SPEC Lab REU R Resources: Data visualization with `ggplot2` Line Graphs

Alix Ziff, Gaea Morales, and Zachary Johnson based on earlier materials by Therese Anders

Summer 2020, Version 3: June 12

ggplot2 continued

This walk-through-work will show us how to make clear & concise line graphs.

Continute working with the WDI dataset from the previous walkthrough on scatterplots, and load ggplot2 as well as the necessary data if you are starting from a new R session.

```
setwd("/Volumes/GoogleDrive/My Drive/Training Data Science/0. Training Data")
```

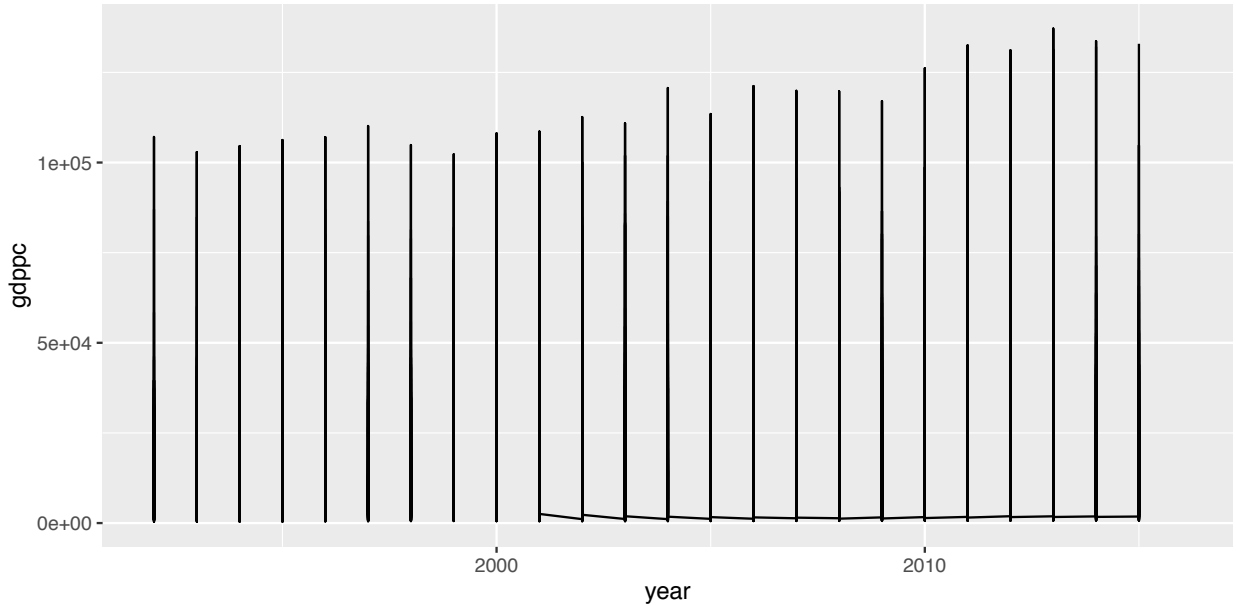
```
dat <- read.csv("wdi_cleaned_part2.csv")

library(ggplot2)
```

Line graphs

One of the most common uses for line graphs is the display of data over time. For example, we could be plotting the evolution of GDP per capita over time. Within the `aes()` command, we specify which variable to be plotted on the x- and y-axis. The geometric object we use for line graphs is `geom_line()`.

We are working with a panel data set. This means that we have multiple observations over time for each country. Displaying all this information without grouping or subsetting does not results in a plot that is useful. In the plot below, `geom_line()` is trying to draw a line *connecting* all non-missing country-year observations for the variable `gdppc`.

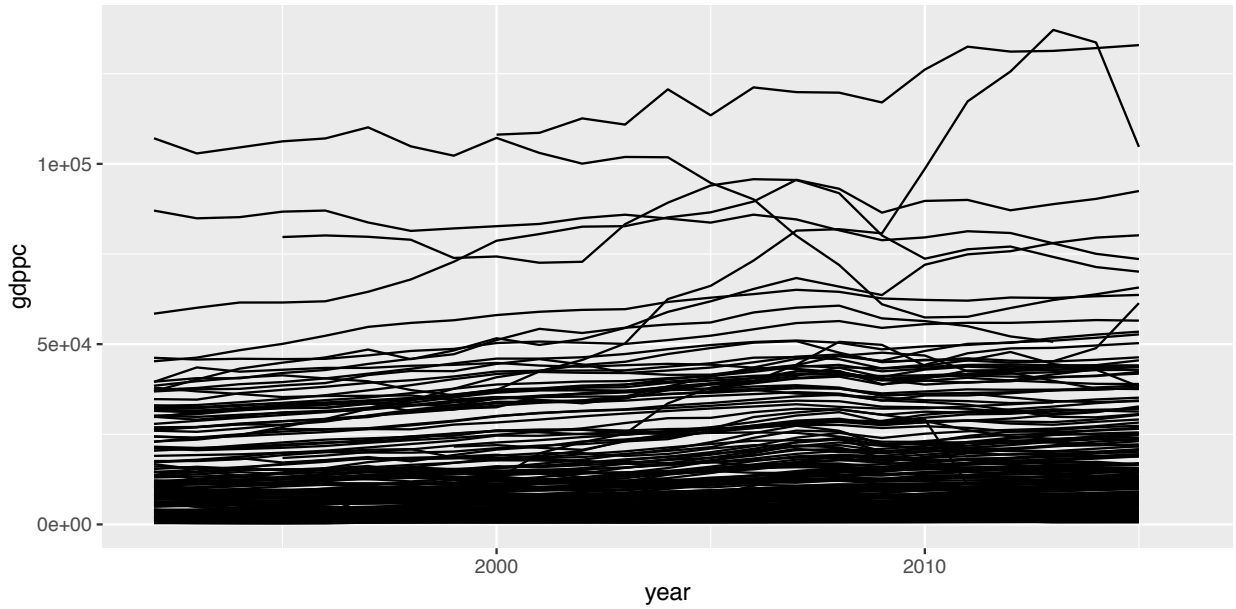
```
ggplot(dat, aes(x = year, y = gdppc)) +
  geom_line() #connecting points on our graph
```



Helpful Hint: Here we are creating a line graph which *connects points*, this is a different type of graph than a scatterplot and shouldn't be confused with trend lines which outlines the pattern of existing points. You can see the connection more clearly the more you zoom in your plot.

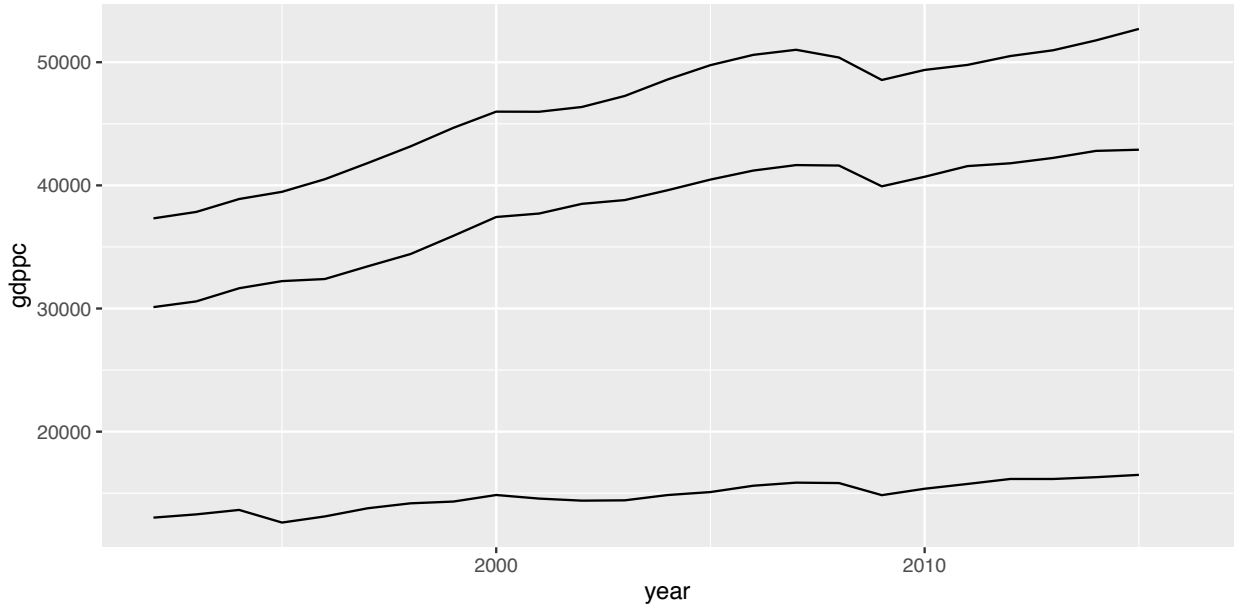
In order to plot one line per country, we can specify the `group` parameter inside the aesthetics argument. However, in this data set, even if we grouped the data by country, there are too many groups (countries) to be displayed in one graph.

```
ggplot(dat, aes(x = year, y = gdppc, group = country)) +
  geom_line() #now we have a horizantal-ish line for each country
```



