Imports

In [34]:
```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, lit, when, isnan, count, mean, stddev, corr, expr
from pyspark.sql.functions import col, sum as _sum, when,month,to_date,corr,year, avg
```

Intializing Spark Session

In [36]:
```
spark = SparkSession.builder.appName("RealEstate").getOrCreate()
```

In [38]:
```
df = spark.read.csv("Real_Estate_Sales_2001-2022_GL.csv",header=True,inferSchema=True)
```

Data Exploration

In [42]:
```
df.printSchema()
```

```
root
 |-- Serial Number: integer (nullable = true)
 |-- List Year: integer (nullable = true)
 |-- Date Recorded: string (nullable = true)
 |-- Town: string (nullable = true)
 |-- Address: string (nullable = true)
 |-- Assessed Value: double (nullable = true)
 |-- Sale Amount: double (nullable = true)
 |-- Sales Ratio: double (nullable = true)
 |-- Property Type: string (nullable = true)
 |-- Residential Type: string (nullable = true)
 |-- Non Use Code: string (nullable = true)
 |-- Assessor Remarks: string (nullable = true)
 |-- OPM remarks: string (nullable = true)
 |-- Location: string (nullable = true)
```

In [44]:
```
df.count()
```

Out[44]:
```
1097629
```

In [48]:
```
df.dtypes
```

Out[48]:
```
[('Serial Number', 'int'),
 ('List Year', 'int'),
 ('Date Recorded', 'string'),
 ('Town', 'string'),
 ('Address', 'string'),
 ('Assessed Value', 'double'),
 ('Sale Amount', 'double'),
 ('Sales Ratio', 'double'),
 ('Property Type', 'string'),
 ('Residential Type', 'string'),
 ('Non Use Code', 'string'),
 ('Assessor Remarks', 'string'),
 ('OPM remarks', 'string'),
 ('Location', 'string')]
```

In [50]:
```
df.head()
```

Out[50]:
```
Row(Serial Number=2020177, List Year=2020, Date Recorded='04/14/2021', Town='Ansonia', Address='323 BEAVER ST',
Assessed Value=133000.0, Sale Amount=248400.0, Sales Ratio=0.5354, Property Type='Residential', Residential Typ
e='Single Family', Non Use Code=None, Assessor Remarks=None, OPM remarks=None, Location='POINT (-73.06822 41.35
014)')
```

Data Cleansing

In [52]:
```
# drop rows that have column date Recoreded is null
clean_df=df.na.drop(subset=['Date Recorded'])
```

In [54]:
```
# drop rows that have column Address is null
clean_df=df.na.drop(subset=['Address'])
```

In [56]:
```
# drop the single row that has neither assessed value nor sale amount nor sales ratio
```

```
clean_df = clean_df.na.drop(subset=["Assessed Value","Sale Amount"])
```

In [64]:
```
# fill Column Property Type null values with "Undefined"
clean_df= clean_df.na.fill({'Property Type':'Undefined'})
```

In [62]:
```
# fill Column Residential Type  null values with "Undefined"
clean_df= clean_df.na.fill({'Residential Type':'Undefined'})
```

In [66]:
```
# fill Column Non Use Code null values with "Undefined"
clean_df= clean_df.na.fill({'Non Use Code':'Undefined'})
```

In [68]:
```
# fill Column Assessor Remarks null values with "Undefined"
clean_df= clean_df.na.fill({'Assessor Remarks':'None'})
```

In [70]:
```
# fill Column OPM remarks null values with "Undefined"
clean_df= clean_df.na.fill({'OPM remarks':'None'})
```

In [72]:
```
# fill Column Location null values with "Undefined"
clean_df= clean_df.na.fill({'Location':'Unknown'})
```

In [74]:
```
#checking if there are any null values left
clean_df.select([
    _sum(when(col(c).isNull(), 1).otherwise(0)).alias(c + "_nulls")
    for c in df.columns
]).show()
```

```
+------------------+--------------+------------------+----------+-------------+-------------------+---------
--------+----------------+------------------+---------------------+------------------+---------------------+
-----------------+-------------+
|Serial Number_nulls|List Year_nulls|Date Recorded_nulls|Town_nulls|Address_nulls|Assessed Value_nulls|Sale Amou
nt_nulls|Sales Ratio_nulls|Property Type_nulls|Residential Type_nulls|Non Use Code_nulls|Assessor Remarks_nulls|
OPM remarks_nulls|Location_nulls|
+------------------+--------------+------------------+----------+-------------+-------------------+---------
--------+----------------+------------------+---------------------+------------------+---------------------+
-----------------+-------------+
|                  0|              0|                  0|         0|            0|                   0|
0|                0|                  0|                     0|                 0|                     0|
0|            0|
+------------------+--------------+------------------+----------+-------------+-------------------+---------
--------+----------------+------------------+---------------------+------------------+---------------------+
-----------------+-------------+
```

In [76]:
```
#dropping duplicated rows
clean_df=clean_df.dropDuplicates()
```

In [78]:
```
#delete sales amount values = 0
clean_df=clean_df.filter(col("Sale Amount") != 0.0)
```

In [84]:
```
#drop incorrect sale price
clean_df = clean_df.filter(clean_df["OPM remarks"] != "INCORRECT SALE PRICE")
```

Questions

1. What are the highest and lowest sale amounts recorded, and which properties correspond to these sales?

In [86]:
```
# Lowest:
lowest_sale = clean_df.orderBy(col("Sale Amount").asc()).limit(1)
lowest_sale.show()

#Highest
highest_sale = clean_df.orderBy(col("Sale Amount").desc()).limit(1)
highest_sale.show()
```

```
+-------------+---------+------------+-------+--------------+-------------+-----------+-----------+----------
---+---------------+------------+----------------+-----------+--------------------+
|Serial Number|List Year|Date Recorded|   Town|        Address|Assessed Value|Sale Amount|Sales Ratio|Property T
ype|Residential Type|Non Use Code|Assessor Remarks|OPM remarks|            Location|
+-------------+---------+------------+-------+--------------+-------------+-----------+-----------+----------
---+---------------+------------+----------------+-----------+--------------------+
|        31255|     2003|   07/22/2004|Bristol|66 EMMETT ST 12|        9520.0|        1.0|     9520.0|    Undefi
ned|       Undefined|           3|            None|       None|POINT (-72.91363 ...|
+-------------+---------+------------+-------+--------------+-------------+-----------+-----------+----------
---+---------------+------------+----------------+-----------+--------------------+


+-------------+---------+------------+--------+--------------------+--------------+-----------+-----------+----
---------+---------------+------------+--------------------+-----------+--------+
|Serial Number|List Year|Date Recorded|    Town|             Address|Assessed Value|Sale Amount|Sales Ratio|Prop
erty Type|Residential Type|Non Use Code|    Assessor Remarks|OPM remarks|Location|
+-------------+---------+------------+--------+--------------------+--------------+-----------+-----------+----
---------+---------------+------------+--------------------+-----------+--------+
|       160391|     2016|   12/06/2016|Stamford|200 HENRY STREET # 5|     3307410.0|    3.955E8|0.008362604|
Undefined|       Undefined|  25 - Other|ENTERPRISE ZONE &...|       None| Unknown|
+-------------+---------+------------+--------+--------------------+--------------+-----------+-----------+----
---------+---------------+------------+--------------------+-----------+--------+
```

- **Lowest: $1**
- **Highest: $395M**

2: What is the average sale price per town and per list year?

In [87]:
```
clean_df.groupBy("Town", "List Year").agg({"Sale Amount": "avg"}).show()
```

```
+------------+---------+-----------------+
|        Town|List Year| avg(Sale Amount)|
+------------+---------+-----------------+
|Old Saybrook|     2021| 722761.7075098815|
|    Brooklyn|     2002| 146965.0909090909|
|      Goshen|     2021| 489993.0714285714|
|     Hampton|     2002|195578.57142857142|
|      Hamden|     2010|238599.57248157248|
|     Sherman|     2004|487318.57843137253|
|    Eastford|     2006|215854.54545454544|
|  Colchester|     2010|240234.11282051282|
| Marlborough|     2010|311928.62903225806|
| Bridgewater|     2008|          465852.4|
|     Windsor|     2010|217862.45307443367|
|     Chaplin|     2011|163904.42105263157|
|     Andover|     2013| 208191.4054054054|
|   Killingly|     2015| 148198.4520884521|
|   Newington|     2013| 254211.3947368421|
|  Canterbury|     2016|171004.09473684212|
|  Stonington|     2016|407765.03855421685|
|     Milford|     2019|463499.32117134565|
|  Rocky Hill|     2019|369088.49159663863|
|   Colebrook|     2022|          312825.0|
+------------+---------+-----------------+
only showing top 20 rows
```

- Town & year show wide price spread
- Old Saybrook (2021): **$722K** tops
- Brooklyn (2002): **$147K** near bottom

3. Are there any seasonal trends in sales prices or volumes based on the `Date Recorded`?

In [93]:
```
clean_df= clean_df.withColumn("Date Recorded", to_date("Date Recorded", "M/d/yyyy"))
clean_df.withColumn("Month", month("Date Recorded")) \
  .groupBy("Month") \
  .agg({"Sale Amount": "avg", "*": "count"}) \
  .show()
```

```
+-----+-----------------+--------+
|Month|  avg(Sale Amount)|count(1)|
+-----+-----------------+--------+
|   12| 464528.2116375185|   91315|
|    1| 416265.9668819175|   69841|
|    6|419820.59928444953|  112948|
|    3| 371954.4099401002|   79466|
|    5| 384871.6408424889|   96298|
|    9|  384995.265380036|   94741|
|    4|385795.37078191026|   83539|
|    8|417736.80124969705|  111387|
|    7|424230.67889088974|  111621|
|   10| 375143.5129536248|   96776|
|   11|  376870.861590956|   87174|
|    2| 369153.5392474376|   60686|
+-----+-----------------+--------+
```

- Dec peaks: avg sale ~$465K, highest volume (91K+)
- Early year dips: Feb/Mar lowest avg & count
- Summer steady sales, mid $410K avg

4. Which towns have the highest sales volumes and total sales values?

In [97]:
```
clean_df.groupBy("Town") \
  .agg({"Sale Amount": "sum", "*": "count"}) \
  .orderBy("sum(Sale Amount)", ascending=False) \
  .show()
```

```
+-------------+------------------+--------+
|         Town|  sum(Sale Amount)|count(1)|
+-------------+------------------+--------+
|    Greenwich|  3.866514031028E10|   18336|
|     Stamford|  3.309213793942E10|   36623|
|      Norwalk|  1.677263746667E10|   26926|
|     Westport|  1.533676587385E10|   10262|
|    Fairfield|  1.325577345632E10|   17913|
|       Darien|    1.1670422931E10|    7287|
|   New Canaan|    1.0941894536E10|    7071|
|      Danbury|    8.87083532432E9|   21884|
|West Hartford|    8.26555080984E9|   22096|
|   Bridgeport|      8.028362582E9|   38122|
|    New Haven|      7.106383588E9|   23696|
|      Milford|    6.52868400562E9|   18569|
|   Ridgefield|      6.440289611E9|    8596|
|    Waterbury|      5.946213551E9|   32652|
|    Stratford|      5.646381156E9|   18504|
|       Wilton|5.528056068620001E9|    6008|
|     Hartford|      5.330687452E9|   19593|
|       Hamden|      5.265019427E9|   18499|
|  Glastonbury|      5.193040475E9|   12512|
|      Shelton|    4.89720446036E9|   12442|
+-------------+------------------+--------+
only showing top 20 rows
```

- Greenwich leads: $38.7B total, 18K+ sales
- Stamford close: $33.1B, 36K+ sales
- Norwalk & Westport also strong in value & count
- Big cities dominate both volume & total sales

5. How does the sale amount correlate with the assessed value across different towns and property types?

In [98]:
```
clean_df.groupBy("Town", "Property Type") \
  .agg(corr("Sale Amount", "Assessed Value").alias("correlation")) \
  .show()
```

Town	Property Type	correlation
Meriden	Residential	0.22430143838731906
Old Saybrook	Undefined	0.680084384246063
Guilford	Vacant Land	0.9270274195322484
Lisbon	Vacant Land	0.8051182179351708
Salem	Vacant Land	-0.12196273821951283
Bridgeport	Vacant Land	0.9631727756450098
Bozrah	Industrial	0.8379529657885672
Bristol	Industrial	0.31240776790964764
East Lyme	Condo	0.07800033286234526
Ellington	Four Family	1.0
Fairfield	Three Family	0.15949941727762226
Bloomfield	Two Family	0.695558911280118
Manchester	Residential	0.8170254878324427
West Haven	Residential	0.16871428262274815
West Haven	Undefined	0.42503649293302603
Rocky Hill	Undefined	0.8499447302080924
Norfolk	Undefined	0.2615991500270722
Farmington	Vacant Land	-0.1522186732255132
Granby	Commercial	0.9361892140918268
Wallingford	Apartments	0.9853623769376771

only showing top 20 rows

- Strongest: Apartments (Wallingford, 0.99), Vacant Land (Bridgeport, 0.96)
- Moderate: Residential varies (e.g., Meriden 0.22, Manchester 0.82)
- Some negative/weak correlations in Vacant Land & others
- Correlation depends heavily on property type and town

6. What are the average sale prices for different property types and residential types?

In [102…

```
clean_df.groupBy("Property Type", "Residential Type") \
  .agg({"Sale Amount": "avg"}) \
  .show()
```

Property Type	Residential Type	avg(Sale Amount)
Apartments	Undefined	2947209.3846153845
Condo	Condo	260242.43557711283
Residential	Four Family	383789.82076637825
Vacant Land	Undefined	417032.7176342025
Residential	Two Family	301357.3277251185
Commercial	Undefined	1677886.2614080505
Three Family	Three Family	179844.5162084856
Undefined	Undefined	415267.50899534574
Two Family	Two Family	199096.43897007575
Residential	Condo	361254.1691745749
Residential	Three Family	308016.99263316585
Four Family	Four Family	314437.28571428574
Residential	Single Family	521401.878611132
Industrial	Undefined	2238044.5635220124
Public Utility	Undefined	213604.4
Single Family	Single Family	388601.55033495213

- Apartments: $2.95M
- Industrial: $2.24M
- Commercial: $1.68M
- Residential (Single Family): $521K
- Residential (Four Family): $384K
- Vacant Land: $417K
- Undefined types hover ~$415K
- Lower averages in Three/Two Family & Condo (~$180K–$ 360K)

7. Which residential types demonstrate the strongest price appreciation trends over the years?

In [103…
```
clean_df = clean_df.withColumn("Year", year("Date Recorded"))

clean_df.groupBy("Residential Type", "Year") \
  .agg(avg("Sale Amount").alias("avg_sale")) \
  .orderBy("Residential Type", "Year") \
  .show()
```

```
+----------------+----+------------------+
|Residential Type|Year|          avg_sale|
+----------------+----+------------------+
|           Condo|1999|           95000.0|
|           Condo|2001|           88000.0|
|           Condo|2004|210966.66666666666|
|           Condo|2005|          138500.0|
|           Condo|2006|244697.18428630193|
|           Condo|2007|268362.33584320673|
|           Condo|2008|253857.84299790018|
|           Condo|2009|225202.00719952982|
|           Condo|2010| 233047.9052268811|
|           Condo|2011|220901.54331838564|
|           Condo|2012|232790.09898135692|
|           Condo|2013| 230723.6143569934|
|           Condo|2014|241354.85729715135|
|           Condo|2015| 231117.6593537618|
|           Condo|2016|459960.51612141257|
|           Condo|2017|246734.70676512626|
|           Condo|2018|250676.81313769164|
|           Condo|2019|263293.79590298503|
|           Condo|2020|247167.04774604697|
|           Condo|2021|425779.66806168837|
+----------------+----+------------------+
only showing top 20 rows
```

Condo:

- 1999 avg: $95K
- Steady rise with fluctuations
- Big jumps: 2016 (460K), 2021(426K)
- Clear upward trend over years

8. How do sales ratios (Sale Amount / Assessed Value) vary by property and residential type?

In [108…
```
clean_df.withColumn("Sales Ratio Calc", col("Sale Amount") / col("Assessed Value")) \
  .groupBy("Property Type", "Residential Type") \
  .agg({"Sales Ratio Calc": "avg"}) \
  .show()
```

```
+--------------+----------------+---------------------+
| Property Type|Residential Type|avg(Sales Ratio Calc)|
+--------------+----------------+---------------------+
|    Apartments|       Undefined|   2.2383061611213835|
|         Condo|           Condo|   3.0544718302982385|
|   Residential|     Four Family|    2.192101402173227|
|   Vacant Land|       Undefined|    39.84027360278055|
|   Residential|      Two Family|   2.0940316488663866|
|    Commercial|       Undefined|   373.93571677012767|
|  Three Family|    Three Family|    1.488899046781721|
|     Undefined|       Undefined|   18.358014466689838|
|    Two Family|      Two Family|   1.3785153567566686|
|   Residential|           Condo|   55.806640425738394|
|   Residential|    Three Family|    2.473646925891281|
|   Four Family|     Four Family|   1.5297006036964562|
|   Residential|   Single Family|   3.7573824059894414|
|    Industrial|       Undefined|    2.691371008295352|
|Public Utility|       Undefined|    5.206196911118471|
| Single Family|   Single Family|   1.9136920614535577|
+--------------+----------------+---------------------+
```

Sales ratios vary by property and residential type:

- Highest ratios:
  - Commercial (Undefined): **373.94**
  - Vacant Land (Undefined): **39.84**

        - Residential - Condo: **55.81**
    - Moderate ratios:
        - Residential - Single Family: **3.76**
        - Condo - Condo: **3.05**
        - Public Utility: **5.21**
        - Industrial: **2.69**
    - Lower ratios (around 1.3 to 2.5):
        - Apartments, Residential (various family types), Two/Three/Four Family, Single Family

9. How have average sale prices changed over the list years?

In [115…

```
clean_df.groupBy("List Year") \
  .agg({"Sale Amount": "avg"}) \
  .orderBy("List Year") \
  .show()
```

```
+---------+-----------------+
|List Year|  avg(Sale Amount)|
+---------+-----------------+
|     2001|248341.23252310505|
|     2002|297613.76168601715|
|     2003|328180.04641758656|
|     2004| 382815.4448291146|
|     2005|365005.27087911195|
|     2006|  475554.282661034|
|     2007|435750.08272027853|
|     2008|325831.08462149446|
|     2009|  355250.327161946|
|     2010|331657.47257472156|
|     2011| 391684.3207468212|
|     2012| 395708.6793285492|
|     2013| 413516.2396414891|
|     2014| 401421.9412196598|
|     2015|345883.76394932583|
|     2016|507761.24927169346|
|     2017| 393755.8584190226|
|     2018| 383992.7016063464|
|     2019|420296.97130813857|
|     2020| 526980.1025382191|
+---------+-----------------+
only showing top 20 rows
```

- Early 2000s: Steady rise from **$248,341** (2001) to around **$475,554** (2006).
- 2007-2010: Drop and fluctuation, lowest around **$325,831** (2008).
- 2011-2015: Moderate recovery, averaging between **$345,884** and **$413,516**.
- 2016 and 2020 show peaks at **$507,761** and **$526,980**, the highest values in the period.
- Some years show declines, e.g., 2017 and 2018 near $383,993 -$ **393,756**.

10. Are there any notable spikes or dips in sale prices or volumes during specific years or market events?

In [116…

```
clean_df.groupBy("List Year") \
  .agg({"Sale Amount": "avg", "*": "count"}) \
  .orderBy("List Year") \
  .show()
```

```
+---------+-----------------+--------+
|List Year|  avg(Sale Amount)|count(1)|
+---------+-----------------+--------+
|     2001|248341.23252310505|   59078|
|     2002|297613.76168601715|   59430|
|     2003|328180.04641758656|   64049|
|     2004| 382815.4448291146|   83477|
|     2005|365005.27087911195|   61437|
|     2006|  475554.282661034|   48763|
|     2007|435750.08272027853|   35614|
|     2008|325831.08462149446|   32734|
|     2009|  355250.327161946|   42508|
|     2010|331657.47257472156|   33491|
|     2011| 391684.3207468212|   31065|
|     2012| 395708.6793285492|   35952|
|     2013| 413516.2396414891|   39943|
|     2014| 401421.9412196598|   49563|
|     2015|345883.76394932583|   46651|
|     2016|507761.24927169346|   49773|
|     2017| 393755.8584190226|   45630|
|     2018| 383992.7016063464|   50674|
|     2019|420296.97130813857|   58954|
|     2020| 526980.1025382191|   66590|
+---------+-----------------+--------+
only showing top 20 rows
```

- **Price spikes:**
  - 2006: Average price jumps to **$475,554** from $365,005 in 2005, with volume dropping to 48,763 from 61,437.
  - 2016: Another spike to **$507,761** with volume steady at 49,773.
  - 2020: Highest average price **$526,980** with volume increasing to 66,590.
- **Price dips:**
  - 2008: Average price drops sharply to **$325,831** during the financial crisis; volume decreases to 32,734.
  - 2010-2011: Prices remain relatively low ($331k -$ 391k) with lower volumes (31k-33k).
- **Volume trends:**
  - Peak volume in 2004 at 83,477 sales, then steady decline till 2008.
  - Volume increases again from 2014 onwards, reaching 66,590 in 2020.