# *In-Class Computing Day 2*

## *Math 253: Statistical Computing & Machine Learning*

This is the second in-class computing task for the course. You've already seen the following instructions from the first day's in-class task. I repeat them here just to remind you.[1]

### *Today's tasks*

These are meant to help you learn some basics in R. Don't expect the answers to be immediately evident to you.

#### *Task 1*

In the Chapter 2 lab from ISL, reference is made to a data set named `Auto`. The book says, "The textfile can be obtained from this book's web site." Here's the address: `http://www-bcf.usc.edu/~gareth/ISL/Auto.csv`

In your `.Rmd` document, write this line and execute it **once**. It will download the file `Auto.data` to your R system.

```
download.file("http://www-bcf.usc.edu/~gareth/ISL/Auto.csv",
              destfile="Auto.csv")
```

With the file on your system, you don't need to download it again, so **comment out** the statement to avoid re-running every time you "compile" document. (The `usc.edu` web site is slow; it's particularly painful to download the file every time you modify your script.)

Create an object called `auto_file_name` that contains the full path name to the `Auto.data` file on your system. You can find the full path name by giving the command `file.choose()` in the console[2] and navigating to the file. Cut and paste the output of `file.choose()` into your solution file.

#### *Task 2*

Create an object called `Auto` using the `read.csv()` method at the middle of p. 49

You will need to replace the short string `"Auto.data"` in the book's commands with the full path name you've just stored in `auto_file_name` in Task 1.

#### *Task 3*

Create an object `task3` that holds a summary of the `horsepower` variable, as with the summary near the bottom of p. 51. **Do not**

[1] **Instructions**

1. Make sure you are working within your personal "project" in RStudio that's linked to your GitHub repository.
2. You'll construct your answers in an `.Rmd` file. A stub of that file *already exists* in your project/repository. This being Day 2, that file is called `02-Programming.Rmd`.
3. The script file **must** compile to HTML when you "knit" it. *If there are parts of your script that are incomplete, comment them out to produce a file that does run to completion while preserving your incomplete work as comments.*
4. Many of the tasks will involve assigning a value to a given name, such as `task3`. Make sure to use this name exactly, including capitalization. If you don't, the scoring system will miss your answer.
5. To help you keep track of your answers, you may want to use comments to divide your `.Rmd` file into sections, like `## Task 3`
6. At the end of the class, go through the stage/commit/pull/push process to synchronize your project with the GitHub server and *vice versa*.

[2] Commands that interact with the user, such as `file.choose()` or `View()` cannot be run properly when compiling your `.Rmd` document. Such commands are to be run only in the R console.

**use** `attach()`.^[Using `attach()` is bad style. It's best to write statements that refer to the data table(s) explicitly. A save alternative to `attach()` is `with()`, as in

```
task3 <- with(Auto, summary(horsepower))
```

] You may, however, use either `$` or `with()` as you prefer.

```
task3 <- summary(Auto$horsepower)
```

### *Task 4*

All the data from the book is in the `ISLR` package. Using the "Packages" tab in RStudio, install this package. In your script file, give the command to load the package.

### *Task 5*

Use the indexing operations described in ISL §2.3.3 to create the following objects:

- `task5top` the subset of `Auto` consisting of the first 5 rows in the first 3 columns.
- `task5bottom` the subset of `Auto` consisting of the last 5 rows in the last 3 columns

### *Task 6*

These programming tasks come with a set of test statements that check which elements of the activity have been done correctly. The `02-Programming.Rmd` file (and all the other daily files) already contain a section at the end of the document that runs the checks and displays the results.

The tests rely on a package called `scoreActivity`. You may need to install this package. This is a one-time operation. Since it was written just for this class, the package is not distributed via the official CRAN system. Run this statement from the console:

```
devtools::install_github("dtkaplan/Math-253/DailyProgrammingTasks/scoreActivity")
```

If it turns out that you don't have the `devtools` package, install it in the usual way using the "Packages" tab in RStudio.