**Estimating Pi using the Monte Carlo Simulation Method**

**Some outcomes of this assignment:**

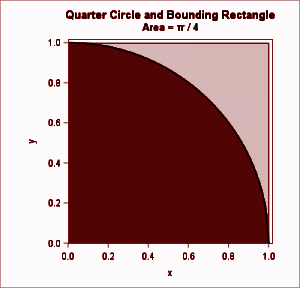
* Monte Carlo Simulation method for calculating Pi
* Different random number generator distributions in NumPy
* Basics of plotting
* Basics of Latex

**Problem Definition:**

Using the well-known Monte Carlo Simulation Method, you will implement a Python function that estimates the Pi value. Furthermore, with different random number generator distributions, you will analyze your Pi estimates, plot the histogram of your generated numbers and you will measure the performance gain by performing operations on NumPy arrays.

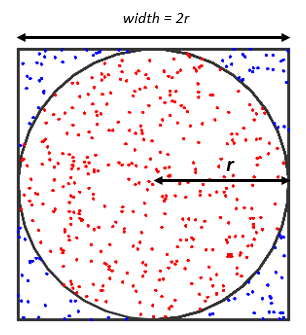
You can use the following function to compute the Monte Carlo estimates of the Pi value:

The curve of the function is shown in the plot below for the interval [0,1]. The bold area under the curve is .



The idea here is that you can estimate areas by generating a uniform sample of random points and counting how many fall into the region. In the Monte Carlo simulation method, you can estimate the Pi value by generating N random points in the unit square ([0,1]) and counting the proportion that falls in the quarter circle. To compute the correct Pi value you must multiply the estimate by 4.

The other approach considers the problem in a 2-D domain. You need to generate random (x, y) pairs in the interval of [-1,1] and then check if . If yes, you need to increase the number of points that are inside the circle. Then, dividing number of points generated inside the circle by the number of points generated inside the square gives an estimation of Pi.



**Assignment:**

1. Writing code: Generate 1,000,000 random numbers and apply the Monte Carlo method for calculating Pi by using the following distributions
   1. Uniform
   2. Normal (Gaussian)
   3. Chisquare
   4. Poisson
   5. Power
   6. Rayleigh
2. Discussion: Compare and discuss the Pi estimates by using those 6 distributions. Which one gives the best estimate of Pi? Explain why.
3. Plotting: Draw the histogram (# bins=1000) plots of the values generated by using 6 distributions.

**Report:**

Your answers and explanations to all the above questions should be included in a PDF file (hw2\_ID\_nameSurname.pdf) that must be formed by using Latex. You can access the template from <https://www.overleaf.com/read/rdxhvszksjxm#4c5362>

You can clone an editable copy from your Overleaf.com dashboard. You can see the following page for more details:

<https://www.overleaf.com/learn/how-to/Copying_a_project#Making_a_copy_of_a_project>

If you are not familiar with latex, you can see this page:

<https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes>

**Submission:**

You should submit a zip file named **ID\_nameSurname.zip** (do not use Turkish characters and empty space in the name of any files) that includes your report in PDF and codes for each distribution of the random number generators.

**Some Notes:**

* Submit only .py source files. Jupyter notebook .ipynb files are not allowed!
* Partial codes will get 0. You must submit the work in full.
* -20 pts for a report in a different Latex format or in MS Word format docx.
* Cheating from any source (including internet) is strictly forbidden.