

```

import pandas as pd
import os
import numpy as np
import plotly as py
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
from plotly.offline import init_notebook_mode
import statsmodels.api as sm
from statsmodels.formula.api import ols
import statistics
import scipy.stats as stats
from scipy.stats import kendalltau
from scipy.stats import spearmanr
from scipy.stats import pearsonr
from statsmodels.graphics.regressionplots import plot_partregress_grid
os.chdir("E:\Ginu_StudyMaterials\Sem2\Dissertation\Data")

property_prices = pd.read_csv("PPR_ALL_v1.csv", na_values = ("N/A",
"NA", "--", " "), encoding = 'unicode_escape')
property_prices

```

	Date of Sale (dd/mm/yyyy) \
0	01/01/2010
1	03/01/2010
2	04/01/2010
3	04/01/2010
4	04/01/2010
...	...
516581	28/01/2022
516582	28/01/2022
516583	28/01/2022
516584	28/01/2022
516585	28/01/2022

County \	Address	Postal Code
0 Dublin	5 Braemor Drive, Churchtown, Co.Dublin	NaN
1 Laois	134 Ashewood Walk, Summerhill Lane, Portlaoise	NaN
2 Dublin	1 Meadow Avenue, Dundrum, Dublin 14	NaN
3 Meath	1 The Haven, Mornington	NaN
4 Kilkenny	11 Melville Heights, Kilkenny	NaN
...	...	...
...		

516581	LACKEN, MULTYFARNHAM, MULLINGAR	NaN
Westmeath		
516582	LARCH HILL, COLMAN, FETHARD	NaN
Tipperary		
516583	SHERRYS WOOD, BELLEWSTOWN, CO MEATH	NaN
Meath		
516584	ST JUDES, STONEYFORD, KILKENNY	NaN
Kilkenny		
516585	SYLVAN, DUBLIN ROAD, BRAY	NaN
Wicklow		

	Price (€) Not Full Market Price VAT Exclusive \	
0	343000	No No
1	185000	No Yes
2	438500	No No
3	400000	No No
4	160000	No No
...	...	...
516581	305000	No No
516582	300000	No No
516583	450000	No No
516584	242000	No No
516585	620000	No No

	Description of Property \
0	Second-Hand Dwelling house /Apartment
1	New Dwelling house /Apartment
2	Second-Hand Dwelling house /Apartment
3	Second-Hand Dwelling house /Apartment
4	Second-Hand Dwelling house /Apartment
...	...
516581	Second-Hand Dwelling house /Apartment
516582	Second-Hand Dwelling house /Apartment
516583	Second-Hand Dwelling house /Apartment
516584	Second-Hand Dwelling house /Apartment
516585	Second-Hand Dwelling house /Apartment

	Property Size Description
0	NaN
1	greater than or equal to 38 sq metres and less...
2	NaN
3	NaN
4	NaN
...	...
516581	NaN
516582	NaN
516583	NaN
516584	NaN
516585	NaN

[516586 rows x 9 columns]

```
property_prices.rename({'Date of Sale (dd/mm/yyyy)': 'date_of_sale',  
'Address': 'address', 'Postal Code': 'postal_code', 'County': 'county',  
'Price (€)': 'price', 'Not Full Market Price': 'FMP', 'VAT  
Exclusive': 'VAT_exclusive', 'Description of  
Property': 'property_description', 'Property Size  
Description': 'property_size_description' }, axis=1, inplace=True)
```

```
town_list = pd.read_csv("ie_towns_sample.csv", na_values = ("N/A",  
"NA", "--", " "))  
town_list
```

	id	name	irish_name	county \
0	1	Abartagh	Abartach	Waterford
1	2	Abberanville	Abberanville	Galway
2	3	Abbernadoorny	NaN	Donegal
3	4	Abbert	An Abart	Galway
4	5	Abbert Demesne	Diméin na hAbarta	Galway
...	...	...	...	...
2035	2036	Ayle Lower	An Aill Íochtarach	Clare
2036	2037	Ayle Upper	An Aill Uachtarach	Clare
2037	2038	Aylmerstown	Baile an Aighlmearaigh	Kildare
2038	2039	Aylwardstown	Baile an Aighleartaigh	Kilkenny
2039	2040	Ayresfields	Páirceanna an Iarsaigh	Kilkenny

	country	eircode	grid_reference	easting	northing
latitude \					
0	Republic of Ireland	P36	X121851	212171	85195
52.01916					
1	Republic of Ireland	H62	M609232	160965	223221
53.25817					
2	Republic of Ireland	F94	G898822	189874	382219
54.68795					
3	Republic of Ireland	H54	M539410	153970	241071
53.41798					
4	Republic of Ireland	H54	M545421	154524	242155
53.42776					
...	...	...	...	...	...
...					
2035	Republic of Ireland	V94	R535836	153502	183628
52.90183					
2036	Republic of Ireland	V94	R525847	152520	184776
52.91206					
2037	Republic of Ireland	R14	S805888	280572	188877
52.94497					
2038	Republic of Ireland	Y34	S662213	266278	121350
52.34021					
2039	Republic of Ireland	R95	S496567	249670	156738
52.65994					

	longitude	postal_town	local_government_area
0	-7.82345	Youghal	Waterford City And County Council
1	-8.58564	Loughrea	Galway County Council
2	-8.15776	Donegal	Donegal County Council
3	-8.69303	Tuam	Galway County Council
4	-8.68485	Tuam	Galway County Council
...	...	...	...
2035	-8.69172	Limerick	Clare County Council
2036	-8.70649	Limerick	Clare County Council
2037	-6.80217	Athy	Kildare County Council
2038	-7.02831	New Ross	Kilkenny County Council
2039	-7.26668	Kilkenny	Kilkenny County Council

	nuts3_region	type
0	South-East	Townland
1	West	Townland
2	Border	Townland
3	West	Townland
4	West	Townland
...	...	...
2035	Mid-West	Townland
2036	Mid-West	Townland
2037	Mid-East	Townland
2038	South-East	Townland
2039	South-East	Townland

[2040 rows x 16 columns]

```
province_list = town_list[['county', 'province']]
```

```
province_list['province'].unique()
```

```
array(['Munster', 'Connacht', 'Ulster', 'Leinster'], dtype=object)
```

```
df1 = province_list.drop_duplicates(subset= ['county'], keep='first')
df1
```

	county	province
0	Waterford	Munster
1	Galway	Connacht
2	Donegal	Ulster
5	Tipperary	Munster
6	Cork	Munster
11	Limerick	Munster
12	Longford	Leinster
13	Roscommon	Connacht
15	Kerry	Munster
16	Wexford	Leinster
17	Clare	Munster
24	Kilkenny	Leinster
25	Mayo	Connacht
28	Cavan	Ulster
29	Kildare	Leinster
31	Meath	Leinster
33	Westmeath	Leinster
44	Wicklow	Leinster
50	Laois	Leinster
55	Sligo	Connacht
64	Dublin	Leinster
72	Offaly	Leinster
73	Louth	Leinster
74	Carlow	Leinster
109	Leitrim	Connacht
114	Monaghan	Ulster

```
df1['county'].unique()
```

```
array(['Waterford', 'Galway', 'Donegal', 'Tipperary', 'Cork',
      'Limerick',
      'Longford', 'Roscommon', 'Kerry', 'Wexford', 'Clare',
      'Kilkenny',
      'Mayo', 'Cavan', 'Kildare', 'Meath', 'Westmeath', 'Wicklow',
      'Laois', 'Sligo', 'Dublin', 'Offaly', 'Louth', 'Carlow',
      'Leitrim',
      'Monaghan'], dtype=object)
```

```
from opencage.geocoder import OpenCageGeocode
key = '40d783cbf75143b48b8528d1804a3ccd' # get api key from:
https://opencagedata.com
```

```
geocoder = OpenCageGeocode(key)
```

```
list_lat = [] # create empty lists
```

```
list_long = []
```

```
for index, row in df1.iterrows(): # iterate over rows in dataframe
```

```

City = row['county']
State = row['province']
query = str(City)+','+str(State)
#loc = row['temp_add']
#query = str(loc)

results = geocoder.geocode(query)
lat = results[0]['geometry']['lat']
long = results[0]['geometry']['lng']

list_lat.append(lat)
list_long.append(long)

# create new columns from lists

df1['lat'] = list_lat

df1['lon'] = list_long

df_merge_col = pd.merge(property_prices, df1, on='county', how='left')

df_merge_col

```

	date_of_sale	address \
0	01/01/2010	5 Braemor Drive, Churchtown, Co.Dublin
1	03/01/2010	134 Ashewood Walk, Summerhill Lane, Portlaoise
2	04/01/2010	1 Meadow Avenue, Dundrum, Dublin 14
3	04/01/2010	1 The Haven, Mornington
4	04/01/2010	11 Melville Heights, Kilkenny
...	...	...
516581	28/01/2022	LACKEN, MULTYFARNHAM, MULLINGAR
516582	28/01/2022	LARCH HILL, COLMAN, FETHARD
516583	28/01/2022	SHERRYS WOOD, BELLEWSTOWN, CO MEATH
516584	28/01/2022	ST JUDES, STONEYFORD, KILKENNY
516585	28/01/2022	SYLVAN, DUBLIN ROAD, BRAY

	postal_code	county	price	FMP	VAT_exclusive \
0	NaN	Dublin	343000	No	No
1	NaN	Laois	185000	No	Yes
2	NaN	Dublin	438500	No	No
3	NaN	Meath	400000	No	No
4	NaN	Kilkenny	160000	No	No
...	...	...	...	..	...
516581	NaN	Westmeath	305000	No	No
516582	NaN	Tipperary	300000	No	No
516583	NaN	Meath	450000	No	No
516584	NaN	Kilkenny	242000	No	No
516585	NaN	Wicklow	620000	No	No

	property_description \
0	Second-Hand Dwelling house /Apartment
1	New Dwelling house /Apartment
2	Second-Hand Dwelling house /Apartment
3	Second-Hand Dwelling house /Apartment
4	Second-Hand Dwelling house /Apartment
...	...
516581	Second-Hand Dwelling house /Apartment
516582	Second-Hand Dwelling house /Apartment
516583	Second-Hand Dwelling house /Apartment
516584	Second-Hand Dwelling house /Apartment
516585	Second-Hand Dwelling house /Apartment

	property_size_description	province \
0		NaN Leinster
1	greater than or equal to 38 sq metres and less...	Leinster
2		NaN Leinster
3		NaN Leinster
4		NaN Leinster
...	...	...
516581		NaN Leinster
516582		NaN Munster
516583		NaN Leinster
516584		NaN Leinster
516585		NaN Leinster

	lat	lon
0	53.349764	-6.260273
1	52.998458	-7.398034
2	53.349764	-6.260273
3	53.649784	-6.588529
4	52.651022	-7.248495
...	...	...
516581	53.557790	-7.347856
516582	52.684821	-7.898128
516583	53.649784	-6.588529
516584	52.651022	-7.248495
516585	52.958147	-6.381971

[516586 rows x 12 columns]

df\_merge\_col.drop\_duplicates()

	date_of_sale	address \
0	01/01/2010	5 Braemor Drive, Churchtown, Co.Dublin
1	03/01/2010	134 Ashewood Walk, Summerhill Lane, Portlaoise
2	04/01/2010	1 Meadow Avenue, Dundrum, Dublin 14
3	04/01/2010	1 The Haven, Mornington
4	04/01/2010	11 Melville Heights, Kilkenny
...	...	...

516581	28/01/2022	LACKEN, MULTYFARNHAM, MULLINGAR
516582	28/01/2022	LARCH HILL, COLMAN, FETHARD
516583	28/01/2022	SHERRYS WOOD, BELLEWSTOWN, CO MEATH
516584	28/01/2022	ST JUDES, STONEYFORD, KILKENNY
516585	28/01/2022	SYLVAN, DUBLIN ROAD, BRAY

	postal_code	county	price	FMP	VAT_exclusive	\
0	NaN	Dublin	343000	No	No	
1	NaN	Laois	185000	No	Yes	
2	NaN	Dublin	438500	No	No	
3	NaN	Meath	400000	No	No	
4	NaN	Kilkenny	160000	No	No	
...	...	...	...	..	...	
516581	NaN	Westmeath	305000	No	No	
516582	NaN	Tipperary	300000	No	No	
516583	NaN	Meath	450000	No	No	
516584	NaN	Kilkenny	242000	No	No	
516585	NaN	Wicklow	620000	No	No	

	property_description	\
0	Second-Hand Dwelling house /Apartment	
1	New Dwelling house /Apartment	
2	Second-Hand Dwelling house /Apartment	
3	Second-Hand Dwelling house /Apartment	
4	Second-Hand Dwelling house /Apartment	
...	...	
516581	Second-Hand Dwelling house /Apartment	
516582	Second-Hand Dwelling house /Apartment	
516583	Second-Hand Dwelling house /Apartment	
516584	Second-Hand Dwelling house /Apartment	
516585	Second-Hand Dwelling house /Apartment	

	property_size_description	province	\
0		NaN	Leinster
1	greater than or equal to 38 sq metres and less...		Leinster
2		NaN	Leinster
3		NaN	Leinster
4		NaN	Leinster
...		...	...
516581		NaN	Leinster
516582		NaN	Munster
516583		NaN	Leinster
516584		NaN	Leinster
516585		NaN	Leinster

	lat	lon
0	53.349764	-6.260273
1	52.998458	-7.398034
2	53.349764	-6.260273
3	53.649784	-6.588529



```

4          52.651022 -7.248495
...
516581    53.557790 -7.347856
516582    52.684821 -7.898128
516583    53.649784 -6.588529
516584    52.651022 -7.248495
516585    52.958147 -6.381971

```

```
[515792 rows x 12 columns]
```

```

df_merge_col['property_description'] =
df_merge_col['property_description'].replace(['Teach/Árasán Cónaithe
Atháimhe', 'Teach/Árasán Cónaithe Nua', 'Teach/?ras?n C?naithe Nua'],
['Second-Hand Dwelling house /Apartment', 'New Dwelling house
/Apartment', 'New Dwelling house /Apartment'])

df_merge_col['property_size_description'] =
df_merge_col['property_size_description'].replace(['n?os l? n? 38 m?
adar cearnach', 'níos mó ná nó cothrom le 38 méadar cearnach agus níos
lú ná 125 méadar cearnach'], ['less than 38 sq metres', 'greater than or
equal to 38 sq metres and less than 125 sq metres'])

df_merge_col['county'] = df_merge_col['county'].replace(['Baile ?tha
Cliath', 'Ní Bhaineann'], ['Dublin', ''])

df_merge_col['postal_code'] =
df_merge_col['postal_code'].replace(['Baile Átha Cliath 3', 'Baile Átha
Cliath 4', 'Baile Átha Cliath 5', 'Baile Átha Cliath 9', 'Baile Átha
Cliath 14', 'Baile Átha Cliath 15', 'Baile Átha Cliath 18', 'Baile ?tha
Cliath 17', 'Ní Bhaineann'], ['Dublin 3', 'Dublin 4', 'Dublin 5', 'Dublin
9', 'Dublin 14', 'Dublin 15', 'Dublin 18', 'Dublin 17', ''])

df_merge_col['property_description'] =
df_merge_col['property_description'].replace(['Second-Hand Dwelling
house /Apartment', 'New Dwelling house /Apartment'], ['Second-Hand',
'NewHouse'])

df_merge_col['property_size_description'] =
df_merge_col['property_size_description'].replace(['greater than 125
sq metres'], ['greater than or equal to 125 sq metres'])

df_merge_col['address'] = df_merge_col['address'].str.title()

#df_merge_col['price'] =df_merge_col['price'].str.replace('£', '')
#df_merge_col['price'] = df_merge_col['price'].str.replace(',', ' ')

# changing the date to pandas datetime format

pd.to_datetime(df_merge_col['date_of_sale'].astype(str), format
='%d/%m/%Y')

```

```

0      2010-01-01
1      2010-01-03
2      2010-01-04
3      2010-01-04
4      2010-01-04
...
516581 2022-01-28
516582 2022-01-28
516583 2022-01-28
516584 2022-01-28
516585 2022-01-28
Name: date_of_sale, Length: 516586, dtype: datetime64[ns]

cd_date1 = pd.to_datetime(df_merge_col['date_of_sale'].astype(str),
format = '%d/%m/%Y')
#pd.to_datetime(df_merge_col['date_of_sale'].astype(str), format
='%d/%m/%Y')
df_merge_col['month_year'] =
pd.to_datetime(df_merge_col['date_of_sale']).dt.to_period('M')

df_merge_col['year'] = cd_date1.dt.year
df_merge_col['month'] = cd_date1.dt.month

df_merge_col = df_merge_col.assign(location=df_merge_col["county"])

# Split the location between Dublin and outside Dublin
df_merge_col['location'] = df_merge_col['location'].map({
    "Cork": "Outside", "Galway": "Outside", "Kildare": "Outside",
    "Meath": "Outside", "Limerick": "Outside",
    "Wexford": "Outside", "Wicklow": "Outside", "Kerry": "Outside",
    "Donegal": "Outside", "Waterford": "Outside",
    "Tipperary": "Outside", "Louth": "Outside", "Mayo": "Outside",
    "Clare": "Outside", "Westmeath": "Outside",
    "Cavan": "Outside", "Sligo": "Outside", "Kilkenny": "Outside",
    "Laois": "Outside", "Roscommon": "Outside",
    "Offaly": "Outside", "Carlow": "Outside", "Leitrim": "Outside",
    "Longford": "Outside", "Monaghan": "Outside",
    "Dublin": "Dublin"})
df_merge_col.head()

   date_of_sale      address
postal_code \
0  01/01/2010      5 Braemor Drive, Churchtown, Co.Dublin
NaN
1  03/01/2010  134 Ashewood Walk, Summerhill Lane, Portlaoise
NaN
2  04/01/2010      1 Meadow Avenue, Dundrum, Dublin 14
NaN
3  04/01/2010      1 The Haven, Mornington
NaN
4  04/01/2010     11 Melville Heights, Kilkenny

```

NaN

	county	price	FMP	VAT_exclusive	property_description	\
0	Dublin	343000	No	No	Second-Hand	
1	Laois	185000	No	Yes	NewHouse	
2	Dublin	438500	No	No	Second-Hand	
3	Meath	400000	No	No	Second-Hand	
4	Kilkenny	160000	No	No	Second-Hand	

	property_size_description	province
lat \		
0		NaN Leinster
53.349764		
1	greater than or equal to 38 sq metres and less...	Leinster
52.998458		
2		NaN Leinster
53.349764		
3		NaN Leinster
53.649784		
4		NaN Leinster
52.651022		

	lon	month_year	year	month	location
0	-6.260273	2010-01	2010	1	Dublin
1	-7.398034	2010-03	2010	1	Outside
2	-6.260273	2010-04	2010	1	Dublin
3	-6.588529	2010-04	2010	1	Outside
4	-7.248495	2010-04	2010	1	Outside

df\_merge\_col.info()

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 516586 entries, 0 to 516585  
Data columns (total 16 columns):
```

#	Column	Non-Null Count	Dtype
0	date_of_sale	516586 non-null	object
1	address	516586 non-null	object
2	postal_code	97243 non-null	object
3	county	516586 non-null	object
4	price	516586 non-null	int64
5	FMP	516586 non-null	object
6	VAT_exclusive	516586 non-null	object
7	property_description	516586 non-null	object
8	property_size_description	52800 non-null	object
9	province	516586 non-null	object
10	lat	516586 non-null	float64
11	lon	516586 non-null	float64
12	month_year	516586 non-null	period[M]
13	year	516586 non-null	int64

```
14 month 516586 non-null int64
15 location 516586 non-null object
dtypes: float64(2), int64(3), object(10), period[M](1)
memory usage: 67.0+ MB
```

```
df_merge_col['price'].astype('int64')
```

```
0      343000
1      185000
2      438500
3      400000
4      160000
```

```
...
516581  305000
516582  300000
516583  450000
516584  242000
516585  620000
```

```
Name: price, Length: 516586, dtype: int64
```

```
for county in df_merge_col['county'].unique():
    print('County {}; max house price €{:.0f}m'.format(county,
(df_merge_col[df_merge_col['county'] == county]
['price'].max())/10**6))
```

```
County Dublin; max house price €182m
County Laois; max house price €21m
County Meath; max house price €8m
County Kilkenny; max house price €13m
County Limerick; max house price €13m
County Carlow; max house price €5m
County Cork; max house price €70m
County Clare; max house price €33m
County Sligo; max house price €6m
County Cavan; max house price €5m
County Tipperary; max house price €6m
County Wicklow; max house price €24m
County Roscommon; max house price €2m
County Wexford; max house price €14m
County Mayo; max house price €8m
County Donegal; max house price €4m
County Longford; max house price €2m
County Galway; max house price €35m
County Offaly; max house price €1m
County Kildare; max house price €26m
County Waterford; max house price €8m
County Louth; max house price €7m
County Kerry; max house price €9m
County Westmeath; max house price €14m
County Monaghan; max house price €2m
County Leitrim; max house price €2m
```

```

county = df_merge_col['county'].unique()

max_house_prices = [(df_merge_col[df_merge_col['county'] == county]
['price'].max())/10**6 for county in df_merge_col['county'].unique()]

max_house_prices

[182.378855,
 20.8,
 7.704846,
 12.6,
 13.151162,
 5.101322,
 69.873482,
 33.18,
 5.559481,
 5.4,
 6.230344,
 23.885325,
 1.67,
 13.995,
 7.762438,
 4.464915,
 2.49,
 34.781,
 1.46,
 26.5,
 7.929515,
 7.2,
 9.41,
 14.0,
 2.151174,
 1.546256]

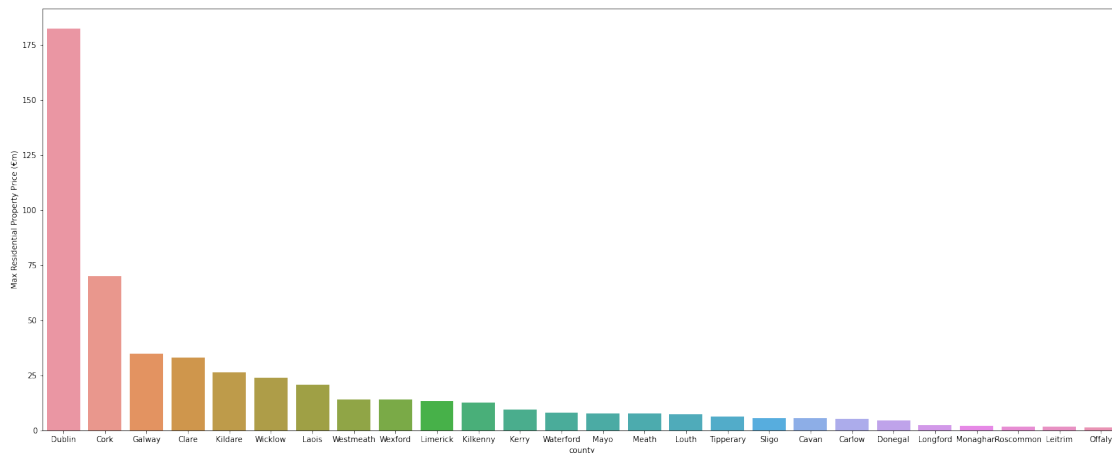
county_price = pd.DataFrame(county)
county_price['max_house_prices'] = max_house_prices
county_price.rename(columns={0: 'county'}, inplace=True)
county_price

```

	county	max_house_prices
0	Dublin	182.378855
1	Laois	20.800000
2	Meath	7.704846
3	Kilkenny	12.600000
4	Limerick	13.151162
5	Carlow	5.101322
6	Cork	69.873482
7	Clare	33.180000
8	Sligo	5.559481
9	Cavan	5.400000
10	Tipperary	6.230344

11	Wicklow	23.885325
12	Roscommon	1.670000
13	Wexford	13.995000
14	Mayo	7.762438
15	Donegal	4.464915
16	Longford	2.490000
17	Galway	34.781000
18	Offaly	1.460000
19	Kildare	26.500000
20	Waterford	7.929515
21	Louth	7.200000
22	Kerry	9.410000
23	Westmeath	14.000000
24	Monaghan	2.151174
25	Leitrim	1.546256

```
fig, ax = plt.subplots(figsize=(25, 10))
plt.ylabel('Max Residential Property Price (€m)')
sns.barplot(x= county_price['county'], y=
max_house_prices,order=county_price.sort_values('max_house_prices',
ascending =False).county )
plt.show()
```



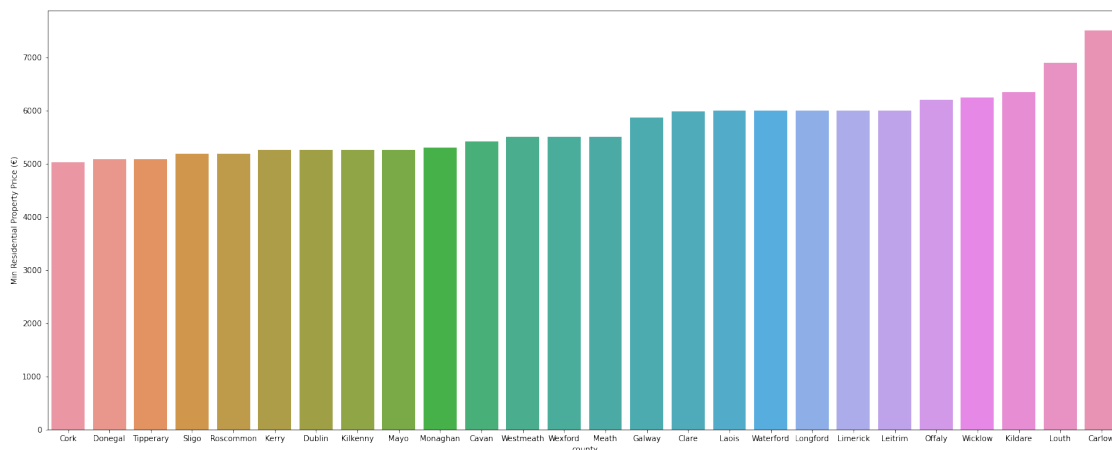
```
min_house_prices = [(df_merge_col[df_merge_col['county'] == county]
['price'].min()) for county in df_merge_col['county'].unique()]
```

```
county_price_min = pd.DataFrame(county)
county_price_min['min_house_prices'] = min_house_prices
county_price_min.rename(columns={0: 'county'},inplace=True)
county_price_min
```

	county	min_house_prices
0	Dublin	5250
1	Laois	6000
2	Meath	5500
3	Kilkenny	5250
4	Limerick	6000

5	Carlow	7500
6	Cork	5031
7	Clare	5987
8	Sligo	5177
9	Cavan	5412
10	Tipperary	5080
11	Wicklow	6248
12	Roscommon	5179
13	Wexford	5500
14	Mayo	5254
15	Donegal	5079
16	Longford	6000
17	Galway	5864
18	Offaly	6200
19	Kildare	6348
20	Waterford	6000
21	Louth	6900
22	Kerry	5250
23	Westmeath	5500
24	Monaghan	5298
25	Leitrim	6000

```
fig, ax = plt.subplots(figsize=(25, 10))
plt.ylabel('Min Residential Property Price (€)')
sns.barplot(x=county_price_min['county'], y= min_house_prices,
order=county_price_min.sort_values('min_house_prices').county )
plt.show()
```



```
property_sizes =
np.delete(df_merge_col['property_size_description'].unique(), 0)
property_sizes
array(['greater than or equal to 38 sq metres and less than 125 sq
metres',
      'greater than or equal to 125 sq metres', 'less than 38 sq
metres'],
      dtype=object)
```

```

median_per_property_size = [df_merge_col['price']
[df_merge_col['property_size_description']==property_size].median()
for property_size in property_sizes]

median_per_property_size

[205000.0, 263436.0, 162832.0]

#property_sizes = list(map(lambda x: x.replace('greater than 125 sq
metres', '= 125 Sqm'), property_sizes))

property_sizes = list(map(lambda x: x.replace('greater than or equal
to 38 sq metres and less than 125 sq metres', '>=38 & <125 Sqm'),
property_sizes))

property_sizes = list(map(lambda x: x.replace('less than 38 sq
metres', '< 38 Sqm'), property_sizes))

property_sizes = list(map(lambda x: x.replace('greater than or equal
to 125 sq metres', '>= 125 Sqm'), property_sizes))

property_sizes

['>=38 & <125 Sqm', '>= 125 Sqm', '< 38 Sqm']

size_price = pd.DataFrame(property_sizes)
size_price

   0
0  >=38 & <125 Sqm
1      >= 125 Sqm
2      < 38 Sqm

size_price['median_per_property_size'] = median_per_property_size
size_price.rename(columns={0: 'property_sizes'}, inplace=True)
size_price

   property_sizes  median_per_property_size
0  >=38 & <125 Sqm          205000.0
1      >= 125 Sqm          263436.0
2      < 38 Sqm          162832.0

fig, ax = plt.subplots(figsize=(10, 5))
sns.barplot(size_price['property_sizes'],
size_price['median_per_property_size'], order=size_price.sort_values('p
roperty_sizes', ascending =False).property_sizes, )
#ax.set_xticklabels(labels=property_sizes, rotation=90)
plt.xlabel('Property Sizes')
#plt.xlabel(" Property Sizes, 0: <38 sqm, 1: =>38 sqm, 2: =>125 sqm")
plt.ylabel('Median House Prices (€)')
plt.show()

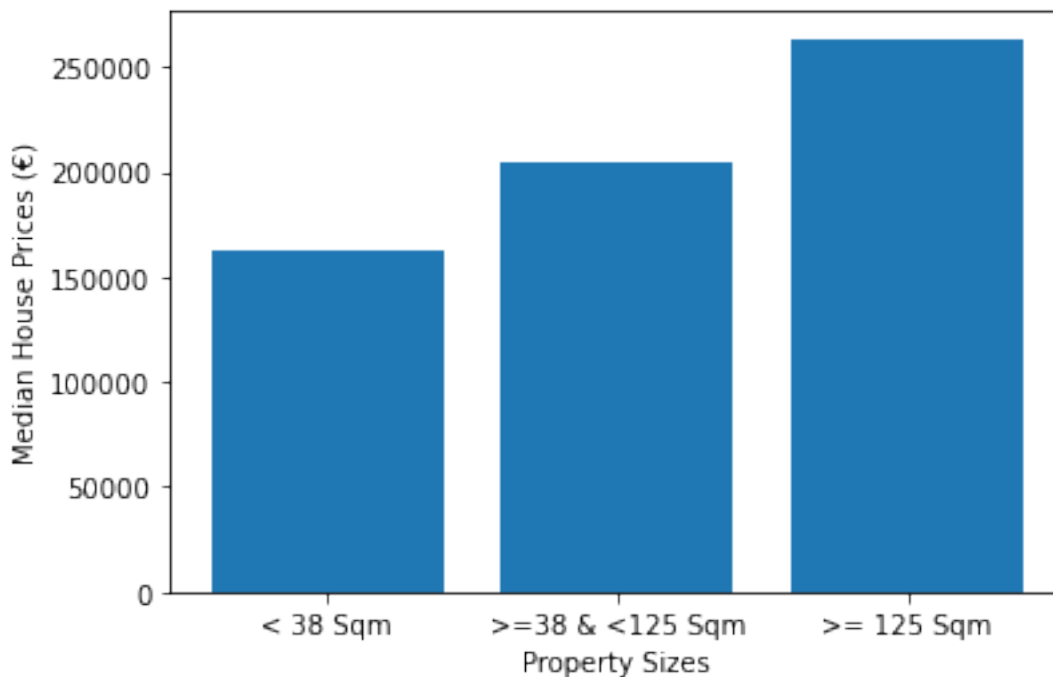
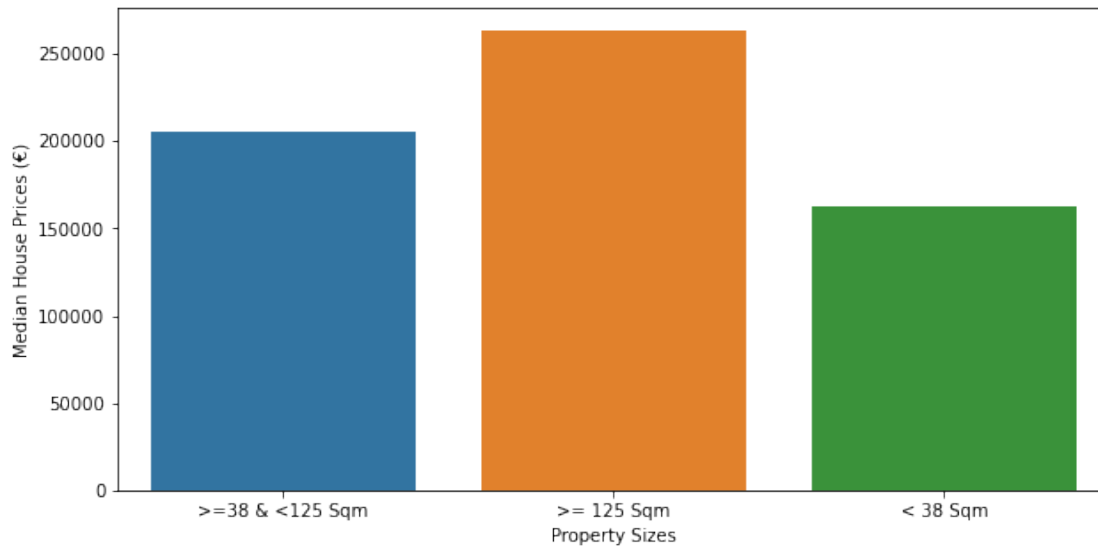
```



```

df_sorted = size_price.sort_values('median_per_property_size')
plt.bar('property_sizes', 'median_per_property_size', data=df_sorted)
plt.xlabel('Property Sizes')
#plt.xlabel(" Property Sizes, 0: <38 sqm, 1: =>38 sqm, 2: =>125 sqm")
plt.ylabel('Median House Prices (€)')
plt.show()

```



```

property_ = (df_merge_col['property_description'].unique())
property_
array(['Second-Hand', 'NewHouse'], dtype=object)

```

```

property_size_house_prices =
[(df_merge_col[df_merge_col['property_description'] ==
property_description]['price'].median()) for property_description in
df_merge_col['property_description'].unique()]

property_size_house_prices

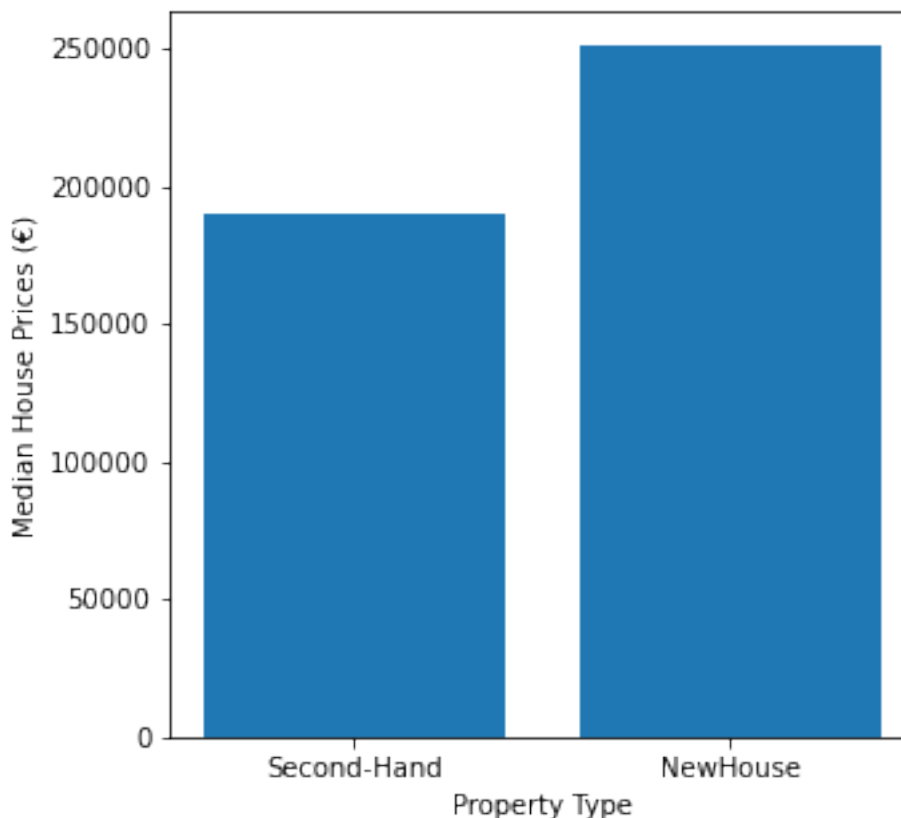
[190000.0, 251101.0]

property_ = list(map(lambda x: x.replace('Second-Hand Dwelling
house /Apartment', 'UsedHouse'), property_))

property_ = list(map(lambda x: x.replace('New Dwelling house
/Apartment', 'NewHouse'), property_))

fig, ax = plt.subplots(figsize=(5, 5))
plt.bar(property_, property_size_house_prices)
#ax.set_xticklabels(labels=property_, rotation=90)
plt.xlabel('Property Type')
plt.ylabel('Median House Prices (€)')
plt.show()

```



```

counties = df_merge_col['county'].unique()
counties

```

```

array(['Dublin', 'Laois', 'Meath', 'Kilkenny', 'Limerick', 'Carlow',
      'Cork', 'Clare', 'Sligo', 'Cavan', 'Tipperary', 'Wicklow',
      'Roscommon', 'Wexford', 'Mayo', 'Donegal', 'Longford',
      'Galway',
      'Offaly', 'Kildare', 'Waterford', 'Louth', 'Kerry',
      'Westmeath',
      'Monaghan', 'Leitrim'], dtype=object)

median_per_county = [df_merge_col['price']
                     [df_merge_col['county']==county].median() for county in counties]

median_per_county = np.asarray(median_per_county)
median_per_county

array([308370., 139995., 233480., 160000., 150000., 140000., 200000.,
      140000., 110000., 104225., 125000., 275000., 89000., 150000.,
      110000., 101743., 84000., 180000., 130000., 250000., 142000.,
      170000., 142500., 138000., 118000., 87000.])

q=[]
for price in median_per_county:
    x = price
    q.append(x)

county_price = pd.DataFrame(q)

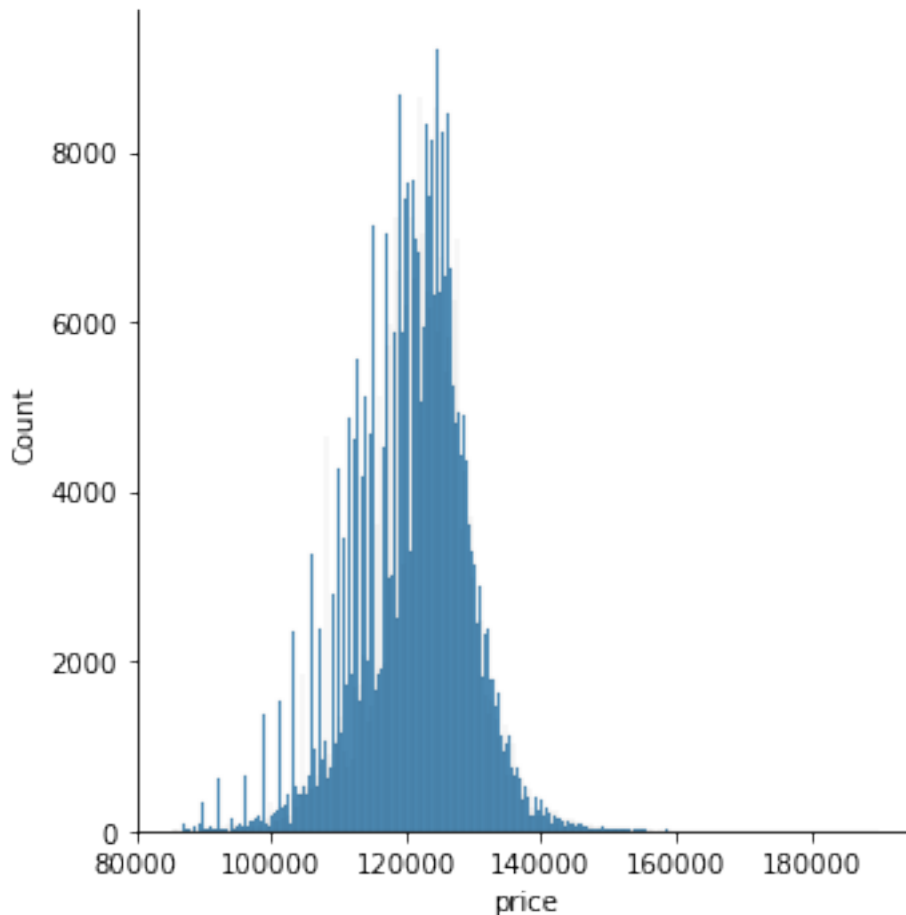
county_price['county'] = counties

county_price.rename(columns={0:'price'},inplace=True)

Univariate Plots
for county in df_merge_col['county'].unique():
    print('County {}; max house price €{:.0f}m'.format(county,
    (df_merge_col[df_merge_col['county'] == county]
    ['price'].max())/10**6))

sns.displot(np.log(df_merge_col['price'])*10**4)
plt.show()

```



df\_merge\_col

	date_of_sale	address \
0	01/01/2010	5 Braemor Drive, Churchtown, Co.Dublin
1	03/01/2010	134 Ashewood Walk, Summerhill Lane, Portlaoise
2	04/01/2010	1 Meadow Avenue, Dundrum, Dublin 14
3	04/01/2010	1 The Haven, Mornington
4	04/01/2010	11 Melville Heights, Kilkenny
...	...	...
516581	28/01/2022	Lacken, Multyfarnham, Mullingar
516582	28/01/2022	Larch Hill, Colman, Fethard
516583	28/01/2022	Sherrys Wood, Bellewstown, Co Meath
516584	28/01/2022	St Judes, Stoneyford, Kilkenny
516585	28/01/2022	Sylvan, Dublin Road, Bray

	postal_code	county	price	FMP	VAT_exclusive
property_description \					
0	NaN	Dublin	343000	No	No
Second-Hand					
1	NaN	Laois	185000	No	Yes
NewHouse					
2	NaN	Dublin	438500	No	No

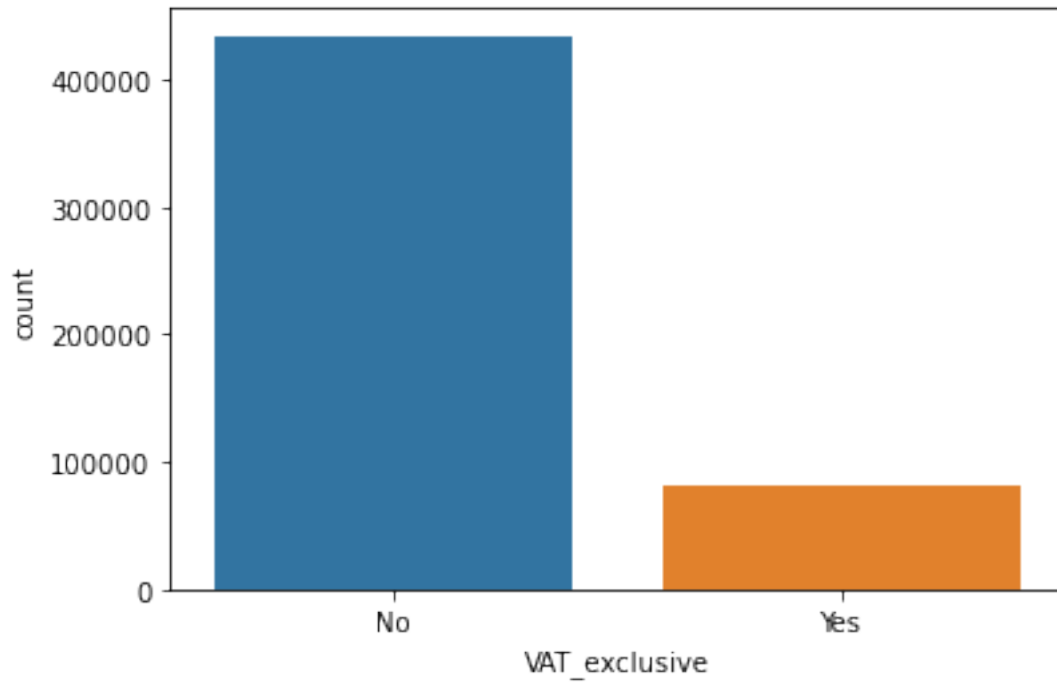
Second-Hand					
3	NaN	Meath	400000	No	No
Second-Hand					
4	NaN	Kilkenny	160000	No	No
Second-Hand					
...	...	...	...	..	...
...					
516581	NaN	Westmeath	305000	No	No
Second-Hand					
516582	NaN	Tipperary	300000	No	No
Second-Hand					
516583	NaN	Meath	450000	No	No
Second-Hand					
516584	NaN	Kilkenny	242000	No	No
Second-Hand					
516585	NaN	Wicklow	620000	No	No
Second-Hand					

	property_size_description	province \
0		NaN Leinster
1	greater than or equal to 38 sq metres and less...	Leinster
2		NaN Leinster
3		NaN Leinster
4		NaN Leinster
...		...
516581		NaN Leinster
516582		NaN Munster
516583		NaN Leinster
516584		NaN Leinster
516585		NaN Leinster

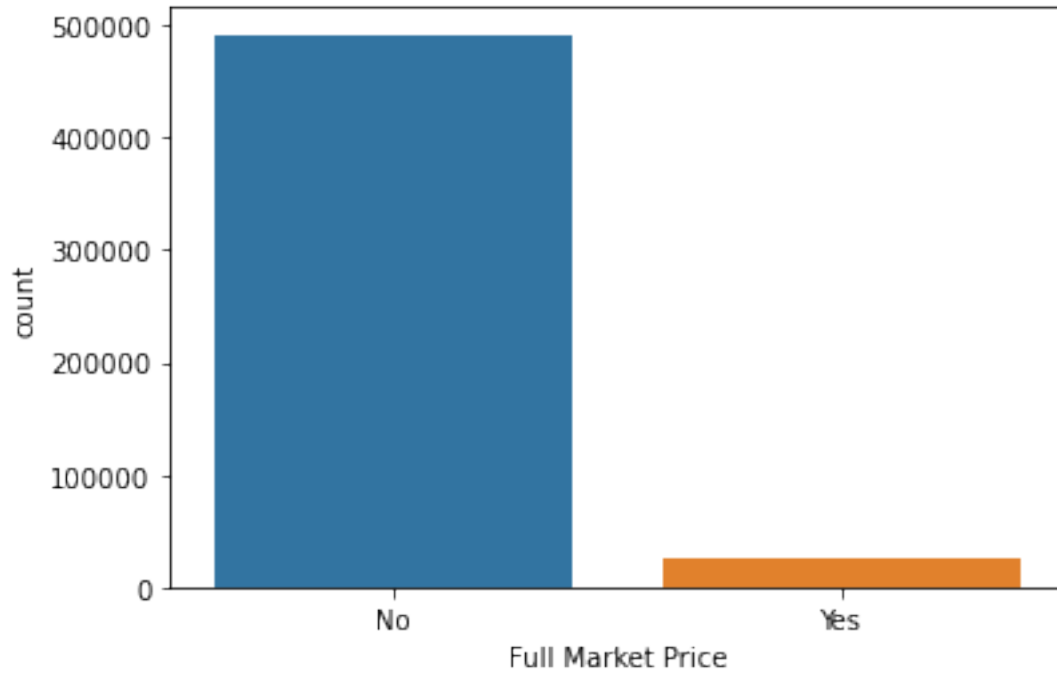
	lat	lon	month_year	year	month	location
0	53.349764	-6.260273	2010-01	2010	1	Dublin
1	52.998458	-7.398034	2010-03	2010	1	Outside
2	53.349764	-6.260273	2010-04	2010	1	Dublin
3	53.649784	-6.588529	2010-04	2010	1	Outside
4	52.651022	-7.248495	2010-04	2010	1	Outside
...	...	...	...	...	...	...
516581	53.557790	-7.347856	2022-01	2022	1	Outside
516582	52.684821	-7.898128	2022-01	2022	1	Outside
516583	53.649784	-6.588529	2022-01	2022	1	Outside
516584	52.651022	-7.248495	2022-01	2022	1	Outside
516585	52.958147	-6.381971	2022-01	2022	1	Outside

[516586 rows x 16 columns]

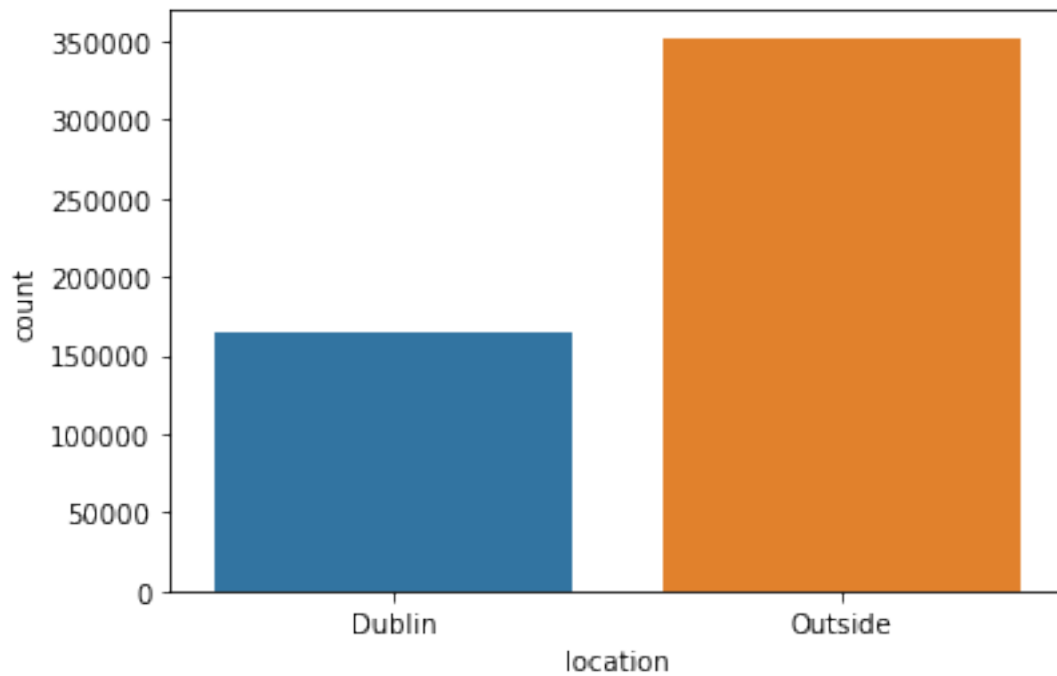
```
sns.countplot(x='VAT_exclusive', data=df_merge_col)
plt.show()
```



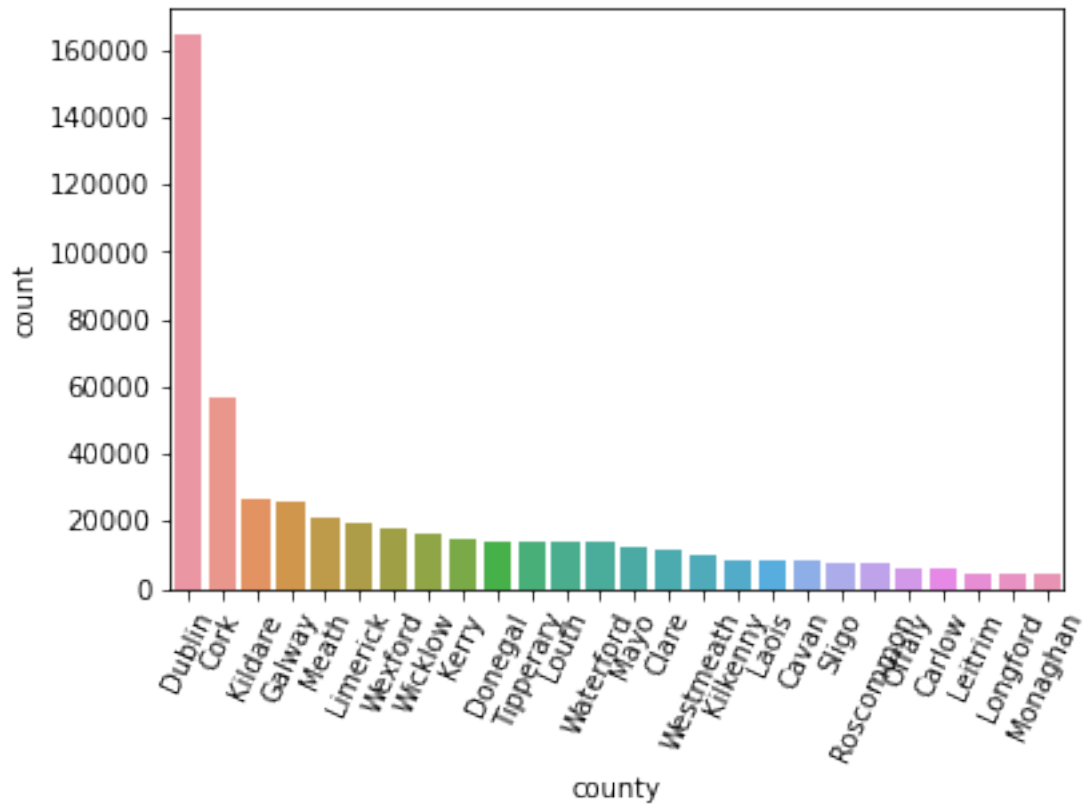
```
p1 = sns.countplot(x='FMP', data=df_merge_col)
p1.set_xlabel("Full Market Price")
plt.show()
```



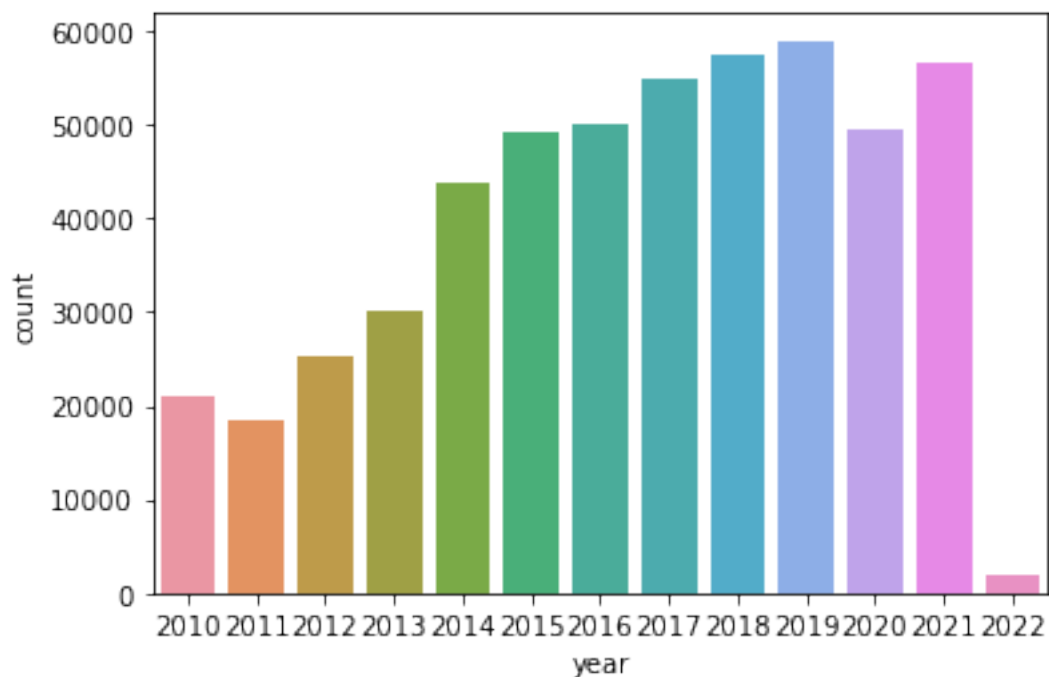
```
sns.countplot(x='location', data=df_merge_col)
plt.show()
```



```
sns.countplot(x='county', data=df_merge_col, order =  
df_merge_col['county'].value_counts().index)  
locs, labels = plt.xticks()  
plt.setp(labels, rotation=65)  
plt.show()
```

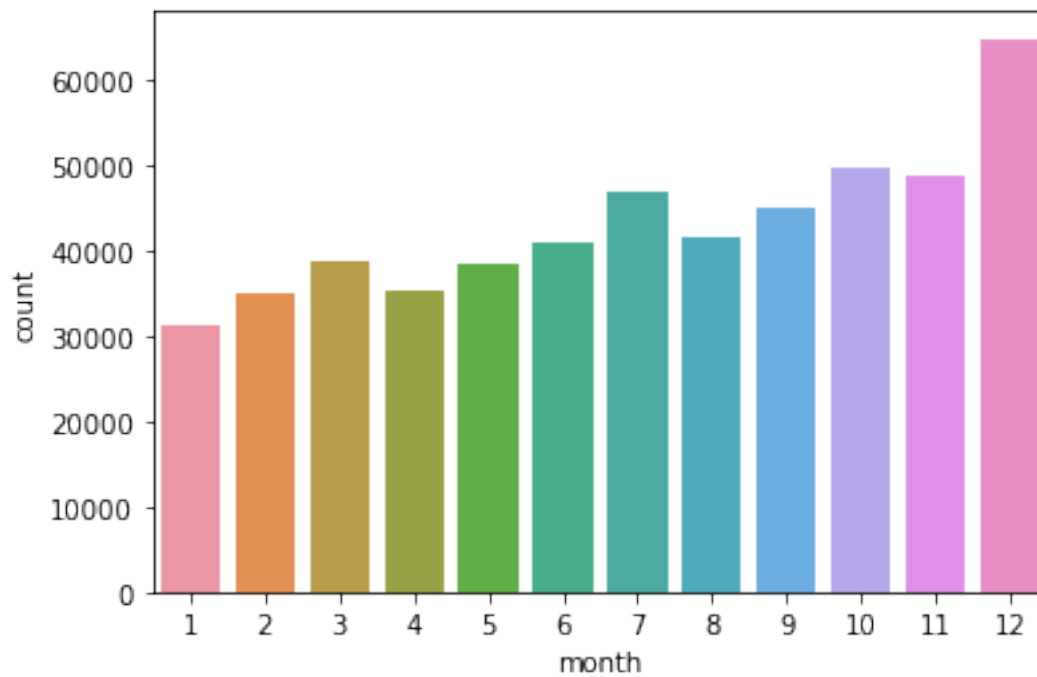


```
#sns.countplot(x='year', data=df_merge_col, order =
df_merge_col['year'].value_counts().index)
sns.countplot(x='year', data=df_merge_col)
plt.show()
```

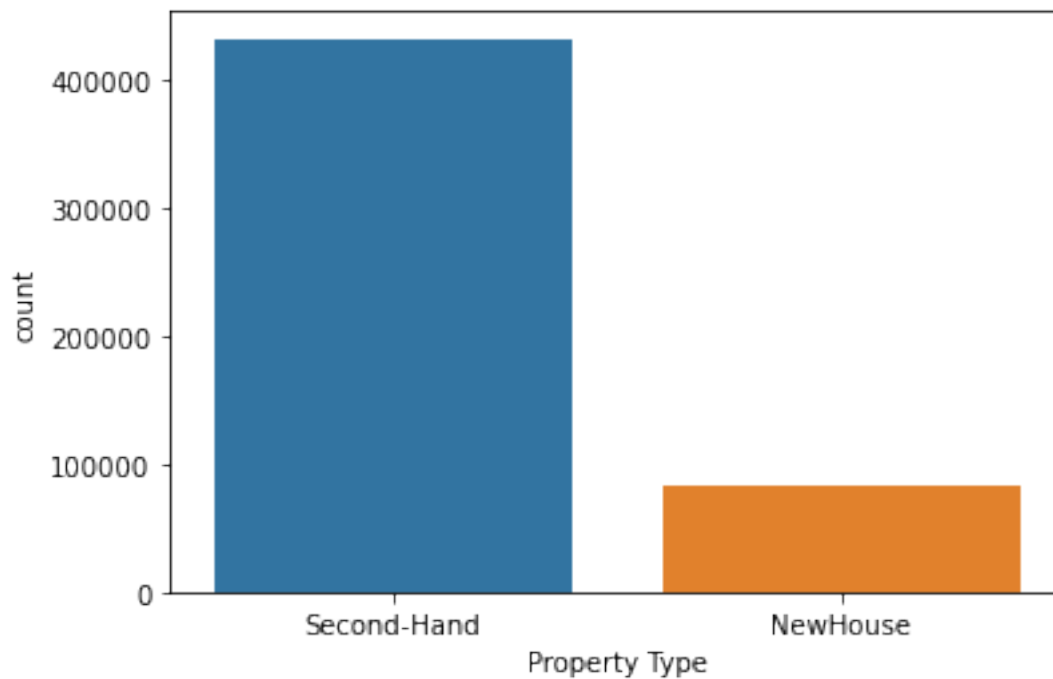




```
sns.countplot(x='month', data=df_merge_col)
plt.show()
```

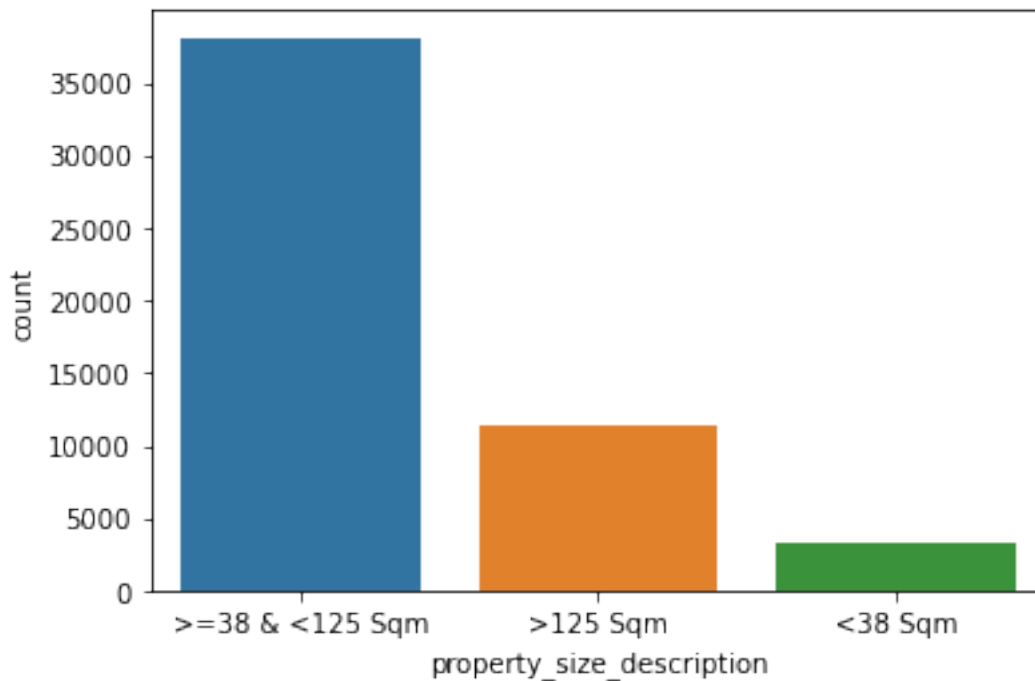


```
sns.countplot(x='property_description', data=df_merge_col)
plt.xlabel('Property Type')
plt.show()
```

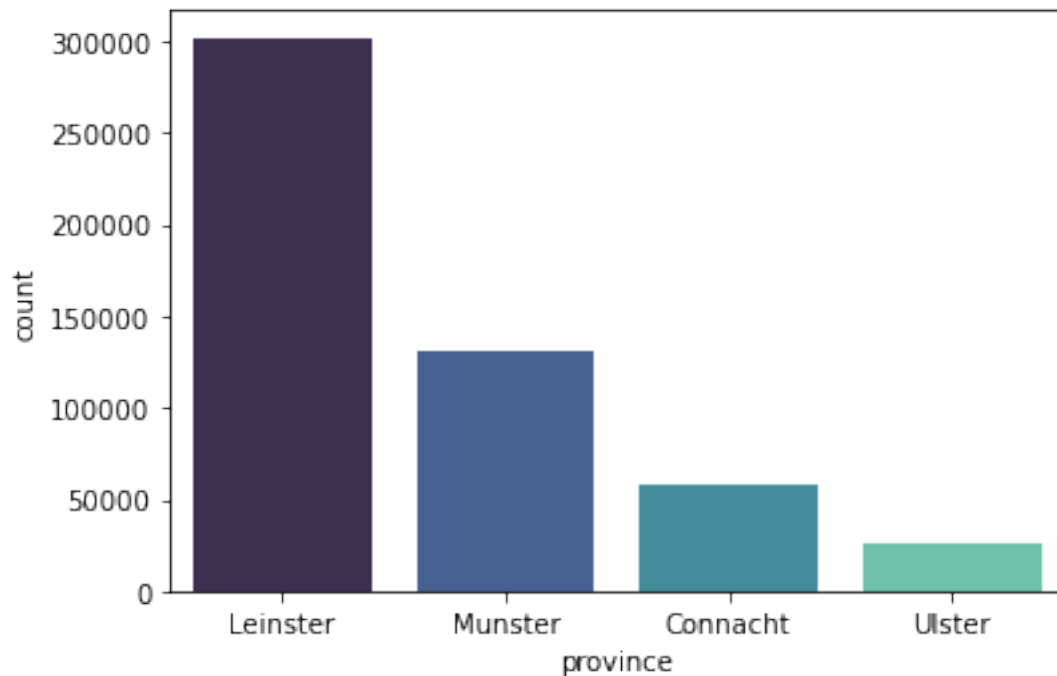


```
df_merge_col['property_size_description'] =
df_merge_col['property_size_description'].replace(['greater than or
equal to 38 sq metres and less than 125 sq metres', 'less than 38 sq
metres', 'greater than or equal to 125 sq metres'], ['>=38 & <125
Sqm', '<38 Sqm', '>=125 Sqm'])
```

```
sns.countplot(x='property_size_description', data=df_merge_col)
plt.show()
```

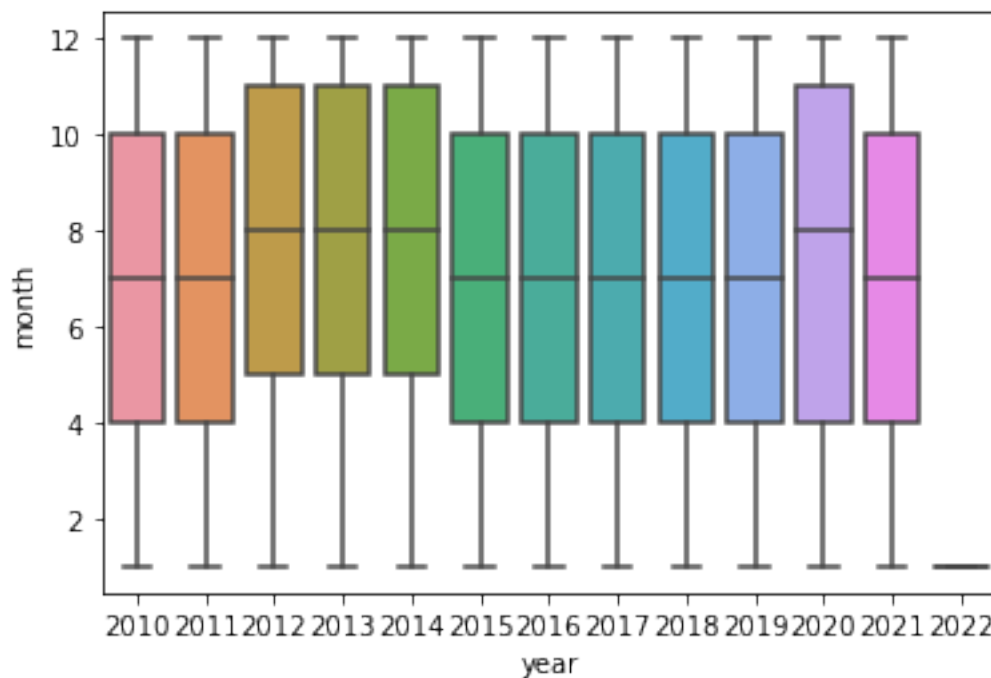


```
sns.countplot(x='province', data=df_merge_col, palette = 'mako')
#sns.color_palette("light:#5A9", as_cmap=True)
plt.show()
```

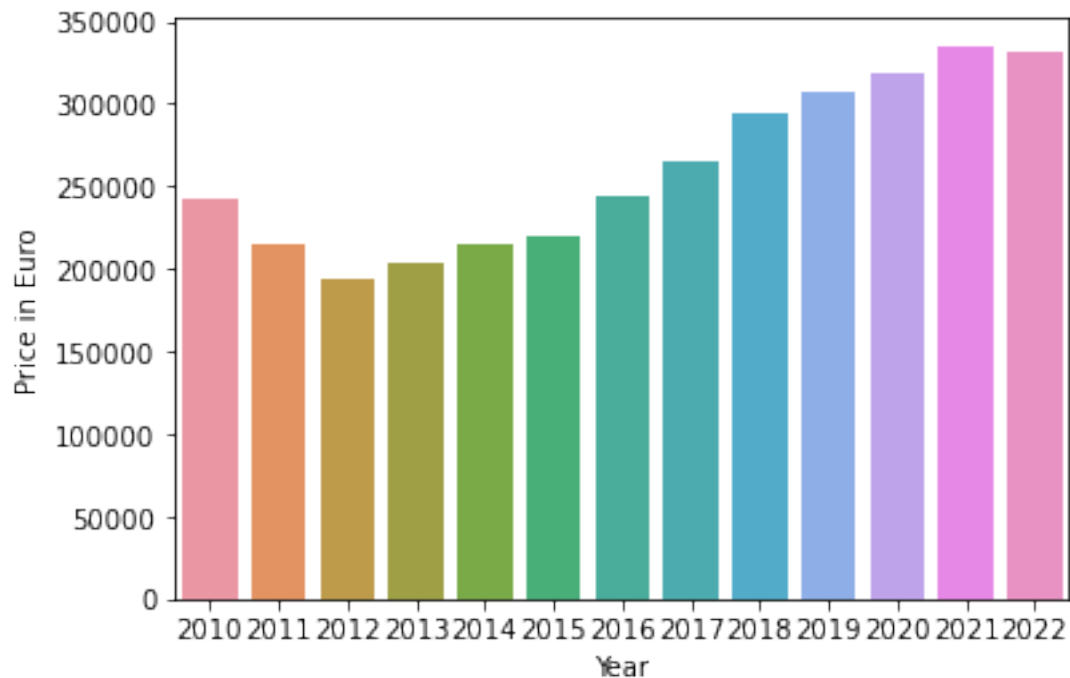


### Bivariate Plots

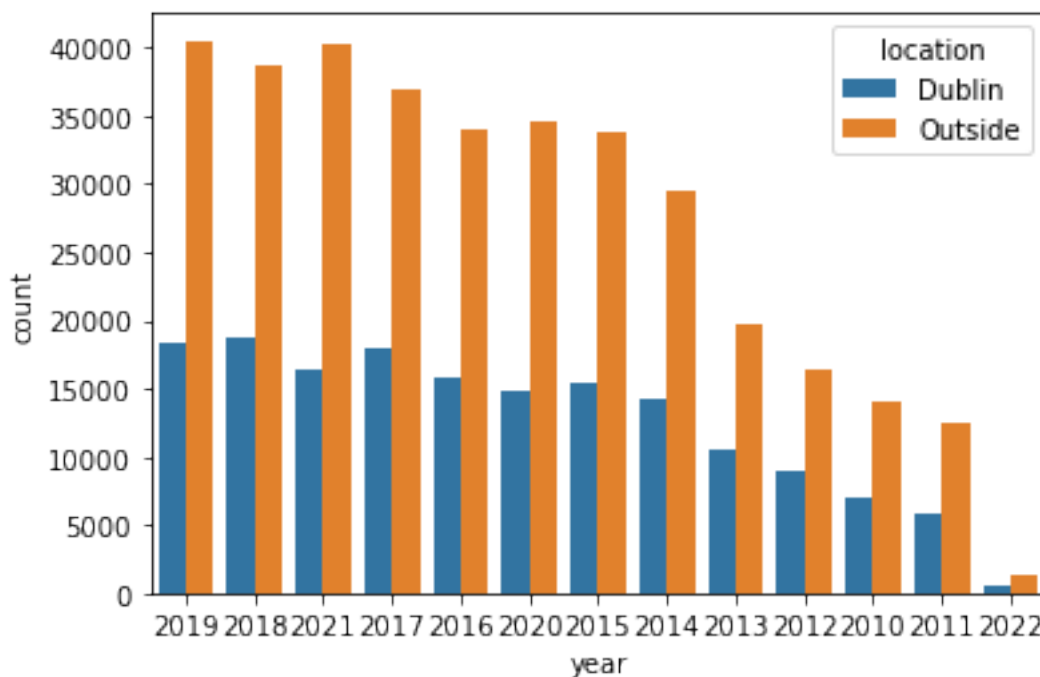
```
sns.boxplot(x='year', y='month', data=df_merge_col)
plt.show()
```



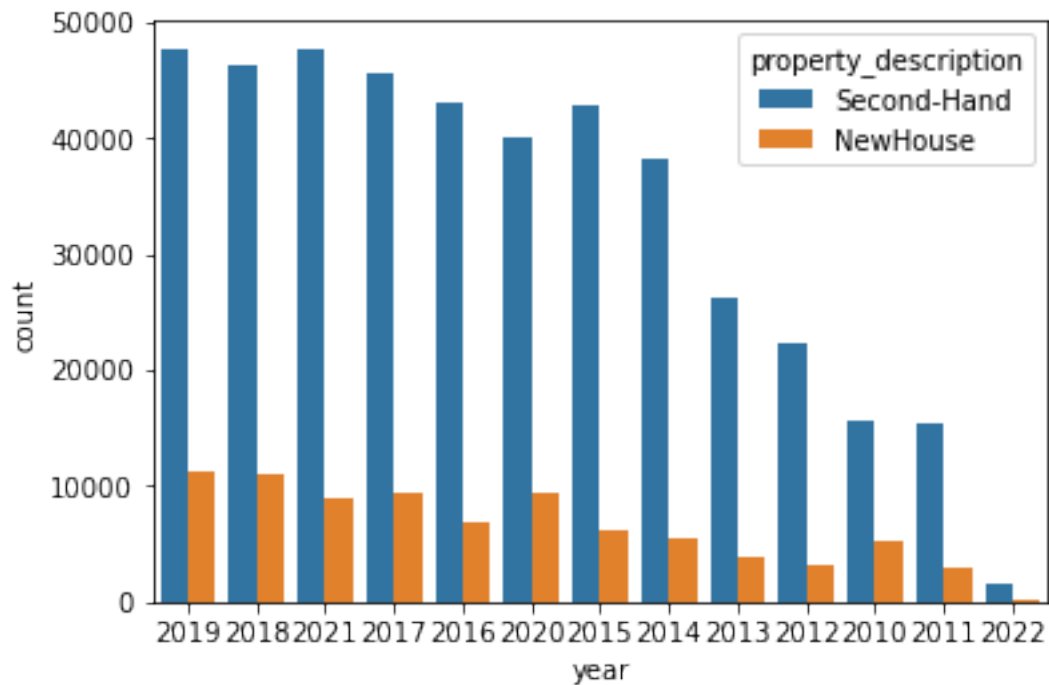
```
sns.barplot(x='year', y='price', data=df_merge_col, ci=None)
plt.xlabel('Year')
plt.ylabel('Price in Euro')
plt.show()
```



