Gapminder; Assignment 3

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

In this assignment we will start with the replication of the dataset in the R-package `gapminder` (Jenny Bryan). We will download data from the full Gapminder dataset *Systema Globalis*.

The Gapminder R-package contains data up to 2007. The current full dataset contains a lot more countries and also more variables. At the same time there seems to be less observations for some of the variables (time series now starts at a later date).

Some of you might also find R for Reproducible Scientific Analysis to be of some help. It's a short course in some of the same topics that we cover and uses the gapminder data.

Variables (in gapminder package) What we will try to recreate, but with new and more up to date data.

1. Country: 142
2. Continent: 5 (Africa, Americas, Asia, Europe, Oceania)
3. Year; 1952–2007
4. lifeExp: le at birth in years
5. pop: population
6. gdbPercap: in US $, inflation-adjusted

The data Clone https://github.com/open-numbers/ddf--gapminder--systema_globalis to your local disk (no need to fork since we will NOT du a pull request). Create a new repository at github and make it part of a RStudio project (as usual). Make a new folder at the root of the new project. Call this folder `Data`and copy the `ddf--gapminder--systema_globalis` folder into this folder.

The report Start a new R-Notebook and save it as `ass3.Rmd` at the root of your project. Use this document to answer assignment 3. In this assignment we focus on nice style compliant R code. In addition, try to give all the code-chunks meaningful names.

You don't need to write a paper in the normal sense. Concentrate on clear code and explain the code in the text.

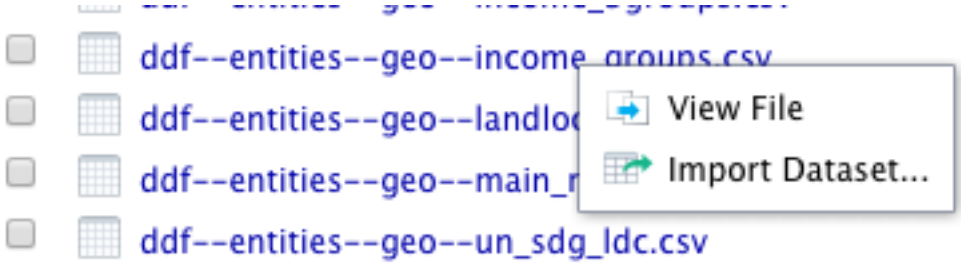


Figure 1: Import csv file.

Some tips If you click on a .csv file in the `Files` tab you will be presented with the following menu

Select `Import Dataset...` and you will get the dialogue box.

Note: In the following I sometimes show my answer. That is done intentionally to keep you on the right track. Your job is to write the code that generate that answer, or another one if you think I have made a mistake.

Answer the following questions

1. What information does the file `ddf_concepts.csv` contain.?
2. What information does the file `ddf--entities--geo--country.csv` contain?
3. What information does the file `ddf--gapminder--systema_globalis/ddf--entities--geo--un_sdg_region.csv` contain?
4. Recreate the `continent` variable with the new data. Only include countries that have a `iso3166_1_alpha3` code. Use data from `ddf--entities--geo--country.csv` and call this tibble `g_c`. Let `g_c` be your main tibble in the following, i.e. add variables to this tibble.

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   `is--country` = col_logical(),
##   iso3166_1_numeric = col_double(),
##   latitude = col_double(),
##   longitude = col_double(),
##   un_state = col_logical()
## )
## i Use `spec()` for the full column specifications.

## cols(
##   country = col_character(),
##   g77_and_oecd_countries = col_character(),
##   income_3groups = col_character(),
##   income_groups = col_character(),
##   `is--country` = col_logical(),
##   iso3166_1_alpha2 = col_character(),
##   iso3166_1_alpha3 = col_character(),
##   iso3166_1_numeric = col_double(),
```

```
##   iso3166_2 = col_character(),
##   landlocked = col_character(),
##   latitude = col_double(),
##   longitude = col_double(),
##   main_religion_2008 = col_character(),
##   name = col_character(),
##   un_sdg_ldc = col_character(),
##   un_sdg_region = col_character(),
##   un_state = col_logical(),
##   unicef_region = col_character(),
##   unicode_region_subtag = col_character(),
##   world_4region = col_character(),
##   world_6region = col_character()
## )
```

5. How many countries are there now?

The functions `unique()` and `length()` might be of some help.

```
## [1] 247
```

6. How many countries are there now in each continent?

