1.3 — Data Visualization with ggplot2 — R Practice

ECON 480 — Fall 2020

Tuesday, August 25, 2020

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.

"Our Plot" from Class

- `R Script`

Download and run in R Studio on your computer (or open the file in our R Studio cloud project and run it there) to see our plot from class.

Exploring the Data

1. We will look at GDP per Capita and Life Expectancy using some data from the gapminder project. There is a handy package called `gapminder` that uses a small snippet of this data for exploratory analysis. Install and load the package `gapminder`. Type `?gapminder` and hit enter to see a description of the data.

2. Let's get a quick look at `gapminder` to see what we're dealing with.

a. Get the `str`ucture of the `gapminder` data.
b. What variables are there?
c. Look at the `head` of the dataset to get an idea of what the data looks like.
d. Get `summary` statistics of all variables.

Simple Plots in Base R

3. Let's make sure you can do some basic plots before we get into the gg. Use base R's `hist()` function to plot a *histogram* of `gdpPercap`.

4. Use base R's `boxplot()` function to plot a *boxplot* of `gdpPercap`.

5. Now make it a *boxplot* by `continent`.[1]

6. Now make a *scatterplot* of `gdpPercap` on the x-axis and `LifeExp` on the y-axis.

Plots with ggplot2

7. Load the package `ggplot2` (you should have installed it previously. If not, install first with `install.packages("ggplot2")`).

8. Let's first make a `bar` graph to see how many countries are in each continent. The only `aesthetic` you need is to map `continent` to `x`. Bar graphs are great for representing categories, but not quantitative data.

9. For quantitative data, we want a `histogram` to visualize the distribution of a variable. Make a `histogram` of `gdpPercap`. Your only `aesthetic` here is to map `gdpPercap` to `x`.

10. Now let's try adding some color, specifically, add an `aesthetic` that maps `continent` to `fill`.[2]

11. Instead of a `histogram`, change the `geom` to make it a `density` graph. To avoid overplotting, add `alpha=0.4` to the `geom` argument (alpha changes the *transparency* of a `fill`).

12. Redo your plot from 11 for `lifeExp` instead of `gdpPercap`.

13. Now let's try a scatterplot for `lifeExp` (as `y`) on `gdpPercap` (as `x`). You'll need both for `aesthetics`. The `geom` here is `geom_point()`.

14. Add some color by mapping `continent` to `color` in your `aesthetics`.

15. Now let's try adding a regression line with `geom_smooth()`. Add this layer on top of your `geom_point()` layer.

16. Did you notice that you got multiple regression lines (colored by continent)? That's because we set a `global aesthetic` of mapping `continent` to `color`. If we want just *one* regression line, we need to instead move the `color = continent` inside the `aes` of `geom_point`. This will only map `continent` to `color` for points, not for anything else.

17. Now add an `aesthetic` to your `points` to map `pop` to `size`.

18. Change the color of the regression line to `"black"`. Try first by putting this inside an `aes()` in your `geom_smooth`, and try a second time by just putting it inside `geom_smooth` without an `aes()`. What's the difference, and why?

19. Another way to separate out continents is with `faceting`. Add `+facet_wrap(~continent)` to create subplots by `continent`.

20. Remove the `facet` layer. The `scale` is quite annoying for the `x`-axis, a lot of points are clustered on the lower level. Let's try changing the scale by adding a layer: `+scale_x_log10()`.

21. Now let's fix the labels by adding `+labs()`. Inside `labs`, make proper axes titles for `x`, `y`, and a `title` to the plot. If you want to change the name of the legends (continent color), add