New Section

```
In [ ]:
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
In [ ]:
auth.authenticate user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get application default()
drive = GoogleDrive(gauth)
In [ ]:
downloaded = drive.CreateFile({'id':'1HwC6NpSnHYvdhL7tIS fxyKPsCJ88-4v'}) # replace the i
d with id of file you want to access
downloaded.GetContentFile('melb data.csv')
In [ ]:
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
df = pd.read csv('melb data.csv')
df.head()
Out[]:
     Suburb
              Address Rooms Type
                                     Price Method SellerG
                                                            Date Distance Postcode Bedroom2 Bathroom
              85 Turner
0 Abbotsford
                               h 1480000.0
                                                  Biggin 3/12/2016
                                                                           3067.0
                          2
                                                                     2.5
                                                                                       2.0
                                                                                                1.0
                   25
1 Abbotsford Bloomburg
                          2
                               h 1035000.0
                                                  Biggin 4/02/2016
                                                                     2.5
                                                                           3067.0
                                                                                       2.0
                                                                                                1.0
              5 Charles
2 Abbotsford
                               h 1465000.0
                                              SP
                                                  Biggin 4/03/2017
                                                                     2.5
                                                                           3067.0
                                                                                       3.0
                                                                                                2.0
                   St
                   40
```

```
In [ ]:
```

df.tail()

Abbotsford

3 Abbotsford Federation

55a Park

Out[]:

	Suburb	Address	Rooms	Туре	Price	Method	SellerG	Date	Distance	Postcode	Bedroom2	Bath
13575	Wheelers Hill	12 Strada Cr	4	h	1245000.0	s	Barry	26/08/2017	16.7	3150.0	4.0	

Biggin 4/03/2017

VB Nelson 4/06/2016

2.5

2.5

3067.0

3067.0

3.0

3.0

2.0

1.0

3

850000.0

h 1600000.0

```
13576 William Statement Avidences Rooms Typle 1031 Protect Meth 68 VS William 26/08/2018 Distantic Postocol Bedroom Bath
                         83
13577 Williamstown
                                       h 1170000.0
                                                           S
                                                                Raine 26/08/2017
                                                                                                             3.0
                      Power
                                  3
                                                                                        6.8
                                                                                               3016.0
                         St
                         96
13578 Williamstown
                                        h 2500000.0
                                                          PI Sweeney 26/08/2017
                                                                                        6.8
                                                                                               3016.0
                                                                                                             4.0
                     Verdon
                    6 Agnes
13579
          Yarraville
                                       h 1285000.0
                                                               Village 26/08/2017
                                                                                        6.3
                                                                                               3013.0
                                                                                                             4.0
```

df.shape

Out[]:

(13580, 21)

In []:

df.describe()

Out[]:

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	В
count	13580.000000	1.358000e+04	13580.000000	13580.000000	13580.000000	13580.000000	13518.000000	13580.000000	7
mean	2.937997	1.075684e+06	10.137776	3105.301915	2.914728	1.534242	1.610075	558.416127	
std	0.955748	6.393107e+05	5.868725	90.676964	0.965921	0.691712	0.962634	3990.669241	
min	1.000000	8.500000e+04	0.000000	3000.000000	0.000000	0.000000	0.000000	0.000000	
25%	2.000000	6.500000e+05	6.100000	3044.000000	2.000000	1.000000	1.000000	177.000000	
50%	3.000000	9.030000e+05	9.200000	3084.000000	3.000000	1.000000	2.000000	440.000000	
75%	3.000000	1.330000e+06	13.000000	3148.000000	3.000000	2.000000	2.000000	651.000000	
max	10.000000	9.000000e+06	48.100000	3977.000000	20.000000	8.000000	10.000000	433014.000000	44
4									F

In []:

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13580 entries, 0 to 13579

Data columns (total 21 columns):

#	Column	Non-Null Count	Dtype
		10500	
0	Suburb	13580 non-null	object
1	Address	13580 non-null	object
2	Rooms	13580 non-null	int64
3	Type	13580 non-null	object
4	Price	13580 non-null	float64
5	Method	13580 non-null	object
6	SellerG	13580 non-null	object
7	Date	13580 non-null	object
8	Distance	13580 non-null	float64
9	Postcode	13580 non-null	float64
10	Bedroom2	13580 non-null	float64
11	Bathroom	13580 non-null	float64
12	Car	13518 non-null	float64
13	Landsize	13580 non-null	float64
14	BuildingArea	7130 non-null	float64
15	YearBuilt	8205 non-null	float64
16	CouncilArea	12211 non-null	object
17	Lattitude	13580 non-null	float64
1 8	T.onatitude	13580 non-niill	float64

```
19 Regionname 13580 non-null object 20 Propertycount 13580 non-null float64
dtypes: float64(12), int64(1), object(8)
memory usage: 2.2+ MB
In [ ]:
df.isnull().sum()
Out[]:
Suburb
                    0
                    0
Address
                    0
Rooms
Type
                    0
                   0
Price
Method
                   0
SellerG
                   0
                   0
Date
Distance
                   0
Postcode
                   0
                  0
Bedroom2
Bathroom
                   0
Car
                  62
Landsize
                  0
BuildingArea
               6450
YearBuilt
                5375
CouncilArea
                1369
Lattitude
                 0
                   0
Longtitude
                  0
Regionname
                   0
Propertycount
dtype: int64
In [ ]:
missing values count = df.isnull().sum()
total_cells = np.product(df.shape)
total missing = missing values count.sum();
percentage missing = (total missing/total cells) * 100
print(f"Missing Percentage: {percentage missing}%")
Missing Percentage: 4.648292306613367%
In [ ]:
for col in df.columns:
    if df[col].dtype != object:
        df[col].fillna(value=df[col].mean(), inplace=True)
df['CouncilArea'].fillna("unknown", inplace=True)
missing values count = df.isnull().sum()
print(missing_values_count)
Suburb
                 0
                 0
Address
                 0
Rooms
                 0
Type
                 0
Price
Method
                0
SellerG
                0
Date
                0
Distance
                0
Postcode
Bedroom2
                0
Bathroom
                0
Car
                0
Landsize
BuildingArea
                0
YearBuilt
                0
CouncilArea
                 0
Lattitude
                 0
Longtitude
```

```
0
Propertycount
dtype: int64
In [ ]:
filtered df = df
filtered df.head()
Out[]:
      Suburb
               Address Rooms Type
                                       Price Method SellerG
                                                               Date Distance Postcode Bedroom2 Bathroom
              85 Turner
0 Abbotsford
                                 h 1480000.0
                                                     Biggin 3/12/2016
                                                                         2.5
                                                                               3067.0
                                                                                            2.0
                                                                                                     1.0
                    St
                    25
1 Abbotsford Bloomburg
                                 h 1035000.0
                                                     Biggin 4/02/2016
                                                                         2.5
                                                                               3067.0
                                                                                            2.0
                                                                                                     1.0
              5 Charles
2 Abbotsford
                                 h 1465000.0
                                                 SP
                                                     Biggin 4/03/2017
                                                                         2.5
                                                                               3067.0
                                                                                            3.0
                                                                                                     2.0
                    St
                    40
3 Abbotsford Federation
                            3
                                    850000.0
                                                     Biggin 4/03/2017
                                                                         2.5
                                                                               3067.0
                                                                                            3.0
                                                                                                     2.0
               55a Park
  Abbotsford
                                 h 1600000.0
                                                 VB Nelson 4/06/2016
                                                                               3067.0
                                                                                                     1.0
                                                                         2.5
                                                                                            3.0
                    St
In [ ]:
filtered df.isnull().sum()
Out[]:
                   0
Suburb
                   0
Address
                   0
Rooms
Type
Price
                   0
Method
                   0
SellerG
Date
                   0
Distance
                   0
                   0
Postcode
Bedroom2
                   0
Bathroom
                   0
Car
Landsize
                   0
BuildingArea
                   0
                   0
YearBuilt
                   0
CouncilArea
                   0
Lattitude
Longtitude
                   0
Regionname
                   0
Propertycount
dtype: int64
In [ ]:
filtered df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13580 entries, 0 to 13579
Data columns (total 21 columns):
     Column
                      Non-Null Count Dtype
 #
 0
     Suburb
                       13580 non-null object
     Address
 1
                       13580 non-null object
                       13580 non-null
                                         int64
```

0

Regionname

2

3

4

Rooms

Type

Price

13580 non-null

13580 non-null

object

float64

```
Method
                    13580 non-null
                                   object
    SellerG
                   13580 non-null object
 7
   Date
                   13580 non-null object
 8
   Distance
                   13580 non-null float64
 9
                   13580 non-null float64
    Postcode
 10 Bedroom2
                   13580 non-null float64
                   13580 non-null float64
 11 Bathroom
                   13580 non-null float64
 12
    Car
                                   float64
 13 Landsize
                    13580 non-null
                                   float64
 14
    BuildingArea
                   13580 non-null
                                   float64
 15
    YearBuilt
                   13580 non-null
 16
    CouncilArea
                   13580 non-null object
 17 Lattitude
                    13580 non-null float64
                   13580 non-null float64
 18 Longtitude
 19 Regionname
                   13580 non-null object
 20 Propertycount 13580 non-null float64
dtypes: float64(12), int64(1), object(8)
memory usage: 2.2+ MB
In [ ]:
filtered df['Suburb'].value counts()
Out[]:
Reservoir
                  359
Richmond
                  260
Bentleigh East
                  249
Preston
                  239
Brunswick
                  222
Upwey
                    1
Rockbank
                   1
New Gisborne
                   1
Attwood
                   1
Officer
                    1
Name: Suburb, Length: 314, dtype: int64
In [ ]:
filtered df['Type'].value counts()
Out[]:
     9449
     3017
    1114
Name: Type, dtype: int64
In [ ]:
filtered df['Method'].value counts()
Out[]:
S
      9022
SP
      1703
      1564
VB
      1199
       92
SA
Name: Method, dtype: int64
In [ ]:
filtered df['SellerG'].value counts()
Out[]:
Nelson
                 1565
Jellis
                 1316
hockingstuart
                 1167
                 1011
Barry
                  701
Ray
```

```
Naison
McNaughton
                    1
Aquire
                    1
Upper
                    1
Jim
Name: SellerG, Length: 268, dtype: int64
In [ ]:
filtered df['CouncilArea'].value counts()
Out[]:
unknown
                     1369
Moreland
                     1163
Boroondara
                     1160
Moonee Valley
                      997
Darebin
                      934
Glen Eira
                      848
Stonnington
                      719
                      692
Maribyrnong
                      647
Yarra
Port Phillip
                      628
                      594
Banyule
Bayside
                      489
Melbourne
                      470
Hobsons Bay
                      434
Brimbank
                      424
Monash
                      333
                      311
Manningham
Whitehorse
                      304
Kingston
                      207
Whittlesea
                      167
Hume
                      164
Wyndham
                       86
Maroondah
                       80
Knox
                       80
Melton
                       66
                       53
Frankston
                      52
Greater Dandenong
                       38
Casey
Nillumbik
                       36
Yarra Ranges
                      18
Cardinia
                       8
Macedon Ranges
                       7
Moorabool
                       1
Unavailable
                        1
Name: CouncilArea, dtype: int64
In [ ]:
filtered df['Regionname'].value counts()
Out[]:
Southern Metropolitan
                              4695
Northern Metropolitan
                              3890
Western Metropolitan
Eastern Metropolitan
                             1471
South-Eastern Metropolitan
                              450
                               53
Eastern Victoria
Northern Victoria
                                41
Western Victoria
                                32
Name: Regionname, dtype: int64
In [ ]:
new df = filtered df.drop(['Suburb', 'Address', 'SellerG', 'Date'], axis="columns")
new_df.head()
Out[]:
```

```
Rooms Type
                     Price Method Distance Postcode Bedroom2 Bathroom Car Landsize BuildingArea
                                                                                                              YearBuilt C
                                 s
0
        2
              h 1480000.0
                                          2.5
                                                 3067.0
                                                               2.0
                                                                          1.0
                                                                               1.0
                                                                                       202.0
                                                                                                 151.96765 1964.684217
1
        2
              h 1035000.0
                                 S
                                          2.5
                                                 3067.0
                                                               2.0
                                                                          1.0
                                                                               0.0
                                                                                       156.0
                                                                                                 79.00000 1900.000000
              h 1465000.0
                                                 3067.0
                                                                               0.0
                                                                                       134.0
                                                                                                 150.00000 1900.000000
                                          2.5
                                                               3.0
                                                                          2.0
3
        3
                  850000.0
                                 PΙ
                                          2.5
                                                               3.0
                                                                          2.0
                                                                               1.0
                                                                                        94.0
                                                                                                151.96765 1964.684217
                                                 3067.0
              h 1600000.0
                                          2.5
                                                                               2.0
                                                                                       120.0
                                                                                                142.00000 2014.000000
        4
                                VB
                                                 3067.0
                                                               3.0
                                                                          1.0
                                                                                                                       •
```

```
In [ ]:
```

