# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

## Кафедра прикладних інформаційних систем

**Звіт до лабораторної роботи №2**

# з курсу «Big Data»

*Студента 4 курсу*

*Групи ПП-41*

*спеціальності 122*

*«Комп'ютерні науки»*

*ОП«Прикладне програмування»*

*Постоюка Максима*

## Київ – 2023

# Тема

## Розвідувальний аналіз даних (EDA). Складання аналітичного звіту

# Мета

Метою лабораторної роботи є отримання практичних навичок виконання розвідувального аналізу даних, використовуючи пакети jupyter, pandas, seaborn. Ознайомлення з методологією складання аналітичного звіту для зовнішнього користувача інформаційного продукту.

# Завдання

### Виконайте дослідження domain experience стосовно американського ринку нерухомості. Ознайомтесь з декількома прикладами аналітичних продуктів від топових гравців на американському ринку, направлених на інвесторів. Питання, які потрібно опрацювати:

### Як топові компанії на ринку складають звіти по нерухомості?

### Які графіки використовуються для донесення інформації?

### Які співвідношення між якими даними по ринку є показовими для інвесторів / керівників агенцій нерухомості?

### Яка термінологія використовується для опису закономірностей на ринку нерухомості?

### Завантажити файли з даними у папку проекту з посилання:

### <https://www1.nyc.gov/site/finance/taxes/property-rolling-sales-data.page>

### Очистити дані.

### Виконайте розвідувальний аналіз, щоб дізнатися, де є викиди або відсутні значення, вирішіть, як ви їх обробляти, переконайтеся, що дати відформатовані правильно, значення, які ви вважаєте числовими, розглядаються як такі і т.д.

### Виконайте аналіз розвідувальних даних (отриманих результатів) для візуалізації та зіставлення за житловими масивами та за часом. Почніть шукати осмислені закономірності у цьому наборі.

### Зберіть висновки до невеликий звіт для генерального директора (графіки, висновки з текстом у окремому файлі), який потребує належного оформлення висновків, структури тощо.

# Виконання завдання

У процесі дослідження domain experience були помічені такі особливості сфери аналітики американського ринку нерухомості:

1. Основними гравцями на ринку, які складають аналітичні звіти є Keller Williams Realty, RE/MAX, Coldwell Banker Real Estate, Century 21, Berkshire Hathaway HomeServices.
2. Для аналізу ринку в аналітичних звітах компаній використовуються такі графіки: гістограми та кругові діаграми. Рідше використовуються лінійні та площові та в одиничних випадках точкові діаграми.
3. Показовими для інвесторів та керівників агенцій нерухомості є графіки, які беруть певне значення (наприклад, ціну продажу, кількість продажів, певні зміни у вимогах до класу будівель) та співвідносять їх з теперешнім та попередніми роками. Також дуже поширені графіки, які демонструють значення у відсотках.
4. Терміни, які часто зустрічаються у аналітичних звітах ринку нерухомості: міський, приміський, спад, власний капітал, роздрібна торгівля, іпотека.

Код:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

import seaborn as sns

import warnings

warnings.filterwarnings("ignore", "is\_categorical\_dtype")

warnings.filterwarnings("ignore", "use\_inf\_as\_na")

warnings.simplefilter(action='ignore', category=FutureWarning)

pd.options.mode.chained\_assignment = None

mpl.rcParams["font.size"] = 18

mpl.rcParams['figure.figsize'] = [18,14]

df = pd.concat(

   map(pd.read\_excel, ['D:/Unik/BigData/Lab2/Data/rollingsales\_bronx.xlsx', 'D:/Unik/BigData//Lab2/Data/rollingsales\_brooklyn.xlsx', 'D:/Unik/BigData//Lab2/Data/rollingsales\_statenisland.xlsx']), ignore\_index=True )

#df = pd.concat(

#    map(pd.read\_excel, ['D:/Unik/BigData/Sokalskyi\_Ivan\_\_PP41\_BigData/Lab2/Data/rollingsales\_bronx.xlsx']), ignore\_index=True )

df.loc[df["EASEMENT"].isnull(), "EASEMENT"] = "Doesn't have an easement"

df.loc[df["EASEMENT"]=='', "EASEMENT"] = "Doesn't have an easement"

df.loc[df["EASEMENT"]==0, "EASEMENT"] = "Doesn't have an easement"

df.loc[df["APARTMENT NUMBER"].isnull(), "APARTMENT NUMBER"] = "Not an apartment"

df.loc[df["APARTMENT NUMBER"]=='', "APARTMENT NUMBER"] = "Not an apartment"

df.loc[df["APARTMENT NUMBER"]==0, "APARTMENT NUMBER"] = "Not an apartment"

df.loc[df["RESIDENTIAL UNITS"].isnull(), "RESIDENTIAL UNITS"] = "Doesn't have residential units"

df.loc[df["RESIDENTIAL UNITS"]=='', "RESIDENTIAL UNITS"] = "Doesn't have residential units"

df.loc[df["RESIDENTIAL UNITS"]==0, "RESIDENTIAL UNITS"] = "Doesn't have residential units"

df.loc[df["COMMERCIAL UNITS"].isnull(), "COMMERCIAL UNITS"] = "Doesn't have commercial units"

df.loc[df["COMMERCIAL UNITS"]=='', "COMMERCIAL UNITS"] = "Doesn't have commercial units"

df.loc[df["COMMERCIAL UNITS"]==0, "COMMERCIAL UNITS"] = "Doesn't have commercial units"

df.loc[df["TOTAL UNITS"].isnull(), "TOTAL UNITS"] = "Doesn't have total units"

df.loc[df["TOTAL UNITS"]=='', "TOTAL UNITS"] = "Doesn't have total units"

df.loc[df["TOTAL UNITS"]==0, "TOTAL UNITS"] = "Doesn't have total units"

df.loc[df["LAND SQUARE FEET"].isnull(), "LAND SQUARE FEET"] = "Land square feet is unknown"

df.loc[df["LAND SQUARE FEET"]=='', "LAND SQUARE FEET"] = "Land square feet is unknown"

df.loc[df["LAND SQUARE FEET"]==0, "LAND SQUARE FEET"] = "Land square feet is unknown"

df.loc[df["GROSS SQUARE FEET"].isnull(), "GROSS SQUARE FEET"] = "Gross square feet is unknown"

df.loc[df["GROSS SQUARE FEET"]=='', "GROSS SQUARE FEET"] = "Gross square feet is unknown"

df.loc[df["GROSS SQUARE FEET"]==0, "GROSS SQUARE FEET"] = "Gross square feet is unknown"

df.loc[df["BUILDING CLASS AT PRESENT"].isnull(), "BUILDING CLASS AT PRESENT"] = "Building class is unknown"

df.loc[df["BUILDING CLASS AT PRESENT"]=='', "BUILDING CLASS AT PRESENT"] = "Building class is unknown"

df.loc[df["BUILDING CLASS AT PRESENT"]==0, "BUILDING CLASS AT PRESENT"] = "Building class is unknown"

df.loc[df["BOROUGH"] == 2, "BOROUGH"] = "Bronx"

df.loc[df["BOROUGH"] == 4, "BOROUGH"] = "Queens"

df.loc[df["BOROUGH"] == 3, "BOROUGH"] = "Brooklyn"

df.loc[df["BOROUGH"] == 1, "BOROUGH"] = "Manhattan"

df.loc[df["BOROUGH"] == 5, "BOROUGH"] = "Staten Island"

df.loc[df["YEAR BUILT"].isnull(), "YEAR BUILT"] = '111111'

df.loc[df["YEAR BUILT"]=='', "YEAR BUILT"] = '111111'

df.loc[df["YEAR BUILT"]=='YEAR BUILT', "YEAR BUILT"] = '111111'

df['YEAR BUILT'] = df['YEAR BUILT'].map(int)

# create a list of our conditions

conditions = [

    (df['YEAR BUILT'] == 111111),

    (df['YEAR BUILT'] < 1900),

    (df['YEAR BUILT'] >= 1900) & (df['YEAR BUILT'] <= 1920),

    (df['YEAR BUILT'] >= 1921) & (df['YEAR BUILT'] <= 1940),

    (df['YEAR BUILT'] >= 1941) & (df['YEAR BUILT'] <= 1960),

    (df['YEAR BUILT'] >= 1961) & (df['YEAR BUILT'] <= 1980),

    (df['YEAR BUILT'] >= 1981) & (df['YEAR BUILT'] <= 2000),

    (df['YEAR BUILT'] >= 2001)

    ]

# create a list of the values we want to assign for each condition

values = ['Year unknown', '1900-', '1900-1920', '1921-1940', '1941-1960', '1961-1980', '1981-2000', '2001+']

# create a new column and use np.select to assign values to it using our lists as arguments

df['TIME PERIOD'] = np.select(conditions, values)

#bronx

dfBronx = df.loc[df["BOROUGH"]=='Bronx']

dfBronxFirstHalf = dfBronx.head(3157).reset\_index()

dfBronxSecondHalf = dfBronx.tail(len(dfBronx) - 3157).reset\_index()

dfBronxNeighborhoodsSize = dfBronxFirstHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfBronxNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsSize)

dfBronxNeighborhoodsSize.set\_xlim(0, 800)

dfBronxNeighborhoodsSize.set(title = 'Numbers of buildings sold by Bronx\'s neighborhoods')

plt.show()

dfBronxNeighborhoodsSize = dfBronxSecondHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfBronxNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsSize)

dfBronxNeighborhoodsSize.set\_xlim(0, 800)

dfBronxNeighborhoodsSize.set(title = 'Numbers of buildings sold by Bronx\'s neighborhoods')

plt.show()

dfBronxNeighborhoodsSizeByTimePeriod = dfBronxFirstHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfBronxNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Bronx\'s neighborhoods')

plt.ylabel('Number of buildings sold')

dfBronxNeighborhoodsSizeByTimePeriod.set\_ylim(0, 800)

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.show()

dfBronxNeighborhoodsSizeByTimePeriod = dfBronxSecondHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfBronxNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Bronx\'s neighborhoods')

dfBronxNeighborhoodsSizeByTimePeriod.set\_ylim(0, 800)

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.show()

dfBronxNeighborhoodsTimePeriod = dfBronx.groupby('TIME PERIOD')['NEIGHBORHOOD'].size().reset\_index(name = 'Numbers of buildings sold')

dfBronxNeighborhoodsTimePeriod =  sns.barplot(x='Numbers of buildings sold', y= 'TIME PERIOD', data=dfBronxNeighborhoodsTimePeriod)

dfBronxNeighborhoodsTimePeriod.set(title = 'Bronx numbers of buildings sold by time periods')

plt.show()

dfBronxNeighborhoodsMaxSalePrice = dfBronxFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfBronxNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsMaxSalePrice)

dfBronxNeighborhoodsMaxSalePrice.set(title = 'Bronx max sale price by neighborhoods')

dfBronxNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfBronxNeighborhoodsMaxSalePrice = dfBronxSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfBronxNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsMaxSalePrice)

dfBronxNeighborhoodsMaxSalePrice.set(title = 'Bronx max sale price by neighborhoods')

dfBronxNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfBronxNeighborhoodsMeanSalePrice = dfBronxFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfBronxNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsMeanSalePrice)

dfBronxNeighborhoodsMeanSalePrice.set(title = 'Bronx mean sale price by neighborhoods')

dfBronxNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfBronxNeighborhoodsMeanSalePrice = dfBronxSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfBronxNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsMeanSalePrice)

dfBronxNeighborhoodsMeanSalePrice.set(title = 'Bronx mean sale price by neighborhoods')

dfBronxNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfBronxSorted = dfBronx.sort\_values(by = ['BUILDING CLASS AT PRESENT'], ascending=False)

dfBronxSortedFirstHalf = dfBronxSorted.head(966).reset\_index()

dfBronxSortedSecondHalf = dfBronxSorted.tail(len(dfBronxSorted.index) - 966).reset\_index()

dfBronxNeighborhoodsSizeByTimePeriod = dfBronxSortedFirstHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfBronxNeighborhoodsSizeByTimePeriod.set(title = 'Bronx numbers of buildings sold by building class at time of sale and time period')

dfBronxNeighborhoodsSizeByTimePeriod.set\_ylim(0, 1000)

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.show()

dfBronxNeighborhoodsSizeByTimePeriod = dfBronxSortedSecondHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfBronxNeighborhoodsSizeByTimePeriod.set(title = 'Bronx numbers of buildings sold by building class at time of sale and time period')

dfBronxNeighborhoodsSizeByTimePeriod.set\_ylim(0, 1000)

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.show()

indexLandSquareFeetWholeArray = dfBronx[ (dfBronx['LAND SQUARE FEET']=='Land square feet is unknown')].index

indexLandSquareFeet = dfBronxFirstHalf[ (dfBronxFirstHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfBronxFirstHalf.drop(indexLandSquareFeet , inplace=True)

dfBronxNeighborhoodsMeanSalePrice = dfBronxFirstHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

new\_row = pd.DataFrame({'NEIGHBORHOOD': ['Mean Land Square feet unknown'], 'mean land square feet': [len(indexLandSquareFeetWholeArray)]})

dfBronxNeighborhoodsMeanSalePrice = pd.concat([dfBronxNeighborhoodsMeanSalePrice, new\_row], ignore\_index=True)

dfBronxNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsMeanSalePrice)

dfBronxNeighborhoodsMeanSalePrice.set(title = 'Bronx mean land square feet by neighborhoods')

dfBronxNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

indexLandSquareFeet = dfBronxSecondHalf[ (dfBronxSecondHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfBronxDropped = dfBronxSecondHalf.drop(indexLandSquareFeet , inplace=True)

dfBronxNeighborhoodsMeanSalePrice = dfBronxSecondHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

dfBronxNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfBronxNeighborhoodsMeanSalePrice)

dfBronxNeighborhoodsMeanSalePrice.set(title = 'Bronx mean land square feet by neighborhoods')

dfBronxNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

#queens

dfQueens = df.loc[df["BOROUGH"]=='Queens']

dfQueensFirstHalf = dfQueens.head(11662)

dfQueensSecondHalf = dfQueens.tail(len(dfQueens) - 11662)

dfQueensNeighborhoodsSize = dfQueensFirstHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfQueensNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsSize)

dfQueensNeighborhoodsSize.set(title = 'Numbers of buildings sold by Queens\'s neighborhoods')

dfQueensNeighborhoodsSize.set\_xlim(0, 4000)

plt.show()

dfQueensNeighborhoodsSize = dfQueensSecondHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfQueensNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsSize)

dfQueensNeighborhoodsSize.set(title = 'Numbers of buildings sold by Queens\'s neighborhoods')

dfQueensNeighborhoodsSize.set\_xlim(0, 4000)

plt.show()

dfQueensNeighborhoodsSizeByTimePeriod = dfQueensFirstHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfQueensNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Queens\'s neighborhoods')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfQueensNeighborhoodsSizeByTimePeriod.set\_ylim(0, 4000)

plt.show()

dfQueensNeighborhoodsSizeByTimePeriod = dfQueensSecondHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfQueensNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Queens\'s neighborhoods')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfQueensNeighborhoodsSizeByTimePeriod.set\_ylim(0, 4000)

plt.show()

dfQueensNeighborhoodsTimePeriod = dfQueens.groupby('TIME PERIOD')['NEIGHBORHOOD'].size().reset\_index(name = 'Numbers of buildings sold')

dfQueensNeighborhoodsTimePeriod =  sns.barplot(x='Numbers of buildings sold', y= 'TIME PERIOD', data=dfQueensNeighborhoodsTimePeriod)

dfQueensNeighborhoodsTimePeriod.set(title = 'Queens numbers of buildings sold by time periods')

plt.show()

dfQueensNeighborhoodsMaxSalePrice = dfQueensFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfQueensNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsMaxSalePrice)

dfQueensNeighborhoodsMaxSalePrice.set(title = 'Queens max sale price by neighborhoods')

dfQueensNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfQueensNeighborhoodsMaxSalePrice = dfQueensSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfQueensNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsMaxSalePrice)

dfQueensNeighborhoodsMaxSalePrice.set(title = 'Queens max sale price by neighborhoods')

dfQueensNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfQueensNeighborhoodsMeanSalePrice = dfQueensFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfQueensNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsMeanSalePrice)

dfQueensNeighborhoodsMeanSalePrice.set(title = 'Queens mean sale price by neighborhoods')

dfQueensNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfQueensNeighborhoodsMeanSalePrice = dfQueensSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfQueensNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsMeanSalePrice)

dfQueensNeighborhoodsMeanSalePrice.set(title = 'Queens mean sale price by neighborhoods')

dfQueensNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfQueensSorted = dfQueens.sort\_values(by = ['BUILDING CLASS AT PRESENT'], ascending=False)

dfQueensSortedFirstHalf = dfQueensSorted.head(4690).reset\_index()

dfQueensSortedSecondHalf = dfQueensSorted.tail(len(dfQueensSorted.index) - 4690).reset\_index()

dfQueensNeighborhoodsSizeByTimePeriod = dfQueensSortedFirstHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfQueensNeighborhoodsSizeByTimePeriod.set(title = 'Queens numbers of buildings sold by building class at time of sale and time period')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfQueensNeighborhoodsSizeByTimePeriod.set\_ylim(0, 4000)

plt.show()

dfQueensNeighborhoodsSizeByTimePeriod = dfQueensSortedSecondHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfQueensNeighborhoodsSizeByTimePeriod.set(title = 'Queens numbers of buildings sold by building class at time of sale and time period')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfQueensNeighborhoodsSizeByTimePeriod.set\_ylim(0, 4000)

plt.show()

indexLandSquareFeetWholeArray = dfQueens[ (dfQueens['LAND SQUARE FEET']=='Land square feet is unknown')].index

indexLandSquareFeet = dfQueensFirstHalf[ (dfQueensFirstHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfQueensFirstHalf.drop(indexLandSquareFeet , inplace=True)

dfQueensNeighborhoodsMeanSalePrice = dfQueensFirstHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

new\_row = pd.DataFrame({'NEIGHBORHOOD': ['Mean Land Square feet unknown'], 'mean land square feet': [len(indexLandSquareFeetWholeArray)]})

dfQueensNeighborhoodsMeanSalePrice = pd.concat([dfQueensNeighborhoodsMeanSalePrice, new\_row], ignore\_index=True)

dfQueensNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsMeanSalePrice)

dfQueensNeighborhoodsMeanSalePrice.set(title = 'Queens mean land square feet by neighborhoods')

dfQueensNeighborhoodsMeanSalePrice.set\_xlim(0, 13000)

plt.show()

indexLandSquareFeet = dfQueensSecondHalf[ (dfQueensSecondHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfQueensDropped = dfQueensSecondHalf.drop(indexLandSquareFeet , inplace=True)

dfQueensNeighborhoodsMeanSalePrice = dfQueensSecondHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

dfQueensNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfQueensNeighborhoodsMeanSalePrice)

dfQueensNeighborhoodsMeanSalePrice.set(title = 'Queens mean land square feet by neighborhoods')

dfQueensNeighborhoodsMeanSalePrice.set\_xlim(0, 13000)

plt.show()

#brooklyn

dfBrooklyn = df.loc[df["BOROUGH"]=='Brooklyn']

dfBrooklynFirstHalf = dfBrooklyn.head(10899)

dfBrooklynSecondHalf = dfBrooklyn.tail(len(dfBrooklyn) - 10899)

dfBrooklynNeighborhoodsSize = dfBrooklynFirstHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfBrooklynNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsSize)

dfBrooklynNeighborhoodsSize.set(title = 'Numbers of buildings sold by Brooklyn\'s neighborhoods')

dfBrooklynNeighborhoodsSize.set\_xlim(0, 1500)

plt.show()

dfBrooklynNeighborhoodsSize = dfBrooklynSecondHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfBrooklynNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsSize)

dfBrooklynNeighborhoodsSize.set(title = 'Numbers of buildings sold by Brooklyn\'s neighborhoods')

dfBrooklynNeighborhoodsSize.set\_xlim(0, 1500)

plt.show()

dfBrooklynNeighborhoodsSizeByTimePeriod = dfBrooklynFirstHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfBrooklynNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Brooklyn\'s neighborhoods')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfBrooklynNeighborhoodsSizeByTimePeriod.set\_ylim(0, 1500)

plt.show()

dfBrooklynNeighborhoodsSizeByTimePeriod = dfBrooklynSecondHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfBrooklynNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Brooklyn\'s neighborhoods')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfBrooklynNeighborhoodsSizeByTimePeriod.set\_ylim(0, 1500)

plt.show()

dfBrooklynNeighborhoodsTimePeriod = dfBrooklyn.groupby('TIME PERIOD')['NEIGHBORHOOD'].size().reset\_index(name = 'Numbers of buildings sold')

dfBrooklynNeighborhoodsTimePeriod =  sns.barplot(x='Numbers of buildings sold', y= 'TIME PERIOD', data=dfBrooklynNeighborhoodsTimePeriod)

dfBrooklynNeighborhoodsTimePeriod.set(title = 'Brooklyn numbers of buildings sold by time periods')

plt.show()

dfBrooklynNeighborhoodsMaxSalePrice = dfBrooklynFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfBrooklynNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsMaxSalePrice)

dfBrooklynNeighborhoodsMaxSalePrice.set(title = 'Brooklyn max sale price by neighborhoods')

dfBrooklynNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfBrooklynNeighborhoodsMaxSalePrice = dfBrooklynSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfBrooklynNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsMaxSalePrice)

dfBrooklynNeighborhoodsMaxSalePrice.set(title = 'Brooklyn max sale price by neighborhoods')

dfBrooklynNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfBrooklynNeighborhoodsMeanSalePrice = dfBrooklynFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfBrooklynNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsMeanSalePrice)

dfBrooklynNeighborhoodsMeanSalePrice.set(title = 'Brooklyn mean sale price by neighborhoods')

dfBrooklynNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfBrooklynNeighborhoodsMeanSalePrice = dfBrooklynSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfBrooklynNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsMeanSalePrice)

dfBrooklynNeighborhoodsMeanSalePrice.set(title = 'Brooklyn mean sale price by neighborhoods')

dfBrooklynNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfBrooklynSorted = dfBrooklyn.sort\_values(by = ['BUILDING CLASS AT PRESENT'], ascending=False)

dfBrooklynSortedFirstHalf = dfBrooklynSorted.head(6972).reset\_index()

dfBrooklynSortedSecondHalf = dfBrooklynSorted.tail(len(dfBrooklynSorted.index) - 6972).reset\_index()

dfBrooklynNeighborhoodsSizeByTimePeriod = dfBrooklynSortedFirstHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfBrooklynNeighborhoodsSizeByTimePeriod.set(title = 'Brooklyn numbers of buildings sold by building class at time of sale and time period')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.legend(bbox\_to\_anchor=(1.0, 1.0))

dfBrooklynNeighborhoodsSizeByTimePeriod.set\_ylim(0, 3000)

plt.show()

dfBrooklynNeighborhoodsSizeByTimePeriod = dfBrooklynSortedSecondHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfBrooklynNeighborhoodsSizeByTimePeriod.set(title = 'Brooklyn numbers of buildings sold by building class at time of sale and time period')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.legend(bbox\_to\_anchor=(1.0, 1.0))

dfBrooklynNeighborhoodsSizeByTimePeriod.set\_ylim(0, 3000)

plt.show()

indexLandSquareFeetWholeArray = dfBrooklyn[ (dfBrooklyn['LAND SQUARE FEET']=='Land square feet is unknown')].index

indexLandSquareFeet = dfBrooklynFirstHalf[ (dfBrooklynFirstHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfBrooklynFirstHalf.drop(indexLandSquareFeet , inplace=True)

dfBrooklynNeighborhoodsMeanSalePrice = dfBrooklynFirstHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

new\_row = pd.DataFrame({'NEIGHBORHOOD': ['Mean Land Square feet unknown'], 'mean land square feet': [len(indexLandSquareFeetWholeArray)]})

dfBrooklynNeighborhoodsMeanSalePrice = pd.concat([dfBrooklynNeighborhoodsMeanSalePrice, new\_row], ignore\_index=True)

dfBrooklynNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsMeanSalePrice)

dfBrooklynNeighborhoodsMeanSalePrice.set(title = 'Brooklyn mean land square feet by neighborhoods')

dfBrooklynNeighborhoodsMeanSalePrice.set\_xlim(0, 60000)

plt.show()

indexLandSquareFeet = dfBrooklynSecondHalf[ (dfBrooklynSecondHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfBrooklynDropped = dfBrooklynSecondHalf.drop(indexLandSquareFeet , inplace=True)

dfBrooklynNeighborhoodsMeanSalePrice = dfBrooklynSecondHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

dfBrooklynNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfBrooklynNeighborhoodsMeanSalePrice)

dfBrooklynNeighborhoodsMeanSalePrice.set(title = 'Brooklyn mean land square feet by neighborhoods')

dfBrooklynNeighborhoodsMeanSalePrice.set\_xlim(0, 60000)

plt.show()

#Manhattan

dfManhattan = df.loc[df["BOROUGH"]=='Manhattan']

dfManhattanFirstHalf = dfManhattan.head(8001)

dfManhattanSecondHalf = dfManhattan.tail(len(dfManhattan) - 8001)

dfManhattanNeighborhoodsSize = dfManhattanFirstHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfManhattanNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsSize)

dfManhattanNeighborhoodsSize.set(title = 'Numbers of buildings sold by Manhattan\'s neighborhoods')

dfManhattanNeighborhoodsSize.set\_xlim(0, 2000)

plt.show()

dfManhattanNeighborhoodsSize = dfManhattanSecondHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfManhattanNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsSize)

dfManhattanNeighborhoodsSize.set(title = 'Numbers of buildings sold by Manhattan\'s neighborhoods')

dfManhattanNeighborhoodsSize.set\_xlim(0, 2000)

plt.show()

dfManhattanNeighborhoodsSizeByTimePeriod = dfManhattanFirstHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfManhattanNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Manhattan\'s neighborhoods')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.ylabel('Number of buildings sold')

dfManhattanNeighborhoodsSizeByTimePeriod.set\_ylim(0, 2000)

plt.show()

dfManhattanNeighborhoodsSizeByTimePeriod = dfManhattanSecondHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfManhattanNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Manhattan\'s neighborhoods')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

plt.ylabel('Number of buildings sold')

dfManhattanNeighborhoodsSizeByTimePeriod.set\_ylim(0, 2000)

plt.show()

dfManhattanNeighborhoodsTimePeriod = dfManhattan.groupby('TIME PERIOD')['NEIGHBORHOOD'].size().reset\_index(name = 'Numbers of buildings sold')

dfManhattanNeighborhoodsTimePeriod =  sns.barplot(x='Numbers of buildings sold', y= 'TIME PERIOD', data=dfManhattanNeighborhoodsTimePeriod)

dfManhattanNeighborhoodsTimePeriod.set(title = 'Manhattan numbers of buildings sold by time periods')

plt.show()

dfManhattanNeighborhoodsMaxSalePrice = dfManhattanFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfManhattanNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsMaxSalePrice)

dfManhattanNeighborhoodsMaxSalePrice.set(title = 'Manhattan max sale price by neighborhoods')

dfManhattanNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfManhattanNeighborhoodsMaxSalePrice = dfManhattanSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfManhattanNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsMaxSalePrice)

dfManhattanNeighborhoodsMaxSalePrice.set(title = 'Manhattan max sale price by neighborhoods')

dfManhattanNeighborhoodsMaxSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfManhattanNeighborhoodsMeanSalePrice = dfManhattanFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfManhattanNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsMeanSalePrice)

dfManhattanNeighborhoodsMeanSalePrice.set(title = 'Manhattan mean sale price by neighborhoods')

dfManhattanNeighborhoodsMeanSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfManhattanNeighborhoodsMeanSalePrice = dfManhattanSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfManhattanNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsMeanSalePrice)

dfManhattanNeighborhoodsMeanSalePrice.set(title = 'Manhattan mean sale price by neighborhoods')

dfManhattanNeighborhoodsMeanSalePrice.set\_xlim(0, 1000000000)

plt.show()

dfManhattanSorted = dfManhattan.sort\_values(by = ['BUILDING CLASS AT PRESENT'], ascending=False)

dfManhattanSortedFirstHalf = dfManhattanSorted.head(9256).reset\_index()

dfManhattanSortedSecondHalf = dfManhattanSorted.tail(len(dfManhattanSorted.index) - 9256).reset\_index()

dfManhattanNeighborhoodsSizeByTimePeriod = dfManhattanSortedFirstHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfManhattanNeighborhoodsSizeByTimePeriod.set(title = 'Manhattan numbers of buildings sold by building class at time of sale and time period')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfManhattanNeighborhoodsSizeByTimePeriod.set\_ylim(0, 8000)

plt.show()

dfManhattanNeighborhoodsSizeByTimePeriod = dfManhattanSortedSecondHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfManhattanNeighborhoodsSizeByTimePeriod.set(title = 'Manhattan numbers of buildings sold by building class at time of sale and time period')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfManhattanNeighborhoodsSizeByTimePeriod.set\_ylim(0, 8000)

