R for Research

Essential Tools for Researchers

Internetrix

Who am I?

Dean Marchiori
Head of Data Science
Internetrix

https://deanmarchiori.github.io/aboutme/
dean.marchiori@internetrix.com.au

Internetrix

At Internetrix, we provide practical solutions to complex problems. From spreadsheets to machine learning, we are experts in the mathematics of making smarter decisions.

Data Science Consulting | Analysis Projects | Training and Team Development | Strategy and Transformation

https://www.internetrix.com.au/services/data-science/

Key principles

```
shareable -> reproducible -> publishable
```

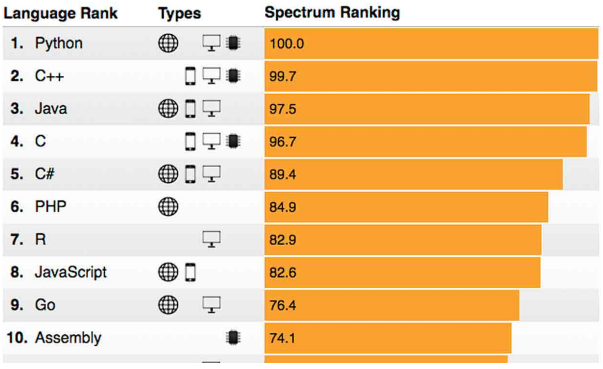
Why R?

R

The R programming language is a popular and open-source tool for data analysis and statistical computing.

https://www.r-project.org/

- Can handle data import, cleaning, analysis, visualisation and publishing
- Highly extensible through package ecosystem - new algos available faster
- Open Source
- Easy to learn from a non-CS background
- GREAT community and documentation



https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages

Getting Data

R packages to get data and access API's

Web Technologies

R has a detailed task view of packages build to interface with the web.

- HTTP Requests
- XML / JSON
- Web Scraping
- Cloud Data Tools (AWS, BigQuery, ..)
- Social Media Clients

https://cran.r-project.org/web/views/WebTechnologies.html

bomrang

Australian Government Bureau of Meteorology (BOM) Data Client.

Provides functions to interface with Australian Government Bureau of Meteorology (BOM) data.

Install from CRAN:

```
install.packages("bomrang")
```

bomrang demo

Get current forecast for Wollongong

```
library(bomrang)
library(dplyr)

weather <- bomrang::get_current_weather(station_name = 'Bellambi')

weather %>%
  filter(local_date_time_full == max(local_date_time_full)) %>%
  select(full_name, local_date_time_full, air_temp, wind_dir, wind_spd_kt)
```

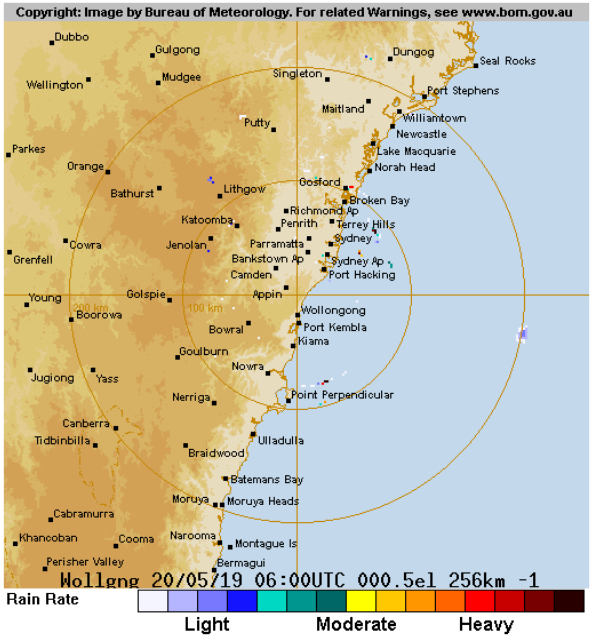
```
##   full_name local_date_time_full air_temp wind_dir wind_spd_kt
## 1  Bellambi  2019-05-20 15:30:00     21.2      NNE           4
```

bomrang demo

Get radar imagery

```
library(bomrang)

imagery <- get_radar_imagery(product_id = "IDR032", path = 'img/radar.png')
```



Analysing Data

faster and more interpretable EDA

skimr

skimr provides a frictionless approach to summary statistics which conforms to the principle of least surprise, displaying summary statistics the user can skim quickly to understand their data.

Install the latest development version:

```
devtools::install_github("ropensci/skimr")
```

skimr demo

```r
library(skimr)

skimr::skim(iris)
```

```
## Skim summary statistics
##  n obs: 150
##  n variables: 5
##
## ── Variable type:factor ──────────────────────────────────────────────────────────────
##  variable missing complete   n n_unique                        top_counts ordered
##   Species       0      150 150        3 set: 50, ver: 50, vir: 50, NA: 0   FALSE
##
## ── Variable type:numeric ─────────────────────────────────────────────────────────────
##      variable missing complete   n mean   sd  p0 p25  p50 p75 p100     hist
##  Petal.Length       0      150 150 3.76 1.77 1   1.6 4.35 5.1  6.9 ▇▁▅▇▇▅
##   Petal.Width       0      150 150 1.2  0.76 0.1 0.3 1.3  1.8  2.5 ▇▁▆▅▁▂
##  Sepal.Length       0      150 150 5.84 0.83 4.3 5.1 5.8  6.4  7.9 ▅▇▆▇▅▂
##   Sepal.Width       0      150 150 3.06 0.44 2   2.8 3    3.3  4.4 ▂▅▇▅▂▁
```

The tidyverse

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Ideally suited for:

- Reading in data
- Manipulating and tidying data
- Visualisation of results

https://www.tidyverse.org/

Install from CRAN

```
install.packages('tidyverse')
```

tidyverse demo

This example shows some more advanced use of data manipulation and visualisation.

Space launches

These are the data behind the "space launches" article, The space race is dominated by new contenders. Principal data came from the Jonathan McDowell's JSR Launch Vehicle Database, available online at http://www.planet4589.org/space/lvdb/index.html.

Example adapted from from https://github.com/dgrtwo/data-screencasts/blob/master/space-launches.Rmd

and the Tidy Tuesday Project

Reading in data

```r
library(tidyverse)

launches <- read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/mast

head(launches)
```

```
## # A tibble: 6 x 11
##   tag            JD launch_date launch_year type          variant mission
##   <chr>       <dbl> <date>            <dbl> <chr>         <chr>   <chr>
## 1 1967-065 2439671. 1967-06-29         1967 Thor Burner 2 <NA>    Secor Type II S/N 10
## 2 1967-080 2439726. 1967-08-23         1967 Thor Burner 2 <NA>    DAPP 3419
## 3 1967-096 2439775. 1967-10-11         1967 Thor Burner 2 <NA>    DAPP 4417
## 4 1968-042 2440000. 1968-05-23         1968 Thor Burner 2 <NA>    DAPP 5420
## 5 1968-092 2440153. 1968-10-23         1968 Thor Burner 2 <NA>    DAPP 6422
## 6 1969-062 2440426. 1969-07-23         1969 Thor Burner 2 <NA>    DAPP 7421
```

Pre-processing

- Use of the pipe %>% operator improved readability with collaborators
- Ability to filter, aggregate, select columns, reorder, text manipulation
- Supports Non-Standard Evaluation (NSE)

```
launches_processed <- launches %>%
  filter(launch_date <= Sys.Date()) %>%
  filter(state_code == "US") %>%
  add_count(type) %>%
  filter(n >= 20) %>%
  mutate(type = fct_reorder(type, launch_date, min),
         agency_type = str_to_title(agency_type))
```

Data Visualisation with ggplot2

`ggplot2` is a well known package for data visualisation based on The Grammar of Graphics.

```
install.packages('ggplot2')
```

Let's run through an example..

Data Visualisation

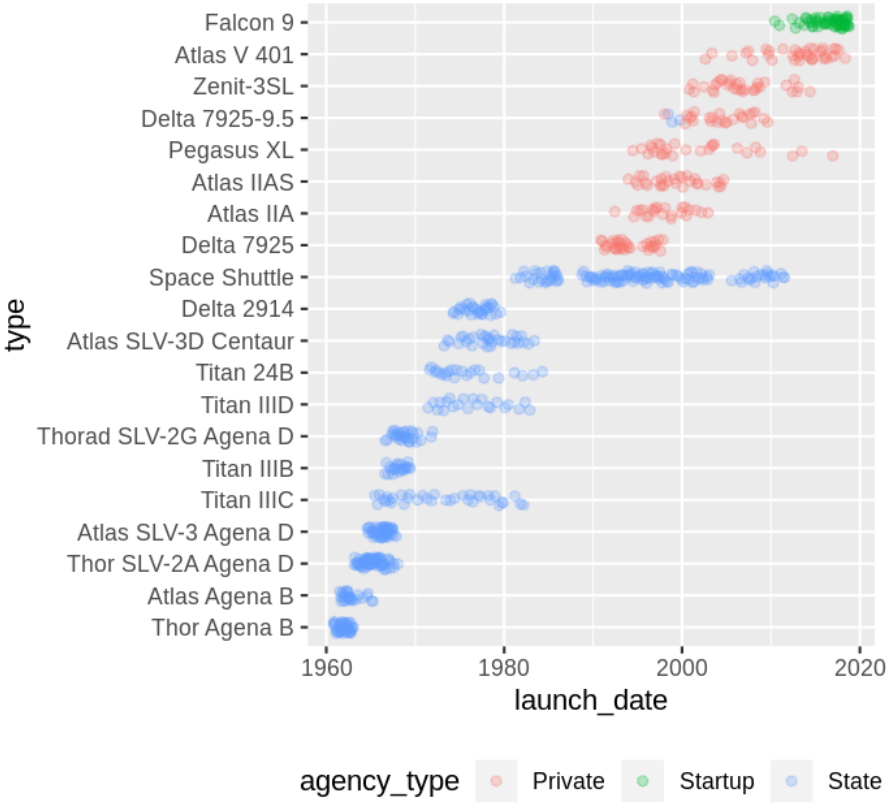
```
ggplot(data = launches_processed,
       aes(x = launch_date,
           y = type)) +
 geom_point()
```

Data Visualisation

```
ggplot(launches_processed,
       aes(x = launch_date,
           y = type)) +
  geom_jitter(alpha = .25,
              width = 0,
              height = .2)
```

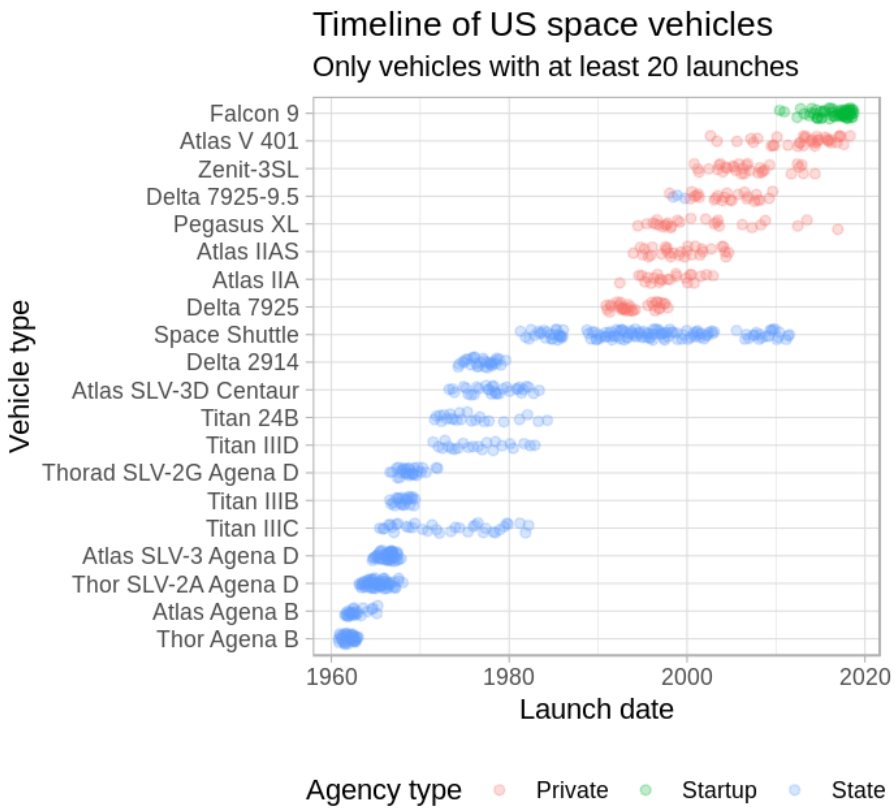
Data Visualisation

```
ggplot(data = launches_processed,
       aes(x = launch_date,
           y = type,
           color = agency_type)) +
  geom_jitter(alpha = .25,
              width = 0,
              height = .2) +
theme(legend.position = "bottom")
```



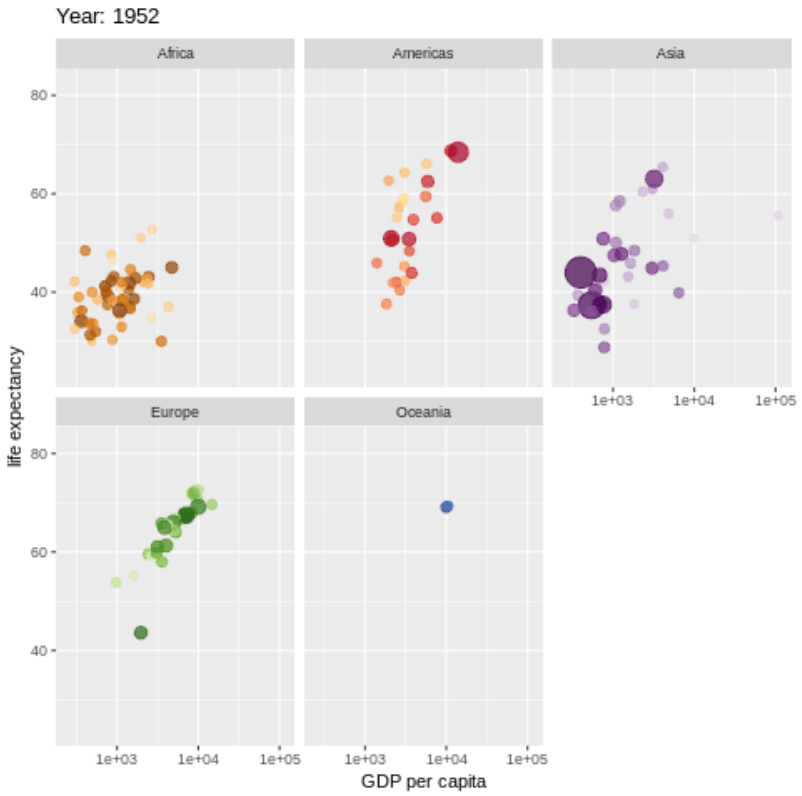
Data Visualisation

```
ggplot(launches_processed,
       aes(x = launch_date,
           y = type,
           color = agency_type)) +
 geom_jitter(alpha = .25,
             width = 0,
             height = .2) +
labs(title = "Timeline of US space vehicles",
     x = "Launch date",
     y = "Vehicle type",
     color = "Agency type",
     subtitle = "Only vehicles with at least 2
theme_light() +
theme(legend.position = "bottom")
```



Animated Visualisations

the `gganimate` package can make animated visualisations easy by extending the `ggplot2` API.



Modelling and Analysis

Traditionally a huge strength in R.

