King Saud University
College of Computer and Information Sciences
Information Technology Department

جامعة الملك سعود
King Saud University

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

# IT362
# Principles of Data Science

## Real estate market Report

| member name | ID |
|---|---|
| Atheer alzaid | 441201404 |
| Shadin Alsaif | 442200395 |
| Dana alsaeedi | 441201237 |
| Nourah albarrak | 439201100 |

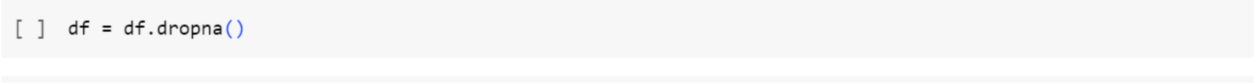
## Data collection

We've collected data from Aqar website by using web scraping which is a very helpful tool to scrape data from a website.
One of the challenges that we’ve faced was some of the attributes which are (furnished, and age)  weren’t  always added to the details of any real estate therefore it will cause a lot of missing values. And since those attributes weren’t important in our analysis we’ve decided not to collect them. Another challenge was the language of the website which is Arabic so we had to translate district, and cities to English.

## Data preparation and cleaning

First we deleted the rows with null values using the method Dropna() **Figure 1**. Second, we checked on attribute types and gave every attribute its correct type. Third, We applied normalization on two of the numerical attributes we have: Price and size **Figure 2**. Then we applied label encoding on the categorical attributes: City and district **Figure 3** and **Figure 4**. Lastly we applied binary encoding for the attribute: Type which has two values: villa and apartment **Figure5**.

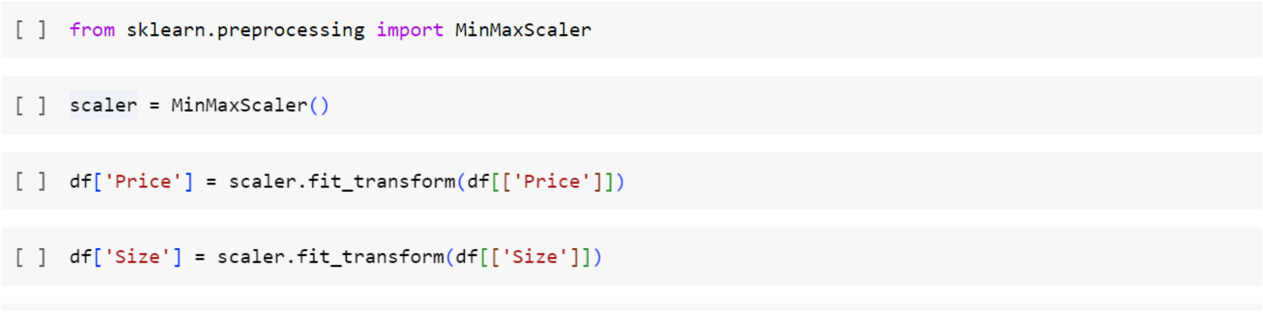
-null values



```
[ ]  df = df.dropna()
```

Figure 1 null values

-Normalization



```
[ ]  from sklearn.preprocessing import MinMaxScaler
```

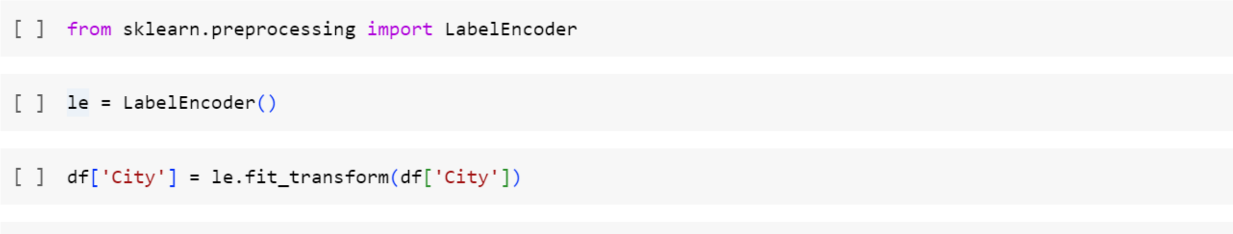
```
[ ]  scaler = MinMaxScaler()
```

```
[ ]  df['Price'] = scaler.fit_transform(df[['Price']])
```

```
[ ]  df['Size'] = scaler.fit_transform(df[['Size']])
```

