Assignment Three

HVL DS 2020

Arian Steen Anne Grete Lilleland

30 10 2020

The Report

Our assignment report will include the explanation of the code used in this report, in order to analyze and look at the data imported from 'Gapminder-Systema_Globalis'.

We will aim at answering questions regarding the data, some of the questions are written below:

- What information does the file ddf_concepts.cssv contain?
- What information does the file ddf–entities–geo–country.csv contain?
- What information does the file ddf–entities–geo–un_sdg_region.csv contain?

We will also recreate the variable 'Continent', with the new data. We will only include countries that have a iso3166_1_alpha3 code.

We will also show the graphics/plots used within our report. Our report will include the code writting within the IDE for R 'Rstudio' and work within the 'tidy data' framework as our focuspoint.

We are going to write our report in English as to streamline the workflow between the book 'R for Everyone' our own report and make the repport more accessible.

The Data

The data that we are going to use in this report is taken from the official Gapminder website. Containing local and global statistics from several hundrer sources. Including but not limited to : Geographical data, Income data, age statistics and population,density data.

The Assignment

1 What information does *'ddf_concepts.csv'* contain? :

This file contains certain information collected within the dataset, including the source url and description of the data collected. This file is used to explain the gapminder datafile we are going to use in our report, as such this is purely an explanatory text file that does not contain statistics, numbers or other variables.

The total number of observations within the file is 596 with 17 variable columns, the different description-variables includes name, catalog , short description , url of the source and type of measurements done.

2 What information does *'ddf–entities–geo–country.csv'* contain? :

This file contains certain information about different countries, the different variables include income, religion , region and wheter or not the country is landlocked (Access to a coastline within the borders of the country). This file contains 273 observations, and 21 variable columns.

3 What information does *'ddf–entities–geo–un_sdg_region.csv'* contain? :

This file contains information about different continents and countries, there are 8 regions in this file, each with their own unique color to make it easier to distinguish between them in a graphical setting.

A sample from the file would be : un_europe_and_northern_america this would be the regions of Europe and Northern America, with its own unique color to make displaying the data easier in a graphical setting.

4 Recreating the continent variable.

Recreate the continent variable with the new data. Only include countries that have aiso3166_1_alpha3code. 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.

```
# We use readr to load the csv file and then create a new continent variable called g_c
g_c <- read_csv("C:/Users/ASCUSERADMIN/Documents/RMASTER/AssignmentThree/Data/ddf--gapminder--systema_g
```

```
##
## -- 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.
```

```
print(g_c) # The print function pertaining to tibbles is useful in our case.
```

```
## # A tibble: 273 x 21
##    country g77_and_oecd_co~ income_3groups income_groups `is--country`
##    <chr>   <chr>            <chr>          <chr>         <lgl>
##  1 abkh    others           <NA>           <NA>          TRUE
##  2 abw     others           high_income    high_income   TRUE
##  3 afg     g77              low_income     low_income    TRUE
##  4 ago     g77              middle_income  lower_middle~ TRUE
##  5 aia     others           <NA>           <NA>          TRUE
##  6 akr_a_~ others           <NA>           <NA>          TRUE
##  7 ala     others           <NA>           <NA>          TRUE
##  8 alb     others           middle_income  upper_middle~ TRUE
##  9 and     others           high_income    high_income   TRUE
## 10 ant     others           <NA>           <NA>          TRUE
## # ... with 263 more rows, and 16 more variables: iso3166_1_alpha2 <chr>,
```

```
## #   iso3166_1_alpha3 <chr>, iso3166_1_numeric <dbl>, iso3166_2 <chr>,
## #   landlocked <chr>, latitude <dbl>, longitude <dbl>,
## #   main_religion_2008 <chr>, name <chr>, un_sdg_ldc <chr>,
## #   un_sdg_region <chr>, un_state <lgl>, unicef_region <chr>,
## #   unicode_region_subtag <chr>, world_4region <chr>, world_6region <chr>
```

Above we have created a new variable with data from ddf_editites-geo-country.csv called g_c, this is unfiltered with the same data