```
## # A tibble: 5 x 2
## # Groups:   continent [5]
##   continent     n
##   <chr>     <int>
## 1 Africa       59
## 2 Americas     55
## 3 Asia         47
## 4 Europe       58
## 5 Oceania      28
```

Adding a new variable

We will now start the process of adding variables from files. We will import data from a `.csc` file and put it in a `tibble`. This tibble will then be joined with our original `tibble` with the help of a `left:join()`.

7. Read in the variable Life Expectancy (`lifeExp`) to `g_c` . You should change the format of the `time` variable to `date` with format `%Y` when you import the data (click on the column name). How many countries have information about `lifeExp`?

```
lifeExp <- read_csv("Data/ddf--gapminder--systema_globalis/countries-etc-datapoints/ddf--datapoints--li
    col_types = cols(time = col_date(format = "%Y")))

lifeExp <- lifeExp %>%
  rename(year = time)

length(unique(lifeExp$geo))
```

```
## [1] 189
```

```
names(g_c)
```

Select relevant variables in g_c

```
##  [1] "country"                "g77_and_oecd_countries" "income_3groups"
##  [4] "income_groups"          "is--country"            "iso3166_1_alpha2"
##  [7] "iso3166_1_alpha3"       "iso3166_1_numeric"      "iso3166_2"
## [10] "landlocked"             "latitude"               "longitude"
## [13] "main_religion_2008"     "name"                   "un_sdg_ldc"
## [16] "un_sdg_region"          "un_state"               "unicef_region"
## [19] "unicode_region_subtag"  "world_4region"          "world_6region"
## [22] "continent"
```

The tibble g_c now contains many variables that we do not need.

8. Reduce g_c to the variables: country, name, iso3166_1_alpha3, main_religion_2008, un_sdg_region, world_4region, continent, world_6region.

Select the variables we need from g_c.

```
g_c <- g_c %>%
  select(country, name, iso3166_1_alpha3, main_religion_2008, un_sdg_region, world_4region, continent, w
  left_join(lifeExp, by = c("country" = "geo")) %>%
  filter(!(is.na(year) & is.na(life_expectancy_years))) %>%
  filter(year < "2020-01-01")
```

9. What is the first observation of lifeExp for the different countries? (Hint; group_by() country and find minimum year for each group by using summarise()). Use a command like table(g_c_min$year_min) to make a table of the distribution of first year of life expectancy data. Find the names of the 3 countries that have the shortest series of Life Expectancy.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
##
## 1800-01-01 1970-01-01
##        186          3
```

```
## # A tibble: 3 x 1
##   country
##   <chr>
## 1 and
## 2 dma
## 3 mhl
```

```
## # A tibble: 3 x 1
##   name
##   <chr>
## 1 Andorra
## 2 Dominica
## 3 Marshall Islands
```

Read in population

10. Read in `total_population` and join with `g_c` . Remember to change the time variable from `integer` to `date`. Then left_join `g_c` and `pop`.

```
g_c <- g_c %>%
  left_join(pop, by = c("country" = "geo", "year" = "time"))
```

Read in urban population

11. Let `u_pop` be urban population. Import `urban_population` and `left_join` with `g_c`.

Read in GDP data

12. Read in `gdp_percapita_us_inflation_adjusted` and call it 'gdp_pc`. Left join the data with `g_c`.

```
##  [1] "country"                                "name"
##  [3] "iso3166_1_alpha3"                       "main_religion_2008"
##  [5] "un_sdg_region"                          "world_4region"
##  [7] "continent"                              "world_6region"
##  [9] "year"                                   "life_expectancy_years"
## [11] "population_total"                       "urban_population"
## [13] "gdppercapita_us_inflation_adjusted"
```

Make a dataset similar to `gapminder`

13. Restrict `g_c` to the period from 1962 to 2017 (in the `gapminder` package there is data from the years 1952 and 1957, but these seems to have disappeared from *systema_globalis)* and select the variables: name, continent, year, lifeExp, pop, gdpPercap. As in gapminder use data from every 5th year, i.e. 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, 2002, 2007, 2012, 2017.

```
# 1962-2017 all nations
dim(my_gapminder)
```

```
## [1] 2262    6
```

Make subset of 'gapminder'

14. Make a subset of `gapminder` , `my_gapminder_1962` ,which include countries *with* data from 1962-2017 (include Venezuela even though it has NA for `gdpPercap`2017). How many countries are now in the dataset? How many countries from each continent? How many NAs are there in `my_gapminder_1962`` ? (Hint: The function `duplicated()` might be of help)

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
# nations with registered data 1962 to 2017
dim(my_gapminder_1962)
```

```
## [1] 1080    6
```

```r
length(unique(my_gapminder_1962$country))
```

```
## [1] 90
```

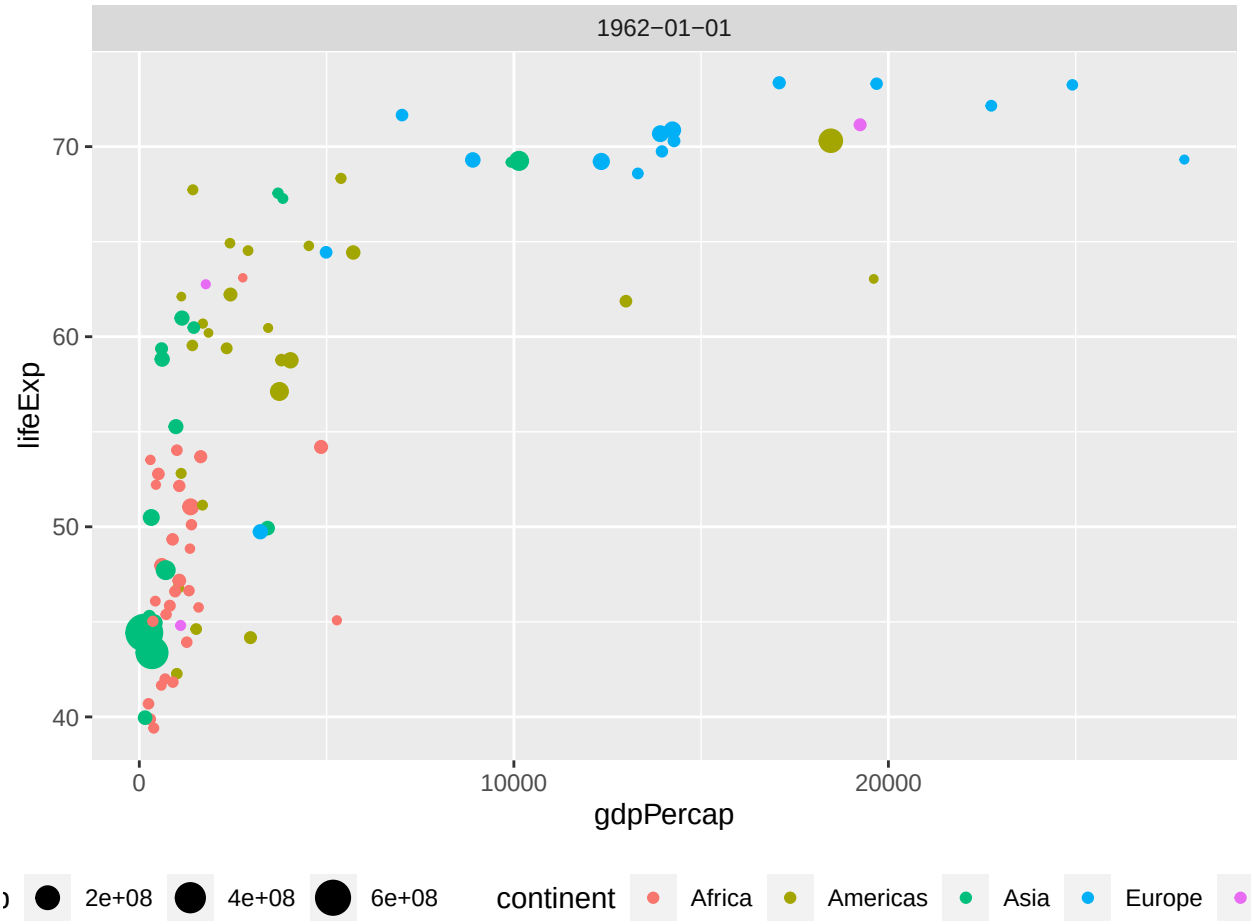
```r
my_gapminder_1962 %>%
  # distinct() is tidyverse for classic unique()
  distinct(country, continent) %>%
  group_by(continent) %>%
  count() %>%
  kable()
```

continent	n
Africa	30
Americas	25
Asia	17
Europe	15
Oceania	3

```
## # A tibble: 1 x 6
##   country   continent year       lifeExp      pop gdpPercap
##   <chr>     <chr>     <date>       <dbl>    <dbl>     <dbl>
## 1 Venezuela Americas  2017-01-01    75.3 29402480        NA
```

```
## [1] "Number of NAs in my_gapminder_1962 is 1"
```

15. Use `ggplot()` and let x be `gdpPercap` , y be `lifeExp` and *size* the population. Make a plot for each of the years 1962, 1987, 2017.



16. Do the same three plots as above, but now use the log transform of `gdpPercap`, i.e. `mapping = aes(x = log(gdpPercap), y...` and use `coord_cartesian(xlim = c(5, 12), ylim = c(30, 100))` after `geom_point` to control the layout of the axes.

17. How will you characterise the development the 55 years from 1962 to 2017?