In [1]:

```
import os
import re

import numpy as np
import pandas as pd
import warnings
from openpyxl import load_workbook
import seaborn as sns
```

In [21]:

```
warnings.filterwarnings("ignore")
```

In [2]:

```
my_path = rf"C:\Dmitry"
my_path = os.getcwd()
```

In [3]:

```
df = pd.read_csv(rf"{my_path}\all_df_flats.csv", delimiter="$")
df.head()
```

Out[3]:

	link	last_update	title	JK	start_price	start_date	building
0	**https://spb.cian.ru/sale/flat/265689456/**	**13** фев, **15:46**	Студия, **21,25** м²	в ЖК «Авиатор **(Aviator)»**	**3 761 250** ₽	**21** окт **2021**	, Корпус **3 (Дом 29к2)**
1	**https://spb.cian.ru/sale/flat/270279423/**	**14** фев, **05:37**	**2**-комн. квартира, **54,9** м²	в ЖК **«Ultra City (**Ультра Сити)»	**13 214 832** ₽	**17** фев **2022**	, Дом **26 (2** очередь, корпус **3)**
2	**https://spb.cian.ru/sale/flat/268907278/**	**19** янв, **16:40**	Студия, **28,83** м²	в ЖК **«Simple** (Симпл)»	**NaN**	**NaN**	, Секция **4**
3	**https://spb.cian.ru/sale/flat/275425156/**	**1** фев, **10:54**	**1**-комн. квартира, **38,11** м²	в ЖК «ЦДС **Dreamline** (Дримлайн)»	**6 966 085** ₽	**1** июл **2022**	, Корпус **1**
4	**https://spb.cian.ru/sale/flat/270498477/**	**9** фев, **18:22**	Студия, **21,4** м²	в ЖК «Цветной Город»	**3 400 460** ₽	**24** фев **2022**	, квартал **19** дом **15**

5 rows × 36 columns

In [5]:

```
df1=df[~df.index.isin(a.index)]
print("Количество после удаления неспарсенных: "+str(len(df1)))
df1=df1[df1['last_update'] != 'unpublished']
print("Количество после удаления снятых с публикации ссылок: "+str(len(df1)))
df1=df1.drop_duplicates()
print("Количество после удаления дублей: "+str(len(df1)))
```

```
Количество после удаления неспарсенных: 56179
Количество после удаления снятых с публикации ссылок: 48562
Количество после удаления дублей: 43012
```

In [6]:

```
q=df1.copy()
q = q.groupby(['link'])['link'].count().reset_index(name ='num')
q=q[q['num'] > 1]
q
```

Out[6]:

	link	num
33	**https://spb.cian.ru/sale/flat/246152880/**	**2**
71	**https://spb.cian.ru/sale/flat/246154107/**	**2**
132	**https://spb.cian.ru/sale/flat/246158053/**	**2**
141	**https://spb.cian.ru/sale/flat/246158589/**	**2**
161	**https://spb.cian.ru/sale/flat/246160386/**	**2**
...	**...**	**...**
35394	**https://spb.cian.ru/sale/flat/281976255/**	**2**
35395	**https://spb.cian.ru/sale/flat/281976257/**	**2**
35396	**https://spb.cian.ru/sale/flat/281976258/**	**2**
35397	**https://spb.cian.ru/sale/flat/281976259/**	**2**
35398	**https://spb.cian.ru/sale/flat/281976261/**	**2**

90 rows × 2 columns

In [7]:

```
df1[df1['link'] == 'https://spb.cian.ru/sale/flat/246152880/']
df1=df1.fillna("")
df1
```

Out[7]:

	link	last_update	title	JK	start_price	start_date
0	**https://spb.cian.ru/sale/flat/265689456/**	**13** фев, **15:46**	Студия, **21,25** м²	в ЖК «Авиатор **(Aviator)»**	**3 761 250** ₽	**21** окт **2021**
1	**https://spb.cian.ru/sale/flat/270279423/**	**14** фев, **05:37**	**2**-комн. квартира, **54,9** м²	в ЖК **«Ultra City (**Ультра Сити**)»**	**13 214 832** ₽	**17** фев **2022**
2	**https://spb.cian.ru/sale/flat/268907278/**	**19** янв, **16:40**	Студия, **28,83** м²	в ЖК **«Simple (**Симпл**)»**		
3	**https://spb.cian.ru/sale/flat/275425156/**	**1** фев, **10:54**	**1**-комн. квартира, **38,11** м²	в ЖК «ЦДС **Dreamline (**Дримлайн**)»**	**6 966 085** ₽	**1** июл **2022**
4	**https://spb.cian.ru/sale/flat/270498477/**	**9** фев, **18:22**	Студия, **21,4** м²	в ЖК «Цветной Город»	**3 400 460** ₽	**24** фев **2022**

...	link ...	last_update ...	title ...	JK ...	start_price ...	start_date ...
56256	**https://spb.cian.ru/sale/flat/283204867/**	сегодня, **08:15**	**1**-комн. апартаменты, **55,43** м²	в ЖК «Апартаменты «е.квартал "Мир внутри"»»	**17 737 600** ₽	**2** фев **2023**
56257	**https://spb.cian.ru/sale/flat/280292013/**	**19** фев, **16:44**	**1**-комн. квартира, **38,96** м²	в ЖК «Левитан»	**5 688 160** ₽	**18** ноя **2022**
56259	**https://spb.cian.ru/sale/flat/279350873/**	**15** фев, **09:40**	Студия, **29,77** м²	в ЖК «Апарт-отель **25/7** Заневский»	**8 139 999** ₽	**21** окт **2022**
56260	**https://spb.cian.ru/sale/flat/283456755/**	**8** фев, **20:57**	**1**-комн. квартира, **47,21** м²	в ЖК «Огни Залива»		
56261	**https://spb.cian.ru/sale/flat/280581997/**	**14** фев, **09:10**	Студия, **24,84** м²	в ЖК «**Astra Marine** на набережной»		

43012 rows × 36 columns

In [8]:

```
df2=df1.copy()
df2 = df2.astype(str)
#df2=df2.fillna("")
df2=df2.groupby('link').agg({lambda x: '@'.join(x)}).reset_index()
df2.columns = df2.columns.get_level_values(0)
#print(df2.columns)
```

In [9]:

```
pd.set_option('display.max_columns', None)
#df2.replace("nan", "")
df2[df2['link'] == 'https://spb.cian.ru/sale/flat/246152880/'] #ебань с объединением строк
```

Out[9]:

	link	last_update	title	JK	start_price	start_d
33	**https://spb.cian.ru/sale/flat/246152880/**	сегодня, **16:59@Обновлено:** сегодня, **16:59**	**3**-комн. квартира, **80,1** м²**@3**-комн. квартира, **80...**	в ЖК «Морская набережная»@ЖК «Морская набережная»	**14 177 700 ₽@14 177 700 ₽**	**4** дек **2020** дек **20**

In [ ]:

```
df3 = df2.copy()

df3['rooms']=np.nan
for i in range(len(df3)):
    if df3['title'][i].replace('\xa0', ' ').replace(" ", "").find('Студ')!=-1:
      df3['rooms'][i]='студия'
    elif df3['title'][i].replace('\xa0', ' ').replace(" ", "").find('Апартаменты-студ')!=-1:
```

```
      df3['rooms'][i]='многокомнатная'
    elif df3['title'][i].replace('\xa0', ' ').replace(" ",
"").find('Многокомнатнаякв')!=-1:
      df3['rooms'][i]='многокомнатная'
    elif df3['title'][i].replace('\xa0', ' ').replace(" ", "").find('комн')!=-1:
        try:
            df3['rooms'][i]=re.match(r'(?<=)(\d*)(?=-комн)', df3['title']
[i].replace('\xa0', ' ')).group(0)
        except AttributeError:
            df3['rooms'][i]=re.match(r'(?<=)(\d*)(?=-комн)', df3['title']
[i].replace('\xa0', ' '))


df3=df3[df3['link'].str.replace('\xa0', ' ').str.find('https')==0]
df3['JK']=df3['JK'].str.extract(r'(?<=«)(.*?)(?=»)')
df3['start_price']=df3['start_price'].str.replace(" ", "").str.replace("@", "").s
tr.extract(r'(?<=)(.*?)(?=₽)').astype('float')
df3['start_date']=df3['start_date'].str.replace('\xa0', ' ').str.replace(" ", "")
.str.replace("янв", "-01-").str.replace("фев", "-02-").str.replace("мар", "-03-")
.str.replace("апр", "-04-").str.replace("мая", "-05-").str.replace("июн", "-06-")
.str.replace("июл", "-07-").str.replace("авг", "-08-").str.replace("сен", "-09-")
.str.replace("окт", "-10-").str.replace("ноя", "-11-").str.replace("дек", "-12-")
.str.extract(r'(\d{2}-\d{2}-\d{4})').astype('datetime64[ns]')
df3['building']=df3['building'].str.extract(r'(?<=, )([^@]*)')
df3['date_readiness']=df3['date_readiness'].str.extract(r'(?<=)([^@]*)')
df3['metro_station']=df3['metro_station'].str.extract(r'(?<=)([А-Яа-я\.\s]*)')
df3['city']=df3['city'].str.extract(r'(?<=)([^@]*)')
for i in range(len(df3)):
    if (df3['district'][i]=='Гатчина') | (df3['district'][i]=='Волхов') |(df3['di
strict'][i]=='Выборг') | (df3['district'][i]=='Кировск') | (df3['district'][i]=='
Кингисепп') | (df3['district'][i]=='Всеволожск'):
      df3['house'][i]=df3['street'][i]
      df3['street'][i]=df3['area'][i]
      df3['area'][i]='-'
    elif df3['district'][i].replace('\xa0', ' ').replace(" ", "").find('р-н')!=-1
:
        try:
            df3['district'][i]=re.search(r'(?<=р-н )([А-Яа-я]*)', df3['district']
[i].replace('\xa0', ' ')).group(0)
        except AttributeError:
            df3['district'][i]=re.search(r'(?<=р-н )([А-Яа-я]*)', df3['district']
[i].replace('\xa0', ' '))
    elif df3['district'][i].replace('\xa0', ' ').replace(" ", "").find('район')!=
-1:
      try:
            df3['district'][i]=re.search(r'(?<=)([А-Яа-я]*)(?= район)', df3['dist
rict'][i].replace('\xa0', ' ')).group(0)
      except AttributeError:
            df3['district'][i]=re.search(r'(?<=)([А-Яа-я]*)(?= район)', df3['dist
rict'][i].replace('\xa0', ' '))
df3['district']=df3['district'].str.extract(r'(?<=)([^@]*)')
df3['area']=df3['area'].str.extract(r'(?<=)([А-Яа-я\.\s]*)')
df3['street']=df3['street'].str.extract(r'(?<=)([0-9А-Яа-я\.\s]*)')
df3['house']=df3['house'].str.extract(r'(?<=)([0-9А-Яа-я\.\s]*)')
df3['developer']=df3['developer'].str.extract(r'(?<=)([А-Яа-яA-Za-z\.«»\s]*)')
df3['date_readiness']=df3['date_readiness'].str.extract(r'(?<=в )(.*)')
df3['full_price']=df3['full_price'].str.replace('\xa0', ' ').str.replace(" ", "")
.str.extract(r'(?<=)(.*?)(?=₽)').astype('float')
df3['price_for_sq_meter']=df3['price_for_sq_meter'].str.replace('\xa0', ' ').str.
replace(" ", "").str.replace("@", "").str.extract(r'(?<=)(.*?)(?=₽)').astype('flo
at')
df3['full_square']=df3['full_square'].str.replace('\xa0', ' ').str.replace(" ", "
").str.replace(",", ".").str.extract(r'(?<=)(.*?)(?=м)').astype('float')
```

```
df3['living_square']=df3['living_square'].str.replace('\xa0', ' ').str.replace("
", "").str.replace(",", ".").str.extract(r'(?<=)(.*?)(?=м)').astype('float')
mean_living_stud=(df3['living_square'][df3['rooms']=='студия']/df3['full_square']
[df3['rooms']=='студия']).mean()
for i in range(len(df3)):
    if df3['rooms'][i]=='студия':
        df3['living_square'][i]=mean_living_stud*df3['full_square'][i]
df3['views']=df3['date_creation'].str.replace('\xa0', ' ').str.replace(" ", "").s
tr.extract(r'(?<=)(\d*)(?=просмотров|просмотра)').astype('float')
df3['date_creation']=df3['date_creation'].str.replace('\xa0', ' ').str.replace(".
","-").str.extract(r'(?<=объявления )([0-9-]*)(?=)').astype('datetime64[ns]')

df3["inf1"] = df3['kitchen_square'] +"_"+ df3['floor'] +"_"+ df3['finishing']
df3["inf2"] = df3['info_about_flat_2'] +"_"+ df3['info_about_flat_3'] +"_"+ df3['
info_about_flat_4'] +"_"+ df3['info_about_flat_5'] +"_"+ df3['info_about_flat_6']
+"_"+ df3['info_about_flat_7'] +"_"+ df3['info_about_flat_8'] +"_"+ df3['info_abo
ut_flat_9']
df3["inf3"] = df3['info_about_house_1'] +"_"+ df3['info_about_house_2'] +"_"+ df3
['info_about_house_3']

df3['kitchen']=df3['inf1'].str.replace('\xa0', ' ').str.replace(" ", "").str.repl
ace(",", ".").str.extract(r'(?<=)([0-9\.]*)(?=м²)').astype('float')
mean_kitchen_stud=(df3['kitchen'][df3['rooms']=='студия']/df3['full_square'][df3[
'rooms']=='студия']).mean()
for i in range(len(df3)):
    if df3['rooms'][i]=='студия':
        df3['kitchen'][i]=mean_kitchen_stud*df3['full_square'][i]
df3['floor1']=df3['inf1'].str.replace('\xa0', ' ').str.replace(" ", "").str.extra
ct(r'(?<=)(\d*)(?=из)').astype('float')
df3['floor_house']=df3['inf1'].str.replace('\xa0', ' ').str.replace(" ", "").str.
extract(r'(?<=из)(\d*)(?=)').astype('float')
df3['last_floor'] = np.where(df3['floor1']==df3['floor_house'], 1, 0)
df3['first_floor'] = np.where(df3['floor1']==1, 1, 0)
df3['hight']=df3['inf2'].str.replace('\xa0', ' ').str.replace(" ", "").str.replac
e(",", ".").str.extract(r'(?<=)([0-9\.]*)(?=м_)').astype('float')
df3['balcony']=df3['inf2'].str.replace('\xa0', ' ').str.replace(" ", "").str.repl
ace(",", ".").str.extract(r'(?<=)([0-9]*)(?=балк)').astype('float').fillna(0) + d
f3['inf2'].str.replace('\xa0', ' ').str.replace(" ", "").str.replace(",", ".").st
r.extract(r'(?<=)([0-9]*)(?=лодж)').astype('float').fillna(0)
df3['balcony']=df3['balcony'].replace(0, np.nan)
df3['toilet_count']=df3['inf2'].str.replace('\xa0', ' ').str.replace(" ", "").str
.replace(",", ".").str.extract(r'(?<=)([0-9]*)(?=разд)').astype('float').fillna(0
) + df3['inf2'].str.replace('\xa0', ' ').str.replace(" ", "").str.replace(",", ".
").str.extract(r'(?<=)([0-9]*)(?=совм)').astype('float').fillna(0)
df3['toilet_count']=df3['toilet_count'].replace(0, np.nan)
df3['toilet_type']=np.nan
for i in range(len(df3)):
    if df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('совмещенный,')!
=-1:
      df3['toilet_type'][i]='оба'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('совм')!=-1:
      df3['toilet_type'][i]='совмещенный'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('разд')!=-1:
      df3['toilet_type'][i]='раздельный'

df3['lift_pass']=df3['inf3'].str.replace('\xa0', ' ').str.replace(" ", "").str.re
place(",", ".").str.extract(r'(?<=)([0-9]*)(?=пасс)').astype('float')
df3['lift_gruz']=df3['inf3'].str.replace('\xa0', ' ').str.replace(" ", "").str.re
place(",", ".").str.extract(r'(?<=)([0-9]*)(?=груз)').astype('float')
df3['remont']=np.nan
for i in range(len(df3)):
    if df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Чистоваясмеб')!
=-1:
```

```
      df3['remont'][i]='чистовая с мебелью'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Чист')!=-1:
      df3['remont'][i]='чистовая'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Без')!=-1:
      df3['remont'][i]='без отделки'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Черн')!=-1:
      df3['remont'][i]='черновая'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Предч')!=-1:
      df3['remont'][i]='предчистовая'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Сотдел')!=-1:
      df3['remont'][i]='с отделкой'

df3['parking']=np.nan
for i in range(len(df3)):
    if df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Откр')!=-1:
      df3['parking'][i]='открытая'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Мног')!=-1:
      df3['parking'][i]='многоуровневая'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Подз')!=-1:
      df3['parking'][i]='подземная'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Гост')!=-1:
      df3['parking'][i]='гостевая'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ",
"").find('Отдельнаямног')!=-1:
      df3['parking'][i]='многоуровневая'

df3['house_type']=np.nan
for i in range(len(df3)):
    if df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Панел')!=-1:
      df3['house_type'][i]='панельный'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Кирп')!=-1:
      df3['house_type'][i]='кирпичный'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Монолитно-кир
')!=-1:
      df3['house_type'][i]='монолитно-кирпичный'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Монол')!=-1:
      df3['house_type'][i]='монолитный'
    elif df3['inf3'][i].replace('\xa0', ' ').replace(" ", "").find('Блоч')!=-1:
      df3['house_type'][i]='блочный'

df3['windows']=np.nan
for i in range(len(df3)):
    if df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Наулицуидвор')!
=-1:
      df3['windows'][i]='на улицу и двор'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Водв')!=-1:
      df3['windows'][i]='во двор'
    elif df3['inf2'][i].replace('\xa0', ' ').replace(" ", "").find('Наул')!=-1:
      df3['windows'][i]='на улицу'


df3.head()
```

In [ ]:

```
df_copy=df3.copy()

df_copy['floor_house']=df_copy['inf1'].str.replace('\xa0', ' ').str.replace(" ",
"").str.extract(r'(?<=из)(\d*)(?=)').astype('float')
df_copy['last_floor'] = np.where(df_copy['floor1']==df_copy['floor_house'], 1, 0)
df_copy['first_floor'] = np.where(df_copy['floor1']==1, 1, 0)
df_copy[df_copy['last_floor']==1]
```

In [ ]:

```
df4 = df3.copy()
df4=df4.drop(columns=['last_update',
                      'title',
                      'kitchen_square',
                      'floor',
                      'finishing',
                      'info_about_flat_1',
                      'info_about_flat_2',
                      'info_about_flat_3',
                      'info_about_flat_4',
                      'info_about_flat_5',
                      'info_about_flat_6',
                      'info_about_flat_7',
                      'info_about_flat_8',
                      'info_about_flat_9',
                      'info_about_house_1',
                      'info_about_house_2',
                      'info_about_house_3',
                      'inf1',
                      'inf2',
                      'inf3'])

df4=df4[df4['JK'].notna()]
df4
```

In [ ]:

```
df_jk = pd.DataFrame(load_workbook(rf"{my_path}\data_JK.xlsx", read_only=False)[
'Лист1'].values)
df_jk.columns = df_jk.iloc[0]
df_jk=df_jk[1:]
df_jk['JK']=df_jk['JK'].str.extract(r'(?<=«)(.*?)(?=»)')
df_jk['height']=df_jk['height'].str.replace('\xa0', ' ').str.replace(" ", "").str
.replace(",", ".").str.extract(r'(?<=)([0-9\.]*)(?=м)').astype('float')
flats_jk = list(set(df4['JK']))
df_jk=df_jk[(df_jk['JK'].isin(flats_jk))]
df_jk=df_jk[['JK','JK_class', 'height', 'parking', 'finishing', 'material',
'developer', 'houses_built', 'houses_in_process']]
df_jk['parking']=df_jk['parking'].str.extract(r'([А-Яа-я\s]*)')
df_jk['finishing']=df_jk['finishing'].str.extract(r'([А-Яа-я\s]*)')
df_jk[df_jk['height'].isna()]
```

In [15]:

```
df5 = df4.copy()
df5 = df5.merge(df_jk, on='JK', how='left')
```

In [ ]:

```
df5['parking']=np.nan
for i in range(len(df5)):
    if pd.notnull(df5['parking_x'][i]):
      df5['parking'][i]=df5['parking_x'][i]
    else:
       df5['parking'][i]=df5['parking_y'][i]
df5['parking']=df5['parking'].replace('—', np.nan)
for i in range(len(df5)):
    if pd.notnull(df5['remont'][i]):
      df5['remont'][i]=df5['remont'][i]
    else:
       df5['remont'][i]=df5['finishing'][i]
```

```
df5['remont']=df5['remont'].replace('—', np.nan)
for i in range(len(df5)):
    if pd.notnull(df5['house_type'][i]):
      df5['house_type'][i]=df5['house_type'][i]
    else:
       df5['house_type'][i]=df5['material'][i]
df5['developer']=np.nan
for i in range(len(df5)):
    if (pd.notnull(df5['developer_x'][i]) and df5['developer_x'][i]!='ПРЕДСТАВИТЕЛЬ ЗАСТРОЙЩИКА' and df5['developer_x'][i]!='ЗАСТРОЙЩИК' and df5['developer_x'][i]!='КОНСУЛЬТАНТ'):
      df5['developer'][i]=df5['developer_x'][i]
    else:
       df5['developer'][i]=df5['developer_y'][i]

df5['developer'] = df5['developer'].str.replace('Застройщик «', '').str.replace('»', '').str.replace('ГК', '').str.replace('Группа', '').str.replace('Петербургская Строительная Компания', 'ПСК').str.replace('РосСтройИнвест', 'РСТИ').str.replace('Росстройинвест', 'РСТИ').str.replace('Строительный холдинг ', '').str.replace('Холдинг «', '').str.replace('Холдинг ', '').str.replace('СЗ ', '').str.replace('. Недвижимость-Северо-Запад', '').str.replace('. Недвижимость', '').str.replace('Санкт', '').str.replace('СПб', '')
df5.head()
```

In [18]:

```
df5.columns
```

Out[18]:

```
Index(['link', 'JK', 'start_price', 'start_date', 'building', 'date_readiness',
       'full_price', 'price_for_sq_meter', 'metro_station', 'metro_distance',
       'city', 'district', 'area', 'street', 'house', 'full_square',
       'living_square', 'developer_x', 'date_creation', 'rooms', 'views',
       'kitchen', 'floor1', 'floor_house', 'last_floor', 'first_floor',
       'hight', 'balcony', 'toilet_count', 'toilet_type', 'lift_pass',
       'lift_gruz', 'remont', 'parking_x', 'house_type', 'windows', 'JK_class',
       'height', 'parking_y', 'finishing', 'material', 'developer_y',
       'houses_built', 'houses_in_process', 'parking', 'developer'],
      dtype='object')
```

In [20]:

```
#df6.to_csv("flats_25.03.csv")
```

In [ ]:

```
for i in list(df6.columns):
    print(f"Количество NA в {i}: {df6[i].isna().sum()}, доля: {df6[i].isna().sum()*100/df6.shape[0]:.2f}%")
```