常見的統計檢定

> 本文假設已經有統計推論及p-value計算技巧下，將不同狀況下的檢定整理如下。
>
> 1. 平均數檢定(t-test or z-test)
> 2. 比例檢定(z-test)
> 3. 母體變異數估計檢定(chi-square)
> 4. 樣本同質性檢定(chi-square)
> 5. 雙樣本母體變異數檢定(F-test)
> 6. 變量獨立性檢定(chi-square)
> 7. 適合度檢定 (chi-square)

[I 平均數檢定]

> 1. 檢定平均數是否符合假設的檢定，分為單樣本平均數檢定，獨立樣本平均數檢定，成對樣本平均數檢定。
> 2. 常用的方法為t-test,Z-test，前者是用Student's t-distribution，後者則是用Normal distribution做統計量。
>
>   比較表如下：

	z-test	t-test
樣本數	>=30	<30
機率分佈	Normal distribution	Student's t-distribution

單樣本平均數檢定

> 目標:在未知母體的平均數情況下，藉由樣本來評估假設是否為真。
>
> H_0: 假設母體的平均數為μ, $\bar{x}$=μ
>
> H_1: 假設母體的平均數不為μ, $\bar{x} \neq \mu$

Step1: 建立假設, 信心水準(1-α)

Step2: 判斷樣本數(sample size,n)是大樣本還是小樣本，決定要用t-test還是用z-test

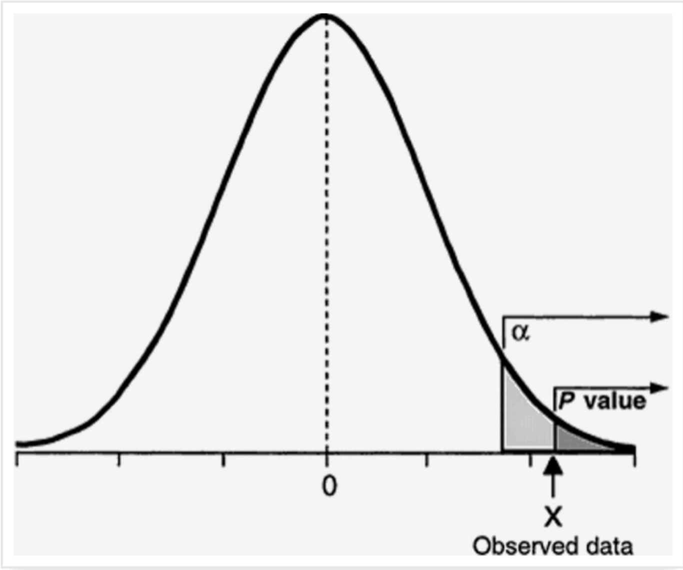
Step3: 計算:樣本平均數($\bar{x}$),樣本標準差(s), 統計量(t or z),找對應自由度下的t_{cr}(t-test)

