HW5 - Make Data Move

05/10/2020

# Explore the recent global developments with R

Today, you will load a filtered gapminder dataset - with a subset of data on global development from 1952 - 2007 in increments of 5 years - to capture the period between the Second World War and the Global Financial Crisis.

**Your task: Explore the data and visualise it in both static and animated ways, providing answers and solutions to 7 questions/tasks below.**

## Get the necessary packages

First, start with installing the relevant packages ‘tidyverse’, ‘gganimate’, and ‘gapminder’.

## 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()

## Look at the data

First, see which specific years are actually represented in the dataset and what variables are being recorded for each country. Note that when you run the cell below, Rmarkdown will give you two results - one for each line - that you can flip between.

unique(gapminder$year)

## [1] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007

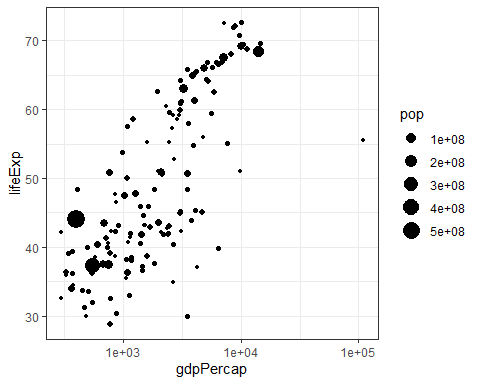
head(gapminder)

## # A tibble: 6 x 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Afghanistan Asia 1952 28.8 8425333 779.  
## 2 Afghanistan Asia 1957 30.3 9240934 821.  
## 3 Afghanistan Asia 1962 32.0 10267083 853.  
## 4 Afghanistan Asia 1967 34.0 11537966 836.  
## 5 Afghanistan Asia 1972 36.1 13079460 740.  
## 6 Afghanistan Asia 1977 38.4 14880372 786.

The dataset contains information on each country in the sampled year, its continent, life expectancy, population, and GDP per capita.

Let’s plot all the countries in 1952.

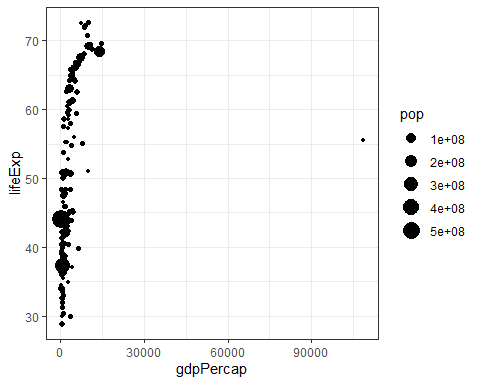
theme\_set(theme\_bw()) # set theme to white background for better visibility  
  
ggplot(subset(gapminder, year == 1952), aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point() +  
 scale\_x\_log10()



We see an interesting spread with an outlier to the right. Answer the following questions, please:

Q1. Why does it make sense to have a log10 scale on x axis? Answer: It is purely an aesthetic choice, which makes the visualization more understandable. A difference of 1 on this scale is equal to a tenfold increase (I think this is the case, but I might have misunderstood the scale, I’m not a mathematician), which spreads the datapoints. This is necessary because of the outlier. Below there is a graph without the log10 scale. Look how crammed it is!

theme\_set(theme\_bw()) # set theme to white background for better visibility  
  
ggplot(subset(gapminder, year == 1952), aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point()



Q2. What country is the richest in 1952 (far right on x axis)? Answer: Kuwait

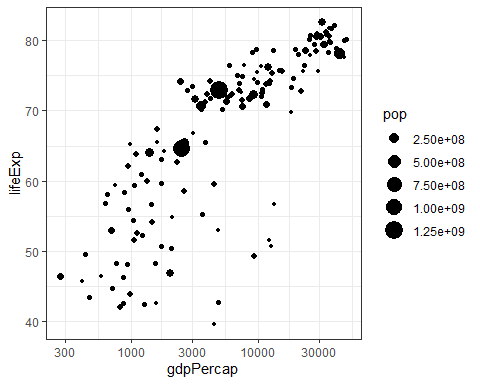
gapminder\_sorted\_1952 <- gapminder %>%   
 select(country, year, gdpPercap)%>%   
 filter(year == 1952) %>%   
 arrange(desc(gdpPercap))   
   
gapminder\_sorted\_1952[1,]

