

In [1]:

```
1 import numpy as np
2 import scipy.stats as sps
3 from tqdm import tqdm
```

Проверка гипотезы однородности для нормальных выборок

In [2]:

```
1 def ftest(x, y):
2     """ Критерий Фишера """
3
4     fstat = x.var(ddof=1) / y.var(ddof=1)
5     pvalue = 2 * np.min([sps.f(len(x)-1, len(y)-1).sf(fstat),
6                           sps.f(len(x)-1, len(y)-1).cdf(fstat)])
7     return pvalue
```

Параметры для всех запусков

In []:

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

Эксперимент 1

Проверяем, что критерий Стьюдента правильно работает, т.е. верная гипотеза отвергается не более чем в α случаях.

In [3]:

```
1 n_errors = 0
2
3 for _ in tqdm(range(n_iter)):
4     x, y = sps.norm.rvs(size=(2, size))
5     n_errors += sps.ttest_ind(x, y, equal_var=True)[1] < alpha
6
7 print('{:.4f} +/- {:.4f}'.format(n_errors / n_iter,
8                                   2 * np.sqrt(alpha / n_iter)))
```

100%|██████████| 200000/200000 [00:45<00:00, 4348.58it/s]

0.0503 +/- 0.0010

Эксперимент 2

Проверяем, что критерий Фишера правильно работает

In [4]:

```
1 n_errors = 0
2
3 for _ in tqdm(range(n_iter)):
4     x, y = sps.norm.rvs(size=(2, size))
5     n_errors += ftest(x, y) < alpha
6
7 print('{:.4f} +/- {:.4f}'.format(n_errors / n_iter,
8                                   2 * np.sqrt(alpha / n_iter)))
```

100%|██████████| 200000/200000 [05:16<00:00, 631.78it/s]

0.0505 +/- 0.0010

Эксперимент 3

Для пары выборок применим сначала критерий Фишера. Если он не отвергает гипотезу, то применим критерий Стьюдента. Поправку на множественную проверку делать не будем. Как видим, реальный уровень значимости полученного критерия почти в 2 раза превышает заявленный.

In [5]:

```
1 n_errors = 0
2
3 for _ in tqdm(range(n_iter)):
4     x, y = sps.norm.rvs(size=(2, size))
5     is_reject = ftest(x, y) < alpha
6     n_errors += is_reject or (sps.ttest_ind(x, y, equal_var=True)[1] < alpha)
7
8 print('{:.4f} +/- {:.4f}'.format(n_errors / n_iter,
9                                   2 * np.sqrt(alpha / n_iter)))
```

100%|██████████| 200000/200000 [06:50<00:00, 487.68it/s]

0.0978 +/- 0.0010

Эксперимент 4

Теперь применим множественную проверку гипотез. Для этого достаточно сравнить минимальное pvalue со значением $\alpha/2$. Тут реальный уровень значимости соответствует заявленному.

In [6]:

```
1 n_errors = 0
2
3 for _ in tqdm(range(n_iter)):
4     x, y = sps.norm.rvs(size=(2, size))
5     pval1 = ftest(x, y)
6     pval2 = sps.ttest_ind(x, y, equal_var=True)[1]
7     n_errors += min(pval1, pval2) < alpha / 2
8
9 print('{:.4f} +/- {:.4f}'.format(n_errors / n_iter,
10                                2 * np.sqrt(alpha / n_iter)))
```

100%|██████████| 200000/200000 [06:55<00:00, 481.76it/s]

0.0494 +/- 0.0010

Прикладная статистика и анализ данных, 2019

Никита Волков

<https://mipt-stats.gitlab.io/> (<https://mipt-stats.gitlab.io/>).