Homework 4

In [2]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

1. Let p denote the probability that a particular item A appears in a simple random sample (SRS). Suppose we collect 5 independent simple random samples, i.e., each SRS is obtained by drawing from the entire population. Let X denote the random variable for the total number of times that A appears in these 5 samples. What is the expected value of X, i.e., $\mathbb{E}[X]$? Note that your answer should be in terms of p.

ANSWER: The expected value of X (E(X)) is summation of all the p, so p+p+p+p+p=5p, which is shown in a Bernoulli's distribution, since this p describes the probability of the item A appearing in ONE simple random sample. Since you have five simple random samples, you will have a possibility of drawing A as 5p.

2. What is $Var(X)$? Again, your answer should be in terms of p.

In [3]:
```
'''Var(X) in terms of p is the summation of p(q) in five SRS, which is 5*p*q or 5*p*(1-p) '''
```

Out[3]:
```
'Var(X) in terms of p is the summation of p(q) in five SRS, which is 5*p*q or 5*p*(1-p) '
```

3. Consider rolling (independently) one fair six-sided die and one loaded six-sided die.

Let X_1 and X_2 denote, respectively, the number of spots from one roll of the fair die and one roll of the loaded die. Suppose the distribution for the loaded die is

$$\begin{aligned}\Pr(X_2 = 1) = \Pr(X_2 = 2) &= \frac{1}{16}\\ \Pr(X_2 = 3) = \Pr(X_2 = 4) &= \frac{3}{16}\\ \Pr(X_2 = 5) = \Pr(X_2 = 6) &= \frac{4}{16}.\end{aligned}$$

Let $Y = X_1X_2$ denote the product of the two numbers of spots.

a. What is the expected value of Y?

In [4]:
```
'''Since you are independently rolling one fair six-sided die and one loaded six-sided die, you get the given results:
If you refer to X1, for each roll of the fair die, you have a 1/6 chance of getting that number chosen, then the estimated
value is: (1/6)(1+2+3+4+5+6)=3.5

With regards to X2, the expected value is: (1/16)(1+2)+(3/16)(3+4)+(4/16)(5+6)= 4.25

so, since Y= 3.5(4.25)= 14.875 or 14.9 or 15.


'''
```

Out[4]:
```
'Since you are independently rolling one fair six-sided die and one loaded six-sided die, you get the given results: \nIf you refer to X1, for each roll of the fair die, you have a 1/6 chance of getting that number chosen, then the estimated \nvalue is: (1/6)(1+2+3+4+5+6)=3.5\n\nWith regards to X2, the expected value is: (1/16)(1+2)+(3/16)(3+4)+(4/16)(5+6)= 4.25\n\nso, since Y= 3.5(4.25)= 14.875 or 14.9 or 15. \n\n\n'
```

b. What is the variance of Y?

In [5]:
```
'''To start, the mean for Y is ((1/6)*1 + ((1/16)*1)) + ((1/6)*2 + ((1/16)*2)) + ((1/6)*3 + ((3/16)*3)) +
 ((1/6)*4 + ((3/16)*4))
+ ((1/6)*5 + ((4/16)*5))+((1/6)*6 + ((4/16)*6)) =  7.75/6 = 1.29

To get the Var(Y), you would do Var(X1), which is (1/6)(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2) = 91/6 = 15.16667
To get the Var(X2), which is (1/16)(1^2 + 2^2) + (3/16) (3^2 + 4^2) + (4/16) (5^2 + 6^2) = 20.5


20.5(15.677777)-14.875^2= ~89.65 as the Var(Y)

'''
```

Out[5]:
```
'To start, the mean for Y is ((1/6)*1 + ((1/16)*1)) + ((1/6)*2 + ((1/16)*2)) + ((1/6)*3 + ((3/16)*3)) + ((1/
6)*4 + ((3/16)*4))\n+ ((1/6)*5 + ((4/16)*5))+((1/6)*6 + ((4/16)*6)) =  7.75/6 = 1.29 \n\nTo get the Var(Y), y
ou would do Var(X1), which is (1/6)(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2) = 91/6 = 15.16667 \nTo get the Var(X
2), which is (1/16)(1^2 + 2^2) + (3/16) (3^2 + 4^2) + (4/16) (5^2 + 6^2) = 20.5 \n \n \n20.5(15.677777)-14.87
5^2= ~89.65 as the Var(Y)\n\n'
```

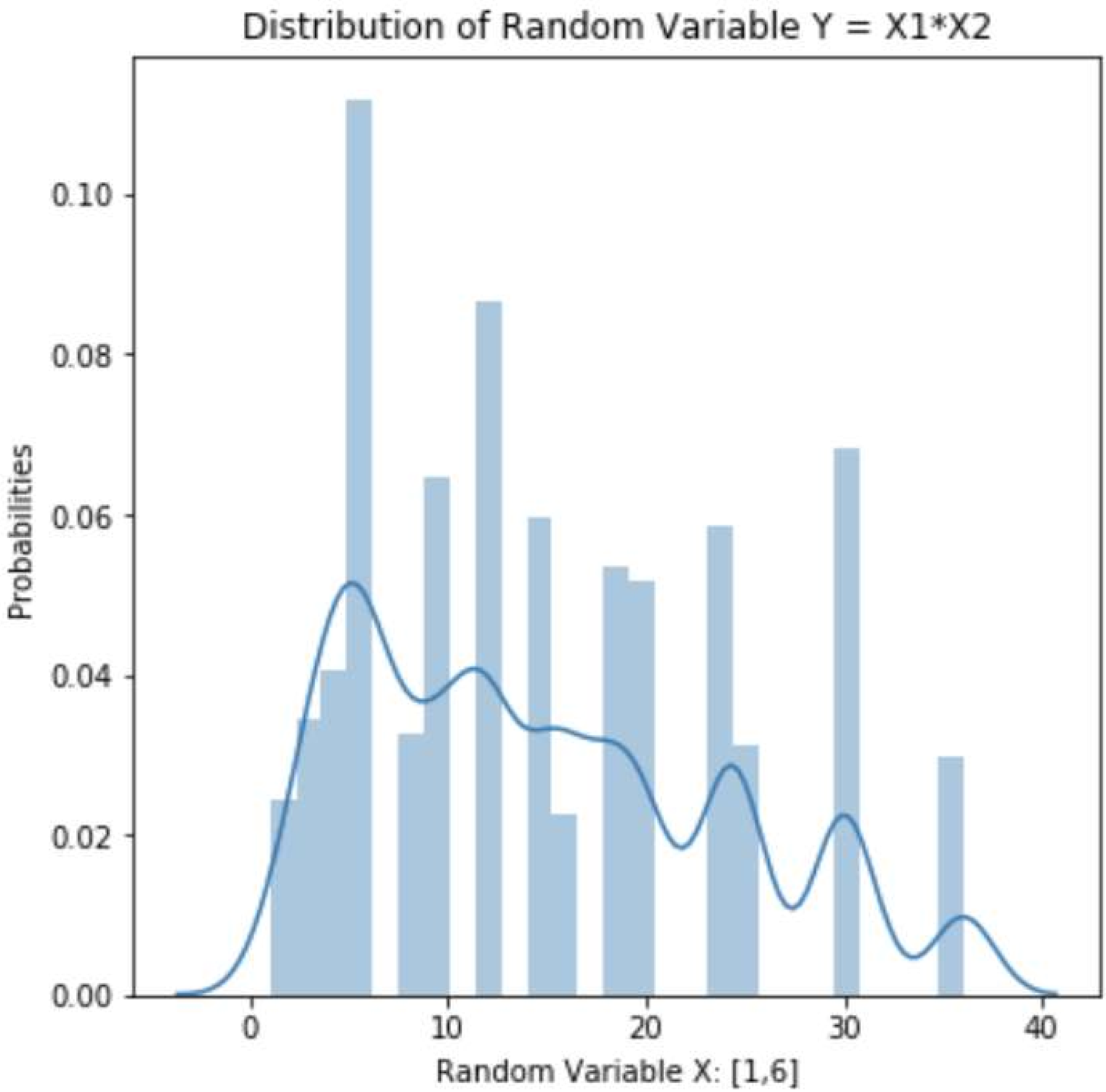
c. Estimate the sampling distribution of Y by simulating 10,000 rolls of the pair of dice. Provide a graphical display of the distribution. Compare the mean and variance from this estimate to the values you computed above.

The plot should look something like:

In [6]:
```
from IPython.display import Image

Image('images/samplingdist.png')
```

Out[6]:

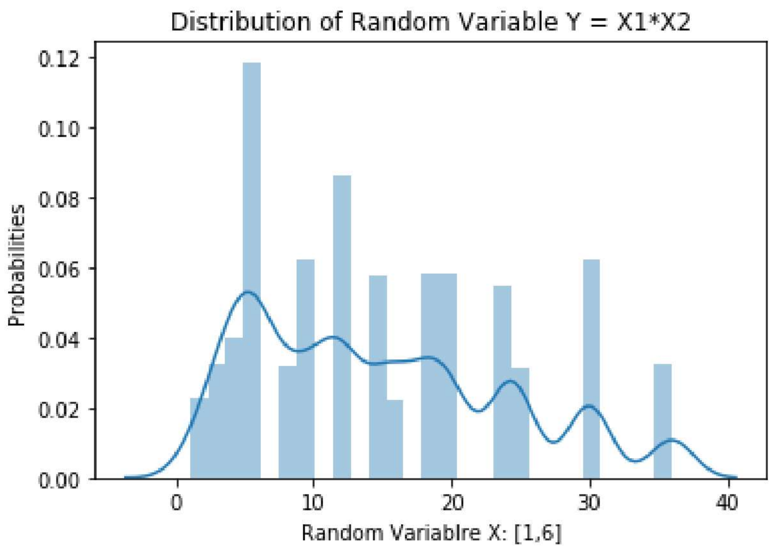

Distribution of Random Variable Y = X1*X2
Probabilities
0.00
0.02
0.04
0.06
0.08
0.10
0
10
20
30
40
Random Variable X: [1,6]

The mean and variance from the sampling distribution should be similar to the values calculated in the previous two parts.

In [20]:
```
### BEGIN SOLUTION
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import binom
import scipy.stats as stats

x1s = np.random.choice([1,2,3,4,5,6],replace=True, size=10000, p=(1/6,1/6,1/6,1/6,1/6,1/6))
x2s = np.random.choice([1,2,3,4,5,6],replace=True, size=10000, p=(1/16,1/16,3/16,3/16,4/16,4/16))
ys = x1s * x2s
print(np.mean(ys))
print(np.var(ys))
sns.distplot(ys)
plt.ylabel('Probabilities');
plt.xlabel('Random Variablre X: [1,6]');
plt.title('Distribution of Random Variable Y = X1*X2');
### END SOLUTION
```

```
14.8061
85.53310279
```



4. Suppose we flip a fair coin 10 times. The probability that all coin flips will be heads is less than 0.001. However, if we repeat the 10 coin flips 1000 times, then the probability that we obtain all heads at least once is about 0.62!

We want to run many replications simulating this experiment, flipping 10 fair coins 1000 times each, in order to better understand the probability of getting all heads.

- Use `np.random.binomial` to repeat 10 flips of a fair coin 1000 times, for 10000 replications
- For each of the 10000 replications, count the number of times you obtain all heads.
- Compute the frequency of getting all heads 0 times, 1 time, 2 times, etc. You should obtain something like the following:

In [8]:
```
Image('images/table.PNG')
```

Out[8]:

	frequency
0	0.3765
1	0.3721
2	0.1779
3	0.0566
4	0.0140
5	0.0025
6	0.0004

In [19]:
```
### BEGIN SOLUTION

freq_list = []
num_10s = 0

for i in range(10000):
    freq_vals = np.random.binomial(10, .5, 1000)
    num_10s = 0
    for j in freq_vals:
        if j == 10:
            num_10s += 1
    freq_list.append(num_10s)

proportions = []
for m in range(7):
    count_occur = 0
    for n in range(10000):
        if freq_list[n] == m:
            count_occur += 1
    proportions.append(count_occur/10000)


coin_freq = pd.DataFrame(proportions, columns = [['frequency']])
coin_freq.loc[:7, ['frequency']]




### END SOLUTION
```

Out[19]:

	frequency
0	0.3821
1	0.3667
2	0.1735
3	0.0593
4	0.0145
5	0.0032
6	0.0005

In [ ]: