**Проект выполнил:** Алексей Становой, в рамках учебного курса "Аналитик данных" Yandex.Practicum Москва, Февраль 2021г. +7 (962) 985-51-77 Телефон - WhatsApp - Telegram e-mail: a.v.stanovoy@yandex.ru

Проект:

# Анализ оттока клиентов в банке "Метанпром".

Цель:

**На основе анализа данных клиентов банка определить характерные черты ушедших клиентов для формирования банком стратегии их удержания.**

Содержание:



Материалы:

CSV-файл, содержащий данные о клиентах банка «Метанпром», расположенного в Ярославле и областных городах: Ростов Великий и Рыбинск.
Колонки:

- 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.

# 1. Загрузка данных и сбор общей информации.

Наверх

In [1]:
```
# Используемые библиотеки
import pandas as pd
from io import BytesIO
import requests
import numpy as np
from IPython.display import display
import seaborn as sns
from matplotlib import pyplot as plt
from scipy import stats as st
import plotly.express as px
from scipy.stats import fligner
```

In [2]:
```
#загрузка данных из внешнего источника
spreadsheet_id = '1E7OWRflF29KcJ0a0qRTd5ljT_lpRhc2Xo_g5grRj_rk'
file_name1 = 'https://docs.google.com/spreadsheets/d/{}/export?format=csv'.format(spreadsheet_id)
r = requests.get(file_name1)
data= pd.read_csv(BytesIO(r.content))
# первые 5 строк таблицы
display(data.head())
# подсчет количества строк и столбцов
print('Количество строк:', np.array(data.shape)[0], '. ',
      'Количество столбцов:', np.array(data.shape)[1])
```

| | userid | score | City | Gender | Age | Objects | Balance | Products | CreditCard | Loyalty | estimated_salary | Churn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 15677338 | 619 | Ярославль | Ж | 42 | 2 | NaN | 1 | 1 | 1 | 101348.88 | 1 |
| **1** | 15690047 | 608 | Рыбинск | Ж | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| **2** | 15662040 | 502 | Ярославль | Ж | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| **3** | 15744090 | 699 | Ярославль | Ж | 39 | 1 | NaN | 2 | 0 | 0 | 93826.63 | 0 |
| **4** | 15780624 | 850 | Рыбинск | Ж | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |

```
Количество строк: 10000 .  Количество столбцов: 12
```

**ВЫВОД:** Для анализа загружена таблица, имеющая 10 000 строк и 12 столбцов.

- Один из столбцов `userid` содержит ID клиента банка.
- В столбце `Churn` содержится целевая переменная, указывающая на наличие или отсутствие факта ооттока.
- Остальные столбцы содержат набор признаков по каждому клиенту.
- Наименования столбцов указаны разным регистром.
- Столбец `Gender` содержит буквенные обозначения пола.

# 2. Предобработка данных:

Наверх

**Для удобства все названия столбцов будут отображены нижним регистром.**

In [3]:
```
# для нижнего регистра применяется метод str.lower()
```

```
data.columns = data.columns.str.lower() # приведем названия столбцов к нижнему регистру
```

**Чтобы была возможность проще выявить различия по полу в столбце `gender` буквенные обозначения "Ж" и "М" будут заменены на 0 и 1 соответственно, а столбец на `male` .**

In [4]:
```
# При помощи .loc будут перебираться значения в столбце "gender" и передаваться
# в столбец "male" в виде числового идентификатора "0" или "1"
data.loc[data['gender'] == 'Ж', 'male'] = 0
data.loc[data['gender'] == 'М', 'male'] = 1
# Выбор подходящего типа данных
data['male'] = data['male'].astype('int64')
```

**Чтобы понять количество прпусков в датасете и определить типы данных будет использован метод "info".**

In [5]:
```
data.info() # общая информация по датасету
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   userid            10000 non-null  int64  
 1   score             10000 non-null  int64  
 2   city              10000 non-null  object 
 3   gender            10000 non-null  object 
 4   age               10000 non-null  int64  
 5   objects           10000 non-null  int64  
 6   balance           6383 non-null   float64
 7   products          10000 non-null  int64  
 8   creditcard        10000 non-null  int64  
 9   loyalty           10000 non-null  int64  
 10  estimated_salary  10000 non-null  float64
 11  churn             10000 non-null  int64  
 12  male              10000 non-null  int64  
dtypes: float64(2), int64(9), object(2)
memory usage: 1015.8+ KB
```

**Проверку дубликатов следует применить как построчно так и в отдельности к столбцу с ID клиентов, чтобы убедиться в корректности учета данных.**

In [6]:
```
# проверка наличия задвоенных строк
print('Количество дублированных строк:', data.duplicated().sum())
# проверка наличия задвоенных "userid"
print('Количество дублированных "userid":', data['userid'].duplicated().sum())
# подсчет количество строк и столбцов
```

```
Количество дублированных строк: 0
Количество дублированных "userid": 0
```

**Отсутствующие значения** имеются только в одном столбце. Этот показатель делит данные на 2 крупных куска:

- 64% - с числовым значением,
- 36% - без значения.

**Так как `userid` не содержит дублей и имеет только функцию индентификации клиента для удобства дальнейшего анализа признаков он будет перенесен в `index` .**

In [7]:
```
data = data.set_index('userid')
```

**ВЫВОД:**

- Название колонок приведены к нижнему регистру, что сделало вывод данных более читабельным и позволяет избежать ошибок при указании их наименований.
- Типы данных соответствуют, содержащимся в столбцах данным.
- Построчных дубликатов, так же, как и дублей в ID клиента не выявлено.
- Пропуски имеются только в одном столбце, `balance` . Это означает, что для части клиентов нет данных по балансу счета. Скорее всего здесь имеет место выделение как самостоятельного признака, отдельного банковского продукта, продвижение которого отслеживается руководством банка.
- Описанное выше указывает на то, что признак `balance` имеет также весомое значение при исследовании оттока и следует изучить влияние на отток величины признака.

- Замена отсутствующих значений на среднее невозможна без искажения общей картины, так как этот признак для каждого клиента индивидуален. Вместо этого следует добавить новый признак, сообщающий об отсутствии (0) или наличии (1) баланса на счете.

In [8]:
```
# Добавление столбца, отражающего наличие или отсутствие признака `balance`
# При помощи .loc будут перебираться значения в столбце "balance" и передаваться
# в столбец "availability_balance" в виде числового идентификатора "1"
data.loc[data['balance'] == data['balance'], 'availability_balance'] = 1
# Далее все пропуски методом fillna() заменяются на "0"
data['availability_balance']=data['availability_balance'].fillna(0).astype('int64')
# вывод пяти первых строк
data.head(5)
```

Out[8]:

| userid | score | city | gender | age | objects | balance | products | creditcard | loyalty | estimated_salary | churn | male | av |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15677338** | 619 | Ярославль | Ж | 42 | 2 | NaN | 1 | 1 | 1 | 101348.88 | 1 | 0 | |
| **15690047** | 608 | Рыбинск | Ж | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 | 0 | |
| **15662040** | 502 | Ярославль | Ж | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 | 0 | |
| **15744090** | 699 | Ярославль | Ж | 39 | 1 | NaN | 2 | 0 | 0 | 93826.63 | 0 | 0 | |
| **15780624** | 850 | Рыбинск | Ж | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 | 0 | |

# 3. Исследовательский анализ данных.

Наверх

**Метод *describe* поможет взглянуть на основные параметры признаков.**

In [9]:
```
data.describe().round(2).T.reset_index() #Транспонирование и округление до 2-х значений после запятой
# сделает вывод компактней и читабельней
```

