Kruskal Wallis test

September 20, 2020

BS. Lê Ngọc Khả Nhi

1 Giới thiệu

Kiểm định Kruskal-Wallis được thiết lập vào năm 1952 bởi 2 nhà thống kê người Mỹ William Kruskal (1919-2005) và Wilson Wallis (1912-1998), như một giải pháp phi tham số thay thế cho phân tích phương sai đơn biến (One-way ANOVA).

Quy trình của kiểm định H theo Kruskal Wallis gồm 4 bước:

1) Hoán chuyển dữ liệu bằng cách xếp thứ hạng
2) Tính giá trị của trị số thống kê H, và giả định H theo quy luật phân phối Chi bình phương với độ tự do = (k-1) với k là số phân nhóm
3) Kiểm định giả thuyết vô hiệu: H0: Phân phối thứ hạng là như nhau trong mỗi phân nhóm (đối thuyết H1: Có sự khác biệt về đặc tính phân phối (của thứ hạng) giữa các phân nhóm). Nếu các phân nhóm có phân phối đồng dạng, ta có thể so sánh trung vị với giả thuyết H1: Có sự khác biệt về trung vị giữa các phân nhóm.
4) Hậu kiểm (posthoc test) bằng một trong 2 phương pháp: Dunn (1964) hoặc Conover (1979), để so sánh bắt cặp tuần tự thứ hạng giữa k phân nhóm.

[1]:
```
warning_status = "ignore"
import warnings
warnings.filterwarnings(warning_status)
with warnings.catch_warnings():
    warnings.filterwarnings(warning_status, category=DeprecationWarning)

import numpy as np
import pandas as pd

from scipy.stats import *
from itertools import combinations

from statsmodels.sandbox.stats.multicomp import multipletests
```

2 Dữ liệu minh họa

Đây là 1 nghiên cứu thực nghiệm khảo sát 8 biến định lượng 'BMI', 'Age','Volume','Surface','DmCO','Thickness','DLCO', 'FVC' giữa 3 phân nhóm : E = Khí phế thũng, F = Xơ phổi, N = Bình thường.

[2]:
```
data = pd.read_csv("https://raw.githubusercontent.com/kinokoberuji/R-Tutorials/
↪master/aerodim.csv",sep=";")

data.head()

data['Class'] = data['Diagnostic'].map({'E':'Khí phế thũng',
                                        'F': 'Xơ phổi',
                                         'N': 'Bình thường'
                                        })

data.head()
```

[2]:

	Poids	Taille	BMI	Age	Sexe	Hb	Diagnostic	GST	Volume	\
0	53	165	19.467401	54	F	13.4	E	1.284048	7.0080	
1	92	170	31.833910	75	H	17.1	E	1.291861	3.7750	
2	69	186	19.944502	41	H	14.6	E	1.211214	7.7695	
3	60	160	23.437500	75	F	13.5	E	1.439790	4.3295	
4	72	172	24.337480	60	H	14.6	E	1.388885	5.5610	

	Surface	DmCO	Thickness	DLCO	FVC	Class
0	19.647869	450.315872	0.142701	6.767	2.30	Khí phế thũng
1	10.519710	72.271882	0.476060	19.146	3.04	Khí phế thũng
2	23.092705	372.142895	0.202952	29.047	5.72	Khí phế thũng
3	10.825326	89.206177	0.396893	14.943	2.57	Khí phế thũng
4	14.414152	180.833030	0.260699	13.888	4.36	Khí phế thũng

3 Xây dựng class nonparametric_ANOVA

Nhi thiết kế một class nonparametric_ANOVA với công dụng như sau:

1) Khởi tạo 1 object với arguments gồm dataframe dữ liệu (data), tên cột biến phụ thuộc (dep) và tên cột biến phân nhóm (grp). Hàm init sẽ tính toán một số thông số kỹ thuật cần thiết và lưu lại trong thuộc tính _param

Các thông số này gồm tên dep, grp, dataframe có thêm 2 cột là rank (thứ hạng) và rdif (khác biệt với thứ hạng turng bình), cỡ mẫu n, số phân nhóm k, tên các phân nhóm (lev), cỡ mẫu mỗi phân nhóm (grp_size), thứ hạng trung bình mỗi phân nhóm (avg_rank) và trung bình khác biệt mỗi phân nhóm (so với thứ hạng trung bình): grp_dif

2) Method kruskal_wallis tính toán giá trị H theo phương pháp chính tắc (không xấp xỉ), sau đó hiệu chỉnh H nếu có những cặp thứ hạng trùng nhau; sau đó dùng kiểm định Chi bình phương để kiểm tra giả thuyết H0.

3) Method dunn_post_hoc áp dụng post-hoc test Dunn, với hiệu chỉnh Bonferroni

4) Method conover_post_hoc áp dụng post-hoc test Conover, với hiệu chỉnh Bonferroni

[11]:
```
class nonparametric_ANOVA():

    def __init__(self, data = None, dep = None, grp = None):

        data = data[[dep, grp]].dropna(how = 'any', axis = 0)
        data = data.reset_index(drop=True)
        n = data.shape[0]
        lev = data[grp].unique()
        k = len(lev)
        r = (n + 1)/2
        data['rank'] = rankdata(data[dep])
        data['rdif'] = (data['rank'] - r)**2
        grp_rnk = data.groupby(grp, observed=True)['rank']
        grp_dif = data.groupby(grp, observed=True)['rdif']
        grp_size = grp_rnk.count()
        avg_rank = grp_rnk.mean()

        self._param = {'data': data,
                       'dep': dep,
                       'grp': grp,
                       'n': n,
                       'k': k,
                       'lev':lev,
                       'avg_rank': avg_rank,
                       'grp_size': grp_size,
                       'grp_dif': grp_dif,
                      }

    def kruskal_wallis(self):

        n = self._param['n']
        grp_size = self._param['grp_size'].ravel()
        avg_rank = self._param['avg_rank'].ravel()
        r = (n+1)/2
        sum_rdif = self._param['grp_dif'].sum().ravel()
        k = self._param['k']
        H = (n-1)* (sum(grp_size * (avg_rank - r)**2)/sum(sum_rdif))
        H /= tiecorrect(self._param['data']['rank'].ravel())
        p = chi2.sf(H, k-1)

        kwt = pd.DataFrame({'Phân nhóm': self._param['grp'],
                    'Biến số': self._param['dep'],
                    'H': H,
                    'Độ tự do': k-1,
```

```
                'Giá trị p': p,
                'Phủ định H0': 'Có thể' if p< 0.05 else 'Không thể',
               }, index = [''])

    return kwt

def dunn_post_hoc(self, p_adjust = 'bonferroni'):

    avg_rank = self._param['avg_rank']
    n = self._param['n']
    grp_size = self._param['grp_size']
    lev = self._param['lev']
    k = self._param['k']
    data = self._param['data']

    vals = data.groupby('rank').count()[self._param['dep']].values
    tie_sum = np.sum(vals[vals != 1] ** 3 - vals[vals != 1])
    tie_sum = 0 if not tie_sum else tie_sum
    x_ties = tie_sum / (12. * (n - 1))

    xmat = np.zeros((k, k))
    combs = combinations(range(k), 2)

    tri_upper = np.triu_indices(xmat.shape[0], 1)
    tri_lower = np.tril_indices(xmat.shape[0], -1)
    xmat[:, :] = 0

    def dunn(i, j):
        diff = np.abs(avg_rank.loc[i] - avg_rank.loc[j])
        A = n * (n + 1) / 12
        B = (1 / grp_size.loc[i] + 1. / grp_size.loc[j])
        z_value = diff / np.sqrt((A - x_ties) * B)
        p_value = 2. * norm.sf(np.abs(z_value))
        return p_value

    for i, j in combs:
        xmat[i, j] = dunn(lev[i], lev[j])

    if p_adjust:
        xmat[tri_upper] = multipletests(xmat[tri_upper], method=p_adjust)[1]

    xmat[tri_lower] = xmat.T[tri_lower]
    np.fill_diagonal(xmat, 1)

    dunn_df = pd.DataFrame(xmat, index=lev, columns=lev)

    print(f"So sánh bắt cặp bằng kiểm định Dunn")
```

```
        return dunn_df

    def conover_post_hoc(self, p_adjust = 'bonferroni'):

        avg_rank = self._param['avg_rank']
        n = self._param['n']
        grp_size = self._param['grp_size']
        lev = self._param['lev']
        k = self._param['k']
        data = self._param['data']
        sum_rdif = self._param['grp_dif'].sum().ravel()
        grp = data.groupby(self._param['grp'], observed=True)['rank']
        sum_grp = grp.sum().ravel()

        vals = data.groupby('rank').count()[self._param['dep']].values
        tie_sum = np.sum(vals[vals != 1] ** 3 - vals[vals != 1])
        tie_sum = 0 if not tie_sum else tie_sum
        x_ties = tie_sum / (12 * (n - 1))

        H = (12 / (n * (n + 1)) * np.sum(sum_grp**2 /grp_size.ravel())) - 3 *␣
↪(n + 1)
        H /= tiecorrect(self._param['data']['rank'].ravel())

        if x_ties == 1:
            S2 = n * (n + 1) / 12
        else:
            S2 = (1 / (n - 1)) * (np.sum(data['rank'] ** 2) - (n * (((n +␣
↪1)**2) / 4)))

        xmat = np.zeros((k, k))
        combs = combinations(range(k), 2)

        tri_upper = np.triu_indices(xmat.shape[0], 1)
        tri_lower = np.tril_indices(xmat.shape[0], -1)
        xmat[:, :] = 0

        def conover(i, j):
            diff = np.abs(avg_rank.loc[i] - avg_rank.loc[j])
            B = (1. / grp_size.loc[i] + 1/grp_size.loc[j])
            D = (n - 1. - H) / (n - k)
            t_value = diff / np.sqrt(S2 * B * D)
            p_value = 2. * t.sf(np.abs(t_value), df=n-k)
            return p_value

        for i, j in combs:
            xmat[i, j] = conover(lev[i], lev[j])
```

```
        if p_adjust:
            xmat[tri_upper] = multipletests(xmat[tri_upper], method=p_adjust)[1]

        xmat[tri_lower] = xmat.T[tri_lower]
        np.fill_diagonal(xmat, 1)

        con_df = pd.DataFrame(xmat, index=lev, columns=lev)

        print(f"So sánh bắt cặp bằng kiểm định Conover")

        return con_df
```

4 Sử dụng class nonparametric_ANOVA cho 1 biến

Nhi test tính năng của class nonparametric_ANOVA cho biến Thickness

[12]:
```
obj = nonparametric_ANOVA(data = data, dep = 'Thickness', grp = 'Class')

obj.kruskal_wallis()
```

[12]:

Phân nhóm	Biến số	H	Độ tự do	Giá trị p	Phủ định H0
Class	Thickness	31.130711	2	1.738009e-07	Có thể

Diễn giải kết quả trong thí dụ này như sau:

Có sự khác biệt ý nghĩa về đặc tính phân phối của Thickness giữa 3 phân nhóm (H(2) = 31.13; p < 0.0001).

Post-hoc test cho phép định vị sự khác biệt giữa các phân nhóm như sau:

[13]:
```
obj.dunn_post_hoc(p_adjust = 'bonferroni')
```

```
So sánh bắt cặp bằng kiểm định Dunn
```

[13]:

	Khí phế thũng	Xơ phổi	Bình thường
Khí phế thũng	1.000000e+00	1.381358e-07	0.001625
Xơ phổi	1.381358e-07	1.000000e+00	0.038338
Bình thường	1.625181e-03	3.833825e-02	1.000000

[14]:
```
obj.conover_post_hoc(p_adjust = 'bonferroni')
```

```
So sánh bắt cặp bằng kiểm định Conover
```

[14]:

	Khí phế thũng	Xơ phổi	Bình thường
Khí phế thũng	1.000000e+00	8.117064e-15	1.728766e-09
Xơ phổi	8.117064e-15	1.000000e+00	1.816640e-06
Bình thường	1.728766e-09	1.816640e-06	1.000000e+00

5 Áp dụng class nonparametric_ANOVA cho hàng loạt biến

Nếu muốn áp dụng quy trình trên cho hàng loat biến, ta chỉ cần viết 1 vòng lặp:

[16]:
```
for v in ['BMI', 'Age','Volume','Surface','DmCO','Thickness','DLCO', 'FVC']:

    print(f"Phân tích biến {v}\n")
    print('-'*50)

    obj = nonparametric_ANOVA(data = data,
                              dep = v,
                              grp = 'Class')

    print(obj.kruskal_wallis())
    print('-'*50)

    print(obj.dunn_post_hoc(p_adjust = 'bonferroni'))
    print('-'*50)

    print(obj.conover_post_hoc(p_adjust = 'bonferroni'))
    print('-'*50)
```

```
Phân tích biến BMI

--------------------------------------------------
 Phân nhóm Biến số          H  Độ tự do  Giá trị p Phủ định H0
     Class     BMI  15.101097         2   0.000526      Có thể
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  0.085790     0.179305
Xơ phổi             0.085790  1.000000     0.000317
Bình thường         0.179305  0.000317     1.000000
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  0.026913     0.068708
Xơ phổi             0.026913  1.000000     0.000065
Bình thường         0.068708  0.000065     1.000000
--------------------------------------------------
Phân tích biến Age

--------------------------------------------------
 Phân nhóm Biến số          H  Độ tự do  Giá trị p Phủ định H0
     Class     Age  20.825387         2    0.00003      Có thể
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng   Xơ phổi  Bình thường
```

```
Khí phế thũng       1.000000  1.000000     0.000092
Xơ phổi             1.000000  1.000000     0.001671
Bình thường         0.000092  0.001671     1.000000
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  1.000000     0.000002
Xơ phổi             1.000000  1.000000     0.000038
Bình thường         0.000002  0.000038     1.000000
--------------------------------------------------
Phân tích biến Volume

--------------------------------------------------
 Phân nhóm Biến số         H  Độ tự do  Giá trị p Phủ định H0
     Class  Volume  6.350145         2   0.041791      Có thể
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  0.312655     0.967492
Xơ phổi             0.312655  1.000000     0.035375
Bình thường         0.967492  0.035375     1.000000
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  0.273967     0.892963
Xơ phổi             0.273967  1.000000     0.032536
Bình thường         0.892963  0.032536     1.000000
--------------------------------------------------
Phân tích biến Surface

--------------------------------------------------
 Phân nhóm  Biến số          H  Độ tự do  Giá trị p Phủ định H0
     Class  Surface  23.309402         2   0.000009      Có thể
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  0.011554     0.000006
Xơ phổi             0.011554  1.000000     0.621356
Bình thường         0.000006  0.621356     1.000000
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng   Xơ phổi   Bình thường
Khí phế thũng   1.000000e+00  0.000150  1.952236e-08
Xơ phổi         1.503726e-04  1.000000  1.542309e-01
Bình thường     1.952236e-08  0.154231  1.000000e+00
--------------------------------------------------
Phân tích biến DmCO
```

```
-------------------------------------------------
 Phân nhóm Biến số         H  Độ tự do  Giá trị p Phủ định H0
     Class    DmCO  8.929143         2    0.01151      Có thể
-------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  0.299002     0.411588
Xơ phổi             0.299002  1.000000     0.008694
Bình thường         0.411588  0.008694     1.000000
-------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  0.223469     0.317760
Xơ phổi             0.223469  1.000000     0.006239
Bình thường         0.317760  0.006239     1.000000
-------------------------------------------------
Phân tích biến Thickness

-------------------------------------------------
 Phân nhóm    Biến số          H  Độ tự do     Giá trị p Phủ định H0
     Class  Thickness  31.130711         2  1.738009e-07      Có thể
-------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng       Xơ phổi  Bình thường
Khí phế thũng   1.000000e+00  1.381358e-07     0.001625
Xơ phổi         1.381358e-07  1.000000e+00     0.038338
Bình thường     1.625181e-03  3.833825e-02     1.000000
-------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng       Xơ phổi   Bình thường
Khí phế thũng   1.000000e+00  8.117064e-15  1.728766e-09
Xơ phổi         8.117064e-15  1.000000e+00  1.816640e-06
Bình thường     1.728766e-09  1.816640e-06  1.000000e+00
-------------------------------------------------
Phân tích biến DLCO

-------------------------------------------------
 Phân nhóm Biến số        H  Độ tự do  Giá trị p Phủ định H0
     Class    DLCO  3.25722         2   0.196202   Không thể
-------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  1.000000     0.445539
Xơ phổi             1.000000  1.000000     0.333138
Bình thường         0.445539  0.333138     1.000000
-------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng  Xơ phổi  Bình thường
```

```
Khí phế thũng       1.000000  1.00000     0.450294
Xơ phổi             1.000000  1.00000     0.340740
Bình thường         0.450294  0.34074     1.000000
--------------------------------------------------
Phân tích biến FVC

--------------------------------------------------
 Phân nhóm Biến số          H  Độ tự do  Giá trị p Phủ định H0
     Class     FVC  11.123843         2   0.003841      Có thể
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Dunn
               Khí phế thũng  Xơ phổi  Bình thường
Khí phế thũng       1.000000  1.00000     0.018824
Xơ phổi             1.000000  1.00000     0.012020
Bình thường         0.018824  0.01202     1.000000
--------------------------------------------------
So sánh bắt cặp bằng kiểm định Conover
               Khí phế thũng   Xơ phổi  Bình thường
Khí phế thũng       1.000000  1.000000     0.009036
Xơ phổi             1.000000  1.000000     0.005741
Bình thường         0.009036  0.005741     1.000000
--------------------------------------------------
```

