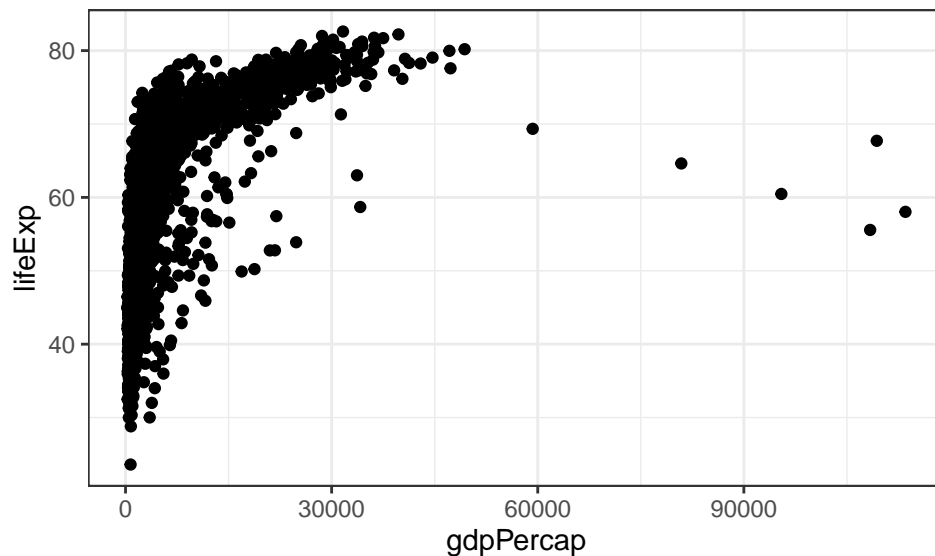


Data Visualization with R - Notes

Chapter 3 - Make plot

When we use the package `ggplot()`, we need to think as layers. First we have to build the base, tell `ggplot` which data we gonna use, then which *axis* and the type of the plot with `geom_...()` function.

```
ggplot(gapminder, aes(gdpPercap, lifeExp)) +  
  geom_point()
```



In this example I've used the function `geom_point()` in order to build a scatter plot.

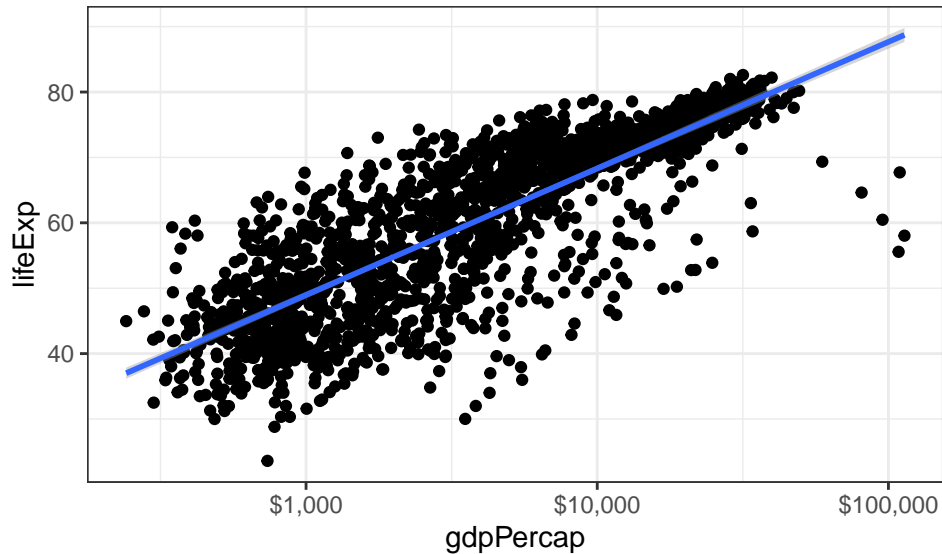
So... just to recap, we need to take the following steps in order to make a plot:

1. Tell the `ggplot()` what data we are using
2. Specify the axis using the function `aes()`
3. Add a layer to the plot using the function `geom_`, which means the type of the plot, like:
 - `geom_points()` -> scatter plot
 - `geom_bar()` -> bar plot

