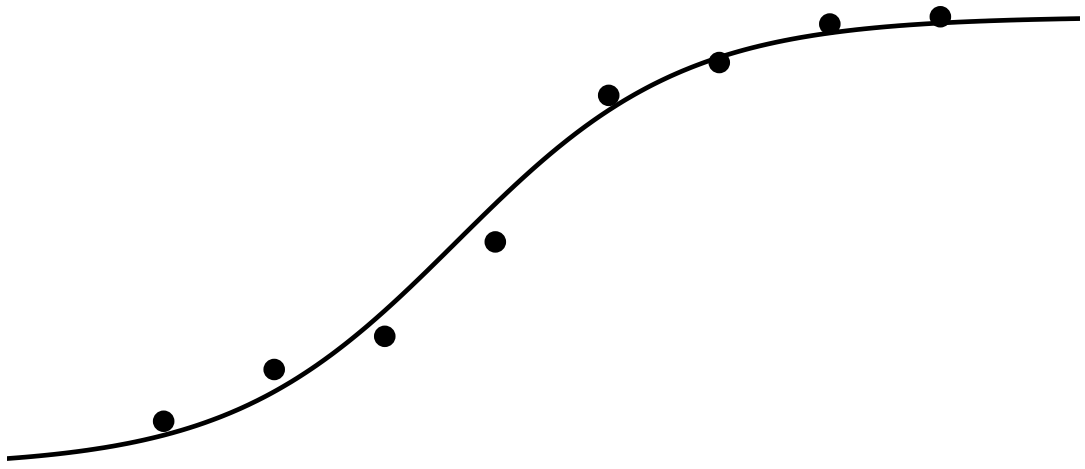
UNIVERSITÄT
BIELEFELD

Generalised Linear Models (GLMs)

312503 (V)

Roland Langrock



Organisational issues

- **Lecturer**: Roland Langrock (roland.langrock@uni-bielefeld.de)
- **Course language**: English
- **Lectures**: Thursday 12h – 14h in T2-205
  (no streaming, no recording)
- **Practicals with R**: with Carlina Feldmann, Monday 14h – 16h, V2-200
- **Exam**: most likely an oral exam in March, exact format depends on module
- **Course material & literature**:
  - slides with and without my scribblings will be in the eKVV (LernraumPlus)
  - Dobson & Barnett: *An Introduction to Generalized Linear Models*
  - Fahrmeir, Kneib, Lang & Marx: *Regression — Models, Methods and Applications*
  - (there are various other useful books and resources on GLMs!)

Contents

Chapter 1: Introduction & motivation

Regression, LMs and GLMs

- regression models are used to explain the directed relationship between a response variable[1] and covariates[2]:

$$Y = f(x_1, \ldots, x_p) + \epsilon, \quad \mathbb{E}(\epsilon) = 0$$

$$\Leftrightarrow \qquad E(Y) = f(x_1, \ldots, x_p)$$

- idea: detect the actual signal, f, whilst accounting for the noise, ϵ
- typical uses of such a regression model:
  - better understand the patterns in the data (e.g. complex economic data)
  - (statistically) test the effect of some covariate (e.g. a drug)
  - predict future outcomes based on fitted model (e.g. football scores)
  - detect outliers (e.g. in a rent index)
- linear models (LMs) constitute a special class of regression models
- generalised linear models (GLMs) generalise LMs, allowing for **many more types of data** and also more complex forms of f

[1] a.k.a. outcome or dependent variable

[2] a.k.a. predictors, independent variables, features, explanatory variables

LMs in a nutshell

- **linear regression**: the case where the signal f can be expressed as a linear combination:

$$\mathbb{E}(Y) = f(x_1, \ldots, x_p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

- **linearity** makes life easier
- assuming that Y is (conditionally) **normally dist.** further simplifies inference
- both assumptions are restrictive and for some data inadequate
- in the following, we will consider example data motivating the consideration of GLMs as more flexible regression models

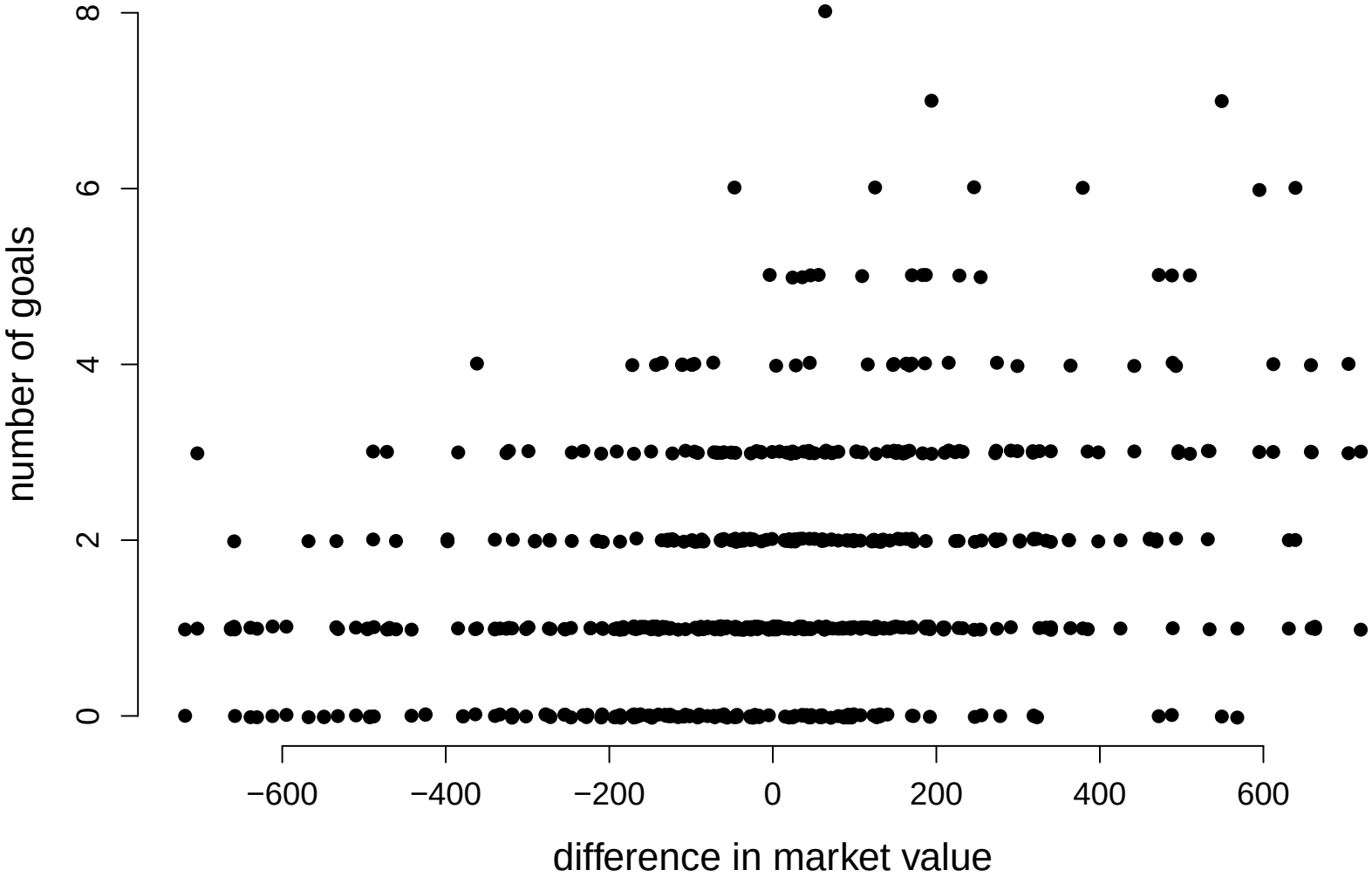
Chapter 1: Introduction & motivation

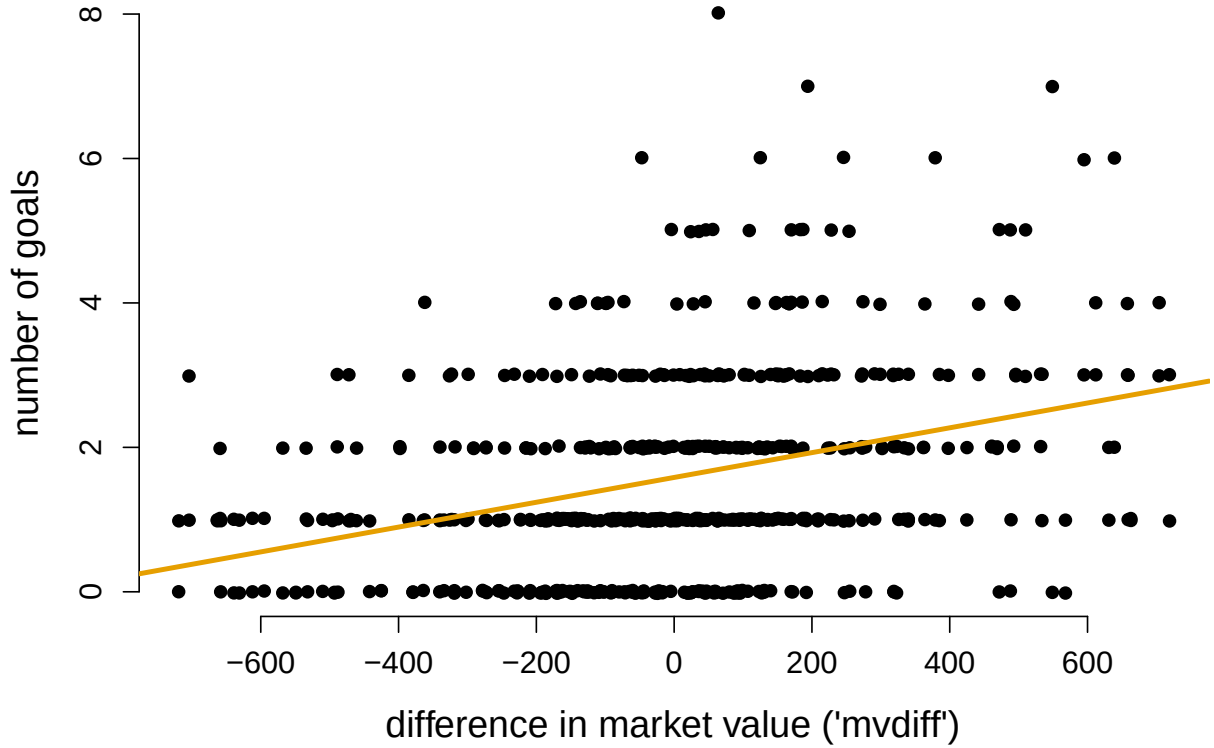
1.2 Motivating examples

Example 1: goals in Bundesliga matches (season 2018/19)

match	team	opponent	goals	diff. in market value (in mill. Euro)
1	BAY	HOF	3	496
1	HOF	BAY	1	-496
2	SCF	FRA	0	-153
2	FRA	SCF	2	153
3	FOR	AUG	1	-45
3	AUG	FOR	2	45
⋮	⋮	⋮	⋮	⋮
306	B04	FRA	6	125
306	FRA	B04	1	-125

Table: Bundesliga data from the 2018/19 season (two rows for each match!).


number of goals
0
2
4
6
8
−600
−400
−200
0
200
400
600
difference in market value



Fitted linear model:

$$\begin{aligned}\mathbb{E}(\text{goals}) &= \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{mvdiff} \\ &= 1.5833 + 0.0017 \cdot \text{mvdiff}\end{aligned}$$

Can we use this model to forecast say the number of goals Dortmund will score against Stuttgart this weekend?

Problems of the model $\mathbb{E}(\text{goals}) = \beta_0 + \beta_1 \cdot \text{mvdiff}$ when forecasting goals:

- without distributional assumption, model doesn't give us e.g. $\Pr(\text{goals} \geq 2)$
- under the usual assumption of normally distributed errors ϵ, the model would yield a continuous forecast distribution for a discrete variable:

$$\text{goals} \sim \mathcal{N}(\beta_0 + \beta_1 \cdot \text{mvdiff}, \sigma^2)$$

- we could instead assume that

$$\text{goals} \sim \mathit{Po}(\lambda), \qquad \lambda = \mathbb{E}(\text{goals}) = \beta_0 + \beta_1 \cdot \text{mvdiff},$$

  but then for score difference < -922, the model predicts a negative λ...

- (goals $\sim \mathit{Po}(\lambda)$ would also imply heteroscedasticity, such that least squares would not be optimal)

Example 2: Donner party

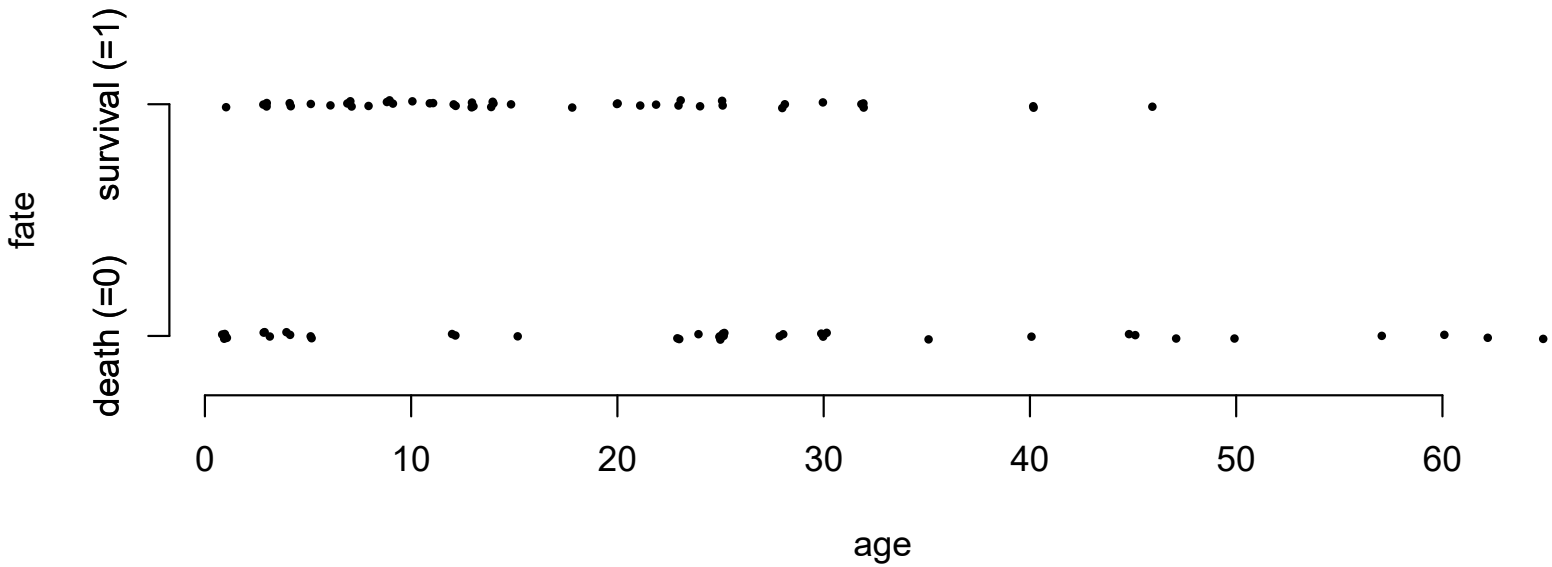
April 1846, USA: a group of 87 migrants — most notably including the Reed & Donner families — leaves Illinois for California hoping for a better life.

- trek had to cross the Rockies, which was possible only from April–Sept.
- the Donner Party got going in May (very late!)
- they took an untested route, resulting in several delays and eventually in them getting trapped in snow storms in October

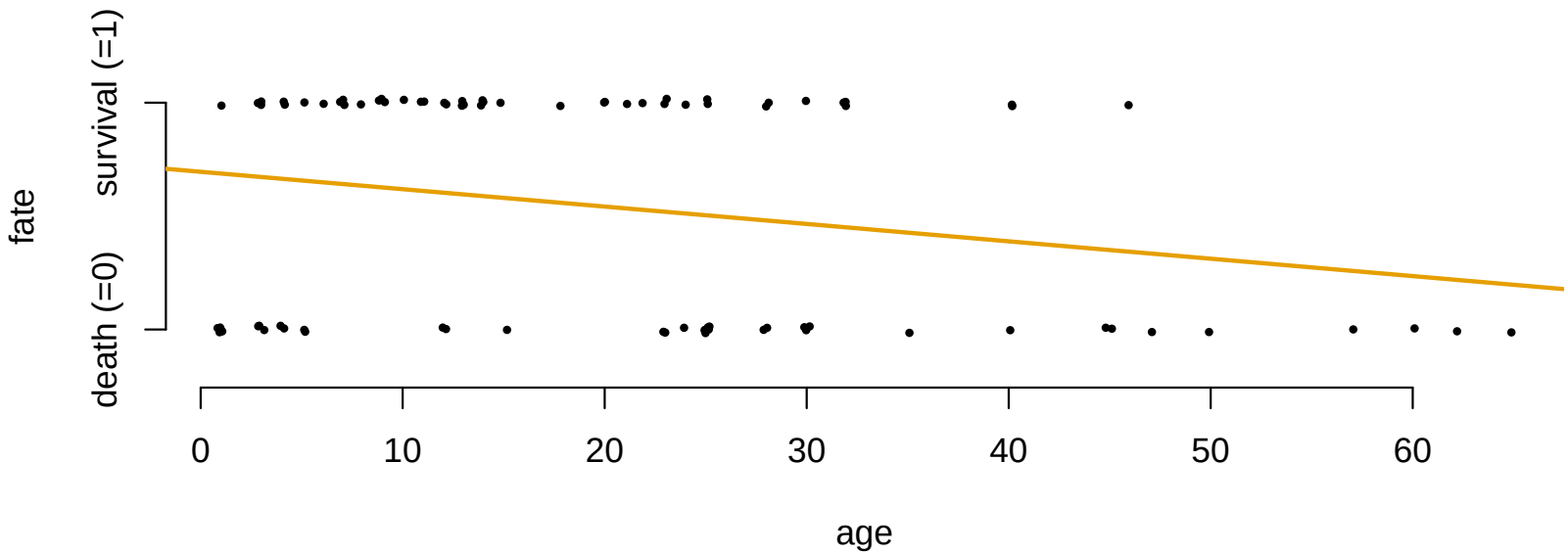
- when the last survivor was rescued ($\sim$ 6 months later), 40 of the 87 members of the Donner Party had died

name	survival	gender	age	size of the kin group
Antoine	no	male	23	1
Edward	yes	male	13	9
Isabella	yes	female	1	9
James	yes	male	4	9
Elisabeth	no	female	45	16
Margaret	no	female	1	4
Sarah	yes	female	22	12
⋮	⋮	⋮	⋮	⋮
Ada	no	female	3	4

Source: Grayson (1990)



- there seems to be some correlation between age and mortality
- not shown here: there is also correlation between
  - size of the group of kin & mortality;
  - gender and mortality (56% of the men died, but only 30% of the women)



Fitted linear model:

$$\begin{aligned}\mathbb{E}(\text{fate}) &= \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{age} \\ &= 0.6952 - 0.0077 \cdot \text{age}\end{aligned}$$

Does this model make any sense?

Obvious problems with the linear model $\mathbb{E}(\text{fate}) = \beta_0 + \beta_1 \cdot \text{age}$:

- what does say $\mathbb{E}(\text{fate}) = 0.6952 - 0.0077 \cdot 20 \approx 0.5412$ even mean?
- under the normal assumption, the model would yield a continuous forecast distribution for a binary variable — doesn't make any sense
- we could instead assume

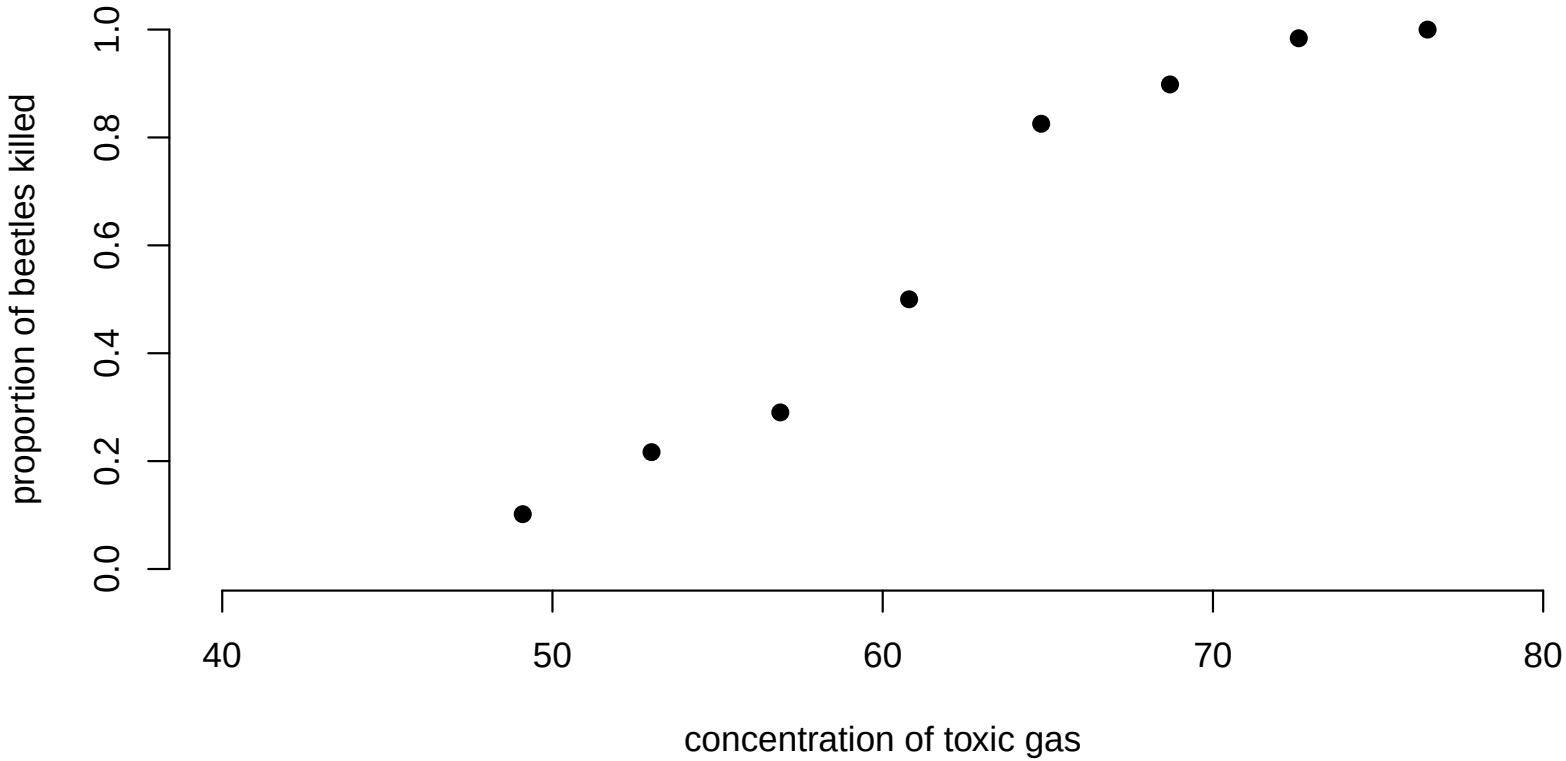
$$\text{fate} \sim \text{Bern}(\pi), \qquad \pi = \mathbb{E}(\text{fate}) = \beta_0 + \beta_1 \cdot \text{age},$$

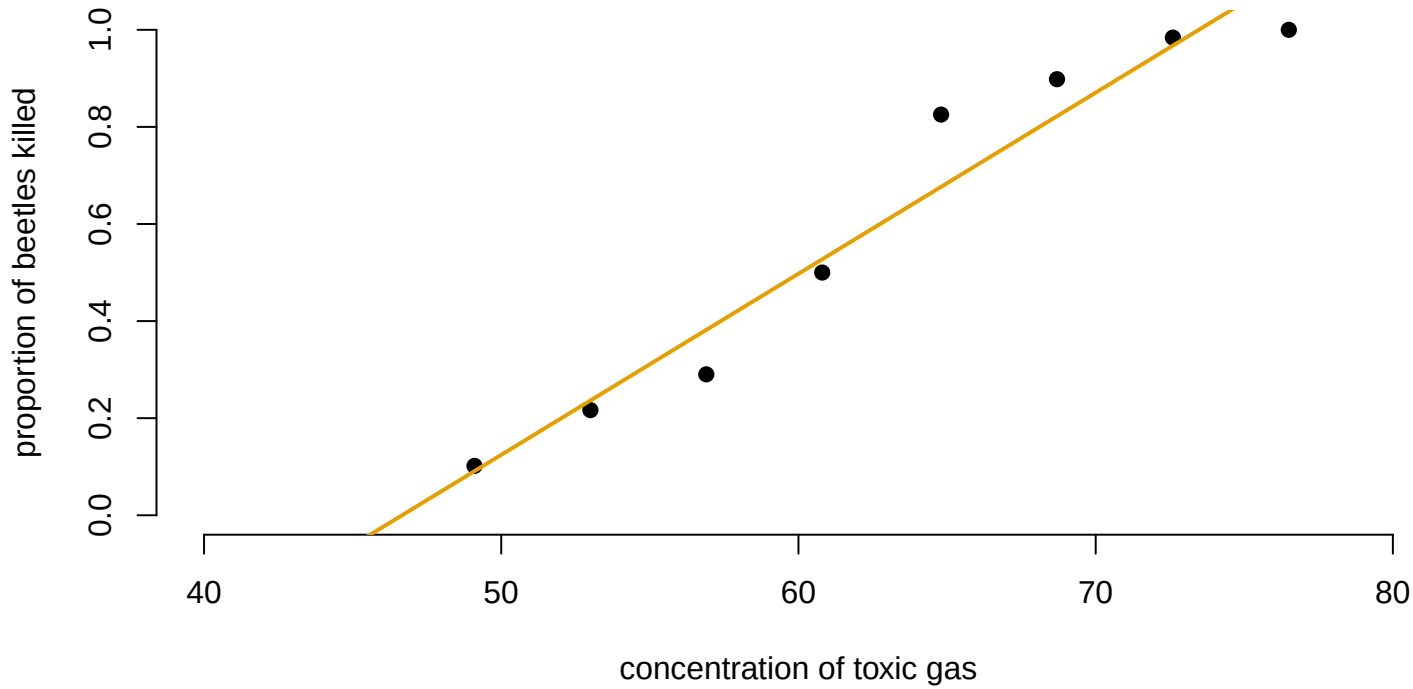
  but then for age > 90, the model predicts a negative survival probability...
- (fate $\sim \text{Bern}(\pi)$ would also imply heteroscedasticity...)

Example 3: beetles

Toxicological study by Bliss (Annals of Applied Biology, 1935):

	n_i, the number of beetles	y_i, the number of beetles killed	y_i/n_i, the proportion of beetles killed	x_i, the concentration of toxic gas
$i = 1$	59	6	0.10	49.1
$i = 2$	60	13	0.22	53.0
$i = 3$	62	18	0.29	56.9
$i = 4$	56	28	0.50	60.8
$i = 5$	63	52	0.83	64.8
$i = 6$	59	53	0.90	68.7
$i = 7$	62	61	0.98	72.6
$i = 8$	60	60	1.00	76.5


proportion of beetles killed
0.0
0.2
0.4
0.6
0.8
1.0
40
50
60
70
80
concentration of toxic gas



Fitted linear model:

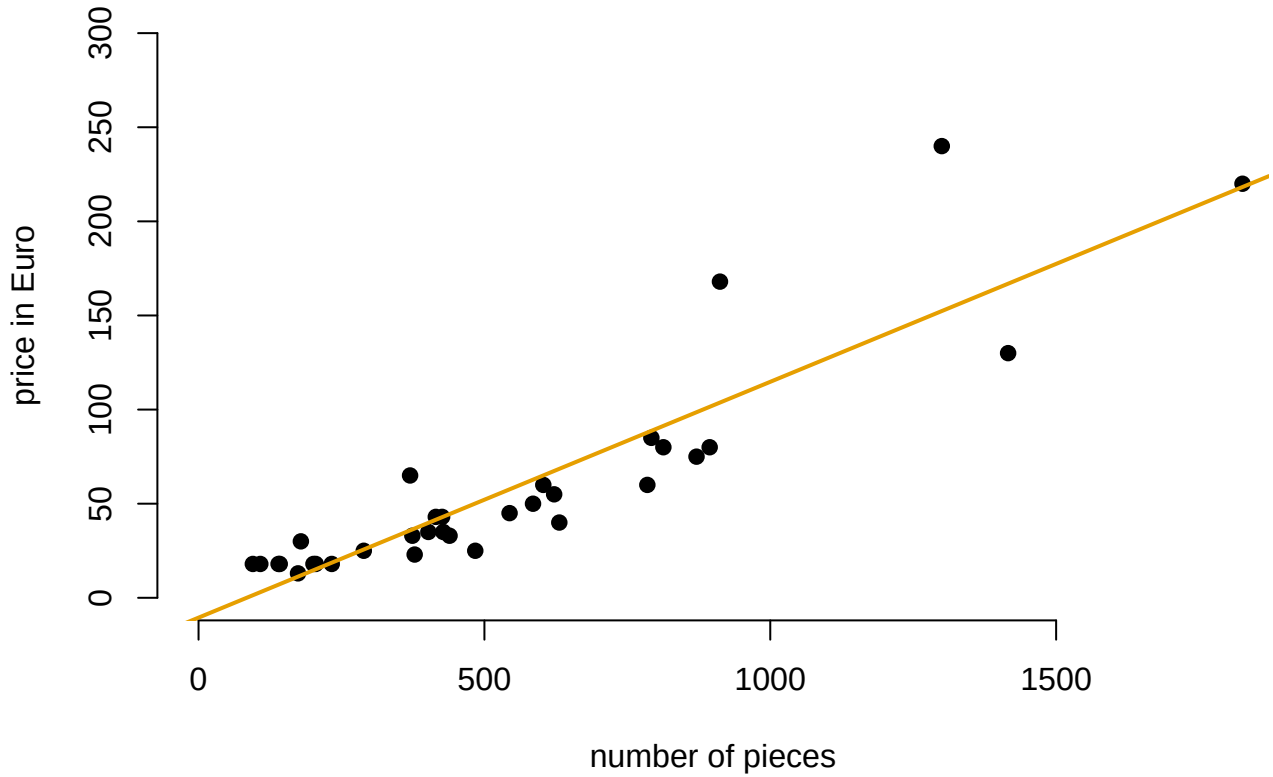
$$\begin{aligned}\mathbb{E}(\text{proportion killed}) &= \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{concentration} \\ &= -1.7406 + 0.0373 \cdot \text{concentration}\end{aligned}$$

Obvious problems with the linear model $\mathbb{E}(\text{prop. killed}) = \beta_0 + \beta_1 \cdot \text{conc.}$:

- model predicts negative proportion of beetles killed for conc. < 46.7...
- ...and proportions larger than one for conc. > 73.4

Example 4: Lego prices

Lego product	price in Euro	number of pieces
“Yellow Submarine”	50	553
“Türme aus Eis”	43	454
“Kräftemessen um Atlantis”	18	197
“Polizeiwache”	80	894
“Hüte Dich vor Vulture”	33	375
⋮	⋮	⋮
“Küstenwachzentrum”	85	792



Fitted linear model:

$$\begin{aligned}\mathbb{E}(\text{price}) &= \hat{\beta}_0 + \hat{\beta}_1 \cdot \text{pieces} \\ &= -10.5366 + 0.1253 \cdot \text{pieces}\end{aligned}$$

Problem of the linear model $\mathbb{E}(\text{price}) = \beta_0 + \beta_1 \cdot \text{pieces}$:

- for products with $<$ 85 pieces, the fitted model predicts a negative price
- this could be overcome by modelling log(price) rather than price directly
- but using GLMs we can also model price directly without the problem above

Chapter 1: Introduction & motivation

1.3 From LMs to GLMs

Outline: from LMs to GLMs

GLMs generalise LMs in two ways:

1. various different distributions can be assumed for the response variable Y

2. instead of modelling $\mathbb{E}(Y)$, a transformation $g\big(\mathbb{E}(Y)\big)$ can be modelled

Notably, GLMs constitute a **unifying framework** which includes many important models — e.g. linear, logistic & Poisson regression — as special cases.

Outline: from LMs to GLMs, historically

- most of the different types of GLMs, in particular Poisson regression and logistic regression, have been around for many decades
- but the unification of the different approaches within a single framework was accomplished only in 1972, by Nelder and Wedderburn

Generalized Linear Models

By J. A. NELDER and R. W. M. WEDDERBURN

Rothamsted Experimental Station, Harpenden, Herts

SUMMARY

The technique of iterative weighted linear regression can be used to obtain maximum likelihood estimates of the parameters with observations distributed according to some exponential family and systematic effects that can be made linear by a suitable transformation. A generalization of the analysis of variance is given for these models using log-likelihoods. These generalized linear models are illustrated by examples relating to four distributions; the Normal, Binomial (probit analysis, etc.), Poisson (contingency tables) and gamma (variance components).

"We hope that the approach [...] will prove to be a useful way of unifying what are often presented as unrelated statistical procedures, and that this unification will simplify the teaching of the subject [...]"

"We believe that the generalized linear models here developed could form a useful basis for courses in statistics"

Outline: challenges when going from LMs to GLMs

Although conceptually fairly straightforward, the extension from LMs to GLMs will bring some challenges, including the following:

- heteroscedasticity: variance of errors will, in general, vary with covariate values $\rightsquigarrow$ weighted least squares instead of ordinary least squares
- in general, there is no analytical solution for the estimation problem $\rightsquigarrow$ numerical maximisation techniques need to be used
- without normality, some inferential tools will become slightly more involved
- model checking is not as straightforward (also due to heteroscedasticity)

Chapter 1: Introduction & motivation

1.4 Some very basic probability theory concepts & notation

Random variables

A **discrete random variable** X takes finitely many or countably many values $x_1, \ldots, x_k, \ldots$.

The **probability mass function (p.m.f.)** of a discrete random variable is

$$f(x) = \begin{cases} \Pr(X = x) & \text{if } x \in \{x_1, x_2, \ldots, x_k, \ldots\} \\ 0 & \text{otherwise} \end{cases}$$

We call X a **continuous random variable** if there is an $f(x) \geq 0$, the **(probability) density function (p.d.f.)**, such that for any interval $[a, b]$:

$$\Pr(a \leq X \leq b) = \int_a^b f(x)dx$$

The **mean** of a random variable is

$$\mu = \mathbb{E}(X) = \begin{cases} \sum_{i \geq 1} x_i f(x_i) & \text{if } X \text{ is discrete} \\ \int_{-\infty}^{\infty} x f(x) dx & \text{if } X \text{ is continuous} \end{cases}$$

(the value which X in the long run takes on average)

Some calculation rules:

$$\mathbb{E}(aX + b) = a\mathbb{E}(X) + b$$

$$\mathbb{E}(g(X)) = \begin{cases} \sum_{i \geq 1} g(x_i) f(x_i) & \text{if } X \text{ is discrete} \\ \int_{-\infty}^{\infty} g(x) f(x) dx & \text{if } X \text{ is continuous} \end{cases}$$

$$\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y)$$

$$\mathbb{E}(X \cdot Y) = \mathbb{E}(X) \cdot \mathbb{E}(Y) \quad \text{if } X, Y \text{ are independent}$$

The **variance** of a random variable is

$$\sigma^2 = \mathrm{Var}(X) = \mathbb{E}(X-\mu)^2$$
$$= \begin{cases} \sum_{i\geq 1}(x_i-\mu)^2 f(x_i) & \text{if } X \text{ is discrete} \\ \int_{-\infty}^{\infty}(x-\mu)^2 f(x)dx & \text{if } X \text{ is continuous} \end{cases}$$

(a measure for how widely spread out realisations of X are around $\mu = \mathbb{E}(X)$)

Some calculation rules:

$$\mathrm{Var}(X) = \mathbb{E}(X^2) - \big(\mathbb{E}(X)\big)^2 = \mathbb{E}(X^2) - \mu^2$$
$$\mathrm{Var}(aX+b) = a^2\mathrm{Var}(X)$$
$$\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) \quad \text{if } X, Y \text{ indep.}$$

Some important distributions

distribution	symbol	p.m.f. / p.d.f. $f(x)$	support $\mathcal{T}$	$\mathbb{E}(X)$	$\mathrm{Var}(X)$
Bernoulli	$Bern(\pi)$	$\pi^x(1-\pi)^{1-x}$	$\{0,1\}$	π	$\pi(1-\pi)$
Binomial	$Bin(n,\pi)$	$\binom{n}{x}\pi^x(1-\pi)^{n-x}$	$\{0,1,\ldots,n\}$	$n\pi$	$n\pi(1-\pi)$
Poisson	$Po(\lambda)$	$\frac{\lambda^x}{x!}e^{-\lambda}$	$\{0,1,2,\ldots\}$	λ	λ
Exponential	$Exp(\lambda)$	$\lambda e^{-\lambda x}$	$(0,\infty)$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gamma	$G(\nu,\theta)$	$\frac{y^{\nu-1}\exp(-y/\theta)}{\Gamma(\nu)\theta^\nu}$	$(0,\infty)$	$\nu\theta$	$\nu\theta^2$
Normal	$\mathcal{N}(\mu,\sigma^2)$	$\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$(-\infty,\infty)$	μ	σ^2

Chapter 2: Revision of standard linear regression models

Linear regression — terminology

- **regression**: the statistical modelling of the relationship between a response variable Y and one or more covariates $x_1, \ldots, x_p$
- general formulation:

$$Y_i = f(x_{i1}, \ldots, x_{ip}) + \epsilon_i, \quad \mathbb{E}(\epsilon_i) = 0, \quad i = 1, \ldots, n$$

$$\Leftrightarrow \quad \mathbb{E}(Y_i) = f(x_{i1}, \ldots, x_{ip}), \quad i = 1, \ldots, n$$

- **linear regression**: the case where f has a linear form,

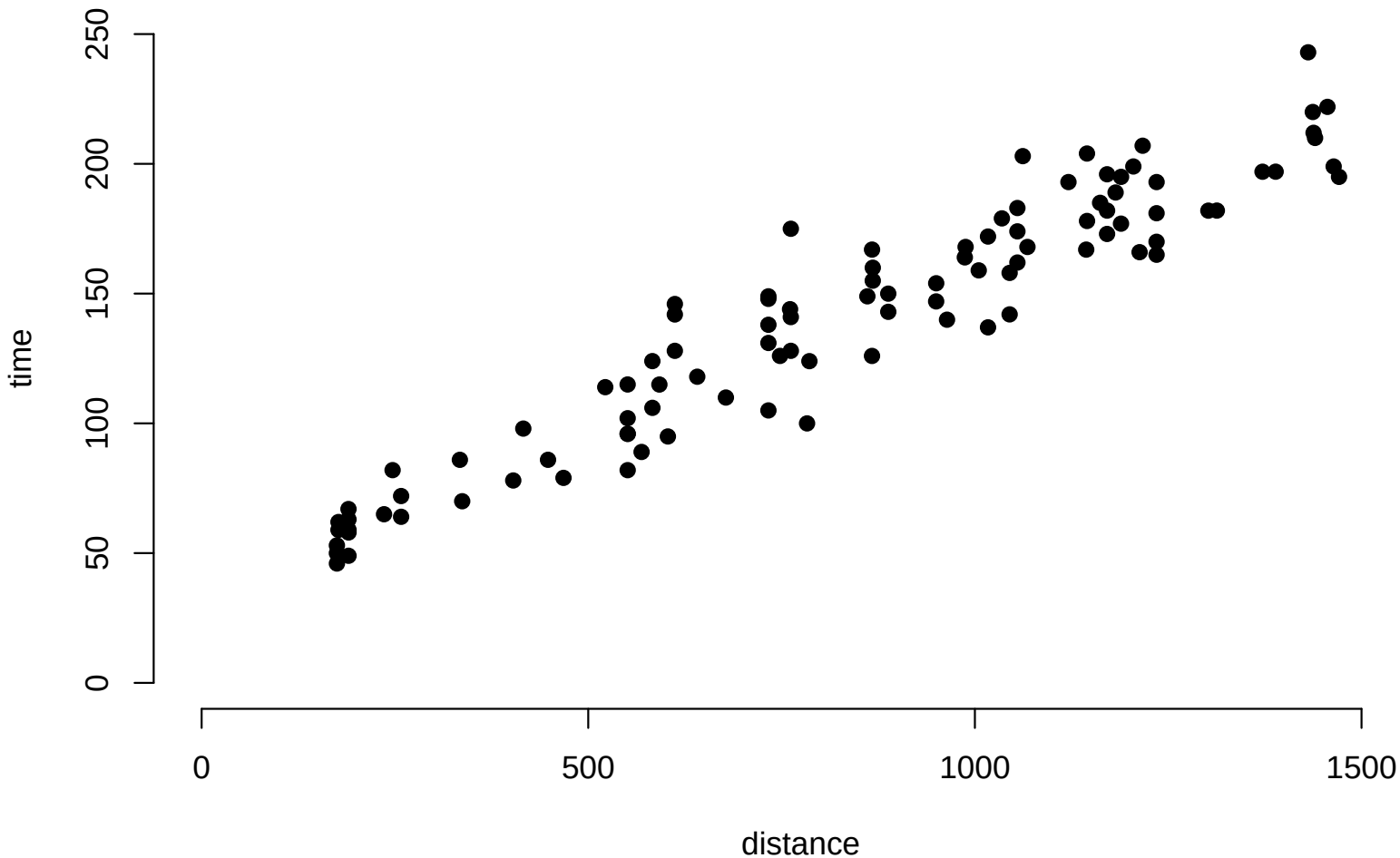
$$\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}, \quad i = 1, \ldots, n$$

- **simple linear regression**: the case $p = 1$ (a single covariate)
- **multiple linear regression**: the case $p \geq 2$ (multiple covariates)
- linearity makes our life much easier (and is often reasonable)

Example — American Airlines flights

- we consider 100 flights of American Airlines
- for each flight, we have data available on the distance flown and on the block (gate-to-gate) time:

	distance flown (in naut. miles)	block times (in minutes)
1	258	64
2	1189	195
3	1145	178
4	258	72
5	403	78
6	612	146
7	175	46
8	733	138
9	783	100
⋮	⋮	⋮
100	1438	212


time
0
50
100
150
200
250
0
500
1000
1500
distance

American Airlines flights modelled using simple linear regression

- observations are pairs $(x_{11}, y_1), (x_{21}, y_2), \ldots, (x_{n1}, y_n)$
  (x_{i1}: distance of i–th flight; y_i: block time of i–th flight; $n = 100$)
- the block times Y_i strongly depend on the distances x_{i1} to be flown
- however, even for similarly long flights there is considerable variation
- a simple linear regression model,

$$Y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i, \quad i = 1, \ldots, n,$$

  here seems reasonable (both conceptually & based on data inspection)

Rationale: find straight line that describes the relationship between x_{i1} and Y_i as well as possible, while also accounting for the noise in the process.

Simple linear regression ($p = 1$) — matrix notation

The model specification

$$Y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i, \quad i = 1, \ldots, n,$$

comprises n equations, one for each data point:

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_{11} + \epsilon_1 \\ Y_2 &= \beta_0 + \beta_1 x_{21} + \epsilon_2 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_{n1} + \epsilon_n \end{aligned}$$

Representing this equation system using **matrix notation** makes things easier!

Let's write the simple linear regression model in matrix notation:

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_{11} + \epsilon_1 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_{n1} + \epsilon_n \end{aligned} \qquad \rightsquigarrow \qquad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon,$$

where $\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}$, $\mathbf{X} = \begin{pmatrix} 1 & x_{11} \\ 1 & x_{21} \\ \vdots & \vdots \\ 1 & x_{n1} \end{pmatrix}$, $\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$, $\epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$

Multiple linear regression ($p \geq 2$) — matrix notation

Completely analogous for multiple covariates:

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_{11} + \ldots + \beta_p x_{1p} + \epsilon_1 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_{n1} + \ldots + \beta_p x_{np} + \epsilon_n \end{aligned} \qquad \rightsquigarrow \qquad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon,$$

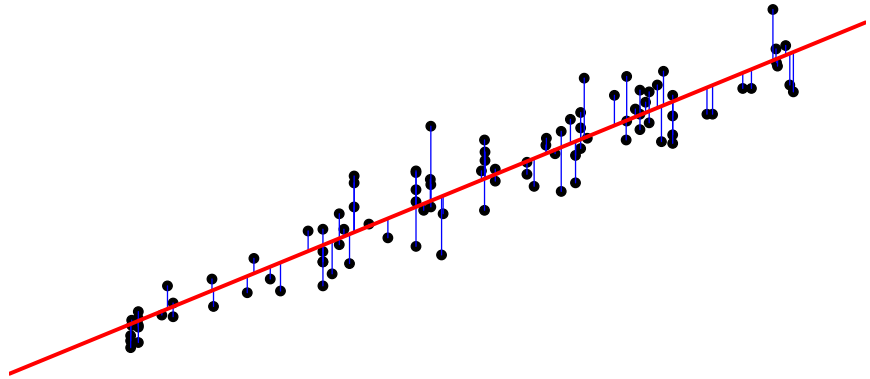
wobei $\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}$, $\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & & \\ 1 & x_{n1} & \ldots & x_{np} \end{pmatrix}$, $\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$, $\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$

We call **X** the **design matrix**.

Chapter 2: Revision of standard linear regression models

2.2 Least squares estimation & statistical inference

Estimation of the coefficients



We consider the **sum of squares** as the "distance" between model and data:

$$S(\beta_0, \beta_1, \ldots, \beta_p) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} \big(y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\big)^2$$

We then want to find the $\beta_0, \beta_1, \ldots, \beta_p$ which minimise $S(\beta_0, \beta_1, \ldots, \beta_p)$ — together these constitute the **least squares estimator** (LSE):

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p) = \underset{\beta_0, \beta_1, \ldots, \beta_p}{\operatorname{argmin}} \; S(\beta_0, \beta_1, \ldots, \beta_p)$$

(key advantage of LSE over maximum likelihood estimation: no distributional assumption needs to be made for the error terms ϵ_i)

Derivation of the LSE for $p = 1$ (in class)

LSE

For the **linear regression model**

$$Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i, \quad \mathbb{E}(\epsilon_i) = 0, \quad i = 1, \ldots, n,$$

in matrix notation

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon,$$

the LSE is the solution to the equation system

$$\mathbf{X}^t\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^t\mathbf{Y},$$

which, if $\mathbf{X}$ has full column rank, is given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{Y}.$$

Worked example in R — rent prices in Bielefeld

	monthly rent (Euro)	area (m^2)	year
1	780	67.2	2018
2	610	69.8	1958
3	687	80.2	1995
4	580	66.8	1977
5	385	52.0	1950
6	594	68.9	1957
7	350	39.0	1991
8	790	81.4	2013
⋮	⋮	⋮	
3854	650	79.1	1955

Loading the data in R:

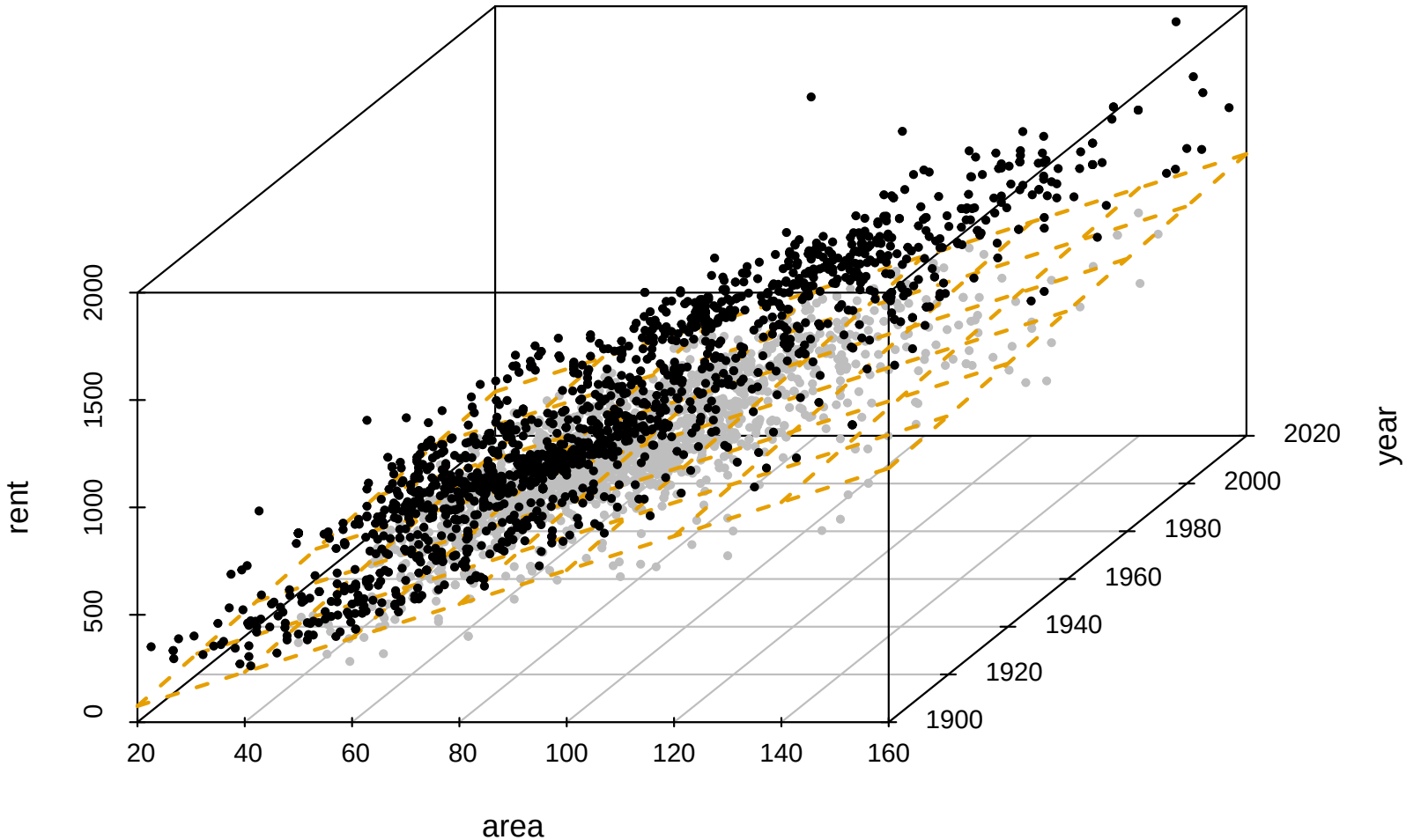
```
> rent_data<-read.csv("http://www.rolandlangrock.com//Daten//rents.csv")
> attach(rent_data)
```

Obtaining the LSE from scratch:

```
> X<-cbind(rep(1,3854),area,year)
> beta<-solve(t(X)%*%X)%*%t(X)%*%rent

> beta
     -2146.724893
area     7.906339
year     1.086172
```

Alternatively, we can simply use the `lm` function:

```
> mod<-lm(rent~area+year)
> summary(mod)
...
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.147e+03  1.459e+02  -14.71   <2e-16 ***
area         7.906e+00  9.265e-02   85.33   <2e-16 ***
year         1.086e+00  7.452e-02   14.58   <2e-16 ***
```



The fitted model:

$$\text{rent}_i = -2146.72 + 7.91 \cdot \text{area}_i + 1.09 \cdot \text{year}_i + \epsilon_i$$

Linear regression — the key assumptions

Standard assumptions:

(i) $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \ \mathbb{E}(\boldsymbol{\epsilon}) = \mathbf{0}$ (linearity)

(ii) $\text{Var}(\epsilon_i) = \sigma^2$ (homoscedasticity)

(iii) $\text{Cov}(\epsilon_i, \epsilon_j) = 0$ for all i, j (uncorrelated errors)

Possible additional assumption:

(iv) $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ (Gaussian errors)

Heteroscedasticity and weighted least squares (illustrated for $p = 1$)

For GLMs, heteroscedasticity will come into play. What does that mean?

- $\text{Var}(\epsilon_i) = \sigma_i{}^2$ (not constant!)
- the individual data points carry different amounts of information:
  - σ_i^2 small $\rightsquigarrow$ a lot of information on f and hence (β_0, β_1)
  - σ_i^2 large $\rightsquigarrow$ little information on f and hence (β_0, β_1)

It then makes intuitive sense[3] to consider the **weighted sum of squares**,

$$S(\beta_0, \beta_1) = \sum_{i=1}^{n} w_i \big(y_i - (\beta_0 + \beta_1 x_{i1})\big)^2 = \sum_{i=1}^{n} w_i \epsilon_i^2,$$

where $w_i = \frac{1}{\sigma_i^2}$, $i = 1, \ldots, n$.

[3] and in some sense ('BLUE') is also mathematically optimal

Derivation of the weighted LSE (in class)

Weighted LSE

For the linear regression model as on slide 45, but with

$$\mathrm{Var}(\epsilon_i) = \sigma_i^2,$$

the weighted LSE is given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}\mathbf{Y},$$

where $\mathbf{W} = \mathrm{diag}\big(w_1, \ldots, w_n\big)$, $w_i = \frac{1}{\sigma_i^2}$, $i = 1, \ldots, n$.

Distribution of the LSE

Reminder: an estimator such as the LSE, $\hat{\boldsymbol{\beta}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{Y}$, is a **random variable**, since it is a function of random variables (here the Y_i).

For example, for the given 100 AA flights, we obtain a specific value for $\hat{\boldsymbol{\beta}}$, but if we consider 100 other AA flights, then we'll get a different estimate.

As a random variable, $\hat{\boldsymbol{\beta}}$ follows a distribution, based on which we can construct

- confidence intervals and
- hypothesis tests

Suppose that (i)–(iii) are satisfied, and that the design matrix $\mathbf{X}$ has full column rank. If additionally (iv) is satisfied[4], then

$$\hat{\boldsymbol{\beta}} \sim \mathcal{N}_{p+1}\left(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^t\mathbf{X})^{-1}\right)$$

If σ^2 is estimated — as is basically always the case in practice — then the components of $\hat{\boldsymbol{\beta}}$ are t–distributed with $n - (p + 1)$ degrees of freedom.

[4] if (iv) is not satisfied, then the estimator is still approximately normally distributed for large n

Distribution of the LSE — illustration

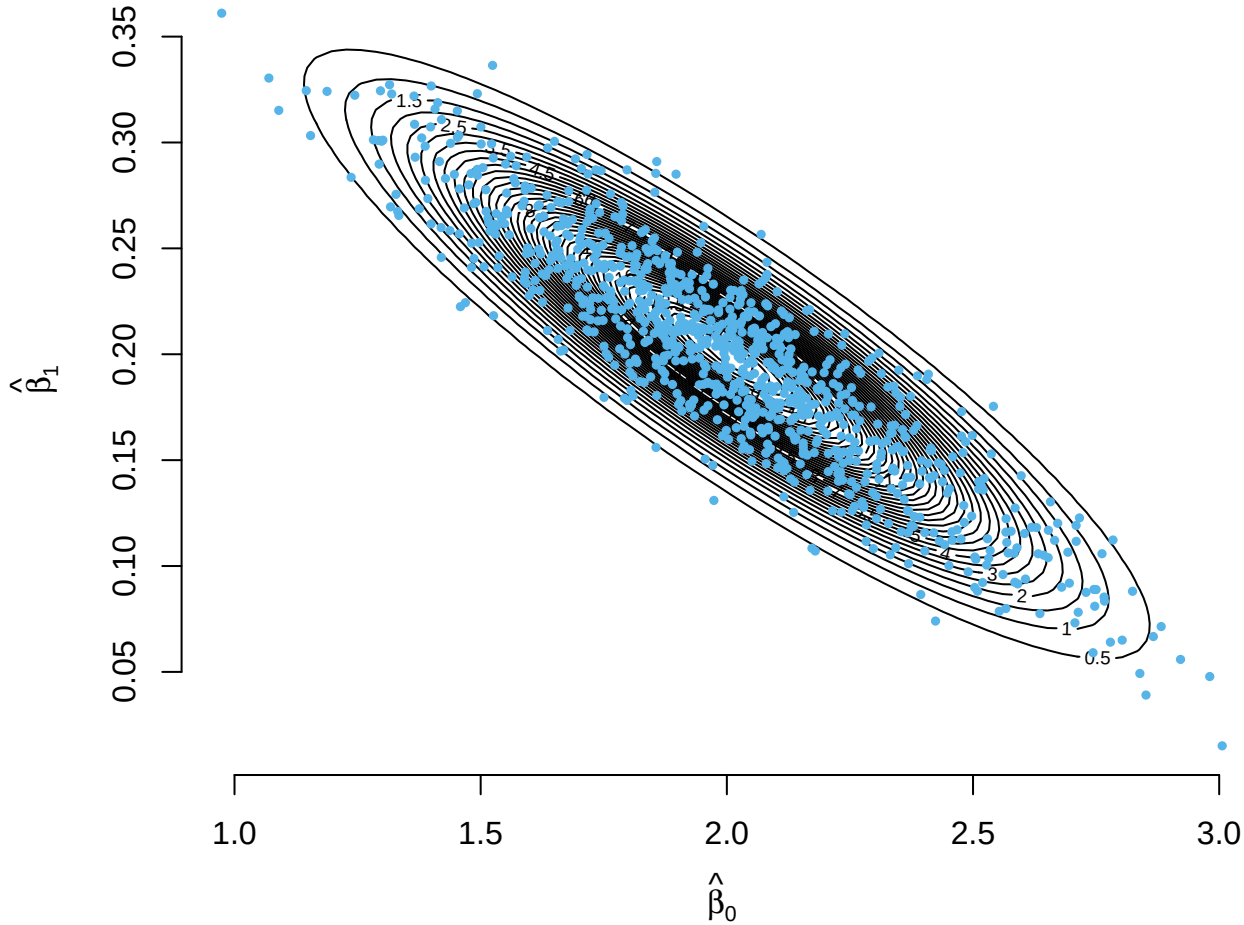


Figure: LSEs obtained in 1000 **simulation runs** (blue dots), each run with 50 data points from the model $Y_i = 2 + 0.2x_{i1} + \epsilon_i$, and theoretical distribution of the LSE (contour lines).

Testing H_0: $\beta_j = 0$

The null hypothesis H_0: $\beta_j = 0$ is rejected if

$$\left| \hat{\beta}_j / \hat{\sigma}_{\hat{\beta}_j} \right| > t_{1-\alpha/2, n-(p+1)}$$

$\rightsquigarrow$ if the test statistic lies in the (extreme) tail of the $t_{1-\alpha/2, n-(p+1)}$ distribution, then the data are very unlikely under H_0, so that we have reason to doubt H_0

$\rightsquigarrow$ we reject H_0 at level α if the probability of observing a test statistic at least as extreme as $\hat{\beta}_j / \hat{\sigma}_{\hat{\beta}_j}$, under H_0 — the so-called **p-value** — is less than α

Chapter 2: Revision of standard linear regression models

2.3 Flexible modelling using linear models

Flexible modelling with linear regression models

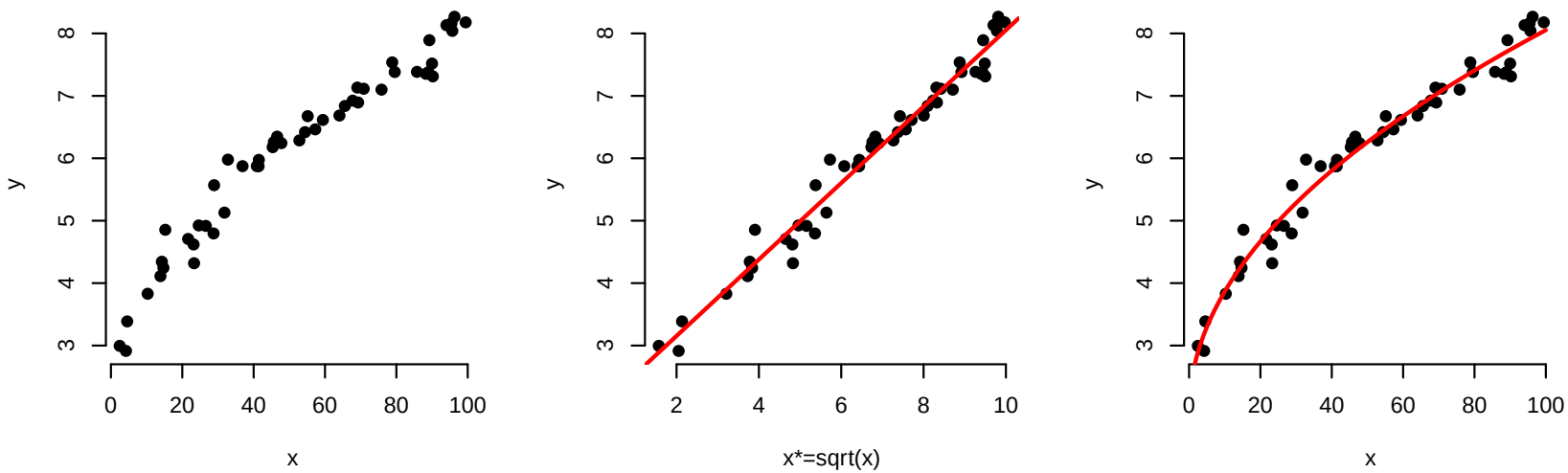
- the assumption of linearity isn't as restrictive as it may seem
- if some covariate doesn't have a linear effect on the response variable, then a **transformation may make the system linear**
- in such cases, a linear regression — using all the basic techniques & tools — is conducted using the transformed variable(s)

Variable transformation

- in some cases, the effect of x_{i1} on the response Y is nonlinear, but the relationship becomes linear after a transformation $h(x_{i1})$
- in such a case, we can simply estimate the linear model

$$Y_i = \beta_0 + \beta_1 x_{i1}^* + \epsilon_i,$$

  where $x_{i1}^* = h(x_{i1})$, using least squares
- given $\hat{\beta}$, we can re-consider the original variable, i.e. $Y_i = \hat{\beta}_0 + \hat{\beta}_1 h(x_{i1}) + \epsilon_i$



Polynomial regression

- polynomial effects can be modelled in the same manner
- for example, the model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \epsilon_i$$

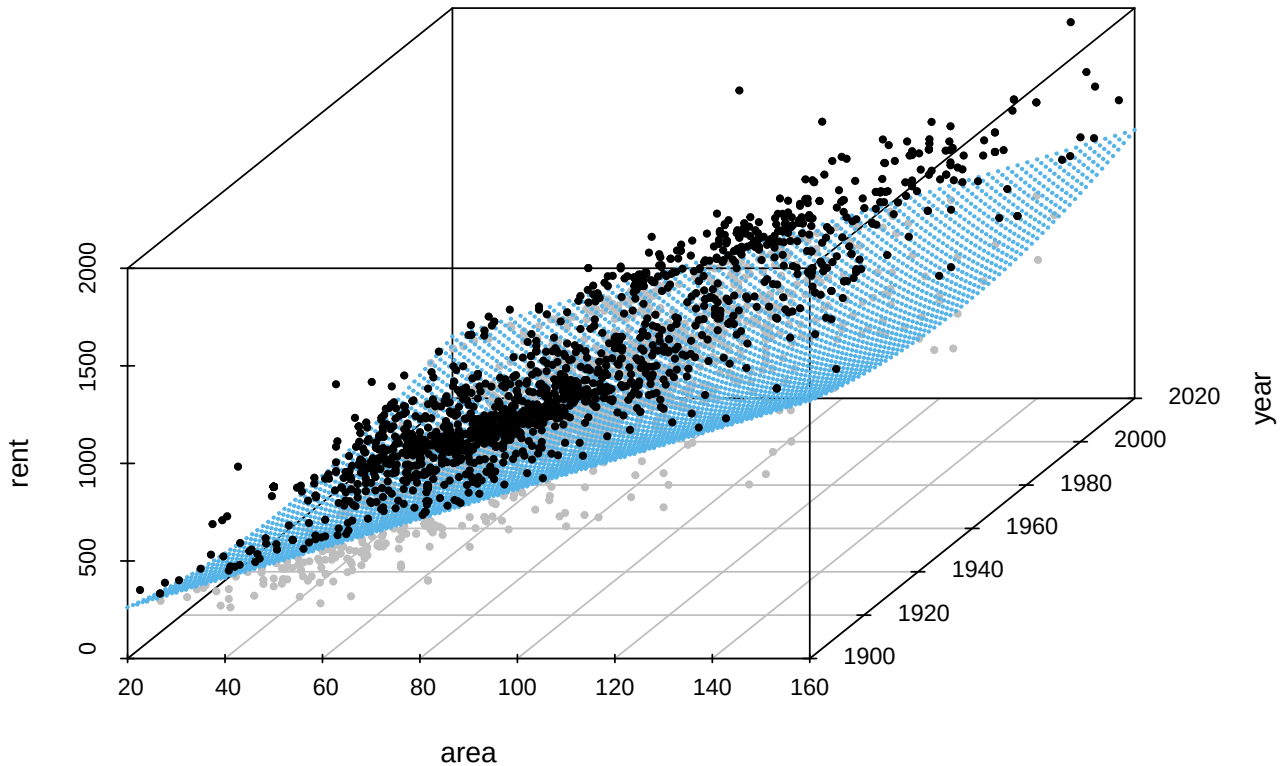
  can be formulated and fitted as the multiple linear model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i,$$

  where $x_{i2} = x_{i1}^2$
- quadratic terms are common, cubic terms sometimes seen, but higher orders are rarely considered, since the models become too unstable

Example quadratic regression — rent prices in Bielefeld

$$\text{rent}_i = 176241 + 7.563 \cdot \text{area}_i - 180.3 \cdot \text{year}_i + 0.046 \cdot \text{year}_i^2 + \epsilon_i$$



In R:

```
mod<-lm(rent~area+year+I(year^2))
```

Example polynomial regression — global warming

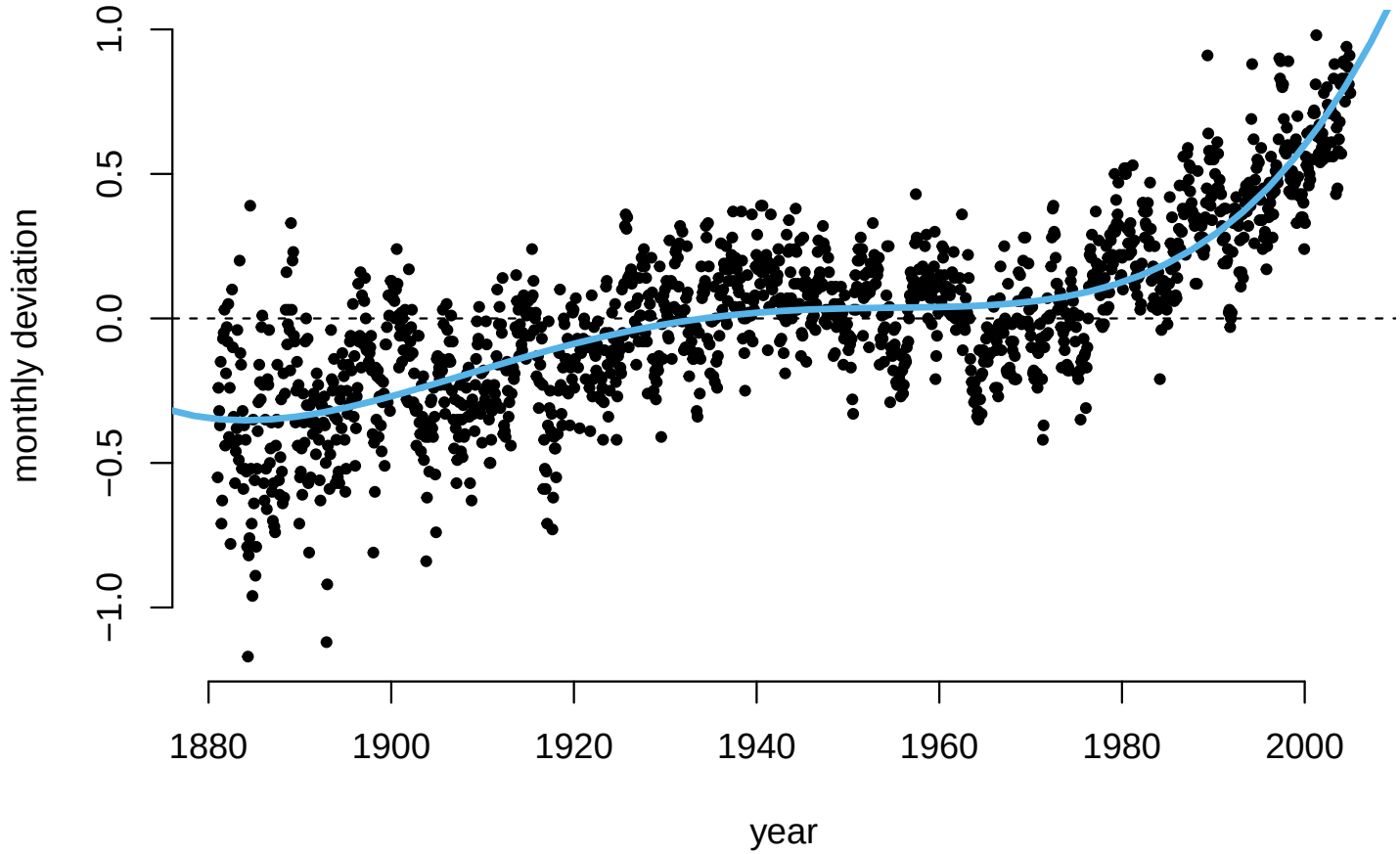


Figure: Monthly deviation of global average temperature from long-term mean, with fitted polynomial curve of order 4.

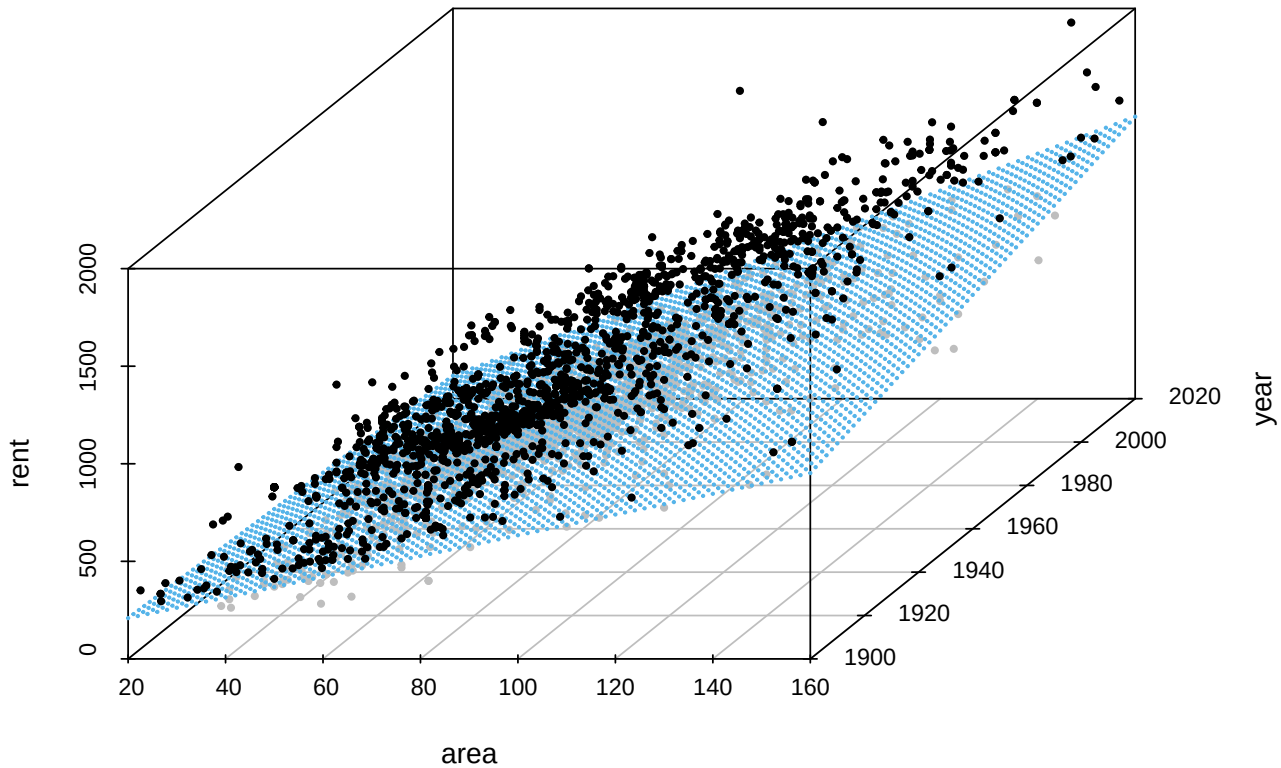
Interaction terms

- an **interaction** exists when the effect of a covariate depends on the value of another covariate
- for example, it could be the case that additional square metres are more expensive in newer than in older appartments
- this can be accounted for using an interaction term:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1} x_{i2} + \epsilon_i,$$

where x_{i1} and x_{i2} correspond to “area” and “year”, respectively

Interaction terms — illustration



$$\text{rent}_i = 2871 - 60.81 \cdot \text{area}_i - 1.46 \cdot \text{year}_i + 0.03 \cdot \text{area}_i \cdot \text{year}_i + \epsilon_i$$

Chapter 3: Non-normal data and the exponential family of distributions

Overview of this chapter

Part I:

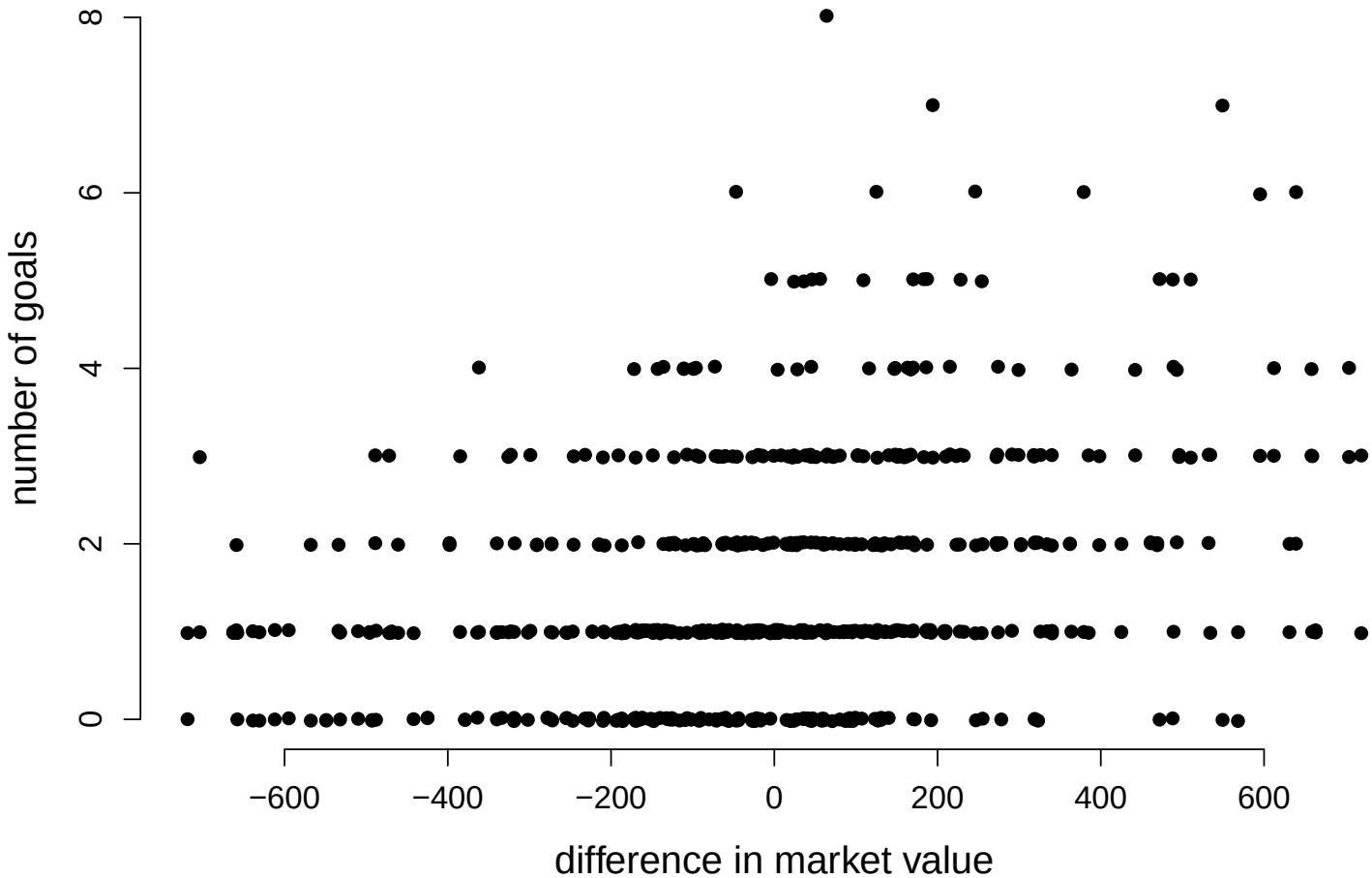
- a look back to the motivating examples from the intro
- what distributions would we want to use?
- how would the corresponding regression models look like?

Part II:

- define unifying framework: the exponential family of distributions

  $\rightsquigarrow$ this will later allow us to develop a set of inferential methods that applies to all regression models covered above (in Part I)

Example goals scored in football matches



Example football scores — response distribution

What would be an appropriate distribution for Y = number of goals?

- normal distribution isn't going to work here — otherwise any forecast distribution will be continuous, which doesn't make sense for count data
- instead, the **Poisson distribution**, with probability mass function

$$p(y) = \frac{\lambda^y}{y!} \exp(-\lambda), \quad y = 0, 1, 2, \ldots,$$

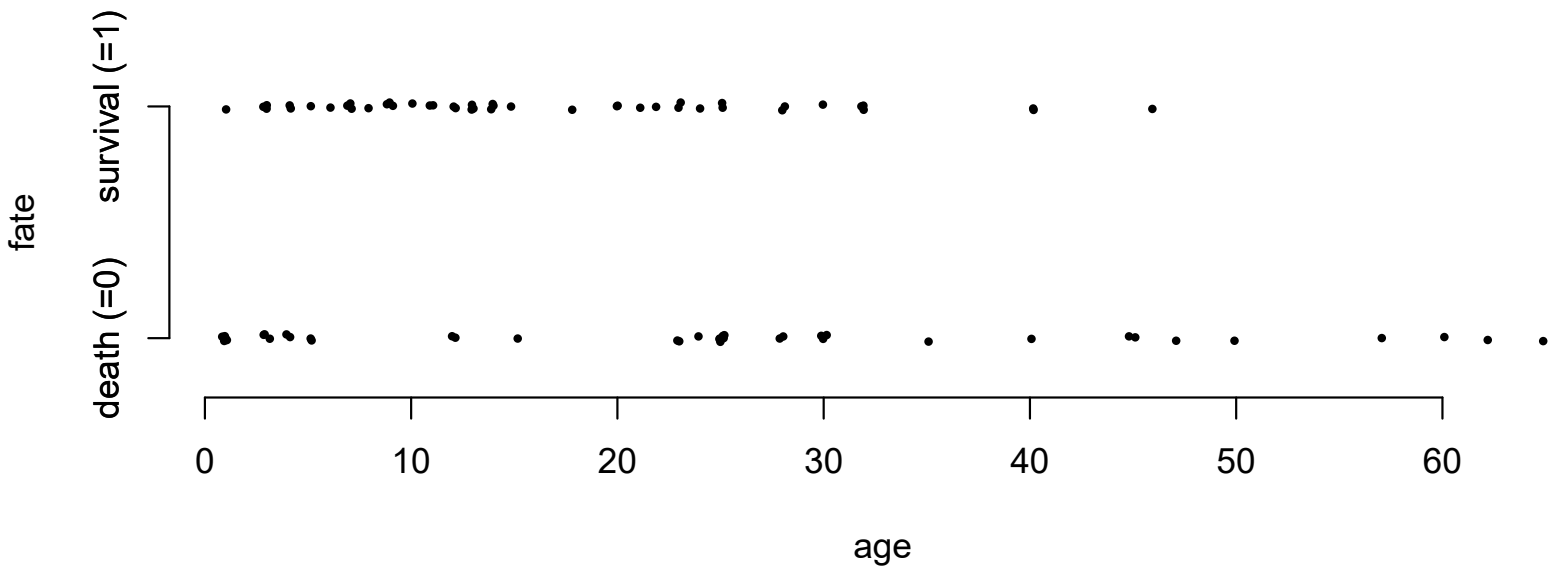
seems to be a sensible choice given that we deal with **count data**[5]

[5] the Poisson distribution is the standard choice for modelling count data — a more flexible alternative is given by the negative binomial distribution

Poisson regression models

- if $Y \sim Po(\lambda)$, then
$$\mathbb{E}(Y) = \text{Var}(Y) = \lambda$$
- the parameter λ of the Poisson distribution **needs to be positive**
- in principle, we could simply consider the model
$$Y_i \sim Po(\lambda_i), \quad \lambda_i = \mathbb{E}(Y_i) = \beta_0 + \beta_1 x_{i1}$$
- however, we then might obtain negative λ_i's for certain values of x_{i1}
- thus, we will (usually) use the following model instead:
$$Y_i \sim Po(\lambda_i), \quad \lambda_i = \mathbb{E}(Y_i) = \exp\big(\beta_0 + \beta_1 x_{i1}\big)$$

Donner party example



Donner party example — response distribution

What would be an appropriate distribution for the binary variable Y?

- normal distribution doesn't make any sense for binary outcomes
- instead, the **Bernoulli distribution**, with probability mass function

$$p(y) = \begin{cases} \pi & \text{if } y = 1; \\ 1 - \pi & \text{if } y = 0, \end{cases}$$

ought to be used

Bernoulli (logistic) regression models

- if $Y \sim \mathit{Bern}(\pi)$, then

$$\mathbb{E}(Y) = \pi, \quad \mathrm{Var}(Y) = \pi(1-\pi)$$

- the success probability per trial, π, **needs to be in** $[0, 1]$
- again, in principle, we could formulate a regression model such as

$$Y_i \sim \mathit{Bern}(\pi_i), \quad \pi_i = \mathbb{E}(Y_i) = \beta_0 + \beta_1 x_{i1}$$

- for some x_{i1} we would then obtain values outside $[0, 1]$ for π_i
- we will (usually) use the following model instead:

$$Y_i \sim \mathit{Bern}(\pi_i), \quad \pi_i = \mathbb{E}(Y_i) = \mathrm{logit}^{-1}(\beta_0 + \beta_1 x_{i1}) = \frac{\exp\left(\beta_0 + \beta_1 x_{i1}\right)}{\exp\left(\beta_0 + \beta_1 x_{i1}\right) + 1}$$

Beetle example

	n_i, the number of beetles	y_i, the number of beetles killed	y_i/n_i, the proportion of beetles killed	x_i, the concentration of toxic gas
$i = 1$	59	6	0.10	49.1
$i = 2$	60	13	0.22	53.0
$i = 3$	62	18	0.29	56.9
$i = 4$	56	28	0.50	60.8
$i = 5$	63	52	0.83	64.8
$i = 6$	59	53	0.90	68.7
$i = 7$	62	61	0.98	72.6
$i = 8$	60	60	1.00	76.5

Beetle example — response distribution

What would be an appropriate distribution for bounded counts?

- the response Y gives counts bounded by the number of beetles exposed
- assuming independence of the individual beetles, the **binomial distribution**, with probability mass function

$$p(y) = \binom{n}{y} \pi^y (1-\pi)^{n-y}, \quad y = 0, 1, \ldots, n,$$

  is the obvious choice

Binomial (logistic) regression models

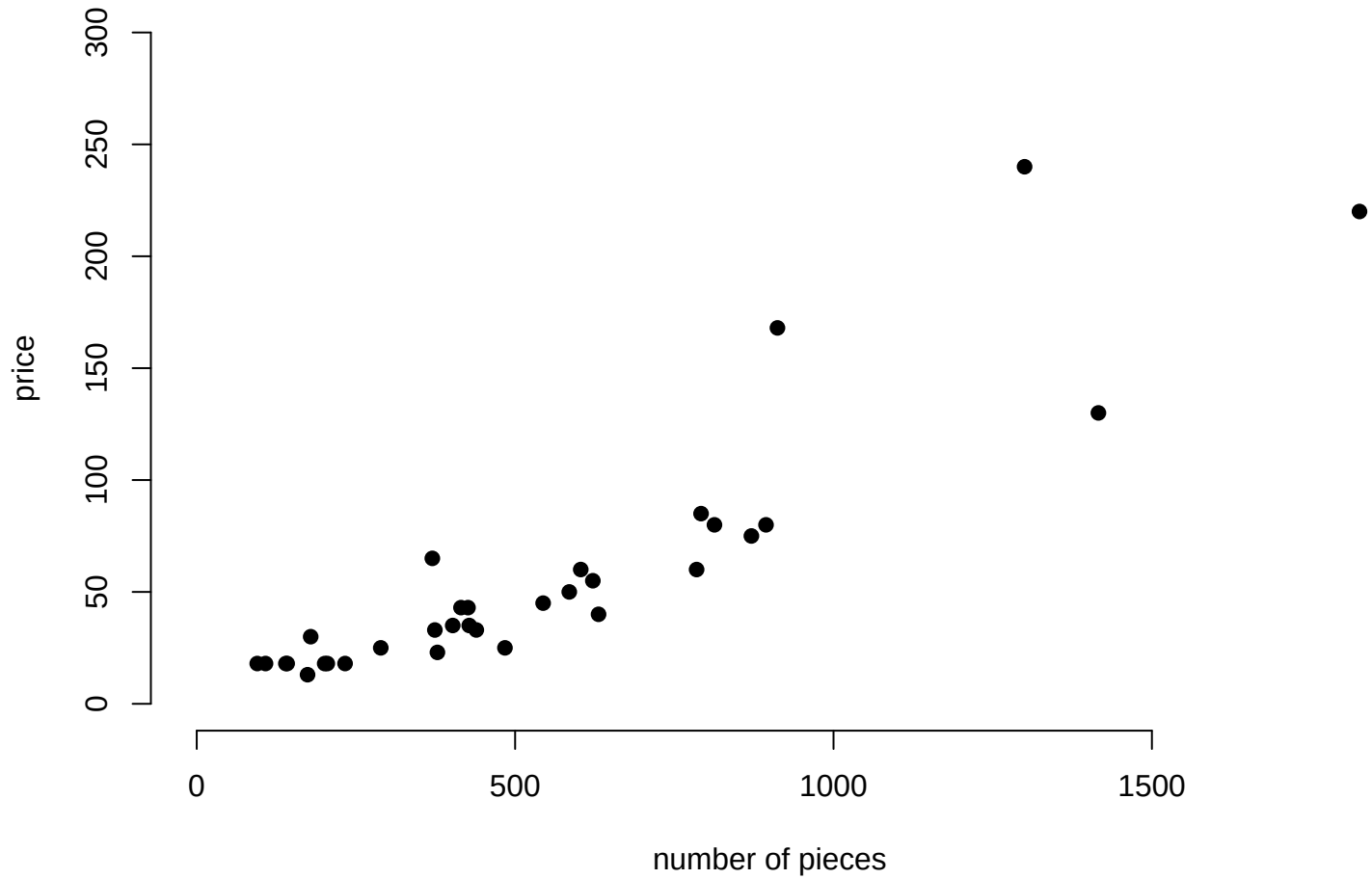
- if $Y \sim Bin(n, \pi)$, then

$$\mathbb{E}(Y) = n\pi \quad \Leftrightarrow \quad \mathbb{E}(Y/n) = \pi$$

- thus, we can use a model analogous to the one considered before:

$$Y_i \sim Bin(n_i, \pi_i), \quad \pi_i = \mathbb{E}(Y_i/n_i) = \text{logit}^{-1}(\beta_0 + \beta_1 x_{i1}) = \frac{\exp(\beta_0 + \beta_1 x_{i1})}{\exp(\beta_0 + \beta_1 x_{i1}) + 1}$$

Example Lego prices



Example Lego prices — response distribution

What would be an appropriate distribution for the positive continuous variable Y?

- normal distribution could be OK if obs. are clearly distinct from 0 — forecast distributions then wouldn't include substantial mass on negative values
- in general, the **gamma distribution** may however be more appropriate[6]

[6] there are alternatives, but the gamma distribution is already quite flexible

The gamma distribution

Density of a gamma-distributed random variable Y:

$$f(y) = \frac{y^{\nu-1}\exp(-y/\theta)}{\Gamma(\nu)\theta^{\nu}}, \quad y > 0$$

The parameters ν and θ are called shape and scale, respectively.

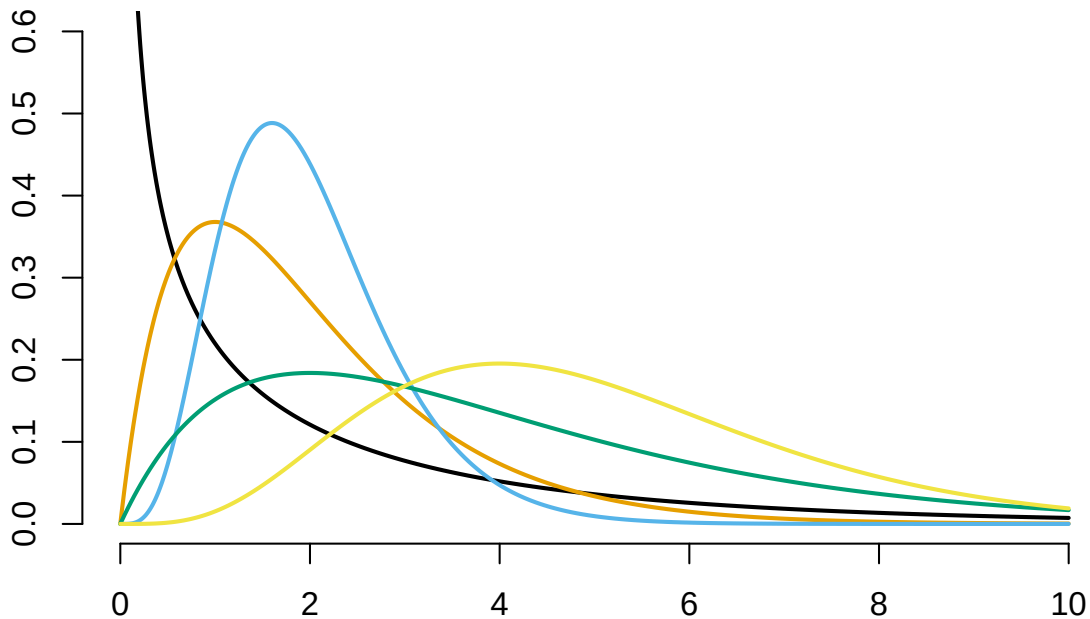


Figure: Example p.d.f.s for gamma distributions with different values of ν and θ.

Gamma regression models

- if $Y \sim G(\nu, \theta)$ (gamma distributed), then

$$\mathbb{E}(Y) = \nu\theta, \quad \text{Var}(Y) = \nu\theta^2$$

- we could for example formulate a regression model

$$Y_i \sim G(\nu_i, \theta_i), \quad \nu_i\theta_i = \mathbb{E}(Y_i) = \beta_0 + \beta_1 x_{i1}$$

- same problem as before: this model might give negative means...
- we could for example use the following model instead[7]:

$$Y_i \sim G(\nu, \theta_i), \quad \nu\theta_i = \mathbb{E}(Y_i) = \exp\big(\beta_0 + \beta_1 x_{i1}\big)$$

[7] note it will later become clear why we're considering ν to be constant

Chapter 3: Non-normal data and the exponential family of distributions

3.2 The exponential family of distributions

Exponential family — motivation

Where do we stand?

- we understand why in some regression scenarios we would like to be able to use response distributions other than the normal
- we have already seen that this will require some transformations (so-called **link functions**) to be applied in order for the models to be sensible

What was our approach so far?

- we looked at special cases (Poisson, Bernoulli, binomial & gamma)

What do we do next?

- introduce a unifying framework for distributions allowed in GLMs
- formulate GLMs and introduce associated inferential methods in the general case, rather than looking at each type of GLM separately — very neat!

The distribution of a random variable Y, dependent on a parameter θ, is said to be in the **exponential family** if it can be written as

$$f_\theta(y) = \exp\big(a(y)b(\theta) + c(\theta) + d(y)\big),$$

where a, b, c and d are fixed functions.[8]

- f_θ can be either probability density function (if Y is continuous-valued) or probability mass function (if Y is discrete-valued)
- note there is only a single parameter, θ — we can consider distributions with more parameters, then regarding some of these as **nuisance parameters**[9]
- if $a(y) = \text{const.} \cdot y$, then the distribution is in the so-called **canonical form**, with **canonical link** $b(\cdot)$ (which has some desirable theoretical properties)

[8] this is in fact just one way to define the exponential family, following Dobson & Barnett (2008) — alternative definitions are equivalent, but sometimes more explicit regarding nuisance parameters

[9] parameters not of primary interest to us, treated as constants — e.g. σ^2 in linear regression

Exponential family — examples

Among others, the following distributions are members of the exponential family:

- $Po(\lambda)$
- $\mathcal{N}(\mu, \sigma^2)$, where σ^2 is regarded as nuisance parameter
- $Bern(\pi)$
- $Bin(n, \pi)$, where n is regarded as nuisance parameter
- $G(\nu, \theta)$, where ν is regarded as nuisance parameter

Proof that $Po(\lambda)$ is in the exp. family (in class)

Proof that $Bin(n, \pi)$ is in the exp. family (in class)

Exponential family — mean and variance

- GLMs allow for any distribution from the exponential family for the response variable (see next chapter)
- in order to introduce the estimation method for general GLMs, we need some general theoretical properties of a distribution in exponential family form

For a random variable Y in exponential family form,

$$f_\theta(y) = \exp\big(a(y)b(\theta) + c(\theta) + d(y)\big),$$

we have that

(i) $\mathbb{E}\big(a(Y)\big) = \frac{-c'(\theta)}{b'(\theta)}$

(ii) $\mathrm{Var}\big(a(Y)\big) = \frac{b''(\theta)c'(\theta) - c''(\theta)b'(\theta)}{[b'(\theta)]^3}$

Proof of (i) (in class)

Proof of (ii) (in class)

Example — mean and variance of the Poisson distribution (in class)

Chapter 4: The class of generalised linear models

We now introduce the class of GLMs, which...

- ...provides a unifying umbrella for important statistical modelling techniques such as logistic regression, Poisson regression and also linear regression
- ...allows us to develop general methods that apply to each special case

A **generalised linear model** (GLM) consists of three components:

1. a distribution from the exponential family, in its canonical form, for the *independent* response variables $Y_1, \ldots, Y_n$:

$$f_{\theta_i}(y_i) = \exp\big(y_i b(\theta_i) + c(\theta_i) + d(y_i)\big),$$

2. a linear predictor, i.e. a linear combination of a set of explanatory variables,

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}, \quad i = 1, \ldots, n$$

3. an invertible and differentiable **link function** g such that

$$g\big(\underbrace{\mathbb{E}(Y_i)}_{\mu_i}\big) = \eta_i$$

Remarks on the definition — Part I

$$g(\mu_i) = g(\mathbb{E}(Y_i)) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}, \quad i = 1, \ldots, n$$

This formulation extends basic linear regression in two ways:

- (potentially) non-Gaussian response variable Y_i
- (potential) use of link function g

We will sometimes use matrix notation to simplify things:

$$g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} = \boldsymbol{\eta},$$

with design matrix $\mathbf{X}$ as in linear regression. Here g is applied componentwise.

Remarks on the definition — Part II

Since the link function is invertible, we can also write

$$\mu_i = \mathbb{E}(Y_i) = g^{-1}(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$$

In particular, this allows us:

- to display the fitted model in an "Y against x" scatter plot of the data
- in other words, to regard $\mathbb{E}(Y)$ as a function of the x variables
- to do prediction based on a fitted GLM

Remarks on the definition — Part III

In many cases we'll have that $\mu_i = \theta_i$, where θ_i is the parameter of the distribution in its exponential family form, e.g. in Poisson regression.

However, in general μ_i can be some function of θ_i — gamma regression is one example where $\mu_i \neq \theta_i$.

So, given some data, how do I specify my GLM?

- first choose a distribution for the response
- the same type of distribution must be used for each Y_i
  (i.e., you can't have $Y_1 \sim \mathcal{N}$ and $Y_2 \sim G$)
- the type of data at hand will suggest appropriate distributions
  (e.g., binary data $\rightsquigarrow$ Bernoulli, count data $\rightsquigarrow$ Poisson, etc.)
- the link function g is also chosen as part of the model specification,
  and can be any function that is differentiable & invertible
- there is usually more than just one adequate function g that can be used

GLMs in R (a brief overview)

In R, GLMs can be fitted using the function `glm`, which has the following form:

```
glm(formula=..., family=...(link=...))
```

The `formula` component is analogous as in case of `lm`
(e.g. `formula=y~x` for response Y and linear predictor $\beta_0 + \beta_1 x$)

With `family`, the distribution assumed for the response is specified.
(e.g. `family=poisson(...)`)

With `link`, the link function is specified.
(e.g. `link="log"`)

If no link function is specified, then **by default the canonical link function** for the given exponential family distribution is used (see next slide).

The **canonical link functions** for the distributions of interest to us are:

distribution	canonical link	
normal	$g(\mu) = \mu$	("identity link")
Poisson	$g(\mu) = \log(\mu)$	("log link")
Bernoulli/binomial	$g(\mu) = \log(\mu/(1-\mu))$	("logit link")
gamma	$g(\mu) = 1/\mu$	("inverse link")

Using the canonical link function simplifies inference in GLMs, but alternative link functions can be used as well.

Chapter 4: The class of generalised linear models

4.2 The special case linear regression

The (standard) **linear regression model**,

$$\mu_i = \mathbb{E}(Y_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip},$$

is a (very simple) GLM, where

(i) the response variable is (commonly assumed to be) normally distributed

(ii) the link function is the identity link ($g(\mu) = \mu$)

- for this special case, we will later see that the much more generally derived GLM estimation method reduces to ordinary least squares

- R code for the case $p = 1$:

```
glm(y~x,family=gaussian)
```

(which gives the same $\hat{\beta}$ as `lm(y~x)`, but is a different function — in particular, the output is slightly different)

Making use of the link function

Assume that $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with a suspected true relationship to x_i of the form[10]

$$\mu_i = \mathbb{E}(Y_i) = \beta_0 x_{i1}^{\beta_1}$$

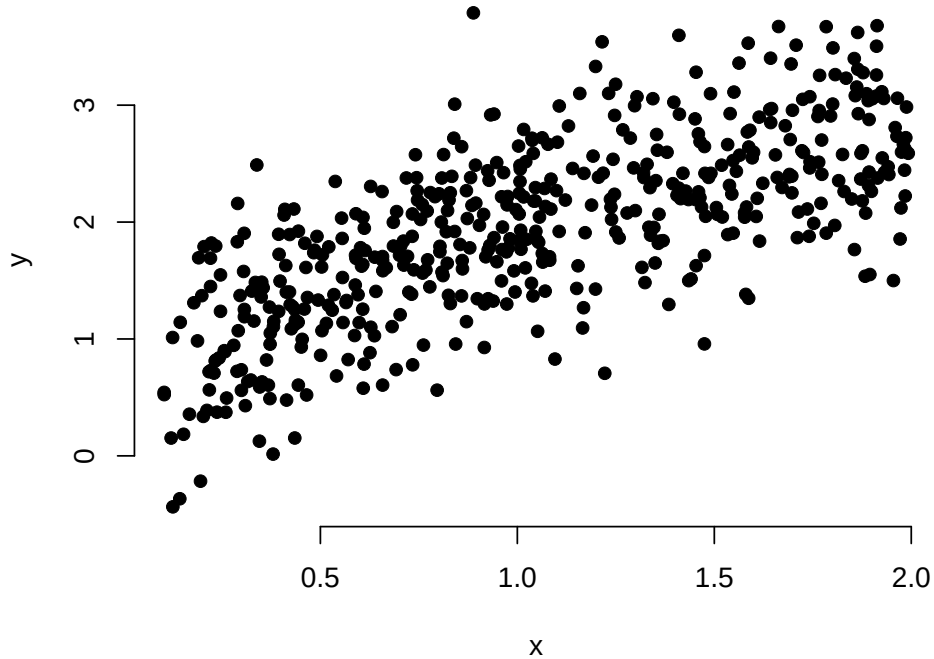
- this isn't a linear regression model, and also can't be converted into a linear regression model using variable transformation
- however, applying the log link, $g(\mu) = \log(\mu)$, to the model eq. we obtain

$$g\big(\mathbb{E}(Y_i)\big) = \log\big(\mathbb{E}(Y_i)\big) = \underbrace{\log(\beta_0)}_{=\beta_0^*} + \beta_1 \underbrace{\log(x_{i1})}_{=x_{i1}^*}$$

- this is a perfectly valid GLM!
  - $\rightsquigarrow$ estimate β_0^* and β_1 using GLM machinery (normal response & log link)
  - $\rightsquigarrow$ obtain estimate $\hat{\beta}_0$ by back-transforming: $\hat{\beta}_0 = \exp(\hat{\beta}_0^*)$

[10] there could e.g. an economic/biological/medical/etc. theory that motivates exactly this formulation

Below are data simulated from $\mathbb{E}(Y_i) = \beta_0 x_i^{\beta_1} = 2x_{i1}^{0.5}$



Fitting the GLM from the previous slide[11] gives:

$$\hat{\beta}_0^* = 0.686 \quad \Rightarrow \quad \hat{\beta}_0 = \exp(\hat{\beta}_0^*) = 1.986$$
$$\hat{\beta}_1 = 0.491$$

[11] `glm(y~log(x),family=gaussian(link="log"))`

A second example of how the link function can be utilised

Now suppose that $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ with a suspected relationship to x_i of the form

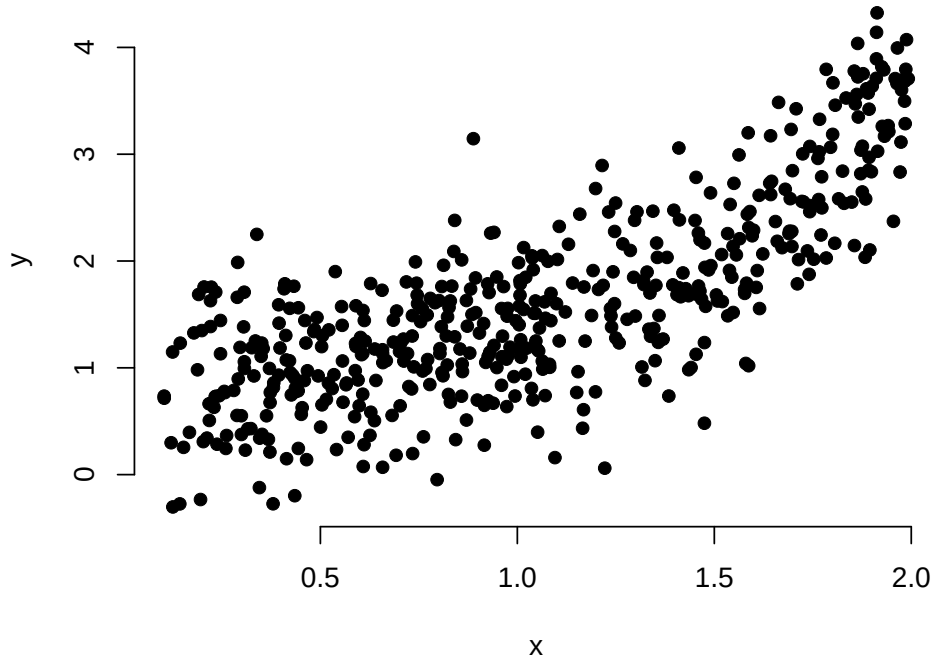
$$\mu_i = \mathbb{E}(Y_i) = \frac{\beta_0}{\beta_1 - x_{i1}}$$

- again, this model can't be fitted using ordinary least squares
- however, applying the inverse link function, $g(\mu) = \mu^{-1}$, we obtain

$$g\big(\mathbb{E}(Y_i)\big) = \big(\mathbb{E}(Y_i)\big)^{-1} = \underbrace{\frac{\beta_1}{\beta_0}}_{=\beta_0^*} - \underbrace{\frac{1}{\beta_0}}_{=\beta_1^*} x_{i1}$$

- again this is a perfectly valid GLM!
  - $\rightsquigarrow$ estimate β_0^* and β_1^* using GLM machinery
  - $\rightsquigarrow$ obtain estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ by back-transforming

Below are data simulated from $\mathbb{E}(Y_i) = \frac{\beta_0}{\beta_1 - x_i} = \frac{2}{2.5 - x_{i1}}$



Fitting the GLM from the previous slide[12] gives:

$$\hat{\beta}_0^* = 1.237, \quad \hat{\beta}_1^* = -0.489$$

$$\Rightarrow \quad \hat{\beta}_0 = -1/\hat{\beta}_1^* = 2.047, \quad \hat{\beta}_1 = -\hat{\beta}_0^*/\hat{\beta}_1^* = 2.532$$

[12] `glm(y~x,family=gaussian(link="inverse"))`

Purposes of the link function

In the previous two examples, we saw that in some cases, the purpose of using a link function is simply to **convert a non-linear model into a linear model**.

The other, more common usage of link functions aims at **meeting range restrictions**, e.g. making sure that the mean of a Poisson response is positive.

Chapter 4: The class of generalised linear models

4.3 Poisson regression

The **Poisson GLM**[13], with canonical link, i.e. $g(\mu) = \log(\mu)$, is

$$\log\big(\mathbb{E}(Y_i)\big) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$$

$$\Leftrightarrow \qquad \mathbb{E}(Y_i) = e^{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}}$$

where the Y_i are independently Poisson distributed.

R code for the case where $p = 1$:

```
glm(y~x,family=poisson)
```

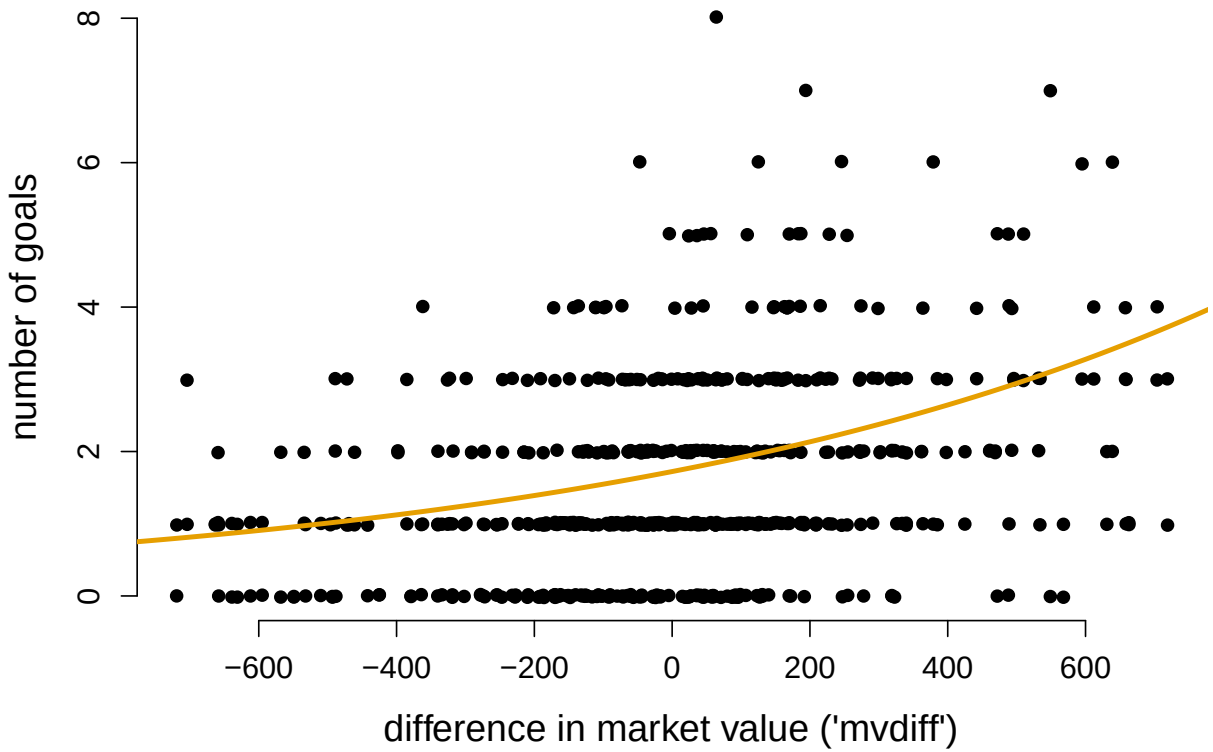
Other link functions can be used — implemented in the R function `glm` are:

- $g(\mu) = \log(\mu)$
- $g(\mu) = \mu$
- $g(\mu) = \sqrt{\mu}$

In practice, the canonical (log) link is almost always used.