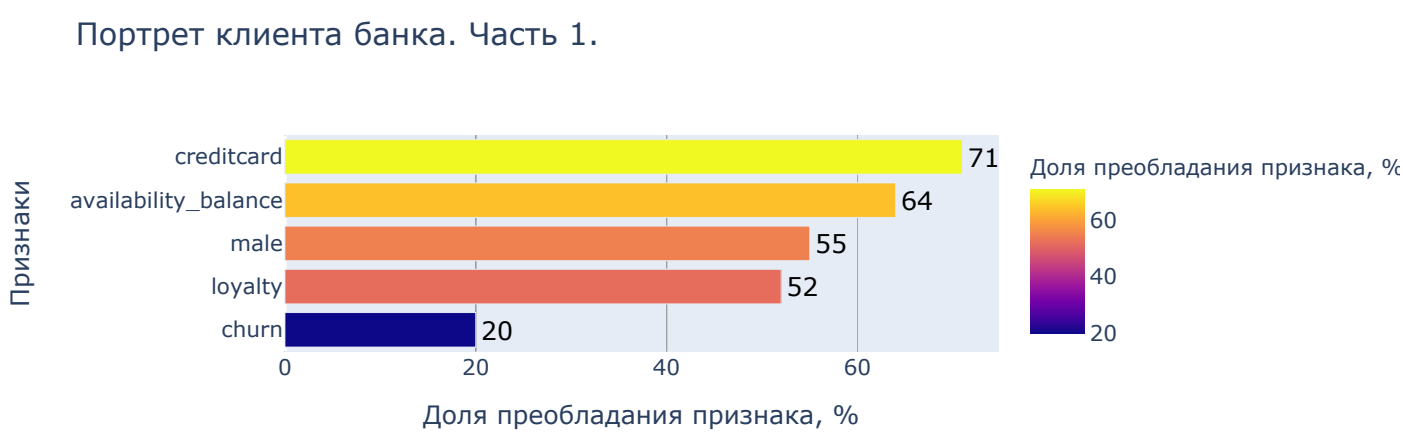
Out[9]:

| | index | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|---|
| **0** | score | 10000.0 | 650.53 | 96.65 | 350.00 | 584.00 | 652.00 | 718.00 | 850.00 |
| **1** | age | 10000.0 | 38.92 | 10.49 | 18.00 | 32.00 | 37.00 | 44.00 | 92.00 |
| **2** | objects | 10000.0 | 5.01 | 2.89 | 0.00 | 3.00 | 5.00 | 7.00 | 10.00 |
| **3** | balance | 6383.0 | 119827.49 | 30095.06 | 3768.69 | 100181.98 | 119839.69 | 139512.29 | 250898.09 |
| **4** | products | 10000.0 | 1.53 | 0.58 | 1.00 | 1.00 | 1.00 | 2.00 | 4.00 |
| **5** | creditcard | 10000.0 | 0.71 | 0.46 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| **6** | loyalty | 10000.0 | 0.52 | 0.50 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| **7** | estimated_salary | 10000.0 | 100090.24 | 57510.49 | 11.58 | 51002.11 | 100193.92 | 149388.25 | 199992.48 |
| **8** | churn | 10000.0 | 0.20 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
| **9** | male | 10000.0 | 0.55 | 0.50 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| **10** | availability_balance | 10000.0 | 0.64 | 0.48 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |

**Графическое отображение признаков покажет их распределение. Так как некоторые признаки `creditcard, loyalty, churn, male, availability_balance` содержат только два числовых категориальных знаяения "0" или "1", то наглядней их показать в виде диаграммы преобладания того или иного признака.**

In [10]:
```
# Подготовка данных для plotly.express.bar: вычисляется среднее, фильтруются признаки,
# занчения переводятся в проценты, сортируются
data_mean=((pd.DataFrame(data.mean(), columns=['mean'])).query('mean <= 1').round(2)*100).astype('int')
data_mean_sort=data_mean.reset_index().sort_values(by='mean', ascending=True)
# визуализация при помощи plotly horizontal bar chart
fig = px.bar(data_mean_sort, x="mean", y="index",
             title='Портрет клиента банка. Часть 1.',
             text="mean", height=300, width=800, color='mean', orientation='h',
```

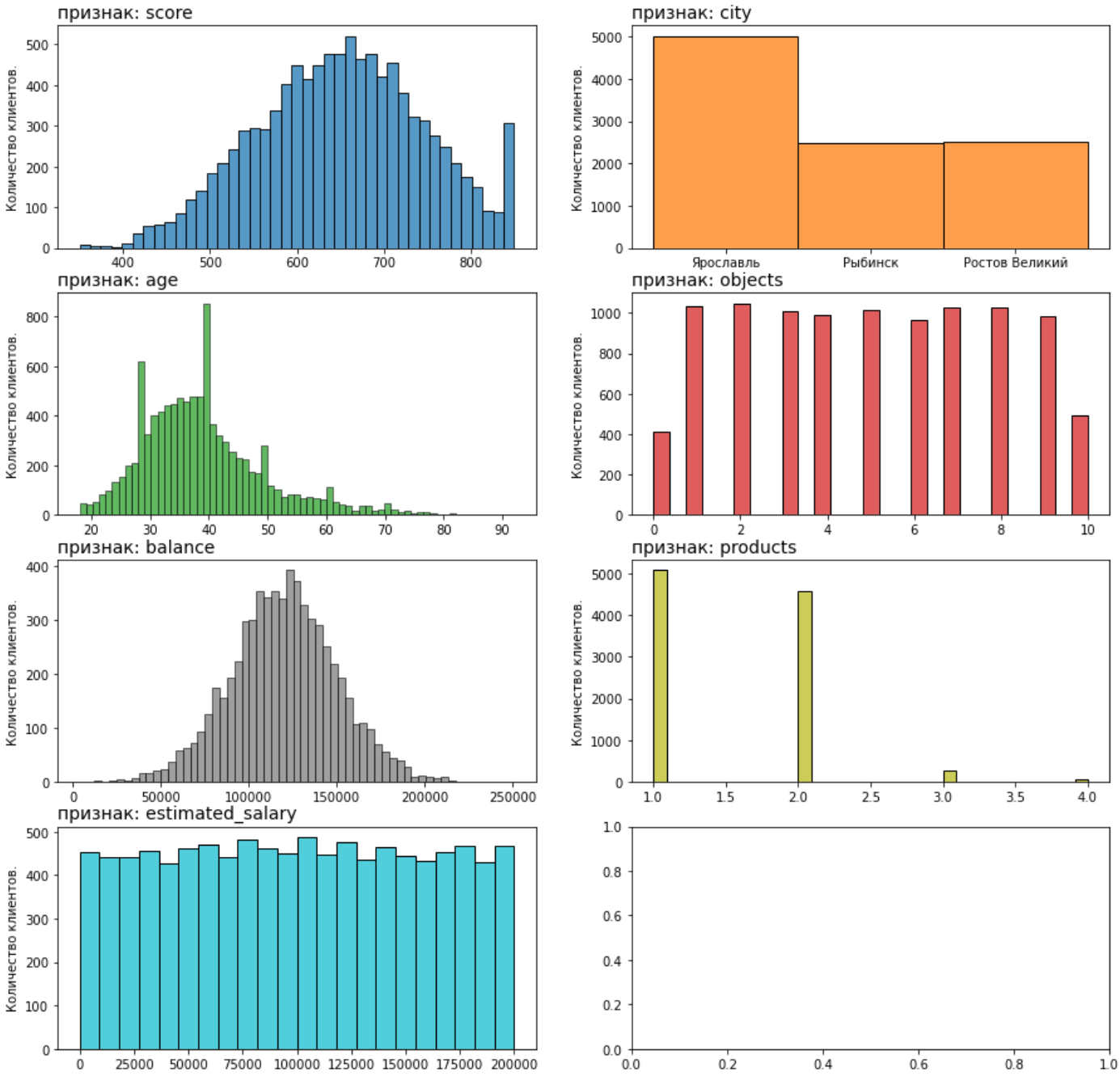
```
            labels={'index' : 'Признаки', 'mean': 'Доля преобладания признака, %'})
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
```




**Остальные признаки будут отлично смотреться в матрице из столбчатых графиков seaborn.histplot**.

In [11]:
```
# выделение оставшихся столбцов для построения графиков
data_plot=['score', 'city', 'age', 'objects', 'balance', 'products', 'estimated_salary']
# Параметры матрицы для отображения графиков
fig, axs = plt.subplots(4, 2, figsize = (15,15))
ax_iter=iter(axs.flat)
fig.suptitle('Портрет клиента банка. Часть 2.', fontsize=18)
# Последовательность цветов
color=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#7f7f7f', '#bcbd22', '#17becf']
# Запустим цикл по построению графиков распределения признаков
for i, c in zip(data_plot, color):
    ax=next(ax_iter)
    sns.histplot(data=data, x=i, ax=ax, color=c)
    ax.set_title(label= 'признак: '+i, loc='left', fontdict={'fontsize':14})
    ax.set_ylabel('Количество клиентов.')
    ax.set_xlabel('')
```

# Портрет клиента банка. Часть 2.




**ВЫВОД:** Вырисовывается следующий портрет клиента банка "Метанпром".

- Кредитный рейтинг **от 350 до 850 баллов** (Соответствует шкале НБКИ (Национальное бюро кредитных историй), где 300 - минимальное значение, а 850 максимальный балл).
- Средний балл кредитного скоринга клиентов банка равен **650**. Выше этого значения баллы почти у 50% клиентов.
- Мужчин среди клиентов банка немного больше, их доля составляет **55%**.
- Средний возраст - **39 лет**.
- В среднем клиенты банка владеют **5-ю объектами**. Такое количество объектов как минимум у 50% клиентов, а не менее 75% клиентов обладают 3 и более объектами.
- Средний баланс на счете **119 827.49**. При этом не мене 75% среди них имеют баланс более 100 000.
- **Более 50%** клиентов пользуются __только 1__ продуктом банка, чуть меньше клиентов используют 2 продукта. 3-4 продукта у одного клиента большая редкость.
- Кредитная карта есть у **71%** клиентов.
- Программой лояльности пользуются **52%** клиентов.
- Средний показатель предполагаемого уровня дохода составляет **100 090.24**. Такой доход имеют более 50% клиентов.

- Уровень оттока составил __20%__.
- Доля обладателей баланса на счете составляет **64%**.

**Взгляд на средние значения признаков** *в двух группах (кто ушел в отток и кто остался)* **в сравнении с общими показателями поможет выявить аномалии и указать на причины оттока.**

In [12]:
```
# Группировка с groupby по столбцу 'churn' и указанием средних признаков.
data_churn=data.groupby('churn').mean()
# Плюс соотношение средних признаков двух групп, чтобы выявить отклонения.
data_churn.loc['Соотношение'] = data_churn.loc[1]/data_churn.loc[0]
data_churn.round(2) #округление значений до 2-х знаков после запятой.
```

Out[12]:

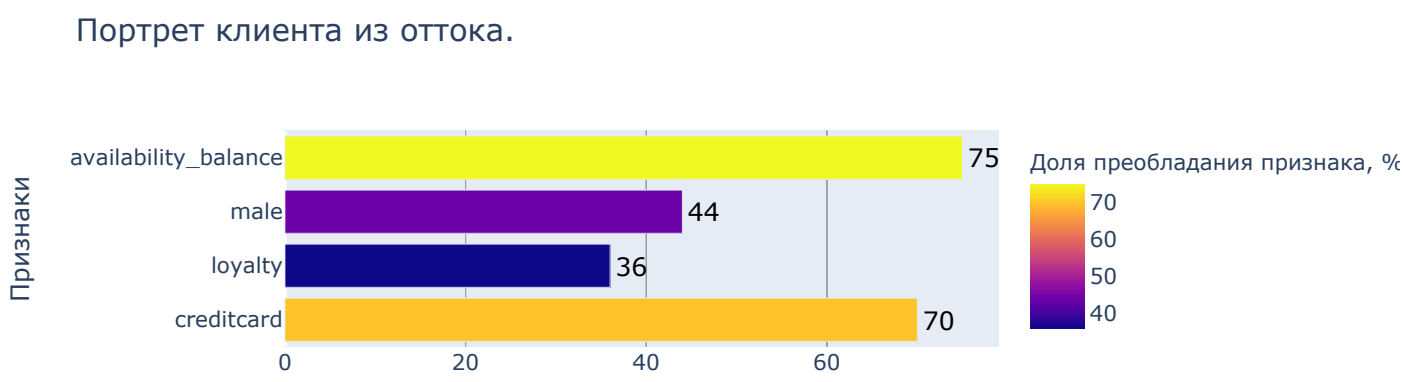
| churn | score | age | objects | balance | products | creditcard | loyalty | estimated_salary | male | availability_balance |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 651.85 | 37.41 | 5.03 | 119535.86 | 1.54 | 0.71 | 0.55 | 99738.39 | 0.57 | 0.61 |
| **1** | 645.35 | 44.84 | 4.93 | 120746.97 | 1.48 | 0.70 | 0.36 | 101465.68 | 0.44 | 0.75 |
| **Соотношение** | 0.99 | 1.20 | 0.98 | 1.01 | 0.96 | 0.99 | 0.65 | 1.02 | 0.77 | 1.24 |

**ВЫВОД:**

- Видно, что в отток уходит больше возрастных клиентов. Средний возраст **45 лет**.
- Среди ушедших доля женщин выше и составляет **56%**, что на 11% больше общебанковских значений.
- Среди ушедших в отток в программе лояльности участвовало не более **36%** клиентов, что ниже средних показателей по банку.
- Доля обладателей баланса на счете составляет **75%** среди ушедших в отток, что на 11% выше общебанковских показателей и на 24% выше показателей группы оставшихся клиентов.
- Остальные показатели в обеих группах схожи.

**Графикики распределения признаков с разбивкой на группы "ушедшие" и "оставшиеся" могут открыть дополнительную информацию о причинах отттока. Также как и ранее признаки будут отображены в 2 приема.**

In [13]:
```
# Подготовка данных для plotly.express.bar: вычисляется среднее, фильтруются признаки,
# занчения переводятся в проценты, сортируются
data_gr=data.groupby('churn').mean().round(2).T
data_g=((data_gr[data_gr[0]<=1])*100).astype('int').reset_index()
data_g=data_g.rename(columns={0: '0_churn', 1: '1_churn'})
# визуализация при помощи plotly horizontal bar chart для признаков оттока
fig = px.bar(data_g, x='1_churn', y="index",
             title='Портрет клиента из оттока.',
             text='1_churn', height=300, width=800, color='1_churn', orientation='h',
             labels={'index': 'Признаки', '1_churn': 'Доля преобладания признака, %'})
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
# визуализация при помощи plotly horizontal bar chart для признаков у оставшихся клиентов
fig = px.bar(data_g, x='0_churn', y="index",
             title='Портрет клиента после оттока.',
             text='0_churn', height=300, width=800, color='0_churn', orientation='h',
             labels={'index' : 'Признаки', '0_churn': 'Доля преобладания признака, %'})
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
```




In [14]:

```
# выделение оставшихся столбцов для построения графиков
data_histplot=['score', 'city', 'age', 'objects', 'balance', 'products', 'estimated_salary']
# Зададим параметры матрицы для отображения графиков
fig, axs = plt.subplots(4, 2, figsize = (15,15))
ax_iter=iter(axs.flat)
# Запустим цикл по построению графиков распределения признаков
for i in data_histplot:
    ax=next(ax_iter)
    sns.histplot(data=data, x=data[i], hue='churn', multiple="dodge", ax=ax, palette=('g', 'r'))
    ax.set_title(label= 'признак: '+i, loc='left', fontdict={'fontsize':14})
    ax.set_ylabel('Количество клиентов.')
    ax.set_xlabel('')
    ax.legend(['ушедшие', 'оставшиеся'])
```

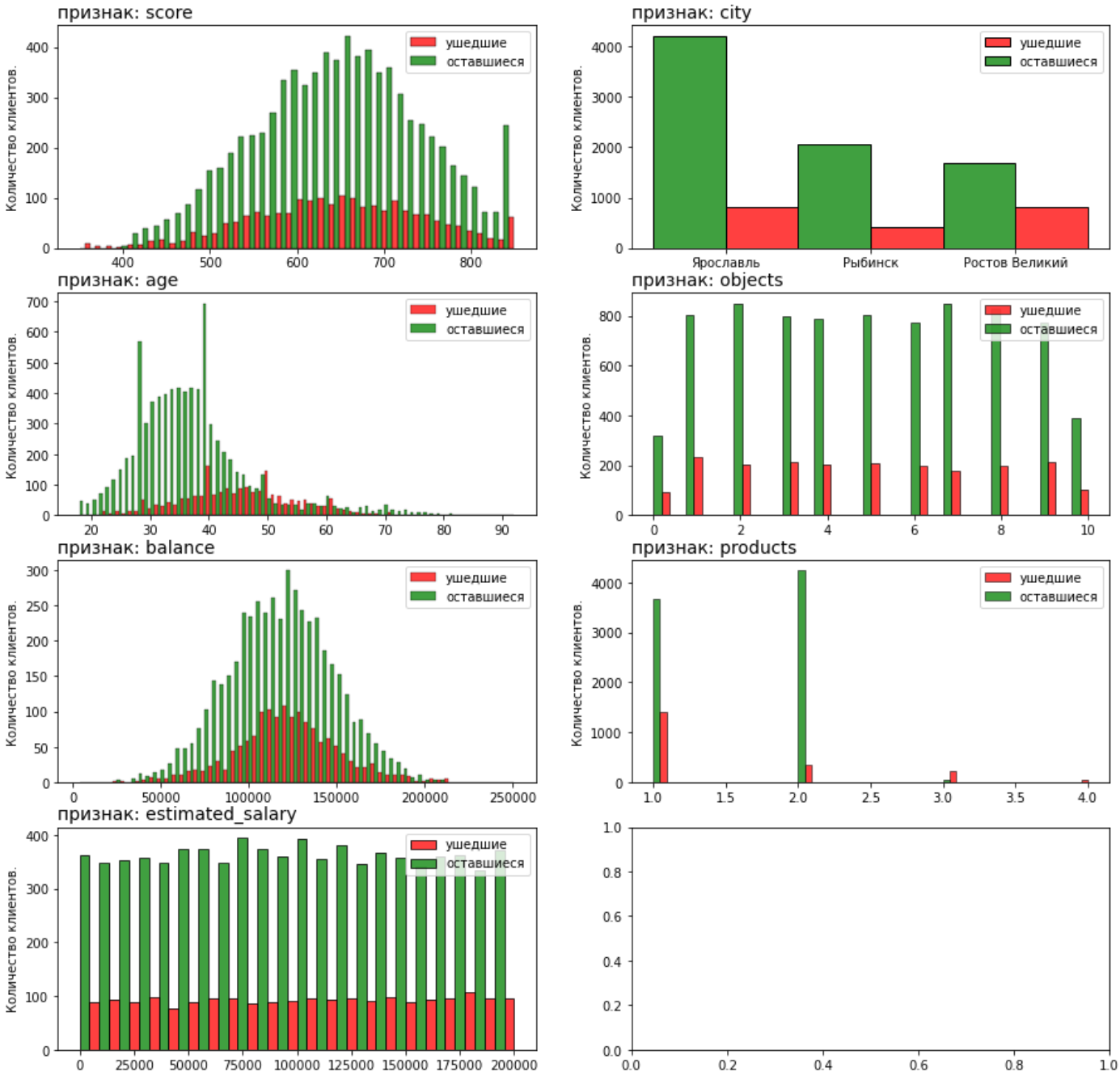
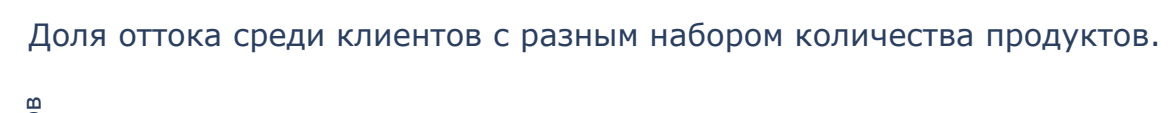
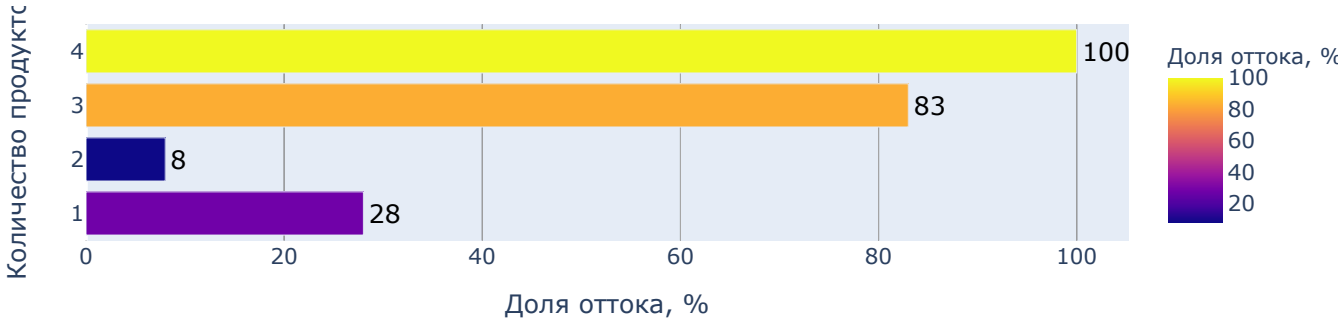
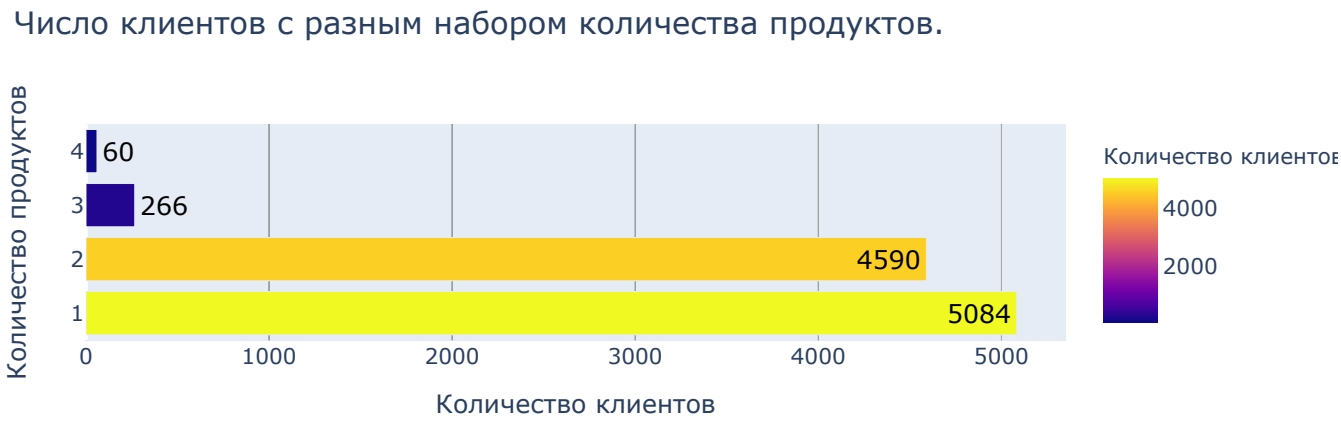



