UNIVERSITY OF LEEDS

An Introduction to R

Ed D. J. Berry

9th January 2017

Overview

UNIVERSITY OF LEEDS

- Why now?
- Why R?
- General tips
- Recommended packages
- Recommended resources

Why now?

Efficiency

- Point-and-click software just isn't time efficient
- Automating tasks will pay off within the time frame of a PhD and thereafter

Why now?

Reproducibility

- There is an increasing expectation that materials, data, and analysis details are provided alongside research to ensure it is reproducible
  - This is easier when things are script based
- Peer Reviewers' Openness Initiative

Why R?

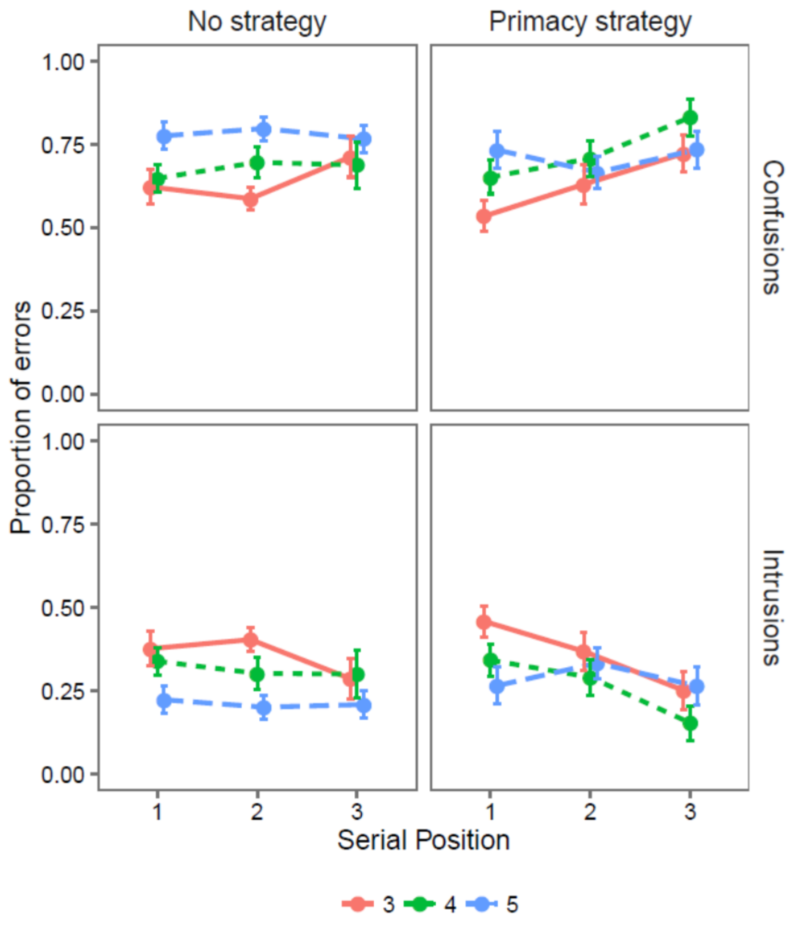
Jobs

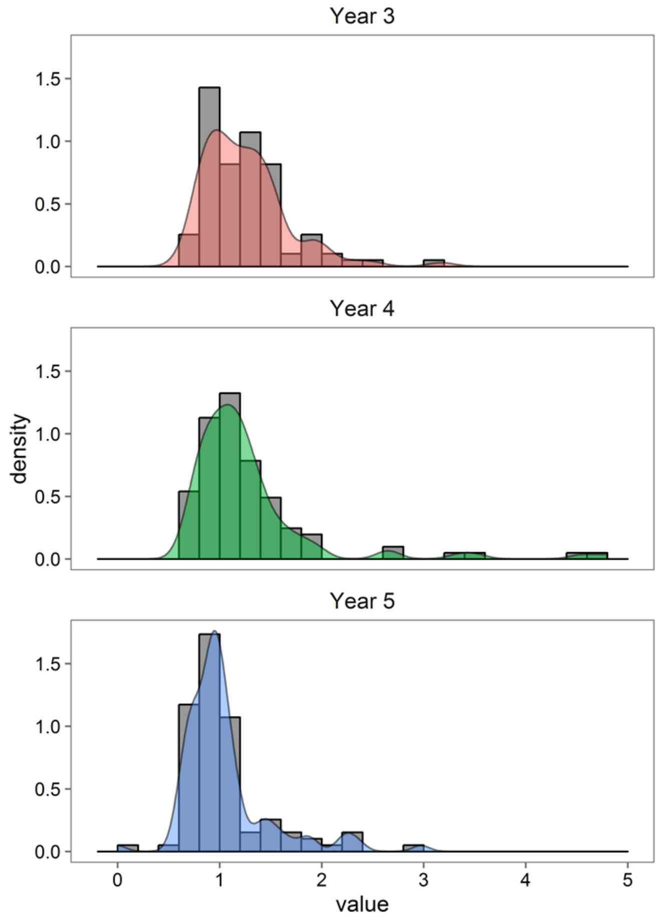
- R is increasingly taught in Psychology departments, including at undergraduate level
- Useful skill for jobs outside academia
- Makes you a more efficient academic

