Computational Statistics (Programmierung mit R)

Portfolioprüfung Teil 2: Implementierungsaufgaben

Prof. Dr. Benjamin Buchwitz

SoSe 2022

Expectations

For each student one task that will be introduced in the following has been prepared. Each task has to be solved **individually** and **autonomously**.
In the first step, you will make yourself familiar with the theoretical background which is needed to solve the task. This means, the theory (e.g. a method) has to be presented and explained in your own words. Describe and explain hypotheses, equations etc., if necessary. Why is the topic - you work on - of interest? Whenever you use literature, cite it!
In the second step, you will do the implementation.
In some of the tasks, you will also have to check and document the functionality and correctness of your implementation as it is an essential step to be sure that your code works appropriately. Furthermore, document your code as it was shown in the lecture and exercise about *Software Development* so that the single implementation steps are easy to understand.

Organization and formal criteria

Your results must be submitted in the form of a PDF-Document that was generated from RMarkdown. Please use the fhswf-seminarpaper Template that is part of the `fhswf` Package (Buchwitz 2022). This template contains a declaration of authorship (Ehrenwörtliche Erklärung). **Your results cannot be graded if this declaration is missing!**

Please ensure that the format of your results and graphics is in accordance to academic standards (label the axes, add a caption to plots, add a legend and so on . . .) and hand in your results (PDF-, RMarkdown- and Bib-file as well as further R scripts (if necessary)) via Email to your corresponding supervisor buchwitz.benjamin@fh-swf.de or behrens.hannah@fh-swf.de **until September 1, 2022**.

Task 1: Implementation of the White Test for Heteroskedasticity and estimation of the *heteroscedasticity-consistent covariance matrix* for a linear regression model

student: **Marvin Schwarze**, supervisors: **Prof. Dr. Benjamin Buchwitz** & **Hannah Behrens**

The task consists of implementing *the White Test for Heteroskedasticity* as well as estimating the *heteroscedasticity-consistent covariance matrix* for a linear regression model - introduced by White (1980) - in R (R Core Team 2021). Therefore, the following steps are necessary:

1. The theory has to be presented and explained since it is the basis for the following implementation. Consequently, hypotheses, equations and so on should be introduced and explained in your own words. Whenever you use literature, cite it! For an introduction to the *White Test* and the estimation of the *heteroscedasticity-consistent covariance matrix* see Wooldridge (2013) and Greene (2012). The data sets which are mentioned by Wooldridge (2013) are available in the R-package `wooldridge` (Shea 2021).

2. Implement *the White Test for Heteroskedasticity* and estimate the *heteroscedasticity-consistent covariance matrix* for a linear regression model. Your created function(s) and your main program have to be documented as it was shown in the lecture and exercise about *Software Development* so that the single implementation steps are easy to understand.
   Your function for the *White test* should return at least the following values:

   - the value of the computed test statistic,
   - the corresponding number(s) of degrees of freedom,
   - the corresponding p-value,

   and it should return text which states

   - what the alternative hypothesis is and
   - whether the null hypothesis has to be rejected or not.

   Your fuction for the estimation of the *heteroscedasticity-consistent covariance matrix* should return a matrix in which the *heteroscedasticity-consistent covariances* are listed in the form of (in your output `x1` and `x2` should be substitued by the concrete names of the variables):

   ```
   ##           Intercept      x1      x2
   ## Intercept    7.4159 -0.1004  0.0122
   ## x1          -0.1004  0.0014 -0.0002
   ## x2           0.0122 -0.0002  0.0001
   ```

3. In order to check whether your code works correctly you should compare your computed results with an example extracted from the literature and/or with (an) already existing function(s) like `sandwich::vcovHC()` with `type = HC` or `type = HC0` (Zeileis, Köll, and Graham 2020, 2004).

Hint: In order to work with matrices in R the following commands might be helpful:

- `%*%`: for matrix multiplication and
- `base::solve()` (R Core Team 2021): for the inverse of a matrix.

Task 2: Creation of a `shiny` web application in order to visualize the relationship between the density, distribution and quantile function of the normal distribution

student: **Robin Langer**, supervisor: **Hannah Behrens**

The task consists of visualizing the relationship between the density, distribution and quantile function of the normal distribution in form of a web application based on the R package `shiny` (Chang et al. 2021). Therefore, the following steps are necessary:

1. The first part consists of presenting and explaining the theory by defining *at least* the density, distribution and quantile function in general and in the context of the normal distribution since these functions and this distribution are the basis for the following implementation. Consequently, equations and so on should be introduced and explained in your own words. Whenever you use literature, cite it!
   For an introduction to the mentioned functions and the normal distribution see for example Fahrmeir et al. (2016) and Weigand (2009). For information about `shiny` web apps see the slides (https://bchwtz.github.io/bchwtz-cswr/web-apps.html) and (https://bchwtz.github.io/bchwtz-cswr/exercises-9.html).