## # A tibble: 1 x 3  
## country year gdpPercap  
## <fct> <int> <dbl>  
## 1 Kuwait 1952 108382.

I started by creating a new object called “gapminder\_sorted\_1952”. This object contains the columns “year”, “country” and “gdpPercap” from the gapminder dataset (the other columns are irrelevant). Then I filter out all rows where the value of year is not 1952. Then I arrange the columns based on the value of the column “gdpPercap” in descending order. Lastly I chose the first row from the new object

You can generate a similar plot for 2007 and compare the differences

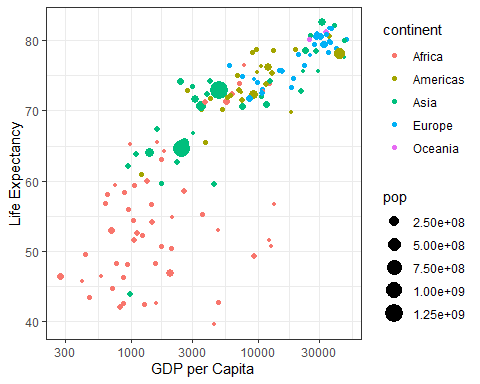
ggplot(subset(gapminder, year == 2007), aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point() +  
 scale\_x\_log10()



The black bubbles are a bit hard to read, the comparison would be easier with a bit more visual differentiation.

Q3. Can you differentiate the continents by color and fix the axis labels?

ggplot(subset(gapminder, year == 2007), aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point(aes(color=continent)) + #Make it a point graph, and set the color by continent  
 labs(y = "Life Expectancy") + #Make the label for the y axis "life Expectancy"  
 labs(x = "GDP per Capita") + #Make the label for the x axis "GDP per Capita"  
 scale\_x\_log10()



Q4. What are the five richest countries in the world in 2007? Answer: Noway, Kuwait, Singapore, United States and Ireland

gapminder\_sorted\_2007 <- gapminder %>%   
 select(country, year, gdpPercap)%>%   
 filter(year == 2007) %>%   
 arrange(desc(gdpPercap))  
  
gapminder\_sorted\_2007[1:5,]

## # A tibble: 5 x 3  
## country year gdpPercap  
## <fct> <int> <dbl>  
## 1 Norway 2007 49357.  
## 2 Kuwait 2007 47307.  
## 3 Singapore 2007 47143.  
## 4 United States 2007 42952.  
## 5 Ireland 2007 40676.

I copy pasted the code from Question 2 and changed the year to 2007 and changed the name of the object to “gapminder\_sorted\_2007”. Then I choose row 1 to 5.

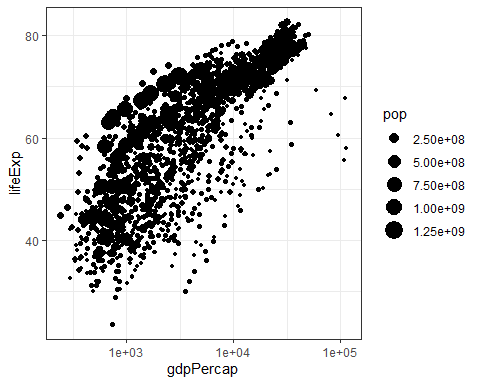
## Make it move!

The comparison would be easier if we had the two graphs together, animated. We have a lovely tool in R to do this: the gganimate package. And there are two ways of animating the gapminder ggplot.

### Option 1: Animate using transition\_states()

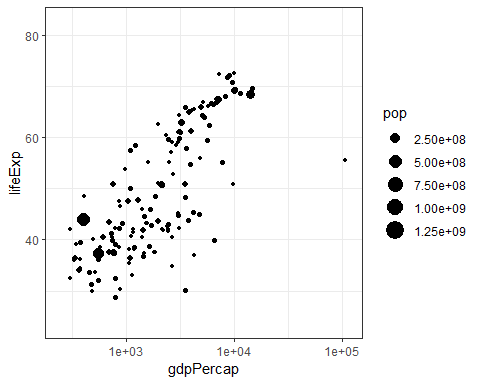
The first step is to create the object-to-be-animated

anim <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point() +  
 scale\_x\_log10() # convert x to log scale  
anim



This plot collates all the points across time. The next step is to split it into years and animate it. This may take some time, depending on the processing power of your computer (and other things you are asking it to do). Beware that the animation might appear in the ‘Viewer’ pane, not in this rmd preview. You need to knit the document to get the viz inside an html file.

