Housing in London

Arpan Avvari

OUTLINE OF TOPICS

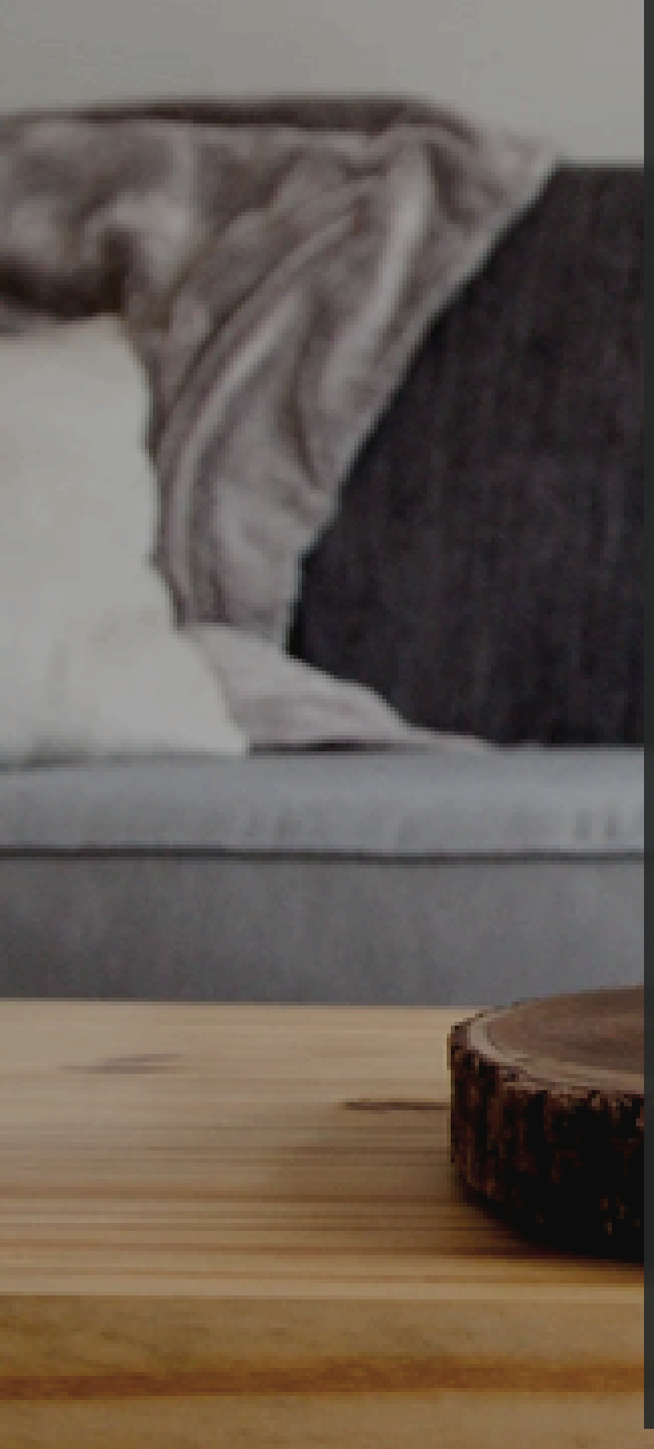
INTRODUCTION

DATA CLEANING

DATA MANIPULATING

QUESTIONS & RESULTS

CONCLUSION



INTRODUCTION

DATASET DETAILS :

Housing in London consists of two datasets which are centered around the housing market of London between the years 1995 and 2019.

Dataset 1 consists of information pertaining to houses sold, like number of houses sold in an area, their average price and so on.

date	area	average_price	code	houses_sold	no_of_crimes	borough_flag

Whereas dataset 2 contains additional information like population of the area, mean salary, life satisfaction and so on.

code	area	date	median_salary	life_satisfaction	mean_salary
recycling_pct	population_size	number_of_jobs	area_size	no_of_houses	borough_flag

Why this?

Real estate in an ever green industry and is volatile. Therefore, buyers and sellers in the housing market need data analytics in order to make wise investment choices and maximize their ROI.

While matching the groups' interest, we searched for real estate data which has well defined columns and least missing values and found this dataset to be a perfect fit.

London

This is the map of London, it has 32 areas.

Our datasets consists of information for every area for every year.

What is the story?

BACKGROUND

Our client, a real estate agent, has their own customers who are planning to invest in London. They had some questions about deciding where to invest. As a real estate data analytics firm we answered their questions.

But before we answered the questions, we pre-processed the data. The next few slides contain information about how the pre-processing is done.

Data Pre-processing

Data Extraction

Data Manipulation

Data Cleaning

Data Extraction

Our raw data not only consist data about areas of London but also about other cities and a few countries around it. We need to extract only London

Further, to consider only the latest and well defined rows we extracted data from 2000 to 2018.

```
[14] Monthdf=Monthdf[Monthdf.code.isin(codes_to_remove)==False]
     Yeardf=Yeardf[Yeardf.code.isin(codes_to_remove)==False]
     #City codes which are not in London are removed
```

```
[15] Monthdf=Monthdf[Monthdf.year.isin(years_to_remove)==False]
     Yeardf=Yeardf[Yeardf.year.isin(years_to_remove)==False]
     #Removing incomplete old data,to deal with only the complete and latest data
```

Data Manipulation

The monthly dataset should be merged with yearly dataset. While variables like number of houses, number of crimes can be summed, taking an average of the average prices is erroneous. Hence the steps of our data manipulation are as follows :

- Grouped monthly data into years for every area

```
[17] newMonthdf = Monthdf.groupby(['year','code']).aggregate({'total_price': 'sum', 'no_of_crimes': 'sum','houses_sold':'sum'}).round(2)
     #Performed group by operation to aggregate month values to year
```

- Calculated new average price

```
[18] newMonthdf['new_average_price']=newMonthdf['total_price']/newMonthdf['houses_sold']
     #Calculated average prices by calculating total price and then dividing by total number of houses to get accurate avergae price
```

- Merged both datasets

```
[20] HousingAll= Yeardf.merge(newMonthdf,left_on=['year','code'],right_on= ['year','code'] )
     #Performed merge to combine both the data sets
```

Data Cleaning

- The merged data set has erroneous data in the form of special characters (#,-), which we changed into NaN.
- Though the extracted and merged dataset consists of information only about London, a few rows are an aggregation of others. The borough flag variable is 1 for areas of London and 0 for all the aggregated values. We cleaned our dataset of the aggregated rows.
- Missing values are later dealt with as per the question requirements.
- First few rows of our dataset after pre-processing :

[48]
Housing

	index	code	area	median_salary	life_satisfaction	mean_salary	recycling_pct	population_size	number_of_jobs	area_size	no_of_houses	borough_flag	year	no_of_crimes	houses_sold	new_average_price
0	0	E09000001	city of london	39104.0	NaN	62819.0	0	7359.0	339000.0	315.0	5009.0	1	2001	0	367	237050.55
1	1	E09000002	barking and dagenham	22323.0	NaN	26050.0	3	165654.0	54000.0	3780.0	68298.0	1	2001	20476	3203	88813.38
2	2	E09000003	barnet	20916.0	NaN	26068.0	8	319481.0	138000.0	8675.0	130515.0	1	2001	28059	6803	185730.54
3	3	E09000004	bexley	20217.0	NaN	23559.0	20	218757.0	75000.0	6429.0	91606.0	1	2001	20224	5270	117150.82
4	4	E09000005	brent	21878.0	NaN	24164.0	7	269620.0	116000.0	4323.0	101427.0	1	2001	29829	4725	157811.22

Which areas have the highest and the lowest growth in terms of price of the house in the last 10 years?

Using a pivot table we reformatted the data to obtain the average price for every year and every area.

Then we calculated change in average price from 2008 to 2018 for every area.

Using min and max functions we found the required answers.

Kensington and chelsea has highest growth wheras Barking and dagenham has the lowest growth from 2008 to 2018

Working :

```
[26] area_year=Housing.pivot_table('new_average_price',index='area',columns='year')
     #Using pivot table to extract required information
```

```
[29] area_yr['change']=area_yr['2018']-area_yr['2008']
     #Calculating change in prices
```

```
[30] highest=area_yr[area_yr['change']==area_yr['change'].max()]
     lowest=area_yr[area_yr['change']==area_yr['change'].min()]
```

Output :

```
[31] print("The area which has the highest growth from 2008 to 2018 is ",highest['area'].iloc[0])
     print("The area which has the lowest growth from 2008 to 2018 is ",lowest['area'].iloc[0])
```

```
The area which has the highest growth from 2008 to 2018 is  kensington and chelsea
The area which has the lowest growth from 2008 to 2018 is  barking and dagenham
```

Which areas have the highest number of houses sold in the latest year?

To obtain the total number of houses sold in every area in 2018, we used Pivot table.

Using sort function and head function we obtained list of 5 areas which have the highest number of houses sold.

Areas Wandsworth, Bromley, Croydon, Barnet and Southwark have the highest number of houses sold in 2018.

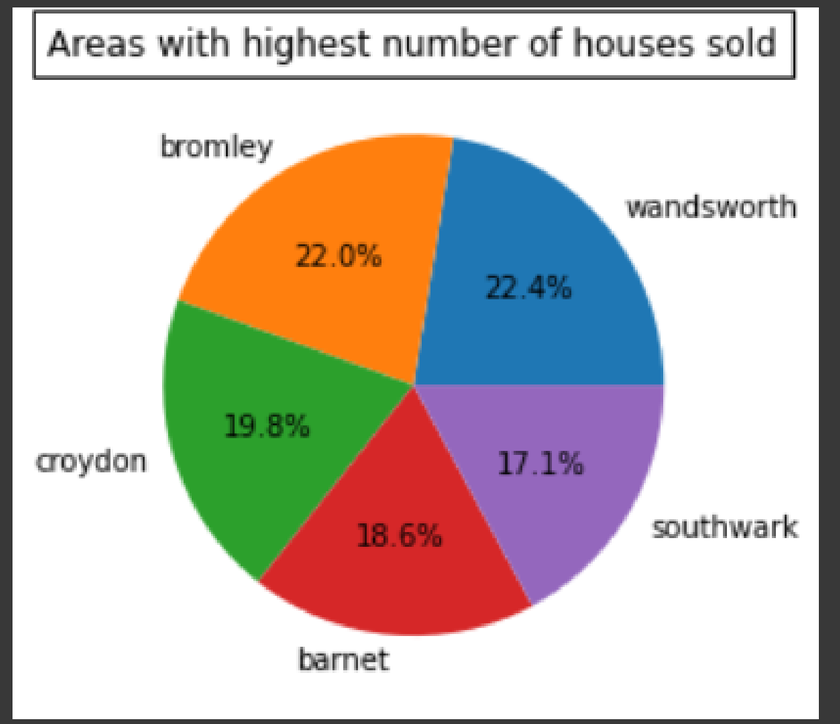
Working :

```
[42] houses=Housing[Housing['year']==2018].pivot_table('houses_sold',index='area',columns='year')
```

```
top5=houses_2018.sort_values(by='2018',ascending=False,ignore_index=True).head(5)
```

```
plt.title("Areas with highest number of houses sold",bbox={'facecolor':'1', 'pad':5})
plt.pie(top5["2018"], labels = top5["area"],autopct='%1.1f%%')
plt.show()
```

Output :



Which areas have the highest crime rates in the latest year?

To calculate crime rate, we divided number of crimes by population, multiplied it by 100. Now crime rate is a percentage of the population of the area.

Using sort function and head function we found the areas where crime rate is highest.

Areas Westminster, Kensington and Chelsea, Camden, Islington and, Hammersmith and Fulham have the highest crime rates.

Working :

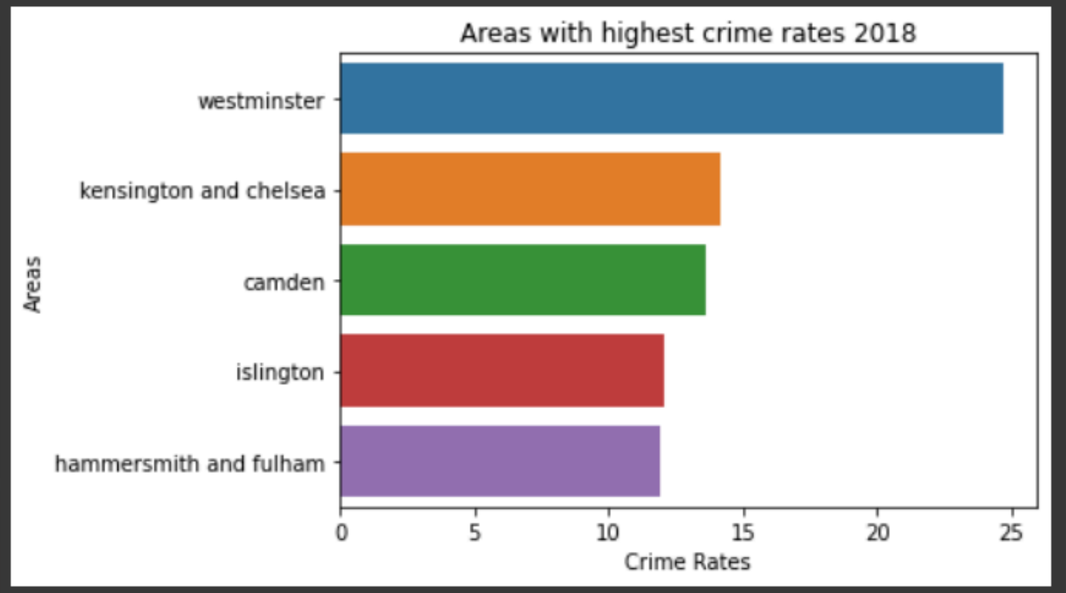
```
[33] df['crime_rate']=(df['no_of_crimes']/df['population_size'])*100
     #Calculated crime rate for every area for 2018
```

```
[34] highest_crimerates=df.sort_values(by='crime_rate',ascending=False,ignore_index=True).head(5)
```

```
[36] ax=plt.subplot()
     ax=sns.barplot(x='crime_rate',y='area',data=highest_crimerates)

     ax.set_title('Areas with highest crime rates 2018')
     ax.set_xlabel('Crime Rates')
     ax.set_ylabel('Areas')
     plt.show()
```

Output :



Based on the results till now, I pick Barnet and Croydon. Compare Life Satisfaction and Average house prices for both.

A pictorial representation would give the best understanding. To plot the same :

- We extracted Barnet and Croydon information into seperate datasets
- Formed axes 1 and 2 with year and average prices for Barnet and Croydon
- Formed axes 3 and 4 with year and life satisfaction for Barnet and Croydon
- Outputs in next slide

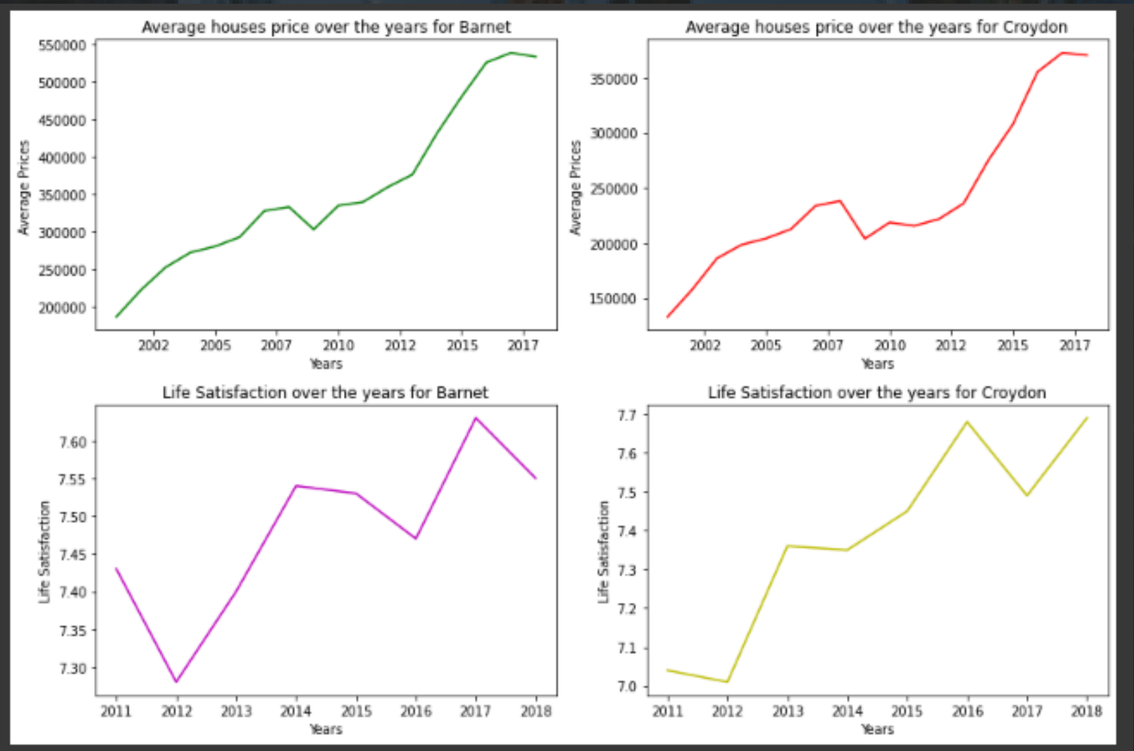
Working :

```
[38] dataset1=Housing[Housing['area']=='barnet']
     dataset2=Housing[Housing['area']=='croydon']

     fig=plt.figure(figsize=(12,8))
     axes1=fig.add_subplot(2,2,1)
     axes2=fig.add_subplot(2,2,2)
     axes3=fig.add_subplot(2,2,3)
     axes4=fig.add_subplot(2,2,4)

     axes1.plot(dataset1['year'],dataset1['new_average_price'],'g')
     axes1.xaxis.set_major_formatter(FormatStrFormatter('%d'))
     axes2.plot(dataset2['year'],dataset2['new_average_price'],'r')
     axes2.xaxis.set_major_formatter(FormatStrFormatter('%d'))
     axes3.plot(dataset1['year'],dataset1['life_satisfaction'],'m')
     axes4.plot(dataset2['year'],dataset2['life_satisfaction'],'y')
```

Output :



Do a number of crimes and mean salary affect the average price of the house?

To find if there is a relationship between the average price of the house and the number of crimes and the mean salary, we ran a *Multi Regression*.
Here the average price is the response variable and the number of crimes and mean salary are predictor variables.

According to the regression model results, for every unit increase in mean salary the average price of the house increase by 6.77 units. The p-value for crime rates is not low enough so its value is not significant.

Working :

```
[48] newdf=Housing[['new_average_price','no_of_crimes','mean_salary']]
```

```
[50] newdf=newdf.dropna(how="any")
     #dropped NaN values in these columns
```

```
[51] housing_model = smf.ols(formula = 'new_average_price ~ no_of_crimes+mean_salary',
                             data=newdf)
     #Performing multi regression
```

Output :

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      new_average_price   R-squared:                       0.179
Model:                            OLS   Adj. R-squared:                  0.177
Method:                 Least Squares   F-statistic:                     64.21
Date:                Wed, 11 May 2022   Prob (F-statistic):           6.07e-26
Time:                        03:38:23   Log-Likelihood:                -7942.9
No. Observations:                 590   AIC:                         1.589e+04
Df Residuals:                     587   BIC:                         1.590e+04
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   7.964e+04   2.87e+04      2.771      0.006    2.32e+04    1.36e+05
no_of_crimes     0.5452      0.642      0.849      0.396      -0.716       1.806
mean_salary      6.7762      0.599     11.314      0.000       5.600       7.952
==============================================================================
Omnibus:                      328.134   Durbin-Watson:                   1.306
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2706.061
Skew:                           2.358   Prob(JB):                         0.00
Kurtosis:                      12.372   Cond. No.                     1.90e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.9e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

CONCLUSION

- Real estate data analytics gave some interesting insights, helped us understand different parameters and working of things, and allowed us to answer possible questions from a realtor.
- Through our analysis we found the areas which have the highest crime rates, the trend of life satisfaction and average pricing of areas over the years, and how the number of crimes and mean salaries affected average prices.
- We found this project very beneficial as it allowed us to explore the concepts of merging, group_by, graph plotting, and regression and upskill our data manipulation and modelling skills.
- The challenges faced in this project were very valuable too as we dealt with difficult data cleaning and data merging.


Thank You