Functions Assignment W6

Christoffer M. Kramer

21/10/2020

# Question 1

Define a defensive function that calculates the Gross Domestic Product of a nation from the data available in the gapminder dataset. Using that function, calculate the GDP of Denmark in the following years: 1967, 1977, 1987, 1997, 2007, and 2017.

First I load the relevant libraries and disable scientific notation.

library(gapminder)  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.3

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

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

options(scipen = 999) #disable scientific notation

Then I create a function to calculate gdp. This solution is found through [software carpentry’s tutorial](http://swcarpentry.github.io/r-novice-gapminder/10-functions/index.html). However, I have changed the names, since I find longer, but more descriptive names easier to understand. It calculates GDP by subsetting the Gapminder data from the parameters “year” and “country”, then it multiplies “gdpPerCap” with “pop” in that subset. The default parameters are Null and it is defensive since it starts out by checking if the parameters are NULL. If no parameters are given, it will work on the whole datase. If only one parameter is given it will create a subset based on that parameter. Lastly, it creates a new column called GDP.

calcGDP <- function(dataset, year=NULL, country=NULL) {  
 if(!is.null(year)) { #If year is not null  
 dataset <- dataset[dataset$year %in% year,] #subset the dataset by year  
 }  
 if (!is.null(country)) { #if country is not null  
 dataset <- dataset[dataset$country %in% country,] #subset the dataset by country  
 }  
   
 gdp\_result <- dataset$pop \* dataset$gdpPercap #calculate gdp by multiplying GDP per capita with the total population.  
 #Save the result in the object "gpd\_result"  
   
 new\_column <- cbind(dataset, gdp=gdp\_result) #create a new column called GDP which contains the object "gdp\_result"   
 return(new\_column)  
}

Lastly, I just need to write the relevant values in to the parameters. Since I don’t want to repeat a lot of code, I create a vector called “years”, which contains all the years, and use it as a parameter

years <- c(1967, 1977, 1987, 1997, 2007, 2017) #Create a vector containing all the years  
calcGDP(gapminder, year = years, country = "Denmark") #calculate the gdp for each year in the vector

## country continent year lifeExp pop gdpPercap gdp  
## 1 Denmark Europe 1967 72.960 4838800 15937.21 77116977700  
## 2 Denmark Europe 1977 74.690 5088419 20422.90 103920280028  
## 3 Denmark Europe 1987 74.800 5127024 25116.18 128771236166  
## 4 Denmark Europe 1997 76.110 5283663 29804.35 157476118456  
## 5 Denmark Europe 2007 78.332 5468120 35278.42 192906627081

Since there aren’t any data from 2017 the function only return 5 rows,

# Question 2

Write a script that loops over each country in the gapminder dataset, tests whether the country starts with a ‘B’ , and print out whether the life expectancy is smaller than 50, between 50 and 70, or greater than 70.

I start by creating an object called “mean\_life\_country”. This object groups by country. It then filters out all rows, where the country does not start with the letter B. Then it summarises by calculating the average life expectancy for each country and creates the column “mean\_life”.

mean\_life\_country <- gapminder %>% #Create a new object called "mean\_life\_country"  
 group\_by(country) %>% #group by country  
 filter(str\_detect(country, "^B")) %>% #filter out rows where the first letter of the column "country" is not B  
 summarise(mean\_life = mean(lifeExp)) #create a new column called "mean\_life"

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

I then create a new csv document which contains the previous object. Then I set the value of the object “mean\_life\_country” to be the csv file.

write.csv(mean\_life\_country, "mean\_life\_country.csv") #create a csv document called "mean\_life\_country.csv  
mean\_life\_country <- read.csv("mean\_life\_country.csv") #let the object "mean\_life\_country" contain the csv file

I then loop through each country in my csv file. I get the value of the column “mean\_life” and store it in a new object called “mean\_life\_i”. Lastly I use if-else statements to check if the value of “mean\_life\_i” is smaller than 50, between 50 and 70, or above 70. It then prints out the value of the country (i) and the value of “mean\_life\_i”.

for (i in mean\_life\_country$country) { #for every country in "mean\_life\_country"  
   
 mean\_life\_i <- mean\_life\_country[mean\_life\_country$country == i, "mean\_life"] #create an object called "mean\_life\_i, which is a subvector of "mean\_life\_country"   
   
 if(mean\_life\_i < 50) { #if "mean\_life\_i i smaller than 50  
 cat("The life expectancy of", i, "is smaller than 50", "(",mean\_life\_i,")\n") #print this out  
 }  
 else if (mean\_life\_i > 50 && mean\_life\_i < 70) { #if "mean\_life\_i" is larger than 50 and smaller than 70  
 cat("The life expectancy of", i, "is between 50 and 70", "(",mean\_life\_i,")\n") #print this out  
   
 } else { #if "mean\_life\_i" is larger than 70  
 cat("The life expectancy of", i, "is larger than 70", "(",mean\_life\_i,")\n") #print this out  
   
 }  
  
}

## The life expectancy of Bahrain is between 50 and 70 ( 65.60567 )  
## The life expectancy of Bangladesh is smaller than 50 ( 49.83408 )  
## The life expectancy of Belgium is larger than 70 ( 73.64175 )  
## The life expectancy of Benin is smaller than 50 ( 48.77992 )  
## The life expectancy of Bolivia is between 50 and 70 ( 52.50458 )  
## The life expectancy of Bosnia and Herzegovina is between 50 and 70 ( 67.70783 )  
## The life expectancy of Botswana is between 50 and 70 ( 54.5975 )  
## The life expectancy of Brazil is between 50 and 70 ( 62.2395 )  
## The life expectancy of Bulgaria is between 50 and 70 ( 69.74375 )  
## The life expectancy of Burkina Faso is smaller than 50 ( 44.694 )  
## The life expectancy of Burundi is smaller than 50 ( 44.81733 )

# Question 3

Write a script that loops over each country in the gapminder dataset, tests whether the country starts with a ‘M’ and graphs life expectancy against time (using plot() function) as a line graph if the mean life expectancy is under 50 years.

