Практическое задание по клиентской аналитике

(итоговые данные можно предоставить в документе ворд или pdf, сильно расписывать не надо, просто выводы по пунктам)

Действия:

1. Залить в свою БД данные по продажам (часть таблицы Orders в csv, исходник здесь https://drive.google.com/drive/folders/1C3HqIJcABblKM2tz8vPGiXTFT7MisrML?usp=sharing (https://drive.google.com/drive/folders/1C3HqIJcABblKM2tz8vPGiXTFT7MisrML?usp=sharing))
2. Проанализировать, какой период данных выгружен
3. Посчитать кол-во строк, кол-во заказов и кол-во уникальных пользователей, кот совершали заказы.
4. По годам посчитать средний чек, среднее кол-во заказов на пользователя, сделать вывод , как изменялись это показатели Год от года.
5. Найти кол-во пользователей, кот покупали в одном году и перестали покупать в следующем.
6. Найти ID самого активного по кол-ву покупок пользователя.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
%config InlineBackend.figure_format = 'svg'
```

In [2]:

```
orders = pd.read_csv('orders_20190822.csv', sep=';')
a = orders.shape[0]
print(f'Первоначальное количество строк = {a}')
```

```
Первоначальное количество строк = 2002804
```

Конвертация данных

In [3]:

```
def convert_data(data):
    new_data = data.replace(',','.')
    return float(new_data)
```

In [4]:

```
orders['price'] = orders['price'].apply(convert_data)
```

In [5]:

```
orders['o_date'] = pd.to_datetime(pd.Series(orders['o_date']), format="%d.%m.%Y")
```

Убираем строки с неположительной и слишком большой ценой

In [6]:

```
negative_list = orders.loc[(orders['price'] <= 0)|(orders['price'] > 100000)]
orders = orders.drop(negative_list.index, axis = 0)
```

1. Проанализировать, какой период данных выгружен

In [7]:

```
orders['o_date'].max(), orders['o_date'].min()
```

Out[7]:

```
(Timestamp('2017-12-31 00:00:00'), Timestamp('2016-01-01 00:00:00'))
```

2. Посчитать кол-во строк, кол-во заказов и кол-во уникальных пользователей, кот совершали заказы.

In [8]:

```
b = orders.shape[0]
print(f'Количество строк после удаления некорректных = {b}')
```

```
Количество строк после удаления некорректных = 2002696
```

In [9]:

```
print(f'После того, как были убраны строки с неположительной и слишком большой ценой ко
```

```
После того, как были убраны строки с неположительной и слишком большой ценой
количество строк сократилось на 108
```

In [10]:

```
number_of_orders = orders['id_o'].unique()
len(number_of_orders)
```

Out[10]:

```
2002696
```

In [11]:

```
number_of_users = orders['user_id'].unique()
len(number_of_users)
```

Out[11]:

```
1015088
```

3. По годам посчитать средний чек, среднее кол-во заказов на пользователя, сделать вывод , как изменялись это показатели Год от года.

In [12]:

```
mean_order = orders.groupby(orders['o_date'].dt.year, as_index=False)[['price']].mean(
mean_order
```

Out[12]:

	mean_price
0	2090.294426
1	2392.783860

In [13]:

```
users_2016 = orders.loc[orders['o_date'].dt.year == 2016, 'user_id'].values
unique_users_2016 = orders.loc[orders['o_date'].dt.year == 2016, 'user_id'].unique()
order_per_user_2016 = len(users_2016) / len(unique_users_2016)

users_2017 = orders.loc[orders['o_date'].dt.year == 2017, 'user_id'].values
unique_users_2017 = orders.loc[orders['o_date'].dt.year == 2017, 'user_id'].unique()
order_per_user_2017 = len(users_2017) / len(unique_users_2017)

order_per_user_2016, order_per_user_2017
```

Out[13]:

```
(1.935104238984639, 1.742957402362599)
```

Вывод:

средний чек вырос с 2090 руб. в 2016 г. до 2393 руб. в 2017 г., среднее количество заказов на покупателя упало с 1,93 в 2016 г. до 1,74 в 2017 г.

4. Найти кол-во пользователей, кот покупали в одном году и перестали покупать в следующем.

In [14]:

```
ex_users = []
for i in unique_users_2016:
    if i not in unique_users_2017:
        ex_users.append(i)
len(ex_users)
```

Out[14]:

```
360216
```

5.Найти ID самого активного по кол-ву покупок пользователя.

К сожалению, на всей БД из более чем 2 млн.строк мой компьютер исполнение кода не потянул

даже за несколько часов, поэтому специально вырезаю из исходного более короткий датафрейм на 100 тыс. строк и отрабатываю на нём.

In [18]:

```
orders = orders[0:100000]
```

In [19]:

```
max_value = 1
max_user_id = 1

for i in orders['user_id']:
    if (orders['user_id'] == i).sum() > max_value:
        max_value = (orders['user_id'] == i).sum()
        max_user_id = i
        a = orders[orders['user_id'] == i].index
        orders['user_id'].drop(a)

max_value, max_user_id
```

Out[19]:

```
(65, 765861)
```

In [20]:

```
print(f'На выборке из первых 100000 строк победил user #{max_user_id}, который произвё
```

```
На выборке из первых 100000 строк победил user #765861, который произвёл 65
покупок.
```

In [ ]:

```

```