

R Assignment

Intensive Statistics Course

Wihan Marais

wihanmarais@sun.ac.za | github.com/WihanZA

26 January 2022

Task 1

Create a new project in which you will complete the assignment.

File > New project > New directory > New project > Choose directory location and name

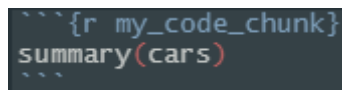
Once you have started operating from your new root folder, create a new folder named “data” in your directory. Now, generate a GitHub Document in which you will present your work and save it in your root folder.

File > New file > R Markdown > From template > GitHub Document

This is known as README file which is typically used to describe the contents and functioning of GitHub repositories or packages. Change the **title** of your new document to your student number. To generate and view the output of this R Markdown file, click the Knit button towards the top of your workspace, or Ctrl + Shift + K.

Task 2

As you can see in the R Markdown file, headings and subheadings are indicated with **##** and **###**. Furthermore, the body of text is interspersed with so-called “code chunks”. This is where you will be typing and executing your code. To create a new code chunk at the location of your cursor, click “Insert new code chunk” at the top of your screen, or Ctrl + Alt + I. For example:



```
```${r my_code_chunk}
summary(cars)
```
```

The ````` indicates the beginning and end of a code chunk. `r` refers to the type of programming language the chunk should expect. `my_code_chunk` refers to the name that you have given the relevant code chunk.

In the `setup` code chunk at the start of the page, you are able to set the default settings for all code chunks’ display and output in the final document. Adjust yours to reflect:

```
knitr::opts_chunk$set(echo = TRUE, eval = TRUE, fig.height = 3.5, fig.width = 6)
```

This tells R Markdown to show both code and output in your final document (`echo`), execute each code chunk (`eval`), and to set the dimensions of your figures to 3.5×6 inches by default.

To complete the setup, load the `pacman` package using `library()` and use `pacman::p_load()` in the `setup` code chunk to install/load the following packages:

- tidyverse
- huxtable
- fixest
- urca

Task 3

Copy the `ts_data.csv` file that I have provided you into your “data” folder. In a new code chunk, read the `.csv` file and store it as data frame in your Global Environment. Use the **tidyverse** syntax (aka piping) and `mutate()` to change the “Year” column to the class **date**. Similarly, subset the data frame to contain observations for France between the years 1985-2010. For the latter, pipe the following

```
%>% dplyr::filter(Year >= as.Date("1985-01-01"), Year <= "2010-01-01")
```

Make sure the data frame is arranged chronologically and perform an Augmented Dickey-Fuller test on the time series, `ln_consump_pc`.

Task 4

Copy the `cs_data.csv` file that I have provided you into your “data” folder. In a new code chunk, read the `.csv` file and store it as data frame in your Global Environment. Using the **fixest** package, perform the following three OLS regressions

- wage on `x1`
- wage on `x1` and `x2`
- wage on `x1` and `x2` and the categorical variable `treat`

Save each of these models and present them in a regression table using **huxreg**.

Conclusion

Knit your completed R Markdown file and send the final `.Rmd` file to wihanmarais@sun.ac.za for grading.