Fundamentals of R

Block 3 - Practical Visualizations

Henrique Sposito and Livio Muller-Silva

2022-10-14

R Markdown: another way to store code

Markdown is a simple formatting syntax for authoring HTML, PDF, and Word documents.

Creating an R Markdown document is just like an R script, you just have to click the new document button and select R Markdowm from the options.

Markdown allows you to mix chunks of code (in light grey) with actual text, and export a document out of it.

You can embed an R code chunk like this:

In the case above, we are just adjusting the setup for the document and loading some packages for our R Markdown document.

This is the best resource for information on R Markdown!

Some Basics:

Section headers work with #:

First-level header

Second-level header

Third-level header

For changing text styles use *:

Italics

Bold

Italics and bold

For inserting R code click on the **C** buttom above or use Cmd + Option + I on MAC (for Windows: Ctrl + Alt + I).

```
as.character("R Markdown is awesome")
```

```
## [1] "R Markdown is awesome"
```

Code chunks can be evaluated (run code?), included (should the code displayed in knitted document?), and much more. rmarkdown, as a tidyverse package, also has a cheat sheet!

When you click the **Knit** button a document in HTML or PDF can be generated that includes both content as well as the output of any embedded R code chunks within the document.

Lastly, R Markdown can be further used to create presentations in R (as the ones we use in class, see the xaringan package) or even to write your Master's thesis (check out iheidown).

Visualizations

##Setting up the Gap minder data

```r
gapminder <- gapminder::gapminder # create an object
summary(gapminder) # summary data
```

Before we start, the ggplot2 book is a great source for you to learn the details of visualizing with ggplot (and it was written using an RMarkdown!!).

Bar plots: Life expectancy from 1957 to 2007 across continents

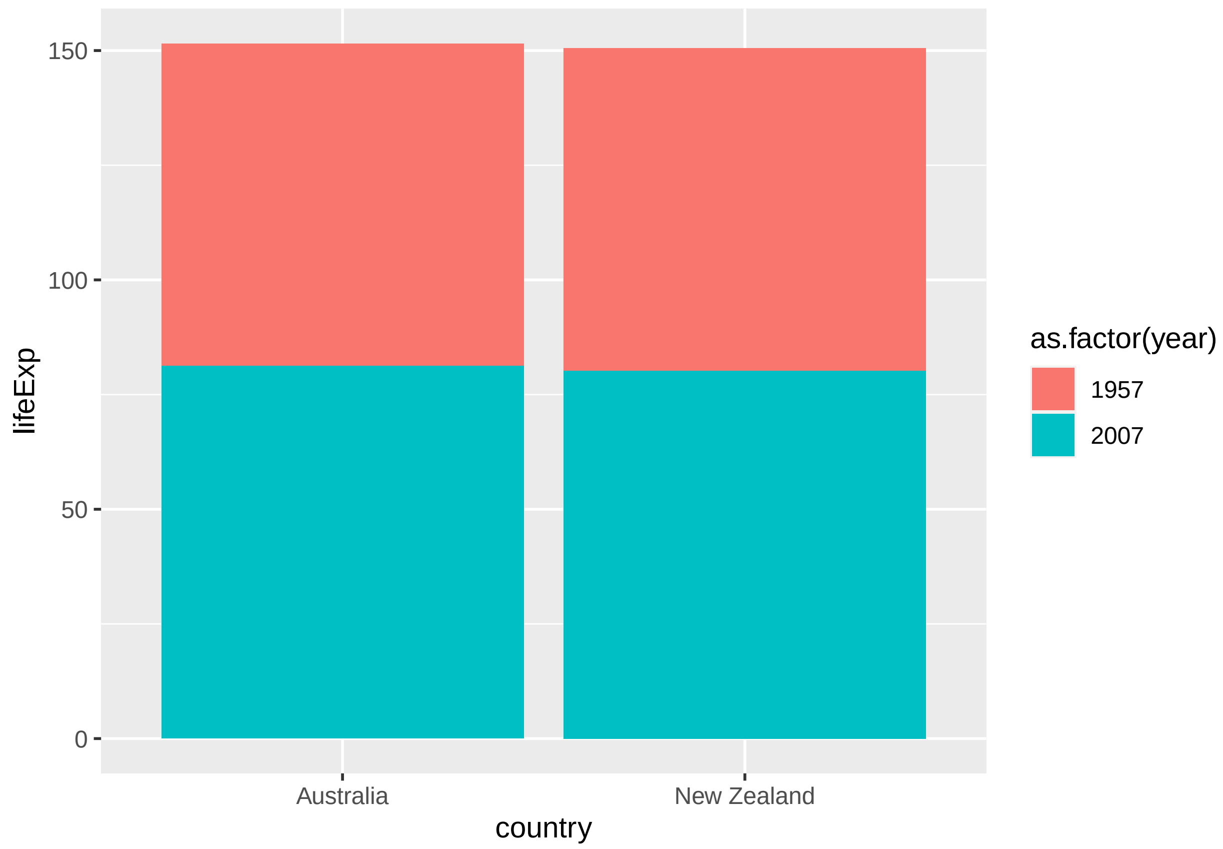
To create bar plot in ggplot2 we use the `geom_col()` function.

What are bar plots good for?

Let's start with a simple plot:

