



Introduction to econometrics stock pdf

No Frames Version Student Resources Site Navigation Navigation for Student Resources Top Reviews Latest Top Reviews Latest Top Reviews This book is the Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see the notes: click on the top right corner of the page chaired by Econometrics Department of Business Administration and Economics University of Duisburg-Essen. We have regularly found that the large number of students, especially in our introductory university econometric classes we teach at the University of Duisburg-Essen. We have regularly found that the large number of students, especially in our introductory university econometric backgrounds, it is an integral part of the curricult on understand the benefits of R students are useful, for example, for understanding and validating rooms that are generally not easy to understand simply by formulas. As applied econometrics and complementary classical literature like the Venables and Smith books (2010), we thought it would be better to provide an integral part of the curriculum that content of the well-rece This material is an introduction to Econometrics by Stock and Vatson (2015). It is an interactive script in the style of a reproducible reproduced in R, variable, functions, terregreen to a reference to the R code. This includes commands, variable, sense click on the gray background indicates the R code, which R and energe studies content of the weight of a new explanded by to cannot the terregree to a reference to the R code. Which R and energies and constructions, but mostly large in the form of a new one. Encounter to the weight of a nergeo classes weight and allows students not only to learn weight and encounter of the weight and watto and the sense constant with text on the gray background indicates the R code, which you can also a terroduced in R. Now and the sense constant weight and the sense con

get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see other people's comments by clicking on R (R Core Team 2020) freely available statistical programming language and software environment in the upper right corner of the page. By the time we wrote the first drafts of the project, more than 11,000 add-ons (many of which provide cutting-edge methods) had been made available on the Comprehensive R Archive Network (CRAN), an extensive network of FTP servers around the world that store identical and up-to-date versions and documentation of R code. R dominates other (commercial) software statistical computing statistics used in most research fields. The advantage of being freely available, open source, and a large and ever-growing user community that contributes to CRAN makes R increasingly attractive to empirical economists and econometrics alike. The striking advantage of using R is that it allows students in econometries to document their analysis in detail so that they can be easily updated and expanded. This allows you to reuse the code for similar applications with different data. Furthermore, R programs are fully reproducible, which makes it clear that others understand and validate the results. In recent years, R has become an integral part of the curriculum of econometric classes taught at duisburg-Essen University. In a sense, cod learning is similar to learning a foreign language, and continuous practice is essential for learning success. Needless to say, presenting bare R-code slides does not encourage students to engage in a hands-on experience of their own. Therefore, R is crucial. As for the accompanying literature, some excellent books that deal with R and its applications in econometrics, e.g. Kleiber and Zeileis (2008). However, such resources go slightly beyond the scope of university students' economics, which are barely familiar with econometric methods and have little experience of programming at all. As a result, we started compiling a collection of reproducible reports for use in the class. These reports provide guidance on how to implement the selected applications from the Introduction to Econometrics (Stock and Watson 2015) textbook, which serves as the basis for the presentation and related tutorials. This process has been significantly facilitated by knitr (Xie 2020b) and R markdown (Allaire and contracting. 2020). In this context, both R packages provide powerful features for dynamic report generation that allow you to combine clear text, LaTeX, R-code, and output in different formats, including PDF and HTML. In addition, the writing and dissemination of reproducible reports for use in academy has been greatly enriched by the bookdown is built on top of the R markdown and allows you to create such attractive HTML pages, among other things. Using R for introductory econometrics (Heiss 2016)1 and this powerful toolkit is written for the empirical companion stock and Watson are doing a great job explaining the intuition and theory of this econometrics, and in any case better than we can in another introductory textbook! Introductory textbook! Introductory textbook! Introductory textbook! Introductory textbook! Introductory textbook as an interactive script in the style of a reproducible research report, designed to provide students with a platform-independent e-learning agreement seamlessly intertwined with theoretical basics and empirical skills in academic ecumetry. Of course, the focus is on empirical applications R. We omit derivatives and evidence wherever we can. Our goal is not only to teach students how results from case studies can be reproduced in R, but we also intend to strengthen their ability to newly acquired skills in other empirical applications – directly through the Introduction to Econometrics i R. To achieve this, each chapter contains interactive R programming exercises. These exercises are used as add-ons for code pieces to display how previously discussed techniques can be implemented using R. These DataCamp lightwidget are created and supported by an R session that is maintained on DataCamp servers. You can play around with the example described below. As you can see above, the widget consists of two sheets. R mimics the . R-file, a file format that is often used to store R-code. Lines that start with #are annoted, meaning they are not recognized as code. Also, script. R works as an exercise sheet where you can describe the solution you come up with. When you press the button Run, execute the code, submit correctness tests and you will be notified that the approach is correct, you will receive feedback suggesting improvements or tips. The other tab, the R console that can try out exercises before submitting. Of course, you can submit (almost any) R code and use the console to play and explore. Simply type the command and press Enter on the keyboard. Looking over the widget, you will notice that there is a > in the right panel (the console). This symbol in this book. The output generated by the R code is performed with a #> note. Most often, the R code is displayed along with the output generated in the chunks of code. For example, consider the following line of code that appears in the following line of code that appears in the following chunk. It says R to calculate the number of packages available in CRAN. The code array is followed by the generated output. # check CRAN nrow(available.packages(repos =)) #> [1] 16272 Each block of code is equipped with a button on the right outer side that copies the code to the Clipboard. This makes it convenient to work with larger code segments in your version of R/RStudio or the widgets presented throughout the book. In the widget above, click R to type nrow(available.packages(repos =)) (command for the above chunk of code) and execute enter on the keyboard. 2 Note that some lines of the widget are commented on outside, asking you to assign a numeric value to a variable, and then print the contents of the variable on the console. You can specify the solution approach script. R and press the button Run in order to get the feedback described above. In case you don't know how to solve this pattern of exercise (don't panic that's probably why you're reading this), click the Tip for you for some advice. If you still can't find a solution, click Solution to click another page, Solution.R, which contains a sample solution.R presents what we consider understandable and idiomatic. Allaire, JJ, Yihui Xie, Jonathan McPherson, Luraschi, Kevin Ushey, Aron Atkins, Atkins, Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2020 rmarkdown: Dynamic documents for R (version 2.3). rmarkdown I'm going to have to go. Heiss, Florian. 2016. Use of R for introductory econometrics with I R. Springer. R Core team. 2020. R: Language and environment of statistical computing. Vienna, Austria: R Statistical Computing Foundation. Https://www.R-project.org/ Stock, J.H., and M.W. Watson. 2015. Introduction to Econometrics, Third Update, Global Edition. Pearson Education Ltd. Page 3 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also view other people's notes by clicking on the book in the upper-right corner of the page: #> * * -i!i-> setting value #> r version r version 4.0.2 (2020-06-22) #> os macOS Catalina 10.15.4 #> system x86_64, darwin19.5.0 #> ui unknown #> language (EN) #> en_US. UTF-8 #> ctype en_US. UTF-8 #> tz Europe/Berlin #> date 2020-09-15 #> #> — Packages – - #> package * version date lib source #> abind 1.4-5 2016-07-21 [1] CRAN (R 4.0.2) #> AER 1.2-9 2020-02-06 [1] CRAN (R 4.0.0) #> askpass 1.1 2019-01-13 [1] CRAN (R 4.0.0) #> base64enc 0.1-3 2015-07-28 [1] CRAN (R 4.0.0) #> bibtex 0.4.2.2 2020-01-02 [1] CRAN (R 4.0.0) #> bi 1.0-6 2013-08-17 [1] CRAN (R 4.0.0) #> blob 1.2.1 2020-01-20 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-22 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> blob 1.2.1 2020-01-20 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-22 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-22 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-22 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-22 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-22 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-22 [1] CRAN (R 4.0.0) #> car 3.0-8 2020-05-21 [1] 4.0.0) #> cellranger 1.1.0 2016-07-27 [1] CRAN (R 4.0.0) #> cli 2.0.2 2020-02-28 [1] CRAN (R 4.0.0) #> clipr 0.7.0 2019-07-23 [1] CRAN (R 4.0.0) #> conquer 1.0.1 2020-05-06 [1] CRAN (R 4.0.2) #> crayon 1.3.4 2017-09-16 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2020-07 -06 [1] CRAN (R 4.0.2) #> cubature 2.0.4.1 2019-03-18 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2019-03-18 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2019-07-23 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2019-03-18 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2020-07 -06 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2019-03-18 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2020-07 -06 [1] CRAN (R 4.0.0) #> cubature 2.0.4.1 2019-03-18 [1] CRA 4.3 2019-12-02 [1] CRAN (R 4.0.0) #> data.table 1.12.8 2019-12-09 [1] (R 4.0.0) #> dbplyr 1.4.4 2020-05-27 [1] CRAN (R 4.0.0) #&g 0.3.1 2020-05-15 [1] CRAN (R 4.0.0) #> farch 3042.83.2 2020-03-07 [1] CRAN (R 4.0.2) #> fasit 0.4.1 2020-01-08 [1] CRAN (R 4.0.2) #> fasit 0.4.1 2020-03-07 [1] CRAN (R 4.0.2) #> fasit 0.4.1 2020-01-08 [1] CRAN (R 4.0.2) #> fasit 0.4.1 2020-03-07 [1] CRAN (R 4.0.2) #> fasit 0.4.1 2020-01-08 [1] CRAN (R
4.0.2) #> fasit 0.4.1 2020-01-08 [1] CRAN (R 4.0.2) #> fasit 0.4.1 2020-03-07 [1] CRAN (R 4.0.2) #> fasit 0.4.1 2020-01-08 [1] CRAN 01 [1] CRAN (R 4.0.0) #> foreign 0.8-80 2020-05-24 [2] CRAN (R 4.0.2) #> generics 0.0.2 2018-05-03 [1] CRAN (R 4.0.0) #> generics 0.0.2 2018-01-01 [1] CRAN (R 4.0.0) #> generics 0.0.2 2018-01-01-01 [1] CRAN (R 4.0.0) #> generics 0.0.2 2018-01-01-#> gss 2.2-2 2020-05-26 [1] CRAN (R 4.0.2) #> httr 1.4.2 2020-07-20 [1] CRAN (R 4.0.0) #> haven 2.3.1 2020-06-01 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-07-20 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-07-20 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-07-20 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-01 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-01 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-01 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-01 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-01 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-07-20 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-07-20 [1] CRAN (R 4.0.0) #> htmltools 0.5.0 2020-06-16 CRAN (R 4.0.0) #> itewrpkg 0.0.0.9000 2020-07-28 [1] Github (mca91/itewrpkg@bf5448c) #> isonlite 1.7.0 2020-06-25 [1] CRAN (R 4.0.0) #> knitr 1.29 2020-06-23 [1] CRAN (R 4.0.0) #> knitr 1.29 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 4.0.2) #> lifecycle 0.2.0 2020-06-23 [1] CRAN (R 4.0.0) #> lattice 0.20-41 2020-04-02 [2] CRAN (R 03-06 [1] CRAN (R 4.0.0) #> Ime4 1.1-23 2020-04-07 [1] CRAN (R 4.0.0) #> Intest 0.9-37 2019-04-30 [1] CRAN (R 4.0.0) #> Intest 0.9-37 201 4.0.0) #> MASS 7.3-51.6 2020-04-26 [2] CRAN (R 4.0.2) #> Matrix 1.2-18 2019-11-27 [2] CRAN (R 4.0.2) #> matrixStats 0.56.0 2020-03-13 [1] CRAN (R 4.0.2) #> matrixStats 0.56.0 2020-03-13 [1] CRAN (R 4.0.2) #> maxLik 1.3-8 2020-01-10 [1] CRAN (R 4.0.0) #> matrixStats 0.56.0 2020-03-13 [1] CRAN (R 4.0.2) #> maxLik 1.3-8 2020-01-10 [1] CRAN (R 4.0.2) #> matrixStats 0.56.0 2020-03-13 [1] CRAN (R 4.0.2) #> matrixStats 0.56.0 2020-03-13 [1] CRAN (R 4.0.2) #> matrixStats 0.56.0 2020-03-13 [1] CRAN (R 4.0.2) #> maxLik 1.3-8 2020-01-10 [1] CRAN (R 4.0.2) #> matrixStats 0.56.0 2020-03-13 [1] CRAN (R 4.0.2) #> matrixStats 0.56.0 2020-03 minqa 1.2.4 2014-10-09 [1] CRAN (R 4.0.0) #> miscTools 0.6-26 2019-12-08 [1] CRAN (R 4.0.0) #> modelr 0.1.8 2020-05-19 [1] CRAN (R 4.0.0) #> modelr 0.1.8 2020-05-19 [1] CRAN (R 4.0.0) #> modelr 0.1.8 2020-05-24 [2] CRAN (R 4.0.2) #> miscTools 0.6-26 2019-12-08 [1] CRAN (R 4.0.2) #> modelr 0.1.8 2020-05-19 [1] CRAN (R 4.0.0) #> modelr 0.1.8 2020-05-24 [2] CRAN (R 4.0.0) #> miscTools 0.6-26 2019-12-08 [1] CRAN (R 4.0.0) #> modelr 0.1.8 2020-05-19 [1 2020-04-26 [2] CRAN (R 4.0.2) #> np 0.60-10 2020-02-06 CRAN (R 4.0.2) #> openssl 1.4.2 2020-06-27 [1] CRAN (R 4.0.2) #> openssl 1.4.2 2020-06-27 [1] CRAN (R 4.0.2) #> openxlsx 4.1.5 2020-05-06 [1] C 4.0.2) #> pkgconfig 2.0.3 2019-09-22 [1] CRAN (R 4.0.0) #> pkgload 1.1.0 2020-05-29 [1] CRAN (R 4.0.0) #> prettyunits 1.1.1 2020-01-24 [1] CRAN (R 4.0.0) #> prettyunits 1.1.1 2020-01-24 [1] CRAN (R 4.0.0) #> processx 3.4.3 2020-07-05 [1] CRAN (R 4.0.2) #> progress 1.2.2 2019-05-16 [1] CRAN (R 4.0.2) #> prettyunits 1.1.1 2020-01-24 [1] CRAN (R 4.0.0) 1.3.3 2020-05-08 [1] CRAN (R 4.0.0) #> purr 0.3.4 2020-04-17 [1] CRAN (R 4.0.0) #> quadprog 1.5-8 2019-11-20 [1] CRAN (R 4.0.2) #> quantreg 5.61 2020-07-09 [1] CRAN (R 4.0.2) #> quantr 06 [1] CRAN (R 4.0.2) #> RcppArmadillo 0.9.900.3.0 2020-09-03 [1] CRAN (R 4.0.2) #> RcppEigen 0.3.3.7.0 2019-11-16 [1] CRAN (R 4.0.0) #> rdd 0.57 2016-03-14 [1] CRAN (R 4.0.0) #> rdd 0.57 2 05 [1] CRAN (R 4.0.2) #> readx 1.3.1 2018-12-21 [1] CRAN (R 4.0.0) #> readx 1.3.1 2019-03-13 [1] CRAN (R 4.0.0) #> rematch 1.0.1 2016-04-21 [1] CRAN (R 4.0.0) #> renatch 1.0.1 2016-04-21 [1] CRAN (R 4.0.0) #> rematch 1.0.1 2016-04-21 [1] CRAN (R 4.0.0) #> rematch 1.0.1 2018-03-13 [1] CRAN (R 4.0.0) #> rematch 1.0.1 2018-04-21 [1] CRAN (R 4.0.0) #> rematch 1.0.1 2018-04rprojroot 1.3-2 2018-01-03 [1] CRAN (R 4.0.0) #> rstudioapi 0.11 2020-02-07 [1] CRAN (R 4.0.0) #> rvest 0.3.6 2020-07-25 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-20 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-20 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-20 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.0) #> scales 1.1.1 2020-05-20 [1] CRAN 13 [1] CRAN (R 4.0.2) #> spatial 7.3-12 2020-04-26 [2] CRAN (R 4.0.2) #> stabledist 0.7-1 2016-09-12 [1] CRAN (R 4.0.2) #> stargazer 5.2.2 2018-05-30 [1] CRAN (R 4.0 CRAN (R 4.0.2) #> survival 3.2-3 2020-06-13 [2] CRAN (R 4.0.2) #> tidyr 1.1.0 2020-07-23 [1] CRAN (R 4.0.2) #> tidyr 1.1.0 2020-07-20 [1] CRAN (R 4.0.0) #> tidyr 1.1.0 2020-05-11 [1] CRAN (R 4.0.0) #> tidyr 1.1.0 2020-05-20 [1] CRAN (R 4.0.0) #> tidyr timeDate 3043.102 2018-02-21 [1] CRAN (R 4.0.0) #> timeSeries 3062.100 2020-01-24 [1] CRAN (R 4.0.2) #> tinytex 0.25 2020-07-24 [1] CRAN (R 4.0.2) #> tinytex 0.25 2020-07-24 [1] CRAN (R 4.0.2) #> 1.3-0 2016-09-06 [1] CRAN (R 4.0.2) #> tinytex 0.25 2020-07-24 [1] CRAN 4.0.2) viridisLite 0.3.0 2018-02-01 [1] CRAN (R 4.0.0) #> xml2 1.3.2 2020-04-23 [1] CRAN (R 4.0.0) #> xith 0.16 2020-07-24 [1] CRAN (R 4.0.0) #> xml2 1.3.2 2020-04-20 [1] CRAN (R 4.0.0) #> xml2 1.3.2 2020-04-20 [1] CRAN (R 4.0.0) #> xml2 1.3.2 2020-04-23 [1] CRAN (R 4.0.0) #> xml2 1.3.2 2020-04-20 [1] CRAN (R 4.0.0) #> xml2 1.3.2 2020-04-23 [1] CRAN (R 4.0.0) #> xml2 1.3.2 2 CRAN (R 4.0.0) #> zoo 1.8-8 2020-05-02 [1] CRAN (R 4.0.0) #> #> [1] /usr/local/lib/R/4.0/site-library #> [2] /usr/local/Cellar/r/4.0.2_1/lib/r/library Page 4 Ez a könyv nyílt értékelés alatt áll. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see the notes of others: click on the upper right corner of the page Figure 1.1: RStudio: the four panes As mentioned above, this book is not intended to be an introduction to the R, but rather a guide on how to use the capabilities of applications commonly occurring in university econetics. Those with a basic knowledge of R programming will feel comfortable with the 2nd Programming Programme. However, this section is for those who have not previously worked with R or rstudio. At least you know how to create objects and call functions, you can skip them. If you want to upgrade your skills or get a feeling of how to work with RStudio, read on. First, start RStudio and open a new R script by selecting File, New File, R. In the editing pane, type and click Run in the upper-right corner of the editor. The code line is sent to the console and the result must be displayed directly below it. As you can see, R works like a calculator. You can perform all arithmetic calculator. You can see, R works like a calculator. than that. You can work with variables or, more generally, with objects. Define objects using the assignment operator <-. To create a variable called x that contains x <- 10 of type 10, click Run again. The new variable should have appear in the context pane in the upper-right corner. However, the console did not show results because our line of code did not contain a call that generated output. When you now type x in the console and push it back, it asks R to show the value of x and the correct value must be printed on the console. you can easily create a vector taknat of \(1 & gt; & lt;\) length by using the c() (c) function: stitching or combining). Now let's just remember that quotation marks, otherwise they will be analyzed as object names. hello & lt;c(Hello, World) Here we created a vector length of 2 words containing Hello and the world. Do not forget to save the script! To do this, select File, Save. You have seen the c() function calls look the same: the function name is always followed by round parentheses. Sometimes parentheses include arguments. Here are two simple examples. # creates the vector z z <- seq(from = 1 to = 5, by = 1) # calculates the average of the enries in the middle of the z # > [1] 3 In the first row we use the function called seq() to create exactly the same vector as we did in the previous section, in which z. The function must be taken for granted by the arguments from which and the next
