РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Дисциплина: Анализ данных

Студент: Леонова Алина

Группа: НФИбд-02-17

Москва 2020

Применение критерия Хи-квадрат

In [1]:
```
import numpy as np
import scipy as sp
import scipy.stats as stats
import pandas as pd
import math
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns; sns.set()
```

In [2]:
```
data = pd.read_csv('StudentsPerformance.csv')
N = len(data)
data
```

Out[2]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
gender                         1000 non-null object
race/ethnicity                 1000 non-null object
parental level of education    1000 non-null object
lunch                          1000 non-null object
test preparation course        1000 non-null object
math score                     1000 non-null int64
reading score                  1000 non-null int64
writing score                  1000 non-null int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

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

Out[4]:

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

Необходимо проверить критерием Хи-квадрат наличие зависимости между заданными параметрами

- H0: между параметрами нет зависимости
- На (H3): между параметрами есть зависимость

Задание 1. Параметры 'lunch' и 'reading score'

In [5]:
```
l = np.array(data['lunch'].values)
rs = np.array(data['reading score'].values)
```

График распределения параметра lunch

In [6]:
```
sns.countplot(l)
```

Out[6]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b07a0340>
```

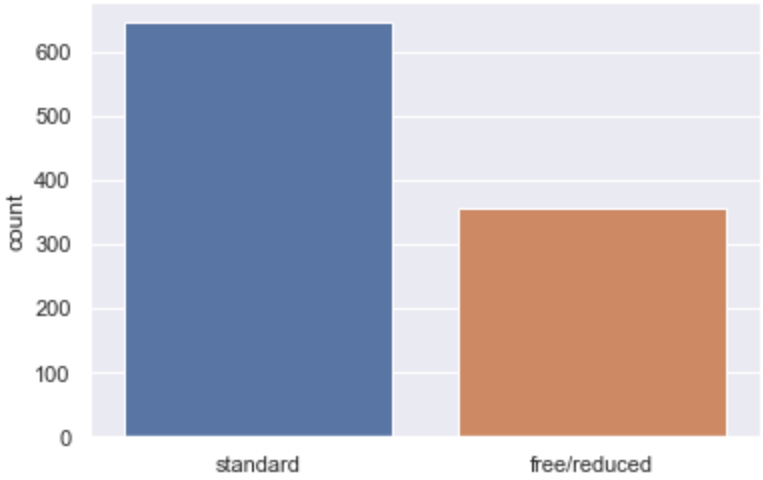


График распределения параметра reading score

In [7]:
```
sns.distplot(rs)
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b084c6d0>
```

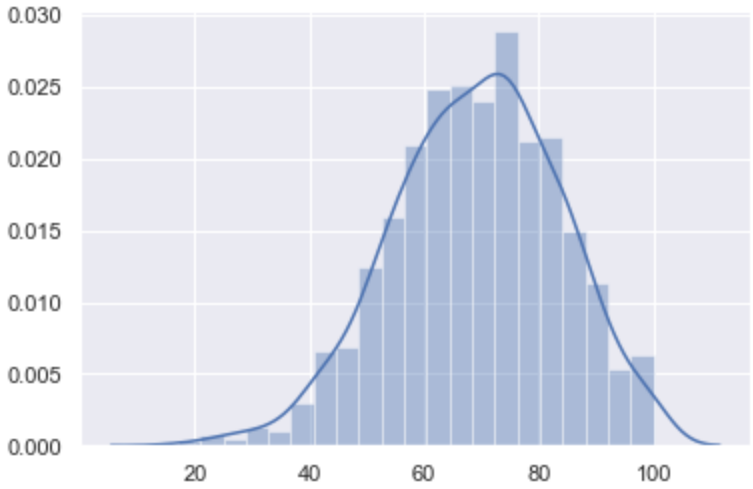
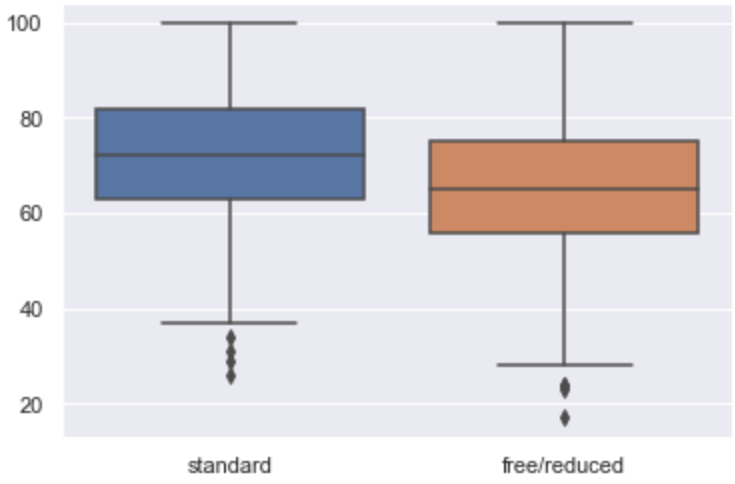


График совместного распределения параметров lunch и reading score

In [8]:
```
sns.boxplot(l, rs)
```

Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b08c46d0>
```



Разбивание баллов за чтение на интервалы

In [9]:
```
z = np.vstack((l, rs))

l1 = np.array([]); l2 = np.array([]); l3 = np.array([]); rs1 = np.array([]); rs2 = np.array([]); r

for i in range(N):
    if z[1][i] < 50:
        l1 = np.append(l1, z[0][i])
        rs1 = np.append(rs1, z[1][i])
    elif z[1][i] >= 50 and z[1][i] < 85:
        l2 = np.append(l2, z[0][i])
        rs2 = np.append(rs2, z[1][i])
    else:
        l3 = np.append(l3, z[0][i])
        rs3 = np.append(rs3, z[1][i])
```

Таблица сопряженности

In [10]:
```
s1 = np.sum(l1 == 'standard')
s2 = np.sum(l2 == 'standard')
s3 = np.sum(l3 == 'standard')
fr1 = len(l1) - s1
fr2 = len(l2) - s2
fr3 = len(l3) - s3

a = np.array([
    [s1, s2, s3],
    [fr1, fr2, fr3]
])

table1 = pd.DataFrame(a, columns = ['0-49', '50 - 84', '85 - 100'], index = ['standard', 'free/rec
table1
```

Out[10]:

	0-49	50 - 84	85 - 100
standard	39	491	115
free/reduced	51	269	35

Значение статистики хи-квадрат

In [11]:
```
def hi_kw(a, n):
    ni = []; nj = []

    # m - число значений lunch
    # k - число значений reading score
    (m, k) = a.shape

    for i in range(m):
        sum = 0
        for j in range(k):
            sum += a[i][j]
        ni.append(sum)

    for j in range(k):
        sum = 0
        for i in range(m):
            sum += a[i][j]
        nj.append(sum)

    hi_kw = 0
    for i in range(m):
        for j in range(k):
            hi_kw += a[i][j]**2 / (ni[i] * nj[j])
```

```
    hi_kw -= 1
    hi_kw *= n

    return hi_kw


hikw1 = hi_kw(a, N)
hikw1
```

Out[11]:
```
27.31088010450833
```

p-значение

In [12]:
```
(m, k) = a.shape
r = (m - 1) * (k - 1)
pv1 = 1 - stats.chi2.cdf(hikw1, df = r)
print(pv1)
print('H0 верна?', pv1 > 0.05)
```

```
1.173593613179591e-06
H0 верна? False
```

- p-значение значительно меньше 0.05

Между параметрами имеется зависимость. **Отвергаем гипотезу H0 в пользу альтернативной.**

Мера Крамера

In [13]:
```
C1 = np.sqrt(hikw1 / (N * min(m - 1, k - 1)))
C1
```

Out[13]:
```
0.16526003783282978
```

- Мера Крамера < 0.3

Вывод: между параметрами есть слабая зависимость

Задание 2. Параметры 'race/ethnicity' и 'writing score'

In [14]:
```
re = np.array(data['race/ethnicity'].values)
ws = np.array(data['writing score'].values)
```

График распределения параметра race/ethnicity

In [15]:
```
sns.countplot(re)
```

Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b095e4c0>
```

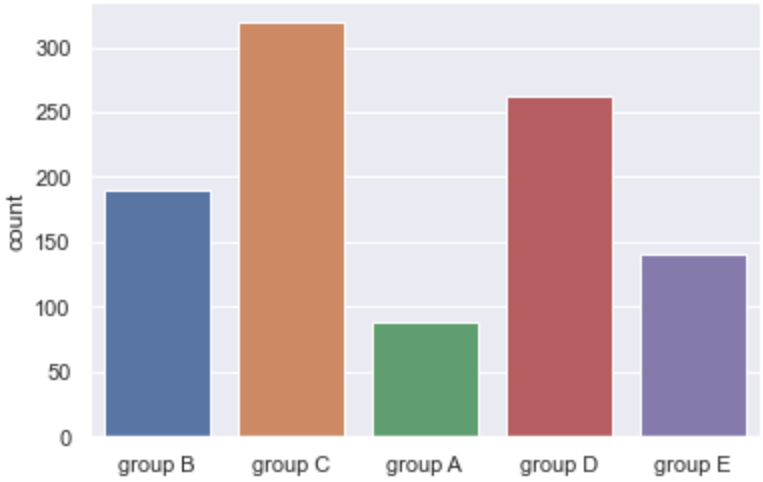


График распределения параметра writing score

In [16]:
```
sns.distplot(ws)
```

Out[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b0854ac0>
```

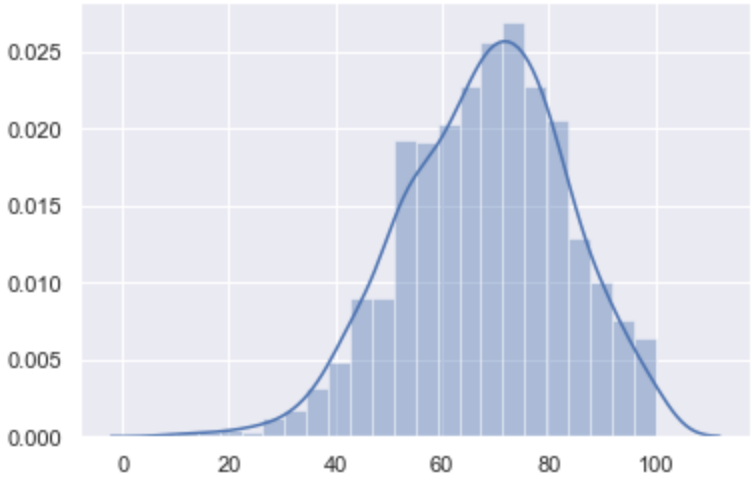
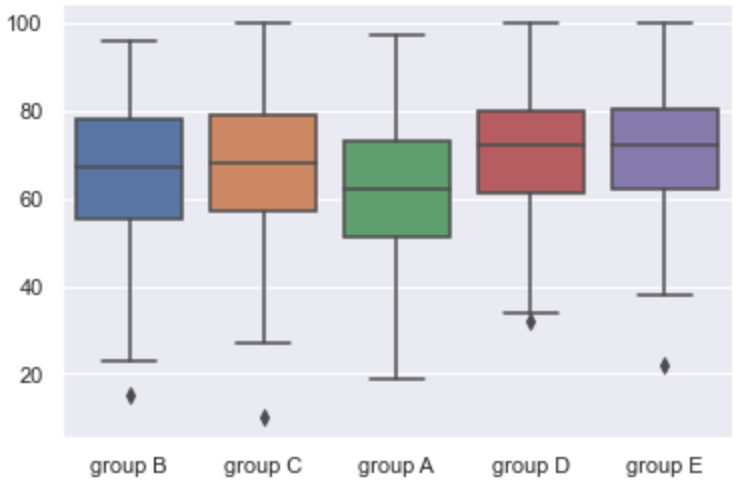


График совместного распределения параметров race/ethnicity и writing score

In [17]:
```
sns.boxplot(re, ws)
```

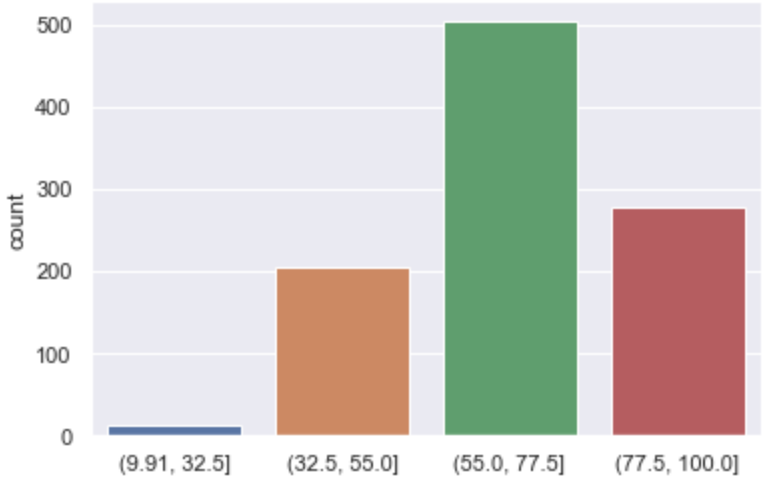
Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b09ea3d0>
```



Разбивание баллов за письмо на интервалы

In [18]:
```
ws_cat = pd.cut(ws, bins = 4)
sns.countplot(ws_cat)
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b0ad6280>
```



In [19]:
```
df2 = pd.DataFrame(np.vstack([re, ws_cat]).T, columns = ['re', 'ws'])
df2
```

Out[19]:

	re	ws
0	group B	(55.0, 77.5]
1	group C	(77.5, 100.0]
2	group B	(77.5, 100.0]
3	group A	(32.5, 55.0]
4	group C	(55.0, 77.5]
...	...	...
995	group E	(77.5, 100.0]
996	group C	(32.5, 55.0]
997	group C	(55.0, 77.5]
998	group D	(55.0, 77.5]
999	group D	(77.5, 100.0]

1000 rows × 2 columns

Таблица сопряженности

In [20]:
```
table2 = pd.crosstab(df2['ws'], df2['re'])
b = np.array(table2)
table2
```

Out[20]:

re	group A	group B	group C	group D	group E
ws					
(9.91, 32.5]	1	6	3	1	1
(32.5, 55.0]	30	42	71	41	21
(55.0, 77.5]	42	93	160	138	71

re ws	group A	group B	group C	group D	group E
(77.5, 100.0]	16	49	85	82	47

Значение статистики хи-квадрат

In [21]:
```
hikw2 = hi_kw(b, N)
hikw2
```

Out[21]:
```
28.21476955336233
```

p-значение

In [22]:
```
(m2, k2) = b.shape
r2 = (m2 - 1) * (k2 - 1)
pv2 = 1 - stats.chi2.cdf(hikw2, df = r2)
print(pv2)
print('H0 верна?', pv2 > 0.05)
```

```
0.00514536770378593
H0 верна? False
```

- p-значение меньше 0.05

Между параметрами имеется зависимость. **Отвергаем гипотезу H0 в пользу альтернативной.**

Мера Крамера

In [23]:
```
C2 = np.sqrt(hikw2 / (N * min(m2 - 1, k2 - 1)))
C2
```

Out[23]:
```
0.0969789832100446
```

- Мера Крамера < 0.3

Вывод: между параметрами есть слабая зависимость

Задание 3. Параметры 'math score' и 'reading score'

In [24]:
```
ms = np.array(data['math score'].values)
rs = np.array(data['reading score'].values)
```

График распределения параметра math score

In [25]:
```
sns.distplot(ms)
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b09bb670>
```

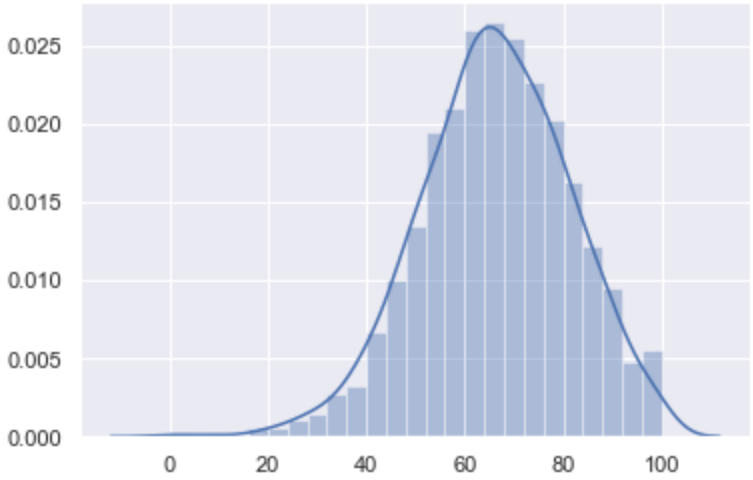


График распределения параметра reading score

In [26]:
```
sns.distplot(rs)
```

Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b0be8610>
```

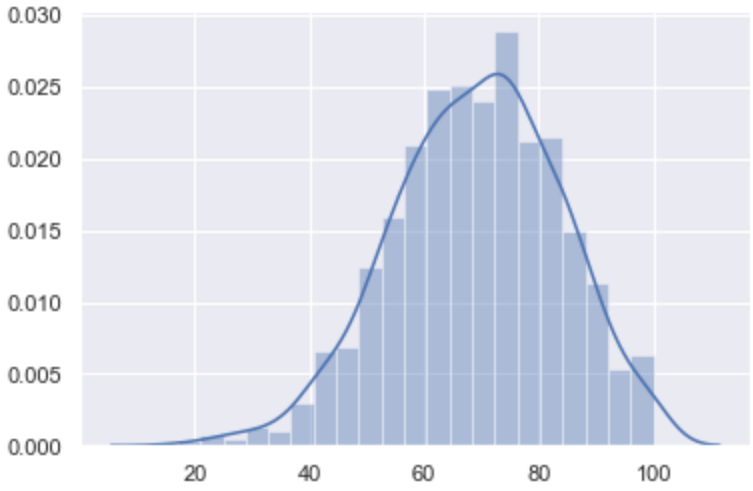
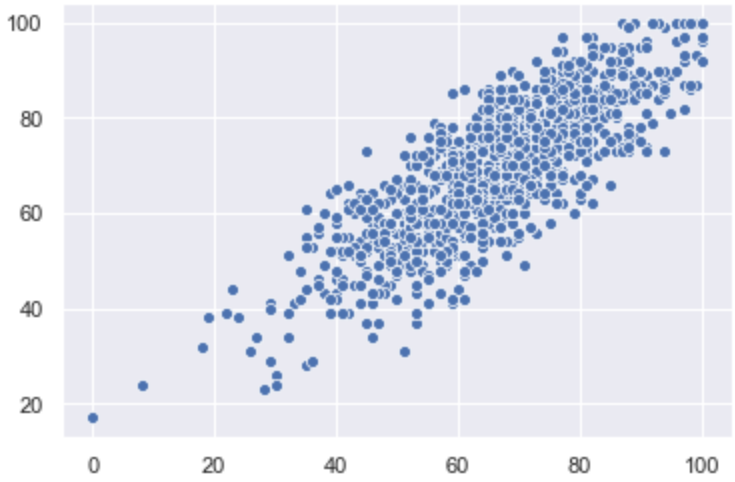


График совместного распределения параметров math score и reading score

In [27]:
```
sns.scatterplot(ms, rs)
```

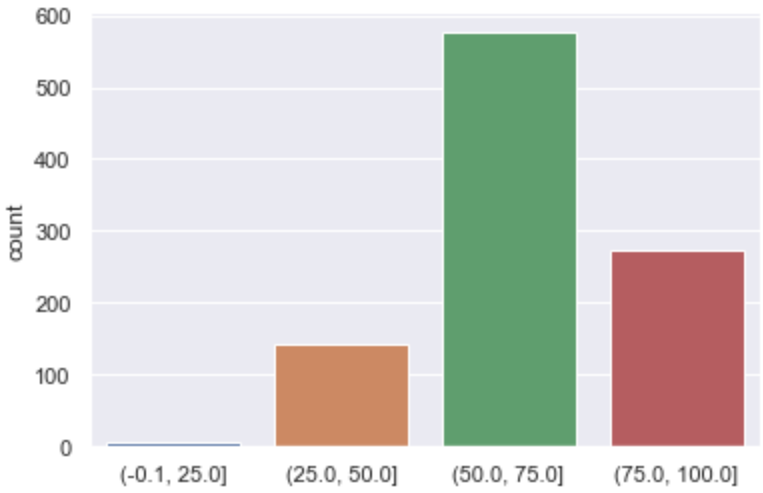
Out[27]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b0c27c10>
```



Разбивание выборок на интервалы

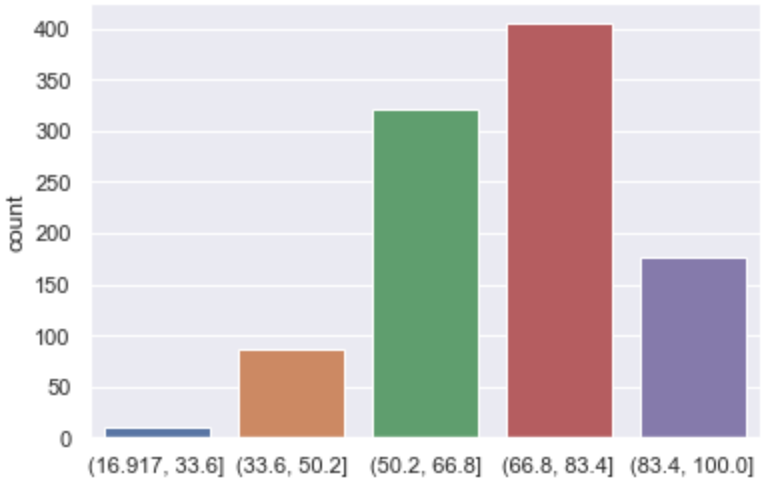
In [28]:
```
ms_cat = pd.cut(ms, bins = 4)
sns.countplot(ms_cat)
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b0ce19a0>
```



In [29]:
```
rs_cat = pd.cut(rs, bins = 5)
sns.countplot(rs_cat)
```

Out[29]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x226b0ceea90>
```



In [30]:
```
df3 = pd.DataFrame(np.vstack([ws_cat, rs_cat]).T, columns = ['ws', 'rs'])
df3
```

Out[30]:

	ws	rs
0	(55.0, 77.5]	(66.8, 83.4]
1	(77.5, 100.0]	(83.4, 100.0]
2	(77.5, 100.0]	(83.4, 100.0]
3	(32.5, 55.0]	(50.2, 66.8]
4	(55.0, 77.5]	(66.8, 83.4]
...	...	...
995	(77.5, 100.0]	(83.4, 100.0]
996	(32.5, 55.0]	(50.2, 66.8]
997	(55.0, 77.5]	(66.8, 83.4]

	ws	rs
998	(55.0, 77.5]	(66.8, 83.4]
999	(77.5, 100.0]	(83.4, 100.0]

1000 rows × 2 columns

Таблица сопряженности

In [31]:
```
table3 = pd.crosstab(df3['ws'], df3['rs'])
c = np.array(table3)
table3
```

Out[31]:

ws \ rs	(16.917, 33.6]	(33.6, 50.2]	(50.2, 66.8]	(66.8, 83.4]	(83.4, 100.0]
(9.91, 32.5]	9	3	0	0	0
(32.5, 55.0]	2	81	122	0	0
(55.0, 77.5]	0	2	200	295	7
(77.5, 100.0]	0	0	0	110	169

Значение статистики хи-квадрат

In [32]:
```
hikw3 = hi_kw(c, N)
hikw3
```

Out[32]:
```
1583.9517232323285
```

p-значение

In [33]:
```
(m3, k3) = c.shape
r3 = (m3 - 1) * (k3 - 1)
pv3 = 1 - stats.chi2.cdf(hikw3, df = r3)
print(pv3)
print('H0 верна?', pv3 > 0.05)
```

```
0.0
H0 верна? False
```

- p-значение значительно меньше 0.05

Между параметрами есть зависимость. **Отвергаем гипотезу H0 в пользу альтернативной.**

Мера Крамера

In [34]:
```
C3 = np.sqrt(hikw3 / (N * min(m3 - 1, k3 - 1)))
C3
```

Out[34]:
```
0.7266250117798791
```

Мера Крамера > 0.3

Вывод: между параметрами есть сильная зависимость