$$t, z = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

Step4: 判斷統計量z or t 是否落在拒絕域

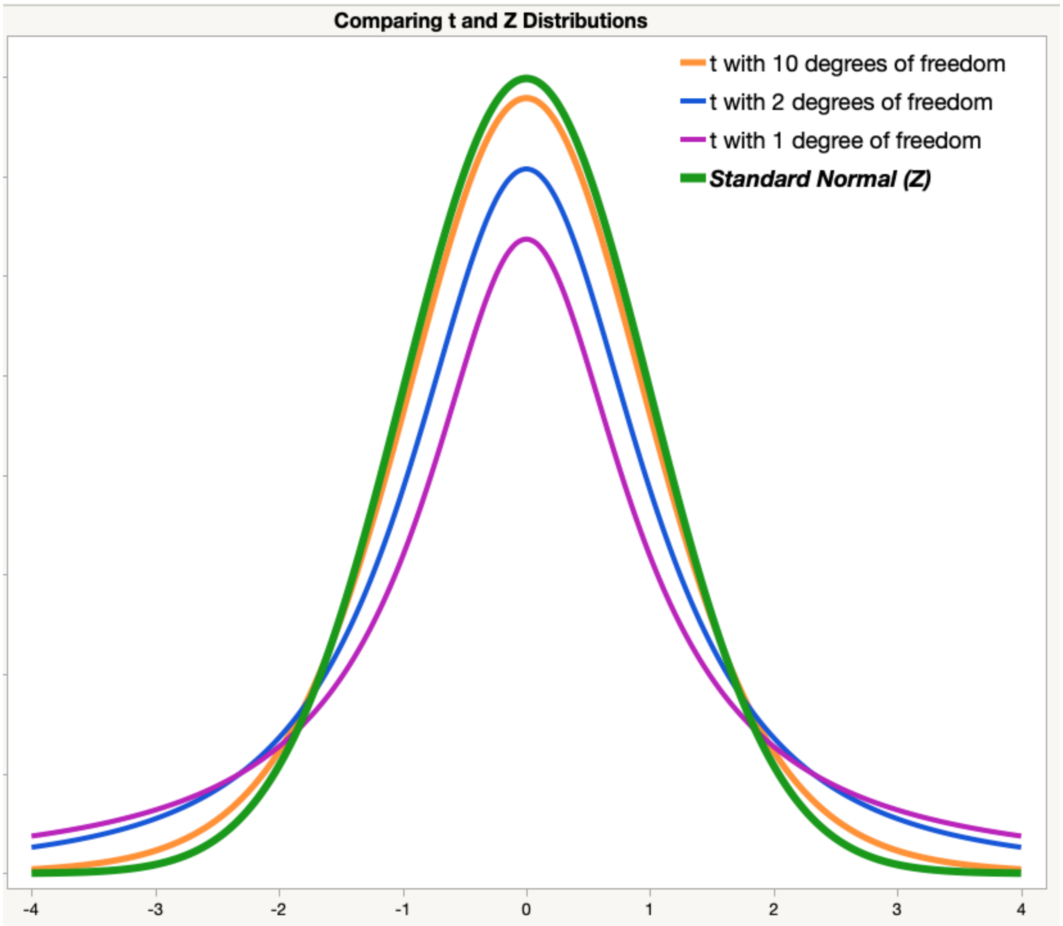
Step5: 若p-value<α，則可以拒絕H_0，否則接受H_0

若計算出來的z或是t落在拒絕域，也就是下圖的灰色區域，或是p-value小於信心水準(α)，要拒絕H_0。



圖片來源:mropengate

`Remark:` 在樣本夠多的情況下，`t-distribution, Normal distribution`會相近，所以都是可以使用的



圖片來源:JMP

- 獨立樣本變數平均數檢定

> 目標:在兩獨立樣本間，判斷平均數是否相等的檢定方法。

> H_0: 假設兩獨立樣本的平均數相同，$\mu_1 = \mu_2$
>
> H_1: 假設兩獨立樣本的平均數不相同，$\mu_1 \neq \mu_2$

Step1: 建立假設, 信心水準(1-α)

Step2: 判斷兩群體樣本數(sample size,n_1,n_2)是大樣本還是小樣本，決定要用t-test還是用z-test

Step3: 計算:兩群體樣本平均數($\bar{x}_1$,$\bar{x}_2$),樣本標準差(s_1,s_2), 統計量(t or z),找對應**自由度**及**信心水準**下的t_{cr}(t-test)

$$t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

$$df = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{\frac{(s_1^2/n_1)^2}{n_1-1} + \frac{(s_2^2/n_2)^2}{n_2-1}}$$

$$t_{criteria} = f^{-1}(\alpha, df)$$

Step4: 判斷統計量z or t 是否落在拒絕域

Step5: 若p-value<α，則可以拒絕H_0，否則接受H_0

- 成對樣本平均數檢定

> 目標:在兩成對樣本間，判斷平均數是否相等的檢定方法。

> H_0: 假設成對樣本的平均數相同，$\mu_1 = \mu_2$
>
> H_1: 假設成對樣本的平均數不相同，$\mu_1 \neq \mu_2$

Step1: 建立假設, 信心水準(1-α)

