Introduction to ggplot

Abhinav Anand

Setup

The following discussion assumes you have donwloaded R and RStudio. Additionally, the package suite `tidyverse()` which includes the package `ggplot2()` needs to be included.

1. For downloading R, visit https://cran.r-project.org/
2. For downloading RStudio visit https://www.rstudio.com/
3. For downloading `ggplot2()`, type `install.packages("ggplot2")` or equivalently for `tidyverse()` type `install.packages("tidyverse")`

Introduction to ggplot

The `gg` of `ggplot` stands for (layered) `grammar of graphics` (Wilkinson 2005), (Wickham 2010). This idea will be further explored by the means of data from the package `gapminder()`. To install, type `install.packages("gapminder")` in the RStudio console.

```
data_gapminder <- gapminder::gapminder
```

Notes

1. Why `<-` as opposed to `=` ?
2. Why `gapminder::gapminder` ?
3. What is a dataframe?

A data frame is a rectangular collection of variables (in the columns) and observations (in the rows). It's different from a 'mere' matrix since the columns have variable names usually.

Questions?

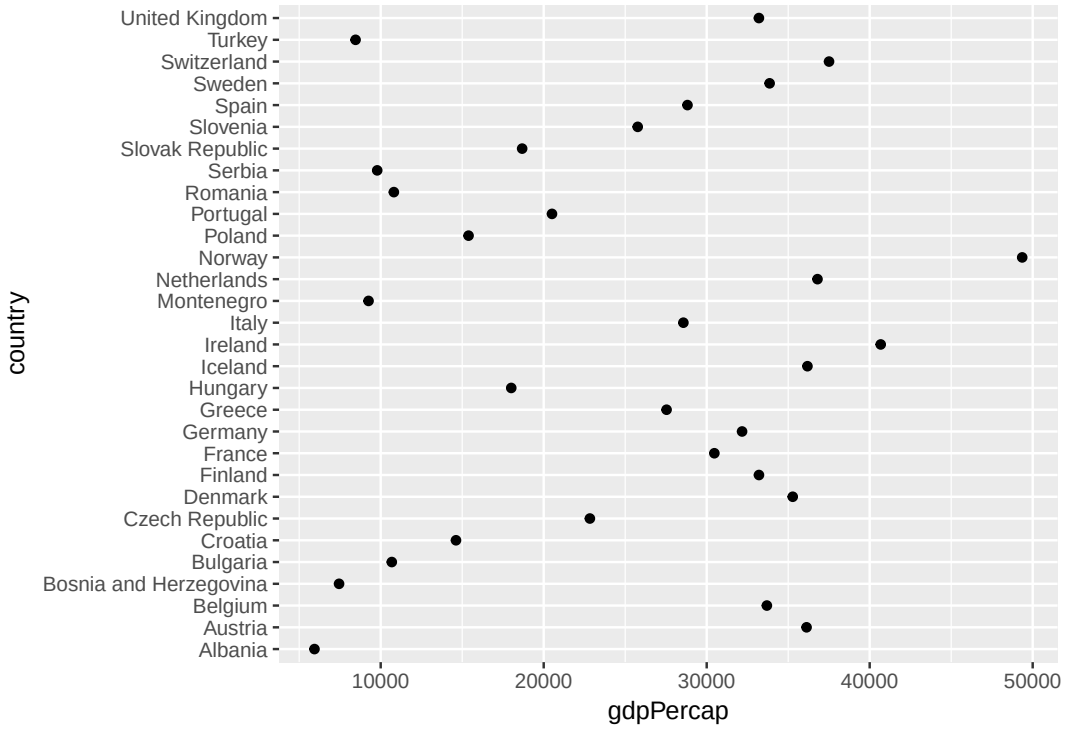
Are "Western" countries richer than "Eastern" countries?

