Getting Started with the New Statistics in R

David Erceg-Hurn, Geoff Cumming, Robert Calin-Jageman

2016-08-01

Overview

R is a popular and powerful free program that can be used to conduct most of the statistical analyses outlined in *Introduction to the New Statistics*. Unlike programs such as SPSS where analyses are usually conducted by clicking on menus, in R analyses are typically performed by typing *commands*.

This document is a brief guide that will help you to get started using the 'new statistics' in R. The guide is split into three sections. The first section provides some tips about installing and learning the basics of R. If you've never used R before you should read this section - if you already know how to use R you can skip it. The second section provides a brief overview of a new R package, itns, that contains the datasets used in *Introduction to the New Statistics*. You can use the datasets in the *itns* package to work through the examples covered in the book and the end-of-chapter exercises. The final section provides a short overview of R packages and functions that can be used to conduct the analyses covered in *Introdudction to the New Statistics*.

You will notice that some words in this document are a blue colour. These are hyperlinks. If you click on the blue text, you will be redirected to websites that contain information about using R.

Part One - Installing R and Learning the Basics

To install R, visit the RStudio website and following the installation instructions. That webpage also contains links to interactive tutorials for R beginners. The tutorials will help you learn how to perform basic tasks like importing and manipulating datasets. Other useful resources for learning R include:

- R for Data Science - An online book by Garrett Grolemund and Hadley Wickham that will teach you how to import, tidy, and explore data.
- Kelly Black's R Tutorial - An introductory tutorial focusing on the basics of R.
- How to Learn R Blog - A collection of resources that will help you learn R.
- Quick-R - A website that contains example code for running basic analyses.
- R Quick Reference Card - A list of key commands built into R.

Also remember that Google is your friend. If you have a question about how to do something in R, it is likely that someone else has already asked the same question and that there is an answer on the Internet. For example if you type 'R how to create a histogram' into Google, you will find many links to webpages showing you the R code that you need to plot a histogram.

In the remainder of the document, we assume that you have a basic understanding of how to use R.

Part Two - The *itns* Dataset Package

'itns' is an R package which contains most of the datasets used in *Introduction to the New Statistics*. The datasets were converted from Microsoft Excel files (found on the book's website) into R data frames. The table on the next page lists the names of the data frames in the package, and the sections of the book where they are mentioned:

`itns` Package Data Frames

Name	Section	Topic
college_survey1	Ch 3 End of Chapter Exercises 2 & 3	Descriptive Statistics & Plots
religious_belief	Ch 3 End of Chapter Exercise 4	Descripitive Statistics & Plots
college_survey1	Ch 5 End of Chapter Exercises 2 & 3	Single Sample Confidence Interval
college_survey2	Ch 5 End of Chapter Excercise 4	Single Sample Confidence Interval
stickgold	Ch 6 End of Chapter Exercise 5	Single Sample Confidence Interval
pen_laptop1	Ch 7.6-7.12	Two Independent Groups
pen_laptop2	Ch 7.36-7.38	Two Independent Groups
anchor_estimate	Ch 7 End of Chapter Exercise 3	Two Independent Groups
clean_moral1	Ch 7 End of Chapter Exercise 4	Two Independent Groups
clean_moral2	Ch 7 End of Chapter Exercise 4	Two Independent Groups
math_gender_iat	Ch 7 End of Chapter Exercise 5	Two Independent Groups
thomason1	Ch 8, 11, 12	Two Dependent Groups, Scatterplots, Regression
thomason2	Ch 8	Two Dependent Groups
thomason3	Ch 8, 12.18	Two Dependent Groups, Regression
emotion_heartrate	Ch 8 End of Chapter Exercise 3	Two Dependent Groups
labels_flavor	Ch 8 End of Chapter Exercise 4	Two Dependent Groups
ma_anchor_adjust	Ch9 End of Chapter Exercise 1	Meta-Analysis
ma_flag_priming	Ch9 End of Chapter Exercise 2	Meta-Analysis
ma_math_gender_iat	Ch9 End of Chapter Exercise 3	Meta-Analysis
ma_power_performance	Ch9 End of Chapter Exercise 4	Meta-Analysis
body_well	Ch 11, 12	Correlation, Regression
exam_scores	Ch 11 End of Chapter Exericse 2	Correlation
sleep_beauty	Ch 11 End of Chapter Exericse 6	Correlation
campus_involvement	Ch 11 End of Chapter Exericse 7	Correlation
home_prices	Ch 12 End of Chapter Exercise 2	Regression
home_prices_holdout	Ch 12 End of Chapter Excerise 2h	Regression
altruism_happiness	Ch 12 End of Chapter Exercise 3	Regression
rattan	Ch 14.10-14.12	One-Way Independent Group Contrasts and Comparisons
organic_moral	Ch 14 End of Chapter Exercise 5	One-Way Independent Group Contrasts and Comparisons
videogame_aggression	Ch 15 End of Chapter Exercise 3	Analysing factorial designs
self_explain_time	Ch 15 End of Chapter Exercise 4	Analysing factorial designs
natsal	Ch 16.11	Robust Methods - Two Independent Groups
dana	Ch 16 End of Chapter Exercise 3	Robust Methods - Two Independent Groups

The `itns` package is not yet on CRAN, but can be downloaded from github using the `devtools` package:

```
# install.packages("devtools")
library(devtools)
install_github("gitrman/itns")
```

Once you have installed the package, you can use the `library()` function to load it, `str()` to examine metadata for each data frame, and functions such as `head()` and `tail()` to print the first or last few rows to your screen.

```
library(itns)      # loads the package
str(pen_laptop1)   # displays metadata
```

```
## 'data.frame':    65 obs. of  2 variables:
##  $ group        : Factor w/ 2 levels "Laptop","Pen": 2 2 2 2 2 2 2 2 2 2 ...
##  $ transcription: num  12.1 6.5 8.1 7.6 12.2 10.8 1 2.9 14.4 8.4 ...
```

```
head(pen_laptop1) # prints the first few rows
```

```
##   group transcription
## 1   Pen          12.1
## 2   Pen           6.5
## 3   Pen           8.1
## 4   Pen           7.6
## 5   Pen          12.2
## 6   Pen          10.8
```

```
tail(pen_laptop1) # prints the last few rows
```

```
##     group transcription
## 60 Laptop          10.3
## 61 Laptop           9.0
## 62 Laptop          12.8
## 63 Laptop          12.0
## 64 Laptop          34.7
## 65 Laptop           4.1
```

To access further details about each dataset, type a question mark and the name of the dataset, for example:

```
  ?pen_laptop1
```

or access the PDF help file **LINK TO GO HERE** on the itns github site.

The datasets in the `itns` package can be used to replicate analyses that appear in *Introduction to the New Statistics*, and to work through the book's end-of-chapter exercises using the packages and functions outlined in the next section of this guide.

Part 3 - Helpful Packages and Functions

Most of the analyses described in *Introduction to the New Statistics* can be conducted using inbuilt R functions, or functions in packages that can be downloaded from CRAN or github. In this section we mention some useful functions and packages, and resources that will help you learn how to use them. This section is *not* intended to be a comprehensive tutorial on how to use each function; rather, our aim is to point you in the direction of resources already on the Internet that will help you learn how to use the packages and functions we mention.

