Pre-programme Assignment 1

YOUR NAME HERE

DATE HERE

Table of Contents

The goal is to test your software installation, to demonstrate competency in Markdown, and in the basics of ggplot.

# R and RStudio installation

You should successfully install R and R studio in your computer. We will do all of our work in this class with the open source (and free!) programming language [R](https://cran.r-project.org/). However, we will use [RStudio](https://www.rstudio.com/) as the main program to access R.

You can find details on how to [install both R and R studio here](http://telapps.london.edu/analytics_with_R/getting-started.html)

## Install tidyverse and gapminder packages

A clean installation of R is known as **base R**. We need to install a couple of packages, namely tidyverse and gapminder. Go to the packages panel in the bottom right of RStudio, click on “Install,” type tidyverse, and press enter. Once it finishes, install gapminder. You’ll see a bunch of output in the RStudio console as all the packages are installed.

You can also just paste and run these two commands

* install.packages("tidyverse")
* install.packages("gapminder")

in the console (bottom left in RStudio) instead of using the packages panel.

You can find details on [R packages here](http://telapps.london.edu/analytics_with_R/packages.html)

## Practice using Markdown

Written assignments will be submitted using [Markdown](https://daringfireball.net/projects/markdown/). Markdown is a lightweight text formatting language that easily converts between file formats. It is integrated directly into [R Markdown](http://rmarkdown.rstudio.com/), which combines R code, output, and written text into a single document (.Rmd).

There is a very nice [Markdwown tutorial](https://commonmark.org/help/tutorial/) that I suggest you go through before working on your assignment. If you want to use a stand-alone Markdown editor [Typora](https://typora.io/) is a lightweight Markdown editor that inherently supports pandoc-flavored Markdown.

## Pandoc

[Pandoc](http://pandoc.org)is a program that converts Markdown files into basically anything else. It was created by [John MacFarlane](https://johnmacfarlane.net), a philosophy professor at the University of California, Berkeley and is widely used as a writing tool and as a basis for publoishing workflow. Kieran Healy’s [Plain Text Social Science workflow](http://plain-text.co) describes how to use Markdown and then convert your Markdown document to HTML, PDF, word, etc.

You should create a file whose name will be your\_name.Rmd, so if I were submitting, my file would be called Kostis\_Christodoulou.Rmd

# Task 1: Short biography written using markdown

You should write within this Rmd file a brief biography of yourself using markdown syntax. I know you have already achieved a lot, but a couple of paragraphs is more than enough.

To achieve full marks, you should include at least 4 of the following elements:

* Headers
* Emphasis (italics or bold)
* Lists
* Links
* Embeding images

Please delete all the intro tet I wrote from line 22 to line 69 and start writing your short biography after this blockquote.

# Task 2: gapminder country comparison

You have seen the gapminder dataset that has data on life expectancy, population, and GDP per capita for 142 countries from 1952 to 2007. To get a glipmse of the dataframe, namely to see the variable names, variable types, etc., we use the glimpse function. We also want to have a look at the first 20 rows of data.

glimpse(gapminder)

## Rows: 1,704  
## Columns: 6  
## $ country <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~  
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~  
## $ year <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~  
## $ lifeExp <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~  
## $ pop <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~  
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~

head(gapminder, 20) # look at the first 20 rows of the dataframe

## # A tibble: 20 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.  
## 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.  
## 11 Afghanistan Asia 2002 42.1 25268405 727.  
## 12 Afghanistan Asia 2007 43.8 31889923 975.  
## 13 Albania Europe 1952 55.2 1282697 1601.  
## 14 Albania Europe 1957 59.3 1476505 1942.  
## 15 Albania Europe 1962 64.8 1728137 2313.  
## 16 Albania Europe 1967 66.2 1984060 2760.  
## 17 Albania Europe 1972 67.7 2263554 3313.  
## 18 Albania Europe 1977 68.9 2509048 3533.  
## 19 Albania Europe 1982 70.4 2780097 3631.  
## 20 Albania Europe 1987 72 3075321 3739.

Your task is to produce two graphs of how life expectancy has changed over the years for the country and the continent you come from.

I have created the country\_data and continent\_data with the code below.

country\_data <- gapminder %>%   
 filter(country == "Greece") # just choosing Greece, as this is where I come from  
  
continent\_data <- gapminder %>%   
 filter(continent == "Europe")

First, create a plot of life expectancy over time for the single country you chose. You should use geom\_point() to see the actual data points and geom\_smooth(se = FALSE) to plot the underlying trendlines. You need to remove the comments **#** from the lines below for your code to run.

# plot1 <- ggplot(data = ??, mapping = aes(x = ??, y = ??))+  
# geom\_??() +  
# geom\_smooth(se = FALSE)+  
# NULL   
  
# plot1

Next we need to add a title. Create a new plot, or extend plot1, using the labs() function to add an informative title to the plot.

# plot 1<- ggplot(data = ??, mapping = aes(x = ??, y = ??))+  
# geom\_??() +  
# geom\_smooth(se = FALSE) +  
# labs(title = " ",  
# x = " ",  
# y = " ") +  
# NULL  
  
  
# print(plot1)

Secondly, produce a plot for all countries in the *continent* you come from. (Hint: map the country variable to the colour aesthetic).

# ggplot(data = , mapping = aes(x = , y = , colour= ))+  
# geom\_?? +   
# geom\_smooth(se = FALSE) +  
# NULL

Finally, using the original gapminder data, produce a life expectancy over time graph, grouped (or faceted) by continent. We will remove all legends, adding the theme(legend.position="none") in the end of our ggplot.

# ggplot(data = gapminder , mapping = aes(x = , y = , colour= ))+  
# geom\_??? +   
# geom\_smooth(se = FALSE) +  
# facet\_wrap(~continent) +  
# theme(legend.position="none") + #remove all legends  
# null

Given these trends, what can you say about life expectancy since 1952? Again, don’t just say what’s happening in the graph. Tell some sort of story and speculate about the differences in the patterns.

Type your answer after this blockquote.

# Task 3: Brexit voting

We will have a quick look at the results of the 2016 Brexit vote in the UK. First we read the data using read\_csv() and have a quick glimpse at the data

brexit\_results <- read\_csv(here::here("Data","brexit\_results.csv"))  
  
  
glimpse(brexit\_results)

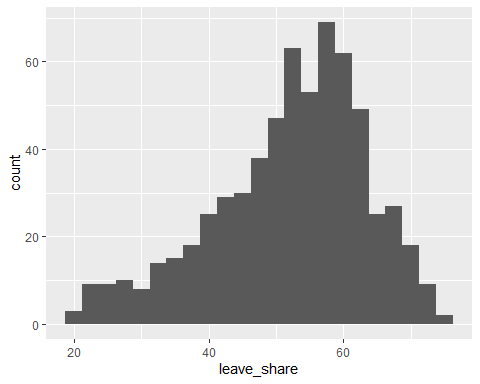
## Rows: 632  
## Columns: 11  
## $ Seat <chr> "Aldershot", "Aldridge-Brownhills", "Altrincham and Sale W~  
## $ con\_2015 <dbl> 50.592, 52.050, 52.994, 43.979, 60.788, 22.418, 52.454, 22~  
## $ lab\_2015 <dbl> 18.333, 22.369, 26.686, 34.781, 11.197, 41.022, 18.441, 49~  
## $ ld\_2015 <dbl> 8.824, 3.367, 8.383, 2.975, 7.192, 14.828, 5.984, 2.423, 1~  
## $ ukip\_2015 <dbl> 17.867, 19.624, 8.011, 15.887, 14.438, 21.409, 18.821, 21.~  
## $ leave\_share <dbl> 57.89777, 67.79635, 38.58780, 65.29912, 49.70111, 70.47289~  
## $ born\_in\_uk <dbl> 83.10464, 96.12207, 90.48566, 97.30437, 93.33793, 96.96214~  
## $ male <dbl> 49.89896, 48.92951, 48.90621, 49.21657, 48.00189, 49.17185~  
## $ unemployed <dbl> 3.637000, 4.553607, 3.039963, 4.261173, 2.468100, 4.742731~  
## $ degree <dbl> 13.870661, 9.974114, 28.600135, 9.336294, 18.775591, 6.085~  
## $ age\_18to24 <dbl> 9.406093, 7.325850, 6.437453, 7.747801, 5.734730, 8.209863~

The data comes from [Elliott Morris](https://www.thecrosstab.com/), who cleaned it and made it available through his [DataCamp class on analysing election and polling data in R](https://www.datacamp.com/courses/analyzing-election-and-polling-data-in-r).