1 Type 13580 non-null object 13580 non-null 2 float64 Price 3 Method 13580 non-null object 4 Distance 13580 non-null float64 5 Postcode 13580 non-null float64 6 Bedroom2 13580 non-null float64 7 Bathroom 13580 non-null float64 8 13580 non-null float64 9 Landsize 13580 non-null float64 10 BuildingArea 13580 non-null float64 13580 non-null 11 YearBuilt float64 13580 non-null 12 CouncilArea object 13 Lattitude 13580 non-null float64 14 Longtitude 13580 non-null float64 15 Regionname 13580 non-null 16 Propertycount 13580 non-null dtypes: float64(12), int64(1), object(4)

```
In [ ]:
```

memory usage: 1.8+ MB

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

for i in new_df.columns:
    if new_df[i].dtype== object:
        new_df[i] = le.fit_transform(new_df[i])
```

In []:

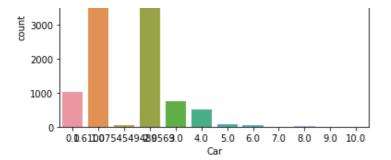
```
new_df.head()
```

Out[]:

	Rooms	Туре	Price	Method	Distance	Postcode	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	С
0	2	0	1480000.0	1	2.5	3067.0	2.0	1.0	1.0	202.0	151.96765	1964.684217	
1	2	0	1035000.0	1	2.5	3067.0	2.0	1.0	0.0	156.0	79.00000	1900.000000	
2	3	0	1465000.0	3	2.5	3067.0	3.0	2.0	0.0	134.0	150.00000	1900.000000	
3	3	0	850000.0	0	2.5	3067.0	3.0	2.0	1.0	94.0	151.96765	1964.684217	
4	4	0	1600000.0	4	2.5	3067.0	3.0	1.0	2.0	120.0	142.00000	2014.000000	

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13580 entries, 0 to 13579
Data columns (total 17 columns):
    Column
                   Non-Null Count
 0
                    13580 non-null int64
    Rooms
 1
    Type
                    13580 non-null int64
    Price
                    13580 non-null float64
 3
    Method
                    13580 non-null int64
   Distance
                    13580 non-null float64
 5
                                   float64
    Postcode
                    13580 non-null
    Bedroom2
                                   float64
                    13580 non-null
 7
                                   float64
    Bathroom
                    13580 non-null
                                   float64
 8
    Car
                    13580 non-null
 9
    Landsize
                    13580 non-null
                                   float64
 10 BuildingArea
                    13580 non-null
                                   float64
 11 YearBuilt
                    13580 non-null float64
                    13580 non-null int64
 12 CouncilArea
 13 Lattitude
                    13580 non-null float64
 14 Longtitude
                    13580 non-null float64
 15 Regionname
                    13580 non-null int64
 16 Propertycount 13580 non-null float64
dtypes: float64(12), int64(5)
memory usage: 1.8 MB
In [ ]:
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.countplot('Price', data=new df)
Out[]:
<matplotlib.axes. subplots.AxesSubplot at 0x7f8fb1964410>
  100
   80
   60
   40
   20
   0
                       Price
In [ ]:
sns.countplot('Car', data=new df)
Out[]:
<matplotlib.axes. subplots.AxesSubplot at 0x7f8f9a01aa10>
  5000
  4000
```

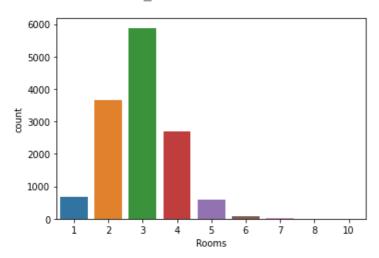
new df.info()



sns.countplot('Rooms', data=new_df)