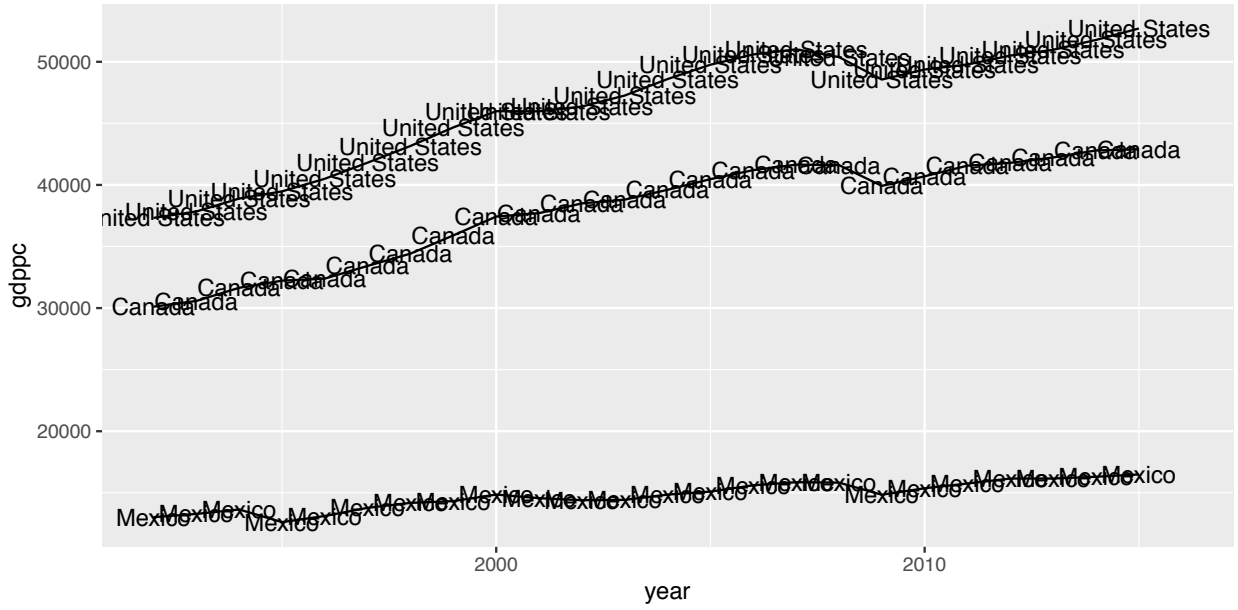
We will therefore use a subset of countries and plot the evolution of their per capita GDP over time. Suppose for example, we wanted to see how the per capita GDP has changed over time in member countries of the The North American Free Trade Agreement (NAFTA), that is Canada, Mexico, and the US. We can use `subset()` inside the `ggplot()` function to plot a separate line for each of the countries.

```
ggplot(subset(dat, country %in% c("Canada", "Mexico", "United States")),
       aes(x = year, y = gdppc, group = country)) +
  geom_line()
```



This graph does not tell us which line represents the evolution of per capita GDP in which country. We could add labels to each line to denote the country it represents. We could do this with the `geom_text()` function but this will add a label for each country-year observation.

```
ggplot(subset(dat, country %in% c("Canada", "Mexico", "United States")),
       aes(x = year, y = gdppc, group = country, label = country)) +
  geom_line() +
  geom_text()#label each country
```



A better way to label each line (and not each country-year observation) is using the `directlabels` package and the `geom_dl()` geometric object in `ggplot()`. *Note* that you have to specify a method for the `geom_dl()` function to work (here: `"last.points"`). We can use the `cex` argument to control the size of the text labels. Here, I am also adjusting the range of the x-axis to ensure that all lines are properly labeled.

The graph shows that the evolution of wealth in the three countries is intimately connected. Crises and upswings in show the same patterns for the US, Canada, and Mexico. However, the increase of per capita GDP appears to be a lot steeper in Canada and the US than in Mexico, despite similar trends.

```
library(directlabels)
library(ggplot2)
library(grid)
library(quadprog)
library(proto)

attach(dat)

ggplot(subset(dat, country %in% c("Canada", "Mexico", "United States")),
       aes(x = year, y = gdppc, group = country)) +
  geom_line() +
  geom_dl(aes(label = country), method = ("last.points"), cex = 0.5) +
  coord_cartesian(xlim = c(1992, 2020))
```

It is not necessary in this example, but sometimes we would like to show which observations go into the computation of the line by adding points. Remember, that ggplot uses layers to graph multiple geometric objects in one plot. We can therefore just overlay the line plot with a `geom_point()` layer.

```
ggplot(subset(dat, country %in% c("Canada", "Mexico", "United States")),
       aes(x = year, y = gdppc, group = country)) +
  geom_line() +
  geom_dl(aes(label = country), method = ("last.points"), cex = 0.5) +
  geom_point(color = "tomato") +
  coord_cartesian(xlim = c(1992, 2020))
```