Lab 1: Introduction to R and RStudio

Hello! Welcome to the sample assignment. This will be a brief Rmd just to get you comfortable with how PH 142 students complete their assignments and get used to the workflow.

As mentioned in the document, we use RMarkdown because of its ability to combine Markdown text with R code chunks. This makes it easier for us to explain the concept we are learning at the time in conjunction with the code associated with that chunk. For example:

A `print` function outputs whatever you provide as the argument. Try replacing the contents inside of the `print` function with your favorite word.

```
print("supercalifragilisticexpialidocious")
```

```
## [1] "supercalifragilisticexpialidocious"
```

Try replacing the `print` function in the next code chunk with your second most favorite word.

```
print("pneumonoultramicroscopicsilicovolcanoconiosis")
```

```
## [1] "pneumonoultramicroscopicsilicovolcanoconiosis"
```

In addition, we are able to use the Ottr Grader features to provide (not really) instant feedback to the students on their code after it's written. They can do it multiple times until they get the correct answer.

Before the examples, please run this line of code to load the `testthat` library, a dependency of Ottr Grader.

Assign the best number in the world to the variable `best_number` in the code chunk below. (Hint: PH 142)

```
best_number <- NULL # YOUR CODE HERE
```

```
. = ottr::check("tests/best_number.R")
```

```
## [1] "Checking that ‘best_number‘ is a numeric"
## -- Failure (???): p1a ----------------------------------------------------------
## is.numeric(best_number) is not TRUE
##
## ‘actual‘:   FALSE
## ‘expected‘: TRUE
##
## [1] "Checking that ‘best_number‘ equals 142"
## -- Failure (???): p1b ----------------------------------------------------------
## best_number == 142 is not TRUE
##
## ‘actual‘:
```

```
## ‘expected‘: TRUE
##
## Test p1a failed:
##     Error: is.numeric(best_number) is not TRUE
##
##     ‘actual‘:   FALSE
##     ‘expected‘: TRUE
##
## Test p1b failed:
##     Error: best_number == 142 is not TRUE
##
##     ‘actual‘:
##     ‘expected‘: TRUE
```

Fill in the following blank by assigning the word (as a string) to the variable `blank`:

```
import numpy as _____
```

```
blank <- NULL # YOUR CODE HERE
```

```
. = ottr::check("tests/blank.R")
```

```
## [1] "Checking that ‘blank‘ is a character."
## -- Failure (???): p2a ----------------------------------------------------------
## is.character(blank) is not TRUE
##
## ‘actual‘:   FALSE
## ‘expected‘: TRUE
##
## [1] "Checking that ‘blank‘ is ’np’"
## -- Failure (???): p2b ----------------------------------------------------------
## blank == "np" is not TRUE
##
## ‘actual‘:
## ‘expected‘: TRUE
##
## Test p2a failed:
##     Error: is.character(blank) is not TRUE
##
##     ‘actual‘:   FALSE
##     ‘expected‘: TRUE
##
## Test p2b failed:
##     Error: blank == "np" is not TRUE
##
##     ‘actual‘:
##     ‘expected‘: TRUE
```

Once a student finishes their assignment, they must use the `turn_in.py` script in order to submit to Gradescope. - First, make sure that you have knitted your file by clicking on the “Knit” button on top of the Source. - Next, type in the following command into the terminal (not the console!)

```
cd; cd dlab_sample_repo/Sample_Assignment/turn_in.py
```

You should be prompted if you have knit your file, then for your Gradescope credentials. For the purposes of this example, we have hardcoded the script to only take in the following credentials (anything else will result in an error): - User: dlab@berkeley.edu - Pass: password123

Note: When typing in the password, the characters you type will not show up in the terminal. This is a security feature for students (and staff) to prevent revealing their passwords when typing it in.

And that's it! This is pretty much the full abridged version of a student's typical workflow. `lab01` has also been provided to show a real example of what students might have to go through.