INTRODUCTION TO DATA VISUALIZATION IN R

Carolyn Coberly

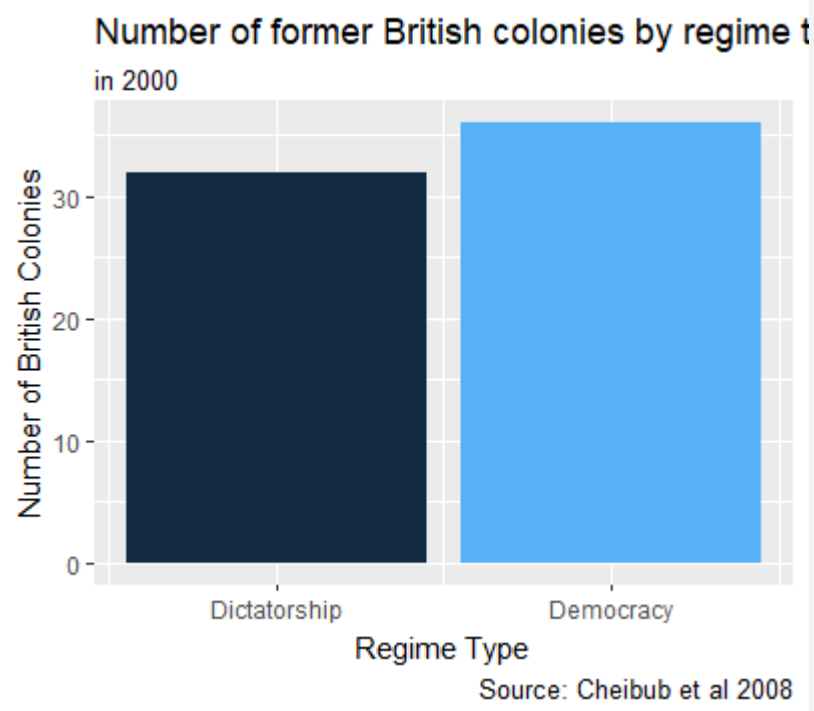
November 2017

GOALS

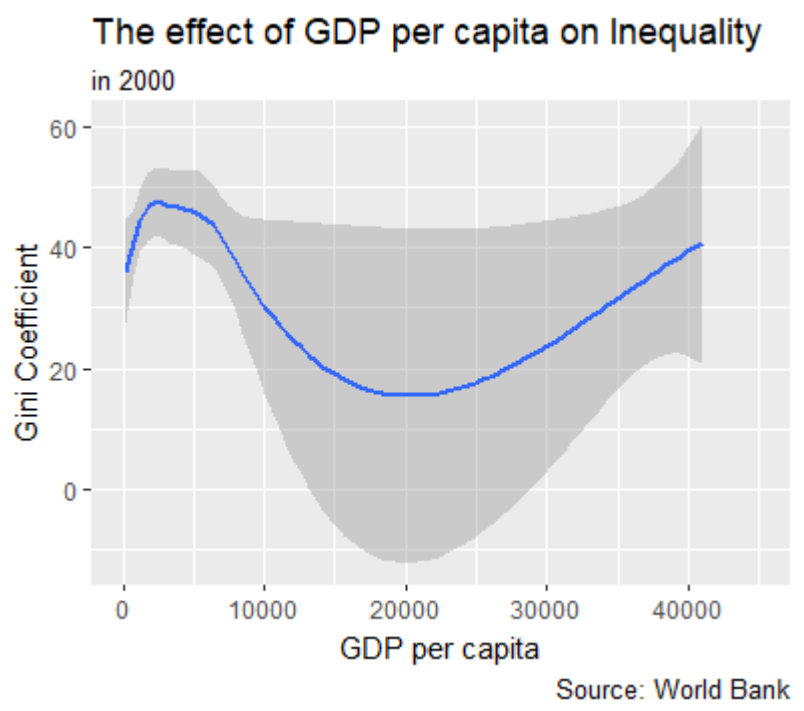
- Learn how to build graphs with ggplot2
  - Types of graphs
  - Colors and labels
  - Layering graphs
- Graph a regression line

