Stochastisches Praktikum II - WiSe 2018/2019

Tatyana Krivobokova
Tutor: Peter Kramlinger

Februar 2019

Chapter 1

General Remarks and Programming Basics

1.1 General Remarks

Why R?

First of all R is free. It is an open source project, which is similar to the S language environment which was developed at Bell Laboratories. Although there are some differences, much code written for S runs unaltered under R. Moreover R is growing at a rapid pace and there are many packages available covering a huge field of topics and methods.

Report and Certificate

In order to get the credits and the certificate for this course, you need to give a detailed report on the exercises.
The report should contain a description of the problem, a description of the methods used to solve the problem and a detailed discussion of the results. You should include all graphics of interest and of course especially the ones you were explicitly asked to generate. Moreover, you should also include the code used to solve the problem (see section 1.2).

1.2 Programming Basics

The goal of this section is to provide you with some basic programming guidelines. In the heat of programming all the code you produce may seem perfectly understandable and clear to you. Anyway, if you don't follow some basic rules, you will run into trouble trying to understand your own code some time later. (There is a saying, that programs older than two weeks might be written by someone else...) In the special case of this practical course, you have to keep in mind, that your code will be corrected by a third person, who must be able to make sense of what you did.

Guidelines for writing code

- Keep it simple

  Think of the reader. Don't write just for yourself. Break down complexity into simpler chunks. Avoid implicit or obscure language features. Minimize scope, both logical and visual.

  Minimizing scope and breaking down complexity are not contradictory to each other. It may seem cool to do a very complex statement in one line, but if you ever tried to

understand a program written by someone else, you surely have already cursed this particular programming style. When in doubt you should strive for clarity first, then efficiency.

- Keep it readable

  Use informative variable names. Good code can be read like a book. Make names clearly unique. Avoid abbreviations whenever possible. Name variables with noun or adjective noun combinations.

  It is quite tempting to use short names for variables and functions. This, however, is one of the main problems of many programs. Stories of programmers who used the name of their girlfriend as the name of a time variable might be known to you in the context of the year 2000 problem. In writing statistical programs people tend to name their variables 'x', 'xx', 'y0' and so on. Use more meaningful names. Especially abbreviations often seem totally clear at the moment but tend to lose their clarity very fast. 'predicted.value' is much more understandable than 'prdVl'. The usual way of separating two words in R is the use of a point like in 'linear.fit'.

- Comment your code

  Clearly comment necessary complexity. Be clear and concise. Say what is happening and why. Do not restate code. Keep code and comments visually separate.

  Comments are the most important part of a program, if you try to understand it later. Comment coherent units of code. Make it easy to look for a special functionality of your code. A good indicator whether you should comment or not is the amount of time you spend on producing the code. If you thought on some special lines of code for hours, they might be worth a short comment.

  Be aware, that comments may also decrease the readability of source code. They might even be misleading, if you fail to update them when changing your program. That is why you should make the code as clear as possible to reduce the need for comments.

  When using program packages like R it is sometimes very helpful to comment on the functions and the parameters of the functions you use. This is especially true for R since many names and parameters of functions do not meet the claim for clarity and intuitive comprehensibility.

Chapter 2

Exercises

1 Tidyverse

Dataset

The dataset *childrenfinal.dta* is obtained from **Kenyan Demographic and Health Survey 2003** and contains various variables sampled in 2003 on the Kenyan children of age between 0 and 5 years. The data are cross-sectional, there are no same children observed. There are 4686 observations on 177 variables, most of the variable names are self-explained.

Exercises

In this exercise the goal is to learn how to work with tools of *tidyverse* package.

(a) The data `childrenfinal.dta` are given in the STATA format. Read the data into R using an appropriate function from *tidyverse*. Next, remove all variables that start with "s", "v" and "m", followed by a number (avoid listing all of them). Check all the remaining variables. Do all variables have reasonable variable type (character, factor, double, integer, etc)? Convert the variables to a suitable type, if necessary.

(b) Make a smaller tibble that contains variables `hypage, ruralfacto, female, zstunt, zweight, zwast, adm2`. Variable `zstunt` is the so-called Z-score for stunting and is defined as the height of a child standardised with the median and standard deviation of heights of children at the same age from a healthy population. Children with Z-score less than -2 are defined to be stunted. Make a scatter plot of `zstunt` against `hypage`. Add a smooth line to the plot. Comment on the results. Now, make smooth plots of `zstunt` against `age` for females and males on one plot, add a suitable legend. Use different colors for males and females. Similarly, plot `zstunt` against `age` for urban and rural children. Comment on results. Experiment with different aesthetics, themes and font sizes for the plots, report your favourite(s).

(c) Plot the map of Kenya with all counties listed in `adm2`. Colour the county areas according to the mean of `zstunt` in the corresponding county. Note that one county (Isiolo) is missing in the data. Make suitable legend and add county names (or corresponding labels) to the map. Comment on the results. In which counties children are stunted?

(d) Finally, write the tibble from (b) into a text file. This file will be used in Exercise 7.

2 Random number generation

Exercises

(a) Switch the default random number generator in R to `Wichmann-Hill`. 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 function `rbinom`. Plot the empirical probability density functions of all three samples on one panel. Comment on the results. Switch the random number generator back to its default.

(b) The aim is to simulate $N = 10\,000$ standard normal distributed random variables with density $f(x) = (2\pi)^{-1/2} \exp(-x^2/2)$ using accept-reject method and a generator for uniform random variables only. As a candidate density g use the density of the standard Cauchy distribution $g(x) = \{\pi(1 + x^2)\}^{-1}$.

- First determine the best value of the constant c, such that $f(x) \leq cg(x)$.
- Obtain N standard normal random variables using the accept-reject method, generating Cauchy distributed random variables using inversion method. Compare estimated and theoretical acceptance probabilities. Plot a histogram of the obtained sample and add the standard normal density to the plot. Make a QQ-plot. Comment on the results.
- Show that it is not possible to simulate from the standard Cauchy density using the accept-reject method with a standard normal candidate density.

R functions

You may find useful the following R functions: `apply, RNGkind, which`.

3 Bootstrap

Dataset

The dataset *shhs1.txt* has been obtained from **Sleep Heart Health Study**, where more information on the data can be found. We will be using only the following variable

`rdi4p:` respiratory disturbance index

Exercises

In the following set the sample size $n = 100$, the number of bootstrap replications $R = 1000$ and the number of Monte Carlo samples $M = 1000$.

(a) Simulate a sample $(x_1, \ldots, x_n)$ from the Weibull distribution with the scale parameter $\lambda = 13$ and shape parameter $k = 1$. The variance of a Weibull distributed random variable is given by $\sigma^2 = \lambda^2 \left[\Gamma(1 + 2/k) - \{\Gamma(1 + 1/k)\}^2\right]$, while the median $x_{med} = \lambda\{\log(2)\}^{1/k}$. We aim to build confidence intervals for σ based on a statistic $\hat{s}^2 = (n-1)^{-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$ and for x_{med} based on the sample median.

- Build two-sided bootstrap percentile confidence intervals for σ and x_{med} at the significance level $\alpha = 0.95$. Use M Monte Carlo samples to estimate the coverage probability of both confidence intervals. Estimate also the average interval length. Comment on the results. How the results change if $n = R = 1000$ and $n = 100$, $R = 5000$? Comment on both accuracy of the coverage probabilities and the average length of confidence intervals in both cases.

- Use R function `bcanon` from the package *bootstrap* to build bootstrap accelerated bias-corrected confidence intervals both for σ and x_{med}. Use M Monte Carlo samples to assess the coverage probability and the average length of the confidence intervals. Comment on the results and differences to bootstrap percentile confidence intervals. Check estimated $\hat{z}_0$ and $\hat{a}$, comment on these values. Explain the performance of all obtained confidence bands.

(b) Consider now variable `rdi4p` from the dataset *shh1.txt*. Plot the histogram of this variable and describe the empirical distribution. Build bootstrap percentile and bootstrap accelerated bias-corrected confidence intervals for the standard deviation and median. Comment on the results.

R functions

You may find useful the following R functions: `apply, bcanon` (library *bootstrap*), `sample`

4 Generalised linear models

Dataset

The dataset *student-mat.csv* can be found on **Kaggle**. This page contains the full description of the data and all the variables. 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.

Exercises

(a) First we need to identify the distribution of each of `G1, G2, G3`. Can each of these variables be assumed to follow a normal distribution? Justify your answer using suitable arguments and graphical tools. Can each of `G1, G2, G3` be assumed to follow a Poisson distribution? Are there signs for over-dispersion or any other anomalies in the distributions of any of `G1, G2, G3`? Support your answer using suitable arguments and graphical tools.

(b) Fit a suitable (generalised) linear model to explain `G1` including all explanatory variables (Model 1). Are all covariates significant? Comment on the goodness-of-fit of this model. Calculate the Pearson residuals and Anscombe residuals and assess how closely they follow a normal distribution. Pursue the residual analysis and comment if the fitted (generalised) linear model is adequate for the data.

(c) Take Model 1, but reduce the covariates to `sex`, `Fedu, studytime, failures, schoolsup, famsup, goout` (Model 2). Are all the covariates significant? Interpret the effect of each covariate on the grade. Assess the goodness-of-fit of this model. Perform analysis of deviance test to compare Model 1 and Model 2. Comment on the results. In Model 2 replace `goout` by `Walc` to get Model 3. How one can compare Model 2 and Model 3? Which model delivers a better fit? Justify your answer.

R functions

You may find useful the following R functions: `glm`.

5 Survival analysis

Dataset

The dataset *Thoracic.txt* can be found on **UCI Machine Learning Repository**. This page contains full description of the data and all the variables. We will be using the following three:

`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

This is an example of interval censoring (failure happened at some point within one year), but we will consider failure times to be at `AGE` +1.

Exercises

The *survival* package provides the function `survfit` that generates from the data a `survfit.object`. Acquaint yourself with the components of the `survfit.object` data structure.

(a) First compute nonparametric estimators of the survivor function: Kaplan-Meier and Flemming-Harrington. Plot them on one plot together with 95% confidence bands. Comment on the differences of both estimators. Fit the exponential and Weibull models to the data. Plot the Kaplan-Meier estimator together with the corresponding confidence bands and both parametric estimators for the survivor function on one plot. Comment on the results. Which parametric model is more realistic? Use appropriate graphical tools to check if the Weibull model is adequate for the data. Comment on the results.

(b) Now consider two groups of patients: smokers and non-smokers. How large is the proportion of smokers in the sample? For each group compute Kaplan-Meier estimators, plot them on one plot together with the corresponding confidence bands. Test formally if the survival time depends on being a smoker using the log-rank test. Comment on the results. Fit the Weibull model to both groups; make sure to estimate both scale and shape parameters for each group. Plot the resulting parametric estimators for the survivor function together with the corresponding Kaplan-Meier estimators. Is the Weibull model an appropriate assumption in both groups?

R functions

You may find useful the following R functions: `survfit` (library *survival*).

6 Kernel density estimation

Dataset

The dataset *StudentsPerformace.csv* can be found on **Kaggle datasets**. This page contains also the background information on the data. 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)

Exercises

(a) Implement kernel density estimation in an R function that depends on the sample, bandwidth and a kernel. Read the data into R. Estimate and plot the density of `math.score` with the Epanechnikov kernel and 4 different choices of the bandwidth, putting them onto one plot. Next, fix the bandwidth you find most reasonable and plot kernel density estimators for 4 different choices of kernel functions (Gauss, Epanechnikov, uniform, tridiagonal), putting them onto one plot. Do not forget to put legends on both plots. Comment on the effect of the bandwidth and of the kernel function on kernel density estimators.

