MM 225: AI and Data Science

Tutorial 2: Computations and simulations

Week 2: Tutorial Group 1

1. Consider a circle of diameter 4 units inscribed inside a square of side length 4 units. Write a python script to do the following:

   - Generate a random points x and y inside the square: that is, generate two random numbers between 0 and 4 and this gives a point (x,y);
   - If the point lies inside the circle, increase the count by one; that is, if $(x-2)^2 + (y-2)^2 \leq 4$, $count \leftarrow count + 1$;
   - Repeat the above two steps a large number of times, say, N times.
   - After N steps, the ratio $\frac{count}{N}$ is the same as the ratio of the area of the circle to that of the square.
   - Since the area of the square is 16, evaluate the value of π. **Hint:** area of the circle is $\pi r^2 = 4\pi$.

   Evaluate the value of π for various N values. Plot the value of π as a function of N. For what value of N, does the value of π converge? How much is the error as compared to the π value given by the `math.pi`? Plot the error as a function of N. Upload your script and the figures.

2. Consider the cycloid curve which is described by the parametric equations $x = r(\theta - \sin(\theta))$ and $y = r(1 - \cos(\theta))$. Assume $r = 2$ and consider θ in the range $(-2\pi, 2\pi)$. Plot the cycloid.

   Hint: You may use the following script as the model for your solution.

```
import matplotlib.pyplot as plt
import numpy as np

ax = plt.figure().add_subplot(projection='3d')

# Prepare arrays x, y, z
theta = np.linspace(-4 * np.pi, 4 * np.pi, 100)
z = np.linspace(-2, 2, 100)
```

```
r = z**2 + 1
x = r * np.sin(theta)
y = r * np.cos(theta)

ax.plot(x, y, z, label='parametric curve')
ax.legend()

plt.show()
```

MM 225: AI and Data Science

Tutorial 2: Introduction to python

Week 2: Tutorial Group 2

You may use the slides of the lecture for reference.

1. Write a python script to simulate and hence calculate the probability of success when you sample 10 fruits from a basket containing 100 fruits out of which 15% of the fruits are bad.

   Hint: The `numpy.random` module contains the `hypergeometric` function call to draw samples from the hypergeometric distribution. Once you have the data, you can use the same to calculate the probability.

2. Write a python script to simulate the 1-D random walk process, which can be described as follows. Flip a fair coin, and take left (right) if the coin turns up a head (tail).

   Write another script to generate data for a random walk of 5, 10, 100, 500, 1000, 5000, 10000 and 20000 steps using your simulation script and to plot the distance from the origin as a function of steps.

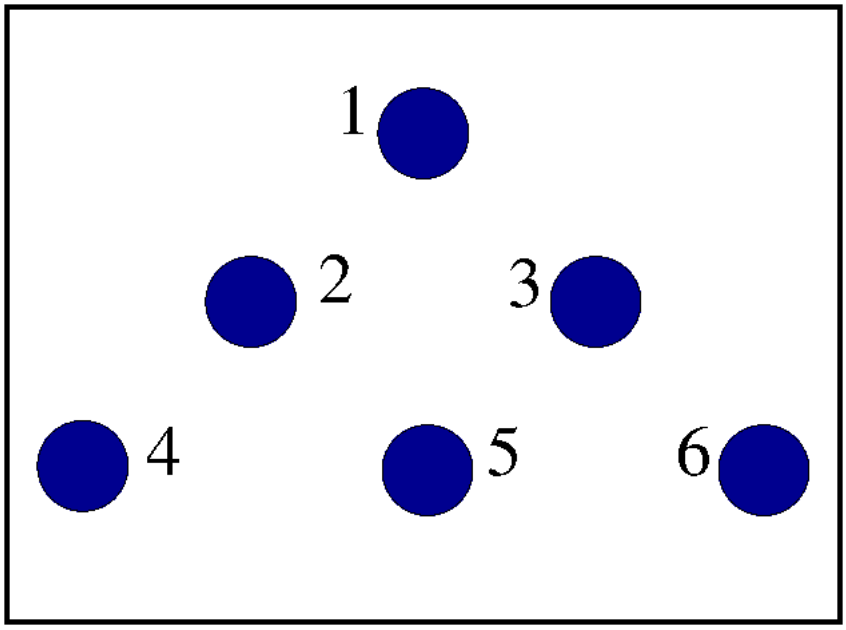
   Upload your scripts as well as the plot in moodle.

MM 225: AI and Data Science

Tutorial 2: Computations and simulations

Week 2: Tutorial Group 3

1. Consider a game played using a board in which six obstacles are placed as shown in the figure.



A ball which is thrown on 1 can either get deflected to the left and reach 2 or get deflected to the right and reach 3; again, from 2, the ball can either get deflected to 4 or 5; from 3, the ball can be deflected to 5 or 6. Assume all these deflections have equal probability.

Write a python script to simulate the process of a large number of balls (say N) which are made to fall on 1.

After getting deflected at these three stages, let us assume that the balls reach the bottom to any one of the four positions, namely, A: left of 4, B: right of 4 (left of 5), C: right of 5 (left of 6) and D: right of 6. Using your simulations, calculate the probabilities for a ball to reach A, B, C and D. Run the simulations using various number of balls (N = 50, 100, 200, 500, 1000 and 5000) and plot how these probabilities change with N.

Upload your scripts and the plots.

2. Using `matplotlib` module, it is possible to make surface plots in 3-D. Using the following example script as a model, write a script which plots the following function:

$$z = \exp\left[-(x^2 + y^2)\right] \tag{1}$$

for x and y in the range (-1,1).

```
import matplotlib.pyplot as plt
import numpy as np

from matplotlib import cm

fig, ax = plt.subplots(subplot_kw={"projection": "3d"})

# Make data.
X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

# Plot the surface.
surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm,
                       linewidth=0, antialiased=False)


# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()
```

MM 225: AI and Data Science

Tutorial 2: Computations and simulations

Week 2: Tutorial Group 4

You may use the slides of the lecture for reference.

1. Write a script to plot $\sin x$ and $\cos x$ for $x = \{-4\pi, 4\pi\}$ in the same figure.

   Hint: Using `matplotlib.pyplot` module and invoking the `plot` command more than once, we can plot more than one curve in the same figure.

2. A student has measured the temperature of a sample ten times. The readings (in °C) are as follows:

   `1100, 1120, 1080, 1090, 1100, 1120, 1080, 1080, 1110, 1110`

   Use the module `statistics` to calculate the statistics of the measurements. (a) How will you report the results of this measurement? (b) How will you report the confidence level of your result? (c) What assumptions are you making in answering the above questions? Upload your script along with your answers.

3. Guru (G) and Hina (H) are playing a penny matching game. The game (based on the description of the wikipedia page) is as follows:

   > It is played between two players, G and H. Each player has a penny and must secretly turn the penny to heads or tails. The players then reveal their choices simultaneously. If the pennies match (both heads or both tails), then G wins and keeps both pennies. If the pennies do not match (one heads and one tails), then H wins and keeps both pennies.

   Write a python script to simulate this game. Make the computer play the game 100 times. Plot the profit or loss made by G in these 100 games.

   Hint: You may assume that both G and H play the game fair; that is, they choose head and tail for their pennies with equal probability.