TYPES OF VISUAL REPRESENTATION

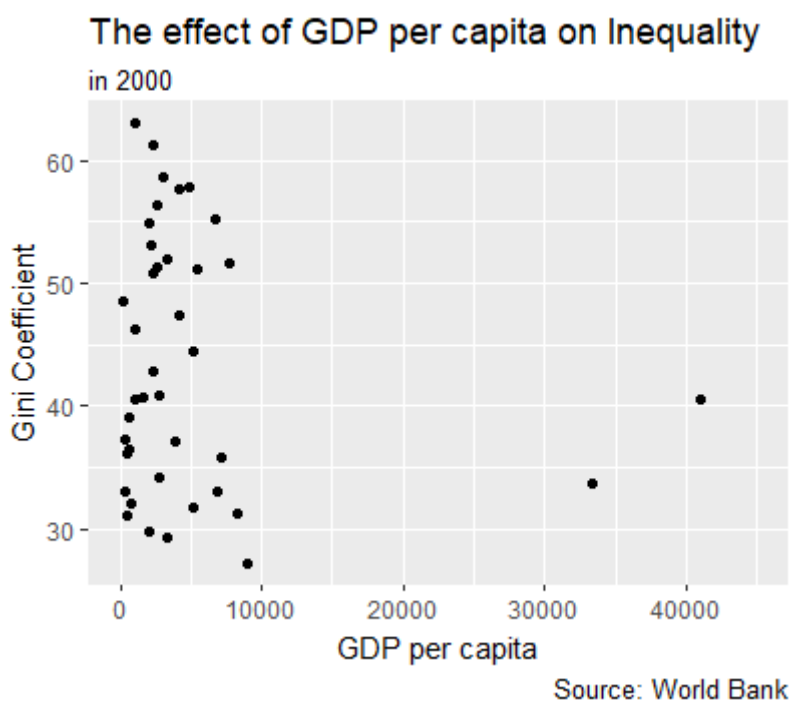
Bar Graph



Line Graph



Scatterplot



ELEMENTS OF A GRAPH

- Data
  - Your data should be in a single dataset
- Graph Type
  - What will best suit your data?
  - Do you need to graph confidence intervals?
- Variables (x and y)
- Labels
  - Titles, axes, data points, legend

DATA AND GRAPH TYPE

- The function ggplot(data = data) should always be the first line of code for a graph.
- You then need to write a + to add layers (graphs, labels, etc.)
- The next line should specify the type of graph you want:
  - geom_bar() creates a bar graph for one or two categorical variables
  - geom_histogram() creates a frequency distribution for a single variable
  - geom_point() creates a scatterplot
  - geom_line() creates a line connecting each data point
  - geom_smooth() creates a best fit line graph with confidence interval

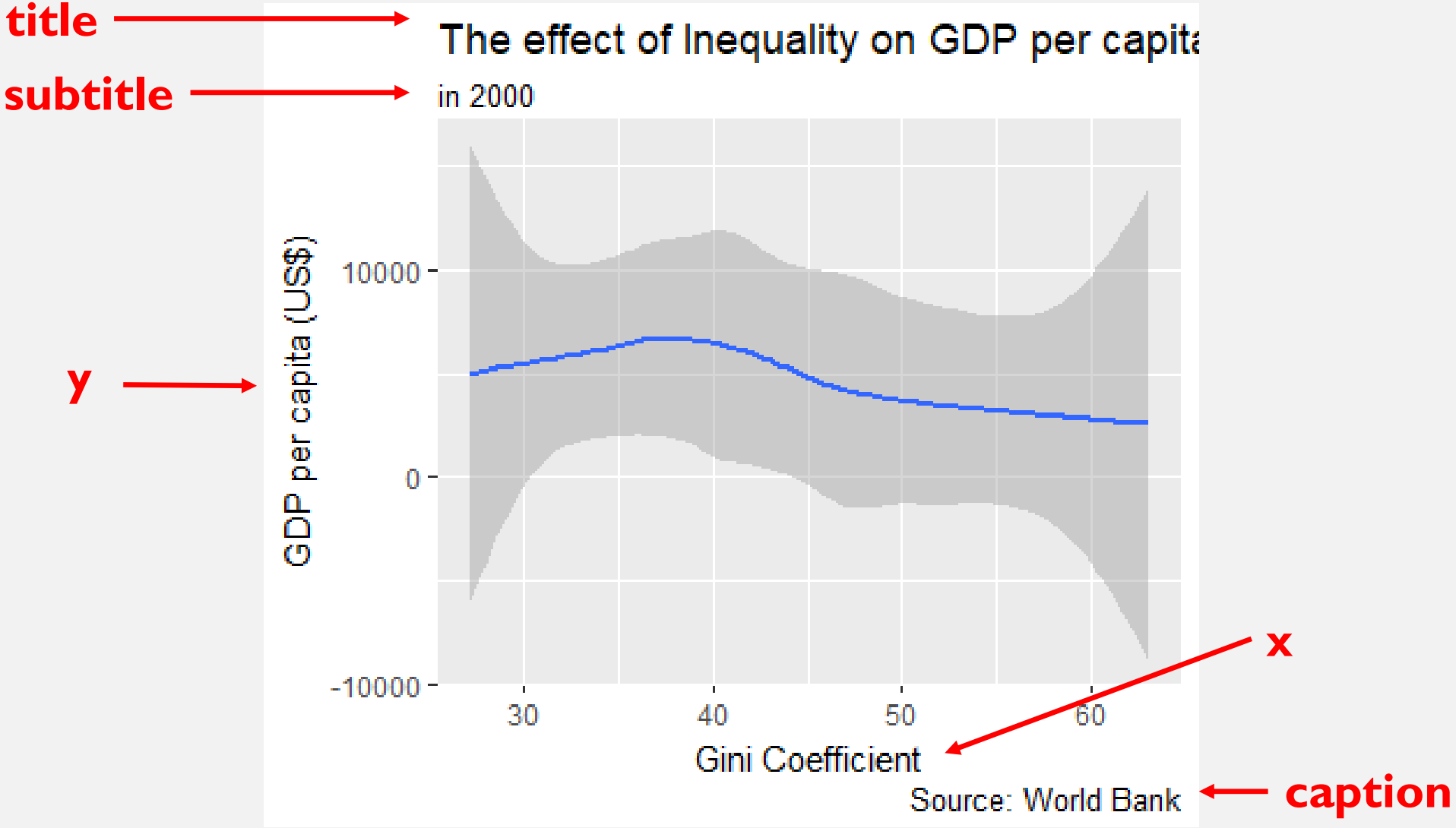
MAPPING

- The process of describing which variables you want to use is called "mapping" because you assign or "map" variables from a dataset onto a graph
- There are several ways you can map variables; you always want to choose aesthetic mapping: *aes()*
- The code *mapping = aes(x = var1, y = var2)* is inserted into the parentheses for graph type
  - For example, *geom_smooth(aes(x = gdp, y = birth_rate))*
- Watch the parentheses!

COLOR

- To change the color of your data:
  - In mapping, specify the variable that determines how colors should vary
    - For bar graphs, use *fill = var*
    - For scatterplots and line graphs, use *color = var*
- To choose which colors to use:
  - Use a palette: scale_color_brewer(palette = "name")
    - Palettes listed at http://www.sthda.com/sthda/RDoc/figure/text-mining/word-cloud-generator-rcolorbrewer-palettes.png
  - Assign colors individually (scatterplots only): scale_color_manual(values = c("value"="color"))
    - Note that you assign colors to the values of the variable that determines how colors vary