```
sns.countplot(x='year',hue='location', data=df_merge_col, order =
df_merge_col['year'].value_counts().index)
plt.show()
```



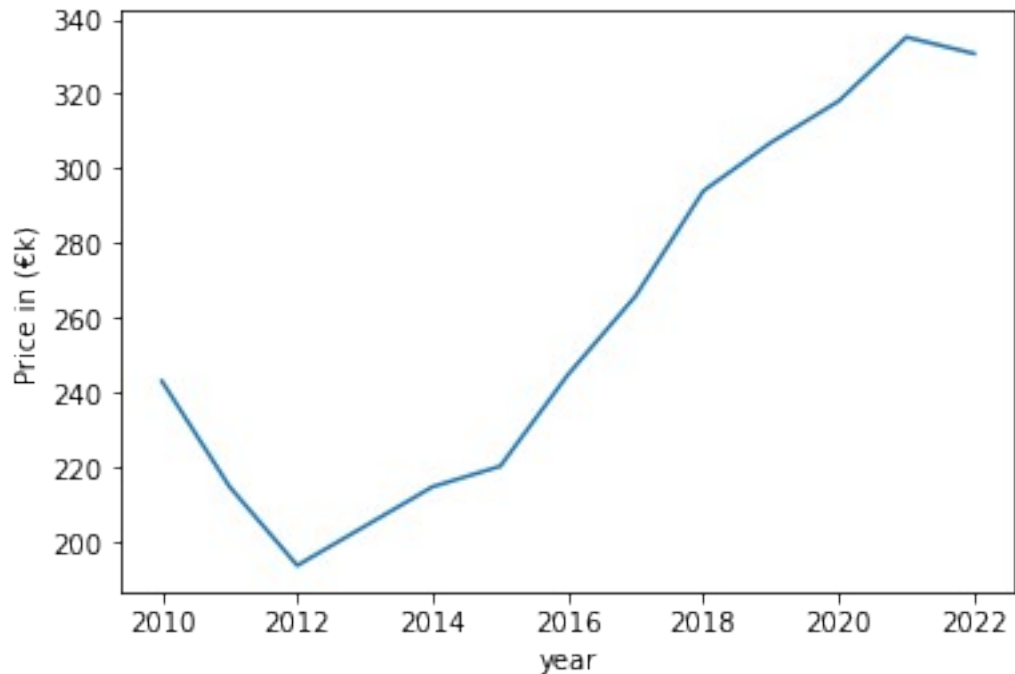
```
sns.countplot(x='year',hue='property_description', data=df_merge_col,
order = df_merge_col['year'].value_counts().index)
plt.show()
```



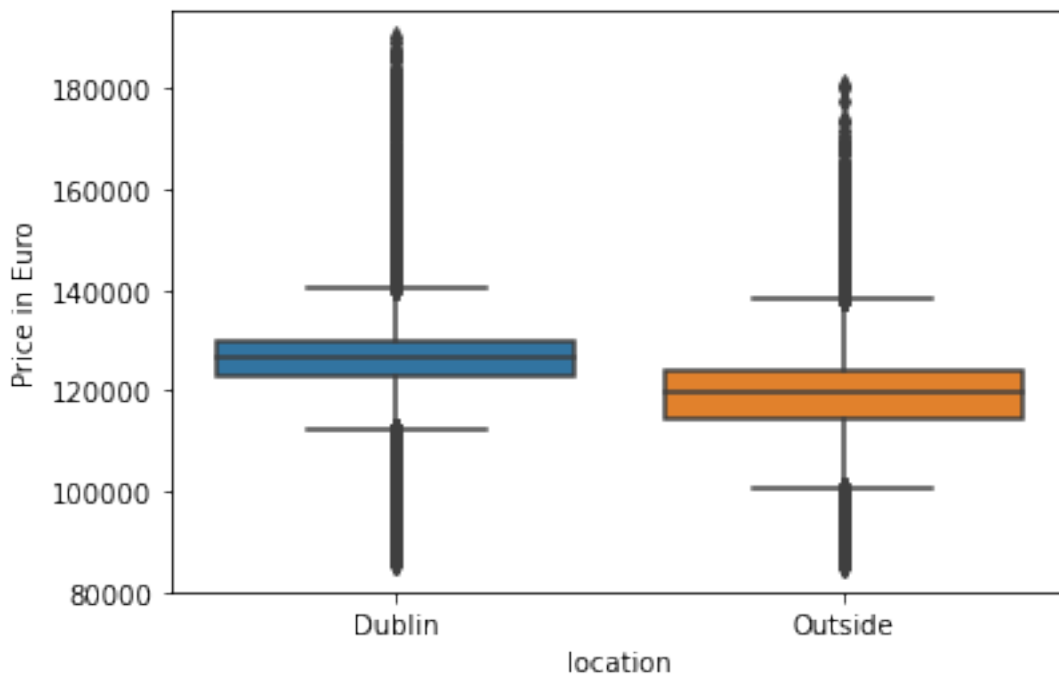
```
sns.countplot(x='year', hue='property_size_description',
data=df_merge_col, order = df_merge_col['year'].value_counts().index)
plt.show()
```



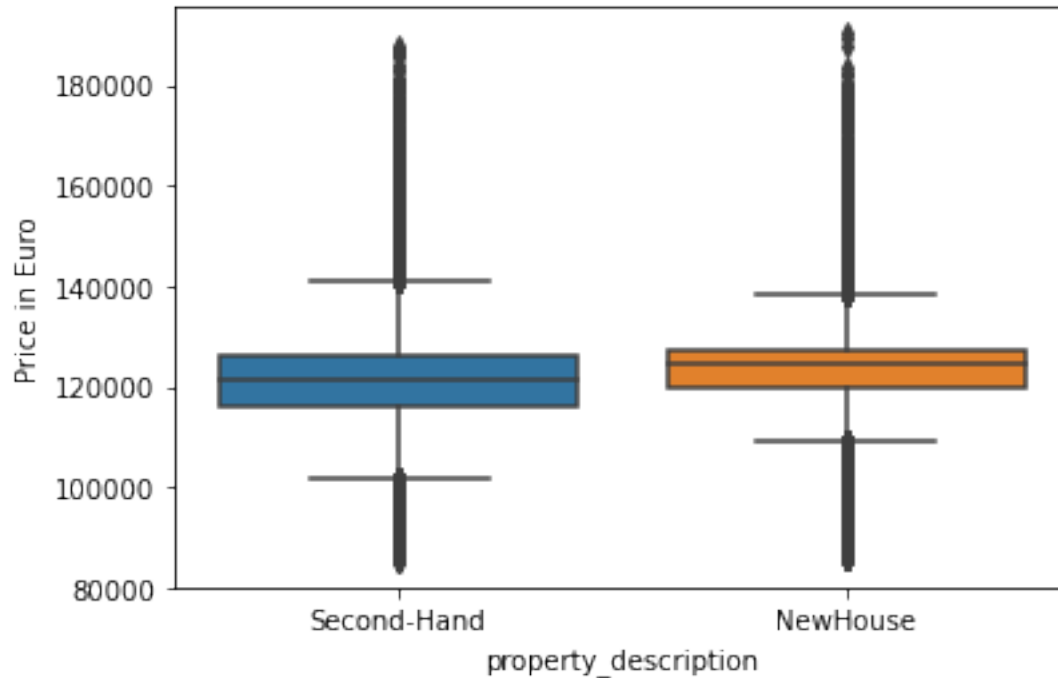
```
sns.lineplot(x='year', y=(df_merge_col['price'])/10**3,
data=df_merge_col, ci=None)
plt.ylabel('Price in (€k)')
plt.show()
```



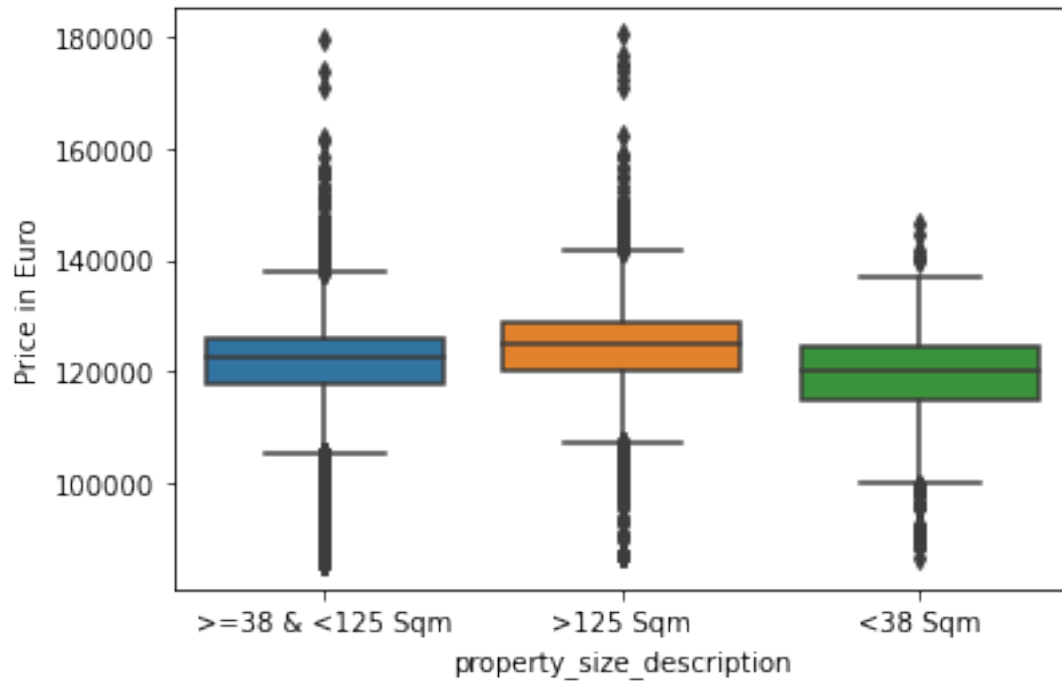
```
# bivariate with Price
#sns.boxplot(x='location', y=np.log(df_merge_col['price']),
#data=df_merge_col)
sns.boxplot(x='location', y=np.log(df_merge_col['price'])*10**4,
data=df_merge_col)
plt.ylabel('Price in Euro')
plt.show()
```



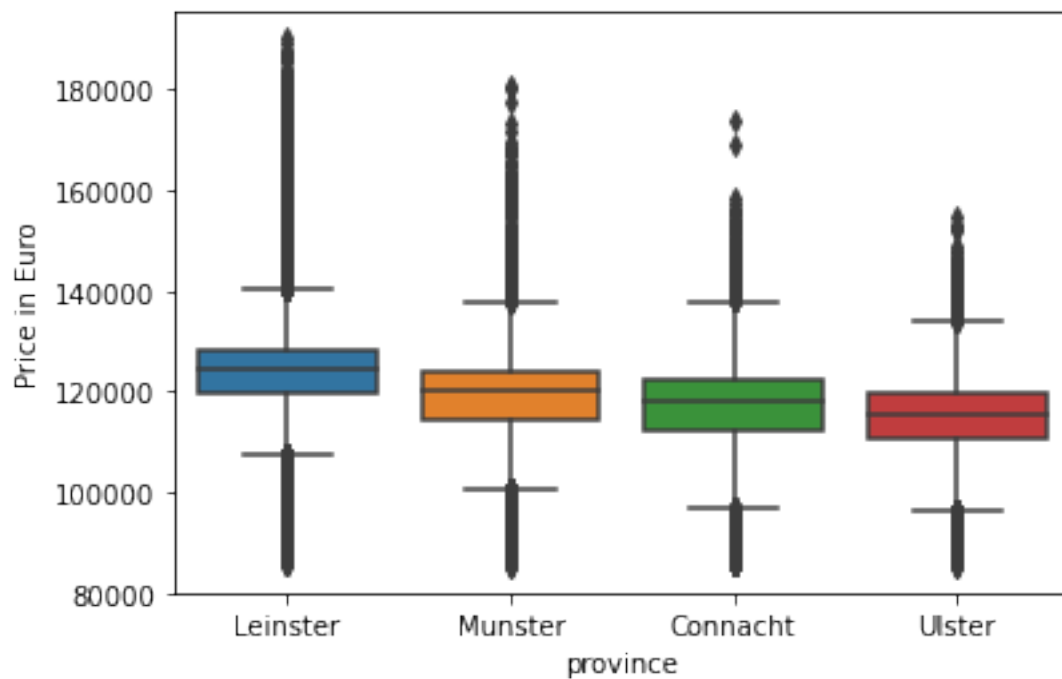
```
sns.boxplot(x='property_description',
y=np.log(df_merge_col['price'])*10**4, data=df_merge_col)
plt.ylabel('Price in Euro')
plt.show()
```



```
sns.boxplot(x='property_size_description',
y=np.log(df_merge_col['price'])*10**4, data=df_merge_col)
plt.ylabel('Price in Euro')
plt.show()
```



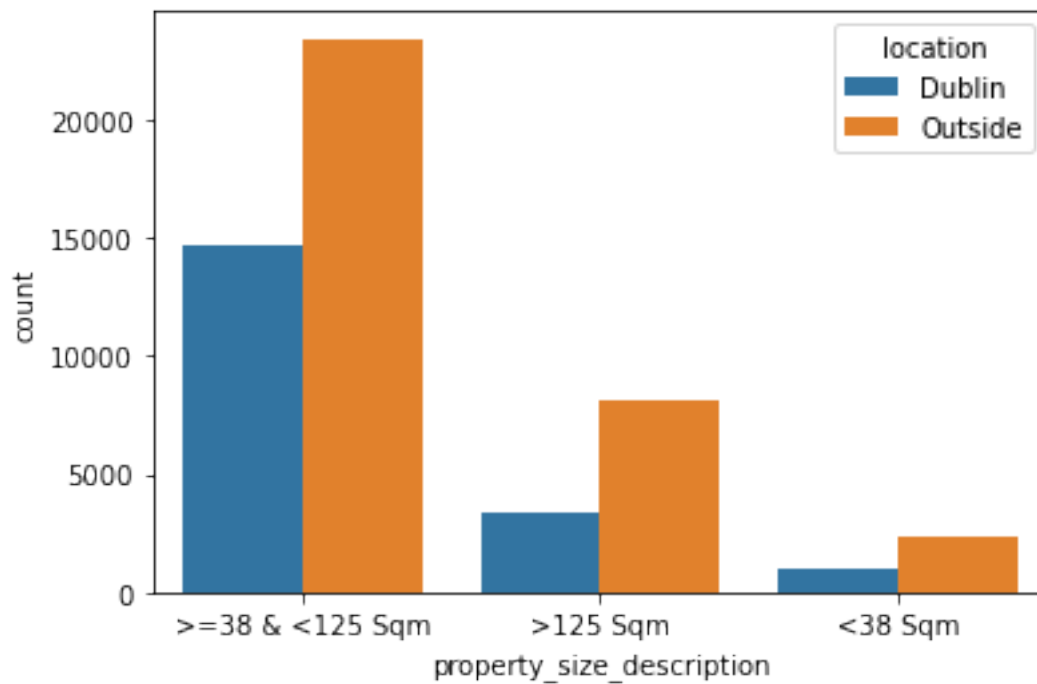
```
sns.boxplot(x='province', y=np.log(df_merge_col['price'])*10**4,
data=df_merge_col)
plt.ylabel('Price in Euro')
plt.show()
```



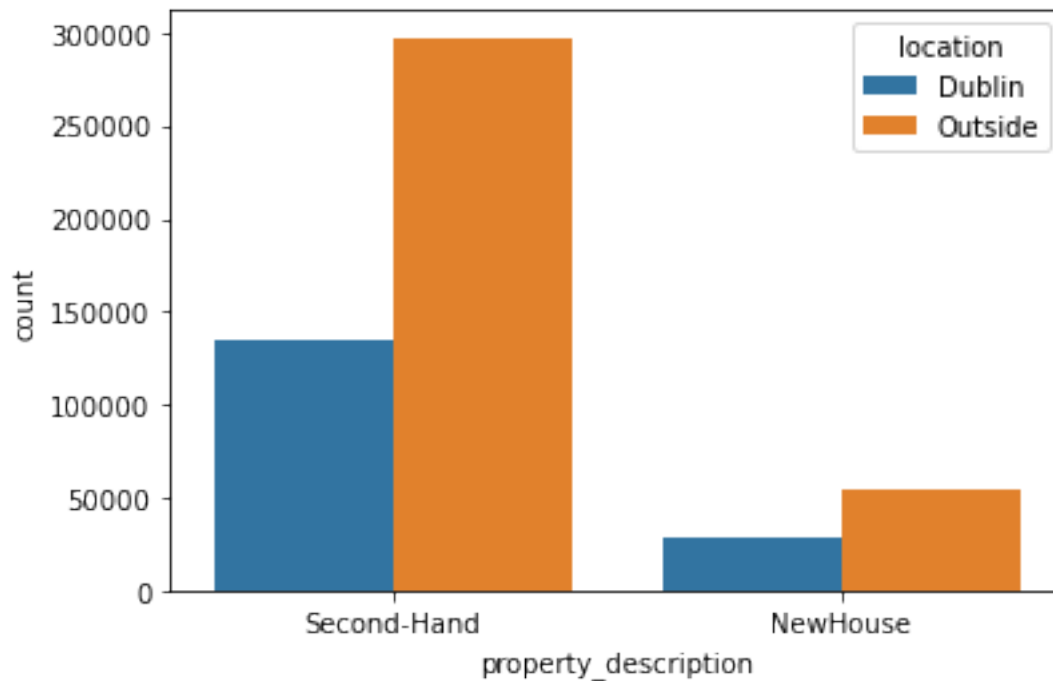
```
# with location
sns.countplot(x='property_size_description', hue='location',
data=df_merge_col, order =
```



```
df_merge_col['property_size_description'].value_counts().index)
plt.show()
```

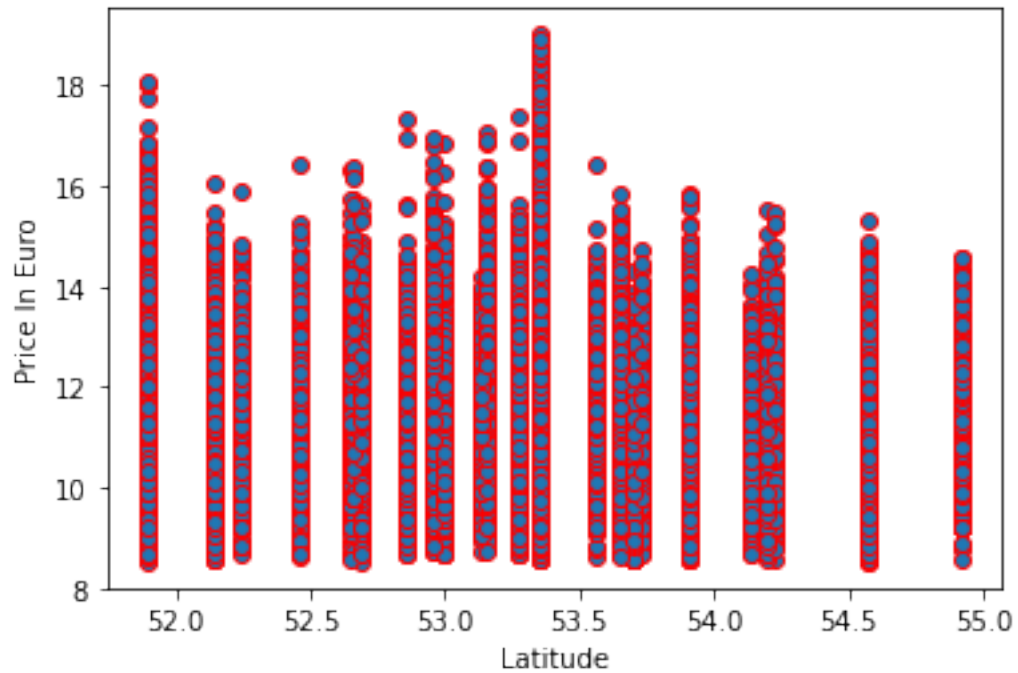


```
sns.countplot(x='property_description', hue='location',
data=df_merge_col, order =
df_merge_col['property_description'].value_counts().index)
plt.show()
```

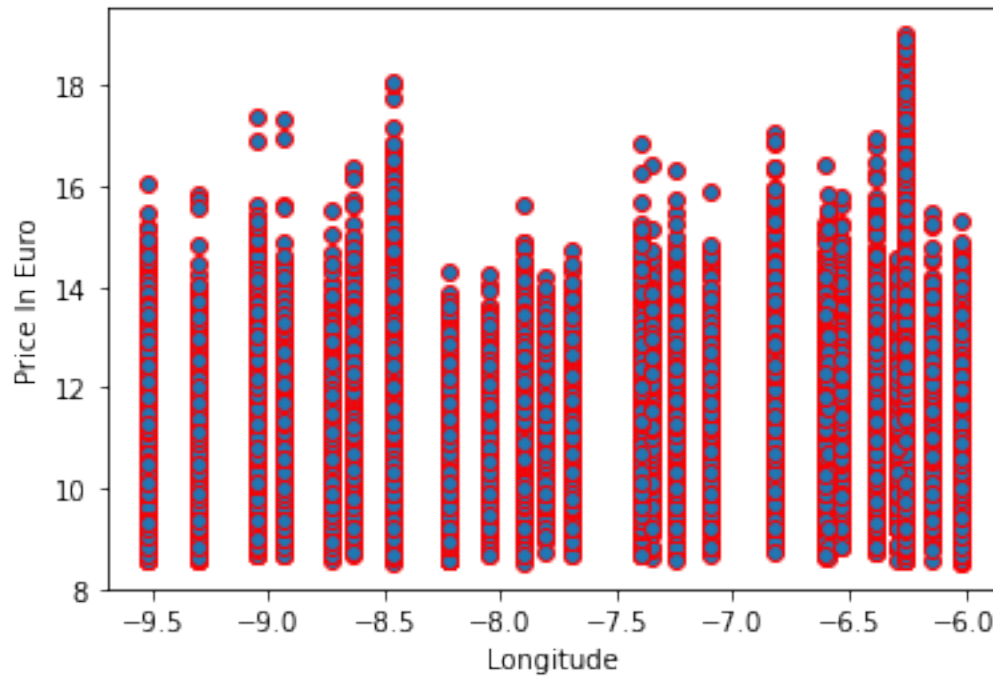


```
# geogrpahy and price
```

```
plt.scatter(df_merge_col['lat'],np.log(df_merge_col['price']) ,edgecol  
ors='r')  
plt.xlabel('Latitude')  
plt.ylabel('Price In Euro')  
plt.show()
```



```
plt.scatter(df_merge_col['lon'],np.log(df_merge_col['price']) ,edgecol  
ors='r')  
plt.xlabel('Longitude')  
plt.ylabel('Price In Euro')  
plt.show()
```

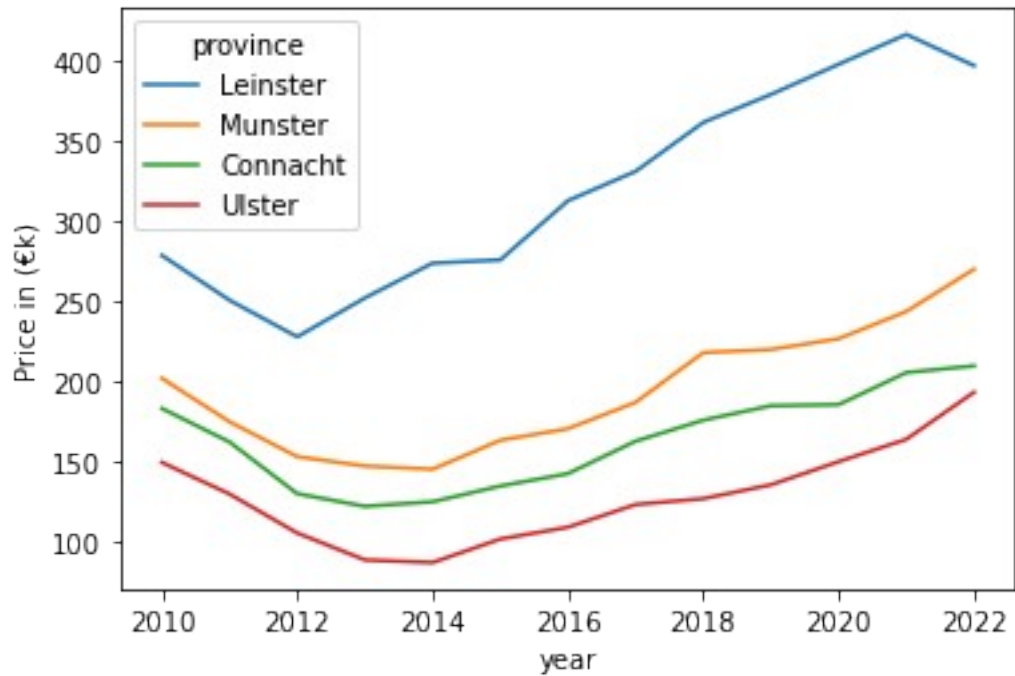


## Multivariate

```
#multivariate
#sns.pairplot(df_merge_col)
#plt.show()

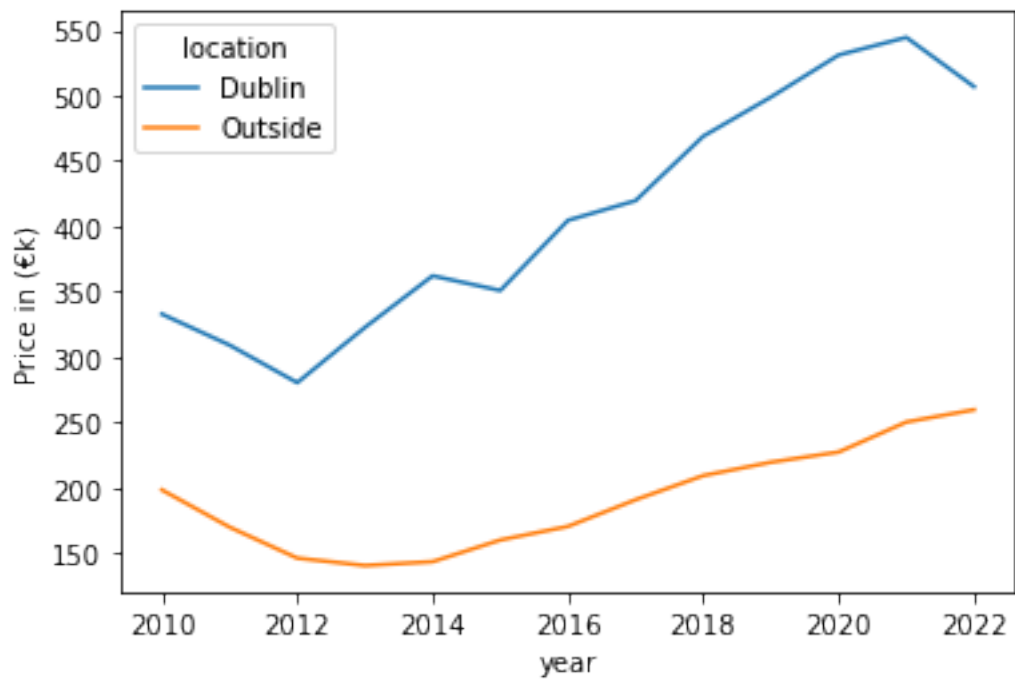
#p1 = sns.lineplot(data=df_merge_col, x="year",
#y=np.log(df_merge_col['price']),hue='province', ci=None)
#p1.set_ylabel("price(log scale)")
#p1.set_yscale('log')
p1 = sns.lineplot(data=df_merge_col, x="year",
y=df_merge_col["price"]/10**3,hue='province', ci=None)
p1.set_ylabel('Price in (€k)')

Text(0, 0.5, 'Price in (€k)')
```



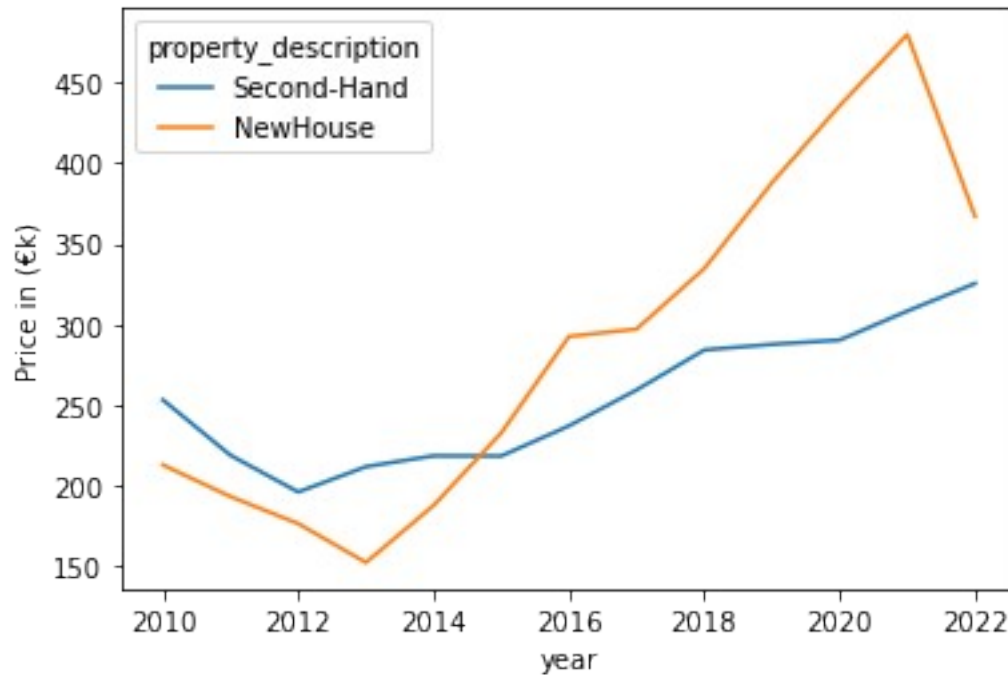
```
p1 = sns.lineplot(data=df_merge_col, x="year",
y=df_merge_col["price"]/10**3,hue='location', ci=None)
p1.set_ylabel('Price in (€k)')
```

```
Text(0, 0.5, 'Price in (€k)')
```

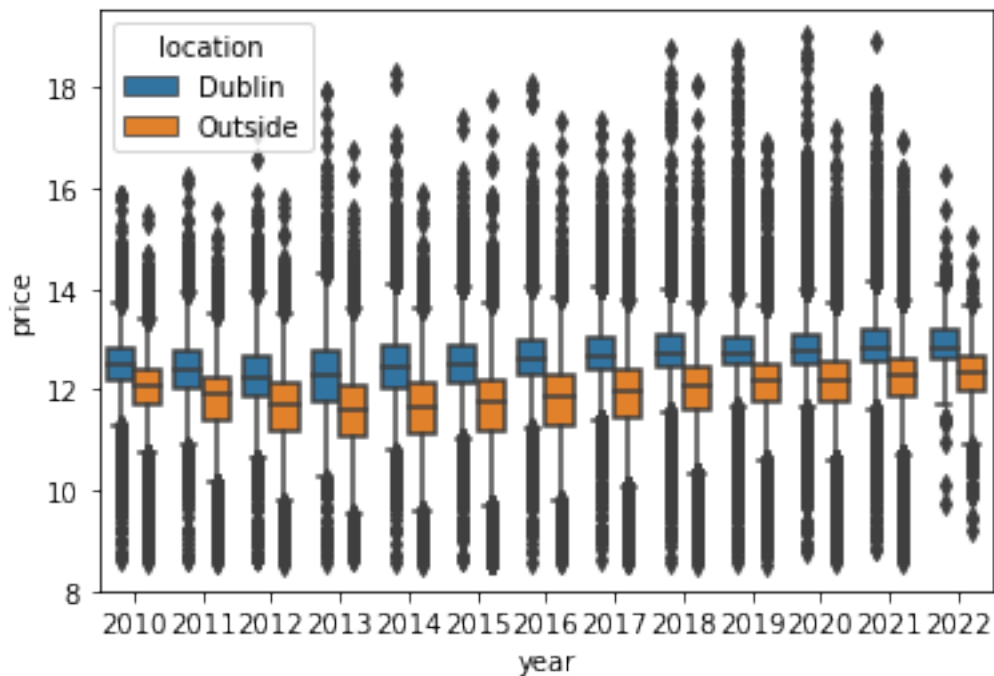


```
p1 = sns.lineplot(data=df_merge_col, x="year",
y=df_merge_col["price"]/10**3,hue='property_description',ci=None)
p1.set_ylabel('Price in (€k)')
```

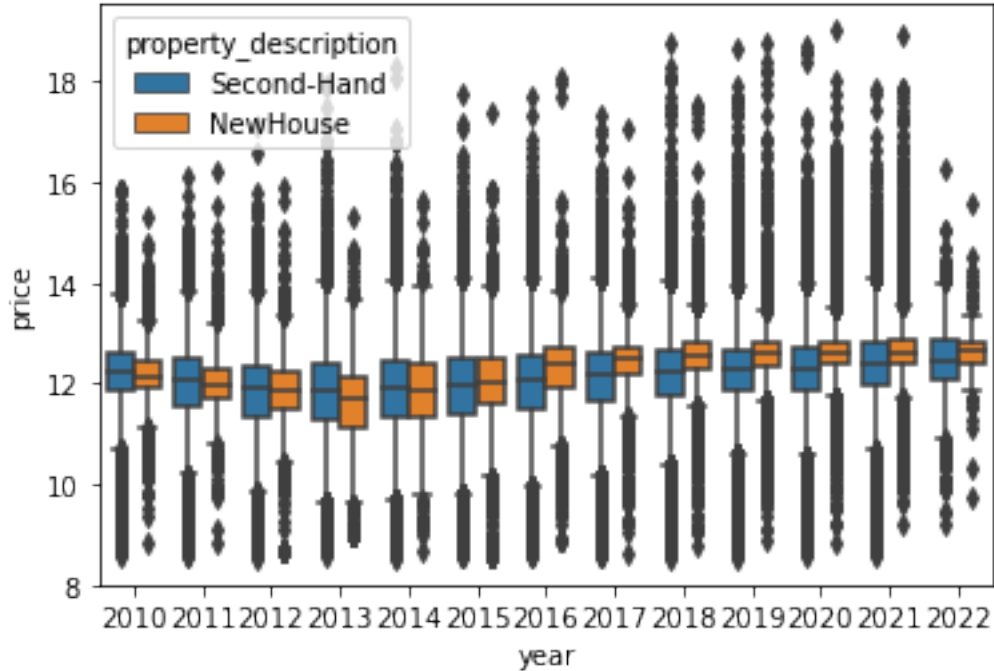
```
Text(0, 0.5, 'Price in (€k)')
```



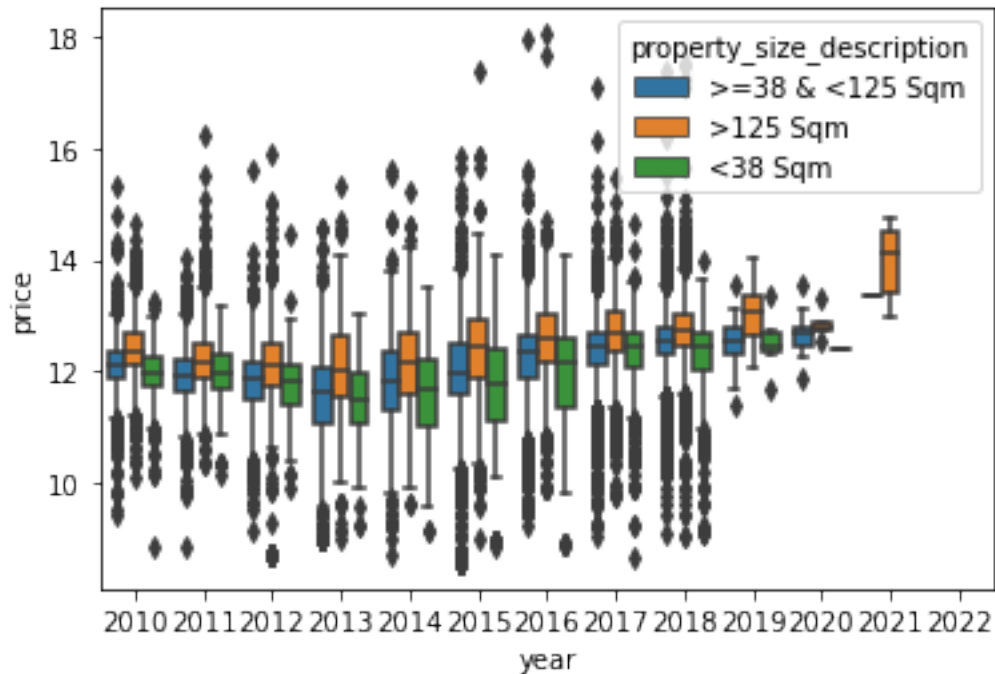
```
sns.boxplot(x='year', y=np.log(df_merge_col['price']), hue='location',
data=df_merge_col)
plt.show()
```



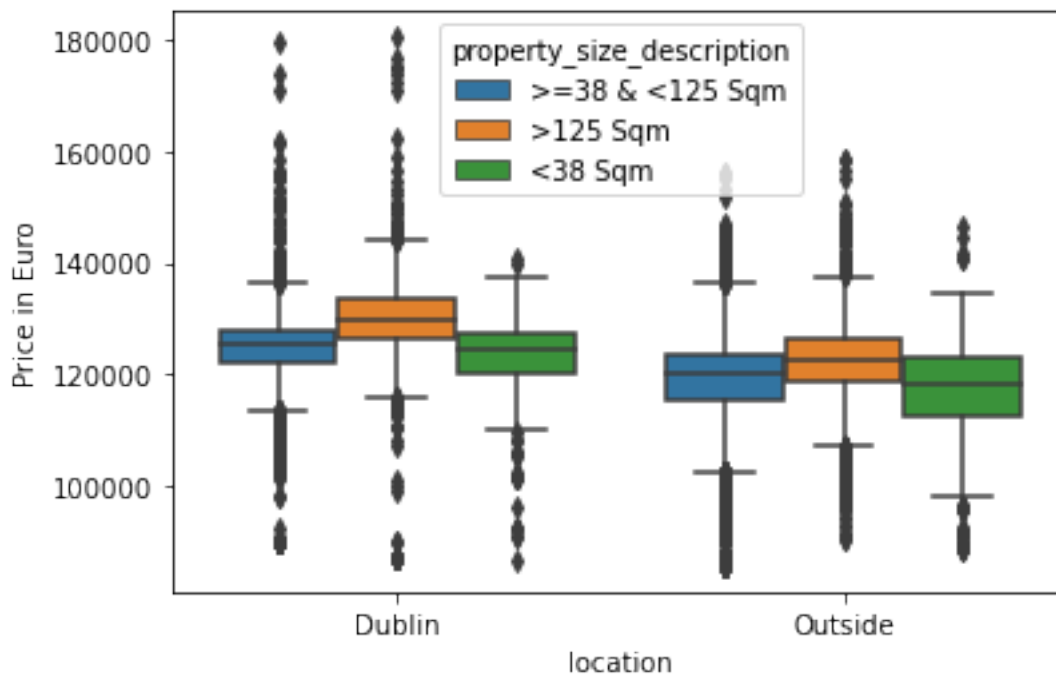
```
sns.boxplot(x='year', y=np.log(df_merge_col['price']),
hue='property_description', data=df_merge_col)
plt.show()
```



```
sns.boxplot(x='year', y=np.log(df_merge_col['price']),
hue='property_size_description', data=df_merge_col)
plt.show()
```

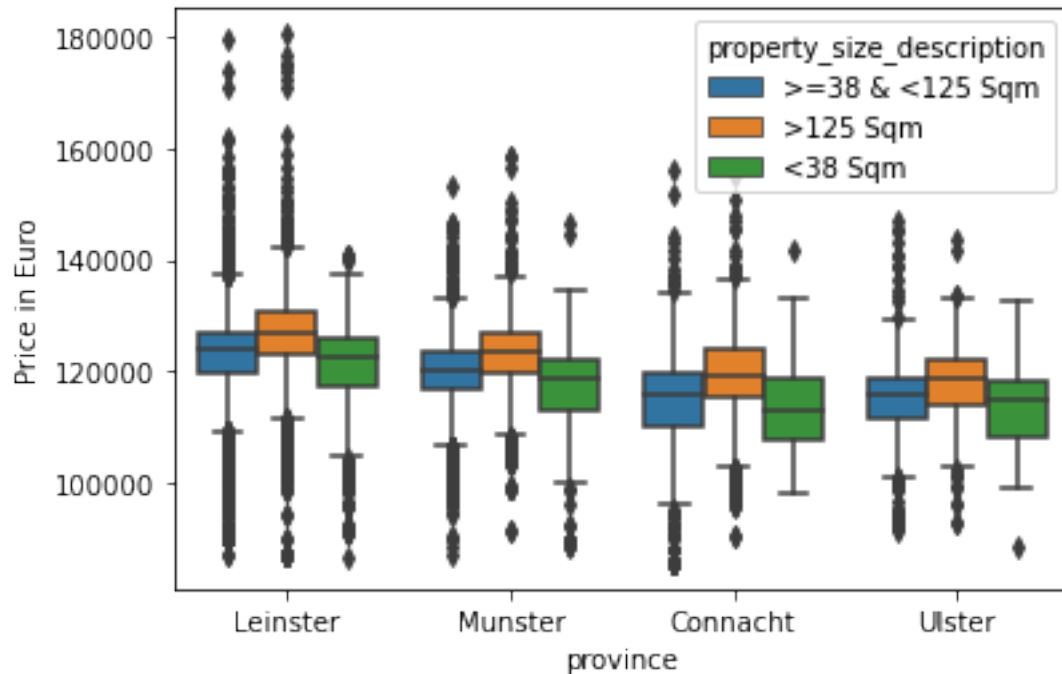


