In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import t as student
```

In [2]:
```
def get_tukey_m(x, y):
    """
    returns statisticks fot Tukey criteria
    :param x: first iterative data
    :param y: the other iterative data
    :return: statistic
    """
    return abs(x.mean() - y.mean()) / (s * (n/2)**(1/2))


def get_statistic_for_pval(k, n, alpha):
    """
    returns statistic for given p-value
    :param k: number of candidates
    :param n: number of
    :param alpha: given value
    :return: statistic for this parametrs
    """
```

Задача 4.2

Рассмотрим некоторую задачу классификации. Пусть задано качество **4** моделей `a1`, `a2`, `a3`, `a4`. Качество полученных моделей показано в таблице.

Исследователю требуется выбрать наилучшую модель. Для выбора лучшей модели исследовать требуется попарно сравнить среднее значение качества всех моделей. Может ли исследователь утверждать что какая-то из моделей лучше другой?

Требуется:

- записать задачу формально;
- предложить статистику для решения данной задачи;
- записать нулевое распределение данной статистики;
- записать явно правило принятия решения на основе статистики и нулевого распределения для обеспечения уровня значимости alpha = 0.05;
- проверить гипотезу по записанному критерию, для данных из условия. Противоречат ли они гипотезе?

In [3]:
```
data = pd.read_csv('data/classifiers.csv')
data.drop(columns=['Номер выборки'], inplace=True)
data.head()
alpha = 0.05
```

In [4]:
```
data.max()
```

Out[4]:
```
a1    86
a2    92
a3    99
a4    51
dtype: int64
```

Похоже что указана точность в процентах

1. Записать задачу формально

Будем пытаться разлчить a_i и a_j с помощью некоторого критерия. Если они разлечим, следующим шагом решим, какая лучше.

H_0 $a_i > a_j$ для фиксированного i и остальных j

Надо только понять какое это i

Воспользуемся критерием Тьюки, **т.к.** всравнивае все со всеми

Подсчитаем S. Буем считать что в каждой выборке 100 элементов, ну там 99 / 100 лучшее значние

In [5]:
```
_, k = data.shape
s_k2 = data.var(axis=0).values
n_k = [100] * k
n = k / sum(1/n_k[i] for i in range(k))
S2 = sum([ (n_k[i] -1)*s_k2[i] / (n - k) for i in range(k)])
hsd = student.ppf(1-alpha, n-k) * np.sqrt(S2 / n)
```

In [6]:
```
for i in range(k):
    for j in range(i+1, k):
        diff = np.abs(data.values[:, i].mean() - data.values[:, i].mean())
        is_sep = np.abs(data.values[:, i].mean() - data.values[:, j].mean()) > hsd
        print(f'{data.columns[i]} vs {data.columns[j]}:, is it separable? {is_sep}')
```

```
a1 vs a2:, is it separable? True
a1 vs a3:, is it separable? True
a1 vs a4:, is it separable? True
a2 vs a3:, is it separable? False
a2 vs a4:, is it separable? True
a3 vs a4:, is it separable? True
```

Вот это удача! Почти все мы можем упорядочить. Сделаем же это!

In [7]:
```
df = pd.DataFrame(index=data.columns.values, columns=['median'])
df['median'] = data.median()
df.sort_values('median', inplace=True)
df
```

Out[7]:

	median
a4	30.5
a1	48.5
a3	57.0
a2	59.0

Вот ето удача! Мы не можем разлечить, чья медиана больше - у 2ого или у третьего, и именно они претендуют на лучший результат. Ну чтож, наука здесь бессильна!

Вывод: на самом деле бывает, что победителей двое - $a2$ и $a3$