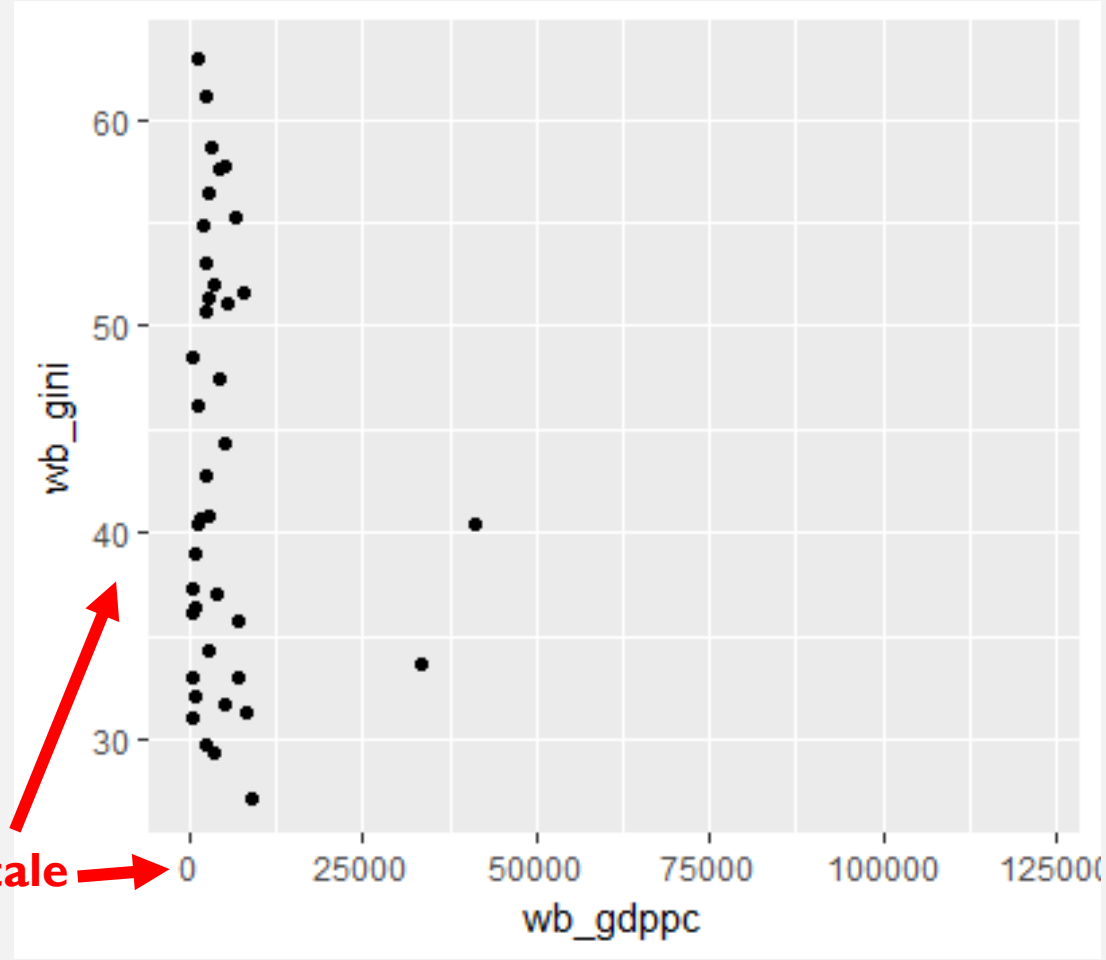
TITLES

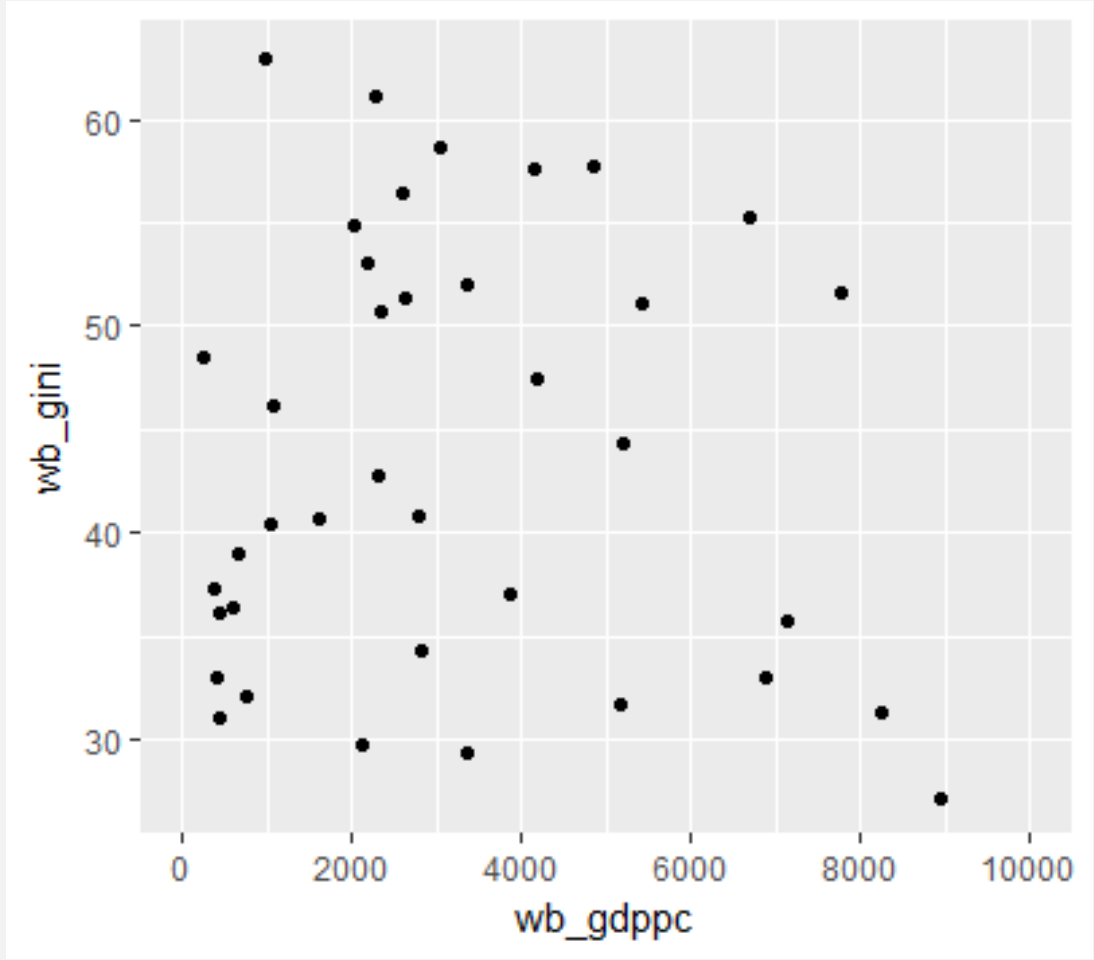

title
subtitle
y
x
caption
The effect of Inequality on GDP per capita
in 2000
GDP per capita (US$)
10000
0
-10000
30
40
50
60
Gini Coefficient
Source: World Bank

TITLES

- Add another line of code (after another +)
  - labs()
  - Specify the language for each label you want in “quotes”
    - labs(title=paste(“title”))
    - Each label has slightly different requirements

AXIS SCALE


60
50
40
30
wb_gini
0
25000
50000
75000
100000
125000
wb_gdppc
Scale


60
50
40
30
wb_gini
0
2000
4000
6000
8000
10000
wb_gdppc

AXIS SCALE

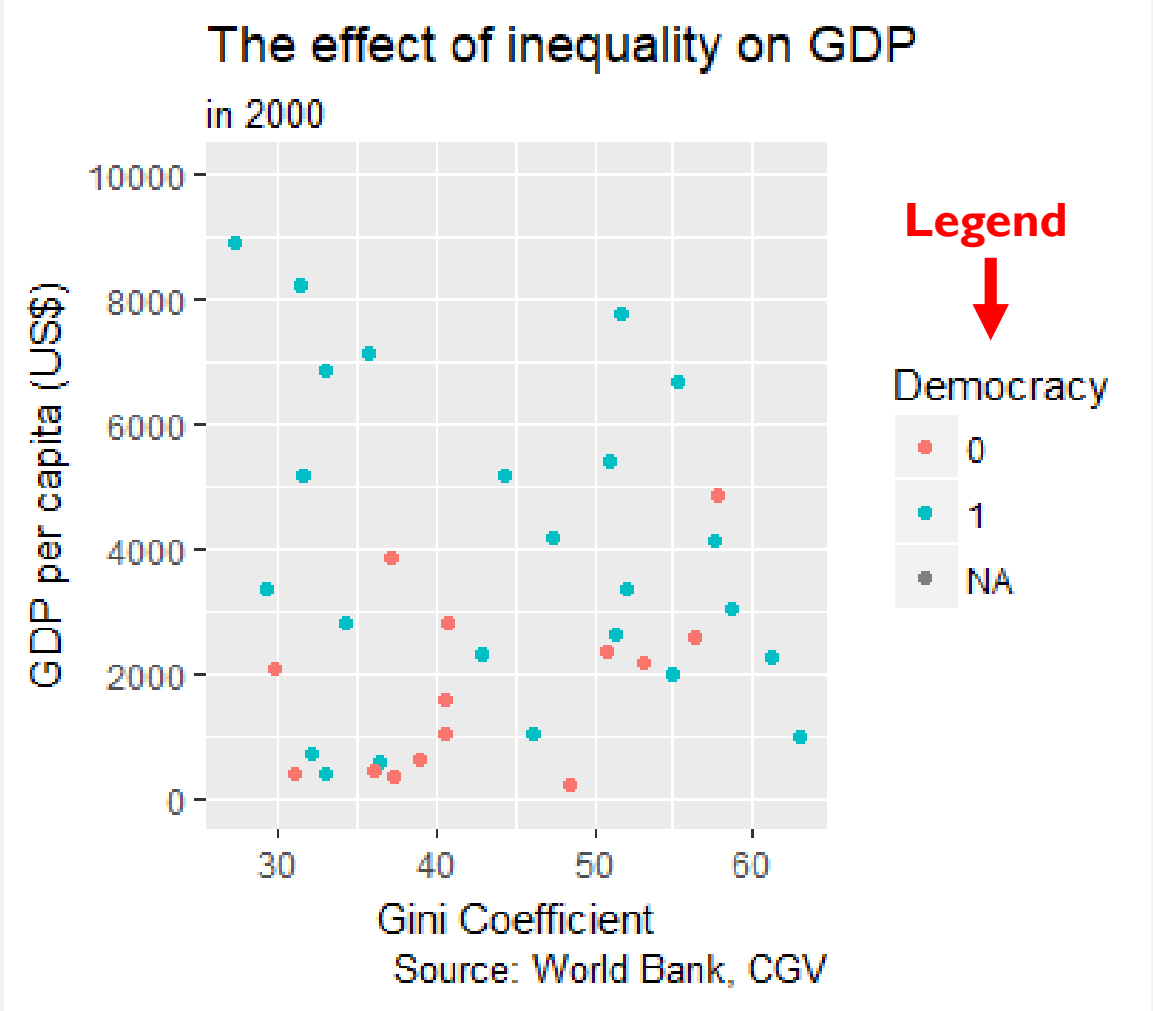
- Range of values (where the scale starts and stops)
  - coord_cartesian(xlim=c(first, last))
  - coord_cartesian(ylim=c(first, last))
- Distance between markers (which values are labeled on the scale)
  - scale_x_continuous(breaks = seq(first, last, by = distance))
  - scale_y_continuous(breaks = seq(first, last, by = distance))
- Each is a separate line under ggplot (not a part of the label function)

LABELING POINTS

- It is possible to use a text phrase (such as country name) instead of or in addition to points on a scatterplot:
  - geom_text(aes(x, y, label=labelvar))
- It is often difficult to distinguish individual names as text. Alternate method:
  - library(ggrepel)
  - geom_label_repel(aes(label=labelvar))
- To only label outliers, subset the data and label using geom_label

LEGENDS

- Only appear when there is a third variable that you use on the graph (e.g., varying the color of your points by another variable)
- Changing the position of the legend:
  - + *theme(legend.position = "position")*
- Changing the title of the legend:
  - In *labs(color = "name")*
- To change the names of the values in the legend, you need to change them (or label them) in the variable.



