# HKS MPA-ID 2019 Pre-Math Camp Assignment

Due: 2019-08-01 on Canvas

Combined with the assigned primers, these set of exercises get you up and running with basic data analysis in R. We realize that this is asking a lot of you, especially if you are new to programming. The Math Camp program will go over the exercise to clarify any points of confusion.[1]

### Submission

Do your work in the rstudio.cloud environment described below and submit only the saved `.R` file to the Assignment Page in Canvas.[2] More details on how to do this are provided at the beginning and end of this assignment.

### Where are we? Where are we headed?

Before you start this practice problem set, you should have completed, or at least reviewed the RStudio Primers:

- Visualization Basics
- Programming Basics
- Work with Tibbles
- Isolating Data with dplyr
- Creating Variables and dataframes

## Problem 1: Familiarize with the Style Guide

Learning any language requires following its form and style. Throughout the course, we will be enforcing a set of common set of guidelines on how R code should be written. Before writing any code, read and try to internalize Book I ("Analyses") of tidyverse style guide (`https://style.tidyverse.org`), especially chapters 1 and 2.

[1] That said, please feel free to contact Shiro (kuriwaki@g.harvard.edu) if you have any questions in the meantime.

[2] If you can't find the link, the formal link to the assignment is: `https://canvas.harvard.edu/courses/62068/assignments/285490`

## Problem 2: Loading a Spreadsheet in RStudio

The interactive windows in the primers got you started in R, but was also restrictive. Most of your data analysis work will involve programming in R on a designated interface called RStudio. Follow the steps below to get set up and load a dataset.

1. **Create a rstudio.cloud account**: On the internet, go to `https://rstudio.cloud`. Please create a new account for yourself. You will use this account for math camp, so we advise you use your HKS email.

2. **Sign into the Class Space**: Once you have signed in, join the Math Camp "space". By joining this group, you can access R material shared with the group. Use this access link `https://rstudio.cloud/spaces/18236/join?access_code=pR6TvDKi39LKuDHl%2BfltWf2nGC%2Fb0VAk4TZ1Kz5i` and make sure your account is listed as a member.

3. **Copy a Project** In the projects tab, go to the Assignment `01_Summer-Assignment` (Figure 1(a)), and click "Start".

4. **Understanding the GUI and R the program**. It will take 30 seconds to about a full minute for a new window to finish loading (Figure 1(b)). Welcome to RStudio!

   RStudio is a *GUI* (Graphical User Interface) for the programming language R. A GUI allows users to interface with the software using graphical aids like buttons and tabs. Most daily software is a GUI (like Microsoft Word or the Control Panel). RStudio is also an "IDE" (Integrated Development Environment) meaning that it provides shortcuts to advanced tools for working with R.

   The *Console* is the core window through which you can observe R operating (through the GUI). All your results, commands, errors, warnings get shown here. A console tells you what's going on now.

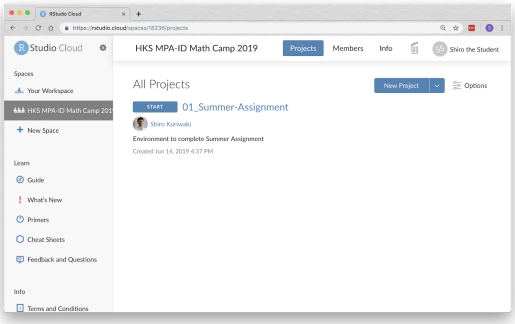
5. **Open a Script**: From the Toolbar's `File`, click to `New File`, then `R Script` (Figure 1(c)). This will create a blank text file with the `.R` file extension. Please finalize your code for this assignment in this file, and submit the saved version (see the end of this assignment for more details). We call this type of file a "script". It is a plain (i.e. no formatting added on) text file with code that is immediately executable.

6. **Read in a Dataset**: Now, let's import a dataset. Here, we'll first rely on the convenience features that the GUI provides.

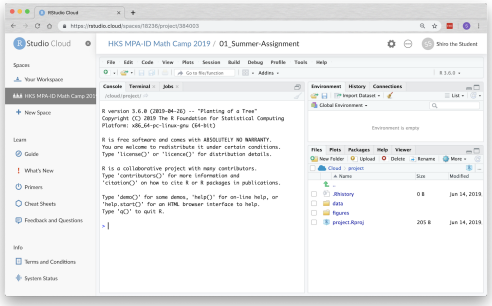
   At the bottom right corner, you should see a "Files" tab. Click through to the folders `data`, then `input`, and click on the filename `WEO-2018.xlsx,"` Choose Import dataset (Figure 1(d)). This starts the process of structuring a piece of R code to read a flat file. One thing you want to change is to make the name of the imported dataset informative, as recommended in the style guide (Figure 1(e)).

   You should see a preview of the spreadsheet and the command that produces it (Figure 1(f)). The bottom-right button, "Import", will send the code directly into the Console.

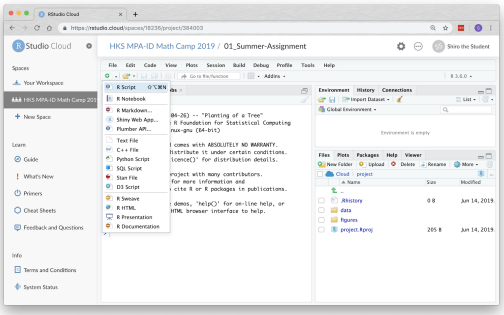
(a) Open the Assignment Project



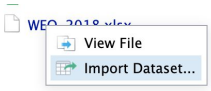

(b) Opened Assignment in the RStudio GUI/IDE




(c) Open a New R Script




(d) Navigate to the data file and Import




(e) Change the assigned name to an informative one


(f) Preview the dataset before completing the Import

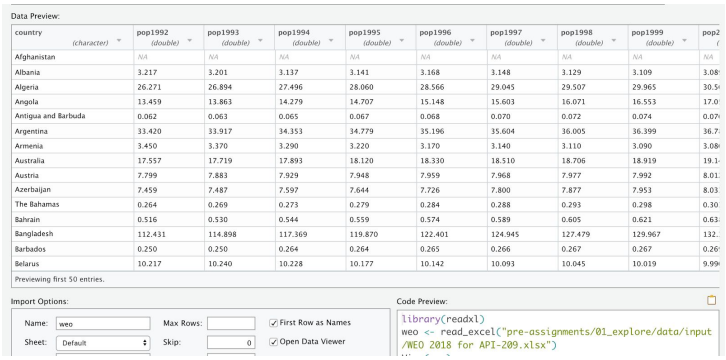



Figure 1: Example Screenshots Corresponding to Problem 2

## Problem 3: Sorting by Values

The following questions are based on the latest version of the World Economic Outlook dataset published by the International Monetary Fund (IMF), which you just read in. Each row in the spreadsheet is a country, with total GDP for a given year adjusted for purchasing power parity (with the `rdgdp` column prefix) and total population (with the `pop` column prefix). GDP values are in millions of 2011 international dollars, so you can directly compare values in different years. Population values are in millions of persons.

**(1)** Write a command (connected by pipes) that (i) first sorts the dataset from lowest to highest real GDP in 2017, and then (ii) outputs a two-column dataset of the country and its GDP.

**(2)** Write a command that is the same as (1) but now sorts it in descending order of 2017 GDP (highest to lowest).

**(3)** The `arrange()` command can sort on more than one variable. To rank countries within their continent, write a command that sorts the countries by continent (in alphabetical order), then by GDP per capita.

**(4)** Write a command that shows African countries in descending order of their 2017 GDP (Use the variable `continent` to filter on African countries).

## Problem 4: GDP per capita

Create a new tibble object called `weo_percep` that is the same as `weo` but also includes:

- A variable called `gdp_percap_2017` that is the country's GDP per capita in 2017,
- A variable called `gdp_percap_1992`, which is the same as above but for 1992, and
- A variable called `growth_2017_1992` which indicates the difference between the two variables above, with positive values indicating growth.

## Problem 5: Graphing

**(1)** Make a scatterplot that shows a countries 1992 GDP per capita on the x-axis and its 2017 GDP per capita on the y-axis.[3]

**(2)** Show the same figure, but coloring the points by continent. That is, countries of the same continent should have the same color.

[3] You might notice that the scatterplot itself is not as informative as it could be. In math camp, we will spend a session discussing the nuts and bolts of making a high-quality graphic that is informative and user-friendly.

**(3)** Show the same figure, but assigning a different shape of point for different continents. Also, set the color to all offices to navy by using the color label `"navy"`.

## Problem 6: Mean and Median

**(1)** Write code that reports the mean of country-level GDP per capita of 1992 in one column and the mean for 2017 in another. Make sure that column names are self-explanatory.

**(2)** Do the same, but showing the median instead of the mean. Make sure to ignore any missing values in the calculation of the median, so a value is returned.

## Problem 7: slice() and filter()

Much of data analysis is understanding how new functions work through reading the documentation and experimentation. The function `slice()` is part of the tidyverse and allows you to filter rows by their position. Notice that `slice()` and `filter()` are similar in that they subset rows of a dataset, but differ in the types of input they require — the former asks for positions, the latter asks for conditions.

Check out the help page of `slice`. Then, write a command that shows the countries with the top three and bottom three 2017 GDPs, thereby combining the output in the first two R exercises.

