Statistical models

June 11, 2024

Contents

Chapter 1

Introduction

1.1 Statistical models: general definitions

Regression analysis consists in the investigation of the relationship that can be expressed as an equation connecting a *response/dependent variable* to one or more explanatory/predictor variables in the following steps:

1. behind any statistical model there are assumptions which are more or less reasonable for our data at hand; in the **specification step** we define the feature/assumptions we are

2. then there will be the **estimation step**: models are characterized by unknown quantities that have to be estimated; depending on the amount of assumptions we are willing to make we can use different estimating procedures (we will focus on ML methods, btw)

3. then some task tipically involved are **hypothesis testing on regression coefficient** and **model comparison** (to choose among candidate models)

1.1.0.1 Random samples

We are interested in a phenomenon `Y` (eg cholesterol level) but for practical reasons we cannot know the distribution of whole the population `P`; so we rely on a *observed sample* on n units $\mathbf{y}$ which is realization of the random mechanism called *random sample* $\mathbf{Y}$ (a collection of random variables). The observed sample is an element of the set of all possible samples $\boldsymbol{\mathcal{Y}}$ we can draw, called *sample space.*
Below some notation

$\mathtt{Y}$ statistical phenomenon of interest in a given population $\mathtt{P}$

$\mathbf{y} = (y_1,\ \ldots,\ y_n)^\top$ Observed sample
(numerical) values observed on n statistical units randomly drawn from the population $\mathtt{P}$

$\Downarrow$

$\boldsymbol{Y} = (Y_1,\ Y_2,\ \ldots,\ Y_n)^\top$ Random sample: *set of r.vs. that describe the possible value of* $\mathtt{Y}$ *in each random draw*
$\Rightarrow$ $\mathbf{y} \in \boldsymbol{\mathcal{Y}} \subseteq \mathbb{R}^n$ *Sample space*
$\Rightarrow$ $f_0(\mathbf{y})$ *Unknown "true" probability mass/density function of* $\boldsymbol{Y}$

1.1.0.2 Parametric statistical models

We want to have information about f_0 using our sample, we have two strategies:

- to introduce a parametric statistical model: we assume that f_0 is element of a broader set $\mathcal{F}$ of probability distribution having the same functional form and which differs by a set of k parameters $\boldsymbol{\theta}$ (which can be a scalar as well); the distribution of our interest is f with $\boldsymbol{\theta}_0$ unknown. So the problem is rephrased from extract information on f_0 to information of $\boldsymbol{\theta}_0$
- adopt a non parametric approach (not our focus here)

So regarding the parametric statistical model

$f_0 \in \mathcal{F} = \left\{ f(\cdot;\boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^k \right\}$ parametric statistical model for $\boldsymbol{Y}_n$
parametric family containing the "true" probability mass/density function of $\boldsymbol{Y}$
$\Rightarrow$ $\boldsymbol{\Theta}$ *parameter space*
$\Rightarrow$ $f_0(\mathbf{y}) = f(\mathbf{y};\boldsymbol{\theta}_0)$ with $\boldsymbol{\theta}_0$ *unknown*

1.1.0.3 Parametric statistical model specification

Model specification is the process of choosing (*"specifying"*) a parametric statistical model $\mathcal{F}$ suitable for $\boldsymbol{Y}$.
Specifying it means introducing a set of assumptions that describe the statistical model; this is a crucial step since inferential procedures rely on the model to be correctly specified. There are tools to check wheter the model assumptions are adequate or not.
Model specification can be based on information about:

- the features of the statistical phenomenon $\mathtt{Y}$ of interest and of the population $\mathtt{P}$ (eg qualitative/quantitative, discrete/continuous, bounded or not). This course will be focused on this task.
- the sampling scheme: this will define the *dependence structure among the rvs in the random sample* $\boldsymbol{Y}$, eg sampling schemes with dependence vs independence among observations.
  We will mainly focus on independent observations.

Once specified the model we can make inference on parameters; errors in model specification will give error in inference.

1.1.0.4 Likelihood function of θ

Supposing we

- have chosen a parametric statistical model/family of distribution *for the random sample* $\boldsymbol{Y}$ $\mathcal{F} = \left\{f(\cdot;\boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^k\right\}$
- have our observed sample - realisation of the random sample $\boldsymbol{Y}$, $\mathbf{y} = (y_1,\ y_2,\ \dots,\ y_n)^\top$

The likelihood function is a way to combine these informations; $L(\boldsymbol{\theta})$ is *Likelihood function of* θ that is a function which treat observed data as fixed $L(\cdot) = L(\cdot;\mathbf{y})$ and of the type $L : \boldsymbol{\Theta} \to \mathbb{R}^+ \cup 0$, so going from the parameter space to the positive reals. Likelihood do depends on sample values so different sample will be characterized by different likelihoods.
Actually we have that for each possible $\boldsymbol{\theta}$ the likelihood function is $c(\mathbf{y})f(\mathbf{y};\boldsymbol{\theta})$, where $c(\mathbf{y})$ represents a multiplicative factor that does not depend on $\boldsymbol{\theta}$; so the likelihood function is proportional to the density/mass function evaluated on the observed sample.
The likelihood function:

- summarize all the information we have about f_0 (the “true" probability distribution of $\boldsymbol{Y}$):
  - on one hand the $f_0 \in \mathcal{F}$ parametric statistical model; the pre-experimental (a priori - before observing the actually drawn sample) information - theoretical assumptions
  - on the other hand the data/empirical evidence $\mathbf{y} = (y_1,\ y_2,\ \dots,\ y_n)^\top$ in the observed sample
- literally shows how the probability/density of observing the actually drawn sample changes, as the value of the unknown parameter $\boldsymbol{\theta}$ changes
- from the practical pov, it can be interpreted asa way to measure the plausibility of each possible value of $\boldsymbol{\theta}$

Example 1.1.1. Assuming each Y_i has a gaussian distribution with common mean and variance and observation are independent, that is

$$Y_i \sim N\big(\mu, \sigma^2\big), \quad \text{IID} \quad i = 1, \dots n, \quad \mu \in \mathbb{R}, \quad \sigma^2 \in \mathbb{R}^+$$

we have

- given that each random variable can take any value on the real line, the sample space is $\mathbb{R}^n$ ($\mathbf{y} \in \boldsymbol{\mathcal{Y}} = \mathbb{R}^n$)
- the parameter space is $R \times \mathbb{R}^+$, the first for mean the second for variance ($\boldsymbol{\theta} = \big(\mu, \sigma^2\big)^\top \in \boldsymbol{\Theta} = \mathbb{R} \times \mathbb{R}^+$)
- the likelihood is the product (being observation independent) of gaussian density functions with data y_i replaced instead of x

$$\begin{aligned} f(\mathbf{y};\boldsymbol{\theta}) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y_i-\mu)^2}{2\sigma^2}\right] \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left[-\frac{\sum_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}\right] \end{aligned}$$

Example 1.1.2. Using a more compact notation we can re-express/summarize the setup/distribution of the random vector $\mathbf{Y} = \{Y_1, \dots, Y_n\}$ by introducing the multivariate normal, with

$$\boldsymbol{Y} \sim MVN_n\left(\underbrace{\boldsymbol{\mu} = \begin{bmatrix}\mu \\ \vdots \\ \mu\end{bmatrix}}_{n\times 1}, \underbrace{\boldsymbol{\Sigma} = \sigma^2\mathbf{I}_n}_{n\times n}\right)$$

Remark 1. Multivariate gaussian are fundamental in inference so a review follows

1.2 Multivariate Gaussian distributions review

1.2.0.1 Joint probability density function

Multivariate gaussian distribution is used when we have a random vector Y which can take values in $\mathbb{R}^n$ and being a vector we will have a vector of means (it contains all the expected values of the elements in the random sample, μ_1 will be the expected value of Y_1) and a variance covariance matrix which will contains variances of the random variables contained in the random vector as long as the relationship/covariances between each pair of them. $\boldsymbol{\mu}$ can be any real valued vector, $\boldsymbol{\Sigma}$ must be square symmetric and positive definite (variances must be striclty positive and covariances are bounded between the product of the square root of the variances

$\boldsymbol{Y} = (Y_1, \dots, Y_n)^\top$	n-dimensional random variable
$\mathbf{y} = (y_1, y_2, \dots, y_n)^\top \in \mathbb{R}^n$	set of possible values of $\boldsymbol{Y}$ (joint realisations of the n random variables)
$\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)^\top$	n-dimensional real-valued vector
$\boldsymbol{\Sigma}$	$n \times n$ real-valued, symmetric matrix (positive definite - invertible)

From a functional pov the joint density expression is as follows. We have an expression involving trhe inverse of varcov matrix, its determinant, the difference between vectors y and mu and lots of quantity that relies on matrix algebra

$$\boldsymbol{Y} \sim MVN_n(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \iff f(y_1, \dots, y_n; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left[-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right]}{(2\pi)^{\frac{n}{2}} \det(\boldsymbol{\Sigma})^{\frac{1}{2}}}$$

The point is that this function assign each vector Y a non negative real value (its density).
With random vector we can apply expected value and variance operators obtaining respectively the vector containing the expected value of each element in the vector, while when applying the vcariance operator one gets the matrix

containing variances (main diagonal) and covariances (off diagonal)

$$\begin{aligned}\mathrm{E}\left[\boldsymbol{Y}\right] &= \boldsymbol{\mu}\\ \mathrm{Var}\left[\boldsymbol{Y}\right] &= \mathrm{E}\left[\boldsymbol{Y}\boldsymbol{Y}^\top\right] - \mathrm{E}\left[\boldsymbol{Y}\right]\mathrm{E}\left[\boldsymbol{Y}\right]^\top = \boldsymbol{\Sigma}\end{aligned}$$

1.2.0.2 Standardised multivariate Gaussian distribution

As long as univariate case we have the special case of standard gaussian random vector which is obtained choosing null vector as means and identity matrix as variance-covariance

$\boldsymbol{\mu} = \mathbf{0}_n = (0, 0, \dots, 0)^\top$ n-dimensional null vector
$\boldsymbol{\Sigma} = \mathbf{I}_n$ $n \times n$ identity matrix

In this case the functional form of the density become simpler because it's the only case in which uncorrelation implies independence and so we can write the joint density as product of marginal ones

$$\begin{aligned}f(y_1, \dots, y_n; \mathbf{0}_n, \mathbf{I}_n) &= \frac{\exp\left[-\frac{1}{2}\sum_{i=1}^n y_i^2\right]}{(2\pi)^{\frac{n}{2}}}\\ &= \prod_{i=1}^n \frac{\exp\left[-\frac{y_i^2}{2}\right]}{\sqrt{2\pi}}\end{aligned}$$

Example 1.2.1 (Examples with $n = 2$)**.** In two dimension we can plot the density; if

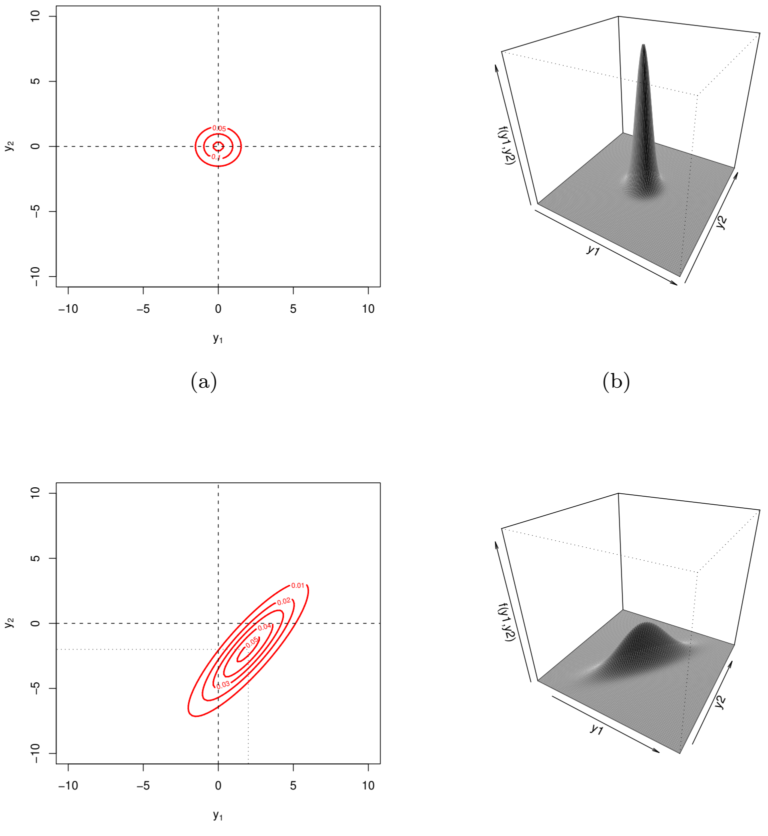
- $\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ distribution plotted in figure 1.1 a and b. with identity varcov matrix shape is a circle

- $\boldsymbol{\mu} = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$ $\boldsymbol{\Sigma} = \begin{bmatrix} 4.8 & 5.4 \\ 5.4 & 7.95 \end{bmatrix}$ distribution plotted in figure 1.1 c and d. By introducing correlation we move toward elliptical distribution. The orientation of the ellipses depend on the correlation, here positive; the stretch of the ellypse depends on the difference between variances

1.2.0.3 Some properties

- each marginal distribution of order $q < n$ (eg any subvector) is a q-dimensional multivariate Gaussian distribution: so if a vector is multivariate gaussian, even the single Y_i composing are as well;

- each conditional distribution of a subvector of Y, $Y_{1a}, Y_{2a}, \dots, Y_{ha}$, given another portion of the subvector $Y_{1b}, Y_{2b}, \dots, Y_{lb}$, then this is an h-dimesional multivariate Gaussian distribution as well. So we can say that mvn we can say is closed both to marginalization (first point) and to conditioning: whenever we extract a marginal or a conditional distribution from a gaussian rv, we obtaina gaussian as well



(a)

(b)

(c)

(d)

Figure 1.1: example1.

- $Y_1, \dots, Y_n$ are independent *if and only if* $\mathbf{\Sigma}$ is diagonal (if and only if they are uncorrelated) and in the case the joint distribution is the product of the n univariate gaussian distribution

$$f(\mathbf{y}; \boldsymbol{\mu}, \mathbf{\Sigma}) = \frac{\exp\left[-\frac{1}{2}\sum_{i=1}^{n} \frac{(y_i-\mu_i)^2}{\sigma_i^2}\right]}{(2\pi)^{\frac{n}{2}} \left[\prod_{i=1}^{n} \sigma_i^2\right]^{\frac{1}{2}}}$$
$$= \prod_{i=1}^{n} \left\{ \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{1}{2\sigma_i^2}(y_i-\mu_i)^2\right] \right\}$$

- linear combinations ov MVN random variables (important property): let
  - $\boldsymbol{Y}$ be a n-dimensional Gaussian vector characterized by parameters $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$
  - $\mathbf{A}$ a $q \times n$ real-valued matrix (fixed not random)
  - $\mathbf{b}$ an n-dimensional real-valued vector,

  then the linear combination using $\mathbf{A}$ and $\mathbf{b}$, $\mathbf{Z} = \mathbf{A}(\boldsymbol{Y} + \mathbf{b})$ (so we add a constant to the vector Y and premultiply by A), is a q-dimensional Gaussian vector with parameters trasformed in the following way: mean $\mathbf{A}(\boldsymbol{\mu} + \mathbf{b})$ and varcov $\mathbf{A\Sigma A}^\top$.
  This reminds the univariate case where if $Z = a(Y + b)$ then $\mathbb{E}[Z] = a(\mathbb{E}[Y] + b)$ and $\mathrm{Var}[Z] = a^2 \mathrm{Var}[Y]$

Important remark 1. Nella notazione del prof quando una lettera maiuscola è ingrassetto, se ha anche italic è un vettore ($\boldsymbol{Y}$), altrimenti solo grassetto per matrici $\mathbf{A}$

Example 1.2.2 (Standardization as linear combination)**.** We can use the last property to standardize any gaussian random vector. If $\mathbf{\Sigma}$ is positive definite, there's a way to obtain $\mathbf{\Sigma}^{\frac{1}{2}}$, inverse of which can be thought as square root since $\mathbf{\Sigma} = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{\Sigma}^{\frac{1}{2}}$. This matrix will be invertible with inverse $\mathbf{\Sigma}^{\frac{1}{2}}$. if $\mathbf{\Sigma}$ is invertible. So in this case, by setting the $\mathbf{A}$ and $\mathbf{b}$ as follows:

NB: we dont' delve in how to obtain the square root here

- $\mathbf{A} = \mathbf{\Sigma}^{-\frac{1}{2}}$ such that $\mathbf{\Sigma} = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{\Sigma}^{\frac{1}{2}}$ and $\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{\Sigma}^{-\frac{1}{2}} = \mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{\Sigma}^{\frac{1}{2}} = \mathbf{I}_n$
- $\mathbf{b} = -\boldsymbol{\mu}$

then $\boldsymbol{Z} = \mathbf{\Sigma}^{-\frac{1}{2}}(\boldsymbol{Y} - \boldsymbol{\mu})$ will be an n-dimensional *standardised* Gaussian vector. Again this is similar to what occurs in the univariate where to standardize a variable we have to subtract the mean and divide by the standard deviation

Chapter 2

Gaussian linear model

2.1 An introductory example

2.1.1 Simple linear regression

2.1.1.1 Setup

It is known a glucose level in blood between 100 and 125 mg/dL represents a risk factor for developing diabetes:

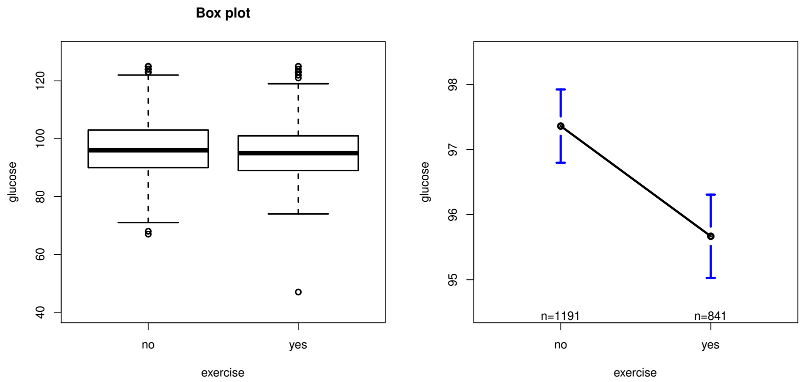
- the **Aim:** Does physical activity (a modifiable factor related to life style) contribute to the reduction of the glucose level, thus preventing a severe disease?
- **available information:** data from an observational study for glucose level and physical activity (yes-no) on a sample of 2032 women not affected by diabetes after menopause

The aim is to look at if there are difference in glucose level between active and non active women; we look at conditional distributions (via boxplot and group means) (fig 2.1):

- he two boxplot are quite similar in terms of variability; a little shift in location of the boxplot (yes slightly lower median) but there's a lot of overlapping.
- by zooming to the means with confidence interval there seems to be a difference here in mean glucose level (rather small 1.5-1.7) and there's no overlap between the 2 ci; probablu the diff between two sample means is statistically significant

2.1.1.2 Models specification

We then state a more formal description of the relation between glucose level and physical activity using a parametric statistical model with parameters related to difference between groups (our main interest); once formalized we will be able to perform ML estimation and hypothesis test using tools exploiting maximum likelihood



(a) (b)

Figure 2.1: glucose and exercise

- the starting point is the joint distribution, which can be splitted in the product of marginal (of exercise) times the conditional of glucose level given exercise which is our main interest

  $$f(\texttt{glucose}_i, \texttt{exercise}_i) = f(\texttt{glucose}_i|\texttt{exercise}_i)f(\texttt{exercise}_i) \quad i = 1, \ldots, 2032$$

  however rather than focusing on joint distribution we focus our attention on conditional distribution

- regarding the conditional distribution we make the following assumptions (which are somewhat reasonable looking at the graphs before):

  A) conditional expected values, that is the expected values of the conditional distribution are summarized by

  $$\mathrm{E}\left[\texttt{glucose}_i|\texttt{exercise}_i\right] = \beta_0 + \beta_1 \mathbf{1}\left\{\texttt{exercise}_i = \texttt{yes}\right\}, \forall i$$

  where

  $$\mathbf{1}\left\{\texttt{exercise}_i = \texttt{yes}\right\} = \begin{cases} 1 & \text{if } \texttt{exercise}_i = \texttt{yes} \\ 0 & \text{otherwise} \end{cases}$$

  B) the conditional variances are supposed to be constant, the two conditional distribution have the same variability (it's independent from the regressors)

  $$\mathrm{Var}\left[\texttt{glucose}_i|\texttt{exercise}_i\right] = \sigma^2, \forall i$$

  C) regarding dependence between observation a reasonable assumption is to assume that the conditional distribvution of two units in the sample are uncorrelated

  $$\mathrm{Corr}\left(\texttt{glucose}_i|\texttt{exercise}_i, \texttt{glucose}_j|\texttt{exercise}_j\right) = 0, \forall i \neq j$$

D) finally having seen quite symmetry of conditional distributions, we assume that conditional distribution are gaussian (with mean and variance as stated before)

$$\texttt{glucose}_i|\texttt{exercise}_i \sim \mathrm{N}\left(\beta_0 + \beta_1 \mathbf{1}\left\{\texttt{exercise}_i = \texttt{yes}\right\}, \sigma^2\right), \forall i$$

2.1.1.3 Estimation

```
> summary(modello1)
Call:
lm(formula = glucose ~ exercise, data = hers.nod)

Residuals:
     Min       1Q   Median      3Q      Max
 -48.668   -6.668   -0.668   5.639   29.332
 Coefficients:
                 Estimate  Std.  Error  t value  Pr(>|t|)
   (Intercept)    97.3610        0.2815   345.85    0.0000
   exerciseyes    -1.6928        0.4376    -3.87    0.0001

Residual standard error:  9.715 on 2030 degrees of freedom
Multiple R-squared:  0.007318, Adjusted R-squared:  0.006829
F-statistic:  14.97 on 1 and 2030 DF, p-value:  0.000113
```

Here the test of interest regards exerciseyes; there is a significant difference (which does not need to be clinically relevant) while `Residual standard error` is an estimate of $\sqrt{\sigma^2}$: there is a lot of variability even within the same conditional distribution and this reflect the fact that R square is low.

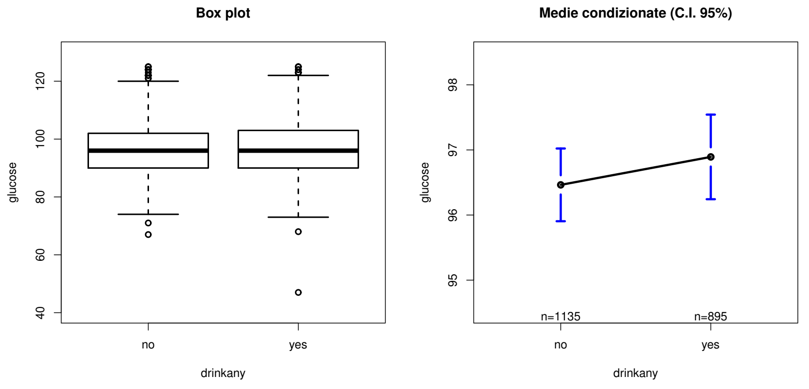
2.1.2 Multiple linear regression

2.1.2.1 Introducing other regressors

Women that are physically active may completely differ from women that are not, due to a number of other characteristics (socio-economical status, life style, health conditions Some of these characteristics could be associated with both the glucose level and physical activity (eg women that are physically active could be younger, haltier and have different habits related to alcohol consumption). Since data were collected through an observational study, these characteristics could act as confounders, thus preventing a correct evaluation of the effect of physical activity on glucose level.
Some plotting regarding drinking age and bmi is done in figures 2.2 (not great differences graphically), 2.3 (slight tendency of decreasing trend), 2.4 (increasing).

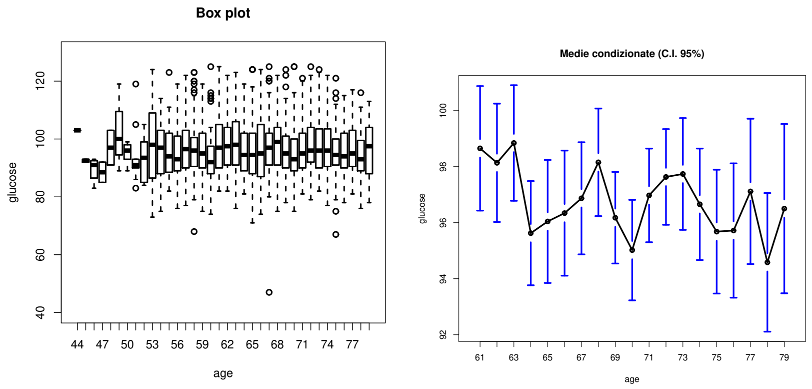
2.1.2.2 Model definition/specification

As done before to put together all



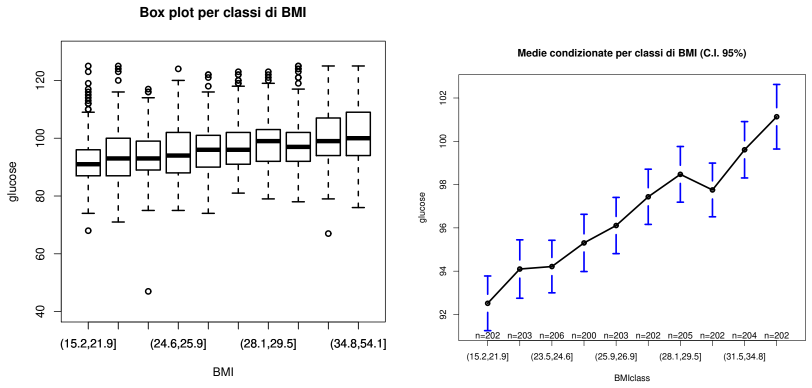
(a) (b)

Figure 2.2: glucose and drink



(a) (b)

Figure 2.3: glucose and age



(a) (b)

Figure 2.4: glucose and bmi

- the joint density is as follow

  $$\begin{aligned}&f(\texttt{glucose}_i, \texttt{exercise}_i, \texttt{drinkany}_i, \texttt{age}_i, \texttt{BMI}_i)\\&= f(\texttt{glucose}_i|\texttt{exercise}_i, \texttt{drinkany}_i, \texttt{age}_i, \texttt{BMI}_i) \cdot f(\texttt{exercise}_i, \texttt{drinkany}_i, \texttt{age}_i, \texttt{BMI}_i)\\&= f(y_i|x_{1i}, x_{2i}, x_{3i}, x_{4i})f(x_{1i}, x_{2i}, x_{3i}, x_{4i}), i = 1, \ldots, 2032\end{aligned}$$

  but our main focus given are the conditional distribution

- to focus on the conditional distribution assumptions

  A) the main extension relative to the univariate case is on the first assumption

  $$\begin{aligned}&\mathrm{E}\left[Y_i|x_{1i}, x_{2i}, x_{3i}, x_{4i}\right]\\&= \mu_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i}\\&= \beta_0 + \beta_1 \mathbf{1}\left\{\texttt{exercise}_i = \texttt{yes}\right\} + \beta_2 \mathbf{1}\left\{\texttt{drinkany}_i = \texttt{yes}\right\} + \beta_3 \texttt{age}_i + \beta_4 \texttt{BMI}_i, \forall i\end{aligned}$$

  B) we assume constant conditional variance

  $$\mathrm{Var}\left[Y_i|x_{1i}, x_{2i}, x_{3i}, x_{4i}\right] = \sigma^2, \forall i$$

  C) we keep the uncorrelation

  $$\mathrm{Corr}\left(Y_i|x_{1i}, x_{2i}, x_{3i}, x_{4i}, Y_h|x_{1h}, x_{2h}, x_{3h}, x_{4h}\right) = 0, \forall i \neq h$$

  D) again the normality assumption

  $$\texttt{glucose}_i|\texttt{exercise}_i, \texttt{drinkany}_i, \texttt{age}_i, \texttt{BMI}_i \sim N\big(\mu_i, \sigma^2\big), \forall i$$

2.1.2.3 Estimation

```
> summary(modello2)
Call:
lm(formula = glucose ~ exercise + drinkany + age + BMI, data = hers.nod)
...
 Coefficients:
                Estimate  Std. Error  t value  Pr(>|t|)
  (Intercept)    78.9624      2.5928    30.45    0.0000
  exerciseyes    -0.9504      0.4287    -2.22    0.0267
  drinkanyyes     0.6803      0.4220     1.61    0.1071
          age     0.0635      0.0314     2.02    0.0431
          BMI     0.4892      0.0416    11.77    0.0000
```

Effect of physical activity changes (it was -1.69) so part of the effect was due to covariates, but is still significant. BMI is another important factor

2.2 General definition

2.2.0.1 Basic assumptions

In general considering

- Y_i: random variable that describes the value for the dependent variable observed on the i-th sample unit ($i = 1, \dots, n$);
- $x_{1i}, x_{2i}, \dots, x_{pi}$ values of the regressors for the i-th sample unit (*covariate pattern*), where p is the number of regressors observed on all the units.

A gaussian parametric model relates Y_i and $x_{1i}, x_{2i}, \dots, x_{pi}$ assuming:

A) the conditional expected value will be assumed to be a linear combination (*linearity assumption* of the expected value)

$$\mathrm{E}\left[Y_i | x_{1i}, \dots, x_{pi}\right] = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}, \forall i$$

B) the conditional variance will be assumed constant (*homoskedasticity assumption*)

$$\mathrm{Var}\left[Y_i | x_{1i}, \dots, x_{pi}\right] = \sigma^2, \forall i$$

C) the *incorrelation assumption*

$$\mathrm{Cor}\left[Y_i | x_{1i}, \dots, x_{pi}, Y_h | x_{1h}, \dots, x_{ph}\right] = 0, \forall i \neq h$$

D) the name of the model comes from the last assumption, the *gaussianity assumption* regarding conditional distributions

$$Y_i | x_{1i}, \dots, x_{pi} \sim N\big(\mu_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}, \sigma^2\big), \forall i$$

Important remark 2 (Linearity). Linearity in this contest mean two different things:

- the conditional expectedvalue is linear *in the regressors* since its a linear combination of regressors given a set of regression coefficient
- it is also linear *in the parameter*: it's a linear combination in the parameters given a certain value for the regressors

It's important to keep in mind the duality of this concept: at some point we'll make gaussian model more flexible by removing one of this two linearity. We'll define model still linear in the parameters but nonlinear in the regressors.

2.2.0.2 Parameter space and sample space

Definition 2.2.1 (Parameter space). Model parameters to be estimated:

- $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^\top \in \mathbb{R}^{(p+1)}$: $(p+1)$-dimensional real-valued vector
- $\sigma^2 \in \mathbb{R}^+$: positive scalar value

So overall the parameter to be estimated can be collected in a vector $\boldsymbol{\theta} = \left(\boldsymbol{\beta}^\top, \sigma^2\right)^\top \in \boldsymbol{\Theta} = \mathbb{R}^{(p+1)} \times \mathbb{R}^+$ where $\boldsymbol{\Theta}$ is the parameter space that is the set of possible values and the set of parameter to be estimated will be $p+2$

Definition 2.2.2 (Conditional sample space)**.** Its the set of possible observation and is given by

$$\mathbb{R} \times \left\{ (x_{1i}, \dots, x_{pi})^\top, i = 1, \dots, n \right\}$$

where the first $\mathbb{R}$ is due to the dependent variable, which can take any real value, and $\left\{ (x_{1i}, \dots, x_{pi})^\top, i = 1, \dots, n \right\}$ is a discrete sets of points of observed data (covariate patterns), which is treated as they were constants/not random (we're ignoring the distribution of the regressors): in other words here we're conditioning on the observed values of the regressors

Example 2.2.1. In the simple case where we have only a dummy variable it is $(\mathbb{R} \times \{0\}) \cup (\mathbb{R} \times \{1\})$

2.2.0.3 Probability density function (1)

Given the assumption provided before we have that the <u>joint</u> density for all the r.vs. $Y_1, \dots, Y_n$ <u>conditional</u> to the regressor values is

$$\begin{aligned} f(y_1, \dots, y_n | x_{11}, \dots, x_{p1}, x_{1n}, \dots, x_{pn}) &\overset{(1)}{=} \prod_{i=1}^n f(y_i | x_{1i}, \dots, x_{pi}) \\ &\overset{(2)}{=} \prod_{i=1}^n \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} (y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_p x_{pi})^2 \right] \right\} \\ &= \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \right\}^n \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_p x_{pi})^2 \right] \end{aligned}$$

where

- (1) the joint conditional distribution is the product of univariate conditional distribution due to independence assumption (uncorrelation + normality = independence)
- (2) due to gaussian distribution: here we substitute with normal density and exploiting the first assumption regarding expected value and the homoskedasticity one

Now we see this latter is just a multivariate gaussian density function where we have a diagonal variance/covariance matrix: any marginal distribution of a multivariate gaussian is still gaussian as we have by assuming incorrelation we are implicitly saying we have independence

2.2.0.4 Matrix representation

It's useful to formalize our model using matrix representation; if:

- $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)^\top$ is n-dimensional random variable that describes the values for the dependent variable jointly observed on n sample units
- $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$ are the observed sample values;

- $\mathbf{x}_i = (x_{0i}, x_{1i} \ldots, x_{pi})^\top$ contains the value of the regressors for the l'i-th sample unit plus an additional element, which is $x_{0i} = 1, \forall i$, constant/"fake" regressor associated with the intercept.
  Therefore $\mathbf{x}_i$ will have $p+1$ elements, where p is the number of regressors;

- on the other hand we define $\mathbf{x}_{[j]} = (x_{j1}, x_{j2} \ldots, x_{jn})^\top$ as the value for a single regressor ($j = 0, \ldots, p$) observed values for all the units (eg for the intercept $x_{[0]} = (1, 1, \ldots, 1)^\top$)

Thus the regressor matrix (matrix containing all the values of the regressors observed on all the units) is an $n \times (p+1)$ matrix and can be seen alternatively as column vector of rows/units or as row vector of columns/regressors:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{p1} \\ 1 & x_{12} & \ldots & x_{p2} \\ \vdots & \vdots & & \vdots \\ 1 & x_{1n} & \ldots & x_{pn} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^\top \\ \hline \mathbf{x}_2^\top \\ \hline \vdots \\ \hline \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{[0]} \big| \, \mathbf{x}_{[1]} \big| \cdots \mathbf{x}_{[p]} \big| \end{bmatrix}$$

Once we've defined this stuff representation we come up with compact notation for all the remaining stuff we've introduced before.

- first, regarding the **conditional expected value** for a single unit, using the matrix notation, its done by vector multiplication of its covariate times the betas

  $$\mathbb{E}\left[Y_i | x_{1i}, \ldots, x_{pi}\right] = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} = \mathbf{x}_i^\top \boldsymbol{\beta}, \quad \forall i$$

  This for a single unitrandom variable, but we can express the vector of conditional expected value for all the sample as

  $$\mathbf{X}\boldsymbol{\beta} = \begin{bmatrix} \mathbf{x}_1^\top \boldsymbol{\beta} \\ \mathbf{x}_2^\top \boldsymbol{\beta} \\ \vdots \\ \mathbf{x}_n^\top \boldsymbol{\beta} \end{bmatrix} = \mathbb{E}\left[\mathbf{Y}\right]$$

- for what concerns the **probability density function** we can express the equation found before more compactly as

  $$\begin{aligned} f\left(\mathbf{y} | \mathbf{X}; \boldsymbol{\beta}, \sigma^2\right) &= \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \right\}^n \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi})^2 \right] \\ &= \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \right\}^n \exp\left[-\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right] \end{aligned}$$

  where we have rewritten the sum of squares in square brackets as dot product of the vector containing the elements of the sum $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ times itself (we have the sum of squares of differences between observed values $\mathbf{y}$ and expected values $\mathbf{X}\boldsymbol{\beta}$)

- according to assumptions A) to E), thus $\mathbf{Y}$, given the regressor values is distributed as multivariate Gaussian being composed by single gaussian, with expected value as derived before, and diagonal varcov matrix (common variance and no covariance between variables)

  $$\mathbf{Y}|\mathbf{X} \sim MVN_n\big(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n\big)$$

  This is a more compact notation we can use when referring to a Gaussian linear regression model

2.2.0.5 An alternative definition

An equivalent alternative definition for the family of gaussian model; the previous definition was focused on assumption of the conditional distribution of Y given the regressor. There's a completely equivalent way to express gaussian models which start from a different starting point. Considering:

- Y_i a random variable that describes the value for the dependent variable observed on the i-th sample unit ($i = 1, \dots, n$)
- $x_{1i}, x_{2i}, \dots, x_{pi}$ the values of the regressors for the i-th sample unit (*covariate pattern*)

rather than focusing on conditional distribution we start assuming that each random variable Y_i in the sample can be decomposed in the sum of two quantities: the deterministic component and the random error.

$$Y_i = \underbrace{\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi}}_{\text{deterministic component}} + \underbrace{\quad \varepsilon_i \quad}_{\text{random error}}$$

This formulation with random error latter emcompass the fact that there's no deterministic relation between X and Y or, in other terms, there's something that cannot be explained in Y_i using only X (so unit with the same X are allowed to have different Y_i).
In this setting the assumptions are focused on the error and especially on its conditional distribution given the regressors:

A) the conditional expected value is null for all the units (some will be positive, some negative but on average it cancels out):

$$\mathrm{E}\left[\varepsilon_i | x_{1i}, \dots, x_{pi}\right] = 0, \forall i$$

B) the conditional variance is constant/independent

$$\mathrm{Var}\left[\varepsilon_i | x_{1i}, \dots, x_{pi}\right] = \sigma^2, \forall i$$

C) there's no conditional correlation between error of different unit

$$\mathrm{Cor}\left[\varepsilon_i | x_{1i}, \dots, x_{pi}, \varepsilon_h | x_{1h}, \dots, x_{ph}\right] = 0, \forall i \neq h$$

D) the conditional distribution of the random error is gaussian

$$\varepsilon_i | x_{1i}, \dots, x_{pi} \sim N\big(0, \sigma^2\big), \forall i$$

Putting all things together considering the sample we have that:

- the vector $\boldsymbol{\varepsilon}$ containing all the unit error $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^\top \ldots$
- has a conditional distribution which is a multivariate gaussian distribution with an expected values vector of 0 and diagonal variance covariance matrix with constant diagonal: $\boldsymbol{\varepsilon}|\mathbf{X} \sim MVN_n\big(\mathbf{0}_n, \sigma^2\mathbf{I}_n\big)$
- now since $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ given the fact that $\mathbf{Y}$ is a linear transformation of $\boldsymbol{\varepsilon}$ ($\mathbf{Y} = \mathbf{A}(\boldsymbol{\varepsilon} + \mathbf{b})$ with $\mathbf{A} = \mathbf{I}_n$ and $\mathbf{b} = \mathbf{X}\boldsymbol{\beta}$), thanks to the properties of multivariate Gaussian distributions, we have that the conditional distribution of $\mathbf{Y}$ is multivariate gaussian as well with the following distribution

$$\mathbf{Y}|\mathbf{X} \sim MVN_n\big(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n\big)$$

2.3 Maximum likelihood estimation

2.3.1 Likelihood and related quantities

2.3.1.1 Likelihood function

The unknown parameters in a Gaussian linear regression models are

- $\boldsymbol{\beta}$ regression coefficients (including the intercept)
- σ^2 conditional variance

The betas are of major interest in the estimation process while σ^2 is a parameter which is estimated and informative but typically of less interest.
Given the regressor values in matrix $\mathbf{X}$ and the observed values for the dependent variable on the sample units in vector $\mathbf{y}$, the likelihood function is obtained using the joint conditional density function which is the product of the single conditional density functions for each unit

$$\begin{aligned}
L\big(\boldsymbol{\beta}, \sigma^2\big) &= L\big(\boldsymbol{\beta}, \sigma^2; \mathbf{y}\,|\mathbf{X}\big) \\
&= \left\{\frac{1}{\sqrt{2\pi\sigma^2}}\right\}^n \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi})^2\right] \\
&= \left\{\frac{1}{\sqrt{2\pi\sigma^2}}\right\}^n \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right]
\end{aligned}$$

We look at this function considering $\mathbf{y}$ and $\mathbf{X}$ fixed and we want to maximize it by choosing the unknown parameters.
Several function can be obtained starting from the likelihood function, developed in what follows.

2.3.1.2 Log-likelihood function

There are practical (with the log we have sum and dealing with maximization of sum is easier than dealing with maximization of product) and technical/theorical reasons for using the log likelihood, which is:

$$\begin{aligned} l\big(\boldsymbol{\beta},\sigma^2\big) &= \ln L\big(\boldsymbol{\beta},\sigma^2\big) \\ &= -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\beta_0-\beta_1 x_{1i}-\ldots-\beta_p x_{pi})^2 \\ &= -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \end{aligned}$$

Note that the first term $-\frac{n}{2}\ln 2\pi$ is an additive constant independent from the unknown parameters and it can be ignored in the maximization.

2.3.1.3 Score function for β

Development Starting from the likelihood function there are other functions that can be derived and are needed in the maximization:

- the *score function* $U(\boldsymbol{\beta}) = \frac{\partial}{\partial\boldsymbol{\beta}}\ln L\big(\boldsymbol{\beta},\sigma^2\big)$ is the gradient of $l\big(\boldsymbol{\beta},\sigma^2\big)$ with respect to $\boldsymbol{\beta}$ (it's a *vector with* $p+1$ *elements*); in other words ...