The current state of Europe (in 2007):

```
data_eur_2007 <- data_gapminder %>%
  dplyr::filter(year == 2007) %>% #isolates variables for year 2007
  dplyr::filter(continent == "Europe")

plot_eur <- ggplot(data = data_eur_2007) +
  geom_point(mapping = aes(x = gdpPercap, y = country))

plot_eur
```



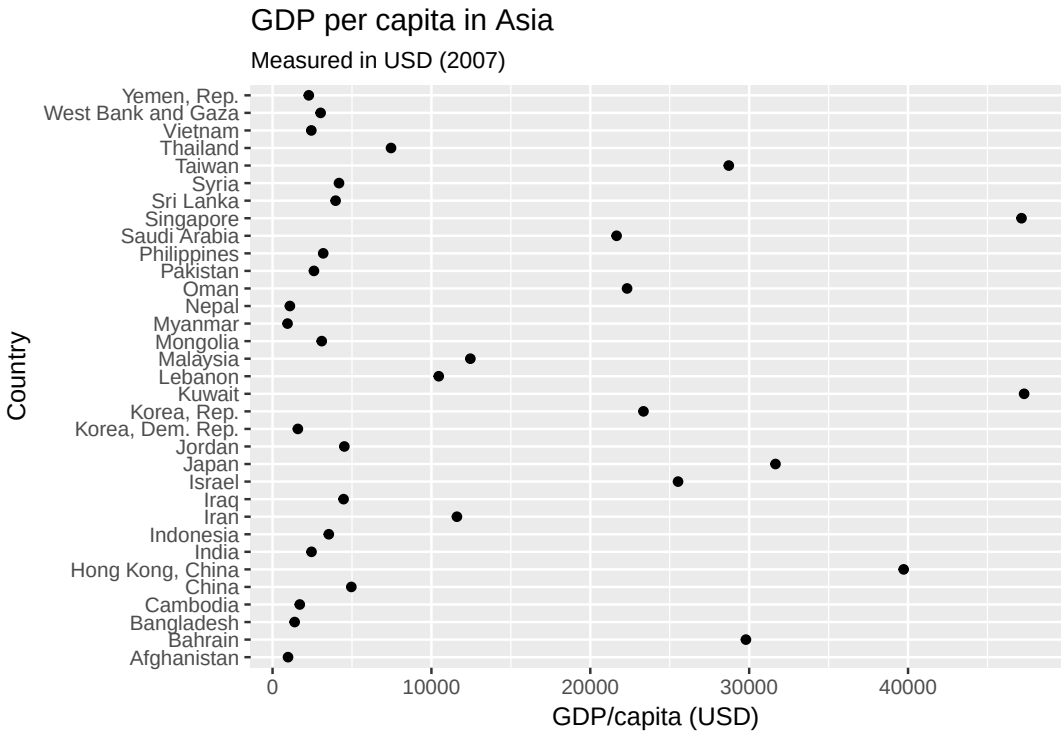
There is high variation—from Albania to Norway.

What about Asian countries in 2007?

```
data_asia_2007 <- data_gapminder %>%
  dplyr::filter(year == 2007) %>% #isolates variables for year 2007
  dplyr::filter(continent == "Asia") #collect only Asian countries
```

```
plot_asia <- ggplot(data = data_asia_2007) +
  geom_point(mapping = aes(x = gdpPercap, y = country)) +
  labs(x = "GDP/capita (USD)",
       y = "Country",
       title = "GDP per capita in Asia",
       subtitle = "Measured in USD (2007)")

plot_asia
```



Again, large variation among Asian countries but what about the bounds? What can we say about the question?

Graphics

We start with the function `ggplot()`. It creates a coordinate system that we will add layers to. The first argument is the dataset to use in the graph.

