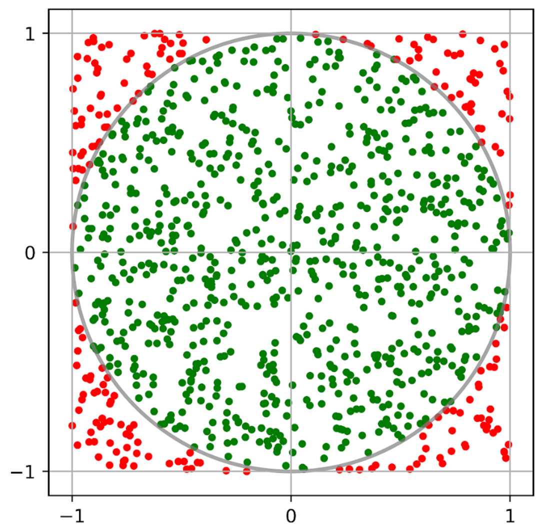
Week 4 Lab problems

EEB 429
Bhaskar Kumawat

Modular workflows using functions

Calculating π

A *Monte Carlo experiment* is a way of calculating certain numerical quantities of interest using random sampling. For example, consider the following diagram:



The radius of the circle in the above diagram is 1.0 (r). The length of the side of the square in the figure is 2.0 (l). Thus, the ratio of area of the circle to the area of the square is,

$$Ratio\ of\ areas = \frac{Area\ of\ the\ Circle}{Area\ of\ the\ Square} = \frac{\pi r^2}{l^2} = \frac{\pi(1)^2}{2^2} = \frac{\pi}{4}$$

Assuming we don't know the value of π beforehand, if we can somehow calculate the areas, we can back-calculate the value of π using the above formula. Here, we will use a simulation of a Monte Carlo experiment to calculate the value of π. The algorithm for doing so is as follows,

1. Randomly pick n points within a square region ranging from -1 to 1 along both x and y-axes.
2. Calculate the number of these points that lie within the circle of radius 1.
3. Then, the approximate ratio of areas (R) is just the number of points that lie within the circle divided by n. Subsequently, $\pi = 4R$.

In this lab, we will work together to create a program to calculate the value of π using this method. Divide into teams of 2-3 people. Each team will create a function as described in the **team memos** below.

Team Memo A: Generating random points

In a new R-script file titled `get_random_points.R`, create a function called `GetRandomPoints` with the following blueprint.

Arguments

1. `n`: Number of random points to generate.

Body

1. Use the `runif()` function to generate `n` number of x-coordinate values picked randomly between -1 and 1. (Check out the help page to learn more about this function)
2. Use the `runif()` function to generate `n` number of y-coordinate values picked randomly between -1 and 1.
3. Bind the two sets of values to create a matrix and return it.

Outputs

1. A matrix with *two columns* and *n* *rows*. Each cell in the matrix contains a random number between -1 and 1.

Note: You can test this function in a separate R-console/script if you want, but the file you submit should only have the function and no other variables outside the function scope. Submit these files on Canvas at the Lab 4 submission assignment. Only one person from a team needs to submit it, but make sure you include full names of everyone in your team as a comment in the code.

Team Memo B: Finding if a point is within a circle

In a new R-script file titled `point_checker.R`, create a function called `IsPointInUnitCircle` with the following blueprint.

Arguments

1. `x`: x-coordinate of the point to check
2. `y`: y-coordinate of the point to check

Body

1. Calculate the distance of the point from the origin. The formula for the distance of a point (x, y) from the origin is given by $\sqrt{x^2 + y^2}$.
2. Compare this distance to the radius of a unit circle (i.e., 1), and return TRUE if the distance is strictly less than this radius. (i.e., if the point is within the circle)

Outputs

1. A single boolean type TRUE or FALSE depending on if the point is inside or outside the unit circle.

Note: You can test this function in a separate R-console/script if you want, but the file you submit should only have the function and no other variables outside the function scope. Submit these files on Canvas at the Lab 4 submission assignment. Only one person from a team needs to submit it, but make sure you include full names of everyone in your team as a comment in the code.

Team Memo C: Calculating π

In a new R-script file titled `pi_from_points.R`, create a function called `CalculatePi` with the following blueprint.

Arguments

1. `points`: A two-column matrix. Each row contains the x and y coordinate of a point.
2. `n`: Number of rows in the above matrix (i.e., the total number of points).

Body

1. Create a variable called `counter` and set it to zero.
2. ***For*** each point in the matrix,
    a. Use a function called IsPointInUnitCircle(x,y) to figure out if the point is within the unit circle. This function returns a TRUE or a FALSE. (To know more about this function, talk to the team working on Memo B)
    b. ***If*** the point is within the unit circle, increment the variable `counter` by 1.
3. Now, `counter`'s value should be the number of points within the circle.
4. Divide `counter` by `n` to get a fraction called `ratio_of_areas`.
5. Calculate the value of `pi` from `ratio_of_areas` (See first page) and return it!

Outputs

1. A single number that contains an estimate of the value of π calculated using the matrix of randomly generated points.

Note: You can test this function in a separate R-console/script if you want, but the file you submit should only have the function and no other variables outside the function scope. Submit these files on Canvas at the Lab 4 submission assignment. Only one person from a team needs to submit it, but make sure you include full names of everyone in your team as a comment in the code.

Team Memo D: Plotting the graph

In a new R-script file titled `plot_pis.R`, create a function called `PlotValues` with the following blueprint.

Arguments

1. `num_points`: A vector that contains different values of the number of points to be used for calculating an estimate of π.
2. `pi_estimates`: A vector that contains different values of π estimated using a Monte Carlo experiment.

Body

1. Create a scatter plot of `num_point` (x-axis) against `pi_estimates` (y-axis) and store it in a variable called `final_plot`.
2. Add a gray, dotted horizontal line to the plot at y=`3.14159265358979323846`.
3. Return the plot.

Outputs

1. A plot of the input vectors with a horizontal line at the correct value of π.

Note: You can test this function in a separate R-console/script if you want, but the file you submit should only have the function and no other variables outside the function scope. Submit these files on Canvas at the Lab 4 submission assignment. Only one person from a team needs to submit it, but make sure you include full names of everyone in your team as a comment in the code.