## Simulation

**Scenario:**

We want to give the vaccination of corona virus to the sample of 10 people in which 3 are Politicians, 3 are medical practitioners and 4 are ordinary citizen.

Two people are selected randomly for vaccine, if X be the number of politician getting vaccine , Y be the number of ordinary citizen to be chosen for the vaccine and Z be the number of medical practitioners to be chosen for vaccine. Find the joint probability distribution for X and Y.

In the above Scenario X, Y and Z are the random variables.

Possible values of X = 0, 1, 2 Possible values of Y = 0, 1, 2

Possible values in combination of X, Y and Z for selecting two individuals are:

(0, 0, 2) : Neither politician nor ordinary citizen, 2 medical practitioners

(0, 1, 1) : No politician, 1 medical practitioner and 1 ordinary citizen

(0, 2, 0) : No politician, 2 ordinary citizen, no medical practitioners

(1, 0, 1) : 1 politician, No ordinary citizen, 1 medical practitioner

(1, 1, 0) : 1 politician, 1 ordinary citizen and no medical practitioners

(2, 0, 0) : 2 politician, neither ordinary citizen nor medical practitioners

Now, we have to calculate the probabilities of each combination to make the joint probability distribution.

$$\text{Total number of possible outcomes} = C_2^{10} = \binom{10}{2} = 45$$

$$P(x=0, y=0) = \frac{\binom{3}{0}\binom{4}{0}\binom{3}{2}}{\binom{10}{2}} = \frac{(1)(1)(3)}{45} = 0.0666$$

$$P(x=0, y=1) = \frac{\binom{3}{0}\binom{4}{1}\binom{3}{1}}{\binom{10}{2}} = \frac{(1)(4)(3)}{45} = 0.2666$$

$$P(x=0, y=2) = \frac{\binom{3}{0}\binom{4}{2}\binom{3}{0}}{\binom{10}{2}} = \frac{(1)(6)(1)}{45} = 0.1333$$

$$P(x=1, y=0) = \frac{\binom{3}{1}\binom{4}{0}\binom{3}{1}}{\binom{10}{2}} = \frac{(3)(1)(3)}{45} = 0.2$$

$$P(x=1, y=1) = \frac{\binom{3}{1}\binom{4}{1}\binom{3}{0}}{\binom{10}{2}} = \frac{(3)(4)(1)}{45} = 0.2666$$

$$P(x=2, y=0) = \frac{\binom{3}{2}\binom{4}{0}\binom{3}{0}}{\binom{10}{2}} = \frac{(3)(1)(1)}{45} = 0.0666$$

The Joint Probability distribution for X,Y will be:

| Y / X | 0 | 1 | 2 | h(Y) |
|---|---|---|---|---|
| **0** | 0.0666 | 0.2666 | 0.1333 | ***0.4666*** |
| **1** | 0.2 | 0.2666 | 0 | ***0.4666*** |
| **2** | 0.0666 | 0 | 0 | ***0.0666*** |
| **g(X)** | ***0.3333*** | ***0.5333*** | ***0.1333*** | ***1.00*** |

Here g(X) shows the marginal probabilities of X and h(Y) shows the marginal probabilities of Y

Now, Let's try to simulate this scenario in python for large number of sample.

**Simulation:**

Simulation involves artificially recreating the random phenomenon. Below simulation is done in jupyter notebook using 'symbulate'. The code and the results are in the screen shots below:

In [1]:

```
# Librarirs
from symbulate import *
from tabulate import tabulate
import matplotlib.pyplot as plt
import seaborn as sn
```

In [2]:
```
P = BoxModel(['Politician', 'Ordinary_citizen', 'Medical_practitioner'],
             size=2, replace = True)
P.sim(10000).tabulate()
```

Out[2]:

| Outcome | Frequency |
|---|---|
| (Medical_practitioner, Medical_practitioner) | 1111 |
| (Medical_practitioner, Ordinary_citizen) | 1115 |
| (Medical_practitioner, Politician) | 1119 |
| (Ordinary_citizen, Medical_practitioner) | 1170 |
| (Ordinary_citizen, Ordinary_citizen) | 1068 |
| (Ordinary_citizen, Politician) | 1078 |
| (Politician, Medical_practitioner) | 1094 |
| (Politician, Ordinary_citizen) | 1109 |
| (Politician, Politician) | 1136 |
| **Total** | **10000** |

In the above code BoxModel is representing a container that includes three types of units "Politician, Ordinary_Citizen and Medical_Practitioners".

Size = 2 represent the size of randomly choosing the two units with replacement from the container.

We simulate the 10,000 outcomes of size 2 and tabulate them. Outcomes and the frequency are given in the table above. We can see that the frequency of the random selection of both Medical_practitioners is 1111, frequency of the random selection of both politician is 1136 and so on.

In [3]:
```
P.sim(10000).tabulate(normalize=True)
```

Out[3]:

| Outcome | Relative Frequency |
|---|---|
| (Medical_practitioner, Medical_practitioner) | 0.1191 |
| (Medical_practitioner, Ordinary_citizen) | 0.1152 |
| (Medical_practitioner, Politician) | 0.1102 |
| (Ordinary_citizen, Medical_practitioner) | 0.1071 |
| (Ordinary_citizen, Ordinary_citizen) | 0.1087 |
| (Ordinary_citizen, Politician) | 0.1068 |
| (Politician, Medical_practitioner) | 0.1089 |
| (Politician, Ordinary_citizen) | 0.1118 |
| (Politician, Politician) | 0.1122 |
| **Total** | **1.0** |

Above table represent the probability of each randomly selected pair. Now, we generate the random variables.

```
In [4]:
1 P = BoxModel([0, 1, 2], size=2, replace = True)
2 # 0 = Politician, 1 = Medical_practitioners, 2 = Ordinary_citizen
3 P.sim(10000).tabulate()
```

Out[4]:

| Outcome | Frequency |
|---|---|
| (0, 0) | 1121 |
| (0, 1) | 1070 |
| (0, 2) | 1124 |
| (1, 0) | 1139 |
| (1, 1) | 1074 |
| (1, 2) | 1201 |
| (2, 0) | 1134 |
| (2, 1) | 1079 |
| (2, 2) | 1058 |
| **Total** | **10000** |

This table is the same but with numerical entries, we need this to generate the random variables. As we mentioned above 0 = Politician, 1 = Medical_practitioners, 2 = Ordinary_citizen.

For random variables we choose the maximum value for ordinary citizen and minimum value for politician.

```
In [5]:
1 X = RV(P, min)  # Politician
2 Y = RV(P, max)  # Ordinary citizen
```

```
In [6]:
1 outcome = (0, 1)
2 X(outcome), Y(outcome)
```

```
Warning: Calling an RV as a function simply applies the function that defines the RV to the input, regardless of whether tha
t input is a possible outcome in the underlying probability space.
Warning: Calling an RV as a function simply applies the function that defines the RV to the input, regardless of whether tha
t input is a possible outcome in the underlying probability space.
```

```
Out[6]: (0, 1)
```

We define the outcome as first value is for random variable X and second for Y.

```
In [7]:
1 x = X.sim(10000)
2 x.mean(), x.var()  # mean and variance of X random variable
```

```
Out[7]: (0.5571, 0.46073959000000003)
```

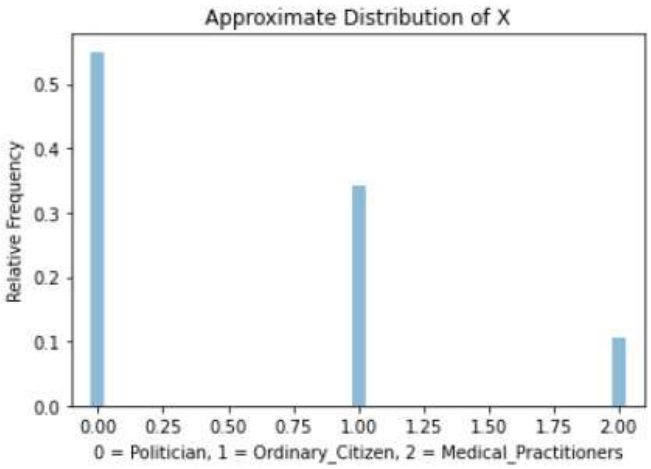
```
In [8]:
1 y = Y.sim(10000)
2 y.mean(), y.var()  # mean and variance of Y random variable
```

```
Out[8]: (1.4518, 0.4618767600000001)
```

Above results shows the mean and variance of the X and Y random variables.

In [9]:

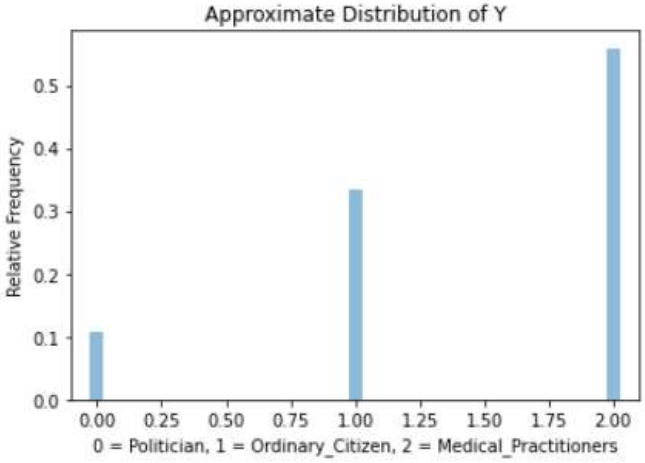
```
x.tabulate(normalize=True)
plt.title("Approximate Distribution of X")
plt.xlabel("0 = Politician, 1 = Ordinary_Citizen, 2 = Medical_Practitioners")
x.plot(linewidth = 8)
```




Above is the approximate distribution of X.

