R-work for Online Assessment

Instructions

You should work through the code below and complete it. Keep the completed code and all the resulting output. Next you should answer the questions in the online quiz. Every student will see a slightly different collection of questions (as we will randomly draw 10 questions from a pool of about 20 questions).

The questions are of four types.

1) Questions that merely ask you to report output from your analysis.
2) Some questions will ask you about R code. For example, you will see a lot of gaps (`XXXX`) in the code and questions may ask you how to complete the code to make the code work. Sometimes the `XXXX` will represent one word and on other occasions it will represent a full line (or two) of code. Other questions may ask you about the output to be produced by a particular bit of code.
3) The third type of questions will test your understanding of econometric issues. For example: "What is the meaning of an estimated coefficient?" "Is a particular coefficient statistically significant?"
4) The fourth type of question, if asked, will be on general programming issues. For example: what is the meaning of a particular error message, or, how would you search for a particular piece of information.

Preparing your workfile

We add the basic libraries needed for this week's work:

```
library(tidyverse)    # for almost all data handling tasks
library(ggplot2)      # to produce nice graphiscs
library(stargazer)    # to produce nice results tables
library(AER)          # access to HS robust standard errors
library(knitr)
source("stargazer_HC.r")  # includes the robust regression display
```

Introduction

The data are a database listing of global power generation plants with a range of information (like location, fuel type). Do read through Sections 1 to 4 of the A_Global_Database_of_Power_Plants.pdf file which contains the database's documentation.

This is the data source: L. Byers, J. Friedrich, R. Hennig, A.,Kressig, Li X., C. McCormick, and L. Malaguzzi Valeri. 2019. "A Global Database of Power Plants." Washington, DC: World Resources Institute. Available online at <www.wri.org/publication/globalpowerplantdatabase>.

There is no real need for you to access this original source. The datafile and the documentation is provided.

Data Upload - and understanding data structure

Upload the data, which are saved in a csv.

```
data_plants <- XXXX("global_power_plant_database.csv")
data_plants <- as.data.frame(XXXX)    # ensure data frame structure
names(XXXX)
```

```
data_plants <- read_csv("global_power_plant_database.csv")
data_plants <- as.data.frame(data_plants)    # ensure data frame structure
names(data_plants)
```

```
##  [1] "country"                "country_long"
##  [3] "name"                   "gppd_idnr"
##  [5] "capacity_mw"            "latitude"
##  [7] "longitude"              "primary_fuel"
##  [9] "other_fuel1"            "other_fuel2"
## [11] "other_fuel3"            "commissioning_year"
## [13] "owner"                  "source"
## [15] "url"                    "geolocation_source"
## [17] "wepp_id"                "year_of_capacity_data"
## [19] "generation_gwh_2013"    "generation_gwh_2014"
## [21] "generation_gwh_2015"    "generation_gwh_2016"
## [23] "generation_gwh_2017"    "estimated_generation_gwh"
```

As you upload the data you may get some warning messages, in particular regarding "parsing failures". Please ignore these messages. Ensure that you have 29910 observations and 24 variables.

Let us look at a particular observation so we can understand the data

```
data_plants[7299,]
```

```
##      country country_long             name  gppd_idnr capacity_mw latitude
## 7299     CHN        China Three Gorges Dam WRI1000452       22500  30.8235
##      longitude primary_fuel other_fuel1 other_fuel2 other_fuel3
## 7299  111.0032        Hydro        <NA>        <NA>          NA
##      commissioning_year owner                        source
## 7299               2003  <NA> China Three Gorges Corporation
##                                                  url geolocation_source
## 7299 http://www.ctgpc.com.cn/sx/sxgczds.php?mClassId=015004               <NA>
##      wepp_id year_of_capacity_data generation_gwh_2013 generation_gwh_2014
## 7299 1012216                    NA                  NA                  NA
##      generation_gwh_2015 generation_gwh_2016 generation_gwh_2017
## 7299                  NA                  NA                  NA
##      estimated_generation_gwh
## 7299                 92452.57
```

This is the famous Three Gorges Hydro Power plant in China. You can see that for each power plant we have the country information (in fact we also have the exact latitude and longitude) and we know its capacity (`capacity_mw`) measured in mega watts (MW). It is 22,500 MW and it is the largest Hydro plant in the world. The `primary_fuel` variable indicates how electricity is being generated. This particular power plant generates electricity using hydro (water) power.

