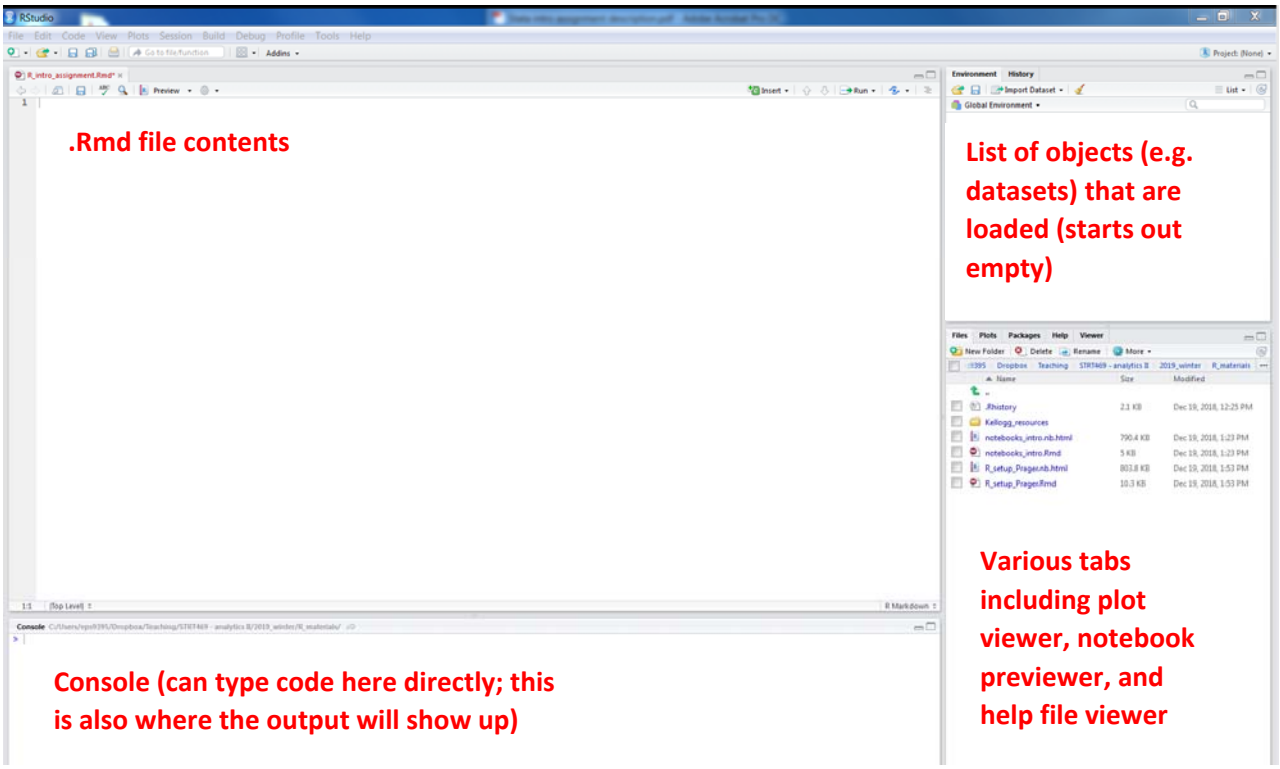
Individual Assignment #0: R Refresher

Note: This assignment is not for credit; it will be marked as complete/incomplete. The assignment can only impact your class participation score. It is designed to translate the Stata skills covered in DECS 431 or an equivalent statistics/analytics course into R. If you are brand new to R, it may require fighting through some frustration, but this will be an excellent investment in the tools you need to succeed in this course. You have the option of switching back to Stata if you prefer.

Preliminaries

1. Make sure you have working copies of **R, RStudio, and Rmarkdown installed** on your computer. There is an intro to R tutorial on Canvas that you can use to get started on this titled *R_setup_Prager.nb.html*, which you can open using a web browser. The .Rmd file with the same name generates the HTML output, as explained in the *R_setup_Prager.nb.html* file itself.
2. **Launch RStudio**. The easiest way to do this is to double-click the RStudio icon in your Start menu. It will look something like this after you open a notebook (next step).

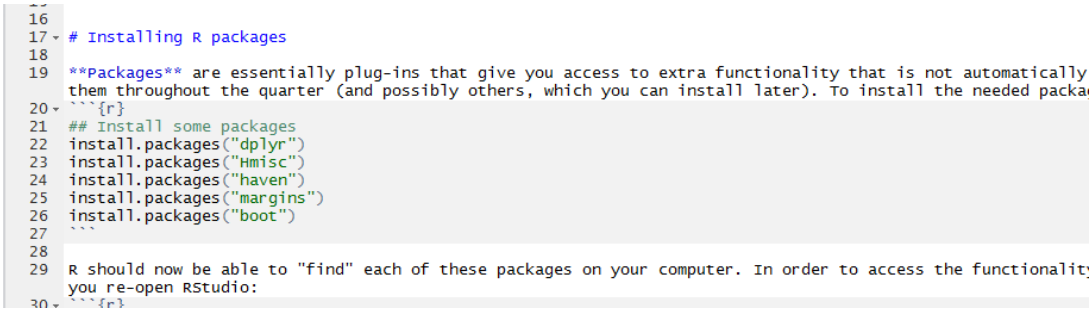


3. **Start your R notebook by going to File → New File → R Notebook.** In the first lines of your new notebook, put the following text (including the dashes):

```
---
title: "Assignment #0 - R Refresher"
author: YOUR NAME HERE
output: html_notebook
---
```

You can now save your .Rmd file (notebook) using File → Save As. This is where you will write your text, your code, etc. All code that you will submit should be written in the .Rmd notebook; code that you run just for yourself may be run from the console if you prefer. Each command goes on a new line of text.

4. **Write some code**. This require telling RStudio that what you are about to write is code, not text. You will need to **surround chunks of code with "flags" that tell R where your code begins and ends**.
   Before the first line of code in a chunk, include
   ```` ```{r} ````
   then a line break, then your code, and then a new line with
   ```` ``` ````
   RStudio will know to interpret the things in between the flags as code and grey out the whole chunk, like this:



```
16
17 # Installing R packages
18
19 **Packages** are essentially plug-ins that give you access to extra functionality that is not automatically
   them throughout the quarter (and possibly others, which you can install later). To install the needed packa
20 ```{r}
21 ## Install some packages
22 install.packages("dplyr")
23 install.packages("Hmisc")
24 install.packages("haven")
25 install.packages("margins")
26 install.packages("boot")
27 ```
28
29 R should now be able to "find" each of these packages on your computer. In order to access the functionalit
   you re-open RStudio:
30 ```{r}
```

   You can also use a shortcut instead of the somewhat clumsy ``` flags by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*. To make R actually run a chunk of code that you have written in your notebook, you will need to tell it to run (a.k.a. execute) the chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

5. **Install packages.** Packages are essentially plug-ins that give you access to extra functionality that is not automatically included in the base installation of R. You will be using them throughout the quarter (and possibly others, which you can install later). To install the needed packages, run the following code:

```
install.packages("dplyr")
install.packages("Hmisc")
install.packages("haven")
install.packages("margins")
install.packages("boot")
```

   R should now be able to "find" each of these packages on your computer. In order to access the functionality inside the packages, you need to load them into R each time you re-open RStudio:

```
library(dplyr)
library(Hmisc)
library(haven)
library(margins)
library(boot)
```

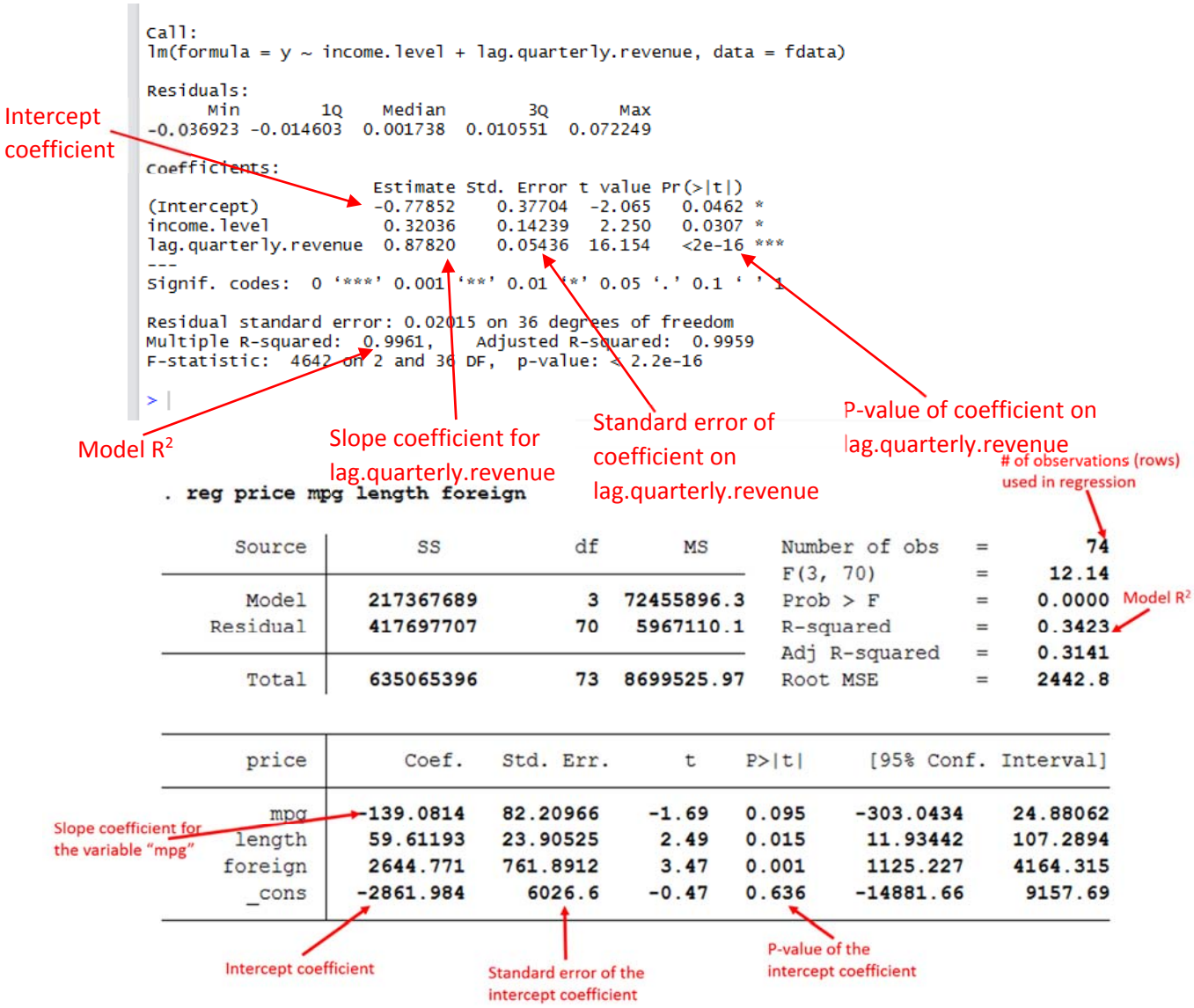
   To suppress the long output from the package installation from showing up in your HTML file and cluttering the nice output, you can specify that R should hide output from a given chunk of code at the top of the chunk by starting with ```` ```{r, results='hide'} ````, like so:

```
```{r, results='hide'}
install.packages("rmarkdown")
```
```

6. If at this point you are completely lost, open the *R_setup_Prager.Rmd* file in RStudio, and you will see what the structure of a notebook should look like.

Assignment Portion

1. Load the *freeny* dataset that automatically comes installed in R and save it to an object called *fdata* using the code:
   `fdata <- freeny`
2. To see what is in the data, use the `View(fdata)` command. This will open a new spreadsheet-style tab in which you can look at your data. When you are done, just close the new tab.
3. Check what variables are in your dataset using the `names(fdata)` command. In the notebook you submit, list the names of the last two variables in the dataset in the main text (not in a code or output chunk).
4. Find the mean of the lag quarterly revenue using the `summary(fdata)` command. Report it in the main text of your notebook.
5. Find the standard deviation of the income level using the `sd(fdata$income.level)` command, which tells R to find the standard deviation (*sd*) of the column called *income.level* of the data object *fdata*.
6. Make a new variable that is the mean of the income level and the price index using the command
   `fdata$newvarname <- (fdata$varname1 + fdata$varname2)/2`
   where you are responsible for figuring out the correct variable names to use (newvarname is the name you want to give to the new variable, whereas varname1 and varname2 are the variable you are trying to average).
7. Run a linear regression where *y* is the dependent variable (a.k.a. y-variable) and the market potential and newvarname are the independent variables (a.k.a. x-variables or explanatory variables) using the command
   ```
   model1 <- lm( y ~ newvarname + market.potential, data=fdata)
   summary(model1)
   ```
   where you are responsible for figuring out the correct variables to use. The first line estimates the linear regression model and saves the results to an object called *model1*, and the second line reports the regression output.
8. The regression should show output in your console and in your .Rmd file that will look something like this (using a different set of variables so as not to give away the answer):



9. Report in your notebook the value of the intercept coefficient and the slope coefficient on market potential. Interpret the coefficient on market potential: based on this regression, for every 1-unit increase in market potential, by how many units would we expect y to rise or fall?
10. Save your .Rmd notebook and submit it through Canvas. Also submit the HTML file generated by the notebook.
11. **If completing this assignment was difficult or took more than 15 minutes (after installing R, RStudio, and Rmarkdown), you are strongly encouraged to review R basics and regression basics <u>prior to the start of the quarter</u>. There are links to useful review resources provided on Canvas. The course will move quickly to more advanced material, and catching up now will pay big dividends later. The professor and the TA are available to answer questions and help you learn, but you must take the initiative to seek out help.**

What to Submit for Grading

Component	**Points (0 total)**	**Submission**	**Notes (additional details below)**
.Rmd file and corresponding HTML file	--	1 per person via Canvas	No points awarded. This assignment is not for credit and counts only for your class participation score.