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-formatted to the form readable in spreadsheet-like file.

Please download the file from http://iqdata.pl/dataprocessing/data/gdp_ppp.xlsx 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:

```
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 the 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')
```
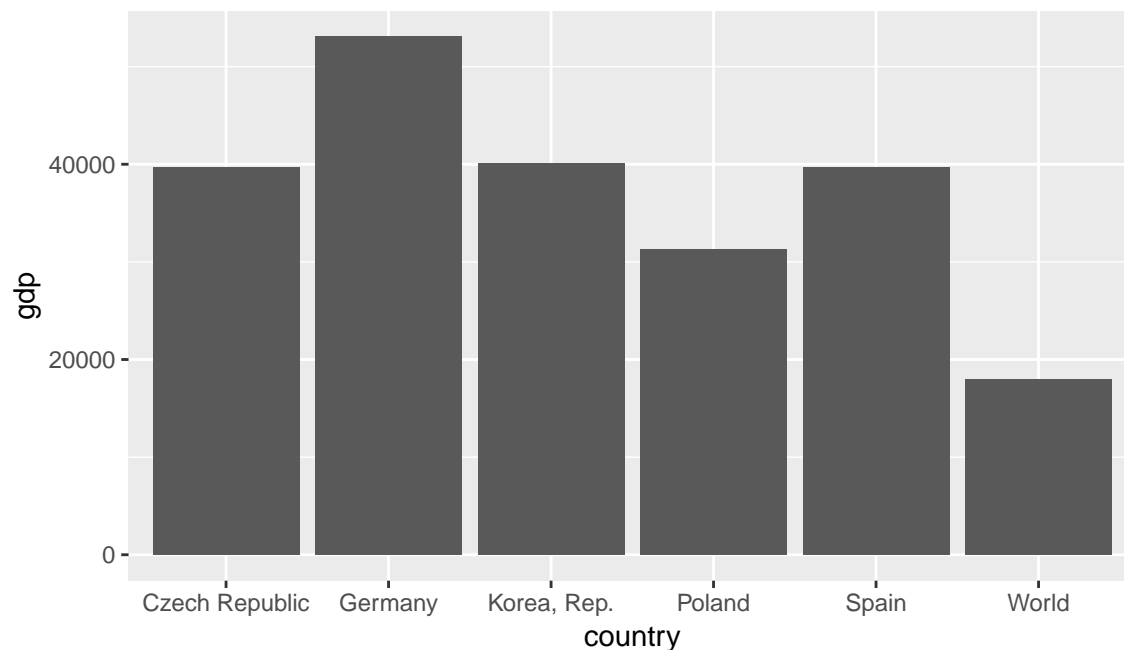
Plot customization

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 chart:
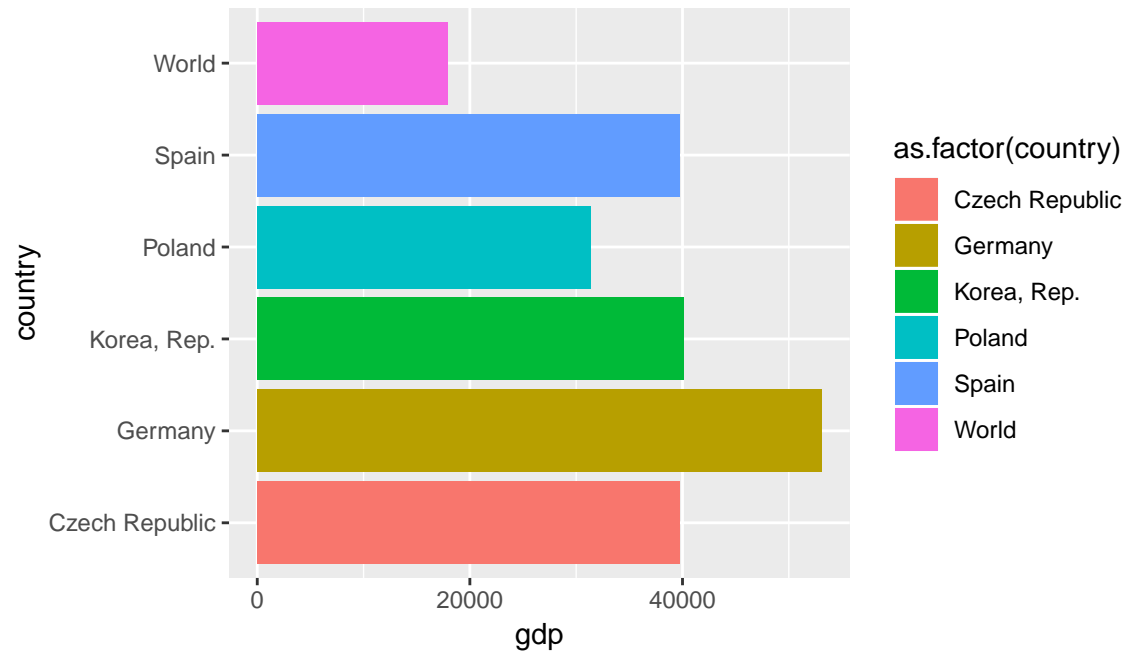
Removing legend