```
p1=sns.boxplot(x='location', y=np.log(df_merge_col['price'])*10**4,
hue='property_size_description', data=df_merge_col)
p1.set_ylabel('Price in Euro')
plt.show()
```

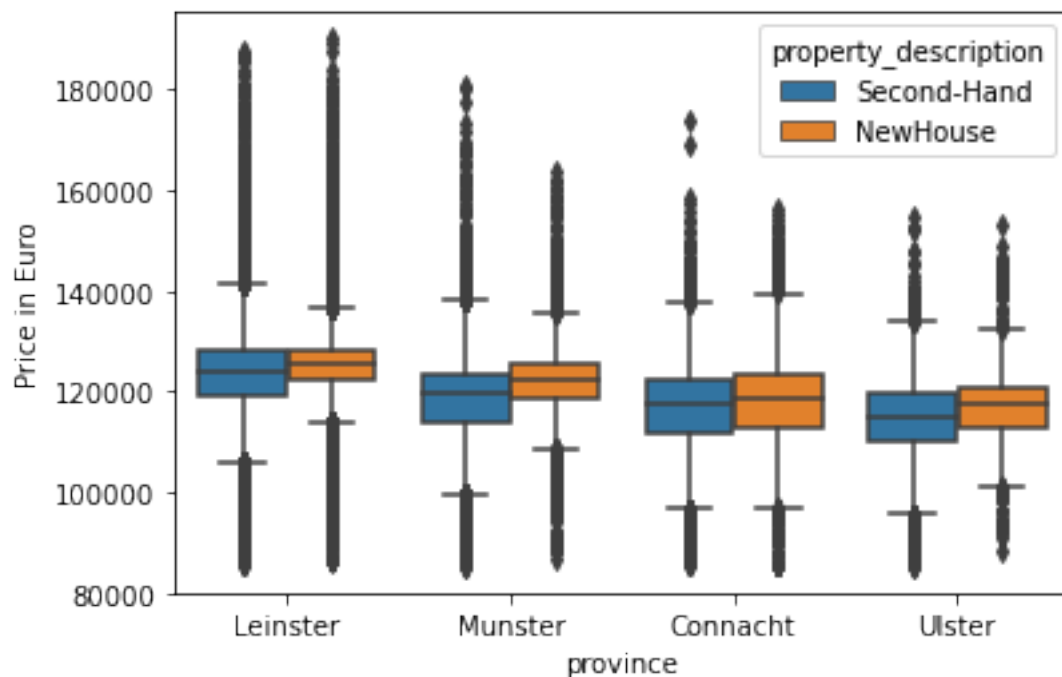


```
p1 =sns.boxplot(x='province', y=np.log(df_merge_col['price'])*10**4,
hue='property_size_description', data=df_merge_col)
```

```
p1.set_ylabel('Price in Euro')
plt.show()
```



```
p1=sns.boxplot(x='province', y=np.log(df_merge_col['price'])*10**4,
hue='property_description', data=df_merge_col)
p1.set_ylabel('Price in Euro')
plt.show()
```

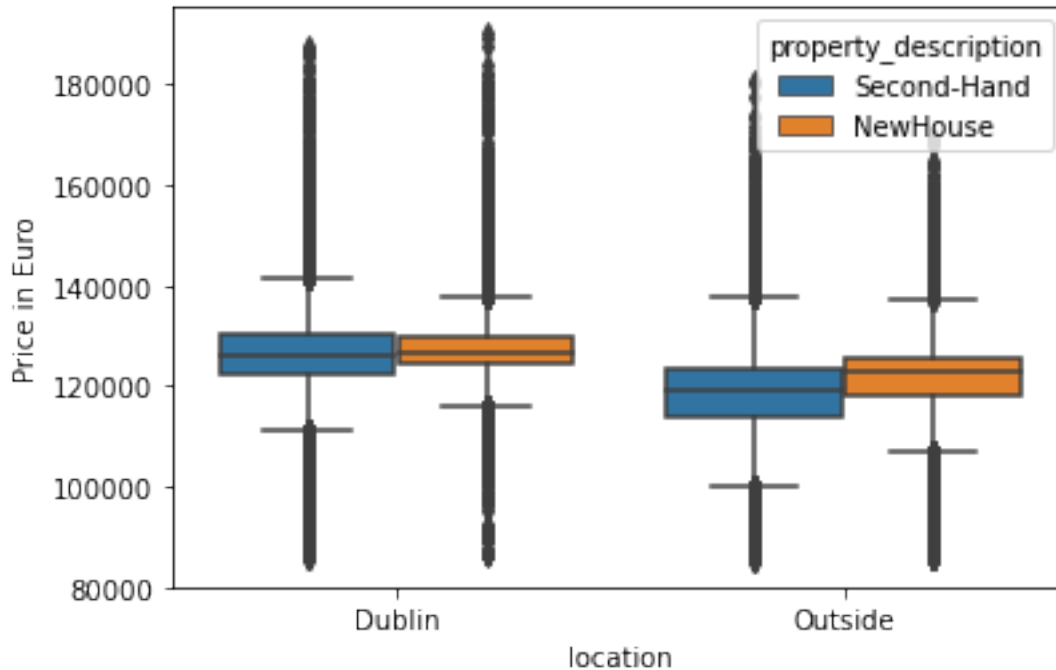




```

p1 = sns.boxplot(x='location', y=np.log(df_merge_col['price'])*10**4,
hue='property_description', data=df_merge_col)
p1.set_ylabel('Price in Euro')
plt.show()

```



```

#df_merge_col.to_csv("PRP.csv", index=False)

```

## Initial Analyses

### MLR

```

rppr1 = df_merge_col.copy()
rppr1.drop(columns
=['month_year', 'date_of_sale', 'address', 'VAT_exclusive', 'FMP', 'postal_
code', 'county'], inplace=True)

```

```

#pd.get_dummies(rppr1["location"])
rppr1["location_Dublin"] = pd.get_dummies(rppr1["location"])["Dublin"]

```

```

#pd.get_dummies(rppr1["property_description"])
rppr1["property_new"] = pd.get_dummies(rppr1["property_description"])
["NewHouse"]

```

```

#pd.get_dummies(rppr1["province"])
#rppr1["provinces_Leinster"] = pd.get_dummies(rppr1["province"])
["Leinster"]
rppr1["provinces_connacht"] = pd.get_dummies(rppr1["province"])
["Connacht"]
rppr1["provinces_Ulster"] = pd.get_dummies(rppr1["province"])["Ulster"]
rppr1["provinces_munster"] = pd.get_dummies(rppr1["province"])
["Munster"]

```

```

from numpy import sqrt
log_price = np.log(rprr1['price'])
transform = sqrt(log_price)

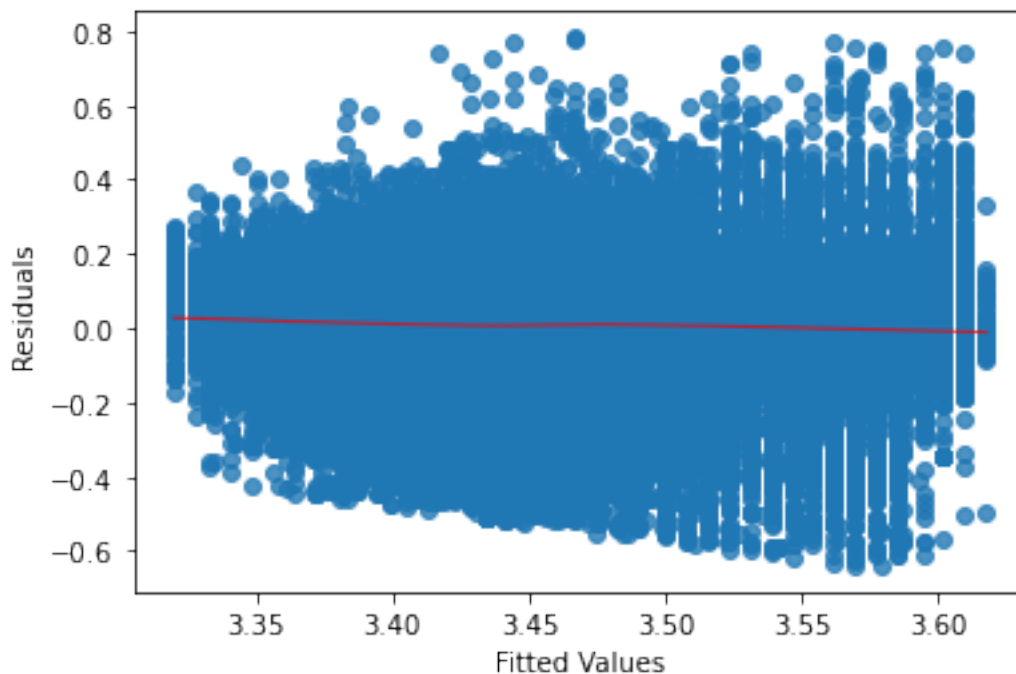
X = rprr1[["location_Dublin","property_new","provinces_connacht",
"provinces_Ulster", "provinces_munster","year", "lat","lon"]]
X = sm.add_constant(X)
y = transform
#X.head(20)

model_full_mlr = sm.OLS(y, X).fit()

#fitted values
model_fitted_vals = model_full_mlr.fittedvalues
#model residuals
model_residuals = model_full_mlr.resid
#standardised residuals
model_norm_residuals =
model_full_mlr.get_influence().resid_studentized_internal

sns.regplot(x=model_fitted_vals,y=model_residuals,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()

```

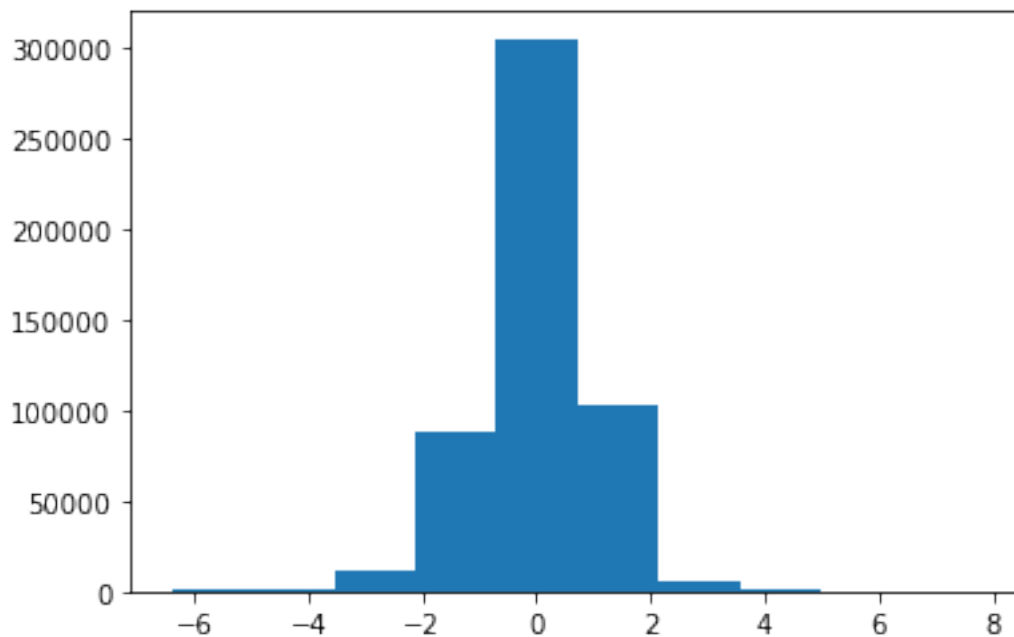
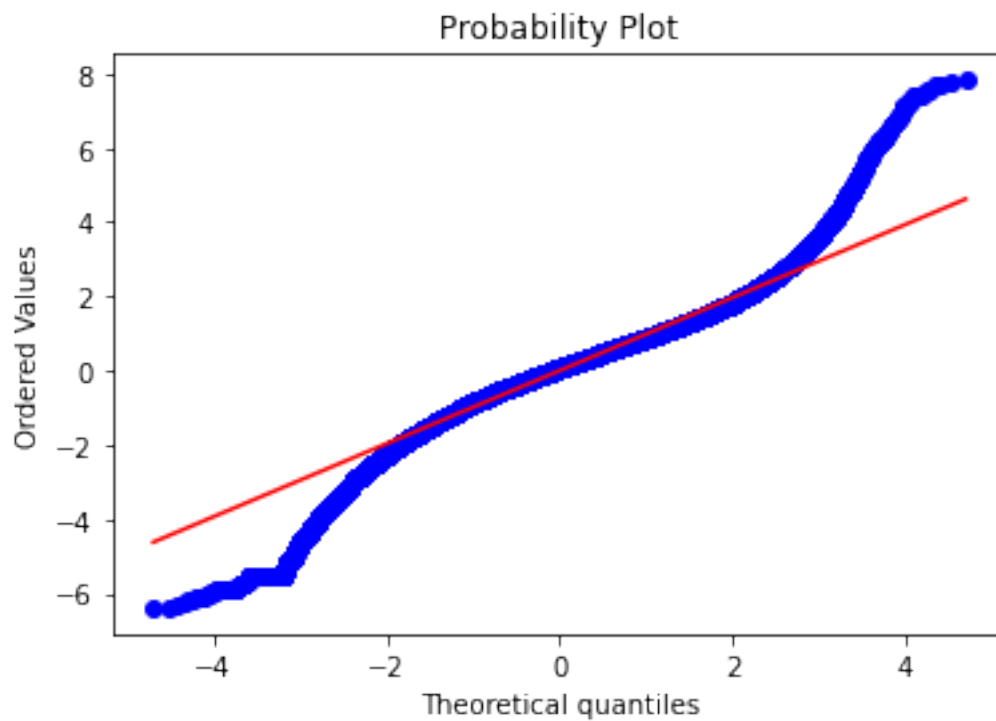


```

stats.probplot(model_norm_residuals, plot=sns.mpl.pyplot)
plt.show()

```

```
plt.hist(model_norm_residuals)
plt.show()
```



```
from statsmodels.formula.api import ols
model_full_mlr1 = ols('log_price ~ C(year)+C(province)
+C(property_description)+lat+lon', data=rppr1).fit()
model_full_mlr1.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
```

# OLS Regression Results

```
=====
Dep. Variable:          log_price    R-squared:
0.228
Model:                  OLS          Adj. R-squared:
0.228
Method:                 Least Squares    F-statistic:
8472.
Date:                  Mon, 05 Sep 2022    Prob (F-statistic):
0.00
Time:                  15:14:06          Log-Likelihood:
5.5626e+05
No. Observations:      516586          AIC:
1.113e+06
Df Residuals:          516567          BIC:
1.113e+06
Df Model:              18

Covariance Type:       nonrobust
```

					coef	std err	
t	P> t	[0.025	0.975]				
-----							
Intercept				21.1564	0.164		
128.865	0.000	20.835	21.478				
C(year)[T.2011]				-0.1552	0.007	-	
21.638	0.000	-0.169	-0.141				
C(year)[T.2012]				-0.3209	0.007	-	
48.358	0.000	-0.334	-0.308				
C(year)[T.2013]				-0.3800	0.006	-	
59.456	0.000	-0.393	-0.368				
C(year)[T.2014]				-0.2768	0.006	-	
46.307	0.000	-0.288	-0.265				
C(year)[T.2015]				-0.2015	0.006	-	
34.342	0.000	-0.213	-0.190				
C(year)[T.2016]				-0.1019	0.006	-	
17.407	0.000	-0.113	-0.090				
C(year)[T.2017]				0.0032	0.006		
0.560	0.576	-0.008	0.015				
C(year)[T.2018]				0.0755	0.006		
13.171	0.000	0.064	0.087				
C(year)[T.2019]				0.1213	0.006		
21.244	0.000	0.110	0.133				

C(year)[T.2020]				0.1523	0.006	
26.007	0.000	0.141	0.164			
C(year)[T.2021]				0.2444	0.006	
42.535	0.000	0.233	0.256			
C(year)[T.2022]				0.3149	0.017	
18.347	0.000	0.281	0.349			
C(province)[T.Leinster]				-0.1093	0.006	-
17.434	0.000	-0.122	-0.097			
C(province)[T.Munster]				-0.1061	0.006	-
18.707	0.000	-0.117	-0.095			
C(province)[T.Ulster]				-0.9669	0.008	-
118.106	0.000	-0.983	-0.951			
C(property_description)[T.Second-Hand]				-0.1311	0.003	-
48.631	0.000	-0.136	-0.126			
lat				-0.1225	0.003	-
40.968	0.000	-0.128	-0.117			
lon				0.3081	0.002	
141.366	0.000	0.304	0.312			

=====

=====

Omnibus:	34379.354	Durbin-Watson:
1.872		
Prob(Omnibus):	0.000	Jarque-Bera (JB):
133193.650		
Skew:	-0.239	Prob(JB):
0.00		
Kurtosis:	5.441	Cond. No.
8.92e+03		

=====

=====

#### Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 8.92e+03. This might indicate that there are strong multicollinearity or other numerical problems.

"""

*#model\_full\_mlr.summary()*

A random sample of 9423 observations is selected from the whole data for statistical analyses.

*# 9423 random sample*

```
rppr_sub = df_merge_col.sample(n=9423, random_state=3)
rppr_sub
```

	date_of_sale	address
91109	10/12/2013	Tainchel, Ashtown Lower, Roundwood

423771	08/05/2020	Greenogue, Kilsallaghan, Co Meath
442477	16/10/2020	6 The Gallops, Coolcots Lane, Wexford
343689	04/12/2018	5 Eden Rd, Birr, Offaly
421939	16/04/2020	69 Castleland Park Way, Balbriggan, County Dublin
...	...	...
404354	11/12/2019	72 St Patricks Rd, Walkinstown Dublin 12, Dublin
218218	05/09/2016	33 Drury Mills, Saggart, Co Dublin
463833	17/02/2021	12 Grand Canal Wood, Allenwood
161348	02/07/2015	The Gables, Kilwogan Lane, Celbridge
499038	30/09/2021	Ardnabourkey, Whitegate, Cork

property_description	postal_code	county	price	FMP	VAT_exclusive
91109	NaN	Wicklow	297500	No	No
Second-Hand					
423771	NaN	Meath	315000	No	No
Second-Hand					
442477	NaN	Wexford	131000	No	No
Second-Hand					
343689	NaN	Offaly	80000	No	No
Second-Hand					
421939	NaN	Dublin	268722	No	Yes
NewHouse					
...	...	...	...	...	...
...					
404354	Dublin 12	Dublin	53750	Yes	No
Second-Hand					
218218	NaN	Dublin	229075	No	Yes
NewHouse					
463833	NaN	Kildare	267000	No	Yes
NewHouse					
161348	NaN	Kildare	470000	No	No
Second-Hand					
499038	NaN	Cork	350000	No	No
Second-Hand					

month_year	property_size_description	province	lat	lon
------------	---------------------------	----------	-----	-----

91109		NaN	Leinster	52.958147	-6.381971
2013-10					
423771		NaN	Leinster	53.649784	-6.588529
2020-08					
442477		NaN	Leinster	52.460187	-6.606516
2020-10					
343689		NaN	Leinster	53.136172	-7.810341
2018-04					
421939		NaN	Leinster	53.349764	-6.260273
2020-04					
...		...	...	...	...
...					
404354		NaN	Leinster	53.349764	-6.260273
2019-11					
218218	>=38 & <125	Sqm	Leinster	53.349764	-6.260273
2016-05					
463833		NaN	Leinster	53.154364	-6.818418
2021-02					
161348		NaN	Leinster	53.154364	-6.818418
2015-02					
499038		NaN	Munster	51.897077	-8.465467
2021-09					

	year	month	location
91109	2013	12	Outside
423771	2020	5	Outside
442477	2020	10	Outside
343689	2018	12	Outside
421939	2020	4	Dublin
...	...	...	...
404354	2019	12	Dublin
218218	2016	9	Dublin
463833	2021	2	Outside
161348	2015	7	Outside
499038	2021	9	Outside

[9423 rows x 16 columns]

Dropping the address, latitude and longitude and date column since it is not using in following analysis.

## ANOVA

A random sample of 9423 obsevrations is seleceted from the whole data for statistical analyses.

```
rppr = rppr_sub.copy()
rppr.drop(columns=['month_year',
'lat', 'lon', 'date_of_sale', 'address', 'VAT_exclusive', 'FMP', 'postal_cod
e', 'county'], inplace=True)
```

```
rppr
```

	price	property_description	property_size_description
province	year	\	
91109	297500	Second-Hand	NaN
Leinster	2013		
423771	315000	Second-Hand	NaN
Leinster	2020		
442477	131000	Second-Hand	NaN
Leinster	2020		
343689	80000	Second-Hand	NaN
Leinster	2018		
421939	268722	NewHouse	NaN
Leinster	2020		
...	...	...	...
...	...		..
404354	53750	Second-Hand	NaN
Leinster	2019		
218218	229075	NewHouse	>=38 & <125 Sqm
Leinster	2016		
463833	267000	NewHouse	NaN
Leinster	2021		
161348	470000	Second-Hand	NaN
Leinster	2015		
499038	350000	Second-Hand	NaN
Munster	2021		

	month	location
91109	12	Outside
423771	5	Outside
442477	10	Outside
343689	12	Outside
421939	4	Dublin
...	...	...
404354	12	Dublin
218218	9	Dublin
463833	2	Outside
161348	7	Outside
499038	9	Outside

```
[9423 rows x 7 columns]
```

```
pd.get_dummies(rppr["location"])
rppr["location_Dublin"]=pd.get_dummies(rppr["location"])["Dublin"]

pd.get_dummies(rppr["property_description"])
rppr["property_new"]=pd.get_dummies(rppr["property_description"])
["NewHouse"]
```

```
#rpp = rppr[['price','year']]
from numpy import sqrt
```



```
log_price = np.log(rprr['price'])
transform = series = sqrt(log_price)
```

```
log_price = np.log(rprr['price'])
```

```
d1= pd.crosstab(index=rppr['location'], columns=rppr["year"],
margins=True)
d1
```

year \ location	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Dublin	121	100	165	195	222	271	268	326	369	328
Outside	243	197	314	374	489	645	577	753	753	764
All	364	297	479	569	711	916	845	1079	1122	1092

year \ location	2021	2022	All
Dublin	290	15	2947
Outside	729	27	6476
All	1019	42	9423

*## property\_size\_description variable has lots of NaN values and hence it is not considered.*

*#perform two-way ANOVA without interaction*

```
model2t = ols('log_price~ C(year) + C(location) +
C(property_description) + C(province)', data=rppr_sub).fit()
```

*#fitted values*

```
model_fitted_vals2 = model2t.fittedvalues
```

*#model residuals*

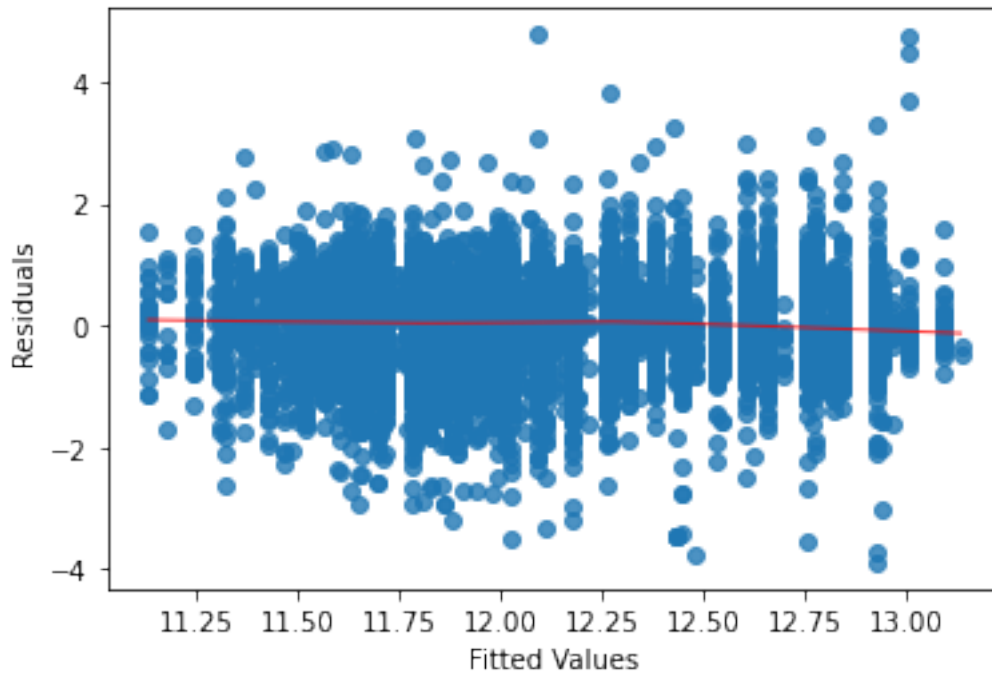
```
model_residuals2 = model2t.resid
```

*#standardised residuals*

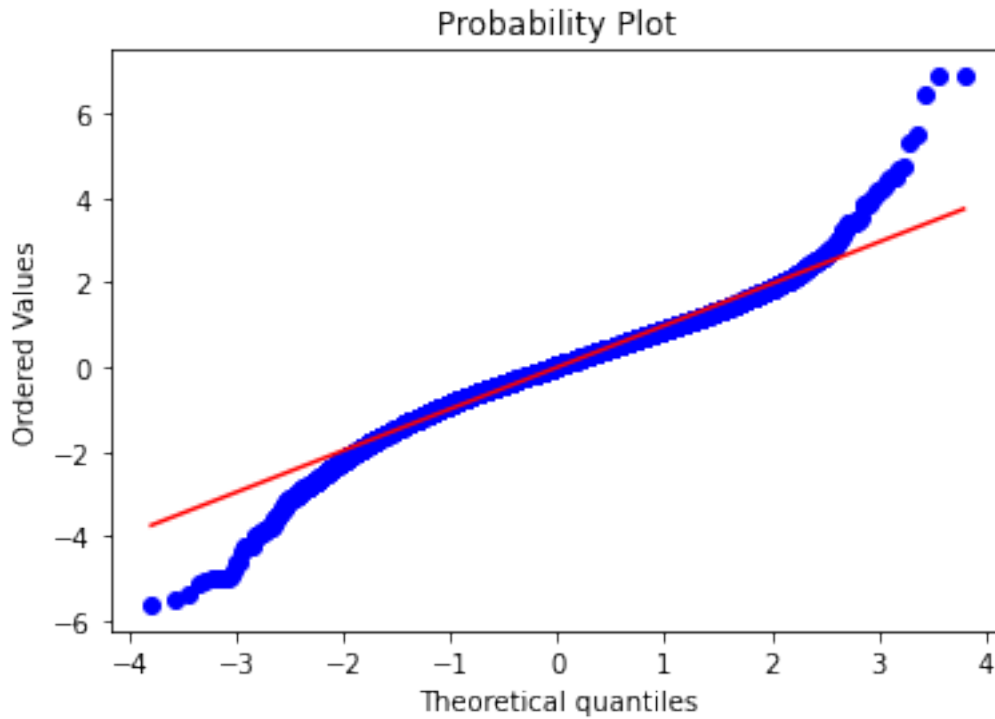
```
model_norm_residuals2t =
```

```
model2t.get_influence().resid_studentized_internal
```

```
sns.regplot(x=model_fitted_vals2,y=model_residuals2,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()
```



```
stats.probplot(model_norm_residuals2t, plot=sns.mpl.pyplot)
plt.show()
```



```
sm.stats.anova_lm(model2t, typ=2)
```

	sum_sq	df	F	
PR(>F)				
C(year)	374.184533	12.0	64.743305	4.008145e-
152				
C(location)	592.285002	1.0	1229.761838	2.604806e-
253				
C(property_description)	35.376031	1.0	73.451281	1.194297e-
17				
C(province)	131.985320	3.0	91.347076	2.880052e-
58				
Residual	4529.690442	9405.0	NaN	
NaN				

```
model2t.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
```

```
"""
```

### OLS Regression Results

```
=====
```

```
=====
Dep. Variable:          log_price    R-squared:
0.282
Model:                  OLS          Adj. R-squared:
0.281
Method:                 Least Squares    F-statistic:
217.6
Date:                  Mon, 05 Sep 2022    Prob (F-statistic):
0.00
Time:                  12:22:54          Log-Likelihood:
-9919.5
No. Observations:      9423             AIC:
1.987e+04
Df Residuals:          9405             BIC:
2.000e+04
Df Model:              17
```

```
Covariance Type:      nonrobust
```

```
=====
```

				coef	std err	
t	P> t	[0.025	0.975]			
-----						
Intercept				12.5072	0.048	
261.978	0.000	12.414	12.601			
C(year) [T.2011]				-0.0936	0.054	-
1.723	0.085	-0.200	0.013			

C(year)[T.2012]				-0.3105	0.048	-
6.424	0.000	-0.405	-0.216			
C(year)[T.2013]				-0.3587	0.047	-
7.686	0.000	-0.450	-0.267			
C(year)[T.2014]				-0.2492	0.045	-
5.561	0.000	-0.337	-0.161			
C(year)[T.2015]				-0.1809	0.043	-
4.200	0.000	-0.265	-0.096			
C(year)[T.2016]				-0.0222	0.044	-
0.509	0.611	-0.108	0.063			
C(year)[T.2017]				0.0294	0.042	
0.699	0.485	-0.053	0.112			
C(year)[T.2018]				0.1305	0.042	
3.116	0.002	0.048	0.213			
C(year)[T.2019]				0.1475	0.042	
3.509	0.000	0.065	0.230			
C(year)[T.2020]				0.2143	0.043	
4.956	0.000	0.130	0.299			
C(year)[T.2021]				0.2978	0.042	
7.012	0.000	0.215	0.381			
C(year)[T.2022]				0.3406	0.113	
3.011	0.003	0.119	0.562			
C(location)[T.Outside]				-0.6631	0.019	-
35.068	0.000	-0.700	-0.626			
C(property_description)[T.Second-Hand]				-0.1662	0.019	-
8.570	0.000	-0.204	-0.128			
C(province)[T.Leinster]				0.2848	0.025	
11.415	0.000	0.236	0.334			
C(province)[T.Munster]				0.1999	0.025	
7.949	0.000	0.151	0.249			
C(province)[T.Ulster]				-0.1873	0.037	-
5.046	0.000	-0.260	-0.115			

=====

=====

Omnibus:	746.293	Durbin-Watson:
1.968		
Prob(Omnibus):	0.000	Jarque-Bera (JB):
3395.607		
Skew:	-0.257	Prob(JB):
0.00		
Kurtosis:	5.896	Cond. No.
32.4		

=====

=====

#### Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

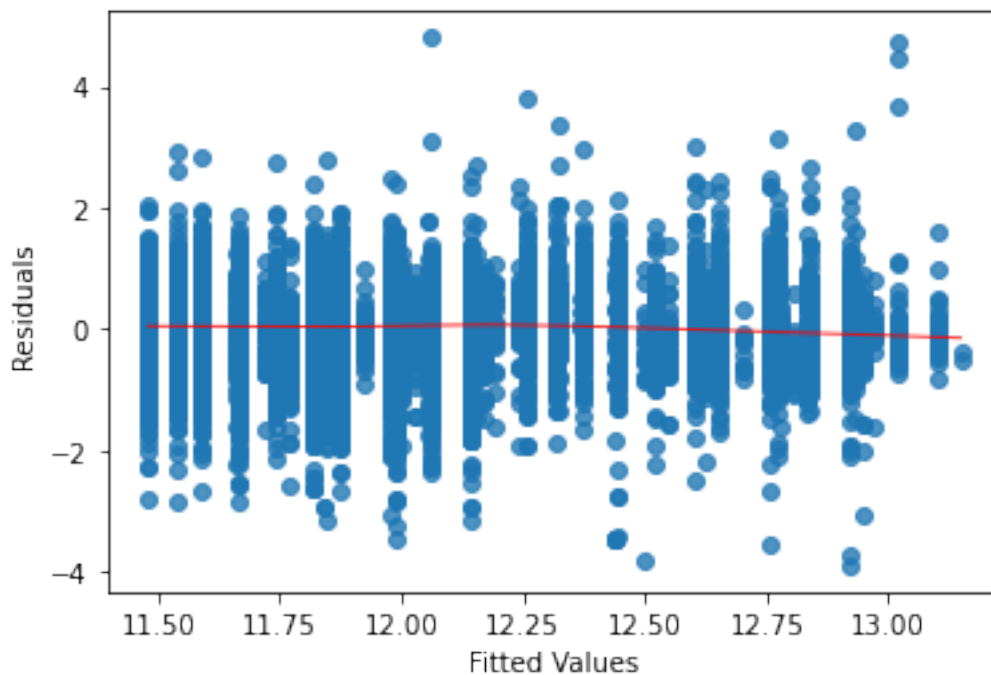
"""

```

#perform two-way ANOVA without interaction
model3 = ols('log_price~ C(year) + C(location) +
C(property_description)',
  data=rppr_sub).fit()
#fitted values
model_fitted_vals3 = model3.fittedvalues
#model residuals
model_residuals3 = model3.resid
#standardised residuals
model_norm_residuals3 =
model3.get_influence().resid_studentized_internal

sns.regplot(x=model_fitted_vals3,y=model_residuals3,
ci=False,lowess=True,
line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.show()

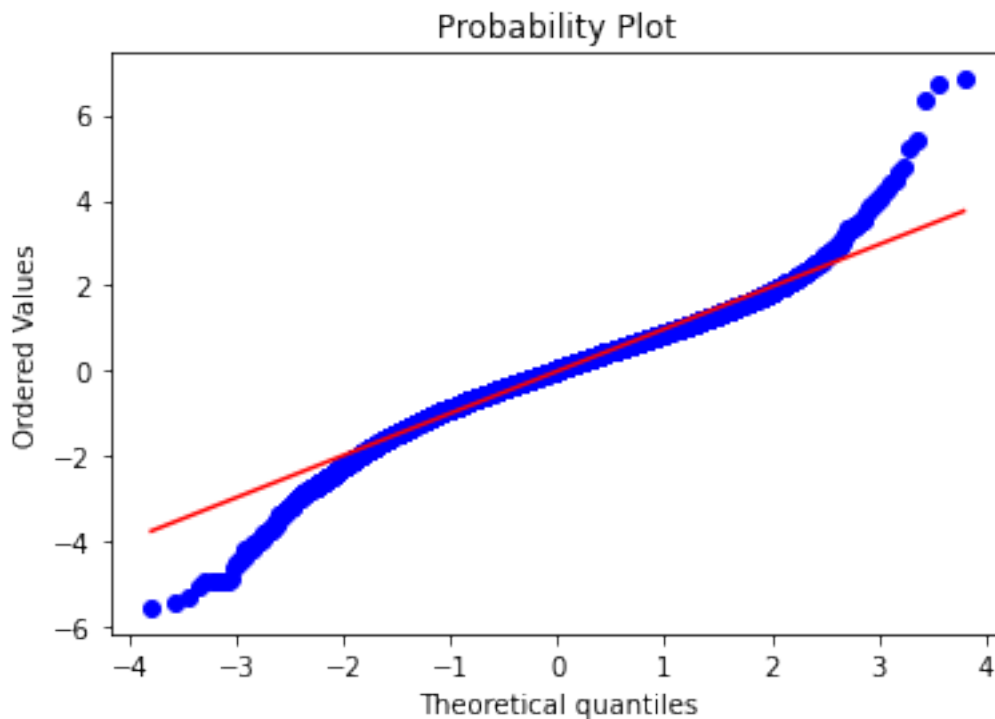
```



```

stats.probplot(model_norm_residuals3, plot=sns.mpl.pyplot)
plt.show()

```



```
model3.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
```

#### OLS Regression Results

```
=====
=====
Dep. Variable:          log_price    R-squared:
0.261
Model:                  OLS          Adj. R-squared:
0.260
Method:                 Least Squares    F-statistic:
237.8
Date:                   Mon, 05 Sep 2022    Prob (F-statistic):
0.00
Time:                   12:22:55          Log-Likelihood:
-10055.
No. Observations:      9423             AIC:
2.014e+04
Df Residuals:          9408             BIC:
2.025e+04
Df Model:              14

Covariance Type:       nonrobust

=====
=====
```

t	P> t	[0.025	0.975]	coef	std err	
-----						
Intercept				12.8049	0.041	
313.873	0.000	12.725	12.885			
C(year)[T.2011]				-0.1028	0.055	-
1.867	0.062	-0.211	0.005			
C(year)[T.2012]				-0.3060	0.049	-
6.242	0.000	-0.402	-0.210			
C(year)[T.2013]				-0.3663	0.047	-
7.738	0.000	-0.459	-0.273			
C(year)[T.2014]				-0.2551	0.045	-
5.612	0.000	-0.344	-0.166			
C(year)[T.2015]				-0.1826	0.044	-
4.179	0.000	-0.268	-0.097			
C(year)[T.2016]				-0.0244	0.044	-
0.552	0.581	-0.111	0.062			
C(year)[T.2017]				0.0271	0.043	
0.635	0.526	-0.057	0.111			
C(year)[T.2018]				0.1294	0.042	
3.047	0.002	0.046	0.213			
C(year)[T.2019]				0.1446	0.043	
3.392	0.001	0.061	0.228			
C(year)[T.2020]				0.2137	0.044	
4.872	0.000	0.128	0.300			
C(year)[T.2021]				0.2957	0.043	
6.867	0.000	0.211	0.380			
C(year)[T.2022]				0.3439	0.115	
2.997	0.003	0.119	0.569			
C(location)[T.Outside]				-0.7802	0.016	-
49.819	0.000	-0.811	-0.749			
C(property_description)[T.Second-Hand]				-0.1791	0.020	-
9.127	0.000	-0.218	-0.141			

=====						
=====						
Omnibus:		725.608		Durbin-Watson:		
1.971						
Prob(Omnibus):		0.000		Jarque-Bera (JB):		
3022.455						
Skew:		-0.280		Prob(JB):		
0.00						
Kurtosis:		5.717		Cond. No.		
30.1						
=====						
=====						

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is

correctly specified.  
"""