HW5 - Make Data Move

05/10/2020

# Explore global development 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’.

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 0.3.5   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

## Look at the data and tackle the tasks

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.

#Basic info on class of variables.  
str(gapminder)

## tibble [1,704 × 6] (S3: tbl\_df/tbl/data.frame)  
## $ country : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ year : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...  
## $ lifeExp : num [1:1704] 28.8 30.3 32 34 36.1 ...  
## $ pop : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 16317921 22227415 ...  
## $ gdpPercap: num [1:1704] 779 821 853 836 740 ...

#Unique years  
unique(gapminder$year)

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

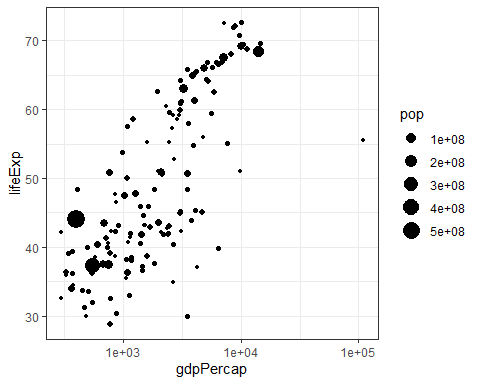
#Get an intuiton of the data structure  
head(gapminder)

## # A tibble: 6 × 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()



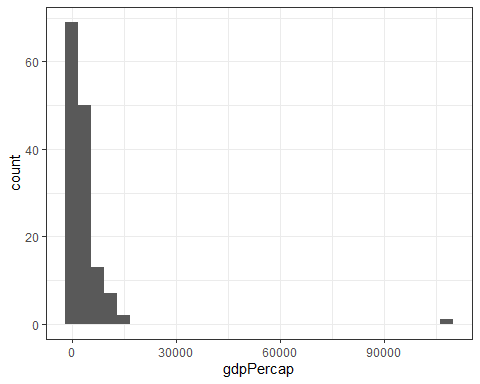
…

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

1. *Why does it make sense to have a log10 scale on x axis?*

#The best way to explain it is to visualize it.  
ggplot(subset(gapminder, year == 1952), aes(gdpPercap)) +  
 geom\_histogram()

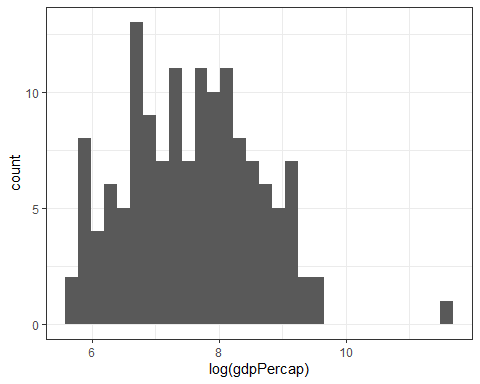
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

 We have a heavily skewed x (gdpPerCap) distribution. It almost looks like a log-normal distribution. (taking the log of the dist would make it normal).

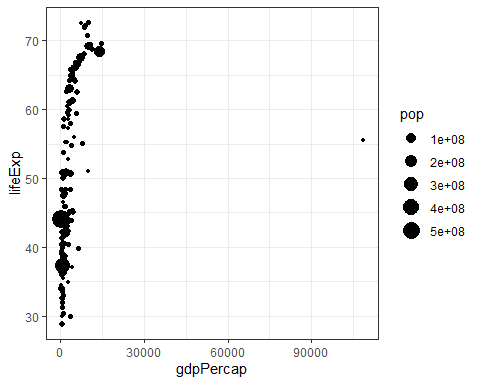
So both for visualization purposes but also for modelling purposes (if you wanna use a generalized linear regression model).

ggplot(subset(gapminder, year == 1952), aes(log(gdpPercap))) +  
 geom\_histogram()

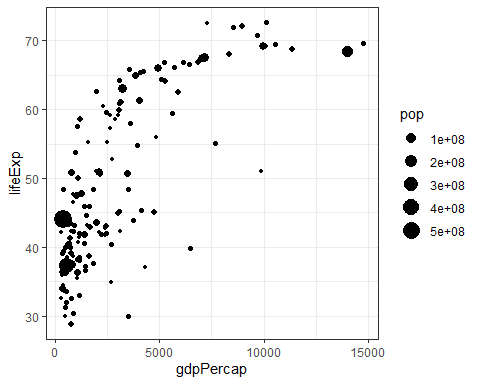
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggplot(subset(gapminder, year == 1952), aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point()



ggplot(subset(gapminder, year == 1952 & gdpPercap < 3e4), aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point()

 While removing the outlier does fix some of our issues it still shows the relationship to be fairly simialir to a logarithmic function. Which we can make linear by taking the log(x). It is indeed very smart.

