Assignment Probability

Date: 04/05/2020 Name: D.Saravanan

1. Three coins are tossed. Find the probability of getting

   a) At least one head
   The outcomes of this experiment are ordered pairs of H and T.
   The sample space is S = {HHH, TTT, HTT, THT, TTH, THH, HTH, HHT}
   The sample space has eight total number of outcomes.
   Let event E = at least one head.
   There are seven outcomes that meet this condition, {HHH, HTT, THT, TTH, THH, HTH, HHT}.

$$P(E) = \frac{7}{8} = 0.875$$

   b) Exactly 2 heads
   Let event E = exactly 2 heads.
   There are three outcomes that meet this condition, {THH, HTH, HHT}.

$$P(E) = \frac{3}{8} = 0.375$$

Program:

```
#!/usr/bin/env python

import numpy as np
from random import randint
from iteration_utilities import duplicates
from iteration_utilities import unique_everseen
from num2words import num2words

print("\n____Probability of tossing coins____")
no_coin = int(input("\nEnter the number of coins: "))
option1 = input("Type L for at least, M for at most, E for exact: ")
option2 = input("Type H for Heads, T for Tails: ")

if option2 == 'H':
    side = 'Heads'
    print("[Note: Assigned value 1 for Heads and 0 for Tails]")
else:
    side = 'Tails'
    print("[Note: Assigned value 0 for Heads and 1 for Tails]")

data = []
numTrails = 10_00_000

for m in range(numTrails):
    flips = [randint(0,1) for n in range(no_coin)]

    if option2 == 'H':
        for n in flips:
            if n == 1: data.append(1)
            else: data.append(0)

    if option2 == 'T':
        for n in flips:
            if n == 0: data.append(1)
            else: data.append(0)
```

```
data = np.array([data[i:i + no_coin] for i in range(0, len(data), no_coin)]).tolist()
sample_space = list(unique_everseen(duplicates(data)))
length = len(sample_space)
print("\nThe sample space has {} total number of outcomes: \n{}".format(length, sample_space))

if option1 == 'L':
    lvalue = int(input("\nAt least number of {}: ".format(side)))

    count = 0
    for _list_ in sample_space:
        sum = 0
        for n in _list_:
            sum += n

        if sum >= lvalue: count += 1

    lvalue = num2words(lvalue)

    print("The Probability of getting at least {} {} = {}\n".format(lvalue, side, count/length))

elif option1 == 'M':
    mvalue = int(input("\nAt most number of {}: ".format(side)))

    count = 0
    for _list_ in sample_space:
        sum = 0
        for n in _list_:
            sum += n

        if sum <= mvalue: count += 1

    mvalue = num2words(mvalue)

    print("The Probability of getting at most {} {} = {}\n".format(mvalue, side, count/length))

else:
    evalue = int(input("\nExact number of {}: ".format(side)))

    count = 0
    for _list_ in sample_space:
        sum = 0
        for n in _list_:
            sum += n

        if sum == evalue: count += 1

    evalue = num2words(evalue)

    print("The Probability of getting exactly {} {} = {}\n".format(evalue, side, count/length))
```

Output:

```
____Probability of tossing coins____

Enter the number of coins: 3
Type L for at least, M for at most, E for exact: L
Type H for Heads, T for Tails: H
[Note: Assigned value 1 for Heads and 0 for Tails]

The sample space has 8 total number of outcomes:
[[0, 1, 0], [1, 0, 1], [0, 1, 1], [0, 0, 1], [0, 0, 0], [1, 1, 1], [1, 1, 0], [1, 0, 0]]

At least number of Heads: 1
The Probability of getting at least one Heads = 0.875


____Probability of tossing coins____

Enter the number of coins: 3
Type L for at least, M for at most, E for exact: E
Type H for Heads, T for Tails: H
[Note: Assigned value 1 for Heads and 0 for Tails]

The sample space has 8 total number of outcomes:
[[1, 1, 0], [1, 1, 1], [0, 1, 1], [0, 1, 0], [0, 0, 1], [1, 0, 1], [0, 0, 0], [1, 0, 0]]

Exact Number of Heads: 2
The Probability of getting exactly two Heads = 0.375
```

2. An integer is choosen at random out of the integers from 1 to 100. What is the probability that it is

   a) Multiple of 5
   The sample space is S = {1, 2, 3, . . ., 100}
   From integers 1 to 100, there are 20 integers that are multiple by 5, (since, $100//5 = 20$).
   Let event E = multiple of 5.
   $$P(E) = \frac{20}{100} = 0.2$$

   b) Divisible by 7
   From integers 1 to 100, there are 14 integers that are divisible by 7, (since, $100//7 = 14$).
   Let event E = divisible by 7.
   $$P(E) = \frac{14}{100} = 0.14$$

   c) Greater then 70
   From integers 1 to 100, there are 30 integers that are greater than 70.
   Let event E = greater then 70.
   $$P(E) = \frac{30}{100} = 0.3$$

Program:

```
#!/usr/bin/env python3
import numpy as np

print("\n_An integer is choosen at random out of the integers from 1 to 100_\n")

numTrails = 10_00_000

data = []
for n in range(numTrails):
    data.append(np.random.randint(1, 101, size=1))

sample_space = np.unique(data)
total_outcome = len(sample_space)
minValue = min(sample_space)
maxValue = max(sample_space)

print("The Sample space has {} total number of outcomes: \n{}".format(total_outcome,
    sample_space))

option1 = input("\nType M for multiples, L for lesser than, G for greater than: ")
if option1 == 'M': choosen = 'multiples of'
elif option1 == 'L': choosen = 'lesser than'
else: choosen = 'greater than'
option2 = int(input("Enter an integer for which to find values that are {}: ".format(choosen)))

values = []
for num in sample_space:
    if option1 == 'M':
        if num%option2 == 0: values.append(num)
    elif option1 == 'L':
        if num < option2: values.append(num)
    else:
        if num > option2: values.append(num)

total = len(values)

print("\nFrom {} to {}, there are {} integers that are {} {}: \n{}".format(minValue, maxValue,
    total, choosen, option2, values))
print("\nThe Probability that an integer is {} {}: {}\n".format(choosen, option2,  total/
    total_outcome))
```

Output:

```
_An integer is choosen at random out of the integers from 1 to 100_

The Sample space has 100 total number of outcomes:
[  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100]

Type M for multiples, L for lesser than, G for greater than: M
Enter an integer for which to find values that are multiples of: 5

From 1 to 100, there are 20 integers that are multiples of 5:
[5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100]

The Probability that an integer is multiples of 5: 0.2


_An integer is choosen at random out of the integers from 1 to 100_

The Sample space has 100 total number of outcomes:
[  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100]

Type M for multiples, L for lesser than, G for greater than: M
Enter an integer for which to find values that are multiples of: 7

From 1 to 100, there are 14 integers that are multiples of 7:
[7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98]

The Probability that an integer is multiples of 7: 0.14


_An integer is choosen at random out of the integers from 1 to 100_

The Sample space has 100 total number of outcomes:
[  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100]

Type M for multiples, L for lesser than, G for greater than: G
Enter an integer for which to find values that are multiples of: 70

From 1 to 100, there are 30 integers that are greater than 70:
[71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
    95, 96, 97, 98, 99, 100]

The Probability that an integer is greater than 70: 0.3
```

3. Generate Prime Numbers till 1000 and find the following among this population.

   a) Mean
   b) Median
   c) Range
   d) Quartile
   e) Inter-Quartile Range
   f) Variance
   g) Standard Deviation

Program:

```
#!/usr/bin/env python3

import numpy as np
import sympy as sp

data = list(sp.primerange(1, 1000))
print("Generate prime number till 1000: \n{}\n".format(data))

print("Mean: {:.1f}".format(np.mean(data)))
print("Median: {:.1f}".format(np.median(data)))
print("Range: {:.0f}".format(np.max(data) - np.min(data)))
print("Quartile 1: {:.1f}".format(np.percentile(data, 25)))
print("Quartile 2: {:.1f}".format(np.percentile(data, 50)))
print("Quartile 3: {:.1f}".format(np.percentile(data, 75)))
print("Inter-Quartile Range: {:.1f}".format(np.percentile(data, 75) - np.percentile(data, 25)))
print("Variance: {:.1f}".format(np.var(data)))
print("Standard Deviation: {:.1f}".format(np.std(data)))
```

Output:

```
Generate prime number till 1000:
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97,
    101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191,
    193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283,
    293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401,
    409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509,
    521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631,
    641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751,
    757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877,
    881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]

Mean: 453.1
Median: 436.0
Range: 995
Quartile 1: 188.5
Quartile 2: 436.0
Quartile 3: 703.0
Inter-Quartile Range: 514.5
Variance: 88389.4
Standard Deviation: 297.3
```