The final project - Animated changes of GDP PPP 1990-2018

Bartosz Czernecki

1/14/2020

Overall aim

The animated charts are increasingly interested in many social media podcasts. For example, the popularity among programming languages is shown in this link: https://www.youtube.com/watch?v=Og847HVwRSI

In the final project we will try to create an animated chart that will show us changes of Growth Domestic Product (GDP) standardized by Purchasing Power Paritity (PPP) in all countries that participate in our course. The values are given in USD.

Dataset

The World Bank publishes GDP PPP reports on annual basis (https://data.worldbank.org/indicator/NY.GDP.PCAP.PP.CD) for all countries around the world. This dataset has been pre-formated to the form readable in spreadsheet-like file.

Please download the file from http://iqdata.pl/dataprocessing/data/ and save it in a known location on your laptop.

Reading dataset

Set your working directory to the location of downloaded file. Please use from the upper menu `SESSION` then `SET WORKING DIRECTORY` and `CHOOSE DIRECTORY`.

Activate library that allows reading "XLSX" format (e.g. `readxl`) and name the read data as `df` object. The structure of the created `df` object should be as shown below for the first 6 rows:

```
## # A tibble: 6 x 31
##   country code  `1990` `1991` `1992` `1993` `1994` `1995` `1996` `1997`
##   <chr>   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Aruba   ABW   24101. 25871. 26533. 27431. 28657. 28649. 28499. 30216.
## 2 Afghan... AFG      NA     NA     NA     NA     NA     NA     NA     NA
## 3 Angola  AGO    3090.  3120.  2908.  2191.  2196.  2496.  2795.  2953.
## 4 Albania ALB    2549.  1909.  1823.  2057.  2290.  2666.  2980.  2717.
## 5 Andorra AND      NA     NA     NA     NA     NA     NA     NA     NA
## 6 Arab W... ARB    6808.  6872.  7255.  7459.  7646.  7774.  8094.  8398.
## # ... with 21 more variables: `1998` <dbl>, `1999` <dbl>, `2000` <dbl>,
## #   `2001` <dbl>, `2002` <dbl>, `2003` <dbl>, `2004` <dbl>, `2005` <dbl>,
## #   `2006` <dbl>, `2007` <dbl>, `2008` <dbl>, `2009` <dbl>, `2010` <dbl>,
## #   `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>, `2015` <dbl>,
## #   `2016` <dbl>, `2017` <dbl>, `2018` <dbl>
```

Re-shaping data

In the final phase of the project we will use `ggplot2` package to create our charts. This package requires so called narrow format of the data frame. Therefore our data frame needs to be reshaped to the form where we will have 4-5 columns with: `country`, `code`, `year` and `gdp`.

It can be done *manually* or with the use of a pivot-like functions. One of my favourie is the gather function from the tidyr package, which probably will have to be installed on your laptop if not used before.

Activate the tidyr and dplyr (or tidyverse) libraries and launch the ?gather command to familiarize yourself with the way it works. Try to reshape our df object to the following structure:

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
head(df2)
```

```
## # A tibble: 6 x 4
##   country     code  year     gdp
##   <chr>       <chr> <chr>  <dbl>
## 1 Aruba       ABW   1990  24101.
## 2 Afghanistan AFG   1990     NA 
## 3 Angola      AGO   1990   3090.
## 4 Albania     ALB   1990   2549.
## 5 Andorra     AND   1990     NA 
## 6 Arab World  ARB   1990   6808.
```

Hints: Use the year column as the **key** argument, and value = "gdp". Save the obtained results as df2.

Choosing countries

Right now our data frame df2 contains data for all countries. We'd like to limit our animation only to countries that participate in our class, i.e.: Czech Rep., Germany, Poland South Korea and Spain. Additionally, we'd like to add average for the entire world which is given as *WLD* in the column code.

Find out codes for the chosen countries and use the filter command to clip the dataset for these 6 codes. Fill in the following command and save results as gdp_tidy

```
# gdp_tidy = filter(df2, code %in% c("COUNTRY_1", "COUNTRY_2", ...) )
```

The first 10 rows of our gdp_tidy should return:

```
## # A tibble: 10 x 4
##    country        code  year     gdp
##    <chr>          <chr> <chr>  <dbl>
##  1 Czech Republic CZE   1990  12660.
##  2 Germany        DEU   1990  19497.
##  3 Spain          ESP   1990  13664.
##  4 Korea, Rep.    KOR   1990   8273.
##  5 Poland         POL   1990   4450 
##  6 World          WLD   1990   5498.
##  7 Czech Republic CZE   1991  11596.
##  8 Germany        DEU   1991  21032.
##  9 Spain          ESP   1991  14449.
## 10 Korea, Rep.    KOR   1991   9346.
```

Creating test data for a single year

In the next steps we will be trying to create a bar chart showing differences in GDP PPP / capita. For testing purposes we can clip our dataset to the last year (2018) available in the `gdp_tidy` object. Use again the `filter` command for column `year` where it is equal to 2018. Save the results as `test`.

```
## # A tibble: 6 x 4
##   country        code  year     gdp
##   <chr>          <chr> <chr>  <dbl>
## 1 Czech Republic CZE   2018  39744.
## 2 Germany        DEU   2018  53075.
## 3 Spain          ESP   2018  39715.
## 4 Korea, Rep.    KOR   2018  40112.
## 5 Poland         POL   2018  31337.
## 6 World          WLD   2018  17948.
```

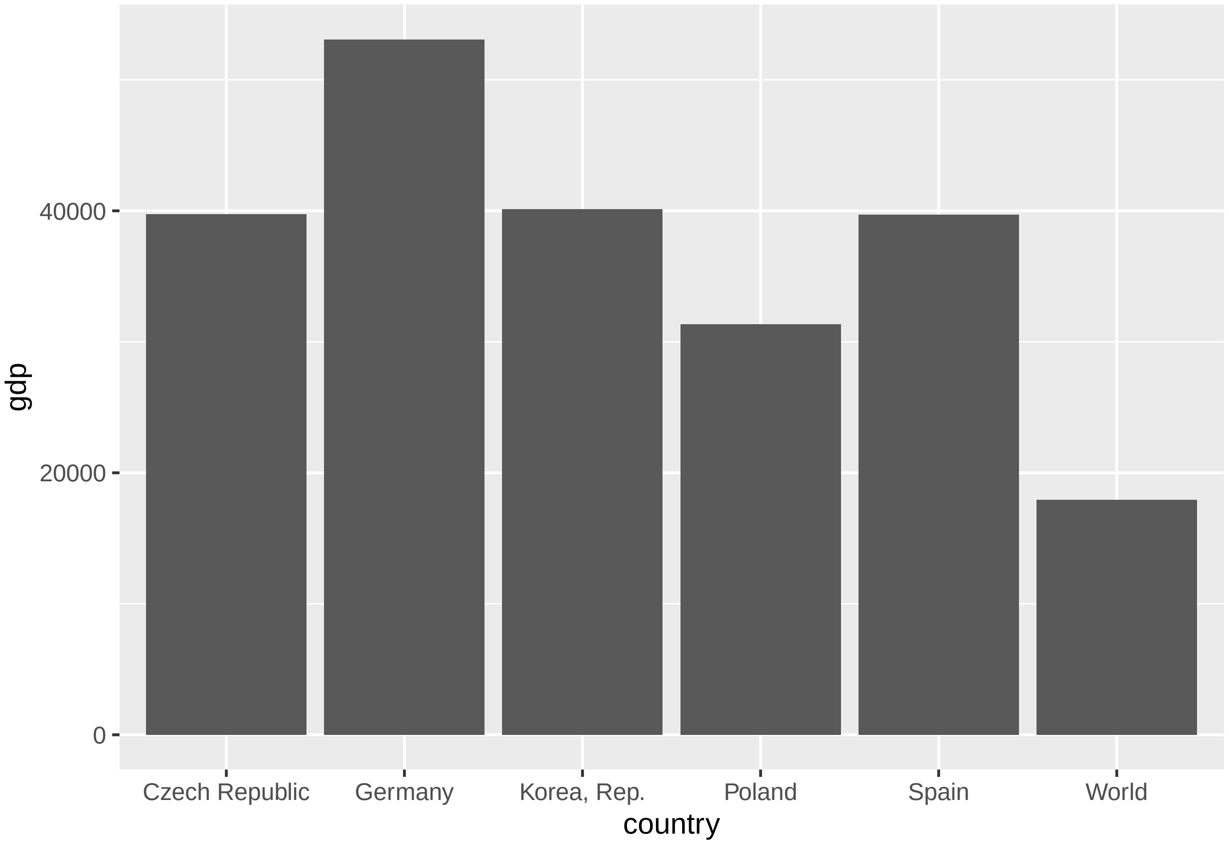
Plotting data with `ggplot2` library

Intro

Activate the `ggplot2` package and try to create a basic plot with any of the chosen geom_. The list of available geoms can be found in a cheat-sheet (https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf). Let's start with *geom_bar* as it is the most generic geom_ type for data structured as our `test` data frame.

