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 and can contain different "types", say numeric and categorical and character and logical all together.

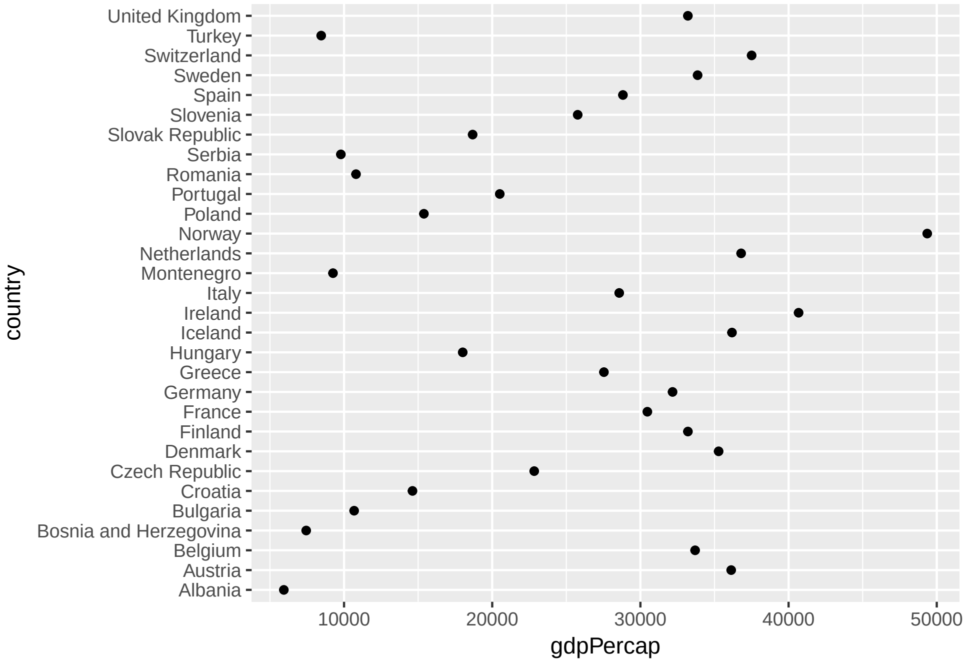
Questions

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)))
```

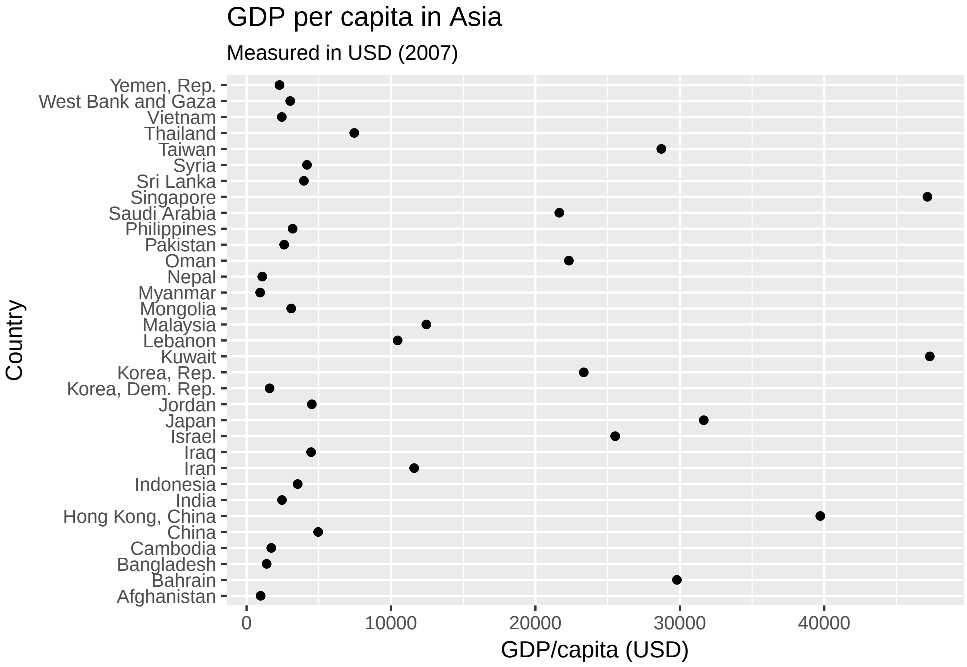


There is high variation—from Albania to Norway.

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)"))
```



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.

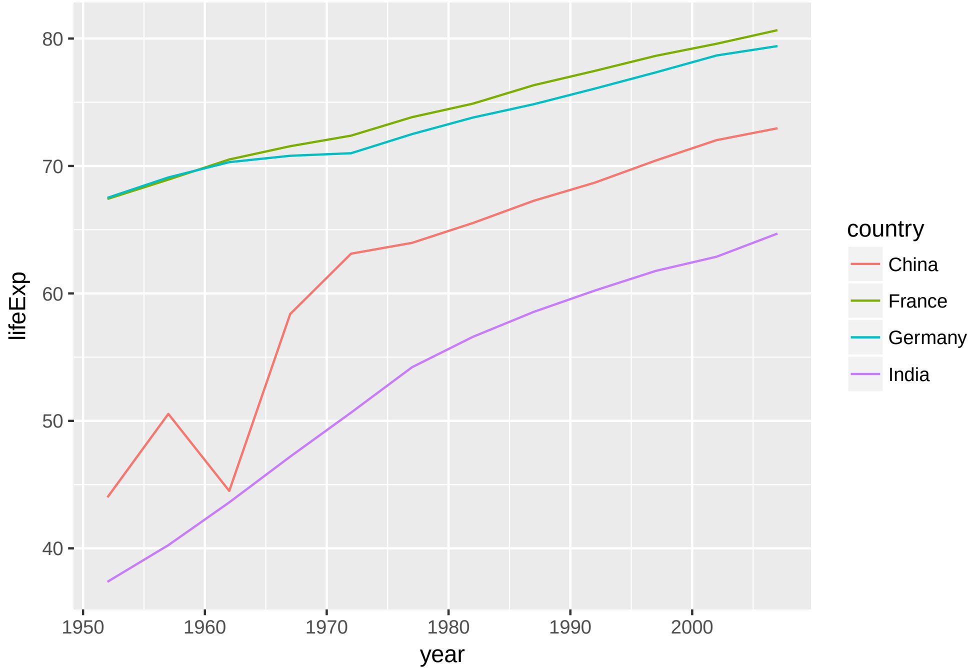
`ggplot(data = data_eur_2007)` creates an empty graph. The function `geom_point()` adds a layer of points to our plot. Each geom function in ggplot2 takes a mapping argument. This defines how variables in our dataset are mapped to aesthetics such as axes, colors, shapes etc. The x and y arguments of `aes()` specify which variables to map to the x and y axes. Variables can also be mapped to aesthetics such as colors, shapes, sizes etc.

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

What about life expectancy in these countries with time?

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

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

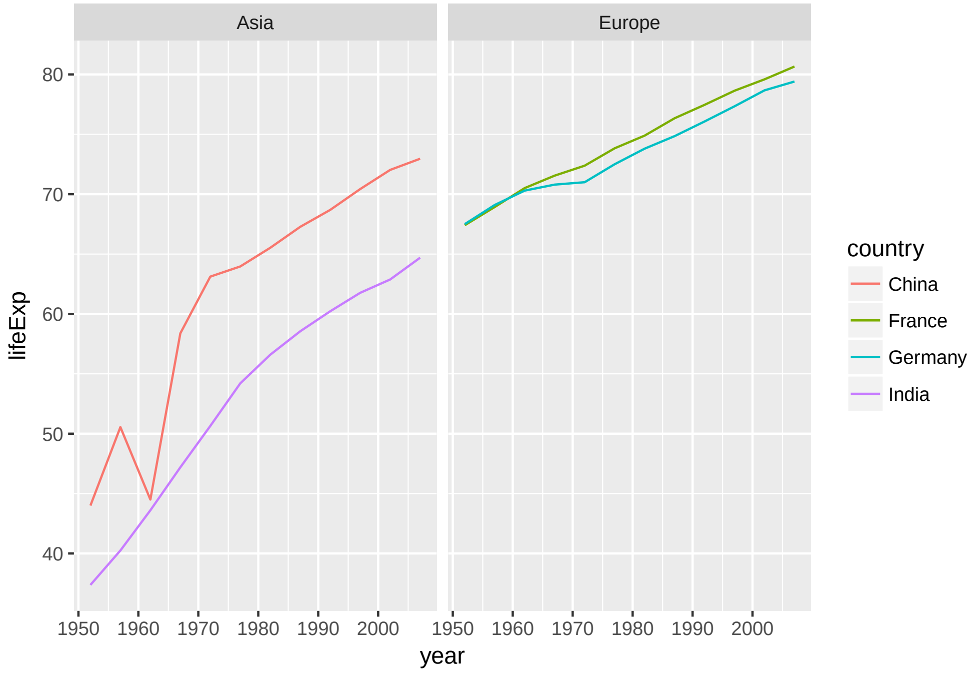


Notes

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_CIFG_life_cont <- ggplot(data = data_CIFG) +
  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()`