Bài thức hành đến đây là hết, các bạn đã có trong tay một giải pháp phi tham số thay thế cho ANOVA đơn biến.

6 Ứng dụng

Kruskall Wallis được xem như 1 hình thức phân tích phương sai (ANOVA), cho phép so sánh đặc tính phân phối (hoặc trung vị) của Y giữa nhiều (3) phân nhóm độc lập.

Một số ứng dụng của kiểm định H Kruskal Wallis:

1) Thay thế cho phân tích phương sai 1 yếu tố (ANOVA đơn biến) trong trường hợp : cỡ mẫu quá thấp, hoặc có vi phạm các giả định về phân phối chuẩn và/hoặc có những outliers

2) Nên sử dụng test Kruskal Wallis khi đại lượng cần so sánh là một biến số không liên tục (thứ hạng, thang điểm). Thí dụ: Những nghiên cứu thuộc chuyên khoa chẩn đoán hình ảnh, giải phẫu bệnh lý thường dùng nhiều biến thứ hạng. Các nghiên cứu dùng thang điểm, bảng câu hỏi… (ví dụ thăm dò chức năng giấc ngủ, mức độ đau, chất lượng cuộc sống…). Những mô hình thí nghiệm trên động vật, tế bào, mô. Các thí nghiệm Western-Blot, hóa mô miễn dịch (do cỡ mẫu quá nhỏ, và/hoặc giá trị không liên tục).

3) Ngay cả khi số liệu phân phối chuẩn, không có điểm cá biệt, bạn vẫn có thể dùng test Kruskal-Wallis, lúc này nó được xem như 1 phương pháp độc lập, không chỉ thay thế cho ANOVA. Bản thân test Kruskal-Wallis dựa trên giả thuyết 0 về đặc tính phân phối, nên cho ra kết luận phổ quát hơn mô hình ANOVA chỉ cho phép chứng minh hiệu ứng ý nghĩa của yếu tố F lên giá trị trung bình của Y. Trong trường hợp ANOVA cho ra kết quả không rõ nét, bạn có thể cân nhắc dùng test Kruskal Wallis với hy vọng tìm ra sự khác biệt (vì ít bảo thủ hơn).

6.1 Lưu ý:

- Sau khi xếp hạng, phân phối của Y xem như được chuẩn hóa, bạn vẫn có thể sử dụng ANOVA đơn biến nếu thích, tuy không chính xác bằng mô hình theo thang đo định lượng liên tục. Ngược lại, test Kruskal Wallis vẫn cần giả định về đồng nhất phương sai.

7 Tài liệu tham khảo

1) Kruskal; Wallis (1952). "Use of ranks in one-criterion variance analysis". Journal of the American Statistical Association 47 (260): 583–621.

2) W. J. Conover, R. L. Iman (1979), On multiple-comparisons procedures, Tech. Rep. LA-7677-MS, Los Alamos Scientific Laboratory.

3) O.J. Dunn (1964). Multiple comparisons using rank sums. Technometrics, 6, 241-252.