Out[]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8f99d88f50>

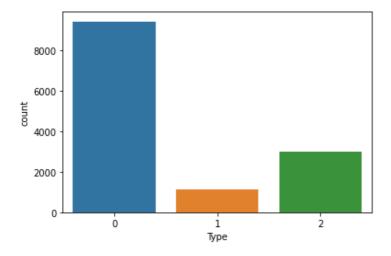


In []:

sns.countplot('Type', data=new_df)

Out[]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8f99b84510>



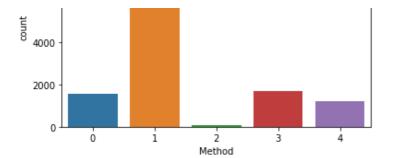
In []:

sns.countplot('Method', data=new_df)

Out[]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8f99968fd0>

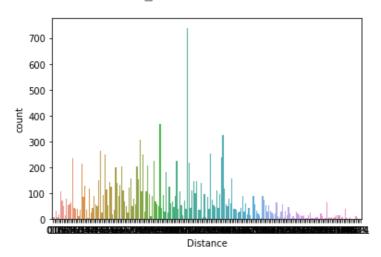




sns.countplot('Distance', data=new_df)

Out[]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8f95264190>

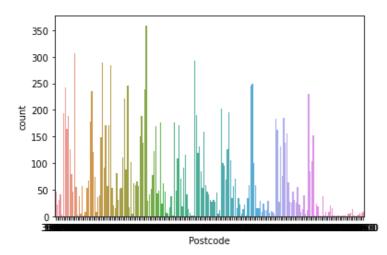


In []:

sns.countplot('Postcode', data=new_df)

Out[]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8f95470a50>



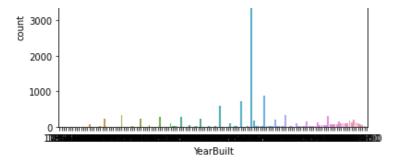
In []:

sns.countplot('YearBuilt', data=new_df)

Out[]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8f9316fc50>

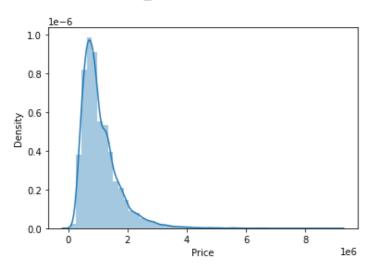




```
sns.distplot(df['Price'])
```

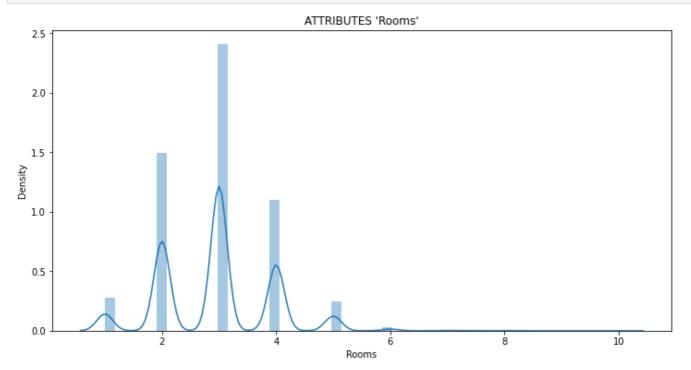
Out[]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f8f9e478cd0>



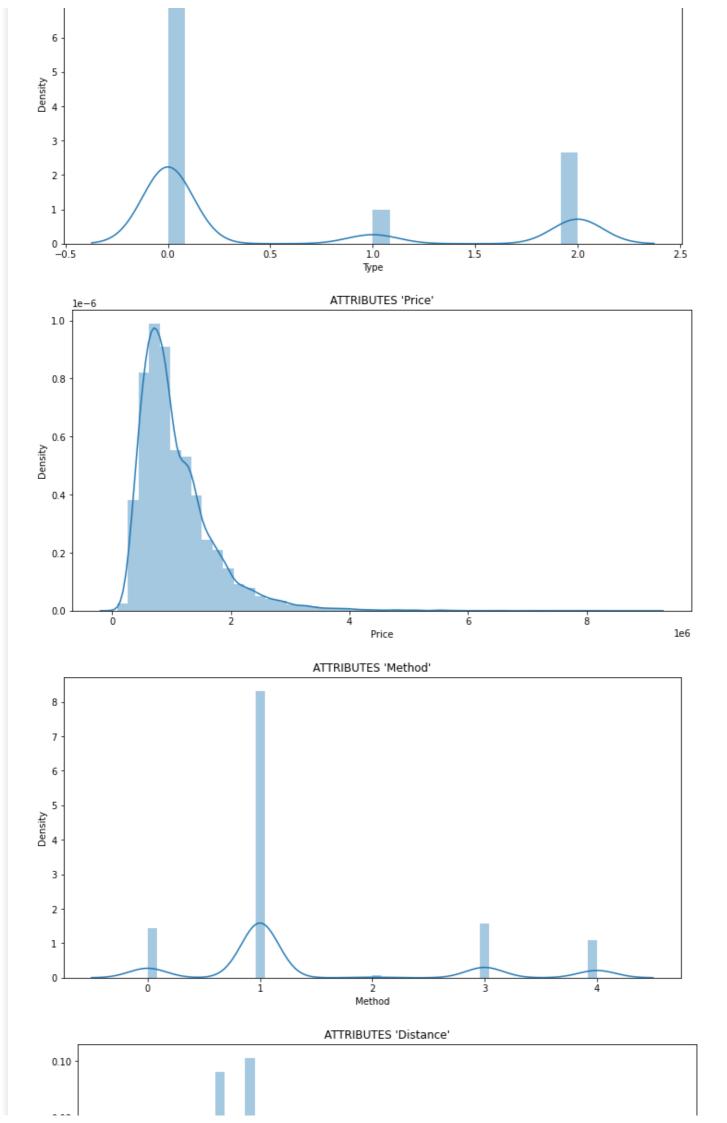
In []:

