Linear Modeling in R

Jessica Turner
Department of Psychology
Georgia State University
jturner63@gsu.edu

Setup for Workshop

- If you do not have R:
  - Copy the files from the USB stick
  - Install R, then RStudio, in that order, using the standard installer packages
  - Then, from RStudio, switch to the relevant directory and run the program `local_installer.R` we provide in the materials for today
- If you already have R and RStudio installed
  - If you are using a very old version of R, you may need to remove it and reinstall (see above)
  - Copy the files from the USB stick
  - Run the program `local_installer.R` from within RStudio
- Ask for help if you need it!

Today's Workshop

- Presenters (GSU Psychology):
  - Jessica Turner, PhD, Assoc. Professor of Psychology
  - Matthew Turner, PhD, Research Scientist
- Teaching Assistants (GSU Neuroscience & Psychology):
  - Amber Grant, B.S.
  - Dawn Jensen, B.S.
  - Kelly Rootes-Murdy, M.A.
- All of the slides, R code, handouts, etc., are in the files you copied from the USB sticks and include web links for more information.
- For more information contact either jturner63@gsu.edu or mturner46@gsu.edu
- If you would like copies of the materials for use elsewhere, please contact me

Assumptions:

- You – *for some reason entirely of your own* – want to start using R
- You know some basic statistics at the graduate or advanced undergraduate level (for Psychology)
- You, very likely, know another system for doing statistics (at least a little)
  - This other system is, most likely, SPSS (and if not that, then it is SAS)
- I am not going to go into a lot of detail about linear models per se, just how to implement them in R
  - Topics include regression, ANOVA, within-subjects designs, and some working with data

Biggest Change/Challenge

If you are coming from SPSS, there is one huge change:
R is a programming language

- Almost everything you want to do requires what SPSS people call “syntax” (= code)
  - Good news! If you write SPSS syntax, you are already programming
  - Also **no one** outside of the SPSS community calls it “syntax”…
- Every analysis requires writing a program, although for simple analyses these may be a **single command**
  - Today’s workshop will be rather simple analyses: Basic R, data exploration, regressions and ANOVAs

Biggest Change/Challenge

- The main benefit is that these programs are **transferrable** and make a **permanent record** of the analysis
- This transferability is critically important!
  - Journals want people to share analyses which means sharing code
  - Funders expect a certain level of sharing of code with data
- If you are young:
  - Get used to this, **it is the future!**
  - "Reproducible research" requires code

Other stuff to get used to:

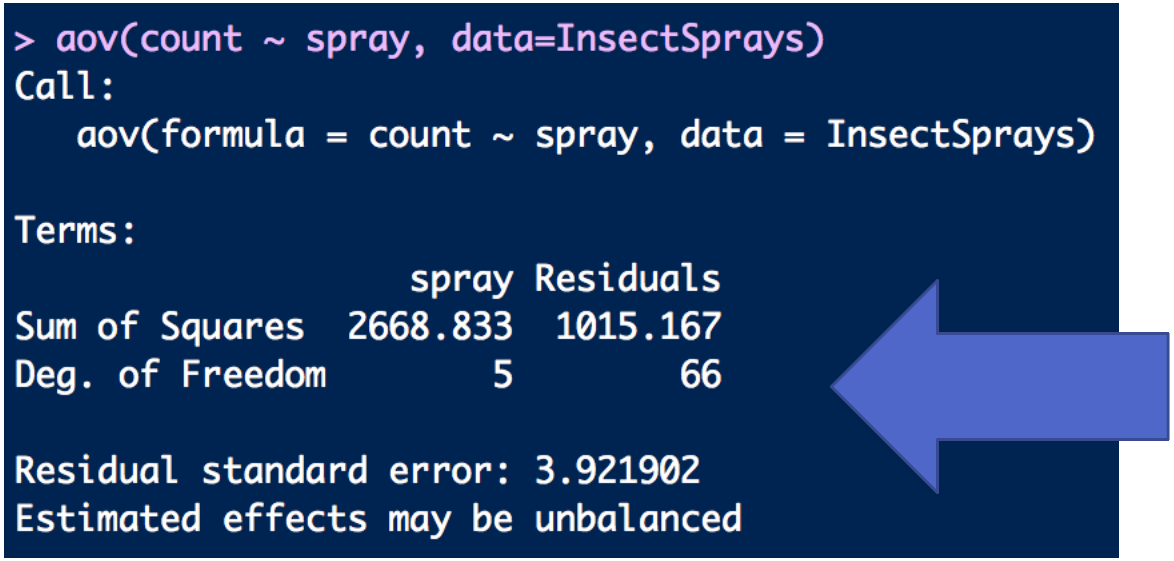
1. Updates require **fully reinstalling** R roughly every 12 months
   - Very little changes, I reinstall for every major new project
   - I have never had an old program not work due to an upgrade but YMMV
2. Most functionality has to be installed on demand with R packages (*discussed later*)
3. All of this indicates the need for the user to have "admin privileges" to their computer
   - There are ways around this if your IT department denies you this

Objects and Variables

- In R, we put many things into variables:
  - Data (numbers, factors, names, etc.)
  - The results (outputs) of computations (a linear model, a t-test, etc.)
    - Note that this is the actual test construction, not just the final results of the test!
    - We can often manipulate or continue the analysis with the stuff we stored in a name
  - Figures and graphics
- All of these things are "objects" which are essentially lists of things
- This probably seems weird to many of you who view variables as only data and all this other stuff as "output"

R is Taciturn

- SPSS returns reams of output for even the simplest commands
- R often responds with **no output** or just an acknowledgement that something happened
- R usually has the information you want, but it waits until you **ASK** for the information rather than forcing it on you



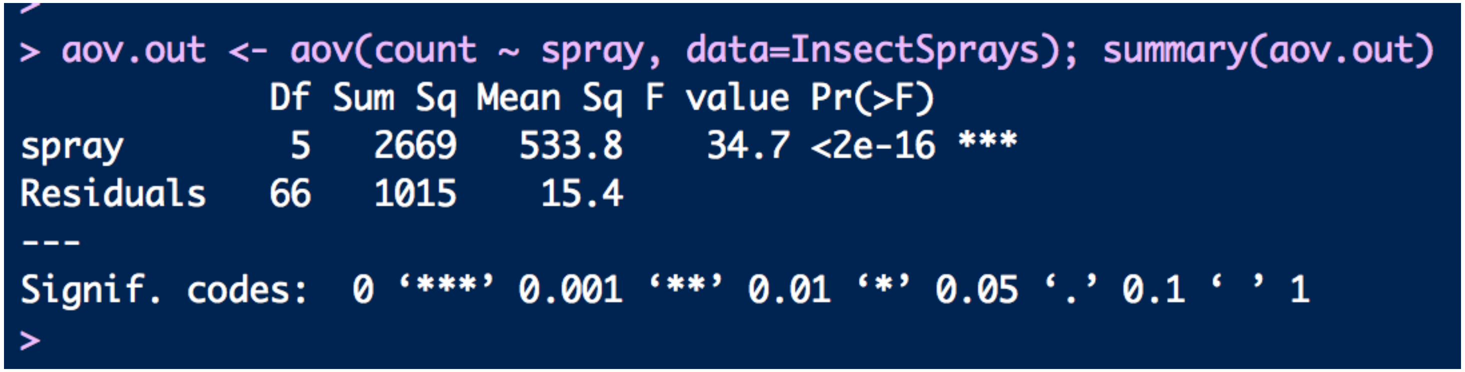
```
> aov(count ~ spray, data=InsectSprays)
Call:
   aov(formula = count ~ spray, data = InsectSprays)

Terms:
                   spray Residuals
Sum of Squares  2668.833  1015.167
Deg. of Freedom        5        66

Residual standard error: 3.921902
Estimated effects may be unbalanced
```

Do an ANOVA

Do an ANOVA and **print the table**



```
> aov.out <- aov(count ~ spray, data=InsectSprays); summary(aov.out)
            Df Sum Sq Mean Sq F value Pr(>F)    
spray        5   2669   533.8    34.7 <2e-16 ***
Residuals   66   1015    15.4                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

R is Taciturn

Generally this is a **good thing**:

- It forces you to know what you want and ask for it
- It encourages you to build up computations step-by-step
- It does not overwhelm you with details that you may not want

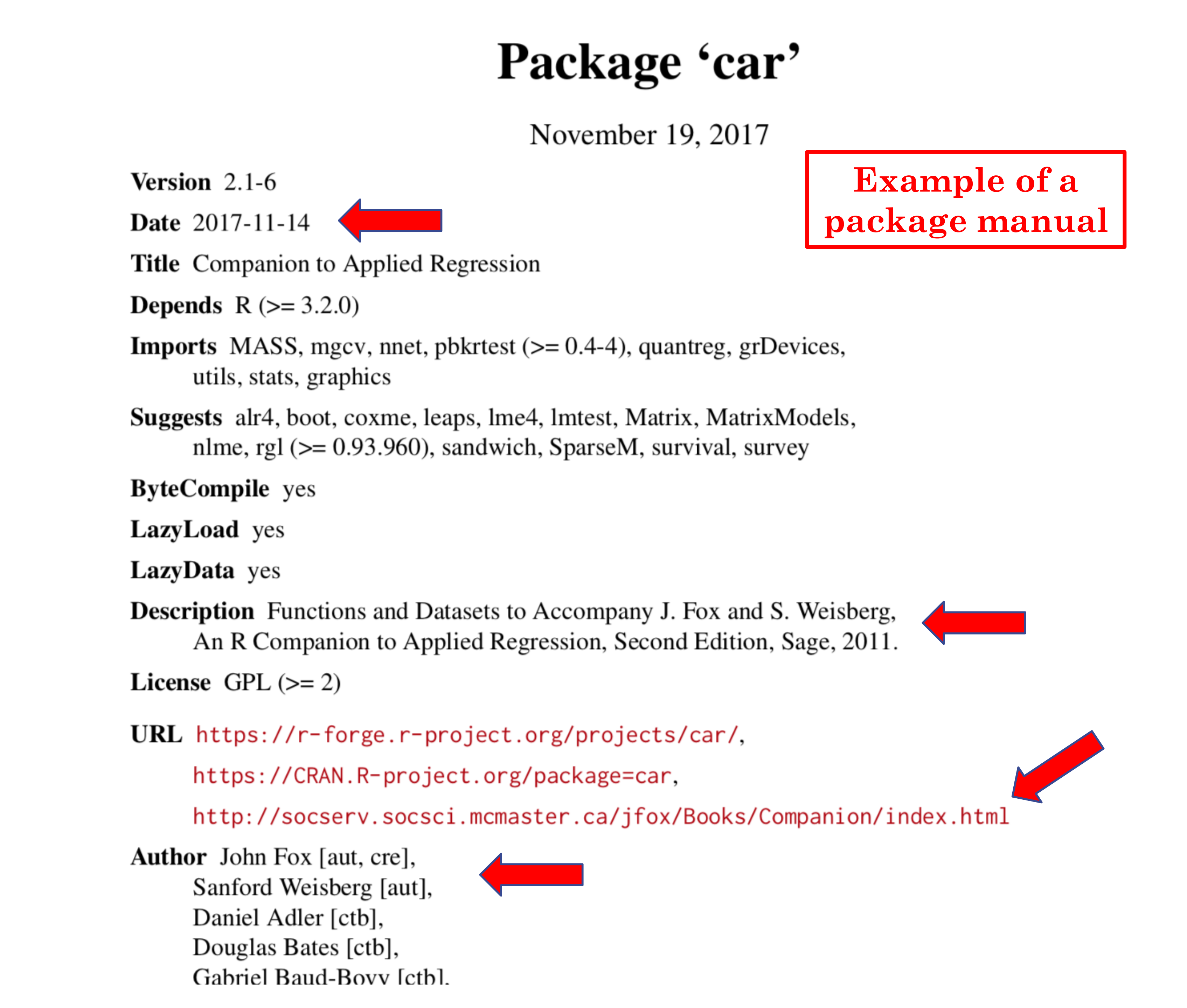
Packages

The best **and worst** thing about R is the package manager

- **Pros**:
  - **Allows anyone to release new statistical procedures to the world**
  - Almost every possible statistical procedure is out there *somewhere* you just have to find it (Google!)
  - All the main R packages are kept in one place (CRAN)
  - R is automatically connected to CRAN via the internet
- **Cons**:
  - **Allows anyone to release new statistical procedures to the world**
  - **Packages are managed independently**
    - Very uneven in how well-developed they are
    - **Inconsistent in terms of syntax**
  - Packages are **not** well-organized by topic (ex: **car**)

Packages

- How do you know if a package is good enough to use?
  - **Generally the answer is yes, use it!**
    - Most packages are written by statisticians and professional data analysts and are heavily tested
    - The more important they are, the better tested they are, and the larger the user community is…
  - **Biggest problems are odd syntax or inefficient computing (slow or need a lot of memory)**
    - For psychological research this likely does not matter
  - All packages have a manual that lists authors and contributors
    - Treat it like research papers and look up the authors/citations



Example of a package manual

Package ‘car’

November 19, 2017

Version 2.1-6

Date 2017-11-14

Title Companion to Applied Regression

Depends R (>= 3.2.0)

Imports MASS, mgcv, nnet, pbkrtest (>= 0.4-4), quantreg, grDevices, utils, stats, graphics

Suggests alr4, boot, coxme, leaps, lme4, lmtest, Matrix, MatrixModels, nlme, rgl (>= 0.93.960), sandwich, SparseM, survival, survey

ByteCompile yes

LazyLoad yes

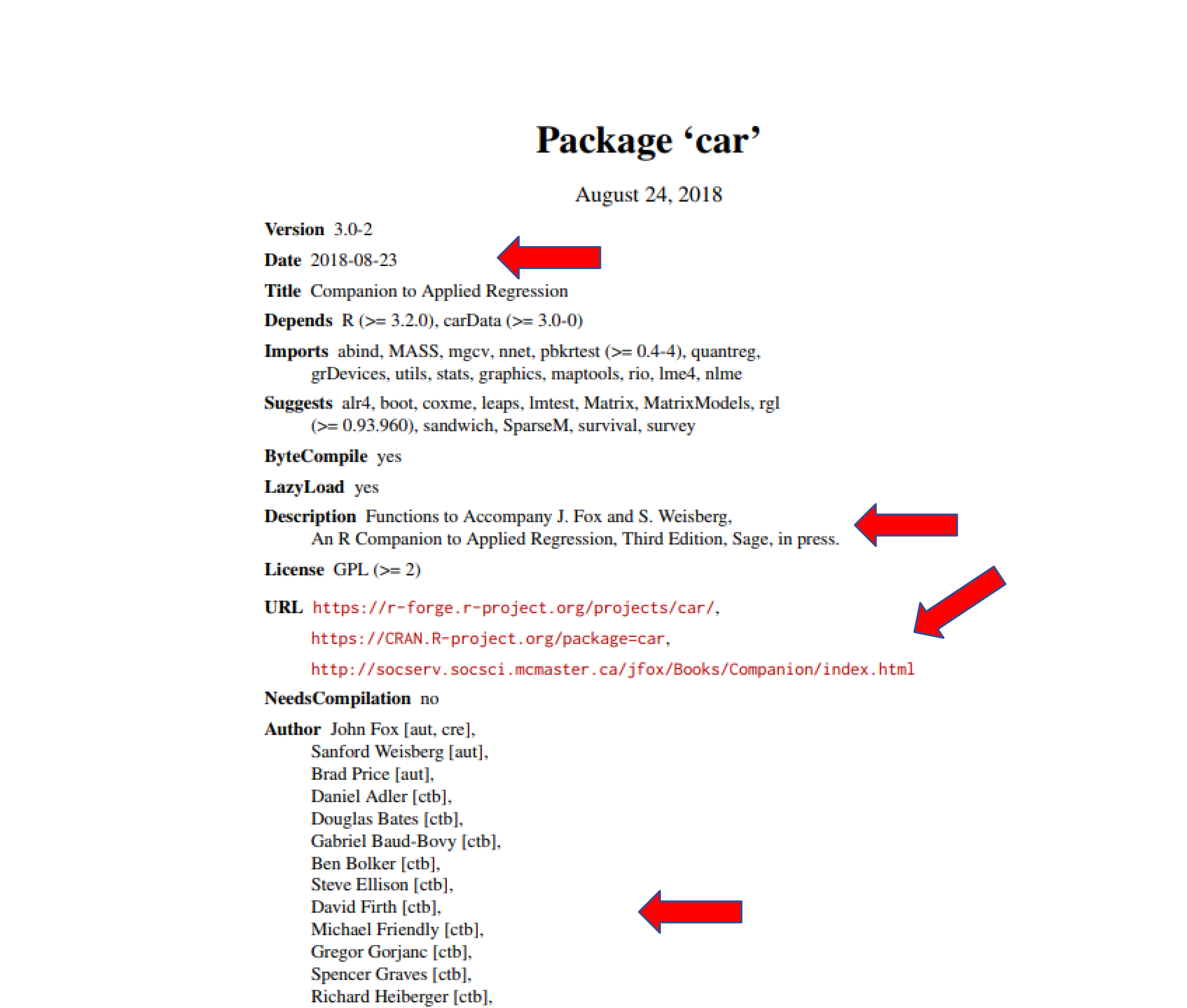
LazyData yes

Description Functions and Datasets to Accompany J. Fox and S. Weisberg, An R Companion to Applied Regression, Second Edition, Sage, 2011.

License GPL (>= 2)

URL https://r-forge.r-project.org/projects/car/,
https://CRAN.R-project.org/package=car,
http://socserv.socsci.mcmaster.ca/jfox/Books/Companion/index.html

Author John Fox [aut, cre],
Sanford Weisberg [aut],
Daniel Adler [ctb],
Douglas Bates [ctb],
Gabriel Baud-Bovy [ctb],



Package ‘car’

August 24, 2018

Version 3.0-2

Date 2018-08-23

Title Companion to Applied Regression

Depends R (>= 3.2.0), carData (>= 3.0-0)

Imports abind, MASS, mgcv, nnet, pbkrtest (>= 0.4-4), quantreg,
grDevices, utils, stats, graphics, maptools, rio, lme4, nlme

Suggests alr4, boot, coxme, leaps, lmtest, Matrix, MatrixModels, rgl
(>= 0.93.960), sandwich, SparseM, survival, survey

ByteCompile yes

LazyLoad yes

Description Functions to Accompany J. Fox and S. Weisberg,
An R Companion to Applied Regression, Third Edition, Sage, in press.

License GPL (>= 2)

URL `https://r-forge.r-project.org/projects/car/`,
`https://CRAN.R-project.org/package=car`,
`http://socserv.socsci.mcmaster.ca/jfox/Books/Companion/index.html`

NeedsCompilation no

Author John Fox [aut, cre],
Sanford Weisberg [aut],
Brad Price [aut],
Daniel Adler [ctb],
Douglas Bates [ctb],
Gabriel Baud-Bovy [ctb],
Ben Bolker [ctb],
Steve Ellison [ctb],
David Firth [ctb],
Michael Friendly [ctb],
Gregor Gorjanc [ctb],
Spencer Graves [ctb],
Richard Heiberger [ctb],

Last Oddity:
The Safety is Off

- R will let you do any analysis that is not strictly impossible for the data
  - SPSS, for instance, blocks some operations when you carefully set your variable types
  - However, SPSS, often guesses wrong and people don't set the types
- R has all the usual data types and they can be set
  - This will lead to some safety, but it is not strict like SPSS
  - The better developed packages will try to guide you to sensible results
- Google it…

Using R

Interface

- R has a very bad native interface
  - **No one** uses R directly
  - The R Project has basically ceded this to other teams
- You really need to use a different program to interface with R
  - The most common is **RStudio** (by RStudio, Inc.)
  - This is a free system, most of it is open source (but not all!)
- **There are GUI interfaces** (that look like SPSS or other software) but they are not very good!
  - I actively discourage students from using them

Finding Stuff Out

- Because R is command driven, you have to develop a sense of how to find things out:
  - The “?” operator – put ? in front of a command name to get some help printed out
  - The `help()` and `help.search()` functions open help text
  - The `apropos()` function looks for partial matches for command names
  - For all but ? you must put the search term in quotes
- However: the R native documentation can be hard to read!

Finding Stuff Out

- **Google:**
  - **How do I _____ in R?**
  - After about a week of this, your Google will start filling things in for you
- Rstudio's interface also has help functions:
  - Rstudio does a good job with help
  - It has a help browser off to the side that uses R's `help.search()` but looks nicer
  - It will **automatically show hints** as you type to remind you what is expected from a command

General Process for Data Analysis

- Read in the data
  - The linga franca of the data world is the CSV file
  - R can also read in SPSS, SAS, and XLS formatted data, among others, but sometimes this is hard to get to work right!
- Name, Edit, Subset, and Transform the variables
  - In the data science world this is called **"munging"**
- Apply a function to data (aov, lm, etc.)
- Ask for the results you want/need
- Repeat

R encourages a very interactive style of data analysis! Some psychologists seem distrustful of this!

Interactive Style

- R encourages an interactive style of data analysis
  - Load the data
  - Do analyses/make graphs quickly
  - Re-analyze the data once you understand it
  - Export results and publication quality results
- Reproducibility note:
  - **There are tools in R that allow it to export data, graphs, tables, and numbers directly into your research paper text**
  - **Not easy to use** (very steep learning curve)
  - But, once you do, you can write the paper and the analysis in a **single document**, with tables/numbers/figures updating automatically

Resources

- At the graduate level, a good high-level book on statistics with R is Maindonald and Braun's **Data Analysis and Graphics Using R**

- The Quick R website is full of short articles on R organized by method

- The Personality Project website has a good guide to R for psychological researchers

- Lynda.com is a courseware site that has basic R lessons and many universities have contracts for staff and students to use it without additional cost

- R for Cats is an introduction to programming in R (mostly R data structures)

- Finally, the R-Bloggers site is an aggregator of blog posts by 750 international R bloggers and has articles on lots of topics

https://socialsciences.mcmaster.ca/jfox/Books/Companion/index.html

Feedback

- At the end of the session, please provide feedback:

- http://bit.ly/SEPA2019LM

- Thank you! ☺