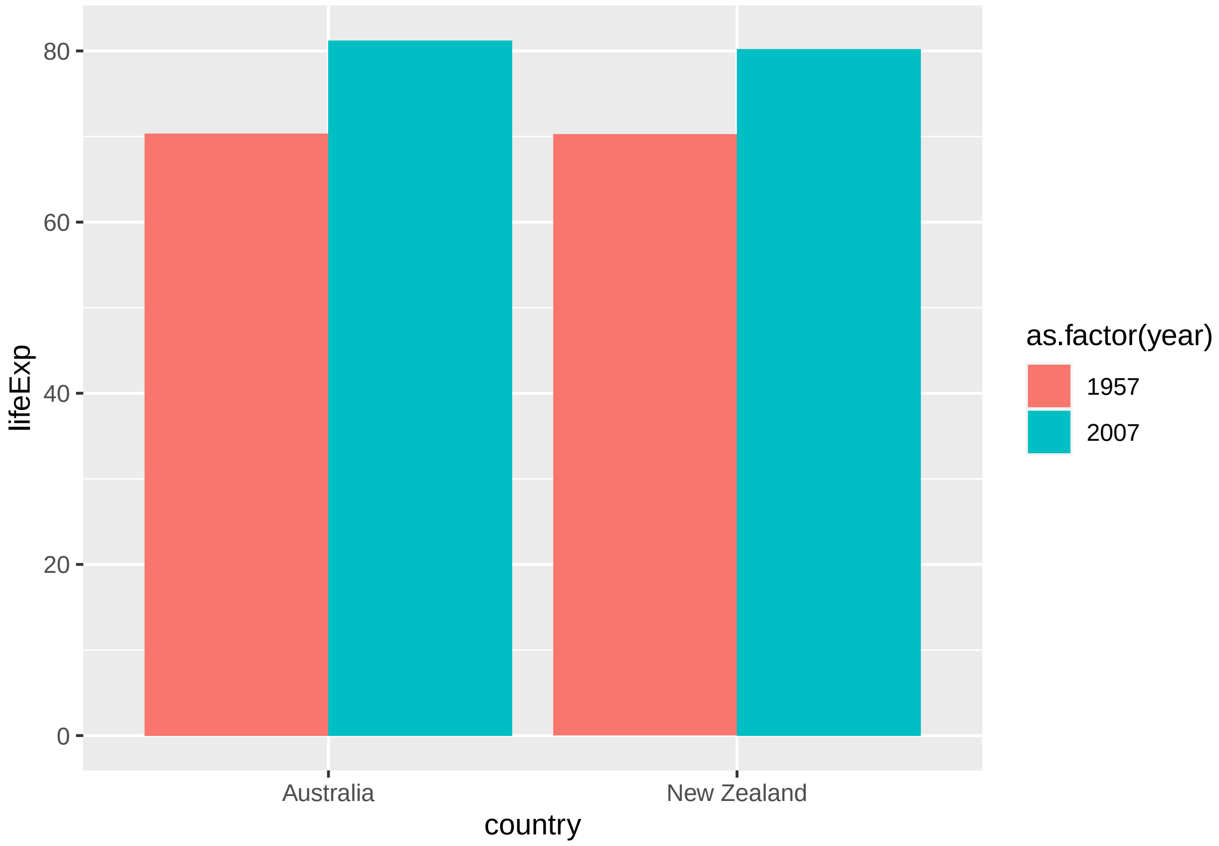
```r
# We could plot countries in one continent only at two different time points, no?

gapminder %>% # select data
  filter(year == 1957 | year == 2007,
         continent == "Oceania") %>% # filter for years of interest and Oceania
  ggplot(aes(x = country, # country in the x axis
             y = lifeExp, # map average life expectancy in the y axis
             fill = as.factor(year))) + # map fill by year (as factor), remember that fill means the co
  geom_col()
```



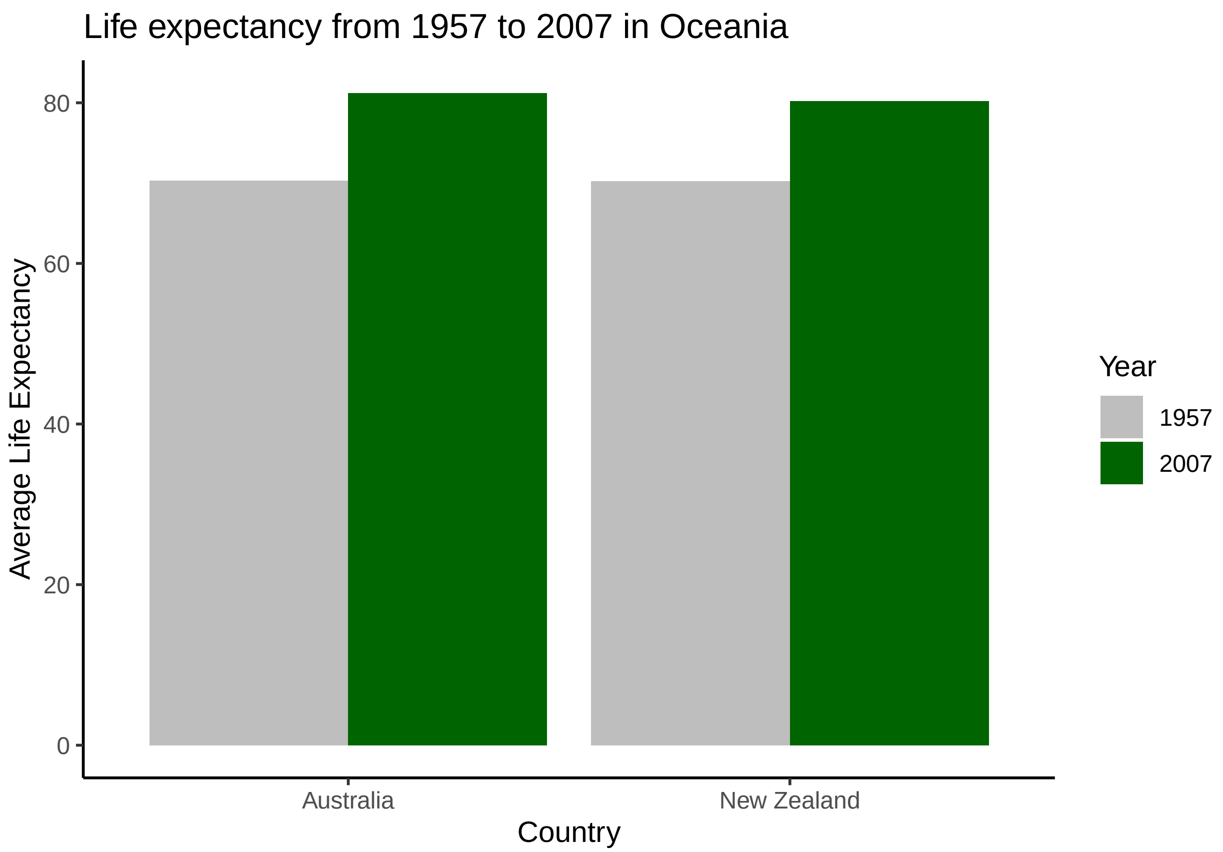
```
# What happened here?
```

```
gapminder %>%
  filter(year == 1957 | year == 2007,
         continent == "Oceania") %>%
  ggplot(aes(x = country,
             y = lifeExp,
             fill = as.factor(year))) +
  geom_col(position= "dodge") # separated the columns.
```



```r
# The plot looks nicer, but there are still some issues with this. What are they?
```

```r
gapminder %>%
  filter(year == 1957 | year == 2007,
         continent == "Oceania") %>%
  ggplot(aes(x = country,
             y = lifeExp,
             fill = as.factor(year))) +
  geom_col(position= "dodge")+ # defining the position of stat in bars
  labs(title = "Life expectancy from 1957 to 2007 in Oceania", # add a title
       x = "Country", # add a label for x axis
       y = "Average Life Expectancy", # add a label for y axis
       fill = "Year") + # sub legend for fill
  scale_fill_manual(values = c("gray", "darkgreen")) + # manually set colors
  theme_classic() # add a theme; theme_classic is a "clean" theme that removes many unncessary things.
```

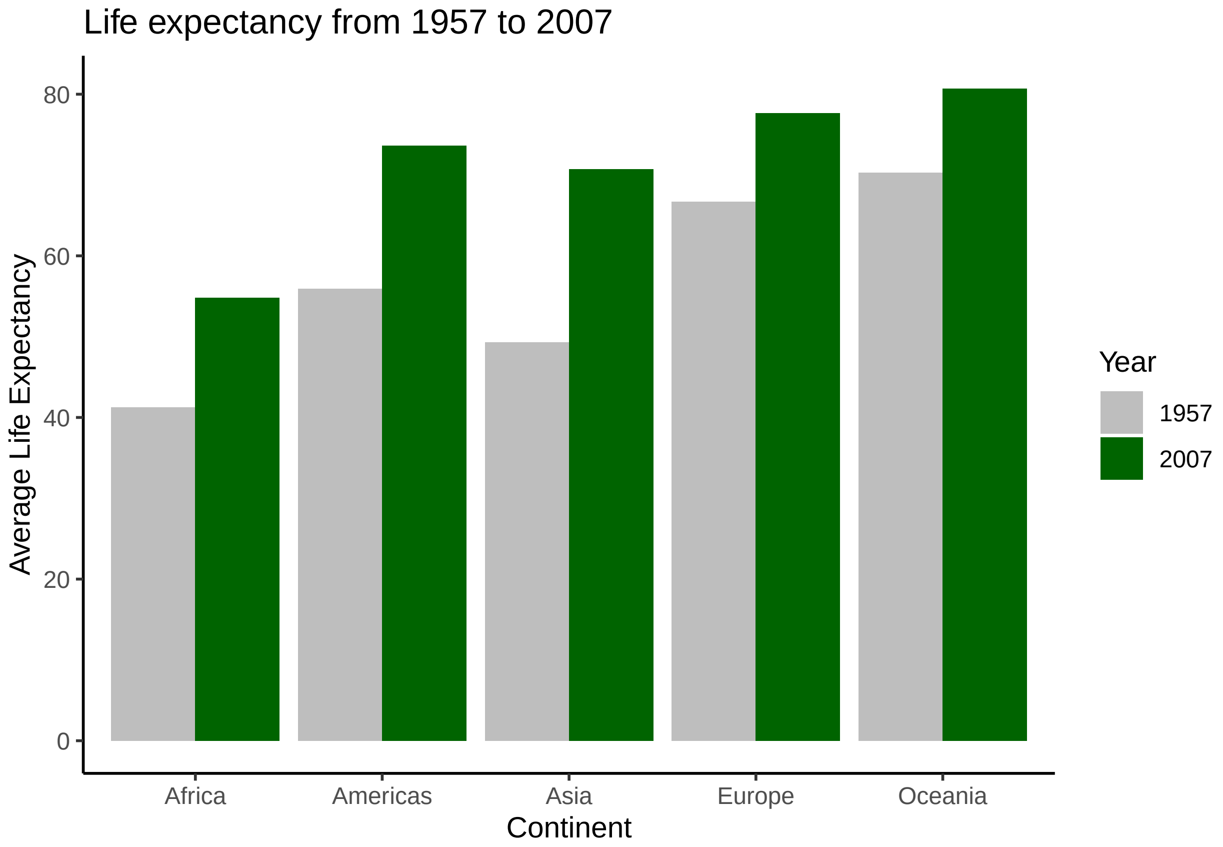


```
# The plot looks nicer, but there are still some issues with this. What are they?
```

Let's use bars to plot the average difference in life expectancy from 1957 to 2007 across continents.

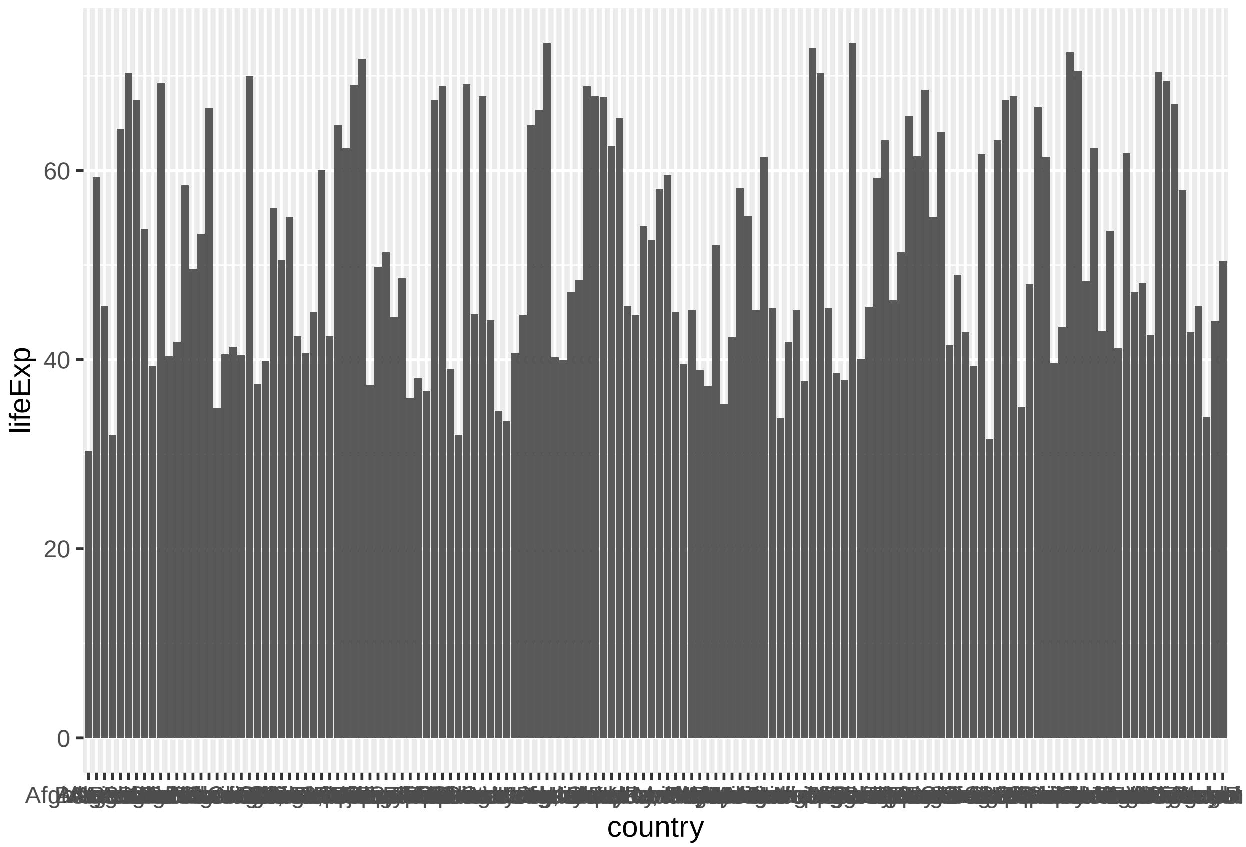
```
gapminder %>% # select data
  filter(year == 1957 | year == 2007) %>% # filter for years of interest
  group_by(continent, year) %>% # group by year and country
  summarise(Avg_life_expectancy = mean(lifeExp)) %>% # get the means by the groups
  ggplot(aes(x = continent, # map continent in the x axis
             y = Avg_life_expectancy, # map average life expectancy in the y axis
             fill = as.factor(year))) + # fill mapping by year
  geom_col(position = "dodge") + # defining the position of stat in bars
  labs(title = "Life expectancy from 1957 to 2007", # add a title
       x = "Continent", # add a label for x axis
       y = "Average Life Expectancy", # add a label for y axis
       fill = "Year") + # sub legend for fill
  scale_fill_manual(values = c("gray", "darkgreen")) + # manually set colors
  theme_classic() # add a theme
```

```
## `summarise()` has grouped output by 'continent'. You can override using the
## `.groups` argument.
```



Would a bar plot be a good choice to plot life expectancy by country in 1957?

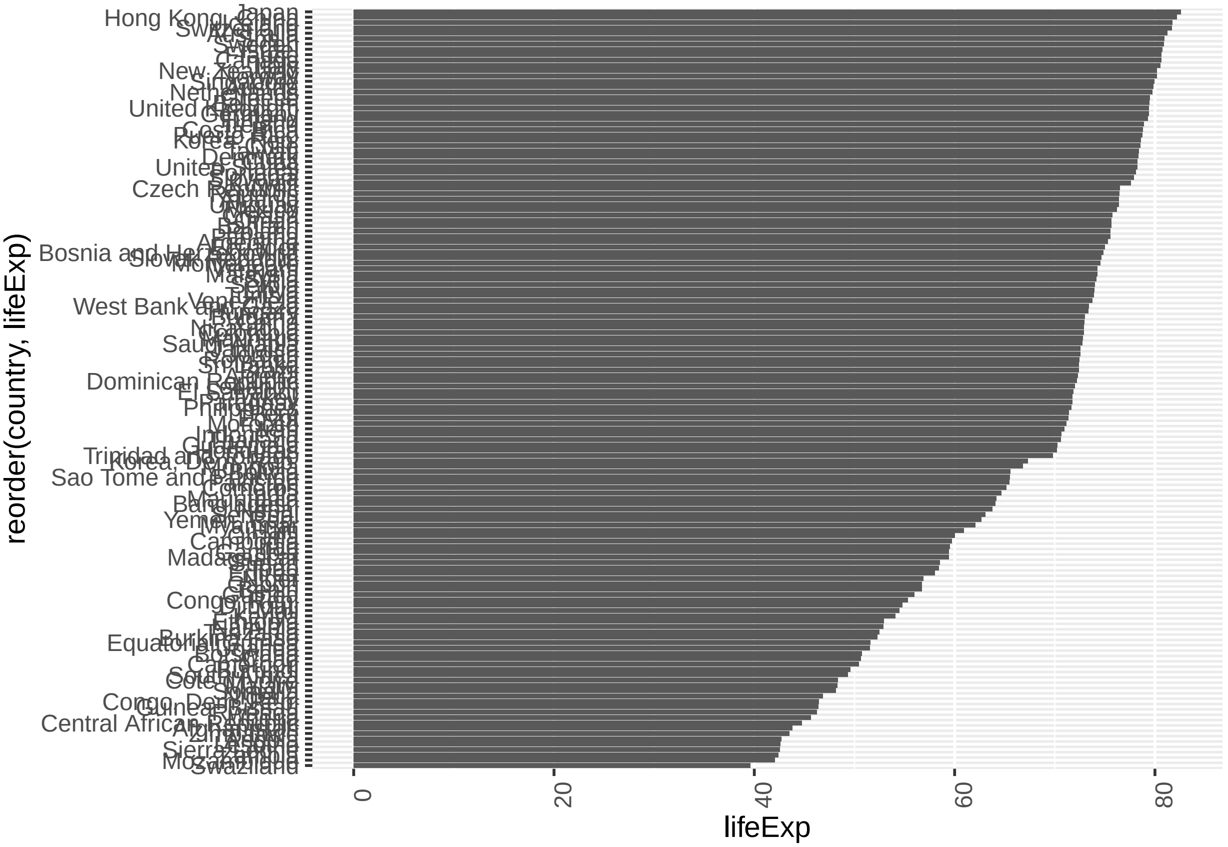
```
gapminder %>% # select data
  filter(year == 1957) %>% # filter for years of interest
  ggplot(aes(x = country, # map country in the x axis
             y = lifeExp)) + # map average life expectancy in the y axis
  geom_col()
```



This is clearly not a good plotting choice (too much information)…

```
# We could perhaps change the angles of the names.
# Do you think this would solve the issue?

# Change the angle of text
gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(x = reorder(country, lifeExp),
             y = lifeExp)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90))+ # change the angle for text in x axis
  coord_flip() # flip coordinates instead
```



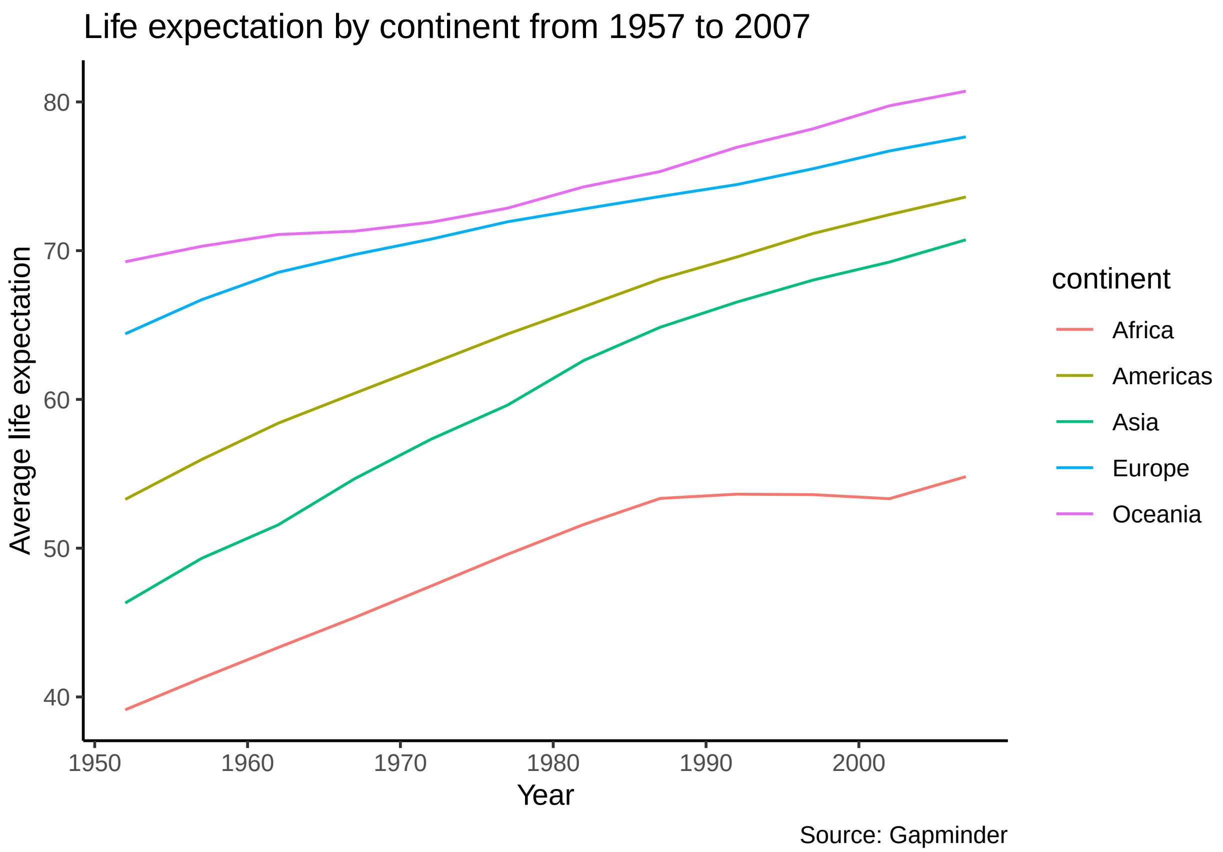
Line plots: The evolution of life expectancy

To create line plots in ggplot2 we use the `geom_line()` function.

What are line plots good for in this case?

```
gapminder %>%
  group_by(continent, year) %>% # group by year and country
  summarise(Avg_life_expectancy = mean(lifeExp)) %>%  #  mean life expectancy
  ggplot(aes(x = year, y = Avg_life_expectancy)) +
  geom_line(aes(color = continent)) + #here we are mapping color by continent at the geom_level
  labs(title = "Life expectation by continent from 1957 to 2007",
       x = "Year",
       y = "Average life expectation",
       caption = "Source: Gapminder") + # add caption
  theme_classic() # add theme
```

```
## ‘summarise()‘ has grouped output by ’continent’. You can override using the
## ‘.groups‘ argument.
```



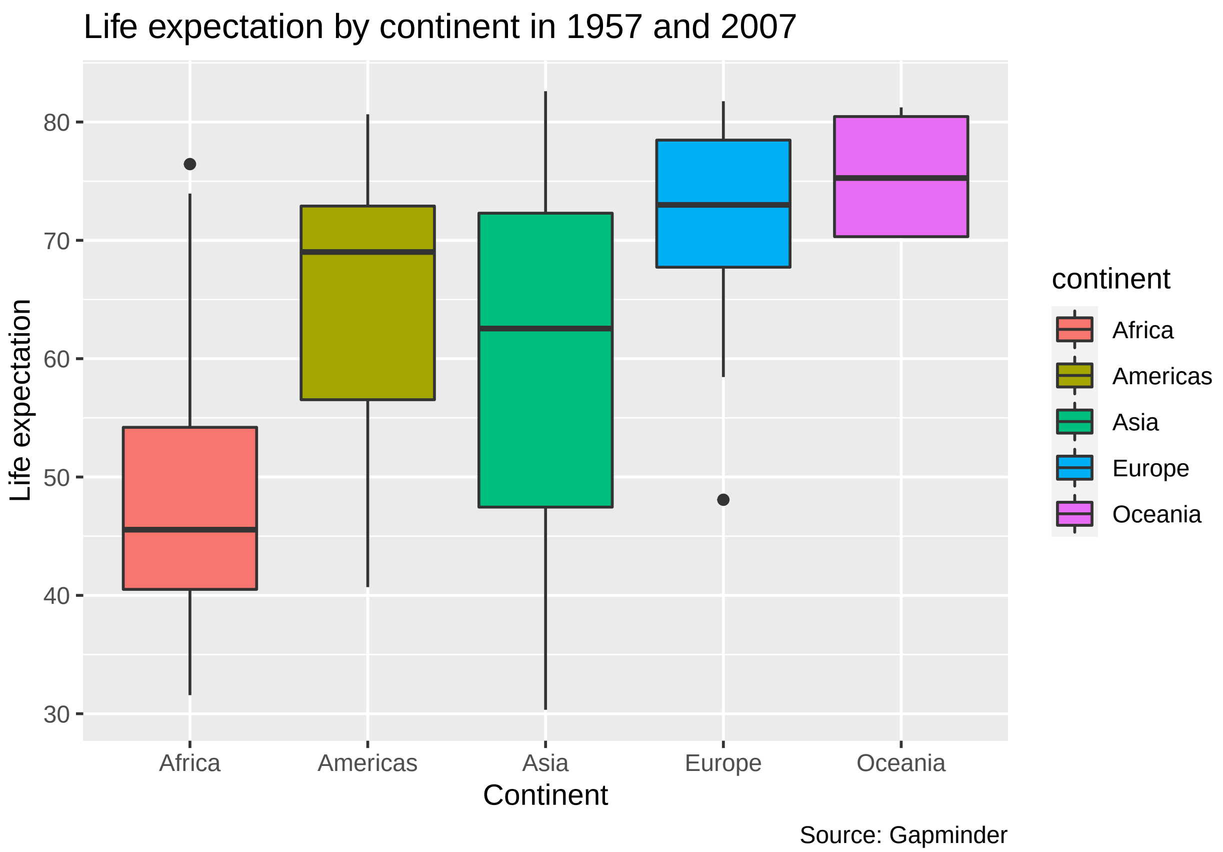
Could you make the same line plots for GDP per capita across continents in time?

Box plots: Distribution of life expectancies across continent

To create scatter plots in ggplot2 we use the `geom_boxplot()` function.

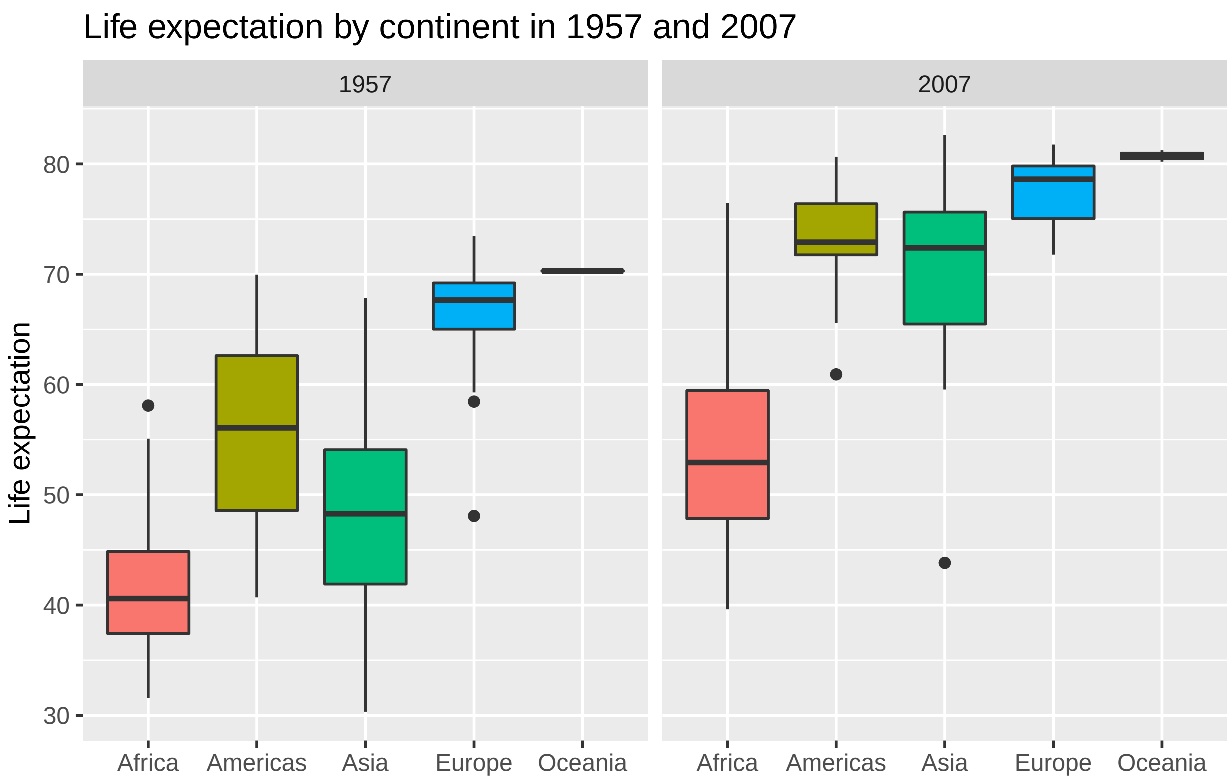
What are box plots good for in this case?

```
gapminder %>%
  filter(year == 1957 | year == 2007) %>%
  ggplot(aes(x = continent, y = lifeExp, fill = continent)) +
  geom_boxplot() +
  labs(title = "Life expectation by continent in 1957 and 2007",
       x = "Continent",
       y = "Life expectation",
       caption = "Source: Gapminder")
```



```
# what is happening here?
# which information is redundant?
```

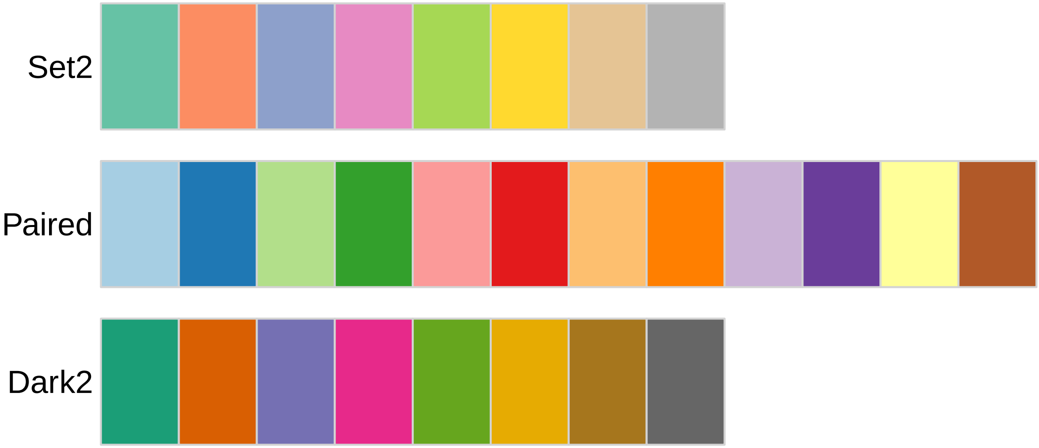
```
gapminder %>%
  filter(year == 1957 | year == 2007) %>%
  ggplot(aes(x = continent, y = lifeExp, fill = continent)) +
  geom_boxplot() +
  facet_wrap(~year)+ #facet_wrap() quickly separates our data in facets
  labs(title = "Life expectation by continent in 1957 and 2007",
       x = "", #continent was redundant, let's remove it
       y = "Life expectation",
       caption = "Source: Gapminder")+
  theme(legend.position="none") # legend on the side was redundant, let's remove it.
```



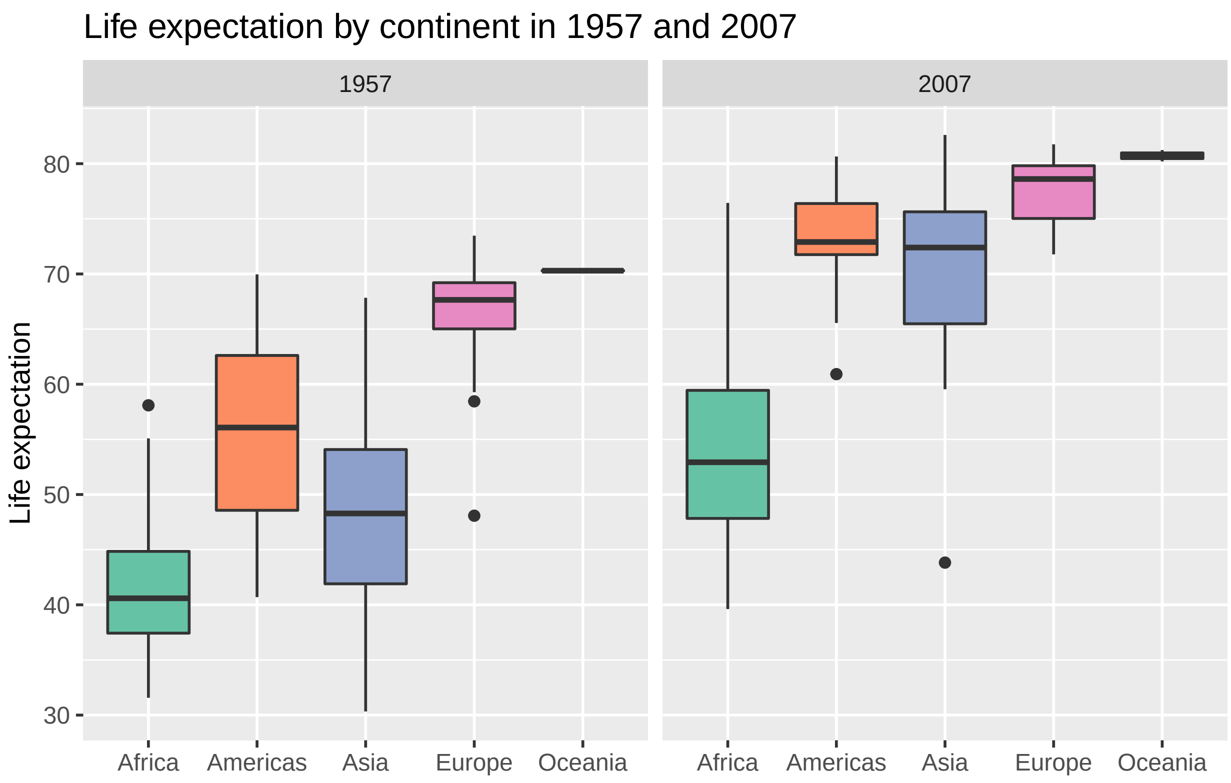
Source: Gapminder

```
#we still have a problem, these colors are not friendly. How can we use the RColorBrewer package from t
```

```
library(RColorBrewer)
display.brewer.all(colorblindFriendly = TRUE, type = "qual") #why type "qual"?
```



```
gapminder %>%
  filter(year == 1957 | year == 2007) %>%
  ggplot(aes(x = continent, y = lifeExp, fill = continent)) +
  geom_boxplot() +
  facet_wrap(~year)+
  labs(title = "Life expectation by continent in 1957 and 2007",
       x = "",
       y = "Life expectation",
       caption = "Source: Gapminder")+
  theme(legend.position="none")+
  scale_fill_brewer(palette="Set2")
```



Source: Gapminder

```
#scale_fill_brewer() colours the aesthetics fill with "Set2".
#scale_colour_brewer() colours the aesthetics color with the palette you indicate.
```

Can you make the same box plot for GDP?

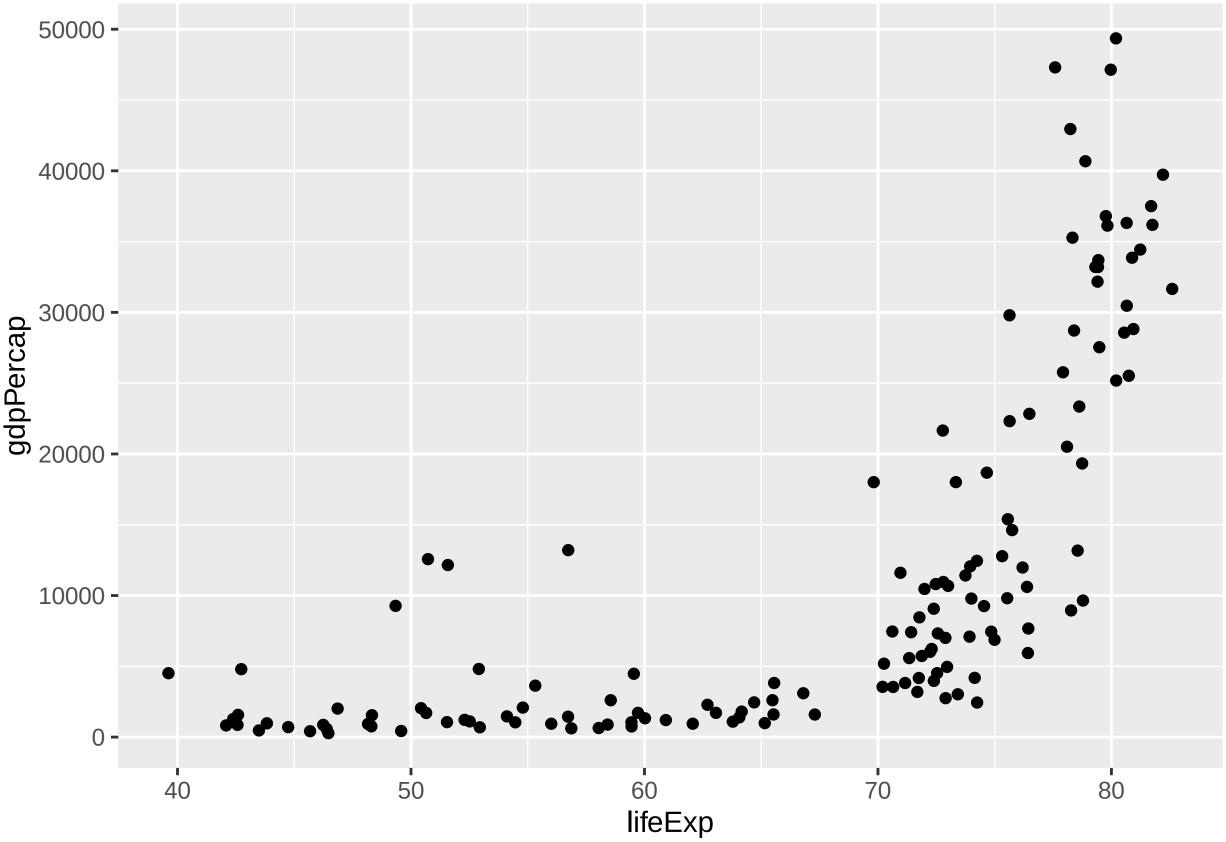
Scatter plots: Population, life expectancy and GDP

To create scatter plots in ggplot2 we use the `geom_point()` function.

What are scatter plots good for in this case?

Let's plot population and GDP per capita, in 2007!

```
gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(x = lifeExp,
             y = gdpPercap)) +
  geom_point() # add points to plot
```

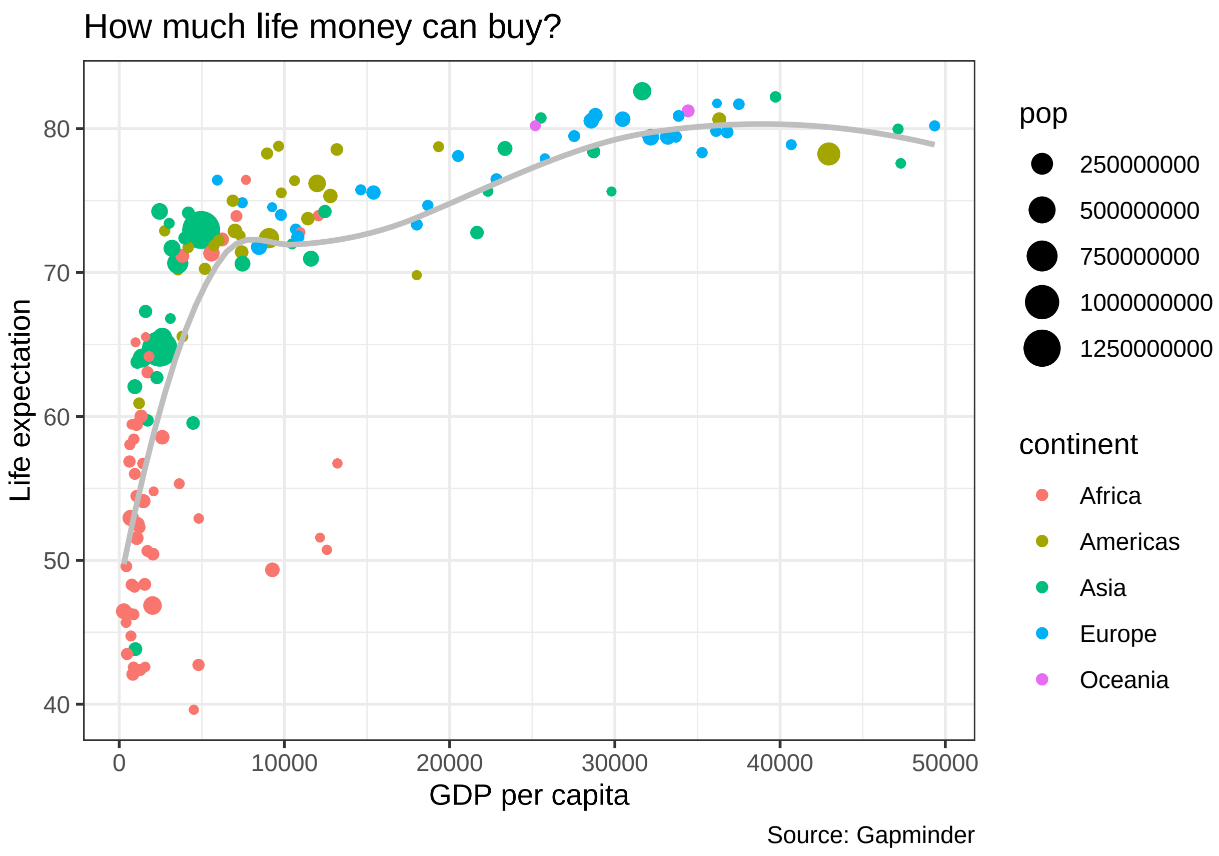


Is this plot informative? How could we improve this?

What if we focus on population, life expectation and GDP?

```
gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes(size = pop, color = continent)) + # color all points blue
  geom_smooth(se = FALSE, color = "gray") + # add a smoothed line
  labs(title = "How much life money can buy?",
       x = "GDP per capita",
       y = "Life expectation",
       caption = "Source: Gapminder") + # add caption
  theme_bw() # add theme
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



Source: Gapminder