Advancing in R

Module 2: Data manipulation

Supplementary exercises

Plotting time series data

The data manipulation supplementary exercises included a question about reshaping time series data. If you haven't done this yet, now's your chance! Once you have reshaped the data, try plotting:

- A histogram of the response variable, with binwidth of 1.5
- Try faceting your histogram by 'treatment', so you can see how the treatment modifier variable affects the distribution of observations
    - In this case, I don't think faceting is actually that useful for investigating the extent to which histograms overlap – I like to use 'density plots' on a single panel for this, using a different fill colour for each categorical variable, and a low alpha to introduce some transparency:

```
OBK_tidy %>%
   ggplot(aes(x = Measurement, fill = treatment)) +
   geom_density(alpha = 0.4)
```

- A scatterplot of response against time
- Add 'geom_line' to your scatterplot to connect individual responses
    - HINT: check the documentation for the 'group' aesthetic in 'geom_line'…
- Add a colour for treatment group
- Which other variables do you want to include? How would you visualise these clearly?
- If faceting in a grid by – for example – gender (horizontally) and period (vertically), you may want to change some of the properties of the facets:
    - The rows are automatically set out in table fashion, with first value at the top. To reverse this, set 'as.table = FALSE'.
    - Facet labels display only the **value** from the **key:value** pair by default. You can change this in several ways…
        - Keys and values for both: labeller = label_both
        - Keys and values for one (e.g. gender): labeller = labeller(gender = label_both)
        - Create your own custom labeller… (see documentation)

Piping manipulations into plots

The ggplot2 package also enables use of the pipe command that you learned about in Module 1. You can chain operations and send the resulting data frame to your ggplot specification using 'data = .'.

Save your current project, and reload the data manipulation project you created in the last module, loading up the ggplot2 and dplyr packages. Create a chain of commands that selects just location, year, month, and maximum temperature; filter for data from years 2000-2010. Instead of ending this here, make a scatterplot of maximum temperature against year. Try adding a smoother to this as well.

Now add in Year as a colour aesthetic in your main ggplot call: is the result what you would have expected? If not, why not? Have a look at the structure of the weather data again. This time, when you add Year as a colour aesthetic, write it as 'colour = factor(Year)'. Do you see why this is different?

Ggplot2 is smart, so it uses the structure of your data to specify things like colours, smoothers, etc. When 'Year' is specified as a continuous variable, the colour range is also continuous, and there is a single smoother for the whole data set (because there isn't an obvious way to slice it up). When you specify that each year is a distinct factor level, making Year a categorical variable, ggplot2 gives distinct (and non-ordered) colours, and a separate smoother for each level.

Use the links contained in 'further reading' (in the main practical handout) to add more to your plot, for example:

- Change the theme;
- Use 'scale_x_continuous' to add labels with the name of each month to the x axis;
- Make a new colour palette for 'year' using 'scale_colour_brewer';
- …and anything else you can think of!