Kaggle Notebook

Table of Contents

1. Name, Version Info
2. Import Libraries, Set Global Variables — JENNIFER DEY, JENNIFER DEY, PRIYA JI, KAUSHIK, PRIYA JI, KAUSHIK, PRIYA JI, JENNIFER DEY, JENNIFER DEY, JENNIFER DEY

```
from scipy.stats import chi2_contingency
```

1. H0 (Null Hypothesis) : The 2 variables is not compared are independent.
2. H1 (Alternate Hypothesis) : The 2 variables are dependent.

If the p value attained after conducting the test is less than 0.05 we reject the Null Hypothesis and accept the Alternate Hypothesis and if the p value is greater than 0.05 we accept the Null Hypothesis and reject the Alternate Hypothesis.

degree of freedom for the chi square distribution is calculated based on the size of the contingency table as

degree of freedom (row - 1) * (col - 1)

```
# Import the required libraries
from scipy.stats import chi2_contingency
from scipy.stats import chi2
from numpy.random import seed
from numpy.random import randn
from numpy.random import normal
from numpy import array, vstack, hstack, exp, log, sqrt, mean, cov, dot, diagonal
```

```
df[['Gender','Married']].sample(5)
```

	Gender	Married
1	Male	Yes
2	Male	Yes
3	Male	No
4	Male	Yes
5	Male	Yes

```
# Frequency table
table = df[['Gender','Married']].groupby(['Gender','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender
Female   75   37
Male     79  398

dof=1
expected_freq=[[ 69  42]
 [ 85 391]]

probability=0.950, critical=3.841, stat=3.564

Independent (fail to reject H0)
significance=0.050, p=0.059

Independent (fail to reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Married']].groupby(['Gender','Education','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education
Female 0    1    1
       1   74   36
Male   0    2   12
       1   77  386

dof=3
expected_freq=[[  0   1]
 [ 69  42]
 [  2  11]
 [ 85 391]]

probability=0.950, critical=7.815, stat=8.129

Dependent (reject H0)
significance=0.050, p=0.044

Dependent (reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Self_Employed','Married']].groupby(['Gender','Education','Self_Employed','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education Self_Employed
Female 0    1    1    0
       1   74   36    0
Male   0    2   12    0
       1   77  386    0

dof=3
expected_freq=[[  0   1   0]
 [ 69  42   0]
 [  2  11   0]
 [ 85 391   0]]

probability=0.950, critical=7.815, stat=8.129

Dependent (reject H0)
significance=0.050, p=0.044

Dependent (reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Self_Employed','Property_Area','Married']].groupby(['Gender','Education','Self_Employed','Property_Area','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education Self_Employed Property_Area
Female 0    1    1    0        Rural        1    0    1
       1    1    0    1        Semiurban    0    1    0
       1   72   36    0        Urban        1    0    0
Male   0    2   12    0        Rural        2   10    0
       1   77  386    0        Semiurban   75  374    0
       1    0    0    1        Urban        0    7    0

dof=5
expected_freq=[[  0   1   0]
 [  1   0   1]
 [ 69  42   0]
 [  2  11   0]
 [ 85 391   0]
 [  0   0   1]]

probability=0.950, critical=11.070, stat=8.129

Dependent (reject H0)
significance=0.050, p=0.044

Dependent (reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Self_Employed','Property_Area','Loan_Status','Married']].groupby(['Gender','Education','Self_Employed','Property_Area','Loan_Status','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education Self_Employed Property_Area Loan_Status
Female 0    1    1    0        Rural        1    0    1        N
       1    1    0    1        Semiurban    0    1    0        Y
       1   72   36    0        Urban        1    0    0        Y
Male   0    2   12    0        Rural        2   10    0        N
       1   77  386    0        Semiurban   75  374    0        Y
       1    0    0    1        Urban        0    7    0        Y

dof=5
expected_freq=[[  0   1   0]
 [  1   0   1]
 [ 69  42   0]
 [  2  11   0]
 [ 85 391   0]
 [  0   0   1]]

probability=0.950, critical=11.070, stat=8.129

Dependent (reject H0)
significance=0.050, p=0.044

Dependent (reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Married']].groupby(['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education Self_Employed Property_Area Loan_Status Dependents
Female 0    1    1    0        Rural        1    0    1        N         0
       1    1    0    1        Semiurban    0    1    0        Y         1
       1   72   36    0        Urban        1    0    0        Y         0
Male   0    2   12    0        Rural        2   10    0        N         1
       1   77  386    0        Semiurban   75  374    0        Y         0
       1    0    0    1        Urban        0    7    0        Y         1

dof=5
expected_freq=[[  0   1   0]
 [  1   0   1]
 [ 69  42   0]
 [  2  11   0]
 [ 85 391   0]
 [  0   0   1]]

probability=0.950, critical=11.070, stat=8.129

Dependent (reject H0)
significance=0.050, p=0.044

Dependent (reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Loan_Amount_Term','Married']].groupby(['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Loan_Amount_Term','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education Self_Employed Property_Area Loan_Status Dependents Loan_Amount_Term
Female 0    1    1    0        Rural        1    0    1        N         0         1
       1    1    0    1        Semiurban    0    1    0        Y         1         0
       1   72   36    0        Urban        1    0    0        Y         0         0
Male   0    2   12    0        Rural        2   10    0        N         1         0
       1   77  386    0        Semiurban   75  374    0        Y         0         0
       1    0    0    1        Urban        0    7    0        Y         1         0

dof=5
expected_freq=[[  0   1   0]
 [  1   0   1]
 [ 69  42   0]
 [  2  11   0]
 [ 85 391   0]
 [  0   0   1]]

probability=0.950, critical=11.070, stat=8.129

Dependent (reject H0)
significance=0.050, p=0.044

Dependent (reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Loan_Amount_Term','Credit_History','Married']].groupby(['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Loan_Amount_Term','Credit_History','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education Self_Employed Property_Area Loan_Status Dependents Loan_Amount_Term Credit_History
Female 0    1    1    0        Rural        1    0    1        N         0         1            1
       1    1    0    1        Semiurban    0    1    0        Y         1         0            0
       1   72   36    0        Urban        1    0    0        Y         0         0            1
Male   0    2   12    0        Rural        2   10    0        N         1         0            1
       1   77  386    0        Semiurban   75  374    0        Y         0         0            1
       1    0    0    1        Urban        0    7    0        Y         1         0            1

dof=5
expected_freq=[[  0   1   0]
 [  1   0   1]
 [ 69  42   0]
 [  2  11   0]
 [ 85 391   0]
 [  0   0   1]]

probability=0.950, critical=11.070, stat=8.129

Dependent (reject H0)
significance=0.050, p=0.044

Dependent (reject H0)
```

```
# Frequency table
table = df[['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Loan_Amount_Term','Credit_History','Loan_ID','Married']].groupby(['Gender','Education','Self_Employed','Property_Area','Loan_Status','Dependents','Loan_Amount_Term','Credit_History','Loan_ID','Married'],).size().unstack()
print('Contingency table :\n',table,'\n')
# Chi-square test of independence.
stat, p, dof, expected = chi2_contingency(table)
print('dof=%d' % dof)
print('expected_freq=%s\n' % str(expected.astype(int)))
# interpret test-statistic
prob = 0.95
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f\n' % (prob, critical, stat))
if abs(stat) >= critical:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
# interpret p-value
alpha = 1.0 - prob
print('significance=%.3f, p=%.3f\n' % (alpha, p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (fail to reject H0)')
```

```
Contingency table :
 Married  No  Yes
Gender Education Self_Employed Property_Area Loan_Status Dependents Loan_Amount_Term Credit_History Loan_ID
Female 0    1    1    0        Rural        1    0    1        N         0         1            1       1
       1    1    0    1        Semiurban    0    1    0        Y         1         0            0       1
       1   72   36    0        Urban        1    0    0        Y         0         0            1       1
Male   0    2   12    0        Rural        2   10    0        N         1         0            1       1
       1   77  386    0        Semiurban   75  374    0        Y         0         0            1       1
       1    0    0    1        Urban        0    7    0        Y         1         0            1       
```