Introduction to R: Assignment

2025-01-15

Contents

1 Tasks for Submission

Please complete all tasks and submit your R code to me at `fkreuser@sun.ac.za`. Note, that all directories must take relative values from your working directory. I will execute your code on my local machine using a fresh installation of R.

1.1 Task 1 - Create and Environment

- Open R Studio and create a new project in which you will complete the assignment.
- Create a new R Script and save it as `[yourstudentnumber].R` in your directory. Write and execute the code necessary to complete the rest of the assignment in this R Script.

  Hint: Run `getwd()` in your console to check the file path of your working directory.

1.2 Task 2 - Load Packages

- Using comments, create headings in your script for each of the subsequent tasks. Write your code under the appropriate headings.

  Hint: In `.R` files, comments are created using the pound sign, i.e., `#`.
- Load the `pacman` package using `install.packages("")` (if necessary) and `library()`.
- Install/load the following packages using the `pacman` package:
  - `tidyverse`
  - `huxtable`
  - `fixest`
  - `readxl`
  - `tsibble`
  - `mFilter`

1.3 Task 3 - Create a Directory

Create a folder named `data` in your current directory using R.

1.4 Task 4 - Load Data

1.4.1 If your antivirus does not block you:

1. Create a variable called `url` that takes the value below:

```
https://www.columbia.edu/~mu2166/book/empirics/usg_data_annual.xls
```

2. Complete the following path object, you can use the file name `usg_data_annual.xls`:

```
file_path <- file.path([directory where data is to be stored], [file name])
```

3. Download the file by completing the code below - note that you must call R-Objects to complete this code. Comment on what your code does.

```
download.file([object for file address on internet] ,
              [object for file path once downloaded],
              mode = "wb")
```

4. Read the data into a dataframe using read_excel.

1.5 Task 5 - Getting Unique Values and Creating Your Dataset

1. Create an array called `countries`, it should keep only the `Country Name` row from the data you imported
2. Replace the `countries` array with an array that contains only `unique()` values
3. Load the function below into R and then include replace [123456789] with your student number.

```
# Function to assign you a random country based on your student number
my_countries <- function(studentnumber, array) {
  set.seed(studentnumber)

  # Fixed first two numbers
  country_index <- c(204, 174)

  # Generate the third number, ensuring it is not 204 or 174
  repeat {
    third_number <- sample(1:214, 1)  # Generate a random number
    if (third_number != 204 && third_number != 174) break  # Exit loop if valid
  }

  # Add the third number to the list of indices
  country_index <- c(country_index, third_number)

  # Get the corresponding countries
  selected_countries <- array[country_index]

  # Create a dataframe with the selected countries and their indices
  return_countries <- data.frame(Index = country_index, Country = selected_countries)

  return(return_countries)
}
keep_countries <- my_countries(123456789, countries)
```

4. Merge `keep_countries` to `usg_data` using the data generated above. Call this new data frame `raw_countries`

1.6 Task 6 - Making the Data Useable

You will see that your dataset is unfortunately quite unusuable. Let's fix that by reshaping the data to be in the standard panel format.

1. Run the code below - provide comments on what each of the lines do and why we would want to do them.

```
long_data <-raw_countries %>%
  pivot_longer(
    cols = `1960`:`2011`,
    names_to = "Year",
    values_to = "Value"
  )
```

2. Run the code below - provide comments on what each of the lines do and why we would want to do them.

```
long_data <- long_data %>%
  select(-`Indicator Name`) %>%
  distinct() %>%
  pivot_wider(
    names_from = `Indicator Code`,
    values_from = Value
  )
```

I include code with which you can get labels for the variables in the dataset as well. You don't have to comment on this, but it would help if you run this.

```
# Attach custom attributes to store Indicator Name as metadata
for (code in unique(raw_countries$`Indicator Code`)) {
  # Find the corresponding indicator name
  label <- unique(raw_countries$`Indicator Name`[raw_countries$`Indicator Code` == code])

  # Assign the label as an attribute to the corresponding column
  attr(long_data[[code]], "label") <- label
}
```

3. Convert the `Year` variable of `long_data` to a number
4. Convert your data into a time-series tibble using the code below.

```
long_data <- long_data %>%
  as_tsibble(index = Year, key = `Country Name`)
```

5. In the dataset that you have now created, change the name of the field `NY.GDP.PCAP.KN` to `gdp_pcap`, the variable must retain its label without a label having to be explicitly added. The name of the dataframe must not change.
6. Create the following variables in the same dataframe:

- Create a variabled called `cons_pcap` that contains the Consumption per Capita in constant local currency units.
- Create a variable called `gdi_pcap` that contains the Investment per Capita in constant local currency units.
- Ensure that your series are the correct order of magnitude.

7. Combine tasks 1 to 6, with the comments you made, into a signle tidyverse pipe.
8. Create a data frame called `data_to_use` that includes the country name, country code, year, as well as the three per capita measures you created above. Ensure that `data_to_use` has no rows with empty values.
9. Create logged values of the national accounts variables. Call these variables `l_[name]`.

1.7 Task 7 - Using your data

1. Provide a table in the form of the one below based on your data.

Table 1: Descriptive Statistics of GDP per Capita and Year Range by Country

Country Name	Mean	StdDev	Observations
Macedonia, FYR	101,496	12,712	22
South Africa	30,679	3,623	52
United States	30,847	8,034	47

Hint: Use a tidyverse pipe that looks something like this and at some point use huxtable.

```
summary_stats <- data_to_use %>%
  as_tibble() %>% # Temporarily remove the tsibble structure
  group_by(...) %>%
  summarise(...)
```

2. Run the regressions below on the `United States` only. In all questions use the `fixest` command.

2.1. $l\_gdp\_pcap_t = \rho_0 + \tau Year_t$

2.2. $l\_gdp\_pcap_t = \rho_0 + \tau Year_t + \rho_1 l\_gdp\_pcap_{t-1}$

2.3. $\triangle l\_gdp\_pcap_t = \rho_0 + \tau Year_t + \rho_1 l\_gdp\_pcap_{t-1}$

```
## NOTE: 1 observation removed because of NA values (RHS: 1).
```

```
## NOTE: 1 observation removed because of NA values (LHS: 1, RHS: 1).
```

2.4. Combine all tables into a huxreg that looks similar to the one the next page below, change the names of variables to reflect their economic meaning. Add any other aesthetic improvements you please.

2.5. Given your answer in 2.4., would you say the data is stationary?

	(1)	(2)	(3)
(Intercept)	-28.112 ***	-3.960	-3.960
	(0.694)	(3.078)	(3.078)
Year	0.019 ***	0.003	0.003
	(0.000)	(0.002)	(0.002)
l_gdp_pcap_lag		0.835 ***	-0.165
		(0.106)	(0.106)
N	47	46	46
R2	0.986	0.994	0.106
logLik	95.443	114.296	114.296
AIC	-186.886	-222.593	-222.593

*** p < 0.001; ** p < 0.01; * p < 0.05.

2 Not for Submission

2.1 Task 8

1. Below, I create the linearly detrended series for `l_gdp_pcap` as `l_gdp_pcap_ldt`. Create the quadratically detrended series and comment on your code. (Hint: you can specify `poly(Year, n)` to create a polyniomial of order `n` in the `linear model` environment. Ex. `lm(y ~ poly(x,n))`)

```
data_to_use <- data_to_use %>%
  group_by_key() %>%
  mutate(
    # Linear detrending
    l_gdp_pcap_ldt = residuals(lm(l_gdp_pcap ~ Year, data = cur_data())),
  ) %>%
  ungroup()
```

2. Below, I detrend the data using the HP-Filter using `\lambda = 100` for all series, do the same for `lcons_pcap`. Create the same for all the series using the `\lambda=6.25`.

```
hp_data <- data_to_use %>%
  mutate(across(
    c(l_gdp_pcap, l_cons_pcap),
    ~ mFilter::hpfilter(.x, freq = 100)$cycle
  ))
```

3. Confirm whether the HP filtered series above are stationary.