- each of its element $U_j(\boldsymbol{\beta}) = \frac{\partial}{\partial\beta_j}\ln L\big(\boldsymbol{\beta},\sigma^2\big)$, $j=0,\ldots,p$ is the first partial derivative of $l\big(\boldsymbol{\beta},\sigma^2\big)$ with respect to β_j $(j=0,\ldots,p)$

$$\begin{aligned} U_j(\boldsymbol{\beta}) &= \frac{\partial}{\partial\beta_j}\left\{-\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\beta_0-\beta_1 x_{1i}-\ldots-\beta_p x_{pi})^2\right\} \\ &= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\beta_0-\beta_1 x_{1i}-\ldots-\beta_p x_{pi})\cdot 2\cdot(-1)\cdot x_{ji} \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\beta_0-\beta_1 x_{1i}-\ldots-\beta_p x_{pi})x_{ji} \end{aligned}$$

  so it ends multiplying the residual time the x of the considered beta over σ^2

Remark 2. per l'esame dice di non chiedere la derivazione ma ci potrebbero essere domande riguardo l'espressione finale

Matrix representation for $U(\beta)$ Exploiting the dot product we can express the score function in matrix form:

- the single element of the vector will be

$$U_j(\boldsymbol{\beta}) = \frac{\sum_{i=1}^{n}(y_i-\beta_0-\beta_1 x_{1i}-\ldots-\beta_p x_{pi})x_{ji}}{\sigma^2} = \frac{\mathbf{x}_{[j]}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^2}$$

- the full vector can be expressed as

$$U(\boldsymbol{\beta}) = \begin{bmatrix} \frac{\mathbf{x}_{[0]}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^2} \\ \frac{\mathbf{x}_{[1]}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^2} \\ \vdots \\ \frac{\mathbf{x}_{[p]}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^2} \end{bmatrix} = \frac{\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^2}$$

So calculating the score function is easy for a computer by applying this last equation

An alternative derivation of $U(\beta)$ An equivalent way to express the score function exploits the differentiation rules for functions with vector arguments (we get the same results we can obtain it in the last way if we dont know these tools)

$$U(\boldsymbol{\beta}) = \frac{\partial}{\partial\boldsymbol{\beta}} l(\boldsymbol{\beta},\sigma^2) = \frac{\partial}{\partial\boldsymbol{\beta}}\left\{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right\}$$

$$-\frac{1}{2\sigma^2}\frac{\partial}{\partial\boldsymbol{\beta}}\left\{\mathbf{y}^\top\mathbf{y} - \underbrace{\mathbf{y}^\top\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y}}_{2\boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{y}} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}\right\}$$

Then recalling that $\frac{\partial}{\partial\boldsymbol{\delta}}\boldsymbol{\delta}^\top\mathbf{A} = \mathbf{A}$ and $\frac{\partial}{\partial\boldsymbol{\delta}}\boldsymbol{\delta}^\top\mathbf{A}\boldsymbol{\delta} = 2\mathbf{A}\boldsymbol{\delta}$

$$U(\boldsymbol{\beta}) = -\frac{1}{2\sigma^2}\left\{\mathbf{0}_{p+1} - 2\boldsymbol{X}^\top\boldsymbol{y} + 2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta}\right\} = \frac{\boldsymbol{X}^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})}{\sigma^2}$$

Remark 3. non ci ha speso molto, forse tornerà utile in futuro

2.3.1.4 Observed Fisher information for β

Derivation We have that

- the observed Fisher information is the negative of the Hessian matrix of the loglike function $l(\boldsymbol{\beta},\sigma^2)$ with respect to $\boldsymbol{\beta}$, so it's $(p+1)\times(p+1)$ *matrix*:

$$i(\boldsymbol{\beta}) = -\frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top}\ln L(\boldsymbol{\beta},\sigma^2)$$

- its generic element $i_{jl}(\boldsymbol{\beta})$ is the second partial derivative of $l(\boldsymbol{\beta},\sigma^2)$ with respect to β_j and β_l $(j,l = 0,\ldots,p)$

$$i_{jl}(\boldsymbol{\beta}) = -\frac{\partial^2}{\partial\beta_j\partial\beta_l}\ln L(\boldsymbol{\beta},\sigma^2)$$

son on a practical pov we have

$$\begin{aligned} i_{jl}(\boldsymbol{\beta}) &= -\frac{\partial}{\partial \beta_l} U_j(\boldsymbol{\beta}) \\ &= -\frac{\partial}{\partial \beta_l} \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi}) x_{ji} \\ &= -\frac{1}{\sigma^2} \sum_{i=1}^{n} x_{ji} x_{li} \cdot (-1) = \frac{1}{\sigma^2} \sum_{i=1}^{n} x_{ji} x_{li} \end{aligned}$$

we end up with a rather simple expression involving the sum of cross product of the j-th and the l-th regressors

Remark 4. Called information because we're measuring the curvature of the loglikelihood function so the more loglikelihood function is curved the more information we have regarding our estimate to be the best

Matrix representation Again exploiting the dot product we can have a compact representation

- starting from the single element we have that

$$i_{jl}(\boldsymbol{\beta}) = \frac{\sum_{i=1}^{n} x_{ji} x_{li}}{\sigma^2} = \frac{\mathbf{x}_{[j]}^\top \mathbf{x}_{[l]}}{\sigma^2}$$

- then for the full matrix

$$i(\boldsymbol{\beta}) = \begin{bmatrix} \frac{\mathbf{x}_{[0]}^\top \mathbf{x}_{[0]}}{\sigma^2} & \frac{\mathbf{x}_{[0]}^\top \mathbf{x}_{[1]}}{\sigma^2} & \ldots & \frac{\mathbf{x}_{[0]}^\top \mathbf{x}_{[p]}}{\sigma^2} \\ \frac{\mathbf{x}_{[1]}^\top \mathbf{x}_{[0]}}{\sigma^2} & \frac{\mathbf{x}_{[1]}^\top \mathbf{x}_{[1]}}{\sigma^2} & \ldots & \frac{\mathbf{x}_{[1]}^\top \mathbf{x}_{[p]}}{\sigma^2} \\ \vdots & & & \\ \frac{\mathbf{x}_{[p]}^\top \mathbf{x}_{[0]}}{\sigma^2} & \frac{\mathbf{x}_{[p]}^\top \mathbf{x}_{[1]}}{\sigma^2} & \ldots & \frac{\mathbf{x}_{[p]}^\top \mathbf{x}_{[p]}}{\sigma^2} \end{bmatrix} = \frac{\mathbf{X}^\top \mathbf{X}}{\sigma^2}$$

An alternative derivation Again, exploiting the differentiation rules for functions with vector arguments, we end with exactly the same results

$$i(\boldsymbol{\beta}) = -\frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} l(\boldsymbol{\beta}, \sigma^2) = -\frac{\partial}{\partial \boldsymbol{\beta}^\top} U(\boldsymbol{\beta}) = -\frac{1}{\sigma^2} \frac{\partial}{\partial \boldsymbol{\beta}^\top} \left(\mathbf{X}^\top \mathbf{y} - \mathbf{X}^\top \mathbf{X} \boldsymbol{\beta} \right)$$

and recalling that $\frac{\partial}{\partial \boldsymbol{\delta}^\top} \mathbf{A} \boldsymbol{\delta} = \mathbf{A}$

$$i(\boldsymbol{\beta}) = -\frac{1}{\sigma^2} \left[\mathbf{0}_{(p+1) \times (p+1)} - \mathbf{X}^\top \mathbf{X} \right] = \frac{\mathbf{X}^\top \mathbf{X}}{\sigma^2}$$

Remark 5. again non ci ha speso molto

2.3.1.5 Expected Fisher information for β

The observed information is a quantity that is specific to a given sample. Along with the observed there's the expected Fisher information as well: it's the expected value of the obserserved Fisher information across the possible samples

- it's a $(p+1) \times (p+1)$ matrix defined as
$$I(\boldsymbol{\beta}) = \mathrm{E}\left[i(\boldsymbol{\beta})\right] = -\mathrm{E}\left[\frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top}\ln L\big(\boldsymbol{\beta},\sigma^2\big)\right]$$
- the generic element is the expected value of the generic element of $i(\boldsymbol{\beta})$, that is
$$I_{jl}(\boldsymbol{\beta}) = \mathrm{E}\left[i_{jl}(\boldsymbol{\beta})\right] = \mathrm{E}\left[\underbrace{\frac{\sum_{i=1}^n x_{ji}x_{li}}{\sigma^2}}_{\text{independent of }\mathbf{Y}}\right] = \frac{\sum_{i=1}^n x_{ji}x_{li}}{\sigma^2}$$
and it turns out to depends only on the value of the j-th and l-th regressor. Note that the expected values are computed considering the conditional distribution of $\mathbf{Y}$ given $\mathbf{X}$ (thus holding fixed the values of the regressors so in our sample the only random quantity is Y which does not figure under expected value which have just a constant)
- finally
$$\mathrm{E}\left[i(\boldsymbol{\beta})\right] = \frac{\mathbf{X}^\top\mathbf{X}}{\sigma^2}$$

Important remark 3. So when dealing with gaussian linear regression models the observed and expected fisher information coincides: this sample in the sample space is as informative as any sample in the sample space with respect to the unknown parameters.
This is something that does not happen all the time; for other more complicated models this does not hold. We will appreciate the benefit of this equivalence when it comes to GLM

2.3.1.6 Properties of the score function

Differently from the fisher information matrixes, the score function $U(\boldsymbol{\beta}) = \dfrac{\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^2}$ is a function/quantity which depends on

- $\boldsymbol{\beta}, \sigma^2$ the *unknown* model parameters
- $\mathbf{X}$ the regressor values
- $\mathbf{y}$ the observed values of the dependent variable (*realisations of the r. v.* $\mathbf{Y}$)

So each sample will be characterized by a different log-likelihood function and score function: the first partial derivative can be different from sample to sample but the second partial derivatives (and information matrix) will be constant (if we condition on X)

We may think as the score function as a random variable itself, being a linear transformation of vector $\mathbf{Y}$; will have it's expected variable, varcov matrix etc. In gaussian linear regression models we can come up with the distribution of the score function:

- conditionally on the regressors values, $U(\boldsymbol{\beta})$ is the realisation of a random vector, that can be expressed as a linear transformation of $\mathbf{Y}$:

$$\mathbf{A} = \frac{\mathbf{X}^\top}{\sigma^2},\ \mathbf{b} = -\mathbf{X}\boldsymbol{\beta} \implies U(\boldsymbol{\beta}) = \mathbf{A}(\mathbf{Y} + \mathbf{b})$$

- assuming the Gaussian linear model assumptions, thanks to the properties of MVN gaussian, if we do the math we end with the fact that the conditional distribution of the score function given $\mathbf{X}$ is MVN as well (with $p+1$ elements) with 0 mean (it's independent of beta, whichever value they have) and variance covariance matrix coinciding with expected fisher information matrix

$$\mathbf{Y}|\mathbf{X} \sim MVN_n\big(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n\big) \implies U(\boldsymbol{\beta})|\mathbf{X} \sim MVN_{p+1}\left(\underbrace{\frac{\mathbf{X}^\top}{\sigma^2}[\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta}]}_{\mathbf{0}_{p+1}}, \underbrace{\frac{\mathbf{X}^\top}{\sigma^2}\sigma^2\mathbf{I}_n\frac{\mathbf{X}}{\sigma^2}}_{\frac{\mathbf{X}^\top\mathbf{X}}{\sigma^2} = I(\boldsymbol{\beta})}\right)$$

the fact that score function has MVN distribution is very simple in gaussian models, but this results can be extended to other kind of models as well.

2.3.1.7 Standardising the score function

We can standardise the score function which make the MVN to have a varcov equal to the identity matrix.
In principle we define the square root of fisher expected information matrix that is $I(\boldsymbol{\beta})^{-\frac{1}{2}}$ such that:

$$\begin{aligned} &I(\boldsymbol{\beta}) = I(\boldsymbol{\beta})^{\frac{1}{2}} I(\boldsymbol{\beta})^{\frac{1}{2}} \\ &I(\boldsymbol{\beta})^{\frac{1}{2}} I(\boldsymbol{\beta})^{-\frac{1}{2}} = I(\boldsymbol{\beta})^{-\frac{1}{2}} I(\boldsymbol{\beta})^{\frac{1}{2}} = \mathbf{I}_{p+1} \end{aligned}$$

We have that $I(\boldsymbol{\beta})^{-\frac{1}{2}} = \sigma\big(\mathbf{X}^\top\mathbf{X}\big)^{-\frac{1}{2}}$ exists if and only if the matrix $\mathbf{X}^\top\mathbf{X}$ is invertible, that is, if and only if $\mathbf{X}$ has full column rank.
If the square root exists we can transform the score function such that we have a resulting standardized MVN as follows:

$$I(\boldsymbol{\beta})^{-\frac{1}{2}} U(\boldsymbol{\beta}) \sim MVN_{p+1}(\mathbf{0}_{p+1}, \mathbf{I}_{p+1})$$

Furthermore we have that the following quadratic form is distributed as chi-square

$$U(\boldsymbol{\beta})^\top I(\boldsymbol{\beta})^{-1} U(\boldsymbol{\beta}) = \frac{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{X}\big(\mathbf{X}^\top\mathbf{X}\big)^{-1}\mathbf{X}^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})}{\sigma^2} \sim \chi^2_{p+1}$$

this expression is nothing but the sum of the squared elements of the standardized score function; if the standardized score function has elements that are all standard gaussian random variables (and those RV are also independent) then they squared sum will be chi-square.

This idea of working with standardized stuff can work theoretically: we will be never be able to compute the actual value of the score function observed in our sample (because it depends on unknown quantity betas and σ^2). However the behaviour of the standardized score function is crucial in order to study the properties of the Maximum likelihood estimators, especially in context different from the standard gaussian, where we aren't able to come up with a closed formula expression to compute the maximum likelihood estimate and we have to use numerical maximization procedures for loglikelihood (while this stuff is unused in the more simple linear regression).

2.3.1.8 Some general properties of the score function

Aside a moment from gaussian model, in general, let

- $L(\boldsymbol{\theta})$ be likelihood function associated with *any* given parametric statistical model ($\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^k$)
- we are able to compute the score function $U(\boldsymbol{\theta}) = \dfrac{\partial}{\partial \boldsymbol{\theta}} \ln L(\boldsymbol{\theta})$

Then under general regularity conditions it is possible to prove that whatever parametric model we're dealing with we have that:

- the expected value of the score function is the null vector: $\mathrm{E}\left[U(\boldsymbol{\theta})\right] = \mathbf{0}_k$
- its variance covariance is equal to the expected fisher information matrix: $\mathrm{Var}\left[U(\boldsymbol{\theta})\right] = I(\boldsymbol{\theta})$
- its standardized version converge in probability to a standardized multivariate normal (zero mean and identity variance covariance matrix): $I(\boldsymbol{\theta})^{-\frac{1}{2}} U(\boldsymbol{\theta}) \xrightarrow{d} MVN_k(\mathbf{0}_k, \mathbf{I}_k)$.
  The idea of standardizing the score function is crucial for studying its asymptotic behaviour: from a technical pov it's possible to prove (look intermediate/advanced texts) that standardized version of the score function converge in distribution to multivariate normal.

These results implies that we can always approximate the distribution of the original score funtion using a MVN with zero expected value and expected fisher information matrix as variance covariance

$$U(\boldsymbol{\theta}) \approx MVN_k(\mathbf{0}_k, I(\boldsymbol{\theta}))$$

The quality of the approximation improves as sample size increase.

So we put aside this results for the future: if some general condition are met our score function can be approximated by a MVN

2.3.2 Maximum likelihood estimation

2.3.2.1 Maximum likelihood estimate for β

The vector $\hat{\mathbf{b}}$ is the maximum likelihood (ML) estimate for $\boldsymbol{\beta}$ if and only if

$$l\left(\hat{\mathbf{b}}, \sigma^2\right) = \max_{\mathbf{b} \in \mathbb{R}^{p+1}} l\left(\mathbf{b}, \sigma^2\right)$$

or equivalently

$$\hat{\boldsymbol{b}} = \arg\max_{\mathbf{b}\in\mathbb{R}^{p+1}} l\big(\mathbf{b}, \sigma^2\big)$$

To find it we need to find the value $\hat{\mathbf{b}}$

- at which the score function (first derivative) is equal to 0, or in matrix terms *the log-likelihood gradient with respect to $\boldsymbol{\beta}$ evaluated at* $\hat{\mathbf{b}}$ be zero vector
$$U\left(\hat{\mathbf{b}}\right) = \frac{\partial}{\partial \boldsymbol{\beta}} l\big(\boldsymbol{\beta}, \sigma^2\big)\bigg|_{\boldsymbol{\beta}=\hat{\mathbf{b}}} = \mathbf{0}_{p+1}$$
- among the several point matching the first condition to have a maximum we need htat second partial derivative evaluated at point must be negative, or in matrix terms the Hessian matrix of the log likelihood function (for $\boldsymbol{\beta}$ evaluated at $\hat{\mathbf{b}}$) be negative definite (equivalent for a matrix of a negative scalar)
$$H\left(\hat{\mathbf{b}}\right) = \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} l\big(\boldsymbol{\beta}, \sigma^2\big)\bigg|_{\boldsymbol{\beta}=\hat{\mathbf{b}}}$$
in this way $\mathbf{z}^\top H\left(\hat{\mathbf{b}}\right)\mathbf{z} < 0$, $\forall \mathbf{z} \neq \mathbf{0}_{p+1}$

So to find maximum we have to find b vectors satysifing these condition

- starting from the first one the score vector is null
$$\begin{aligned} U(\mathbf{b}) = \mathbf{0}_{p+1} &\iff \frac{\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{b})}{\sigma^2} = \mathbf{0}_{p+1} \\ &\iff \frac{\mathbf{X}^\top \mathbf{y}}{\sigma^2} = \frac{\mathbf{X}^\top \mathbf{X}\mathbf{b}}{\sigma^2} \end{aligned}$$
Now if the matrix $\mathbf{X}$ has full column rank (its column are linearly independent) then $\mathbf{X}^\top\mathbf{X}$ is invertible and we can premultiply both terms of the equation for $(X^\top X)^{-1}$ obtaining
$$\mathbf{b} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}$$
- for the second partial derivative we have that
$$H(\boldsymbol{\beta}) = -i(\boldsymbol{\beta}) = -\frac{\mathbf{X}^\top\mathbf{X}}{\sigma^2}$$
σ^2 is unknown but must be positive; if the matrix $\mathbf{X}$ has full column rank any combination of columns will be a vector different from zero vector, so $z^\top X$ will be different from zero, then $z^\top X^\top X z$ will be always strictly positive scalar and with a minus behind the results will be a negative constant. So $\forall \mathbf{b} \in \mathbb{R}^{p+1}$ we have that $H(\boldsymbol{\beta})$ is negative definite.
So we have the assurance that this is a maximum

Therefore our mle is

$$\hat{\mathbf{b}} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}$$

and as far as $\boldsymbol{\beta}$ is concerned, maximum likelihood estimation is equivalent to least square estimation for Gaussian linear models

2.3.2.2 Properties of the ML estimator for β

The MLE estimators $\hat{\mathbf{b}} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}$ depends on:

- $\mathbf{X}$ the regressor values
- $\mathbf{y}$ the observed outcomes (*realisations of the r. v.* $\mathbf{Y}$)

Conditionally on the regressors values $\mathbf{X}$, $\hat{\mathbf{b}}$ is the realisation of a random vector, that can be expressed as a linear transformation of $\mathbf{Y}$

$$\mathbf{A} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top, \mathbf{b} = \mathbf{0}_{p+1} \implies \hat{\mathbf{B}} = \mathbf{A}\mathbf{Y} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y}$$

Thus according to the Gaussian linear model assumption, having $\mathbf{Y}|\mathbf{X} \sim MVN_n\left(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n\right)$ it turns out that the ML estimator is respectively unbiased, having a varcov matrix corresponding to the inverse of the expected information matrix and gaussian:

$$\begin{aligned}
&\mathrm{E}\left[\hat{\mathbf{B}}\middle|\mathbf{X}\right] = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}\\
&\mathrm{Var}\left[\hat{\mathbf{B}}\middle|\mathbf{X}\right] = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\sigma^2\mathbf{I}_n\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} = \sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} = I(\boldsymbol{\beta})^{-1}\\
&\hat{\mathbf{B}}|\mathbf{X} \sim MVN_{p+1}\left(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}\right)
\end{aligned}$$

Furthermore regarding variability, thanks to Rao-Cramer theorem, having variance covariance matrix coinciding with the inverse of expected information matrix we conclude that the MLE is the efficient estimator for β.
(inverse of cramer-rao lower bound is the minimum variance that we can achieve for an unbiased estimator for a given parameter: if exists an estimator achieving the cramer-rao lower bound, then that estimator is unique, so there are no other estimators with less variability).
So if the model assumption holds the mle estimator for beta are not only unbiased but also efficient.

It is important to check whether the assumptions are adequate for the specific dataset we're dealing with

2.3.2.3 Some general results related to ML method

Some general words on ML method: let

- $\hat{\mathbf{T}}$ be Maximum likelihood estimator for $\boldsymbol{\theta}$ (a random variable on the sample space)
- $\hat{\mathbf{t}} = \arg\max_{\mathbf{t}\in\Theta} l(\mathbf{t})$ be the maximum likelihood estimate for $\boldsymbol{\theta}$ (*sample realisation of* $\hat{\mathbf{T}}$)

Under general regularity conditions (the same for property of the score function) it is possibile to show that:

- the standardized version of the ML estimator has an asymptotic distribution converging to standard MVN $I(\boldsymbol{\theta})^{\frac{1}{2}}\left(\hat{\mathbf{T}} - \boldsymbol{\theta}\right) \xrightarrow{d} MVN_k(\mathbf{0}_k, \mathbf{I}_k)$;
- thus in general $\hat{\mathbf{T}} \approx MVN_k\left(\boldsymbol{\theta}, I(\boldsymbol{\theta})^{-1}\right)$

So no matter what model we are dealing with, if the model satisfies the basic regularity conditions, whatever functional form it takes (even when an explicit analytical form for computing $\hat{\mathbf{t}}$ does not exist)

- the ML estimator for $\boldsymbol{\theta}$ is *asymptotically unbiased* (we have no guarantee that it is unbiased but at least asymptotically when sample size increase it is)
- it is *asymptotically efficient* (having the asymptotic variance covariance matrix coinciding with the inverse of the expected fisher information, the cramer rao lower bound).

For gaussian linear model they are unbiased and efficient as well (for any sample size).

2.3.2.4 Maximum likelihood estimate for σ^2

The other parameter of the gaussian model is σ^2; when performing regression analysis main focus is the betas, but we still have a variance so we need to compute the estimate.
Once we've found the ML estimates for the regression coefficients we can look for a ML estimate for the σ^2. It is possible to prove (compute the first and second partial derivative of loglikelihood with respect to σ^2, set the first equal to zero and select among them where second partial derivative is negative: here is a scalar so it's a simple real valued function) that the estimator of σ^2, s^2 is basically the variance of sample raw residuals

$$\hat{s}^2 = \underset{s^2 \in \mathbb{R}^+}{\arg\max}\, l\left(\hat{\mathbf{b}}, s^2\right) = \frac{\sum_{i=1}^n \left(y_i - \mathbf{x}_i^\top \hat{\mathbf{b}}\right)^2}{n} = \frac{\left(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\right)^\top \left(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}\right)}{n} = \frac{\mathbf{e}^\top \mathbf{e}}{n}$$

There will be a score function, an observed and expected information matrix as well but we don't focus on it being less interesting for our purpose.
A single raw residual is something like

$$e_i = y_i - \mathbf{x}_i^\top \hat{\mathbf{b}}, \quad i = 1, \dots, n$$

while exploiting matrix algebra we see the vector of raw residuals $\mathbf{e}$ can be expressed as

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\underbrace{\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}}_{\hat{\mathbf{b}}} = \left[\mathbf{I}_n - \underbrace{\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top}_{\mathbf{H}}\right]\mathbf{y} = \underbrace{[\mathbf{I}_n - \mathbf{H}]}_{\mathbf{M}}\mathbf{y}$$

It ends with $\mathbf{e}$ being expressed as linear transformation of the vector $\mathbf{y}$

- $\mathbf{H}$ the so called *hat matrix* (which transforms the observed $\mathbf{y}$ in the fitted $\hat{\mathbf{y}} = X\hat{\mathbf{b}}$
- $\mathbf{M} = \mathbf{I}_n - \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top$, sometimes referred as *residual maker matrix* (transforming $\mathbf{y}$ into $\mathbf{e}$), is
  - symmetric
  - idempotent ($\mathbf{MM} = \mathbf{M}$)
  - usually not diagonal
  - not invertible

2.3.2.5 Properties of raw residuals

Given the Gaussian linear model assumptions, it is possible to prove that:

- being a linear trasformation of vector $\mathbf{y}$ will have a multivariate gaussian distribution (if the gaussian assumption are ok its mean is 0 while the varcov is not diagonal):

$$\mathbf{e}|\mathbf{X} \sim MVN_n\left(\mathbf{0}_n, \sigma^2\mathbf{M}\right)$$

- it can be proved that if model assumptions holds, the sum of the squares of residuals divided by σ^2 (conditional on the value of the regressors), is distributed as a Chi square with $n-(p+1)$ (where p is the number of regressors) degrees of freedom:

$$\left.\frac{\mathbf{e}^\top\mathbf{e}}{\sigma^2}\right|\mathbf{X} \sim \chi^2_{n-p-1}$$

- therefore the expected value of the sum of the squares of residuals conditional to the regressor is (taking

$$\mathrm{E}\left[\mathbf{e}^\top\mathbf{e}|\mathbf{X}\right] = \sigma^2(n-p-1)$$

- $\dfrac{\mathbf{e}^\top\mathbf{e}}{\sigma^2}$ is independent of $\hat{\mathbf{B}}$ (ML estimates of $\boldsymbol{\beta}$)

These properties are crucial for establishing the properties of ML estimator for σ^2

2.3.2.6 Properties of the maximum likelihood estimator for σ^2

The estimator for σ^2 is

$$\hat{S}^2 = \frac{\left(\mathbf{Y}-\mathbf{X}\hat{\mathbf{B}}\right)^\top\left(\mathbf{Y}-\mathbf{X}\hat{\mathbf{B}}\right)}{n}$$

Given the Gaussian linear model assumptions, and exploiting the properties of the raw residuals, it is possible to prove that:

- the maximum likelihood estimator for σ^2 is biased since

$$\mathrm{E}\left[\hat{S}^2\middle|\mathbf{X}\right] = \sigma^2\frac{n-p-1}{n} \neq \sigma^2$$

  The (negative) bias we have (the ML estimate tend to underestimate the true σ^2) get cancelled out as n increases (so ML estimator is asymptotically unbiased as seen before), that is

$$\mathrm{E}\left[\hat{S}^2\middle|\mathbf{X}\right] \underset{n\to\infty}{\longrightarrow} \sigma^2$$

- if we are interested in unbiased estimator for σ^2 a corrected expression is obtaining by dividing by $n-p-1$ (degrees of freedom) instead of n

$$S^2 = \frac{\left(\mathbf{Y}-\mathbf{X}\hat{\mathbf{B}}\right)^\top\left(\mathbf{Y}-\mathbf{X}\hat{\mathbf{B}}\right)}{n-p-1}$$

(this is nothing but a generalization of what occurs with the sample variance where using the mean make the degrees of freedom to decrease of 1) The unbiased version (not the ML one) is typically what is returned from software to estimate σ^2

- both ML and unbiased estimator $\hat{S}^2$, S^2 are independent of $\hat{\mathbf{B}}$.
  This is important property when we want to compute test statistics (all the most relevant test statistics can be expressed as ratio between numerator depending on ML estimates of beta and the denominator depending on an estimate of σ^2: knowing that num and denom are independent makes the derivation of distribution of test statistics under null hypothesis much easier.

2.3.2.7 Standardised residuals

We have seen that raw residual have a more or less known distribution (not considering σ^2)

$$\mathbf{e}|\mathbf{X} \sim MVN_n\big(\mathbf{0}_n, \sigma^2\mathbf{M}\big)$$

In general $\mathbf{M}$:

- is not diagonal, so resids are not independent
- is usually not homoschedastic: its diagonal elements of $\mathbf{M}$ differ from one another. They differ because the diagonal elements of M depend of the value of the regressors associated with each unit in the sample, eg for the i-th unit it will be

  $$\mathbf{M}_{ii} = 1 - \mathbf{x}_i^\top \big(\mathbf{X}^\top\mathbf{X}\big)^{-1}\mathbf{x}_i = 1 - \mathbf{H}_{ii}$$

  units with different covariate pattern will have residuals whose variance is different

The variance of each residual depends on σ^2 (for which we have an unbiased estimator) and the $\mathbf{M}$ diagonal entry; these latter are a function of the regressors, so condition on the regressor we can compute the exact value.
Starting from the raw residual we can come up with two refined version:

- **Pearson residuals**: $e_i^P = \dfrac{e_i}{\sqrt{s^2}} \quad i = 1, \dots, n$ obtained dividing each raw residual for the square root of unbiased estimate of σ^2.
  In this way however we don't obtain yet a residual which is homoschedasti but it will show up as well in the GLM
- **Standardised residuals**: $r_i = \dfrac{e_i}{\sqrt{s^2(1-\mathbf{H}_{ii})}} \quad i = 1, \dots, n$ which divided the residual for a measure that take into account the i-th diagonal element on the residual maker matrix).

If all the assumptions of gaussian model are met it is possible to prove that we have that the asymptotic is the following

$$\mathbf{r} = (r_1, r_2, \dots, r_n)^\top \Big| \mathbf{X} \xrightarrow{d} MVN_n(\mathbf{0}_n, \mathbf{I}_n)$$

Approximately (by n fixed), standardised residuals from a Gaussian linear models are equivalent to an observed sample drawn from an n-dimensional standardised Gaussian random vector.

Important remark 4. Idea: once we fitted the model we can inspect the standardized residuals (eg by plots) and if we find some deviation of behaviour from standard MVN (IID random vector of standard gaussians), then we can conclude that the model assumptions are not adequate.
When assumption holds this would not happen.
This is one of the most crucial step to do

Chapter 3

Linear hypotheses

Important remark 5. Typically we are interested in test hypotheses on parameters: we will focus on linear hypotheses, so called because they can be expressed as a linear system (involving the regression coefficient betas).

3.1 Linear hypotheses

3.1.1 Linear hypotheses on β

Our setup

- Gaussian linear model: $\boldsymbol{Y}|\mathbf{X} \sim MVN_n\big(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n\big)$
- the $(p+1) \times 1$ parameter vector

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

- suppose we have
    - $\mathbf{K}$, $q \times (p+1)$ matrix, composed of known constants with full row rank q (rows are linearly independent: q must be smaller or equal to $p+1$)
    - $\mathbf{t}$, $q \times 1$ vector composed of known constants

  Any linear hypotheses on $\boldsymbol{\beta}$ can be expressed as a system of linear equations:

$$H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{t}$$

  In the latter we're specifying that linear combinations of regressors are equal to given constants

Example 3.1.1 (Linear hypotheses on β: some examples)**.** Supposing $p = 3$, the following are different systems of hypotheses to be tested

(A) if $\mathbf{K} = [0\ 1\ 0\ 0]$, and $\mathbf{t} = 0$ then we obtain a simple test on a single coefficient

$$H_0 : \beta_1 = 0$$

(B) if $\mathbf{K} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $\mathbf{t} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ then we pick two coefficients and put them equal to 0 we have

$$H_0 : \begin{cases} \beta_1 = 0 \\ \beta_3 = 0 \end{cases}$$

or in a more common/compact way

$$\beta_1 = \beta_3 = 0$$

(C) if $\mathbf{K} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $\mathbf{t} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ then

$$H_0 : \begin{cases} \beta_1 = 0 \\ \beta_2 = 0 \\ \beta_3 = 0 \end{cases}$$

or

$$\beta_1 = \beta_2 = \beta_3 = 0$$

which is called *linear independence* test (if null is true there's independence between the dependent variable and the regressors: the latters have no effect on the dependent variable)

(D) if $\mathbf{K} = [0\ 1\ 0\ -1]$, $\mathbf{t} = 0$ then

$$H_0 : \beta_1 = \beta_3$$

In this case by chosing a different $\mathbf{K}$ we relate coefficient between them, not only to constants.
Note that $H_0 : \beta_1 = \beta_3$ is much more general than $H_0 : \beta_1 = \beta_3 = 0$ seen in (B); here they can be equal no matter what value they take. In some situations is useful to test hypothesis on equivalence of regression coefficients without specifying the given value

(E) we can test that a coefficient to be a constant, not necessarily 0: eg if $\mathbf{K} = [0\ 1\ 0\ 0]$ and $\mathbf{t} = 3$

$$H_0 : \beta_1 = 3$$

3.1.2 Nested linear models

Linear hypotheses (A), (B) and (C) in the previous example lead to Gaussian linear models that can be obtained by removing some regressors from the starting model: eg if the starting model is

$$\mathrm{E}\left[Y_i | x_{1i}, x_{2i}, x_{3i}\right] = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i}$$

then:

$$\text{(A)} \;\Rightarrow \mathrm{E}_{H_0}\left[Y_i | x_{1i}, x_{2i}, x_{3i}\right] = \beta_0 + \beta_2 x_{2i} + \beta_3 x_{3i} = \mathrm{E}\left[Y_i | x_{2i}, x_{3i}\right]$$

$$\text{(B)} \;\Rightarrow \mathrm{E}_{H_0}\left[Y_i | x_{1i}, x_{2i}, x_{3i}\right] = \beta_0 + \beta_2 x_{2i} = \mathrm{E}\left[Y_i | x_{2i}\right]$$

$$\text{(C)} \;\Rightarrow \mathrm{E}_{H_0}\left[Y_i | x_{1i}, x_{2i}, x_{3i}\right] = \beta_0 = \mathrm{E}\left[Y_i\right]$$

These are nested models, that is models that are obtained after removing one or more regressors from a starting one.

3.1.3 Likelihood ratio test (LRT) statistics - 1

To compare models we use the LRT statistics, which is a general test which we can use to test any kind of hypothesis on the parameter of a parametric statistical model.
It's defined as

$$LRT = \frac{L\left(\hat{\mathbf{b}}, \sigma^2\right)}{L\left(\hat{\mathbf{b}}_{H_0}, \sigma^2\right)}$$

so as the ratio between

- $\hat{\mathbf{b}} = \arg\max_{\mathbb{R}^{(p+1)}} l\left(\mathbf{b}, \sigma^2\right)$, that is the maximized likelihood (value of likelihood function at the ML estimates)
- $\hat{\mathbf{b}}_{H_0} = \arg\max_{\{\mathbf{b}:\mathbf{Kb}=\mathbf{t}\}\subset\mathbb{R}^{(p+1)}} l\left(\mathbf{b}, \sigma^2\right)$ the maximized likelihood under the restriction imposed by the system of linear hypothesis (that is considering in the parameter space only those elements introduced by the linear restriction, we have a constrainted maximization)

Equivalently, using the loglikelihood we have the differences, that is

$$2\left[l\left(\hat{\mathbf{b}}, \sigma^2\right) - l\left(\hat{\mathbf{b}}_{H_0}, \sigma^2\right)\right]$$

The point now we focus on is how to find the denominator of the LRT

3.2 Constrained maximum likelihood estimation

3.2.1 The Method of Lagrange multipliers

This is a generic method for constrainted optimization: the idea is to work on a slightly different version of the function to be optimized.
$\hat{\mathbf{b}}_{H_0}$ maximises $l\left(\boldsymbol{\beta}, \sigma^2\right)$ in the parameter subspace $\{\mathbf{b} : \mathbf{Kb} = \mathbf{t}\} \subset \mathbb{R}^{(p+1)}$. Rather than maximizing l

- we maximize a modified version l^*
$$l^*(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\alpha}) = l(\boldsymbol{\beta}, \sigma^2) - \boldsymbol{\alpha}^\top(\mathbf{K}\boldsymbol{\beta} - \mathbf{t})$$
where $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_q \end{bmatrix}$ is a $q \times 1$ vector containing unknown *Lagrange multipliers*. So the original likelihood is modified using $\boldsymbol{\alpha}$ and something regarding the linear restriction we're interested in
- the maximization is done with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$: what we found of out with respect of $\boldsymbol{\beta}$ is a set of value satisfying the conditions and maximizing the likelihood under them

Important remark 6. Some technical passages follows: take the main message above and don't worry

To do the maximization, the following system equations must be solved:

$$\begin{cases} U(\mathbf{b}) = \dfrac{\partial}{\partial \boldsymbol{\beta}} l^*(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\alpha})\Big|_{\boldsymbol{\beta}=\mathbf{b}} = \mathbf{0}_{p+1} \\ \\ U(\mathbf{a}) = \dfrac{\partial}{\partial \boldsymbol{\alpha}} l^*(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\alpha})\Big|_{\boldsymbol{\alpha}=\mathbf{a}} = \mathbf{0}_q \end{cases}$$

It's a $p+1+q$ equation sistem where the first $p+1$ are related to betas and last q to alphas).
We have to compute the first partial derivatives with respect both to beta and alphas:

$$\frac{\partial}{\partial \boldsymbol{\beta}} l^*(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\alpha}) = U(\boldsymbol{\beta}) - \frac{\partial}{\partial \boldsymbol{\beta}} \boldsymbol{\alpha}^\top(\mathbf{K}\boldsymbol{\beta} - \mathbf{t}) = \frac{\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{\sigma^2} - \mathbf{K}^\top\boldsymbol{\alpha}$$
$$\frac{\partial}{\partial \boldsymbol{\alpha}} l^*(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\alpha}) = -\frac{\partial}{\partial \boldsymbol{\alpha}} \boldsymbol{\alpha}^\top(\mathbf{K}\boldsymbol{\beta} - \mathbf{t}) = -(\mathbf{K}\boldsymbol{\beta} - \mathbf{t})$$

Remembering general rules:

$$\frac{\partial}{\partial \boldsymbol{\delta}} \mathbf{A}\boldsymbol{\delta} = \mathbf{A}^\top$$
$$\frac{\partial}{\partial \boldsymbol{\delta}} \boldsymbol{\delta}^\top\mathbf{A} = \mathbf{A}$$

The idea is to solve first the first $p+1$ equations with respect to the betas, then we plug the solutions in the remaining q equations.
Consider the first $p+1$ equations:

$$\frac{\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{b})}{\sigma^2} - \mathbf{K}^\top\mathbf{a} = \mathbf{0}_{p+1}$$
$$\frac{\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{b})}{\sigma^2} = \mathbf{K}^\top\mathbf{a}$$
$$\mathbf{X}^\top\mathbf{y} - \mathbf{X}^\top\mathbf{X}\mathbf{b} = \sigma^2\mathbf{K}^\top\mathbf{a}$$
$$\mathbf{X}^\top\mathbf{X}\mathbf{b} = \mathbf{X}^\top\mathbf{y} - \sigma^2\mathbf{K}^\top\mathbf{a}$$

If $\mathbf{X}$ has full column rank $(p+1)$ then

$$\begin{aligned}\hat{\mathbf{b}}_{H_0} &= \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y} - \sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\mathbf{a}\\ &= \hat{\mathbf{b}} - \sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\mathbf{a}\end{aligned}$$

So this last is what we'll use in the remaining equations; note that σ^2 and $\mathbf{a}$ are unknown.
Now exploiting the formula for $\hat{\mathbf{b}}_{H_0}$ in the last q equations:

$$\begin{aligned}\mathbf{K}\left[\hat{\mathbf{b}} - \sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\mathbf{a}\right] &= \mathbf{t}\\ \mathbf{K}\hat{\mathbf{b}} - \sigma^2\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\mathbf{a} = \mathbf{t}\\ \sigma^2\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\mathbf{a} = \mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\end{aligned}$$

If $\mathbf{K}$ has full row rank (q)

$$\hat{\mathbf{a}} = \frac{1}{\sigma^2}\left[\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-1}\left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right)$$

So finally, to obtain the constrained maximum likelihood estimate, by substituting $\hat{\mathbf{a}}$ for $\boldsymbol{\alpha}$ in the formula for $\hat{\mathbf{b}}_{H_0}$:

$$\begin{aligned}\hat{\mathbf{b}}_{H_0} &= \hat{\mathbf{b}} - \sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\frac{1}{\sigma^2}\left[\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-1}\left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right)\\ &= \hat{\mathbf{b}} - \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\left[\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-1}\left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right) \qquad (3.1)\end{aligned}$$

So we can get a general analytical expression (complicated ok but don't worry) to compute the constrainted betas for any possible system of hypotheses.
Note that, even the constrained maximum likelihood estimate $\hat{\mathbf{b}}_{H_0}$ can be computed without knowing the true value of σ^2. As expected the returned betas satisfy the systems of constraints/linear hypotheses since:

$$\mathbf{K}\hat{\mathbf{b}}_{H_0} = \mathbf{K}\hat{\mathbf{b}} - \underbrace{\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\left[\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-1}}_{\mathbf{I}_q}\left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right) = \mathbf{K}\hat{\mathbf{b}} - \mathbf{K}\hat{\mathbf{b}} + \mathbf{t} = \mathbf{t}$$

3.2.2 Residuals of the constrained model

In order to have the expression to compute the LRT for gaussian models it is worth look at the residuals associated with the constrainted model. Basic matrix algebra show that:

$$\begin{aligned}\mathbf{e}_{H_0} &= \mathbf{y} - \mathbf{X}\hat{\mathbf{b}}_{H_0}\\ &= \mathbf{y} - \mathbf{X}\hat{\mathbf{b}}_{H_0} - \mathbf{X}\hat{\mathbf{b}} + \mathbf{X}\hat{\mathbf{b}}\\ &= \mathbf{y} - \mathbf{X}\hat{\mathbf{b}} + \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right) = \mathbf{e} + \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)\end{aligned}$$

Looking at the last we conclude that the residual of the constrainted model are equal to the residual of the unconstrainted plus something else (depending on

data and difference between constraint and unconstrainted estimates).
What if we compute the sum of the squared constrainted residuals? we have:

$$\begin{aligned}\mathbf{e}_{H_0}^\top \mathbf{e}_{H_0} &= \left[\mathbf{e} + \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)\right]^\top \left[\mathbf{e} + \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)\right] \\ &= \mathbf{e}^\top \mathbf{e} + \mathbf{e}^\top \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right) + \left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)^\top \mathbf{X}^\top \mathbf{e} + \left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)^\top \mathbf{X}^\top \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)\end{aligned}$$

so we end with four terms. Now with basic algebra we can show that in general:

$$\begin{aligned}\mathbf{X}^\top \mathbf{e} = \mathbf{X}^\top \left[\mathbf{y} - \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}\right] &= \mathbf{X}^\top \mathbf{y} - \underbrace{\mathbf{X}^\top \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}}_{\mathbf{I}_{p+1}} \mathbf{X}^\top \mathbf{y} \\ &= \mathbf{0}_{p+1}\end{aligned}$$

So coming back to the sum of square of constrainted residuals it simplifies to the sum of squares of unconstrainted residuals plus a quadratic function

$$\mathbf{e}_{H_0}^\top \mathbf{e}_{H_0} = \mathbf{e}^\top \mathbf{e} + \left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)^\top \mathbf{X}^\top \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)$$

Now if $\mathbf{X}$ has full column rank, then $\mathbf{X}^\top \mathbf{X}$ is positive definite and so if $\hat{\mathbf{b}} \neq \hat{\mathbf{b}}_{H_0}$:

$$\begin{aligned}&\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)^\top \mathbf{X}^\top \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right) > 0 \\ &\mathbf{e}_{H_0}^\top \mathbf{e}_{H_0} > \mathbf{e}^\top \mathbf{e}\end{aligned}$$

So when we introduce linear restriction we end up with a constrainted models where squared sum of residuals is always larger than unrestricted one (by introducing restriction we deteriorate the model). The amount of difference depends on the difference between constraited and undconsrtaited estimates and the matrix $\mathbf{X}$.
We can see what happens if we replace $\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}$ with what found before (in 3.1) that is:

$$\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{K}^\top \left[\mathbf{K}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{K}^\top\right]^{-1} \left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right)$$

Therefore the difference between the sum of squared residuals is:

$$\begin{aligned}&\mathbf{e}_{H_0}^\top \mathbf{e}_{H_0} - \mathbf{e}^\top \mathbf{e} \\ &= \left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right)^\top \mathbf{X}^\top \mathbf{X}\left(\hat{\mathbf{b}} - \hat{\mathbf{b}}_{H_0}\right) \\ &= \left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right)^\top \left[\mathbf{K}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{K}^\top\right]^{-1} \mathbf{K}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{K}^\top \left[\mathbf{K}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{K}^\top\right]^{-1} \left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right) \\ &= \left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right)^\top \left[\mathbf{K}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{K}^\top\right]^{-1} \left(\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right)\end{aligned}$$

In the end, to know how much the errors increase we do not actually need to fit the model under the restriction because $\hat{\mathbf{b}}_{H_0}$ is not in the final formula.
Thanks to this results we're able to compute the value of the lrt simply by starting from the unconstrainted ML estimate

3.3 Likelihood ratio properties

3.3.1 LRT statistics - 2

Developing a bit the loglikelihood version we have

$$\Delta l = 2\ln\left[\frac{L\left(\hat{\mathbf{b}},\sigma^2\right)}{L\left(\hat{\mathbf{b}}_{H_0},\sigma^2\right)}\right] = -n\ln 2\pi\sigma^2 - \frac{\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{b}}\right)^\top\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{b}}\right)}{\sigma^2} + n\ln 2\pi\sigma^2 + \frac{\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{b}}_{H_0}\right)^\top\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{b}}_{H_0}\right)}{\sigma^2}$$

$$= \frac{\mathbf{e}_{H_0}^\top\mathbf{e}_{H_0} - \mathbf{e}^\top\mathbf{e}}{\sigma^2} = \frac{\left(\mathbf{K}\hat{\mathbf{b}}-\mathbf{t}\right)^\top\left[\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-1}\left(\mathbf{K}\hat{\mathbf{b}}-\mathbf{t}\right)}{\sigma^2}$$

So it turns out it depends on the difference of sum of squared residuals and, as we've anticipated, it is not necessary to know $\hat{\mathbf{b}}_{H_0}$ in order to compute the LR test statistic and to derive its distribution.
Once computed $\hat{\mathbf{b}}$, once chosen $\mathbf{K}$ and $\mathbf{t}$, in order to compute LRT statistics we don't need the model under restriction (we can forget about lagrange multiplier)
In the quadratic form at the numerator the closer $K\hat{\mathbf{b}}-\mathbf{t}$ is to $\mathbf{0}$ (so the closer is $\hat{\mathbf{b}}$ to $\hat{\mathbf{b}}_{H_0}$) the smaller will be the value of the test statistics; on the contrary the larger the difference between $\hat{\mathbf{b}}$ and $\hat{\mathbf{b}}_{H_0}$ the larger will be the value of the test stastics.
So the closer the unconstrainted model is to the constrainted one the smaller will be the value of the test statistics

Important remark 7. per il prof importante la formula finale dell'LRT

Important remark 8. To do proper test, we need to know the distribution of the LRT.

3.3.2 LRT statistic distribution - σ^2 known

There are different way to come up with the distribution. One is the following: start by hypothesizing σ^2 is known.
By recalling properties of the maximum likelihood estimator for $\boldsymbol{\beta}$ we have that

$$\hat{\mathbf{B}} \sim MVN_{p+1}\left(\boldsymbol{\beta},\sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\right)$$

the variance is the inverse of the expected fisher information matrix. If we apply a linear transformation then

$$\mathbf{K}\hat{\mathbf{B}}-\mathbf{t} \sim MVN_q\left(\mathbf{K}\boldsymbol{\beta}-\mathbf{t},\sigma^2\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right)$$

If by hypothesis H_0 is true $\mathbf{K}\boldsymbol{\beta}=\mathbf{t}$ so $\mathbf{K}\boldsymbol{\beta}-\mathbf{t}=\mathbf{0}_q$ therefore

$$\mathbf{K}\hat{\mathbf{B}}-\mathbf{t}|H_0 \sim MVN_q\left(\mathbf{0}_q,\sigma^2\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right)$$

Applying the standardization for a MVN, by dividing for the square root of variance/covariance matrix, we end up with a standardized MVN

$$\boldsymbol{Z} = \frac{1}{\sigma}\left[\mathbf{K}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-\frac{1}{2}}\left(\mathbf{K}\hat{\mathbf{B}}-\mathbf{t}\right)|H_0 \sim MVN_q(\mathbf{0}_q,\mathbf{I}_q)$$

It turns out that the sum of squares of the vector $\mathbf{Z}$, that is $\boldsymbol{Z}^\top \boldsymbol{Z}$, is exactly the expression for the LRT statstics we found before, that is we end up with the previous quadratic form (sum of squares). But since we're taking the sum of q standardized independent gaussian rv, we obtain that are distributed as a χ^2 with q decgrees of freedom (element in $\mathbf{Z}$)

$$\boldsymbol{Z}^\top \boldsymbol{Z} = \frac{\left(\mathbf{K}\hat{\mathbf{B}} - \mathbf{t}\right)^\top \left[\mathbf{K}\left(\mathbf{X}^\top \mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-1} \left(\mathbf{K}\hat{\mathbf{B}} - \mathbf{t}\right)}{\sigma^2} = \Delta l | H_0 \sim \chi^2_q$$

If σ^2 were known than LRT would be $\sim \chi^2_q$ under null hypothesis. In the more realistic situation where σ^2 is unknown we replace it with an estimate. We now see what is the impact of this replacing in the distribution.

3.3.3 LRT statistic distribution - σ^2 unknown

We know that (*Proprieties of raw residuals*) the sum of raw residual squared over σ^2, $\dfrac{\mathbf{e}^\top \mathbf{e}}{\sigma^2}$:

- is chi squared distributed $\dfrac{\mathbf{e}^\top \mathbf{e}}{\sigma^2} \sim \chi^2_{n-p-1}$
- is independent between $\hat{\mathbf{B}}$ and S^2

we can in some sense replace the unknown σ^2 with an estimate based on the sum of the squares of residuals, and in doing so we end with a new test statistic representable as a ratio of independent chi-squares.
If H_0 is true

$$\begin{aligned}\frac{\Delta l}{\frac{\mathbf{e}^\top \mathbf{e}}{\sigma^2}} \frac{n-p-1}{q} &= \frac{\frac{\mathbf{e}^\top_{H_0}\mathbf{e}_{H_0} - \mathbf{e}^\top \mathbf{e}}{\sigma^2}}{\frac{\mathbf{e}^\top \mathbf{e}}{\sigma^2}} \frac{n-p-1}{q} \\ &= \frac{\frac{\left(\mathbf{K}\hat{\mathbf{B}}-\mathbf{t}\right)^\top \left[\mathbf{K}\left(\mathbf{X}^\top \mathbf{X}\right)^{-1}\mathbf{K}^\top\right]^{-1}\left(\mathbf{K}\hat{\mathbf{B}}-\mathbf{t}\right)}{q}}{\frac{\mathbf{e}^\top \mathbf{e}}{n-p-1}} | H_0 \sim \frac{\frac{\chi^2_q}{q}}{\frac{\chi^2_{n-p-1}}{n-p-1}} = F_{(q,n-p-1)}\end{aligned}$$

From a technical pov what we can do is divide the LRT and the sum of squared residuals by their degrees of freedom (lrt has q degrees of freedom, sum of squares of residuals over $sigma^2$ has $n-p-1$) and then divide the obtained quantity: by doing this we cancel out σ^2 and so we are left with the ratio of two independent chi square distributed statistics (at denominator we have the unbiased estimator of σ^2) divided by they degrees of freedom.
So here we have a test statistics which involves only known quantities (once we have our sample) and is distributed as an F with q and $n-p-1$ degrees of freedom.
This test statistics formally speaking is not the LRT (which is at the numerator) but basically we can compute it and we now its distribution under null so we can use it for inference

3.3.4 Applications

Comparison between complete and reduced models When linear hypotheses lead to the removal of q regressors, raw residuals $\mathbf{e}_{H_0}$ correspond to the residuals of a reduced model (nested in the complete model) and the previous general test becomes writable as

$$\frac{\Delta l}{\frac{\mathbf{e}^\top \mathbf{e}}{\sigma^2}} \frac{n-p-1}{q} = \frac{\frac{SSE_{M_{H_0}} - SSE_{M_C}}{q}}{\frac{SSE_{M_C}}{n-p-1}}$$

where

- SSE_{M_C} is the residual sum of squares for the complete model (with all regressors)
- $SSE_{M_{H_0}}$ is the residual sum of squares for the reduced model (after excluding q regressors)

Wald test statistics There is an interesting property of the LRT for hypotheses such $H_0 : \beta_j = 0$. Here we have that LRT takes a simplified expression, as ratio of the estimate of interest and the square root of its variance

$$\Delta l = \frac{\hat{B}_j^2}{\sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}_{jj}} = \left[\frac{\hat{B}_j}{\sigma \sqrt{(\mathbf{X}^\top \mathbf{X})^{-1}_{jj}}} \right]^2$$

where $\left(\mathbf{X}^\top \mathbf{X}\right)^{-1}_{jj}$ is the j-th element on the main diagonal of $\left(\mathbf{X}^\top \mathbf{X}\right)^{-1}$. In case:

- σ^2 *known* then $\frac{\hat{B}_j}{\sigma \sqrt{(\mathbf{X}^\top \mathbf{X})^{-1}_{jj}}} | H_0 \sim N(0,1)$
- σ^2 *unknown* we replace it with unbiased estimator we end up with the test statistics $\frac{\hat{B}_j}{S \sqrt{(\mathbf{X}^\top \mathbf{X})^{-1}_{jj}}} | H_0 \sim t_{n-p-1}$

3.4 Confidence intervals

Starting from properies of ML estimators we can come up with confidence intervals for a parameter

Considering the pivotal quantity for β_j: we know that ML estimator has a gaussian distribution we can standardize it by subtracting its expected value (being unbiased its the real beta) and divide by the square root of its variance will have a standard gaussian distribution

$$\frac{\hat{B}_j - \beta_j}{\sigma \sqrt{(\mathbf{X}^\top \mathbf{X})^{-1}_{jj}}} \sim N(0,1)$$

then we can end up with

- a gaussian intervals (if σ^2 known) at a $1-\alpha$ confidence level:

$$\left[\hat{b}_j - z_{\frac{\alpha}{2}}\sqrt{\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}_{jj}}, \hat{b}_j + z_{\frac{\alpha}{2}}\sqrt{\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}_{jj}}\right]$$

- a student-t intervals (if σ^2 unknown) at a $1-\alpha$ confidence level:

$$\left[\hat{b}_j - t_{\frac{\alpha}{2},n-p-1}\sqrt{s^2(\mathbf{X}^\top\mathbf{X})^{-1}_{jj}}, \hat{b}_j + t_{\frac{\alpha}{2},n-p-1}\sqrt{s^2(\mathbf{X}^\top\mathbf{X})^{-1}_{jj}}\right]$$

Important remark 9. We seen main tools for linear hypothesis testing (for betas, we could test variance as well but less interesting). Now we see some example of use of this tools, as well as the use of categorical regressors.

Chapter 4

Use of categorical regressors

4.1 Unordered categories

4.1.1 Motivating example

Example 4.1.1 (Glucose level in blood and ethnic origin)**.** A glucose level in blood between 100 and 125 mg/dL represents a risk factor for developing diabetes.
Aim: Are there systematic differences in the glucose level among people with different *ethnic origins*?
Available information: Glucose level and ethnic origin (White/African American/other) on a sample of 2020 women not affected by diabetes after menopause. Some basic info in graphs 4.1: in the sample most of the women are Caucasian. In terms of conditional distribution more or less the boxplox show similar variability with strong overlap, with some differences in location. The zoom on conditional means + CI highlight the differences in average glocose level (there's differences in the width due to differences in group sample sizes).
Hypothesis of interest: Absence of significant differences in the average glucose level among different ethnic groups. We want to check that the conditional expected value is the same for different groups:

$$H_0 : \mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{White}\right] = \mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{Other}\right] = \mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{African American}\right]$$
$$i = 1, \dots, 2020$$

Remark 6. Inferential tools we can adopt for the hypothesis of interest are:

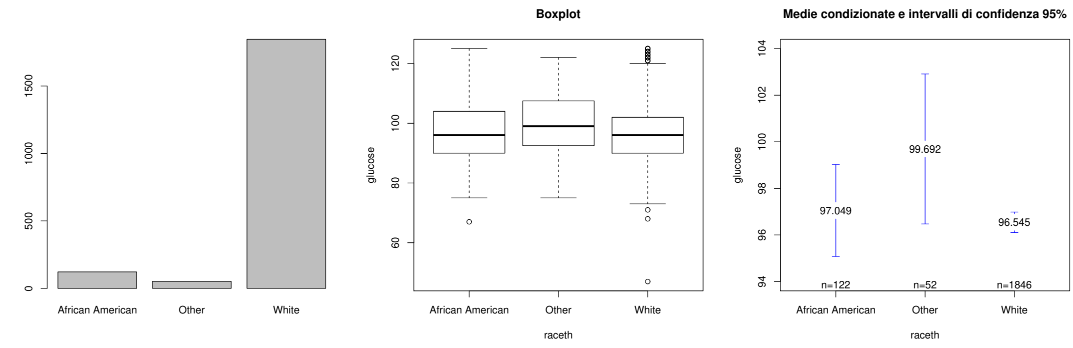
- One-way ANOVA
- Gaussian linear models with indicator/dummy variables

4.1.2 One-way ANOVA

```
> summary(aov(glucose ~ raceth,data=hers.nod))
              Df  Sum Sq  Mean Sq  F value  Pr(>F)
 raceth        2     521   260.51    2.747  0.0643
 Residuals  2017  191259    94.82
```

4.1.3 Linear regression with a qualitative regressor

First we have to do numeric coding of categorical regressor :



(a) (b) (c)

Figure 4.1: Glucose level and ethnic origins

- we can introduce 3 dummy variables, one for each category as follows (1 for the category considered, 0 for the others):

	x_{Ai}	x_{Oi}	x_{Wi}
	$\texttt{African American}_i$	$\texttt{Other}_i$	$\texttt{White}_i$
$\texttt{raceth}_i = \texttt{African American}$	1	0	0
$\texttt{raceth}_i = \texttt{Other}$	0	1	0
$\texttt{raceth}_i = \texttt{White}$	0	0	1

  So 3 indicator variables allow to code a qualitative regressor with 3 categories

- however it is necessary to consider the context of a multiple linear regression model. These 3 indicator variables sums up to 1, for any sample unit:

$$x_{Ai} + x_{Oi} + x_{Wi} = 1$$

  If they are included in a linear model along with an intercept term, the corresponding regressor matrix $\mathbf{X}$ will not have full column rank (being the intercept regressor a linear combination, simple sum of, the dummies introduced)

- what we can do is basically two thing:
  1. exclude one of the indicator variables: the corresponding category is termed *baseline/reference category*.
  2. exclude the intercept from the estimation and leave all the three dummies

4.1.3.1 Using a baseline category

For the first strategy suppose we exclude the african american dummy from the estimation, obtaining the model

$$\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i\right] = \beta_0 + \beta_1 \texttt{Other}_i + \beta_2 \texttt{White}_i$$

then:

$$\begin{aligned}
\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{African American}\right] &= \beta_0 \\
\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{Other}\right] &= \beta_0 + \beta_1 \\
\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{White}\right] &= \beta_0 + \beta_2
\end{aligned}$$

African american becomes the reference category since each regression coefficient (β_1, β_2) represents the difference between the conditional expected value given the corresponding category and the conditional expected value given the *reference category* (which is β_0).

In this context our hypothesis the absence of significant differences in the average glucose level among different ethnic groups

$$H_0 : \mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{White}\right] = \mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{Other}\right] = \mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{African American}\right]$$

can be translated in a system of linear hypotheses on parameters of the gaussian model like

$$H_0 : \begin{cases} \beta_0 = \beta_0 + \beta_1 \\ \beta_0 = \beta_0 + \beta_2 \\ (\beta_0 + \beta_1 = \beta_0 + \beta_2) \end{cases}$$

basic algebra leads to the equivalent $H_0 : \beta_1 = \beta_2 = 0$.
The results of the gaussian linear regression model are presented below

```
> summary(modello1)
Call:
lm(formula = glucose ~ raceth, data = hers.nod)
...

 Coefficients:
                Estimate  Std. Error  t value  Pr(>|t|)
 (Intercept)     97.0492      0.8816  110.081    <2e-16
 racethOther      2.6431      1.6127    1.639     0.101
 racethWhite     -0.5042      0.9103   -0.554     0.580

Residual standard error:  9.738 on 2017 degrees of freedom


Multiple R-squared:  0.002717, Adjusted R-squared:  0.001728
F-statistic:  2.747 on 2 and 2017 DF, p-value:  0.06434
```

The relevant test statistic:

- t test statistics allow to evaluate differences between each category and the reference category: the regression coefficients for the two indicator variables $\texttt{Other}_i$ and $\texttt{White}_i$ are not significantly different from 0;
- last row reports the F test statistic of our interest in this case (the linear independence hypothesis) which test that all the betas are 0. In this case we cannot refuse the null hypothesis so there's no evidence on effect of race on the dependent variable (despite being near the 0.05 treshold).
  The test can be reproduced using

```
> K1
    1  2  3
 1  0  1  0
 2  0  0  1

> t1

[1] 0 0

> linearHypothesis(modello1, K1, t1, test="F")
Linear hypothesis test

Hypothesis:

racethOther = 0
racethWhite = 0
Model 1:  restricted model

Model 2:  glucose ~ raceth
     Res.Df      RSS   Df   Sum of Sq        F    Pr(>F)
 1     2019   191780
 2     2017   191259    2      521.02   2.7473   0.06434
```

In this example:

- $\mathbf{e}_{H_0}^\top \mathbf{e}_{H_0} = 191780$ (sum of squared residuals of the restricted model)
- $\mathbf{e}^\top \mathbf{e} = 191259$ (sum of squared residuals of the unrestricted model),
- $q = 2$,
- $\mathbf{e}_{H_0}^\top \mathbf{e}_{H_0} - \mathbf{e}^\top \mathbf{e} = 521.02$
- the statistic is

$$\frac{\mathbf{e}_{H_0}^\top \mathbf{e}_{H_0} - \mathbf{e}^\top \mathbf{e}}{\mathbf{e}^\top \mathbf{e}} \frac{n-p-1}{q} = 2.743$$

- finally we see that the p-value coincides with the p-value reported in the model above

Choice of the reference category Regarding:

- the choice of the reference category is arbitrary
- the estimates for the regression coefficients will change, but the global measures remains the same

The default choice in R is the first category, in alphabetical order:

```
                  Other  White
 African American     0      0
            Other     1      0
            White     0      1
```

Instead if we use caucasian women as reference category:

```
                   1  2
 African American  1  0
            Other  0  1
            White  0  0
```

The meaning of the regression coefficients changes accordingly (we use different symbols δ to denote it):

$$\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i\right] = \delta_0 + \delta_1 \texttt{raceth1}_i + \delta_2 \texttt{raceth2}_i + \varepsilon_i, \quad i = 1, \ldots, 2020$$

with

$$\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{African American}\right] = \delta_0 + \delta_1$$
$$\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{Other}\right] = \delta_0 + \delta_2$$
$$\mathrm{E}\left[\texttt{glucose}_i | \texttt{raceth}_i = \texttt{White}\right] = \delta_0$$

The results of changing the categories are the following

```
> summary(modello2)
Call:
lm(formula = glucose ~ raceth, data = hers.nod)
...
 Coefficients:
                Estimate   Std. Error   t value   Pr(>|t|)
 (Intercept)     96.5450       0.2266   425.979     <2e-16
 raceth1          0.5042       0.9103     0.554     0.5797
 raceth2          3.1473       1.3693     2.299     0.0216

Residual standard error:  9.738 on 2017 degrees of freedom

Multiple R-squared:  0.002717, Adjusted R-squared:  0.001728
F-statistic:  2.747 on 2 and 2017 DF, p-value:  0.06434
```

So we've seen:

- the estimates for the regression coefficients (table above) has changed (intercept is the estimate of the expected value for the reference group which is changed, same for the others): we note the difference of raceth2 which was highlighted by changing the reference category.
  This can happen in real life: if we have a significant F and nonsignificant t maybe switching the reference category helps finding the difference
- the global measures (last paragraph below) remains the same regardless the reference category chosen

4.1.3.2 Exclusion of the intercept

If one consider a regression model without intercept, it is possible to include all the 3 indicator variables (without choosing a reference category). The model fitted will be (again different symbols)

$$\mathrm{E}\left[\texttt{glucose}_i|\texttt{raceth}_i\right] = \mu_1\texttt{African American}_i + \mu_2\texttt{Other}_i + \mu_3\texttt{White}_i \quad i = 1, \dots, 2020$$

and regarding the interpretation of coefficients

$$\begin{aligned}\mathrm{E}\left[\texttt{glucose}_i|\texttt{raceth}_i = \texttt{African American}\right] &= \mu_1\\ \mathrm{E}\left[\texttt{glucose}_i|\texttt{raceth}_i = \texttt{Other}\right] &= \mu_2\\ \mathrm{E}\left[\texttt{glucose}_i|\texttt{raceth}_i = \texttt{White}\right] &= \mu_3\end{aligned}$$

In this setup the hypothesis we're interested is is

$$H_0 : \begin{cases}\mu_1 = \mu_2\\ \mu_1 = \mu_3\end{cases}$$

or simply $H_0 : \mu_1 = \mu_2 = \mu_3$, so here we don't need to set anything equal to 0.
The estimation without intercept is done by putting `-1` in the formula:

```
> summary(modello3)
Call:
lm(formula = glucose ~ raceth - 1, data = hers.nod)
...
 Coefficients:
                          Estimate   Std.  Error   t value   Pr(>|t|)
 racethAfrican American    97.0492        0.8816    110.08     <2e-16
 racethOther               99.6923        1.3504     73.83     <2e-16
 racethWhite               96.5450        0.2266    425.98     <2e-16

Residual standard error:  9.738 on 2017 degrees of freedom


Multiple R-squared:  0.99, Adjusted R-squared:  0.99
F-statistic:  6.634e+04 on 3 and 2017 DF, p-value:  < 2.2e-16
```

There are some **WARNING** in removing intercept in R: the t test is against a null of beta to be 0 which in this case (and often) is non interesting.
Removing intercept messes up things especially in the summary part

- In this setting the function `lm` computes R^2 using $\sum_{i=1}^n y_i^2$ as denominator instead of total variability $\sum_{i=1}^n (y_i - \mu_y)^2$
- the F test statistic is referred to the null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3 = 0$ where the last $= 0$ is not of interest/meaningless in our case and is easily rejected
  To compute the proper test we can rely on the general approach as follows

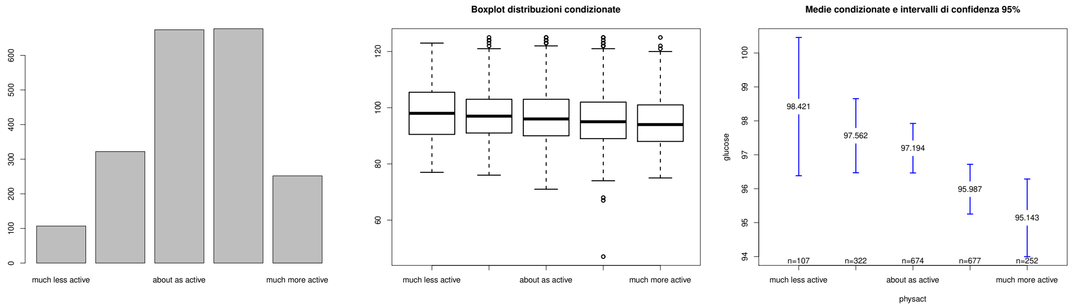
  ```
  > K3
      1   2   3
   1  1  -1   0
   2  1   0  -1

  > t3

  [1] 0 0
  ```

  With the previous K3 matrix we picked

  $$\begin{bmatrix}1 & -1 & 0\\ 1 & 0 & -1\end{bmatrix}$$

  in order to obtain

  $$\begin{cases}\mu_1 = \mu_2\\ \mu_1 = \mu_2\end{cases}$$



(a) (b) (c)

Figure 4.2: Glucose level and physical activity

By the way we would get exactly the same by setting either one of the following matrices

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

only changing the way the test is constructed not the results. eg the left matrix would be

$$\begin{cases} \mu_1 = \mu_2 \\ \mu_2 = \mu_3 \end{cases}$$

Going with the estimates we end up with the same results as the first model

```
> linearHypothesis(modello3,K3,t3,test="F")
Linear hypothesis test

Hypothesis:

racethAfrican American - racethOther = 0
racethAfrican American - racethWhite = 0
Model 1:  restricted model

Model 2:  glucose ~ raceth
     Res.Df     RSS   Df   Sum of Sq        F    Pr(>F)
 1     2019   191780
 2     2017   191259    2      521.02   2.7473   0.06434
```

4.2 Ordered Categories

4.2.1 Motivating example

Example 4.2.1 (Glucose level in blood and physical activity)**.** A glucose level in blood between 100 and 125 mg/dL represents a risk factor for developing diabetes.
Aim: Does physical activity (a modifiable factor related to life style) contribute to the reduction of the glucose level, thus preventing a severe disease?
Available information: Glucose level and physical activity level (`much less active`, `somewhat less active`, `about as active`, `somewhat more active`, `much more active`) on a sample of 2032 women not affected by diabetes after menopause.
The boxplots (fig 4.2) has more or less the same variability (strong overlap of distribution) and there's a decreasing trend of glucose mean level of as physical activity increases (maybe it's not that clinically relevant btw).

Remark 7. We can deal with this kind of data

- employing the same strategy of reference category seen for unordered data
- choosing a coding which acknowledge the ordering

4.2.2 Model with reference category

The estimated model with `much less active` as reference category

```
> physact1<-lm(glucose~physact,data=hers.nod)
> summary(physact1)
...
                               Estimate   Std. Error   t value   Pr(>|t|)
 (Intercept)                     98.421        0.939   104.784      0.000
 physactsomewhat less active     -0.858        1.084    -0.792      0.429
 physactabout as active          -1.226        1.011    -1.213      0.225
 physactsomewhat more active     -2.434        1.011    -2.408      0.016
 physactmuch more active         -3.278        1.121    -2.924      0.003
...

Residual standard error:  9.716 on 2027 degrees of freedom

Multiple R-squared:  0.008668, Adjusted R-squared:  0.006712
F-statistic:  4.431 on 4 and 2027 DF, p-value:  0.001441
```

There seems to be a significant impact of the physical activity on glucose level. Looking both at the coefficient and their P-values it seems to be a scaletta.
To reproduce the F test

```
> K1
    1  2  3  4  5
 1  0  1  0  0  0
 2  0  0  1  0  0
 3  0  0  0  1  0
 4  0  0  0  0  1

> t1

[1] 0 0 0 0


   > linearHypothesis(physact1,K1,t1,test="F")
Linear hypothesis test

Hypothesis:

physactsomewhat less active = 0
physactabout as active = 0
physactsomewhat more active = 0
physactmuch more active = 0
Model 1:  restricted model

Model 2:  glucose ~ physact
     Res.Df        RSS   Df   Sum of Sq      F   Pr(>F)
 1     2031  193017.70
 2     2027  191344.61    4     1673.09   4.43   0.0014
```

Important remark 10. linear models in statistics, rencher, consigliato per le proprieta varie dei modelli

4.2.3 Model with incremental/split coding

Before we used the same coding as used in the unordered categorical groups; an alternative coding scheme can be used if there is a "natural" order among the categories such as in this case

	x_{Bi}	x_{Ci}	x_{Di}	x_{Ei}
`much less active`	0	0	0	0
`somewhat less active`	1	0	0	0
`about as active`	1	1	0	0
`somewhat more active`	1	1	1	0
`much more active`	1	1	1	1

The number of dummy is still 4 to represent 5 categories, but they're defined differently (the first dummy take value 0 for the first category and 1 for the other and so on).

It is possible to show that these alternative indicator variables can be obtained by linear combination (summing subsets) of the indicator variables introduced above.
It's called split coding because implicitly we split categories into two subsets.
What happens with such coding? the model has still 5 parameters ...

$$\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i\right] = \beta_0 + \beta_B x_{Bi} + \beta_C x_{Ci} + \beta_D x_{Di} + \beta_E x_{Ei} i = 1, \ldots, 2032$$

but their interpretation changes in the sense that each regression coefficient represents the difference between the conditional expected values associated with two consecutive categories

$$\begin{aligned}
&\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{much less active}\right] = \beta_0\\
&\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{somewhat less active}\right] = \beta_0 + \beta_B\\
&\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{about as active}\right] = \beta_0 + \beta_B + \beta_C\\
&\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{somewhat more active}\right] = \beta_0 + \beta_B + \beta_C + \beta_D\\
&\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{much more active}\right] = \beta_0 + \beta_B + \beta_C + \beta_D + \beta_E
\end{aligned}$$

When it comes to estimates ...

```
> physact2 <-lm(glucose~physact,data=hers.nod)
> summary(physact2)
...
               Estimate   Std.  Error   t value   Pr(>|t|)
 (Intercept)     98.421          0.939  104.784      0.000
 physactd1       -0.858          1.084   -0.792      0.429
 physactd2       -0.368          0.658   -0.559      0.576
 physactd3       -1.208          0.529   -2.284      0.022
 physactd4       -0.844          0.717   -1.177      0.239
...
Residual standard error:  9.716 on 2027 degrees of freedom

Multiple R-squared:  0.008668, Adjusted R-squared:  0.006712
F-statistic:  4.431 on 4 and 2027 DF, p-value:  0.001441
```

In the interprestation:

- we have the same intercept as before (the expected value of the first category)
- the same happen for `physactd1` which is comparing the second group to the first one
- the remaining betas have different estimates because comparing to considered category to the previous one (instead of the first one).
- looking at the F test (all dummy variable = 0) we have the exact *same results* as the other coding scheme

In order to reproduce the F in the general context the matrix **K** is equal. The results will be the same as previously seen

```
> K2
     1   2   3   4   5
 1   0   1   0   0   0
 2   0   0   1   0   0
 3   0   0   0   1   0
 4   0   0   0   0   1

> t2

[1] 0 0 0 0

> linearHypothesis(physact2,K2,t2,test="F")
Linear hypothesis test

Hypothesis:

physactd1 = 0
physactd2 = 0
physactd3 = 0
physactd4 = 0
Model 1:  restricted model

Model 2:  glucose ~ physact
     Res.Df         RSS   Df   Sum of Sq      F   Pr(>F)
 1     2031   193017.70
 2     2027   191344.61    4     1673.09   4.43   0.0014
```

4.2.4 Linear trend hypothesis

What's the advantage of using the incremental coding? It matters if we're interested in some hypothesis in which the natural ordering of the categs is involved.
One of the typical hypothesis we could be interested is the so called linear trend hypothesis: it assumes that the change in conditional expected values given is constant, as we move from category to the next, no matter which consecutive couple of category we compare.
This is done by introduction of suitable linear constraints in the regression coefficients associated with the incremental coding scheme as

$$H_0 : \beta_B = \beta_C = \beta_D = \beta_E = \beta(\neq 0)$$

if $\beta > 0$ we'll have a constant increase in the conditional expected value or contrary for $\beta < 0$. By testing the hypothesis we check if we can replace the categories dummies with a numerical regressor taking the values 0 to 4 and by using a single coefficient β

$$\begin{aligned}
\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{much less active}\right]|\, H_0 &= \beta_0 + 0\cdot\beta = \beta_0\\
\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{somewhat less active}\right]|\, H_0 &= \beta_0 + 1\cdot\beta\\
\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{about as active}\right]|\, H_0 &= \beta_0 + 2\cdot\beta\\
\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{somewhat as active}\right]|\, H_0 &= \beta_0 + 3\cdot\beta\\
\mathrm{E}\left[\texttt{glucose}_i|\texttt{physact}_i = \texttt{much more active}\right]|\, H_0 &= \beta_0 + 4\cdot\beta
\end{aligned}$$

To implement this, after fitting the model, the linear constraints on coefficients in the general framework are (for four coefficient we need three equalities)

$$H_0 : \beta_B = \beta_C = \beta_D = \beta_E = \beta(\neq 0) \Longrightarrow H_0 : \begin{cases}\beta_B = \beta_C\\ \beta_C = \beta_D\\ \beta_D = \beta_E\end{cases}$$

which can be implemented as > `K.lin`

```
   1  2   3   4   5
1  0  1  -1   0   0
2  0  0   1  -1   0
3  0  0   0   1  -1
```

```
> t.lin

[1] 0 0 0
```

In the previous scheme we could have implemented three other constraints as well (obtainin same results) eg

$$\begin{cases}\beta_B = \beta_C\\ \beta_B = \beta_D\\ \beta_B = \beta_E\end{cases}$$

However In our case the results are as follows

```
> linearHypothesis(physact2,K.lin,t.lin,test="F")
Linear hypothesis test

Hypothesis:

physactd1 - physactd2 = 0
physactd2 - physactd3 = 0
physactd3 - physactd4 = 0
Model 1:  restricted model

Model 2:  glucose ~ physact
     Res.Df        RSS   Df   Sum of Sq      F   Pr(>F)
 1     2030  191419.47
 2     2027  191344.61    3       74.86   0.26   0.8511
```

So the linear trend hypothesis is not rejected (the fourth parameters -0.85, -0.36, -1.2, -0.84 are not significantly different from each other) and we can simplify the data by substituting numerical coding. There seems to be a constant difference in the expected value when we

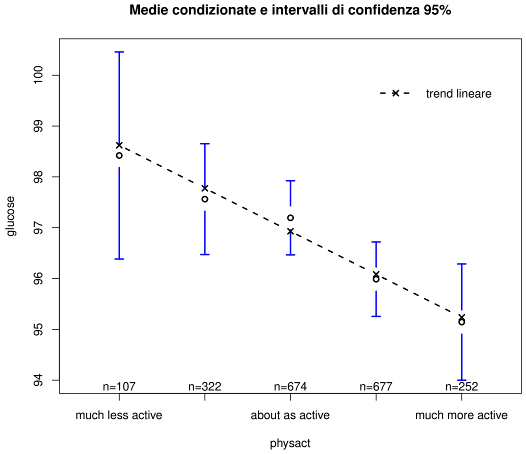


Figure 4.3: Estimated glucose conditional mean by activity level

compare consecutive pairs of groups.
The parameters of the constrained model can be estimated by coding the ordered categorical regressor using integer scores from 0 to 4 and by fitting a new Gaussian linear model

```
> hers.nod$physact.num<-as.numeric(hers.nod$physact)-1
> physact3<-lm(glucose physact.num,data=hers.nod)
> summary(physact3)
...
                Estimate   Std.  Error   t value   Pr(>|t|)
 (Intercept)      98.622          0.523   188.592      0.000
 physact.num      -0.847          0.206    -4.117      0.000
...

Residual standard error:  9.711 on 2030 degrees of freedom

Multiple R-squared:  0.00828, Adjusted R-squared:  0.007792
F-statistic:  16.95 on 1 and 2030 DF, p-value:  3.993e-05
```

As we can see the estimate -0.84 is basically a mean of the estimated coefficient of the model with categorical coding.
So the estimated conditional expected value is plotted in figure 4.3 where it overlap well with descriptive data.
We could get linear hypothesis test models using `anova` as well and comparing the two estimates (restricted and unrestricted models) - btw > `anova(physact3,physact2)`

```
Analysis of Variance Table

Model 1:  glucose ~ physact.num

Model 2:  glucose ~ physact
    Res.Df          RSS   Df   Sum of Sq      F    Pr(>F)
 1    2030    191419.47
 2    2027    191344.61    3       74.86   0.26    0.8511
```

So in general to obtain the test we need either:

- the starting model and the set of restrictions with `linearHypothesis`;
- the starting model and reduced model with `anova`.

If fitting the constrainted model is trivial (eg remove some regressors) we can go with `anova`, otoh more complex hypotheses/constrainted models can be tackled with `linearHypothesis`

Chapter 5

Models evaluation and comparison criteria

5.1 (Residual) deviance of a Gaussian linear model

5.1.1 Saturated models

We start introducing the concept of saturated model: we have our sample, model and its assumption therefore

$$M : \mathbf{Y}|\mathbf{X} \sim MVN_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$$

To this model we can associate a saturated model.

Definition 5.1.1 (Saturated model)**.** The saturated model for M is a model with a *number of parameters* for the expected values that is equal to the *number of unique covariate patterns* in the matrix $\mathbf{X}$ (equal to the unique values for $\mathbf{x}_i^\top \boldsymbol{\beta}$).

Remark 8. If there are (many) numerical regressors it may be that each row have a unique covariate pattern; this is quite common in observational studies. Here we will focus on this situation.

Important remark 11. If the number of unique covariate patterns is equal to n (*each sample unit is characterised by a specific combination of regressor values*), then the saturated model can be defined as follows:

$$M_{sat} : \mathbf{Y}|\mathbf{X} \sim MVN_n(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_n)$$

with $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)^\top \in \mathbb{R}^n$.
So instead of $\mathbf{X}\boldsymbol{\beta}$ for each unit we'll have a specific parameter μ_i associated with the expected value for Y (associated to that specific covariate pattern) without any explicit math/functional relationship with $\mathbf{X}$ (we're implicitly assuming characterized by different covariate pattern will have different expected value: in some sense there's still a functional kind of relationship). In a sense is the model with the highest possible flexibility in describing the relationship between X and Y

5.1.2 Maximum likelihood estimation of $\mu_1, \mu_2, \dots, \mu_n$

Trying to fit the saturated model

- its log-likelihood function is the same except that rather than having a likelihood function depending on $p + 1$ parameters (betas), we have one depending on n (μ_i), one for each unit

$$l(\mu_1, \dots, \mu_n, \sigma^2) = -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln \sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu_i)^2$$

- if we try to fit it, maximum likelihood estimate of μ_i will be obtained having

$$\left.\begin{aligned} \frac{\partial}{\partial \mu_i} l(\mu_1, \ldots, \mu_n, \sigma^2) &= \frac{y_i - \mu_i}{\sigma^2} \\ \frac{\partial^2}{\partial \mu_i^2} l(\mu_1, \ldots, \mu_n, \sigma^2) &= -\frac{1}{\sigma^2} \end{aligned}\right\} \quad \Rightarrow \quad \hat{m}_i = y_i \quad i = 1, \ldots, n$$

So we'll have n first partial derivatives which are very simple and depends only on one parameter. Taking the second partial derivative with respect to all parameters (Hessian) will be different from zero only for the same parameter, the results will be on the diagonal of the Hessian and will be constant (outside the diagonal the hessian has null entries so its diagonal)
The condition will lead to having as coefficient basically the observed value of y_i: because first partial derivatives equated to 0 leads there and hessian is negative definite matrix (all eigenvalue, elements of the diagonal in the diagonal matrix, are all negative).
So being $\mu_i = y_i$ the sum of the residuals of this model will be all $= 0$ because fitted value from the model coincide with observed value

So:

- the loglikelihood computed at its maximum will be the quantity

$$l(\hat{m}_1, \ldots, \hat{m}_n, \sigma^2) = -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2$$

This does not depend on μ, only on σ^2: this quantity can be interpreted as the maximum possible value for the log-likelihood associated to Gaussian models for $\mathbf{Y}|\mathbf{X}$, given the observed sample $\mathbf{y}$

- any Gaussian linear model for $\mathbf{Y}|\mathbf{X}$ will have a maximum value for the log-likelihood that is smaller than the previous quantity, given the observed sample $\mathbf{y}$.
Any model (by imposing linear restriction on the expected values) will have a lower ML, since a part the quantity we have $-sumofsquareofresiduals$

Important remark 12. So why bother: the problem with saturated model is that we don't have a function defining the association between regressors and Y which is of primary interest for understanding how each regression impact on y (here the saturated model is useless).

Important remark 13. The saturated model has to be taken as benchmark/best model (in terms of loglikelihood) for certain data

5.1.3 Comparisons with the saturated model

Any Gaussian linear model for $\mathbf{Y}|\mathbf{X}$ can be seen as a model that introduces some constraints on the parameters of the saturated model. These constraints can be expressed through a linear system:

$$\left.\begin{aligned} &M_{sat} : \mathbf{Y}|\mathbf{X} \sim MVN_n(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_n) \\ &H_0 : \boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}, \quad \boldsymbol{\beta} \in \mathbb{R}^{p+1} \end{aligned}\right\} \quad \Rightarrow \quad M : \mathbf{Y}|\mathbf{X} \sim MVN_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$$

At least theoretically, we can use the LRT statistics to quantify the distance between our model and the saturated model (that is the adequacy of the constraints in our model).
This comparison can be done theoretically: it cannot be done from a practical POV. To understand why lets define a measure

Definition 5.1.2 ((Residual) deviance of a Gaussian linear model)**.** It's twice the difference in the log likelihood ratio between the likelihood of the saturated model and the likelihood of the model at hand/considered:

$$\begin{aligned} D = 2 \ln \left[\frac{L(\hat{m}_1, \ldots, \hat{m}_n, \sigma^2)}{L(\hat{\mathbf{b}}, \sigma^2)} \right] &= 2 \left[l(\hat{m}_1, \ldots, \hat{m}_n, \sigma^2) - l(\hat{\mathbf{b}}, \sigma^2) \right] \\ = 2 \left[-\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 + \frac{n}{2} \ln 2\pi + \frac{n}{2} \ln \sigma^2 + \frac{\mathbf{e}^\top \mathbf{e}}{2\sigma^2} \right] &= \frac{\mathbf{e}^\top \mathbf{e}}{\sigma^2} \end{aligned}$$

This difference is basically sum of squared residuals of the considered model divided by σ^2.

Remark 9. Some authors/software use the expression “residual deviance” to denote $\mathbf{e}^\top\mathbf{e}$, and the expression “scaled deviance" to denote D

Important remark 14. Note that:

- we know that in principle if the fitted model is adequated for the data/close to the saturated model (the null hypothesis introducing the linearity in the expected value is adequate) thanks to the property of sum of squares of residuals, we can say that the deviance has a chi square distribution

  $$\left.\frac{\mathbf{e}^\top\mathbf{e}}{\sigma^2}\right| H_0 : \boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta} \sim \chi^2_{n-p-1}$$

  problem is that deviance depends on σ^2. When discussing use LRT previously, we circumvented the use of σ^2 by replacing it with an estimate; we have two way to obtain an estimate of σ^2 here:

  - from the saturated model and from the other: with the saturated model the sum of squares of residual is 0, the implicit estimate from the saturated model is 0, and so the deviance ratio/test statistics goes to $+\infty$
  - from the restricted/fitted model where the estimate is $\dfrac{\mathbf{e}^\top\mathbf{e}}{n-p-1}$ so putting in D will make it results $D = n - p - 1$

  Clearly both cases are not useful (one always infinite the other always constant); generally speaking we cannot exploit the deviance to perform test on the considered model

- however we can take $\mathbf{e}^\top\mathbf{e}$ as a measure telling us how a fitted model is close to the best possible model: the smaller the closer the model is to the best one

Remark 10. In some models we’ll be able to exploit residual deviance to perform goodness of fit test

Remark 11. We wont spend time where the number of covariate pattern is less than the number of units; in those cases we’re able to exploit deviance to performe goodness of fit test. However these are quite rather rare/the exception

5.1.4 R^2 coefficient

Let’s put aside the idea of using a test statistics to check if the model is adequate or not, lets make the most of the results shown before. We’ve seen the larger the $\mathbf{e}^\top\mathbf{e}$ the larger the deviance so at least we can use $\mathbf{e}^\top\mathbf{e}$ to perform some sort of subjective evaluation of the goodnes of fit.

We use the well known R^2 coefficient which is just a normalized version of sum of square of residuals and can be interpreted as the fraction of variability of y which can be explained by the fitted model:

$$R^2 = 1 - \frac{\mathbf{e}^\top\mathbf{e}}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Important remark 15. It is possible to prove that

$$0 \leq \mathbf{e}^\top\mathbf{e} \leq \sum_{i=1}^n (y_i - \bar{y})^2$$

Therefore:

- $R^2 \in [0, 1]$
- $R^2 = 1$ if and only if $\mathbf{e}^\top\mathbf{e} = 0$
  In this case the Gaussian linear model M is “equivalent” to the corresponding saturated model
- $R^2 = 0$ if and only if $\mathbf{e}^\top\mathbf{e} = \sum_{i=1}^n (y_i - \bar{y})^2$ (where sum of squares of residuals coincides with sample deviance, which happens if all the fitted values are all equal among each other and all equal to the sample mean).
  In this case the Gaussian linear model M is “equivalent” to the Gaussian linear model that assumes linear independence of $\mathbf{Y}$ from all regressors

5.2 Comparisons among Gaussian linear models

Now we switch the focus from evaluating a single model to choosing the most adequate Gaussian linear model for a given random sample $\mathbf{Y}$ (among 2+ of them).

5.2.1 Choice among two Gaussian linear models

Simplest situation: two candidate models differing by different sets of regressors: $\mathbf{X}_A \neq \mathbf{X}_B$

$$M_A : \mathbf{Y}|\mathbf{X}_A \sim MVN_n\big(\mathbf{X}_A\boldsymbol{\beta}_A, \sigma^2\mathbf{I}_n\big), \quad \boldsymbol{\beta}_A \in \mathbb{R}^{p_A+1}$$
$$M_B : \mathbf{Y}|\mathbf{X}_B \sim MVN_n\big(\mathbf{X}_B\boldsymbol{\beta}_B, \sigma^2\mathbf{I}_n\big), \quad \boldsymbol{\beta}_B \in \mathbb{R}^{p_B+1}$$

Here without loss of generality here we assume the first has more columns in the regressor matrix than the second: $p_A > p_B$.
We can distinguish two main situation: we can compare nested or nonnested models.

5.2.1.1 Nested models and LRT

Definition 5.2.1 (Nested model)**.** Model M_B is nested in model M_A when matrix $\mathbf{X}_B$ is obtained by removing one or more columns from matrix $\mathbf{X}_A$

Important remark 16. M_B can be obtained by introducing suitable linear constraints on the parameteres of M_A (setting some of them equal to 0):

$$\left.\begin{aligned} &M_A : \mathbf{Y}|\mathbf{X}_A \sim MVN_n\big(\mathbf{X}_A\boldsymbol{\beta}_A, \sigma^2\mathbf{I}_n\big) \\ &H_0 : \mathbf{K}_B\boldsymbol{\beta}_A = \mathbf{t}_B \end{aligned}\right\} \quad \Rightarrow \quad M_B : \mathbf{Y}|\mathbf{X}_B \sim MVN_n\big(\mathbf{X}\boldsymbol{\beta}_B, \sigma^2\mathbf{I}_n\big)$$

The transformation is the following

- $\mathbf{K}_B$: $(q) \times (p_A + 1)$ matrix each row of this matrix contains a 1 in a specific position (corresponding to one of the q regressors excluded from M_A), and 0 elsewhere
- $\mathbf{t}_B = \mathbf{0}_q$

The number of regressors excluded from M_A to obtain M_B will be

$$q = p_A - p_B$$

Important remark 17. A likelihood ratio test can be used to choose among M_A and M_B; in particular, such test ends up in a difference of the two corresponding (scaled) deviances:

$$\begin{aligned} \Delta l = 2\ln\left[\frac{L\Big(\hat{\mathbf{b}}_A, \sigma^2\Big)}{L\Big(\hat{\mathbf{b}}_{A|H_0}, \sigma^2\Big)}\right] &= 2\ln\left[\frac{L\Big(\hat{\mathbf{b}}_A, \sigma^2\Big)}{L\Big(\hat{\mathbf{b}}_B, \sigma^2\Big)}\right] = \frac{\mathbf{e}_B^\top\mathbf{e}_B - \mathbf{e}_A^\top\mathbf{e}_A}{\sigma^2} \\ &= D(M_B) - D(M_A) = \Delta D \end{aligned}$$

If H_0 is true - if M_B is as "adequate" as M_A then:

- $\Delta D|\, M_B \sim \chi^2_q$
- $\left.\dfrac{\Delta D}{D_{M_A}} \dfrac{n - p_A - 1}{q}\right| M_B \sim F_{(q, n-p_A-1)}$

If the null is not rejected we will stick with the reduced model; it rejected we'll select the complete model.

5.2.1.2 Non-nested models and adjusted R^2

Remark 12. We cannot exploit the LRT when we 're dealing with two non-nested models.

Definition 5.2.2 (Non nested model)**.** The two models are characterised by two sets of regressors that are only partially overlapping, or non-overlapping.
Model M_B can be obtained by both excluding some (or all) regressors in model M_A and adding some regressors to model M_A

Important remark 18. The differences between the two deviances (used for LRT statistic) does not have a known random distribution, and thus a likelihood ratio test cannot be used to choose between the two models.

Important remark 19. Therefore we must rely on different criteria for selecting models.
In the literature we have plenty method of model selection criteria, that is quantity that can be computed for all the models involved in comparison procedure to chose the best one.
Different criteria are obtained by choosing the definition of best/what to optimize.

Adjusted R^2

Remark 13. It's one of the most common measure used in gaussian lm, and is obtained by introducting a slight multiplicative factor. $\dfrac{n-1}{n-p-1}$.

Definition 5.2.3.

$$R^2_{adj} = 1 - \left(\frac{\mathbf{e}^\top \mathbf{e}}{\sum_{i=1}^n (y_i - \bar{y})^2} \cdot \frac{n-1}{n-p-1} \right)$$

Important remark 20 (Problem with the classic one). It is possible to prove that $\mathbf{e}^\top \mathbf{e}$ never increase after adding a regressor to a Gaussian linear model *even if the regressor is irrelevant - see the part regarding linear hypotheses*).
So if M_A andM_B have different numbers of regressors, the use of R^2 could favour the model with the largest number of regressors

Important remark 21. With the adjusted method when p (number of of columns) in regressor matrix increases:

- the left factor $\dfrac{\mathbf{e}^\top \mathbf{e}}{\sum_{i=1}^n (y_i - \bar{y})^2}$ goes down as usual
- the factor $\dfrac{n-1}{n-p-1}$ become larger and counterbalance the first factor behaviour
- therefore *a reduction in* $\mathbf{e}^\top \mathbf{e}$ *due to the introduction of an irrelevant regressor can be balanced out by the corresponding increase in* $\dfrac{n-1}{n-p-1}$

We have that:

- differently from R^2, R^2_{adj} is not affected by the effect of the number of regressors on $\mathbf{e}^\top \mathbf{e}$ and can actually decrease as the number of regressors in the model increases
- the best model is still the one achieving the **maximum value for** R^2_{adj} (among all the considered models)
- the range for R^2_{adj} is slightly difference from R^2: indeed when $R^2 = 1$, the $R^2_{adj} = 1$ as well, but when $R^2 = 0$, we have that $\dfrac{\mathbf{e}^\top \mathbf{e}}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1$ and $R^2_{adj} = 1 - \dfrac{n-1}{n-p-1} \leq 0$.
So the minimum value of R^2_{adj} depends on the number of parameters and it decreases as p increasese; can be negative as well, since $\dfrac{n-1}{n-p-1} \geq 1$, while the traditional R^2 cannot be negative.

Remark 14. What happens when we compare models with the same number of parameters? Consider two models M_A and M_B such that $p_A = p_B = p$. Then under the hypothesis $R^2_{adj}(M_A)$ is better ...

$$\begin{aligned}
R^2_{adj}(M_A) > R^2_{adj}(M_B) &\iff 1 - \left(\frac{\mathbf{e}_A^\top \mathbf{e}_A}{\sum_{i=1}^n (y_i - \bar{y})^2} \cdot \frac{n-1}{n-p-1} \right) > 1 - \left(\frac{\mathbf{e}_B^\top \mathbf{e}_B}{\sum_{i=1}^n (y_i - \bar{y})^2} \cdot \frac{n-1}{n-p-1} \right) \\
&\iff \frac{\mathbf{e}_A^\top \mathbf{e}_A}{\sum_{i=1}^n (y_i - \bar{y})^2} \cdot \frac{n-1}{n-p-1} < \frac{\mathbf{e}_B^\top \mathbf{e}_B}{\sum_{i=1}^n (y_i - \bar{y})^2} \cdot \frac{n-1}{n-p-1} \\
&\iff \mathbf{e}_A^\top \mathbf{e}_A < \mathbf{e}_B^\top \mathbf{e}_B \\
&\iff R^2(M_A) > R^2(M_B)
\end{aligned}$$

...we end up with knowing that $R^2(M_A)$ will be better as well. So the conclusion we'll draw will coincides between the two

5.2.2 Other comparison methods

5.2.2.1 Prediction error and Leave-One-Out Cross-Validation

Remark 15. This is another tool which optimize another quantity: if we want to use our model to make prediction (rather than studying which regressors have impact on dependent variable) we may want to focus more directly on its performance in the task.
Basic idea: between two regression models, choose the one with the *smallest prediction error.*
One way to estimate the prediction error is to do LOOCV

Important remark 22. In order to compute LOOCV for a given regression model, the estimation procedure should be repeated n times after omitting each sample unit. Then, each fitted model is used to compute a prediction for the corresponding omitted sample unit

Definition 5.2.4. By defining $\hat{m}_i^{[-i]}$ as the estimate of $\mathrm{E}\left[Y_i|\mathbf{x}_i\right]$ obtained after excluding the i-th unit from the observed sample (*independent of the i-th unit*) we get that

$$LOOCV = \frac{\sum_{i=1}^{n}\left(y_i - \hat{m}_i^{[-i]}\right)^2}{n}$$

It is possible to prove that this quantity is an *unbiased estimate of the prediction error*

Remark 16. Some general remarks:

- the quantities $y_i - \hat{m}_i^{[-i]}$ are also referred to as *deleted residuals*
- some authors/softwares use the acronym *PRESS* (PRedictive Error Sum of Square) to denote *LOOCV*
- this is a general procedure we can use with any technique applied to do the prediction (eg other models/methods as well): this criterion is very used on non-parametric regression techniques because it doesnt rely on strong assumption of distribution of Y given X

Important remark 23 (Functioning). We have that:

- Differently from $\mathbf{e}^\top\mathbf{e}$, *LOOCV* may increase if an irrelevant regressor is added to the model
- The best model is the one achieving the **minimum value for** *LOOCV* (among all the considered models, even with different number of parameters)

Important remark 24 (*LOOCV* for Gaussian linear regression models). In case of gaussian linear regression models there is an interesting properties: LOOCV *can be computed without repeating the fitting process n times.*
By defining:

- $\hat{\mathbf{b}}^{[-i]}$ as the ML estimate of $\boldsymbol{\beta}$ obtained after excluding the i-th unit from the observed sample (*independent of the i-th unit*)
- $\hat{m}_i^{[-i]} = \mathbf{x}_i^\top\hat{\mathbf{b}}^{[-i]}$ as estimate of $\mathrm{E}\left[Y_i|\mathbf{x}_i\right]$ obtained after excluding the i-th unit from the observed sample (*independent of the i-th unit*)

It is possible to prove that the deleted residual for the single unit

$$y_i - \hat{m}_i^{[-i]} = y_i - \mathbf{x}_i^\top\hat{\mathbf{b}}^{[-i]} = \frac{y_i - \mathbf{x}_i^\top\hat{\mathbf{b}}}{1-\mathbf{H}_{ii}} = \frac{e_i}{1-\mathbf{H}_{ii}}$$

can be obtained by dividing the raw residual by 1 - the i-th diagonal element of the Hat matrix

5.2.2.2 Akaike information criterion (AIC)

Remark 17. There are model selection criteria which requires the specification of a parametric statistical model:

- the AIC, which is a general purpose criteria to perform model selection
- BIC

Important remark 25. Letting:

- $\mathbf{Y}$ be random sample with unknown probability/density function $f_0(\mathbf{y})$
- $\mathcal{F}_A = \left\{f_A(\cdot;\boldsymbol{\theta}_A), \boldsymbol{\theta}_A \in \boldsymbol{\Theta_A} \subseteq \mathbb{R}^{k_A}\right\}$ parametric statistical model A

- $\mathcal{F}_B = \left\{f_B(\cdot;\boldsymbol{\theta}_B), \boldsymbol{\theta}_B \in \boldsymbol{\Theta_B} \subseteq \mathbb{R}^{k_B}\right\}$ the parametric statistical model B

The idea behind AIC: *between two statistical models, choose the one that contains the element that is the most "similar" to $f_0(\cdot)$".*
One way of quantify the similarity is the so called Kullback-Leibler divergence:

$$\mathcal{K}(f_A, f_0) = \mathrm{E}\left[\ln \frac{f_0(\mathbf{Y})}{f_A(\mathbf{Y};\boldsymbol{\theta}_A)}\right]$$

This expected value is computed with respect to f_0: we can think about if as the *amount of information that is lost when $f_0(\cdot)$ is approximated with $f_A(\cdot;\boldsymbol{\theta}_A) \in \mathcal{F}_A$.*
So we should compute this for model A and B and look which is the smallest possible value for this quantity and then choose the corrisponding model.
It seems to be a quite complicated task to perform: interestingly Akaike proved that under suitable regularity conditions,

$$\min_{\mathcal{F}_A} \mathcal{K}(f_A, f_0) < \min_{\mathcal{F}_B} \mathcal{K}(f_B, f_0) \iff \underbrace{-2\ln L_A\left(\hat{\boldsymbol{\theta}}_A\right) + 2k_A}_{AIC(M_A)} < \underbrace{-2\ln L_B\left(\hat{\boldsymbol{\theta}}_B\right) + 2k_B}_{AIC(M_B)}$$

Definition 5.2.5 (AIC)**.** In general, for a parametric statistical model:

$$AIC = -2\ln L\left(\hat{\boldsymbol{\theta}}\right) + 2k$$

where $L\left(\hat{\boldsymbol{\theta}}\right)$ is the maximized likelihood of the model and k is the number of parameter to be estimated in the statistical model

Example 5.2.1. So in a linear model $y = \beta_0 + \beta_1 x$ the parameters are 3: β_0, β_1 and σ^2

Important remark 26. Regarding the two components:

- $-2\ln L\left(\hat{\boldsymbol{\theta}}\right)$ measures the goodness of fit of a statistical model to the data: *in general, this quantity decreases as the number of parameters increases*
- $2k$ measures the complexity of a statistical model: *it increases as the number of the parameters increases*

So the best model is the one achieving the **minimum value for** *AIC* (among all the considered models) *best trade-off between goodness of fit and complexity.*
The two components acts differently regarding the number of parameters 5.1: the loglikelihood part tend to decrease as model increases because adding flexibility so the model can go closer and closer to the data (think the saturated model, with the largest possible loglikelihood and the smaller possible -2loglik)

Important remark 27 (AIC and gaussian models). In the specific case of Gaussian linear models we have that maximum likelihood estimates of $\boldsymbol{\beta}$ and σ^2 are considered to compute

$$\begin{aligned}-2\ln L\left(\hat{\mathbf{b}}, \hat{s}^2\right) &= -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\frac{\mathbf{e}^\top\mathbf{e}}{n} - \frac{1}{2\frac{\mathbf{e}^\top\mathbf{e}}{n}}\mathbf{e}^\top\mathbf{e} \\ &= n\ln 2\pi + n + n\ln\frac{\mathbf{e}^\top\mathbf{e}}{n}\end{aligned}$$

and so

$$AIC = n\ln 2\pi + n + n\ln\frac{\mathbf{e}^\top\mathbf{e}}{n} + 2(p+2)$$

Here above $k = p + 1$ (columns + intercept) so total parameter are $p + 2$ (+sigma)
Finally, on of the advantages of AIC is that we could use to compare gaussian model to other types of models (with different distributional assumptions for Y | X); however when all the candidate models are Gaussian linear models, the formula above can be further simplified to:

$$AIC = n\ln\frac{\mathbf{e}^\top\mathbf{e}}{n} + 2(p+2)$$

We can ignore the first part $n\ln 2\pi + n$ (which is constant for all the models considered)

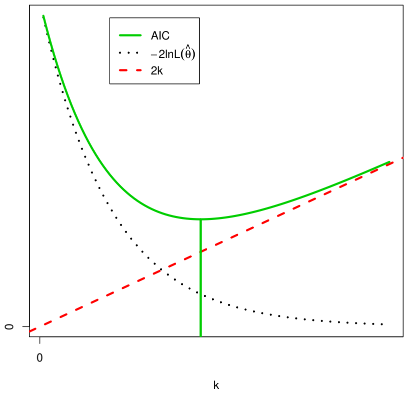


Figure 5.1: AIC components

5.2.2.3 (Schwartz) Bayesian information criterion (BIC)

Another general purpose tool is this one. The general setting is the same as AIC; considering

- $\mathbf{y}$ observed sample
- $\mathcal{F}_A$ parametric statistical model A
- $\mathcal{F}_B$ parametric statistical model B

In the Bayesian framework, each element in $\mathcal{F}_A$ and in $\mathcal{F}_B$ has an a priori probability of being the true distribution that generated $\mathbf{y}$:

- $g_A(\boldsymbol{\theta}_A)$ *a priori* probability/density function for distributions belonging to $\mathcal{F}_A$;
- $g_B(\boldsymbol{\theta}_B)$ *a priori* probability/density function for distributions belonging to $\mathcal{F}_B$.

Basic idea: *Between two statistical models, choose the one characterised by the the highest probability of having generated the observed sample*, which is computed with the following integral (for the first model)

$$\Pr(\mathbf{y}|\,\mathcal{F}_A) = \int g_A(\boldsymbol{\theta}_A) f_A(\mathbf{y};\boldsymbol{\theta}_A) d\boldsymbol{\theta}_A$$

This would require to specify: a priori distribution on the parameters, the computation of this integral. Neither of the two task is trivial.
Howevere, luckily, Schwartz proved that, under suitable regularity conditions,

$$\Pr(\mathbf{y}|\,\mathcal{F}_A) > \Pr(\mathbf{y}|\,\mathcal{F}_B) \iff \underbrace{-2\ln L_A\left(\hat{\boldsymbol{\theta}}_A\right) + \ln(n)k_A}_{BIC(M_A)} < \underbrace{-2\ln L_B\left(\hat{\boldsymbol{\theta}}_B\right) + \ln(n)k_B}_{BIC(M_B)}$$

Definition 5.2.6. For a generic parametric statistical model:

$$BIC = -2\ln L\left(\hat{\boldsymbol{\theta}}\right) + \ln(n)k$$

if we look at the expression it's similar to AIC, despite being obtained from two completely different perspectives.
We have

- the same term $-2\ln L\left(\hat{\boldsymbol{\theta}}\right)$ measures the goodness of fit of a statistical model to the data *in general, this quantity decreases as the number of parameters increases*
- the term $\ln(n)k$ measures the complexity of a statistical model *it increases as the number of the parameters increases*

Important remark 28. Again the best model is the one achieving the **minimum value for** *BIC* (among all the considered models) *best trade-off between goodness of fit and complexity* 5.2

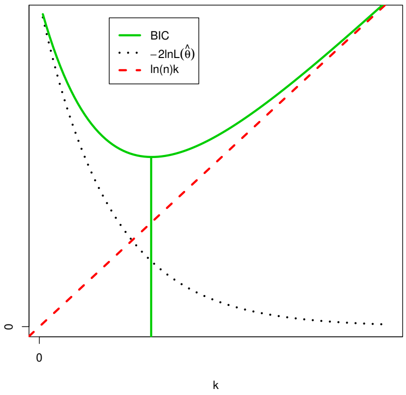


Figure 5.2: BIC components

Finally in case of for Gaussian linear models we have the specific formula where maximum likelihood estimates of $\boldsymbol{\beta}$ and σ^2 are considered

$$-2\ln L\left(\hat{\mathbf{b}}, \hat{s}^2\right) = n\ln 2\pi + n + n\ln\frac{\mathbf{e}^\top\mathbf{e}}{n}$$
$$BIC = n\ln 2\pi + n + n\ln\frac{\mathbf{e}^\top\mathbf{e}}{n} + \ln(n)(p+2)$$

And when all the candidate models are Gaussian linear models, the following simplified formula can be used:

$$BIC = n\ln\frac{\mathbf{e}^\top\mathbf{e}}{n} + \ln(n)(p+2)$$

5.2.2.4 *AIC* or *BIC*

Although derived within two completely different framework, *AIC* and *BIC* have a very similar functional form. The main "practical" difference is the way in which model complexity is weighted

- in general, *BIC puts more weight on model complexity* when we have more than 8 units, since $n > 8 \implies ln(n) > 2$
- for a given observed sample, *BIC* tends to favour less complex models (than those selected according to *AIC*)
- under suitable conditions, both criteria are consistent (when the sample size is large, they select the "best" model - according to the corresponding conceptual framework)
- there is not any test to evaluate the significance of the difference among *AIC* (or *BIC*) values

In fig **??** graphical comparison which shows why BIC tend to select more simple models than AIC

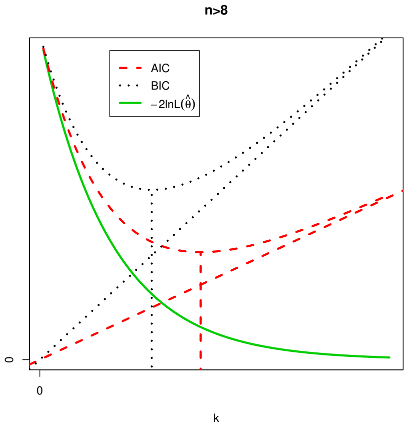


Figure 5.3: AIC and BIC components

Chapter 6

Lab1

We need package **car** to be installed, the dataset used is one of birtweight:

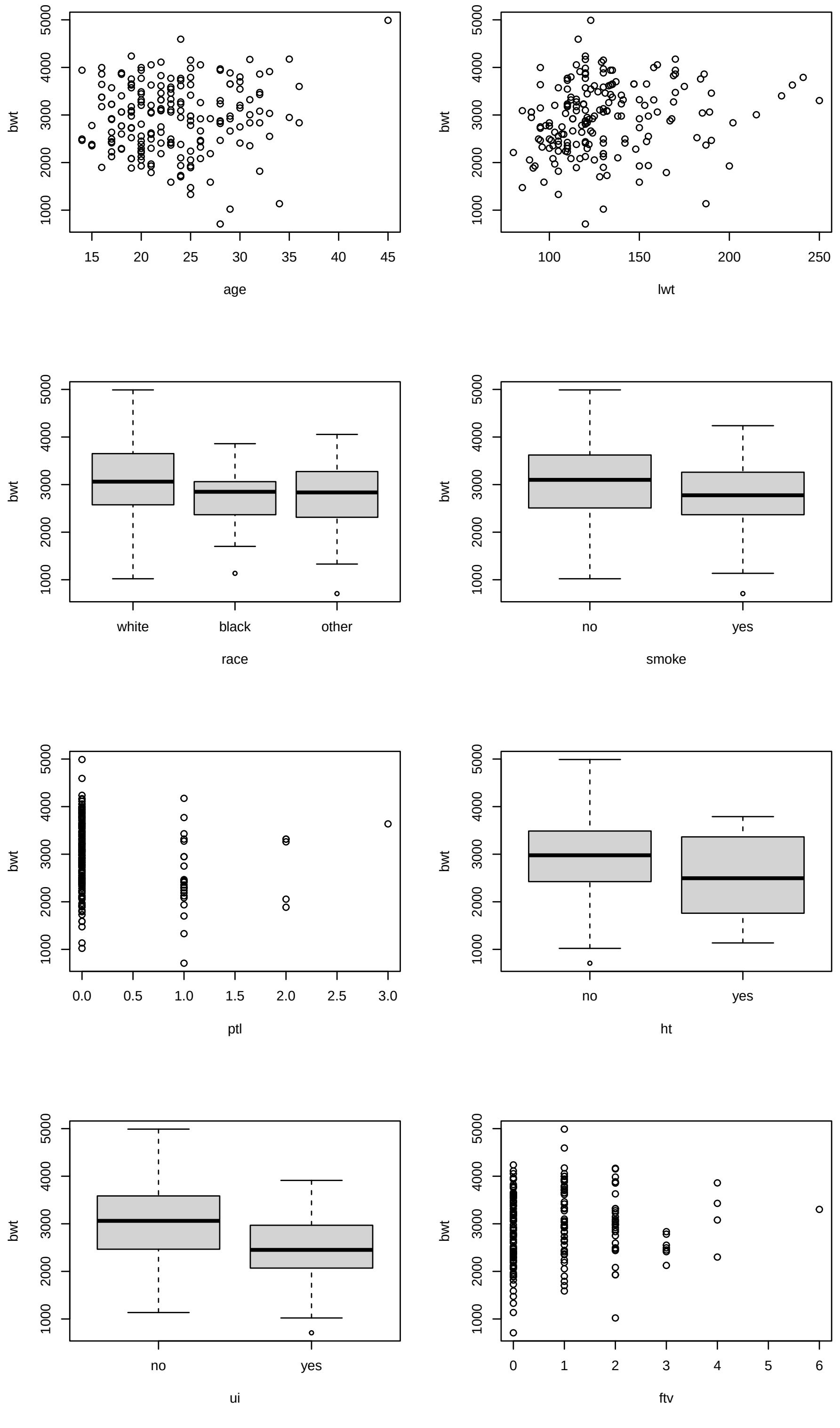
- **bwt** Dependent variable, weight of the baby at birth (grams)
- **age** mother's age
- **lwt** mother's weight before pregnancy (in pounds, lbs)
- **race** ethnicity (1=white, 2=black, 3=other)
- **smoke** smoking habit of the mother (0=no, 1=yes)
- **ptl** number of premature pregnancies mother had before the recorded one
- **ht** hypertension? (0=no, 1=yes)
- **ui** uterine irritability? (0=no, 1=yes)
- **ftv** number of medical check-ups during the first 3 months of pregnancy

6.1 Model estimation

```
getwd()

## [1] "/home/l/.sintesi/sintesi_math/statistical_models"

## Load data
## db <- read.csv("statistical_models/lab_galimberti/lab1/birthwt.csv",  sep = ";")
db <- read.csv("lab_galimberti/lab1/birthwt.csv",  sep = ";")
str(db)

## 'data.frame': 189 obs. of  9 variables:
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : int  2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: int  0 0 1 1 1 0 0 0 1 1 ...
##  $ ptl  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ht   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ui   : int  1 0 0 1 1 0 0 0 0 0 ...
##  $ ftv  : int  0 3 1 2 0 0 1 1 1 0 ...
##  $ bwt  : int  2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...

## recoding qualitative predictors
db$race <- factor(db$race, labels = c("white", "black", "other"))
db$smoke <- factor(db$smoke, labels = c("no", "yes"))
db$ht <- factor(db$ht, labels = c("no", "yes"))
db$ui <- factor(db$ui, labels = c("no", "yes"))
head(db)
```

```
##   age lwt  race smoke ptl ht  ui ftv  bwt
## 1  19 182 black    no   0 no yes   0 2523
## 2  33 155 other    no   0 no  no   3 2551
## 3  20 105 white   yes   0 no  no   1 2557
## 4  21 108 white   yes   0 no yes   2 2594
## 5  18 107 white   yes   0 no yes   0 2600
## 6  21 124 other    no   0 no  no   0 2622
```

```
# graphical bivariate plots
par(mfrow = c(4,2))
plot(bwt ~ age, data = db)
plot(bwt ~ lwt, data = db)
plot(bwt ~ race, data = db)
plot(bwt ~ smoke, data = db)
plot(bwt ~ ptl, data = db)
plot(bwt ~ ht, data = db)
plot(bwt ~ ui, data = db)
plot(bwt ~ ftv, data = db)
```


bwt
1000
2000
3000
4000
5000
15
20
25
30
35
40
45
age
100
150
200
250
lwt
white
black
other
race
no
yes
smoke
0.0
0.5
1.0
1.5
2.0
2.5
3.0
ptl
no
yes
ht
no
yes
ui
0
1
2
3
4
5
6
ftv

```
## fitting a Gaussian multiple linear regression model including all
## the regressors
model1 <- lm(bwt ~ ., data = db)
summary(model1)

##
## Call:
## lm(formula = bwt ~ ., data = db)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1825.26  -435.21    55.91   473.46  1701.20
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2927.962    312.904   9.357  < 2e-16 ***
## age           -3.570      9.620  -0.371 0.711012
## lwt            4.354      1.736   2.509 0.013007 *
## raceblack   -488.428    149.985  -3.257 0.001349 **
## raceother   -355.077    114.753  -3.094 0.002290 **
## smokeyes    -352.045    106.476  -3.306 0.001142 **
## ptl          -48.402    101.972  -0.475 0.635607
## htyes       -592.827    202.321  -2.930 0.003830 **
## uiyes       -516.081    138.885  -3.716 0.000271 ***
## ftv          -14.058     46.468  -0.303 0.762598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 650.3 on 179 degrees of freedom
## Multiple R-squared:  0.2427,Adjusted R-squared:  0.2047
## F-statistic: 6.376 on 9 and 179 DF,  p-value: 7.891e-08
```

the

- t-value column is the Wald test statistics from the lecture
- p values are two sided
- F test statisticss is the linear indipendence test and it check whether at least one of the coefficient is different from 0

```
## alternative function, glm, will be used extensively in the second
## part of the course
model1bis <- glm(bwt ~ ., family = gaussian, data = db)
summary(model1bis)

##
## Call:
## glm(formula = bwt ~ ., family = gaussian, data = db)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2927.962    312.904   9.357  < 2e-16 ***
## age           -3.570      9.620  -0.371 0.711012
## lwt            4.354      1.736   2.509 0.013007 *
## raceblack   -488.428    149.985  -3.257 0.001349 **
## raceother   -355.077    114.753  -3.094 0.002290 **
## smokeyes    -352.045    106.476  -3.306 0.001142 **
## ptl          -48.402    101.972  -0.475 0.635607
## htyes       -592.827    202.321  -2.930 0.003830 **
## uiyes       -516.081    138.885  -3.716 0.000271 ***
## ftv          -14.058     46.468  -0.303 0.762598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 422918)
##
##     Null deviance: 99969656  on 188  degrees of freedom
## Residual deviance: 75702317  on 179  degrees of freedom
## AIC: 2996.6
##
## Number of Fisher Scoring iterations: 2
```

When using lm R uses normality assumption and OLS estimation (minimize the error) when using glm R uses maximum likelihood estimation (maximize the likelihood).
The obtained model is equivalent, at least in the coefficient table; the final summary part is different

- null deviance: deviance associated to the model without any regressor (only intercept)
- residual deviance: deviance for the fitted model with the considered regressors.
  In this case deviance is used with slightly different meaning: for Gaussian model residual deviance is sum of squared residuals divided by σ^2, while here we get the sum of squared residuals.
- finally we have AIC as well

6.2 Model adequacy

To check the adequacy of the model assumptions via graphical displays of the residuals we first extract the main components

```
## See ?fitted, ?residuals, ?residuals.lm, ?rstandard
yih <- fitted(model1)    # obtain fitted values hat{y}
e1 <- residuals(model1) # extract raw residuals
r1 <- rstandard(model1) # standardized residuals (raw_resid/sqrt(s (1-H_ii)))
```

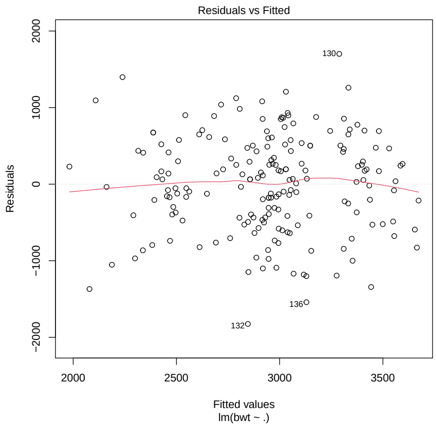
6.2.1 Linearity in the regressors

Linearity of the conditional expected value is tackled using residuals vs fitted plot;

- when we have one regressor the simple inspection of scatterplot of the two variable can help spot the assumption violation of linearity of the conditional expected value in the regressor
- when we have multiple regressors inspecting the bivariate scatterplot with y and one x at a time will not be sufficient.
  We have to go with residuals vs fitted: a severe departure from the zero horizontal line over the range of x suggest a departure from the linearity assumption and the presence of nonlinear terms. A moving average can help in doing this (point should be evenly scattered around zero).

In this case we can see that point are very close to the zero line: this suggest that linearity of the effect of the regressor on y is an adequate assumption. No systematic patterns emerges in the average of raw residuals

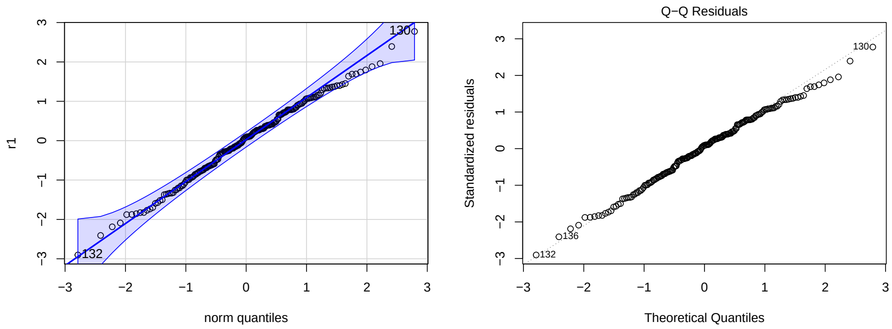
```
## ## residuals vs fitted
## plot(yih, e1) ## by hand
plot(model1, which = 1) ## ?plot.lm: 1 is residual vs fitted
```



6.2.2 Normality

If the model assumption holds, the *standardized residual* should behave as a vector of standardized gaussian vector of rv (iid from std normal).
One way to compare the empirical distribution to the theoretical one is to use qqplot: these are plot built to compare quantiles of distribution. Idea is that if two distributions have the same quantiles they're similar.
Normality can be checked by comparing with theoretical normal quantiles with the empirical quantiles of the standardized residuals: the points should be not far from the line of correspondance.
Each point is a unit in the sample: for each point we associate two quantile of this observation, one from the standardized gaussian distribution and the other the empirical distribution.
These quantiles are quite close to a straight line; the car function adds a confidence bands and all the point in this plot falls within them; note that small departures from the line in the tail can be tolerated.
we can conclude there are no relevant/systematic differences between observed and theoretical quantiles meaning the distribution of our residuals does not differ from the distribution of a sample drawn from iid std gaussian.

```
par(mfrow = c(1, 2))
car::qqPlot(r1)         ## using car library

## [1] 132 130

plot(model1, which = 2) ## standard R
```

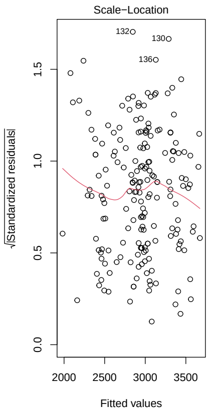


6.2.3 Homoscedasticity

The constant variance in the conditional distribution can be investigated

-
- with scale-location plot which look at standardized residuals. If the model assumptions holds the standardized residuals should have a constant variance: the raw residuals have variance depending on σ^2 and on the diagonal elements of hat matrix (so each unit can be caracterized by different variability), but if we standardize we're removing the impact of the difference in the regressors in the variability of the residuals, the impact of σ^2 and so all have same variability.
  The standardized residual are centered so the variability depends only on the absolute value; to check we have to look at strange pattern of the absolute value of the standardized residuals. if the absolute value of standardized residuals is approximately constant then it's evidence in favour of homosckedasticity.
  The R version computes a running mean: if we get a moving average that is approximately constant, the magnitude of the standardized residual is independent of the fitted value. If otoh in this plot we have a pattern deviating from the constant we should use remedies: in essence applying transformation to dependent variable/boxcox to stabilize conditional variances.
- An additional check is the Box-Cox transformation to stabilize the conditional variances (see forthcoming lecture): an optimal value for lambda is not different from 1 and suggests that no transformation is needed and the conditional variances are approximately constant (independent of the conditional expected values).
  This lambda is fitted maximizing loglik as well

```
par(mfrow = c(1,2))
## scale-location plot
plot(model1, which = 3)
MASS::boxcox(model1) ##?MASS::boxcox

## Error in eval(mf, parent.frame()):  oggetto 'db' non trovato
```



6.3 Hypothesis testing

So for this dataset the model assumption seems to be adequately respected and we can start looking at inferential tasks (who relies on model assumptions: pvalues are meaningful if and only model assumptions are met).
Otherwise those p-value could be misleading since are obtained relying on assumptions that are not adequate (eg form of distribution under the null hypothesis)

Remark 18. If we want to test using the LRT statistic in the general framework we can use equivalently `lht` or `linearHypothesis` from the package `car`

6.3.1 Linear independence

For the linear independence (all $\beta_j = 0$ forall $j \geq 1$, intercetta esclusa) we define the system with

```
## K1 has to have 9 rows (number of linear constraints) and 10 columns
## (number of constrainable parameters)

(K1 <- cbind(rep(0, 9), diag(9))) # 9 regressor coefficient and 1 intercept

##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    1    0    0    0    0    0    0    0     0
##  [2,]    0    0    1    0    0    0    0    0    0     0
##  [3,]    0    0    0    1    0    0    0    0    0     0
##  [4,]    0    0    0    0    1    0    0    0    0     0
##  [5,]    0    0    0    0    0    1    0    0    0     0
##  [6,]    0    0    0    0    0    0    1    0    0     0
##  [7,]    0    0    0    0    0    0    0    1    0     0
##  [8,]    0    0    0    0    0    0    0    0    1     0
##  [9,]    0    0    0    0    0    0    0    0    0     1

t1 <- rep(0,9)

car::lht(model1, # the unconstraint model
         K1,     # hypothesis matrix
         t1)     # rhs, constants on the right

## Linear hypothesis test
##
## Hypothesis:
## age = 0
## lwt = 0
## raceblack = 0
## raceother = 0
## smokeyes = 0
## ptl = 0
## htyes = 0
## uiyes = 0
## ftv = 0
##
## Model 1: restricted model
## Model 2: bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv
##
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    188 99969656
## 2    179 75702317  9  24267339 6.3756 7.891e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Regarding parameters
## - rhs is zero by default so in this case we could omit it
## - test param F when the value of $sigma^2$ is unknown and must be
##   estimated from the data (default for lm) or is Chisq (if we know
##   it)
```

In this case of hypothesis, another way to obtain this is via the F statistic with **anova** function comparing the unconstrained model with the constrainted one (containing only the intercept, fitting the constrainted model sometimes is straightforward)

```
## Build first a nested model with only the intercept
model2 <- update(model1, . ~ 1) # . here means the same as before here
anova(model2, model1)
```

```
## Analysis of Variance Table
##
## Model 1: bwt ~ 1
## Model 2: bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    188 99969656
## 2    179 75702317  9  24267339 6.3756 7.891e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## but we can extract RSS from the two models using deviance which
## coincides with what found before
deviance(model1)

## [1] 75702317

deviance(model2)

## [1] 99969656
```

6.3.2 Single beta = 0

Just to become acquainted with `lht` lets test $H_0 : \beta_{age} = 0$

```
## K2 : we take the first row of K1
(K2 <- K1[1, ])

##  [1] 0 1 0 0 0 0 0 0 0 0

t2 <- 0
car::lht(model1, K2, t2)

## Linear hypothesis test
##
## Hypothesis:
## age = 0
##
## Model 1: restricted model
## Model 2: bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv
##
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    180 75760555
## 2    179 75702317  1     58238 0.1377  0.711

## otherwise with anova function we remove the age
model3 <- update(model1, . ~ . -age)
anova(model3, model1)

## Analysis of Variance Table
##
## Model 1: bwt ~ lwt + race + smoke + ptl + ht + ui + ftv
## Model 2: bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    180 75760555
## 2    179 75702317  1     58238 0.1377  0.711

## in both cases p-value is the the same looking at t.value associated
## to age while the test statistics F is just the t to the power 2
(age_line <- coef(summary(model1))["age", ])

##   Estimate Std. Error    t value   Pr(>|t|)
## -3.5699344  9.6202315 -0.3710861  0.7110122
```

```
age_line["t value"]^2

##   t value
## 0.1377049
```

6.3.3 Equality of two coefficients

If we focus on impact of race, let's check equality of dummy variables $H_0 : \beta_{black} = \beta_{other}$

```
## look
# K3 : modify third row of K1 (first 1 is in the fouth column for
# raceblack)
K3 <- K1[3, ]
K3[5] <- -1
K3

##  [1]  0  0  0  1 -1  0  0  0  0  0

t3 <- 0
car::lht(model1, K3, t3)

## Linear hypothesis test
##
## Hypothesis:
## raceblack - raceother = 0
##
## Model 1: restricted model
## Model 2: bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv
##
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    180 75998327
## 2    179 75702317  1    296010 0.6999 0.4039
```

The conclusion is that the hypothesis of equality should not be rejected: there's no systematic difference afro and other (while the main model tells that there are difference between those and the baseline category).
How to fit the constrainted model in this situation? What we can do is to rethiking numeric coding for the categorical regressor: if we have two dummy in our model and the coefficients are to be put equal (putting the dummy for black equal to the one with white) means that actually we need just only one dummy variable (taking 0 if we have white, and 1 otherwise)

```
## build the restricted model by recoding the original 'race' variable
db$nowhite <- db$race != "white"
table(db$race, db$nowhite)

##
##         FALSE TRUE
##   white    96    0
##   black     0   26
##   other     0   67

model4 <- update(model1, . ~ . - race + nowhite, data = db)
anova(model4, model1)

## Analysis of Variance Table
##
## Model 1: bwt ~ age + lwt + smoke + ptl + ht + ui + ftv + nowhite
## Model 2: bwt ~ age + lwt + race + smoke + ptl + ht + ui + ftv
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    180 75998327
## 2    179 75702317  1    296010 0.6999 0.4039
## we obtain the same results
## summary(model4)
## summary(model1)
```

6.4 Comparison of non-nested models via AIC

Suppose we want to compare `model3` and `model4` which are not nested via AIC (to choose how to handle race). There are two function in R to compute AIC: as we have seen for AIC and linear regression model we have two expression

- `AIC`: full/standard AIC

$$AIC = n \ln 2\pi + n + n \ln \frac{\mathbf{e}^\top \mathbf{e}}{n} + 2(p+2)$$

- `extractAIC`: simplified obtained by ignoring the component $n \ln 2\pi + n$, not depending on the fit (in reality it ignore σ^2 too, common, by considering only $p+1$)

So values returned by the two functions *differ* only due to a constant depending on the sample size. Now, when:

- the *type of model is the same* (eg all gaussian linear) one could choose one or another, provided it will be consistently applied in all the model subject to comparison;
- we want to *compare models characterized by different distributional assumptions* (say gaussian vs lognormal linear regression model), we will use `AIC` to taking in account the whole expression

Take home message by luca: use `AIC`

```
## the models are not nested
coef(model3)

## (Intercept)         lwt   raceblack   raceother    smokeyes         ptl
## 2856.777674    4.244002 -476.906895 -348.974228 -348.048285  -53.514450
##       htyes       uiyes         ftv
## -590.305095 -512.742236  -17.157735

coef(model4)

## (Intercept)         age         lwt    smokeyes         ptl       htyes
## 2971.998538   -2.832732    3.946122 -366.180997  -47.525354 -591.884795
##       uiyes         ftv nowhiteTRUE
## -512.895777  -15.275928 -397.274691

## the two function results: full AIC with AIC
AIC(model3)

## [1] 2994.712

AIC(model4)

## [1] 2995.305

## simplified version with extractAIC: it provides the number of
## regression coefficient as well
extractAIC(model3)

## [1]    9.000 2456.354

extractAIC(model4)

## [1]    9.000 2456.946
## their differences is constant depending on sample size basically
AIC(model3) - extractAIC(model3)[2]

## [1] 538.3588

AIC(model4) - extractAIC(model4)[2]

## [1] 538.3588

## if we want to compute the simplified
nrow(db)*log(sum(residuals(model3)^2)/nrow(db))+2*length(coefficients(model3))

## [1] 2456.354

nrow(db)*log(sum(residuals(model4)^2)/nrow(db))+2*length(coefficients(model4))

## [1] 2456.946

## the additional part of the full formula, by hand
nrow(db)*log(2*pi)+nrow(db)+2

## [1] 538.3588

## Finally in this case since the two models have the same number of
## parameters/complexity (look extractAIC) the model with the lowest
## AIC is also the model with the smallest RSS (here model 3)
deviance(model3)

## [1] 75760555

deviance(model4)

## [1] 75998327
```

Chapter 7

Introducing nonlinearity

In order to make gaussian more flexible we want to overcome model assumptions which may be not respected with our data.
The first departure from the classical assumption is the violation of *linearity assumption.* There are two source of linearity

1. linearity of the regressor
2. linerity in the model parameters

7.1 A motivating example

Example 7.1.1 (Crash test data)**.** In order to evaluate the efficacy of helmets, a research team performed an experiment. In particular, after applying an accelerometer to the head of a crash test dummy, they simulated a motorcycle crash. A total of $n = 133$ readings were recorded (measured in grams), at different time points after the impact (measured in milliseconds)
In

- in figure 7.1 (a) the plot shows a clear nonlinear dependence pattern;
- if we ignore it, results from a simple linear model are the following

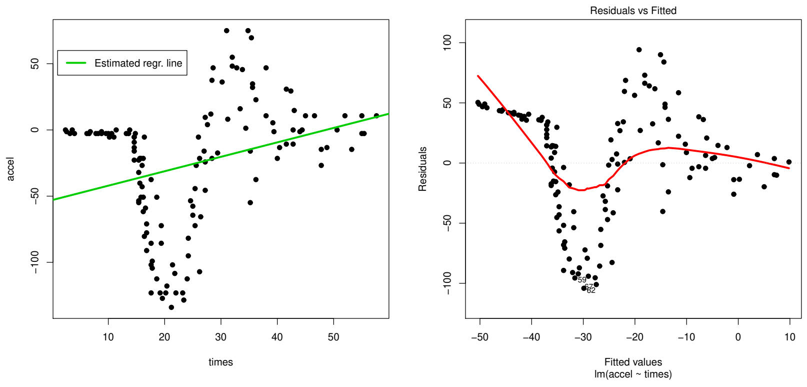
```
                 Estimate   Std. Error   t value   Pr(>|t|)
  (Intercept)    -53.008         8.712    -6.084      0.000
        times      1.091         0.307     3.552      0.001

Multiple R-squared:  0.08785, Adjusted R-squared:  0.08089
```

  we find a significant effect of time on acceleration, a very small R^2 fraction of explained variability; all these results (p-value) are obtained assuming that actual relationship between time and acceleration is linear, which is clearly not the case. In this simple univariate case plotting the x and y is enough to see it; in the multivariate model case, inspetting the bivariate plots might not be enough to understand wheter there is a violation of linearity, a possible way to circumvent is to look at residuals. One plot commonly examined is the so called *residuals vs fitted* plot

- the (raw) residual vs fitted plot 7.1 (b) suggests a clear violation of the linearity assumption: if the model assumption holds we have that

$$\begin{aligned} &\mathbb{E}\left[\mathbf{Y}|\mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta} \\ &\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{b}} \\ &\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}, \quad \mathbb{E}\left[\mathbf{e}|\mathbf{X}\right] = \mathbf{0} \end{aligned}$$

  If the functional relashionship $\mathbb{E}\left[\mathbf{Y}|\mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta}$ is correctly specified then $\mathbb{E}\left[\mathbf{e}|\mathbf{X}\right] = \mathbf{0}$ the expected value is independent of the regressors and it is equal to 0; we expect value of residuals scattered around 0, irrespective of the fitted values.
  In the right case, we expect point of this plot to be randomly scattered around zero



(a) (b)

Figure 7.1: Crash test data

without showing any kind pattern
In this example this is not the case: there is an evident pattern in the average value of the residuals (the red line is the moving average of the residuals). Initially the residuals are sistematically larger than 0 (here the model underestimates the real value; in a second part sistematically lower, in a third part larger again and then tend to stabilize around 0. We would see a red line which is kinda flat at zero

Remark 19. In situation like this we can work on the assumptionsspecifying the functional form linking the value of the regressors to the expected value of **Y**

Remark 20. We'll focus on simpler situation with one regressor at a time; this can be easily extended having more than one regressors at the same time

7.2 Gaussian nonlinear regression models

When we have a single regressor, a Gaussian nonlinear model can be defined by replacing the linearity assumption:

$$\mathrm{E}\left[Y_i|x_i\right] = \beta_0 + \beta_1 x_i$$

with the following assumption:

$$\mathrm{E}\left[Y_i|x_i\right] = h(x_i; \beta_0, \ldots, \beta_p)$$

where $h(\cdot; \beta_0, \ldots, \beta_p)$ is the nonlinear known functional form of the relation which depends on regressor and $p+1$ **unknown** parameters ($p \geq 1$).
Depending on the choice of the functional form of $h(\cdot; \beta_0, \ldots, \beta_p)$, different departures from linearity can be accomodated.

7.3 Polynomial regression

One of the simplest way of overcoming linearity is by introduction of polynomials

7.3.1 Introducing nonlinearity through polynomials

We have:

- Y_i Random variable that describes the value for the dependent variable observed on the i-th sample unit ($i = 1, \ldots, n$)
- x_i value of the regressor for the i-th sample unit

Then the so called Gaussian polynomial regression model is thus characterized by just changing the first assumption

A) we by using a polynomial of order $p \geq 1$;

$$\mathrm{E}\left[Y_i | x_{1i}, \ldots, x_{pi}\right] = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_p x_i^p = \sum_{j=0}^{p} \beta_j x_i^j, \forall i$$

B) still constant variance $\mathrm{Var}\left[Y_i | x_{1i}, \ldots, x_{pi}\right] = \sigma^2 \ \forall i$

C) still uncorrelation $\mathrm{Cor}\left[Y_i | x_{1i}, \ldots, x_{pi}, Y_h | x_{1h}, \ldots, x_{ph}\right] = 0 \ \forall i \neq h$

D) still conditional gaussianity $Y_i | x_{1i}, \ldots, x_{pi} \sim N\left(\sum_{j=0}^{p} \beta_j x_i^j, \sigma^2\right) \ \forall i$

In this case

$$h(x; \beta_0, \ldots, \beta_p) = \sum_{j=1}^{p} \beta_j x^j$$

At some point we'll have to decide which is the adequate value for p for our data at hand: the larger hte degree of polynomial the more flexible the function will be.

Interesting thing about polynomial is that while the polynomial is a nonlinear function in x, but it is still linear in the unknown parameters $\beta_0, \ldots, \beta_p$.

7.3.2 Matrix notation

Let

- $\mathbf{x}_i = \left(1 = x_i^0, x_i^1, \ldots, x_i^p\right)^\top$ be powers of the regressor value for the l'i-th sample unit: in some sense we pretend each power of a regressor (up to order p) is a separate regressor
- $\mathbf{x}_{[j]} = \left(x_1^j, x_2^j \ldots, x_n^j\right)^\top$ the powers of order j of all the n regressor values ($j = 0, \ldots, p$), with as always the regressor for the intercept as $x_{[0]} = (1, 1, \ldots, 1)^\top$

Then we can express the regressor $n \times (p+1)$ matrix as

$$\mathbf{X} = \begin{bmatrix} x_1^0 & x_1^1 & \ldots & x_1^p \\ x_2^0 & x_2^1 & \ldots & x_2^p \\ \vdots & \vdots & & \vdots \\ x_n^0 & x_n^1 & \ldots & x_n^p \end{bmatrix} = \left[\begin{array}{c} \mathbf{x}_1^\top \\ \hline \mathbf{x}_2^\top \\ \hline \vdots \\ \hline \mathbf{x}_n^\top \end{array}\right] = \left[\mathbf{x}_{[0]} \middle| \ \mathbf{x}_{[1]} \middle| \cdots \mathbf{x}_{[p]} \middle| \right]$$

And the conditional expected values in compact form:

$$\mathrm{E}\left[Y_i | x_i\right] = \sum_{j=0}^{p} \beta_j x^j = \mathbf{x}_i^\top \boldsymbol{\beta}, \forall i$$

$$\mathrm{E}\left[\boldsymbol{Y} | \mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta}$$

so the conditional expected value is stilla a $\mathbf{X}\boldsymbol{\beta}$ formulation (having considered each power as separate regressor (so from a notational pov there's no difference between a multiple regression model and a polynomial regression model, the only differences is the meaning of column of $\mathbf{X}$; the same happened for categorical regressor).

7.3.3 Linear basis expansions

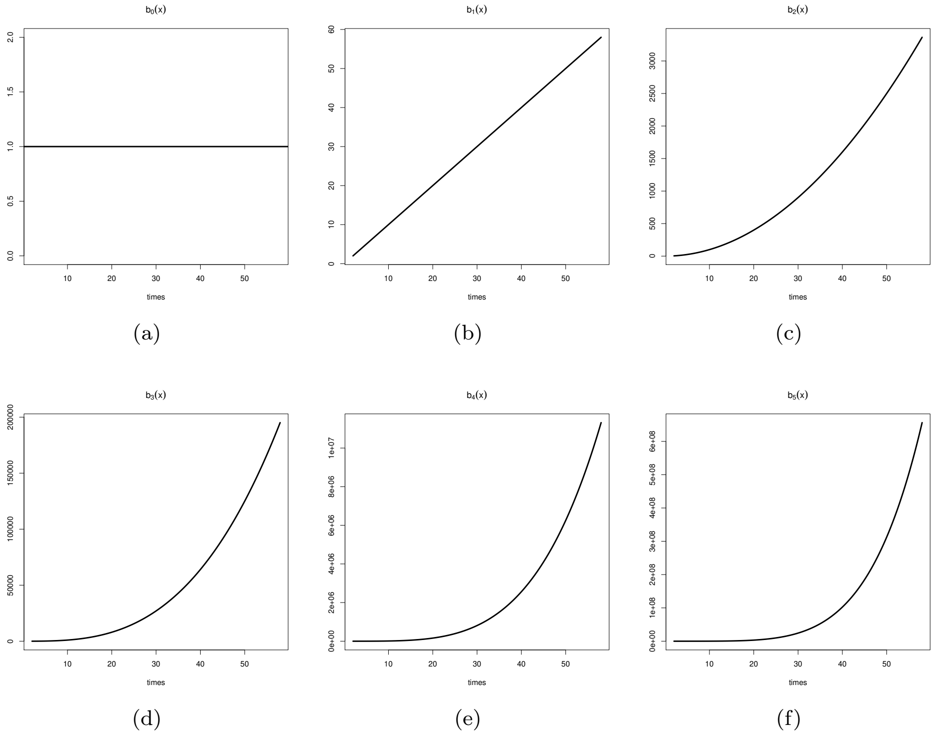
Polynomial are just one of the main examples of so called linear basis expansions, which is basically "write a nonlinear function as a linear combination of of nonlinear transformation of x".

The nonlinear fuctions $h(x; \beta_0, \ldots, \beta_p)$ used in polynomial regression models can be represented using a linear basis expansion:

$$h(x; \beta_0, \ldots, \beta_p) = \sum_{j=0}^{p} \beta_j b_j(x)$$

The functions $b_j(x)$ ($j = 0, \ldots, p$) are called *basis*. They are nonlinear transformations of x with a known functional form and without unknown parameters

There's huge set on nonlinear function in x that can be represented as linear combination of parameters beta with basis function $b_j(x)$ that are nonlinear transformation of x with a known functional form and without unknown parameter;



(a) (b) (c)

(d) (e) (f)

Figure 7.2: Poly crash

Important remark 29. With polynomial this is trivial because we use just power transformation of the regressors as basis in this linear basis expansion that is

$$b_j(x) = x^j, \quad (j = 0, \dots, p)$$

Example 7.3.1 (Polynomial basis up to $p = 5$ for the crash test data)**.** In figure7.2. So the conditional expected value will be a linear combination of these functions, which are $x_i^0, \dots x_i^5$.

Important remark 30. The trick of linear basis expansion is very useful in context of nonlinear regression, because it simplifies a lot issues related to estimation which does not change and the estimator for maximum likelihood estimation are obtained as usual with

$$\hat{\mathbf{b}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$$

Thus any property of $\hat{B}$ we have proved will also hold for this linear basis expansion stuff

7.3.4 ML estimation

Important remark 31 (Issues with straight power polynomials). One complication that occurs with straght polynomials as defined before, however, is that given their particular structure: the columns of $\mathbf{X}$ tend to be *highly correlated* (nearly linearly dependent), especially if the value of x are all positive or negative, and thus we can have:

- numerical instability due to the fact that $\mathbf{X}^\top \mathbf{X}$ is nearly singular (its determinant really close to zero and problem in computing its inverse);
- possible inflation in the standard error estimates (especially when p is large compared with n)

Example 7.3.2 (Crash test data - polynomial of order 5)**.** Sample correlation matrix among powers of the regressor are very large

	times1	times2	times3	times4	times5
times1	1.0000	0.9688	0.9112	0.8499	0.7928
times2	0.9688	1.0000	0.9833	0.9479	0.9066
times3	0.9112	0.9833	1.0000	0.9895	0.9662
times4	0.8499	0.9479	0.9895	1.0000	0.9931
times5	0.7928	0.9066	0.9662	0.9931	1.0000

R^2 measuring the linear dependence of each power on the other ones

	times1	times2	times3	times4	times5
R_j^2	0.9995298	0.9999860	0.9999972	0.9999971	0.9999776

There's almost perfect linear dependence between each column and the remainings.

Important remark 32 (Orthogonal polynomials)*.* These issues can be overcome by using a different linear basis expansion for polynomial, that is using *orthogonal polynomials*. For any matrix $\mathbf{X}$ and any vector $\boldsymbol{\beta}$:

- the matrix $\mathbf{X}$ is transformed into a matrix $\tilde{\mathbf{X}}$ (whose columns are orthogonal and have unit norm), such that $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} = \mathbf{I}_{p+1}$.
  (the actual recursive formula to be applied at each column of $\mathbf{X}$ to obtain $\tilde{\mathbf{X}}$ is omitted)
- for any $\boldsymbol{\beta} \in \mathbb{R}^{(p+1)}$ it exists a unique $\boldsymbol{\theta} \in \mathbb{R}^{(p+1)}$ ensuring that $\mathbf{X}\boldsymbol{\beta} = \tilde{\mathbf{X}}\boldsymbol{\theta}$

Example 7.3.3 (Crash test data - ortogonal polynomial basis - $p = 5$)**.** In figure 7.3.

Example 7.3.4 (Polynomials estimates comparison)**.** As parameter estimates:

- Original polynomial:

```
Coefficients:

               Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)   -105.8767      34.9883   -3.0261     0.0030
     times1     49.7816      10.3625    4.8040     0.0000
     times2     -6.3588       1.0149   -6.2655     0.0000
     times3      0.2969       0.0425    6.9819     0.0000
     times4     -0.0057       0.0008   -7.2385     0.0000
     times5      0.0000       0.0000    7.2504     0.0000
```

- Orthogonal polynomial:

```
Coefficients:

               Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)    -25.5459       2.9396   -8.6902     0.0000
  times1ort    164.5566      33.9014    4.8540     0.0000
  times2ort    131.2271      33.9014    3.8708     0.0002
  times3ort   -239.7898      33.9014   -7.0732     0.0000
  times4ort     -6.7378      33.9014   -0.1987     0.8428
  times5ort    245.7987      33.9014    7.2504     0.0000
```

Parameters estimate and t/p differs because we have two complete different set of bases. However in terms of the summary statistics:

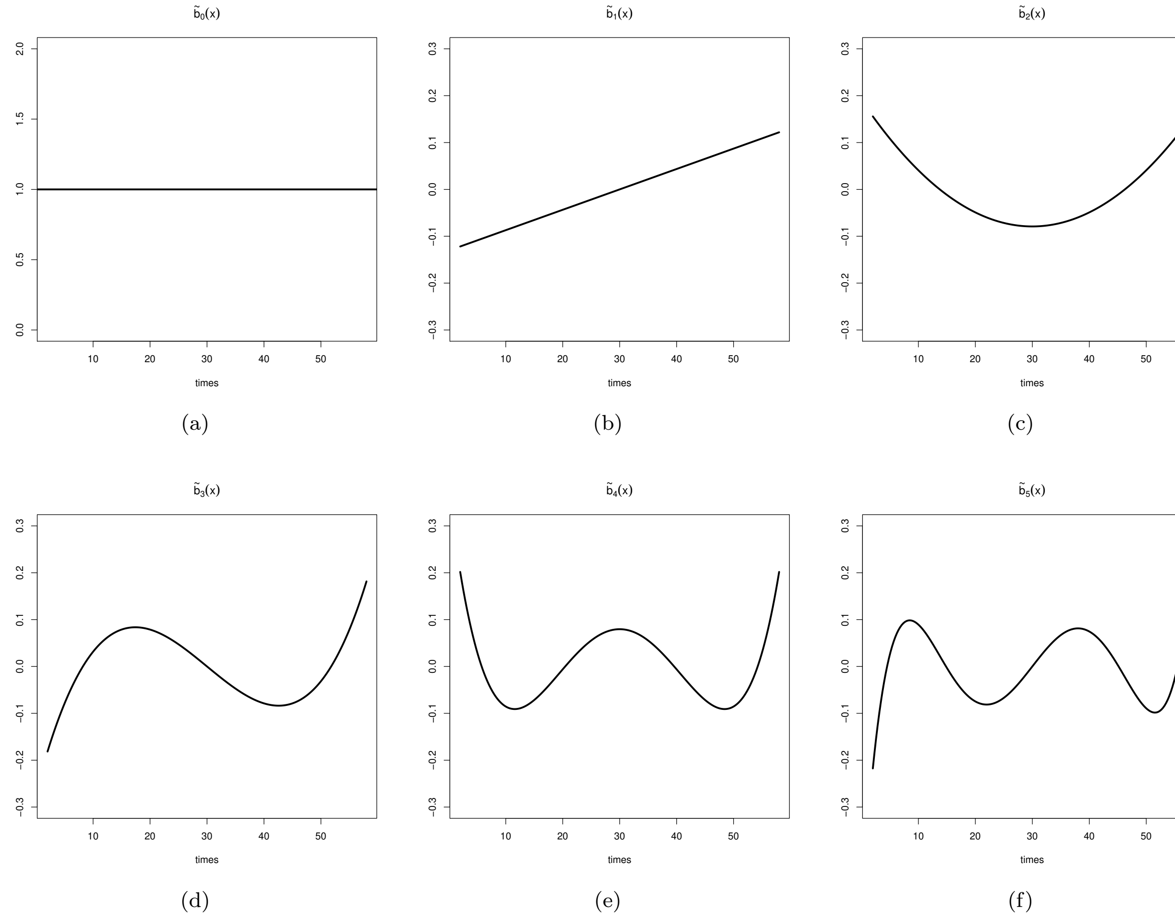
- Original polynomial:

```
Residual standard error:  33.9 on 127 degrees of freedom

Multiple R-squared:  0.5264, Adjusted R-squared:  0.5078
F-statistic:  28.24 on 5 and 127 DF, p-value:  < 2.2e-16
```

- Orthogonal polynomial:

```
Residual standard error:  33.9 on 127 degrees of freedom

Multiple R-squared:  0.5264, Adjusted R-squared:  0.5078
F-statistic:  28.24 on 5 and 127 DF, p-value:  < 2.2e-16
```

so here the results are the same. The overall test check not the linear independence assumption: it is an hypothesis where we compare the use of full polynomials vs using a constant function (so we are checking if the regressor is useful at all).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7.3: Ortogonal poly

7.3.5 Properties of regression models with orthogonal polynomials

When considering orthogonal polynomials, it is possible to prove that:

- the estimate for the intercept $\tilde{\beta}_0$ coincides with sample mean of $\bar{y}$
- the estimate for the regression coefficient associated with the j-th orthogonal bases $\tilde{b}_j(x)$ (j-th column of $\tilde{\mathrm{X}}$) coincides with the estimate of the slope of the simple Gaussian linear regression model with intercept and that base considered

  $$M_j : Y_i|x_{1i}, \ldots, x_{pi} \sim N\Big(\tilde{\beta}_0 + \tilde{\beta}_j \tilde{b}_j(x_i), \sigma^2\Big), \quad \forall i$$

  So the inclusion of an additional term in the orthogonal polynomial does not alter the estimates for the terms already included in the model
- The R^2 for the polynomial model of order p can be decomposed in the sum of the R^2s of the p simple Gaussian linear regression models M_j ($j = 1, \ldots, p$), each involving only one of the orthogonal basis.

  Therefore *the contribution of each polynomial term in explaining the variability of the dependent variable can be evaluated independently*

Example 7.3.5 (Car crash data continued)**.** THe following are the models including the intercept and the first, or second, or . . . term. We have that:

- linear term

```
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)  -25.5459       4.0170     -6.36     0.0000
  times1ort  164.5566      46.3264      3.55     0.0005
```

- quadratic term

```
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)  -25.5459       4.0868     -6.25     0.0000
  times2ort  131.2271      47.1316      2.78     0.0062
```

- cubic term

```
              Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)   -25.5459       3.7935     -6.73     0.0000
  times3ort  -239.7898      43.7484     -5.48     0.0000
```

- quartic term

```
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)  -25.5459       4.2057     -6.07     0.0000
  times4ort   -6.7378      48.5026     -0.14     0.8897
```

- quintic term

```
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)  -25.5459       3.7713     -6.77     0.0000
  times5ort  245.7987      43.4931      5.65     0.0000
```

We have that the intercept in the model is always the same; the estimated coefficient of the five separated models, are the same as well as the full estimate shown previously.

For what concerns decomposition of R^2 we have that the sum of the 5 r.squared of the model above is equivalent to the R square of the model with all the polynomials terms. Looking at this table we can say that most of the variability of acceleration is explained by cubic and quintic effect of time.

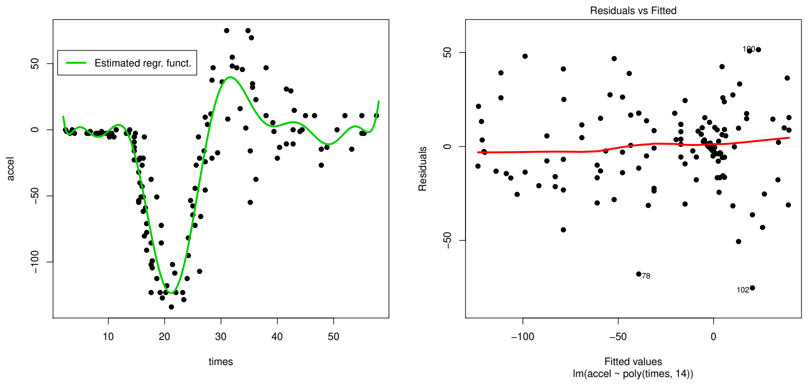
	R_j^2
linear term	0.08785
quadratic term	0.05587
cubic term	0.18660
quartic term	0.00014
quintic term	0.19600
total	0.5264

This useful decomposition is not possible if we use power transformation (because of the columns being not orthogonal)

7.3.6 Hypothesis testing

one of the key point is *choosing the degree of polynomials*: polynomial by construction are nested models. One possible strategy to choose the level, could be by exploting hypothesis testing. This can be done using the *usual F test statistics*, putting the last term equal to 0. Comparisons between nested polynomials (*choice of the degree of the polynomial*)

$$\left.\begin{aligned} &M_A : \mathrm{E}\left[Y_i|x_i\right] = \textstyle\sum_{j=0}^{p} \beta_j x^j \\ &H_0 : M_B : \mathrm{E}\left[Y_i|x_i\right] = \textstyle\sum_{j=0}^{p-q} \beta_j x^j \quad (q \leq p) \end{aligned}\right\} \quad \Rightarrow \quad H_0 : \beta_{p-q+1} = \beta_{p-q+2} = \ldots = \beta_p = 0$$



(a) (b)

Figure 7.4: Crash poly 14

The likelihood ratio test:

$$\Delta l = \frac{\mathbf{e}_B^\top \mathbf{e}_B - \mathbf{e}_A^\top \mathbf{e}_A}{\sigma^2} = D(M_B) - D(M_A) = \Delta D$$

If H_0 is true - if the polynomial of order $p - q$ is as "adequate" as the polynomial of order p:

$$\Delta D |\, M_B \sim \chi^2_q$$

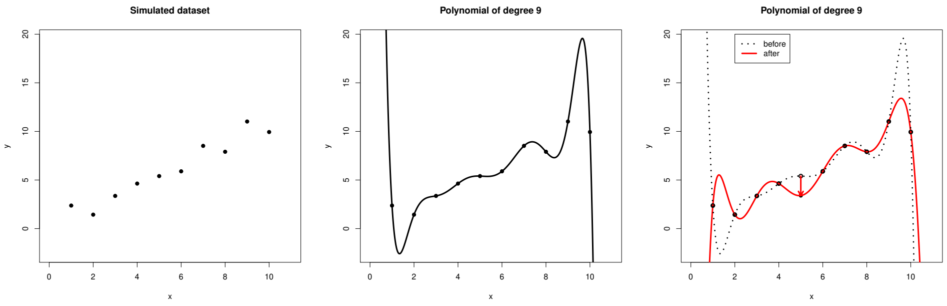
$$\left. \frac{\Delta D}{D_{M_A}} \frac{n - p_A - 1}{q} \right| M_B \sim F_{(q, n-p_A-1)}$$

Example 7.3.6 (Crash test data - polynomial of order 14)**.** The plots in fig 7.4 suggest that a polynomial (up to) of order 14 could be adequate to describe the effect of time on accelaration (no clear pattern in the average value of the residuals).
Can we get a similar ability even using only a poly of 12? To make a comparison between polynomials (order 14 vs 12) we can set the proper matrix to set to zero the last two coefficients and

```
> K
     1   2   3   4   5   6   7   8   9   10   11   12   13   14   15
 1   0   0   0   0   0   0   0   0   0    0    0    0    0    1    0
 2   0   0   0   0   0   0   0   0   0    0    0    0    0    0    1

> t

[1] 0 0

> linearHypothesis(poly14,K,t,test="F")

Linear hypothesis test

Hypothesis:

poly(times, 14)13 = 0
poly(times, 14)14 = 0
Model 1:  restricted model

Model 2:  accel ~poly(times, 14)
      Res.Df        RSS    Df    Sum of Sq       F    Pr(>F)
 1       120   61693.46
 2       118   61442.12     2       251.33    0.24    0.7860
```

We cannot refuse the null hypothesis of thirteenth and fourtenth polynoms beta being 0 so the polynomial of order 14 is not significantly better than the polynomial of order 12.

Important remark 33 (Model selection with polynomials)*.* Some strategy for model selection:



(a) (b) (c)

Figure 7.5: Cautionary remarks

- to estimate a polynomial of high order which fit data well and then come back to a lower order via testing in case higher order are not needed. When a significant worsening in the model is detected we stop;
- otherwise we can use AIC/BIC

7.3.7 Some cautionary remarks on polynomial regression

Use of polynomials can have some drawbacks; understand them formally is rather tricky but the idea is that ML estimates of the coefficients of a polynomial regression function are affected by *each* observation in the sample. in a sense polynomials are a global (as opposed to local) function.
This can lead to undesirable side-effects. In particular, a small change in one observed value for the dependent variable can lead to a dramatic change in the fitted function (even for values of the regressors that are far from that observation).
This erratic/unstable behaviour is exacerbated:

- near the boundaries of the regressor range
- when the degree of the polynomial is large (compared with the sample size)

Example 7.3.7. In figure 7.5

- (a) we have a dataset with 10 obs:
- (b) suppose we decide to fit a model with polynomial of order 9: we end up with a model with 10 parameter; this will be a saturated model (as many params as the number of covariate patterns) and the resulting fit will interpolate exactly the observed values.
In general if we choose an high value of level of polynomials we can go very close to our data; the problem with this approach is that if we have a small change with the y for any of these unit we will end up with an estimated function is completely different
- (c) we changed a bit the value of y for a unit in the middle of x: note the effect on the fitted regression function (all of it, even in areas far from the unit changed) due to a small change in a single observation

So polynomials of high degree are highly sensible to changes in the data.

Remark 21. How can we possibly deal with this problem of volatility but retaining the benefit/flexibility of polynomials? One strategy is to consider another kind of nonlinear functions which shares some property with polynomials.
the idea of this class is to give up the use of global expression for the function linking expected value of Y given X, but resort on the use of local definition

7.4 Piecewise linear regression

In these model we make a step backward using linearity again to handle nonlinearity, but a step forward since the linearity is not assumed on whole range of values of x, but its assumed

only locally.
With piecewise we divide the range of x in intervals

7.4.1 Piecewise linear functions

Definition 7.4.1 (Piecewise linear function)**.** Suppose that the range of x is partitioned into $K+1$ intervals using a known sequence of K values $l_1 < l_2 < \ldots, < l_K$ (called "*knots*"): a function $h(x)$ is said to be *piecewise linear* with fixed knots $l_1 < l_2 < \ldots, < l_K$ if:

$$h(x) = \begin{cases} \beta_{01} + \beta_{11}x & x < l_1 \\ \vdots & \vdots \\ \beta_{0k} + \beta_{1k}x & l_{k-1} \leq x < l_k \quad (k = 2, \ldots, K) \\ \vdots & \vdots \\ \beta_{0K+1} + \beta_{1K+1}x & x \geq l_K \end{cases}$$

in every interval is allowed to be characterized by a different intercept and different slope.

Important remark 34. The total number of free parameters of a piecewise linear function is given by $2 \cdot (K+1) = 2K+2$, that is *2 parameters for each interval (1 linear function for each interval).*
It's a large number of unknown parameters: the larger of number of knots/intervals, the more complex (number of parameter) h will be. Complexity increases linearly with number of number intervalse.

Remark 22. The most disturbing thing of these model, is that we're givin up the *continuity*; in each interval we can have an independent slope/intercept, which does are not conjoint on the knots and thus the function is not continuous (on the knot).
A remedy is the following

Definition 7.4.2 (Continuous piecewise linear functions)**.** A function $h(x)$ is said to be a *continuous piecewise linear* function with fixed knots $l_1 < l_2 < \ldots, < l_K$, if it is a piecewise continuous linear fuction that satisfies the following *additional continuity contraints* at each knot:

$$\begin{cases} \beta_{01} + \beta_{11}l_1 = \beta_{02} + \beta_{12}l_1 \\ \vdots \\ \beta_{0k} + \beta_{1k}l_k = \beta_{0k+1} + \beta_{1k+1}l_k \quad (k = 2, \ldots, K-1) \\ \vdots \\ \beta_{0K} + \beta_{1K}l_K = \beta_{0K+1} + \beta_{1K+1}l_K \end{cases}$$

that is the knot value coincides from the function to its left and to its right. We have K restrictions on the previous function (one for each knot): by doing this we loose flexibility which is clear in the total number of parameters.

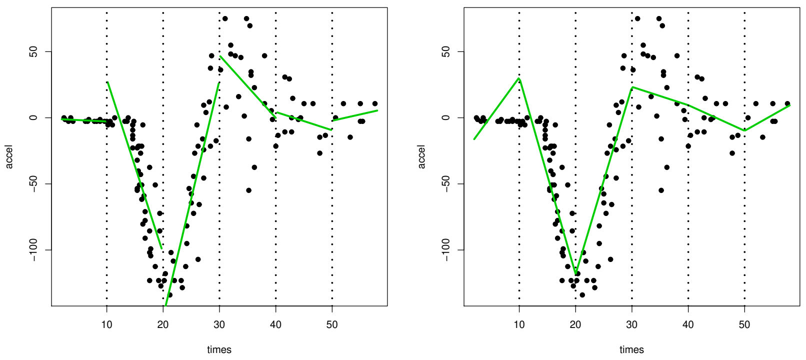
Important remark 35. The total number of free parameters of a continuous piecewise linear function is given by $2 \cdot (K+1) - K = 2K + 2 - K = K + 2$ (*2 parameters for each interval* $-$ *K restriction/constraints* to guarantee continuity)

Example 7.4.1 (Crash test data - piecewise & cont. piecewise linear functions)**.** In figure 7.6 two examples of a piecewise linear both non (a) and continuous (b) estimated on crash data using 5 knots (the dashed vertical lines denote the location of the knots)

Remark 23. In the following we see that even this class of function admit a linear basis representation so it's easy to get the estimate for the parameter of the function.

7.4.2 Linear basis expansion for cont. piecewise linear functions

It is possible to prove that:



(a)

(b)

Figure 7.6: Piecewise

- any continuous piecewise linear function with fixed knots $l_1 < l_2 < \dots, < l_K$ can be represented using a linear basis expansion with $K + 2$ bases b_j (function of the value of the regressors with non linear but known structure) and corresponding θ_j parameters:

$$h(x) = \sum_{j=1}^{K+2} \theta_j b_j(x)$$

- this linear basis expansion is not unique, in the sense that there *exist several possible choices* for the basis functions $b_j(\cdot)$

7.4.2.1 Truncated linear basis

One possible set of bases we can use is the on

$$b_j(x) = \begin{cases} x^0 = 1 & j = 1 \\ x^1 = x & j = 2 \\ (x - l_{j-2})_+ & j = 3, \dots, K + 2 \end{cases}$$

where $(\cdot)_+$ denotes the positive portion of its argument:

$$(r)_+ = \begin{cases} r & r \geq 0 \\ 0 & r < 0 \end{cases}$$

Example 7.4.2 (Crash test data - example of truncated linear basis)**.** In figure 7.7 a graphical representation of the 7 bases used to build the continuous piecewise linear function shown before; the first base is constant 1, the second is the identity, the third is the positive portion of $(x - 10)$, the fourth of $(x - 20)$ and so on . . .

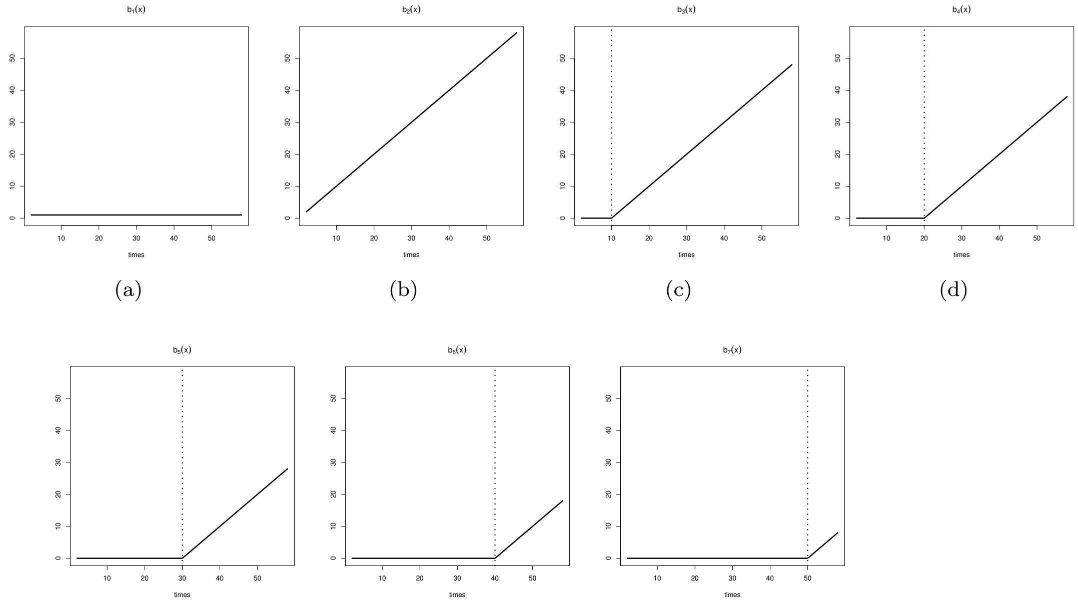
Important remark 36. With piecewise constant stuff we have continuity but the function is not derivable/well behaved in the knots; here comes the spline functions.

Remark 24. In a sense continuous piecewise linear function are a class of the broader next group of functions

7.5 Regression splines

7.5.1 Spline functions

Definition 7.5.1 (Spline function)**.** A function $h(x)$ is said to be a spline function of degree m with fixed knots $l_1 < l_2 < \dots, < l_K$ if:



(a) (b) (c) (d)

(e) (f) (g)

Figure 7.7: Crash test - an example of truncated linear basis

- it can be described as a polynomial of degree m in each interval defined by the knots:

$$h(x) = \begin{cases} \sum_{j=0}^{m} \beta_{j1} x^j & x < l_1 \\ \vdots & \vdots \\ \sum_{j=0}^{m} \beta_{jk} x^j & l_{k-1} \leq x < l_k \quad (k = 2, \dots, K) \\ \vdots & \vdots \\ \sum_{j=0}^{m} \beta_{jK+1} x^j & x \geq l_K \end{cases}$$

- its partial derivatives with respect to x are continuous up to the order $m-1$.

Example 7.5.1. Continuous piecewise linear functions are splines of degree 1 (there's a straight line in each interval) and the function is continuous up to order 0 (the derivative of order 0, the function itself, is continuous $\frac{\partial^0}{\partial x^0} h(x) = h(x)$)

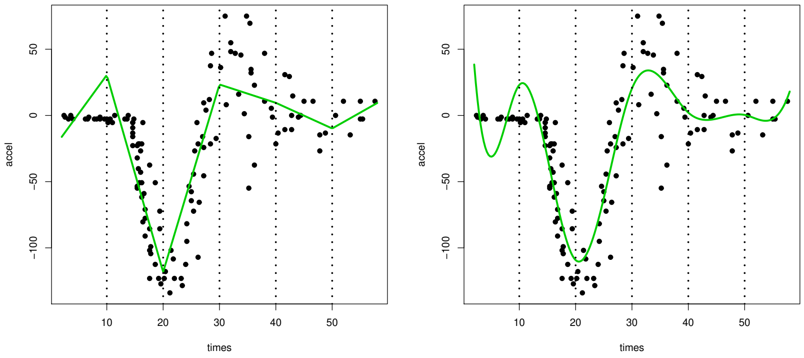
Important remark 37 (Number of parameters). Considering that

- we have $m+1$ parameters (polynomial of order m) in each interval
- with K knots we have $K+1$ intervals
- we have m constraints/restriction for each of the K knot (to impose continuity up to the order $m-1$ of partial derivatives)

the total number of parameters we have when dealing with a spline function is given by

$$(K+1)(m+1) - Km = Km + K + m + 1 - Km = K + m + 1$$

Example 7.5.2. If $m = 1$ (continuous piecewise linear function) we get $K+2$ parameters as saw before



(a)

(b)

Figure 7.8: Linear vs cubic spline

7.5.2 Cubic splines

Important remark 38. Among the class of splines function the cubic splines are an interesting case

Definition 7.5.2 (Cubic spline function)**.** A function $h(x)$ is a cubic spline with fixed knots $l_1 < l_2 < \ldots, < l_K$ if it is a spline function of degree 3

Remark 25. Totally, the total number of parameters of a cubic spline with K fixed knots is given by $K + 4$

Remark 26. Notice that while the splines in general are continuous at the knots, their first partial derivative are not (so left and right limits of second partial derivative may differ).
In the first order (piecewise linear) we can spot easily when this occurs (at the knots); by looking at quadratic spline again we can see that something is “odd” at the knots.

Important remark 39 (Why interesting?). Some authors say that cubic splines are the lowest order splines for which *the discontinuity in the partial derivatives at the knots cannot be noticed by the human eye*; so this kind of splines are the first one being very smooth.
A more substantial reason is that cubic splines have interesting mathematical properties as we will see next (coming from a completely different approach).

Example 7.5.3 (Crash test data - linear vs cubic regression splines)**.** In figure 7.8, the dashed vertical lines denote the location of the knots; the cubic is very smooth in the knots but still the behaviour is not satisfactory is some interval (look the first one)

7.5.3 Linear basis expansion for spline functions

Remark 27. Despite the complicated definition function (polynomial of order m and constraints) is possible to express splines matrixly-easily for estimation by introducing bases of $K + m + 1$ function (not subject to any restriction, easier to deal with) b_j combined linearly with θ_j parameters

Important remark 40. It is possible to prove that:

- any spline function of degree m with fixed knots $l_1 < l_2 < \ldots, < l_K$ can be represented using a linear basis expansion:

$$h(x) = \sum_{j=1}^{K+m+1} \theta_j b_j(x)$$

- again this linear basis expansion is not unique, in the sense that there exist several possible choices for the basis functions $b_j(\cdot)$ to represent the spline function

7.5.3.1 Truncated power basis for spline functions

Remark 28. An example of basis that can be used to represent are the truncated power basis which are a generalization of the truncated linear basis (if we set $m = 1$ in the following we have the same results introduced before). The first m bases are x^0 up to x^m;the remaining K bases can be obtained as positive part with respect a knot (to the power m in this case);

Definition 7.5.3.

$$b_j(x) = \begin{cases} x^{j-1} & j = 1, \dots, m+1 \\ (x - l_{j-m-1})_+^m & j = m+2, \dots, K+m+1 \end{cases}$$

where

$$(r)_+^m = \begin{cases} r^m & r \geq 0 \\ 0 & r < 0 \end{cases}$$

Remark 29. These are the most simple basis; but suffers from some issues.

Remark 30. Despite their simple and intuitive structure, truncated power basis are rarely used in practice (they are actually not implemented in `R`).

Important remark 41 (Problems). This is due to the fact that the columns of the corresponding matrix $\mathbf{X}$ tend to be highly correlated (nearly linearly dependent), thus leading to nearly singularity of $\mathbf{X}^\top\mathbf{X}$ and numerical instability in the estimation process.

7.5.3.2 B-spline basis functions

Remark 31. An alternative linear basis expansion representation that does not suffer these problems can be obtained by resorting to the so-called B-spline basis functions.
We can think of it as the equivalent of orthogonal polynomial basis function for spline

Important remark 42. B-spline basis functions are defined using a recursive formula (omitted): each B-spline function takes non-zero values only between a pair of knots (the actual definition of this interval depends on m)

Example 7.5.4 (Crash test data - an example of B-spline basis for linear splines)**.** In figure 7.9: if we want to use b-spline basis to represent linear splines function, this is what we get from the recursive formulas, that is function that are most equal to 0 with exception of som intervals (eg the tird only between the first and the third knot).

Example 7.5.5 (Crash test data - an example of B-spline basis for cubic splines)**.** By applying the recursive formulas we can obtain the bases for a *cubic* spline (here 9 bases $K + 4$, with $K = 5$), in figure 7.10
Again they're non 0 only on some subinterval (up to 4 consecutive intervals) and are much smoother than the previous ones

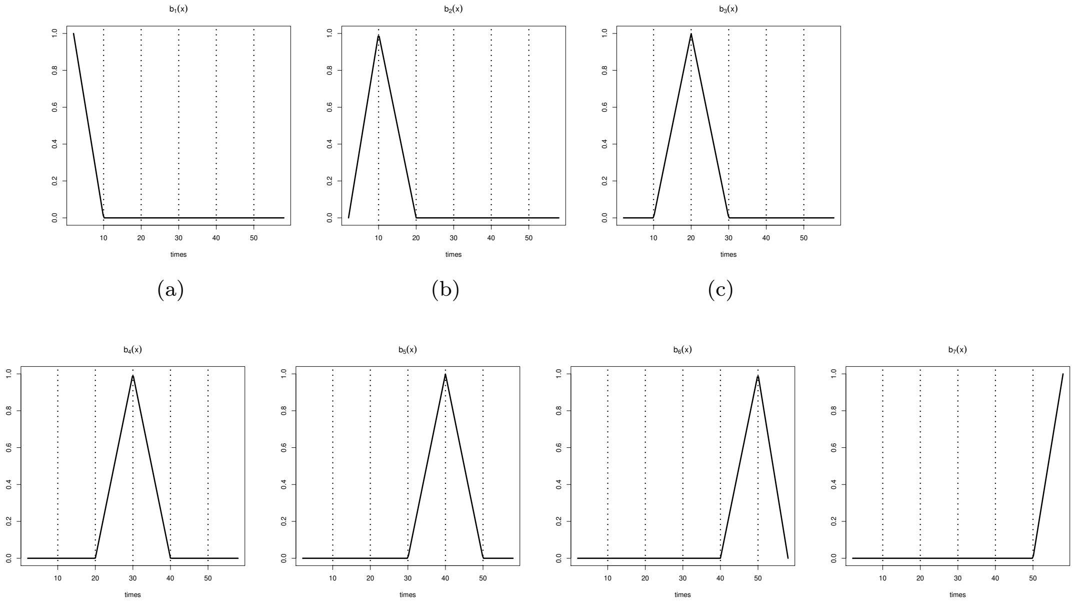
7.5.4 Concluding remarks

7.5.4.1 Estimation

In general, maximum likelihood estimates of the parameters $\theta_1, \dots, \theta_{K+m+1}$ for a given linear basis expansion can be *easily obtained using standard tools* for Gaussian regression models with regression functions that are linear in the unknown parameters.

7.5.4.2 Inference

Comparisons among regression spline models require *some caution.*
In practice, to use splines we have to choose the degree of the spline function, and the number/location of the knots; for the same dataset, even using the same degree of the splines function, we can come up with several competing moedel, depending on number/location of knots.



(a) (b) (c)

(d) (e) (f) (g)

Figure 7.9

- when using a truncated power basis representation and we're comparing splines function obtained by **adding or removing** specific knots, then we can use *likelihood ratio test* since the models are nested: here removing 1 knot is equivalent to set to 0 the parameter associated to the basis in which that knot is involved.
  When dealing with B-spline otoh we have to use model selection because models are not nested anymore, because the removal of one knot does not to boil down to setting a $theta_j$ to 0
- a change in the **location of one** (or more) *knot* leads to a non-nested model: here we adopt *model selection criteria*

In general the safer approach which can handle different type of splines is: just do the model selection stuff (AIC/BIC, LOOCV error etc).

7.5.4.3 Location of knots

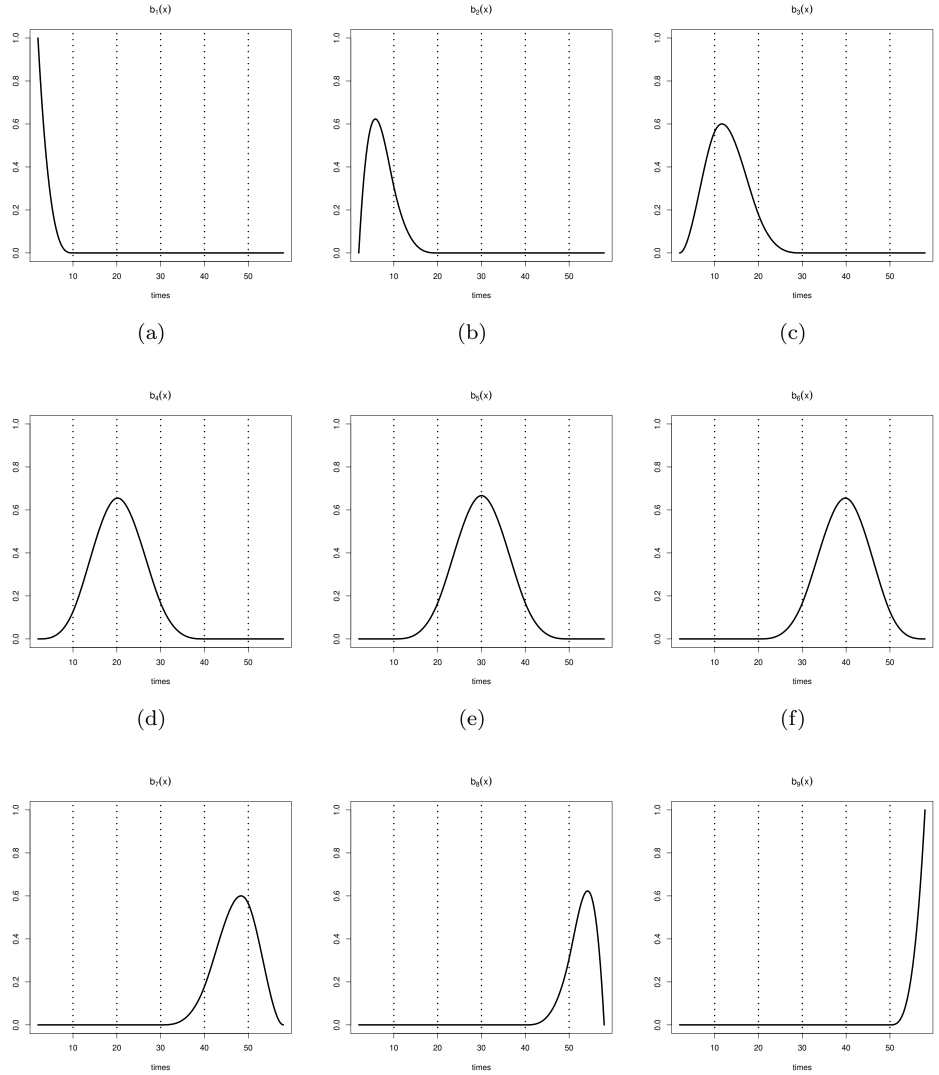
Remark 32. When using splines we have to decide both number and location

Important remark 43 (Knots location choice). Regarding location:

- it's a subjective choice, we could consider any location (eg by educated guessing or looking at scatterplot of the data);
- we may use *equidistant* knots;
- otherwise knots located at the *quantiles of the regressor*: this choice guarantees an approximate constant number of sample units within each interval (differently from the previous)

Example 7.5.6 (Crash test data - a comparison among alternative models)**.** In:

- figure 7.11 theres a summary for AIC comparison of several alternative models (both by type of h function and number of parameters); for all the models, the AICs tend to decrees up to a certain point where we reach an optimal level of number of parameter (different numbers for the three methods btw)
- the best models are



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7.10

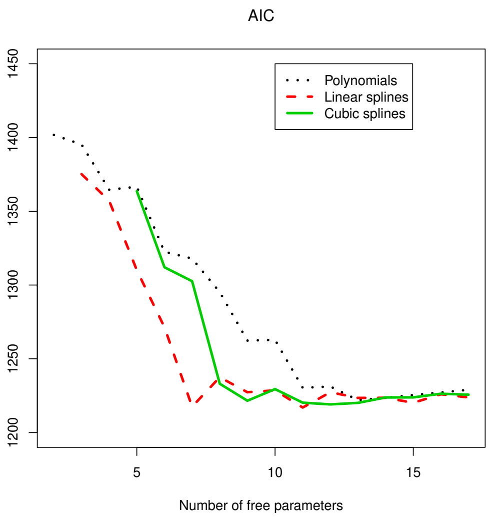


Figure 7.11: Splines comparison

- polynomial of order 12 with then 13 parametesr
- linear spline with 9 knots and therefore 11 parameters, with knots located with quantiles
- cubic spline (8 knots, 12 parameters), knots again located

the results are in **??**, as well with their residuals; the first polynomial model, having a global solution, get a little bit wild in the final part

The best option among these would be linear splines according to AIC; then we could argue regarding the smoothness but this is a point where there's subjective opinion.
In terms of residual more or less all the mean value are 0 over the range of x (retta piana)

7.5.4.4 Polynomials & spline functions - some remarks

They key drawback of polynomials is their sensitivity to the data in all the x range.
Differently from polynomials, spline functions have a local nature:

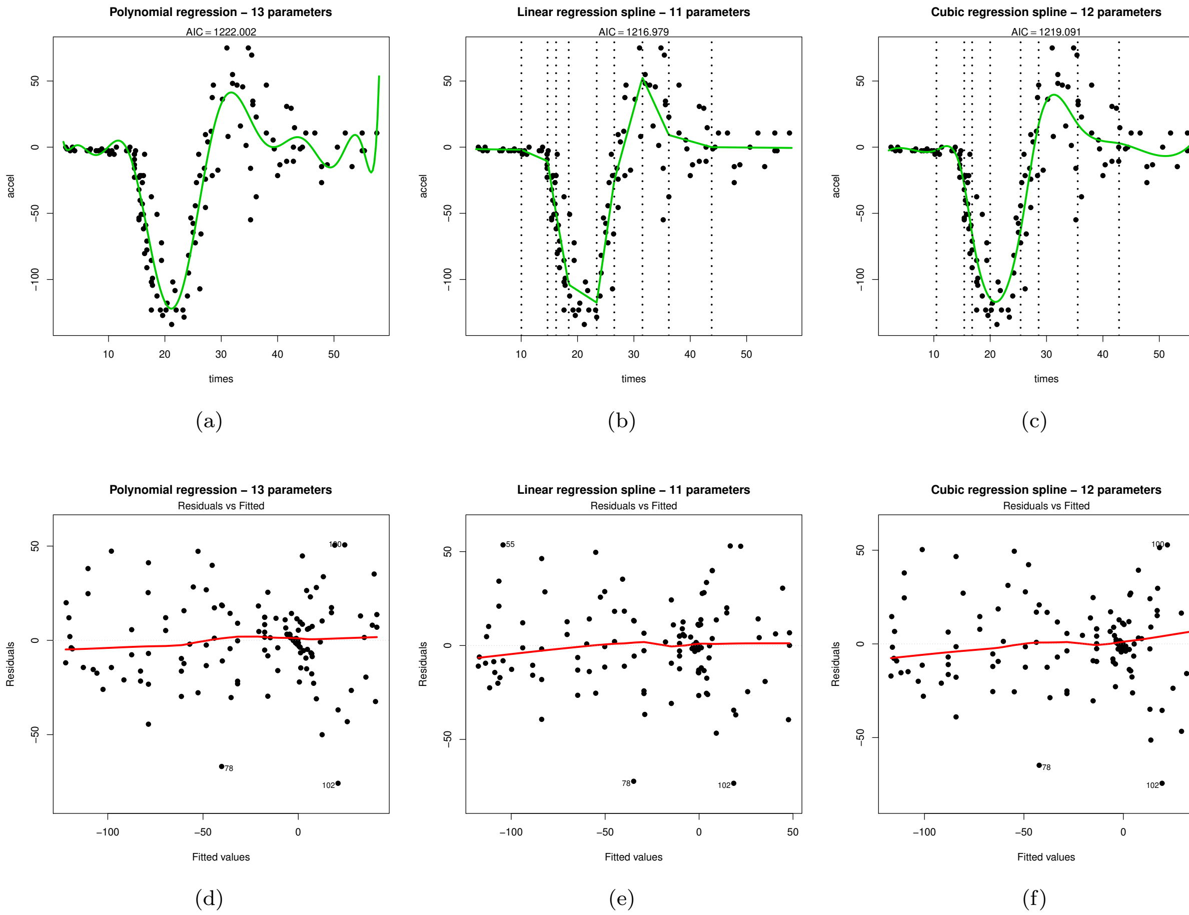
- they are structured as local polynomials defined on non-overlapping intervals;
- a change in one observed value for the dependent variable affects only some of the polynomials that compose the spline functions (the ones close to the interval to which the observation belongs), but leaves the other polynomials unchanged

Example 7.5.7 (Polynomials & spline functions)**.** In figure 7.13:

- in (a) going back to the data in the lecture of polynomials, we fitted a cubic spline (with 6 knots, 10 parameters btw); we come up with and estimated function which perfectly interpolate the data (its a saturated model as well)
- in (b) a slight change in the value of y for a unit, thanks to the local nature of the spline function we have an impact that is local only, there are no ripercussion elsewhere

7.5.4.5 Concluding remarks

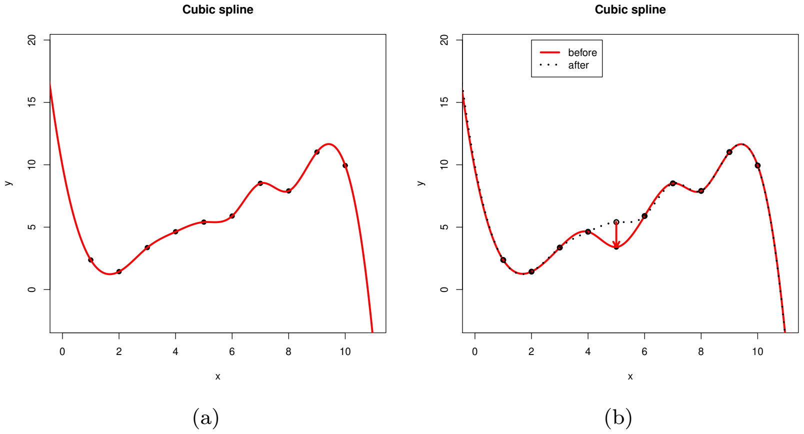
Besides polynomials and splines, there exist many other examples of functions (which can be extended to deal with more than one regressor) that admit a linear basis expansion representation and so can introduce non linear relatioship between x and y but preserving linearity/simplicity in the estimation/parameters.



(a) (b) (c)

(d) (e) (f)

Figure 7.12: Crash - Best splines models and their residuals



(a) (b)

Figure 7.13: Poly/splines comparison

Polynom and splines function can be used as well when we have more than 1 regressor and we want to allow each of these regressors to have a non linear impact on the independent variable One way to do so is via the so called class of *additive models*

$$\mathrm{E}\left[Y_i|x_{1i}, \ldots x_{pi}\right] = h_1(x_{1i}; \boldsymbol{\beta}_1) + \ldots + h_p(x_{1pi}; \boldsymbol{\beta}_p)$$

In this case rather than having a linear effect for each regressor we use a nonlinear h function: so we use say a spline to describe the impact of x_1 + another spline for the impact of x_2 and so on and so forth.
This permits to overcome the linearity of the expected values in the regressors but still preserves the linerity in the model parameters

7.5.4.6 Problems with splines so far

Regression (cubic) splines: are an attractive solution to enhance the flexibility of Gaussian regression model bu they suffer from a major shortcoming: the **choice of the number and of the locations of the knots**:

- it is basically not possible to completely avoid any amount of subjectivity and to make this choice in a systematic/objective manner;
- considering equidistant knots or placing knots at quantiles are suboptimal strategies, leading to complications when performing model comparisons (models with different numbers of knots are not nested).

Chapter 8

Introducting regularization

Remark 33. We go on with techniques that allow to on with nonlinearity aside from polynomial and splines (using as regressor function)

8.1 Smoothing splines

8.1.1 A (seemingly unrelated) alternative approach

We consider the regression model with three main assumptions

A) the conditional expected value can be written as a function h of x_i

$$\mathrm{E}\left[Y_i|x_i\right] = h(x_i), \forall i$$

There are no explicit assumptions on the functional form of $h(\cdot)$ and of the conditional distribution of $Y_i|x_i$ are introduced. The only requirement are:

- existence and continuity of its second partial derivative

$$h''(x) = \frac{\partial^2}{\partial x^2} h(x)$$

- the square of the second partial derivative must be integrable integral defined

$$(0 \leq) \int \left[h''(t)\right]^2 dt < +\infty$$

the integral is computed on the entire range of x and will be positive

B) the conditional variance is constant

$$\mathrm{Var}\left[Y_i|x_i\right] = \sigma^2, \forall i$$

C) there's no correlation in the conditional distributions $\mathrm{Cor}\left[Y_i|x_i, Y_h|x_h\right] = 0$, $\forall i \neq h$

Important remark 44. Note for these three assumption we are not defininig a parametric statistical model/distributional assumptions; this is an example of *nonparametric model* because we're saying something of $Y|x$ but we're not fully specifying the conditional distributions.

Important remark 45 (Roughness of a function). The integral showing up in the conditions:

$$\int \left[h''(t)\right]^2 dt$$

can be interpreted as a measure of the *roughness/departure from linearity* of $h(\cdot)$; we may think of it as a measure of the total variability of the first partial derivative $h'(\cdot)$:

- if $h(\cdot)$ is linear, its first partial derivative is constant and then $h''(x) = 0$ and $\int \left[h''(t)\right]^2 dt = 0$ (furthermore $h''(\cdot)$ is not affected if a constant or a linear term is added to $h(\cdot)$)
- if $h(\cdot)$ is wiggly, since h is increasing and decreasing $h'(\cdot)$ is variable/nonconstant and thus $h''(x) \neq 0$; the larger the absolute value of $h''(\cdot)$, the larger $\int \left[h''(t)\right]^2 dt$

8.1.2 Penalized least squares estimation

Remark 34. The idea is to penalize for the level of roughness/non linearity directly in the estimation process

Important remark 46 (Penalized estimation procedure). An estimate for $h(\cdot)$ can be obtained by minimizing

$$pls_\lambda(h(\cdot)) = \sum_i (y_i - h(x_i))^2 + \lambda \int \left[h''(t)\right]^2 dt$$

which is related to:

- $\sum_i (y_i - h(x_i))^2$ determines the goodness of fit to the data (*the smaller, the better*)
- $\int \left[h''(t)\right]^2 dt$ acts like a penalty for roughness (*the smaller, the better*)
- $\lambda \geq 0$ is the regularization/smoothness parameter controlling the *trade-off between goodness of fit and roughness*

Important remark 47 (Role of the smoothing parameter). The smoothing parameter λ controls the trade-off between goodness of fit and roughness:

- if $\lambda = 0$ no penalization for roughness is imposed and *the resulting fitted model will be equivalent to a saturated model* (we allow the function h to be as flexible as possible)
- as $\lambda \to +\infty$ any nonlinear function (with $h''(x) \neq 0$) is excluded (*only linear functions are considered*)

Example 8.1.1 (Crash test data - penalized LS estimation)**.** An example of the impact/tradeoff of λ on the function h estimated is shown in figure 8.1.

Important remark 48 (Which kind of h to consider). Now, in practical terms, we could/should consider all the function h with continuous second partial derivative and all the other requirements; so the space of possible functions would be very huge and from a practical POV exploring this space of function could be a complicated task.
Luckily for us there's an interesting result that ease the exploration and the choice of h a very easy task to perform.

8.1.3 Penalized LS estimation and spline functions

Proposition 8.1.1. *It is possible to prove that, for a given value of λ:*

- *$pls_\lambda(h(\cdot))$ admits a unique minimizer:*
- *the minimizer of $pls_\lambda(h(\cdot))$ is a **natural cubic spline** with knots located at the unique values of x_i ($i = 1, \dots, n$).*

Important remark 49. So we do not need to explore the whole set of h function respecting requirements, we can focus on a specific subset of splines. They are cubic splines (and so know they have linear basis shit) and we've solved the problem of number/location of knots (just use for each unique value)

Remark 35. For this result the penalized LS approach described previously is also known as **smoothing spline** approach

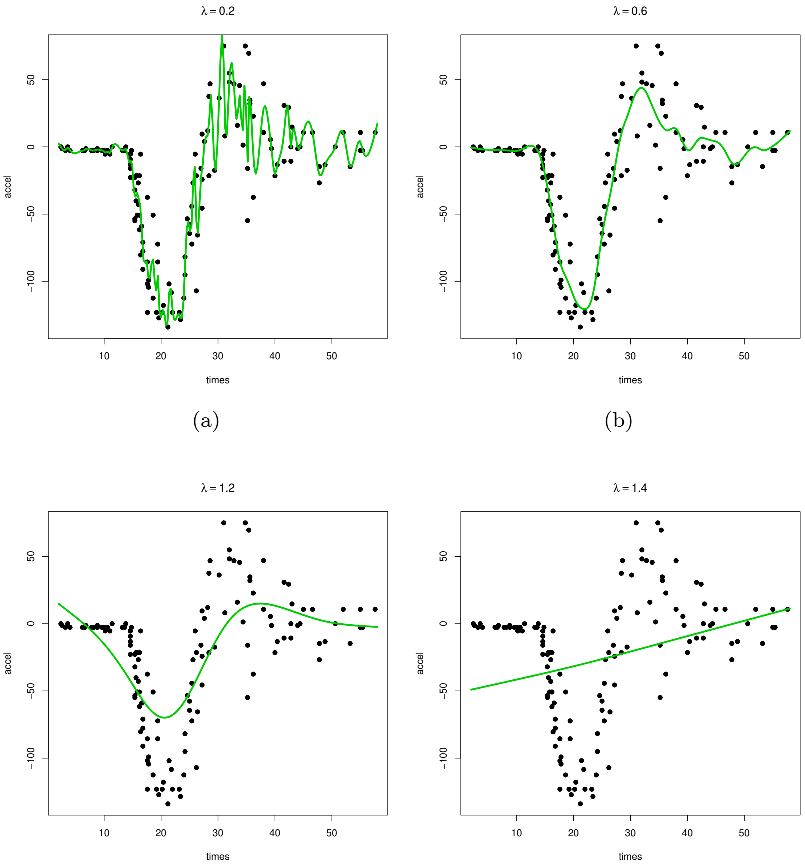
Remark 36. Smoothing splines differ from regression splines due to the presence of the penalty term $\int \left[h''(t)\right]^2 dt$, that implicitly introduce constraints on the parameters of the spline function. The strength of these constraints is controlled by the smoothing parameter λ

8.1.4 Natural cubic splines

Definition 8.1.1. A function $h(x)$ is a natural cubic spline with fixed knots $l_1 < l_2 < \dots, < l_K$ if it is *a cubic spline function with the additional constraints* that:

$$h(x) = \beta_{01} + \beta_{11}x, \quad \text{if } x < l_1$$
$$h(x) = \beta_{0K} + \beta_{1K}x, \quad \text{if } x > l_K$$

Remark 37. So:



$\lambda = 0.2$

$\lambda = 0.6$

(a)

(b)

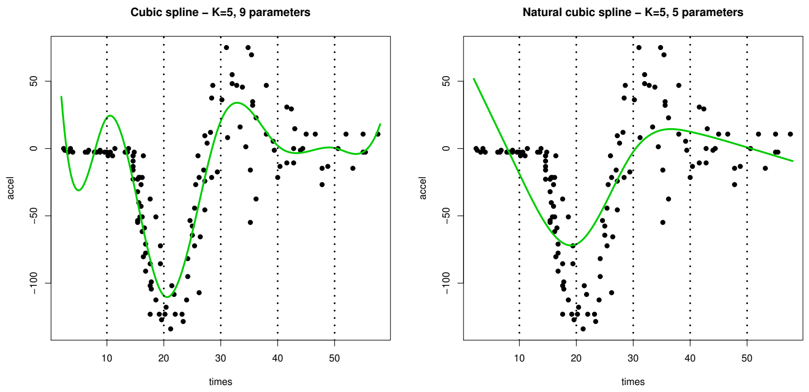
$\lambda = 1.2$

$\lambda = 1.4$

(c)

(d)

Figure 8.1: Crash test data - penalized LS estimation



(a)

(b)

Figure 8.2: Natural cubic spline

- we still have to choose the knots;
- the first and last polynomials are forced to have degree 1 (with second partial derivatives equal to 0), not 3
- instead of $k+4$ (cubic spline), the total number of free parameters of a natural cubic spline with K fixed knots is given by K (4 additional restrictions are imposed, that is to have linear first and last segment we leave 4 coefficient set to 0 basically)

Example 8.1.2 (Crash test data - natural cubic splines vs. cubic splines)**.** In figure 8.2 the comparison between the two where the major differences are in the first and last interval (but not only since splines are somewhat connected

Proposition 8.1.2 (Linear basis expansion for natural cubic spline functions)**.** *Being subset of cubic splines is possible to prove that:*

- *any natural cubic spline function with fixed knots $l_1 < l_2 < \ldots < l_K$ can be represented using a linear basis expansion (with K elements/bases):*

$$h(x) = \sum_{j=1}^{K} \theta_j b_j(x)$$

- *this linear basis expansion is* not unique*, in the sense that there exist several possible choices for the basis functions $b_j(\cdot)$, and can be obtained starting from the basis functions of the corresponding cubic spline function with the same fixed knots*

Example 8.1.3 (Crash test data - an example of basis for natural cubic splines)**.** How we can build a basis for natural cubic splines? The following is just an example.
We can start from the truncated power basis for a cubic splines and operate some changes we can build a set of K bases for a natural cubic spline:

$$b_j(x) = \begin{cases} x^0 = 1 & j = 1 \\ x^1 = x & j = 2 \\ d_{j-2}(x) - d_{K-1}(x) & j = 3, \ldots, K \end{cases}$$

where

$$d_{j-2}(x) = \frac{(x - l_{j-2})_+^3 - (x - l_K)_+^3}{l_K - l_{j-2}}$$

and

$$(r)_+^3 = \begin{cases} r^3 & r \geq 0 \\ 0 & r < 0 \end{cases}$$

Important remark 50. The fact that the first two base are the constant and identity function means that natural cubic splines family of functions contains as special cases both costant function and linear function. Using these basis we can obtain it just by imposing restriction on the parameters:

- $h(x)$ is constant if $\theta_j = 0$ for $j = 2, \dots, K$
- $h(x)$ is linear if $\theta_j = 0$ for $j = 3, \dots, K$

8.1.5 Penalized LS estimation (matrix notation)

Remark 38. In order to solve the finding of h by penalized we can focus on this subsect of splines (natural cubic splines); let's see in practice how we can find the solution which minimizing the stuff

We see what happens in the special case *when the number of unique values for x_i ($i = 1, \dots, n$) is equal to n* (so number of knots are n as well - situation like the saturated model) we have to build a basis of n components:

$$h(x_i) = \sum_{j=1}^{n} \theta_j b_j(x_i), \quad i = 1, \dots, n$$

where:

- $\mathbf{N}$ is $n \times n$ matrix we can build containing the values of the n basis evaluated on each sample unit
$$\mathbf{N} = \begin{bmatrix} b_1(x_1) & \dots & b_n(x_1) \\ \vdots & \ddots & \vdots \\ b_1(x_n) & \dots & b_n(x_n) \end{bmatrix}$$
This will play the same role as the regressor matrix in the usual regression context we've discussed so far
- $\boldsymbol{\theta}$ is n-dimensional vector with unknown parameters

Thus we can express the function h applied to all the elements in the sample as

$$\begin{bmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{bmatrix} = \mathbf{N}\boldsymbol{\theta}$$

Now the problem is finding estimates for thetas counting on the fact that the minimizer of the penalized stuff is a natural cubic spline.
When the number of unique values for x_i ($i = 1, \dots, n$) is equal to n if $h(x_i)$ is a natural cubic spline with fixed knots at the unique values for x_i ($i = 1, \dots, n$), it is possible to prove that:

- we can rewrite the second derivative part of the penalization as a quadratic form of the parameter vector $\boldsymbol{\theta}$ (which easier than an integral):
$$\int \left[h''(t)\right]^2 dt = \boldsymbol{\theta}^\top \mathbf{P} \boldsymbol{\theta}$$
where $\mathbf{P}$ is an $n \times n$ symmetric matrix whose entries depend only on the differences between consecutive values of x_i (the actual formulas are omitted);
- overall the penalized least square criterion $pls_\lambda(h(\cdot))$ contemplating both residuals and wizziness of function can be re-expressed as follows:
$$pls_\lambda(h(\cdot)) = pls_\lambda(\boldsymbol{\theta}) = (\mathbf{y} - \mathbf{N}\boldsymbol{\theta})^\top (\mathbf{y} - \mathbf{N}\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}^\top \mathbf{P} \boldsymbol{\theta}$$
therefore finding the function $h(\cdot)$ minimizing $pls_\lambda(h(\cdot))$ is equivalent to finding the vector $\boldsymbol{\theta}$ minimizing $pls_\lambda(\boldsymbol{\theta})$ (being the only unknown, for the moment we take λ as a fixed value we've choosen, al massimo ne adottiamo diversi);
- it is possible to prove that the estimates of our interest for $\boldsymbol{\theta}$
$$\hat{\mathbf{t}}_\lambda = \arg\min_{\mathbf{t} \in \mathbb{R}^n} pls_\lambda(\mathbf{t}) = \left(\mathbf{N}^\top \mathbf{N} + \lambda \mathbf{P}\right)^{-1} \mathbf{N}^\top \mathbf{y}$$
the expression closely reminds of the expression for the ordinary least square estimate (with exception of the $\lambda\mathbf{P}$: when $\lambda = 0$ we have the ordinary least square criterion, the larger λ the greatest the impact)

Proposition 8.1.3. *Therefore the estimated conditional expected values $\hat{h}_\lambda(x_i)$ are, finally, a linear trasformation of* **y** *(premultiplied by a matrix) given by*

$$\begin{bmatrix}\hat{h}_\lambda(x_1)\\ \vdots \\ \hat{h}_\lambda(x_n)\end{bmatrix} = \hat{\mathbf{h}}_\lambda = \mathbf{N}\hat{\mathbf{t}}_\lambda = \underbrace{\mathbf{N}\left(\mathbf{N}^\top\mathbf{N} + \lambda\mathbf{P}\right)^{-1}\mathbf{N}^\top\mathbf{y}}_{\mathbf{S}_\lambda}$$

Remark 39. $\hat{\mathbf{h}}_\lambda$ is an example of linear smoother, obtained using the smoothing matrix $\mathbf{S}_\lambda$. *The subscript λ has been added to emphasize the fact that the values of these estimates depend on the specific value of the smoothing parameter*
$\mathbf{S}_\lambda$ is the matrix that smoothes the observed values $\mathbf{y}$ in order to get the fitted ones

Remark 40. Now we consider how to choose λ.

8.1.6 Choice of the smoothing parameter

Remark 41. In the smoothing spline approach the problem of selecting the number and the location of the knots is bypassed; however the smoothing parameter λ plays a crucial role in governing the goodness of fit and the complexity of the estimated regression function

Remark 42. We have several way to choose λ; we cannot use AIC or BIC because they need a parametric model and likelihood which is not necessarily the case here (we didn't impose it in the requirements regarding shape of distribution/normality etc)

8.1.6.1 Leave one out crossvalidation

We:

- consider a grid of values for the parameters
- fit for each lambda the model avoiding one unit
- calculate the following mean and choose the λ minimizing it

$$LOOCV(\lambda) = \frac{1}{n}\sum_i \left(y_i - \hat{h}_\lambda^{[-i]}(x_i)\right)^2$$

where $\hat{h}_\lambda^{[-i]}(x_i)$ is estimate of $\mathrm{E}\left[Y_i | x_{1i}, \ldots, x_{pi}\right]$ obtained after excluding the i-th unit from the observed sample (*independent from i*).

Even here is possible to prove something similar to what seen before (using the hat matrix, here we use S to transform y into prediction not hat matrix), that is:

$$y_i - \hat{h}_\lambda^{[-i]}(x_i) = \frac{y_i - \hat{h}_\lambda(x_i)}{1 - \mathbf{S}_{\lambda,ii}}$$

where $\mathbf{S}_{\lambda,ii}$ is the i-th element of the main diagonal of $\mathbf{S}_\lambda$ and thus *LOOCV for smoothing splines can be computed without repeating the fitting process n times.*

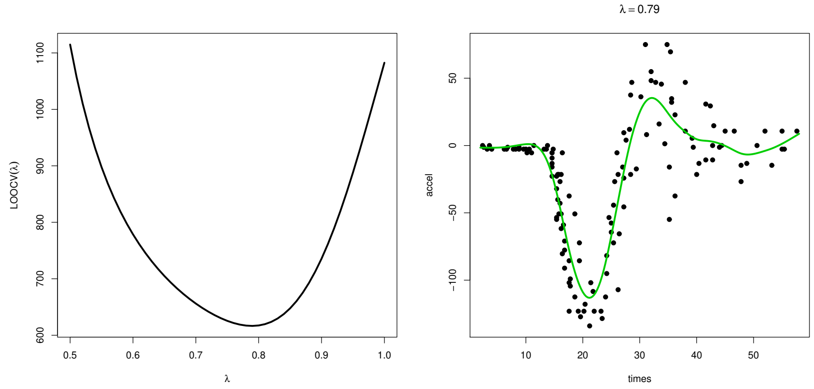
Example 8.1.4 (Crash test data - optimal value for λ - *LOOCV*)**.** In figure 8.3 on the left the plot describing the error as a function of lambda (from 0.5 to 1, selected by trial and error) and on the right the model estimated with best lambda.

8.1.6.2 Generalized Cross-Validation

As an alternative to *LOOCV*, some authors suggest the minimization of the following criterion (which looks like loocv, by involving residual and smoother matrix):

$$GCV(\lambda) = \frac{1}{n}\sum_i \left(\frac{y_i - \hat{h}_\lambda(x_i)}{1 - \dfrac{\mathrm{Tr}(\mathbf{S}_\lambda)}{n}}\right)^2 = \frac{n}{(n - \mathrm{Tr}(\mathbf{S}_\lambda))^2}\sum_i \left(y_i - \hat{h}_\lambda(x_i)\right)^2$$

where $\mathrm{Tr}(\cdot)$ is the trace operator (*sum of the diagonal elements*).
Most interesting properties of the trace of the smoothing matrix are (it is possible to prove):



(a) (b)

Figure 8.3: Splines lambda with loocv

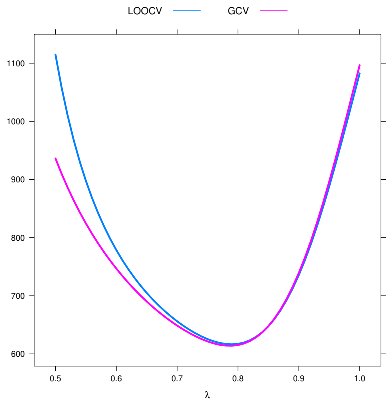


Figure 8.4: Lambda: GCV vs LOOCV

- when $\lambda \to +\infty$ (fitted function become a linear function) $\text{Tr}(\mathbf{S}_\lambda) \to 2$: the trace tends to 2 (2 are free parameters one have in a linear function)
- if $\lambda \to 0$ then $\text{Tr}(\mathbf{S}_\lambda) \to n$ (n is the number of total parameters for the natural cubic spline we're using to represent the function h; and n would be the number of free parameters if we were to fit a natural cubic spline with knots located at the unique values for x)

So $\text{Tr}(\mathbf{S}_\lambda) = \text{edf}_\lambda$ is basically a way of quantifying the effective degrees of freedom of $\hat{h}_\lambda(\cdot)$ (free parameters in the corresponding fitted function: it ranges between the two extremes 2 - the smallest possible number of parameters by restricting our attention to linear function - and otoh n - the maximum number of params we can have by setting $\lambda = 0$ and going for the saturated model).
Although $\hat{h}_\lambda(\cdot)$ depends on n parameters, the presence of the penalty term in the penalized LS criterion imposes some restrictions on the estimated parameters (the effective dimension of $\hat{\mathbf{t}}_\lambda$ is lower than n)

Example 8.1.5 (Crash test data - optimal value for λ - *GCV* vs. *LOOCV*)**.** In figure 8.4 the comparison of error; here there is a substantial agreement between the two criteria on the best lambda.

8.1.7 Penalized estimation final remarks

Important remark 51. Although the theoretical results, the linking of penalized estimation to splines requires:

- least squares as a measure of goodness of fit
- natural cubic splines with knots at unique values of x_i $(i = 1, \dots, n)$
- a penalization term based on second partial derivatives

That is in these circumstances, they arise naturally as an optimal strategy.

Important remark 52. However the idea of *penalized estimation* can be extended beyond smoothing splines using:

- different goodness of fit measures (e. g.: loglikelihood functions not only error);
- other penalization schemes (eg not only roughness)
- cubic spline with $1 << K << n$ knots (not only $K = n$)

Remark 43. These ideas will be explored in the context of p-splines which are just an example.

8.2 P-splines

8.2.1 Gaussian regression models based on P-splines

Important remark 53 (Basic idea). In the context of Gaussian models with cubic regression splines (so expected value can be represented by a cubic spline, heteroskedasticity, non correlation and normality assumption), we can reintroduce the loglikelihood.
An alternative strategy to avoid selection of the number/location of the K knots could be obtained by:

1. choosing a relatively large number of equally spaced knots (eg $K = 20$ or $K = 40$)
2. defining a penalized/regularized log-likelihood function measuring the roughness of the resulting cubic spline (this will depend from the basis expansion used/the parameter in the estimation)

8.2.2 P-splines penalizations

Definition 8.2.1 (P-spline approach)**.** When *B-spline basis* functions are used, two penalty terms based on differences of coefficients are usually introduced.
The idea of using cubic splines represented by use of B-spline basis functions and penalty terms based on differences leads to the so-called P-spline approach (P is used to remember penalization)

Definition 8.2.2 ((Squared) first order difference penalization)**.** It's defined as

$$J_1(\boldsymbol{\theta}) = \sum_{j=2}^{K+4} (\theta_j - \theta_{j-1})^2$$

We compute difference between pairs of consecutive parameters of the $K + 4$ bases we have. The bases have a sort of natural ordering (eg see fig 7.10) since each one take positive value to the right of where the precedent basis; here we exploit this ordering.
The rational behind this penalty term is that it's 0 (theta are all equal among each other) the resulting cubic spline function will be a constant function:

$$J_1(\boldsymbol{\theta}) = 0 \iff \sum_j \theta_j b_j(x) \text{ is constant in } x$$

Infact whatever value of x we consider if we take the sum of the value of the $K + 4$ b-spline bases, it's always equal to 1; if are constant $\theta_j = \theta$ then the linear combination of thetas and basis will be constant as well $(\theta \cdot 1)$.
Otoh the penalty will grow larger and larger as we found greater differences in the θ_j (and the complexity of the resulting function increases).

Definition 8.2.3 ((squared) second-order differences)**.** Defined as

$$J_2(\boldsymbol{\theta}) = \sum_{j=3}^{K+4} [(\theta_j - \theta_{j-1}) - (\theta_{j-1} - \theta_{j-2})]^2 = \sum_{j=3}^{K+4} (\theta_j - 2\theta_{j-1} + \theta_{j-2})^2$$

Here we compare two consecutive pairs of differences between coefficients and its turns out (it's possible to prove) that the minimization of the penalty terms occur not when the function is constant but when it's linear

$$J_2(\boldsymbol{\theta}) = 0 \iff \sum_j \theta_j b_j(x) \text{ is linear in } x$$

As much as this penalty term increases, the more the resulting function will be nonlinear

Important remark 54. So we have two penalty terms designed to deal with slightly different complexity, that take value 0 when the function is "simple" and both tend to increases as the function increases in complexity.

Important remark 55. Both $J_1(\boldsymbol{\theta})$ and $J_2(\boldsymbol{\theta})$ admit a matrix representation

Definition 8.2.4 (Matrix representation)**.**

$$J_1(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{D}_1^\top \mathbf{D}_1 \boldsymbol{\theta} = \boldsymbol{\theta}^\top \mathbf{P}_1 \boldsymbol{\theta}$$

with

$$\mathbf{D}_1 = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 & 1 \end{bmatrix}$$

being a $(K+3) \times (K+4)$ simple matrix composed of only -1, 1 and 0.
The structure of the matrix remeber those from linear hypotheses (comparing two consecutive coefficients): by postmultiplying this for $\boldsymbol{\theta}$ we obtain a vector with the differences of consecutive θ_j, that is $\mathbf{D}_1\boldsymbol{\theta} = \begin{bmatrix} \theta_2 - \theta_1 \\ \theta_3 - \theta_2 \\ \cdots \end{bmatrix}$. If we transpose this we obtain $\boldsymbol{\theta}^\top \mathbf{D}_1^\top$ in the equation; if we further multiplying for itself $\boldsymbol{\theta}^\top \mathbf{D}_1^\top \mathbf{D}_1 \boldsymbol{\theta}$ we obtain the sum of squares of differences between consecutive pairs of θ_j

Definition 8.2.5 (Matrix representation)**.**

$$J_2(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{D}_1^\top \mathbf{D}_2^\top \mathbf{D}_2 \mathbf{D}_1 \boldsymbol{\theta} = \boldsymbol{\theta}^\top \mathbf{P}_2 \boldsymbol{\theta}$$

with

$$\mathbf{D}_2 = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 & 1 \end{bmatrix}$$

being $(K+2) \times (K+3)$ matrix. Here we use both $\mathbf{D}_1$ and $\mathbf{D}_2$: D_1 is used as in the previous example to compute the differences between consecutive θ_j; then is premultiplied for $\mathbf{D}_2$ which structure is somewhat similar to D_1 and is needed to compute the differences between pair of consecutive differences (with the first row we compute the difference $(\theta_3 - \theta_2) - (\theta_2 - \theta_1)$). So

$$\mathbf{D}_2\mathbf{D}_1\boldsymbol{\theta} = \begin{bmatrix} (\theta_3 - \theta_2) - (\theta_2 - \theta_1) \\ (\theta_4 - \theta_3) - (\theta_3 - \theta_2) \\ \cdots \end{bmatrix}$$

The same is done a second time and transposed so we have the square of these differences.

Important remark 56. Point is with these setup we can write $\mathbf{J}_1(\boldsymbol{\theta}), \mathbf{J}_2(\boldsymbol{\theta})$ simply as a quadratic form (respectively $\boldsymbol{\theta}^\top P_1 \boldsymbol{\theta}$ and $\boldsymbol{\theta}^\top P_2 \boldsymbol{\theta}$) as we did for the smoothing spline approach: the content of the penalty matrix $\mathbf{P}_1, \mathbf{P}_2$ now is different and depend on the basis chosen; using b-spline we have simple structure that can be easily computed.
We can use this quadratic form for the penalized estimation.

Remark 44. Now we have all the ingredient needed to come up with the penalized loglik function

8.2.3 Penalized log-likelihood and derived function

Definition 8.2.6 (Penalized log-likelihood)**.**

$$pl_\lambda\big(\boldsymbol{\theta}, \sigma^2\big) \propto -\frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}) - \frac{\lambda}{2}\boldsymbol{\theta}^\top\mathbf{P}\boldsymbol{\theta} \tag{8.1}$$

where the first part is the loglik for the gaussian regression model (same as regression spline), while the second is the penalization part (let be $\mathbf{P} = \mathbf{P}_1$ or $\mathbf{P} = \mathbf{P}_2$, whatever):

- $\boldsymbol{\theta}$ is $(K+4)$-dimensional vector with unknown parameters
- $\mathbf{X}$ is $n \times (K+4)$ matrix containing the values of the $K+4$ B-spline basis evaluated on each sample unit

  $$\mathbf{X} = \begin{bmatrix} b_1(x_1) & \dots & b_{K+4}(x_1) \\ \vdots & \ddots & \vdots \\ b_1(x_n) & \dots & b_{K+4}(x_n) \end{bmatrix}$$

  The subscript distinguishing penalty terms defined by first-order differences and second-order differences has been dropped for the sake of simplicity
- here the λ is divided by 2 just for math convenience: in computing first and second partial derivatives it's convenient to have the same multiplicative factor for all the three terms of the algebric sum (i guess the $\propto$ comes from this)

Important remark 57 (Derived quantities)*.* Some relevant quantities related to $\boldsymbol{\theta}$ derived from $pl_\lambda(\boldsymbol{\theta}, \sigma^2)$ are first and second partial derivatives:

- the penalized score function

  $$U_\lambda(\boldsymbol{\theta}) = \frac{\partial}{\partial\boldsymbol{\theta}} pl_\lambda\big(\boldsymbol{\theta}, \sigma^2\big) = \frac{\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})}{\sigma^2} - \lambda\mathbf{P}\boldsymbol{\theta} = U(\boldsymbol{\theta}) - \lambda\mathbf{P}\boldsymbol{\theta}$$

  The first partial derivative turns out to be the usual stuff plus the first partial derivative of the penalty term; being this last a quadratic it first partial derivatives is twice (which cancel out $\lambda/2$) times something else
- penalized (observed/expected) Fisher information

  $$i_\lambda(\boldsymbol{\theta}) = I_\lambda(\boldsymbol{\theta}) = -\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top} pl_\lambda\big(\boldsymbol{\theta}, \sigma^2\big) = \frac{\mathbf{X}^\top\mathbf{X}}{\sigma^2} + \lambda\mathbf{P} = I(\boldsymbol{\theta}) + \lambda\mathbf{P}$$

  again this is the basic stuff with the only variation of the penalization stuff

8.2.4 Penalized ML estimation

Important remark 58*.* It is possible to prove that *maximizing* the penalized log-likelihood $pl_\lambda\big(\boldsymbol{\theta}, \sigma^2\big)$ with respect to θ is equivalent to *minimizing* the following penalized least squares criterion:

$$(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}) + \tilde{\lambda}\boldsymbol{\theta}^\top\mathbf{P}\boldsymbol{\theta}$$

where $\tilde{\lambda} = \lambda\sigma^2$.
This latter equation was obtained starting from 8.1, ignoring the first term and multiplying all the remaining for $-2\sigma^2$.