plt.show()

indexLandSquareFeetWholeArray = dfManhattan[ (dfManhattan['LAND SQUARE FEET']=='Land square feet is unknown')].index

indexLandSquareFeet = dfManhattanFirstHalf[ (dfManhattanFirstHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfManhattanFirstHalf.drop(indexLandSquareFeet , inplace=True)

dfManhattanNeighborhoodsMeanSalePrice = dfManhattanFirstHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

new\_row = pd.DataFrame({'NEIGHBORHOOD': ['Mean Land Square feet unknown'], 'mean land square feet': [len(indexLandSquareFeetWholeArray)]})

dfManhattanNeighborhoodsMeanSalePrice = pd.concat([dfManhattanNeighborhoodsMeanSalePrice, new\_row], ignore\_index=True)

dfManhattanNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsMeanSalePrice)

dfManhattanNeighborhoodsMeanSalePrice.set(title = 'Manhattan mean land square feet by neighborhoods')

dfManhattanNeighborhoodsMeanSalePrice.set\_xlim(0, 23000)

plt.show()

indexLandSquareFeet = dfManhattanSecondHalf[ (dfManhattanSecondHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfManhattanDropped = dfManhattanSecondHalf.drop(indexLandSquareFeet , inplace=True)

dfManhattanNeighborhoodsMeanSalePrice = dfManhattanSecondHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

dfManhattanNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfManhattanNeighborhoodsMeanSalePrice)

dfManhattanNeighborhoodsMeanSalePrice.set(title = 'Manhattan mean land square feet by neighborhoods')

dfManhattanNeighborhoodsMeanSalePrice.set\_xlim(0, 23000)

plt.show()

#Staten Island

dfStatenIsland = df.loc[df["BOROUGH"]=='Staten Island']

dfStatenIslandFirstHalf = dfStatenIsland.head(3382)

dfStatenIslandSecondHalf = dfStatenIsland.tail(len(dfStatenIsland) - 3382)

dfStatenIslandNeighborhoodsSize = dfStatenIslandFirstHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfStatenIslandNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsSize)

dfStatenIslandNeighborhoodsSize.set(title = 'Numbers of buildings sold by Staten Island\'s neighborhoods')

dfStatenIslandNeighborhoodsSize.set\_xlim(0, 800)

plt.show()

dfStatenIslandNeighborhoodsSize = dfStatenIslandSecondHalf.groupby('NEIGHBORHOOD').size().reset\_index(name = 'Numbers of buildings sold')

dfStatenIslandNeighborhoodsSize =  sns.barplot(x='Numbers of buildings sold', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsSize)

dfStatenIslandNeighborhoodsSize.set(title = 'Numbers of buildings sold by Staten Island\'s neighborhoods')

dfStatenIslandNeighborhoodsSize.set\_xlim(0, 800)

plt.show()

dfStatenIslandNeighborhoodsSizeByTimePeriod = dfStatenIslandFirstHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfStatenIslandNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Staten Island\'s neighborhoods')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfStatenIslandNeighborhoodsSizeByTimePeriod.set\_ylim(0, 800)

plt.show()

dfStatenIslandNeighborhoodsSizeByTimePeriod = dfStatenIslandSecondHalf.groupby(['NEIGHBORHOOD', 'TIME PERIOD']).size().unstack().plot(kind='bar', stacked=True)

dfStatenIslandNeighborhoodsSizeByTimePeriod.set(title = 'Numbers of buildings sold by Staten Island\'s neighborhoods')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfStatenIslandNeighborhoodsSizeByTimePeriod.set\_ylim(0, 800)

plt.show()

dfStatenIslandNeighborhoodsTimePeriod = dfStatenIsland.groupby('TIME PERIOD')['NEIGHBORHOOD'].size().reset\_index(name = 'Numbers of buildings sold')

dfStatenIslandNeighborhoodsTimePeriod =  sns.barplot(x='Numbers of buildings sold', y= 'TIME PERIOD', data=dfStatenIslandNeighborhoodsTimePeriod)

dfStatenIslandNeighborhoodsTimePeriod.set(title = 'Staten Island numbers of buildings sold by time periods')

plt.show()

dfStatenIslandNeighborhoodsMaxSalePrice = dfStatenIslandFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfStatenIslandNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsMaxSalePrice)

dfStatenIslandNeighborhoodsMaxSalePrice.set(title = 'Staten Island max sale price by neighborhoods')

dfStatenIslandNeighborhoodsMaxSalePrice.set\_xlim(0, 100000000)

plt.show()

dfStatenIslandNeighborhoodsMaxSalePrice = dfStatenIslandSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].max().reset\_index(name = 'Max sale price')

dfStatenIslandNeighborhoodsMaxSalePrice =  sns.barplot(x='Max sale price', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsMaxSalePrice)

dfStatenIslandNeighborhoodsMaxSalePrice.set(title = 'Staten Island max sale price by neighborhoods')

dfStatenIslandNeighborhoodsMaxSalePrice.set\_xlim(0, 100000000)

