Grade: 30/30

HW 1 - Monte Carlo
University of Chicago - Linear Algebra and Python
Professor Chao-Jen Chen

Jennifer Gunawan
Duke University
September 2023

Problem 1

In the first problem, I conducted a Monte Carlo simulation to predict the number of times we need to roll a fair 6-sided die to see every number. We can analytically solve this problem in two ways: (1) we can calculate the expected number of trials as $E[X] = n(1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}) \Rightarrow 6(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6})$. (2) we can use a recursive formula, given: $E[X] = 1 + \frac{n-1}{n}E[x_1]$, where $X_1 = \frac{6}{6} + \frac{6}{5} + \frac{6}{4} + \frac{6}{3} + \frac{6}{2} + \frac{6}{1}$. In both ways, E[X] = 14.7.

We could implement the Monte Carlo simulation in Python to computationally solve the problem (see Python code below). I used for and while loops with the random library to find the expected number of rolls. I report the expected number of rolls for three cases:

1. M = 1,000: 14.621
2. M = 10,000: 14.688
3. M = 100,000: 14.70127

I notice that the expected number of rolls approaches the analytical solution (14.7) as the number of trials (M) increases. This is a good sign because the Monte Carlo simulation is converging towards the theoretical expectation.

Problem 2

In the second problem, I attempt to find the number of n-digit phone numbers that doesn't start with 0 or 1 and don't have "911" in them, where $3 \leq n \leq 8$ and n is an integer. Sample test cases include $n = 3$ and $n = 7$; both answers print *True* (see below for Python code).

Homework M2HW01

In [1]:

```
import sys
import platform
import notebook
import numpy as np
print(f'Operating System={platform.system()} {platform.release()}')
print(f'Python={sys.version}')
print(f'notebook={notebook.__version__}')
print(f'numpy={np.version.version}')
```

```
Operating System=Windows 10
Python=3.11.3 | packaged by Anaconda, Inc. | (main, Apr 19 2023, 23:46:3
4) [MSC v.1916 64 bit (AMD64)]
notebook=6.5.4
numpy=1.24.3
```

Problem 1

In [2]:

```
import random
```

In [3]:

```
def M2HW01P1(M = 100, N=6):
    """
    M: number of trials
    N: number of faces in a die, which should be set to 6 for this problem
    """

    expectation_estimate = 0

    ######################################################################

    ### Fill in here with your Monte Carlo simulation to calculate expecta

    #random.seed(32) #to check work; not part of the actual assignment.
    for _ in range(M):
        numbers_seen = set()
        rolls = 0

        while len(numbers_seen) < N:
            roll = random.randint(1,N)
            numbers_seen.add(roll)
            rolls += 1

        expectation_estimate += rolls

    expectation_estimate /= M

    ######################################################################

    return expectation_estimate
```

In [4]:

```
%time e = M2HW01P1(100000)
print(e) # e should be close to 14.7.

# Call M2HW01P1 for the three cases (case #3 shown above but the number mo
num_trials = [1000, 10000, 100000]

for M in num_trials:
    expected_rolls = M2HW01P1(M)
    print(f"M = {M}; Expected number of rolls = {expected_rolls}")
```

```
CPU times: total: 391 ms
Wall time: 572 ms
14.70674
M = 1000; Expected number of rolls = 14.621
M = 10000; Expected number of rolls = 14.688
M = 100000; Expected number of rolls = 14.70127
```

Problem 2

In [5]:

```
def M2HW01P2(n):
    if n < 3 or n > 8:
        return "n not within range (3 <= n <= 8)"

    count = 0

    for num in range(10**(n-1), 10**n):
        if str(num)[0] not in ["0", "1"]: #phone numbers that don't start
            if "911" not in str(num): #phone numbers that don't contain "
                count += 1
    return count
```

In [6]:

```
# sample test cases. All of them should print True.
print(M2HW01P2(n=3) == 799)
print(M2HW01P2(n=7) == 7958028)
```

```
True
True
```

In [ ]: