Data Visualisation

Bigabwamukama Lawrence

Table of contents

Task 1: Import

The required packages for this homework exercise have already been added.

1. Run the code chunk with the label 'load-packages' to load the required packages. Tipp: Click on the green play button in the top right corner of the code chunk.

```
library(gapminder)
library(ggplot2)
library(dplyr)
library(readr)
library(sf)
library(rnaturalearth)
```

Task 2: Transform for data in 2007

Fill in the gaps

1. A code chunk has already been created below.
2. Start with the `gapminder` object and add the pipe operator at the end of the line.
3. On a new line use the `filter()` function to narrow down the data for observation of the year 2007.
4. Use the assignment operator to assign the data to an object named `gapminder_2007`.
5. Run the code contained in the code chunk and fix any errors.
6. Next to the code chunk option `#| eval:` change the value from `false` to `true`.
7. Render the document and fix any errors.

```
gapminder_2007 <- gapminder |>
  filter (year == 2007)
```

Task 3: Summarize data for life expectancy by continent

Fill in the gaps

1. A code chunk has already been created.
2. Start with the `gapminder_2007` object and add the pipe operator at the end of the line.
3. On a new line use the `group_by()` function to group the operations that follow by continent. Add the pipe operator at the end of the line.
4. On a new line use the `summarise()` function to calculate the number of observations (count) and median life expectancy.
5. Use the assignment operator to assign the data to an object named `gapminder_summary_2007`.

6. Run the code contained in the code chunk and fix any errors.
7. Next to the code chunk option `#| eval:` change the value from `false` to `true`.
8. Render the document and fix any errors.

```
gapminder_summary_2007 <- gapminder_2007 |>
  group_by( continent) |>
  summarise(
    count = n(),
    lifeExp = median(lifeExp)
  )
```

Task 4: Summarize data for life expectancy by continent and year

Fill in the gaps

1. A code chunk has already been created.
2. Start with the `gapminder` object and add the pipe operator at the end of the line.
3. On a new line use the `group_by()` function to group the operations that follow by continent and year. Add the pipe operator at the end of the line.
4. On a new line use the `summarise()` function to calculate and median life expectancy.
5. Use the assignment operator to assign the data to an object named `gapminder_summary_continent_year` .
6. Run the code contained in the code chunk and fix any errors.
7. Next to the code chunk option `#| eval:` change the value from `false` to `true`.
8. Render the document and fix any errors.

```
gapminder_summary_continent_year <- gapminder |>
  group_by(continent, year) |>
  summarise(
    lifeExp = median(lifeExp)
    )
```

Task 5: Data visualization

Thank you for working through the previous tasks. We are convinced that you have done a great job, but because the task descriptions aren't always unambiguous, we have imported the data that we would have expected to be created and stored in the objects `gapminder_2007`, `gapminder_summary_2007` and `gapminder_summary_continent_year` at the previous code chunks. This is to ensure that you can work through the following tasks.

1. Run the code contained in the code chunk below to import the data.

```
gapminder_2007 <- read_rds(here::here("data/gapminder-2007.rds"))

gapminder_summary_2007 <- read_rds(here::here("data/gapminder-summary-2007.rds"))

gapminder_summary_continent_year <- read_rds(here::here("data/gapminder-summary-continent-
```

Task 6: Create a boxplot

1. Add a new code chunk below point 5.
2. Use the `ggplot()` function and the `gapminder_2007` object to create a boxplot with the following aesthetic mappings:

- continent to the x-axis;
- life expectancy to the y-axis;
- continent to color using the `fill = continent` argument inside `aes()`

3. Do not display (ignore) the outliers in the plot. **Note: Use a search engine or an AI tool to find the solution and add the link to the solution you have found.**

   Answer

   The link is https://www.perplexity.ai/search/r-label-loaddata-jdZ0GD7JQiGgdST9RyvQiQ?s=c

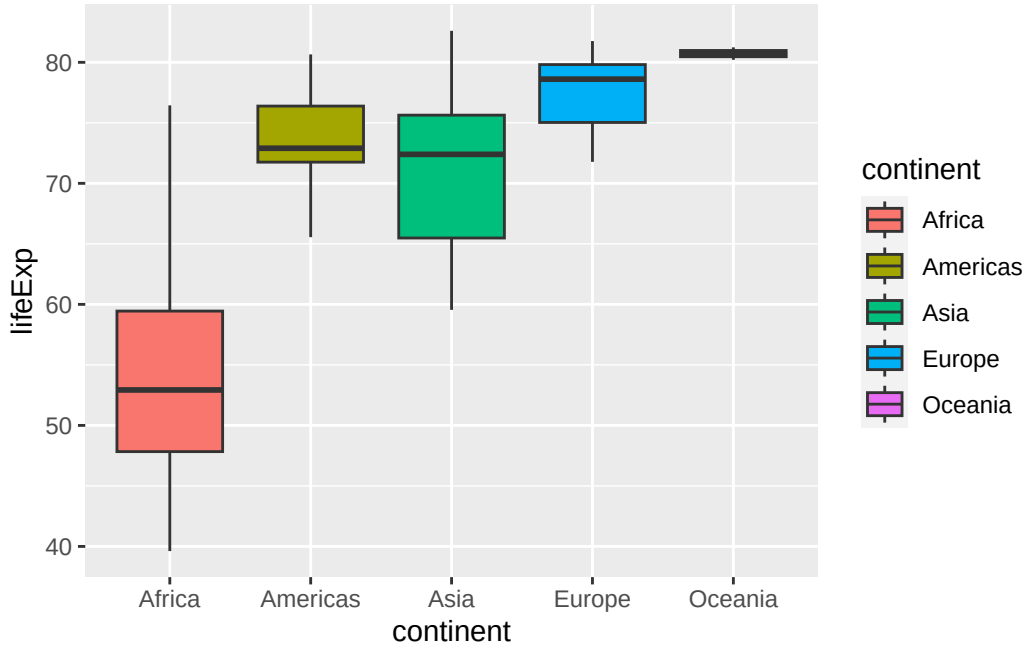
4. Run the code contained in the code chunk and fix any errors.
5. What are the data types of the three variables used for aesthetic mappings?

   Answer

   the three data types are type factor (continent and fill) and type numeric (lifeExp)

```
ggplot(data = gapminder_2007,
       mapping = aes(x = continent,
```

```
                    y = lifeExp,
                    fill= continent))+
  geom_boxplot(outlier.shape = NA)
```



Task 7: Create a timeseries plot

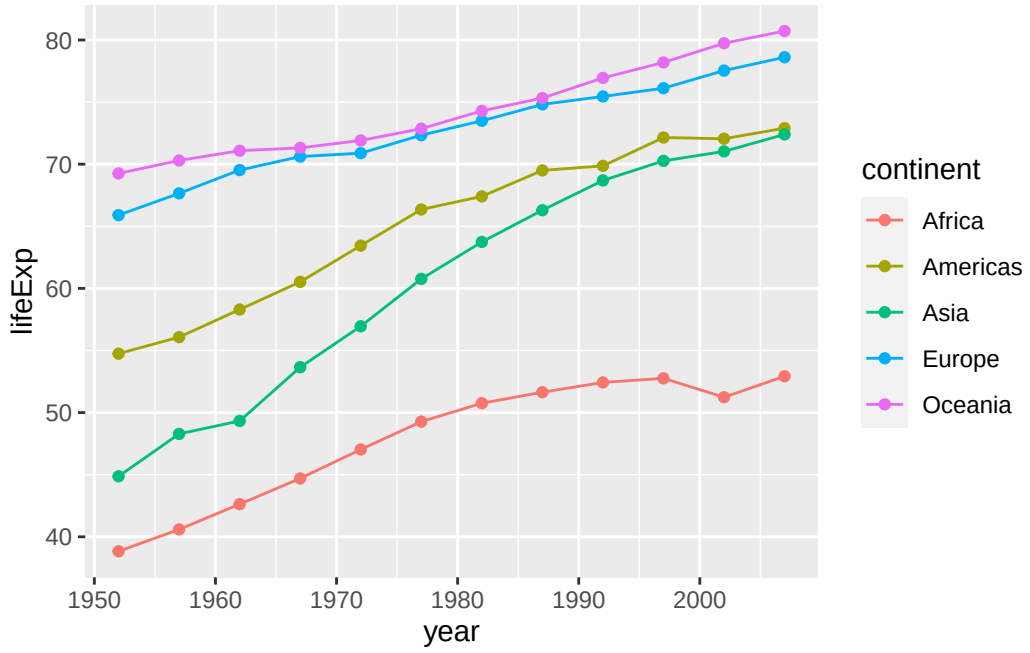
1. Add a new code chunk below.
2. Use the `ggplot()` function and the `gapminder_summary_continent_year` object to create a connected scatterplot (also called timeseries plot) using the `geom_line()` and `geom_point()`functions with the following aesthetic mappings:

- year to the x-axis;
- life expectancy to the y-axis;
- continent to color using the `color = continent` argument inside `aes()`

3. Run the code contained in the code chunk and fix any errors.

```
ggplot(data = gapminder_summary_continent_year,
       mapping = aes(x = year,
                     y = lifeExp,
                     color = continent))+
  geom_line() +
```

```
  geom_point()
```



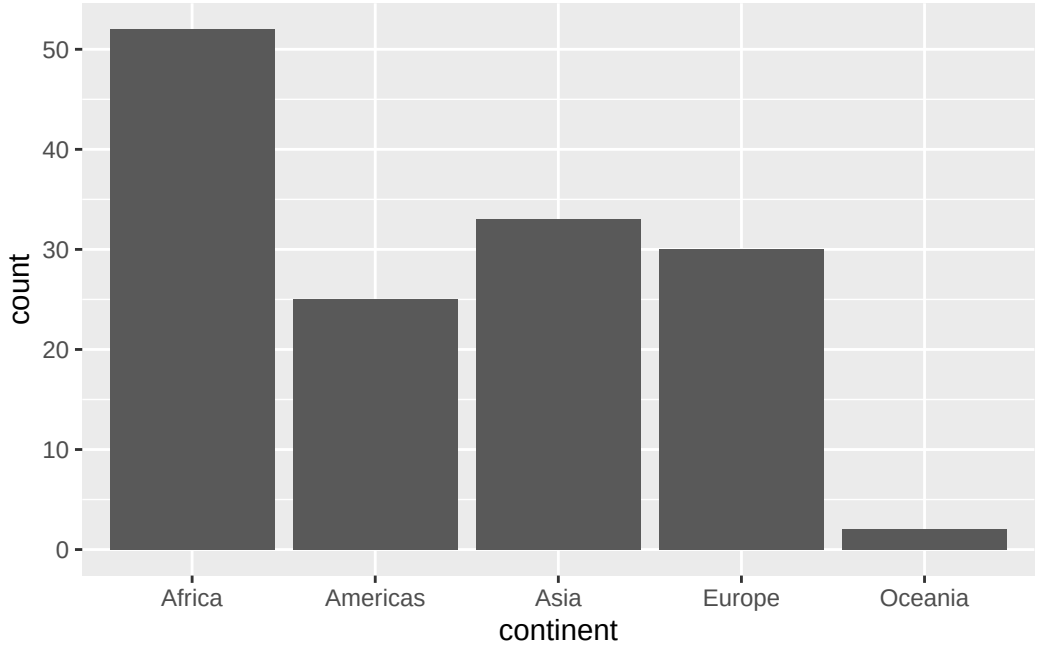
Task 8: Create a barplot

with `geom_col()`

1. Add a new code chunk below.
2. Use the `ggplot()` function and the `gapminder_summary_2007` object to create a barplot using the `geom_col()` function with the following aesthetic mappings:

- continent to the x-axis;
- count to the y-axis;

```
ggplot(data = gapminder_summary_2007,
       mapping = aes(x = continent,
                     y = count))+
  geom_col()
```

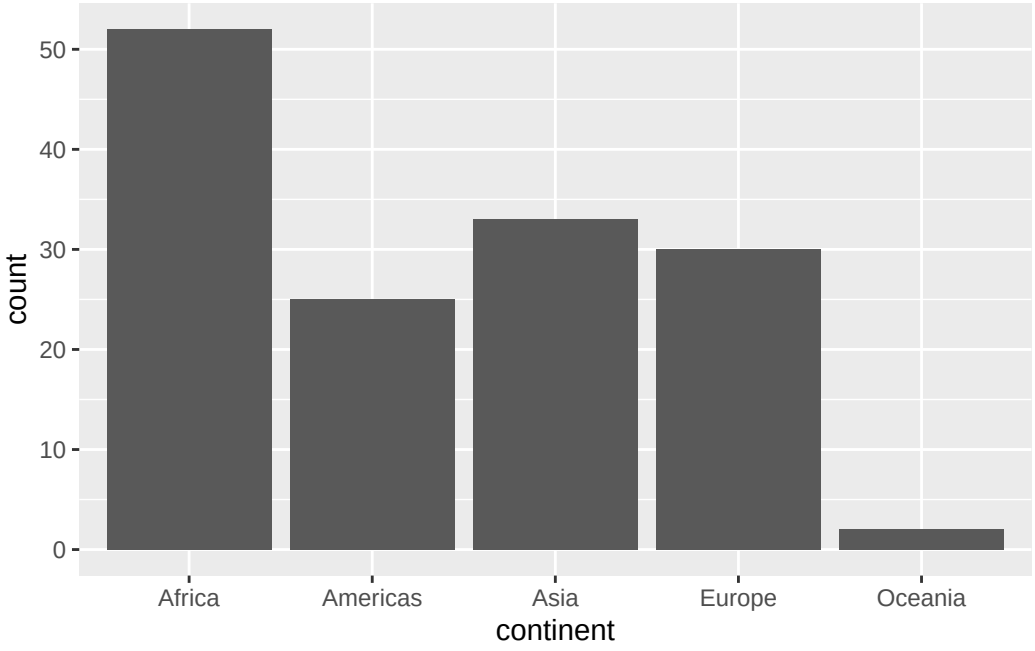


3. Run the code contained in the code chunk and fix any errors.

with `geom_bar()`

1. Add a new code chunk below.

```
ggplot(data = gapminder_2007,
       mapping = aes(x = continent,
                     ))+
  geom_bar()
```



2. Use the `ggplot()` function and the `gapminder_2007` object to create a barplot using the `geom_bar()` function with the following aesthetic mappings:

- continent to the x-axis;

3. Run the code contained in the code chunk and fix any errors.

4. The plot is identical to the plot created with `geom_col()`. Why? What does the `geom_bar()` function do? Write your text here:

   Answer

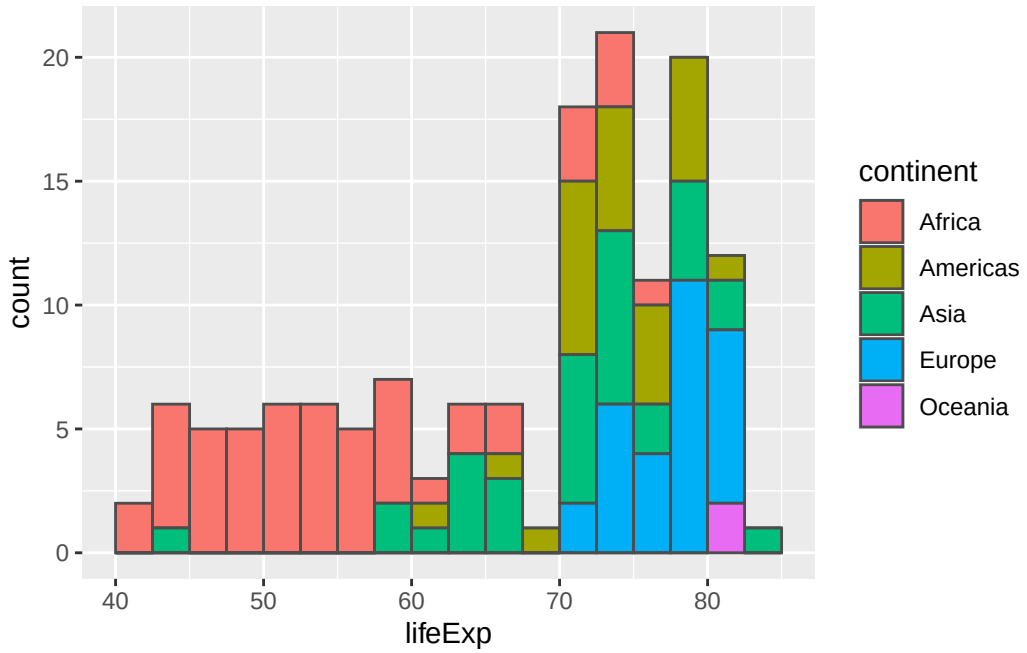
   The **`geom_bar()`** and **`geom_col()`** functions in ggplot2 are used to create bar charts. The main difference between these two functions is the default value of the **`stat`** argument. The **`geom_bar()`** function uses **`stat="count"`** by default, which means that it counts the number of occurrences of each category in the data and plots the counts as the height of the bars. On the other hand, the **`geom_col()`** function uses **`stat="identity"`** by default, which means that it uses the values in the data as the height of the bars.

Task 9: Create a histogram

1. Add a new code chunk below.

```
ggplot(data = gapminder_2007,
       mapping = aes(x = lifeExp,
                     fill = continent
                     ))+
  geom_histogram( col = "grey30",breaks = seq(40, 85, 2.5)

 )
```



2. Use the `ggplot()` function and the `gapminder_2007` object to create a histogram using the `geom_histogram()` function with the following aesthetic mappings:

- life expectancy to the x-axis;
- continent to color using the `fill = continent` argument inside `aes()`

3. Run the code contained in the code chunk and fix any errors.

4. Inside the `geom_histogram()` function, add the following arguments and values:

- `col = "grey30"`
- `breaks = seq(40, 85, 2.5)`

5. Run the code contained in the code chunk and fix any errors.

6. Describe how the `geom_histogram()` function is similar to the `geom_bar()` function.

   Answer

   `geom_histogram()` and **`geom_bar()`** are similar in that they both create bar charts using the **`geom`** layer in ggplot2. They also both use the **`stat`**layer to calculate the counts or densities of observations in each category or bin.

7. What happens by adding the 'breaks' argument? Play around with the numbers inside of `seq()` to see what changes. Describe here what you observe:

   Answer

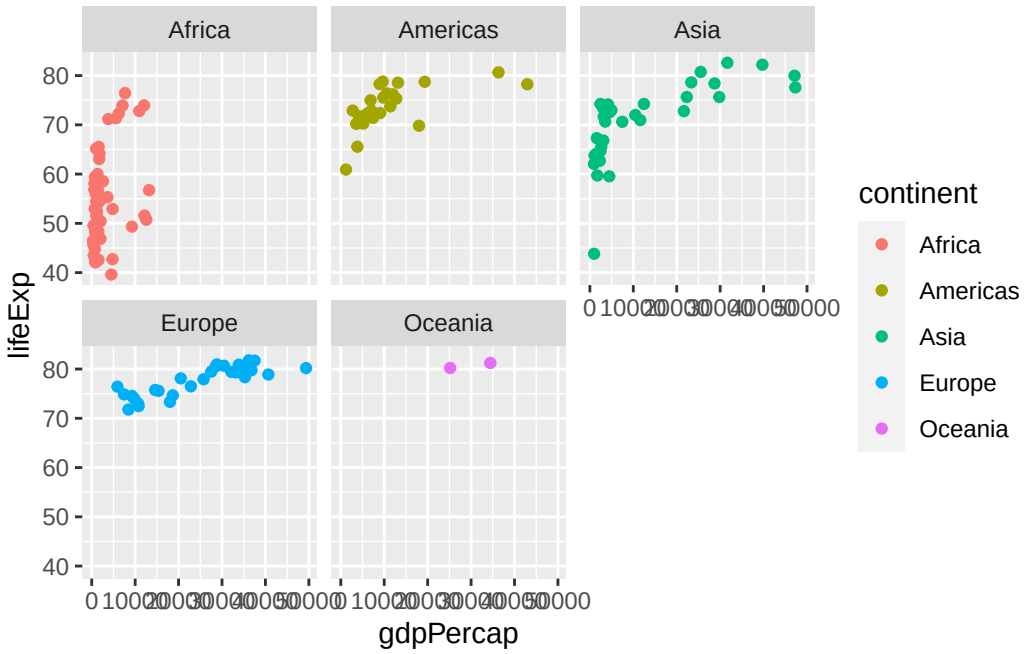
   Changing the numbers inside the **`seq()`** function would change the number and position of the breaks on the x-axis.

Task 10: Scatterplot and faceting

1. Add a new code chunk below.

```
ggplot(data = gapminder_2007,
       mapping = aes(x = gdpPercap,
                     y = lifeExp,

                     color = continent)) +
  geom_point() +
facet_wrap(~continent)
```



2. Use the `ggplot()` function and assign `gapminder_2007` and create a scatterplot using the `geom_point()`function with the following aesthetic mappings:

- gdpPercap the x-axis;
- lifeExp to the y-axis;
- population to the size argument;

  Note: this brought and error and is not included

- country to color using the `color = continent` argument inside `aes()`

4. Run the code contained in the code chunk and fix any errors.
5. Use the variable continent to facet the plot by adding: `facet_wrap(~continent)`.
6. Run the code contained in the code chunk and fix any errors.

Task 11: Create a lineplot and use facets

1. A code chunk with complete code has already been prepared.
2. Run the code contained in the code chunk and fix any errors.
3. Remove the '#' sign at the line that starts with the `scale_color_manual()` function

4. What is stored in the `country_colors` object? Find out by executing the object in the Console (type it to the Console and hit enter).

   Answer

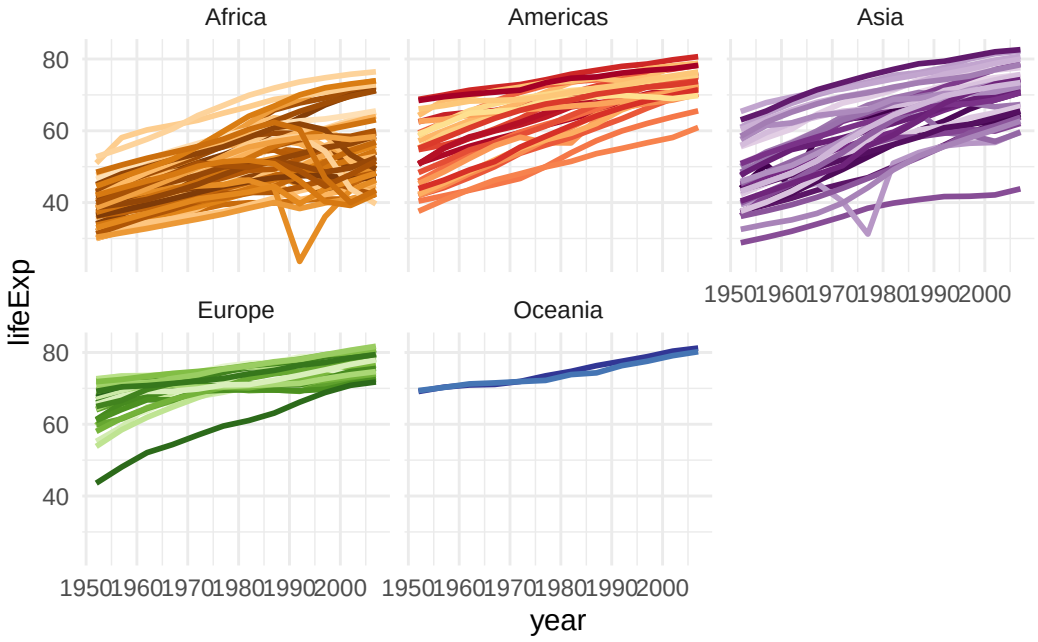
   It shows a list of various countries and their color codes for different colours in the console.

5. Do the same again, but with a question mark `?country_colors`. it shows help about the country_colors

6. Next to the code chunk option `#| eval:` change the value from `false` to `true`.

7. Render the document and fix any errors.

```
ggplot(data = gapminder,
       mapping = aes(x = year,
                     y = lifeExp,
                     group = country,
                     color = country)) +
  geom_line(lwd = 1, show.legend = FALSE) +
  facet_wrap(~continent) +
  scale_color_manual(values = country_colors) +
  theme_minimal()
```



Task 12: Create a choropleth Maps

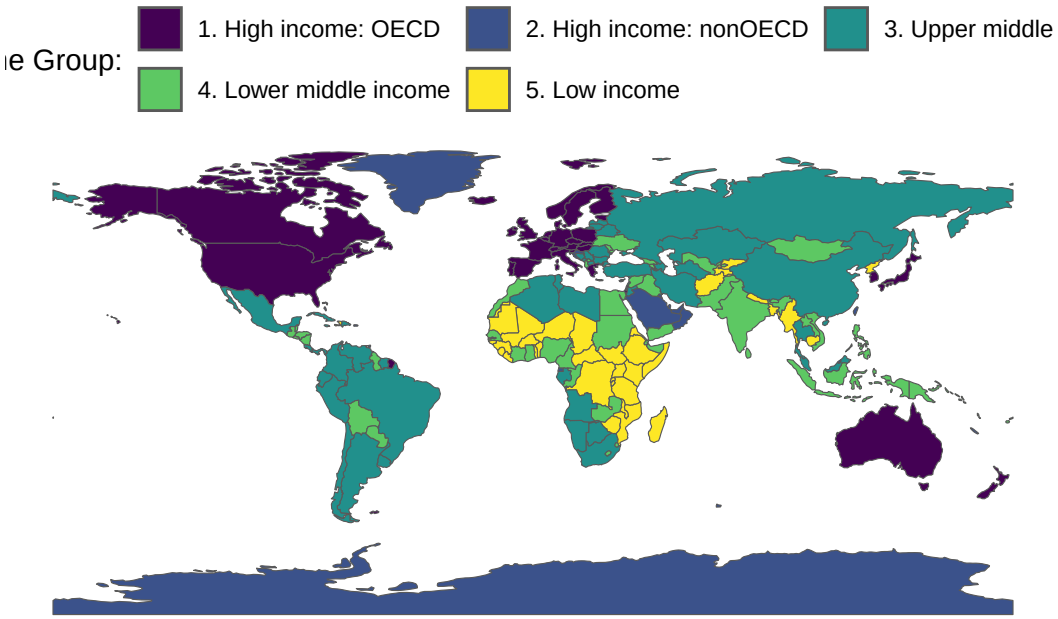
You can also prepare maps with `ggplot2`. It's beyond the scope of the class to teach you the foundations of spatial data in R, but a popular package to work with spatial data is the `sf` (Simple Features) R Package. The `rnaturalearth` R Package facilitates world mapping by making Natural Earth map data more easily available to R users.

The code chunk below contains code for a world map that shows countries by income group. To view the map, do the following:

1. Run the code contained in the code chunk and fix any errors.
2. Next to the code chunk option `#| eval:` change the value from `false` to `true`.
3. Render the document and fix any errors.

```
world <- ne_countries(scale = "small", returnclass = "sf")

world |>
  mutate(income_grp = factor(income_grp, ordered = T)) |>
  ggplot(aes(fill = income_grp)) +
  geom_sf() +
  theme_void() +
  theme(legend.position = "top") +
  labs(fill = "Income Group:") +
  guides(fill = guide_legend(nrow = 2, byrow = TRUE))
```



The code for the code chunk is taken from here: More here: https://bookdown.org/alhdzsz/data_viz_ir/maps.ht

Working with spatial data in R

If you are interested in working with spatial data in R, then we recommend the following resources for further study:

- Geocompuation with R - Book: https://geocompr.robinlovelace.net/
- Simple Features for R - Article: https://r-spatial.github.io/sf/articles/sf1.html
- tmap: thematic maps in R - R Package: https://r-tmap.github.io/tmap/

Task 13: Data communication

In the YAML header (between the three dashes at the top of the document)

1. Add your name as the author of this document
2. Render the document and fix any errors

Task 14: Complete assignment

Stage, Commit & Push to GitHub

1. Open the Git pane in RStudio. It's in the top right corner in a separate tab.
2. **Stage** your changes by checking appropriate box next to all files (if you select one file with your mouse, you can then highlight them all with Ctrl + A on your keyboard and check all boxes)
3. Write a meaningful commit message (e.g. "Completed homework assignment 02.) in the **Commit message** box
4. Click **Commit**. Note that every commit needs to have a commit message associated with it.

Open an issue on GitHub

Once you have ensured that the Quarto document renders without errors and you have pushed all your changes to GitHub, you can complete the assignment by opening an issue on

1. Open github.com in your browser.
2. Navigate to the GitHub organisation for the course.
3. Find the repository md-02-assignments that ends with your GitHub username.
4. Click on the "Issues" tab.
5. Click on the green "New issue" button.

6. In the “Title” field write: “Completed module 2 assignments”.
7. In the “Leave a comment” field, tag the course instructors @larnsce @mianzg @sskorik01 and ask some questions, if you like.