In [1]:

```python
import numpy as np
import pandas as pd
import scipy.stats as sps
import matplotlib.pyplot as plt

%matplotlib inline
```

### Критерий Бартлетта

$X_{ij} \sim \mathcal{N}(\mu_j, \sigma_j^2), \;\; i = 1, \dots, n_j, \;\; j = 1, \dots, k$

$\mathrm{H}_0 \colon \sigma_1 = \ldots = \sigma_k$

`bartlett` [(https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.bartlett.html#scipy.stats.bartlett)](https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.bartlett.html#scipy.stats.bartlett) `(sample1, sample2, ...): statistic, pvalue`

### $F$-критерий однофакторного дисперсионного анализа

$X_{ij} \sim \mathcal{N}(\mu_j, \sigma^2), \;\; i = 1, \dots, n_j, \;\; j = 1, \dots, k$

$\mathrm{H}_0 \colon \mu_1 = \ldots = \mu_k$

`f_oneway` [(https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.f_oneway.html)](https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.f_oneway.html) `(sample1, sample2, ...): statistic, pvalue`

In [14]:

```python
samples = []
for i in range(5):
    samples.append(sps.norm.rvs(size=20+i))
sps.bartlett(*samples), sps.f_oneway(*samples)
```

Out[14]:

```
(BartlettResult(statistic=5.302409501752963, pvalue=0.2576514519129599),
 F_onewayResult(statistic=0.7483479585522701, pvalue=0.5612738682824921))
```

In [15]:

```python
samples = []
for i in range(5):
    samples.append(sps.norm(loc=i).rvs(size=20+i))
sps.bartlett(*samples), sps.f_oneway(*samples)
```

Out[15]:

```
(BartlettResult(statistic=9.146888050337433, pvalue=0.057531028801199424),
 F_onewayResult(statistic=46.38914385075102, pvalue=2.1428650506386594e-22))
```

In [16]:

```python
samples = []
for i in range(5):
    samples.append(sps.norm(scale=1+i/2).rvs(size=20+i))
sps.bartlett(*samples), sps.f_oneway(*samples)
```

Out[16]:

```
(BartlettResult(statistic=35.02702491180489, pvalue=4.586347994808979e-07),
 F_onewayResult(statistic=1.3661820497330388, pvalue=0.2507041179938111))
```

In [19]:

```python
samples = []
for i in range(5):
    samples.append(sps.norm(loc=i, scale=1+i/2).rvs(size=20+i))
sps.bartlett(*samples), sps.f_oneway(*samples)
```

Out[19]:

```
(BartlettResult(statistic=19.614971279326745, pvalue=0.0005948230266618849),
 F_onewayResult(statistic=14.064849135285003, pvalue=3.186325091862291e-09))
```

### Критерий Краскела-Уоллиса

$X_{ij}, \quad i = 1, \dots, n_j, \quad j = 1, \dots, k$-- однофакторная модель, случай независмых выборок

$\mathrm{H}_0\colon \mu_1 = \dots = \mu_k$

`kruskal` [(https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.kruskal.html#scipy.stats.kruskal)](https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.kruskal.html#scipy.stats.kruskal) `(sample1, sample2, ...)`: `statistic, pvalue`

In [20]:
```
samples = []
for i in range(5):
    samples.append(sps.norm.rvs(size=20+i))
sps.kruskal(*samples)
```

Out[20]: `KruskalResult(statistic=3.9994042474280036, pvalue=0.4060864820570774)`

In [21]:
```
samples = []
for i in range(5):
    samples.append(sps.norm(loc=i).rvs(size=20+i))
sps.kruskal(*samples)
```

Out[21]: `KruskalResult(statistic=76.38012704917844, pvalue=1.0172837687548495e-15)`

In [22]:
```
samples = []
for i in range(5):
    samples.append(sps.norm(scale=i+1).rvs(size=20+i))
sps.kruskal(*samples)
```

Out[22]: `KruskalResult(statistic=0.854885258956358, pvalue=0.9309421349904955)`

In [23]:
```
samples = []
for i in range(5):
    samples.append(sps.expon.rvs(size=20+i))
sps.kruskal(*samples)
```

Out[23]: `KruskalResult(statistic=0.9416866792360565, pvalue=0.918505484762733)`

In [24]:
```
samples = []
for i in range(5):
    samples.append(sps.expon(scale=i+1).rvs(size=20+i))
sps.kruskal(*samples)
```

Out[24]: `KruskalResult(statistic=19.558231702777107, pvalue=0.0006103333303977086)`

## Критерий Фридмана

$X_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, k$-- однофакторная модель, случай связных выборок

$\mathrm{H}_0\colon \beta_1 = \dots = \beta_k$

`friedmanchisquare` [(https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.friedmanchisquare.html#scipy.stats.friedmanchisquare)](https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.stats.friedmanchisquare.html#scipy.stats.friedmanchisquare) `(sample1, sample2, ...)`: `statistic, pvalue`

Все выборки одинакового размера, количество выборок не менее 3.

In [34]:
```
sample_size = 30
factor_size = 5
alpha = np.linspace(0, 10, sample_size)[:, np.newaxis]
beta = np.zeros(factor_size)[np.newaxis, :]

samples = sps.norm(loc=1+alpha+beta).rvs()
print(samples.shape)
sps.friedmanchisquare(*samples.T)
```

```
(30, 5)
```

Out[34]:
```
FriedmanchisquareResult(statistic=3.8666666666666742, pvalue=0.424351184807717
83)
```

In [35]:
```
beta = np.arange(factor_size)[np.newaxis, :]

samples = sps.norm(loc=1+alpha+beta).rvs()
print(samples.shape)
sps.friedmanchisquare(*samples.T)
```

```
(30, 5)
```

Out[35]:
```
FriedmanchisquareResult(statistic=82.42666666666673, pvalue=5.3299900664472323
e-17)
```

In [37]:
```
beta = np.arange(factor_size)[np.newaxis, :]

samples = sps.norm(loc=1+alpha+beta, scale=alpha).rvs()
print(samples.shape)
sps.friedmanchisquare(*samples.T)
```

```
(30, 5)
```

Out[37]:
```
FriedmanchisquareResult(statistic=10.186666666666724, pvalue=0.037397999848609
37)
```

---

Прикладная статистика и анализ данных, 2019

Никита Волков

[https://mipt-stats.gitlab.io/ (https://mipt-stats.gitlab.io/)](https://mipt-stats.gitlab.io/)