Let us ensure that categorical variables are stored as `factor` variables. It is easiest to work with these in R. In particular the `primary_fuel` variable should be defined as a factor variable.

```
data_plants$primary_fuel <- XXXX
```

```
data_plants$primary_fuel <- as_factor(data_plants$primary_fuel)
```

Task 1

Find out what the 10 largest hydro plants are named, in which countries they are and what their generation capacity are (i.e. create a Top 10 League Table).

```
task1 <- data_XXXX %>%  filter(primary_fuel == XXXX) %>%
            select(country, XXXX, XXXX) %>%  arrange(desc(XXXX))
kable(task1[1,10,]) # in R %>% print() will show ok
```

You should find the Three Gorges Dam at the top of your table.

```
task1 <- data_plants %>%  filter(primary_fuel == "Hydro") %>%
            select(country, name, capacity_mw) %>%  arrange(desc(capacity_mw))
kable(task1[1,10,]) # in R %>% print() will show ok
```

|| || || ||

Create a similar table for the top 10 largest nuclear power plants.

```
task1 <- data_plants %>%  filter(primary_fuel == "Nuclear") %>%
            select(country, name, capacity_mw) %>%  arrange(desc(capacity_mw))
kable(task1[1,10,],format = "html") # in R %>% print() will show ok
```

Task 2

Create a new variable called `renewable`. This should take the value "renew" for any power plant which produces electricity using hydro, wind, solar, biomass, wave and tidal or geothermal. All other generation types should get a "non_renew" in the `renewables` variable.

We did something similar in Demo Class 1 (using the `fct_recode` function)

```
data_plants <- data_plants %>%
    mutate(renewable = fct_recode(primary_fuel,
    "renew" = "Hydro",    # new level = old level
    "non_renew"  = "Gas",
    "non_renew"  = "Other",
    "non_renew"  = "Oil",
    "renew" = "Wind",
    "non_renew"  = "Nuclear",
    "non_renew"  = "Coal",
    "renew" = "Solar",
    "non_renew"  = "Waste",
    "renew" = "Biomass",
    "renew" = "Wave and Tidal",
    "non_renew"  = "Petcoke",
    "renew" = "Geothermal",
    "non_renew"  = "Cogeneration",
    "non_renew"  = "Storage"))
```

You have done it right if you can replicate this output, meaning that there are 19867 generation unit entries which use renewable sources.

```
summary(data_plants$renewable)
```

```
##     renew non_renew
##     19867     10043
```

The generation capacity of a power plant is measured in megawatt (MW) (and included as the variable `capacity_mw` in the dataset). Sometimes capacity is also reported in gigawatt (GW). Add a new variable, `capacity_gw`, to `data_plants` which measures a generation unit's capacity in GW. You will have to find out how to translate MW to GW.

```
# either of the following
data_plants <- data_plants %>%  mutate(capacity_gw = capacity_mw/1000)
data_plants$capacity_gw <- data_plants$capacity_mw/1000
```

After creating `cpacity_gw` use the `summary` function to create some summary statistics for this new variable. You should find the maximum capacity (in GW) to be 22.5 and the mean capacity to be 0.1863 GW.

```
summary(data_plants$capacity_gw)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.001000  0.004774  0.018900  0.186295  0.100000 22.500000
```

Task 3

Let's calculate the capacity of nuclear power stations in the USA (in GW)?

```
data_plants %>%  filter(primary_fuel == "Nuclear", country == "USA") %>%
  summarise(sum(capacity_gw))
```

```
##   sum(capacity_gw)
## 1         106.1482
```

Now produce a table which calculates the capacity of nuclear, coal, hydro, wind, solar and gas powered electricity generation for the following countries (Brazil, USA, UK, China). Meaning for each of these countries you want a capacity for nuclear power generation capacity, another for hydro, another for wind, etc. As usual there are several ways to achieve the same but if you use piping in combination with `filter`, `group_by` and `summarise` you can use techniques which

```
sel_countries <- c("BRA",XXXX)
sel_fuel <- c("Nuclear",XXXX)
Table2 <- data_plants %>%
  XXXX(XXXX %in% XXXX, XXXX %in% XXXX) %>%
  group_by(country,primary_fuel) %>%
  XXXX(cap = XXXX(XXXX)) %>%
  spread(country,cap) %>%  print()   # the spread(country,cap) part is optional but nice
```

```
sel_countries <- c("BRA", "USA", "GBR", "CHN")
sel_fuel <- c("Nuclear","Coal", "Hydro", "Wind", "Solar", "Gas")
Table2 <- data_plants %>%
  filter(country %in% sel_countries, primary_fuel %in% sel_fuel) %>%
  group_by(country,primary_fuel) %>%
  summarise(cap = sum(capacity_gw)) %>%
  spread(country,cap) %>% print()
```

```
## # A tibble: 6 x 5
##   primary_fuel      BRA    CHN   GBR   USA
##   <fct>           <dbl>  <dbl> <dbl> <dbl>
```

```
## 1 Hydro        98.0      259.    4.12 101.
## 2 Gas          11.3       59.8  29.9  526.
## 3 Wind         10.3       51.0  17.6   88.5
## 4 Nuclear       1.99      33.4   8.92 106.
## 5 Coal          2.79     956.   12.3  283.
## 6 Solar         0.00657    3.02  8.10  27.4
```

You have it correct if you find that the capacity of gas fired electricity generation in Brazil is approximately 11.286 GW.

Merge with other data

Let's load a few country indicators:

```r
country_ind <- read_csv("CountryIndicators.csv", na = "#N/A")
```

Check out the names of the variables.

```r
names(country_ind)
```

```
## [1] "country"        "geoID"          "Land_Area_sqkm" "HealthExp"
## [5] "GDPpc"          "population"
```

The country indicator in this file, `geoID` is a two letter code, but the country indicator in `data_plants`, the variable `country` is a three letter code. We need a common country code so that we can match up the two data files. You should always try to use some such code rather than the actual country names as there are too many variations in country names which may prevent the merging functioon from merging data. We have learned in a previous project, when dealing with Covid-19 data, that we can use a little function to translate between the two different country codes.

```r
library(countrycode)
country_ind$country <- countrycode(country_ind$geoID, origin = "iso2c", destination = "iso3c")
```

Now we are in a position to merge to data as `data_plants$country` and `country_ind$country` both contain the three letter country codes.

Task 4

Merge the two data files `data_plants` and `country_ind` using the three letter country codes. Bring the data together in a new data file called `data_combined`.

```r
data_combined <- XXXX(data_plants,XXXX,all.x = TRUE)
```

```r
data_combined <- merge(data_plants,country_ind,all.x = TRUE)
```

Use the `stargazer` function to calculate summary statistics for the following variables: `capacity_gw`, `commissioning_year`. Here is an example of how to use the `stargazer` function to calculate summary statistics.

```r
stargazer(data_combined[,c("capacity_mw","estimated_generation_gwh")],type = "text")
```

```
##
## =================================================================================
## Statistic                  N     Mean    St. Dev.   Min  Pctl(25) Pctl(75)    Max
## ---------------------------------------------------------------------------------
## capacity_mw              29,910 186.295  525.704    1      4.8      100      22,500
```

```
## estimated_generation_gwh 21,791 847.036 4,067.435 0.000  10.083  339.874  450,562.700
## ----------------------------------------------------------------------------------------
```

```
stargazer(data_combined[,c("capacity_gw","commissioning_year")],type = "text")
```

```
##
## =======================================================================================
## Statistic            N      Mean    St. Dev.    Min    Pctl(25)  Pctl(75)     Max
## ---------------------------------------------------------------------------------------
## capacity_gw        29,910   0.186    0.526     0.001     0.005     0.100    22.500
## commissioning_year 16,303 1,995.486  23.526  1,896.000 1,986.000 2,012.064 2,018.000
## ---------------------------------------------------------------------------------------
```

You should find, for instance, that the mean value of the `capacity_gw` is 0.186 and that there are 16,303 observations (N) for the variable `commissioning_year`.

Clearly information about the commissioning year is missing for many generation units. Furthermore, when you investigate the values that most values are given as full year values, e.g. 2012, but for some observations you get values like 1966.808. What this means is that this particular power plant was commissioned some time in autumn of 1966. For now we are only interested in the full year information. For that purpose we will only use the full number information.

```
data_combined$commissioning_year <- floor(data_combined$commissioning_year)
```

Use the help or a search engine to figure out what the `floor` function and its sister function `ceil` do.

Task 5

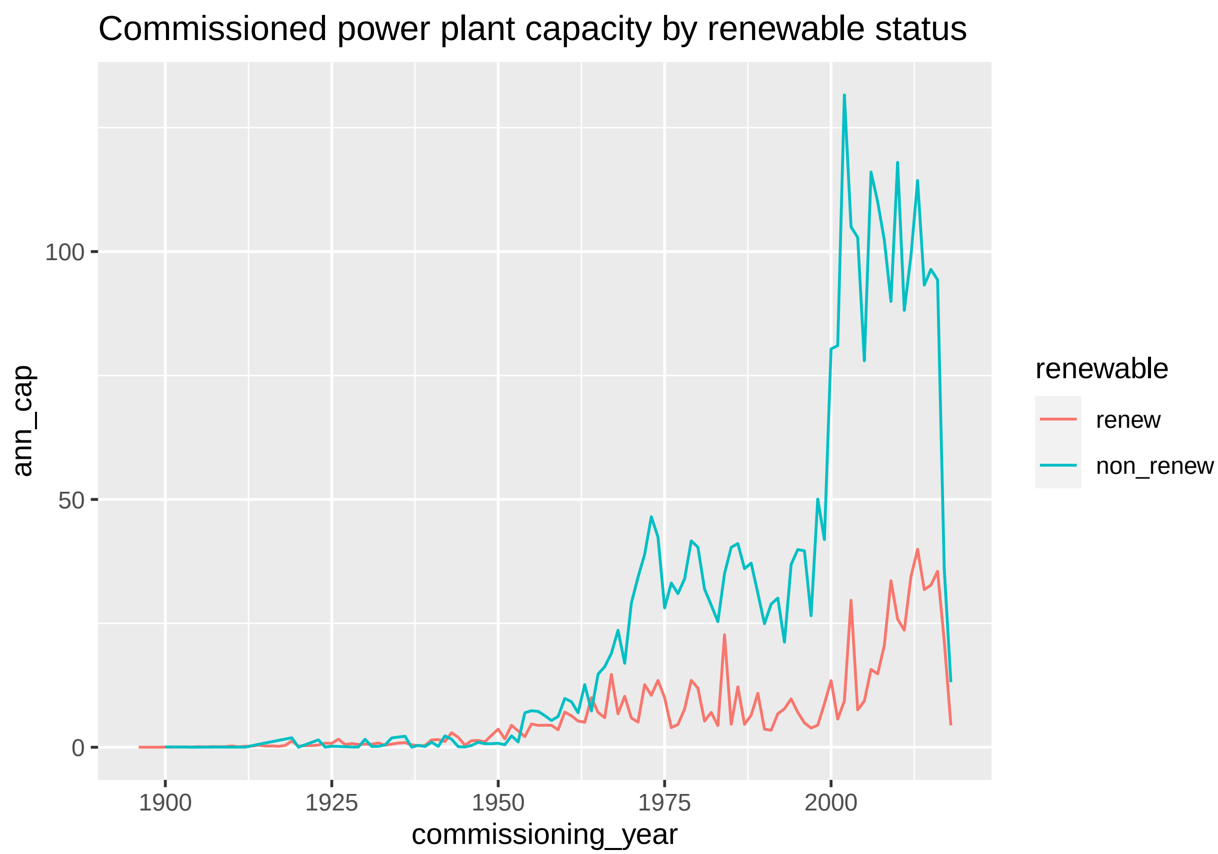
We want to investigate whether, in more recent years, more renewable capacity is being installed. The `commissioning_year` variable indicates in what year a particular generator has been installed. Let's create annual data representing the freshly commissioned capacity in a particular year.

```
new_gen <- data_combined %>% filter(!is.na(commissioning_year)) %>%
            group_by(XXXX,renewable) %>%
            summarise(ann_cap = XXXX(capacity_gw))
```

```
new_gen <- data_combined %>% filter(!is.na(commissioning_year)) %>%
            group_by(commissioning_year,renewable) %>%
            summarise(ann_cap = sum(capacity_gw))
```

Let's plot the result

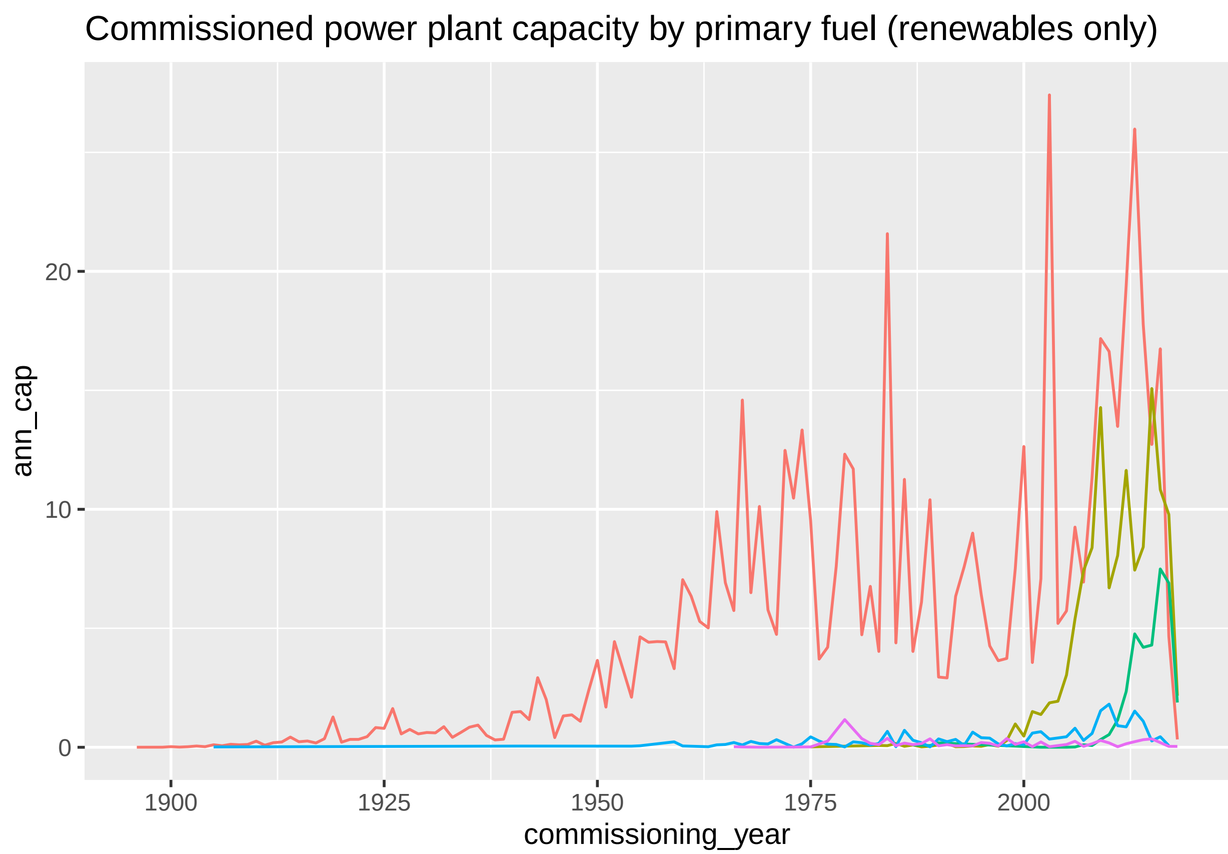
```
plot2 <- ggplot(new_gen,aes(x=commissioning_year,y=ann_cap, color=renewable)) +
          geom_line() +
          ggtitle("Commissioned power plant capacity by renewable status")
plot2
```



The last year in the dataset is 2018. But it is likely that the information for that year is incomplete. Recall also that a large proportion of power plants do not have a commissioning year information. It may well be that certain patterns (like the huge rise of commissioning capacity around the year 2000) is not really a reflection of the actual commissioning pattern but rather a result of changing reporting patterns. Also, in the document the authors argue that some capacity may not be well captured by their database. Which type of generation is likely to be less accurately recorded?

Repeat the above exercise only for renewable fuel power plants and create a plot which shows the development of the commissioned capacity by `primary_fuel` type. You should be able to replicate the plot below (but you should include a legend as in the previous plot).

```
new_gen <- data_combined %>% filter(!is.na(commissioning_year),renewable == "renew") %>%
            group_by(commissioning_year,primary_fuel) %>%
            summarise(ann_cap = sum(capacity_gw))
plot3 <- ggplot(new_gen,aes(x=commissioning_year,y=ann_cap, color=primary_fuel)) +
          geom_line() +
          ggtitle("Commissioned power plant capacity by primary fuel (renewables only)")+
          theme(legend.position = "none")
plot3
```



Preparing some data

In this section we shall run some regressions. Let us first prepare some data.

```
reg_data <- data_combined %>%  filter(commissioning_year > 2000) %>%
              group_by(country,renewable) %>%
              summarise(cap = sum(capacity_gw),
                        gdp_pc = first(GDPpc),
                        pop = first(population)) %>%
              pivot_wider(names_from = renewable,values_from = cap) %>%
              filter(!is.na(renew),!is.na(non_renew)) %>%
              mutate(gen_pc = 1000000*(renew+non_renew)/pop,
                     prop_ren = 100*renew/(renew+non_renew))
```

Let's look at the data for a few large countries:

```
reg_data %>%  filter(country %in% c("CHN","IND","PAK","RUS","USA"))
```

```
## # A tibble: 5 x 7
## # Groups:   country [5]
##   country gdp_pc        pop  renew non_renew gen_pc prop_ren
##   <chr>    <dbl>      <dbl>  <dbl>     <dbl>  <dbl>    <dbl>
## 1 CHN      9364. 1397715000 122.        828.  0.679     12.8
## 2 IND      2055. 1366417754  20.3       168.  0.138     10.8
## 3 PAK      1339.  216565318   2.62        4.94 0.0349    34.6
## 4 RUS     11456.  144373535   3.34       11.6  0.104     22.3
```

```
## 5 USA     62918.  328239523 117.      300.   1.27       28.1
```

Task 6

Figure out what the entries in the above table mean, i.e. what the newly calculated variables represent.