We will now filter out the countries that have iso3166_1_alpha3code. This is an international stand ISO code pertaining to countries/geographical locations. We use the filtering option to extract this information from the first data set (ddf_entities_geo_country.csv)

```
# This code helps us filter out countries in our dataset that fit the iso3166_1_alpha3 ISO standard. We
g_c <- g_c %>%
  mutate(continent = case_when(
    world_4region == "asia" & un_sdg_region %in% c("un_australia_and_new_zealand", "un_oceania_exc_austr
    world_4region ==  "asia" & !(un_sdg_region %in% c("un_australia_and_new_zealand", "un_oceania_exc_au
    world_4region == "europe" ~ "Europe",
    world_4region == "africa" ~ "Africa",
    world_4region ==  "americas" ~ "Americas")
  ) %>%
 filter(!is.na(iso3166_1_alpha3))
```

We have now filtered out the countries with that particular code.

5 How many countries are there now?

```
unique(g_c) # This function shows us the number and length of g_c.
```

```
## # A tibble: 247 x 22
##    country g77_and_oecd_co~ income_3groups income_groups `is--country`
##    <chr>   <chr>            <chr>          <chr>         <lgl>
##  1 abw     others           high_income    high_income   TRUE
##  2 afg     g77              low_income     low_income    TRUE
##  3 ago     g77              middle_income  lower_middle~ TRUE
##  4 aia     others           <NA>           <NA>          TRUE
##  5 ala     others           <NA>           <NA>          TRUE
##  6 alb     others           middle_income  upper_middle~ TRUE
##  7 and     others           high_income    high_income   TRUE
##  8 are     g77              high_income    high_income   TRUE
##  9 arg     g77              middle_income  upper_middle~ TRUE
## 10 arm     others           middle_income  upper_middle~ TRUE
## # ... with 237 more rows, and 17 more variables: iso3166_1_alpha2 <chr>,
## #   iso3166_1_alpha3 <chr>, iso3166_1_numeric <dbl>, iso3166_2 <chr>,
## #   landlocked <chr>, latitude <dbl>, longitude <dbl>,
## #   main_religion_2008 <chr>, name <chr>, un_sdg_ldc <chr>,
## #   un_sdg_region <chr>, un_state <lgl>, unicef_region <chr>,
## #   unicode_region_subtag <chr>, world_4region <chr>, world_6region <chr>,
## #   continent <chr>
```

```
length(g_c$name) # This function shows us the number of variables (22) pertaining to g_c.
```

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

As we can see, *we now have 247 observations* and 22 variables to work with (We started with 273 observations and 21 variables). We have now filtered out the countries with iso3166_1_alpha3 code. This means that *26 countries in our data did not fit the iso3166_1_alpha3code.* **There are now 247 countries that fit the isocode.**

6 Number of countries in each continent?

```
g_c %>%
    count(continent)
```

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

```
# Using the count function we can show how many countries there are in each continent
contnumbercountries <- c(59,55,47,58,28)
mean(contnumbercountries)
```

```
## [1] 49.4
```

In the following order: 59,55,47,58,28 countries in Africa,Americas,Asia,Europe and Oceania

7 Adding a new variable

```
lifeExp <-read_csv("Data/ddf--gapminder--systema_globalis/countries-etc-datapoints/ddf--datapoints--life
lifeExp <- lifeExp%>%
rename(year = time)
length(unique(lifeExp$geo))
```

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

After importing the data, there are 189 countries with information about Life Expectancy (lifeExp).

8 Reducing g_c variables

```r
names(g_c) # Here we can see the current variables in our file. We can further pull out the ones we are
```

```
##  [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"
```

We are now going to reduce g_c to the variables: country, name, iso3166_1_alpha3, main_religion_2008, un_sdg_region,world_4region, continent, world_6region.

```r
g_c <- g_c%>% # Here we are selecting the different variables and pulling them out of the data
    select(country, name, iso3166_1_alpha3, main_religion_2008, un_sdg_region, world_4region, continent

  left_join(lifeExp, by =c("country" = "geo")) %>%
    filter(!(is.na(year)& is.na(life_expectancy_years))) %>%
    filter(year<"2020-01-01")
```

9 Observations on life expectancy

10 Reading in population

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

12 Reading in gdp_percapita_us_inflation_adjusted

13 Making a gapminder-like dataset

14 Making subset of gapminder

15

16

17