Stochastics Lab Course II

Khwam Tabougua Trevor

March 2019

INTRODUCTION

The "Stochastics Lab course II" is a practical Course for statistics and stochastics applications with R programming language. The course lasted for two weeks in March 2019. The report written on LaTeX, contains a description of the problem, a description of the methods used to solve the problem and a detailed discussion of the results.

CONTENTS

CHAPTER

ONE

TIDYVERSE

1.1 Problem description

R base tools can accomplish "almost" every programming tasks. However, when using large datasets or when implementing complex tasks(like graphs, maps, tidying, etc), things get complicated. We want to enhance our algorithms fo batter results or productivity. To this aim, we will use the Tidyverse package.

1.2 Methods

Tidyverse is a collection of packages for data manipulation, exploration and visualization. The core packages are **ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr, and forcats**, but we will only be using ggplot2, dplyr, tidyr, and tibble.

- **ggplot2** is a system for declaratively creating graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.
- **dplyr** is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges such as adding new variables (that are functions of existing variables), picking variables based on their names, selecting rows (based on their value), reducing multiple values down to a single summary, and changing the ordering of the rows.
- **tidyr** package goal is to help you create tidy data. Tidy data is data where each variable is in a column, each observation is a row, and Each value is a cell.
- **tibble** package goal is to use tibbles, which are modern take on data frames. They keep the features that have stood the test of time, and drop the features that used to be convenient but are now frustrating (i.e. converting character vectors to factors).

1.3 Results

(a) After loading and filtering the data childrenfinal.dta, we convert some variables (namely `tetanusmother`, `breastfeeding`, `wantedchild`, `anetalvisits`, and `placedelivery`) into double labeled <*dbl*> (doubles, or real numbers).

(b) The smooth line in figure 1.1 indicates that the `zstunt` is deteriorating between the ages of 0 and 20, before stabilizing.

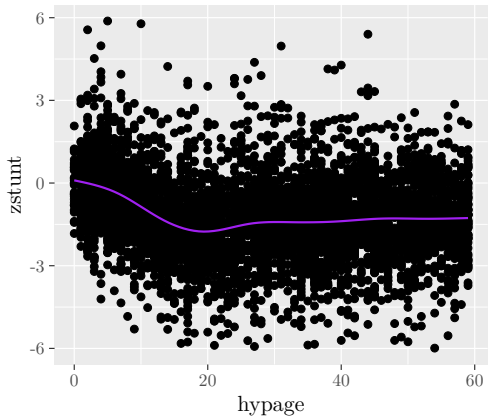
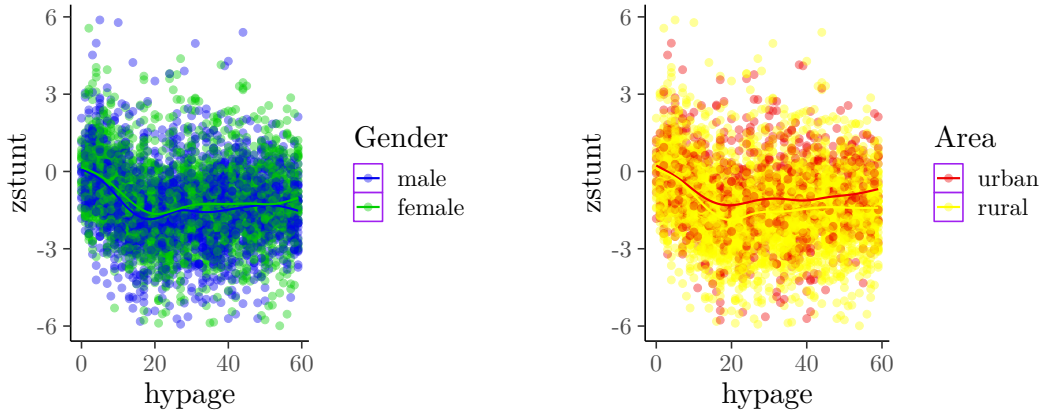


Figure 1.1: Scatter plot of `zstunt` against `hypage` with a smooth line (in purple)



(a) For each gender

(b) For each area

Figure 1.2: Smooth plots of `zstunt` against `hypage`

The smmooth line in the graphics of figure 1.2 indicates that `zstunt` is more detoriated in rural places, and for female genders.

(c) Figure 1.3 shows the map of Kenya, with colors indicating the amount of `zstunt` in counties. The color in Isolo' county is grey because we had no data available for that county. The county where children are stunted is West Pokot since in the county `zstunt` < -2.

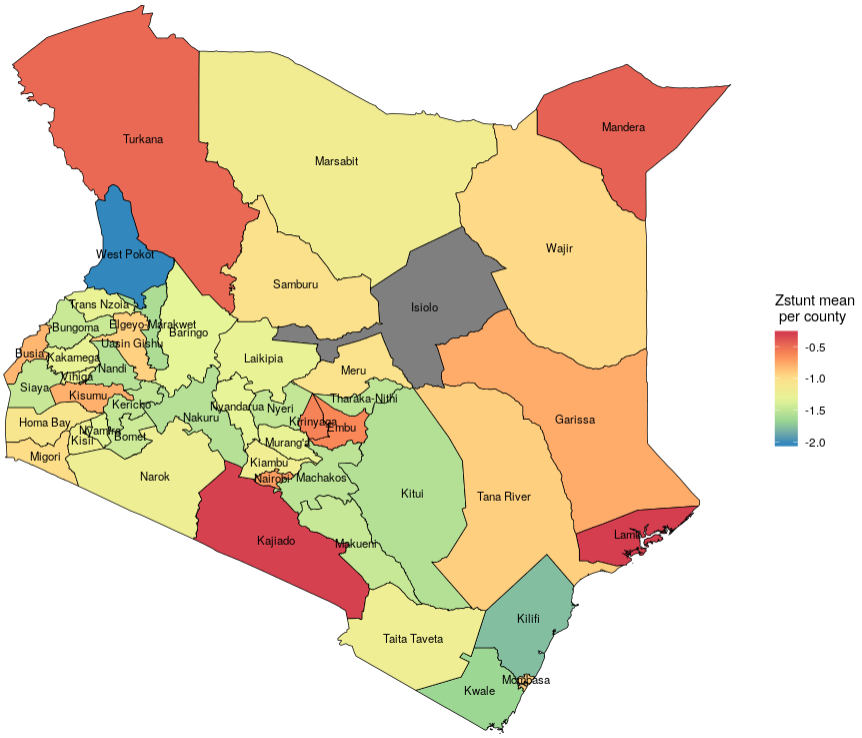


Figure 1.3: Plot the map of Kenya with the mean of `zstunt` in the each county

CHAPTER

TWO

RANDOM NUMBER GENERATION

2.1 Problem description

Generating a random variable from any distribution is very essential for stochastics studies. However, true random numbers are not always available, but we can use some algorithms that generate some pseudo-random numbers.

2.2 Methods

Pseudo-random numbers are a sequence of numbers that appear "random" or approximate properties of random numbers with the following properties:

- Good approximation of the properties of random numbers.
- Number can be easily and efficiently generated.
- Reproducibility (truly random numbers never satisfy this).

With the help of some probability properties, it is typically enough to be able to use a uniform distributed random variable, in order to generate any pseudo-random numbers from a given distribution. The simplest idea for generating uniformly distributed pseudo-random numbers is using a *linear congruent generators* (LCG):

1. Choose positive integer parameters a, c and m.
2. Choose an initial value $x_0 \in \{0, 1, \ldots, m-1\}$ (this value is called the seed).
3. For each $n \in \mathbb{N}$, compute

$$x_{n+1} := ax_n + c \mod m, \qquad \forall n \in \mathbb{N} \tag{2.2.1}$$

4. The psuedorandom numbers are the sequence $x_1; x_2; x_3; \ldots$.

We make some observations regarding the LCG algorithm.

- The LCG can generates at most m distinct numbers which are contained in $\{0, 1, \ldots, m-1\}$
- As soon as some number in the sequence, say x_{n0}, is repeated (i.e.,$\exists p$ such that $x_{n_0+p} = x_{n0}$), then the same is true for the entire sequence:

$$x_{n_0+p+j} = x_{n_0+j}, \qquad \forall j \geqslant 1 \tag{2.2.2}$$

  The number p is called the period of the sequence. This musst be less than or equal to m.
- Suppose that (under the right conditions) the LCG approximates a uniform distribution with parameters 0 and m. In this case, x_n/m would approximate a uniform distribution with parameters 0 and 1.

To generate uniformly random numbers, the period has to be maximal (i.e., $p = m$), so that we sample every value in the sequence before repeating any. One can show that LCG has a full period $m = 2^b$, $b \geqslant 2$ if and only if $c \in (0; m)$ is odd and $a \mod 4 = 1$.

After generating uniform pseudo-random numbers, we can easily obtain random variables from other distributions. The **inversion method** is one way of doing so, with the help of the following theorem:

Theorem 1 *Let F be a distribution function on $\mathbb{R}$. The quantile function F^{-1} is defined by*

$$F^{-1}(u) = inf\{x : F(x) \geq u, 0 < u < 1\}$$

If $U \sim U_{[0;1]}$, then $F^{-1}(u)$ has a distribution function F.

Hence, for continuous distributions (exponential, Pareto, standard Cauchy, etc) where F^{-1}, we simply simulate U_i (with LCG) and set $X_i = F^{-1}(U_i)$. If F^{-1} cannot be inverted analytically, appropriate numerical methods can be applied.

Let X be a discrete random variable with ordered possible values $\{x_1, x2, \ldots\}$, so that $F(x) = \sum_{i:x_i \leqslant x} P(X = x_i)$ and

$$F^{-1}(r) = \min\{x_k \in \{x_1, x_2, \ldots\} : \sum_{j=1}^{k} P(X = x_j) = \sum_{j=1}^{k} p_j \geqslant r\}$$

Then the inverse method becomes: set $X = x_1$ if and only if $U_i \in [0, p_1)$ and $X = x_k$ if and only if $U_i \in \Big[\sum_{j=1}^{k-1} p_j, \sum_{j=1}^{k} p_j\Big)$, $k = 2, 3, \ldots$. Note that

$$P(X = x_k) = P\Big(\sum_{j=1}^{k-1} p_j \leqslant U_i < \sum_{j=1}^{k} p_j\Big) = \sum_{j=1}^{k} p_j - \sum_{j=1}^{k-1} p_j = p_k$$

For example, to simulate a Bernoulli random variable $Ber(p)$, generate $U \in U_{[0,1]}$ and set $X = 0$, if $U \leqslant 1 - p$ and $X = 1$ if $U > 1 - p$.

Another general approach to pseudo-random variables generation is the **acceptance-rejection method**.

Data: Two probability densitiy functions: f for X and g for Y

1. Find a constant $M > 0$ such that $sup_x \frac{f(x)}{g(x)} \leqslant c$;
2. Obtain a sample y from Y ;
3. Obtain a sample u from the uniform distribution on $[0, 1]$;
4. **if** $u < \dfrac{f(y)}{cg(y)}$ **then**
5. | Accept y as a sample drawn from f;
6. **else**
7. | Reject the value of y and return to the sampling step (line 2);

Result: y, a sample drawn from f (using g)

NB: Computing c could be difficult, but one can show that $c = sup_x \frac{f(x)}{g(x)}$

2.3 Results

(a) With the Wichmann-Hill pseudo-random number generator in R, we Simulate $N = 1000$ binomial random variables $B(n = 10, p = 0.4)$ using three approaches: inversion

method, by simulating corresponding Bernoulli random variables by inversion method and using R built-in function rbinom. From the figure 2.1, the histograms of the three samples present the same shape but are different. This proves that the "random" numbers generated by our methods are just approximations of true random numbers.

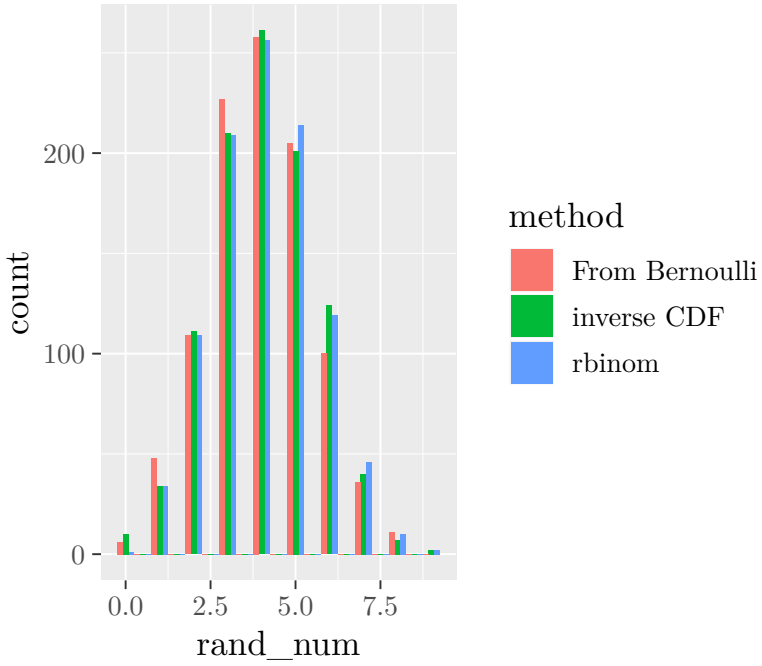


Figure 2.1: Histogram of the empirical CDF of all three samples

(b) To use accept-reject method (and a generator for uniform random variables only), of $N = 10000$ standard normal distributed random variables with density $f(x) = (2\pi)^{-\frac{1}{2}} e^{-\frac{x^2}{2}}$, the density of the standard Cauchy distribution is used: $g(x) = \{\pi(1 + x^2)^{-1}\}$.

- The constant value c for this method is given by $sup_x \frac{f(x)}{g(x)} = 1.520347$
- Then after computing the N standard normal random variables, we notice that the estimated and theoretical acceptance probabilities are almost equal. This is well depicted with in the figure 2.2,

where the histogram of the obtain sample is symmetric and has the same shape as the standard normal density curve.

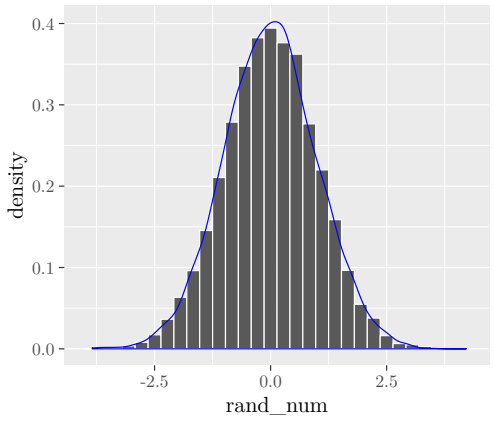


Figure 2.2: Histogram of the obtained sample and the standard normal density (in blue)

- The QQ-plot in figure 2.3 shows points following the identity line. Hence the accept-reject method used to simulate a standard normal distributed sample (using the the standard Cauchy density) is well accurate.

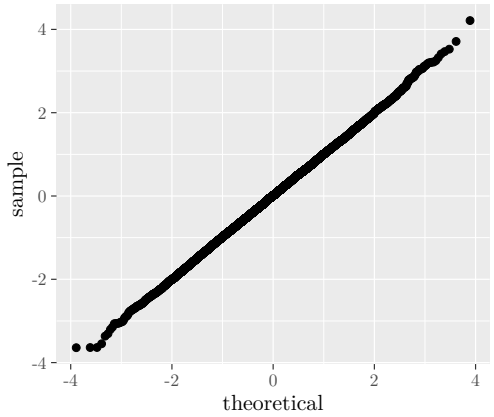


Figure 2.3: QQ-plot

- However, it is not possible to simulate ample distributed from the standard Cauchy density using the accept-reject method with a standard normal candidate density, simply because cannot find a c such that $g(x) \leqslant cf(x)$ is verified (because $sup_x \frac{g(x)}{f(x)} = \infty$).

CHAPTER

THREE

BOOTSTRAP

3.1 Problem description

Suppose that a sample $\mathbf{X} = \{X_1, \ldots, X_n\}$ is used to estimate a parameter θ of the distribution P (which is unknown) and let $\hat{\theta} = S(\mathbf{X})$ be a statistic that estimates θ. For the purpose of statistical inference on θ, we are interested in the sampling distribution of $\hat{\theta}$ (or certain aspects of it) so as to assess the accuracy of our estimator or to set confidence intervals for our estimate of θ. If the true distribution P were known, we could draw samples $\mathbf{X}_l, l = 1, \ldots, R \in \mathbb{N}$ from P and use Monte Carlo methods to estimate the sampling distribution of our estimate $\hat{\theta}$. The problem is that P is unknown and we cannot sample from it.
The following section explains how to use bootstrap to make the interference on $\hat{\theta}$.

3.2 Methods

The bootstrap is a computerintensive resampling method, which principle can be summarized by the following schematic diagram:

Real World			**Bootstrap World**	
Unknown probability distribution	Observed random sample		Empirical distribution	Bootstrap sample
$P \longrightarrow \mathbf{X} = \{X_1, \ldots, X_n\}$		$\Longrightarrow$	$\hat{P} \longrightarrow \mathbf{X}^* = \{X_1{}^*, \ldots, X_n{}^*\}$	
$\downarrow$			$\downarrow$	
$\hat{\theta} = S(\mathbf{X})$			$\hat{\theta}^* = S(\mathbf{X}^*)$	
Statistic of interest			Bootstrap replication	

Then The idea is to sample from an empirical distribution function. Recall that for random variables $Y = \{Y_1, \ldots, Y_n\}$, the empirical distribution function is defined via $F_n(y) = n^{-1} \sum_{i=1}^{n} \mathbb{I}(Y_i \leqslant y)$ (we will

use the notation F_B for the bootstrap empirical distribution). If the sample of size n is from a continuous distribution, then each observation has a probability $1/n$ and sampling from F_n would be equivalent to draw with replacement from the sample. Hence the following algorithm:

1. Draw n times with replacement from $\mathbf{X}$ to get a bootstrap sample $\mathbf{X}_1^*$ of size n. Repeat R times to get R bootstrap samples $\mathbf{X}_1^*, \ldots, \mathbf{X}_R^*$, each of size n.
2. Compute bootstrap statistics $S(\mathbf{X}_1^*), \ldots, S(\mathbf{X}_R^*)$.
3. Make inference about θ based on $S(\mathbf{X}_1^*), \ldots, S(\mathbf{X}_R^*)$.

We can also evaluate the goodness of the estimators (point or interval) based on the bootstrap sample. We construct confidence intervals for θ from the bootstrap replications (see step 2 in the above algorithm).

First recall the definition of a confidence interval. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a sample from a population with distribution $P \in \mathcal{P} = \{P_\theta : \theta \in \Theta \subset \mathbb{R}^d\}$. Let $C(\mathbf{X})$ depend only on the sample $\mathbf{X}$ and $\theta \in \Theta$ be an unknown parameter of interest. If

$$\inf_{P \in \mathcal{P}} P(\theta \in C(\Theta)) \geqslant 1 - \alpha$$

for a fixed $\alpha \in (0, 1)$, then $C(\Theta)$ is a **confidence set** for θ with **level of significance** $1 - \alpha$. If the parameter θ is real-valued, then $C(\Theta) = [\underline{\theta}(\mathbf{X}), \bar{\theta}(\mathbf{X})]$, for a pair of real-valued statistics $\underline{\theta}$ and $\bar{\theta}$ is called a confidence interval for θ.
Therefore, a natural way to construct the bootstrap confidence interval is to use empirical quantiles of the bootstrap distribution of $S(\mathbf{X})$: compute $\hat{\theta}_i^* = S(\mathbf{X}_i^*)$, $i = 1, \ldots, R$ bootstrap statistics and set the confidence interval for θ by $[\theta_L^*, \theta_U^*]$, where θ_L^* and θ_U^* are respectively $\lfloor R(\frac{1-\alpha}{2}) \rfloor$-th and $\lfloor R(1-\frac{1-\alpha}{2}) \rfloor$-th value in the ordered list of $\hat{\theta}_i^*$. Such confidence intervals are called **bootstrap percentile** confidence intervals. By defining $F_B(x) = P(\hat{\theta}^* \leqslant x)$, note that we have $P(\hat{\theta}^* \leqslant \hat{\theta}_L^*) \approx \frac{1}{2}\alpha$ and $P(\hat{\theta}^* \geqslant \hat{\theta}_U^*) \approx \frac{1}{2}\alpha$, which makes a coverage probability of $1 - \alpha$.

The confidence interval should have equal probability to both sides of $\hat{\theta}^*$, that is $P(\hat{\theta}^* \leqslant \theta \leqslant \hat{\theta}_U^*) = P(\hat{\theta}_L^* \leqslant \theta \leqslant \hat{\theta}^*)$. If $\hat{\theta}^*$ is not the median of the bootstrap distribution, this condition is not fulfilled. An appropriate correction is given by $\hat{\theta}_{LC}^* = F_B^{-1}(\Phi[z_{\frac{\alpha}{2}} + 2\hat{z}_0])$ and $\hat{\theta}_{UC}^* = F_B^{-1}(\Phi[z_{1-\frac{\alpha}{2}} + \hat{z}_0])$ (respectively the bias-corrected lower and upper confidence bound for θ), where $\Phi(.)$ is the cdf of the standard normal distribution and $\hat{z}_0 = \Phi^{-1}\{F_B(\hat{\theta})\}$. This interval is the **bias corrected percentile interval**. In practice, $\hat{\theta}_{LC}^* = \lfloor R\alpha_1 \rfloor$ and $\hat{\theta}_{UC}^* = \lfloor R\alpha_2 \rfloor$, with $\alpha_1 = \Phi(z_{\frac{\alpha}{2}} + 2\hat{z}_0)$ and $\alpha_2 = \Phi(z_{1-\frac{\alpha}{2}} + 2\hat{z}_0)$.

An extension of the bias corrected percentile confidence interval, the BC_a (**bias-corrected accelerated** bootstrap) confidence interval described as follow:

$$\alpha_1 = \Phi\Big(\hat{z}_0 + \frac{\hat{z}_0 + z_{\alpha/2}}{1 - \hat{a}(\hat{z}_0 + z_{\alpha/2})}\Big)$$

$$\alpha_2 = \Phi\Big(\hat{z}_0 + \frac{\hat{z}_0 + z_{1-\alpha/2}}{1 - \hat{a}(\hat{z}_0 + z_{1-\alpha/2})}\Big),$$

where

$$\hat{a} = \frac{\sum_{i=1}^n (\bar{\theta}_J - \hat{\theta}_i)^3}{6\Big\{\sum_{i=1}^n (\bar{\theta}_J - \hat{\theta}_i)^2\Big\}^{3/2}}$$

With $\bar{\theta}_J = n^{-1}\sum_{i=1}^n \hat{\theta}_{(i)}$, for $\hat{\theta}_{(i)}$ as the estimator of θ obtained without observation i, i.e., $\hat{\theta}_{(i)} = S(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$. It is easily performed in **R** with **bootstrap::bcanon**.

3.3 Results

(a) Let's consider a Weibull distribution with scale parameter λ, shape parameter k, variance σ^2 and median x_{med}. From a sample $(x_1, \ldots, x_n)$ simulated from the Weibull distribution with $\lambda = 13$ and $k = 1$, we aim to to build confidence intervals for σ based on a statistics $\hat{s}^2 = (n-1)^{-1} \sum_{i=1}^{n} (x_i - \bar{X})^2$, and x_{med} based on the sample median.

- First, the sample size is set as $n = 100$, the number of bootstrap replications $R = 1000$ and the number of Monte Carlo samples $M = 1000$. We build two-sided bootstrap percentile confidence intervals for σ and x_{med} at the significance level $\alpha = 0.05$, and Use M Monte Carlo samples to estimate the coverage probability(CP) and the average interval length(AIL) for both confidence intervals(CI). We get the following results:

	x_{med} CI	σ CI
CP	0.944	0.856
AIL	5.103662	6.006730

Table 3.1: Confidence intervals coverage probability and average interval length: $n = 100$, $R = 1000$

The coverage probability for x_{med} confidence interval is pretty close to $1 - \alpha = 0.95$, the same for σ confidence interval, but less than the CP x_{med} CI. This might suggest that the bootstrap percentile confidence interval approximates x_{med} CI more than σ CI.

- Now, use the following settings: $n = R = 1000$ to get the results in table 3.2 and $n = 100$, $R = 5000$ and obtain the corresponding results in table 3.3

	x_{med} CI	σ CI
CP	0.947	0.935
AIL	1.620852	2.211146

Table 3.2: Confidence intervals coverage probability and average interval length: $n = R = 1000$

	x_{med} CI	σ CI
CP	0.946	0.848
AIL	5.101970	5.981314

Table 3.3: Confidence intervals coverage probability and average interval length: $n = 100$, $R = 5000$

We can notice that the CP value for both confidence intervals are again close to 0.95, but huge differences with the AIL. Actually, we want the length of the confidence intervals to be narrow as possible, and the AIL in table 3.2 are the smallest AIL, and the CP are the largest.

Hence, increasing the sample size and the number of bootstraps replications improves the accuracy of the bootstrap.

- With $n = 100$, $R = 1000$ and $M = 1000$, we build bootstrap accelerated bias-corrected (bc_a) confidence intervals both for σ and x_{med}, and Use M Monte Carlo samples to assess the coverage probability and the average length of the confidence intervals to obtain the following table.

	x_{med} CI	σ CI
CP	0.956	0.912
AIL	5.030710	6.661682

Table 3.4: Confidence intervals coverage probability and average interval length: bc_a

The CP values for the bc_a confidence intervals are more closer to 0.95 (especially for σCI)than the ones of the bootstrap percentile confidence intervals(see table 3.1). We also notice a slight difference in the AIL for both confidence intervals in both methods.

	x_{med}	σ
$\hat{z}_0$	-0.04063	0.1113
$\hat{a}$	-1.475×10^{-15}	0.09085

Table 3.5: Average $\hat{z}_0$ and $\hat{a}$

The results in table 3.5 show that $\hat{z}_0$ is negative for x_{med} confidence intervals and positive for σ confidence intervals, which are respectively signs of overestimation and underestimation. The estimated accelerated term $\hat{a}$ is slightly to zero for both confidence intervals, indicating a slight asymmetry in the data. Since all these values are close to zero, we can that the confidence bands have a good performance.

(b) From the dataset *shhs1.txt* has been obtained from Sleep Heart Health Study, we are using the variable **rdi4p**: respiratory disturbance index. Figure 3.1, we notice that the **rdi4p** is skewed on the right(positiv skewness).

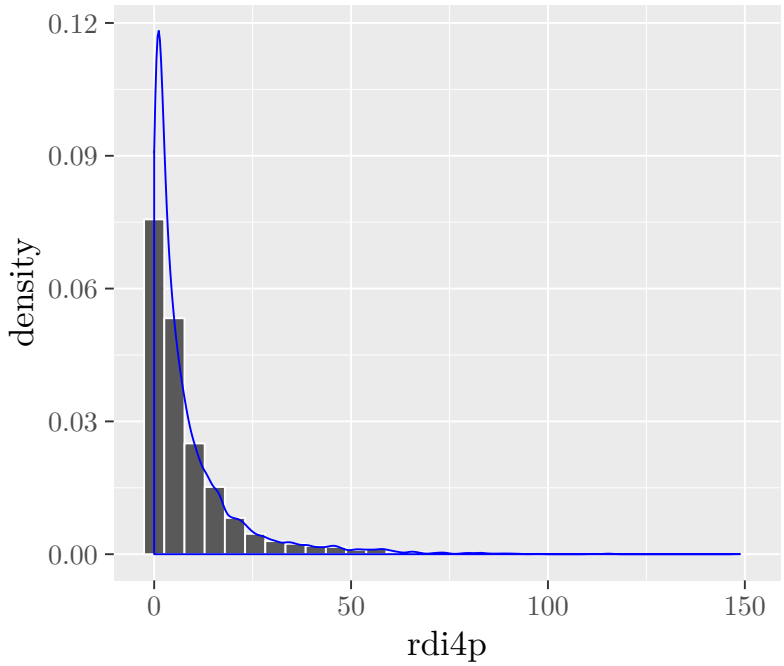


Figure 3.1: Histogram of **rdi4p** with the empirical distribution

By building bootstrap percentile and bootstrap accelerated bias-corrected confidence intervals for the standard deviation and median, we get the following results (with $R = 1000$)

	x_{med}	σ
CI	[3.951, 4.419]	[11.785, 13.045]
CI_{length}	0.498	1.26

Table 3.6: Results for bootstrap percentile confidence interval

	x_{med}	σ
CI	[3.944, 4.429]	[11.792, 13.103]
CI_{length}	0.485	1.311
$\hat{z}_0$	0.00251	−0.00251
$\hat{a}$	0	0.0272

Table 3.7: Results for bc_a

The results from tables 3.6 and 3.7 show that the x_{med} CI are almost the same, while we notice a slight difference σ CI. Actually notice shift to the right in the σ CI with bc_a, but with slight shift to the right in the CI for bc_a. This can be explained by the fact that the empirical distribution is right-skewed, and that $\hat{a}$ is null for x_{med} CI, and close to zero for σ CI. We can notice that $\hat{z}_0$ is close to zero, indicating good minimization of overestimation (for σ CI) and underestimation (for x_{med} CI).

CHAPTER

FOUR

GENERALISED LINEAR MODELS

4.1 Problem description

In its simplest form, a linear model specifies the (linear) relationship between a dependent variable Y (normally distributed), and a set of independent variables X_i, $i = 1, \ldots, k \in \mathbb{N}$, so that $Y = b_0 + b_1 X_1 + \ldots + b_k X_k$, where b_0 is the regression coefficient for the intercept and the b_i values are the regression coefficients (for variables 1 through k). However, there are many relationships that cannot adequately be summarized by a simple linear equation, for two major reasons:

- *Distribution of the dependent variable.* The dependent variable of interest may have a non-continuous distribution, and thus, the predicted values should also follow the respective distribution; any other predicted values are not logically possible.
- *Link function.* The second reason why the linear model might be inadequate to describe a particular relationship is that the effect of the predictors on the dependent variable may not be linear in nature.

Generalized linear models (GLMs) extend linear models to accommodate both non-normal response distributions and transformations to linearity.

4.2 Methods

Let $(Y_1, X_1), \ldots, (Y_n, X_n)$ be independent pairs of observations, where Y_i is real-valued and X_i are $\mathbb{R}^k$-valued random variables. Generalised linear models (GLMs) have the following three-part specification:

- **The random component** (=response from an overdispersed exponential family). The data $Y_1, \ldots, Y_n$ are such that $Y_1|X_1, \ldots, Y_n|X_n$ are independent and $Y_i|X_i$ has the p.d.f.

$$f_{\eta,\psi}(y_i|x_i) = \exp\left\{ \frac{\eta_i y_i - \kappa(\eta_i)}{\psi_i} \right\} h(y_i, \psi_i), \qquad i = 1, \ldots, n,$$

where η_i is called canonical parameter and i is an unknown scale or dispersion parameter. Functions κ and h are known and $\kappa^{''}(\eta) > 0$ is assumed. Note that

$$\mu(\eta_i) := \mathtt{E}(Y_i|X_i) = \kappa_0(\eta_i) \quad \mathtt{and} \quad var(Y_i|X_i) = \psi_i \kappa^{''}(\eta), \qquad i = 1, \ldots, n$$

- **The systematic component** (=linear predictor) Canonical parameter η_i is assumed to be related to X_i. The term $X_i^t\beta$ for unknown $\beta \in \mathbb{R}^d$ is called the **linear predictor or systematic component**.
- The **link function** between random and systematic components. The relationship between η_i and $X_i^t\beta$ is described through

$$g\{\mu(\eta_i)\} = X_i^t\beta, \qquad i = 1, \ldots, n$$

where g is called a link function. The link function g is assumed to be a known, one-to-one, third-order continuously differentiable function. If $g = \mu^{-1}$ then $\eta_i = X_i^t\beta$, and g is called the **canonical or natural link function**. If g is not canonical, then

it is assumed that $d(g \circ \mu)(\eta)/d\eta \neq 0$ for all η.

In a GLM, the parameter of interest is β. Parameters ψ_i are considered to be nuisance parameters. It is often assumed that $\psi_i = \psi/t_i$, $i = 1, \ldots, n$ with an unknown ψ and known t_i's or, alternatively $\psi_i = a()$ for some known function a. Note that ψ_i enter $var(Y_i|X_i) = \psi_i\kappa''(\eta_i)$, making it more flexible, that is allowing for over- or underdispersion.

Exemple: Let Let $Y_i|X_i \backsim Poi(\lambda_i)$. We can write the density

$$f_\eta(y_i) = \mathtt{exp}\{y_i log(\lambda_i) - \lambda_i\}\frac{1}{y_i!}\mathbb{I}_{\{1,2,\ldots\}}(y_i)$$

that is, the canonical parameter $\eta_i = \mathtt{log}(\lambda_i), \kappa(\eta_i) = \lambda_i = \mathtt{exp}(\eta_i)$, $\psi_i = 1$ and $h(y_i) = (y_i)^{-1}\mathbb{I}_{\{1,2,\ldots\}}(y_i)g(y_i)$. Since $E(Y_i|X_i) = \kappa_0(\eta_i) = \mathtt{exp}(\eta_i) =: \mu(\eta_i)$, the canonical link is
$g(x) = \mu^{-1}(x) = \mathtt{log}(x)$, which is called the **log-link** ($g(\mu(\eta_i)) = \eta_i$). Hence,

$$\mathtt{log}\{\mathtt{E}(Y_i|X_i = x_i)\} = x_i^t\beta,$$

where $x_i \in \mathbb{R}^k$, $\quad i = 1, \ldots, n$.

Estimation

Let $\theta = (\beta, \psi)$ and $(g \circ \mu)^{-1} = \zeta$ (for a canonical link $\zeta(x) \equiv x$). Then

$$\ell(\theta) = \sum_{i=1}^{n}\left[\frac{\zeta(X_i^t\beta)Y_i - \kappa\{\zeta(X_i^t\beta)\}}{a(\psi)} + \mathtt{log}h(Y_i, \psi)\right].$$

Further, consider the canonical link. Taking derivatives w.r.t. β and we get the following score equations

$$\frac{\partial \ell(\theta)}{\partial \beta} = \frac{1}{a(\psi)} \sum_{i=1}^{n} \{Y_i - \mu(X_i^t)\} X_i = 0$$

$$\frac{\partial \ell(\theta)}{\partial \psi} = \sum_{i=1}^{n} \left[\frac{\partial \texttt{log} h(y_i, \psi)}{\partial \psi} + \{a^{-1}(\psi)\}' \{X_i^t \beta Y_i + \kappa(X_i^t \beta)\} \right] = 0$$

Where $\kappa(X_i^t \beta) = \mu(X_i^t \beta)$ was used. If MLE of β exists, then it can be found from the first equation without estimating. Estimation of ψ from the second equation in many cases is a difficult task and depends on a particular distribution. To estimate β and study its properties we also need

$$-\frac{\partial^2 \ell(\theta)}{\partial \beta \partial \beta^t} = \frac{1}{a(\psi)} \sum_{i=1}^{n} \left[\kappa(X_i^t \beta)'' X_i X_i^t \right] =: -\frac{F_n(\beta)}{a(\psi)}$$

With this, we can set up the Newton-Raphson algorithm as

$$\hat{\beta}^{(j+1)} = \hat{\beta}^{(j)} + \left\{ F_n(\hat{\beta}^{(j)}) \right\}^{-1} S_n(\hat{\beta}^{(j)}), \quad j = 0, 1, 2, \ldots,$$

where $S_n(\hat{\beta}^{(j)}) = a(\psi) \partial \ell(\theta) / \partial \beta$.

Goodness-of-fit and models' comparison

Now, we want to to assess how good the model fits the data, i.e., to measure the discrepancy between the data $Y_i | X_i$ and estimated $\texttt{E}(Y_i | X_i) = \mu_i$. First, some definitions. The **null model** is simplest model, and has only one parameter, representing a common mean μ, say, for all $Y_i | X_i$. At the other extreme is the **full model**, which has n parameters, one for each observation. The full model gives a baseline for measuring the discrepancy for an intermediate model with k parameters. Assume for the moment that ψ is known and denote $\ell(\hat{\mu}, \psi)$ the log-likelihood with $\hat{\mu} = g^{?1}(X\hat{\beta})$. The maximum likelihood in the full model is then $\ell(Y, \psi)$ ($= \mu_i$ are replaced by Y_i). Then the **deviance of the fitted model** is defined as

$$D(Y, \hat{\mu}) = a(\psi) 2\{\ell(Y, \psi) - \ell(\hat{\mu}, \psi)\}$$

Note that $D(Y, \hat{\mu})/a(\psi)$ is called the **scaled deviance**(or the deviance for $2\{\ell(Y, \psi) - \ell(\hat{\mu}, \psi)\}$). The **generalised Pearson statistic** is defined via

$$\chi^2 = \frac{\sum_{i=1}^{n} (Y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)}$$

. The following methods are used to measure the goodness-of-fit, and compare models:

- **Analysis of deviance:** Scaled deviance can be used to compare two nested models, i.e. the parameter space under one model is a subspace of that under the second model. let M_k and

M_q, with $q < k$ (k and q are the number of parameters in M_k and M_q respectively) two nested models. Let us denote D_{M_k} and D_{M_q} respectively as the scaled deviance of M_k and M_q. Since we have assume that ψ is known, we have the following formula:

$$\frac{D_{M_q} - D_{M_k}}{\psi} \overset{approx}{\sim} \chi^2_{k-q}$$

A widely used rule of thumb(to measure goodness-of-fit) is that a good fit has the scaled deviance about $n?k$, which is the expectation of a χ^2_{n-k} distributed random variable. Large values of the scaled deviance are considered to indicate a bad fit. However, this has to be treated with care. For Poisson data with large λ_i and Binomial data with large m_i, the approximation to χ^2_{n-k} works reasonable, but not in many other cases. Therefore we can use other methods.

- **Residual analysis:** Here, the residuals used are expected to behave approximately as zero-mean normally distributed variables. **Pearson residuals** defined via

$$r_i^p = \frac{Y_i - \hat{\mu}_i}{d\sqrt{V(\hat{\mu}_i)}}, \quad i = 1, \ldots, n$$

. Pearson residuals have the disadvantage of being skewed for non-normal responses. As a remedy, we have the **Anscombe residulas**, which in the special case of the Poisson distribution is given by

$$r_i^a = \frac{3(Y_i^{2/3} - \hat{\mu}_i^{2/3})}{2\hat{\mu}_i^{1/6}}, \quad i = 1, \ldots, n$$

.

- **Deviance residuals:** based on the deviance, they are defined by

$$r_i^d = \texttt{sign}(Y_i - \hat{\mu}_i)\sqrt{2\{\ell_i(Y_i, \psi) - \ell_i(\hat{\mu}_i, \psi)\}}, \quad i = 1, \ldots, n$$

where ℓ_i is the log-likelihood corresponding to the i-th observation, so that $\sum_{i=1}^n (r_i^d)^2 = D(Y, \hat{\mu}))$. A standardised version of the deviance (as well as Pearson) residuals are used:

$$\frac{r_i^d}{\sqrt{a(\hat{\psi})(1 - h_i)}}, \quad , i = 1, \ldots, n$$

where $h_i = H_{i,i}$ with the hat matrix H taking now the form $H = W^{1/2}X(X^tWX)^{-1}X^tW^{1/2}$, where W is the weight matrix from the Fisher scoring. In an adequate model the plot of standardised residuals against $\hat{\eta} = X\hat{\beta}$ should show *no patterns*. The **null pattern** is a distribution of residuals with mean zero and constant variance.

- **Akaikeinformation criterion (AIC) and Bayes information criterion (BIC):** These criterions can be used to compare models with different subset of parameters or even to compare

two different models (e.g., with different link functions or a non-linear and with a linear model). These two criteria are most popular examples of penalised goodness-of-fit criteria

$$AIC(M) = -2\ell(M) + 2|M|$$

$$BIC(M) = -2\ell(M) + \texttt{log}(n)|M|,$$

where $\ell(M)$ denotes the log-likelihood corresponding to a model M and $|M|$ is the number of parameters in that model M. The models, selected with these criteria are then

$$\hat{M}_{AIC} = \texttt{arg}\min_{M \in \mathcal{M}} AIC(M)$$

$$\hat{M}_{BIC} = \texttt{arg}\min_{M \in \mathcal{M}} BIC(M)$$

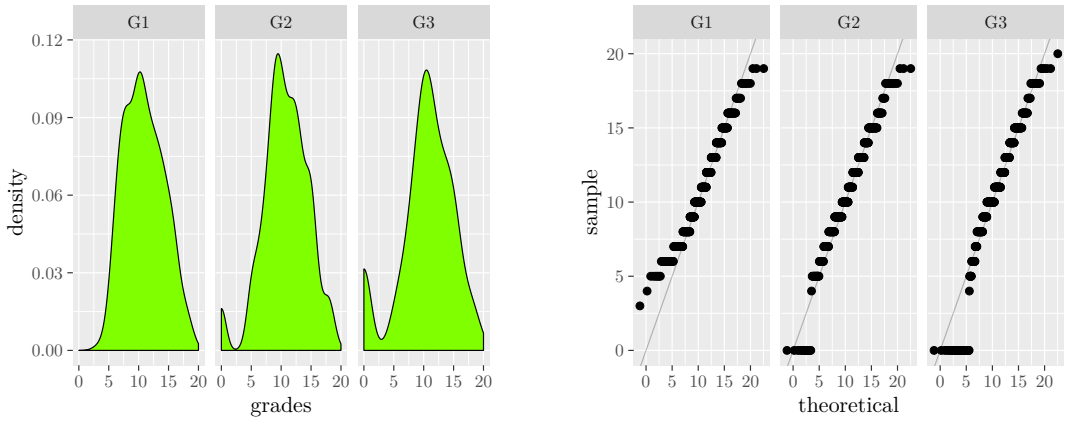
4.3 Results

For the exercise, the dataset *student-mat.csv* can be found on Kaggle. Variables G1, G2, G3 are first, second and final grades in mathematics. The remaining variables are explanatory variables. We would like to identify variables that explain grades in mathematics.

(a) First of all, we need to identify the distribution of G1, G2, and G3. From the Q-Q plots with normal theoretical distribution of figure 4.1b, we notice too many zero points and points away on the tails. Moreover, the emperical densities plots are skewed on the left, that is way different from the bell-shape of a normal distribution. Therefore G1, G2, and G3 are not normally distributed. On the other hand, the figure 4.2 of the Q-Q plot with Poisson as theoretical distribution, displays points along the identity line suggesting that we might have a Poisson distribution. However, there are some zero points (especially in G1 and G2), which is a sign of under-dispersion, and over-dispersion for G1. Actually since the variables are Poisson distributed, hence their means should be equal their variances. But different results (see table 4.1) confirm the latter assumption.

	mean	var
G1	10.909	11.017
G2	10.714	14.149
G3	10.415	20.989

Table 4.1: Variances and means of G1, G2, and G3



(a) Emperical densities (b) Q-Q plot with normal theoretical distribution

Figure 4.1: Checking normality assumption

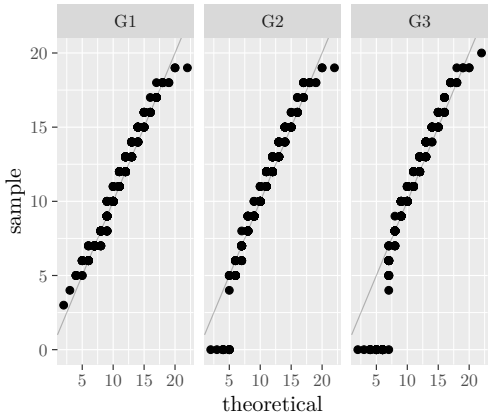
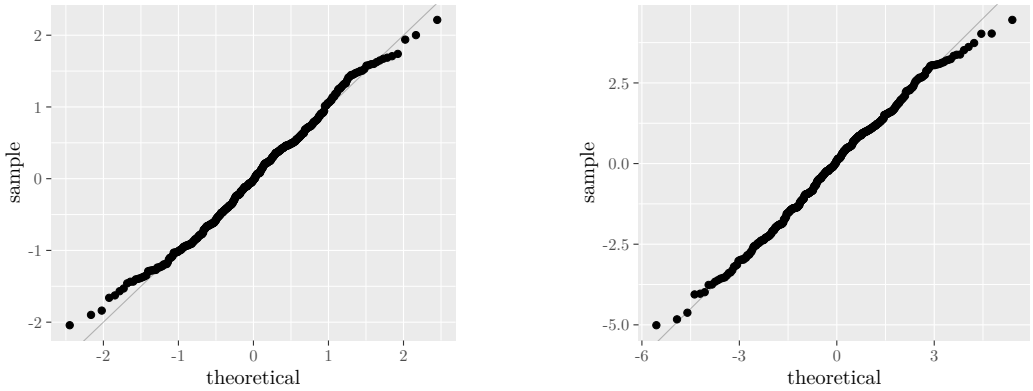


Figure 4.2: Q-Q plot with Poisson as theoretical distribution

(b) A generalised linear model(Model 1) is fitted to explain G1 with all explanatory variables. With a $\alpha = 0.05$, we notice that all covariates are not significant, which might be a sign of a bad fitted model. After calculating the Pearson residuals and Anscombe residuals, we come upn with the plots in figure 4.2. We can clearly assume that these residuals are normaly distributed, because points on the Q-Q plots are distributed along the identity line. Furthermore, figure 4.3a we can see that the normality assumption is again fulfill and points on other plots present no pattern, therefore we can say that the fitted (generalised) linear model is adequate for the data, but since we have too many unsignificant covariates, we can conclude that this model is not adequate to the data.