**График распределения признака `products` среди ушедших и оставшихся вызывает интерес для более детального рассмотрения под разным углом.**

In [15]:
```
# Групировка методом groupby позволит увидеть долю оттока среди обладателей разного количества продуктов,
# а также общее количество обладателей каждого набора.
d_p1=((data.groupby('products').agg({'churn': 'mean'}).round(2))*100).astype('int').reset_index()
d_p2=data.groupby('products').agg({'churn': 'count'}).reset_index()
# визуализация при помощи plotly для доли оттока
fig = px.bar(d_p1, x="churn", y="products",
             title='Доля оттока среди клиентов с разным набором количества продуктов.',
             text="churn", height=300, width=800, color='churn', orientation='h',
             labels={'products' : 'Количество продуктов', 'churn': 'Доля оттока, %'})
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
# визуализация при помощи plotly для количества клиентов с разным набором количества продуктов
fig = px.bar(d_p2, x="churn", y="products",
             title='Число клиентов с разным набором количества продуктов.',
             text="churn", height=300, width=800, color='churn', orientation='h',
             labels={'products' : 'Количество продуктов', 'churn': 'Количество клиентов'})
fig.update_traces(textposition='auto', textfont=dict(color='black', size=14))
fig.show()
```






**ВЫВОД:**

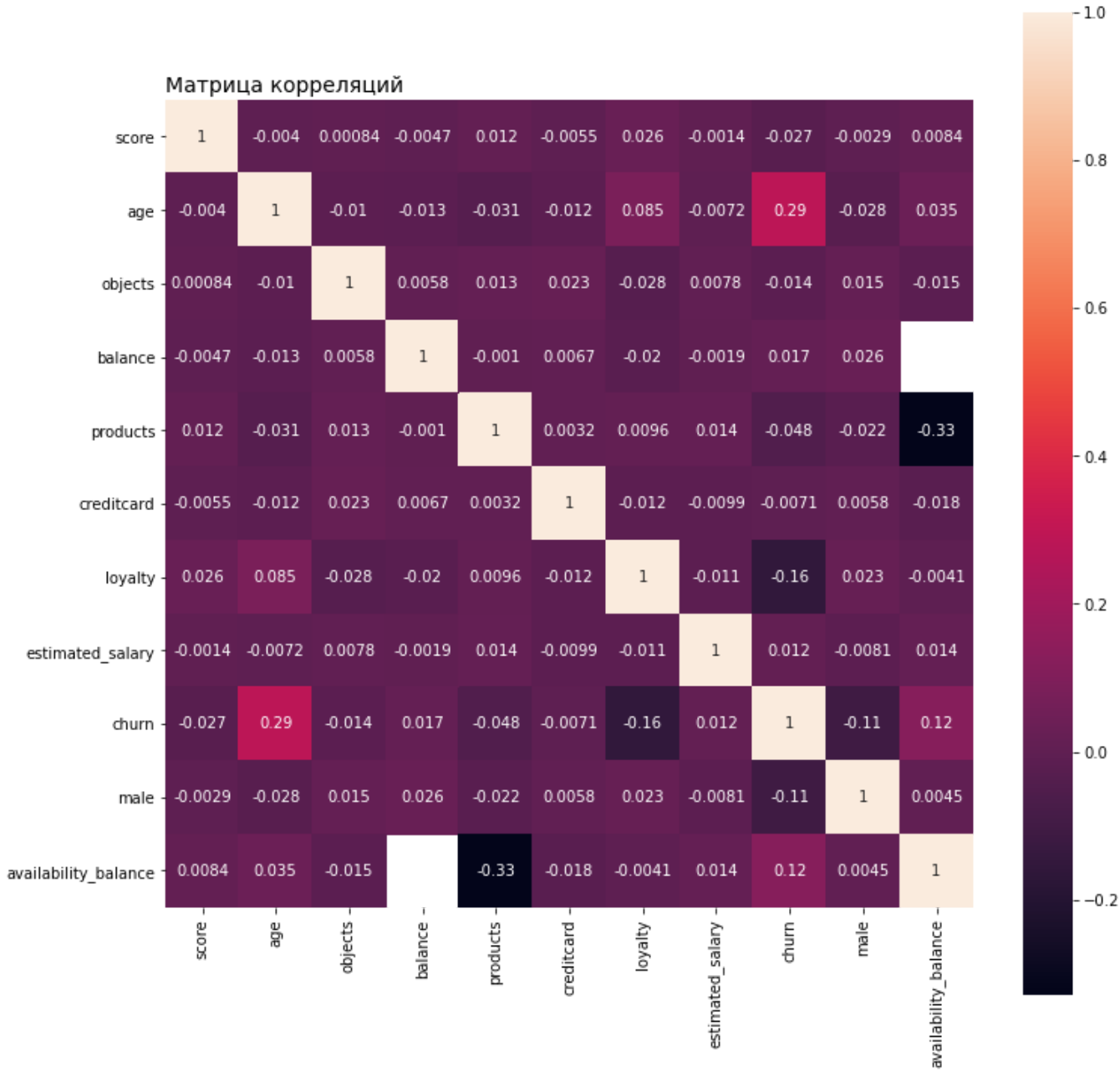
В дополнение к описанному выше графики показывают:

- Распределение кредитного рейтинга схоже в обеих группах.
- Доля оттока **в Ростове Великом** значительно **выше** чем в других городах.
- Большая часть ушедших клиентов сконцентрирована около 40-50 лет, тогда как оставшаяся часть - около 30-40 лет.
- Среди клиентов, которые пользуются 2-мя продуктами отток (**около 8%**) ниже чем среди тех, у кого только 1 продукт (**28%**). При этом численность пользователей обоих наборов продуктов сопоставима. Отток клиентов среди обладателей более чем 2 продуктов гораздо выше, но их численность невелика, чтобы сделать полноценный вывод о закономерностях такого явления, но можно предположить, что есть продукты банка, которые вызывают негатив у клиентов.
- Распределение предполагаемой заработной платы клиентов в обеих группах имеет схожие черты.

**Корреляционный анализ поможет выявить возможно скрытые взаимосвязи между признаками**.

In [16]:
```
# вычисление корреляцию.
corr_m = data.corr()
# размеры отображения матрицы
plt.figure(figsize=(12, 12))
# построение тепловой карты
sns.heatmap(data=corr_m, square = True, annot = True)\
    .set_title("Матрица корреляций", loc='left', fontdict={'fontsize':14})
```

Out[16]:
```
Text(0.0, 1.0, 'Матрица корреляций')
```




**ВЫВОД:**

- Несколько признаков слабо, но коррелируют с целевой переменной, оттоком.
  - Положительная корреляция:
    - на увеличение оттока влияет возраст клиентов и наличие баланса на счете.
  - Отрицательная корреляция:
    - с увеличением числа клиентов-мужчин снижается отток,
    - с увеличением участников программ лояльности также снижается отток.
- Слабая отрицательная корреляция есть и между `availability_balance` и `products`, что может говорить о увеличении одного признака при уменьшении другого.