Estimating a regression model

We shall estimate the following regression models (`mod1`)

$$prop\_ren = \beta_0 + \beta_1\ gdp\_pc + \beta_2\ ln(pop) + u$$

and (`mod2`)

$$gdp\_pc = \alpha_0 + \alpha_1\ gen\_pc + \alpha_2\ prop\_ren + u$$

Task 7

Estimate the models above using the following skeleton code:

```
mod1 <- lm(XXXX ~ XXXX+log(pop), data = reg_data)
mod2 <- lm(XXXX)
stargazer_HC(mod1,mod2)
```

```
mod1 <- lm(prop_ren ~ gdp_pc+log(pop), data = reg_data)
mod2 <- lm(gdp_pc ~ gen_pc+prop_ren, data = reg_data)
stargazer_HC(mod1,mod2,type_out="text")
```

```
##
## ====================================================================
##                                       Dependent variable:
##                               --------------------------------------
##                                    prop_ren              gdp_pc
##                                      (1)                  (2)
## --------------------------------------------------------------------
## gdp_pc                            -0.0005**
##                                   (0.0002)
##
## log(pop)                           -2.746
##                                    (2.469)
##
## gen_pc                                              16,703.610***
##                                                      (6,241.402)
##
## prop_ren                                              -104.817**
##                                                        (53.146)
##
## Constant                           89.447**         14,646.640***
##                                   (43.509)           (3,652.218)
##
## --------------------------------------------------------------------
```

```
## Observations                          48                  48
## R2                                   0.097               0.256
## Adjusted R2                          0.057               0.223
## Residual Std. Error (df = 45)       28.495            15,642.950
## F Statistic (df = 2; 45)            2.427*              7.756***
## ===================================================================
## Note:                                  *p<0.1; **p<0.05; ***p<0.01
##                                  Robust standard errors in parenthesis
```

```
stargazer_HC(mod1,type_out="text")
```

```
##  (Intercept)       gdp_pc     log(pop)
## 4.350931e+01 1.966712e-04 2.469083e+00
##
## ==========================================================
##                               Dependent variable:
##                         --------------------------------------
##                                     prop_ren
## ----------------------------------------------------------
## gdp_pc                             -0.0005**
##                                    (0.0002)
##
## log(pop)                            -2.746
##                                     (2.469)
##
## Constant                           89.447**
##                                    (43.509)
##
## ----------------------------------------------------------
## Observations                          48
## R2                                   0.097
## Adjusted R2                          0.057
## Residual Std. Error            28.495 (df = 45)
## F Statistic                  2.427* (df = 2; 45)
## ==========================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
##                         Robust standard errors in parenthesis
```

```
stargazer_HC(mod2,type_out="text")
```

```
## (Intercept)      gen_pc    prop_ren
##  3652.21778  6241.40214    53.14571
##
## ==========================================================
##                               Dependent variable:
##                         --------------------------------------
##                                      gdp_pc
## ----------------------------------------------------------
## gen_pc                            16,703.610***
##                                    (6,241.402)
##
## prop_ren                           -104.817**
##                                     (53.146)
##
## Constant                          14,646.640***
```

```
##                                (3,652.218)
##
## -----------------------------------------------------------
## Observations                       48
## R2                                0.256
## Adjusted R2                       0.223
## Residual Std. Error        15,642.950 (df = 45)
## F Statistic                7.756*** (df = 2; 45)
## ===========================================================
## Note:                        *p<0.1; **p<0.05; ***p<0.01
##                      Robust standard errors in parenthesis
```

If you have done this correctly, you will find that that your estimated constant for `mod1` is 89.447.

Think about the interpretation of the results. In particular, does any of the above allow a causal interpretation? Also think about how you would perform inference (t-tests) on any of the estimated coefficients. For instance, how would you test $H_0 : \alpha_1 = 0$ against $H_A : \alpha_1 \neq 0$. Or how would you test $H_0 : \alpha_1 = 10,000$ against $H_A : \alpha_1 \neq 10,000$. Be prepared to be asked to do this during the test.

END OF INSTRUCTIONS

Do you want to read more? Energy economics is an important applied field of economics. Here is a link to the World Energy Outlook 2020 Report by the International Energy Agency https://www.iea.org/reports/world-energy-outlook-2020.