(a) Pearson residuals (b) Anscombe residuals

Figure 4.3: Q-Q plots of residuals with normal theoretical distribution: Model 1

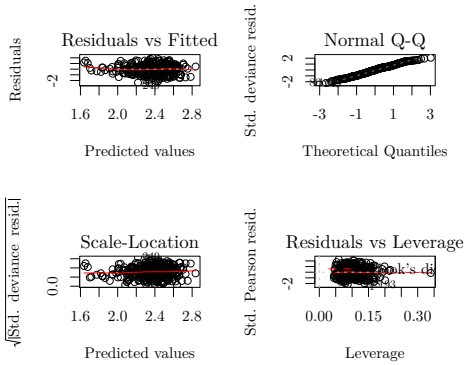


Figure 4.4: Residuals analysis: Model 1

(c) Now a another model, Model.2, is made by reducing variables from Model.1 to **sex**, **Fedu**, **studytime**, **failures**, **schoolsup**, **famsup**, **goout**. This time, all covariates are significant. **sex**, **Fedu**, **studytime** have a positive effect on grades, meaning that a female student whose father is well educated and who spend time studying has good grades. Whereas **failures**, **schoolsup**, **famsup**, **goout** have a negative effect on grades, meaning that a student who has already failed, with no extra educational support, no family educational support, and always going out has bad grades. Figure 4.5 show points along the identity of the Normal Q-Q plot, and no patterns in the residuals plots. Hence Model.2 is good model.

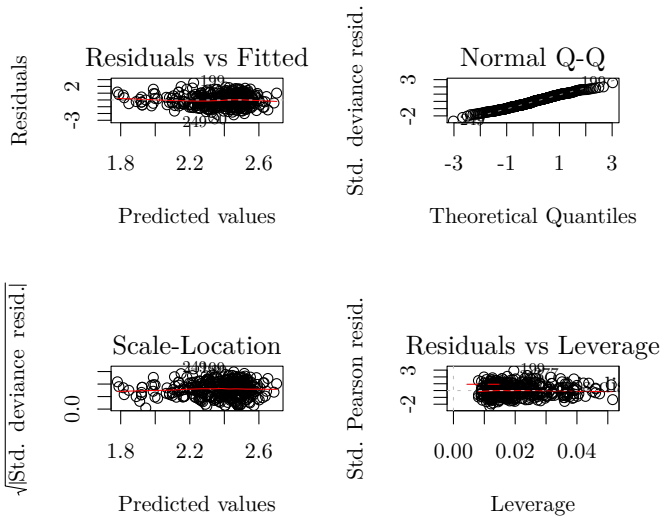


Figure 4.5: Residuals analysis: Model 2

By comparing Model.1 and Model.2 with ANOVA, we have a p-value = 0.1858 $> \alpha$, then we reject Model.1. Model.2 delivers better fit than Model.1. A third model, Model.3 is created by replacing in Model.2 **goout** by **Walc**. Then we want to compare these two models, but since they are not nested, we cannot use ANOVA as we did previously. So we opt for residuals analysis and AIC comparison. From figures 4.5 and 4.6, we can notice that the two models are pretty close, which makes difficult to make a choice. Nevertheless, the AIC of Model.2 is less than Model.3 AIC (respectively 1975.1 and 1977.2). Hence Model.2 delivers a better fit.

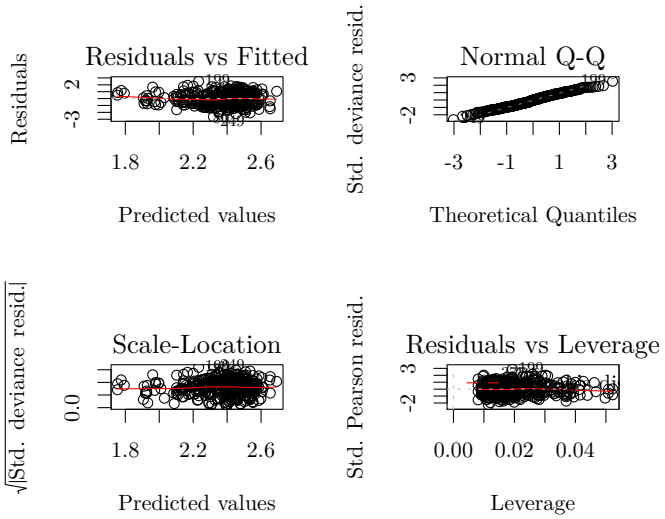


Figure 4.6: Residuals analysis: Model 3

CHAPTER

FIVE

SURVIVAL ANALYSIS

5.1 Problem description

We want to analyze data where the outcome variable is the time until the occurrence of an event of interest. The event can be death, occurrence of a disease, marriage, divorce, etc. The time to event or survival time can be measured in days, weeks, years, etc. For example, if the event of interest is heart attack, then the survival time can be the time in years until a person develops a heart attack. subjects are usually followed over a specified time period and the focus is on the time at which the event of interest occurs. Why not use linear regression to model the survival time as a function of a set of predictor variables? First, survival times are typically positive numbers; ordinary linear regression may not be the best choice unless these times are first transformed in a way that removes this restriction. Second, and more importantly, ordinary linear regression cannot effectively handle the censoring of observations. Why not compare proportion of events in your groups using risk/odds ratios or logistic regression? Simply because it ignores time.
To tackle these issues, we'll use some survival analysis.

5.2 Methods

Let T be a non-negative random variable that represents the time to event. We assume its CDF as F that has pdf f. Central concepts of the survival analysis are the *survivor function* (the probability that a subject will survive past time t) $S(t) = P(T > t) = 1 - F(t)$, the *hazard function* $h(t) = \frac{f(t)}{1-F(t)}$ (loosely speaking, it is the probability density of failure at time t, given survival to then), and the *cumulative hazard function* (accumulated risk up to time t) $H(t) = \int_0^t h(s)ds = -log\{S(t)\}$. Thus we have $S(t) = \texttt{exp}\{-H(t)\}$ and $f(t) = h(t)\texttt{exp}\{-H(t)\}$.

Exemples: Some common parametric distributions

1. Exponential distribution: $h(t) = \lambda$ and $S(t) = \texttt{exp}(-\lambda t)$

2. Weibull distribution: $h(t) = \alpha\lambda^{\alpha}t^{\alpha-1}$ and $S(t) = \texttt{exp}\{(-\lambda t)^{\alpha}\}$

Ideally, we would have independent realisations of T: $t_1, \ldots, t_n$. However, in practice the failure time cannot always be observed due to various reasons. This phenomenon is called *censoring*. We have *Type I censoring*, where T is observed until some pre-determined time c. If $T < c$, we observe the value t_i of T , if $T > c$, we only know that T survived beyond c. *Type II censoring* (rarely used) arises when n independent variables are observed until there have been r failures, so only $0 < T_{(1)} < \cdots < T(r)$ are observed. These are all examples of *right-censoring*. *Left-censoring* (the time of origin is not known) is less common.

Under censoring one rather deals with $Y_i = \min\{T_i, C_i\}$ (the observed response), where C_i denotes the censoring time for the ith subject. That is, a pair $(y_i; \delta_j)$ is observed, where Let δ denotes the event indicator.

$$\delta_i = \begin{cases} 0 & \text{if the event was observed } T_i \leq C_i \\ 1 & \text{if the response was censored } T_i > C_i \end{cases}$$

Note that T and C are independent.

Let's assume that T has a continuous distribution F and there are n data points available $(y_1, \delta_1), \ldots, (y_n, \delta_n)$, where $y_i = \min\{t_i, \delta_i\}$. Assume that $F(x) = F(x, \theta)$ is a some parametric distribution and that censoring variables C_i have CDF G and pdf g, which are independent on ?. The log-likelihood contribution from y_i can be represent as

$$\ell(\theta) = \sum_{i=1}^{n} [\delta_i \texttt{log}\{h(y_i; \theta)\} - H(y_i; \theta)]$$

For exponential distribution, have

$$\ell(\theta) = \sum_{i=1}^{n} (\delta_i log(\lambda) - \lambda y_i) = \log(\lambda) \sum_{i=1}^{n} \delta_i - \lambda \sum_{i=1}^{n} y_i$$

implying

$$\hat{\lambda}_{ML} = \frac{\sum_{i=1}^{n} \delta_i}{\sum_{i=1}^{n} y_i}.$$

An approximate confidence interval for λ (using asymptotic normality of maximum likelihood estimators) as

$$[\hat{\lambda}(1 - z_{\alpha/2}/\sqrt{r}), \hat{\lambda}(1 + z_{\alpha/2}/\sqrt{r}], \quad r = \sum_{i=1}^{n} \delta_i.$$

A commonly used parametric distribution for modelling lifetimes with monotone hazard is the Weibull distribution. Values for λ and α can be estimated by the maximum likelihood similarly to the exponential distribution, however, this has to be done numerically.

Nonparametric estimators

Often it is unclear which parametric model would be appropriate for the data (if any). A standard tool for initial data inspection, for suggesting plausible models and for checking their fit is a nonparametric estimator of the survivor function. For no censored observations, we could estimate $\hat{S}(t) = n^{-1} \sum_{i=1}^{n} \mathbb{I}(T_i > t)$. For censored observation, let $0 \leq \tau_1 < \tau_n < \dots$ be the ordered uncensored failure times. Let r_i denote the number of units that are still in risk at τ_i (=not failed yet or censored) and d_i the number of units that fail at τ_i. The **Kaplan-Meier estimator** for the survivor function S is given by

$$\hat{S}_{KM}(t) = \prod_{\{j:\tau_j<t\}} \left(1 - \frac{d_j}{r_j}\right)$$

A further estimator for S is the **Fleming-Harrington estimator** $\hat{S}_{FH}(t)$. It is a plug in estimator defined by

$$\hat{S}(t) = \texttt{exp}\{-\hat{H}(t)\},$$

where $\hat{H}(t) = \sum_{\{j:\tau_j<t\}} \frac{d_j}{r_j}$, is the *Nelson-Aalen* estimator for H.

Confidence bands

Assume that $\hat{S}$ is an estimator for S (e.g. the Kaplan-Meier or the Fleming-Harrington estimator) and let $\texttt{vâr}(\texttt{log}(\hat{S}))$ be some estimate for the variance of log(S). An approximate confidence band (contained in $[0, 1]$) is given by

$$[\texttt{exp}(-\texttt{exp}(B^-), \texttt{exp}(-\texttt{exp}(B^+)],$$

where $B^{\pm} = \texttt{log}(-\texttt{log}\hat{S}(t)) \pm z_{\alpha/2}\texttt{log}^{-1}\hat{S}(t)\sqrt{\texttt{vâr}(\texttt{log}(\hat{S}))}$

Log-rank test

We wish to decide whether or not two (or more) samples stem from the same survivor function or not. Assume that the failure times $\tau_1 < \dots < \tau_k$ are realizations of two random variables T_1 and T_2 corresponding to two groups of items (patients). For each observed failure time τ_j we consider the contingency table

Groups	failure at time τ_j	items at risk at time τ_j
1	d_{1j}	r_{1j}
2	d_{2j}	r_{2j}
$1+2$	d_j	r_j

Under the null-hypothesis that $T_1 = T_2$ the expected number of failures at time τ_j in group 1 and 2 are hypergeometrically distributed with parameters r_j, r_{1j}, d_j and r_j, r_{2j}, d_j respectively. Thus, mean and variance of the number of failures in group 1 and 2 can be computed as

$$e_{1j} = \frac{d_j}{r_j} r_{1j} \quad \texttt{and} \quad e_{2j} = \frac{d_j}{r_j} r_{2j}$$

and

$$v_{1j} = v_{2j} = \frac{d_j r_{1j} r_{2j} (r_j - d_j)}{r_j^2 (r_j - 1)}$$

Under the null-hypothesis, the statistic

$$\chi^2 = \frac{\left[\sum_{j=1}^{k} (d_{1j} - e_{1j})\right]^2}{\sum_{j=1}^{k} v_{1j}}$$

is χ^2-distributed with 1 degree of freedom.

Graphical tool to check if the Weibull model is adequate
Under the assumption that T is Weibull distributed, one has

$$\texttt{log}\{-\texttt{log}(S(t))\} = \alpha \texttt{log}(t) + \texttt{log}(\lambda), \quad , t > 0.$$

Now let $\hat{S}(t)$ be a nonparametric estimate for S (e.g. the Kaplan-Meier estimator $\hat{S}_{KM}$).Then the plot $\texttt{log}\{-\texttt{log}(\hat{S}(t))\}$ against $\texttt{log}(t)$ should approximately be a straight line with slope α and intercept $-\texttt{log}(\lambda)$.

5.3 Results

The dataset of interest for the exercise is *Thoracic.txt*, and can be found on UCI Machine Learning Repository. We will be using the following three variables:

PRE30: if a person is a smoker

AGE: patient age at surgery

Risk1Y: is TRUE if a person has died within a year after the surgery

We will consider failure times to be at AGE +1.

(a) After computing nonparametric estimators of the survivor function $S(t)$, we generate the figure 5.1, with 95% confidence bands.We notice that the two estimators are almost the same, but for a person above 80 years old, Fleming-Harrington estimate a survival probability greater than the one of Kaplan-Meier.

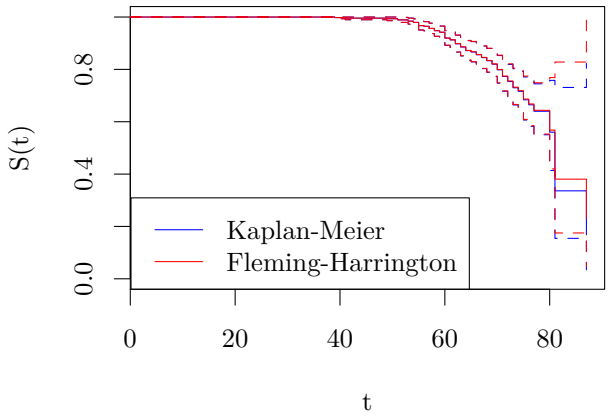


Figure 5.1: Plot of nonparametric estimators of the survivor function

Then we fit the parametric (exponential and Weibull) models to the data, and plot them with the Kaplan-Meier estimator (with its corresponding confidence band) as shown in figure 5.2. Both parametric estimators are decreasing, but the exponential estimator for $S(t)$ is linear, while the Weibull fit has almost the shape as the nonparametric fit Kaplan-Meier. Hence the exponential model is a very bad model one for this data, while the Weibull estimator would be appropriate, that is realistic.

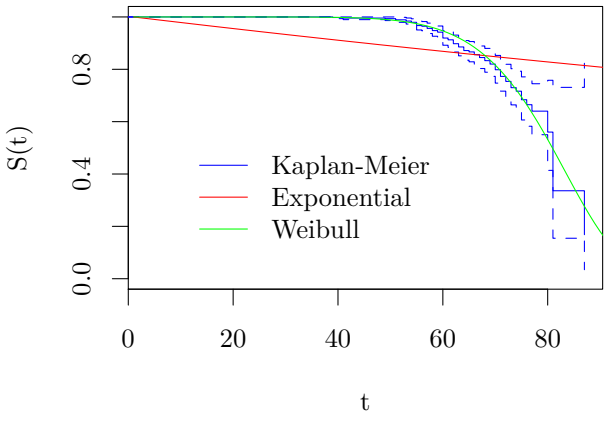


Figure 5.2: Plot of Kaplan-Meier & parametric estimators of the survivor function

To check if the Weibull model is adequate for the data, we use the graphical method presented in the last paragraph of the Results section, and obtain figure 5.3. We can see that the points follow $D0$ line pattern, meaning that the Weibull model is effectively appropriate for the data.

