

(Reproducible) Data Visualisation with R and how to make interactive things with R

Charlotte Hadley

That title is too long

I'm going to talk about 4 things:

- R: What it is and the R user community
- [Static] Data visualisation with {ggplot2}
- Interactive data visualisation with R
- Reproducible research [and data visualisation with R]

R

Who has never used R before?

What do you get when you install R?

There are 3 things you get when you install R:

- The gubbins necessary to run R code
- A collection of R packages we call **base R** because *everyone* has them
- An application for running R code

On macOS that application is called
`R.app`

On Windows that application might
be called `Rgui.exe` or `R i386`.

| These are terrible ways to use R.

RStudio

RStudio is *the place to write R code*

RStudio is a traditional IDE designed for the R language.

It also provides built in tools for:

- Static data visualisation
- Interactive data visualisation with htmlwidgets
- Web application development with Shiny
- Reproducible research patterns with RStudio projects and RMarkdown.

We're going to start using RStudio project NOW and explain why later.

CRAN

CRAN is where R lives

You downloaded R from CRAN. It's also the package repository for R.

It contains 18,000+ packages.

They're all¹ installed in the same way

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

1 - If available for the current version of R

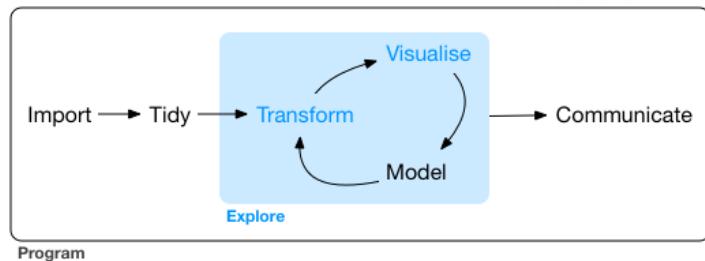
Packages you'll need

```
install.packages(c("fivethirtyeight", "fontawesome", "ggplot2",
  "here", "highcharter", "janitor", "leaflet", "plot
  "sf", "shiny", "tidyverse", "xaringan", "visNetwo
```

tidyverse

What's the tidyverse

The tidyverse is a collection of packages for doing data science¹



It's the backbone of the incredible
[FREE] [R for Data Science book](#).

And the [R for Data Science Online Learning Community](#).

1 - Data cleaning, wrangling, modelling and static data visualisation.

R Community

#rstats



Want to learn more about Twitter and R? t4rstats.com

Some data visualisations - please?

The `{ggplot2}` chapter

Charlotte Hadley

fivethirtyeight::bechdel

FiveThirtyEight.com does really good data journalism.

They use R for most of their data analysis and make their code/cleaned datasets available through an R package - {fivethirtyeight}.

We're going to look at the dataset from this article: [The Dollar-And-Cents Case Against Hollywood's Exclusion of Women](#)

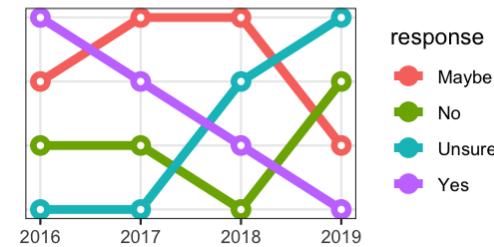
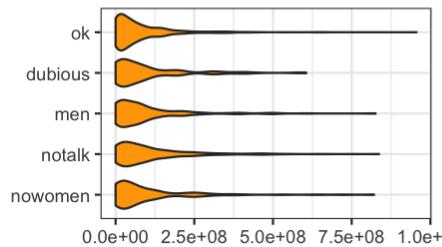
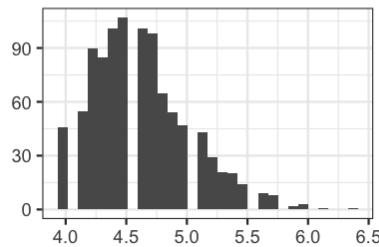
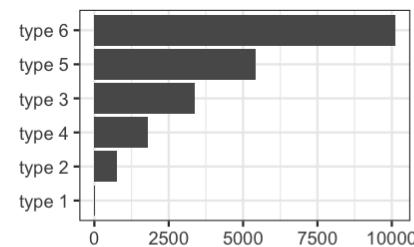
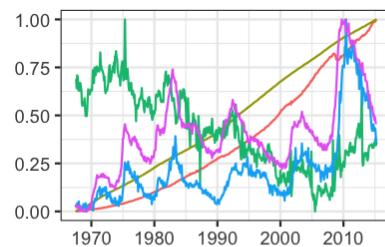
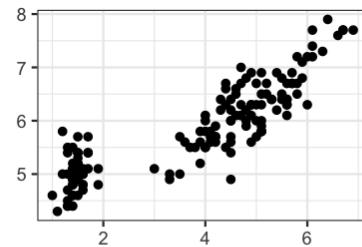
Alison Bechdel's 'The Rule' strip; Dykes to Watch Out For, 1985



ggplot2: A Grammar of Graphics

{ggplot2} is an incredibly powerful and flexible tool for building static dataviz.

We can build (almost)¹ **any** static chart we can conceive of.



[1] - Dual y-axis charts must be transformations of one another ([for good reasons](#))

ggplot2: Charts are built, not chosen

The most common workflow for dataviz tools is:

1. Choose the chart we want
2. Find the charting function for that chart
3. Specify the options/attributes/style of the chart

But in `{ggplot2}`, things are much different:

1. Choose the chart we want
2. Ensure the **data** contains the necessary structure for our chart
3. Build up a chart by mapping our data to the **aesthetics** of the chart
4. Customise the chart **scales** and **theme**.

Building blocks of a {ggplot2} chart

-  Aesthetics
-  Geoms
-  Scales
-  Guides
-  Theme

Building blocks of a {ggplot2} chart

-  **Aesthetics**: these create mappings between columns in datasets and the coordinate system of the chart.

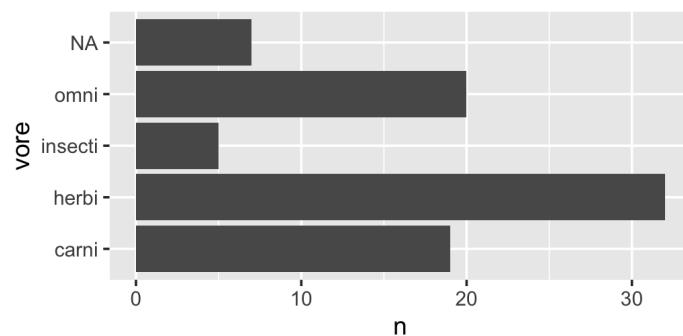
Building blocks of a {ggplot2} chart

-  Aesthetics
-  Geoms: these use the aesthetics to draw layers onto our charts

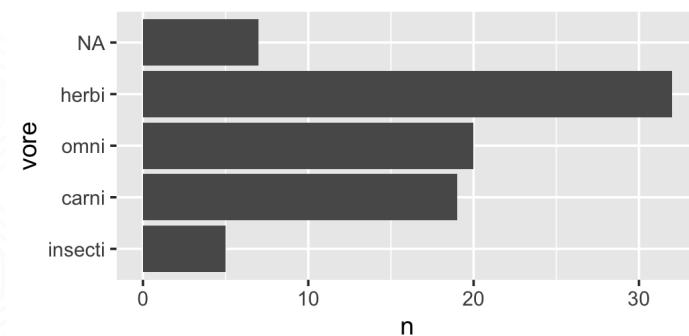
Geoms & categorical variables

In R we use `factors` to control the order in which geoms are drawn (and appear in legends)

- Default (alphabetical) ordering



- Reordered by another column



Building blocks of a {ggplot2} chart

-  Aesthetics
-  Geoms
-  Scales: these affect how the aesthetics appear in the chart

Building blocks of a {ggplot2} chart

-  Aesthetics
-  Geoms
-  Scales
-  Guides: these affect how the guides (or legends) are constructed from the aesthetics and scales.

Building blocks of a {ggplot2} chart

-  Aesthetics
-  Geoms
-  Scales
-  Guides
-  Theme: this affects the look and feel of the chart.

Why is {ggplot2} awesome?

Maps with {ggplot2}

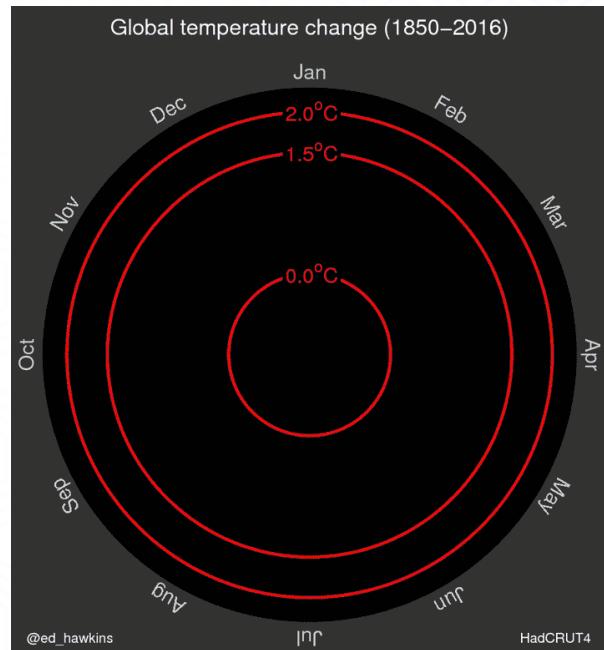
We can use {ggplot2} to make maps

```
ggplot() +  
  geom_sf(data = countries_sf,  
          aes(fill = lastcensus,  
              shape = "No data\available")) +  
  scale_fill_viridis_c()
```

