Overview

This worksheet provides a set of exercises for you to practice the final exam. There is no submission, and no grade associated with them. There are 10 parts to this worksheet. Download the accompanying worksheet data W6-8.zip from Canvas.

Part 1 – Scatter plots and regression

1. Using the `mpg` dataset, create a scatter point graph of displacement and highway mileage. Set the color aesthetic for cylinders.
2. The graph in #1 has mapped cylinders to a continuous scale. This is not what you wanted, as there are only 5 types of cylinder vehicles: 4, 5, 6, and 8. Create a new version of the graph which ensures cyl is mapped to a discrete/categorical scale. That is, the points should be coloured by the 4 unique categories in cyl, with the appropriate legend.
3. Overlay linear regression fits to you graph; there should be a separate fit for each cylinder. Note, there are too few 5-cylinder vehicles for the fit to display.
4. The default color of regression confidence intervals is grey. These shaded intervals are polygons. Knowing this, map the colour shading of these to cyl, so the points, regression lines, and the shaded confidence intervals all share the same colour.
5. Let's prepare the graphic for presentation. Drop the redundant legend. Set the transparency on your confidence intervals to 0.1. Rename the x-y axes to "Engine displacement (Litres)", "Highway mileage (miles/gallons)", and legend title to "Number of engine cylinders". Add a title. Your final graphic should look like Figure 1 below.

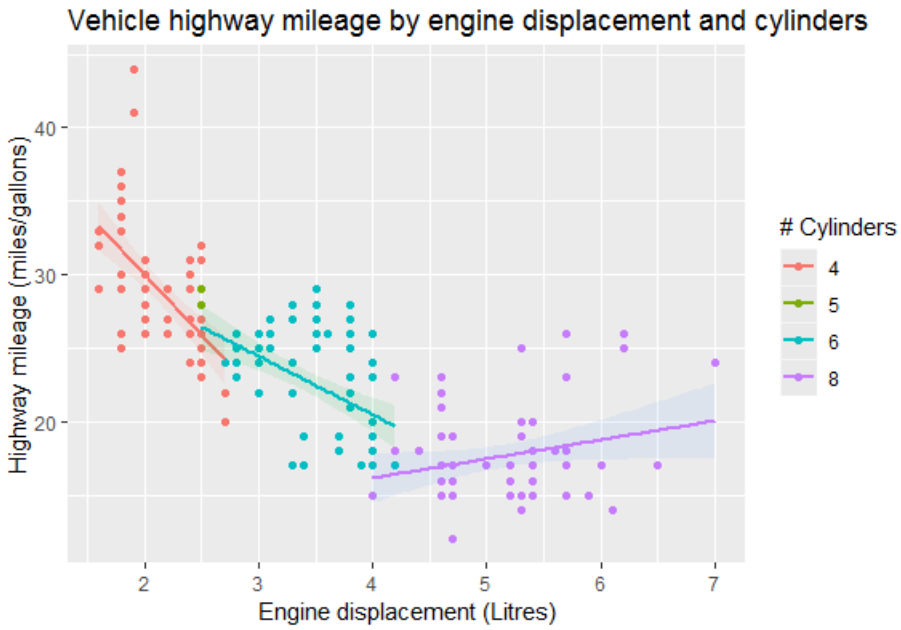

Figure 1. Car highway mileage (mpg) as a function of engine displace (L)

Part 2 – Playing with colour

6. ggplot uses a representation of colour known as Hue, Chroma, Luminance. Play around with the HCL color picker to get a better idea of these properties - https://blocks.roadtolarissa.com/mbostock/3e115519a1b495e0bd95
7. Load W6-8.RData data file from Canvas.
8. Using `d_p2`, create a bar chart of cond(x) and yval(y), fill(cond).
9. When mapping data to colour, ggplot picks colours that are evenly spaced along the HCL color wheel, starting at Hue = 15 with a range of [0, 360], fixed Chroma = 100, and fixed luminance = 65. Explore the function `scale_fill_hue()`.
10. Using the graph from #7, create separate bar graphs for each set of the following:
    a. Chroma = 100, luminance = 65
    b. Chroma = 45, luminance = 65
    c. Chroma = 100, luminance = 30
    d. Chroma = 20, luminance = 100
    e. Chroma = 100, luminance = 100
    f. Hue range [0, 360], chroma 100, luminance = 65
    g. Hue range [0, 360], chroma 100, luminance = 65, hue starting = 15. Look familiar?
    h. Chroma 100, luminance = 65, hue starting = 15; compare that with hue starting = 0. What's going on here?
    i. Chroma 100, luminance = 65, direction = -1.
    j. Hue range [275, 340].
    k. Finally, hue range [275, 340], hue start = -275. Compare that with hue range [0, 65], hue start = 0. Were you right in your answer to #8?