The default legend is redundant as we only show country names which are also labeled on the Y axis. Legend as many other ggplot items are controlled by the `theme()` function. In order to get rid off the legend please add to the plotting syntax: `theme(legend.position = "none")`
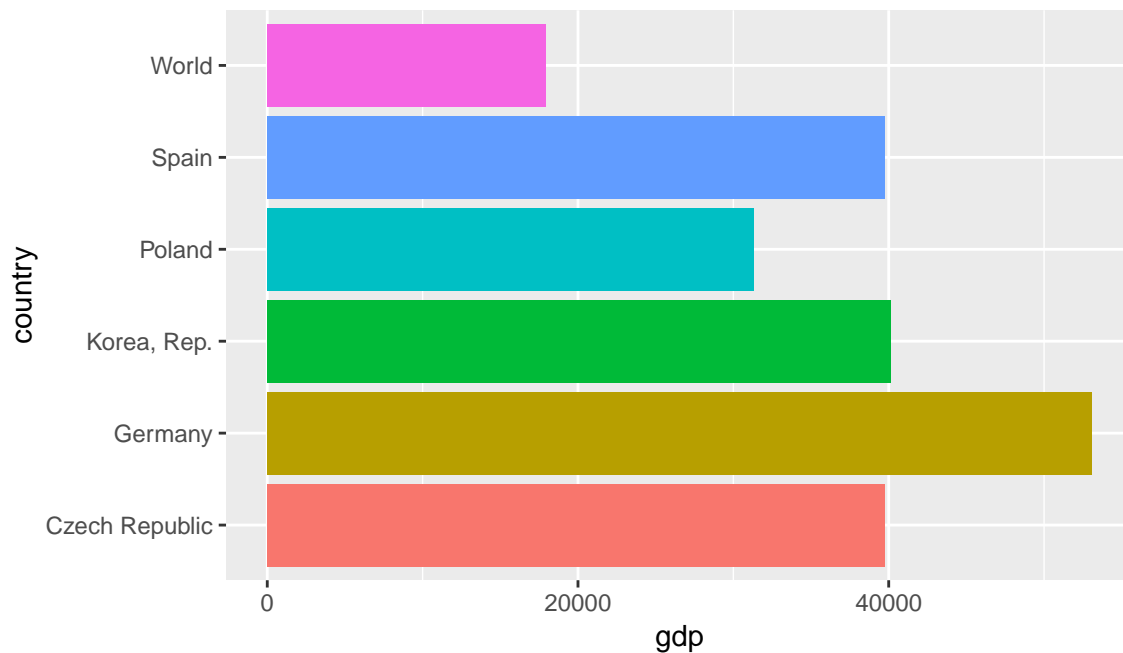
Titles & Labels

Our chart miss titles and labels. We can add them with `labs()` function at the end of `ggplot` syntax. For example:

```
labs(title = 'GDP PPP / capita',
     subtitle =  "For selected countries",
     caption  = "GDP PPP in USD/capita | Data Source: World Bank Data")
```
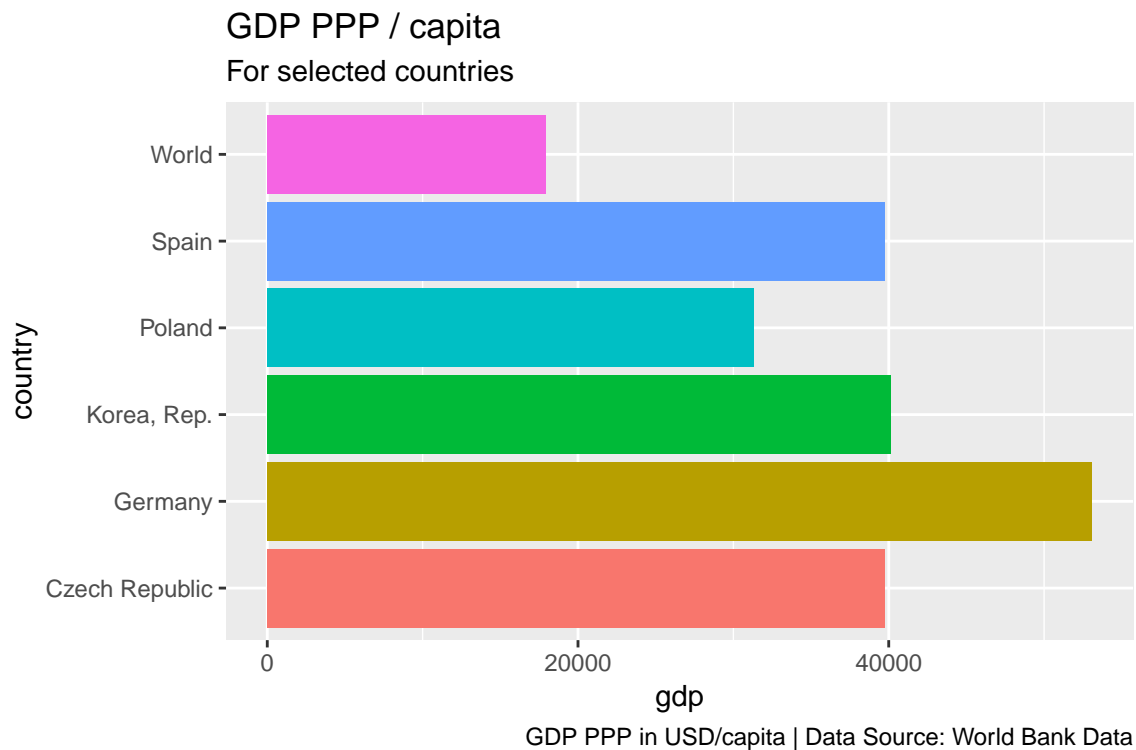
Adding text

We can also add another `geom_` to plot to show as text/label the real (or rounded) value for GDP that will be visible on the height of each bar. Try to play a bit with the `geom_text()` template given below and modify it to get the exemplary result:

```
geom_text(aes(y = ____, label = _____), hjust = 2, size= 5, col = '_____')+
```
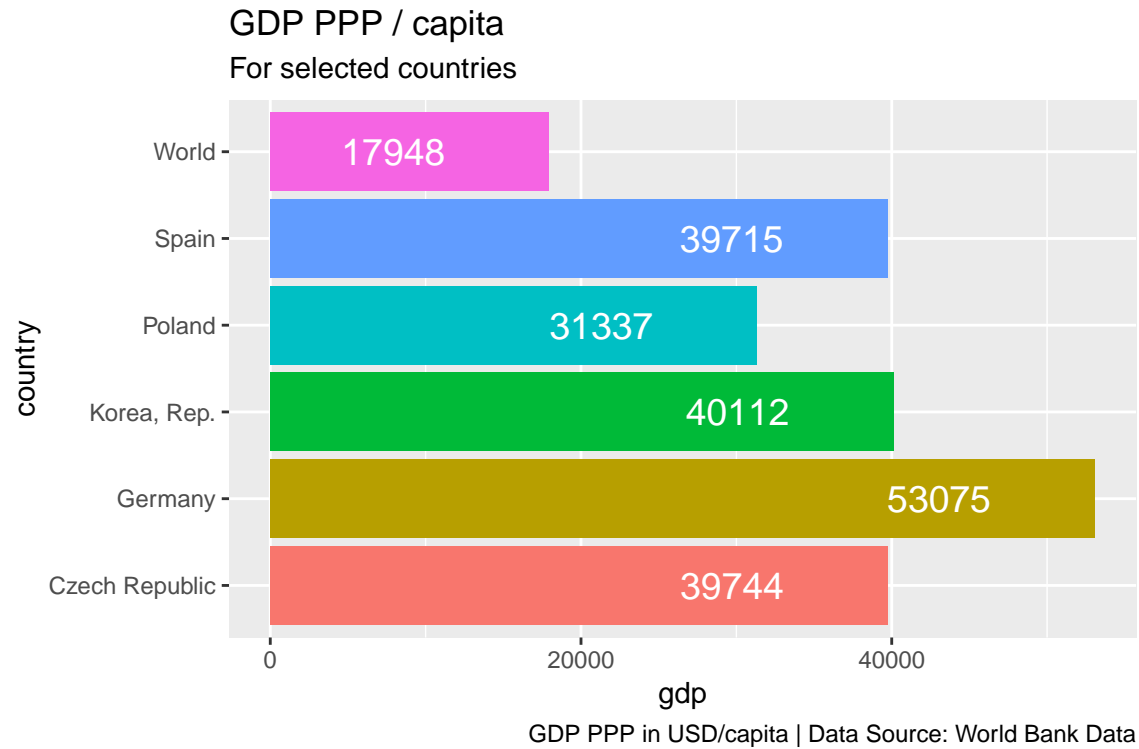

Plotting the full dataset

So far we have worked on a clipped dataset for year 2018 only. We can look at the whole dataset in a similar way by creating a separate charts for each individual year. It can be very easily done with the `facet_grid()` or `facet_wrap()` functions that are added to the created grammar/template of our chart. As the argument for this command please use the column name that will be used for the division with tilde (~) as prefix. Do not forget to change the `test` data frame with the full dataset.

GDP PPP / capita
For selected countries

country (y-axis) | **gdp** (x-axis)

GDP PPP in USD/capita | Data Source: World Bank Data

1990
- World: 5498
- Spain: 13664
- Poland: 4450
- Korea, Rep.: 8273
- Germany: 19497
- Czech Republic: 12660

1991
- World: 5684
- Spain: 14449
- Poland: 4900
- Korea, Rep.: 9346
- Germany: 21032
- Czech Republic: 11596

1992
- World: 5851
- Spain: 14843
- Poland: 5000
- Korea, Rep.: 10044
- Germany: 21763
- Czech Republic: 11788

1993
- World: 6040
- Spain: 14960
- Poland: 5700
- Korea, Rep.: 10875
- Germany: 21863
- Czech Republic: 12062

1994
- World: 6273
- Spain: 15570
- Poland: 6800
- Korea, Rep.: 12008
- Germany: 22760
- Czech Republic: 12673

1995
- World: 6528
- Spain: 16262
- Poland: 7665
- Korea, Rep.: 13299
- Germany: 23528
- Czech Republic: 13753

1996
- World: 6815
- Spain: 16958
- Poland: 8255
- Korea, Rep.: 14433
- Germany: 24048
- Czech Republic: 14583

1997
- World: 7104
- Spain: 17911
- Poland: 8922
- Korea, Rep.: 15406
- Germany: 24570
- Czech Republic: 14731

1998
- World: 7278
- Spain: 19096
- Poland: 9471
- Korea, Rep.: 14621
- Germany: 25345
- Czech Republic: 14906

1999
- World: 7536
- Spain: 19954
- Poland: 10022
- Korea, Rep.: 16393
- Germany: 26291
- Czech Republic: 15324

2000
- World: 7986
- Spain: 21587
- Poland: 10653
- Korea, Rep.: 18083
- Germany: 27203
- Czech Republic: 16191

2001
- World: 8261
- Spain: 22959
- Poland: 11102
- Korea, Rep.: 19168
- Germany: 28380
- Czech Republic: 17554

2002
- World: 8552
- Spain: 24372
- Poland: 11781
- Korea, Rep.: 20775
- Germany: 29179
- Czech Republic: 18189

2003
- World: 8917
- Spain: 25018
- Poland: 12257
- Korea, Rep.: 21377
- Germany: 29875
- Czech Republic: 19426

2004
- World: 9519
- Spain: 26120
- Poland: 13341
- Korea, Rep.: 22937
- Germany: 31305
- Czech Republic: 20799

2005
- World: 10133
- Spain: 27607
- Poland: 13896
- Korea, Rep.: 24196
- Germany: 31794
- Czech Republic: 21956

2006
- World: 10959
- Spain: 30688
- Poland: 15142
- Korea, Rep.: 25821
- Germany: 34125
- Czech Republic: 23776

2007
- World: 11722
- Spain: 32438
- Poland: 16788
- Korea, Rep.: 27822
- Germany: 36252
- Czech Republic: 26125

2008
- World: 12261
- Spain: 33263
- Poland: 18310
- Korea, Rep.: 28656
- Germany: 37802
- Czech Republic: 27845

2009
- World: 12233
- Spain: 32099
- Poland: 19246
- Korea, Rep.: 28325
- Germany: 36823
- Czech Republic: 27599

2010
- World: 12877
- Spain: 31680
- Poland: 21048
- Korea, Rep.: 30365
- Germany: 38950
- Czech Republic: 27667

2011
- World: 13592
- Spain: 31868
- Poland: 22851
- Korea, Rep.: 31229
- Germany: 42542
- Czech Republic: 28797

2012
- World: 14152
- Spain: 31720
- Poland: 23833
- Korea, Rep.: 32097
- Germany: 43360
- Czech Republic: 29047

2013
- World: 14764
- Spain: 32434
- Poland: 24719
- Korea, Rep.: 32616
- Germany: 4499
- Czech Republic: 30486

2014
- World: 15343
- Spain: 33526
- Poland: 25612
- Korea, Rep.: 33588
- Germany: 4701
- Czech Republic: 32263

2015
- World: 15815
- Spain: 34912
- Poland: 26856
- Korea, Rep.: 35761
- Germany: 476
- Czech Republic: 33691

2016
- World: 16358
- Spain: 36555
- Poland: 27735
- Korea, Rep.: 37143
- Germany: 495
- Czech Republic: 35231

2017
- World: 17151
- Spain: 38889
- Poland: 29922
- Korea, Rep.: 38824
- Germany: 52
- Czech Republic: 38020

2018
- World: 17948
- Spain: 39715
- Poland: 31337
- Korea, Rep.: 40112
- Germany: 53
- Czech Republic: 39744

7