As we can saw in our previous plot... the data is bunched up against the left side of the plot. One way to “correct” this, in order to have a better view of the data, is to use the x axis as log, with the function `scale_x_log10()`

```
p <- ggplot(gapminder, aes(gdpPercap, lifeExp))

p + geom_point() +
  geom_smooth(method = "lm")+ # Add linear regression line
  scale_x_log10(labels = scales::dollar) # Change the x label to dollar
```



But... what if we want to distinguish between countries? It's easy, we just need to add the argument `color =` or `fill =` inside the `aes()` function. We use each of the arguments in the following situations:

- `color =` we use in line plot, scatter plot... It changes the color of the dots, lines and outlines.
- `fill =` we use in bar plots, histograms, density plot...

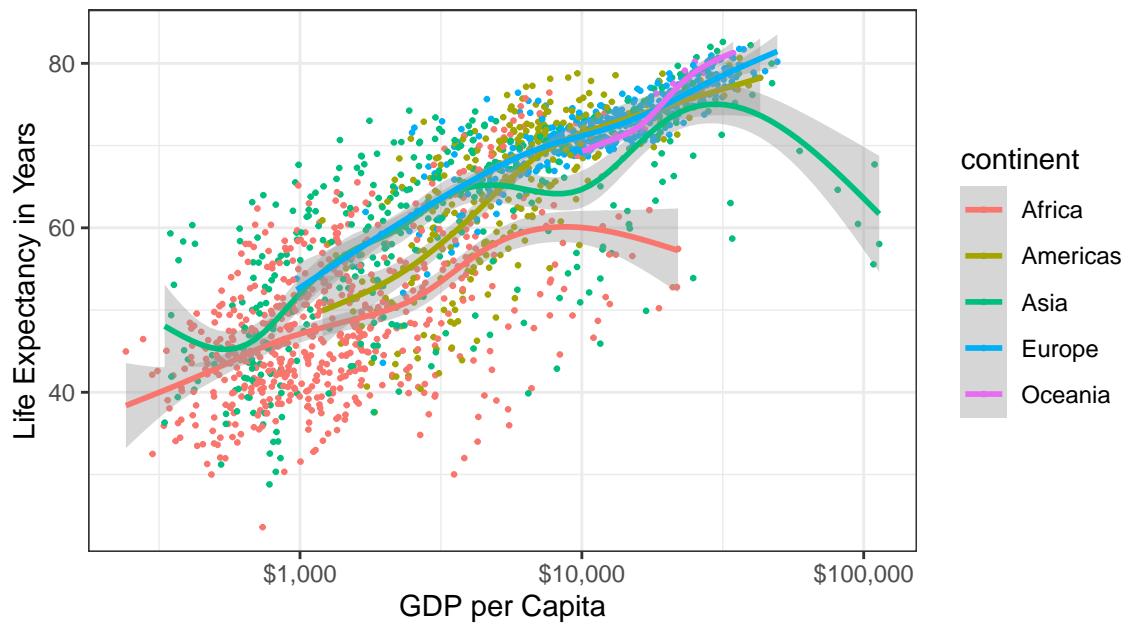
Another thing to do, is to change the plot labels with the function `labs()`.

```
p <- ggplot(gapminder, aes(gdpPercap, lifeExp, color = continent))

p + geom_point(size = 0.5)+ # Changing the dots size
  geom_smooth(method = "gam")+
  scale_x_log10(labels = scales::dollar)+
  labs(x = "GDP per Capita", y = "Life Expectancy in Years",
       title = "Economic Growth and Life Expectancy",
       subtitle = "data points are country-years",
       caption = "Source: Gapminder.")
```