- Wide range of linear models, glm, decision trees, machine learning models.

```
fit <- lm(formula = Sepal.Width ~ Petal.Length + Petal.Width, data = iris)
```

Commonly used formula interface:

`Sepal.Width ~ Petal.Length + Petal.Width`

Modelling and Analysis

```r
summary(fit)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Petal.Length + Petal.Width, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.06198 -0.23389  0.01982  0.20580  1.13488
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.58705    0.09373  38.272  < 2e-16 ***
## Petal.Length -0.25714    0.06691  -3.843  0.00018 ***
## Petal.Width   0.36404    0.15496   2.349  0.02014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3893 on 147 degrees of freedom
## Multiple R-squared:  0.2131,    Adjusted R-squared:  0.2024
## F-statistic:  19.9 on 2 and 147 DF,  p-value: 2.238e-08
```

Modelling and Analysis

Good support for machine learning, model tuning, cross-validation.

The `caret` package (short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models.

```
install.packages('caret')
```

Modelling and Analysis

Logistic Regression

```
fit_glm <- caret::train(label ~ x + y,
                        data = training,
                        method = "glm")
```

CART Decision Tree

```
fit_rpart <- caret::train(label ~ x + y,
                          data = training,
                          method = "rpart")
```

Random Forest

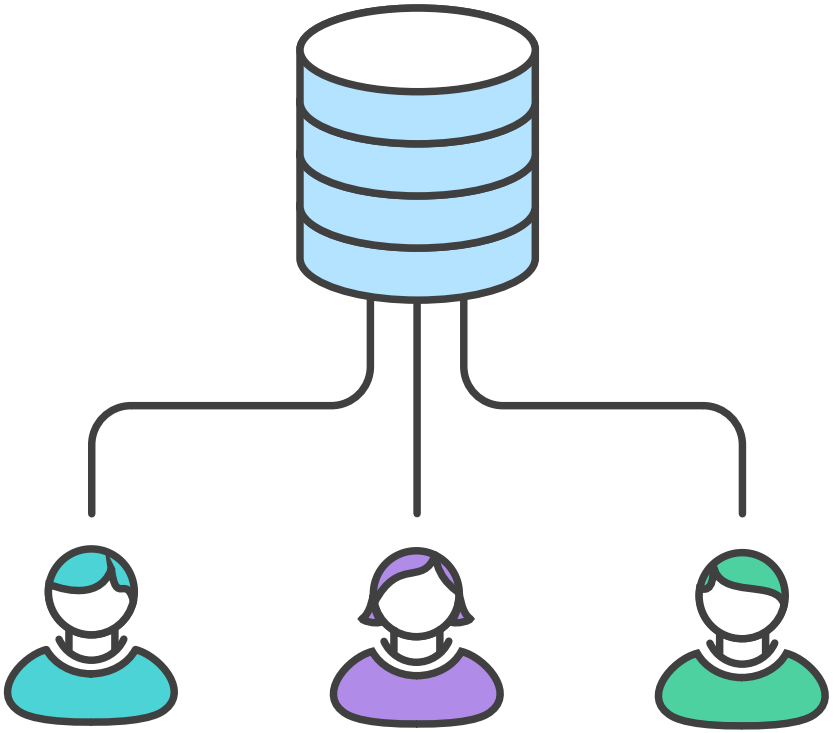
```
fit_rf <- caret::train(label ~ x + y,
                       data = training,
                       method = "rf")
```

Working with others

Version control and collaboration

Git / Github

- Powerful version control system used in software development
- Also well suited to managing research work
- Controls the 'source code' of your work with multiple contributors



Use cases for git in science

1. Lab notebook
2. Facilitating Collaboration
3. Backup and Fail-safe against data loss
4. Freedom to explore new ideas and methods
5. Mechanism to solicit feedback and reviews
6. Increase transparency and verifiability
7. Managing large data
8. Lowering barriers to reuse

Ram, K. (2013). Git can facilitate greater reproducibility and increased transparency in science. Source Code for Biology and Medicine, 8(1). https://doi.org/10.1186/1751-0473-8-7

Lots of great git resources:

Git for Scientists - Miles McBain

Atlassian Git Tutorials

Git: A powerful tool to facilitate greater reproducibility and transparency in science

Git and Github - R Packages

A Quick Introduction to Version Control with Git and GitHub

Getting your work out there

Papers, Talks, Posters, Blogs and more

R Markdown

> R Markdown provides an authoring framework for data science. You can use a single R Markdown file to both
>
> - save and execute code
> - generate high quality reports that can be shared with an audience

source: https://rmarkdown.rstudio.com/

R Markdown Gallery

https://rmarkdown.rstudio.com/gallery.html

Shiny

Shiny is a framework to build interactive web apps straight from your R code.

- Get your code into users' hands
- Easily deploy and host apps on the cloud
- Turn your research outputs into useful production ready tools

https://shiny.rstudio.com/

Shiny User Showcase

Shiny Demo

https://deanmarchiori.shinyapps.io/abtester/

A/B Testing Tool

This program will calculate the response rate for a test vs control (or A vs. B) experiment and conduct a statistical test to determine if there is a significant difference in response rates between the groups

Campaign Name

New Link

A Group

Name of the 'A' group

A Group

Number of subjects

1000

Number of responses

250

B Group

Name of the 'B' group

B Group

Number of subjects

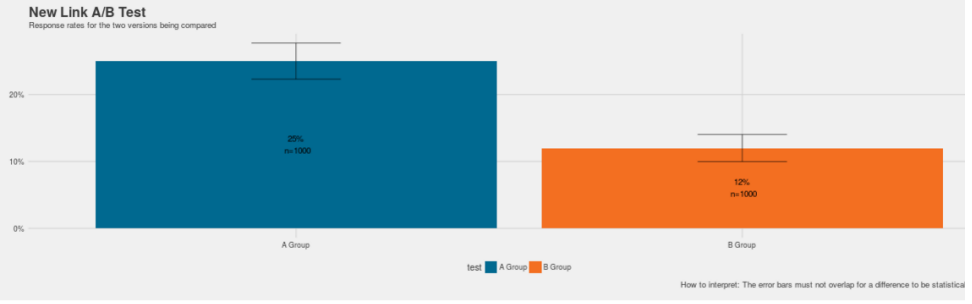
1000

Number of responses

120

Select Confidence Level

99% 95% 90%



Conclusion

The two groups are different at a statistically significant level. Based on the sample of responders in this campaign, the estimate for the true uplift is between 9.64% and 16.36% with a 95% confidence level

Response Table

	Subjects	Responses	Non-Responses	Response Rate (%)
A Group	1000	250	750	25%
B Group	1000	120	880	12%

R Packages

> In R, the fundamental unit of shareable code is the package. A package bundles together code, data, documentation, and tests, and is easy to share with others

