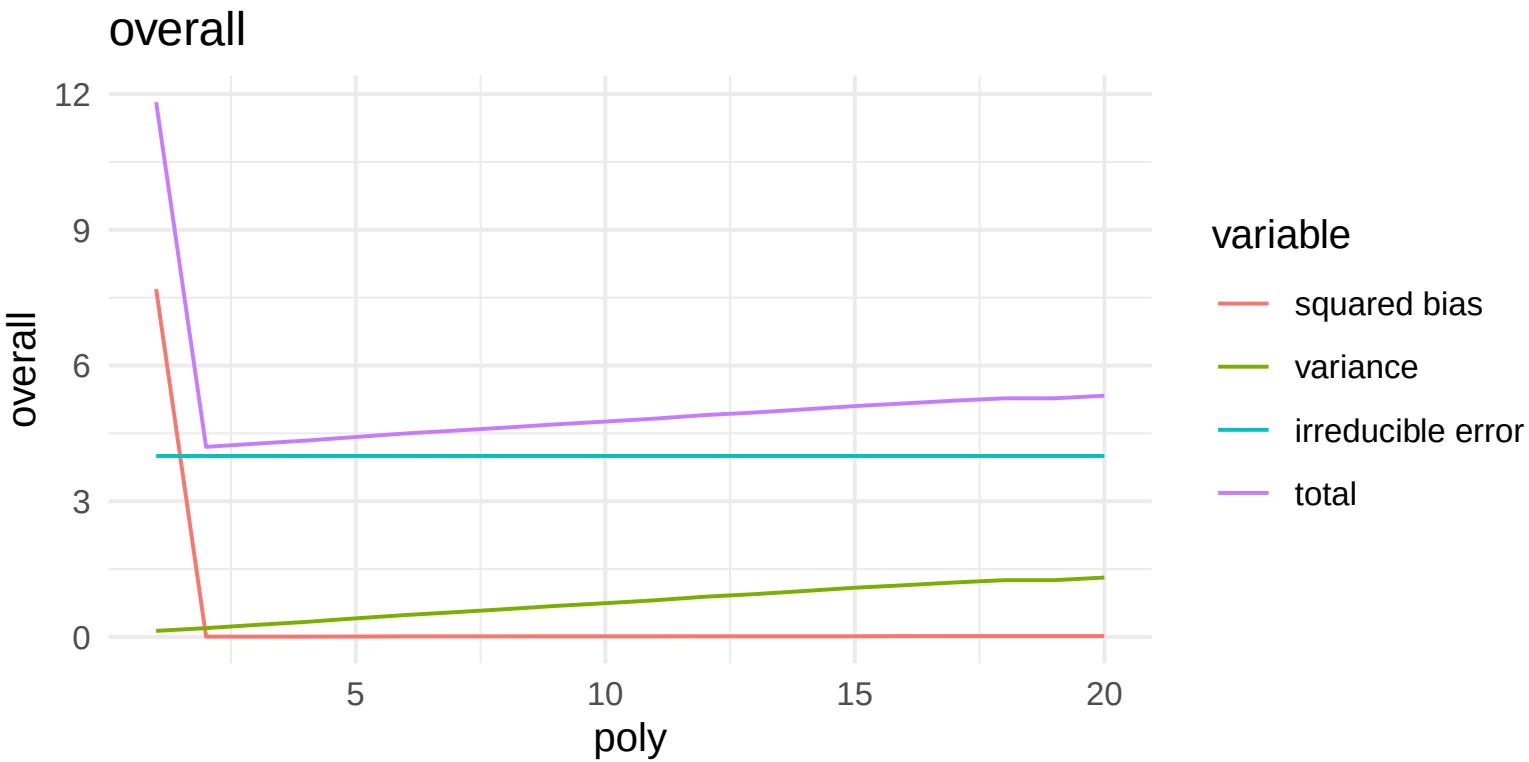
$Y = x^2$ is still the *truth*.



For four different polynomial models (poly1,2,10 and 20), the squared bias, variance, irreducible error and the total sum. Plots based on 100 simulations for the polynomial example.



At four different values for x_0, the squared bias, variance, irreducible error and the total sum. Plots based on 100 simulations for the polynomial example.



Overall version (averaging over 61 gridpoints of x).

References