

```
In [3]: import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sps
import pandas as pd
from scipy import stats
from statsmodels.sandbox.stats.multicomp import multipletests

%matplotlib inline
```

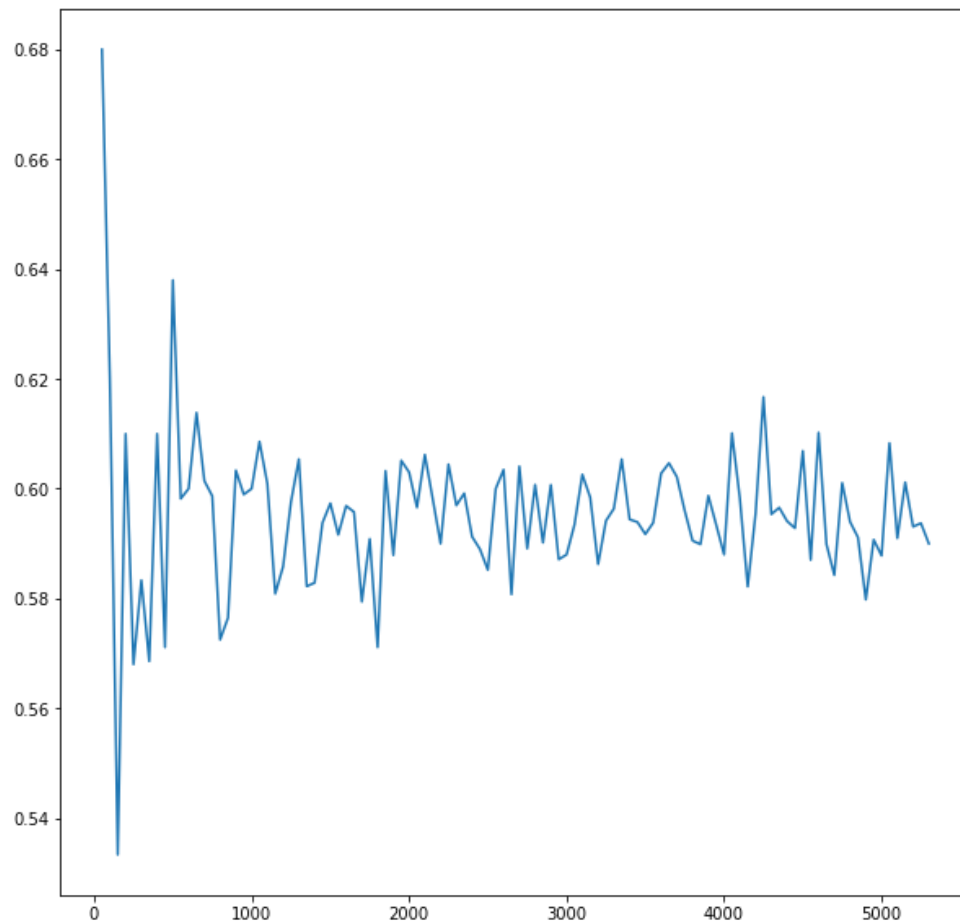
```
In [13]: def criteria_true_pos_rate(n = 100 , size = 100 ,alpha = 0.05):
    tpr = 0
    for i in range(size):
        sample_1 = sps.t(df=10).rvs(size=n)
        sample_2 = sps.t(df=10).rvs(size=n)

        cr = sps.ttest_ind(sample_1, sample_2)[0]
        if (abs(cr) < sps.t(df = 2*n -2).cdf(1 - alpha / 2) ):
            tpr += 1

    return tpr
```

```
In [*]: Y = np.arange(50, 5000, 50)
X = []
for y in Y:
    res = criteria_true_pos_rate(size = y)
    X.append(res)
```

```
In [17]: X = np.array([X[i] / Y[i] for i in range(len(X))])  
  
plt.figure(figsize=(10, 10))  
plt.plot(Y[:len(X)-1], X[:-1])  
plt.show()
```



Уровень значимости данной статистики близок к 0.6, а значит количество ошибок первого рода настолько огромно, что лучше избежать её использования