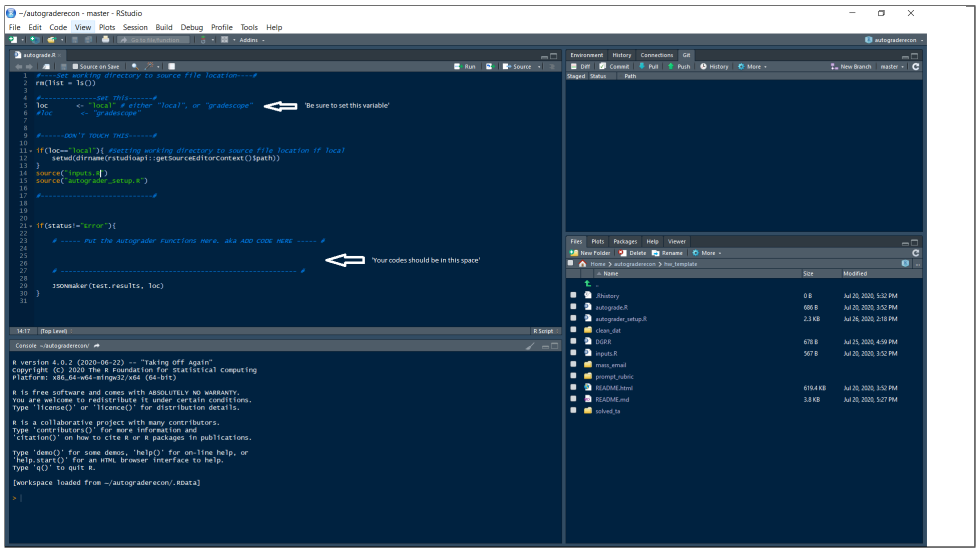
writing

Writing the Autograder

1: The bulk of your autograder codes should be in **autograde.R** in the respective homework folder. Make sure that the autograder codes are placed in the right location, and *loc* is set to correct variable for either `local` or `gradescope` grading.



2: Start your autograder by providing correct answers to each question. Make sure the answer variable's names are different from the ones specified in the prompts, and it is advised to comment your answers to later reference.

3: Compare your answers with the ones provided by students, and grade each questions accordingly. It is recommended to test if said variable exists before comparing. You can do so with **is.error()** function from **berryFunctions** library.

```
if (is.error(student.name)){"Variable name does not exist"} else{
  #Test if student answer is correct#
}
```

4: How to grade XXX?

> *1* How do you grade a tibble? Three functions can be used to compare objects like dataframe and tibble. `all.equal()` is the default function of R. It can perform crude comparison between tibble objects, but minor changes like ordering or different grouping could affects its output. `compare()` from library *compare*. It has several built-in arguments that allowed you to specify if orders, data_types, or capitalizations matter in the comparison. But this function works poorly with tibbles especially when grouping, or factor levels are different. `all_equal()` from library

tidyverse. It is similar to `compare()` in that it allowed you specify if order, etc. matter. It performs better with tibbles that have factors. But it would not recognise 1*n dataframes as valid input. Thus, it is recommended to use `ass_equal(as.tibble(df1),as.tibble(df2))`. Troubleshooting: it often happens when minor class attributes could affects the autograder, like student answer has factor as a column while your answer converted it to characters. Or differences like double versus integer. Some easy fixes for these issues are `type_convert()` fomr library *readr*.

2 How do you grade a function? Sadly no function in R provided tools that could compare if two functions are fundamentally different. A good approximation to this could be to provide a variaty of inputs to feed into both functions and compare answers. It is adviced to prepare inputs that covers multiple data types, with either valid or invalid output for more accuracy. A great example can be found on Michael Guerzhoy's Github.

3 How do you grade a numeric value with potential round-offs? Though this arrises less than often, it could be the case when a numeric answer requries multiple steps of calculation, and some round-offs are involved. Many methods exist to fix this issue. One of them is to use `almost.equal()` from library *berryFunctions*. This function allowed you to do fuzzy comparison with a set scale.

5: Save the grades in json file. This step is mostly automatic. In **autograder_setup.R**, functions should already initiate a dataframe **test.results**, with each row corresponding to one question and four columns: *description of the question*, *the score*, *max score*, and *notes*. To save the grades, you should make to fill each cell with corresponding inputs.

```
if(status!="Error"){

    # ----- Put the Autograder Functions Here. aka ADD CODE HERE ----- #


  # Example: Suppose the student earned 5 out 10 in problem 3.
 test.results[3,] <- c('Problem 3: Example(10pt)',5,10,"This should be your feedback to the student")
```

6: Pushing to Github by Git push.

7: My autograder broke, why? If this happens to you, don't fret. This happens **A LOT**. Try to see if you have done the following: *1* Pushed **autograde.R** before saving *loc* to *gradescope 2* Created the autograder in a branch other than **master** and did not merge the branches after pushing *3* Used functions from packages that are not installed but not declared, this can fixed by 'library(NameOfthePackage)' *4* Used functions from packages that are not installed in the Virtual Machine, to fix this you should edit **setup.sh** in your **gradescope.zip**. For more debugging tips, please refer to VM manual. *5* some parts of the autograder returned NA instead of correct grading, this is usually hinted by:

```
Error in JSONmaker(test.results, loc) :
  Error: There is a missing value somewhere (NA). Please make sure all
             values are filled in. If a student recieved 0 points please make sure
             there is a 0 in the matrix
```