# explanation

YifanZhang

10/6/2021

**inward-facing file**

## wrangling_code.R

```
library(tidyverse)
library(dplyr)
library(readxl)
library(data.table)
# Year 1799 - 2049
# including 195 countries
# data_1: lifer expectancy in different countries from 1799 to 2049(expected)
# data_2: income per person(gdp) in different countries from 1799 to 2049(expected)
data_1 <- read.csv("https://raw.githubusercontent.com/MA615-Yifan/MA615_hw2/main/life_expectancy_years.c
                   header = TRUE)
data_1 <- data_1[,1:252]
data_2 <- read.csv("https://raw.githubusercontent.com/MA615-Yifan/MA615_hw2/main/income_per_person_gdppe
                   header = TRUE)
```

the content in this chunk is to load required package and load two dataset(both in csv form). Meanwhile, I write the code "data_1 <- data_1[,1:252]" because the second dataset only contains data from 1799 to 2049. Since our present time is 2021, datas of year after this are prediciton data, it is reasonable to remove the 2050-2099 data in first dataset.

```
# country_n <- number of country
# year_n <- number of years, notice that the first column are the names of the country
# thus, number of years should be dim(data_1)[2] - 1
country_n <- dim(data_1)[1]
year_n <- dim(data_1)[2] - 1

# construct different indicators
# set an empty vector 'country' to store names of countries
country <- vector()
for (i in 1: country_n) {
  country <- c(country, rep(data_1$country[i], year_n))
}
```

since for each country, we have year_n data for each indicator, name of each country should repeat year_n times and repeated names of countries are stored in the first column later.

```
# set an empty vector 'life_expect' for the first indicator 'life expectancy'
# notice that what we extract with data_1[i, ] is a list, unlist() is necessary here
# what's more, the first element in data_1[i, ] (after unlist) is the name of
life_expect <- vector()
for (i in 1: country_n){
  a <- data_1[i, ]
  b <- unlist(a)
  life_expect <- c(life_expect, as.numeric(b[2:(year_n + 1)]))
}
```

notice that what we extract with data_1[i, ] is a list, unlist() is necessary here. what's more, the first element in data_1[i, ] (after unlist) is the name of country, so the code "as.numeric(b[2:(year_n + 1)])" is to remove that.

```
# set an empty vector 'income_per_person' for the first indicator 'income_per_person'
income_per_person <- vector()
for (i in 1: country_n) {
  a <- data_2[i, ]
  b <- unlist(a)
  income_per_person <- c(income_per_person, as.numeric(
    sub("k", "e3", b[2:(year_n + 1)], fixed = TRUE)))
}
```

similar with previous chunk. the only difference is that the income data using 'k' to represent 1000$, so I use sub() function to substitute 'k' with 1000. fortunately, that make sense.

```
# generate variable 'year'
year <- as.numeric(rep(c(1799:2049), country_n))
data <- data.frame(country, year, life_expect, income_per_person)
```

it is easy to understand why I have to repeat "c(1799:2049)" for country_n times. afterwards, using data.frame() function to put all variables in a data table. uptill now, data tidy is done. "Each variable forms a column", "Each observation forms a row" and "Each type of observational unit forms a table" are satisfied.

### inward-facing file

```
# my function needs four input: dataset, country1, country2 and an indicator
# country1 and country2 form an comparision group and indicator decide what to plot.
data_visulization <- function(data, country1, country2, indicator){
# consider whether both of the input country are in the dataset. if 'yes' continue, otherwise, stop fun
  if(dim(filter(data, country == country1))[1] == 0 | dim(filter(data, country == country2))[1] == 0){
    stop("error: country does not exist.")
  }
# distinguish which indicator is put into function
  if(indicator == "life"){
# filter the data
    data1 <- data %>%
      filter(country == country1)
    data2 <- data %>%
      filter(country == country2)
# add an average value "mean year" line to the plot
    mean_year <- data %>% group_by(year) %>%
      summarize(mean = mean(life_expect, na.rm = TRUE))
    mean_year <- data.frame(country = rep("world mean", dim(mean_year)[1]), year = mean_year$year, life_
# put all of the plotting data in to table
    plot_data <- rbind(data.frame(country = data1$country, year = data1$year, life_expect = data1$life_
                       data.frame(country = data2$country, year = data2$year, life_expect = data2$life_
                       mean_year)
# using the names of countries to distinguish the plot
    ggplot(data = plot_data, mapping = aes(x = year, y = life_expect,
                                           group = country, color = country, shape = country), xlim = c
      geom_line() + geom_point(size = .7) +
      xlab("Year") + ylab("life expectancy")
  }
  else if(indicator == "income"){
    data1 <- data %>%
      filter(country == country1)
    data2 <- data %>%
      filter(country == country2)
    mean_income <- data %>% group_by(year) %>%
      summarize(mean = mean(income_per_person, na.rm = TRUE))
    mean_income <- data %>% group_by(year) %>%
      summarize(mean = mean(income_per_person, na.rm = TRUE))
    mean_income <- data.frame(country = rep("world mean", dim(mean_income)[1]), year = mean_income$year
    plot_data <- rbind(data.frame(country = data1$country, year = data1$year, income_per_person = data1$
                       data.frame(country = data2$country, year = data2$year, income_per_person = data2$
                       mean_income)
    ggplot(data = plot_data, mapping = aes(x = year, y = income_per_person,
                                           group = country, color = country, shape = country), xlim = c
      geom_line() + geom_point(size = .7) + xlab("Year") + ylab("income per person($)")
  }
# if the input indicator is neither 'life expectancy' nor 'income-per-person', stop function and send e
  else{
    stop("error: indicator does not exist. ")
  }
}
```

all the code above can be divided into three parts: 1. make judgements, whether the input value is right to draw a plot. if ‘yes’, do the plot; otherwise, stop and give an error as feedback. 2. construct plot data. since my visulization is to do a comparision beteween two countries. I thought adding an average line of whole world to the plot sounds reasonable and makes the comparision more straightforward. 3. draw the plot, since both two countries’ data and the mean data are bind by row, in the plotting step, I uses names of countries to be the color so that making different line in the plot is easy to be distinguish.

### presentation

```
source("wrangling_code.R", echo=TRUE)
source("visulization_function.R", echo=TRUE)
# notice: indicator can only be chosen from "life" or "income", otherwise, report error.
# Please type the right country names! otherwise, the function will report error.
# PS: you can click to open the 'data' in work place to check the name of country
data_visulization(data, country1 = "United States", country2 = "Japan", indicator = "life")
data_visulization(data, country1 = "United States", country2 = "Japan", indicator = "income")
```

compare the life expectancy and income per person for United States and Japan(both are developed countries) from year 1799 ot 2049. and the following code is to show the income difference between developing countries and developed countries.

```
data_visulization(data, country1 = "United States", country2 = "China", indicator = "income")
data_visulization(data, country1 = "United States", country2 = "India", indicator = "income")
data_visulization(data, country1 = "Japan", country2 = "China", indicator = "income")
data_visulization(data, country1 = "Japan", country2 = "India", indicator = "income")
```