

# DATA VISUALISATION

## LAB WORKSHEET 8

Creator: Dr Mark Taylor

---

### What's Happening in this Document?

The main point of this worksheet is a quick introduction to interactives.

This is just to give you a sense of what's out there! I would also recommend getting your head round some of the several extensions to ggplot2 at <https://exts.ggplot2.tidyverse.org/gallery/> in case you ever want to create some really exciting data visualisations in the future!

### Getting started with plotly

Let's load the tidyverse...

```
library(tidyverse)
```

...and let's install and load plotly. It's also fairly likely that R will attempt to install several other packages in order to get plotly loaded:

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

*Note: The usual resulting graphs have not been included after given code - this is on purpose.*

**plotly** is a tool for generating interactive graphics that isn't limited to ggplot2, and more generally isn't limited to R. What we're doing here is working within a ggplot2 framework to generate plotly objects: using R to generate interactives that can be dropped into different settings.

(In practice, the easiest thing is to integrate these interactives into .html files that you've generated using RMarkdown.)

## Loading some data

Let's use some World Bank data! Let's draw a scatterplot of accountability and stability of governments across the world.

Our World Bank dataset contains data on Voice and Accountability, which captures perceptions of the extent to which a country's citizens are able to participate in selecting their government, as well as freedom of expression, freedom of association, and a free media.

```
accountability_and_voice <-  
  read_csv("https://bit.ly/worldbank_av_data")  
  
accountability_and_voice_recent <-  
  accountability_and_voice %>%  
  filter(year == 2017)
```

We've done two things here: first, we loaded the data in from the web (because it's quick); second, filtered it so we're just looking at 2017.

1. Let's now draw a simple scatterplot.

```
ggplot(data = accountability_and_voice_recent) +  
  aes(x = accountability,  
      y = stability) +  
  geom_point()
```

Looks OK, save for the fact it's a bit boring (which we can fix through the standard techniques we've seen so far). How can we make it interactive?

# Introducing ggplotly

2. The `ggplotly` command turns ggplot objects into plotly objects. Let's start with the simplest case, in which we'll just wrap the earlier graph in ggplotly.

```
ggplotly(  
  ggplot(data = accountability_and_voice_recent) +  
  aes(x = accountability,  
      y = stability) +  
  geom_point())
```

While this might look exactly the same as before, if you hover the mouse over different points in the **Viewer** pane, you'll see it's now identifying the exact values of our points' x and y values: accountability, and stability.

3. However, the current presentation is a bit uninspiring: which countries are we looking at? We can tweak this by adding a `text` aesthetic. This is a bit of a bodged approach (I'm sure there's a more efficient way to do it): what we need is to combine the values of different variables for each observation with some bits of text.

```
ggplotly(  
  ggplot(data = accountability_and_voice_recent) +  
    aes(x = accountability,  
        y = stability,  
        text = paste('Country: ', country,  
                     'Accountability: ', accountability,  
                     'Stability: ', stability)) +  
  geom_point())
```

This looks OK, but there's a couple of issues. The first is that our old text hasn't disappeared, introducing redundancy. The second is that our text is all along a single line, which probably isn't what we want.

4. Let's address both of these problems here.

```
ggplotly(  
  ggplot(data = accountability_and_voice_recent) +  
    aes(x = accountability,  
        y = stability,  
        text = paste('Country: ', country,  
                     '<br> Accountability: ', accountability,  
                     '<br> Stability: ', stability)) +  
    geom_point(),  
    tooltip = "text")
```

A couple of changes here.

The first is that our quoted segments in the `text = paste()` bit of `aes()` now include

.  
means **line break**, so the instructive text that appears when we hover over a point now includes these line breaks.

The second is that we're no longer simply using `ggplotly()` to wrap the `ggplot` command: we're now including some code at the end, where we're specifying what the **tooltip** (the thing that appears when we hover over a point) should consist of. In this instance, we want it to consist of the text that we specified earlier, and nothing else.

5. One last thing to do to this graph before we move on. At the moment, the tooltip's giving us absolutely loads of decimal places, which can make it hard to interpret what's going on. We probably don't need more than two. And we can address this using the **round** command, like so:

```
ggplotly(  
  ggplot(data = accountability_and_voice_recent) +  
    aes(x = accountability,  
        y = stability,  
        text = paste('Country: ', country,  
                     '<br> Accountability: ',  
                     round(accountability,  
                             digits = 2),  
                     '<br> Stability: ',  
                     round(stability,  
                             digits = 2))) +  
    geom_point(),  
  tooltip = "text")
```

Here, we've specified which variables we want to round, and how many digits we want.

# More interaction, more animation

6. So far, we've just been looking at how the relationship between variables. But how might we look at how it's changed over time? Instead of looking at the version of the data that's just from 2017, let's look at the data over time.

```
ggplotly(  
  ggplot(data = accountability_and_voice) +  
    aes(x = accountability,  
        y = stability,  
        frame = year,  
        ids = country,  
        text = paste('Country: ', country,  
                     '<br> Accountability: ', round(accountability, digits = 2),  
                     '<br> Stability: ', round(stability, digits = 2)))+  
    geom_point() +  
    labs(x = "Accountability",  
         y = "Stability",  
         title = "More accountable countries are more stable"),  
    tooltip = "text")
```

A couple of changes here: - we've used the version of the data that covers the entire period - we've added a couple of extra aesthetic mappings: – **frame** (which determines which variable shows which observations are shown when), and – **ids** (which determines how points are connected between frames) - we've tidied up a bit, with titles and axis labels

In applying **frame** and **ids**, we've now got a **Play** button and a slider, so we can both see how this relationship has changed over time, and see what it was like at particular points.

## Bonus Exercise

There's loads more you can do with plotly. If you're interested in this, spend some time Googling plotly combined with ggplot2 and try a new function that we have not covered in this worksheet.

And, indeed, spend some time with exploring other ggplot2 extensions and try some out if you like: <https://exts.ggplot2.tidyverse.org/gallery/>