Step2: 判斷樣本數(sample size,n)是大樣本還是小樣本，決定要用t-test還是用z-test

Step3: 計算:成對樣本差值平均數($\bar{D}$),差值的標準差(S_D),統計量(t or z),找對應**自由度**及**信心水準**下的 t_{cr}(t-test)

$$\bar{D} = \frac{\Sigma(x_{1i} - x_{2i})}{n}, i = 1 \sim n$$

$$S_D = S.D(D_i)$$

$$t = \frac{\bar{D}}{\frac{S_D}{\sqrt{n}}}$$

$$df = n - 1$$

$$t_{cr} = f^{-1}(\alpha, df)$$

Step4: 判斷統計量z or t 是否落在拒絕域

Step5: 若p-value<α，則可以拒絕H_0，否則接受H_0

In [22]:

```
# code example I
#ref url: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats

# one sample t-test
import numpy as np
from scipy import stats

np.random.seed(123)
alpha=0.05
n=25
#產生隨機樣本
rvs = stats.norm.rvs(loc=5, scale=5, size=(n))
#計算統計量t及p-value
t,p_value=stats.ttest_1samp(rvs,5,alternative="two-sided")
##計算臨界的t值
t_cr_l=stats.t.ppf(alpha/2,df=n-1)
t_cr_u=stats.t.ppf(1-alpha/2,df=n-1)
print("雙尾t檢定")
print("p-value=",p_value)
print("t=",t)
print("Critical t=",[t_cr_l,t_cr_u])


t,p_value=stats.ttest_1samp(rvs,5,alternative="less")
t_cr_l=stats.t.ppf(alpha,df=n-1)
print()
print("左尾t檢定")
print("p-value=",p_value)
print("t=",t)
print("Critical t=",[t_cr_l])

t,p_value=stats.ttest_1samp(rvs,5,alternative="greater")
t_cr_u=stats.t.ppf(1-alpha,df=n-1)
print()
print("右尾t檢定")
print("p-value=",p_value)
print("t=",t)
print("Critical t=",[t_cr_u])
```

```
雙尾t檢定
p-value= 0.5748193742925578
t= 0.5687359228814379
Critical t= [-2.063898561628021, 2.0638985616280205]

左尾t檢定
p-value= 0.7125903128537212
t= 0.5687359228814379
Critical t= [-1.7108820799094282]

右尾t檢定
p-value= 0.2874096871462789
t= 0.5687359228814379
Critical t= [1.7108820799094275]
```

In [23]:

```
# Compare two independant sample mean

#ref url:https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.

import numpy as np
```

```python
from scipy import stats
np.random.seed(123)
print("獨立雙樣本t檢定")
print()
## Assume equal var.
alpha=0.05
n1=20
n2=25
rvs1=stats.norm.rvs(loc=0,scale=1,size=n1)
rvs2=stats.norm.rvs(loc=0.8,scale=1,size=n2)

t,p_value=stats.ttest_ind(rvs1,rvs2,equal_var=True,alternative="two-sided"

t_cr_l=stats.t.ppf(alpha/2,df=n1+n2-2)
t_cr_u=stats.t.ppf(1-alpha/2,df=n1+n2-2)
print("Assume equal var.")
print("雙尾t檢定")
print("p-value=",p_value)
print("t=",t)
print("Critical t=",[t_cr_l,t_cr_u])
print("p-value>0.05，接受H0")

## Assume difffernet var.
np.random.seed(123)
##計算welch's test的自由度
def welch_dof(x1,x2):
    var1=np.var(x1,ddof=1)
    var2=np.var(x2,ddof=1)
    N1=x1.size
    N2=x2.size
    v1=N1-1
    v2=N2-1
    dof=((var1/N1+var2/N2)**2)/((var1**2/N1**2/v1)+(var2**2/N2**2/v2))
    return dof

alpha=0.05
n1=20
n2=25
rvs1=stats.norm.rvs(loc=0,scale=1,size=n1)
rvs2=stats.norm.rvs(loc=0.8,scale=1.2,size=n2)


t,p_value=stats.ttest_ind(rvs1,rvs2,equal_var=False,alternative="two-sided'
df=welch_dof(rvs1,rvs2)

t_cr_l=stats.t.ppf(alpha/2,df=df)
t_cr_u=stats.t.ppf(1-alpha/2,df=df)
print()
print("Assume difffernet var.")
print("雙尾t檢定")
print("p-value=",p_value)
print("t=",t)
print("Critical t=",[t_cr_l,t_cr_u])
print("p-value>0.05，接受H0")
```

```
獨立雙樣本t檢定

Assume equal var.
雙尾t檢定
p-value= 0.2742916206016663
t= -1.1073667693451013
Critical t= [-2.0166921941428138, 2.0166921941428133]
p-value>0.05，接受H0

Assume difffernet var.
雙尾t檢定
p-value= 0.4022272707184602
t= -0.8464650449898035
Critical t= [-2.0197949733181733, 2.019794973318173]
p-value>0.05，接受H0
```

In [24]:

```python
# Compare two pair sample mean
#ref url:https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.

import numpy as np
from scipy import stats
np.random.seed(123)
n=25
alpha=0.05
rvs1=stats.norm.rvs(loc=0,scale=1,size=n)
rvs2=stats.norm.rvs(loc=1.5,scale=1.2,size=n)

t,p_value=stats.ttest_rel(rvs1,rvs2,alternative="two-sided")
t_cr_l=stats.t.ppf(alpha/2,df=n-1)
t_cr_u=stats.t.ppf(1-alpha/2,df=n-1)
print("成對樣本雙尾t檢定")
print("p-value=",p_value)
print("t=",t)
print("Critical t=",[t_cr_l,t_cr_u])
print("p-value<0.05，拒絕H0")
```

```
成對樣本雙尾t檢定
p-value= 0.0012643661104337048
t= -3.6515518795533652
Critical t= [-2.063898561628021, 2.0638985616280205]
p-value<0.05，拒絕H0
```

II 比例檢定

> 目標：用來判別兩個獨立樣本比例是否有顯著差異的檢定方法。
>
> H_0: 假設兩獨立樣本的p相同，$p_1 = p_2$。
>
> H_1: 假設兩獨立樣本的p不相同，$p_1 \neq p_2$。 分配: t- or Normal distribution

Step1: 建立假設，決定單尾還是雙尾(看H_1)

Step2: 計算兩獨立樣本的proportion, $\hat{p_1} = \frac{x_1}{n_1}$, $\hat{p_2} = \frac{x_2}{n_2}$, 其中$x_1$、$x_2$是成功次數(positive 的次數)。

Step3: 設定信心水準(1-α),由機率分佈模型決定拒絕域,t_{cr} or z_{cr}

Step4: 計算

$$p_c = \frac{(x_1 + x_2)}{(n_1 + n_2)} = \frac{n_1\hat{p_1} + n_2\hat{p_2}}{n_1 + n_2}$$

,

$$S_{\hat{p_1}-\hat{p_2}} = \sqrt{\frac{p_c(1-p_c)}{n_1} + \frac{p_c(1-p_c)}{n_2}}$$

Stet5: 計算統計量t or z = $\frac{(\hat{p_1}-\hat{p_2})-(p_1-p_2)}{S_{\hat{p_1}-\hat{p_2}}}$

Step6: 計算p-value，看p-value是否小於α

Example:

假設籃球命中率男生樣本為$n_1 = 50$,命中率數目$x_1 = 32$,女生樣本為40,命中率數為$x_2 = 24$，想知道男生命中率是否多於10%。

step1:

H_0:$(p_1 - p_2) < 0.1$

H_1:$(p_1 - p_2) \geq 0.1$

信心水準95%,$\alpha = 0.05$,因為是H_1是$(p_1 - p_2)$大於某值，所以是右尾檢定。

step3:

$z_{cr} = N^{-1}_{standard}(0.95) = 1.645$

step4:

$$p_c = \frac{32+24}{50+40} = \frac{56}{90} = 0.622$$

$$\hat{p}_1 = 32/50 = 0.64,\ \hat{p}_2 = 24/40 = 0.6$$

$$S_{\hat{p_1}-\hat{p_2}} = \sqrt{\frac{0.622(1-0.622)}{50} + \frac{0.622(1-0.622)}{40}}) = 0.102$$

step5:

$$z = \frac{(\hat{p_1}-\hat{p_2})-(p_1-p_2)}{S_{\hat{p_1}-\hat{p_2}}} = \frac{0.04-0.1}{0.102} = -0.588$$

step6:

$p - value = 1 - N^{-1}(z = -0.588, mean = 0, std = 1, cumulative) = 1 - 0.278 =$

推論：因為0.722大於0.05，接受H_0

In [25]:

```
# code example II

#ref url:https://www.statsmodels.org/stable/generated/statsmodels.stats.pro

#ref url:https://online.stat.psu.edu/stat800/lesson/5/5.5

import statsmodels.stats.proportion as proportion
probs=np.array([0.65,0.6])
nobs=np.array([50,40])
success=(probs*nobs).astype(int)

z,p_value=proportion.proportions_ztest(count=success,nobs=nobs,value=0.1,al
print()
print("Z=",z)
print()
print("p-value=",p_value)
print()
print("=> p-value > 0.05, accept H0")
```

```
Z= -0.5833833511969476

p-value= 0.7201823685074527

=> p-value > 0.05, accept H0
```

III 單樣本母體變異數估計檢定

> 目標：用來測定母體變異數是否為某個值的檢定
>
> H_0:$\sigma^2 = \sigma_0^2$
>
> H_1:$\sigma^2 > \sigma_0^2$
>
> H_1:$\sigma^2 < \sigma_0^2$
>
> H_1:$\sigma^2 \neq \sigma_0^2$
>
> 模型: chi-square distribution(卡方分配)

step1: 建立假說，設定信心水準，決定單尾還是雙尾

step2: 決定拒絕H_0的critical χ_{cr}^2

- H_1:$\sigma^2 > \sigma_0^2$

$$\chi^2 > \chi^2_{(alpha,\nu)}$$

- H_1:$\sigma^2 < \sigma_0^2$

$$\chi^2 < \chi^2_{(1-alpha,\nu)}$$

- H_1:$\sigma^2 \neq \sigma_0^2$

$$\chi^2 < \chi^2_{(1-\frac{alpha}{2},\nu)}$$

$$or$$

$$\chi^2 > \chi^2_{(\frac{alpha}{2},\nu)}$$

step3: 計算樣本標準差S及 χ^2

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2}$$

step4: 計算p-value，並判斷是否有小於α

In [26]:

```
# code example III
import scipy.stats as stats
import numpy as np
np.random.seed(123)
std=1
n=200
rvs=stats.norm.rvs(loc=2,scale=std,size=n)
rvs_var=np.var(rvs,ddof=1)
chi2=(n-1)*rvs_var/std**2

print("chi2=",chi2)
p_value=1-stats.chi2.cdf(chi2,df=n-1)
print("p_value=",p_value)

chi2_cr_l=stats.chi2.ppf(0.05/2,n-1)
print("chi2_cr_l=",chi2_cr_l)
chi2_cr_u=stats.chi2.ppf(1-0.05/2,n-1)
print("chi2_cr_u=",chi2_cr_u)
print("=> p-vlaue>0.05且chi2落在接受域，故接受H0")
```

```
chi2= 221.5018384804587
p_value= 0.13111743854343627
chi2_cr_l= 161.82618239364686
chi2_cr_u= 239.9596818276442
=> p-vlaue>0.05且chi2落在接受域，故接受H0
```

IV 多樣本同質性檢定

> 目標：用來判別不同組別的資料是否來自同一個母群體。
>
> H_0:各組別來自相同群體
>
> H_1:各組別來自不同群體
>
> 模型: chi-square distribution(卡方分配)

step1: 建立假設，信心水準(1-α)

step2: 計算個欄位期望值

step3: 計算統計量$\chi^2 = \frac{(Observed\ value - Expected\ value)^2}{expected\ value}$

step4: 計算自由度，令r等於組別數，c為各組別的觀察值。

$$df = (r-1)(c-1)$$

step5: 計算p-value，若p-value<α，則拒絕H_0

Example

假設有如下數據，g_i代表組別，c_i代表各組的觀察值，在信心水準95%的情況下，各組是否同質？

	Observed value		
	c1=Yes	c2=No	Total
g1	103	8440	8543
g2	52	4289	4341
g3	65	6428	6493
Total	220	19157	19377

資料來源:umenlearning

step2:

各欄位的期望值=各組別的總數*(各觀察值類別的總數/總樣本數)