Part 3 – Proportional data

The data in `d_p3` contains response data to the question "How much do you completing surveys?".

11. Using `d_p3`, convert the Response column to factor, with the labels Never, Rarely, Sometimes, Often, Always, for levels 1 through 5, respectively. Convert the Gender column to factor, with labels women and men, for levels 1 and 2, respectively.
12. Create a pie chart, visualizing Summary (y), and facetting by Gender. Change the color scheme to something befitting a five-category pie chart (e.g., colorBrewer "Blues"). Drop the x & y axis names, and set the break points (ticks) as a sequence from 0.1 to 1, in increments of 0.1. Use labels from 10 to 100% (try using scales::). Set the color to azure2. Reverse the legend ordering.
13. Modify the theme, placing the legend at the bottom, and set the legend text size to 10.
14. Load the 'ggthemes' package, and find a favorite theme. Try stata and economist_white. Your final graphic should look like Figure 2 below.

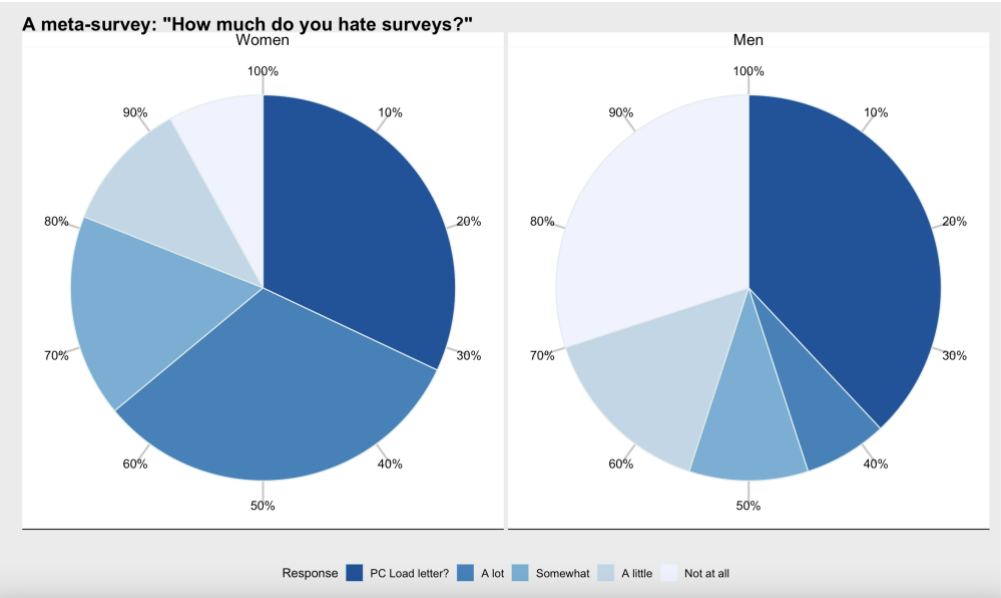

Figure 2. A survey on surveys.

Part 4 – Correlation plots

15. Load the GGally package. Visualise a correlation matrix of the `mtcars` dataset.
16. Style your correlation plot with the following features: add labels, 9 numeric breaks, colour palette "RdBu", label size 3.5, have the label alpha transparency vary with the correlation value, legend size 12, give your plot a title.
17. Add a name to your legend with the following:
    `name = expression(Correlation~italic(rho))`. Your final graphic should look like Figure 3.

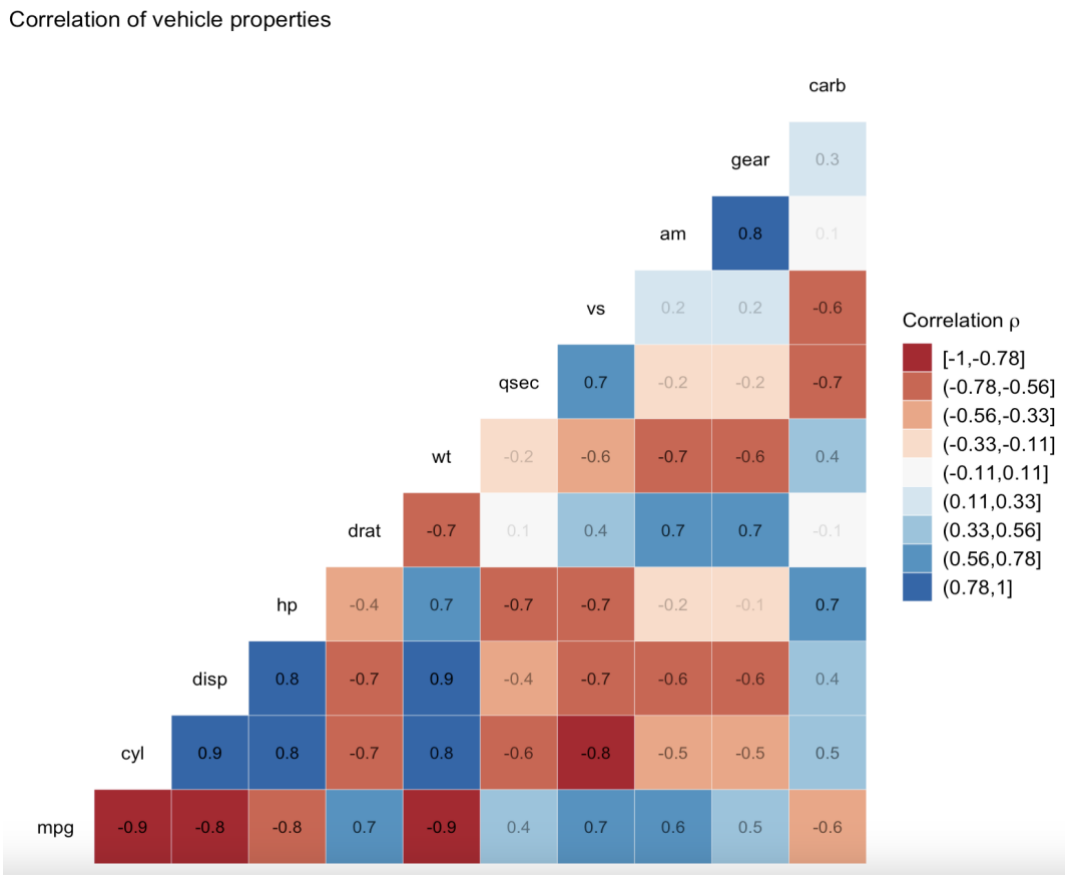

Figure 3. Correlation of vehicle properties in the mtcars dataset.

Part 5 – Polygon maps

18. Using the maps library, use `map_data()` to extract a `county` map, for the `region` “new york". Rename the long column to lon, and subregion to region. Convert it to a tibble. Store the result in `d_ny`.
19. Using `d_ny`, create a polygon map of New York state, filling by region.
20. Apply the following features: colour of region borders to “grey20”, line width 0.2, disable the legend for region, fill with the viridis colour scale (plasma palette), use coord_quickmap, and add a title. Your final graphic should look like Figure 4.