The Notion of Geoms in `ggplot()`

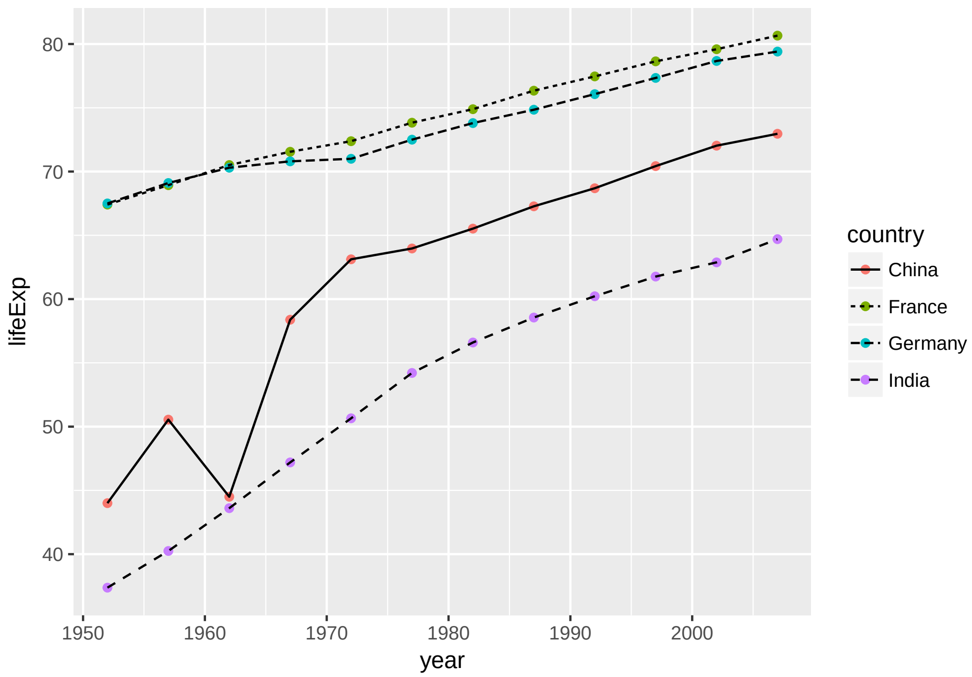
Sometimes one needs bar charts, sometimes histograms; at other times scatterplots or lines etc. All these are different “geoms” in `ggplot()`. Scatterplots can be made with `geom_point()`, lineplots can be implemented with `geom_line()`; boxplots require `geom_boxplot()`, barcharts need `geom_bar()` and so on. Multiple geoms could be part of the same graph. The library ggplot2 provides over 30 geoms.

Each geom will need “aesthetic” parameters: for example, which datasets form the x axis? Which ones form the y axis? What colors to use for different variables?

Somewhat unsurprisingly, not every aesthetic works with every geom. We could set the shape parameter of a point, but cannot do so for a line.

This is how we could include multiple geoms in the same plot:

```
(plot_CIFG_mult_geom <- ggplot(data = data_CIFG) +
   geom_point(mapping = aes(x = year, y = lifeExp, color = country)) +
   geom_line(mapping = aes(x = year, y = lifeExp, linetype = country))
  )
```



Note however, that we need to rewrite the code for aesthetic parameters: `x = year, y = lifeExp`. We could re-express the same idea in fewer lines—by passing some common aesthetic parameters as global options—by including them in the main `ggplot` argument:

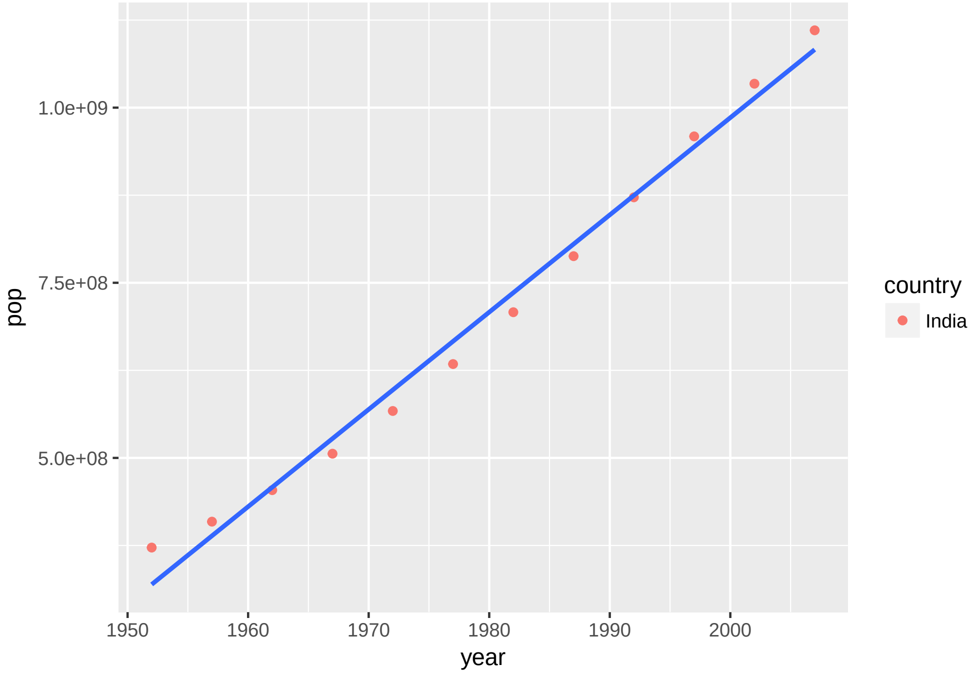
```
plot_CIFG_mult_geom_2 <- ggplot(data = data_CIFG,
                                mapping = aes(x = year,
                                              y = lifeExp)) +
  geom_point(mapping =  aes(color = country)) +
  geom_line(mapping = aes(linetype = country))
```