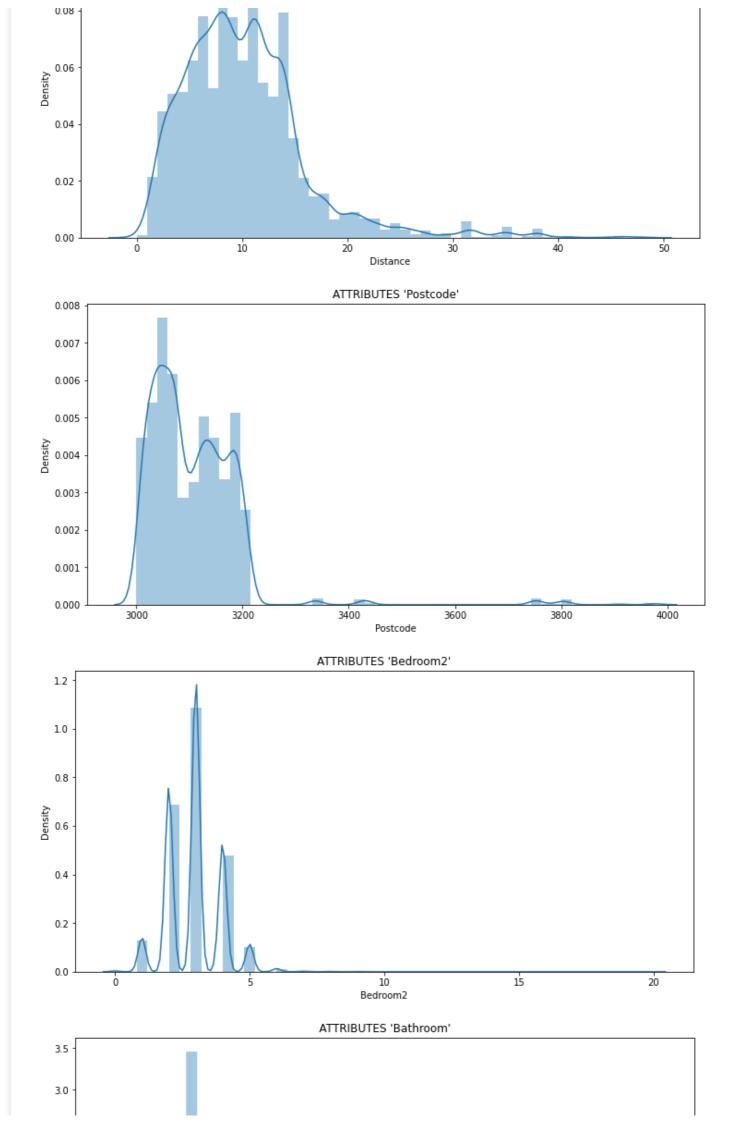
```
for i in new_df.columns[:-1]:
    plt.figure(figsize=(12,6))
    plt.title("ATTRIBUTES '%s'"%i)
    sns.distplot(new_df[i])
```

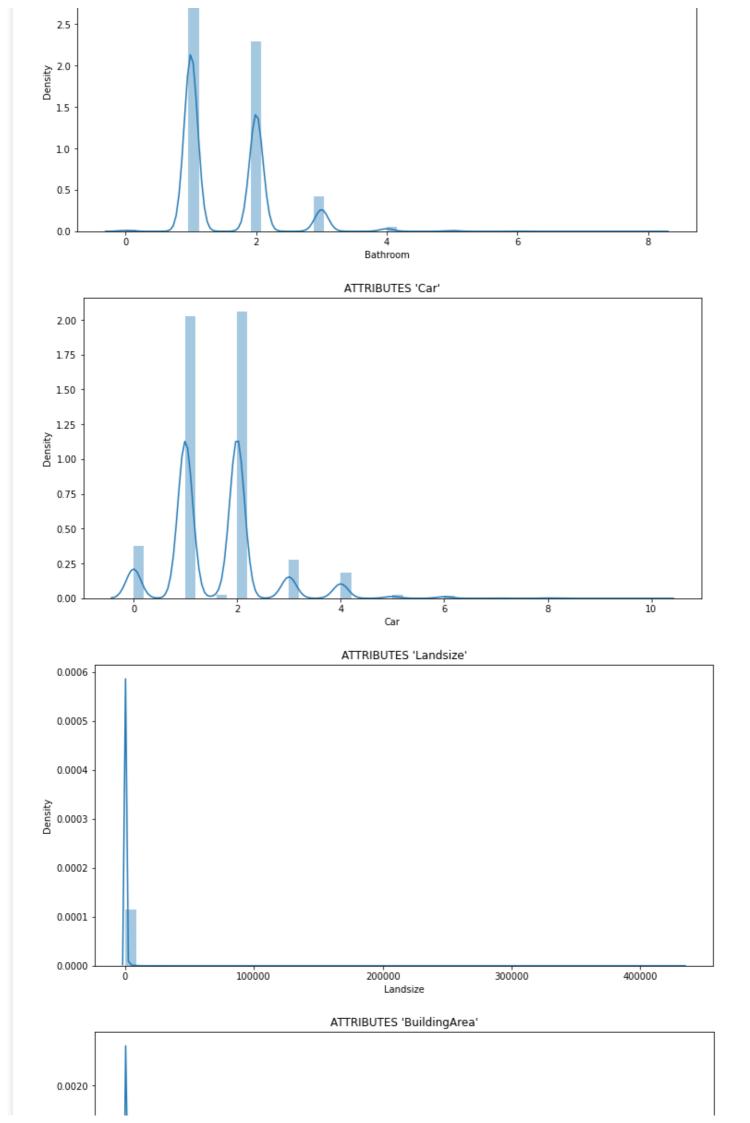


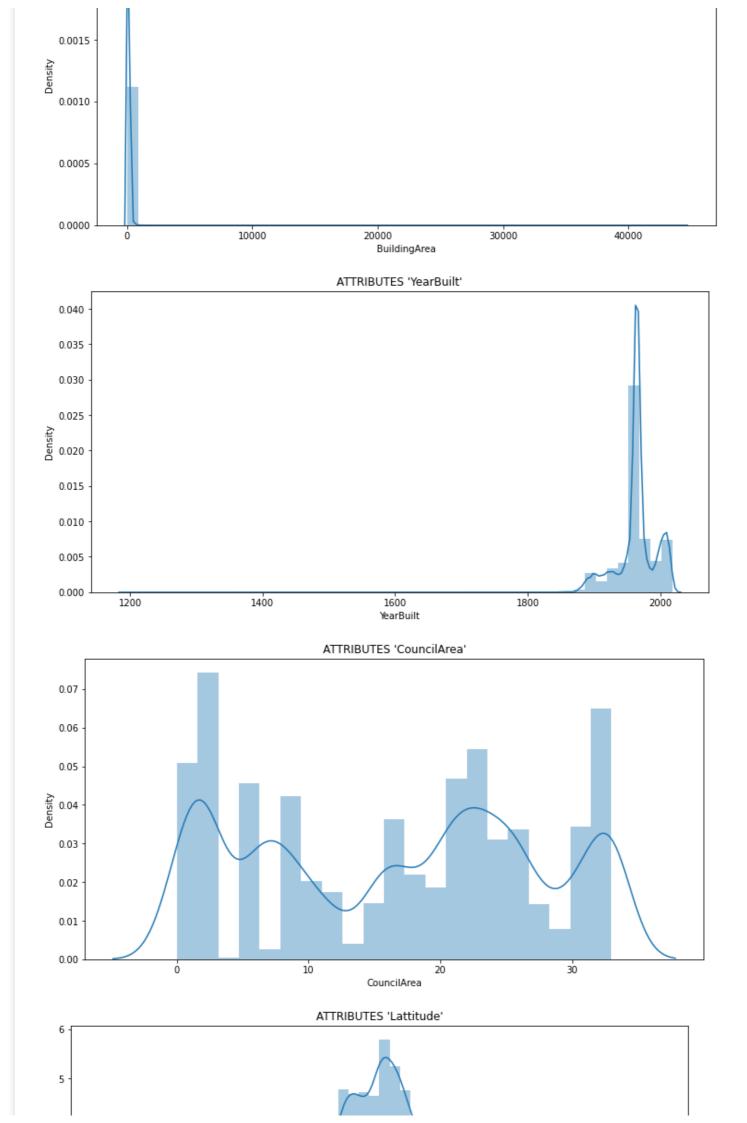
ATTRIBUTES 'Type'

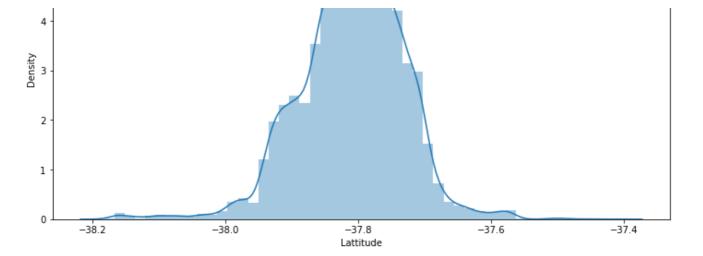
7 -

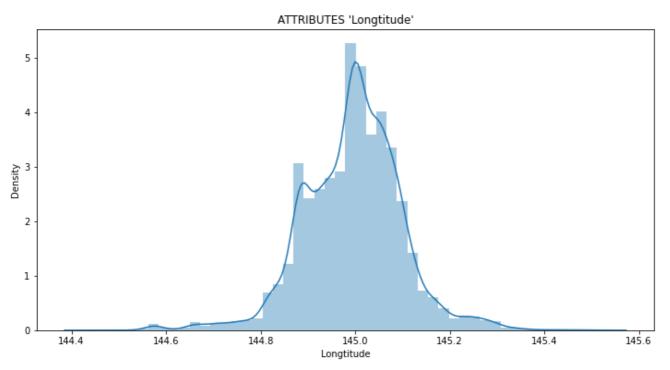


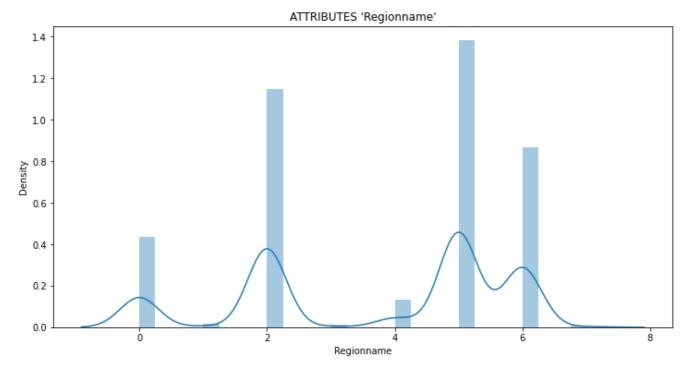






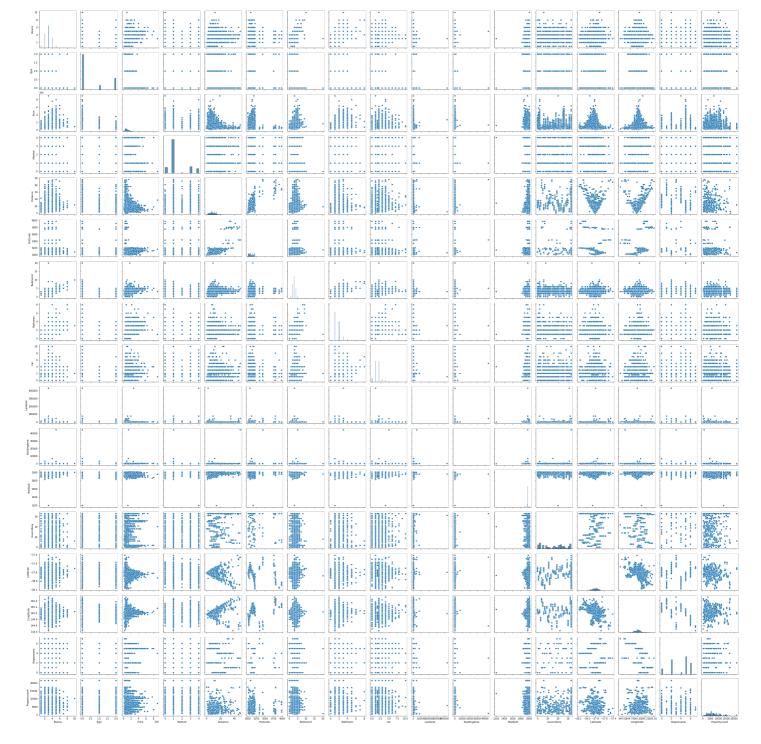






In []:
sns.pairplot(data=new_df)

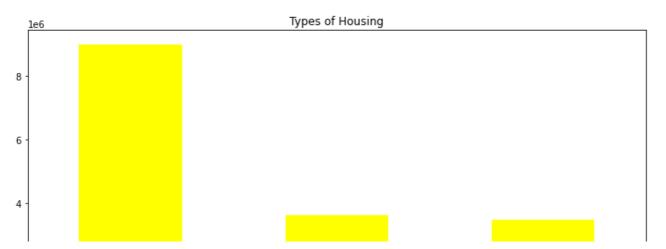
Out[]:

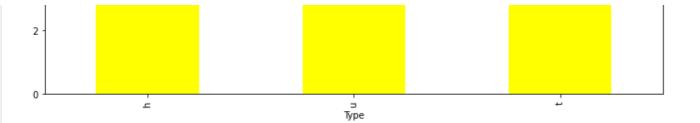


```
df.groupby('Type')['Price'].max().sort_values(ascending=False).head(10).plot(kind='bar',
figsize=(12,6),color='yellow')
plt.title('Types of Housing')
```

Out[]:

Text(0.5, 1.0, 'Types of Housing')

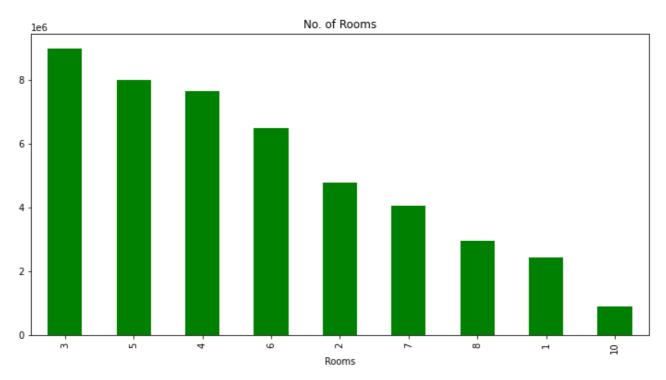




```
df.groupby('Rooms')['Price'].max().sort_values(ascending=False).head(10).plot(kind='bar'
, figsize=(12,6),color='green')
plt.title('No. of Rooms')
```

Out[]:

Text(0.5, 1.0, 'No. of Rooms')

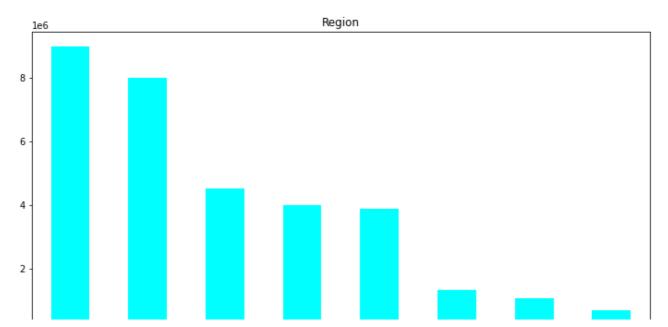


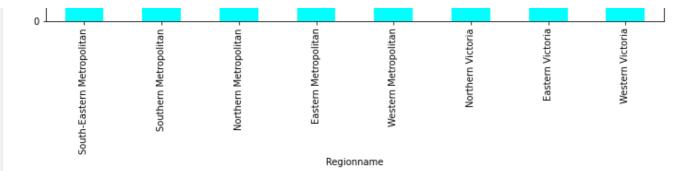
In []:

```
df.groupby('Regionname')['Price'].max().sort_values(ascending=False).head(10).plot(kind=
'bar', figsize=(12,6),color='aqua')
plt.title('Region')
```

Out[]:

Text(0.5, 1.0, 'Region')

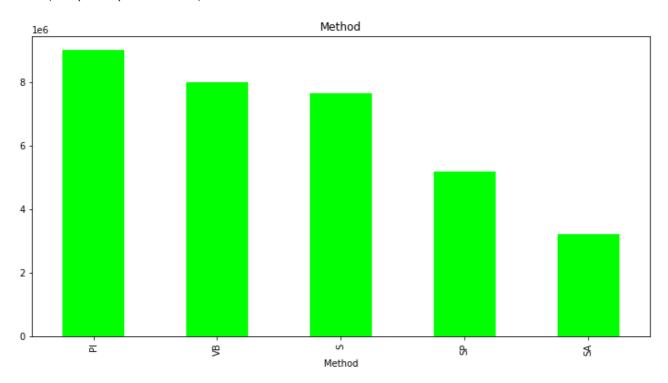




```
df.groupby('Method')['Price'].max().sort_values(ascending=False).head(10).plot(kind='bar
', figsize=(12,6),color='lime')
plt.title('Method')
```

Out[]:

Text(0.5, 1.0, 'Method')



In []:

```
fig = plt.figure(figsize=(10,6))
sns.heatmap(new_df.corr(), annot=True)
```

Out[]:

<matplotlib.axes. subplots.AxesSubplot at 0x7f8f9c84c190>

```
1.0
      Rooms - 1 -0.55 0.5 -0.0510.29 0.055 0.94 0.59 0.41 0.0260.0910.0520.0540.016 0.1 -0.0290.082
             -0.55 1 -0.410.078-0.230.027-0.53-0.23-0.27-0.0180.0530.27-0.0040.110.00710.0650.083
                                                                                                            0.8
                  -0.41 1 0.0380.16 0.11 0.48 0.47 0.24 0.038 0.07 -0.260.0640.21 0.2 0.0790.042
     Method -0.0510.0780.038 1 0.0520.016-0.050.0039.0220.0180.0088.0240.026.000166.0360.0170.034
             -0.29 -0.23 -0.160.052 1 0.43 0.3 0.13 0.26 0.0250.074 0.19-0.0840.13 0.24-0.0970.055
                                                                                                            - 0.6
   Postcode -0.0550.027 0.11-0.0160.43 1 0.061 0.11 0.05 0.0250.0410.0250.00510.41 0.45-0.0110.062
  Bedroom2 - 0.94 -0.53 0.48 -0.05 0.3 0.061 1 0.58 0.4 0.0260.0890.0420.0450.016 0.1 -0.0350.081
                                                                                                            - 0.4
                  -0.23 0.470.00350.13 0.11 0.58 1 0.32 0.0370.084 0.12-0.0330.071 0.12 0.0260.052
   Bathroom -
              0.41 0.27 0.24-0.0220.26 0.05 0.4 0.32 1 0.0270.0680.0790.0790.0020.0630.0160.024
                                                                                                            0.2
   Landsize -0.0260.0180.0380.0180.0250.0250.0260.0370.027 1 0.0950.007 D.0150.00970.011-0.01-50.006
BuildingArea -0.0910.0530.070.0088.0740.0410.0890.0840.0680.095 1 0.0030200024.0320.0170.0024.023
                                                                                                            - 0.0
   YearBuilt -0.0520.27 -0.260.0240.19 0.0250.0420.12 0.07 90.007 0.003 1 -0.0280.0470.00270.020.0049
 CouncilArea -0.0540.0040.0640.0260.0840.00540.0450.0330.0790.016.00024.028 1 0.09-0.0910.0910.0910.002
   Lattitude -0.016-0.11-0.20.0001-0.13-0.410.0160.0710.0012.00910.0320.0470.09 1 -0.36-0.320.047
  Longtitude - 0.1 0.0071 0.2 -0.0360.24 0.45 0.1 0.12 0.0630.0110.0170.00270.0910.36 1 -0.480.066
```

```
Propertycount -0.0820.0830.0420.0340.0550.0620.0810.0520.024.0069.02 D.0049.002 D.0470.066-0.14
                                                       YearBuilt
                                                               Lattitude
                                                    BuildingArea
                                    3edroom2
                                                            CouncilArea
                                                                        Regionname
                                                                           Propertycount
                                                                   Longtitude
In [ ]:
X = new df.drop(['Price'], axis="columns")
Out[]:
   Rooms Type Method Distance Postcode Bedroom2 Bathroom Car Landsize BuildingArea
                                                                                              YearBuilt CouncilArea
                                                                         202.0
0
        2
                              2.5
                                     3067.0
                                                  2.0
                                                             1.0
                                                                1.0
                                                                                  151.96765 1964.684217
                                                                                                                31
1
        2
              0
                      1
                              2.5
                                    3067.0
                                                  2.0
                                                             1.0 0.0
                                                                         156.0
                                                                                  79.00000 1900.000000
                                                                                                                31
        3
                              2.5
                                     3067.0
                                                  3.0
                                                             2.0
                                                                 0.0
                                                                         134.0
                                                                                  150.00000 1900.000000
                                                                                                                31
                                     3067.0
                                                  3.0
                                                                          94.0
                                                                                  151.96765 1964.684217
3
        3
              0
                      0
                              2.5
                                                             2.0
                                                                 1.0
                                                                                                                31
                              2.5
                                     3067.0
                                                  3.0
                                                             1.0 2.0
                                                                         120.0
                                                                                  142.00000 2014.000000
                                                                                                                31
In [ ]:
y = new df['Price']
y.head()
Out[]:
      1480000.0
1
      1035000.0
      1465000.0
      850000.0
4
      1600000.0
Name: Price, dtype: float64
In [ ]:
from sklearn.model selection import train test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
In [ ]:
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(X train, y train)
In [ ]:
```

Regionname +0.0290.0650.0790.0170.0970.0110.0350.0260.0160.0150.00240.02-0.0920.32 -0.48 1 -0.14