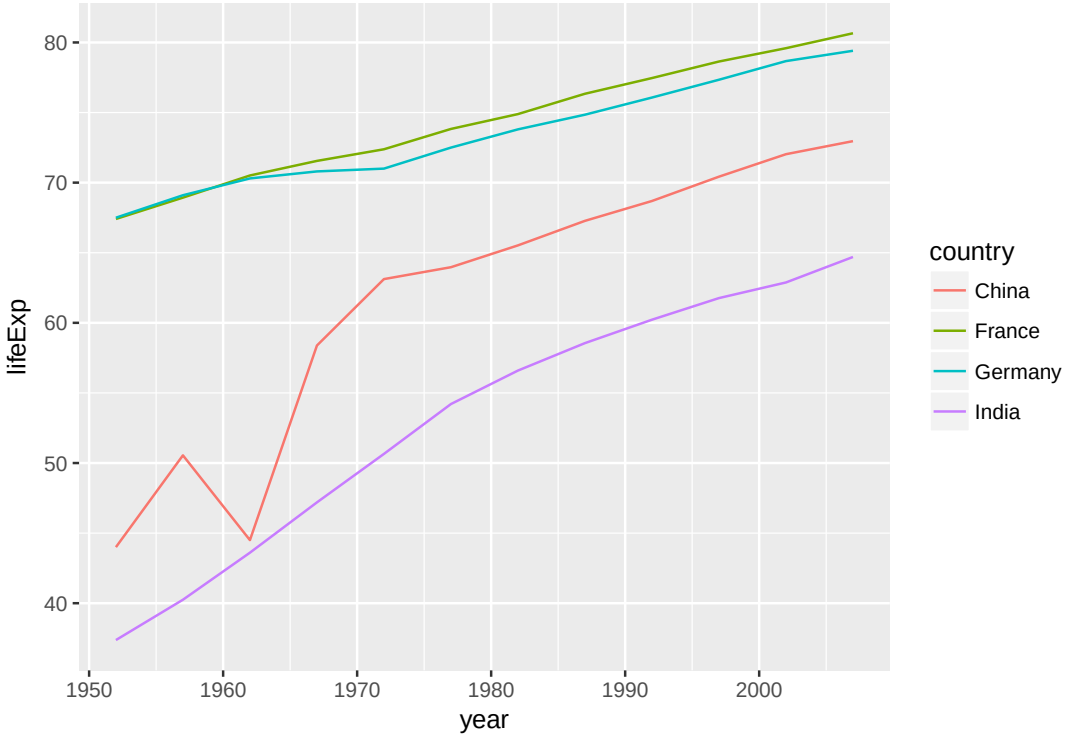
`ggplot(data = data_eur_2007)` creates an empty graph. The function `geom_point()` adds a layer of points to our plot. Each geom function in ggplot2 takes a mapping argument. This

defines how variables in our dataset are mapped to aesthetics such as axes, colors, shapes etc. The x and y arguments of `aes()` specify which variables to map to the x and y axes. Variables can also be mapped to aesthetics such as colors, shapes, size etc.

A More Granular Look: China, India, France, Germany

```
data_CIFU <- data_gapminder %>%
  dplyr::filter(country %in% c("China", "India", "France", "Germany"))

(plot_CIFU_life_exp <- ggplot(data = data_CIFU) +
  geom_line(mapping = aes(x = year, y = lifeExp, color = country))
  )
```



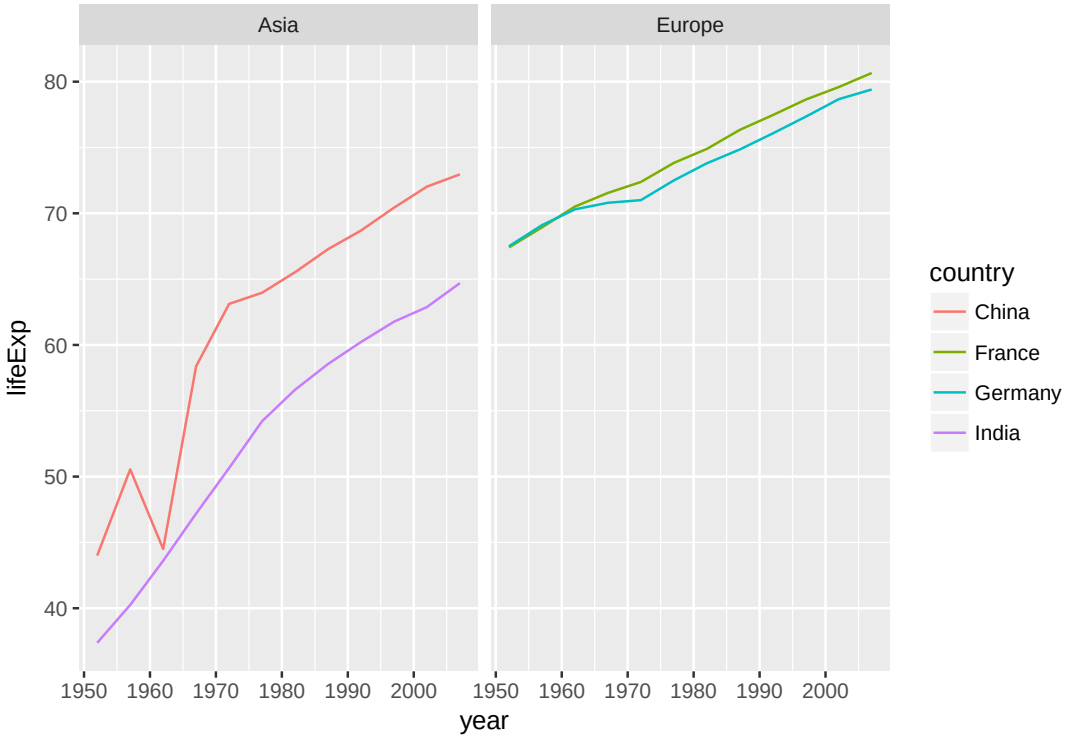
Notes

1. Plots can be stored as variables too.
2. Other aesthetic attributes: shape, size, alpha (transparency) etc.
3. Note where to put the + sign.

4. `geom_line()` as opposed to points. Other "geoms" are `geom_smooth`, `geom_boxplot`, `geom_bar` etc.

Faceting

```
(plot_CIFU_life_cont <- ggplot(data = data_CIFU) +
  geom_line(mapping = aes(x = year, y = lifeExp, color = country)) +
  facet_wrap(~ continent)
 )
```



Notes

1. To facet on one variable ('continent' here), use `facet_wrap()`.
2. To facet on two variables, use `facet_grid()`

Geoms in `ggplot()`

REWRITE:

A geom is the geometrical object that a plot uses to represent data. People often describe plots by the type of geom that the plot uses. For example, bar charts use bar geoms, line charts use line geoms, boxplots use boxplot geoms, and so on. Scatterplots break the trend; they use the point geom. As we see above, you can use different geoms to plot the same data.

However, not every aesthetic works with every geom. You could set the shape of a point, but you couldn't set the "shape" of a line. On the other hand, you could set the linetype of a line. Multiple geoms could be part of the same graph. ggplot2 provides over 30 geoms

To display multiple geoms in the same plot, add multiple geom functions to ggplot():

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) + geom_smooth(mapping = aes(x = displ, y = hwy))

This, however, introduces some duplication in our code. Imagine if you wanted to change the y-axis to display cty instead of hwy. You'd need to change the variable in two places, and you might forget to update one. You can avoid this type of repetition by passing a set of mappings to ggplot(). ggplot2 will treat these mappings as global mappings that apply to each geom in the graph. In other words, this code will produce the same plot as the previous code:

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_point() + geom_smooth()

Bar Charts

Boxplots

`coord_flip()`

p2 <- ggplot(housing, aes(x = Home.Value)) p2 + geom_histogram()

`themes()`: Built-in themes include:

theme_gray() (default) theme_bw() theme_classc()s

The Main FAQ

Wide Versus Long Data

References

Wickham, Hadley. 2010. "A Layered Grammar of Graphics." *Journal of Computational and Graphical Statistics* 19 (1): 3–28.

Wilkinson, Leland. 2005. *The Grammar of Graphics (Statistics and Computing)*. Berlin, Heidelberg: Springer-Verlag.