Basic Descriptive Statistics

Functions to compute basic descriptive statistics are built into R. These include `mean()`, `median()`, `minimum()`, `maximum()`, `var` for variance, `sd()` for the standard deviation, `IQR()` for interquantile range, `range()`, `quantile()` for percentiles, and `summary()` which for numeric variables returns the minimum, 25th percentile, median, mean, 75th percentile, and maximum. Some examples of these functions in action are given below. See John Quirk's tutorial on using basic descriptive statistics for more information.

```
  # Compute basic descrpitive statistics for transcription score in pen_laptop1 data frame
  # Mean
    mean(pen_laptop1$transcription)
```

```
## [1] 11.53385
```

```r
  # Median
    median(pen_laptop1$transcription) # Median
```

```
## [1] 10.7
```

```r
  # Standard Deviation
    sd(pen_laptop1$transcription)
```

```
## [1] 6.690695
```

```r
  # 0 to 100th percentile in steps of 10%
    quantile(pen_laptop1$transcription, probs = seq(0, 1, .1))
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##  1.00  3.38  6.24  8.50  9.16 10.70 12.04 13.20 17.06 18.92 34.70
```

```r
  # Example of summary function output
    summary(pen_laptop1$transcription)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    8.00   10.70   11.53   15.20   34.70
```

Summary Statistics By Group

You will sometimes want to compute descriptive statistics separately for multiple groups. There are many ways to do this. One option is to use the `group_by()` and `summarise()` functions in the `dplyr` package, for example:

```r
# Compute mean and standard deviation separately for the laptop and pen groups
  library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
  pen_laptop1 %>%
    group_by(group) %>%
    summarise(
      mean = mean(transcription),
      sd = sd(transcription)
    )
```

```
## Source: local data frame [2 x 3]
##
##    group      mean       sd
##   (fctr)     (dbl)    (dbl)
## 1 Laptop 14.519355 7.285576
## 2    Pen  8.811765 4.749339
```

For more information read the see the section on *Grouped Operations* in the dplyr tutorial.

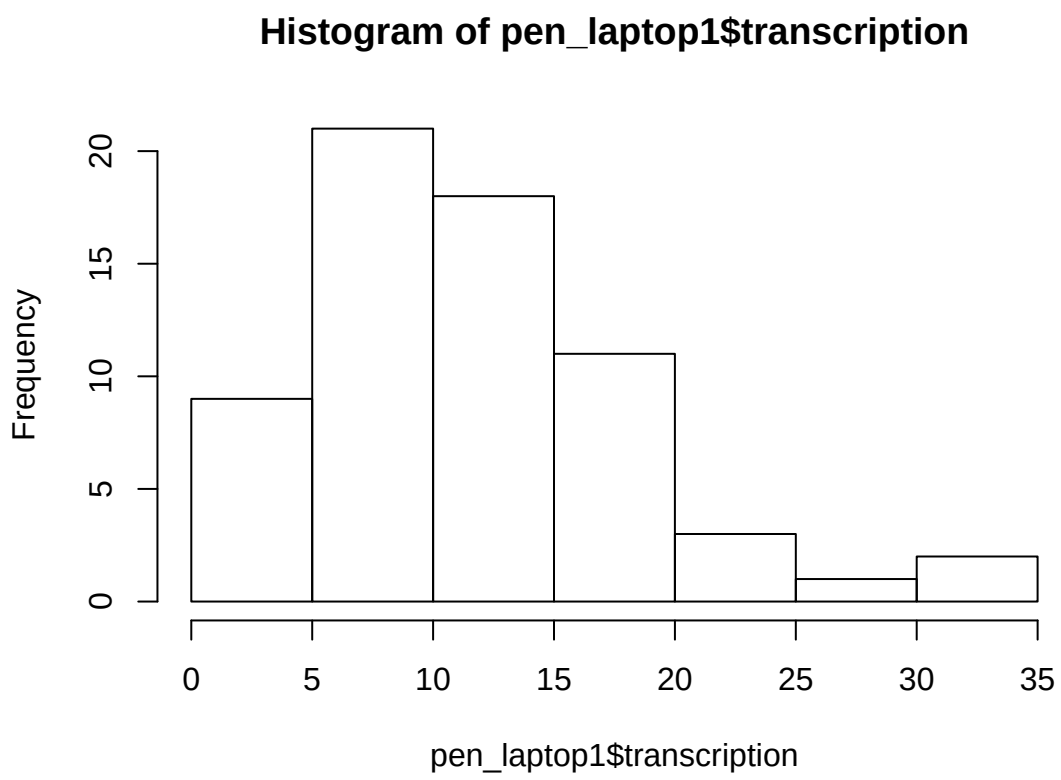
Other options for computing descriptive statistics separately for different groups include using the inbuilt R function aggregate() or the doBy package.

Data Visualisation (Plotting)

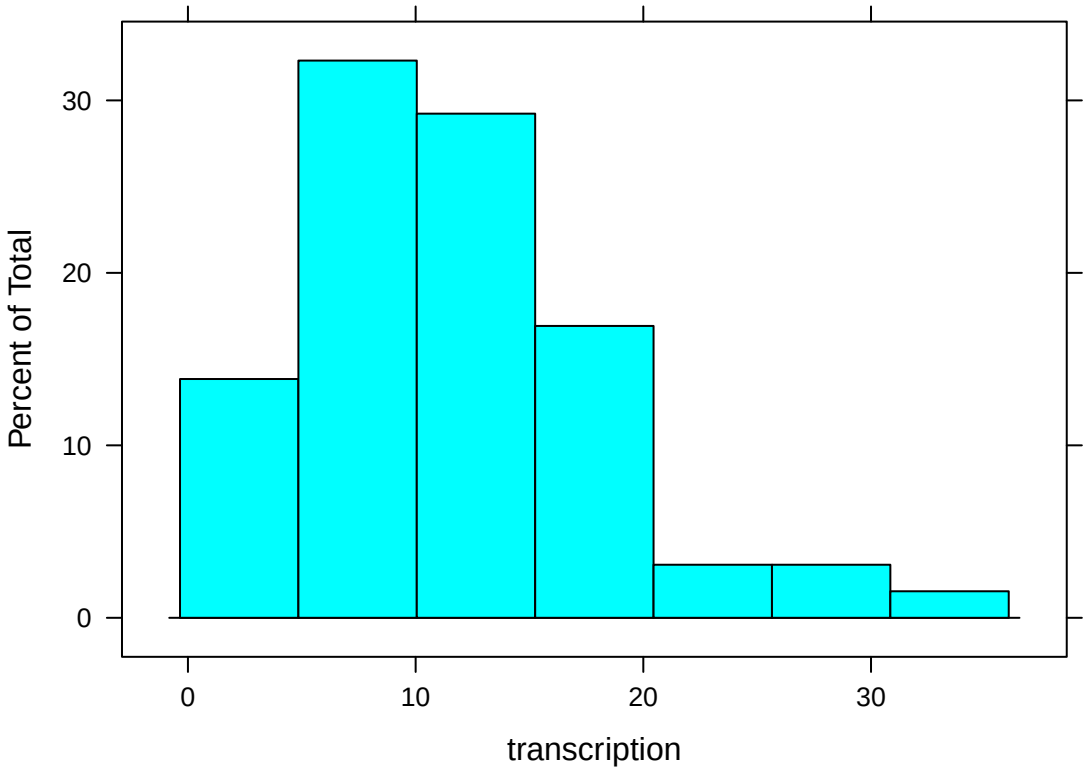
R has three systems that can be used for data visualisation - Base graphics, lattice, and ggplot2. The STHDA wesbite has guides to creating graphics using all three systems.

`Base graphics`, `lattice`, and `ggplot2` all have functions for plotting histograms and dotpolots, covered in Chapter 3 of *Introduction to The New Statistics*. Here are some examples of simple histograms produced by the three packages:

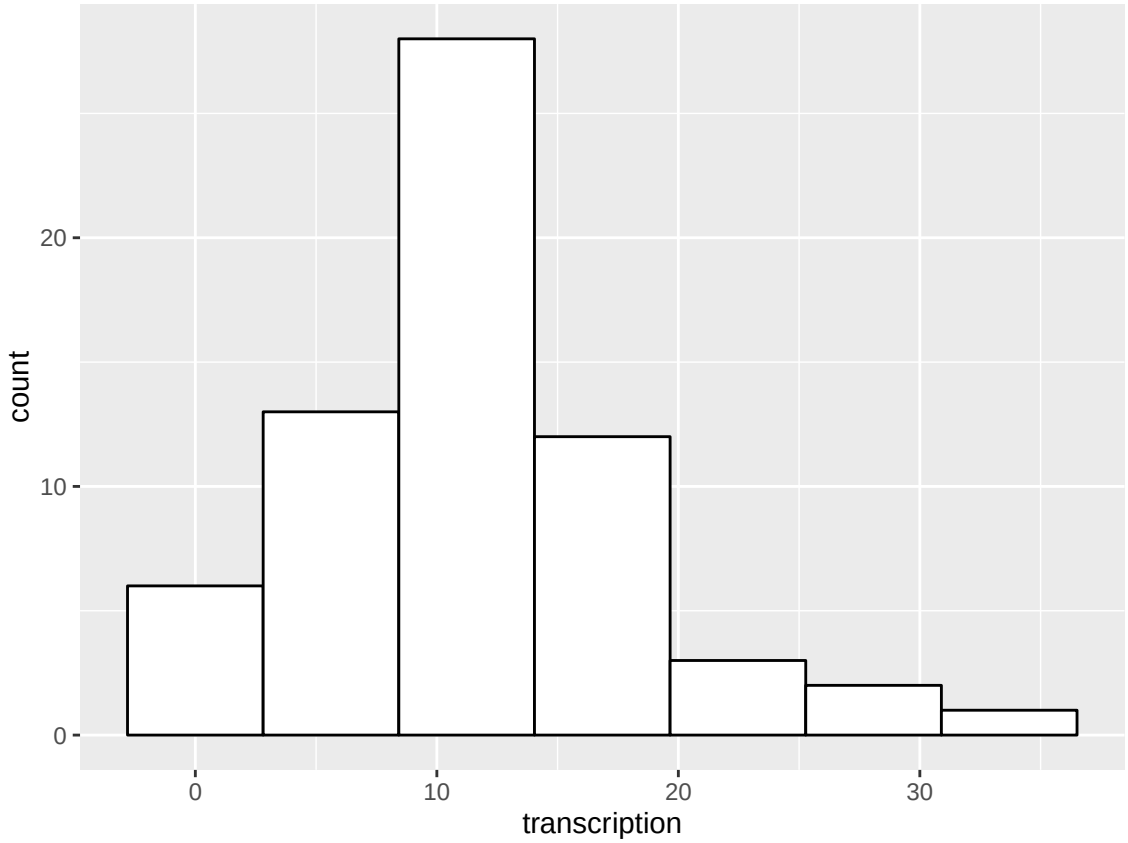
```
# Examples of histograms created using the lattice graphics package
# Base histogram
  hist(pen_laptop1$transcription)
# lattice histogram
  library(lattice)
```



```
histogram(~transcription, data = pen_laptop1)
```




```
# ggplot2
  library(ggplot2)
  ggplot(pen_laptop1, aes(transcription) ) + geom_histogram(bins = 7, colour="black", fill="white")
```



If you are new to R and want to learn one graphics package, we recommend learning how to use `ggplot2` as it is the most powerful and flexible system. Resources that will help you learn how to use `ggplot2` include:

- Winston Chang's R Graphics Cookbook.
- STHDA's ggplot2 essentials.
- Hadley Wickham's ggplot2 book.
- DataCamp's ggplot2 courses.
- Harvard Introduction to ggplot2.
- R4Stats ggplot2 tutorial.
- ggplot2 online documentation.
- R-Studio's Data Visualisation cheatsheet.
- An online workshop about creating publication quality graphics using the ggplot2 and lattice graphics packages by Tim Appelhans.

If you are interested in learning the lattice package, a good place to start is the STHDA Lattice Guide. R-Studio also have a good Guide to R Graphics using lattice. There is also a book about the Lattice package.

ggplot2 histogram and dotplot tutorials

- R Bloggers - How to make a histogram with ggplot2
- STHDA Histogram Tutorial
- STHDA Guide to making dotplots
- ggplot2 documentation for the geom_dotplot() geom

Z-Scores

John Quick's tutorial shows how to use R's inbuilt scale() function to compute Z-scores. See also Seam Dolinar's tutorial on calculating Z scores and finding tail probabilities.

P-values and Confidence Intervals for a Single Sample

Kelly Black has written tutorials showing how to compute p values using z- or t-distributions, and how to calculate confidence intervals for means using normal or t-distributions.

t.test() function

The `t.test()` function is built into R. It produces confidence intervals and p-values for single samples, two independent groups, and paired samples.

```
# Single Sample
t.test(pen_laptop1$transcription)
```

```
##
##  One Sample t-test
##
## data:  pen_laptop1$transcription
## t = 13.898, df = 64, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   9.875973 13.191719
## sample estimates:
## mean of x
##  11.53385
```

```
# Two Independent Groups - by default the Welch T-Test (equal variances not assumed) is calculated
t.test(transcription ~ group, data = pen_laptop1)
```

```
##
##  Welch Two Sample t-test
##
## data:  transcription by group
## t = 3.7031, df = 50.816, p-value = 0.0005254
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.612991 8.802189
## sample estimates:
## mean in group Laptop    mean in group Pen
##            14.519355             8.811765
```

```
# Two Independent Groups - assuming variances are equal
t.test(transcription ~ group, data = pen_laptop1, var.equal = TRUE)
```

```
##
##  Two Sample t-test
```

```
##
## data:  transcription by group
## t = 3.7738, df = 63, p-value = 0.0003579
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.685265 8.729915
## sample estimates:
## mean in group Laptop    mean in group Pen
##            14.519355             8.811765
```

```
# Paired Samples
t.test(thomason1$pre, thomason1$post, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  thomason1$pre and thomason1$post
## t = -3.8555, df = 11, p-value = 0.002674
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.618115 -0.715218
## sample estimates:
## mean of the differences
##               -1.666667
```

MBESS package

Ken Kelly's MBESS (Methods for the Behavioural and Social Sciences) package contains numerous functions for computing confidence intervals for many effect sizes, including standardised mean differences, mean contrasts in one-way and factorial designs, unstandardised and standardised regression coefficients, R-squared, etc. The MBESS also includes functions for power analysis and sample size planning for precision. The MBESS website contains links to two journal articles about the package, and help files.

Below is an example of how a confidence interval for the standardized mean difference can be computed using MBESS.

```
# Use dplyr package to extract transcription scores for the laptop and pen groups in the pen_laptop1 da
   # library(dplyr) # load dplyr if it is has not already been loaded
    laptop <- pen_laptop1 %>% filter(group == "Laptop")
    pen <- pen_laptop1 %>% filter(group == "Pen")
# Load MBESS library
    library(MBESS)
# Use the smd() function to compute Cohen's d
    # Biased Estimate
    d_biased <- smd(laptop$transcription, pen$transcription)
    d_biased
```

```
## [1] 0.9371681
```

```
  # Unbiased estimate
    d_unbiased <- smd(laptop$transcription, pen$transcription, Unbiased = TRUE)
    d_unbiased
```

```
## [1] 0.9259595
```

```
# Use ci.smd() to compute a 95% confidence interval for the biased estimate
    ci.smd(smd = d_biased, n.1 = nrow(pen), n.2 = nrow(laptop))
```

```
## $Lower.Conf.Limit.smd
## [1] 0.4204238
##
## $smd
## [1] 0.9371681
##
## $Upper.Conf.Limit.smd
## [1] 1.447208
```

```
# Repeat for the unbiased estimate
    ci.smd(smd = d_unbiased, n.1 = nrow(pen), n.2 = nrow(laptop))
```

```
## $Lower.Conf.Limit.smd
## [1] 0.4098655
##
## $smd
## [1] 0.9259595
##
## $Upper.Conf.Limit.smd
## [1] 1.435414
```

Meta-Analysis

There are numerous R packages that can be used to conduct meta-analyses for a wide variety of effect sizes such as means, mean differences, standardized mean differences, proportions, odds rations, etc. See the CRAN Meta-Analysis Task View for a comprehensive list of them.

A popular and well documented package for conducting meta-analyses in R is metafor. See the comprehensive metafor website for more information.

metagear is a relatively new package which has meta-analytic capabilities as well as functions that help users conduct systematic reviews and generate PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) flow charts. This vignette provides an overview of the metagrear package.

Other useful sources of information about conducting meta-analyses in R include:

- A.C Del Re's Practical Tutorial on conducting Meta-Analysis in R using the metafor and MAd packages
- Stephanie Kovalchik's Tutorial on Meta-Analysis in R from the 2013 useR! Conference
- Schwarzer, Carpenter, and Rucker's Meta-Analysis with R book
- R-Studio's tutorial on running meta-analyses in R using the metafor package
- Simon Knight's Guide to Meta-Analysis in R - part 1 and part 2.
- Stephanie Hick's Easy Introduction to Meta-Analysis in R using the meta package

multicon plots

For Cat's Eye Pictures, may also integrate into graphics section

egraph() - A function for plotting means as dots and error bars without caps around them catseye() - A function for creating cat's eye plots of group means diffPlot() - A function for creating difference plots for two group (both paired and independent) comparisons

Correlation

Basic scatterplot Adding a regression line

CIs on correlations CI on difference between two independent correlations Figure to display difference in two ind corr

Regression

Calculation and figure for regression line in scatterplot

CI on the regression slope

lm() is the inbuilt R function for ordinary least-squares regression. confint() gives confidence intervals on regression parameters

Also MBESS can do this (apaprently) as well as for standardized coefficients.

Link to graphics.

Show ggplot2 scatterplot with OLS line, maybe also show smoother, mention it can also fit robust and other regression lines.

Also countless other packages for fitting a variety of regression models in R.

Prediction intervals for individual values of Y at particular X values

Categorical Data - Frequencies, Proportions, and Risk

CI for a proportion CI for difference between two independent proprtions Figure for CI on a single proportion Figure for difference with CI

ESCI uses Newcombe 1998 methods

Ratio between two variables - frequency tables CI on the difference Chi Square / Phi Coefficients

vcd and vcdExtra packages for visualising categorical data

PropCIs Does CIs for single, independent, and paired proportions. Includes risk score CI.

library(pairwiseCI) # for ARR NHS and RR using Score Method. Uses Prop.diff and Prop.ratio

And chi-square function I used for recent WCBCT analyses.

R manual to accompany Agresti's Categorical Data Analysis 2nd Ed by Laura Thompson http://www.stat.ufl.edu/~aa/cda/Thompson_manual.pdf

Extended Designs - One-Way and Factorial Designs

Independent Variable One way independent groups design CI for planned contrasts, on two means Figures if possible

One way repeated esigns, if possible (beyond scope of book and ESCI)

ANOVA p values

Extended Designs - Two Indepenent Variables

ESCi only does 2 x 2 design Means adn CI on mean effects CI on single DF interaction (as difference in differences)

Simple man effect

Nice to include two way repeated measures if possible

Robust Methods

WRS and WRS2 packages

Trimmed means for two independent means

BootES https://web.williams.edu/Psychology/Faculty/Kirby/bootes-kirby-gerlanc-in-press.pdf

Questions for Geoff and Bob

Datasets I can't find referenced in text

Name	Section
flag_priming	Ch 7 ???
super_golf	Ch 7 ???
habituation	Ch 8 ???
learning_genes	Ch 8 ???
sensitization	Ch 8 ???
ma_gambler_fallacy	Ch9 ???
ma_anchor_adjust_chicago	Ch9 ???
ma_anchor_adjust_everest	Ch9 ???
ch11_ex7	Ch 11 ???
ch12_ex3	Ch 12 ???
inauthentic	Ch 14 ???
iqboost	Ch 14 ???
blame1	Ch 15 ???
blame2	Ch 15 ???