Plotting linear trends: `geom_smooth()`

Fitting a line to a set of observations is linear smoothing. Fitting a polynomial of degree 2 is quadratic smoothing and so on. When we wish to plot observations and the best linear fit computed by linear regression, we need to use `geom_smooth()` with the option `method = "lm"` where `lm()` stands for “linear model”.

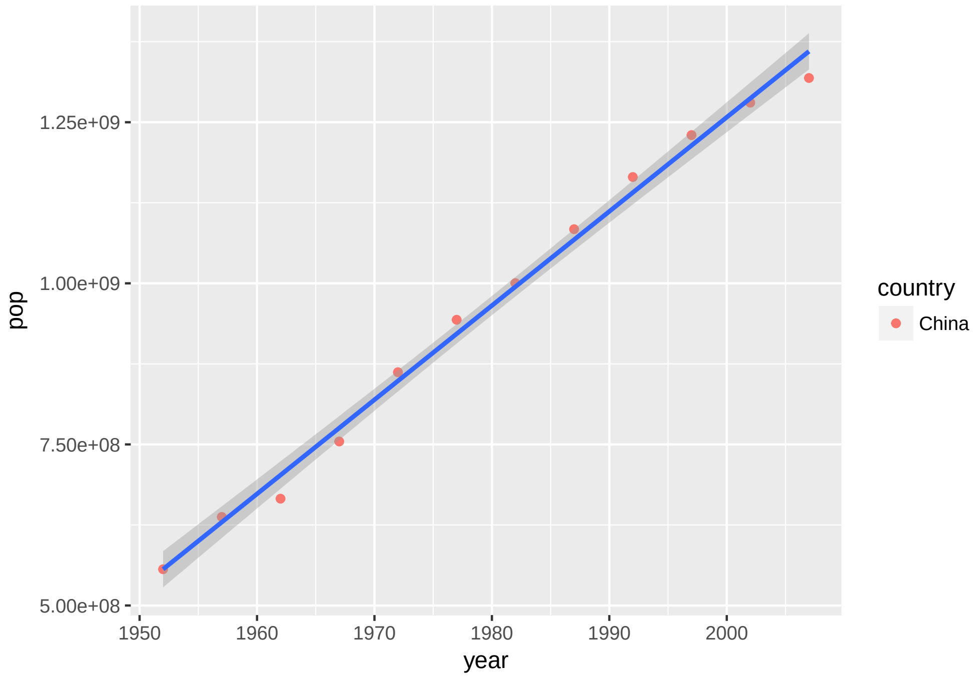
Question: What was population growth in India?

```
(plot_CIFG_pop_Ind <- ggplot(data = filter(data_CIFG, #why no +?
                                           country == "India"),
                         mapping = aes(x = year, y = pop)) +
  geom_point(mapping = aes(color = country)) +
  geom_smooth(method = "lm", se = F) #se = "standard errors"
 )
```



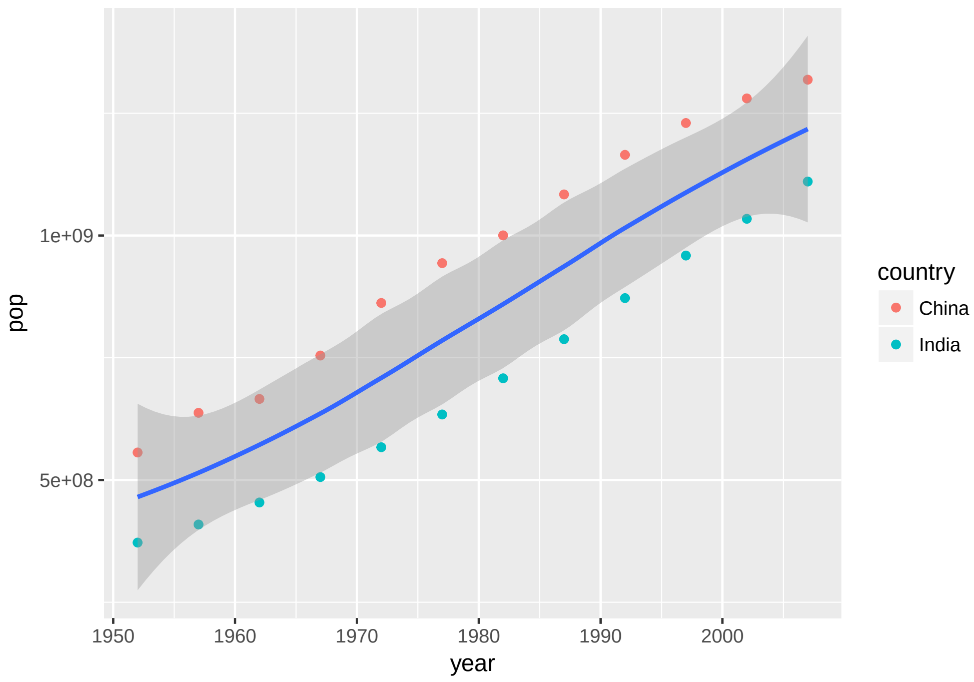
Question: What about the same in China?

```
(plot_CIFG_pop_China <- ggplot(data = filter(data_CIFG,
                                             country == "China"),
                         mapping = aes(x = year, y = pop)) +
  geom_point(mapping = aes(color = country)) +
  geom_smooth(method = "lm") #se = T by default
 )
```



The default smoothing method however is “loess” (locally weighted scatterplot smoothing)

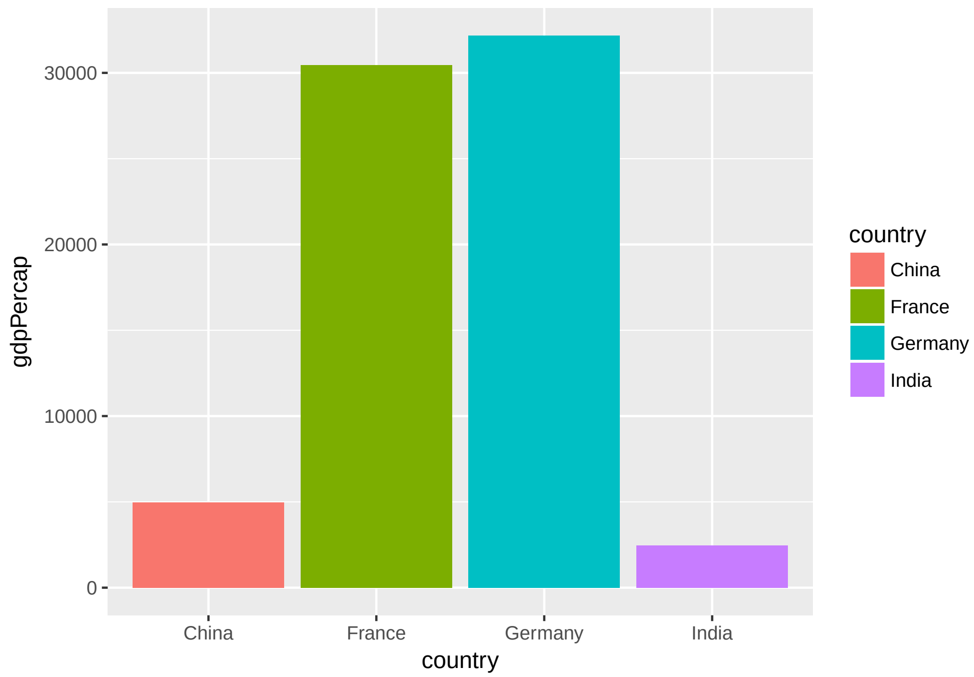
```
(plot_CIFG_pop_Ind <- ggplot(data = filter(data_CIFG,
                                            continent == "Asia"),
                             mapping = aes(x = year, y = pop)) +
  geom_point(mapping = aes(color = country)) +
  geom_smooth()
 )
```



Bar Charts

What about GDP per capita of CIFG countries in 2007?

```
(plot_CIFG_pop_2007 <- ggplot(data = filter(data_CIFG,
                                            year == 2007),
                              mapping = aes(x = country,
                                            y = gdpPercap)) +
   geom_bar(mapping = aes(fill = country), #what does this mean?
            stat = "identity") #what does this mean?
 )
```



REWRITE:

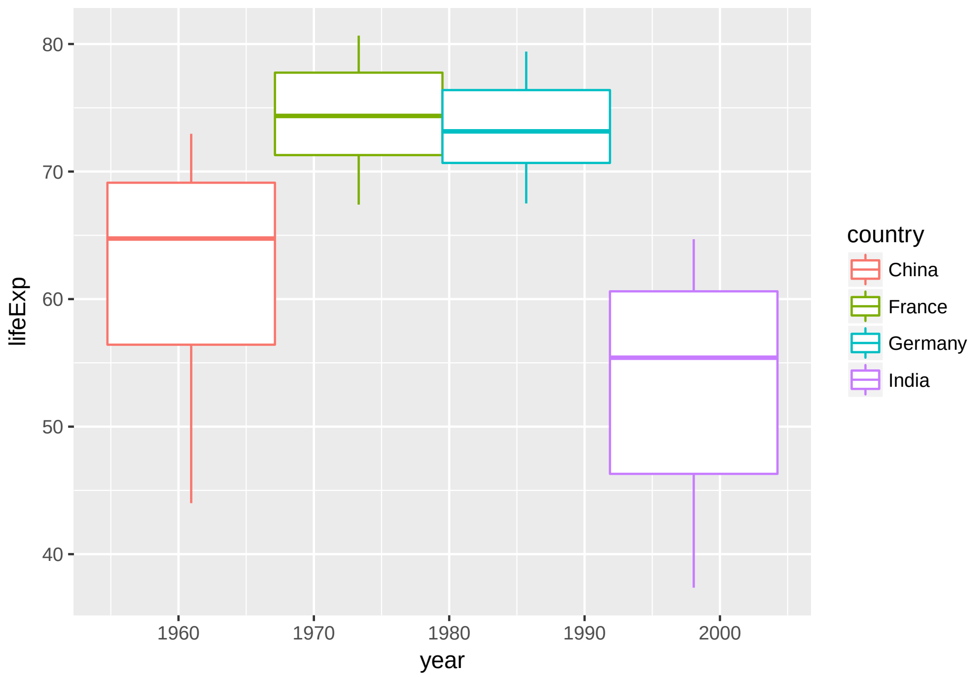
Some plot types (such as scatterplots) do not require transformations–each point is plotted at x and y coordinates equal to the original value. Other plots, such as boxplots, histograms, prediction lines etc. require statistical transformations:

for a boxplot the y values must be transformed to the median and 1.5(IQR)

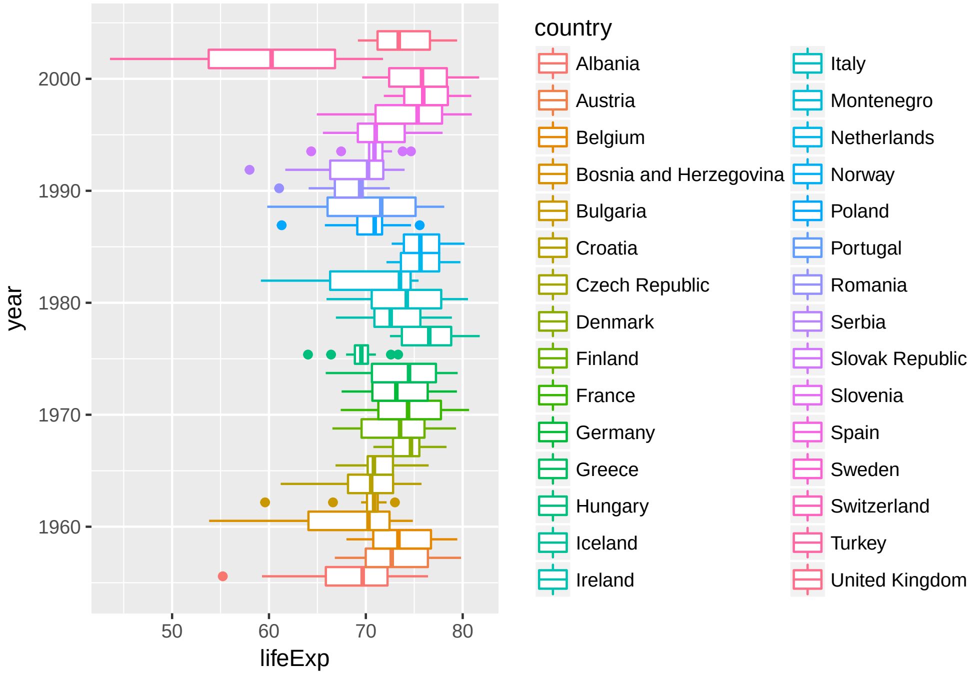
Each geom has a default statistic, but these can be changed. For example, the default statistic for geom_bar is stat_bin:

Boxplots

```
(plot_CIFG_box <- ggplot(data = data_CIFG,
                         mapping = aes(x = year,
                                       y = lifeExp,
                                       color = country)) +
   geom_boxplot()
   )
```

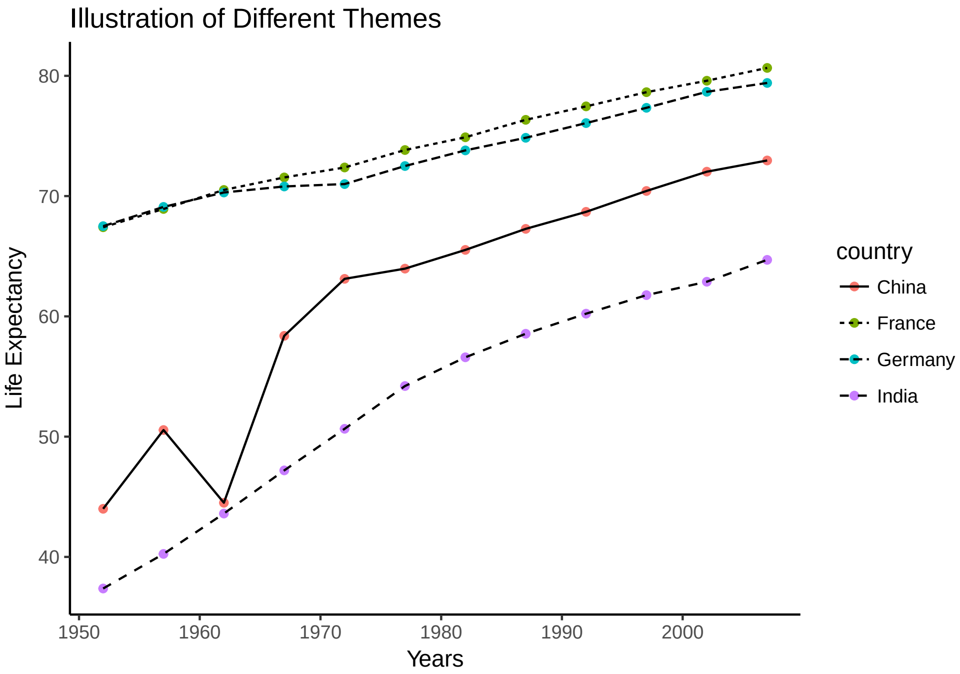


```
(plot_eur_box_flip <- ggplot(data = filter(data_gapminder,
                                          continent == "Europe"),
                         mapping = aes(x = year,
                                       y = lifeExp,
                                       color = country)) +
   geom_boxplot() +
   coord_flip()
   )
```

