

9_Calculating Chi-Square Test

9.1_To check goodness of data

```
In [1]: import numpy as np
```

```
In [4]: ob = np.array([22,17,20,26,22,13])  
ex = np.array([20,20,20,20,20,20])
```

```
In [5]: ob-ex
```

```
Out[5]: array([ 2, -3,  0,  6,  2, -7])
```

```
In [9]: np.sum(np.square(ob-ex)/ex)
```

```
Out[9]: 5.1000000000000005
```

9.2_To check dependency of variables

```
In [22]: row1 = np.array([40,45,25,10])  
row2 = np.array([35,30,20,30])
```

```
In [25]: sum_r1 = np.sum(row1)  
sum_r2 = np.sum(row2)  
sum_row = np.array([sum_r1, sum_r2])  
sum_row
```

```
Out[25]: array([120, 115])
```

```
In [26]: sum_col = row1 + row2  
sum_col
```

```
Out[26]: array([75, 75, 45, 40])
```

```
In [32]: exp = []  
for i in sum_row:  
    for j in sum_col:  
        exp.append(i*j/235)  
print(exp)
```

```
[38.297872340425535, 38.297872340425535, 22.97872340425532, 20.425531914893618, 36.7  
02127659574465, 36.702127659574465, 22.02127659574468, 19.574468085106382]
```

```
In [34]: # join both columns for observed values  
obj = np.array([40,45,25,10,35,30,20,30])
```

```
In [36]: np.sum(np.square(obj - exp)/exp)
```

Out[36]: 13.788747987117553

In []:

