In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import time
import datetime
```

In [2]:

```
df = pd.read_csv('orders_all.csv',sep=';', thousands=',', parse_dates=[3])
print('В файле содержится ', len(df), 'записей.')
print('Среди них имеется ', df[df['o_date']=='00.00.0000']['price'].value_counts
()[0],'с датой 00.00.0000')
```

```
В файле содержится  4365731 записей.
Среди них имеется  55492 с датой 00.00.0000
```

In [3]:

```
df.head()
```

Out[3]:

	id_order	id_user	price	o_date
0	129	1	1337	26.04.2013
1	130	155	182	26.04.2013
2	131	1	602	26.04.2013
3	132	1	863	26.04.2013
4	133	1	2261	29.04.2013

I. Без удаления "нулевых" дат

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

In [4]:

```
df.loc[df['o_date']=='00.00.0000','o_date'] = np.nan
df['o_date'] = pd.to_datetime(df['o_date'],dayfirst=True)
print(f"В файле csv представлены данные за {str(df[df['o_date']!='00.00.0000']
['o_date'].min()).split(' ')[0]} - {str(df['o_date'].max()).split(' ')[0]}")
print(f"Всего {df.shape[0]} строк.")
print(f"Всего {len(set(df['id_order']))} заказов.")
print(f"Всего {len(set(df[df['price']>0]['id_user']))} уникальных пользователей,
совершивших заказ (price > 0).")
```

```
В файле csv представлены данные за 2013-04-26 - 2018-12-27
Всего 4365731 строк.
Всего 4365731 заказов.
Всего 2146690 уникальных пользователей, совершивших заказ (price >
0).
```

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

In [5]:

```
df['price'].groupby(df['o_date'].dt.year).mean()
```

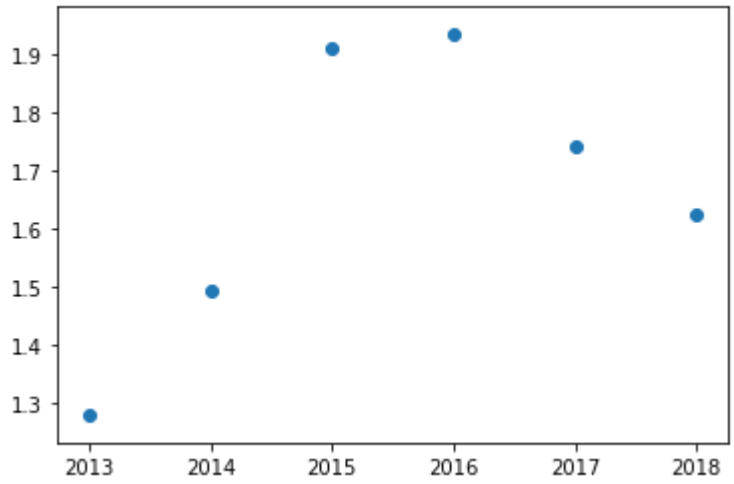
Out[5]:

```
o_date
2013.0    2.071190e+11
2014.0    2.287714e+03
2015.0    2.018164e+03
2016.0    2.095945e+03
2017.0    2.398827e+03
2018.0    2.302783e+03
Name: price, dtype: float64
```

In [6]:

```
print('среднее количество заказов на пользователя ',round(df[['id_user','id_orde
r']].groupby('id_user').count().mean()[0],2))
stat_list = []
year_list = []
for i in df['o_date'].dt.year.unique():
    x = df.loc[pd.DatetimeIndex(df['o_date']).year == i,['o_date','id_user']].gr
oupby('id_user').count().mean()
    print(f'Среднее количество заказов на каждого пользователя в {i} году = {x}'
    )
    stat_list.append(x)
    year_list.append(i)
plt.scatter(year_list,stat_list)
plt.show()
```

```
среднее количество заказов на пользователя  2.03
Среднее количество заказов на каждого пользователя в 2013.0 году = o
_date    1.281306
dtype: float64
Среднее количество заказов на каждого пользователя в 2014.0 году = o
_date    1.495294
dtype: float64
Среднее количество заказов на каждого пользователя в 2015.0 году = o
_date    1.912768
dtype: float64
Среднее количество заказов на каждого пользователя в 2016.0 году = o
_date    1.935209
dtype: float64
Среднее количество заказов на каждого пользователя в 2017.0 году = o
_date    1.742966
dtype: float64
Среднее количество заказов на каждого пользователя в nan году = o_da
te   NaN
dtype: float64
Среднее количество заказов на каждого пользователя в 2018.0 году = o
_date    1.6247
dtype: float64
```



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

In [7]:

```
%%time

df['o_date_year'] = df['o_date'].dt.year
df_1 = df[['o_date_year','id_user','id_order']].groupby(['id_user','o_date_year'
]).count()
list_years = list(pd.DatetimeIndex(df['o_date']).year.unique())
check_list = set() # сначала - пустой список (клиентов не было: первый год - нет
разницы с предыдущим, значит, и вывода не будет)
for i in list_years:
    # во втором цикле: показать пользователей (check_list из предыдущего цикла),
которые покупали в предыдущий год и ничего не купили в данном году (список польз
ователей из текущего цикла)
    check_list_inactive =  check_list - set(df_1.loc[(slice(None),[i]),:].index.
get_level_values(0))
    if len(check_list_inactive)>0:
        print(f'в {i} году не стали ничего покупать: {len(check_list_inactive)}
 пользователей.')
        pass
    # i-год
    check_list = set(df_1.loc[(slice(None),[i]),:].index.get_level_values(0))
```

```
в 2014.0 году не стали ничего покупать: 23709 пользователей.
в 2015.0 году не стали ничего покупать: 130076 пользователей.
в 2016.0 году не стали ничего покупать: 233546 пользователей.
в 2017.0 году не стали ничего покупать: 360225 пользователей.
в nan году не стали ничего покупать: 654894 пользователей.
CPU times: user 2.88 s, sys: 276 ms, total: 3.16 s
Wall time: 3.16 s
```

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

In [8]:

```
print(f'Максимальное количество покупок совершил пользователь с ID = {df["id_use
r"].value_counts().max()}')
```

```
Максимальное количество покупок совершил пользователь с ID = 53010
```

II. После удаления "нулевых дат"

т.к. при всех значениях o_date = 00.00.0000 price = 0, строки с этими датами можно удалить - это не реальные заказы скорее всего нереальных юзеров в нереальные даты

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

In [9]:

```
df.dropna(subset = ['o_date'],inplace=True)
print(f"В файле csv представлены данные за {str(df[df['o_date']!='00.00.0000']
['o_date'].min()).split(' ')[0]} - {str(df['o_date'].max()).split(' ')[0]}")
print(f"Всего {df.shape[0]} строк.")
print(f"Всего {len(set(df['id_order']))} заказов.")
print(f"Всего {len(set(df[df['price']>0]['id_user']))} уникальных пользователей,
совершивших заказ (price > 0).")
```

```
В файле csv представлены данные за 2013-04-26 - 2018-12-27
Всего 4310239 строк.
Всего 4310239 заказов.
Всего 2146690 уникальных пользователей, совершивших заказ (price >
0).
```

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

In [10]:

```
df['price'].groupby(df['o_date'].dt.year).mean()
```

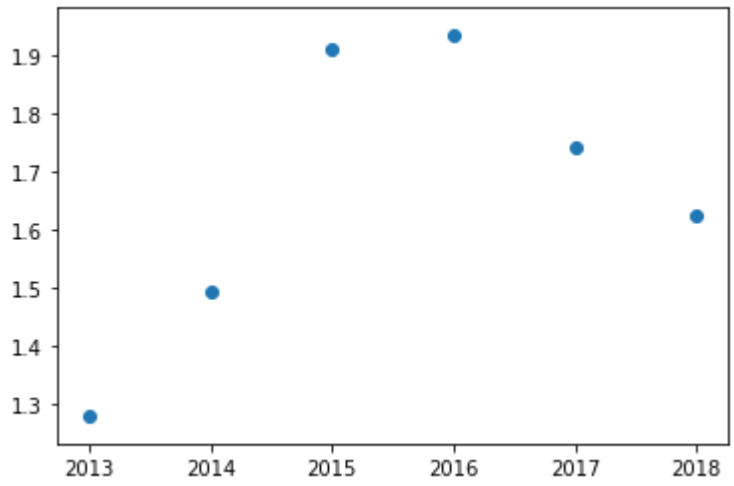
Out[10]:

```
o_date
2013    2.071190e+11
2014    2.287714e+03
2015    2.018164e+03
2016    2.095945e+03
2017    2.398827e+03
2018    2.302783e+03
Name: price, dtype: float64
```

In [11]:

```
print('среднее количество заказов на пользователя ',round(df[['id_user','id_orde
r']].groupby('id_user').count().mean()[0],2))
stat_list = []
year_list = []
for i in df['o_date'].dt.year.unique():
    x = df.loc[pd.DatetimeIndex(df['o_date']).year == i,['o_date','id_user']].gr
oupby('id_user').count().mean()
    print(f'Среднее количество заказов на каждого пользователя в {i} году = {x}'
    )
    stat_list.append(x)
    year_list.append(i)
plt.scatter(year_list,stat_list)
plt.show()
```

```
среднее количество заказов на пользователя  2.01
Среднее количество заказов на каждого пользователя в 2013 году = o_d
ate    1.281306
dtype: float64
Среднее количество заказов на каждого пользователя в 2014 году = o_d
ate    1.495294
dtype: float64
Среднее количество заказов на каждого пользователя в 2015 году = o_d
ate    1.912768
dtype: float64
Среднее количество заказов на каждого пользователя в 2016 году = o_d
ate    1.935209
dtype: float64
Среднее количество заказов на каждого пользователя в 2017 году = o_d
ate    1.742966
dtype: float64
Среднее количество заказов на каждого пользователя в 2018 году = o_d
ate    1.6247
dtype: float64
```



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

In [12]:

```
%%time

df['o_date_year'] = df['o_date'].dt.year
df_1 = df[['o_date_year','id_user','id_order']].groupby(['id_user','o_date_year'
]).count()
list_years = list(pd.DatetimeIndex(df['o_date']).year.unique())
check_list = set() # сначала - пустой список (клиентов не было: первый год - нет
разницы с предыдущим, значит, и вывода не будет)
for i in list_years:
    # во втором цикле: показать пользователей (check_list из предыдущего цикла),
которые покупали в предыдущий год и ничего не купили в данном году (список польз
ователей из текущего цикла)
    check_list_inactive =  check_list - set(df_1.loc[(slice(None),[i]),:].index.
get_level_values(0))
    if len(check_list_inactive)>0:
        print(f'в {i} году не стали ничего покупать: {len(check_list_inactive)}
 пользователей.')
        pass
    # i-год
    check_list = set(df_1.loc[(slice(None),[i]),:].index.get_level_values(0))
```

```
в 2014 году не стали ничего покупать: 23709 пользователей.
в 2015 году не стали ничего покупать: 130076 пользователей.
в 2016 году не стали ничего покупать: 233546 пользователей.
в 2017 году не стали ничего покупать: 360225 пользователей.
в 2018 году не стали ничего покупать: 541277 пользователей.
CPU times: user 2.92 s, sys: 209 ms, total: 3.13 s
Wall time: 3.13 s
```

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

In [13]:

```
print(f'Максимальное количество покупок совершил пользователь с ID = {df["id_use
r"].value_counts().max()}')
```

```
Максимальное количество покупок совершил пользователь с ID = 6549
```

In [ ]:

```

```