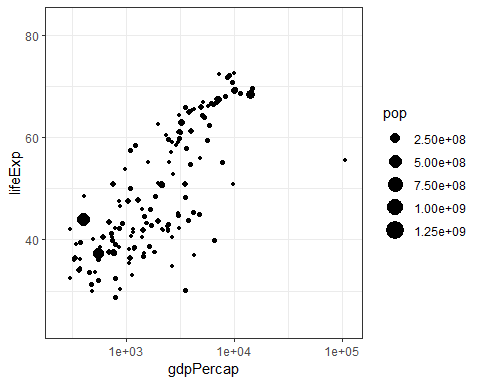
anim + transition\_states(year,   
 transition\_length = 1,  
 state\_length = 1)

 Notice how the animation moves jerkily, ‘jumping’ from one year to the next 12 times in total. This is a bit clunky, which is why it’s good we have another option.

### Option 2 Animate using transition\_time()

This option smoothes the transition between different ‘frames’, because it interpolates and adds transitional years where there are gaps in the timeseries data.

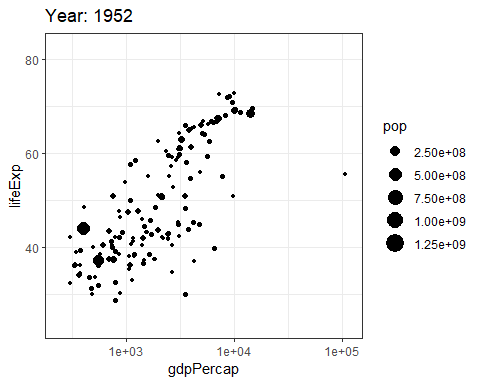
anim2 <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point() +  
 scale\_x\_log10() + # convert x to log scale  
 transition\_time(year)  
 anim2



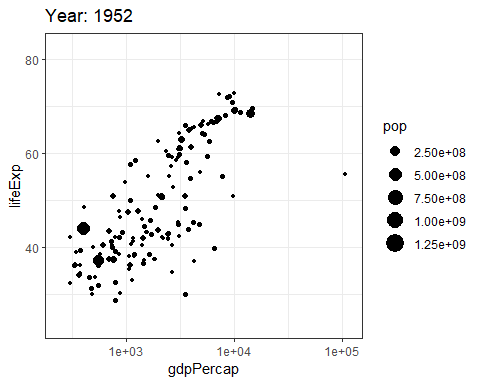
The much smoother movement in Option 2 will be much more noticeable if you add a title to the chart, that will page through the years corresponding to each frame.

Q5 Can you add a title to one or both of the animations above that will change in sync with the animation? [hint: search labeling for transition\_states() and transition\_time() functions respectively]

anim2 <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point() +  
 scale\_x\_log10() + # convert x to log scale  
 transition\_time(year) +  
 labs(title = "Year: {frame\_time}") #Create a title label consisting of Year and the variable "frame\_time" which contains the time. I found the solution here: https://github.com/thomasp85/gganimate/issues/295 and here: https://www.rdocumentation.org/packages/gganimate/versions/1.0.6/topics/transition\_time  
 anim2

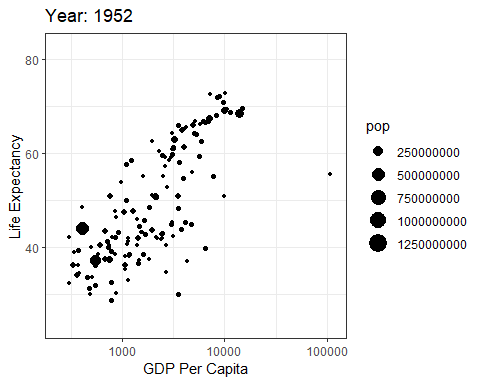


anim + transition\_states(year,   
 transition\_length = 1,  
 state\_length = 1) +  
 labs(title = "Year: {closest\_state}") #Create a title label consisting of the closest state (the year). Solution found by looking at the documentation for transition\_states here: https://www.rdocumentation.org/packages/gganimate/versions/1.0.6/topics/transition\_states

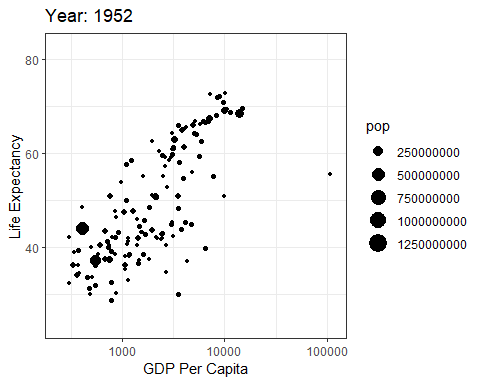


Q6 Can you made the axes’ labels and units more readable? Consider expanding the abreviated lables as well as the scientific notation in the legend and x axis to whole numbers.[hint:search disabling scientific notation]

options(scipen = 999) #Remove scientific notation in the rest of the document. Solution found here: https://statisticsglobe.com/disable-exponential-scientific-notation-in-r  
 anim2 <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point() +  
 scale\_x\_log10() + # convert x to log scale  
 transition\_time(year) +  
 labs(title = "Year: {frame\_time}") + #Create a title label. Consisting of Year and the variable frame\_time which contains the time. I found the solution here: https://github.com/thomasp85/gganimate/issues/295 and here: https://www.rdocumentation.org/packages/gganimate/versions/1.0.6/topics/transition\_time  
 labs(y = "Life Expectancy") + #Change the label for the y-axis to "Life Expectancy"  
 labs(x = "GDP Per Capita") #Change the label for the x-axis to "GDP per Capita".  
 anim2



anim + transition\_states(year,   
 transition\_length = 1,  
 state\_length = 1) +  
 labs(title = "Year: {closest\_state}") + #Create a title label consisting of the closest state (the year). Solution found by looking at the documentation for transition\_states here: https://www.rdocumentation.org/packages/gganimate/versions/1.0.6/topics/transition\_states   
labs(y = "Life Expectancy") + #Change the label for the y-axis to "Life Expectancy"  
labs(x = "GDP Per Capita") #Change the label for the x-axis to "GDP per Capita".



Q7 Come up with a question you want to answer using the gapminder data and write it down. Then, create a data visualisation that answers the question and explain how your visualization answers the question. (Example: you wish to see what was mean life expectancy across the continents in the year you were born versus your parents’ birth years). [hint: if you wish to have more data than is in the filtered gapminder, you can load either the gapminder\_unfiltered dataset and download more at <https://www.gapminder.org/data/> ]

Question: What is the average GDP per capita and life expentancy for each continent? How has these numbers changed historically for each continent?

Answer: First, I need a new obiect. This object needs to group by continent and year. It also need to summarise the gdp and life expectancy according to the year and the continent.

mean\_life\_gdp <- gapminder %>%   
 group\_by(continent, year) %>%   
 summarise(mean\_gdp = mean(gdpPercap),  
 mean\_life\_exp = mean(lifeExp),  
 mean\_pop = mean(pop))

## `summarise()` regrouping output by 'continent' (override with `.groups` argument)

mean\_life\_gdp

## # A tibble: 60 x 5  
## # Groups: continent [5]  
## continent year mean\_gdp mean\_life\_exp mean\_pop  
## <fct> <int> <dbl> <dbl> <dbl>  
## 1 Africa 1952 1253. 39.1 4570010.  
## 2 Africa 1957 1385. 41.3 5093033.  
## 3 Africa 1962 1598. 43.3 5702247.  
## 4 Africa 1967 2050. 45.3 6447875.  
## 5 Africa 1972 2340. 47.5 7305376.  
## 6 Africa 1977 2586. 49.6 8328097.  
## 7 Africa 1982 2482. 51.6 9602857.  
## 8 Africa 1987 2283. 53.3 11054502.  
## 9 Africa 1992 2282. 53.6 12674645.  
## 10 Africa 1997 2379. 53.6 14304480.  
## # ... with 50 more rows

This object is called “mean\_life\_gdp” and it contains the gapminder data, but it groups by year and continent. Then it creates three new columns “mean\_pop” which contains the average population of each continent by year, “mean\_gdp” which contains the average gpd per capita, and “mean\_life\_exp” which contains the average life expectancy.

Now I need to visualize it. I create the object “anim3” which plots the object “mean\_life\_gdp”.The x-axis is the mean\_gdp column, the y-axis is the mean\_life\_exp column and the size of each point is determined by the average population size. Each continent is differentiated by color and each points changes according to the “year” column. The title displays the year, the x-axis is called “Average GDP Per Capita”, the y-axis is called “Average life expectanct”. The scale is logarithmic.

anim3 <- ggplot(mean\_life\_gdp, aes(mean\_gdp, mean\_life\_exp, size = mean\_pop)) +  
 geom\_point(aes(color = continent)) + #Color each continent  
 scale\_x\_log10() + # convert x to log scale  
 transition\_time(year) + #Transition by year  
labs(title = "Year: {frame\_time}") + #Create af title with the current year  
 labs(y = "Average life expectancy") + #Create an y-axis called "Average life expectancy"  
 labs(x = "Average GDP per Capita") #Create a x-axis called "Average GDP per Capita"  
anim3