1. *Who is the outlier (the richest country in 1952 - far right on x axis)?*

#Method1  
gapminder %>%   
 filter(year == 1952) %>%   
 arrange(desc(gdpPercap)) %>%   
 head(1)

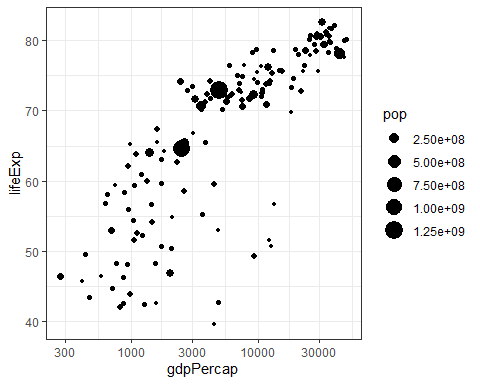
## # A tibble: 1 × 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Kuwait Asia 1952 55.6 160000 108382.

#Method 2 (slighly more elegant and adaptable to other situations.)  
subset(gapminder, year == 1952)[which.max(subset(gapminder, year == 1952)$gdpPercap),]

## # A tibble: 1 × 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Kuwait Asia 1952 55.6 160000 108382.

Next, 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()



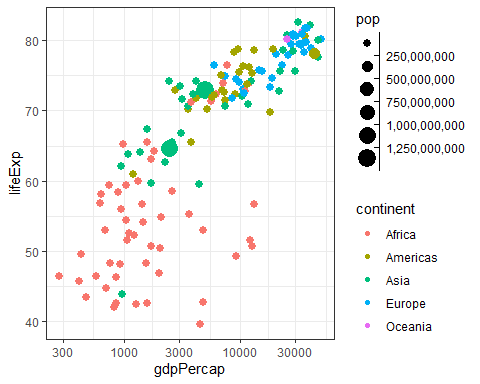
…

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

Tasks:

1. *Differentiate the* ***continents*** *by color, and fix the axis labels and units to be more legible (****Hint****: the 2.50e+08 is so called “scientific notation”, which you might want to eliminate)*

ggplot(subset(gapminder, year == 2007), aes(gdpPercap, lifeExp, size = pop, color = continent)) +  
 geom\_point() +  
 scale\_x\_log10() +   
 scale\_size\_binned(labels = scales::comma)



1. *What are the five richest countries in the world in 2007?*

gapminder %>%   
 filter(year == 2007) %>%   
 arrange(desc(gdpPercap)) %>%   
 head(5)

## # A tibble: 5 × 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Norway Europe 2007 80.2 4627926 49357.  
## 2 Kuwait Asia 2007 77.6 2505559 47307.  
## 3 Singapore Asia 2007 80.0 4553009 47143.  
## 4 United States Americas 2007 78.2 301139947 42952.  
## 5 Ireland Europe 2007 78.9 4109086 40676.

## Make it move!

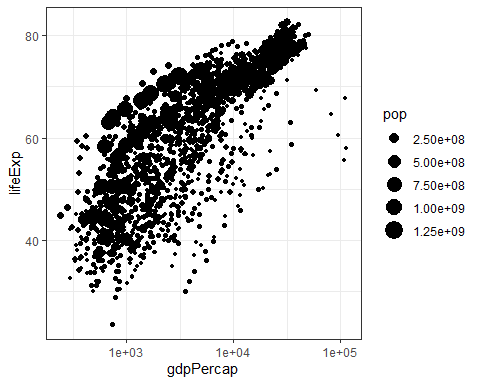
The comparison would be easier if we had the two graphs together, animated. We have a lovely tool in R to do this: the gganimate package. Beware that there may be other packages your operating system needs in order to glue interim images into an animation or video. Read the messages when installing the package.

Also, 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

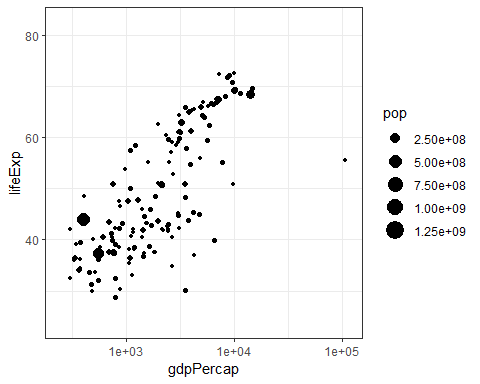
anim <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +  
 geom\_point() +  
 scale\_x\_log10() # convert x to log scale  
anim



…

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 bottom right ‘Viewer’ pane, not in this rmd preview. You need to knit the document to get the visual inside an html file.

