|  |
| --- |
| Name: D G L P Geethanjana |
| Student Reference Number: 10820935 |

Text

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Module Code:  PUSL2018 | Module Name: Computational Theory and Statistics for Computing | |
| Coursework Title: PUSL2018 C1 | | |
| Deadline Date: 04/01/2023 | | Member of staff responsible for coursework:  Dr. Rasika Ranaweera |
| Programme: BSc (Hons) Computer Science | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  10820935 - D G L P Geethanjana  10820936 - R T Athukorala  10820945 - RMKM Rathnayake  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  Signed : | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |

\*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

**Task 1 - Game Theory**

Three tasks—A, B, and C—must be accomplished according to the project's work plan, albeit there may be some ambiguity regarding how long they will take to complete. However, it is determined to simulate the completion times in days using a Monte Carlo simulation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Activity** | **Optimistic** | **Most Likely** | **Pessimistic** | **PERT Estimate** |
| A | 8 | 10 | 12 | 60 |
| B | 10 | 12 | 14 | 72 |
| C | 12 | 14 | 16 | 84 |
| Total | 30 | 36 | 42 | 216 |

Calculation of the duration of each activity by using PERT Formula;

PERT Estimate = ( Optimistic Estimate+ 4 x Most likely Estimate + Pessimistic Estimate) / 6

Total Completion Time of the project is = 60 + 72 + 84 = 216 days.

For the best case, completion time of the project is ;

Total Completion Time = 8 + 10 + 12 = 30 days.

For the worst case, completion time of the project is ;

Total Completion Time = 12 + 14 + 16 = 42 days.

|  |  |
| --- | --- |
| **Duration (in days)** | **Likelihood of Completion** |
| 30 - 35 | 43% |
| 36 – 38 | 42% |
| 39 - 42 | 15% |

Code

import cv2

import numpy as np

import math

import random

import matplotlib.pyplot as plt

np.random.seed(500)

x = np.random.randint(low = 30, high = 42, size = 500)

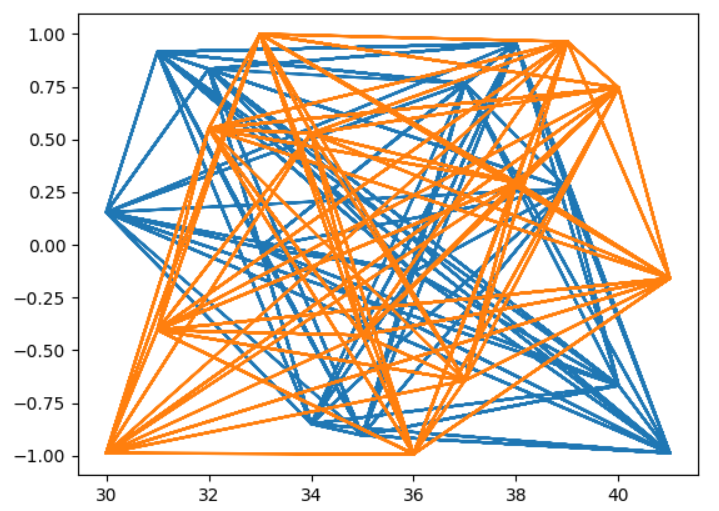
plt.plot(x, np.cos(x))

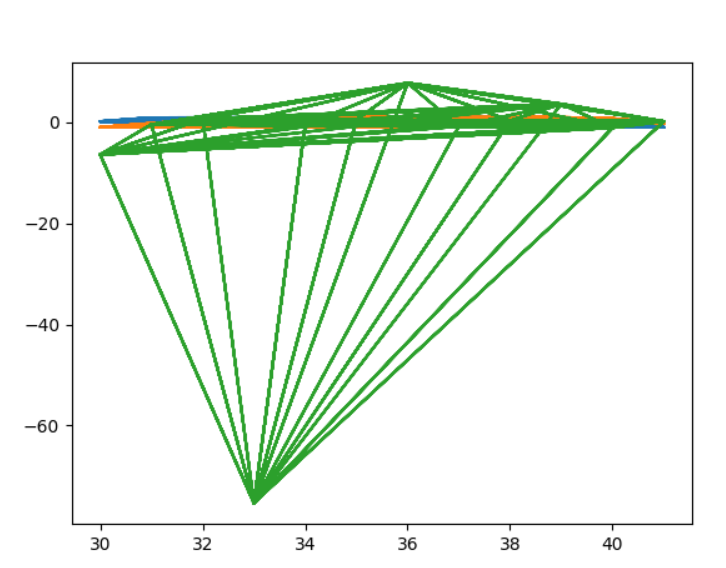
plt.plot(x, np.sin(x))

#plt.plot(x, np.tan(x))

plt.show()

Output





**Task 2 - Probability**

Here, it’s given that the family has 3 children which can either be a boy or a girl. Since being a boy or a girl are independent and equally likely events. Hence, we have

P(Boy) = P(Girl) = 1/2

We want to find,

P(atleast a girl) = P(1 girl + 2 girls + 3 girls) = 1 − P(no girls)

So,

P(atleast a girl) = 1 − P(1/2 × 1/2 × 1/2) = 1 − (1/2)3 = 1 − (1/8) = 7/8 = 0.875

P(all are girls) = 1 − P(atleast a girl) = 1 − (7/8) = 1/8 = 0.125

Code

# Importing Packages

import matplotlib.pyplot as plt

import random

# Creating Probability Function, Girl = 1, Boy = 0

def count\_girl():

child\_1 = random.randint(0, 1)

child\_2 = random.randint(0, 1)

child\_3 = random.randint(0, 1)

# Determining if the child are the same gender

if child\_1 == child\_2 == child\_3:

same\_num = True

else:

same\_num = False

return same\_num

# Inputs

num\_simulations = 10000

max\_num\_count = 1000

count = 1

# Tracking

girl\_probability = []

count\_balance = []

# Creating Figure for Simulation Balances

fig = plt.figure()

plt.title("Monte Carlo [" + str(num\_simulations) + "simulations]")

plt.xlabel("Simulations")

plt.ylabel("Count")

plt.xlim([0, max\_num\_count])

# For loop to run for the number of simulations desired

for i in range(num\_simulations):

balance = [1000]

num\_count = [0]

num\_girl = 0

# Run until the player has rolled 1,000 times

while num\_count[-1] < max\_num\_count:

same = count\_girl()

# Result if the dice are the same number

if same:

balance.append(balance[-1] + count)

num\_girl += 1

# Result if the dice are different numbers

else:

balance.append(balance[-1] - count)

num\_count.append(num\_count[-1] + 1)

# Store tracking variables and add line to figure

girl\_probability.append(num\_girl/num\_count[-1])

count\_balance.append(balance[-1])

plt.plot(num\_count, balance)

# Showing the plot after the simulations are finished

plt.show()

# Averaging win probability and end balance

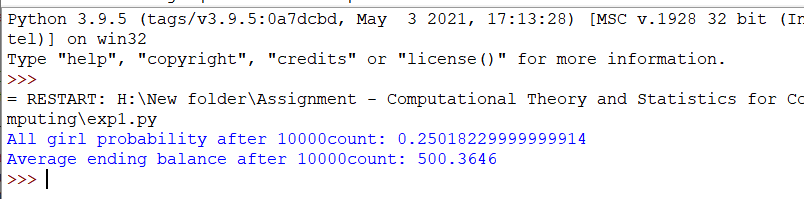
overall\_girl\_probability = sum(girl\_probability)/len(girl\_probability)

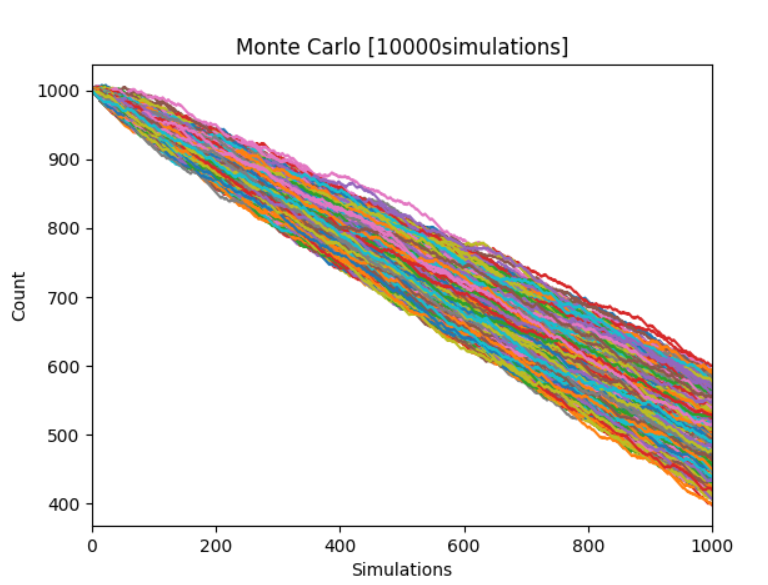
overall\_count\_balance = sum(count\_balance)/len(count\_balance)

# Displaying the averages

print("All girl probability after " + str(num\_simulations) + "count: " + str(overall\_girl\_probability))

print("Average ending balance after " + str(num\_simulations) + "count: " + str(overall\_count\_balance))

Output



**REFERENCES**

Sokolowski, J.A. (2010) “Monte Carlo Simulation,” *Modeling and Simulation Fundamentals*, pp. 131-145. Available at: https://doi.org/10.1002/9780470590621.ch5.

U, O. (no date) *A mathematical formulation of the Monte Carlo method - Osaka U*, *Towards Data Science*. Available at: http://www4.math.sci.osaka-u.ac.jp/~sugita/Public/imath/ipaper/MCM\_SS\_E\_digest.pdf (Accessed: January 4, 2023).

Chen, J.D. (2018) *Monte-Carlo Simulation-Based Statistical Modeling*, *Springer Link*. SPRINGER Verlag, SINGAPOR. Available at: https://link.springer.com/book/10.1007/978-981-10-3307-0.

*Probability: The basics (article)* (no date) *Khan Academy*. Khan Academy. Available at: https://www.khanacademy.org/math/statistics-probability/probability-library/basic-theoretical-probability/a/probability-the-basics (Accessed: January 4, 2023).

(no date) *Stat 391 home*. Available at: https://sites.stat.washington.edu/mmp/courses/391/spring22/ (Accessed: January 4, 2023).

(no date) *Python tutorial*. Available at: https://www.w3schools.com/python/ (Accessed: January 4, 2023).

*Learn Python programming* (no date) *Programiz*. Available at: https://www.programiz.com/python-programming (Accessed: January 4, 2023).

**GROUP DETAILS AND CONTRIBUTION**

|  |  |  |
| --- | --- | --- |
| Student Name | Student ID | Contribution |
| D G L P Geethanjana | 10820935 | * Task 2 Mathematical Implementation * Task 2 Python Code and Pseudo Code |
| R T Athukorala | 10820936 | * Task 1 Monte Carlo Python Code * Gathering Ideas about Tree Diagram |
| RMKM Rathnayake | 10820945 | * Task 1 Monte Carlo Pseudo Code * Research about Mathematical Implementation |