[13] also known as Poisson regression

Poisson GLM fitted to Bundesliga matches (season 2018/19)




```
> mod<-glm(goals~mvdiff,family=poisson)
> mod$coeff
(Intercept)       MWdiff
   0.416864     0.001078
```

Forecasting Bundesliga matches

If for a moment we're willing to assume that our Poisson GLM is a good model[14], then we can now make probabilistic forecasts of Bundesliga matches.

Example from the next matchday:

- Schalke's market value: 57
- Bayern's market value: 879
- market value difference: -822 (from Schalke's perspective)

$\rightsquigarrow$ goals by Schalke $\sim Po\left(e^{0.4169+0.0011\cdot(-822)}\right) = Po\left(e^{-0.487}\right) = Po(0.614)$

$\rightsquigarrow$ goals by Bayern $\sim Po\left(e^{0.4169+0.0011\cdot 822}\right) = Po\left(e^{1.321}\right) = Po(3.748)$

[14] which it isn't, it's too simple, many important covariates are missing

	0	1	2	3	4	5	6
0	0.013	0.048	0.090	0.112	0.105	0.079	0.049
1	0.008	0.029	0.055	0.069	0.064	0.048	0.030
2	0.002	0.009	0.017	0.021	0.020	0.015	0.009
3	0.000	0.002	0.003	0.004	0.004	0.003	0.002
4	0.000	0.000	0.001	0.001	0.001	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table: Probabilities of match outcomes (Schalke goals in rows, Bayern goals in columns).

Derived probabilities:

- of Schalke winning the match: 0.026
- of a draw: 0.064
- of Bayern winning the match: 0.910

Interpretation of the regression coefficients

Consider a simple Poisson GLM of the form:

$$\mathbb{E}(Y_i) = e^{\beta_0 + \beta_1 x_{i1}} \tag{1}$$

What's the interpretation of β_1?

Recall that, in simple linear regression, $\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_{i1}$, an increase in x_{i1} by 1 (unit) *adds* β_1 to $\mathbb{E}(Y_i)$.

This is what happens when in (1) the covariate value x_{i1} increases by 1 unit:

$$e^{\beta_0 + \beta_1 (x_{i1}+1)} = e^{\beta_0 + \beta_1 x_{i1} + \beta_1} = e^{\beta_0 + \beta_1 x_{i1}} \cdot e^{\beta_1}$$

In other words, an increase in x_{i1} by 1 changes $\mathbb{E}(Y_i)$ by the *factor* e^{β_1}.

- ⇝ additive change in covariate leads to multiplicative change in response
- ⇝ the exact effect of a change depends on the level of the response
- ⇝ the interpretation is not exactly intuitive

Chapter 4: The class of generalised linear models

4.4 Logistic regression

Bernoulli response variables

- consider now the case where the response Y_i is a Bernoulli trial with success probability π_i ($\rightsquigarrow$ Donner party example)
- this time, we need a function g which within the model formulation

$$g(\underbrace{\mathbb{E}(Y_i)}_{=\mu_i=\pi_i}) = \eta_i = \beta_0 + \beta_1 x_{i1}$$

  guarantees that $\pi_i \in [0, 1]$ (and analogously for general p)

- in other words, g^{-1} needs to be a mapping from $\mathbb{R}$ to $[0, 1]$
- we would like g^{-1} to be such that
  - its image is the entire interval $[0, 1]$
  - an increase in η_i leads to an increase in $\theta_i = g^{-1}(\eta_i)$

The logit link

The **inverse logit link** function,

$$\begin{aligned} \text{logit}^{-1} : \mathbb{R} &\longrightarrow [0, 1] \\ \eta &\mapsto \frac{\exp(\eta)}{\exp(\eta) + 1} \end{aligned}$$

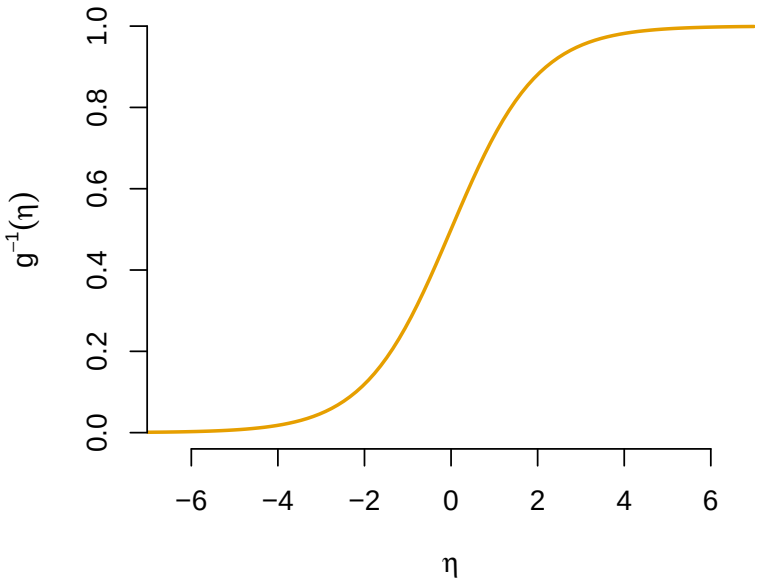
is strictly monotonically increasing with

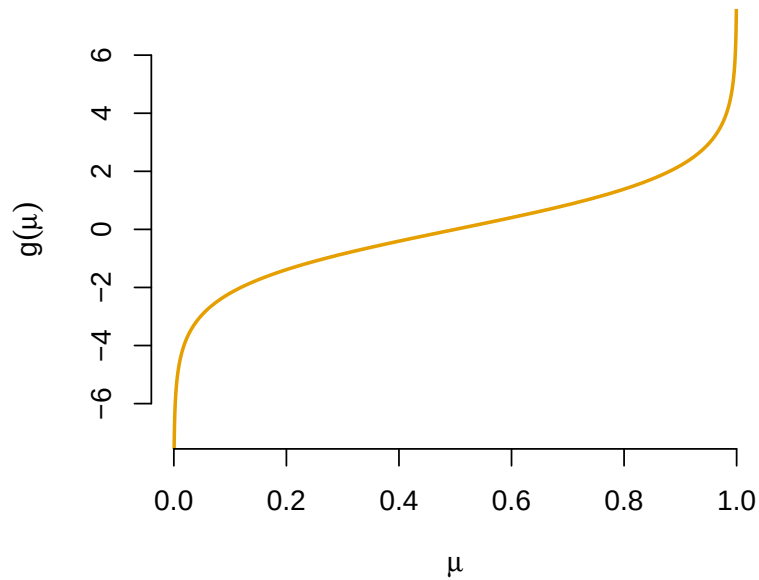
$$\lim_{\eta \to -\infty} \text{logit}^{-1}(\eta) = 0 \quad \text{and} \quad \lim_{\eta \to \infty} \text{logit}^{-1}(\eta) = 1.$$

Its inverse function is the **logit link**,

$$\text{logit}(\mu) = \log\left(\frac{\mu}{1 - \mu}\right),$$

i.e. the canonical link function for the binomial distribution.





- the function $g(\mu) = \text{logit}(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$ is called the **logit function**
- the function $g^{-1}(\eta) = \text{logit}^{-1}(\eta) = \frac{\exp(\eta)}{\exp(\eta)+1}$ is the **inverse logit function**[15]

[15] also known as the **logistic function** (I prefer "inverse logit" to contrast it with the logit function)

Bernoulli GLM (logistic regression)

The **Bernoulli GLM**, with canonical link[16], i.e. $g(\mu) = \text{logit}(\mu)$, is

$$\text{logit}\big(\mathbb{E}(Y_i)\big) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip},$$

$$\Leftrightarrow \qquad \mathbb{E}(Y_i) = \frac{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}}{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}} + 1},$$

where the Y_i are independently Bernoulli distributed.

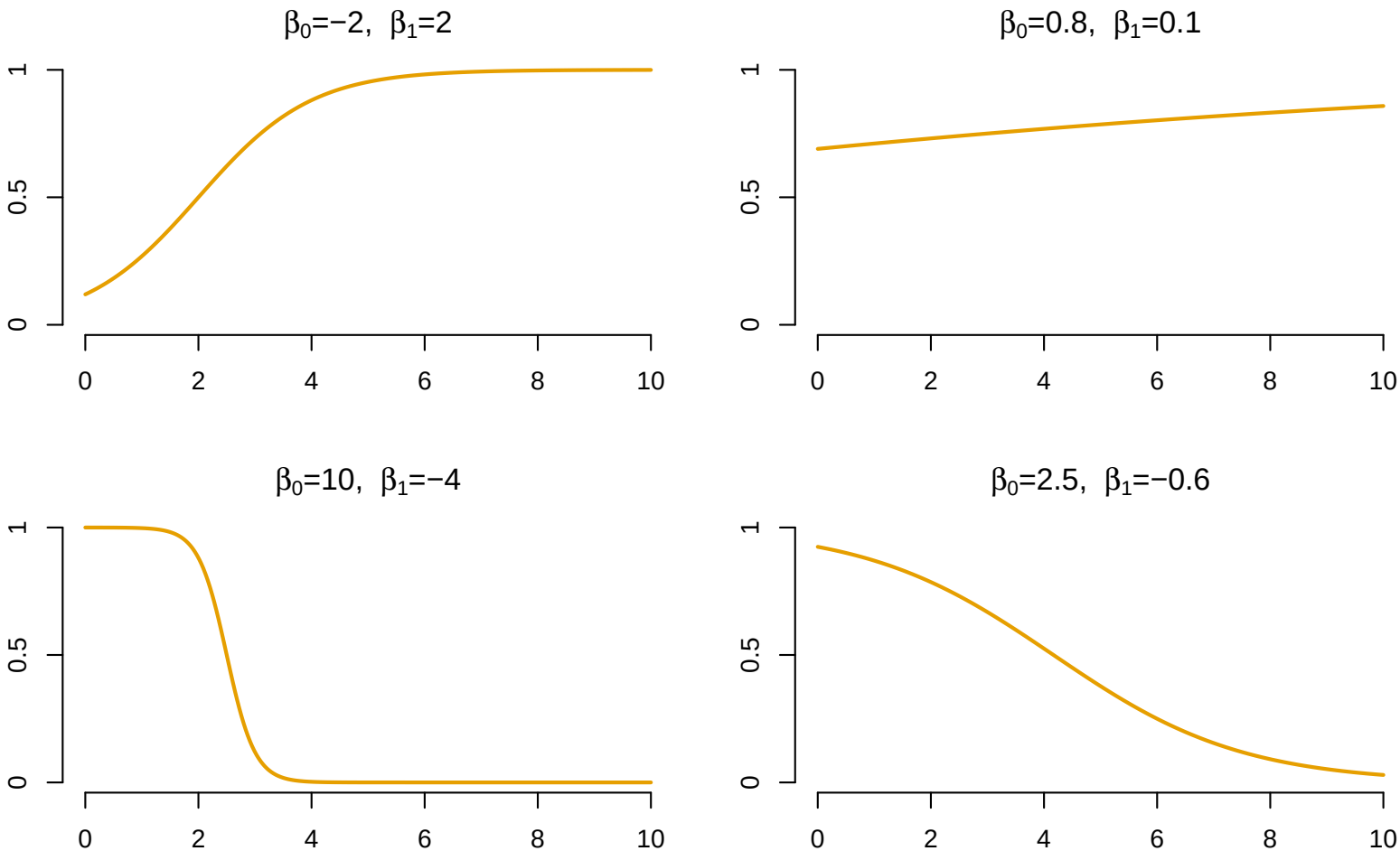
Again, other link functions can be used — implemented in `glm` are:

- $g(\mu) = \text{logit}(\mu)$
- $g(\mu) = \text{probit}(\mu)$
- $g(\mu)$ equal to the quantile function of the Cauchy distribution

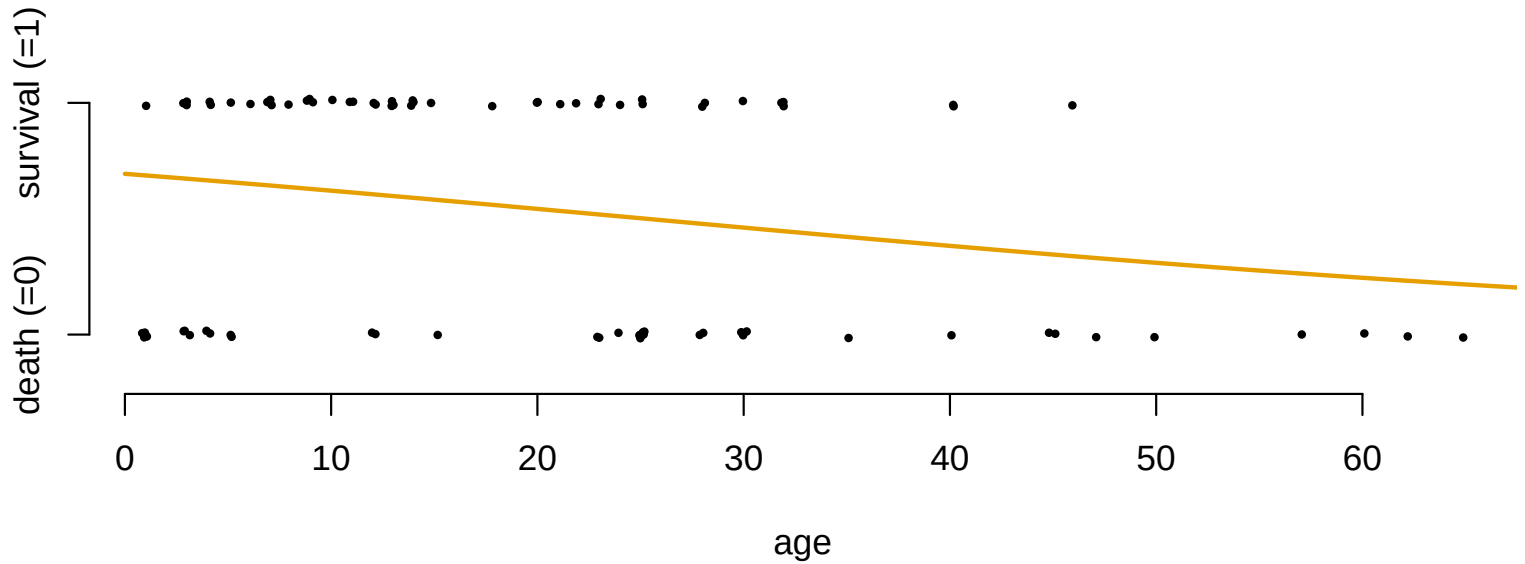
In R, logit and inverse logit are simply `qlogis` and `plogis`, respectively.

[16] in this case usually referred to as **logistic regression**

Some illustrations of how the function $\text{logit}^{-1}\left(\beta_0 + \beta_1 x_{i1}\right)$ can look like:



Logistic regression in the Donner party example




```
> mod<-glm(survival~age,family=binomial)
> mod$coeff
(Intercept)         age
 0.81732676 -0.03236981
```

Interpretation of the model parameters

- interpretation of the estimated regression coefficiencts again isn't as straightforward as with linear regression models
- ideally, we'd like to make statements such as:

  "If x increases by 1 unit, then ... (?)"

- to complete this sentence, we first define the **odds** (of success),

$$\text{odds}(\text{success}) = \frac{\Pr(\text{success})}{\Pr(\text{no success})}$$

- for example, if your chance of passing the exam is 90%, then your odds are 9/1 ("9 to 1" — you pass in 9 out of 10 cases)

Consider now the so-called **odds ratio**:

$$\text{odds ratio} = \frac{\text{odds(success for } x+1)}{\text{odds(success for } x)}$$

For the logistic regression model $\mathbb{E}(Y) = \text{logit}^{-1}(\beta_0 + \beta_1 x)$, we have:

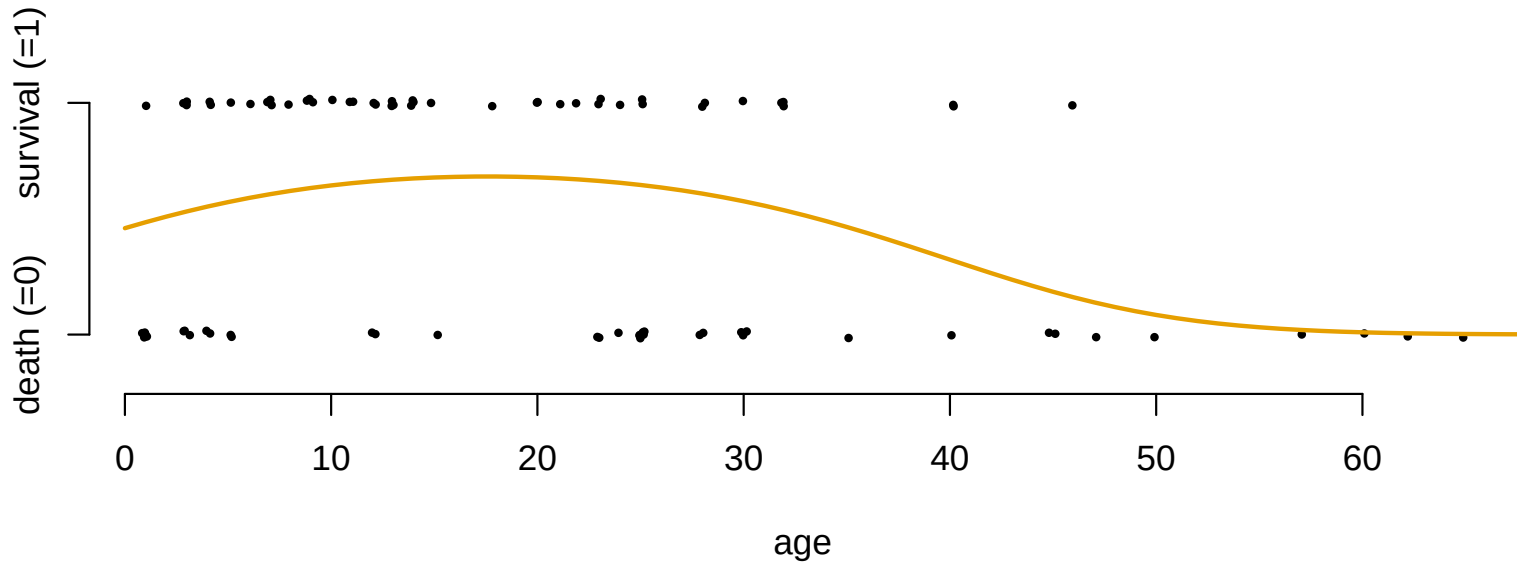
$$\text{odds(success)} =$$

$$\text{odds ratio} =$$

Interpretation in the Donner Party example:

- $\exp(\hat{\beta}_1) = \exp(-0.0324) = 0.968$, i.e. the odds of survival decrease by a factor 0.968 if age increases by 1 (year)

Logistic regression with quad. predictor in the Donner party example




```
> mod<-glm(survival~age+I(age^2),family=binomial)
> mod$coeff
(Intercept)           age      I(age^2)
  -0.163558      0.105413     -0.002995
```

Bernoulli response & probit link

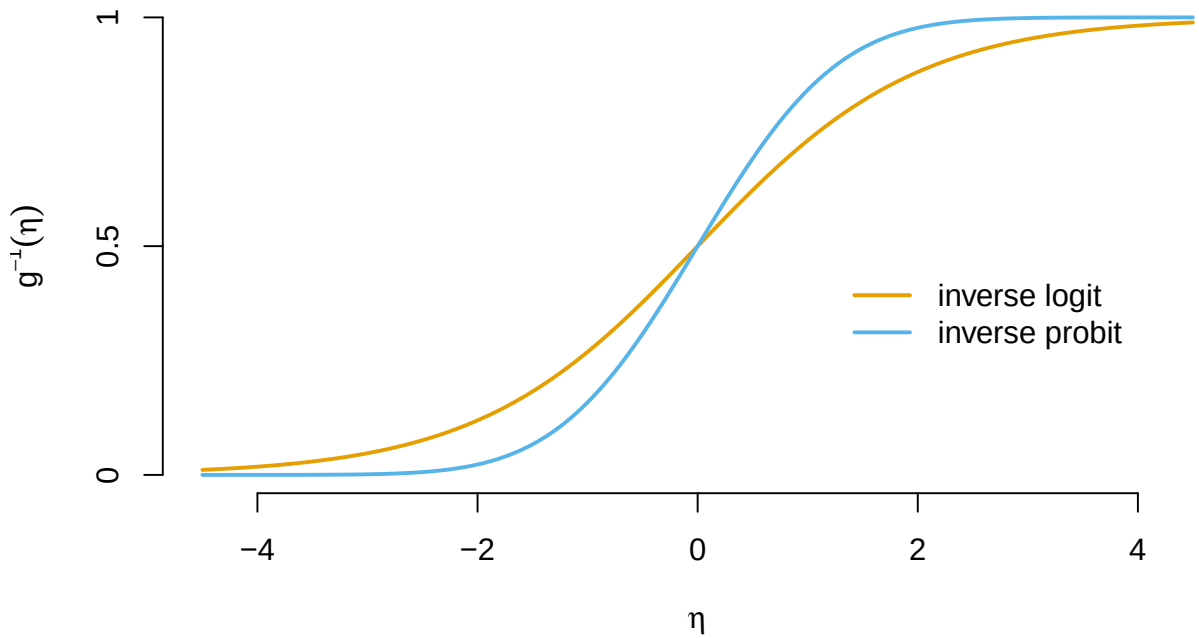
- for Bernoulli responses, the **probit link** is also popular
- the inverse of the probit link is simply the cumulative distribution function of the standard normal distribution:

$$\text{probit}^{-1}(\eta) = F_{\mathcal{N}(0,1)}(\eta) = \int_{-\infty}^{\eta} f_{\mathcal{N}(0,1)}(z)dz$$

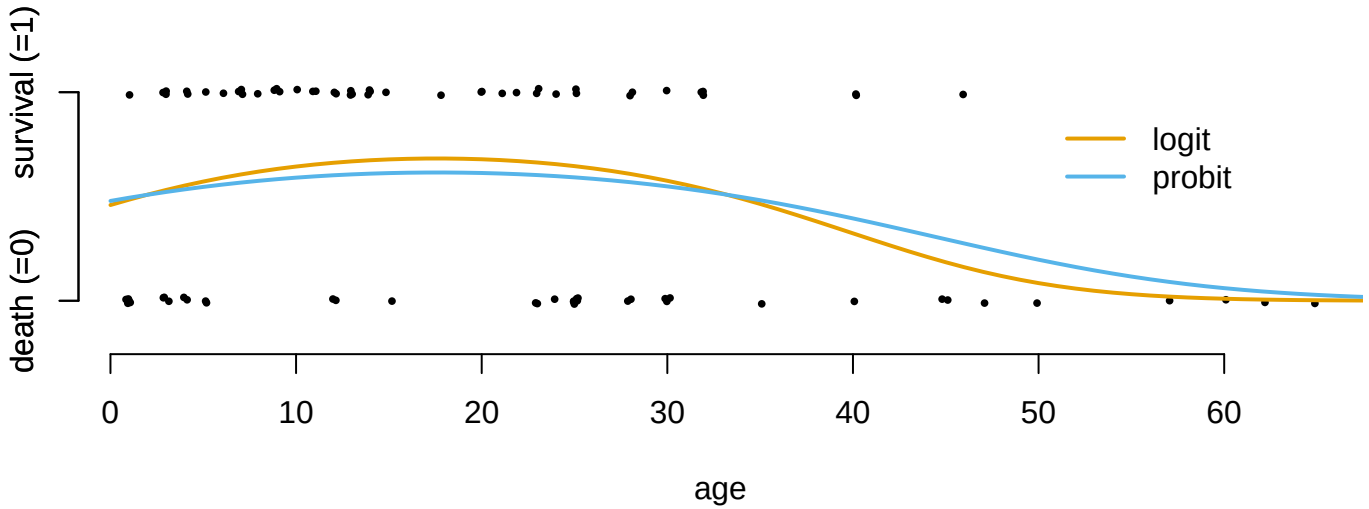
- the probit link hence is the quantile function of the standard normal
- in R, probit and inverse probit are simply `qnorm` and `pnorm`, respectively

Logit vs. probit link

- preference which of the two is used tends to vary by discipline
  (e.g. economists often use the probit link)
- in practice, it usually makes little difference which of the two is used



Donner party example — logit vs. probit




```
> mod<-glm(survival~age+I(age^2),family=binomial)
> mod$coeff
(Intercept)             age      I(age^2)
-0.163557749   0.105412918 -0.002994803

> mod<-glm(survival~age+I(age^2),family=binomial(link="probit"))
> mod$coeff
(Intercept)             age      I(age^2)
-0.08565929   0.06294872 -0.00178696
```

Binomial GLM (still logistic regression)

- consider a response Y_i which is the sum of n indep. Bernoulli trials, each with success probability π_i, such that $Y_i \sim Bin(n_i, \pi_i)$ (⇝ beetle example)
- the binomial distribution is a member of the exponential family, with nuisance parameter n_i and canonical link the **logit**
- in a **binomial GLM**, we model the (expected) **proportion of successes**:

$$g(\underbrace{\mathbb{E}(Y_i/n_i)}_{=\pi_i}) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip},$$

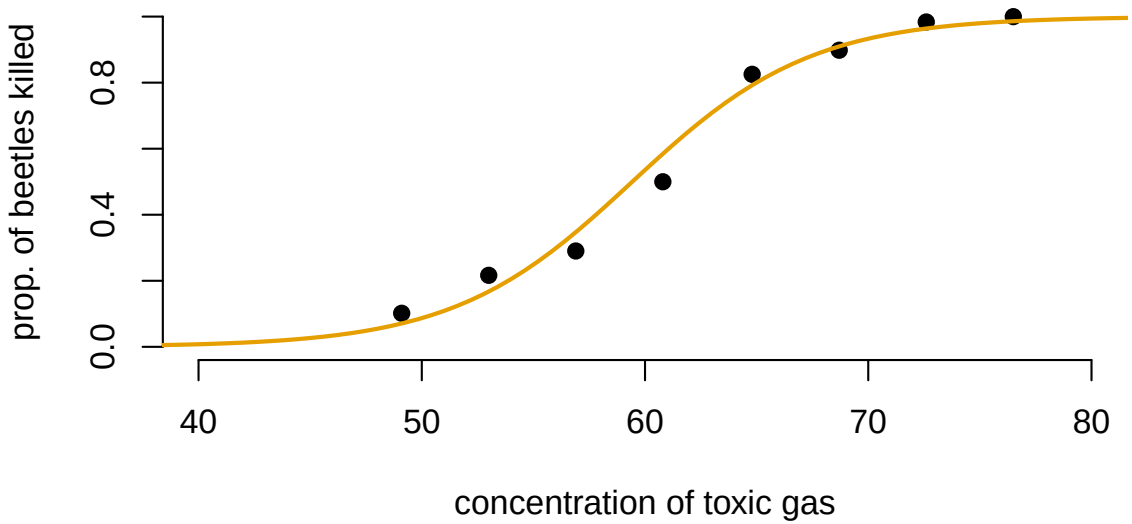
  where g could for example be the logit or the probit link
- a Bernoulli GLM is a special case of a binomial GLM (where $n_i = 1$ for all i)
- when using `glm` in R, unless you're fitting a Bernoulli GLM, you have to provide information on the n_i

Logistic regression with binomial response — beetle example

Using the command `glm(cbind(y,n-y)~x,family=binomial)`, fitting the GLM

$$\text{logit}\big(\pi_i\big) = \text{logit}\big(\mathbb{E}\big(Y_i/n_i\big)\big) = \beta_0 + \beta_1 x_{i1}$$

to the beetle data gives $\hat{\beta}_0 = -14.82$ and $\hat{\beta}_1 = 0.25$. (significant at the 1% level)



Note that it's equivalent to fit a Bernoulli GLM where the faith of each individual beetle is treated as one data point.

Chapter 4: The class of generalised linear models

4.5 Gamma regression

Some preliminary remarks

- gamma regression can be useful when modelling strictly positive response variables (e.g. income, rent, price, etc.)
- recall that for the gamma distribution, defined using ν (shape) and θ (scale), mean and variance are given by $\nu\theta$ and $\nu\theta^2$, respectively
- the gamma distribution can be written in exponential family form by regarding ν as a nuisance parameter (think of σ^2 in linear regression)
- in a gamma GLM, we again model the mean as a function of covariates, assuming a fixed (nuisance) shape ν
- canonical link: $g(\mu) = \frac{1}{\mu}$, the **inverse link**
- note that the canonical link unfortunately is not range-preserving, i.e. it does not guarantee that the resulting mean of the gamma distribution is positive

Gamma GLM (gamma regression)

The **gamma GLM**, usually referred to as **gamma regression**, with canonical link, i.e. $g(\mu) = \frac{1}{\mu}$, is

$$\begin{aligned} \big(\mathbb{E}(Y_i)\big)^{-1} &= \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}, \\ \Leftrightarrow \qquad \mathbb{E}(Y_i) &= \frac{1}{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}}, \end{aligned}$$

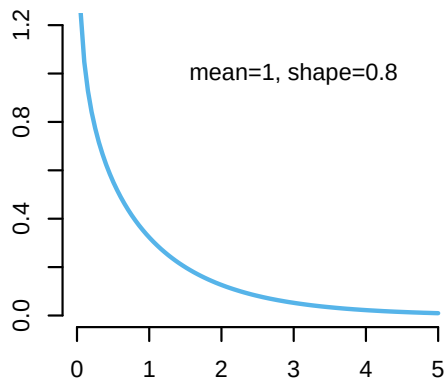
where the Y_i are independently gamma distributed.

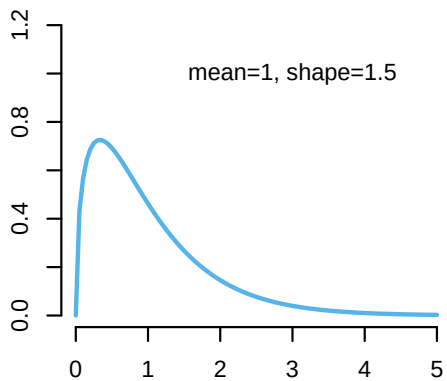
Again, other link functions can be used — implemented in `glm` are:

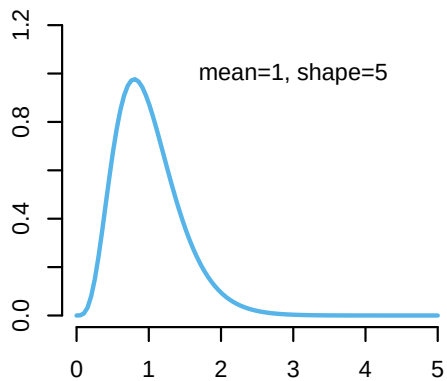
- $g(\mu) = \mu^{-1}$
- $g(\mu) = \log(\mu)$
- $g(\mu) = \mu$

Out of these only the log link is range-preserving, and therefore this link is usually used in practice (despite theoretical advantages of the canonical link function).

The role of ν







Effectively, ν is the analogue to σ^2 in linear regression models:

- we're not modelling ν using covariates (it's a nuisance parameter)
- but ν does affect the distribution — the plots here illustrate possible shapes of the distribution (around a mean of 1) that result from the value of ν
- thus, ν **determines the shape** — but note it also affects the variance:

$$\text{Var}(Y_i) = \nu\theta_i^2 = \frac{1}{\nu}\mathbb{E}(Y_i)^2$$

Some properties of the gamma GLM

The model implies a **constant coefficient of variation** (CV):

$$\text{CV}(Y_i) = \frac{\text{sd}(Y_i)}{\mathbb{E}(Y_i)} = \frac{\sqrt{\nu}\theta_i}{\nu\theta_i} = \nu^{-0.5} = \text{const.},$$

which is something that is indeed often found in practice[17]. Put differently, gamma GLMs by default accommodate heteroscedasticity.

The gamma distribution is **right-skewed** (i.e. has a heavy right tail), which often fits nicely to real data (e.g. income data).

The exponential distribution is the special case of the gamma distribution where $\nu = 1$ — hence no additional "exponential GLM" required.

[17] variability in observations increases linearly in mean

Gamma GLM in the Lego example (inverse link)

```
> mod<-glm(price~pieces,family=Gamma)
> mod$coeff
(Intercept)       pieces
  3.104e-02   -1.569e-05
> summary(mod)
 [...]
 (Dispersion parameter for Gamma family taken to be 0.3207701)
```

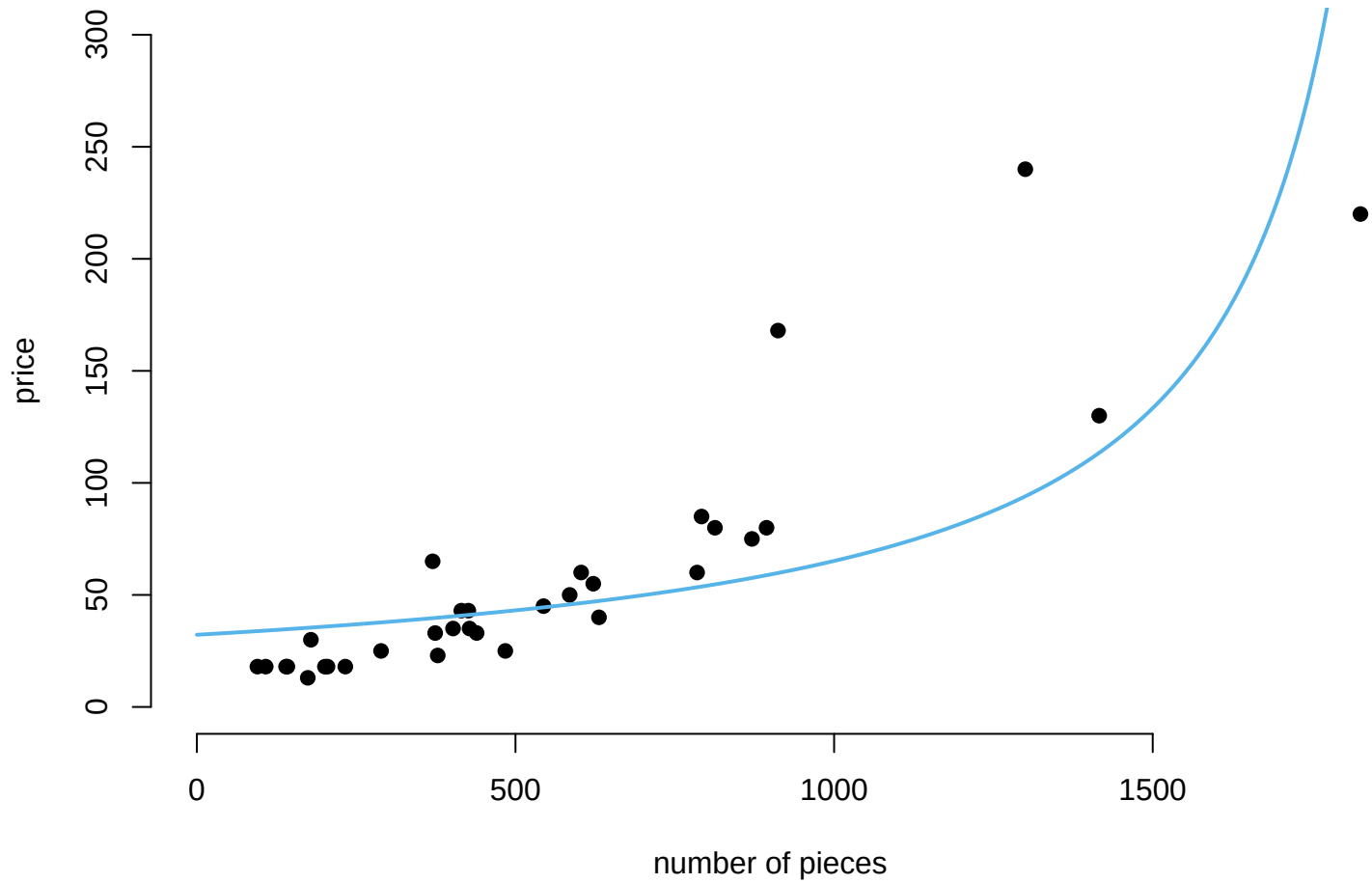
The dispersion parameter displayed here is related to ν as follows:

$$\nu = \frac{1}{\text{dispersion}}$$

Thus, the model fitted here is

$$\mathbb{E}(\text{price}_i) = \frac{1}{0.031 - 0.000016 \cdot \text{pieces}_i},$$

with price being gamma distributed with shape $\hat{\nu} = \frac{1}{0.321} = 3.117$.



Now that didn't work so well...

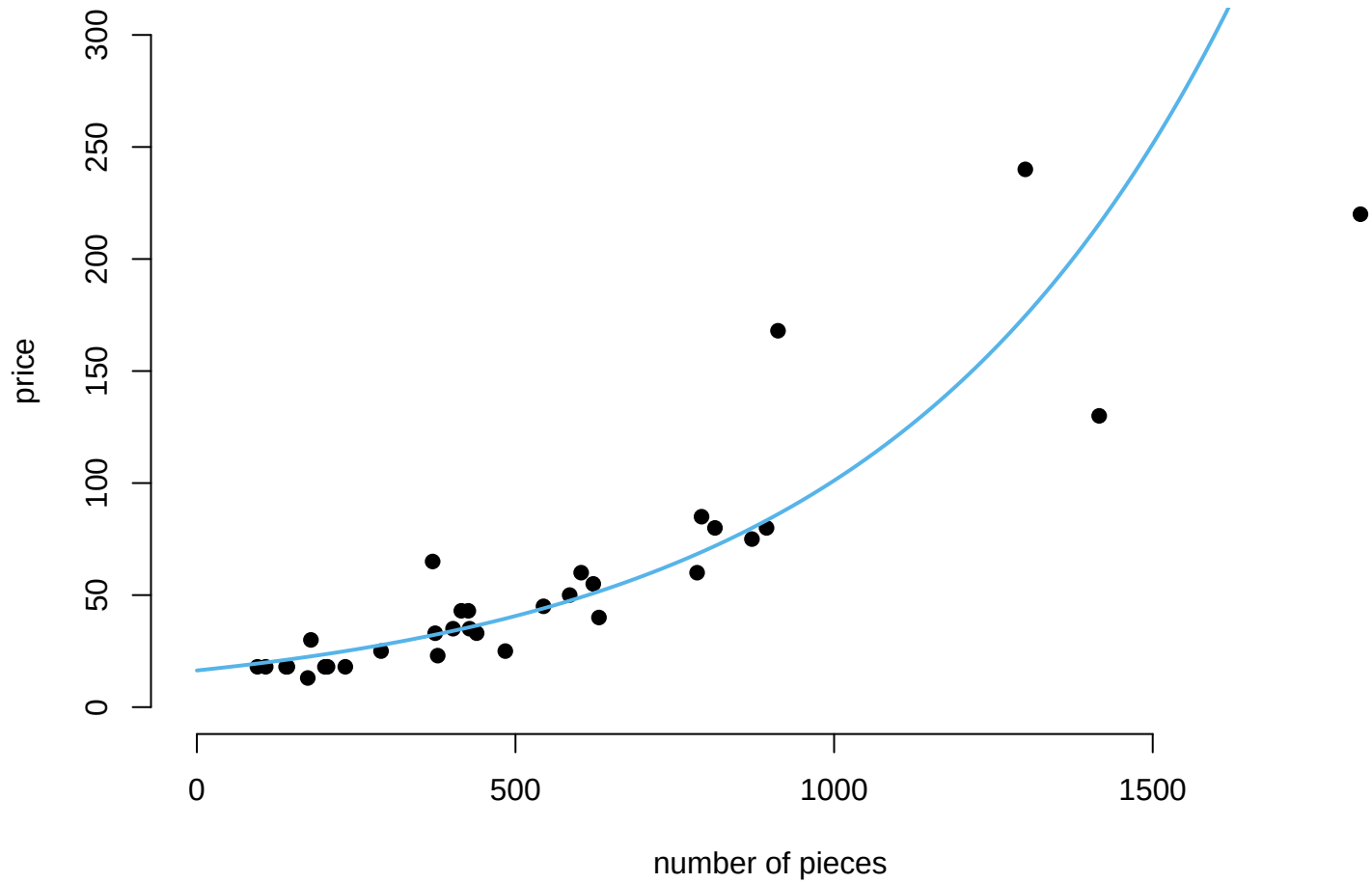
Gamma GLM in the Lego example (log link)

```
> mod<-glm(price~pieces,family=Gamma(link="log"))
> mod$coeff
(Intercept)      pieces
   2.794996    0.001821
> summary(mod)
 [...]
 (Dispersion parameter for Gamma family taken to be 0.1157603)
```

The model fitted here is

$$\mathbb{E}(\text{price}_i) = e^{2.795+0.0018\cdot\text{pieces}_i},$$

with price being gamma distributed with shape $\hat{\nu} = \frac{1}{0.116} = 8.639$.



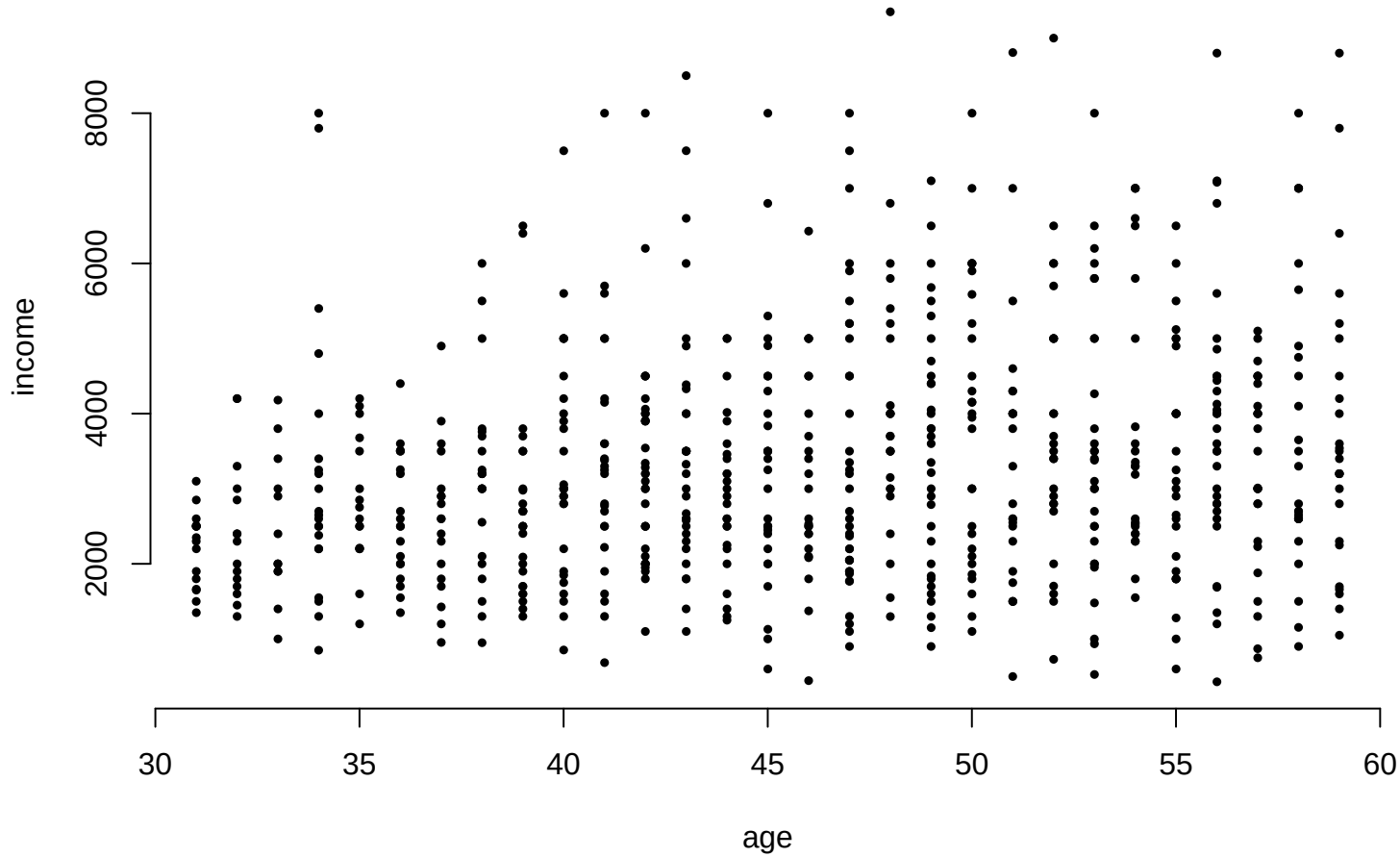
- looks much better!!
- however, linear regression with a quad. predictor would here also be just fine

A second real-data example to further illustrate gamma GLMs

ID	income	age	gender
1	4750	58	Female
2	3254	36	Female
3	6500	54	Male
4	2600	58	Female
5	850	34	Male
⋮	⋮	⋮	⋮
718	4400	36	Male

Table: Income data for 718 individuals in Germany in 2006.

Scatterplot income vs. age



$\rightsquigarrow$ looks like the response distribution is right-skewed...

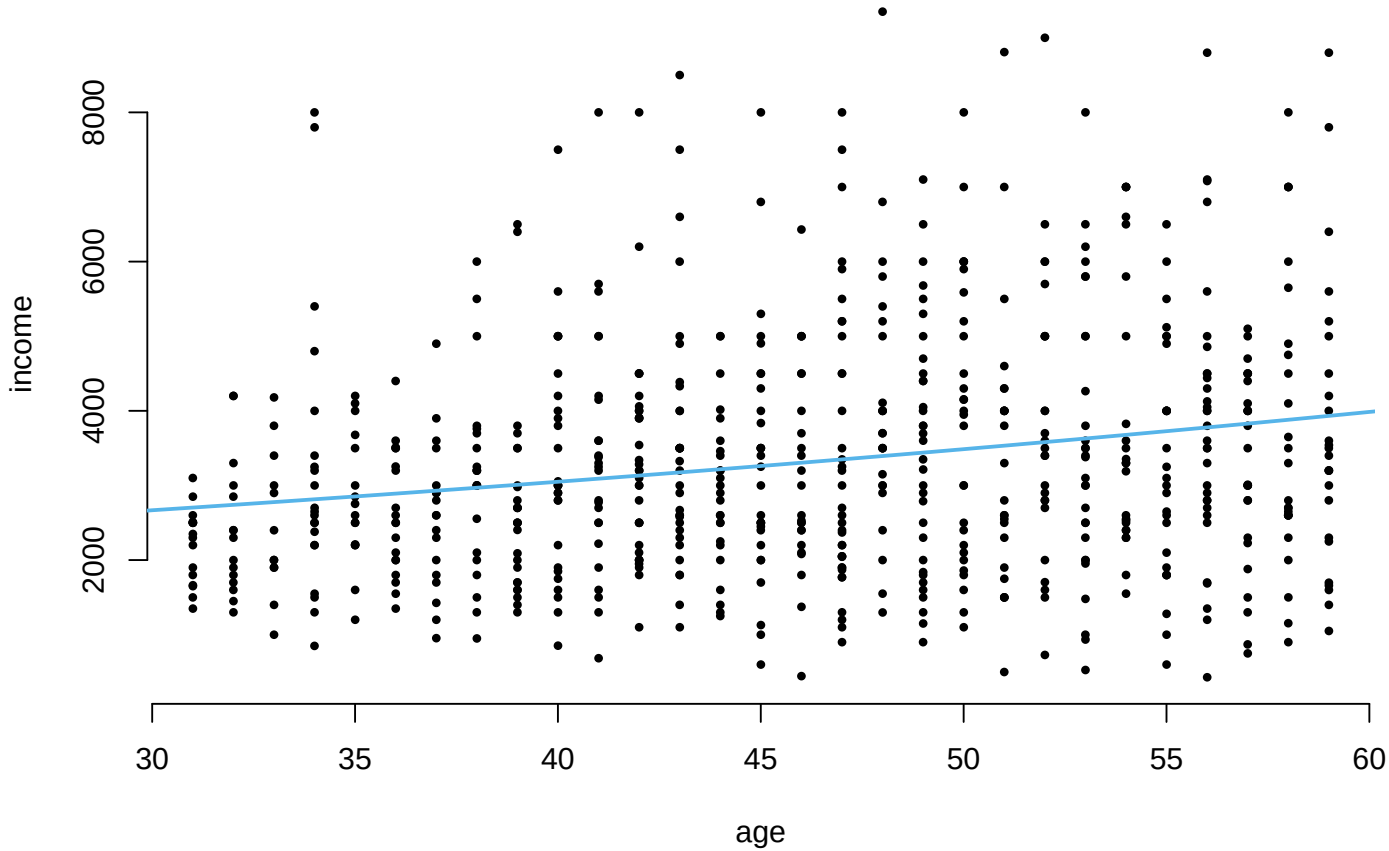
Gamma GLM in the income example (log link)

```
> mod<-glm(income~age,family=Gamma(link="log"))
> mod$coeff
(Intercept)         age
 7.48755268  0.01337986
> summary(mod)
 [...]
 (Dispersion parameter for Gamma family taken to be 0.2218875)
```

The model fitted here is

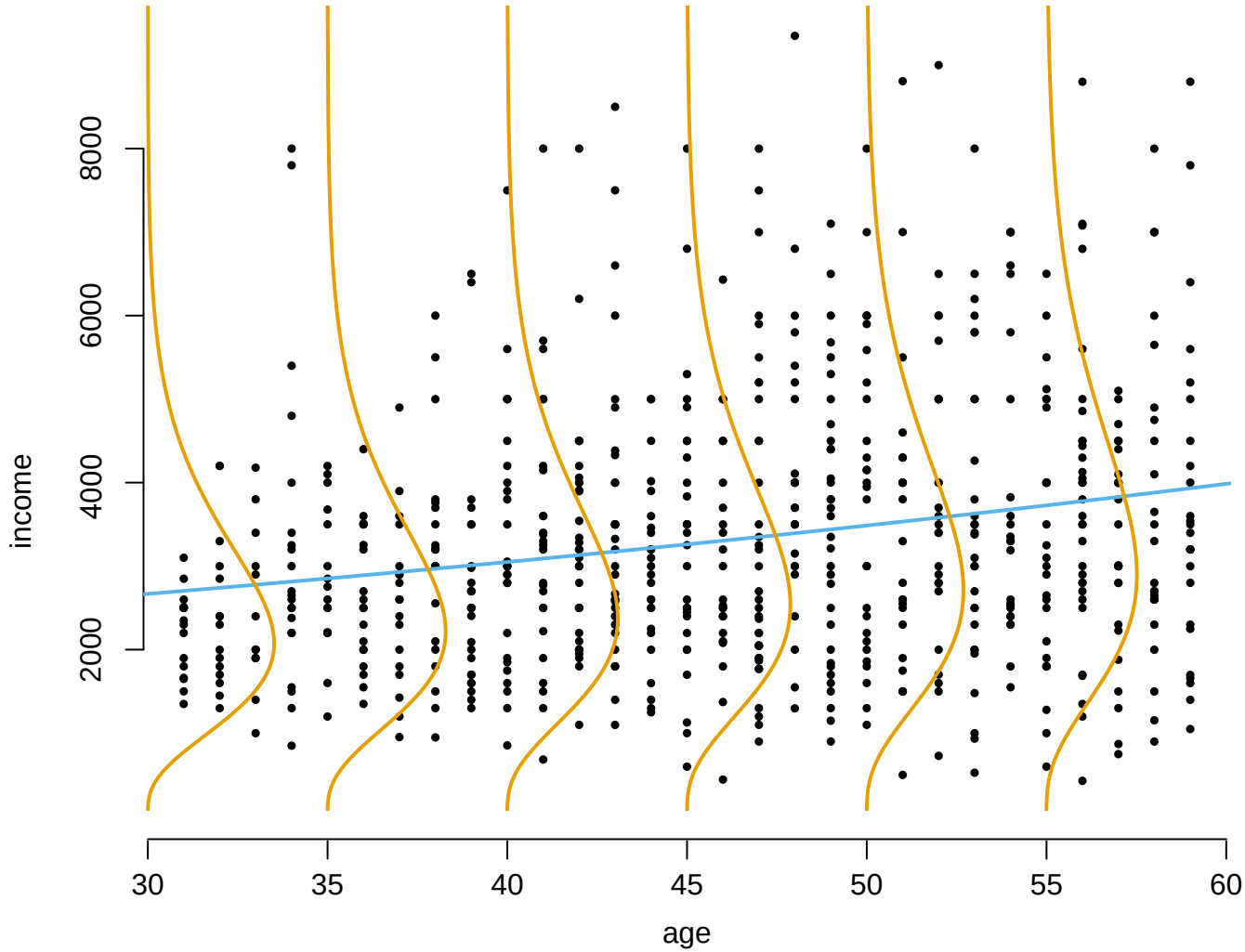
$$\mathbb{E}(\text{income}_i) = e^{7.488+0.0134\cdot\text{age}_i},$$

with price being gamma distributed with shape $\hat{\nu} = \frac{1}{0.222} = 4.507$.

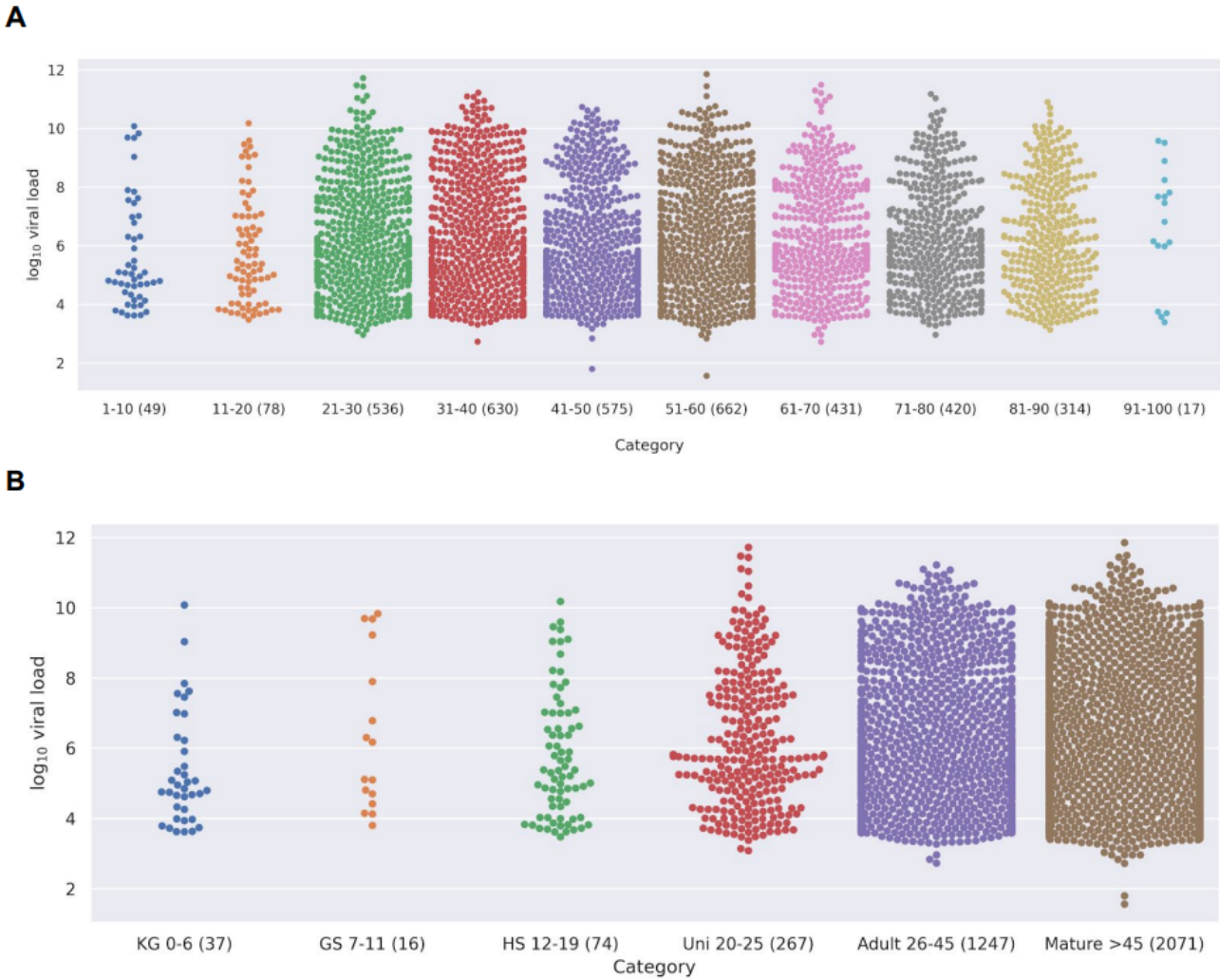


⇝ looks OK

Illustration of the fitted response distribution



Gamma regression in the media: Drosten study, version 1



Media coverage of the Drosten study

From the revised version of the Drosten study

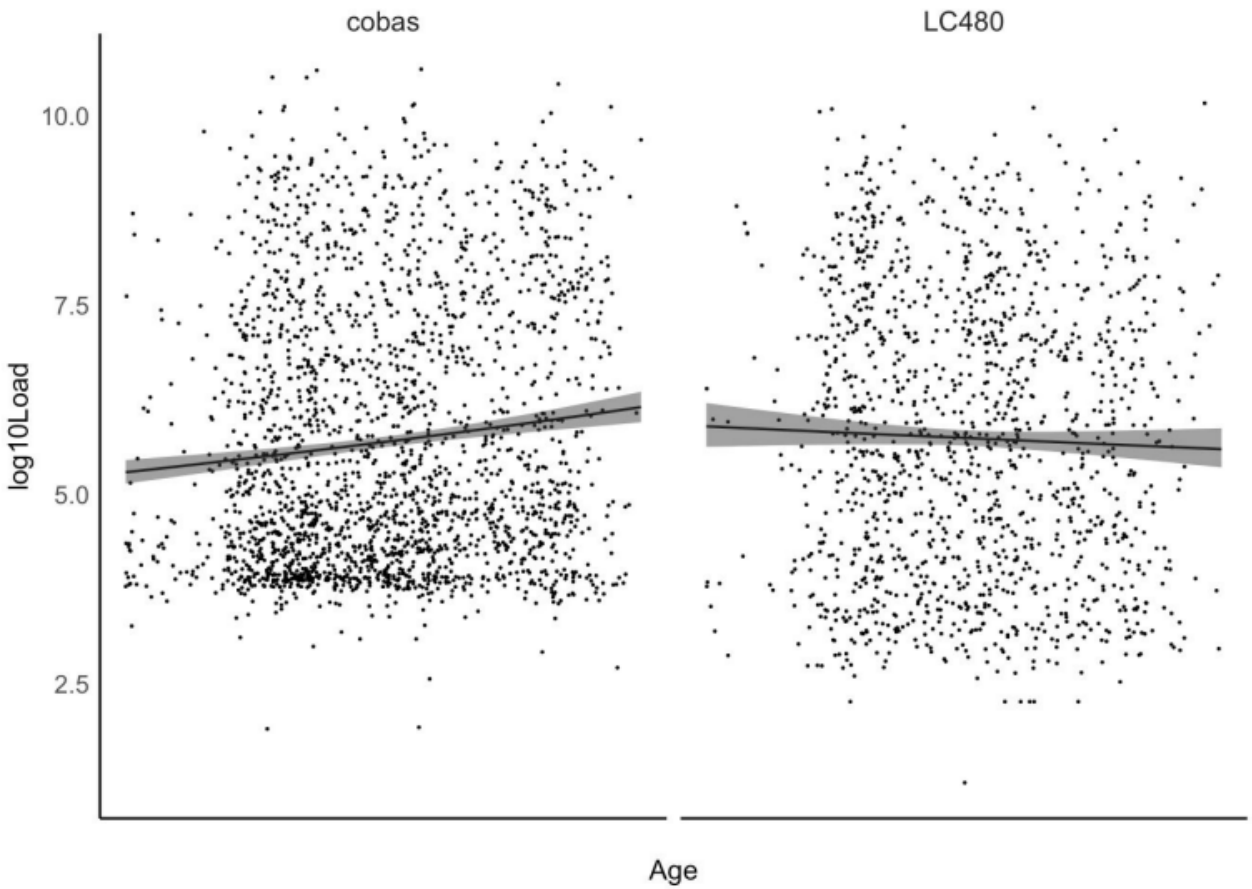


Figure 6: Conditional effect of age from a Bayesian gamma regression predicting viral load from age, while adjusting for type of PCR system (LC480 or cobas).

Current status and what's next

Main learning outcomes so far:

- overview of scenarios in which linear regression won't work well
- how these scenarios can be addressed using the GLM framework:
  - ⇝ flexible distributional assumption for response variable
  - ⇝ use of link function
- the main special cases (normal, Poisson, Bernoulli/binomial, gamma):
  - ⇝ model formulation
  - ⇝ possible link functions
  - ⇝ main properties
- how to fit GLMs in R using the function `glm()`, including the main syntax

Things we want to understand next:

- what's behind `glm()`, i.e. how does the estimation actually work?
- properties of the estimators
- how to select between competing models
- how to check if a model is actually a good model

Chapter 5: Parameter estimation and inference

Chapter 5: Parameter estimation and inference

5.1 Why not simply least squares?

Heteroscedasticity in GLMs

- in a GLM, the variance of the response Y_i is, in general, not constant
  (in linear regression, constant variance was one of the main assumptions)
- for example, in a Poisson GLM, $\mathbb{E}(Y_i) = \text{Var}(Y_i) = e^{\beta_0 + \beta_1 x_{i1}} \neq \text{const.}$

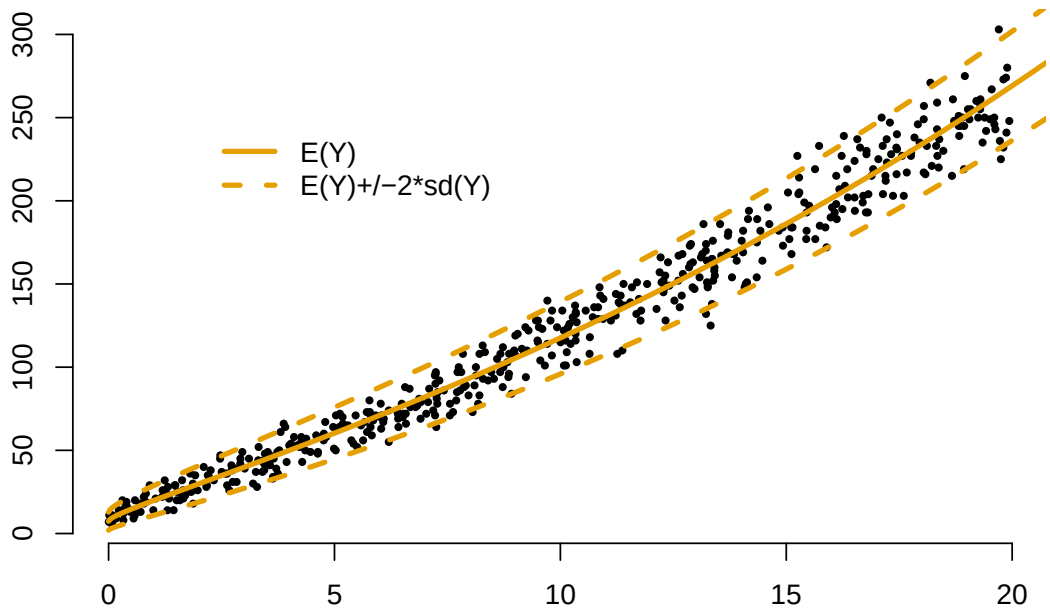


Figure: Data simulated from a Poisson GLM.

The data points should hence be weighted according to their variances[18]:

- high variance $\rightsquigarrow$ little information (large residuals expected) $\rightsquigarrow$ small weight
- low variance $\rightsquigarrow$ much information (small residuals expected) $\rightsquigarrow$ large weight

So we'd like to use **weighted least squares**, but we have a catch–22:

- need to know the variances to calculate weights and hence fit the model...
- ...but need to know the model in order to calculate the variances

Let's start with **maximum likelihood estimation** instead — however, later on, we will see that this actually matches weighted least squares estimation.

[18] put differently, the distances in the sum of squares need to be seen relative to the error variance

Ordinary least squares vs. maximum likelihood

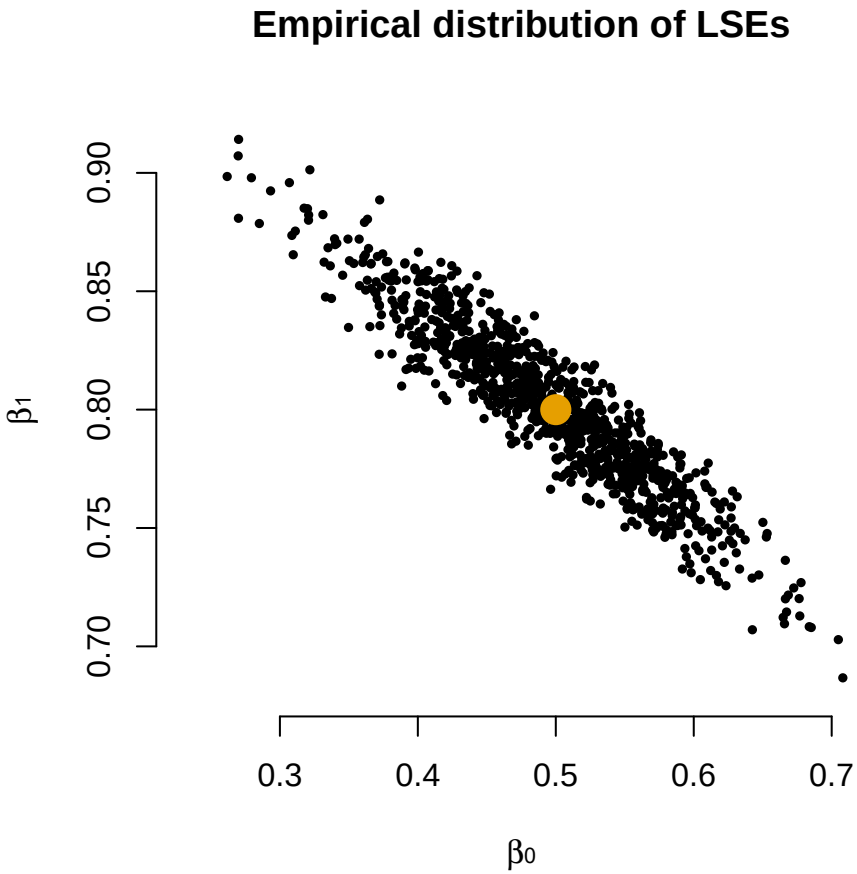
In order to illustrate the non-optimality of ordinary least squares for GLMs, I simulated 1000 data sets from the Poisson GLM

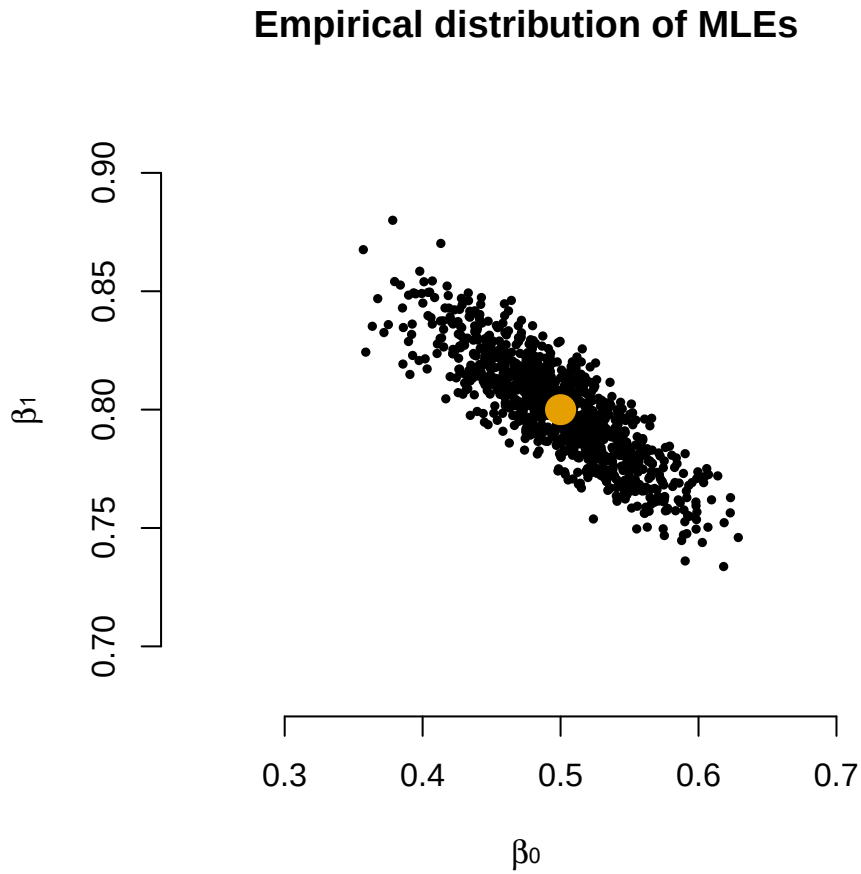
$$\mathbb{E}(Y_i) = e^{\beta_0 + \beta_1 x_i} = e^{0.5 + 0.8 x_{i1}}, \quad i = 1, \ldots, 400.$$

The table shows performance measures of the 1000 ordinary least squares estimates (LSEs) and of the 1000 max. likelihood estimates (MLEs) obtained.

	LSEs		MLEs	
	bias	std. dev.	bias	std. dev.
$\hat{\beta}_0$	-0.004	0.074	-0.001	0.048
$\hat{\beta}_1$	0.002	0.034	0.000	0.022

Both estimators are unbiased, but the precision is much higher for the MLE!


Empirical distribution of LSEs
β1
0.70
0.75
0.80
0.85
0.90
0.3
0.4
0.5
0.6
0.7
β0


Empirical distribution of MLEs
β1
0.70
0.75
0.80
0.85
0.90
0.3
0.4
0.5
0.6
0.7
β0

Chapter 5: Parameter estimation and inference

5.2 Maximum likelihood estimation

Parameter estimation — why maximum likelihood?

- maximum likelihood (ML) estimation is an approach for fitting a model to data (i.e. estimating its parameters)
- key idea: good parameter estimates make the observed data look plausible
- ML estimation: select those parameters for which the model has the highest likelihood[19] of having generated the observed data
- ML estimation...
  - ...is intuitively appealing,
  - ...is practically feasible in many cases,
  - ...and has desirable theoretical properties

[19] = probability, chance

Maximum likelihood estimation — how does it work?

Given data $(y_i, x_{i1}, \ldots, x_{ip})$, $i = 1, \ldots, n$, we regard the joint density/probability of all obs. as a function — the **likelihood function** — of the parameter vector:

$$\mathcal{L}(\boldsymbol{\beta}) = \mathcal{L}(\beta_0, \ldots, \beta_p) = f_{\boldsymbol{\beta}}(y_1, \ldots, y_n)$$

The **maximum likelihood estimate** (MLE) is the vector $\boldsymbol{\beta}$ that maximises $\mathcal{L}(\boldsymbol{\beta})$.

Since log is strictly monotone[20], we have that

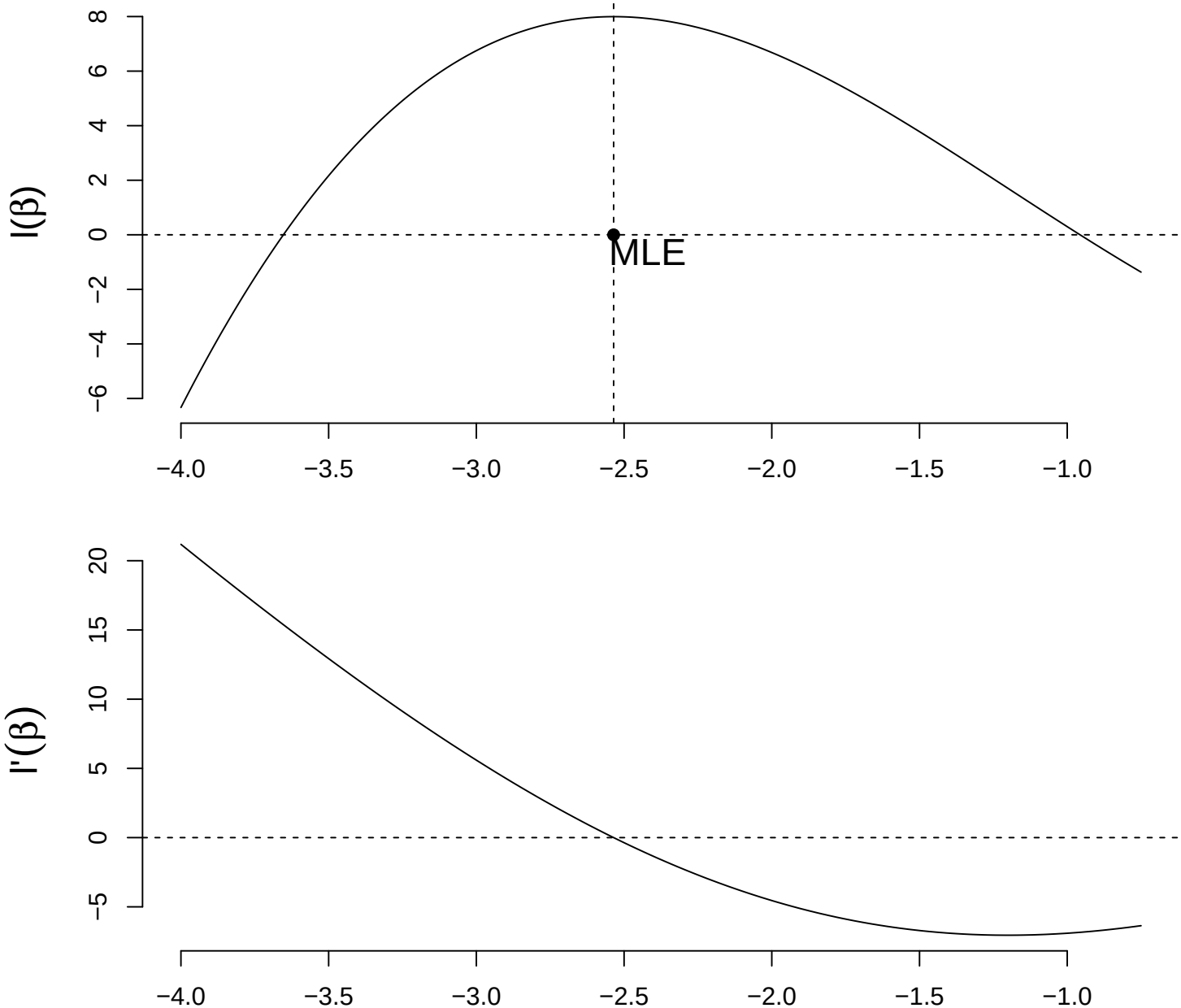
$$\boldsymbol{\beta} \text{ maximises } \mathcal{L}(\boldsymbol{\beta}) \quad \Leftrightarrow \quad \boldsymbol{\beta} \text{ maximises } \ell(\boldsymbol{\beta}) = \log \mathcal{L}(\boldsymbol{\beta})$$

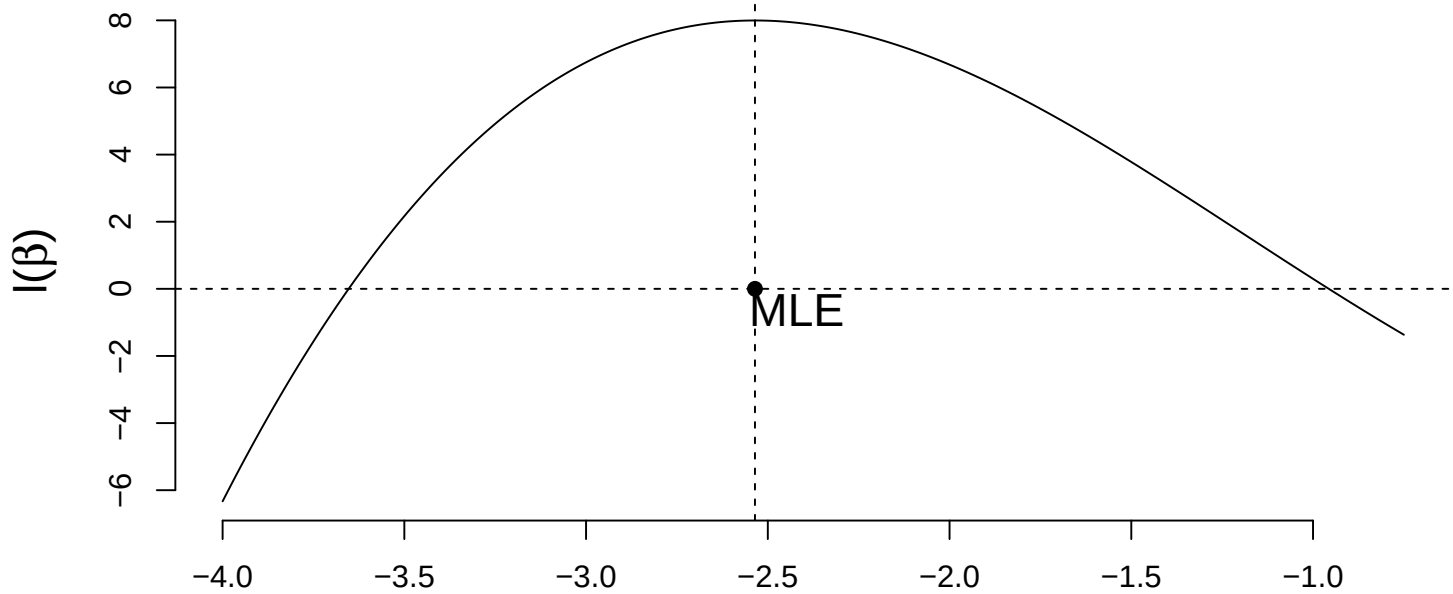
Maximising $\ell(\boldsymbol{\beta})$ is often easier in practice.

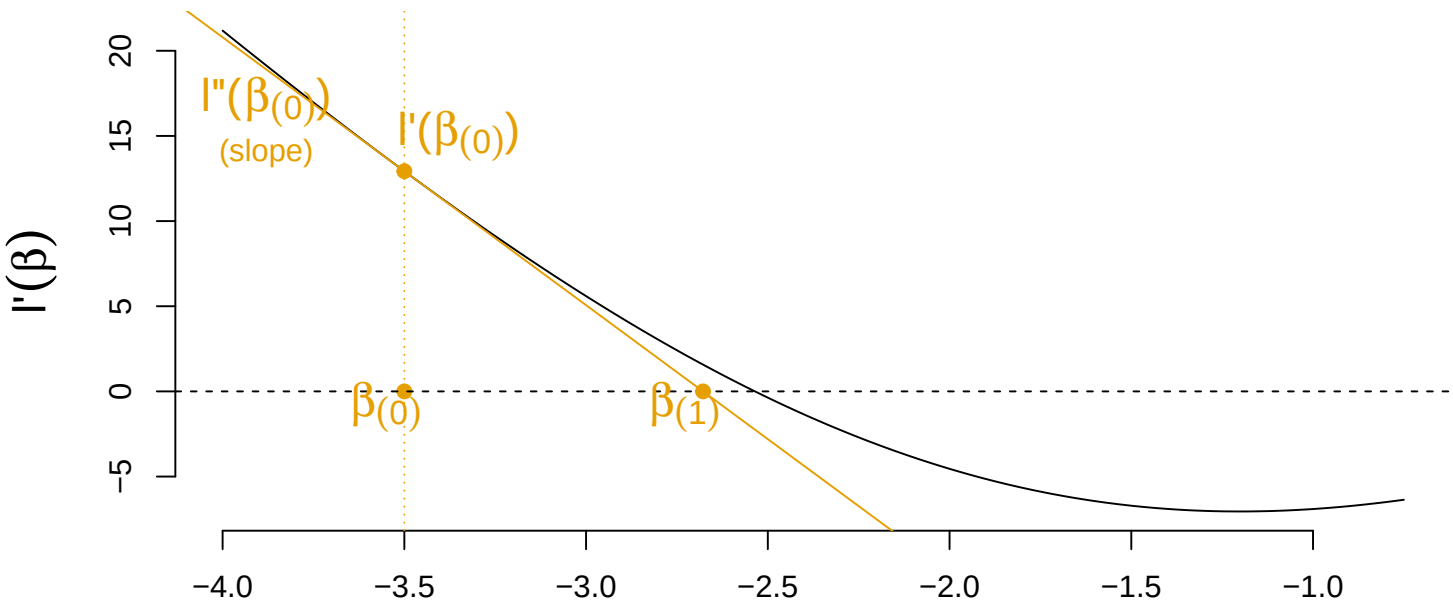
[20] i.e. $x < y \Leftrightarrow \log(x) < \log(y)$

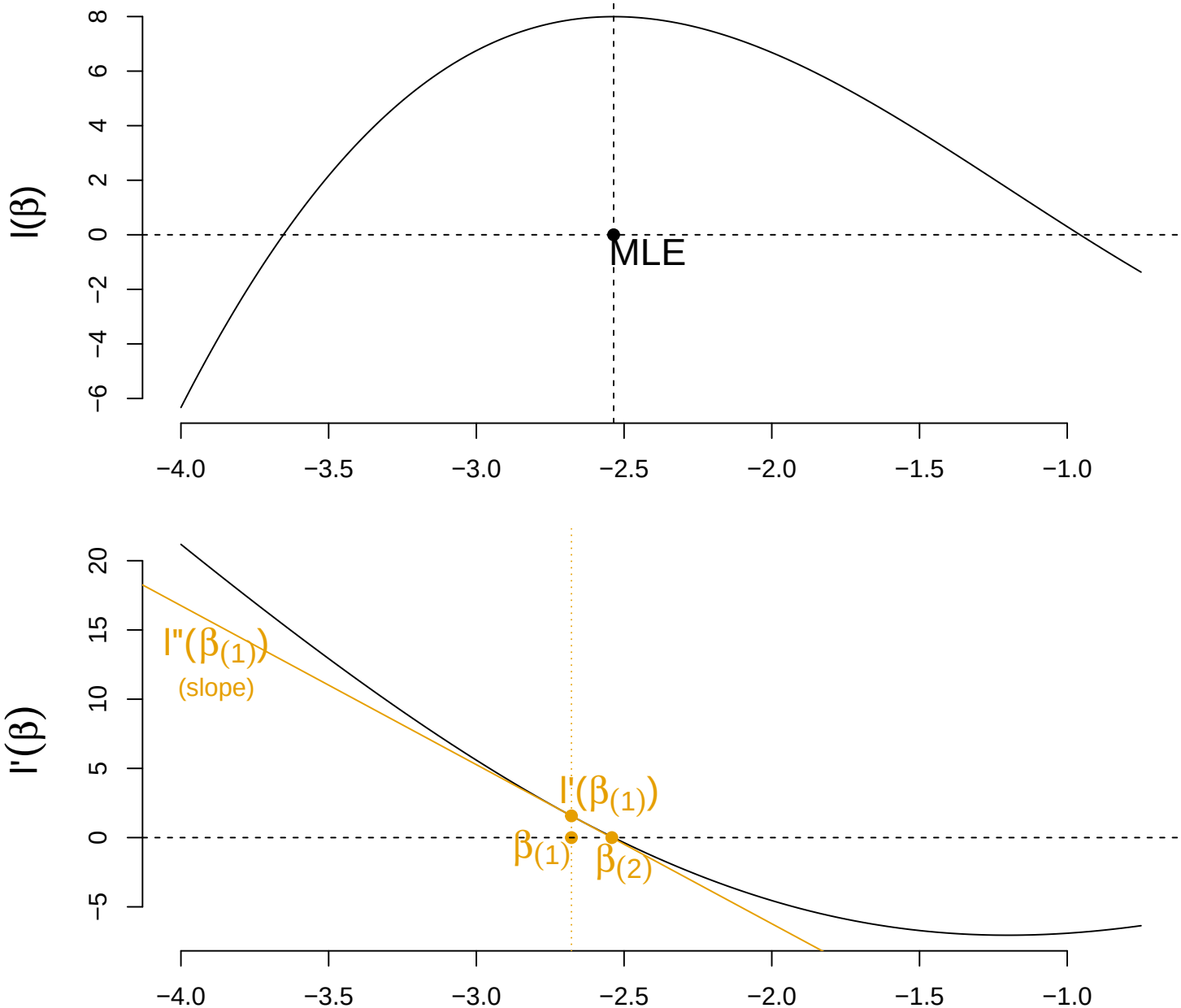
Numerical maximisation of the (log-)likelihood

- there is usually no closed-form (i.e. analytical) solution for the MLE of a GLM
- instead, **numerical search algorithms** are used — general workflow:
  - guess the value of the parameter vector as $\boldsymbol{\beta}_{(0)}$ (initial value)
  - obtain improved guess $\boldsymbol{\beta}_{(1)}$ based on $\boldsymbol{\beta}_{(0)}$
  - obtain improved guess $\boldsymbol{\beta}_{(2)}$ based on $\boldsymbol{\beta}_{(1)}$
  - ...
  - terminate algorithm when changes in $\ell(\boldsymbol{\beta})$ are negligible
- on the next slides, we *graphically* illustrate the **Newton-Raphson method**


l(β)
8
6
4
2
0
−2
−4
−6
MLE
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
l'(β)
20
15
10
5
0
−5
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0


l(β)
8
6
4
2
0
−2
−4
−6
MLE
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0


l'(β)
20
15
10
5
0
−5
l"(β(0))
(slope)
l'(β(0))
β(0)
β(1)
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0


l(β)
8
6
4
2
0
−2
−4
−6
MLE
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
l'(β)
20
15
10
5
0
−5
l"(β(1))
(slope)
l'(β(1))
β(1)
β(2)
−4.0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0

Analytic derivation of Newton-Raphson

For simplicity, let's consider a univariate parameter β. We then need to solve

$$\frac{\partial \ell}{\partial \beta} = \ell'(\beta) = 0$$

A Taylor expansion about an initial guess $\beta_{(0)}$ gives

$$0 = \ldots$$

which gives us

$$\ldots$$

$$\beta \approx \beta_{(1)} = \beta_{(0)} - \frac{\ell'(\beta_{(0)})}{\ell''(\beta_{(0)})}.$$

- given a guessed $\beta_{(0)}$ for the MLE, this update gives us an improved $\beta_{(1)}$
- from $\beta_{(1)}$, we then calculate $\beta_{(2)}$, from which we calculate $\beta_{(3)}$, etc.
- this repeated application of the update is the Newton-Raphson method
- the algorithm stops when $\ell'(\beta_{(r)}) \approx 0$
- in general, this may converge to a local rather than the global maximum!
- however, for GLMs, the likelihood function is strictly concave — such that there are no local maxima — when using the canonical link function

For a univariate β, the **Newton-Raphson method** involves the update

$$\beta_{(r+1)} = \beta_{(r)} - \frac{\ell'(\beta_{(r)})}{\ell''(\beta_{(r)})}, \quad r = 0, 1, 2, \ldots.$$

For a parameter *vector* $\boldsymbol{\beta}$, the Newton-Raphson method looks as follows:

$$\boldsymbol{\beta}_{(r+1)} = \boldsymbol{\beta}_{(r)} - \mathbf{H}(\boldsymbol{\beta}_{(r)})^{-1} \frac{\partial \ell}{\partial \boldsymbol{\beta}_{(r)}}, \quad r = 0, 1, 2, \ldots,$$

where $\mathbf{H}(\boldsymbol{\beta}_{(r)})$ is the Hessian matrix of the log-likelihood function.

Terminology and notation (for given $\boldsymbol{\beta}$):

- the gradient $S(\boldsymbol{\beta}) = \left(\frac{\partial \ell}{\partial \beta_0}, \ldots, \frac{\partial \ell}{\partial \beta_p}\right)$ is called **score statistic**
- $\mathcal{J}(\boldsymbol{\beta}) = -\mathbf{H}(\boldsymbol{\beta})$ is the **observed Fisher information**

With this notation, the scheme becomes

$$\boldsymbol{\beta}_{(r+1)} = \boldsymbol{\beta}_{(r)} + \mathcal{J}(\boldsymbol{\beta}^{(r)})^{-1} S(\boldsymbol{\beta}^{(r)})$$

In practice, $\mathcal{J}(\boldsymbol{\beta})$ is often replaced by the **expected Fisher information**, $\mathcal{I}(\boldsymbol{\beta}) = \mathbb{E}\big(\mathcal{J}(\boldsymbol{\beta})\big)$, leading to the scheme:

$$\boldsymbol{\beta}^{(r+1)} = \boldsymbol{\beta}^{(r)} + \mathcal{I}(\boldsymbol{\beta}^{(r)})^{-1} S(\boldsymbol{\beta}^{(r)}).$$

This is the so-called **method of scoring**.

Possible advantages:

- expected Fisher information is often easier to calculate — in particular[21]:

$$\mathcal{I}(\boldsymbol{\beta}) = \mathbb{E}\left[\left(\frac{\partial \ell}{\partial \boldsymbol{\beta}}\right)\left(\frac{\partial \ell}{\partial \boldsymbol{\beta}}\right)^t\right],$$

  in other words information on *first* derivatives is sufficient!
- better numerical properties/more stable

[21] this is in fact a standard result in ML theory — covered e.g. in "Foundations of Statistical Inference"

Towards a general strategy for fitting GLMs

Making use of the exponential family form, we will now:

- derive the score statistic $S(\boldsymbol{\beta})$ for a (general) GLM
- (from this) derive the expected Fisher information $\mathcal{I}(\boldsymbol{\beta})$ for a (general) GLM

With these at hand, we can then implement the method of scoring:

1. choose initial value $\boldsymbol{\beta}$
2. while $S(\boldsymbol{\beta}) \neq \mathbf{0}$, repeat

$$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \mathcal{I}(\boldsymbol{\beta})^{-1} S(\boldsymbol{\beta})$$

3. return $\boldsymbol{\beta}$

Chapter 5: Parameter estimation and inference

5.3 The GLM likelihood and its numerical maximisation

Likelihood of a GLM

For a GLM,

$$g(\mu_i) = g\big(\mathbb{E}(Y_i)\big) = \eta_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip},$$

with $f_{\theta_i}(y_i) = \exp\big(y_i b(\theta_i) + c(\theta_i) + d(y_i)\big)$, the log-likelihood is given by

$$\begin{aligned}
\ell(\boldsymbol{\beta}) = \log \mathcal{L}(\boldsymbol{\beta}) &= \log f_{\boldsymbol{\beta}}(y_1, \ldots, y_n) \\
&= \log \prod_{i=1}^{n} f_{\boldsymbol{\beta}}(y_i) \\
&= \log \prod_{i=1}^{n} \exp\big(y_i b(\theta_i) + c(\theta_i) + d(y_i)\big) \\
&= \sum_{i=1}^{n} \underbrace{\big(y_i b(\theta_i) + c(\theta_i) + d(y_i)\big)}_{=\ell_i}
\end{aligned}$$

Likelihood of a GLM

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \big(y_i b(\theta_i) + c(\theta_i) + d(y_i)\big)$$

To see that $\ell(\boldsymbol{\beta})$ is indeed a function of $\boldsymbol{\beta}$, note that:

- $\boldsymbol{\beta}$ determines the vector of linear predictors $\boldsymbol{\eta}$...
- ...which in turn determines the vector of expected values $\boldsymbol{\mu}$...
- ...which in turn determines the vector $\boldsymbol{\theta}$ appearing in the exp. family form[22]...
- ...based on which the above expression can be calculated

Thus,

$$\ell(\boldsymbol{\beta}) = \ell\Big(\boldsymbol{\theta}\Big(\boldsymbol{\mu}\big(\boldsymbol{\eta}(\boldsymbol{\beta})\big)\Big)\Big)$$

[22] where usually, but not always, $\mu_i = \theta_i$

Score statistic and expected Fisher information for GLMs

For a GLM specified as on slide 164, let

$$S_j = \frac{\partial \ell}{\partial \beta_j} \quad \text{and} \quad \mathcal{I}_{jk} = -\mathbb{E}\,\frac{\partial^2 \ell}{\partial \beta_j \partial \beta_k}, \quad \text{for } j, k = 0, 1, \ldots, p,$$

here omitting the dependence on $\boldsymbol{\beta}$ for notational simplicity. Defining $x_{i0} = 1$ for all i, we have

(i) $S_j = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\mathrm{var}(Y_i)} x_{ij} \frac{\partial \mu_i}{\partial \eta_i}$

(ii) $\mathcal{I}_{jk} = \sum_{i=1}^{n} \frac{x_{ij} x_{ik}}{\mathrm{var}(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2$

Proof of (i) (in class)

Proof of (i), continued (in class)

Proof of (ii) (in class)

Score statistic and expected Fisher information for GLMs

Defining a diagonal matrix $\mathbf{W}$ with entries $w_{ii} = \frac{1}{\text{var}(Y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2$, we can write

$$\mathcal{I} = \begin{pmatrix} \mathcal{I}_{00} & \dots & \mathcal{I}_{0p} \\ \vdots & \ddots & \vdots \\ \mathcal{I}_{p0} & \dots & \mathcal{I}_{pp} \end{pmatrix} = \mathbf{X}^t \mathbf{W} \mathbf{X},$$

where $\mathbf{X}$ is the design matrix of the GLM.

Again utilising the w_{ii}, we further obtain

$$S_j = \sum_{i=1}^{n} x_{ij} w_{ii} (y_i - \mu_i) \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^{-1},$$

such that

$$S(\boldsymbol{\beta}) = \frac{\partial \ell}{\partial \boldsymbol{\beta}} = (S_0, \dots, S_p)^t = \mathbf{X}^t \mathbf{W} \boldsymbol{\nu},$$

where $\nu_i = (y_i - \mu_i) \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^{-1}$.

Method of scoring for GLMs

For the method of scoring, we obtain:

$$\begin{aligned}
\boldsymbol{\beta}^{(r+1)} &= \boldsymbol{\beta}^{(r)} + \mathcal{I}(\boldsymbol{\beta}^{(r)})^{-1} S(\boldsymbol{\beta}^{(r)}) \\
&= \boldsymbol{\beta}^{(r)} + \left(\mathbf{X}^t \mathbf{W} \mathbf{X}\right)^{-1} \mathbf{X}^t \mathbf{W} \boldsymbol{\nu} \\
&= \left(\mathbf{X}^t \mathbf{W} \mathbf{X}\right)^{-1} \mathbf{X}^t \mathbf{W} \mathbf{X} \boldsymbol{\beta}^{(r)} + \left(\mathbf{X}^t \mathbf{W} \mathbf{X}\right)^{-1} \mathbf{X}^t \mathbf{W} \boldsymbol{\nu} \\
&= \left(\mathbf{X}^t \mathbf{W} \mathbf{X}\right)^{-1} \mathbf{X}^t \mathbf{W} \left(\mathbf{X} \boldsymbol{\beta}^{(r)} + \boldsymbol{\nu}\right) \\
&= \left(\mathbf{X}^t \mathbf{W} \mathbf{X}\right)^{-1} \mathbf{X}^t \mathbf{W} \mathbf{z}, \qquad (2)
\end{aligned}$$

where $z_i = \eta_i + (y_i - \mu_i) \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^{-1}$.

Equation (2) is used in order to set up an **iterative scheme**:

- use some initial approximation $\boldsymbol{\beta}^{(0)}$ (e.g. the LS estimate)
- compute an improved $\boldsymbol{\beta}^{(1)}$ based on $\boldsymbol{\beta}^{(0)}$ using (2)
- compute an improved $\boldsymbol{\beta}^{(2)}$ based on $\boldsymbol{\beta}^{(1)}$ using (2)
- ...
- (repeat until convergence)

The updating scheme

$$\boldsymbol{\beta}^{(r+1)} = \left(\mathbf{X}^t\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^t\mathbf{W}\mathbf{z},$$

with $\mathbf{W}$ and $\mathbf{z}$ as just defined, is called **iteratively reweighted least squares** (IRLS) algorithm — a special case of the method of scoring.

Thus, in the end of the day, we see that ML estimation for GLMs boils down to what we intuitively expected in the first place: weighted least squares!

IRLS for Poisson GLMs

As an example, consider the basic Poisson GLM,

$$\log(\mu_i) = \beta_0 + \beta_1 x_{i1}$$

In this case, we have

$$w_{ii} =$$

$$z_i =$$

```
> set.seed(1)
> x<-runif(100,-3,3)
> y<-rpois(100,exp(0.5+0.8*x))
> X<-cbind(rep(1,length(x)),x)
>
> # function that, for given beta and x, returns W
> W<-function(beta,x){
+   eta<-beta[1]+beta[2]*x
+   return(diag(exp(eta)))
+ }
>
> # function that, for given beta, x and y, returns z
> z<-function(beta,x,y){
+   eta<-beta[1]+beta[2]*x
+   eta+y/exp(eta)-1
+ }
>
> mean(y)
[1] 3.7
> beta<-c(log(3.7),0) # initial guess for ML estimate
> for (iter in 2:10){ # run IRLS as a loop
+   beta<-as.vector(solve(t(X)%*%W(beta,x)%*%X)%*%t(X)%*%W(beta,x)%*%z(beta,x,y))
+   print(beta)
+ }
[1] 1.2426273 0.6135974
[1] 0.7238925 0.7336630
[1] 0.5101647 0.8146078
[1] 0.4792807 0.8284501
[1] 0.4787140 0.8287070
[1] 0.4787138 0.8287071
[1] 0.4787138 0.8287071
[1] 0.4787138 0.8287071
[1] 0.4787138 0.8287071
```

IRLS for the basic Gaussian GLM

For some GLMs, the scheme simplifies. For example, for a Gaussian response and the identity link, we have

$$w_{ii} = \frac{1}{\sigma^2} \qquad \text{and} \qquad z_i = y_i,$$

such that

$$\beta = \left(\mathbf{X}^t\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^t\mathbf{W}\mathbf{z} = \left(\mathbf{X}^t\mathbf{X}\right)^{-1}\mathbf{X}^t\mathbf{y}$$

In this case, we can of course find the solution analytically — it's simply the LSE!

Chapter 5: Parameter estimation and inference

5.4 Estimator properties and uncertainty quantification

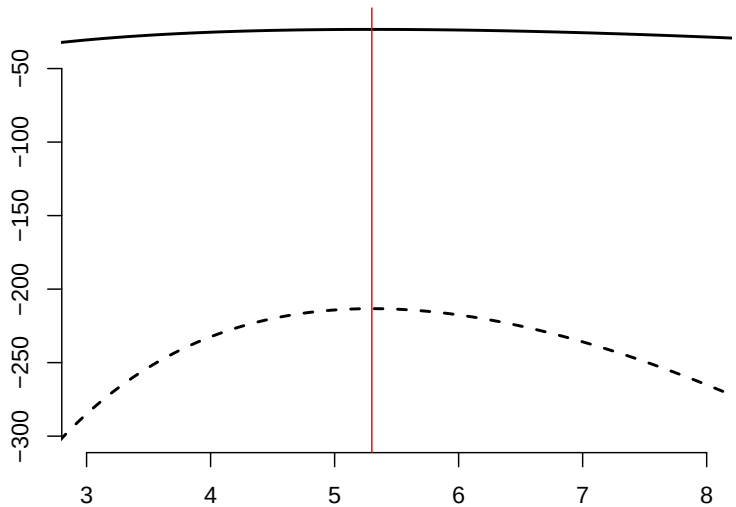
Inference — overview

- we now know how to fit a GLM to data, i.e. how to find the MLE $\hat{\boldsymbol{\beta}}$ of the parameter vector $\boldsymbol{\beta}$ (with `glm()` in R doing all the hard work for us)
- in practice, we usually also want to quantify the **uncertainty in** $\hat{\boldsymbol{\beta}}$
- standard MLE theory can be applied in order to find the (approximate) distribution of $\hat{\boldsymbol{\beta}}$, based on which we can:
  - calculate CIs
  - conduct hypothesis tests
- we won't provide proofs that all regularity conditions are fulfilled — these are technical and not very interesting

How to measure uncertainty? (simple motivating example)

Suppose that $Y_1, \ldots, Y_n \overset{\text{iid}}{\sim} Po(\lambda)$. The plot below shows the log-likelihood, $\ell(\lambda)$,

- for $y_1, \ldots, y_{10}$ with $\bar{y} = 5.3$ (solid line)
- for $y_1, \ldots, y_{100}$ with $\bar{y} = 5.3$ (dashed line)



- the amount of information and hence the *curvature* are higher for $n = 100$
- the Fisher information is a measure of the curvature!

Asymptotic behaviour and approximate distribution of the MLE

MLE theory states that for large sample sizes (i.e. $n \to \infty$),

- $\hat{\boldsymbol{\beta}}$ will approximately follow a multivariate normal distribution[23]
- the MLE is approximately unbiased
- the variance-covariance matrix is the inverse expected Fisher information

Putting it all together:

> For large n, the approximate distribution of the MLE of the GLM parameters is obtained as
>
> $$\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \mathcal{I}^{-1}),$$
>
> where $\mathcal{I} = \mathbf{X}^t\mathbf{W}\mathbf{X}$ is the expected Fisher information at the MLE.

[23] so that if we kept drawing samples from the true model and re-calculating $\hat{\boldsymbol{\beta}}$ for each new sample, then the $\hat{\boldsymbol{\beta}}$s would be normally distributed

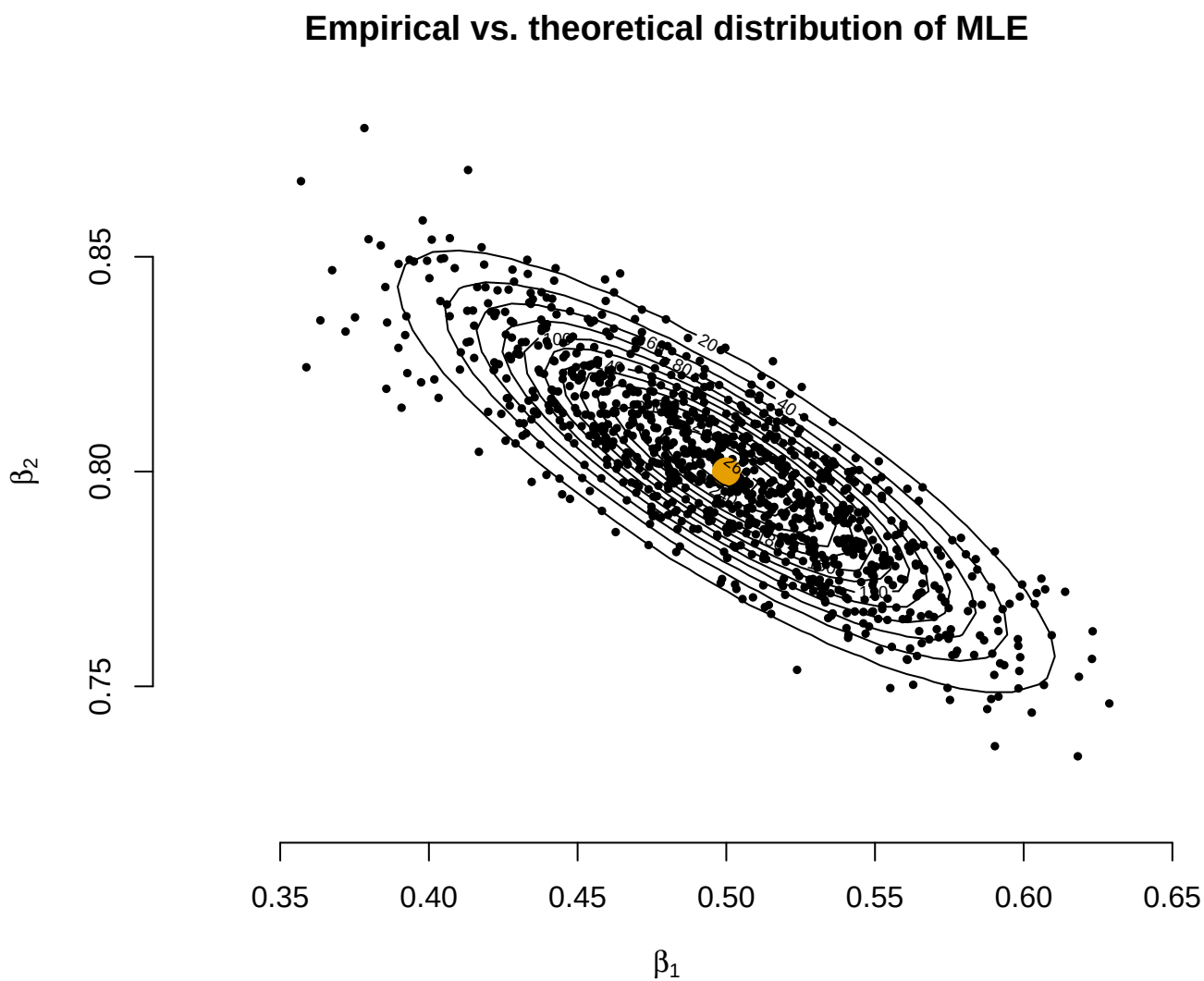


Figure: MLEs obtained for 500 data sets simulated from the Poisson GLM where $\mathbb{E}(Y_i) = e^{0.5+0.8x_{i1}}$, $i = 1, \ldots, 400$, and theoretical distribution of the MLE (contour lines).

Confidence intervals

For $n \to \infty$, and letting Σ_{jj} denote the $(j+1)$–th diagonal element of $\mathcal{I}^{-1}$ and z_α the α–quantile of the standard normal distribution, we thus have: (in class)

For reasonably large n,

$$\left[\hat{\beta}_j + z_{0.025}\sqrt{\Sigma_{jj}};\ \hat{\beta}_j + z_{0.975}\sqrt{\Sigma_{jj}}\right]$$

is an approximate 95% **confidence interval** for β_j.

Hypothesis testing

Suppose that we want to test $H_0 : \beta_j = 0$ against $H_1 : \beta_j \neq 0$. Then: (in class)

For reasonably large n, the following decision rule yields an approximate significance test of $H_0 : \beta_j = 0$ against $H_1 : \beta_j \neq 0$, at level α:

$$|Z| = \left|\hat{\beta}_j / \sqrt{\Sigma_{jj}}\right| > z_{1-\alpha/2} \quad \rightsquigarrow \quad \text{reject } H_0$$

$$|Z| = \left|\hat{\beta}_j / \sqrt{\Sigma_{jj}}\right| \leq z_{1-\alpha/2} \quad \rightsquigarrow \quad \text{retain } H_0$$

Confidence intervals and hypothesis testing in R

```
glm(formula = survival ~ age, family = binomial)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.81733    0.37549   2.177   0.0295 *
age         -0.03237    0.01509  -2.145   0.0320 *
```

- `summary(glm(...))` in R provides us with parameter estimates and standard errors[24], such that confidence intervals can easily be calculated
- it also gives us the test statistic Z and the corresponding p–value

[24] the standard errors given in the output here are $\sqrt{\Sigma_{00}}$ and $\sqrt{\Sigma_{11}}$, respectively

GLMs with nuisance parameters

- for small sample sizes, the above CIs and hypothesis tests should be used only if Σ_{jj} doesn't involve an additional unknown nuisance parameter
- e.g. linear models and gamma GLMs involve nuisance parameters
- if there is an additional unknown nuisance parameter, then the quantiles of the standard normal need to be replaced by those of the $t_{n-(p+1)}$ distribution, just like for linear models — this is done automatically in R:

```
glm(formula = income ~ education, family = Gamma(link = "log"))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.47987    0.18392  40.670  < 2e-16 ***
education    0.11982    0.01661   7.216 1.06e-10 ***
```

- but note that the $t_{n-(p+1)}$ distribution converges to the $\mathcal{N}(0,1)$ distribution, such that for large n it doesn't make a difference which quantiles are used

Uncertainty quantification based on a parametric bootstrap

- if we could repeatedly simulate new data from the true model, then we could calculate the MLEs for very many simulated data sets, and quantify estimation uncertainty based on the variation of the MLEs obtained

- but we don't know the true model...
- bootstrap idea: assume that the fitted model is a good approximation of the true model and use it to investigate the behaviour of the estimator
- more specifically, simulate data from fitted model and refit model to simulated data
- repeat this lots of times (e.g. 999 times) and estimate standard errors and confidence intervals from the sample of estimates

Uncertainty quantification based on (percentile) bootstrapping

```
> mod<-glm(survival~age,family=binomial)
>
> betas<-matrix(NA,999,2)
>
> for (boot in 1:999){
+  surv.sim<-rbinom(85,size=1,prob=plogis(mod$coeff[1]+mod$coeff[2]*age))
+  mod.boot<-glm(surv.sim~age,family=binomial)
+  betas[boot,]<-as.numeric(mod.boot$coeff)
+ }
>
> apply(betas,2,sd)
[1] 0.39653442 0.01631586
>
> sorted.betas<-apply(betas,2,sort)
> sorted.betas[c(25,975),]
           [,1]        [,2]
[1,] 0.08964336 -0.06898677
[2,] 1.66009397 -0.00299247
```

Standard errors using asymp. results: 0.37549 for β_0, 0.01509 for β_1

CIs using asympt. results: $[0.081, 1.553]$ for β_0, $[-0.062, -0.003]$ for β_1

Uncertainty in mean prediction — motivation

We've been considering **predictions of the mean of the response** all the time: for a GLM

$$g(\mu_i) = \eta_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip},$$

and any given covariate values $x_1, \ldots, x_p$, we predict

$$\hat{\mu} = g^{-1}(\hat{\eta}) = g^{-1}(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p)$$

- this can easily be done manually
- alternatively, `predict(mod,...)` in R gives the $\hat{\eta}$, from which the $\hat{\mu}$ can easily be obtained by applying the inverse link function (e.g. `plogis(predict(mod,newdata=data.frame(age=20)))`)
- `predict(mod,type="response",...)` directly gives the $\hat{\mu}$

But how can we **quantify the uncertainty in the predicted mean**?

Uncertainty in mean prediction — Option 1

- the easiest way to obtain an uncertainty quantification for the prediction of the mean is as follows:
  - observe that, based on the approximate normality of the estimators, the predictor η is also approximately normally distributed (for given covariate values)
  - calculate the corresponding confidence interval for the predictor
  - transform this CI using the inverse link function
- example logistic regression in R:

```
pre<-predict(mod,se.fit=T,newdata=data.frame(x=...))
```

then[25]

```
plogis(pre$fit+qnorm(c(0.025,0.975))*pre$se.fit})
```

- note that this gives a CI for the mean, not for the actual observation!

[25] noting that `predict` by default gives predictions for the linear predictor, not the response

Uncertainty in mean prediction — Option 2

- the CIs from the previous slide will, in general, not be symmetric around the predicted mean (which is no problem except that some people don't like it...)
- to obtain symmetric CIs, we need to translate the standard error estimates for the $\hat{\beta}$'s into a standard error estimate for $\hat{\mu}$
- this can be achieved using the **delta method**[26] — details not provided here
- in R:

```
pre<-predict(mod,type="response",se.fit=T,newdata=...)
```

gives both point prediction and standard error for $\hat{\mu}$ (obtained via delta method), from which confidence intervals can easily be calculated:

```
pre$fit+qnorm(c(0.025,0.975))*pre$se.fit
```

[26] which, for an approx. normally distributed $\boldsymbol{\beta}$ gives the variance of the approx. normal dist. of $f(\boldsymbol{\beta})$

Uncertainty in mean prediction — Option 3

For small sample sizes, we may not want to trust the asymptotic theory.

In such a case we can use a bootstrap:

1. simulate data from fitted model and refit model to simulated data
2. predict mean under model fitted to simul. data (at covariate value of interest)
3. repeat 1. and 2. lots of times and obtain approximate confidence intervals from the sample of predicted means

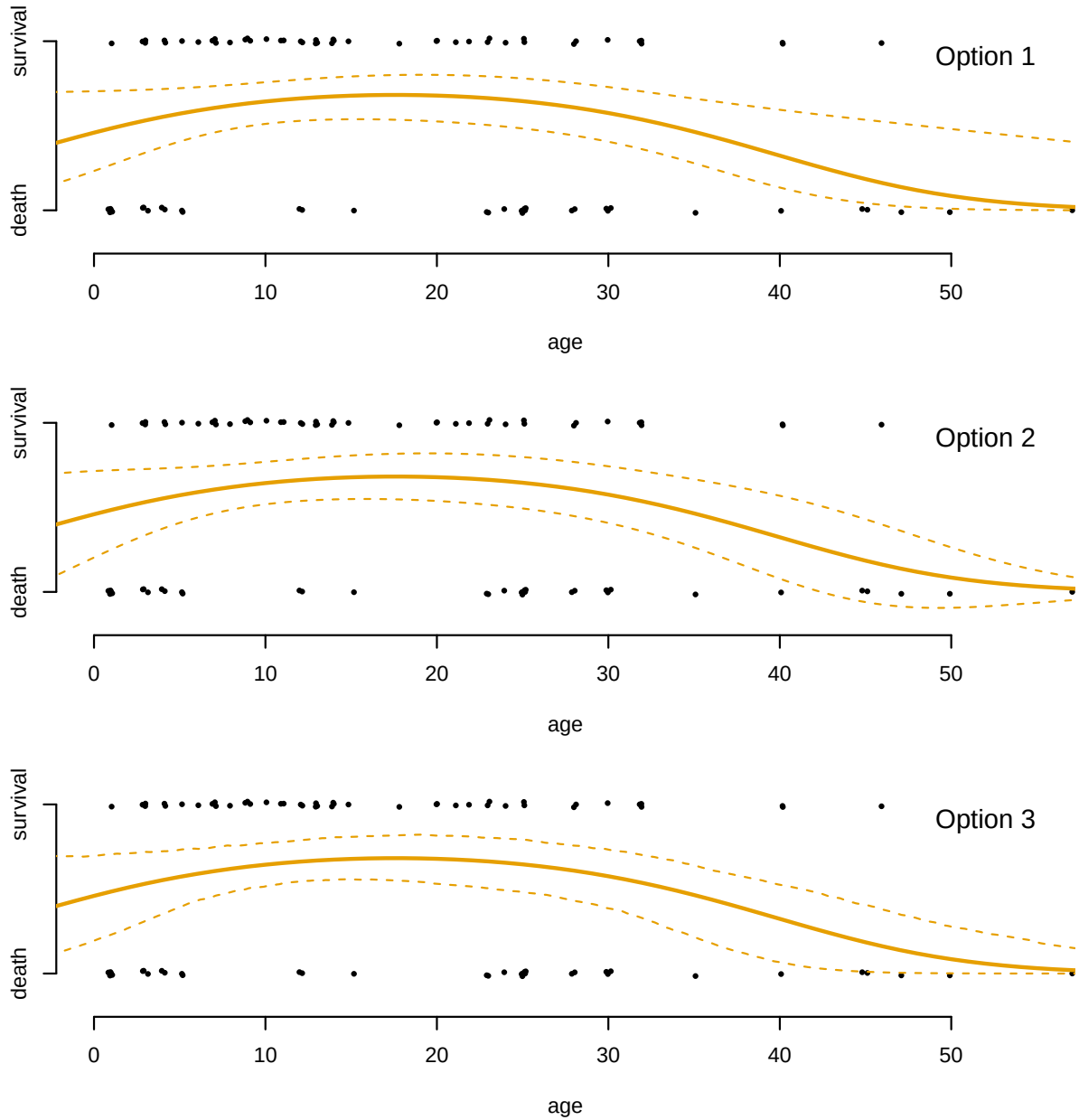


Figure: Survival prob. with 95% CIs, $\text{logit}(\mathbb{E}(\text{survival}_i)) = \beta_0 + \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{age}_i^2$.

Outlook

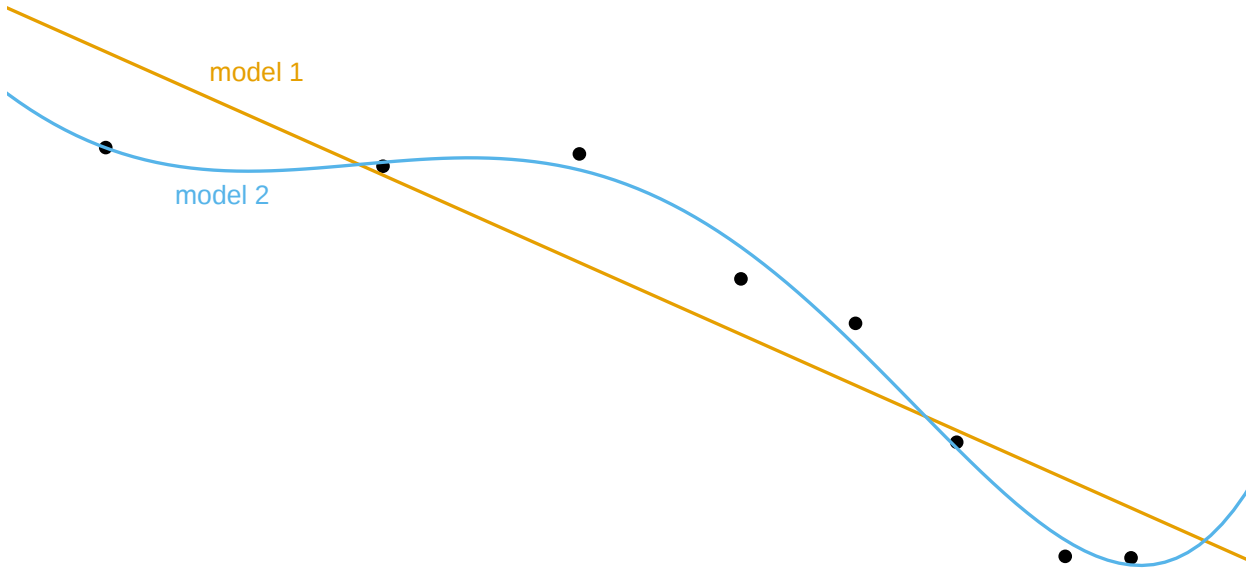
We can now formulate GLMs, fit GLMs to data, and also interpret and further investigate the estimated parameters.

Up next in Chapter 6:

- how to choose between different plausible models
  (model selection)
- how to check if a chosen model captures all relevant features of the data
  (model checking)

Chapter 6: Model selection & model checking

The bias-variance trade-off explained in one picture



Model 1 seems too inflexible $\rightsquigarrow$ systematic pattern not captured $\rightsquigarrow$ **high bias**.

Model 2 seems too flexible $\rightsquigarrow$ overfitting $\rightsquigarrow$ **high variance**.

Bias-variance trade-off — an illustration based on simulated data

1. simulate 100 data points from $Y_i = 1 + 2x_i - 2x_i^2 + \epsilon_i$, $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$

2. fit the linear models

$$\begin{aligned}
&\text{model 1:} && Y_i = \beta_0 + \beta_1 x_i + \epsilon_i \\
&\text{model 2:} && Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i \\
&\text{model 3:} && Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i \\
&\text{model 4:} && Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \epsilon_i
\end{aligned}$$

3. calculate, for each of the models, the integrated squared error (ISE),

$$\text{ISE} = \int \left(f(x) - \hat{f}(x)\right)^2 dx,$$

   where $f(x)$ is the true and $\hat{f}(x)$ is the estimated regression function

4. repeat 1.–3. 5000 times

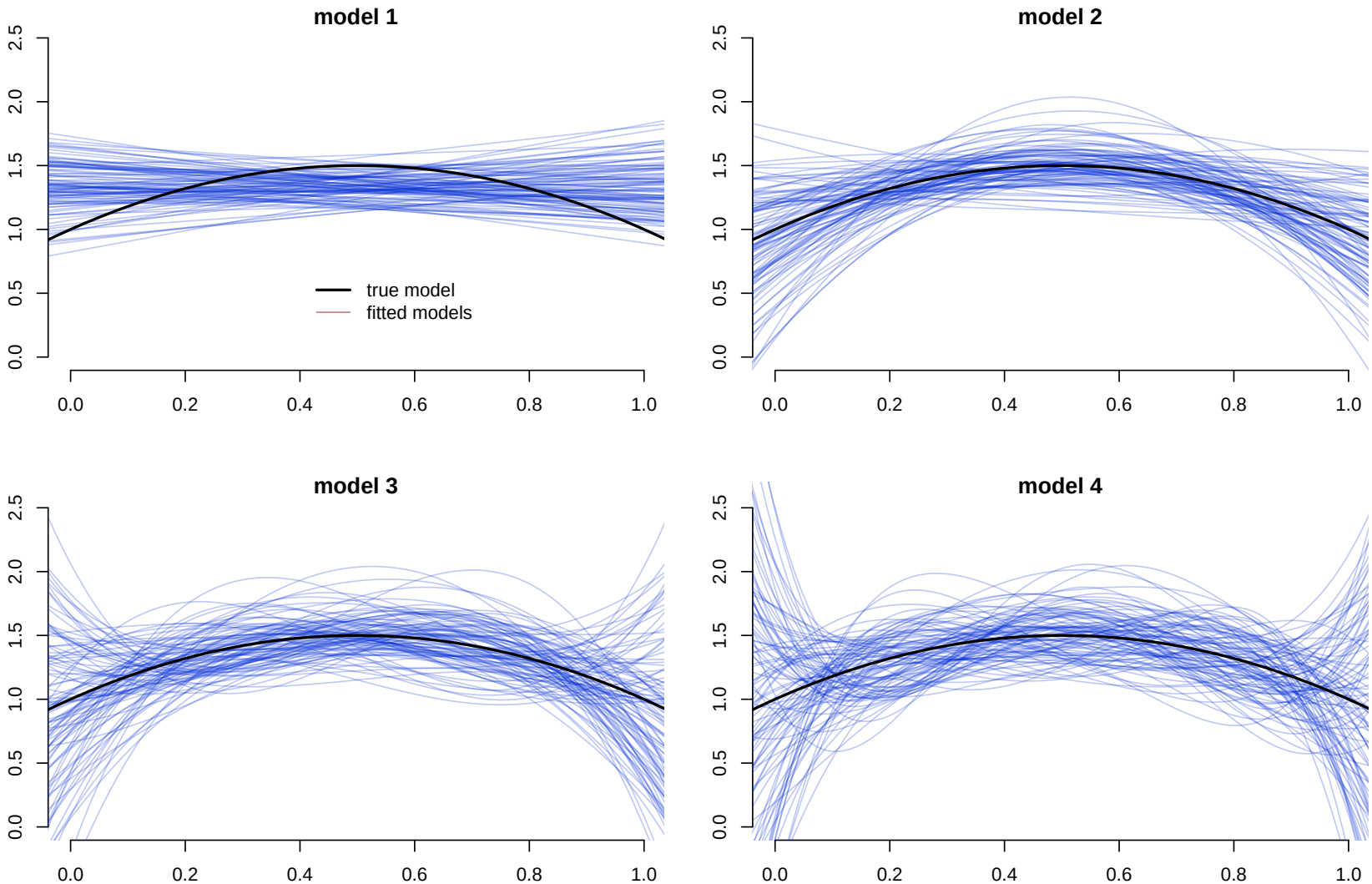
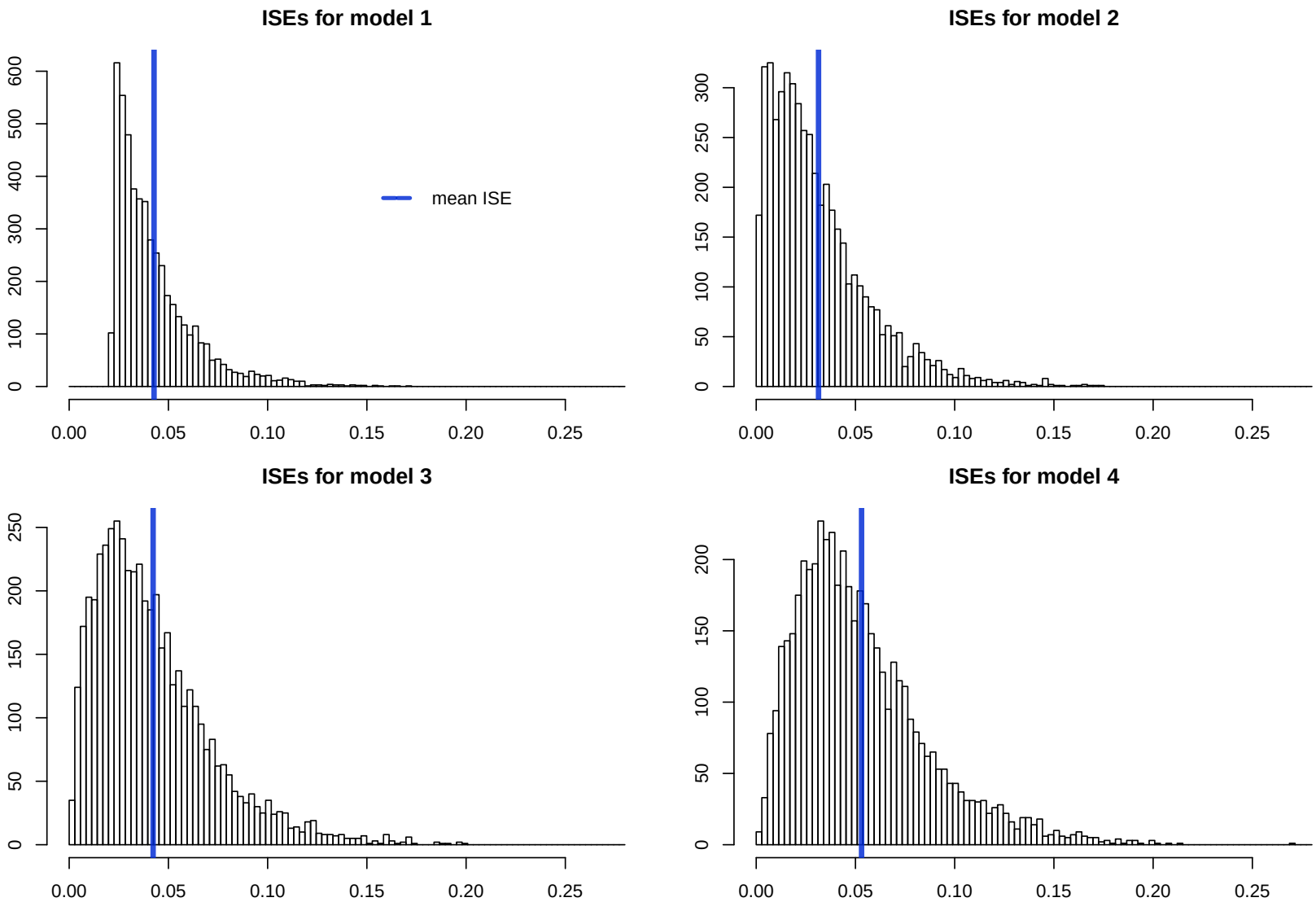


Figure: Illustration of the first 100 (of 5000) fitted regression functions, for models 1–4.


ISEs for model 1
mean ISE
0
100
200
300
400
500
600
0.00
0.05
0.10
0.15
0.20
0.25
ISEs for model 2
0
50
100
150
200
250
300
0.00
0.05
0.10
0.15
0.20
0.25
ISEs for model 3
0
50
100
150
200
250
0.00
0.05
0.10
0.15
0.20
0.25
ISEs for model 4
0
50
100
150
200
0.00
0.05
0.10
0.15
0.20
0.25

Model selection — overview

Aim: to find the **right balance between flexibility and parsimony**[27].

A model should be:

- sufficiently flexible to capture all systematic effects...
- ...yet not overly flexible as to avoid accidentally modelling noise

In other words: we increase complexity of a model only if it's worth it!

[27] or, in other words, the balance between overfitting and underfitting

Model selection for GLMs — outline

We first discuss problems with repeated application of hypothesis tests.

We then consider the **Akaike Information Criterion** (AIC) as one example of a generally applicable model selection criterion.

Other criteria such as the BIC and cross-validation will only briefly be mentioned.

Chapter 6: Model selection & model checking

6.2 Variable selection via hypothesis testing

Types of model selection within regression

Given a model formulation, say Poisson regression, model selection could mean:

1. choosing which explanatory variables to include in the model
2. deciding if polynomial and/or interaction terms are required

Model selection at a higher level:

3. choose a suitable model formulation
   (e.g. linear model vs. gamma GLM, identity vs. log link, ...)

Variable selection within regression

For 1. and 2., we could in principle use hypothesis tests.

For example, for Poisson regression, we reject $H_0 : \beta_j = 0$ if

$$|Z| = |\hat{\beta}_j / \hat{\sigma}_{\hat{\beta}_j}| > z_{1-\alpha/2}$$

This way, we can decide for each candidate variable if it should be in the model:

- H_0 rejected $\rightsquigarrow$ indication that there is an effect $\rightsquigarrow$ keep variable
- H_0 retained $\rightsquigarrow$ no strong evidence $\rightsquigarrow$ remove variable to avoid overfitting

We often wish to decide which of *several* covariates to include in a GLM.

One way to do this is to implement a **backward** (or **forward**) **selection scheme**. For example, a backward selection scheme would proceed as follows:

1. fit the most complicated model, incorporating all covariates
2. for each parameter, calculate the associated *p*–value
3. if all *p*–values are below α then stop — otherwise refit the model excluding the covariate with the highest *p*–value above α, and return to step 2.

(forward selection is analogous, just in the other direction)

Variable selection for the Donner party data

We illustrate backward variable selection in the Donner party example.

The most complex model we consider is

$$\begin{aligned}\text{logit}(\pi_i) = \beta_0 &+ \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{age}_i^2 + \beta_3 \cdot \text{gender}_i + \beta_4 \cdot \text{size of kin}_i + \beta_5 \cdot \text{size of kin}_i^2 \\ &+ \beta_6 \cdot \text{age}_i \cdot \text{gender}_i + \beta_7 \cdot \text{age}_i \cdot \text{size of kin}_i + \beta_8 \cdot \text{size of kin}_i \cdot \text{gender}_i,\end{aligned}$$

where $Y_i \sim Bern(\pi_i)$ and $Y_i = 1$ corresponds to survival of the i–th individual.

```
glm(formula = survival ~ age + gender + kin + I(age^2) + I(kin^2) +
    age:gender + age:kin + gender:kin, family = binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.754155   2.091708  -3.229 0.001242 **
age          0.355893   0.108331   3.285 0.001019 **
gender      -1.157136   1.925471  -0.601 0.547865
kin          1.354551   0.375136   3.611 0.000305 ***
I(age^2)    -0.006504   0.002150  -3.026 0.002479 **
I(kin^2)    -0.069048   0.018564  -3.719 0.000200 ***
age:gender   0.015283   0.056669   0.270 0.787395
age:kin     -0.009188   0.006902  -1.331 0.183141
gender:kin   0.253622   0.146957   1.726 0.084380 .
```

$\rightsquigarrow$ remove the interaction term age/gender.

```
glm(formula = survival ~ age + gender + kin + I(age^2) + I(kin^2) +
    age:kin + gender:kin, family = binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.742761   2.064782  -3.266 0.001092 **
age          0.353833   0.106939   3.309 0.000937 ***
gender      -0.806660   1.388806  -0.581 0.561355
kin          1.334686   0.363283   3.674 0.000239 ***
I(age^2)    -0.006448   0.002129  -3.029 0.002454 **
I(kin^2)    -0.068278   0.018202  -3.751 0.000176 ***
age:kin     -0.008405   0.006229  -1.349 0.177255
gender:kin   0.241426   0.138267   1.746 0.080797 .
```

Note that including interaction terms but not the corresponding main effects has undesirable consequences, inter alia, on interpretability — usually to be avoided!

$\rightsquigarrow$ remove the interaction term age/kin.

```
glm(formula = survival ~ age + gender + kin + I(age^2) + I(kin^2) +
    gender:kin, family = binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.961651   1.421438  -3.491 0.000482 ***
age          0.311140   0.096012   3.241 0.001193 **
gender      -1.042794   1.315171  -0.793 0.427838
kin          1.050951   0.276157   3.806 0.000141 ***
I(age^2)    -0.007299   0.002123  -3.438 0.000586 ***
I(kin^2)    -0.060806   0.016850  -3.609 0.000308 ***
gender:kin   0.260438   0.135354   1.924 0.054339 .
```

$\rightsquigarrow$ remove the interaction term gender/kin.

```
glm(formula = survival ~ age + gender + kin + I(age^2) + I(kin^2),
    family = binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.786191   1.321596  -3.622 0.000293 ***
age          0.275527   0.086491   3.186 0.001444 **
gender       1.274255   0.646112   1.972 0.048588 *
kin          0.891171   0.245333   3.632 0.000281 ***
I(age^2)    -0.006313   0.001893  -3.334 0.000856 ***
I(kin^2)    -0.046482   0.013674  -3.399 0.000676 ***
```

$\rightsquigarrow$ this is our **final model**.

Are hypothesis tests sufficient?

Unfortunately, model selection using hypothesis tests brings various problems:

- multicollinearity can lead to the exclusion of relevant variables
- forward/backward selection may yield different models — how to choose?
- multiple testing drastically increases the probability of a type I error
- in fact, this kind of testing does not even constitute a valid significance test — we're looking at the *minimum* (absolute) z value, which is not $\mathcal{N}$–distributed
- can't be used for all model selection problems[28]
- may not be feasible if there's a very large number of candidate models

[28] e.g. to compare $\mathbb{E}(Y_i) = \beta_0 + \beta_1 \cdot x_i$ vs. $\log\big(\mathbb{E}(Y_i)\big) = \beta_0 + \beta_1 \cdot \sqrt{x_i} + \epsilon_i$

Chapter 6: Model selection & model checking

6.3 Akaike Information Criterion

Kullback-Leibler divergence

Idea: consider a quantity measuring the discrepancy $\Delta(\hat{M}, M_0)$ between fitted model ($\hat{M}$) and true model (M_0), and find model that minimises this quantity.

Different discrepancy measures can be considered, but the **Kullback-Leibler divergence** leads to particularly nice results — it is given by

$$\Delta(\hat{M}, M_0) = \mathbb{E}_{M_0} \log \left(\frac{\mathcal{L}_{M_0}}{\mathcal{L}_{\hat{M}}} \right),$$

with the subscript at $\mathbb{E}$ indicating the source of the randomness $\rightsquigarrow$ this is the expectation with respect to observations drawn randomly from M_0.

Illustration (in class)

$$\Delta(\hat{M}, M_0) = \mathbb{E}_{M_0} \log \left(\frac{\mathcal{L}_{M_0}}{\mathcal{L}_{\hat{M}}} \right)$$

- ⇝ this discrepancy deems a model good if, **on average over many samples**, it assigns a high probability to observations generated from the true model
- ⇝ small values indicate that the fitted model is close to the true model
- ⇝ obviously, M_0 is unknown, so $\Delta(\hat{M}, M_0)$ can't be calculated
- ⇝ however, it can be estimated!

The KL divergence can be written as

$$\Delta(\hat{M}, M_0) = \mathbb{E}_{M_0} \log \mathcal{L}_{M_0} - \mathbb{E}_{M_0} \log \mathcal{L}_{\hat{M}}.$$

For model selection, we can drop the first term since $\hat{M}$ has no influence on it.

Thus, we try to maximise

$$\mathbb{E}_{M_0} \log \mathcal{L}_{\hat{M}},$$

which **still depends on the true model** M_0 (due to the expectation).

A seemingly natural estimator would be the plug-in estimator:

$$\log \mathcal{L}_{\hat{M}}$$

Plug-in estimator for the relevant part of the KL divergence:

$$\log \mathcal{L}_{\hat{M}}$$

Unfortunately, this estimator is **positively biased due to overfitting**.
(the model fits the given sample better than an average sample)

Illustration: (in class)

From the KL divergence to the Akaike Information Criterion

Akaike showed that, under several regularity conditions,

$$\text{bias of the plug-in estimator} \approx \text{number of parameters of } \hat{M} \quad (= K).$$

Note the remarkable simplicity of this result!!

Correcting the plug-in estimator for the approximate bias, we obtain as criterion:

$$\log \mathcal{L}_{\hat{M}} - K$$

For historical reasons, instead of selecting the model that maximises this quantity, the quantity is usually multiplied by -2, leading to

$$-2 \log \mathcal{L}_{\hat{M}} + 2K,$$

which we then wish to *minimise*.

For a given model, the **Akaike Information Criterion** (AIC) is

$$\text{AIC} = -2 \log \mathcal{L}_{\hat{M}} + 2K,$$

where $\mathcal{L}_{\hat{M}}$ is the maximal value of the log-likelihood of the model. Given a set of candidate models, we choose the one with the *smallest* AIC.

- the AIC **rewards model fit** ($2 \log \mathcal{L}_{\hat{M}}$) yet **penalises complexity** ($2K$)
- this reflects the trade-off between **flexibility**[29] and **parsimony**[30]
- the AIC prefers a complex model over a simple model only if the log-likelihood improvement outweighs the increase in complexity
- R gives the AIC of a GLM in the output of `summary(glm(...))`

[29] rewarding it: complex models can fit the data better & hence lead to smaller $-2 \log \mathcal{L}_{\hat{M}}$

[30] by penalising complexity

AIC-based selection of a model for the Donner party data

We illustrate AIC-based model selection in the Donner party example.

The most complex model we consider again is

$$\begin{aligned}\text{logit}(\pi_i) = \beta_0 &+ \beta_1 \cdot \text{age}_i + \beta_2 \cdot \text{age}_i^2 + \beta_3 \cdot \text{gender}_i + \beta_4 \cdot \text{size of kin}_i + \beta_5 \cdot \text{size of kin}_i^2 \\ &+ \beta_6 \cdot \text{age}_i \cdot \text{gender}_i + \beta_7 \cdot \text{age}_i \cdot \text{size of kin}_i + \beta_8 \cdot \text{size of kin}_i \cdot \text{gender}_i\end{aligned}$$

There are $2^8 = 256$ submodels of the full model from the previous slide.

AIC values for what based on previous considerations are the most plausible candidate models, plus the two extreme cases as benchmarks:

age	gender	kin	age^2	kin^2	age/gen.	age/kin	gen./kin	AIC
✓	✓	✓	✓	✓	✓	✓	✓	92.847
✓	✓	✓	✓	✓		✓	✓	**90.921**
✓	✓	✓	✓	✓	✓		✓	92.821
✓	✓	✓	✓	✓	✓	✓		94.177
✓	✓	✓	✓	✓			✓	90.964
✓	✓	✓	✓	✓		✓		92.226
✓	✓	✓	✓	✓	✓			94.281
✓	✓	✓	✓	✓				93.108
✓		✓	✓	✓				95.413
								119.258

$\rightsquigarrow$ **chosen model** includes all variables except the interaction age/gender.

The importance of checking the (absolute) goodness of fit

An AIC value such as 90.921, taken on its own, doesn't tell us anything — it is of interest only when compared to AICs of competing models.

The AIC measures the **relative goodness of fit** (relative to competing models).

This means that even if we consider hundreds of candidate models, the selected model might still be a bad one.

It's important to also investigate the **absolute goodness of fit**, for example using residual analyses (later in this chapter!).

Selection bias

Too much selection can do more harm than good — such an "overdose of selection" leads to a problem called **selection bias**.

To illustrate the problem, suppose that 100 candidate models, $M_1, \ldots, M_{100}$, are considered, and that the models are ranked in terms of their AIC values:

rank	model	AIC values
1	M_{44}	$\text{AIC}_{M_{44}}$
2	M_{13}	$\text{AIC}_{M_{13}}$
3	M_{92}	$\text{AIC}_{M_{92}}$
4	M_{28}	$\text{AIC}_{M_{28}}$
$\vdots$	$\vdots$	$\vdots$
100	M_7	AIC_{M_7}

where $\text{AIC}_{M_{44}} < \text{AIC}_{M_{13}} < \text{AIC}_{M_{92}} < \text{AIC}_{M_{28}} < \ldots < \text{AIC}_{M_7}$.

If these AIC values were the **actual KL discrepancies** between true model and fitted model, then this would all be fine.

However, recall that we're only **estimating the KL discrepancies**. The ranked list of models with the actual KL discrepancies will (in general) look different, say

rank	model	KL discrep.
1	M_{28}	$\text{KL}_{M_{28}}$
2	M_{67}	$\text{KL}_{M_{67}}$
3	M_{13}	$\text{KL}_{M_{13}}$
4	M_{44}	$\text{KL}_{M_{44}}$
$\vdots$	$\vdots$	$\vdots$
100	M_{50}	$\text{KL}_{M_{50}}$

where $\text{KL}_{M_{28}} < \text{KL}_{M_{67}} < \text{KL}_{M_{13}} < \text{KL}_{M_{44}} < \ldots < \text{KL}_{M_{50}}$.

The KL–based list ranks the models from **best to worst**, whereas the AIC-based list ranks the models from **apparently best to apparently worst**.

For some models, the KL discrepancy will be overestimated by the AIC, whereas for others it will be underestimated, potentially leading to ranks being swapped.

In other words, due to particular details of the sample at hand, the AIC will rank some models too high (they get lucky!), and others too low (bad luck!).

Consequences:

- model selected based on AIC appears to perform better than it really does[31]
- variables will be included because "they got lucky"[32]

The more models we consider, the bigger a problem this becomes.

[31] such that we may be overly confident in our predictions

[32] note the analogy to *p*–hacking!

Model averaging

Model averaging addresses the uncertainties involved in model selection.

Acknowledging that several models may describe the data about equally well, it is natural to build predictions based on averaging predictions from several models.

A possible **multi-model prediction** of a future value Y of the response is

$$\hat{y} = \sum_{h=1}^{H} w_h \hat{y}^{(h)},$$

where $\hat{y}^{(h)}$ is the prediction under M_h, and where w_h are **Akaike weights**[33],

$$w_h = \frac{e^{-0.5\Delta\text{AIC}_{M_h}}}{\sum_{j=1}^{H} e^{-0.5\Delta\text{AIC}_{M_j}}},$$

with ΔAIC_{M_h} the difference in AIC values between M_h and the "best" model.

[33] these can be interpreted as the probabilities of the corresponding model being the best

An alternative model selection criterion: the BIC

The AIC tends to be too generous with respect to increasing model complexity.

The **Bayesian Information Criterion (BIC)** is a more conservative alternative:

$$\text{BIC} = -2 \log \mathcal{L}_{\hat{M}} + \log(n) \cdot K$$

- looks similar to AIC, but has a completely different theoretical foundation[34]
- complexity penalty is higher when $n \geq 8$
- the BIC thus tends to select simpler models

[34] the AIC attempts to minimise discrepancy between model and reality, while the BIC seeks the model which is most likely to be true

Cross-validation as another approach to model selection

Cross-validation considers the criterion

$$\text{CV} = \sum_{i=1}^{n} \left(y_i - \hat{y}_{-i}\right)^2,$$

where $\hat{y}_{-i}$ is the prediction of y_i using the GLM fitted to all data except (x_i, y_i).

- the i–th observation is regarded as a future observation
- the GLM is fitted to the other $n - 1$ observations
- then we check the prediction of y_i under this fitted model
- this is repeated for all $i = 1, \ldots, n$

From a set of candidate models, we choose the one giving the smallest CV.

Practical recommendations

- there is **no universally applicable strategy** that can guarantee a satisfactory outcome
- model selection criteria **point us in some direction**, but taken on their own don't provide much evidence for anything
- in practice, model selection involves:
  - **thinking** about which candidate models to consider — if possible, model formulation should be aligned with any relevant theory
  - intuition and, sometimes, pragmatism
  - it is always advisable to thoroughly investigate (all) strong candidate models[35]
- up next: model checking, where we discuss the last point in more detail

[35] in order to get a better understanding of the different models' performances, which will help making an informed choice between candidate models, taking the study aim into account

Chapter 6: Model selection & model checking

6.4 Deviance

For a GLM under consideration, the **likelihood ratio statistic** is given as:

$$\lambda = \frac{\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sat}})}{\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sim}})},$$

where, within the model class defined by the distributional assumption,

- $\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sat}})$ denotes the likelihood under the *saturated* (or full) model, where one parameter is estimated for each of the n observations
- $\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sim}})$ is the likelihood under the *simplified* model (the GLM), where say only two parameters are used to explain n observations

Example Poisson GLM (with one covariate):

$$Y_i \sim Po(\theta_i)$$

$$\hat{\boldsymbol{\theta}}_{\text{sat}} = (\hat{\theta}_{1,\text{sat}}, \ldots, \hat{\theta}_{n,\text{sat}}) = (y_1, \ldots, y_n)$$
$$\hat{\boldsymbol{\theta}}_{\text{sim}} = (\hat{\theta}_{1,\text{sim}}, \ldots, \hat{\theta}_{n,\text{sim}}) = (e^{\hat{\beta}_0+\hat{\beta}_1 x_1}, \ldots, e^{\hat{\beta}_0+\hat{\beta}_1 x_n})$$

Simple properties of the likelihood ratio statistic

$$\lambda = \frac{\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sat}})}{\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sim}})}$$

- λ measures the discrepancy between simplified model and saturated model
- $\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sim}})$ can't be higher than $\mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sat}})$, hence $\lambda \geq 1$
- if the simplified model is adequate, then λ shouldn't be "much higher" than 1, indicating that the saturated model doesn't explain the data much better

Likelihood ratio statistic in the football example

Recall the Poisson GLM fitted to the football data:

$$\text{goals}_i \overset{\text{iid}}{\sim} Po(\lambda_i), \quad \lambda_i = \exp\big(\hat{\beta}_0 + \hat{\beta}_1 \cdot \text{mvdiff}_i\big)$$

The λ_i, $i = 1, \ldots, 612$, hence are determined by (only) two model parameters.

Saturated model: estimate $\hat{\lambda}_i$ for each $i = 1, \ldots, 612$, without imposing any structure such as a regression model $\rightsquigarrow$ 612 parameters!

Using the λ_i implied under the two different models, we obtain

$$\lambda = \frac{\mathcal{L}(\hat{\boldsymbol{\lambda}}_{\text{sat}})}{\mathcal{L}(\hat{\boldsymbol{\lambda}}_{\text{sim}})} = \frac{e^{-591.81}}{e^{-957.02}} = e^{365.21}$$

All very well, but what does a value like

$$\lambda = e^{365.21}$$

really tell us??

- we initially noted that if the simplified model is adequate, then λ shouldn't be much higher than 1
- but what is "much higher than 1"?
- whether or not the value of λ is large depends on:
  i) the sample size
  ii) the complexity of the simplified model[36]
- what we need is **a scale** that takes both into account

[36] and hence the difference in the number of parameters — here the saturated model has 610 additional parameters!

Deviance

Considering a simple transformation of λ, called the **deviance**,

$$D = 2 \log \lambda = 2\big(\log \mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sat}}) - \log \mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sim}})\big),$$

we obtain such a scale:

Suppose that a GLM with $p + 1$ parameters is fitted to n observations.

Under the null hypothesis that the GLM considered adequately describes the data, we then have

$$D = 2 \log \lambda \sim \chi^2_{n-(p+1)},$$

approximately for large n.

Likelihood ratio test (LRT) based on the deviance

- the chi-squared distribution is the scale at which we can measure the (transformed) likelihood ratio statistic
- clearly, the worse the model, the larger is λ, hence the larger is D — thus, a large D indicates that the GLM does not explain the data well
- if the observed D is not consistent with its approx. theoretical distribution, then we reject the null hypothesis that the GLM describes the data well
- we reject the GLM at the α significance level (usually 0.05) if the deviance D is larger than the $(1-\alpha)$–quantile of the $\chi^2_{n-(p+1)}$ distribution
- this test is called **likelihood ratio test** (LRT)
- in R, `summary(glm(...))` gives the deviance (see “Residual deviance”)

Deviance and LRT in the football example

In the football example, we had $\lambda = e^{365.21}$, such that

$$D = 2 \log \lambda = 730.42$$

Under the null hypothesis that the GLM describes the data well, $D \sim \chi^2_{610}$.[37]

The p–value, i.e. the probability of observing a D larger than 730.42 under the null[38], is 0.00055, i.e. very small.

$\rightsquigarrow$ we reject the hypothesis that the GLM is an adequate description of the data!

[37] $610 = 612 - (1 + 1) = n - (p + 1)$

[38] obtained via `1-pchisq(730.42,610)` in R

```
> summary(mod)

Call:
glm(formula = goals ~ score_diff, family = poisson)

[...]

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.4168639  0.0334556  12.460   <2e-16 ***
mvdiff      0.0010783  0.0001177   9.162   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 813.63  on 611  degrees of freedom
Residual deviance: 730.43  on 610  degrees of freedom
AIC: 1918

Number of Fisher Scoring iterations: 5
```

Deviance as a goodness-of-fit check

- the deviance can be used to assess **overall goodness-of-fit**:
  - null hypothesis not rejected $\rightsquigarrow$ GLM describes the data reasonably well
  - null hypothesis rejected $\rightsquigarrow$ there is some lack of fit
- if the LRT rejects the GLM, this means that there is relevant structure in the data not yet captured by the model
- this often happens even if the model has already captured a lot of structure and effectively can't be improved any further!
- in any case, further & more detailed checks should then be conducted to investigate what's going on (and hence to decide what to do)

Remarks on the deviance

The LRT reflects the classical trade-off between complexity and parsimony:

- if model complexity of the GLM considered is increased, the resulting likelihood will be greater or equal to the likelihood of the simpler model
- thus, the deviance of the more complex model will necessarily be smaller or equal to that of the simpler model
- but the scale changes as well: in the more complex model the 0.95-quantile of the corresponding χ^2 distribution[39] will be lower than for the simple model
- so the more complex model has “a higher bar to cross” to be satisfactory

[39] which serves as the threshold deciding whether or not the null is accepted

LRT for a simple model of the Donner Party data

```
> summary(mod)

glm(formula = survival ~ age, family = binomial)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.81733    0.37549   2.177   0.0295 *
age         -0.03237    0.01509  -2.145   0.0320 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 117.26  on 84  degrees of freedom
Residual deviance: 112.25  on 83  degrees of freedom
AIC: 116.25

> 1-pchisq(112.25,83)
[1] 0.01794567
```

$\rightsquigarrow$ the LRT rejects this model (it is not adequate)

LRT for a more complex model of the Donner Party data

```
> summary(mod)

Call:
glm(formula = survival ~ sex + age + I(age^2), family = binomial)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.041802   0.659164  -1.580  0.11399
sex          1.410763   0.559009   2.524  0.01161 *
age          0.156285   0.068051   2.297  0.02164 *
I(age^2)    -0.004130   0.001583  -2.608  0.00911 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 117.258  on 84  degrees of freedom
Residual deviance:  97.858  on 81  degrees of freedom
AIC: 105.86

> 1-pchisq(97.858,81)
[1] 0.09784643
```

$\rightsquigarrow$ the LRT does not reject this model (it seems to be adequate)

Chapter 6: Model selection & model checking

6.5 Residual analyses for GLMs

Residual analysis for linear regression models

For a linear regression model,

$$Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i,$$

the residuals are simply the vertical distances between observed responses and predictions based on the fitted model:

$$\epsilon_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \ldots + \hat{\beta}_p x_{ip}) = y_i - \hat{y}_i, \quad i = 1, \ldots, n.$$

Using suitable **residual plots**, we can check the adequacy of the assumption of linear effects and the (commonly made) assumption(s) that $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.[40]

[40] cf. assumptions (i), (iii), (iv) and (v) on slide 49.

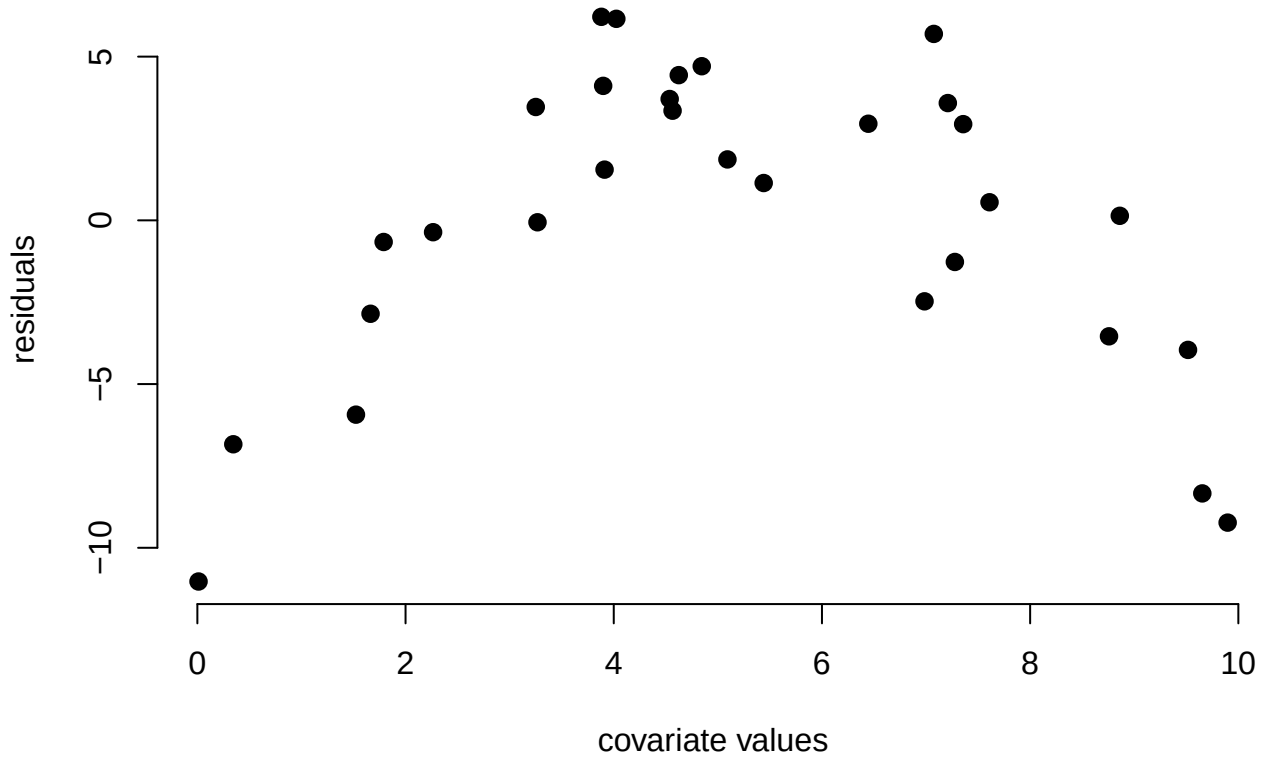
Residual plots

The following residual plots are often useful:

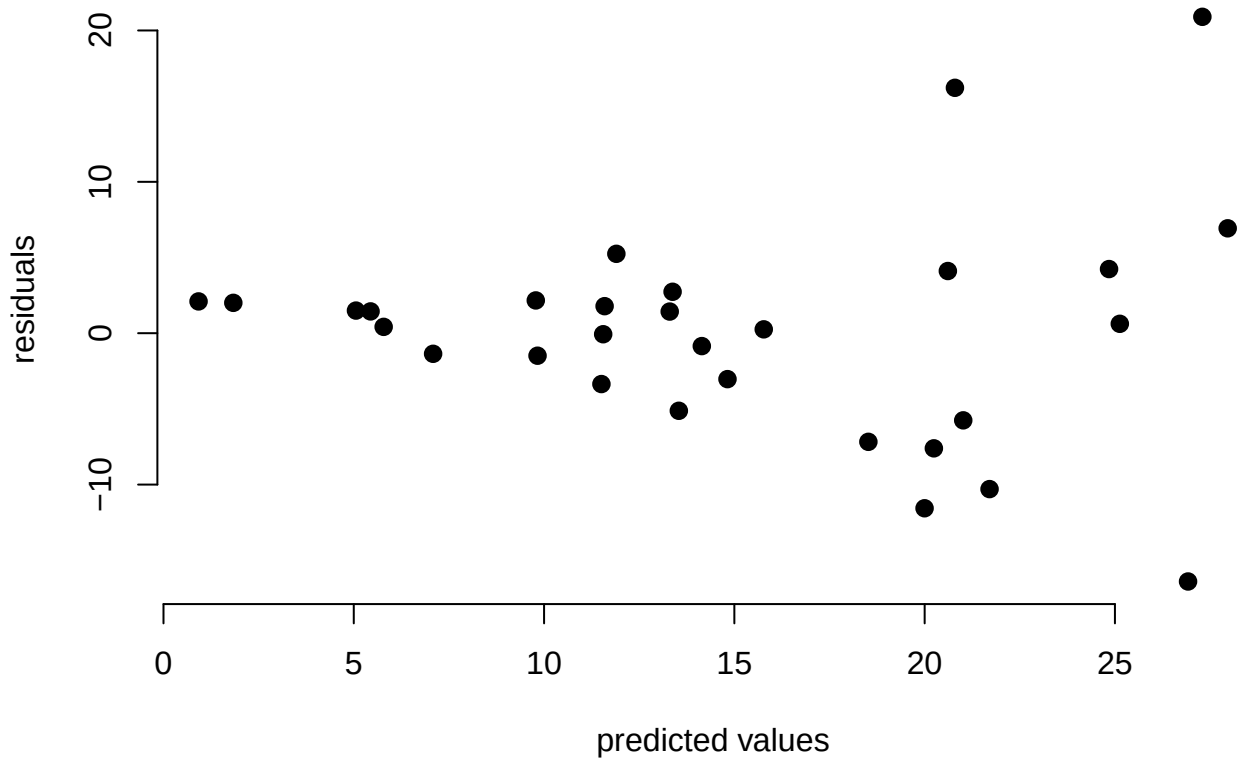
- a plot of the residuals against an individual covariate
  (to display possible problems with the way the covariate effect is modelled)
- a plot of the residuals against the corresponding predicted values
  (to display possible heteroscedasticity)
- a plot of the residuals against observation index (especially for time series)
  (to display possible problems with the assumed independence of the ϵ_i)

All of the above plots can also help to identify potential outliers.

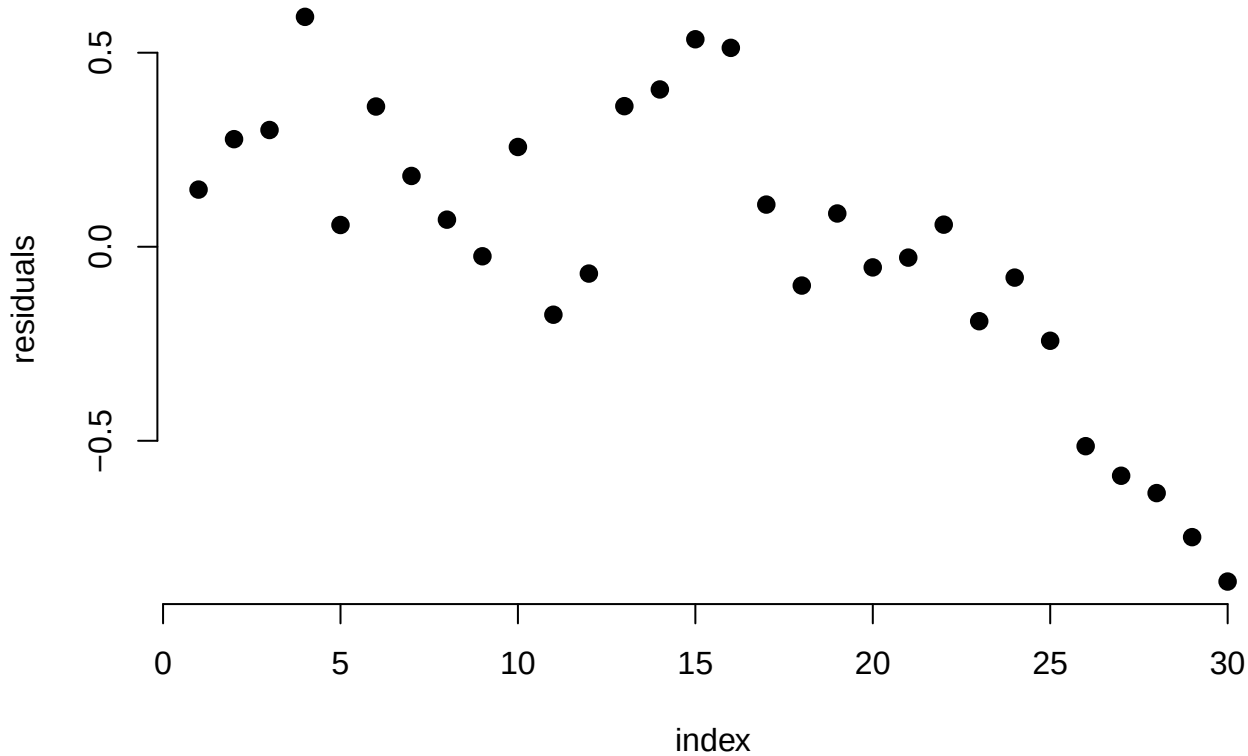
Residual analysis for linear regression models — Example 1



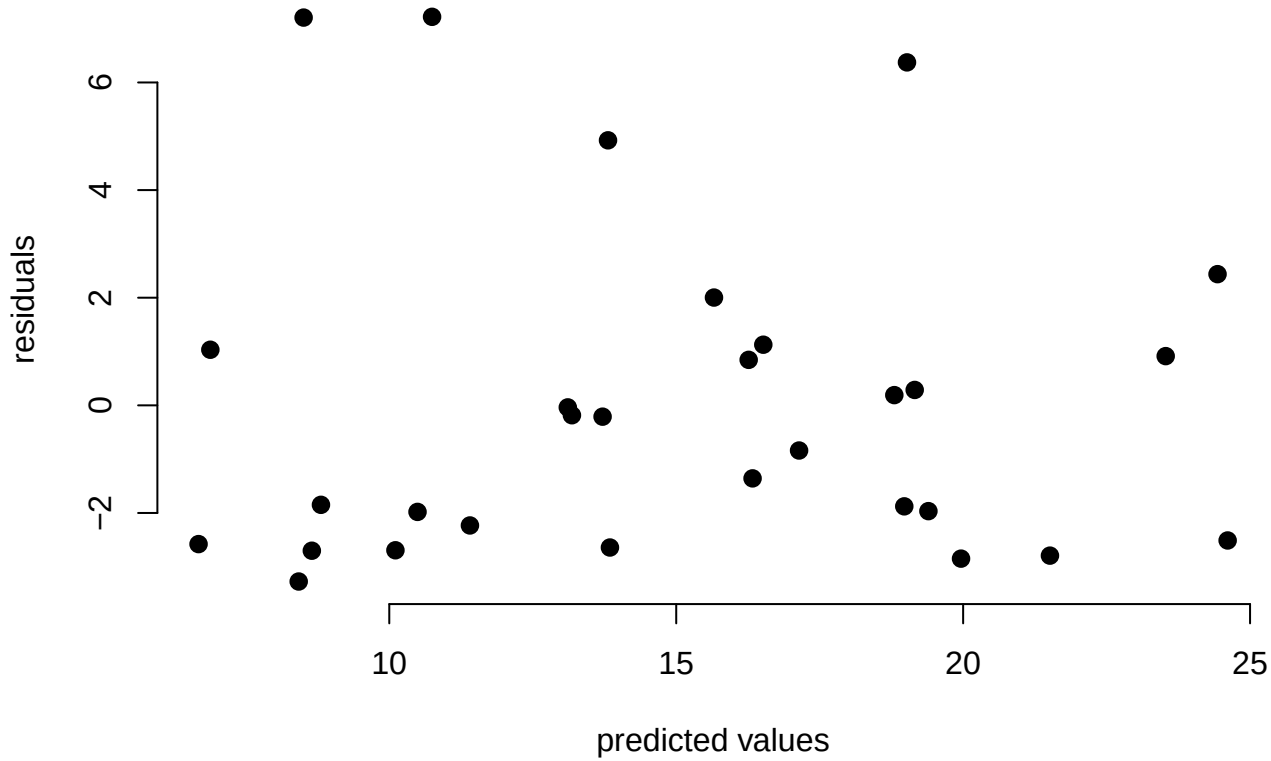
Residual analysis for linear regression models — Example 2



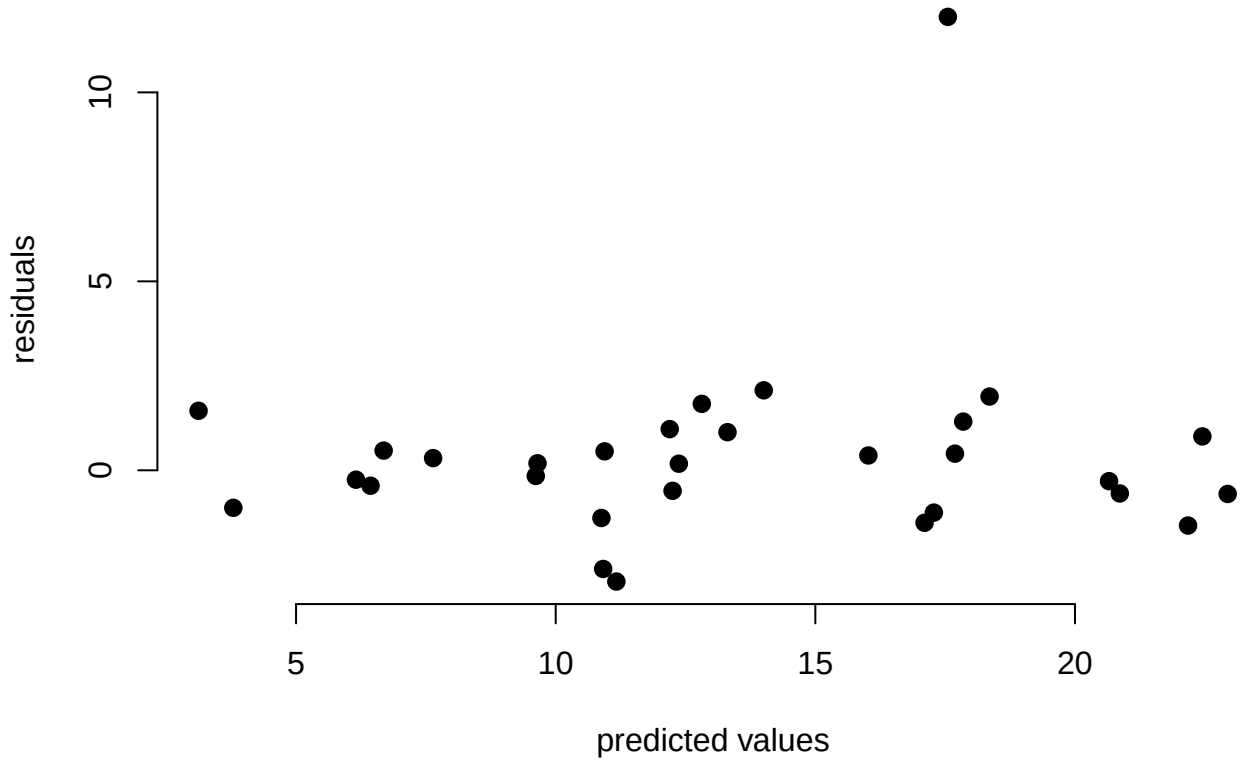
Residual analysis for linear regression models — Example 3



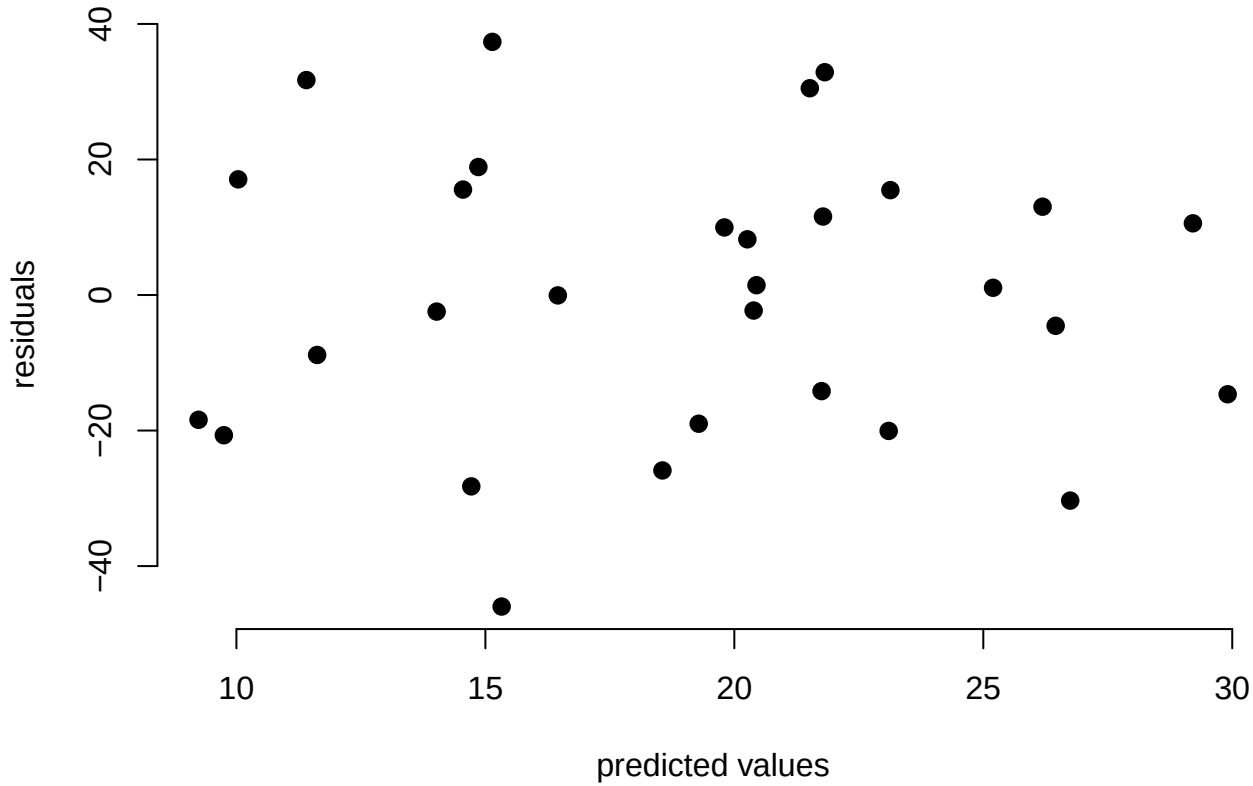
Residual analysis for linear regression models — Example 4



Residual analysis for linear regression models — Example 5



Residual analysis for linear regression models — Example 6



Residual analysis for linear regression models — summary

What to look out for:

- pattern in residuals vs. covariate values plot $\rightsquigarrow$ perhaps nonlinearity
- funnel-shaped residual plot $\rightsquigarrow$ perhaps heteroscedasticity
- pattern in residuals vs. index plot $\rightsquigarrow$ observations possibly not independent
- asymmetric distribution of residuals around 0 $\rightsquigarrow$ perhaps non-normality
- pattern looks random except for 1-2 extreme values $\rightsquigarrow$ outliers?

Rule-of-thumb: **the more random, the better**!

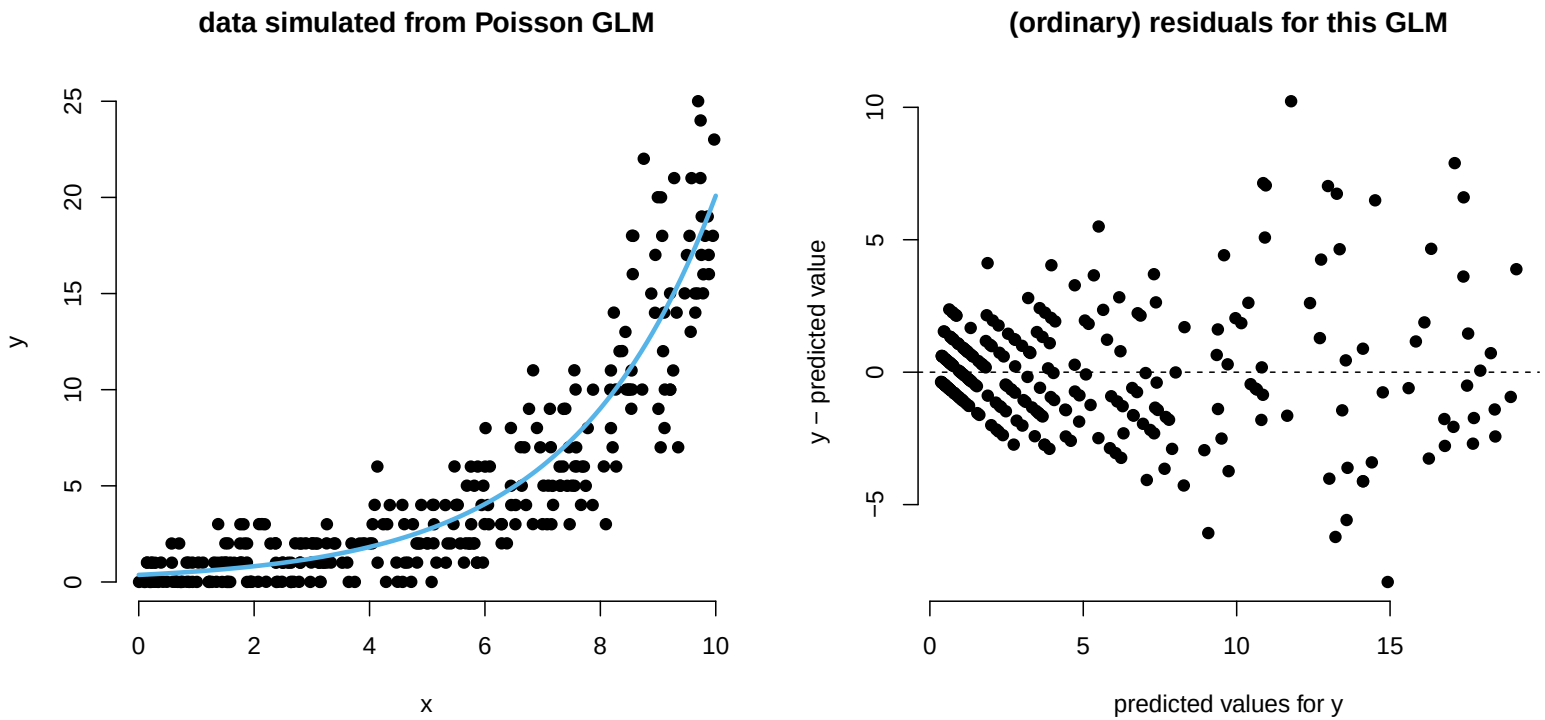
Model checking for GLMs

For GLMs, the situation is slightly more involved:

- in general, the error variance is not constant
  (which needs to be taken into account — see next slide)

- the error terms are in general not normally distributed

Illustration of error terms in GLMs



- for this artificial data generated from a Poisson GLM, the variance in the error increases as the predictor value increases
- based on this residual plot, it is next to impossible to judge whether or not the mean-variance relation implied by the model is adequate

Pearson residuals

An obvious and simple way to obtain meaningful residuals for GLMs is to standardise the ordinary residuals:

$$\epsilon_i^p = \frac{y_i - \hat{y}_i}{\hat{\sigma}_i},$$

where $\hat{\sigma}_i$ is the estimated standard deviation of Y_i under the fitted model.

These are so-called **Pearson residuals**.

If the model is adequate, the Pearson residuals should have

- zero mean
- variance approximately equal to 1

Pearson residuals

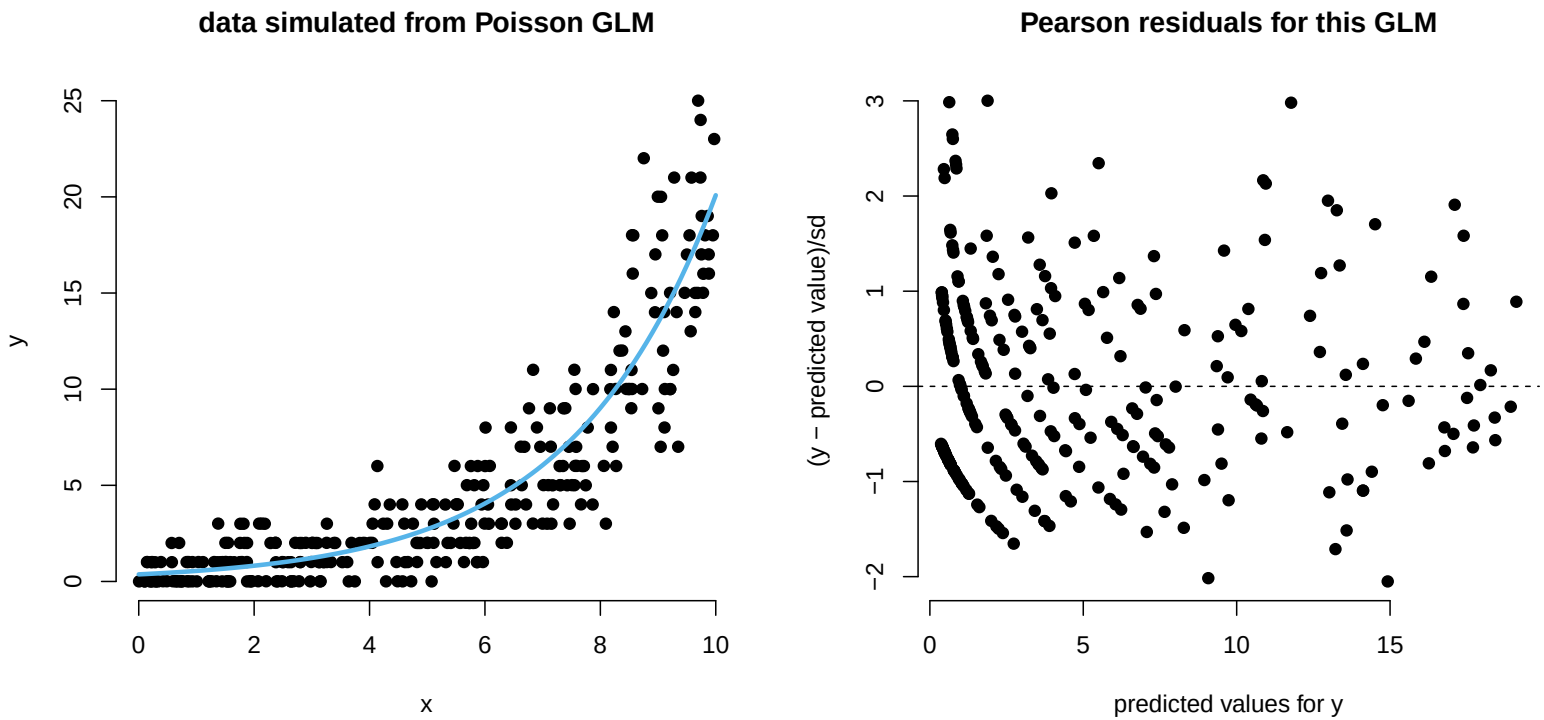
For a Poisson GLM,

$$\epsilon_i^p =$$

For a Bernoulli GLM,

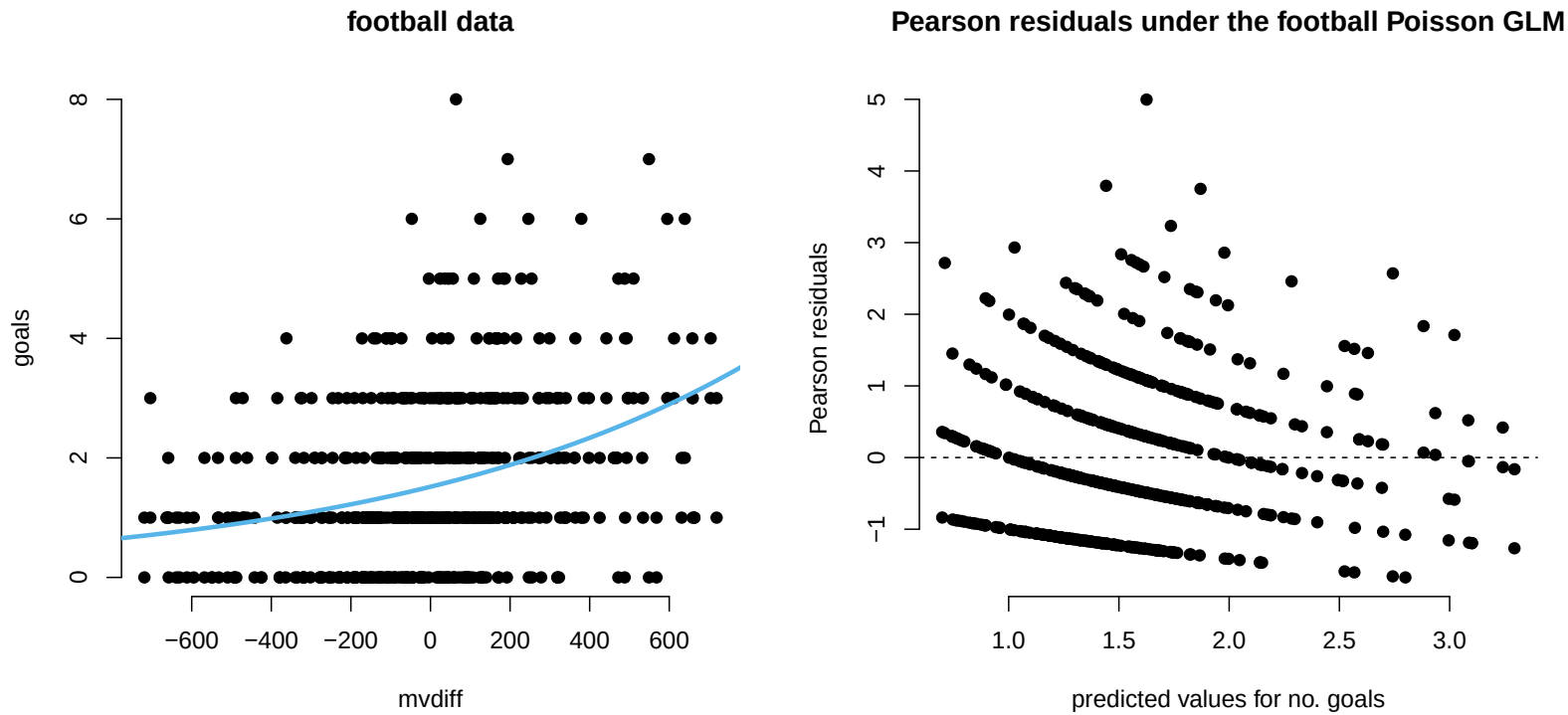
$$\epsilon_i^p =$$

Illustration of Pearson residuals



- same artificial data as before
- no clear pattern (to be expected, as the correct model is used here!)
- sample variance of Pearson residuals ≈ 1

Pearson residuals in the football example



- a few outliers, but overall no clear pattern
- however, the sample variance of the Pearson residuals here is ≈ 1.08
- in other words, the variance of the observations is slightly higher than implied under the fitted Poisson GLM ($\rightsquigarrow$ **overdispersion**)
- this is probably the main reason why the GLM was rejected by the LRT

Deviance residuals

The distribution of Pearson residuals if often highly asymmetric[41], so we can't use the normal distribution as a benchmark, e.g. to identify outliers.

In this respect, deviance residuals are often preferable — these are obtained by noting that in the deviance,

$$\begin{aligned} D &= 2\big(\log \mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sat}}) - \log \mathcal{L}(\hat{\boldsymbol{\theta}}_{\text{sim}})\big) \\ &= 2\Big(\sum_{i=1}^{n}\big(y_i b(\hat{\theta}_{i,\text{sat}}) + c(\hat{\theta}_{i,\text{sat}}) + d(y_i)\big) - \sum_{i=1}^{n}\big(y_i b(\hat{\theta}_{i,\text{sim}}) + c(\hat{\theta}_{i,\text{sim}}) + d(y_i)\big)\Big) \\ &= \sum_{i=1}^{n} \underbrace{2\big(y_i(b(\hat{\theta}_{i,\text{sat}}) - b(\hat{\theta}_{i,\text{sim}})) + c(\hat{\theta}_{i,\text{sat}}) - c(\hat{\theta}_{i,\text{sim}})\big)}_{=d_i}, \end{aligned}$$

the i–th summand, d_i, gives the contribution of the i–th data point to the deviance.

[41] since the standardisation doesn't address the asymmetry of the response's distribution

Deviance residuals

The **deviance residuals** are defined as

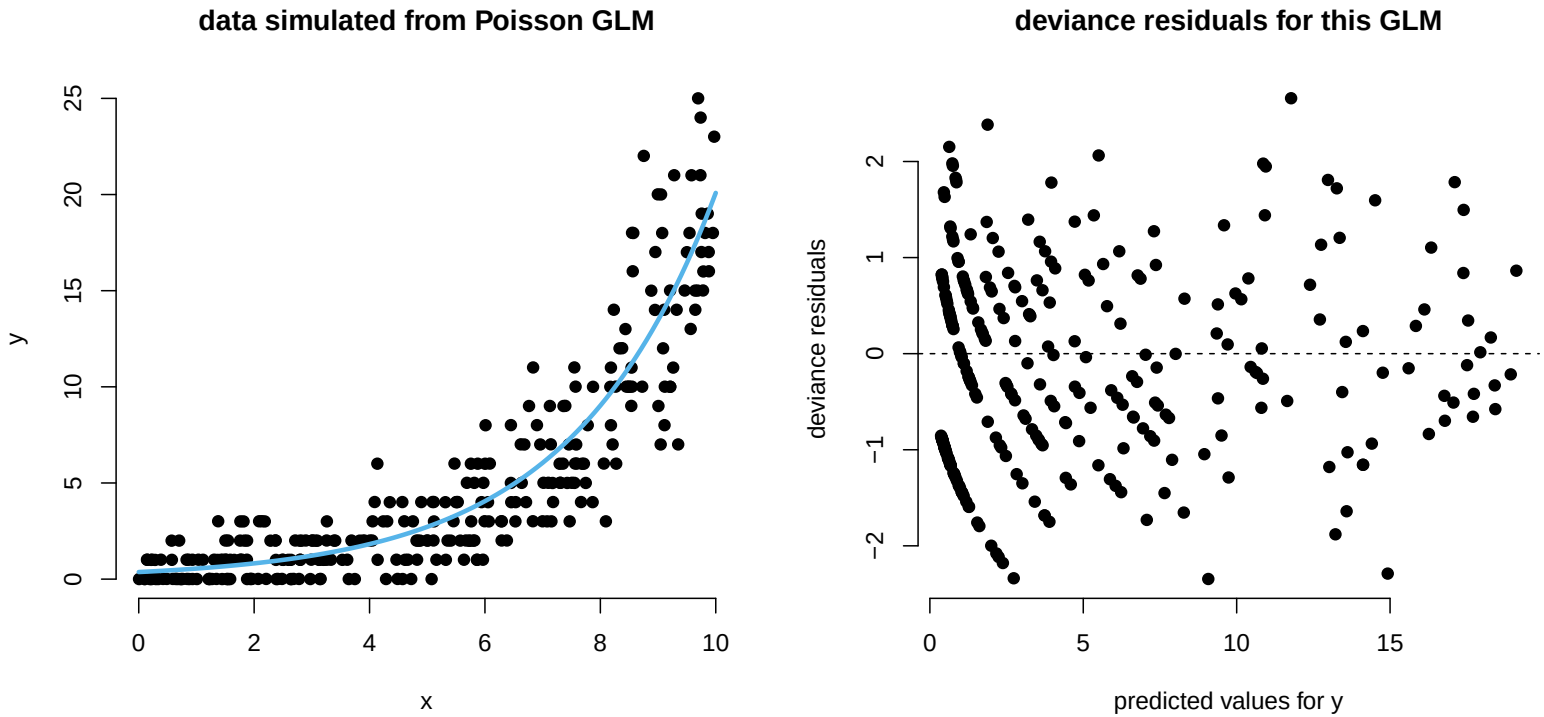
$$\epsilon_i^d = \text{sign}(y_i - \hat{y}_i)\sqrt{d_i}$$

(by taking the square root, we achieve that for the standard linear regression model the deviance residuals reduce to ordinary residuals)

If the fitted model is adequate, then these residuals are approximately normally distributed, with mean zero and constant variance.[42]

[42] (unstandardised) deviance residuals, as above, do not have unit variance (but can be standardised)

Illustration of deviance residuals



- same artificial data as before
- as for the Pearson residuals, no clear pattern

GLM residuals in R

In R, for a fitted GLM object `mod`, pearson residuals and deviance residuals can be obtained by

```
residuals(mod,type="pearson")
```

and

```
residuals(mod,type="deviance")
```

respectively.

Summary — finding a suitable GLM for given data

In practice, the search for an adequate model involves:

1. exploratory data analysis!!!
2. formulating and fitting plausible candidate models
3. using model selection tools to choose best model from the candidate models
4. checking if the chosen model captures all relevant structure in the data:
   - LRT does not lead to rejection, residual plots look good $\rightsquigarrow$ take as final model
   - any problems are revealed $\rightsquigarrow$ go back to the drawing board (step 2.) and use your insights from 4. to formulate better candidate models
   - alternatively, if model can't easily be improved and/or lack of fit is not pertinent to study aim, then we may stick to it

Steps 2., 3. and 4. require a **good intuition for the data at hand** and careful thinking, **taking the study aims into account**.

Chapter 7: Mixed models

Chapter 7: Mixed models

7.1 Illustrating example & motivation

Another look at the football model

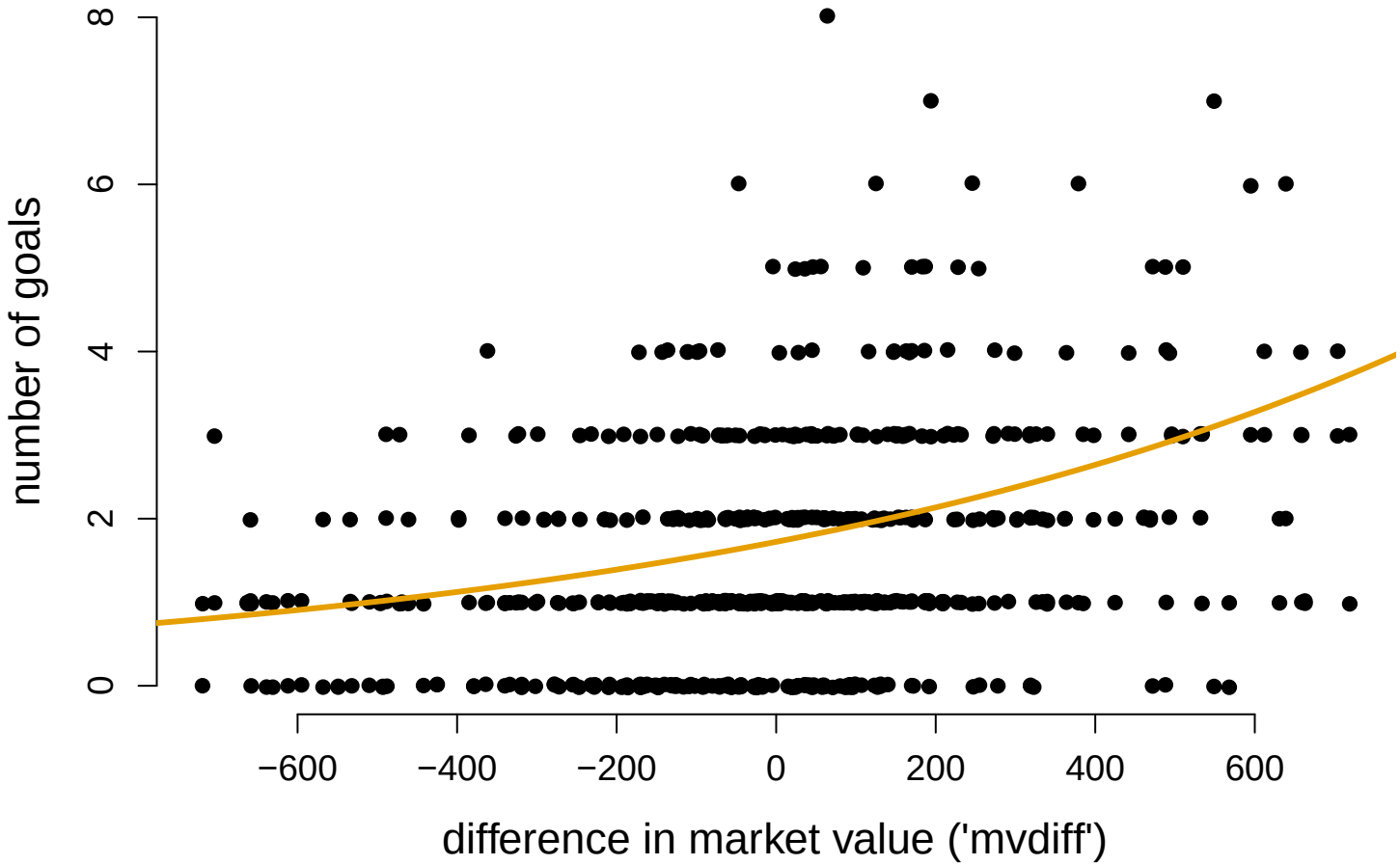
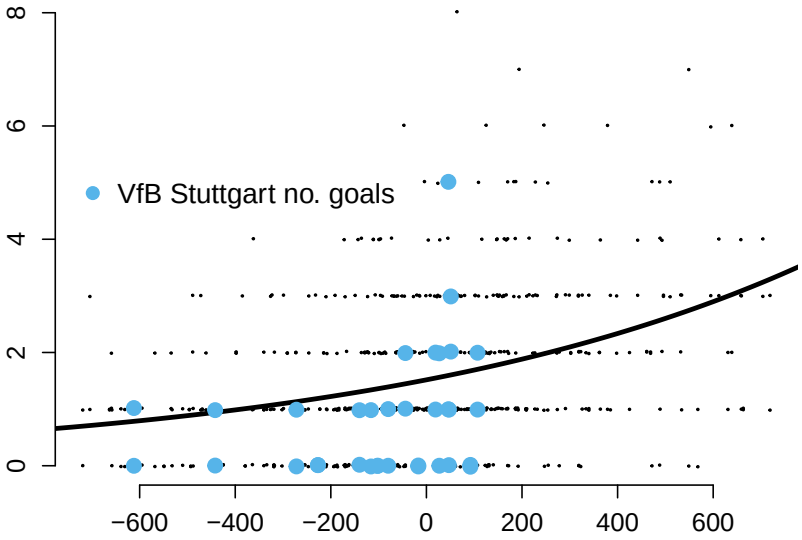
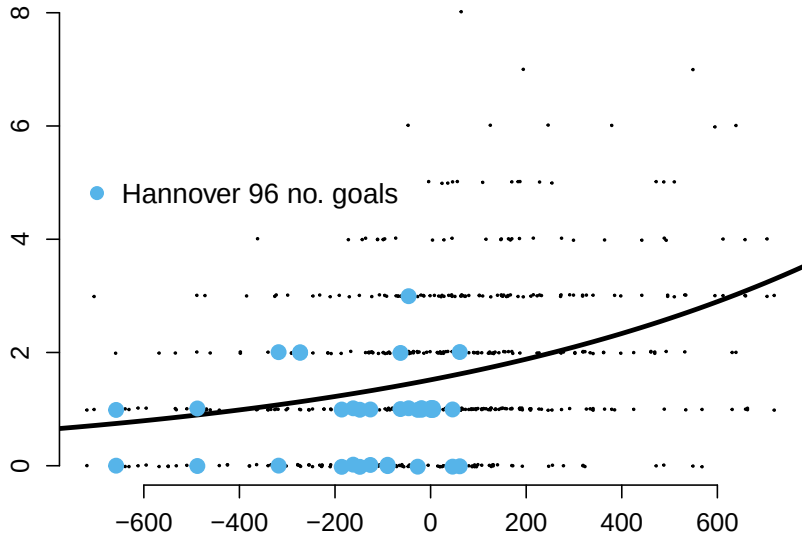
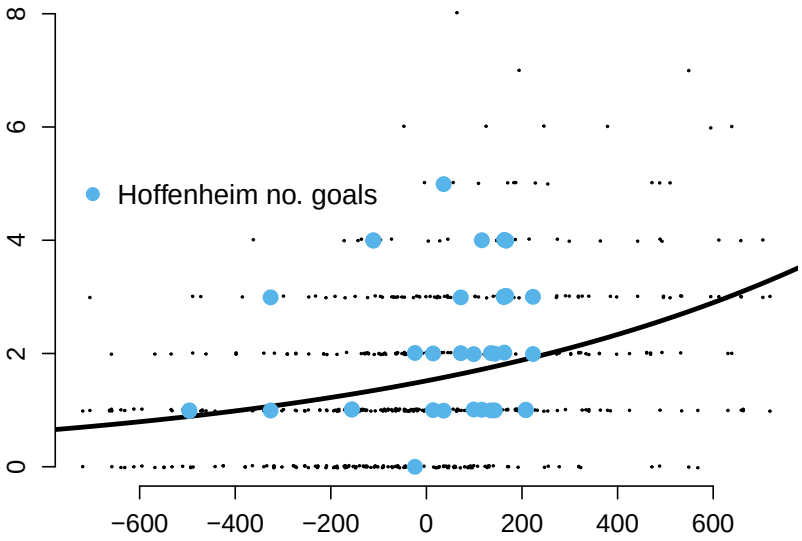
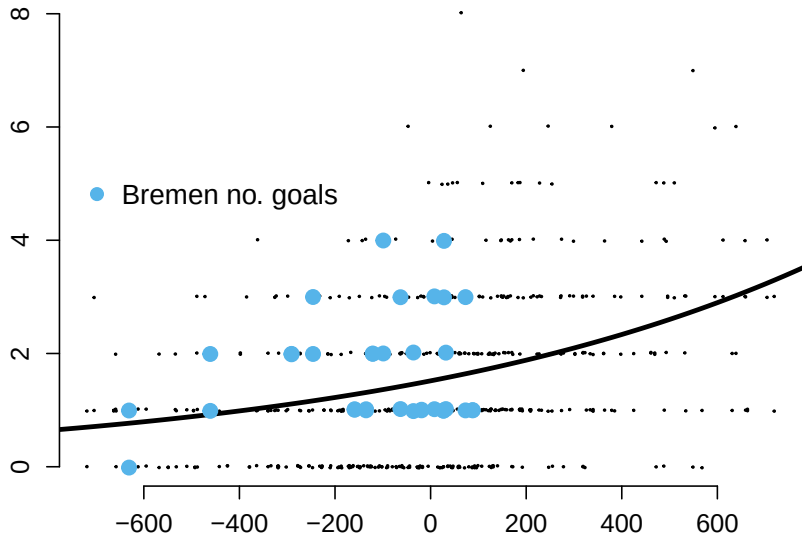


Figure: Poisson GLM $\mathbb{E}(\text{goals}) = e^{0.417+0.001 \cdot \text{mvdiff}}$ fitted to the 18/19 Bundesliga data.









⇝ some teams overall underperformed in this season, others overperformed

- ⇝ each of the 18 teams is associated with 34 of the 612 data points
- ⇝ some teams were systematically better/worse than implied by the model

Simple model not taking into account team-specific effects:

$$\mathbb{E}(\text{goals}) = e^{\beta_0 + \beta_1 \cdot \text{mvdiff}}$$

$$\ell = -957,\ \text{AIC} = 1918,\ \text{BIC} = 1927$$

Alternative model with team-specific effects modelled using dummy variables[43]:

$$\mathbb{E}(\text{goals}) = e^{\beta_0 + \beta_1 \cdot \text{mvdiff} + \beta_2 \cdot I_{\text{B04}} + \ldots + \beta_{18} \cdot I_{\text{WOB}}}$$

$$\ell = -935,\ \text{AIC} = 1908,\ \text{BIC} = 1992$$

Problem with the the latter model: fairly many parameters (i.e. rather complex).

[43] here with “AUG” (Augsburg) representing the reference category

$$\mathbb{E}(\text{goals}) = e^{\beta_0 + \beta_1 \cdot \text{mvdiff} + \beta_2 \cdot I_{\text{B04}} + \ldots + \beta_{18} \cdot I_{\text{WOB}}}$$

The model as it stands effectively estimates one intercept for each team.

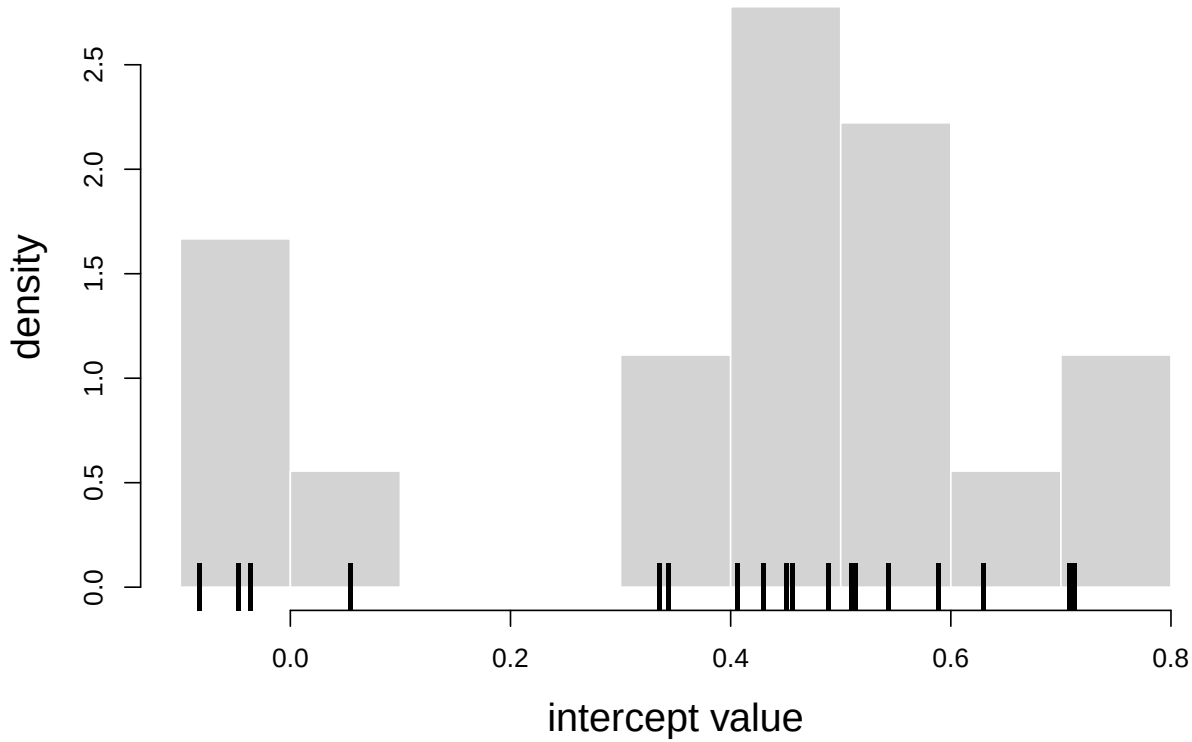


Figure: Histogram of the 18 intercepts estimated in the Poisson GLM for the football data.

Alternative approach, with a so-called **mixed model** where for each team the intercept is assumed to be *a realisation of a random variable*:

$$\mathbb{E}(\text{goals}_{ij}) = e^{\beta_{0,i} + \beta_1 \cdot \text{mvdiff}_{ij}}, \quad i = 1, \ldots, 18, \quad j = 1, \ldots, 34$$

$$\beta_{0,i} \sim \mathcal{N}(\beta_0, \sigma^2), \quad i = 1, \ldots, 18$$

- i refers to the team, j refers to the matchday[44]
- each team has its own intercept $\beta_{0,i}$, a so-called **random effect**
- we estimate, in addition to β_1, the parameters β_0 and σ^2

Equivalent specification, separating population-wide intercept β_0 and unit-specific random deviation:

$$\mathbb{E}(\text{goals}_{ij}) = e^{\beta_0 + \beta_1 \text{mvdiff}_{ij} + \gamma_{0i}}, \quad \gamma_{0i} \sim \mathcal{N}(0, \sigma^2)$$

[44] e.g. $\text{goals}_{3,11}$ would be the number of goals scored by the third team ("BAY") on matchday 11

Mixed model fitted in the football example

Skipping the technical details for now, the model was estimated as follows:

$$\mathbb{E}(\text{goals}_{ij}) = e^{0.403+0.001\cdot\text{mvdiff}_{ij}+\gamma_{0i}}, \quad \gamma_{0i} \sim \mathcal{N}(0, 0.17^2)$$

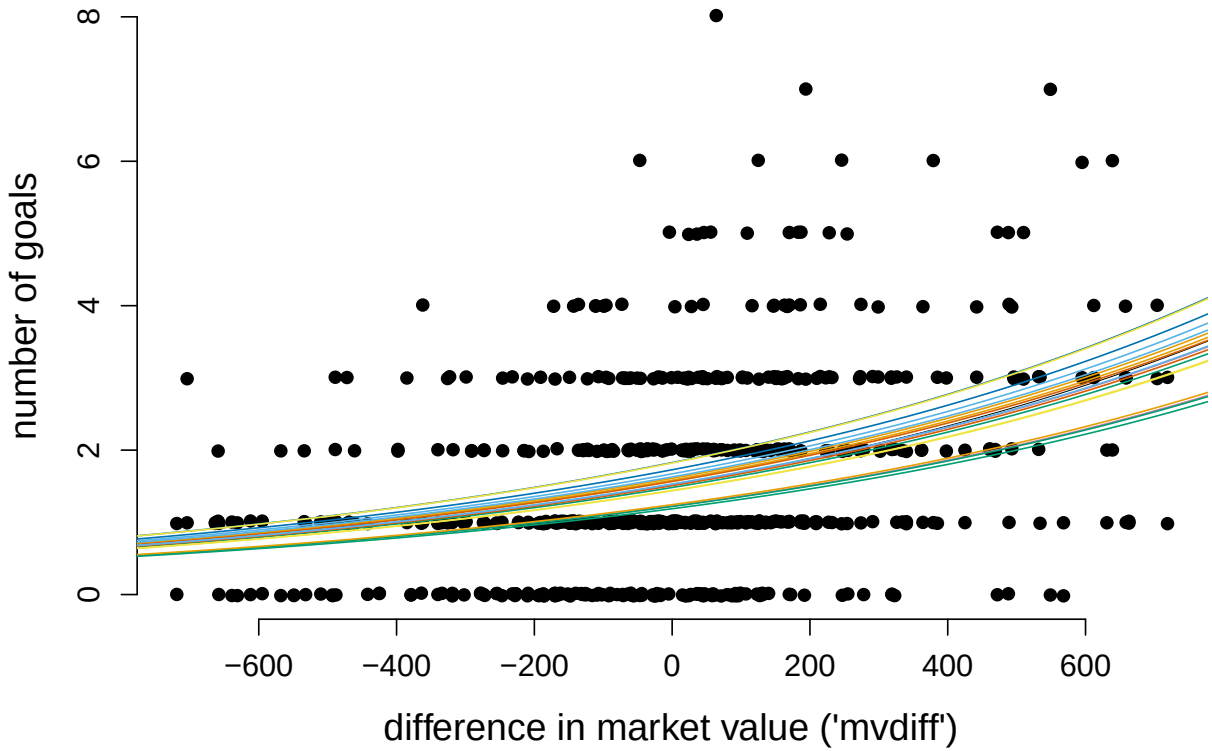


Figure: The 18 regression functions as inferred under the fitted mixed model.

Mixed models as parsimonious approaches for addressing heterogeneity

$$\mathbb{E}(\text{goals}_{ij}) = e^{\beta_0 + \beta_1 \text{mvdiff}_{ij} + \gamma_{0i}}, \quad \gamma_{0i} \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, 18$$

Let's summarise the main idea:

- some data sets have a **hierarchical structure**, where a bunch of data points is available for each of several observational units
- the regression function may vary slightly across the units
- such heterogeneity is most parsimoniously modelled using random effects
- it often also makes conceptual sense to frame systematic deviation from a population model as a *random* rather than a particular *subject's* effect[45]
- mixed models borrow strength across units yet account for heterogeneity

[45] e.g. Bremen's overperformance *last season* surely doesn't mean they *consistently* overperform

Potential pitfalls when neglecting hierarchical structures

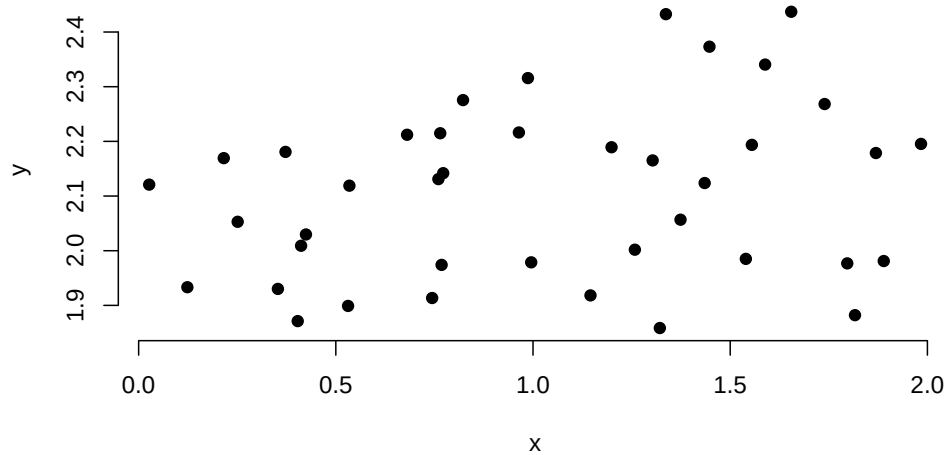
Not accounting for hierarchical structures...

1. ...is (clearly) inadequate if interest lies in the variation across units
2. ...may invalidate standard errors, CIs and hypothesis tests
   (when neglecting heterogeneity the independence assumption is violated!)
3. ...may in extreme cases lead to Simpson's paradox

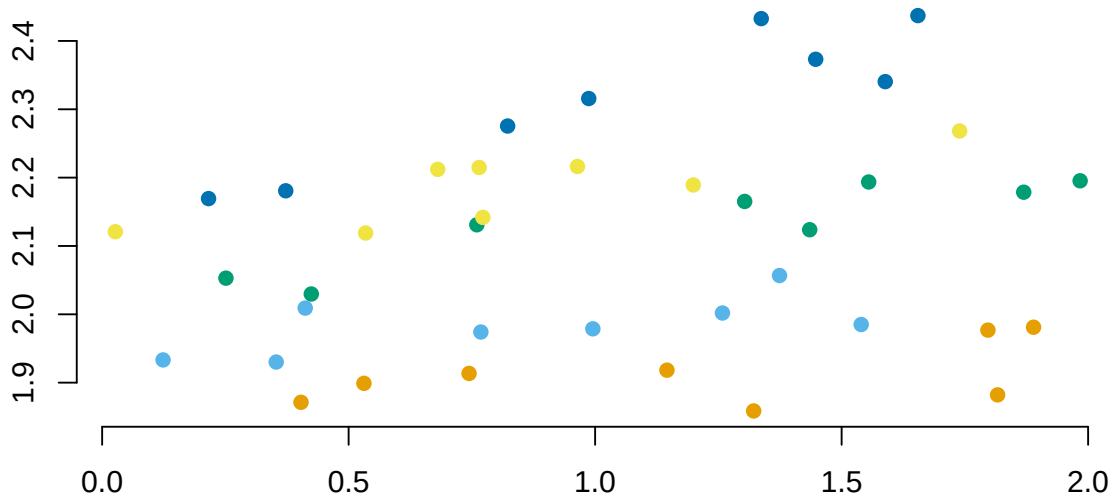
2. and 3. will be illustrated on the next few slides.

Uncertainty quantification neglecting hierarchical structure — toy example

x	y	unit
0.83	1.99	A
2.31	2.03	A
1.20	1.91	A
2.69	2.61	A
2.84	1.99	A
0.12	2.42	A
1.52	1.34	A
2.67	2.30	A
1.65	2.01	B
1.32	2.02	B
...	...	...
2.35	1.85	E
0.28	2.50	E
1.40	3.18	E
1.53	1.84	E




```
> mod<-lm(y~x)
> summary(mod)

Residuals:
     Min       1Q   Median       3Q      Max
-0.27720 -0.12646  0.02025  0.11538  0.30571

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.03674    0.05189   39.25   <2e-16 ***
x            0.06751    0.04441    1.52    0.137
```

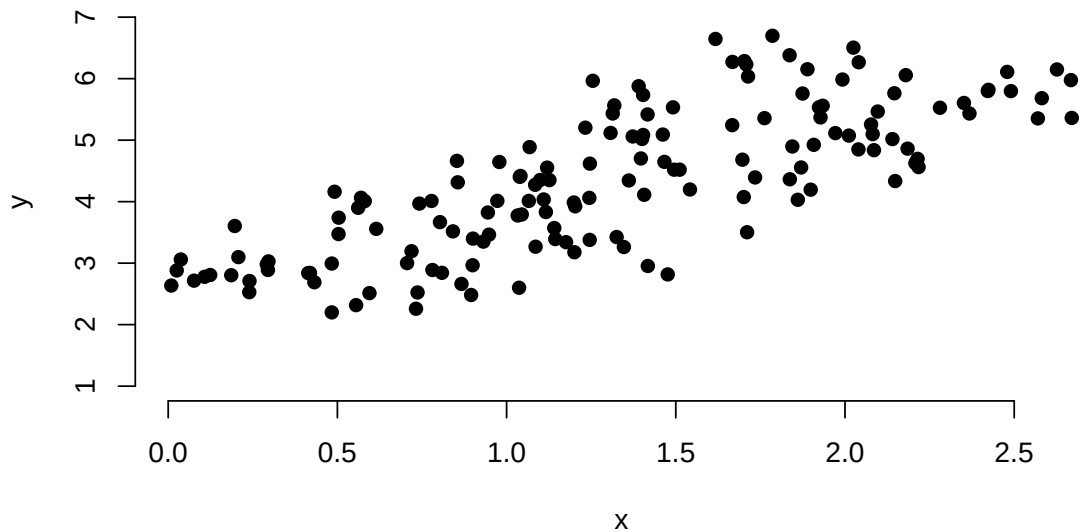



```
> mod<-lm(y~x+unit)
> summary(mod)

Residuals:
      Min        1Q    Median        3Q       Max
-0.083414 -0.023089  0.005116  0.028026  0.092471

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.80798    0.02176  83.102  < 2e-16 ***
x            0.08677    0.01287   6.743 9.50e-08 ***
unitB        0.10167    0.02203   4.614 5.41e-05 ***
unitC        0.22190    0.02156  10.291 5.56e-12 ***
unitD        0.30489    0.02208  13.807 1.69e-15 ***
unitE        0.41626    0.02165  19.226  < 2e-16 ***
```

Simpson's paradox — toy example

Displayed on the right
is an artificial data set:

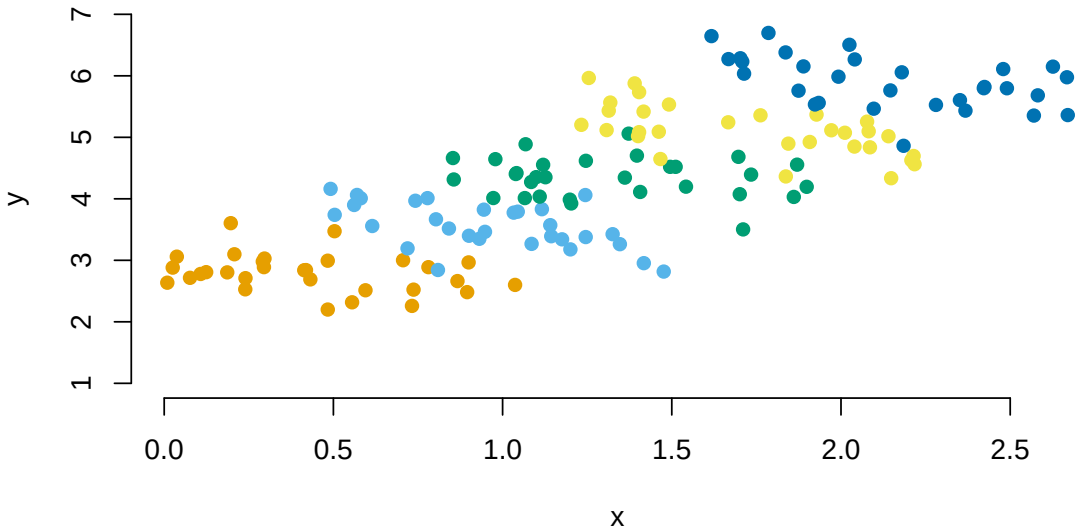
x	y
0.08	2.72
0.90	3.06
1.04	2.70
0.30	3.06
0.19	2.82
0.04	3.06
0.20	3.62
...	...
2.35	5.84
2.18	6.28



$\rightsquigarrow$ clearly a positive effect of X on Y, right?