> 以(g1,c1)這個欄位為例，期望值等於$8543 \times \frac{220}{19377} = 96.994$

step3: 計算χ^2

$$\chi^2 = \frac{(103-96.994)^2}{96.994} + \frac{(8440-8446)^2}{8446} + \frac{(52-49.286)^2}{49.286} + \frac{(4289-4291.7)^2}{4291.7} + \frac{(65-73.719)^2}{73.719} + \frac{(6428-6419}{6419.3}$$

step4: 計算自由度

$$df = (3-1) \times (2-1) = 2$$

step5: 計算p-value

$$P(\chi^2 < 1.57, 2, cumulative) = 0.544$$

$$p - value = 1 - 0.544 = 0.456$$

推論：因為p-value>0.05，故接受H_0，各組別是同質的。

In [27]:

```python
# code example IV

#ref url:https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.c

import scipy.stats

table=[[103,8440],[52,4289],[65,6428]]
chi2,p,dof,expected=stats.chi2_contingency(table)
print()
print("chi2=",chi2)
print()
print("p=",p)
print()
print("dof=",dof)
print()
print("expected=")
print(expected)
print()
print("=> p-value > 0.05: accept H0")
```

```
chi2= 1.57040735910991

p= 0.4560268089882763

dof= 2

expected=
[[  96.99437477 8446.00562523]
 [  49.28626722 4291.71373278]
 [  73.719358   6419.280642  ]]

=> p-value > 0.05: accept H0
```

V 雙樣本母體變異數檢定

> 目標：檢定兩樣本的變異數(variance)是否相同
>
> H_0: 假設兩樣本母體變異數相同，$\sigma_1^2 = \sigma_2^2$
>
> H_1:$\sigma_1^2 > \sigma_2^2$
>
> H_1:$\sigma_1^2 < \sigma_2^2$
>
> H_1:$\sigma_1^2 \neq \sigma_2^2$

step1: 建立假設，信心水準

step2: 計算兩樣本的標準差(S_1,S_2)

step3: 計算F數

$$F = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} = \frac{S_1^2}{S_2^2}$$

step4: 以自由度及信心水準建立拒絕的臨界值

- H_1:$\sigma_1^2 > \sigma_2^2$

$$F_{cr} = F_{(\alpha,\nu_1=n_1-1,\nu_2=n_2-1)}$$

- H_1:$\sigma_1^2 < \sigma_2^2$

$$F_{cr} = F_{(1-\alpha,\nu_1=n_1-1,\nu_2=n_2-1)}$$

- H_1:$\sigma_1^2 \neq \sigma_2^2$

$$F_{cr} = F_{(1-\alpha/2,\nu_1=n_1-1,\nu_2=n_2-1)}$$
$$or$$
$$F_{cr} = F_{(\alpha/2,\nu_1=n_1-1,\nu_2=n_2-1)}$$

step4: 計算p-value,若$p_value < \alpha$，則拒絕H_0

In [28]:

```python
# code example V

from scipy.stats import f_oneway
import scipy.stats
n1=60
n2=75
rvs1=stats.norm.rvs(loc=0,scale=1,size=n1)
rvs2=stats.norm.rvs(loc=1,scale=3,size=n2)
F,p=f_oneway(rvs1,rvs2)
print("F=",F)
print("p-value=",p)
print("=> p-value<0.05:拒絕H0")
```

```
F= 3.092055970374195
p-value= 0.0809754877107539
=> p-value<0.05:拒絕H0
```

VI 變量獨立性檢定

> 目標:用來檢定個變量間的獨立性。
>
> 理論:藉由兩事件若獨立則P(A and B)=P(A)P(B)來計算期望值，最後用卡方檢定來判斷觀察值和期望值的匹配程度。
>
> H_0:假設變量間獨立
>
> H_1:假設變量不獨立

step1: 建立假設，信心水準(1-α)

step2: 計算個欄位期望值

step3: 計算統計量$\chi^2 = \frac{(Observed\ value - Expected\ value)^2}{expected\ value}$

step4: 計算自由度，令r等於組別數，c為各組別的觀察值。

$$df = (r-1)(c-1)$$

step5: 計算p-value，若p-value<α，則拒絕H_0

Example

假設想要知道主餐和副餐的點法有沒有相關，信心水準95%。

資料如下：

		主餐			
		醬拉	海拉	蒜拉	
附餐	豬排	6	0	1	7
	腰內肉	1	2	4	7
	飯糰	1	3	2	6
		8	5	7	20

step2: 計算期望值個欄位

假設獨立的情況下則P(主餐 and 附餐)=P(主餐)×P(附餐)

以主餐＝醬拉，附餐＝豬排為例

P(主餐＝醬拉)=8/20, P(附餐=豬排)=7/20

則P(醬拉 and 豬排)=P(醬拉)×P(豬排)=0.14

各欄位的理論機率如下圖

		主餐			
		醬拉	海拉	蒜拉	
附餐	豬排	0.14	0.0875	0.1225	0.35
	腰內肉	0.14	0.0875	0.1225	0.35
	飯糰	0.12	0.075	0.105	0.3
		0.4	0.25	0.35	1

對應出現次數如下圖

		主餐			
		醬拉	海拉	蒜拉	
附餐	豬排	2.8	1.75	2.45	7
	腰內肉	2.8	1.75	2.45	7
	飯糰	2.4	1.5	2.1	6
		8	5	7	20

step3: 計算卡方值

$$\chi^2 = \Sigma_{i=1}^{i=3}\Sigma_{i=1}^{j=3}\frac{(O_{ij} - E_{ij})^2}{E_{ij}} = 10.76$$

step4: 計算自由度，令r等於附餐種類數，c為主餐種類數目。

$$df = (r-1)(c-1) = 4$$

step5:

臨界的卡方值

$$\chi^2_{(p=0.95,df=4)} = 9.487$$

計算p-value:

$$P(\chi^2 < 10.76, \nu = 4, cumulative) = 0.97$$

$$p - value = 1 - 0.971 = 0.029$$

推論：因為p-value=0.029<0.05，故拒絕H_0，主餐和附餐並非獨立。

In [29]:

```
# code example VI

#ref url:https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.c

import scipy.stats

table=[[6,0,1],[1,2,4],[1,3,2]]
chi2,p,dof,expected=stats.chi2_contingency(table)
print()
print("chi2=",chi2)
print()
print("p=",p)
print()
print("dof=",dof)
print()
print("expected=")
print(expected)
print()
print("=> p-value < 0.05: reject H0")
```

```
chi2= 10.760204081632654

p= 0.029395374418220755

dof= 4

expected=
[[2.8  1.75 2.45]
 [2.8  1.75 2.45]
 [2.4  1.5  2.1 ]]

=> p-value < 0.05: reject H0
```

[VII 適合度檢定]

> 目標：用來檢定某樣本是否符合預期的分佈模型。
>
> 理論：若是樣本符合模型，那觀察值和模型產出的理論值(期望值)應該要很接近，故用卡方檢定。
>
> H_0:樣本符合模型
>
> H_1:樣本不符合模型

step1: 建立假設，信心水準(1-α)

step2: 計算期望值

step3: 計算統計量$\chi^2 = \frac{(Observed\ value - Expected\ value)^2}{expected\ value}$

step4: 計算自由度。

$$df = n - 1$$

step5: 計算p-value，若p-value<α，則拒絕H_0

Example

想知道某骰子各面出現的機率是否符合均勻分布，假設信心水準95%，數據如下：

Event	Observed value	Expected value
1	95	100
2	105	100
3	100	100
4	102	100
5	96	100
6	102	100
Total	600	600

step3: 計算統計量

$$\chi^2 = \frac{(Observed\ value - Expected\ value)^2}{Expected\ value} = 0.74$$

step4: 計算自由度。

$$df = n - 1 = 6 - 1 = 5$$

step5: 計算p-value，拒絕的臨界值

$$\chi^2_{(p=0.95,\nu=5)} = 11.07$$

$$P(\chi^2 \leq 0.74) = 0.019$$

$$p - value = 1 - 0.019 = 0.981$$

推論：因為p-value=0.981>$\alpha = 0.05$，故接受H_0，該骰子的分佈為均勻分布。

In [30]:

```
# code example
# ref url: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stat
import scipy.stats
from scipy.stats import power_divergence
import numpy as np
obs=np.array([95,105,100,102,96,102])
exp=np.array([100]*6)
statistic,p=power_divergence(obs,exp)
print()
print('statistic(chi2)=',statistic)
print()
print('p-value=',p)
print("=> p-value > 0.05, accept H0")
```

```
statistic(chi2)= 0.7400000000000001

p-value= 0.980701472519648
=> p-value > 0.05, accept H0
```