Банки — Сегментация пользователей по потреблению

Задача:

Анализ и сегментация клиентов регионального банка по количеству потребляемых продуктов.

- Исследовательский анализ данных;
- Сегментация пользователей на основе данных о количестве потребляемых продуктов;
- Формулировка и проверка статистических гипотез различия дохода между теми клиентами, которые пользуются двумя продуктами банка, и теми, которые пользуются одним.

Описание данных

Датасет содержит данные о клиентах банка «Метанпром». Банк располагается в Ярославле и областных городах: Ростов Великий и Рыбинск.

- `userid` — идентификатор пользователя,
- `score` — баллы кредитного скоринга,
- `City` — город,
- `Gender` — пол,
- `Age` — возраст,
- `Objects` — количество объектов в собственности,
- `Balance` — баланс на счёте,
- `Products` — количество продуктов, которыми пользуется клиент,
- `CreditCard` — есть ли кредитная карта,
- `Loyalty` — активный клиент,
- `estimated_salary` — заработная плата клиента,
- `Churn` — ушёл или нет.

Оглавление

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import math
import datetime
from IPython.display import display
from plotly import graph_objects as go
from scipy import stats as st
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import Lasso, Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import KMeans
```

Шаг 1. Предобработка и исследовательский анализ данных

In [2]:
```
bank_dataset = pd.read_csv('/datasets/bank_dataset.csv')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 12 columns):
userid              10000 non-null int64
score               10000 non-null int64
City                10000 non-null object
Gender              10000 non-null object
Age                 10000 non-null int64
Objects             10000 non-null int64
Balance             6383 non-null float64
Products            10000 non-null int64
CreditCard          10000 non-null int64
Loyalty             10000 non-null int64
estimated_salary    10000 non-null float64
Churn               10000 non-null int64
dtypes: float64(2), int64(8), object(2)
memory usage: 937.6+ KB
```

In [4]:
```
bank_dataset.duplicated().sum()
```

Out[4]: 0

Столбец `Balance` содержит 3617 пропусков. Предположим, что пропущенное значение — это нулевой баланс на счёте. Заменим пропуски нулями.

In [5]:
```
# bank_dataset['Balance'] = bank_dataset['Balance'].fillna(0)
```

Вывод: Дататасет состоит из 12 столбцов и 10000 строк. Названия столбцов одностилевые, не стану их заменять. Дубликаты отсутствуют.

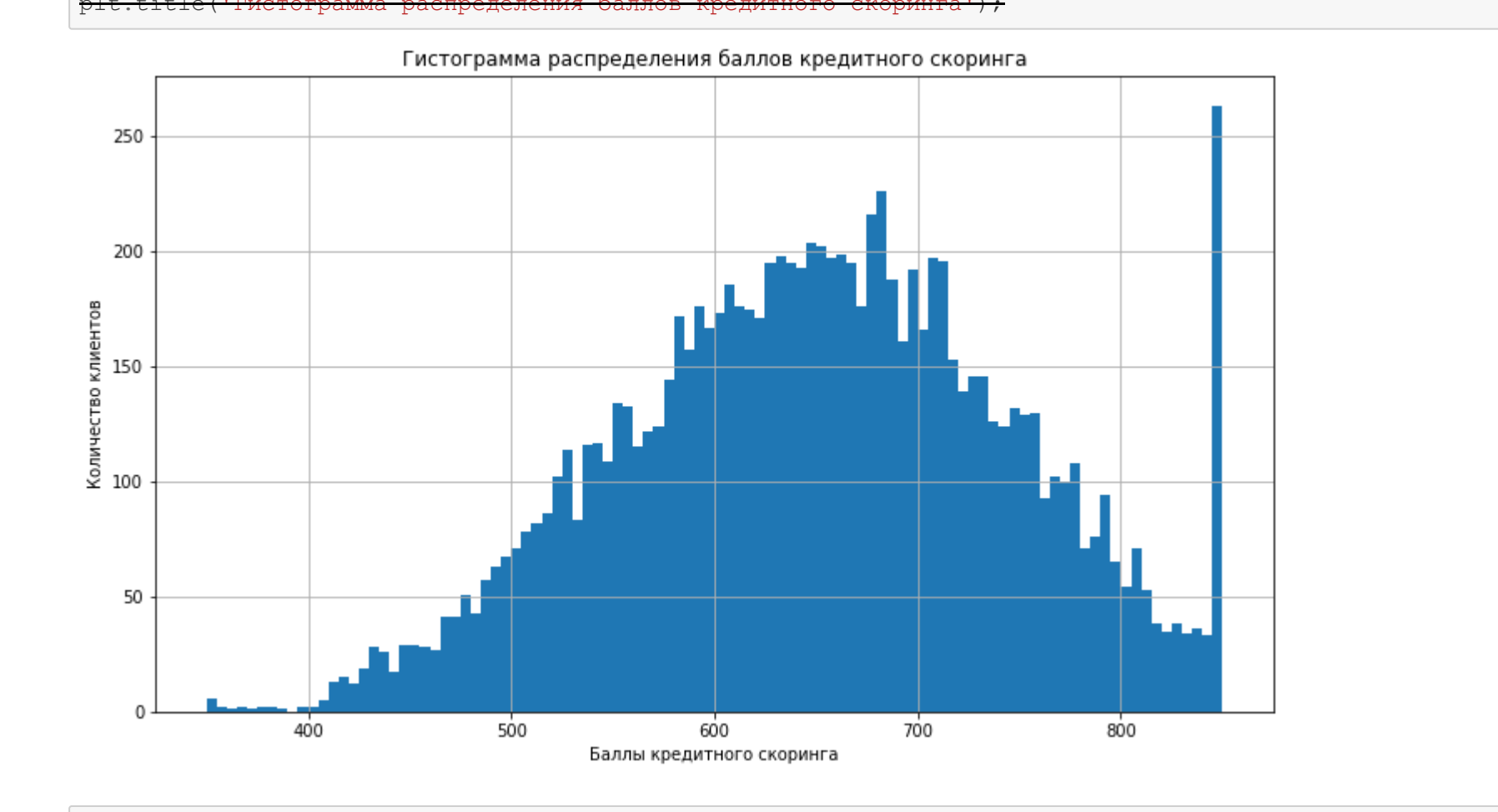
Проанализируем данные методом `describe()` и построим гистограммы распределения основных параметров.

In [6]:
```
bank_dataset.describe().T
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
userid	10000.0	1.573368e+07	71936.186123	15608437.00	1.567126e+07	1.573347e+07	1.579597e+07	15858426.00
score	10000.0	6.505288e+02	96.653299	350.00	5.840000e+02	6.520000e+02	7.180000e+02	850.00
Age	10000.0	3.892180e+01	10.487806	18.00	3.200000e+01	3.700000e+01	4.400000e+01	92.00
Objects	10000.0	5.012800e+00	2.892174	0.00	3.000000e+00	5.000000e+00	7.000000e+00	10.00
Balance	6383.0	1.198275e+05	30095.056462	3768.69	1.001820e+05	1.198397e+05	1.395123e+05	250898.09
Products	10000.0	1.530200e+00	0.581654	1.00	1.000000e+00	1.000000e+00	2.000000e+00	4.00
CreditCard	10000.0	7.055000e-01	0.455840	0.00	0.000000e+00	1.000000e+00	1.000000e+00	1.00
Loyalty	10000.0	5.151000e-01	0.499797	0.00	0.000000e+00	1.000000e+00	1.000000e+00	1.00
estimated_salary	10000.0	1.000902e+05	57510.492818	11.58	5.100211e+04	1.001939e+05	1.493882e+05	199992.48
Churn	10000.0	2.037000e-01	0.402769	0.00	0.000000e+00	0.000000e+00	0.000000e+00	1.00

In [7]:
```
bank_dataset['score'].hist(figsize=(12, 7), bins=100)
ax = plt.gca()
ax.set_xlabel('Баллы кредитного скоринга')
ax.set_ylabel('Количество клиентов')
plt.title('Гистограмма распределения баллов кредитного скоринга');
```

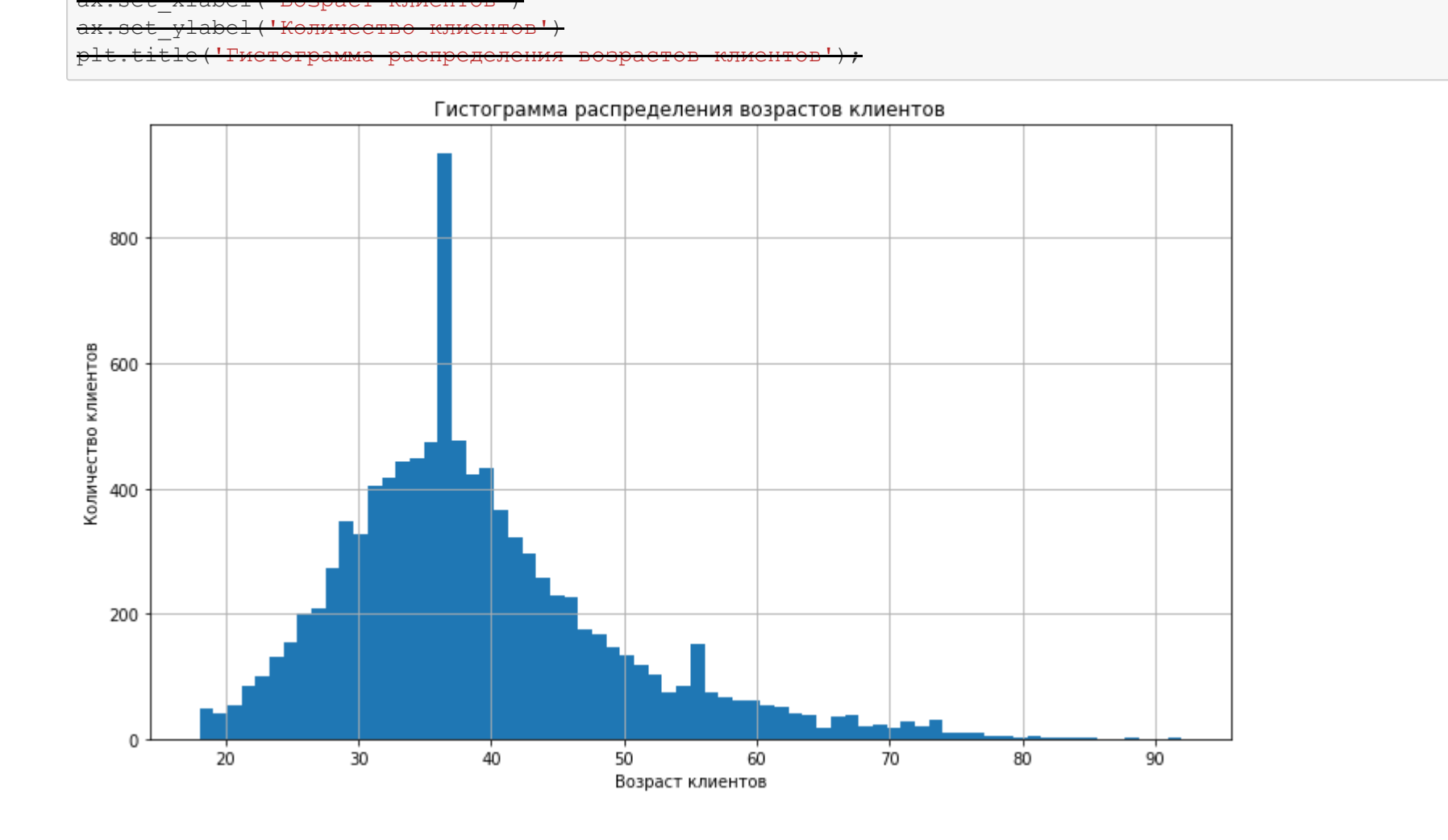


In [8]:
```
bank_dataset['score'].value_counts()
```

Out[8]:
```
850    233
678     63
655     54
705     53
667     53
      ...
419      1
417      1
373      1
365      1
401      1
Name: score, Length: 460, dtype: int64
```

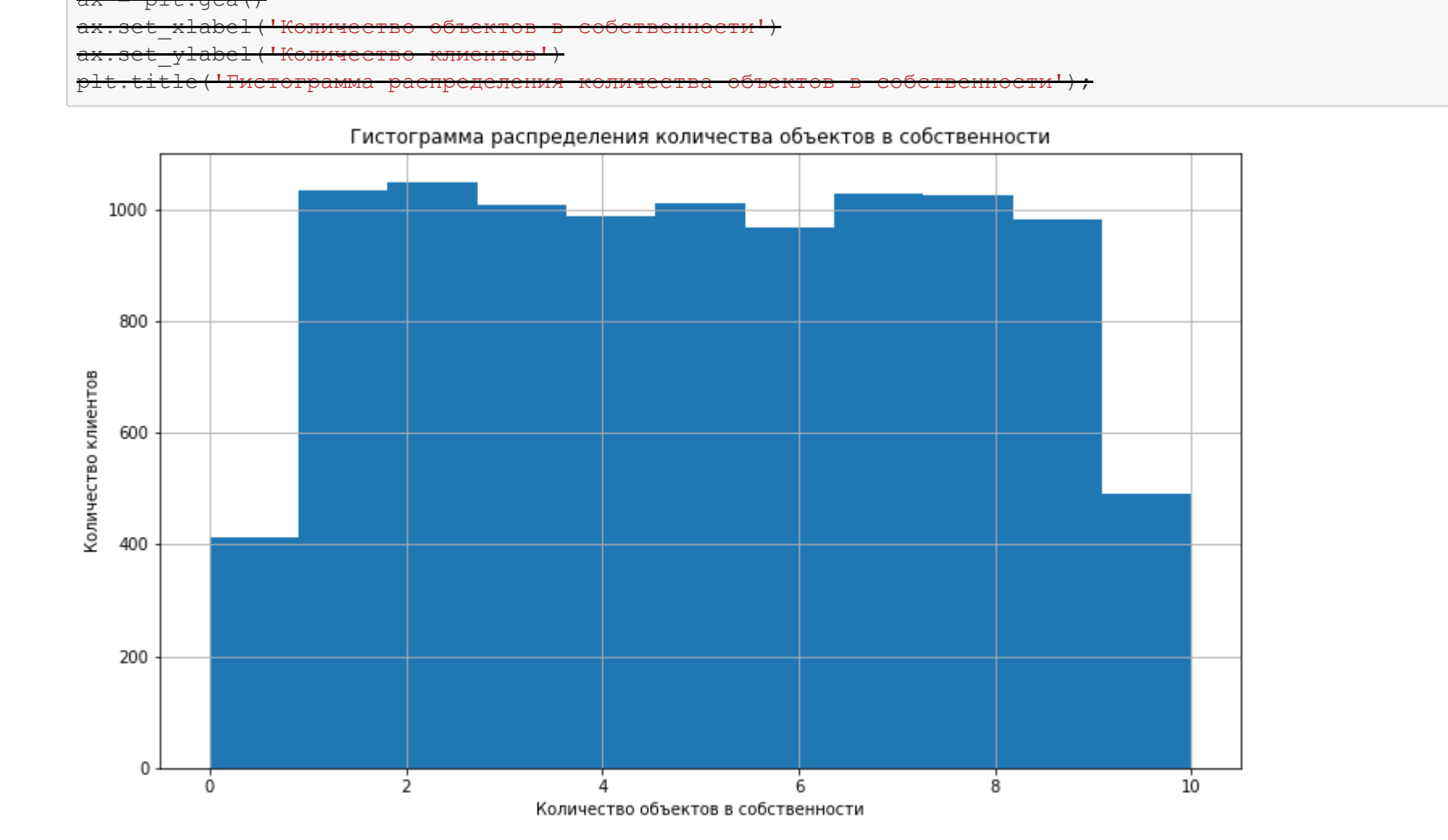
Вывод: у большинства клиентов 600-700 баллов кредитного скоринга. Баллы распределены нормально, однако присутствует аномальное количество клиентов (233) с максимальным количеством баллов (850). Пока оставим выброс без изменения.

In [9]:
```
bank_dataset['Age'].hist(figsize=(12, 7), bins=len(bank_dataset['Age'].unique()))
ax = plt.gca()
ax.set_xlabel('Возраст клиентов')
ax.set_ylabel('Количество клиентов')
plt.title('Гистограмма распределения возрастов клиентов');
```



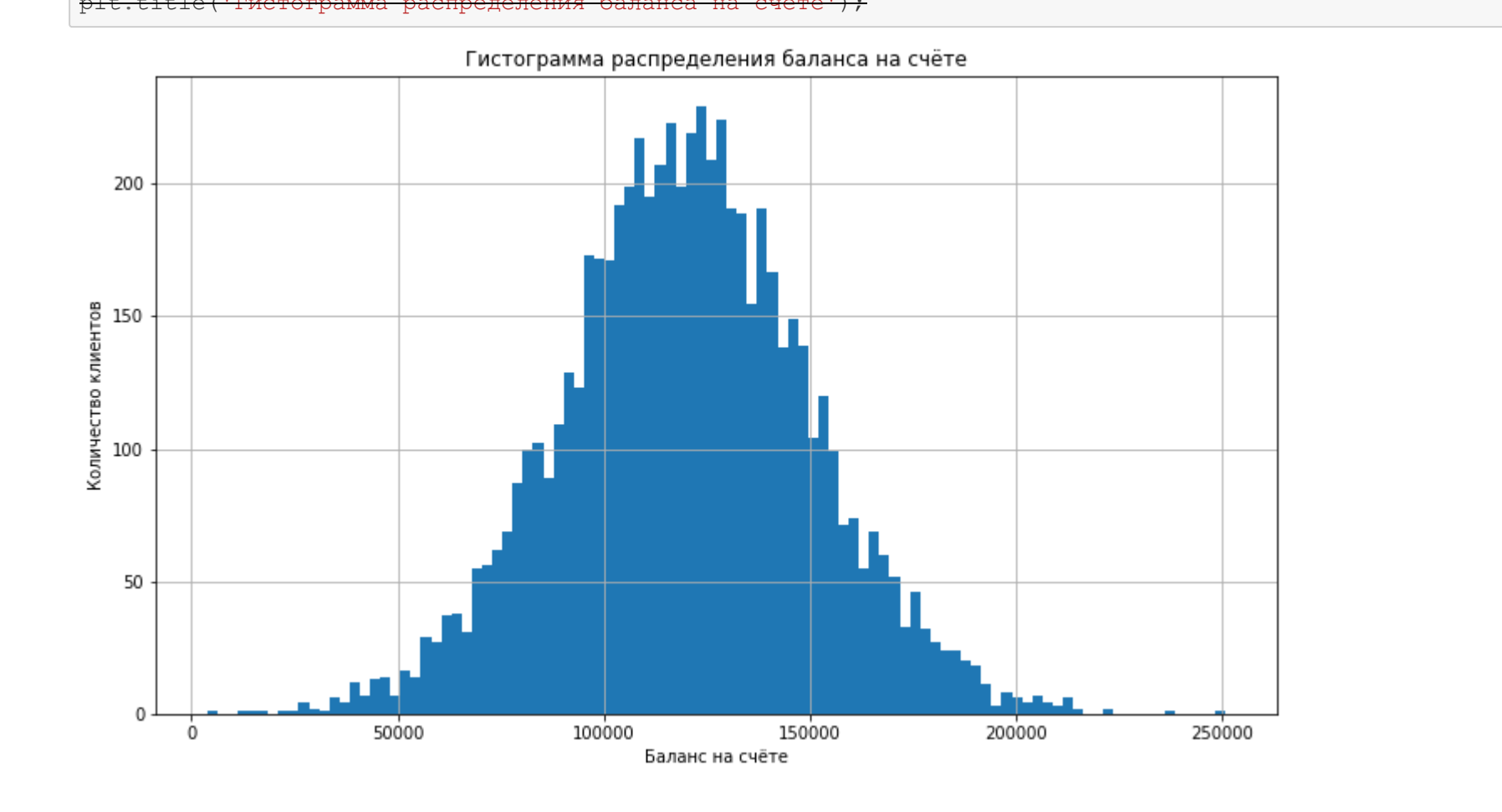
Вывод: большинство клиентов возрасте от 30 до 40, данные распределены нормально.

In [10]:
```
bank_dataset['Objects'].hist(figsize=(12, 7), bins=11)
ax = plt.gca()
ax.set_xlabel('Количество объектов в собственности')
ax.set_ylabel('Количество клиентов')
plt.title('Гистограмма распределения количества объектов в собственности');
```



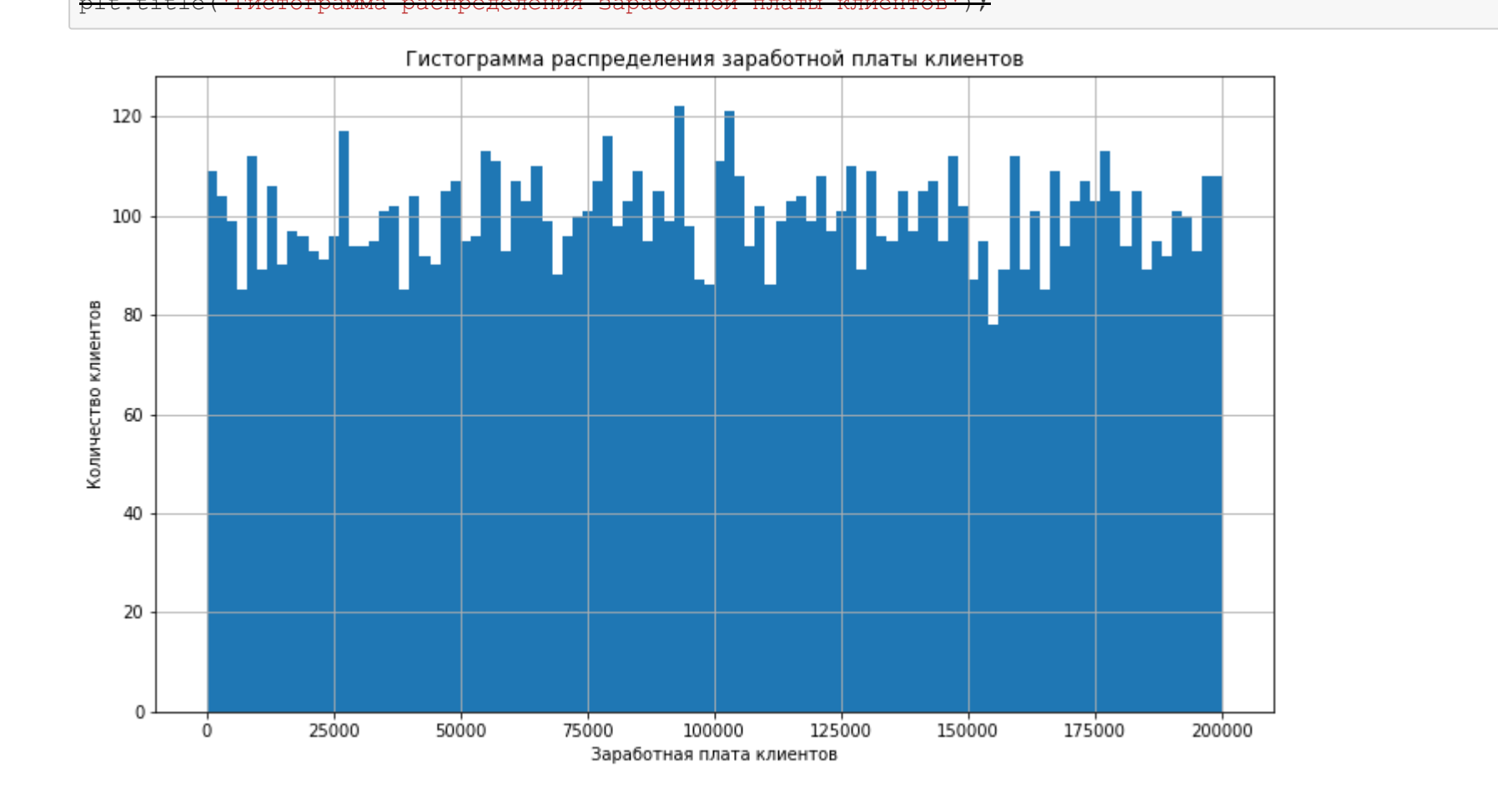
Вывод: количество объектов в собственности распределено нормально.

In [11]:
```
bank_dataset[bank_dataset['Balance'] != 0]['Balance'].hist(figsize=(12, 7), bins=100)
ax = plt.gca()
ax.set_xlabel('Баланс на счёте')
ax.set_ylabel('Количество клиентов')
plt.title('Гистограмма распределения баланса на счёте');
```



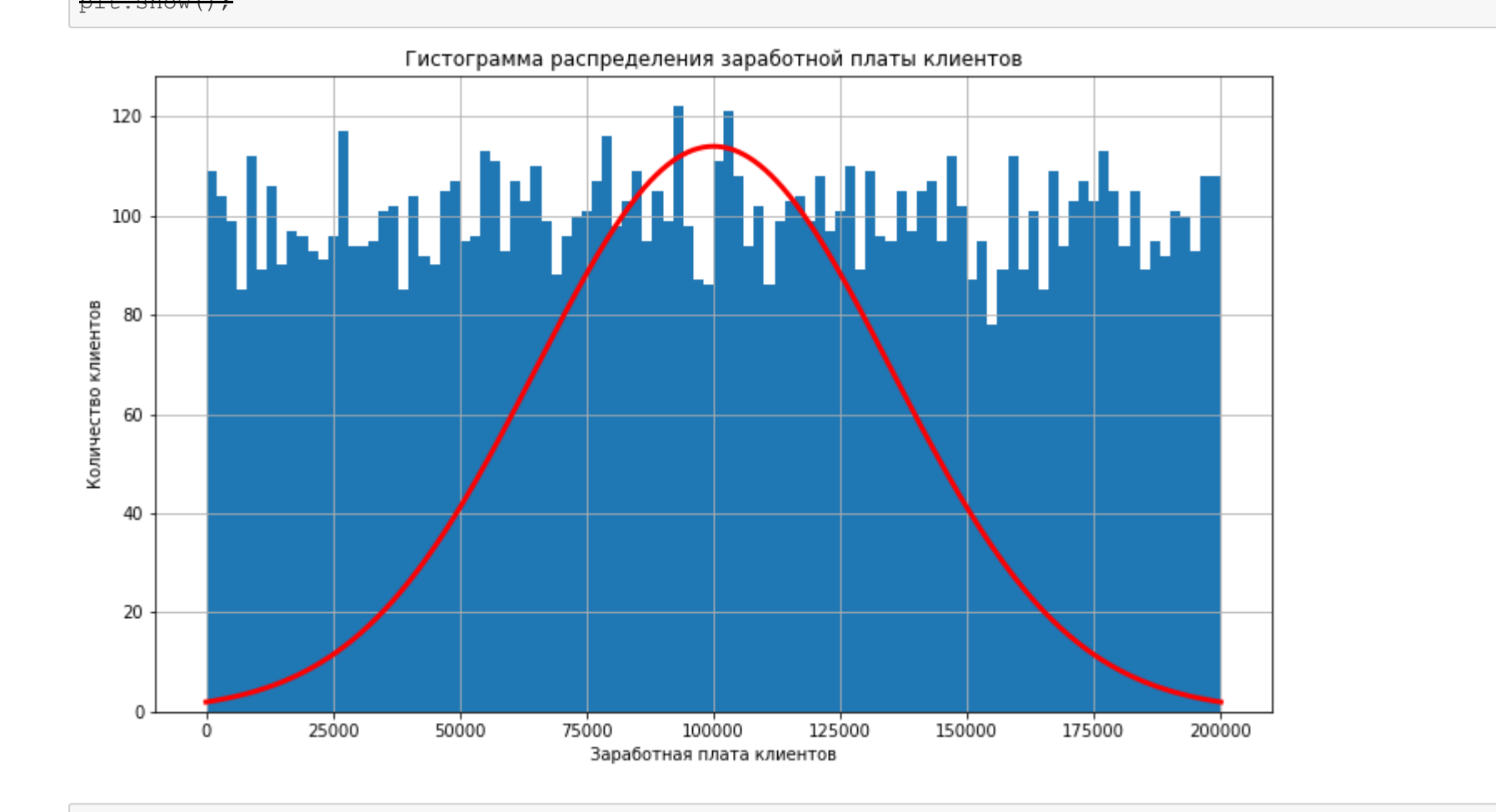
Вывод: у большинства клиентов от 100 до 150 тыс. на счёте, баланс на счёте распределён нормально.

In [12]:
```
bank_dataset['estimated_salary'].hist(figsize=(12, 7), bins=100)
ax = plt.gca()
ax.set_xlabel('Заработная плата клиентов')
ax.set_ylabel('Количество клиентов')
plt.title('Гистограмма распределения заработной платы клиентов');
```



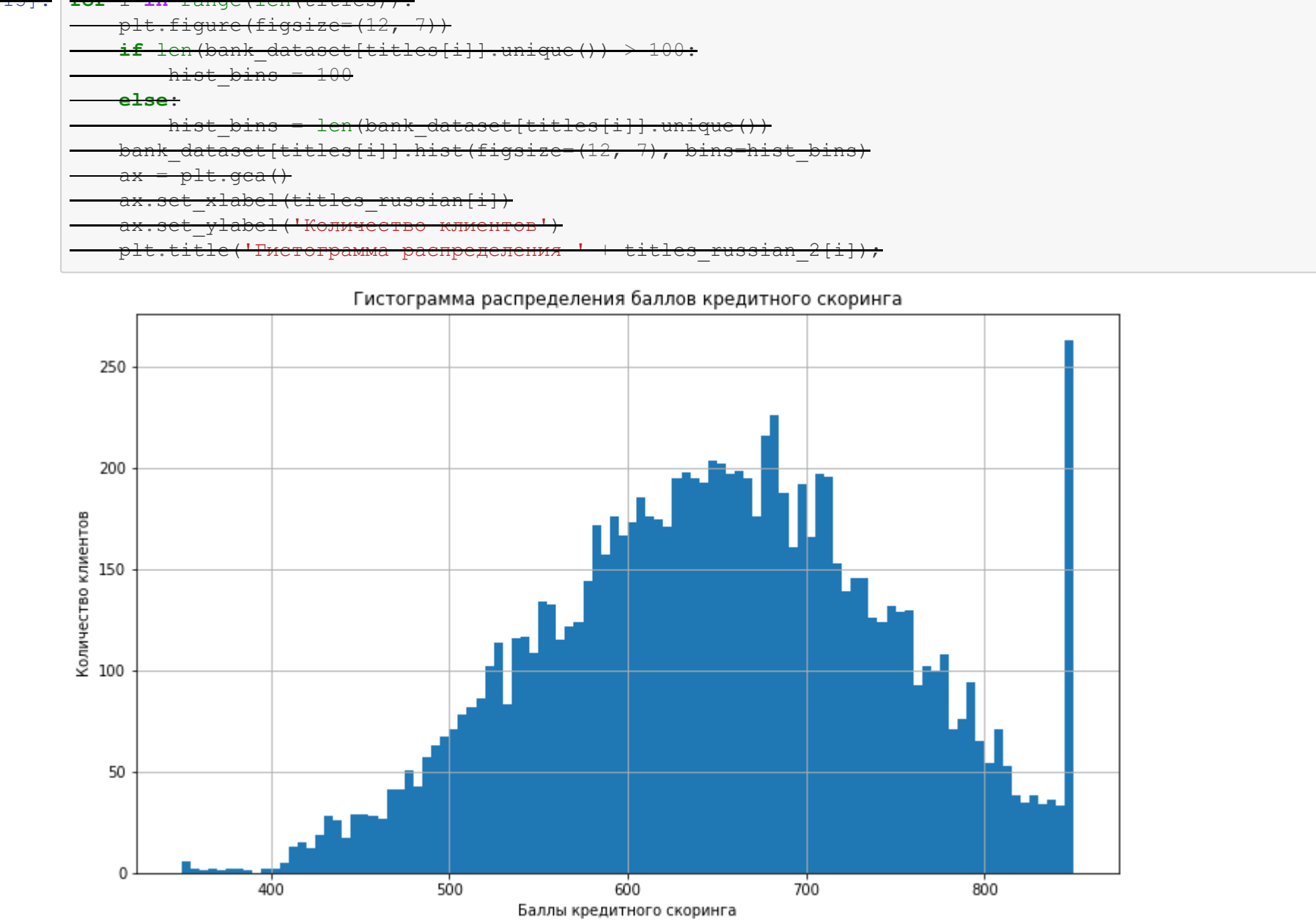
Вывод: средняя заработная плата клиента — 100000. Распределение заработных плат равномерное. Есть как аномально низкие (<6000), так и аномально высокие (> 196000). *(Почему? Предположение: в данных нарочно представлены все уровни доходов в примерно равном количестве.)*

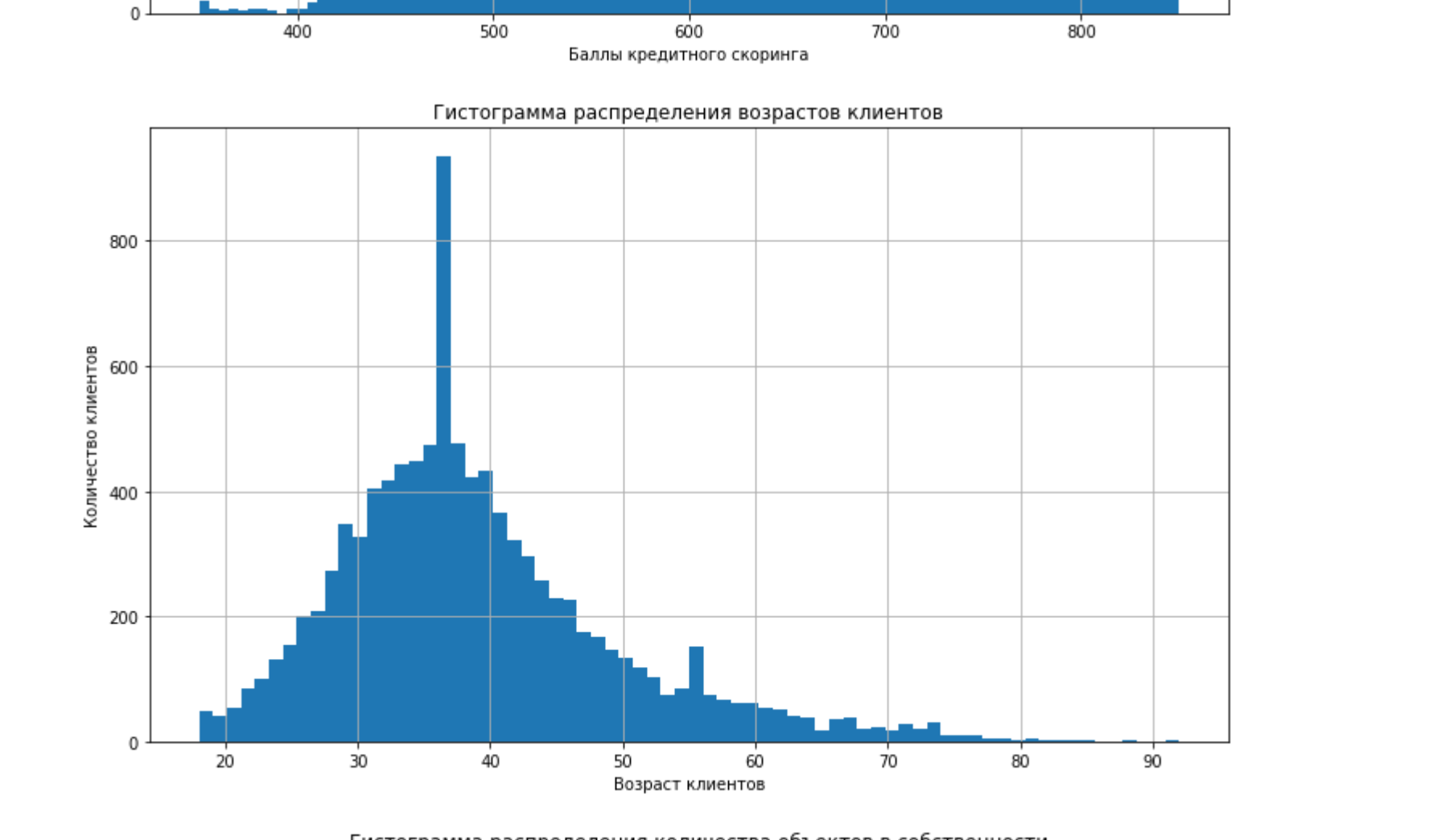
In [13]:
```
from scipy.stats import norm
bank_dataset['estimated_salary'].hist(figsize=(12, 7), bins=100)
ax = plt.gca()
ax.set_xlabel('Заработная плата клиентов')
ax.set_ylabel('Количество клиентов')
plt.title('Гистограмма распределения заработной платы клиентов')
x_axis = np.arange(0, 200000)
plt.plot(x_axis, 1e7*norm.pdf(x_axis, 100000, 35000), linewidth=3, color='r')
plt.show();
```

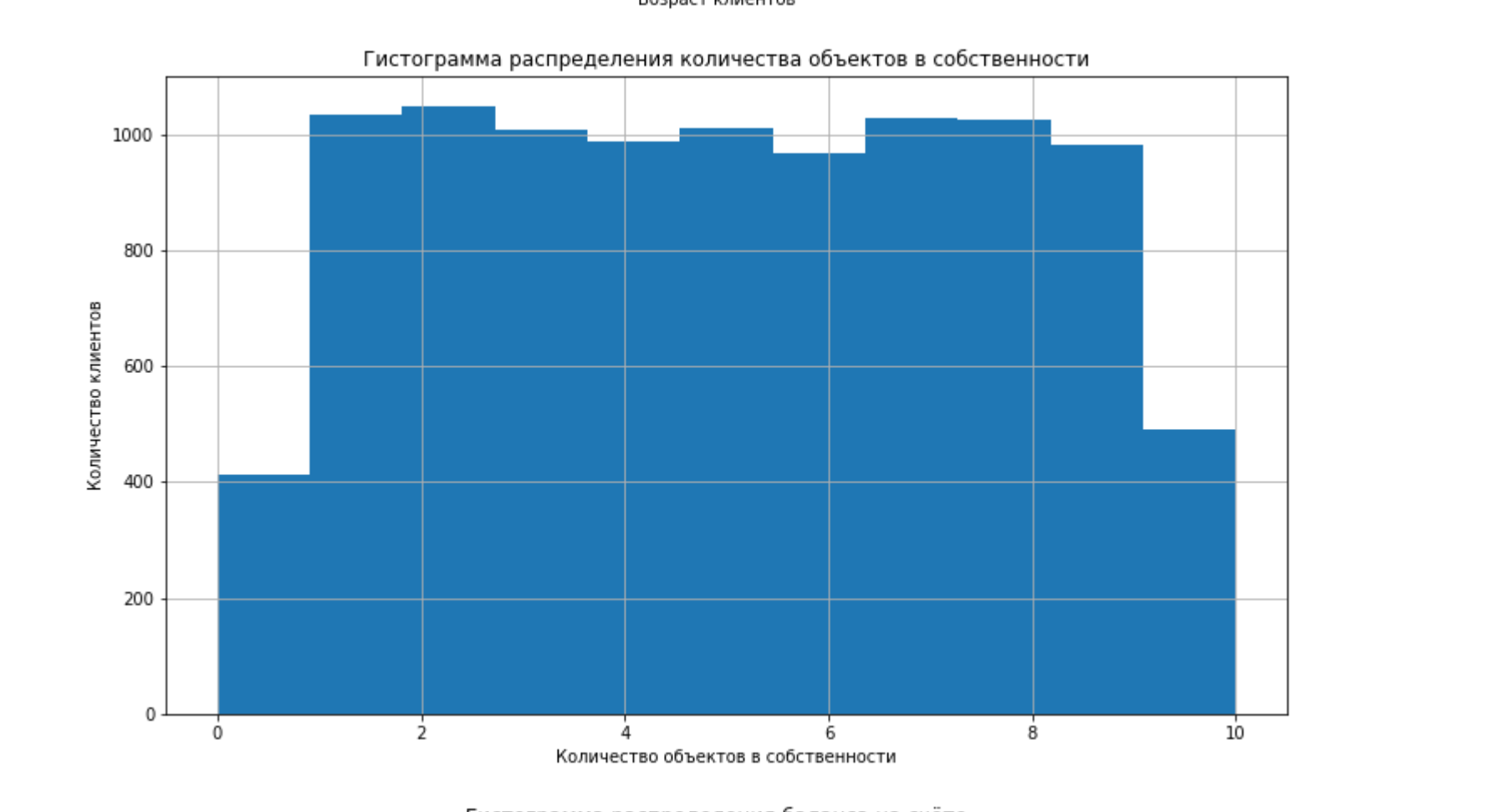


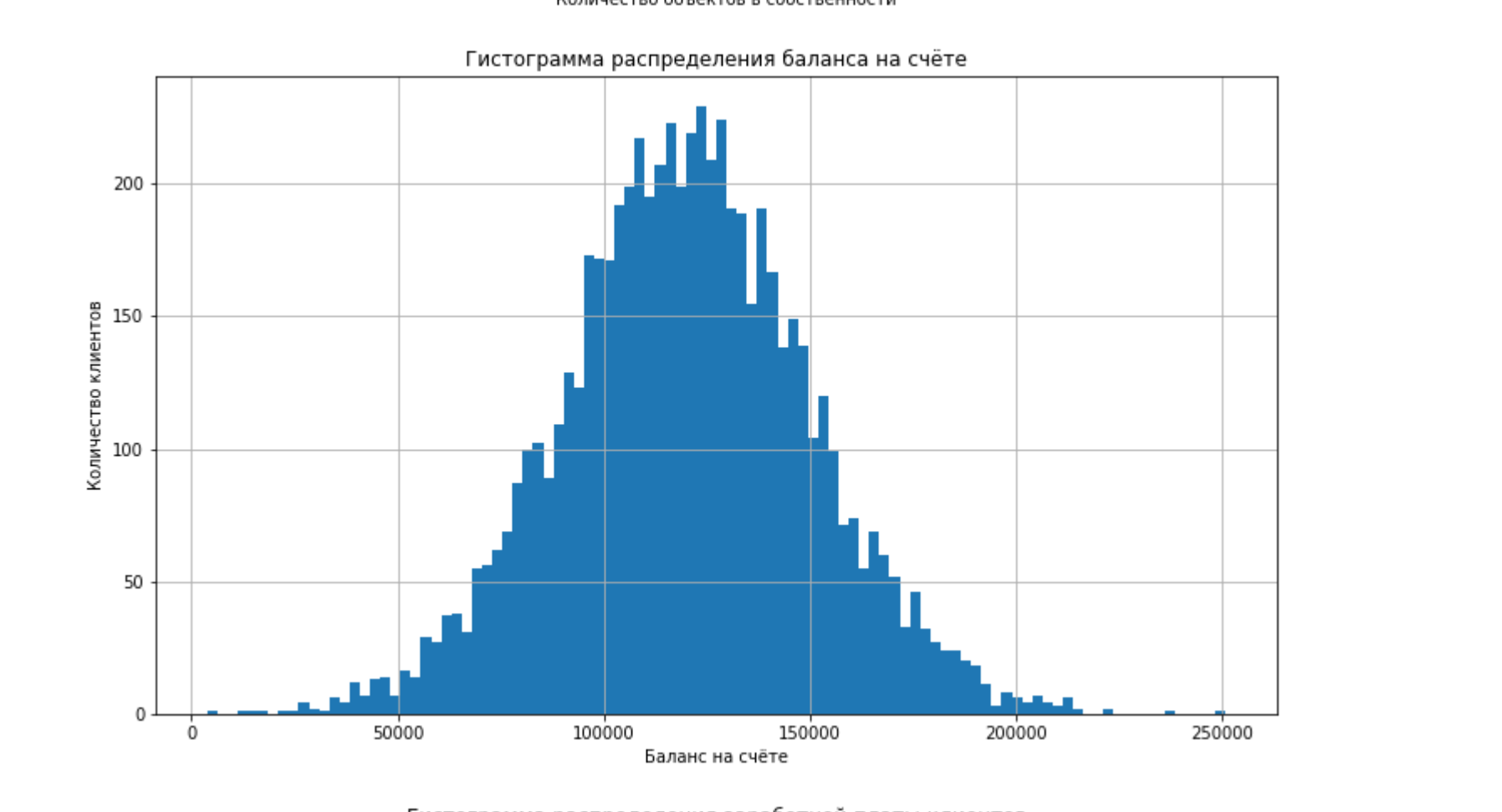
In [14]:
```
titles = ['score', 'Age', 'Objects', 'Balance', 'estimated_salary']
titles_russian = ['Баллы кредитного скоринга', \
                  'Возраст клиентов', \
                  'Количество объектов в собственности', \
                  'Баланс на счёте', \
                  'Заработная плата клиентов']
titles_russian_2 = ['баллов кредитного скоринга', \
                    'возрастов клиентов', \
                    'количества объектов в собственности', \
                    'баланса на счёте', \
                    'заработной платы клиентов']
```

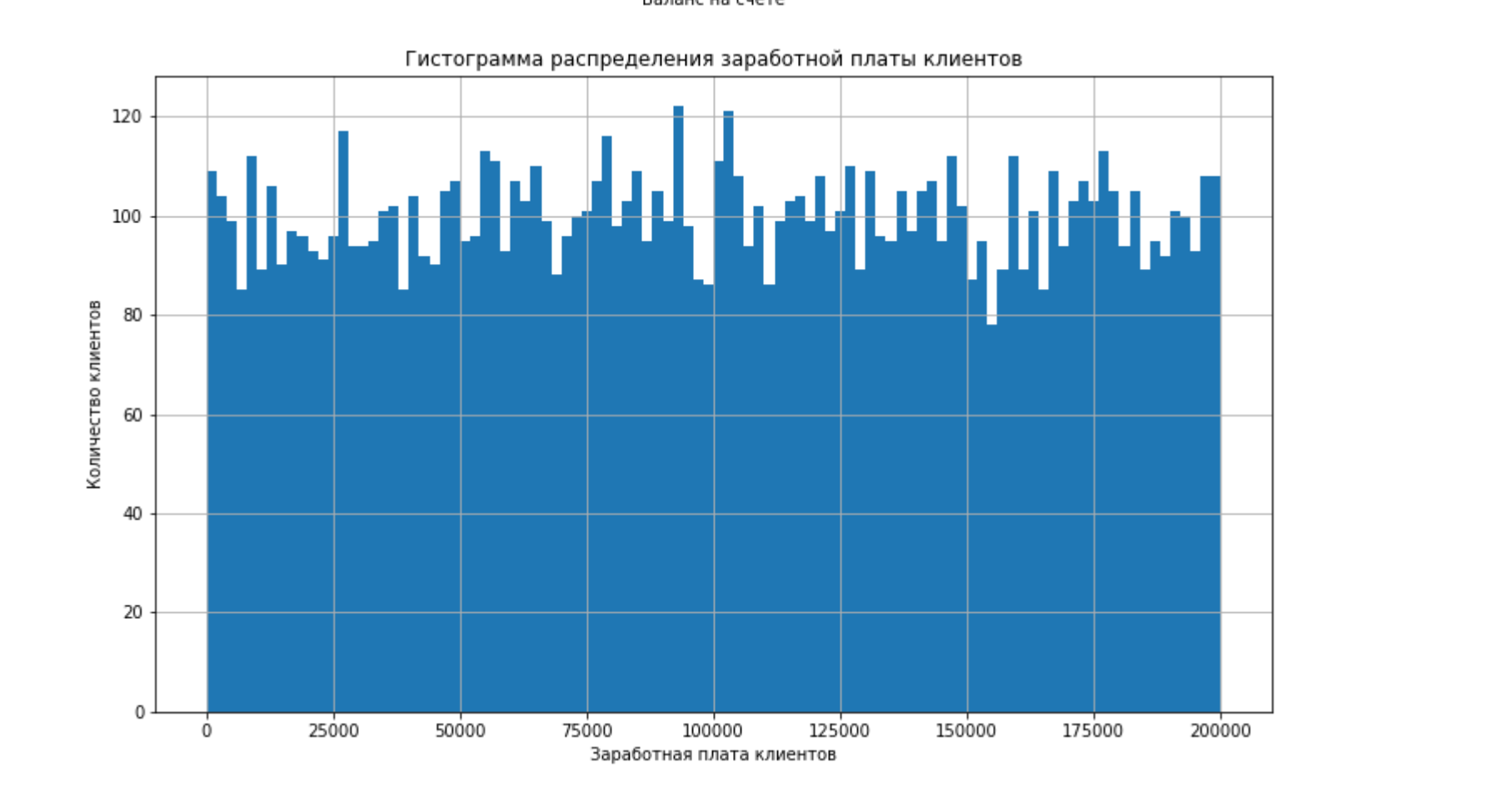
In [15]:
```
for i in range(len(titles)):
    plt.figure(figsize=(12, 7))
    if len(bank_dataset[titles[i]].unique()) > 100:
        hist_bins = 100
    else:
        hist_bins = len(bank_dataset[titles[i]].unique())
    bank_dataset[titles[i]].hist(figsize=(12, 7), bins=hist_bins)
    ax = plt.gca()
    ax.set_xlabel(titles_russian[i])
    ax.set_ylabel('Количество клиентов')
    plt.title('Гистограмма распределения ' + titles_russian_2[i]);
```





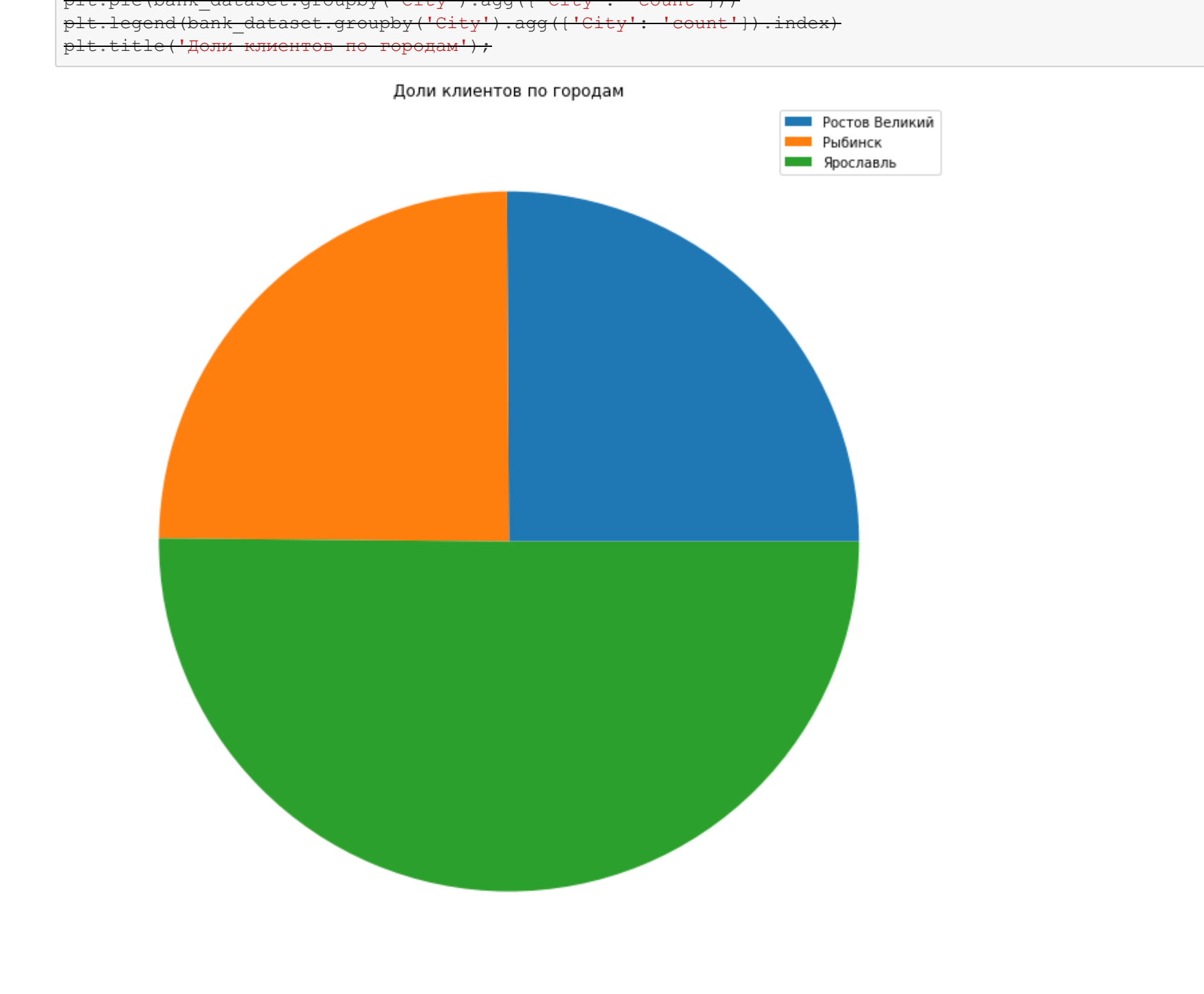






Оценим, как распределены клиенты по городам.

In [16]:
```
plt.figure(figsize=(12, 12))
plt.pie(bank_dataset.groupby('City').agg({'City': 'count'}))
plt.legend(bank_dataset.groupby('City').agg({'City': 'count'}).index)
plt.title('Доли клиентов по городам');
```



Вывод: клиентов в Ярославле в два раза больше, чем в Ростове Великом и Рыбинске.

Построим матрицу корреляций и отобразим её при помощи тепловой карты.

In [17]:
```
plt.figure(figsize = (14,14))
plt.title('Тепловая карта матрицы корреляций признаков')
sns.heatmap(bank_dataset.corr(), cmap='Blues', linewidths=1, linecolor='gray', square = True, annot = True);
```

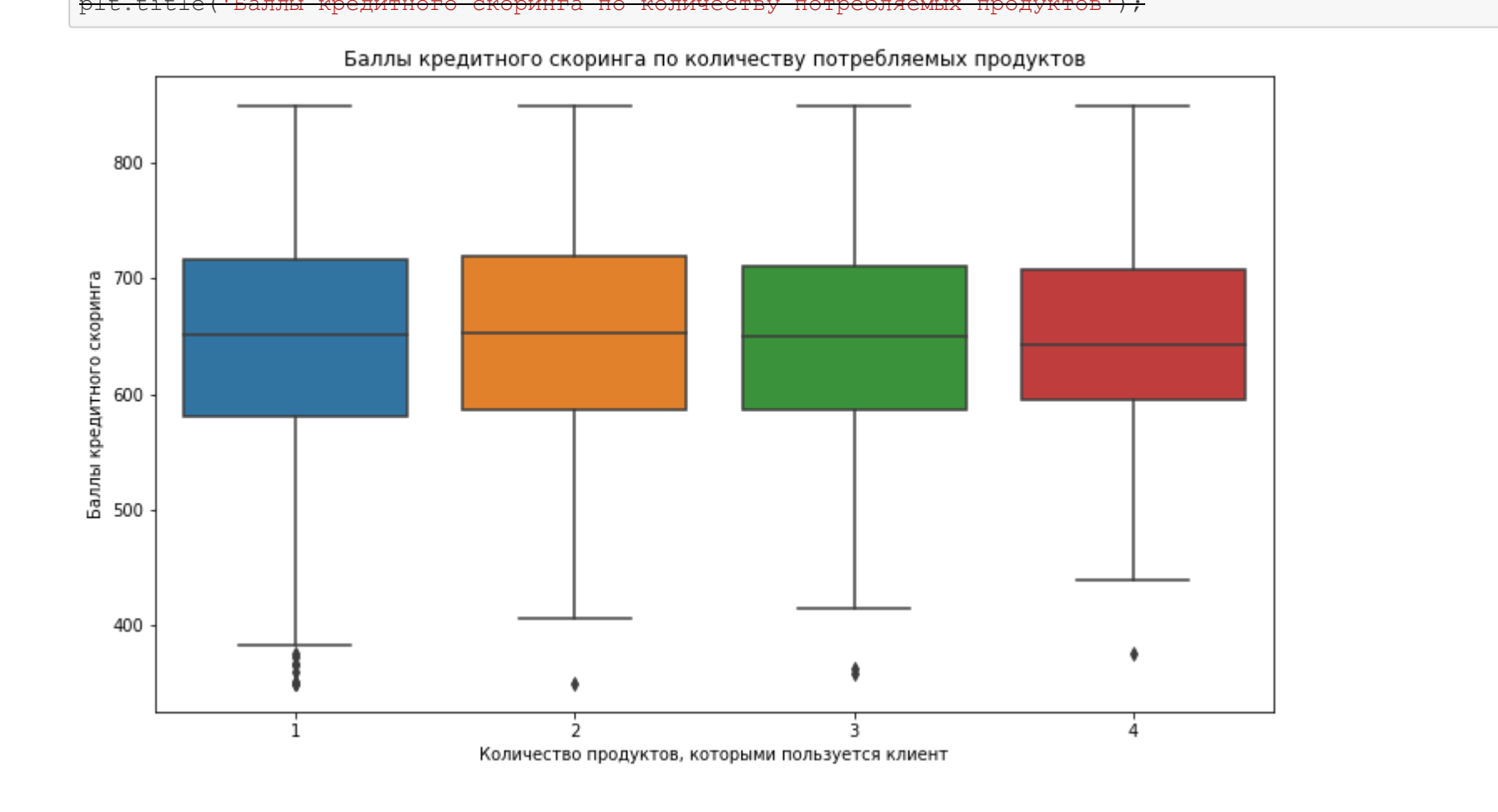
Вывод: сильно коррелирующие признаки отсутствуют. Самые сильные связи наблюдаются между оттоком и возрастом, количеством используемых продуктов и балансом на счёте.

К оглавлению

Шаг 2. Сегментация пользователей на основе данных о количестве потребляемых продуктов

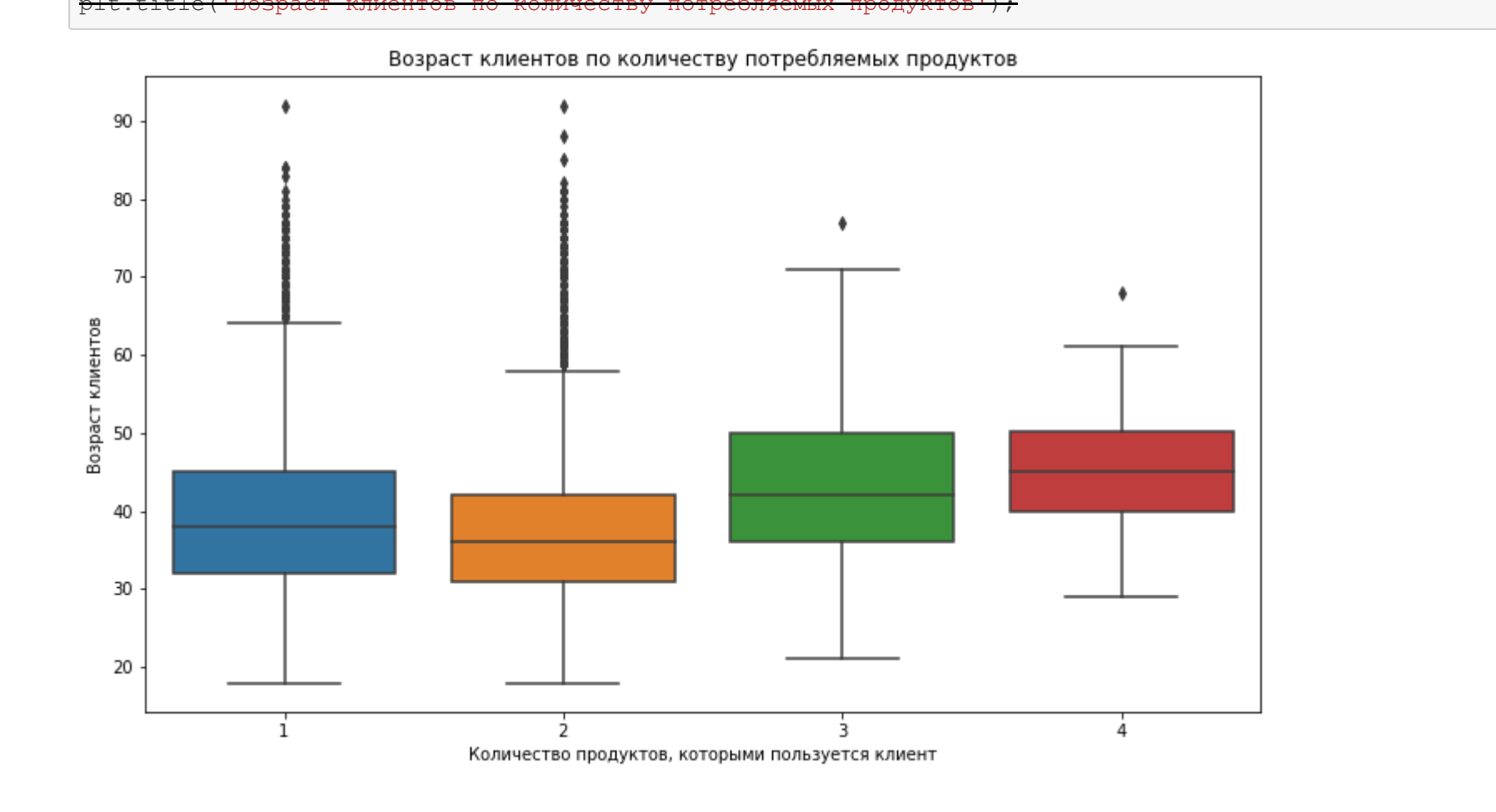
Сгруппируем пользователей по количеству потребляемых продуктов и сравним параметры каждой группы методом `boxplot()`.

