

Writing Assignment 3 - Computer Program

Max Bolger

October 2021

1 Introduction

Problem - Two integers, p, q , are selected *uniformly* at random from the interval $[1, 10000]$. What is the probability that $\frac{p}{q}$ is in *lowest terms*? I have solved the problem two different ways that yield the same answer.

Solution Technique 1 - Solution technique 1 was more for fun than it was aiming for efficiency (solution 2 is better in that regard) for solution technique 1, I generate random m sized samples of two numbers, p, q drawing from a uniform distribution from the interval $[1, 10000]$ and check whether or not $\frac{p}{q}$ is in lowest terms for each pair of p and q in the sample. The average amount of times $\frac{p}{q}$ is in lowest terms is calculated and this process is repeated n times. Ultimately, the average of these averages is the final calculation (similar to a re-sampling process). More specifically, an m sized sample (default is 100) is created for both p and q and the observations are *zipped* together into m *tuples* of p and q . A loop then iterates through each tuple and detects whether or not $\frac{p}{q}$ is in lowest terms. If it is, a 1 is appended to a list. If not, a 0 is appended to the same list (a list of whether or not $\frac{p}{q}$ is in lowest terms). The mean value of this list is calculated and that mean value is passed into a different list (a list of the means). This entire process is repeated n times. Thus, the mean value of the list of means is a simulated re-sampling probability that $\frac{p}{q}$ is in lowest terms.

Solution Technique 2 - Solution technique 2 is more concise in that it doesn't repeat the whole sampling process n times. Rather, it just generates one m sized random sample for p, q , zips each observation together, and checks if they are in lowest terms. If they aren't, a zero is appended to a list. If it is, a 1 is appended to the same list. The mean value of this list is our answer.

Definition 1 (Uniform distribution) *Uniform distribution is a type of probability distribution in which all outcomes are equally likely.*

Definition 2 (Lowest terms) *The form of a fraction in which the numerator and denominator have no factor in common except 1.*

Definition 3 (Zipped) The `zip()` function in python returns a zip object, which is an iterator of tuples where the first item in each passed iterator is paired together, and then the second item in each passed iterator are paired together etc.

Definition 4 (Tuple) A tuple is one of four built-in python data types and are used to store multiple items in a single variable.

2 Answer

As n increases, the probability of $\frac{p}{q}$ being in lowest terms converges at $\approx .608$ (With technique 1, the mean of 10000 sample means was .608. With technique 2, a sample of 100,000,000 resulted in a mean of .60798).

3 Pseudocode (Solution Technique 2)

Algorithm 1: lowestTermsCheck (p,q)

Input : Two integers, p, q

Output: A list of binaries where 1 = *lowest terms* and 0 = *not lowest terms*

```

1 newList = [ ]
2 if gcd(p,q) is 1 then
3   | append 1 to newList
4 else
5   | append 0 to newList
6 end if
7 return newList
```

Algorithm 2: lowestTermsSim (m)

Input : An integer, m , that will be used as a sample size

Output: A probability that $\frac{p}{q}$ will be in lowest terms

```

1 x = random sample of size m from a uniform distribution [1,10000]
2 y = random sample of size m from a uniform distribution [1,10000]
3 for i,j in zip(p,q) do
4   | lowestTermsCheck(i,j)
5 end for
6 s = mean value of newList
7 return s
```

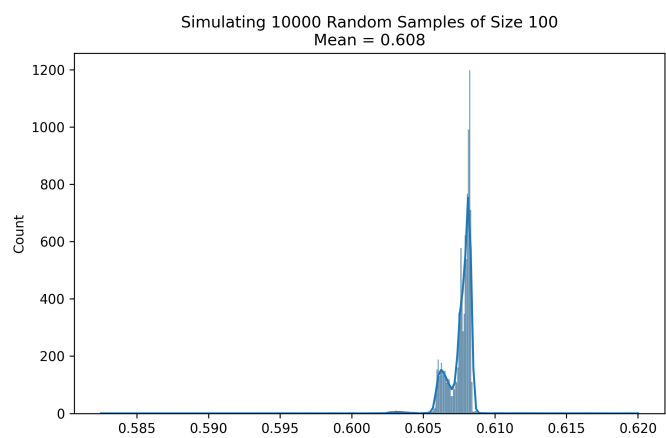
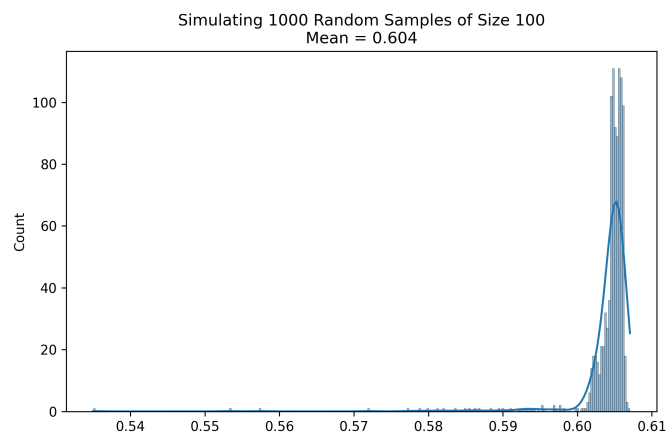
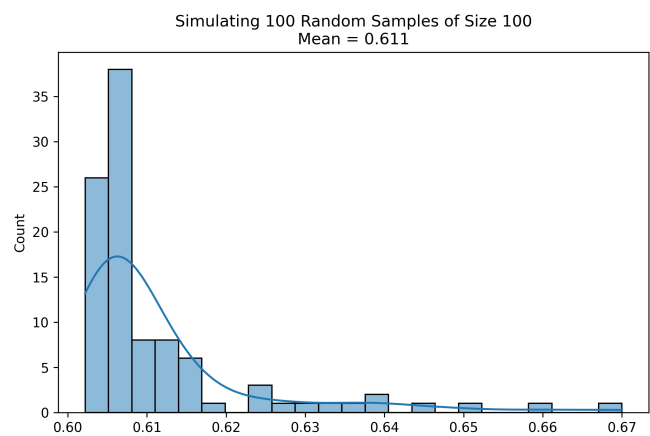
4 Steps (Solution Technique 2)

1. Import the necessary packages.
2. Create an empty list to store binaries of whether or not each observation of p, q is in lowest terms.
3. Define a function that calculates whether or not $\frac{p}{q}$ is in lowest terms. If it is, append a 1 to the outcomes list. If it isn't, pass a 0.
4. Define a function that generates an m sized sample of p, q , zip each observation together and iterate through them, passing them through the previously defined lowest terms check function.
5. Calculate the mean of the outcomes list. This is our probability that $\frac{p}{q}$ is in lowest terms.

5 Code (Solution Technique 1)

```
1 import numpy as np
2 from math import gcd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5
6 means = []
7 lt = []
8
9 def lowestTermsCheck(p,q):
10
11     if gcd(p,q) == 1:
12         lt.append(1)
13
14     else:
15         lt.append(0)
16
17 def lowestTermsSim(n,m=100):
18
19     for _ in range(n):
20         p = np.random.randint(low=1, high=10001, size=m)
21         q = np.random.randint(low=1, high=10001, size=m)
22
23         for i,j in zip(p,q):
24             lowestTermsCheck(i,j)
25
26         means.append(np.mean(lt))
27
28 fig, ax = plt.subplots(figsize=(8,5))
29 s = sns.histplot(means,kde=True)
30 plt.title(
31     f'Simulating {n} Random Samples of Size {m}\n' \
32     f'Mean = {round(np.mean(means),3)}'
33 )
34
35 return s
```

Listing 1: Solution technique 1 incorporates a simulation process of many samples for a visual representation of convergence



6 Code (Solution Technique 2)

```
1 import numpy as np
2 from math import gcd
3
4 lt = []
5
6 def lowestTermsCheck(p,q):
7
8     if gcd(p,q) == 1:
9         lt.append(1)
10
11     else:
12         lt.append(0)
13
14 def lowestTermsSim(m):
15
16     p = np.random.randint(low=1, high=10001, size=m)
17     q = np.random.randint(low=1, high=10001, size=m)
18
19     for i,j in zip(p,q):
20         lowestTerms(i,j)
21
22     s = (np.mean(lt))
23
24     return s
25
26 lowestTermsSim(100000000)
27
28 0.60798292
```

Listing 2: Solution technique 2 is more efficient and generates only one sample