plt.show()

dfStatenIslandNeighborhoodsMeanSalePrice = dfStatenIslandFirstHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfStatenIslandNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsMeanSalePrice)

dfStatenIslandNeighborhoodsMeanSalePrice.set(title = 'Staten Island mean sale price by neighborhoods')

dfStatenIslandNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfStatenIslandNeighborhoodsMeanSalePrice = dfStatenIslandSecondHalf.groupby('NEIGHBORHOOD')['SALE PRICE'].mean().reset\_index(name = 'mean sale price')

dfStatenIslandNeighborhoodsMeanSalePrice =  sns.barplot(x='mean sale price', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsMeanSalePrice)

dfStatenIslandNeighborhoodsMeanSalePrice.set(title = 'Staten Island mean sale price by neighborhoods')

dfStatenIslandNeighborhoodsMeanSalePrice.set\_xlim(0, 10000000)

plt.show()

dfStatenIslandSorted = dfStatenIsland.sort\_values(by = ['BUILDING CLASS AT PRESENT'], ascending=False)

dfStatenIslandSortedFirstHalf = dfStatenIslandSorted.head(1104).reset\_index()

dfStatenIslandSortedSecondHalf = dfStatenIslandSorted.tail(len(dfStatenIslandSorted.index) - 1104).reset\_index()

dfStatenIslandNeighborhoodsSizeByTimePeriod = dfStatenIslandSortedFirstHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfStatenIslandNeighborhoodsSizeByTimePeriod.set(title = 'Staten Island numbers of buildings sold by building class at time of sale and time period')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfStatenIslandNeighborhoodsSizeByTimePeriod.set\_ylim(0, 2000)

plt.show()

dfStatenIslandNeighborhoodsSizeByTimePeriod = dfStatenIslandSortedSecondHalf.groupby(['BUILDING CLASS AT TIME OF SALE', 'TIME PERIOD'], sort = False).size().unstack().plot(kind='bar', stacked=True)

dfStatenIslandNeighborhoodsSizeByTimePeriod.set(title = 'Staten Island numbers of buildings sold by building class at time of sale and time period')

plt.ylabel('Number of buildings sold')

plt.legend(loc='center left', bbox\_to\_anchor=(1.0, 0.5))

dfStatenIslandNeighborhoodsSizeByTimePeriod.set\_ylim(0, 2000)

plt.show()

indexLandSquareFeetWholeArray = dfStatenIsland[ (dfStatenIsland['LAND SQUARE FEET']=='Land square feet is unknown')].index

indexLandSquareFeet = dfStatenIslandFirstHalf[ (dfStatenIslandFirstHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfStatenIslandFirstHalf.drop(indexLandSquareFeet , inplace=True)

dfStatenIslandNeighborhoodsMeanSalePrice = dfStatenIslandFirstHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

new\_row = pd.DataFrame({'NEIGHBORHOOD': ['Mean Land Square feet unknown'], 'mean land square feet': [len(indexLandSquareFeetWholeArray)]})

dfStatenIslandNeighborhoodsMeanSalePrice = pd.concat([dfStatenIslandNeighborhoodsMeanSalePrice, new\_row], ignore\_index=True)

dfStatenIslandNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsMeanSalePrice)

dfStatenIslandNeighborhoodsMeanSalePrice.set(title = 'Staten Island mean land square feet by neighborhoods')

dfStatenIslandNeighborhoodsMeanSalePrice.set\_xlim(0, 85000)

plt.show()

indexLandSquareFeet = dfStatenIslandSecondHalf[ (dfStatenIslandSecondHalf['LAND SQUARE FEET']=='Land square feet is unknown')].index

dfStatenIslandDropped = dfStatenIslandSecondHalf.drop(indexLandSquareFeet , inplace=True)

dfStatenIslandNeighborhoodsMeanSalePrice = dfStatenIslandSecondHalf.groupby('NEIGHBORHOOD')['LAND SQUARE FEET'].mean().reset\_index(name = 'mean land square feet')

dfStatenIslandNeighborhoodsMeanSalePrice =  sns.barplot(x='mean land square feet', y= 'NEIGHBORHOOD', data=dfStatenIslandNeighborhoodsMeanSalePrice)

dfStatenIslandNeighborhoodsMeanSalePrice.set(title = 'Staten Island mean land square feet by neighborhoods')

dfStatenIslandNeighborhoodsMeanSalePrice.set\_xlim(0, 85000)

plt.show()

**Висновок**

З графіків можемо зробити висновок, що боро Бруклін та Манхеттен є надзвичайно привабливими для інвестицій, так як користуються високим попитом на сьогоднішній день та мають високий показник середньої ціни продажу. Також треба зауважити, що у всіх районах, окрім Стейтен Айленд, клас будівель R4 користувався найбільшим попитом, а клас D4 користувався надзвичайним попитом до 2001 року, але потім впав до мізерних показників продажу. У Стейтен Айленд найбільш популярними класами є A1, A5, B2.