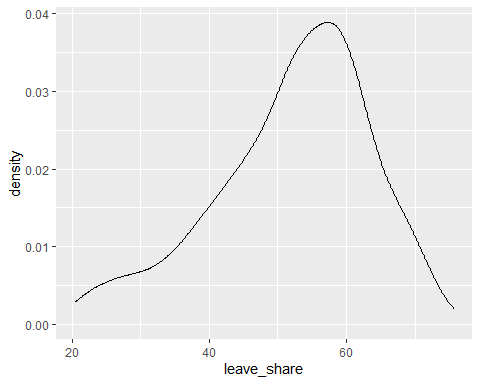
Our main outcome variable (or y) is leave\_share, which is the percent of votes cast in favour of Brexit, or leaving the EU. Each row is a UK [parliament constituency](https://en.wikipedia.org/wiki/United_Kingdom_Parliament_constituencies).

To get a sense of the spread of the data, plot a histogram and a density plot of the leave share in all constituencies.

ggplot(brexit\_results, aes(x = leave\_share)) +  
 geom\_histogram(binwidth = 2.5)



ggplot(brexit\_results, aes(x = leave\_share)) +  
 geom\_density()



One common explanation for the Brexit outcome was fear of immigration and opposition to the EU’s more open border policy. We can check the relationship (or correlation) between the proportion of native born residents (born\_in\_uk) in a constituency and its leave\_share. To do this, let us get the correlation between the two variables

brexit\_results %>%   
 select(leave\_share, born\_in\_uk) %>%   
 cor()

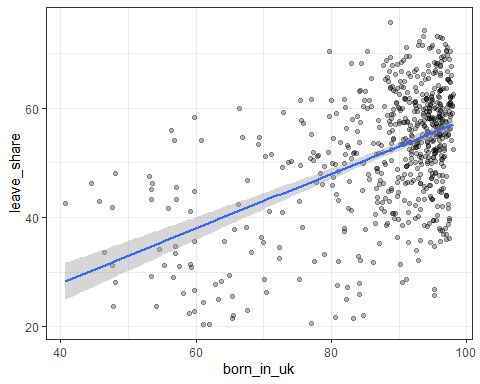
## leave\_share born\_in\_uk  
## leave\_share 1.0000000 0.4934295  
## born\_in\_uk 0.4934295 1.0000000

The correlation is almost 0.5, which shows that the two variables are positively correlated.

We can also create a scatterplot between these two variables using geom\_point. We also add the best fit line, using geom\_smooth(method = "lm").

ggplot(brexit\_results, aes(x = born\_in\_uk, y = leave\_share)) +  
 geom\_point(alpha=0.3) +  
 geom\_smooth(method = "lm") +  
 theme\_bw() +  
 NULL

## `geom\_smooth()` using formula 'y ~ x'



You have the code for the plots, I would like you to revisit all of them and use the labs() function to add an informative title, subtitle, and axes titles to all plots.

What can you say about the relationship shown above? Again, don’t just say what’s happening in the graph. Tell some sort of story and speculate about the differences in the patterns.

Type your answer after, and outside, this blockquote.

# Submit the assignment

Knit the completed R Markdown file as ah HTML or Word document (use the “Knit” button at the top of the script editor window) and upload it to Canvas.

## Details

If you want to, please answer the following

* Who did you collaborate with: TYPE NAMES HERE
* Approximately how much time did you spend on this problem set: ANSWER HERE
* What, if anything, gave you the most trouble: ANSWER HERE