DataViz for SocScientists Notes

Caitlin S. Ducate

5/21/2020

```r
# Setup
myPackages <- c("tidyverse", "broom", "coefplot", "cowplot",
                "gapminder", "GGally", "ggrepel", "ggridges", "gridExtra",
                "here", "interplot", "margins", "maps", "mapproj",
                "mapdata", "MASS", "quantreg", "rlang", "scales",
                "survey", "srvyr", "viridis", "viridisLite", "devtools")
install.packages(myPackages)
devtools::install_github("kjhealy/socviz")
```

Chapter 1: Look At Data

Why look at data?

- Because numbers can be misleading & describe a variety of patterns that will only come to light when we can see all of the data at once

Principles of bad figure making

- "Chart junk": extraneous stuff that doesn't add to the data story
  - In some cases, though, a memorable graph will have a bit of superfluous design if it is clever
- Bad data: the data being presented tell a misleading story
- Problems with perception: the chart may be free of junk, but human visual perception will be misled by the chart's layout or dimensions

Human Perception

- Humans are better at seeing gradients when they are all the same hue and chroma but vary in luminance
- Need to be careful with color choice to make sure colors step through the options as intended
  - In other words, colors can be misleading if picked wrong (e.g. one color can unintentionally stand out more than the others)
- Shape and color are two "channels" that can encode information visually about your data
  - Color channel seems to work better than shape channel
  - Should try to avoid showing data through multiple channels
- Gestalt Rules
  - Proximity: things close together seem related
  - Similarity: things that look alike seem related
  - Connection: things visually tied together seem related
  - Continuity: Partially hidden objects are perceptually completed
  - Figure & ground: visual elements seen in either the foreground or the background
  - Common fate: elements moving in the same direction are seen as a unit (e.g. school of fish)

Decoding Graphs

- Humans do best when judging the relative position of things on a common scale
- Humans do worst when judging quantities as angles or areas (esp. areas of circles)

Honest & Good Judgment

- Not always good rules of thumb for what is an honest representation
  - Sometimes it makes sense not to start your Y-axis at 0, and if your axes are labeled, not necessarily misleading

Chapter 2: Getting Started

```
# Load in libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------------------
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts -----------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(socviz)
```

- Mostly an overview of R & RStudio
- A `tibble` is a tidyverse data.frame

```
# Tiny data set from socviz package
class(titanic)
```

```
## [1] "data.frame"
```

```
# Turn titanic into a tidyverse tibble
titanic_tb <- as_tibble(titanic)
titanic_tb
```

```
## # A tibble: 4 x 4
##   fate     sex        n percent
##   <fct>    <fct>  <dbl>   <dbl>
## 1 perished male    1364    62
## 2 perished female   126     5.7
## 3 survived male     367    16.7
## 4 survived female   344    15.6
```

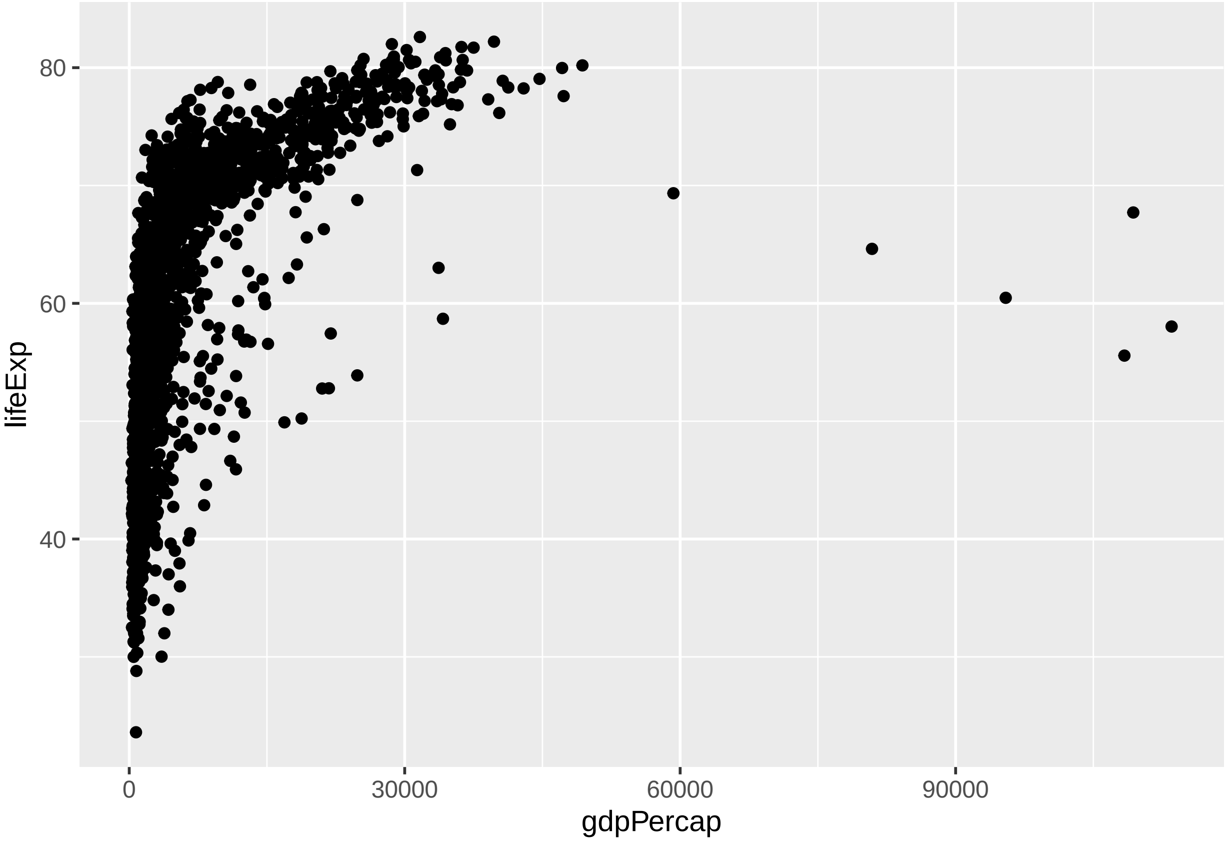
- The package `haven` is also good for reading in data of various formats
- Apparently "tidy" data is long format rather than wide format
  - **Note**: I am intrigued

Making my first figure

```
library(gapminder)
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

```
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap, y = lifeExp))
p + geom_point()
```


lifeExp
gdpPercap