I use same approach as above. But now I need to also keep the years. I, therefore, create two csv files.

The first csv file (country\_m) filters out all countries, where the first letter is “M”.This will be used for plotting.

country\_m <- gapminder %>% #Create a new object called "country\_m"  
 group\_by(country) %>% #group by country  
 filter(str\_detect(country, "^M")) #filter out rows where the first letter of the column "country" is not M  
  
 write.csv(country\_m, "country\_m.csv") #create a csv document called "country\_m.csv  
 country\_m <- read.csv("country\_m.csv") #let the object "country\_m" contain the csv file

The second csv file is a summary of the average life expectancy of each country (based on the previous csv file), with column a called “mean\_life”. This will be used in order to check whether the average life expectancy is under 50.

mean\_life\_country\_m <- country\_m %>% #Create a new object called "mean\_life\_country\_m"  
 group\_by(country) %>% #group by country  
 summarise(mean\_life = mean(lifeExp)) #create a new column called "mean\_life" with the average life expectancy

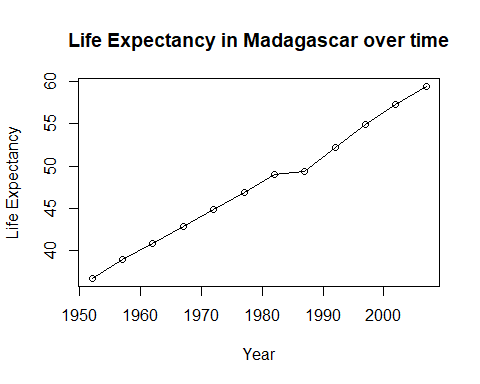
## `summarise()` ungrouping output (override with `.groups` argument)

write.csv(mean\_life\_country\_m, "mean\_life\_country\_m.csv") #create a csv file called "mean\_life\_country.csv  
 mean\_life\_country\_m <- read.csv("mean\_life\_country\_m.csv") #let the object "mean\_life\_country" contain the csv file

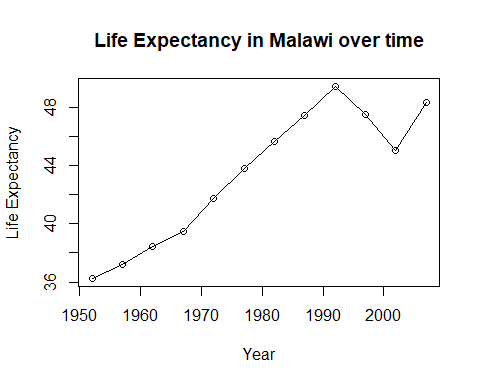
I then create a loop, which essentially works as the previous loop. It loops over each country in the csv file “mean\_life\_country\_m”. It then creates an object called “mean\_life\_i” which is a subvector of “mean\_life\_m” containing the value of the column “mean\_life”“. It then checks whether or not”mean\_life\_i" is below 50. If this is true it plots out the life expectancy for each year.

for (i in mean\_life\_country\_m$country) { #for every country in "mean\_life\_country"  
   
 mean\_life\_i <- mean\_life\_country\_m[mean\_life\_country\_m$country == i, "mean\_life"] #create an object called "mean\_life\_i, which is a subvector of "mean\_life\_country"   
   
 if(mean\_life\_i < 50) { #if "mean\_life\_i is smaller than 50  
 cat("The life expectancy of", i, "is smaller than 50", "(",mean\_life\_i,") plotting life expectancy\n") #print this out  
   
 with(subset(country\_m, country==i), #with a subset of data from country\_m  
 plot(year, lifeExp, #plot year on x axis and life lifeExp on the y axis  
 type="o", #Make it at line and point plot  
 main = paste("Life Expectancy in", i, "over time"), #Create Title  
 ylab = "Life Expectancy", #Rename y axis  
 xlab = "Year" #Rename x axis  
 ) # end plot  
 ) # end with  
 }  
}

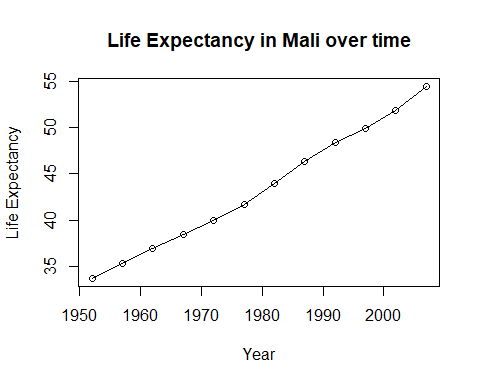
## The life expectancy of Madagascar is smaller than 50 ( 47.77058 ) plotting life expectancy



## The life expectancy of Malawi is smaller than 50 ( 43.35158 ) plotting life expectancy



## The life expectancy of Mali is smaller than 50 ( 43.4135 ) plotting life expectancy



## The life expectancy of Mozambique is smaller than 50 ( 40.3795 ) plotting life expectancy

