Module 5 Exercise - Part A

1. Preliminaries
   
a. Create a new RMarkdown PDF document in a module5 folder of your course repository.
   
   • Be sure to commit and push changes to your course repository after each question at a minimum (so after Q1, Q2, Q3).
   
   • Set the following output options:
     
     – Data frames should be printed using kable.
     – Code should be highlighted using the zenburn theme.

b. Initialize the Ecdat library
   
   • For the first part of the exercise, you will be working with the Ecdat package, which has many economics datasets. Hence you will need to install and initialize the package.

2. Educational outcomes using the Project STAR dataset
   
a. Create the dataset
   
   For the first part of the exercise, you will be working with data from Project STAR, an educational experiment conducted in Tennesse that examine the effect of student-teacher ratio on outcomes. The experiment randomized students into three class types: (1) a “regular” class size with 22-25 students, (2) a small class size with 13-17 students, and (3) a regular class size where the teacher was assisted by a teacher’s aide.
   
   a. Create an “star_data” data frame from the “Star” data in Ecdat as follows:
   
   ```r
   star_data <- Star
   ```

   b. Turn the data set into a tibble

   c. Select all variables except for “treadssk” to keep.

   d. Rename the variables as follows:
   
   • Change “tmathssk” to “math_score”
   • Change “classk” to “class_type”
   • Change “totexpk” to “teacher_exper”
   • Change “totexpk” to “teacher_exper”
   • Change “freelunk” to “free_lunch”
   • Change “schidkn” to “school”

   e. Create the following new variables:
   
   • “log_math_score”, equal to the log of “math_score”
   • “teacher_exper_sq”, equal to the square of “teacher_exper”

   f. Convert the school variable to a factor.

2. Perform regression
   
a. Create a regression object that regresses logged math scores on teacher experience, experience^2, class type, sex, race, free lunch status, and fixed effects for each school.
   
   • First, use the teacher_exper_sq and log_math_score variables in the regression
• Then rewrite the regression to appropriately express the log and squared variables using only 
  math_score and teacher_exper in the regression.

b. Display the summary output of the regression.

c. Use coeftest to produce a regression summary with HC1 robust standard errors, saving this to 
  star_reg_robust (do not display it).

d. Display the star_reg_robust regression output with stargazer, setting the following options:
  - omit="school" to omit the school fixed effects.
  - type="latex" to produce Latex-type output.
  - header=FALSE to hide the autogenerated Header description.
  - float=FALSE (a Latex option to ensure the tables stays in the position it was placed).
  - You will also probably need to add the option results="asis" to your code chunk so that the 
    Latex output from stargazer will render in the Rmarkdown document.

e. Display the regression table wit HC1 robust standard errors using the tidy function.
  - Pipe the tidy regression output to the following stringr expression in to remove the school fixed 
    effects from the output.
  ```r
  filter(!str_detect(term, "(school)"))
  ```
  - Bonus (ie not required): Replace the variable names in “term” with names of your own.

3. Perform model diagnostics

Write the statistical decision from each test in your RMarkdown report:
  a. Test for heteroskedasticity.
  b. Test for missing polynomial terms.

4. Visualize the relationship between math scores and class type with a box plot.

  a. Re-rerun the regression above, but omit class_type from the regressors.
  b. Create a tibble with class_type and the residuals from (a).
  c. Appropriately rename the columns of the tibble.
  d. Create a bar plot from the tibble, displaying the average residual math score by class type
     - Since we are not interested in a count of values for our bar plots, but instead the average values for 
       the residual math score, use the following options inside of geom_bar: stat = “summary”, fun.y = 
       “mean”.
     - Adding a graph title, axis labels, and style it like a Stata graph using theme_stata() from the 
       ggrthemes package.