In [18]:
```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Products', y='score', data=bank_dataset)
ax = plt.gca()
ax.set_xlabel('Количество продуктов, которыми пользуется клиент')
ax.set_ylabel('Баллы кредитного скоринга')
plt.title('Баллы кредитного скоринга по количеству потребляемых продуктов');
```



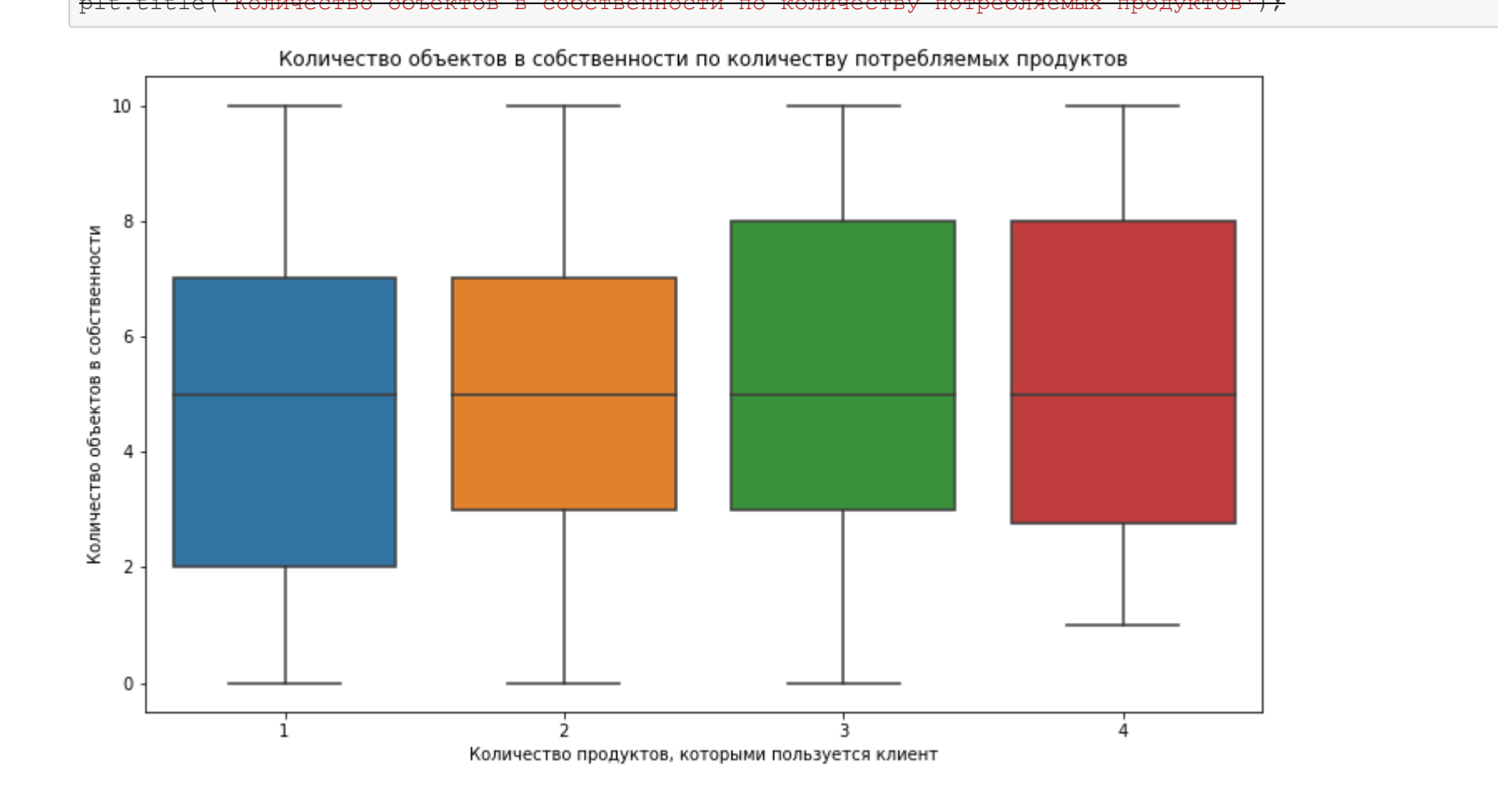
Вывод: количество продуктов, которыми пользуется клиент, не зависит от баллов кредитного скоринга.

In [19]:
```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Products', y='Age', data=bank_dataset)
ax = plt.gca()
ax.set_xlabel('Количество продуктов, которыми пользуется клиент')
ax.set_ylabel('Возраст клиентов')
plt.title('Возраст клиентов по количеству потребляемых продуктов');
```



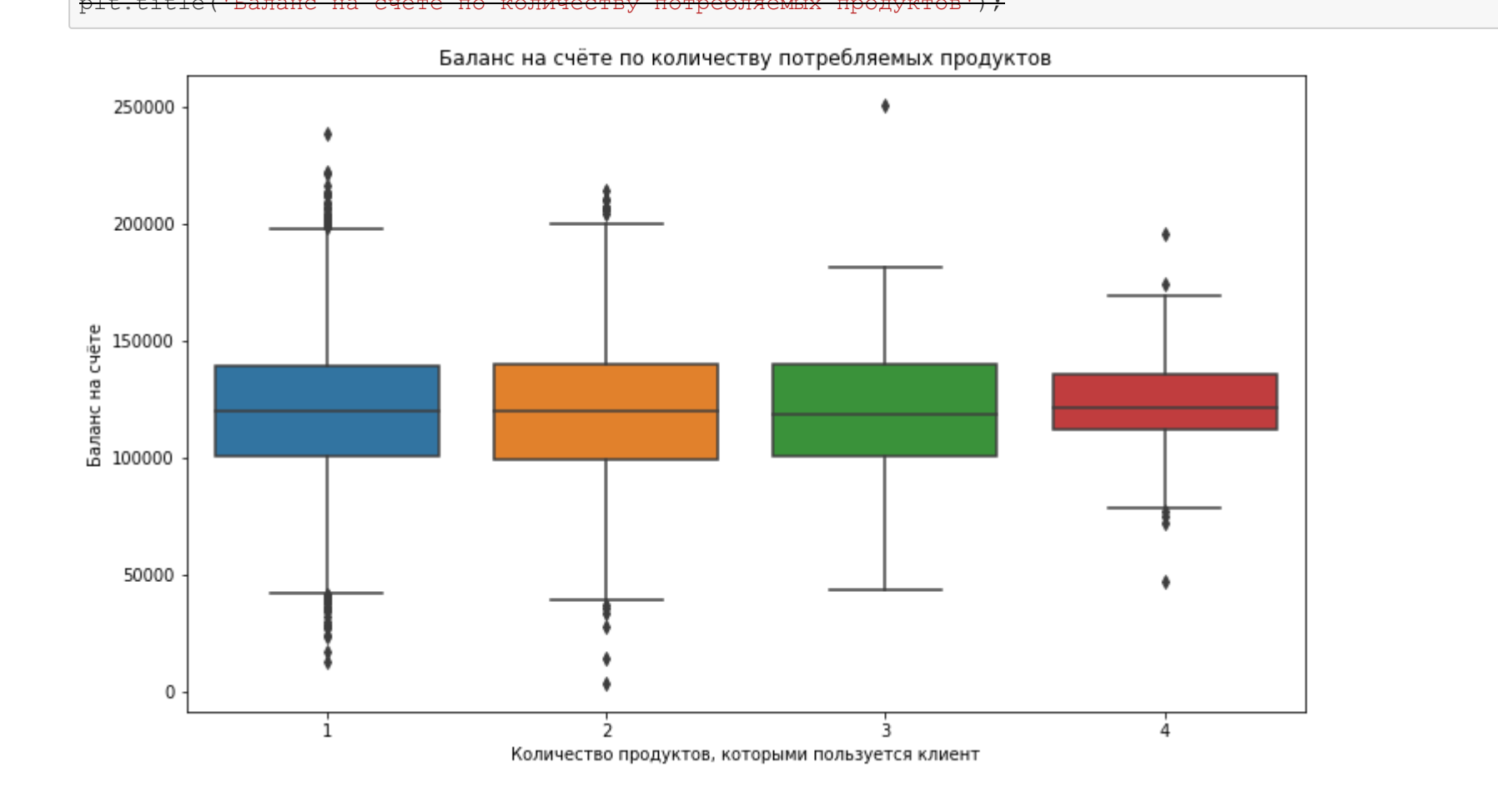
Вывод: с возрастом клиенты используют большее число продуктов, однако пенсионеры, как правило, пользуются одним-двумя продуктами.

In [20]:
```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Products', y='Objects', data=bank_dataset)
ax = plt.gca()
ax.set_xlabel('Количество продуктов, которыми пользуется клиент')
ax.set_ylabel('Количество объектов в собственности')
plt.title('Количество объектов в собственности по количеству потребляемых продуктов');
```



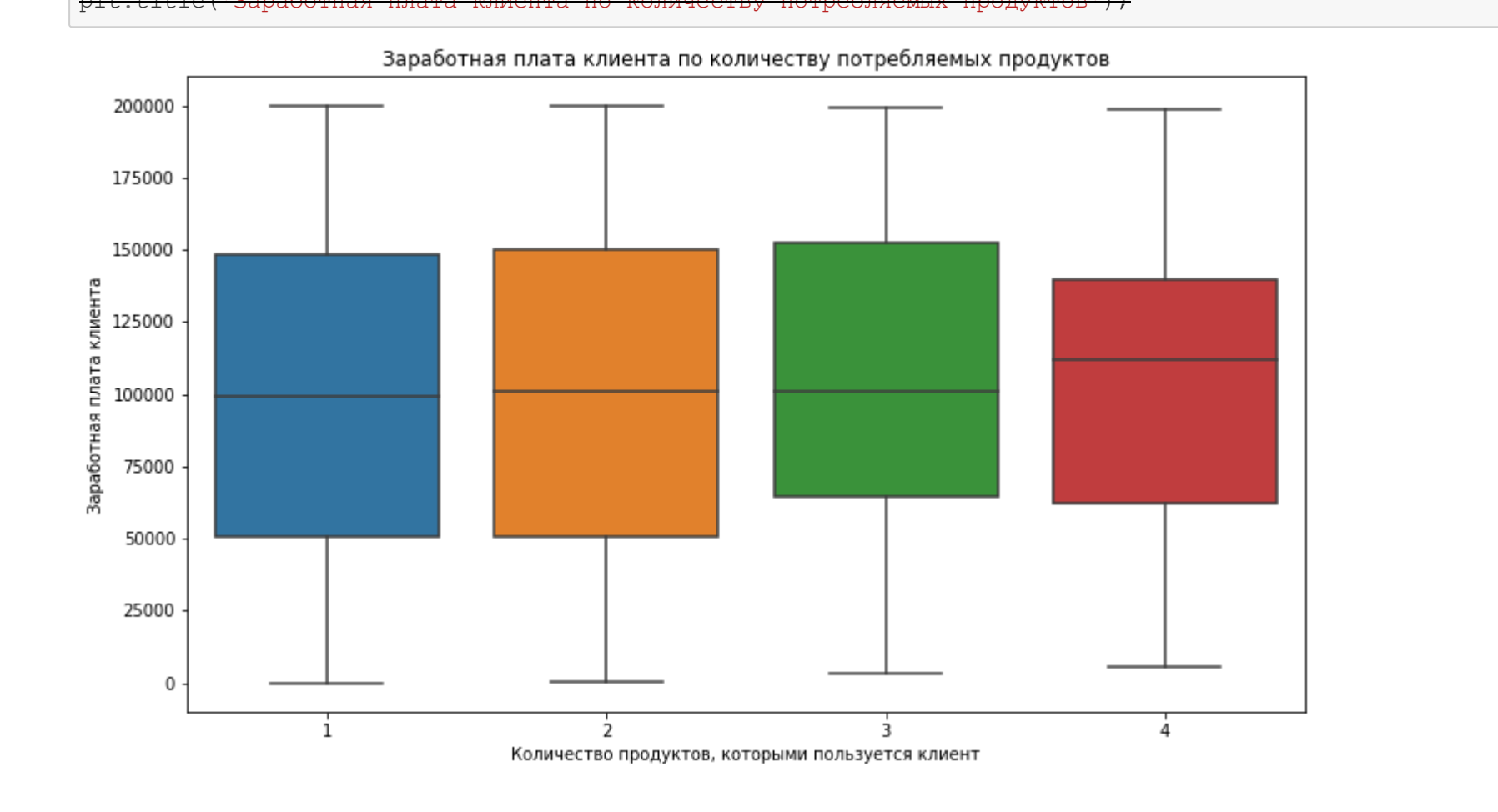
Вывод: у клиентов, потребляющих большее количество продуктов, как правило, больше объектов в собственности.

In [21]:
```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Products', y='Balance', data=bank_dataset)
ax = plt.gca()
ax.set_xlabel('Количество продуктов, которыми пользуется клиент')
ax.set_ylabel('Баланс на счёте')
plt.title('Баланс на счёте по количеству потребляемых продуктов');
```



Вывод: баланс выше у клиентов, пользующихся либо всего одним продуктом, либо сразу многими.

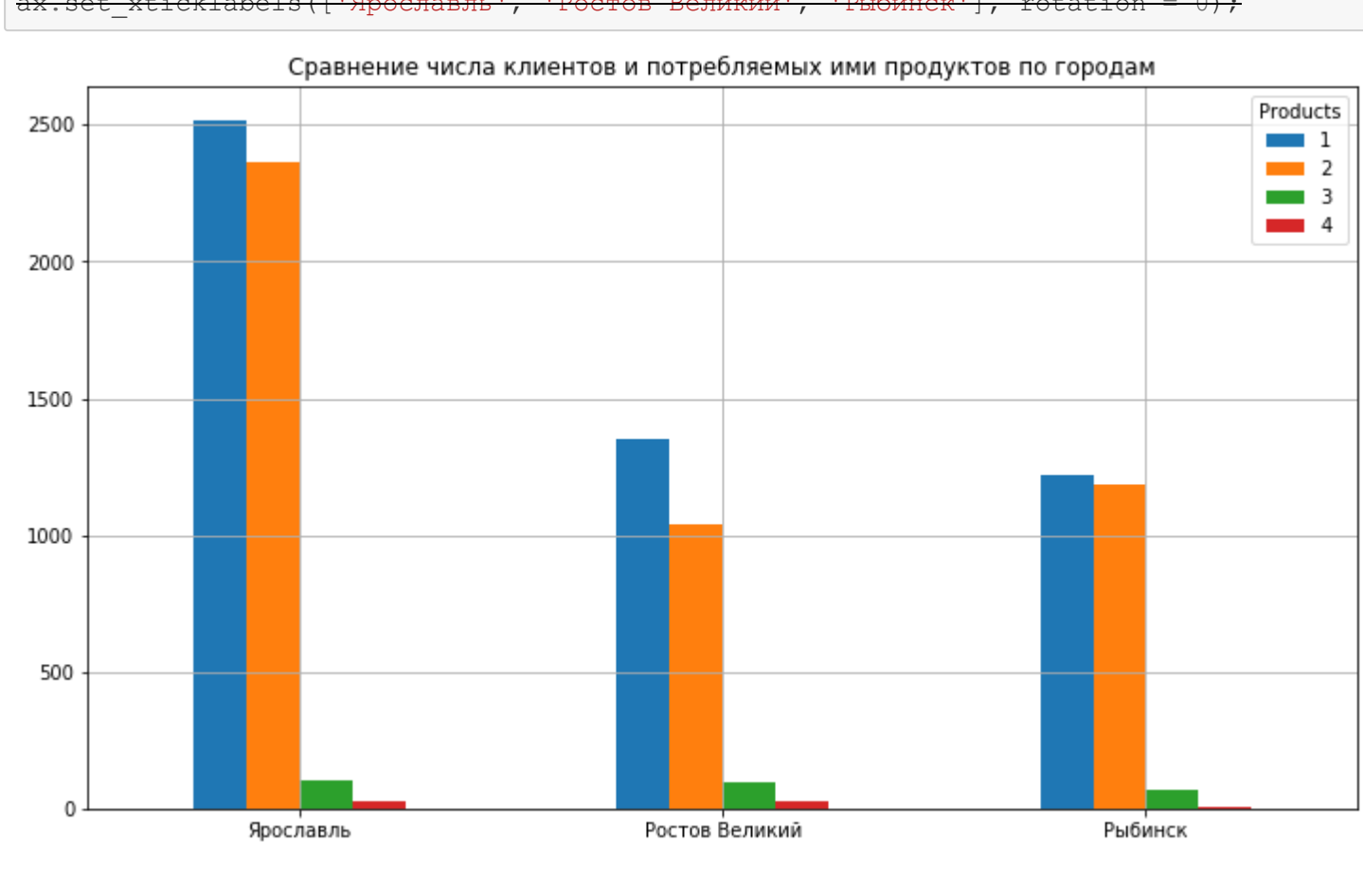
In [22]:
```
plt.figure(figsize=(12, 7))
sns.boxplot(x='Products', y='estimated_salary', data=bank_dataset)
ax = plt.gca()
ax.set_xlabel('Количество продуктов, которыми пользуется клиент')
ax.set_ylabel('Заработная плата клиента')
plt.title('Заработная плата клиента по количеству потребляемых продуктов');
```



Вывод: у клиентов, потребляющих большее количество продуктов, зарплата незначительно выше.

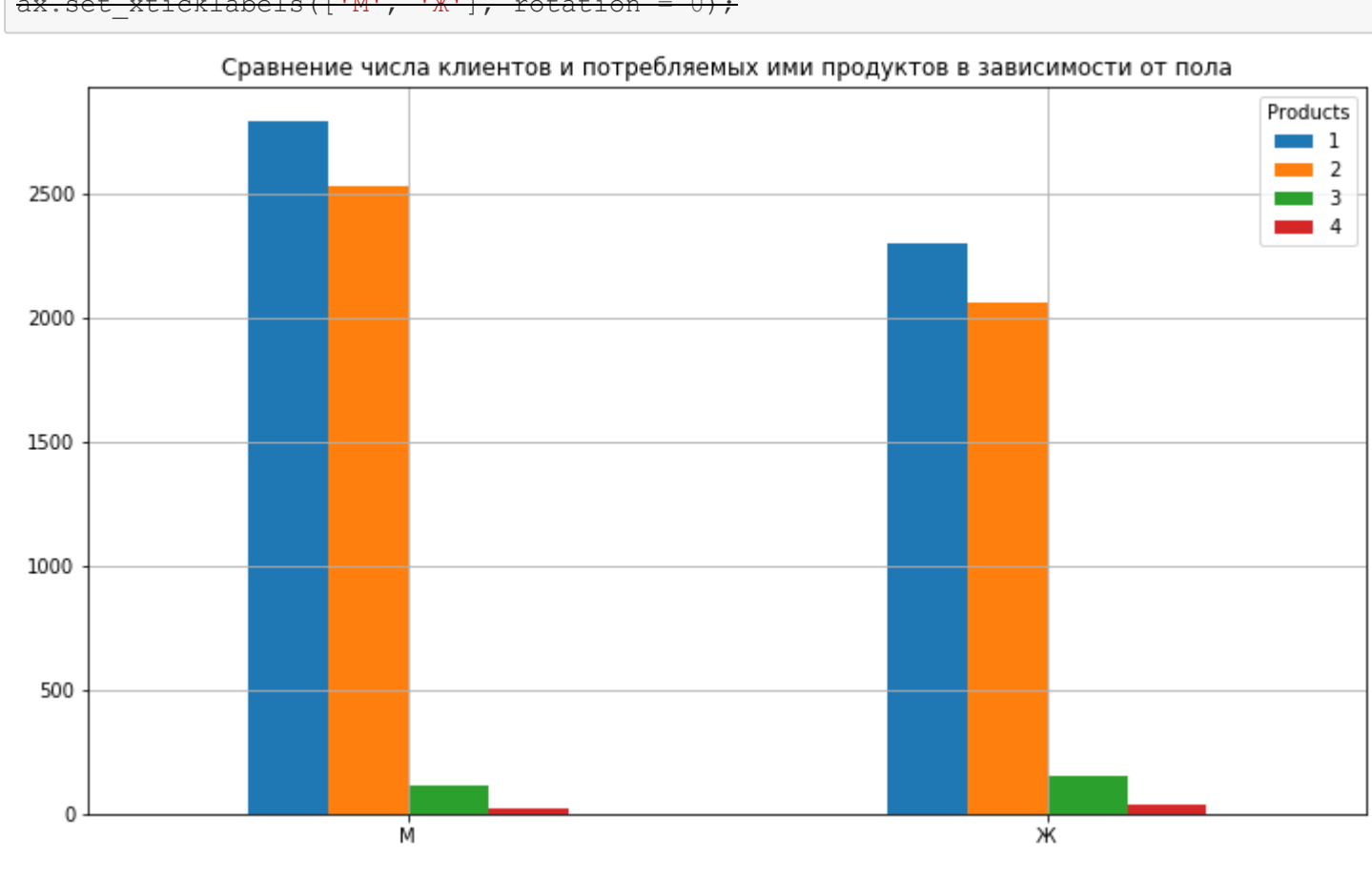
Для сравнения логических и дискретных переменных построим столбчатые диаграммы.

In [23]:
```python
pd.concat([bank_dataset[bank_dataset['City'] == 'Ярославль'].groupby('Products').agg({'Products': 'count'}), \
           bank_dataset[bank_dataset['City'] == 'Ростов Великий'].groupby('Products').agg({'Products': 'count'}), \
           bank_dataset[bank_dataset['City'] == 'Рыбинск'].groupby('Products').agg({'Products': 'count'})], axis=1).T.\
plot(kind='bar', figsize=(12, 7), grid=True, title='Сравнение числа клиентов и потребляемых ими продуктов по городам')
ax = plt.gca()
ax.set_xticklabels(['Ярославль', 'Ростов Великий', 'Рыбинск'], rotation = 0);
```



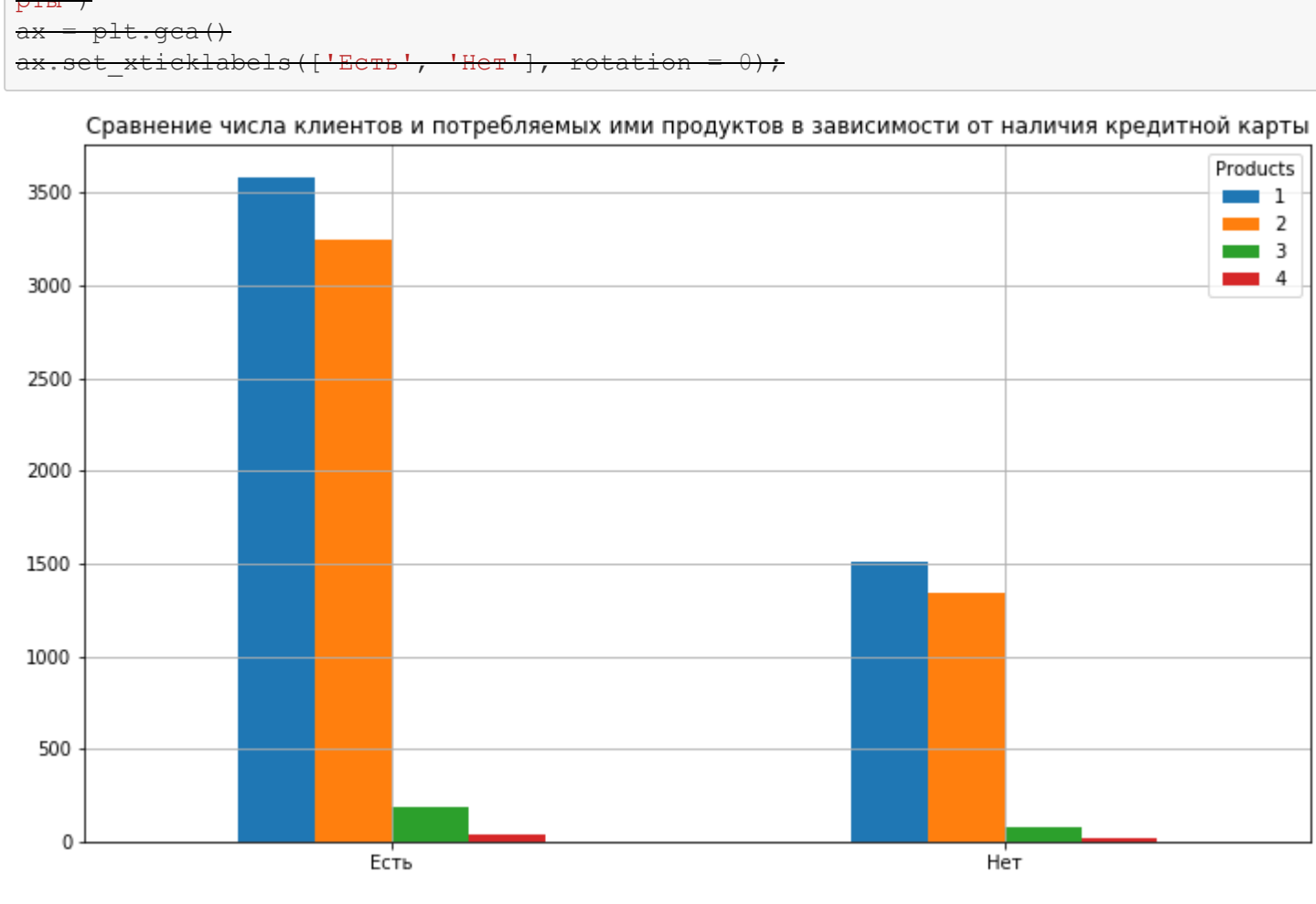
Вывод: относительное количество потребляемых продуктов внутри городов различается незначительно. По абсолютному количеству лидирует Ярославль (в два раза больше Ростова Великий и Рыбинска).

In [24]:
```python
pd.concat([bank_dataset[bank_dataset['Gender'] == 'М'].groupby('Products').agg({'Products': 'count'}), \
           bank_dataset[bank_dataset['Gender'] == 'Ж'].groupby('Products').agg({'Products': 'count'})], axis=1).T.\
plot(kind='bar', figsize=(12, 7), grid=True, \
     title='Сравнение числа клиентов и потребляемых ими продуктов в зависимости от пола')
ax = plt.gca()
ax.set_xticklabels(['М', 'Ж'], rotation = 0);
```



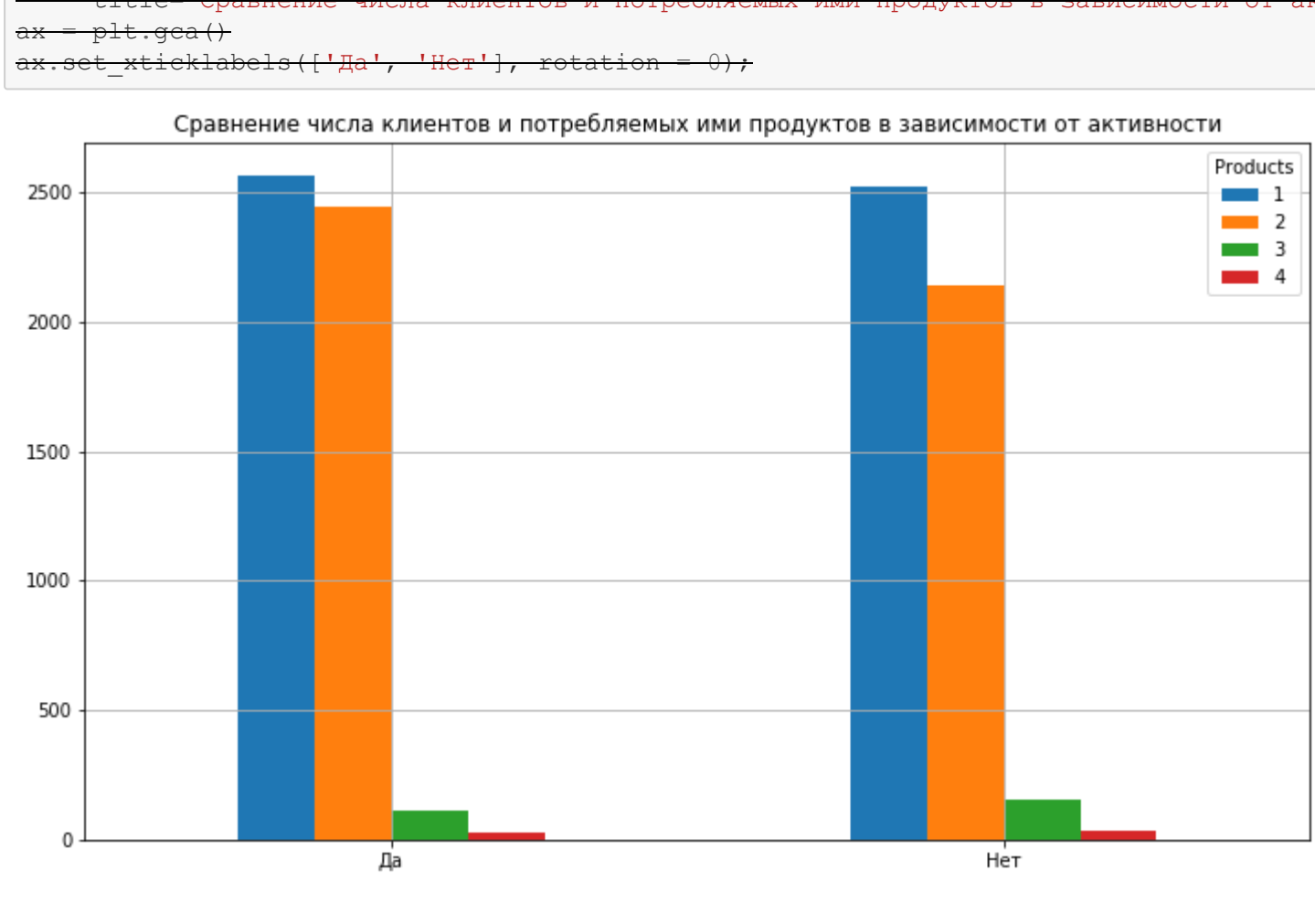
Вывод: количество потребляемых продуктов не зависит от пола. Общее число клиентов-мужчин незначительно превышает число клиентов-женщин.

In [25]:
```python
pd.concat([bank_dataset[bank_dataset['CreditCard'] == 1].groupby('Products').agg({'Products': 'count'}), \
           bank_dataset[bank_dataset['CreditCard'] == 0].groupby('Products').agg({'Products': 'count'})], axis=1).T.\
plot(kind='bar', figsize=(12, 7), grid=True, \
     title='Сравнение числа клиентов и потребляемых ими продуктов в зависимости от наличия кредитной карты')
ax = plt.gca()
ax.set_xticklabels(['Есть', 'Нет'], rotation = 0);
```



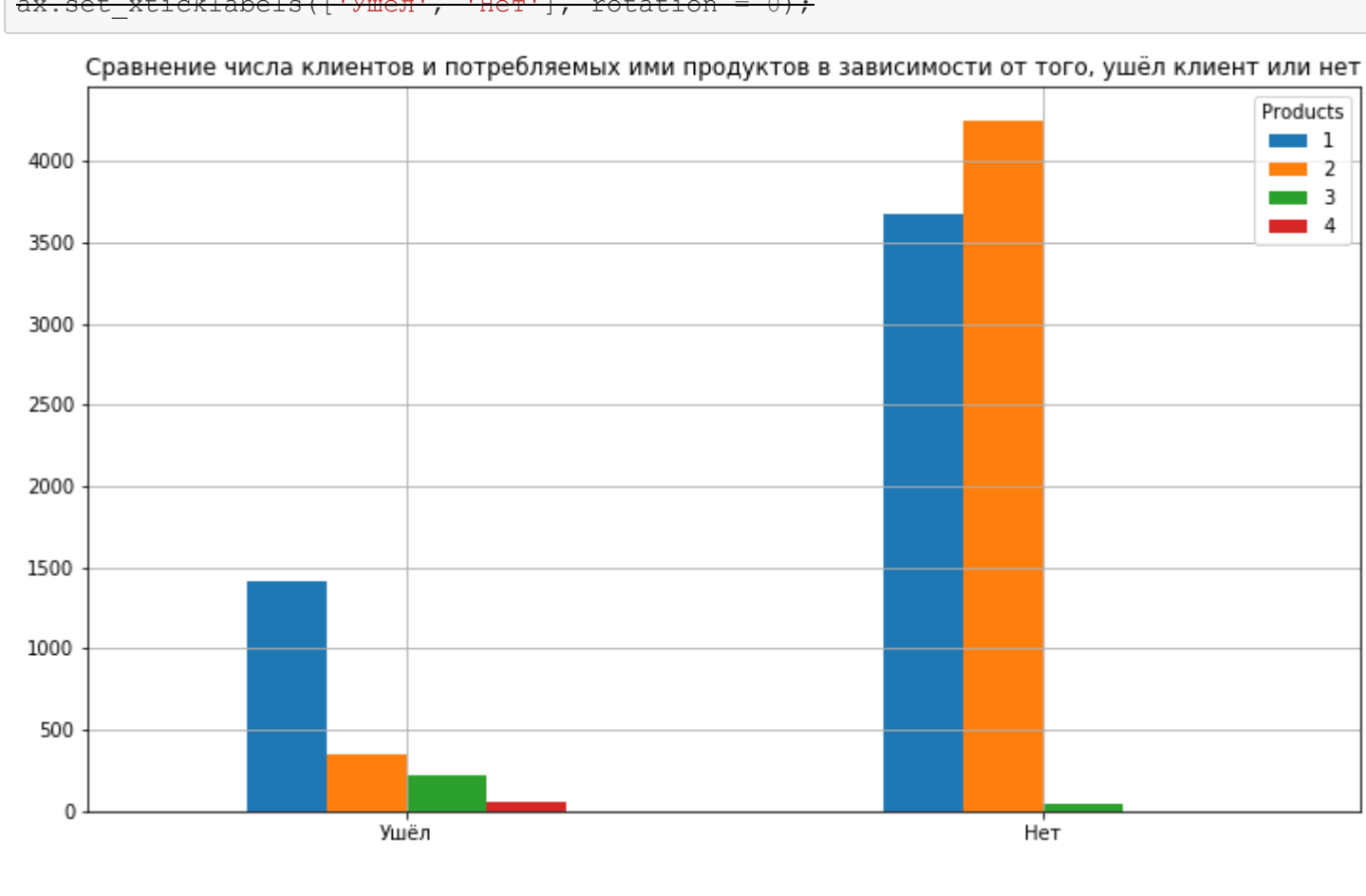
Вывод: количество потребляемых продуктов не зависит от наличия кредитной карты. Владельцев кредитной карты в два с лишним раза больше, чем клиентов без неё.

In [26]:
```python
pd.concat([bank_dataset[bank_dataset['Loyalty'] == 1].groupby('Products').agg({'Products': 'count'}), \
           bank_dataset[bank_dataset['Loyalty'] == 0].groupby('Products').agg({'Products': 'count'})], axis=1).T.\
plot(kind='bar', figsize=(12, 7), grid=True, \
     title='Сравнение числа клиентов и потребляемых ими продуктов в зависимости от активности')
ax = plt.gca()
ax.set_xticklabels(['Да', 'Нет'], rotation = 0);
```



Вывод: количество потребляемых продуктов не зависит от активности клиента.

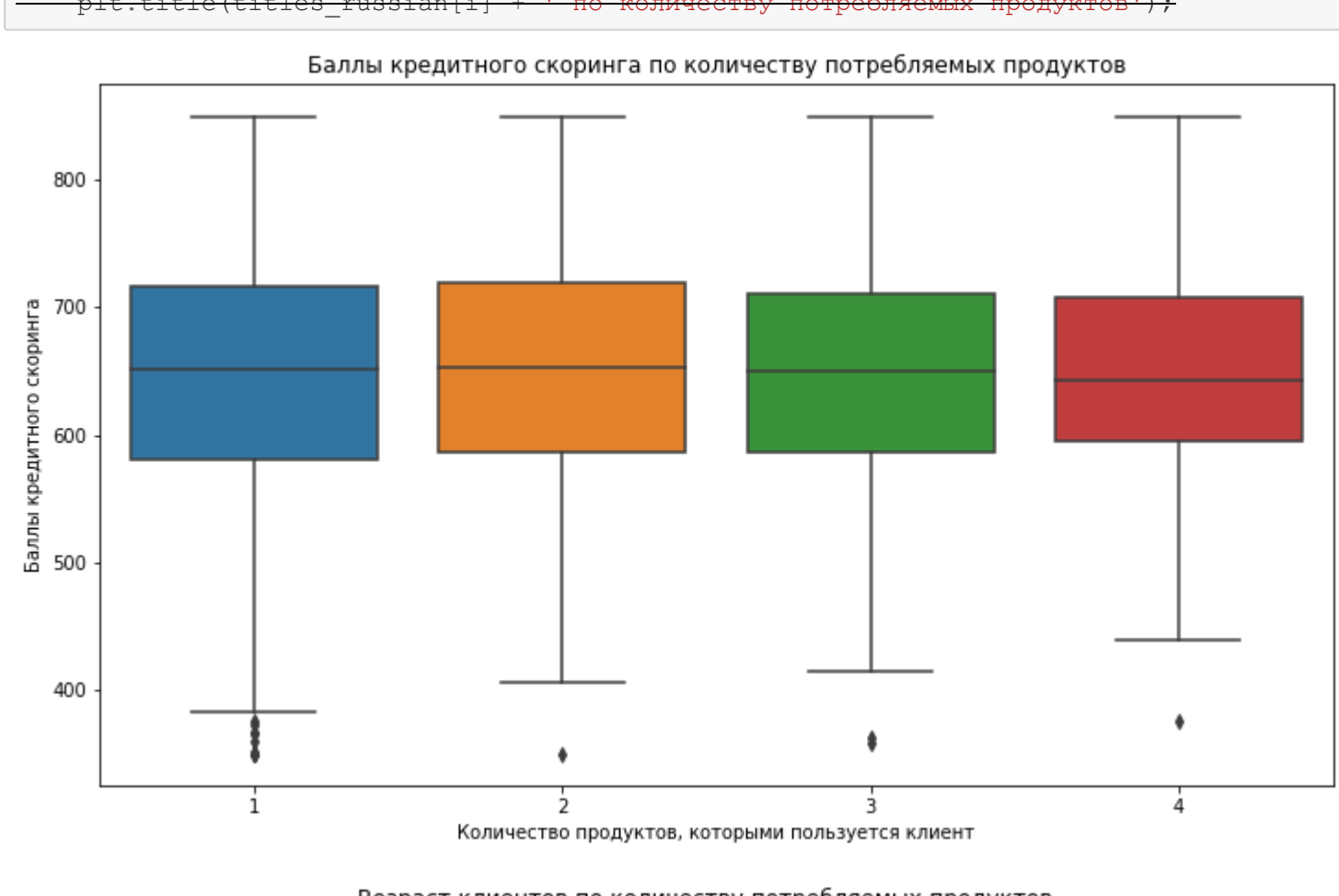
In [27]:
```python
pd.concat([bank_dataset[bank_dataset['Churn'] == 1].groupby('Products').agg({'Products': 'count'}), \
           bank_dataset[bank_dataset['Churn'] == 0].groupby('Products').agg({'Products': 'count'})], axis=1).T.\
plot(kind='bar', figsize=(12, 7), grid=True, \
     title='Сравнение числа клиентов и потребляемых ими продуктов в зависимости от того, ушёл клиент или нет')
ax = plt.gca()
ax.set_xticklabels(['Ушёл', 'Нет'], rotation = 0);
```

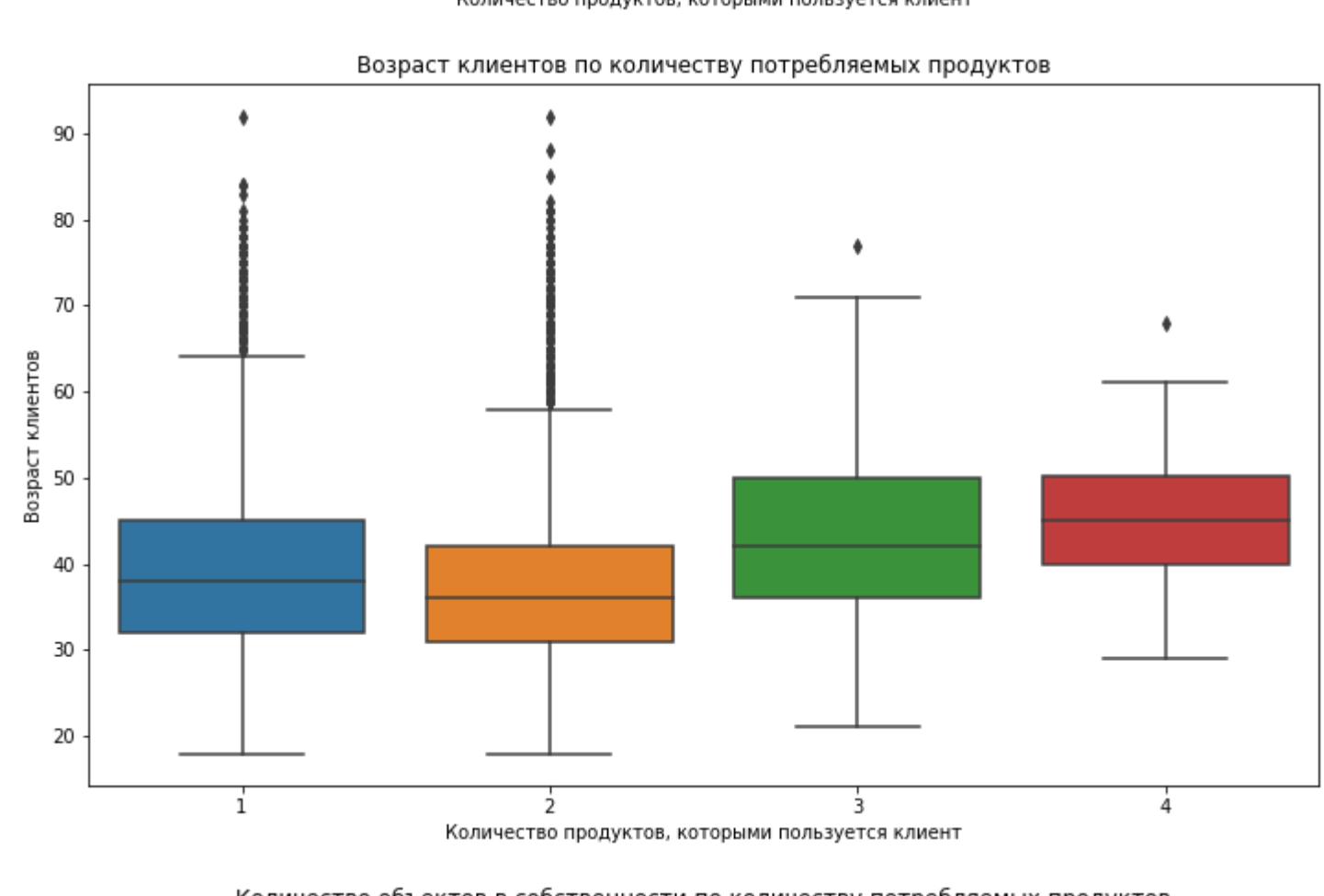


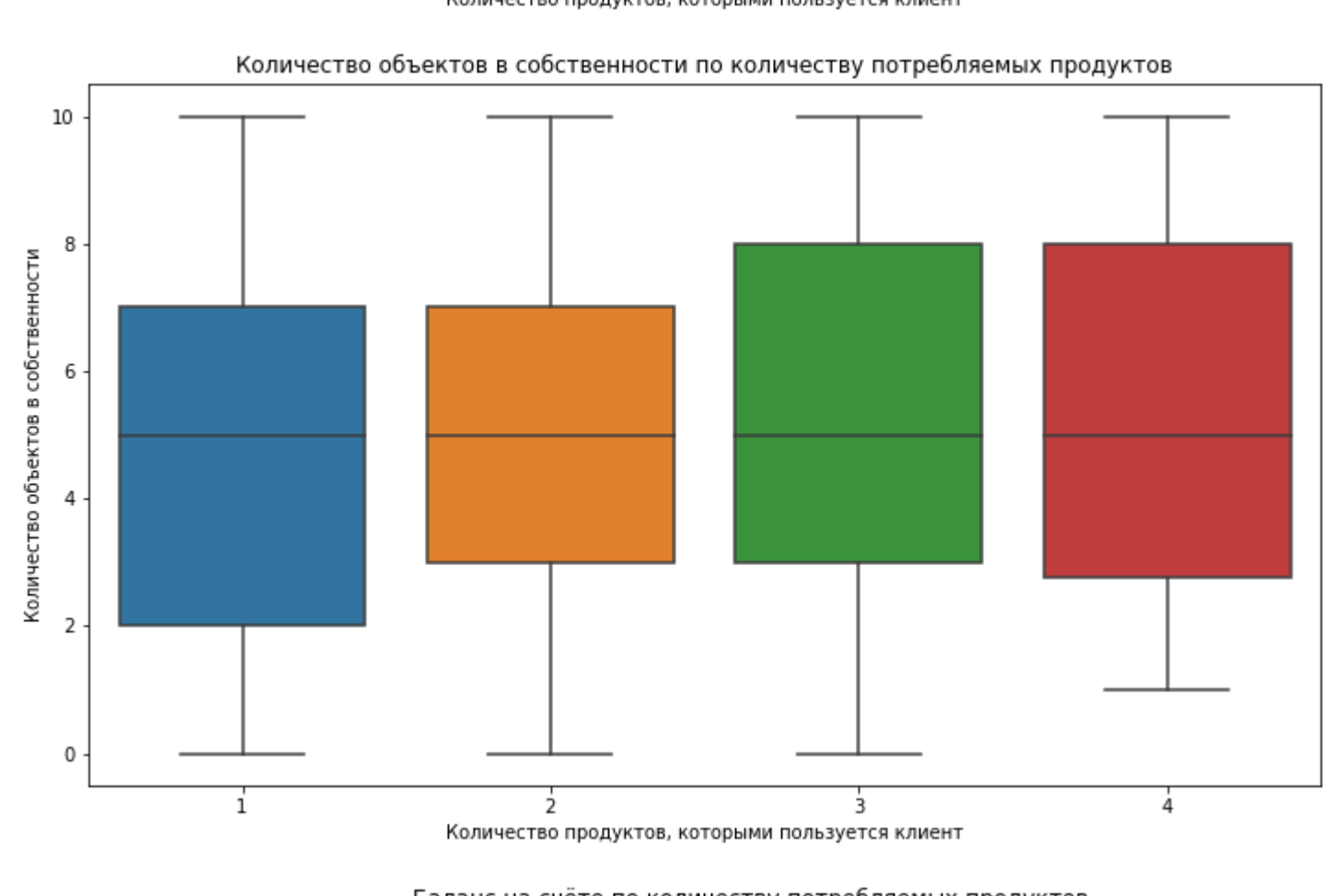
Сгруппируем пользователей по количеству потребляемых продуктов и сравним параметры каждой группы методом `boxplot()`.

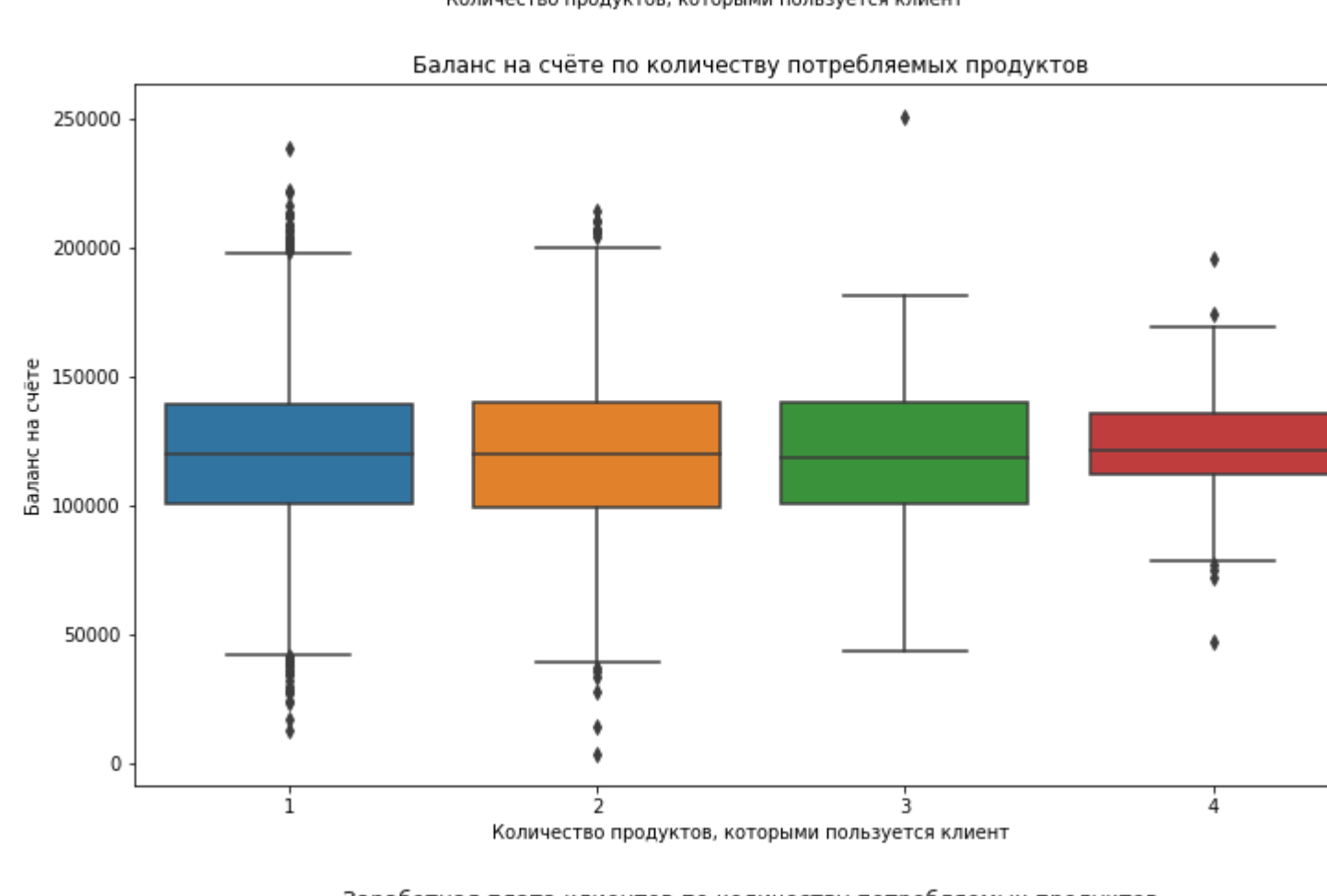
In [28]:
```python
titles = ['score', 'Age', 'Objects', 'Balance', 'estimated_salary']
titles_russian = ['Баллы кредитного скоринга', \
                  'Возраст клиентов', \
                  'Количество объектов в собственности', \
                  'Баланс на счёте', \
                  'Заработная плата клиентов']
```

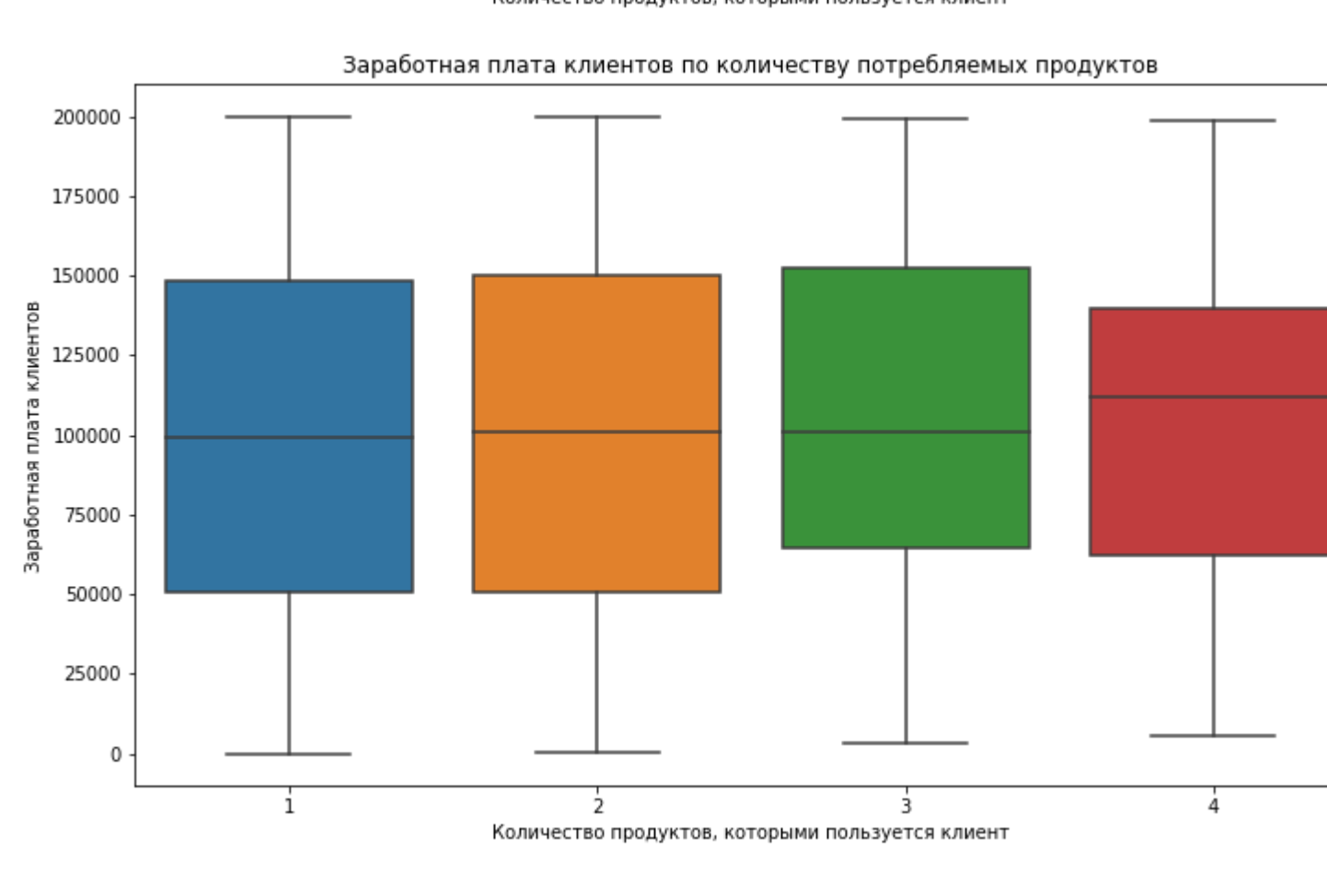
In [29]:
```python
for i in range(len(titles)):
    plt.figure(figsize=(12, 7))
    sns.boxplot(x='Products', y=titles[i], data=bank_dataset)
    ax = plt.gca()
    ax.set_xlabel('Количество продуктов, которыми пользуется клиент')
    ax.set_ylabel(titles_russian[i])
    plt.title(titles_russian[i] + ' по количеству потребляемых продуктов');
```







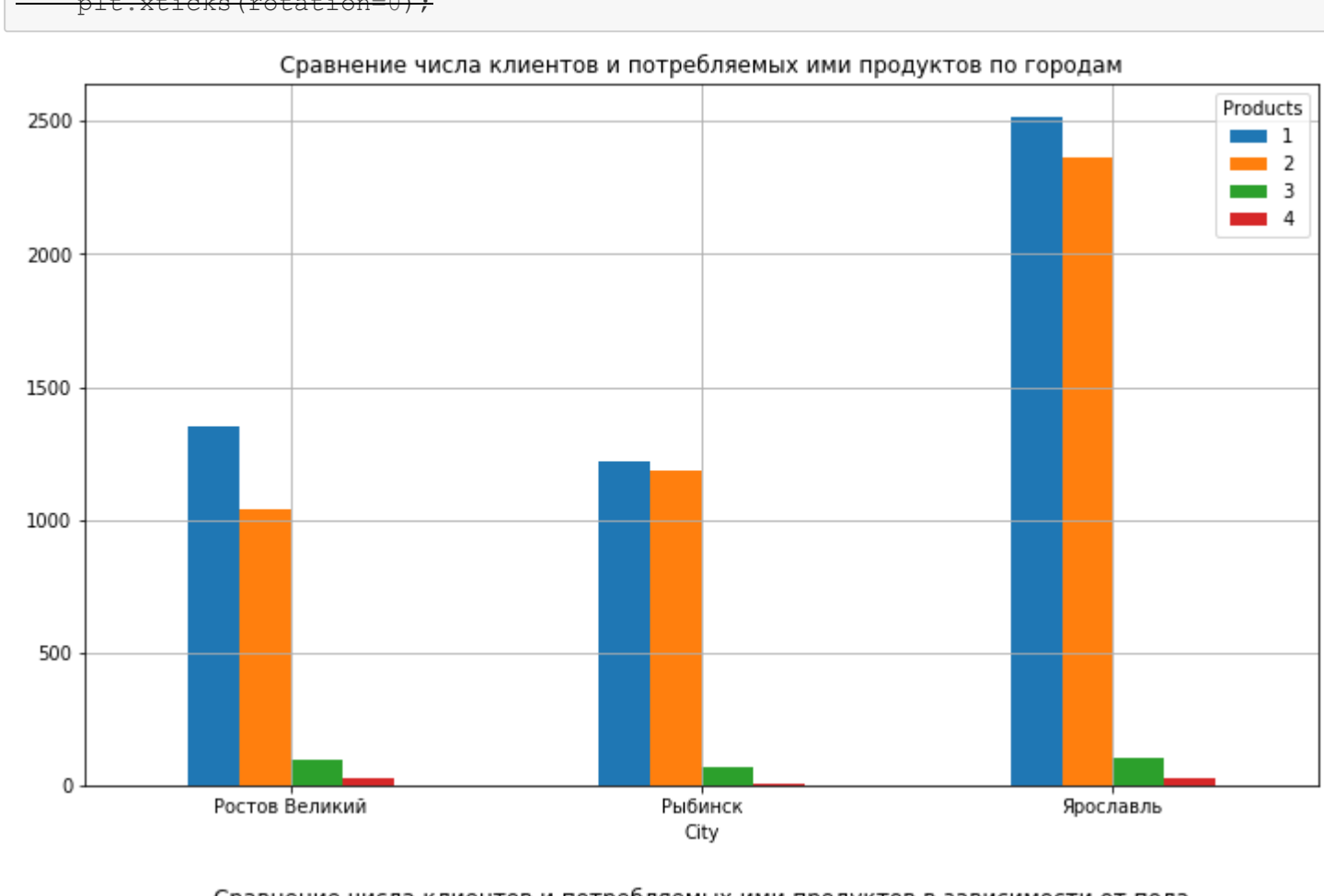


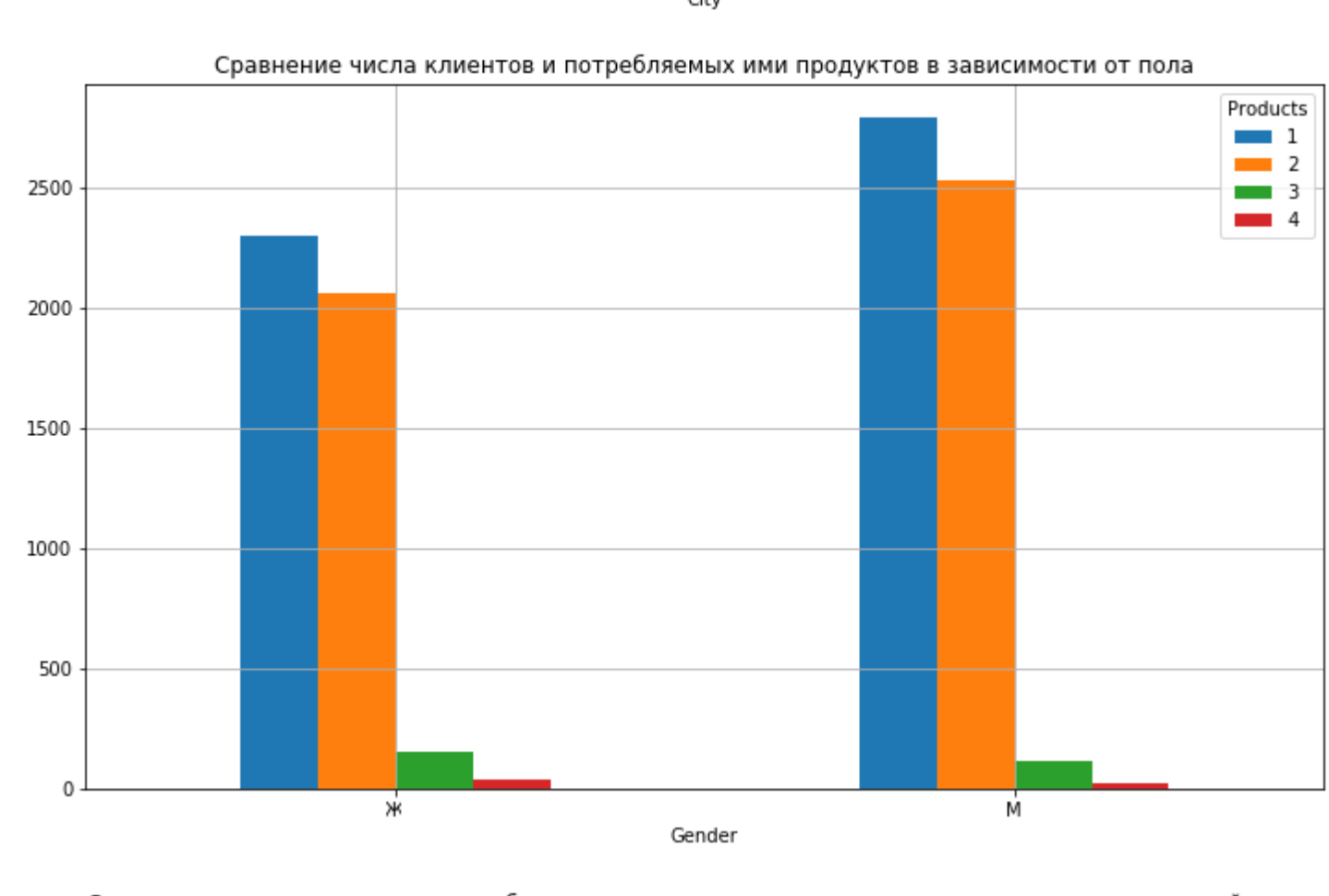


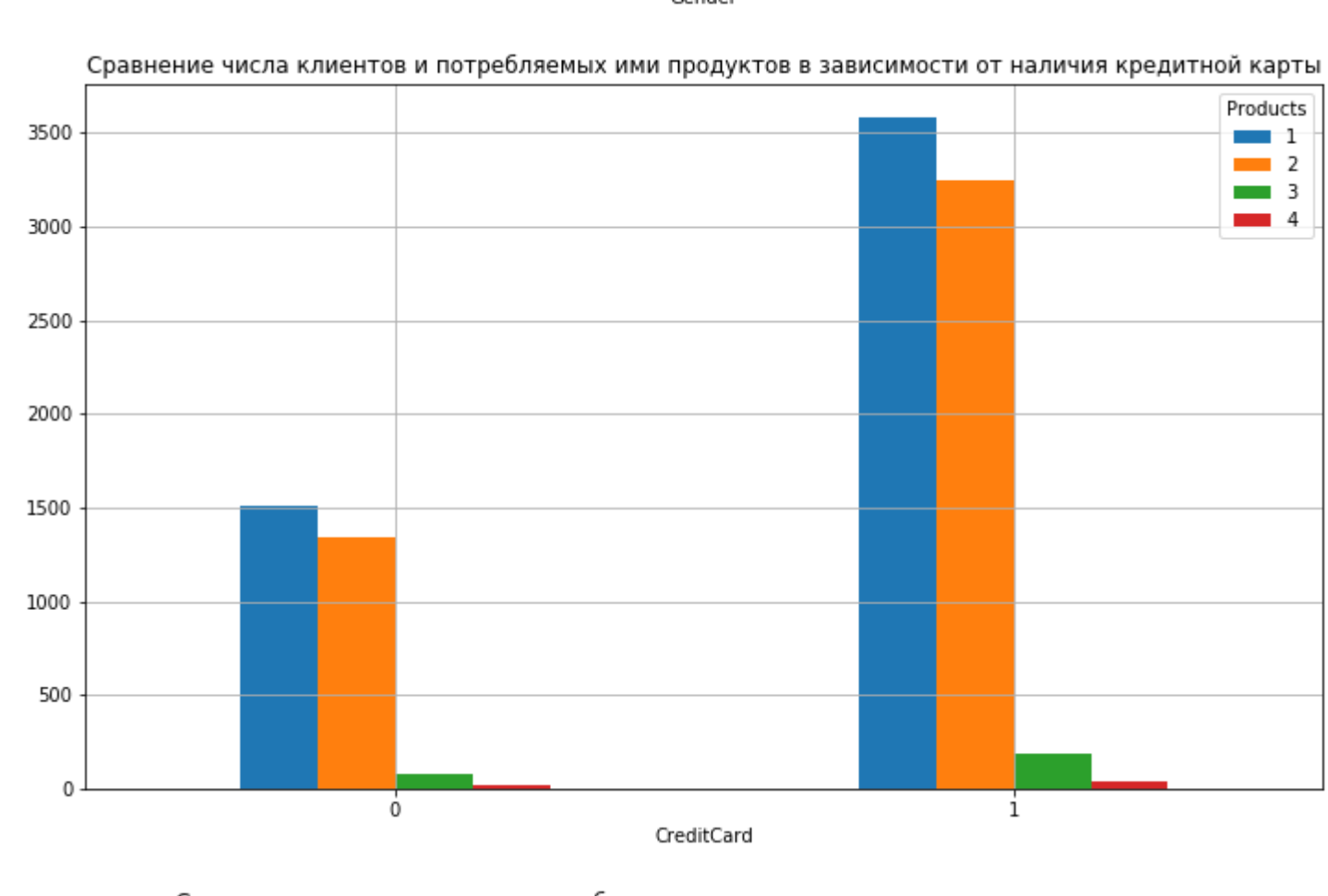
Для сравнения логических и дискретных переменных построим столбчатые диаграммы.

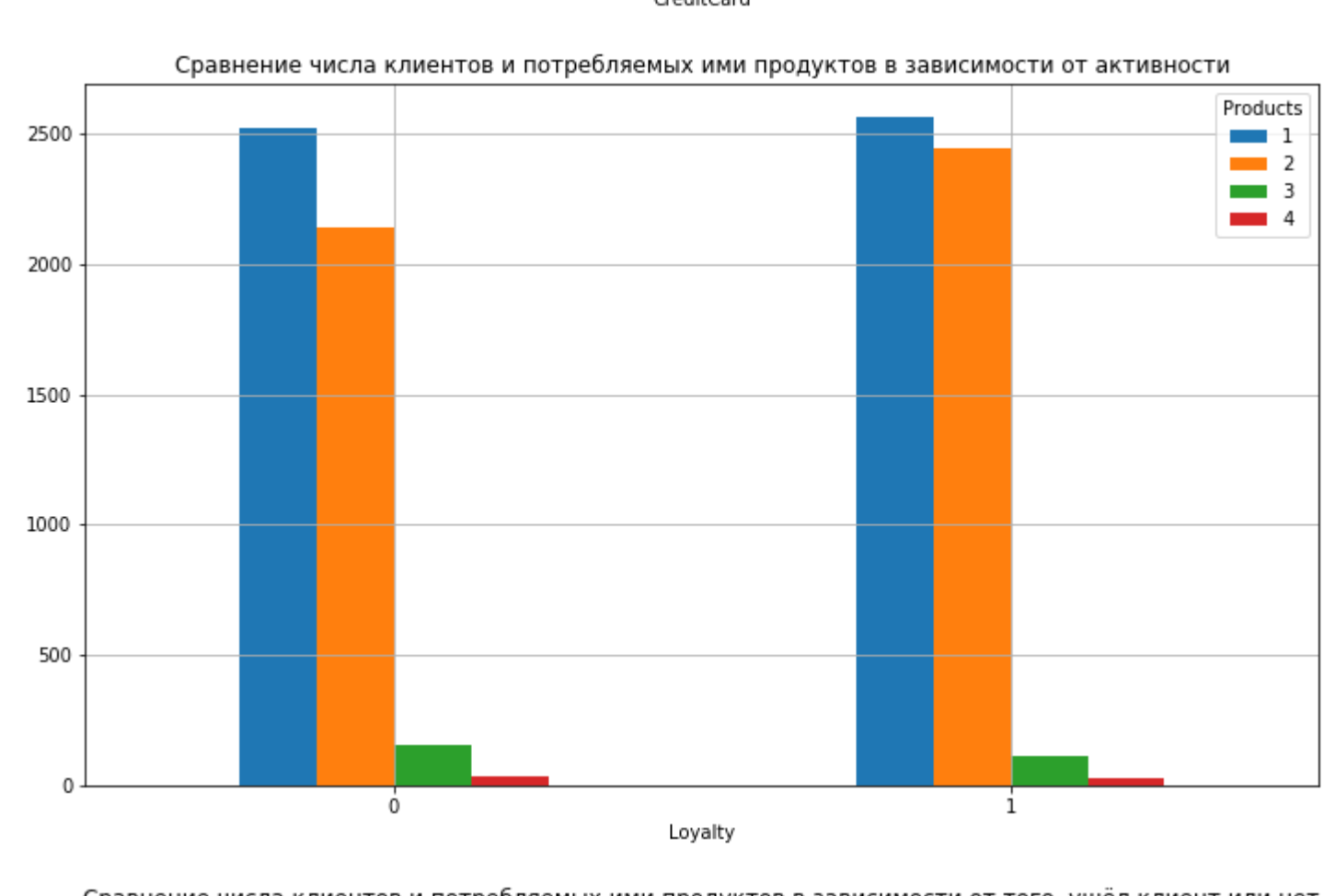
In [30]:
```python
titles = ['City', 'Gender', 'CreditCard', 'Loyalty', 'Churn']
titles_russian = ['по городам', \
                  'в зависимости от пола', \
                  'в зависимости от наличия кредитной карты', \
                  'в зависимости от активности', \
                  'в зависимости от того, ушёл клиент или нет']
```

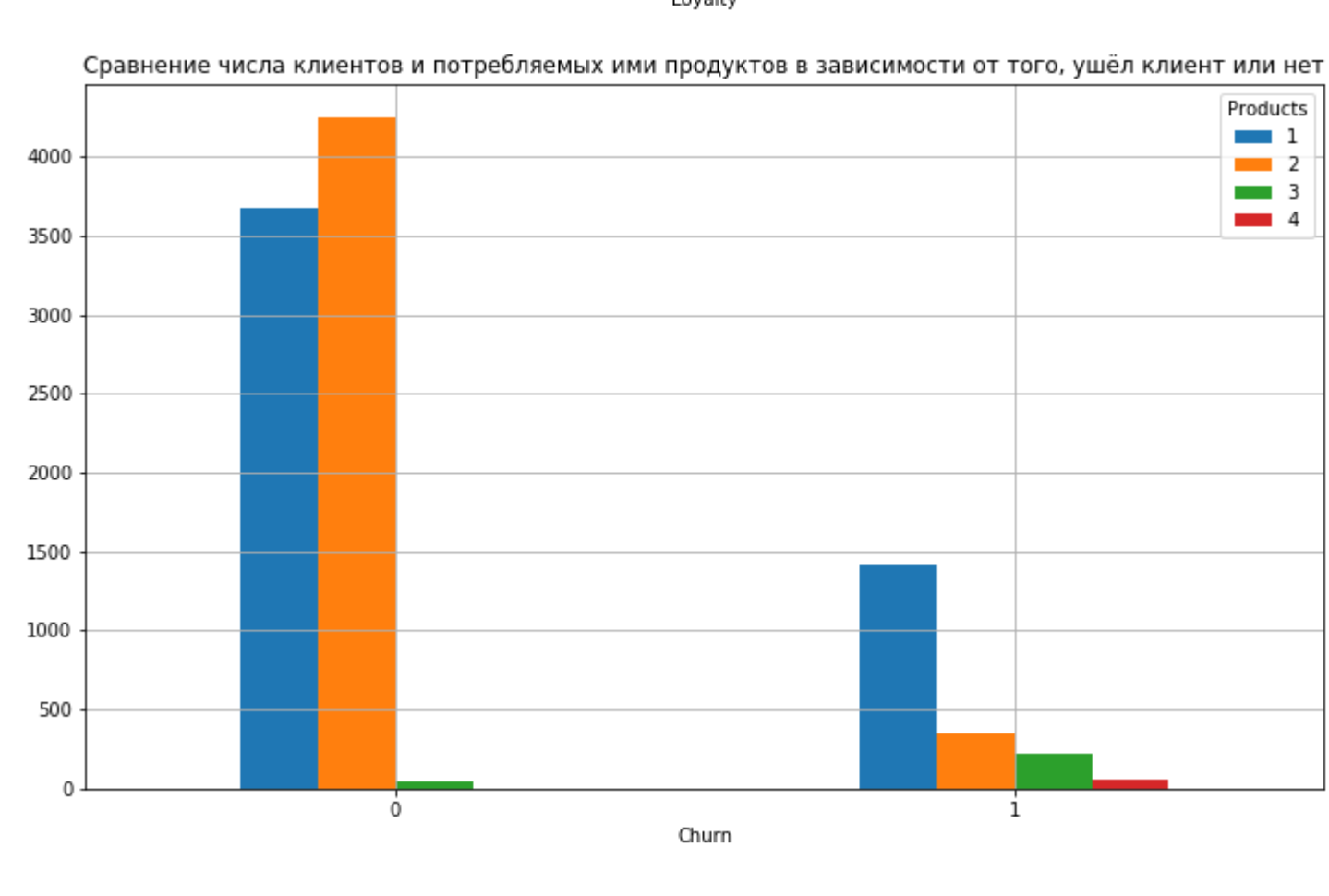
In [31]:
```python
for i in range(len(titles)):
    bank_dataset.pivot_table(index=titles[i], columns='Products', values='userid', aggfunc='count').\
    plot(kind='bar', figsize=(12, 7), grid=True, \
         title='Сравнение числа клиентов и потребляемых ими продуктов ' + titles_russian[i])
    ax = plt.gca()
    plt.xticks(rotation=0);
```





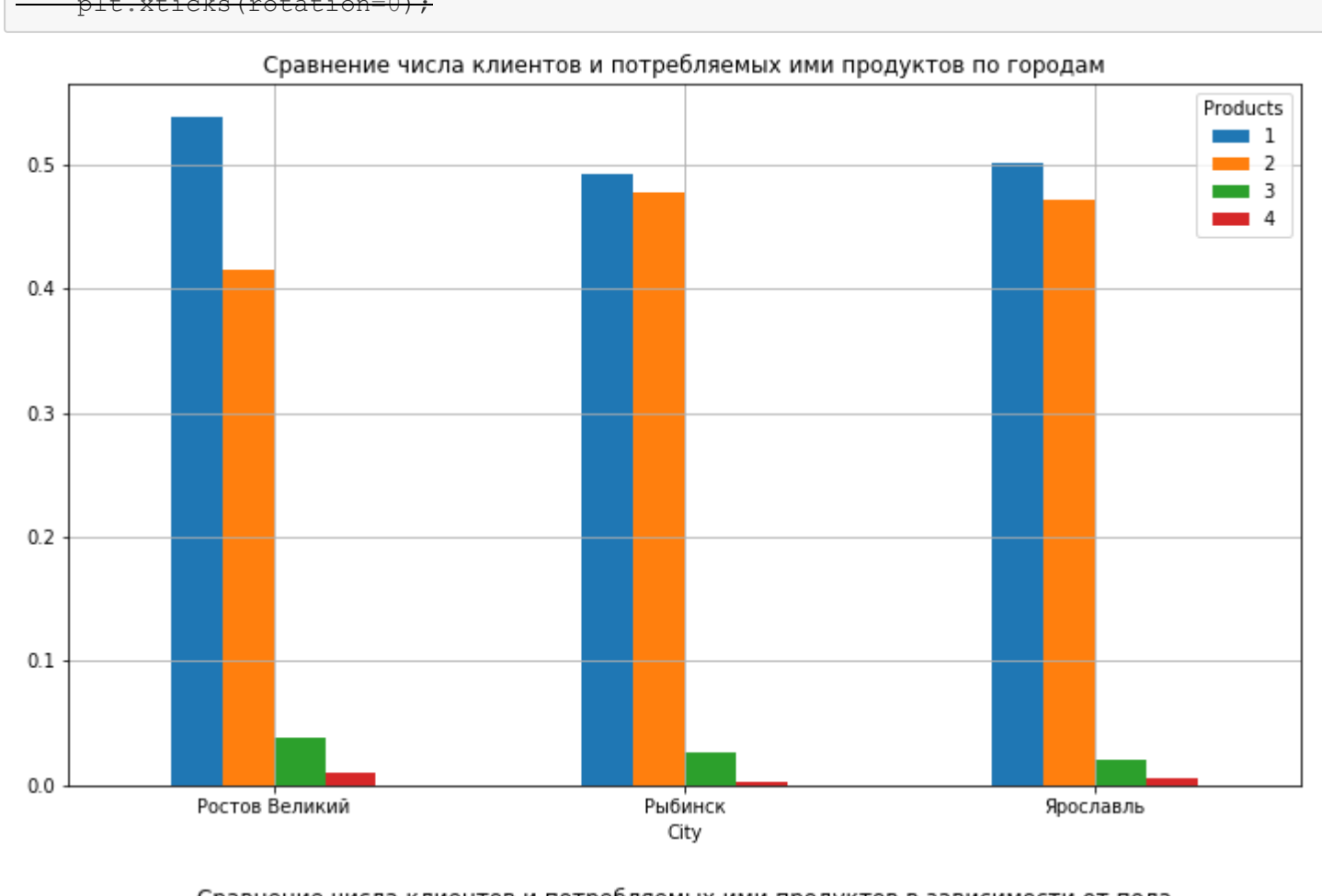


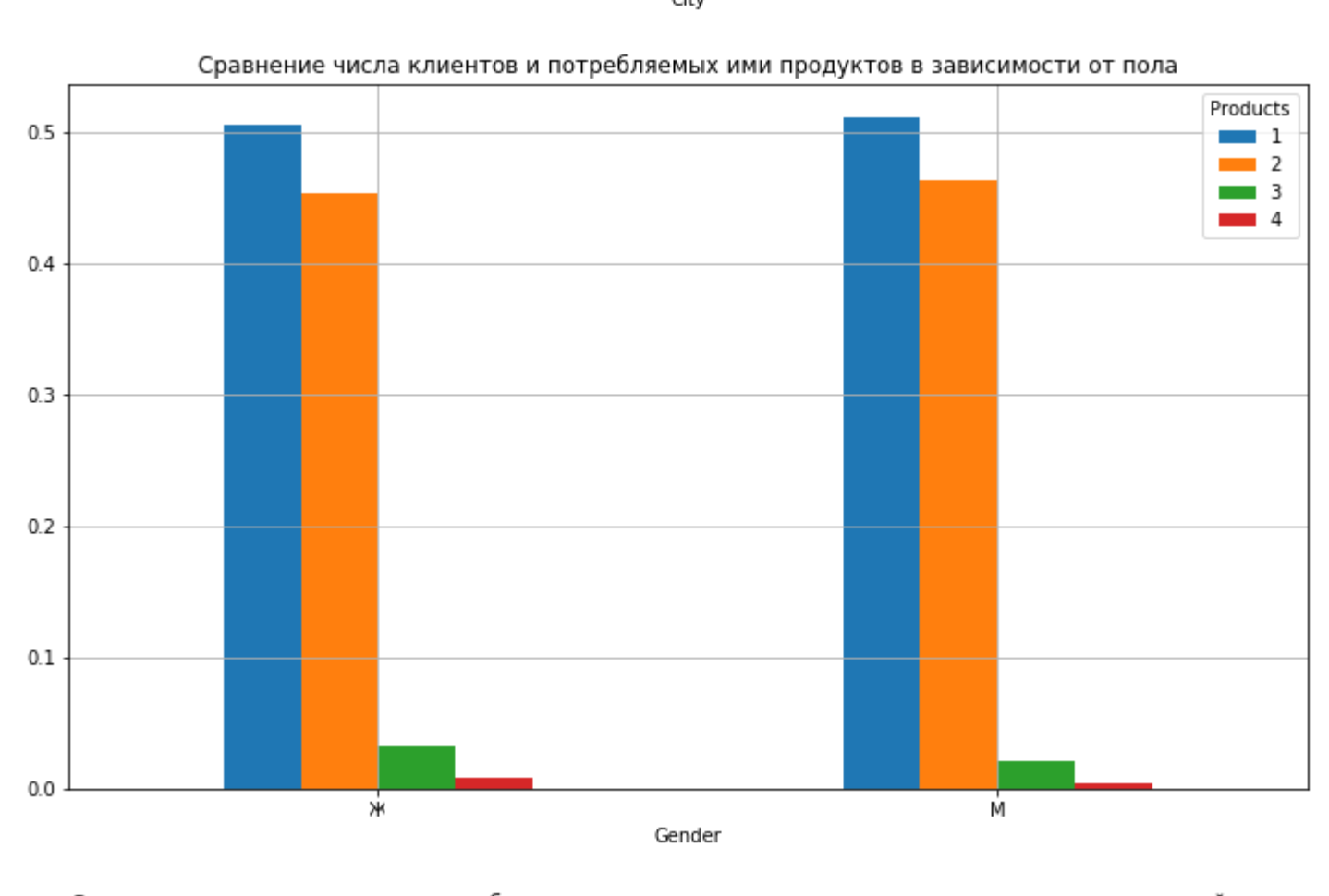


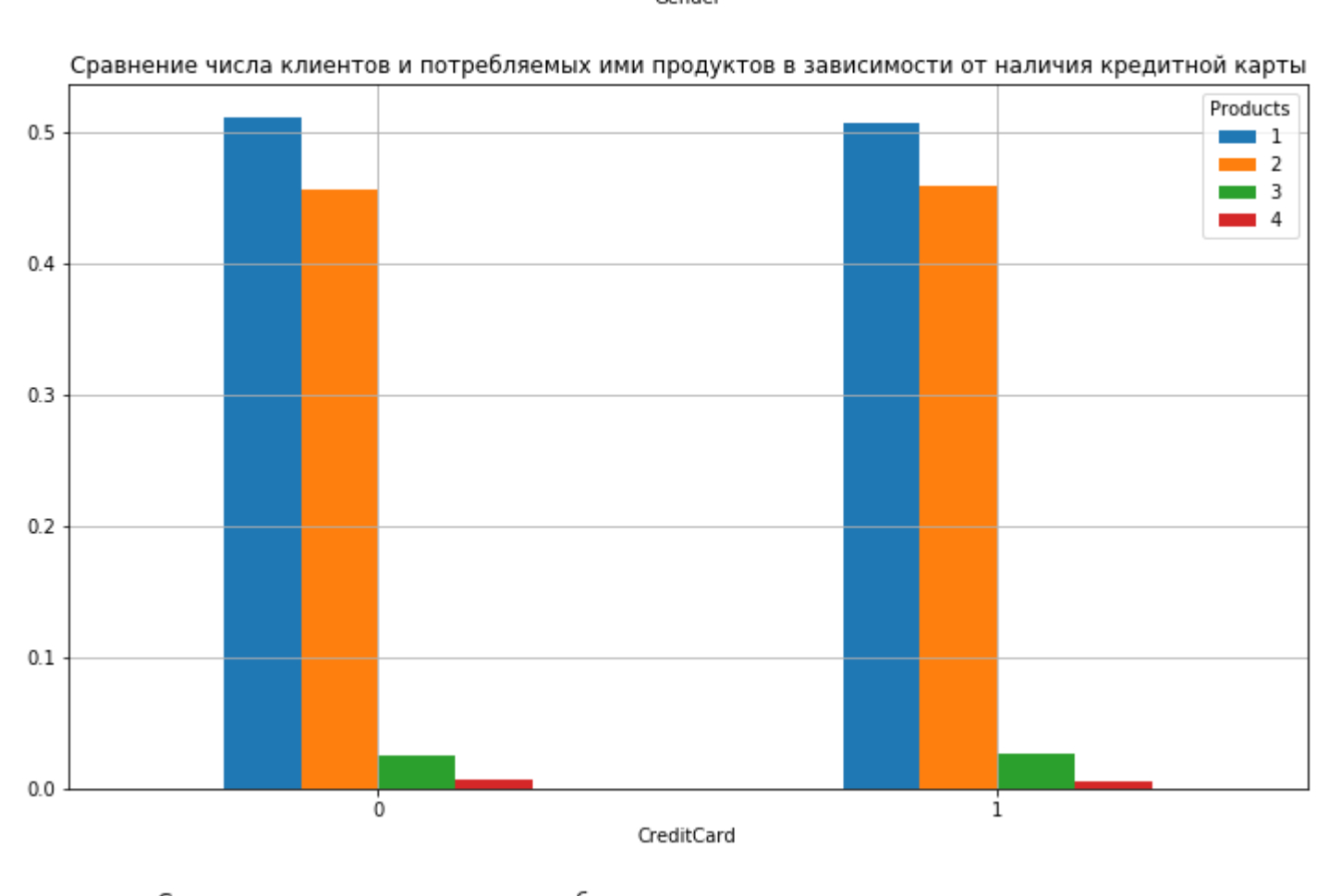


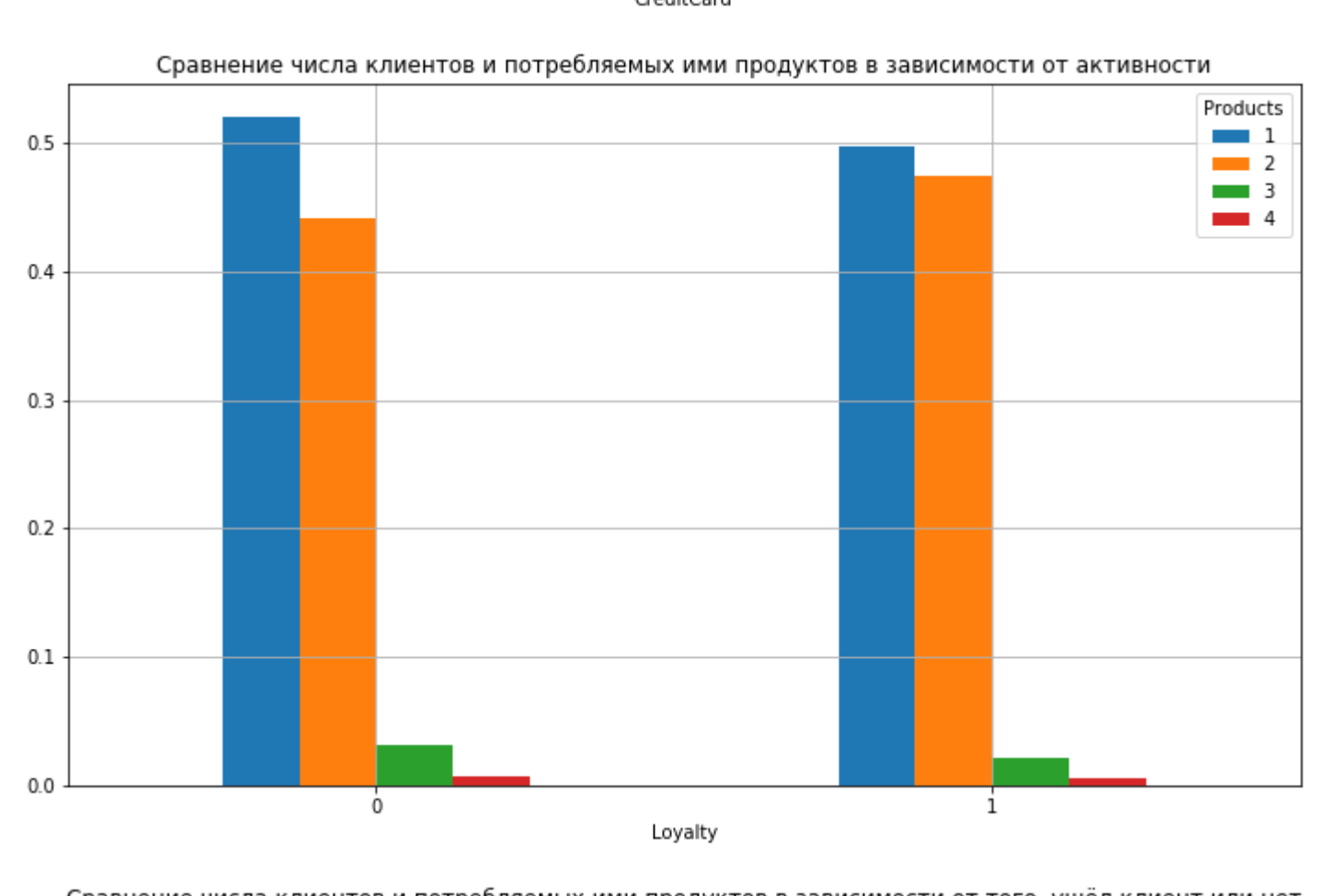
Выведем тоже самое в относительных значениях.

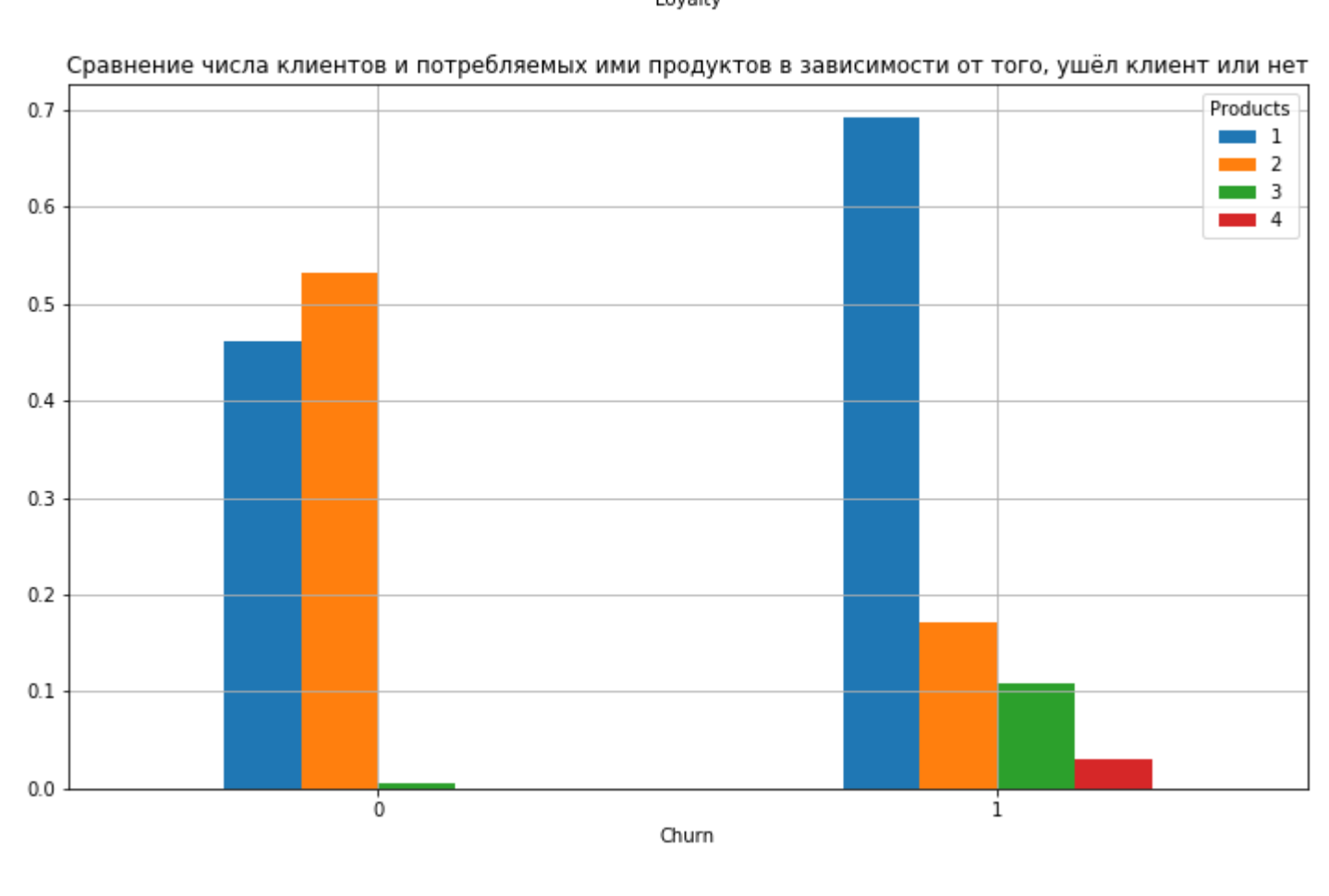
In [32]:
```python
for i in range(len(titles)):
    data = bank_dataset.pivot_table(index=titles[i], columns='Products', values='userid', aggfunc='count')
    data['sum'] = data.sum(axis=1)
    for j in range(1,5):
        data[j] = data[j] / data['sum']
    data.iloc[:,0:4].plot(kind='bar', figsize=(12, 7), grid=True,
                          title='Сравнение числа клиентов и потребляемых ими продуктов ' + titles_russian[i])
    ax = plt.gca()
    plt.xticks(rotation=0);
```











Вывод: у 70% ушедших клиентов был всего один продукт.

In [33]:
```python
bank_dataset.pivot_table(index='Products', aggfunc='mean')
```

Out[33]:

Products	Age	Balance	Churn	CreditCard	Loyalty	Objects	estimated_salary	score	userid
1	39.673092	119894.163723	0.277144	0.703777	0.504131	4.974626	99487.256851	649.120968	1.573237e+07
2	37.753595	119660.944362	0.075817	0.707190	0.532898	5.051852	100452.012821	652.188671	1.573484e+07
3	43.195489	119475.686310	0.827068	0.714286	0.424812	5.003759	104318.130564	648.105263	1.574182e+07
4	45.683333	122260.610870	1.000000	0.683333	0.483333	5.300000	104763.723333	653.583333	1.571936e+07

Клиенты с одним продуктом: средний возраст 40, но высокий процент пенсионеров. Средняя сумма на балансе 119894, высокая волатильность. Средняя вероятность ухода, но 70% от общего числа ушедших клиентов. Чаще всего есть кредитная карта. Средняя активность. Среднее количество объектов в собственности — 5, но часто ниже. Заработная плата средняя. Среднее число баллов кредитного скоринга, но часто ниже.

Клиенты с двумя продуктами: средний возраст 38, но высокий процент пенсионеров. Средняя сумма на балансе 119660, высокая волатильность. Наименьшая вероятность ухода. Чаще всего есть кредитная карта. Активность выше среднего. Среднее количество объектов в собственности — 5. Заработная плата средняя. Среднее число баллов кредитного скоринга.

Клиенты с тремя продуктами: средний возраст 43, процент пенсионеров ниже среднего. Средняя сумма на балансе 119475, волатильность ниже среднего. Почти все клиенты ушли. Чаще всего есть кредитная карта. Активность ниже среднего. Среднее количество объектов в собственности — 5, часто выше. Заработная плата выше среднего. Среднее число баллов кредитного скоринга.

Клиенты с четырьмя продуктами: средний возраст 46, процент пенсионеров низкий. Средняя сумма на балансе 122260 — выше, чем у остальных групп, волатильность низкая. Все клиенты ушли. Чаще всего есть кредитная карта, но реже, чем у остальных групп. Средняя активность. Среднее количество объектов в собственности — 5, часто выше. Заработная плата выше среднего. Среднее число баллов кредитного скоринга.

[К оглавлению]

Шаг 3. Формулировка и проверка статистических гипотез различия дохода

Сгруппируем клиентов, которые пользуются двумя продуктами банка и которые пользуются одним.

In [34]:
```python
bank_dataset_A = bank_dataset[bank_dataset['Products'] == 2]
```

In [35]:
```python
bank_dataset_B = bank_dataset[bank_dataset['Products'] == 1]
```

Сформулируем нулевую гипотезу:

H_0: уровень дохода клиентов, которые пользуются двумя продуктами банка и которые пользуются одним, не различается.

H_a: есть статистически значимая разница в уровне дохода клиентов, которые пользуются двумя продуктами банка и которые пользуются одним.

Воспользуемся критерием Манна-Уитни,поскольку он не требует нормального распределения выборок.

In [36]:
```python
alpha = .05 # критический уровень статистической значимости

results = st.mannwhitneyu(bank_dataset_A['estimated_salary'], bank_dataset_B['estimated_salary'])
print('p-значение: ', "{0:.3f}".format(results.pvalue))

if (results.pvalue < alpha):
    print("Отвергаем нулевую гипотезу: разница статистически значима")
else:
    print("Не получилось отвергнуть нулевую гипотезу, вывод о различии сделать нельзя")
```

```
p-значение:  0.204
Не получилось отвергнуть нулевую гипотезу, вывод о различии сделать нельзя
```

Вывод: уровень дохода клиентов, которые пользуются двумя продуктами банка и которые пользуются одним, не различается.

In [37]:
```python
bank_dataset_A['estimated_salary'].mean()
```

Out[37]: `100452.01282135077`

In [38]:
```python
bank_dataset_B['estimated_salary'].mean()
```

Out[38]: `99487.25685090481`

Проверим, различается ли баланс у клиентов, которые пользуются двумя продуктами банка и которые пользуются одним.

Сформулируем нулевую гипотезу:

H_0: баланс на счетах клиентов, которые пользуются двумя продуктами банка и которые пользуются одним, не различается.

H_a: есть статистически значимая разница в балансах счетов клиентов, которые пользуются двумя продуктами банка и которые пользуются одним.

In [39]:
```python
alpha = .05 # критический уровень статистической значимости

results = st.mannwhitneyu(bank_dataset_A['Balance'], bank_dataset_B['Balance'])
print('p-значение: ', "{0:.3f}".format(results.pvalue))

if (results.pvalue < alpha):
    print("Отвергаем нулевую гипотезу: разница статистически значима")
else:
    print("Не получилось отвергнуть нулевую гипотезу, вывод о различии сделать нельзя")
```

```
p-значение:  0.000
Отвергаем нулевую гипотезу: разница статистически значима
```

In [40]:
```python
bank_dataset_A['Balance'].mean()
```

Out[40]: `119660.94436180903`

In [41]:
```python
bank_dataset_B['Balance'].mean()
```

Out[41]: `119894.16372337879`

Вывод: средний баланс на счетах клиентов, которые пользуются двумя продуктами банка, в полтора раза выше среднего баланса на счетах клиентов, которые пользуются одним продуктом.

[К оглавлению]

Выводы

Уровень дохода клиентов, которые пользуются двумя продуктами банка и которые пользуются одним, не различается.

Средний баланс на счетах клиентов, которые пользуются двумя продуктами банка, в полтора раза выше среднего баланса на счетах клиентов, которые пользуются одним продуктом.

Количество продуктов, которыми пользуется клиент, не зависит от баллов кредитного скоринга.

С возрастом клиенты используют бóльшее число продуктов, однако пенсионеры, как правило, пользуются одним-двумя продуктами.

У клиентов, потребляющих бóльшее количество продуктов, как правило, больше объектов в собственности.

Баланс выше у клиентов, пользующихся либо всего одним продуктом, либо сразу многими.

У клиентов, потребляющих бóльшее количество продуктов, зарплата незначительно выше.

Относительное количество потребляемых продуктов внутри городов различается незначительно. По абсолютному количеству лидирует Ярославль (в два раза больше Ростова Великий и Рыбинска).

Количество потребляемых продуктов не зависит от пола. Общее число клиентов-мужчин незначительно превышает число клиентов-женщин.

Количество потребляемых продуктов не зависит от наличия кредитной карты. Владельцев кредитной карты в два с лишним раза больше, чем клиентов без неё.

Количество потребляемых продуктов не зависит от активности клиента.

У 70% ушедших клиентов был всего один продукт.

[К оглавлению]