In [10]:

```
y.tabulate(normalize=True)
plt.title("Approximate Distribution of Y")
plt.xlabel("0 = Politician, 1 = Ordinary_Citizen, 2 = Medical_Practitioners")
y.plot(linewidth = 8)
```




Above is the approximate distribution of Y.

Below is the joint probability distribution of X and Y.

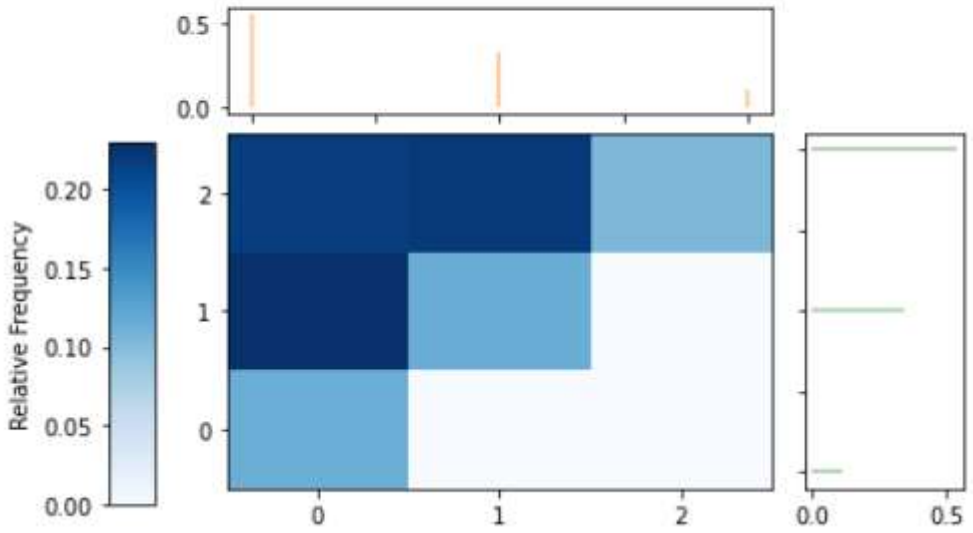
```
In [11]:
1 joint = (X & Y).sim(10000).tabulate(normalize=True)
2 joint
```

Out[11]:

| Value | Relative Frequency |
|---|---|
| (0, 0) | 0.1109 |
| (0, 1) | 0.2275 |
| (0, 2) | 0.2255 |
| (1, 1) | 0.1026 |
| (1, 2) | 0.2243 |
| (2, 2) | 0.1092 |
| **Total** | **1.0** |

Here is the marginal and joint probability distribution's plot.

```
In [12]:
1 (X & Y).sim(10000).plot(
2 type=["tile", "marginal"])
```




Now, we convert above graph of joint and marginal probabilities into table.

```
In [13]:
1 ## Tabulate the joint and marginal probabilities
2 data = [('0', '0.1101', '0.223', '0.2253', '0.5584'),
3         ('1', '0', '0.1068', '0.2203', '0.3271'),
4         ('2', '0', '0', '0.1145', '0.1145'),
5         ('g(x)', '0.1101', '0.3298', '0.5601', '1.00')]
6 heads = ['X/Y', '0', '1', '2', 'h(y)']
```

In [14]:
```
print(tabulate(data, headers = heads, tablefmt = 'github'))
```

| X/Y | 0 | 1 | 2 | h(y) |
|---|---|---|---|---|
| 0 | 0.1101 | 0.223 | 0.2253 | 0.5584 |
| 1 | 0 | 0.1068 | 0.2203 | 0.3271 |
| 2 | 0 | 0 | 0.1145 | 0.1145 |
| g(x) | 0.1101 | 0.3298 | 0.5601 | 1 |

g(x) and h(y) are the marginal probabilities of X and Y respectively.

In [15]:
```
# Marginal of X
dataX =[('g(x)', '0.1101', '0.3298', '0.5601')]
heads = ['X', '0', '1', '2']
print(tabulate(dataX, headers = heads, tablefmt = 'github'))
```

| X | 0 | 1 | 2 |
|---|---|---|---|
| g(x) | 0.1101 | 0.3298 | 0.5601 |

In [16]:
```
# Marginal of Y
dataY =[('h(y)', '0.5584', '0.3271', '0.1145')]
heads = ['Y', '0', '1', '2']
print(tabulate(dataY, headers = heads, tablefmt = 'github'))
```

| Y | 0 | 1 | 2 |
|---|---|---|---|
| h(y) | 0.5584 | 0.3271 | 0.1145 |

In [17]:
```
# Covariance and Correlation plot
df = data = np.array([x,y])
covMatrix = np.cov(df,bias=True)
sn.heatmap(covMatrix, annot=True, fmt='g')
plt.show()
```