In [97]:
```
import pandas as pd
import numpy as np

df = pd.read_csv("sample_csv.csv")
```

In [98]:
```
df.head(1000)
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance
0	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5
1	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5
2	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5
3	Abbotsford	40 Federation La	3	h	850000.0	PI	Biggin	4/03/2017	2.5
4	Abbotsford	55a Park St	4	h	1600000.0	VB	Nelson	4/06/2016	2.5
...	...	...	...	...	...	...	...	...	...
995	Box Hill	6 Archibald	3	h	4000000.0	VB	Lindellas	18/06/2016	13.1

In [99]:
```
df.isnull()
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...
0	False	False	False	False	False	False	False	False	False	False	...
1	False	False	False	False	False	False	False	False	False	False	...
2	False	False	False	False	False	False	False	False	False	False	...
3	False	False	False	False	False	False	False	False	False	False	...
4	False	False	False	False	False	False	False	False	False	False	...
...	...	...	...	...	...	...	...	...	...	...	...
13575	False	False	False	False	False	False	False	False	False	False	...
13576	False	False	False	False	False	False	False	False	False	False	...
13577	False	False	False	False	False	False	False	False	False	False	...
13578	False	False	False	False	False	False	False	False	False	False	...
13579	False	False	False	False	False	False	False	False	False	False	...

13580 rows × 21 columns

In [100]:
```
df.isnull().sum()
```

```
Suburb             0
Address            0
Rooms              0
Type               0
Price              0
Method             0
SellerG            0
Date               0
Distance           0
Postcode           0
Bedroom2           0
Bathroom           0
Car               62
Landsize           0
BuildingArea    6450
YearBuilt       5375
CouncilArea     1369
Lattitude          0
Longtitude         0
Regionname         0
Propertycount      0
dtype: int64
```

In [101]:
```
df.describe()
```

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	
count	13580.000000	1.358000e+04	13580.000000	13580.000000	13580.000000	13580.000000	135
mean	2.937997	1.075684e+06	10.137776	3105.301915	2.914728	1.534242	1.6
std	0.955748	6.393107e+05	5.868725	90.676964	0.965921	0.691712	0.9
min	1.000000	8.500000e+04	0.000000	3000.000000	0.000000	0.000000	0.0
25%	2.000000	6.500000e+05	6.100000	3044.000000	2.000000	1.000000	1.0
50%	3.000000	9.030000e+05	9.200000	3084.000000	3.000000	1.000000	2.0
75%	3.000000	1.330000e+06	13.000000	3148.000000	3.000000	2.000000	2.0
max	10.000000	9.000000e+06	48.100000	3977.000000	20.000000	8.000000	10.

In [102]:
```
df.dtypes
```

```
Suburb           object
Address          object
Rooms             int64
Type             object
Price           float64
Method           object
SellerG          object
Date             object
Distance        float64
Postcode        float64
Bedroom2        float64
Bathroom        float64
Car             float64
Landsize        float64
BuildingArea    float64
YearBuilt       float64
CouncilArea      object
Lattitude       float64
Longtitude      float64
Regionname       object
Propertycount   float64
dtype: object
```

In [103]:
```
df.shape
```

```
(13580, 21)
```

In [104]:
```
df.Type.value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0        h
1        h
2        h
3        h
4        h
        ..
13575    h
13576    h
13577    h
13578    h
13579    h
Name: Type, Length: 13580, dtype: object>
```

In [106]:
```
df['Regionname'].unique()
```

```
array(['Northern Metropolitan', 'Western Metropolitan',
       'Southern Metropolitan', 'Eastern Metropolitan',
       'South-Eastern Metropolitan', 'Eastern Victoria',
       'Northern Victoria', 'Western Victoria'], dtype=object)
```

In [107]:
```
df['Regionname'] = df['Regionname'].map({'Northern Metropolitan': 1, 'West
df.head(1000)
```

ype	Price	Method	SellerG	Date	Distance	Postcode	...	Bathroom	Car	Landsize
	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	...	1.0	1.0	202.0
	1035000.0	S	Biggin	4/02/2016	2.5	3067.0	...	1.0	0.0	156.0
	1465000.0	SP	Biggin	4/03/2017	2.5	3067.0	...	2.0	0.0	134.0
	850000.0	PI	Biggin	4/03/2017	2.5	3067.0	...	2.0	1.0	94.0
	1600000.0	VB	Nelson	4/06/2016	2.5	3067.0	...	1.0	2.0	120.0
.	...	...	...	...	...	...	...	...	...	...
	4000000.0	VB	Lindellas	18/06/2016	13.1	3128.0	...	1.0	2.0	763.0
	928000.0	S	Lindellas	18/06/2016	13.1	3128.0	...	1.0	1.0	307.0
	761000.0	S	Marshall	18/06/2016	13.1	3128.0	...	2.0	1.0	176.0
	636000.0	S	Philip	19/11/2016	13.1	3128.0	...	1.0	1.0	151.0
	1625000.0	S	Noel	19/11/2016	13.1	3128.0	...	2.0	2.0	620.0

In [108]:
```
df.dtypes
```

```
Suburb           object
Address          object
Rooms             int64
Type             object
Price           float64
Method           object
SellerG          object
Date             object
Distance        float64
Postcode        float64
Bedroom2        float64
Bathroom        float64
Car             float64
Landsize        float64
BuildingArea    float64
YearBuilt       float64
CouncilArea      object
Lattitude       float64
Longtitude      float64
Regionname        int64
Propertycount   float64
dtype: object
```

In [109]:
```
df['YearBuilt']
```

```
0           NaN
1        1900.0
2        1900.0
3           NaN
4        2014.0
          ...
13575    1981.0
13576    1995.0
13577    1997.0
13578    1920.0
13579    1920.0
Name: YearBuilt, Length: 13580, dtype: float64
```

In [110]:
```
df['YearBuilt'] = df['YearBuilt'].astype(float).astype("Int64")
```

In [111]:
```
df['YearBuilt']
```

```
0        <NA>
1        1900
2        1900
3        <NA>
4        2014
         ...
13575    1981
13576    1995
13577    1997
13578    1920
13579    1920
Name: YearBuilt, Length: 13580, dtype: Int64
```