Applied Statistics in R
Exercises

Sebastian Hellmann

Winter Semester 2024/25 TUM School of Management
Chair of Behavioral Research Methods

Course regulations

1 Course exam - the report

To pass the course and get credits, you must submit a report with accompanying R code. The report may be written in English or German. You can (but don't have to) work together in pairs. Please ensure that everyone contributes equally to the results.

Content of the report

For the submission, chose **three exercises out of exercises 5 to 10**. For each exercise, the report should include:

- a brief description of the problem
- a description of the methods to solve the problem
- a detailed discussion of the results
- all graphics of interest (not only the ones that are specifically asked for)

Code files

The R code has to be self-contained. You can load packages, but if you split your code across different scripts, write a main script for each exercise that sources the different parts. Please, include the loaded data sets in the submission and try to use relative paths, such that other users do not have to adapt the paths when running the script. → See also 3 for more information on how to write good code.

Alternatively, you can write your report and code together in RMarkdown files and send the raw Rmd-files together with the PDFs.

Deadline for submission: **September 13, 2025**

Please send your reports in a zip-folder, including the report files and R-code named with your names and the course name per mail.

2 Hints for writing the report

Write your report with a focus on the description of problems and results. Structure your text with headings, subheadings, and paragraphs. Best practice would be to enumerate graphics and tables and reference them in the text, when you interpret the results. Code is best not included unless it is necessary for some explanations. Outputs should also be only included, if necessary for interpreting the results and should always be wrapped to the text width! (E.g. when showing summaries of model fits.) **Visualize your outcomes!** Do not only include the plots I ask you to do. Try to summarize and visualize the important aspects of your analyses in figures and describe what we see. Do not present results in form of page-long tables of numbers.

References

You may include references to blog-posts or even better proper literature. Please use footnotes for mentioning references and provide full citations (use a List of References, if you have many).

Data sets

Please, include the data sets in the submission and try to use **relative paths** in your code. This way, I may run the code directly on my machine without adapting paths or copying the data set in the right location.

Using packages

If you use functions from packages, which were not recommended and do not belong to the standard packages (mainly the tidyverse), please include a brief description as code comments or in the report text what the function does precisely.

R Markdown

If you use R Markdown, write your report, setting ‘echo=FALSE‘ and include only code which is necessary for specific explanations. However, I am happy to receive submissions that include a knitted file with ‘echo=TRUE‘ which helps me in the grating.

3 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 make a very complex statement in one line, but if you have 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 for 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 the 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 of whether you should comment or not is the amount of time you spend on producing the code. If you thought about 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.

Know your resources

Learning how to program can be challenging in the beginning. The best thing you can do is to just try, probably fail, and learn from errors! → Learning programming is trial-and-error learning. Some (hopefully) useful hints to make your life easier:

Function documentation

If you have a problem or questions concerning a specific function or want to explore resources from the R documentation, use:

```
> help.start()
```

You can open the documentation of a specific function in two ways. First, using the console and using `?` or the `help` function. Examples for the `sum` function:

```
> help(sum)
> ?sum
```

Alternatively (depending on your OS), you can type the function name in your source panel, move your cursor to the function, and press F1.

Tutorials and online forumns

In addition, there are a lot of good tutorials and forums that answer (almost) any question (particularly StackOverflow!). If you want to do something you have not done before, just use your favorite search engine and try to describe your problem. Also, large language models (such as ChatGPT) are pretty powerful in creating functioning code for specific problems. To maximize your learning in this course, I would ask you to try to **make as little use of LLMs as possible**.

Cheatsheets

If you are new to a programming language and you don't want to use online search for everything, I would recommend using cheatsheets, e.g.

- base R
- tidyverse (e.g., dplyr and ggplot2)

Read the error message

In the beginning, R's error messages might be cryptic. Just copy-pasting the error message and the function you were using into a search engine is a good approach in the begging. Besides, following checklist might help you to deal with errors (not exhaustive!):

- Check whether you have the right number of left and right parenthesis (especially, if the console shows a `+` sign, you missed a closing bracket!
- `Error: Object 'xyz' not found` or `Error: Could not find function 'xyz'`:
  - Did you misspell a variable/function name?
  - Did you put brackets after a variable name or missed brackets after a function name?
  - Did you try to use a variable/function that was not yet declared (because you are not running a script in the right order)?
- Check whether the arguments you pass into functions have the right type (by comparing to the documentation)
- Did you provide the arguments in the right order or named the arguments?

Exercises

1 Sample statistics

Dataset

Go to UCI Machine learning repository and download the data on the white wine quality. This page contains also the background information on the data. In our analysis we will only consider the following variables:

- `volatile.acidity`: Volatile acidity
- `residual.sugar`: Residual sugar
- `pH`: pH level
- `quality`: Wine quality in a score between 0 and 10

Exercises

(a) Read the data into R. Add to the data frame a new binary variable `good` which is 1 if `quality > 5` and 0 otherwise. We would like to compare `volatile.acidity` and `residual.sugar` for good and bad wines.

(b) First consider variable `residual.sugar`.

- Plot histograms of `residual.sugar` for good and bad wines using different methods available in R to choose the bin width. Comment on the shape of both histograms and differences in distribution, if any.
- Calculate the summary statistics for both wine groups, that is: mean, median, standard deviation, interquartile range, minimum and maximum of both samples. Display the results in a table and comment on the differences between both groups, if any.
- Generate boxplots for both samples, placing them into one graphic. What do you observe?
- Generate a QQ-plot to compare two groups. Make sure to choose the same range for both axes. Add a $y = x$ line to the plots. Comment on the results.
- Plot the empirical distribution functions of both groups in one graphic. Use different styles and add a legend. Interpret the results.

(c) Consider now `volatile.acidity` for good and bad wines. Use boxplots, histograms, QQ-plots, summary statistics and empirical distribution functions to compare this variable for good and bad wines. Comment on the results.

R functions

You may find useful the following R functions: `read.csv, summary, boxplot, qqplot, abline, ecdf`.

2 Examine the distribution of data

Dataset

Consider the dataset from the previous exercise and its variable `pH`.

Exercises

(a) Plot a histogram of `pH` for all wines and add to the plot a normal density, estimating the parameters from the data. Produce same histograms with corresponding normal densities for good and bad wines separately. Do you observe any differences in the distributions?

(b) Generate QQ-plots of `pH` for good, bad and all wines to compare empirical quantiles of the samples to the theoretical quantiles of a normal distribution. Produce PP-plots for all three datasets. Comment on the differences between QQ-plots and PP-plots. Do you think all samples follow a normal distribution?

(c) Plot the empirical distribution functions F_n for all three datasets. Add the pointwise confidence bands for $\alpha = 0.05$ using the central limit theorem and Slutzky's lemma (ensure that the confidence bands are in $[0; 1]$).

(d) Plot the empirical distribution functions F_n for all three datasets together with the uniform confidence bands for $\alpha = 0.05$. Compare to the bands obtained in (c).

(e) Plot the empirical distribution functions of `pH` for good and bad wines together with the uniform confidence bands in one plot. What can you conclude from this plot?

R functions

You may find useful the following R functions: `hist, dnorm, qqnorm, qqline, stepfun`.

3 Maximum likelihood estimation

Let X be Laplace distributed with parameters $(\mu, \sigma)^t \in \mathbb{R} \times (0, \infty)$, i.e., X has density

$$f(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x-\mu|}{\sigma}\right).$$

Exercises

(a) Consider a sample of independent observations $(X_1, \ldots, X_n)$. Show – mathematically – that the maximum likelihood estimator for μ is the median of the sample. Is it unique? (No coding required in this exercise, use similar argumentation as in the slides about MLE).

(b) Generate $n = 20$ independent realisations of X with $\mu = 1$ and $\sigma = 1$. Determine the maximum likelihood estimator of μ based on this sample using R function `quantile`. Function `quantile` has 9 types of sample quantiles. Experiment with the different types of quantiles that are most suitable for the data (justify). Are there any differences? Increase now the sample to $n = 1000$ and compare different median estimators. Comment on the result.

(c) Write your own function that calculates the maximum likelihood estimator for a Laplace sample numerically using R function `optimise`. Describe how R function `optimise` finds the maximum. Can you employ a Newton-Raphson algorithm for this problem? Generate

$n = 20$ and $n = 1000$ independent realisations of X with $\mu = 1$ and $\sigma = 1$. Calculate the maximum likelihood estimators based on both samples with your function and using `quantile`. Compare both estimators, comment on the results.

(d) Let us now study the distribution of the maximum likelihood estimator. For this, calculate $M = 5000$ maximum likelihood estimators of μ based on the sample of $n = 20$ random variables generated from the Laplace distribution with $\mu = 1$ and $\sigma = 1$. Repeat the same for the sample size $n = 1000$. Use histograms and QQ-plots to check if both Monte Carlo samples follow a normal distribution. Compare variances of both distributions, comment on the results.

R functions

You may find useful the following R functions: `rlaplace` (in package `rmutil`), `optimise`, `quantile`.

4 Linear Regression

Dataset

Go to Kaggle.com and download the data on house prices. This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015. There are altogether 19 variables, but we will consider in the analysis only the following ones:

- `price`: Price
- `bedrooms`: Number of bedrooms
- `bathrooms`: Number of bathrooms per bedroom
- `sqft_living`: Square footage of the home
- `floors`: Total floors in house
- `view`: Has been viewed (1 for viewed; 0 for not viewed)
- `condition`: How good is the condition (from 1 to 5)
- `grade`: Grade given to the housing unit based on King County grading system (from 1 to 13)
- `yr_built`: Year the house was built

Exercises

(a) Estimate a linear model with the response variable price and all remaining variables as covariates. Are all variables significant? How large is R^2 and how can this be interpreted? Perform the residual analysis to validate the model. Are there any departures from the linear regression model assumptions?

(b) Produce a histogram and a QQ-plot of the response variable `price`, as well as of its log-transform `log(price)`. Compare both distributions to the normal one. Fit now a linear model with the response variable `log(price)`. Compare the estimated model with the one from (a) in terms of R^2, significance and effect of covariates and model fit (via residual analysis). Which model is more adequate?

(c) In the model from (b) interpret the effect of each covariate on the response. Plot each covariate against `log(price)`. Is the assumption of the linear dependence between covariates and response plausible for all covariates? Add to the model from (b) squared terms for `yr_built` and `sqft_living`. Are these terms signiffcant? Does adding these two terms improve the model fit in terms of R^2?

(d) Now we would like to compare how well models from (b) and (c) make prediction. For this divide the dataset into a training and a test set. Sample randomly 10 806 rows to include into the training set and the rest will be the test set. To ensure comparability of the results `set.seed(1122)` before sampling. Fit both models on the training set and make prediction on the test set. Calculate the mean squared difference between predicted values and values of `log(price)` from the test set for each model. Which prediction error is smaller? Try to extend the model to improve the prediction: my best model gives prediction error of 0.09557445.

R functions

You may find useful the following R functions: `update, sample, predict`.

5 Penalized regression

Dataset

Consider the dataset `Hitters` on baseball players, included in the R packages `ISLR`.
It contains the following 20 variables:

- `AtBat`: Number of times at bat in 1986
- `Hits`: Number of hits in 1986
- `HmRun`: Number of home runs in 1986
- `Runs`: Number of runs in 1986
- `RBI`: Number of runs batted in 1986
- `Walks`: Number of walks in 1986
- `Years`: Number of years in the major leagues
- `CAtBat`: Number of times at bat during his career
- `CHits`: Number of hits during his career
- `CHmRun`: Number of home runs during his career
- `CRuns`: Number of runs during his career
- `CRBI`: Number of runs batted in during his career
- `CWalks`: Number of walks during his career
- `League`: A factor with levels A and N indicating player's league at the end of 1986
- `Division`: A factor with levels E and W indicating player's division at the end of 1986
- `PutOuts`: Number of put outs in 1986

- `Assists`: Number of assists in 1986
- `Errors`: Number of errors in 1986
- `Salary`: 1987 annual salary on opening day in thousands of dollars
- `NewLeague`: A factor with levels A and N indicating player's league at the beginning of 1987

We would like to explain `Salary` of a player using the other variables. The function `glmnet` in the package `glmnet` is used to fit both ridge and lasso regression. Supply $\alpha = 0$ for ridge and $\alpha = 1$ for lasso when calling `glmnet`. Make sure you understand the interpretation of λ in both cases (note that it is not the same for lasso and ridge!). Read the documentation `?glmnet` to understand what is the exact form of the objective function that is being minimised.

Exercises

(a) Load the dataset into R and create a new dataset containing only those players for which all data is available.

(b) Find the *condition number* (ratio between largest and smallest eigenvalue) of $X^t X$, where y is the salary and X represents all the other variables. What can you say about the condition number? Does it help if you standardise the design matrix, such that its columns (without the intercept) have mean zero and variance one?

(c) Fit a standard linear model (no regularisation) and a ridge regression with $\lambda = 70$. Compare the size of the coefficients in the two models. What do you observe?

(d) The value 70 for λ is arbitrary and we would like to

find a data-driven way to choose it. The criterion to compare is the mean squared prediction error as in exercise 4 (d) on linear regression. Split the data randomly into a training and a test set with `set.seed(1122)`.

(e) Write a function that takes λ as argument, fits a ridge regression on the training sets and calculates the mean squared prediction error on the test set. Run this function on a logarithmic grid (e.g., `10^seq(from = 10, to = -2, length = 100)`). Plot the results against $log(\lambda)$ and graphically find the value λ_{opt} that minimises the mean squared prediction error.

(f) Fit a ridge regression with λ_{opt} on all the data, and interpret some of the coefficients. Which are the most important variables? Are there coefficients that equal zero exactly?

(g) Repeat parts (d), (e) and (f) for lasso instead of ridge. Are there now coefficients that are equal to zero?

R functions

You may find useful the following R functions: `complete.cases, model.matrix, eigen, range, glmnet, sapply, coef`

6 Logistic regression

Dataset

Go to UCI Machine learning repository and download the data on blood donation. This page contains also the background information on the data. The goal is to build a model that allows to predict best if a donor will donate blood. The dataset contains the following variables:

- `recency`: months since last donation
- `frequency`: total number of donations
- `amount`: total blood donated in c.c.
- `time`: months since first donation
- `donation`: 1 stands for donating blood, 0 stands for not donating blood

Exercises

(a) Read the data into R and

fit a generalised linear model with the binary response `donation` and covariate `frequency` using the canonical link function. Fit the same model replacing the covariate by `amount`. Compare it to the first model. Plot the variable `frequency` against `amount`. Comment on the results. Do you need both of these variables in the model?

(b) Fit now the GLM model with the response `donation` and covariate `recency` using all link functions available in the `glm` function. Compare obtained estimators and comment on the differences.

(c) Now we would like to build a model that makes the best prediction for the blood donations.

- First divide the dataset into a training and a test set. Sample randomly 374 rows to include into the training set and the rest will be the test set. To ensure comparability of the results `set.seed1122` before sampling. Fit a GLM model with the response donation and canonical link on the training set, choosing appropriate covariates. Predict the model on the test set.
- With the predicted probability perform the classification: set the predicted ith value of `donation` to 0, if the corresponding ith predicted probability is less than 0.5 and to 1 otherwise. Assess the goodness of your classification calculating the classification error

$$CE = \frac{1}{374}\sum_{i=1}^{374} |y_i^{test} - \hat{y}_i^{test}|,$$

where y_i^{test} is the ith value of `donation` from the test set and $\hat{y}_i^{test}$ is its prediction. Try to extend the model to improve the classification error. Can you beat a performance of 0.21? (If not, just try!)

7 Generalized linear models

Dataset

The dataset *student-mat.csv* can be found on Kaggle. This page contains a full description of the data and all the variables. Variables `G1`, `G2`, and `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, and 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, and 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, or 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.

8 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: total number of segments in 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 $\mu_i = \bar{x}^t_{(p)i}\beta + v_i$, four predictors are compared and evaluated with respect to their reliability. Here, for the i-th county and a specified crop, $\bar{x}_{(p)i}$ 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 $\gamma_i = \sigma_v^2\left(\sigma_v^2 + n_i^{-1}\sigma_e^2\right)^{-1}$, 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: $\mu_i^0 = \bar{x}^t_{(p)i}\hat{\beta}$.
- Adjusted survey predictor: $\mu_i^1 = \bar{x}^t_{(p)i}\hat{\beta} + \left(\bar{y}_i - \bar{x}_i^t\hat{\beta}\right)$.
- (Empirical) BLUP: $\mu_i^{\gamma_i} = \bar{x}^t_{(p)i}\hat{\beta} + \hat{\gamma}_i\left(\bar{y}_i - \bar{x}_i^t\hat{\beta}\right)$.
- Survey predictor: $\bar{y}_i$.

(Here, $0, 1$, and γ_i are superscripts.) An estimate for the mean squared error $\mathrm{MSE}_{\mu_i}\left(\mu_i^d\right) = \mathrm{E}\left(\mu_i - \mu_i^d\right)$ for μ_i^d is given by

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

where $\hat{V}(\hat{\beta})$ 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`).

9 Principal component analysis

Dataset

The dataset iris is a classical dataset in statistics, and has been already analysed in
Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, Part II, 179-188.
It contains the following variables, all measured in centimeters:

- `Sepal.Length`: length of the sepal

- `Sepal.Width`: width of the sepal
- `Petal.Length`: length of the petal
- `Petal.Width`: width of the petal
- `Species`: type of the iris: setosa, versicolor, or virginica.

There are 50 flowers from each of the three species. All measures are in centimeters. The dataset is accessible automatically in R; try `?iris`. We shall use principal component analysis for visualisation of the data and K-means classification. In R, this can be done with the functions `prcomp` and `kmeans`.

Exercises

(a) Create a reduced dataset discarding the `Species`. Calculate loadings and scores for the reduced dataset using the empirical covariance matrix. What proportion of total variation in the data is explained by the first two principal components? Interpret the first two principal components.

(b) Do the same, now using the empirical *correlation* matrix. Are the results similar?

(c) Create a new dataset where the petal length is measured in millimetres instead of centimetres. Have the principal components and proportion of variance explained changed when using the covariance matrix? And when using the correlation matrix?

(d) Henceforth use the original dataset (with all measurements in centimetres) and the covariance matrix. Plot the first two principal components against each other, marking the different species by colour. What do you observe? Would you expect that this two-dimensional plot of the data give a reasonable representation of the relative position of the observations in the original four-dimensional space?

(e) Perform K-means clustering for the first two principal components obtained in (a) with $K = 3$. Try to find the best solution to the clustering problem (the output of the algorithm depends quite heavily on the seed/starting values). Do clusters coincide with the iris species? What is the classification error? Note that you may need to relabel ”by hand” the result in `kmeans(...)$cluster` when comparing it with the variable `Species`. The latter needs to be converted into an integer. Is the classification error when using the entire dataset much smaller?

(f) Comment on the shape of the cluster K-means generates and the geometry of the data. Find and use a classification algorithm that better suits the geometry of the data.

R functions

You may find useful the following R functions: `prcomp, kmeans, as.integer`.

10 Time Series

Dataset

Access the `cmort` dataset from the `astsa` package.
The data contains the weekly cardiovascular mortality in Los Angeles County from 1970 to 1979 from following study: Shumway, R.H. (1988). *Applied statistical time series analysis.* Prentice-Hall,

Englewood Cliffs

Exercises

(a) Fit an AR(2) to the data using linear regression as in the example on the Recruitment data presented in the lecture.

(b) Use the estimated coefficients from (a) to forecast the following 4 weeks together with a 95%-CI.

(c) Now use the Yule-Walker method to estimate the model. Compare the estimates and standard errors of the coefficients from the Yule-Walker method to the results from (a).

(d) Predict the following 4 weeks using the estimations from the Yule-Walker method.

(e) Compare the estimated standard errors of the coefficients obtained by linear regression with their corresponding asymptotic approximations as given by the asymptotic distribution of the estimators on the slides.

(f) Try to fit an ARMA(2,2) model to the data. Does the more complex model provides a better fit or is an AR(2) model enough?

(g) Compare the models you fit in (a), (c) and (f) to the data visually and visualize their predictions.

R functions

You may find useful the following R functions: `ar.yw, predict, arima`.