Eli Sylvia-Lourde

Written HW Chapter 5: 5.1 #4, 5.3 #'s 1,3

Math 114 Mathematical Modeling

St. Mary's College

March 13th, 2019

5.1 # 4

Use Monte Carlo simulation to approximate the area under the curve $f(x) = \sqrt{x}$ over the interval $\frac{1}{2} \leq x \leq \frac{3}{2}$. In order to solve this, we use the following equation:

$$(b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i) \approx \int_a^b f(x)dx$$

```
from random import uniform
from scipy import random
import math
import numpy as np

def func(x):
    return math.sqrt(x)

def main():
    N = 10000
    a = 0.5
    b = 1.5
    xrand = random.uniform(a,b,N)
    good = 0
    total = len(xrand)
    integral = 0.0

    for i in range(N):
        integral += (func(xrand[i]))
    answer = (b-a)/float(N)*integral

    print(answer)

main()
```

5.3 #’s 1,3

1) You arrive at the beach for a vacation and are dismayed to learn that the local weather station is predicting a 50% chance of rain every day. Using Monte Carlo simulation, predict the chance that it rains three consecutive days during your vacation. In order to solve this, we create random numbers, use a baseline to determine if they should be counted as ’rain’ or ’no rain’, and then counted the number of times it consecutively rained for three days. Finally, we divided this by the number unique 3-day spans for the given N. The equation to find this is: $N - 2$.

```
def beach_dayz():
    N = 10002
    # xrand = []
    # for i in range(N):
    #     xrand.append(rand_int())
    # 1 means no rain
    xrand = random.uniform(0,1,N)
    print(xrand)
    good = 0
    streak = 0
    for i in range(N):
        if (xrand[i] >= 0.5):
            xrand[i] = 1
        else:
            xrand[i] = 0


    print(xrand)

    # Now we need to find all ’runs’ of rain

    runs = []

    for i in range(N):
        if (xrand[i] == 1):
            runs.append(1)
        if (xrand[i] == 0 and len(runs) > 0):
            if (runs[-1] == 1):
                runs.append(0)


    num_of_threes = 0

    for i in range(len(runs)-3):
        if (runs[i] == 1 and runs[i+1] == 1 and runs[i+2] == 1):
            num_of_threes += 1

    print(num_of_threes)

    print("Number of times it rains for three consecutive days: {}".format(num_of_threes

    # We care about the number of three day 'sets' that it could have rained for

    print("Number of total 3 Cycles: {}".format(N-2))

    print("{}/{} = {}".format(num_of_threes,(N-2),(num_of_threes)/(N-2)))

beach_dayz()
```

3) Use Monte Carlo simulation to simulate the sum of 100 rolls of a pair of fair dice.

Where c1, c2, ..., c6 represent how often the dice lands on a given side. C1 is side 1, c2 is side 2, etc...

```
def fair_coin():

    a = 0

    b = 1

    N = 100

    c_1 = 0

    c_2 = 0

    c_3 = 0

    c_4 = 0

    c_5 = 0

    c_6 = 0

    xrand = random.uniform(a,b,N)
    for i in range(len(xrand)):
        if (xrand[i] <= 1/6 ):
            c_1 += 1
        elif (xrand[i] <= 2/6 ):
            c_2 += 1
        elif (xrand[i] <= 3/6 ):
            c_3 += 1
        elif (xrand[i] <= 4/6 ):
            c_4 += 1
        elif (xrand[i] <= 5/6 ):
            c_5 += 1
        elif (xrand[i] <= 6/6 ):
            c_6 += 1
    print(c_1)
    print(c_2)
    print(c_3)
    print(c_4)
    print(c_5)
    print(c_6)
fair_coin()
```