2. Visualize the relationships by creating a `shiny` web app (Chang et al. 2021). Document your code as it was shown in the lecture and exercise about *Software Development* so that the single implementation steps are easy to understand.
   Your `shiny` web app should *at least* consist of the following characteristics and elements:

   - The user can choose some x values, a mean (μ) and a standard deviation (σ). Based on these values, a density, distribution and a quantile function of the normal distribution will be plotted.
   - You can visualize the relations between the density, distribution and quantile function for example by allowing that the user can choose an x value from the plotted density function and the corresponding values in the plotted distribution and quantile function will be marked (e.g. colored) in order to read off the interesting values. Additionally, print the interesting values.
   - Create interactive plots by making use of the R function `plotly::ggplotly()` (Sievert 2020) that is based on the R package `ggplot2` (Wickham 2016).

3. Based on your `shiny` web app, explain the relationships between the density, distribution and quantile function of the normal distribution.

Task 3: Reimplementation of the `optim()`-function with the `Nelder-Mead`-algorithm as chosen method

student: **Ferdinand Moritz Schreiber**, supervisor: **Prof. Dr. Benjamin Buchwitz**

The task consists of an implementation of the Nelder-Mead Simplex Algorithm used for non-linear optimization (Nelder and Mead 1965) and an exemplary application of that implementation. Therefore, the following steps are necessary:

1. The Nelder-Mead Algorithm is one of the non-linear optimization algorithms that does not require differentiation.The theory and intuition has to be presented and explained (e.g. Mirroring, Expansion, Contraction, Compression) since it is the basis for the following implementation. Consequently, starting from the simplex the algorithmic procedure should be introduced, explained and visualised in your own words. Whenever you use literature, cite it! For an introduction to the *Nelder-Mead Algorithm* see Nelder and Mead (1965) (and the correction that eradicates the errors in that first paper).

2. Implement *Nelder-Mead Algorithm* and use your implementation to show the functionality using an example of your choice (preferably some form of paramter estimation for an non-linear model). Your created function(s) and your main program have to be documented as it was shown in the lecture and exercise about *Software Development* so that the single implementation steps are easy to understand. Your function for the *Nelder-Mead Algorithm* should return at least the values returned by `optim()`.

3. In order to check whether your code works correctly you should compare your computed results with the `optim()` function. Explain the differences you observe.

References

Buchwitz, Benjamin. 2022. *Fhswf: Fhswf.*

Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2021. *Shiny: Web Application Framework for r.* https://CRAN.R-project.org/package=shiny.

Fahrmeir, L., C. Heumann, R. Künstler, I. Pigeot, and G. Tutz. 2016. *Statistik. Der Weg Zur Datenanalyse.* 8th ed. Springer-Lehrbuch. Berlin: Springer. https://doi.org/10.1007/978-3-662-50372-0.

Greene, William H. 2012. *Econometric Analysis.* 7th ed. Pearson.

Nelder, J. A., and R. Mead. 1965. "A Simplex Method for Function Minimization." *The Computer Journal* 7 (4): 308–13. https://doi.org/10.1093/comjnl/7.4.308.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Shea, Justin M. 2021. *Wooldridge: 115 Data Sets from "Introductory Econometrics: A Modern Approach, 7e" by Jeffrey m. Wooldridge.* https://CRAN.R-project.org/package=wooldridge.

Sievert, Carson. 2020. *Interactive Web-Based Data Visualization with r, Plotly, and Shiny.* Chapman; Hall/CRC. https://plotly-r.com.

Weigand, C. 2009. *Statistik Mit Und Ohne Zufall. Eine Anwendungsorientierte Einführung.* 2nd ed. Physica-Lehrbuch. Physica. https://doi.org/10.1007/978-3-7908-2347-9.

White, Halbert. 1980. "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity" 48 (4): 817–38. https://doi.org/https://doi.org/10.2307/1912934.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wooldridge, Jeffrey M. 2013. *Introductory Econometrics. A Modern Approach.* 5th ed. USA: South-Western. https://economics.ut.ac.ir/documents/3030266/14100645/Jeffrey/_M./_Wooldridge_Introductory/_Econometrics/_A/_Modern/_Approach/___2012.pdf.

Zeileis, A., S. Köll, and N. Graham. 2004. "Econometric Computing with HC and HAC Covariance Matrix Estimators." *Journal of Statistical Software* 11 (10): 1–17.

———. 2020. "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in r." *Journal of Statistical Software* 95 (1): 1–36.