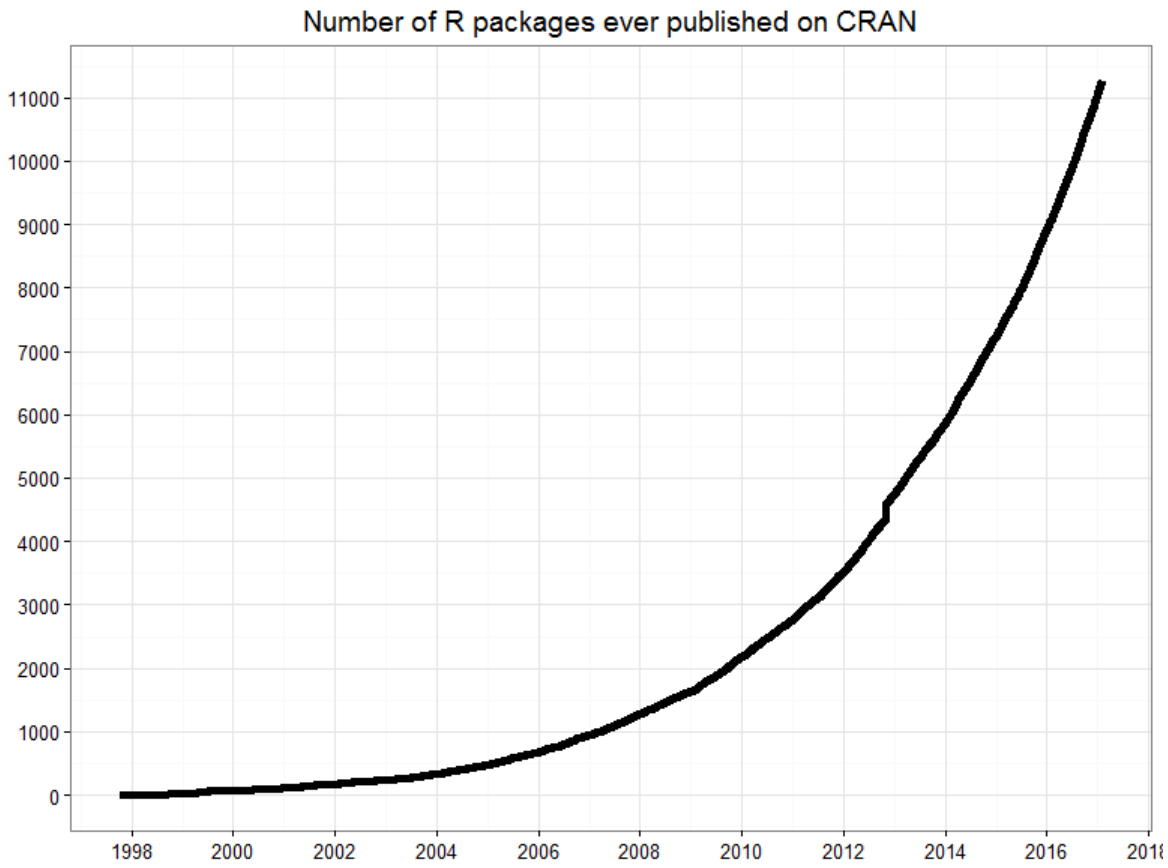
source: R Packages - Hadley Wickham

Why R Packages?

- Amplify your research work by turning it into open-source software
- Publish and licence your software to the community
- Ensure it can be run and maintained by others and not trapped on the 'C Drive'

ROpenSci has a great model for promoting open data and software for science.

R Packages



source: https://blog.revolutionanalytics.com/2017/01/cran-10000.html

References

Adam H Sparks, Mark Padgham, Hugh Parsonage and Keith Pembleton (2017). bomrang: Fetch Australian Government Bureau of Meteorology Weather Data The Journal of Open Source Software, 2(17). DOI: 10.21105/joss.00411

Adam H Sparks, Jonathan Carroll, Dean Marchiori, Mark Padgham, Hugh Parsonage and Keith Pembleton. (2018). bomrang: Australian government Bureau of Meteorology (BOM) data from R. R package version 0.4.0. https://CRAN.R-project.org/package=bomrang

Amelia McNamara, Eduardo Arino de la Rubia, Hao Zhu, Shannon Ellis and Michael Quinn (2019). skimr: Compact and Flexible Summaries of Data. R package version 1.0.6. https://github.com/ropenscilabs/skimr

Hadley Wickham (2017). tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

JJ Allaire and Yihui Xie and Jonathan McPherson and Javier Luraschi and Kevin Ushey and Aron Atkins and Hadley Wickham and Joe Cheng and Winston Chang and Richard Iannone (2018). rmarkdown: Dynamic Documents for R. R package version 1.11. URL https://rmarkdown.rstudio.com.

References (cont.)

Yihui Xie and J.J. Allaire and Garrett Grolemund (2018). R Markdown: The Definitive Guide. Chapman and Hall/CRC. ISBN 9781138359338. URL https://bookdown.org/yihui/rmarkdown.

Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan McPherson (2018). shiny: Web Application Framework for R. R package version 1.2.0. https://CRAN.R-project.org/package=shiny

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Thomas Lin Pedersen and David Robinson (2019). gganimate: A Grammar of Animated Graphics. R package version 1.0.3. https://CRAN.R-project.org/package=gganimate

Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan and Tyler Hunt. (2019). caret: Classification and Regression Training. R package version 6.0-84. https://CRAN.R-project.org/package=caret

Thanks!

Fire some questions at me