Statistical models

August 29, 2024

Contents

I Gaussian linear models **11**

1 Introduction **13**
1.1 Statistical models: general definitions 13
1.1.0.1 Random samples 14
1.1.0.2 Parametric statistical models 14
1.1.0.3 Parametric statistical model specification 14
1.1.0.4 Likelihood function of θ 15
1.2 Multivariate Gaussian distributions review 16
1.2.0.1 Joint probability density function 16
1.2.0.2 Standardised multivariate Gaussian distribution 17
1.2.0.3 Some properties 19

2 Gaussian linear model **21**
2.1 An introductory example 21
2.1.1 Simple linear regression 21
2.1.1.1 Setup 21
2.1.1.2 Models specification 21
2.1.1.3 Estimation 23
2.1.2 Multiple linear regression 23
2.1.2.1 Introducing other regressors 23
2.1.2.2 Model definition/specification 23
2.1.2.3 Estimation 25
2.2 General definition 25
2.2.0.1 Basic assumptions 25
2.2.0.2 Parameter space and sample space 26
2.2.0.3 Probability density function (1) 27
2.2.0.4 Matrix representation 27
2.2.0.5 An alternative definition 29
2.3 Maximum likelihood estimation 30
2.3.1 Likelihood and related quantities 30
2.3.1.1 Likelihood function 30
2.3.1.2 Log-likelihood function 30
2.3.1.3 Score function for β 31
2.3.1.4 Observed Fisher information for β 32
2.3.1.5 Expected Fisher information for β 33
2.3.1.6 Properties of the score function 34
2.3.1.7 Standardising the score function 35
2.3.1.8 Some general properties of the score function . . 36

2.3.2 Maximum likelihood estimation 36
2.3.2.1 Maximum likelihood estimate for β 36
2.3.2.2 Properties of the ML estimator for β 38
2.3.2.3 Some general results related to ML method . . . 38
2.3.2.4 Maximum likelihood estimate for σ^2 39
2.3.2.5 Properties of raw residuals 40
2.3.2.6 Properties of the maximum likelihood estimator for σ^2 40
2.3.2.7 Standardised residuals 41

3 Linear hypotheses **43**
3.1 Linear hypotheses 43
3.1.1 Linear hypotheses on β 43
3.1.2 Nested linear models 45
3.1.3 Likelihood ratio test (LRT) statistics - 1 45
3.2 Constrained maximum likelihood estimation 45
3.2.1 The Method of Lagrange multipliers 45
3.2.2 Residuals of the constrained model 47
3.3 Likelihood ratio properties 49
3.3.1 LRT statistics - 2 49
3.3.2 LRT statistic distribution - σ^2 known 49
3.3.3 LRT statistic distribution - σ^2 unknown 50
3.3.4 Applications 51
3.4 Confidence intervals 51

4 Use of categorical regressors **53**
4.1 Unordered categories 53
4.1.1 Motivating example 53
4.1.2 One-way ANOVA 53
4.1.3 Linear regression with a qualitative regressor 53
4.1.3.1 Using a baseline category 54
4.1.3.2 Exclusion of the intercept 57
4.2 Ordered Categories 58
4.2.1 Motivating example 58
4.2.2 Model with reference category 59
4.2.3 Model with incremental/split coding 59
4.2.4 Linear trend hypothesis 61

5 Models evaluation and comparison criteria **63**
5.1 (Residual) deviance of a Gaussian linear model 63
5.1.1 Saturated models 63
5.1.2 Maximum likelihood estimation of $\mu_1, \mu_2, \ldots, \mu_n$ 63
5.1.3 Comparisons with the saturated model 64
5.1.4 R^2 coefficient 65
5.2 Comparisons among Gaussian linear models 66
5.2.1 Choice among two Gaussian linear models 66
5.2.1.1 Nested models and LRT 66
5.2.1.2 Non-nested models and adjusted R^2 66
5.2.2 Other comparison methods 68
5.2.2.1 Prediction error and Leave-One-Out Cross-Validation 68

5.2.2.2 Akaike information criterion (AIC) 68
5.2.2.3 (Schwartz) Bayesian information criterion (BIC) 70
5.2.2.4 *AIC* or *BIC* 71

6 Lab1 **73**
6.1 Model estimation 73
6.2 Model adequacy 77
6.2.1 Linearity in the regressors 77
6.2.2 Normality 78
6.2.3 Homoscedasticity 79
6.3 Hypothesis testing 79
6.3.1 Linear independence 80
6.3.2 Single beta = 0 81
6.3.3 Equality of two coefficients 82
6.4 Comparison of non-nested models via AIC 83

7 Introducing nonlinearity **85**
7.1 A motivating example 85
7.2 Gaussian nonlinear regression models 86
7.3 Polynomial regression 86
7.3.1 Introducing nonlinearity through polynomials 86
7.3.2 Matrix notation 87
7.3.3 Linear basis expansions 87
7.3.4 ML estimation 88
7.3.5 Properties of regression models with orthogonal polynomials 91
7.3.6 Hypothesis testing 91
7.3.7 Some cautionary remarks on polynomial regression 93
7.4 Piecewise linear regression 93
7.4.1 Piecewise linear functions 94
7.4.2 Linear basis expansion for cont. piecewise linear functions 94
7.4.2.1 Truncated linear basis 95
7.5 Regression splines 95
7.5.1 Spline functions 95
7.5.2 Cubic splines 97
7.5.3 Linear basis expansion for spline functions 97
7.5.3.1 Truncated power basis for spline functions . . . 98
7.5.3.2 B-spline basis functions 98
7.5.4 Concluding remarks 98
7.5.4.1 Estimation 98
7.5.4.2 Inference 98
7.5.4.3 Location of knots 99
7.5.4.4 Polynomials & spline functions - some remarks . 101
7.5.4.5 Concluding remarks 101
7.5.4.6 Problems with splines so far 103

8 Introducting regularization **105**
8.1 Smoothing splines 105
8.1.1 A (seemingly unrelated) alternative approach 105
8.1.2 Penalized least squares estimation 106
8.1.3 Penalized LS estimation and spline functions 106

8.1.4 Natural cubic splines . . . 106
8.1.5 Penalized LS estimation (matrix notation) . . . 109
8.1.6 Choice of the smoothing parameter . . . 110
8.1.6.1 Leave one out crossvalidation . . . 110
8.1.6.2 Generalized Cross-Validation . . . 110
8.1.7 Penalized estimation final remarks . . . 112
8.2 P-splines . . . 112
8.2.1 Gaussian regression models based on P-splines . . . 112
8.2.2 P-splines penalizations . . . 112
8.2.3 Penalized log-likelihood and derived function . . . 114
8.2.4 Penalized ML estimation . . . 114
8.2.5 Estimation of σ^2 . . . 115
8.2.6 Choice of the smoothing parameter . . . 115
8.2.7 Inference . . . 117
8.2.8 Hypothesis testing . . . 118

9 Lab 2 - flexible gaussian regression models 121
9.1 Polynomial regression . . . 121
9.2 Regression splines . . . 125
9.2.1 B-splines . . . 125
9.2.2 Smoothing splines . . . 128
9.2.3 P-splines . . . 130
9.2.4 Comparison and wrapup . . . 133
9.3 TODO . . . 134

10 Variable transformations 135
10.1 Introduction . . . 135
10.1.1 A motivating example . . . 135
10.1.2 Transformable nonlinearity . . . 136
10.1.2.1 Multiplicative models: an example . . . 136
10.1.2.2 Transformable nonlinearity . . . 136
10.1.3 Lognormal random variables . . . 137
10.1.3.1 Distribution/shape . . . 137
10.1.3.2 Moments . . . 138
10.1.4 Gaussian linear models for log transformations . . . 139
10.1.4.1 Model . . . 139
10.1.4.2 Loglikelihood . . . 140
10.1.4.3 Estimation for conditional expected values . . . 140
10.1.4.4 Models comparison criteria . . . 141
10.2 Heteroschedasticity and variance-stabilising transformations . . . 142
10.2.0.1 Variance-stabilising transformations . . . 144
10.2.0.2 Box-Cox transformation . . . 144
10.2.0.3 Cautionary remarks . . . 145

II Generalized linear models 147

11 Introduction to GLM 149
11.1 Motivating examples . . . 149
11.2 Exponential families . . . 152

11.2.1 Families of order 1 . . . 152
11.2.2 Nuisance parameters and weights . . . 153
11.2.3 Some relevant properties . . . 154
11.3 Generalised linear models . . . 155
11.3.1 Definitions . . . 155
11.3.2 Choice of the link function . . . 158
11.3.3 GLM and Gaussian linear models (recappone) . . . 159
11.3.4 Log-likelihood . . . 159
11.3.5 Score function . . . 160
11.3.6 Observed Fisher information . . . 161
11.3.7 Expected Fisher information . . . 162
11.4 Canonical link and 162
11.4.1 ... log-likelihood . . . 162
11.4.2 ... score function . . . 163
11.4.3 ... Fisher information . . . 163
11.4.4 Example using Gaussian linear models . . . 163
11.5 Inference for a GLM . . . 164

12 Poisson regression models **165**
12.1 Model definition and maximum likelihood estimation . . . 165
12.1.1 Model definition . . . 165
12.1.2 Introduction of offsets (*compensating terms*) . . . 166
12.1.3 Interpretation of the model estimates . . . 167
12.1.4 Log-likelihood . . . 167
12.1.5 Score function . . . 168
12.1.6 Fisher information . . . 168
12.1.7 Hessian matrix . . . 168
12.1.8 Matrix representation . . . 168
12.1.9 Maximum likelihood estimation . . . 169
12.2 Optimization algorithms . . . 170
12.2.1 The Newton-Raphson algorithm . . . 170
12.2.2 Fisher scoring algorithm . . . 171
12.2.3 A comparison between the two algorithms . . . 173
12.2.4 Application to Poisson regression estimation . . . 173
12.2.5 Iterative reweighted least squares . . . 174
12.2.5.1 Adjusted/pseudo dependent variable . . . 175
12.2.5.2 Initialisation . . . 175
12.2.5.3 Maximum likelihood estimator . . . 176
12.2.5.4 Estimation of the asymptotic variance . . . 177
12.3 Deviance, residuals and model selection criteria . . . 178
12.3.1 (Residual) deviance . . . 178
12.3.1.1 Saturated model . . . 178
12.3.1.2 Maximum likelihood estimation of $\mu_1, \mu_2, \ldots, \mu_n$ 178
12.3.1.3 (Residual) deviance for a Poisson regression model 179
12.3.1.4 An approximation to D . . . 179
12.3.1.5 Pearson χ^2 statistics . . . 180
12.3.1.6 Goodness of fit test . . . 181
12.3.2 Residuals . . . 182
12.3.2.1 Residuals for Poisson regression models . . . 182
12.3.2.2 Properties and graphical analysis of the residuals 183

12.3.3 Comparisons among Poisson regression models 184
12.3.3.1 Choice among Poisson regression models 184
12.4 Poisson regression example 186
12.4.0.1 Introduction 186
12.4.0.2 First model estimation 186
12.5 Other count models 192

13 Lab 3 - Poisson regression 193
13.1 Intro 193
13.2 Estimation 193
13.3 Model adequacy 196
13.3.1 Goodness of fit test 196
13.3.2 Residuals 197
13.4 Hypothesis testing 197
13.4.1 Does the `dist` regressor impact the model? 197
13.4.2 Simplification to a dummy 199
13.5 Interpreting coefficients 204
13.6 Focus on the offset 204
13.6.1 Don't forget it 204
13.6.2 Offset or standard regressor? 205

14 Generalised linear models for binary outcomes 207
14.1 Binary outcomes 207
14.1.1 Introduction 207
14.1.2 Bernoulli distributions and exponential families 207
14.2 GLM for binary outcomes 208
14.2.1 Basic definition 208
14.2.2 Log-likelihood for Bernoulli GLMs 209
14.2.3 Bernoulli GLMs & covariate patterns 209
14.2.4 Log-likelihood for Bernoulli GLMs & covariate patterns . 209
14.2.5 Binomial distributions 210
14.2.6 The covariate pattern based data structure 210
14.2.7 Relative frequencies and their properties 211
14.3 Logistic regression models 212
14.3.1 Model definition 212
14.3.2 Logistic function 212
14.3.3 Log-likelihood 213
14.3.4 Score function 214
14.3.5 Fisher information 214
14.3.6 Hessian matrix 215
14.3.7 Matrix representation 215
14.3.8 Properties of the score function 216
14.4 Maximum likelihood estimation 216
14.4.1 Newton-Raphson/Fisher scoring algorithm 216
14.4.2 Iterative reweighted least squares 217
14.4.3 Initialisation 217
14.4.4 Maximum likelihood estimator 218
14.4.5 Estimation of the asymptotic variance 218
14.5 Deviance, residuals, hypothesis testing and model selection criteria 218
14.5.1 (Residual) deviance: saturated model 218

14.5.2 Maximum likelihood estimation of $\pi_1, \pi_2, \ldots, \pi_n$ 219
14.5.3 Evaluation of the log-likelihood function for saturated models 219
14.5.4 Deviance comparison for a logistic regression model . . . 220
14.5.5 An approximation to D: pearson χ^2 statistics 221
14.6 Residuals 222
14.7 Residuals analysis when data are sparse 223
14.7.1 A simulation example 223
14.7.2 Data sparsity and curves on residuals vs fitted plot 224
14.7.3 Residuals and the central limit theorem 227
14.7.4 Aggregated Pearson residuals 227
14.8 Testing the adequacy of a logistic regression model 229
14.8.1 Goodness of fit test 229
14.8.2 Hosmer-Lemeshow goodness of fit test 230
14.9 Interpretation 231
14.9.1 Interpretation of model parameters 231
14.9.2 Change in the coding scheme for the dependent variable . 231
14.10Linear hypotheses testing 232
14.10.1General linear hypotheses on β 232
14.10.2Single coefficient hypothesis $H_0 : \beta_h = 0$ $(h = 1, \ldots, p)$ . . 233
14.11Models comparison 233
14.11.1Choice among logistic regression models 233
14.11.2Nested models 233
14.11.3Non-nested models 234
14.12Example: logistic regression 234
14.13GLM for binary outcomes: choice of the link function 240
14.13.1Setup refresher 240
14.13.2Common choices for $g(\cdot)$ 241
14.13.3Probability functions as link functions 242
14.14The use of non-canonical link functions 244
14.14.1Log-likelihood and sufficient statistics 244
14.14.2Score function 245
14.14.3Observed Fisher information 245
14.14.4Expected Fisher information 246
14.14.5Matrix representation 246
14.14.6Maximum likelihood estimation 246
14.14.7Maximum likelihood estimator asymptotics 247
14.14.8Deviance, residuals and goodness of fit tests 247
14.14.9Hypothesis testing and model comparisons 247
14.15Example: probit and cloglog 248
14.15.1Goodness of fit test 248
14.15.2Comparison among link functions 249

15 Lab 4 and 5 GLM for binary outcomes 251
15.1 Lab 4 - Regression for binary outcomes 251
15.1.1 Covariate level dataset 251
15.1.2 Unit level dataset 254
15.1.3 Change in the link function 256
15.2 Lab 5 - Dealing with sparse data 257

16 GLM with Gamma probabilistic component **263**
16.1 The model . . . 263
16.1.1 Gamma RV . . . 263
16.1.2 GLM with Gamma probabilistic component . . . 264
16.1.3 Systematic component . . . 265
16.1.4 Log-likelihood and score function . . . 266
16.1.5 Fisher information matrices . . . 266
16.1.6 Matrix representation . . . 267
16.1.7 Maximum likelihood estimation of $\beta_0, \ldots, \beta_p$ . . . 267
16.1.8 Saturated model and deviance . . . 267
16.1.9 Residuals and estimation of ν . . . 268
16.1.10 Testing linear hypotheses on $\beta_0, \ldots, \beta_p$ . . . 268
16.2 Worked example . . . 269
16.2.1 Data and estimates . . . 269
16.2.2 Comparison between Gamma and Lognormal random variables . . . 271
16.3 Data transformation vs GLM . . . 272

17 Lab6 - Gamma GLM **275**
17.1 Estimates . . . 275
17.2 Hypothesis testing . . . 279
17.3 Comparison with lognormal regression . . . 282
17.3.1 The AIC of lognormal model . . . 284

18 Enhancing the flexibility of GLMs via regularization and additive modelling: an introduction **287**
18.1 P-splines for non-Gaussian outcomes . . . 287
18.1.1 GLMs based on P-splines . . . 287
18.1.2 Timber data example . . . 288
18.2 Introduction to generalized additive models . . . 289
18.2.1 Definition of a generalised additive model (GAM) . . . 289
18.2.2 Identifiability of additive predictors . . . 290
18.2.3 Penalized ML estimation for GAM . . . 290
18.2.4 Ozone example . . . 291
18.3 Further extensions . . . 294

Part I

Gaussian linear models

Chapter 1

Introduction

Remark 1 (Exam). 8 domande aperte (range 0-2) e 13 risp multiple (range 0-1) e 80 minuti.
multiple choice: alcune fatte a radio button con 1 corretta (1 punto per giusta o -0.2 sbagliata), in altre è possibile scegliere più di una opzione (ma non è detto che sian corrette piu di una, può essere che la corretta sia solo una) (i punti sono la percentuale di corrette con malus su errori).
hint:

- controlla le dimensioni delle matrici se ci sono piu matrici
- se la risposta deve esser un vettore, controlla che la smatriciata risulti in un vettore
- occhio nelle risposte possibili multiple se ci sono due risposte equivalenti (tipo il BIC negli esempi)

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

`Y`	statistical phenomenon of interest in a given population `P`
$\mathbf{y} = (y_1,\ \ldots,\ y_n)^\top$	Observed sample *(numerical) values observed on n statistical units* *randomly drawn from the population* `P`
$\Downarrow$	
$\boldsymbol{Y} = (Y_1,\ Y_2,\ \ldots,\ Y_n)^\top$	Random sample: *set of r.vs. that describe the possible value of* `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 `Y` of interest and of the population `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* $\boldsymbol{Y}$ $\mathcal{F} = \{f(\cdot;\boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^k\}$
- have our observed sample - realisation of the random sample $\boldsymbol{Y}$, $\mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_n)^\top$

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 $\mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_n)^\top$ 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

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

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 $(\boldsymbol{\theta} = \left(\mu, \sigma^2\right)^\top \in \boldsymbol{\Theta} = \mathbb{R} \times \mathbb{R}^+)$
- 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 $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$ 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 2. 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 stricly positive and covariances are bounded between the product of the square root of the variances

$\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$	n-dimensional random variable
$\mathbf{y} = (y_1, y_2, \ldots, y_n)^\top \in \mathbb{R}^n$	set of possible values of $\boldsymbol{Y}$ (joint realisations of the n random variables)
$\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)^\top$	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

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

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)

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

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

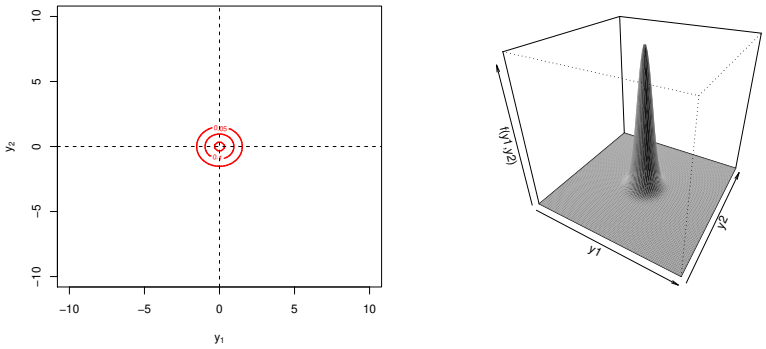
$\boldsymbol{\mu} = \mathbf{0}_n = (0, 0, \ldots, 0)^\top$ n-dimensional null vector
$\boldsymbol{\Sigma} = \mathbf{I}_n$ $n \times n$ identity matrix

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

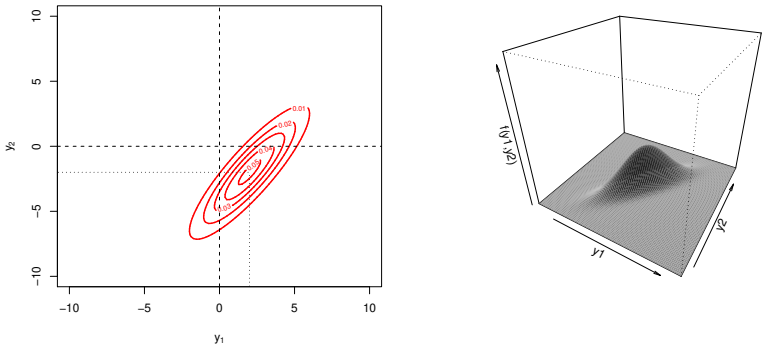
$$\begin{aligned} f(y_1, \ldots, y_n; \mathbf{0}_n, \mathbf{I}_n) &= \frac{\exp\left[-\frac{1}{2}\sum_{i=1}^n y_i^2\right]}{(2\pi)^{\frac{n}{2}}} \\ &= \prod_{i=1}^n \frac{\exp\left[-\frac{y_i^2}{2}\right]}{\sqrt{2\pi}} \end{aligned}$$

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



(a) (b)



(c) (d)

Figure 1.1: example1.

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
- $Y_1, \dots, Y_n$ are independent *if and only if* $\boldsymbol{\Sigma}$ 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

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

- linear combinations ov MVN random variables (important property): let
  - $\boldsymbol{Y}$ be a n-dimensional Gaussian vector characterized by parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$
  - $\mathbf{A}$ 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 $\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top$.
  This reminds the univariate case where if $Z = a(Y + b)$ then $\mathbb{E}[Z] = a(\mathbb{E}[Y] + b)$ and $\operatorname{Var}[Z] = a^2 \operatorname{Var}[Y]$

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 $\boldsymbol{\Sigma}$ is positive definite, there's a way to obtain $\boldsymbol{\Sigma}^{\frac{1}{2}}$, inverse of which can be thought as square root since $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{\Sigma}^{\frac{1}{2}}$. This matrix will be invertible with inverse $\boldsymbol{\Sigma}^{\frac{1}{2}}$. if $\boldsymbol{\Sigma}$ is invertible. So in this case, by setting the $\mathbf{A}$ and $\mathbf{b}$ as follows:

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

then $\boldsymbol{Z} = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{Y} - \boldsymbol{\mu})$ 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

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

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:

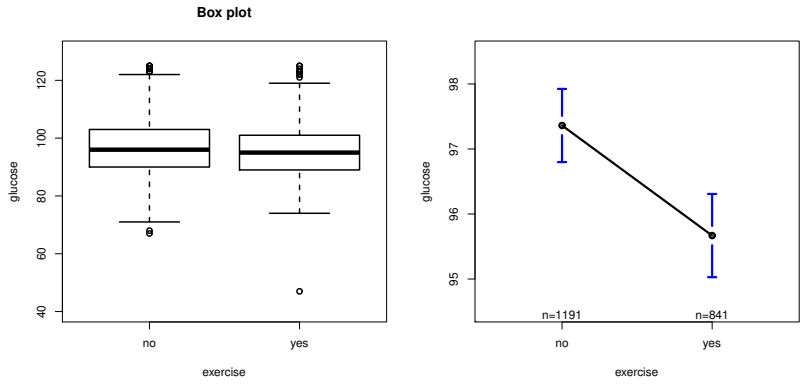
- the **Aim:** Does physical activity (a modifiable factor related to life style) contribute to the reduction of the glucose level, thus preventing a severe disease?
- **available information:** data from an observational study for glucose level and physical activity (yes-no) on a sample of 2032 women 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.

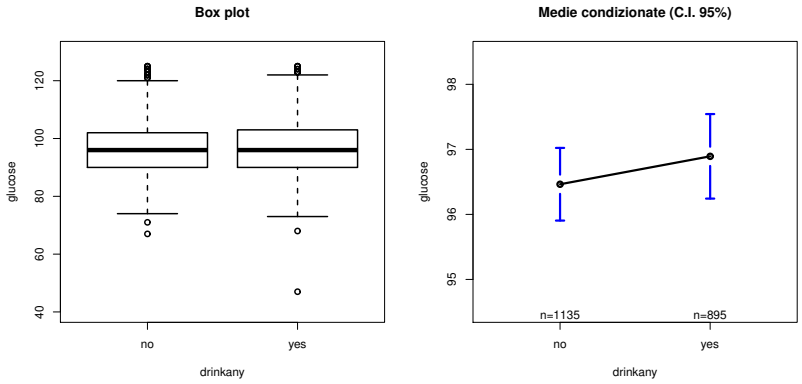
2.1.2 Multiple linear regression

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).

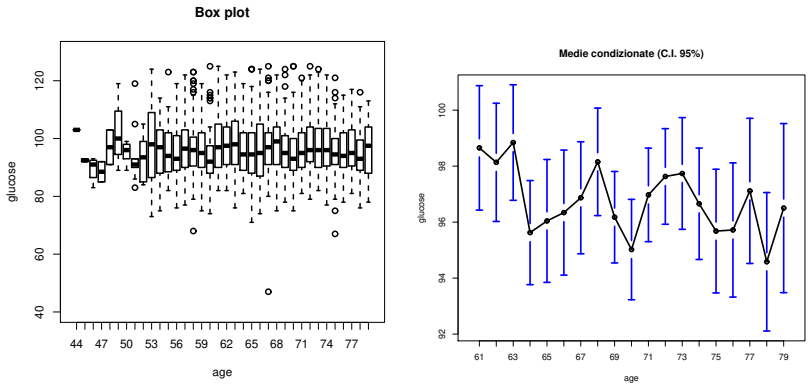
2.1.2.2 Model definition/specification

As done before to put together all



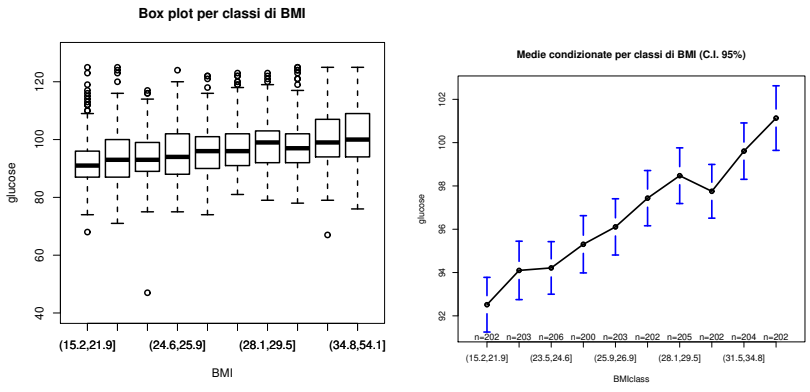
(a) (b)

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

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

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 ($i = 1, \ldots, n$);
- $x_{1i}, x_{2i}, \ldots, x_{pi}$ 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 $x_{1i}, x_{2i}, \ldots, x_{pi}$ assuming:

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

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

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

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

C) the *incorrelation assumption*

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

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

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

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:

- $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^{(p+1)}$: $(p+1)$-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

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

where the first $\mathbb{R}$ is due to the dependent variable, which can take any real value, and $\left\{ (x_{1i}, \ldots, x_{pi})^\top, i = 1, \ldots, n \right\}$ 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 joint density for all the r.vs. $Y_1, \ldots, Y_n$ conditional to the regressor values is

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

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:

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

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

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)

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

  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 ($i = 1, \ldots, n$)
- $x_{1i}, x_{2i}, \ldots, x_{pi}$ 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.

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

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):

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

B) the conditional variance is constant/independent

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

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

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

D) the conditional distribution of the random error is gaussian

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

Putting all things together considering the sample we have that:

- the vector $\boldsymbol{\varepsilon}$ containing all the unit error $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^\top$ ...
- 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:

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

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* $U(\boldsymbol{\beta}) = \dfrac{\partial}{\partial \boldsymbol{\beta}} \ln L(\boldsymbol{\beta}, \sigma^2)$ is the gradient of $l(\boldsymbol{\beta}, \sigma^2)$ with respect to $\boldsymbol{\beta}$ (it's a *vector with* $p+1$ *elements*); in other words ...
- each of its element $U_j(\boldsymbol{\beta}) = \dfrac{\partial}{\partial \beta_j} \ln L(\boldsymbol{\beta}, \sigma^2)$, $j = 0, \ldots, p$ is the first partial derivative of $l(\boldsymbol{\beta}, \sigma^2)$ 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 3. 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 4. 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 5. 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 6. 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 $U(\boldsymbol{\beta}) = \frac{\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{\sigma^2}$ 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

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

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:

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

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

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

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

$$U\left(\hat{\mathbf{b}}\right) = \frac{\partial}{\partial\boldsymbol{\beta}} l\left(\boldsymbol{\beta}, \sigma^2\right)\Big|_{\boldsymbol{\beta}=\hat{\mathbf{b}}} = \mathbf{0}_{p+1}$$

- 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)

$$H\left(\hat{\mathbf{b}}\right) = \frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top} l\left(\boldsymbol{\beta}, \sigma^2\right)\Big|_{\boldsymbol{\beta}=\hat{\mathbf{b}}}$$

  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 $\mathbf{Y}$

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

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

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

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):

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

- 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

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

  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

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

- 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

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

  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**: $e_i^P = \dfrac{e_i}{\sqrt{s^2}} \quad i = 1, \ldots, n$ 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**: $r_i = \dfrac{e_i}{\sqrt{s^2(1-\mathbf{H}_{ii})}} \quad i = 1, \ldots, n$ 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

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

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:

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

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

(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, $\frac{\mathbf{e}^\top\mathbf{e}}{\sigma^2}$:

- is chi squared distributed $\frac{\mathbf{e}^\top\mathbf{e}}{\sigma^2} \sim \chi^2_{n-p-1}$
- 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

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

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:

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

Remark 7. Inferential tools we can adopt for the hypothesis of interest are:

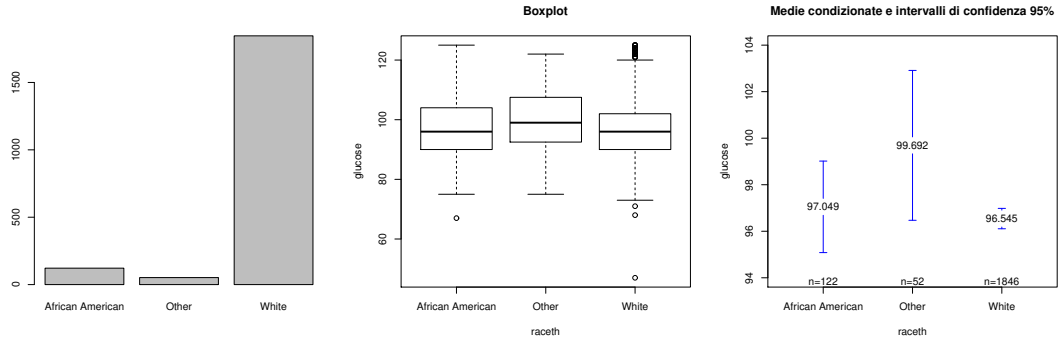
- One-way ANOVA
- Gaussian linear models with indicator/dummy variables

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

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

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

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

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)

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

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

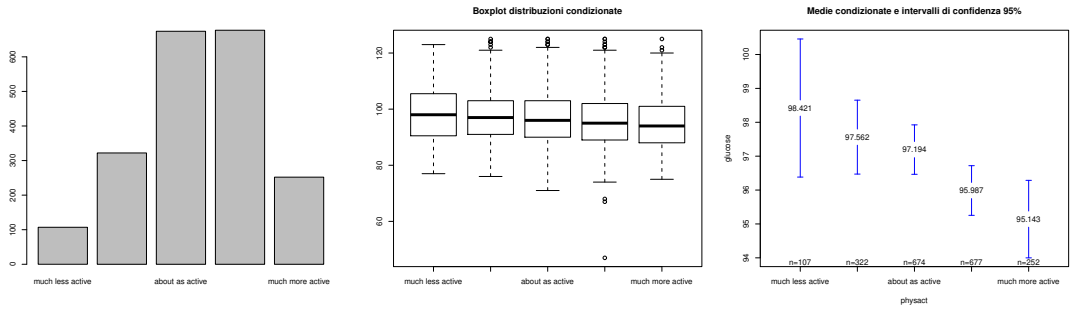
```
> K3
     1    2    3
 1   1   -1    0
 2   1    0   -1

> 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 8. 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

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

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 $\mathbf{K}$ 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

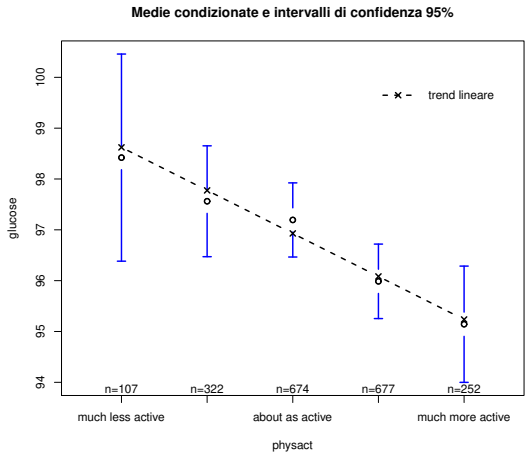


Figure 4.3: Estimated glucose conditional mean by activity level

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

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

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 9. 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:

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

with $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)^\top \in \mathbb{R}^n$.
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 $\mu_1, \mu_2, \ldots, \mu_n$

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

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

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

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

  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

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

  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:

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

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:

$$D = 2 \ln \left[\frac{L\big(\hat{m}_1, \ldots, \hat{m}_n, \sigma^2\big)}{L\Big(\hat{\mathbf{b}}, \sigma^2\Big)} \right] = 2 \left[l\big(\hat{m}_1, \ldots, \hat{m}_n, \sigma^2\big) - l\Big(\hat{\mathbf{b}}, \sigma^2\Big) \right]$$

$$= 2 \left[-\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 + \frac{n}{2} \ln 2\pi + \frac{n}{2} \ln \sigma^2 + \frac{\mathbf{e}^\top \mathbf{e}}{2\sigma^2} \right] = \frac{\mathbf{e}^\top \mathbf{e}}{\sigma^2}$$

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

Remark 10. 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

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

  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 11. In some models we'll be able to exploit residual deviance to perform goodness of fit test

Remark 12. 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):

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

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:

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

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

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 $\frac{\mathbf{e}^\top \mathbf{e}}{\sum_{i=1}^n (y_i - \bar{y})^2}$ goes down as usual
- the factor $\frac{n-1}{n-p-1}$ 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* $\frac{n-1}{n-p-1}$

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 $\frac{\mathbf{e}^\top \mathbf{e}}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1$ and $R^2_{adj} = 1 - \frac{n-1}{n-p-1} \leq 0$. 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 $\frac{n-1}{n-p-1} \geq 1$, while the traditional R^2 cannot be negative.

Remark 15. 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 16. 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 $\mathrm{E}[Y_i|\mathbf{x}_i]$ 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 17. 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 $\mathrm{E}[Y_i|\mathbf{x}_i]$ 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 18. 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 $f_0(\mathbf{y})$
- $\mathcal{F}_A = \left\{f_A(\cdot;\boldsymbol{\theta}_A), \boldsymbol{\theta}_A \in \boldsymbol{\Theta}_{\boldsymbol{A}} \subseteq \mathbb{R}^{k_A}\right\}$ 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

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

and so

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

Here above $k = p + 2$ (p betas + intercept + σ^2).
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)

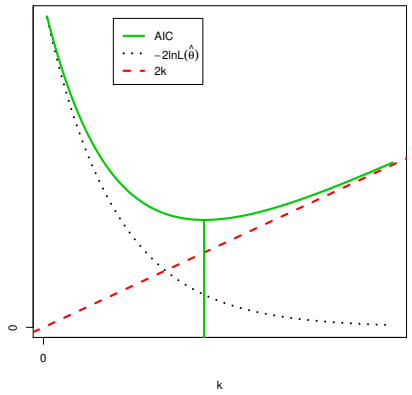


Figure 5.1: AIC components

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

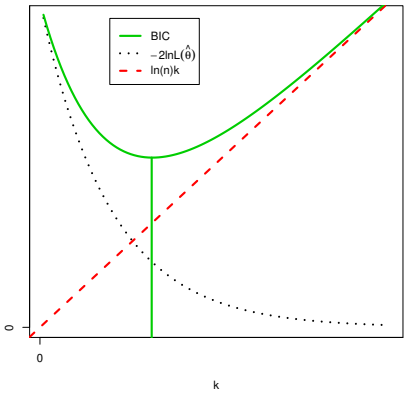


Figure 5.2: BIC components

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

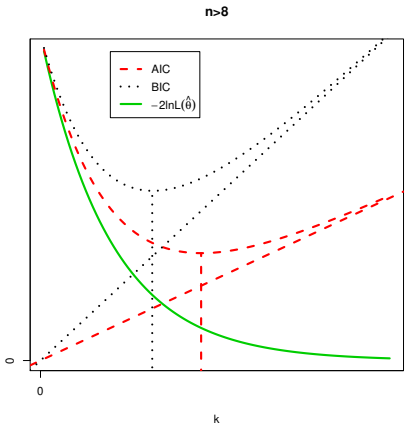


Figure 5.3: AIC and BIC components

Chapter 6

Lab1

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

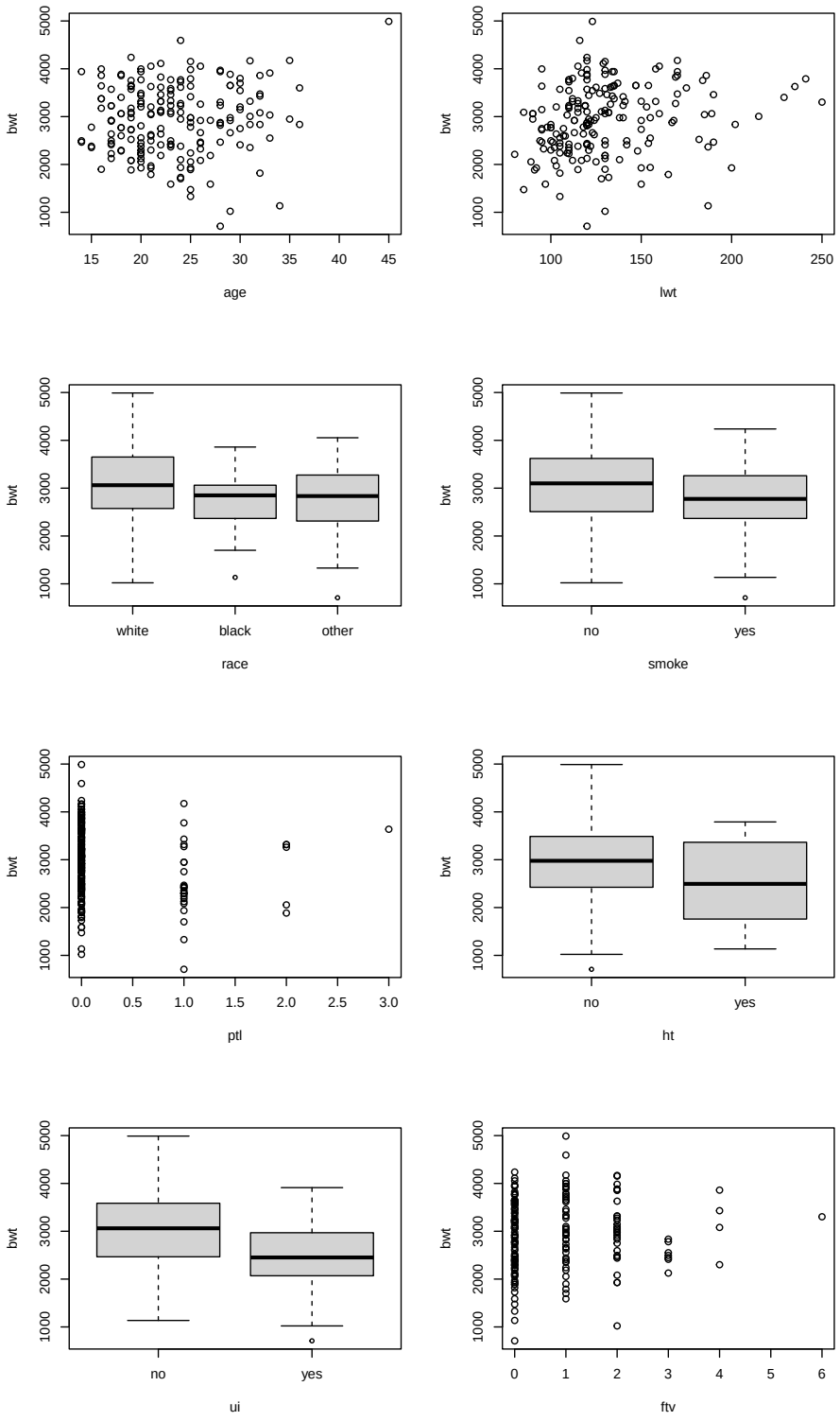
- `bwt` Dependent variable, weight of the baby at birth (grams)
- `age` mother's age
- `lwt` mother's weight before pregnancy (in pounds, lbs)
- `race` ethnicity (1=white, 2=black, 3=other)
- `smoke` smoking habit of the mother (0=no, 1=yes)
- `ptl` number of premature pregnancies mother had before the 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
age
lwt
race
white
black
other
smoke
no
yes
ptl
ht
ui
ftv

```r
## 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

```r
## 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)
##
## Deviance 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
##
## (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)))
```

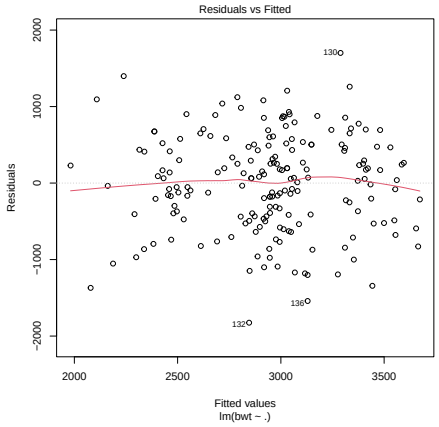
6.2.1 Linearity in the regressors

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

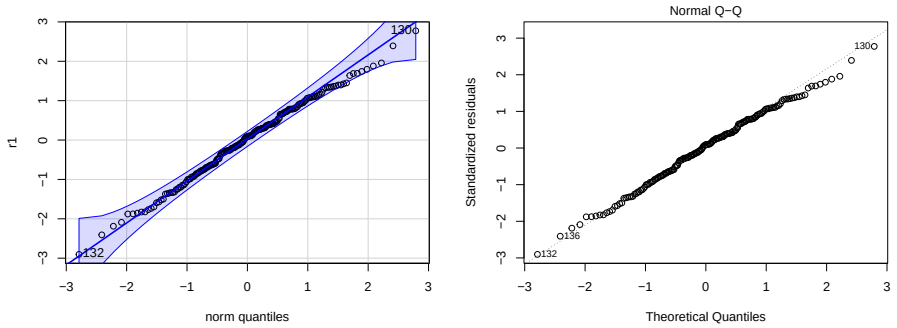
```
## ## residuals vs fitted
## plot(yih, e1) ## by hand
plot(model1, which = 1) ## ?plot.lm: 1 is residual vs fitted
```



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
```

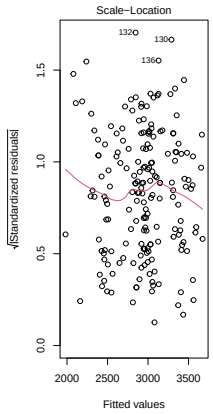


6.2.3 Homoscedasticity

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 is.data.frame(data):  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 19. 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

## bwt 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
```

```r
## 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

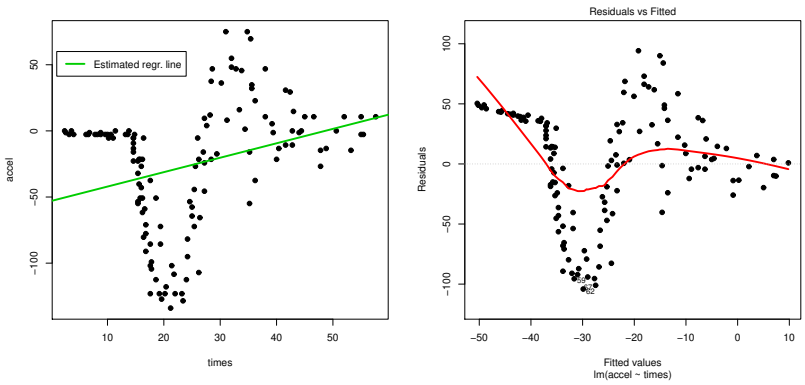
```
              Estimate   Std.  Error   t value   Pr(>|t|)
(Intercept)    -53.008         8.712    -6.084      0.000
      times      1.091         0.307     3.552      0.001

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

$$\mathbb{E}[\mathbf{Y}|\mathbf{X}] = \mathbf{X}\boldsymbol{\beta}$$
$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{b}}$$
$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}, \quad \mathbb{E}[\mathbf{e}|\mathbf{X}] = \mathbf{0}$$

  If the functional relashionship $\mathbb{E}[\mathbf{Y}|\mathbf{X}] = \mathbf{X}\boldsymbol{\beta}$ is correctly specified then $\mathbb{E}[\mathbf{e}|\mathbf{X}] = \mathbf{0}$ 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 20. In situation like this we can work on the assumptionsspecifying the functional form linking the value of the regressors to the expected value of $\mathbf{Y}$

Remark 21. 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

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

And the conditional expected values in compact form:

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

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

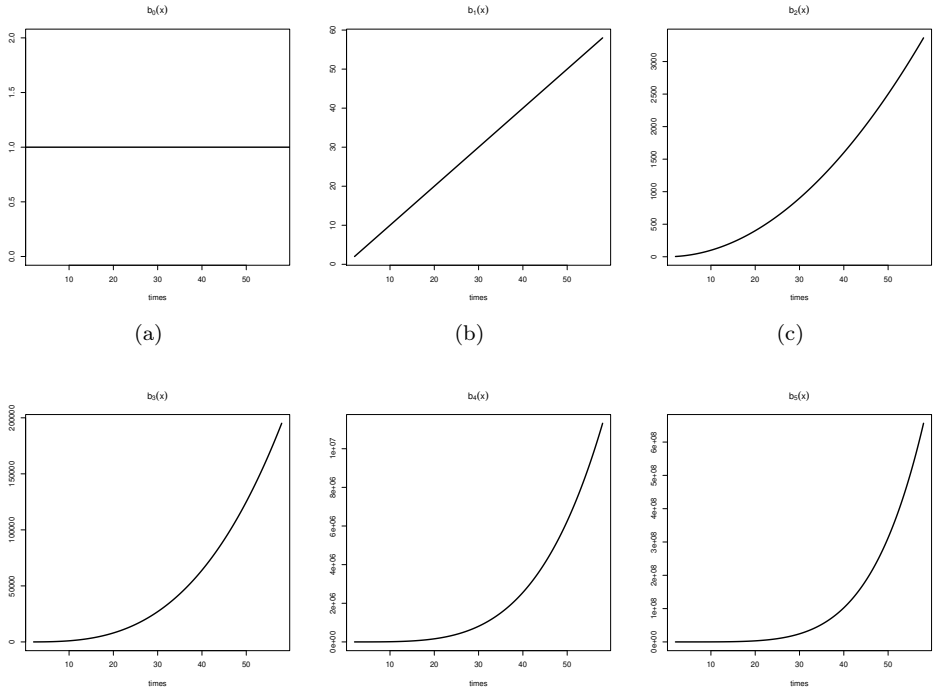
so the conditional expected value is stilla a $\mathbf{X}\boldsymbol{\beta}$ formulation (having considered each power as separate regressor (so from a notational pov there's no difference between a multiple regression model and a polynomial regression model, the only differences is the meaning of column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

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

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 $x_i^0, \ldots x_i^5$.

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:

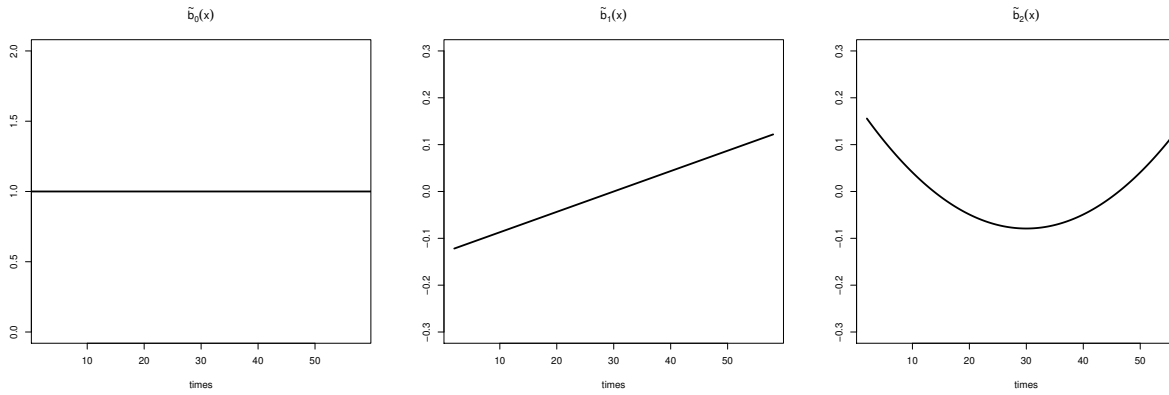
- Original polynomial:

```
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
```

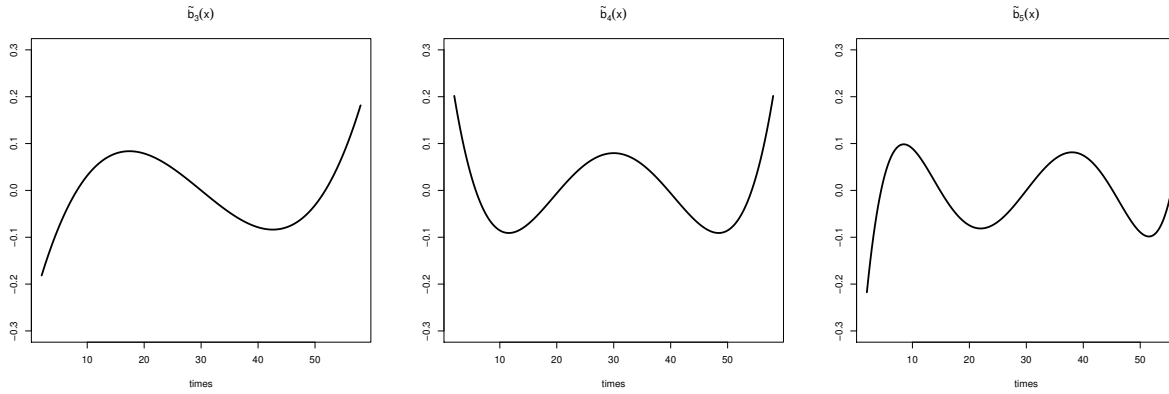
- Orthogonal polynomial:

```
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.

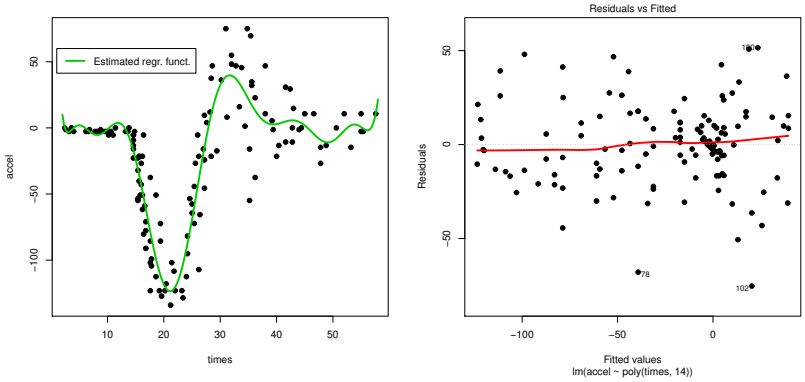
	R_j^2
linear term	0.08785
quadratic term	0.05587
cubic term	0.18660
quartic term	0.00014
quintic term	0.19600
total	0.5264

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*)

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



(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:

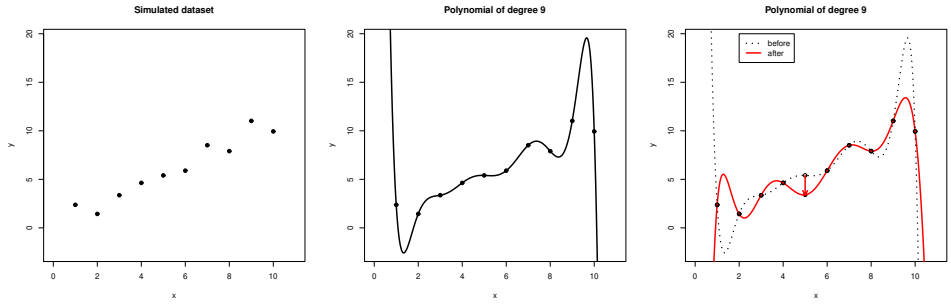
$$\Delta D | \, M_B \sim \chi^2_q$$
$$\left. \frac{\Delta D}{D_{M_A}} \frac{n-p_A-1}{q} \right| M_B \sim F_{(q,n-p_A-1)}$$

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 fourteenth 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 22. 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 23. 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.

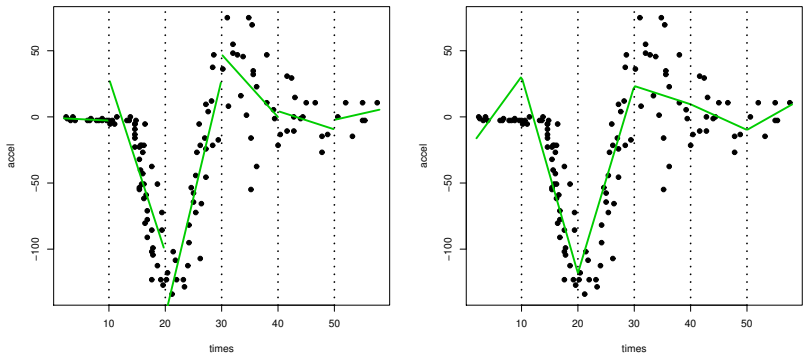
Important remark 35. The total number of free parameters of a continuous piecewise linear function is given by $2 \cdot (K+1) - K = 2K + 2 - K = K + 2$ (*2 parameters for each interval* $-$ *K restriction/constraints* to guarantee continuity)

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 24. 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 $l_1 < l_2 < \ldots, < l_K$ 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

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

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 ...

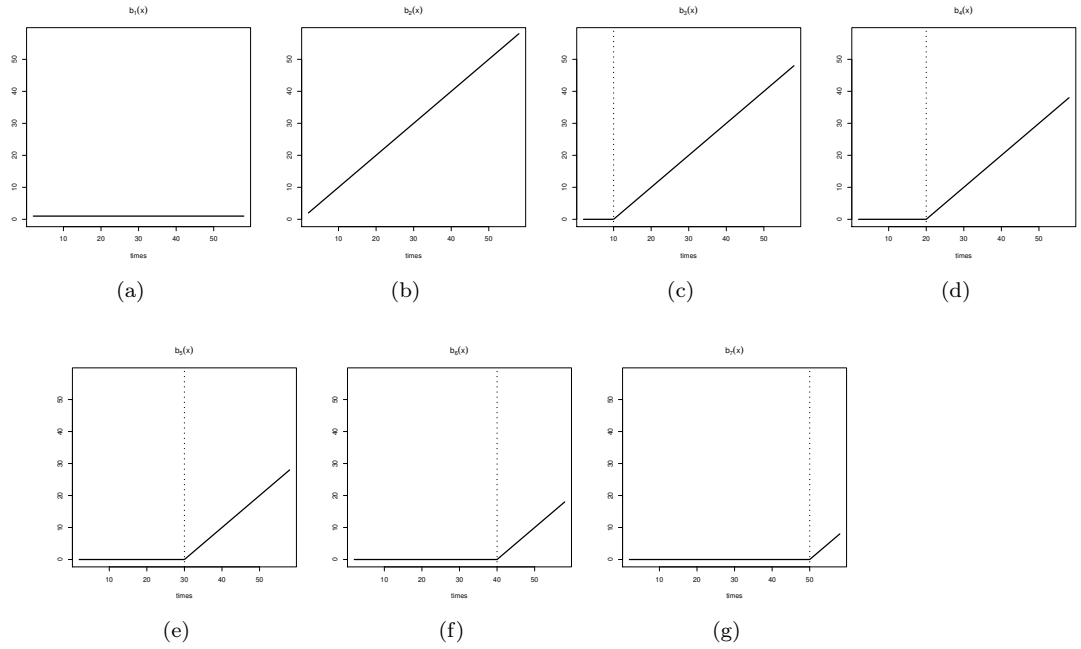
Important remark 36. With piecewise constant stuff we have continuity but the function is not derivable/well behaved in the knots; here comes the spline functions.

Remark 25. 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 $l_1 < l_2 < \ldots, < l_K$ 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:

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

- 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 $\frac{\partial^0}{\partial x^0} h(x) = h(x)$)

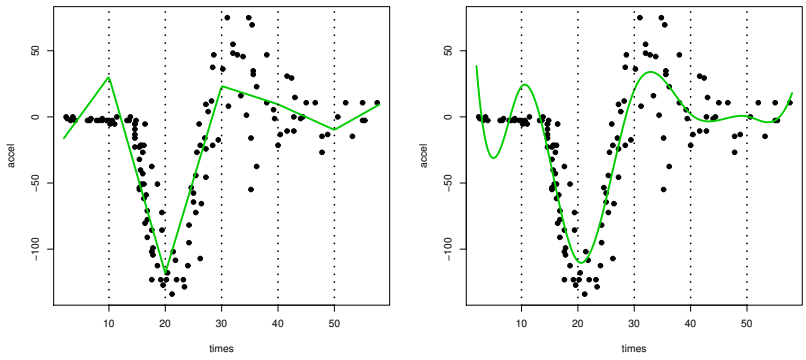
Important remark 37 (Number of parameters). Considering that

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

Remark 27. 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 28. 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 29. 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 30. These are the most simple basis; but suffers from some issues.

Remark 31. 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 32. 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

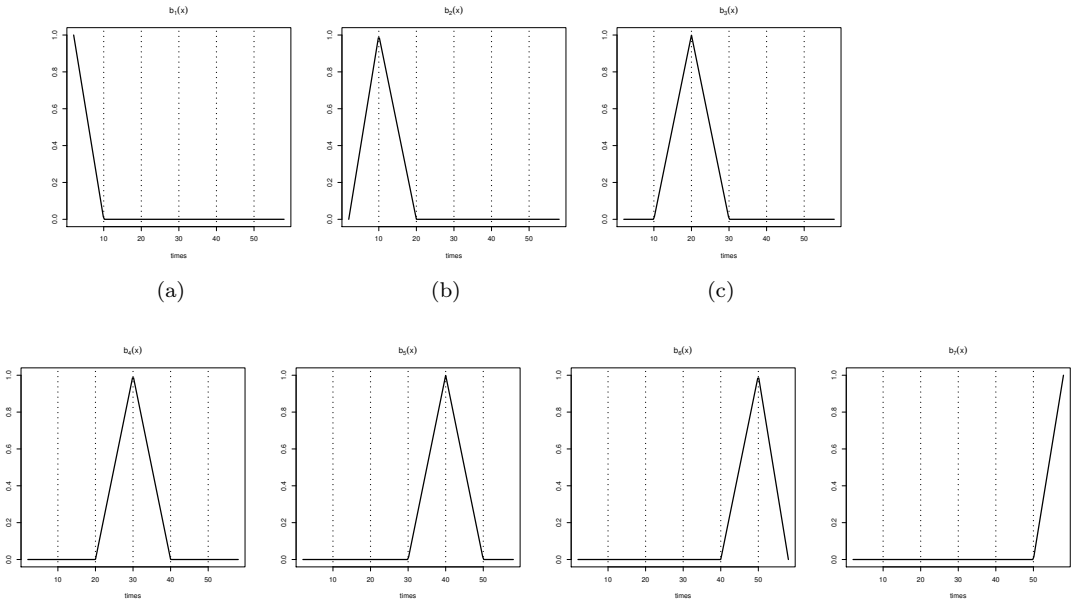
7.5.4 Concluding remarks

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

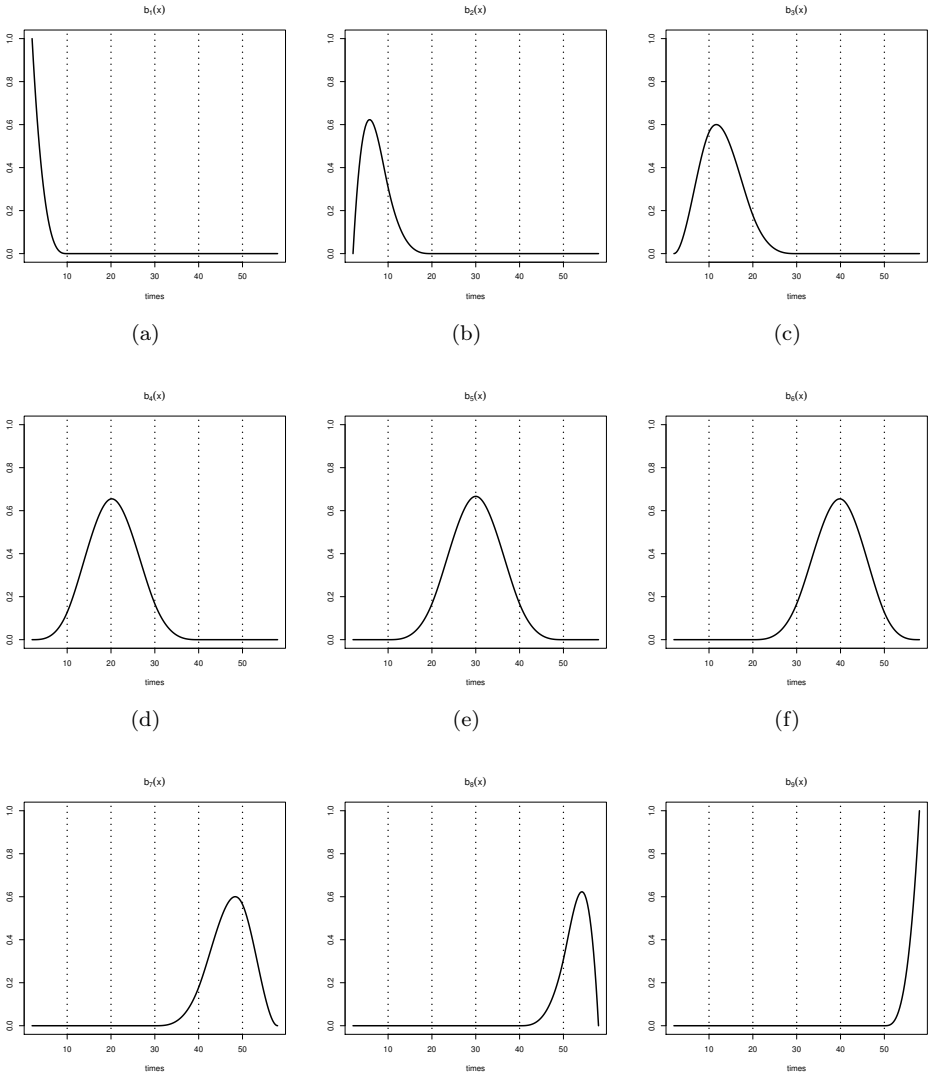
Remark 33. When using splines we have to decide both number and location

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

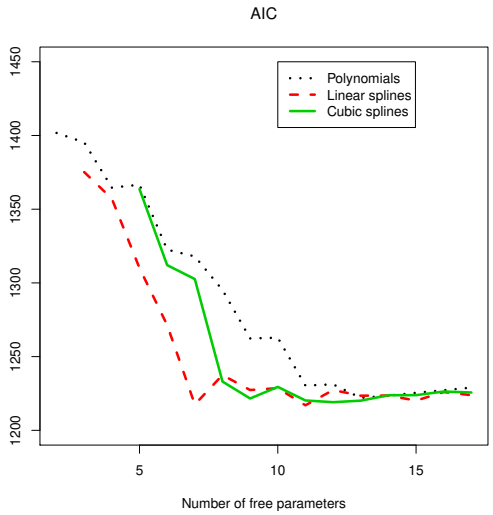


Figure 7.11: Splines comparison

- 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.

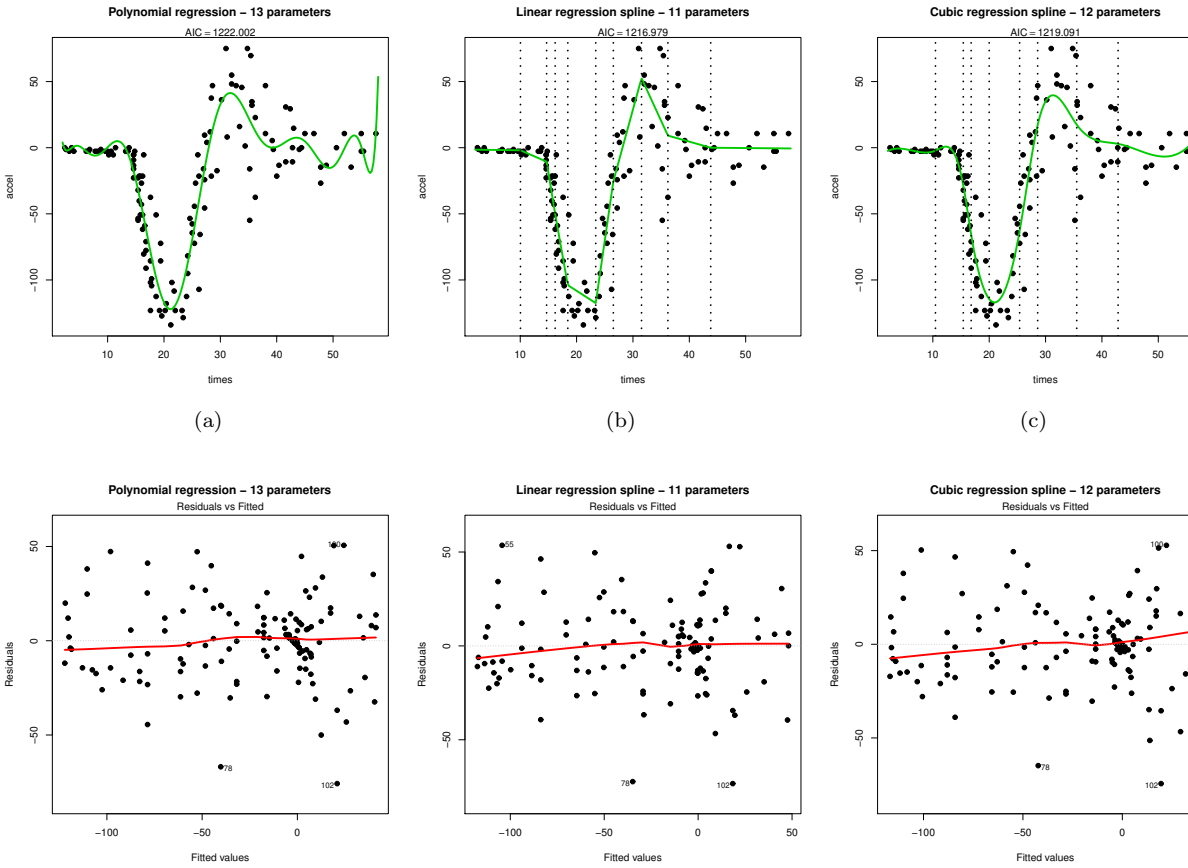


Figure 7.12: Crash - Best splines models and their residuals

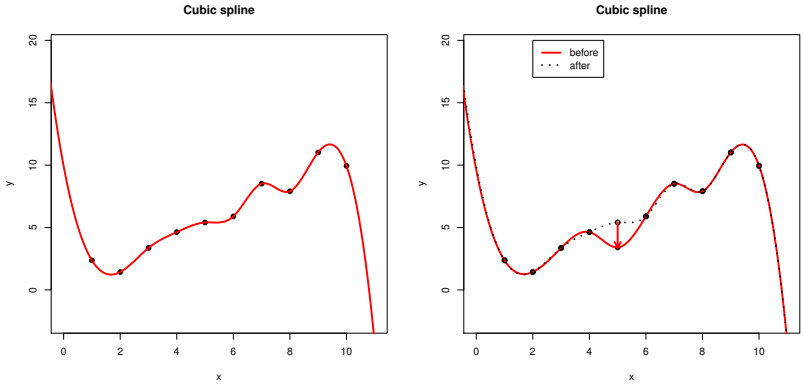


Figure 7.13: Poly/splines comparison

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 34. 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 35. 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 $(i = 1, \ldots, n)$.*

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 36. For this result the penalized LS approach described previously is also known as **smoothing spline** approach

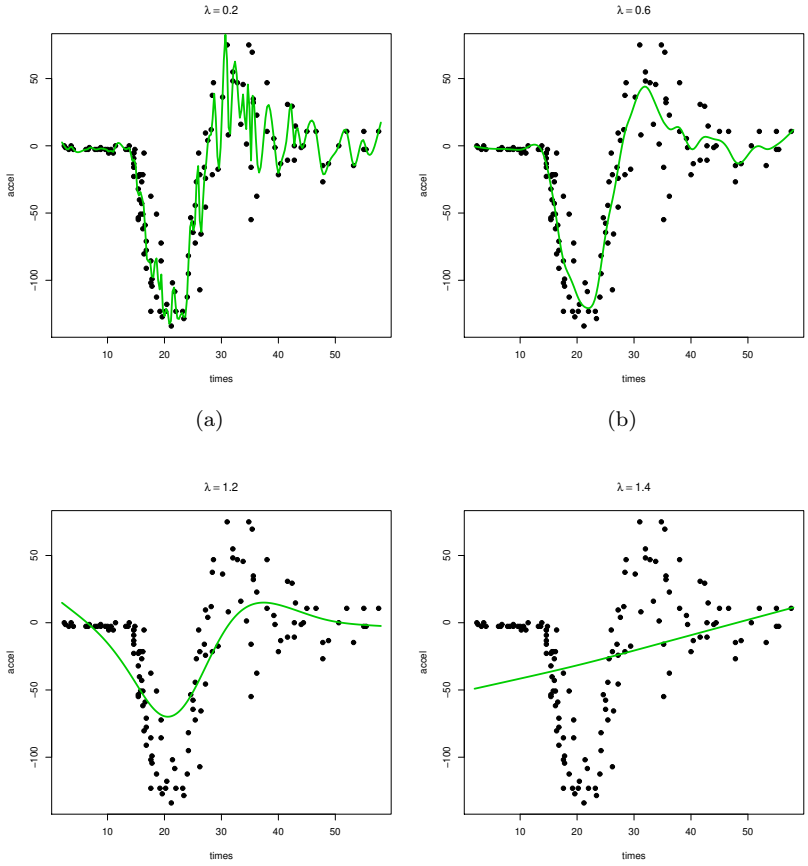
Remark 37. Smoothing splines differ from regression splines due to the presence of the penalty term $\int \left[h''(t)\right]^2 dt$, that implicitly introduce constraints on the parameters of the spline function. The strength of these constraints is controlled by the smoothing parameter λ

8.1.4 Natural cubic splines

Definition 8.1.1. A function $h(x)$ is a natural cubic spline with fixed knots $l_1 < l_2 < \ldots, < l_K$ if it is *a cubic spline function with the additional constraints* that:

$$h(x) = \beta_{01} + \beta_{11} x, \quad \text{if } x < l_1$$
$$h(x) = \beta_{0K} + \beta_{1K} x, \quad \text{if } x > l_K$$

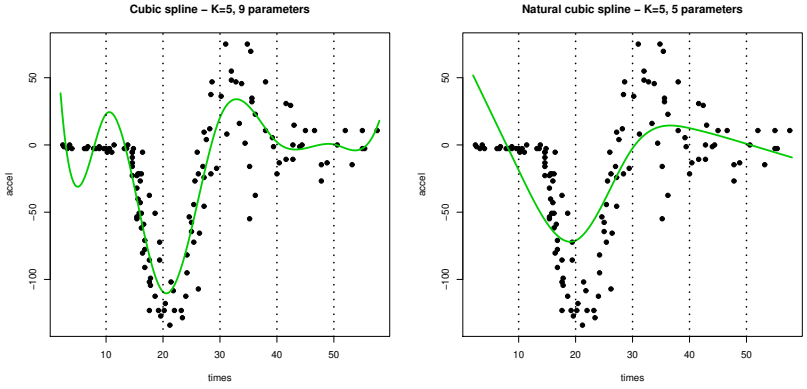
Remark 38. So:



(a) (b)

(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 39. 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 $\boldsymbol{\theta}$

  $$\hat{\mathbf{t}}_\lambda = \underset{\mathbf{t} \in \mathbb{R}^n}{\arg\min}\, pls_\lambda(\mathbf{t}) = \left(\mathbf{N}^\top \mathbf{N} + \lambda \mathbf{P}\right)^{-1} \mathbf{N}^\top \mathbf{y}$$

  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* $\mathbf{y}$ *(premultiplied by a matrix) given by*

$$\begin{bmatrix} \hat{h}_\lambda(x_1) \\ \vdots \\ \hat{h}_\lambda(x_n) \end{bmatrix} = \hat{\mathbf{h}}_\lambda = \mathbf{N}\hat{\mathbf{t}}_\lambda = \underbrace{\mathbf{N}\Big(\mathbf{N}^\top\mathbf{N} + \lambda\mathbf{P}\Big)^{-1}\mathbf{N}^\top}_{\mathbf{S}_\lambda}\mathbf{y}$$

Remark 40. $\hat{\mathbf{h}}_\lambda$ 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 41. Now we consider how to choose λ.

8.1.6 Choice of the smoothing parameter

Remark 42. 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 43. 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

$$LOOCV(\lambda) = \frac{1}{n}\sum_i \Big(y_i - \hat{h}_\lambda^{[-i]}(x_i)\Big)^2$$

  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.*

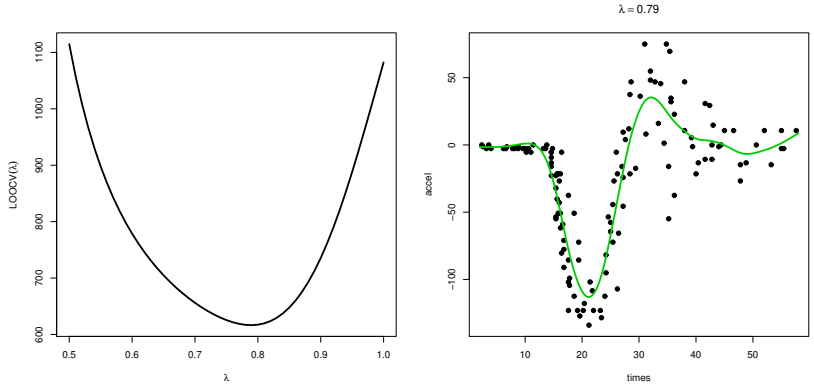
Example 8.1.4 (Crash test data - optimal value for λ - *LOOCV*)**.** In figure 8.3 on the left the plot describing the error as a function of lambda (from 0.5 to 1, selected by trial and error) and on the right the model estimated with best lambda.

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):

$$GCV(\lambda) = \frac{1}{n}\sum_i \left(\frac{y_i - \hat{h}_\lambda(x_i)}{1 - \dfrac{\mathrm{Tr}(\mathbf{S}_\lambda)}{n}}\right)^2 = \frac{n}{(n - \mathrm{Tr}(\mathbf{S}_\lambda))^2}\sum_i \Big(y_i - \hat{h}_\lambda(x_i)\Big)^2$$

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

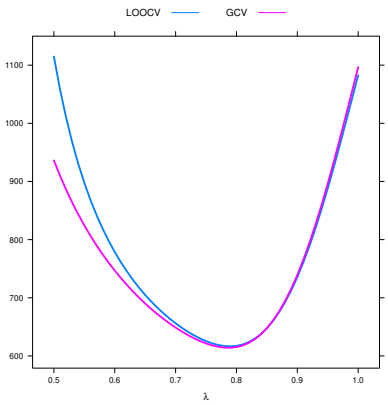


Figure 8.4: Lambda: GCV vs LOOCV

- 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 44. 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 45. 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)**.**

$$pl_\lambda(\boldsymbol{\theta}, \sigma^2) \propto -\frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}) - \frac{\lambda}{2}\boldsymbol{\theta}^\top\mathbf{P}\boldsymbol{\theta} \tag{8.1}$$

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

  $$U_\lambda(\boldsymbol{\theta}) = \frac{\partial}{\partial\boldsymbol{\theta}} pl_\lambda(\boldsymbol{\theta}, \sigma^2) = \frac{\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})}{\sigma^2} - \lambda\mathbf{P}\boldsymbol{\theta} = U(\boldsymbol{\theta}) - \lambda\mathbf{P}\boldsymbol{\theta}$$

  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

  $$i_\lambda(\boldsymbol{\theta}) = I_\lambda(\boldsymbol{\theta}) = -\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top} pl_\lambda(\boldsymbol{\theta}, \sigma^2) = \frac{\mathbf{X}^\top\mathbf{X}}{\sigma^2} + \lambda\mathbf{P} = I(\boldsymbol{\theta}) + \lambda\mathbf{P}$$

  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 $pl_\lambda(\boldsymbol{\theta}, \sigma^2)$ 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 46. 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:

$$s^2 = \frac{\sum_i \left(y_i - \hat{h}_{\tilde{\lambda}}(x_i)\right)^2}{n - \mathrm{Tr}(\mathbf{S}_{\tilde{\lambda}})}$$

Some authors suggest replacing the effective degrees of freedom $\mathrm{Tr}(\mathbf{S}_{\tilde{\lambda}})$ with the equivalent number of parameters: $2\mathrm{Tr}(\mathbf{S}_{\tilde{\lambda}}) - \mathrm{Tr}(\mathbf{S}_{\tilde{\lambda}}\mathbf{S}_{\tilde{\lambda}})$

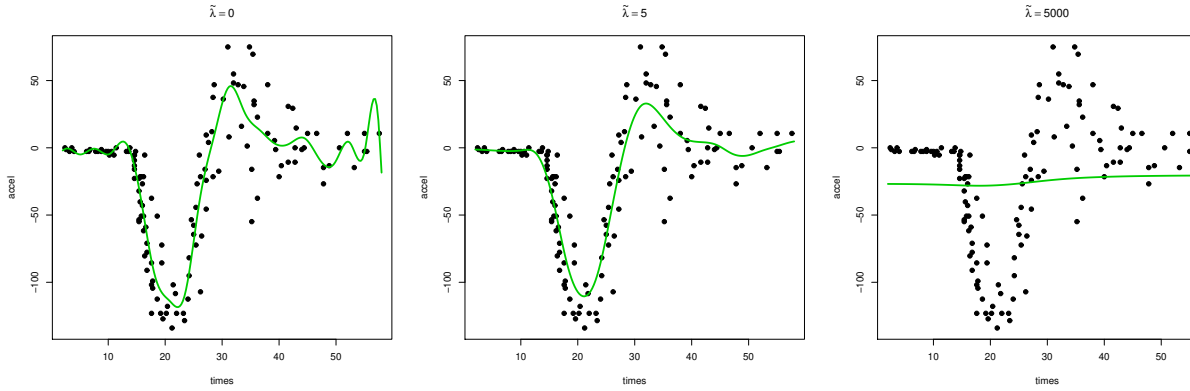
Remark 47. In both cases, when $\lambda = 0$ the trace and this second approach will end with $K+4$ as complexity

8.2.6 Choice of the smoothing parameter

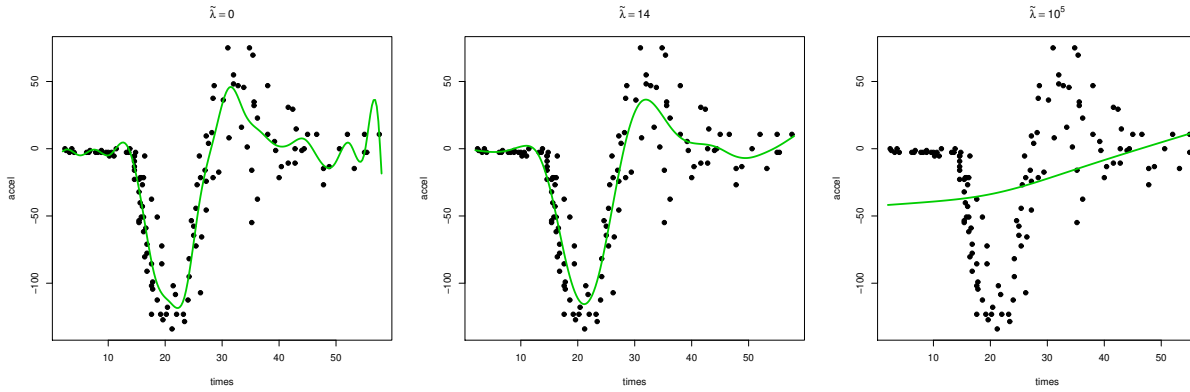
Remark 48. The selection of the optimal value for $\tilde{\lambda}$ can be based on a model selection criterion such as *LOOCV* (it is not necessary to refit the model n times), *GCV* (generalized cv) or here we can use *AIC* or *BIC* because we have a parametric definition and a likelihood; in these 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

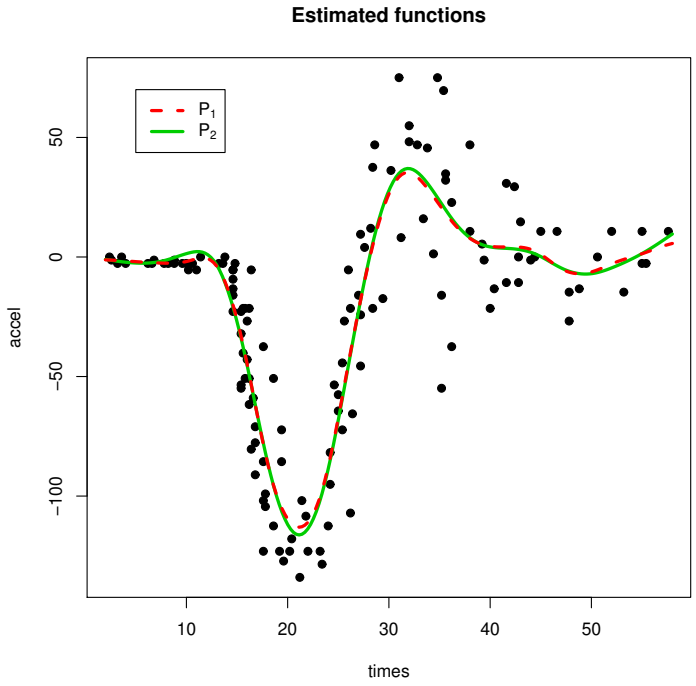


Figure 8.6

- the number of parameter can be obtained as

$$\mathrm{Tr}\big(\mathbf{S}_{\tilde{\lambda}}\big) + 1$$

(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	$\tilde{\lambda}$	$\mathrm{Tr}\big(\mathbf{S}_{\tilde{\lambda}}\big)$	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:

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

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 49. however the distributional result in 8.2 can be exploited to draw approximate inferential conclusions about $h(\cdot)$

8.2.8 Hypothesis testing

Remark 50. 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):

$$H_0\text{: } h(\cdot) \text{ is a constant function}$$

- *linearity* assumption (basically we test if a nonlinear/complex structure is absolutely needed)

$$H_0\text{: } h(\cdot) \text{ is a linear function}$$

  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)

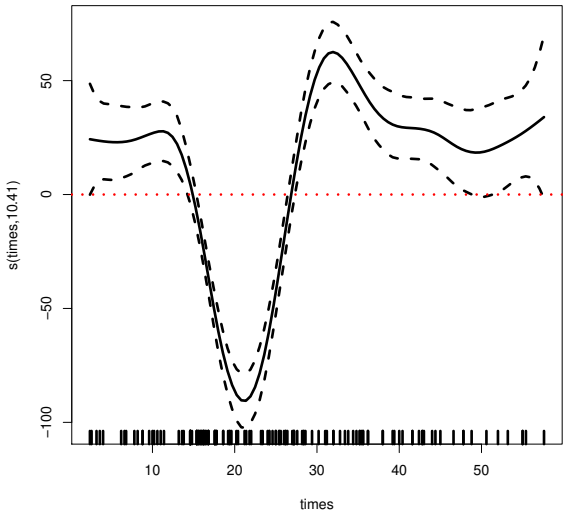


Figure 8.7: Crash test data - P-splines

- 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 ($\Rightarrow$ 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 $\mathrm{Tr}\left(\mathbf{S}_{\tilde{\lambda}}\right) - 1$ 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
```

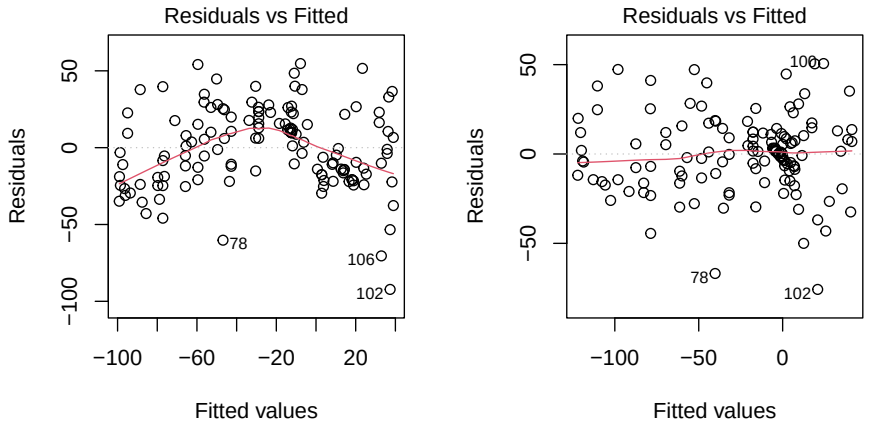
We can compare nested models (say if 12 degrees are needed or 9 are sufficient): we note that the choice of the bases does not alter the results of the F test which is the same. We have that 12 are needed becaus there's significant difference between the twos (significant worsening from 12 to 9 in the 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 51. 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.

```
library(mgcv)

## Caricamento del pacchetto richiesto:  nlme
## This is mgcv 1.9-1.  For overview type 'help("mgcv-package")'.

## linear splines with 20 equispaced knots + penalty based on the squared
## first-order differences; k = 20 + 1 + 1 = 22
summary(psplinel <- gam(accel ~ s(times, bs = "ps", k = 22, m = c(0, 1)),
                        data = mcycle))

##
## Family: gaussian
## Link function: identity
```

```
##
## Formula:
## accel ~ s(times, bs = "ps", k = 22, m = c(0, 1))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -25.546      1.974  -12.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F p-value
## s(times) 12.51     21 21.28  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.778   Deviance explained = 79.9%
## GCV = 576.84  Scale est. = 518.26    n = 133
```

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))

##
## Family: gaussian
## Link function: identity
##
## Formula:
## accel ~ s(times, bs = "ps", k = 24, m = c(2, 2))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -25.546      1.966  -12.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df    F p-value
## s(times) 10.41  12.44 37.2  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.78   Deviance explained = 79.7%
## GCV = 562.23  Scale est. = 513.98    n = 133
```

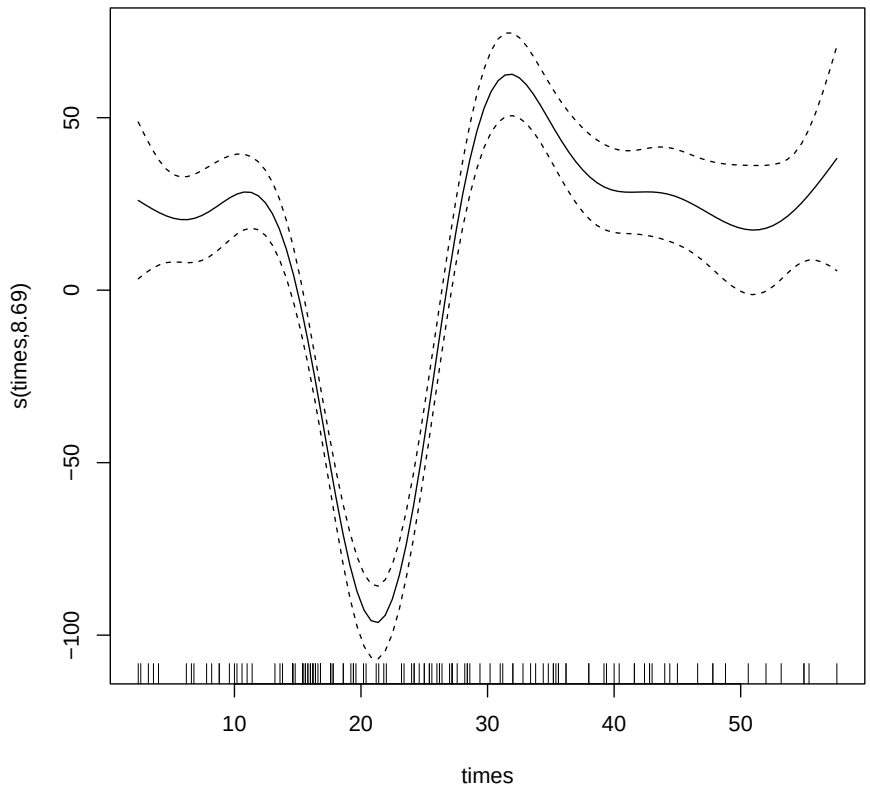
In this second case GCV is slightly better than the previous. What happens if we use the default parameters?

```
## default setting
summary(gam1 <- gam(accel ~ s(times), data = mcycle))

##
## Family: gaussian
## Link function: identity
##
## Formula:
```

```
## accel ~ s(times)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -25.546      1.951   -13.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##            edf Ref.df     F p-value
## s(times) 8.693  8.972 53.52  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.783   Deviance explained = 79.8%
## GCV = 545.78  Scale est. = 506       n = 133
```

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)
```



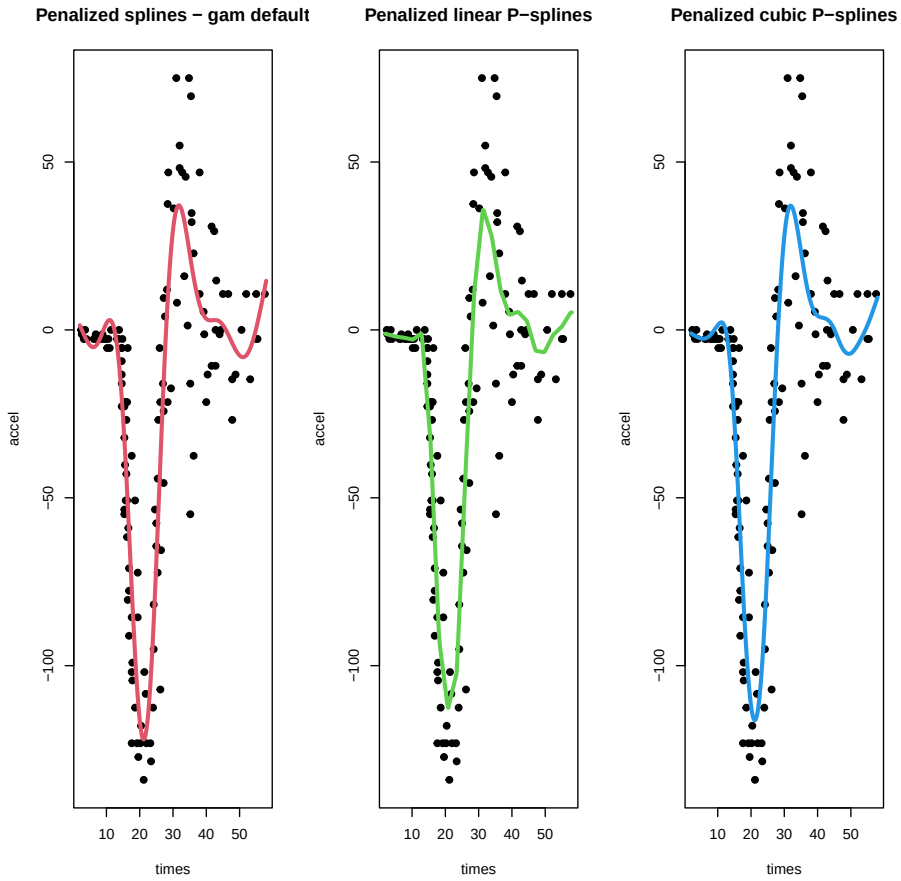
9.2.4 Comparison and wrapup

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)

## [1] 1223.521

AIC(psplinec)

## [1] 1220.542

AIC(gam1)

## [1] 1216.889
```

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")
pred.psplinel <- predict(psplinel, newdata = new_data, type = "response")
pred.psplinec <- predict(psplinec, newdata = new_data, type = "response")

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)
plot(mcycle$times, mcycle$accel, xlab = "times", ylab = "accel",
     pch = 19, main = "Penalized linear P-splines")
lines(timesp, pred.psplinel, lwd = 3, col = 3)
plot(mcycle$times, mcycle$accel, xlab = "times", ylab = "accel",
     pch = 19, main = "Penalized cubic P-splines")
lines(timesp, pred.psplinec, lwd = 3, col = 4)
```



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

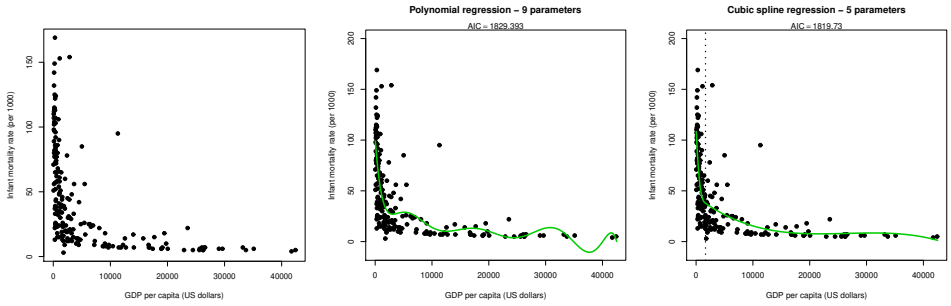
Variable transformations

Remark 52. 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:

$$Y_i = \alpha x_i^{\beta_1} \exp(\varepsilon_i), \quad \varepsilon_i | x_i \sim N\left(0, \sigma^2\right) \text{ IID}$$

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 53. What are the interesting features of this specific moltiplicative model? Although the considered model

$$Y_i = h(x_i, \varepsilon_i; \boldsymbol{\theta}) = \alpha x_i^{\beta_1} \exp(\varepsilon_i), \quad \varepsilon_i | x_i \sim N\left(0, \sigma^2\right) \text{ IID}$$

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

$$g(Y_i) = \beta_0 + \beta_1 b(x_i) + \varepsilon_i, \quad \varepsilon_i | x_i \sim N\left(0, \sigma^2\right) \text{ IID}$$

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 54. 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:
```

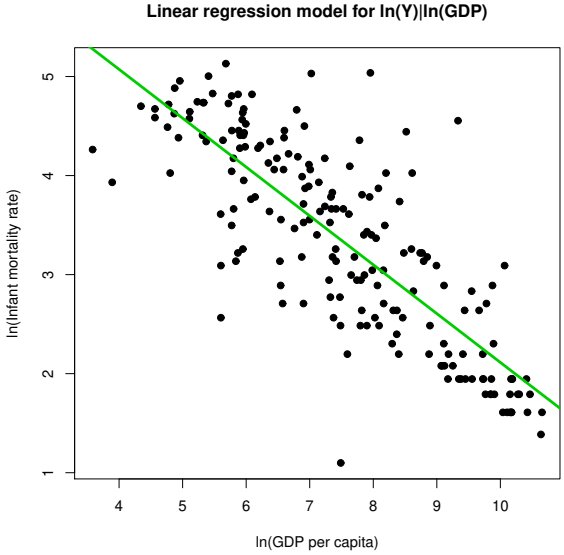


Figure 10.2: lg(mortality) vs log(gdp)

```
                  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 55. 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 56. 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

$$f(y_i; \mu_i, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{y_i} \exp\left[-\frac{(\ln y_i - \mu_i)^2}{2\sigma^2} \right]$$

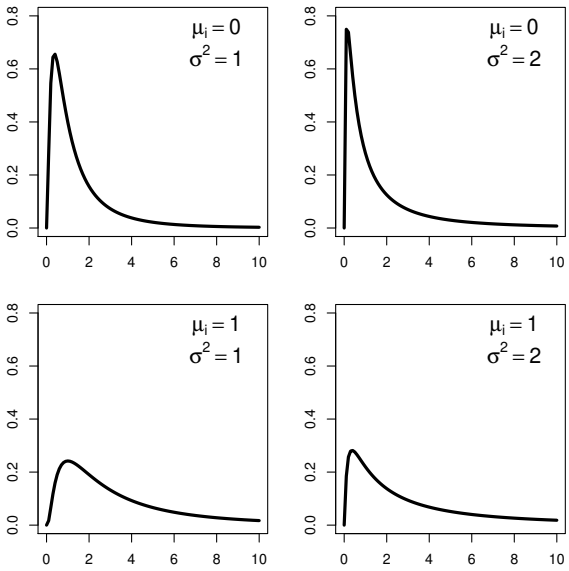


Figure 10.3: Lognormal shapes

where:

$$y_i \in (0, +\infty)$$
$$\mu_i \in (-\infty, +\infty)$$
$$\sigma^2 \in (0, +\infty)$$

Remark 57 (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 58 (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*:

$$\begin{aligned}\operatorname{Var}[Y_i] &= \exp\left[2(\mu_i+\sigma^2)\right]\left[1-\exp\left(-\sigma^2\right)\right]\\ &= \{\mathrm{E}[Y_i]\}^2\left[\exp\left(\sigma^2\right)-1\right]\\ &\propto \{\mathrm{E}[Y_i]\}^2\end{aligned}$$

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):

$$\mathrm{CV}[Y_i] = \frac{\sqrt{\operatorname{Var}[Y_i]}}{\mathrm{E}[Y_i]} = \frac{\mathrm{E}[Y_i]\sqrt{\exp(\sigma^2)-1}}{\mathrm{E}[Y_i]} = \sqrt{\exp(\sigma^2)-1}$$

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 59. 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 ...

$$\ln Y_i|x_{1i}\dots x_{pi} \sim N(\beta_0+\beta_1x_{1i}+\ldots+\beta_px_{pi},\sigma^2), \quad i=1,\dots,n \text{ independent}$$

implies ($\Longleftrightarrow$ actually) assuming a lognormal regression model for the original variable (it's a model with conditional lognormal distributions):

$$Y_i|x_{1i}\dots x_{pi} \sim \ln N(\beta_0+\beta_1x_{1i}+\ldots+\beta_px_{pi},\sigma^2), \quad i=1,\dots,n \text{ independent}$$

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)

$$\begin{aligned}\mathrm{E}[Y_i|x_{1i}\dots x_{pi}] &= \exp\left(\beta_0+\beta_1x_{1i}+\ldots+\beta_px_{pi}+\frac{\sigma^2}{2}\right)\\ &> \exp(\beta_0+\beta_1x_{1i}+\ldots+\beta_px_{pi}) = \exp(\mathrm{E}[\ln Y_i|x_{1i}\dots x_{pi}])\\ \operatorname{Var}[Y_i|x_{1i}\dots x_{pi}] &\propto \{\mathrm{E}[Y_i|x_{1i}\dots x_{pi}]\}^2\end{aligned}$$

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 60. 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 61. 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

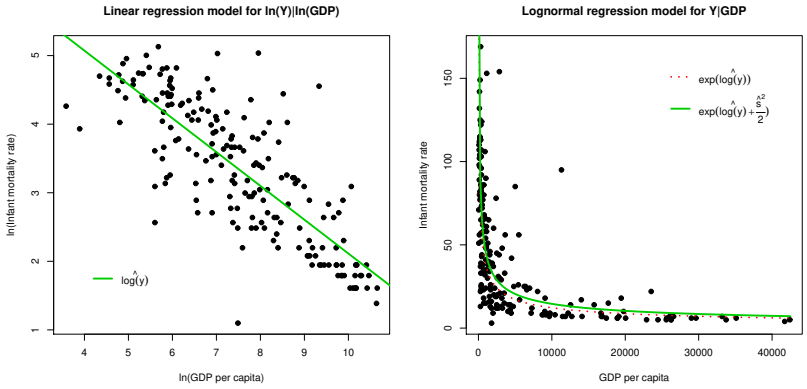
- the estimation for the logarithmic transformation:

$$\widehat{\ln y_i} = \mathrm{E}[\ln Y_i | \widehat{x_{1i} \ldots x_{pi}}] = \hat{b}_0 + \hat{b}_1 x_{1i} + \ldots + \hat{b}_p x_{pi}$$

- 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 $\exp\left[\widehat{\ln y_i}\right]$ $(i = 1, \dots, n)$ 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 62. 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)

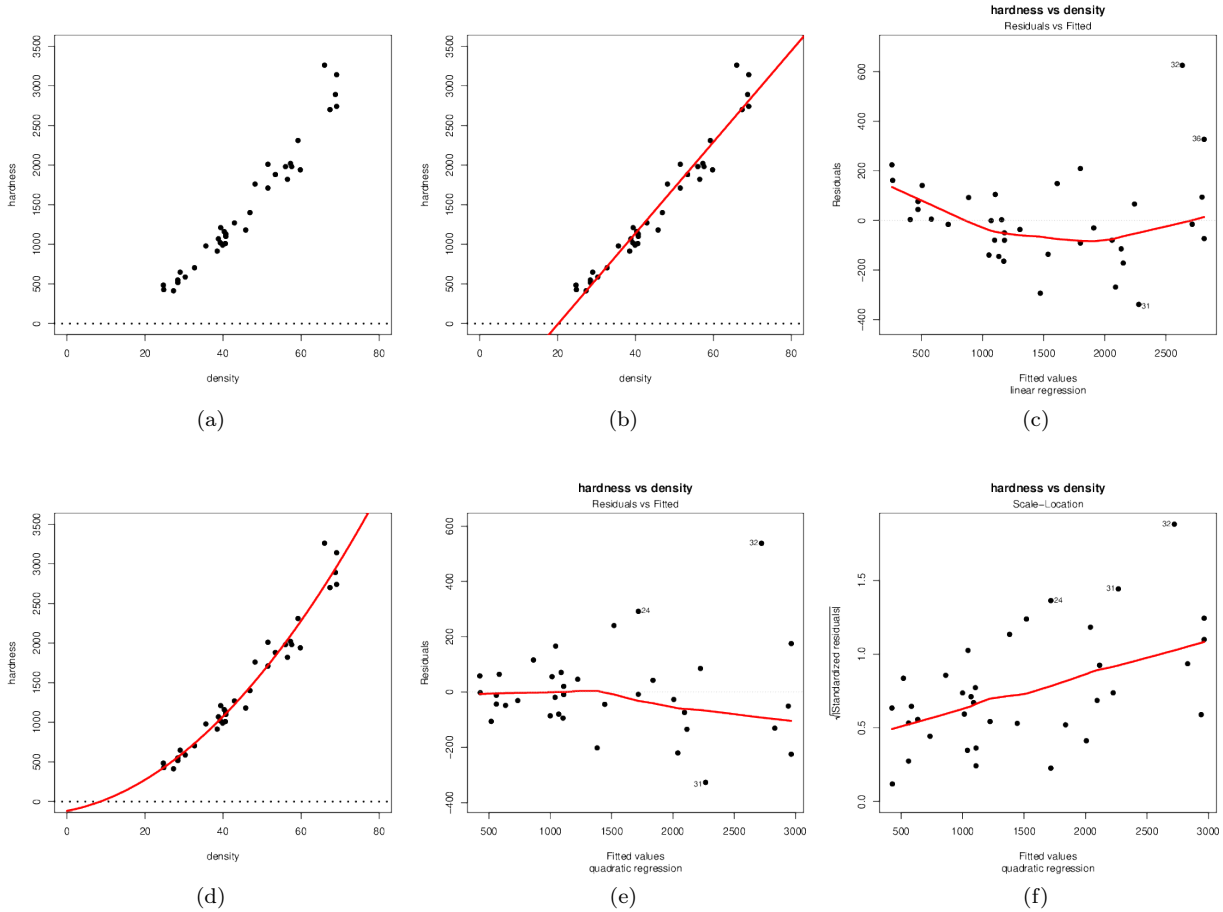


Figure 10.5: Timber data

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 63. 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 64.
- 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* $\operatorname{Var}[Y_i|x_{1i},\dots,x_{pi}] = \sigma_i^2 \forall i$*; when the dependent variables* Y_i *take* only positive values*, it is possible to prove that:*

$$\begin{aligned}
&\sigma_i^2 \propto \mathrm{E}[Y_i|x_{1i},\dots,x_{pi}] \implies \operatorname{Var}\left[\sqrt{Y_i}|x_{1i},\dots,x_{pi}\right] \approxeq \sigma^2 \text{ constant}\\
&\sigma_i^2 \propto (\mathrm{E}[Y_i|x_{1i},\dots,x_{pi}])^2 \implies \operatorname{Var}[\ln(Y_i)|x_{1i},\dots,x_{pi}] \approxeq \sigma^2\\
&\sigma_i^2 \propto (\mathrm{E}[Y_i|x_{1i},\dots,x_{pi}])^3 \implies \operatorname{Var}\left[Y_i^{-0.5}|x_{1i},\dots,x_{pi}\right] \approxeq \sigma^2\\
&\sigma_i^2 \propto (\mathrm{E}[Y_i|x_{1i},\dots,x_{pi}])^4 \implies \operatorname{Var}\left[Y_i^{-1}|x_{1i},\dots,x_{pi}\right] \approxeq \sigma^2\\
&\dots
\end{aligned}$$

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 65. 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

$$l(\boldsymbol{\beta}, \sigma^2, \lambda) = -\frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(\frac{y_i^\lambda - 1}{\lambda} - \beta_0 - \beta_1 x_{1i} - \dots - \beta_p x_{pi}\right)^2$$

Remark 66 (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)

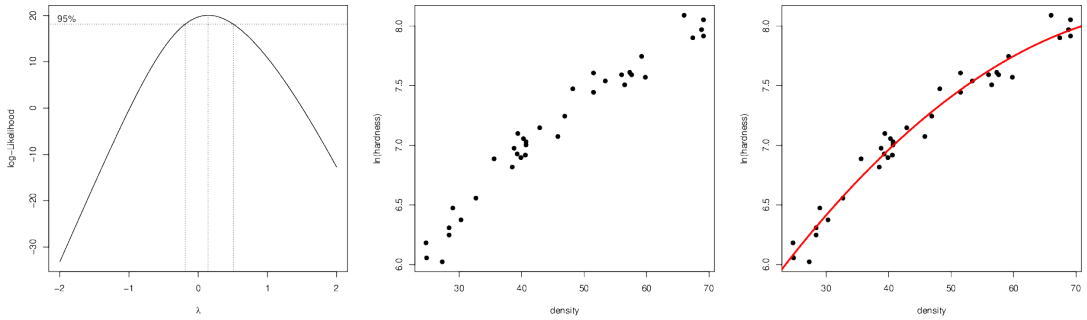
Example 10.2.3 (Timber data)**.** In figure 10.6

- (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

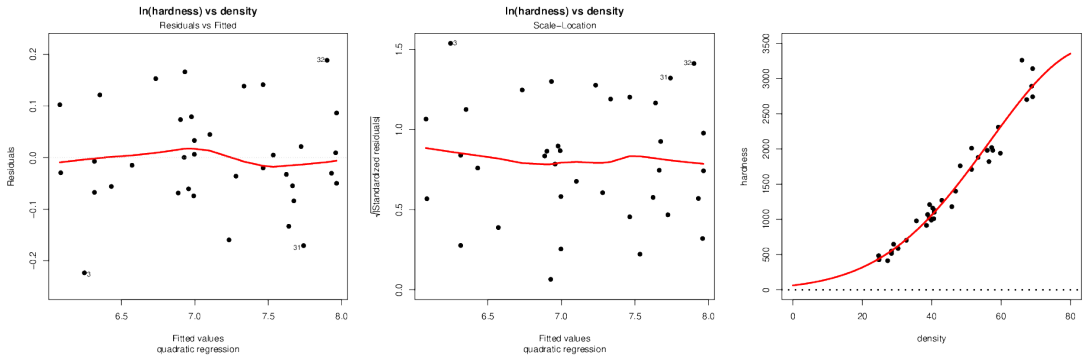
10.2.0.3 Cautionary remarks

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

Part II

Generalized linear models

Chapter 11

Introduction to GLM

11.1 Motivating examples

Example 11.1.1 (Insurance)**.** An insurance company is interested in evaluating the main risk factors that affect the number car accident claims made by its policyholders. In particular, the following factors are investigated:

- age of the policyholder (4 classes)
- district of residence of the policyholder (4 districts)
- type of car (4 classes)

The dataset is in table 11.1; regarding the dependent variable

- $\mathtt{n} \in \mathbb{R}^+$ number of policyholders characterised by a given covariate pattern/combination of values for the three factors (in this case is $\in \mathbb{N}$ but in general can be a positive real which measures the *exposure level*)
- $\mathtt{c} \in \mathbb{N}$ number of car accident claims made by policyholders characterised by a given covariate pattern (*number of events - count variable*)
- $Y = \frac{\mathtt{c}}{\mathtt{n}}$ (average) number of claims per policyholder (*rate / number of events per unit of exposure*); so in general is positive or null and in theory can be > 1 (since each person could have more than 1 accident)

The gaussian linear model on Y was estimated and residual analysis is in figure 11.1 (a) and (b); in (b) there's a problem in homoschedasticity assumption where the spread increases with fitted (a with linearity too).
We could apply a trasformation but here the outcome can take the value 0 (so no boxcox)

Example 11.1.2 (Labour force)**.** In a study on women labour force participation, the following information was collected on a random sample of swiss women:

- non-labour income (logarithmic scale)
- age class (decades)
- education level (years)
- number of young children (under 7 years of age)
- number of old children (over 7 years of age)
- nationality (two classes: foreign/non-foreign)

The data is in table 11.2, while the dependent variable is $\mathtt{participation} \in \{\mathtt{yes}, \mathtt{no}\}$ (does the individual participate in the labor force: have a job or is looking for, *dichotomous variable*)

$$Y = \mathbf{1}\{\mathtt{participation} = \mathtt{yes}\} = \begin{cases} 0 & \text{if } \mathtt{participation} = \mathtt{no} \\ 1 & \text{if } \mathtt{participation} = \mathtt{yes} \end{cases}$$

The gaussian linear model residual analysis is in fig 11.1 (c) and (d): clearly residual

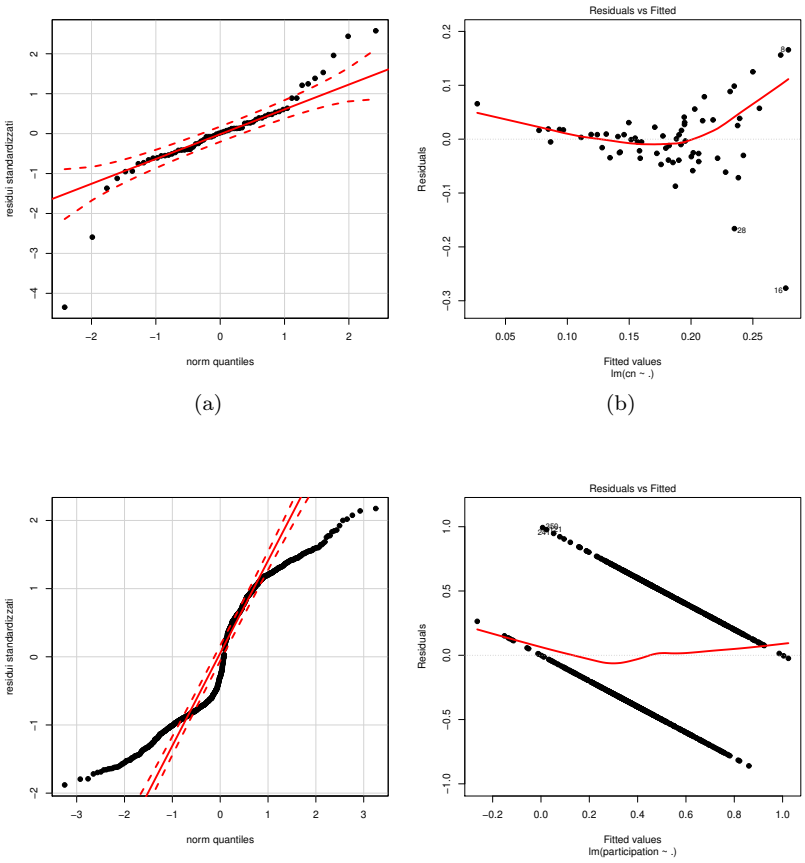


Figure 11.1: Gaussian regression residuals for insurance (a,b) and laborforce (c,d)

	n	c	age	dist	car
1	197	38	<25	rural	<1
2	284	63	<25	rural	1-1.5
3	133	19	<25	rural	1.5-2
4	24	4	<25	rural	>2
5	85	22	<25	small towns	<1
6	149	25	<25	small towns	1-1.5
7	66	14	<25	small towns	1.5-2
8	9	4	<25	small towns	>2
9	35	5	<25	large towns	<1
⋮	⋮	⋮	⋮	⋮	⋮

Table 11.1: Insurance data

- are not normal (c)
- the residuals in general should have a random pattern but in (d) they show a particular patter (they all lie on two straight lines which are between 1 and -1)
- furthermore for some unit the model produce fitted/estimated expected values (look x axis) that are negative or > 1

	participation	income	age	education	youngkids	oldkids	foreign
1	no	10.79	3.00	8.00	1.00	1.00	no
2	yes	10.52	4.50	8.00	0.00	1.00	no
3	no	10.97	4.60	9.00	0.00	0.00	no
4	no	11.10	3.10	11.00	2.00	0.00	no
5	no	11.11	4.40	12.00	0.00	2.00	no
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 11.2: Labour dataset

Important remark 80 (Limitations of Gaussian linear models). These two examples show some of the "intrinsic" limitations of Gaussian linear models. These limitations are due to:

- specific features of the dependent variable (support of the random variable/*distributional form* of Y): assuming a gaussian distribution for the conditional random variable given that this variable assumes 0/1 is forced/doesn't make sense. It's not needed a variable assumes values between $-\infty$ and ∞ but at least can assume several numeric values/wide subset of real line
  Anogher problem is *coherence of the parameter space* if estimated conditional expected value falls outside the parameter space: if dependent variables takes value 0 or 1, its expected value must be bounded between 0 and 1 (differently from the linear probability model)
- some other limitations arisen from the examples are violations of linearity or homoschedasticity: sometimes (but not always) these limitations can be overcome by suitably transforming the dependent variable and/or the regressors. In many practical situation we don't have the guarantee that a transformation that fix both problems exists.

Remark 67. Rather than sticking with the gaussian linear model, we can try a wider class of regression models.

Important remark 81 (Generalised linear models (GLM) as a solution). Class of regression models (including Gaussian linear models as special cases) that allows to overcome some of the previously introduced limitations, that often arise in many practical situations.

This class allows a *unified approach* to describe many specific regression models (eg binary/count) that where independently developed.
This unified approach provide a general common inferential framework (estimation/hypothesis testing) that is "almost" equal to the one associated with Gaussian linear models.
From historical pov these class of models was identified after a number of specific models were proposed in the literature: some researcher noted that many different models used in different context were sharing some common features so enframed all these model in a general framework that permit us to study the subject in a top down perspective

Remark 68. In order to use the top down approach to these models we need to make a probability calculus detour.

11.2 Exponential families

Remark 69. This class of probability density function are the core of GLM

Remark 70. In the probability literature there's no agreement on the notation to describe this family of distribution (so different books have different notation.

11.2.1 Families of order 1

Important remark 82. We focus on exponential families of distributions of order 1

Definition 11.2.1 (Exponential families of distributions of order 1)**.** Let

- Y be a random variable with probability mass/density function $f(y;\theta)$;
- $\theta \in \Theta \subseteq \mathbb{R}$ is an (unknown) parameter characterizing the density f of Y which belongs to a parameter space Θ; the order 1 refers to the fact that θ is one-dimensional/scalar

The distribution of Y belongs to an exponential family (of order 1) with natural parameter $b(\theta)$, written $Y \sim \mathrm{EF}(b(\theta))$, if and only if we can write down its probability/density as:

$$f(y;\theta) = \exp\left[a(y)\cdot b(\theta) + c(\theta) + d(y)\right]$$

where:

- $a(\cdot)$ e $d(\cdot)$ are *known* functions depending only on y;
- $b(\cdot)$ e $c(\cdot)$ are *known* functions depending only on θ

Definition 11.2.2 (Canonical form)**.** The exponential family is said to be expressed in **canonical form** if $a(\cdot)$ is the identity function and therefore

$$f(y;\theta) = \exp\left[y\cdot b(\theta) + c(\theta) + d(y)\right]$$

Remark 71. the function $b(\theta)$ is called the natural parameter

Remark 72. From now on we focus on distribution expressed in canonical form; let's see some examples.

Example 11.2.1 (Poisson random variables)**.** If $Y \sim \mathrm{Poi}(\theta)$, with $y \in \mathbb{N}$ $\theta \in \mathbb{R}^+$ it's pmf can be written as

$$f(y,\theta) = \frac{\theta^y \exp(-\theta)}{y!} = \exp\left[y\ln\theta - \theta - \ln y!\right]$$

where

- $a(y) = y$, $d(y) = -\ln y!$
- $b(\theta) = \ln\theta$, $c(\theta) = -\theta$

Example 11.2.2 (Bernoulli random variables)**.** If $Y \sim \mathrm{Ber}(\theta)$ with $y \in \{0,1\}$, and $\theta \in [0,1]$ its pmf can be written as

$$\begin{aligned} f(y,\theta) = \theta^y(1-\theta)^{1-y} &= \exp\left\{y\ln\theta + \ln(1-\theta) - y\ln(1-\theta)\right\} \\ &= \exp\left\{y\ln\frac{\theta}{1-\theta} + \ln(1-\theta)\right\} \end{aligned}$$

where

- $a(y) = y$, $d(y) = 0$
- $b(\theta) = \ln\dfrac{\theta}{1-\theta}$, $c(\theta) = \ln(1-\theta)$

Remark 73. A slighlty more general definition follows

11.2.2 Nuisance parameters and weights

Remark 74. The first definition we gave can be extended to encompass other information needed to deal with other less simple distribution.
These information are nuisance pameter ϕ and weights w

Definition 11.2.3 (Exponential families of distributions of order 1)**.** Let

- Y be a random variable with probability mass/density function $f(y;\theta,\phi,w)$ where ...
- $\theta \in \Theta \subseteq \mathbb{R}$ is an (unknown) parameter characterizing the density f of Y which belongs to a parameter space Θ (the order 1 refers to the fact that θ is one-dimensional/scalar);
- $\phi \in \Phi \subseteq \mathbb{R}^+$ is a striclty positive called *nuisance parameter* (usually *unknown*);
- $w \in \mathbb{R}^+$ is a strictly positive *known quantity* called *weight*

The distribution of Y belongs to an exponential family (of order 1) expressed in *canonical form* with natural parameter $b(\theta)$, nuisance parameter ϕ and weight w, written $Y \sim \mathrm{EF}(b(\theta),\phi,w)$, if and only if we can write down its probability/density as:

$$f(y;\theta,\phi,w) = \exp\left\{\frac{w}{\phi}\left[yb(\theta)+c(\theta)\right]+d(y,\phi,w)\right\} \tag{11.1}$$

where

- $b(\cdot)$ and $c(\cdot)$ are *known* functions that do not depend on y and w, only on θ;
- $d(\cdot)$ is a *known* function that *does not* depend on θ (may depend on the other introduced parameters to make it more general)

Remark 75. this is a more general definition for the exponential family. Four possible situations: we may have situations where we don't have a nuisance parameter (eg $\phi = 1$) but we have a weight; or other cases where the weight is not present ($w = 1$); or situation where we have both of them (or nothing both set to 1).

Example 11.2.3 (Gaussian random variables)**.** Here we have nuisance parameter but not the weight. Here the θ is the expected value while the nuisance parameter is the variance. So if $Y \sim \mathrm{N}(\theta,\phi)$, with $y \in \mathbb{R}$, $\theta \in \mathbb{R}$, $\phi \in \mathbb{R}^+$ its density can be written as

$$\begin{aligned} f(y;\theta,\phi) &= \frac{1}{\sqrt{2\pi\phi}}\exp\left\{-\frac{(y-\theta)^2}{2\phi}\right\} \\ &= \exp\left\{\frac{1}{\phi}\left[y\theta-\frac{\theta^2}{2}\right]+\left[-\frac{y^2}{2\phi}-\frac{\ln 2\pi\phi}{2}\right]\right\} \end{aligned}$$

where

- $w = 1$
- $b(\theta) = \theta$, $c(\theta) = -\dfrac{\theta^2}{2}$
- $d(y,\phi,w) = -\dfrac{y^2}{2\phi} - \dfrac{\ln 2\pi\phi}{2}$

Example 11.2.4 (Number of events per unit of exposure)**.** In this case we have a weight. Suppose $y^* \in \mathbb{N}$ (it's a count of events) and $Y^* \sim \mathrm{Poi}(w\theta)$, whose parameter (the expected value) is composed by the product of $\theta \in \mathbb{R}^+$ (*unknown*, a sort of mean/rate of event per single exposure) and $w \in \mathbb{R}^+$ (*known*, which is the amount of exposure, eg time or number of person etc).
Starting from it we can define the number of events per unit of exposure as

$$Y = \frac{Y^*}{w}$$

If $Y^* \sim \mathrm{Poi}(w\theta)$ we can derive that the distribution of Y: it's a ratio (between a count and an exposure) so it will not take (only) integer values, so it can't be a Poisson. However its distribution is somehow generated using the Poisson for Y^*: it's a trasformation where all the values of y^* are divided for w (so $y = y^*/w$):

$$\begin{aligned} f(y;\theta,w) &= \frac{(w\theta)^{wy}\exp(-w\theta)}{(wy)!} \\ &= \exp\left\{w\left[y\ln\theta-\theta\right]+wy\ln w-\ln(wy)!\right\} \end{aligned}$$

If $Y^* \sim \mathrm{Pois}\,(w\theta)$, Y is not a poisson but still is part of the exponential family since:

- $\phi = 1$; here differently from the poisson we have the weight w
- $b(\theta) = \ln\theta$, $c(\theta) = -\theta$: they are not changed from poisson
- $d(y, \phi, w) = wy \ln w - \ln(wy)!$ (this is different since we have the weight)

Example 11.2.5 (Relative frequency of successes in Bernoulli trials)**.** Similarly let $y^* \in \{0, 1, \dots, w\}$ be the count of number of success; let $Y^* \sim \text{Bin}(w, \theta)$, with $\theta \in [0, 1]$ and $w \in \mathbb{N}^+$. We can define the relative frequency of successes in w Bernoulli trials as

$$Y = \frac{Y^*}{w}$$

Again the distribution of Y won't be binomial anymore but is derived from that and specifically

$$\begin{aligned} f(y; \theta, w) &= \binom{w}{wy} \theta^{wy} (1-\theta)^{w(1-y)} \\ &= \exp\left\{ w \left[y \ln \frac{\theta}{1-\theta} + \ln(1-\theta) \right] + \ln \binom{w}{wy} \right\} \end{aligned}$$

and the derived random variable is in the exponential family since

- $\phi = 1$
- $b(\theta) = \ln \dfrac{\theta}{1-\theta}$, $c(\theta) = \ln(1-\theta)$: again natural parameter and c are the same
- $d(y, \phi, w) = \ln \binom{w}{wy}$: in this case d is not 0 anymore but depends on w and y

The core is similar to the binomial btw

Important remark 83 (The thing with exponential family). Regardless of the specific formulation functional forms (b, c, d), presence or absence of weight and nuisance, the exponential family shares a lot of interesting properties

11.2.3 Some relevant properties

Regardless the EF distribution we're dealing with, if $Y \sim \text{EF}(b(\theta), \phi, w)$ its possible to prove that

- its expected value is obtained by the ratio between partial derivatives of b and c and is:

  $$\text{E}[Y] = -\frac{c'(\theta)}{b'(\theta)}$$

  where

  $$b'(\theta) = \frac{\partial}{\partial\theta} b(\theta), \quad c'(\theta) = \frac{\partial}{\partial\theta} c(\theta)$$

- the variance, otoh, is always the following

  $$\text{Var}[Y] = \frac{\phi}{w} \frac{\text{E}'[Y]}{b'(\theta)} = \frac{\phi}{w} \frac{c'(\theta) b''(\theta) - c''(\theta) b'(\theta)}{[b'(\theta)]^3}$$

  where

  $$\text{E}'[Y] = \frac{\partial}{\partial\theta} \text{E}[Y] \quad b''(\theta) = \frac{\partial^2}{\partial\theta^2} b(\theta) \quad c''(\theta) = \frac{\partial^2}{\partial\theta^2} c(\theta)$$

Remark 76. This equations simplifies from the practical pov, often these calculation are easier than a lot of integrals/expectation.
Let's see how to exploit with some examples to check what is already known

Example 11.2.6 (Poisson random variables)**.** Letting

$$b'(\theta) = \frac{\partial}{\partial\theta} \ln\theta = \frac{1}{\theta}, \quad c'(\theta) = \frac{\partial}{\partial\theta}(-\theta) = -1$$

we have as previsto

$$\text{E}[Y] = -\frac{-1}{\frac{1}{\theta}} = \theta$$

Then letting

$$b''(\theta) = \frac{\partial}{\partial\theta}\frac{1}{\theta} = -\frac{1}{\theta^2}, \quad c''(\theta) = \frac{\partial}{\partial\theta}(-1) = 0, \quad \phi = 1, \quad w = 1$$

we have

$$\mathrm{Var}\left[Y\right] = \frac{1}{1}\frac{-1\cdot -\frac{1}{\theta^2} - \frac{1}{\theta}\cdot 0}{\left[\frac{1}{\theta}\right]^3} = \frac{\theta^3}{\theta^2} = \theta$$

Remark 77. If we model the conditional distribution with a poisson we have that the variance will coincide with the expected value (heteroskedasticity)

Example 11.2.7 (Number of events per unit of exposure)**.** Here the transformed variable is no longer poisson but we still can get easily expected value and variance. We have

$$\mathrm{E}\left[Y\right] = -\frac{-1}{\frac{1}{\theta}} = \theta$$

and

$$\mathrm{Var}\left[Y\right] = \frac{1}{w}\frac{-1\cdot -\frac{1}{\theta^2} - \frac{1}{\theta}\cdot 0}{\left[\frac{1}{\theta}\right]^3} = \frac{\theta^3}{w\theta^2} = \frac{\theta}{w}$$

Example 11.2.8 (Gaussian random variables)**.** Finally if

$$b'(\theta) = \frac{\partial}{\partial\theta}\theta = 1, \quad c'(\theta) = \frac{\partial}{\partial\theta}\left(-\frac{\theta^2}{2}\right) = -\theta$$

then

$$\mathrm{E}\left[Y\right] - \frac{-\theta}{1} = \theta$$

And considering

$$b''(\theta) = \frac{\partial}{\partial\theta}1 = 0 \quad c''(\theta) = \frac{\partial}{\partial\theta}(-\theta) = -1, \quad w = 1$$

then

$$\mathrm{Var}\left[Y\right] = \frac{\phi}{1}\frac{(-\theta)\cdot 0 - 1\cdot(-1)}{[1]^3} = \phi$$

11.3 Generalised linear models

Remark 78. Exponential family have interesting feature to be exploited in the context of regression analyis, since we can think of a general class of model where the thses distribution are used to define the so called probabistic component associated with the model.
We can especially enlarge the class of conditional distribution of Y given X

11.3.1 Definitions

Definition 11.3.1 (Generalised linear model)**.** Let

- Y_i r.v. that describes the possible value of the dependent variable on the i-th sample unit ($i = 1, \ldots, n$)
- $\mathbf{x}_i = (x_{0i}, x_{1i}, \ldots, x_{pi})^\top$ be $p+1$-dimensional vector containing the values of the regressors for the i-th sample unit ($x_{0i} = 1\ \forall i$ constant regressor associated with the model intercept)

A generalised linear model (GLM) is a statistical model for the random sample $\mathbf{Y}$ characterised by:

- a **probabilistic component**: the assumption regarding conditional probability mass/density funcion for Y_i, given the regressors

- a **systematic/deterministic component**: the functional relationship between the regressors and the (conditional) expected values of Y_i

Definition 11.3.2 (Probabilistic component of a GLM)**.** Tipically any glm model is a model where the conditional distribution of Y given x is assumed to belong to a given exponential family. Specifically we assume that

- all the conditional distributions belong to the same exponential family expressed in canonical form with natural parameter $b(\theta)$

  $$Y_i|\mathbf{x}_i \sim \text{EF}(b(\theta_i), \phi, w_i)$$

  where

  - ; the functions $b(\cdot)$, $c(\cdot)$ and $d(\cdots)$ are the same for all Y_i (they do not depend on i)
  - each conditional distribution (for $i = 1, \ldots, n$) can take a *different value* for the unknown parameter θ_i and different (known) weight w_i
  - all the conditional distributions share the *same value* for the nuisance parameter ϕ, that can be either known or unknown

- *(conditional) independence* among the n r. v. $Y_1|\mathbf{x}_1, \ldots, Y_n|\mathbf{x}_n$

Definition 11.3.3 (Systematic/deterministic component of a GLM)**.** It's defined by introducing:

- the usual set of *regression coefficient* $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\top \in \mathbb{R}^{p+1}$ vector of unknown parameters (intercept included)
- the so called *linear predictor* obtained by combining linearly regressors and regression coefficient

  $$\eta_i = \beta_0 x_{0i} + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} = \mathbf{x}_i^\top \boldsymbol{\beta}$$

- a *link function* g vwhich is used to link the linear predictor to the expected value of Y given x and especially

  $$g(\mathbb{E}[Y_i|\mathbf{x}_i]) = \eta_i$$

  In the glm setting, rather than setting the linear predictor directly equal to the conditional expected value (as done for gaussian) we're using the link function which connect the twos.
  The link function must:

  - have known functional form (does not depend on any unknown quantity)
  - be differentiable
  - be invertible, with $g^{-1}(\cdot) = h(\cdot)$ so we can express the conditional expected value as $\mathbb{E}[Y_i|\mathbf{x}_i] = h(\eta_i)$

Remark 79. Looking at $\mathbb{E}[Y_i|\mathbf{x}_i] = h(\eta_i)$ is evident that whenever we choose a link function that is nonlinear we will end up with a conditional expected value Y_i which is nonlinear both in the parameters and in the regressors

Remark 80. Glm generalize gaussian model in two direction: the probabilistic component have not to be gaussian, while in the systematic component we're not necessarily linear in the relation between betas parameter, regressors and Y (due to the introduction of a generic link function).

Example 11.3.1 (Gaussian linear models (retold))**.** We have $\mathbf{Y}|\mathbf{X} \sim MVN_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n)$ is obtained using

- as probabilistic component
  - conditional gaussian distribution $Y_i|\mathbf{x}_i \sim N(\theta_i, \phi)$, with $i = 1, \ldots, n$
  - (conditional) independence among the n r. v. $Y_1|\mathbf{x}_1, \ldots, Y_n|\mathbf{x}_n$
- as systematic component
  - we have as usual $\eta_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}$, $i = 1, \ldots, n$

– as link function g we take the identity function $\theta_i = \mathbb{E}[Y_i|\mathbf{x}_i] = \eta_i$

Remark 81. Now we mention quickly the model we'll focus on in the following lectures

Example 11.3.2 (Poisson regression models)**.** We have:

- Probabilistic component
  - the conditional distribution is poisson $Y_i|\mathbf{x}_i \sim \text{Poi}(\theta_i)$, $i = 1, \dots, n$
  - we assume (conditional) independence among the n r. v. $Y_1|\mathbf{x}_1, \dots, Y_n|\mathbf{x}_n$
- Systematic component: we use the logarithm as link function

$$\ln(\mathbb{E}[Y_i|\mathbf{x}_i]) = \eta_i \implies \mathbb{E}[Y_i|\mathbf{x}_i] = \exp(\eta_i)$$

Poisson regression are useful for count variable; by construction the expected value of a count is strictly positive, the choiche the log/exp link guarantees that whatever the value of the linear predictor η_i is, the corresponding expected value will be striclty positive

Example 11.3.3 (A class of regression models for rates)**.** Connected with poisson model in this case we have the model for number of events per unit of exposure where:

- Probabilistic component
  - assuming the count of events is poisson distributed $Y_i^*|\mathbf{x}_i \sim \text{Poi}(w_i\theta_i) \qquad i = 1, \dots, n$
  - we can obtain the random variable rate $Y_i = \dfrac{Y_i^*}{w_i}$ which is no longer poisson but is still member of exponential family
  - and assume (conditional) independence among the n r. v. $Y_1|\mathbf{x}_1, \dots, Y_n|\mathbf{x}_n$
- Systematic component
  - we use again logarithm as link function

$$\ln(\mathbb{E}[Y_i|\mathbf{x}_i]) = \eta_i$$

thus $\mathbb{E}[Y_i|\mathbf{x}_i] = \exp(\eta_i)$.

Important remark 84. One point worth discussing: we're building a regression model for the rate Y_i, which is linked to the number of events Y_i^* we implicitly we are defining a regression model for the number of events; given the simple relation between Y_i and Y_i^* we have that

$$\mathbb{E}[Y_i|\mathbf{x}_i] = \mathbb{E}\left[\frac{Y_i^*}{w_i}|\mathbf{x}_i\right]$$

thus being w_i an unknown quantity we can take it out from expectation

$$\mathbb{E}[Y_i^*|\mathbf{x}_i] = w_i\,\mathbb{E}[Y_i|\mathbf{x}_i]$$

So

$$\ln(\mathrm{E}[Y_i^*|\mathbf{x}_i]) = \ln(w_i\mathrm{E}[Y_i|\mathbf{x}_i]) = \underbrace{\ln(w_i) + \eta_i}_{\eta_i^*}$$

$$\mathrm{E}[Y_i^*|\mathbf{x}_i] = \exp(\eta_i^*) = w_i\exp(\eta_i)$$

where η_i^* linear predictor in Poisson regression models for Y_i^*, with an **offset** (compensating term - *regressor with regression coefficient set to 1*) equal to $\ln(w_i)$.
Building the model for the rates Y_i (number of events per unit of exposure) is basically building a model for the number of events (irrespective of the exposure level) but including in the linear predictors not only regressors and coefficients but an additional regressor $\ln w_i$ characterized by a regressor coefficient equal to 1. Extendedly we have

$$\mathrm{E}[Y_i^*|\mathbf{x}_i] = \exp(\eta_i^*) = \exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \ln(w_i))$$

The $\ln(w_i)$ is a sort of additional regressor which can be included in the model, but with the condition that it regressor coefficient is fixed and equal to 1.
This way to include a regressor but fixing is known in the literature as adding an offset to the regressor model: this is a sort of compensating term for the fact that each unit has a different weight.
Point is that we can use poisson model both for count variable or for rates (by including an offset)

Example 11.3.4 (Logistic regression models)**.** We have:

- Probabilistic component we use distribution of relative frequency of successes
  - $Y_i^*|\mathbf{x}_i \sim \text{Bin}(w_i, \theta_i)$, $i = 1, \dots, n$
  - $Y_i = \dfrac{Y_i^*}{w_i}$
  - (conditional) independence among the n r. v. $Y_1|\mathbf{x}_1, \dots, Y_n|\mathbf{x}_n$
- in the systematic component
  - as link we use the so called *logit* function
    $$\text{logit}(\mathbb{E}[Y_i|\mathbf{x}_i]) = \ln \frac{\mathbb{E}[Y_i|\mathbf{x}_i]}{1 - \mathbb{E}[Y_i|\mathbf{x}_i]} = \eta_i$$
    the inverse of the logit function is the so-called logistic function
    $$\mathbb{E}[Y_i|\mathbf{x}_i] = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$
    logistic function bounds between 0 and 1 anything so using the logistic regressio model we will always obtain an extimated expected value for our dependent variable that is between 0 and 1. Finally we have that the estimated number of successes is just the number of trials times the estimated probability of success
    $$\text{E}[Y_i^*|\mathbf{x}_i] = w_i \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$

Remark 82. These are the most known and used models; there are other examples of regression that can be built with this framework

11.3.2 Choice of the link function

Remark 83. One of the new entry in the framework is the link function: how to chose it among the function respecting the link function requirements?
We could define a gaussian model without a identity link function, or poisson without log link and so on; in principle we could choose the best link function for a specific dataset (eg looking/comparing at fit statistics). There is one further requirements however in the choice

Important remark 85. Each probabilistic component is characterised by a specific set of possible value for $\mathbb{E}[Y_i|\mathbf{x}_i]$
$$\mathbb{E}[Y_i|\mathbf{x}_i] \in \Omega \subseteq \mathbb{R}$$
for example a poisson regression should have strictly positive conditional expected value.
In choosing the link function g we should guarantee that *the range of its inverse* $h(\cdot) = g^{-1}(\cdot)$ coincide with the parameter space of the probabilistic component Ω
$$h(\cdot) : \mathbb{R} \longmapsto \Omega$$

Definition 11.3.4 (Canonical link functions)**.** Each exponential family has *a special* link function (known as the **canonical** link function), that is obtained by equating the natural parameter (of the considered exponential family) to the linear predictor:
$$b(\theta_i) = \eta_i$$

Example 11.3.5. For example:

- Gaussian r. vs: $\theta_i = \eta_i$ identity function
- Poisson r. vs. / number of events per unit of exposure: $\ln \theta_i = \eta_i$ logarithm
- Bernoulli r. vs./ relative frequency: $\Rightarrow \ln \dfrac{\theta_i}{1 - \theta_i} = \eta_i$ logistic function

Important remark 86 (On the use of canonical link functions). It must be noted that

- the use of canonical link functions lead to some analytical *simplifications* and to some *theoretical properties*;
- not necessarily canonical link functions are the best choice for the fit
  - they could not be adequate to describe the effect of the regressors on $\mathbb{E}[Y_i|\mathbf{x}_i]$ (eg maybe other links fit better);
  - sometimes when $b(\theta_i) \in \Psi \subset \mathbb{R}$ (its a subset of the real line) there could be some compatibility problems with the range of η_i (which ranges on the real line): the two ranges should overlap

11.3.3 GLM and Gaussian linear models (recappone)

GLMs allow to overcome some limitations of Gaussian linear models

- use of probability mass/density functions that differ from the Gaussian one
  - with supports that differ from $\mathbb{R}$
  - with (conditional) variance that is not constant (linked to the conditional expected values)
- non-linearity of the conditional expected values (wrt both the regressors and the parameters $\beta_0, \beta_1, \ldots, \beta_p$
  - use of link functions that differ from the identity one

One of the key features of generalised linear models is the possibility to match any probabilistic component with any systematic component (at least in principle)

Remark 84. In the upcoming slides we use the general framework of glm to derive the most important quantities for GLMs; these are general expressions that will need to be particularized for the model/data at hand.
We start with a general expression for the loglikelihood

11.3.4 Log-likelihood

In the GLM we're basically linking the expected value to a function of the linear predictor; in this context the unknown parameters are the betas.
We've also seen that when dealing with exponential family, the models are characterized by a parameter θ_i related to the expected value and that no matter which exponential family we choose, the expected value is related to the first derivative of b and c

$$\mathrm{E}[Y] = -\frac{c'(\theta)}{b'(\theta)}$$

So in our framework it turns out that

$$g\left(-\frac{c'(\theta_i)}{b'(\theta_i)}\right) = \eta_i \iff -\frac{c'(\theta_i)}{b'(\theta_i)} = h(\eta_i)$$

Therefore there is an implicit functional relationship betweeen θ_i and the $\beta_0, \ldots, \beta_p$.

We want to come up with an expression for the loglikelihood of the betas (aside for the ϕ), that is $l(\beta_0, \ldots, \beta_p, \phi)$. The first step to obtain it is by exploiting the conditional independence assumption, so the loglikelihood can be written as the sum of the n individual contribution to the loglik

$$l(\beta_0, \ldots, \beta_p, \phi) = \sum_{i=1}^{n} l_i(\beta_0, \ldots, \beta_p, \phi)$$

where the individual components are as follow: they are log of density/mass belonging to the exponential family evaluated at the i-th unit; we can express it in a general way using the exponential family function as in 11.1

$$\begin{aligned} l_i(\beta_0, \ldots, \beta_p, \phi) &= \ln f(y_i; \theta_i, \phi, w_i) \\ &= \frac{w_i}{\phi}\left[y_i b(\theta_i) + c(\theta_i)\right] + d(y_i, \phi, w_i) \end{aligned}$$

In this last equation we don't have explicit presence of betas, only thetas; but we know that somehow $theta_i$ are connected with betas (since both are connected to the expected value).
So the general formula will be

$$l(\beta_0, \ldots, \beta_p, \phi) = \sum_{i=1}^{n} \frac{w_i}{\phi}\left[y_i b(\theta_i) + c(\theta_i)\right] + d(y_i, \phi, w_i) \tag{11.2}$$

We work on this general loglik to obtain other stuff

11.3.5 Score function

The first quantity to be derived is the score function: its a vector with $p+1$ element of the first partial derivatives of the loglik

$$U_j(\boldsymbol{\beta}) = \frac{\partial l(\beta_0, \dots, \beta_p, \phi)}{\partial \beta_j} \overset{(1)}{=} \sum_{i=1}^{n} \frac{\partial l_i(\beta_0, \dots, \beta_p, \phi)}{\partial \beta_j}$$
$$\overset{(2)}{=} \sum_{i=1}^{n} \frac{\partial l_i(\beta_0, \dots, \beta_p, \phi)}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mathbb{E}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j}$$

where:

- in (1) we used conditional independence assumption
- in (2) rather than computing directly the first partial derivative with respect of β_j of each contribution of the likelihood l_i we apply the chain rule for the derivative of composite function (we have implicitly the relation between betas and thetas behind). Each of the four component is much easier to deal with than the whole derivative itself.

Remark 85. For the first element of the chain rule

$$\frac{\partial l_i(\beta_0, \dots, \beta_p, \phi)}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left\{ \frac{w_i}{\phi} [y_i b(\theta_i) + c(\theta_i)] + d(y_i, \phi, w_i) \right\}$$
$$\overset{(1)}{=} \frac{w_i}{\phi} \left[y_i b'(\theta_i) + c'(\theta_i) \right]$$
$$= \frac{w_i}{\phi} b'(\theta_i) \left[y_i + \frac{c'(\theta_i)}{b'(\theta_i)} \right]$$
$$= \frac{w_i}{\phi} b'(\theta_i) \{ y_i - \mathbb{E}[Y_i|\mathbf{x}_i] \}$$

where in (1) d does not depend on θ_i so we discard it in derivation.
For the second element of the chain, $\frac{\partial \theta_i}{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}$, this can be rewritten as

$$\frac{\partial \theta_i}{\partial \mathbb{E}[Y_i|\mathbf{x}_i]} = \frac{1}{\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \theta_i}}$$

Now remembering that we have

$$\mathrm{Var}[Y_i|\mathbf{x}_i] = \frac{1}{b'(\theta)} \frac{\phi}{w_i} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \theta_i}$$

we can express the second element of the chain as a function of variance of Y_i

$$\frac{\partial \theta_i}{\partial \mathbb{E}[Y_i|\mathbf{x}_i]} = \frac{1}{\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \theta_i}} = \frac{1}{\frac{w_i}{\phi} b'(\theta_i) \mathrm{Var}[Y_i|\mathbf{x}_i]}$$

The third component $\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i}$ is basically the first partial derivative of the link function and we leave it unchanged.
For the fourth component (first partial derivative of the linear predictor with respect to $beta_j$) is just the j-th regressor:

$$\frac{\partial \eta_i}{\partial \beta_j} = x_{ji}$$

Remark 86. To put all things together we have, and substituting first second and fourth element we have:

$$U_j(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\partial l_i(\beta_0, \dots, \beta_p, \phi)}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mathbb{E}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_j}$$
$$= \sum_{i=1}^{n} \frac{\frac{w_i}{\phi} b'(\theta_i) \{ y_i - \mathbb{E}[Y_i|\mathbf{x}_i] \}}{\frac{w_i}{\phi} b'(\theta_i) \mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} x_{ji}$$

and finally

$$U_j(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} x_{ji} \tag{11.3}$$

There are simplification and in the end we have a simple expression which does not depend: no matter what specific glm we're considering we can write the generic element in the score function (vector) as sum over the unit of y_i minus the conditional expected value divided by its conditional variance, times the first derivative of the link function, times the j-th covariate

Important remark 87. This is a generic formula: once we choose the probabilistic component and the link function we can substitute to obtain the real score function

Remark 87. Let's see how it works in a practical situation

Example 11.3.6 (Score function of a Gaussian linear model)**.** We've already seen how to compute the score function; let's verify it here. Having $\mathbf{Y}|\mathbf{X} \sim MVN_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n)$, recall that, for $i = 1, \ldots, n$:

- the link function is the identity $\mathbb{E}[Y_i|\mathbf{x}_i] = \eta_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}$ thus its first derivative $\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} = 1$
- the conditional variance is constant $\mathrm{Var}[Y_i|\mathbf{x}_i] = \sigma^2$

Thus by just plugging components in 11.3 we have easily:

$$U_j(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{(y_i - \eta_i)}{\sigma^2} \cdot 1 \cdot x_{ji} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi}) x_{ji}$$

11.3.6 Observed Fisher information

Similarly we can develop a general expression for observed Fisher information (the matrix containing all the second partial derivatives with respect of any pair of regression coefficient, with minus sign). It's generic element (first differentiate with respect to β_j and then β_j) will be

$$\begin{aligned}
i_{jl}(\boldsymbol{\beta}) &= -\frac{\partial^2}{\partial \beta_j \beta_l} l(\beta_0, \ldots, \beta_p, \phi) \overset{(1)}{=} -\frac{\partial}{\partial \beta_l} U_j(\boldsymbol{\beta}) \\
&\overset{(2)}{=} -\frac{\partial}{\partial \beta_l} \sum_{i=1}^{n} \frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} \\
&= -\sum_{i=1}^{n} \left[\frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \right] \frac{\partial}{\partial \beta_l} \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} - \sum_{i=1}^{n} \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} \frac{\partial}{\partial \beta_l} \left(\frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \right)
\end{aligned}$$

where:

- in (1) it's the first partial derivative of the first partial derivative (which is the score function/vector element);
- in (2) we substituted the expression j-th element of the score function found in 11.3 rearranged just a bit; since it's a sum its differenciation is just the sum of differentiated components. Each element of the sum can be viewed as the product of two quantities: the first part being $y_i - \mathbb{E}[Y_i|\mathbf{x}_i]$ and the second the other part $\frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i}$. So we have to compute the sum of the derivatives of the product of these two functions (considering that β_l appears in both of them
- in (3) si sviluppa applicando la regola del prodotto (i termini sono un po' organizzati alla katzoo per far capire su cosa si sta derivando ma pace)

We can simplify it a bit considering the derivation ofthe expected value, which is a compound function depending from the linear predictor and it on the betas (so we apply the chain rule):

$$\frac{\partial}{\partial \beta_l} \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} = -\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta_l} = -\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} x_{li}$$

with $\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i}$ being the first partial derivative of the link function (with respect to η_i).
So we end up with

$$i_{jl}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{x_{ji} x_{li}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \left(\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \right)^2 - \sum_{i=1}^{n} \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} \frac{\partial}{\partial \beta_l} \left(\frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \right) \tag{11.4}$$

which will be particularized for data/hypotheses at hand.

Example 11.3.7 (Observed Fisher information of a Gaussian linear model)**.** Similarly to what seen regarding the score function we obtain the same results as seen in the first lectures.

$$i_{jl}(\boldsymbol{\beta}) \overset{(1)}{=} \sum_{i=1}^{n} \frac{x_{ji}x_{li}}{\sigma^2}(1)^2 - \sum_{i=1}^{n} \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} \underbrace{\frac{\partial}{\partial\beta_l}\left(\frac{x_{ji}}{\sigma^2}\cdot 1\right)}_{=0,\forall j} = \frac{1}{\sigma^2}\sum_{i=1}^{n} x_{ji}x_{li}$$

where in (1), we have that $\frac{\partial\,\mathbb{E}[Y_i|\mathbf{x}_i]}{\partial\eta_i}$ is the first partial derivative of the identity (link) function which is 1

11.3.7 Expected Fisher information

It's again the expected value of the fisher information matrix; its values are computed conditionally on the regressor values, so we assume/treat they are constant. The generic element will be

$$\begin{aligned}
I_{jl}(\boldsymbol{\beta}) &= \mathrm{E}\left[i_{jl}(\boldsymbol{\beta})\right] \\
&= \sum_{i=1}^{n} \mathrm{E}\left[\frac{x_{ji}x_{li}}{\mathrm{Var}[Y_i|\mathbf{x}_i]}\left(\frac{\partial\,\mathbb{E}[Y_i|\mathbf{x}_i]}{\partial\eta_i}\right)^2\right] - \sum_{i=1}^{n} \mathrm{E}\left[\{Y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\}\frac{\partial}{\partial\beta_l}\left(\frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]}\frac{\partial\,\mathbb{E}[Y_i|\mathbf{x}_i]}{\partial\eta_i}\right)\right] \\
&= \sum_{i=1}^{n} \frac{x_{ji}x_{li}}{\mathrm{Var}[Y_i|\mathbf{x}_i]}\left(\frac{\partial\,\mathbb{E}[Y_i|\mathbf{x}_i]}{\partial\eta_i}\right)^2 - \sum_{i=1}^{n} \frac{\partial}{\partial\beta_l}\left(\frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]}\frac{\partial\,\mathbb{E}[Y_i|\mathbf{x}_i]}{\partial\eta_i}\right)\underbrace{\mathrm{E}\left[Y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\right]}_{0} \\
&= \sum_{i=1}^{n} \frac{x_{ji}x_{li}}{\mathrm{Var}[Y_i|\mathbf{x}_i]}\left(\frac{\partial\,\mathbb{E}[Y_i|\mathbf{x}_i]}{\partial\eta_i}\right)^2
\end{aligned}$$

In the end we come up with a simple formula depending on the sum of crossproduct of regressors, the conditional variance and the first derivative of the link function

Example 11.3.8. If we apply it to the gaussian we get same result as before; in particular, we have an observed fisher information matrix coinciding with expected one.

Remark 88. Having expected and observed fisher information matrix coinciding happens on very specific condition: we'll see that

- we need to be in the context of glm
- the link function must coincides with the canonical one

11.4 Canonical link and . . .

Remark 89. Let's see what happen to the loglik if we consider the canonical link function

11.4.1 . . . log-likelihood

To consider the canonical link function we replace $b(\theta_i)$ with η_i (as from definition 11.3.4) in the expression derived above; starting from the loglik of equation 11.2 we have:

$$\begin{aligned}
l(\beta_0,\ldots,\beta_p,\phi) &= \sum_{i=1}^{n} \frac{w_i}{\phi}\left[y_i\eta_i + c(\theta_i)\right] + d(y_i,\phi,w_i) \\
&= \sum_{i=1}^{n} \frac{w_i}{\phi}y_i\eta_i + \sum_{i=1}^{n} \frac{w_i}{\phi}c(\theta_i) + \sum_{i=1}^{n} d(y_i,\phi,w_i) \\
&\overset{(1)}{=} \frac{1}{\phi}\left[\beta_0\sum_{i=1}^{n} w_i y_i x_{0i} + \ldots + \beta_p\sum_{i=1}^{n} w_i y_i x_{pi}\right] + \sum_{i=1}^{n} \frac{w_i}{\phi}c(\theta_i) + \sum_{i=1}^{n} d(y_i,\phi,w_i)
\end{aligned}$$

where in (1) we split the first component by taking advantage of the fact that η_i is a linear combination of betas and regtressors.
We end up with a definition of loglik where each beta is multiplied by a quantity such as $\sum_{i=1}^{n} w_i y_i x_{0i},\ldots,\sum_{i=1}^{n} w_i y_i x_{pi}$. Each of these quantity is a sum of the product of the weight, the y and one regressor (its a weighted crossproduct of the dependent variable and a

regressor).
Point is that the summary quantities $\sum_{i=1}^n w_i y_i x_{0i}, \dots, \sum_{i=1}^n w_i y_i x_{pi}$ are *(minimal) sufficient statistics for* $\boldsymbol{\beta}$: in terms of loglik, the information that really matters (in order to make inference for the regressor coefficient) are not the single/individual y_i and $x_{1i}, \dots, x_{pi}$, but it's sufficient to know these summary statistics/the actual information is in them.
In order to write down loglik of our model, we can compress the info in the dataframe into fewer $p+1$ summary quantities

Important remark 88 (Sufficient statistics). The idea of sufficient statistics is that if we have two samples characterized by the same value for this set of sufficient statistics the inference/inferential results on the unknown parameters obtained by the two samples must coincide (the two samples are carrying the same amount/type of information)

Remark 90. Eg from a cs point of view we could discard the dataframe and keep only these numbers

11.4.2 ... score function

First a bit of black magic (we exploit properties of exponential family) to express the first partial derivative of the link function as an expression of the conditional variance

$$\begin{aligned}\frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} &= \frac{\partial \mathbb{E}[Y_i|\mathbf{x}_i]}{\partial \theta_i}\frac{\partial \theta_i}{\partial \eta_i} \\ &= \frac{w_i}{\phi} b'(\theta_i) \operatorname{Var}[Y_i|\mathbf{x}_i] \frac{1}{\frac{\partial \eta_i}{\partial \theta_i}} \\ &= \frac{w_i}{\phi} b'(\theta_i) \operatorname{Var}[Y_i|\mathbf{x}_i] \frac{1}{\frac{\partial b(\theta_i)}{\partial \theta_i}} \\ &= \frac{w_i}{\phi} \operatorname{Var}[Y_i|\mathbf{x}_i]\end{aligned}$$

Then we can substitute this in the standard 11.3

$$\begin{aligned}U_j(\boldsymbol{\beta}) &= \sum_{i=1}^n \frac{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]}{\operatorname{Var}[Y_i|\mathbf{x}_i]} \frac{w_i}{\phi} \operatorname{Var}[Y_i|\mathbf{x}_i] x_{ji} \\ &= \frac{1}{\phi} \sum_{i=1}^n w_i \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} x_{ji} \end{aligned} \tag{11.5}$$

We end with an even simpler formula

11.4.3 ... Fisher information

In a similar way if we start from the generic element of the observed fisher information

$$i_{jl}(\boldsymbol{\beta}) = \sum_{i=1}^n \frac{x_{ji}x_{li}}{\operatorname{Var}[Y_i|\mathbf{x}_i]} \left(\frac{w_i}{\phi} \operatorname{Var}[Y_i|\mathbf{x}_i]\right)^2 - \sum_{i=1}^n \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} \underbrace{\frac{\partial}{\partial \beta_l}\left(\frac{x_{ji}}{\operatorname{Var}[Y_i|\mathbf{x}_i]} \frac{w_i}{\phi} \operatorname{Var}[Y_i|\mathbf{x}_i]\right)}_{0}$$

$$= \sum_{i=1}^n \left(\frac{w_i}{\phi}\right)^2 \operatorname{Var}[Y_i|\mathbf{x}_i] x_{ji}x_{li} \tag{11.6}$$

$$= I_{jl}(\boldsymbol{\beta}) \tag{11.7}$$

We see that if we use the canonical link, we get a simplified expression (the second factor does not depend on β_l so it cancels out) which does not involve y_i; therefore going for the expected value of this we'll have that the expected information will coincides with the observed one.

Important remark 89. For any regression model not being glm with canonical link will be characterized by an expected information different from the observed

11.4.4 Example using Gaussian linear models

By applying the simplified expression involving the canonical link we obtain again

- $\sum_{i=1}^n y_i, \sum_{i=1}^n y_i x_{1i}, \dots, \sum_{i=1}^n y_i x_{pi}$ are (minimal) sufficient statistics $\boldsymbol{\beta}$

- $U_j(\boldsymbol{\beta}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi})x_{ji}$
- $i_{jl}(\boldsymbol{\beta}) = I_{jl}(\boldsymbol{\beta}) = \sum_{i=1}^{n}\left(\frac{1}{\sigma^2}\right)^2 \sigma^2 x_{ji}x_{li} = \frac{1}{\sigma^2}\sum_{i=1}^{n} x_{ji}x_{li}$

11.5 Inference for a GLM

Remark 91. We briefly conclude the general GLM discussion with some inferential consideration. The fact that loglik of any glm has a common structure allows us to view inference in a unified way.

We can prove that for any glm:

- the score function is asintotically mvn gaussian

$$U(\boldsymbol{\beta}) \xrightarrow{d} MVN_{p+1}(\mathbf{0}_{p+1}, I(\boldsymbol{\beta}))$$

- by introducing link function (which can be non linear) we have some complication in maximum likelihood estimation for $\boldsymbol{\beta}$ (no closed formula) and so we must rely on numerical optimization techniques (Newton-Raphson and/or Fisher scoring algorithms)
- maximum likelihood estimators are asintotically normal

$$\hat{\boldsymbol{B}} \xrightarrow{d} MVN_{p+1}\left(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}\right)$$

- goodness of fit/adequacy of the model will be based on graphical analysis of residuals while hypothesis testing based on the residual deviance (if ϕ is known)
- Linear hypothesis on $\boldsymbol{\beta}$: $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{t}$ can be performed and specifically we have that
  - likelihood ratio test statistic: $2\ln\frac{L(\hat{\mathbf{b}},\hat{\phi})}{L(\hat{\mathbf{b}}_{H_0},\hat{\phi}_{H_0})}\Big| H_0 \xrightarrow{d} \chi^2_q$
  - Wald test statistic: $\left[\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right]^\top \left[\mathbf{K}\widehat{I(\boldsymbol{\beta})}^{-1}\mathbf{K}^\top\right]^{-1}\left[\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right]\Big| H_0 \xrightarrow{d} \chi^2_q$

Chapter 12

Poisson regression models

12.1 Model definition and maximum likelihood estimation

Example 12.1.1 (Counts as dependent variables)**.** We could have:

- number of car accident claims made by policyholders to an insurance company
- number of points scored by a basketball player during a regular season
- number of imperfections on a glass plate
- number of patients with a given disease hospitalised in a given town

12.1.1 Model definition

Let

- Y_i be the r.v. describing the observed value for the dependent variable on the i-th sample unit ($i = 1, \ldots, n$)
- $\mathbf{x}_i = (x_{0i}, x_{1i} \ldots, x_{pi})^\top$ be the $p+1$-dimensional vector containing the regressor values observed on the i-th sample unit ($x_{0i} = 1\ \forall i$ constant regressor associated with the model intercept)
- $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^\top$ be the $p+1$-dimensional vector containing the model parameters (including the intercept)
- $\eta_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} = \mathbf{x}_i^\top \boldsymbol{\beta}$ be the linear predictor associated with the i-th statistical unit

A poisson regression model assumes:

- that the independent variable is conditionally distributed according to a Poisson distribution, written equivalently as
$$Y_i|\mathbf{x}_i \sim \text{Poi}(\exp(\eta_i))$$
or
$$f(y_i|\mathbf{x}_i) = \frac{\exp(\eta_i)^{y_i} \exp(-\exp(\eta_i))}{y_i!}$$
- the observations in the sample are conditionally independent (this to get the joint distribution/likelihood of the sample): $Y_i|\mathbf{x}_i$ independent ($i = 1, \ldots, n$)

Such a model belong to the GLM family using exponential distribution:

- Probabilistic component:
$$f(y_i|\mathbf{x}_i) = \frac{\mu_i^{y_i} \exp(-\mu_i)}{y_i!} = \{y_i \ln \mu_i - \mu_i - \ln y_i!\}$$
$$Y_i|\mathbf{x}_i \sim \text{EF}(\ln \mu_i, \phi = 1, w_i = 1)$$
$$\text{E}[Y_i|\mathbf{x}_i] = \mu_i$$
$$\text{Var}[Y_i|\mathbf{x}_i] = \mu_i$$

- Systematic component (using canonical link function)

$$\mu_i = \exp(\eta_i) \iff \ln \mu_i = \eta_i$$

Remark 92 (Main characteristics). It's:

- a nonlinear model, due to the log/exponential link function, both in regressors and betas
- an heteroskedastic by design: an increase in the expected value causes an increases in the variance.

Remark 93 (Dispersion). In the model the conditional variance is equal to the conditional expected value. Other situations could be

- *overdispersion*: occurs when conditional variance is larger than conditional expected value
- *underdispersion*: when conditional variance is larger than conditional expected value

In these situations poisson regression is not suitable.

Remark 94.

Example 12.1.2 (Count variables and exposure levels)**.** In the count stuff its important to handle the exposure to have a sort of normalization (eg in order to not impute to a certain regressor a difference that is dued to difference in exposure level, especially in observational study where exposure is tipically given and cannot be influenced by the researcher). Regarding the examples introduced before, some reasonable exposure level could be:

- for number of car accident claims made by policyholders to an insurance company the *number of policyholders with a specific covariate pattern*
- for number of points scored by a basketball player during a regular season the *number of minutes (time) played during the regular season*
- for number of imperfections on a glass plate the *dimension (surface) of the glass plate*
- for number of patients with a given disease hospitalised in a given town the *number of inhabitants of the town* $\times$ *time length*

12.1.2 Introduction of offsets (*compensating terms*)

Suppose for each unit we have the exposure level w_i of the i-th sample unit:

- one way to deal with exposure level would be divide y_i by w_i
- in the context of a poisson distribution this is somehow equivalent to introduce a modification to the linear predictor so one way to deal is by introducing offset/compensating terms. The linear predictor associated with the i-th statistical unit becomes

$$\eta_i^* = \eta_i + \ln w_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \beta_{p+1} \ln w_i$$

  with offset equal to $\ln w_i$ and β_{p+1} which is not estimated but restricted/set equal to 1.
  Adding $\ln w_i$ to the predictor means that

$$\exp(\eta_i^*) = w_i \exp(\eta_i)$$

  $\exp(\eta_i)$ will be the expected number of events per unit of exposure while $\exp(\eta_i^*)$ the number of events (which is obtained by multiplying the expected number of events for unit of exposure for the actual exposure)

In this case the model become assuming

- $Y_i|\mathbf{x}_i \sim \text{Poi}(\exp(\eta_i^*))$ or, equivalently, $f(y_i|\mathbf{x}_i) = \dfrac{[w_i \exp(\eta_i)]^{y_i} \exp(-w_i \exp(\eta_i))}{y_i!}$
- $Y_i|\mathbf{x}_i$ independent $(i = 1, \ldots, n)$

Important remark 90. The base poisson model is for count; by including exposure as offset we can study rates where exposure level matters/differs.
inclusion of an offset in the linear predictor allows to deal with Poisson regression models and regression models for rate derived from Poisson distributions within a "unified" framework; the attention will be focused on the former in the following

12.1.3 Interpretation of the model estimates

In the poisson model with the exponential link function the conditional expected value/mean count is linked to the covariates via a non linear function *both* in the regressor and in the parameters:

$$\mathrm{E}\left[Y_i|\mathbf{x}_i\right] = \exp\left(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}\right) \tag{12.1}$$

If we want to evaluate the effect of a single regressor x_j on the expected outcome $\mathrm{E}\left[Y_i|\mathbf{x}_i\right]$, we focus on the first partial derivative of the expression above, which shows that each regressor has a non-linear effect:

$$\frac{\partial}{\partial x_{ji}}\mathrm{E}\left[Y_i|\mathbf{x}_i\right] = \beta_j \exp\left(\eta_i\right)$$

this latter shows how:

- the *direction of the change* in $\mathrm{E}\left[Y_i|\mathbf{x}_i\right]$ due to a unit increase in x_{ji} depends on the *sign* of β_j (being the other exponential strictly positive). If we have a positive β_j an increase in x_{ji} will lead to an increase in $\mathrm{E}\left[Y_i|\mathbf{x}_i\right]$ and viceversa.
- the *magnitude* of the change is not constant as in case of gaussian regression, but depends also on the values of all the regressors (in η_i).

So the interpretation of regression coefficient is slightly more complicated (as in the case of any GLM with a nonlinear/identity link function).

In the specific context of poisson regression, however, we can take advantage of the fact that we're using the exponential as link function. (as in log-normal regression) We're applying the exponenytial to a linear predictor, which so can be written as product of several factor, each involving just one regression coefficient

Multiplicative models for $\mathrm{E}\left[Y_i|\mathbf{x}_i\right]$:

- by the properties of the exponential, we can rewrite the conditional expected value of 12.1, say *before* applying a unit increase in x_{ji}, as:

$$\mathrm{E}\left[Y_i|\mathbf{x}_i\right] = \exp\left(\beta_0\right) \cdot \ldots \cdot \exp\left(\beta_j x_{ji}\right) \cdot \ldots \cdot \exp\left(\beta_p x_{pi}\right)$$

- the conditional expected value *after* a unit increase in x_{ji} will become

$$\begin{aligned}\mathrm{E}\left[Y_i|\mathbf{x}_i+\right] &= \exp\left(\beta_0\right) \cdot \ldots \cdot \exp\left[\beta_j(x_{ji}+1)\right] \cdot \ldots \cdot \exp\left(\beta_p x_{pi}\right) \\ &= \exp\left(\beta_0\right) \cdot \ldots \cdot \exp\left[\beta_j x_{ji}\right] \cdot \exp\left(\beta_j\right) \cdot \ldots \cdot \exp\left(\beta_p x_{pi}\right)\end{aligned}$$

Therefore:

- if we take the ratio, it turns out that the multiplicative change due to a unit increase in x_{ji} is just:

$$\frac{\mathrm{E}\left[Y_i|\mathbf{x}_i+\right]}{\mathrm{E}\left[Y_i|\mathbf{x}_i\right]} = \exp\left(\beta_j\right) = \begin{cases} < 1 & \text{if } \beta_j < 0 \\ = 1 & \text{if } \beta_j = 0 \\ > 1 & \text{if } \beta_j > 0 \end{cases}$$

- thus percentage change due to a unit increase in x_{ji} is thus $\left[\exp\left(\beta_j\right) - 1\right]$ %.

12.1.4 Log-likelihood

What happens to the loglik? By doing substitution with the results for loglik using the canonical link function (log/exp):

$$\begin{aligned} l(\boldsymbol{\beta}) &= \sum_{i=1}^{n}\left[y_i\eta_i - \exp(\eta_i) - \ln(y_i!)\right] \\ &= \left[\beta_0\sum_{i=1}^{n} y_i + \ldots + \beta_p\sum_{i=1}^{n} y_i x_{pi}\right] - \sum_{i=1}^{n}\exp(\eta_i) - \sum_{i=1}^{n}\ln y_i! \end{aligned}$$

it turns out that the loglikelihood depends on y_i only through the summaries $\sum_{i=1}^{n} y_i, \ldots, \sum_{i=1}^{n} y_i x_{pi}$ which are therefore (minimal) sufficient statistics for $\boldsymbol{\beta}$.

Remark 95. When exposure levels w_i are considered, it is possible to prove that the log-likelihood is only changed by the additive constant $\sum_{i=1}^{n} y_i \ln w_i$, that does not involve $\boldsymbol{\beta}$

12.1.5 Score function

Particularizing the general formula for the generic element of the score function for a GLM with canonical link function we have that in the Poisson we don't have nuisance parameter (so $\phi = 1$) and $\mathbb{E}[Y_i|\mathbf{x}_i] = \exp(\eta_i)$. So:

$$U_j(\boldsymbol{\beta}) = \frac{1}{\phi}\sum_{i=1}^{n} w_i \{y_i - \mathbb{E}[Y_i|\mathbf{x}_i]\} x_{ji} = \frac{1}{1}\sum_{i=1}^{n} 1 \cdot [y_i - \exp(\eta_i)] x_{ji}$$
$$= \sum_{i=1}^{n} [y_i - \exp(\eta_i)] x_{ji}$$

Remark 96. When exposure levels w_i are considered, it is possible to prove that the score function remains unchanged, provided that the offset terms $\ln w_i$ are included in the linear predictors η_i

12.1.6 Fisher information

Considering the general formula for the generic element of the (observed and expected) Fisher information matrix for a GLM with canonical link function:

$$i_{jl}(\boldsymbol{\beta}) = I_{jl}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(\frac{w_i}{\phi}\right)^2 \mathrm{Var}[Y_i|\mathbf{x}_i] x_{ji}x_{li} = \sum_{i=1}^{n} \left(\frac{1}{1}\right)^2 \exp(\eta_i) x_{ji}x_{li}$$
$$= \sum_{i=1}^{n} \exp(\eta_i) x_{ji}x_{li}$$

Remark 97. When exposure levels w_i are considered, it is possible to prove that the Fisher information matrices remain unchanged, provided that the offset terms $\ln w_i$ are included in the linear predictors

12.1.7 Hessian matrix

The (j, l)-th element of the Hessian matrix of the log-likelihood function

$$H_{jl}(\boldsymbol{\beta}) = \frac{\partial^2}{\partial\beta_j \partial\beta_l} l(\boldsymbol{\beta}) = -i_{jl}(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \exp(\eta_i) x_{ji}x_{li}$$

12.1.8 Matrix representation

This representation will be useful in the following. Let:

- $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\top$ be the n-dimensional vector that contains the observed values of the dependent variable on the n sample units
- $\mathbf{X}$ be the $n \times (p+1)$ matrix that contains the observed values of the regressors on the n sample units
- $\boldsymbol{\mu}$ be the n-dimensional vector that contains the conditional expected values of the dependent variable for the n sample unit

  $$\boldsymbol{\mu} = (\exp(\eta_1), \exp(\eta_2), \ldots, \exp(\eta_n))^\top$$

  In this case they're the exponential of the linear predictor; here differently from gaussian model, due to the link function we can explicitly define $\boldsymbol{\mu}$ using linear algebra (eg exponential of a vector or matrix is not defined)
- $\mathbf{W}$ be the $n \times n$ diagonal matrix that contains the conditional variances of the dependent variable for the n sample unit on the main diagonal

  $$\mathbf{W} = \begin{bmatrix} \exp(\eta_1) & 0 & 0 & \ldots & 0 \\ 0 & \exp(\eta_2) & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \exp(\eta_n) \end{bmatrix}$$

  Thanks to the poisson case each element in the diagonal correspond to the element in $\boldsymbol{\mu}$; the element of the diagonal are not constant, differently from the gaussian case where it was $\sigma^2\mathbf{I}$

Thanks to these definitions we can rewrite in a compact way all (not only the single elements of) the:

- score function:

$$U(\boldsymbol{\beta}) = \mathbf{X}^\top [\mathbf{y} - \boldsymbol{\mu}] \tag{12.2}$$

  Each element computes the inner product of the value of a regressor and the difference between $\mathbf{y}$ and $\boldsymbol{\mu}$
  eg for the first element is the first column of $\mathbf{X}$ (all constant) multiplied for the difference between the observed values and the predictions

- Hessian matrix of the log-likelihood function

$$H(\boldsymbol{\beta}) = -\mathbf{X}^\top \mathbf{W} \mathbf{X} \tag{12.3}$$

- (observed and expected) Fisher information matrix

$$i(\boldsymbol{\beta}) = I(\boldsymbol{\beta}) = \mathbf{X}^\top \mathbf{W} \mathbf{X} \tag{12.4}$$

These are similar/generalization of what seen in the gaussian models

Remark 98. This representation is convenient in ML estimation.

12.1.9 Maximum likelihood estimation

Definition 12.1.1 (ML estimate)**.** $\hat{\mathbf{b}}$ is the maximum likelihood estimate of $\boldsymbol{\beta}$ if and only if

$$l\left(\hat{\mathbf{b}}\right) = \max_{\mathbf{b}} l(\mathbf{b})$$

and that occurs if and only if:

$$U\left(\hat{\mathbf{b}}\right) = \frac{\partial}{\partial \boldsymbol{\beta}} l(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\hat{\mathbf{b}}} = \mathbf{0}_{p+1}$$
$$H\left(\hat{\mathbf{b}}\right) = \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} l(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\hat{\mathbf{b}}} \quad \text{is negative definite}$$

that is the score vector (first derivative) evaluated at $\hat{\mathbf{b}}$ is null and the log-likelihood hessian matrix evaluated at $\hat{\mathbf{b}}$ is negative definite (negative second derivative)

Important remark 91 (Procedure). Problem here is that if we look at the first condition regarding the score function, the maximum likelihood estimate $\hat{\mathbf{b}}$ can be obtained by solving the following system of equations with respect to $\mathbf{b}$:

$$U(\mathbf{b}) = \mathbf{X}^\top [\mathbf{y} - \mathbf{m}] = \mathbf{0}_{p+1}$$

where we replaced the unknown expected value vector $\boldsymbol{\mu}$ with an estimated one (at a given vector $\mathbf{b}$), that is

$$\mathbf{m} = \begin{bmatrix} \exp(b_0 + b_1 x_{11} + b_2 x_{21} + \ldots + b_p x_{p1}) \\ \exp(b_0 + b_1 x_{12} + b_2 x_{22} + \ldots + b_p x_{p2}) \\ \vdots \\ \exp(b_0 + b_1 x_{1n} + b_2 x_{2n} + \ldots + b_p x_{pn}) \end{bmatrix} = \begin{bmatrix} \exp(\mathbf{x}_1^\top \mathbf{b}) \\ \exp(\mathbf{x}_2^\top \mathbf{b}) \\ \vdots \\ \exp(\mathbf{x}_n^\top \mathbf{b}) \end{bmatrix}$$

Remark 99. Problem is that this is system of non-linear equations in $\mathbf{b}$ (having the exponential): in general, this system does not have an explicit solution (*it is not possible to obtain an analytical formula to compute* $\hat{\mathbf{b}}$). So we need to rely on numerical algorithm to find an approximation to the solution of the system/the problem.

Remark 100. There are few exceptions in particular cases where this does not happen; in general most of glm with a link function that is not the identity will have this fact.

Remark 101. We see two algorithm

12.2 Optimization algorithms

12.2.1 The Newton-Raphson algorithm

Considering a:

- $\boldsymbol{\theta}$ k-dimensional parameter vector
- a scalar Log-likelihood function $l(\boldsymbol{\theta})$
- a $k \times 1$ *vector* Log-likelihood gradient $U(\boldsymbol{\theta}) = \dfrac{\partial}{\partial \boldsymbol{\theta}} l(\boldsymbol{\theta})$
- a $k \times k$ Log-likelihood Hessian matrix $H(\boldsymbol{\theta}) = \dfrac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} l(\boldsymbol{\theta})$

the Newton-Raphson algorithm finds a sequence of approximated solutions $\hat{\mathbf{t}}$ of a (non-linear) system

$$U(\mathbf{t}) = \mathbf{0}_k$$

The algorithm goes like that:

- We choose an initial solution $\mathbf{t}^{(1)}$ for $\hat{\mathbf{t}}$, this system is locally approximated with a linear system.
  First-order Taylor series expansion is the crucial ingredient: starting from a system of linear equation we obtain a local approximation of this system such that is composed of linear equation. This can be done using first order Taylor series expansion (basic math tools).
  So replacement is as follows:

  $$U(\mathbf{t}) \approxeq U\left(\mathbf{t}^{(1)}\right) + H\left(\mathbf{t}^{(1)}\right)\left(\mathbf{t} - \mathbf{t}^{(1)}\right) = \mathbf{0}_k$$

  where:

  - $U\left(\mathbf{t}^{(1)}\right) = \frac{\partial}{\partial \boldsymbol{\theta}} l(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\mathbf{t}^{(1)}}$ is our *log-likelihood gradient (score) evaluated at* $\mathbf{t}^{(1)}$
  - $H\left(\mathbf{t}^{(1)}\right) = \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} l(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\mathbf{t}^{(1)}}$ is our *log-likelihood Hessian matrix evaluated at* $\mathbf{t}^{(1)}$

  So the system is replaced with its evaluation for the initial solution plus the hessian matrix (evaluated at initial solution) times the difference between the unknown $\mathbf{t}$ minus the initial solution. This latter is a local approximation of the system composed of linear equations in unknown $\mathbf{t}$
- a second approximation $\mathbf{t}^{(2)}$ to $\hat{\mathbf{t}}$ is obtained by solving the linear system above with respect to $\mathbf{t}$

  $$U\left(\mathbf{t}^{(1)}\right) + H\left(\mathbf{t}^{(1)}\right)\left(\mathbf{t} - \mathbf{t}^{(1)}\right) = \mathbf{0}_k$$

  In particular:

  $$\begin{aligned} U\left(\mathbf{t}^{(1)}\right) + H\left(\mathbf{t}^{(1)}\right)\left(\mathbf{t} - \mathbf{t}^{(1)}\right) = \mathbf{0}_k &\iff H\left(\mathbf{t}^{(1)}\right)\mathbf{t} = H\left(\mathbf{t}^{(1)}\right)\mathbf{t}^{(1)} - U\left(\mathbf{t}^{(1)}\right) \\ &\overset{(1)}{\iff} \mathbf{t}^{(2)} = \mathbf{t}^{(1)} - H\left(\mathbf{t}^{(1)}\right)^{-1} U\left(\mathbf{t}^{(1)}\right) \end{aligned}$$

  where in (1) we just premultiplied both members for the inverse of $H\left(\mathbf{t}^{(1)}\right)$ (which *must be invertible*)
- in general in the r-th step, the $(r+1)$-th approximation $\mathbf{t}^{(r+1)}$ to $\hat{\mathbf{t}}$ is obtained using the recursive formula

  $$\mathbf{t}^{(r+1)} = \mathbf{t}^{(r)} - H\left(\mathbf{t}^{(r)}\right)^{-1} U\left(\mathbf{t}^{(r)}\right)$$

  If some regularity conditions are met, it is possible to prove that

  $$\lim_{r \to \infty} \mathbf{t}^{(r)} = \hat{\mathbf{t}}$$

  The sequence of approximations converges to the solution of the non-linear system $U(\mathbf{t}) = \mathbf{0}_k$

- Obv we need to choose when to stop: the recursive formula

$$\mathbf{t}^{(r+1)} = \mathbf{t}^{(r)} - H\left(\mathbf{t}^{(r)}\right)^{-1} U\left(\mathbf{t}^{(r)}\right)$$

is repeatedly applied until a stopping criterion is met. We could choose alternatively as stopping criterion (tipically the second or the third, or a combination of the twos, are preferred):

1. the *euclidean norm between two consecutive approximations*

$$\|\mathbf{t}^{(r+1)} - \mathbf{t}^{(r)}\|_2 < \epsilon, \quad \text{with } \epsilon > 0$$

2. the *difference in the log-likelihood evaluated at two consecutive approximations*

$$l\left(\mathbf{t}^{(r+1)}\right) - l\left(\mathbf{t}^{(r)}\right) < \epsilon, \quad \text{with } \epsilon > 0$$

3. the euclidean norm of the score function (being this closer and closer to $\mathbf{0}$ as steps goes by)

$$\|U(\mathbf{t}^{(r+1)})\|_2 < \epsilon, \quad \text{with } \epsilon > 0$$

The final approximation to $\hat{\mathbf{t}}$ is given by the last element of the sequence

Example 12.2.1 (Example when $k = 1$)**.** In a simple situation where θ is just a scalar (one parameter to optimize); in figure 12.1 the first derivative of the loglikelihood which is maximized for some value between 1.5 and 2. In

- (b) our initial guess is $t^{(1)}$ near 0.5: resorting on the first order of taylor series expansion means considering the tangent line to the function in $t^{(1)}$. So we're locally approximating the function U using its tangent.
  Having replaced a nonlinear with a linear system we find the solution between the tangent line and the 0 (here is something just above 1) and use it as a new guess
- (c), (d), (e) this iterates on and on

12.2.2 Fisher scoring algorithm

Remark 102. Another popular algorithm can be thought as a variation of Newton-raphson: NR is based on first order Taylor series expansion in which we have the hessian matrix of loglik; in Fisher scoring algorithm we're replacing the Hessian matrix (observed fisher information matrix) with its expected value (expected fisher information matrix).
The log-likelihood Hessian matrix evaluated at $\mathbf{t}^{(r)}$ is replaced with its expected value in the recursive formula:

$$E\left[H\left(\mathbf{t}^{(r)}\right)\right] = -I\left(\mathbf{t}^{(r)}\right)$$

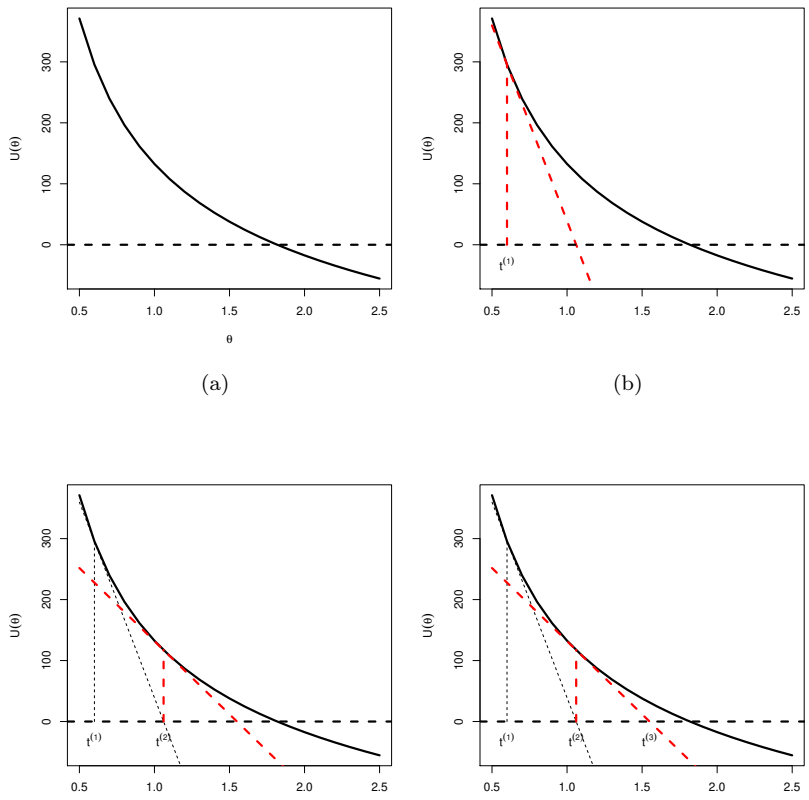
where $I\left(\mathbf{t}^{(r)}\right)$ is the expected Fisher information evaluated at $\mathbf{t}^{(r)}$.

Definition 12.2.1 (Fisher scoring algorithm)**.** The $(r+1)$-th approximation $\mathbf{t}^{(r+1)}$ to $\hat{\mathbf{t}}$ is obtained using the recursive formula

$$\mathbf{t}^{(r+1)} = \mathbf{t}^{(r)} + I\left(\mathbf{t}^{(r)}\right)^{-1} U\left(\mathbf{t}^{(r)}\right)$$

so in this last rather than a $-$ here we have a $+$.

Remark 103. While NR is a general algorithm, the Fisher scoring is somewhat a variation tailored to deal with loglik function for ML estimation (in order to use it we need to compute the expected value which has a meaning iff we're dealing with samples/likelihood/sample space/inference).



(a) (b)

(c) (d)

Figure 12.1: Newton raphson

12.2.3 A comparison between the two algorithms

Important remark 92. Generally speaking:

- Newton-Raphson is faster in convergence/reaching stopping criterion
- Fisher scoring is more stable (`glm` R function is based on this)
  - When using the expected value of Fisher information matrix we've basically the guarantee that we'll deal with a matrix which is always invertible
  - with NR we're working with Hessian matrix which in some situations (outside the glm framework) is not invertible or might become positive definite; this issues can be avoided by considering the expected value of the Hessian matrix
- it is possible to define mixed strategies: the Fisher scoring algorithm is used for a given number of steps, then the final steps are performed according to the Newton-Raphson algorithm

Important remark 93. In the context of glm, when we use of the canonical link function the Newton-Raphson coincides with the Fisher scoring algorithm since expected and observed fisher information matrix coincides and the choice between them is irrelevant

Remark 104. After this recap of NR and Fisher scoring let's see what happens in Poisson model

12.2.4 Application to Poisson regression estimation

Given the initial guess $\mathbf{b}^{(1)}$ to $\hat{\mathbf{b}}$ we can obtain first and second derivative by substitution in eq 12.2 and 12.3

$$U\left(\mathbf{m}^{(1)}\right) = \mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}^{(1)}\right], \qquad H\left(\mathbf{m}^{(1)}\right) = -\mathbf{X}^\top \mathbf{W}^{(1)} \mathbf{X}$$

where we obtained the estimated mean and variance/covariance again by substituting:

$$\mathbf{m}^{(1)} = \begin{bmatrix} \exp(\mathbf{x}_1^\top \mathbf{b}^{(1)}) \\ \exp(\mathbf{x}_2^\top \mathbf{b}^{(1)}) \\ \vdots \\ \exp(\mathbf{x}_n^\top \mathbf{b}^{(1)}) \end{bmatrix}, \quad \mathbf{W}^{(1)} = \begin{bmatrix} \exp(\mathbf{x}_1^\top \mathbf{b}^{(1)}) & 0 & 0 & \dots & 0 \\ 0 & \exp(\mathbf{x}_2^\top \mathbf{b}^{(1)}) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \exp(\mathbf{x}_n^\top \mathbf{b}^{(1)}) \end{bmatrix}$$

As provided by the algorithm, the starting system of non-linear equations in $\mathbf{b}$:

$$\mathbf{X}^\top [\mathbf{y} - \mathbf{m}] = \mathbf{0}_{p+1}$$

is locally approximated with the system of linear equations in $\mathbf{b}$

$$\mathbf{X}^\top [\mathbf{y} - \mathbf{m}] \approxeq \mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}^{(1)}\right] - \mathbf{X}^\top \mathbf{W}^{(1)} \mathbf{X} \left[\mathbf{b} - \mathbf{b}^{(1)}\right] = \mathbf{0}_{p+1}$$

Here NR and Fisher scoring algorithm coincides since we're using canonical link and the choice between them is irrelevant (expected and observed fisher information matrix coincides).
A new approximation $\mathbf{b}^{(2)}$ to $\hat{\mathbf{b}}$ is obtained solving the latter system wrt $\mathbf{b}$:

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

And finally (to keep the comparison with algorithm section formula):

$$\begin{array}{lll} \mathbf{b}^{(2)} & = \mathbf{b}^{(1)} & \underbrace{+\left(\mathbf{X}^\top \mathbf{W}^{(1)} \mathbf{X}\right)^{-1}}_{-H\left(\mathbf{t}^{(1)}\right)^{-1}} \quad \underbrace{\mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}^{(1)}\right]}_{U\left(\mathbf{t}^{(1)}\right)} \\ \mathbf{t}^{(2)} & = \mathbf{t}^{(1)} & \end{array}$$

Important remark 94 (Invertibility of $\mathbf{X}^\top \mathbf{W}^{(1)} \mathbf{X}$). For the application of the algorithm we need $\mathbf{X}^\top \mathbf{W}^{(1)} \mathbf{X}$ to be invertible; this happens if and only if:

- $\mathbf{X}$ has full column rank (as usual);

- $\mathbf{W}^{(1)}$ has strictly positive elements on its main diagonal: this is a new condition. In case of Poisson regression proving this is straightforward (on the diagonal we're using an exponential of the linear predictor).

Important remark 95 (Poisson estimation). In general we apply the recursive recursive formula ...

$$\begin{array}{rcl} \mathbf{b}^{(r+1)} & = & \mathbf{b}^{(r)} \underbrace{+\left(\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X}\right)^{-1}}_{-H\left(\mathbf{t}^{(r)}\right)^{-1}} \underbrace{\mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}^{(r)}\right]}_{U\left(\mathbf{t}^{(r)}\right)} \\ \mathbf{t}^{(r+1)} & = & \mathbf{t}^{(r)} \end{array}$$

... until a stopping criteria is met, either regarding:

- *euclidean norm between two consecutive approximations*

$$\|\mathbf{b}^{(r+1)} - \mathbf{b}^{(r)}\|_2 < \epsilon \quad \text{with } \epsilon > 0$$

- *difference in the log-likelihood evaluated at two consecutive approximations*

$$l\left(\mathbf{b}^{(r+1)}\right) - l\left(\mathbf{b}^{(r)}\right) < \epsilon \quad \text{with } \epsilon > 0$$

Remark 105. Usually with very few steps (4/5) the Fisher scoring is able to achieve convergence coming close enough to the maximum likelihood estimates (thanks to the fact that the loglik of a glm is usually well behaved with unique maximum and no saddle point)

12.2.5 Iterative reweighted least squares

Remark 106. In this section we give a closer look at the recursive formula

$$\mathbf{t}^{(r+1)} = \mathbf{t}^{(r)} - H\left(\mathbf{t}^{(r)}\right)^{-1} U\left(\mathbf{t}^{(r)}\right)$$

in the context of glm

Point is that manipulating a bit the expression used for recursive optimization

$$\mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}^{(r)}\right] - \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X} \left[\mathbf{b} - \mathbf{b}^{(r)}\right] = \mathbf{0}_{p+1}$$

we can re-express it in order to isolate $\mathbf{b}$ on one side of the system; then we move on:

$$\begin{aligned} \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X}\mathbf{b} &= \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X}\mathbf{b}^{(r)} + \mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}^{(r)}\right] \\ &\overset{(1)}{=} \mathbf{X}^\top \mathbf{W}^{(r)} \left\{ \mathbf{X}\mathbf{b}^{(r)} + \left[\mathbf{W}^{(r)}\right]^{-1} \left[\mathbf{y} - \mathbf{m}^{(r)}\right] \right\} \\ &\overset{(2)}{=} \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{z}^{(r)} \end{aligned}$$

where

- in (1) we work to isolate the term $\mathbf{X}^\top \mathbf{W}^{(r)}$ by postmultiplying at the second member $\mathbf{X}^\top$ for $W^{(r)}$ and its inverse and then gathering $\mathbf{X}^\top \mathbf{W}$ by both addends
- in (2) the last we just performed the naming/substitution

$$\mathbf{z}^{(r)} = \mathbf{X}\mathbf{b}^{(r)} + \left[\mathbf{W}^{(r)}\right]^{-1} \left[\mathbf{y} - \mathbf{m}^{(r)}\right]$$

  which is a vector that can be obtained once we've fixed the value of $\mathbf{b}^{(r)}$ by doing the proper substitution in the other vectors/matrix

In the end our system can be written as

$$\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X}\mathbf{b} = \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{z}^{(r)}$$

meaning that the solution of the system with respect to $\mathbf{b}$ (the $(r+1)$-th $\mathbf{b}^{(r+1)}$ to $\hat{\mathbf{b}}$) can be written by premultiplying both the temrs of the system by $(\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X})^{-1}$

$$\begin{aligned} \mathbf{b}^{(r+1)} &= (\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{z}^{(r)} \\ &= \ldots \qquad\qquad = \mathbf{b}^{(r)} + \left(\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X}\right)^{-1} \left[\mathbf{y} - \mathbf{m}^{(r)}\right] \end{aligned}$$

Important remark 96 (The iterative reweighted least square thing). It is interesting to note that in

$$\mathbf{b}^{(r+1)} = (\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{z}^{(r)}$$

we find something familiar with what we've seen for gaussian regression model where

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

The difference between this and the solution for the ordinary least square:

- this is an iterative/recursive expression while the ols is a closed/direct formula
- the vector $\mathbf{z}$ plays the role of the ols $\mathbf{y}$
- in the formula above we have the presence of $\mathbf{W}^{(r)}$ which elements basically plays the role of a weights (which are equal to conditional variances): the larger the conditional variance, the stronger will be the role of the corresponding unit in determining the $\mathbf{b}^{(r+1)}$

Thanks to this second connection between general OLS solution with the actual expression used in our recursive glm algorithm we can call NR/Fisher scoring as an *iterative reweigthed least square*: we're iteratively applying least square after reweighting each unit in the sample. This iterative formula arises only in the context of glm

weighted least squares estimate for the regression coefficients of a linear regression model that links the dependent variable $Z^{(r)}$ to the regressors $X_1, \ldots, X_p$, with individual weights equal to $m_i^{(r)}$ (current estimate of the expected value)

12.2.5.1 Adjusted/pseudo dependent variable

Remark 107. For the fact that in the recursive formula the vector $\mathbf{z}$ plays the role of the usual ols $\mathbf{y}$, its elements are called either *ajdusted* or *pseudo* dependent variable

Definition 12.2.2. The i-th element of $\mathbf{z}^{(r)}$ is

$$z_i^{(r)} = \mathbf{x}_i^\top \mathbf{b}^{(r)} + \frac{1}{m_i^{(r)}} \left[y_i - m_i^{(r)} \right] \tag{12.5}$$

where $m_i^{(r)}$ is the current estimate of the expected value and is the value of the so called *adjusted dependent variable - pseudo-dependent variable* at the r-th step of the Newton-Raphson algorithm.

Remark 108. So basically we augment the actual value of the linear predictor $\eta_i^{(r)} = \mathbf{x}_i^\top \mathbf{b}^{(r)}$ with a sort of stadardized residual, obtained as difference between y_i and current estimate of the expected value (coming from $\mathbf{m}$) divided by the current estimate of the variance (same as expected value here for poisson) coming from $\mathbf{W}$

Remark 109. In equation 12.5 of $z_i^{(r)}$ one can recognize the first order Taylor series expansion of the $\ln(y_i)$, the canonical link function used to define the poisson regression model applied to the dependent variable.
$z_i^{(r)}$ can be interpreted as an approximation to $\ln(y_i)$ obtained using a first order Taylor series expansion at $m_i^{(r)}$

$$\ln(y_i) \approxeq \ln\left(m_i^{(r)}\right) + \left.\frac{\partial \ln(y_i)}{\partial y_i}\right|_{y_i = m_i^{(r)}} \left[y_i - m_i^{(r)} \right]$$

Remark 110. This iterated stuff and replacing y with iterated z will arise in any glm

Remark 111. This connection of pseudo dependent variable and actual values of the dependent variable (z and y) can be exploited to initialize the algorithm (we still have to choose $\mathbf{b}^{(1)}$)

12.2.5.2 Initialisation

The Newton-Raphson can be initialised by setting:

- $\mathbf{W}^{(0)} = \mathbf{I}_n$ identity matrix
- $z_i^{(0)} = \ln(y_i + 0.5)$ (0.5 *is added in order to avoid* $\ln(0)$ in zero counts)

- as an educated guess instead of random initialization we can use

$$\mathbf{b}^{(1)} = \left(\mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{z}^{(0)} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{z}^{(0)}$$

  which after substitution basically is applying ols to transformed dependent variable; this can be used to initialize.

Remark 112. Once having $\mathbf{b}_1$ we will revaluate $\mathbf{W}$, $\mathbf{z}$ and feed them in the recursive expression until a stopping criterion is met

12.2.5.3 Maximum likelihood estimator

Remark 113. The ML estimate obtained recursively will be a realization of a ML estimator: each sample will be characterized by its own y and loglikelihood, so the estimator will have its own distribution.
In absence of an analytical formula for $\hat{\mathbf{B}}$, the *properties of the maximum likelihood estimator must be investigated starting from the (asymptotic) properties of the score function* $U(\boldsymbol{\beta})$

Under general regularity conditions (that hold for Poisson regression models), as we've said, it can be shown that the score function is characterized by an asymptotic mvn distribution with following features:

$$U(\boldsymbol{\beta}) = \mathbf{X}^\top \left[\mathbf{y} - \boldsymbol{\mu}\right] \xrightarrow{d} MVN_{p+1}\left(\mathbf{0}_{p+1}, \mathbf{X}^\top \mathbf{W} \mathbf{X}\right)$$

The trick to exploit this general result to derive the asymptotic properties of ML estimator for the betas in Poisson is using following (linear) Taylor series expansion at the maximum likelihood estimate: by construction we know that it will be equal to zero (the score function is always equal to zero when we consider the maximum likelihood estimate) so equating this and using the approximation we end up with

$$\mathbf{0}_{p+1} = U\left(\hat{\mathbf{B}}\right) \approxeq U(\boldsymbol{\beta}) - I(\boldsymbol{\beta}) \left[\hat{\mathbf{B}} - \boldsymbol{\beta}\right]$$

(here we're somewhat reverting: rather than expanding the score function at a known value we do it at the unknown true value of the parameter $\boldsymbol{\beta}$). In the expression we have:

- the score function evaluated at $bm\beta$, $U(\boldsymbol{\beta})$
- the expected fisher info evaluated at $bm\beta$, $I(\boldsymbol{\beta})$
- the unknown $\hat{\mathbf{B}}$
- the "constant" $\boldsymbol{\beta}$

Thus it is possible to prove that the score function $U(\boldsymbol{\beta})$ can be approximated as a linear transformation of the maximum likelihood estimator $\hat{\boldsymbol{B}}$

$$U(\boldsymbol{\beta}) \approxeq I(\boldsymbol{\beta}) \left[\hat{\mathbf{B}} - \boldsymbol{\beta}\right]$$

This is something that we could do in theory (we don't know the actual value of β): but we know that the score function is related to the maximum likelihood estimator.
Having the score function a mvn distribution, and applying on it an approximated linear transformation, we can exploit the mvn distribution properties to show that the maximum likelihood estimator has at least an asymptotic mvn distribution.
We have that:

$$U(\boldsymbol{\beta}) \approxeq I(\boldsymbol{\beta}) \left[\hat{\mathbf{B}} - \boldsymbol{\beta}\right] = \mathbf{X}^\top \mathbf{W} \mathbf{X} \left[\hat{\mathbf{B}} - \boldsymbol{\beta}\right] \xrightarrow{d} MVN_{p+1}\left(\mathbf{0}_{p+1}, \mathbf{X}^\top \mathbf{W} \mathbf{X}\right)$$

therefore if $U(\boldsymbol{\beta})$ is mvn then even $\hat{\mathbf{B}}$ will be mvn and especially From now on we can exploit the property of mvn: by premultipling by the inverse of fisher information matrix and adding the unknown constant β we end with:

$$\hat{\mathbf{B}} \xrightarrow{d} MVN_{p+1}\left(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}\right)$$

Thus the *the maximum likelihood estimator is asymptotically unbiased, efficient and has a multivariate Gaussian distribution.*

Important remark 97 (Perspective). This is one of the most important results about ML estimation which is the reason why the ML methods is one of the most widely method to perform inference: no matter which model we're dealing with (glm etc), if some basic regularity condition are met (and are met for most glm) the ML method will provide estimators that are "well behaved" (have all the required properties for an estimator, at least asymptotically: unbiasedness, efficiency and normality).
For gaussian linear regression models this property holds not only asymptotically but for any value of n (also for finite samples).

Remark 114. A rough rule of thumb is that we should have 10/20 observation per parameter in the model to invoke the "large sample size" approximation and the multivariate normality for glm parameters.

Important remark 98 (Other useful equations). we

- when we have a mvn we can kinda standardize it to have a standard mvn, and here is:

$$\left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)^{1/2}\left[\hat{\mathbf{B}}-\boldsymbol{\beta}\right] \xrightarrow{d} MVN_{p+1}(\mathbf{0}_{p+1}, I_{p+1})$$

  where we diminished for the expected value and premultiplied for the "square root" $\left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)^{1/2}$. This result is interesting for the following fact

- Once we standardize and compute the sum of the squared value of the standardized gaussian distribution we end up with a chi-square random variuable

$$\left[\hat{\mathbf{B}}-\boldsymbol{\beta}\right]^\top \left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)\left[\hat{\mathbf{B}}-\boldsymbol{\beta}\right] \xrightarrow{d} \chi^2_{(p+1)}$$

  This above is the quadratic form based on the asymptotic $\left(\mathbf{X}^\top \mathbf{W}\mathbf{X}\right)^{1/2}\left[\hat{\mathbf{B}}-\boldsymbol{\beta}\right]$: the quadratic form can be interpreted as the sum of squared independent standardized gaussian random variable. So it will be chisq distributed with degrees of freedom equal to the number of gaussian we sum (squared).

12.2.5.4 Estimation of the asymptotic variance

The result

$$\hat{\mathbf{B}} \xrightarrow{d} MVN_{p+1}\left(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}\right)$$

provides $I(\boldsymbol{\beta})^{-1}$ as the asymptotic variance-covariance matrix of the ML estimator; this asymptotic matrix will depend on $\boldsymbol{\beta}$, but when we performed the estimate for $\boldsymbol{\beta}$ we can compute an estimate for the asymptotic variance as well.
So $\mathbf{X}^\top \mathbf{W}\mathbf{X}$ unknown (depends on $\boldsymbol{\beta}$) can be estimated using $\mathbf{X}^\top \hat{\mathbf{W}}\mathbf{X}$ by just plugging in the ML estimate where needed, that is:

$$\hat{\mathbf{W}} = \begin{bmatrix} \exp(\mathbf{x}_1^\top \hat{\mathbf{b}}) & 0 & 0 & \dots & 0 \\ 0 & \exp(\mathbf{x}_2^\top \hat{\mathbf{b}}) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \exp(\mathbf{x}_n^\top \hat{\mathbf{b}}) \end{bmatrix}$$

After computing $\mathbf{X}^\top \hat{\mathbf{W}}\mathbf{X}$ (estimate for expected fisher information) and then we've just to compute the inverse

$$\text{Var}\left[\hat{\mathbf{B}}\right] \approxeq \left(\mathbf{X}^\top \hat{\mathbf{W}}\mathbf{X}\right)^{-1}$$

Important remark 99. From the diagonal element of this matrix we'll have an estimate of the asymptotic variances for each regression coefficient.
We can use this information to perform t test, confidence interval for each beta and so on, as we did for gaussian linear regression model.
The only difference, again, is that are asymptotic results.

Important remark 100. The actual varcov matrix for ML estimator for regression coefficient of a poisson regression model is unknown, but if we have a sufficiently large sample we can approximate this variance using the expression above.

Important remark 101 (Uno sguardo indietro). By introducing the link function and all the glm-stuff, we pay a little price in features of the estimator ("only" asymptotically optimal) but otoh we gain a lot of flexibility

12.3 Deviance, residuals and model selection criteria

Remark 115. We've estimated our model; we have to check if it's adequate to describe the data or not.
We have to check wheter the assumption (both for probabilistic and deterministic component) characterizing the model we've fit to data are actually adequate or not.

12.3.1 (Residual) deviance

Remark 116. We can define the deviance also for poisson regression models; first ingredient to do it is to define the saturated model.

12.3.1.1 Saturated model

Whenever we fit a poisson regression model, we basically fit a model based on the two following assumptions (we included the link function directly in the specification of the probabilistic component):

$$M : \begin{cases} Y_i|\mathbf{x}_i \sim \text{Poi}(\exp(\eta_i)) & \\ & \text{independent } i = 1, \dots, n \\ \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} & \end{cases}$$

As in the gaussian setting we can associate a 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 number of unique values for η_i)
- if the simplest case where number of unique covariate patterns is equal to n (*each sample unit is characterised by a specific combination of regressor values*), the saturated model can be defined as follows:

  $$M_{sat} : Y_i|\mathbf{x}_i \sim \text{Poi}(\mu_i)\text{independent}i = 1, \dots, n$$

  that is we assume that each unit in the sample has a specific conditional expected value μ_i
- in this latter case we can try/fit this saturated model;

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

- its log-likelihood function will become simpler (we don't have lp and exponential):

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

  when it comes to compute the maximum likelihood estimate of μ_i by partial derivatives, we end up with simple structures involving only the first addend:

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

  in the second partial derivative (matrix) we'll have $-\frac{y_i}{\mu_i^2}$ on the main diagonal and 0 elsewhere (if we compute the second partial derivative with respect of μ_i and μ_j, we have to derive $\frac{y_i - \mu_i}{\mu_i}$ for μ_j which is 0).
  Setting the first derivative equal to zero (and checking it's a maximum on the second) leads to the ML estimate for μ_i is just the observed value $\hat{m}_i = y_i$.
  We're making no summary of the actual data, but in terms of adequacy the saturated model is the best possible model we can get); the value of the loglikelihood evaluated

at ML estimates is the *maximum possible value for the log-likelihood* associated with Poisson models for $\mathbf{Y}|\mathbf{X}$, given the observed sample $\mathbf{y}$, and is:

$$l(y_1, \dots, y_n) = \sum_{i=1}^{n} (\underbrace{y_i \ln y_i}_{0 \ln 0 \equiv 0} - y_i) - \sum_{i=1}^{n} \ln y_i!$$

here, by convention, whether one count in the data is zero we substitute $0 \ln 0$ (not defined) with just 0.
However, in general, any Poisson regression model for $\mathbf{Y}|\mathbf{X}$ will shows a maximum value for the log-likelihood that is *smaller* than this above (given the observed sample $\mathbf{y}$).

Remark 117. therefore, as done for gaussian model, we can quantify the loss of adequacy due to introduction of specific definition of systematic component/regressor by comparing the loglik of the saturated model with the one of the fitted model. (we take the saturated model as benchmark).
This can be done by taking the LRT statistics (or any of its transformation)

12.3.1.3 (Residual) deviance for a Poisson regression model

We defined the deviance as 2 times the logarithm of the ratio between the likelihood of saturated model and the one of the fitted:

$$\begin{aligned} D &= 2 \ln \left[\frac{L(y_1, \dots, y_n)}{L\left(\hat{\mathbf{b}}\right)} \right] \overset{(1)}{=} 2 \left[l(y_1, \dots, y_n) - l\left(\hat{\mathbf{b}}\right) \right] \\ &\overset{(2)}{=} 2 \left\{ \sum_{i=1}^{n} [y_i \ln y_i - y_i] - \sum_{i=1}^{n} \ln y_i! - \sum_{i=1}^{n} \left[y_i \ln \exp(\mathbf{x}_i^\top \hat{\mathbf{b}}) - \exp(\mathbf{x}_i^\top \hat{\mathbf{b}}) \right] + \sum_{i=1}^{n} \ln y_i! \right\} \\ &= \dots \overset{(3)}{=} 2 \sum_{i=1}^{n} \left[y_i \ln \frac{y_i}{\hat{m}_i} - (y_i - \hat{m}_i) \right] \end{aligned} \tag{12.6}$$

where

- in (1) we transform in twice the difference of the loglikelihoods
- in (2) we write extendedly the two parts (loglik of saturated model in the first line, of the fitted model in the second)
- in (3) we substituted $\hat{m}_i = \exp(\mathbf{x}_i^\top \hat{\mathbf{b}})$ as the fitted conditional expected value for the i-th unit

The structure of the deviance in the final equation is slighty different from the gaussian model (which depend on choice of probabilistic component): its always the sum of individual contribution, each measuring the distance between observed counts y_i and fitted value/estimated expected counts $\hat{m}_i$. The distance here is measured using a metric that exploits not only the difference but also the ratio (by construction the denominator having an exponential will be never 0) of the twos.
Here differently from the gaussian case where the deviance included the unknown nuisance parameter σ^2, here without nuisance parameter (in the conditional distribution) this deviance can be computed once fitted the model just by plugging the ML estimates.
We here can exploit this deviance to build a proper goodness of fit test: we set a threshold for this deviance, if the obtained value is lower than the treshold we can consider the fitted model as good as/not far from the saturated/best possible model.
If the deviance exceeds this threshold we can conclude that the fitted model/systematic component we choosed, is not adequate/too far from the benchmark to consider it adequate.

12.3.1.4 An approximation to D

Remark 118. Before looking at the test is worth having a closer look to the D expression, because with a bit of effort, we can come up with a simplified expression to compute the deviance which is more familiar.

In the equation of D if we consider the *second* order Taylor series expansion of $y_i \ln \frac{y_i}{\hat{m}_i}$ (we think it as a function of y_i, $f(y_i)$) at $\hat{m}_i$ we have:

$$f(y_i) = y_i \ln \frac{y_i}{\hat{m}_i} \approxeq f(\hat{m}_i) + \left.\frac{\partial}{\partial y_i} f(y_i)\right|_{y_i=\hat{m}_i} \cdot (y_i - \hat{m}_i) + \frac{1}{2} \left.\frac{\partial^2}{\partial^2 y_i} f(y_i)\right|_{y_i=\hat{m}_i} \cdot (y_i - \hat{m}_i)^2$$
$$= \hat{m}_i \ln \frac{\hat{m}_i}{\hat{m}_i} + \left.\frac{\partial}{\partial y_i} y_i \ln \frac{y_i}{\hat{m}_i}\right|_{y_i=\hat{m}_i} (y_i - \hat{m}_i) + \frac{1}{2} \left.\frac{\partial^2}{\partial y_i^2} y_i \ln \frac{y_i}{\hat{m}_i}\right|_{y_i=\hat{m}_i} (y_i - \hat{m}_i)^2$$

Let's develop the component of the expression; for the first partial derivative we have that

$$\begin{aligned}
\frac{\partial}{\partial y_i} y_i \ln \frac{y_i}{\hat{m}_i} &= \frac{\partial}{\partial y_i} y_i \ln y_i - \frac{\partial}{\partial y_i} y_i \ln \hat{m}_i \\
&= y_i \cdot \frac{1}{y_i} + 1 \cdot \ln y_i - \ln \hat{m}_i \\
&= 1 + \ln y_i - \ln \hat{m}_i \\
\left.\frac{\partial}{\partial y_i} y_i \ln \frac{y_i}{\hat{m}_i}\right|_{y_i=\hat{m}_i} &= 1 + \ln \hat{m}_i - \ln \hat{m}_i = 1
\end{aligned}$$

For the second partial derivative

$$\begin{aligned}
\frac{\partial^2}{\partial y_i^2} y_i \ln \frac{y_i}{\hat{m}_i} &= \frac{\partial}{\partial y_i} \left[1 + \ln y_i - \ln \hat{m}_i\right] = \frac{1}{y_i} \\
\left.\frac{\partial^2}{\partial y_i^2} y_i \ln \frac{y_i}{\hat{m}_i}\right|_{y_i=\hat{m}_i} &= \frac{1}{\hat{m}_i}
\end{aligned}$$

So to wrap up our approximation become

$$y_i \ln \frac{y_i}{\hat{m}_i} \approxeq \underbrace{\hat{m}_i \ln \frac{\hat{m}_i}{\hat{m}_i}}_{0} + (y_i - \hat{m}_i) + \frac{1}{2} \frac{(y_i - \hat{m}_i)^2}{\hat{m}_i}$$

12.3.1.5 Pearson χ^2 statistics

If we plug these approximated expression in D:

$$\begin{aligned}
D &= 2 \sum_{i=1}^{n} \left[y_i \ln \frac{y_i}{\hat{m}_i} - (y_i - \hat{m}_i) \right] \\
&\approxeq 2 \sum_{i=1}^{n} \left[(y_i - \hat{m}_i) + \frac{1}{2} \frac{(y_i - \hat{m}_i)^2}{\hat{m}_i} - (y_i - \hat{m}_i) \right] \\
&= \sum_{i=1}^{n} \frac{(y_i - \hat{m}_i)^2}{\hat{m}_i} = \chi^2
\end{aligned}$$

If we look at the final expression we recognize the Pearson χ^2 statistics (used to compare frequency tables): it's a comparison of the observed counts minus the expected counts, to the square, divided by the expected cunts.

$$D \approxeq \sum_i \frac{(O_i - E_i)^2}{E_i}$$

Remark 119 (Historical fact). Pearson derived the χ^2 statistics as an approximation of the likelihood ratio test statistic

Here the deviance is nothing more than a lrt statistic comparing observed values (from the saturated model) with the fitted one (obtained by the considered model)

By introducing the approximation it is clearer that basically we're comparing the observed counts with what we get from the fitted model (expected counts): the closer the expected and observed, the smallest the deviance.

A model which can reproduce perfectly the observed value will have zero deviance; in practice this will never happen (counts are integer, expected counts are exponential of real quantity) but the closest the two, the less our statistics

A further interpretation of this pearson χ^2 statistics relies on the fact that for poisson regression model variances and expected value coincides; so we can interpret the approximate D as

$$D \approxeq \sum_i \frac{y_i - \widehat{\mathbb{E}[Y_i|X_i]}^2}{\widehat{\text{Var}[Y_i|X_i]}}$$

By considering this alternative interpretation we recognize something similar to the gaussian deviance which should be something similar to $\frac{y_i - \widehat{\mathbb{E}[Y_i|X_i]}^2}{\sigma^2}$ (with the difference that here we don't have a constant variance, being equal to expected value)

Remark 120. In the end how can we use this two quantity (deviance or its approximation with pearson χ^2 statistic) to judge the adequacy of the fitted model? We can implement a goodness of fit test

Remark 121. In the following test we focus on the systematic component. It's not adequate to choose wether the poisson distribution is adequate to describe the data, but can establish whether the systematic component choosen is adequate to describe the data (eg we selected a proper subset of regressor to explain the variation of conditional expected value to the unit).

12.3.1.6 Goodness of fit test

If the systematic component of a Poisson regression model is adequate (*correctly specified*), or, equivalently, if the null hypothesis

$$\begin{cases} Y_i|\mathbf{x}_i \sim \text{Poi}(\mu_i) \text{independent} i = 1, \dots, n \\ H_0 : \ln \mu_i = \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} \ (+ \ln w_i, \text{ in case of exposure level/offset}) \end{cases}$$

or equivalently $H_0 : \mu_i = (w_i) \exp(\eta_i)$ is true, then it can be proved that:

- *the two statistics are asymptotically equivalent*: $\chi^2 \xrightarrow{d} D$;
- D and X^2 are both *asymptotically independent* from the ML estimator $\hat{\mathbf{B}}$
- the conditional distribution of both under null will converge to a χ^2 with $n - p - 1$ df: $D|\, H_0 \xrightarrow{d} \chi^2_{n-p-1}$ and $X^2 \big|\, H_0 \xrightarrow{d} \chi^2_{n-p-1}$

Important remark 102 (On asymptotic in poisson models). Note that differently from asymptotic results previously described (asymptotic meaning $n \to \infty$), those related to D and X^2 hold for n fixed, and $w_i \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right) \longrightarrow \infty \forall i$.
This means that this this goodness of fit test:

- can be performed irrespective of sample size (n can be large or small);
- *can be used only when the estimated counts* $\hat{m}_i = w_i \exp\left(\mathbf{x}_i^\top \hat{\mathbf{b}}\right)$ *(or* w_i*) are "reasonably" large*

Remark 122. This last condition reminds the shit studied with pearson chi square statistics with cell frequencies; we use the chi square distribution there to compare observed and expected counts only if the expected count of the cell is reasonably large (otherwise use fisher exact test) exact test).

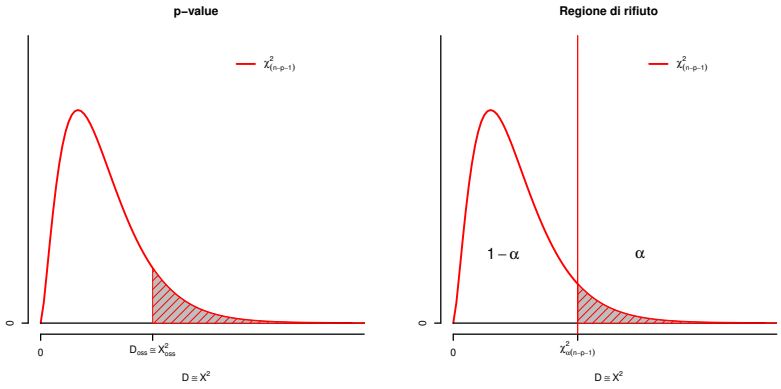
Important remark 103 (How does the test works?). Suppose we're in a situation where we can apply this goodness of fit test (D or χ^2); once obtained the observed value for the test statistic, knowing that the statistic is distributed according to a χ^2 with $(n - p - 1)$ df, we compute the p-value p associated to the statistic and for a given significance level α:

- if $p < \alpha \iff D_{oss} \approxeq X^2_{oss} > \chi^2_{\alpha(n-p-1)}$ *implies* H_0 is rejected (the value of the deviance is too far from zero) and we conclude that *the model is not adequate*;
- otherwise if $p > \alpha \iff D_{oss} \approxeq X^2_{oss} < \chi^2_{\alpha(n-p-1)} \implies H_0$ is not rejected and we conclude that *the model is adequate*

fig 12.2

Remark 123. The practical meaning of rejection or non rejection of null hypothesis in the context of goodness of fit test is different from pov:

- if we reject the null the model that we have must be discarded (so this is a conclusive decision);



(a) (b)

Figure 12.2: Goodness of fit test

- if we don't reject the null we can conclude that is adequate
  However we *don't have the guarantee* that that model is the only adequate model or the data we're dealing with (so this is *not a conclusive decision*). There might be other models that are adequate as well.

Remark 124. The possibility of performing a goodness of fit test is a plus compared to the standard gaussian model: if the model is not adequate we're interested in investigating the causes of non adequacy.
This can be done by exploring the residuals for our model, however with some difficulties

12.3.2 Residuals

Important remark 104. Differently for the Gaussian case, where are defined as difference between observed and fitted value, in GLM *there is not a straightforward definition of residual* because *it is not possible to exploit an additive structure (systematic component + error term).*

Remark 125. Different definitions of residual are present in the statistical literature; the most used residuals, on which we focus, are:

- deviance residuals;
- Pearson residuals.

Other GLM residuals are: Ascombe residuals, pseudo-residuals, etc.

12.3.2.1 Residuals for Poisson regression models

Definition 12.3.1 (Deviance residuals)**.** They are defined as *the squared root (with sign) of the doubled individual contribution of the i-th unit to the deviance (the generic term of the deviance)*:

$$e_i^D = \text{sign}(y_i - \hat{m}_i)\sqrt{2\left[y_i \ln \frac{y_i}{\hat{m}_i} - (y_i - \hat{m}_i)\right]}, \quad i = 1, \dots, n$$

where:

- the term within parenthesis can be proved to be positive;
- sign is defined as

$$\text{sign}(y_i - \hat{m}_i) = \begin{cases} -1 & \text{if} y_i < \hat{m}_i \\ \\ +1 & \text{if} y_i \geq \hat{m}_i \end{cases}$$

- they are called deviance residuals because if squared and summed return the deviance

$$\sum_{i=1}^{n}\left(e_i^D\right)^2 = D$$

Definition 12.3.2 (Pearson residuals)**.** Defined as the squared root (with sign) of the generic term of the Pearson χ^2 statistic:

$$e_i^P = \frac{y_i - \hat{m}_i}{\sqrt{\hat{m}_i}}, \quad i = 1, \ldots, n$$

The name comes from the fact that squared and summed returns the Pearson χ^2 statistic:

$$\sum_{i=1}^{n}\left(e_i^P\right)^2 = X^2$$

Remark 126. It's trivial to show that for gaussian regression model that deviance and χ^2 statistics coincides while in GLM χ^2 statistics is an approximation of the deviance D; in one way we can think of Pearson residuals as approximation of the deviance residuals (more or less similar values).

12.3.2.2 Properties and graphical analysis of the residuals

Important remark 105 (Properties of residuals). If a Poisson regression model is adequate (correctly specified) we would expect to have

- both deviance and Pearson residuals have null expected value (so we expect residual to be spread around 0):

$$\mathrm{E}\left[e_i^D \middle| \mathbf{x}_i\right] \approxeq \mathrm{E}\left[e_i^P \middle| \mathbf{x}_i\right] \approxeq 0$$

- residuals are characterized by heteroskedasticity and the conditional variance of residuals is

$$\mathrm{Var}\left[e_i^D \middle| \mathbf{x}_i\right] \approxeq \mathrm{Var}\left[e_i^P \middle| \mathbf{x}_i\right] \approxeq 1 - H_{ii}$$

  where H_{ii} is the l'i-th element of the main diagonal of the matrix:

$$\mathbf{H} = \mathbf{W}^{\frac{1}{2}}\mathbf{X}\left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{W}^{\frac{1}{2}}$$

  which resembles the hat matrix for gaussian models: the only difference is that we have here the $\mathbf{W}$ matrix (we have to take into account we have heteroskedasticity and variances of residuals are characterized by different conditional variances)

- *asymptotically*, both deviance and Pearson residuals are equivalent and they are distributed as independent Gaussian random variables: so more or less the behaviour of the plot should be the same as the gaussian model's one (when our model is correctly specified).
  Again here note that as for D and X^2, the *asymptotic properties of deviance and Pearson residuals hold for* fixed n and expected count sufficiently large $w_i \exp\left(\mathbf{x}_i^\top\boldsymbol{\beta}\right) \longrightarrow \infty, \forall i$

Remark 127. If

- Goodnes of fit (GOF) test leads to a non rejection of the model by inspecting the residuals we would expect getting the previous behaviour (in principle there would not the need to look at the residuals given that the test testify the appropriateness of the model);
- GOF test leads to rejection of the models, by inspecting the residuals we should find something different from what depicted above

Important remark 106 (Graphical analysis of residuals). If a Poisson regression model is not adequate, according to a goodness of fit test, the graphical analysis of residuals can help in finding the misspecification source. In particular, misspecification errors in the systematic component can be detected by examining the plots of both:

- e_i^P vs. $\mathbf{x}_i^\top\hat{\mathbf{b}}$

- e_i^P vs. $\hat{m}_i = w_i \exp\left(\mathbf{x}_i^\top \hat{\mathbf{b}}\right)$

Important remark 107 (Detecting the source of missspecification). Due to the fact that we have a nonlinear model involving link function and linear prediction, inadequacy of systematic component can be dued to:

- misspecification with link function (we should therefore replace exponential with something else)
- a misspecification in the linear predictor (eg we omitted some relevant regressors, or the included regressors are not well described say by simple linear impact/term and we need to focus on nonlinear contributions)
- a combination of both

In Poisson and GLM generally it's very difficult to understand what's the nature of missspecification simply by looking at the plot of the residuals.
But with a bit of training one can distinguish among the above sources of inadequacy.

12.3.3 Comparisons among Poisson regression models

Remark 128. The advantage of having a Goodness of fit test is that we can rule out some models; however there could be other appropriate models, having different systematic component/linear predictor, so we need tools to compare them.
We'll focus on covariates: we won't tackle change in the link function we would end up with models that are not Poisson regression model (which requires exp link)

12.3.3.1 Choice among Poisson regression models

What to do if we have more than 1 candidate model passing the GOF test? Which is the most adequate Poisson regression model for a given random sample $\mathbf{Y}$, with $Y_i|\mathbf{x}_i \sim \text{Poi}(\mu_i)$ and independent $(i = 1, \dots, n)$?

In the simplest situation we assume *two candidate models* characterised by *different sets of regressors*: $\mathbf{x}_{Ai} \neq \mathbf{x}_{Bi}, \forall i$:

$$M_A : \ln \mu_i = \eta_{Ai} = \mathbf{x}_{Ai}^\top \boldsymbol{\beta}_A (+\ln w_i), \quad \text{with } \boldsymbol{\beta}_A \text{ a } (p_A + 1) \text{ dimensional vector}$$

$$M_B : \ln \mu_i = \eta_{Bi} = \mathbf{x}_{Bi}^\top \boldsymbol{\beta}_B (+\ln w_i), \quad \text{with } \boldsymbol{\beta}_B \text{ a } (p_B + 1) \text{ dimensional vector}$$

Without loss of generality we assume model A is characterized by a larger number of *parameters* $p_A > p_B$ (not necessarily a larger number of *regressors*). However *in the following for simplicity we focus on the case of numerical regressors* where number of parameters and of regressors coincides.
We can incur basically in two situation: the models are nested or not.

Nested models This occurs when vectors $\mathbf{x}_{Bi}$ can be obtained by removing one or more regressors from vectors $\mathbf{x}_{Ai}$; thus M_B can be described/obtained by introducing suitable linear constraints on the parameteres of M_A (setting them to 0), that is

$$\left.\begin{array}{l} M_A : \ln \mu_i = \eta_{Ai} = \mathbf{x}_{Ai}^\top \boldsymbol{\beta}_A (+\ln w_i) \\ \\ H_0 : \mathbf{K}_B \boldsymbol{\beta}_A = \mathbf{t}_B \end{array}\right\} \Rightarrow M_B : \ln \mu_i = \eta_{Bi} = \mathbf{x}_{Bi}^\top \boldsymbol{\beta}_B (+\ln w_i)$$

where

- $q = p_A - p_B$ is the number of regressors excluded from M_A to obtain M_B
- $\mathbf{K}_B$ is a $(q) \times (p_A + 1)$ matrix with a number of rows corresponding to the number of regressors to be removed from M_A to obtain M_b; each row contains a 1 in a specific position (corresponding to one of the q regressors excluded from M_A), and 0 elsewhere
- $\mathbf{t}_B = \mathbf{0}_q$

So in this setup we can recast the choice between two models in testing a system of linear hypotheses. A likelihood ratio test can be exploited to choose among M_A and M_B like in the gaussian case. In particular, such test can be expressed as a function (the difference) of the two corresponding deviances (by adding and subtracting the likelihood of the saturated model):

$$\Delta l = 2 \ln \left[\frac{L\left(\hat{\mathbf{b}}_A\right)}{L\left(\hat{\mathbf{b}}_{A|H_0}\right)} \right] = 2 \ln \left[\frac{L\left(\hat{\mathbf{b}}_A\right)}{L\left(\hat{\mathbf{b}}_B\right)} \right] = D(M_B) - D(M_A) = \Delta D$$

where $L\left(\hat{\mathbf{b}}_A\right)$ is the likelihood of the unconstrained model (M_A), while $L\left(\hat{\mathbf{b}}_{A|H_0}\right)$ is the likelihood of the constrained model (which will coincide with the likelihood of model M_b).
We expect the deviance of M_B to be larger (after removing covariates we're loosing flexibility and the resulting model is expected to be farther from the saturated model) and thus the delta being positive. So the test is an estimate on how much the deviance increases by removing those regressors.
If M_B is as "adequate" as M_A and we're removing irrelevant regressors we might expect the increase in the deviance could be small/neglgible; otoh more important variables exclusion may provoke a large increase in the deviance.
If H_0 is true (that is if M_B is as "adequate" as M_A) then the test is asymptotically distributed like a chi-squared with q df (number of omitted regressors):

$$\Delta D|\, M_B \xrightarrow{d} \chi^2_q$$

In this case, this asymptotic results holds in the classic manner that is if $n \to \infty$.

Remark 129. If we remember, when dealing with LRT statistic in gaussian model, we've seen the 2 log LRT could be written using an expression involving only estimates for the unrestricted model, the matrix $\mathbf{K}$ and the vector $\mathbf{b}$

$$\frac{(\mathbf{Kb}-\mathbf{t})^\top \left[\mathbf{K}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{K}^\top\right]^{-1}(\mathbf{Kb}-\mathbf{t})}{\sigma^2} \tag{12.7}$$

with $(\mathbf{X}^\top\mathbf{X})^{-1}/\sigma^2$ the inverse of the expected Fisher information matrix.
Something similar happen here as well

It can be proved that:

- the difference in deviances ΔD is *asymptotically* equivalent to this expression:

  $$\left[\mathbf{K}_B\hat{\mathbf{b}}_A-\mathbf{t}_B\right]^\top\left[\mathbf{K}_B\widehat{I(\boldsymbol{\beta}_A)}^{-1}\mathbf{K}_B^\top\right]^{-1}\left[\mathbf{K}_B\hat{\mathbf{b}}_A-\mathbf{t}_B\right]$$

  we have more or less the same structure as in 12.7.
  So even for poisson we can actually test the hypothesis without fitting the model under the restriction, but just using things from the model without restrictions, and by applying the formula above.
  In the contex of gaussian model there's an exact coincidence while here (and for any GLM) is only asymptotical
- so having asymptotic equivalence we can apply this simpler formula to perform a test, considered that it will be again χ^2, with q df

  $$\left[\mathbf{K}_B\hat{\mathbf{b}}_A-\mathbf{t}_B\right]^\top\left[\mathbf{K}_B\widehat{I(\boldsymbol{\beta}_A)}^{-1}\mathbf{K}_B^\top\right]^{-1}\left[\mathbf{K}_B\hat{\mathbf{b}}_A-\mathbf{t}_B\right]\Big|\, M_B \xrightarrow{d} \chi^2_q$$

Non-nested models In case vectors $\mathbf{x}_{Bi}$ cannot be obtained by removing one or more that one regressor from vectors $\mathbf{x}_{Ai}$

- Model M_B can be obtained by simultaneously excluding some (or all) regressors in model M_A and adding some regressors to model M_A
- thus *the two models are characterised by two sets of regressors that are only partially overlapping, or non-overlapping*
- The differences between the two deviances does not have a known random distribution, and thus a likelihood ratio test cannot be used to choose between the two models

In this cases we rely on AIC and BIC, that for Poisson regression models are respectively:

$$\begin{aligned}
AIC &= -2\ln L\left(\hat{\mathbf{b}}\right)+2(p+1)\\
&= -2\sum_{i=1}^{n}\left[y_i(\mathbf{x}_i^\top\hat{\mathbf{b}})-\exp(\mathbf{x}_i^\top\hat{\mathbf{b}})\right]+2\sum_{i=1}^{n}\ln y_i!+2(p+1)\\
BIC &= -2\ln L\left(\hat{\mathbf{b}}\right)+\ln(n)\cdot(p+1)\\
&= -2\sum_{i=1}^{n}\left[y_i(\mathbf{x}_i^\top\hat{\mathbf{b}})-\exp(\mathbf{x}_i^\top\hat{\mathbf{b}})\right]+2\sum_{i=1}^{n}\ln y_i!+\ln(n)\cdot(p+1)
\end{aligned}$$

Again lowest AIC/BIC are for the models with best compromise between fit/adequacy and complexity.
Note that *the additive constant* $2\sum_{i=1}^{n}\ln y_i!$ *can be ignored when all competing models have a Poisson probabilistic component.*

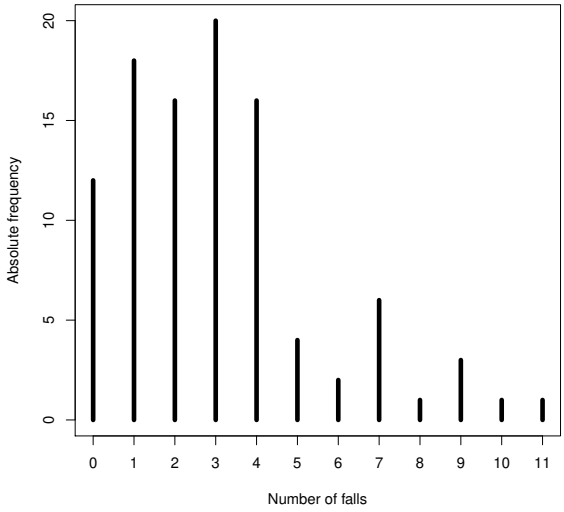


Figure 12.3: Number of falls - observed values

12.4 Poisson regression example

12.4.0.1 Introduction

A researcher in geriatrics designed a randomized prospective study to investigate the effects of an *aerobic exercise training* (treatment) on the *frequency of falls* (outcome) with 100 subject (at least 65 years old and in reasonably good health); gender, values of a balance index, strenght index were also registered (the higher the balance index, the more stable the subject, and the higher the strength index, the stronger subject). After 6 months from randomization, **observed falls** are represented in figure 12.3

12.4.0.2 First model estimation

We have:

- with standard coded dummy (`train`: `NO` = `0` for the control groups, `YES` = `1` for experimental group; `gender`: `FEMALE` = `0`, `MALE` = `1`) the **model estimates** are reported below

```
Coefficients:
              Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)     0.489         0.337     1.453      0.146
 trainYES       -1.069         0.133    -8.031      0.000
 genderMALE     -0.047         0.120    -0.388      0.698
 bal             0.009         0.003     3.207      0.001
 str             0.009         0.004     1.986      0.047
   Null deviance:  199.19 on 99 degrees of freedom

Residual deviance:  108.79 on 95 degrees of freedom
AIC: 377.29
```

  In the table we have

  - the estimates
  - the *asymptotic* standard error (ML are asymptotically efficient and their varcov is the inverse of the expected fisher information matrix; so here the standard

error are the roots of diagonal elements of $(\mathbf{X}^\top \hat{\mathbf{W}} \mathbf{X})^{-1}$, the inverse of expected fisher information matrix evaluated at the ML estimates),

- the z values (again asymptotic): since ML estimators are asymptotically gaussian, in order to test the null $H_0 : \beta_j = 0$ we use the asymptotic std error and just calculate z by estimate / standard error. It plays the same role of the t test statistics in the gaussian regression model; there the distribution for the t test statistics is known whichever the sample size is, while here we have an asymptotic test for when sample size goes to infinity (incidentally with a t distribution and let degrees of freedom go to infinity with sample size, one will approach the standard gaussian distribution, so there's coherence between these two results)
- the (asymptotic) p values associated with two tailed $H_0 : \beta_j = 0$ using a gaussian distribution

The second part contains some summary measures related to the model (null and residual deviance, while we talked about one single deviance): these are measures used to choose among two models:

- the null deviance is the deviance for the model containing only the intercept (the null model $\mu_i = \exp(\beta_0)$, producing a costant expected value, the model where we assume that no dependent variable has an impact on the dependent variable)
- the residual deviance is the residual of the actual model we've fitted: obtained by plugging in the formula of the deviance the ML estimates we've obtained

The close the deviance to 0, the better. Here we see that the null deviance is higher than the residual deviance: it can be proved that (as in gaussian regression for RSS decrease), if we include regressors in the model we'll have an increase in the loglikelihood, meaning a decrease in the deviance.
Finally we have the AIC: still the lower the better but differently from deviance (for which is 0) we don't have a lower bound and is meaningful only in comparison between models

- for the goodness of fit tests we have that if the model bla bla bla, the deviance is chi square distributed. So having 100 units with 5 parameters (4 regressor coefficient + 1 intercept) so the degrees of freedom are 95:
  - focusing on the model with all the regressor, having deviance $D = 108.790$, we have that $\Pr\left(\chi^2_{(95)} \geq 108.79\right) = 0.158$
  - the value of pearson $\chi^2 = 105.547$ (a sort of asymptotic approximation for the deviance, there are some differences but are relative small 3 point on over 105) and the p-value $\Pr\left(\chi^2_{(95)} \geq 105.547\right) = 0.216$

  In both cases the null is not discarded and so *the model can be considered adequate* (actually the rejection value/area for a χ^2 with 95 degrees of freedom is 118, figure 12.4): we can say that deviance and chi square statistics *is not significantly larger than 0*.
  This kind of test is very useful (not available in standard gaussian model due to the fact that in the poisson distribution we don't have a nuisance parameter) but should be used only if requirements are respected. Why the use of these test is still questionable here?: we've not check the actual conditions regarding (large value for all the estimated expected counts: if we look at figure 12.3 most of the units are characterized by small counts so in this case the use of test statistics could be questionable)
- aside from the fact that the model can be considered adequate, the plot of residuals (fig 12.5) no evident patterns emerge from the plots ((deviance) residuals vs fitted lo stesso plot si dovrebbe avere anche coi pearson residuals, qqplot), consistently with the result of the goodness of fit test:
  - on the left hand side the residuals are almost evenly spread around zero for any predicted value.
    Something a little bit strange: it seems there are "lines" of residuals there are actually 12 lines/curves of residuals.
    The dependent variable is discrete, the value that we can observe are $0, \ldots, 11$: the lower curve is composed of residuals having dependent variable $y_i = 0$, the

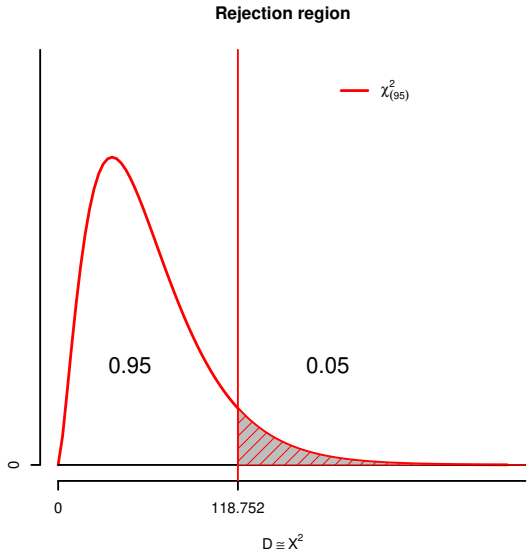


Figure 12.4: Goodness of fit test

second lower is from unit having $y_i = 1$ and so on. We see the lower the observed value the more separated is the corresponding line (the line for $y_i = 0$ is very far from the other) while the residual lines for higher observed value are closer and closer.
When we have dataset with large counts all this lines will become closer and closer and at some point will become indistinguishable: this is why the use of goodness of fit test is questionable. We still recognize a sort of pattern which is not the usual pattern one would expect from residual from a model adequate for the data.
In model with discrete valued outcome, when looking at residual vs fitted plot if one can easily recognize lines of residuals, we're in a situation in which the asymptotics condition needed to use the goodness of fit test is questionable; this sort of behaviour can happen with binary outcomes as well.

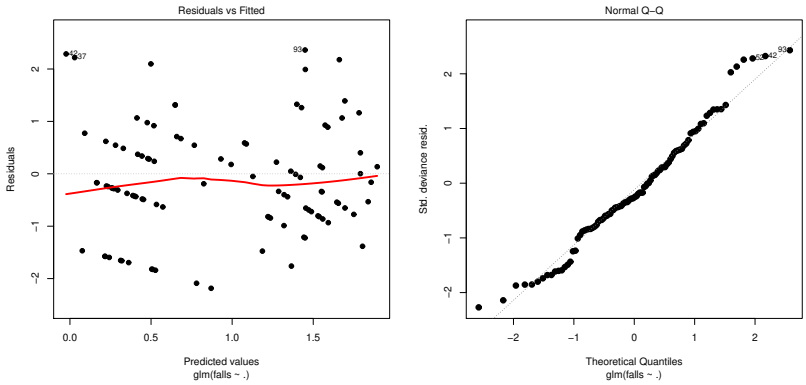
– in the qqplot the gaussian distribution seems reasonable being the dot near the line

If on the other hand we would have fit a gaussian regression model to the data, things would have gone worst in term of residuals (fig 12.6): these plots highlight the inadequacy of the Gaussianity (look qqplot) and homoschedasticity assumption (in higher fitted values display larger residual variability). We furthermore still note the "set of lines" plot style of residuals: here the line are not curved since gaussian model assumes linearity of the expected values in the regressor (identity link while for poisson it was exp).
So for sure the gaussian model is worst than the poisson model.

- for what concerns **interpretation of model parameters**/estimated effects:

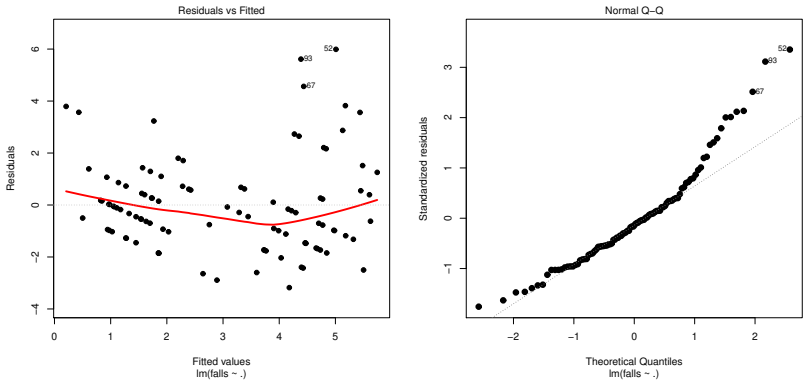
	$\hat{b}_j$	$\exp\left(\hat{b}_j\right)$
trainYES	-1.069	0.343
genderMALE	-0.047	0.954
bal	0.009	1.010
str	0.009	1.009

After exponentiation we have that:



(a) (b)

Figure 12.5: Poisson regression residuals



(a) (b)

Figure 12.6: Gaussian regression residuals

- the expected number of falls for individuals enrolled in the training program is approximately one third of the expected number of falls for untrained individuals, for given gender, balance index and strength index;
- the expected number of falls for males is approximately 5 percent lower than the expected number of falls for female, after controlling for all the other regressors in the model (btw this coefficient was not significant);
- the expected number of falls increases of about 1 percent for each additional point in the balance index, holding fixed the values for the other regressors in the model;
- the expected number of falls increases of about 1 percent for each additional point in the strength index, holding fixed the values for the other regressors in the model.

Regarding the last two, maybe one feeling more strong/balanced acts more confidently and in the end less securely: maybe these subject are more exposed

- for what concerns hypothesis testing,
  - to test the overall independence hypothesis

$$H_0 : \beta_{\texttt{train}} = \beta_{\texttt{gender}} = \beta_{\texttt{balance}} = \beta_{\texttt{strength}} = 0$$

we implement the LRT comparing the full model with the reduced/null model. For the full model we have:

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.489	0.337	1.453	0.146
trainYES	-1.069	0.133	-8.031	0.000
genderMALE	-0.047	0.120	-0.388	0.698
bal	0.009	0.003	3.207	0.001
str	0.009	0.004	1.986	0.047

```
   Null deviance:  199.19 on 99 degrees of freedom

Residual deviance:  108.79 on 95 degrees of freedom
AIC: 377.29
```

For the reduced/null model containing only the intercept (where null and residual deviance coincides and AIC increases)

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.112	0.057	19.386	0.000

```
   Null deviance:  199.19 on 99 degrees of freedom

Residual deviance:  199.19 on 99 degrees of freedom
AIC: 459.69
```

Finally the likelihood ratio test gives first the residual deviance for the reduced and the full model:

Model	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
falls~1	99	199.19			
falls~train+gender+bal+str	95	108.79	4	90.40	0.0000

The deviance/statistics is just the difference between the twos:

$$2 \ln \frac{L(F)}{L(R)} = -2 \ln \left[L(R) - L(F) \right] = 199.19 - 108.79 = 90.40$$

the degrees of freedom are given by the difference in number of parameters (4 coefficients set equal to 0).
The asymptotic p-value computed using the χ^2 distribution is statistically significant and the null rejected so with LRT we can conclude that at least one of the four regressors is significantly associated with the number of falls.
Similarly we could apply the asymptotic equivalent **Wald test** $(\mathbf{Kb} - \mathbf{t})^\top (\mathbf{K} I(\hat{\mathbf{b}})^{-1} \mathbf{K})^{-1} (\mathbf{Kb} - \mathbf{t})^\top$, which produce a different value, 80, but leads to the same conclusion:

```
Hypothesis:
trainYES = 0
genderMALE = 0
balance = 0
strength = 0


Model 1:  restricted model
Model 2:  falls ~train + gender + balance + strength


    Res.Df      Df    Chisq   Pr(>Chisq)
 1  99.000
 2  95.000   4.000   80.250         0.000
```

When sample size is finite/low and LRT does not coincide with Wald:

* Wald is an approximation of LRT
* to compute Wald we do not need to compute the restricted model

All in all use LRT in simple cases , or Wald test statistics where the reduced model is depending on more complex restrictions.
Finally the asymptotics conditions to use both LRT and Wald to test hypotheses on regression coefficient is to both have a correct model and large sample size

– to test a single coefficient $H_0 : \beta_j = 0$, let's test the efficacy of the training scheme that is $H_0 : \beta_{\texttt{train}} = 0$.
We can look at the asymptotic p-value (that is basically the Wald test statistic associated to the coefficient) found in the table of the full model, which is clearly significant:

```
Coefficients:
               Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)      0.489         0.337     1.453      0.146
 trainYES        -1.069         0.133    -8.031      0.000
 genderMALE      -0.047         0.120    -0.388      0.698
 bal              0.009         0.003     3.207      0.001
 str              0.009         0.004     1.986      0.047
   Null deviance:  199.19 on 99 degrees of freedom

Residual deviance:  108.79 on 95 degrees of freedom
AIC: 377.29
```

We can compute the LRT by first computing the reduced model where we removed the train dummy variable (leading to an increased deviance and AIC):

```
Coefficients:
               Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)      0.366         0.322     1.138      0.255
 genderMALE      -0.228         0.117    -1.944      0.052
 bal              0.009         0.003     2.979      0.003
 str              0.006         0.004     1.466      0.143
   Null deviance:  199.19 on 99 degrees of freedom

Residual deviance:  182.31 on 96 degrees of freedom
AIC: 448.81
```

The likelihood ratio test is as follows:

```
 Model                        Resid.  Df   Resid.  Dev   Df   Deviance   Pr(>Chi)
 falls~gender+bal+str                 96        182.31
 falls~train+gender+bal+str           95        108.79    1      73.52     0.0000
```

Calculations were as follows

$$2\ln\frac{L(F)}{L(R)} = -2\ln\left[L(R) - L(F)\right] = 182.31 - 108.79 = 73.52$$

$$\cong \left(\frac{\hat{\beta}^2_{\texttt{train}}}{s^2\left[\hat{\beta}_{\texttt{train}}\right]}\right) = \frac{-1.069^2}{0.133^2} = 64.502$$

note that approximately the LRT statistic of 73 is close to the square of the z test statistics, which is 64

So, to conclude any test suggest that the aerobic training significantly reduced the average number of falls, after controlling for all the other subject characteristics.

– finally just by quickly looking at wald tests there's no effect of gender, there's effect of balance index and finally the effect of strength seems to be questionable/borderline

12.5 Other count models

Remark 130. Poisson model is one model for count data; there are others (some belongs to GLM family, other not) that are devoted to relax some limitation of the Poisson, some of which are:

- **overdispersion**: in poisson we have equivalence between variance and expected value but in practical situation often the conditional variance is larger that the conditional expected value
- **excess of zeros**: in many practical situation one may have lot of unit having zero as y_i; in the poisson distribution the amount of zero depends on the shape of the poisson distributio (λ) but sometimes the actual number of zero is larger than any possible poisson distribution
- **truncated counts**: it's a minor issue but happen when the minimum value for counts is not 0 but actually an higher value (eg the minimum count is 1 or 2); thing is that in the poisson distribution there's always a minimum mass given to the excluded modalità (eg respectively 0 or 0 and 1)

In any book these model will be small variations of Poisson so good luck and study hard.

Chapter 13

Lab 3 - Poisson regression

13.1 Intro

The dataset is one of the example used for poisson regression on insurance car accident claims

```
library(car)

## Caricamento del pacchetto richiesto:  carData
##
## Caricamento pacchetto:  'car'
## Il seguente oggetto è mascherato da 'package:lbmisc':
##
##     recode

library(lbdatasets)

##
## Caricamento pacchetto:  'lbdatasets'
## Il seguente oggetto è mascherato da 'package:MASS':
##
##     anorexia

head(claims)

##     n  c age       dist   car
## 1 197 38 <25      rural    <1
## 2 284 63 <25      rural 1-1.5
## 3 133 19 <25      rural 1.5-2
## 4  24  4 <25      rural    >2
## 5  85 22 <25 small towns    <1
## 6 149 25 <25 small towns 1-1.5

## n: number of policyholders for each covariate pattern (that is, each data entry of the cross-table built on [age,d
## c: number of claims for each covariate pattern
## age: arranged in classes (<25, 25-29, 30-35, >35)
## dist: type of area where the policyholders live rural, small town, large towns, major cities
## car: engine capacity (<1 liter, 1-1.5 liter, 1.5-2 liter, >2 liter)
```

13.2 Estimation

We now estimate a generalized linear model for Poisson data with offset. `n` here is an exposure level: to introduce it as an offset variable (estimated coefficient imposed equal to 1 we use `offset(log(n)` in the formula terms). This is basically equivalent to divide each `c` (number

of claims) by the corresponding `n` (number of policy-holders): I guess `log(n)` since using the canonical link function we have mu = exp(bX con offset) and setting un b=1 col suo x = log(n) we can basically rewrite mu/n = exp(bx senza offset)

```
m1 <- glm(c ~ offset(log(n)) + car + dist + age,
          data = claims,
          family = poisson)
## Note: with "family" we both specify the conditional distribution
## and link function: see ?family or ?poisson for possible choices
## with glm. Each of them has the canonical link function as default
summary(m1)

##
## Call:
## glm(formula = c ~ offset(log(n)) + car + dist + age, family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46558  -0.50802  -0.03198   0.55555   1.94026
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.82174    0.07679 -23.724  < 2e-16 ***
## car1-1.5          0.16134    0.05053   3.193 0.001409 **
## car1.5-2          0.39281    0.05500   7.142 9.18e-13 ***
## car>2             0.56341    0.07232   7.791 6.65e-15 ***
## distsmall towns   0.02587    0.04302   0.601 0.547597
## distlarge towns   0.03852    0.05051   0.763 0.445657
## distmajor cities  0.23421    0.06167   3.798 0.000146 ***
## age25-29         -0.19101    0.08286  -2.305 0.021149 *
## age30-35         -0.34495    0.08137  -4.239 2.24e-05 ***
## age>35           -0.53667    0.06996  -7.672 1.70e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.26  on 63  degrees of freedom
## Residual deviance:  51.42  on 54  degrees of freedom
## AIC: 388.74
##
## Number of Fisher Scoring iterations: 4
```

The structure is what we've already seen. We see that for this model the dispersion parameter is set 1 (in poisson we don't have a nuisance parameter, dispersion parameter is the jargon for nuisance parameter), while the `glm` reports the number of the iterations in optimization. `glm` uses Fisher scoring (always, irrespective of the use of canonical link); the recursive formula was applied four time and at the fourth step the stopping criterion was met. A quick look shows that of the three dummy variable associated with area/`dists` only one seems to be significant; what happens if we change the reference category for one variable

```
## change the reference level when coding a qualitative regressor
## with ?contrasts and ?contr.treatment

## display current dummy creation
contrasts(claims$dist)

##              small towns large towns major cities
## rural                  0           0            0
## small towns            1           0            0
## large towns            0           1            0
## major cities           0           0            1
```

```
## set reference category: first argument is number of categories
## (here 4), second is the progressive id of the chosen comparison
## group (here the fourth, major cities)
contrasts(claims$dist) <- contr.treatment(4, base = 4)
contrasts(claims$dist)

##              1 2 3
## rural        1 0 0
## small towns  0 1 0
## large towns  0 0 1
## major cities 0 0 0

## here we can just update the model (not changing dataset or linear
## predictor or probabilistic component/link function)
m1_b <- update(m1)
summary(m1_b)

##
## Call:
## glm(formula = c ~ offset(log(n)) + car + dist + age, family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46558  -0.50802  -0.03198   0.55555   1.94026
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.58753    0.09276 -17.115  < 2e-16 ***
## car1-1.5     0.16134    0.05053   3.193 0.001409 **
## car1.5-2     0.39281    0.05500   7.142 9.18e-13 ***
## car>2        0.56341    0.07232   7.791 6.65e-15 ***
## dist1       -0.23421    0.06167  -3.798 0.000146 ***
## dist2       -0.20834    0.06476  -3.217 0.001294 **
## dist3       -0.19568    0.06984  -2.802 0.005081 **
## age25-29    -0.19101    0.08286  -2.305 0.021149 *
## age30-35    -0.34495    0.08137  -4.239 2.24e-05 ***
## age>35      -0.53667    0.06996  -7.672 1.70e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.26  on 63  degrees of freedom
## Residual deviance:  51.42  on 54  degrees of freedom
## AIC: 388.74
##
## Number of Fisher Scoring iterations: 4

## Deviances
summary(m1)$deviance

## [1] 51.42003

summary(m1_b)$deviance

## [1] 51.42003

summary(m1)$df.residual

## [1] 54
```

Updating the model yields changes only in the intercept and hte values for `dist` dummy variables: now all the three are significant but at the same time their coefficient are approximately 0.2. Other than that for other coefficients or summary measures the two models are identical (deviances, degrees of freedom too)

13.3 Model adequacy

13.3.1 Goodness of fit test

Deviance/Chi-square statistics has an asymptotic χ^2 distribution if we have expected counts that are sufficiently large. Here we're dealing with expected counts associated with different exposure level; one condition to be sure that expected counts are large enough is to have exposure level that are large enough. This because the expected counts will be given by the exposure levels times the estimated rates of occurrence

```
## the offset term should be "big enough" to ensure
## asymptotic properties of the deviance
sort(claims$n)

##  [1]    3    7    9   16   18   20   24   24   25   29   31   33   35   39   40
## [16]   43   48   53   66   68   71   72   73   78   81   85   89   99  114  121
## [31]  122  133  139  149  151  155  175  197  221  240  245  246  264  284  286
## [46]  313  316  322  344  355  419  452  536  648  692  696  724  931 1110 1635
## [61] 1640 1680 2443 3582
```

In this case, we see that apart for one covariate pattern, for the others there seems to be a sufficiently large exposure level (all of them must be large enough: are 3, 7 and 9 that? take test with a grain of salt)
However let's use the asymptotic chi-square distribution for the deviance (with $n - p - 1$ df) to calculate the asymptotic goodness of fit test p-value

```
## Test
pchisq(summary(m1)$deviance, summary(m1)$df.residual, lower.tail = FALSE)

## [1] 0.5745071
```

A large p-value non rejecting the null hypothesis: the systematic component we've chosen seems to be adequate to describe the observed counts (this does not mean that other models cannot as well).
We can check the Pearson chi square statistics by obtaining the residuals

for what concerns the **residuals** to extract them we use the omonym function on the glm (see `?residuals.glm`)

```
## Deviance residuals, the default: if we sum their squares we get the
## (residual) deviance displayed in the summary
resid_m1_dev <- residuals(m1, type = "deviance")
sum(resid_m1_dev^2)

## [1] 51.42003

## Pearson's residuals: by summing them we get the pearson chi square
## statistics, which is quite close to the deviance (the two
## quantities coincides only asymptotically) ...
resid_m1_p <- residuals(m1, type = "pearson")
sum(resid_m1_p^2)

## [1] 48.62934

## ... and can be used for doing the Pearson's test
pchisq(sum(resid_m1_p^2), summary(m1)$df.residual, lower.tail = FALSE)

## [1] 0.6809085
```

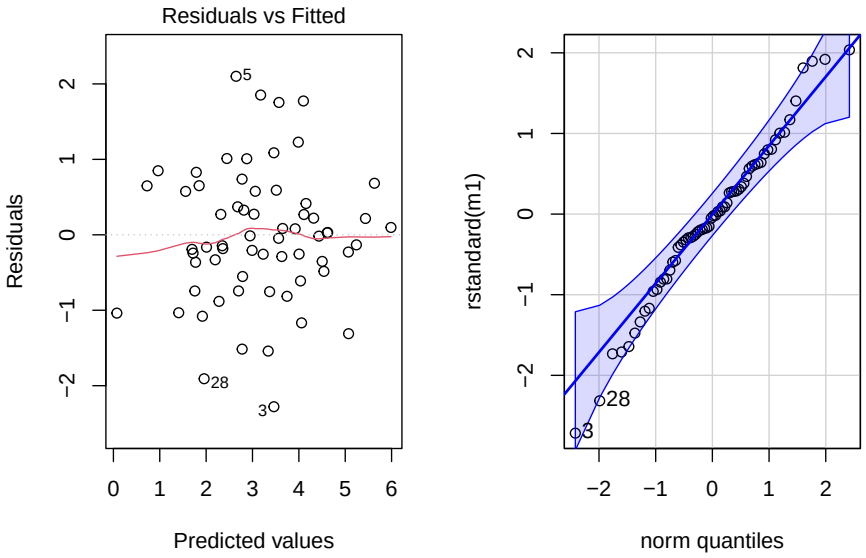


Figure 13.1: Residuals plots

Since pearson is slightly lower than the deviance, the p-val will be slightly larger. However for both of them, the model seems appropriate for the data at hand.

13.3.2 Residuals

To perform graphical inspection of residuals in fig 13.1: as expected from the GOF test residuals seems good. On the left are evenly scattered with no particular trend suggesting that the choice of the systematic component is adequate and the qqplot show behaviour quite close (points within bands) to the theoretical one.
Tipically when the test yields to rejection of null hypothesis we have some sort of strange behaviour/pattern in the residuals

```
### Graphical inspection of the residuals
par(mfrow = c(1,2))
## 1) linearity of the (log) Expected Value: (pearson) residual vs fitted
plot(m1, which = 1)
## 2) gaussianity of the standardized residuals
qqPlot(rstandard(m1))
```

```
## [1]  3 28
```

13.4 Hypothesis testing

13.4.1 Does the `dist` regressor impact the model?

We can check the hypothesis by comparing the model with and without the regressor using LRT; we do the things for the two fitted models (different for `dist` baseline category).

```r
## we remove dist from right hand side of the formula (other things equal)
m2 <- update(m1, . ~ . - dist)
summary(m2)

##
## Call:
## glm(formula = c ~ car + age + offset(log(n)), family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.61407  -0.59513  -0.07229   0.78529   2.71480
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.79527    0.07519 -23.877  < 2e-16 ***
## car1-1.5     0.16248    0.05052   3.216   0.0013 **
## car1.5-2     0.39466    0.05498   7.178 7.09e-13 ***
## car>2        0.56956    0.07227   7.881 3.24e-15 ***
## age25-29    -0.18709    0.08282  -2.259   0.0239 *
## age30-35    -0.33715    0.08129  -4.148 3.36e-05 ***
## age>35      -0.52692    0.06977  -7.553 4.27e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.259  on 63  degrees of freedom
## Residual deviance:  65.291  on 57  degrees of freedom
## AIC: 396.61
##
## Number of Fisher Scoring iterations: 4

m2_b <- update(m1_b, . ~ . - dist)
summary(m2_b)

##
## Call:
## glm(formula = c ~ car + age + offset(log(n)), family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.61407  -0.59513  -0.07229   0.78529   2.71480
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.79527    0.07519 -23.877  < 2e-16 ***
## car1-1.5     0.16248    0.05052   3.216   0.0013 **
## car1.5-2     0.39466    0.05498   7.178 7.09e-13 ***
## car>2        0.56956    0.07227   7.881 3.24e-15 ***
## age25-29    -0.18709    0.08282  -2.259   0.0239 *
## age30-35    -0.33715    0.08129  -4.148 3.36e-05 ***
## age>35      -0.52692    0.06977  -7.553 4.27e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.259  on 63  degrees of freedom
## Residual deviance:  65.291  on 57  degrees of freedom
## AIC: 396.61
```

```
##
## Number of Fisher Scoring iterations: 4
```

Remark: after removing the regressor we have an increase of residual deviance, an increase of AIC and lose 3 regression coefficients (because dist has 4 levels).
The LRT (Likelihood Ratio Test) performed by `anova` function has additional options (`test`) once applied to glm objects (see `?anova.glm`): here we specify the distribution we want to use to compute the p-value. Here we'll use chi-square distribution.

```
anova(m2, m1, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: c ~ car + age + offset(log(n))
## Model 2: c ~ offset(log(n)) + car + dist + age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        57     65.291
## 2        54     51.420  3   13.871 0.003086 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m2_b, m1_b, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: c ~ car + age + offset(log(n))
## Model 2: c ~ offset(log(n)) + car + dist + age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        57     65.291
## 2        54     51.420  3   13.871 0.003086 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Remembering that H_0 is "the reduced model is not significantly worse":

- both the model yields the same results, as reasonable
- H_0 is rejected, so we cannot drop `dist` from the model, it's a significant variable: there significant differences among policyholders living in different areas

13.4.2 Simplification to a dummy

In `m1_b`

```
summary(m1_b)

##
## Call:
## glm(formula = c ~ offset(log(n)) + car + dist + age, family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46558  -0.50802  -0.03198   0.55555   1.94026
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.58753    0.09276 -17.115  < 2e-16 ***
## car1-1.5     0.16134    0.05053   3.193 0.001409 **
## car1.5-2     0.39281    0.05500   7.142 9.18e-13 ***
## car>2        0.56341    0.07232   7.791 6.65e-15 ***
## dist1       -0.23421    0.06167  -3.798 0.000146 ***
```

```
## dist2       -0.20834    0.06476  -3.217 0.001294 **
## dist3       -0.19568    0.06984  -2.802 0.005081 **
## age25-29    -0.19101    0.08286  -2.305 0.021149 *
## age30-35    -0.34495    0.08137  -4.239 2.24e-05 ***
## age>35      -0.53667    0.06996  -7.672 1.70e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.26  on 63  degrees of freedom
## Residual deviance:  51.42  on 54  degrees of freedom
## AIC: 388.74
##
## Number of Fisher Scoring iterations: 4
```

we've seen that dist1, dist2, dist3 corresponding to the coefficient for the dummy of rural/small town/large town are more or less all set to -0.20 compared to major cities.
We could ask if we can simplify the contribution of `dist` to a dichotomization of major cities vs all the others: the specific linear test to be implement depends on which model we start from.

Considering the models fitted so far, to test the same idea we have to set different null hypotheses (given the different parametrization): each of them can be tested using two (asymptotically equivalent) procedures.

Model with base major cities Considering the model with reference level of dist set to "major cities" `m1_b` the hypothesis to be tested is

$$H_0 : \beta_{dist1} = \beta_{dist2} = \beta_{dist3}$$

which means no difference between policyholders living in areas other than Major Cities (we're not setting them equal to 0, just checking if they can be equal).
We can test it with two procedures

- a Wald test, define the hypothesis through a system of linear equations K*beta = t

```
(K_b <- matrix(c(0, 0, 0, 0, 0, 0, 0, 1, 1, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0),
               nrow = 2, ncol = 10))
```

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

```
(t_b <- matrix(c(0, 0), 2, 1))
```

```
##      [,1]
## [1,]    0
## [2,]    0
```

```
## So the first restriction/line impose equivalence between fifth and
## sixth; the second between fifth and seventh
lht(m1_b, K_b, rhs = t_b)
```

```
## Linear hypothesis test
##
## Hypothesis:
## dist1 - dist2 = 0
## dist1 - dist3 = 0
##
## Model 1: restricted model
## Model 2: c ~ offset(log(n)) + car + dist + age
##
##   Res.Df Df  Chisq Pr(>Chisq)
## 1     56
## 2     54  2 0.7155     0.6992
```

Thus the null is not reject leading to possible simplification in the model. The benefit of this is that we don't need to refit the model under restriction (which is simple btw)

- an LRT test: we build a constrained (smaller) model in which H_0 is imposed through coding of "dist", using major cities as reference category

```
claims$distnocity <- factor(claims$dist != "major cities")
table(claims$distnocity, claims$dist)

##
##         rural small towns large towns major cities
##   FALSE     0           0           0           16
##   TRUE     16          16          16            0

m3_b <- update(m2, . ~ . + distnocity)
summary(m3_b)

##
## Call:
## glm(formula = c ~ car + age + distnocity + offset(log(n)), family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.52958  -0.54340  -0.08503   0.60327   1.99844
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.59171    0.09264 -17.182  < 2e-16 ***
## car1-1.5        0.16229    0.05052   3.213 0.001315 **
## car1.5-2        0.39352    0.05498   7.157 8.25e-13 ***
## car>2           0.56540    0.07228   7.823 5.18e-15 ***
## age25-29       -0.18902    0.08282  -2.282 0.022477 *
## age30-35       -0.34211    0.08130  -4.208 2.58e-05 ***
## age>35         -0.53275    0.06979  -7.634 2.28e-14 ***
## distnocityTRUE -0.21850    0.05853  -3.733 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.259  on 63  degrees of freedom
## Residual deviance:  52.135  on 56  degrees of freedom
## AIC: 385.46
##
## Number of Fisher Scoring iterations: 4

anova(m3_b, m1_b, test = "Chisq") # more or less this is close to the wald test one

## Analysis of Deviance Table
##
## Model 1: c ~ car + age + distnocity + offset(log(n))
## Model 2: c ~ offset(log(n)) + car + dist + age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        56     52.135
## 2        54     51.420  2  0.71483   0.6995
```

Again the null should be rejected

So here we can simplify the model by collapsing dist in major cities vs others.

Model with base rural Considering the first model with base category set to "rural"

```
summary(m1)

##
## Call:
## glm(formula = c ~ offset(log(n)) + car + dist + age, family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46558  -0.50802  -0.03198   0.55555   1.94026
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.82174    0.07679 -23.724  < 2e-16 ***
## car1-1.5          0.16134    0.05053   3.193 0.001409 **
## car1.5-2          0.39281    0.05500   7.142 9.18e-13 ***
## car>2             0.56341    0.07232   7.791 6.65e-15 ***
## distsmall towns   0.02587    0.04302   0.601 0.547597
## distlarge towns   0.03852    0.05051   0.763 0.445657
## distmajor cities  0.23421    0.06167   3.798 0.000146 ***
## age25-29         -0.19101    0.08286  -2.305 0.021149 *
## age30-35         -0.34495    0.08137  -4.239 2.24e-05 ***
## age>35           -0.53667    0.06996  -7.672 1.70e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.26  on 63  degrees of freedom
## Residual deviance:  51.42  on 54  degrees of freedom
## AIC: 388.74
##
## Number of Fisher Scoring iterations: 4
```

to test the same hypothesis we have to jointly set/check/test that there are no differences between rural and small towns and between rural and large towns:

$$H_0 : \beta_{\text{small towns}} = \beta_{\text{large towns}} = 0$$

the way to do it:

- via Wald test statistic: define the hypothesis through a system of linear equations K*beta = t

```
(K <- matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0),
             2, 10))

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

(t <- matrix(c(0, 0), 2, 1))

##      [,1]
## [1,]    0
## [2,]    0

## we just set the fifth and sixth coefficient equal to 0
lht(m1, K, rhs = t)

## Linear hypothesis test
##
## Hypothesis:
## distsmall towns = 0
## distlarge towns = 0
```

```
##
## Model 1: restricted model
## Model 2: c ~ offset(log(n)) + car + dist + age
##
##   Res.Df Df  Chisq Pr(>Chisq)
## 1     56
## 2     54  2 0.7155     0.6992

## same results as before with lht(m1_b, K_b, rhs = t_b)
```

- via LRT: we build a constrained (smaller) model in which H0 is imposed through coding of "dist", using as reference category a collapsed category including rural areas, small towns and large towns

```
claims$distcity <- factor(claims$dist == "major cities")
table(claims$distcity, claims$dist)

##
##         rural small towns large towns major cities
##   FALSE    16          16          16            0
##   TRUE      0           0           0           16

levels(claims$distcity) <- c("Other", "MajorCity")
m3 <- update(m2, . ~ . + distcity)
summary(m3)

##
## Call:
## glm(formula = c ~ car + age + distcity + offset(log(n)), family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.52958  -0.54340  -0.08503   0.60327   1.99844
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.81021    0.07532 -24.034  < 2e-16 ***
## car1-1.5           0.16229    0.05052   3.213 0.001315 **
## car1.5-2           0.39352    0.05498   7.157 8.25e-13 ***
## car>2              0.56540    0.07228   7.823 5.18e-15 ***
## age25-29          -0.18902    0.08282  -2.282 0.022477 *
## age30-35          -0.34211    0.08130  -4.208 2.58e-05 ***
## age>35            -0.53275    0.06979  -7.634 2.28e-14 ***
## distcityMajorCity  0.21850    0.05853   3.733 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 236.259  on 63  degrees of freedom
## Residual deviance:  52.135  on 56  degrees of freedom
## AIC: 385.46
##
## Number of Fisher Scoring iterations: 4

anova(m3, m1, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: c ~ car + age + distcity + offset(log(n))
## Model 2: c ~ offset(log(n)) + car + dist + age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        56     52.135
## 2        54     51.420  2  0.71483   0.6995
```

```
## same result as before obtained with anova(m3_b, m1_b, test = "Chisq")
```

In both cases, the result obtained with the likelihood ratio test statistic is approximately equivalent to the result with the Wald test statistic.

13.5 Interpreting coefficients

Given we're working with a poisson regression model it can be convenient to look at exponential of regression coefficients, that can interpreted as multiplicative factor affecting the expected number of counts/rates.
Here given that we have an offset term (log(n)), we are actually regressing onto a *rate* (akin to a "risk of"):

```
exp(coefficients(m3))
```

```
##       (Intercept)          car1-1.5          car1.5-2             car>2
##         0.1636202         1.1762025         1.4821860         1.7601435
##          age25-29          age30-35            age>35 distcityMajorCity
##         0.8277722         0.7102694         0.5869893         1.2442031
```

For example:

- the rate of claims from policyholders living in Major Cities (keeping everything else fixed) is 1.24 times the rate for the other areas (baseline). There's a 24% increment in the rate by changing `dist` from any other area to Major Cities (eg by traffic).
- the rate of claims from polichyholders older than 35 (keeping everything else fixed) is 0.58 times the rate of claims from policyholders younger than 25: moving from the baseline level for age to the last class there's a 42% decrement in the rate of claims

13.6 Focus on the offset

13.6.1 Don't forget it

When we've different exposure levels it is crucial including the offset term; let see what happens if we fit the model without offset

```
m4 <- glm(c ~ car + age + distcity, data = claims, family = poisson)
summary(m4)
```

```
##
## Call:
## glm(formula = c ~ car + age + distcity, family = poisson, data = claims)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.1404  -2.0722  -0.6461   1.5351   6.4814
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.46014    0.07708  31.917  < 2e-16 ***
## car1-1.5           0.98960    0.05045  19.617  < 2e-16 ***
## car1.5-2           0.47070    0.05490   8.574  < 2e-16 ***
## car>2             -0.58927    0.07211  -8.172 3.04e-16 ***
## age25-29           0.56769    0.08272   6.863 6.74e-12 ***
## age30-35           0.68217    0.08108   8.413  < 2e-16 ***
## age>35             2.19916    0.06965  31.575  < 2e-16 ***
## distcityMajorCity -1.06075    0.05849 -18.135  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4236.68  on 63  degrees of freedom
## Residual deviance:  491.64  on 56  degrees of freedom
## AIC: 824.96
##
## Number of Fisher Scoring iterations: 5

pchisq(summary(m4)$deviance, summary(m4)$df.residual, lower.tail = FALSE)

## [1] 6.330768e-71
```

we have:

- a much larger deviance, a much larger AIC, and looking at the goodness of fit test the p-value is extremely small (so the model is not adequate)
- there are some regression coefficients which sign change due to correlation between the predictors and the offset: eg with `distcityMajorCity` ignoring the exposure level we have a negative coefficient suggesting that major cities has a positive impact (policy holders coming from major cities are less risky than policy holders from other areas). However this is just an artifact due to the correlation between the predictors and the offset: in this case the lower value for claims in Major Cities is NOT due to a lower risk (rate of claims) there are simply less policyholders in Major Cities, and thus the exposure is lower

```
(cs <- tapply(claims$c, claims$distcity, sum)) # small claims in

##     Other MajorCity
##      2825       326

(ns <- tapply(claims$n, claims$distcity, sum)) # but even smallers number of policy holders

##     Other MajorCity
##     21365      1994

cs / ns  ## so the raw rate of claims is actually higher in major cities

##     Other MajorCity
## 0.1322256 0.1634905
```

13.6.2 Offset or standard regressor?

A second point is: is it reasonable to treat `log(n)` as an offset, rather than as an additional regressor? In principle we could insert it like a normal variable, let's see what happens (there should be a positive association between number of policy holders and numer of claims)

```
mod_offreg <- glm(c ~ log(n) + car + age + distcity,
                  data = claims,
                  family = poisson)
summary(mod_offreg)

##
## Call:
## glm(formula = c ~ log(n) + car + age + distcity, family = poisson,
##     data = claims)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.50337  -0.54687  -0.09821   0.57223   1.99536
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.74398    0.23182  -7.523 5.36e-14 ***
```

```
## log(n)             0.98526    0.04881  20.185  < 2e-16 ***
## car1-1.5           0.17407    0.06384   2.727 0.006398 **
## car1.5-2           0.39435    0.05505   7.163 7.88e-13 ***
## car>2              0.54781    0.09281   5.902 3.58e-09 ***
## age25-29          -0.17924    0.08894  -2.015 0.043880 *
## age30-35          -0.32872    0.09265  -3.548 0.000388 ***
## age>35            -0.49465    0.14430  -3.428 0.000608 ***
## distcityMajorCity  0.19870    0.08788   2.261 0.023754 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4236.679  on 63  degrees of freedom
## Residual deviance:   52.044  on 55  degrees of freedom
## AIC: 387.37
##
## Number of Fisher Scoring iterations: 4
```

As expected thgere's a positive association: furthermore its value is near 1, like for what is set for offset variable.
We could check wheter if it should be used as a genuine regressor or as an offset checking it's value is not different from 1 just by testing $H_0 : \beta_offset = 1$: *if we reject this hypothesis* then *we should not use it as an offset* and rather use it like a standard regressor/variable

```
### Hypothesis testing: we test just one linear restriction so we have
### a matrix K with one row
K_offreg <- rep(0, 9)
K_offreg[2] <- 1 # select the coefficient of log(n)
t_offreg <- 1    # set it to 1
lht(mod_offreg, K_offreg, rhs = t_offreg)

## Linear hypothesis test
##
## Hypothesis:
## log(n) = 1
##
## Model 1: restricted model
## Model 2: c ~ log(n) + car + age + distcity
##
##   Res.Df Df  Chisq Pr(>Chisq)
## 1     56
## 2     55  1 0.0911     0.7627
```

The regression coefficient associated with `log(n)` is not significantly different from 1, so we can use it via offset.
One final thing; if we were to perform a GOF test on the model with `log(n)` as regressor:

```
pchisq(summary(mod_offreg)$deviance, summary(mod_offreg)$df.residual, lower.tail = FALSE)

## [1] 0.5882963
```

The p value is still very large: even the model where we're using the `log(n)` as regressor could be considered an adequate model for the data.
So we came up with at least three models passing the GOF test (original with offset, dichotomic `dist` with offset, dichotomic `dist` without offset). So the GOF is not conclusive; we can look for the lowest AIC (which is dichotomic `dist` with offset).

Chapter 14

Generalised linear models for binary outcomes

Remark 131. **Overdispersion** problem can be a thing even in dichotomic outcome data: we have a math relation between expected and variance but sometimes in real data the data is larger than expected when using binomial distribution. Extensions are available: we don't see them here.

14.1 Binary outcomes

14.1.1 Introduction

Example 14.1.1 (Some binary (dicothoumous) outcomes)**.** We could have:

- *presence/absence of a given feature*: employed/unemployed woman
- *success/failure of a given trial*: recovery from a disease (yes/no), presence/absence of side-effects after a given treatment
- *positive/negative answer to a question*: satisfied/unsatisfied costumer, in favor/against a given law
- ...

Important remark 108 (Dummy/indicator variables). As seen any dichotomous outcomes can be numerically coded using a dummy/indicator variable (with 1 the outcome of interest):

$$Z = \begin{cases} 0 & \text{absence/failure/negative answer} \\ 1 & \text{presence/success/positive answer} \end{cases}$$

Thus:

- Z_j is r.v. describing the observed value for the dependent variable on the j-th sample unit ($j = 1, \dots, N$, note capital N)
- $\mathbf{x}_j = (x_{0j}, x_{1j} \dots, x_{pj})^\top$ is the $(p+1)$-dimensional vector containing the regressor values observed on the j-th sample unit ($x_{0j} = 1, \forall j$ constant regressor associated with the model intercept)

Remark 132 (On notation change). Note here we used a slightly different notation Z_j with $j = 1, \dots, N$, instead of Y_i with $i = 1, \dots, n$; this because Y_i will appear again later, with a slightly different meaning

14.1.2 Bernoulli distributions and exponential families

When dealing with variable taking value 0/1, a natural choice from a probabilistic pov is the use of bernoulli distribution

We say that $z_j \in \{0,1\}$ are distributed according to a bernoulli $Z_j \sim \text{Ber}(\pi_j)$ with parameter $\pi_j \in [0,1]$ iff:

$$f(z_j, \pi_j) = \pi_j^{z_j}(1-\pi_j)^{1-z_j} = \exp\left\{\frac{1}{1}\left[z_j \ln\frac{\pi_j}{1-\pi_j} + \ln(1-\pi_j)\right] + 0\right\}$$

The bernoulli distribution belongs to the exponential family (without explicit nuisance parameter and weight):

$$Z_j \sim \text{EF}(\text{logit}(\pi_j), \phi = 1, w_j = 1)$$

having the following distinctive features:

- the natural parameter associated is the logit function of π_j

$$b(\pi_j) = \ln\frac{\pi_j}{1-\pi_j} = \text{logit}(\pi_j)$$

- $c(\theta) = \ln(1-\pi_j)$
- the d function is a null constant: $d(z_j, \phi, w_j) = 0$

Thanks to the properties of exponential families of distributions:

$$\text{E}[Z_j] = -\frac{c'(\pi_j)}{b'(\pi_j)} = -\frac{-\frac{1}{1-\pi_j}}{\frac{1}{\pi_j} + \frac{1}{1-\pi_j}} = \frac{\pi_j(1-\pi_j)}{1-\pi_j} = \pi_j$$

$$\text{Var}[Z_j] = \frac{\phi}{w_j}\frac{\text{E}'[Z_j]}{b'(\pi_j)} = \frac{1}{1}\frac{1}{\frac{1}{\pi_j} + \frac{1}{1-\pi_j}} = \frac{1}{\frac{1}{\pi_j(1-\pi_j)}} = \pi_j(1-\pi_j)$$

So again if we're going to use the bernoulli distribution for our regression problem, we're in a situation where the conditional variance will be a function of the expected value (as in Poisson) so we're in an heteroskedastic setting with the functional form given by the expression $\pi_j(1-\pi_j)$ (it's a non monotonic function of π_j: start close to 0, max at $\pi_j = 0.5$ and approaches 0 at 1 ... the max variability for bernoulli distribution is achieved when $\pi_j = 0.5$).

Remark 133. In general, for each distribution of the exponential family we have a specific relation between the expected value and the variance

Remark 134. Belonging to the exponential family, the binomial distribution can be used for the probabilistic component of a GLM

14.2 GLM for binary outcomes

14.2.1 Basic definition

An example of GLM for binary outcome can be defined by considering:

- as the *Probabilistic component* the bernoulli distribution and assuming independence between the N obs

$$Z_j|\mathbf{x}_j \sim \text{Ber}(\pi_j), \quad \text{independent } j = 1, \ldots, N$$

- as *Systematic component* we'll choose a set of regressor and a proper link function h

$$\text{E}[Z_j|\mathbf{x}_j] = \pi_j = h(\eta_j) = h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)$$

  Since $\pi_j \in [0,1]$, the link function $h(\cdot)$ should be chosen among all functions with image coinciding with interval $[0,1]$: $h(\cdot) : \mathbb{R} \longmapsto [0,1]$

Remark 135. So the use of the exponential as link function in this kind of model can be questionable since the image is $\mathbb{R}^+$; if we are sure that any linear predictor ≤ 0 we could use the exponential, in theory. Otherwise we could end up with a probability estimate that is larger than one and the estimate variance is negative.

14.2.2 Log-likelihood for Bernoulli GLMs

No matter which link function one choose, any GLM with a probabilistic component based on the Bernoulli distribution is characterised by the following log-likelihood function:

$$l(\beta_0, \ldots, \beta_p) = \sum_{j=1}^{N} \left\{ z_j \ln \frac{h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)}{1 - h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)} + \ln \left[1 - h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)\right] \right\}$$

where:

- z_j is 0 or 1 dependeding on the unit
- $\ln \frac{h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)}{1 - h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)}$ is the logit function applied to π_j
- $\ln \left[1 - h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)\right]$ is the logarithm of 1 - the probability

This is the loglikelihood for GLM in which we have a *unit data structure* where:

- $\mathbf{z} = (z_1, \ldots, z_N)^\top$ is a vector of 0/1 values (*1 value for each sample unit*)
- $\mathbf{X}$ is a $N \times (p+1)$ matrix (*1 row for each sample unit*)

Remark 136. We'll see that in model for binary outcomes we can simplify the expression of the likelihood (using an alternative data structure).
What happens to the likelihood if we have some *units sharing the same covariate pattern*? Checking covariate pattern becomes crucial when evaluating goodness of fit of the model (with test or residual)

14.2.3 Bernoulli GLMs & covariate patterns

Suppose that some sample units are characterised by the same covariate pattern, thus *the matrix* $\mathbf{X}$ *contains some identical rows*; for the notation we have:

- n is number of unique covariate patterns in the sample (*number of unique rows in* $\mathbf{X}$). We'll have that $n < N$ (less than the number of units);
- $\mathbf{x}_i = (x_{0i}, x_{1i} \ldots, x_{pi})^\top$ is the i-th covariate pattern ($i = 1, \ldots, n$)
- n_i is the number of sample units showing the i-th covariate pattern. Units characterised by the same i-th covariate pattern will show the *same value for the linear predictor* and for the *conditional expected value* (being both betas and regressors common):

  $$\mathbf{x}_j^\top \boldsymbol{\beta} = \beta_0 x_{0j} + \beta_1 x_{1j} + \ldots + \beta_p x_{pj} = \beta_0 x_{0i} + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} = \mathbf{x}_i^\top \boldsymbol{\beta} = \eta_i$$

  $$\pi_j = h\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right) = h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right) = h(\eta_i) = \pi_i$$

- the number of units showing the i-th covariate pattern *and* a value of z_j equal to 1 is just obtained by summing within the covariate pattern group:

  $$y_i = \sum_{j:\eta_j = \eta_i} z_j$$

Remark 137. So y_i is the number of successes out of n_i independent trials which are also identically distributed (bernoulli with π_i): therefore these y_i successes will be binomial distributed

Remark 138. If we go back to our likelihood we can rewrite the same loglikelihood with as many contribution as the number n of covariate patterns

14.2.4 Log-likelihood for Bernoulli GLMs & covariate patterns

In presence of repeated covariate patterns, the log-likelihood function of any GLM with a Bernoulli probabilistic component can be re-expressed as the sum of n elements, one for each unique covariate pattern, as follows:

$$l(\beta_0, \ldots, \beta_p) = \sum_{i=1}^{n} \underbrace{\left\{ y_i \ln \frac{h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1 - h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)} + n_i \ln \left[1 - h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)\right] \right\}}_{\sum_{j:\eta_j = \eta_i} \left\{ z_j \ln \frac{h(\eta_i)}{1 - h(\eta_i)} + \ln[1 - h(\eta_i)] \right\}} \tag{14.1}$$

for all units characterized by the same covariate pattern, we'll have a single contribution (guarda sotto l'underbrace) to the loglikelihood

- z_j the specific success or failure for the unit time
- a constant common/value both for the b and the c funtion: $\ln \frac{h(\eta_i)}{1-h(\eta_i)}$ and $\ln\left[1-h(\eta_i)\right]$
- these can summed/regrouped: in some sense what we're doing is to compressing the data by covariate pattern, removing "redundant" information.

The "relevant" information about $\boldsymbol{\beta}$ is provided by the number of successes y_i associated with each covariate pattern, among the n_i of its units (rather than the individual successes).
This is just a rewriting so maximization using this one will give the same results compared to maximizing by single unit.

14.2.5 Binomial distributions

Recalling that if $Z_1, \ldots, Z_{n_i}$ are i.i.d. Bernoulli random variables with common parameter π_i, their sum is binomial distributed

$$Y_i = \sum_{j=1}^{n_i} Z_j \sim \text{Bin}(n_i, \pi_i)$$

we have that, apart from an additive constant depending only on the values y_i (not on the betas), the loglikelihood $l(\beta_0, \ldots, \beta_p)$ found in 14.1 coincides with the log-likelihood function associated with the following n binomial random variables:

$$Y_i|\mathbf{x}_j \sim \text{Bin}\Big(n_i, \pi_i = h\Big(\mathbf{x}_i^\top \boldsymbol{\beta}\Big)\Big) \quad \text{independent } i = 1, \ldots, n$$

14.2.6 The covariate pattern based data structure

Important remark 109 (Notation of covariate-pattern based data structure). We have

- $\mathbf{y} = (y_1, \ldots, y_n)^\top$ be the number of presences/successes/positive answers (*1 value for each unique covariate pattern*)
- $\mathbf{n} = (n_1, \ldots, n_n)^\top$ be the number of "trials" (*1 value for each unique covariate pattern*) or
- some software ask, instead of $\mathbf{n}$ but equivalently, the difference $\mathbf{n}-\mathbf{y} = (n_1 - y_1, \ldots, n_n - y_n)^\top$, that is the number of absences/failures/negative answers (*1 value for each unique covariate pattern*)
- $\mathbf{X}$ be the regressor matrix, a $n \times (p+1)$ matrix (*1 row for each each unique covariate pattern*)

Remark 139. Let's compare it with the classical *unit based* data structure.

Important remark 110 (Comments about the two data structures). We have that:

- the two data structures coincide if $n_i = 1, \forall i$ (if $n = N$): *the 0/1 structure is a special case of the successes/trials structure*
- in the context of GLM the distiction between two data structures becomes crucial for some asymptotic properties of GLMs (with a Bernoulli probabilistic component), since two distinct asymptotic concepts can be studied/exploited:

  A) some asymptotic stuff holds if the total sample size goes to infinity (is sufficiently large): $N = \sum_i n_i \longrightarrow \infty$

  B) other asymptotics properties holds if and only if the number of unit for *each* covariate pattern go to ∞ (are sufficiently large): $n_i \longrightarrow \infty \ \forall i$ (*fixed cell asymptotics*)

  Note that B) $\implies$ A), but A) $\not\implies$ B)

14.2.7 Relative frequencies and their properties

Remark 140. Another step useful from technical pov when dealing with glm with bernoulli prob component is to consider relative frequencies rather than number of successes

When we have a binomial distribution we can transform number of successes in the relative frequencies of successes just by dividing by n: in our context

- we mean by $p_i = \dfrac{y_i}{n_i}$ the relative frequency of presences/successes/positive answers out of n_i trials (*1 value for each unique covariate pattern*)
- we'll denote this random variable as P_i

The log-likelihood function of any GLM with a Bernoulli probabilistic component can also be re-expressed in terms of p_i (it's exactly the same, the only thing that changes is the y_i/n_i (vs y_i)):

$$l(\beta_0, \dots, \beta_p) = \sum_{i=1}^{n} n_i \left\{ \underbrace{\frac{y_i}{n_i}}_{p_i} \ln \frac{h(\mathbf{x}_i^\top \boldsymbol{\beta})}{1 - h(\mathbf{x}_i^\top \boldsymbol{\beta})} + \ln \left[1 - h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right) \right] \right\}$$

Having a random variable distributed according to a binomial distribution also the relative frequency will have a binomial/exponential family distribution; if

$$Y_i | \mathbf{x}_i \sim \text{Bin}\left(n_i, \pi_i = h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)\right) \quad \text{independent } i = 1, \dots, n$$

this *implies* that

$$P_i = \left. \frac{Y_i}{n_i} \right| \mathbf{x}_i \sim \text{EF}\left(b(\pi_i) = \text{logit}\left[h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)\right], \phi = 1, w_i = n_i\right) \quad \text{independent } i = 1, \dots, n$$

so in the exponential family context

- the natural parameter is the logit transformation of probability of success
- n_i plays the role of the weight in the general exponential family framework
- no nuisance parameters

Important remark 111. when dealing with binary outcomes one can work with three different loglikelihood:

- the loglikelihood associated to the bernoulli rv at individual level;
- the loglikelihood associated to the binomial rv at covariate pattern level;
- the loglikelihood associated to the *relative frequencies* of successes, again at a covariate pattern level.

In terms of general theory for GLMs, the relevant likelihood are either the first (bernoulli) and the third (relative frequencies).
Either working with one or another we get exactly the same loglikelihood .
In order to perform some tasks we can choose any of the two; however when evaluating the goodness of fit of the model, as we'll see, it is crucial to use the second datastructure, so the loglikelihood associated to relative frequencies of successes

NB: cioé credo intenda che la loglike è diversa ma si massimizza con lo stesso set di parametri

Important remark 112 (Relative frequencies - quick reminder on some properties). Thanks to the properties of exponential families of distributions,

$$\text{E}[P_i] = -\frac{c'(\pi_i)}{b'(\pi_i)} = -\frac{-\dfrac{1}{1-\pi_i}}{\dfrac{1}{\pi_i} + \dfrac{1}{1-\pi_i}} = \frac{\pi_i(1-\pi_i)}{1-\pi_i} = \pi_i$$

$$\text{Var}[P_i] = \frac{\phi}{w_i} \frac{\text{E}'[Z_i]}{b'(\pi_i)} = \frac{1}{n_i} \frac{1}{\dfrac{1}{\pi_i} + \dfrac{1}{1-\pi_i}} = \frac{1}{\dfrac{n_i}{\pi_i(1-\pi_i)}} = \frac{\pi_i(1-\pi_i)}{n_i}$$

so the expected value coincides with the probability of success and the variance is the variance of a binomial distribution divided by n_i (which is function of the regressors, so we have intrinsic heteroskedasticity, with variance reaching the peak when $\pi_i = 0.5$ and approaching 0 when π_i tends to 0 or 1

14.3 Logistic regression models

Remark 141. The previous was a general introduction for GLM with binary outcomes: key point is that the natural choice for probabilistic component in a bernoulli/binomial/relative frequency distribution; there are other distributions which are variations of these distributions. Now we start looking at first specific example of GLM for binary data, that is at model using specific link function; we'll the move to models using other link functions.

Remark 142. The very first example is the well known logistic regression model

14.3.1 Model definition

Let:

- Y_i be a r. v. describing the observed number of presences/successes/positive answers on the n_i sample units showing the i-th covariate pattern
- $\mathbf{x}_i = (x_{0i}, x_{1i} \dots, x_{pi})^\top$ be a $p+1$-dimensional vector containing the regressor values characterising the i-th covariate pattern - $x_{0i} = 1 \ \forall i \ (i = 1, \dots, n)$
- $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$ be the linear predictor associated with the i-th covariate pattern
- n_i be the number of sample units showing the i-th covariate pattern
- as assumption, the conditional distribution of $Y_i|\mathbf{x}_i$ is binomial with n_i number of trial and probability of success expressed as logistic function of the linear predictor

$$Y_i|\mathbf{x}_i \sim \text{Bin}\left(n_i, \pi_i = \frac{\exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1 + \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}\right) \quad \text{independent } i = 1, \dots, n$$

- **Probabilistic component**: as seen if we have a binomial conditional distribution we can obtain also the distribution of relative frequency of successes P_i; it has a distribution belonging to an exponential family satisfying all the requirements to be used as probabilistic component of a GLM. We have the same function b (logit of π_i), the same function c, and weights n_i):

$$Y_i|\mathbf{x}_i \sim \text{Bin}(n_i, \pi_i) \iff P_i = \left.\frac{Y_i}{n_i}\right| \mathbf{x}_i \sim \text{EF}(b(\pi_i) = \text{logit}(\pi_i), \phi = 1, w_i = n_i)$$

  When we model the relative frequency of successes we model parallely as well the number of successes be cause by construction it will be n_i times the expected value of relative frequency, while the variance in number of successes will be the one in relative frequency multiplied n_i^2

$$\begin{aligned} &\text{E}\left[P_i|\mathbf{x}_i\right] = \pi_i \implies \text{E}\left[Y_i|\mathbf{x}_i\right] = n_i\pi_i \\ &\text{Var}\left[P_i|\mathbf{x}_i\right] = \frac{\pi_i(1-\pi_i)}{n_i} \implies \text{Var}\left[Y_i|\mathbf{x}_i\right] = n_i\pi_i(1-\pi_i) \end{aligned}$$

- **Systematic component**: the link function is the logistic function

$$\mathbb{E}\left[P_i|\mathbf{x}_i\right] = \pi_i = \frac{\exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1 + \exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}$$

14.3.2 Logistic function

Important remark 113 (Logistic function focus). Defined as

$$\frac{\exp(\cdot)}{1 + \exp(\cdot)} : \mathbb{R} \longmapsto (0, 1)$$

is displayed in 14.1. Its features are coherent with the parameter space associated with the binomial distribution of relative frequencies of success because by construction whichever value of linear predictor η_i we consider, we'll always get a value of logistic function bounded between 0 and 1.
It's monotonic (and thus invertible), its also continue and differentiable; so it satisfies all the requirements of the link functions for GLM.

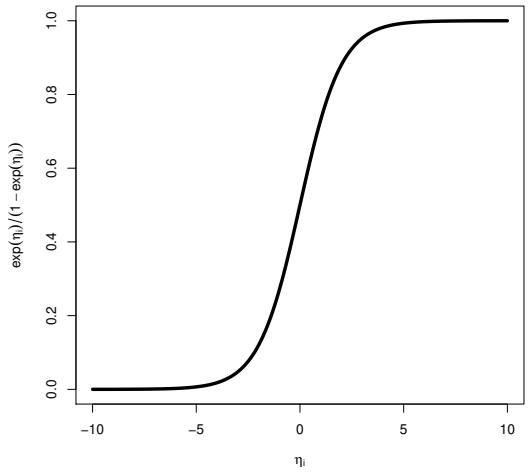


Figure 14.1: Logistic function

OO: non chiaro dopo che la logit function is as link function

Important remark 114 (Canonical link function). Furthermore the logistic function is the **canonical link function** associated with the bernoulli/relative frequency distribution. With a little of manipulation we can workout the expression for the *inverse of the logistic function*

$$\begin{aligned}
\pi_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)} &\iff \pi_i\left[1+\exp(\eta_i)\right] = \exp(\eta_i) \\
&\iff \pi_i = \exp(\eta_i) - \pi_i \exp(\eta_i) \\
&\iff \pi_i = \exp(\eta_i)(1-\pi_i) \\
&\iff \frac{\pi_i}{1-\pi_i} = \exp(\eta_i) \\
&\iff \text{logit}(\pi_i) = \eta_i
\end{aligned}$$

So the inverse of the logistic function is just the logit function: this means that logistic regression models belongs not only to the class of glm with bernoulli/relative frequency conditional distribution, but are also based on the canonical link function for these exponential family. By using logistic regression model we're implicitly equating the linear predictor to the natural parameter of the bernoulli distribution or to the relative frequency associated with binomial distribution.

Remark 143. In the next part we'll follow the steps followd with poisson regression model, exploiting the general results for GLM to get loglikelihood and associated relevant quantities for logistic regression models.

14.3.3 Log-likelihood

We're using the logit function as the link function: by replacing it in the general definition of the loglikelihood and with a bit of manipulation we can see that:

$$\begin{aligned}
l(\boldsymbol{\beta}) &= \sum_{i=1}^{n} n_i \left\{ \frac{y_i}{n_i}\eta_i - \ln\left[1 - \frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right]\right\} + c \\
&= \left[\beta_0 \sum_{i=1}^{n} y_i + \ldots + \beta_p \sum_{i=1}^{n} y_i x_{pi}\right] - \sum_{i=1}^{n} n_i \ln\left[1 - \frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right] + c
\end{aligned}$$

so also for logistic regression models we can work out a set of (minimal) *sufficient statistics* for $\boldsymbol{\beta}$, that are $\sum_{i=1}^{n} y_i, \ldots, \sum_{i=1}^{n} y_i x_{pi}$.

Finally, in the equation *c is an additive constant that does not depend on* $\boldsymbol{\beta}$ (and thus can be ignored during maximization)

Remark 144. So what carries the relevant information of the parameters is not the individual information but it's what happens at the covariate pattern level (number of successes and failures for a specific covariate pattern).
With logistic regression we can make a further "compression" of the data because what matters are the $(p+1)$ summaries of sufficient statistics: the total sum of successes plus the sum of the crossproduct of the successes in each covariate pattern times the considered covariate value.

Remark 145. When we're dealing with logistic regression and we focus on sampling properties/asymptotic properties we will consider a sample space in which we have a fixed regressor matrix (we condition on the regressor matrix).
When we condition on the regressor matrix at the individual level we're implicitly also fixing the regressor matrix at the covariate pattern level, with one key thing to keep in mind: when conditioning/fixing the covariate pattern implicitly we're conditioning/fixing the number of unit n_i associated to it as well.

14.3.4 Score function

Considering the first partial derivative, the general formula for the generic element of the score function (for a GLM with canonical link function) is

$$\begin{aligned} U_j(\boldsymbol{\beta}) &\overset{(1)}{=} \frac{1}{\phi}\sum_{i=1}^{n} w_i \left\{p_i - \mathbb{E}\left[P_i|\mathbf{x}_i\right]\right\} x_{ji} \\ &= \frac{1}{1}\sum_{i=1}^{n} n_i \left[p_i - \frac{\exp\left(\eta_i\right)}{1+\exp\left(\eta_i\right)}\right] x_{ji} \\ &= \sum_{i=1}^{n} \left[y_i - n_i \frac{\exp\left(\eta_i\right)}{1+\exp\left(\eta_i\right)}\right] x_{ji} \\ &= \sum_{i=1}^{n} \left\{y_i - \underbrace{\mathbb{E}\left[Y_i|\mathbf{x}_i\right]}_{\mu_i}\right\} x_{ji} \end{aligned}$$

in (1) we applied the basic definition and then substituted. Of note, in order to derive the generic expression for an element of the score function we started from the general expression obtained when examining the GLM class, and by particularizing the generic expression using the (features of the) relative frequency of successes.
While we started from relative frequencies, the final expression is based on the number of successes.
From the technical pov the last expression is the same we've seen for the poisson regression model.

14.3.5 Fisher information

Similar idea is exploited for the observed/expected Fisher information matrix (they're the same since we're using canonical link function). Starting from the general definition and plugging in the relative frequency stuff, we end with the generic element of the (observed and expected) Fisher information matrix for a GLM with canonical link function as follows:

$$\begin{aligned} i_{jl}(\boldsymbol{\beta}) &= I_{jl}(\boldsymbol{\beta}) \\ &= \sum_{i=1}^{n} \left(\frac{w_i}{\phi}\right)^2 \mathrm{Var}\left[P_i|\mathbf{x}_i\right] x_{ji}x_{li} \\ &= \sum_{i=1}^{n} \left(\frac{n_i}{1}\right)^2 \frac{\pi_i(1-\pi_i)}{n_i} x_{ji}x_{li} \\ &= \sum_{i=1}^{n} n_i \frac{\exp\left(\eta_i\right)}{\left[1+\exp\left(\eta_i\right)\right]^2} x_{ji}x_{li} \\ &= \sum_{i=1}^{n} \mathrm{Var}\left[Y_i|\mathbf{x}_i\right] x_{ji}x_{li} \end{aligned}$$

Again with a little bit of manipulation we get an equivalent expression based on number of successes.
Again we recognize an expression very similar to the poisson regression model one; only difference is that here we have the conditional variance for the number of successes (binomial distribution) instead of the conditional variance for the counts (poisson distribution).

14.3.6 Hessian matrix

Finally the (j,l)-th element of the Hessian matrix of the log-likelihood function (just minus the observed fisher information matrix)

$$H_{jl}(\boldsymbol{\beta}) = \frac{\partial^2}{\partial\beta_j\partial\beta_l} l(\boldsymbol{\beta}) = -i_{jl}(\boldsymbol{\beta}) = -\sum_{i=1}^{n} n_i \frac{\exp(\eta_i)}{[1+\exp(\eta_i)]^2} x_{ji}x_{li}$$

14.3.7 Matrix representation

All these quantities can be represented using matrix notation:

- $\mathbf{y} = (y_1, \ldots, y_n)^\top$ is an n-dimensional vector that contains the observed number of presences/successes/positive answers on the n unique covariate patterns;
- $\mathbf{X}$ is an $n \times (p+1)$ matrix that contains the n unique covariate patterns;
- the n-dimensional vector that contains the conditional expected values of $Y_1, \ldots, Y_n$ for the n unique covariate patterns is

$$\boldsymbol{\mu} = \left(n_1 \frac{\exp(\eta_1)}{1+\exp(\eta_1)}, \ldots, n_n \frac{\exp(\eta_n)}{1+\exp(\eta_n)} \right)^\top$$

- the $n \times n$ diagonal matrix that contains the conditional variances of $Y_1, \ldots, Y_n$ for the n unique covariate patterns on the main diagonal

$$\mathbf{W} = \begin{bmatrix} n_1 \frac{\exp(\eta_1)}{[1+\exp(\eta_1)]^2} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & n_n \frac{\exp(\eta_n)}{[n+\exp(\eta_n)]^2} \end{bmatrix}$$

Remark 146. These are almost the same stuff already seen for the of poisson regression with just two minor modification:

- the definition of the element contained in $\boldsymbol{\mu}$ (which have a different math expression since: we're using a different link function, the logistic instead of the exponential, we consider the numerosity for the covariate pattern)
- the conditional variances associated to the binomial distribution which are diagonal elements on the $\mathbf{W}$ matrix

Once filled the $\boldsymbol{\mu}$ vector and the $\mathbf{W}$ matrix we can rewrite the following quantities in matrix form as well. The equation are exactly the same as for the poisson regression model (the only difference are in $\boldsymbol{\mu}$ and $\mathbf{W}$ content):

- Score function

$$U(\boldsymbol{\beta}) = \mathbf{X}^\top [\mathbf{y} - \boldsymbol{\mu}]$$

- Hessian matrix of the log-likelihood function

$$H(\boldsymbol{\beta}) = -\mathbf{X}^\top \mathbf{W} \mathbf{X}$$

- (observed and expected) Fisher information matrix

$$i(\boldsymbol{\beta}) = I(\boldsymbol{\beta}) = \mathbf{X}^\top \mathbf{W} \mathbf{X}$$

Remark 147. With this common matrix representation we can also get a common expression for the newton raphson and fisher scoring algorithm needed to derive the ML estimates

Remark 148. *For logistic regression models, the Newton-Raphson algorithm is equivalent to the Fisher scoring algorithm*

14.3.8 Properties of the score function

Just a reminder of the usual properties:

$$\begin{aligned}
&\mathrm{E}\left[U(\boldsymbol{\beta})\right] = \mathbf{0}_{p+1} \\
&\mathrm{Var}\left[U(\boldsymbol{\beta})\right] = I(\boldsymbol{\beta}) \\
&I(\boldsymbol{\beta})^{-\frac{1}{2}} U(\boldsymbol{\beta}) \xrightarrow{d} MVN_{p+1}(\mathbf{0}_{p+1}, \mathbf{I}_{p+1}) \\
&\implies U(\boldsymbol{\beta}) \approx MVN_{p+1}(\mathbf{0}_k, I(\boldsymbol{\beta}))
\end{aligned}$$

with the specification that, particulary in this case, *asymptotic properties of the score function hold both if*

A) $N = \sum_i n_i \longrightarrow \infty$ and thus even if

B) $n_i \longrightarrow \infty \; \forall i$

Sufficient condition is that we have a large sample size, now matter how many units we have foreach covariate pattern (even in situation where we have 1 unit per covariate pattern we can still rely on this asymptotic behaviour, provided that we have a total sample size which is sufficiently large).

14.4 Maximum likelihood estimation

Here nothing new: $\hat{\mathbf{b}}$ is the maximum likelihood estimate of $\boldsymbol{\beta}$ if and only if

$$l\left(\hat{\mathbf{b}}\right) = \max_{\mathbf{b}} l(\mathbf{b})$$

or, equivalently, if and only if:

- *log-likelihood gradient evaluated at* $\hat{\mathbf{b}}$ is null

$$U\left(\hat{\mathbf{b}}\right) = \frac{\partial}{\partial \boldsymbol{\beta}} l(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\hat{\mathbf{b}}} = \mathbf{0}_{p+1}$$

- the *log-likelihood hessian matrix evaluated at* $\hat{\mathbf{b}}$ is negative definite:

$$H\left(\hat{\mathbf{b}}\right) = \frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} l(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\hat{\mathbf{b}}} \quad \text{negative definite}$$

The maximum likelihood estimate $\hat{\mathbf{b}}$ can be obtained by solving the following system of non-linear equations wrt $\mathbf{b}$:

$$U(\mathbf{b}) = \mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}\right] = \mathbf{0}_{p+1}$$

where

$$\mathbf{m} = \begin{bmatrix} n_1 \dfrac{\exp(\mathbf{x}_1^\top \mathbf{b})}{1 + \exp(\mathbf{x}_1^\top \mathbf{b})} \\ \vdots \\ n_n \dfrac{\exp(\mathbf{x}_n^\top \mathbf{b})}{1 + \exp(\mathbf{x}_n^\top \mathbf{b})} \end{bmatrix}$$

Remark 149. In general, this system does not have an explicit solution (*it is not possible to obtain an analytical formula to compute* $\hat{\mathbf{b}}$). So we rely on numerical techniques as Newton rapson/fisher scoring

14.4.1 Newton-Raphson/Fisher scoring algorithm

Remark 150. For logistic regression, since we're dealing with canonical link function there are no differences between the twos.

Remark 151. Having a common matrix representation will leads to the same general expression for the recursive formula where the only difference is the actual content of the matrix $\mathbf{W}^{(r)}$ and the vector $\mathbf{m}^{(r)}$:

- the diagonal elements of $\mathbf{W}^{(r)}$ are obtained by evaluating the conditional variances of Y_i at the current approximation for the ML estimate

- the $\mathbf{m}^{(r)}$ will be the expected vlaue of number of successes evaluated at the current approzimation for the ML estimate

In this case again we can rewrite the recursive formula to get something resembling OLS expression: here the pseudo dependent variable is $\mathbf{z}^{(r)}$ is the same general expression we've seen for poisson. The common expression is due to the fact that we're using conditional distribution belonging to the exponential family and using a canonical link function.

The $(r+1)$-th approximation $\mathbf{b}^{(r+1)}$ to $\hat{\mathbf{b}}$ is obtained using the recursive formula

$$\begin{aligned}\mathbf{b}^{(r+1)} &= \mathbf{b}^{(r)} + \left(\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X}\right)^{-1} \mathbf{X}^\top \left[\mathbf{y} - \mathbf{m}^{(r)}\right] \\ &= \left(\mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{W}^{(r)} \mathbf{z}^{(r)}\end{aligned}$$

where

$$\mathbf{m}^{(r)} = \begin{bmatrix} n_1 \pi_1^{(r)} \\ \\ \vdots \\ n_n \pi_n^{(r)} \end{bmatrix}, \quad \mathbf{W}^{(r)} = \begin{bmatrix} n_1 \pi_1^{(r)}(1-\pi_1^{(r)}) & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n_n \pi_n^{(r)}(1-\pi_n^{(r)}) \end{bmatrix}$$

$$\pi_i^{(r)} = \frac{\exp(\mathbf{x}_i^\top \mathbf{b}^{(r)})}{1+\exp(\mathbf{x}_i^\top \mathbf{b}^{(r)})}, \quad i = 1, \dots, n$$

$$\mathbf{z}^{(r)} = \mathbf{X}\mathbf{b}^{(r)} + \left[\mathbf{W}^{(r)}\right]^{-1} \left[\mathbf{y} - \mathbf{m}^{(r)}\right]$$

14.4.2 Iterative reweighted least squares

The i-th value of the pseudo-dependent variable $\mathbf{z}^{(r)}$ at the $r-$th step of the Newton-Raphson/Fisher scoring algorithm is equal to

$$\begin{aligned} z_i^{(r)} &= \mathbf{x}_i^\top \mathbf{b}^{(r)} + \frac{1}{n_i \pi_i^{(r)}\left(1-\pi_i^{(r)}\right)} \left[y_i - n_i \pi_i^{(r)}\right] \\ &= \mathbf{x}_i^\top \mathbf{b}^{(r)} + \frac{1}{\pi_i^{(r)}\left(1-\pi_i^{(r)}\right)} \left[p_i - \pi_i^{(r)}\right] \end{aligned}$$

Remark 152. $z_i^{(r)}$ can be *interpreted as* an approximation to $\text{logit}(p_i)$, the value of the link function applied to the observed relative frequency, obtained using a first order Taylor series expansion at current approximation of maximum likelihood estimate $\pi_i^{(r)}$

$$\text{logit}(p_i) \approxeq \text{logit}\left(\pi_i^{(r)}\right) + \left.\frac{\partial \text{logit}(p_i)}{\partial y_i}\right|_{p_i = \pi_i^{(r)}} \left[p_i - \pi_i^{(r)}\right]$$

Remark 153. what's relevant about pseudo dep variable is that it can give us an hint on how to initialize the recursive formula in the NR/Fisher scoring algorithm

14.4.3 Initialisation

The Newton-Raphson/Fisher scoring algorithm can be initialised by setting

- $\mathbf{W}^{(0)} = \mathbf{I}_n$ identity matrix;
- $z_i^{(0)} = \ln \frac{y_i + 0.5}{n_i - y_i + 1}$, that is we approximate the pseudo dependent variable by computing the logit of the observed relative frequency (here 0.5 *and* 1 *are added in order to avoid* $\ln(0)$ or $\ln(+\infty)$)
- $\mathbf{b}^{(1)} = \left(\mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{z}^{(0)} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{z}^{(0)}$

Remark 154. Then the algorithm is applied until a stopping criterion condition is met and we keep the last value of $\mathbf{b}^{(r+1)}$ computed.

14.4.4 Maximum likelihood estimator

In absence of an analytical formula for $\hat{\mathbf{B}}$, the properties of the maximum likelihood estimator must be investigated starting from the (asymptotic) properties of the score function $U(\boldsymbol{\beta})$ (*as for Poisson regression models*).
If a logistic regression model is correctly specified, it is possible to prove that (using the same tools used for Poisson) $U(\boldsymbol{\beta})$ *is approximately equivalent to a linear transformation of* $\hat{\mathbf{B}}$:

$$U(\boldsymbol{\beta}) \approxeq I(\boldsymbol{\beta}) \left[\hat{\mathbf{B}} - \boldsymbol{\beta}\right]$$

This implies that the ML estimator is *asymptotically well behaved* (asymptotically unbiased, efficient, multivariate gaussian):

$$\hat{\mathbf{B}} \xrightarrow{d} MVN_{p+1}\left(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}\right)$$

Again, the asymptotic properties of the maximum likelihood estimator hold if:

A) $N = \sum_i n_i \longrightarrow \infty$ (sufficient condition) and thus even if ...

B) $n_i \longrightarrow \infty \; \forall i$

14.4.5 Estimation of the asymptotic variance

To estimate the asymptotic variance $I(\boldsymbol{\beta})^{-1}$: $\mathbf{X}^\top \mathbf{W} \mathbf{X}$ is uknown (it depends on $\boldsymbol{\beta}$) it can be estimated using $\mathbf{X}^\top \hat{\mathbf{W}} \mathbf{X}$ where we substitute the ML estimates:

$$\hat{\mathbf{W}} = \begin{bmatrix} n_1 \dfrac{\exp(\mathbf{x}_1^\top \hat{\mathbf{b}})}{\left[\exp(\mathbf{x}_1^\top \hat{\mathbf{b}})\right]^2} & 0 & 0 & \dots & 0 \\ 0 & n_2 \dfrac{\exp(\mathbf{x}_2^\top \hat{\mathbf{b}})}{\left[\exp(\mathbf{x}_2^\top \hat{\mathbf{b}})\right]^2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & n_n \dfrac{\exp(\mathbf{x}_n^\top \hat{\mathbf{b}})}{\left[\exp(\mathbf{x}_n^\top \hat{\mathbf{b}})\right]^2} \end{bmatrix}$$

14.5 Deviance, residuals, hypothesis testing and model selection criteria

14.5.1 (Residual) deviance: saturated model

Remark 155. The starting point to evaluate adequacy will be the saturated model.
To spoil things, in order to evaluate adequacy of logistic regression model will be fundamental to focus on the covariate pattern data structure; in order to define the saturaated model we've to focus on covariate pattern data structure

Important remark 115. Given a specific model that we've fit:

$$M : \begin{cases} Y_i|\mathbf{x}_i \sim \text{Bin}\left(n_i, \frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right) \\ \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} \end{cases} \qquad \text{independent } i = 1, \dots, n$$

we can define the corresponding saturated model for M, which will be a model with a number of parameters π_i for the expected values that is equal to the number of unique *covariate patterns* in the matrix $\mathbf{X}$ (equal to the number of unique values for η_i):

$$M_{sat} : Y_i|\mathbf{x}_i \sim \text{Bin}(n_i, \pi_i) \quad \text{independent } i = 1, \dots, n$$

(without explicit expression linking π_i to $\mathbf{x}_i$)

14.5.2 Maximum likelihood estimation of $\pi_1, \pi_2, \ldots, \pi_n$

- the Log-likelihood function of the saturated model is given by n contribution, one for each covariate pattern:

$$l(\pi_1, \ldots, \pi_n) = \sum_{i=1}^{n} \left[y_i \ln \frac{\pi_i}{1-\pi_i} + n_i \ln(1-\pi_i) \right] + c$$

  here rather than having the link fuction and/using a linear predictor, we have a generic π_i (probability of success for the i-th covariate pattern) with no relation with the covariate. In this way each probability in each covariate pattern group is basically independent from the other

- the resulting Maximum likelihood estimate of π_i will be obtained starting from

$$\left.\begin{array}{l} \dfrac{\partial}{\partial \pi_i} l(\pi_1, \ldots, \pi_n) = \dfrac{y_i - n_i \pi_i}{\pi_i(1-\pi_i)} \\ \dfrac{\partial^2}{\partial \pi_i^2} l(\pi_1, \ldots, \pi_n) = \dfrac{-y_i + 2y_i\pi_i - n_i\pi_i^2}{\pi_i^2(1-\pi_i)^2} \end{array}\right\} \Rightarrow \hat{\pi}_i = \frac{y_i}{n_i} = p_i, \ i = 1, \ldots, n$$

  the estimate probability of success for the saturated models $\hat{\pi}_i$ coincides with just the observed relative frequency (ratio between observed success over trial) in each covariate pattern

- thus the estimated expected value for the number of successes will be the number of trials per the relative frequency, which is the observed number of successes

$$\hat{\mathrm{E}}\left[Y_i | \mathbf{x}_i\right] = n_i p_i = y_i, \quad i = 1, \ldots, n$$

Remark 156. Again the saturated model reproduce exactly the information available in the data in their covariate pattern data structure

14.5.3 Evaluation of the log-likelihood function for saturated models

Important remark 116. The likelihood of the saturated model works as benchmark; evaluated using its ML estimate will be:

$$\begin{aligned} l(p_1, \ldots, p_n) &= \sum_{i=1}^{n} \left[y_i \ln \frac{\frac{y_i}{n_i}}{1-\frac{y_i}{n_i}} + n_i \ln\left(1 - \frac{y_i}{n_i}\right) \right] \\ &= \sum_{i=1}^{n} \left[y_i \ln \frac{y_i}{n_i - y_i} + n_i \ln \frac{n_i - y_i}{n_i} \right] \qquad (14.2) \\ &= \sum_{i=1}^{n} \left[y_i \ln \frac{\hat{\mathrm{E}}\left[Y_i | \mathbf{x}_i\right]}{n_i - \hat{\mathrm{E}}\left[Y_i | \mathbf{x}_i\right]} + n_i \ln \frac{n_i - \hat{\mathrm{E}}\left[Y_i | \mathbf{x}_i\right]}{n_i} \right] \end{aligned}$$

where 14.2 is the actual computation formula (the last is more an "interpretation")

Remark 157. What might happen in real situation is that for some covariate pattern one can have 0 or n_i observed successes:

- in case $y_i = 0 \implies \hat{\pi}_i = 0$ we would have $0 \ln \frac{0}{n_i}$ which is not defined: we replace it with 0;
- in case $y_i = n_i \implies \hat{\pi}_i = 1$ we replace $n_i \ln \frac{n_i}{0} + n_i \ln \frac{0}{n_i}$, undefined, with 0

R does this for us by default.

Important remark 117. In general the saturated model maximum likelihood will be the *Maximum possible value for the log-likelihood associated with binomial models for* $\mathbf{Y}|\mathbf{X}$*, given the observed sample* $\mathbf{y}$; therefore *any logistic regression model for* $\mathbf{Y}|\mathbf{X}$*, where we introduce a math expression linking x and y, shows a maximum value for the log-likelihood that is smaller than that value, given the observed sample* $\mathbf{y}$. This happens because we're introducing restrictions.
To evaluate the closeness of the fitted model to the saturated one again what we do is to consider the difference between the loglikelihoods

14.5.4 Deviance comparison for a logistic regression model

Remark 158. As done for poisson we can quantify the distance between the saturated and fitted model.

$$
\begin{aligned}
D &= 2\ln\left[\frac{L(p_1,\dots,p_n)}{L\left(\hat{\mathbf{b}}\right)}\right] = 2\left[l(p_1,\dots,p_n) - l\left(\hat{\mathbf{b}}\right)\right] \\
&\overset{(1)}{=} 2\left\{\sum_{i=1}^{n}\left[y_i\ln\frac{y_i}{n_i-y_i} + n_i\ln\frac{n_i-y_i}{n_i}\right] - \sum_{i=1}^{n}\left[y_i\ln\frac{\hat{m}_i}{n_i-\hat{m}_i} + n_i\ln\frac{n_i-\hat{m}_i}{n_i}\right]\right\} \\
&= 2\left\{\sum_{i=1}^{n}\left[y_i\ln y_i - y_i\ln(n_i-y_i) + n_i\ln(n_i-y_i) - n_i\ln n_i\right] + \right. \\
&\quad \left. -\sum_{i=1}^{n}\left[y_i\ln\hat{m}_i - y_i\ln(n_i-\hat{m}_i) + n_i\ln(n_i-\hat{m}_i) - n_i\ln n_i\right]\right\} \\
&= 2\sum_{i=1}^{n}\left\{y_i\left[\ln y_i - \ln\hat{m}_i\right] - y_i\left[\ln(n_i-y_i) - \ln(n_i-\hat{m}_i)\right] + \right. \\
&\quad \left. + n_i\left[\ln(n_i-y_i) - \ln(n_i-\hat{m}_i)\right] - n_i\left[\ln n_i - \ln n_i\right]\right\} \\
&= 2\sum_{i=1}^{n}\left[y_i\ln\frac{y_i}{\hat{m}_i} + (n_i-y_i)\ln\frac{n_i-y_i}{n_i-\hat{m}_i}\right]
\end{aligned}
$$

where in (1) the fitted model component (the second one) we obtained $\hat{m}_i$ as the estimated number of successes obtained with the fitted model, that is

$$\hat{m}_i = n_i\hat{\pi}_i = n_i\frac{\exp(\mathbf{x}_i^\top\hat{\mathbf{b}})}{1+\exp(\mathbf{x}_i^\top\hat{\mathbf{b}})}$$

In the final equation we're *comparing observed/estimated successes/failures*; for each covariate pattern:

- observed successes y_i are compared with (estimated) expected successes $\hat{m}_i$ through the quantity (similar to poisson one term)
$$y_i\ln\frac{y_i}{\hat{m}_i}$$
- observed failures $n_i - y_i$ are compared with (estimated) expected failures $n_i - \hat{m}_i$ through the quantity
$$(n_i-y_i)\ln\frac{n_i-y_i}{n_i-\hat{m}_i}$$
- for practical computation if $y_i = 0$ or $y_i = n_i$, we replace $0\ln 0$ with 0
- if observed and expected successes coincides, that is $y_i = \hat{m}_i$ (then failure coincides $n_i - y_i = n_i - \hat{m}_i$), the contribution of that covariance pattern to the deviance will be zero
$$y_i\ln\frac{y_i}{\hat{m}_i} = (n_i-y_i)\ln\frac{n_i-y_i}{n_i-\hat{m}_i} = 0$$
- can be shown that the larger $|y_i - \hat{m}_i|$, the larger will be the contribution to the deviance of the corresponding covariate pattern
$$y_i\ln\frac{y_i}{\hat{m}_i} + (n_i-y_i)\ln\frac{n_i-y_i}{n_i-\hat{m}_i}$$

Important remark 118 (On relevance of data structure). It is here that the distinction between unit level and covariate pattern level data structure/likelihood plays a fundamental role.
We can rewrite the covariate pattern contribution to residual deviance in terms of singular unit contribution, but there's no "direct link" between the two (eg by replacing the sum y_i

with the addend z_j and replacing the expected count $\hat{m}_i$ with expected probability $\hat{\pi}_j$. For the key ingredients of the likelihood comparison, considering a single covariate pattern:

$$y_i \ln \frac{y_i}{\hat{m}_i} = y_i \ln \frac{y_i}{n_i \hat{\pi}_i} = \left(\sum_{j:\eta_j=\eta_i} z_j \right) \ln \frac{\left(\sum_{j:\eta_j=\eta_i} z_j \right)}{\left(\sum_{j:\eta_j=\eta_i} \hat{\pi}_j \right)} \neq \sum_{j:\eta_j=\eta_i} \left(z_j \ln \frac{z_j}{\hat{\pi}_j} \right)$$

$$(n_i - y_i) \ln \frac{n_i - y_i}{n_i - \hat{m}_i} \neq \sum_{j:\eta_j=\eta_i} (1 - z_j) \ln \frac{1 - z_j}{1 - \hat{\pi}_j}$$

The inequalities are due to the presence/properties of logarithm (log of sum is different from sum of logarithm).
Thus overall we cannot decompose the residual deviance in the sum of single/individual contribution:

$$2 \sum_{i=1}^{n} \left[y_i \ln \frac{y_i}{\hat{m}_i} + (n_i - y_i) \ln \frac{n_i - y_i}{n_i - \hat{m}_i} \right] \neq 2 \sum_{i=1}^{N} \left[z_j \ln \frac{z_j}{\hat{\pi}_j} + (1 - z_j) \ln \frac{1 - z_j}{1 - \hat{\pi}_i} \right] \tag{14.3}$$

This has practical consequences

- while calculating $l(\beta_0, \ldots, \beta_p)$ we can choose any data structure (sample units/covariate patterns), *D must always be computed using the covariate pattern data structure (unless $n_i = 1 \ \forall i$)*;
- so irrespective of the fit (which can happen at unit or covariate pattern level), *observed and fitted values must be compared at covariate pattern level, and not at sample unit level (unless $n_i = 1, \forall i$).*

If we have more than one unit per covariate pattern, the expression on the right hand side of 14.3 is meaningless.

14.5.5 An approximation to D: pearson χ^2 statistics

Second order Taylor series expansion can be used to simplify D components near $\hat{m}_i$ (*as for Poisson regression models*):

$$y_i \ln \frac{y_i}{\hat{m}_i} \approxeq \underbrace{\hat{m}_i \ln \frac{\hat{m}_i}{\hat{m}_i}}_{0} + (y_i - \hat{m}_i) + \frac{1}{2} \frac{(y_i - \hat{m}_i)^2}{\hat{m}_i}$$

$$\begin{aligned} n_i - y_i \ln \frac{n_i - y_i}{n_i - \hat{m}_i} &\approxeq [(n_i - y_i) - (n_i - \hat{m}_i)] + \frac{1}{2} \frac{[(n_i - y_i) - (n_i - \hat{m}_i)]^2}{n_i - \hat{m}_i} \\ &\approxeq -(y_i - \hat{m}_i) + \frac{1}{2} \frac{[(n_i - y_i) - (n_i - \hat{m}_i)]^2}{n_i - \hat{m}_i} \end{aligned}$$

If we plug these two approximations in D we get back once again to a pearson χ^2 statistics (a difference of observed - expected squared over expected, both for successes and failures, for each covariate patterns):

$$\begin{aligned} D &= 2 \sum_{i=1}^{n} \left[y_i \ln \frac{y_i}{\hat{m}_i} + (n_i - y_i) \ln \frac{n_i - y_i}{n_i - \hat{m}_i} \right] \\ &\approxeq 2 \sum_{i=1}^{n} \left[(y_i - \hat{m}_i) + \frac{1}{2} \frac{(y_i - \hat{m}_i)^2}{\hat{m}_i} - (y_i - \hat{m}_i) + \frac{1}{2} \frac{[(n_i - y_i) - (n_i - \hat{m}_i)]^2}{n_i - \hat{m}_i} \right] \\ &= \sum_{i=1}^{n} \left[\frac{(y_i - \hat{m}_i)^2}{\hat{m}_i} + \frac{[(n_i - y_i) - (n_i - \hat{m}_i)]^2}{n_i - \hat{m}_i} \right] = \chi^2 \end{aligned}$$

We can think as observed/expected successes/failures organized in contingency tables, by covariate pattern, as follows. The residual deviance can be approximated by applying the Pearson χ^2 statistic to compare these two contingency tables (observed vs theoretical).

	Observed	
	Successes	*Failures*
Cov. pattern $\mathbf{x}_1$	y_1	$n_1 - y_1$
Cov. pattern $\mathbf{x}_2$	y_2	$n_2 - y_2$
$\vdots$	$\vdots$	$\vdots$
Cov. pattern $\mathbf{x}_n$	y_n	$n_n - y_n$

	Expected	
	Successes	*Failures*
Cov. pattern $\mathbf{x}_1$	$\hat{m}_1$	$n_1 - \hat{m}_1$
Cov. pattern $\mathbf{x}_2$	$\hat{m}_2$	$n_2 - \hat{m}_2$
$\vdots$	$\vdots$	$\vdots$
Cov. pattern $\mathbf{x}_n$	$\hat{m}_n$	$n_n - \hat{m}_n$

We can make a further step elaborating the Pearson χ^2 statistics to express it in a more com-

pact way:

$$\begin{aligned}
\chi^2 &= \sum_{i=1}^{n}\left[\frac{(y_i-\hat{m}_i)^2}{\hat{m}_i}+\frac{[(n_i-y_i)-(n_i-\hat{m}_i)]^2}{n_i-\hat{m}_i}\right] \\
&= \sum_{i=1}^{n}\left[\frac{(y_i-\hat{m}_i)^2}{\hat{m}_i}+\frac{(y_i-\hat{m}_i)^2}{n_i-\hat{m}_i}\right] = \sum_{i=1}^{n}\left[\frac{(y_i-n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i}+\frac{(y_i-n_i\hat{\pi}_i)^2}{n_i(1-\hat{\pi}_i)}\right] \\
&= \sum_{i=1}^{n}\frac{(y_i-n_i\hat{\pi}_i)^2-(y_i-n_i\hat{\pi}_i)^2\hat{\pi}_i^2+(y_i-n_i\hat{\pi}_i)^2\hat{\pi}_i^2}{n_i\hat{\pi}_i(1-\hat{\pi}_i)} \\
&= \sum_{i=1}^{n}\frac{(y_i-n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i(1-\hat{\pi}_i)} = \sum_{i=1}^{n}\left[\frac{y_i-n_i\hat{\pi}_i}{\sqrt{n_i\hat{\pi}_i(1-\hat{\pi}_i)}}\right]^2
\end{aligned}$$

the last term resemble

$$\left(\frac{y_i-\hat{E}\left[Y_i|\mathbf{x}_i\right]}{\sqrt{\text{Var}\left[Y_i|\mathbf{x}_i\right]}}\right)$$

this last setup of the residual deviance (observed - expected over the square root of the variance) is a common pattern found in all the model seen so far, directly (gaussian, se non sbaglio) or via/using approximation (poisson and here)

14.6 Residuals

Definition 14.6.1 (Deviance residuals)**.** It's defined as the squared root (with sign) of the generic term of the deviance

$$e_i^D = \text{sign}(y_i-\hat{m}_i)\sqrt{2\left[y_i\ln\frac{y_i}{\hat{m}_i}+(n_i-y_i)\ln\frac{n_i-y_i}{n_i-\hat{m}_i}\right]}, \quad i=1,\dots,n$$

where

$$\text{sign}(y_i-\hat{m}_i) = \begin{cases} -1 & \text{if } y_i < \hat{m}_i \\ +1 & \text{if } y_i \geq \hat{m}_i \end{cases}$$

and is possible to prove that $\left[y_i\ln\frac{y_i}{\hat{m}_i}+(n_i-y_i)\ln\frac{n_i-y_i}{n_i-\hat{m}_i}\right]$ is always ≥ 0 (so sqrt make is doable)

Important remark 119. We have that $\sum_{i=1}^{n}\left(e_i^D\right)^2 = D$.

Definition 14.6.2 (Pearson residuals)**.** Defined as the squared root (with sign) of the generic term of the Pearson χ^2 statistic

$$\begin{aligned}
e_i^P &= \frac{y_i-n_i\hat{\pi}_i}{\sqrt{n_i\hat{\pi}_i(1-\hat{\pi}_i)}} \quad i=1,\dots,n \\
&= \frac{y_i-\hat{\text{E}}\left[Y_i|\mathbf{x}_i\right]}{\sqrt{\hat{\text{Var}}\left[Y_i|\mathbf{x}_i\right]}} \\
&= \frac{p_i-\hat{\text{E}}\left[P_i|\mathbf{x}_i\right]}{\sqrt{\hat{\text{Var}}\left[P_i|\mathbf{x}_i\right]}}
\end{aligned}$$

From the last twos, it can be read both focusing on number of successes or relative frequency of successes

Important remark 120. We have that $\sum_{i=1}^{n}\left(e_i^P\right)^2 = \chi^2$.

Proposition 14.6.1 (Residuals properties)**.** *Same as the poisson If a logistic regression model is adequate (correctly specified):*

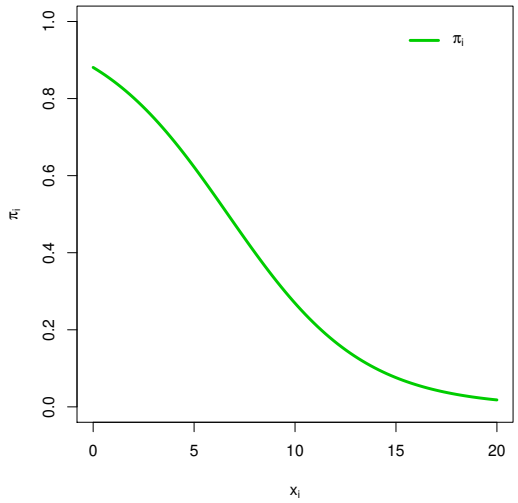


Figure 14.2: Sparse residuals example

- *both deviance and Pearson residuals have null conditional expected value*

$$\mathrm{E}\left[e_i^D \middle| \mathbf{x}_i\right] \approxeq \mathrm{E}\left[e_i^P \middle| \mathbf{x}_i\right] \approxeq 0$$

- *the residual variance*

$$\mathrm{Var}\left[e_i^D \middle| \mathbf{x}_i\right] \approxeq \mathrm{Var}\left[e_i^P \middle| \mathbf{x}_i\right] \approxeq 1 - H_{ii}$$

  is not constant but it's dependent on the quantity H_{ii} which is the $l'i$-th element of the main diagonal of the matrix (somewhat related to the hat matrix):

$$\mathbf{H} = \mathbf{W}^{\frac{1}{2}}\mathbf{X}\left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{W}^{\frac{1}{2}}$$

- *asymptotically, deviance and Pearson residuals are equivalent and distributed as independent Gaussian random variables.*
  ***WARNING**: the asymptotic properties of deviance and Pearson residuals hold if n is considered fixed and the number of units for each covariate pattern goes to ∞: $n_i \longrightarrow \infty$ $\forall i$ (fixed cell asymptotics). So* here is not enough *to have an overall number of units which is large: it must be large within each covariate pattern.*

14.7 Residuals analysis when data are sparse

Definition 14.7.1 (Data sparsity)**.** In the context of regression models for binary outcomes, the expression "data sparsity" denotes situations in which n_i is small for any covariate pattern.

Remark 159. This is very common if there is at least a single numeric covariate.

14.7.1 A simulation example

The prof generated the data starting from the model with a single regressor (depicted in figure 14.2)

$$\pi_i = \frac{\exp(2 - 0.3x_i)}{1 + \exp(2 - 0.3x_i)}, \quad i = 1, \ldots, n$$

using alternatively

- 100 covariate patterns (chosen *100 different values for the regressor*) $\{x_i, i = 1, \ldots, 100\}$

- 3 possible numnerosity for each covariate pattern for n_i: 1, 5, 20
- three observed samples are simulated considering three total sample sizes $\sum_{i=1}^{n} n_i = N$ (equal to 100, 500 and 2000) using a binomial with as π_i $(j = 1, \ldots, n)$ the values obtained from $\pi_i = \frac{\exp(2-0.3x_i)}{1+\exp(2-0.3x_i)}$
- for each observed samples,a logistic regression model is fitted (*no mispecification*)

The:

- data on relative frequencies are reported in fig 14.3: each relative frequency can take up to $n_i + 1$ unique values $(0, \frac{1}{n_i}, \ldots, \frac{n_i - 1}{n_i}, 1)$.
  We see that as n_i increases, the more the observed relative frequencies become closer and closer to the true/theoretical ones;
- estimated coefficients are in the following (remebering $\beta_0 = 2$ and $\beta_1 = -0.3$):

n_i	$\hat{b}_0$	$\hat{b}_1$
1	1.858	-0.259
5	2.191	-0.307
20	2.078	-0.302

  All the three estimated models provide satisfactory results (the estimated probabilities are very close to the true ones); can be considered adequate. ML estimated vs real function is reported in fig 14.4; as the total sample size increases, estimated probabilities becomes closer and closer to real ones
- pearson residuals vs (estimated) linear predictors are reported in fig 14.5; despite the adequacy of the three models, these plots seems to show anomalous patterns
  - residuals lies on curves, whose number depends on n_i
  - the anomaly seems to become less evident as n_i increases

  Despite the three models are adequate, the residuals lies on curves. The number of curves appearing in the plot depends on the number of unit per each covariate pattern (we have $n_i + 1$ lines)
- quantile quantile plots are in fig 14.6: the anomalous patterns emerging in the residual/fitted plot have an impact also on the quantile quantile plots; residual becomes better/more gaussian as n_i increases;
- finally, to rule out the relevance of sample size N (at least compared to within covariate pattern sample size, n_i) another simulation was done in an extreme case increasing number of covariate pattern while keeping numerosity for each $n_i = 1$, $N = 2000$ (fig 14.7, this plots should be compared with other plots using $N = 2000$ but higher covariate pattern numerosity): here,
  - the difference between estimated and theorical in a) is null we get estimated betas and probability perfectly overlapping (for ML, it doesnt matter if we have many observation per covariate pattern, as long as sample size is high enough)
  - the anomaly in the residuals (the b) figure is still composed of two curves and we have a strange behaviour in c) as well) does not seem to be related with the overall sample size, but it is connected with the number of sample units associated with each covariate pattern

  In this extreme case we can also derive the equation for these two curves

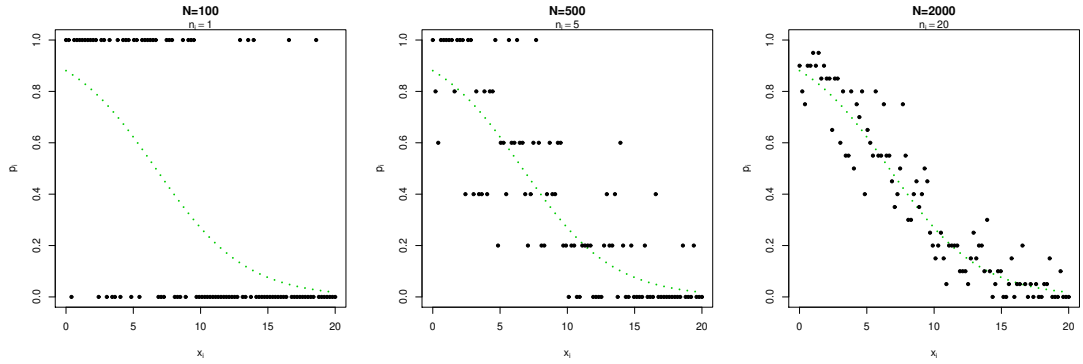
14.7.2 Data sparsity and curves on residuals vs fitted plot

As said in limiting situations (where $n_i = 1, \forall i$) we can have a formula describing curves on the residuals vs fitted plot.

Proposition 14.7.1. *The pearson residuals (as function of linear predictor) will take one of two possible values (for each covariate pattern):*

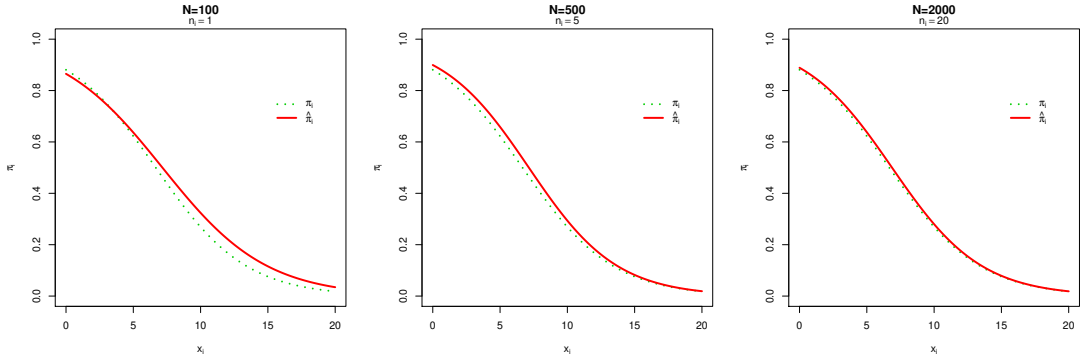
$$\frac{y_i - \hat{\pi}_i}{\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}} = \begin{cases} -\sqrt{\frac{\hat{\pi}_i}{1-\hat{\pi}_i}} = -\sqrt{\exp\left(\mathbf{x}_i^\top \hat{\mathbf{b}}\right)} & \text{if } y_i = 0 \\ \sqrt{\frac{1-\hat{\pi}_i}{\hat{\pi}_i}} = \sqrt{\exp\left(\mathbf{x}_i^\top \hat{\mathbf{b}}\right)^{-1}} & \text{if } y_i = 1 \end{cases}$$

These values lie on two curves, both being continuous functions in $\hat{\eta}_i = \mathbf{x}_i^\top \hat{\mathbf{b}}$.



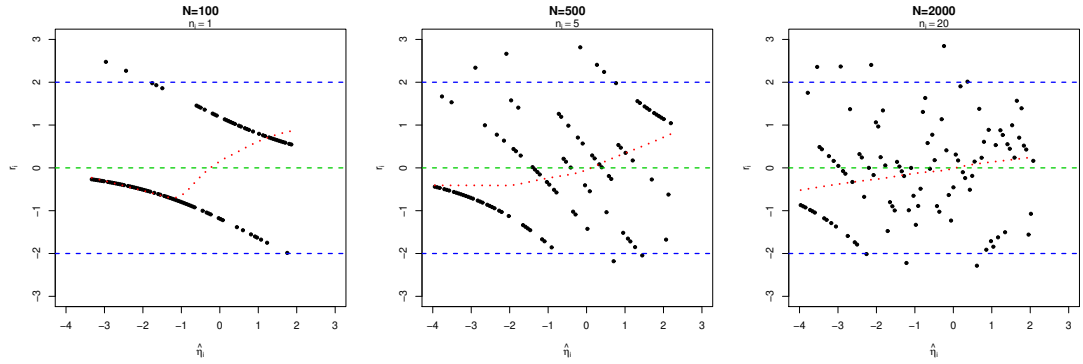
(a) (b) (c)

Figure 14.3: Prof generated data: observed relative frequencies for each covariate pattern



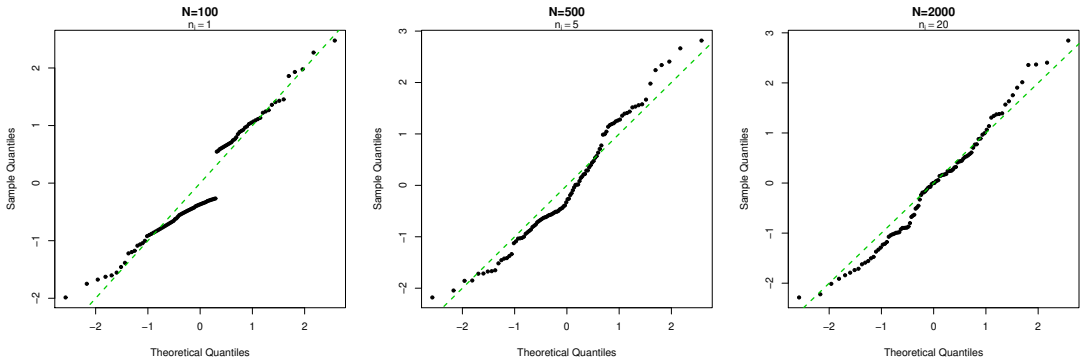
(a) (b) (c)

Figure 14.4: Estimated logistic vs real one



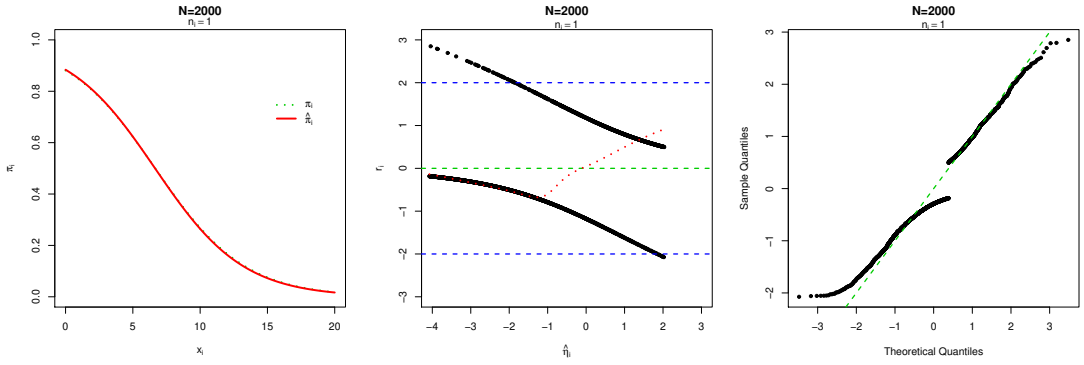
(a) (b) (c)

Figure 14.5: Residual plots: residuals vs (estimated) linear predictors



(a) (b) (c)

Figure 14.6: QQ plots



(a) (b) (c)

Figure 14.7: An extreme situation: $n_i = 1$, $N = 2000$

Important remark 121. The behaviour (the residuals lying on two curves in case of 1 obs per covariate pattern) is not related to the adequacy of the fitted model, which can be adequate or not

Important remark 122. Similar conclusions can be drawn when $n_i > 1$ ($n_i + 1$ curves can be identified); this sparsity problem arises also with deviance residuals

14.7.3 Residuals and the central limit theorem

Why we need a large number of units per covariate pattern to have residuals that are well behaved?
Remember that Pearson residuals are observed successes minus expected number of successes divided by the squared root of the expected variance. The number of successes y_i in our context is the sum of independent and identically distributed bernoulli random variable.
Recall that, when $Z_1, \ldots, Z_{n_i}$ are iid Bernoulli rvs (that is $Z_j \sim \text{Ber}(\pi_i)$ are independent for $j = 1, \ldots, n_i$) then, if the element of the sums goes to infinity $n_i \to \infty$ we can get an approximation of the distribution of the sum through the central limit theorem

$$\frac{\sum_{j=1}^{n_i} Z_j - \text{E}\left[\sum_{j=1}^{n_i} Z_j\right]}{\sqrt{\text{Var}\left[\sum_{j=1}^{n_i} Z_j\right]}} = \frac{Y_i - n_i\pi_i}{\sqrt{n_i\pi_i(1-\pi_i)}} \xrightarrow{d} \text{N}\left(()0, 1\right)$$

If we consider the standardized version as the $n_i \to \infty$ we have that the sum converges to standard gaussian.
This is the idea (behind the proof) of why we need many observation for each covariate pattern in order to get residuals that are well behaved asymptotically.

14.7.4 Aggregated Pearson residuals

Remark 160. How can we deal in situation where we do not have sufficient number of observation for each covariate pattern?

Important remark 123. In presence of data sparsity, Pearson (and deviance) residuals do not provide any valuable information about the adequacy of a logistic regression model.

An alternative definition of residual can be used, based on aggregating sample units in homogeneous groups, leading to the so-called *aggregated Pearson residuals*; the idea is to group units having approximately the same covariate pattern/probability of success and threat them as they effectly were of the exact same covariate pattern. Precisely:

- the estimated values $\hat{\eta}_i$ (or, equivalently the estimated probabilities $\hat{\pi}_i$) can be used for defining such homogeneus groups: sample units showing similar $\hat{\eta}_i$ are assigned to the same group;
- the range of values for $\hat{\eta}_i$ can be split into G sub-intervals, according to $G-1$ thresholds: *these threshold should be chosen according to the quantiles of* $\hat{\eta}_i$ *in order to obtain groups with similar sizes*;
- when $n_i > 1$, all sample units showing the i-th covariate pattern are assigned to the same group;
- then within each of the G group we can compute the *aggregated Pearson residuals* as:

$$\bar{e}_l^P = \frac{y_l - n_l\bar{\pi}_l}{\sqrt{n_l\bar{\pi}_l(1-\bar{\pi}_l)}}, \quad l = 1, \ldots, G$$

  with:

  - y_l the observed successes for sample units assigned to the $l-$th group
  - $n_l \approxeq \dfrac{N}{G}$ the number of sample units assigned to the $l-$th group
  - $\bar{\pi}_l$ the (weighted) average of the estimated probabilities $\hat{\pi}_i$ for sample units aassigned to the $l-$th group (the corresponding n_i are used as weights)
  - $\bar{\eta}_l$ the (weighted) average of linear predictor/estimated values $\hat{\eta}_i$ for sample units assigned to the $l-$th group (the corresponding n_i are used as weights)

So this is just the definition of the pearson residual; the only difference is that rather than computing the residual foreach covariate pattern, we're computing it after re-grouping the units for covariate pattern similarity

Example 14.7.1 (Aggregated Pearson residuals - $n_i = 1,\ N = 2000,\ G = 10$)**.** Let's consider the situation where we have 2000 units, each characterized by a different covariate pattern and we choose $G = 10$ (so each covariate pattern group will have 200 units)

Intervals for $\hat{\eta}_i$	n_l	y_l	$\bar{\pi}_l$ (average $\hat{\pi}_i$)	Pearson aggregated residual $\bar{e}_l^P$	$\bar{\eta}_l$ (average $\hat{\eta}_i$)
$(-\infty$,-3.46]	200	4	0.024	-0.406	-3.770
(-3.46,-2.85]	200	5	0.044	-1.290	-3.160
(-2.85,-2.24]	200	15	0.077	-0.090	-2.549
(-2.24,-1.63]	200	24	0.131	-0.474	-1.938
(-1.63,-1.02]	200	39	0.216	-0.708	-1.328
(-1.02,-0.412]	200	74	0.333	1.112	-0.717
(-0.412,0.198]	200	105	0.475	1.407	-0.106
(0.198,0.809]	200	117	0.622	-1.073	0.504
(0.809,1.42]	200	154	0.749	0.683	1.115
(1.42,$+\infty$)	200	163	0.844	-1.145	1.725

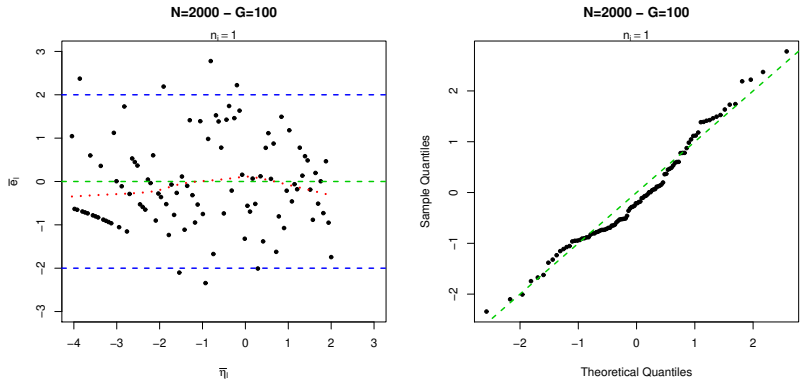
The first column shows the values of the intervals computed on the basis of linear predictor (probability of success will be a monotone so quantiles on the two will lead to the same group); the total number of successes in each group y_l (as well as the $\overline{\pi}_l$) tends to increas with the value of the linear predictor. More or less in the first line we have that $0.024 \approxeq 4/200$ (but it's not the same). Then we have the pearson aggregated residual and the mean linear predictor per group.
We could us a larger G, eg if we want to plot the aggregated residuals: we do it in the following example.

Example 14.7.2 (Aggregated Pearson residuals - $n_i = 1,\ N = 2000,\ G = 100$)**.** If we set $G = 100$ we get 100 groups each with 20 units (which is a relative large number of units); in figure 14.8 we see the behaviour of the pearson aggregated residuals this way obtained.
If we compare this plots with the one obtained at the beginning of the simulation (with 20 natural covariate pattern, fig 14.5 c) they are basically indistinguishable.
The curves based pattern seems less evident after the aggregation: looking the residuals in this aggregated way we have one more tools to judge. Here the plot suggests that the model can be considered adequate: the points are evenly scattered around zero with local average somewhat close to 0; the qqplot shows points not far from the straight line.

Important remark 124. Two important things to remember:

- role of having a large number of units for each covariate pattern when looking on adequacy on the model: the covariate level data structure plays a fundamental role when evaluating the adequacy of the model because it has an effect on the quantity that must be computed in order to judge the adequacy of the model. We have to use this structure when computing deviance, chi-square statistic and corresponding residuals

- the role of condition B) ($n_i \to \infty \forall i$) in terms of asymptotic behaviour when focusing on the residuals; as one can foresee, since the pearson/deviance residuals are nothing but the building block of pearson χ^2 and residual deviance, also the asymptotic behaviour of the two latters quantities (and the possibility of using them to evaluate the goodness of fit) will be influenced by number of units we have foreach covariate pattern, and will be well behaved/legitimate to use iff the number of units foreach covariate pattern is reasonably large.
  So we need condition B) not only for residuals but even for testing adequacy of logistic regression models.



(a) (b)

Figure 14.8: Aggregated Pearson residuals - $n_i = 1,\ N = 2000,\ G = 100$

14.8 Testing the adequacy of a logistic regression model

14.8.1 Goodness of fit test

Proposition 14.8.1. *If the systematic component of a logistic regression model is adequate (correctly specified), or, equivalently, if the following null hypothesis is true:*

$$\begin{cases} Y_i|\mathbf{x}_i \sim \text{Bin}(n_i, \pi_i), & \textit{independent } i = 1, \dots, n \\ H_0 : \text{logit}(\pi_i) = \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} \end{cases}$$

then:

- *deviance and* χ^2 *are* asymptotically equivalent *(differences between them tends to vanish):* $\chi^2 \xrightarrow{d} D$
- D *and* χ^2 *are* asymptotically independent from ML estimator $\hat{\mathbf{B}}$
- *both are asymptotically chi-square distributed with* $n - p - 1$ *df:* $D|\, H_0 \xrightarrow{d} \chi^2_{n-p-1}$ *&* $\chi^2 \big|\, H_0 \xrightarrow{d} \chi^2_{n-p-1}$

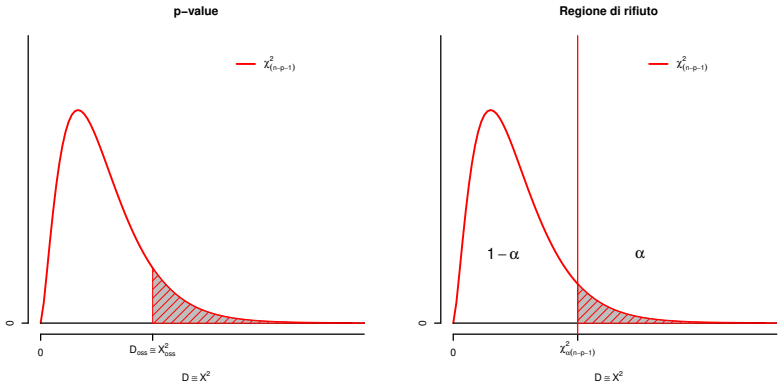
Important remark 125 (Asymptotics needed condition). **WARNING**: the asymptotic properties of the residual deviance and of the Pearson χ^2 statistic hold if n is considered fixed and $n_i \longrightarrow \infty, \forall i$ (*fixed cell asymptotics*). So practically speaking if there's no covariate pattern with low number of units.

Remark 161 (Come funziona il test). So if we have a sufficient number of unit for each covariate pattern, we can use the test and as seen for the poisson regression model (fig 14.9), and for a given significance level α

- we have p-value $< \alpha \iff D_{oss} \cong \chi^2_{oss} > \chi^2_{\alpha(n-p-1)}$; in this case H_0 rejected and so we conclude that *the model is not adequate*
- we have p-value $> \alpha \iff D_{oss} \cong \chi^2_{oss} < \chi^2_{\alpha(n-p-1)}$; here H_0 is not rejected and so we conclude that *the model is adequate*

Important remark 126 (Goodness of fit and data sparsity). In presence of data sparsity it can be proved that:

- the distribution of D and χ^2 is *unknown*, even if $N \longrightarrow \infty$ (total sample size is very large)



(a) (b)

Figure 14.9: Chi square, you know

- both D and χ^2 can take large values even if the null hypothesis H_0 is true even if the model is adequate: so aside the unknown distribution we cannot trust the value the statistics assumes
- in the extreme case of 1 unit per covariate pattern, $n_i = 1, \forall i$, D has a degenerate distribution given the ML estimate $\hat{\mathbf{b}}$: it takes a fixed value and we no long have independence between deviance and ML estimator

So there's no way of building a gof test based on these statistics an alternative goodness of fit statistic should be exploited

14.8.2 Hosmer-Lemeshow goodness of fit test

Remark 162. The solution to the sparseness GOF test is linked to the idea of grouping units based on covariate patterns, as seen in aggregated Pearson residuals, and is given by the Hosmer Lemeshow test.
The idea is to compute aggregated pearson residuals such that in each group we have a sufficient large number of units and then build the test statistics, which is nothing but the sum of the squared aggregated pearson residuals.

Proposition 14.8.2. *Given a partition of the sample units in G groups (as for the aggregated Pearson residuals) it's defined as*

$$D_{HL} = \sum_{l=1}^{G} \frac{(y_l - n_l \bar{\pi}_l)^2}{n_l \bar{\pi}_l (1 - \bar{\pi}_l)} = \sum_{l=1}^{G} \left(\bar{e}_l^P\right)^2$$

Hosmer and Lemeshow proved that, if the systematic component of a logistic regression model is adequate (correctly specified), or equivalently, the following null hypothesis is true:

$$\begin{cases} Y_i | \mathbf{x}_i \sim \text{Bin}(n_i, \pi_i) \quad \textit{independent } i = 1, \dots, n \\ H_0 : \text{logit}(\pi_i) = \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} \end{cases}$$

then the statistics is asymptotically chi-square distributed with $G-2$ df, that is $D_{HL} | H_0 \xrightarrow{d} \chi^2_{G-2}$ as $N \to \infty$ (so we circumvent the problem of sparsity here).

Remark 163. Hosmer and Lemeshow suggest to set $G = 8$ or $G = 10$.

14.9 Interpretation

14.9.1 Interpretation of model parameters

Once evaluated the adequateness of the model we can procede in interpreting the parameters. Since in general we have:

$$\pi_i = \frac{\exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi})}{1 + \exp(\beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi})}$$

and thus looking at first partial derivative

$$\frac{\partial}{\partial x_{ji}} \pi_i = \beta_j \frac{\exp(\eta_i)}{[1 + \exp(\eta_i)]^2}$$

each regressor has a *non-linear effect* (since we're using the logistic function) on π_i. Similarly to what we've seen for poisson model:

- the *direction of the change* in π_i due to a unit increase in x_{ji} depends on the *sign* of β_j
- the *magnitude* of the change depends *also* on the *values of all the other regressors and regression coefficients*

With a bit of manipulation we can find the intepretation for β (or $\exp(\beta)$). We have that

- odds *before* a unit increase in x_{ji}

$$\frac{\pi_i}{1 - \pi_i} = \exp(\beta_0) \cdot \ldots \cdot \exp(\beta_j x_{ji}) \cdot \ldots \cdot \exp(\beta_p x_{pi})$$

- odds *after* a unit increase in x_{ji}

$$\frac{\pi_{i+}}{1 - \pi_{i+}} = \exp(\beta_0) \cdot \ldots \cdot \exp[\beta_j (x_{ji} + 1)] \cdot \ldots \cdot \exp(\beta_p x_{pi})$$

- Multiplicative change due to a unit increase in x_{ji}

$$\frac{\pi_{i+}}{1 - \pi_{i+}} = \exp(\beta_j) \frac{\pi_i}{1 - \pi_i}$$

  We say logistic model is multiplicative with respect to the odds of π_i $\dfrac{\pi_i}{1 - \pi_i}$ (or it's *linear* with respect to the logit of π_i)

- Odds ratio associated with a unit increase in x_{ji}

$$\frac{\dfrac{\pi_{i+}}{1 - \pi_{i+}}}{\dfrac{\pi_i}{1 - \pi_i}} = \exp(\beta_j) = \begin{cases} < 1 & \text{if } \beta_j < 0 \\ = 1 & \text{if } \beta_j = 0 \\ > 1 & \text{if } \beta_j > 0 \end{cases}$$

14.9.2 Change in the coding scheme for the dependent variable

What happens if we change the coding scheme for the dependent variable? (eg 1 = absence of a characteristic).
If we define $z_j^* = 1 - z_j$, for $j = 1, \ldots, N$ this reflects on

$$\begin{aligned} \pi_j^* &= \Pr(z_j^* = 1) = \Pr(\texttt{disease}_j = \texttt{absent}) = 1 - \Pr(\texttt{disease}_j = \texttt{present}) \\ &= 1 - \Pr(z_j = 1) = 1 - \pi_j \end{aligned}$$

We can show that once estimated the model we have that the new coefficient $\boldsymbol{\beta}^*$ are the just the old with reversed sign, that is $\boldsymbol{\beta}^* = -\boldsymbol{\beta}$. We have:

$$\begin{aligned}
\pi_j^* = \frac{\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}^*\right)}{1+\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}^*\right)} &= \Pr(z_j^* = 1) = \Pr(z_j = 0) \\
&= 1 - \Pr(z_j = 1) = 1 - \frac{\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)}{1+\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)} \\
&= \frac{1+\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right) - \exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)}{1+\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)} = \frac{1}{1+\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)} \\
&= \frac{\exp\left(-\mathbf{x}_j^\top \boldsymbol{\beta}\right)}{\exp\left(-\mathbf{x}_j^\top \boldsymbol{\beta}\right)} \frac{1}{1+\exp\left(\mathbf{x}_j^\top \boldsymbol{\beta}\right)} \\
&= \frac{\exp\left(-\mathbf{x}_j^\top \boldsymbol{\beta}\right)}{1+\exp\left(-\mathbf{x}_j^\top \boldsymbol{\beta}\right)}
\end{aligned}$$

So looking at the equivalence posed we conclude $\boldsymbol{\beta}^* = -\boldsymbol{\beta}$. For what concerns the loglik, we obtain the same amount since

$$\begin{aligned}
l^*\left(\beta_0^*, \ldots, \beta_p^*\right) &= \sum_{j=1}^N \left\{ z_j^* \ln \frac{\pi_j^*}{1-\pi_j^*} + \ln\left[1-\pi_j^*\right] \right\} = \sum_{j=1}^N \left\{ (1-z_j) \ln \frac{1-\pi_j}{\pi_j} + \ln\left[\pi_j\right] \right\} \\
&= \sum_{j=1}^N \left\{ -z_j \ln \frac{1-\pi_j}{\pi_j} + \ln \frac{1-\pi_j}{\pi_j} + \ln\left[\pi_j\right] \right\} \\
&= \sum_{j=1}^N \left\{ z_j \ln \frac{\pi_j}{1-\pi_j} + \ln\left[1-\pi_j\right] \right\} \\
&= l(\beta_0, \ldots, \beta_p)
\end{aligned}$$

So changing the dummy coding for the dependent variable change sign to the estimated beta but leave the loglikelihood unchanged

14.10 Linear hypotheses testing

Remark 164. Again we have the general LRT and the wald test; here we repeat stuff said for poisson regreesion models

14.10.1 General linear hypotheses on β

Important remark 127. Considering the null $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{t}$ we can implement the following *asymptotically equivalent* tests:

- likelihood ratio test statistic:

$$2 \ln \frac{L(\hat{\mathbf{b}})}{L(\hat{\mathbf{b}}_{H_0})} \Bigg| H_0 \xrightarrow{d} \chi_q^2$$

- the (approximation) Wald test statistic:

$$\left[\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right]^\top \left[\mathbf{K}\widehat{I(\boldsymbol{\beta})}^{-1}\mathbf{K}^\top\right]^{-1} \left[\mathbf{K}\hat{\mathbf{b}} - \mathbf{t}\right] \Bigg| H_0 \xrightarrow{d} \chi_q^2$$

Important remark 128. The asymptotic properties hold if just $N = \sum_i n_i \longrightarrow \infty$ (and thus even if $n_i \longrightarrow \infty \; \forall i$); so *data sparsity does not affect the asympotic behaviour of these two statistics for logistic regression models*

14.10.2 Single coefficient hypothesis $H_0 : \beta_h = 0$ $(h = 1, \dots, p)$

When focusing on hypotheses regarding a single coefficient, the Wald test statistic simplifies to the ratio of estimated coefficient over the sqrt of its asymptotic estimated variance (or equivalently its squared version):

$$\left. \frac{b_h}{\sqrt{\widehat{I(\boldsymbol{\beta})}^{-1}_{h+1,h+1}}} \right| H_0 \xrightarrow{d} N(0,1) \qquad \left. \frac{b_h^2}{\widehat{I(\boldsymbol{\beta})}^{-1}_{h+1,h+1}} \right| H_0 \xrightarrow{d} \chi^2_1$$

where $\widehat{I(\boldsymbol{\beta})}^{-1}_{h+1,h+1}$ is the $h+1-$th element on the main diagonal of $\widehat{I(\boldsymbol{\beta})}^{-1}$ ($+1$ because the first element on the main diagonal refers to β_0).
This is what R reports in the coefficient table.

14.11 Models comparison

As in hypotheses testing, no particular news with respect to poisson regression models.

14.11.1 Choice among logistic regression models

Our problem is choose the most adequate logistic regression model for a given random sample $\mathbf{Y}$. Supposing $Y_i|\mathbf{x}_i \sim \text{Bin}(n_i, \pi_i)$, independent $(i = 1, \dots, n)$ we analyze the simplest situation where we have two candidate models

$$M_A : \text{logit}(\pi_i) = \eta_{Ai} = \mathbf{x}_{Ai}^\top \boldsymbol{\beta}_A, \quad \boldsymbol{\beta}_A \in \mathbb{R}^{p_A+1}$$
$$M_B : \text{logit}(\pi_i) = \eta_{Bi} = \mathbf{x}_{Bi}^\top \boldsymbol{\beta}_B \quad \boldsymbol{\beta}_B \in \mathbb{R}^{p_B+1}$$

which differs because they are characterised by *different sets of regressors*: $\mathbf{x}_{Ai} \neq \mathbf{x}_{Bi}, \forall i$ (without loss of generality: $p_A > p_B$)

14.11.2 Nested models

If the two models are nested:

- vectors $\mathbf{x}_{Bi}$ can be obtained by removing one or more that one regressor from vectors $\mathbf{x}_{Ai}$
- M_B can be obtained by introducing suitable linear constraints on the parameteres of M_A (setting them to 0):

$$\left.\begin{array}{l} M_A : \text{logit}(\pi_i) = \eta_{Ai} = \mathbf{x}_{Ai}^\top \boldsymbol{\beta}_A \\ H_0 : \mathbf{K}_B \boldsymbol{\beta}_A = \mathbf{t}_B \end{array}\right\} \quad \Rightarrow \quad M_B : \text{logit}(\pi_i) = \eta_{Bi} = \mathbf{x}_{Bi}^\top \boldsymbol{\beta}_B$$

  where:

  - $q = p_A - p_B$ are the number of regressors excluded from M_A to obtain M_B
  - $\mathbf{K}_B$ is the $(q) \times (p_A + 1)$ matrix with 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
  - $\mathbf{t}_B = \mathbf{0}_q$

Definition 14.11.1 (LRT)**.** A likelihood ratio test can be exploited to choose among M_A and M_B. In particular:

- such test can be expressed as a function of the two corresponding deviances

$$\Delta l = 2 \ln \left[\frac{L\left(\hat{\mathbf{b}}_A\right)}{L\left(\hat{\mathbf{b}}_{A|H_0},\right)} \right] = 2 \ln \left[\frac{L\left(\hat{\mathbf{b}}_A\right)}{L\left(\hat{\mathbf{b}}_B\right)} \right] = D(M_B) - D(M_A) = \Delta D$$

- under the hypothesis H_0 is true (if M_B is as "adequate" as M_A) then $\Delta D | M_B \xrightarrow{d} \chi^2_q$

- the asymptotic properties of ΔD at the previous point hold if $N = \sum_i n_i \longrightarrow \infty$ (it is not necessary that $n_i \longrightarrow \infty \; \forall i$, thus *data sparsity does not affect the asympotic properties of differences between residual deviances of logistic regression models*

Remark 165. We otherwise can implement the asymptotically equivalent Wald test

Definition 14.11.2 (Wald test)**.** Can be also exploited to choose among M_A and M_B. Such test can be expressed as a function of the ML estimator for $\boldsymbol{\beta}_A$ In particular:

$$\left[\mathbf{K}_B\hat{\mathbf{b}}_A - \mathbf{t}_B\right]^\top \left[\mathbf{K}_B \widehat{I(\boldsymbol{\beta}_A)}^{-1}\mathbf{K}_B^\top\right]^{-1} \left[\mathbf{K}_B\hat{\mathbf{b}}_A - \mathbf{t}_B\right] \Bigg| M_B \xrightarrow{d} \chi^2_q$$

And asymptotic properties holding if $N = \sum_i n_i \longrightarrow \infty$ (again here is not necessary that $n_i \longrightarrow \infty \; \forall i$ and so *data sparsity does not affect the asympotic properties of the Wald test statistic for logistic regression models*

14.11.3 Non-nested models

If the models are non-nested:

- vectors $\mathbf{x}_{Bi}$ cannot be obtained by removing one or more that one regressor from vectors $\mathbf{x}_{Ai}$
- Model M_B can be obtained by simultaneously excluding some (or all) regressors in model M_A and adding some regressors to model M_A
- *The two models are characterised by two sets of regressors that are only partially overlapping, or non-overlapping*
- the differences between the two deviances does not have a known random distribution, and thus a likelihood ratio test cannot be used to choose between the two models, so we have to rely on something else

Again we use AIC and BIC to choose among models, which are defined like:

$$\begin{aligned} AIC &= -2\ln L\left(\hat{\mathbf{b}}\right) + 2(p+1) \\ &= -2\sum_{i=1}^{n}\left\{ y_i(\mathbf{x}_i^\top\hat{\mathbf{b}}) + n_i \ln\left[1 - \frac{\exp\left(\mathbf{x}_i^\top\hat{\mathbf{b}}\right)}{1+\exp\left(\mathbf{x}_i^\top\hat{\mathbf{b}}\right)}\right]\right\} - 2\sum_{i=1}^{n}\ln\binom{n_i}{y_i} + 2(p+1) \\ BIC &= -2\ln L\left(\hat{\mathbf{b}}\right) + \ln(N)\cdot(p+1) \\ &= -2\sum_{i=1}^{n}\left\{ y_i(\mathbf{x}_i^\top\hat{\mathbf{b}}) + n_i \ln\left[1 - \frac{\exp\left(\mathbf{x}_i^\top\hat{\mathbf{b}}\right)}{1+\exp\left(\mathbf{x}_i^\top\hat{\mathbf{b}}\right)}\right]\right\} - 2\sum_{i=1}^{n}\ln\binom{n_i}{y_i} + \ln(N)\cdot(p+1) \end{aligned}$$

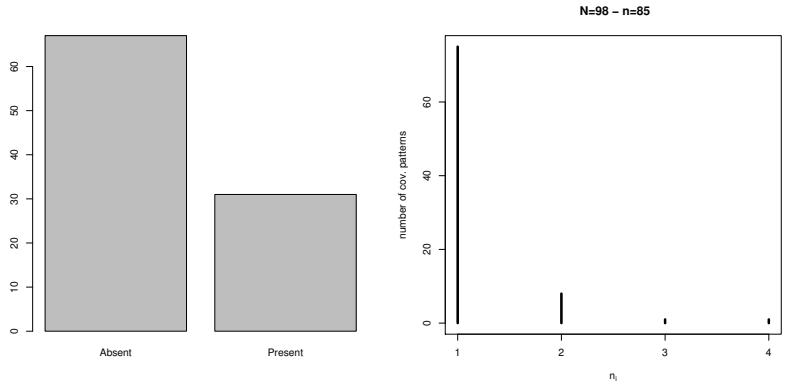
Note that the additive constant $-2\sum_{i=1}^{n}\ln\binom{n_i}{y_i}$ (not depending on model parameter) can be ignored when all competing models have a binomial probabilistic component.

14.12 Example: logistic regression

In a health study to investigate an epidemic outbreak of a disease that is spread by mosquitoes in a city, 98 individuals were randomly sampled. For each individual, information about the following variables was collected:

- `disease`: absence/presence of specific symptoms associated with the disease
- `age`: age of the individual (years)
- `area`: sector of the city in which the individual lives (two categories: sector 1/sector 2)
- `status`: socio-economic status of the household to which the individual belongs (three categories: lower/medium/upper)

Is there a significant association between the presence of the disease syptoms and any of the regressors? we have that:



(a) (b)

Figure 14.10: Epidemiologic example: dependent variable and covariate pattern numerosity

- dependent variable - observed values are depicted in figure 14.10 a) (most non affected by the disease), while number of observations for each covariate pattern in figure b): we see that most of the covariate patterns are associated with only one sample unit (98 individuals but 85 covariate patterns) so *this data set is sparse*;
- we adopt the following dummy variable coding schemes: `disease` (absence = 0, presence = 1) thus will be

$$\pi_j = \Pr(z_j = 1) = \Pr(\texttt{disease}_j = \texttt{present}), \quad j = 1, \ldots, N$$

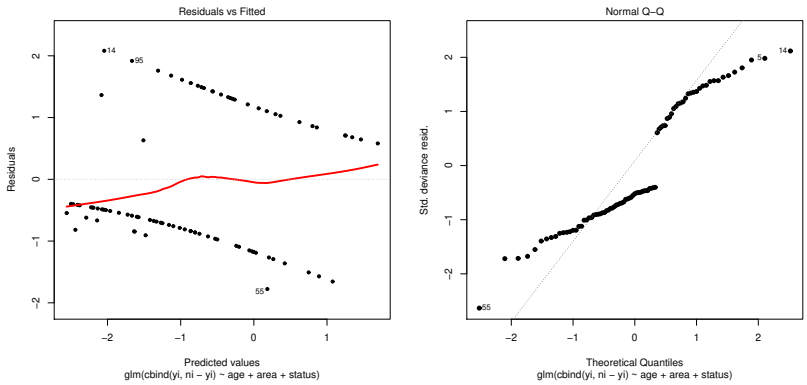
  Furthermore `area` (sector 1 = 0, sector 2 = 1), `status` (using lower socio economic status as reference)
- usign data in covariate pattern data structure, the estimated model `Call:`

```
glm(formula = cbind(yi, ni - yi) ~age + area + status, family = "binomial", data
= disease.cov)
Coefficients:
                Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)      -2.618          0.613    -4.270      0.000
 age               0.030          0.014     2.203      0.028
 areaSect2         1.575          0.502     3.139      0.002
 statusMiddle      0.714          0.654     1.092      0.275
 statusUpper       0.305          0.604     0.505      0.613
   Null deviance:  115.726 on 84 degrees of freedom

Residual deviance:  94.462 on 80 degrees of freedom
```

  Here:

  - the null deviance (which would be the deviance of the model fitted using only the intercept) corresponds, *up to a constant*, to minus twice the maximized log-likelihood for a logistic regression model that contains only the intercept (without regressors)
  - the residual deviance (which would be the actual model deviance) corresponds, up to a constant, to minus twice the maximized log-likelihood of the fitted model
- otherwise using the sample unit data structure `Call:`

```
glm(formula = disease ~age + area + status, family = "binomial", data = disease)
Coefficients:
```



(a) (b)

Figure 14.11: The residuals

```
                Estimate   Std.  Error   z value   Pr(>|z|)
(Intercept)       -2.618          0.613    -4.270      0.000
age                0.030          0.014     2.203      0.028
areaSect2          1.575          0.502     3.139      0.002
statusMiddle       0.714          0.654     1.092      0.275
statusUpper        0.305          0.604     0.505      0.613
   Null deviance:  122.32 on 97 degrees of freedom

Residual deviance:  101.05 on 93 degrees of freedom
```

Here note that:

- the maximum likelihood estimates (and the other related quantities) *are not affected by the particular data structure*
- the deviances obtained from the sample using this data structure *are different* from the previous one (also in the corresponding degrees of freedom)
- if we provide `glm` the data in covariate pattern data structure if will compute the deviance using the left hand side of equation 14.3; if we provide using the unit based data structure it will compute using the wrong right hand side;
- the *correct values* can be obtained only considering the covariate pattern data structure (left hand side)
- basically/unfortunately `glm` does not check if in the data there are repeated covariate pattern so we must pay attention if we intend to use the deviances for testing; in this case *this is not the case* since data are sparse and we must rely on other stuff for GOF test

- we can look at residuals in figure 14.11: however, due to data sparsity, these residuals do not provide any valuable information about the adequacy of the fitted model.
  It's however important to use the covariate pattern data structure to compute residuals, as done in this example (the pearson residual formula, observed successes - expected / sqrt estimated variance, is meaninful only applied to that data structure), otherwise we get residuals aligned along two straight lines even if there are covariate pattern with more than one unit
- for GOF test, since we have sparse data we go with Hosmer-Lemeshow test ($G = 10$, approximately 10 units each), which was constructed using the following table (here using the estimated probability but it's the same)

Intervals for $\hat{\pi}_i$	n_l	Observed			Expected		$\bar{e}_l^P$
		Absence	Presence		Absence	Presence	
[0.0718,0.0897]	10.00	10.00	0.00		9.21	0.79	0.923
(0.0897,0.111]	10.00	9.00	1.00		8.98	1.02	0.020
(0.111,0.163]	10.00	9.00	2.00		9.49	1.51	-0.428
(0.163,0.185]	9.00	8.00	1.00		7.42	1.58	0.509
(0.185,0.258]	9.00	7.00	2.00		7.01	1.99	-0.006
(0.258,0.323]	10.00	7.00	3.00		7.03	2.97	-0.023
(0.323,0.42]	10.00	3.00	6.00		5.71	3.29	-1.877
(0.42,0.513]	10.00	6.00	4.00		5.36	4.64	0.404
(0.513,0.659]	10.00	5.00	5.00		4.31	5.69	0.439
(0.659,0.845]	10.00	3.00	7.00		2.47	7.53	0.390

in terms of observed and expected presences/absences we see that the model make a good job (they are very similar, with only exception of the 7-th group, where in terms of observed data we have majority presence while in expected majority of absence).
We have that $\chi^2_{HL} = 5.327$, and $\Pr\left[\chi^2_{(8)} \geq 5.327\right] = 0.722$, so *the fitted logistic regression model can be considered adequate for the data*

- regarding the interpretation of model parameters estimated coefficient and odds ratio are reported in the following table

	b_k	$\exp(b_k)$
`age`	0.030	1.0302
`areaSect2`	1.575	4.8295
`statusMiddle`	0.714	2.0422
`statusUpper`	0.305	1.3570

Therefore we have (ignoring statistical significance for the moment):

  - the odds of an individual having contracted the disease increase by about 3.0 percent with each additional year of age, for given city sector location and socio-economic status
  - the odds of an individual from sector 2 having contracted the disease are almost five times as great as for an individual from sector 1, for given age and socio-economic status
  - the odds of an individual with middle socio-economic status having contracted the disease are almost twice times as great as for an individual with lower socio-economic status, for given age and city sector location
  - the odds of an individual with upper socio-economic status having contracted the disease are about 35 percent larger than the odds of an individual with lower socio-economic status, for given age and city sector location

- for hypothesis testing, for the linear independence hypothesis

$$H_0 : \beta_{\texttt{age}} = \beta_{\texttt{Sect2}} = \beta_{\texttt{Middle}} = \beta_{\texttt{Upper}} = 0$$

we have

  - Full model:

```
Coefficients:
                Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)      -2.618         0.613    -4.270      0.000
 age               0.030         0.014     2.203      0.028
 areaSect2         1.575         0.502     3.139      0.002
 statusMiddle      0.714         0.654     1.092      0.275
 statusUpper       0.305         0.604     0.505      0.613
   Null deviance:  115.726 on 84 degrees of freedom

Residual deviance:  94.462 on 80 degrees of freedom
AIC: 107.47
```

  - Reduced model:

```
Coefficients:
```

```
                Estimate   Std.  Error   z value   Pr(>|z|)
  (Intercept)     -0.771         0.217    -3.548      0.000
    Null deviance:  115.73 on 84 degrees of freedom

Residual deviance:  115.73 on 84 degrees of freedom
AIC: 120.73
```

Going from full to reduced model we note an increase in the residual deviance and in AIC. The likelihood ratio test for comparison is (applying `anova` function):

Model	Resid.	Df	Resid. Dev	Df	Deviance	Pr(>Chi)
disease~1		84	115.73			
disease~age+sector+status		80	94.46	4	21.26	0.0003

Calculated as:

$$2\ln\frac{L(F)}{L(R)} = -2\ln\left[L(R) - L(F)\right] = 115.73 - 94.46 = 21.26$$

We have that the null hypothesis should be rejected:

- at least one of the three regressors is significantly associated with the presence of the disease (at a significance level $\alpha = 0.01$)
- note that the degrees of freedom for this test statistic are equal to 4, since 4 regression coefficients are set equal to 0, according to H_0
- the same result (in terms of test statistic and p-value) can be obtained considering models fitted using the sample unit data structure

Just as note *when performing LRT to compare two models there's no need to have dataset covariate pattern data structure*: the LRT is a difference between residual deviances where the contribution of the saturated model cancels out from both the residual deviances

- for an overall test using the Wald test, with `lht` function (without need to fit the reduced model we just set a proper $\mathbf{K}$ and $\mathbf{t}$) otoh:

```
Hypothesis:
age = 0
areaSect2 = 0
statusMiddle = 0
statusUpper = 0


 Model 1:  restricted model
Model 2:  cbind(yi, ni - yi) ~age + area + status


    Res.Df   Df   Chisq   Pr(>Chisq)
1       84
2       80    4   16.65       0.0023
```

So the Wald test leads to similar p-value and the same conclusion, even when applied to the model fitted on the sample unit data structure.
Again it needs only that sample size is large enough, it's not affected by sparsity in the data

- for the specific hypothesis on socioeconomic status to check wheter there are significant differences between groups $H_0 : \beta_{\texttt{Middle}} = \beta_{\texttt{Upper}} = 0$
  - Full model:
    `Coefficients:`

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.618	0.613	-4.270	0.000
age	0.030	0.014	2.203	0.028
areaSect2	1.575	0.502	3.139	0.002
statusMiddle	0.714	0.654	1.092	0.275
statusUpper	0.305	0.604	0.505	0.613

```
    Null deviance:  115.726 on 84 degrees of freedom

Residual deviance:  94.462 on 80 degrees of freedom
AIC: 107.47
```

– Reduced model:

```
Coefficients:

              Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)    -2.335         0.511    -4.569      0.000
 age             0.029         0.013     2.224      0.026
 areaSect2       1.673         0.487     3.434      0.001
   Null deviance:  115.726 on 84 degrees of freedom

Residual deviance:  95.668 on 82 degrees of freedom
AIC: 104.68
```

And thus the Likelihood ratio test is:

Model	Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
disease~age+sector	82	95.67			
disease~age+sector+status	80	94.46	2	1.21	0.5474

computed as

$$2 \ln \frac{L(F)}{L(R)} = -2 \ln [L(R) - L(F)] = 95.67 - 94.46 = 1.21$$

We don't reject the null hypothesis:

– there are not significant differences in the probability of having the disease among the three categories of socio-economic status, for given age and city sector location

– note that the degrees of freedom for this test statistic are equal to 2, since 2 regression coefficients are set equal to 0, in order to exclude the socio-economic status from the full model

- for $H_0 : \beta_{\texttt{age}} = 0$

– Full model:

```
Coefficients:

                Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)      -2.618         0.613    -4.270      0.000
 age               0.030         0.014     2.203      0.028
 areaSect2         1.575         0.502     3.139      0.002
 statusMiddle      0.714         0.654     1.092      0.275
 statusUpper       0.305         0.604     0.505      0.613
   Null deviance:  115.726 on 84 degrees of freedom

Residual deviance:  94.462 on 80 degrees of freedom
AIC: 107.47
```

– Reduced model:

```
Coefficients:

                Estimate   Std.  Error   z value   Pr(>|z|)
 (Intercept)      -1.917         0.481    -3.984      0.000
 areaSect2         1.620         0.486     3.336      0.001
 statusMiddle      0.713         0.636     1.120      0.263
 statusUpper       0.478         0.583     0.820      0.412
   Null deviance:  115.726 on 84 degrees of freedom

Residual deviance:  99.612 on 81 degrees of freedom
AIC: 110.62
```

And the likelihood ratio test is

```
Model                       Resid.  Df   Resid. Dev   Df   Deviance   Pr(>Chi)
disease~sector+status            81         99.61
disease~age+sector+status        80         94.46    1       5.15     0.0233
```

$$2\ln\frac{L(F)}{L(R)} = -2\ln\left[L(R) - L(F)\right] = 99.61 - 94.46 = 5.15$$
$$\approx \left(\frac{b_{\text{age}}^2}{s^2\left[b_{\text{age}}\right]}\right) = \frac{0.03^2}{0.014^2} = 4.854$$

The statistics of lrt and (squared) wald-z are more or less similar and the p-value conclusion the same: the age of an individual has a significant effect on the probability of having the disease, for given city sector location and socio-economic status

- what happens if we change the coding scheme for the dependent variable? (setting to 1 the absence of symptoms)
  - Original coding scheme:
    ```
    Coefficients:
                   Estimate   Std.  Error   z value   Pr(>|z|)
     (Intercept)     -2.618         0.613    -4.270      0.000
     age              0.030         0.014     2.203      0.028
     areaSect2        1.575         0.502     3.139      0.002
     statusMiddle     0.714         0.654     1.092      0.275
     statusUpper      0.305         0.604     0.505      0.613
    Residual deviance:  94.462 on 80 degrees of freedom
    ```
  - Alternative coding scheme:
    ```
    Coefficients:
                   Estimate   Std.  Error   z value   Pr(>|z|)
     (Intercept)      2.618         0.613     4.270      0.000
     age             -0.030         0.014    -2.203      0.028
     areaSect2       -1.575         0.502    -3.139      0.002
     statusMiddle    -0.714         0.654    -1.092      0.275
     statusUpper     -0.305         0.604    -0.505      0.613
    Residual deviance:  94.462 on 80 degrees of freedom
    ```

So the two models are equivalent (they have the same residual deviance): the change in the coding scheme affects only the signs of the regression coefficients and the ztest statistics, but the two sided p-value is unaffected.

14.13 GLM for binary outcomes: choice of the link function

14.13.1 Setup refresher

Remark 166. Just a reminder of GLM models for binary outcomes: here we used the definition using the relative frequency associated with binomial distribution.

- Probabilistic component

$$Y_i|\mathbf{x}_i \sim \text{Bin}(n_i, \pi_i) \iff P_i = \left.\frac{Y_i}{n_i}\right| \mathbf{x}_i \sim \text{EF}(b(\pi_i) = \text{logit}(\pi_i), \phi = 1, w_i = n_i)$$

Thus

$$\text{E}\left[P_i|\mathbf{x}_i\right] = \pi_i \implies \text{E}\left[Y_i|\mathbf{x}_i\right] = n_i\pi_i$$
$$\text{Var}\left[P_i|\mathbf{x}_i\right] = \frac{\pi_i(1-\pi_i)}{n_i} \implies \text{Var}\left[Y_i|\mathbf{x}_i\right] = n_i\pi_i(1-\pi_i)$$

- the Systematic component

$$\mathrm{E}\left[P_i|\mathbf{x}_i\right] = \pi_i = h\Big(\mathbf{x}_i^\top \boldsymbol{\beta}\Big), \text{or, equivalently, } g(\pi_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$$

  We keep in mind that:

  - the choice of the link function affects only the systematic component
  - since the expected value of the relative frequency is a probability $\pi_i \in [0,1]$, the *link* function $h(\cdot)$ should be chosen among all functions satisfying the following requirement (*bounded*):

$$h(\cdot) : \mathbb{R} \longmapsto [0,1]$$

    (image of the linear predictor bounded between 0 and 1)
  - furthermore, $h(\cdot)$ should be *differentiable and invertible*, so that

$$\mathbf{x}_i^\top \boldsymbol{\beta} = h^{-1}(\pi_i) = g(\pi_i)$$

  Especially the last two are important for choice of link functions.

14.13.2 Common choices for $g(\cdot)$

Remark 167. Here we look at the inverse of the link function (which is function of the probability and returns the linear predictor)

Proposition 14.13.1 (Common inverse of link function)**.** *Here are the logit, probit and c-log-log functions:*

$$\begin{aligned}
\text{logit}(\pi_i) &= \ln \frac{\pi_i}{1-\pi_i} = \mathbf{x}_i^\top \boldsymbol{\beta} \quad \textit{(canonical)} \\
\text{probit}(\pi_i) &= \Phi^{-1}(\pi_i) = \mathbf{x}_i^\top \boldsymbol{\beta} \\
\text{cloglog}(\pi_i) &= \ln\left[-\ln(1-\pi_i)\right] = \mathbf{x}_i^\top \boldsymbol{\beta}
\end{aligned}$$

where in the probit $\Phi^{-1}(\cdot)$ denotes the inverse (cumulative) probability function of a standard Gaussian r.v.

Remark 168. Some notes:

- c-log-log means the double logarithm of the *complementary* to 1 of the probability; the minus ouside the innermost log is due to the fact that this will take negative or zero values.
- in the probit case, we *have* the inverse of the link function $\Phi^{-1}(\cdot)$ even if we don't know its closed formula (but that is not a problem, we use numerical algorithm to approximate it: important thing is the function is invertible)

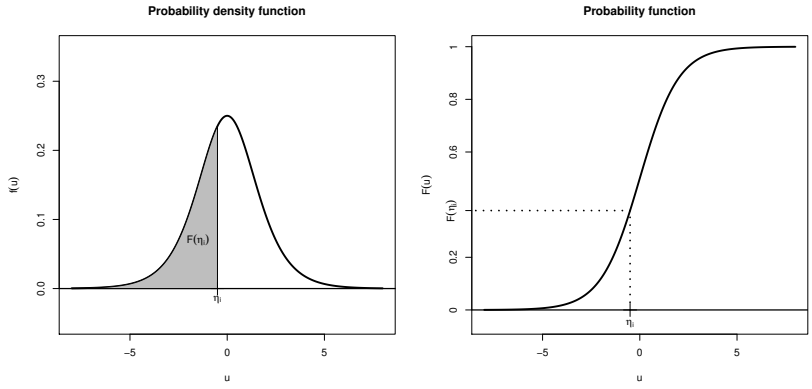
Remark 169 (Historical perspective). We have:

- 1922: Fisher introduces the *c-log-log* link function. Fisher applied the cloglog to the observed relative frequency and the fitted the model
- 1933: first use of the *probit* link function
- 1944: the *logit* link function is proposed
- 1972: Nelder and Wedderburn present their first work on GLM

Remark 170. Beside these three commonly used link function there are others. What could be a general approach which could be used to define a link function?
We need to come up with a function which is:

- bounded between 0 and 1
- differentiable
- invertible

There are several methods to define link functions. A very general approach to define a link function is just the *use of probability functions* (as done for probit)



(a) (b)

Figure 14.12: Probability functions as link functions

14.13.3 Probability functions as link functions

Important remark 129. Letting

- U be an absolutely continuous r.v.
- $f(u)$ be the (known) probability density function of U such that $f(u) > 0, \forall u \in \mathbb{R}$
- $F(u) = \int_{-\infty}^{u} f(t)dt$ be the (cumulative) probability function of U

Thus:

- $F(\cdot) : \mathbb{R} \longmapsto [0, 1]$
- $F(\cdot)$ is differentiable: $\dfrac{\partial}{\partial u} F(u) = f(u)$
- $F(\cdot)$ is invertible: $f(u) > 0$ implies that $F(u)$ is monotonically increasing

Remark 171. Graphically speaking (fig 14.12) starting from a probability density function (left) if we take the value of the linear predictor for any unit (η_i) and compute the integral from $-\infty$ up to η_i of the density one will get the expected value for the relative frequency. Let's see some examples in what follows: both logit, probit and cloglog can be viewed as special case of this general approach. These function are actually the inverse of the cumulative distribution function F.

Example 14.13.1 (Logistic random variable (logit function))**.** Defined as

$$f(u) = \frac{\exp(u)}{[1 + \exp(u)]^2} \implies F(u) = \frac{\exp(u)}{1 + \exp(u)}$$

For this distribution we have:

$$\mathrm{E}[U] = 0, \quad \mathrm{Var}[U] = \frac{\pi^2}{3}$$

To derived the g function (logit), inverse of the $h = F$ function

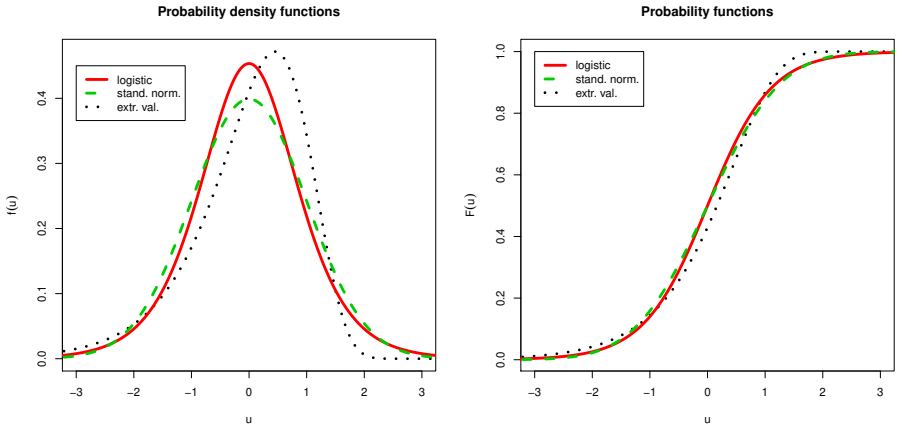
$$\pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} \implies \eta_i = \ln \frac{\pi_i}{1 - \pi_i}$$

Example 14.13.2 (Standard normal random variable (probit function))**.** Defined as

$$f(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) \implies F(u) = \Phi(u)$$

For this distribution we have:

$$\mathrm{E}[U] = 0, \quad \mathrm{Var}[U] = 1$$



(a) (b)

Figure 14.13: Probability functions as link functions - graphical comparisons

To derived the g function (probit), inverse of the $h = F$ function

$$\pi_i = \Phi(\eta_i) \implies \eta_i = \Phi^{-1}(\pi_i)$$

Recalling that $\Phi(u)$ (and thus Φ^{-1}) does not have a closed form expression.

Example 14.13.3 (Extreme (minimum) value random variable (cloglog function))**.** Defined as:

$$f(u) = \exp\left[u - \exp(u)\right] \implies F(u) = 1 - \exp\left[-\exp(u)\right]$$

For this distribution we have:

$$\mathrm{E}[U] = -0.5772, \quad \mathrm{Var}[U] = \frac{\pi^2}{6}$$

To derived the g function (c-log-log), inverse of the $h = F$ function

$$\pi_i = 1 - \exp\left[-\exp(\eta_i)\right] \implies \eta_i = \ln\left[-\ln(1 - \pi_i)\right]$$

Remark 172. To have an idea of the shape look at fig 14.13 The probability density functions and the corresponding probability functions have been rescaled in order to have expected values equal to 0 and variances equal to 1.
Even after this normalization both logistic and standard normal are symmetric around while the extr. val. is skewed to the right (this why mode is not 0).
There's a different tail in logistic vs normal (which we'll see furthrmore): logistic tails are heavier than normal ones (tails of std gaussian approach zero faster than the logistic ones).

Important remark 130 (Link functions - graphical comparison on the logit scale). In figure 14.14 the link function[1] plotted not on original but on the logit scale (so the logit is a straight line, because one is the inverse of the other, considered as reference).
The plot is similar, while the *major differences* between the three link functions resides for the extreme sides of the linear predictor range of variation, where logit/probability predicted are actually somewhat different. In general:

- when considering values for π_i the range from 0.1 to 0.9 (corresponding to values for $\text{logit}(\pi_i)$ in the range from -2.197 to 2.197), the three functions are almost equivalent, and show an almost linear behaviour

[1]Note that the values of the linear predictors have been rescaled to remove possible differences in the scale of the regression coefficients

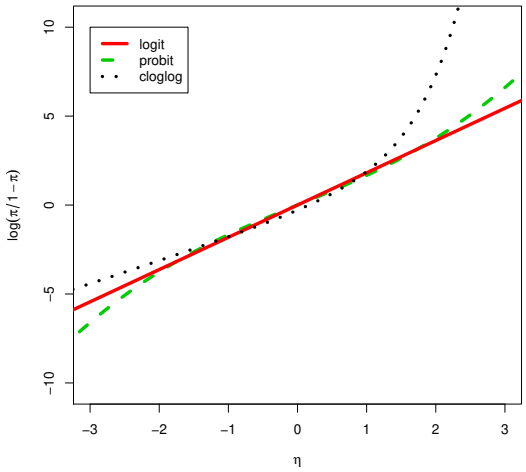


Figure 14.14: Link function graphical comparison (on the logit scale)

- the probit function approaches estreme values for π_i (0 and 1) at a faster rate than the logit function (the logistic distribution has thicker tails than the standard normal distribution)
- the c-log-log function has different tail behaviours:
  - it approaches the value $\pi_i = 0$ at a slower rate than the other two functions
  - it approaches the value $\pi_i = 1$ at a faster rate than the other two functions

  Thus the extreme (minimum) value distribution is skewed to the left.

Remark 173. When dealing with extremely rare or extremely common binary outcome it might end up that:

- we have differences between the three
- one/some link function yields a model that isn't adequate, considering GOF tests (while for other link functions we get models judged adequate)

Otherwise, when dealing with probabilities in the midrange, the behaviour will be similar (all the three yields adequate or non-adequate models).

14.14 The use of non-canonical link functions

Remark 174. Here we look at the theoretical implications of using non-canonical (non-logit) link functions in GLMs for binary outcomes.
If we want to fit GLM with our own link function (not provided by `glm` in R) this is what is needed.

14.14.1 Log-likelihood and sufficient statistics

Starting from the loglik the general expression when using GLM with bernoulli/binomial relative freqs:

$$l(\beta_0, \dots, \beta_p) = \sum_{i=1}^{n} \left\{ y_i \ln \frac{h(\mathbf{x}_i^\top \boldsymbol{\beta})}{1 - h(\mathbf{x}_i^\top \boldsymbol{\beta})} + n_i \ln \left[1 - h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right) \right] \right\} + c$$

Here if we choose a link function that is no longer the logistic link function, that is

$$h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right) \neq \frac{\exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1+\exp\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}$$

then the logit transformation in the loglikelihood will not be anymore the linear predictor, that is

$$\ln \frac{h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)}{1-h\left(\mathbf{x}_i^\top \boldsymbol{\beta}\right)} \neq \mathbf{x}_i^\top \boldsymbol{\beta}$$

(the equivalence above holds iff we choose the logit link function, not with probit/cloglog or anything else).
So we will not be able to write down loglik in compact way with an expression involving $\sum_{i=1}^n y_i, \ldots, \sum_{i=1}^n y_i x_{pi}$ (cross product of each regressor in the linear predictor with the dependent variable) which will *not be sufficient statistics* for $\boldsymbol{\beta}$ any more. So we loose one of the property of canonical link function.

Remark 175. This is not necessarily a problem on a practical point of view: we just need the whole sample/dataset since knowing only these statistics is no longer sufficient

14.14.2 Score function

What happens to the score function? Considering the general formula for the generic element of the score function for a GLM we've seen we can come up with different expression depending on whether we're using the canonical link function or not .
If we give up with the canonical link we have to turn back to the general expression for the score function:

$$\begin{aligned}
U_j(\boldsymbol{\beta}) &= \sum_{i=1}^n \frac{p_i - \mathbb{E}\left[P_i|\mathbf{x}_i\right]}{\operatorname{Var}\left[P_i|\mathbf{x}_i\right]} \frac{\partial \mathbb{E}\left[P_i|\mathbf{x}_i\right]}{\partial \eta_i} x_{ji} \\
&= \sum_{i=1}^n \frac{\frac{y_i}{n_i} - \pi_i}{\frac{\pi_i(1-\pi_i)}{n_i}} \frac{\partial \pi_i}{\partial \eta_i} x_{ji} \\
&= \sum_{i=1}^n \frac{1}{n_i} \frac{y_i - n_i\pi_i}{\frac{\pi_i(1-\pi_i)}{n_i}} \frac{\partial \pi_i}{\partial \eta_i} x_{ji} \\
&= \sum_{i=1}^n \frac{y_i - \mathbb{E}\left[Y_i|\mathbf{x}_i\right]}{\pi_i(1-\pi_i)} \frac{\partial \pi_i}{\partial \eta_i} x_{ji}
\end{aligned}$$

(here the sum on i is on covariate patterns or unit depending on the structure of the data). By particularizing the general expression in case of relative frequency (second equation) we can perform some simplification by rewriting in terms of number of successess and their expected value (last equation) but we see that in the last one we have to deal explicitly with $\frac{\partial \pi_i}{\partial \eta_i}$ (which is just the first partial derivative of the h function, that is $\frac{\partial \pi_i}{\partial \eta_i} = \frac{\partial}{\partial \eta_i} h(\eta_i)$): to evaluate this first partial derivative, if we're using a density function to define the link function, here we use the value of the density function.

14.14.3 Observed Fisher information

Remark 176. We already know that when dealing with noncanonical link function we loose the equivalence between expected and observed Fisher information matrix

Considering the general formula for the generic element of the observed Fisher information matrix for a GLM:

$$\begin{aligned}
i_{jl}(\boldsymbol{\beta}) &= \sum_{i=1}^n \frac{x_{ji}x_{li}}{\operatorname{Var}\left[P_i|\mathbf{x}_i\right]} \left(\frac{\partial \mathbb{E}\left[P_i|\mathbf{x}_i\right]}{\partial \eta_i}\right)^2 - \sum_{i=1}^n \left\{p_i - \mathbb{E}\left[P_i|\mathbf{x}_i\right]\right\} \frac{\partial}{\partial \beta_l} \left(\frac{x_{ji}}{\operatorname{Var}\left[P_i|\mathbf{x}_i\right]} \frac{\partial \mathbb{E}\left[P_i|\mathbf{x}_i\right]}{\partial \eta_i}\right) \\
&= \sum_{i=1}^n \frac{n_i x_{ji}x_{li}}{\pi_i(1-\pi_i)} \left(\frac{\partial \pi_i}{\partial \eta_i}\right)^2 - \sum_{i=1}^n \left\{y_i - n_i\pi_i\right\} \frac{\partial}{\partial \beta_l} \left(\frac{x_{ji}}{\pi_i(1-\pi_i)} \frac{\partial \pi_i}{\partial \eta_i}\right)
\end{aligned}$$

again this is a matter of starting from the general expression and replaceing

14.14.4 Expected Fisher information

When we move to the expected, the expression simplifies; considering the general formula for the generic element of the expected Fisher information matrix for a GLM:

$$\begin{aligned} I_{jl}(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \frac{x_{ji}x_{li}}{\text{Var}\left[P_i|\mathbf{x}_i\right]}\left(\frac{\partial\,\mathbb{E}\left[P_i|\mathbf{x}_i\right]}{\partial\eta_i}\right)^2 \\ &= \sum_{i=1}^{n} \frac{n_i x_{ji}x_{li}}{\pi_i(1-\pi_i)}\left(\frac{\partial\pi_i}{\partial\eta_i}\right)^2 \neq i_{jl}(\boldsymbol{\beta}) \end{aligned}$$

In the last we explicitly take into account the square of the first partial derivative of the link $\left(\frac{\partial\pi_i}{\partial\eta_i}\right)^2 = \left(\frac{\partial}{\partial\eta_i}h(\eta_i)\right)^2$

14.14.5 Matrix representation

We can come up with matrix notation for the score vector and expected information matrix:

- for the Score function:

$$U(\boldsymbol{\beta}) = \mathbf{X}^\top\mathbf{Q}\left[\mathbf{y}-\boldsymbol{\mu}\right]$$

- for the Expected Fisher information matrix we have to introduce an additional (more general) matrix $\mathbf{Q}$ to take in account that differently from logistic we have a slightly more complicated formula:

$$I(\boldsymbol{\beta}) = \mathbf{X}^\top\mathbf{Q}\mathbf{W}\mathbf{Q}\mathbf{X} \neq i(\boldsymbol{\beta})$$

where $\mathbf{Q}$ is a diagonal matrix whose elements are defined using the first partial derivative of the link function

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{\pi_1(1-\pi_1)}\frac{\partial\pi_1}{\partial\eta_1} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\pi_n(1-\pi_n)}\frac{\partial\pi_n}{\partial\eta_n} \end{bmatrix}$$

However in the logistic case all simplifies to what already seen, that is if $\pi_i = \frac{\exp\left(\mathbf{x}_i^\top\boldsymbol{\beta}\right)}{1+\exp\left(\mathbf{x}_i^\top\boldsymbol{\beta}\right)}$, then all is simplified since its first partial derivative and the $\mathbf{Q}$ matrix are respectively

$$\begin{aligned} &\frac{\partial\pi_i}{\partial\eta_i} = \pi_i(1-\pi_i) \\ &\mathbf{Q} = \mathbf{I}_n \end{aligned}$$

14.14.6 Maximum likelihood estimation

Generally, maximum likelihood estimates of $\boldsymbol{\beta}$ can be obtained only using numerical optimisation techniques:

- since $I(\boldsymbol{\beta}) \neq i(\boldsymbol{\beta})$, the Newton-Raphson algorithm *does not coincide* with the Fisher Scoring algorithm, but the differences in the final results are usually negligible;
- the Fisher Scoring algorthm is usually preferred (default in R), due to its numerical stability ($I(\boldsymbol{\beta})$ is always invertible);
- the recursive formula associated with the Fisher Scoring algorithm can be expressed as the solution of an iterative reweighted least square problem:

$$\begin{aligned} \mathbf{b}^{(r+1)} &= \mathbf{b}^{(r)} + \left(\mathbf{X}^\top\mathbf{Q}^{(r)}\mathbf{W}^{(r)}\mathbf{Q}^{(r)}\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Q}^{(r)}\left[\mathbf{y}-\mathbf{m}^{(r)}\right] \\ &= \left(\mathbf{X}^\top\mathbf{Q}^{(r)}\mathbf{W}^{(r)}\mathbf{Q}^{(r)}\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Q}^{(r)}\mathbf{W}^{(r)}\mathbf{z}^{(r)} \end{aligned}$$

  it involves quantities evaluated all at the current approximation for the ML estimate $\mathbf{b}^{(r)}$, that are:

  - the inverse of the expected fisher information matrix $\left(\mathbf{X}^\top\mathbf{Q}^{(r)}\mathbf{W}^{(r)}\mathbf{Q}^{(r)}\mathbf{X}\right)^{-1}$

- the score function $\mathbf{X}^\top \mathbf{Q}^{(r)} \left[\mathbf{y} - \mathbf{m}^{(r)}\right]$
- the pseudo dependent variable defined closely to what already seen

$$\mathbf{z}^{(r)} = \mathbf{Q}^{(r)}\mathbf{X}\mathbf{b}^{(r)} + \left[\mathbf{W}^{(r)}\right]^{-1}\left[\mathbf{y} - \mathbf{m}^{(r)}\right]$$

with the difference that here we have $\mathbf{Q}^{(r)}$.

Remark 177. Also with non canonical link we can interpret the Fisher scoring algorithm as an iterative re-weighted least square algorithm.

Remark 178. non è escluso il ricordarsi la formula ricorsiva a memoria per l'esame

14.14.7 Maximum likelihood estimator asymptotics

In terms of asymptotic behaviour of ML estimator, if a GLM for a binary outcome is correctly specified (we've choosen a proper linear predictor and link function), it is possible to prove that the ML estimator is well behaved at least asymptotically (*unbiased*, *efficient*, *mvn*):

$$\hat{\mathbf{B}} \xrightarrow{d} MVN_{p+1}\left(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}\right)$$

with:

- asymptotic properties of ML estimator holds if total sample size is large, that is as $N = \sum_i n_i \longrightarrow \infty$: so it is not necessary that $n_i \longrightarrow \infty \; \forall i$;
- asymptotic variance of $\hat{\mathbf{B}}$ can be estimated using

$$\widehat{I(\boldsymbol{\beta})}^{-1} = \left[\mathbf{X}^\top \hat{\mathbf{Q}}\hat{\mathbf{W}}\hat{\mathbf{Q}}\mathbf{X}\right]^{-1}$$

where $\hat{\mathbf{Q}}$ is obtained by evaluating $\mathbf{Q}$ at $\hat{\mathbf{b}}$.

14.14.8 Deviance, residuals and goodness of fit tests

For deviance, approximated χ^2 and HL statistics we have as always:

$$D = 2\sum_{i=1}^{n}\left[y_i \ln\frac{y_i}{\hat{m}_i} + (n_i - y_i)\ln\frac{n_i - y_i}{n_i - \hat{m}_i}\right] \approxeq \sum_{i=1}^{n}\frac{(y_i - n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i(1-\hat{\pi}_i)} = \chi^2$$

$$D_{HL} = \sum_{l=1}^{G}\frac{(y_l - n_l\bar{\pi}_l)^2}{n_l\bar{\pi}_l(1-\bar{\pi}_l)} = \sum_{l=1}^{G}\bar{e}_l^P$$

In these formula above the choice of the link function affects only the formula for computing $\hat{\pi}_i = h\left(\mathbf{x}_i^\top\hat{\mathbf{b}}\right)$ which determines the estimated/expected number of successes. Once estimated the model and the betas we compute $\hat{\pi}_i$ (for each unit/covariate pattern) and then we use it normally, by plugging these quantities in the formula above.
Finally:

- if we had *sparse data* we'll have to deal it as done before. If we have a sufficient number of units per covariate pattern then we can use residual deviance or pearson χ^2 statistics; if data are sparse to check adequacy of the model we use with Hosmer-Lemeshow test;
- a change in the link function *may improve/deteriorate* the adequacy of a model.

Remark 179. As always, when dealing with GOF test we have to check the number of units per each covariate pattern since residual deviance and pearson χ^2 has limiting distribution known if we have a large number of units for each covariate pattern

14.14.9 Hypothesis testing and model comparisons

Remark 180 (Hypothesis testing). Finally if we're interesting on linear hypotheses, the choice of the link function does not alter the machinery: Linear hypotheses on $\boldsymbol{\beta}$ can be tested using either the LRT statistic or the Wald test statistic, no matter which link function has been chosen (however recall that both can be used assuming that the model, and so the link function as well, is adequate, using deviance/chis/HL)

Remark 181 (Model comparison). When we want to compare models we have to keep in mind that:

- two GLMs for a binary outcome are **nested** if and only if they have the same link function;
- so if we want to compare binary models with *different link functions* we can use only *model selection criteria* such as the *AIC* or the *BIC*

14.15 Example: probit and cloglog

We continue with the example on epidemic outbreak of a disease that is spread by mosquitoes in a city started in section 14.12. By doing the estimate using different link functions we obtain

Logistic regression model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.618	0.613	-4.270	0.000
age	0.030	0.014	2.203	0.028
areaSect2	1.575	0.502	3.139	0.002
statusMiddle	0.714	0.654	1.092	0.275
statusUpper	0.305	0.604	0.505	0.613

Probit regression model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.593	0.339	-4.696	0.000
age	0.018	0.008	2.316	0.021
areaSect2	0.953	0.295	3.225	0.001
statusMiddle	0.442	0.383	1.156	0.248
statusUpper	0.187	0.352	0.532	0.594

Bernoulli GLM with c-log-log link function

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.475	0.497	-4.976	0.000
age	0.021	0.009	2.283	0.022
areaSect2	1.215	0.405	3.002	0.003
statusMiddle	0.685	0.518	1.322	0.186
statusUpper	0.287	0.495	0.581	0.561

Important remark 131. Some comments:

- the link function affects the *scale*/unit of measure of the regression coefficients: it *does not make sense to compare the estimated regression* coefficient for the same regressor obtained using different link functions. Eg it would be wrong to say age has a stronger effect when using the logistic link function compared to the others.
  however there's consistency in the sign of the coefficient: all these three link functions have sigmoid monotonic increasing shape and thus there's a positive coefficient for regressor that increases the probability of event (and negative coefficient otherwise);
- by applying tha chain rule of derivation on the link function one has:

$$\frac{\partial \pi_i}{\partial x_{ji}} = \frac{\partial \pi_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial x_{ji}} = \beta_j \frac{\partial h(\eta_i)}{\partial \eta_i}$$

  When $\frac{\partial h(\eta_i)}{\partial \eta_i} > 0$ (*the link function is monotonically increasing*, as in all the cases we've seen and as seen in the logistic case):

  - the direction of the change in π_i due to a unit increase in x_{ji} depends on the sign of β_j (if positive sign means positive association with event probability and viceversa);
  - the magnitude of the change depends also on the values of all the regressors.
- differently from the logistics, when dealing with other link function, we loose the connection between the regression coefficients and the odds ratio $\frac{\frac{\pi_{i+}}{1-\pi_{i+}}}{\frac{\pi_i}{1-\pi_i}}$. So it's impossible to come up with transformation of regression coefficients that can be linked with odds-ratio. The reason for the widespread use of logistic is the last connection: odds-ratio is very popular in anglosaxon world and is simpler to understand the model quantities/associations.

14.15.1 Goodness of fit test

For the:

- probit link function, the Hosmer-Lemeshow test ($G = 10$)

		Observed			Expected	
Intervals for $\hat{\pi}_i$	n_l	Absence	Presence		Absence	Presence
[0.0598,0.0796]	10.00	10.00	0.00		9.33	0.67
(0.0796,0.103]	10.00	9.00	1.00		9.07	0.93
(0.103,0.162]	11.00	9.00	2.00		9.54	1.46
(0.162,0.186]	9.00	8.00	1.00		7.42	1.58
(0.186,0.261]	9.00	7.00	2.00		6.98	2.02
(0.261,0.33]	10.00	6.00	4.00		6.97	3.03
(0.33,0.421]	9.00	4.00	5.00		5.68	3.32
(0.421,0.513]	10.00	6.00	4.00		5.35	4.65
(0.513,0.659]	10.00	5.00	5.00		4.32	5.68
(0.659,0.852]	10.00	3.00	7.00		2.45	7.55

  Thus $\chi^2_{HL} = 3.5298$ and $\Pr\left[\chi^2_{(8)} \geq 3.5298\right] = 0.8969$;

- c-log-log link function the Hosmer-Lemeshow test ($G = 10$, same number of groups but different groups defined on the fitted probability which are influenced by the link function)

		Observed			Expected	
Intervals for $\hat{\pi}_i$	n_l	Absence	Presence		Absence	Presence
[0.0841,0.0993]	10.00	10.00	0.00		9.10	0.90
(0.0993,0.119]	10.00	9.00	1.00		8.89	1.11
(0.119,0.168]	10.00	8.00	2.00		8.59	1.41
(0.168,0.185]	10.00	9.00	1.00		8.20	1.80
(0.185,0.252]	9.00	8.00	1.00		7.03	1.97
(0.252,0.303]	10.00	4.00	6.00		7.23	2.77
(0.303,0.399]	9.00	5.00	4.00		5.93	3.07
(0.399,0.514]	10.00	5.00	5.00		5.49	4.51
(0.514,0.67]	10.00	7.00	3.00		4.40	5.60
(0.67,0.911]	10.00	2.00	8.00		2.25	7.75

  Thus $\chi^2_{HL} = 10.86$ and $\Pr\left[\chi^2_{(8)} \geq 10.86\right] = 0.2098$.

In both cases the fitted model can be considered adequate for the data.

14.15.2 Comparison among link functions

Here we report residual deviances, the HL test statistic and the AIC:

Link function	D	D_{HL}	AIC
logit	94.462	5.3267	107.47
probit	94.081	3.5298	107.09
c-log-log	94.689	10.8600	107.70

As seen:

- D should not be used to compare GLMs with different link functions, as they are not nested models (*the distribution of the difference between two residual deviances is not known for non-nested models*)
- D_{HL} should not be used to compare Bernoulli GLMs with different link functions, as the values of D_{HL} have been obtained using different splittings of the data into subgroups (*even if the number of subgroups is the same, the composition of the subgroups depends on the estimated* $\hat{p}_i$)
- the "best" link function can be selected according to the AIC (*note that there is no guarantee that the model with the smallest AIC is an adequate model*)

The smallest AIC/choosen model is the probit.

Remark 182. For these models the residual deviance is not well behaved (we're dealing with sparse data) nevertheless since these models were fitted on same data, they have a common saturated model and the differences we see in residual deviances are reflecting -2 maximized loglik.

TODO: check riflessione here

In this example since models have the same number of params the differences in residual deviances reflect differences in AIC: model with smallest AIC is also model with smallest deviance. So we could compare them based on deviance because characterized by same linear predictor but different link function.
whenever comparing models characterized by different link functions and different linear predictors then the proper tool to make this comparison are either AIC or BIC.
So we start to narrow down the models to compare using GOF/HR to select only the proper one, then we compare them (if more than 1) using AIC/BIC to select the best one.

Chapter 15

Lab 4 and 5 GLM for binary outcomes

15.1 Lab 4 - Regression for binary outcomes

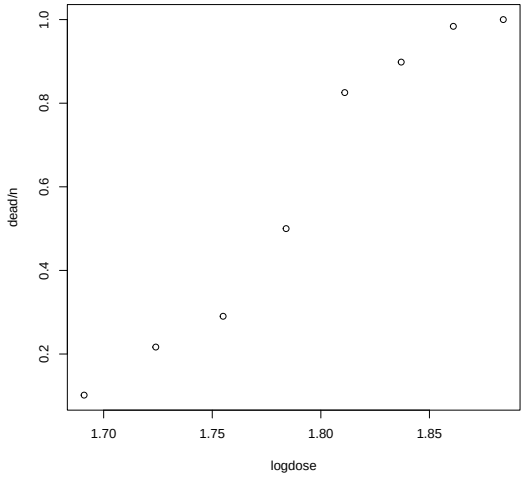
`beetles1` and `beetles2` contain the same data but in covariate pattern and unit level dataset respectively.
Data come from an experiment where beetles were divided in 8 groups and each group was exposed to a level of poison and after sometime the researcher counted the number of death per poison level: idea is to study the connection between the dose of poison and probability of death.

15.1.1 Covariate level dataset

`beetles1` contains one row for each unique covariate pattern and the variables are

- `logdose`: logarithm of the dose of poison
- `n`: number of beetles exposed to a given dose of poison
- `dead`: number of beetles dead among those exposed to a given dose of poison

```
## data structures and use of the glm function
library(lbdatasets)
str(beetles1)

## 'data.frame': 8 obs. of  3 variables:
##  $ logdose: num  1.69 1.72 1.75 1.78 1.81 ...
##  $ n      : int  59 60 62 56 63 59 62 60
##  $ dead   : int  6 13 18 28 52 53 61 60

beetles1

##   logdose  n dead
## 1   1.691 59    6
## 2   1.724 60   13
## 3   1.755 62   18
## 4   1.784 56   28
## 5   1.811 63   52
## 6   1.837 59   53
## 7   1.861 62   61
## 8   1.884 60   60

## graphical representation of the relative frequencies
with(beetles1, plot(logdose, dead / n))
```



In presence of a covariate pattern data structure, the **dependent variable** can be specified in two different ways which lead to the same result:

- by defining a matrix containing one column for the number of "successes" (number of dead beetles) and one column for the number of "failures" (number of survived beetles) for each covariate pattern:

```
summary(logistic1 <- glm(cbind(dead, n - dead) ~ logdose,
                         data = beetles1, family = binomial))

##
## Call:
## glm(formula = cbind(dead, n - dead) ~ logdose, family = binomial,
##     data = beetles1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5878  -0.4085   0.8442   1.2455   1.5860
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.740      5.182  -11.72   <2e-16 ***
## logdose       34.286      2.913   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.116  on 6  degrees of freedom
## AIC: 41.314
##
## Number of Fisher Scoring iterations: 4
```

  Note that in case of a constant number of units for each covariate pattern (eg a scalar value 5), the second column of the matrix can be obtained by substracting the number of successes from the constant (example: `5-dead`)

- by providing the relative frequency on the left handside and by setting the argument `weights` equal to the number of units observed for each covariate pattern (not documented in `glm` btw)

```
summary(logistic2 <- glm(dead / n ~ logdose, weights = n,
                         data = beetles1, family = binomial))

##
## Call:
## glm(formula = dead/n ~ logdose, family = binomial, data = beetles1,
##     weights = n)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5878  -0.4085   0.8442   1.2455   1.5860
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.740      5.182  -11.72   <2e-16 ***
## logdose       34.286      2.913   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.116  on 6  degrees of freedom
## AIC: 41.314
##
## Number of Fisher Scoring iterations: 4
```

In term of goodness of fit, to check adequacy of model to the data, we have more than 50 beetles for each covariate pattern so we go with GOF test based on residual deviance:

```
## Again the two way to estimate yields same results
deviance(logistic1)

## [1] 11.11558

deviance(logistic2)

## [1] 11.11558

summary(logistic1)$df.residual

## [1] 6

summary(logistic2)$df.residual

## [1] 6

## critical value
qchisq(0.05, summary(logistic1)$df.residual, lower.tail = FALSE)

## [1] 12.59159

## p-value
pchisq(deviance(logistic1), summary(logistic1)$df.residual, lower.tail = FALSE)

## [1] 0.08486944
```

So there is no significant evidence against the fitted model, which can be considered adequate for data at hand

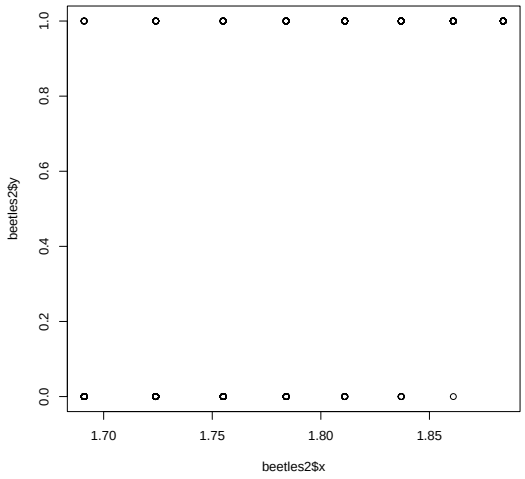
15.1.2 Unit level dataset

Using `beetles2` to reproduce what we've found: it contains two columns

- `x`: logarithm of the dose of poison the beetle was exposed to
- `y`: Is the beetle dead (0=no, 1=yes)

If we try to plot data here is uninformative since we have both 0 and 1 for each covariate pattern/logdose level

```
str(beetles2)

## 'data.frame': 481 obs. of  2 variables:
##  $ x: num  1.69 1.69 1.69 1.69 1.69 ...
##  $ y: int  1 1 1 1 1 1 0 0 0 0 ...

## graphical representation  - uninformative plot
plot(beetles2$x, beetles2$y)
```



In essence `beetles1` e `beetles2` contain the same information structured in different ways: we can obtain one from another and viceversa.

```
## unique covariate patterns can be found using unique
unique(beetles2$x)

## [1] 1.691 1.724 1.755 1.784 1.811 1.837 1.861 1.884

table(beetles2$x, beetles2$y) # this basically coincides with beetles1

##
##          0  1
##   1.691 53  6
##   1.724 47 13
##   1.755 44 18
##   1.784 28 28
##   1.811 11 52
##   1.837  6 53
##   1.861  1 61
##   1.884  0 60

beetles1
```

```
##   logdose  n dead
## 1   1.691 59    6
## 2   1.724 60   13
## 3   1.755 62   18
## 4   1.784 56   28
## 5   1.811 63   52
## 6   1.837 59   53
## 7   1.861 62   61
## 8   1.884 60   60
```

As before in presence of individual data, the dependent variable can be specified in two different ways:

- by directly specifing a dummy/indicator variable as dependent variable

```
summary(logistic4 <- glm(y ~ x, data = beetles2, family = binomial))

##
## Call:
## glm(formula = y ~ x, family = binomial, data = beetles2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4946  -0.5979   0.2047   0.4491   2.3765
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.740      5.182  -11.72   <2e-16 ***
## x             34.286      2.913   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.44  on 480  degrees of freedom
## Residual deviance: 372.35  on 479  degrees of freedom
## AIC: 376.35
##
## Number of Fisher Scoring iterations: 5
```

- by defining a matrix containing one column for the "success" (the beetle died) and one column for the "failure"(the beetle survived)

```
summary(logistic3 <- glm(cbind(y, 1 - y) ~ x, data = beetles2, family = binomial))

##
## Call:
## glm(formula = cbind(y, 1 - y) ~ x, family = binomial, data = beetles2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4946  -0.5979   0.2047   0.4491   2.3765
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.740      5.182  -11.72   <2e-16 ***
## x             34.286      2.913   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.44  on 480  degrees of freedom
## Residual deviance: 372.35  on 479  degrees of freedom
```

```
## AIC: 376.35
##
## Number of Fisher Scoring iterations: 5
```

The two methods

- contain the same estimates for betas as `logistic1` and `logistic2`; in general: *as far parameter inference is concerned, the choice of the data structure is irrelevant*
- the **difference** we found are in *deviances* (eg needed if we want to perform GOF test) and model selection criteria AIC; in this case the correct values can be obtained *only* using a covariate pattern data structure (there is no internal check in the *glm* function to detect the possible presence of repeated covariate patterns)

15.1.3 Change in the link function

Estimate of probit and cloglog models; the model will have the same linear predictor but different link function (so it will be a different systematic component):

```
## probit
probit_mod <- update(logistic1, family = binomial(probit))
deviance(probit_mod)

## [1] 9.986957

pchisq(deviance(probit_mod), summary(probit_mod)$df.residual, lower.tail = FALSE)

## [1] 0.1252024

## clogog
cloglog_mod <- update(logistic1, family = binomial(cloglog))
deviance(cloglog_mod)

## [1] 3.514334

pchisq(deviance(cloglog_mod), summary(cloglog_mod)$df.residual, lower.tail = FALSE)

## [1] 0.7420616
```

So all these three models seems adequate (their deviances are not significantly larger than 0) to describe the data; the model with the cloglog link function has a smaller deviance.
Note that the three considered models are not nested: a change in the link function does not produce a nested model (nested are obtained only by removing regressors). Therefore the difference of their deviances has unknown distribution.
Thus to choose among them we use AIC:

```
sapply(list("logistic" = logistic1,
            "probit"   = probit_mod,
            "cloglog"  = cloglog_mod),
       AIC)

## logistic   probit  cloglog
## 41.31361 40.18499 33.71237
```

Not necessarily the canonical link function is the one producing the best results: here, according to the AIC, the cloglog link function should be preferred.
So one possible way to improve a GLM could be not only by changing the linear predictor by adding/subtracting regressors, but also by changing the link function.
Note that in this example, since the models have the same linear predictor (that is, the same number of parameters) the ordering of the models according to the AIC coincides with the ordering of the models according to the (residual) deviance (model with the smallest deviance will also have lowest AIC).

15.2 Lab 5 - Dealing with sparse data

Here we will use the packages `AER` (contains the data), `ResourceSelection` (contains an R function to compute the Hosmer-Lemeshow test statistic) and `car` (contains functions to build quantile-quantile plots).
The dataset `SwissLabor` provides individual information (sample units data structure): `participation` (in the labor force) is the outcome, the remaining regressors

```
library(AER)

## Caricamento del pacchetto richiesto:  lmtest
## Caricamento del pacchetto richiesto:  zoo
##
## Caricamento pacchetto:  'zoo'
## I seguenti oggetti sono mascherati da 'package:base':
##
##     as.Date, as.Date.numeric
## Caricamento del pacchetto richiesto:  sandwich
## Caricamento del pacchetto richiesto:  survival
##
## Caricamento pacchetto:  'survival'
## Il seguente oggetto è mascherato da 'package:lbdatasets':
##
##     ovarian

library(ResourceSelection)

## ResourceSelection 0.3-6 2023-06-27

library(car)
data(SwissLabor) # ?SwissLabor for details
str(SwissLabor)

## 'data.frame': 872 obs. of  7 variables:
##  $ participation: Factor w/ 2 levels "no","yes": 1 2 1 1 1 2 1 2 1 1 ...
##  $ income       : num  10.8 10.5 11 11.1 11.1 ...
##  $ age          : num  3 4.5 4.6 3.1 4.4 4.2 5.1 3.2 3.9 4.3 ...
##  $ education    : num  8 8 9 11 12 12 8 8 12 11 ...
##  $ youngkids    : num  1 0 0 2 0 0 0 0 0 0 ...
##  $ oldkids      : num  1 1 0 0 2 1 0 2 0 2 ...
##  $ foreign      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

Let's see the number of covariate patterns

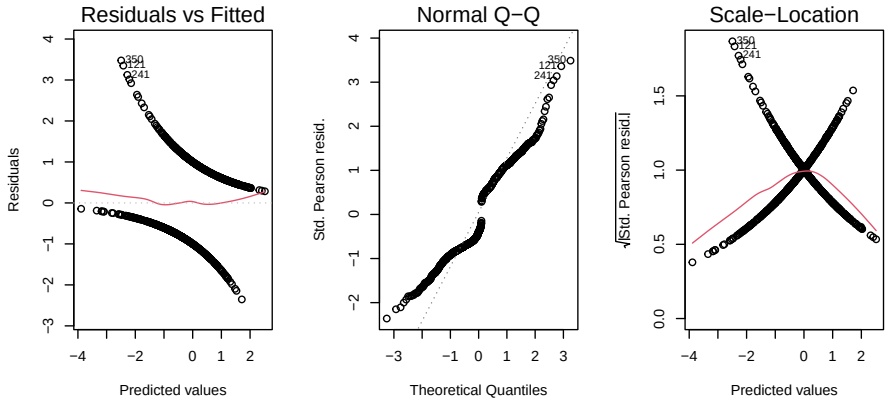
```
## determining the unique covariate patterns
cov_patterns <- unique(SwissLabor[, -1]) # rm the outcome
nrow(cov_patterns)

## [1] 872

nrow(SwissLabor)

## [1] 872
```

The number of covariate pattern coincides with the sample size this implies that $n_i = 1$ for each covariate pattern; so we are in a sparse data situation.
If we fit a logistic regression model using a dichotomous factor as dependent variable by default R focuses on the second category (in alphabetical order in this example 1 if participation = yes)

```
## model fitting
summary(model <- glm(participation ~ ., data = SwissLabor, family = binomial))
```

```
##
## Call:
## glm(formula = participation ~ ., family = binomial, data = SwissLabor)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9384  -0.9727  -0.5383   1.0675   2.2681
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 10.37435    2.16685   4.788 1.69e-06 ***
## income      -0.81504    0.20550  -3.966 7.31e-05 ***
## age         -0.51033    0.09052  -5.638 1.72e-08 ***
## education    0.03173    0.02904   1.093    0.275
## youngkids   -1.33072    0.18017  -7.386 1.51e-13 ***
## oldkids     -0.02199    0.07377  -0.298    0.766
## foreignyes   1.31040    0.19976   6.560 5.38e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1203.2  on 871  degrees of freedom
## Residual deviance: 1052.8  on 865  degrees of freedom
## AIC: 1066.8
##
## Number of Fisher Scoring iterations: 4
```

```
## graphical analysis of the residuals: all are uninformative since we have sparse data
par(mfrow = c(1, 3))
plot(model, which = 1) # the curves style due to the unit based data structure (not informative)
plot(model, which = 2) # not informative
plot(model, which = 3) # not informative
```



Looking at the summary statistics (deviance/AIC) we know they're not to be trusted. We could start with a GOF test ...

```
## goodness of fit test
deviance(model) # residual
```

```
## [1] 1052.798
```

```
summary(model)$df.residual # degrees of freedom
```

```
## [1] 865

qchisq(0.05, summary(model)$df.residual, lower.tail = FALSE) # critical value

## [1] 934.5329

pchisq(deviance(model), summary(model)$df.residual, lower.tail = FALSE) # p-value

## [1] 1.117042e-05
```

... however $n_i = 1$ for each i is a violation of the assumptions needed to perform a goodness of fit test based on the residual deviance so the goodness of fit test based on the deviance is unreliable.
Thus we go with Hosmer-Lemeshow test

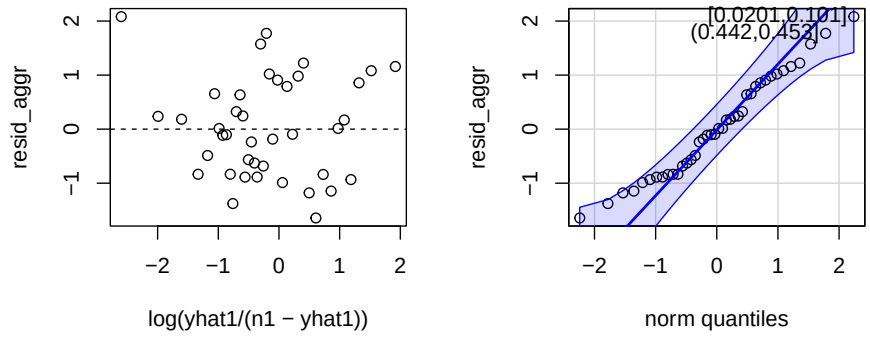
```
## see ?hoslem.test and ?predict.glm (by default g = 10)
(testHL <- hoslem.test(SwissLabor$participation == "yes",
                       predict(model, type = "response")))

##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  SwissLabor$participation == "yes", predict(model, type = "response")
## X-squared = 4.2006, df = 8, p-value = 0.8386
```

Since we don't reject the null the model can be considered adequate. When using this `hoslem.test` function we get also additional infos

```
## the HL test is just a chi-square used to compare these two tables:
## 1) observed frequencies among the 10 groups of probability
testHL$observed

##
## cutyhat          y0 y1
##   [0.0201,0.222] 74 14
##   (0.222,0.291]  64 23
##   (0.291,0.337]  64 23
##   (0.337,0.383]  57 30
##   (0.383,0.431]  52 35
##   (0.431,0.484]  43 44
##   (0.484,0.564]  40 47
##   (0.564,0.658]  35 52
##   (0.658,0.758]  29 58
##   (0.758,0.925]  13 75

## 2) expected frequencies among the 10 groups (col yhat 1 sum of estimated probs for the group)
testHL$expected

##
## cutyhat             yhat0    yhat1
##   [0.0201,0.222] 75.56766 12.43234
##   (0.222,0.291]  64.17984 22.82016
##   (0.291,0.337]  59.68108 27.31892
##   (0.337,0.383]  55.66075 31.33925
##   (0.383,0.431]  51.65704 35.34296
##   (0.431,0.484]  47.44441 39.55559
##   (0.484,0.564]  41.39471 45.60529
##   (0.564,0.658]  33.80097 53.19903
##   (0.658,0.758]  25.06687 61.93313
##   (0.758,0.925]  16.54667 71.45333
```

Since the function provides several infos, we can compute the aggregate Pearson residuals to build plots. To get meaningful plot we should consider more than 10 groups; here we use 40 groups (we have 20 units per group which is sufficiently large to not have the curves pattern)

```
## computation of the aggregate Pearson residuals
(testHL40 <- hoslem.test(SwissLabor$participation == "yes",
                         predict(model, type = "response"),
                         g = 40))

##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  SwissLabor$participation == "yes", predict(model, type = "response")
## X-squared = 33.622, df = 38, p-value = 0.672

y1 <- testHL40$observed[, 2] #observed n of successes (we need the 2nd col)
yhat1 <- testHL40$expected[, 2] # expected n of successes
n1 <- rowSums(testHL40$observed) # n of units within each group (sum of the two cols)
resid_aggr <- (y1 - yhat1) / sqrt((yhat1 * (n1 - yhat1) / n1)) #aggregated pearson residuals

# we can reproducem the HL statistic with this
sum(resid_aggr^2)

## [1] 33.62235

## residual plots: we plot the residuals against the logit
## transformation of estimated probability of success
par(mfrow = c(1, 2))
plot(log(yhat1 / (n1 - yhat1)), resid_aggr)
abline(h = 0, lty = 2)
qqPlot(resid_aggr)
```




```
## [0.0201,0.101]  (0.442,0.453]
##              1             22
```

The aggregated pearson residual are scattered around 0 and are reasonably normal/within bands.
Going for the interpretation of models parameters

```
summary(model)

##
```

```
## Call:
## glm(formula = participation ~ ., family = binomial, data = SwissLabor)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9384  -0.9727  -0.5383   1.0675   2.2681
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 10.37435    2.16685   4.788 1.69e-06 ***
## income      -0.81504    0.20550  -3.966 7.31e-05 ***
## age         -0.51033    0.09052  -5.638 1.72e-08 ***
## education    0.03173    0.02904   1.093    0.275
## youngkids   -1.33072    0.18017  -7.386 1.51e-13 ***
## oldkids     -0.02199    0.07377  -0.298    0.766
## foreignyes   1.31040    0.19976   6.560 5.38e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1203.2  on 871  degrees of freedom
## Residual deviance: 1052.8  on 865  degrees of freedom
## AIC: 1066.8
##
## Number of Fisher Scoring iterations: 4

round(exp(model$coefficients), 5)

## (Intercept)      income         age   education   youngkids     oldkids
## 32027.37103     0.44262     0.60030     1.03224     0.26429     0.97825
##  foreignyes
##     3.70767
```

we have that

- the odds of a foreign woman being in the labor force are almost four times as great as for an Swiss woman, ater controlling for all the other regressors
- each additional decade in age causes a 40% decrease in the the odds of being in the labor force, after controlling for all the other regressors

For what concerns the choice of the link function lets try something different from straight logistic:

```
## probit
model_probit <- glm(participation ~ .,
                    data = SwissLabor,
                    family = binomial(probit))
(testHL_probit <- hoslem.test(SwissLabor$participation == "yes",
                              predict(model_probit, type = "response")))

##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  SwissLabor$participation == "yes", predict(model_probit, type = "response")
## X-squared = 5.9077, df = 8, p-value = 0.6576

## cloglog
model_cloglog <- glm(participation ~ .,
                     data = SwissLabor,
                     family = binomial(cloglog))
(testHL_cloglog <- hoslem.test(SwissLabor$participation == "yes",
                               predict(model_cloglog, type="response")))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  SwissLabor$participation == "yes", predict(model_cloglog, type = "response")
## X-squared = 11.723, df = 8, p-value = 0.164
```

So all three models seem adequate, according to the Hosmer-Lemeshow goodness of fit test; let's do model comparison by AIC

```
### model comparison
sapply(list("logistic" = model,
            "probit"   = model_probit,
            "cloglog"  = model_cloglog),
       AIC)

## logistic   probit  cloglog
## 1066.798 1066.983 1063.356
```

Again the cloglog seems slightly better (despite the fact that has the largest HL test statistics).
We do not compare models by the HL statistics because it's based on different groups (based on predicted probabilities or linear components which are different for the betas).

Chapter 16

GLM with Gamma probabilistic component

16.1 The model

16.1.1 Gamma RV

Remark 183. There's no agreement on a standard parametrization regarding gamma so we could end up with the same density function using parameters with different meaning.
The following is the most convenient way to represent the density for regression problems (from prof's pov). We have two parameters: μ (not usen often, but here is used to mean that it's the expected value) and ν

Definition 16.1.1 (Gamma random variables)**.** These are random variable taking only strictly positive values:

$$Y \sim \text{Ga}(\mu, \nu)$$
$$f(y; \mu, \nu) = \frac{1}{\left(\frac{\mu}{\nu}\right)^{\nu}\Gamma(\nu)} y^{\nu-1} \exp\left(-\frac{\nu}{\mu}y\right)$$

with the following *strictly positive quantities*:

$$y \in \mathbb{R}^+, \quad \mu \in \mathbb{R}^+, \quad \nu \in \mathbb{R}^+$$

The presence of the gamma integral is the reason of the name for the random variable/distribution and is defined as:

$$\Gamma(\nu) = \int_0^{\infty} t^{\nu-1} \exp(t) dt = (\nu - 1)\Gamma(\nu - 1)$$

Remark 184. Interesting feature of the gamma integral is the recursiveness: it's a kind generalization of the factorial for an integer value.

Remark 185 (Graphical display). In terms of general features of this kind of distribution (fig 16.1): by changing the expected value μ we have also a change

Example 16.1.1 (Example usage)**.** It's used for statistical phenomena that *take only positive values* and *show skewed distributions* such as: intensities/densities, durations/waiting times, earnings/expenditures.
Are closed to lognormal distribution when it comes to practical usage

Proposition 16.1.1 (Moments)**.** *When using the parametrization above we have that expected value coincides with μ while variance depends on both parameters (proportional to the squared expected value):*

$$\text{E}[Y] = \mu$$
$$\text{Var}[Y] = \frac{\mu^2}{\nu}$$

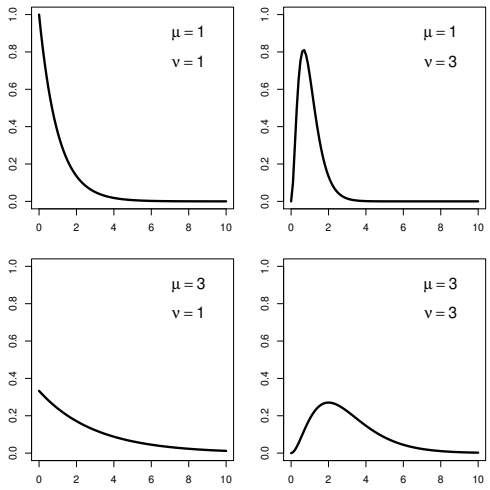


Figure 16.1: Gamma densities for different parameters values.

Thus the coefficient of variation is constant not depending on the expected value:

$$\mathrm{CV}[Y] = \frac{\sqrt{\mathrm{Var}[Y]}}{\mathrm{E}[Y]} = \frac{\sqrt{\frac{\mu^2}{\nu}}}{\mu} = \frac{1}{\sqrt{\nu}}$$

Example 16.1.2 (Special cases: exponential variables)**.** Exponential RV is obtained setting $\nu = 1$

$$Y \sim \mathrm{Exp}(\mu) \implies Y \sim \mathrm{Ga}(\mu, \nu = 1), (\Gamma(1) = 1)$$

Example 16.1.3 (Special cases: χ^2 variables)**.** χ^2 is obtained setting the expected value to positive integer and ν on its half:

$$Y \sim \chi^2(g), g \in \mathbb{N}^+ \implies Y \sim \mathrm{Ga}(\mu = g, \nu = g/2)$$

Important remark 132 (Alternative parameterisations). There are other parametrization in the literature: common choices are definition based on the so called:

- *scale parameter* $\alpha = \frac{\mu}{\nu}$:

$$Y \sim \mathrm{Ga}(\alpha, \nu) \implies f(y; \alpha, \nu) = \frac{1}{(\alpha)^\nu \Gamma(\nu)} y^{\nu-1} \exp\left(-\frac{y}{\alpha}\right)$$

- *rate parameter* (inverse of the scale) $\theta = \frac{\nu}{\mu}$:

$$Y \sim \mathrm{Ga}(\theta, \nu) \implies f(y; \theta, \nu) = \frac{\theta^\nu}{\Gamma(\nu)} y^{\nu-1} \exp(-\theta y)$$

Remark 186. We don't use these (most common parametrization) in the context of regression analysis because in both we don't have a single parameter strictly related to the expected value (central in regression analysis), we have two in both; in regression analysis is more convenient having just one parameter linked to the expected value.

16.1.2 GLM with Gamma probabilistic component

Suppose that we have a random sample of n elements: we assume that each $Y_i \sim \mathrm{Ga}(\mu_i, \nu)$, $i = 1, \ldots, n$ and observation are conditionally independent. We can allow Y_i to have different expected value μ_i but we force the nuisance parameter ν to be common across the units

Gamma distribution belongs to the exponential family; to show it (requires some extra manipulation of the math expression of the density funciton) but we can rewrite the density as:

$$f(y_i;\mu_i,\nu) = \frac{1}{\left(\frac{\mu_i}{\nu}\right)^{\nu}\Gamma(\nu)} y_i^{\nu-1}\exp\left(-\frac{\nu}{\mu_i}y_i\right)$$
$$= \exp\left\{\frac{1}{\frac{1}{\nu}}\left[y_i\left(-\frac{1}{\mu_i}\right) - \ln\mu_i\right] + [(\nu-1)\ln y_i - \ln\Gamma(\nu) + \nu\ln\nu]\right\}$$

where the main ingrediente of exponential family are:

$$\theta_i = \mu_i,\ \phi = \frac{1}{\nu},\ w_i = 1$$
$$a(y_i) = y_i$$
$$b(\mu_i) = -\frac{1}{\mu_i}$$
$$c(\mu_i) = -\ln\mu_i$$
$$d(y_i,\nu) = (\nu-1)\ln y_i - \ln\Gamma(\nu) + \nu\ln\nu$$

so a is the identity (as it should be if we want to use this gamma as probabilistic component of glm), the natural parameter b is $-1/\mu_i$, the nuisance parameter ϕ is $1/\nu$, we have a c function involving only μ_i and a d function which doesn't depend on μ_i; we do not have weights ($w_i = 1$). Thus:

$$Y_i \sim \mathrm{Ef}\left(-\frac{1}{\mu_i}, \frac{1}{\nu}, w_i = 1\right), i = 1,\ldots,n \text{ conditionally indenpendent}$$

16.1.3 Systematic component

How to define the systematic component:

- we'll have the linear predictor as usual
$$\eta_i = \mathbf{x}_i^{\top}\boldsymbol{\beta} = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}$$
- we'll have to choose a link function
$$g(\mathrm{E}[Y_a i|\mathbf{x}_i]) = g(\mu_i) = \eta_i$$
$$h(\eta_i) = \mu_i$$
$$g(\cdot):\mathbb{R}^+ \longmapsto \mathbb{R};\quad h(\cdot):\mathbb{R}\longmapsto\mathbb{R}^+$$

  $\mu_i > 0$ so in term of h function (to be applied to the linear predictor in order to get the expected value we should prefer function whose image is the positive real line/always stricly positive (in order to have expected value falling in proper parameter space). This pose an issue as far as use of canonical function.

  - the natural parameter is $-1/\mu_i$; for the *canonical link function*, we should set the linear predictor equal to the natural parameter
  $$b(\mu_i) = -\frac{1}{\mu_i} = \eta_i$$
  meaning that basically we're defining $\mu_i = -1/\nu_i$.
  Why the use the natural parameter may be problematic? in order to ensure that we get an expected value which is strictly positive we would need to impose a *restriction on the linear predictor*: *η_i must be strictly negative, in order to ensure that* $\mu_i \in \mathbb{R}^+$; this is somehow in contrast to the usage of linear predictor (which tipically can be anything in $\mathbb{R}$ and this simplifies a lot estimation).
  Imposing restriction on the linear predictor, means imposing restriction on the estimation procedure; that's why gamma example is interesting. Even if we know that canonical link functions comes with some interesting theoretical properties, not necessarily we must always use/rely canonical link function: tipically when using gamma distribution the canonical link function is not commonly used.
  Since we've to deal with expected value that must be strictly positive, the first

choice usually expoited is the exponential/log duo as in Poisson GLM, so we can set the expected value to the exponential of the linear predictor or equivalently the logarithm of the expected value equal to the linear predictor

$$\ln \mu_i = \eta_i$$

No problem btw, important is to use a link function which is meaningful and provides expected value on the proper range.

16.1.4 Log-likelihood and score function

As already seen once we've choosen a link function we can start looking at loglik and associated quantities. For the loglik will be:

$$l(\beta_0, \dots, \beta_p, \nu) = \sum_{i=1}^{n} \left\{ -\frac{\nu}{h(\eta_i)} y_i - \nu \ln h(\eta_i) + (\nu - 1) \ln y_i - \ln \Gamma(\nu) + \nu \ln \nu \right\}$$

While starting from the generic expression of score function we can particularize it to take into account specificities of gamma distribution in terms of expected value and variance:

$$\begin{aligned} U_j(\boldsymbol{\beta}) &= \frac{\partial}{\partial \beta_j} l(\beta_0, \dots, \beta_p, \nu) \\ &= \sum_{i=1}^{n} \left\{ \frac{y_i - \mathrm{E}[Y_i|\mathbf{x}_i]}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathrm{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} x_{ji} \right\} \\ &= \sum_{i=1}^{n} \left\{ \nu \frac{y_i - \mu_i}{\mu_i^2} \frac{\partial \mu_i}{\partial \eta_i} x_{ji} \right\} \end{aligned}$$

The only non explicit part is the first partial derivative of the link function $\frac{\partial \mu_i}{\partial \eta_i}$, which obv depends on the chosen link function. If we use the exponential $h(\eta_i) = \exp(\eta_i)$ we'll have a further semplification since $\frac{\partial \mu_i}{\partial \eta_i} = \exp(\nu_i) = \mu_i$ and so

$$U_j(\boldsymbol{\beta}) = \sum_{i=1}^{n} \nu \frac{y_i - \mu_i}{\mu_i} x_{ji}$$

16.1.5 Fisher information matrices

By not using canonical link function (as tipically occurs) here we'll have two different expression, one for the observed, the other for the expected. For the observed:

$$\begin{aligned} i_{jl}(\boldsymbol{\beta}) &= -\frac{\partial^2}{\partial \beta_j \beta_l} l(\beta_0, \dots, \beta_p, \nu) \\ &= \sum_{i=1}^{n} \left\{ \frac{x_{ji} x_{li}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \left(\frac{\partial \mathrm{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \right)^2 \right\} - \sum_{i=1}^{n} \left\{ (y_i - \mathrm{E}[Y_i|\mathbf{x}_i]) \frac{\partial}{\partial \beta_l} \left(\frac{x_{ji}}{\mathrm{Var}[Y_i|\mathbf{x}_i]} \frac{\partial \mathrm{E}[Y_i|\mathbf{x}_i]}{\partial \eta_i} \right) \right\} \\ &= \sum_{i=1}^{n} \left\{ \frac{\nu x_{ji} x_{li}}{\mu_i^2} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 \right\} - \sum_{i=1}^{n} \left\{ (y_i - \mu_i) \frac{\partial}{\partial \beta_l} \left(\frac{\nu x_{ji}}{\mu_i^2} \frac{\partial \mu_i}{\partial \eta_i} \right) \right\} \end{aligned}$$

While for the expected

$$I_{jl}(\boldsymbol{\beta}) = E[i_{jl}(\boldsymbol{\beta})] = \sum_{i=1}^{n} \left\{ \frac{\nu x_{ji} x_{li}}{\mu_i^2} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 \right\}$$

Again if we choose exp/log link function, and thus $\mu_i = \exp \eta_i$ we get that expected fisher information matrix will have elements with simple structure since $\left(\frac{\partial \mu_i}{\partial \eta_i} \right) = \mu_i^2$ and so

$$I_{jl}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \nu x_{ji} x_{li}$$

16.1.6 Matrix representation

We can get a matrix representation for both score vector and expected fisher information; as seen when dealing with non-canonical link function equation becomes slightly more complicated, but by properly defining the diagonal element of a $\mathbf{Q}$ matrix, we can get an expressions very similar (in the structure) to the equivalent expressions for GLM for binary outcome with non-canonical link functions:

- the Score function

$$U(\boldsymbol{\beta}) = \mathbf{X}^\top \mathbf{Q} \left[\mathbf{y} - \boldsymbol{\mu}\right]$$

- the Expected Fisher information matrix:

$$I(\boldsymbol{\beta}) = \mathbf{X}^\top \mathbf{Q}\mathbf{W}\mathbf{Q}\mathbf{X}$$

where:

$$\mathbf{Q} = \begin{bmatrix} \frac{\nu}{\mu_1^2}\frac{\partial \mu_1}{\partial \eta_1} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\nu}{\mu_n^2}\frac{\partial \mu_n}{\partial \eta_n} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \frac{\mu_1^2}{\nu} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\mu_n^2}{\nu} \end{bmatrix}$$

That is in $\mathbf{Q}$ on the diagonal has ν/μ_1^2 which is 1/variance of the dependent variable multiplied for the first partial of the expected value, while on the main diagonal of $\mathbf{W}$ we the conditional variances

Just for completeness if we use the canonical link function, $\mu_i = -\frac{1}{\eta_i}$, and $\frac{\partial \mu_i}{\partial \eta_i} = \mu_i^2$, then we have a simplification where $\mathbf{Q}$ matrix disappears (being diagonal ν being substituted by the constant itself):

$$\begin{aligned} \mathbf{Q} &= \nu \mathbf{I}_n \\ U(\boldsymbol{\beta}) &= \nu \mathbf{X}^\top \left[\mathbf{y} - \boldsymbol{\mu}\right] \\ I(\boldsymbol{\beta}) &= \nu^2 \mathbf{X}^\top \mathbf{W}\mathbf{X} \end{aligned}$$

Important remark 133. Once again we can see that each member of GLM is characterized by specific probabilistic and systematic component, the general structure for likelihood, loglikelihood, score function and expected Fisher information matrix we can recognize a unifying structure.

16.1.7 Maximum likelihood estimation of $\beta_0, \dots, \beta_p$

In terms of ML estimation to cut long story short (same stuff as usual):

- we have to use numerical optimisation techniques (Newton-Raphson/Fisher scoring) to maximize the log-likelihood
- if the model is correctly specified (*adequate*), then the maximum likelihood estimators are asymptotically well behaved (*Gaussian*, *unbiased* and *efficient*)

$$\hat{\mathbf{B}} \xrightarrow{d} MVN_{p+1}\left(\boldsymbol{\beta}, I(\boldsymbol{\beta})^{-1}\right)$$

- once estimated the betas we can get estimates of conditional expected values by plugging in the linear predictor and applying the link function:

$$\hat{\mathrm{E}}[Y_i|\mathbf{x}_i] = \hat{m}_i = h\left(\mathbf{x}_i^\top \hat{\mathbf{b}}\right)$$

16.1.8 Saturated model and deviance

Some quick summary if we compute the saturated model and debiance associated to GLM based on gamma distribution. Here we moved back to the situation where we have 1 unit for each covariate pattern.

So assuming that each unit is characterised by a different covariate pattern, we move toward a model that has as many parameters as covariate patterns (one for each unit) where the estimated expected value will coincide with the observed value y_i and thus perfectly fit the

data (not useful from a practical pov, since it doesn't tell us why some units are characterized by a specific expected value, but is useful for benchmarking for any fitted model);

$$
\begin{aligned}
l(\mu_1, \ldots, \mu_n, \nu) &= \sum_{i=1}^{n} \left\{ -\frac{\nu}{\mu_i} y_i - \nu \ln \mu_i + (\nu - 1) \ln y_i - \ln \Gamma(\nu) + \nu \ln \nu \right\} \\
U_i(\boldsymbol{\mu}) &= \frac{\partial}{\partial \mu_i} l(\mu_1, \ldots, \mu_n, \nu) = \nu \frac{y_i - \mu_i}{\mu_i^2} \\
\hat{m}_i &= y_i \\
D &= 2 \left[l(y_1, \ldots, y_n, \nu) - l(\hat{\mathbf{b}}, \nu) \right] = \ldots = \\
&= 2\nu \sum_{i=1}^{n} \left[\frac{y_i - \hat{m}_i}{\hat{m}_i} - \ln \frac{y_i}{\hat{m}_i} \right] \cong \nu \sum_{i=1}^{n} \left[\frac{y_i - \hat{m}_i}{\hat{m}_i} \right]^2
\end{aligned}
$$

where $\hat{m}_i = h\left(\mathbf{x}_i^\top \hat{\mathbf{b}}\right)$.
Peculiar of the gamma is the fact that the *residual deviance* D of a GLM with Gamma probabilistic component *cannot be used as a goodness of fit test statistic*: this is due to due to the presence of a nuisance parameter ν in the final equations. It's similar to what happens for gaussian linear regression model.
In the last equation we simplified the expression of D with something which resemble a pearson χ^2 statistic.
As D cannot be used to perform GOF test but at least its components/ingredients are exploited to come up with residuals estimation

16.1.9 Residuals and estimation of ν

These are

$$
\begin{aligned}
\text{Deviance residuals: } e_i^D &= \text{sign}(y_i - \hat{m}_i) \sqrt{2\hat{\nu} \left[\frac{y_i - \hat{m}_i}{\hat{m}_i} - \ln \frac{y_i}{\hat{m}_i} \right]} \\
\text{Pearson residuals: } e_i^P &= \sqrt{\hat{\nu}} \frac{(y_i - \hat{m}_i)}{\hat{m}_i}
\end{aligned}
$$

We have that:

- To evaluate these residuals we need an estimate for nuisance parameter ν which implement the usual choice

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

  or, equivalently

  $$\hat{\phi} = \frac{1}{\hat{\nu}} = \frac{\sum_{i=1}^{n} \left[\frac{y_i - \hat{m}_i}{\hat{m}_i} \right]^2}{n - p - 1}$$

- if the model is *correctly specified* and the true value of ν is *large*, it is possible to prove that the (standardised) residuals are asymptotically independent, homoscedastic and with a Gaussian distribution.
  So we can use them to explore the residuals in order to understand if the model is adequate or not by looking at residuals vs fitted plot, scale location plot or qqplot.

16.1.10 Testing linear hypotheses on $\beta_0, \ldots, \beta_p$

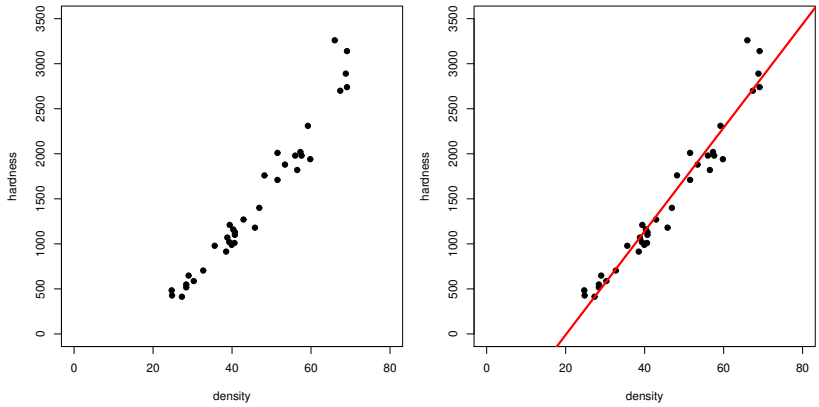
Again here to test $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{t}$ nothing new, we can use:

- Likelihood ratio test statistic

  $$-2 \ln \frac{l(\hat{\mathbf{b}}, \hat{\nu})}{l(\hat{\mathbf{b}}_{H_0}, \hat{\nu})} \Bigg| H_0 \xrightarrow{d} \chi^2_{(q)}$$

- or its approximation via the Wald test statistic

  $$\left[\mathbf{K}\hat{\mathbf{b}} - \mathbf{t} \right]^\top \left[\mathbf{K} \left(\widehat{I(\boldsymbol{\beta})} \right)^{-1} \mathbf{K}^\top \right]^{-1} \left[\mathbf{K}\hat{\mathbf{b}} - \mathbf{t} \right] \Bigg| H_0 \xrightarrow{d} \chi^2_{(q)}$$



(a) (b)

Figure 16.2: Data and gaussian model

In both cases, if the *model is correctly specified*, the*sample size is sufficiently large* (as in other GLM different from gaussian), χ^2 distributions (with q degrees of freedom) can be used to approximate the rejection region for H_0/to approximate the p-value associated with the observed value of the test statistic.

16.2 Worked example

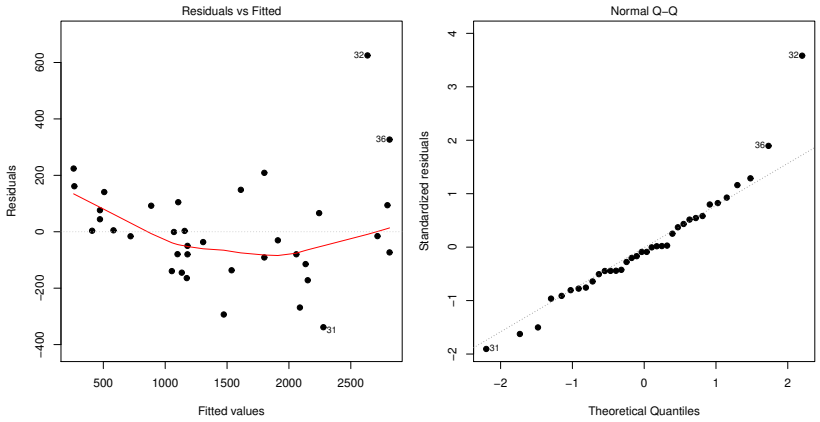
16.2.1 Data and estimates

Same dataset used for transformation to stabilize the variance/log normal regression on $n = 36$ observation on pieces of wood with:

- `hardness`: hardness of an hardwood timber (Y)
- `density`: density of an hardwood timber (X)

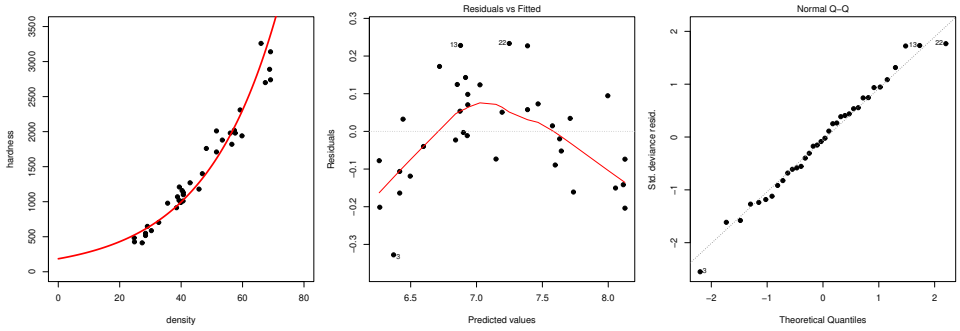
Idea is to study to what extent the hardness depend on the density

- the data is plotted in fig 16.2 a) while a gaussian linear model estimate is reported in b). Despite the fact that hardness cannot take negative values, the linear model predict negative values below a certain treshold;
- graphical residual analysis of the gaussian model is in fig 16.3 and shows a possible violation of the omoschedasticity assumption and thus the inadequacy of gaussian regression model;
- let's see what happens if we move to gamma glm with systematic component based on exp/log link function $g(\mu_i) = \ln \mu_i = \beta_0 + \beta_1 x_i$ in fig 16.4 with its residual analysis as well. In this case the estimated nuisance parameter $\hat{\nu} = 55.669$ (we can think of it as *sufficiently large*). From the residual we note there's something wrong (residual not spread around zero) and especially possible nonlinear contribution of the regressor to η_i (residuals have constant variability, but nonconstant moving average)
- a second gamma model with systematic component using a quadratic effect $g(\mu_i) = \ln \mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$ is in figure 16.5. We see that with the estimated function approximate more sinuously the data; the introduction of the quadratic term fixes the residuals vs fitted and the qqplot is close to a straight line as well.
  The change in linear predictor yields to a change in the estimate of $\hat{\nu} = 98.242$ which is even more large. Other way to introduce nonlinearity with splines will be seen next. According to this plots, a model with a quadratic predictor, which is still linear in the parameter, combined with a logarithmic link function, seems adequate from the residuals pov
  `Coefficients`



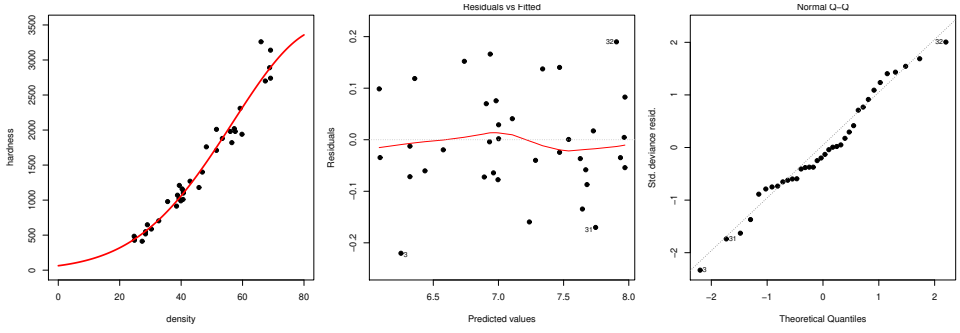
(a) (b)

Figure 16.3: Gaussian residual analysis



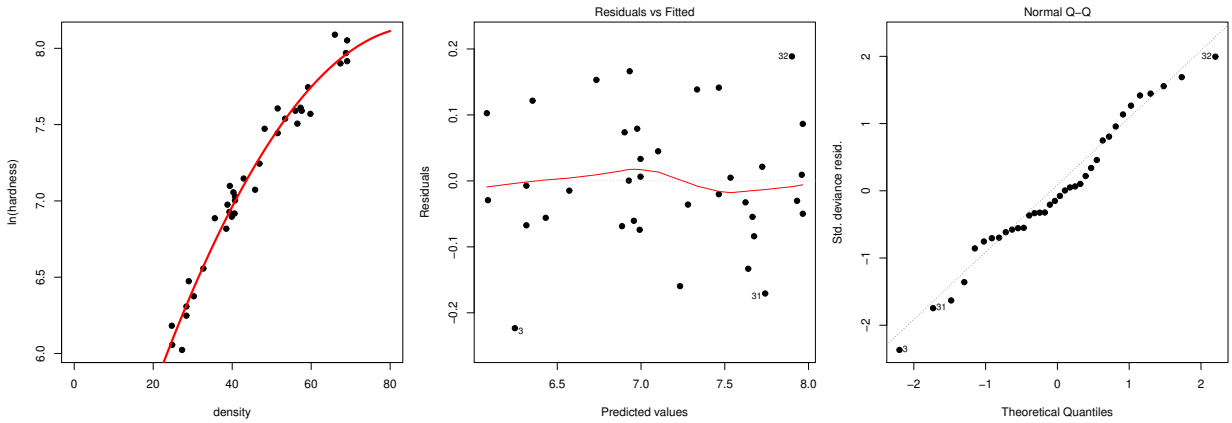
(a) (b) (c)

Figure 16.4: The gamma estimate



(a) (b) (c)

Figure 16.5: Gamma 2



(a) (b) (c)

Figure 16.6: Lognormal model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.1470	0.2089	19.85	0.0000
density	0.0913	0.0093	9.80	0.0000
I(density^2)	-0.0005	0.0001	-5.33	0.0000

All regression coefficients are significantly larger than zero. The impact of unit change of density will be multiplicative and will depend on the starting point of density (employing somehow the exponential of coefficients): here there's a significant quadratic effect on the log scale.

- alternatively: let's consider again (as in box-cox part) a log transformation of Y as the dependent variable (which was chosen as proper transformation in order to stabilize the variance.
  We fitted a lognormal regression model $\ln Y \sim N\left(\delta_0 + \delta_1 x_i + \delta_2 x_i^2, \sigma^2\right)$ (so quadratic model on the log density) in figure 16.6. According to residuals plots, a lognormal regression model with a quadratic component seems adequate.
  Recall gaussian regression model on the log of y means a lognormal regression model on the original variable: remember this. Sometimes in practical situation we can use model characterized by different probabilistic component

16.2.2 Comparison between Gamma and Lognormal random variables

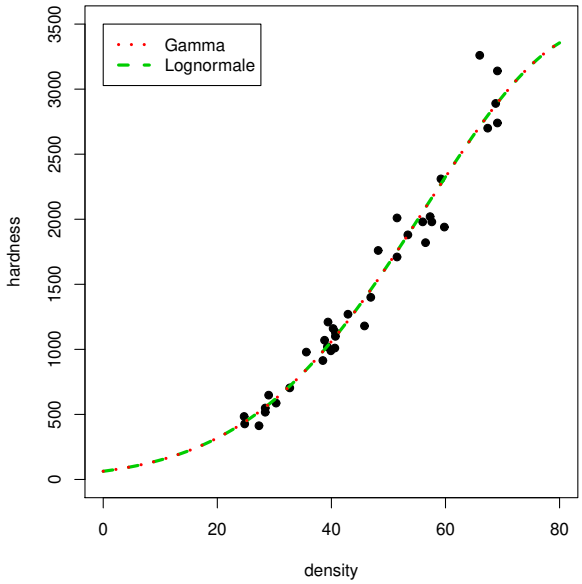
What's the connection between lognormal and gamma distribution

- the lognormal distribution *does not belong to an exponential family* expressed in canonical form; thus lognormal regression models do not belong to the GLM family. This is not a problem as long as the model is adequate;
- gamma and lognormal distribution both have conditional variance proportional to the square of the expected value so conditional coefficient of variations are constant/independent of expected value. For lognormal model this is:

$$\implies \mathrm{CV}\left[Y_i\right] = \sqrt{\exp(\sigma^2) - 1}$$

- however, when the (conditional) coefficient of variation is lower than 0.8, the *lognormal distribution and the Gamma distribution are very close*. In this example we calculate the CV with different formulas seen:

$$\hat{\nu} = 98.242 \implies \widehat{\mathrm{CV}} = \frac{1}{\sqrt{98.242}} = 0.101$$
$$\hat{\sigma}^2 = 0.010 \implies \widehat{\mathrm{CV}} = \sqrt{\exp(0.010) - 1} = 0.100$$



(a)

Figure 16.7: Gamma vs lognormal

In situations like this, in terms of residuals, often either both methods produce an adequate model or neither one or the other leads to an adequate one.
Indeed in the plot 16.7 we see that by comparing the estimated expected value they're almost equivalent; according to the AIC, the Gamma GLM seems slightly better than the lognormal regression model (455.6448 vs 455.7336).
So:

- when CV is small they'll behave similarly
- when CV is large they'll behave differently and it could be that neither one of the two is adequate

16.3 Data transformation vs GLM

Important remark 134. **Data transformation** (in particular when applied to the dependent variable) is a practical solution to (simultaneously) address three issues:

- obtaining a new dependent variable whose range is comparable to the range of the linear predictor
- stabilising the conditional variance of the dependent variable
- obtaining a new dependent variable with a symmetric (or, even better, Gaussian) conditional distribution

The first issue is substantial and it is related to a proper definition of the effects of the regressors on the conditional expecetd value of the dependent variable.
The second and third issue are mostly related to the efficiency of the inferential procedures.
When the *interpretation of the results on the original scale* of the dependent variable is important, *transforming the dependent variable may introduce some difficulties*. Furthermore often is difficult to find transformation which fix all the problems. Nevertheless, data transformation can be useful, especially for *explorative* data analysis: eg if we know log transform this can be an hint to move in a GLM using gamma distribution.

Important remark 135. Generalised linear models allow to act separatedly on pieces of the model:

- address the first issue through the choice of the systematic component
- deal with heteroschedastic and asymmetric conditional distribution through the choice of the probabilistic component

Within the GLM framework, the systematic component can be chosen (almost) regardless of the probabilistic components thus allowing more flexibility. This does not mean that GLM is panacea for everything.

Chapter 17

Lab6 - Gamma GLM

Data on 4 chimpanzees who were trained to learn 10 words; for each chimps and words the variable recorded was the minutes taken (observation times couldn't be considered independent but for this practical session we ignore this fact and take them as coming from independent units).

```
library(MLGdata)

##
## Caricamento pacchetto:  'MLGdata'
## Il seguente oggetto è mascherato da 'package:carData':
##
##     Wool
## Il seguente oggetto è mascherato da 'package:nlme':
##
##     Orthodont

data(Chimps) ## ?Chimps
str(Chimps)

## 'data.frame': 40 obs. of  3 variables:
##  $ chimp: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  $ word : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ y    : num  178 60 177 36 225 345 40 2 287 14 ...

## y: time needed to learn a word (minutes)
## chimp: chimpanze (factor - 4 categories)
## word: word to be learned (factor - 10 categories)
```

17.1 Estimates

Using a standard linear model and looking at the residuals ...

```
## Gaussian linear model
m_normal_id <- glm(y ~ chimp + word, data = Chimps)
par(mfrow = c(1, 3))
plot(m_normal_id, which = 1, pch = 19)
plot(m_normal_id, which = 3, pch = 19)
plot(m_normal_id, which = 2, pch = 19)
```

there is a clear violation of the homoscedasticity assumption and it's not a classical scatter in residuals vs fitted; scale location suggest variability increases with expected value, while qqplot deviates from straight line in the last part (heavy tail with values above the line

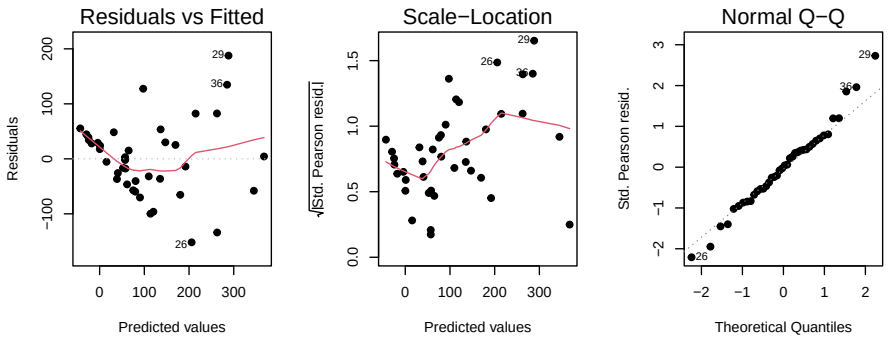
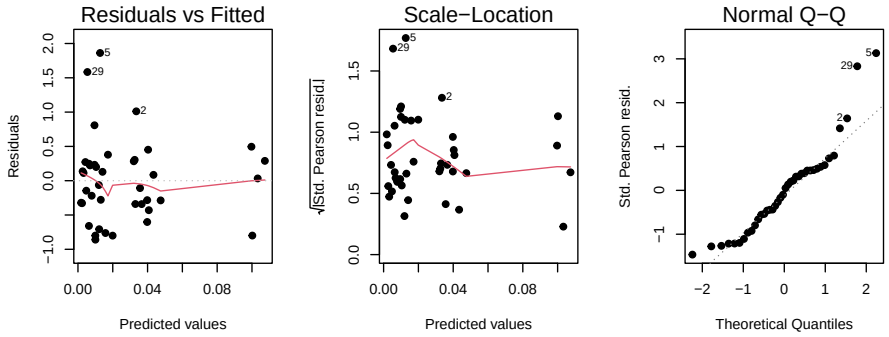


Figure 17.1: Gaussian model residuals

meaning that the observed quantiles tend to be larger than theoretical ones).
All these plots suggest that gaussian model is not adequated; common choice for time data like in this case is the use of gamma glm. We now fit it with canonical link function (there could be problem for linear predictors which are positive, `glm` should gives warning not in this case)

```
## Gamma GLM with canonical link function
m_gamma_can <- glm(y ~ chimp + word, family = Gamma, data = Chimps)
par(mfrow = c(1, 3))
plot(m_gamma_can, which = 1, pch = 19)
plot(m_gamma_can, which = 3, pch = 19)
plot(m_gamma_can, which = 2, pch = 19)
```



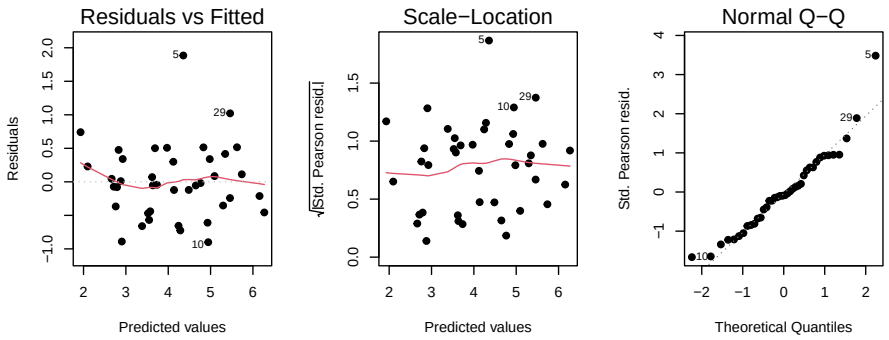
The use of a gamma GLM with canonical link function leads to an improvement, even though some "strange" patterns are still present in the residual plots:

- in the residual vs fitted we do not have predicted value that are negative (good), the residuals seems scattered around zero but variance seems to decrease somewhat
- in the scale-location the magnitude of standardized residuals seems more or less constant and the problem with heteroskedasticity seems to be solved
- in terms of qqplot there are some issues.

We try to change link function. We have `identity` and `log`. We go with this second:

```
## Gamma GLM with logarithmic link function
m_gamma_log <- glm(y ~ chimp + word, family = Gamma(link = log), data = Chimps)
```

```
par(mfrow = c(1, 3))
plot(m_gamma_log, which = 1, pch = 19)
plot(m_gamma_log, which = 3, pch = 19)
plot(m_gamma_log, which = 2, pch = 19)
```



residual plots for the gamma GLM with logarithmic function suggest that this third model is most adequate (there's still a heavy tail on the qqplot).
Recall that both for gaussian model and glm with gamma probabilistic component we cannot perform a proper goodness of fit test because both model are characterized by the presence of the nuisance parameters (σ^2 for gaussian , ν for gamma) and this prevent us to use residual deviance to perform GOF tests. So to judge adequacy of the model we can only rely on graphical/visual tools.
Here the choice link function, assuming that both the model are adequate, we can evaluate AIC (since different link function are adopted)

```
## Choice of the link function
AIC(m_gamma_can)

## [1] 424.0369

AIC(m_gamma_log)

## [1] 418.4183
```

Also according to the AIC, other than residuals, the logarithmic link function should be preferred. See the summary

```
summary(m_gamma_log)

##
## Call:
## glm(formula = y ~ chimp + word, family = Gamma(link = log), data = Chimps)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.67657  -0.45103  -0.04876   0.30739   1.28473
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.4585     0.3754  14.540 2.73e-14 ***
## chimp2       -0.9734     0.2945  -3.305 0.002686 **
## chimp3       -0.8078     0.2945  -2.743 0.010679 *
## chimp4       -0.1128     0.2945  -0.383 0.704786
## word2        -1.7707     0.4657  -3.803 0.000744 ***
## word3        -0.3649     0.4657  -0.784 0.440070
```

```
## word4        -1.8214     0.4657  -3.911 0.000559 ***
## word5        -1.1018     0.4657  -2.366 0.025409 *
## word6         0.2786     0.4657   0.598 0.554607
## word7        -1.7249     0.4657  -3.704 0.000963 ***
## word8        -2.5546     0.4657  -5.486 8.28e-06 ***
## word9         0.8109     0.4657   1.741 0.093009 .
## word10       -0.5137     0.4657  -1.103 0.279722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.4336663)
##
##     Null deviance: 60.378  on 39  degrees of freedom
## Residual deviance: 14.972  on 27  degrees of freedom
## AIC: 418.42
##
## Number of Fisher Scoring iterations: 12

# exp(coef(m_gamma_log))
```

Here the exponential of the intercept will be the expected learning time for the first chimpanzee on the first word:

- here differently from poisson and binomial model we have an asymptotic t (not z) because to get the asymptotic standard error we have a nuisance parameter which have to replaced with an estimate (in the expected fisher information matrix), so the asymptotic distribution will be a T; again when degrees of freedom tends to ∞ the differences tends to vanish
- new here with respect to GLM for binary outcome or poisson is the estimate presence of a `dispersion parameter` (intended to be our *nuisance parameter*) which is 0.43.
- here null and residual deviances are not the actual deviances but are the scaled deviances. That is standard deviance is

$$D = 2\nu \sum_{i=1}^{n} \left[\frac{y_i - \hat{m}_i}{\hat{m}_i} - \ln \frac{y_i}{\hat{m}_i} \right]$$

  which is a metric measuring distance between observed and predicted value, while the scaled deviance reported by R just removes the ν

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

  In the gaussian model we get scaled residual deviance as well (sum of raw residuals to the square, not divided by σ^2 which is the actual deviance)

To get an estimate for the nuisance/dispersion parameter we can use the expression

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

so we can start from the scaled pearson residual $\frac{(y_i - \hat{m}_i)}{\hat{m}_i}$ (scaled because ignoring $\sqrt{\hat{\nu}}$, sum their squares, and put them below degrees of freedom.
In R we have a function which can extract a residual and that we can specify pearson residual but returns scaled pearson residuals by default:

```
## WARNING: the residuals.glm function ignore the presence of the
## nuisance parameter when asked to compute the Pearson residuals
## pearson residuals for GLMs with nuisance parameters are actual
## scaled pearson residual, ignoring nu
residuals_p <- residuals(m_gamma_log, type = "pearson")

## This is how they are actually calculated
```

```
check <- (Chimps$y - predict(m_gamma_log, type = "response")) /
    predict(m_gamma_log, type = "response")
all(residuals_p == check)

## [1] TRUE
```

So to apply the formula to compute $\hat{\nu}$, we have that $n - p - 1$ is 40 observation and we have 3 dummy for the chimps and 9 for the words so $40 - 12 - 1 = 27$ which can be found in the object `df` of the model summary:

```
## n-p-1 is in summary(m_gamma_log)$df[2], so to compute the estimate of nu
(nu <- summary(m_gamma_log)$df[2] / sum((residuals_p)^2))

## [1] 2.305921
```

2.30 is different from what reported using the `summary` (which was 0.43): what the glm function call dispersion parameter is not our nuisance parameter ν, but is a dispersion parameter for the gamma distribution coinciding with $\phi = 1/\nu$:

```
1 / nu

## [1] 0.4336663

1 / summary(m_gamma_log)$dispersion

## [1] 2.305921
```

Estimation of the conditional coefficient of variation, according to the properties of the gamma is $1/\sqrt{\nu}$, so we have two way consistently with what seen above

```
sqrt(summary(m_gamma_log)$dispersion)

## [1] 0.6585334

(cv_gamma_log <- 1 / sqrt(nu))

## [1] 0.6585334
```

Here the estimated conditional cv is quite small (below the rule of thumb treshold of 0.8) so lognormal model and gamma should behave similarly (and thus even lognormal model should provide a satisfactory fit, as done by the gamma glm).

17.2 Hypothesis testing

We may be interested in checking if there are differences both in the ability of the chimps and or on words difficulties.
We start by independence hypothesis (no differences due to chimps or words)

```
## null model: common learning time for all the rows to exp(4.67)?
summary(m_gamma_log0 <- glm(y ~ 1,
                            family = Gamma(link = log),
                            data = Chimps))

##
## Call:
## glm(formula = y ~ 1, family = Gamma(link = log), data = Chimps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4502  -1.4079  -0.7327   0.5468   1.9718
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.6763     0.1879   24.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.41206)
##
##     Null deviance: 60.378  on 39  degrees of freedom
## Residual deviance: 60.378  on 39  degrees of freedom
## AIC: 457.24
##
## Number of Fisher Scoring iterations: 6

## use of anova function to compare nested models
anova(m_gamma_log0, m_gamma_log, test = "Chisq") # Some authors suggest using F

## Analysis of Deviance Table
##
## Model 1: y ~ 1
## Model 2: y ~ chimp + word
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        39     60.378
## 2        27     14.972 12   45.406 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

so in general, there are significant differences in the learning times.
Some authors suggest using F distribution instead of Chisq: for gamma glm we have loglik including a nuisance parameter so we have to replace it with an estimate (as we did for gaussian regression model, where theoretically distribution is chisq if σ^2 is known, but since it's not and we replace with an estimate $\hat{\sigma^2}$ we have to change the test to get an F). In the context of gaussian regression model we have exact results (unrelated to n chisq or F will be the distribution of the statistic, depending on σ^2 being known or not). But with other glm we always deal with asymptotic distrbution and when degrees of freedom goes to infinity, differences between chisquare and F distribution tend to vanish and leads to conclusion that are similar. Here we used the chisq.
Checking for differences among chimpanzes:

```
summary(m_gamma_log2 <- glm(y ~ word,
                            family = Gamma(link = log),
                            data = Chimps))

##
## Call:
## glm(formula = y ~ word, family = Gamma(link = log), data = Chimps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4630  -0.6394  -0.1157   0.4218   1.4801
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0938     0.3900  13.062 6.52e-14 ***
## word2        -1.7615     0.5515  -3.194  0.00329 **
## word3        -0.3231     0.5515  -0.586  0.56240
## word4        -1.9367     0.5515  -3.512  0.00143 **
## word5        -0.8669     0.5515  -1.572  0.12646
## word6         0.3594     0.5515   0.652  0.51954
## word7        -1.7351     0.5515  -3.146  0.00372 **
## word8        -2.8165     0.5515  -5.107 1.72e-05 ***
```

```
## word9         0.6620     0.5515   1.200  0.23940
## word10       -0.6540     0.5515  -1.186  0.24496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.6083111)
##
##     Null deviance: 60.378  on 39  degrees of freedom
## Residual deviance: 21.192  on 30  degrees of freedom
## AIC: 427.33
##
## Number of Fisher Scoring iterations: 6
```

```
anova(m_gamma_log2, m_gamma_log, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ word
## Model 2: y ~ chimp + word
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        30     21.192
## 2        27     14.972  3   6.2204 0.002473 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There are significant differences in the learning times among (at least two) chimpanzes, after controlling for the word to be learned.
Checking for differences among word

```
summary(m_gamma_log3 <- glm(y ~ chimp,
                            family = Gamma(link = log),
                            data = Chimps))
```

```
##
## Call:
## glm(formula = y ~ chimp, family = Gamma(link = log), data = Chimps)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5444  -1.2342  -0.8477   0.4142   2.5240
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.9156     0.3785  12.988 3.75e-15 ***
## chimp2       -0.9211     0.5352  -1.721   0.0939 .
## chimp3       -0.5373     0.5352  -1.004   0.3221
## chimp4        0.1539     0.5352   0.288   0.7753
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.432443)
##
##     Null deviance: 60.378  on 39  degrees of freedom
## Residual deviance: 53.430  on 36  degrees of freedom
## AIC: 457.34
##
## Number of Fisher Scoring iterations: 6
```

```
anova(m_gamma_log3, m_gamma_log, test = "Chisq")
```

```
## Analysis of Deviance Table
##
```

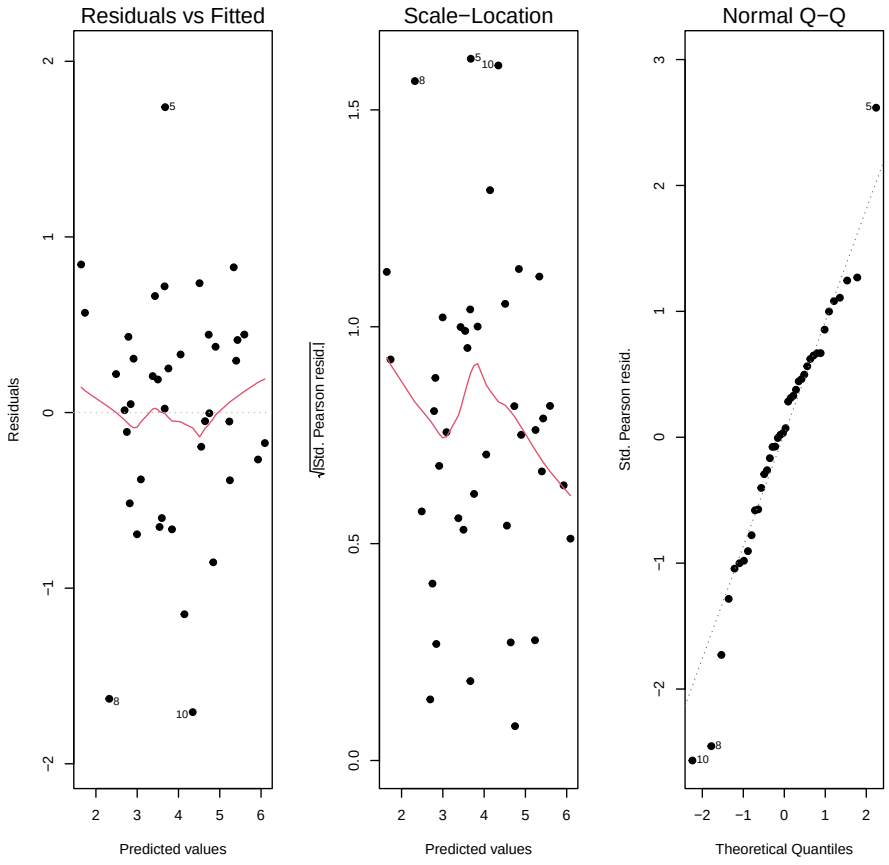
```
## Model 1: y ~ chimp
## Model 2: y ~ chimp + word
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        36     53.430
## 2        27     14.972  9   38.459 2.991e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There are significant differences in the learning times among (at least two) words, after controlling for the different ability of each chimpanze.

17.3 Comparison with lognormal regression

For this dataset we got low CV which means lognormal regression should perform similarly to gamma GLM; let's see

```
m_lognormal <- glm(log(y) ~ chimp + word, data = Chimps)
par(mfrow = c(1, 3))
plot(m_lognormal, which = 1, pch = 19)
plot(m_lognormal, which = 3, pch = 19)
plot(m_lognormal, which = 2, pch = 19)
```



As anticipated, residual plots for the lognormal regression model suggest that also this fourth model is adequate (similar to those from the gamma glm). Let's look at the output

```
summary(m_lognormal)

##
## Call:
## glm(formula = log(y) ~ chimp + word, data = Chimps)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.70580  -0.38195   0.03512   0.41762   1.73890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.2328     0.4610  11.352 8.74e-12 ***
## chimp2       -0.6813     0.3616  -1.884  0.07035 .
## chimp3       -0.5885     0.3616  -1.627  0.11528
## chimp4        0.1658     0.3616   0.459  0.65026
## word2        -1.8020     0.5718  -3.152  0.00395 **
## word3        -0.5001     0.5718  -0.875  0.38945
## word4        -1.8566     0.5718  -3.247  0.00311 **
## word5        -1.5556     0.5718  -2.721  0.01125 *
## word6         0.1976     0.5718   0.346  0.73233
## word7        -1.7320     0.5718  -3.029  0.00535 **
## word8        -2.9097     0.5718  -5.089 2.40e-05 ***
## word9         0.6941     0.5718   1.214  0.23529
## word10       -0.8880     0.5718  -1.553  0.13205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.6538002)
##
##     Null deviance: 68.675  on 39  degrees of freedom
## Residual deviance: 17.653  on 27  degrees of freedom
## AIC: 108.8
##
## Number of Fisher Scoring iterations: 2
```

It doesn't make sense to make comparison among the estimate coefficients with gamma glm, since we're usign different conditional distribution so they're expressed on different scale. We've a dispersion parameter which is connected with σ^2. To show how this was obtained, we used the (scaled) pearson residuals squared, summed and divided by df

```
resids <- residuals(m_lognormal, type = "pearson")
(sigma2 <- sum(resids^2) / summary(m_lognormal)$df[2])

## [1] 0.6538002

summary(m_lognormal)$dispersion

## [1] 0.6538002

## NOTE: also for the lognormal model fitted using the glm function
## (that is, a glm model with Gaussian probabilistic component for the
## logarithm of the dependent variable) the residuals.glm function
## ignore the nuisance parameter when asked to compute the Pearson
## residuals, thus returning the raw residuals for log(y)
##
## So check these are actually the scaled which here are defined as
## observed - fitted.
check <- log(Chimps$y) - predict(m_lognormal, type = "response")
all(resids == check)

## [1] TRUE
```

17.3.1 The AIC of lognormal model

Later we will compute AIC for the model and to do it we need the ML estimates for σ^2 (biased)

```
(sigma2_ml <- sum(residuals(m_lognormal, type = "pearson")^2) / nrow(Chimps))

## [1] 0.4413152
```

For the estimation of the conditional coefficient of variation, for lognormal is $\sqrt{\exp\sigma^2 - 1}$ (according to properties of lognormal distribution)

```
(cv_lognormal <- sqrt(exp(sigma2) - 1))  # obtained using the unbiased estimate

## [1] 0.9606426

(cv_lognormal_ml <- sqrt(exp(sigma2_ml) - 1)) # using the ML estimate biased one

## [1] 0.7448158
```

Both the estimates are close to the 0.8 rule of thumb treshold we've mentioned (where gamma and lognormal behaves similarly), as seen above the residuals are ok and the use of lognormal could be fine.

To choose between Gamma GLM and lognormal model we have to adopt model selection criteria such as AIC. The matter here is that

```
AIC(m_lognormal)

## [1] 108.7952

AIC(m_gamma_log)

## [1] 418.4183
```

The AIC are very different but thing is that the one referring to the lognormal model is AIC for a gaussian model fitted on $\ln(Y)$, while the remaining AIC is for a gamma GLM fitted on the original Y; thus **the two AICs are not directly comparable** (they actually look at different outcome variables). In order to compare them we need all the AIC are computed with respect to the same dependent variable.

To adjust the AIC for lognormal regression model to make it comparable we have to add a constant which involves $\log(y)$ and, copypasteing from previous sections we have that for the loglikelihood

$$\begin{aligned} l_{\ln N}(\boldsymbol{\beta}, \sigma^2|Y) &= \sum_{i=1}^{n}\left\{\ln\left[\frac{1}{\sqrt{2\pi\sigma^2}}\right] + \ln\left[\frac{1}{y_i}\right] - \frac{(\ln y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi})^2}{2\sigma^2}\right\} \\ &= -\sum_{i=1}^{n}\ln y_i - \frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 + \frac{\sum_{i=1}^{n}(\ln y_i - \beta_0 - \beta_1 x_{1i} - \ldots - \beta_p x_{pi})^2}{2\sigma^2} \\ &= -\sum_{i=1}^{n}\ln y_i + l_N(\boldsymbol{\beta}, \sigma^2|\ln Y) \end{aligned}$$

while for the AIC we have

$$\begin{aligned} AIC_{\ln N}(M|Y) &= -2l_{\ln N}(\hat{\mathbf{b}}, \hat{s}|Y) + 2(p+2) \\ &= -2\left[-\sum_{i=1}^{n}\ln y_i + l_N(\hat{\mathbf{b}}, \hat{s}|\ln Y)\right] + 2(p+2) \\ &= -2l_N(\hat{\mathbf{b}}, \hat{s}|\ln Y) + 2(p+2) + 2\sum_{i=1}^{n}\ln y_i \\ &= AIC_N(M|\ln Y) + 2\sum_{i=1}^{n}\ln y_i \end{aligned}$$

```
## adjust the AIC
AIC(m_lognormal) + 2 * sum(log(Chimps$y))

## [1] 422.5233

## otherwise to recompute AIC using the actual definition (-2loglik
## +2p) we first compute the likelihood by hand for each unit using
## lognormal density (see ?dlnorm) at the maximum likelihood estimates
## for the expected value and sigma^2.
lik_lnorm <- rep(NA, nrow(Chimps))
for (i in seq_len(nrow(Chimps))){
    lik_lnorm[i] <- dlnorm(Chimps$y[i],
                           ## we use as mean the predicted value for the unit
                           meanlog = predict(m_lognormal)[i],
                           ## the sqrt of ML estimate for sigma^2 seen before
                           sdlog = sqrt(sigma2_ml))
}
-2 * sum(log(lik_lnorm)) + 2 * 14

## [1] 422.5233

## recalling that ...
AIC(m_gamma_log)

## [1] 418.4183
```

The gamma GLM seems to be better for the lognormal for this specific dataset.

Chapter 18

Enhancing the flexibility of GLMs via regularization and additive modelling: an introduction

Remark 187 (Idea). Way in which we can make GLM more flexible: aside from polinomial, we can add splines and penalized splines.
We can replace a variable with a basis representing it, procede to estimate and then nothing changes in terms of inferential procedures.

NB: Not part of the exam

18.1 P-splines for non-Gaussian outcomes

18.1.1 GLMs based on P-splines

The P-spline approach can be extended beyond Gaussian models exploiting the GLM framework, by:

- replacing the linear predictor in the systematic component of a GLM with *a linear basis expansion*; within p-splines we can use b-spline bases with a *large* number of knots:

$$g(\mathrm{E}\left[Y_i|x_i\right]) = \sum_{j=1}^{K+4} \theta_j b_j(x)$$

  (note that this quantity is still linear in the unknown parameteres).

- exploiting *penalized maximum likelihood estimation* (in order to control overfitting), something like in:

$$\ell(\boldsymbol{\theta}) - \frac{\lambda}{2}\boldsymbol{\theta}^t P \boldsymbol{\theta}$$

  For this maximization (*formulas are omitted for the sake of simplicity*):

  - general expressions for the penalized score function and the penalized expected Fisher information matrix can be easily obtained;
  - pseudo-Fisher scoring algorithms can be defined using the penalized score function and the penalized expected Fisher information matrix;
  - value of λ will be choosen using some model selection procedure; the *GCV* criterion can be easily extended, starting from the residual deviance;
  - hypothesis testing can be based on approximate asymptotic results

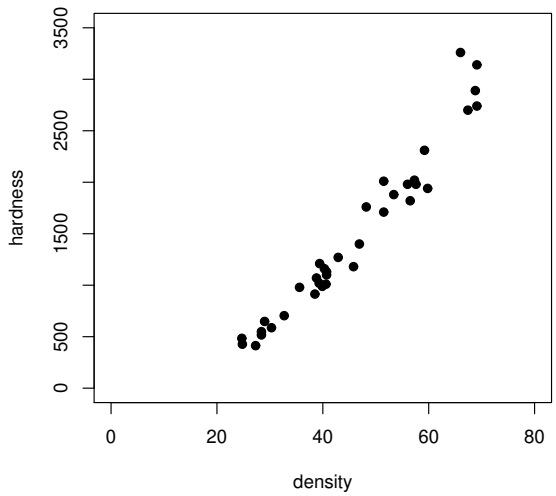


Figure 18.1: Timber data

18.1.2 Timber data example

To have an idea let's go back to the timber data example:

- data is in fig 18.1; we concluded the previous part noting that gamma glm with log link function and using a quadratic predictor was the best model to describe the connection between density
- what happens if we replace the quadratic predictor of the systematic component with something based on b-splines

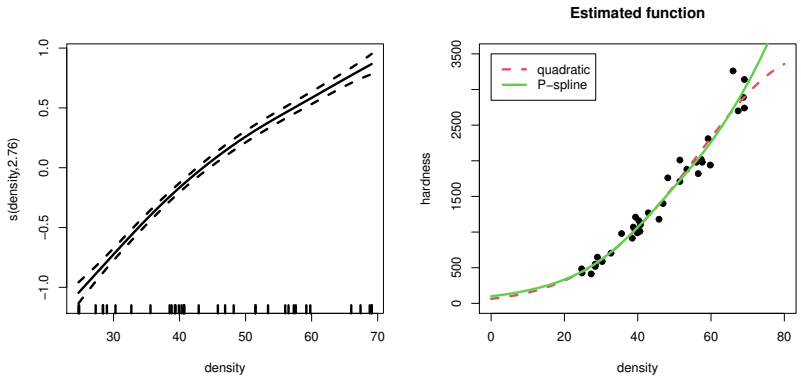
$$\ln \mu_i = \sum_{j=1}^{K+4} \theta_j b_j(x)$$

using the `gam` function (package: `mgcv`) using $K = 20$ knots with second-order differences penalty, (specifying the `family` argument as in `glm` to choose the probabilistic component), we get: `Parametric coefficients:`

```
              Estimate   Std.  Error   t value   Pr(>|t|)
 (Intercept)     7.141         0.017   431.680      0.000
Approximate significance of smooth terms:

                edf   Ref.df         F   p-value
 s(density)   2.763    3.373   352.900     0.000
```

Looking at the estimates:

  - gam computes an intercept (first part) plus an estiamated smooth function centered at 0 (second part)
  - edf are 2.7 (effective degrees of freedom, a quantification of complexity of fitted function - calculated with same rules as gaussian - associated to the *optimal value* for the smoothing parameter λ: `gam` internally performs an automatic selection for optimal value for lambda); the model with the quadratic effect had two degrees of freedom. The fitted gam function is only slightly more complex than a quadratic
  - the estimated function (p-value of s(density)) is statistically significant different from a constant line at 0, so there's an effect of density on hardness, as seen before



(a) (b)

Figure 18.2: Fitted function and prediction

To compare the model we have the following table:

Systematic component	$\ln L$	n. parameters	AIC
quadratic function	-223.8224	3(+1)	455.6448
P-spline	-222.7507	3.7632(+1)	455.0279

we have that

- the effective number of parameters for the P-spline model are only slightly larger than 3 (the fitted function is approximately quadratic);
- the model based on P-spline is only slighlty better than the model with a quadratic effect

- in figure 18.2 we have the plot of fitted function (note it's vertically centered around 0 and is represented on the scale of log expected value) and the models prediction/comparison with quadratic; they're very close and the main difference is on higher density where the green seems to catch better the last 4

18.2 Introduction to generalized additive models

Remark 188. Above we seen just one regressor; we exted the idea of applying splines to more than one regressor through GAM.
GLM can be seen as a special case of GAM: the key in the acronym is the replacement of *linear* with *additive*. What does this mean in the framework?
It has to do on how we define the systematic component.

18.2.1 Definition of a generalised additive model (GAM)

Remark 189. As usual let

- Y_i be r.v. that describes the possible value of the dependent variable on the i-th sample unit ($i = 1, \dots, n$)
- $x_{1i}, \dots, x_{pi}$ be the values of the regressors for the i-th sample unit

Definition 18.2.1. Generalized additive models (GAM) are statistical model for random samples $\mathbf{Y}$ characterised by:

- a more flexible **systematic/deterministic component** where linear predictors are replaced with additive predictors:

$$\eta_i = \alpha_0 + s_1(x_{1i}) + s_2(x_{2i}) + \dots + s_p(x_{pi})$$

η_i is expressed as sum of a constant α_0 (the intercept) plus p smooth function which are nonlinear, unknown and depending only on the i-th regressor ($i = 1, \ldots, p$). These functions as (additional) source of nonlinearity in the model
There's no further assumption on the functional form of $s_l(\cdot)$ (nonparametric model)

- **probabilistic components** similar to those of GLM: conditional distributions belonging to the same exponential family + conditional independence.
We still use a link function g with known functional form, differentiable and invertible, such that $g^{-1}(\cdot) = h(\cdot)$ and

$$g(\mathrm{E}[Y_i|x_{1i}, \ldots, x_{pi}]) = \alpha_0 + s_1(x_{1i}) + s_2(x_{2i}) + \ldots + s_p(x_{pi})$$
$$\mathrm{E}[Y_i|x_{1i}, \ldots, x_{pi}] = h(\alpha_0 + s_1(x_{1i}) + s_2(x_{2i}) + \ldots + s_p(x_{pi}))$$

18.2.2 Identifiability of additive predictors

Important remark 136. One point to consider before moving on estimation.
Thing is: any additive predictor is *identifiable up to constant shifts in the smooth functions* $s_l(\cdot)$

If we say s *must* be nonlinear, we have an issue with identifiability: we could add a constant to one of these s function and subtract the same constant from another of these s

$$s_1^*(x_{1i}) = s_1(x_{1i}) + c, \quad s_2^*(x_{2i}) = s_1(x_{1i}) - c$$

This will alter the actual value s_1^* and s_2^* but the resulting additive predictor will be the same as the original one:

$$\begin{aligned}\eta_i^* &= \alpha_0 + s_1^*(x_{1i}) + s_2^*(x_{2i}) + \ldots + s_p(x_{pi}) \\ &= \alpha_0 + s_1(x_{1i}) + s_2(x_{2i}) + \ldots + s_p(x_{pi}) \\ &= \eta_i\end{aligned}$$

Therefore the following restrictions are usually introduced in order to avoid this issue with identifiability/uniqueness:

$$\sum_i s_l(x_{li}) = 0, \quad l = 1, \ldots, p$$

18.2.3 Penalized ML estimation for GAM

A popular strategy to fit GAMs is based on an extension of the penalized ML approach described previously:

- we assume each smooth function $s_l(\cdot)$ is approximated using a spline function and represented using a specific set of m_l basis. As example, a B-splines with K knots (with K large):

$$s_l(x_{lj}) = \sum_{j=1}^{m_l} \theta_{lj} b_{lj}(x_{li})$$

Thus We define for each regressor a set of knots, we introduce a number of basis associated to the knots and replace each s_l with a linear basis expansion. In order to control for the actual flexibility it's introduced ...

- a specific penalty term: a matrix represenation is chosen for the parameters vectors $\boldsymbol{\theta}_l = (\theta_{l1}, \ldots, \theta_{lm_l})^\top$

$$J_l(\boldsymbol{\theta}_l) = \boldsymbol{\theta}_l^\top \mathbf{P}_l \boldsymbol{\theta}_l$$

example: first- or second-order differnces

- in order to control smoothness, a specific smoothing parameter $\lambda_l \geq 0$ is allowed for each smooth function

Tuning all the stuff above permits to smooth the function the right way.
So things come a little more complicated if we have more than one regressor: if we want to tune each smooth function for each regressor we have to choose several smoothing parameter (p).
What happens is that we'll have a penalized log-likelihood function:

$$pl_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_p) = \ln L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_p) + \sum_{l=1}^{p} \lambda_l J_l(\boldsymbol{\theta}_l)$$

in which :

- rather than havin 1 intercept and 1 parameter foreach regressor, we'll have an intercept plus a *set* of parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p$ for each regressor (each of which is associated with a specific set of basis), so we increase the columns of matrix $\mathbf{X}$ depending on number of knots and basis we're using.
- The penalty term is composed of the sum of several penalty term, one foreach smooth function

We have that:

- the actual form of $pl_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_p)$ and the possible presence of nuisance parameters/weights depend on the specific probabilistic component;
- general expressions for the penalized score function and the penalized expected Fisher information matrix can be easily obtained
- pseudo-Fisher scoring algorithms can be defined using the penalized score function and the penalized expected Fisher information matrix
- the *GCV* criterion can be easily extended, starting from the residual deviance (*note that an optimal value for the smoothing parameter of each $s_l(\cdot)$ can be selected*)
- hypothesis testing can be based on approximate asymptotic results (*although also in this context p-values are usually anticonservative*)

Again Formulas are omitted for the sake of simplicity.

18.2.4 Ozone example

18.3

In a study on air pollution in the Los Angels area, a researcher is interested in evaluating the effects of some regressors on the daily atmospheric ozone concentration `O3` (ppm - *count variable*). In particular, the following regressors are considered:

- `temp`: temperature (degree F)
- `ibt`: inversion base height (feet)
- `ibh`: inversion base temperature (feet)

we have:

- data referring to 330 different days are plotted in figure 18.3 (`O3` dependent variable); `temp` and `ibt` seems to have an impact (nonlinear) on the dependent variable, with an increasing spread as well (due to the count nature of dependent)
- a poisson gam was estimated using `gam` function (package: `mgcv`) - default settings, somethin like

```
Parametric coefficients:
              Estimate   Std.  Error   t value   Pr(>|t|)
 (Intercept)      2.291         0.019   119.903      0.000
Approximate significance of smooth terms:

             edf   Ref.df         F   p-value
 s(wind)   2.324    2.972     5.060     0.145
 s(temp)   4.251    5.238   101.571     0.000
  s(ibt)   1.714    2.201     1.769     0.431
  s(ibh)   4.658    5.644    68.202     0.000
```

  The summary info in the second tables: we have still a `edf` resulting from the tuning (eg using GCV). in some cases we have more compex relationship (eg `ibh` and `temp` with quartic function) in other situation smoother/simpler functions.
  P-values can be used to test if the estimated smooth function is significantly different from a constant function (presence of an effect of the regressor)

- in 18.4 the actual estimated function, where we see that both for `wind` and `ibt` we have estimated function flat/close to constant line 0 (which is included in the confidence bands).
  On the contrary there are certain relation between `temp` and `ibh` with the dependent variable.

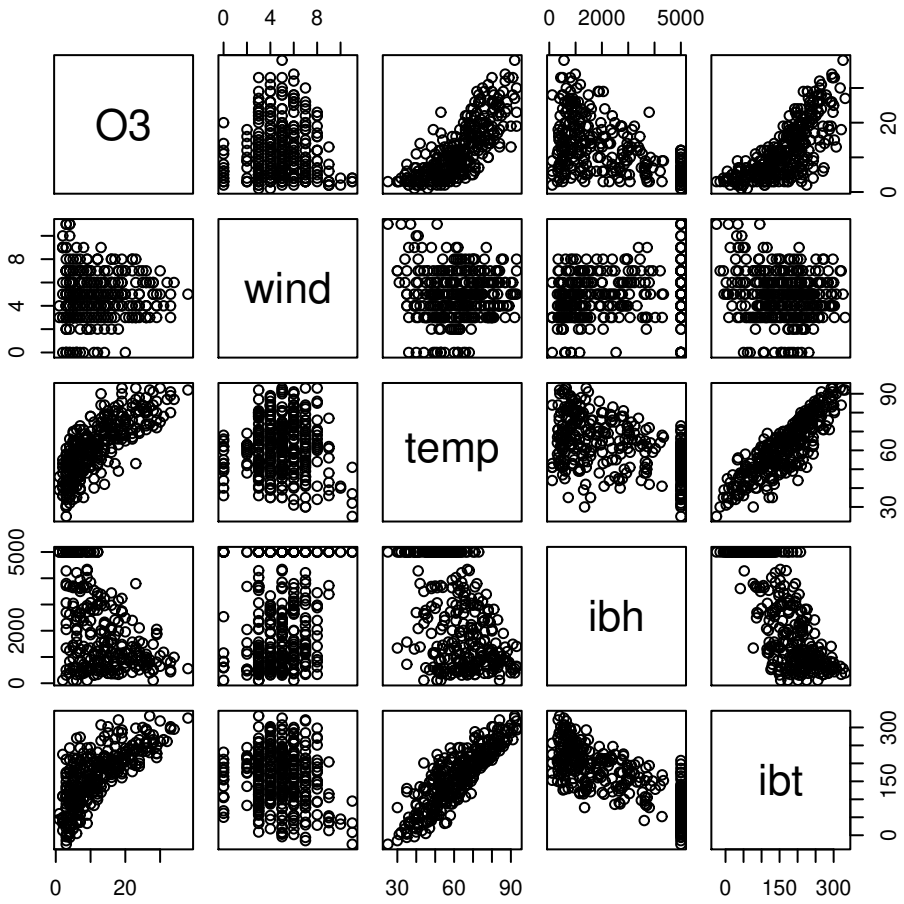


Figure 18.3: Ozone data

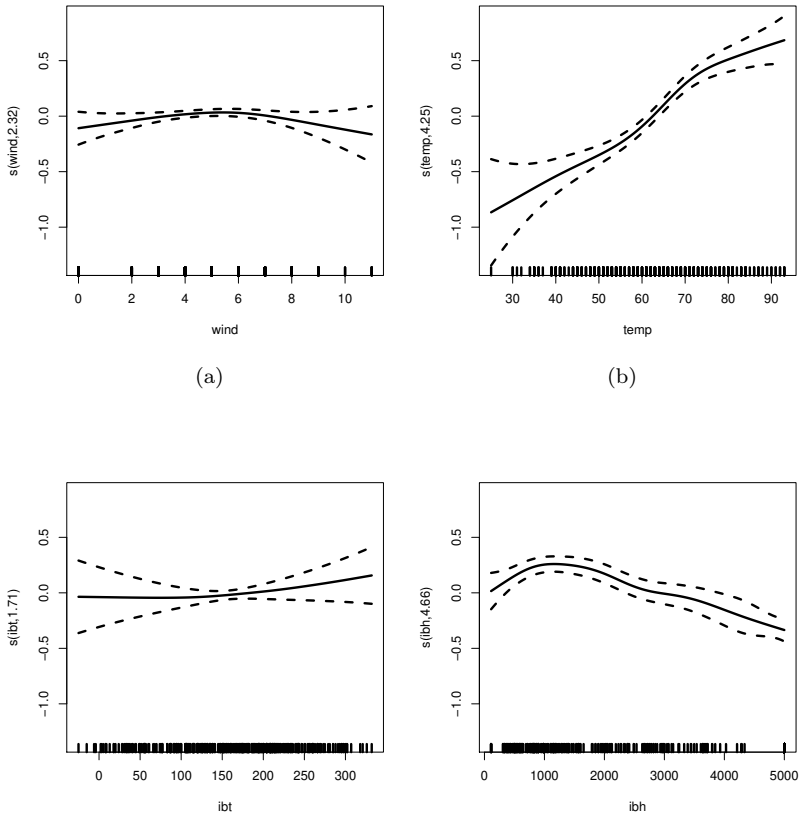


Figure 18.4: Estimated smooth functions with confidence bands

- for the comparison between GLM and GAM: recall that linear functions are a special case of spline functions, thus *a linear predictor can be obtained by introducing suitable linear constraints on the parameters of the linear basis expansions associated to a given additive predictor*

```
Model 1:  O3 ~ wind + temp + ibt + ibh
Model 2:  O3 ~ s(wind) + s(temp) + s(ibt) + s(ibh)
     Resid.  Df   Resid.  Dev        Df   Deviance   Pr(>Chi)
 1    325.0000      519.0159
 2    312.9443      456.3987   12.0557    62.6172     0.0000
```

  From above we conclude that *at least one of the regressors included in the GAM model has a significantly nonlinear effect on the dependent variable.*
  Furthermore *the GAM model is also better in terms of AIC*: we complex the model in terms of number of parameters but the fit get better as well

Predictor	$\ln L$	n. parameters	*AIC*
linear	-931.8987	5	1873.797
additive	-900.5901	13.947	1829.075

18.3 Further extensions

Several extensions of GAM models are possible:

- semiparametric models can be defined, by defining systematic components in which only some of the regressors have a nonlinear smooth effect:

$$\eta_i = \alpha_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + s_3(x_{3i}) \ldots + s_p(x_{pi})$$

  This, for example, allows the inclusion of categorical regressors in GAM models

- the additivity assumption can be overcome by allowing interactions, which can also be represented using smooth multivariate functions depending on more than one regressor

$$\eta_i = \alpha_0 + s_{12}(x_{1i}, x_{2i}) + s_3(x_{3i}) \ldots + s_p(x_{pi})$$