LAYERING GRAPHS

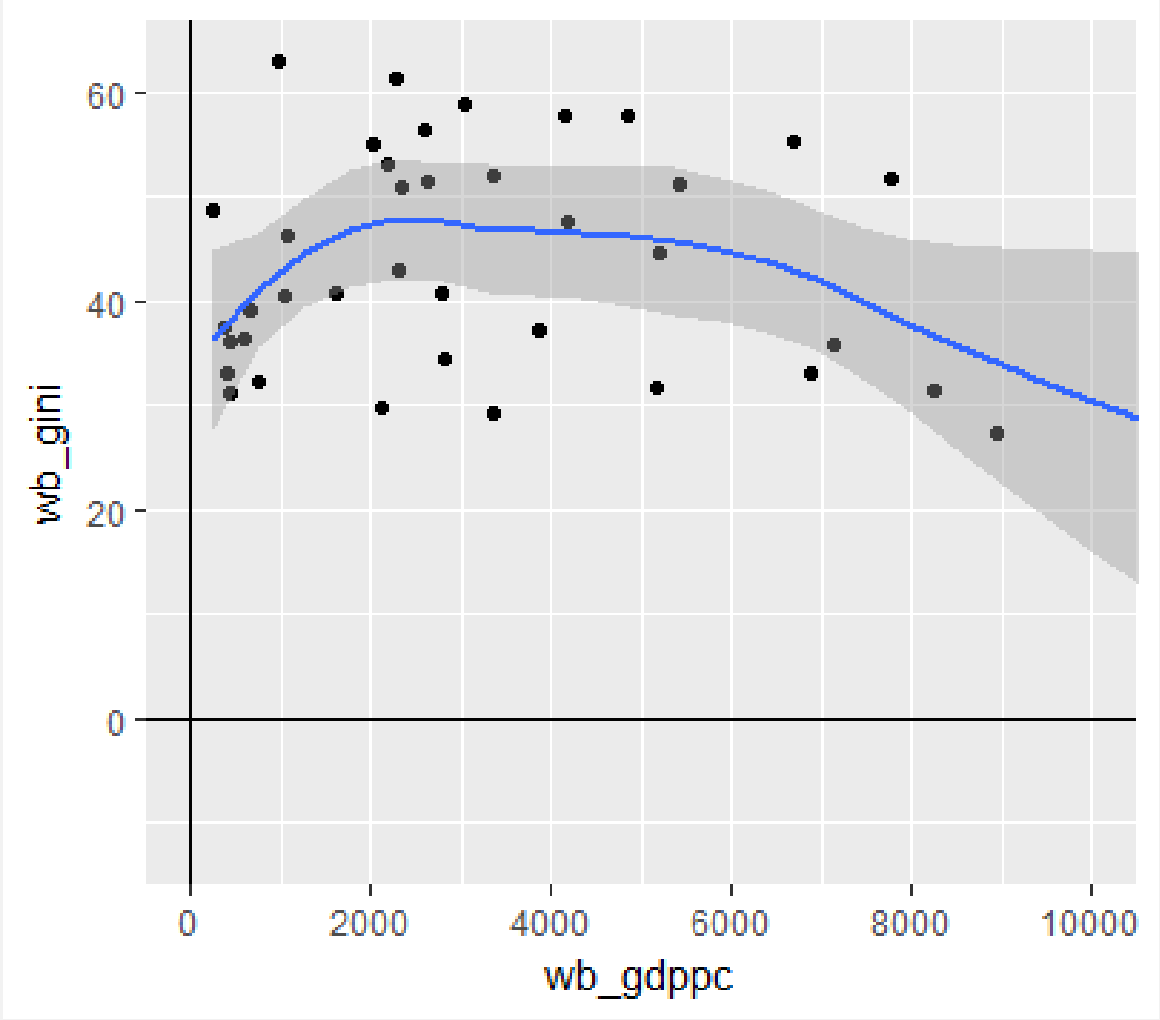
You can place more than one type of function on the same graph simply by adding them together. For example, to show both a scatterplot and a line that fits it:

ggplot(data, mapping) +

geom_point() +

geom_smooth()

Similarly, you can add reference lines at the intercept of your choice with the functions for horizontal and vertical lines.



GRAPH A BIVARIATE LINEAR REGRESSION

$$y = \alpha + \beta_1 x_1 + \varepsilon$$

- geom_smooth() returns a best fit line as its default, but this is not a linear regression.
- To graph the line associated with the coefficient of a bivariate OLS regression, you use the same function, but specify the model and formula:
  - *geom_smooth(aes(x=var1, y=wb_var2), method='lm', formula= y ~ x)*

HELP!

- *R for Data Science* by Garrett Grolemund and Hadley Wickham
  - Available at http://r4ds.had.co.nz/ or for purchase on Amazon.
- *R Graphics Cookbook* by Winston Chang
  - https://ase.tufts.edu/bugs/guide/assets/R%20Graphics%20Cookbook.pdf
- Cheat Sheets:
  - ggplot2: https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf