



IIT ROORKEE



NPTEL ONLINE
CERTIFICATION COURSE

χ^2 Test of Independence - II

Dr. A. Ramesh

DEPARTMENT OF MANAGEMENT STUDIES



Agenda

- Using python to test the independence of variables
- Understanding goodness of fit test for Poisson

Example

- Record of 50 students studying in ABN School is taken at random, the first 10 entries are like this:

res_num	aa	pe	sm	ae	r	g	c
1	99	19	1	2	0	0	1
2	46	12	0	0	0	0	0
3	57	15	1	1	0	0	0
4	94	18	2	2	1	1	1
5	82	13	2	1	1	1	1
6	59	12	0	0	2	0	0
7	61	12	1	2	0	0	0
8	29	9	0	0	1	1	0
9	36	13	1	1	0	0	0
10	91	16	2	2	1	1	0

Example

Here :

- res_num = registration no.
- aa= academic ability
- pe = parent education
- sm = student motivation
- r = religion
- g = gender

Python code

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

```
In [2]: acad = pd.read_csv('AcademicAbilityData.csv')
```

```
In [3]: acad
```

Out[3]:

	res_num	aa	pe	sm	ae	r	g	c
0	1	99	19	1	2	0	0	1
1	2	46	12	0	0	0	0	0
2	3	57	15	1	1	0	0	0
3	4	94	18	2	2	1	1	1
4	5	82	13	2	1	1	1	1
5	6	59	12	0	0	2	0	0
6	7	61	12	1	2	0	0	0
7	8	29	9	0	0	1	1	0
8	9	36	13	1	1	0	0	0
9	10	91	16	2	2	1	1	0
10	11	55	10	0	0	1	0	0
11	12	58	11	0	1	0	0	0

Hypothesis

- Test the hypothesis that “gender and student motivation” are independent

Python code

```
In [19]: #Cross table between gender and student's motivation  
obs =pd.pivot_table(acad[['g','sm']],index = 'g',columns='sm',aggfunc=len)  
obs
```

Out[19]:

	sm	0	1	2
g				
0	10	13	6	
1	4	9	8	

Observed values

Gender	Student motivation			
	0 (Disagree)	1 (Not decided)	2 (Agree)	Row Sum
→ 0 (Male)	10	13	6	<u>29</u>
→ 1(Female)	4	9	8	<u>21</u>
Column Sum	<u>14</u>	<u>22</u>	<u>14</u>	50

Expected frequency (contingency table)

Gender	Student motivation		
	0	1	2
0	$29 \times 14 / 50 =$ 8.12	<u>12.76</u>	<u>8.12</u>
1	<u>5.88</u>	<u>9.24</u>	<u>5.88</u>

Frequency Table

Gender	Student motivation		
	0	1	2
0	$f_o = 10$ $f_e = 8.12$	$f_o = 13$ $f_e = 12.76$	$f_o = 6$ $f_e = 8.12$
1	$f_o = 4$ $f_e = 5.88$	$f_o = 9$ $f_e = 9.24$	$f_o = 8$ $f_e = 5.88$

Chi sq. calculation

$$\chi^2 = \sum \sum \left(\frac{f_o - f_e}{f_e} \right)^2$$

$$= 0.435 + 0.005 + 0.554 + 0.601 + 0.006 + 0.764$$

$$= \underline{2.365}$$

Python code

```
In [11]: ## Perform chi2 test to check independence  
from scipy.stats import chi2_contingency
```

```
In [14]: chi2, p, dof, tbl = chi2_contingency(obs)
```

```
In [15]: chi2
```

```
Out[15]: 2.3649585225939904
```

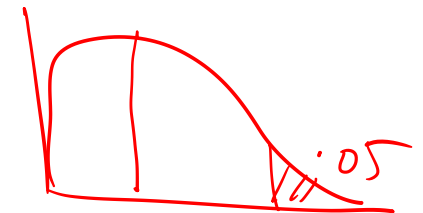
```
In [16]: p
```

```
Out[16]: 0.3065178579178871
```

```
In [17]: dof
```

```
Out[17]: 2
```

$$\alpha = 5\% = 0.05$$



$$(2-1)(3-1) = 2$$

Python code

```
In [11]: ## Perform chi2 test to check independence  
from scipy.stats import chi2_contingency
```

```
In [14]: chi2, p, dof, tbl = chi2_contingency(obs)
```

```
In [15]: chi2
```

```
Out[15]: 2.3649585225939904
```

```
In [16]: p
```

```
Out[16]: 0.3065178579178871
```

```
In [17]: dof
```

```
Out[17]: 2
```

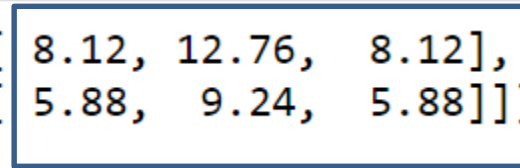
Degrees of
freedom =
 $(2-1)*(3-1)$

Python code

```
In [12]: ▶ tbl
```

```
Out[12]: array([[ 8.12, 12.76,  8.12],  
                [ 5.88,  9.24,  5.88]])
```

Contingency
table



χ^2 Goodness of Fit Test



χ^2 Goodness-of-Fit Test

- The χ^2 goodness-of-fit test compares *expected* (theoretical) *frequencies* of categories from a population distribution to the *observed* (actual) *frequencies* from a distribution to determine whether there is a difference between what was expected and what was observed

χ^2 Goodness-of-Fit Test

$$\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

$$df = k - 1 - p$$

where: f_o = frequency of observed values

f_e = frequency of expected values

k = number of categories

p = number of parameters estimated from the sample data

Goodness of Fit Test: Poisson Distribution

1. Set up the null and alternative hypotheses.

H_0 : Population has a Poisson probability distribution

H_a : Population does not have a Poisson distribution

2. Select a random sample and

- Record the observed frequency f_i for each value of the Poisson random variable.
- Compute the mean number of occurrences μ .

3. Compute the expected frequency of occurrences e_i for each value of the Poisson random variable.

Goodness of Fit Test: Poisson Distribution

4. Compute the value of the test statistic

$$\chi^2 = \sum_{i=1}^k \frac{(f_i - e_i)^2}{e_i}$$

where:

f_i = observed frequency for category i

e_i = expected frequency for category i

k = number of categories

Goodness of Fit Test: Poisson Distribution

5. Rejection rule:

p -value approach: Reject H_0 if $p\text{-value} \leq \alpha$

Critical value approach: Reject H_0 if $\chi^2 \geq \chi_\alpha^2$

where α is the significance level and
there are $k - 2$ degrees of freedom

$$k - 1 - \rho$$

Goodness of Fit Test: Poisson Distribution

- Example: Parking Garage

In studying the need for an additional entrance to a city parking garage, a consultant has recommended an analysis, that approach is applicable only in situations where the number of cars entering during a specified time period follows a Poisson distribution.

Goodness of Fit Test: Poisson Distribution

A random sample of 100 one- minute time intervals resulted in the customer arrivals listed below. A statistical test must be conducted to see if the assumption of a Poisson distribution is reasonable.

# Arrivals	0	1	2	3	4	5	6	7	8	9	10	11	12
Frequency	0	1	4	10	14	20	12	12	9	8	6	3	1

Goodness of Fit Test: Poisson Distribution

- Hypotheses

H_0 : Number of cars entering the garage during a one-minute interval is Poisson distributed

H_a : Number of cars entering the garage during a one-minute interval is not Poisson distributed

Python Code

```
In [1]: import scipy  
        from scipy.stats import chi2  
        from scipy.stats import poisson
```

```
In [2]: import pandas as pd  
        import numpy as np
```

```
In [3]: data = pd.read_excel('P_distribution.xlsx')  
        data
```

```
Out[3]:
```

	Arrivals	Frequency
0	0	0
1	1	1
2	2	4
3	3	10
4	4	14
5	5	20
6	6	12
7	7	12
8	8	9
9	9	8
10	10	6
11	11	3
12	12	1

Goodness of Fit Test: Poisson Distribution

- Estimate of Poisson Probability Function

$$\text{Total Arrivals} = 0(0) + 1(1) + 2(4) + \dots + 12(1) = 600$$

$$\text{Estimate of } \mu = 600/100 = 6$$

$$\text{Total Time Periods} = 100$$

Hence,

$$= \frac{e^{-\mu} \mu^x}{x!}$$

$$f(x) = \frac{6^x e^{-6}}{x!}$$

$$\mu = \frac{\sum f n}{\sum f} = \frac{600}{100} = 6$$

Goodness of Fit Test: Poisson Distribution

- Expected Frequencies

x	$f(x)$	$nf(x)$	x	$f(x)$	$nf(x)$
0	.0025	<u>.25</u>	7	.1377	13.77
1	.0149	<u>1.49</u>	8	.1033	10.33
2	.0446	<u>4.46</u>	9	.0688	6.88
3	.0892	<u>8.92</u>	10	.0413	4.13
4	.1339	<u>13.39</u>	11	.0225	2.25
5	.1606	<u>16.06</u>	<u>12+</u>	<u>.0201</u>	<u>2.01</u>
6	.1606	<u>16.06</u>	Total	1.0000	100.00

Python code

```
In [4]: Observed_Freq = data['Frequency']
```

```
In [5]: total_arrival = 600  
total_time_period = 100  
mu = total_arrival/total_time_period
```

```
In [6]: Expected_Freq = []  
for i in range(len(Observed_Freq)):  
    E_Freq = 100*poisson.pmf(i, mu)  
    Expected_Freq.append(E_Freq)
```

```
In [7]: Expected_Freq
```

```
Out[7]: [0.24787521766663584,  
1.4872513059998145,  
4.461753917999444,  
8.923507835998894,  
13.385261753998332,  
16.062314104797995,  
16.06231410479801,  
13.767697804112569,  
10.32577335308442,  
6.883848902056284,  
4.130309341233764,  
2.2528960043093247,  
1.1264480021546681]
```

Python code

```
In [8]: Expected_Freq_round_off = [round(elem, 2) for elem in Expected_Freq]
Expected_Freq_round_off
```

```
Out[8]: [0.25,
1.49,
4.46,
8.92,
13.39,
16.06,
16.06,
13.77,
10.33,
6.88,
4.13,
2.25,
1.13]
```

```
In [9]: df = pd.DataFrame(list(zip(Observed_Freq, Expected_Freq_round_off)), columns = ['Observed Frequency', 'Expected Frequency'])
df
```

```
Out[9]:
```

	Observed Frequency	Expected Frequency
0	0	0.25
1	1	1.49
2	4	4.46
3	10	8.92
4	14	13.39
5	20	16.06
6	12	16.06
7	12	13.77
8	9	10.33
9	8	6.88
10	6	4.13
11	3	2.25
12	1	1.13

Goodness of Fit Test: Poisson Distribution

- Observed and Expected Frequencies

i	f_i	e_i	$f_i - e_i$
<u>0 or 1 or 2</u>	<u>5</u>	<u>6.20</u>	-1.20
3	10	8.92	1.08
4	14	13.39	0.61
5	20	16.06	3.94
6	12	16.06	-4.06
7	12	13.77	-1.77
8	9	10.33	-1.33
9	8	6.88	1.12
10 or more	10	<u>8.39</u>	1.61

Python code

```
In [10]: obs_freq = [5, 10, 14, 20, 12, 12, 9, 8, 10]  
         expected_freq = [6.20, 8.92, 13.39, 16.06, 16.06, 13.77, 10.33, 6.88, 8.39]
```

```
In [11]: scipy.stats.chisquare(obs_freq, expected_freq)
```

```
Out[11]: Power_divergenceResult(statistic=3.2738182931105193, pvalue=0.916017731732134)
```

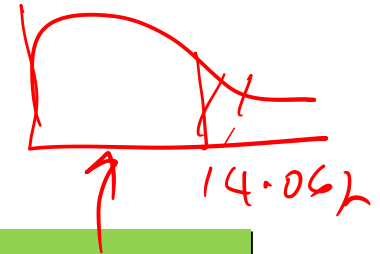
Goodness of Fit Test: Poisson Distribution

- Rejection Rule

With $\alpha = .05$ and $\underline{k} - p - 1 = 9 - 1 - 1 = 7$ d.f.
(where k = number of categories and p = number of population parameters estimated),

$$\chi^2_{.05} = 14.067$$

Reject H_0 if $p\text{-value} \leq .05$ or $\chi^2 \geq 14.067$.



- Test Statistic

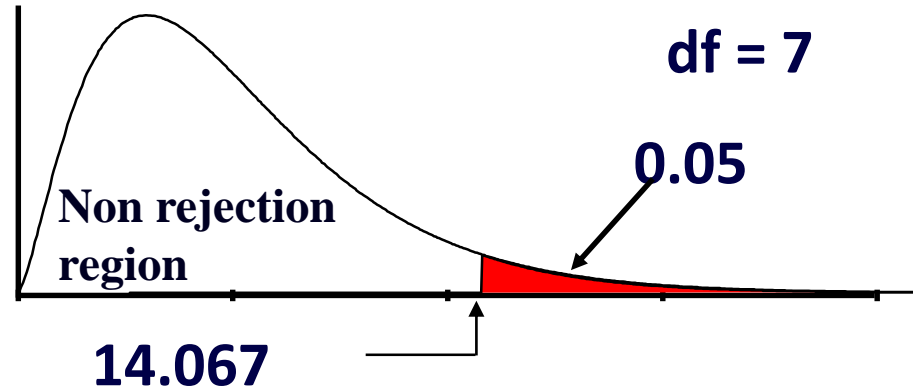
$$\chi^2 = \frac{(-1.20)^2}{6.20} + \frac{(1.08)^2}{8.92} + \dots + \frac{(1.61)^2}{8.39} = 3.268$$

Python code

```
In [4]: from scipy.stats import chi2  
chi2.ppf(0.95,7)
```

```
Out[4]: 14.067140449340167
```


Goodness of Fit Test: Poisson Distribution



$$\chi_{Cal}^2 = \underline{3.268} < 14.067, \text{ do not reject } H_0.$$

Thank You