(b) Implement the cross-validation criterion to find the optimal bandwidth. Compare your resulting bandwidths with the ones obtained by built-in R functions `bw.ucv` and `bw.bcv` used in `density` for all three scores `math.score`, `reading.score` and `writing.score`.

(c) Use your implementation of the kernel density estimator with the cross-validated bandwidth to compare graphically densities of all three scores of the students that did not take part in the preparation course with the students who attended the preparation course. Comment on the results.

R functions

You may find useful the following R functions: `apply, density, integrate, optimize, outer, Vectorize`.

7 Nonparametric regression: local polynomials

Dataset

Consider again 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 $zwast_i = f(hypage_i) + \epsilon_i$, for $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, \ldots, n$.

Exercises

(a) Write an R function that calculates a local polynomial fit and depends on the response, covariate, bandwidth, polynomial degree, kernel and the number of derivatives. First fix the polynomial degree to 1, estimate f with 4 different bandwidths and plot all resulting fits on one plot. Next, choose the most reasonable bandwidth, fix the polynomial degree to 1, estimate f with 4 kernel from the exercise on the kernel density estimation and put all fits on another plot. Comment on the results.

(b) Write a function that calculates the optimal bandwidth with Generalised Cross Validation (GCV). Use Epanechnikov kernel and GCV-bandwidth to estimate f using polynomial degrees from 1 to 4. List obtained GCV-bandwidths for each polynomial degree and comment on the results. Plot all four fits putting the curves on the same plot. Comment on the differences in the overall fits and at the boundaries. Do you find fits with the obtained GCV-bandwidths reasonable?

(c) Use 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. Plot all four derivative fits putting the curves on the same plot. Comment on the difference. Interpret the results. Are there indications that the Z-score is improving after 2 years? Do you find derivative fits with the obtained GCV-bandwidths reasonable?

R functions

You may find useful the following R functions: `lm, influence`.

8 Nonparametric regression: regression splines

Dataset

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 $\alpha = 0.55$.

Exercises

(a) Write an R function that depends on the response, covariate, number of knots and spline degree that calculate a regression spline fit for f. First fix the spline degree to 2 and estimate f with various number of knots (take 4 options), plotting all fits on one plot. Comment on the results. Now, fix the number of knots to 4, estimate f using splines of degree from 1 to 4 and plot all obtained fits on one plot. Comment on the results.

(b) Write a function that estimates the optimal number of equidistant knots with Generalised Cross Validation (GCV), first ignoring that the data are dependent. Calculate the fits with the number of knots obtained with GCV and spline degrees from 1 to 4. Plot resulting estimators on one plot. Do you think these estimators are reasonable?

(c) Now we would like to incorporate our knowledge about dependence in the data into regression spline estimation and GCV criterion. Update your functions for regression splines and GCV so that they take into account that the errors ϵ_i follow and autoregressive process of order one. Calculate the fits with the number of knots obtained with this updated GCV criterion and spline degrees from 1 to 4. Plot all resulting estimators on one plot. Comment on the results. Add to the plot a parametric fit, such that $f(t_i) = \sum_{j=0}^{4} \beta_j t_i^j$. Do you think it is reasonable to model the order parameter as a polynomial of 4th degree? What about polynomials of degree 5 or 3? Justify your answer.

R functions

You may find useful the following R functions: `spline.des` (library *splines*), `toeplitz`.

9 Mixed effects models and small area estimation

Dataset

Consider the survey and satellite data measuring the area for corn and soy fields in North-Central Iowa from 1978. Information is only available for few segments for the counties of interest. Detailed information was made available by passes of NASA's LANDSAT satellites. The number of pixels for both crops is given up to segment level. 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.

Exercises

(a) Fit a suitable linear model to both the hectares of corn and soybeans for segment for each county. Explain your choice of included parameters. What are the limitations of the linear model? You might want to create a `groupedData`-object and use the `nlme`-function `lmList`.

(b) 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. Make and justify the model assumptions and discuss the fits. Do they exhibit notable differences between the crops?

(c) In order to obtain predictions for $\mu_i = \bar{x}_{ip}^t\beta + v_i$, four predictors are compared and evaluated with respect to their reliability. Here, for the i-th county and a specified crop, $\bar{x}_{ip}$ is the population mean of the explanatory variables and $\bar{x}_i$ the mean over the observed segments only. Further, $\hat{\beta}$ is the weighted least-squares estimator for β and $\gamma_i = \sigma_v^2(\sigma_v^2 + n_i^{-1}\sigma_e^2)^{-1}$, where n_i the number of observations in the i-th county and σ_v^2 and σ_e^2 the variances of random effect and error, respectively. Also, $\bar{y}_i = n_i^{-1}\sum_{j=1}^{n_i} y_{ij}$.

- 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: $\mu_i^{\gamma_i} = \bar{x}_{ip}^t\hat{\beta} + \hat{\gamma}_i(\bar{y}_i - \bar{x}_i^t\hat{\beta})$.
- Survey predictor: $\bar{y}_i$.

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

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

where $\widehat{V}(\hat{\beta})$ is the covariance matrix of $\hat{\beta}$. Create a list with the predictions using each of the above predictors and print the respective MSE for each county and both crops. Discuss the results.

(d) Estimate the total county field size for both crops and plot the results by the BLUP from part (c) as well as the predictor only relying on the survey data in a table and onto a map of Iowa. You may use the packages `ggplot2` for plotting and `maps` and `mapdata` for modelling the data frame. Comment on the results.

R functions

You may find useful the following R functions: `lme` (library *nlme*).

10 Analysis of big data with partial least squares

Dataset

On page **myPersonality.org** you can find the article *Mining Big Data to Extract Patterns and Predict Real-Life Outcomes* by M. Kosinski, Y. Wang, H. Lakkaraju, and J. Leskovec, Psychological Methods, 2016 and the corresponding data. 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 article contains information on these data sets and some hints on the analysis in R.

Exercises

(a) Read all three data sets into R. Update the `users-likes` matrix adding two columns. The first column in the ith row should contain the row number in the `users` matrix, where the ith user of `users-likes` appears. The second column in the ith row should contain the row number in the `likes` matrix, where ith Like of `users-likes` matrix appears. With the help of these two new columns and function `sparseMatrix` from the library *Matrix* build a matrix `UL` with 1 at position i, j, if user i made Like j. Next, remove users that made less than 80 Likes and Likes that have less than 150 users. Finally, store the sparse matrix as a regular matrix in R.

(b) With the help of Partial Least Squares (PLS) we would like to find a model that allows to predict the user's age based on Likes (s)he made. First split the `UL` matrix, as well as the corresponding `age` vector (found in `users.csv`) into a test and training set. For this, sample randomly two thirds of all rows to include into the training set and the rest will be the test set. To ensure comparability of the results set `set.seed{1122}` before sampling. 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), obtain the prediction on the test set and compare it with the true `age` values from the test set, calculating the Pearson correlation coefficient. Plot the Pearson correlation coefficients against model dimension and find the PLS model dimension, d_{opt}, say, that

corresponds to the maximal Pearson correlation. Plot the values predicted by the PLS model of dimension d_{opt} on the test set against corresponding `age` values from the test set. Add the identity line and comment on the results.

(c) Let us now investigate which Likes predict the age best, using the best predictive model you identified in (b). Find 6 Likes that have the largest positive effect on the age and 6 Likes that have the largest negative effect on the age. Interpret and comment on the results.

(d) Rerun the analysis removing users that made less than 60 Likes and Likes that have less than 120 users in (a). How does this influence the results?

R functions

You may find useful the following R functions: `match, plsr predict.plsr` (both library *pls*), `sample`.