Тема - Изучение проблемы оттока клиентов в банке

- **Анализ банковских данных** - Дипломный проект

Содержание

Описание проекта

Проанализировать клиентов регионального банка и сегментировать пользователей по количеству потребляемых продуктов, обращая особое внимание на отток.

- Проведите исследовательский анализ данных,
- Сегментируйте пользователей на основе данных о количестве потребляемых продуктов,
- Сформулируйте и проверьте статистические гипотезы.

  Проверьте гипотезу различия возраста между теми клиентами, которые пользуются двумя продуктами банка, и теми, которые пользуются одним.

  Сформулируйте и проверьте статистическую гипотезу относительно представленных данных.

  Описание данных:

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

Колонки:

- `userid` — идентификатор пользователя,
- `score` — баллы кредитного скоринга,
- `City` — город,
- `Gender` — пол,
- `Age` — возраст,
- `equity` — приблизительная оценка собственности клиента (в баллах),
- `Balance` — баланс на счёте,
- `Products` — количество продуктов, которыми пользуется клиент,
- `credit_card` — есть ли кредитная карта,
- `last_activity` — был ли клиент активен последнее время,
- `salary` — заработная плата клиента,
- `Churn` — уходит или нет.

По итогам исследования необходимо подготовить презентацию.

Шаг 1. Открыть датасет и познакомиться с данными

Загрузка необходимых библиотек

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import seaborn as sns
import numpy as np
import warnings
warnings.simplefilter(action="ignore", category=FutureWarning)
from scipy.stats import ttest_ind
from scipy.stats import mannwhitneyu
from scipy.stats import shapiro
import statsmodels.api as sm
```

Открытие датасета и первичное знакомство с данными

In [2]:
```
try:
    df = pd.read_csv('bank_scrooge.csv')
except:
    path = "https://drive.google.com/uc?export=download&id=1-U61mhTz_N1ARjy2XSAZ7IlQqGjeqF
    df = pd.read_csv(path)
```

In [3]:
```
df.head()
```

Out[3]:

	USERID	score	city	gender	age	equity	balance	products	credit_card	last_activity	EST_SALARY	ch
0	183012	850.0	Рыбинск	Ж	25.0	1	59214.82	2	0	1	75719.14	
1	146556	861.0	Рыбинск	Ж	37.0	5	850594.33	3	1	0	86621.77	
2	120722	892.0	Рыбинск	Ж	30.0	0	NaN	1	1	1	107683.34	

	USERID	score	city	gender	age	equity	balance	products	credit_card	last_activity	EST_SALARY	ch
3	225363	866.0	Ярославль	Ж	51.0	5	1524746.26	2	0	1	174423.53	
4	157978	730.0	Ярославль	М	34.0	5	174.00	1	1	0	67353.16	

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   USERID         10000 non-null  int64  
 1   score          10000 non-null  float64
 2   city           10000 non-null  object 
 3   gender         10000 non-null  object 
 4   age            9974 non-null   float64
 5   equity         10000 non-null  int64  
 6   balance        7705 non-null   float64
 7   products       10000 non-null  int64  
 8   credit_card    10000 non-null  int64  
 9   last_activity  10000 non-null  int64  
 10  EST_SALARY     10000 non-null  float64
 11  churn          10000 non-null  int64  
dtypes: float64(4), int64(6), object(2)
memory usage: 937.6+ KB
```

In [5]:
```
df.shape
```

Out[5]:
```
(10000, 12)
```

В датасете 10 000 строк и 12 столбцов

Переименование столбцов (приведение к `snowcase`)

In [6]:
```
df.rename(columns = {'USERID':'user_id', 'EST_SALARY':'salary'}, inplace = True )
```

Проверим количество пропусков в датасете

In [7]:
```
pd.DataFrame(round(df.isna().mean()*100,1)).sort_values(0,ascending=False).style.backgrour
```

Out[7]:

	0
balance	23.000000
age	0.300000
user_id	0.000000
score	0.000000
city	0.000000
gender	0.000000
equity	0.000000
products	0.000000

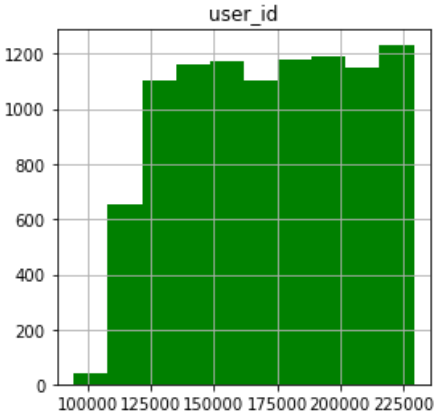
	0
credit_card	0.000000
last_activity	0.000000
salary	0.000000
churn	0.000000

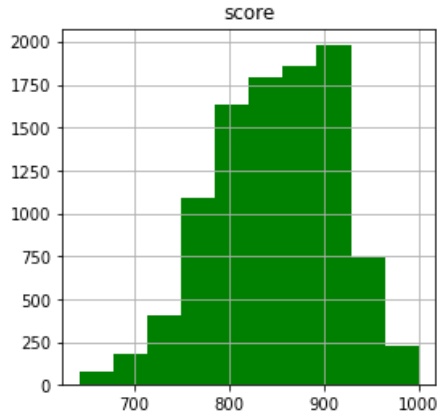
Как мы видим, **23% пропусков в поле с балансом на счете.**

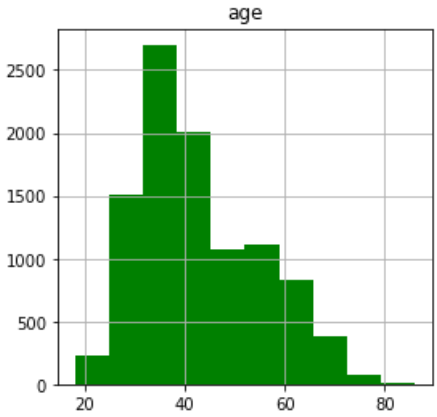
Также **0.3% пропусков в поле возраст клиента**. Изучим природу пропусков во 2й части исследования.

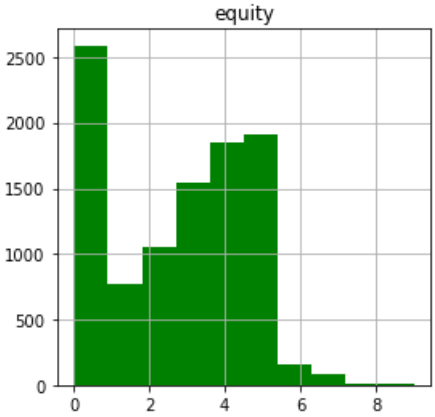
Построим гистограммы для числовых столбцов таблицы

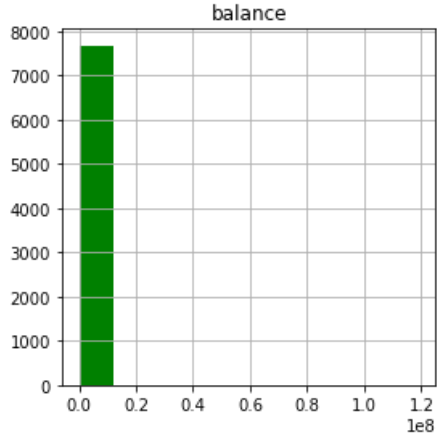
In [8]:
```
ax = df.hist(figsize=(15, 20),color='green')
```

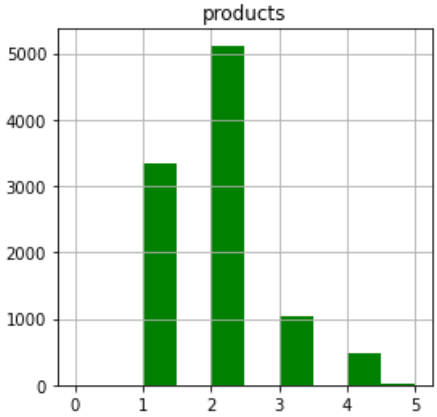


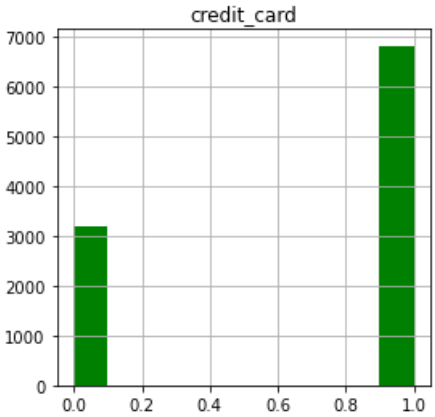


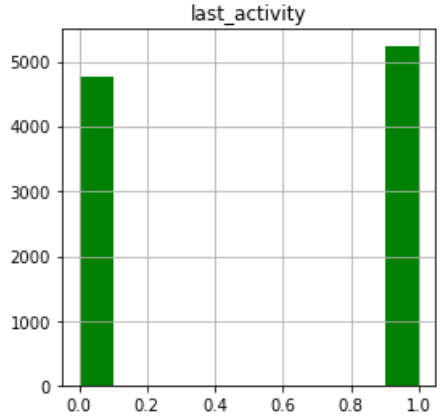


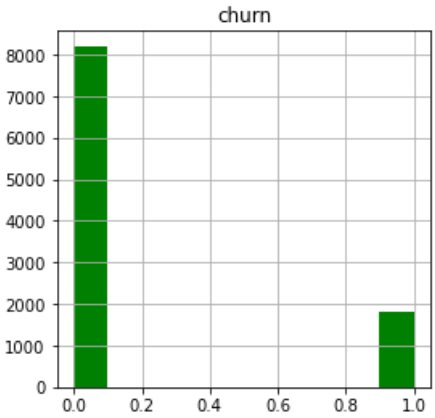












Как мы видим, в столбцах `balance` и `salary` присутствуют выбросы. Проанализируем их в дальнейшем.

Изучим уникальные значения столбцов

Посчитаем уникальные значения поля `user_id`

In [9]:

```
len(df['user_id'].unique())
```

Out[9]:
```
9927
```

Количество уникальных `user_id` - 9927, всего строк 10 000 - в нашем датасете есть дубликаты

Посмотрим на уникальные значения поля `score`

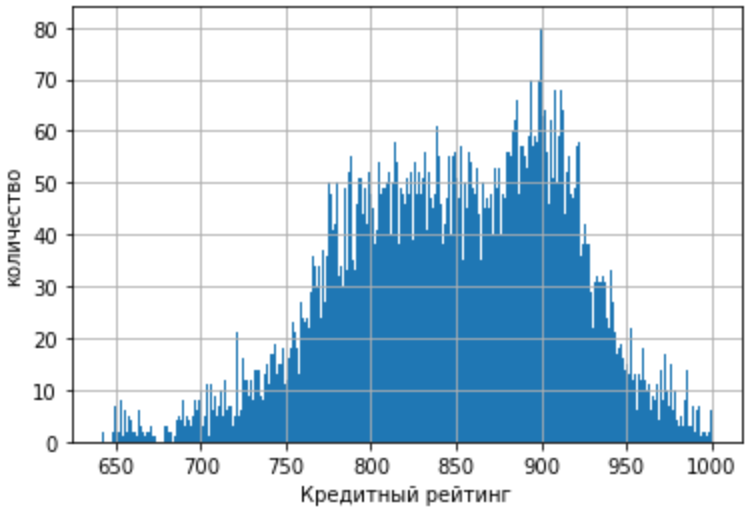
In [10]:
```
df['score'].unique()[1:10]
```

Out[10]:
```
array([861., 892., 866., 730., 856., 807., 825., 923., 906.])
```

Переведем значение столбца `score` в `int`

In [11]:
```
df["score"] = df["score"].astype(int)
```

In [12]:
```
ax = df['score'].hist(bins=df['score'].max()- df['score'].min()+1)\
.set(xlabel='Кредитный рейтинг',ylabel='количество', label = 'Распределение кредитного рей
```

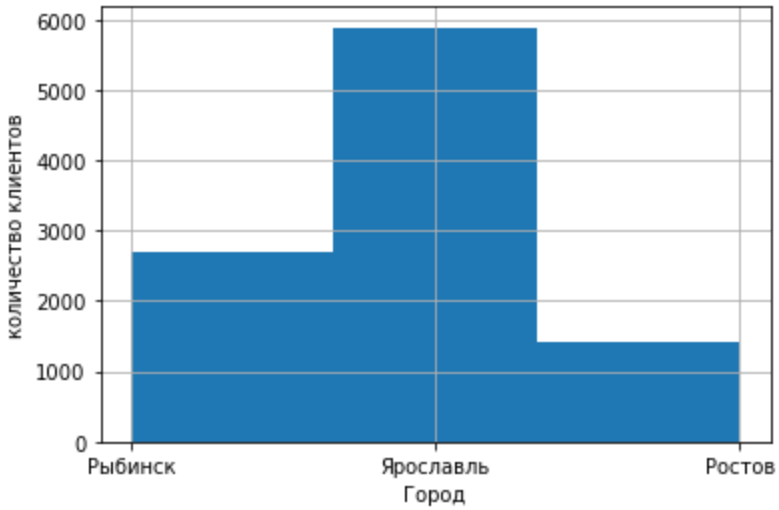


Посмотрим на уникальные значения поля `city`

In [13]:
```
df['city'].unique()
```

Out[13]:
```
array(['Рыбинск', 'Ярославль', 'Ростов'], dtype=object)
```

In [14]:
```
ax = df['city'].hist(bins=3).set(xlabel='Город',ylabel='количество клиентов', label = 'Рас
```



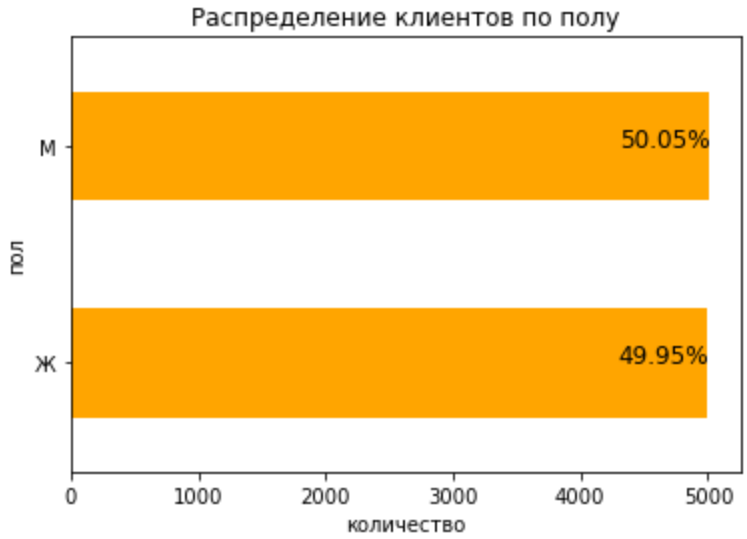
В датасете 3 города - Рыбинск, Ярославль и Ростов. Большинство клиентов из Ярославля, на втором месте - Рыбинск, затем - Ростов.

Посмотрим на уникальные значения поля gender

In [15]:
```
df['gender'].unique()
```

Out[15]:
```
array(['Ж', 'М'], dtype=object)
```

In [16]:
```
total = df.groupby('gender')['user_id'].count().sum()

ax = df.groupby('gender')['user_id'].count().plot(kind='barh', color='orange')
ax.set_xlabel('количество')
ax.set_ylabel('пол')
ax.set_title('Распределение клиентов по полу')

for i, v in enumerate(df.groupby('gender')['user_id'].count().values):
    ax.text(v-700,i, f"{v/total*100:.2f}%", color='black', fontsize=12)

plt.show()
```



В датасете - **2 типа пола** - **женский** (Ж) и **мужской** (М).

Женщин и мужчин в нашем датасете примерно одинаковое число.

Посмотрим на уникальные значения поля age

In [17]:
```
df['age'].unique()[1:10]
```

Out[17]:
```
array([37., 30., 51., 34., 56., 39., 38., 54., 67.])
```

In [18]:
```
df['age'].isna().sum()
```

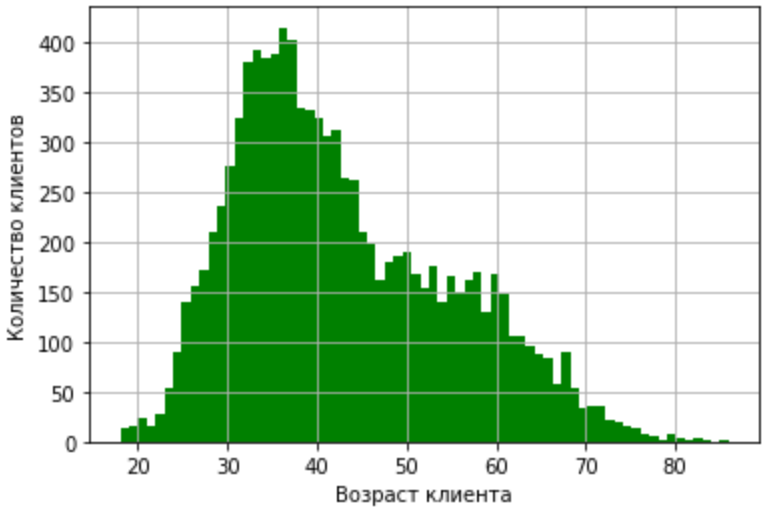
Out[18]:
```
26
```

В столбце `age` - есть 26 пропусков.

Переведем тип столбца `age` в `int`

In [19]:
```
df['age'] = df['age'].astype(int, errors='ignore')
```

In [20]:
```
ax = df['age'].hist(bins=69, color = 'green')\
.set(xlabel='Возраст клиента',ylabel='Количество клиентов', label = 'Распределение возраст
```



In [21]:
```
df['age'].mode()
```

Out[21]:
```
0    36.0
dtype: float64
```

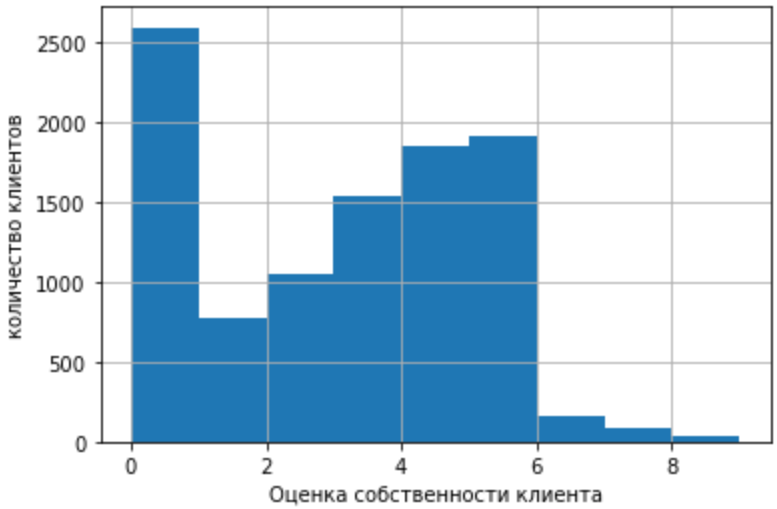
Большинству наших клиентов от 30 до 40 лет. Мода возраста клиента - 36 лет.

Посмотрим на уникальные значения поля `equity`

In [22]:
```
df['equity'].unique()
```

Out[22]:
```
array([1, 5, 0, 4, 3, 2, 6, 7, 8, 9], dtype=int64)
```

In [23]:
```
ax = df['equity'].hist(bins=9)\
.set(xlabel='Оценка собственности клиента',ylabel='количество клиентов', label = 'Распреде
```



В поле приблизительная оценка собственности клиента есть значения от 0 до 9 (значение в балах, больше значение - более состоятельный клиент)

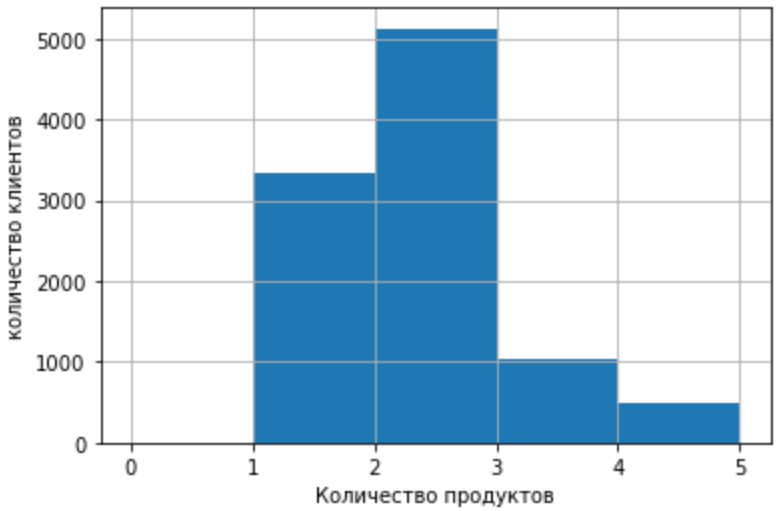
У большинства клиентов - оценка собственности составляет - 0, либо от 1 до 5

Посмотрим на уникальные значения поля products

In [24]:
```
df['products'].unique()
```

Out[24]:
```
array([2, 3, 1, 4, 5, 0], dtype=int64)
```

In [25]:
```
ax = df['products'].hist(bins=5)\
.set(xlabel='Количество продуктов',ylabel='количество клиентов', label = 'Распределение кл
```



In [26]:
```
df.groupby('products')['user_id'].count().sort_values(ascending = False)
```

Out[26]:
```
products
2    5126
1    3341
3    1039
4     474
5      19
0       1
Name: user_id, dtype: int64
```

В поле количество продуктов у клиента есть значения от 0 до 5 продуктов Банка.

У большинства клиентов от 1 до 4 продуктов Банка.