Animated GIF with {ganimate}

The [{ganimate}](#) package allows us to create animated GIF with {ggplot2}.

GIFS?



{ggplot2} extensions

There are 100+ registered {ggplot2} extensions -
exts.ggplot2.tidyverse.org/gallery.

The interactive things chapter

Charlotte Hadley

htmlwidgets: JavaScript

As Dr Alfie Abdul-Rahman will explain later today - the interactive web is driven by JavaScript*.

Just like R, there are Javascript packages [or frameworks] that are designed for building data visualisations.

htmlwidgets gives R interactivity

The `{htmlwidgets}` package is a tool that allows R developers build R wrappers for JavaScript.

This means we R users can build interactive charts, maps and more with R code instead of needing to learn JavaScript.

htmlwidgets.org

The spiritual home of `htmlwidgets` is htmlwidgets.org and is maintained by the folks at RStudio.

- htmlwidgets.org introduces the basics of building your own `htmlwidget` packages
- htmlwidgets.org has a showcase of the 13 most high-profile `htmlwidget` packages (many of which have been developed by RStudio themselves)
- htmlwidgets.org has a gallery of many other `htmlwidget` packages.

htmlwidgets sanity check

Every single `htmlwidget` package is completely unique in the way that data and chart options are specified.

Why?

- Underlying JavaScript libraries/frameworks are often completely different to one another.
- Developers have a choice between an easy to use `htmlwidget` and a package that allows you to do *everything* you could do with the native JS library.
- Developers have a choice in how `tibble` columns are specified to the chart; `"columns_as_strings"`, `naked_columns`, `~formula_columns`.

**What shall we
make?**

Building web apps with `{shiny}`

Interactive Data Network

I started the idn.it.ox.ac.uk service to support researchers in building interactive data visualisations of Open Access datasets.

We needed a tool that:

- Was easy to learn (look what we've done with R already!)
- Allowed the web app to download the **canonical dataset** - the one with a DOI.
- Was affordable for a pilot project in IT Services.

{shiny} was the only tool that came anywhere near meeting these requirements.

Shiny

Shiny is a self-contained web framework for building web apps with R.

Which means all you need is RStudio and {shiny} installed to build apps.

Sharing shiny apps

There are 3 options for sharing shinys apps.

- The freemium shinyapps.io service
- Installing the Open Source Shiny Server product on your own servers
- Paying RStudio for RStudio Connect.

Reproducibility with R

Charlotte Hadley

... RStudio projects?

We're using RStudio projects because they improve the reproducibility (and transportability) of our code.

Let me demonstrate how by quickly creating a text file...

Production-level reproducibility

The [{renv}](#) package snapshots R packages into your RStudio project - ensuring your code will run into the future.

RMarkdown

Life without RMarkdown

1. Write your report in Word
2. Write your code in R
3. Copy and paste your charts into Word
4. Update different parts of the report/code
5. ... are the charts up to date?
... are the figures in text (eg 73%) up to date?

There's a disconnect between the data and report - which is ripe for reproduciblity issues for both the author and reader.

RMarkdown is incredible

RMarkdown is a technology that allows you to build reports, presentations and more that incorporate R code.

Here are just some of the things you can build:

- Interactive HTML reports (containing {htmlwidgets})
- Beautiful PDF reports [for printing]
- Fully fledged books - remember [R for Data Science](#)
- Word documents [because sometimes you have to make them]
- PowerPoint slides
- Data blogs and other websites

Parameterised RMarkdown docs

The thing that makes RMarkdown particularly powerful is the ability to programmatically generate **many** reports from datasets.

I worked on a consultancy project with **R for the Rest of Us** to build reports for all US states.

The PDF reports sucked in text from .docx files, Excel workbooks and the US Census API for each state.

[funderscommittee.org/2020census
statereports](https://funderscommittee.org/2020census/statereports)