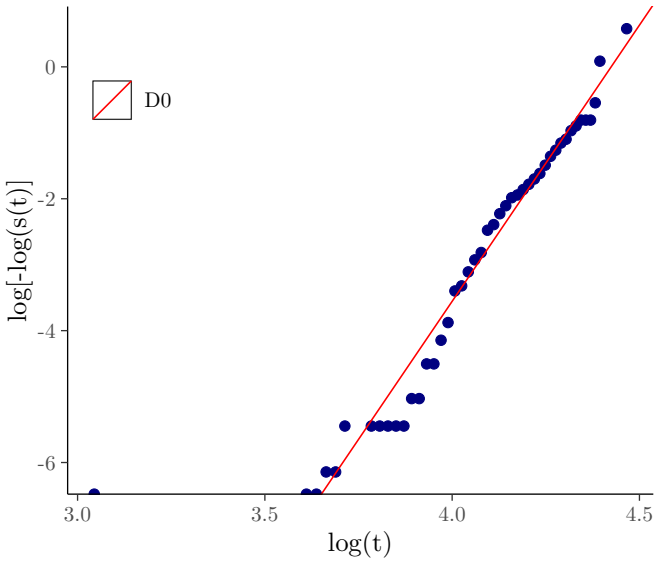


Figure 5.3: Plot of $\texttt{log}\{-\texttt{log}(\hat{S}(t))\}$ against $\texttt{log}(t)$, with the line $D0$ with slope α and intercept $-\texttt{log}(\lambda_{weibull})$

(b) Now we consider two groups of patients: smokers and non-smokers. Smokers make up 82.1% in the sample. For each group, the Kaplan-Meier estimators are computed, and plotted together with the corresponding confidence bands in figure 5.4. Then we want to test if the survival time depends on being a smoker. With the log-rank test, we have have the $\chi^2_1 = 2.7$ with p-value$= 0.1 > 0.05$. Hence the two groups do not share the same survival function, in other words the survival time do depend on being a smoker.

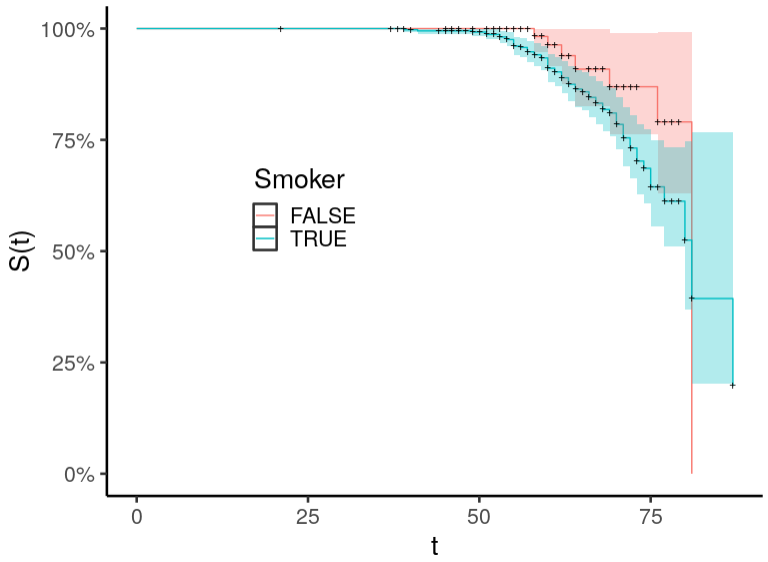


Figure 5.4: Plot of the Kaplan-Meier estimators fit for smokers and nonsmokers

We fit the Weibull model to both groups. The results are in table 5.1, and the figure 5.5 is the plot of the resulting parametric estimators for the survivor function together with the corresponding Kaplan-Meier estimators.

	Scale	Shape
smokers	0.01205778	8.161903
nonsmokers	0.01181161	11.13959

Table 5.1: Weibull model estimated parameters for each group

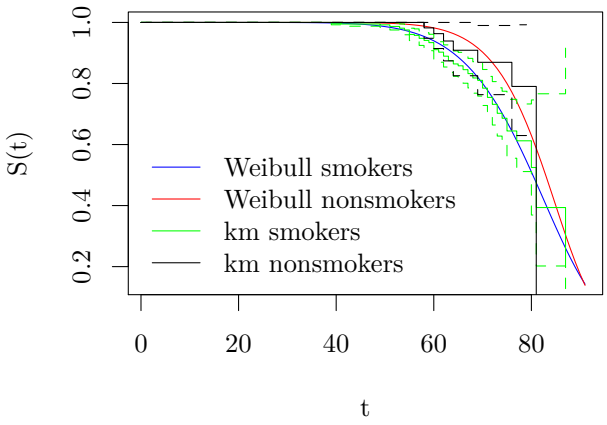
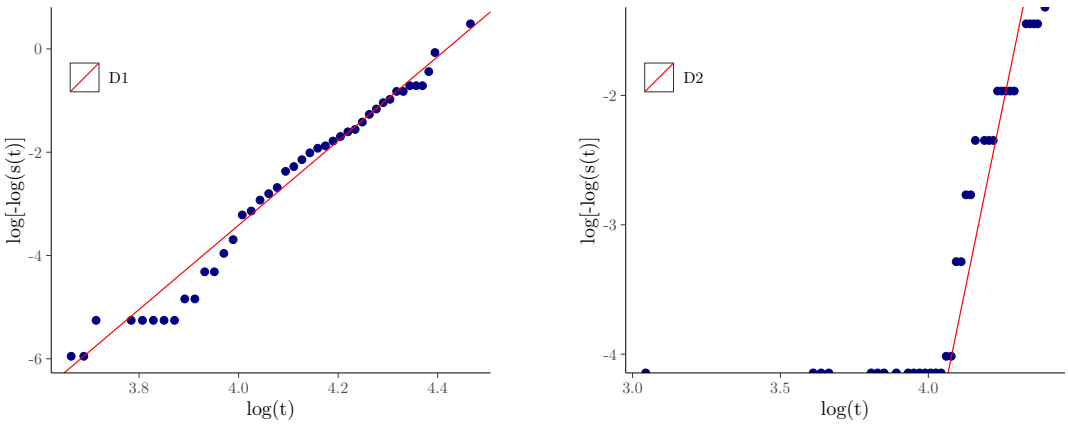


Figure 5.5: Plot of the nonparametric & parametric fit of the survivor function for each group

To check Weibull model assumption in both groups, we use figures in 5.6. We can see that the points in plot 5.6a follow $D1$ line pattern, meaning that the Weibull model is effectively appropriate for the smokers' group. However, the plot 5.6b highlight many points on the x-axis causing a different pattern with the line $D2$. Hence we can assume that the Weibull model is not appropriate for the nonsmokers group.



(a) Smokers: $D1$ is the line with slope α and intercept $-\texttt{log}(0.01205778)$

(b) Nonsmokers: $D2$ is the line with slope α and intercept $-\texttt{log}(0.01181161)$

Figure 5.6: Checking Weibull model assumption in both groups

CHAPTER

SIX

KERNEL DENSITY ESTIMATION

6.1 Problem description

Consider observations which are realizations of univariate random variables, $X_1, \ldots, X_n \sim F$ where F denotes an unknown cumulative distribution function. The goal is to estimate the distribution F. In particular, we are interested in estimating the density $f = F^{'}$, assuming that it exists.
Instead of assuming a parametric model for the distribution (e.g. Normal distribution with unknown expectation and variance), we rather want to be "as general as possible": that is, we only assume that the density exists and is suitably smooth (e.g. differentiable). It is then possible to estimate the unknown density function $f(\cdot)$.

6.2 Methods

Definition

Let $X_1, \ldots, X_n \overset{iid}{\sim} F$ with a given density $F^{'} = f$. A **kernel density estimator** for f is defined via

$$\hat{f}(x; h) = \sum_{i=1}^{n} K\Big(\frac{x - X_i}{h}\Big), \quad x \in \mathbb{R}, \quad h > 0.$$

Thereby $K : \mathbb{R} \to \mathbb{R}$, such that $\int_{-\infty}^{\infty} K(x)dx = 1$ is known as **kernel** and $h > 0$ is called **bandwidth**.
Some classical kernels:

1. $0.5\mathbb{I}(|x| \leq 1)$ (the rectangular or uniform kernel)
2. $(1 - |x|)\mathbb{I}(|x| \leq 1)$ (the triangular kernel)
3. $0.75(1 - x^2)\mathbb{I}(|x| \leq 1)$ (the Epanechnikov kernel)
4. $2^{-1/2}\texttt{exp}(-x^2/2)$ (the Gaussian kernel)

Now we want to find a practical way of choosing K and h. The optimal bandwidth is given by $h_{CV} = \texttt{arg}\min_{h>0} CV(h)$, where $CV(.)$ is the **(leave-one-out) cross-validation criterion**.

$$CV(h) = \int \{\hat{f}(x;h)\}^2 dx - 2\frac{1}{n(n-1)h}\sum_{i=1}^{n}\sum_{j\neq i} K(\frac{X_j - X_i}{h}).$$

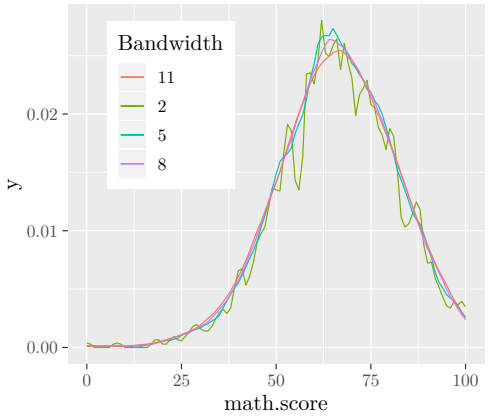
Then, the cross-validation kernel density estimator is define via

$$\hat{f}(x;h_{CV}) = \frac{1}{nh_{CV}}\sum_{i=1}^{n} K(\frac{X_i - x}{h_{CV}})$$

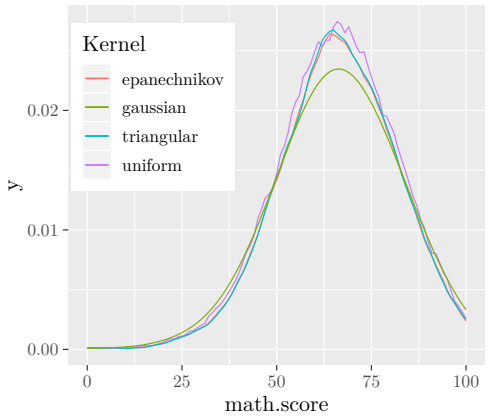
6.3 Results

The dataset used for the exercise is *StudentsPerformace.csv* and can be found on Kaggle datasets. In our analysis we will only consider the following variables:
`test.preparation.course`: If a student took part at the preparation course
`math.score`: Score on the math exam (0-100)
`reading.score`: Score on the reading exam (0-100)
`writing.score`: Score on the writing exam (0-100)

(a) After the implement of the kernel density estimation in an **R** function, we have the following plot 6.1a. Since we want to avoid under- or oversmoothing, the ideal bandwidth would be 8. This Bandwidth is then used to plot 6.1b with four different kernels. We can notice that the Epanechnikov kernel function would be ideal for the kernel density estimation because its curve is not too smooth or too rough. However, the kernels curves are pretty close, which is not the case for the bandwidths. Hence, the choice of the kernel does not matter very much as the choice of the bandwidth.



(a) with the Epanechnikov kernel and 4 different bandwidths



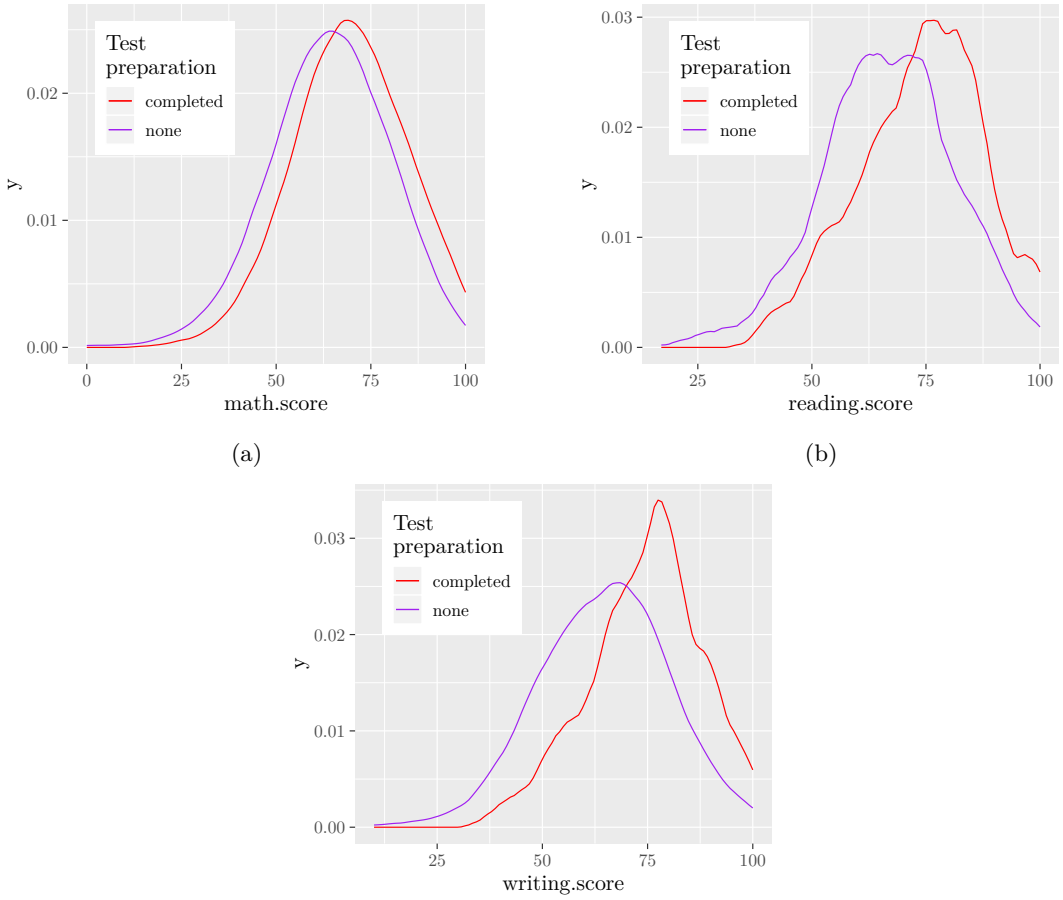
(b) with four kernel functions: bandwidth = 8

Figure 6.1: Plots of kernel density estimators of math.score

(b) After the implemention of the cross-validation (CV) criterion to find the optimal bandwidth, we use it to find the optimal bandwidth in density for all three scores math.score, reading.score and writing.score. The same is done with **R** built-in functions `bw.ucv`, and `bw.bcv`, then we have the table 6.1. The cross-validation criterion returns the highest optimal bandwidth for all three scores. We can notice that the obtained bandwidth with the implemented CV are greater than the ones of **R** built-in functions.

	CV	`bw.ucv`	`bw.bcv`
math.score	5.506	4.644	4.257
reading.score	4.465	3.756	4.393
writing.score	5.489	4.265	4.175

Table 6.1: Optimal bandwidth for each samples with different methods



(a) (b)

(c)

Figure 6.2: Densities plots of all three scores of the students that did not take part in the preparation course with the students who attended the preparation course

(c) From figure 6.2, we notice that the densities are skewed to the left. We also notice that they have the same shape, with different modes. This means that the two groups may have same distribution with different parameters.

CHAPTER

SEVEN

NONPARAMETRIC REGRESSION: LOCAL POLYNOMIALS

7.1 Problem description

To study the relation between a dependent variable Y and an independent variable X, the common method used is linear regression. When appropriate, this method is very useful as its suppose a simple model of the form

$$Y = \beta_0 + \beta_i x_i + \epsilon_i \tag{7.1.1}$$

This is advantageous since it is easy to interpret and to calculate. Moreover, when the assumptions on the residues ϵ_i are verified, we can run some tests on the parameters.
However, the restricted assumption of linearity is frequently not fulfilled, eventually when the data set is very large. In that case, we would like to find a complex model that will better highlight the relation between Y and X. A first approach for this aim would be to specify another parametric form for this relation, for example a transformation of the observations or a polynomial regression. Nonetheless it remains difficult to find the suitable relation since the form of the data does not really change after these transformations. That is why in this section, we opt for a non-parametric regression technique (local polynomials) in which data choose their own form of relation (the predictor does not take a predetermined form but is constructed according to information derived from the data) making things more flexible.

7.2 Methods

Let $(Y_1, X_1), \ldots, (Y_n, X_n)$ be iid as (Y, X) random variables, $Y \in \mathbb{R}$ and $X \in \mathbb{R}^d$. Consider a random design nonparametric regression model

$$Y_i = f(X_i) + \epsilon_i, \quad i = 1, \ldots, n$$

$$\mathtt{E}(\epsilon_i | X_i) = 0, \quad \mathtt{E}(\epsilon_i^2 | X_i) = \sigma^2.$$

Let$K : \mathbb{R}^d \to \mathbb{R}_+$ be a kernel function, and denote $e_k = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{R}^(\ell + 1)$ a unit vector with 1 at k-th position, $k = 1, \ldots, \ell + 1$. Moreover, we define

$$P(X_i - x) = \{1, (X_i - x), \ldots, (X_i - x)^\ell\}^t$$

$$X = \begin{pmatrix} 1 & (X_1 - x) & \cdots & (X_1 - x)^\ell \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (X_n - x) & \cdots & (X_1 - x)^\ell \end{pmatrix}, \quad Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

$$V = \texttt{diag}\left\{ K\left(\frac{X_1 - x}{h}\right), \ldots, K\left(\frac{X_n - x}{h}\right) \right\}$$

Then a **local polynomial estimator** of $f^{(k-1)}(x)$ is a linear estimator

$$\hat{f}^(k-1)(x) = (k-1)! e_k^t \left(\frac{1}{nh} X^t V X\right)^{-1} P(X_i - x) K\left(\frac{X_i - x}{h}\right) = \sum_{i=1}^n W_{k,i}(x) Y_i$$

with the weight function

$$W_{k,i}(x) = \frac{(k-1)!}{nh} e_k^t \left(\frac{1}{nh} X^t V X\right)^{-1} P(X_i - x) K\left(\frac{X_i - x}{h}\right).$$

The bandwiddth h can be chosen efficiently with the following GCV(generalized cross-validation)

$$GCV(h) = \frac{\sum_{i=1}^n \left\{ Y_i - \hat{f}(X_i; h) \right\}^2}{\{1 - n^{-1} \sum_{i=1}^n W_{k,i}(h)\}^2}$$

7.3 Results

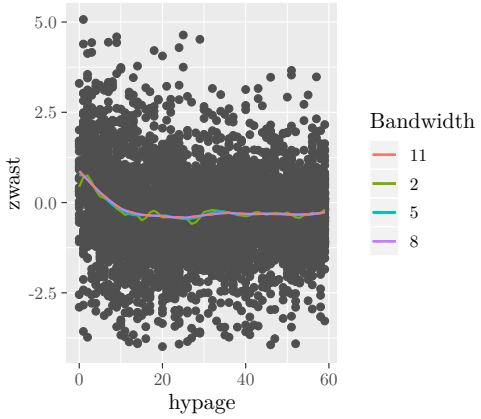
We use the dataset from Exercise 1 on Kenyan children. We are interested in the following two variables

hypage: Age of a child
zwast: Z-score for wasting

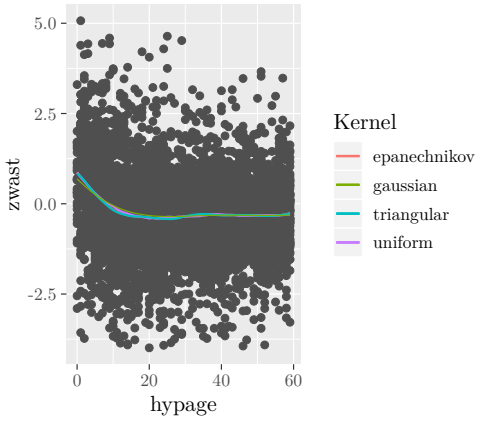
Z-score for wasting is defined as the weight of a child standardised with the median and standard deviation of children with the same height from the healthy population. We would like to investigate how the Z-score for wasting changes with age, that is we consider the model

$$\texttt{zwast}_i = f(\texttt{hypage}_i) + \epsilon_i, \quad \texttt{for} \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, n.$$

(a) After implementing a local polynomial fit and by fixing the polynomial degree to 1, we have the plots in figure 7.1. Since we want to avoid under- or oversmoothing, the ideal bandwidth from the plot 7.1a would be 8. This bandwidth is then used to plot the estimate of f with 4 different bandwidths (figure 7.1b). It is clear that the curves of the estimator present the same shape, and almost the same level of smoothness. In this case, all the kernels might be used for further analysis.



(a) with the Epanechnikov kernel and 4 different bandwidths



(b) with four kernel functions: bandwidth = 8

Figure 7.1: Plots of local polynomial estimator of f

(b) Now we want to find the optimal bandwidth with Generalised Cross Validation (GCV). For this aim, we implement a function that calculates the GCV, then with used Epanechnikov kernel and obtained GCV-bandwidth to estimate f using polynomial degrees from 1 to 4. The results are in table 7.1. We note that the GCV-bandwidths are less than 8, and are different. This means that we will have different estimators for different polynomial degrees.

Polynomial degree	1	2	3	4
GCV-bandwidth	4.999956	10.99995	10.99994	8.403991

Table 7.1: Optimal bandwidth for each polynomial degree

The plot of all four fits are in figure 7.2. We notice that the curves follow the same pattern in general, with less smoothness meaning that the obtained GCV-bandwidths are completely reasonable. However, the fitting curve for the polynomial degree 1 is very smooth compared to the others, especially with the fitting curve of the polynomial degree which is less smooth than the others. Moreover, the boundaries highlight slight signs of over-fitting(for polynomial degree 4), or under-fitting (for polynomial degree 2). Overall, using the polynomial degree 3 with the corresponding optimal bandwidth would be ideal.

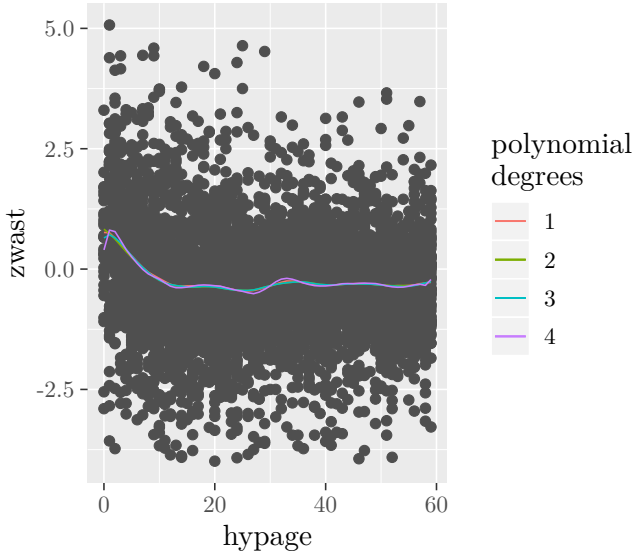


Figure 7.2: Plot of all four fits obtained using the GCV-bandwidths from table 7.1

(c) In this question, we use the function **localpoly.reg** of library *NonpModelCheck* to calculate the first derivative of the function of `zwast` with the GCV-bandwidth and polynomial degrees from 1 to 4. Figure 7.3 shows the plot of all four derivative fits. We can see noticeable differences on boundaries, since the curves start at very different `zwast` values. This suggest that some derivative fits(especailly the one from the polynomial degree 4) takes into account a lot of outliers, while others might use them less. Nevertheless, we do have signs of better `zwast` after 2 years. Actually, after 2 years, the derivative fitting curves head towards 0 (and we also have some fluctuations around 0). Finally, the derivative fits curves we got here are too smooth, therefore GCV-bandwidths from 7.1 are not reasonable or not optimal. We may solve this issue just by using the corresponding derivatives for each polynomial degree to find the optimal bandwidths.

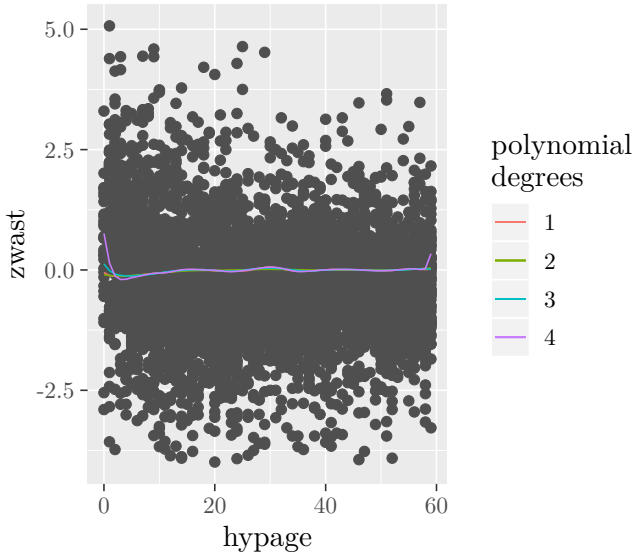


Figure 7.3: Plot of all four derivative fits obtained using the GCV-bandwidths from table 7.1

CHAPTER

EIGHT

NONPARAMETRIC REGRESSION: SPLINES

8.1 Problem description

In the previous chapter, we have explored the local polynomials regression method. However, this method suffers from Runge's Phenomena (especially when having lots of data). To solve this problem, we will generalized the local polynomials regression with splines.

8.2 Methods

Denote

$$\mathcal{P}_m = \{p : p(x) = \sum_{i=1}^{m} c_i x^{i-1}, \quad c_1, \ldots, c_m, x \in \mathtt{R}\}$$

Next, consider an interval $[a, b)(a, b \in \mathbb{R}, -\infty < a < b < \infty)$.
Let $a = \tau_0 < \tau_1 < \cdots < \tau_k < \tau_{k+1} = b$, $\tau_i \in \mathtt{R}$ and denote $\Delta = \{\tau_i\}_0^{k+1}$ a partition of $[a, b)$ into $k+1$ subintervals $[\tau_i, \tau_{i+1})$, $i = 0, \ldots, k$.
Definition Let Δ_k be a partition of $[a, b)$, then
$\mathcal{PP}_m(\Delta_k) = \{p : \exists p_0, \ldots, p_k \in \mathcal{P}_m$ such that $p(x) = p_i(x)$ for $x \in [\tau_i; \tau_{i+1})$, $i = 0, \ldots, k\}$
is the **space of piecewise polynomials of order** m **based on** Δ_k.

Definition
Let Δ_k be a partition of $[a, b)$. Let $m \in \mathbb{N}$ and $M = (m_1, \ldots, m_k)$ be a vector of integers with $1 \leqslant m_i \leqslant m$, $i = 1, \ldots, k$. Then, the space
$\mathcal{S}_m(M, \Delta_k) = \Big\{s : \exists s_0, \ldots, s_k \in \mathcal{P}_m$ such that $s(x) = s_i(x)$ for $x \in [\tau_i; \tau_{i+1})$, $i = 0, \ldots, k$ and $s_{i-i}^{(j)}(\tau_i) = s_i^{(j)}(\tau_i)$, $j = 0, 1, \ldots, m-1-m_i$, $i = 0, \ldots, k\Big\}$
is the **space of polynomial splines of order** m **and multiplicities** M **based on** Δ_k. Further we will deal with $\mathcal{S}_m(M = (1, \ldots, 1), \Delta_k)$ only and will denote it just $\mathcal{S}_m(\Delta_k)$.

Definition B-splines $N_i(x)$ of order m can be defined as:

1. m polynomials pieces of degree $m-1$, which join at $m-1$ inner knots
2. $N_i(x) > 0$ for $x \in [\tau_i, \tau_{i+m})$ and is zero for xs outside of this interval
3. $N_i(x)$ overlaps with $2(m-1)$ pieces of its neighbors
4. $\sum_{i=j-m+1}^{j} N_i(x) = 1$, $x \in [\tau_j, \tau_{j+1})$

Now consider a fixed design regression model with deterministic $\{x_i\}_{i=1}^n \in [0,1]$

$$Y_i = f(x_i) + \epsilon_i, \quad \texttt{cov}(\epsilon_i \epsilon_j) = \sigma^2 \delta_{ij}, \quad \texttt{E}(\epsilon_i) = 0, \quad i = 1, \ldots, n.$$

Regression function f is estimated by regression splines, that is

$$\hat{f}_n = N(\cdot)\texttt{arg}\min_{\beta \in \mathbb{R}^{k+m}} (Y - N\beta)^t (Y - N\beta) = N(\cdot)(N^t N)^{-1} N^t Y,$$

where $N = \{N(x_1)^t, \ldots, N(x_n)^t\}^t$ is the basis matrix with $N(x) = \{N_1(x), \ldots, N_{k+m}(x)\}$ as some basis of $\mathcal{S}_m(\Delta_k)$.

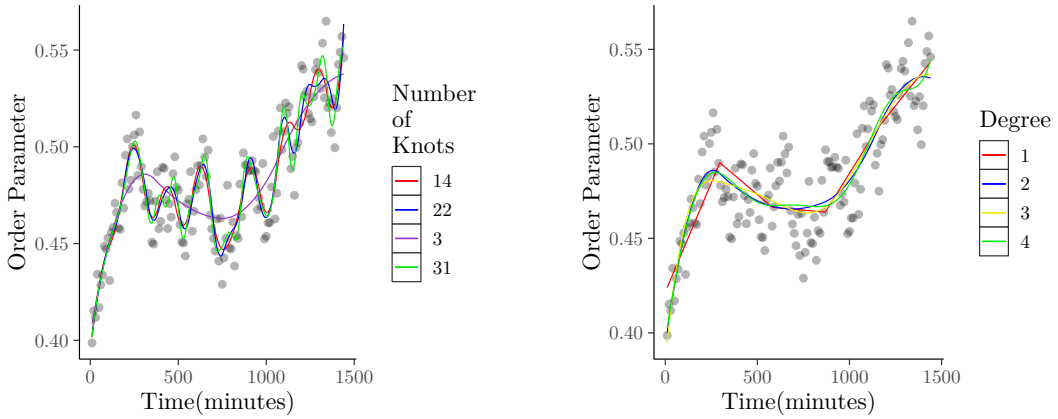
Furthermore, in order to find the optimal knot, one can use the following **generalized cross validation**,

$$\texttt{GCV}(k) = \frac{||\hat{f}_n - f||^2}{[1 - tr\{N(N^t N)^{-1} N^t\}/n]^2} = \frac{||\hat{f}_n - f||^2}{(1 - k/n)^2}.$$

8.3 Results

The dataset *stemcells.txt* contains 144 observations of the order parameter of a living stem cell observed every 10 minutes over 24 hours. We would like to understand how the order parameter evolves over time, i.e., we consider the model $OP_i = f(t_i) + \epsilon_i$, $i = 1, \ldots, 144$, where t_i are the time points. Thereby, ϵ_i are not independent and identically distributed, but rather follow an autoregressive process of the first order. That is, $\epsilon_i = \alpha\epsilon_{i?1} + \xi_i$ for $\xi_i \sim \mathcal{N}(0, \sigma^2)$ and $\alpha \in (0, 1)$. Note that the correlation matrix R of $(\epsilon_1, \ldots, \epsilon_n)^t$ is given by $R_{i,j} = \alpha^{|i-j|}$. For this data you can take $\alpha = 0.55$.

(a) First we implemented an R function that depends on the response, covariate, number of knots and spline degree that calculate a regression spline fit for f. By fixing the spline degree to 2 and estimate f with various number of knots, we get obtain the following figure 8.1a. It noticeable that the fitting curve is too smooth for small number of knots(which may cause underfitting) and less smooth for larger values (which may cause overfitting). Then by fixing the number of knots to 4, we estimate f using splines of degree from 1 to 4 and obtain the plot 8.1b. We can notice that per intervals, we have a line for the degree 1, which is curved as the the number increases. Hence for a good fit, we need to find the optimal number of knots and the spline degree.



(a) with spline degree 2, and various number of knots (b) with 4 number of knots and four spline degrees

Figure 8.1: Plots of regression spline fit for f

(b) In order to estimate the optimal number of (equidistant) knots, a function using the Generalised Cross Validation (GCV) is implemented. First, we ignore that the data are dependent, and obtain the results in table 8.1. Then we calculate the fits with the number of knots obtained with GCV and spline degrees from 1 to 4, and the resulting estimators are represented in plot 8.2. It is noticeable that we are in case of overfitting, making these estimators unreasonable.

Spline degree	1	2	3	4
GCV knots' number	29	33	35	21

Table 8.1: GCV knots' number for each spline degree

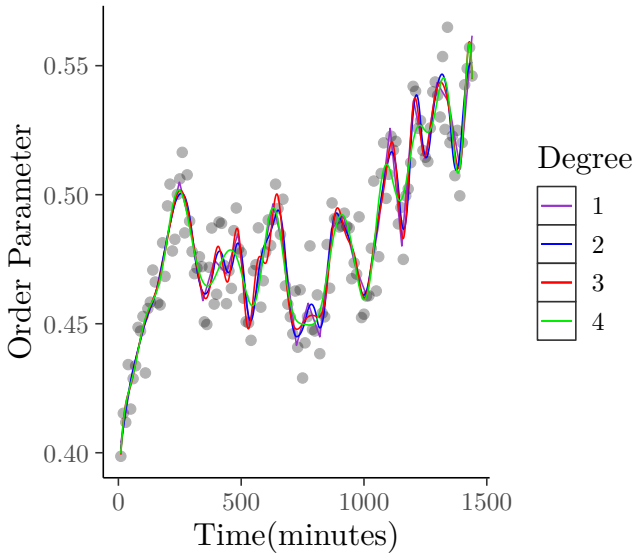


Figure 8.2: Spline regression estimators with obtained GCV knots' number

(c) In this question, we update the functions for regression splines and GCV so that they take into account that the errors ϵ_i follow and autoregressive process of order one, and have the results in table 8.2. Then we calculate the fits with the number of knots obtained with this updated GCV criterion and spline degrees from 1 to 4. The resulting estimators are plotted in figure 8.3. We notice that the estimators fit here are more relevant, since for each spline degree, the smoothness minimizes over- or under-fitting.

Spline degree	1	2	3	4
GCV knots' number	6	3	1	1

Table 8.2: GCV knots' number (auto-regressive) for each spline degree

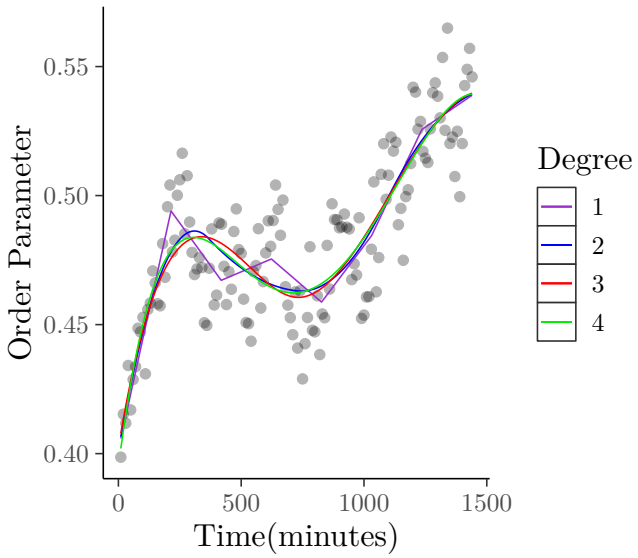
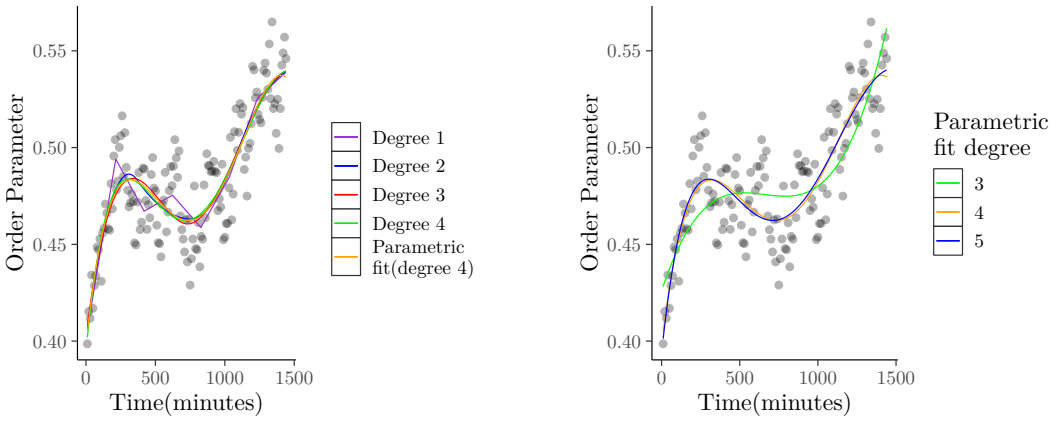


Figure 8.3: Spline regression estimators with obtained auto-regressive GCV knots' number

Furthermore, we add to the plot a parametric fit, such that $f(t_i) = \sum_{j=0}^{4} \beta_j t_i^j$.



(a) Parametric fit plot added to the plot 8.3

(b) Plot of the parametric for various degrees

Figure 8.4

We can notice from 8.4a that the parametric fit of degree 4 is very close to the splines fit with degrees 2, 3 and 4. Hence it is resonable to use a fitting model. Figure 8.4b indicates that it also works for polynomial of 5th degree, but not for 3rd degree, because an underfitting problem.

CHAPTER

NINE

MIXED MODELS

9.1 Problem description

Let's illustrate the problem in this section with the following exemple. The (simple) linear model state there is a linear dependence between explanatory variables $Y_{i,t}$, and the time t (dependent variable), with an error term $\epsilon_{i,t}$

$$Y_{i,t} = \alpha + \beta_t + \epsilon_{i,t}, \quad \epsilon_{i,t} \sim \mathcal{N}(0, \sigma^2)$$

t, α, β: fixed effects. α, β, σ: parameters to estimate. One limit of this model is that it considers $Y_{i,t}$ to be independent, which is not often the case as they can be correlated. Therefore, take this into account, a random effect is added to the fixed effects. We obtain a (linear) mixed model

9.2 Methods

The data we consider are of the form Y_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, k_i$. That is, there are n subjects (clusters) and for each subject i there are k_i observations. A mixed model can be represented as

$$Y = X\beta + Zu + \epsilon, \quad \mathtt{E}\begin{pmatrix} u \\ \epsilon \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathtt{cov}\begin{pmatrix} u \\ \epsilon \end{pmatrix} = \begin{pmatrix} \sigma_u^2 D & 0 \\ 0 & \sigma_\epsilon^2 \Sigma \end{pmatrix},$$

for $Y, \epsilon \in \mathbb{R}^N$, $N = \sum_{i=1}^n k_i$, $Z = \mathtt{blockdiag}(Z_1, \ldots, Z_n) \in \mathbb{R}^{N \times nm}$, $u \in \mathbb{R}^{nm}$ Typically, a normality assumption is made both for ϵ and u.
A good exemple of mixed models' application is in **small area estimation**, where it is assumed that the means in "small areas" differ for a random amount:

$$Y_i = X_i\beta + \mathbb{I}_{n_i} u_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}_{k_i}(0_{k_i}, \sigma^2 I_{k_i}), \quad u_i \sim \mathcal{N}(0, \sigma_u^2), , \quad i = 1, \ldots, n,$$

where in each ith small area there are k_i observations available.

In the exercise, we are interested in the estimation of u. Since it is assumed to be random, one typically uses the term "predictor" for $\hat{u}$, which is obtained as a *best linear unnbiased predictor* (**BLUP**)

$$\hat{u} = \frac{1}{\lambda} Z^t V^{-1} (Y - X\hat{\beta})$$

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

where $V = I_N + ZZ^t/\lambda$ for $\lambda = \frac{\sigma_\epsilon^2}{\sigma_u^2}$, $N = \sum_{i=1}^n k_i > p$

9.3 Results

For the data, we consider the survey and satellite data measuring the area for corn and soy fields in NorthCentral Iowa from 1978. The data set is available as *landsat* in the R-package JoSAE. We are interested in obtaining reliable estimates for the total size of corn and soy production for each of the 12 counties in the data set, respectively. Variables of interest:

`SegmentsInCounty`: number of of segments of county.

`SegementID`: identificator for segment.

`HACorn`: hectares of corn for given segment.

`HASoybeans`: hectares of soybeans for given segment.

`PixelsCorn`: pixels for corn for given segment.

`PixelsSoybeans`: pixels for soybeans for given segment.

`MeanPixelsCorn`: mean of pixels for corn over all segments in given county.

`MeanPixelsSoybeans`: mean of pixels for soybeans over all segments in given county.

`CountyName`: county identificator of the segment.

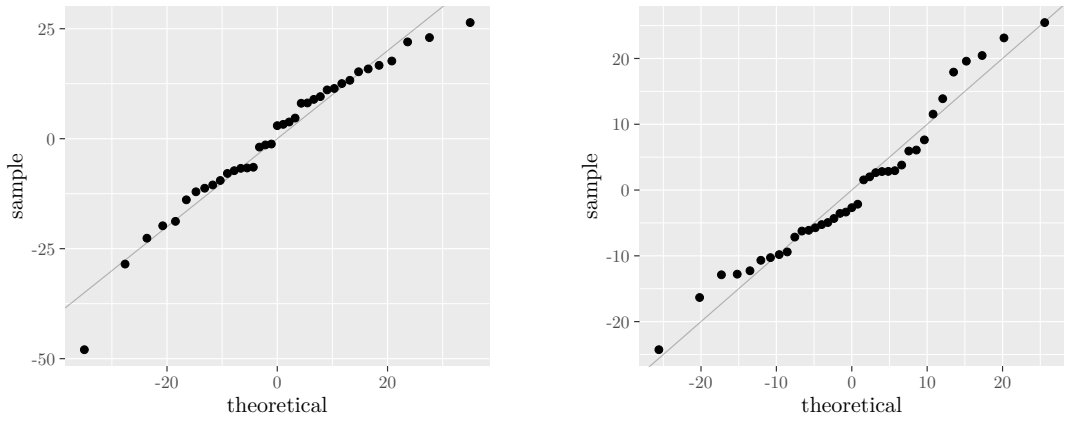
(a) In order to fit a linear model to both the hectares of corn and soybeans for segment for each county, we chose `PixelsCorn` and `PixelsSoybeans` variables since we want to estimate for the total size of corn and soy production. The limitations of this linear model is that it could not estimate the hectares of corn and soybeans for segment in some counties; probably because of a random effect that the model cannot take into consideration.

(b) We fit a linear mixed model $y_{ij} = x^t\beta + v_i + e_{ij}$ for both crops such that segments share the same countywide random effect. We can interpret from table 9.1 that Soybeans tend to have

	Fixed effects		Random effects (StdDev)	
	(Intercept)	Pixels	(Intercept)	Residual
Corn	5.4661894	0.3878358	7.926246	17.03993
Soybeans	-3.8223556	0.4756781	15.46753	13.41709

Table 9.1: Results of the linear mixed model

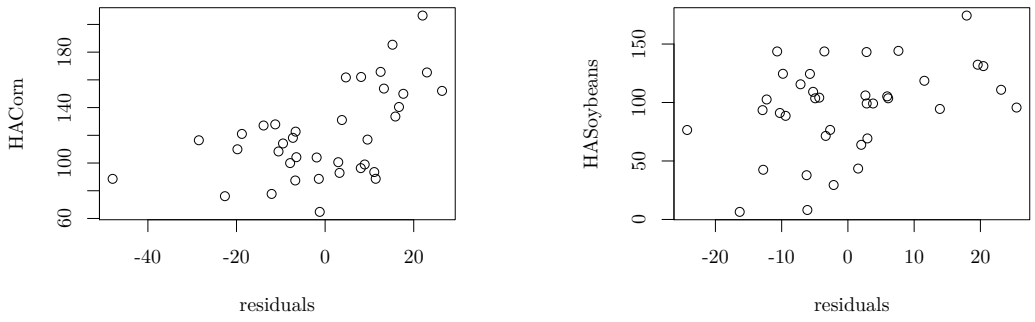
Figure 9.1 shows that the Q-Q plot indicate points falling on the identity line, which means that the residuals are well normally distributed.



(a) Corn linear mixed model (b) Soybean linear mixed model

Figure 9.1: Residuals Q-Q plot for normality assumption

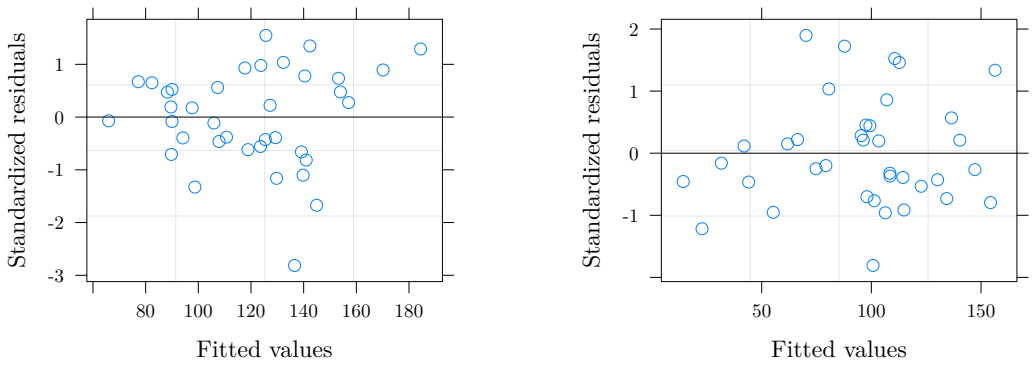
Plots in figure 9.2 show points randomly displaced, which means that the linearity assumption is fulfill.