Посмотрим на уникальные значения поля `balance`

В связи с наличием выбросов в данных полях - построим график **BoxPlot** для дальнейшего анализа

In [27]:
```
ax = sns.boxplot(df['balance']).set(title='Распределение баланса клиентов', xlabel='Баланс
```

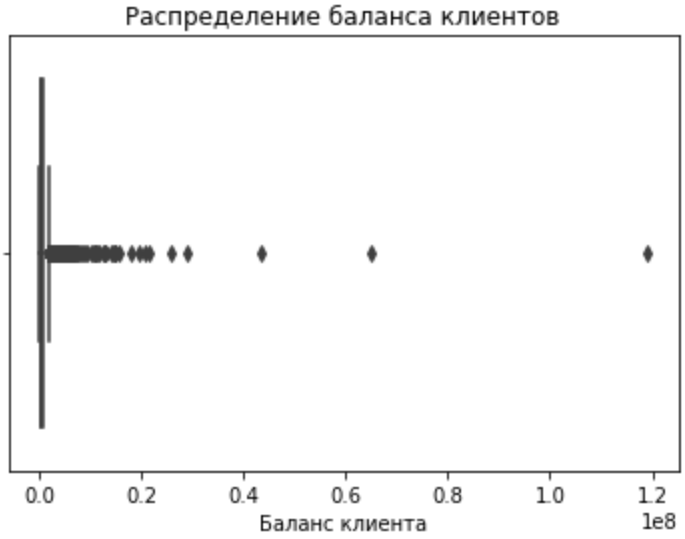


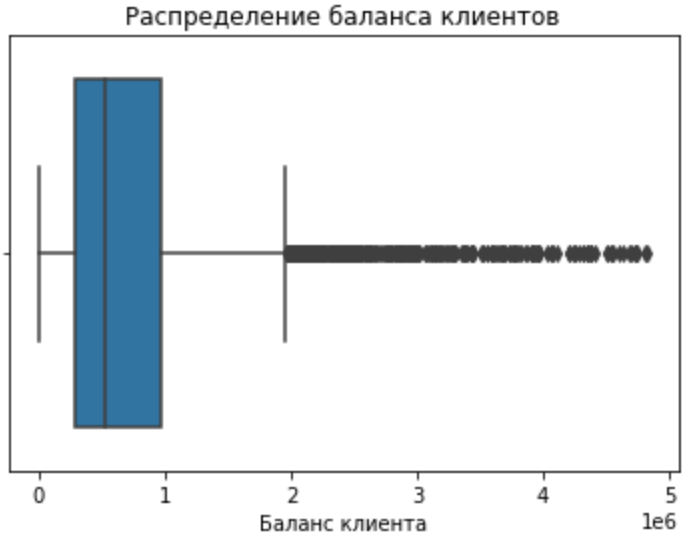
График является нечитаемым из за нескольких крупных выбросов -

Посчитаем **99% - перцентиль** - и нарисуем `boxplot` - взяв за его границу посчитанное значение

In [28]:
```
round(np.nanpercentile(df['balance'], 99),2)
```

Out[28]:
```
4827443.49
```

In [29]:
```
ax = sns.boxplot(df[df['balance']<np.nanpercentile(df['balance'], 99)]['balance'])\
.set(title='Распределение баланса клиентов', xlabel='Баланс клиента')
```



In [30]:
```
df['balance'].median()
```

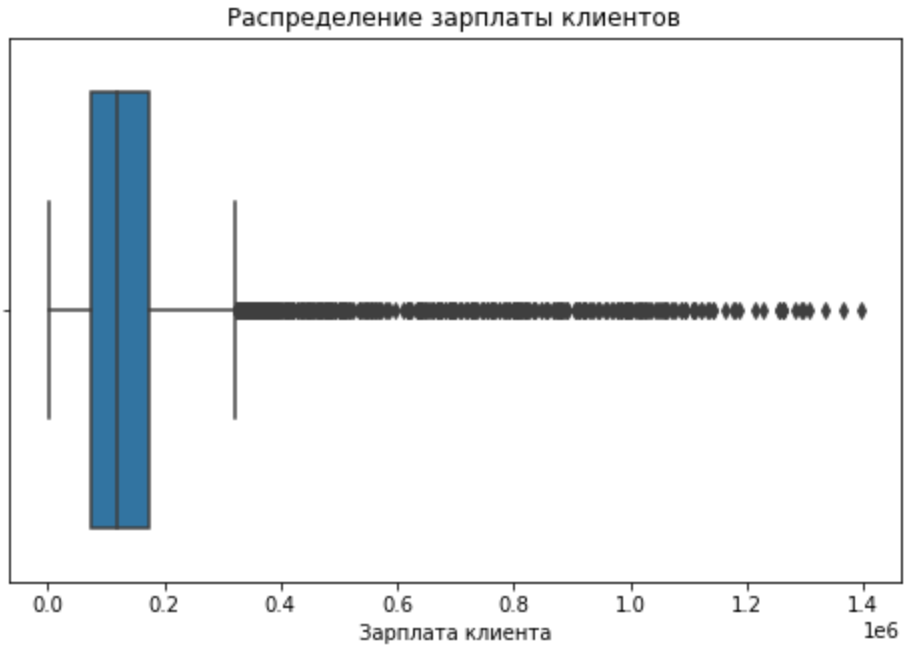
Out[30]:
```
524272.2
```

Большинство наших клиентов имеют **баланс на счете от 0 до 2 млн. рублей**

Медианное значение - 524 000 рублей.

Посмотрим на уникальные значения поля `salary`

In [31]:
```
plt.figure(figsize=(8, 5))
ax = sns.boxplot(df['salary']).set(title='Распределение зарплаты клиентов', xlabel='Зарпла
```



In [32]:
```
np.nanpercentile(df['salary'], [50,95,97,99])
```

Out[32]:
```
array([119658.105 , 316885.3445, 453350.7859, 887903.8546])
```

Медианная зарплата наших клиентов - 120 000 рублей

99% наших клиентов получают зарплату до **888 0000 рублей**

Посмотрим на уникальные значения поля `credit_card`, `churn` и `last_activity`

In [33]:
```
col = ['credit_card', 'churn', 'last_activity']
for i in col:
    print(f'Столбец - {i} - уникальные значения:')
    print(df[i].unique())
    print()
```

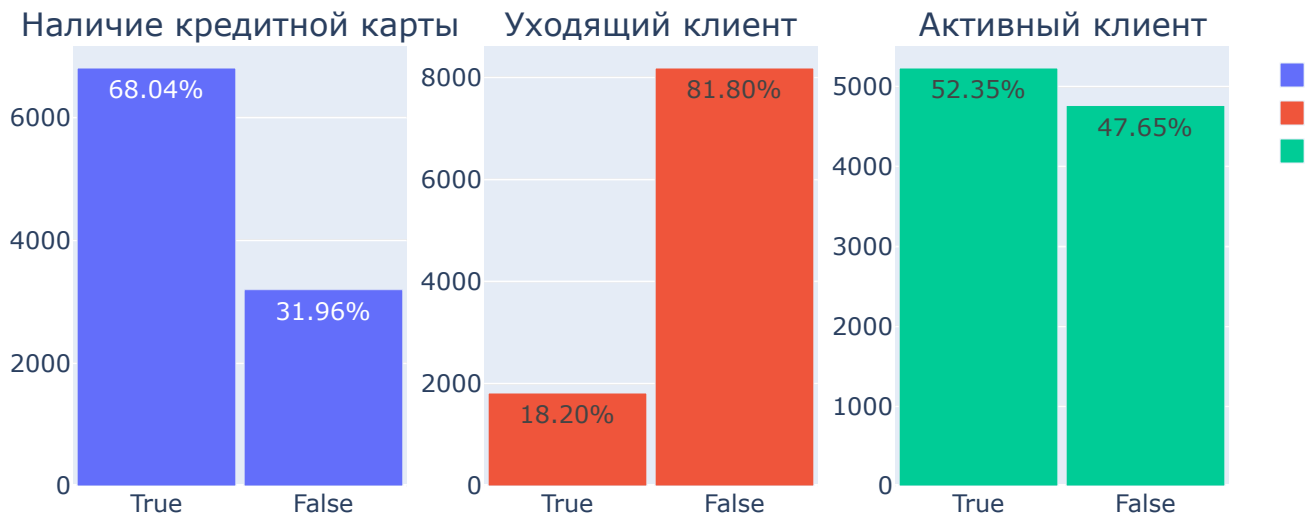
```
Столбец - credit_card - уникальные значения:
[0 1]

Столбец - churn - уникальные значения:
[1 0]

Столбец - last_activity - уникальные значения:
[1 0]
```

В столбцах - наличие кредитной карты, уходит клиент или нет и последняя активность - **значения 1 и 0**

https://community.plotly.com/t/how-to-visualize-3-columns-with-boolean-values/36181/2

In [34]:

```
fig = make_subplots(rows=1, cols=3, subplot_titles=('Наличие кредитной карты', 'Уходящий к
L= len(df)

cnames = ['credit_card', 'churn', 'last_activity']
for k, name in enumerate(cnames):
    n_true = df[name].sum()
    fig.add_trace(go.Bar(x=['True', 'False'], y=[n_true, L-n_true], name=name,
                         text=[f"{n_true/L*100:.2f}%", f"{(L-n_true)/L*100:.2f}%"]), 1,k+1)
fig.update_layout(barmode='relative',  bargap=0.05, width=800, height=400)
```



Как мы видим -

1) **У 68% наших клиентов есть кредитная карта**

2) **52.35% наших клиентов являются активными** (совершили действие за последний месяц)

3) **18.20 % наших клиентов являются уходящими**. Данную проблему нам и предстоит изучить.

Вывод по шагу №1

Мы провели предварительное знакомство с данными: 1) В нашем датасете 10 000 строк и 12 столбцов

2) Перевели все названия столбцов к `snowcase`

3) Пропуски есть в поле `balance` - 23% и поле `age` - 0,3%

4) В датасете присутствуют клиенты из 3 городов - Рыбинск, Ярославль, Ростов

5) У большинства клиентов - оценка собственности `equity` составляет - 0, либо от 1 до 5

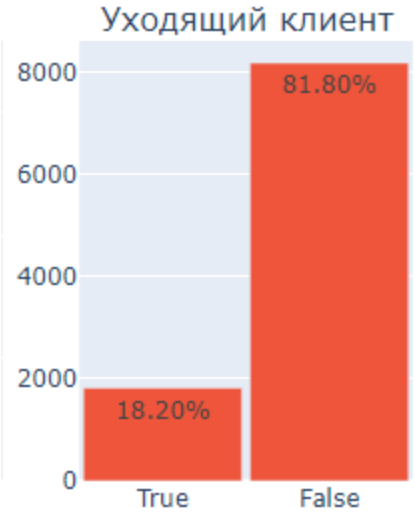
6) У большинства клиентов от 1 до 4 продуктов банка (поле - `products`)

7) Медианное значение баланса клиента - 524 000 рублей

8) Медианная зарплата клиента - 120 000 рублей

9) У 68% наших клиентов есть кредитная карта

10) **18 % нашего банка являются уходящими** (с этой проблемой нам и предстоит разобраться)



Шаг 2. Предобработка данных

Проверка типов данных

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   user_id        10000 non-null  int64  
 1   score          10000 non-null  int32  
 2   city           10000 non-null  object 
 3   gender         10000 non-null  object 
 4   age            9974 non-null   float64
 5   equity         10000 non-null  int64  
 6   balance        7705 non-null   float64
 7   products       10000 non-null  int64  
 8   credit_card    10000 non-null  int64  
 9   last_activity  10000 non-null  int64  
 10  salary         10000 non-null  float64
 11  churn          10000 non-null  int64  
dtypes: float64(3), int32(1), int64(6), object(2)
memory usage: 898.6+ KB
```

- Требуется перевести столбец `age` в `int` - но мы сможем это сделать, когда избавимся от пропусков
- Столбцы `last_activity`, `salary` и `churn` - по логике являются типом `boolean`, но для удобства расчетов - оставим их в типе данных `int` со значениями - 1 и 0

Поиск и удаление явных и неявных дубликатов

In [36]:
```
#Проверим датасет на дубликаты
```

```
print(f"Количество явных дубликатов в датафрейме - {df.duplicated().sum()}")
```

```
Количество явных дубликатов в датафрейме - 0
```

In [37]:
```
df[df.duplicated(subset=['user_id'],keep=False)].sort_values('user_id',ascending=False).he
```

Out[37]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary
6457	228075	839	Рыбинск	М	39.0	5	507199.85	3	0	1	85195.80
1247	228075	932	Ярославль	М	NaN	5	7601719.20	2	1	1	408121.16
8205	227795	840	Рыбинск	М	34.0	2	350768.03	1	1	0	102036.14
8497	227795	839	Ярославль	М	34.0	2	326593.14	2	1	0	103314.92
4216	226719	903	Рыбинск	Ж	63.0	0	NaN	1	1	0	138582.58
2597	226719	990	Ярославль	М	37.0	4	14648692.14	2	0	0	934412.61

In [38]:
```
print(f"Количество повторов поля user_id - {df.duplicated(['user_id']).sum()}")
```

```
Количество повторов поля user_id - 73
```

Повторы в `user_id` возникают из за того, что в **разных городах может быть клиент с одинаковым `user_id`**

In [39]:
```
df[df.duplicated(subset=['user_id','city'],keep=False)].sort_values('user_id',ascending=Fa
```

Out[39]:

user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	churn

Явных и неявных дубликатов в нашем датасете - нет.

Поиск и обработка пропусков в данных . Изучение природы пропусков

Посмотрим еще раз на пропуски -

- поле `balance` - 23% пропусков
- поле `age` - 0,3% пропусков

In [40]:
```
pd.DataFrame(round(df.isna().mean()*100,1))\
.sort_values(0,ascending=False).head(3).style.background_gradient('coolwarm')
```

Out[40]:

	0
balance	23.000000
age	0.300000
user_id	0.000000

Изучим природу пропусков в поле `balance` -

In [41]:
```
df[df['balance'].isna()].sort_values('user_id',ascending =False)
```

Out[41]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	chur
8913	229145	698	Ярославль	Ж	37.0	0	NaN	1	1	1	255439.00	

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	churn
2373	229054	883	Ярославль	М	27.0	0	NaN	1	1	1	144948.73	
8622	229052	921	Рыбинск	М	42.0	0	NaN	1	1	1	221661.10	
3798	229050	899	Ярославль	Ж	40.0	0	NaN	1	0	0	169445.35	
8026	229036	948	Рыбинск	Ж	43.0	0	NaN	1	0	0	241225.29	
...	...	...	...	...	...	...	...	...	...	...	...	
7017	114422	708	Ярославль	Ж	69.0	0	NaN	1	1	0	159013.27	
3980	114347	703	Ярославль	Ж	33.0	0	NaN	1	1	0	171038.71	
4580	114209	892	Ярославль	Ж	29.0	0	NaN	1	1	0	120174.24	
4025	114196	915	Рыбинск	М	31.0	0	NaN	1	1	1	133324.89	
1834	114182	890	Рыбинск	М	32.0	0	NaN	1	1	0	82847.95	

2295 rows × 12 columns

Сгруппируем данные по полю `equity` - приблизительная оценка собственности клиента

In [42]:
```
df[df['balance'].isna()].groupby('equity')['user_id'].count()
```

Out[42]:
```
equity
0    2180
1     114
3       1
Name: user_id, dtype: int64
```

Как мы видим, большинство пропусков в поле `balance` связано с отсутствием данных о собственности клиента.

Природа пропусков в поле `balance` - **MNAR (Missing Not At Random / Отсутствует не случайно)** — пропуски зависят от переменных, которых нет в данных, объяснить взаимосвязи с данными не получается, без дополнительного обоснования их нельзя отбрасывать или заполнять одним значением, т.к. это приведёт к заметным искажениям.

Во избежании искажения данных - заполнять поле `balance` не будем

Изучим природу пропусков в поле `age` -

In [43]:
```
df[df['age'].isna()].sort_values('user_id',ascending =False)
```

Out[43]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	cl
1247	228075	932	Ярославль	М	NaN	5	7601719.20	2	1	1	408121.16	
8070	226550	940	Рыбинск	М	NaN	0	NaN	1	0	1	147696.95	
9104	222480	776	Рыбинск	Ж	NaN	5	796735.09	1	1	1	55073.63	
9634	221809	917	Ярославль	М	NaN	0	NaN	1	1	1	192644.15	
8632	221197	893	Ярославль	М	NaN	0	NaN	1	1	0	173929.92	
2444	221156	913	Ярославль	М	NaN	0	NaN	1	1	1	135693.24	
7248	219343	920	Рыбинск	Ж	NaN	0	NaN	1	1	0	159248.67	

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	cl
5470	218868	827	Рыбинск	Ж	NaN	4	448959.07	2	1	1	67835.95	
8293	216848	930	Ярославль	М	NaN	0	NaN	1	1	1	199542.51	
7409	214031	777	Ярославль	М	NaN	2	171510.23	1	1	1	75409.63	
8449	210898	805	Ярославль	Ж	NaN	0	NaN	1	0	1	922080.25	
4912	210674	834	Рыбинск	М	NaN	1	238330.52	2	0	1	93775.06	
7236	210135	908	Рыбинск	Ж	NaN	4	1120340.31	3	1	1	85002.15	
8385	206759	915	Рыбинск	М	NaN	0	NaN	1	1	0	71179.53	
9301	202983	942	Рыбинск	Ж	NaN	0	NaN	1	1	1	163804.73	
8015	198635	670	Ярославль	Ж	NaN	0	NaN	1	1	1	168699.33	
2165	187635	692	Рыбинск	Ж	NaN	0	NaN	1	1	1	160368.82	
9380	187459	894	Рыбинск	М	NaN	0	NaN	1	1	0	178012.28	
9632	185829	927	Ярославль	М	NaN	0	NaN	1	1	0	231254.86	
7345	184913	829	Ярославль	Ж	NaN	3	188648.77	2	0	1	75206.90	
9667	163657	849	Ярославль	М	NaN	4	1254013.85	2	1	1	119106.67	
5495	151662	884	Рыбинск	Ж	NaN	0	NaN	1	1	1	137500.77	
9457	141945	929	Ярославль	М	NaN	0	NaN	1	1	0	381868.89	
9819	140934	832	Рыбинск	Ж	NaN	3	385763.16	2	0	1	59651.35	
3091	138660	836	Ростов	Ж	NaN	5	294315.53	2	0	1	63310.22	
8785	127440	663	Ярославль	М	NaN	0	NaN	1	1	1	117197.56	

Логику появления пропусков в поле `age` - объяснить мы не можем. Как мы видим, в строках с отсутствующим значением `age` есть также пропуски в поле `balance`, которые мы решили оставить.

In [44]:
```
df[df['age'].isna()&df['balance'].isna()].sort_values('salary',ascending =False)
```

Out[44]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	chur
8449	210898	805	Ярославль	Ж	NaN	0	NaN	1	0	1	922080.25	
9457	141945	929	Ярославль	М	NaN	0	NaN	1	1	0	381868.89	
9632	185829	927	Ярославль	М	NaN	0	NaN	1	1	0	231254.86	
8293	216848	930	Ярославль	М	NaN	0	NaN	1	1	1	199542.51	
9634	221809	917	Ярославль	М	NaN	0	NaN	1	1	1	192644.15	
9380	187459	894	Рыбинск	М	NaN	0	NaN	1	1	0	178012.28	
8632	221197	893	Ярославль	М	NaN	0	NaN	1	1	0	173929.92	
8015	198635	670	Ярославль	Ж	NaN	0	NaN	1	1	1	168699.33	
9301	202983	942	Рыбинск	Ж	NaN	0	NaN	1	1	1	163804.73	
2165	187635	692	Рыбинск	Ж	NaN	0	NaN	1	1	1	160368.82	
7248	219343	920	Рыбинск	Ж	NaN	0	NaN	1	1	0	159248.67	
8070	226550	940	Рыбинск	М	NaN	0	NaN	1	0	1	147696.95	

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	chur
5495	151662	884	Рыбинск	Ж	NaN	0	NaN	1	1	1	137500.77	
2444	221156	913	Ярославль	М	NaN	0	NaN	1	1	1	135693.24	
8785	127440	663	Ярославль	М	NaN	0	NaN	1	1	1	117197.56	
8385	206759	915	Рыбинск	М	NaN	0	NaN	1	1	0	71179.53	

Как мы видим, у большинства клиентов нашего банка, у которых отсутствуют данные в полях баланс и возраст - один продукт нашего банка и это кредитная карта.

Характер пропусков - **MCAR (Missing Completely At Random / Отсутствует совершенно случайно)** — пропуски не зависят от переменных и не ведут к систематической ошибке (смещение, bias), но увеличивают случайную ошибку (разброс, variance), можно их отбрасывать или заполнять, т.к. заполнение не повлияет на характер связи между заполняемой переменной и остальными.

Зполним для таких клиентов значение `age` и `balance` - медианным значениям по похожим клиентам.

In [45]:
```
df.loc[(
    df['age'].isna()
    &df['balance'].isna()
    &df['credit_card']==1),'balance'] = df[df['age'].isna()&df['balance'].isna()&df['credi
.fillna(df.query('credit_card==1 and products==1')['balance'].median())
```

In [46]:
```
df.loc[(
    df['age'].isna()
    &df['balance'].isna()
    &df['credit_card']==1),'age'] = df[df['age'].isna()&df['balance'].isna()&df['credit_ca
.fillna(df.query('credit_card==1 and products==1')['age'].median())
```

In [47]:
```
df[df['age'].isna()&df['balance'].isna()].sort_values('salary',ascending =False)
```

Out[47]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	chur
8449	210898	805	Ярославль	Ж	NaN	0	NaN	1	0	1	922080.25	
8070	226550	940	Рыбинск	М	NaN	0	NaN	1	0	1	147696.95	

Посмотрим на пропуски в поле `age` в строках, где нет пропусков в поле `balance`

In [48]:
```
df[df['age'].isna()&df['balance'].notna()].sort_values('balance',ascending =False)
```

Out[48]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	cl
1247	228075	932	Ярославль	М	NaN	5	7601719.20	2	1	1	408121.16	
9667	163657	849	Ярославль	М	NaN	4	1254013.85	2	1	1	119106.67	
7236	210135	908	Рыбинск	Ж	NaN	4	1120340.31	3	1	1	85002.15	
9104	222480	776	Рыбинск	Ж	NaN	5	796735.09	1	1	1	55073.63	
5470	218868	827	Рыбинск	Ж	NaN	4	448959.07	2	1	1	67835.95	
8385	206759	915	Рыбинск	М	NaN	0	427024.50	1	1	0	71179.53	
9634	221809	917	Ярославль	М	NaN	0	427024.50	1	1	1	192644.15	

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	cl
9632	185829	927	Ярославль	М	NaN	0	427024.50	1	1	0	231254.86	
9457	141945	929	Ярославль	М	NaN	0	427024.50	1	1	0	381868.89	
9380	187459	894	Рыбинск	М	NaN	0	427024.50	1	1	0	178012.28	
9301	202983	942	Рыбинск	Ж	NaN	0	427024.50	1	1	1	163804.73	
8785	127440	663	Ярославль	М	NaN	0	427024.50	1	1	1	117197.56	
8632	221197	893	Ярославль	М	NaN	0	427024.50	1	1	0	173929.92	
8293	216848	930	Ярославль	М	NaN	0	427024.50	1	1	1	199542.51	
2165	187635	692	Рыбинск	Ж	NaN	0	427024.50	1	1	1	160368.82	
8015	198635	670	Ярославль	Ж	NaN	0	427024.50	1	1	1	168699.33	
7248	219343	920	Рыбинск	Ж	NaN	0	427024.50	1	1	0	159248.67	
5495	151662	884	Рыбинск	Ж	NaN	0	427024.50	1	1	1	137500.77	
2444	221156	913	Ярославль	М	NaN	0	427024.50	1	1	1	135693.24	
9819	140934	832	Рыбинск	Ж	NaN	3	385763.16	2	0	1	59651.35	
3091	138660	836	Ростов	Ж	NaN	5	294315.53	2	0	1	63310.22	
4912	210674	834	Рыбинск	М	NaN	1	238330.52	2	0	1	93775.06	
7345	184913	829	Ярославль	Ж	NaN	3	188648.77	2	0	1	75206.90	
7409	214031	777	Ярославль	М	NaN	2	171510.23	1	1	1	75409.63	

Пропуски в поле age есть у клиентов из разных городов, с разным уровнем зарплаты и балансов лицевого счета .

Природа пропусков в поле age - **MAR (Missing At Random / Отсутствует случайно)** — в рамках каждой из групп, которая есть в описываемой данными совокупности, распределение пропусков случайно, можно их отбросить.

Но мы их оставим для дальнейшего исследования.

Создание новых столбцов для анализа, в т.ч. кодирование категориальных столбцов (пол, город)

Закодируем столбцы с полом и городом для дальнейшего построения корреляций.

In [49]:
```
df['gender_code'] = df ['gender']
df['city_code'] = df ['city']
df = pd.get_dummies (df, columns=['gender_code', 'city_code'], drop_first= False )
df = df.rename(columns={'gender_code_Ж': 'female', 'gender_code_М': 'male','city_code_Рост
                        'city_code_Рыбинск':'Рыбинск','city_code_Ярославль':'Ярославль'})

df.head()
```

Out[49]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	churn
0	183012	850	Рыбинск	Ж	25.0	1	59214.82	2	0	1	75719.14	1
1	146556	861	Рыбинск	Ж	37.0	5	850594.33	3	1	0	86621.77	0
2	120722	892	Рыбинск	Ж	30.0	0	NaN	1	1	1	107683.34	0

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	churn
3	225363	866	Ярославль	Ж	51.0	5	1524746.26	2	0	1	174423.53	1
4	157978	730	Ярославль	М	34.0	5	174.00	1	1	0	67353.16	1

Добавим столбец с количеством продуктов (без кредитной карты)

In [50]:
```
df['products_wo_credit'] = df['products']-df['credit_card']
df.head()
```

Out[50]:

	user_id	score	city	gender	age	equity	balance	products	credit_card	last_activity	salary	churn
0	183012	850	Рыбинск	Ж	25.0	1	59214.82	2	0	1	75719.14	1
1	146556	861	Рыбинск	Ж	37.0	5	850594.33	3	1	0	86621.77	0
2	120722	892	Рыбинск	Ж	30.0	0	NaN	1	1	1	107683.34	0
3	225363	866	Ярославль	Ж	51.0	5	1524746.26	2	0	1	174423.53	1
4	157978	730	Ярославль	М	34.0	5	174.00	1	1	0	67353.16	1

Вывод по Шагу 2. Предобработка данных

1) Изменены типы данных у столбцов

2) Явных и неявных дубликатов не обнаружено. (мы обнаружили, что **в разных городах может быть клиент с одинаковым `user_id`**)

3) Большинство пропусков в поле `balance` связано с отсутствием данных о собственности клиента.Во избежании искажения данных - заполнять поле `balance` не будем

4) Природа пропусков в поле `age` - **MAR (Missing At Random / Отсутствует случайно)** — в рамках каждой из групп, которая есть в описываемой данными совокупности, распределение пропусков случайно, можно их отбросить, но мы их оставим для дальнейшего исследования.

5) Мы закодировали категориальные переменные в отдельные столбцы - поле `city` и `gender`

6) Добавили столбец с количеством продуктов (без кредитной карты) - `products_wo_credit`

Шаг 3. Исследовательский анализ данных

Посчитаем процент отточных клиентов в нашем датасете

In [51]:
```
print(f"Количество отточных клиентов - {df['churn'].mean()*100}%")
```

```
Количество отточных клиентов - 18.2%
```

Посмотрим на зависимость оттока пользователей от возраста клиента

In [52]:
```
df['churn']. corr(df['age'])
```

Out[52]:
```
-0.049796603717276955
```

In [53]:

```
plt.figure(figsize=(8, 5))
ax = sns.kdeplot (df['age'],common_norm=False,hue=df['churn'], bw_method =0.07)
ax.grid(True)
ax.set_xlim(18, 90)
ax.set(title='Зависимость оттока от возраста клиента)', xlabel='Возраст клиентов')
plt.show()
```

Судя по графику,

- Большой отток у молодых клиентов от 25 до 34 лет
- Резкий пик оттока у клиентов от 50 до 60 лет

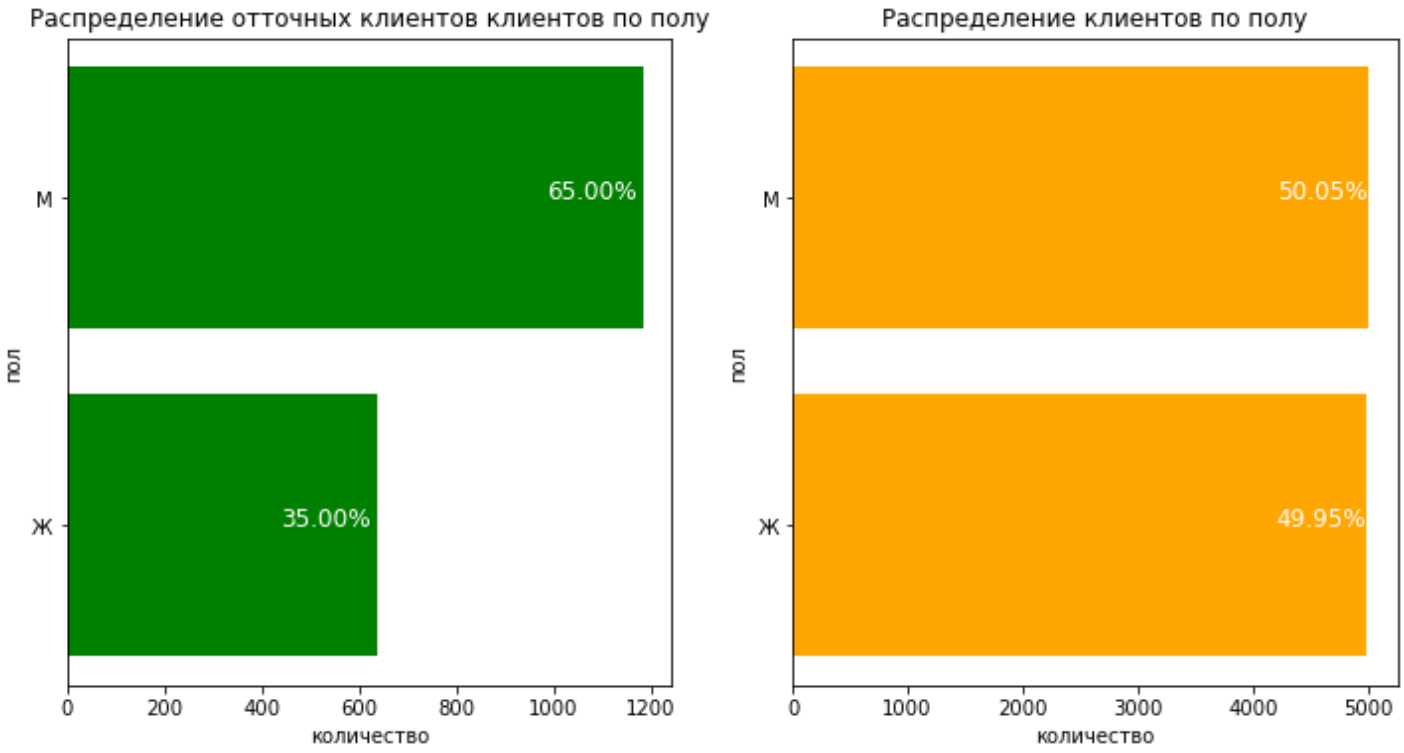
Посмотрим на зависимость оттока пользователей от пола клиента

In [54]:
```
df.query('churn==1').groupby('gender')['user_id'].count()
```

Out[54]:
```
gender
Ж     637
М    1183
Name: user_id, dtype: int64
```

In [55]:
```
# задаем данные
total_churn = df.query('churn==1').groupby('gender')['user_id'].count().sum()
total_clients = df.groupby('gender')['user_id'].count().sum()

# создаем subplot
fig, axs = plt.subplots(1, 2, figsize=(12, 6))

# первый график - распределение отточных клиентов по полу
axs[0].barh(df.query('churn==1').groupby('gender')['user_id'].count().index,
            df.query('churn==1').groupby('gender')['user_id'].count().values, color='green
axs[0].set_xlabel('количество')
axs[0].set_ylabel('пол')
axs[0].set_title('Распределение отточных клиентов клиентов по полу')

for i, v in enumerate(df.query('churn==1').groupby('gender')['user_id'].count().values):
    axs[0].text(v-200, i, f"{v/total_churn*100:.2f}%", color='white', fontsize=12)

# второй график - распределение всех клиентов по полу
axs[1].barh(df.groupby('gender')['user_id'].count().index,
            df.groupby('gender')['user_id'].count().values, color='orange')
```

```
axs[1].set_xlabel('количество')
axs[1].set_ylabel('пол')
axs[1].set_title('Распределение клиентов по полу')

for i, v in enumerate(df.groupby('gender')['user_id'].count().values):
    axs[1].text(v-800, i, f"{v/total_clients*100:.2f}%", color='white', fontsize=12)

# отображаем графики
plt.show()
```



65% отточных клиентов - мужчины, при том что, изначально клиентов банка - мужчин и женщин - у нас поровну.

Посмотрим на зависимость оттока клиентов от наличия кредитной карты

In [56]:
```
df.query('churn==1').groupby('credit_card')['user_id'].count()
```

Out[56]:
```
credit_card
0     817
1    1003
Name: user_id, dtype: int64
```

In [57]:
```
total = df.query('churn==1').groupby('credit_card')['user_id'].count().sum()

ax = df.query('churn==1').groupby('credit_card')['user_id'].count().plot(kind='barh', colo
ax.set_xlabel('количество')
ax.set_ylabel('Наличие кредитной карты')
ax.set_title('Распределение отточных клиентов по наличию кредитной карты')

for i, v in enumerate(df.query('churn==1').groupby('credit_card')['user_id'].count().value
    ax.text(v-200,i, f"{v/total*100:.2f}%", color='white', fontsize=12)

plt.show()
```

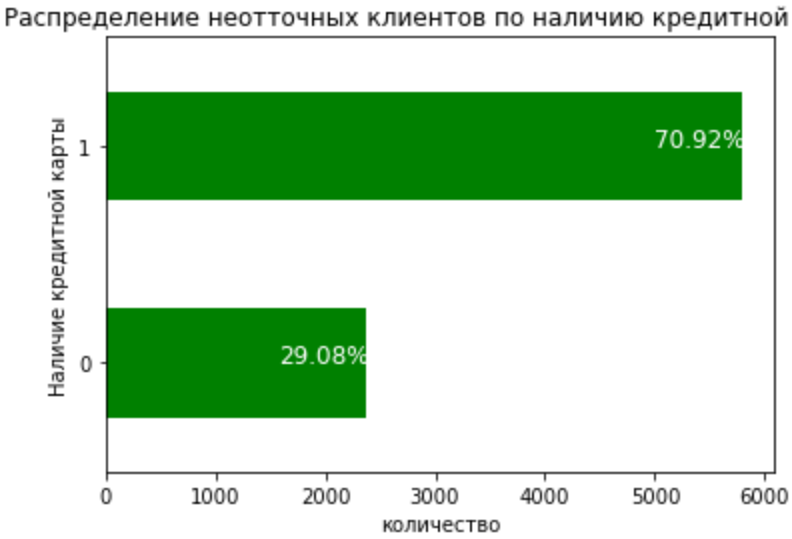
У **отточных клиентов процент наличия кредитной карты составляет - 55,11%**

Посмотрим на **процент наличия кредитной карты у неотточных клиентов**

In [58]:
```
df.query('churn==0').groupby('credit_card')['user_id'].count()
```

Out[58]:
```
credit_card
0    2379
1    5801
Name: user_id, dtype: int64
```

In [59]:
```
total = df.query('churn==0').groupby('credit_card')['user_id'].count().sum()

ax = df.query('churn==0').groupby('credit_card')['user_id'].count().plot(kind='barh', colo
ax.set_xlabel('количество')
ax.set_ylabel('Наличие кредитной карты')
ax.set_title('Распределение неотточных клиентов по наличию кредитной карты')

for i, v in enumerate(df.query('churn==0').groupby('credit_card')['user_id'].count().value
    ax.text(v-800,i, f"{v/total*100:.2f}%", color='white', fontsize=12)

plt.show()
```

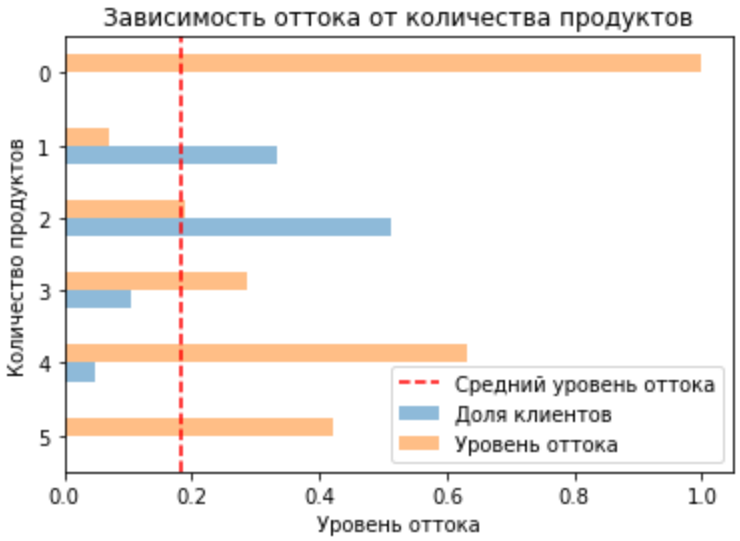


У неотточных - процент наличия кредитной карты больше. 71% против 29% .

Запомним это и в дальнейшем порекомендуем менеджерам чаще предлагать кредитные карты нашим клиентам.

Посмотрим на зависимость оттока клиента от количества продуктов, которые использует клиент

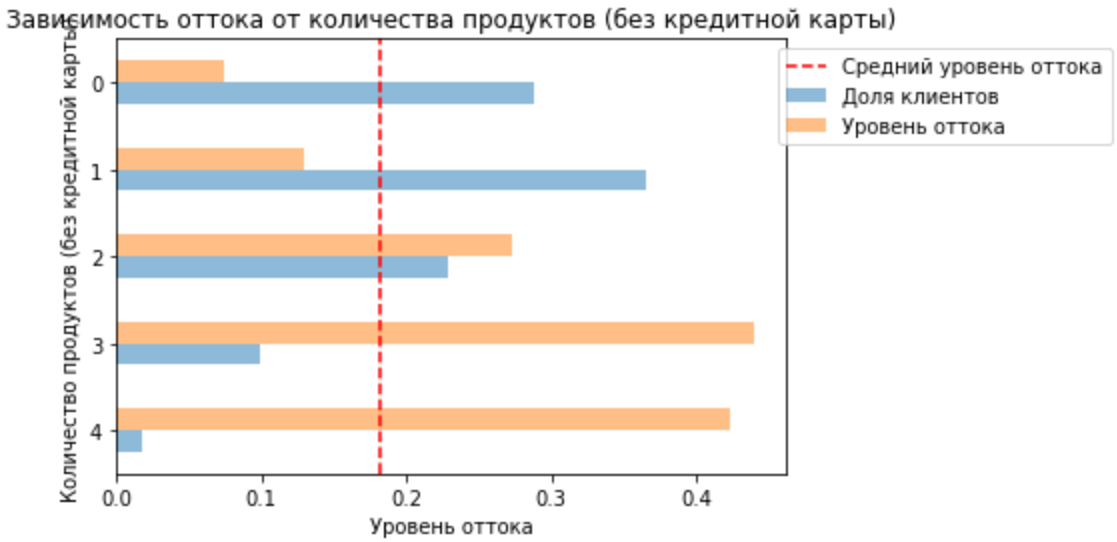
In [60]:
```
chunk = df.groupby('products').agg({"churn":["count","mean"]})\
.sort_index(ascending=False)\
                .droplevel(0,axis=1)\
                .set_axis(["Доля клиентов","Уровень оттока"],axis=1)
chunk["Доля клиентов"] = chunk["Доля клиентов"] / chunk["Доля клиентов"].sum()
#chunk.index = [str(idx) + "products" + col for idx in chunk.index]
ax = chunk.plot(kind="barh",alpha=.5)
ax.set(title='Зависимость оттока от количества продуктов', xlabel='Уровень оттока',ylabel=
plt.axvline(df.churn.mean(),color="r",label="Средний уровень оттока", linestyle='--')
plt.legend()
plt.show()
```



- Судя по графику - **Большинство отточных клиентов - клиенты владеющие не менее 3мя банковскими продуктами**. С увеличением количества продуктов - процент отточных клиентов увеличивается.

Построим график без учета кредитных карт

In [61]:
```
chunk = df.groupby('products_wo_credit').agg({"churn":["count","mean"]})\
.sort_index(ascending=False)\
                .droplevel(0,axis=1)\
                .set_axis(["Доля клиентов","Уровень оттока"],axis=1)
chunk["Доля клиентов"] = chunk["Доля клиентов"] / chunk["Доля клиентов"].sum()
ax = chunk.plot(kind="barh",alpha=.5)
ax.set(title='Зависимость оттока от количества продуктов (без кредитной карты)',
       xlabel='Уровень оттока',ylabel='Количество продуктов (без кредитной карты)')
plt.axvline(df.churn.mean(),color="r",label="Средний уровень оттока", linestyle='--')
plt.legend(loc='upper right',bbox_to_anchor=(1.5, 1))
plt.show()
```



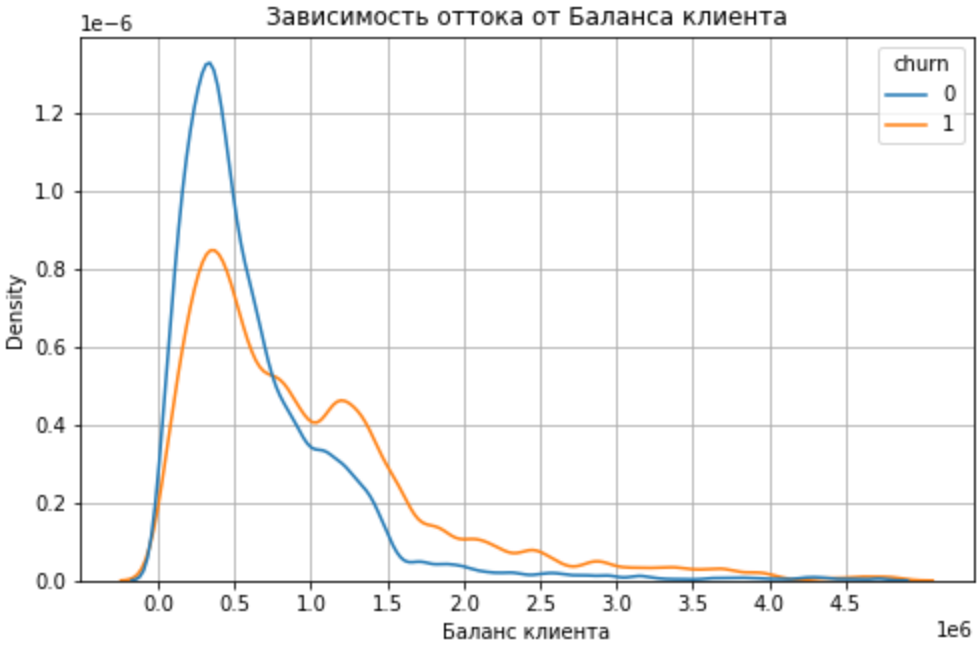
На данном графике явно виден отток клиентов, у которых 2 и более банковских продукта (без учета кредитной карты)

Посмотрим на зависимость оттока клиентов от баланса на счете

Построим график зависимости, ограничив баланс 99 - перцентилем (для удаления выбросов)

In [62]:

```
plt.figure(figsize=(8, 5))
ax = sns.kdeplot (df[df['balance']<np.nanpercentile(df['balance'], 99)]['balance'],common_
ax.set(title='Зависимость оттока от Баланса клиента', xlabel='Баланс клиента')
ax.grid(True)
ax.set_xticks(range(0, 5000000, 500000))
plt.show()
```



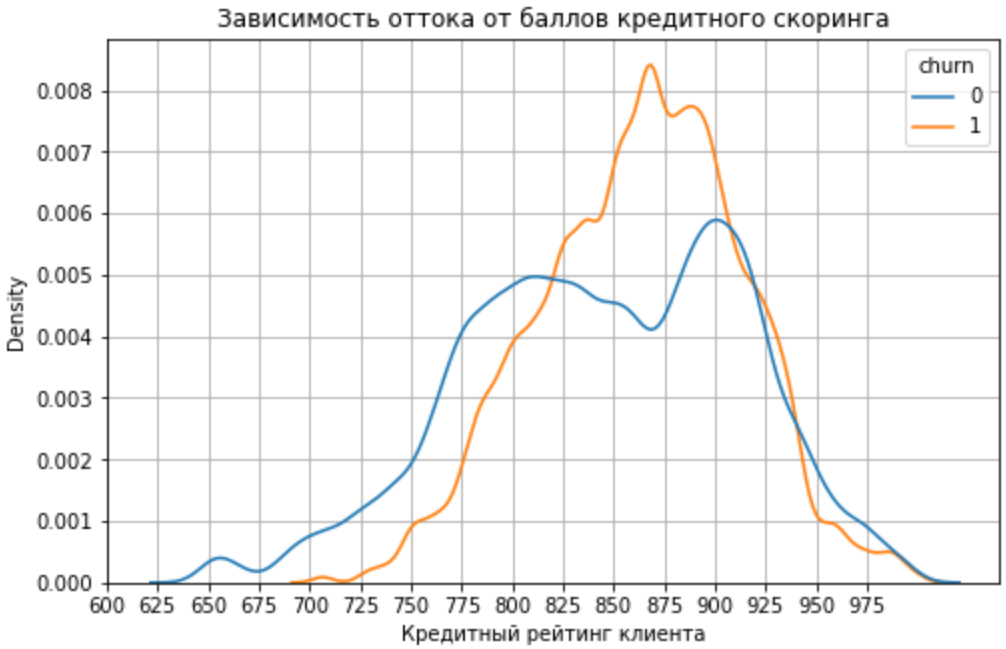
Как мы видим, очень **у большого процента отточных клиентов - баланс составляет от 750 000 до 4 000 000 рублей**

Изучим зависимость оттока клиентов от баллов кредитного скоринга

In [63]:

```
plt.figure(figsize=(8, 5))
ax = sns.kdeplot (df['score'],common_norm=False,hue=df['churn'], bw_method =0.1)
ax.set(title='Зависимость оттока от баллов кредитного скоринга', xlabel='Кредитный рейтинг
```

```
ax.grid(True)
ax.set_xticks(range(600, 1000, 25))
plt.show()
```

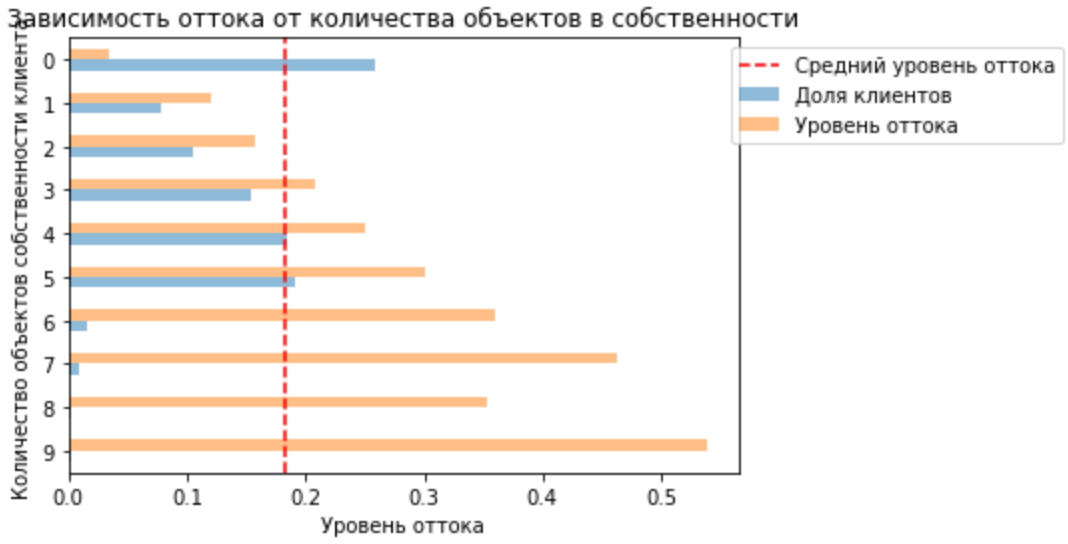


Большой всплеск оттока наблюдается у клиентов с кредитным рейтингом от 810 до 910.

Изучим зависимость оттока клиентов от количества объектов в собственности

In [64]:

```
chunk = df.groupby('equity').agg({"churn":["count","mean"]})\
.sort_index(ascending=False)\
                .droplevel(0,axis=1)\
                .set_axis(["Доля клиентов","Уровень оттока"],axis=1)
chunk["Доля клиентов"] = chunk["Доля клиентов"] / chunk["Доля клиентов"].sum()
ax = chunk.plot(kind="barh",alpha=.5)
ax.set(title='Зависимость оттока от количества объектов в собственности',
       xlabel='Уровень оттока',ylabel='Количество объектов собственности клиента')
plt.axvline(df.churn.mean(),color="r",label="Средний уровень оттока", linestyle='--')
plt.legend(loc='upper right',bbox_to_anchor=(1.5, 1))
plt.show()
```



Наибольший отток наблюдается у клиентов, у которых в собственности от 3 до 9 объектов недвижимости.

С увеличением числа объектов собственности - уровень оттока увеличивается

Посмотрим на зависимость города клиента на отток клиентов

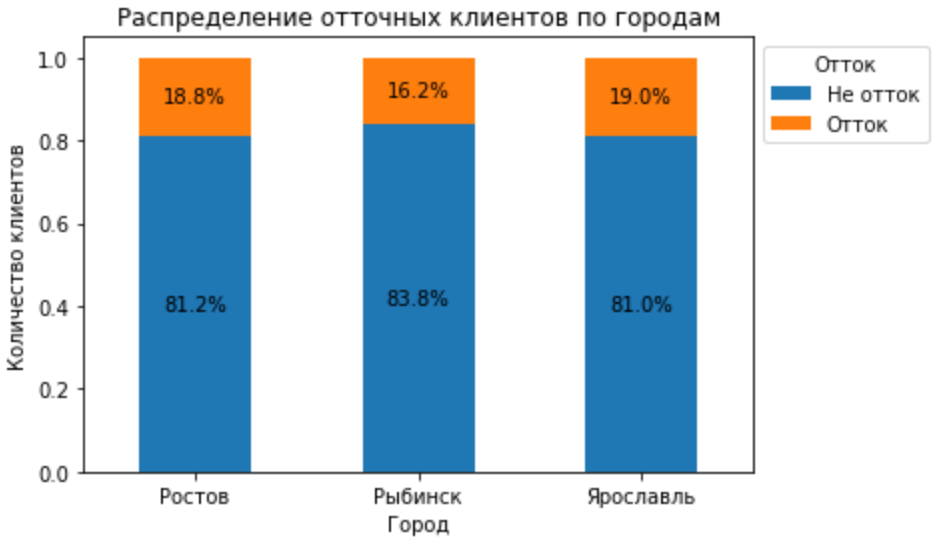
In [65]:
```
df.groupby('city')['churn'].value_counts().unstack(fill_value=0)
```

Out[65]:

churn	0	1
city		
Ростов	1151	266
Рыбинск	2258	437
Ярославль	4771	1117

In [66]:
```
grouped = df.groupby('city')['churn'].value_counts(normalize = True).unstack(fill_value=0)

# Построение столбчатой диаграммы
plt.figure(figsize=(12, 5))
ax = grouped.plot.bar(stacked=True)
ax.set_xlabel('Город')
ax.set_ylabel('Количество клиентов')
ax.set_title('Распределение отточных клиентов по городам')
ax.legend(title='Отток', labels=['Не отток', 'Отток'], bbox_to_anchor=(1,1))
ax.set_xticklabels(grouped.index, rotation=0)

# Добавим проценты на график
for container in ax.containers:
    ax.bar_label(
        container, label_type='center',
        labels=[f"{round(val*100,1)}%" for val in container.datavalues]
    )

plt.show()
```

```
<Figure size 864x360 with 0 Axes>
```



Как мы видим, процент отточных клиентов по каждому городу примерно одинаковый -

- Ярославль - 19 % отточных клиентов
- Ростов - 18,8% отточных клиентов
- Рыбинск - 16,2% отточных клиентов

Посмотрим зависимость зарплаты клиента, от города пользователя

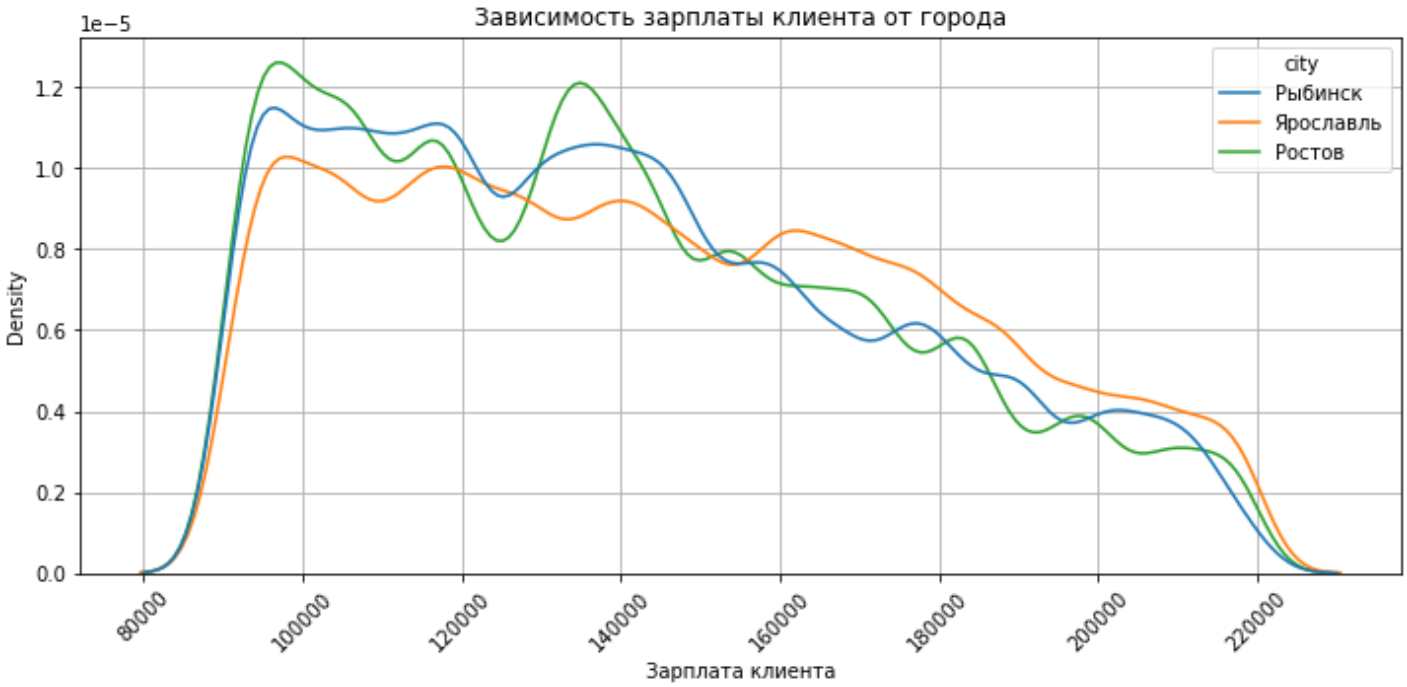
In [67]:
```
plt.figure(figsize=(12, 5))
ax = sns.kdeplot (df[df['salary']< np.nanpercentile(df['salary'], 99)]['salary'],common_no
ax.set(title='Зависимость оттока от зарплаты клиента', xlabel='Зарплата клиента')
ax.grid(True)
ax.set_xticks(range(0, 1000000, 50000))
plt.xticks(rotation = 45)
plt.show()
```

Большой отток наблюдается у наших клиентов с зарплатой от 90 000 до 220 000 рублей.

Посмотрим зависимость зарплаты клиента, от города пользователя для целевой группы клиентов

Рассмотрим зарплату от 90 000 до 220 000 рублей

In [68]:
```
plt.figure(figsize=(12, 5))
ax = sns.kdeplot (df[df['salary'].between(90000,220000)]['salary'],common_norm=False,hue=c
ax.set(title='Зависимость зарплаты клиента от города', xlabel='Зарплата клиента')
ax.grid(True)
plt.xticks(rotation = 45)
plt.show()
```

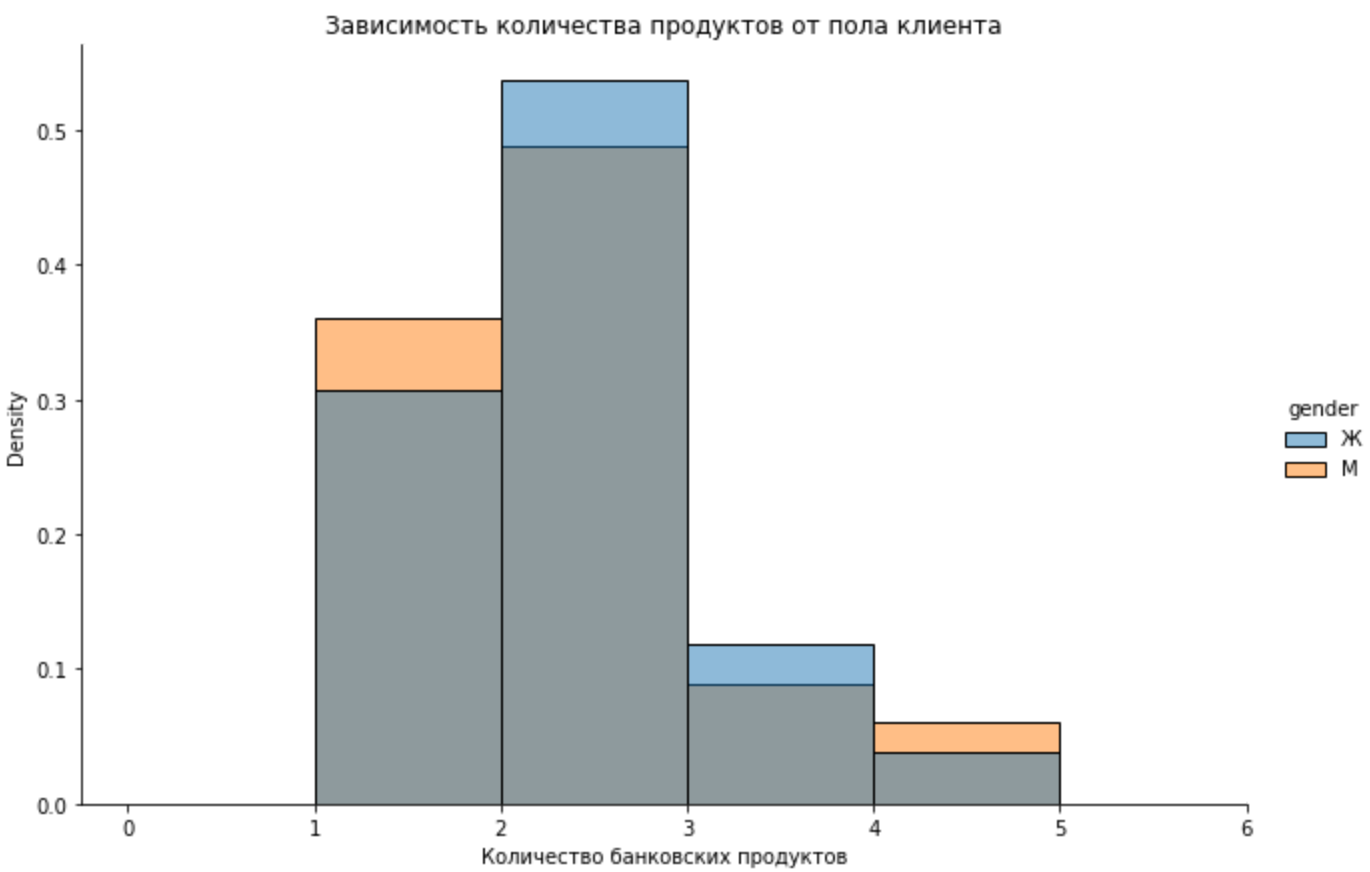


- Зарплата от 90 000 до 150 000 рублей по большой части у жителей Ростова и Рыбинска
- От 150 000 рублей у жителей Ярославля.

Посмотрим на зависимость пола клиента на количество продуктов, которыми пользуется клиент

In [69]:

```
ax = sns.displot(
    data=df, x='products', hue='gender', common_norm=False, stat="density",binwidth=1,
   height=6, aspect=1.5
)
ax.set(title='Зависимость количества продуктов от пола клиента',
       xlabel='Количество банковских продуктов',
      xticks=np.arange(df['products'].min(), df['products'].max() + 2, 1))
plt.show()
```



Судя по графику плотности распределения -

- У мужчин чаще всего бывает 1 или 4 продукта
- У женщин - 2 и 3 продукта

Посмотрим на зависимость баланса на счете клиентов, от города

Посмотрим на города клиентов, у которых баланс лицевого счета составляет от 750 000 до 4 000 000 рублей

In [70]:
```
df[df['balance'].between(750000,4000000)].groupby('city')['user_id'].count()\
.sort_values(ascending = False).reset_index(drop=False)
```

Out[70]:

	city	user_id
0	Ярославль	1543
1	Рыбинск	697
2	Ростов	352

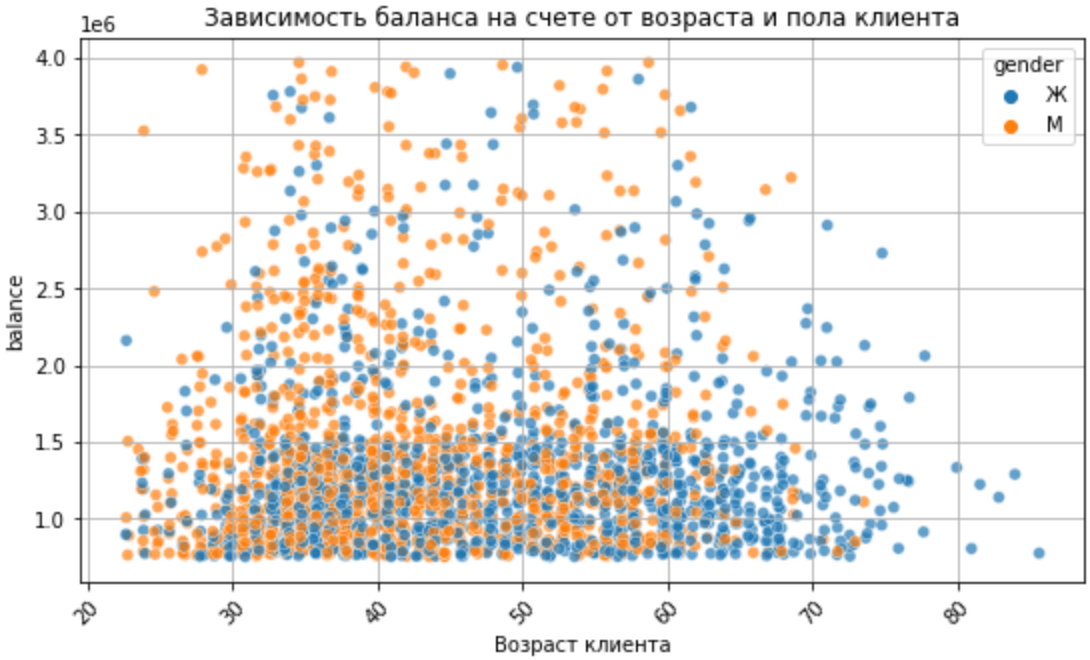
Большинство клиентов в абсолютном значении с указанным балансом, проживают в Ярославле

Посмотрим на зависимость баланса на счете от возраста и пола клиента

Для отточного сегмента с балансом от 750 000 до 4 000 000 рублей

In [71]:
```
plt.figure(figsize=(9, 5))
ax = sns.scatterplot(
    data=df.assign(age=df.age - np.random.random(df.shape[0])/2)\
    .query('balance < 4000000 & balance > 750000'),x='age',y='balance',hue = 'gender',alph
)
```

```
ax.set(title='Зависимость баланса на счете от возраста и пола клиента', xlabel='Возраст кл
ax.grid(True)
plt.xticks(rotation = 45)
plt.show()
```



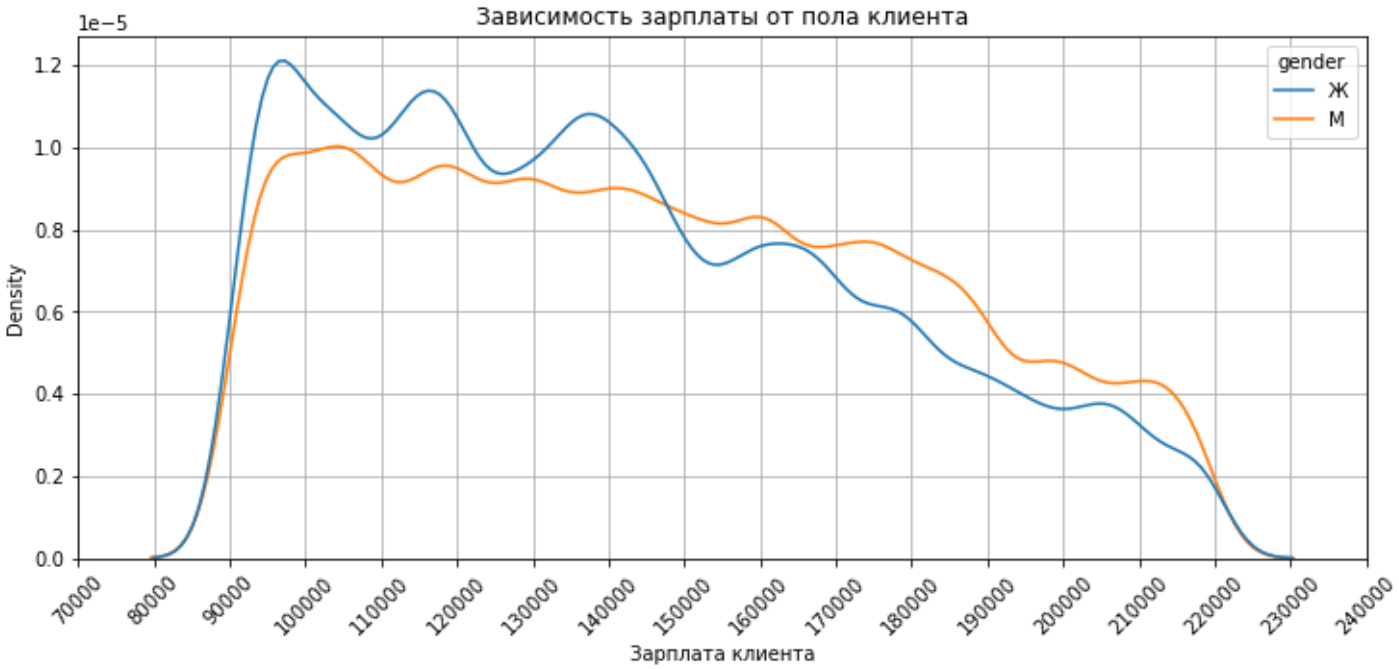
Как мы видим, указанный баланс чаще встречается у мужчин в возрасте от 18 до 60 лет, а после 60ти лет - перевес у женщин.

Посмотрим на зависимость зарплаты на счете от возраста и пола клиента

Для отточного сегмента с зарплатой от 90 000 до 220 000 рублей

In [72]:

```
plt.figure(figsize=(12, 5))
ax = sns.kdeplot (df.query('salary < 220000 & salary > 90000')['salary'],
                  common_norm=False,
                  hue=df.query('salary < 220000 & salary > 90000')['gender'],
                  bw_method =0.1)
ax.set(title='Зависимость зарплаты от пола клиента', xlabel='Зарплата клиента')
ax.grid(True)
plt.xticks(rotation = 45)
ax.set_xticks(range(70000, 250000, 10000))
plt.show()
```

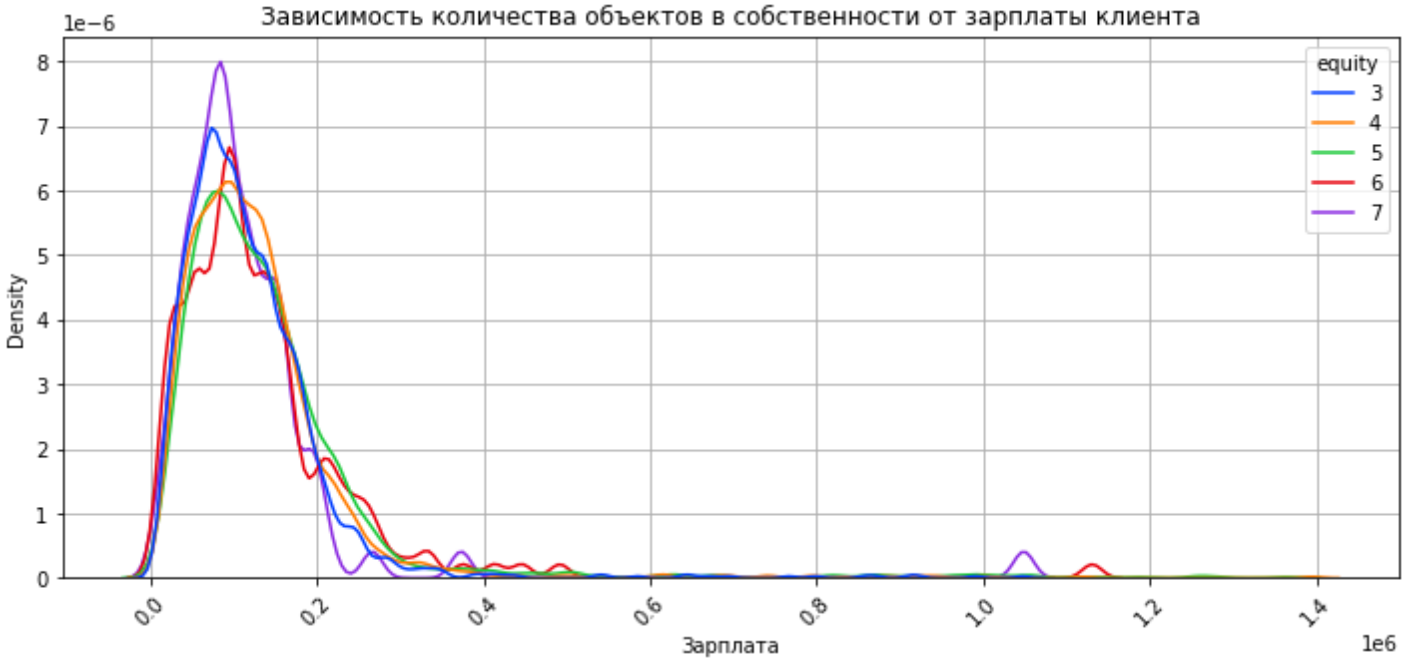


- В нашем датасете больше женщин с заработной платой от 90 000 до 145 000 рублей
- После зарплаты 150 000 до зарплаты 220 000 идет перевес мужского населения

Посмотрим на зависимость количества объектов в собственности у клиента от заработной платы

Нас интересует отточный сегмент - с числом объектов собственности от 3 до 7

In [73]:
```
plt.figure(figsize=(12, 5))
ax = sns.kdeplot (df.query('equity <= 7 & equity >= 3')['salary'],
                  common_norm=False,
                  hue=df.query('equity <= 7 & equity >= 3')['equity'],
                  bw_method =0.1,
                 palette='bright')
ax.set(title='Зависимость количества объектов в собственности от зарплаты клиента', xlabel
ax.grid(True)
plt.xticks(rotation = 45)
plt.show()
```



В отточном сегменте с зарплатой от 90 000 до 220 000 рублей чаще всего встречаются клиенты с количеством объектов собственности - 3,6,7

Изучим зависимость баллов кредитного скоринга клиента от его возраста и пола

Изучим отточный сегмент с кредитным рейтингом от 810 до 910

In [74]:
```
# создание графика
plt.figure(figsize=(12, 5))
ax = sns.scatterplot(
    data=df.assign(age=df.age - np.random.random(df.shape[0])/2).query('score > 810 and sc
    x='age', y='score',hue = 'gender',alpha=.45,size=1)

# настройка осей и заголовка
ax.set(
    title='Распределение балов кредитного скоринга клиента от его возраста и пола',
    xlabel='Возраст клиента',
    ylabel='Кредитный скоринг клиента'
      )
ax.grid(True)

# отображение графика
plt.show()
```

В нашей отточной категории клиентов с баллами кредитного скоринга от 810 до 910 -

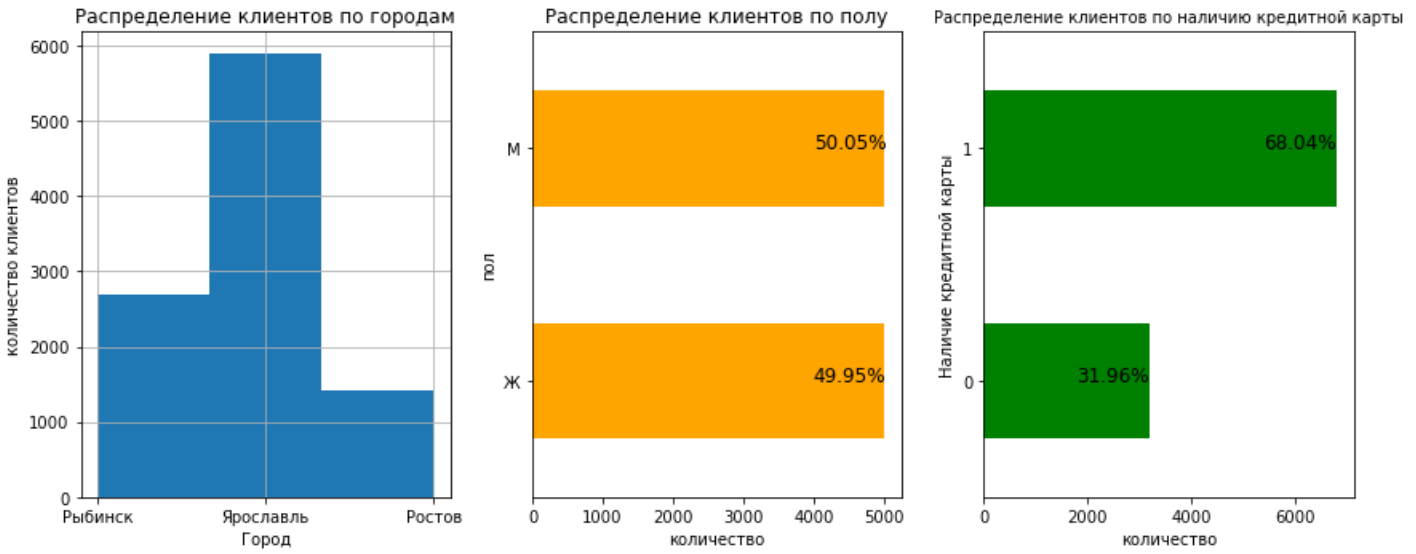
- в возрасте до 50 лет - чаще встречаются мужчины
- в возрасте после 60 лет - чаще встречаются женщины

Построим "портрет" среднестатистического клиента банка

Построим "портрет" среднестатистического клиента банка

In [75]:
```
def portrait(df,text1=1000,text2=1400):
    fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(12,5))

    df['city'].hist(bins=3, ax=ax1)
    ax1.set(xlabel='Город', ylabel='количество клиентов', title='Распределение клиентов по
```

```
    total = df.groupby('gender')['user_id'].count().sum()
    ax2 = df.groupby('gender')['user_id'].count().plot(kind='barh', color='orange', ax=ax2
    ax2.set_xlabel('количество')
    ax2.set_ylabel('пол')
    ax2.set_title('Распределение клиентов по полу')

    for i, v in enumerate(df.groupby('gender')['user_id'].count().values):
        ax2.text(v-text1,i, f"{v/total*100:.2f}%", color='black', fontsize=12)

    total = df.groupby('credit_card')['user_id'].count().sum()
    ax3 = df.groupby('credit_card')['user_id'].count().plot(kind='barh', color='green', ax
    ax3.set_xlabel('количество')
    ax3.set_ylabel('Наличие кредитной карты')
    ax3.set_title('Распределение клиентов по наличию кредитной карты',fontsize=10)

    for i, v in enumerate(df.groupby('credit_card')['user_id'].count().values):
        ax3.text(v-text2,i, f"{v/total*100:.2f}%", color='black', fontsize=12)

    plt.tight_layout()
    plt.show()

    #Медианные значения
    avg_rates = ['score','equity','balance','salary']
    for i in avg_rates:
        print(f"Медианное значение {i} - {round(df[i].median(),2)}")
    #Средние значения
    avg_rates = ['age','equity','products','products_wo_credit']
    for i in avg_rates:
        print(f"Среднее значение {i} - {round(df[i].mean(),2)}")


portrait(df)
```

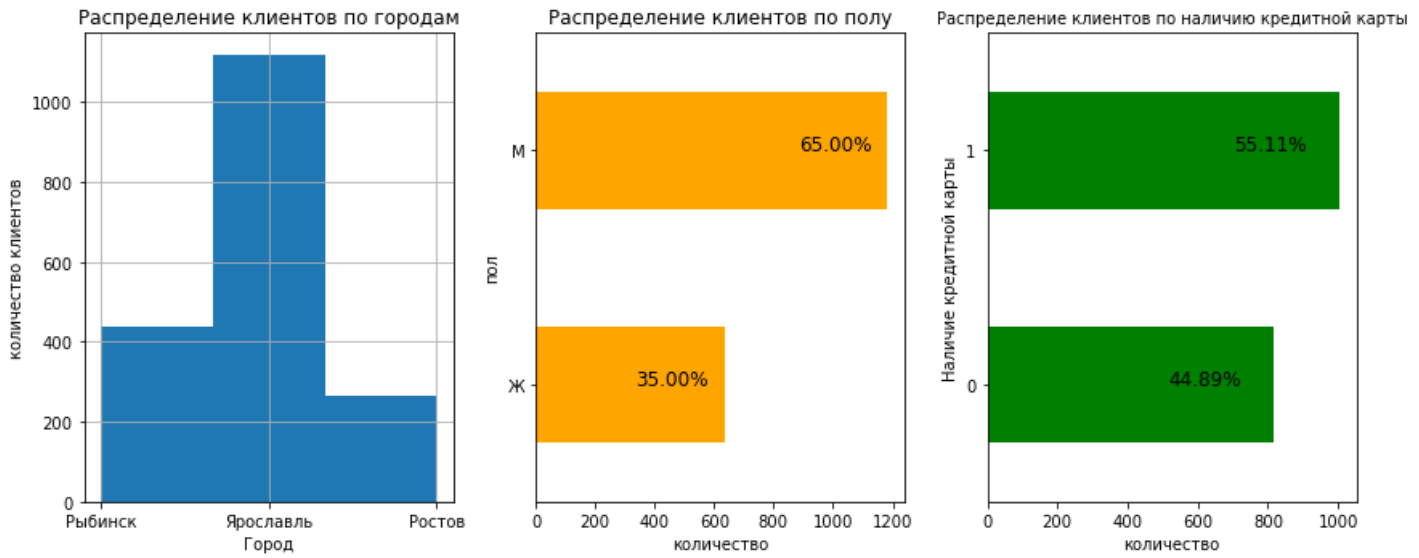



```
Медианное значение score - 853.0
Медианное значение equity - 3.0
Медианное значение balance - 521905.58
Медианное значение salary - 119658.1
Среднее значение age - 42.73
Среднее значение equity - 2.63
Среднее значение products - 1.87
Среднее значение products_wo_credit - 1.19
```

Портрет среднего клиента банка - Мужчина 40 лет из Ярославля с кредитной картой, с кредитным рейтингом -853, с балансом счет - 520 000 рублей, зарплатой - 120 000 рублей, использующий 2 банковских продукта, с оценкой собственности - 3.0 .

Построим "портрет" отточного клиента банка

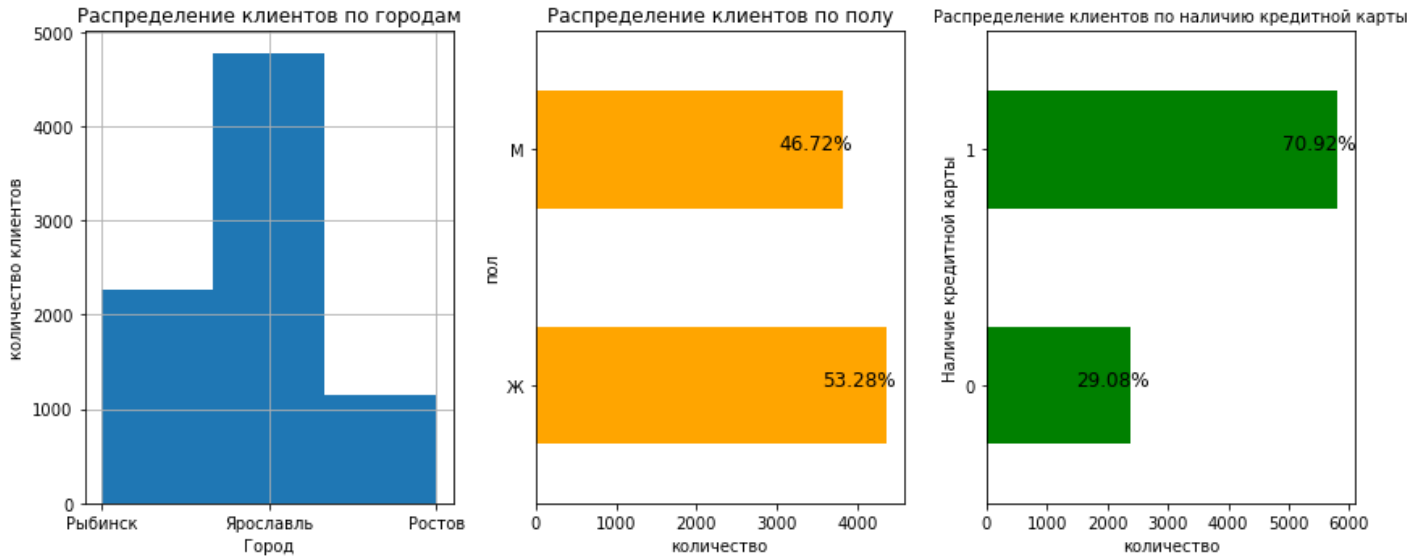
In [76]:
```
portrait(df.query('churn == 1'),text1=300,text2=300)
```




```
Медианное значение score - 866.0
Медианное значение equity - 4.0
Медианное значение balance - 782410.47
Медианное значение salary - 125390.62
Среднее значение age - 41.45
Среднее значение equity - 3.76
Среднее значение products - 2.38
Среднее значение products_wo_credit - 1.82
```

Построим "портрет" неотточного клиента банка

In [77]:
```
portrait(df.query('churn == 0'),text1=800,text2=900)
```




```
Медианное значение score - 848.0
Медианное значение equity - 3.0
Медианное значение balance - 474672.71
Медианное значение salary - 118257.71
Среднее значение age - 43.02
Среднее значение equity - 2.37
Среднее значение products - 1.76
Среднее значение products_wo_credit - 1.05
```

Выводы по "портретам" клиента

При сравненении портретов **отточного** и **неотточного** клиентов мы видм следующие видимые различия

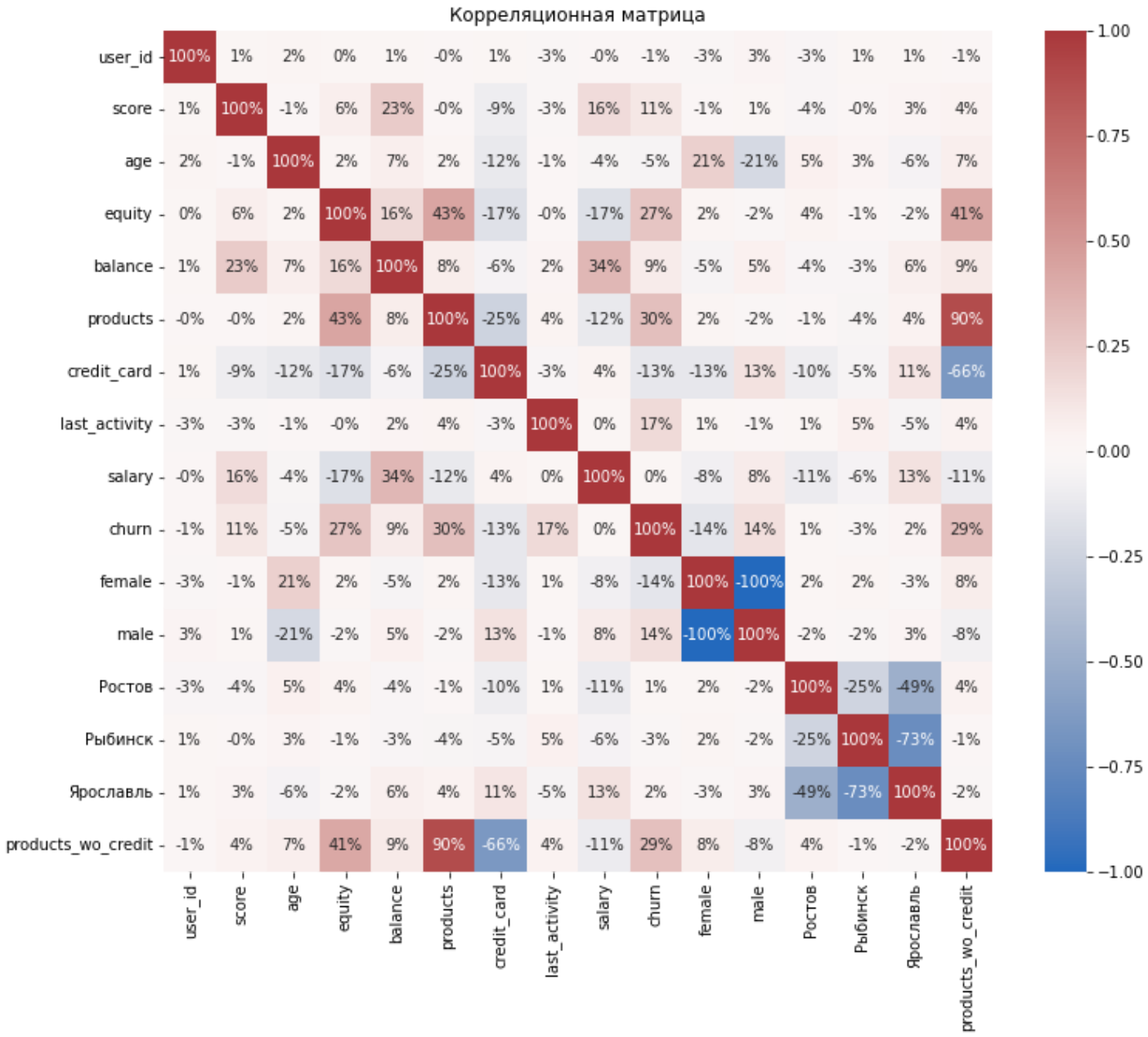
- Различие по оценке собственности клиента `equity` (*далее - проведем проверку данной гипотезы*)
- Различие по оценке количества `банковских продуктов` (с учетом `products` и без учета кредитной карты `products_wo_credit`)
- Различие по оценке медианного баланса на счете `balance` | Портрет клиента | Оценка собственности клиента | Количество продуктов | Количество продуктов (без кредитной карты)| Медианный баланс клиента | | --- | --- | --- |--- |--- | | Отточный клиент | 3.76 | 2.38 |1.82 |782 410.47 руб. | | Неотточный клиент | 2.37 | 1.76 |1.05 |474 672.71 руб.|

- Отточными клиентами чаще становятся мужчины
- У неотточных клиентов больше процент владения кредитной картой - 71% против 55%

Построим график корреляции оттока от других параметров в датасете

In [78]:

```
corr = df.corr()

# создание графика корреляционной матрицы
plt.figure(figsize=(12, 10))
sns.heatmap(corr, annot=True,fmt='.0%', cmap='vlag')

# настройка заголовка графика
plt.title('Корреляционная матрица')

# отображение графика
plt.show()
```



Нас интересует целевая характеристика - **отток клиента**.

Слабая корреляция (по шкале Чеддока) наблюдается между оттоком и следущими параметрами :

- уровень кредитного рейтинга клиента - **11%**
- количество объектов в собственности - **27%**
- наличие кредитной карты - **-13%** (отрицательная корреляция)
- Активность клиента - **17%**
- Женский пол клиента - **-14%** (отрицательная корреляция)
- Мужской пол клиента - **14%**
- количество банковских продуктов без учета кредитной карты - **29%**

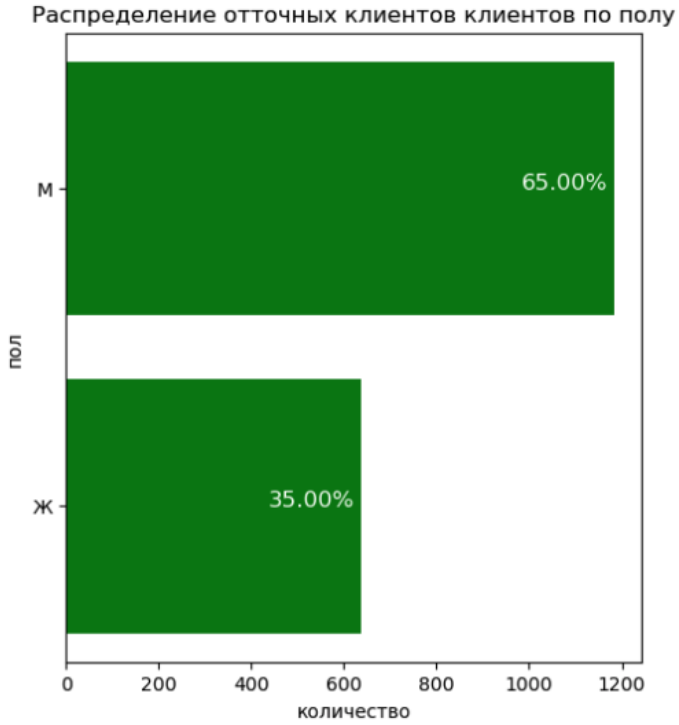
Умеренная корреляция наблюдается между **оттоком** и **количеством продуктов банка у клиента** - **30%**

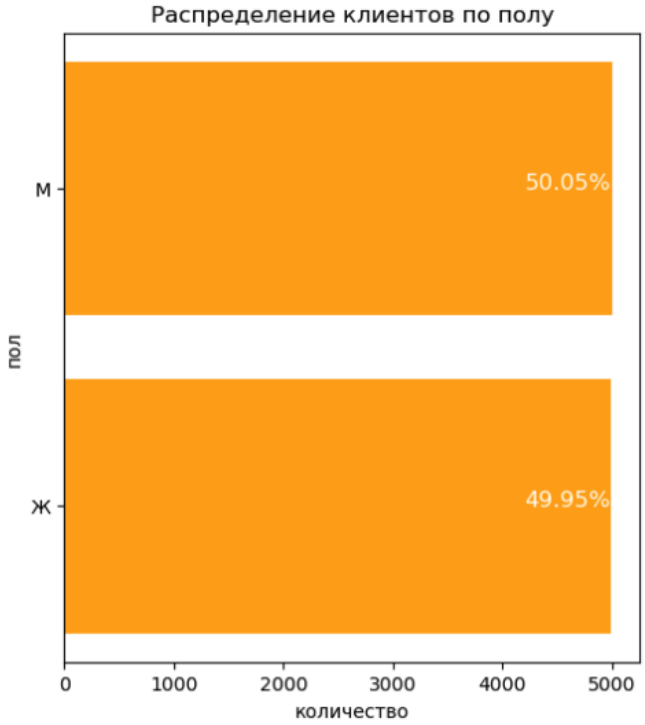
Вывод по Шагу 3 - Исследовательский анализ данных

1) Количество отточных клиентов в исходном датасете - 18.2%

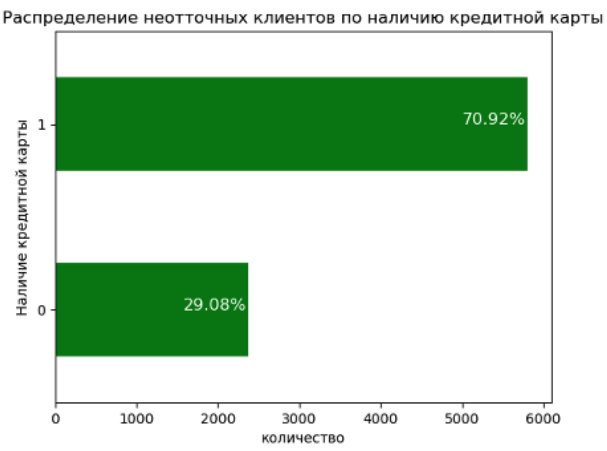
2) Большой отток у возрастных групп от 25 до 34 лет и у группы от 50 до 60 лет

3) Чаще всего отточными клиентами являются мужчины - 65%,при том что, изначально клиентов банка - мужчин и женщин - у нас поровну.

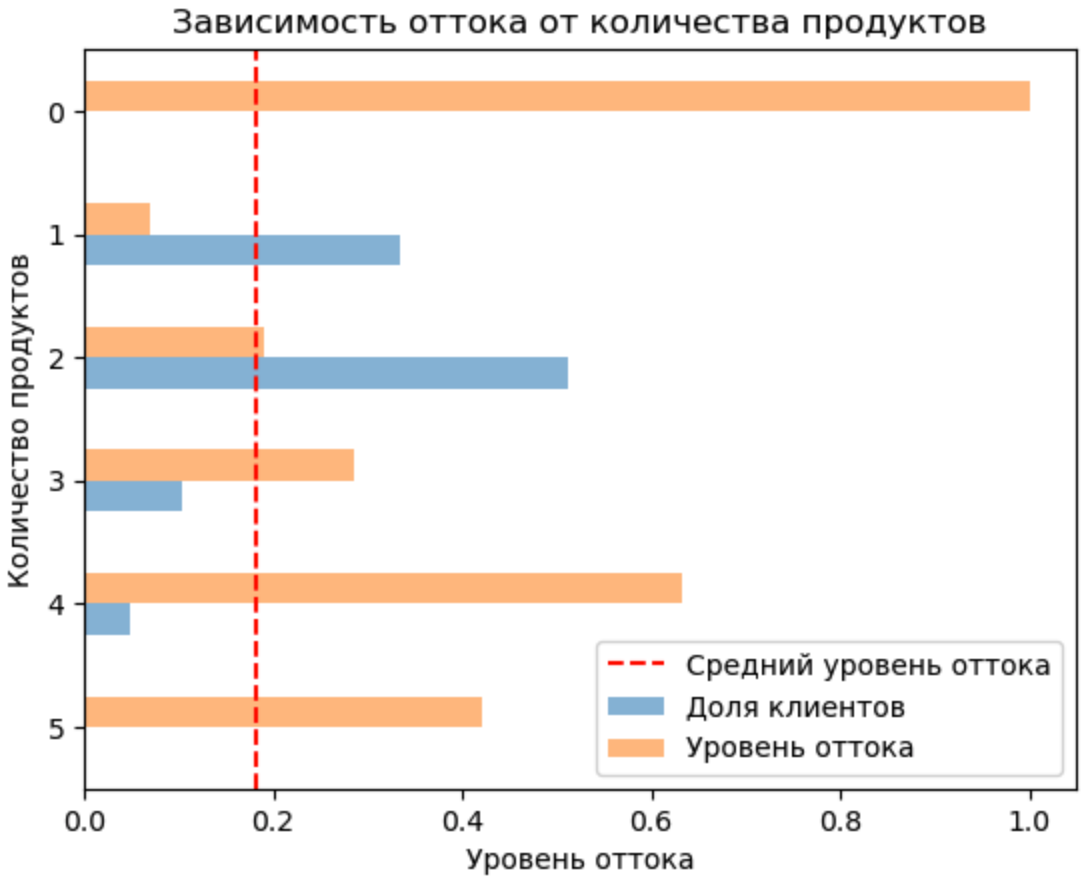




4) У неотточных клиентов - процент наличия кредитной карты больше, чем у отточных 71% против 29%

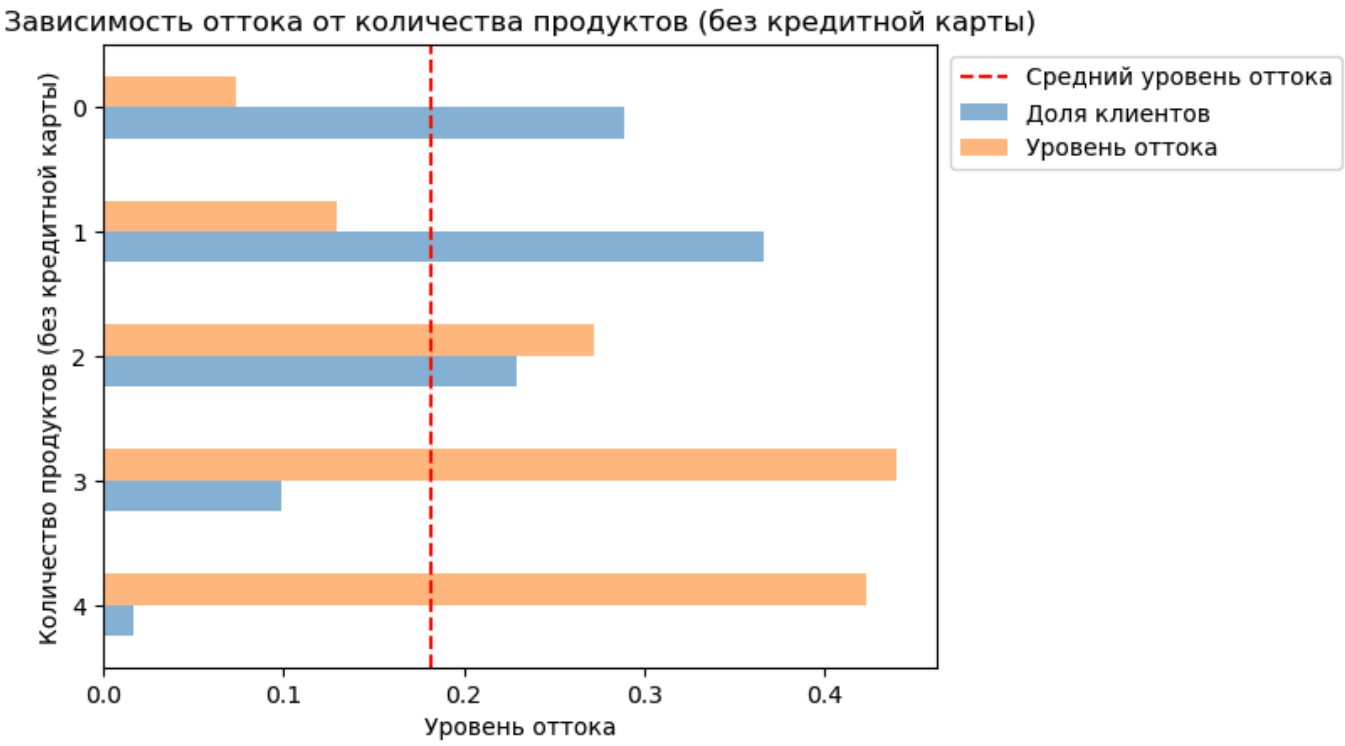


5) Большинство отточных клиентов - клиенты владеющие **не менее 3мя банковскими продуктами** (поле

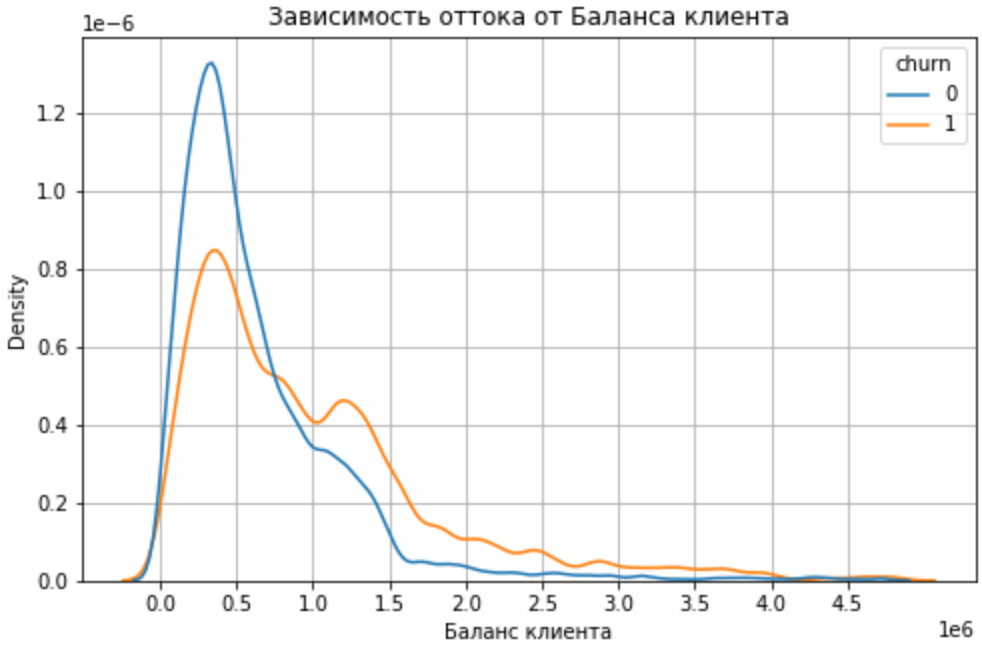


- `products`)

Если рассматривать клиентов без учета наличия кредитной карты - то отток начинается от **2х банковских продуктов**



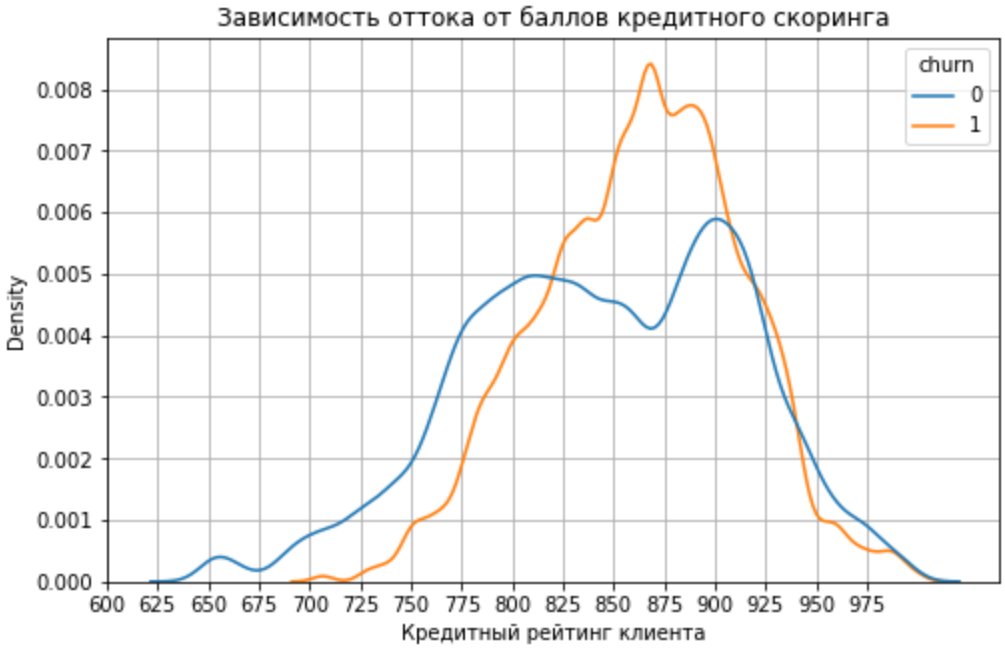
6) у большого процента отточных клиентов - баланс составляет от 750 000 до 4 000 000 рублей



Большинство клиентов в абсолютном значении с указанным балансом, проживают в Ярославле

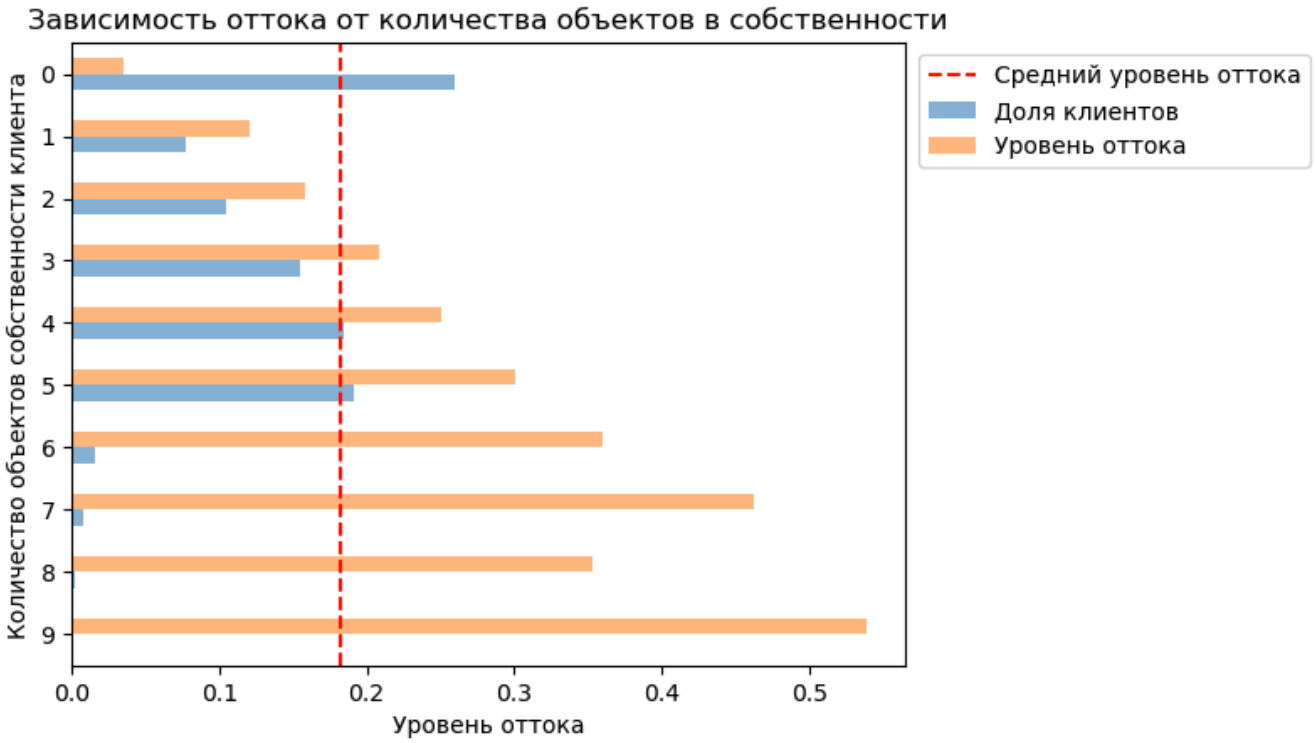
	city	user_id
0	Ярославль	1543
1	Рыбинск	697
2	Ростов	352

7) Большой всплеск оттока наблюдается у клиентов с кредитным рейтингом от 810 до 910.



- в возрасте до 50 лет - чаще встречаются мужчины
- в возрасте после 60 лет - чаще встречаются женщины

8) Наибольший отток наблюдается у клиентов, у которых в собственности от 3 до 9 объектов.

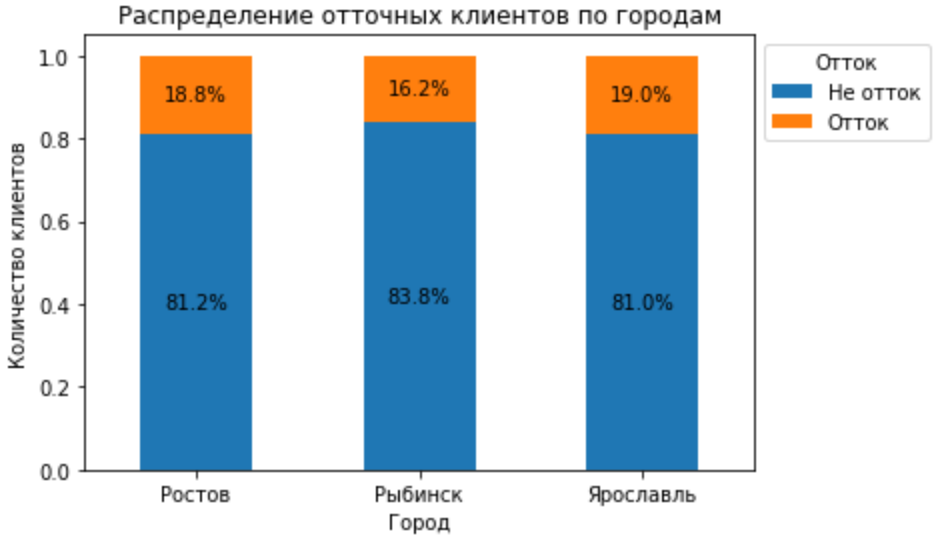


С увеличением числа объектов собственности - уровень оттока увеличивается

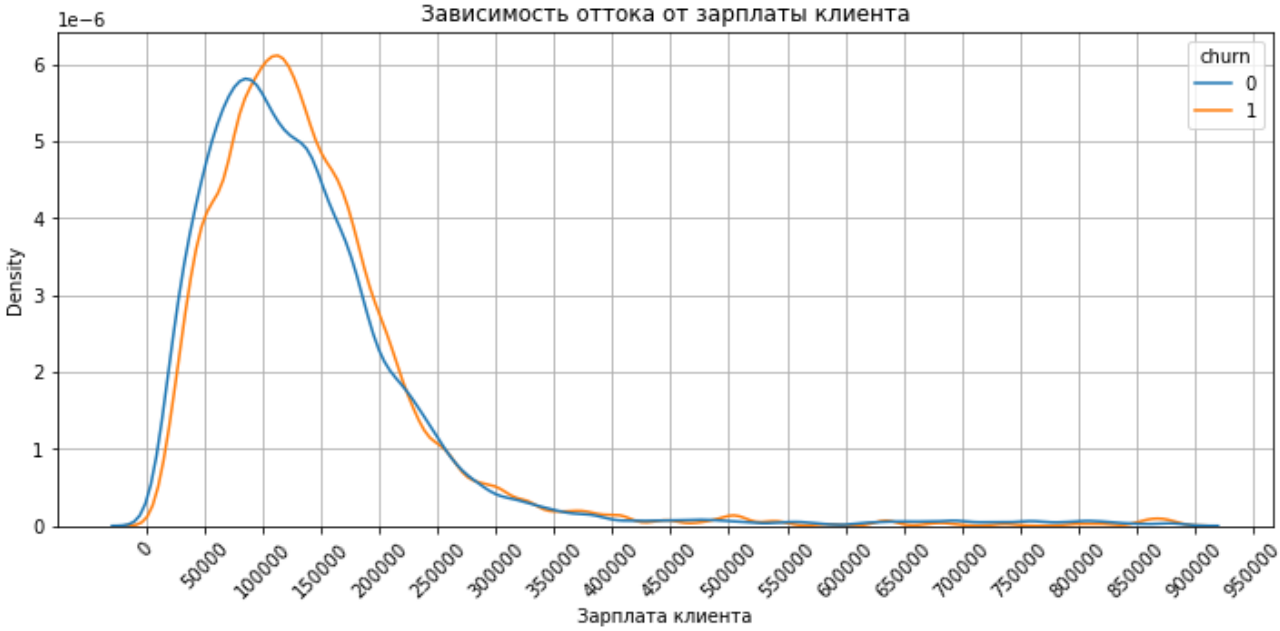
9) процент отточных клиентов по каждому городу примерно одинаковый -

- Ярославль - 19 % отточных клиентов
- Ростов - 18,8% отточных клиентов

- Рыбинск - 16,2% отточных клиентов

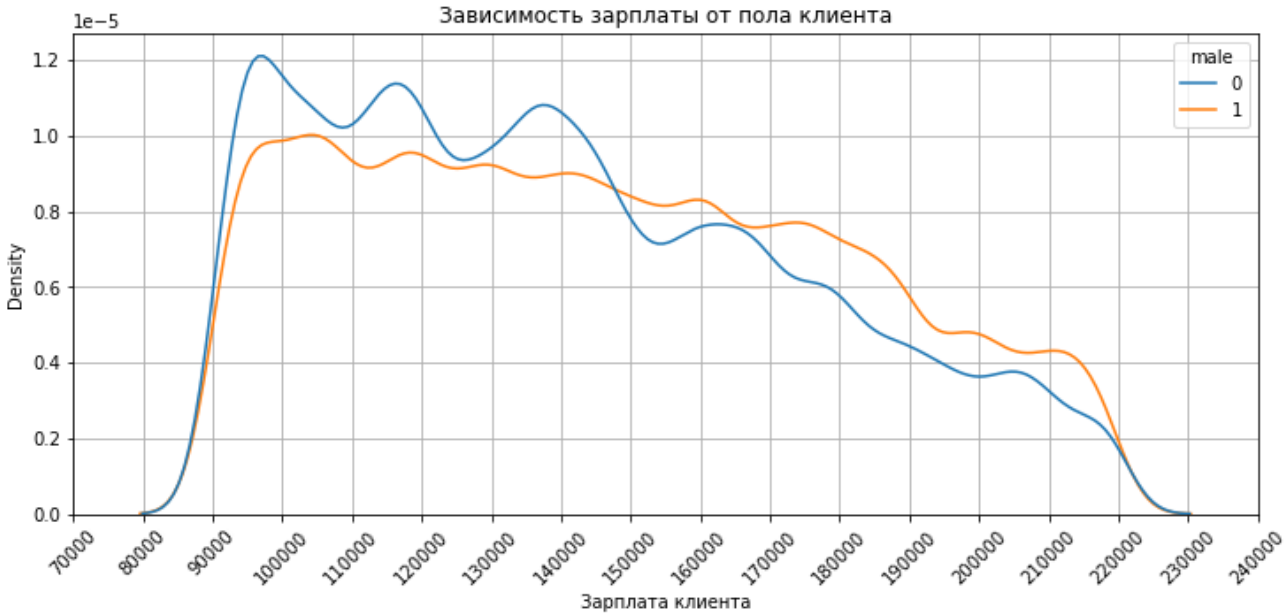


10) Большой отток наблюдается у наших клиентов с зарплатой от 90 000 до 220 000 рублей.

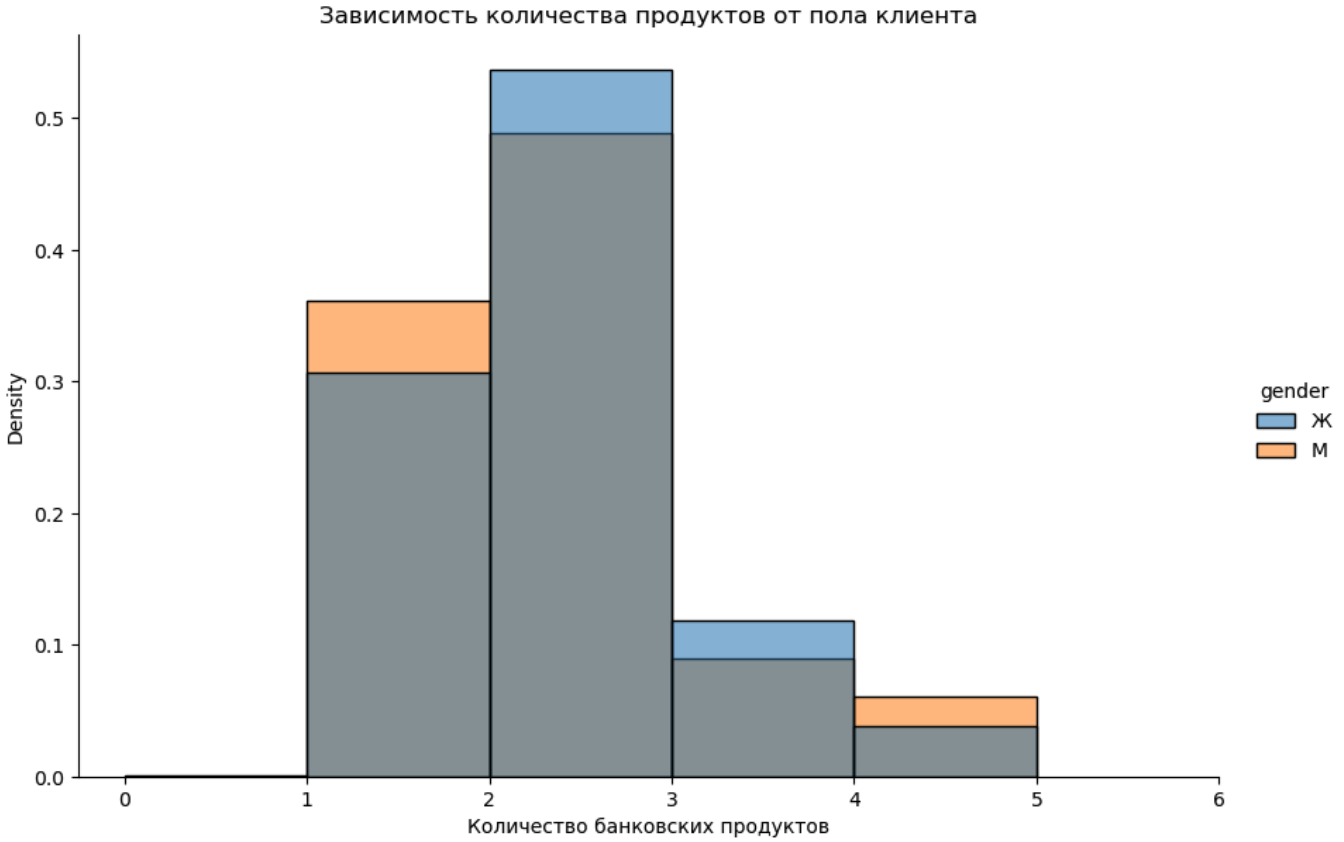


- Уровень Зарплаты от 90 000 до 150 000 рублей по большой части у жителей Ростова и Рыбинска
- Зарплата от 150 000 рублей у жителей Ярославля
- В данном сегменте чаще всего встречаются клиенты с количеством объектов недвижимости - 3,6 или 7
- В нашем датасете больше женщин с заработной платой от 90 000 до 145 000 рублей

- После зарплаты 150 000 до зарплаты 220 000 идет перевес мужского населения



11) У мужчин чаще всего бывает 1 или 4 банковских продукта, у женщин - 2 и 3 продукта



12) **Портрет среднего клиента банка** - Мужчина 43 лет из Ярославля с кредитной картой, с кредитным рейтингом -853, с балансом счет - 520 000 рублей, зарплатой - 120 000 рублей, использующий 2 банковских продукта, с оценкой собственности 2.63 .

13) При сравненении портретов **отточного** и **неотточного** клиентов мы видм следующие видимые различия

- Различие по оценке собственности клиента `equity`
- Различие по оценке количества `банковских продуктов` (с учетом `products` и без учета кредитной карты `products_wo_credit`)

- Различие по оценке медианного баланса на счете `balance` | Портрет клиента | Оценка собственности клиента | Количество продуктов | Количество продуктов (без кредитной карты)| Медианный баланс клиента | | --- | --- | --- |--- |--- | | Отточный клиент | 3.76 | 2.38 |1.82 |782 410.47 руб. | | Неотточный клиент | 2.37 | 1.76 |1.05 |474 672.71 руб.|
- Отточными клиентами чаще становятся мужчины
- У неотточных клиентов больше процент владения кредитной картой - 71% против 55%

14) Слабая корреляция (по шкале Чеддока) наблюдается между **оттоком клиентов** и следущими параметрами :

- уровень кредитного рейтинга клиента - **11%**
- количество объектов в собственности - **27%**
- наличие кредитной карты - **-13%** (отрицательная корреляция)
- Активность клиента - **17%**
- Женский пол клиента - **-14%** (отрицательная корреляция)
- Мужской пол клиента - **14%**
- количество банковских продуктов без учета кредитной карты - **29%**

Умеренная корреляция наблюдается между **оттоком клиентов** и **количеством продуктов банка у клиента** - **30%**

Шаг 4. Проверка Гипотез

Гипотеза №1.

Проверим гипотезу различия возраста между теми клиентами, которые пользуются двумя продуктами банка, и теми, которые пользуются одним.

Для проверки данной гипотезы мы будем использовать **U-критерий Манна-Уитни для независимых выборок**.

Мы также могли использовать **t-критерий Стьюдента**, т.к. выборка довольна большая (значительно больше 30ти элементов)

Обычно мощность U-критерия Манна-Уитни выше, чем мощность t-критерия Стьюдента, поскольку U-критерий Манна-Уитни не предполагает никаких предположений о распределении выборок и, следовательно, может использоваться в более широком диапазоне случаев, когда данные могут иметь нетипичное распределение. Т-критерий Стьюдента, с другой стороны, более чувствителен к нормальности данных и может давать неправильные результаты, если данные сильно отклоняются от нормального распределения.

Для более точных результатов воспользуемся `критерием Манна-Уитни`

Для проверки гипотез - удалим все пропуски в поле `age`

Нулевая гипотеза - H0 - Возраст между клиентами, которые используют один и два продукта банка не отличается.

Альтернативная гипотеза - H1 - Возраст между клиентами, которые используют один и два продукта банка отличается.

Уровень значимости alpha - 0.05

In [79]:
```
one_product = df[df['products']==1]['age'].dropna()
two_products = df[df['products']==2]['age'].dropna()
alpha = 0.05

stat, p = mannwhitneyu(one_product, two_products, alternative='two-sided')
print(f'p-value: {round(p,4)}')

if (p<alpha):
    print('Отвергаем нулевую гипотезу.\nСредний возраст различается между группами')
else:
    print('Мы не можем отклонить нулевую гипотезу.\nСредний возраст не различается между г
```

```
p-value: 0.0571
Мы не можем отклонить нулевую гипотезу.
Средний возраст не различается между группами
```

Гипотеза №2.

Проверим гипотезу различия в количестве объектов в собственности (поле - `equity`) между отточными и неотточными клиентами.

Для проверки гипотезы будем использовать `тест Манна-Уитни`

Нулевая гипотеза - H0 - Количество объектов в собственности между отточными и неотточными клиентами не отличается.

Альтернативная гипотеза - H1 - Количество объектов в собственности между отточными и неотточными клиентами отличается.

Уровень значимости alpha - 0.05

In [80]:
```
churn_clients = df[df['churn']==1]['equity']
non_churn_clients = df[df['churn']==0]['equity']
alpha = 0.05

stat, p = mannwhitneyu(churn_clients, non_churn_clients, alternative='two-sided')
print(f'p-value: {p}')

if (p<alpha):
    print('Отвергаем нулевую гипотезу.\nКоличество объектов в собственности между отточным
else:
    print('Мы не можем отклонить нулевую гипотезу.\nКоличество объектов в собственности ме
```

```
p-value: 2.2272704044623725e-158
Отвергаем нулевую гипотезу.
Количество объектов в собственности между отточными и неотточными клиентами отличается
```

Гипотеза №3.

Проверим гипотезу различия в количестве банковских продуктов (поле - `products`) между отточными и неотточными клиентами.

Для проверки гипотезы будем использовать `тест Манна-Уитни`

Нулевая гипотеза - H0 - Средний баланс неотточных клиентов равен среднему балансу отточных клиентов.

Альтернативная гипотеза - H1 - Средний баланс неотточных клиентов отличается от среднего баланса отточных клиентов.

Уровень значимости alpha - 0.05

In [81]:
```
churn_balance = df[df['churn']==1]['balance'].dropna()
non_churn_balance = df[df['churn']==0]['balance'].dropna()

alpha = 0.05

stat, p = mannwhitneyu(churn_clients, non_churn_clients, alternative='two-sided')
print(f'p-value: {p}')

if p<alpha:
    print('Отвергаем нулевую гипотезу.\nСредний баланс неотточных клиентов отличается от с
else:
    print('Мы не можем отклонить нулевую гипотезу.\nСредний баланс неотточных клиентов ра
```

```
p-value: 2.2272704044623725e-158
Отвергаем нулевую гипотезу.
Средний баланс неотточных клиентов отличается от среднего баланса отточных клиентов.
```

Вывод по проверке гипотез

В результате статистической проверки гипотез, мы **подтвердили следующие утверждения** -

1) Количество объектов в собственности `equity` между отточными и неотточными клиентами отличается.

2) Средний баланс `balance` неотточных клиентов отличается от среднего баланса отточных клиентов.

Мы **не смогли подтвердить** следующую гипотезу -

1) **Средний возраст между теми клиентами, которые пользуются двумя продуктами банка, и теми, которые пользуются одним отличается**

Сегментация на основе продуктов и стратегических показателей

Построение сегментов пользователей на основе нашего исследования

1 Сегмент - Мужчины, в возрасте 51-60 лет, с количеством банковских продуктов(без кредитной карты) от 2х

In [82]:
```
segment1 = df.query('male==1 and products_wo_credit >=2 and age>50 and age<=60 ')\
.groupby('churn')['user_id'].count()

def churn_rate(segment):
    total = segment['user_id'].sum()
    churn = round(segment[segment['churn']==1]['user_id'].sum()/total*100,2)
    print(f'Количество людей в сегменте - {total}')
    print(f"Процент оттока - {churn}%")

churn_rate(segment1.reset_index(drop=False))
```

```
Количество людей в сегменте - 279
Процент оттока - 54.48%
```

2 Сегмент - Клиенты (М,Ж) с количеством банковских продуктов от 3х, количеством объектов в собственности от 3х и зарплатой от 120 000 рублей.

In [83]:
```
segment2 = df\
.query('equity>3  & salary > 120000 & products >3')\
.groupby('churn')['user_id'].count()

churn_rate(segment2.reset_index(drop=False))
```

```
Количество людей в сегменте - 189
Процент оттока - 77.25%
```

3 Сегмент - Мужчины, в возрасте 51-59 лет, с балансом счета от 750 000 , с количеством продуктов от 2х

In [84]:
```
segment3 = df\
.query('age>=51 and age<60 and male==1 and balance >750000 and products >= 2 ')\
.groupby('churn')['user_id'].count()

churn_rate(segment3.reset_index(drop=False))
```

```
Количество людей в сегменте - 223
Процент оттока - 60.99%
```

4 Сегмент - Мужчины с кредитным рейтингом от 810 до 910, с количеством продуктов от 3х и балансом от 750 000 до 4 000 000 рублей

In [85]:
```
segment4 = df\
.query('male==1 & score>810 &score<910 & products >= 3 &balance > 750000 &balance < 400000
.groupby('churn')['user_id'].count()

churn_rate(segment4.reset_index(drop=False))
```

```
Количество людей в сегменте - 191
Процент оттока - 68.59%
```

5 Сегмент - Мужчины, в возрасте 25-34 лет, с балансом счета от 750 000 рублей

In [86]:
```
segment5 = df\
.query('age>=25 and age<=34 and male==1 and balance >750000  ')\
.groupby('churn')['user_id'].count()
churn_rate(segment5.reset_index(drop=False))
```

```
Количество людей в сегменте - 300
Процент оттока - 58.33%
```

6 Сегмент - Мужчины из Ярославля, с количеством продуктов больше 3х и балансом от 750 000 до 4 000 000 рублей

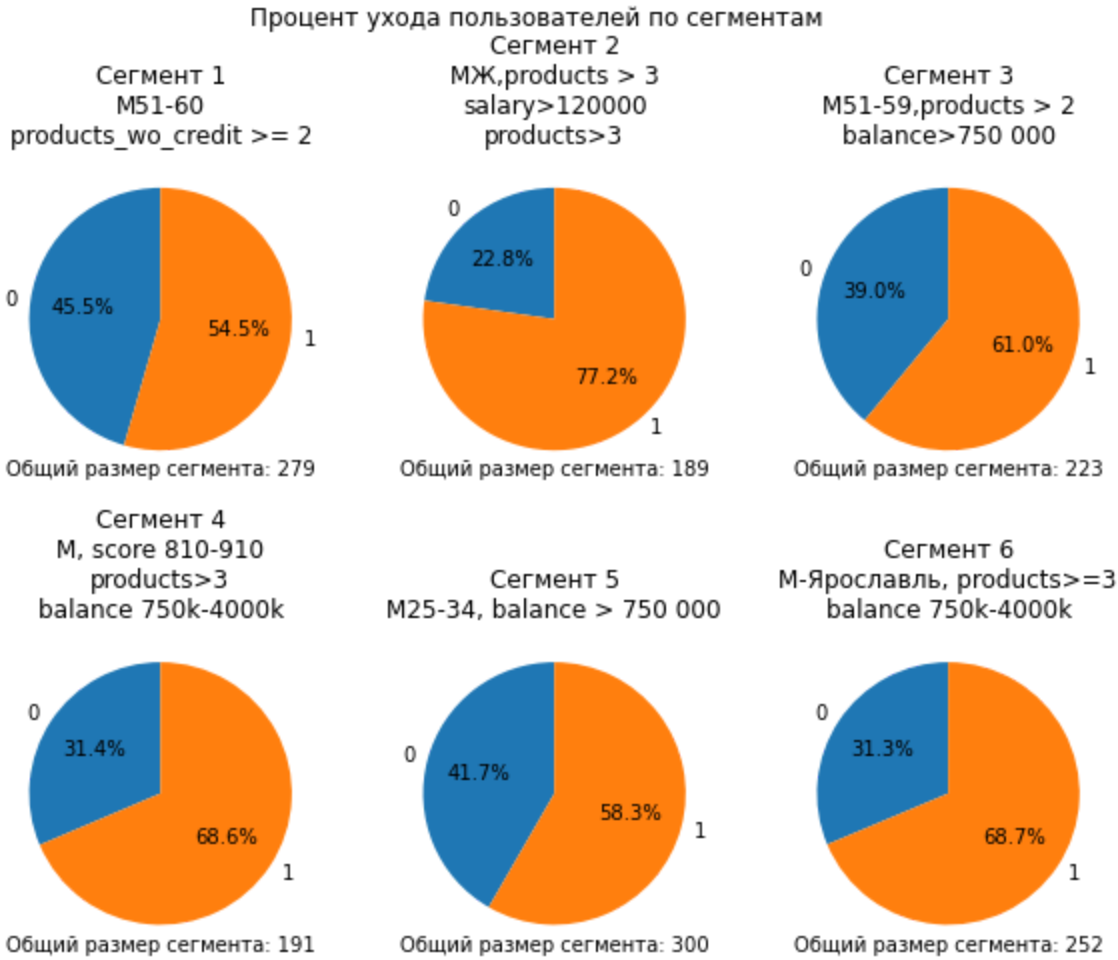
In [87]:
```
segment6 = df\
.query('male==1 and balance > 750000 &balance < 4000000 and products >=3 and city=="Яросла
.groupby('churn')['user_id'].count()
churn_rate(segment6.reset_index(drop=False))
```

```
Количество людей в сегменте - 252
Процент оттока - 68.65%
```

Проверим наши созданные сегменты на отточность

In [88]:
```
# Создаем словарь для хранения данных каждого сегмента
segments = {
    'Сегмент 1\nM51-60\nproducts_wo_credit >= 2': segment1,
    'Сегмент 2\nМЖ,products > 3\nsalary>120000\nproducts>3': segment2,
    'Сегмент 3\nM51-59,products > 2\nbalance>750 000': segment3,
    'Сегмент 4\nM, score 810-910\nproducts>3\nbalance 750k-4000k': segment4,
    'Сегмент 5\nM25-34, balance > 750 000': segment5,
```

```
    'Сегмент 6\nM-Ярославль, products>=3\nbalance 750k-4000k': segment6

}

# Создаем фигуру и добавляем 6 ячеек
fig, axs = plt.subplots(2, 3, figsize=(10, 8))

# Итерируемся по словарю и строим графики для каждого сегмента в соответствующей ячейке
for i, (title, segment) in enumerate(segments.items()):
    counts = segment.values
    labels = segment.index
    percent = counts / counts.sum() * 100

    row = i // 3
    col = i % 3

    axs[row, col].pie(percent, labels=labels, autopct='%1.1f%%', startangle=90)
    axs[row, col].set_title(title)
    axs[row, col].annotate('Общий размер сегмента: {}'.format(counts.sum()), xy=(0, -1.2),

# Добавляем общий заголовок для всей фигуры
fig.suptitle('Процент ухода пользователей по сегментам')


# Отображаем фигуру
plt.show()
```



Процент оттока по сегментам составялет **от 54,5% до 77,2%** (с численностью от 189 до 300 человек)

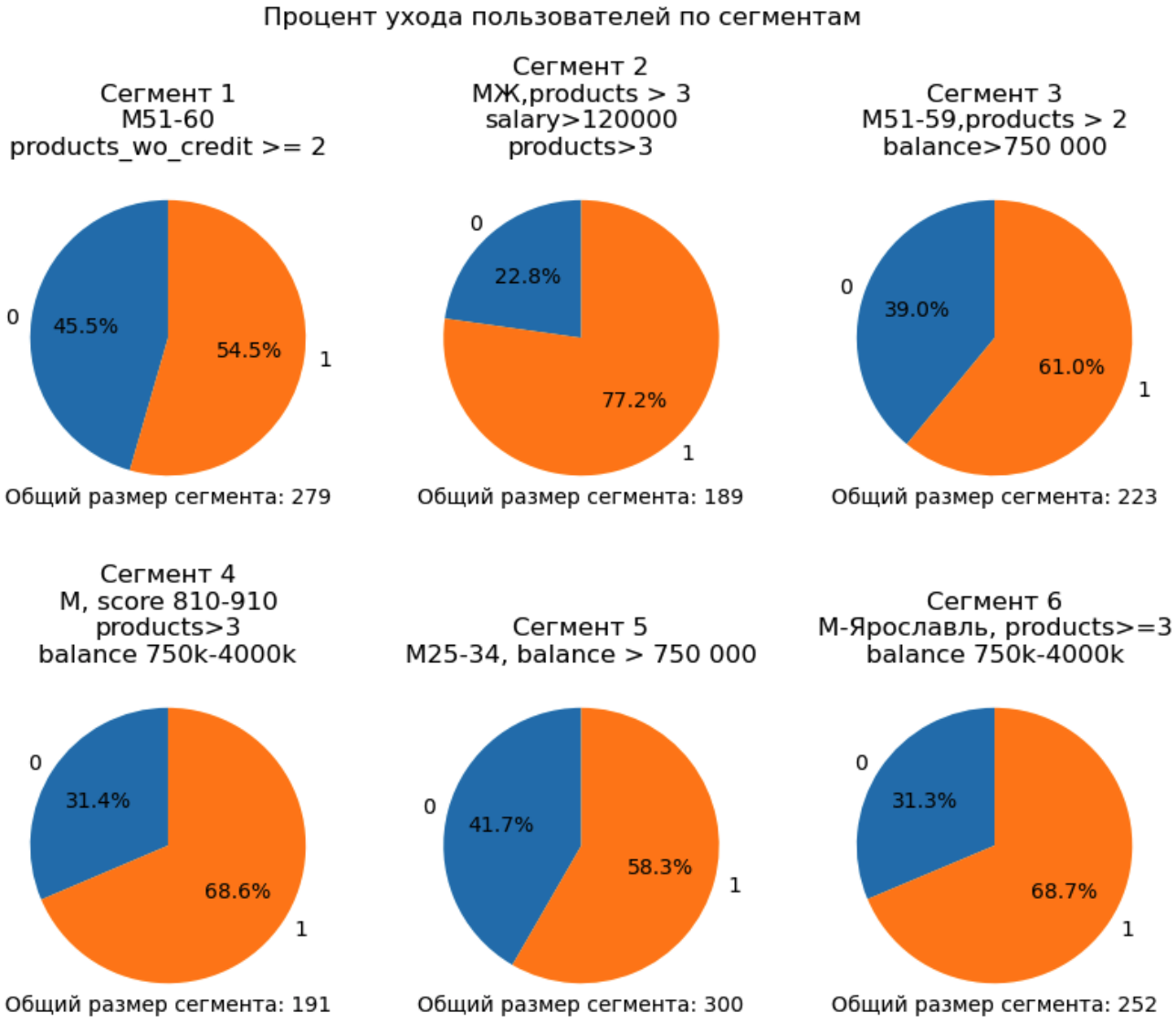
Процент оттока в исходном датасете составлял 18,2 % . Считаю - что разбиение на сегменты прошло успешно.

Выводы и рекомендации для Заказчика

Задачей проекта было **проанализировать клиентов регионального банка** и сегментировать пользователей по количеству потребляемых продуктов, обращая особое внимание на отток.

Отток в исходном датасете составлял - 18,2 %.

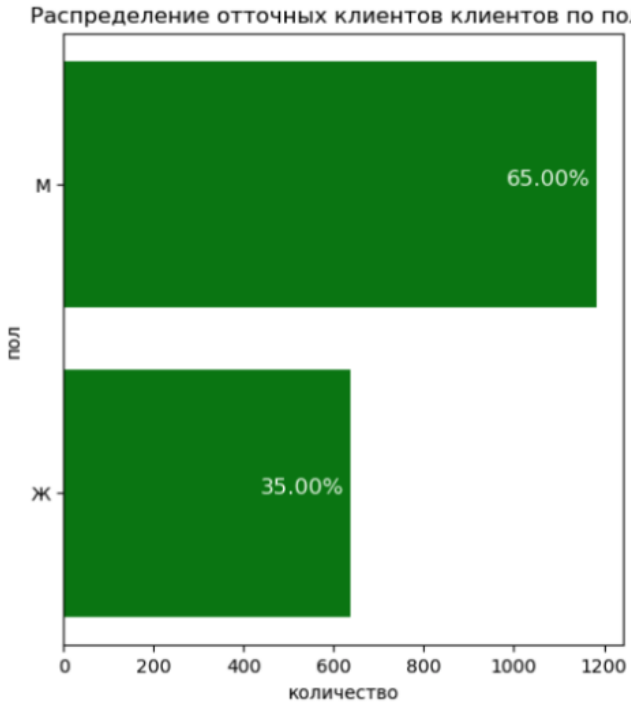
Мы выделили **6 отточных сегментов** - **с процентом оттока от 54,5% до 69,5 %.**

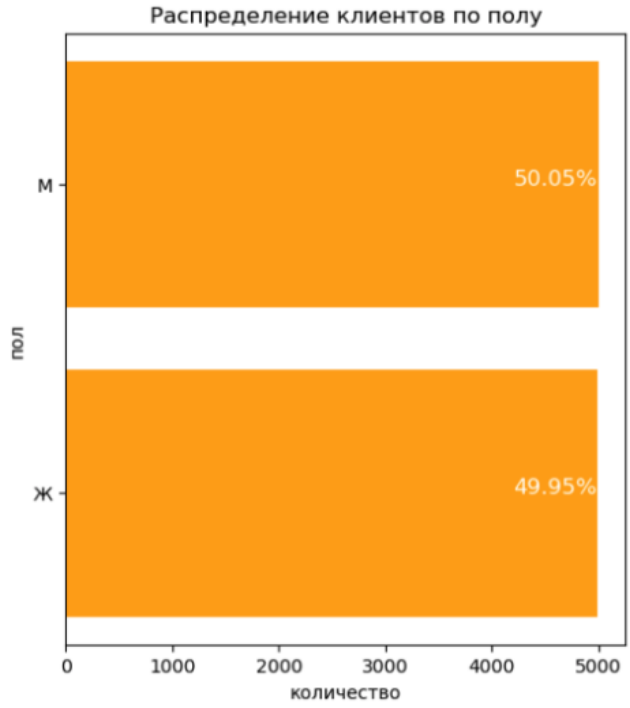


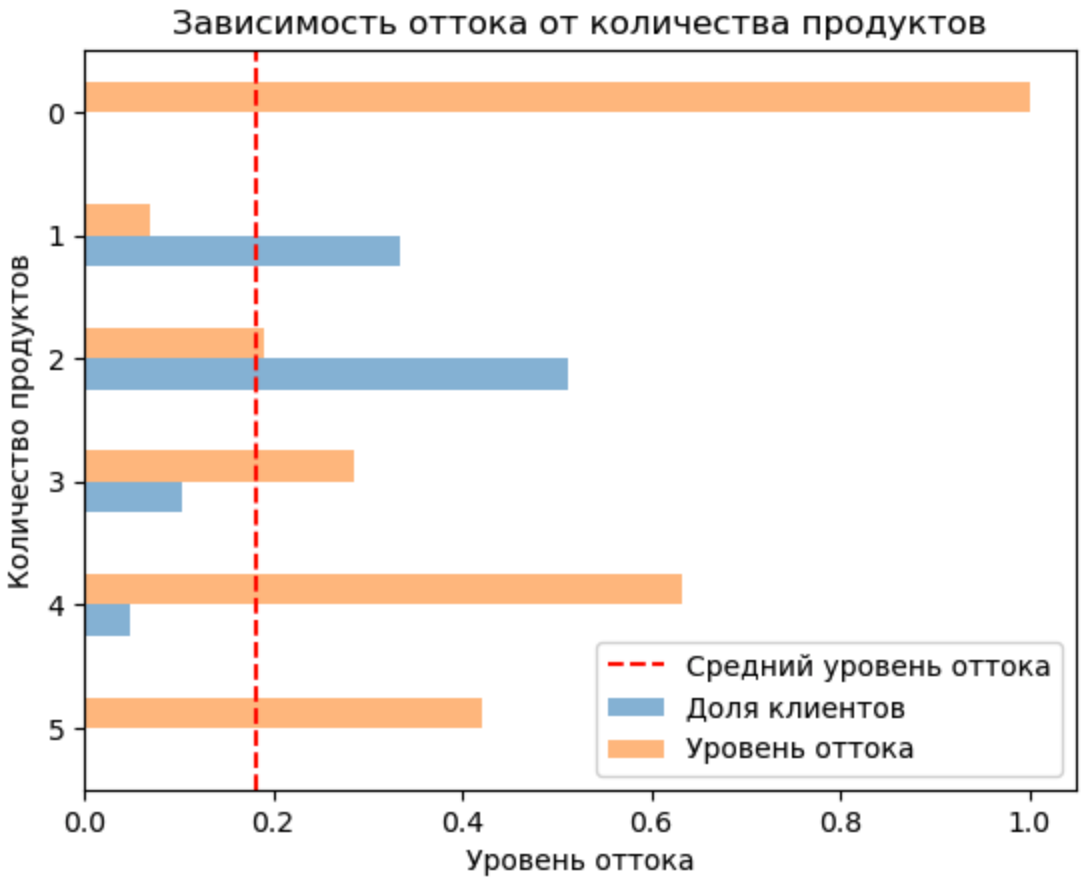
- **Сегмент 1** - Мужчины, 51-60 лет, количество банковских продуктов(без учета кредитной карты) от 2х - 279 человек
- **Сегмент 2** - Клиенты (М,Ж) с количеством банковских продуктов от 3х, количеством объектов в собственности от 3х и зарплатой от 120 000 рублей.
- **Сегмент 3** - Мужчины, 51-59лет, количество продуктов от 2х, баланс от 750 000 рублей - размер сегмента - 223 человека
- **Сегмент 4** - Мужчины, кредитный рейтинг 810-910, количество продуктов от 3х, баланс от 750 000 до 4 млн.руб. - размер сегмента - 191 человек
- **Сегмент 5** - Мужчины, 25-34 года, баланс от 750 000 рублей - размер сегмента - 300 человек
- **Сегмент 6** - Мужчины из Ярославля, банковские продукты от 3х, баланс от 750 000 до 4 млн.рублей

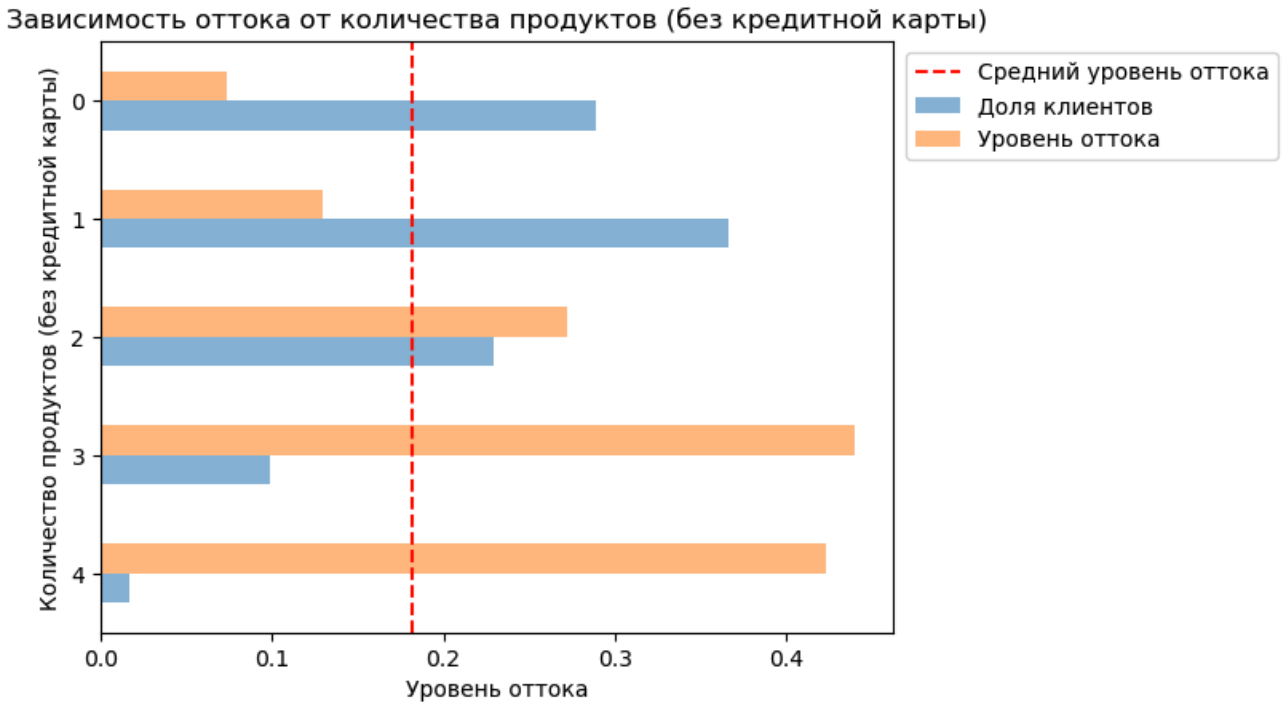
- **Отточными** чаще всего являются мужчины **возраста от 51-60** с **балансом от 750 000 рублей** и **количеством банковских продуктов от 3х**. **Кредитный рейтинг** клиентов - **от 810 до 910**, **количество объектов в собственности от 3х**. Клиенты пробуют различные наши продукты и не

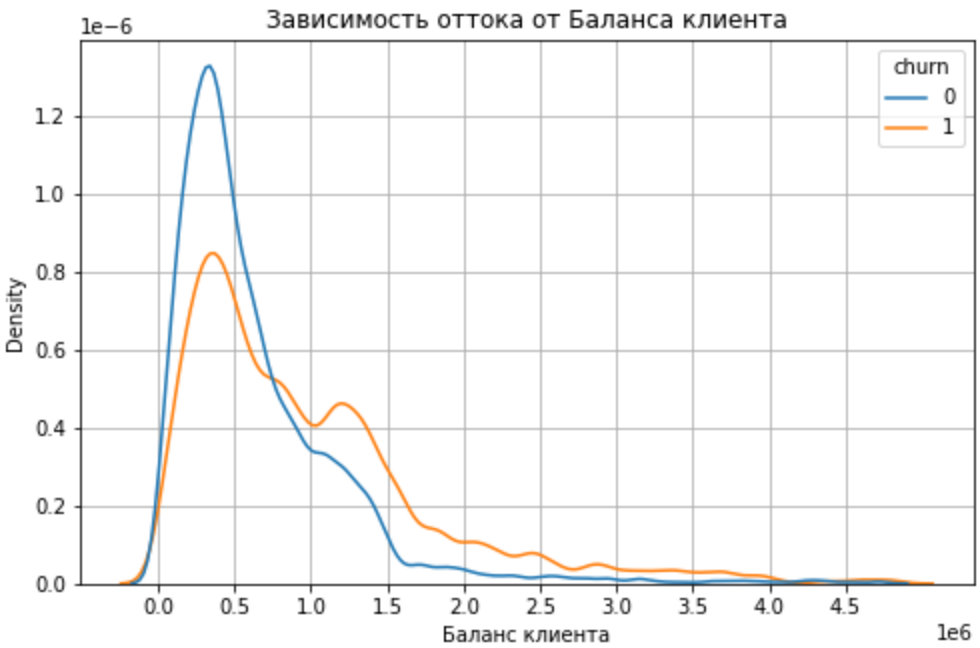
остаются довольны.



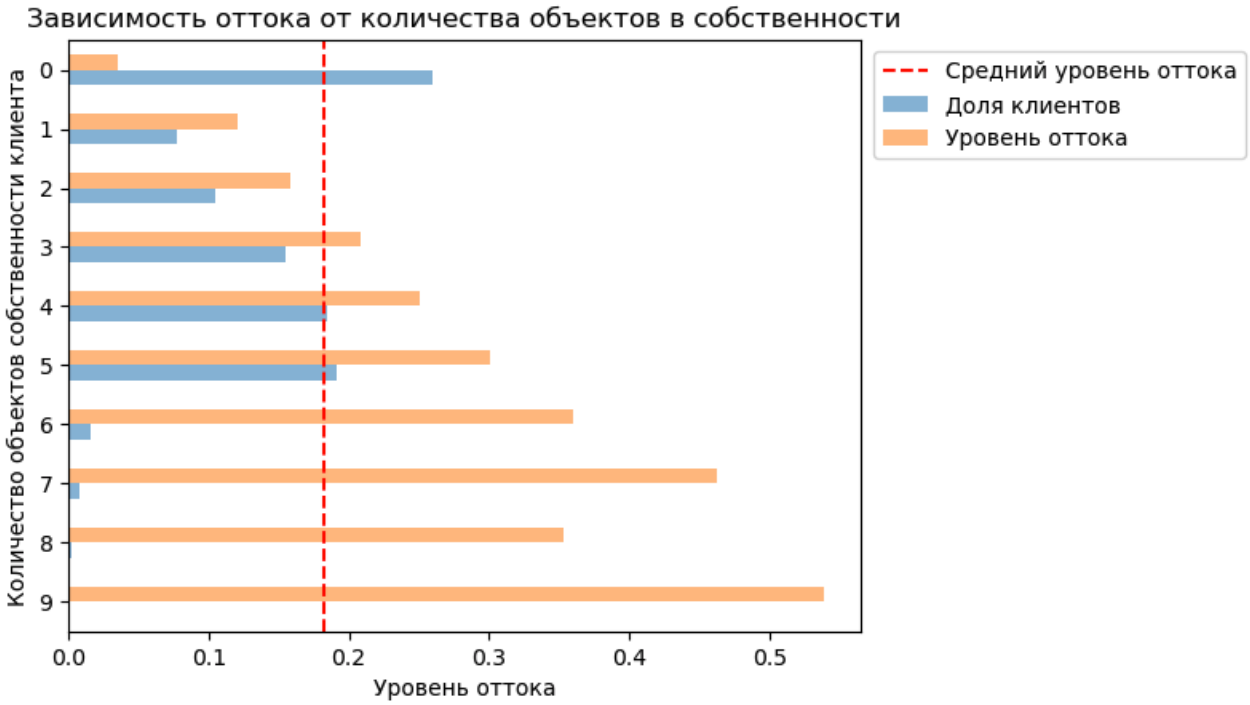



Зависимость оттока от количества продуктов
Количество продуктов
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
Уровень оттока
Средний уровень оттока
Доля клиентов
Уровень оттока


Зависимость оттока от количества продуктов (без кредитной карты)
Количество продуктов (без кредитной карты)
0
1
2
3
4
0.0
0.1
0.2
0.3
0.4
Уровень оттока
Средний уровень оттока
Доля клиентов
Уровень оттока


Зависимость оттока от Баланса клиента
1e−6
churn
0
1
Density
Баланс клиента
1e6

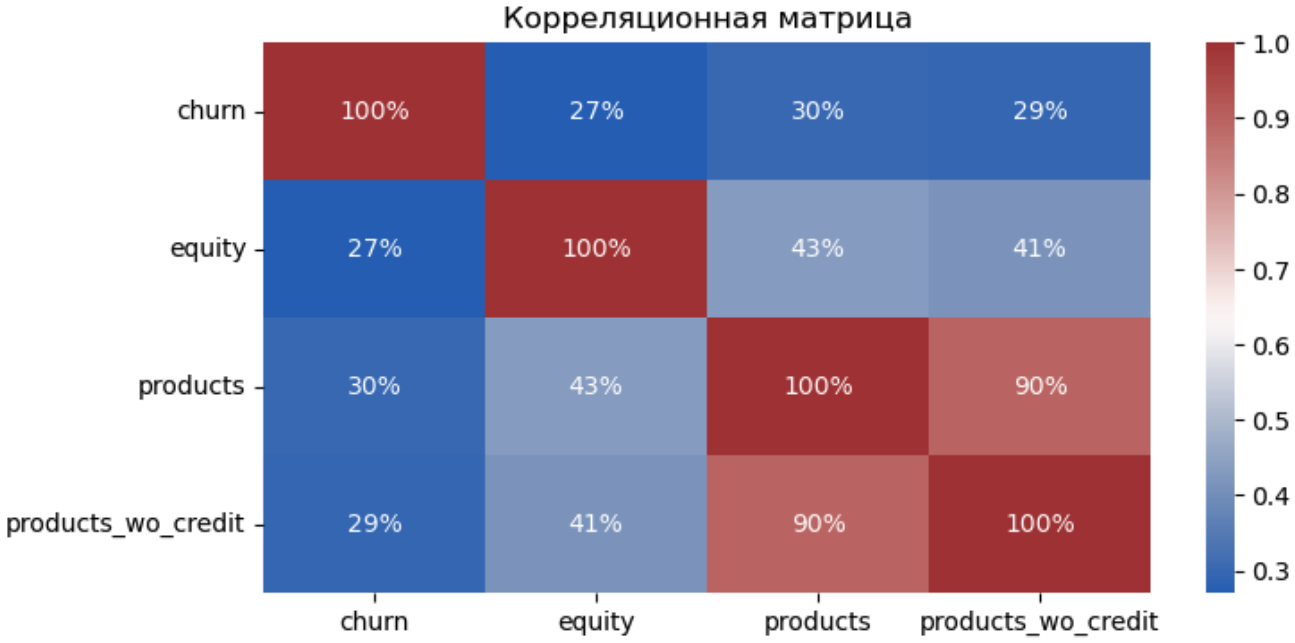

Зависимость оттока от баллов кредитного скоринга
churn
0
1
Density
Кредитный рейтинг клиента



Слабая корреляция (по шкале Чеддока) наблюдается между **оттоком** и следующими параметрами :

- уровень кредитного рейтинга клиента - **11%**
- количество объектов в собственности - **27%**
- наличие кредитной карты - **-13%** (отрицательная корреляция)
- Активность клиента - **17%**
- Женский пол клиента - **-14%** (отрицательная корреляция)
- Мужской пол клиента - **14%**
- количество банковских продуктов без учета кредитной карты - **29%**

Умеренная корреляция наблюдается между **оттоком** и **количеством продуктов банка у клиента** - **30%**



Рекомендации -

- Предлагается, для **сегментов - 3,6** запустить акцию по особым условия для **депозитного вклада - повышенные годовые проценты** при сумме вклада от 1 млн. рублей.
- также предлагаю рассмотреть возможность **льготного (или бесплатного) премиум обслуживания** клиентов при балансе от 1 млн.рублей. В премиум обслуживание должно входить - выделенный специалист для клиента, вклад с повышенной ставкой, повышенный кэшбэк с покупок, доступ в бизнес залы аэропортов.
- Для **молодежного сегмента (сегмент 5)** - добавить **льготное премиум обслуживание** (указанное раннее), добавить условия более льготного кредитования со **сниженной ставкой кредита**.
- для **сегмента 1** и **сегмента 4** (кредитный рейтинг 810-910) - добавим возможность получить **кредит на более выгодных условиях (со сниженной ставкой)**
- Для **сегмента 2** добавить дополнительные кэшбэк программы, добавить **льготное премиум обслуживание** (указанное раннее)

Презентация

- ссылка на презентацию https://disk.yandex.ru/i/nkda2A7Wdt_0fw

Дашборд

- ссылка на Дашборд https://clck.ru/342aCZ