

In-Class Computing Project 1

Math 253: Statistical Computing & Machine Learning

Starting out in R

This is the first in-class computing task for the course. These instructions apply to this and future computing tasks.

1. You'll construct your answers in an `.Rmd` document that lives in your individual `math253-<your name>` repository. The document for today is called `01-Programming.Rmd`. Upcoming days' work will involve files named in the same fashion.
2. When you are finished with your work, you will compile the `.Rmd` file to HTML. The compilation **must** run to completion when you "source" it. That is, you should be able to see the resulting HTML file.

If there are chunks in your document that are incomplete, comment them out to produce a file that does compile while preserving your incomplete work.

3. 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.
4. To help you keep track of your answers, use subsection headings to divide your `.Rmd` file, like this:

```
## Task 3
```

5. At the end of the class, stage the documents you've changed (including the HTML document), commit, and push to GitHub

Before we start, you will need to install on your RStudio system a package that provides tests for your document, so that you can see what commands are not yet right. Here's the command to install the package, which you need do only once.

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

There is a package called `mosaicData`. Write the command that will load this package into R. Note that `load()` is (surprisingly) not the correct function to do this.

You may find that `mosaicData` is not yet installed on your R system. If not, use the *Packages* tab in RStudio to install it. This is done outside of your script. Once installed, a package can be used in any script you like without re-installation.

Task 2

Create an object named `task2` that contains a character string like “Today is Tue Sep 20 07:16:11 2016”. Your command should be such as to create a string with the correct date on any day in the future that the document is compiled.

You will find two functions to be useful:

- `paste()`, which concatenates strings.
- A function that looks at the system clock to figure out what day it is. I’m not going to tell you which function it is. You figure it out yourself. Here are some possibilities: `now()`, `today()`, `day_time()`, `day()`, `time()`, `date_time()`, `date()`, `at_present()`, `clock()`, `calendar()`.

Task 3

In the `mosaicData` package, there is a data frame named `Galton`. Create these objects:

- `task3a` containing the names of the variables in `Galton`
- `task3b` containing the number of cases in `Galton`
- `task3c` containing the mean of `height` in `Galton`

Task 4

Create an object `task4` containing the following object:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Hint: the object is a matrix.

Task 5

Create these objects:

- `task5x` a vector with 1000 random values between 0 and 1. (Hint: `runif()`)
- `task5y` another such vector.
- `task5pi` the value of π estimated from a “Monte Carlo” simulation. That is, for each of `task5x` and `task5y`, square the vector. Then add them together and take the square root. This will give the length from the origin to each of 1000 random points in the unit square in the first quadrant. Finally, compute what fraction of these lengths are ≤ 1 . That fraction is an estimate of $\pi/4$.

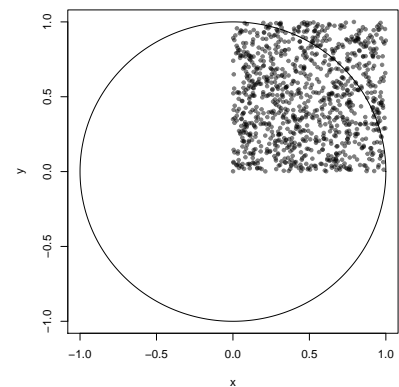


Figure 1: 1000 random points uniformly distributed in the unit square in the first quadrant. The unit circle is shown for comparison.