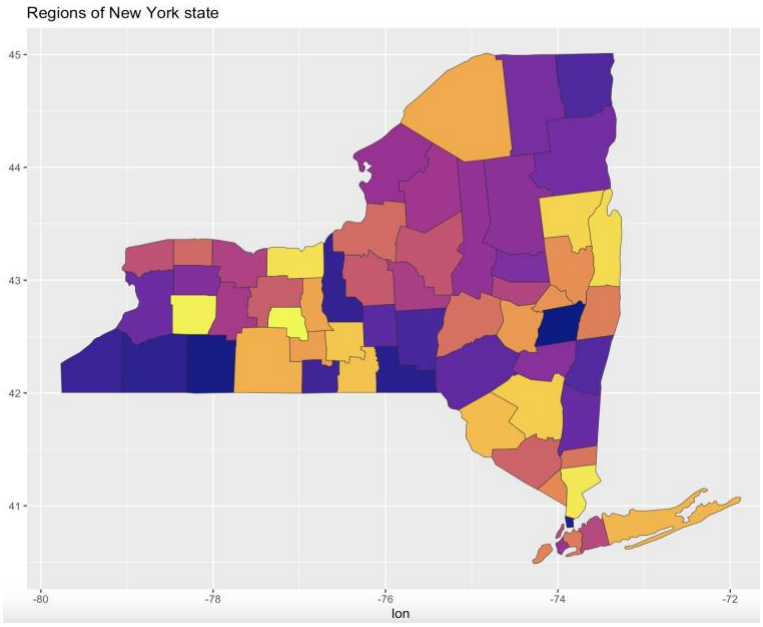

Figure 4. Map of New York state, shaded by region

Part 6 – sf maps

For this part of the worksheet, we will use the `ozmaps` package. This provides a map of Australia in the simple features (sf) format. Install and load the `ozmaps` package. Load the `sf` package.

Australia is an island nation, composed of six states and two territories (note, there are several other minor territories which we won't examine here).

21. Using `ozmap_data()`, load map data for 'states', filtering out 'Other Territories'. Store the result in the variable `oz_states`.
22. Using sf, visualise the map, using CRS 7844.
23. Colour region borders with grey20, line width 0.2, fill regions by 'NAME' using the Viridis colour scale, disable the legend and axis names, add region labels with geom_sf_label mapping label to 'NAME', and add a title. Your final graphic should look like Figure 5.

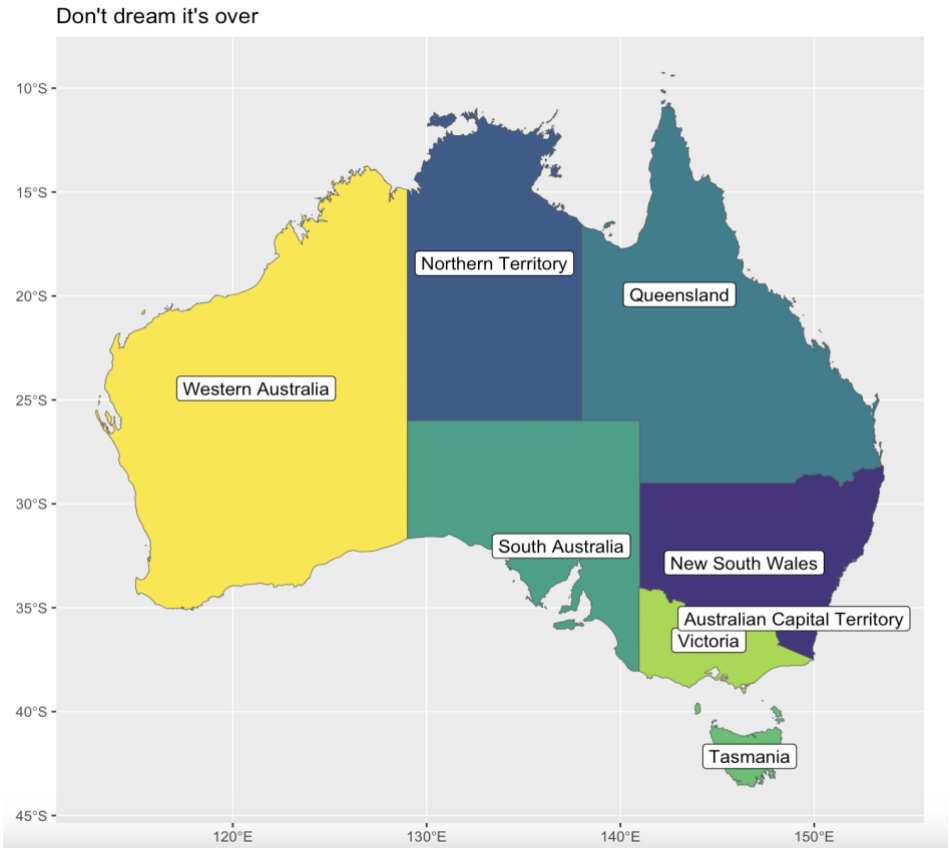

Figure 5. How's the serenity?