Same data but complemented with information on units:

x	y	unit
0.08	2.72	A
0.90	3.06	A
1.04	2.70	A
0.30	3.06	A
0.19	2.82	A
0.04	3.06	A
0.20	3.62	A
...	...	
2.35	5.84	E
2.18	6.28	E



$\rightsquigarrow$ unit-specific intercepts & *negative* effect of X on Y!

Things we've learned so far

- it can be crucially important to account for hierarchical data structure
- especially when there are many units, it may make sense to use random effects[46] as to parsimoniously model heterogeneity across units

[46] as opposed to unit-specific dummy variables

Chapter 7: Mixed models

7.2 Linear mixed models (LMMs)

Some terminology

Regarding the data:

- now interested in hierarchical structures (also repeated measurements[47])
- specifically, situations where we have n_i observations for each of m units

Regarding the model for such data:

- different labels[48]: **mixed model**, **hierarchical model**, or **multi-level model**
- "mixed" as the model contains both fixed effects and random effects

Regarding the effects being modelled:

- fixed effects describe (population-level) covariate-response relationship
- random effects model unexplained variation across units

[47] panel data, where several consecutive observations are made for each of a set of individuals — i.e. repeated measurements — constitute a special case of a hierarchical structure

[48] depending on context and/or strand of literature

Potential applications

Hierarchical structures — with several data points collected for each of multiple units — occur all over the place:

- students' performances measured in different universities
- patients treated in different hospitals
- multiple test results for each of several patients
- flat rent prices from multiple cities
- behavioural data from several animals
- etc.

Baseline linear regression model in hierarchical setting

The data from now on are vectors of the form

$$(Y_{ij}, \mathbf{x}_{ij}),$$

where $i = 1, \ldots, m$ indicates the observational unit and $j = 1, \ldots, n_i$ the observation therein.

In a setting with such hierarchically structured data — not yet accounting for heterogeneity — the linear regression model can simply be written as

$$Y_{ij} = \beta_0 + \beta_1 x_{ij1} + \ldots + \beta_p x_{ijp} + \epsilon_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n_i$$

$$Y_{ij} = \beta_0 + \beta_1 x_{ij1} + \ldots + \beta_p x_{ijp} + \epsilon_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n_i$$

In matrix notation, with *one equation for each unit*[49]:

$$\mathbf{Y}_i = \mathbf{X}_i \boldsymbol{\beta} + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, m,$$

with

$$\mathbf{Y}_i = \begin{pmatrix} Y_{i1} \\ \vdots \\ Y_{in_i} \end{pmatrix}, \quad \mathbf{X}_i = \begin{pmatrix} 1 & x_{i11} & \ldots & x_{i1p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{in_i1} & \ldots & x_{in_ip} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\epsilon}_i = \begin{pmatrix} \epsilon_{i1} \\ \vdots \\ \epsilon_{in_i} \end{pmatrix}$$

[49] alternatively, we could stack the vectors/matrices such that there would be only one equation covering all observations — but explicitly distinguishing units will be useful later on!

First step: let's add a unit-specific intercept

To account for heterogeneity that manifests itself in the height of the regression function, we can add a unit-specific deviation from the population intercept:

$$Y_{ij} = \beta_0 + \beta_1 x_{ij1} + \ldots + \beta_p x_{ijp} + \gamma_{0i} + \epsilon_{ij}$$

As seen before in the football example, we could now...

a) ...estimate γ_{0i} for each $i = 1, \ldots, m$ (dropping the β_0 from the model)

b) ...assume say $\gamma_{0i} \sim \mathcal{N}(0, \sigma^2)$ and estimate σ^2 (only!)

In many cases, especially when m is large, b) will be preferable.

$$Y_{ij} = \beta_0 + \beta_1 x_{ij1} + \ldots + \beta_p x_{ijp} + \gamma_{0i} + \epsilon_{ij}$$

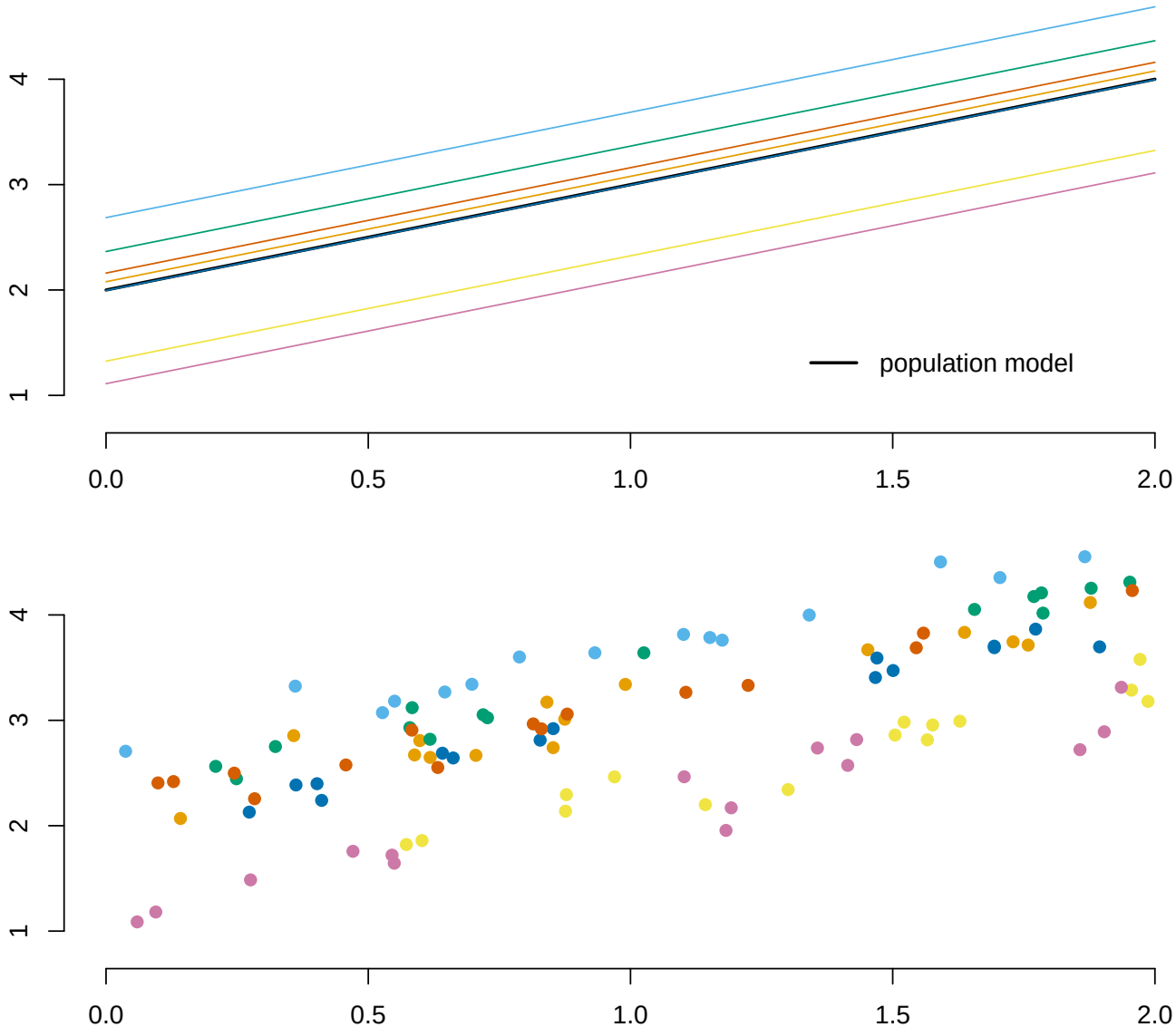
In matrix notation:

$$\mathbf{Y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{U}_i \boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, m,$$

with $\mathbf{Y}_i$, $\mathbf{X}_i$, $\boldsymbol{\beta}$ and $\boldsymbol{\epsilon}_i$ exactly as before and

$$\mathbf{U}_i = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad \boldsymbol{\gamma}_i = (\gamma_{0i})$$

Illustration of the random-intercept model



Second step: unit-specific covariate effects

To also address potential heterogeneity in the covariate effects, we can further add unit-specific deviations from the overall slope(s):

$$Y_{ij} = \beta_0 + \beta_1 x_{ij1} + \ldots + \beta_p x_{ijp} + \gamma_{0i} + \gamma_{1i} x_{ij1} + \ldots + \gamma_{pi} x_{ijp} + \epsilon_{ij}$$

Again, we could now...

a) ...estimate $\boldsymbol{\gamma}_i = (\gamma_{0i}, \ldots, \gamma_{pi})$ for each $i = 1, \ldots, m$ (dropping all βs)

b) ...assume say $\boldsymbol{\gamma}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and estimate the var.-cov. matrix $\mathbf{Q}$

In this setting, a) would effectively amount to separately fitting m regression models $\rightsquigarrow$ not parsimonious at all!

$$Y_{ij} = \beta_0 + \beta_1 x_{ij1} + \ldots + \beta_p x_{ijp} + \gamma_{0i} + \gamma_{1i} x_{ij1} + \ldots + \gamma_{pi} x_{ijp} + \epsilon_{ij}$$

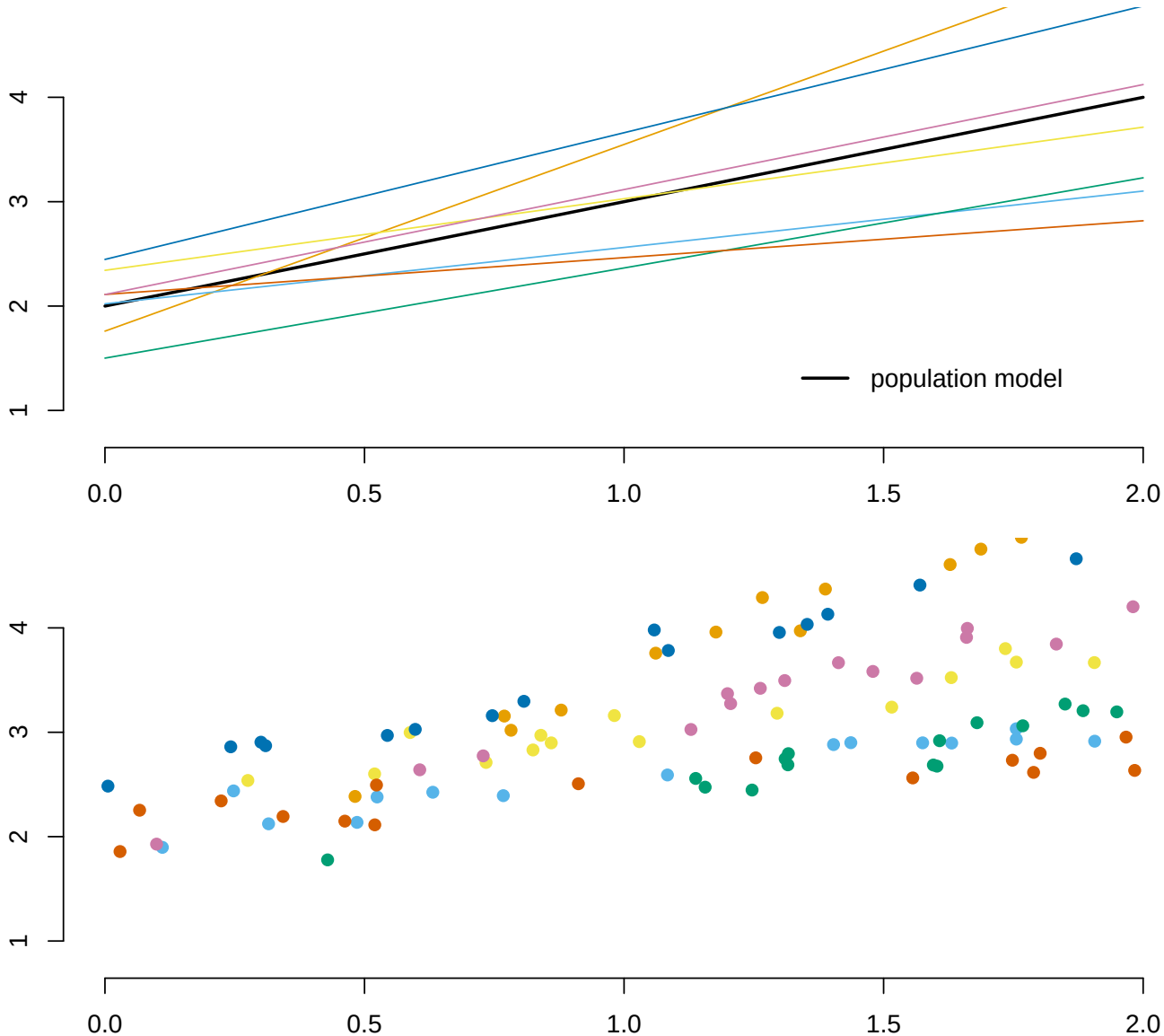
In matrix notation:

$$\mathbf{Y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{U}_i \boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, m,$$

with $\mathbf{Y}_i$, $\mathbf{X}_i$, $\boldsymbol{\beta}$ and $\boldsymbol{\epsilon}_i$ exactly as before and now

$$\mathbf{U}_i = \begin{pmatrix} 1 & x_{i11} & \ldots & x_{i1p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{in_i1} & \ldots & x_{in_ip} \end{pmatrix}, \quad \boldsymbol{\gamma}_i = \begin{pmatrix} \gamma_{0i} \\ \gamma_{1i} \\ \vdots \\ \gamma_{pi} \end{pmatrix}$$

Illustration of the random-intercept & random-slope model



Putting it all together: general model formulation

The **linear mixed model** (LMM), with **fixed effects** β and **random effects** γ_i, is defined as

$$\mathbf{Y}_i = \mathbf{X}_i\beta + \mathbf{U}_i\gamma_i + \epsilon_i,$$

for units $i = 1, \ldots, m$, the i–th of which comprises n_i observations. The columns of $\mathbf{U}_i$ usually are a subset of the columns of $\mathbf{X}_i$[50].

Distributional assumptions:

$$\gamma_i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad \epsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{n_i})$$

[50] e.g. only the intercept, or the intercept plus 1-2 of the slope coefficients could be random effects

A concrete example — sleep deprivation study

subject ID	reaction time in ms	nights of sleep deprivation
308	249.5600	0
308	258.7047	1
⋮	⋮	⋮
308	466.3535	9
333	283.8424	0
333	289.5550	1
⋮	⋮	⋮
333	362.0428	9
⋮	⋮	⋮
372	269.4117	0
372	273.4740	1
⋮	⋮	⋮
372	364.1236	9

Table: Sleep deprivation study with 18 participants, each of whom was subjected to nine consecutive nights with only three hours of sleep.

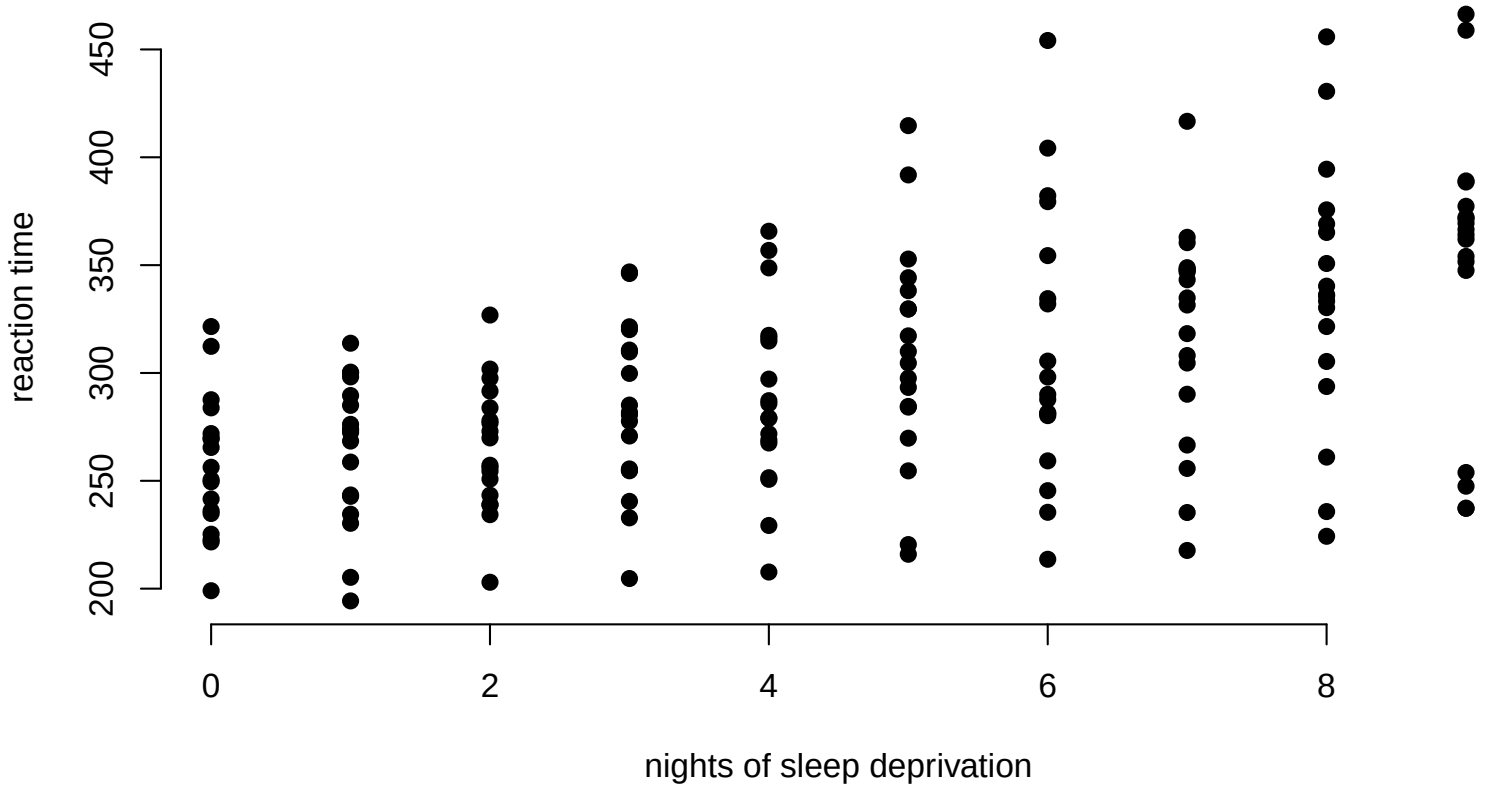


Figure: Visualisation of the influence of the magnitude of sleep deprivation on reaction times, here completely neglecting the hierarchical structure of the data set.

```
> mod<-lm(reaction~nights)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  251.405      6.610  38.033  < 2e-16 ***
nights        10.467      1.238   8.454 9.89e-15 ***
```

Fitted linear model:

$$\mathbb{E}(\text{reaction}) = 251.4 + 10.7 \cdot \text{nights}$$

Residual analysis for the fitted linear model

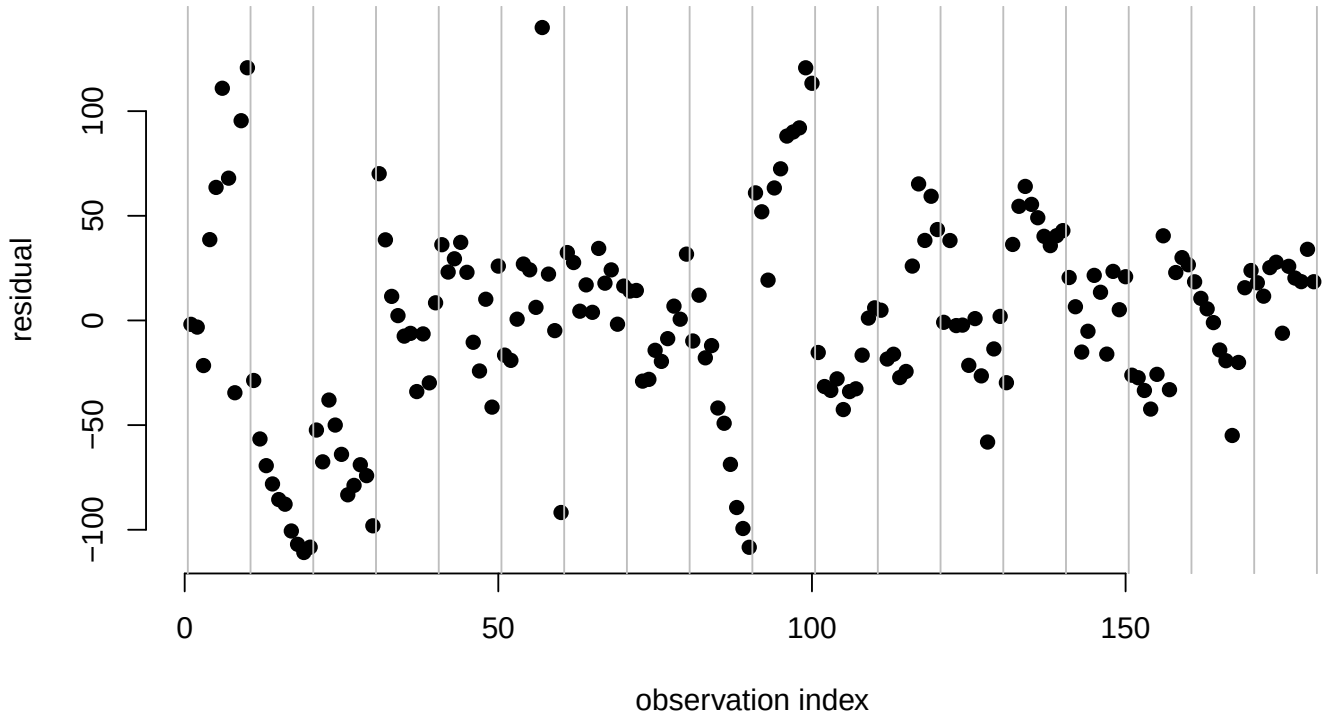


Figure: Residuals of the linear model plotted against observation index.

What we see is a problem that very often occurs when neglecting hierarchical structures: the independence assumption is violated!

Exploratory data analysis w.r.t. heterogeneity across observational units

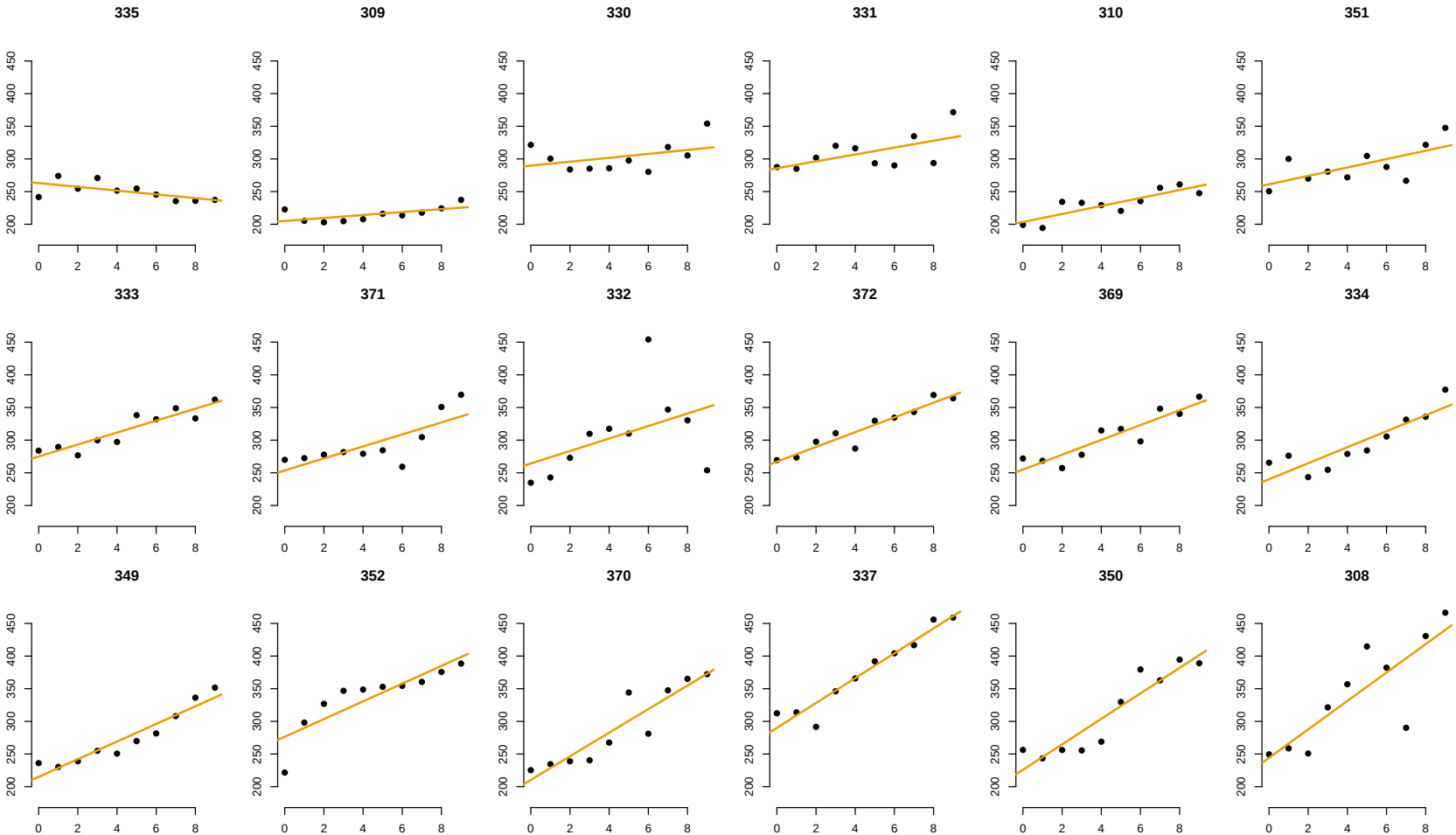


Figure: Subject-specific linear models illustrating heterogeneity in both intercept *and* slope.

```
> mod<-lm(reaction~nights+subject+nights*subject)
> summary(mod)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    244.193     15.042  16.234  < 2e-16 ***   nights:subject309  -19.503      3.985  -4.895 2.61e-06 ***
nights          21.765      2.818   7.725 1.74e-12 ***   nights:subject310  -15.650      3.985  -3.928 0.000133 ***
subject309     -39.138     21.272  -1.840 0.067848 .     nights:subject330  -18.757      3.985  -4.707 5.84e-06 ***
subject310     -40.708     21.272  -1.914 0.057643 .     nights:subject331  -16.499      3.985  -4.141 5.88e-05 ***
subject330      45.492     21.272   2.139 0.034156 *     nights:subject332  -12.198      3.985  -3.061 0.002630 **
subject331      41.546     21.272   1.953 0.052749 .     nights:subject333  -12.623      3.985  -3.168 0.001876 **
subject332      20.059     21.272   0.943 0.347277       nights:subject334   -9.512      3.985  -2.387 0.018282 *
subject333      30.826     21.272   1.449 0.149471       nights:subject335  -24.646      3.985  -6.185 6.07e-09 ***
subject334      -4.030     21.272  -0.189 0.850016       nights:subject337   -2.739      3.985  -0.687 0.492986
subject335      18.842     21.272   0.886 0.377224       nights:subject349   -8.271      3.985  -2.076 0.039704 *
subject337      45.911     21.272   2.158 0.032563 *     nights:subject350   -2.261      3.985  -0.567 0.571360
subject349     -29.081     21.272  -1.367 0.173728       nights:subject351  -15.331      3.985  -3.848 0.000179 ***
subject350     -18.358     21.272  -0.863 0.389568       nights:subject352   -8.198      3.985  -2.057 0.041448 *
subject351      16.954     21.272   0.797 0.426751       nights:subject369  -10.417      3.985  -2.614 0.009895 **
subject352      32.179     21.272   1.513 0.132535       nights:subject370   -3.709      3.985  -0.931 0.353560
subject369      10.775     21.272   0.507 0.613243       nights:subject371  -12.576      3.985  -3.156 0.001947 **
subject370     -33.744     21.272  -1.586 0.114870       nights:subject372  -10.467      3.985  -2.627 0.009554 **
subject371       9.443     21.272   0.444 0.657759
subject372      22.852     21.272   1.074 0.284497
```

Fitted linear model with subject-specific intercepts and slopes:

$$\mathbb{E}(\text{reaction}) = 244.2 + 21.8 \cdot \text{nights} - 39.1 \cdot I_{\text{subject309}} - 40.7 \cdot I_{\text{subject310}} + \ldots$$
$$- 19.5 \cdot \text{nights} \cdot I_{\text{subject309}} - 15.7 \cdot \text{nights} \cdot I_{\text{subject310}} + \ldots$$

Very many parameters & difficult to interpret!!

Natural linear *mixed* model for the sleep deprivation data:

$$\mathbb{E}(\text{reaction}_{ij}) = \beta_0 + \beta_1 \cdot \text{nights}_j + \gamma_{0i} + \gamma_{1i} \cdot \text{nights}_j$$

$$i = 1, \ldots, 18, \quad j = 1, \ldots, 10$$

$$\boldsymbol{\gamma}_i = \begin{pmatrix} \gamma_{0i} \\ \gamma_{1i} \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$

Different R packages allow to fit such models — `lme4` in particular is very popular.

```
> library(lme4)
> mod<-lmer(reaction~nights+(nights|subject))
> summary(mod)

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subject  (Intercept) 611.90   24.737
          nights       35.08    5.923   0.07

Fixed effects:
            Estimate Std. Error t value
(Intercept)  251.405      6.824  36.843
nights        10.467      1.546   6.771
```

Fitted linear mixed model:

$$\mathbb{E}(\text{reaction}_{ij}) = 251.4 + 10.5 \cdot \text{nights}_j + \gamma_{0i} + \gamma_{1i} \cdot \text{nights}_j$$

$$\boldsymbol{\gamma}_i = \begin{pmatrix} \gamma_{0i} \\ \gamma_{1i} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 611.9 & 10.3 \\ 10.3 & 35.1 \end{pmatrix} \right)$$

The fixed effects 251.4 and 10.5 can be interpreted as the **population means** of intercept and slope, respectively.

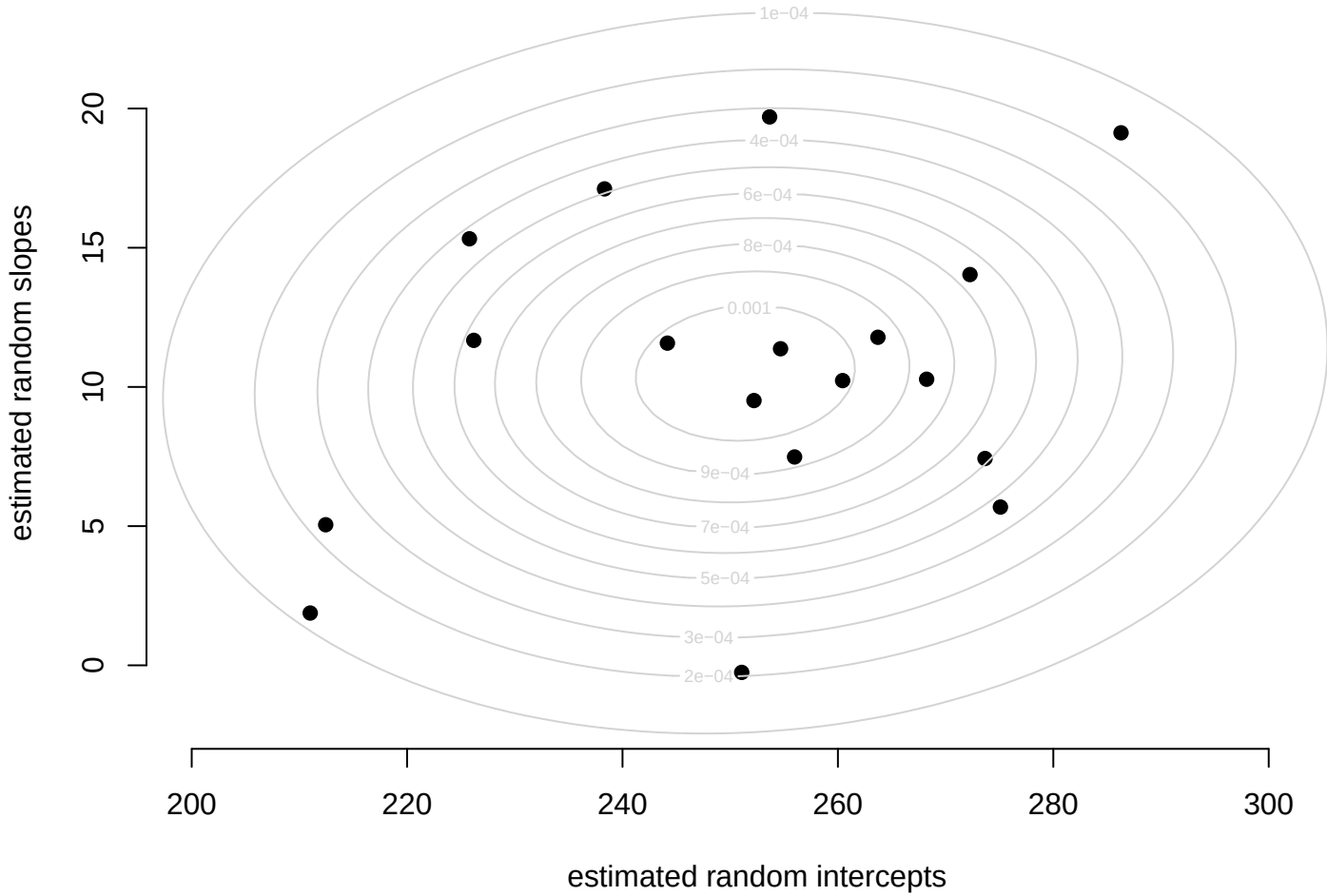


Figure: Estimated subject-specific intercepts and slopes obtained using `ranef(...)` from `lme4`— the bivariate normal random effects distribution is indicated by the contour lines.

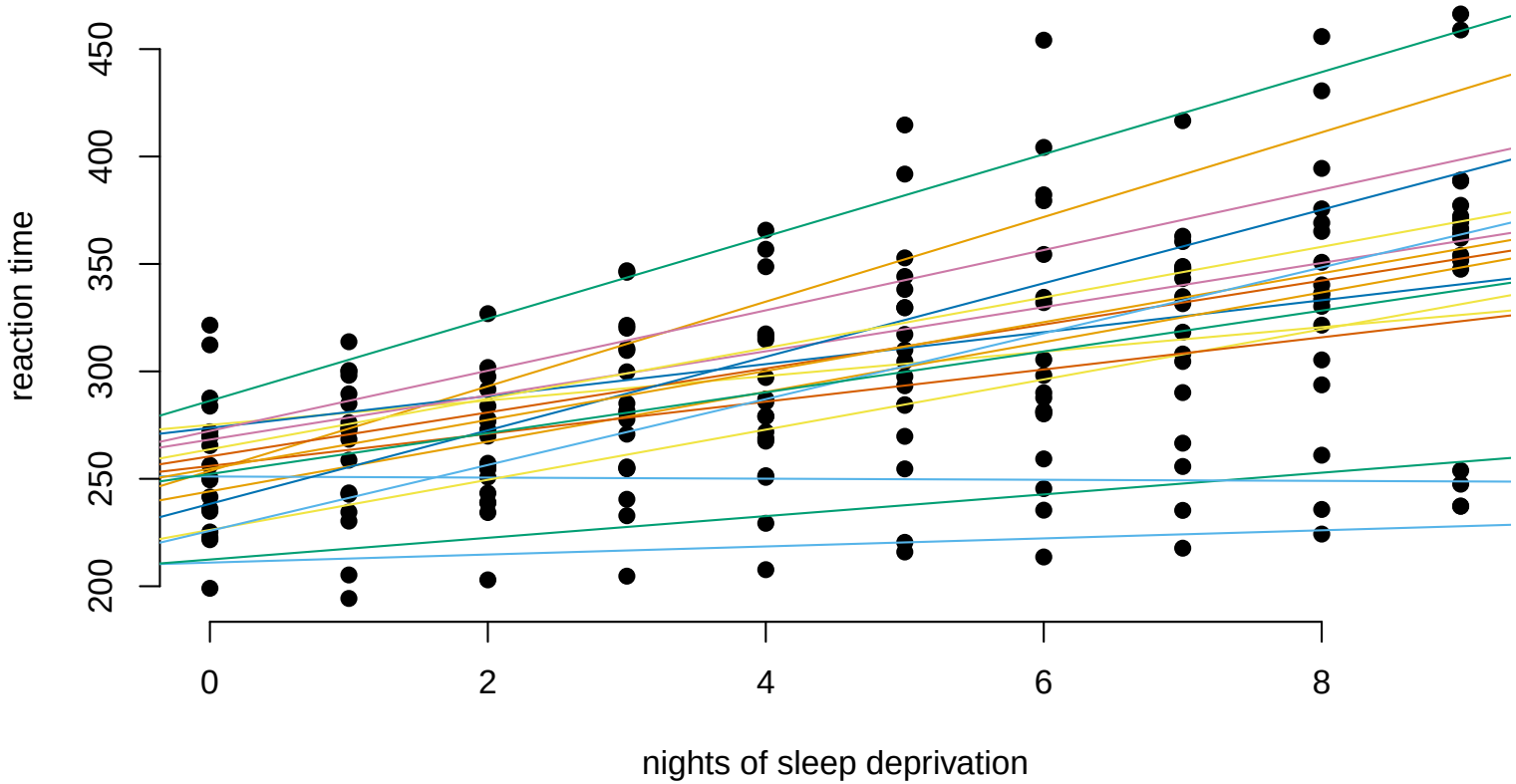


Figure: The 18 subject-specific regression functions under the fitted LMM.

Some remarks on the `lmer` syntax

When the (factor) variable `g` indicates the observational units, then the general `lmer` syntax for fitting an LMM is

```
lmer(response ~ FEexpr + (REexpr | g))
```

- `FEexpr` is a formula for a linear predictor, just as it would be used in `lm` or `glm`, comprising only fixed effects
- `REexpr` is the analogous formula but for the random effects, with one realisation for each level of the factor `g`

Table: Examples for the `lmer` syntax.

formula	meaning
`x+(x|g)`	random intercept, random slope
`x+(1|g)`	random intercept, fixed slope
`x+(x-1|g)`	fixed intercept, random slope

Table: Comparison of various candidate models for the sleep deprivation data.

predictor of the model fitted	short description	number of param.	AIC	BIC
`nights`	basic LM	3	1906.3	1915.9
`nights+subject+ nights*subject`	LM with subject-specific intercepts and slopes	37	**1711.9**	1830.0
`nights+ (nights|subject)`	LMM with random intercepts and slopes	6	1755.6	**1774.8**

On the next few slides, we briefly sketch parameter estimation in LMMs — rather technical & *not exam-relevant*, will only be shown if time allows.

LMM with m unit-specific matrix equations:

$$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{U}_i\boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, m$$

Stacking the m equations, we obtain a *single* matrix equation:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U}\boldsymbol{\gamma} + \boldsymbol{\epsilon},$$

with $\boldsymbol{\beta}$ as before, the vectors

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_m \end{pmatrix}, \quad \boldsymbol{\gamma} = \begin{pmatrix} \boldsymbol{\gamma}_1 \\ \vdots \\ \boldsymbol{\gamma}_m \end{pmatrix}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_m \end{pmatrix},$$

each of length $\sum_{i=1}^{m} n_i$, and the design matrices

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_m \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{U}_m \end{pmatrix}.$$

We can thus write the LMM as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U}\boldsymbol{\gamma} + \boldsymbol{\epsilon}, \tag{3}$$

with

$$\begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \right), \tag{4}$$

where

$$\mathbf{G} = \begin{pmatrix} \mathbf{Q} & & 0 \\ & \ddots & \\ 0 & & \mathbf{Q} \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \sigma^2\mathbf{I}_{n_1} & & 0 \\ & \ddots & \\ 0 & & \sigma^2\mathbf{I}_{n_m} \end{pmatrix}$$

Equations (1) and (2) together constitute a very general form which is often useful — for example, certain nonparametric models can be framed as such a model[51].

[51] then with different specifications of the matrices involved than in case of the LMMs seen so far

Maximum likelihood estimation of fixed and random effects[52]

Re-expressing the LMM as follows,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \underbrace{\mathbf{U}\boldsymbol{\gamma} + \boldsymbol{\epsilon}}_{=\boldsymbol{\epsilon}^*} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}^*,$$

the so-called **marginal model formulation** is obtained as

$$\mathbf{Y} \sim \mathcal{N}\big(\mathbf{X}\boldsymbol{\beta}, \underbrace{\mathbf{U}\mathbf{G}\mathbf{U}^t + \mathbf{R}}_{=\mathbf{V}}\big).$$

The estimators of the fixed and random effects are obtained as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t\hat{\mathbf{V}}^{-1}\mathbf{X})^{-1}\mathbf{X}^t\hat{\mathbf{V}}^{-1}\mathbf{Y} \quad \text{and} \quad \hat{\boldsymbol{\gamma}} = \hat{\mathbf{G}}\mathbf{U}^t\hat{\mathbf{V}}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}),$$

respectively, plugging in the maximum likelihood estimator $\hat{\mathbf{V}}$ for $\mathbf{V}$ — the latter is relatively easily obtained using profile likelihood.

[52] skipping the rather tedious technical details
(cf. Chapter 7 in “Regression: Models, Methods and Applications” by Fahrmeir et al., 2013)

Chapter 7: Mixed models

7.3 Generalised linear mixed models (GLMMs)

From LMMs to GLMMs — preliminary remarks

In terms of the *model formulation*, the extension from LMMs to GLMMs is completely straightforward, as it only concerns the linear predictor.

We have in fact already seen a GLMM, namely the random-intercept model for the football data at the beginning of this chapter.

Estimation of GLMMs is however much more involved — not covered here.

A **generalised linear mixed model** (GLMM), with fixed effects β and random effects γ_i, is specified as follows:

1. the response variables have a hierarchical structure, with m observational units, the i–th of which comprises n_i observations:

$$Y_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n_i$$

2. conditional on the γ_i, the response variables Y_{ij} are independent of each other and follow some distribution from the exponential family

3. the conditional mean of the response is linked to the linear predictor via an invertible and differentiable link function g,

$$g\big(\mathbb{E}(Y_{ij})\big) = \eta_{ij}$$

4. the linear predictor involves both fixed and random effects,

$$\boldsymbol{\eta}_i = \mathbf{X}_i\beta + \mathbf{U}_i\gamma_i, \quad i = 1, \ldots, m$$

5. the random effects are iid with

$$\gamma_i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$$

Fitting GLMMs using `glmer` from the `lme4` package

```
> library(lme4)
> mod<-glmer(goals~mvdiff+(1|team),family=poisson)
> summary(mod)

Random effects:
 Groups Name        Variance Std.Dev.
 Team   (Intercept) 0.02829  0.1682

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.4025963  0.0523257   7.694 1.43e-14 ***
MWdiff      0.0010419  0.0001586   6.569 5.08e-11 ***
```

Fitted Poisson GLMM:

$$\mathbb{E}(\text{goals}_{ij}) = e^{0.403+0.001\cdot\text{mvdiff}_{ij}+\gamma_{0i}}, \quad \gamma_{0i} \sim \mathcal{N}(0, 0.17^2)$$

Table: Comparison of various candidate models for the football data.

predictor of the model fitted	short description	number of param.	AIC	BIC
`mvdiff`	basic Poisson GLM	2	1918.0	1926.9
`mvdiff+team`	Poisson GLM with subject-specific intercepts	19	**1907.6**	1991.6
`mvdiff+ (1|team)`	Poisson GLMM with random intercepts	3	1911.9	**1925.1**
`mvdiff+team+ mvdiff*team`	Poisson GLM with subject-specific intercepts and slopes	36	1921.9	2080.9
`mvdiff+ (mvdiff|team)`	Poisson GLMM with random intercepts and slopes	5	1913.5	1935.5

A second GLMM example — speed dating

subject ID	match yes (1) / no (0)	gender of person considered (1: male)	attractiveness of partner (scale: 0-10)	shared interests (scale 0-10)
1	0	0	6	5
1	1	0	7	6
⋮	⋮	⋮	⋮	⋮
1	0	0	5	8
2	0	0	5	3
2	0	0	8	6
⋮	⋮	⋮	⋮	⋮
2	0	0	6	8
⋮	⋮	⋮	⋮	⋮
452	0	1	7	1
452	1	1	6	8
⋮	⋮	⋮	⋮	⋮
452	0	1	3	1

Table: Speed dating experiment with 452 participants, each with multiple dates.

```
> mod<-glm(match~gender+attr+shar,family=binomial)
> summary(mod)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.85168    0.16348 -29.677  < 2e-16 ***
gender      -0.22181    0.06805  -3.259  0.00112 **
attr         0.30349    0.02189  13.866  < 2e-16 ***
shar         0.26295    0.01930  13.627  < 2e-16 ***
```

Fitted logistic regression model (a GLM):

$$\texttt{logit}\big(\Pr(\text{match})\big) = -4.85 - 0.22 \cdot \text{gender} + 0.30 \cdot \text{attr} + 0.26 \cdot \text{shar}$$

$$\text{AIC} = 5494.7$$

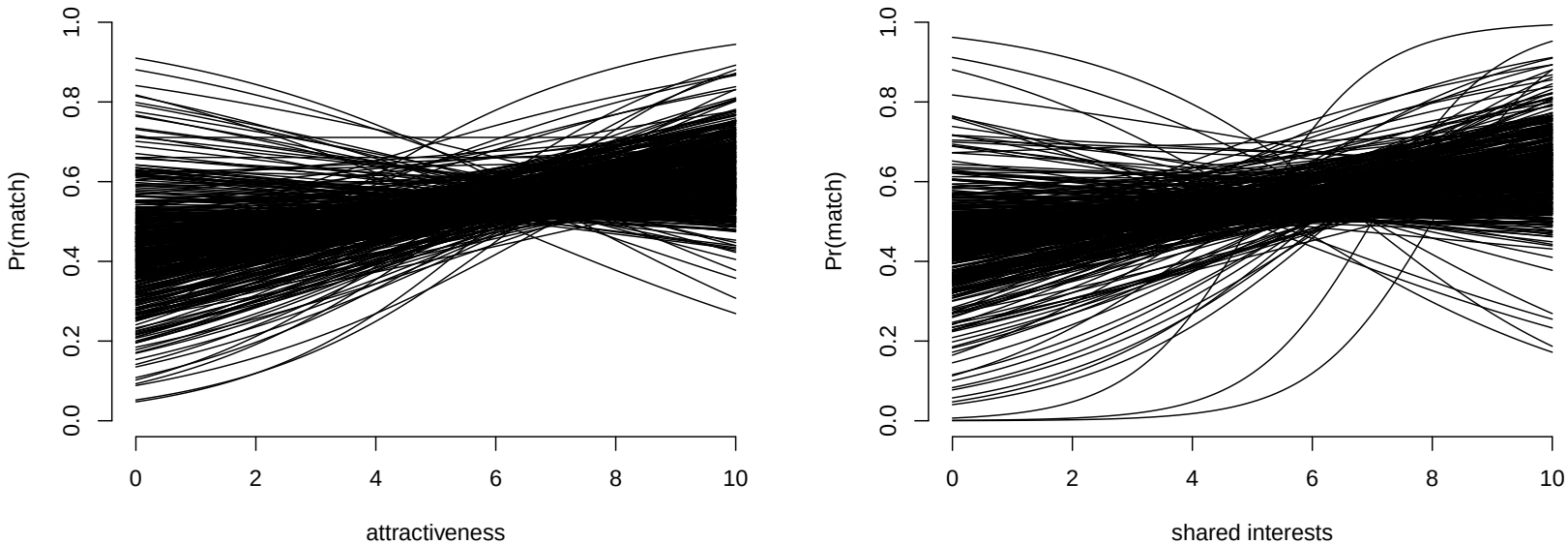


Figure: Subject-specific GLMs $\texttt{logit}\big(\Pr(\text{match})\big) = \beta_0 + \beta_1 \cdot \text{attr}$ and ... $= \beta_0 + \beta_1 \cdot \text{shar}$ illustrating potential heterogeneity across participants.

```
> mod<-glmer(match~gender+attr+shar+(1|iid),family=binomial)
> summary(mod)

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 0.5824   0.7632

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.75389    0.21238 -27.093   <2e-16 ***
gender      -0.25883    0.10465  -2.473   0.0134 *
attr         0.37643    0.02530  14.876   <2e-16 ***
shar         0.32012    0.02256  14.189   <2e-16 ***
```

Fitted *random-intercept* logistic regression model (a GLMM):

$$\texttt{logit}\big(\Pr(\text{match}_{ij})\big) = -5.75 - 0.26 \cdot \text{gender}_i + 0.38 \cdot \text{attr}_{ij} + 0.32 \cdot \text{shar}_{ij} + \gamma_{0i}$$

$$i = 1, \ldots, 452, \quad j = 1, \ldots, n_i, \quad \gamma_{0i} \sim \mathcal{N}(0, 0.76^2), \quad \text{AIC} = 5352.0$$

Interpretation: the random intercept γ_{0i} effectively accounts for the fact that some people are “pickier” than others.

Fitted GLMM including *two additional random slopes*, assuming random effects to be uncorrelated (just for simplicity):

$$\texttt{logit}\big(\Pr(\text{match}_{ij})\big) = -5.65 - 0.25 \cdot \text{gender}_i + 0.36 \cdot \text{attr}_{ij} + 0.32 \cdot \text{shar}_{ij} + \gamma_{0i} + \gamma_{1i} \cdot \text{attr}_{ij} + \gamma_{2i} \cdot \text{shar}_{ij}$$

$$\gamma_{0i} \sim \mathcal{N}(0, 0.25^2), \quad \gamma_{1i} \sim \mathcal{N}(0, 0.09^2), \quad \gamma_{2i} \sim \mathcal{N}(0, 0.05^2), \quad \text{AIC} = 5343.9$$

Interpretation of the random slopes: some people put more emphasis on attractiveness/shared interests than others.

Recommendations regarding the use of mixed models

Exploratory data analysis (EDA) with respect to potential heterogeneity:

- take a look at unit-specific scatterplots
- when possible, fit unit-specific regression models and
  a) plot the unit-specific regression functions
  b) inspect empirical distribution of estimated coefficients

Model specification:

- should be driven by EDA as well as domain expertise
- when there's a random slope, then there should usually also be
  a) a corresponding fixed effect
  b) a random intercept
- don't include too many random slopes — estimation becomes unstable!

Summary of Chapter 7

- in practice, we are often faced with hierarchically structured data:
  - several observations for each of multiple observational units
  - repeated measurements for several subjects ($\rightsquigarrow$ panel data)
- in such cases, basic LMs and GLMs are often invalid due to the independence assumption being violated
- LMMs & GLMMs are parsimonious models that can accommodate unit-specific deviations from an overall pattern
- inferential machinery is much more involved & there are many pitfalls!!

Chapter 8: Extensions & summary

Outlook: Extensions

Things you may come across at some point:

- nonparametric effect modelling
- dealing with overdispersion
- zero-inflated Poisson regression
- multinomial (logistic) regression
- beta regression for proportions
- Bayesian estimation of GLMs
- lasso/boosting/etc.

Some of these extensions/techniques are briefly illustrated in the following.

Chapter 8: Extensions & summary

8.1 Nonparametric effect modelling

Back to square one...

On slides 5 & 6, we went straight from the general regression model formulation

$$Y_i = f(x_{i1}, \ldots, x_{ip}) + \epsilon_i, \quad \mathbb{E}(\epsilon_i) = 0,$$

to the (restrictive) linear model where

$$f(x_{i1}, \ldots, x_{ip}) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}$$

We can go a long way based on this (simplified) model formulation, however there are cases where we want more flexibility.

