

```
In [1]: 1 import numpy as np
2 import scipy.stats as sps
3 from tqdm import tqdm
4 import pandas as pd
5 from statsmodels.sandbox.stats.multicomp import multipletests
```

Комбинирование критериев

$(X_1, \dots, X_n), (Y_1, \dots, Y_m)$ --- выборки

Являются ли они нормальными или нет, однородными или нет?

H_1 : выборка (X_1, \dots, X_n) нормальна

H_2 : выборка (Y_1, \dots, Y_m) нормальна

H_3 : выборки однородны

```
In [2]: 1 def test(x, y, alpha=0.05):
2     """
3     Реализация комбинации критериев с условием FWER <= alpha.
4     Возвращает три булевских значения. Если True, то соответствующая гипотеза отклоняется.
5     """
6
7     result = np.zeros(3, dtype=bool)
8
9     # Первая процедура
10    rejected = multipletests([# критерий Шапиро-Уилка для выборки X
11                             sps.shapiro(x)[1],
12                             # критерий Шапиро-Уилка для выборки Y
13                             sps.shapiro(y)[1],
14                             # критерий Стьюдента для проверки однородности
15                             sps.ttest_ind(x, y)[1]],
16                             method='holm',
17                             alpha=alpha)[0]
18
19    # Результат проверки нормальности сразу пишем в ответ
20    result[:2] = rejected[:2]
21
22    # Если нормальность отвергается хотя бы для одной выборки,
23    # то для проверки однородности реализуем вторую процедуру,
24    # в которой в данном случае только критерий Уилкоксона-Манна-Уитни.
25    # Результат применения этой процедуры пишем в ответ.
26    if rejected[:2].any():
27        result[2] = sps.mannwhitneyu(x, y, alternative='two-sided')[1] < alpha
28
29    # Если нормальность не отвергается, то в ответ пишем результат Стьюдента
30    else:
31        result[2] = rejected[2]
32
33    return result
34
35
36 def result_table(rejected):
37     """ Составляет красивую таблицу """
38
39     # Трюк с категориальными нужен, чтобы явно указать множество значений.
40     # Если этого не сделать, то при отсутствии какого-то значения все полетит.
41     table = pd.crosstab(index=pd.Categorical(rejected[:, :2].any(axis=1),
42                                             categories=[False, True]),
43                         columns=pd.Categorical(rejected[:, 2],
44                                                 categories=[False, True]),
45                         normalize=True)
46
47     table.index = ['Нормальность не отвергается',
48                   'Нормальность отвергается хотя бы для одной']
49     table.columns = ['Однородность не отвергается',
50                     'Однородность отвергается']
51
52     return table
```

Параметры для всех запусков

```
In [3]: 1 size = 100 # размер выборки
2 n_iter = 200000 # количество итераций в одном эксперименте
3 alpha = 0.05 # уровень значимости
```

Эксперимент 1

Все гипотезы верны. Ошибка первого рода происходит, если хотя бы одна гипотеза отвергается. Из таблицы видно, что Большая часть ошибок приходится на отвержение нормальности. Это объясняется тем, что для этого используется два критерия, причем доля соответствующих ошибок примерно в 2 раза больше доле случаев отвержения однородности.

```
In [4]: 1 rejected = np.zeros((n_iter, 3), dtype=bool)
2 x = sps.norm.rvs(size=(n_iter, size))
3 y = sps.norm.rvs(size=(n_iter, size))
4
5 for i in tqdm(range(n_iter)):
6     rejected[i] = test(x[i], y[i])
7
8 n_errors = rejected.any(axis=1).sum()
9 print('FWER = {:.4f} +/- {:.4f}'.format(n_errors / n_iter, 2 * np.sqrt(alpha / n_iter)))
10 result_table(rejected)
```

100%|██████████| 200000/200000 [1:14:33<00:00, 44.70it/s]

FWER = 0.0495 +/- 0.0010

Out[4]:

	Однородность не отвергается	Однородность отвергается
Нормальность не отвергается	0.950535	0.016020
Нормальность отвергается хотя бы для одной	0.031070	0.002375

Эксперимент 2

Выборки нормальные, но имеют разное среднее. Ошибка первого рода происходит, если отвергается нормальность хотя бы одной из выборок. Во всех случаях комбинированный критерий верно отверг однородность.

```
In [5]: 1 rejected = np.zeros((n_iter, 3), dtype=bool)
2 x = sps.norm.rvs(size=(n_iter, size))
3 y = sps.norm(loc=1).rvs(size=(n_iter, size))
4
5 for i in tqdm(range(n_iter)):
6     rejected[i] = test(x[i], y[i])
7
8 n_errors = rejected[:, :2].any(axis=1).sum()
9 print('FWER = {:.4f} +/- {:.4f}'.format(n_errors / n_iter, 2 * np.sqrt(alpha / n_iter)))
10 result_table(rejected)
```

100%|██████████| 200000/200000 [1:15:42<00:00, 44.03it/s]

FWER = 0.0490 +/- 0.0010

Out[5]:

	Однородность не отвергается	Однородность отвергается
Нормальность не отвергается	0.0	0.951015
Нормальность отвергается хотя бы для одной	0.0	0.048985

Эксперимент 3

Выборки не являются нормальными, но при этом имеют распределение Стьюдента с 10 степенями свободы, которое достаточно похоже на нормальное. Ошибка первого рода происходит только при отвержении однородности, что происходит только примерно в $\alpha/2$ случаев. Почти в 3/4 случаев нормальность не отвергается, т.е. предположения критерия Стьюдента, вообще говоря, не выполнены. Тем не менее, он хорошо работает, поскольку выборки в целом обладают некоторыми свойствами нормальности.

```
In [6]: 1 rejected = np.zeros((n_iter, 3), dtype=bool)
2 x = sps.t(df=10).rvs(size=(n_iter, size))
3 y = sps.t(df=10).rvs(size=(n_iter, size))
4
5 for i in tqdm(range(n_iter)):
6     rejected[i] = test(x[i], y[i])
7
8 n_errors = rejected[:, 2].sum()
9 print('FWER = {:.4f} +/- {:.4f}'.format(n_errors / n_iter, 2 * np.sqrt(alpha / n_iter)))
10 result_table(rejected)
```

100%|██████████| 200000/200000 [1:21:09<00:00, 41.07it/s]

FWER = 0.0258 +/- 0.0010

Out[6]:

	Однородность не отвергается	Однородность отвергается
Нормальность не отвергается	0.71739	0.01102
Нормальность отвергается хотя бы для одной	0.25682	0.01477

Эксперимент 4

Аналогично предыдущему случаю, но используется распределение Стьюдента с 5 степенями свободы, которое уже менее похоже на нормальное. Это приводит к тому, что почти в 3/4 случаев нормальность все же отвергается. Реальное значение FWER выше, чем в предыдущем случае, но уровня α не достигает.

```
In [7]: 1 rejected = np.zeros((n_iter, 3), dtype=bool)
2 x = sps.t(df=5).rvs(size=(n_iter, size))
3 y = sps.t(df=5).rvs(size=(n_iter, size))
4
5 for i in tqdm(range(n_iter)):
6     rejected[i] = test(x[i], y[i])
7
8 n_errors = rejected[:, 2].sum()
9 print('FWER = {:.4f} +/- {:.4f}'.format(n_errors / n_iter, 2 * np.sqrt(alpha / n_iter)))
10 result_table(rejected)
```

100%|██████████| 200000/200000 [1:23:41<00:00, 39.83it/s]

FWER = 0.0407 +/- 0.0010

Out[7]:

	Однородность не отвергается	Однородность отвергается
Нормальность не отвергается	0.288485	0.003995
Нормальность отвергается хотя бы для одной	0.670800	0.036720

Эксперимент 5

Выборки по-прежнему однородны, но при этом имеют экспоненциальное распределение, которое совсем не похоже на нормальное. В каждом случае нормальность была отвергнута хотя бы для одной выборки (а может всегда для обеих выборок), поэтому однородность всегда проверялась во второй процедуре.

```
In [8]: 1 rejected = np.zeros((n_iter, 3), dtype=bool)
2 x = sps.expon.rvs(size=(n_iter, size))
3 y = sps.expon.rvs(size=(n_iter, size))
4
5 for i in tqdm(range(n_iter)):
6     rejected[i] = test(x[i], y[i])
7
8 n_errors = rejected[:, 2].sum()
9 print('FWER = {:.4f} +/- {:.4f}'.format(n_errors / n_iter, 2 * np.sqrt(alpha / n_iter)))
10 result_table(rejected)
```

100%|██████████| 200000/200000 [1:22:52<00:00, 40.22it/s]

FWER = 0.0497 +/- 0.0010

Out[8]:

	Однородность не отвергается	Однородность отвергается
Нормальность не отвергается	0.00000	0.00000
Нормальность отвергается хотя бы для одной	0.95031	0.04969