Categorical Data

- Overview
  - Categorical data is divided into several groups.
    -. nominal data
    -. ordinal data
  - Numerical data can be also treated as categorical data by the class

Contingency Table

- Contingency Table
  - The first row and the first column indicate different levels of the two different variables.
  - The other cells show the frequencies of corresponding levels.

X Y	level 1	level 2	level 3	Total
level 1	n_{11}	n_{12}	n_{13}	$n_{1.}$
level 2	n_{21}	n_{22}	n_{23}	$n_{2.}$
Total	$n_{.1}$	$n_{.2}$	$n_{.3}$	$n_{..} = N$

Contingency Table

- Frequency
  - Observed frequency (O_{ij})
    : The counts for the given levels from the actual data
  - Expected frequency (E_{ij})
    : The expected frequency for the given levels when H_0 is true

Contingency Table

- Expected Frequency
  - The expected probability at the i th row ($P_{i.}$)

$$p_{i.} = \frac{n_{i.}}{N}$$

  - The expected probability at the j th column ($P_{.j}$)

$$p_{.j} = \frac{n_{.j}}{N}$$

  - The expected frequency in the cell of the i th row and and the j th column

$$n_{ij} = N \cdot p_{i.} \cdot p_{.j} = \frac{n_{i.}n_{.j}}{N}$$

Chi-square Tests

- Chi-square tests

  - It computes test statistic by comparing the observed frequencies and the expected frequencies in the cells.

  - Test statistic (T)

$$T = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi^2_{(r-1)\cdot(c-1)}$$

  - O_{ij}: observed frequency, E_{ij}: expected frequency,
    r : the number of rows, c : the number of columns

Chi-square Tests

- Likelihood ratio tests

  - If n is large enough, G^2 also follows chi-square distribution

  - Test statistic (T)

$$G^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\log(\frac{O_{ij}}{E_{ij}}) \sim \chi^2_{(r-1)\cdot(c-1)}$$

  - O_{ij}: observed frequency, E_{ij}: expected frequency,
    r : the number of rows, c : the number of columns

Contingency Table

pd.crosstab (index, columns, margins, margins_name)

-. *Index: values to group by in the rows*
-. *columns: values to group by in the columns*
-. *margines = True or False: If true, show the total*
-. *margins_name : the names of row and column margins.*

Contingency Table

- Example
  - Summarize ‘edu_income9.csv’ using contingency table. The row is based on ‘education’ and column is based on ‘income’ in the table.

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

In [2]: data1 = pd.read_csv("edu_income9.csv")
        data1.head()

Out[2]:
```

	education	income
0	college	high
1	college	high
2	college	high
3	college	high
4	college	high

Contingency Table

- Example (continue..)
  - Contingency table:

```
In [10]: table1 = pd.crosstab(index = data1['education'], columns = data1['income'],margins=True, margins_name="Total")
         table1
Out[10]:
```

income education	high	low	midium	Total
college	255	81	105	441
high-school	111	65	93	269
middle-school	90	86	114	290
Total	456	232	312	1000

Contingency Table

from scipy.stats import chi2_contingency

chi2_contingency (observed)

-. *observed: the contingency table*
-. *Returns :*
  (1) chi2 : the test statistic
  (2) p : p-value
  (3) dof : degrees of freedom
  (4) expected : the expected frequencies

Contingency Table

- Example
  - With the previous example, test whether the income levels are different by the education at $\alpha = 0.05$.

$$H_0\text{: } (p_{college,high}, p_{college,mid}, p_{college,low})$$
$$= (p_{high-schoo\ ,high}, p_{high-schoo\ ,mid}, p_{high-school,low})$$
$$= (p_{middle-schoo\ ,high}, p_{middle-school,mid}, p_{middle-schoo\ ,low})$$

Contingency Table

- Example (continue..)

```
In [18]: from scipy.stats import chi2_contingency

         chi2, pval, dof, expected = chi2_contingency(table1)

In [19]: print('Test statistic: ',np.round(chi2,4))
         print('p-value :', np.round(pval,6))
         print('Degrees of freedom :', dof)
         print('Expected Freq :', expected)

Test statistic:  53.6209
p-value : 0.0
Degrees of freedom : ~~3~~ 4
Expected Freq : [[ 201.096  102.312  137.592  441.   ]
 [ 122.664   62.408   83.928  269.   ]
 [ 132.24    67.28    90.48   290.   ]
 [ 456.     232.     312.    1000.   ]]
```

Contingency Table

pd.pivot_table (data, index, columns, values, aggfunc, margins)
: create a spread-sheet style pivot table

-. *data: data-frame*
-. *index: keys to group by on the pivot table index*
-. *columns: keys to group by on the pivot table columns*
-. *values : observed frequencies*
-. *aggfunc: a list of functions*
-. *margins =True or False: If true, compute the total*

Chi-square Tests

- Example
  - The table shows the frequencies of the participants in the school festival by the grade. Test whether the proportions are the same by the grade at $\alpha = 0.05$.

	G1	G2	G3	G4	Total
Attend	6	14	13	7	40
Absent	48	32	47	33	160
Total	54	46	60	40	200

Chi-square Tests

- Create a dataframe for pivot table

```
In [23]: data2 = pd.DataFrame({"grade": ["G1", "G1", "G2", "G2", "G3","G3", "G4", "G4"],
                              "status": ["Attend", "Absent", "Attend", "Absent","Attend", "Absent",
                                        "Attend", "Absent"],
                              "Observed": [6, 48, 14, 32, 13, 47, 7, 33]})

         data2.head()
Out[23]:
```

	grade	status	Observed
0	G1	Attend	6
1	G1	Absent	48
2	G2	Attend	14
3	G2	Absent	32
4	G3	Attend	13

Chi-square Tests

- Create a pivot table

```
In [32]: table2 = pd.pivot_table(data2, values=['Observed'], index=['status'],
                                 columns=['grade'], aggfunc=np.sum, margins=True, margins_name="Total")
         table2
Out[32]:
```

	Observed				
grade	G1	G2	G3	G4	Total
status					
Absent	48	32	47	33	160
Attend	6	14	13	7	40
Total	54	46	60	40	200

Chi-square Tests

- Example
  - Chi-square test

$$H_0: p_{G1} = p_{G2} = p_{G3} = p_{G4}$$

In [34]:
```
chi2, pval, dof, expected = chi2_contingency(table2)
print('Test statistic: ',np.round(chi2,4))
print('p-value :', np.round(pval,6))
```

```
Test statistic:  6.0575
p-value : 0.640789
```

Chi-square Tests

- Practice
  - The table was made to observe whether there is association between smoking and lung cancer. Test whether the association is valid between smoking and lung cancer at $\alpha = 0.05$.

	Smoker	Non-smoker	Total
Lung cancer	117	33	150
Healthy	30	120	150
Total	147	153	300

Fisher's Exact Tests

- Overview
  - The observed probability can be computed if (a+b),(c+d),(a+c),(b+d) are fixed in the table.

a	b	a+b
c	d	c+d
a+c	b+d	n

$$p_0 = \frac{(a+b)! \cdot (c+d)! \cdot (a+c)! \cdot (b+d)!}{n! \cdot a! \cdot b! \cdot c! \cdot d!}$$

Fisher's Exact Tests

- Overview
  - Fisher's exact test is used when the proportion of the cells whose frequencies are less than 5 is $\geq 20\%$ in the table.
  - P-value is computed by the sum of probabilities of tables whose probabilities being observed are smaller than the probability of the given table being observed when (a+b),(c+d),(a+c),(b+d) are fixed in the table.
  - We can reject H_0 if p-value is smaller than α, and conclude that the two variables are associated.

Chi-square Tests

from scipy.stats import fisher_exact

fisher_exact (observed, alternative)

-. *observed: the 2x2 contingency table without margin*
-. *alternative = ‘two-sided’, ‘less’ or ‘greater’*

한양대학교 HANYANG UNIVERSITY

Fisher’s Exact Tests

- Example
  - Compute the p-value and observed probability.

	A	B	Total
G1	1	8	9
G2	4	5	9
Total	5	13	18

한양대학교 HANYANG UNIVERSITY

Fisher's Exact Tests

- Example
  - Pivot table

In [83]:
```
data3 = pd.DataFrame({"ab": ["A", "A", "B", "B"],
                      "g12": ["G1", "G2", "G1", "G2"],
                      "Observed": [1, 4, 8, 5,]})

table3 = pd.pivot_table(data3, values=['Observed'], index=['g12'],
                        columns=['ab'])
table3
```

Out[83]:

	Observed	
ab	A	B
g12		
G1	1	8
G2	4	5

Fisher's Exact Tests

- Example
  - Observed probability

In [87]:
```
import math

def observed_prob(table):
    n, p = table.shape
    out1 = 1
    out2 = 1
    tot_n = 0

    for i in range(n):
        tot_n += np.sum(table.iloc[i,:])
        out1 *= math.factorial(np.sum(table.iloc[i,:]))
        for j in range(p):
            out2 *= math.factorial(table.iloc[i,j])

    out2 *= math.factorial(tot_n)
    for j in range(p):
        out1 *= math.factorial(np.sum(table.iloc[:,j]))

    result = out1/out2
    return result
```

In [88]:
```
print("observed probability is ",np.round(observed_prob(table3),4))
```

```
observed probability is  0.1324
```

Fisher's Exact Tests

- Example
  - Compute p-value

```
In [90]: from scipy.stats import fisher_exact

         _, pval = fisher_exact(table3, alternative='two-sided')
         print('p-value is ', round(pval,4))

         p-value is  0.2941
```