Why R?

UNIVERSITY OF LEEDS

Pretty graphs





Why R?

Range of packages

- There are R packages for a huge range of analyses
- Great data manipulation packages
- Slides
- Documents
    - Including books
- Interactive HTML applications

Why R?

Reproducibility… again

- R projects
- R Markdown

Why R

UNIVERSITY OF LEEDS

- It's free
- Big community
- R has the happiest commenters

Recommended packages

General comments

UNIVERSITY OF LEEDS

- Given the age of R there are many ways to complete a task
- Most data manipulation tasks can be done with 'base R'
  - However, this often isn't the most efficient or readable approach

tidyverse

- A collection of packages by Hadley Wickham for:
  - Data visualisation (ggplot2)
  - Data manipulation (dplyr)
  - Data tidying (tidyr)
  - Importing data (readr)
  - Functional programming (purrr)
    - See here for a full list of the included packages
- These packages are all designed to work nicely together
- More readable by people than most R code

Installing tidyverse

- To install and load any package you just do:

```
install.packages("tidyverse")
library(tidyverse)
```

- You need to load a package in with `library()` for any new R session you want to use it with
- Loading tidyverse loads all the packages described previously

Recommended packages

UNIVERSITY OF LEEDS

The pipe operator

- The pipe operator is key to why the tidyverse packages are so usable and readable
- It passed the thing on its left as the first argument to a function on its right
  - x %>% f() is equivalent to f(x)

```r
x <- c(10, 5, 15)

mean_x <- x %>% mean()
```

- This is amazing for chaining together the various steps some data goes through without needing to create intermediary objects
- Doesn't work so smoothly with some packages

ggplot2

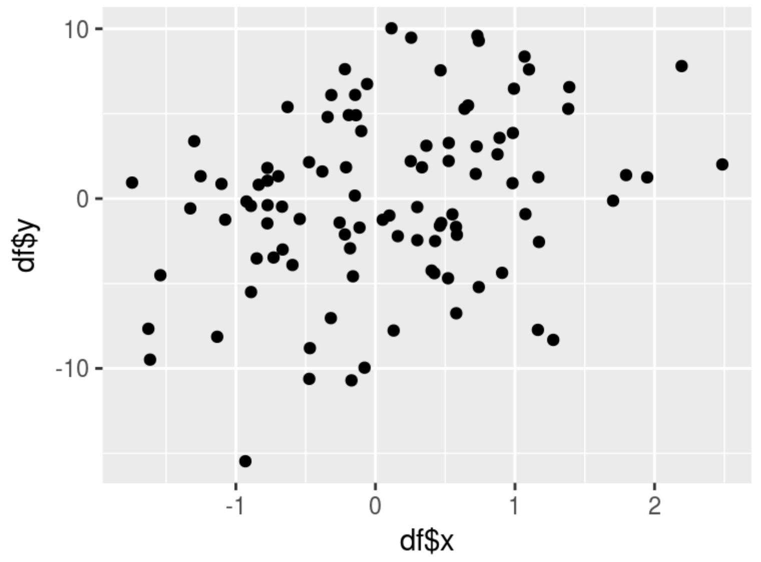
UNIVERSITY OF LEEDS

- Build graphs by specifying:
  - Aesthetics: physical properties of the plot mapped to variables in the data (x & y positions, size, shape, colour etc.)
  - Geometries: what to actually use to represent the data (lines, bars, points etc.)

ggplot2

qplot

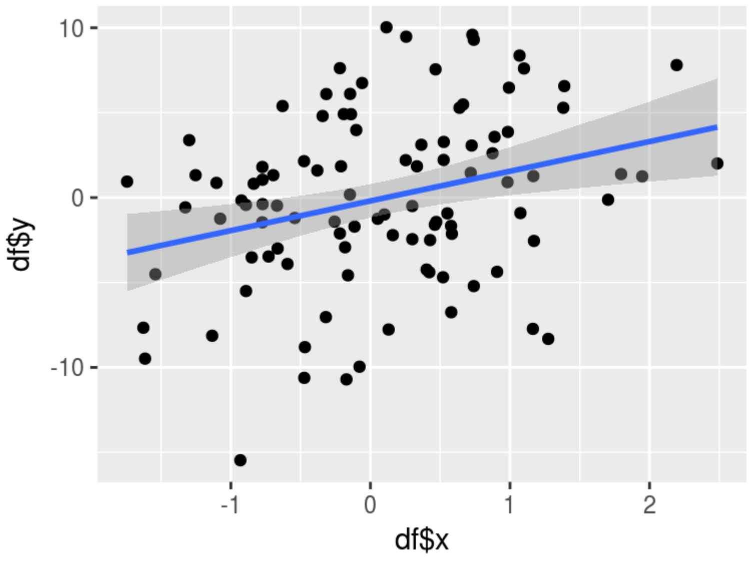
```
qplot(x = df$x, y = df$y)
```



ggplot2

qplot

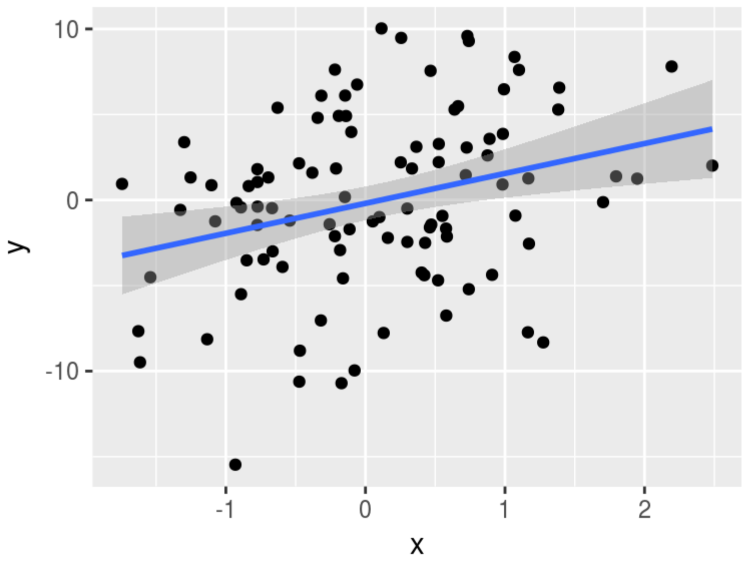
```
qplot(df$x, df$y) +
  geom_smooth(method = "lm")
```



ggplot2

ggplot

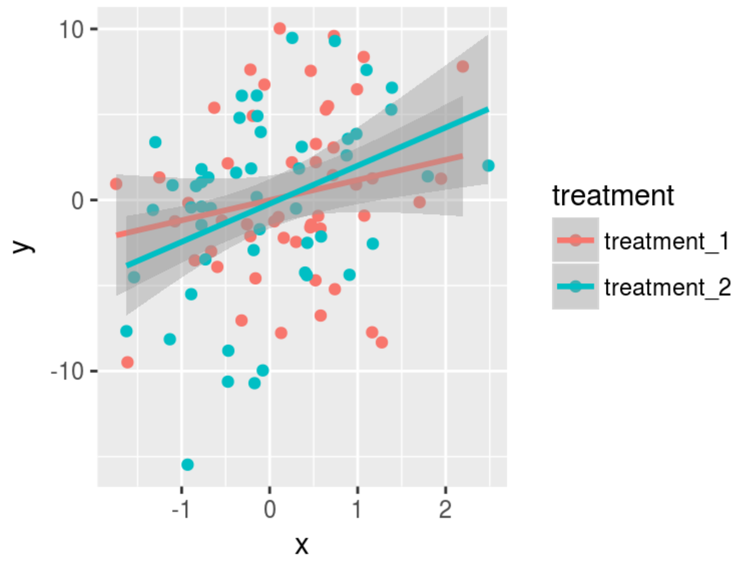
```
ggplot(data = df, mapping = aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(method = "lm")
```



ggplot2

ggplot

```
ggplot(data = df, mapping = aes(x = x, y = y, colour = treatment)) +
  geom_point() +
  geom_smooth(method = "lm")
```



ggplot2

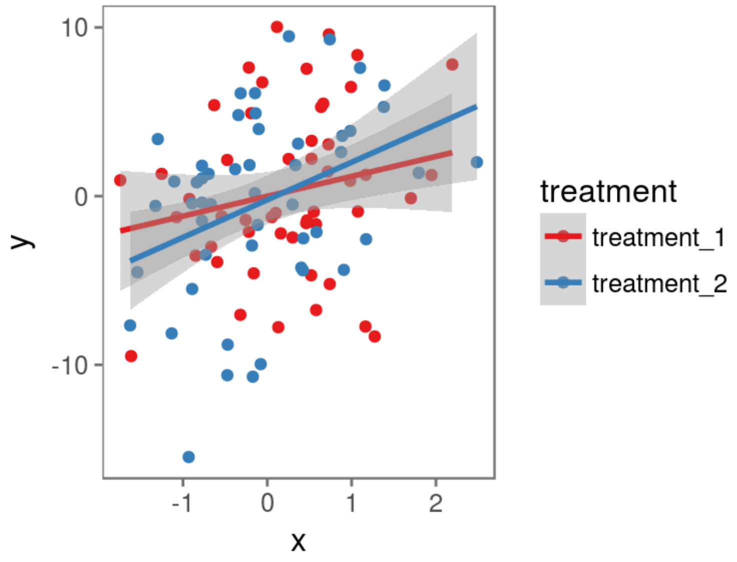
Other tips

- The package `ggthemes` is good for providing premade plot 'styles'
- `RColorBrewer` is useful for colours
  - Useful info on colour in `ggplot2` here
- `cowplot` is good for creating grids of labelled plots for papers
  - `cowplot` vignette

ggplot2

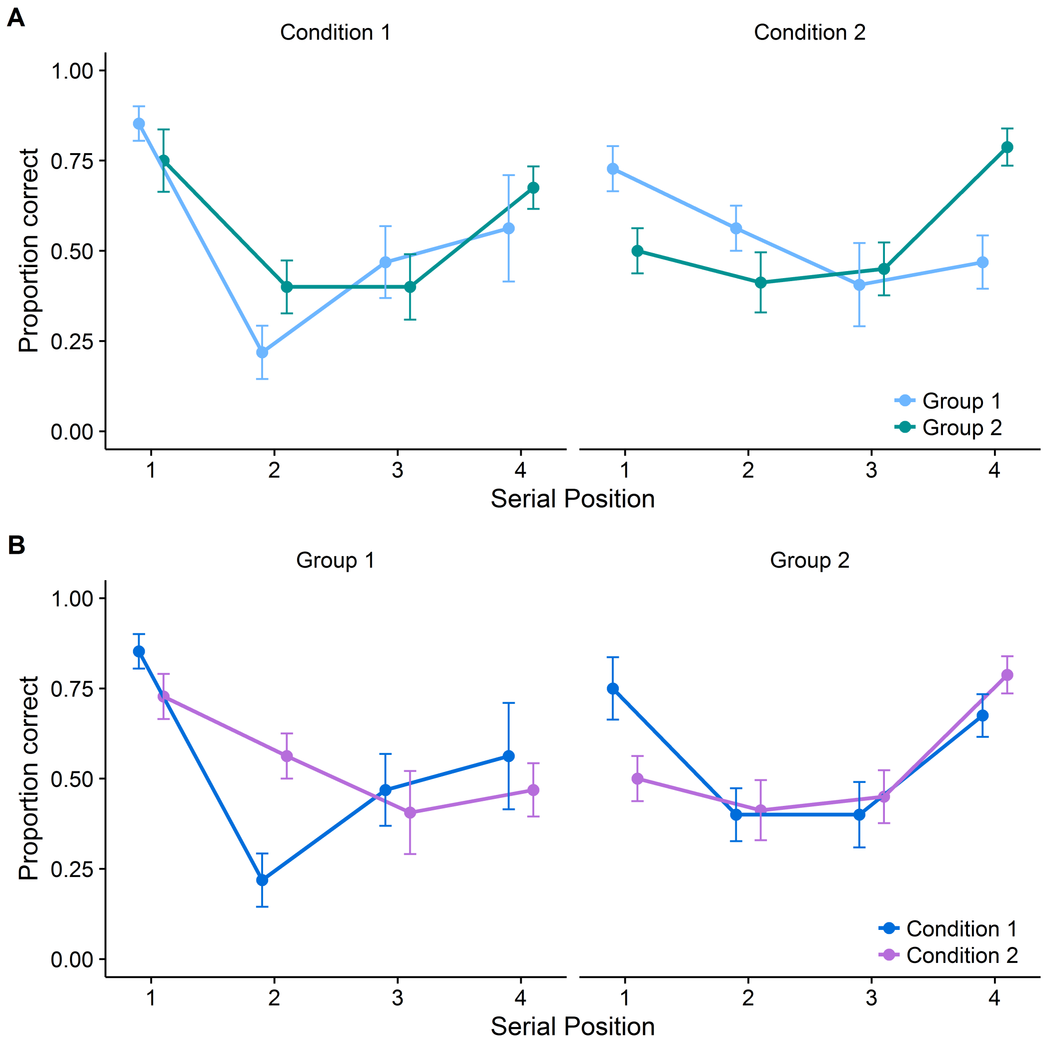
Other tips

```
ggplot(df, aes(x, y, colour = treatment)) +
  geom_point() +
  geom_smooth(method = "lm") +
  theme_few() +
  scale_color_brewer(palette = "Set1")
```



cowplot

UNIVERSITY OF LEEDS



Data visualisation

UNIVERSITY OF LEEDS

Other options

- Three main options for data visualation: `base`, `lattice`, and `ggplot2`
- `base` automatically produces certain plots when called on certain objects
  - e.g. calling `plot()` on a regression model object will produce diagnostic plots
- In my view `ggplot2` is the easiest to learn - but that's probably because it the only one I'm good at!
  - See these posts for arguments for and against `ggplot2` over `base` for plots

dplyr

overview

- `dplyr` is designed around a set of basic 'verbs':
  - `filter()`: filter rows
  - `arrange()`: arrange rows (e.g. ascending)
  - `select()`: select columns
  - `distinct()`: get unique rows
  - `mutate()`: create new variables
  - `summarise()`: summarise the data
- Also has functions for joining data and lots of 'helper' functions

dplyr

UNIVERSITY OF LEEDS

Some example data

```
## # A tibble: 10 x 5
##       id    stage      cond1        cond2  group
##    <int>    <chr>      <dbl>        <dbl>  <chr>
##  1     1 practice  0.1203974 -0.394476858 group1
##  2     1     test  0.8622419  0.112896796 group1
##  3     2 practice  0.5662425 -0.069661281 group1
##  4     2     test -0.9968107  0.733580258 group1
##  5     3 practice -0.3010821  0.892817363 group1
##  6     3     test -0.9256125 -0.015851477 group1
##  7     4 practice  1.2274515 -0.870920015 group1
##  8     4     test  0.7435982 -0.007121835 group1
##  9     5 practice -0.1309911 -0.650193954 group1
## 10     5     test  0.6061486  1.444081676 group1
```

dplyr

Summarising the data

UNIVERSITY OF LEEDS

```
sum_stats <- df1 %>%
  filter(stage == "test") %>%
  mutate(cond_diff = cond1 - cond2) %>%
  group_by(group) %>%
  summarise(mean = mean(cond_diff),
            sd = sd(cond_diff),
            n = n(),
            se  = sd/sqrt(n))
```

```
## # A tibble: 2 x 5
##    group       mean       sd     n        se
##    <chr>      <dbl>    <dbl> <int>     <dbl>
## 1 group1 -0.9246006 1.197004    15 0.3090652
## 2 group2 -0.3049516 1.264237    15 0.3264246
```

purrr

overview

- `purrr` is a package for 'functional programming'
- The functions you're likely to use most are the `map()` functions
  - Apply a function to a list, vector or dataframe
  - Have versions where you specify the class of the object you're expecting back
    - 'Safer' than the `apply` family
      - Either work or break with an informative error message
- Lots of other functions that are useful for writing your own functions
- Cool purrr tutorial here

purrr

example

```r
list <- paste("data/", list.files("data"), sep = "")

df <- map_df(list, read_csv)
```

tidyr

Overview

- Functions for tidying data
- The thing to use for moving between long and wide data
- E.g. suppose we have the wide data from before

```
## # A tibble: 6 x 5
##      id stage      cond1        cond2  group
##   <int> <chr>      <dbl>        <dbl>  <chr>
## 1     1  test  0.8622419  0.112896796 group1
## 2     2  test -0.9968107  0.733580258 group1
## 3     3  test -0.9256125 -0.015851477 group1
## 4     4  test  0.7435982 -0.007121835 group1
## 5     5  test  0.6061486  1.444081676 group1
## 6     6  test -1.3852922  1.179975817 group1
```

tidyr

Example

```
df1_long <- df1 %>%
  gather(condition, score, cond1:cond2) %>%
  arrange(id)
```

```
## # A tibble: 6 x 5
##      id    stage  group condition      score
##   <int>    <chr>  <chr>     <chr>      <dbl>
## 1     1 practice group1     cond1  0.1203974
## 2     1     test group1     cond1  0.8622419
## 3     1 practice group1     cond2 -0.3944769
## 4     1     test group1     cond2  0.1128968
## 5     2 practice group1     cond1  0.5662425
## 6     2     test group1     cond1 -0.9968107
```

Recommended packages

broom

- For cleaning up the outputs of modelling functions (vignette)
- Work very well with dplyr (vignette)

```
## # A tibble: 8 x 5
##      id  year      memory   attention attainment
##   <int> <chr>       <dbl>       <dbl>      <dbl>
## 1     1  five -0.05477005 -1.20337765 -0.2399355
## 2     2  five  0.54707674 -0.94270907 -0.2753261
## 3     3  five  0.98387147  0.33104915  0.2411040
## 4     4  five  0.27871988  0.30906999  0.7614589
## 5     5  five  1.57665149 -0.09960154  2.6852044
## 6     6  five  1.00881206 -0.61045548  0.1791086
## 7     7  five -0.66699081  1.41081796  0.1678306
## 8     8  five  0.41851488  1.87813642  1.7091251
```

Recommended packages

UNIVERSITY OF LEEDS

broom example (adapted from vignette)

```
df2 %>%
  group_by(year) %>%
  do(tidy(lm(attainment ~ memory + attention, data = .)))
```

```
## # A tibble: 6 x 6
## # Groups:   year [2]
##    year        term     estimate std.error   statistic      p.value
##   <chr>       <chr>        <dbl>     <dbl>       <dbl>        <dbl>
## 1  five (Intercept)  0.033675914 0.1917730  0.17560298 8.619162e-01
## 2  five      memory  0.755290501 0.2160890  3.49527562 1.653449e-03
## 3  five   attention  0.491667705 0.2118922  2.32036765 2.811909e-02
## 4   two (Intercept) -0.005000834 0.1982218 -0.02522847 9.800583e-01
## 5   two      memory  1.088468229 0.1702252  6.39428304 7.537514e-07
## 6   two   attention  0.903462260 0.1851783  4.87887852 4.217471e-05
```

Recommended packages

rmarkdown

- `rmarkdown` provides a range of tools for creating dynamic documents in R (see this intro)
- Can be used to create:
  - Reports (e.g. a paper)
    - Outputs to MS Word, PDF, or HTML
  - Slides
  - Interactive Notebooks
  - Books via `bookdown`
  - See here for a full list of formats

Recommended packages

rmarkdown

- Code can be embedded to make reproducible reports

74 participants took part in Experiment 1 (Mean age = **8.65**, SD = **1.32**).

`` `r nrow(wide)` `` participants took part in Experiment 1 (Mean age = `` `r round(mean(wide$age), 2)` ``, SD = `` `r round(sd(wide$age), 2)` ``).

- See here for a useful function I've written for rounding values in R Markdown

General tips

UNIVERSITY OF LEEDS

R Studio

- Lots of useful stuff for R Markdown
- Generally just nice
- Download here: https://www.rstudio.com/products/rstudio/download/
- Some tips for using R Studio

General tips

UNIVERSITY OF LEEDS

R Projects

- Great for organising bits of code related to a single project
  - Raw data, processing script, processed data, analysis scripts, and manuscript all in one place
- Git intergration
- With a bit of code at the top of the manuscript you have a fully reproducible workflow
- Introduction to projects here

```
source("scripts/data-processing.R")
source("scripts/analysis.R")
source("scripts/exploratory-analysis.R")
```

General tips

Organising projects

- Some useful advice here on setting up a project (plus Git)
  - The advice for organisation is to have folders for data, figures, scripts and write-up
- When working with a project the top folder is your working directory. This makes it very easy to call files from other folders

```
read_csv("data/exp-data-07-16.csv")
```

UNIVERSITY OF LEEDS

General tips

Projects and R markdown

- Note that when you run an R Markdown file the working directory is changed to the location of that file.
- if we have our file in a subfolder we need to change the way we call a file in different subfolder

```r
source("../scripts/data-processing.R")
```

- `"../"` tells it to start at the parent folder of the current working directory

General tips

Reuse and improve

- Obviously the point of writing scripts is that they can be reused
- However, don't just settle for whatever worked first. Try to work out the best way to do something
    - 'If you aren't getting frustrated you aren't learning' - Hadley Wickham
- Trying to improve old code when you revisit is a useful exercise

General tips

UNIVERSITY OF LEEDS

Misc tips

- Find out all the stuff (e.g. functions) in a package loaded in with `library()`
  - `ls("package:dplyr")`
- Show the documentation for a function
  - `?select`
- Save and object and also print it to the console
  - `(x <- rnorm(10))`
- Shortcut for `<-`
  - `Alt+-`
- Re-run the code that was last run (great when, e.g., developing a plot)
  - `Ctrl+Shift+P`

Recommended resources

Recommended resources

UNIVERSITY OF LEEDS

Datacamp

- This is how I did most of my learning
- They have loads of great courses, particularly:
  - Introduction to R & Intermediate R
  - Data Manipulation in R with dplyr, Joining Data in R with dplyr & Cleaning Data in R
  - Data Visualisation with ggplot2
  - Writing Functions in R
  - Reporting with R Markdown
- Some of the courses can be tried for free
- Free when used for teaching (see here)

Recommended resources

UNIVERSITY OF LEEDS

Online stuff

- Coursera Data Science Specialisation
  - I've not done it but hear that it's good.
- R Cookbook
  - Though some of the solutions are now a bit out of date
- R Studio blogs
  - Lots of good blogs accompany package updates.

Recommended resources

Online stuff

- Stack Overflow
  - You'll end up there by Googling a question
- R Studio cheatsheets
- R Studio R Markdown website
- There are loads of good blogs out there that you can find from Googling
  - Note the date of a blog post as the advice might be out of date

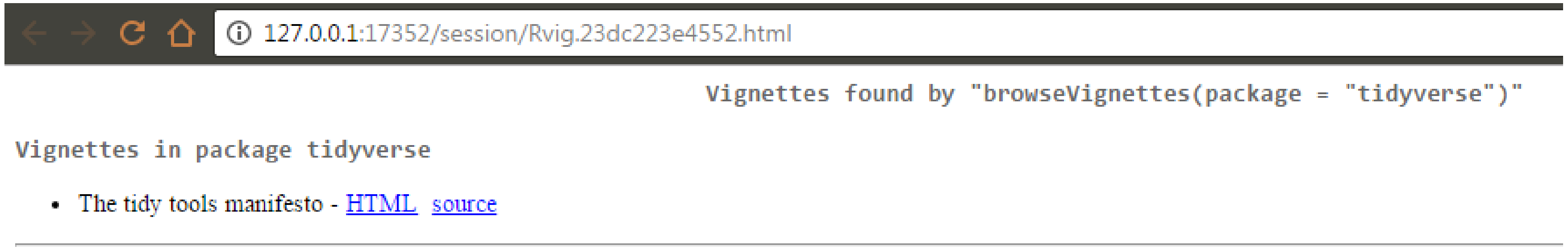
Recommended resources

UNIVERSITY OF LEEDS

Online stuff

- Package vignettes
    - E.g. Intro to dplyr

```
browseVignettes(package = "tidyverse")
```



127.0.0.1:17352/session/Rvig.23dc223e4552.html

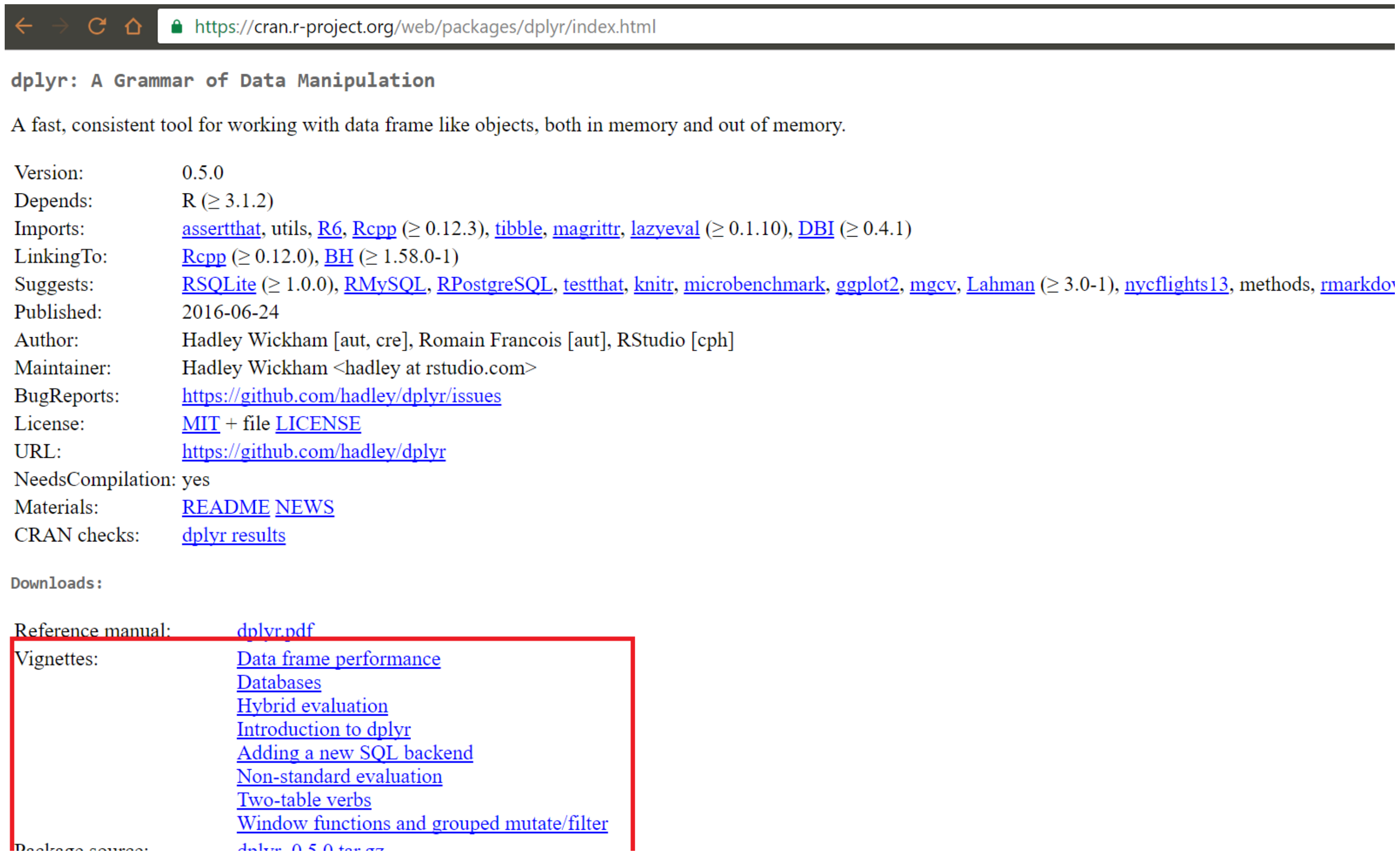
```
Vignettes found by "browseVignettes(package = "tidyverse")"

Vignettes in package tidyverse
```

- The tidy tools manifesto - HTML source

Recommended resources

Finding vignettes via Googling



https://cran.r-project.org/web/packages/dplyr/index.html

dplyr: A Grammar of Data Manipulation

A fast, consistent tool for working with data frame like objects, both in memory and out of memory.

Version:	0.5.0
Depends:	R (≥ 3.1.2)
Imports:	assertthat, utils, R6, Rcpp (≥ 0.12.3), tibble, magrittr, lazyeval (≥ 0.1.10), DBI (≥ 0.4.1)
LinkingTo:	Rcpp (≥ 0.12.0), BH (≥ 1.58.0-1)
Suggests:	RSQLite (≥ 1.0.0), RMySQL, RPostgreSQL, testthat, knitr, microbenchmark, ggplot2, mgcv, Lahman (≥ 3.0-1), nycflights13, methods, rmarkdov
Published:	2016-06-24
Author:	Hadley Wickham [aut, cre], Romain Francois [aut], RStudio [cph]
Maintainer:	Hadley Wickham <hadley at rstudio.com>
BugReports:	https://github.com/hadley/dplyr/issues
License:	MIT + file LICENSE
URL:	https://github.com/hadley/dplyr
NeedsCompilation:	yes
Materials:	README NEWS
CRAN checks:	dplyr results

Downloads:

Reference manual:	dplyr.pdf
Vignettes:	Data frame performance Databases Hybrid evaluation Introduction to dplyr Adding a new SQL backend Non-standard evaluation Two-table verbs Window functions and grouped mutate/filter
Package source:	dplyr_0.5.0.tar.gz

UNIVERSITY OF LEEDS

Recommended resources

Books

- R for Data Science by Garrett Grolemund & Hadley Wickham
- bookdown: Authoring Books and Technical Documents with R Markdown by Yihui Xie
- Dynamic Documents with R and Knitr by Yihui Xie
- Discovering Statistics Using R by Andy Field
  - Published in 2012 so the stuff on data manipulation, in particular, is out of date

Where to get the code for these slides

UNIVERSITY OF LEEDS

- https://github.com/eddjberry/intro-to-R-talks
- See the file intro-to-R.Rmd
    - There is also a knitted HTML version of the slides on GitHub