

# Практическая работа #8

1.

Провести дисперсионный анализ для определения того, есть ли различия среднего роста среди взрослых футболистов, хоккеистов и штангистов. Даны значения роста в трех группах случайно выбранных спортсменов: Футболисты: 173, 175, 180, 178, 177, 185, 183, 182. Хоккеисты: 177, 179, 180, 188, 177, 172, 171, 184, 180. Штангисты: 172, 173, 169, 177, 166, 180, 178, 177, 172, 166, 170.

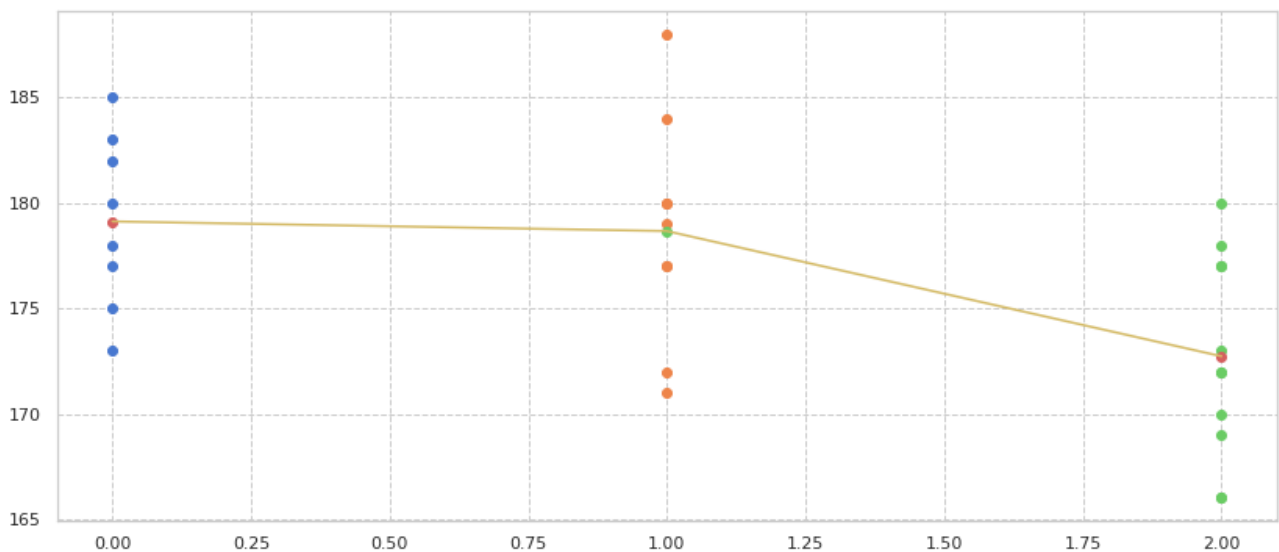
Условия для дисперсионного анализа:

1. Значения следуют нормальному распределению,
2. Однородность дисперсий,
3. Независимость

```
In [114... import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
%matplotlib inline
```

```
In [115... footballs = np.array([173, 175, 180, 178, 177, 185, 183, 182])
hockeys = np.array([177, 179, 180, 188, 177, 172, 171, 184, 180])
weightlifters = np.array([172, 173, 169, 177, 166, 180, 178, 177, 172, 166, 170])
```

```
In [116... plt.figure(figsize=(14, 6))
plt.grid(linestyle='--')
plt.scatter([0]*len(footballs), footballs)
plt.plot(0, footballs.mean(), 'ro')
plt.scatter([1]*len(hockeys), hockeys)
plt.plot(1, hockeys.mean(), 'go')
plt.scatter([2]*len(weightlifters), weightlifters)
plt.plot(2, weightlifters.mean(), 'ro');
plt.plot([0, 1, 2], [footballs.mean(), hockeys.mean(), weightlifters.mean()], 'y');
```



```
In [117... n1, n2, n3 = len(footballs), len(hockeys), len(weightlifters)
k = 3
y1_mean = footballs.mean()
y2_mean = hockeys.mean()
y3_mean = weightlifters.mean()
Y = np.concatenate([footballs, hockeys, weightlifters])
Y_mean = Y.mean()
# сумма квадратов отклонений значений из всех групп от общего среднего
s2 = np.sum((Y - Y_mean)**2)

# сумма квадратов отклонений значений групповых средних от общего среднего
s2_f = ((y1_mean - Y_mean)**2) * n1 + ((y2_mean - Y_mean)**2) * n2 + ((y3_mean - Y_mean)**2) * n3

# остаточная сумма квадратов отклонений значений каждой из групп от среднего каждой из групп
s2_residual = np.sum((footballs - y1_mean)**2) + np.sum((hockeys - y2_mean)**2) + np.sum((weightlifters - y3_mean)**2)

s2, s2_f + s2_residual
```

```
Out[117... (830.9642857142854, 830.964285714286)
```

```
In [118... # факторная дисперсия
# факторная сумма квадратичных отклонений делится на степень свободы числителя
sigma2_f = s2_f / (k-1)
sigma2_f
```

```
Out[118... 126.9537337662339
```

```
In [119... # остаточная дисперсия  
# остаточная сумма квадратов отклонений делится на внутреннюю степень свободы  
sigma2_residual = s2_residual / (sum([n1, n2, n3]) - k)  
sigma2_residual
```

```
Out[119... 23.08227272727273
```

```
In [120... # критерий Фишера  
F_h = sigma2_f / sigma2_residual  
F_h
```

```
Out[120... 5.500053450812598
```

```
In [121... stats.f_oneway(footballs, hockeys, weightlifters)
```

```
Out[121... F_onewayResult(statistic=5.500053450812596, pvalue=0.010482206918698694)
```