(a) Corn linear mixed model (b) Soybean linear mixed model

Figure 9.2: Plots of explanatory variable of each model, against their residuals to check linearity assumption

Finally, homoskedasticity of the models is also fulfill, because plots in figure 9.3 show no patterns.



(a) Corn linear mixed model

(b) Soybean linear mixed model

Figure 9.3: Fitted values vs residuals for each model

(c) Now we want to compare and evaluate with respect to their reliability, in order to obtain predictions for $\mu_i = \bar{x}_{ip}^t \beta + v_i$. Here, for the ith county and a specified crop, x_{ip} is the population mean of the explanatory variables and $\bar{x}_i$ the mean over the observed segments only. Also $\hat{\beta}$ is the weighted least-squares estimator for β and $\gamma_i = \sigma_v^2(\sigma_v^2 + n_i^{-1}\sigma_\epsilon^2)^{-1}$ where n_i the number of observations in the ith county, σ_v^2 and σ_ϵ^2 the variances of random effect and error, respectively.

- Regression predictor: $\mu_i^0 = \bar{x}_{ip}^t \hat{\beta}$.
- Adjusted survey predictor: $\mu_i^1 = \bar{x}_{ip}^t \hat{\beta} + (\bar{y}_i - \bar{x}_i^t \hat{\beta})$
- (Empirical) BLUP: $\mu_i^\gamma = \bar{x}_{ip}^t \hat{\beta} + \hat{\gamma}_i(\bar{y}_i - \bar{x}_i^t \hat{\beta})$
- Survey predictor: $\bar{y}_i = n_i^{-1} \sum_{n_i}^{j=1} y_{ij}$

An estimate for the mean squared error $\texttt{MSE}_{\mu_i}(\mu_i^d) = \texttt{E}(\mu_i - \mu_i^d)$ for μ_i^d is given by

$$\widehat{\texttt{MSE}}_{\mu_i}(\mu_i^d) = (1-d)^2 \hat{\sigma}_v^2 + \frac{d^2 \hat{\sigma}_\epsilon^2}{n_i} + 2(d - \hat{\gamma}_i)(\bar{x}_{ip} - d\bar{x}_i)^t \hat{V}(\hat{\beta}) \bar{x}_i + (\bar{x}_{ip} - d\bar{x}_i)^t \hat{V}(\hat{\beta})(\bar{x}_{ip} - d\bar{x}_i),$$

where $\hat{V}(\hat{\beta})$ is the covariance matrix of $\hat{\beta}$. Tables 9.2 and 9.3 the respective MSE. We can notice the results really differ form one county to another, which tells us that in order ot have good predictions for μ_i, it might better to use different predictors fo each County.

Counties	μ_i^0	μ_i^1	μ_i^{γ}	$\bar{y}_i$
Cerro Gordo	72.05803	283.61578	61.66718	302.12264
Hamilton	71.97262	280.62908	61.72427	306.22149
Worth	71.73962	287.68234	61.13340	292.90282
Humboldt	69.41864	138.57817	53.48326	167.63961
Franklin	65.34791	95.65654	3.27852	97.63366
Pocahontas	64.34482	92.96560	43.50445	99.09076
Winnebago	65.82172	96.77393	43.28370	96.78756
Wright	65.61789	95.44570	43.89862	101.67489
Webster	64.08903	72.66995	38.47434	72.59204
Hancock	63.18738	59.00156	34.43863	59.81280
Kossuth	62.27490	58.09003	33.53265	58.50767
Hardin	63.19271	50.62384	32.28565	51.50612

Table 9.2: Results of estimated predictors' MSE for Corn

Counties	μ_i^0	μ_i^1	μ_i^{γ}	$\bar{y}_i$
Cerro Gordo	233.9442	182.87015	116.21477	208.61640
Hamilton	235.5158	180.10725	107.79506	180.73675
Worth	234.7851	179.91494	108.30543	183.17044
Humboldt	231.4987	96.67826	74.44887	100.92499
Franklin	223.0584	60.24062	49.31044	60.55884
Pocahontas	219.5358	59.75710	49.03094	60.22818
Winnebago	222.6925	60.45358	49.56226	61.04741
Wright	225.2767	61.91758	50.95105	62.20606
Webster	217.4335	44.87112	38.54084	45.10319
Hancock	219.6162	37.21818	32.99170	37.69316
Kossuth	220.0136	36.18285	31.92317	36.19472
Hardin	220.1881	31.62848	28.61561	31.80195

Table 9.3: Results of estimated predictors' MSE for Soybeans

(d) The estimation of the total county field size for both crops, using the BLUP and Survey from (c) is highlighted in the table 9.4 and on maps 9.4 and 9.5. We can notice that the size given by the BLUP and the Survey are completely different. This confirms the fact that we need to chose the appropriate predictor in each County(by taking the one with the lowest MSE for each County).

	Corn		Soybeans	
Counties	**BLUP**	**Survey**	**BLUP**	**Survey**
Cerro Gordo	36232.59	48947.27	14850.16	1534.673
Hamilton	37163.87	28934.53	18334.02	20850.800
Worth	32692.98	22032.77	17913.60	21267.008
Humboldt	33537.21	43869.76	18034.58	7739.632
Franklin	43710.09	50475.53	12457.91	9868.135
Pocahontas	28151.91	26365.93	27971.91	29333.507
Winnebago	33973.49	32903.88	18073.97	16418.839
Wright	37119.98	43470.81	24957.12	21649.008
Webster	29310.33	30829.88	27162.40	27916.228
Hancock	39007.80	34376.57	19952.53	23338.179
Kossuth	33598.79	32926.76	24392.10	24112.061
Hardin	42742.30	37426.91	13180.96	15894.074

Table 9.4: Results of the estimation the total county field size for both crops, by the BLUP and survey form part (c)

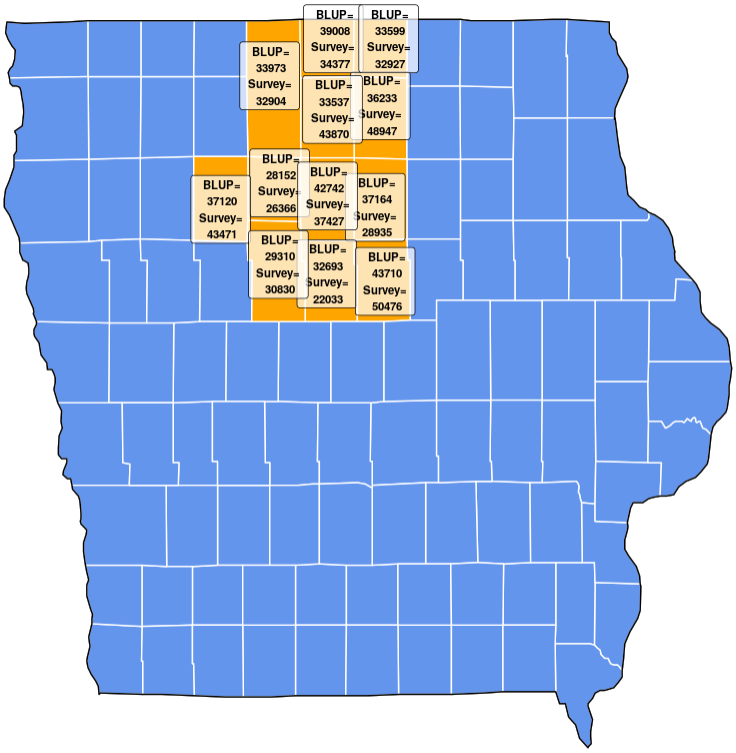


Figure 9.4: Map of Iowa with estimated the total county field size for corn

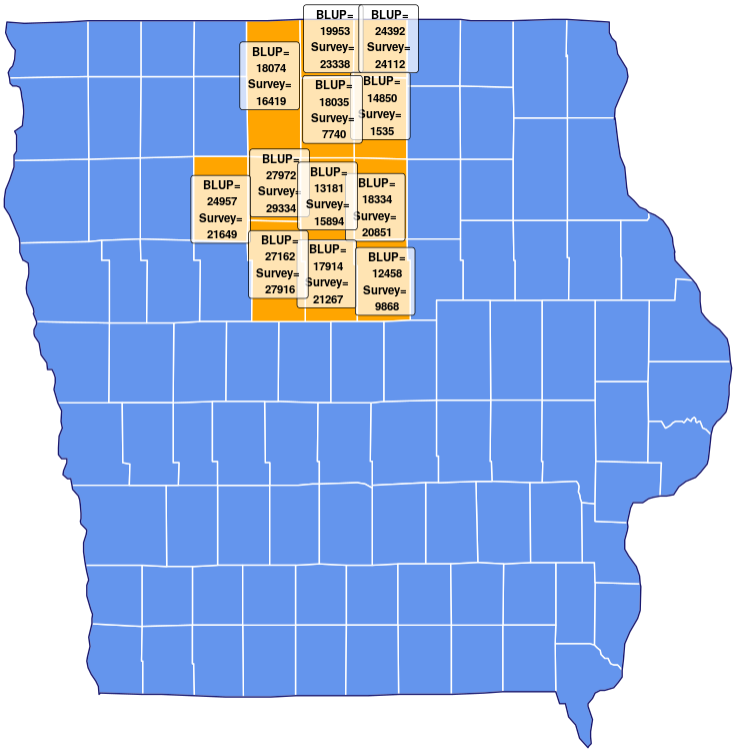


Figure 9.5: Map of Iowa with estimated the total county field size for soybean

The huge differences between the slopes and intercepts of the models describe notable differences between the crops.

CHAPTER

TEN

PARTIAL LEAST SQUARES

10.1 Problem description

In some problems of linear regression or prediction phenomenon, explicable variables can be correlated, sometimes causing multicollinearity. This phenomenon that occurs most of the time in big data analysis, is a consequence of bad results related to regressions' coefficients estimated with least squares. To solve this problem, we have principal component analysis (PCA), and the partial least squares (PLS). We are using the latter.

10.2 Methods

Let $x \in \mathbb{R}^m$ and $y \in \mathbb{R}$ be two zero-mean random variables. Denote $\sigma = cov(x)$ and $\delta = cov(x, y)$. Assume $y = x\beta + e$, $\mathtt{E}(e) = 0$ and $cov(e) = \sigma^2$. This is called a population model.
Now, let $Y = (y_1, \ldots, y_n)^t \in \mathtt{R}^n$ be a vector of n independent copies of y and $X = (x_1^t, \ldots, x_n^t)^t \in \mathtt{R}^{n \times m}$ be a matrix of n independent copies of x. In particular, $Y = X\beta + \epsilon$, where $\mathtt{E}(\epsilon) = 0_n$ and $cov(\epsilon) = \sigma^2 I_n$. Denote $A = X^t X$ and $b = X^t Y$; note that A and b are proportional to the sample estimators of σ and δ, respectively.
Denote now $K_d(b; A) = span(b, Ab, \ldots, A^{d-1}b)$ the d-dimensional Krylov space, where $d \leq rk(A)$ should be chosen data-driven. Then, the partial least squares estimator can be defined as

$$\hat{\beta}_{PLS}^d = \mathtt{arg} \min_{\beta \in K_d} = ||Y - X\beta||^2$$

There are several ways to compute PLS, but in **R** it can be done with libraries **plsr** and **predict.plsr**.

10.3 Results

The data of interest for this exercise can be find on the page myPersonality.org . There are three files:

users.csv contains psychodemographic user profiles

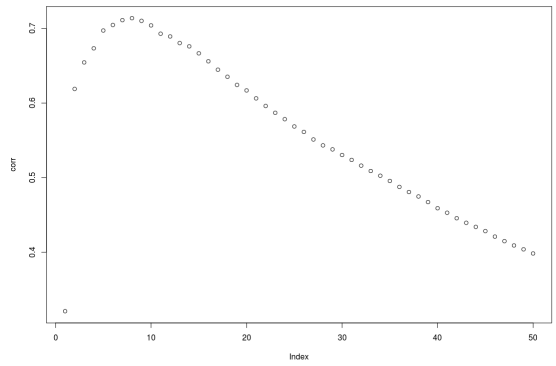
likes.csv contains anonymised IDs and names of Facebook Likes

users-likes.csv contains the associations between users and their Likes

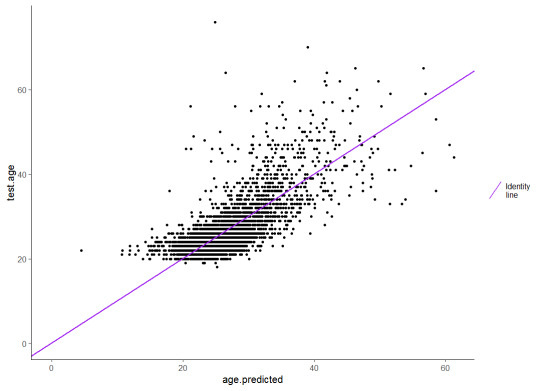
The seed value for used in for the exercise is 1122.

(a) Check the R code related to this exercise

(b) With the PLS, we would like to find a model that allows to predict the user?s age based on Likes (s)he made. First we split the dataset into a test and training set, by sampling randomly two thirds of all rows to include into the training set and the rest will be the test set. Then on the training set fit PLS regression models with age as a response variable and with up to 50 PLS components. For each PLS model dimension (from 1 to 50), we obtain the prediction on the test set and compare it with the true age values from the test set, and calculating the Pearson correlation coefficient. The plot 10.1a clearly shows that we have a maximal value $d_{opt} =$, which is then used as the model of dimension. The plot 10.1b is the Plot the values predicted by the PLS model of dimension $d_{opt} = 8$ on the test set against corresponding age values from the test set. This plot shows that the model is pretty accurate since points are along the identity line.



(a) Plot of the Pearson correlation coefficients against model dimension



(b) Plot of the values predicted by the PLS model of dimension d_{opt} on the test set against corresponding age values from the test set

Figure 10.1

(c) Now we investigate which Likes predict the age best, using the best predictive model identified in (b). From figure 10.2, we can notice that likes with the largest positive effect on the age, that is elderly people likes, are effectively topics of elders. While likes with the largest negative effect on the age, that is youngsters likes, are also things for young people. Hence we can assume that the model is accurate.

```
                                    The Hangover
                                       0.7501700
                                    M&M's U.S.A.
                                       0.8149683
                                   Bret Michaels
                                       0.8281955
                                             PBS
                                       0.8759854
I am so old I have actually dialed a rotary phone before!
                                       0.9416454
                                Small Business Saturday
                                       1.1580465
```

(a) The 6 Likes having the largest positive effect on the age

```
                                  The Constitution
                                      -0.6911923
                                   Country music
                                      -0.6075625
                                Humans of New York
                                      -0.5611161
                                   Michael Phelps
                                      -0.5518198
I've lived in 3 decades, 2 centuries & 2 milleniums & I'm not even 20 yet!
                                      -0.5488915
                                       Red Bull
                                      -0.5189740
```

(b) The 6 Likes that have the largest negative effect on the age

Figure 10.2

(d) Now we repeat the analysis removing users that made less than 60 Likes and Likes that have less than 120 users in (a). Due to the huge matrix' size, I could not run de model. However, it is expected to have the same likes and users as in figure 10.2, but with more significant coefficients. For a plot line the one in figure 10.1b, we should have points more closer to the identity line.