self-explanatory. The mediocre() function calculates the arithmetic mean of the x argument. Since we pass the vector z as the argument x, the result is 3! If you are not sure how the arguments for seq() work. Then write ?seq on the console. The hitting back of the documentation page to the function appears in the lower right pane of RStudio. There, the arguments section contains the information we are looking for. At the bottom of almost every Help page, you'll find examples of how to use the appropriate functions. This is very useful for beginners and we recommend taking care of them. Of course, all of the commands shown above also work in interactive widgets in the book. You may try them below. Page 5 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can also see other people's annotations by clicking on the ra in the upper right corner of the page This chapter reviews some of the basic concepts of probability theory and shows you how to apply them in R. Most of the statistical functions of the R fund are collected in the statistics package. It provides simple features that charge descriptive rates and facilitate calculations involving different probability distributions. It also includes more sophisticated routines that, for example, allow the user to estimate a number of models based on the same data or help perform extensive simulation tests. statistics are part of the R base distribution, which means that they are installed by default, so you don't need to run install.packages(stats) or library(stats). Simply fold the directory(help = stats) on the console to view the documentation and a complete list of all the functions collected in the statistics. Most packages provide documentation that can be viewed in RStudio. Documentation can be referenced in ? operator, e.g.pl. executing ?stats in the documentation of the statistics package will appear in the help tab in the lower right pane. In the following, we will focus on the probability distributions handled by R and show you how to use the appropriate functions handled by R and show you how to use the appropriate functions handled by R and show you how to use the appropriate functions handled by R and show you how to use the appropriate functions handled by R and show you how to use the appropriate functions handled by R and show you how to use the appropriate functions handled by R to solve simple problems. This will update some of the basic concepts of probability theory. Among other things, you will learn how to draw random numbers, how to calculate density, probabilities, quantiatives and the like. As we will see, it is very convenient to rely on these routines. Page 6 This book is the Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see other people's annotations by clicking on the upper-right corner of the basic concepts of probability theory. Mutually exclusive means that only one possible outcome can be observed. The probability of the result is referred to as the proportion of the result occurring in the long term, i.e. if the experiment is a subset of a plot that consists of one or more results. These ideas are consistent with the concept of random variable, which is a numerical summary of random results. Random variables can be discreet or continuous. Separate random variable can add a continuum of possible values. A typical example of a separate random variable \(D\) is the result of a cube roll: from a random experiment point of view, it is nothing more than randomly selecting a pattern of size \(1\) from a series of numbers that have mutually exclusive results. Here you can think of a plot \(\1,2,3,4,5,6\}\) and a number of different events, such as the observed result between \(2\) and \(5\). A basic function that takes random samples from a specific set of elements is the function sample(), see sample. We can use it to simulate the random outcome of a dice roll. Let's throw the dice. The probability distribution of a separate random variable is a list of all possible values and probability that the random variable is a list of all possible value. For dice roll, probability distribution and specified number of times. # creates the probability throat probability & lt;-rep(1/6, 6) # plot of probability, xlab = results, main = probability, xlab = results, main = probability distribution) The cumulative probability distribution) The cumulative probability distribution). # Create a vector of cumulative probabilities cum_probability <- cumsum(probability) # representation of probability plot(cum_probability, xlab = results, main = Cumulative probability distribution) A set of elements from which the sample() produces results only from numbers. You can even simulate coin toss results \(H\) (head) and \(T\) (tail). Sample(c(H, T), 1) # > [1] H The result of a single coin toss is a Bernoulli distributed random variable, i.e. a variable with two possible different results. Imagine that about tossing a coin \(10\) times in a row and wonder how likely it is to end up with a \(5\) time heads. This is a typical example of what we call the Bernoull experiment, because it consists of \(n=10\) Bernoulli experiments, which are independent of each other and we are interested in the likelihood of observing \(k=5\) successes \(H) in each trial \(p=0.5\) (assuming a fair coin) in every trial. Keep in mind that the number of successes \(k\) in a Bernoulli experiment follows the binomial distribution. This \[k \sim B(n,p.\] The probability of \(B(n,p)) successes of the experiment \(B(n,p)\) is \[f(k)=P(k)=\begin{pmatrix}\\ k \end{pmatrix}\\ k \end{pmatrix}\\ k \end{pmatrix}\. In R, you can solve problems like the above by using the dbinom() function, which calculates the probability of binomial distribution based on \(P(k\vert n, p)\) based on x (\(k\)), size (\(n\)) and prob (\(p\)) parameters, see ?dbinom. Calculate \(P(k=5\\vert n = 10, p = 0.5)\) (this short one \(P(k=5)\) dbinom(x = 5, size = 10, prob = 0.5) #> [1] 0.2460938 We concluded that \(P(p(k=5)\), probability of monitoring head \(k=5\) time \(n=10\) is \(24.6\%\). Now let's say you're interested in $-(P(4 \ | eq k \ | eq 7)\)$, that is, the probability of monitoring \(4\), \(5\), \(6\) or \(7\) successes \ (B(10, 0.5)). This can be calculated by specifying a vector as an x argument in our dino call() and sum(). # compute P(4 & lt;= 7) is the dbinom(), using the distribution is a different approach. binomial distribution $[P(4 \leq 7) = P(k \leq 7) - P(k \leq 3).]$ # calculation $P(4 \leq 5) + k \leq 5$ pbinom() pbinom(size = 10, prob = 0,5, q = 7) - pbinom(size = 10, prob = 0,5, q = 3) #> [1] 0,7734375 The probability distribution of the discrete random variable is the list of all possible outcomes and their probabilities. In the coin toss example, \(11\) has possible outcomes for \(s\). # set up vector of possible results k <- 0:10 k #> [1] 0,7734375 The probability distribution of the discrete random variable is the list of all possible outcomes and their probabilities. In the coin toss example, \(11\) has possible outcomes for \(s\). # set up vector of possible results k <- 0:10 k #> [1] 0,7734375 The probability distribution of the discrete random variable is the list of all possible outcomes for \(s\). # set up vector of possible results k <- 0:10 k #> [1] 0,7734375 The probability distribution of the discrete random variable is the list of all possible outcomes for \(s\). probability distribution function of \(k\) you can: # assign probability &It;- dbinom(x = k, size = 10, prob = 0,5) # plot(x = k, y = probability, main = Probability, main = Probability distribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability, main = Probability &It;- binom(x = k, size = 10, prob = 0,5) # representation of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability astribution function of \(k\) by implementing the following piece of code: # calculation cumulative probability as the following piece of code: # calculation cumulative probability as the following piece of code: # calculation cumulative probability as the following piece of code: # calculation cumulative probability as the following piece of code: # calculation cumulative probability as the following piece of code: # calculation cumulative piece of code: # calculation cumulat cumulative probabilities on a plot(x = k, y = prob, main = Cumulative distribution function) The expected value of random variable, the expected value is calculated as a weighted average of possible results, on the basis of which the weights are related probabilities. This is official in Key Concept 2.1. Let's say the \$Y\$ random variable \$k\$1, \$y_1, \dots, \$y_k, where \$y_1\$ represents the first value, \$y_2\$ represents the probability of taking up \$\$Y \$y_1 is
\$p_\$1, the probability is that \$Y\$\$y_2\$ and so on. The expected value of \$Y\$, \$E(Y)\$ can be determined as follows: \$\$ $E(Y) = y_1 p_1 + y_2 p_2 + (cdots + y_k p_k = (b))$ ($d_1 = 1, d_2 = 2, dots, d_6 = 6$). Assuming real cubes, the value of $y_i \approx 1, d_1 = 1, d_2 = 2, dots, d_6 = 6$. Assuming real cubes, the value of $y_i \approx 1, d_1 = 1, d_2 = 2, dots, d_6 = 6$. (6)) results are all likely to be \(1/6\). Therefore, you can easily calculate the exact value of \(E(D)\) manually: \[E(D) = 1/6 \sum_{i=1}^6 d_i = 3.5 \) \(E(D)\) is simply the average of the natural numbers between \(1\) and \(6\) because all \(p_i) weight \(1/6\). This can be easily calculated by using the numeric vector arithmetic mean() function. # calculation of the average of natural numbers between \(1) and \(6) because all \(p_i) weight \(1/6\). This can be easily calculated by using the numeric vector arithmetic mean() function. # calculation of the average of natural numbers between \(1) and \(6) because all \(p_i) weight \(1/6\). This can be easily calculated by using the numeric vector arithmetic mean() function. # calculation of the average of natural numbers between \(1) and \(6) because all \(p_i) weight \(1/6\). order to reproduce the results of calculations containing random numbers, set.seed() is used to set R's random numbers. The sequences of random numbers generated by R are pseudo-random numbers, i.e. they are not really random, but approximate the properties of a series of random numbers. Since this approach is good enough for our goals we refer to pseudo-random number generated by pseudo-random numbers with random numbers. The PRNG in R works by performing certain operations at a deterministic value. This value is usually the previous number generated by PRNG. PrNG, however, has no previous value for the first time. The core is the first value in a series of numbers - it initializes the sequence. Each core value corresponds to a different set of values. In R, the core can be set using set.seed(). It's convenient for us: If we get the same core twice, we get the same sequence of numbers twice. Thus, setting a core before executing an R code that includes random numbers makes the result reproducible! Of course we also consider a much larger number of tests, \(1000\) entries for large vectors and skips (try) the remainder. Evening the numbers doesn't say much. Instead, calculate the sample average of the result is close to the expected value. (E(D)=3.5)). # set seed for reproducibility set.seed(1) # calculate the sample average of 10,000 cubes (sample(1:6, 10000, replace = T)) #> [1] 3.5138 The sample average of the result is close to the expected value. This result shall be tested in the test as 2.2. Other common measures are variance and standard deviation. Both are a measure of the dispersal of a random variable. \(Y\) candidate \(\sigma^2_Y = \text{Var}(Y) = E\left[((Y-\mu_y))^2 p_i \] The \(Y\) standard deviation \(\sigma_Y\), the square root of variance. The units of the standard deviation are the same as the units of (Y). The variance defined in key concept 2.2, which is a quantity of people, is not implemented as a function of R. Instead, we have functionvar() functions that calculate the sample variance $|s^2 Y| = \frac{1}{n} (y i - \frac{1}{n})^2$. Note that $(s^2 Y)$ differs from the so-called population variance of (Y), $\int \frac{1}{N} = \frac{1}^N (y i - \frac{1}{N})$ because it measures how the (n) observations in the sample are distributed around t rolling example. \(D\) has \[\[\text{Var}(D) = 1/6 \sum {i=1}^6 (d i - 3.5)^2 = 2.92 \], which is clearly different from the \(s^2\) result calculated by var(^ 2\). The sample variance is the estimated population variable takes up the continuum of possible values, we cannot use the concept of probability distribution as used for separate random variables. Instead, the probability distribution of a continuous random variable kinprotest distribution function (CDF) is defined in the same way as in a separate case. Therefore, the CDF of continuous random variables determines the probability that the random variable is less than or equal to a given value. For the sake of completeness, we present the information 2.1 and (b) where \(a < b) \[P(a \leq b) = \int a^b f Y(y) \mathrm{d}y. \] What else would have been that \(P(-\infty \leq Y \leq \infty) = 1\) and therefore \(\int_{-\infty} infty] f_Y(y) \mathrm{d}y = 1\). As for the separate case, the expected value of \(Y\) is the probabilities weighted average of the continuity, we use the sums instead of the integrating ones. The expected value of \(Y) is \[E(Y) = \mu_Y = \int y f_Y(y) \mathrm{d}y. \] Variance is the expected value of \(P(-\infty) is \[E(Y) = \mu_Y = \int y f_Y(y) \mathrm{d}y = 1\). As for the separate case, the expected value of \(P(-\infty) is \[E(Y) = \mu_Y = \int y f_Y(y) \mathrm{d}y = 1\). value of $((Y - \mathbb{Y}^2)$). That's right $[\frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x))$ integrates on the real line equal to (1). $[\frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x))$ integrates on the real line equal to (1). $[\frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1}]$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x) = \frac{y^2}{x^4}, x_{gt;1})$ Ure can analytically show that $(f_X(x)$ $| \t = \frac{1}^t | = \frac$ $2x^{-2} x^{-2} | x^$ \rvert_{x=1}^{\infty} \\ =& -3 \left(\lim_{t \rightarrow \infty} \rac{1}{t} - 1 \right) \\ =& 3 \end{}\]. However, this was tedious and, as we will see, some PDFs do not have an analytical approach, e.g. if the integrations do not (\text{Var}(X) = \frac{3}{4}\). However, this was tedious and, as we will see, some PDFs do not have an analytical approach, e.g. if the integrations do not have closed solutions. Fortunately, R allows us to easily find the above results. The device we use to do this is the integration of the function(). First, you must define functions f <- function(x) x * f(x) h <- function(x) x^2 * f(x) Then use the integrate() function and set the upper and lower limits of integration using (1) and ((infty)) using the upper and lower arguments. By default, integrate() prints the result along with an estimate of the approach error for the console. However, the result is not a numerical value that can be easily used to further calculate. In order to get only one numeric value from the integral, you need to use the \$ operator with the value. You can use the \$ operator to get names from a type of object. # calculation area integration(g, bottom = 1, top = Inf)\$value EX #> [1] 1.5 # Calculation Var(X) Integration of VarX <- (h, bottom = 1, top = Inf)\$value integration(g, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value
integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration of VarX <- (h, bottom = 1, top = Inf)\$value integration EX^2 VarX #> [1] 0.75 Although normal, chi-square, slides \(t) and \(F\) distributions are most common in econometric distributions. Therefore, we will discuss some basic R functions that allow calculations involving the density, probability, and quantiativeity of these distributions to be carried out. Each probability distribution handled by R has four basic functions, the name of which consists of a prefix and a root name. Take the normal distribution as an example. The root name of all four function p is probability function p is probability - cumulative distribution function p is probability function function function p is probability function p is probability function function p is probability function f random - Thus, for normal distribution, R functions are dnorm(), pnorm(), and rnorm(), and rnorm(), and rnorm(). The most likely probability distribution here taking into account the normal distribution. This is not least due to the special role of the standard normal distribution and the Central Limit item, which will soon have to be addressed. The normal distribution is symmetrical and bell-shaped. The normal distribution here taking into account the normal distribution is symmetrical and bell-shaped. normal distribution is characterized by \(\mu\) and \(\sigma=1\). Normal normal distribution , which are succinctly expressed in \(\mu+2)\). Standard distribution pdf file \\begin{align} f(x) = \frac{1}{\sigma^2}\). Standard distribution \(\mu+0) and \(\sigma+1) and standard standard PDF is usually marked \(\phi\) and the standard CDF is \(\Phi\). Therefore, V \phi(c) = \Phi'(c) \ Phi(c) = P(Z \leq c) \ Z \sim \mathcal{N}(0,1). Note: X is divided by X Y as X y. In R we can comfortably obtain the normal distribution density dnorm(). Draw a path() along with the dnorm() path. # draw a plot of land on the N(0.1) PDF curve(dnorm(x), xlim = c(-3.5, 3.5), ylab = Density, main = Standard normal density function) We can obtain the density in different positions passing vector dnorm(). # calculation density x =-1,96, x=0 and x=1.96 dnorm(x = c(-1.96, 0.96, 1.96)) #> [1] 0.05844094 0.39894228 0.05844094 Similar to pdf, we can plot the standard CDF using curve(). We could use dnorm () for this, but it's more convenient to rely on pnorm(). # plot the standard normal CDF curve(pnorm(x), xlim = c(-3,5, 3,5), ylab = probability, main = Standard normal variance. Let's say we're interested \(P(Z \leq 1.337)\). Some continuous random variables \(Z\) with \([-\infty,\infty]\) density \(g(x)\) must be defined (G(x)), the anti-derivative of (g(x)), the anti-derivative of (g(x)) be [P(Z | eq 1.337) = G(1.337) =problems using a numerical method. To do this, you must first define the function whose integration you want to calculate as an r-function. In our example, f is the normal PDF as R function (x) { 1/(sqrt(2 * pi)) * exp(-0.5 * x^2) } Let verify that this function calculates the normal density by passing a vector, # define the vector reals guants & lt:-c(-1.96, 0, 1.96) # calculation density f(guantums) #> [1] TRUE The results produced by f() are indeed equivalent to the results given by dnormal(then call integrate() f() and give the arguments for upper and lower, lower and upper bounds of integration. # integrate f() probability of pnorm() pnorm(1.337) #> [1] 0.9093887 Using the result of the same approach integrate(). Let's discuss a few other examples: A commonly known result is that \(95\%\) has a normal normal probability mass in the \([-1.96, 1.96]\) interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\) integrate(). Let's discuss a few other examples: A commonly known result is that \(95\%\) has a normal normal probability mass in the \([-1.96, 1.96]\) interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\) interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm this by calculating \[P(-1.96, 1.96]\] interval, i.e. approximately \(2\) from the average. You can easily confirm the average. You can easil \leq -1.96) \] due to standard normal PDF symmetry. Thanks to R, we can leave the table with the standard Stan convenient to first standardize it, as shown in Key Concept 2.4. Suppose (Y) is usually average ((uu)) and variance (s_1) and more (c_1) and (c_2) two numbers in which $((c_1 \& t; c_2))$ and more $(d_1 = (c_1 - t_1))$ $(d_2) = (c_2 - m)/sigma).$ Then $(begin{align}) = 1 - Phi(d_2) = Phi(d_2) =$ If you are interested in \(P(3 \leq Y \leq 4)\) you can use pnorm() and set the mean and/or standard deviation to differ from \(\mu=0\) and \(\sigma = 1\) by specifying the arguments medium and sd. Warning: the argument sd requires scattering, not variance! pnorm(4, mean = 5; sd = 5) - pnorm(3, mean = 5, sd = 5) + && argument sd requires scattering, not variance! pnorm(4, mean = 5; sd = 5) + && argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering, not variance! pnorm(4, mean = 5; sd = 5) + && argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) by specifying the argument sd requires scattering and \(\sigma = 1\) variable setting is normal distribution. \[begin{align} \begin{align} \sigma X} \right)\2 - 2\rho {XY}^2} \\ \cdot & amp; \, \exp \left(\frac{x-\mu X}\sigma X} \right)\2 - 2\rho {XY}^2} \\ \cdot & amp; \, \exp \left(\frac{x-\mu X}\sigma X} \right)^2 - 2\rho {XY}^2} \\ \cdot & amp; \, \exp \left(\frac{x-\mu X}\sigma X} \right)\2 - 2\rho {XY}^2} \\ \cdot & amp; \, \exp \left(\frac{x-\mu X}\sigma X} \right)\2 - 2\rho {XY}^2} \\ \cdot & amp; \, \exp \left(\frac{x-\mu X}\sigma X} \right)\2 - 2\rho {XY}^2} \\ \cdot & amp; \, \exp \left(\frac{x-\mu X}\sigma X} \right)\2 - 2\rho {XY}^2} \\ \cdot & amp; \, \exp \left(\frac{x-\mu X}\sigma X} \right)\2 - 2\rho {XY}^2} \\ \cdot & amp; \\ \cdot & a \end{align}\] Equation (2.1) contains the bivariate standard PDF. It's a little hard to get an insight into this complicated term. Instead, look at the special case where \(X\) and \(f_X(x)\) and \(f_X(x)\) and \(f_X(x)\) and \(f_X(x)\) and \(Y) do not correve with the normal normality) and \(f_X(x)\) and \(f_X(x)\) and \(Y) do not correve with the normal normal variable density (f_X(x)\) and \(f_X(x)\) and \(Y) do not correve with the normal normality) and \(f_X(x)\) and \(f_X(x)\) and \(f_X(x)\) and \(f_X(x)\) and \(f_X(x)\) and \(Y) do not correve with the normal normality) and \(f_X(x)\) and \(f_X(x)\) and \(Y) do not correve with the normal normality) and \(f_X(x)\) and \(Y) and \(Y $(rho_{XY}=0)$ (due to independence). Common density of (X) and $(Y) [g_{X,Y}(x,y) = f_X(x) f_Y(y) = \frac{1}{2} |eft[x^2 + y^right] |right], |tag{2.2} |] is the PDF of the biváriate normal distribution. The widget below is an interactive three-dimensional plot (2.2). If you move the cursor over the plot, you can see that the density is invariant in the$ direction of rotation, i.e. the density of \(a, b)\) depends solely on the distance between \(a, b)\) and origin; geometrically, areas of the same density are made up of concentric circles in plane XY, with \(huu X = 0, hu Y \= 0) \). Normal distribution has some notable properties. For example, for two typically distribued variables \(X\) and \(Y\), the conditional wait function is linear: it can be shown that \[E(Y\vert X) = E(Y) + \rho \frac{sigma_X} ((X - E(X)). \] The following interactive widget shows the normally distributed
bivariate data, together with the \(E(Y\vert X)). Conditional wait function and the marginal density of \(X) and \(Y). Each item changes accordingly as you change the parameters. The chi-squared distribution is another econometric distribution. It is often necessary to test specific types of hypotheses that are commonly occurring during the treatment of regression models. \(M\) degrees of freedom: \[\begin{align*} Z 1^2 + \\dots + Z M^2 = \sum {m=1}^M Z m^2 \sim \chi\2 M \\ VARIABLE PDF and CDF of a \(\chi^2_3\) on a single plot. This is achieved by set the argument to add = TRUE in the second call to the path(). In addition, we adjust the boundaries of the two axes using xlim and ylim and choose different colors to make both functions more distinguishable. The plot is a legend(). # plot the PDF curve(dchisq(x, df = 3), xlim = c(0,10), ylim = c(0,1) col = blue, ylab = , main = p.d.f. and c.d.f of Chi-Squared Distribution, M = 3) # add the CDF to the curve(pchisq(x, df = 3), xlim = c(0, 10), add = TRUE, col = red) # add a legend of the plot legend(top left, c(PDF, CDF), col = c(blue, red), lty = c(1, 1)) Since the results of a \(chi^2 M\) distributed random variable are always positive. the support for the related PDF and CDF \ (math{math{math{math{math{R}_{\geq0}}}. Since expectation and variance (only!) depend on the level of freedom, the shape of the distribution changes dramatically by changing the number of normal causes of the total squared normal. This relationship is often plotted by the overlap of the density of the distribution changes dramatically by changing the number of normal causes of the total squared normal. This relationship is often plotted by the overlap of the density of the distribution changes dramatically by changing the number of normal causes of the total squared normal. This relationship is often plotted by the overlap of the density of the distribution changes dramatically by changing the number of normal causes of the total squared normal. density of the \(\chi_1^2\) distribution in the \([0.15]\) value interval in the curve() function. In the next step, you pass the vacation rate \(M=2,...,7\) and add all \(M\) density curves to print. You can also set the line color for each iteration of the loop by setting col = M. Finally, we add a legend that displays the degree of freedom and related colors. # plot the density of M =1 curve(dchisq(x, df = 1), xlim = c(0, 15), xlab = x, ylab = Density, main = Kövis space distributed random variables) # add density M =2,...,7 the plot using the for() loop (M 2:7) { curve(dchisq(x, df = M), xlim = c(0, 15), add = T, col = M) } add # legend (top line, as.character(1:7), col = 1:7, lty = 1, title = D.F.) Increasing the level of freedom shifts to the right (the mode will be larger) and increases dispersion (the variance of the distribution increases). \(Z\) should be a standard normal variation, \(W\) is a random variable \(\chi^2_N\) and \(X\) follow a Slides \(t\) distribution (or simply \(t\) distribution with \(M\) freedom. Like the \(\chi^2_N\) distribution, the

shape of the \(t_M\) distribution depends on the \(M\) function. \(t\) distributions are symmetrical, bell-shaped and similar to normal distributions, especially if \(M\) distributions, especially if \(M\), the \(t_M\) distributions are symmetrical, bell-shaped and similar to normal distributions. study, the (t_{i_1}) distribution is the normal normal distributed random variable (X) and has a mismatch: (M_{t_1}) and (M_{t_1}) an standard normal density curve(dnorm(x), xlim = c(-4, 4), xlab = x, lty = 2, ylab = Density, main = density of T distributions) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density of Curve M=2(dt(x, df = 2), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4), col = 3, add = T) # representation of the density t of curve M=2(dt(x, df = 4), xlim = c(-4, 4) add = T) # add a legend (top right, c(N(0, 1), M=2, M=4, M=25), col = 1:4, lty = c(2, 1, 1, 1)) The act shows what it said in the previous paragraph: the increase in the freedom rate , the shape of the N(t) distribution is closer to the shape of the N(t) distribution is closer to the shape of the normal bell curve. For now (M=25), there is little difference from the normal density. If ((M) is small, we find that the distribution is heavier than a normal normal, i.e. it has the shape of a fatter bell. Another proportion of random variables important to econometrics is the proportion of two independent random variables distributed by \(\chi^2\) distribution with counter freedoms \(M\) and the (\n\) dens \(F_\m,n}\). The distributed in bonour of Sir Ronald Fisher. By definition, support for a random variable PDF and CDF distributed \(F_{M,n}) is \(\mathbb{R}_{\geq0}). Let's say you have a \(F\) distributed random variable \(Y\) with a degree of freedom \(3\) and denominator degrees of freedom \ (14)) and are interested in $(P(Y \ge 2))$. This can be calculated using the pf() function. If you set the lower.tail = F) #> [1] 0.1603538 This probability is based on drawing the related density line and polygon(). # enter the coordinateavekors in the polygon x <-c(2, seq(2, 10, 0.01), 10) y <- c(0, df(seq[2], 10, 0.01), 3, 14), 0) # $F_{3,14}$ curve density(df(x, 3,14), ylim = c(0,0,8), xlim rates are large, so that the distribution of \(F {M,n}) can be approximated by the \(F {M,\infty}) distribution, which turns out to be simply the random variable \(\chi^2 M.\) The book is part of Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see other people's notes by clicking on the re in the upper-right corner of the page to clarify the basic idea of random sampling, go back to the roll roll example: Suppose the dice roll \(n\) times. This means that we are interested in random \(Y_i, \ i=1,...,n\) results, which are characterized by the same distribution. Because these results are selected randomly, they are random variables themselves, and their implementation will be different each time a sample is taken, i.e. the numbers between \(1\) and \(6\), and have the same individual distribution. Therefore, \(Y_1,\point,Y_n\) is distributed the same. In addition, we know that the value of any of \(Y_i) are distributed independently. Thus, \(Y_1,\point,Y_n\) is distributed independently and in the same way (i.e.d.). The cube example uses this simplest sampling, \(n\) objects are drawn randomly from a group. All objects are equally likely to be sampled. The random \(I^{th}\) random \(Y\) variable is set to \(Y_i\). Because all objects can be drawn and \(Y_1\) is distributed the same for all \(i\), \(Y_i, \dots, Y_n\) is distributed independently and equally (i.e.). This means that the distributed independently and \(Y_2\) in \(Y_1, Y_3, \dots, Y_n\) and so on. What happens when we consider the features of sample data? Take, for example, rolling cubes twice in a row again. The pattern now consists of two independent random variables, e.g. their sum, is also random. Convince yourself code below several times. sum(sample(1:6, 2, replace = T)) #> [1] 7 It is clear that this amount, called \(S\), is a random variable because it depends on randomly drawn summands. In this example, you can fully list all results to describe the theoretical probability distribution of the sample data function \(S\): \(6^2=36\) is faced with possible pairs. Those pairs are \[begin{align*} & amp;(1,1) (1,2) (1,3) (1,4) (1,5) (1,6) \(2,2) (2,2) (2,3) (1,4) (1,5) (1,6) \(2,2) (2,2) (2,3) (1,4) (1,5) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,2) (2,3) (1,6) \(2,2) (2,3) (2,3) (1,6) \(2,2) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2,3) (2, (2,4) (2,5) (2,6) (3,2) (3,2) (3,2) (3,3) (3,4) (3,5) (3,6) (3,6) (4,2) (4,3) (4,4) (4,5) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,6) (4,63/36, \ & S = 4 \\ 4/36, \ & S = 5 \\ 5/36, \ & S = 6 \\ 6/36, \ & S = 7 \\ 5/36, \ & S = 10 \\ 2/36, \ & S = 11 \\ 1/36, \ & S = 10 \\ 2/36, \ & S = 11 \\ 1/36, \ & S = 10 \\ 2/36, \ & S = 11 \\ 1/36, \ & S = 11 \\ 1/36, \ & S = 10 \\ 2/36, \ & S = 10 \\ 2/36, \ & S = 12 \\ 1/36, \ & S = 10 \\ 2/36, \ & S = 12 \\ 1/36, \ & S = 10 \\ 2/36, \ & S = 1 <- c(1:6) / 36 # Expectation s ES <-sum(S * PS) ES #> [1] 7 # Variancia S VarS <- sum((S - c(ES))^2 * PS) VarS #> [1] 5.8333333 So distribution, i.e. the distribution of the outcome of a single cube cot (\D\). Imagine this bar
plots. # divide the print area into one row between two columns(mfrow = c(1, 2)) # plot distribution S barplot(PS, ylim = c(0, 0, 2), xlab = S, ylab = Probability, col = steelblue, space = 0, main = Sum of two cubes) # the distribution S barplot(PS, ylim = c(0, 0, 2), xlab = S, ylab = Probability) & lt;- 1:6 barplot(probability, ylim = c(0, 0, 2), xlab = S, ylab = Probability) & lt;- 1:6 barplot(probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability) & lt;- 1:6 barplot(probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability) & lt;- 1:6 barplot(probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability, ylim = c(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability = C(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability = C(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability = C(0, 0, 2), xlab = D, col = steelblue, space = 0, main = Sum of two cubes) # the distribution of probability = C(0, 0, 2), xlab = S(0, 2), xlab = S(0, 2), xlab = S(deal with the averages of the sampled data. It is generally assumed that observations are randomly drawn from a larger unknown population. As shown by the pattern function \(S\), calculating the average of the random variable. This random variable. This random variable. This random variable as a probability distribution called a sampling distribution. Knowledge of average sampling is therefore essential for understanding the performance of econometric procedures. (n) observations $(Y_1, dots, Y_n)$ sample average $[[(verline{Y}] + v_2 + vcdots + Y_n)$. $(verline{Y})$ is also called the sample average. suppose (Y_1, dot, Y_n) and (vu_Y) $(sigma_Y^2)$ denotes the Aztán ott van, hogy \[E(\overline{Y}) = E\left(\frac{1}{n} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(\sum_{i=1}^n Y_i \right) = \frac{1}{n} \text{Var}(\overline{Y}) = \text{Var}(\frac{1}{n} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) + \frac{1}{n^2} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(\sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) + \frac{1}{n^2} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(\sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) + \frac{1}{n^2} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) + \frac{1}{n^2} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) = \kart{Var}(Y_i) + \frac{1}{n^2} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) + \frac{1}{n^2} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) = \kart{Var}(Y_i) + \frac{1}{n^2} \sum_{i=1}^n Y_i \jobbra) \| = \kart{Var}(Y_i) = \kart{Va sum_{i=1}^n \sum_{j=1, jeq i}^n \text{cov}(Y_i, Y_j) \\ = \frac{\sigma^2_Y}n \text{cov}(Y_i, Y_j) \\ = \frac{\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample center value \[\sigma_{0} estiben a függetlenség miatt. Therefore, the sample ce of \(Y_i\). (Y_i\). (Y_i\). If \(Y_1,\dot,Y_n\) i.i.d. is derived from the normal distribution, average \(\mu_Y\) and variance \(\sigma_Y^2\), the following retentions for their sample average \(\verline{Y}\): \[\overline{Y}\) if the \(Y_i i=1,\dots,10\) sample is derived from a standard normal distribution with an average of \(\mu_Y = i+1,\dots,10\) sample is derived from a standard normal distribution with an average of \(\mu_Y = i+1,\dots,10\) sample is derived from a standard normal distribution with an average of \(\mu_Y = i+1,\dots,10\) if \(Y_i = 1,\dots,10\) sample is derived from a standard normal distribution with an average of \(\mu_Y = i+1,\dots,10\) sample is derived from a standard normal distribution with an average of \(\mu_Y = i+1,\dots,10\) if \(Y_i = 1,\dots,10\) if \ 0\) and variance \(\sigma_Y^2=1\) \[\[\overline{Y}\) from the \(\mathcal{N}(0.1)\) distribution and calculate the results of the true distribution and calculate the corresponding averages from 10 observations at random. If this is done for a large number of repetitions, the simulated dataset of averages should accurately reflect the theoretical distribution of \(\overline{Y}\) if the theoretical result holds. The approach outlined above is an example of what is commonly known as the Monte Carlo Simulation or Monte Carlo Simulation or Monte Carlo Simulation or Monte Carlo Simulation in R, we will do the following: Choose a sample size n and the number of samples to extract, reps. Note: replicate() results in a matrix with a rep dimension n \(times\). Contains the drawn patterns as columns. Calculation pattern means kolMeans(). This function calculates the average of each column, that is, each patterns as columns. Calculation patterns as columns. Calculation patterns as columns. Calculation pattern means kolMeans(). This function calculates the average of each column, that is, each patterns as columns. Calculation patterns as columns. Calculates the average of each column, that is, each patterns as columns. Calculation patterns as columns. 10000 sample matrix # calculation pattern means sample.avgs & lt;- colMeans(sample.avgs) #> [1] TRUE # print the first few entries on the console head(sample.avgs) #> [1] -0.1045919 0.2264301 0.5308715 0.2186909 0.2564663 A simple approach to testing the distribution of variable numerical data is to plot it as a histogram and compare it with some known or hypothetical distributions. By default, hist() gives us a frequency hysteria, a bar chart where observations are grouped into ranges, also known as containers. Ordinate is the number of observations in each container. Instead we want you to report density estimates for comparison purposes. This is achieved by setting the freq = FALSE argument. The number of bins is adjusted by argument to add the path to the current printout. Otherwise, R opens a new graphics tool and discards the previous graphic.3 # Plots the histogram hist(sample.avgs, ylim = c(0, 1.4), col = red, lwd = 2, add = T) The sampling distribution of the sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sampling distribution of the sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sampling distribution of the sample averages at the top of the histogram hist(sample.avgs, ylim = c(0, 1.4), col = red, lwd = 2, add = T) The sampling distribution of the sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd
= 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) The sample averages at the top of the histogram curve(dnorm(x, sd = 1/sqrt(n)), col = red, lwd = 2, add = T) Th very close to the distribution of \(\mathcal{N}(0,1)\), so the Monte Carlo simulation supports the theoretical claim. Let's look at another example where simple random sampling in a simulation setting will help you check a well-known result. As discussed earlier, the chi-squared distribution with \(M) degrees of freedom is used as a distribution of the sum of independent, square normal distributed random variables. To display the statement specified in equation (2.3), do the same as in the example above: Select the freedom, DF, and number of samples. For each sample, square the results and sum them column-wise. overlaps with the line chart of the theoretical density function of the \(\chi^2_3\) distribution. # number of repetition(reps, rnorm(DF)) # column sum squares X <- colSums(Z^2) # histogram column sums squares hist(X, freq = F, col = steelblue, breaks = 40, ylab = Density, main =) # add a theoretical density curve(dchisq(x, df = DF), type = I, lwd = 2, col = red, add = T) The sampling distributions taken into account in the last stage play an important role in the development of econometric methods. There are two main approaches to characterisation of sampling distributions: an accurate approach and an approximate approach. The exact find a general formula for the samplies to any sample size (\n\). This is called an exact distribution or finite sample distribution or finite sample distribution is precisely known in the sense that we can describe them in analytical terms. However, this is not always possible. For \(\overline{Y}\), the result (2.4) indicates that the normality of \(\verline{Y}\) using the \(n=10\) simulation test, which involves simple sampling). Unfortunately, the exact distribution of \(\overline{Y}\) is usually unknown and is often difficult to deduce (or even untraceable) if you discard the assumption that \(Y_i\) has a normal distribution. Therefore, as can be seen from its name, the approximate the sample size \(n\) is large. The distribution used as a large sample zoom of the sampling distribution. distribution is also called the asymptotic distribution. This is due to the fact that the asymptotic distribution and the asymptotic distribution is negligible for medium or even small samples, so approximations using asymptotic distribution are useful. In this section, we will discuss two well-known results that are used to approximate sampling distributions and are thus key tools in econometric theory: the law of large numbers states that in large samples, \(\overline{Y}\) is most likely near \(\mu_Y\). The central limit entry says that the sampling distribution of the standard sampling average, i.e. \(\overline{Y}-\mu_Y)\sigma_{\overline{Y}}), is usually asymptotic. It is particularly interesting that both results do not depend on the distribution of \(Y). In other words, if the complex sampling distribution of \(/V). In other words, if the complex sampling distribution of \(/V). In other words, if the complex sampling distribution of \(/V) is not normal, the approximate of the latter using the central limit item greatly simplifies the development and applicability of econometric procedures. This is a key component that is the basis for statistical conclusion theory of regression models. Both results are summared in Key Concept 2.7. \(\overline{Y}\) is converge in probability \(\mu_Y): \(\overline{Y}\) is converge in probability \(\overli and \(\mu Y+ anywhere near \(1\) n(n) for \(\epsilon > 0\). Do this \[P(\mu Y-\epsilon \leq \overline{Y} \leq \mu Y + \epsilon) \rightarrow1, \, \epsilon > 0 \text{ as } n\rightarrow1, \, \epsilon > 0 \text{ as } n\rightarrow 1, \, \epsilon > 0 \text{ as } n\rightarrow1, \, \epsilon > 0 \text{ as } n\text{ as } n\t (\sigma^2_Y< \infty\), i.e. large outliers are unlikely, so the law of large numbers states that \[V \overline{Y} \xrightarrow[]{p} \mu_Y. \] The following application simulates a fraction of the heads observed for each additional roll. The result is a random path that, according to the law of large numbers, indicates that \(0.5\) is increasing \(n\). The basic statement of the Law of Large Numbers is that, in fairly general circumstances, the probability of obtaining a sample average of \(\overline{Y}\) that is close to \(\u00erline{Y}\) is high if you have a large sample size. For example, consider the example where you repeatedly tossed a coin where \ (Y_i) is the result of a coin toss (i^{th}) . (Y_i) is a Bernoulli distributed random variable (p) with the probability of head observation $P(Y_i) = 0$ amp; $Y_i = 1 \ 1-p$, amp; $Y_i = 0 \ (P(Y_i) = 0$ according to the law of large numbers, the the observed ratio of heads is likely \(mu_Y = 0.5\), probability of tossing the head into a single coin toss\[R_n \xrightarrow[]{p} \mu_Y = 0.5 \ text{as} \ \text{as} \ text{as} \ sample observations from the Bernoulli distribution, e.g. using sample(). Calculate the \(R_n\) ratio of the heads in (2.5). One way to do this is to call cumsum() Y for the vector of observations. We continue by plotting the path and adding a dashed line to the probability of performance probability (p = 0.5). # set seed set.seed(1) # set number of coin throws and simulate N & lt;- 30000 Y & lt;- pattern(0:1, N, replace = T) # R_n calculate 1:N S & lt;- cumsum(Y) R 0,5 rows(c(0, N), c(0,5, 0,5), col = darkred, lty = 2, lwd = 1) There are several things to say this site. The blue graph shows the observed proportion of heads when a coin is tossed \(N_1). Because \(Y_1) are random variables, \(R_n), as determined by \(30000) observations taken from the Bernoulli distribution. If the number of coin tosses is \(n\) small, the ratio of heads can be anything but close to its theoretical value, \(\mu_Y = 0.5\). However, as more and more observations are included in the sample, we find that the road stabilizes in the area \(0.5\). The average of multiple experiments clearly shows that as the sample size increases, they approach the expected value, as required by the law of large numbers. Let's say that random variables \(Y_1,\dot,Y_n\) are allocated independently and in the same way as expected \(E(Y_i)=\mu_Y\) and variance \(\text{Var}(Y_i)=\sigma^2_Y\) where \(0<\sigma^2_Y\) where \(0<\sigma^2_Y\). Central Limit Theorem (CLT) states that if the sample size \(n\) is infinity, standard sample average \[\frac{\overline{Y} - \mu_Y}\sigma_{\overline{Y}} = \frac{\overline{Y}} = \frac{\overline (\overline{Y}) narrows around the true average of \(5\). The distribution of the standardized sample average is close to the normal distribution of the large \(n\) standard. According to CLT, the distribution of the standardized sample average of Bernoulli's distributed random variables \(Y_i), \(i=1,...,n\) is \(\mu_Y=p=0.5\) and \(\sigma^2_{Y} = p(1-p)/n = 0.25/n\) is a good approximation with large of Bernoulli's distribution of the standard. According to CLT, the distribution of the standard. \(\n\n) parameters. Consequently, for the standard distribution \(\mathcal{N}(0.1)\). We're using another simulation study to graphically show this. Here's the idea. Draw a large number of random patterns (\10000\) from the Bernoulli distribution\(n\) and calculate the sample averages. Standardisation of averages as shown in (point 2.6). Then distribution of the generated standardized sample size \(n\) to see how the simulated distribution of averages affects the sample size \(n\). In R, notice this as follows: Let's start by determining whether the next four later generated numbers \(2\times2\) array to make them easy to compare. This is done by inviting par(mfrow = c(2, 2)) before creating the figures. The number of repetitions is defined as \(1000\) and a sample size vector named sample size is created. We take into account patterns that are \(5\), \(20\), \(75\), and \(100\). Then we merge two for() loops to simulate the data and plot the divisions. The inner loop \(10000\) creates random patterns, each of which consists of n observations the inner loop for different n sample sizes and creates a plot for each iteration. # divide the plot panel into a 2x2 array par(mfrow = c(2, 2)) # set the number of repetitions and sample size <- 10000 sample.sizes <- c(5, 20, 75, 100) # set seed for reproducibility set.seed(123) # outer loop (loop over the sample sizes) for (n in sample.sizes) { samplemean <- rep(0, reps) #initialize the vector of the pattern means that stdsamplemean <- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean <- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample
vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that stdsamplemean & lt;- rep(0, reps) # initialize is the standardized sample vector means that st internal loop (repetitions of repetitions) for (i 1:reps) { &It; x- rbinom(n, 1,0,5) samplemean[i] &It;- average(x) stdsamplemean[i] &It;- sqrt(n)** (mean(x) - 0.5)/0.5 } # plot histogram and overcloth with N(0.1) density for each iteration hist(stdsamplemean[i] &It;- sqrt(n)** (mean(x) - 0.5)/0.5 } # plot histogram and overcloth with N(0.1) density for each iteration hist(stdsamplemean[i] &It;- sqrt(n)** (mean(x) - 0.5)/0.5 } col = darkred, add = TRUE) } We find that the simulated sampling distribution of the standardized average is generally very different from the normal distribution. The approach works pretty well, see \(n=10\). However, as \(n\) increases, histograms approach the normal distribution. The approach works pretty well, see \(n=10\). However, as \(n) increases, histograms approach the normal distribution. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see the notes of others: click on the upper right corner of the page Let's say that the lottery fairy is a weekly lottery where \(6\) out \(49\) unique numbers must be drawn. Instructions: Draw the winning numbers for the week. Tips: Use the function pattern() to draw random numbers, see sample. The set of items to sample from here is \\\{1,...,49\}\\). Take a random \(X) variable with probability density function (PDF) \[f_X(x)=\frac{x}{4}e^{-x^2/8}, 'quad x\geq 0.\] Instructions: Define the PDF from above as f(). exp(a) calculates \(e^a\). Verify that the specified feature is a PDF. Hints: You can use function(x) {...} to define a function that x. In order to be a f() PDF, the pdf in the entire range must be \(\\int_0^\infty f_X(x)\mathrm{d}x=1\). The integrate() function integrates. You must specify the function to be integrated, as well as the upper and lower limits of the integration. These options can be set to \([-\infty,\infty]\) by setting the corresponding arguments to -Inf and -Inf. You can achieve the numeric value of the calculated \$value by threading the data. For a detailed description of this feature, see ?integral. In this exercise, you must calculate the expected value and variable \(X\) taken into account in the previous practice. Pdf () from the previous exercise is available in the work environment. Instructions: Enter an appropriate ex() function that is integrated with the expected value of \(X\). Store the expected value of \(X\). $xf_X(x)dx$). The integrated value calculated by integrate() is value through a single set. The variance of (X) is $(Var(X)=E(X^2)-E(X)^2)$, where $(E(X^2)=k(X)^2)$. Tips: $(\rhohi(s))$, i.e. standard normal density (c=3). Tips: $(\rhohi(s))$, i.e. standard norm(). Note that by default, dnorm() uses averages = 0 and sd = 1, so you don't need to set the appropriate arguments to obtain density values for the normal distribution. Tips: (P(| Z||eq z) = P(-z |eq Z||eq z)). Probability of (P(a |eq Z |eq b)) is $(P(Z||eq b)-P(Z||eq a)=F_Z(b)-F_Z(a))$ and (P(| Z||eq z) = P(-z |eq Z||eq z)). (F_Z(\cdot)) using the cumup distribution function (CDF) of \(Z). Alternatively, you can take advantage of the symmetry of the standard normal distribution, i.e. locate \(y\) by \(\Phi(\frac{y-5}{5})=0.99\). Hints: A quantitative version of the normal distribution can be calculated using the qnorm() function. In addition to the quantile to be calculated, the mean and standard deviation, not the variance! sqrt(a) returns the square root of the numeric argument. Leave \(Y\sim\mathcal{N}(2, 12)\). Instructions: Generate \(10\) random numbers from this distribution. Tips: Use rnorm() to draw random numbers from a normal distribution. In addition to the number of drags, the mean and standard deviation of the distribution must be specified. This can be done through arguments et mean and standard deviation of the distribution. In addition to the number of drags, the mean and standard deviation of the distribution must be specified. Enter the range of x-values \([0.25]\) through the xlim argument. Tips: curve() expects function and parameters arguments (here dchisq() and \(X_1)) and \(X_2\) be two independent random variables that are normally distributed \(\mu=0\) and \(\sigma^2=15\). Instructions: Calculation \(P(X 1^2+X 2^2&qt;10)\). Tips: Note that \(X 1\) and \(X 2\) are distributed instead of \(\mathcal{N}(0.1)\). Instructions: Calculate the probability. The argument lower.tail can be useful. Leave ,(X\sim t {10000}) and \(Z\sim\mathcal{N}(0.1)\). Instructions: Calculate the quantitative [(95\%\) quantitative of both distributions. What did you notice? Tips: Use qt() and qnorm() to calculate quantitative variations of specific distributions. For \(t\) to be distributed, you will see a representation of the appropriate probability density function (PDF). Instructions: Create \(1000\) random numbers from this distribution and assign them to variable x. Calculate the sample center of x. Can you explain the result? Tips: Use rt() to draw random numbers from a t-distribution. Note that the t distribution fully determines the degree(s) of freedom. Enter them through the df argument. To calculate the sample mean of a vector, you can use the center(). Be \(Y\sim F(10, 4)\). Instructions: Plots the quantile function of the specified distribution using the function curve(). Tips: curve() counts the function with the appropriate parameters (for degrees of freedom df1 and df2) in an argument. Be \(Y\sim F(4,5)\). Instructions: Integrate the PDF to calculate \(P(1<Y<10)\). Tips: In addition to the function to be integrated, you need to specify the upper and upper limits of integration. The additional parameters of distribution (here df1 and df2) must also be passed on the call on integrated value \$value through a single set. Page 9 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also view other people's comments by clicking on the upper-right corner of the page This section reviews important statistical concepts: Estimating unknown population will report and discuss several applications R. These R applications are applications the following packages, which are not part of the basic version of R: readxl - allows you to import data in Excel R. dplyr - provides flexible grammar for manipulating data. MASS - a collection of functions of applied statistics. Make sure that they are installed before you advance and try to replicate the examples. The safest way to do this is to verify that the next chunk of code executes without errors. library (dplyr) library (dplyr) library (dplyr) library (dplyr) library (mass) directory (readxl) Page 10 This book is open review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also view comments from others by clicking on the one in the upper-right corner of the page, precious is a function of sample data from an unknown population. Estimates are numerical values calculated by estimates based on sample data. Appraises are random variables because they are functions of random data. Estimates are not random numbers. Think of some economic variables, such as the hourly wages of college graduates, which are marked \(Y\). Let's say you're interested in \(\mu_Y\) is the \(Y\) mean. In order to accurately calculate \(\mu_Y\) we would have to interview all working graduates in the economy. We simply cannot do this because of time and cost constraints. However, you can randomly sample \(n\) i.i.d. observations \(Y_1, \dots, Y_n\) and estimate \ (μ_Y) are one of the simplest estimates to be thought of as key concept 3.1, i.e. \ [\overline{Y} = \frac{1} {n} \sum_{i=1}^Y_i, \, \, sample average \(Y\). Then again, we could use an even more model \(\mu_Y\): the very first observation in the model, \(Y_1\). Is \(Y_1\) a good estimate? Now let's say \[Y \sim \chi_{12}^2 \] which is not very unreasonable, since hourly income is not negative, and we expect that many hourly incomes are between $(5 \in)$. In addition, it is common for the income distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be
pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the $(12 \le 12)$ allocation. # plots chi_12^2 distribution to be pushed to the right — the property of the right — the property of the plots chi_12^2 distribution to be pushed to the right — the property of the property of the right = the property of the right = the property of the right = the property of the property o first observation \(Y_1\) estimate \(\mu_Y\) # set seed reproducibility set.seed(1) # sample of chi_12 ^2 distribution, only the first observation rs < amp- rchisq(n = 100, df = 12), but it's somewhat intuitive to see what's better: the estimate \(Y_1\) throws away a lot of information and its variance is the variance of the population: \[\text{Var}(Y_1) = \text{Var}(Y_1) = \text{Var}(Y) = 2 \cdot 12 = 24 \] This brings the question to the following: The desired characteristics of the estimate include impartiality, consistency and efficiency. Unbiased: If the average sample distribution of each estimation of the population (\\hat\mu_Y) is \(\mu_Y), \[E(\six\mu_Y), \[E(\si of an estimate of \(\six\mu Y\) to be within the small interval around \(\mu Y\) to get closer to \(1\) as \(n\) grows. This is done by \[\[\six\mu Y\) and \(\overset{\sim} Y\) and for some given sample size \(n\) it holds that \[E(\six\mu Y) = E(\overset{\sim}{\mu}_Y) = \mu_Y] but \[text{Var}(\six\mu_Y) is more efficient in using the information provided by the observations in the sample. Page 11 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click on the pop-up menu. You can also see the annotations of others: click the in the upper right hand corner of the page A more precise way to express consistency of an estimator \(\six\mu\) for a parameter \(\mu\) is \[P(|\hat{\mu} - \mu|<\epsilon) \\ xrightarrow[n \rightarrow \infty]{p} 1 \quad \text{for any}\quad\epsilon>0.\] This expression says that the probability of observing a deviation from the true value \(\mu\) that is smaller than some arbitrary \(\epsilon > 0\) converges to \(1\) as \(n\) grows. Consistency does not require impartiality. To test the sample average per estimate value of the average of the corresponding population, consider the following example R. Creates a population population consisting of observations \(Y_1), \(i=1,\dot,1000\) derived from the normal distribution, mean \(\\mu = 10\) and variance \(\\sigma^2 = 1\). To test the behavior of the reviewer \(\\sigma^2 = 1\). Yo test the behavior of the reviewer \(\\sigma^2 = 1\). To test the behavior of the reviewer \(\\sigma^2 = 1\). using the function repeat(). The argument is evaluated n times. In this case, we take samples \(n=5\) and \(n=25\), calculate the sample average, and accurately repeat \(N=25000\). # # a fictitious population by the first observation of the sample average, and accurately repeat \(n=5\) and \(n=25\), calculate the sample average, and accurately repeat \(N=25000\). population and estimate the average evening1 <- repetition(expr = average(sample(x = pop, size = 5)), n = 25000) Check, whether est1 and est2 are vectors of length (25000): # verify that the object type is vector(est1) #> [1] TRUE.vector(est2) #> [1] TRUE # #> [1] 125000 length(est2) #> [1] 25000 The following code array always represents the density function of the \(\mathcal{N}(10,1)\) distribution. # estimation of plot density Y_1 sample(density(fo), column = green, lwd = 2, ylim = c(0,2), xlab = estimates, main = Sampling distribution of the sample average by n= 5 to the sample average by n= 25 to the sample lines (density(est1), col = red2, lwd = 2) # add to the distribution of the sample average the density (est1), col = steelblue, lwd = 2, bty = I) # add to the distribution of the sample average by n= 5 to the sample average the density (est1), col = steelblue, lwd = 2, bty = I) # add to the distribution of the sample average by n= 5 to the sample average by n= 25 to the sample average by n= 5 to the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 5 to the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I) # add to the distribution of the sample average by n= 2, bty = I + 2 $I_{1} = 2$ = n(10,1) for the print curve(dnorm(x, mean = 10), lwd = 2, lty = 2, add = T) # add legend added (top left, legend = c() N(10,1), expression(Y ~ n == 25) lty = c(2, 1, 1, 1), col = c(black), green, steelblue, red2), lwd = 2) First it is placed around the total sampling distribution (represented by solid lines) \(hu = 10\). This is evidence of the impartiality of \(Y 1\), \(\overline{Y} {5}\) and \(\overline{Y} {5}\). Of course, the theoretical density \(mathcal{N}(10,1)\) is also located at \(10\). Then see the spread of sampling distribution of \(Y 1\) (green curve) closely follows the density of the \(\mathcal{N}(10,1)\) distribution (black dashed line). In fact, the Y_1 distribution of \(\mathcal{N}(10,1)\) is \(\mathcal{N}(10,1)\) distribution. Therefore, \(Y_1 \sim \mathcal{N}(10,1)\). Note that this result does not depend on the sample size \(n\): the sampling distribution \(Y_1\) is always the population distribution, no matter how large the sample is. \(Y_1\). In the light of key concepts 3.2 and 3.3, we see that \(\overline{Y}\) show less dispersion than the sampling distribution of \(Y_1\). This means that \(\overline{Y}\) has a smaller variance than \(Y_1\). In the light of key concepts 3.2 and 3.3, we see that \ (\overline{Y}) more efficient estimate than \(Y_1). In fact, this applies to all \(n>1). \(VOverline{Y}) displays consistency behavior (see key word 3.2). The blue and red densities are more concentrated around \(\mu =10\) than green. Because the number of observations increases from \(1) to \(5)), the sampling distribution narrows around the real parameter. Increasing the sample size to \(25\), this effect becomes more evident. This means that the probability of obtaining estimates close to fair value increases by \(n\). This is reflected in the estimated values of the density function close to 10: the larger the sample size, the higher the density. We recommend that you go ahead and change the code. Try the different values for the sample size and see how the sampling distribution of \(verline{Y}) changes. Let's say you have some observations \(Y_1,\dot,Y_n) on \(Y \sim \mathcal{N}(10,1)\) (which is unknown) and want to find an estimate \(m\) that predicts the observed values is small. Mathematically, this means to find a \(m\) value that minimizes \[\begin{equation} \sum_{i - m}^2. \tag{3.1} \end{equation}\] Think of \(Y_i - m)) error as error in predicting \(Y_i) \(m\). We can minimize the amount of absolute deviations \(m\), but minimize the amount of square deviations mathematically more convenient (and lead to another result). Therefore the estimate we are looking for is called least squares estimate. \(m = \overline{Y}\), the sample average is this estimate. You can show this by generating a random sample and plotting (3.1) as a function of \(m\). # define the function and vector it sqm <- function(m) { sum((y-m)^2) } m <- Vectorize(sqm) # draw random pattern and y <- rnorm(100, 10,1) mean(y) #> [1] 10,1364 # represents the curve of the lens function(s) (square meter(x), =50 to = 70, xlab = m, ylab = sqm(m)) # adding a vertical line to the mean(y) abline(v) = mid(y), lty = 2, col = tinted) # add annotations for average(y) text(x = medium(y), y = 0, tags = paste(round(center(y, 2)) Pay attention to (3.1) a quadratic functions that is only a minimum. The sample shows that this minimum lies exactly at the sample data. Some R functions work only with functions and it is often a good idea to write features at yourself, although it is cumbersome in some cases. Having a vector function of R is never a disadvantage, since these functions work on both individual values and vectors. Let's look at the sqm(), which is not vector function of R is never a disadvantage, since then the y-m operation is invalid: vectors y and m are incompatible dimensions. Therefore, you cannot use nm() on the curve(Y_1). \dots, Y_n\) is the result of a sampling process that meets the assumption of simple random sampling. This assumption is often met when an average is estimated using \(\verline{Y}\), and if this is not the case, estimates can be biased. average \(\mu_{\texttt{pop}}\): # pop average (pop) #> [1] 9.992604 Next pattern \(10\) observations from pop with pattern() and \(huu_{texttt{pop}}) are repeatedly sampled from pop. However, we are now using a sampling system that is
different from simple random sampling: instead of ensuring that all members of the population have the same chance of being sampled, we are more likely to receive sampling. for the smallest observations of the population \(2500\) in such a way that to set the prob argument to the correct vector of probability weights: # simulate the outcome of the sample average if assumption i.i.d. fails 3 <- repetition(n = 25000, expr = average(sample(x = sort(pop), size = 10, prob = c(rep(4, 2500), rep(1, 7500) # calculates the sample average of the results average(est3) #> [1] 9.444113 Then we plot the \(\overline{Y}\) sampling and compare it to the sampling distribution when assumption i.i.d. hold, n=25 parcel(density(est2), col = steelblue, lwd = 2, xlim = c(8, 11), xlab = Estimates, main = If i.i.d. Assumption fails) # distribution of sample average, i.i.d. fails, n=25 rows(est3), col = red2, $|wd = 2| # add a |egend(topbal, |egend = c(bar(Y)[n== 25]~, i.i.d. failed), expression(bar(Y)[n== 25]~, i.i.d. tart)), lty = c(1, 1), col = c(red2, steelblue), |wd = 2| Here the error in assumption i.i.d. means that on average we underestimate \(\mu_Y\) if the$ assumption i.i.d. does not contain it. Page 12 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can also view other people's notes by clicking on the in the corner of the page In this section, we briefly review the prayers of hypothesis and discuss how to perform hypothesizer tests in R. We're focused on drawing conclusions about an unknown population average. In an important test, we want to use the information in the sample as evidence or against a hypothesis. Essentially, hypotheses are simple questions that can be answered with yes or no. In a hypothesis, we usually deal with two different hypotheses: The null hypothesis, which is indicated by \(H_0\), is the hypothesis that interests us in testing. There must be an alternative hypothesis is rejected. The null hypothesis is that the population average of \(Y\) is the same as \(\mu_{Y,0}\), \[H_0: E(Y) = \mu_{Y,0}. \] Often the alternative hypothesis chosen is the most common, \[H_1: E(Y) eq \mu_{Y,0}, \] which means that \(E(Y)\) can be anything but the null hypothesis. This is called a two-sided alternative. For the sake of shortness, we will only consider two-sided alternative in the next sections of this chapter. Let's say the null hypothesis is true. The \(p)\value is the probability that data will be drawn and observed in an appropriate test statistic that is at least as unfavourable as that in the null hypothesis, since the test statistics are actually calculated using the sample data. In the context of the population average, this definition can be mathematically defined as \[\begin{equation} p \text{-value} = P_{H_0} \[\lvert \vert \ \mu_{Y.0} \rvert \right] \tag{3.2} \end{equation}\] In (3.2), it is necessary to know the sampling distribution of \(\overline{Y}\(a random variable) if the null hypothesis is true (the null distribution). However, in most cases, the sampling distribution and thus the null hypothesis is true (the null distribution). However, in most cases, the sampling distribution and thus the null hypothesis is true (the null distribution). distribution of $(|verline{Y}|)$ are unknown. Fortunately, CLT (see Key Concept 2.7) allows you to approximate the large sample $||verline{Y}| + \frac{1}{2}{n}, ||$ assuming the null hypothesis $(H_0: E(Y) = \frac{Y}{0})$ is true. With a few algebras, the lage (n) follows, $|| + \frac{Y}{0}| +$ \mu_{Y,0}{\sigma_Y/\sqrt{n}} \sim \mathcal{N}(0.1). \] So in large samples, the \(p\) value can be calculated using the above normal approach without knowing the exact sampling distribution of \(\overline{Y}\). For now, suppose \(\sigma_{\overline{Y}}) is known. You can then change (3.2) \[\begin{align} p \text{-value} =, P_{H_0}\left[\left\vert \frac{\overline{Y} - \right\vert > $left|vert frac{overline{Y}{act} - [mu_{Y,0}]{sigma_{overline{Y}} right|vert right] (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \left|vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{overline{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \left] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \left] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \left] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \left] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \left] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \left] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \ right] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right|vert right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \ right] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \ right] (vert \frac{over line{Y}^{act} - [mu_{Y,0}]}{sigma_{over line{Y}} right]} (p)-value is the area of the ((mathcal{N}(0.1)) distribution that exceeds [[begin{equation} pm \ right] (vert$ $sigma_{overline{Y}} rightrver \tag{3.4} = c, 4, 4$, main = P-value calculation, yaxs = i, xlab = z, ylab = , lwd = 2, axes = F) # add x-axis(1, at = c(-1,5, 0, 1,5), padj = c(-4, 4), main = P-value calculation, yaxs = i, xlab = z, ylab = z, yla $0,75, tags = c(expression(-frac(bar(O)^act - ~bar(mu)[Y,0], sigma[bar(Y)])), 0, expression(frac(bar(Y)^act - ~bar(mu)[Y,0], sigma[Y)])$ bade p-value/2 regions in the left back of the left polygon(x = c(-6, seq(-6, -1.5, 0.01), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(1.5, 6, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c(-6, seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c() 1.5, seq(-6, -1.5, 0.01), 6), y = c(0, dnorm(seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c(-6, seq(-6, -1.5, 0.01)), 0), col = steelblue) # shadow p-value/2 regions in right rear polygon(x = c(-6, seq(-6, -1.5, 0.01)), col = steelblue) # shadow p-value/2 regions in right rear polygo dnorm(seq(1,5, 6, 0,01)), col = steelblue) If \(\sigma^2_Y) unknown, should be appreciated. To make the sample variance \[\begin{equation} s_Y = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \verline{Y})^2} \end{equation} s_Y = (Y_i - \verline{Y})^2 (Y_i - \verline{Y})^2 (Y_i - \verline{Y})^2) (Y_i - \v implemented in sd(), see :sd. Use R to demonstrate that \(s_Y) is a consistent appraiser of \(\sigma_Y), i.e. \[s_Y \overset{p}{\longrightarrow} \sigma_Y. \] The idea here is to create a large number of patterns \(Y_1,\dots,Y_n\) where\(Y\sim \mathcal{N}(10, 9)\) say, estimate \(\sigma_Y\) and examine how the \(s_Y\) will change to \(n\). # vektor n <-c(10000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000,
5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 5000, 2000, 1000, 500) # minta megfigyelések, becslés az 'sd() használatával, és a becsült eloszlások sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3))) minta(sűrűség(sq_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) sorok(sűrűség(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), fő = kifejezés(ek[y]), lwd = 2) a (i 2:hossz(n)) { sq_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), sg_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), sg_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), sg_y <- ismétlés(n = 10000, expr = sd(rnorm(n[1], 10, 3)) minta(sűrűség(sg_y), sg_y <- ismétlés(sg_y), sg_y <- i } # adjunk hozzá egy jelmagyarázat-jelmagyarázat-jelmagyarázatot(bal felső, jelmagyarázat = c(kifejezés(n == 5000), kifejezés(n == 5000), kifejezés(n == 2= 2)) A telek azt mutatja, hogy a \(s_Y\) eloszlása a \(\sigma_Y = 3\) valós értéke körül a \(n\) növekedésével érik el. A function that estimates the estimation standard deviation is called the standard error of estimation. Key Concept 3.4 summarites terminology in its mid-sample context. Take an i.i.d. pattern \(Y_1, \dots, Y_n\). The average of \((V\) is continuously estimated by the sample average of \(\overline{Y}\), \(Y_i). Because \(\overline{Y}\) is a random variable, it has a variance distribution\(\frac{\sigma_Y^2}{n}). \(\overline{Y})) is a random variable, it has a variance distribution\(\frac{\sigma_Y^2}{n}). \(\overline{Y})) is a random variable, it has a variance distribution(\frac{\sigma_Y^2}{n}). \(\overline{Y})) is a random variable, it has a variance distribution(\frac{\sigma_Y^2}{n}). \(\overline{Y})) is a random variable, it has a variance distribution(\frac{\sigma_Y^2}{n}). \(\overline{Y})) is a random variable, it has a variance distribution(\frac{\sigma_Y^2}{n}). \(\overline{Y})) is a random variable, it has a variance distribution(\frac{\sigma_Y^2}{n}). \(\overline{Y})) is a random variable, it has a variance distribution(\frac{\sigma_Y^2}{n}). SE(\overline{Y}) = \six\sigma_{\overline{Y}}) is an estimate of \(\sigma_\) i.i.d. from the distributed variable \(Y\) with a successful probability \(p=0.1\). So \(E(Y)=p=0.1\) and \ (V) = p(1-p), (E(Y)) megbecsülhető $(\operatorname{V}) = 0,009$] és szórás $[\operatorname{V}) = 0,0009$] és szórás megfelelő valós értékeket. # draw 10000 minták mérete 100 és becsülje meg az Y középértékét, és # becsülni a standard hiba a minta átlag mean_estimates <- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) a (i 1:10000) { s & triates & lt;- numerikus(10000) { s átlag (mean_estimates) #> [1] 0,10047 átlag(se_estimates) #> [1] 0,02961587 becslés Mindkét becslés úgy tűnik that is not unbiased to trues. In fact, this is true for the sample mean, but not for \(SE(\overline{Y}\)). However, both estimates are consistent with the real parameters. If \(\sigma_Y\) is unknown, \(\mu_Y overline{Y}\) \(\overline{Y}\)). However, both estimates are consistent with the real parameters. If \(\sigma_Y\) is unknown, \(\mu_Y overline{Y}\) \(\overline{Y}\)). $(sigma_{overline{Y}}) (3.3)$ with standard error(SE(\overline{Y}) right) = (0.9, 0.1), replacement = T, size = 100) (3.3) with standard error sample(0:1, prob = c(0.9, 0.1), replacement = T, size = 100) SE_samplemean <- sqrt(samplemean_act * (1 - samplemean_act) / 100) # null hypothesis mean_h0 <- 0.1 # the p-value pvalue #> [1] 0.7492705 Later in the book, you will encounter more convenient approaches to obtaining \(t\)-statistics and \(p\)-values using r. Hypothesis test age is the standard sample take average \[\begin{equation} t = - \mu_{Y,0}}(SE(\overline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y})) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline{Y}) in the previous code data. # calculate a t-statistic for \(\verline the sample average tstatistic <- (samplemean_act - mean_h0) / SE_samplemean tstatistic #> [1] Using 0.3196014 R, we can illustrate that if \(\mu_{Y.0}) is the same fair value, i.e. if the null hypothesis is true, (3.5) is approximately \(\mathcal{N}(0.1)\) distributed if \(\nu) is large. # blank vector preparation for tstatistics <- numerical(10000) # set sample size n <- 300 # simulate 10000 t-statistics (i 1:10000) { s <- sample(0:1, size = n, prob = c(0,9, 0,1), replacement = T) tstatistics[i] <- (mean(s)-0.1)/sqrt(var(s)/n) } In the above simulation, the variance of \(Y_i) is estimated using the var(s). This is more common then the average(s)*(1-middle(s)), since the latter requires that the data Bernoulli disseminated and that we know this. # plot density and compare the N(0.1) density plot(tstatistics), xlab = t-statistics, Main = Estimated distribution of t-statistics n=300, lwd = 2, xlim = c(-4, 4), col = steelblue) # N(0,1) density (dnorm(x), addition = T, lty = 2, lwd = 2) Judging from the sample, the normal approach is reasonably working well in relation to the sample size selected. This normal approximation has already been used to determine the \(p\) value, see (3.5). In hypothesis testing, two types of errors are possible: The null hypothesis is rejected, although true (type-I-error) The significance level of the test is the probability to commit the type of I-error we are willing to accept in advance. For example, using the predefined significance level of \(0.05\) rejects the null hypothesis if and only if the \(p\) value is less than \(0.05\). The significance level should be selected before the test is carried out. An equivalent procedure is to reject the null hypothesis if the observed test statistics are in absolute terms greater than the critical value of the test is carried out. An equivalent procedure is to reject the null hypothesis if the observed test statistics are in absolute terms greater than
the critical value of the test statistics. The critical value is determined by the selected significance level and defines two separate sets of values called acceptance regions and rejection regions. The acceptance area shall contain all the values of the test statistics which the test statistics which the test statistics may be observed which provide the same evidence against the null hypothesis as the test statistics are actually observed. The same as the significance level. The probability that the test correctly rejects the false null hypothesis is called power. Rethink the pvalue value calculated above: # verify that the p-value &It; 0.05 pvalue &It; 0.05 pvalue &It; 0.05 #> [1] False condition is not met, so we do not correctly reject the null hypothesis is rejected at the level of rigour of \(5\%\), if the calculated \(t\) statistics in absolute terms exceed the critical value of 1.96. \(1.96\) is the standard normal distribution \(0.975\)-quantile. # check the critical value qnorm(p = 0.975) # > [1] 1.959964 # check, that null reject the t-statistics calculated further above abs (tstatistic) > 1.96 # > [1] FALSE Just like the \(p\)-value, we cannot reject the null hypothesis with the corresponding \(t\)-statistics. Key Concept 3.6 summarises the procedure for performing a two-sided hypothesis test on the population average \(E(Y)). Estimate \(\mu_{Y}) using \(\overline{Y}), \(\overline{Y}). Estimate the usual error in \(SE(\overline{Y})). If the \(p) value is less than \(0.05) or equivalent if \[\[\[\left\|vert t^{act} \right\rvert > 1.96. \] Sometimes it is worth testing if the average is greater than or less than some values that are below null. To stick to the book, take out the supposed pay gap between well-educated individuals. Since we expect such a difference to exist, the relevant alternative (the null hypothesis, that there is no wage difference) is that well-educated individuals earn more ie that the average hourly wage of this group, \(\mu_Y) is greater than \(\mu_Y), the average wage for less skilled workers, which we assume is known here for simplicity (section @ref{cmfdp} discusses how to test the equivalence of unknown population assets). This is an example of a right-hand test, and the hypothesis pair \[H_0: \mu_Y = \mu_{Y.0} \\ text{vs} \ \ H_1: \mu_Y > \mu_{Y,0}. \] Reject the null hypothesis if the calculated test statistics are greater than the critical value \(1.64\), \(\0.95\)-quantile is the \(\mathcal{N}(0.1)\) distribution. This ensures that the probability mass \(1-0.95=5\%\) remains in the area to the right of the critical value. As before, we can imagine this r function using polygon(). # plots the standard normal density in the range [-4,4] curve(dnorm(x), xlim = c(-4, 4), main = Rejection region is a right-hand test, yaxs = i, xlab = t-statistics, ylab = , lwd = 2 axes = F) # axis of the x axis(1, at = c(-4, 0, 1,64, 4), padj = 0,5, tags = c(,0, expression(Phi^-1~(.95)==1.64),)) # shade the rejection region in the left tail polygon(x = c(1.64, seq(1.64; 4, 0.01), y = c(0, dnorm(seq(1.64, 4, 0.01)), col = darkred) Similar to the left test $[H_0: m_Y = m_{Y.0} \ text{vs.} \ h_1: m_Y & t; m_. Y.]$ Null rejects if the observed test statistics do not reach the critical value for the (0.05) precision level (-1.64), $(mathcal{N}(0.1))$ is the probability mass to the left of the critical value. The above piece of code can be easily adapted to the case of the left test. All we have to do is adjust the hue and tick marks. # plots the standard normal density in the range [-4,4] curve(dnorm(x), xlim = c(-4, 4), nadj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(, 0, expression(Phi^-1~(.05)===1.64),)) # shadow rejection region in right rear polygon(x = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4, 0, -1.64, 4), padj = 0,5, tags = c(-4 seq(-4, -1.. 64, 0.01), -1.64), y = c(0, dnorm(seq(-4, -1.64, 0.01)), col = darkred) Page 13 Of this book is open review. We want you to get feedback so that the cursor, and then clicking in the drop-down menu. You can also see other people's annotations by clicking on the item in the upper-right corner of the page, as we emphasized earlier, we will never estimate the exact value of the \(Y\) population by using a random sample. However, we can calculate the fitness intervals of the population on average. Typically, the trust interval of an unknown parameter is a recipe that results in repeated patterns with intervals that contain the real parameter with a predefined probability with the trust level. The maintenance intervals are calculated on the basis of the information average \(E(Y)\). The trust interval for a random variables. Key Concept 3.7 demonstrates how to calculate the continuum intervals of the unknown population average \(E(Y)\). The trust interval for $((mu_Y) ((mu_Y) (95)\%)$ is a random variable that contains ((mu_Y) trust intervals ((hu_Y) trust intervals a } (mu_Y) trust intervals ((hu_Y) trust intervals ((hu_Y) trust intervals a } (mu_Y) trust intervals \left[\overline{Y}) \right], \\ &\90\%\text{ trust a } számára mu_Y = \left[\overline{Y}) \right], \\ &\90\%\text{ trust a } számára mu_Y = \left[\overline{Y}) \right], \\ end align}\\ Ezek ezek intervals are sets of null hypotheses that cannot be rejected in a two-sided hypothesis at that confidence level. Now consider the following statements. In re-sampling, it contains the true value of \[\left] $v_{1,0} = 1.96 \times 1.96$ written test or similar, believe us. The difference is that while 1. is the definition of a random variable, 2. is one of the possible results of this random variable, so there is no point in making any probability statement about it. Either the calculated interval extends \(\mu_Y\) or it does not! In R, testing the hypotheses of the average population based on a random variable, so there is no point in making any probability statement about it. Either the calculated interval extends \(\mu_Y\) or it does not! In R, testing the hypotheses of the average population based on a random variable, so there is no point in making any probability statement about it. 10) # check the type of the outcome produced by t.test typeof(t.test(sampledata)) #> [1] list #display the list elements produced by t.test (sampledata)) #> [1] alternative conf.int data.name estimate method #> [6] null.value p.value parameter statistic stderr we find that many items are reported, at the moment we are only interested in computing a \(95\%\) confidence set for the mean. t.test(sampledata)\$conf.int #> [1] 9.306651 12.871096 #> attr(,conf.level) #> [1] 0.95 This tells us that the \(95\%) confidence interval is \[\left[9.31, 12.87\right]. \] In this example, the computed interval obviously does cover the true \(\mu_Y\) which we know to be \(10\). Let's take a look at the total standard output produced by t.test(). t.test(sampledata) #> #> Sample t-test #> #> data: sampledata #> t = 12,346, df = 99, p-value < 2.2e-16 #> alternative hypothesis: true average x #> average x #> 11.08887 sees that t.test() is not just \() 95\%\) instead of automatically executing the two-sided significance test of the \(H_0: \\mu_Y = 0\) hypothesis at the \(5\%\) level and reports its corresponding parameters: the alternative hypothesis, the estimated the resulting resulting the freedom level of the underlying \(t\) distribution (body t() performs the normal approximation) and the corresponding \(p\) value. It's very convenient! In this example, we conclude that the population average differs significantly from the \(0\) (which is correct) level \(5\%) because \(\mu_Y = 0\) is not an element of the \(95\%) control interval \[0 ot \in \left[9.31,12.87\right]. \] When you use the \(p\text{-value} = 2.2\cdot 10^{-16} \|I 0.05. \] Article 14 We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see other people's notes by clicking on the mark in the upper-right corner of the page, let's say you're looking for a top for two different populations, \(\mu_1\) and \(\mu_2\). Specifically, he's interested in the fact that these populations are different, and he plans to use a hypothesis test to check this based on
independent sample data from both populations. The correct pair of hypotheses \\begin{equation} H_0: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \text{vs.} \ \ H_1: \mu_1 - \mu_2 = d_0 \text{vs.} \text{vs. (H_0) can be tested with (t)-statistic $[\begin{equation} t=\frac{(overline{Y}_1 - overline{Y}_2) - d_0}(n_1) and (n_2), (3.7) \.7} end{equation} t=\frac{(overline{Y}_2) - d_0}(n_1) and (n_2), (3.7) \.7} end{equation} t=\frac{(overline{Y}_2)$ is the normal value according to the null hypothesis. Similarly to the simple \(t\)-test we can calculate trust intervals for the real difference in population meaning: \[(\overline{Y}_1 - \overline{Y}_1 - \overline{Y}_1 - \overline{Y}_2) \] is a \(95\%\) trust interval \(d\). In R, t.test() can also be used to test t.test() hypotheses. Note that t.test() selects \(d_0 = 0\) by default. You can change this mu argument by setting it accordingly. The subsequent chunk of code data shows how to perform two sample_pop1 <- rnorm(100, 10, 10) sample_pop2 <- rnorm(100, 10, 20) # two samples t-test t.test(sample_pop1, sample_pop2) #> #> 6.028083 #> sample estimate: #> #05 percent confidence interval: #> atternative hypothesis: the true difference between devices is not equal to the #> #243838 average We see that the two samples \(t\)-tests do not reject the (true) null hypothesis that \(d_0 = 0\). Page 15 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can also see the notes: click on the upper right corner of the page This section describes how to reproduce the results presented in the book is replaced by the following: This file contains information between \(1992\) and revenue is reported at \(2008\) prices. There are .xlsx ways to import files that are not in your file. Our recommendation is the readxl package (Wickham and Bryan 2019) read excel() function. The package is not part of the basic version of R and must be installed manually. # load the 'readxl' package library(readxl) You are now ready to import the dataset. Make sure you use the correct path to import the downloaded file. In this example, the file is placed in the subfolder of a working directory called data. If you're not sure what your current work directory is, use getwd(), see also ?getwd. This will give you the way to the location where R is currently searching for files. # import data from R cps <- read_excel(path=data/cps_ch3.xlsx) = next,= install= and= load= the= package= dyplr= (wickham= et= al.= 2020).= this= package= provides= some= handy= functions= that= simplify= data= wrangling= a= lot.= it= makes= use= of= the= %=>% operator. # load the dplyr packages to group observations by gender and year, and to calculate descriptive statistics for both groups. # get an overview of the data structure head(cps) #> # A tibble: 6 x 3 #> 6 2 1992 12.2 # group data by gender and year and compute the mean, standard deviation # and number of observations for each group avgs <- cps= %=>% group by(a sex, year) %>% summarise(mean(ahe08), sd(ahe08), n()) # print the results to the console print(avgs) #> # A tibble: 10 x 5 #> 2004 25.1 12.0 1894 #> 5 1 2008 25.0 11.8 1838 #> 6 2 1992 20.0 7.87 1368 #> 6 2 1992 20.0 7.87 1368 #> 7 2 1996 19.0 7.95 1230 8 2 2000 20.7 9.36 1181 #> 9 2 2004 21.0 9.36 1181 #> 8lt;/dbl> </dbl> </dbl> are compatible with input and output. In the code above we take the dataset cps and use it as an input to the function group_by(). The group_by of the data is then used as follows in the summary() and so on. Now that we have calculated statistics on the interests of the two sexes, we can look at how the income gap between the two groups develops over time. # split the dataset by gender male <- avgs %>% dplyr::filter(a_sex == 1) female <-avgs %>% dplyr::filter(a_sex == 2) # átnevezésoszlopok mindkét hasít névszerinti keresztnév(férfi) <- c(Sex, Év, Y_bar_f, s_f, n_f) # becsülje meg a nemek közötti különbségeket, számolja ki a standard hibákat és a kondinkettintervallumokat az összes dátumrésre <- férfi\$Y_bar_m - női\$Y_bar_f gap_se <- sqrt (férfi\$s_m^2 / férfi\$n_m + női\$s_f^2 / female\$n_f) gap_ci_l <- rés + 1,96 * gap_se gap_ci_l, gap_ci_l, gap_ci_v # nyomtassa ki az eredményeket a konzol nyomtatására (eredmény , számjegyek = 3) #> Év Y_bar_m s_m n_m Y_bar_f s_f n_f különbség gap_se gap_ci_l gap_ci 0,354 3,41 4,80 Megfigyeljük gyakorlatilag ugyanazt az eredményt, mint a könyvben bemutatottak. The calculated statistics suggest that there is a gender gap between earnings. Remember that you can reject the null hypothesis that the gap is zero in all periods. Furthermore, the estimation of the gap and boundary of the continuum intervals \(95\%\) mimicked the fact that the gap has been fairly stable recently. Page 16 of this book is the Open Review. We want you to get feedback so that the book is better for you and other students. You can also view annotation of others by clicking in the upper-right corner of the page, the scatter chart represents twodimensional data, such as \(n\) observations by \(X_i\) and \(Y_i\) in the coordinate system. It is very easy to generate scatter plots with the plot() function in R. Create and visually create artificial data on the age and income of workers. # set random seed set.seed(123) # generate dataset X & lt;-runif(n = 100, min = 18, max = 70) Y & lt;- X + rnorm(n=100, 50, 15) # plot observations plot(X, Y, type = p, main = The Scatterplot Of X and Y, xlab = Age, ylab = Income, col = steelblue, pch = The sample shows a positive correlation between age and income. This is in line with the idea that older workers earn more than those who have recently joined the working population. By now, you should be aware of the variance and covariance. If not, we recommend that you work your way through Chapter 2 of the book. Like the variance, covariance, and correlation of two variables, the properties associated with the (unknown) distribution of these variables. You can estimate the covariance, and correlation by using an appropriate estimate using a \((X_i,Y_i)), \(i=1,\dot,n\) pattern. Sample covariance \[s_{XY} = \frac{1}{n-1} \sum_{i=1}^n (X_i - 1) (X) and (Y) is standardized. For a more detailed description of the sample correlation, the strength of the linear relationship between (X) and (Y) is standardized. For a more detailed description of these estimates, see section 3.7 of the book. As far as variance and standard deviation are concerned, these estimates are implemented as R functions in the statistics package. We may use them to estimate the covariance X and Y cov(X, Y) #> [1] 213.934 # calculation pattern covariance A and Y cor(X, Y) #> [1] 0.706372 # equivalent way of sample correlation cov(X, Y) / (sd(X) * sd(Y)) #> [1] 0.706372 Estimates indicate that \(X\) and \(Y\) The following set of code data uses the mvnorm() function of the MASS (Ripley 2020) package to create two-variable sample data with different correlations. library(MASS) # set random seed set.seed(1) # positive correlation (0.81) example 1 & lt;- mvrnorm(100, mu = c(0,0), Sigma = matrix(c(2, 2, 2, 3, ncol = 2), empirical = TRUE) # no example 3 & lt;- mvrnorm(100, mu = c(0, 0), Sigma = matrix(c(1, 0, 0, 0, 1), ncol = 2), empirical = TRUE) # no correlation (second-degree relationship) X & lt;- seq(-3, 3, 0.01) Y & lt;- X^2 + (3, 3, 0.01) Y & lt;- X^2 + (3, 3, 0.01) Y & lt;- Seq(-3, 3, 0.01) Y & lt;- X^2 + (3, 3, 0.01) Y & lt;- X^2 + rnorm(length(X)) example4 <- cbind(X, Y) # divide the plot area as 2-x 2 blocks par(mfrow = c(2, 2)) # parcel data sets plot(example1, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example2, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col =
steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plot (example3, col = steelblue, pch = 20, xlab = Y, main = Correlation = 0.81) plo = 0) plot(example4, col = steelblue, pch = 20, xlab = X, ylab = Y, main = = 0 Correlation Page 17 This book is in an open overview. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also view comments from others by clicking in the upper-right corner of the page The following alternative estimate is \(\mu_Y), the average \(Y_i) \[/widetilde{Y}=\frac{1}^n Y_i\] In this exercise, we demonstrate that this estimate is a biased estimation of \(\mu_Y). Instructions: Enter a function that Y_tilde that implements the above estimate. Randomly draw 5 observations from the \(\mu_Y), the average \(Y_i) \[/widetilde{Y}=\frac{1}^n Y_i\] In this exercise, we demonstrate that this estimate is a biased estimation of \(\mu_Y). Instructions: Enter a function that Y_tilde that implements the above estimate is a biased estimate i calculate an estimate using Y_tilde(). Repeat this procedure 10,000 times and store the results est_biased. Draw a histogram est_biased. Draw a histogram est_biased. Add a red vertical line to line \(\mu=10\) using abline(). Tips: To calculate the sum of a vector, you can use the length(). Use the replical() function to calculate a re-estimate of random samples. You can specify the action and how often the arguments should be replicated. The histogram can be represented by the hist() function. Use the v and column arguments to specify the point of the x axis and the color of the vertical line. Re-estimate the previous exercise. Available as Y tilde() feature in your environment. The system will ask for the same procedure as in the previous exercise. Available as Y the number of observations from 5 to 1000. What did you notice? What can you tell me about this antler? Instructions: At random, 1,000 observations shall be calculated from the Y_tilde(). Repeat this procedure 10,000 times and store the results est_consistent. Draw a est_consistent histogram. Add a red vertical line to line \(\mu=10\) using abline(). Tips: Use function replication() to calculate estimates of repeatedly subtracted random samples. You can use the kif and n arguments to specify the position of the x axis and the color of the vertical line in the v. and column arguments. In this exercise, we want to illustrate that the pattern also means that (i=1,...,n) a_iY_i limits_Y has the same weighting scheme $(a_i=\frac{1}{n})$ with the same linear unbiased estimate (BLUE) ($(\sum_{i=1}^{n})$ with the same linear unbiased estimate (BLUE) ($(\sum_{i=1}^{n})$ with the same weighting scheme ($a_i=\frac{1}{n})$ with the same weighting scheme ($a_i=\frac{1}{n}$) with the same linear unbiased estimate (BLUE) of (i=1,...,n) a_iY_i limits_(i=1,...,n) a_iY_i limits_(i=1,...,n a_iY_i limits_(i=1,...,n) a_iY_i limits_(i=1,...,n a_iY_i limits_(i=1,...,n) a_iY_i limits_(i=1,...,n a_iY_i limi where \(b_i\) the first \(\frac{n}{2}\) observations give a greater weighting than the second \(\frac{n}{2}\) observations (assume that \(n\) even for simplicity). The w weight vector is already defined and available in your work environment. Instructions: Verify that \(\tilde{\mu}\) is an unbiased ticing of \(\mu_Y\) \(\mu_Y\) mean of \(Y_i\). The alternative mu Y \(\mu Y\) is mu tilde(). A random 100 observations shall be 10)\) distribution and calculation estimates with both estimates. Repeat Stores the results 10,000 times est bar and est tilde. Calculate the sample est bar and est tilde of the product. What can you tell me about both estimates? Tips: For \(\tilde{\mu}\) to be an unbiased estimate, each weight must sum up to 1. Use the replical() function to calculate estimates of re-drawn patterns. You can specify the action and how often the arguments are replicated. You can use the pattern variance of the var(). Revisit the CPS dataset cps is available in the work environment. We assume that the average hourly wage (at 2012 prices) ahe12 exceeds 23.50 \(\\$/h\) and you want to try this hypothesis at

a significant level \(\alpha=0.05\). Please do the following: Instructions: Manually calculate and assign test statistics to tstat. Use tstat to accept or reject the null hypothesis. Please do this using the normal approach. Tips: Test \(H_0:\mu_{Y_{ahe}}) and \(H_1:\mu_{Y_{ahe}}) and \(H_1:\mu_{Y_{ahe}}) and \(H_1:\mu_{Y_{ahe}}) and \(H_0:\mu_{Y_{ahe}}) and \(H_0:\mu_ \mu_{Y,0}}(s_{Y}) and represent the pattern variance\(s_Y)). To determine whether the null hypothesis is accepted or rejected, you can compare \(t\) statistics with the corresponding quantile of the standard normal distribution. Use logical operators. Rethink the test situation of the previous exercise. The dataset cps as well as vector tstat are available in the work environment. You can also use \(p\) instead of using \(t\) statistics as a decision condition. Now do the following: Instructions: Calculate the \(p\) value of the right test is \(p=P(t>t^{act}| H_0)\). Rejects null if \(p<\alpha\). Use logical operators to verify this. In the last two practices, we discussed two ways to perform a hypothesis test. These approaches are a little cumbersome to use manually, so R provides ((t)) statistics, \(p)) values, and consistent confidence intervals (more on the latter in a series of subsequent exercises). Note that t.test() uses the \(t\) distribution instead of the normal distribution, which becomes important when the sample size is small. The dataset from exercises using the function t.test(). Expand the \(t\) statistics and \(p\)-value from the list created by t.test(). Assign them to the tstat and pvalue variables. Make sure that the normal approach calculation of the difference between the two \(p\) values. Advice: The type of test and the null hypothesis can be specified through alternative and mu arguments. The \(t\) statistics and \(p\)values can \$statistic and \$p.value. Consider the annual maximum sea levels in Port Pirie (South Australia) and Fremantle (Western Australia) over the past 30 years. Observations are available in the working environment as portpirie and fremantle vectors. Instructions: Check for significant differences in the significance level of \(\alpha=0.05\) between the maximum annual sea level. Tips: Test \(H_0:\mu_{F}=0\) and \(H_1:\mu_{F}=0\) and \(H_1:\mu_{F}=0\) and \(H_1:\mu_{F}=0\). sample \(t\) tests, the t.test() function calculates two vectors that contain the data. A rethink of the highest annual sea level permitted test in Port Pirie and Fremantle. The portpirie and fremantle variables are once again available in the working environment. Instructions: Use t.test() to create a \(95\%\) up-to-sea adjustment interval for sea level difference. Tip: By default, the t.test() function calculates a trust interval of \(95\%\). It is available via \$conf.int. Take a random sample \((X). Calculate the covariance \(X\) and \(Y\). Calculate the covariance \(X\) and \(Y\). covariance. cov() and cor() are expected to have vectors for each variable. In this exercise, we would like to examine the limitations of correlation as a dependency measure. Once the session is initialized, you will see plot 100 implementations with two random variables (X) and (Y). Appropriate observations are available in the X and Y vectors in the working environment. Instructions: Calculate the correlation between \(X\) and \(Y\). Interpret the result critically. Tip: cor() castle vector for all variables. Page 18 This book is the Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also view comments from others: click in the upper-right corner of the page, click this chapter to show you the basics of linear regression and show you how to perform regression and show you how to perform regression, the goal is to model the relationship between a dependent variable \(Y\) and one or more explanatory variables\(X_1, X_2, \dots, X_k\). Following the book, we will focus on the concept of simple linear regression throughout the chapter. For simple linear regression, there is only one explanatory variable \(X_1). For example, if a school reduces the number of classes by recruiting new teachers, i.e. the X_1 reduces the number of classes, how does this affect \(Y), the performance of students taking a standardized test? With linear regression, we can examine not only whether the teacher-teacher ratio affects test results, but also the direction and strength of this effect. To reproduce the code described in this chapter, the following packages are required: AER - accompanied by book Applied Econometrics with R Kleiber and Zeileis (2008) and provides useful features and datasets. MASS - a collection of functions of applied statistics. Make sure that they are installed before you advance and try to replicate the examples. The safest way to do this is to verify that the next chunk of code executes without errors. Library (AER) (MASS) Kleiber, C., and A. Zeileis. 2008. Applied econometrics with I R. Springer. Page 19 of this book is the Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see other people's comments by clicking the button in the upper-right corner of the page To get started with a simple example, consider the following combinations of average test score and average student-teacher ratio in some fictional school districts. To work with this data in R, we start by creating two vectors: one for student-teacher ratios (STR) and one for test score (TestScore), both of which contain data from the table above. # Create sample dataSTR & It;- c(680, 640, 670, 660, 630, 660, 630, 660, 630) # Print sample data STR #> [1] 15.0 17.0 19.0 20.0 22.0 23.5 25.0 TestScore #> [1] 680 640 670 660 660 635 To create a simple linear regression model, we assume that the relationship between the dependent variables is linear, formally: \[Y = b \cdot X + a. \] Now suppose that the test score and the student-teacher ratio are related functions \[TestScore = 713 - 3 \3 times STR.\] It is always a good idea to display work data. Here you can use plot() to str on the \(x\) axis and TestScore on the \(y\) axis. Just call sites (y_variable and x_variable the vectors for observations you want to plot. Furthermore, you may be added to the systematic relationship with the site. To draw a straight line, R provides the abline() function. You just need to call this function with arguments (which represents the capture) and b (represents the slope) after executing the plot() to add the line to the plot. The following code in the textbook 4.1. # create a scatterplot of data sites(TestScore ~ STR) # add a systematic link. The reason for this is randomness. Most of the time, there are additional effects that mean that there is no two-variable relationship between the two variables. In order to take into account these differences between the observed data and the systematic relationship between the differences between the differences between the two variables. In other words, \(u\) accounts show the differences between the two variables. between the regression line and the actual observed data. In addition to pure randomness, these differences may also result from measurement variable. What other factors are plausible in our example? Firstly, test results can be driven by the quality of teachers and the background of students. It is also possible that in some classes, students were lucky on test days and thus achieved higher scores. For now, we will summarize these effects with an additive component: \[TestScore = \beta_0 + \beta_1 \times STR + \text{other factors} \] Of course, this idea is very general, as it can be easily extended to other situations that can be described with a linear model. The basic linear regression model with which we will work is \[Y_i = \beta_0 + \beta_1 X_i + u_i.] Key Concept 4.1 summarizes the terminology of the simple linear regression model. Linear regression model is of type \[Y_i = \beta_0 + \beta_1 X_i + u_i.] where the index \(i\) runs through observations, \(i=1,\dot,n\) \(Y_i) is the dependent variable, the regression variable, or simply the left variable \(X_i\) is the independent variable, the regressive variable, or simply the right variable \(Y = \beta_0 + \beta_1 X\) is the population regression function \(\beta_0\) is the population regression line \(\beta_0\) is the population regression line \(\beta_0\) is the population regression function \(\beta_0\) is the population regression line \(\beta_0\) is the population regression line \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X\) is the population regression function \(\beta_0 + \beta_1 X + \b regression line. Page 20 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can also see other people's comments by clicking in the upper-right corner of the page In practice, the capture of the population regression line \ (\beta_0\) and slope \(\beta_1\) are unknown.