**Нужно будет рассмотреть подробней характер взаимосвязи `products` с другими признаками, так как мы уже знаем, что доля оттока среди обладателей разного набора продуктов отличается. Взгляд на средние значения признаков позволит выделить особенностикаждой группы.**

In [17]:
```
# Группировка с groupby по столбцу 'products' и указанием средних признаков.
data.groupby('products').mean().round(2)
```

Out[17]:

| products | score | age | objects | balance | creditcard | loyalty | estimated_salary | churn | male | availability_balance |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 649.12 | 39.67 | 4.97 | 119894.16 | 0.70 | 0.50 | 99487.26 | 0.28 | 0.55 | 0.82 |

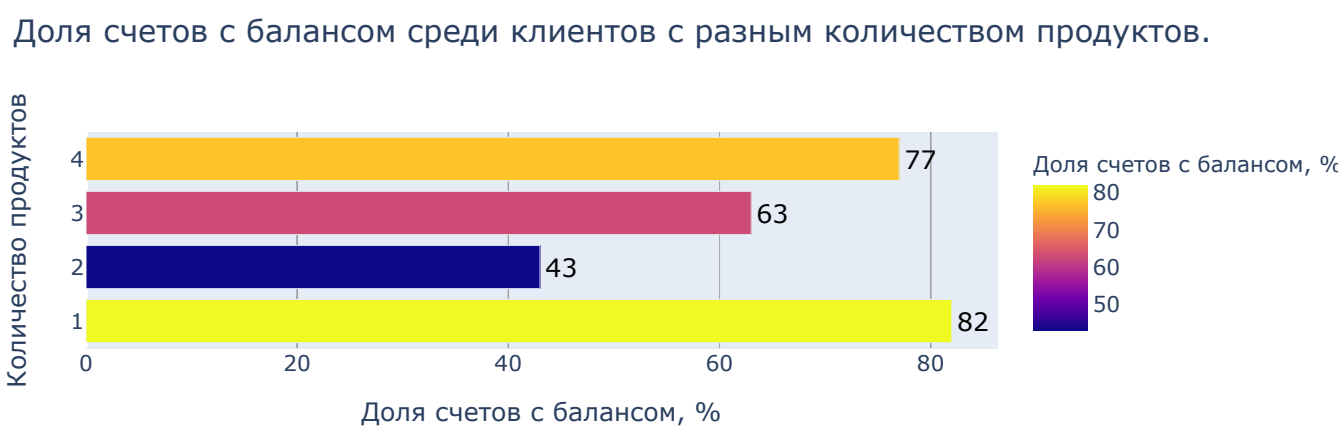
| products | score | age | objects | balance | creditcard | loyalty | estimated_salary | churn | male | availability_balance |
|---|---|---|---|---|---|---|---|---|---|---|
| **2** | 652.19 | 37.75 | 5.05 | 119660.94 | 0.71 | 0.53 | 100452.01 | 0.08 | 0.55 | 0.43 |
| **3** | 648.11 | 43.20 | 5.00 | 119475.69 | 0.71 | 0.42 | 104318.13 | 0.83 | 0.44 | 0.63 |
| **4** | 653.58 | 45.68 | 5.30 | 122260.61 | 0.68 | 0.48 | 104763.72 | 1.00 | 0.37 | 0.77 |

**Вывод:**

- У самой малочисленной группы, клиентов с 3-4 продуктами выше среднего по банку следующие признаки:
  - средний возраст на 5-7 лет,
  - доля женской аудитории на 11-18%
- Среди клиентов с 1-2 продуктами, которые составляют костяк клиентской базы,основные параметры сопоставимы с общебанковскими.
- Заметно, что низкая доля оттока среди клиентов с 2 продуктами коррелируется с низким уровнем обладателей баланса на счете, также как и более высокие доли оттока среди клиентов с другими наборами продуктов показывают связь с повышением доли обладателей баланса на счете.
- Можно предположить, что речь идет о новой банковской услуге, которую банк активно продвигает, привлекая новых клиентов и предлагая ее существующим. Этим можно объяснить высокую долю проникновения среди клиентов у которых уже есть несколько продуктов (3-4), а следовательно и лояльность к предложениям банка выше.

**Как выглядит на графике распределение признака `availability_balance` по количеству продуктов?**

In [18]:
```
# Группировка методом groupby позволит увидеть долю обладателей баланса на счете среди
# клиентов с разным набором количества продуктов.
d_a=((data.groupby('products').agg({'availability_balance': 'mean'}).round(2))*100).astype('int').reset_ind
# визуализация при помощи plotly для доли обладателей баланса на счете
fig = px.bar(d_a, x="availability_balance", y="products",
             title='Доля счетов с балансом среди клиентов с разным количеством продуктов.',
             text="availability_balance", height=300, width=800, color='availability_balance', orientation='
             labels={'products' : 'Количество продуктов', 'availability_balance': 'Доля счетов с балансом, %
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
```




# 4. Отличительные черты клиентов в городах присутствия банка.

Наверх

**Взгляд на средние** *значения признаков сгруппированных по городам* **позволит определить есть ли различия в ситуации с оттоком клиентов.**

In [19]:
```
# Группировка с groupby по столбцу 'city' и указанием средних признаков.
data.groupby('city').mean().round(2)
```

Out[19]:

| city | score | age | objects | balance | products | creditcard | loyalty | estimated_salary | churn | male | availability_balance |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ростов Великий** | 651.45 | 39.77 | 5.01 | 119730.12 | 1.52 | 0.71 | 0.50 | 101113.44 | 0.32 | 0.52 | 1.00 |
| **Рыбинск** | 651.33 | 38.89 | 5.03 | 119814.99 | 1.54 | 0.69 | 0.53 | 99440.57 | 0.17 | 0.56 | 0.52 |
| **Ярославль** | 649.67 | 38.51 | 5.00 | 119927.77 | 1.53 | 0.71 | 0.52 | 99899.18 | 0.16 | 0.55 | 0.52 |

**ВЫВОД:**

- Как ранее уже было отмечено на графиках распределения признаков в Ростове Великом самая высокая доля оттока. Видно, что она составляет **32%** тогда как в Рыбинске и Ярославле это показатель равен **17% и 16% соответственно**.
- Кроме того, в Ростове Великом все **100%** клиентов имеют балансна счете, тогда как в Рыбинске и Ярославле он есть только у **52%**.
- Все остальные средние признаки схожи во всех городах.

