Irish PPR Project

April 19, 2022

[1]:
```
# Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#Setting a grid style background
plt.style.use("ggplot")
```

[2]:
```
# Data loading (and setting the sale date as a "datetime" datatype)
property_prices = pd.read_csv("Property_Price_Register_Ireland-28-05-2021.csv.
 ↪zip", parse_dates=["SALE_DATE"])
```

[3]:
```
# Checking the dataframe details
print(property_prices.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 476745 entries, 0 to 476744
Data columns (total 9 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   SALE_DATE           476745 non-null  datetime64[ns]
 1   ADDRESS             476745 non-null  object
 2   POSTAL_CODE         89764 non-null   object
 3   COUNTY              476745 non-null  object
 4   SALE_PRICE          476745 non-null  float64
 5   IF_MARKET_PRICE     476745 non-null  int64
 6   IF_VAT_EXCLUDED     476745 non-null  int64
 7   PROPERTY_DESC       476745 non-null  object
 8   PROPERTY_SIZE_DESC  52792 non-null   object
dtypes: datetime64[ns](1), float64(1), int64(2), object(5)
memory usage: 32.7+ MB
None
```

[4]:
```
# Viewing the first few lines
print(property_prices.head())
```

```
   SALE_DATE                                    ADDRESS POSTAL_CODE  \
```

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

     COUNTY  SALE_PRICE  IF_MARKET_PRICE  IF_VAT_EXCLUDED  \
0    Dublin    343000.0                0                0
1     Laois    185000.0                0                1
2    Dublin    438500.0                0                0
3     Meath    400000.0                0                0
4  Kilkenny    160000.0                0                0

                           PROPERTY_DESC  \
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

                                PROPERTY_SIZE_DESC
0                                              NaN
1  greater than or equal to 38 sq metres and less…
2                                              NaN
3                                              NaN
4                                              NaN
```

[5]:
```
# Check for duplicate rows
property_prices.duplicated().sum()
```

[5]:
```
763
```

[6]:
```
# View detail of the duplicate items
property_prices[property_prices.duplicated(keep=False)]
```

[6]:
```
         SALE_DATE                                  ADDRESS POSTAL_CODE  \
103     2010-01-06        Curragraigue, Blennerville, Tralee         NaN
104     2010-01-06        Curragraigue, Blennerville, Tralee         NaN
175     2010-01-08         58 Castlewood Park, Little Island         NaN
176     2010-01-08         58 Castlewood Park, Little Island         NaN
471     2010-01-15                   Old Road, Monasterboice         NaN
…           …                                          …          …
475546 2021-05-13    DONAGHERS LANE, CURRYHILLS, PROSPEROUS         NaN
475919 2021-05-17  UPPER BRIDGE STREET, SKIBBEREEN, CO CORK         NaN
475920 2021-05-17  UPPER BRIDGE STREET, SKIBBEREEN, CO CORK         NaN
476628 2021-05-26       CLOGHERNOOSH, BLACKVALLEY, BEAUFORT         NaN
476629 2021-05-26       CLOGHERNOOSH, BLACKVALLEY, BEAUFORT         NaN
```

```
         COUNTY  SALE_PRICE  IF_MARKET_PRICE  IF_VAT_EXCLUDED  \
103       Kerry    265000.0                0                0
104       Kerry    265000.0                0                0
175        Cork    250000.0                0                0
176        Cork    250000.0                0                0
471       Louth    324000.0                0                0
…           …          …               …              …
475546  Kildare    175000.0                0                0
475919     Cork     17500.0                1                0
475920     Cork     17500.0                1                0
476628    Kerry     10000.0                0                0
476629    Kerry     10000.0                0                0

                                PROPERTY_DESC PROPERTY_SIZE_DESC
103     Second-Hand Dwelling house /Apartment                NaN
104     Second-Hand Dwelling house /Apartment                NaN
175     Second-Hand Dwelling house /Apartment                NaN
176     Second-Hand Dwelling house /Apartment                NaN
471     Second-Hand Dwelling house /Apartment                NaN
…                                         …                …
475546  Second-Hand Dwelling house /Apartment                NaN
475919  Second-Hand Dwelling house /Apartment                NaN
475920  Second-Hand Dwelling house /Apartment                NaN
476628  Second-Hand Dwelling house /Apartment                NaN
476629  Second-Hand Dwelling house /Apartment                NaN

[1431 rows x 9 columns]
```

[7]:
```
# Dropping duplicate rows and saving as a new dataframe
property= property_prices.drop_duplicates()
```

[8]:
```
# Check number of duplicates in new dataframe
property.duplicated().sum()
```

[8]:
```
0
```

[9]:
```
# Checking the number of missing values in each column
property.isnull().sum()
```

[9]:
```
SALE_DATE                0
ADDRESS                  0
POSTAL_CODE         386471
COUNTY                   0
SALE_PRICE               0
IF_MARKET_PRICE          0
IF_VAT_EXCLUDED          0
```

```
PROPERTY_DESC              0
PROPERTY_SIZE_DESC    423427
dtype: int64
```

[10]:
```
# Filling null values in postal code column
property["POSTAL_CODE"].fillna("No data", inplace = True)
```

[11]:
```
# Check the percentage of values that are still missing
property.isnull().mean() * 100
```

[11]:
```
SALE_DATE              0.000000
ADDRESS                0.000000
POSTAL_CODE            0.000000
COUNTY                 0.000000
SALE_PRICE             0.000000
IF_MARKET_PRICE        0.000000
IF_VAT_EXCLUDED        0.000000
PROPERTY_DESC          0.000000
PROPERTY_SIZE_DESC    88.958616
dtype: float64
```

[12]:
```
# Dropping the "Property Size" column as more than 88% of data is missing
property.drop("PROPERTY_SIZE_DESC", inplace=True, axis=1)
```

[13]:
```
# Checking we now should have 8 columns instead of 9
print(property.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 475982 entries, 0 to 476744
Data columns (total 8 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   SALE_DATE        475982 non-null  datetime64[ns]
 1   ADDRESS          475982 non-null  object
 2   POSTAL_CODE      475982 non-null  object
 3   COUNTY           475982 non-null  object
 4   SALE_PRICE       475982 non-null  float64
 5   IF_MARKET_PRICE  475982 non-null  int64
 6   IF_VAT_EXCLUDED  475982 non-null  int64
 7   PROPERTY_DESC    475982 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(2), object(4)
memory usage: 32.7+ MB
None
```

[14]:
```
# Adding the month column and abbreviating the first 3 letters
property['MONTH'] = property['SALE_DATE'].dt.month_name().str[:3]
```

[15]:
```
# Add new column for year
property['YEAR'] = property['SALE_DATE'].dt.year
```

[16]:
```
# Checking changes to columns
print(property.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 475982 entries, 0 to 476744
Data columns (total 10 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   SALE_DATE        475982 non-null  datetime64[ns]
 1   ADDRESS          475982 non-null  object
 2   POSTAL_CODE      475982 non-null  object
 3   COUNTY           475982 non-null  object
 4   SALE_PRICE       475982 non-null  float64
 5   IF_MARKET_PRICE  475982 non-null  int64
 6   IF_VAT_EXCLUDED  475982 non-null  int64
 7   PROPERTY_DESC    475982 non-null  object
 8   MONTH            475982 non-null  object
 9   YEAR             475982 non-null  int64
dtypes: datetime64[ns](1), float64(1), int64(3), object(5)
memory usage: 39.9+ MB
None
```

[17]:
```
# Rearranging the columns
property = property[['SALE_DATE', 'MONTH', 'YEAR', 'ADDRESS', 'COUNTY',␣
 ↪'POSTAL_CODE','SALE_PRICE','IF_MARKET_PRICE',␣
 ↪'IF_VAT_EXCLUDED','PROPERTY_DESC']]
```

[18]:
```
# Checking changes
print(property.head())
```

```
   SALE_DATE MONTH  YEAR                                          ADDRESS  \
0 2010-01-01   Jan  2010          5 Braemor Drive, Churchtown, Co.Dublin
1 2010-01-03   Jan  2010  134 Ashewood Walk, Summerhill Lane, Portlaoise
2 2010-01-04   Jan  2010             1 Meadow Avenue, Dundrum, Dublin 14
3 2010-01-04   Jan  2010                          1 The Haven, Mornington
4 2010-01-04   Jan  2010                   11 Melville Heights, Kilkenny

     COUNTY POSTAL_CODE  SALE_PRICE  IF_MARKET_PRICE  IF_VAT_EXCLUDED  \
0    Dublin     No data    343000.0                0                0
1     Laois     No data    185000.0                0                1
2    Dublin     No data    438500.0                0                0
3     Meath     No data    400000.0                0                0
4  Kilkenny     No data    160000.0                0                0

                        PROPERTY_DESC
```

```
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
```

[19]:
```
# Replacing 1 and 0 with yes and no (for market price and vat excluded columns)
property.IF_VAT_EXCLUDED.replace((1, 0), ("Yes", "No"), inplace=True)
property.IF_MARKET_PRICE.replace((1, 0), ("Yes", "No"), inplace=True)
```

[20]:
```
# Checking changes
print(property.head())
```

```
   SALE_DATE MONTH  YEAR                                          ADDRESS  \
0 2010-01-01   Jan  2010          5 Braemor Drive, Churchtown, Co.Dublin
1 2010-01-03   Jan  2010  134 Ashewood Walk, Summerhill Lane, Portlaoise
2 2010-01-04   Jan  2010             1 Meadow Avenue, Dundrum, Dublin 14
3 2010-01-04   Jan  2010                          1 The Haven, Mornington
4 2010-01-04   Jan  2010                 11 Melville Heights, Kilkenny

     COUNTY POSTAL_CODE  SALE_PRICE IF_MARKET_PRICE IF_VAT_EXCLUDED  \
0    Dublin     No data    343000.0              No              No
1     Laois     No data    185000.0              No             Yes
2    Dublin     No data    438500.0              No              No
3     Meath     No data    400000.0              No              No
4  Kilkenny     No data    160000.0              No              No

                          PROPERTY_DESC
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
```

[21]:
```
# Checking changes & data types
print(property.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 475982 entries, 0 to 476744
Data columns (total 10 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   SALE_DATE        475982 non-null  datetime64[ns]
 1   MONTH            475982 non-null  object
 2   YEAR             475982 non-null  int64
 3   ADDRESS          475982 non-null  object
 4   COUNTY           475982 non-null  object
 5   POSTAL_CODE      475982 non-null  object
```

```
 6   SALE_PRICE       475982 non-null  float64
 7   IF_MARKET_PRICE  475982 non-null  object
 8   IF_VAT_EXCLUDED  475982 non-null  object
 9   PROPERTY_DESC    475982 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(1), object(7)
memory usage: 39.9+ MB
None
```

[22]:
```
# Drop rows for year 2021
property.drop(property[property['YEAR'] > 2020].index, inplace = True)
```

[23]:
```
# Checking the dataframe now ends at 2020
print(property.tail())
```

```
        SALE_DATE MONTH  YEAR                                    ADDRESS  \
457820 2020-12-31   Dec  2020        53 TRIMBLESTON, GOATSTOWN, DUBLIN 14
457821 2020-12-31   Dec  2020          54 STONEBRIDGE, RATOATH, CO. MEATH
457822 2020-12-31   Dec  2020  APT 9, THE FOUR SEASONS HOTEL, BALLSBRIDGE
457823 2020-12-31   Dec  2020              OAKVALE HOUSE, APT, BARRACK ST
457824 2020-12-31   Dec  2020                PIPPIN, WYNDHAM AVENUE, BRAY

           COUNTY POSTAL_CODE  SALE_PRICE IF_MARKET_PRICE IF_VAT_EXCLUDED  \
457820     Dublin   Dublin 14    400000.0              No              No
457821      Meath     No data    367000.0              No              No
457822     Dublin    Dublin 4   1200000.0              No              No
457823  Roscommon     No data     30000.0              No              No
457824    Wicklow     No data    497500.0              No              No

                              PROPERTY_DESC
457820  Second-Hand Dwelling house /Apartment
457821  Second-Hand Dwelling house /Apartment
457822  Second-Hand Dwelling house /Apartment
457823  Second-Hand Dwelling house /Apartment
457824  Second-Hand Dwelling house /Apartment
```

[24]:
```
# Check for correlation between numeric variables
property.corr()
```

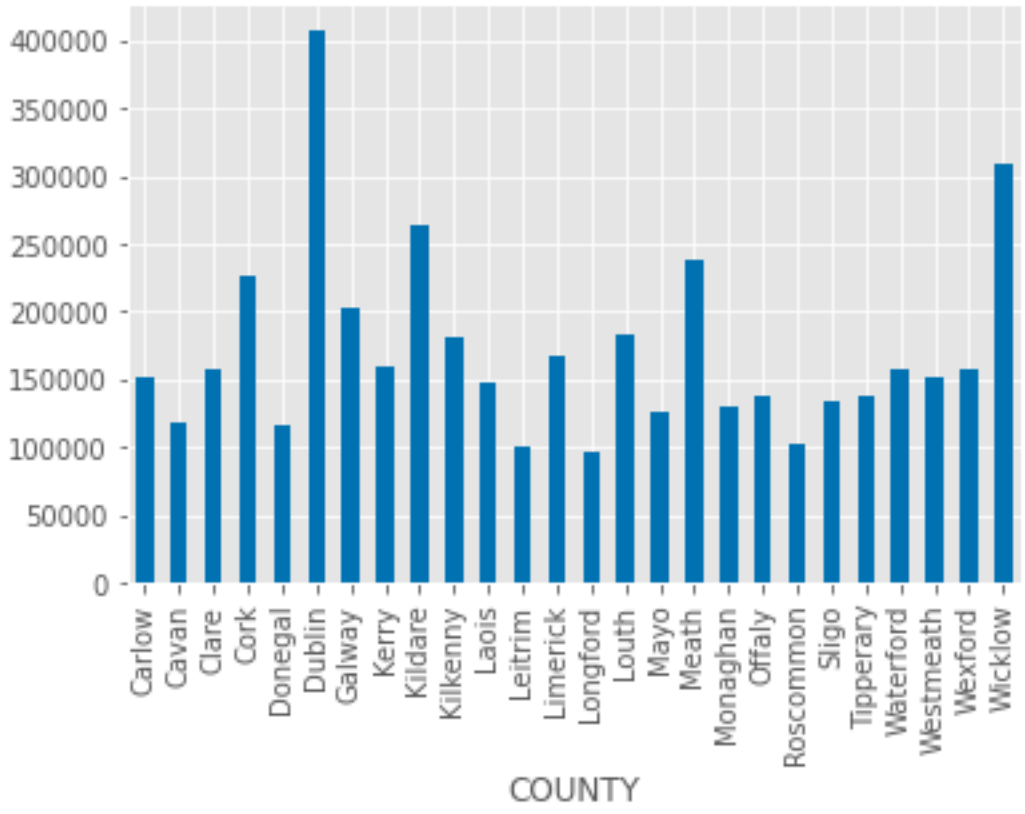
[24]:
```
                YEAR  SALE_PRICE
YEAR        1.000000    0.043687
SALE_PRICE  0.043687    1.000000
```

[25]:
```
# Overview of statistics
property.describe()
```

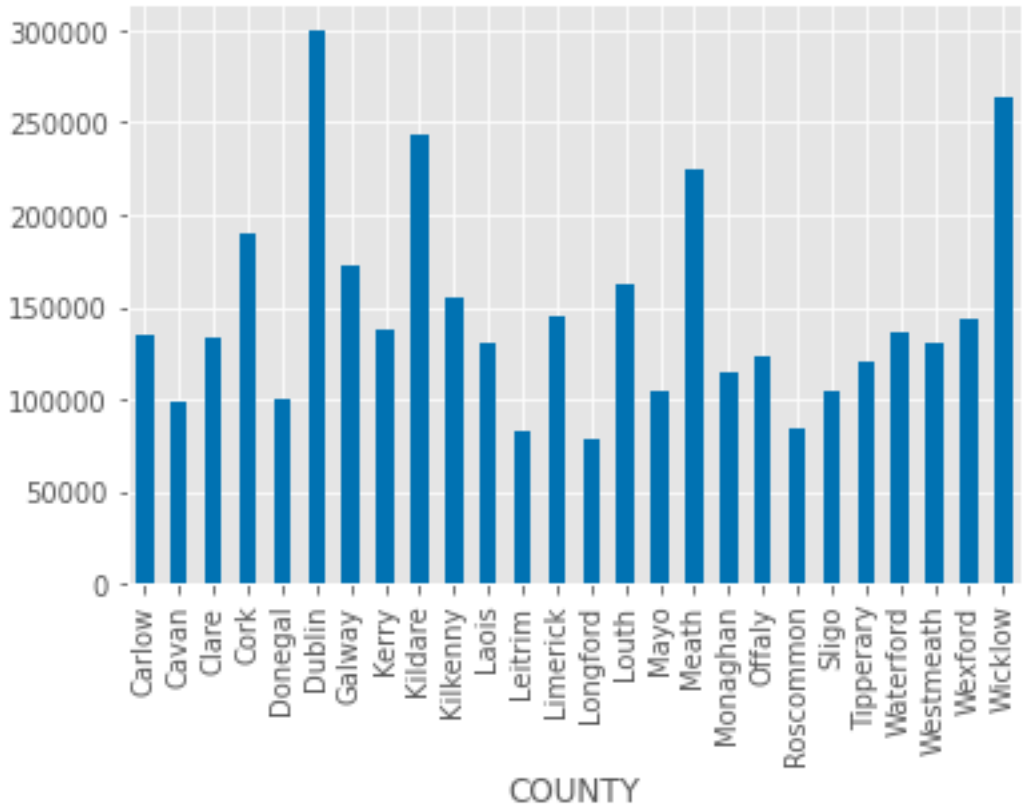
[25]:
```
                YEAR    SALE_PRICE
count  457070.000000  4.570700e+05
```

```
mean      2015.994093  2.566540e+05
std          2.848685  8.242662e+05
min       2010.000000  5.030530e+03
25%       2014.000000  1.150000e+05
50%       2016.000000  1.950000e+05
75%       2018.000000  3.000000e+05
max       2020.000000  1.422566e+08
```

[26]:
```
# Average price by county
plt.style.use("seaborn-colorblind")
avg_price_by_county = property.groupby("COUNTY")["SALE_PRICE"].mean()
avg_price_by_county.plot(kind="bar")
plt.show()
```

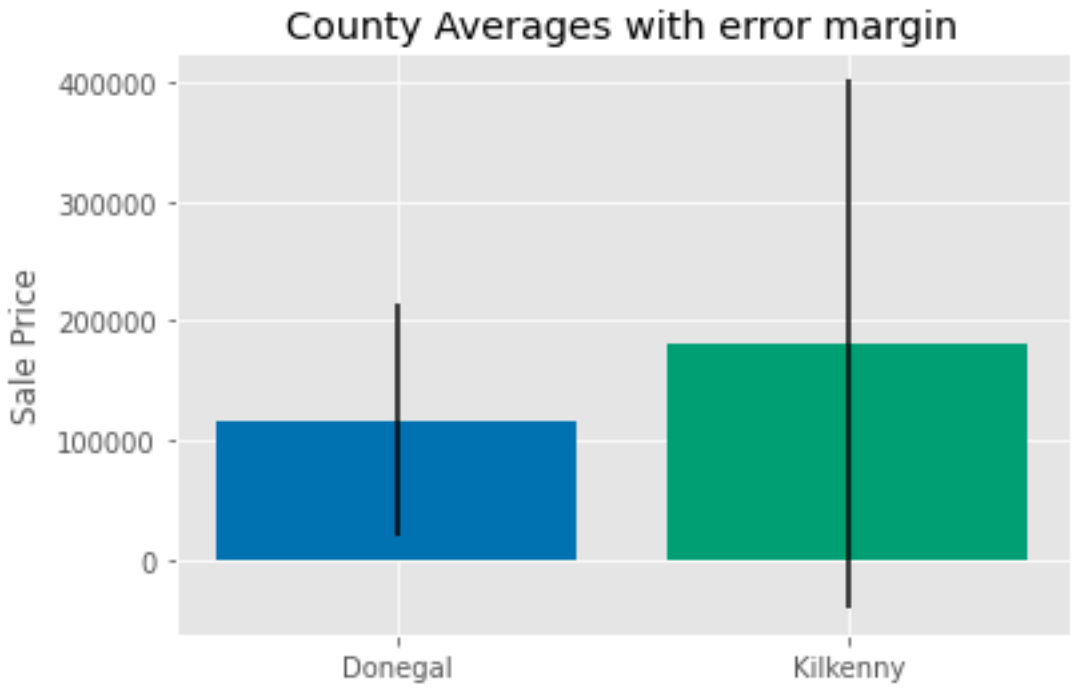


[27]:
```
# Median price by county
avg_price_by_county = property.groupby("COUNTY")["SALE_PRICE"].median()
avg_price_by_county.plot(kind="bar")
plt.show()
```

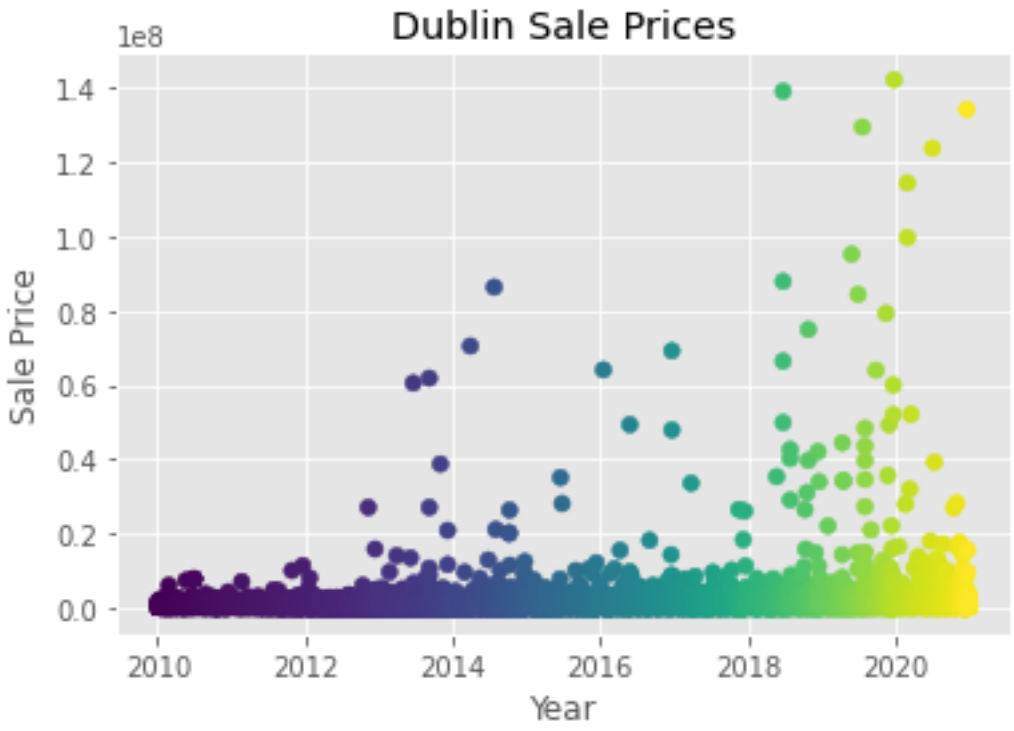


[28]:
```
# Creating variables for a few counties
Dublin = property[property["COUNTY"] == "Dublin"]
Donegal = property[property["COUNTY"] == "Donegal"]
Kilkenny = property[property["COUNTY"] == "Kilkenny"]
```

[29]:
```
# Comparison of 2 counties
plt.style.use("seaborn-colorblind")
fig, ax = plt.subplots()
ax.bar("Donegal",
Donegal["SALE_PRICE"].mean(),
yerr=Donegal["SALE_PRICE"].std())
ax.bar("Kilkenny",
Kilkenny["SALE_PRICE"].mean(),
yerr=Kilkenny["SALE_PRICE"].std())
ax.set_ylabel("Sale Price")
ax.set_title("County Averages with error margin")
plt.show()
```

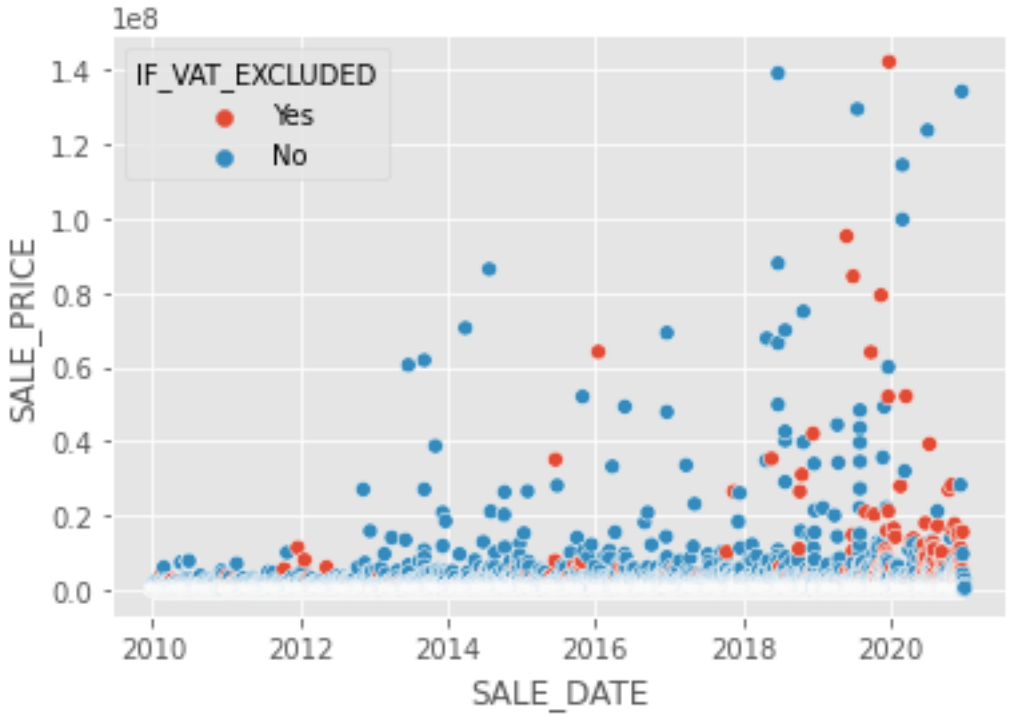


[30]:
```
# Scatter plot for Dublin
fig, ax = plt.subplots()
ax.scatter(Dublin["SALE_DATE"], Dublin["SALE_PRICE"],
c=Dublin.index)
ax.set_xlabel("Year")
ax.set_ylabel("Sale Price")
ax.set_title("Dublin Sale Prices")
plt.show()
```



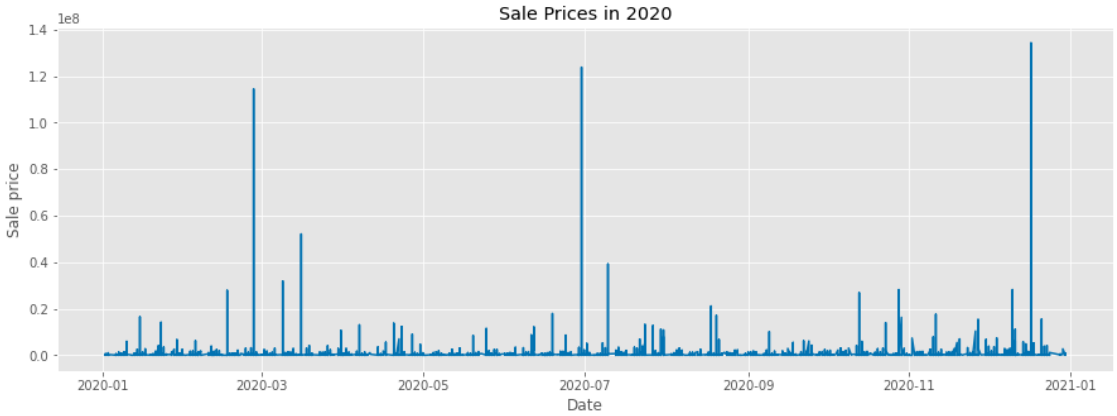
[31]:
```
# Exploring sales prices and if they reached market price
plt.style.use("ggplot")
sns.scatterplot(x="SALE_DATE",
y="SALE_PRICE",
data=property, hue="IF_MARKET_PRICE", hue_order=("Yes","No"))
plt.show()
```

[32]:
```
# Exploring sales prices and if they excluded vat
sns.scatterplot(x="SALE_DATE",
y="SALE_PRICE",
data=property, hue="IF_VAT_EXCLUDED", hue_order=("Yes","No"))
plt.show()
```



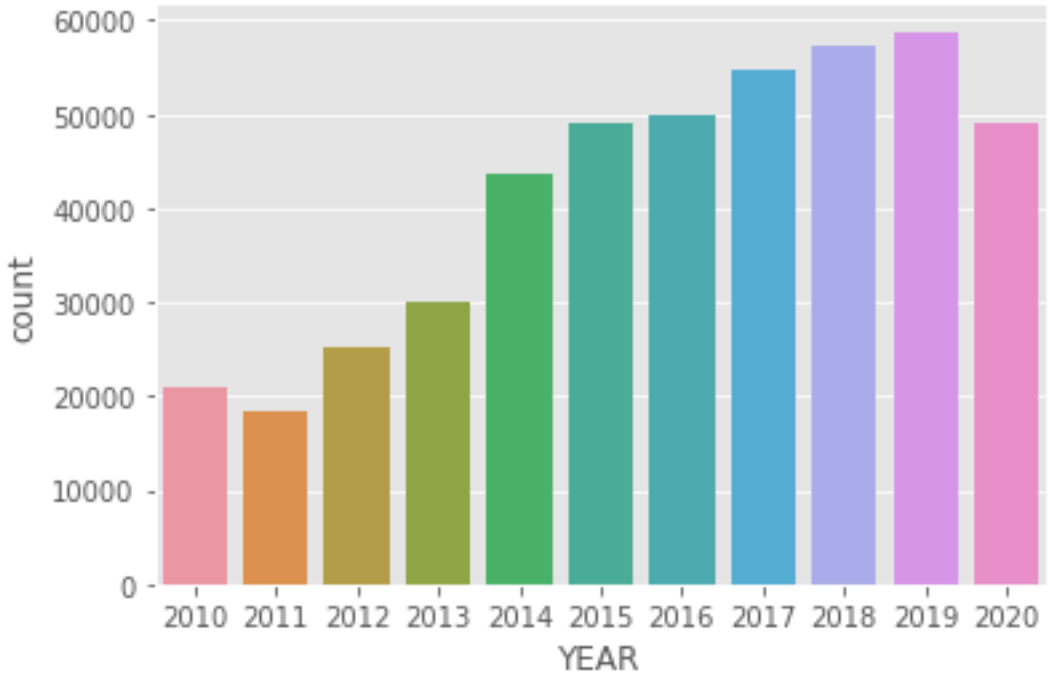
[33]:
```
# Looking at sales for 2020
property2020 = property[
    (property['SALE_DATE'] > '2020-01-01' ) &
    (property['SALE_DATE'] < '2020-12-31')]
```

[34]:
```
# Plotting 2020
plt.style.use("seaborn-colorblind")
fig, ax = plt.subplots()
ax.plot(property2020["SALE_DATE"], property2020["SALE_PRICE"])
ax.set_xlabel("Date")
ax.set_ylabel("Sale price")
ax.set_title("Sale Prices in 2020")
fig.set_size_inches([15,5])
plt.show()
```



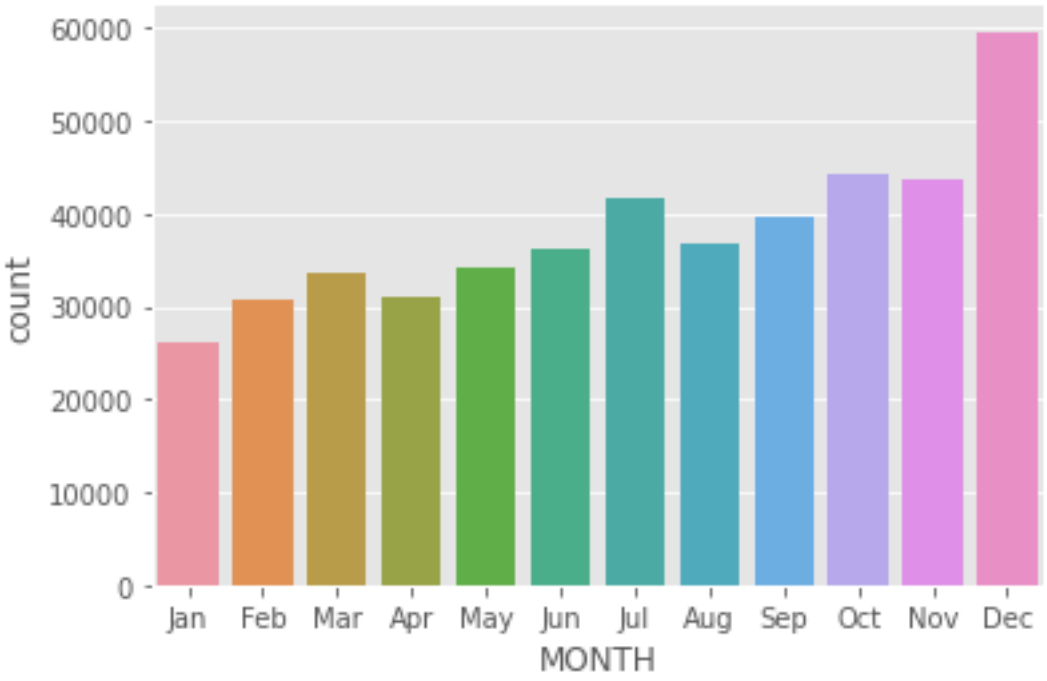
[35]:
```
# Counting totals for each year
sns.countplot(x = "YEAR" , data = property)
```

[35]: <AxesSubplot:xlabel='YEAR', ylabel='count'>



[36]:
```
#Counting totals for each month
sns.countplot(x = "MONTH" , data = property)
```

[36]: <AxesSubplot:xlabel='MONTH', ylabel='count'>


60000
50000
40000
30000
20000
10000
0
count
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
MONTH