

# Use\_PY\_in\_Advanced\_Statistics

June 28, 2018

## 1 Use Python in Advanced Statistics

*python*

### 1.1 Chapter One Probability

#### 1.1.1 Random Experiment and Sample Space

```
* * *  
* * //10
```

#### Sample Space

- Scapital S
- $S^e$
- 

- $A \subseteq B$ ;
- $A = B$ ;
- 
- $A \cup B$
- $A - B, AB$
- 

#### 1.1.2

01

$A \cap P(A) = P \cap A P(A)$  1.  $P(A) \geq 0$ ; 2.  $P(S) = 1$ ; 3.  $A_1, A_2, \dots, A_i A_j = \emptyset, i \neq j$   $P(\cup A_i) = \sum P(A_i)$

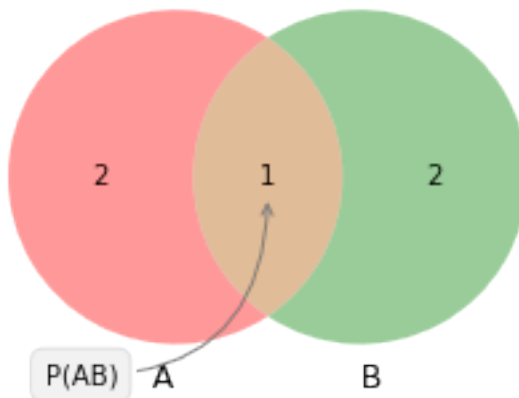
**Conditional Probability:**  $P(A|B)$  BAABSBB  
 plot one plot two

```
In [51]: from matplotlib import pyplot as plt
import numpy as np
import sympy

from matplotlib_venn import venn3, venn3_circles
plt.figure(figsize=(4,4))
v = venn2(subsets=(2,2,1), set_labels = ('A', 'B'))

plt.title("Sample Venn diagram - plot one")
plt.annotate('P(AB)', xy=v.get_label_by_id('11').get_position() - np.array([0, 0.05]),
             ha='center', textcoords='offset points', bbox=dict(boxstyle='round,pad=0'),
             arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0.5',color='g'))
plt.show()
```

Sample Venn diagram - plot one



```
In [52]: from matplotlib import pyplot as plt
import numpy as np
import sympy

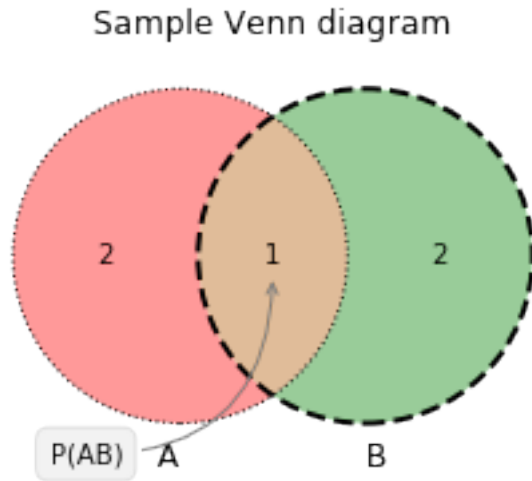
from matplotlib_venn import venn3, venn3_circles
plt.figure(figsize=(4,4))
v = venn2(subsets=(2,2,1), set_labels = ('A', 'B'))

c = venn2_circles(subsets=(2, 2, 1), linestyle='dashed')
```

```

c[0].set_lw(1.0)
c[0].set_ls('dotted')
plt.title("Sample Venn diagram")
plt.annotate('P(AB)', xy=v.get_label_by_id('11').get_position() - np.array([0, 0.05])
             ha='center', textcoords='offset points', bbox=dict(boxstyle='round,pad=0',
             arrowprops=dict(arrowstyle='->', connectionstyle='arc3,rad=0.5',color='g')
plt.show()

```



$$P(B|A) = \frac{P(AB)}{P(A)} \quad P(A|B) = \frac{P(AB)}{P(B)}$$

$S = \{ (, ), (, ), (, ), (, ) \}$

$B = \{ (, ), (, ), (, ) \}$

$A = \{ (, ) \}$

$$P(A|B) = \frac{1}{3} \quad P(A) = \frac{1}{4} \quad P(A) \neq P(A|B) \quad S_B = B$$

## Random Variable

- 
- 

$S \ni X(e)$

- 
- 
- 
- 
- 
- 

• \* \* \*  
• \* \* \*

### 1.1.3 Random Variable

""" 1 > 3000X > > >

\$X = \$

$X\{X > 10000\}\{X < 3000\}$

The Classification of the Random Variable    Discrete Random Variable    Continuous Random Variable  
##### Discrete Random Variable

```
In [66]: import numpy as np
         from scipy import stats
         import matplotlib.pyplot as plt

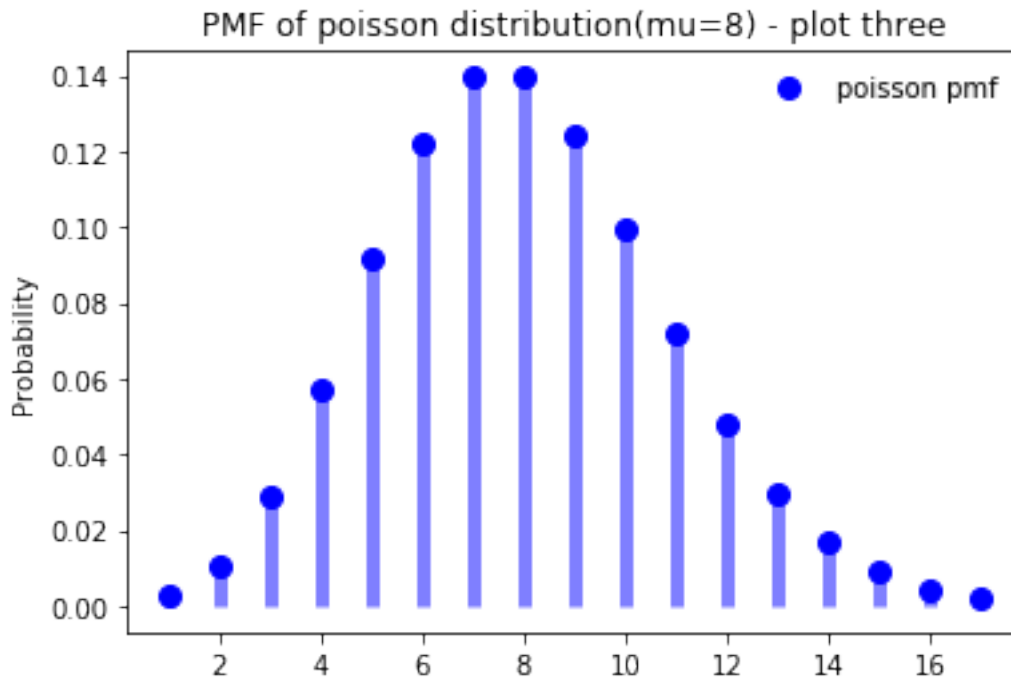
         def poisson_pmf(mu=3):

             poisson_dis = stats.poisson(mu)
             x = np.arange(poisson_dis.ppf(0.001), poisson_dis.ppf(0.999))
             print(x)

             fig, ax = plt.subplots(1, 1)
             ax.plot(x, poisson_dis.pmf(x), 'bo', ms=8, label='poisson pmf')
             ax.vlines(x, 0, poisson_dis.pmf(x), colors='b', lw=5, alpha=0.5)
             ax.legend(loc='best', frameon=False)
             plt.ylabel('Probability')
             plt.title('PMF of poisson distribution(mu={}) - plot three'.format(mu))
             plt.show()

         poisson_pmf(mu=8)

[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15.
 16. 17.]
```



```
In [65]: def binom_pmf(n=1, p=0.1):
    binom_dis = stats.binom(n, p)
    x = np.arange(binom_dis.ppf(0.0001), binom_dis.ppf(0.9999))
    print(x)

    fig, ax = plt.subplots(1, 1)
    ax.plot(x, binom_dis.pmf(x), 'bo', label='binom pmf')
    ax.vlines(x, 0, binom_dis.pmf(x), colors='b', lw=5, alpha=0.5)
    ax.legend(loc='best', frameon=False)
    plt.ylabel('Probability')
    plt.title('PMF of binomial distribution(n={}, p={}) - plot four'.format(n, p))

    plt.show()

    binom_pmf(n=20, p=0.6)
```

[ 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18.]