Fill in the following blank fields in a below attached pseudocode to create a chart that will be similar to our plot shown in a figure below:

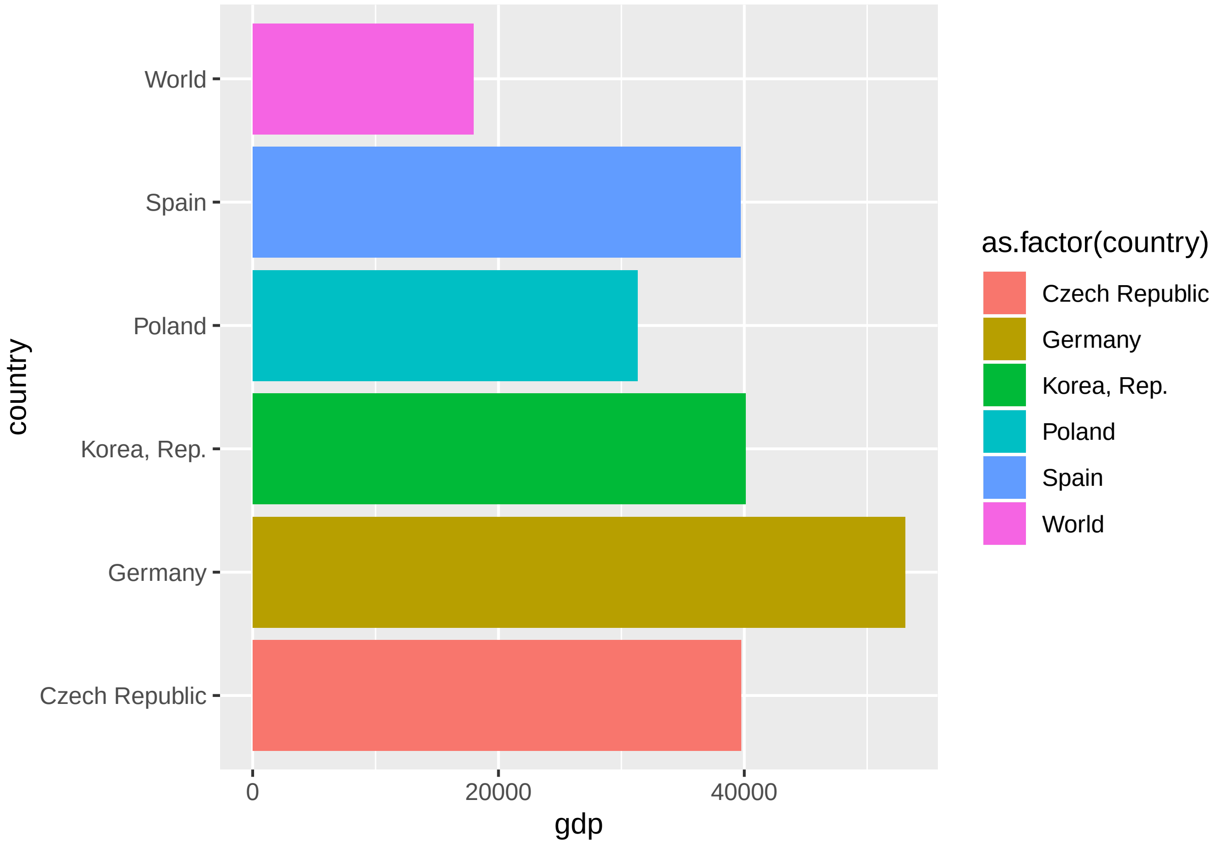
```
ggplot(test, aes(x = ________, y = ____ )) +
  geom_bar(stat = 'identity')
```



Plot customization

We can change the default plotting options by modyfying or adding extra arguments to the ggplot syntax. Try to add argument `fill = as.factor(country)` inside the `aes()` statement.

At the end of the created code add (i.e. with `+`) `coord_flip()` function which rotates the chart by 90 degrees. The results should be similar to the below shown chart:



ggplot(test, aes(x = country, y =gdp, fill =as.factor(country))) + geom_bar(stat = ‘identity’) + coord_flip()