1.3 — Data Visualization with ggplot2 — R Practice

ECON 480 — Fall 2021

Thursday, SEptember 2, 2021

Getting Set Up

Before we begin, start a new file with `File` → `New File` → `R Script`. As you work through this sheet in the console in `R`, also add (copy/paste) your commands that work into this new file. At the end, save it, and run to execute all of your commands at once.

First things first, load `tidyverse`.

```
library("tidyverse")
```

Warm Up to `dplyr` With `gapminder`

1. Load `gapminder`. First, `select()` the variables `year`, `lifeExp`, `country`.
2. `select()` all variables *except* `pop`.
3. `rename()` `continent` to `cont`.
4. `arrange()` by year.
5. `arrange()` by year, but in descending order.
6. `arrange()` by year, then by life expectancy.
7. `filter()` observations with `pop` greater than 1 billion.
8. Of those, look only at `India`.
9. Try out the pipe (`%>%`) if you haven't already, by chaining commands: `select()` your data to look only at `year`, `gdpPercap`, and `country` in the year `1997`, for countries that have a `gdpPercap` greater than 20,000, and `arrange()` them alphabetically.
10. `mutate()` a new variable called `GDP` that is equal to `gdpPercap * pop`.
11. `mutate()` a new population variable that is the `pop` in millions.
12. `summarize()` to get the average GDP per capita.
13. Get the number of observations, average, minimum, maximum, and standard deviation for GDP per capita.
14. Get the average GDP per capita over time. Hint, first `group_by()` year.
15. Get the average GDP per capita by continent.
16. Get the average GDP per capita by year and by continent. [Hint: do `year` first, if you do `continent` first, there are no years to group by!] Then save this as another `tibble` called `gdp`. Create a `ggplot` of a `line` graph of average continent GDP over time using the `gdp` data.

17. Try it again all in one command with the pipe `%>%`. Instead of saving the data as `gdp`, pipe it right into `ggplot`! [Hint: You can use `.` as a placeholder.]

Example: the Economics of College Majors

Now let's step it up to work with some data "in the wild" to answer some research questions. This will have you combine your `dplyr` skills and add some new things such as importing with `readr`.

Let's look at fivethirtyeight's article " The Economic Guide To Picking A College Major ". fivethirtyeight is great about making the data behind their articles public, we can download all of their data here. Search for `college majors` and click download (the blue arrow button). [This will download a `.zip` file that contains many spreadsheets. Unzip it with a program that unzips files (such as WinZip, 7-zip, the Unarchiver, etc).] We will look at the `recent-grads.csv` file.

The description in the `readme` file for the data is as follows:

Variable	Description
`Rank`	Rank by median earnings
`Major_code`	Major code, FO1DP in ACS PUMS
`Major`	Major description
`Major_category`	Category of major from Carnevale et al
`Total`	Total number of people with major
`Sample_size`	Sample size (unweighted) of full-time, year-round ONLY (used for earnings)
`Men`	Male graduates
`Women`	Female graduates
`ShareWomen`	Women as share of total
`Employed`	Number employed (ESR == 1 or 2)
`Full_time`	Employed 35 hours or more
`Part_time`	Employed less than 35 hours
`Full_time_year_round`	Employed at least 50 weeks (WKW == 1) and at least 35 hours (WKHP >= 35)
`Unemployed`	Number unemployed (ESR == 3)
`Unemployment_rate`	Unemployed / (Unemployed + Employed)
`Median`	Median earnings of full-time, year-round workers
`P25th`	25th percentile of earnigns
`P75th`	75th percentile of earnings
`College_jobs`	Number with job requiring a college degree
`Non_college_jobs`	Number with job not requiring a college degree
`Low_wage_jobs`	Number in low-wage service jobs

18. Import the data with `read_csv()` and assign it to an object called `majors`. [One way to avoid error messages is to move (on your computer) `recent_grads.csv` to the same folder as R's working directory, which again you can check with `getwd()`.] The first argument of this command is the name of the original file, in quotes. [If the file is in a different folder, the argument is the full path in quotes.]
19. Look at the data with `glimpse()`. This is a suped-up version of `str()` in `tidyverse`.
20. What are all of the *unique* values of `Major`? How many are there?
21. Which major has the *lowest* unemployment rate?
22. What are the top 3 majors that have the highest percentage of women?
23. Make a boxplot of `Median` wage by `Major_Category`. [You won't be able to read the labels easily, so add `theme(axis.text.x=element_text(angle=45, hjust=1)` to angle `x`-axis labels (and move them down by 1)]

24. Which major category is the least popular in this sample? [Hint: use `group_by` first.]
25. Is there a systematic difference in median earnings between STEM majors and non-STEM majors? First define:

```
stem_categories <- c("Biology & Life Science",
                     "Computers & Mathematics",
                     "Engineering",
                     "Physical Sciences")
```

Next, make a variable called `stem`, for whether or not a `Major_category` is `"stem"` or `"not_stem"`. Then `summarize()` `median` for stem and not stem groups.

[Hint: try out the `ifelse()` function which has three inputs: condition(s) for a variable(s), what to do if `TRUE` (the if), and what to if `FALSE` (the else), i.e.

```
stem = ifelse(my_conditions,
       yes = do_this_if_TRUE,
       no = do_this_if_FALSE)
```

You'll of course need to change the `do_this` into something!]