## [Optional] More Graphing

Make a graph like the one shown in Figure 2. Follow both the graphical components of the graph shown as you see them, as well as the description of the measures as described in the Figure caption. *Note*: This problem is a challenge problem, and involves some commands not covered in the primers.

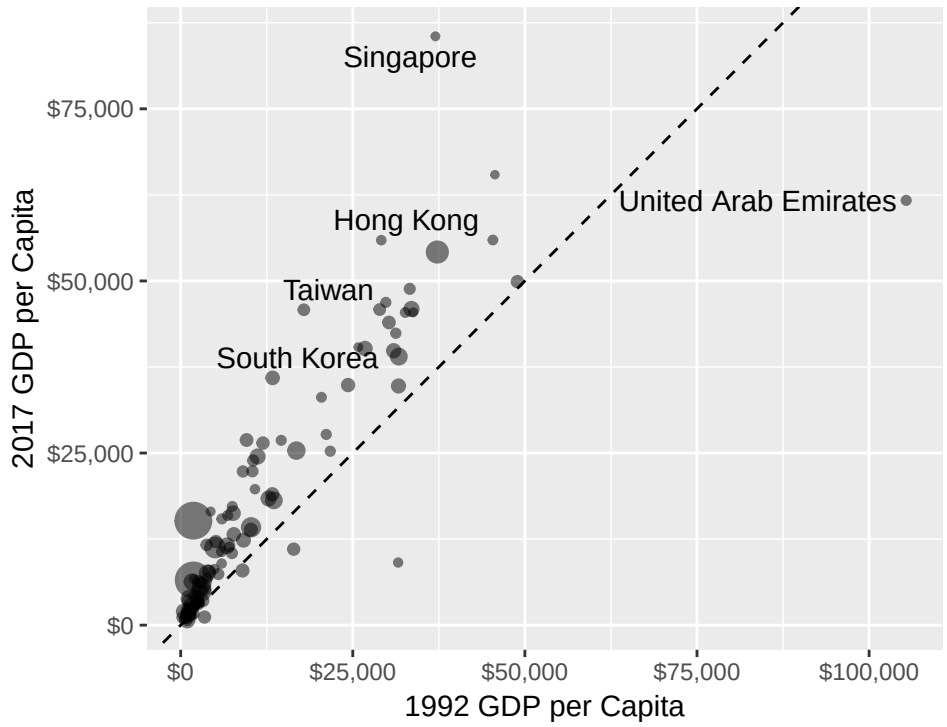



Figure 2: Changes in GDP per capita between 1992 and 2017.

## Submitting

Once you have completed or made an attempt for all the problem, please clean up your R script, download it from the cloud, and submit it to Canvas.

Math camp instructors will check and provide comments for your code. You should follow these guidelines to clean up your final submission (and should do so for all future scripts):

- Delete any failed attempts or duplicative code.
- Label the relevant question number by comment (e.g., `## Problem 1.1`. Follow the style guide for the exact format)
- Try restarting (Toolbar `Session` > `Restart`) and running your entire code at once (e.g., Select All Text and Run, or `Run All` by `option` + `command` + `R`. This ensures that your code is replicable.
- Follow other guidelines from the style guide, such as putting the `library()` command at the beginning of the script.
- To help us sort through all submissions, please name your script with your last name. e.g., `kuriwaki_assignment.R`.

After editing your code, save it to the main project folder, and then download it by right-clicking the file icon (in the File Pane), and selecting `Export` (Figure 3). Download the script and attach it to your Canvas submission.

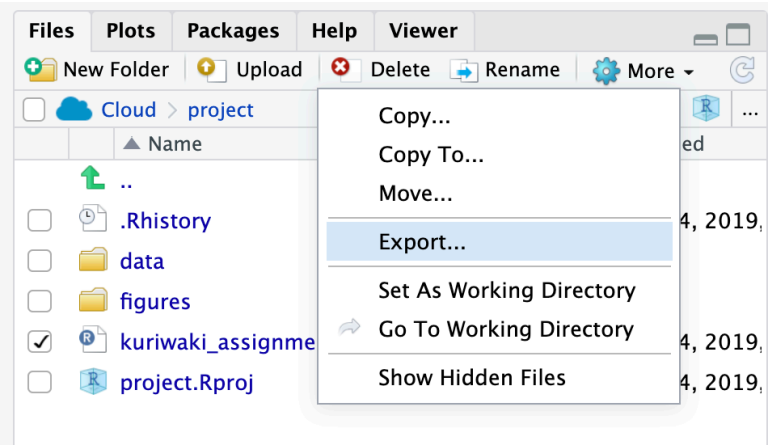



Figure 3: Downloading your final script