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

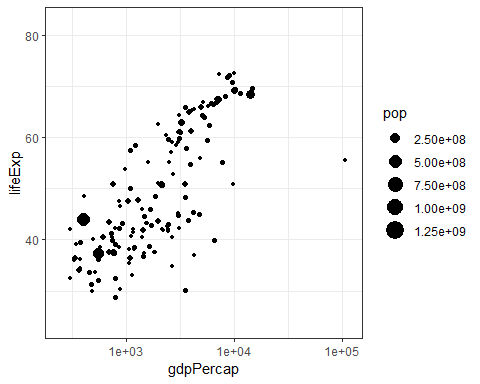
 …

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

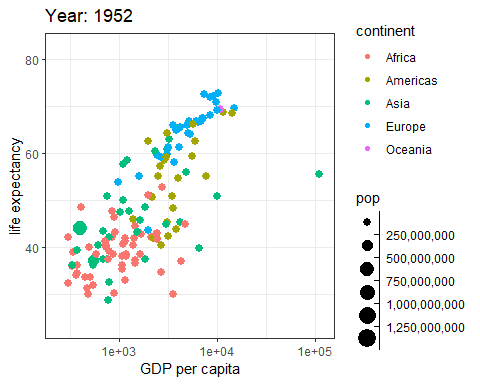


The much smoother movement in Option 2 will be much more noticeable if you add a title to the chart, that will page through the years corresponding to each frame.

Now, choose one of the animation options and get it to work. You may need to troubleshoot your installation of gganimate and other packages

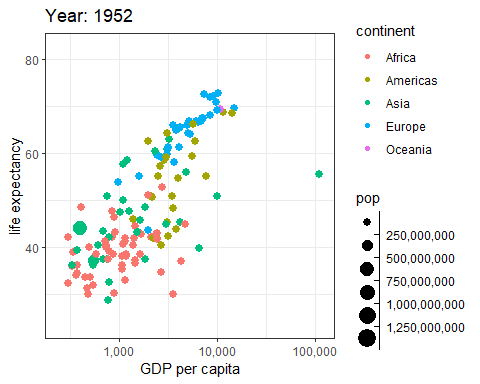
1. *Can you add a title to one or both of the animations above that will change* *in sync with the animation?* *(****Hint****: search labeling for transition\_states() and transition\_time() functions respectively)*

ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, color = continent)) +  
 scale\_x\_log10() +   
 geom\_point() +  
 labs(title = 'Year: {frame\_time}', x = 'GDP per capita', y = 'life expectancy') +  
 scale\_size\_binned(labels = scales::comma) +  
 transition\_time(year)



1. *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.*

ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, color = continent)) +  
 scale\_x\_log10(labels = scales::comma) +   
 geom\_point() +  
 labs(title = 'Year: {frame\_time}', x = 'GDP per capita', y = 'life expectancy') +  
 scale\_size\_binned(labels = scales::comma) +  
 transition\_time(year)



1. *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/*](https://www.gapminder.org/data/) *]*

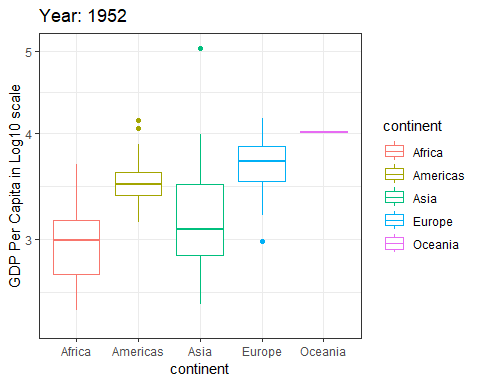
gapminder

## # A tibble: 1,704 × 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.  
## 7 Afghanistan Asia 1982 39.9 12881816 978.  
## 8 Afghanistan Asia 1987 40.8 13867957 852.  
## 9 Afghanistan Asia 1992 41.7 16317921 649.  
## 10 Afghanistan Asia 1997 41.8 22227415 635.  
## # … with 1,694 more rows

#### Question:

How has GDP Per Capita changed in the different continents over time.

ggplot(gapminder, aes(y = log10(gdpPercap), x = continent, col = continent)) +   
 geom\_boxplot() +   
 scale\_y\_log10(labels = scales::comma) +  
 labs(title = 'Year: {frame\_time}', y = "GDP Per Capita in Log10 scale") +  
 transition\_time(year)



ggplot(gapminder, aes(y = gdpPercap, x = continent, col = continent)) +   
 geom\_boxplot() +   
 scale\_y\_log10(labels = scales::comma) +  
 labs(title = 'Year: {frame\_time}', y = "GDP Per Capita Original Scale") +  
 transition\_time(year)