Economic Growth and Life Expectancy

data points are country-years



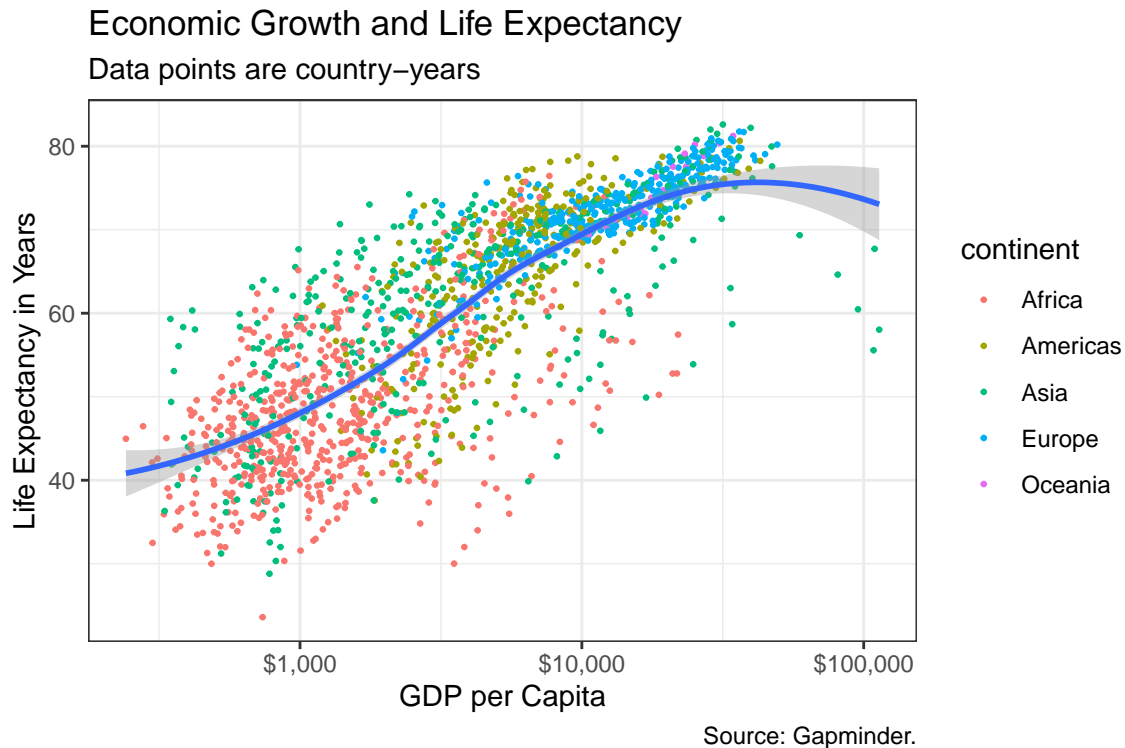
Source: Gapminder.

Aesthetics can be mapped per Geom

With too many smoothers, the plot feels a bit strange and it doesn't show us the correct message. Perhaps if we just use one line, it would be better. By default geoms inherit their mapping from `ggplot()` function. We can change this by specifying different aesthetics for each geom, using the same argument `mapping = aes(...)`.

```
p <- ggplot(gapminder, aes(gdpPercap, lifeExp))

p + geom_point(mapping = aes(color = continent), size = 0.5) +
  geom_smooth(method = "loess") +
  scale_x_log10(labels = scales::dollar) +
  labs(x = "GDP per Capita", y = "Life Expectancy in Years",
       title = "Economic Growth and Life Expectancy",
       subtitle = "Data points are country-years",
       caption = "Source: Gapminder.")
```



Chapter 4 - Show the Right Numbers

It's common to write things wrong when we are programming and we gotta keep in mind that code almost never works as we wish for the first time when we write it. In ggplot it is not different, errors in code can result in figures that don't look right, sometimes those errors can make data visualization beautiful wrong, as we can see the twitter @accidental__aR. Sometimes our code will not even produce a plot cuz some syntax error, we will forget some "+" sign between geom_ function or forget where u put the parenthesis, when it occurs... its almost as ggplot is complain with that u make a mistake and something is not right, other times the code will run and will produce a plot, but not as you expected.

In this chapter, we gonna pass through some features that commonly cause trouble in ggplot, which means how to the ggplot:

- About internal structure of the data (*grouping*)
- How to divide the data (*faceting*)
- How to calculate and summarize data before plotting (*transforming*)

Grouped Data and the "Group" Aesthetic