**Графики сделают различающиеся признаки более наглядными.**

In [20]:

```
# Групировка методом groupby позволит увидеть общее количество клиентов,
# а также долю обладателей баланса на счете в разных городах
d_city_count=data.groupby('city').agg({'churn': 'count'}).reset_index()
data_city_ratio=((data.groupby('city').agg({'churn': 'mean'}).round(2))*100).astype('int').reset_index()
data_balance_ratio=((data.groupby('city').agg({'availability_balance': 'mean'}).round(2))*100).astype('int'
# визуализация при помощи plotly для количества клиентов в разных городах
fig = px.bar(d_city_count, x="churn", y="city",
             title='Число клиентов в городах.',
             text="churn", height=300, width=800, color='churn', orientation='h',
             labels={'city' : 'Города', 'churn': 'Количество клиентов'})
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
# визуализация при помощи plotly для доли обладателей счета с балансом в разных городах
fig = px.bar(data_city_ratio, x="churn", y="city",
             title='Доли оттока в разных городах.',
             text="churn", height=300, width=800, color='churn', orientation='h',
             labels={'city' : 'Города', 'churn': 'Доля оттока, %'})
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
# визуализация при помощи plotly для доли обладателей счета с балансом в разных городах
fig = px.bar(data_balance_ratio, x="availability_balance", y="city",
             title='Доли обладателей счета с балансом в разных городах.',
             text="availability_balance", height=300, width=800, color='availability_balance', orientation='
             labels={'city' : 'Города', 'availability_balance': 'Доля обладателей счета с балансом, %'})
fig.update_traces(textposition='outside', textfont=dict(color='black', size=14))
fig.show()
```

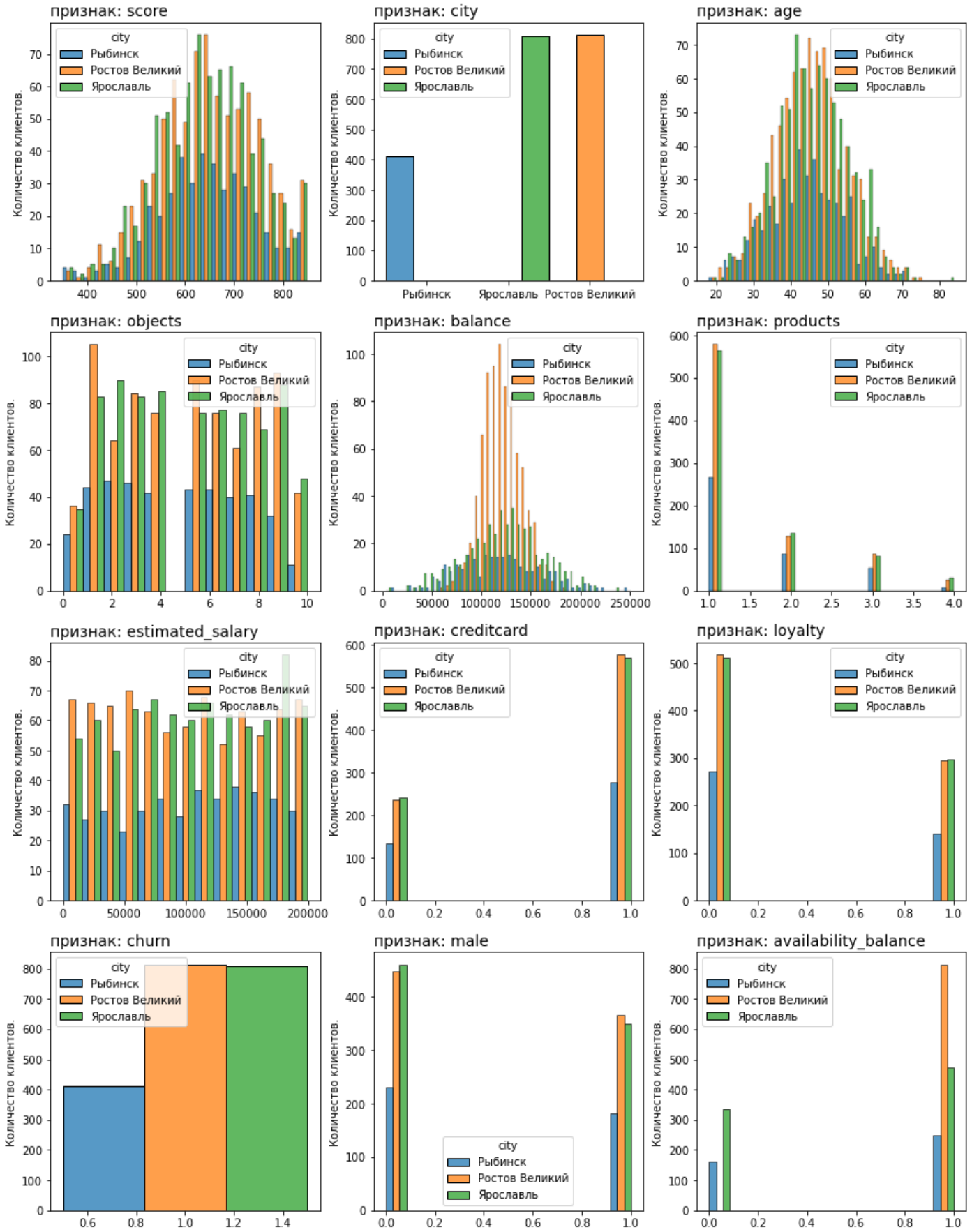


**ВЫВОД:**

- Ростов Великий, не являясь самым большим по размеру клиентской базы городом, между тем лидирует по показателям оттока и доле обладателей баланса насчете.

**Есть ли различия в портрете ушедшего клиента в разных городах?**

In [21]:
```
data_histplot=['score', 'city', 'age', 'objects', 'balance', 'products', 'estimated_salary',
               'creditcard', 'loyalty', 'churn', 'male', 'availability_balance']
# Зададим параметры матрицы для отображения графиков
fig, axs = plt.subplots(4, 3, figsize = (15,20))
ax_iter=iter(axs.flat)
# Запустим цикл по построению графиков распределения признаков
for i in data_histplot:
    ax=next(ax_iter)
    sns.histplot(data=data.query('churn == 1'), x=data[i], hue='city', multiple="dodge", ax=ax)
    ax.set_title(label= 'признак: '+i, loc='left', fontdict={'fontsize':14})
    ax.set_ylabel('Количество клиентов.')
    ax.set_xlabel('')
```




**ВЫВОД**:

- Графики наглядно подтверждают выводы, сделанные ранее по средним значениям:
  - Портреты клиентов, ушедших из банка в разных городах схожи по большинству признаков.
  - Различие состоит в количестве обладателей баланса на счете.

# 5. Проверка гипотез.

Наверх

**Для проверки гипотезы о наличии различии дохода между теми клиентами, которые ушли и теми, которые остались необходимо:**

1. *подготовить 2 выборки, одна с данными о доходе ушедших клиентов, другая с данными дохода оставшихся клиентов,*
2. *проверить выборки на нормальность,*
3. *проверить гипотезу о равенстве дисперсий выборок,*
4. *определить метод для проверки основных гипотез.*
5. *сформулировать основные гипотезы.*
6. _проверить выборки используя выбранный критерий.

## 1. Подготовка выборок для проверки гипотез.

In [22]:
```
# Срез данных по целевой переменной даст 2 выборки.
salary_churn=data.query('churn == 1')['estimated_salary'] # выборка с предполагаемым доходом ушедших
salary_rest=data.query('churn == 0')['estimated_salary'] # выборка с предполагаемым доходом оставшихся
```

In [23]:
```
# Использование метода describe() для сбора информации о выборках в одном датафрейме
# С помощью describe() и reset_index() создается датафрейм с данными первой выборки
data_describe=salary_churn.describe().reset_index().round(2)
# Переименование столбца
data_info=data_describe.rename(columns={'estimated_salary': 'salary_churn'})
# Добавление в датафрейм результата метода describe(), примененного ко второй выборке
data_info['salary_rest']=salary_rest.describe().reset_index()['estimated_salary'].round(2)
data_info #вывод
```

Out[23]:

| | index | salary_churn | salary_rest |
|---|---|---|---|
| **0** | count | 2037.00 | 7963.00 |
| **1** | mean | 101465.68 | 99738.39 |
| **2** | std | 57912.42 | 57405.59 |
| **3** | min | 11.58 | 90.07 |
| **4** | 25% | 51907.72 | 50783.49 |
| **5** | 50% | 102460.84 | 99645.04 |
| **6** | 75% | 152422.91 | 148609.96 |
| **7** | max | 199808.10 | 199992.48 |

**ВЫВОД:**

Пулучено 2 выборки.

- Выборка с предполагаемым доходом ушедших `salary_churn` содержит 2037 значений. Вторая выборка `salary_rest`, содержащая данные о доходе оставшихся, больше почти в 4 раза (7963 значения).
- Обе выборки имеют схожие средние значения и стандартные отклонения.

## 2. Проверка гипотез о нормальности выборок, используя *критерий Шапиро-Уилк и критерий согласия Пирсона.*

Формулировка вспомогательных гипотез:

*Нулевая гипотеза $H_0$:* **Распределение в выборках нормально.**

*Альтернативная гипотеза $H_1$:* **Распределение в выборках не нормально.**

In [24]:
```
# Создание функции для проверки гипотез о нормальности выборок
def checking_for_normality(sample):
    alpha = .05 # критический уровень статистической значимости

    results1 = st.shapiro(sample) # критерий Шапиро-Уилк
    p_value1 = results1[1] # второе значение в массиве результатов (с индексом 1) - p-value

    results2 = st.normaltest(sample) # критерий согласия Пирсона
```

```
    p_value2 = results2[1] # второе значение в массиве результатов (с индексом 1) - p-value
# вывод результатов
    print('Критерий Шапиро-Уилк, p-значение: ', p_value1)
    if (p_value1 < alpha):
        print("Критерий Шапиро-Уилк: Отвергаем нулевую гипотезу: распределение не нормально")
    else:
        print("Критерий Шапиро-Уилк: Не получилось отвергнуть нулевую гипотезу, всё нормально")

    print("")

    print('Критерий согласия Пирсона, p-значение: ', p_value2)
    if (p_value2 < alpha):
        print("Критерий согласия Пирсона: Отвергаем нулевую гипотезу: распределение не нормально")
    else:
        print("Критерий согласия Пирсона: Не получилось отвергнуть нулевую гипотезу, всё нормально")
```

**Проверяем выборку содержащую данные о доходах** *ушедших клиентов.*

In [25]:
```
# использование функции с первой выборкой
checking_for_normality(salary_churn)
```

```
Критерий Шапиро-Уилк, p-значение:  4.594104044877579e-25
Критерий Шапиро-Уилк: Отвергаем нулевую гипотезу: распределение не нормально

Критерий согласия Пирсона, p-значение:  0.0
Критерий согласия Пирсона: Отвергаем нулевую гипотезу: распределение не нормально
```

**Проверяем выборку содержащую данные о доходах** *оставшихся клиентов.*

In [26]:
```
# использование функции со второй выборкой
checking_for_normality(salary_rest)
```

```
Критерий Шапиро-Уилк, p-значение:  3.797518838320254e-43
Критерий Шапиро-Уилк: Отвергаем нулевую гипотезу: распределение не нормально

Критерий согласия Пирсона, p-значение:  0.0
Критерий согласия Пирсона: Отвергаем нулевую гипотезу: распределение не нормально
C:\Users\Alex\anaconda3\lib\site-packages\scipy\stats\morestats.py:1681: UserWarning:

p-value may not be accurate for N > 5000.
```

## 3. Сравнение дисперсий выборок.

Формулировка вспомогательных гипотез:

Нулевая гипотеза $H_0$: Дисперсии двух выборок не различаются.

Альтернативная гипотеза $H_1$: Дисперсии выборок различаются.

Если нормальности нет, годится тест Флигнера-Киллина

In [27]:
```
alpha = .05 # критический уровень статистической значимости
results = fligner(salary_churn, salary_rest) # применение теста Флигнера-Киллина
# вывод результата
print('p-значение: ', results.pvalue)

if (results.pvalue < alpha):
    print("Отвергаем нулевую гипотезу: дисперсии различаются")
else:
    print("Не получилось отвергнуть нулевую гипотезу, дисперсии не различаются")
```

```
p-значение:  0.34153963766212375
Не получилось отвергнуть нулевую гипотезу, дисперсии не различаются
```

**ВЫВОД:** Большое значение p говорит о том, что нулевую гипотезу невозможно отбросить и дисперсии двух выборок совпадают.

## 4. Определение метода для проверки основных гипотез.

Распределение в обеих выборках не нормально. Дисперсии не различаются. Обе выборки достаточно большие, более 2000 значений. В условиях того, что распределение в обеих выборках не нормально, использовать Т-тест (или

тест Стьюдента) не получится. Альтернативой ему служит тест Манна-Уитни, основанный на структурном подходе, или непараметрический.

## 5. Формулировка основных гипотез.

*Нулевая гипотеза $H_0$:* **Нет различий в распределении признака в двух выборках.**

*Альтернативная гипотеза $H_1$:* **Есть различия в распределении признака в двух выборках.**

## 6. Проверка различий в выборках.

In [28]:
```
alpha = .05 # критический уровень статистической значимости
results = st.mannwhitneyu(salary_churn, salary_rest) # применение теста Манна-Уитни
# вывод результата
print('p-значение: ', 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:  0.11352575465076892
Не получилось отвергнуть нулевую гипотезу, вывод о различии сделать нельзя
```

**ВЫВОД:**

P-value значительно больше значения alpha равного 0.05. Значит, причин отвергать нулевую гипотезу нет и можем считать, что между группами ушедших и оставшихся клиентов нет статистически значимых различий распределения признака, предпалагаемого дохода. С вероятностью в почти 12% такие различия можно получить случайно.

# 6. Выводы.

Наверх

- **ОБЩИЕ ВЫВОДЫ:**
  - Для ушедших клиентов банка характерны следующие признаки:
    - Общий уровень оттока по банку составляет **20%**.
    - Лидером по оттоку с **32%** является Ростов Великий.
    - Среди ушедших клиентов преобладают люди **40-50 лет**.
    - Доля мужчин в оттоке **44%**, что ниже среднего по банку на 11%.
    - Доля участников программы лояльности в оттоке **36%**, что на 16% ниже чем было в целом по банку.
    - Почти все клиенты, у которых более 2-х продуктов, а значит должны быть самыми лояльными, ушли. Это может говорить о том, что произошедшие в банке изменения вызвали негативную реакцию у постоянных клиентов, которых и так было очень мало.
    - Преобладающий состав клиентов имеет 1-2 банковских продукта, а значит можем заключить, что вероятность оттока ниже, если клиенты пользуются не одной банковской услугой.
    - Доля обладателей баланса на счете различается по городам. Возможно имеет место чересчур активная маркетинговая активность продвижения банковского продукта, по которому опять же Ростов Великий лидирует, выполнив план на 100%. Агрессивный маркетинг или разочарование от нового продукта вероятно повлияли на отток.
- **РЕКОМЕНДАЦИИ:**
  - Продолжить развивать программу лояльности, так как анализ показал меньший отток среди участников программы.
  - Разработать программы удержания клиентов старше 40 лет, а также женской аудитории.
  - Увеличивать количество используемых продуктов клиентами как минимум до 2-х, ориентируясь на конкретные потребности клиента, избегая чрезнерного давления.
  - Чтобы выявить дополнительно слабые стороны в продуктовой линейке или в сервисе банка,

- провести опрос бывших клиентов о причинах ухода,
- провести анализ оттока по всем ключевым банковским продуктам.

## 7. Ссылка на презентацию.

Наверх

https://1drv.ms/b/s!AisE2sdC3isfgbQXJMzYA2jlGyKVjg

## 8. Ссылка на дашборд.

Наверх

https://public.tableau.com/views/___16141083640810/Dashboard1?:language=en&:display_count=y&publish=yes&:origin=viz_