


[skilV/rb_udsiko/blob/main/ipybn/UDS_IKO1.ipynb](#)

```
Ввод [1]: import pandas as pd
import os
```

```
Ввод [2]: import datetime
datetime.datetime.now().strftime('%Y-%m-%d')
```

```
Out[2]: '2023-02-17'
```

```
Ввод [4]: sourcefolder = '../data/'
#filesall =
[entry.name[:15] for entry in os.scandir(sourcefolder) if entry.name[0]!='~']
# [_ for _ in enumerate(filesall)]
```

```
Out[4]: ['iiko_m', 'res', 'UDS']
```

1 defs

```
Ввод [3]: def ABC0(df):
    gdf = df.groupby(by='ID Клиента').aggregate({'Клиент': 'first', 'ID операции': 'count', 'Оплата RUB': 'sum'})\
    .reset_index().sort_values(by='Оплата RUB', ascending=False).rename(columns={'ID операции': 'count'})
    gdf['cumsumОплата RUB'] = gdf['Оплата RUB'].cumsum()
    gdf['percОплата RUB'] = gdf['cumsumОплата RUB']/gdf['Оплата RUB'].sum()
    gdf['ABCOплата RUB'] = gdf['percОплата RUB'].apply(lambda x: 'A1' if x<=0.50 else 'A2' if x<=0.80 else 'B' if x<=0.95 else 'C')
    print(gdf[['ABCOплата RUB']].value_counts())
    # TODO !!!занесем в df категории
    return gdf
```

```
Ввод [4]: def havephone(df):
    print(f"имеем базу {len(df)} строк и {len(df[~df['Телефон'].isna()])} с телефоном\
    это {100*len(df[~df['Телефон'].isna()])//len(df)} процентов")
```

```
Ввод [5]: def havephonebyset(s):
    df = allclients[allclients['ID Клиента'].isin(s)]
    print(f"имеем базу {len(df)} строк и {len(df[~df['Телефон'].isna()])} с телефоном\
    это {100*len(df[~df['Телефон'].isna()])//len(df)} процентов")
    return df
```

```
Ввод [6]: def ABC1(df):
    gdf = df.groupby(by='ID Клиента').aggregate({'Клиент': 'first', 'ID операции': 'count', 'Оплата RUB': 'sum'})\
    .reset_index().sort_values(by='Оплата RUB', ascending=False).rename(columns={'ID операции': 'count'})
    gdf['cumsumОплата RUB'] = gdf['Оплата RUB'].cumsum()
    gdf['percОплата RUB'] = gdf['cumsumОплата RUB']/gdf['Оплата RUB'].sum()
    gdf['ABCOплата RUB'] = gdf['percОплата RUB'].apply(lambda x: 'A' if x<=0.80 else 'B' if x<=0.95 else 'C')
    print(gdf[['ABCOплата RUB']].value_counts())
    return gdf

def svod(df):
    return df.groupby(by='year').aggregate({'ID операции': 'count', 'Оплата RUB': 'sum',
                                             'Оплата баллы/ Оплата баллами': 'sum',
                                             'Оплата сертификатом': 'sum',
                                             'Скидка RUB': 'sum'})
```

```
Ввод [7]: #находим подозрительных во всей выборке для этого группируем по дням
def suspicious(df, n):

    df_by_date = df[['ID Клиента', 'date', 'Клиент', 'ID операции']]\
    .groupby(by=['date', 'ID Клиента']).aggregate({'Клиент': 'first', 'ID операции': 'count'})\
    .reset_index().rename(columns={'ID операции': 'counts'})

    #множество подозрительных и групповых (ходят п и более раз в день)
    return set(df_by_date.loc[df_by_date['counts']>=n, 'ID Клиента'].unique())
```

```
Ввод [8]: ▾ #находим явных читеров, ненормально если 4 и более визитов в во всей выборке для этого группируем по дням
▾ def suspicious (df,n):

    df_by_date = df[['ID Клиента','date','Клиент','ID операции']]\
    .groupby(by=['date','ID Клиента']).aggregate({'Клиент':'first','ID операции':'count'})\
    .reset_index().rename(columns={'ID операции':'counts'})

    #множество подозрительных и групповых (ходят п и более раз в день)
    return set(df_by_date.loc[df_by_date['counts']>=n,'ID Клиента'].unique())
```

Ввод []:

2 Загрузка данных

2.1 ИКО OLAP_uds1 >> iiko

```
Ввод [105]: sourcefolder = '../data/OLAP_uds1/'
```

```
Ввод [106]: [x.name for x in os.scandir(sourcefolder)]
```

```
Out[106]: ['2023-10-01_2023-10-31OLAP_uds1.xlsx',
'experim2023-10-03_2023-10-09OLAP_uds1.xlsx',
'qwerty.xlsx']
```

```
Ввод [107]: ▾ sourcefolder = '../data/OLAP_uds1/'
▾ filesall = [entry.name[:15] for entry in os.scandir(sourcefolder)\
if entry.name[0]!='~']

▾ files = [entry.name for entry in os.scandir(sourcefolder)\
# if entry.name[:14]=='OLAP_T2prog_21'
# or entry.name[:14]=='OLAP_T2prog_22'
# or entry.name[:14]=='OLAP_T2prog_23'\
]
[x for x in enumerate(files)]
```

```
Out[107]: [(0, '2023-10-01_2023-10-31OLAP_uds1.xlsx'),
(1, 'experim2023-10-03_2023-10-09OLAP_uds1.xlsx'),
(2, 'qwerty.xlsx')]
```

```
Ввод [108]: iiko = pd.read_excel(sourcefolder+files[0],header = 4)
```

```
Ввод [109]: iiko.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 355580 entries, 0 to 355579
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Учетный день                          63 non-null    object
1   Отделение                             655 non-null   object
2   Номер кассы                           665 non-null   object
3   Кассир                                 1328 non-null  object
4   Номер чека                            138155 non-null float64
5   Номер смены                           138635 non-null object
6   Номер карты оплаты                    216720 non-null object
7   Блюдо                                 276346 non-null object
8   Процент скидки                       276346 non-null float64
9   Unnamed: 9                            0 non-null     float64
10  Тип скидки                            117730 non-null object
11  Количество блюд                       355580 non-null int64
12  Сумма без скидки, р.                  355580 non-null float64
13  Сумма со скидкой, р.                  355580 non-null float64
dtypes: float64(5), int64(1), object(8)
memory usage: 38.0+ MB
```

```
Ввод [110]: iiko.columns
```

```
Out[110]: Index(['Учетный день', 'Отделение', 'Номер кассы', 'Кассир', 'Номер чека',
'Номер смены', 'Номер карты оплаты', 'Блюдо', 'Процент скидки',
'Unnamed: 9', 'Тип скидки', 'Количество блюд', 'Сумма без скидки, р.',
'Сумма со скидкой, р.'],
dtype='object')
```

```
Ввод [111]: ▾ #Заполняем
▾ list_for_ffill=['Учетный день', 'Отделение', 'Номер кассы', 'Кассир', 'Номер чека',
'Номер смены', 'Номер карты оплаты']

▾ for file in list_for_ffill:
    iiko[file].ffill(inplace=True)
```

Ввод [112]: iiko

Out[112]:

	Учетный день	Отделение	Номер кассы	Кассир	Номер чека	Номер смены	Номер карты оплаты	Блюдо	Процент скидки	Unnamed: 9	Тип скидки	Количество блюд	Сумма без скидки, р.	Сумма со скидкой, р.
0	2022-10-01 00:00:00	203 мкр корпус 22	25	Коркина Полина	35101.0	318	*****8445	Плюшка московская 75 г	0.0000	NaN	Скидка утренняя 30%	1	30.00	30.00
1	2022-10-01 00:00:00	203 мкр корпус 22	25	Коркина Полина	35101.0	318	*****8445	Шанежка провансаль 110 гр	0.3000	NaN	Скидка утренняя 30%	1	75.00	52.50
2	2022-10-01 00:00:00	203 мкр корпус 22	25	Коркина Полина	35101.0	318	*****8445 всего	NaN	NaN	NaN	NaN	2	105.00	82.50
3	2022-10-01 00:00:00	203 мкр корпус 22	25	Коркина Полина	35102.0	318	*****5155	Кекс домашний с изюмом	0.0000	NaN	NaN	2	70.00	70.00
4	2022-10-01 00:00:00	203 мкр корпус 22	25	Коркина Полина	35102.0	318	*****5155	Крендель	0.0000	NaN	NaN	2	92.00	92.00
...
355575	2022-10-31 00:00:00	Юнимопл	11	Валитова Марина Михайловна	122763.0	898	DDDD*****1170	Шанежка с картофелем 110 г	0.2965	NaN	UDS Discount	1	55.00	38.69
355576	2022-10-31 00:00:00	Юнимопл	11	Валитова Марина Михайловна	122763.0	898	DDDD*****1170 всего	NaN	NaN	NaN	NaN	3	226.00	159.00
355577	2022-10-31 00:00:00	Юнимопл	11	Валитова Марина Михайловна всего	122763.0	898	DDDD*****1170 всего	NaN	NaN	NaN	NaN	344	29066.98	26513.58
355578	31.10.2022 всего	Юнимопл	11	Валитова Марина Михайловна всего	122763.0	898	DDDD*****1170 всего	NaN	NaN	NaN	NaN	13735	1114191.28	899240.50
355579	Итого	Юнимопл	11	Валитова Марина Михайловна всего	122763.0	898	DDDD*****1170 всего	NaN	NaN	NaN	NaN	416980	34561171.49	28634291.74

355580 rows × 14 columns

Ввод [114]: #iiko_raw = iiko
#iiko = iiko_raw

Ввод [117]: iiko = iiko_raw[~iiko_raw['Блюдо'].isna()].copy()
#[['Количество блюд', 'Сумма без скидки, р.',
'Сумма со скидкой, р.']].sum()

Ввод [118]: iiko['Учетный день'] = iiko['Учетный день'].astype('datetime64')

Ввод [119]: iiko.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 276346 entries, 0 to 355575
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Учетный день          276346 non-null  datetime64[ns]
1   Отделение             276346 non-null  object
2   Номер кассы           276346 non-null  object
3   Кассир                276346 non-null  object
4   Номер чека            276346 non-null  float64
5   Номер смены           276346 non-null  object
6   Номер карты оплаты    276346 non-null  object
7   Блюдо                 276346 non-null  object
8   Процент скидки       276340 non-null  float64
9   Unnamed: 9            0 non-null       float64
10  Тип скидки            117724 non-null  object
11  Количество блюд       276346 non-null  int64
12  Сумма без скидки, р.  276346 non-null  float64
13  Сумма со скидкой, р.  276346 non-null  float64
dtypes: datetime64[ns](1), float64(5), int64(1), object(7)
memory usage: 31.6+ MB
```

Ввод []: 92148 7610318 6361119.07

Ввод [122]: iiko[['Количество блюд', 'Сумма без скидки, р.',
'Сумма со скидкой, р.']].sum()

Out[122]: Количество блюд 418467.00
Сумма без скидки, р. 34561622.49
Сумма со скидкой, р. 28634671.41
dtype: float64

Ввод [608]: #iiko.to_excel(sourcefolder+'qwerty.xlsx')

Ввод [121]: iiko = iiko[~(iiko['Количество блюд']==0)].copy()

Ввод [123]: iiko

Out[123]:

	Учетный день	Отделение	Номер кассы	Кассир	Номер чека	Номер смены	Номер карты оплаты	Блюдо	Процент скидки	Unnamed: 9	Тип скидки	Количество блюд	Сумма без скидки, р.	Сумма со скидкой, р.
0	2022-10-01	203 мкр корпус 22	25	Коркина Полина	35101.0	318	*****8445	Плюшка московская 75 г	0.0000	NaN	Скидка утренняя 30%	1	30.0	30.00
1	2022-10-01	203 мкр корпус 22	25	Коркина Полина	35101.0	318	*****8445	Шанежка провансаль_110 гр	0.3000	NaN	Скидка утренняя 30%	1	75.0	52.50
3	2022-10-01	203 мкр корпус 22	25	Коркина Полина	35102.0	318	*****5155	Кекс домашний с изюмом	0.0000	NaN	NaN	2	70.0	70.00
4	2022-10-01	203 мкр корпус 22	25	Коркина Полина	35102.0	318	*****5155	Крендель	0.0000	NaN	NaN	2	92.0	92.00
5	2022-10-01	203 мкр корпус 22	25	Коркина Полина	35102.0	318	*****5155	Круассан с шоколадом	0.0000	NaN	NaN	2	198.0	198.00
...
355571	2022-10-31	Юнимолл	11	Валитова Марина Михайловна	122761.0	898	DDDD*****7523	Багет Провансаль с колбасой 150 г	0.1646	NaN	UDS Discount	1	79.0	66.00
355572	2022-10-31	Юнимолл	11	Валитова Марина Михайловна	122762.0	898	DDDD*****1461	Багет Провансаль с колбасой 150 г	0.0000	NaN	NaN	1	79.0	79.00
355573	2022-10-31	Юнимолл	11	Валитова Марина Михайловна	122763.0	898	DDDD*****1170	Багет Провансаль 150 г	0.2964	NaN	UDS Discount	1	75.0	52.77
355574	2022-10-31	Юнимолл	11	Валитова Марина Михайловна	122763.0	898	DDDD*****1170	Закрытая пицца с курицей 120 г	0.2965	NaN	UDS Discount	1	96.0	67.54
355575	2022-10-31	Юнимолл	11	Валитова Марина Михайловна	122763.0	898	DDDD*****1170	Шанежка с картофелем 110 г	0.2965	NaN	UDS Discount	1	55.0	38.69

276290 rows × 14 columns

Ввод []:

2.2 Загрузка справочника клиентов UDS

Ввод [31]: allclients = pd.read_excel('../data/UDS/clients/UDS. Клиенты.xlsx', header =1)

C:\Users\diva0\anaconda3\lib\site-packages\openpyxl\styles\stylesheet.py:226: UserWarning: Workbook contains no default style, apply openpyxl's default
warn("Workbook contains no default style, apply openpyxl's default")

Ввод [32]: allclients.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 104552 entries, 0 to 104551  
Data columns (total 17 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   Регистрация            104552 non-null object  
1   ID Клиента              104552 non-null int64  
2   Клиент                  104552 non-null object  
3   E-mail                  15258 non-null object  
4   Телефон                 44616 non-null float64  
5   Дата рождения          60317 non-null object  
6   По рекомендации участника 91854 non-null object  
7   Источник трафика        12590 non-null object  
8   ID пригласившего        91854 non-null float64  
9   Баллы                   104552 non-null float64  
10  Оплачено (RUB)           104552 non-null float64  
11  Уровень                 104552 non-null float64  
12  Рекомендации            104552 non-null float64  
13  Заметки                  0 non-null float64  
14  Статус                   104552 non-null object  
15  Дата последней покупки  100685 non-null object  
16  Теги                     0 non-null float64  
dtypes: float64(8), int64(1), object(8)  
memory usage: 13.6+ MB
```

Ввод []:

2.3 UDS пекарня, без булочки>> uds

Ввод [9]: folder_uds = '../data/UDS/'

Ввод [10]: files = [entry.name for entry in os.scandir(folder_uds)]
[file for file in enumerate(files)]

Out[10]: [(0, '2022-06-01_2022-11-30UDS_P.xlsx'),
(1, '2022-12-01_2023-01-12UDS_B.xlsx'),
(2, '2022-12-01_2023-01-12UDS_P.xlsx'),
(3, 'clients'),
(4, 'UDS')]

Ввод [11]: udsp1 = pd.read_excel(folder_uds+files[0], header =1)

C:\Users\diva0\anaconda3\lib\site-packages\openpyxl\styles\stylesheet.py:226: UserWarning: Workbook contains no default style, apply openpyxl's default
warn("Workbook contains no default style, apply openpyxl's default")

Ввод [12]: udsp2 = pd.read_excel(folder_uds+files[2], header =1)

Ввод [13]: uds = pd.concat([udsp1,udsp2])

```
uds.info()
```

```

class 'pandas.core.frame.DataFrame'>
Int64Index: 377556 entries, 0 to 65409
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID операции                          377556 non-null float64
1   Data                                  377556 non-null object
2   Номер чека                           373652 non-null float64
3   ID Клиента                           377556 non-null int64
4   Клиент                               377556 non-null object
5   Уровень                              377556 non-null float64
6   Кассир                               377511 non-null object
7   Тип операции                         377556 non-null object
8   Счет RUB                             377556 non-null float64
9   Скидка RUB                           377556 non-null float64
10  Оплата RUB                           377556 non-null float64
11  Оплата Баллы/ Оплата Баллами         377556 non-null float64
12  Оплата сертификатом                  377556 non-null float64
13  Состояние                            377556 non-null object
14  Филиал                               74386 non-null  object

dtypes: float64(8), int64(1), object(6)
memory usage: 46.1+ MB

```

```
uds[['Состояние', 'Тип операции']].value_counts()
```

Состояние	Тип операции	
Выполнена	Стандартная покупка	369793
Возврат	Стандартная покупка	3859
Отменена	Стандартная покупка	3859
Выполнена	Покупка по прайсу	45

dtype: int64

```
uds = uds[uds['Состояние'] != 'Выполнена'].copy()
```

```
uds.rename(columns={'Data':'Data_text_UDS'}, inplace=True)
```

```
uds['datetime'] = uds['Дата_text_UDS'].apply(lambda x: pd.to_datetime(x, format='%d.%m.%Y %H:%M:%S'))
```

```
uds['week'] = uds['datetime'].dt.isocalendar().week
uds['day_name'] = uds['datetime'].dt.day_name()
uds['year'] = uds['datetime'].dt.year
uds['month'] = uds['datetime'].dt.month
uds['day'] = uds['datetime'].dt.day
```

```
#делаем колонку с датой без времени
uds['date'] = pd.to_datetime(uds['datetime']).apply(lambda x: x.date())
```

```
#проверяем
uds[['datetime','date']].sample(10)
```

	datetime	date
294391	2022-06-11 19:15:04	2022-06-11
7274	2023-01-08 09:18:28	2023-01-08
239948	2022-07-20 10:38:45	2022-07-20
22822	2022-11-18 17:08:12	2022-11-18
256159	2022-07-07 16:24:30	2022-07-07
28432	2022-12-22 17:01:52	2022-12-22
274875	2022-06-24 13:40:46	2022-06-24
49284	2022-12-10 12:40:55	2022-12-10
94333	2022-10-13 09:41:41	2022-10-13
65409	2022-12-01 08:01:04	2022-12-01

```
: ▾ #uds.drop(columns='datetime', inplace=True)
```

```
uds.sample(10)
```

71416	529462860.0	24.10.2022 8:59:21	88198.0	1099568304812	Лана Петрова	1.0	Петрова А.Е	Стандартная покупка	233.0	0.0	...	0.0	Выполнена	NaN	2022-10-24 08:59:21	43
132101	513494726.0	24.09.2022 15:13:13	17319.0	1099562367089	Карина Кардашевская	1.0	Маркелова Е.Н	Стандартная покупка	158.0	0.0	...	0.0	Выполнена	NaN	2022-09-24 15:13:13	38
292030	460756706.0	13.06.2022 16:10:29	229635.0	1099559851502	Анжепа Дмитриева	1.0	Вершинина Н.В	Стандартная покупка	390.0	0.0	...	0.0	Выполнена	NaN	2022-06-13 16:10:29	24
46832	536076903.0	05.11.2022 14:16:51	31989.0	1099566104758	Алексей Барабанский	1.0	Егорова Е.П.	Стандартная покупка	198.0	0.0	...	0.0	Выполнена	NaN	2022-11-05 14:16:51	44
82115	526533762.0	18.10.2022 19:22:35	120997.0	1099578179224	+79142736241	1.0	Валитова М.М.	Стандартная покупка	189.7	0.0	...	0.0	Выполнена	NaN	2022-10-18 19:22:35	42

```
#УНИКАЛЬНЫХ КЛИЕНТОВ
len(uds['ID Клиента'].unique())
```

```
Ввод [24]: uds.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 369838 entries, 0 to 65409
Data columns (total 22 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   ID операции                             369838 non-null float64
1   Дата_текст_UDS                          369838 non-null object
2   Номер чека                             369793 non-null float64
3   ID Клиента                             369838 non-null int64
4   Клиент                                 369838 non-null object
5   Уровень                               369838 non-null float64
6   Кассир                                 369793 non-null object
7   Тип операции                          369838 non-null object
8   Счет RUB                              369838 non-null float64
9   Скидка RUB                            369838 non-null float64
10  Оплата RUB                            369838 non-null float64
11  Оплата баллы/ Оплата баллами          369838 non-null float64
12  Оплата сертификатом                  369838 non-null float64
13  Состояние                            369838 non-null object
14  Филиал                               72922 non-null object
15  datetime                             369838 non-null datetime64[ns]
16  week                                 369838 non-null UInt32
17  day_name                             369838 non-null object
18  year                                 369838 non-null int64
19  month                               369838 non-null int64
20  day                                 369838 non-null int64
21  date                                 369838 non-null datetime64[ns]
dtypes: UInt32(1), datetime64[ns](2), float64(8), int64(4), object(7)
memory usage: 63.8+ MB
```

3 Обработка

В первую очередь подчищаем от данных мешающих статистике

```
Ввод [26]: #откат к сырым данным
#uds_row = uds.copy()
#uds = uds_row.copy()
```

3.1 Визитов в день (исключаем клиентов 4 более операции в день)

```
Ввод [27]: uds.columns
```

```
Out[27]: Index(['ID операции', 'Дата_текст_UDS', 'Номер чека', 'ID Клиента', 'Клиент',
              'Уровень', 'Кассир', 'Тип операции', 'Счет RUB', 'Скидка RUB',
              'Оплата RUB', 'Оплата баллы/ Оплата баллами', 'Оплата сертификатом',
              'Состояние', 'Филиал', 'datetime', 'week', 'day_name', 'year', 'month',
              'day', 'date'],
              dtype='object')
```

```
Ввод [28]: #df таблица в которой подсчитана сумма операций в день по клиенту
df = uds[['ID Клиента', 'date', 'Клиент', 'Оплата RUB', 'ID операции']] \
        .groupby(by=['ID Клиента', 'date']) \
        .aggregate({'Клиент': 'first', 'Оплата RUB': 'sum', 'ID операции': 'count'})
```

```
Ввод [29]: #будем считать что 4 более операции в день это ненормально, нарушает
# статистику (первый столбец - количество операций в день по клиенту,
# второй столбец - сколько таких клиентов)
df.value_counts('ID операции')
```

```
Out[29]: ID операции
1      328321
2      17524
3       1381
4        189
5         66
6         18
7         12
8          9
11         7
9          6
10         5
13         4
12         3
26         3
21         3
17         3
22         2
16         2
15         2
20         1
18         1
29         1
30         1
40         1
41         1
45         1
61         1
62         1
63         1
dtype: int64
```

3.1.1 Сохраним и удалим из статистики все операции этих клиентов

Ввод [37]:

```
#смотрим всего 334 случая превышения, но для чистоты анализа удалим все
#транзакции этих клиентов
df[df['ID операции']>=4].reset_index().sort_values('ID операции',ascending=False)
```

Out[37]:

	ID Клиента	date	Клиент	Оплата RUB	ID операции
166	1099568819101	2022-10-13	Дарья Намитова	14199.7	63
171	1099568819101	2022-10-24	Дарья Намитова	12063.4	62
169	1099568819101	2022-10-20	Дарья Намитова	12534.0	61
165	1099568819101	2022-10-12	Дарья Намитова	9091.5	45
167	1099568819101	2022-10-16	Дарья Намитова	11867.4	41
...
79	1099559258552	2022-08-29	Влада Юзенко	708.8	4
78	1099559258552	2022-08-14	Влада Юзенко	970.9	4
77	1099559258552	2022-07-16	Влада Юзенко	1087.0	4
75	1099558862851	2022-10-23	Павел Мархашкинов	441.0	4
0	1099551461830	2022-11-02	Анастасия Торчик	316.0	4

344 rows × 5 columns

Ввод [38]:

```
_.to_excel('4 более операции в день.xlsx')
```

Ввод [34]:

```
#множество клиентов имевших 4 и более чеков в день
clients4moredayset = set(df[df['ID операции']>=4].sort_index()\
                           .reset_index()['ID Клиента'].unique())
len(clients4moredayset)
```

Out[34]: 169

Ввод [35]:

```
#Это телефоны клиентов 4 более операции в день
havephonebyset(clients4moredayset)[['Клиент','E-mail','Телефон','Дата рождения']]
```

имеем базу 168 строк и 141 с телефоном это 83.92857142857143 процентов

Out[35]:

	Клиент	E-mail	Телефон	Дата рождения
4576	Марина Кольшкина	NaN	7.924664e+10	27.03.1969
5625	Николай Кочнев	NaN	7.999175e+10	14.10.1992
7202	Елена Баишева	NaN	7.914288e+10	21.12.1967
7711	Артур Оконешников	NaN	NaN	13.02.2004
10065	Соёлма Молонова	NaN	7.924770e+10	13.07.1999
...
103518	Михаил Платонов	NaN	7.965767e+10	16.05.2000
103551	Алёна Слепцова	NaN	7.924174e+10	08.06.1999
103981	Наталья Ефремова	NaN	7.999173e+10	25.08.1997
104089	Марина Артахинова	NaN	7.996315e+10	28.04.1999
104244	Анастасия Торчик	NaN	7.914294e+10	19.02.1987

168 rows × 4 columns

Ввод [36]:

```
_.to_excel('4 более операции в день это ненормально_все_дни.xlsx')
```

Ввод [39]:

```
#Вот операции этих клиентов за весь период
uds[uds['ID Клиента'].isin(clients4moredayset)]
```

Out[39]:

	ID операции	Дата_text_UDS	Номер чека	ID Клиента	Клиент	Уровень	Кассир	Тип операции	Счет RUB	Скидка RUB	...	Оплата сертификатом	Состояние	Филиал	datetime	week	day_nap
0	550695451.0	30.11.2022 22:23:04	5344.0	1099565969450	Наталья Вершинина	1.0	Соколов Валерий	Стандартная покупка	216.0	0.0	...	0.0	Выполнена	NaN	2022-11-30 22:23:04	48	Wednesd
7	550672201.0	30.11.2022 21:36:06	233933.0	1099559989844	Вероника Кузьмина	1.0	Кузьмина В.А	Стандартная покупка	411.0	0.0	...	0.0	Выполнена	NaN	2022-11-30 21:36:06	48	Wednesd
17	550654065.0	30.11.2022 20:58:58	8128.0	1099577437494	нюргуяна романова	1.0	Романова Н.В	Стандартная покупка	344.0	0.0	...	0.0	Выполнена	NaN	2022-11-30 20:58:58	48	Wednesd
18	550652724.0	30.11.2022 20:55:47	8127.0	1099577437494	нюргуяна романова	1.0	Романова Н.В	Стандартная покупка	103.2	0.0	...	0.0	Выполнена	NaN	2022-11-30 20:55:47	48	Wednesd
35	550642184.0	30.11.2022 20:31:30	45512.0	1099574317947	Анита Durova	1.0	Дурова А	Стандартная покупка	207.0	0.0	...	0.0	Выполнена	NaN	2022-11-30 20:31:30	48	Wednesd
...
65066	550891941.0	01.12.2022 10:31:50	74493.0	1099555647222	Сахаайа Никанорова	1.0	Молонова С.Н	Стандартная покупка	228.0	0.0	...	0.0	Выполнена	NaN	2022-12-01 10:31:50	48	Thursd
65223	550888628.0	01.12.2022 9:32:17	36240.0	1099551926183	Наталья Ефремова	1.0	Уарова.Я.М	Стандартная покупка	75.0	0.0	...	0.0	Выполнена	NaN	2022-12-01 09:32:17	48	Thursd
65251	550888207.0	01.12.2022 9:19:03	233953.0	1099562446655	Миша Софронов	1.0	Кузьмина В.А	Стандартная покупка	372.0	0.0	...	0.0	Выполнена	NaN	2022-12-01 09:19:03	48	Thursd
65259	550888151.0	01.12.2022 9:16:16	8133.0	1099564406955	Ариана Николаева	1.0	Христофорова А.	Стандартная покупка	140.0	0.0	...	0.0	Выполнена	NaN	2022-12-01 09:16:16	48	Thursd
65368	550886353.0	01.12.2022 8:31:02	2655.0	1099555772968	Нина Барашкова	1.0	Вершинина.Н.В	Стандартная покупка	96.0	0.0	...	0.0	Выполнена	NaN	2022-12-01 08:31:02	48	Thursd

8898 rows × 22 columns

```
Ввод [40]: #Сумма '4 более операции в день это ненормально_все дни.xlsx'
#8563 операций 1626300.68 оплата 141947.32 оплата баллами 1932.0

svod(_)
```

Out[40]:

	ID операции	Оплата RUB	Оплата баллы/ Оплата баллами	Оплата сертификатом	Скидка RUB
year					
2022	8660	1634947.78	142895.22	1932.0	0.0
2023	238	54223.34	5993.36	0.0	0.0

```
Ввод [40]: #uds_row = uds.copy()
```

```
Ввод [41]: len(uds_row)
```

Out[41]: 369838

```
Ввод [42]: #!!!! Удаляем 4 и более операций в день
uds = uds[~uds['ID Клиента'].isin(clients4moredayset)].copy()
```

3.2 Анализ всех покупок по среднему чеку ('cut_avgcheck')

Вводим распределение всех продаж по среднему чеку uds['cut_avgcheck']

```
Ввод [43]: uds.columns
```

Out[43]: Index(['ID операции', 'Дата_text_UDS', 'Номер чека', 'ID Клиента', 'Клиент', 'Уровень', 'Кассир', 'Тип операции', 'Счет RUB', 'Скидка RUB', 'Оплата RUB', 'Оплата баллы/ Оплата баллами', 'Оплата сертификатом', 'Состояние', 'Филиал', 'datetime', 'week', 'day_name', 'year', 'month', 'day', 'date'], dtype='object')

```
Ввод [45]: uds[['ID операции', 'Дата_text_UDS', 'Номер чека', 'ID Клиента', 'Клиент', 'Оплата RUB']] \
.sort_values('Оплата RUB', ascending=False)
```

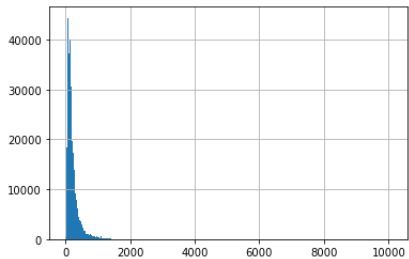
Out[45]:

	ID операции	Дата_text_UDS	Номер чека	ID Клиента	Клиент	Оплата RUB
16498	567834290.0	29.12.2022 11:20:13	12685.0	1099556797360	Александр Конторусов	10077.00
265269	470242434.0	01.07.2022 9:09:10	26268.0	1099559455669	Ekaterina Sivtceva	8262.00
28876	563289697.0	22.12.2022 13:44:22	8793.0	1099581895826	+79141106070	6418.00
247195	476812000.0	14.07.2022 12:17:15	244539.0	1099573684686	ИННА Аммосова	6090.00
29989	562855675.0	21.12.2022 19:14:12	97510.0	1099580953139	Юлия Трофимова	5700.00
...
18435	545065039.0	21.11.2022 13:13:04	36304.0	1099579223067	+79148279001	5.00
65172	550889126.0	01.12.2022 9:45:50	38888.0	1099573826333	Ольга Степанищева	5.00
60086	552356229.0	03.12.2022 19:52:21	39540.0	1099555229357	Святослав Сляднев	5.00
36964	539417620.0	11.11.2022 10:45:56	92876.0	1099580648174	Полина Маккормик	5.00
3843	574874075.0	10.01.2023 16:27:35	7872.0	1099557270787	Наталья Габышева	3.96

360940 rows x 6 columns

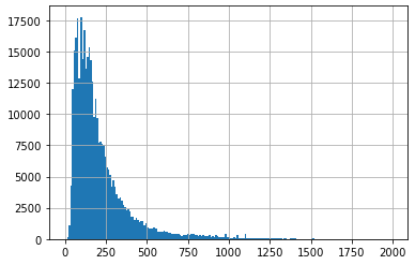
```
Ввод [46]: uds['Оплата RUB'].hist(bins = 400)
```

Out[46]: <AxesSubplot:>



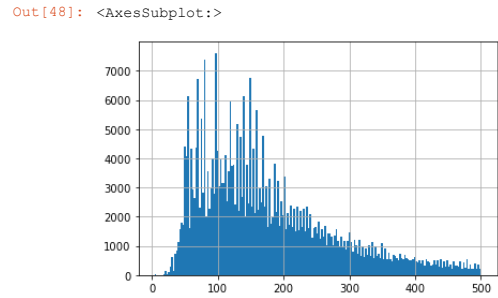
```
Ввод [47]: uds.loc[uds['Оплата RUB']<2000, 'Оплата RUB'].hist(bins = 200)
```

Out[47]: <AxesSubplot:>



Ввод [48]:

uds.loc[uds['Оплата RUB']<500,'Оплата RUB'].hist(bins=200)



Ввод [49]:

cut_bins = [0, 100, 200, 750, 1500, 100000]
cut_labels = ['0-100', '100-200', '200-750', '750-1500', '>1500']
uds['cut_avgcheck'] = pd.cut(uds['Оплата RUB'],
bins=cut_bins,
labels=cut_labels)

Ввод [50]:

#Распределение всех продаж по среднему чеку
uds['cut_avgcheck'].value_counts()

Out[50]:

100-200	133790
200-750	122950
0-100	93751
750-1500	9697
>1500	752

Name: cut_avgcheck, dtype: int64

3.2.1 Список часто покупающих на сумму более 3000 руб (Андрей Крылов, Роза Шейхрамова ...)

Ввод [54]:

#список купивших на сумму более 3000 руб
uds[uds['Оплата RUB']>3000][['ID операции', 'Дата_text_UDS', 'Номер чека', 'ID Клиента', 'Клиент',
'Скидка RUB', 'Оплата RUB',]]

Out[54]:

	ID операции	Дата_text_UDS	Номер чека	ID Клиента	Клиент	Скидка RUB	Оплата RUB
33487	540661657.0	13.11.2022 12:19:27	665.0	1099574768293	Андрей Крылов	0.0	3989.0
44392	536945086.0	07.11.2022 10:32:05	65851.0	1099579311427	+79142297756	0.0	3353.0
78656	527319813.0	20.10.2022 13:37:32	89913.0	1099559992438	+79142701017	0.0	3061.0
111068	519405238.0	05.10.2022 9:41:04	144082.0	1099574768293	Андрей Крылов	0.0	3720.0
123931	515621492.0	28.09.2022 16:02:27	86631.0	1099555886332	Евгения Румянцева	0.0	3194.6
127482	514948244.0	27.09.2022 9:34:45	65903.0	1099566887839	Роза Шейхрамова	0.0	5672.5
127509	514947965.0	27.09.2022 9:26:30	65903.0	1099566887839	Роза Шейхрамова	0.0	5672.5
145403	510290868.0	18.09.2022 16:16:48	85060.0	1099576615105	Сергей Горбунов	0.0	3294.0
161792	505667488.0	10.09.2022 10:28:45	18146.0	1099574768293	Андрей Крылов	0.0	3279.0
173423	502823455.0	04.09.2022 14:06:23	16648.0	1099574768293	Андрей Крылов	0.0	3077.0
197788	495470531.0	21.08.2022 13:38:10	6719.0	1099563692814	Сергей Иванов	0.0	4044.0
198275	495439639.0	21.08.2022 9:06:07	6589.0	1099574768293	Андрей Крылов	0.0	3939.0
199508	494943160.0	20.08.2022 9:01:10	13087.0	1099574768293	Андрей Крылов	0.0	3306.0
206584	492605245.0	15.08.2022 14:17:22	194145.0	1099555940874	+79627368694	0.0	3151.0
220803	487493606.0	04.08.2022 16:09:10	190899.0	109954909030	кустана готовцева	0.0	3025.0
247195	476812000.0	14.07.2022 12:17:15	244539.0	1099573684686	ИННА Аммосова	0.0	6090.0
265269	470242434.0	01.07.2022 9:09:10	26268.0	1099559455669	Ekaterina Sivtceva	0.0	8262.0
272642	468021284.0	26.06.2022 13:09:06	74003.0	1099569433998	Valeria Sergeeva	0.0	3766.5
15359	568189702.0	29.12.2022 19:09:05	143731.0	1099574768293	Андрей Крылов	0.0	3001.0
16498	567834290.0	29.12.2022 11:20:13	12685.0	1099556797360	Александр Конторусов	0.0	10077.0
16824	567812571.0	29.12.2022 8:47:36	242023.0	1099560432382	крестина егорова	0.0	3890.0
18280	567114750.0	28.12.2022 11:52:29	12269.0	1099582592008	Ирина Смирникова	0.0	3160.0
18432	567042635.0	28.12.2022 10:48:19	241776.0	1099582183248	+79245635500	0.0	4689.0
20475	566408348.0	27.12.2022 9:44:21	98122.0	1099566674228	Дмитрий Чурин	0.0	5000.0
22286	565752407.0	26.12.2022 10:50:55	193384.0	1099568576972	Сардаана Егорова	0.0	3305.0
23913	565177070.0	25.12.2022 11:57:06	44825.0	1099557298176	Мария Мансурова	0.0	4370.7
24841	564584572.0	24.12.2022 15:55:13	142895.0	1099574768293	Андрей Крылов	0.0	3692.8
24975	564550127.0	24.12.2022 14:57:10	10198.0	1099581988452	+79841079101	0.0	4584.0
28876	563289697.0	22.12.2022 13:44:22	8793.0	1099581895826	+79141106070	0.0	6418.0
29989	562855675.0	21.12.2022 19:14:12	97510.0	1099580953139	Юлия Трофимова	0.0	5700.0
56795	553506945.0	06.12.2022 9:15:28	46025.0	1099552234489	Карина Протасова	0.0	3735.0

Ввод []:

3.3 Общая информация 2022 за 7 месяцев

```
Ввод [51]: # исследуем 2022 год
uds = uds[uds['year']==2022]
uds.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 348915 entries, 1 to 65409
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID операции                          348915 non-null float64
1   Data_text_UDS                       348915 non-null object
2   Номер чека                          348870 non-null float64
3   ID Клиента                          348915 non-null int64
4   Клиент                             348915 non-null object
5   Уровень                            348915 non-null float64
6   Кассир                             348870 non-null object
7   Тип операции                       348915 non-null object
8   Счет RUB                           348915 non-null float64
9   Скидка RUB                         348915 non-null float64
10  Оплата RUB                         348915 non-null float64
11  Оплата баллы/ Оплата баллами       348915 non-null float64
12  Оплата сертификатом                348915 non-null float64
13  Состояние                          348915 non-null object
14  Филиал                             69580 non-null object
15  datetime                           348915 non-null datetime64[ns]
16  week                               348915 non-null UInt32
17  day_name                           348915 non-null object
18  year                               348915 non-null int64
19  month                              348915 non-null int64
20  day                                348915 non-null int64
21  date                               348915 non-null datetime64[ns]
22  cut_avgcheck                       348915 non-null category
dtypes: UInt32(1), category(1), datetime64[ns](2), float64(8), int64(4), object(7)
memory usage: 60.6+ MB

Ввод [55]: print (f" Всего за период с {uds['datetime'].min().strftime('%d.%m.%Y')} по {uds['datetime'].max().strftime('%d.%m.%Y')} \
{len(uds)} чеков по UDS, количество уникальных клиентов {len(uds['ID Клиента'].unique())}")

Всего за период с 01.06.2022 по 31.12.2022 348915 чеков по UDS, количество уникальных клиентов 64973
```

3.4 Сводная таблица по суммам оплаты, оплаты баллами и сертификатами

```
Ввод [56]: #Смотрим общую сумму за период данных
uds.groupby(by='year').aggregate({'ID операции': 'count', 'Оплата RUB': 'sum',
                                  'Оплата баллы/ Оплата баллами': 'sum',
                                  'Оплата сертификатом': 'sum',
                                  'Скидка RUB': 'sum'})

Out[56]:
   ID операции  Оплата RUB  Оплата баллы/ Оплата баллами  Оплата сертификатом  Скидка RUB
year
2022         348915  75302981.38                6962869.6                83085.7             0.0

Ввод [57]: #По месяцам с суммами
uds.groupby_money = uds.groupby(by='month').aggregate({'ID операции': 'count', 'Оплата RUB': 'sum',
                                                        'Оплата баллы/ Оплата баллами': 'sum',
                                                        'Оплата сертификатом': 'sum',
                                                        'Скидка RUB': 'sum'})

uds.groupby_money

Out[57]:
   ID операции  Оплата RUB  Оплата баллы/ Оплата баллами  Оплата сертификатом  Скидка RUB
month
6             44908   9106490.30                742731.30                11850.3             0.0
7             37922   7998208.19                624876.91                 195.0             0.0
8             43323   9532985.36                831063.54                1513.0             0.0
9             58527  12059103.51                1211289.89                54814.6             0.0
10            60763  12855261.63                1333569.27                1219.8             0.0
11            52761  11462296.68                1082614.42                1019.9             0.0
12            50711  12288635.71                1136724.27                12473.1             0.0
```

4 Анализ пробной группы клиентов - супер клиентов которые все 7 месяцев прошлого года каждый месяц приходят хоть 1 раз (время жизни клиента более 6 месяцев)

Ввод [59]:

```
#Группировка по клиентам и месяцам - посмотрим сколько раз в месяц ходят клиенты
uds_by_clnt = uds.groupby(by=['month','ID Клиента'])\
    .aggregate({'Клиент':'first','Оплата RUB':'sum','Оплата баллы/ Оплата баллами':'sum','Оплата сертификатом':'sum','ID операции':'count'})\
    .sort_values(by='Оплата RUB', ascending=False)\
    .rename(columns={'ID операции':'чеков'})
uds_by_clnt
```

Out[59]:

	month	ID Клиента	Клиент	Оплата RUB	Оплата баллы/ Оплата баллами	Оплата сертификатом	чеков
	72998	9 1099574768293	Андрей Крылов	20360.5	2677.0	0.0	11
	96750	10 1099574768293	Андрей Крылов	16833.0	0.0	0.0	7
	52646	8 1099574768293	Андрей Крылов	16261.0	0.0	0.0	6
	61048	9 1099556887839	Роза Шейхрамова	15820.0	434.0	29.0	19
	128307	12 1099555471162	Валентина Петрова	15460.6	1251.0	0.0	16

	146740	12 1099581587527	+79246685581	10.0	0.0	0.0	1
	99213	10 1099577195042	Семен Кириллин	5.0	0.0	0.0	1
	26925	7 1099559929593	Оксана Фиськова	5.0	0.0	0.0	1
	103920	10 1099579017084	+79142616170	5.0	0.0	0.0	1
	113584	11 1099561190152	+79963177987	5.0	0.0	0.0	1

148630 rows × 7 columns

Ввод [60]:

```
# Множества клиентов по месяцам
clntset = {}
for m in range(6,13):
    clntset[m] = set(uds_by_clnt.loc[uds_by_clnt['month']==m, 'ID Клиента'])
```

Ввод [61]:

```
# имеем суперклиентов
superclnset = set.intersection(clntset[6],clntset[7],clntset[8],clntset[9],clntset[10],clntset[11],clntset[12])
len(superclnset)
```

Out[61]: 1989

Ввод [62]:

```
uds_superclnts = uds[uds['ID Клиента'].isin(superclnset)]
uds_superclnts['Клиент'].value_counts()
```

Out[62]:

Татьяна Анисимова	207
Клара Каназарова	164
иван Бухаров	162
Наталья Кухарева	152
Екатерина Иванова	145
...	
Валентина Старцева	8
Мария Галаева	8
Елисей Лукьянов	8
Анна Дьячковская	7
Оля П	2

Name: Клиент, Length: 1956, dtype: int64

4.1 Общая информация по группе суперклиентов

Ввод [63]:

```
#В поисках идеального клиента сузили количество клиентов до суперклиентов (т.е. Клиенты - время жизни которых более 6 мес)
#период ABC анализа 7 мес с 01
```

Ввод [64]:

```
#df - база только суперклиентов
df = uds[uds['ID Клиента'].isin(superclnset)]
```

Ввод [65]:

```
print (f" Всего за период с {df['datetime'].min().strftime('%d.%m.%Y')} по {df['datetime'].max().strftime('%d.%m.%Y')} \
{len(df)} чеков по UDS, количество уникальных клиентов {len(df['ID Клиента'].unique())}")
```

Всего за период с 01.06.2022 по 31.12.2022 66390 чеков по UDS, количество уникальных клиентов 1989

4.1.1 Количество дней в исследуемом периде diffdays

Ввод [66]:

```
#Количество дней в исследуемом периде
diffdays = (df['datetime'].max()-df['datetime'].min()).days
diffdays
```

Out[66]: 213

Ввод [67]:

```
df.groupby(by='year').aggregate({'ID операции':'count','Оплата RUB':'sum',
                                'Оплата баллы/ Оплата баллами':'sum',
                                'Оплата сертификатом':'sum',
                                'Скидка RUB':'sum'})
```

Out[67]:

	ID операции	Оплата RUB	Оплата баллы/ Оплата баллами	Оплата сертификатом	Скидка RUB
year					
2022	66390	13280629.07	1475162.03	14968.4	0.0

Ввод [68]:

```
#Данные всей выборки по сумме
uds.groupby(by='year').aggregate({'ID операции':'count','Оплата RUB':'sum',
                                'Оплата баллы/ Оплата баллами':'sum',
                                'Оплата сертификатом':'sum',
                                'Скидка RUB':'sum'})
```

Out[68]:

	ID операции	Оплата RUB	Оплата баллы/ Оплата баллами	Оплата сертификатом	Скидка RUB
year					
2022	348915	75302981.38	6962869.6	83085.7	0.0

```
Ввод [69]: #Средний чек по этой группе немного ниже
13280629.07/66390, 75302981.38/348915

Out[69]: (200.03960039162524, 215.8204186693034)

Ввод [70]: #Процент группы по выручке хороший для анализа
100*13280629.07/75302981.38

Out[70]: 17.636259317519205

Ввод [71]: #при этом по количеству клиентов
len(uds['ID Клиента'].unique()), len(df['ID Клиента'].unique())

Out[71]: (64973, 1989)

Ввод [72]: len(df['ID Клиента'].unique())/len(uds['ID Клиента'].unique())

Out[72]: 0.03061271605128284

!!! 3% по численности суперклиентов клиентов дает 17% выручки
```

4.2 Произведем ABC анализ этой группы

```
Ввод [73]: dfABC = ABC0(df)

ABCОплата RUB
A2          632
B           528
A1          513
C           316
dtype: int64

Ввод [74]: dfABC

Out[74]:
   ID Клиента  Клиент  count  Оплата RUB  cumsumОплата RUB  percОплата RUB  ABCОплата RUB
492  1099555739354  Георгий Птицын    137    38048.70         38048.70         0.002865         A1
292  1099555215524  Екатерина Иванова    114    37800.60         75849.30         0.005711         A1
1602 1099562994900  Надежда Макарова     63    34941.70        110791.00         0.008342         A1
756  1099556397923  Виталий Журавский    103    34647.90        145438.90         0.010951         A1
1425 1099560013378      Макс Сем       101    33421.13        178860.03         0.013468         A1
...      ...      ...      ...      ...      ...      ...      ...
1476 1099560955823      +79248736615     9      920.00        13277376.87         0.999755         C
998  1099557336937  Елена Черемкина     17     918.50        13278295.37         0.999824         C
1136 1099557923599  Анна Андреева      8     846.60        13279141.97         0.999888         C
709  1099556266549  Наталия Каплина      8     831.10        13279973.07         0.999951         C
1381 1099559716520      Яна Олесова     10     656.00        13280629.07         1.000000         C

1989 rows x 7 columns

Ввод [75]: dfABC['средний_чек213'] = dfABC['Оплата RUB']/dfABC['count']

Ввод [76]: dfABC.groupby(by='ABCОплата RUB').aggregate({'ID Клиента':'count','count':'sum',
'Оплата RUB':'sum','средний_чек213':'mean'}).reset_index().sort_values(by='ABCОплата RUB')

Out[76]:
   ABCОплата RUB  ID Клиента  count  Оплата RUB  средний_чек213
0              A1          513    28384    6640163.33      280.988003
1              A2          632    20525    3982693.41      226.321182
2              B          528    12189    1990988.93      182.215666
3              C          316    5292     666783.40      136.237514

Ввод [ ]: # TODO !!!занесем в df категории
```

4.3 Разбор категорий А В С по общему количеству посещений

```
Ввод [77]: #Гистограмма показывает x - количество всего посещений за период 213 дней y - сколько клиентов

print(diffdays, dfABC.loc[dfABC['ABCОплата RUB']=='A1','count'].median(),
diffdays/dfABC.loc[dfABC['ABCОплата RUB']=='A1','count'].median())
dfABC.loc[dfABC['ABCОплата RUB']=='A1','count'].hist(bins=300)

213 51.0 4.176470588235294

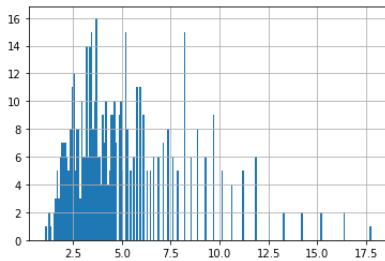
Out[77]: <AxesSubplot:>
```



Ввод [81]:

```
▼ #Гистограмма показывает x - раз во сколько дней ходит клиент у - сколько клиентов
▼ dfABC.loc[dfABC['ABCOплата RUB']=='A1','count']\
  .apply(lambda x: diffdays/x).hist(bins=200)
▼ print(dfABC.loc[dfABC['ABCOплата RUB']=='A1','count']\
  .apply(lambda x: diffdays/x).median(),
  dfABC.loc[dfABC['ABCOплата RUB']=='A1','count']\
  .apply(lambda x: diffdays/x).mean()
  )
```

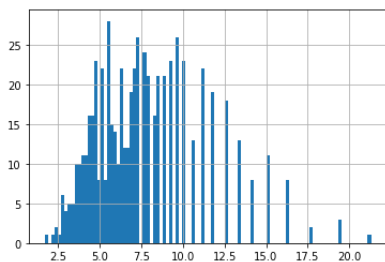
4.176470588235294 4.987928093568766



Ввод [557]:

```
▼ #Гистограмма показывает x - раз во сколько дней ходит клиент у - сколько клиентов
▼ dfABC.loc[dfABC['ABCOплата RUB']=='A2','count']\
  .apply(lambda x: diffdays/x).hist(bins=100)
▼ print(dfABC.loc[dfABC['ABCOплата RUB']=='A2','count']\
  .apply(lambda x: diffdays/x).median(),
  dfABC.loc[dfABC['ABCOплата RUB']=='A2','count']\
  .apply(lambda x: diffdays/x).mean()
  )
```

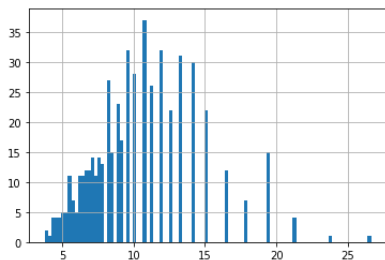
7.344827586206897 7.776155905933606



Ввод [82]:

```
▼ #Гистограмма показывает x - раз во сколько дней ходит клиент у - сколько клиентов
▼ dfABC.loc[dfABC['ABCOплата RUB']=='B','count']\
  .apply(lambda x: diffdays/x).hist(bins=100)
▼ print(dfABC.loc[dfABC['ABCOплата RUB']=='B','count']\
  .apply(lambda x: diffdays/x).median(),
  dfABC.loc[dfABC['ABCOплата RUB']=='B','count']\
  .apply(lambda x: diffdays/x).mean()
  )
```

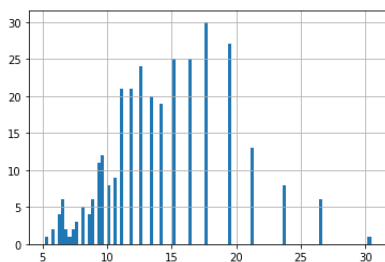
10.142857142857142 10.424743129234441



Ввод [83]:

```
▼ #Гистограмма показывает x - раз во сколько дней ходит клиент у - сколько клиентов
▼ dfABC.loc[dfABC['ABCOплата RUB']=='C','count']\
  .apply(lambda x: diffdays/x).hist(bins=100)
▼ print(dfABC.loc[dfABC['ABCOплата RUB']=='C','count']\
  .apply(lambda x: diffdays/x).median(),
  dfABC.loc[dfABC['ABCOплата RUB']=='C','count']\
  .apply(lambda x: diffdays/x).mean()
  )
```

13.3125 14.19709799257788



4.4 Изучение динамики ABC суперклиентов по месяцам

4.4.1 superclnset - множество супер клиентов, uds_superclnts - продажи суперклиентов по чекам

```
Ввод [84]: uds_superclnts = uds[uds['ID Клиента'].isin(superclnset)]

Ввод [85]: uds_superclnts.columns

Out[85]: Index(['ID операции', 'Дата_text_UDS', 'Номер чека', 'ID Клиента', 'Клиент',
              'Уровень', 'Кассир', 'Тип операции', 'Счет RUB', 'Скидка RUB',
              'Оплата RUB', 'Оплата баллы/ Оплата баллами', 'Оплата сертификатом',
              'Состояние', 'Филиал', 'datetime', 'week', 'day_name', 'year', 'month',
              'day', 'date', 'cut_avgcheck'],
              dtype='object')
```

4.4.2 Присоединить к продажам метки ABC

```
Ввод [87]: uds_superclnts = pd.merge(left=uds_superclnts, right=dfABC[['ID Клиента', 'ABCOплата RUB']],
                                     how='left', left_on='ID Клиента', right_on='ID Клиента')
```

```
Ввод [88]: # Подготовим для среднего чека, они пока равны, тк uds_superclnts по 1 чеку
uds_superclnts['Ср чек RUB'] = uds_superclnts['Оплата RUB']
```

```
Ввод [89]: uds_superclnts[uds_superclnts['ABCOплата RUB']=='A1'].groupby(by='month')\
            .aggregate({'ID операции': 'count', 'Оплата RUB': 'sum',
                        'Оплата баллы/ Оплата баллами': 'sum', 'Ср чек RUB': 'mean'})
# A1
```

```
Out[89]:
```

	ID операции	Оплата RUB	Оплата баллы/ Оплата баллами	Ср чек RUB
month				
6	3889	888035.79	84764.41	228.345536
7	3865	852593.17	78578.13	220.593317
8	4019	919520.88	95346.52	228.793451
9	4454	1004857.02	116381.98	225.607773
10	4445	1053065.70	123479.20	236.910169
11	4015	971397.35	101398.55	241.942055
12	3697	950693.42	96212.28	257.152670

```
Ввод [90]: uds_superclnts[uds_superclnts['ABCOплата RUB']=='A2'].groupby(by='month')\
            .aggregate({'ID операции': 'count', 'Оплата RUB': 'sum',
                        'Оплата баллы/ Оплата баллами': 'sum', 'Ср чек RUB': 'mean'})
# A2
```

```
Out[90]:
```

	ID операции	Оплата RUB	Оплата баллы/ Оплата баллами	Ср чек RUB
month				
6	3111	571518.45	57023.15	183.708920
7	2648	491322.82	49587.78	185.544872
8	2810	524357.49	57066.81	186.604089
9	3241	621167.55	77317.15	191.659226
10	3182	621583.67	79011.03	195.343705
11	2869	578080.82	64547.58	201.492095
12	2664	574662.61	62443.99	215.714193

```
Ввод [91]: uds_superclnts[uds_superclnts['ABCOплата RUB']=='B'].groupby(by='month')\
            .aggregate({'ID операции': 'count', 'Оплата RUB': 'sum',
                        'Оплата баллы/ Оплата баллами': 'sum', 'Ср чек RUB': 'mean'})
# B
```

```
Out[91]:
```

	ID операции	Оплата RUB	Оплата баллы/ Оплата баллами	Ср чек RUB
month				
6	1784	270736.21	28406.79	151.757965
7	1632	248332.28	24843.62	152.164387
8	1707	282472.51	34706.79	165.478916
9	1935	300593.64	41875.46	155.345550
10	1895	309983.19	42694.21	163.579520
11	1647	284057.16	35674.14	172.469435
12	1589	294813.94	34226.56	185.534261

5 Контакты, возраст и прочая информация по группе

5.0.1 Сколько имеем номеров телефонов

```
Ввод [92]: havephone(allclients[allclients['ID Клиента'].isin(superclnset)])

имеем базу 1989 строк и 1733 с телефоном     это 87.12921065862243 процентов
```

```
Ввод [93]: df_contacts = havephonebyset(superclnset).copy()

имеем базу 1989 строк и 1733 с телефоном     это 87.12921065862243 процентов
```

5.0.2 Возраст группы

Ввод [94]:

df_contacts[['Дата рождения', 'Клиент']]

Out[94]:

	Дата рождения	Клиент
4956	01.12.1983	Мария Андреева
5984	03.05.1959	Галина Николаевна
9241	15.01.2007	Полина Макормик
10566	09.11.1975	Анна Кудрявцева
10682	10.08.1964	Мадасов Анатолий
...
104502	21.12.1989	Zoya Mixajlova
104537	10.06.1982	Наталья Старостина
104539	18.10.1988	Алина Павлова
104547	14.10.1999	Айталипа Семенова
104548	25.04.1988	Мария Михайлова

1989 rows × 2 columns

Ввод [95]:

df_contacts.loc[df_contacts['Дата рождения'].isna(),['Дата рождения', 'Клиент']]

Out[95]:

	Дата рождения	Клиент
14116	NaN	Василий Кондаков
35211	NaN	+79834687207
35912	NaN	Соколова Марина
36698	NaN	Сахая Куличина
37614	NaN	Галина Иванова
...
102745	NaN	Виктория Трофимова
103315	NaN	Варвара Кондакова
103959	NaN	Надежда К
104152	NaN	Сардана Устинова
104246	NaN	Аина Никитина

232 rows × 2 columns

Ввод [96]:

df_contacts['age'] = (datetime.datetime.now() - pd.to_datetime(allclients['Дата рождения'])).astype('<m8[Y]')

Ввод [98]:

df_contacts['age'].hist(bins=70, legend=True)



Ввод [99]:

age_bins = [0, 16, 25, 40, 50, 100]
age_labels = ['0-16', '16-25', '25-40', '40-50', '>50']
df_contacts['cutage'] = pd.cut(df_contacts['age'],
 bins=age_bins,
 labels=age_labels)

Ввод [100]:

#TODO посмотреть по категориям

df_contacts = pd.merge(left=df_contacts, right=dfABC[['ID Клиента', 'ABCOплата RUB']],
 how='left',
 left_on='ID Клиента',right_on='ID Клиента')

Ввод [101]:

	10:55:33	кудрявцева	Николаевна
4	23.11.2022 15:25:45	1099580199262	Мадасов Анатолий	NaN	7.924370e+10	10.08.1964	Дуйшеев Бекболсун	UDS App	2.748785e+11	352.55	16095.50	1.0	0.0	NaN	
...	
1984	26.04.2021 12:37:53	1099551037906	Zoya Mixajlova	NaN	7.968157e+10	21.12.1989	NaN	NaN	NaN	144.63	12039.31	1.0	0.0	NaN	
1985	26.04.2021 10:16:57	1099551036724	Наталья Старостина	NaN	7.914112e+10	10.06.1982	Мария Пак	NaN	1.099551e+12	115.55	5926.30	3.0	2.0	NaN	
1986	26.04.2021 10:16:28	1099551036719	Алина Павлова	a1ina@bk.ru	7.914293e+10	18.10.1988	Мария Пак	NaN	1.099551e+12	81.13	7756.86	3.0	2.0	NaN	
1987	24.04.2021 11:36:17	1099550970256	Айталипа Семенова	NaN	7.914291e+10	14.10.1999	NaN	NaN	NaN	194.79	9042.70	1.0	1.0	NaN	
1988	24.04.2021 8:06:38	1099550969406	Мария Михайлова	NaN	7.984110e+10	25.04.1988	NaN	NaN	NaN	265.60	24326.00	1.0	0.0	NaN	

1989 rows × 15 columns

Ввод [137]:

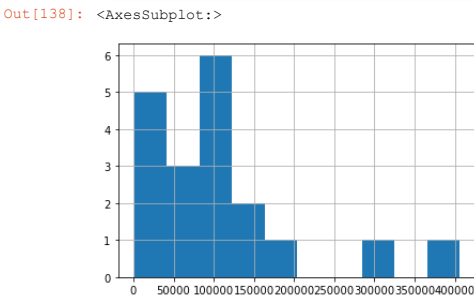
merged.groupby(by='Отделение').sum()[['Оплата RUB', 'Сумма со скидкой, р.']]

Out[137]:

Отделение	Оплата RUB	Сумма со скидкой, р.
50 лет октября 6	93263.55	93608.47
Авиационная 3	22766.90	28369.04
Алексеева 7	95695.81	95695.70
ДОСТАВКА Кузьмина 26/4	1026.00	3220.95
Дзержинского 1	95660.17	95659.93
Дзержинского 30а	147960.79	158500.49
Дзержинского 42 к.3	36915.96	37217.09
Кирова 13	69064.36	71054.09
Кирова 20	66501.95	66696.95
Кузьмина 34а	96398.59	96398.38
Лермонтова 62/2	202672.10	203394.81
Можайского 17/1	66946.70	67331.68
Ойунского 23 Первая	304115.48	305453.81
Орджоникидзе 18	138107.99	138316.79
Пояркова 21	116414.46	117425.29
С. Данилова 1а	21987.60	21987.59
Юнимолл	85113.92	87588.82
ул. 203 мкр, 8 корпус	406149.87	406719.09
ул. Ф.Попова д. 15	23974.50	24365.46

Ввод [138]:

merged.groupby(by='Отделение').sum()[['Оплата RUB']].hist()



Ввод [134]:

merged.columns

Out[134]:

Index(['ID операции', 'Data_text_UDS', 'Номер чека', 'ID Клиента', 'Клиент', 'Уровень', 'Кассир_x', 'Тип операции', 'Счет RUB', 'Скидка RUB', 'Оплата RUB', 'Оплата баллы/ Оплата баллами', 'Оплата сертификатом', 'Состояние', 'Филиал', 'datetime', 'week', 'day_name', 'year', 'month', 'day', 'date', 'cut_avgcheck', 'ABCOплата RUB', 'Ср чек RUB', 'Учетный день', 'Отделение', 'Номер кассы', 'Кассир_y', 'Сумма со скидкой, р.'], dtype='object')

Ввод []:

Ввод [760]:

```
#iiko_bascets =  
s = set()  
  
iiko['Влюдо'].apply(lambda x: x)
```

Out[760]:

```
0          Булочка с помадкой 115 г  
1          Самса с курицей 140 г  
2          Кекс домашний с изюмом  
3          Эчпочмак 100 г  
5          Пончик жареный 85 г  
...  
78574      Пирожок с яйцом и зел. луком 60 г  
78575      Вагет Провансаль 150 г  
78576      Булочка Улитка (с изюмом) 100 г  
78578      Крендель  
78579      Хачапури 90 г  
Name: Влюдо, Length: 61105, dtype: object
```

Ввод []:

Ввод []:

Ввод []:

Ввод [658]:

#TODO странно этот не нашелся, возможно Булочка
iiko.loc[12]

Out[658]: Учетный день 2022-10-03 00:00:00
Отделение 203 мкр корпус 22
Номер кассы 25
Кассир Коркина Полина
Номер чека 35316.0
Номер смены 320
Номер карты оплаты *****6968
Блюдо Пирожок жареный с мясом и рисом 130г
Процент скидки 0.2881
Unnamed: 9 NaN
Тип скидки UDS Discount
Количество блюд 1
Сумма без скидки, р. 59
Сумма со скидкой, р. 42.0
Name: 12, dtype: object

Ввод [657]:

iiko[iiko['Тип скидки'] == 'UDS Discount']

Out[657]:

	Учетный день	Отделение	Номер кассы	Кассир	Номер чека	Номер смены	Номер карты оплаты	Блюдо	Процент скидки	Unnamed: 9	Тип скидки	Количество блюд	Сумма без скидки, р.	Сумма со скидкой, р.
12	2022-10-03	203 мкр корпус 22	25	Коркина Полина	35316.0	320	*****6968	Пирожок жареный с мясом и рисом 130г	0.2881	NaN	UDS Discount	1	59	42.00
16	2022-10-03	203 мкр корпус 22	25	Коркина Полина	35318.0	320	*****1982	Ромовая Баба 180 г	0.1176	NaN	UDS Discount	1	85	75.00
18	2022-10-03	203 мкр корпус 22	25	Коркина Полина	35320.0	320	*****5867	Булочка Синнамон 150 г	0.0743	NaN	UDS Discount	1	79	73.13
19	2022-10-03	203 мкр корпус 22	25	Коркина Полина	35320.0	320	*****5867	Круассан с шоколадом	0.0744	NaN	UDS Discount	1	99	91.63
20	2022-10-03	203 мкр корпус 22	25	Коркина Полина	35320.0	320	*****5867	Пончик жареный 85 г	0.0744	NaN	UDS Discount	1	79	73.12
...
78521	2022-10-09	Юнимолл	11	Валитова Марина Михайловна	119725.0	876	DDDD*****6212	Расстегай с рыбой 80 г	0.1490	NaN	UDS Discount	1	79	67.23
78522	2022-10-09	Юнимолл	11	Валитова Марина Михайловна	119725.0	876	DDDD*****6212	Самса с курицей 140 г	0.1490	NaN	UDS Discount	1	92	78.29
78523	2022-10-09	Юнимолл	11	Валитова Марина Михайловна	119725.0	876	DDDD*****6212	Сосиска в тесте 100 г	0.1491	NaN	UDS Discount	2	158	134.45
78556	2022-10-09	Юнимолл	11	Валитова Марина Михайловна	119734.0	876	(нет карты)	Самса с курицей 140 г	0.0673	NaN	UDS Discount	4	368	343.23
78557	2022-10-09	Юнимолл	11	Валитова Марина Михайловна	119734.0	876	(нет карты)	Хачапури 90 г	0.0673	NaN	UDS Discount	2	152	141.77

12022 rows × 14 columns

Ввод [659]:

ddd = iiko.loc[12, 'Учетный день']
ddd

Out[659]: Timestamp('2022-10-03 00:00:00')

Ввод [660]:

ccc = iiko.loc[12, 'Номер чека']
ccc

Out[660]: 35316.0

Ввод []:

uds[35316

Ввод [663]:

uds[(uds['date']==ddd)]

Out[663]:

	ID операции	ID Дата_text_UDS	Номер чека	ID Клиента	Клиент	Уровень	Кассир	Тип операции	Счет RUB	Скидка RUB	...	Состояние	Филиал	datetime	week	day_name	year	month
113511	518665260.0	03.10.2022 22:26:53	282531.0	1099564273007	Яна Уарова	1.0	Уарова.Я.М	Стандартная покупка	98.0	0.0	...	Выполнена	NaN	2022-10-03 22:26:53	40	Monday	2022	10
113512	518660456.0	03.10.2022 22:15:30	282526.0	1099576178135	Слава Иванов	1.0	Уарова.Я.М	Стандартная покупка	179.9	0.0	...	Выполнена	NaN	2022-10-03 22:15:30	40	Monday	2022	10
113513	518656324.0	03.10.2022 22:06:02	282522.0	1099576689522	Антон Михайлов	1.0	Уарова.Я.М	Стандартная покупка	124.6	0.0	...	Выполнена	NaN	2022-10-03 22:06:02	40	Monday	2022	10
113514	518650579.0	03.10.2022 21:56:13	282519.0	1099556798881	Семен Афанасьев	1.0	Уарова.Я.М	Стандартная покупка	123.9	0.0	...	Выполнена	NaN	2022-10-03 21:56:13	40	Monday	2022	10
113515	518648499.0	03.10.2022 21:51:06	282516.0	1099554913692	Александр Бояров	1.0	Уарова.Я.М	Стандартная покупка	90.3	0.0	...	Выполнена	NaN	2022-10-03 21:51:06	40	Monday	2022	10
...
115571	518298083.0	03.10.2022 8:04:56	21353.0	1099563998520	Наталья Грищенко	1.0	Кретова Татьяна Николаевна	Стандартная покупка	189.0	0.0	...	Выполнена	Павильон	2022-10-03 08:04:56	40	Monday	2022	10
115572	518298079.0	03.10.2022 8:04:45	281617.0	1099556824735	Анжелика Захарова	1.0	Уарова.Я.М	Стандартная покупка	368.0	0.0	...	Выполнена	NaN	2022-10-03 08:04:45	40	Monday	2022	10
115573	518298047.0	03.10.2022 8:02:23	281616.0	1099576609357	Валера Решетников	1.0	Уарова.Я.М	Стандартная покупка	218.0	0.0	...	Выполнена	NaN	2022-10-03 08:02:23	40	Monday	2022	10
115574	518298032.0	03.10.2022 8:01:22	281615.0	1099567508284	+79142857912	1.0	Уарова.Я.М	Стандартная покупка	79.0	0.0	...	Выполнена	NaN	2022-10-03 08:01:22	40	Monday	2022	10
115575	518298029.0	03.10.2022 8:01:07	21352.0	1099564925363	Варвара Михайлова	1.0	Кретова Татьяна Николаевна	Стандартная покупка	237.0	0.0	...	Выполнена	Павильон	2022-10-03 08:01:07	40	Monday	2022	10

1978 rows × 23 columns

Ввод [662]:

uds[(uds['date']==ddd)&(uds['Номер чека']==35316)]

Out[662]:

ID операции	Дата_text_UDS	Номер чека	ID Клиента	Клиент	Уровень	Кассир	Тип операции	Счет RUB	Скидка RUB	...	Состояние	Филиал	datetime	week	day_name	year	month	day	date	cut_avgcheck
0 rows × 23 columns																				

Ввод [632]:

uds[(uds['day'] == 3)&(uds['month'] == 10)&(uds['Оплата RUB'] == 311)][['ID операции', 'Дата_text_UDS', 'Номер чека', 'ID Клиента', 'Клиент', 'Кассир', 'Тип операции', 'Счет RUB', 'Скидка RUB', 'Оплата RUB', 'Оплата баллы/ Оплата баллами', 'Оплата сертификатом', 'Состояние', 'Филиал', 'date', 'cut_avgcheck']]

Out[632]:

ID операции	Дата_text_UDS	Номер чека	ID Клиента	Клиент	Кассир	Тип операции	Счет RUB	Скидка RUB	Оплата RUB	Оплата баллы/ Оплата баллами	Оплата сертификатом	Состояние	Филиал	date	cut_avgcheck	
114955	518317499.0	03.10.2022 12:13:58	21465.0	1099576019845	Валерия Николаева	Кретьова Татьяна Николаевна	Стандартная покупка	311.0	0.0	311.0	0.0	0.0	Выполнена	Павильон	2022-10-03	200-750

Ввод [666]:

iiko[iiko['Номер чека']==281615]

Out[666]:

	Учетный день	Отделение	Номер кассы	Кассир	Номер чека	Номер смены	Номер карты оплаты	Блюдо	Процент скидки	Unnamed: 9	Тип скидки	Количество блюд	Сумма без скидки, р.	Сумма со скидкой, р.
6163	2022-10-03	Ойунского 23 Первая	5	Уарова.Яна Михайловна	281615.0	1286	DDDD*****2891	Багет Провансаль с колбасой 150 г	0.0	NaN	NaN	1	79	79.0

7 Дальнейшая работа в этом направлении

- распределение и динамика суперклиентов по ТТ в динамике по месяцам
- изучение корзины суперклиентов для нахождения способов роста суперклиентов
- анкетирование, обзвон, возможно сбор - подарки - личный контакт для изучения с целью удержания и роста
- нахождение других типов клиентов - "кормильцев", с помощью паттернов корзины
- пересмотреть расчет, может стоит смотреть без - Список часто покупающих на сумму более 3000 руб (Андрей Крылов, Роза Шейхрамова ...)

Ввод []:

Ввод []:

8 TO DELETE

Ввод []:

8.1 (УДАЛИТЬ) Сводные данные по количеству визитов клиента в месяц

Ввод [313]:

#Группировка по клиентам и месяцам - посмотрим сколько раз в месяц ходят клиенты
uds_by_clnt = uds.groupby(by=['month','ID Клиента'])\
 .aggregate({'Клиент':'first','Оплата RUB':'sum','Оплата баллы/ Оплата баллами':'sum','Оплата сертификатом':'sum','ID операции':'count'})\
 .sort_values(by='Оплата RUB', ascending=False)\
 .rename(columns={'ID операции':'чеков'})
uds_by_clnt

Out[313]:

	month	ID Клиента	Клиент	Оплата RUB	Оплата баллы/ Оплата баллами	Оплата сертификатом	чеков
73003	9	1099574768293	Андрей Крылов	20360.5	2677.0	0.0	11
96756	10	1099574768293	Андрей Крылов	16833.0	0.0	0.0	7
52650	8	1099574768293	Андрей Крылов	16261.0	0.0	0.0	6
61053	9	109956887839	Роза Шейхрамова	15820.0	434.0	29.0	19
128315	12	1099555471162	Валентина Петрова	15460.6	1251.0	0.0	16
...
146750	12	1099581587527	+79246685581	10.0	0.0	0.0	1
26927	7	1099559929593	Оксана Фиськова	5.0	0.0	0.0	1
103926	10	1099579017084	+79142616170	5.0	0.0	0.0	1
113591	11	1099561190152	+79963177987	5.0	0.0	0.0	1
99219	10	1099577195042	Семен Кириллин	5.0	0.0	0.0	1

148641 rows × 7 columns

Ввод [315]:

uds_by_clnt.info()

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 148641 entries, 73003 to 99219  
Data columns (total 7 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   month                 148641 non-null  int64  
1   ID Клиента            148641 non-null  int64  
2   Клиент                 148641 non-null  object  
3   Оплата RUB            148641 non-null  float64  
4   Оплата баллы/ Оплата баллами  148641 non-null  float64  
5   Оплата сертификатом   148641 non-null  float64  
6   чеков                  148641 non-null  int64  
dtypes: float64(3), int64(3), object(1)  
memory usage: 9.1+ MB
```

Ввод [31]:

uds_by_clnt.to_excel('группировка по клиентам и месяцам с 01.06.2022 по 31.12.2022.xlsx')