Lab 2 - Math 58 / 58b: Introduction to ggplot

your name here

due Feb 4, 2020

Lab Goals

The goals for today are to learn how to make plots using `ggplot`.

- `ggplot` structure of **layering** the pieces of a plot on top of one another using the `+` command
- `geom_point`, `geom_histogram`, `geom_boxplot`, `geom_line`

(same) Advice for turning in the assignment

- knit early and often. In fact, go ahead and knit your .Rmd file to a .pdf right now. Maybe set a timer so that you knit every 5 minutes. Do **not** wait until you are done with the assignment to knit.
- Save the .Rmd file somewhere you can find it. Don't keep everything in your downloads folder. Maybe make a folder called `Stats HW` or something.
- You should only turn in the last four questions at the very bottom of this assignment. However, you will need some of the commands above, otherwise the commands at the bottom won't run!
- Remove (delete) the vast majority of the lab (maybe save it somewhere else?) so that you are not turning in a 10 page assignment. Better for the trees and easier for the graders to grade.

Getting started

Work through the chapter at:

https://r4ds.had.co.nz/data-visualisation.html (Read through / work through **only sections 3.1 through 3.4**).

Some things to notice:

- when layering graph pieces, use `+`. (When layering data wrangling, use `%>%`.)
- `geom_XXX` will put the `XXX`-type-of-plot onto the graph.
- `aes` is the function which takes the **data columns** and puts them onto the graph. `aes` is used only with data columns and you *always* need it if you are working with data variables.
- A full set of types of plots is given here: https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf (and in many other places online).

Load packages

In this lab we will explore the data using `ggplot` which is included in the `tidyverse` package. The data (called `gapminder`) can be found in the package `dslabs`.

Let's load the packages.

```
library(tidyverse)  # ggplot lives in the tidyvers
library(dslabs)  # dataset for the lab
```

The data

Gapminder is a creation of Hans Rosling and his team to collect and visualize country level information over time. The `gapminder` R package and dataset provide a very few variables for us to work with.

```
names(gapminder)
```

```
## [1] "country"          "year"             "infant_mortality"
## [4] "life_expectancy"  "fertility"        "population"
## [7] "gdp"              "continent"        "region"
```

```
glimpse(gapminder)
```

```
## Observations: 10,545
## Variables: 9
## $ country          <fct> Albania, Algeria, Angola, Antigua and Barbuda...
## $ year             <int> 1960, 1960, 1960, 1960, 1960, 1960, 1960, 196...
## $ infant_mortality <dbl> 115.40, 148.20, 208.00, NA, 59.87, NA, NA, 20...
## $ life_expectancy  <dbl> 62.87, 47.50, 35.98, 62.97, 65.39, 66.86, 65....
## $ fertility        <dbl> 6.19, 7.65, 7.32, 4.43, 3.11, 4.55, 4.82, 3.4...
## $ population       <dbl> 1636054, 11124892, 5270844, 54681, 20619075, ...
## $ gdp              <dbl> NA, 13828152297, NA, NA, 108322326649, NA, NA...
## $ continent        <fct> Europe, Africa, Africa, Americas, Americas, A...
## $ region           <fct> Southern Europe, Northern Africa, Middle Afri...
```

To Turn In

1. Create a scatterplot with life expectancy on the y-axis and fertility rate on the x-axis. Color the points based on their continent.

Feel free to change the x and y axis labels to be more descriptive.

```
___ %>%
  ggplot(aes( x = ___, y = ___, color = ___) ) +
  ___()
```

2. Re-do the previous plot by first filtering the dataset to include only those observations from 1962. Make one more scatterplot for 2003.

Name one difference and one similarity in the pair of plots from 1962 and 2003.

```
gapminder %>%
  ___(___ == 1962) %>%
  ggplot(aes( x = ___, y = ___, color = ___) ) +
  ___()
```

3. Create a boxplot of `gdp` (on the y-axis) broken down by `continent` (on the x-axis).

```
___ %>%
  ggplot(aes(x = ___, y = ___)) +
  ___()
```

4. Re-do the previous plot, but change the y-axis to be on a log scale (see: `?scale_y_log10`).

Solution

```
___ %>%
  ggplot(aes(x = ___, y = ___)) +
  ___() +
  ___()
```