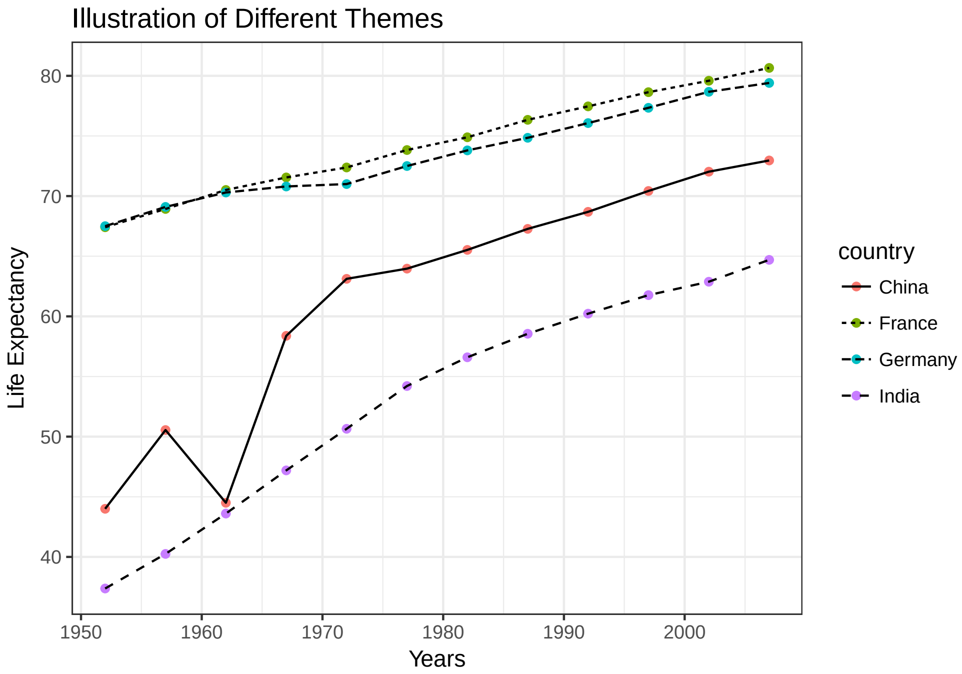


Themes

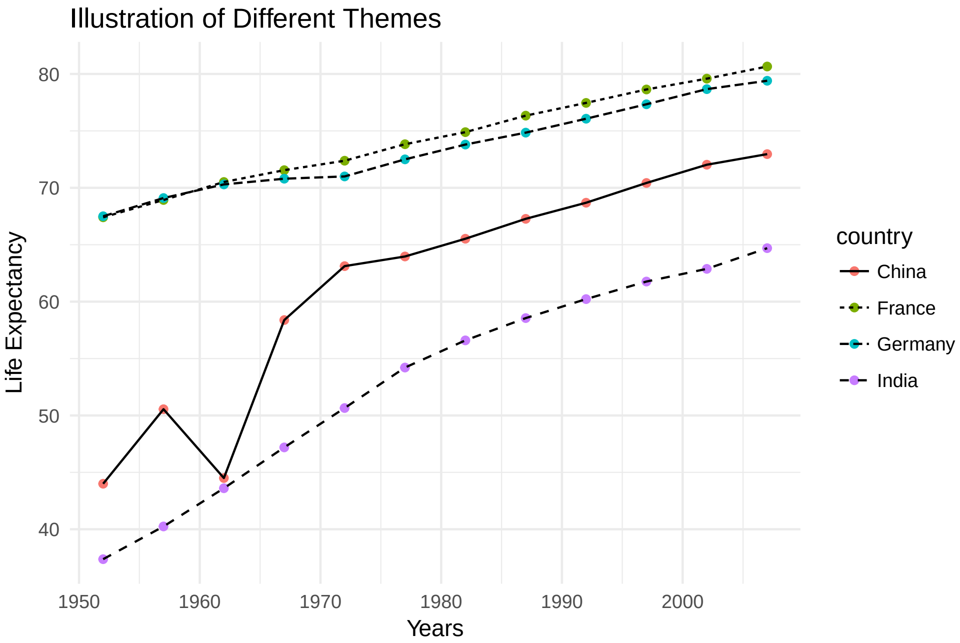
```
(plot_CIFG_theme <- ggplot(data = data_CIFG,
                                mapping = aes(x = year,
                                              y = lifeExp)) +
  geom_point(mapping =  aes(color = country)) +
  geom_line(mapping = aes(linetype = country)) +
   labs(x = "Years",
        y = "Life Expectancy",
        title = "Illustration of Different Themes") +
  theme_classic() #note the classic theme
 )
```



```
(plot_CIFG_theme <- ggplot(data = data_CIFG,
                                mapping = aes(x = year,
                                              y = lifeExp)) +
  geom_point(mapping =  aes(color = country)) +
  geom_line(mapping = aes(linetype = country)) +
   labs(x = "Years",
        y = "Life Expectancy",
        title = "Illustration of Different Themes") +
  theme_bw() #note the bw theme (black and white)
 )
```



```
(plot_CIFG_theme <- ggplot(data = data_CIFG,
                                mapping = aes(x = year,
                                              y = lifeExp)) +
  geom_point(mapping =  aes(color = country)) +
  geom_line(mapping = aes(linetype = country)) +
   labs(x = "Years",
        y = "Life Expectancy",
        caption = "Insert caption here",
        title = "Illustration of Different Themes") +
  theme_minimal() #note the minimal theme
 )
```



Insert caption here

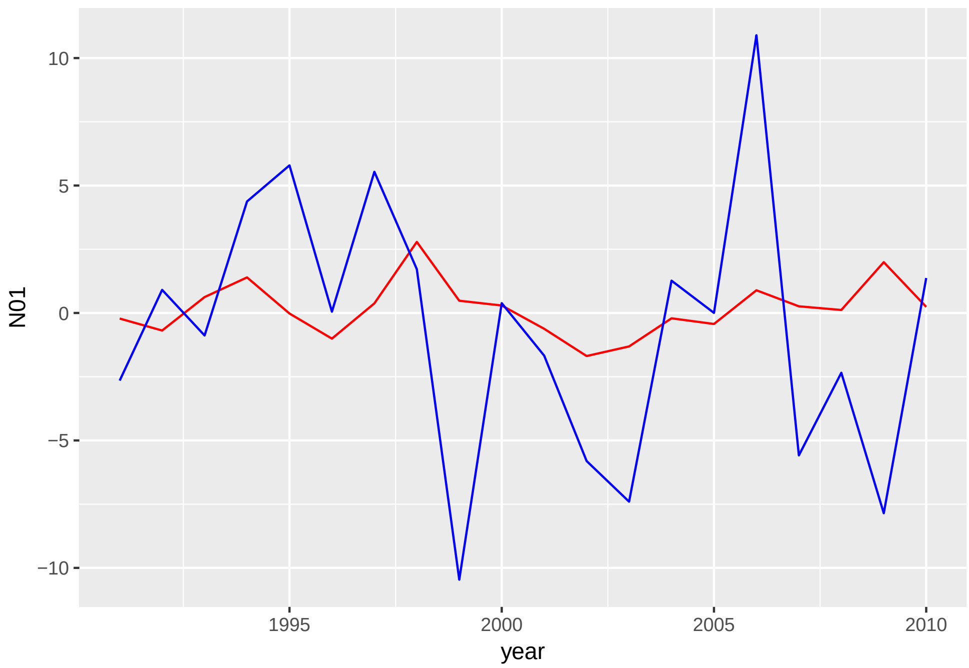
The Main FAQs

1. plotting multiple variables in one frame

Map Aesthetic To Different Columns The most frequently asked question goes something like this: I have two variables in my data.frame, and I'd like to plot them as separate points, with different color depending on which variable it is. How do I do that?

```r
temp <- cbind(1991:2010, rnorm(20, 0, 1), rnorm(20, -1, 5)) %>%
  tibble::as_tibble()
names(temp) <- c("year", "N01", "N25")

(ggplot(data = temp,
        mapping = aes(x = year)) +
    geom_line(mapping = aes(y = N01), color = "red") +
    geom_line(mapping = aes(y = N25), color = "blue")
  )
```



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.