Remark 45*.* Note that from a matrix pov, this is just the penalized criterion that we exploited with smoothing spline approach. The main differences are two:

- rather than having a matrix $\mathbf{N}$ containing as many column as the unique values of x and in each column the values associated to the basis we use to represent the natural cubic splines, here we have a matrix $\mathbf{X}$ with (fewer) $K+4$ columns and each column with one of the b-spline basis obtained by considering the knots we've choosen (spread equidistant on the range of x)
- rather than having a matrix $\mathbf{P}$ associated with the integral of the function roughness we have another $\mathbf{P}$ associated with first or second order differences between parameters θ

Important remark 59. So we have estimates and predicted values are somewhat similar to what seen previously:

$$\hat{\mathbf{t}}_{\tilde{\lambda}} = \left(\mathbf{X}^\top\mathbf{X} + \tilde{\lambda}\mathbf{P}\right)^{-1}\mathbf{X}^\top\mathbf{y}$$
$$\hat{\mathbf{h}}_{\tilde{\lambda}} = \mathbf{X}\hat{\mathbf{t}}_{\tilde{\lambda}} = \underbrace{\mathbf{X}\left(\mathbf{X}^\top\mathbf{X} + \tilde{\lambda}\mathbf{P}\right)^{-1}\mathbf{X}^\top}_{\mathbf{S}_{\tilde{\lambda}}}\mathbf{y}$$

with again either $\mathbf{P} = \mathbf{P}_1$ or $\mathbf{P} = \mathbf{P}_2$ and $\tilde{\lambda} = \lambda\sigma^2$.
Furthermore it is possible to prove some of the smoothing matrix, that is when:

$$\tilde{\lambda} = 0 \implies \mathrm{Tr}\left(\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\right) = K + 4$$
$$\tilde{\lambda} \to +\infty \implies \mathrm{Tr}\left(\mathbf{X}\left(\mathbf{X}^\top\mathbf{X} + \tilde{\lambda}\mathbf{P}_1\right)^{-1}\mathbf{X}^\top\right) \to 1$$
$$\implies \mathrm{Tr}\left(\mathbf{X}\left(\mathbf{X}^\top\mathbf{X} + \tilde{\lambda}\mathbf{P}_2\right)^{-1}\mathbf{X}^\top\right) \to 2$$

So if $\lambda = \tilde{\lambda} = 0$ the trace is the actual number of parameters (if we ignore the penalty term we end up fitting a cubic spline with $K + 4$ free parameters).
As $\tilde{\lambda} \to 0$ the trace converges to 1 for the first difference and 2 for the second differences (number of free parameter)
So the trace goes from the maximum number of free parameters to the minimum number of free parameters as lambda goes from 0 to ∞; so the trace can be used to measure the actual complexity of the fitted function

Example 8.2.1 (Crash test data - P-splines - penalized ML estimation)**.** In fig 8.5:

- in (a,b,c) we consider a fixed number of knot $K = 20$ (at max 24 parameter for level of complexity) with penalty $\mathbf{P}_1$ (*squared first-order differences*): we have that $\hat{h}_{\tilde{\lambda}}(\cdot)$ approaches a constant as $\tilde{\lambda} \to +\infty$
- in (d,e,f) we consider $K = 20$, $\mathbf{P}_2$ penalty (*squared second-order differences*): here ignoring the penalization with $\lambda = 0$ (case d) gives the exactly same results as above (case a, just a cubic spline with 24 free parameters) but as $\tilde{\lambda} \to +\infty$ we have that $\hat{h}_{\tilde{\lambda}}(\cdot)$ approaches a linear function

Man this shit is soo flexible for any kind of shape

8.2.5 Estimation of σ^2

An estimate of σ^2 can be obtained mimicking what we've done in gaussian model (dividing the sum of squares of residuals by $n-$ numbers of parameters $p+1$, which here can be represented by the trace of the smoothing matrix), that is using the following expression:

$$s^2 = \frac{\sum_i \left(y_i - \hat{h}_{\tilde{\lambda}}(x_i)\right)^2}{n - \mathrm{Tr}\left(\mathbf{S}_{\tilde{\lambda}}\right)}$$

Some authors suggest replacing the effective degrees of freedom $\mathrm{Tr}\left(\mathbf{S}_{\tilde{\lambda}}\right)$ with the equivalent number of parameters: $2\mathrm{Tr}\left(\mathbf{S}_{\tilde{\lambda}}\right) - \mathrm{Tr}\left(\mathbf{S}_{\tilde{\lambda}}\mathbf{S}_{\tilde{\lambda}}\right)$

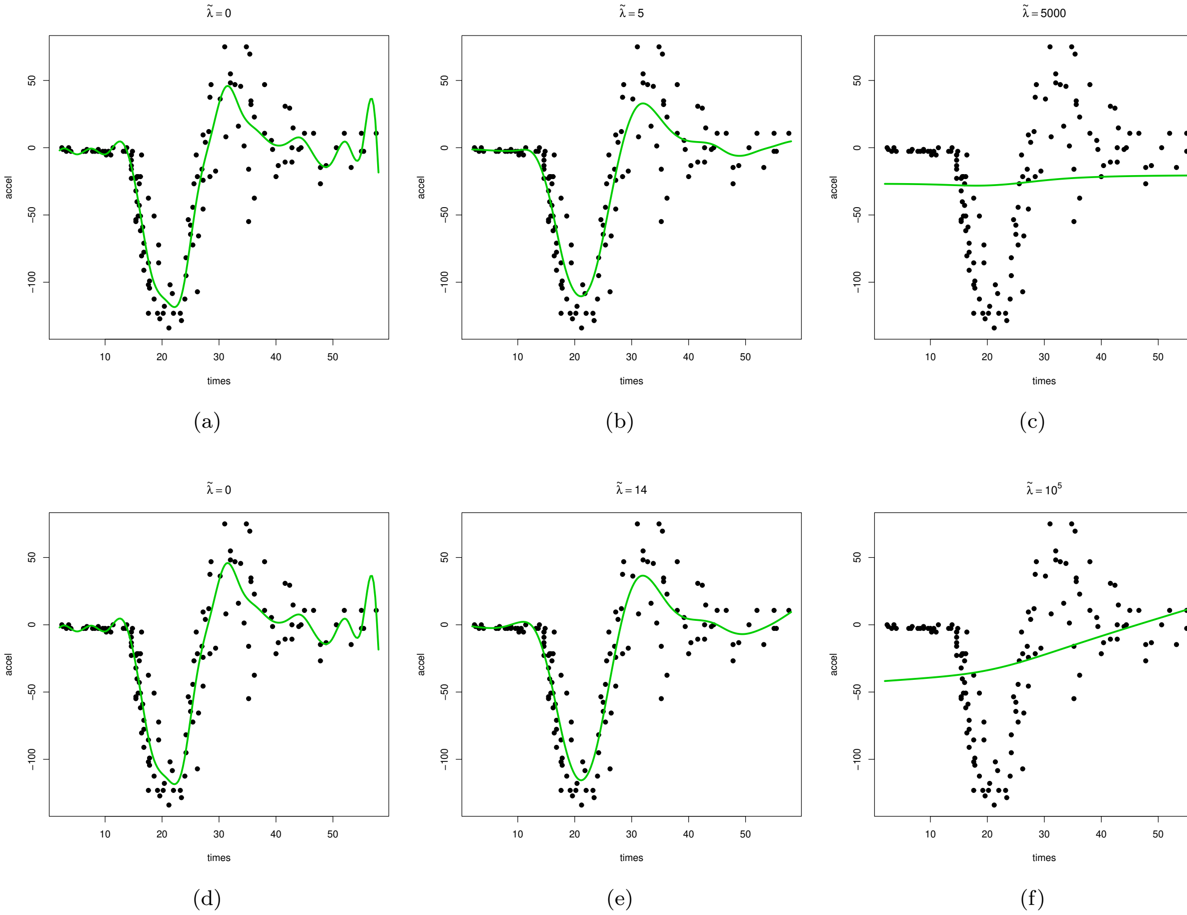
Remark 46. In both cases, when $\lambda = 0$ the trace and this second approach will end with $K+4$ as complexity

8.2.6 Choice of the smoothing parameter

Remark 47. The selection of the optimal value for $\tilde{\lambda}$ can be based on a model selection criterion such as *LOOCV* (it is not necessary to refit the model n times), *GCV* (generalized cv) or here we can use *AIC* or *BIC* because we have a parametric definition and a likelihood; in these cases:

- the maximum likelihood estimate for σ^2 is needed (biased but asymptotically unbiased):

$$\hat{s}^2 = \frac{1}{n}\sum_i \left(y_i - \hat{h}_{\tilde{\lambda}}(x_i)\right)^2$$



(a)

(b)

(c)

(d)

(e)

(f)

Figure 8.5: Psplines

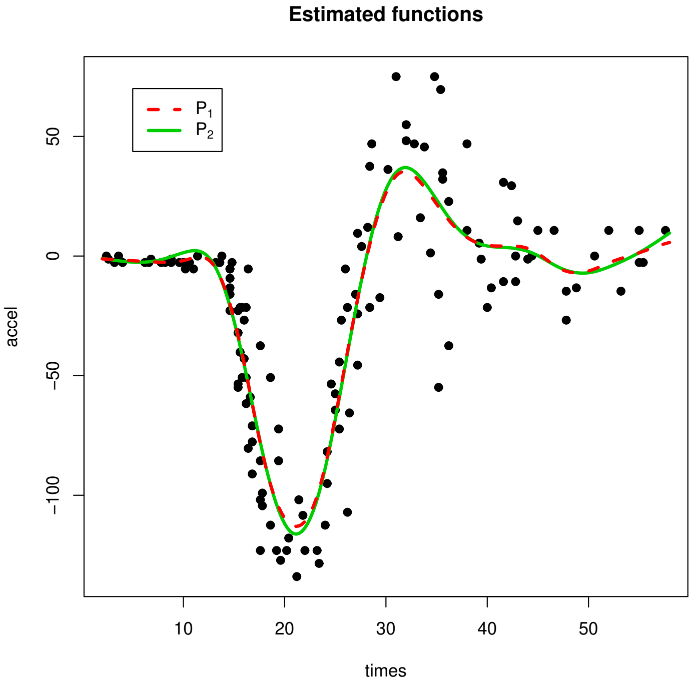


Figure 8.6

- the number of parameter can be obtained as

$$\operatorname{Tr}(\mathbf{S}_{\tilde{\lambda}}) + 1$$

  (complexity $+1$ for σ^2)

Example 8.2.2 (Crash test data - P-splines - optimal $\tilde{\lambda}$)**.** In table both the GCV criterion and the AIC for the best within first order and second order penalization

Penalty	K	$\tilde{\lambda}$	$\operatorname{Tr}(\mathbf{S}_{\tilde{\lambda}})$	GCV	AIC
first-order diff.	20	3.369	12.275	569.540	1222.092
second-order diff.	20	12.182	11.414	562.227	1220.542

Then the resulting estimates are presented in figure8.6; both are very similar.

- with the second order difference we end with a slightly less complex model (trace/number of params is lower)
- if we have to choose one, we go with the second order both in term of GCV or AIC

8.2.7 Inference

Important remark 60 (Starting point)*.* We start from our full model assumptions:

$$\boldsymbol{Y}|\mathbf{X} \sim MVN_n(\mathbf{h}, \sigma^2\mathbf{I}_n)$$

where

- the conditional expected value $\mathbf{h} = (h(x_1), \ldots, h(x_n))^\top$ is an n-dimensional vector obtained using an unknown function h
- we have the usual homoskedasticity uncorrelation assumption (look the variance covariance matrix)
- the usual normality assumption

If for example we choose to approximate h using a cubic spline and we use a p-spline approach to control the actual complexity of the estimated function we can come up with estimates for the θ_j parameters.

Important remark 61 (Our focus). Usually, in the context of nonlinear regression, the interest is in the function $\hat{h}(\cdot)$ as a whole rather than in single parameters θ_j (which is related to the specific choice of the basis which is just instrumental/not of interest; here the parameters are not associated with relation between variables and do not have a practical meaningful interpretation). We're not strictly interested in inference on the θ_j but the interest is o the behaviour of the estimated function, so the $\hat{\boldsymbol{h}}_{\tilde{\lambda}}$.

Important remark 62 (Distributions). What can we say of the sampling properties/distribution of the estimated function $\hat{\boldsymbol{h}}_{\tilde{\lambda}}|\mathbf{X}$? $\hat{\boldsymbol{h}}_{\tilde{\lambda}}$ are a linear trasformation of the Y obtained by premultiply it for the smoother matrix; this means that they will inherit properties due to the gaussianity

$$\hat{\boldsymbol{h}}_{\tilde{\lambda}}|\mathbf{X} = \mathbf{S}_{\tilde{\lambda}}\boldsymbol{Y}|\mathbf{X} \sim MVN_n\left(\mathbf{S}_{\tilde{\lambda}}\mathbf{h}, \sigma^2\mathbf{S}_{\tilde{\lambda}}\mathbf{S}_{\tilde{\lambda}}\right) \tag{8.2}$$

Important remark 63 (Bias). In general:

- we have that:

  $$\mathrm{E}\left[\hat{\boldsymbol{h}}_{\tilde{\lambda}}|\mathbf{X}\right] = \mathbf{S}_{\tilde{\lambda}}\mathbf{h} \neq \mathbf{h}$$

  So P-splines are biased estimators for the unknown function $h(\cdot)$

  $$\mathbf{h} - \mathbf{S}_{\tilde{\lambda}}\mathbf{h} \neq \mathbf{0}_n$$

- this bias is due both to the *constraints* implicitly imposed by the penalty term and to the fact that *splines are used as an approximation* to $h(\cdot)$;
- usually the bias is small/negligible, especially when we have a large sample size (as sample size increases is possible to prove that the bias vanishes)

Remark 48. however the distributional result in 8.2 can be exploited to draw approximate inferential conclusions about $h(\cdot)$

8.2.8 Hypothesis testing

Remark 49. The approximate distributional results for $\hat{\mathbf{h}}_\lambda$ can be exploited to test some hypothesis on $h(\cdot)$.

Important remark 64. Which kind of hypotheses are meaningful in the context of nonlinear/penalized regression?

- *independence* assumption (we test if h is constant and so there's no relation between x and y):

  $$H_0\colon\ h(\cdot) \text{ is a constant function}$$

- *linearity* assumption (basically we test if a nonlinear/complex structure is absolutely needed)

  $$H_0\colon\ h(\cdot) \text{ is a linear function}$$

  So aside from graphical visualization we can use more formal test to check if the linearity assumption of the gaussian model hold or not

Now:

- to test this hypothesis we can use the the fact that cubic splines admit constant and linear functions as special cases (are nested in cubic splines, easier to think of in truncated power basis than b-splines): so it is possible to perform an approximate likelihood ratio test/Wald test statistic (usually resulting in approximate F test), after expressing these hypothesis in terms of linear restrictions on the cubic spline coefficients (omitted) and so using the general linear hypothesis test framework (in other words we compare the found model with the restricted one, costant or linear, model);
- WARNING: the resulting p-values rely on several approximations and do not take into account the uncertainty related to the choice of the smoothing parameter λ (different sample might lead to different optimal value in the smoothing parameter, trace of smoother matrix and complexity); in particular, they tend to underestimate (be smaller than) the actual p-values (anticonservative)

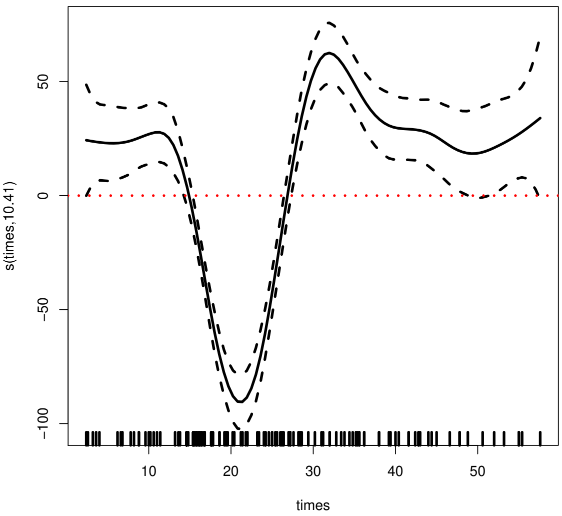


Figure 8.7: Crash test data - P-splines

- it is also possible to derive approximate pointwise confidence bands for $h(\cdot)$

Example 8.2.3 (Crash test data - P-splines - R output)**.** Using `gam` function from package `mgcv` to fit p-splines with second-order differences penalty and $K = 20$ we obtain (these are the test from the model of second-order-diff in example 8.2.2) `Parametric coefficients:`

```
               Estimate  Std. Error  t value  Pr(>|t|)
 (Intercept)    -25.546       1.966  -12.995     0.000
Approximate significance of smooth terms:

               edf   Ref.df       F   p-value
 s(times)   10.414   12.438  37.198     0.000
R-sq.(adj) = 0.78 Deviance explained = 79.7 GCV = 562.23 Scale est.  = 513.98 n = 133
```

The estimated function $\hat{h}_{\tilde{\lambda}}$ is decomposed in two parts:

- first an intercept/constant (⇒ 1 degrees of freedom)
- second a (nonconstant) function (`s(times)`) centred around 0; according to the approximate p-value (based on the null hypothesis that `s` function is constant and equal to 0 so the resulting h function is equal to a constant intercept and nothing else: here we test the independence hypothesis), one may conclude that the estimated function *differ significantly from a constant one*
- in the s(times) part we have $\text{Tr}(\mathbf{S}_{\tilde{\lambda}}) - 1$ effective degrees of freedom `edf` (-1 because 1 is assigned to the intercept)

TODO: non chiaro independence hypothesis e second order differences penalty non ci sono relazioni?

The distributional results we've found can be used as in figure 8.7 to draw confidence bands (associated to the (centred) estimated function): the fact that confidence bands do not include neither a constant function nor a linear one is evidence that for this specific dataset we need to resort on nonlinear regression stuff (particularly constant line at 0 is not contained in the confidence band, consistently with the inferential conclusion from the tests)

Chapter 9

Lab 2 - flexible gaussian regression models

We use the `mcycle` dataset from `MASS` (same as the theorethical part) where `times` is independent and `accel` is dependent

```
library(MASS)
data(mcycle)
str(mcycle) # ?mcycle

## 'data.frame': 133 obs. of  2 variables:
##  $ times: num  2.4 2.6 3.2 3.6 4 6.2 6.6 6.8 7.8 8.2 ...
##  $ accel: num  0 -1.3 -2.7 0 -2.7 -2.7 -2.7 -1.3 -2.7 -2.7 ...
```

9.1 Polynomial regression

Including polynomials is easily done with `poly` (see `?poly`)specifying the degree (eg cubic function we specify 3);

```
head(raw <- poly(1:5, degree = 3, raw = TRUE))  # standard powers

##      1  2   3
## [1,] 1  1   1
## [2,] 2  4   8
## [3,] 3  9  27
## [4,] 4 16  64
## [5,] 5 25 125

head(orth <- poly(1:5, degree = 3, raw = FALSE)) # orthogonal powers

##                  1          2             3
## [1,] -6.324555e-01  0.5345225 -3.162278e-01
## [2,] -3.162278e-01 -0.2672612  6.324555e-01
## [3,] -3.510833e-17 -0.5345225  1.755417e-16
## [4,]  3.162278e-01 -0.2672612 -6.324555e-01
## [5,]  6.324555e-01  0.5345225  3.162278e-01

## they differs by correlation
cor(raw)
```

```
##           1         2         3
## 1 1.0000000 0.9811049 0.9431175
## 2 0.9811049 1.0000000 0.9892158
## 3 0.9431175 0.9892158 1.0000000
```

```
cor(orth)
```

```
##              1             2             3
## 1 1.000000e+00  3.957338e-18  1.355253e-20
## 2 3.957338e-18  1.000000e+00 -4.824700e-18
## 3 1.355253e-20 -4.824700e-18  1.000000e+00
```

We typically don’t want standard/raw polynimials for both difficulties in estimation and impact on variance of coefficients.
The `poly` function can be used in a formula; now we fit a polynomial of order 12, both raw (simple polynomial) and orthogonal and note that the two models are characterized by the same summary statistics, but by different coefficients (due to the differences in the bases)

```
summary(poly12raw <- lm(accel ~ poly(times, degree = 12, raw = TRUE), data = mcycle)) ### raw powers
```

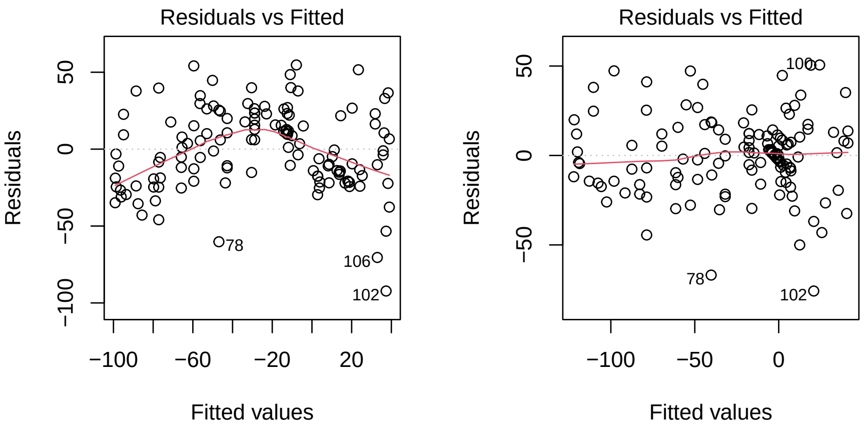
```
##
## Call:
## lm(formula = accel ~ poly(times, degree = 12, raw = TRUE), data = mcycle)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.781 -12.284   1.046  11.995  50.613
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            2.386e+02  3.768e+02   0.633  0.52777
## poly(times, degree = 12, raw = TRUE)1  -2.950e+02  3.488e+02  -0.846  0.39932
## poly(times, degree = 12, raw = TRUE)2   1.440e+02  1.285e+02   1.121  0.26445
## poly(times, degree = 12, raw = TRUE)3  -3.674e+01  2.543e+01  -1.445  0.15106
## poly(times, degree = 12, raw = TRUE)4   5.485e+00  3.063e+00   1.791  0.07589
## poly(times, degree = 12, raw = TRUE)5  -5.106e-01  2.401e-01  -2.126  0.03552
## poly(times, degree = 12, raw = TRUE)6   3.085e-02  1.271e-02   2.426  0.01673
## poly(times, degree = 12, raw = TRUE)7  -1.239e-03  4.627e-04  -2.677  0.00847
## poly(times, degree = 12, raw = TRUE)8   3.332e-05  1.159e-05   2.875  0.00478
## poly(times, degree = 12, raw = TRUE)9  -5.930e-07  1.961e-07  -3.024  0.00305
## poly(times, degree = 12, raw = TRUE)10  6.701e-09  2.140e-09   3.131  0.00219
## poly(times, degree = 12, raw = TRUE)11 -4.353e-11  1.359e-11  -3.202  0.00175
## poly(times, degree = 12, raw = TRUE)12  1.239e-13  3.817e-14   3.245  0.00152
##
## (Intercept)
## poly(times, degree = 12, raw = TRUE)1
## poly(times, degree = 12, raw = TRUE)2
## poly(times, degree = 12, raw = TRUE)3
## poly(times, degree = 12, raw = TRUE)4  .
## poly(times, degree = 12, raw = TRUE)5  *
## poly(times, degree = 12, raw = TRUE)6  *
## poly(times, degree = 12, raw = TRUE)7  **
## poly(times, degree = 12, raw = TRUE)8  **
## poly(times, degree = 12, raw = TRUE)9  **
## poly(times, degree = 12, raw = TRUE)10 **
## poly(times, degree = 12, raw = TRUE)11 **
## poly(times, degree = 12, raw = TRUE)12 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.67 on 120 degrees of freedom
## Multiple R-squared:  0.7998,Adjusted R-squared:  0.7798
## F-statistic: 39.96 on 12 and 120 DF,  p-value: < 2.2e-16

summary(poly12 <- lm(accel ~ poly(times, degree = 12), data = mcycle)) ### orthogonal polynomials

##
## Call:
## lm(formula = accel ~ poly(times, degree = 12), data = mcycle)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.781 -12.284   1.046  11.995  50.613
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -25.546      1.966 -12.993  < 2e-16 ***
## poly(times, degree = 12)1   164.557     22.674   7.257 4.26e-11 ***
## poly(times, degree = 12)2   131.227     22.674   5.788 5.82e-08 ***
## poly(times, degree = 12)3  -239.790     22.674 -10.576  < 2e-16 ***
## poly(times, degree = 12)4    -6.738     22.674  -0.297 0.766859
## poly(times, degree = 12)5   245.799     22.674  10.841  < 2e-16 ***
## poly(times, degree = 12)6   -83.906     22.674  -3.701 0.000326 ***
## poly(times, degree = 12)7  -153.596     22.674  -6.774 4.94e-10 ***
## poly(times, degree = 12)8   163.064     22.674   7.192 5.97e-11 ***
## poly(times, degree = 12)9    31.879     22.674   1.406 0.162319
## poly(times, degree = 12)10 -141.518     22.674  -6.241 6.77e-09 ***
## poly(times, degree = 12)11   24.240     22.674   1.069 0.287191
## poly(times, degree = 12)12   73.586     22.674   3.245 0.001520 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.67 on 120 degrees of freedom
## Multiple R-squared:  0.7998,Adjusted R-squared:  0.7798
## F-statistic: 39.96 on 12 and 120 DF,  p-value: < 2.2e-16
```

We can compare nested models (say if 12 degrees are needed or 9 are sufficient): we note that the choice of the bases does not alter the results of the F test which is the same. We have that 12 are needed becaus there's significant difference between the twos (significant worsening from 12 to 9 in the model)

```
## comparisons between nested models
poly9raw <- lm(accel ~ poly(times, degree = 9, raw = TRUE), data = mcycle)
poly9 <- lm(accel ~ poly(times, degree = 9), data = mcycle)
anova(poly9raw, poly12raw)

## Analysis of Variance Table
##
## Model 1: accel ~ poly(times, degree = 9, raw = TRUE)
## Model 2: accel ~ poly(times, degree = 12, raw = TRUE)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    123 87723
## 2    120 61693  3     26030 16.877 3.281e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(poly9, poly12)

## Analysis of Variance Table
##
## Model 1: accel ~ poly(times, degree = 9)
## Model 2: accel ~ poly(times, degree = 12)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    123 87723
## 2    120 61693  3     26030 16.877 3.281e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## same could have be done with lht, obv
```

Infact by looking at residuals vs fitted for `poly9raw` we see the presence of non linearity yet, where for order 12 the pattern is basically absent (a polynomial of order 9 is not enough to catch the specific nonlinearity in the relation between x and y in the dataset)

```
par(mfrow = c(1,2))
plot(poly9raw, which = 1)
plot(poly12raw, which = 1)
```



The fine property of orthogonal polynomials is that when using them, the exclusion of some bases does not alter the estimates for the regression coeffficients associated with the remaining bases (this is dued to the absence of correlation)

```
### first 10 estimated regression coefficients are the same
round(coefficients(poly9), 4)

##              (Intercept) poly(times, degree = 9)1 poly(times, degree = 9)2
##                 -25.5459                 164.5566                 131.2271
## poly(times, degree = 9)3 poly(times, degree = 9)4 poly(times, degree = 9)5
##                -239.7898                  -6.7378                 245.7987
## poly(times, degree = 9)6 poly(times, degree = 9)7 poly(times, degree = 9)8
##                 -83.9062                -153.5964                 163.0643
## poly(times, degree = 9)9
##                  31.8789

round(coefficients(poly12), 4)[1:10]

##               (Intercept) poly(times, degree = 12)1 poly(times, degree = 12)2
##                  -25.5459                  164.5566                  131.2271
## poly(times, degree = 12)3 poly(times, degree = 12)4 poly(times, degree = 12)5
##                 -239.7898                   -6.7378                  245.7987
## poly(times, degree = 12)6 poly(times, degree = 12)7 poly(times, degree = 12)8
##                  -83.9062                 -153.5964                  163.0643
## poly(times, degree = 12)9
##                   31.8789
## this does not happen for raw polynomials
round(coefficients(poly9raw), 4)

##                          (Intercept) poly(times, degree = 9, raw = TRUE)1
##                             346.6858                            -228.7551
## poly(times, degree = 9, raw = TRUE)2 poly(times, degree = 9, raw = TRUE)3
##                              49.6261                              -4.8113
## poly(times, degree = 9, raw = TRUE)4 poly(times, degree = 9, raw = TRUE)5
##                               0.2271                              -0.0049
## poly(times, degree = 9, raw = TRUE)6 poly(times, degree = 9, raw = TRUE)7
##                               0.0000                               0.0000
## poly(times, degree = 9, raw = TRUE)8 poly(times, degree = 9, raw = TRUE)9
##                               0.0000                               0.0000

round(coefficients(poly12raw), 4)[1:10]

##                           (Intercept) poly(times, degree = 12, raw = TRUE)1
##                              238.5981                             -294.9920
## poly(times, degree = 12, raw = TRUE)2 poly(times, degree = 12, raw = TRUE)3
##                              144.0253                              -36.7427
## poly(times, degree = 12, raw = TRUE)4 poly(times, degree = 12, raw = TRUE)5
##                                5.4849                               -0.5106
## poly(times, degree = 12, raw = TRUE)6 poly(times, degree = 12, raw = TRUE)7
##                                0.0308                               -0.0012
## poly(times, degree = 12, raw = TRUE)8 poly(times, degree = 12, raw = TRUE)9
##                                0.0000                                0.0000
```

9.2 Regression splines

Remark 50. In R we don't have a truncated power basis expansion because that set of bases has large correlation issues among the value of each bases

9.2.1 B-splines

The b-spline bases expansion are done via `bs` in the `splines` package (it's installed by default, see `?bs`); to get the basis we provide the `degree` and the position of the knots.

```
library(splines)
head(bs(1:5, knots = c(2, 4), degree = 1))

##        1   2 3
## [1,] 0.0 0.0 0
## [2,] 1.0 0.0 0
## [3,] 0.5 0.5 0
## [4,] 0.0 1.0 0
## [5,] 0.0 0.0 1

head(bs(1:5, knots = c(2, 4), degree = 2))

##              1         2         3 4
## [1,] 0.0000000 0.0000000 0.0000000 0
## [2,] 0.6666667 0.3333333 0.0000000 0
## [3,] 0.1666667 0.6666667 0.1666667 0
## [4,] 0.0000000 0.3333333 0.6666667 0
## [5,] 0.0000000 0.0000000 0.0000000 1
```

The bases here does not all sums to one because of `intercept = FALSE` by default (this because by being used in lm, we typically don't want the basis to provide an intercept). However let's try:

- linear spline (`degree = 1`) with $K = 9$ knots (10 intervals) located at the quantiles
- cubic spline with $K = 8$ knots at the quantiles

```
## linear spline (9 knots + 1 spline degree + 1 intercept = 11 coefficient)
K.l     <- 9
(knots.l <- quantile(mcycle$times, probs = (1:K.l)/(K.l + 1)))

##   10%   20%   30%   40%   50%   60%   70%   80%   90% 
## 10.04 14.68 16.20 18.44 23.40 26.52 31.52 36.20 43.80

summary(lspline <- lm(accel ~ bs(times, knots = knots.l, degree = 1), data = mcycle))

## 
## Call:
## lm(formula = accel ~ bs(times, knots = knots.l, degree = 1), 
##     data = mcycle)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -73.344 -11.889  -0.575  11.153  53.567 
## 
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                               -1.5222    10.5720  -0.144 0.885747
## bs(times, knots = knots.l, degree = 1)1   -0.4519    15.1657  -0.030 0.976278
## bs(times, knots = knots.l, degree = 1)2   -9.0326    12.5879  -0.718 0.474397
## bs(times, knots = knots.l, degree = 1)3  -48.7028    12.6269  -3.857 0.000185
## bs(times, knots = knots.l, degree = 1)4 -102.4070    13.3444  -7.674 4.57e-12
## bs(times, knots = knots.l, degree = 1)5 -115.9931    14.0102  -8.279 1.83e-13
## bs(times, knots = knots.l, degree = 1)6  -23.8462    12.8728  -1.852 0.066378
## bs(times, knots = knots.l, degree = 1)7   54.0932    14.2669   3.792 0.000234
## bs(times, knots = knots.l, degree = 1)8   10.6931    13.1640   0.812 0.418203
## bs(times, knots = knots.l, degree = 1)9    1.5024    12.9866   0.116 0.908090
## bs(times, knots = knots.l, degree = 1)10   0.9684    15.4128   0.063 0.950003
##                                            
## (Intercept)                                
## bs(times, knots = knots.l, degree = 1)1    
## bs(times, knots = knots.l, degree = 1)2    
## bs(times, knots = knots.l, degree = 1)3  ***
## bs(times, knots = knots.l, degree = 1)4  ***
## bs(times, knots = knots.l, degree = 1)5  ***
## bs(times, knots = knots.l, degree = 1)6  .  
## bs(times, knots = knots.l, degree = 1)7  ***
## bs(times, knots = knots.l, degree = 1)8    
## bs(times, knots = knots.l, degree = 1)9    
## bs(times, knots = knots.l, degree = 1)10   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 22.4 on 122 degrees of freedom
## Multiple R-squared:  0.8014,Adjusted R-squared:  0.7851 
## F-statistic: 49.22 on 10 and 122 DF,  p-value: < 2.2e-16

## cubic spline (8 knots + 3 degree + 1 intercept are 12 coefficients)
K.c <- 8
knots.c <- quantile(mcycle$times, probs = (1:K.c)/(K.c+1))
cspline <- lm(accel ~ bs(times, knots = knots.c, degree = 3), data = mcycle)
summary(cspline)

## 
## Call:
## lm(formula = accel ~ bs(times, knots = knots.c, degree = 3), 
##     data = mcycle)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -74.384 -11.040  -0.056  11.560  52.830
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                 -1.749     14.625  -0.120 0.905000
## bs(times, knots = knots.c, degree = 3)1      4.228     37.658   0.112 0.910789
## bs(times, knots = knots.c, degree = 3)2    -11.359     24.404  -0.465 0.642434
## bs(times, knots = knots.c, degree = 3)3     13.580     20.002   0.679 0.498488
## bs(times, knots = knots.c, degree = 3)4    -84.830     17.155  -4.945 2.48e-06
## bs(times, knots = knots.c, degree = 3)5   -132.878     22.153  -5.998 2.13e-08
## bs(times, knots = knots.c, degree = 3)6    -94.175     18.835  -5.000 1.96e-06
## bs(times, knots = knots.c, degree = 3)7     74.754     20.195   3.702 0.000324
## bs(times, knots = knots.c, degree = 3)8      2.639     19.839   0.133 0.894388
## bs(times, knots = knots.c, degree = 3)9     10.840     24.738   0.438 0.662025
## bs(times, knots = knots.c, degree = 3)10   -19.858     29.590  -0.671 0.503421
## bs(times, knots = knots.c, degree = 3)11    15.108     23.786   0.635 0.526526
##
## (Intercept)
## bs(times, knots = knots.c, degree = 3)1
## bs(times, knots = knots.c, degree = 3)2
## bs(times, knots = knots.c, degree = 3)3
## bs(times, knots = knots.c, degree = 3)4  ***
## bs(times, knots = knots.c, degree = 3)5  ***
## bs(times, knots = knots.c, degree = 3)6  ***
## bs(times, knots = knots.c, degree = 3)7  ***
## bs(times, knots = knots.c, degree = 3)8
## bs(times, knots = knots.c, degree = 3)9
## bs(times, knots = knots.c, degree = 3)10
## bs(times, knots = knots.c, degree = 3)11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.5 on 121 degrees of freedom
## Multiple R-squared:  0.8012,Adjusted R-squared:  0.7831
## F-statistic: 44.33 on 11 and 121 DF,  p-value: < 2.2e-16
```

Classical b-splines vs R's one As example using the linear splines we get different stuff by messing around with the intercept: it seems r square increases: the difference is dued to the fact that R will mess up when removing intercept by thinking regressors are centered even if they arent

```
summary(lspline2 <- lm(accel ~ bs(times, knots = knots.l, degree = 1, intercept = TRUE) - 1, data = mcycle))

##
## Call:
## lm(formula = accel ~ bs(times, knots = knots.l, degree = 1, intercept = TRUE) -
##     1, data = mcycle)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.344 -11.889  -0.575  11.153  53.567
##
## Coefficients:
##                                                            Estimate
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)1    -1.52225
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)2    -1.97414
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)3   -10.55488
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)4   -50.22500
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)5  -103.92921
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)6  -117.51533
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)7   -25.36843
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)8    52.57096
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)9     9.17081
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)10   -0.01986
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)11   -0.55383
##                                                             Std. Error t value
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)1    10.57201  -0.144
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)2     8.18618  -0.241
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)3     7.26903  -1.452
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)4     6.81826  -7.366
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)5     8.16317 -12.731
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)6     9.18759 -12.791
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)7     7.34609  -3.453
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)8     9.57955   5.488
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)9     7.84378   1.169
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)10    7.54213  -0.003
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)11   11.21545  -0.049
##                                                             Pr(>|t|)
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)1  0.885747
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)2  0.809840
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)3  0.149059
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)4  2.28e-11 ***
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)5   < 2e-16 ***
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)6   < 2e-16 ***
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)7  0.000762 ***
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)8  2.25e-07 ***
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)9  0.244609
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)10 0.997904
## bs(times, knots = knots.l, degree = 1, intercept = TRUE)11 0.960697
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.4 on 122 degrees of freedom
## Multiple R-squared:  0.845,Adjusted R-squared:  0.831
## F-statistic: 60.47 on 11 and 122 DF,  p-value: < 2.2e-16

## If we look at the fitted value, they provide exactly the same
## fitted values
head(cbind(fitted(lspline), fitted(lspline2)))

##        [,1]      [,2]
## 1 -1.522246 -1.522246
## 2 -1.534076 -1.534076
## 3 -1.569564 -1.569564
## 4 -1.593224 -1.593224
## 5 -1.616883 -1.616883
## 6 -1.747009 -1.747009
```

If we use bs do use the the default `intercept = FALSE` to get a bspline bases slighlty altered to account for the intercept by default added in an lm; otherwise to exploit the classic/exact definition of bspline we do have to remove the `intercept = TRUE` and removing from the R formula. In R is convenient to use these alternative splines

9.2.2 Smoothing splines

Continue with the approach that does not impose a specific parametric structure of conditional expected value Y|X but ask that the corresponding function is contionuous with regular partial derivatives up to the second order, with square second one integrable. If we try to minimize the pls criterion (sum of squares with roughness) it can be proved that minimizer function

is actually a natural cubic spline (has a specific parametric structure) with knots at unique values for the regressors.
The smoothing splines via `smooth.spline` in `stats` package does all this (without need to tell where to locate the knots etc). The main arguments:

- we just give the vector containing the value of the regressor `x` and the vector containing the dependent variable `y`
- `all.knots = FALSE` if we set to `TRUE` we get the genuine definition with knots located at all the distinct point of x
- by default, the optimal value of the smoothing parameter is selected according to GCV (generalized cross validation); if one want the LOOCV criterion set `cv = TRUE`
- otherwise if one want to specify the smoothing parameter:
  - the argument `lambda` does not correspond to the actual smoothing parameter lambda discussed in class (it's a trasformation of the smoothing parameter)
  - `spar` is the actual smoothing parameter (our lambda in class)

```
## fit the classic smooth spline (genuine definition with all distinct
## values) in our example (we don't specify any value for the
## smoothing parameter so it's selected by GCV)
(sspline <- smooth.spline(x = mcycle$times, y = mcycle$accel, all.knots = TRUE))

## Call:
## smooth.spline(x = mcycle$times, y = mcycle$accel, all.knots = TRUE)
##
## Smoothing Parameter  spar= 0.7670834  lambda= 0.000110663 (12 iterations)
## Equivalent Degrees of Freedom (Df): 12.2553
## Penalized Criterion (RSS): 38606.57
## GCV: 565.4861
```

we find that:

- the best $\lambda = 0.76$ (value of λ minimizing the crossvalidation criterion)
- the value of the (prediction) CV error minimized to choose lambda is 565
- the equivalent degrees of freedom (trace of the smoothing matrix, actual level of complexity of fitted function) is 12.2
- the minimized function once obtained lambda (the penalized rss) is 38606

Now we see some other stuff

```
## smoothing spline with fixed smoothing parameter (say inferior
## penalization so the resulting complexity/degrees of freedom
## increases; at the same time the cv error increases because we're
## far from the optimal 0.7)
(sspline1 <- smooth.spline(mcycle$times, mcycle$accel, all.knots = TRUE, spar = 0.1))

## Call:
## smooth.spline(x = mcycle$times, y = mcycle$accel, spar = 0.1,
##     all.knots = TRUE)
##
## Smoothing Parameter  spar= 0.1  lambda= 1.677848e-09
## Equivalent Degrees of Freedom (Df): 88.22152
## Penalized Criterion (RSS): 1094.118
## GCV: 1623.464

## otherwise we can obtain smoothing spline with (approximate) fixed
## complexity/ Equivalent/Effective degrees of freedom (fixed trace
## for the smoothing parameter) by setting df. Eg setting to 2 we get
## approximately the equivalent of a linear function/regression model
(sspline2 <- smooth.spline(mcycle$times, mcycle$accel, all.knots = TRUE, df = 2))
```

```
## Call:
## smooth.spline(x = mcycle$times, y = mcycle$accel, df = 2, all.knots = TRUE)
##
## Smoothing Parameter  spar= 1.499963  lambda= 21.83207 (31 iterations)
## Equivalent Degrees of Freedom (Df): 2.010935
## Penalized Criterion (RSS): 257182.4
## GCV: 2174.768

## df is not exactly the same, it's approximate
```

9.2.3 P-splines

a crossover between smoothing and regression splines: we can mix basis and penalty used here.
A generic function that can fit psplines is `gam` in the `mgcv` (standard) package:

- it fits generalized additive models (we'll come back at the very end on this) using splines and penalized maximum likelihood approach
- in this function as well, by default the smoothing parameters are selected automatically, but the user has the option to set them to pre-specified values

We here see a subset of functionality and especially how to use it to get psplines: penalized cubic splines where the splines are represented using b-splines and the penalty term is one of the two we've seen.
The function works more or less like `glm`, `family=gaussian`:

- the `formula` param works similarly to glm with one difference (see `?formula.gam`): we have to specify for which regression we want to use the flexible spline to represent the impact; we put these variables within the `s()` function in the formula
- in `s()` ((see `?s` and `?smooth.terms` for all the available choices for smoothing)
  - to have psplines we need to set `k` which is the dimension of the basis/total number of parameters: it must be set to $K + m + 1$, according to the notation used in class)
  - `bs` is used to choose the kind of basis used: we're interested in `ps` (for penalized spline)
  - `m` is used to set a vector with the degree of spline and the penalty term, `c(m1, m2)`, where:
    - `m1` determines the degree of the spline, but it refers to the maximum order of its partial derivatives that must be continuous, $m - 1$, according to the notation used in class.
      For example: `m1 = 0` for linear splines, `m1 = 1` for quadratic splines and `m1 = 2` for cubic splines.
    - `m2` is the order of the (squared) differences used to define the penalty term. For example: `m2 = 1` for first-order differences, `m2 = 2` for second-order differences.

```
## linear splines with 20 equispaced knots + penalty based on the squared
## first-order differences; k = 20 + 1 + 1 = 22
summary(pspline1 <- gam(accel ~ s(times, bs = "ps", k = 22, m = c(0, 1)),
                        data = mcycle))

## Error in gam(accel ~ s(times, bs = "ps", k = 22, m = c(0, 1)), data = mcycle):  non
trovo la funzione "gam"
```

The summary is the same as that seen in class, separating the intercept and the regressors part; here we are more interested in the second part. edf $= 12.51$ is trace - 1; we have the stats and we didn't selected penalization param λ.
Let's see another example with cubic splines and some changes

```
## cubic splines with 20 equispaced knots + penalty on the squared
## second-order differences
summary(psplinec <- gam(accel ~ s(times, bs = "ps", k = 24, m = c(2, 2)),
                        data = mcycle))

## Error in gam(accel ~ s(times, bs = "ps", k = 24, m = c(2, 2)), data = mcycle):  non
trovo la funzione "gam"
```

In this second case GCV is slightly better than the previous. What happens if we use the default parameters?

```
## default setting
summary(gam1 <- gam(accel ~ s(times), data = mcycle))

## Error in gam(accel ~ s(times), data = mcycle):  non trovo la funzione "gam"
```

In this case we get another version of spline (not bspline + penalization 1 or 2). With default setting we get even better results in terms of GCV.
Finally, in general by applying plot we can visualize the estimated effect of x on y with confidence bands: this is centered around 0 (because here we consider only the second part, not the intercept)

```
## estimated centered smooth function: (see ?plot.gam)
plot(gam1)

## Error in eval(expr, envir, enclos):  oggetto 'gam1' non trovato
```

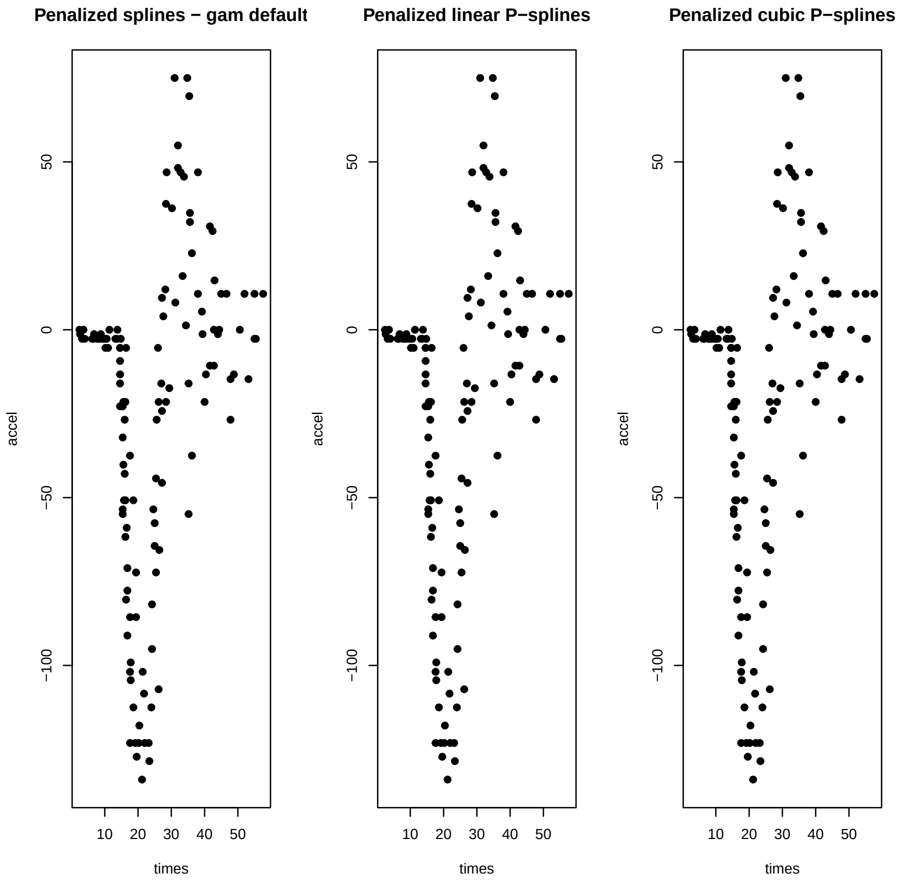
9.2.4 Comparison and wrapup

Starting by `AIC` (AIC is not available for smoothing splines computed with the `smooth.spline` function) we have:

```
## comparison among fitted models
AIC(poly12)

## [1] 1222.002

AIC(poly12raw)

## [1] 1222.002

AIC(lspline)

## [1] 1216.979

AIC(cspline)

## [1] 1219.091

AIC(psplinel)

## Error in eval(expr, envir, enclos):  oggetto 'psplinel' non trovato

AIC(psplinec)

## Error in eval(expr, envir, enclos):  oggetto 'psplinec' non trovato

AIC(gam1)

## Error in eval(expr, envir, enclos):  oggetto 'gam1' non trovato
```

More or less all the models provide similar results; the best is the last/default, with linear spline with 8 knot is close.
For graphical comparison between observed and fitted values we use predict (see `?predict.gam`); for example using results from the `gam` function we end up with similar results with the following commands

```
timesp <- seq(2,58,length.out=200)
new_data <- data.frame(times=timesp)
pred.gam1 <- predict(gam1, newdata = new_data, type = "response")

## Error in eval(expr, envir, enclos):  oggetto 'gam1' non trovato

pred.psplinel <- predict(psplinel, newdata = new_data, type = "response")

## Error in eval(expr, envir, enclos):  oggetto 'psplinel' non trovato

pred.psplinec <- predict(psplinec, newdata = new_data, type = "response")

## Error in eval(expr, envir, enclos):  oggetto 'psplinec' non trovato

par(mfrow = c(1,3))
plot(mcycle$times, mcycle$accel, xlab = "times", ylab = "accel",
     pch = 19, main = "Penalized splines - gam default")
lines(timesp, pred.gam1, lwd = 3, col = 2)

## Error in eval(expr, envir, enclos):  oggetto 'pred.gam1' non trovato

plot(mcycle$times, mcycle$accel, xlab = "times", ylab = "accel",
     pch = 19, main = "Penalized linear P-splines")
lines(timesp, pred.psplinel, lwd = 3, col = 3)

## Error in eval(expr, envir, enclos):  oggetto 'pred.psplinel' non trovato

plot(mcycle$times, mcycle$accel, xlab = "times", ylab = "accel",
     pch = 19, main = "Penalized cubic P-splines")
```



```
lines(timesp, pred.psplinec, lwd = 3, col = 4)

## Error in eval(expr, envir, enclos):  oggetto 'pred.psplinec' non trovato
```

9.3 TODO

da guardare

- sto riassuntone delle splines disponibili in `https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-019-0666-3`
- splines2 package `https://cran.r-project.org/web/packages/splines2/vignettes/splines2-intro.html`

Chapter 10

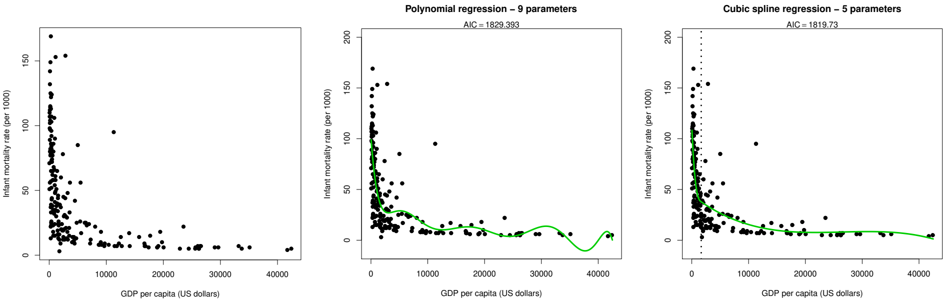
Variable transformations

Remark 51. Again in this section we extend the flexibility of the standard model to overcome possible difficulties related to adeguacy of its assumption.
by seeing an alternative strategy to deal with inadequacy of *linearity* assumption; up to now we introduced *nonlinearity in the regressor* but keeping linearity in the parameters/coefficients so we can easily fit these kind of model with standard procedure and using same inferential apparatus.
Another simple way to inject nonlinearity in gaussian models, that holds most of the stuff already seen (estimation inference), is the trasformation of the independent variable. This will lead to *nonlinearity in the parameters*

10.1 Introduction

10.1.1 A motivating example

Example 10.1.1 (Infant mortality vs GDP)**.** A researcher is interested in evaluating the effects of economic conditions on mortality, starting from information about 193 countries. In particular, for each country, the observed quantities are: infant mortality rate (per 1000 live births) and GDP per capita (US Dollars).
In figure 10.1 (a) the plot shows a clear nonlinear dependence pattern with a rapid drop and then a plateau; in (b) and (c) a polynomial and cubic spline regression respectively



(a) (b) (c)

Figure 10.1: example gdp mortality

10.1.2 Transformable nonlinearity

10.1.2.1 Multiplicative models: an example

One possible choice could be the following. Let, as usual:

- Y_i be the random variable that describes the value for the dependent variable observed on the i-th sample unit ($i = 1, \dots, n$)
- x_i be the value of the regressor for the i-th sample unit

Rather than using the linear regression with an additive error we can resort (among the other) to one particular example which belongs to the class of multiplicative models:

$$Y_i = \alpha x_i^{\beta_1} \exp(\varepsilon_i), \quad \varepsilon_i | x_i \sim N\big(0, \sigma^2\big) \text{ IID}$$

which is different from the previous ones because:

- it does not have an additive error term: it's a *multiplicative model* because the contribution of the random part ε_i and the regressors to Y_i are multiplicative;
- this model is *nonlinear* neither in the regressor (we have a power transformation of the regressors) nor in the parameters (nonlinear in the parameter β_1): it furthermore involves a nonlinear transformation of ε_i;
- it's a *non-gaussian* regression model for $Y_i|x_i$: $Y_i|x_i$ does not have a Gaussian distribution, $\varepsilon_i|x_i$ is gaussian but it's trasformed using a nonlinear function exp and the resulting trasnformation is no longer a gaussian (being this family closed only to linear transformation)

So most of the standard assumption are gone.

10.1.2.2 Transformable nonlinearity

Remark 52. What are the interesting features of this specific moltiplicative model? Although the considered model

$$Y_i = h(x_i, \varepsilon_i; \boldsymbol{\theta}) = \alpha x_i^{\beta_1} \exp(\varepsilon_i), \quad \varepsilon_i | x_i \sim N\big(0, \sigma^2\big) \text{ IID}$$

is not linear in (some of) the unknown parameteres, there could be a function $g(\cdot)$ such that once applied to the both sides of the equation such that the resulting transformed will be a gaussian model linear in the parameter

$$g(Y_i) = \beta_0 + \beta_1 b(x_i) + \varepsilon_i, \quad \varepsilon_i | x_i \sim N\big(0, \sigma^2\big) \text{ IID}$$

So the model we started is a nonlinear but can be trasformed into linear (with standard other properties as well) by applying a proper function to both sides of the equation.

Important remark 65. In this specific example, if $Y_i > 0$ ($i = 1, \dots, n$) and we assume that $\alpha > 0$, we can apply log to both sides of the equation:

$$\begin{aligned} \ln Y_i &= \ln\left[\alpha x_i^{\beta_1} \exp(\varepsilon_i)\right] \\ &= \underbrace{\ln \alpha}_{\beta_0} + \beta_1 \underbrace{\ln x_i}_{b(x_i)} + \varepsilon_i \end{aligned}$$

So by transforming the dependent variable we end up with a new regression model where:

- the dependent variable is $\ln Y_i$
- thanks to the specific choice on the right hand side of the equation we have just a right hand function carachterized by linearity in the parameter (which can be mapped back to the original ones) and in the trasformed covariates (it's nonlinear in x, but linear in $\ln x$) and an additive gaussian erro r
- so we obtain a Gaussian regression model for $\ln Y_i|x_i$ that is **linear in the parameters**

Remark 53. This is just one example that we can find in the literature about models that are nonlinear but can be linearized by introducing a suitable transformation

Example 10.1.2 (ln(Infant mortality) vs ln(GDP) - linear regression)**.** If we want to fit this model on our data what we can do is simply transform both the y and x; in fig 10.2 the results (the pattern after the transformation seems to be linear) and following the estimation of the model

```
Coefficients:
```

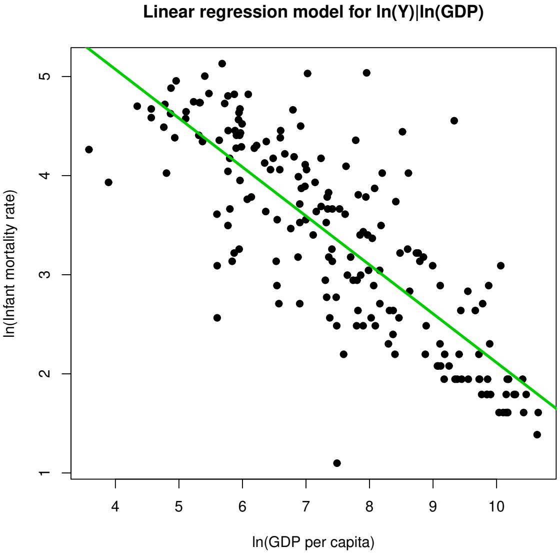


Figure 10.2: lg(mortality) vs log(gdp)

```
                    Estimate   Std. Error   t value   Pr(>|t|)
       (Intercept)     7.045        0.199    35.379      0.000
  log(GDPperCapita)   -0.493        0.026   -19.070      0.000
Residual standard error:  0.5938 on 191 degrees of freedom

Multiple R-squared:  0.6556, Adjusted R-squared:  0.6538
F-statistic:  363.7 on 1 and 191 DF, p-value:  < 2.2e-16
```

Remark 54. The process of finding a proper transformation (in this case the plot suggest a linear very good approximation) is done by trial and error but is still an attractive way to overcome nonlinearity

Important remark 66. it's not a free lunch however: by fitting a model on a transformed dependent variable we can exploit all the inferential tools we've seen but use of these tools will be meaningful if and only if we focus on the transformed dependent variable
At some point we'll want to go back to the original value instead of the transformed one (say for prediction or other); in general there are some problems in working out the true distribution for the original value

Remark 55. however when the transformation is the logarithm we can workout it the original variable distribution

10.1.3 Lognormal random variables

10.1.3.1 Distribution/shape

Important remark 67 (The distribution). Whenever we have $Y_i > 0$ non-negative dependent random variables ($i = 1, \dots, n$) which are indepent from each other, it is possible to prove that if it's logarithm is normally distributed than the starting Y_i is lognormally distributed (and viceversa):

$$\ln Y_i \sim N(\mu_i, \sigma^2) \iff Y_i \sim \ln N(\mu_i, \sigma^2)$$

In this case the density of the original/starting lognormal Y_i is somewhat similar to standard normal

$$f(y_i; \mu_i, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{y_i} \exp\left[-\frac{(\ln y_i - \mu_i)^2}{2\sigma^2}\right]$$

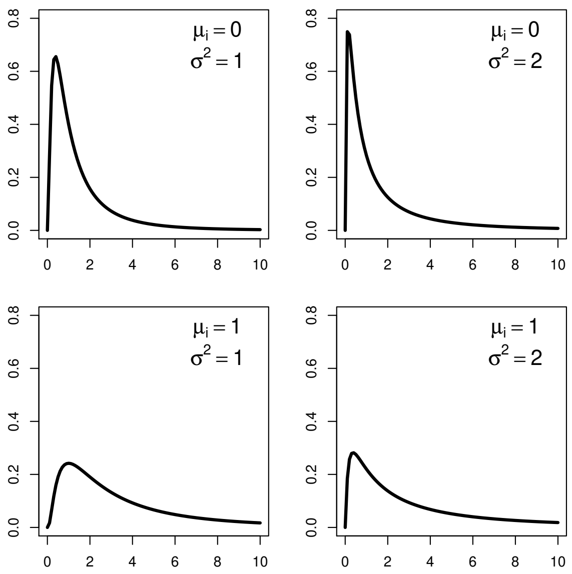


Figure 10.3: Lognormal shapes

where:

$$
\begin{aligned}
y_i &\in (0, +\infty) \\
\mu_i &\in (-\infty, +\infty) \\
\sigma^2 &\in (0, +\infty)
\end{aligned}
$$

Remark 56 (Differences). So only two differences with standard gaussian:

- rather than having $y_i - \mu_i$ we have $\ln y_i - \mu_i$
- we have the $1/y_i$ out of exponential which is just the $\frac{\partial \ln Y_i}{\partial Y_i}$ of the density transformation formula from the virols times

Remark 57 (Shapes). In figure 10.3 various shapes for parameters μ and σ^2: both μ and σ^2 have impact on location and on the shape, as we'll see in the moments (differently from standard gaussian where μ controls location and σ^2 the shape).
Look at the picture either by row or by column to know the impact of sigma or mu respectively

Example 10.1.3 (Use of Lognormal distributions)**.** It's used for statistical phenomena that take only positive values and show skewed distributions such as: intensities/densities, durations/waiting times, earnings/expenditures.

10.1.3.2 Moments

Important remark 68 (Expected value). As we've seen the *expected value* depends both on μ and σ^2 by the following formula

$$
\begin{aligned}
\mathrm{E}[Y_i] &= \exp\left(\mu_i + \frac{\sigma^2}{2}\right) \\
&> \exp(\mu_i) = \exp(\mathrm{E}[\ln Y_i])
\end{aligned}
$$

So the expected value of Y_i is strictly larger than the exponential of the expected value of the logarithm of Y_i. The logarithm of Y_i is the gaussian distribution; if i want to go back to the original scale i'll have a dependent variable Y_i whose expected value is not simply the exponential $\exp \mu_i$ but it's larger and is $\exp \mu_i + \frac{\sigma^2}{2}$.
The reason for the inequality lies in the Jensen inequality since given that log is concave

$$
\ln \mathrm{E}[Y_i] \geq \mathrm{E}[\ln Y_i]
$$

Important remark 69 (Variance). For *the variance*:

$$\begin{aligned}\mathrm{Var}\left[Y_i\right] &= \exp\left[2\left(\mu_i+\sigma^2\right)\right]\left[1-\exp\left(-\sigma^2\right)\right]\\ &= \left\{\mathrm{E}\left[Y_i\right]\right\}^2\left[\exp\left(\sigma^2\right)-1\right]\\ &\propto \left\{\mathrm{E}\left[Y_i\right]\right\}^2\end{aligned}$$

also the variance of Y_i depend on both μ and σ^2.
Although σ^2 does not depend on i, the n random variables are not *homoscedastic* since the variability of Y_i depend on the expected value so units caracterized by different parameter μ_i will be characterized by different variances even though they have a common param σ.

Important remark 70 (Coefficient of variation). Even though we have heteroskedasticity in the n random variables, they have the same coefficient of variation (CV):

$$\mathrm{CV}\left[Y_i\right] = \frac{\sqrt{\mathrm{Var}\left[Y_i\right]}}{\mathrm{E}\left[Y_i\right]} = \frac{\mathrm{E}\left[Y_i\right]\sqrt{\exp\left(\sigma^2\right)-1}}{\mathrm{E}\left[Y_i\right]} = \sqrt{\exp\left(\sigma^2\right)-1}$$

this is a consequence of the variance to be proportional to the square of the expected value

10.1.4 Gaussian linear models for log transformations

Remark 58. what are the implication when we use lognormal distribution in context of regression, by fitting a gaussian regression model to the logarithm of Y_i?

10.1.4.1 Model

Assuming a gaussian linear regression model for the log of Y_i ...

$$\ln Y_i|x_{1i}\ldots x_{pi} \sim N(\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{pi},\sigma^2),\quad i=1,\ldots,n \text{ independent}$$

implies ($\iff$ actually) assuming a lognormal regression model for the original variable (it's a model with conditional lognormal distributions):

$$Y_i|x_{1i}\ldots x_{pi} \sim \ln N(\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{pi},\sigma^2),\quad i=1,\ldots,n \text{ independent}$$

Regarding this latter model we have that is nonlinear both with respect to the betas and in the x (being this stuff related to expected value via an exponential)

$$\begin{aligned}\mathrm{E}\left[Y_i|x_{1i}\ldots x_{pi}\right] &= \exp\left(\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{pi}+\frac{\sigma^2}{2}\right)\\ &> \exp\left(\beta_0+\beta_1 x_{1i}+\ldots+\beta_p x_{pi}\right) = \exp\left(\mathrm{E}\left[\ln Y_i|x_{1i}\ldots x_{pi}\right]\right)\\ \mathrm{Var}\left[Y_i|x_{1i}\ldots x_{pi}\right] &\propto \left\{\mathrm{E}\left[Y_i|x_{1i}\ldots x_{pi}\right]\right\}^2\end{aligned}$$

Again the expected value is strictly larger that the exponential of the expected value of the first model (regarding $\log Y_i$) fitted on the transformed

Important remark 71 (Naive backtransformation? NO). So we can fit a model on the trasformed variable and performe inference on the parameters beta but if we're interested in estimating the expected/fitted value for the original variable we cannot simply take the fitted value for the transformed variable and apply the inverse transformation (the exponential), but we have to take into account the fact that we have also the $\frac{\sigma^2}{2}$ parameter
Going back on the original variable must be done with care (adjustments? include the estimate of $\sigma^2/2$ in doin the prediction)

Important remark 72. Another property characterizing this model is again that if we fit an homoskedastic model on the logarithm of Y_i we're fitting an heteroskedastic model on the original variable Y_i since the variance of Y_i given the regressors will be proportional to the square of expected value and is no longer constant but a nonlinear function of the regressors.

Important remark 73. As we'll see, the fact that working with nonlinear transformation of the dependent variable leads to models that on the original scale are characterized by heteroskedasticity can be exploited also to address issues related to heteroskedasticity

In the end by defining gaussian linear regression models on log transformation we're implicitly introducing non linear regression model on the original variable where also variances are no longer a constant

Remark 59. Another perspective from where looking at the two models is that by fitting a gaussian classical linear model with additive regressors and error term for the logarithmic transformation ...

$$\begin{aligned}&\ln Y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)\\ &\Updownarrow\\ &Y_i = \exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \epsilon_i), \quad \epsilon_i \sim N(0, \sigma^2)\\ &\Updownarrow\\ &Y_i = \exp(\beta_0) \cdot \exp(\beta_1 x_{1i}) \ldots \cdot \exp(\beta_p x_{pi}) \cdot \exp(\epsilon_i), \quad \exp(\epsilon_i) \sim \ln N(0, \sigma^2)\end{aligned}$$

... we end up/are implicitly defining a model with a Lognormal nonlinear model with multiplicative regression function and error term for the original variable Y_i.

This furthermore means that the interpretation of the betas is different: an increase of 1 unit in one x_i will make the y be multiplied by $\exp(\beta_i)$

10.1.4.2 Loglikelihood

Going for the maximization, the loglikelihood can be written as the logarithm of the likelihood obtained by the product of independent lognormal densities:

$$\begin{aligned}l_{\ln N}(\boldsymbol{\beta}, \sigma^2|Y) &= \sum_{i=1}^{n} \left\{ \ln\left[\frac{1}{\sqrt{2\pi\sigma^2}}\right] + \ln\left[\frac{1}{y_i}\right] - \frac{(\ln y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi})^2}{2\sigma^2} \right\}\\ &= -\sum_{i=1}^{n} \ln y_i - \frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 + \frac{\sum_{i=1}^{n}(\ln y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi})^2}{2\sigma^2}\\ &= -\sum_{i=1}^{n} \ln y_i + l_N(\boldsymbol{\beta}, \sigma^2|\ln Y)\end{aligned}$$

where the subscript $l_{\ln N}$ emphasizes the fact that the loglikelihood is obtained starting from the lognormal conditional distribution for the original variable Y
Thus apart from an additive constant that does not involve the model parameters, loglikelihoods for (conditional) independent Lognormal distributions are *equivalent* to loglikelihoods for (conditional) independent Gaussian distributions (after applying the logarithm to the original variables):

$$l_{\ln N}(\boldsymbol{\beta}, \sigma^2|Y) \approxeq l_N(\boldsymbol{\beta}, \sigma^2|\ln Y)$$

These results implies if i'm interested in a lognormal model for Y_i a quick and simple way to get the maximum likelihood estimates for the model parameters is by taking the $\ln Y_i$ and fitting the ML estimates for the gaussian model on the transformed variable; in order to maximize the loglik for lognormal model it is enough to maximize the loglik for the normal model on the logarithm of Y_i
So same inferential results (estimates, properties and hypotheses testing) as for Gaussian linear models.

Remark 60. As mentioned before the only problem we can get is the prediction (conditional expected values of Y_i) on the Y_i scale using model estimated on $\ln Y_i$

10.1.4.3 Estimation for conditional expected values

Important remark 74 (Biased for mean). We have that

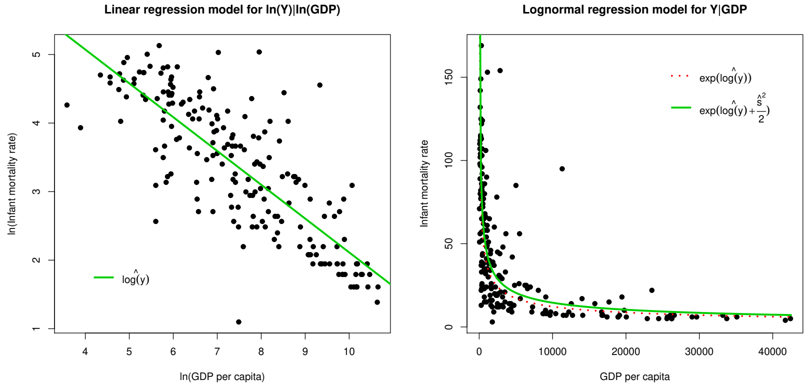
- the estimation for the logarithmic transformation:

$$\widehat{\ln y_i} = \mathrm{E}\,[\ln \widehat{Y_i | x_{1i}} \ldots x_{pi}] = \hat{b}_0 + \hat{b}_1 x_{1i} + \ldots + \hat{b}_p x_{pi}$$

- while the estimation for the original variable:

$$\widehat{y_i} = \mathrm{E}\,[Y_i | \widehat{x_{1i} \ldots} x_{pi}] = \exp\left[\hat{b}_0 + \hat{b}_1 x_{1i} + \ldots + \hat{b}_p x_{pi} + \frac{s^2}{2}\right] > \exp\left[\widehat{\ln y_i}\right]$$

  Therefore the inverse (exponential) function applied to the estimated conditional expected value of the logarithmic transfomation leads to *biased* estimates for the conditional expected value of the original variable.
  We cannot simply take the fitted value on the $\ln Y_i$ and exponentiate it; we must take $\frac{s^2}{2}$ into account



(a) (b)

Figure 10.4: Lognormal reg

Important remark 75 (Unbiased for median). It is possible to prove that $\exp\left[\widehat{\ln y_i}\right]$ ($i = 1, \ldots, n$) are unbiased estimates for the conditional *medians* of the original variable (basicly quantiles are equivariant to nonlinear monotone transformation).

Example 10.1.4 (Infant mortality vs GDP - lognormal regression)**.** Figure 10.4 (a) show the estimated model on the $\ln Y_i$ scale, while on the right the prediction: the biased red is just the left green exponentiated while the unbiased green is the correct estimate.
With the polynomial and the spline we need to choose among the three models seen so far for this data: we have two models assuming a conditional gaussian distribution for Y (poly and spline) and one assuming lognormal distribution for Y.
We have to pay attention on computing of AIC and BIC (which function `extractAIC` or `AIC`)

10.1.4.4 Models comparison criteria

AIC and *BIC* can be used to perform model selection among models with different assumptions about the conditional distribution of the dependent variable.
We start from the standard formula in both and develop a bit to compare the lognormal results with the normal siebling model comparison

$$\begin{aligned}
AIC_{\ln N}(M|Y) &= -2l_{\ln N}(\hat{\mathbf{b}}, \hat{s}|Y) + 2(p+2) \\
&= -2\left[-\sum_{i=1}^{n} \ln y_i + l_N(\hat{\mathbf{b}}, \hat{s}|\ln Y)\right] + 2(p+2) \\
&= -2l_N(\hat{\mathbf{b}}, \hat{s}|\ln Y) + 2(p+2) + 2\sum_{i=1}^{n} \ln y_i \\
&= AIC_N(M|\ln Y) + 2\sum_{i=1}^{n} \ln y_i \\
BIC_{\ln N}(M|Y) &= BIC_N(M|\ln Y) + 2\sum_{i=1}^{n} \ln y_i
\end{aligned}$$

Important remark 76. In order to correctly compute the AIC/BIC for the lognormal regression model we basically compute the the AIC and BIC for the gaussian model on $\ln Y$ (after fitting the model) and add a quantity dependent only on the y_i that is $2\sum_{i=1}^{n} \ln y_i$. This latter thing lead to the following consideration
If some models are fitted on y_i and some others on the log Y_i we cannot make direct comparison:

- for lognormal model in the AIC of the $\ln Y_i$ model we have to add/adjust the quantity $2\sum_{i=1}^{n} \ln y_i$
- for other transformation (not log) *this correction is not available* and become impossible making comparison between transformed and untrasformed dependent linear model and

Important remark 77. Comparisons among *AIC* (or *BIC*) values are admissible if and only if these model comparison criteria are computed with reference to the same random sample: it does not make sense to compare tha *AIC*/*BIC* of a Gaussian regression model fitted on $\ln Y$ with the *AIC*/*BIC* of a Gaussian regression model fitted on Y

Example 10.1.5 (Final comparison among models)**.** To conclude the example we have three models and their following stats:

Cond. distribution	GDP effect	n. of param.	log-likelihood	*AIC*	*BIC*
Gaussian	polynomial	10	−903.696	1829.393	1865.282
Gaussian	cubic spline	6	−903.865	1819.730	1839.306
Lognormal	nonlinear*	3	−816.171	1638.342	1648.130

The first two are fitted on the original Y_i while * is fitted using $\ln Y_i$ (using as regressor the log of gdp).
AIC can be directly compared for model on the same dependent variable: in order to compare the third model with the first two we need to go back to the original scale; this is done by applying the correction (i guess) to the AIC seen before.
In this example the log trasnformation was successful: we have lower number of parameter, but obtain an higher loglikelihood; thus a greatly reduced lower AIC/BIC.

Example 10.1.6 (Esempio da stackexchange `https://stats.stackexchange.com/questions/61332`)**.** A fictious example in the case of comparison of loglinear vs rest of the world dove si aggiunge $2\sum_i \log(y_i)$ all'AIC del modello su logaritmo:

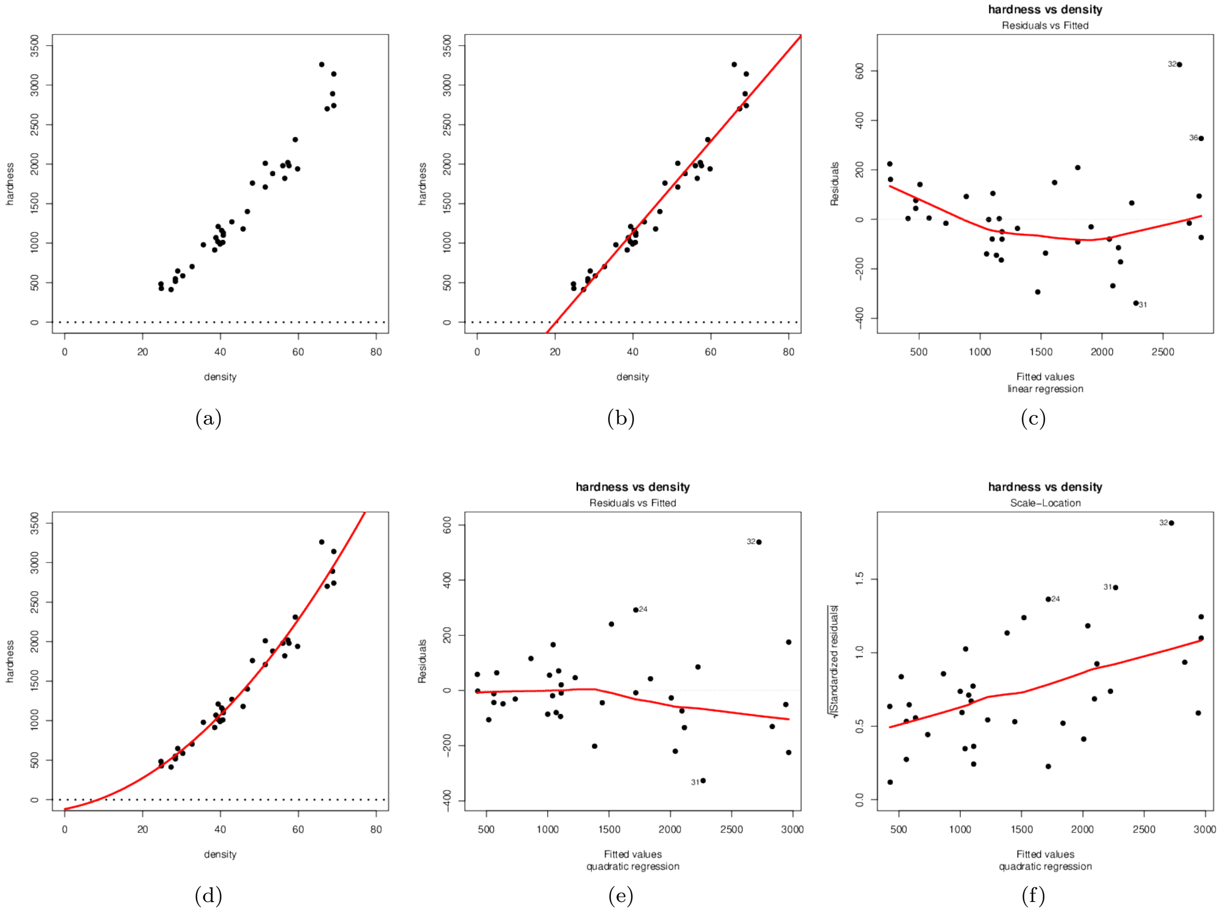
```
seedrates <- data.frame(rate = c(50, 75, 100, 125, 150),
                        grain = c(21.2, 19.9, 19.2, 18.4, 17.9))
quad.lm <- lm(grain ~ poly(rate,2), data=seedrates)
loglin.lm <- lm(log(grain) ~ log(rate), data=seedrates)
oldopt <- options(digits = 2)
AIC(quad.lm, loglin.lm)

##           df   AIC
## quad.lm    4  -4.1
## loglin.lm  3 -37.2
```

We need to add `2*sum(log(seedrates$grain)) = 29.6` to the AIC for the loglinear model (or, subtract it from the AIC for the quadratic model).

```
AIC(quad.lm, loglin.lm) + matrix(ncol=2, c(0,0,0, 2*sum(log(seedrates$grain))))

##           df  AIC
## quad.lm    4 -4.1
## loglin.lm  3 -7.6

options(oldopt)
```

10.2 Heteroschedasticity and variance-stabilising transformations

Remark 61. Transforming the dependent variable can be helpful not onlyu to adjust for violation of linearity but can be used for inadequacy of homoskedasticity assumption.
When using log of dependent we're implicitly assuming that the conditional distribution of the original Y_i are no longer homoskedastic (conditional variance proportional to square of expected value)



(a) (b) (c)

(d) (e) (f)

Figure 10.5: Timber data

Example 10.2.1 (A motivating example: Timber data)**.** We have data ($n = 36$) on `hardness` (hardness of an hardwood timber, Y, *cannot take negative values*) and `density` (density of an hardwood timber, X). Data is in figure 10.5 (a):

- by looking at the data one could think a gaussian regression model should be ok to express the relation between hardness and density. So in (b) and (c) a gaussian linear model (fitted values) and its residuals vs fitted. Some problems:
  - for density below 20 our model would predict an hardness that is negative (but this can't be given the phenomenon)
  - the residuals seem to show a pattern in their average value that suggests avoiding the linearity hypothesis and trying a quadratic model
- in (d) we use a quadratic estimate and there seems to be some improvements in the fitted values, which is more or less confirmed by the residuals vs fitted (e); however this latter and (f) seems to suggest an increasing variance/more spreadness as the fitted values increases

The inclusion of a quadratic effect seems reasonable, but the model is still inadequate. There is a clear pattern in the magnitude of the standardised residuals: it tends to increase as the fitted value increases. This is a symptom of heteroschedasticity of the conditional distributions

10.2.0.1 Variance-stabilising transformations

Remark 62. Other approach to handle heteroskedasticity will be seen in the second part of the course; here we see a simple way, the variance stabilizing transformation.
The idea behind is to find a transfortmation of the dependent variable such as the resulted transform becomes homoskedastic, removing the differences in the conditional variances.
There are several example of transformation: different transformations work well in different conditions.
We focus on one example of these transformation: the Box-Cox transformation.

Remark 63.

- Can be applied when we have *dependent variable taking only positive values*;
- works well when we have conditional variances that are *proportional to power transformation of the expected value* (eg is obtained

Proposition 10.2.1 (Boxcox truth)**.** *Consider a Gaussian regression model with heteroschedastic conditional distributions* $\operatorname{Var}[Y_i|x_{1i},\ldots,x_{pi}] = \sigma_i^2 \forall i$*; when the dependent variables* Y_i *take* only positive values*, it is possible to prove that:*

$$\begin{aligned}
\sigma_i^2 \propto \operatorname{E}[Y_i|x_{1i},\ldots,x_{pi}] &\implies \operatorname{Var}\left[\sqrt{Y_i}|x_{1i},\ldots,x_{pi}\right] \approxeq \sigma^2 \ \textit{constant}\\
\sigma_i^2 \propto (\operatorname{E}[Y_i|x_{1i},\ldots,x_{pi}])^2 &\implies \operatorname{Var}[\ln(Y_i)|x_{1i},\ldots,x_{pi}] \approxeq \sigma^2\\
\sigma_i^2 \propto (\operatorname{E}[Y_i|x_{1i},\ldots,x_{pi}])^3 &\implies \operatorname{Var}\left[Y_i^{-0.5}|x_{1i},\ldots,x_{pi}\right] \approxeq \sigma^2\\
\sigma_i^2 \propto (\operatorname{E}[Y_i|x_{1i},\ldots,x_{pi}])^4 &\implies \operatorname{Var}\left[Y_i^{-1}|x_{1i},\ldots,x_{pi}\right] \approxeq \sigma^2\\
\ldots
\end{aligned}$$

Example 10.2.2. Therefore:

- looking at the first row: if the variance is proportional to the expected value, then if we take the square root of the dependent variable its conditional variance will be approximately constant/homoskedastic
- it proportional to the square of the conditional expected value then we go with the log of the dependent variable to have homoskedasticity
- etc

The idea is that the higher the power of the proportion means that the variance increase rapidly with the value; and so we must squeeze/compress more the dependent variable to have homoskedasticity

10.2.0.2 Box-Cox transformation

Definition 10.2.1 (Boxcox transformation)**.** We change Y_i according to the following equation (somewhat similar to a power transformation) and a parameter λ

$$Y_i^* = \frac{Y_i^\lambda - 1}{\lambda}, \quad \lambda \in \mathbb{R}$$

When we consider the limit for $\lambda \to 0$ we have that actually

$$\lim_{\lambda \to 0} \frac{Y_i^\lambda - 1}{\lambda} = \ln(Y_i)$$

while for $\lambda = 1$ we have no transformation

Remark 64. We have basically to choose lambda: we don't know in advance what's the optimal value that we shuld apply. The idea was to obtain by standard ML (with the other parameters) where where rather than y_i we substitute the transformation and maximize for its parameter as well.
Differently from what we've seen with the betas it has no closed formula/analytical expression but np

Important remark 78 (Choice of λ). It's done via maximum likelihood estimation

$$l(\boldsymbol{\beta}, \sigma^2, \lambda) = -\frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(\frac{y_i^\lambda - 1}{\lambda} - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi}\right)^2$$

Remark 65 (Technical difficulties)*.* As said, there is not an analytical formula for computing $\hat{\lambda}$: it must be found numerically.
For a grid of possible/candidate values (say from −3 to 3, 100 uniformly distributed length):

- we calculate the boxcox transformation
- we estimate models parameters with the transformation as dependent by optimizing for betas and σ^2
- we take the maximized loglikelihood for each λ

the corresponding log-likelihoods are computed and compared after normalization (so there will be specific estimates for $\boldsymbol{\beta}$ e σ^2 associated with each point in the grid)

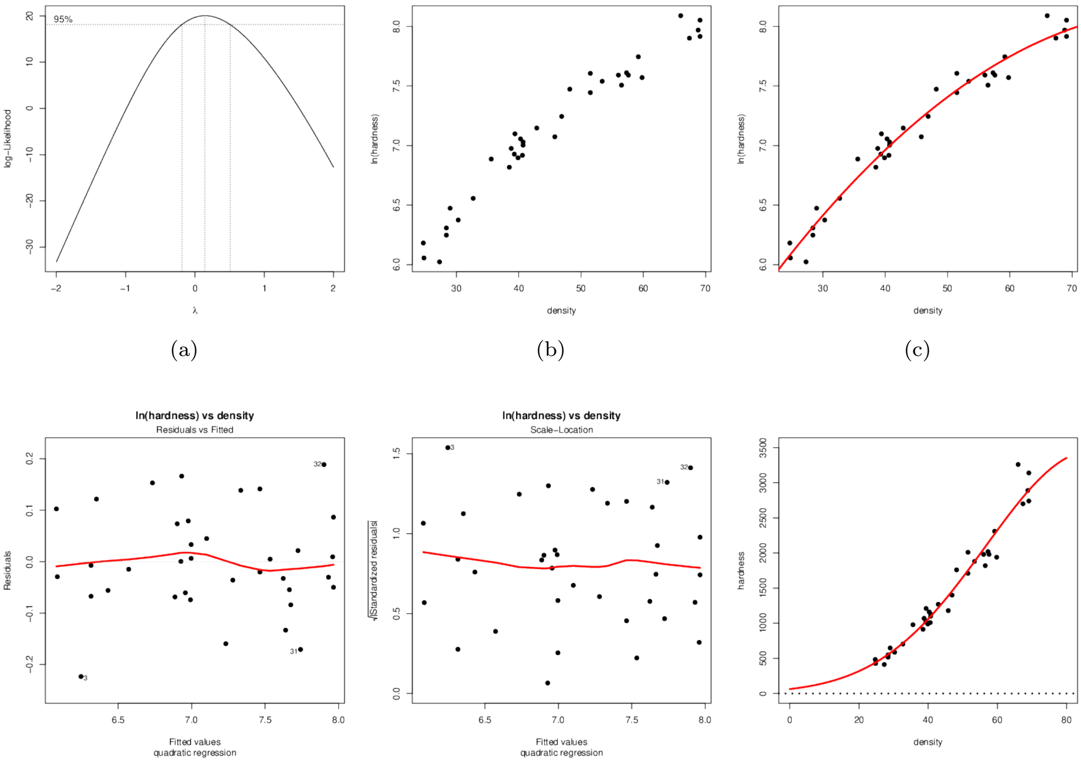
Example 10.2.3 (Timber data)**.** In figure 10.6

- (a) choice of λ: The plot suggests a value for λ close to/not different from 0 (and suggests a logarithmic transformation)
- (b) relation between $\ln(Y)$ and x: looking at the plot there seems to be still some curvature, so an idea could be fitting a quadratic regression model on the $\ln(Y)$;
- in (c) the fit while: in (d) the residuals plots no evident pattern in the average of the residuals (linearity) and in (e) no problem on the magnitude of the standardised residuals (heteroskedasticity as well); so this fit should be ok
- in (f) we go back on the original scale by adding $s^2/2$ to the fitted value on $\ln Y_i$ and then back-exponentiating: the exponentiation will make all the prediction positive incidentally

10.2.0.3 Cautionary remarks

Important remark 79*.* Some remarks

- the Box-Cox transformation is *only an example* of variance-stabilising transformation (most popular btw): alternative transformations have been proposed to deal with dependent variables that *can also take negative values*;
- the nonlinear nature of the transformation poses nontrivial issues if one is interested in obtaining information about the original dependent variable: generally, the *form of the conditional distribution for the original variable is not known* (the only exception being the logarithmic transformation)



(a) (b) (c)

(d) (e) (f)

Figure 10.6: Timber boxcox