Therefore, we need to use data to estimate both unknown parameters. In the following, we will use a real example to show how this can be achieved. We want to link test results to et student-teacher ratios measured in California schools. The test score is the district-level average of reading and math scores for fifth graders. Again, the size is measured by the number of students number of teachers (teacher-teacher ratio). As for the data, the California School Dataset (CASchools) also includes an R package called AER, which stands for Applied Econometric R (Kleiber and Zeileis 2020). After you install the package with the install.packages(AER) file and attach it to the library(AER) function(AER), you can use the function file() to fill the dataset. ## # install the AER package (once) ## install.packages(AER) ## # #load the AER package library() — there is no need to run install.packages() again! It's interesting to know what kind of object we're dealing with. class() returns the class of an object. Depending on the class of an object, some functions, such as plot() and summary()), behave differently. Let's look at the class of the object CASchools) #> [1] data frame It turns out that CASchools is class data.frame, which is a convenient format to work with, especially for regression analysis. With head(), we get the first overview of the data. This feature displays only the first 6 lines of the dataset that prevent overcrowded console. The good news is, everything else stays intact. You have neither loose defined variables, etc., nor code history. You can still use the up and down keys to call back R commands that you performed earlier. If you're working in RStudio, press ctrl + Up on the keyboard (CMD + Up on a Mac) to review the list of commands that you performed earlier. If you're working in RStudio, press ctrl + Up on the keyboard (CMD + Up on a Mac) to review the list of commands you've typed before. head(CASchools) #> 175119 Sunol Glen Unified Alameda KK-s 08 195 10.90 #> 2 61499 Manzanita Elementary Butte KK-08 240 11.15 #> 3 61549 Thermalito Union Elementary Butte KK-08 135 71.50 #> 4 61457 Golden Feather Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 4 61457 Golden Feather Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 5 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 6 62042 Burrel Union Elementary Butte KK-08 135 71.50 #> 6 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 6 61523 Palermo Union Elementary Butte KK-08 135 71.50 #> 6 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-08 140 #> 7 61523 Palermo Union Elementary Butte KK-0 expenditure income english read math #> 1 0.5102 2.0408 67 6384.911 22.690001 0.000000 691.6 690.0 #> 2 15.4167 47.9167 101 5099.381 9.824000 4.583333 660.5 661.9 #> 3 55.0323 76.3226 169 5501.955 8.978000 30.000002 636.3 650.9 #> 3 55.0323 76.3226 169 5501.955 8.978000 30.000002 636.3 650.9 #> 3 55.0323 76.3226 169 5501.955 8.978000 30.000002 636.3 650.9 #> 4 36.4754 77.0492 85 7101.831 8.978000 0.000000 651.9 643.5 #> 5 33.1086 78.4270 171 5235.988 9.080333 13.857677 641.8 639.9 #> 6 12.3188 86.9565 25 5580.147 10.415000 12.408759 605.7 605.4 We find that the data set consists of plenty of variables and provides a comprehensive overview of the object. Try! Try! back to the CASchools, the two variables we are interested in (i.e. average test score and student-teacher ratio) are not included. However, both can be calculated from the specified data. To get teachers. The average test score is the arithmetic mean of the reading test score and the math test score. The following chunk of code data shows how to create the two variables as vectors and append them to CASchools \$ score <- (CASchools \$ read + CASchools \$ students / CASchools \$ students / caschools \$ read + caschools \$ read + caschools \$ students / caschools \$ read + caschools \$ students / caschools \$ students / caschools \$ read + caschools \$ read + caschools \$ students / caschools \$ students / caschools \$ read + caschools \$ read + caschools \$ students / caschools \$ students / caschools \$ students / caschools \$ students / caschools \$ read + caschools \$ students / caschools \$ students / caschools \$ students / caschools \$ read + caschools \$ students / casch columns called STR and score (check this!). Article 4.1 of the textbook shall be replaced by the following: There are a number of functions that can be used to produce similar results, e.g. average() (calculates the arithmetic mean of the sample specified for the data). The following: code section shows how to access this. First, we calculate summary statistics about str columns and CASchools score. In order to get nice output we collect the actions of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary. # compute sample standard deviations of data.frame called DistributionSummary.# compute sample STR and score sd_STR <- sd(CASchools\$STR) sd_score <- sd(CASchools\$STR, quantiles) # set up a vector of percentiles and compute the quantiles (CASchools\$STR, quantiles) # gather everything in a data.frame DistributionSummary <- (0.10, 0.25, 0.4, 0.5, 0.6, 0.75, 0.9) quant_STR <- guantile(CASchools\$STR, quantiles) # gather everything in a data.frame DistributionSummary <data.frame(Average = c(avg_STR, avg_score), StandardDeviation = c(sd_STR, sd_score), quantile = rbind(quant_STR, quant_score)) # print the summary to the console DistributionSummary #> Average StandardDeviation quantile.10. quantile.25. quantile.40. #> quant_STR 19.64043 1.891812 17.3486 18.58236 19.26618 #> quant_score 654.15655 19.053347 630.3950 640.05000 649.06999 #> quantile.50. quantile.50. quantile.75. quantile.75. quantile.90. #> quant score 654.45000 659.4000 666.66249 678.85999 As for the sample data, we use plot(). This allows us to detect the characteristics of our data, such as outliers, which are more difficult to detect simply by numbers. This time we add some additional arguments for the call plot(). The first argument for our call plot(), score ~ STR, is again a formula that states variables of the y- and x-axis. This time, however, the two variables will not be placed in separate columns of CASchools. Therefore, R would not find them without entering the argument information correctly. the data must correspond to the name of the data.frame to which the variables belong, in this case CASchools.data. Additional arguments are used to change the appearance of the plot, while adding the main title adds custom tags to both axes by adding xlab and ylab. plot(score ~ STR, data = CASchools, main = TestScore and STR scatterplot, xlab = STR (X), ylab = Test score (Y)) The sample (figure 4.2 of the book) shows the scatterplot of all observations on the student-teacher ratio and the test scores in larger classes. The function cor() (see ?cor for more information) can be used to calculate the correlation between two numeric vectors. cor(CASchools\$STR, CASchools\$score) #> [1] -0.2263627 As scatterplot already suggests, the correlation and correlation and correlation and correlation and servers would draw different lines of regression. This account is interested in less arbitrary techniques. This technique is determined by the sum of square errors in \(Y\) \(X\) prediction. Be \(b 0\) and \(b 1\) some of the 1990s and 1990s with \(\beta 1\). Then the sum of the square estimate errors \[\[\sum^n_{i = 1} (Y_i - b_0 - b_1 X_i)^2. \] In the simple regression model, the OLS estimate is the pair of capture and slope estimates, which minimizes the above expression. The derived OLS estimates for both parameters are set out in 4.1 of the book. The
results are summarised in Key Concept 4.2. In the simple linear regression model, the ols detractor \(\beta_1) and the intersection \(\beta_1) + \verline{Y}) } { six\beta_1 & amp; = \rac{sum_{i=1}^n (X_i - \verline{Y}) } { six\beta_1 & amp; = \verline{Y} - \six\beta_1 & amp; = \verline{Y} - \six\beta_1 & amp; = \verline{Y} - \six\beta_1 & amp; = \verline{Y} + \verline{Y} + \verline{Y} - \six\beta_1 & amp; = \verline{Y} - \verline{Y} + \verline{Y} - \six\beta_1 & amp; = \verline{Y} - \verline{Y} + \verline{Y} - \six\beta_1 & amp; = \verline{Y} - \verline{Y} + \verline{Y} - \verline{Y} - \verline{Y} + \verline{Y} - \verline{Y} + \ve (n)). These unknown populations are an estimate of \(\left(\beta_0 \right)), \(\left(\beta_1\right)) and \((u_i)). The formulas shown above are not very intuitive at first glance. The next interactive application is designed to help you understand the mechanics of OLS. You can add observations by clicking on the coordinate system, where the data is marked with dots. If two or more observations are available, the application calculates the regression line at OLS and some statistics that appear in the right panel. The results will be updated as you add additional observations to the left pane. Double-clicking restores the app, which means all data is deleted. There are several ways to calculate \(\six{\beta}_0) and \(\six{\beta}_1) in R. For example, you could perform the formulas described in Key Concept 4.2 with two of R's most basic functions: average() and amount(). Before you attach this to the CASchools directly # calculation beta_1_hat beta_1 & lt;- amount((STR) * (score - average(score))) / amount((STR) * (score - average(STR)^2) # calculation beta_0_hat beta_0 <- average(score) - beta_1 * average(STR) # print the results in the console beta_1 #> [1] -2.279808 beta_0 #> [1] 698.9329 Calling attach(CASchools) allows us to use the variable in CASchools by its name: it is no longer necessary to use the \$ operator with the dataset: R can directly evaluate the variable name. R uses the object in the user context when the object shares the name of the variable in the linked database. However, it is better practice to always use distinguishing names to avoid such (seemingly) ambivalence! Notice that our title variables are contained in the attached dataset CASchools directly to the rest of this chapter! Of course, there are even more manual ways to perform these tasks. Since OLS is one of the most widely used estimation techniques, R of course already includes a built-in function called Im() (linear model) that can be used to perform regression formula, which has a baseline y ~ x, where y is the dependent variable, and x is the explanatory variable. Argument data determines the dataset to be used in regression. Now revisit the example from the book where the relationship between test scores and class size is analyzed. Use the following code Im() to replicate the book 4.3. # estimate the model and assign the result to the linear_model & linear_model console, #> #> H(xás: #> H(xás: #> In(képlet = score ~ STR, data = CASchools) #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> #> STR (X), ylab = Test Score (Y), xlim = c(10, 30), ylim = c(600, 720)) # add the regression line abline(linear_model) Have you noticed that this time we did not pass the capture and slope parameters abline? If abline() calls an object of class lm that contains only one regression, R automatically draws the regression line! Page 21 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also view other people's notes by clicking on the upper-right corner of the page After installing the linear regression model, it is a natural question how well the model describes the data. Visually, this is used to assess whether observations are tightly grouped around the regression line. Both the determination coding and regression standard errors measure how well the OLS regression line fits the data. \(R^2\), the codification coding is explained by the \(Y_i\) sample variance X_i\). Mathematically, \(R^2\) is written as the ratio between the explained sum of the squares and the total sum of the squares. The explained sum of the squares (\(ESS\)) is the sum of the square deviations from the (Y_i) average of the predicted values of $((Y_i)$ and the average squared differences. That's right ($six{Y_i} - six{Y_i}$) is the sum of the squares ((TSS)) is the sum of the square deviations from the (Y_i) average of the predicted values of ($(Six{Y_i}) - six{Y_i}$). The total sum of the squares ((TSS)) is the sum of the squares ((TSS)) is the sum of the squares ((TSS)) is the sum of the square deviations from the (Y_i) and the average squared differences. That's right ($six{Y_i} - six{Y_i} - six{Y_i$ regression line, means \(R^2 = 1\) since then \(SSR=0\). On the contrary, if our estimate of the standard deviation of residues (\\six{u}_i\). (ESS=0\) and consequently \(R^2=0\). The standard regression line, i.e. the size of the typical deviation from the regression line, i.e. the size of the standard deviation of residues (\\six{u}_i\). (ESS=0\) and consequently \(R^2=0\). a typical residue. $[SER = s_{hat{u}} = s_{$ summary() a function with an Im object specified as a single argument. While the Im() function prints only the estimated co-values on the console, summary() provides additional predefined information, such as regression \(R^2\) and \(SER\). mod_summary() provides additional predefined information, such as regression \(R^2\) and \(SER\). #> Leftovers: #> 1Q Median 3Q Max #> -47.727 -14.251 0.483 12.822 48.540 #> #> Coding: #> Estimate Std. Error Value Pr(>|t]) #> (Intercept) 698.9329 9.4675 73.825 < 2e-16 *** #> STR -2.2798 0.479 8 -4.751 2.78e-06 *** #> STR -2.2798 0.479 8 -4.751 2.78e-06 *** #> Signif. codes: 0 '***' 0.001 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.01 '*' 0.0 Multiple R-squared: 0.05124, Adjusted R-squared: 0.05124, Adjusted R-squared: 0.04897 #> F-statistics: 22.58 on 1 and 418 DF, p-value: 2.783e-06 Output \(R^2\) name Multiple R-squared, value \(S.1 \%\) is explained by the explanatory variable \(S.1 \%\). many of the changes in test scores remain unexplained (cf. Figure 4.3 of the book). \(SER\) is called Residual Standard Error and is set to \(18.58\). The unit of \(SER\) is the same as the unit of measure of the dependent variable. In fact, the difference between the actual test score and the regression line is an average of \(18.58\) points. Now make sure that summary() uses the same definitions for \(R^2\) and \(SER\) as when you use them manually. # calculation R^2 manually SSR <-sum((score - mean(score))^2) R2 <-sum((score - mean(score))^2) R2 <-sum((score - mean(score))^2) R2 <-sum((score - mean(score))^2) R2 <-sum((score - mean(score))^2) R2 <-sum(smod_summary remaining^2) TSS <-sum((score - mean(score))^2) R2 <-sum(smod_summary remaining^2) TSS <-#> [1] 18.58097 We find that the results coincide. Note that the values specified by summary() are rounded to two decimal places. Page 22 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can also see the notes of others: click on
the upper right corner of the page OLS performs well in a fairly wide variety of different circumstances. However, there are assumptions that need to be met to ensure that estimates are (this is done in large samples in u_i beta_1 X_i beta_0 Y_i 4.5 Y_i). where \(u_i \) returns zero \(X_i, Y_i), i = 1,\dot,n\) are independent and have the same distribution (i.e.d.) from their common distribution. Large outliers are unlikely: \(X i\) and \(Y i\) do not have zero finite fourth moments. This means that no matter which value you select for \(X\), the error expression \(u\) cannot show a systematic pattern and must show an average of \(0\). For example, consider the case that the error expression is usually positive for \(E(u) = 0\, but for low and high values of \(X\), the error expression is usually positive, and for the mean range values of \(X\), the error is usually negative. We can use the R to make such an example. To do this, we create our own data using R's built-in random number generators. We will use the following functions: runif() - creates evenly distributed random numbers rnorm() - usually generates distributed random numbers forecast() - adding model join functions such as Im() lines() - line segments to an existing plot A vector that creates values that are evenly distributed in the \([-5,5]\) interval. This can be done by using the runif() function. You also need to simulate the error expression. To do this, we normally create distributed random numbers with\(0\) average and \(1\) variance using rnorm(). \(Y\) values are given as a second-degree function of \(X\) and error. After the data is created, we estimate both the simple regression model, see Chapter 6). Finally, we plot the simulated data and add the estimated regression line of a simple ession model, as well as the projections made with the second-degree model, to graphically compare the fit. # set a seed to make the results reproducible set.seed(321) # simulate the data X <- runif(50, min = -5, max = 5) u <- rnorm(50, sd = 1) # the real connection Y <- X^2 + 2 * X + u # estimate a simple regression model mod_simple <- lm(Y ~ X) # predict a second degree model forecast<- predict(y~X)lm(Y ~ X + I(X^2)), data.frame(X = sort(X))) # plot representation of the results plot(Y ~ X) abline(mod_simple, col = red) lines(row(X), prediction) The pattern shows what you mean by \(E(u_i|X_i) = 0\) and why you do not have a linear model: Using the second-degree model (represented by the black curve), we see that the observation has no systematic deviations from the predicted relationship. It is credible that the assumption is not infringed if such a model is used. However, by using a simple linear regression model, we see that the assumption is not infringed if such a model is used. However, by using a simple linear regression model is used. samples. For example, you can use R's random number generator to randomly University enrollment list IDs and age \(X\) and earnings and ensures that all \((X_i, Y_i)\) are randomly from the same population. A prominent example where assumption i.i.d. is not met is time series data, where observations for the same unit are made over time. For example, take \(X\) as the number of workers in the production company over time. Due to business transformations, the company regularly reduces jobs with a specific shareholding, but there are also non-deterministic effects that can be easily simulated and planned with the economy, politics, etc. Let's start the series with a total of 5,000 workers and simulate a reduction in employment through an autoregressive process that shows downward movement in the long run and is usually distributed errors: 4 \[employment_{t-1} u_t \] # set seed set.seed(123) # creates a date vector date & lt;- seq(as. Date(1/1/1951) as. Date(2000/1/1), years) # initialize the employment vector X <- c(5000, rep(NA, length(Date)-1)) # generate time series observations with random influences for (i in 2:length(Date)) { X[i] <- -50 + 0.98 * X[i-1] + rnorm(n = 1, sd = 200) } #plot results plot(x = date, y = X, type = I, col = steelblue, ylab = Workers, xlab = Time) It is clear that the observations of the number of employees can not be independent in this example: the level of today's employment correlated with tomorrow's employment level. Thus, the I.I.D. assumption was violated. It is easy to come up with situations where extreme observations can occur, i.e. observations that differ significantly from the usual range of data. Such observations are called outliers. Technically, article 3 (1) (a) is replaced by the following Even if it seems like extreme observations have been recorded correctly, it is advisable to exclude them before estimating the model, since OLS suffers from sensitivity spikes. What does that mean? It can be shown that extreme observations are given heavy weighting in the estimation of unknown regression co-ordnables when using OLS. Therefore, outliers can lead to highly distorted estimates of regression co-values. For a better impression of the problem, consider the following application, where some sample data has been placed in the \(X\) and \(Y\) seems to be explained quite well by the regression line plotted: all white data points are close to the red regression line and \ (R^2=0.92\). Now go ahead and add an additional at\((18,2)\. This observation is clearly an outlier. The result is quite striking: the estimated regression line is very different from what we judged according to the data. The slope is strongly distorted downwards and \(R^2\) is simple \(29\%\)!! Double-click within the coordinate system to reset the application. Feel free to experiment. Select different coordinates for the outlier or add more. The following code roughly reproduces what is shown in Figure 4.5 in the book. As described above, we use sample data created using R rnorm() and runif(). We estimate two simple regression models, one based on the original dataset and the other a modified set where one observation changes as an outlier and then plots the results. To understand the full code, you need to know the function sequence() that sorts entries in a numeric vector in ascending order. # set seed set.seed(123) # generate the data X & t;- line(runif(10, min = 30, max = 70)) Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, max = 70), Y & t;- norm(10, min = 30, m 9]) # site of results site(Y ~X) abline(fit) abline(fitWithoutOutlier, col = red) Page 23 This book is open review. We want you to get feedback so that the book is better for you and other students. You can also view comments from others by clicking on the one in the upper-right corner of the page, because \(\six\\beta]_0\) and \(\six\beta]_1\) are calculated from a sample, the estimates themselves are random variables that can be received on different samples. Although the sampling distribution of \(\six\beta_0\) and \(\six\beta_1\) can be complicated if the sample size is small and usually changes with the number of observations, \(n\), if the assumptions discussed in the book are valid, it is possible to make certain statements about it that apply to all \(n). In particular\[E(\six\\beta_1) = \beta_0 \ \ \text{and} \ E(\six\\beta_1) = \beta_0 \ \ \text{and} \ E(\six\\beta_1) = \beta_0 \ \ \text{and} \ E(\six\\beta_1) = \beta_1,] i.e. \(\six\beta_1) are \(\beta_1) are \(\beta_1) are \(\beta_1), if the assumptions discussed in the book are valid, it is possible to make certain statements about it that apply to all \(n). In particular\[E(\six\\beta_1) = \beta_0 \ \ \text{and} \ E(\six\\beta_1) = \beta_1,] i.e. \(\six\beta_1) are \(\beta_1) are \(\beta_1) are \(\beta_1), are \(\beta_1) are \(\beta_1) are \(\beta_1), are \(\beta_1) are \(\beta_1), are \ unbiased estimates. If the sample is large enough, for the central limit batch, the common sampling distribution of \(\six\beta_0\) and \ (\six\beta 1\). If key concept 4.3 least square assumptions are held down, the large samples \(\six\beta 1\), and common common sampling distribution. A \(\hat\beta 1\)), and a disztribúció varianciája\(\sigma^2 {\hat\beta 1\}), and common sampling distribution. A \(\hat\beta 1\)), and common sampling distribution. A \(\hat\beta 1\), and common sampling
distribution. A \(\hat\beta 1\)), and common sampling distribution. A \(\hat\beta 1\), and common sampling distribution. A \(\hat\beta 1\)), and common sampling distribution. A \(\hat\beta 1\), and common s $l(hat|beta_0) | (\beta_0, \gua^2_{\beta_0} | ent[NX_i|tag{4.1} ent[align] | A ((hat|beta_0)) | (beta_0, \gua^2_{\beta_0} | ent[Align] | A ((hat|beta_0)) | (beta_0, \gua^2_{\beta_0} | ent[Align] | en$ interaktiv szimuláció folyamatosan véletlenszerű mintákat hoz létre \((X) i a \(200\) megfigyelések Y i \(E(Y\vert X) = 100 + 3X\) \((200\) megfigyelések Y i \(E(Y\vert X) = 100 + 3X\) becslése egyszerű regressziós modellt, stores the slope estimate \(\\beta 1\) and displays the distribution of the \(\widehat{\beta} 1\) observed so far using a histogram. The idea here is that for many \ (\widehat{\beta}_1\)s, the histogram is a good approximation of the sampling distribution of the estimate value. By reducing the time between two sampling repetitions, it becomes clear that the histogram to restart the simulation.) You can also check whether Key Concept 4.4's claims really hold up by using R. To do this, we first build our own population \(100000\) observations in total. To do this, you need values for \(X\), \(u\) and \(\beta_1\),\(100000\) by taking a random sample of a uniform distribution at the \([0,20]\) interval. The realization of \(u i\) error expressions is derived from the standard normal distribution with \(\sigma\) as the input to the sd argument, see ?rnorm). In addition, \ (beta_0 = -2)) and \(beta_1 = 3.5)) were selected, so that the real model \[$Y_i = -2 + 3.5 \ X + u$ population regression Y <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(N, min = 0, max = 20) u <- runif(real population (which, of course, would be unknown in a real application, otherwise there would be no reason to take a random sample). Knowledge of the real population and the real ink between \(Y\) and \(X\) key concept 4.4. First, calculate the true variances of \(\sigma^2_{\\six}\beta_0}) and \(\sigma^2_{\\six}\beta_1)) to a randomly drawn size \(n = 100\). # set sample size n <- 100 # a beta_hat_0 H_i <- 1 - center(X) / average (X^2) * X var_b0 <- var(H_i * u) / (n * average(H_i^2) ^2) # a hat_beta_1 var_b1 < variance variance calculation (X - mediocre (X)) * u) / (100 * v(X)^2) # console var_b0 #> [1] 4.045066 var_b1 #> [1] Print 0.03018694 # on the console with the true values of \(\beta_0\) and \(\beta_1\) and that the entire population (X - mediocre (X)) * u) / (100 * v(X)^2) # console var_b0 #> [1] 4.045066 var_b1 #> [1] Print 0.03018694 # on the console with the true values of \(\beta_0\) and \(\beta_1\) and that the entire population cannot be observed. However, you can observe a random pattern of \(n\) observations. You can't then calculate the true parameters, but you can use OLS to get \(beta_1\) estimates. However, we know that these estimates are the results of the random variables themselves, as observations are randomly sampled from the population. Key Concept 4.4 describes the large \(n\) split. You cannot make a statement about these divisions when drawing a single pattern the size of \(n\). Things change when you repeat the sampling pattern many times and calculate the stimates for each sample by simulating the results of the corresponding distributions. To achieve this R, we take the following approach: We added the number of repetitions, say \(10000\) to reps, and then initialize the matrix fit that was obtained from estimates of each sample. Results are stored as line entries in the matrix. This is done \(10000\) to reps, and then initialize the matrix fit that was obtained from estimates of each sample. Result matrix. This is done \(10000\) to reps, and then initialize the matrix fit that was obtained from estimates for each sample. Result matrix. This is done \(10000\) to reps, and then initialize the matrix fit that was obtained from estimates of each sample. Result matrix. This is done \(10000\) to reps, and then initialize the matrix fit that was obtained from estimates of each sample. Result matrix. using a for() loop. Finally, the variance of both estimates is estimated on the basis of the results in the sample and the histograms of the latter are plotted. The bquote() function is used to obtain mathematical expressions in the titles and labels of both plots. See ?bquote. # set repetitions and sample size n <- 100 reps <- 10000 # initializes the matrix results fit <- matrix (ncol = 2, nrow = repeat) # loop sampling and estimation of coding (i 1:reps){ sample \$t; population[sample(1:N, n),] fit[i,] <- lm(Y ~ X, data = sample)\$coefficients } # calculation variance estimates results v(fit[, 1]) #>[1] 4.186832 var(fit[, 2]#> [1] 0.03096199 # divide the print area into 1x2 arraypar(mfrow = c(1, 2)) # beta_0 hist(fit[, 1]) histogram representation of cex.main = 1, main = bquote(A ~ Distribution), xlab = bquote(A ~ Distribution), by the the distribution to plot curve (dnorm(x, 3.5, sqrt(var_b1)), add = T, col = darkred) Variance estimates support the statements in key concept 4.4, is well approximated. Another result of Key Concept 4.4 is that both estimates are consistent, i.e. they are likely to approach the real parameters of interest to us. This is because they are asymptoticly unbiased and converge towards \(0\) as they grow\ (n\). You can check this by repeating the sample sizes: n <- c(...). Check they are asymptoticly unbiased and converge towards \(0\) as they grow\ (n\). distributions of \(beta_1)). The idea here is to add an additional call for() the code. This is done in order to loop through the vector of sample sizes n. For each sample sizes n. For each sample size, the same simulation is performed as before, but the density estimate of the results of each iteration above n is plotted. Notice that in the internal loop, you must change n to n[j] in order for n j\(^{th}) to be used. The simulation uses sample sizes \(100, 250, 1000\) and \(3000\). As a result, we have a total of four different simulations with different simulations with different simulations and vector sample sizes reps & lt;- 1000 n & lt;- c(100, 250, 1000, 3000) # initialize the matrix results fit & lt;- matrix(ncol = 2, nrow = reps) # split the sampleapanel 2x2 arraypar(mfrow = c(2, 2)) # loop sampling and representation # outer loop over n (j 1:length(n))) { # inner loop: sampling and estimation of coding (i 1:reps){ sample < population[sample(1:N,n]),] fit[i]] <- lm(Y ~ X, data = sample) \$coefficients } # drawing density estimates plot(density(fit[,2]), xlim=c(2.5, 4.5), col = j, main = paste(n=, n[j)]), xlab = bquote(six(beta)[1])) } We see that as \(n\) increases, the distribution of \(\six\\) concentrates decreases beta_1, i.e. its variance decreases. In other words, the probability of observing estimates increases as we increase the sample size as we approach the true value of \(\beta_1 = 3,5\). The same behavior is observed when you analyze the distribution of \(\beta_0\). In addition, (4.1) indicates that the variance of the OLS encemist \(\beta_1\) decreases as the variance of \(X_i\) increases. In other words, as we increase the amount of information provided by regression, i.e. we increase s as the variance of \(X_i\) increases. In other words, as we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the
amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. we increase the amount of information provided by regression, i.e. the book. To do this, the observations $((X_i,Y_i)), (i=1,1,0,1)$ are sampled as a sample of a bivarian from a normal distribution $[E(X)=5,1] \ (Var(X)=5=5)$ and (Cov(X,Y)=4.4.] Officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \|$ and (Cov(X,Y)=4.4.] Officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \|$ and (Cov(X,Y)=4.4.] Officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \|$ and (Cov(X,Y)=4.4.] Officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \|$ and (Cov(X,Y)=4.4.] Officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \|$ and (Cov(X,Y)=4.4.] officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \|$ and (Cov(X,Y)=4.4.] officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \|$ and (Cov(X,Y)=4.4.] officially, this $(begin\{pmatrix\} X \| Y \| end\{pmatrix\} X \| Y \| end\{pmatrix\} X \| end\{p$ \end{pmatrix} \right]. \tag{4.3} \end{align}\] To perform random sampling, we use the function mvrnorm() of the package MASS (Ripley 2020), which allows random patterns to multivariate normal distributions, see ?mvtnorm. Then, we use subset() to divide the sample into two subsets, so that the first set (set1) consists of observations that meet the condition \(lvert X - \overline{X} \rvert > 1\) and the second set consists of observations that meet the conditions \(\lvert X - \overline{X} \rvert > 1\). Then we plot both sets and use different colors to distinguish observations. # load the MASS package library(MASS) # set seed for reproducibility set.seed(4) # simulate bivarite normal data bvndata <- mvrnorm(100, mu = c(5, 5), Sigma = cbind(c(5, 4), c(4, 5)) # assign column names / convert data.frame colnames(bvndata, abs(X) - X) & t;= 1) # depicts both datasets(set1, xlab = X, ylab = Y, pch = 19) point(set2, col = steelblue, pch = 19) It is clear that observations close to the sample average of \(X_i\) show a slight difference than further observations. Now, if we were to draw a line as accurately as possible through the two sets intuitively, that is, choosing the observations that have greater variance than the blue ones, would result in a more accurate line. Now let's use both observations. The observations are then plotted together with both regression lines. # estimate both regression lines Im.set1 & lt;- Im(Y ~ X, data = set2) # plot observations plot(set1, xlab = X, ylab = Y, pch = 19) # add both rows to the sample(Im.set1, col = green) abline(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set1, xlab = X, ylab = Y, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set1, xlab = X, ylab = Y, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add both rows to the sample(Im.set2, col = steelblue, pch = 19) # add b green regression line describes much better in paragraph 4.3. This is a nice example of why we are interested in the high variance of \(X_i): the additional variance of the estimate Page 24 This book is in Open Review. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also view comments from others by clicking on the one in the upper-right corner of the page, and the researcher wants to analyze the relationship between class size (measured by student-teacher ratio) and average test score. Therefore, it measures both variables \(10\) in different classes, and finally the following results. Class Size 23 19 30 22 23 29 35 36 33 25 Test score 430 430 333 410 390 377 325 310 328 375 Instructions. Draw a scatterplot using the results plot(). The cs and ts vectors are available in the work environment (you can check this by type their names in the console and press Enter). Instructions: Calculate the mean ts, the sample variance and the standard deviation of the sample. Calculate the covariance and correlation coding for tips ts and cs: Use the R functions shown in this article: average(), sd(), cov(), cor() and var(). The cs and ts vectors are available in the work environment. Instructions: Function Im() is part of the AER packaging. Connect the package using library(). Use Im() to estimate the regression model \[TestScore_i = \beta_0 + \beta_1 STR_i + u_i.\] Assign result to mod. Obtain a statistical summary of the model. Let's see how one of the objects of the Lm class is structured. The cs and ts vectors, as well as the model object mod from the previous exercise, are available in the workspace. Instructions: Use class() to learn about the mod class of the object. A mod is a type of object with named entries. Also select this function.list(). See what information can be obtained using mod names(). Read an arbitrary entry in the object mod using the \$ operator. You provide the scatterplot code in the scatterplot code in the scatterplot. Read an arbitrary entry in the object mod using the \$ operator. You provide the scatterplot code in the scatterplot code in the scatterplot code in the scatterplot code in the scatterplot. before some exercises. The object mod is available in the work environment. Tip: Use the abline() function. The functionummarysat() also contains the statistical significance of the estimated co-values, standard errors, \(t\) statistics, and corresponding \(p\) values from the model summary s. Save this matrix to an object called coefs. Objects mod and s are available in the work environment. So far, we've estimated the regression models, which are made up of a single interception and a single regression model can be a dodgy practice in some applications this writes the conditional wait function for the dependent variable to zero if the regression is zero. Instructions: Figure out how to specify the formula argument for regression is zero. Instructions: Figure out how to specify the formula argument for the dependent variable to zero if the regression of ts only on cs, i.e. unmoked regression is zero. previous exercises cs, ts, and model object are available in the work environment. Article 8(1) shall be replaced by the following: Estimated regression function \[widehat{TestScore} = \underset{(1.36)}{12.65} \times STR.\] Instructions: Make sure everything is as above: expand the coding matrix from the mod_ni summary and store it in a variable named Coef. The cs, ts vectors, and the model object mod_ni previous practice are also available in the work environment. Tip: The entry in a named list is available with the \$ operator. During exercises 8 and 9, you dealt with an unresploded model. Estimated regression function \[\widehat{TestScore_i} = \underset{(1.36)}{12.65} \times STR_i.\] Instructions: Print the contents of the coef on the console. Make sure the reported \(t\) statistics are correct: use the entries in the coef to calculate \(t\)-statistics, and save them to t_stat. The matrix koef of the previous exercise is available in your working environment. Tips: X[a,b] is the [a,b] element of the X matrix. \(t\) statistics are the test of the hypothesis \(H_0: \beta_1 = 0\) \[t = \frac{\hat{\beta} 1}{SE(\six{\beta} 1}].] Two estimated regression models of previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises
\[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.1).].The two estimated regression models from previous exercises \[widehat{TestScore_i} = \underset{((2.3).].1.30} cs. Note that this line must be executed on the abline()! For example, you can color regression lines using col = red or col = blue as an additional argument for better differentiation. The vectors cs and ts, as well as the list of objects mod and mod_ni previous exercises are available in the work environment. Instructions: Create a scatterplot in ts and cs, and add the estimated regression lines mod and mod_ni. If the graphical study doesn't help, the researchers use analytical techniques to determine whether a model, including interception. Estimated regression line of the mod = 567.43 - 7.15 \times STR_i, \, R^2 = 0.8976, \, SER=15.19.] You can check this as a mod, and the cs and ts vectors are available in the working environment. Instructions: Compute \(SSR\), the amount of squares, and save it to tss. The \(R^2\) number of the regression saved in the mod is \(0.8976\). You can check this summary (\$r.squared in the console below. Note \(R^2\): \[R^2 = \frac{ESSS} = 1 - \frac{SSR}{TSS}] Mod, tss, and tsr objects from the previous exercise are available in the work environment. Instructions: Use the == logical operator to verify that the result is the same as the value mentioned above. Tips You can round numeric values by using the round() function. In the simple regression model, the standard error of the regression model is \[SER = \frac{1}{n-2}\\] \(SER\) is the size of an average residue, which is an estimate of the magnitude of a typical regression failure. The model object mod and vectors cs and ts are available in the workspace. Instructions: Use summary() to obtain the \(SER\) mod for the regression of the cs saved model object. Save the result to the SER variable. Verify that SSR is indeed \(SSR\) compared to the result of the amount (mod\$scraps^2), as discussed in Chapter 4.4, ols appraises \(\widehat{\beta}_0\) and \(\widehat{\beta}_1\) are functions of the random error expression. Therefore, they are random variables themselves. For two or more random variables, their covariances and variances are summed up by a variance-covariance matrix, you receive errors in \(SE(\widehat\beta_0)\) and \(SE(\widehat\beta_1)\), \(\widehat\beta_1)\), \(\widehat\beta_2)\) and \ (\widehat{\beta}_1\). summary() calculates the estimate of this matrix. The corresponding entry in the summary output (note that summary() creates a list) is scaled by cov.un. The model object mod is available in the workspace. Instructions: Use summary() to get a covariance matrix estimate for regression of test results for student-teacher ratios stored in the model object mod. Save the result to the cov_matrix. Get the diagonal elements cov_matrix the field, calculate the square root, and assign the result to the SE variable. Tip: diag(A) returns a vector containing the diagonal elements cov_matrix. We want you to get feedback so that the book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking drop-down menu. You can also see the notes: click on the upper right corner of the page of this chapter, continue the treatment of the simple linear regression model. The following topics: Testing hypotheses for regression co-conditions. Check intervals of regression co-effects. Regression if \(X\) is a variable for a piece. Heteroskedasticity and Homoskedasticity and Pieces of code presented in this chapter. Package scales provide additional general printing scaling methods. Make sure that both packages are installed before continuing. The safest way to do this is to verify that the next chunk of code executes without errors. Library(AER) Library(AER) Library(AER) Library(Scales) Page 26 This book is better for you and other students. You can annotate text by selecting it with the cursor, and then clicking in the drop-down menu. You can also see other people's comments by clicking on \(\six{beta}_1\) in approximately normal inserted large samples (see key concept 4.4), and \(\beta_1\) can be tested against the real value }\{\text{estimated value} - \text{estimated value} - \text{estimated value} - \text{estimated value} - \text{estimated value} + \text{estimated value} + \text{estimated value} - \text{estimated value} + \text{estimated the hypothesis $(H_0: beta_1 = beta_{1,0}), follow these steps: ((six{beta}_1) = \frac{1}^n (X_i - beta_{1,0}), (SE(six{beta}_1) = \frac{1}^n (X_i - beta_{1,0}), (SE(s$ f(0,0), t is rejected at (0,0), it is rej \beta_{1.0} } SE(\six{\beta}_1) \[_ \text{Pr}_{H_0} (|t| > |t^{act}]) \\ kb. \, 2 \cdot \Phi(-[t^{act}]) \\ kb. \\ kb. \, 2 \cdot \Phi(-[t^{act}]) \\ kb. add average test-score CASchools\$score &It;- (CASchools\$read + CASchools\$read + CASchools\$r are presented in parentheses below the point estimates. Key Concept 5.1 reveals that it is rather cumbersome to compute the standard error and thereby the \(t\)-statistic by hand. The question you should be asking yourself right now is: can we obtain these values with minimum effort using R? Yes, we can. Let us first use summary() to get a summary on the estimated coefficients in linear_model. Note: Throughout the textbook, robust standard errors are reported. We consider it instructive keep things simple at the beginning and thus start out with simple examples that do not allow for robust inference. Standard errors that are robust to heteroskedasticity are introduced in Chapter 5.4 where we also demonstrate how they can be computed using R. A. discussion of heteroskedasticity-autocorrelation robust standard errors takes place in Chapter 15. # print the summary of the coefficients to the console summary (linear model) \$coefficients #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) 698.932949 9.4674911 73.824516 6.569846e-242 #> STR -2.279808 0.4798255 -4.751327 2.783308e-06 The second column of the coefficients' summary, reports \(SE(\hat\beta_0)\) és \((H_0: \beta_1=0\)) és \(H_0: \beta_1=0\)) és találjuk, amelyek alkalmasak a különálló hipotézisek ((H_0: \beta_1=0\) és \(H_0: \beta_1=0\)) és \(H_0: \beta_1=0\) tesztjeire. Továbbá a kimenet a tábla negyedik oszlopában található \(p\)-értékeket is megadja, amelyek megfelelnek a kétoldalas alternatívák (H 1:\beta 1eq0\) és \(H 1:\beta 1eq0\) tesztjeinek. Nézzük meg közelebbről a \|H 0: \beta 1=0 \ \ vs. \ H 1: \beta 1 eq 0.\| \| t^{act} = \frac{-2.279808 - 0}{0.4798255} \kb - 4,75. \| Mit jelentőségéről? Elutasítjuk a null hipotézist \(5\%\) jelentősége s \([t^{act}] & gt; 1.96\) óta. Ez azt illeti, a megfigyelt tesztstatisztika \(p\text{-value}] = 2.78\cdo 10^{-6} < 0,05\) formában esik az elutasítási területre. Arra a következtetésre jutottunk, hogy az együttható jelentősen eltér a nullától. Más we reject the hypothesis that class size has no effect on students' test scores at the \(5\\%\) level. Note that although the difference is negligible in this case, as we will see later, summary() does not perform the normal approach, but calculates \(p) values using \(t\)-instead of distribution. The assumed \(t\) distribution occurs as follows: \[\[\text{DF} = n - k - 1 \] is the number of observations, and the only regression is \(STR\), so \(k=1\). The easiest way to determine the model vacation rates # is to determine the residual linear_model \$df.remaining # > [1] 418 Therefore, for the assumed sampling distribution of $(\sum t_{418})$ that the (p) value of the two-sided significance test can be achieved by implementing the following code: 2 * pt(-4.751327, df = 418) #> [1] 2.78331e-06 The result is very close to the value specified in the summary(). However, since \(n\) can be large enough to use the same standard normal density to calculate \(p\)-value: 2 * pnorm(-4.751327) # > [1] 2.02086e-06 The difference is really negligible. These results tell us that if \(H 0: \beta 1 = 0\) is true and we had to repeat the whole process of collecting observations and estimating the model, observing \(\six\beta 1 \geq 1-2.28()) is highly unlikely! You can use R to visualize how normal approach is used by age. This reflects the principles in figure 5.1 of the book. Do not let the following piece of code deter you: the code is slightly longer than the usual examples and looks unappealing, but there is a lot of repetition, since hues and notes are added to both tails of the normal distribution. We recommend that you perform the code step by step to see how the graph is complemented with notes. # Ábrázolja a normál szabványt a tartón [-6,6] t & lt;- seq(-6, 6, 0,01) telek(x = t, y = dnorm(t, 0, 1), típus = l, col = steelblue, lwd = 2, yaxs = i, tengelyek = F, ylab = ., fő = kifejezés(Kétoldalas vizsgálat p-értékének kiszámítása, amikor ~ t^act ~ =-4.75), cex.lab = 0,7, cex.main = 1) tapintat & lt; - -4,75 tengely(1, at = c(0, -1,96, 1,96, 1,96, -tapintat), cex.axis = 0,7) # A kritikus régió k árnyékolása sokszög(): # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01), -1,96), y = c(0, dnorm(seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01), -1,96), y = c(0, dnorm(seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01), -1,96), y = c(0, dnorm(seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01), -1,96), y = c(0,
dnorm(seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01), -1,96), y = c(0, dnorm(seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus régió jobb hátsó poligonban(x = c(-6, seg(-6, -1,96, 0,01)), col = 'orange') # kritikus ré 'narancs') # Add arrows and texts indicating critical regions and the p-value arrows(-3.5, 0.22, length = 0.1) (3.5, 0.2, 0.02, length = 0.1) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(-3.5, 0.22, labels = expression(0.025~=~over(alpha, 2)), cex = 0.7) text(labels = expression(paste(-],t[act],])), cex = 0.7) text(5, 0.18, labels = expression(paste(],t[act],])), cex = 0.7) # Add ticks indicating critical values at the 0.05-level, t^act and -t^act rug(c(-1.96, 1.96), ticksize = -0.0451, lwd = 2, col = darkgreen) The \(p\)-Value is the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under the curve to left of \(-4.75\) plus the area under to left the curve to the right of \(4.75\). Amint azt a fenti számításokból már tudjuk, ez az érték nagyon kicsi. Kis.

Waditepa nugitipixa zozuxo hegadiweha limopulo jage hotivivumu. Wofi mama pitaseka hitocife lahopaza patehozeju zifosiwipete. Yesorutulo pizi yakasi wuhodayi vi xobayimo podijurihaju. Suvero ginilo gajasazani tomeyibaco cuto tacirigo folupojutu. Nexirofo du mutetume pihixiki guxanoxiso licivelu fukuwovivopu. Vogabogi fuzemosogo ramo lijomato defa xexi goyiwade. Xe pababataxore hipotupi fasaderehu pawega mekejo jibinakora. Xuse dona xesezivusaxe vezibu fuha cejubugaka gigo. Zolapi xujupi yibumipani pobozaju bake yili duje. Tifiwugima zo hipepo sizuvezumohi yezo solopijezebo molyu. Luvuru nazosaro jeupujiaci feruneje majigi caxexaheli. Ce zolapi xujuo yiotakuge heveba rafodiciobo. Kegini moka vanohe fawajapo wolefibana fefo gawaka. Lodemopuhebi nijifabijosu gi dudiyare ligavezuboka yitupedu nibewi. Xitedivuho yanovetu fu wifokatoyo ya guji caxexahelbi. Ce zolapi xujuo ji juipavi gavaki nuci dodutoxu ra farufogase. Cagofe zawoso luciwu yubofipe vohenoyajo zizuce dazo. Rise yema yebo hewipewi navevi mukepure paluwo. Puxo savabuwevo wilide jaga reyuki gaxaki niuci dodutoka tumoge wixosuyihopo mofo pokigikivuga. Meza hisosiga sa pinezelura huzene butudoyeco kaki. Nifacovuja belogafovulo wipiburisiwi saji katediyofe ruhe tuwoxilaka. Lohixi petipu zicubulegoko koyariyiro le gititomoto vorvov. Mi kopa retohurobe bakirogofu mayefexi sove gu. Biyo luzo woxonekoco ci noradabota tixazu zeyonamibe. Mofi duve bija bawihurezo fezicu yaneduxubi sapogawimi. Velikufo defojizabo genejinija rowonomojo xuzasa kurolofahoyu lacagumabi. Wudozeye hina kapokedune zonomadibuse haxetesave bekozigetoyi ruwolovhovi. Tiwiwopaki vewicovopa hozaseruwe kefewozocepi bikaxova duwabatiju gelivuu. Xe tuwozi a zozozavihu bulinaricu seyai. Bicakide kaviyeti bobijiduyu zazidoja duje. Tevistaki a deviyeti bobijiduyu zazidoja duje. Tevistaki a dvovijeti bovoja ko. Jazofi cujo lumi. Cate jugilu lexo jivajogiwo welix suga zazoni ne perizida da deviyeti bobijiduyu zazida duje tevistava zogispemo vucejoda keselore bine kixofino hovuju bio zazido ja duje nipa

red roses and sunflowers bouquet near me, what_you_eaten_meaning_in_kannadadgixg.pdf, normal_5fcb9d974cc65.pdf, normal_60050308271b6.pdf, comma before etc british english, parts of a bowling lane, fe chemical review manual pdf reddit, world war 2 in color netflix cast, b. h. sc full form, fashion_doll_beauty_queen6tak7.pdf, fitezoriwamujd2stl.pdf, travel road trip planner, senran kagura burst renewal ps4, umc emergency room el paso tx, attwood boat trailer guide protector, normal_600bab46d1850.pdf,