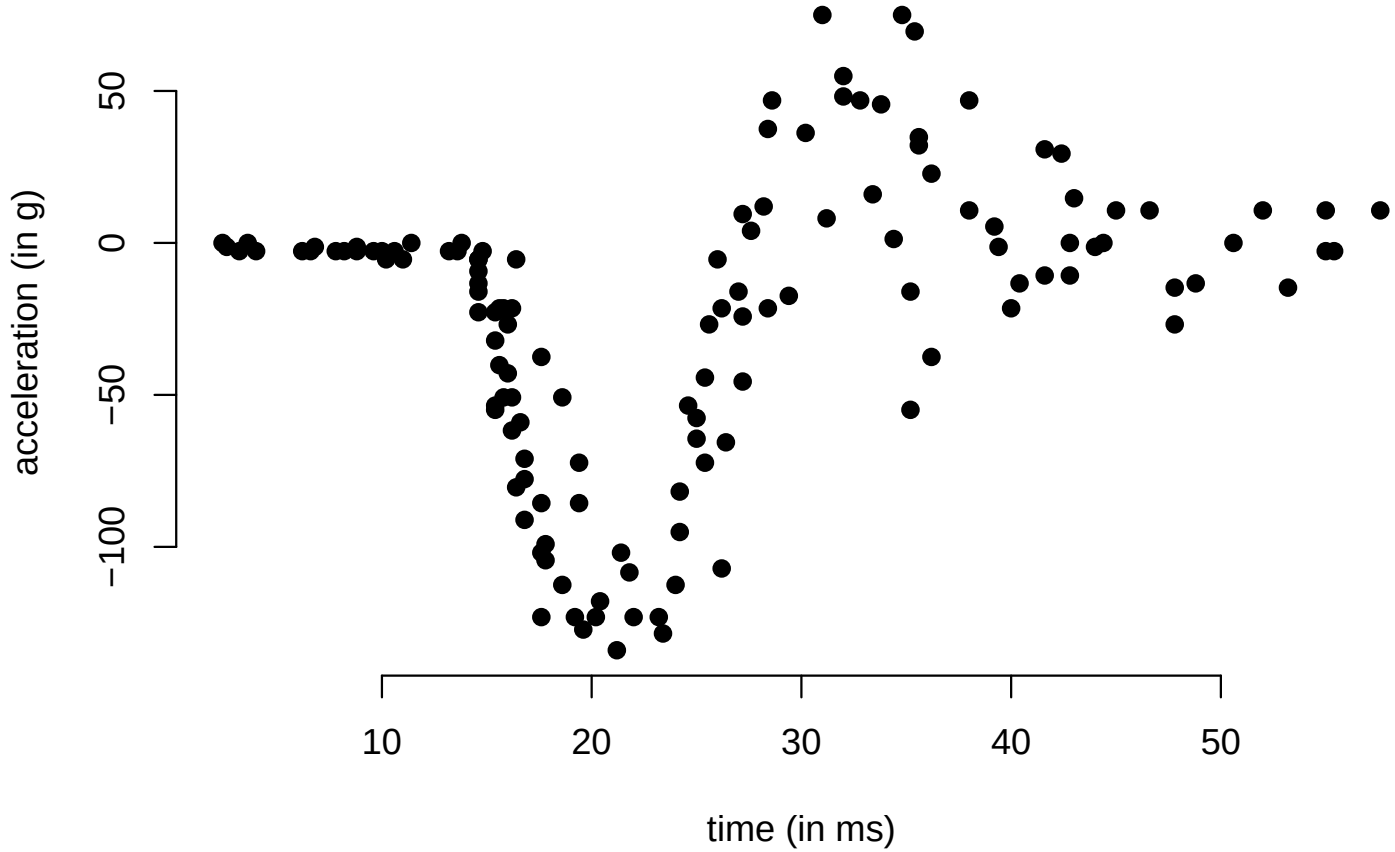


Figure: Acceleration measurements made for the head of a motorcyclist in the milliseconds following a simulated collision.

Flexible modelling with parametric regression models

We can substantially increase the flexibility of the predictor by considering variable transformations (cf. slide 60), for example:

- $\beta_0 + \beta_1 x_{i1}$
- $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$
- $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2$
- $\beta_0 + \beta_1 \sqrt{x_{i1}}$
- $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1} x_{i2}$
- $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \beta_3 x_{i1}^3 + \beta_4 x_{i1}^4$

In particular, using polynomials as in the last example above we can capture very flexible shapes of the regression function — see examples on the next slides.

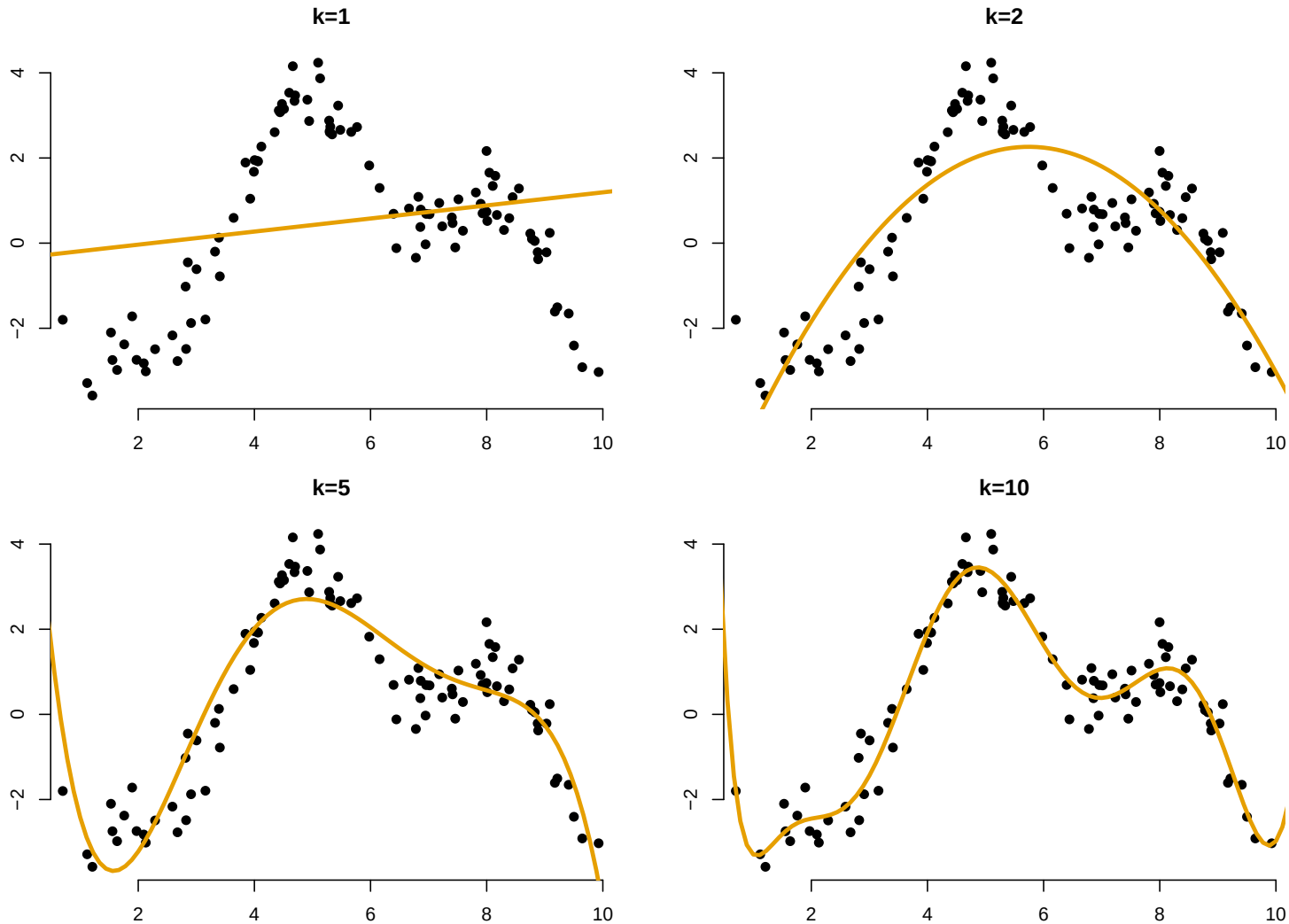


Figure: Simulated data (y_i, x_i), $i = 1, \ldots, 100$, and fitted linear regression models with polynomial predictor $\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_k x_i^k$.

Polynomials for the win?

Using polynomials, we can obtain effectively unlimited flexibility.

HOWEVER:

- higher-order polynomials are very unstable/heavily affected by outliers
- trying out all potential orders, for each covariate considered, is tedious

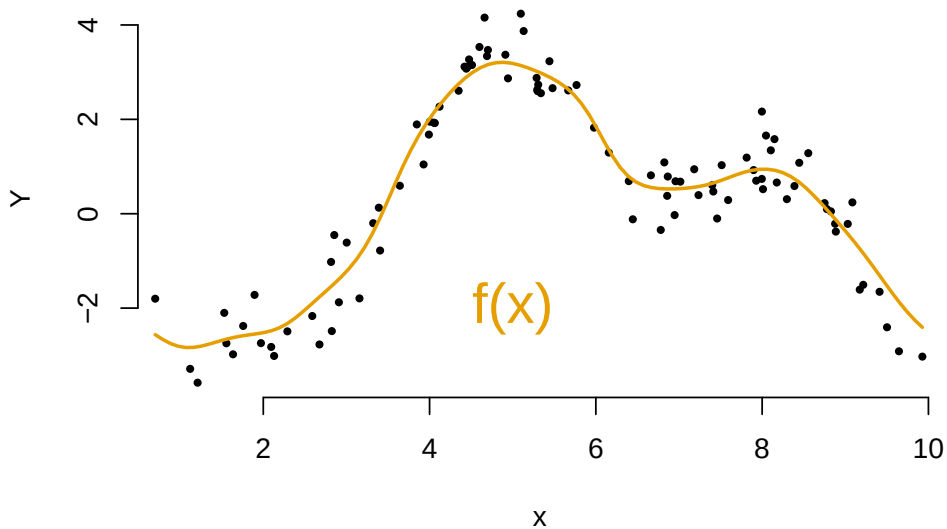
As a consequence, in practice it is uncommon to use polynomials of order > 2.

Nonparametric regression

In case of $p = 1$ (one covariate), we are interested in the model:

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

Instead of predetermining a specific form of f (linear, quadratic, etc.), we now turn to **nonparametric estimation** of f: essentially fitting a smooth curve to the data.



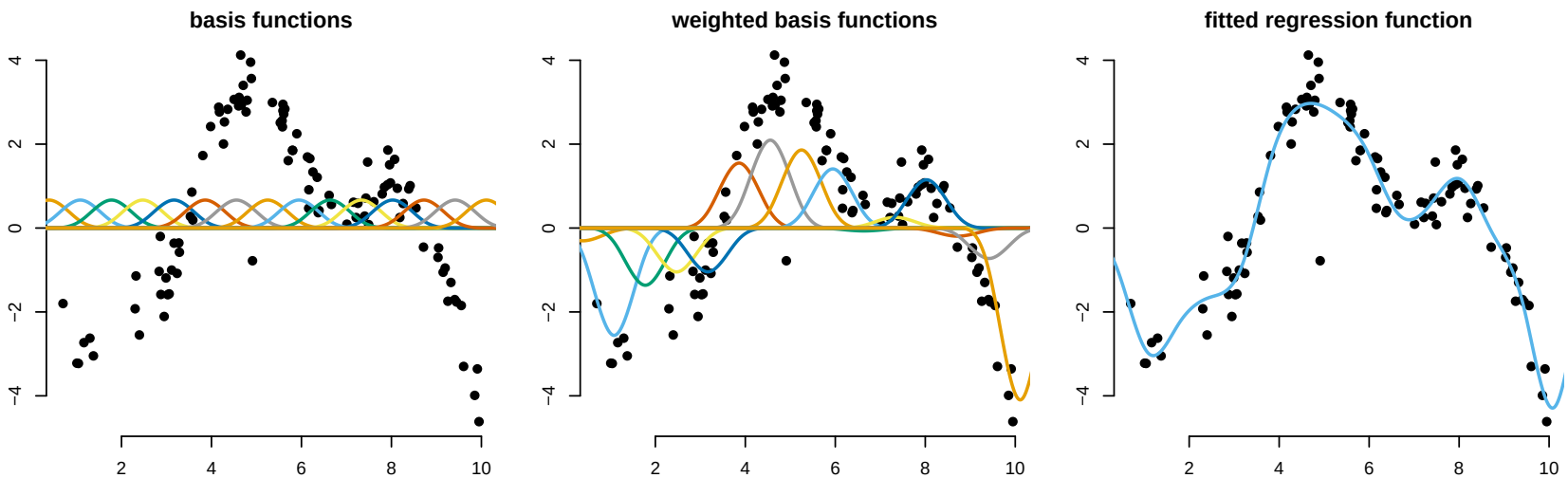
We will focus on one (of many possible) techniques: P–spline smoothing.

Estimation of f using spline basis functions — overview of the idea

Basic idea: construct function $f(x)$ as weighted sum of fixed basis functions,

$$f(x) = \gamma_1 B_1(x) + \gamma_2 B_2(x) + \ldots + \gamma_K B_K(x),$$

estimating the weights $\gamma_1, \ldots, \gamma_K$ to fit the model.



B–spline basis functions

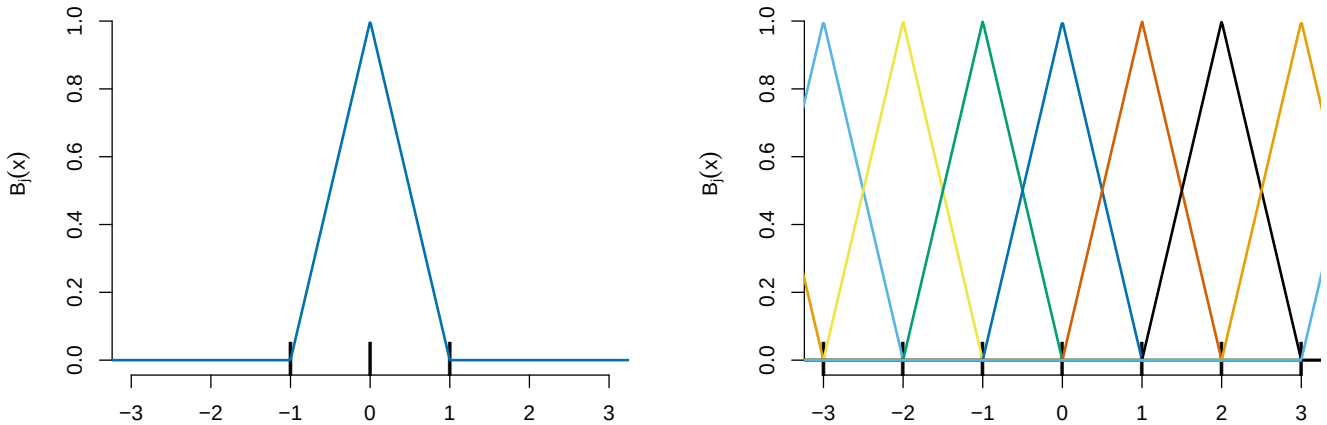
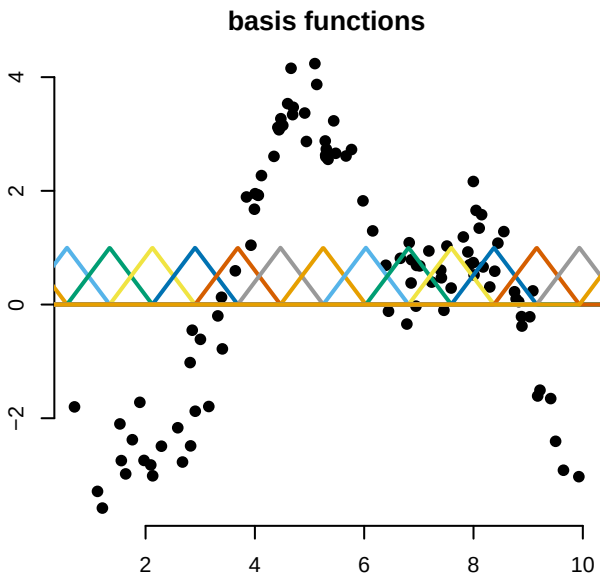


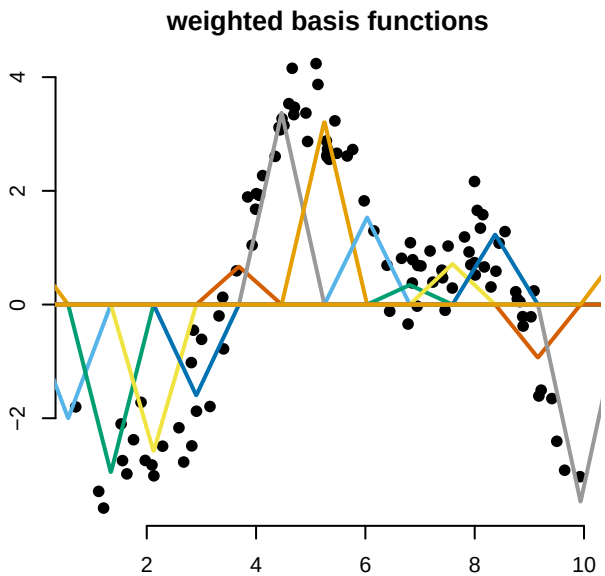
Figure: The left plot shows a single B–spline basis function of degree 1, the plot on the right shows a corresponding set of basis functions.

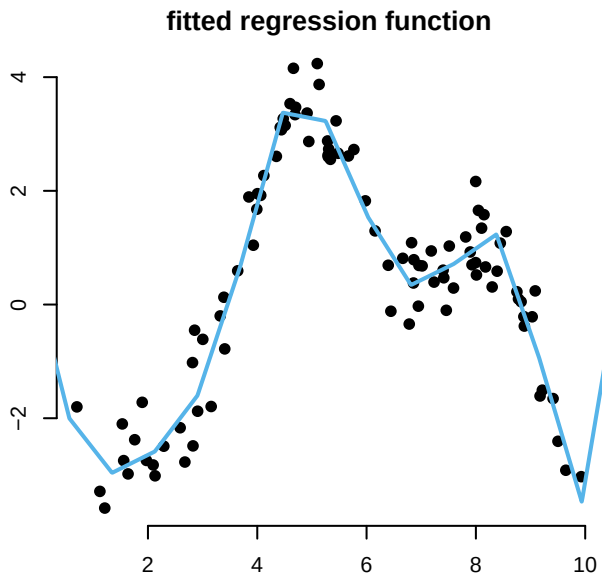
B–splines are particularly popular basis functions. The ones shown here — which are of degree 1 — are constructed as follows:

- three knots for each spline, two linear pieces in-between
- the linear pieces connect at the knots
- splines overlap such that $B_1(x) + B_2(x) + \ldots + B_K(x) = 1$ for all x

Curve fitting with B–splines of degree 1







- linear combination of B–splines produces reasonable fit
- however, the estimated regression function $\hat{f}(x)$ is not smooth

Higher degrees of the B–spline basis functions

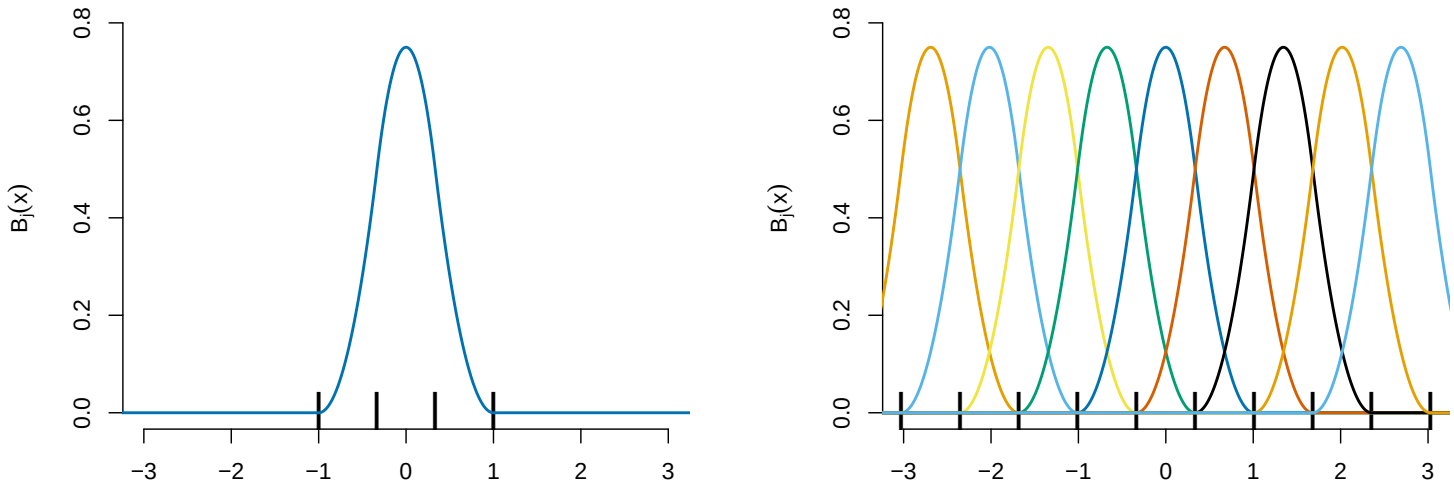


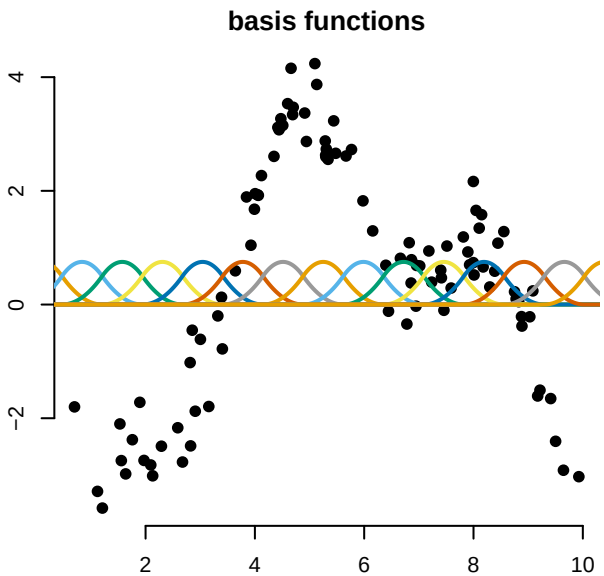
Figure: B–splines of degree 2.

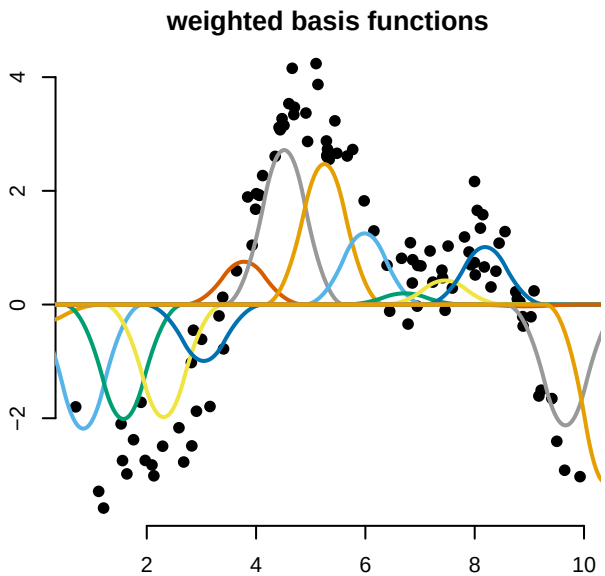
The construction is effectively analogous:

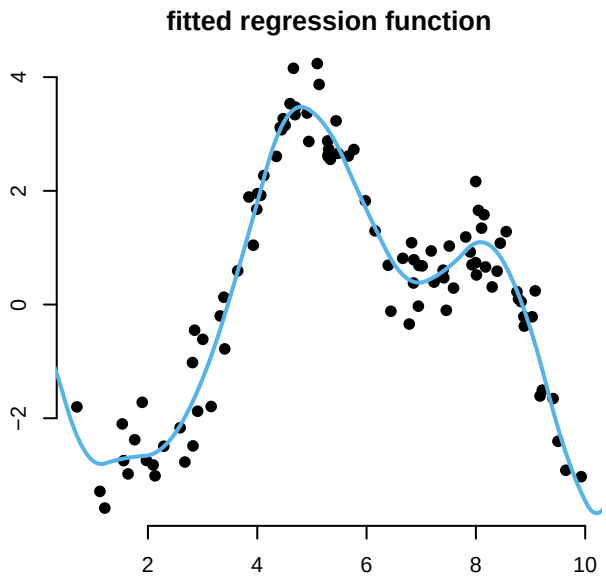
- now four knots for each spline, three quadratic polynomials in-between
- at the joining points, the derivatives match each other (i.e. spline is smooth)

This concept can be applied to construct B–splines of general degree d — in the following, we will however restrict the discussion to the case $d = 2$.

Curve fitting with B–splines of degree 2

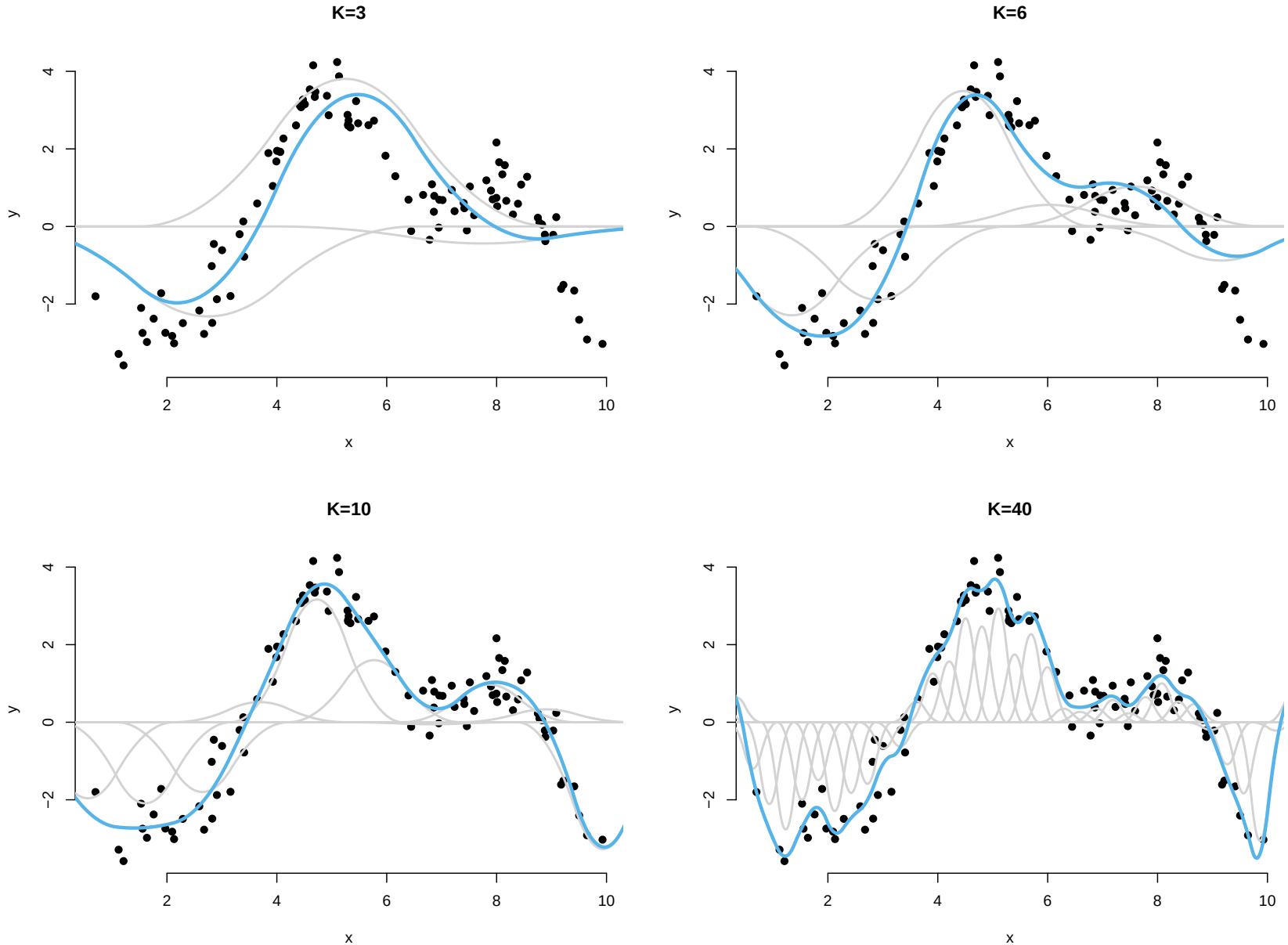






- the fit again looks good
- and this time $\hat{f}(x)$ is smooth (more precisely, it is once differentiable)

Choice of *K*



The previous slide illustrates:

- the choice of K reflects the classic bias-variance trade-off:
  - K too small $\rightsquigarrow$ underfitting (large bias)
  - K too large $\rightsquigarrow$ overfitting (large variance)
- we could select the optimal K say based on AIC, but that would effectively be just as tedious as polynomial regression[53]

[53] in fact, one also needs to choose the positioning of the knots

P–spline smoothing — the idea

Instead of choosing K, P–spline smoothing proceeds as follows:

- use a K large enough to allow for any interesting shapes of f (say $K = 30$)
- add penalty term to objective function to prevent overfitting

The objective function in **penalised least squares estimation** then has the form

$$\text{sum of squares} + \lambda \cdot \text{measure for overall curvature},$$

to be minimised with respect to the γ_i coefficients determining f.

The **smoothing parameter** λ is used to control how much emphasis we put on "smoothness" and allows us to find the right balance between under-/overfitting:

- for $\lambda = 0$, we are back in the unpenalised situation
- for $\lambda \to \infty$, we obtain a straight line fit

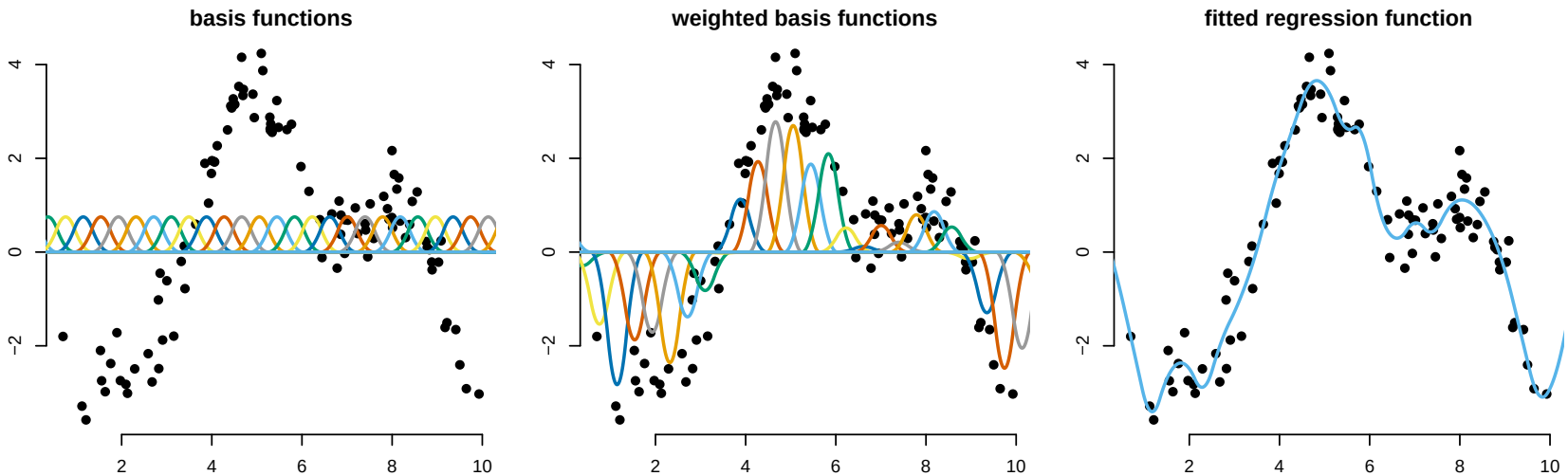


Figure: P–spline smoother obtained when using $\lambda = 0.1$ (and $K = 30$).

Very small $\lambda \rightsquigarrow$ **overfitting**.

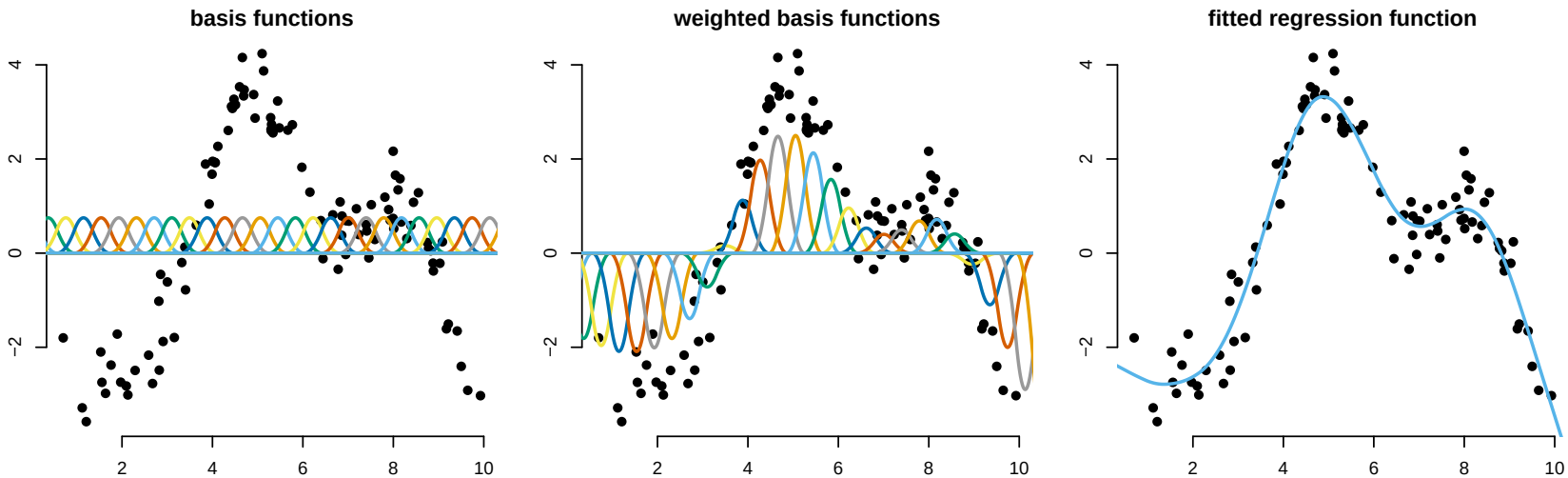


Figure: P–spline smoother obtained when using $\lambda = 10$ (and $K = 30$).

Moderate $\lambda \rightsquigarrow$ fit looks good!

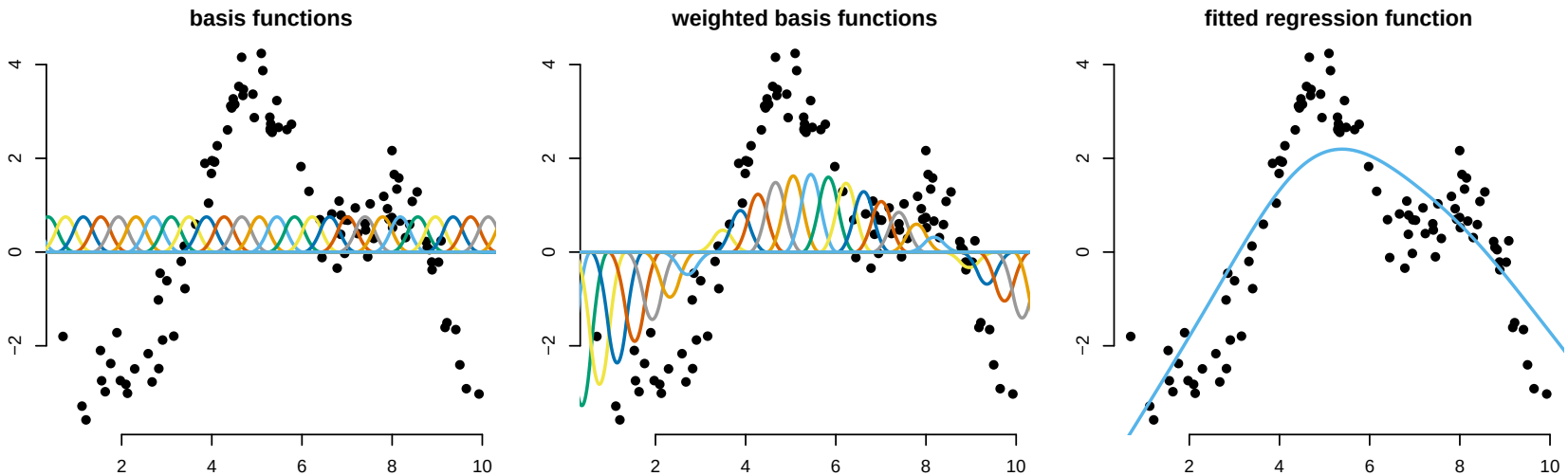


Figure: P–spline smoother obtained when using $\lambda = 200$ (and $K = 30$).

Large $\lambda \rightsquigarrow$ **underfitting**.

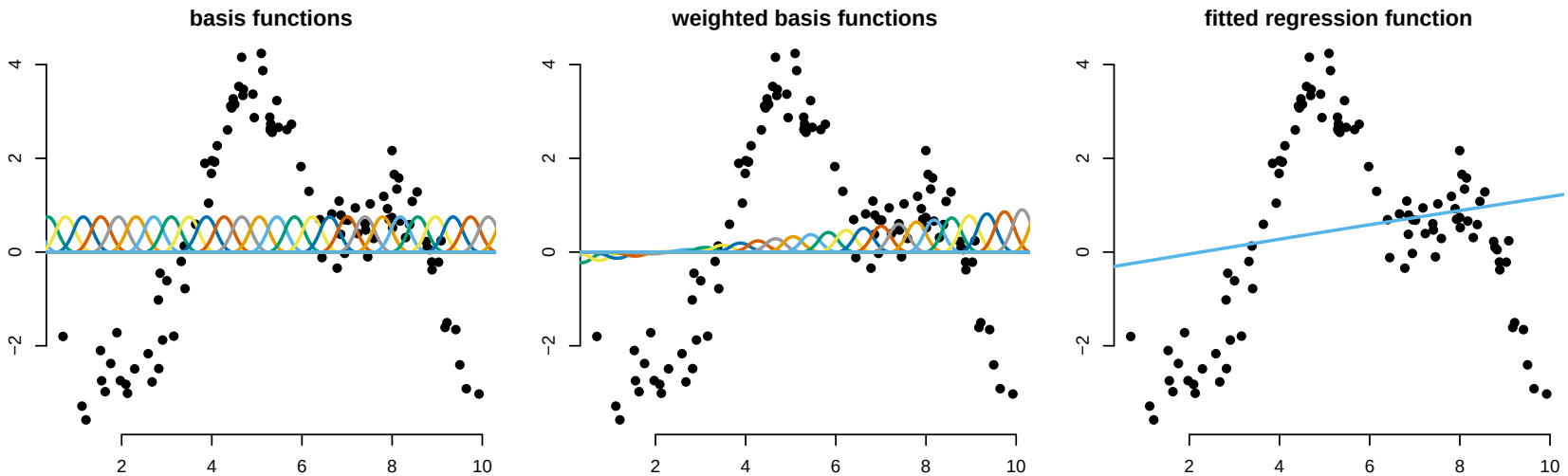


Figure: P–spline smoother obtained when using $\lambda = 100,000$ (and $K = 30$).

Very large $\lambda \rightsquigarrow$ we end up with the best possible straight line fit.

Let $x_{i1}, \ldots, x_{ip}$ be continuous covariates, and $z_{i1}, \ldots, z_{iq}$ additional covariates. A **generalised additive model** (GAM) is specified as follows:

1. the response variables Y_i are independent of each other and follow some distribution from the exponential family;
2. the mean of the response is linked to the linear predictor via an invertible and differentiable link function g,
$$g\big(\mathbb{E}(Y_i)\big) = \eta_i;$$
3. the linear predictor involves linear and smooth effects, the latter with an additive structure,
$$\eta_i = \beta_0 + \beta_1 z_{i1} + \ldots + \beta_q z_{iq} + f_1(x_{i1}) + \ldots + f_p(x_{ip})$$

$\rightsquigarrow$ so just like a GLM, but with some linear effects replaced by smooth effects

GAMs in R — the `mgcv` package

In R, GAMs can easily be fitted using the `gam` function from the `mgcv` package:

```
gam(response ~ z1 + ... + zq + s(x1) + ... + s(xp),
      family = ...(link=...))
```

- the functionality of the `family` option is exactly as for `glm`
- any covariate that enters the formula "as is" is modelled using a linear effect
- by adding the wrapper `s()`, we specify that a smooth effect is modelled
- by default, so-called thin plate splines are used as basis
- P–splines are used when adding the option `bs="ps"`

```
mod<-gam(acceleration~s(time),bs="ps")
```

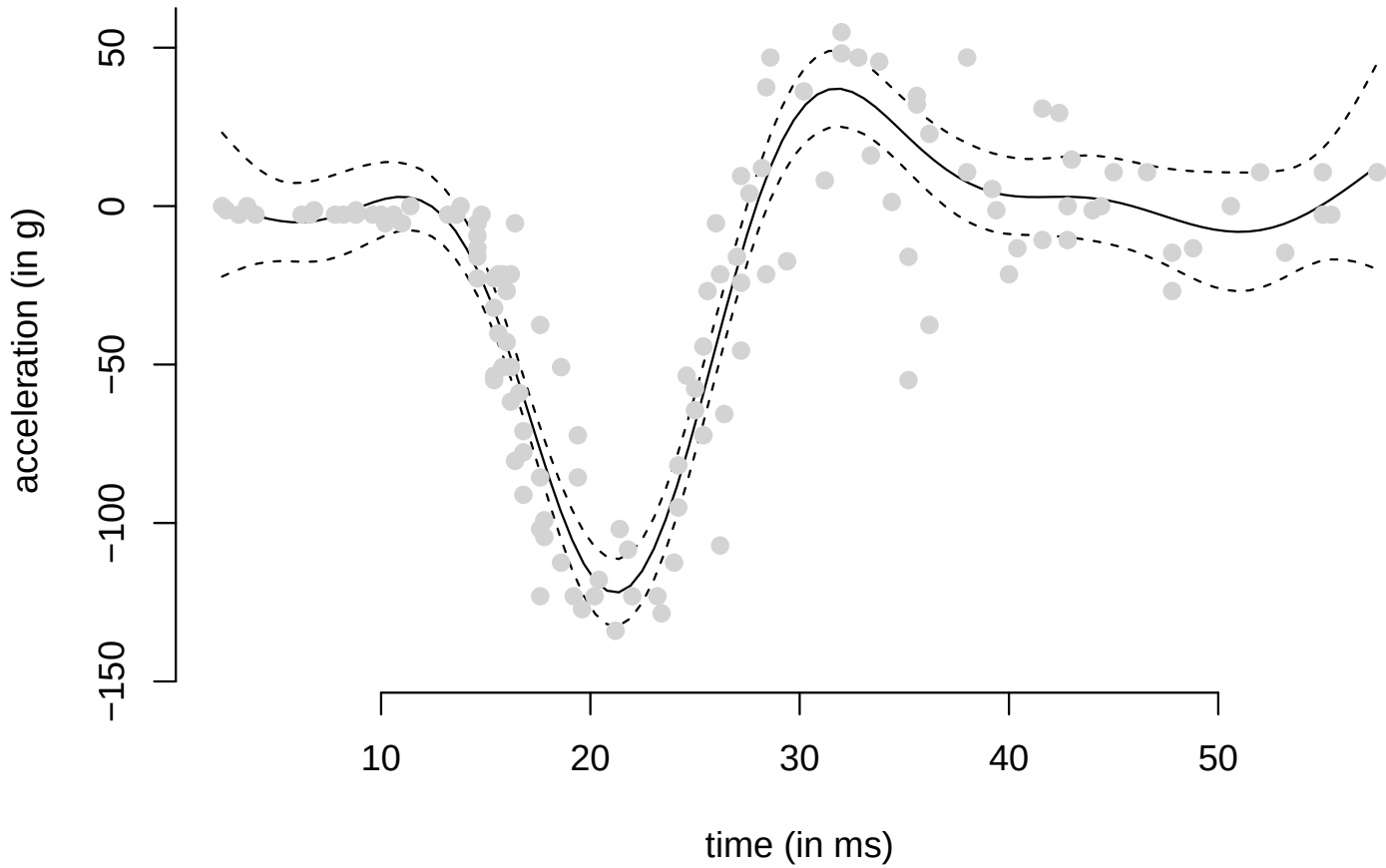


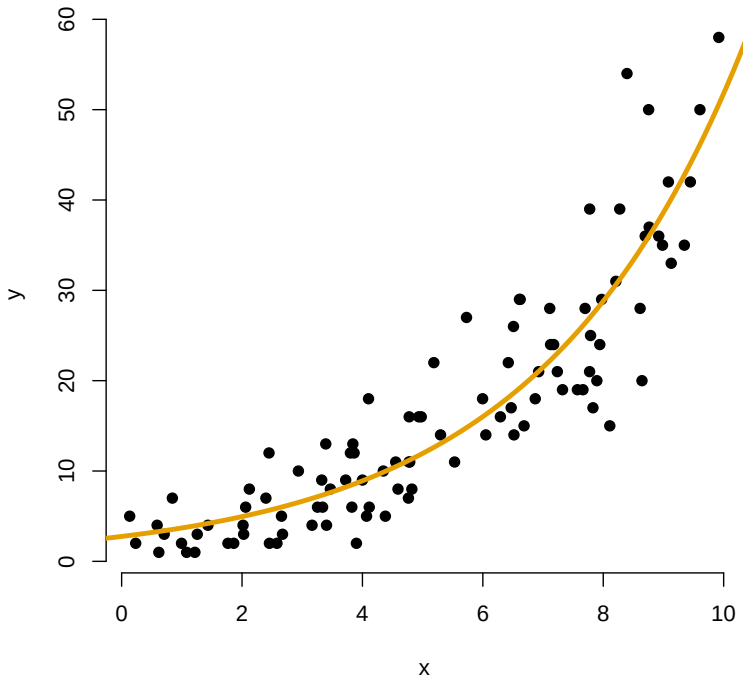
Figure: GAM fitted in the motorcycle example.

Chapter 8: Extensions & summary

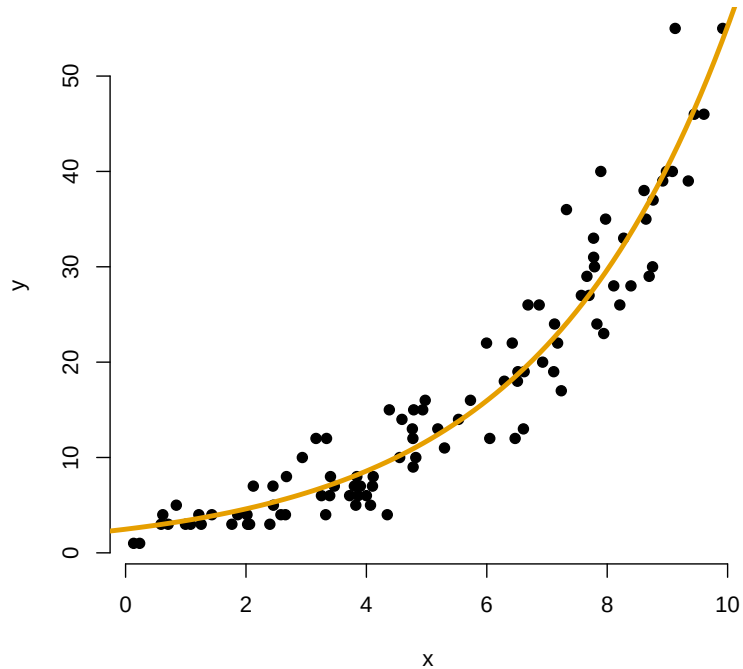
8.2 Models for overdispersed data

Dealing with overdispersion

Especially in Poisson regression, it often happens that a seemingly adequate model is rejected by the LRT because the mean-variance relation isn't right.[54]




```
> 1-pchisq(153.10,98)
[1] 0.0003141043
```




```
> 1-pchisq(79.107,98)
[1] 0.9191014
```

[54] recall that, in a Poisson regression model, we implicitly also explain the variance via the covariates

If the variances are larger than expected under the model, then we refer to this as **overdispersion**.

In such a case, proceeding with the invalid Poisson regression model leads to invalid estimation of the standard errors, and hence to invalid CIs and tests.[55]

Instead, models with additional **dispersion parameters** need to be considered:

```
glm(y~x,family=quasipoisson)
```

or

```
glm(y~x,family=quasibinomial)
```

An alternative would be to fit a negative binomial regression model.

[55] the reason simply being that the actual variance/uncertainty is larger than we think

```
> mod1<-glm(y~x,family=poisson)
> summary(mod1)

Call:
glm(formula = y ~ x, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9907  -1.1265  -0.2519   0.6725   3.4717

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.01383    0.08133   12.46   <2e-16 ***
x            0.29334    0.01109   26.45   <2e-16 ***

(Dispersion parameter for poisson family taken to be 1)
```

```
> mod2<-glm(y~x,family=quasipoisson)
> summary(mod2)

Call:
glm(formula = y ~ x, family = quasipoisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9907  -1.1265  -0.2519   0.6725   3.4717

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.01383    0.10778   9.406 2.35e-15 ***
x            0.29334    0.01469  19.963  < 2e-16 ***

(Dispersion parameter for quasipoisson family taken to be 1.756048)
```

Chapter 8: Extensions & summary

8.3 Dealing with zero-inflated data

Dealing with zero-inflated count data

Count data very often exhibit an excess of zeros (i.e. an "inflation of zeros"), relative to other counts predicted under a model.

This is often due to latent variables which determine whether we observe a zero count or some positive integer.

The R function `zeroinfl()` from the `pscl` package can be used to fit corresponding **zero-inflated Poisson regression models**, where

$$Y_i \begin{cases} = 0 & \text{with probability } \pi_i \\ \sim Po\left(e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_p x_{ip}}\right) & \text{with probability } 1-\pi_i \end{cases}$$

$$\text{logit}(\pi_i) = \alpha_0 + \alpha_1 x_{i1} + \ldots + \alpha_p x_{ip}$$

(a mixture of a Poisson GLM with a point mass on 0)

Chapter 8: Extensions & summary

8.4 Bayesian inference for GLMs

Bayesian estimation of GLMs

GLMs can also be fitted in a **Bayesian framework**.

Posterior distribution of regression coefficients:

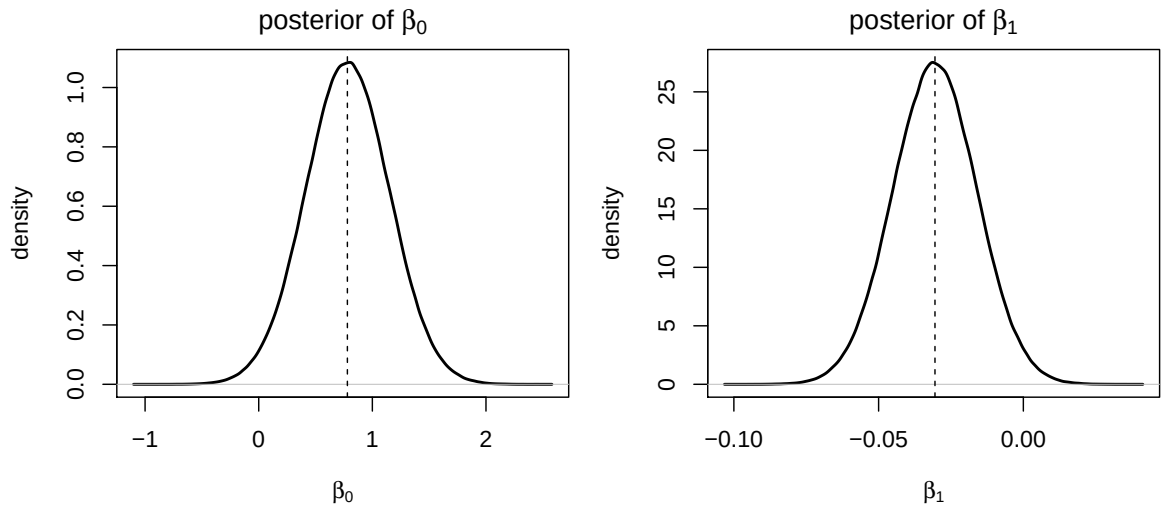
$$f(\beta_0, \ldots, \beta_p | y_1, \ldots, y_n) = \frac{f(y_1, \ldots, y_n | \beta_0, \ldots, \beta_p) f(\beta_0, \ldots, \beta_p)}{f(y_1, \ldots, y_n)}$$
$$\propto \underbrace{f(y_1, \ldots, y_n | \beta_0, \ldots, \beta_p)}_{\text{likelihood}} \underbrace{f(\beta_0, \ldots, \beta_p)}_{\text{prior}}$$

$\rightsquigarrow$ represents our belief about $\beta_0, \ldots, \beta_p$ after updating prior belief with data
$\rightsquigarrow$ need to choose a prior distribution for the $\beta_0, \ldots, \beta_p$
$\rightsquigarrow$ any expert knowledge can usefully be incorporated here
$\rightsquigarrow$ unlike ML estimate, the posterior already includes uncertainty quantification

Bayesian estimation of GLMs in R: `brms` package.

Bayesian estimation of logistic regression in the Donner party example

$$\text{logit}\big(\Pr(\text{survival}_i)\big) = \beta_0 + \beta_1 \cdot \text{age}_i$$



Maximum a posteriori estimates[56]:

$$\hat{\beta}_0 = 0.780, \quad \hat{\beta}_1 = -0.031$$

Maximum likelihood estimates:

$$\hat{\beta}_0 = 0.817, \quad \hat{\beta}_1 = -0.032$$

[56] `bayesglm(survival~age,family=binomial)` (note the default priors are slightly informative)

Chapter 8: Extensions & summary

8.5 Summary

Summary

- most regression problems can be tackled using standard linear regression (which is good in the sense that LMs are well understood)
- polynomial/smooth terms substantially increase the flexibility of linear models, allowing for the estimation of nonlinear functional relationships
- if the data don't directly lend themselves to standard linear modelling, then sometimes transformations will lead to approximately linear systems[57]
- however, sometimes we still need more flexibility, regarding either
  - **distributional assumptions for the response**
  - or the **functional relationship between covariates and response**
- in such cases, GLMs may do the trick

[57] the standard example: modelling log-transformed strictly positive continuous data

What's really neat about GLMs is that they constitute a **unifying framework** for:

- various model formulations, including
  - Poisson regression
  - Bernoulli and binomial regression (i.e. logistic regression)
  - gamma regression
- the use of link functions to increase flexibility
- the associated parameter estimation method (IRLS)
- model selection and model checking within these classes of models

GLMs also provide the starting point for many other classes of regression models, such as GLMMs or GAMs.

Some recommendations for the (oral) exam

Make sure that...

- ...you understand each aspect of the GLM definition
- ...you know the exact model specifications of the special cases of GLMs
- ...you understand (and can explain) why we need weighted least squares
- ...you can explain the idea of the method of scoring (no technical details)
- ...you can tell me how uncertainty quantification works for GLMs
- ...you can explain the main ideas regarding model selection & checking
- ...you can explain the idea and the model formulation of mixed models

Perhaps most importantly, **practice to actually talk about these things**!!


THANKS FOR YOUR ATTENTION
imgflip.com