Homework 4

In [1]:
```
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.

5p

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

5p(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

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

Let $Y = X_1 X_2$ denote the product of the two numbers of spots.

a. What is the expected value of Y?

E[Y] = E[X1]E[X2] = (1/6)*21* {(1/16)*3 + (3/16)*7 + (4/16)*11}

```
= 14.875
```

b. What is the variance of Y?

Var(Y) = Var(X1)Var(X2)+Var(X1)(E(X2))^2+Var(X2)(E(X1))^2

```
= ((35/12)*(35/16)) + ((35/12)*(4.25*4.25)) + ((35/16) * (3.5*3.5))

= 85.859375
```

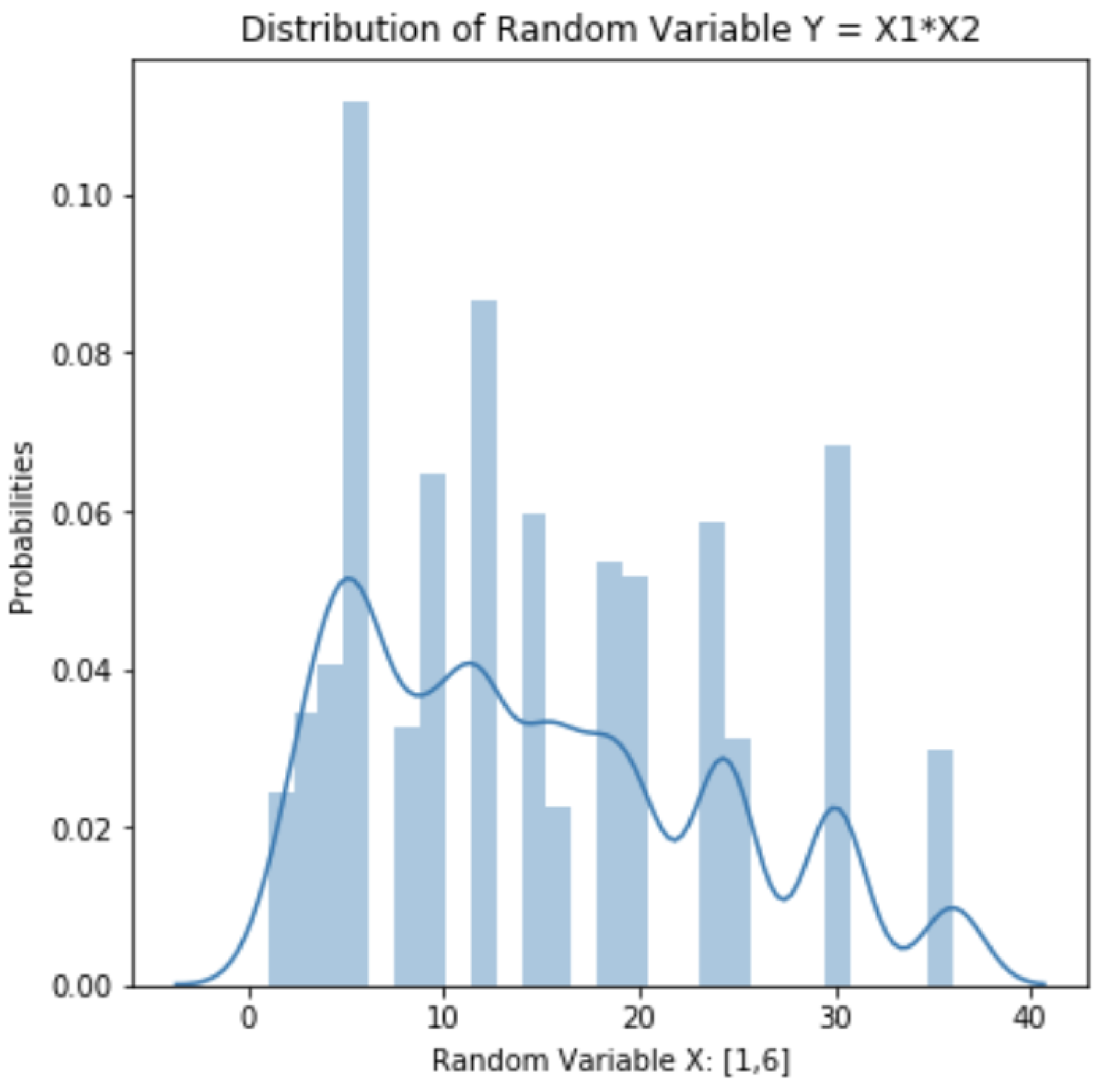
c. Estimate the sampling distribution of Y by simulating 10,000 rolls of the pair of dice. Provide a graphical display of the distribution. Compare the mean and variance from this estimate to the values you computed above.

The plot should look something like:

In [4]:
```
from IPython.display import Image

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

Out[4]:

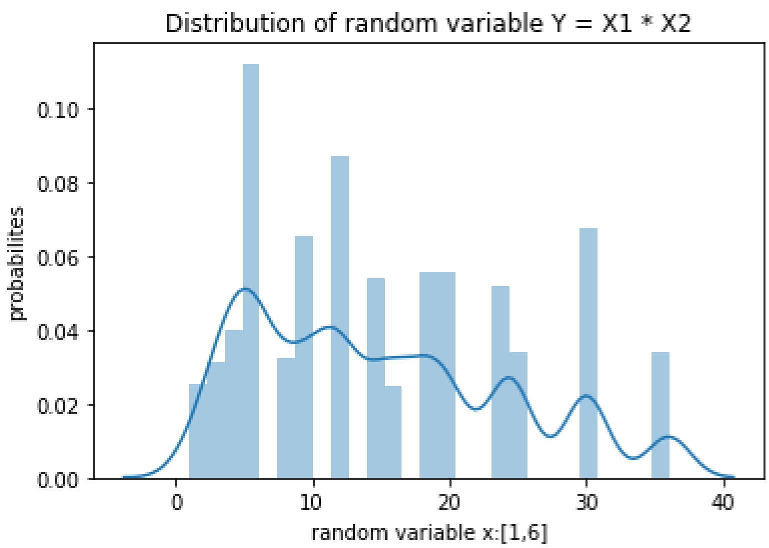


The mean and variance from the sampling distribution should be similar to the values calculated in the previous two parts.

In [12]:
```
import numpy as np
import seaborn as sns

die1 = np.random.choice(7, 10000, p=[0,(1/6),(1/6),(1/6),(1/6),(1/6),(1/6)])
die2 = np.random.choice(7, 10000, p=[0,(1/16),(1/16),(3/16),(3/16),(4/16),(4/1
6)])
dice = die1 * die2
histo = sns.distplot(dice)
histo.set_xlabel('random variable x:[1,6]')
histo.set_ylabel('probabilites')
histo.set_title('Distribution of random variable Y = X1 * X2')
```

Out[12]:
```
Text(0.5, 1.0, 'Distribution of random variable Y = X1 * X2')
```



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 [1]:
```
Image('images/table.PNG')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-e40721009109> in <module>
----> 1 Image('images/table.PNG')

NameError: name 'Image' is not defined
```

In [14]:
```
import pandas as pd

head_list = []
for i in range(10000):
    vals = np.random.binomial(10,.5,1000)
    sum = 0
    for i in vals:
        if i == 10:
            sum += 1
    head_list.append(sum)
head = pd.DataFrame(head_list)
head[0] = head[0].value_counts()/10000
head = head.dropna()
display(head)
```

	0
0	0.3797
1	0.3652
2	0.1810
3	0.0571
4	0.0140
5	0.0026
6	0.0003
7	0.0001

In [ ]: