

# Assignment\_6

## Python for Financial Analysis

---

Salomons Commercial is a full service commercial real estate brokerage and property management firm



<http://www.salomonscommercial.com/properties>

# Class to analyze Sold/Leased/Active Single User Industrial Buildings

```
import pandas as pd
import seaborn as sns

# define a class to call object to analyze specific data for stats on single user
# industrial properties in Red Deer, Alberta, Canada
# that have sold/leased or are currently active for sale/lease

class SingleUserIndustrial():

    def __init__(self, excelsheet: str):
        self._fn = f'{excelsheet}.xlsx'
        print(f'Trying to read from {excelsheet}\n')
        self._df = None

    def read_from_xlsx(self):
        self._df = (pd.read_excel(self._fn, sheet_name='Industrial Building Sales')
                    .drop(['Address', 'Municipality', 'Zoning', 'Lat', 'Long'], axis=1)
                    .set_index("Type"))

        self._df = self._df.fillna(0).astype({'Footprint SF': 'int64', 'Asking Price/Price Sold': 'int64'})
        print(f'{self._fn} has {len(self._df)} records.')
        display(self._df.head())
```

# Page 2

```
def leased(self):
    leased = self._df.loc['Leased'].copy()
    leased.drop(['Asking Price/Price Sold', 'Price PSF'], axis=1, inplace=True)
    leased.rename(columns={"Date Sold/Leased": "Date Leased"}, inplace=True)
    print(f'Leased Type has {len(leased)} records')
    display(leased.head())
    return leased

def sold(self):
    sold = self._df.loc['Sold'].copy()
    sold.drop(['Lease Rate PSF'], axis=1, inplace=True)
    sold.rename(columns={"Asking Price/Price Sold": "Price Sold", "Date Sold/Leased": "Date Sold"}, inplace=True)
    print(f'Sold Type has {len(sold)} records')
    display(sold.head())
    return sold

def active(self):
    active = self._df.loc['Active'].copy()
    active.drop(['Date Sold/Leased'], axis=1, inplace=True)
    active.rename(columns={"Asking Price/Price Sold": "Asking Price"}, inplace=True)
    print(f'Active Type has {len(active)} records')
    display(active.head())
    return active
```

# Page 3

```
def leased_avg(self):
    return self.leased()['Lease Rate PSF'].mean()

def sold_avg(self):
    return self.sold()['Price PSF'].mean()

def active_area(self):
    return self.active()['Building SF'].sum()

def sns_leased(self):
    sns_leased = self.leased().set_index('Date Leased')
    sns.set_style("white")
    sns.scatterplot(data=sns_leased, x="Date Leased", y="Lease Rate PSF", palette='dark')

def sns_sold(self):
    sns_sold = self.sold().set_index('Date Sold')
    sns.set_style("white")
    sns.scatterplot(data=sns_sold, x="Date Sold", y="Price PSF", palette='dark')

def sns_total(self):
    sns_total = self._df.reset_index()
    sns.set_style("white")
    sns.relplot(data=sns_total, x="Parcel Acres", y="Building SF", hue="Type", palette='dark')
```

# Page 4

```
si = SingleUserIndustrial('2022-03 Single User Sale Lease Comps')
si.read_from_xlsx()
print(f'The Average Lease Rate is ${si.leased_avg():,.2f} PSF\n')
print(f'The Average Sold Price is ${si.sold_avg():,.2f} PSF\n')
print(f'There is currently {si.active_area():,.0f} SF available for sale or lease\n')

print(si.sns_sold())
si.sns_total()
```

# Analysis

- Attached PDF shows the output from the previous function calls
- Shows how the values of single user industrial buildings have sold and leased
- As well as the total building area currently available on the market
- Link to view properties on Google Maps:
- <https://batchgeo.com/map/5ea5c8e21f2f22d73571f0cd2b1d951a>