

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

dataset = pd.read_csv('v2_forCSV.csv')
#Reading

dataset['Census Tract']=pd.read_csv('v2_forCSV.csv')['Census Tract']

#removes spelling mistakes, and coverts data types
for i in range(dataset['Front'].size):

    dataset['Front'][i]=round(float(str(dataset['Front'][i]).replace(',',''), 2))
    dataset['Depth'][i]=round(float(str(dataset['Depth'][i]).replace(',',''), 2))
    dataset['Total Value'][i]=round(float(str(dataset['Total Value'][i]).replace(',',''), 2))
    dataset['Sft Price'][i]=round(float(str(dataset['Sft Price'][i]).replace(',',''), 2))
    dataset['Sale Price'][i]=round(float(str(dataset['Sale Price'][i]).replace(',',''), 2))
    dataset['Land Value'][i]=round(float(str(dataset['Land Value'][i]).replace(',',''), 2))
    if(False==(pd.isna(dataset['Year Built'][i]))):
        dataset['Year Built'][i]=round(int(str(dataset['Year Built'][i]).replace(',',''), 2))
    else:
        dataset['Year Built'][i]=-1
    if(False==(pd.isna(dataset['Total Living Area'][i]))):
        dataset['Total Living Area'][i]=round(float(str(dataset['Total Living Area'][i]).replace(',',''), 2))
    else:
        dataset['Total Living Area'][i]=-1

for i in range(dataset['Front'].size):
    if(dataset['Front'][i]>1000):
        print(dataset['Front'][i])
dataset.dtypes

#coverts street values errors
for i in range(dataset['Street'].size):
    if(pd.isna(dataset['Street'][i])):
        dataset['Street'][i]=None

```

```

    print("\n")
    if(pd.isna(dataset['Census Tract'][i])):
        dataset['Census Tract'][i]=None
    else:
        dataset['Census Tract'][i]=round(float(str(dataset['Census Tract'][i])
.replace('UNKNOWN', '-1')), 2)
        dataset['Census Tract'][i]=round(float(str(dataset['Census Tract'][i])
.replace(',', '')), 2)

#Converts datatypes
dataset['Total Living Area'] = dataset['Total Living Area'].astype('float64')
dataset['Deed Date'] = dataset['Deed Date'].astype('datetime64[ns]')
dataset['Council District'] = dataset['Council District'].astype('string')
dataset['Police District'] = dataset['Police District'].astype('string')
dataset['Neighborhood'] = dataset['Neighborhood'].astype('string')
dataset['Print Key'] = dataset['Print Key'].astype('string')
dataset['Street'] = dataset['Street'].astype('string')
dataset['Front'] = dataset['Front'].astype('float64')
dataset['Land Value'] = dataset['Land Value'].astype('float64')
dataset['Total Value'] = dataset['Total Value'].astype('float64')
dataset['Sale Price'] = dataset['Sale Price'].astype('float64')
dataset['Year Built'] = dataset['Year Built'].astype('int64')
dataset['Census Tract'] = dataset['Census Tract'].astype('float64')
dataset.dtypes

#String Upper case
for i in range(dataset['Neighborhood'].size):
    dataset['Neighborhood'][i]=str(dataset['Neighborhood'][i]).upper()
    dataset['Police District'][i]=str(dataset['Police District'][i]).upper()
    dataset['Council District'][i]=str(dataset['Council District'][i]).upper()
    dataset['Street'][i]=str(dataset['Street'][i]).upper()

#remove string dupliactes b
for i in range(dataset['Neighborhood'].size):
    dataset['Neighborhood'][i]=str(dataset['Neighborhood'][i]).replace(' ', '')
    dataset['Police District'][i]=str(dataset['Police District'][i]).upper()
    dataset['Counsil District'][i]=str(dataset['Counsil District'][i]).upper()
    dataset['Street'][i]=str(dataset['Street'][i]).upper()

```

```

dataset['Police District']

# Describes data and does some basic math operations.
Dataset.info
dataset.describe().transpose()

#Sft prices using seaborn
import seaborn as sns

fig = plt.figure(figsize=(10,7))
fig.add_subplot(2,1,1)
sns.distplot(dataset['Sft Price'])
fig.add_subplot(2,1,2)
sns.boxplot(dataset['Sft Price'])
plt.tight_layout()

#visualizing more on year, zipcode, area, class of property(refer to
dataset original documentaion)
dataset['Deed Date'] = dataset['Deed Date'].astype('datetime64[ns]')
years=[]
sftprice=[]
zipcodes=[]
for i in range (dataset['Deed Date'].size):
    years.append(dataset['Deed Date'][i].year)
for i in range (dataset['Sft Price'].size):
    if(dataset['Sft Price'][i]>5):
        sftprice.append(dataset['Sft Price'][i])
        zipcodes.append(dataset['Zipcode'][i])
fig = plt.figure(figsize=(40,20))
fig.add_subplot(2,2,1)
sns.scatterplot(years,dataset['Sft Price'])
fig.add_subplot(2,2,2)
sns.scatterplot(zipcodes,sftprice)
fig.add_subplot(2,2,3)
sns.scatterplot(dataset['Front']*dataset['Depth'],dataset['Sale Price'])
fig.add_subplot(2,2,4)
sns.scatterplot(dataset['Property Class'],dataset['Sft Price'])
x=[]
y=[]
#plot for year built
for i in range(dataset['Year Built'].size):
    if(dataset['Year Built'][i]>2000):
        x.append(dataset['Year Built'][i])
        y.append(dataset['Sft Price'][i])

```

```

fig.add_subplot(2,2,1)
sns.scatterplot(x,y)

#visualizing floors,Conditon,bedrooms
fig = plt.figure(figsize=(15,7))
fig.add_subplot(2,2,1)
sns.countplot(dataset['Overall Condition'])
fig.add_subplot(2,2,2)
sns.countplot(dataset['# of Beds'])
fig.add_subplot(2,2,3)
sns.countplot(dataset['# of Baths'])
plt.tight_layout()

dataset['date'] = pd.to_datetime(dataset['Deed Date'])
dataset['month'] = dataset['date'].apply(lambda date:date.month)
dataset['year'] = dataset['date'].apply(lambda date:date.year)

#data visualization sft price vs months, years, zipcode, overall conditon
fig = plt.figure(figsize=(16,5))
fig.add_subplot(1,2,1)
dataset.groupby('month').mean()['Sft Price'].plot()
fig.add_subplot(1,2,2)
dataset.groupby('year').mean()['Sft Price'].plot()
fig = plt.figure(figsize=(16,5))
fig.add_subplot(2,2,1)
dataset.groupby('Zipcode').mean()['Sft Price'].plot()
fig.add_subplot(2,2,2)
dataset.groupby('Overall Condition').mean()['Sft Price'].plot()

#plot for year vs sft
dataset[dataset['Deed Date'].dt.year>2016].groupby(dataset['Deed Date'].dt
.year).mean()['Sft Price'].plot()
fig = plt.figure(figsize=(16,5))

```