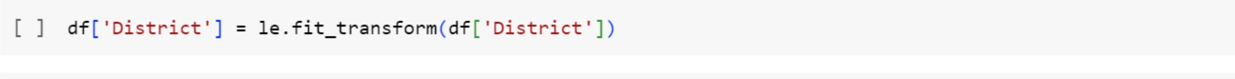
Figure 2 normalization

-Label encoding



```
[ ] from sklearn.preprocessing import LabelEncoder

[ ] le = LabelEncoder()

[ ] df['City'] = le.fit_transform(df['City'])
```

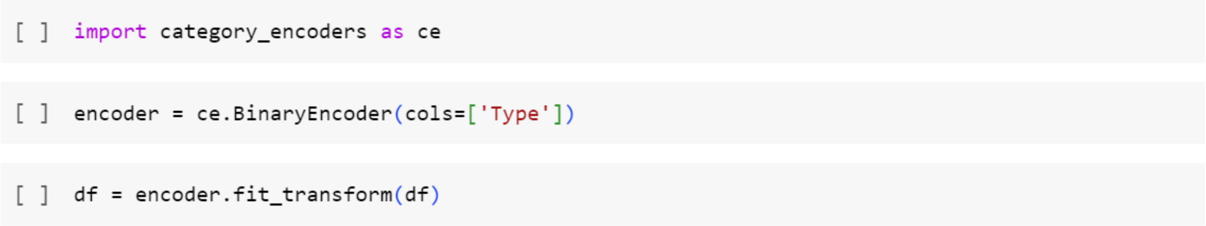
Figure 3 label encoding



```
[ ] df['District'] = le.fit_transform(df['District'])
```

Figure 4 label encoding

-Binary encoding



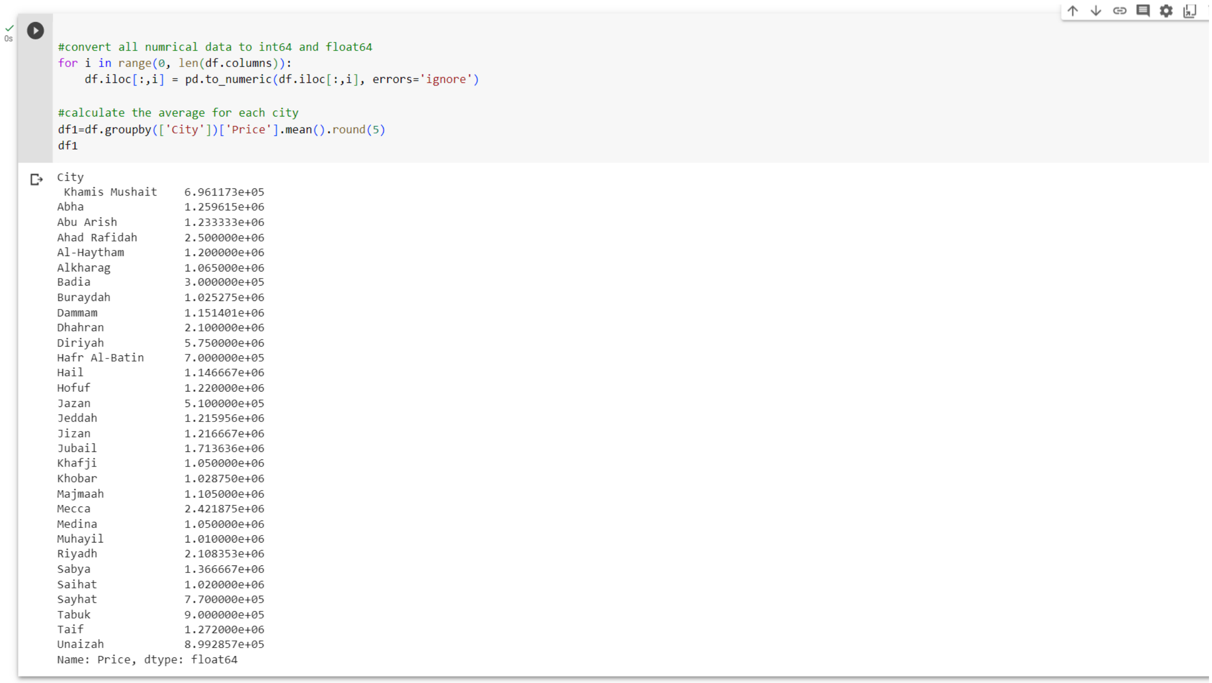
```
[ ] import category_encoders as ce

[ ] encoder = ce.BinaryEncoder(cols=['Type'])

[ ] df = encoder.fit_transform(df)
```

Figure 5 binary encoding

## Results (questions and answers)

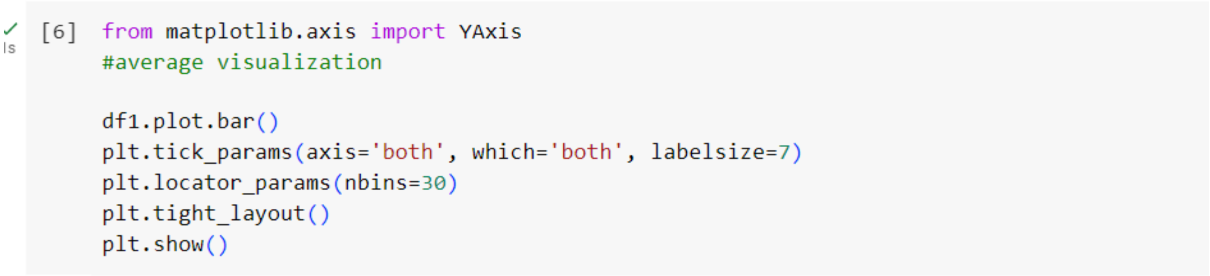
### 1. What is the average price of properties in each city?

We found the average price of properties in Saudi Arabia is different depending on the city and it is in a range between 9.0 million SAR(for Tabuk) to  1.1 million SAR (for muhayil).



```
#convert all numrical data to int64 and float64
for i in range(0, len(df.columns)):
    df.iloc[:,i] = pd.to_numeric(df.iloc[:,i], errors='ignore')

#calculate the average for each city
df1=df.groupby(['City'])['Price'].mean().round(5)
df1
```

```
City
 Khamis Mushait    6.961173e+05
Abha              1.259615e+06
Abu Arish         1.233333e+06
Ahad Rafidah      2.500000e+06
Al-Haytham        1.200000e+06
Alkharag          1.065000e+06
Badia             3.000000e+05
Buraydah          1.025275e+06
Dammam            1.151401e+06
Dhahran           2.100000e+06
Diriyah           5.750000e+06
Hafr Al-Batin     7.000000e+05
Hail              1.146667e+06
Hofuf             1.220000e+06
Jazan             5.100000e+05
Jeddah            1.215956e+06
Jizan             1.216667e+06
Jubail            1.713636e+06
Khafji            1.050000e+06
Khobar            1.028750e+06
Majmaah           1.105000e+06
Mecca             2.421875e+06
Medina            1.050000e+06
Muhayil           1.010000e+06
Riyadh            2.108353e+06
Sabya             1.366667e+06
Saihat            1.020000e+06
Sayhat            7.700000e+05
Tabuk             9.000000e+05
Taif              1.272000e+06
Unaizah           8.992857e+05
Name: Price, dtype: float64
```



```
[6] from matplotlib.axis import YAxis
    #average visualization

    df1.plot.bar()
    plt.tick_params(axis='both', which='both', labelsize=7)
    plt.locator_params(nbins=30)
    plt.tight_layout()
    plt.show()
```

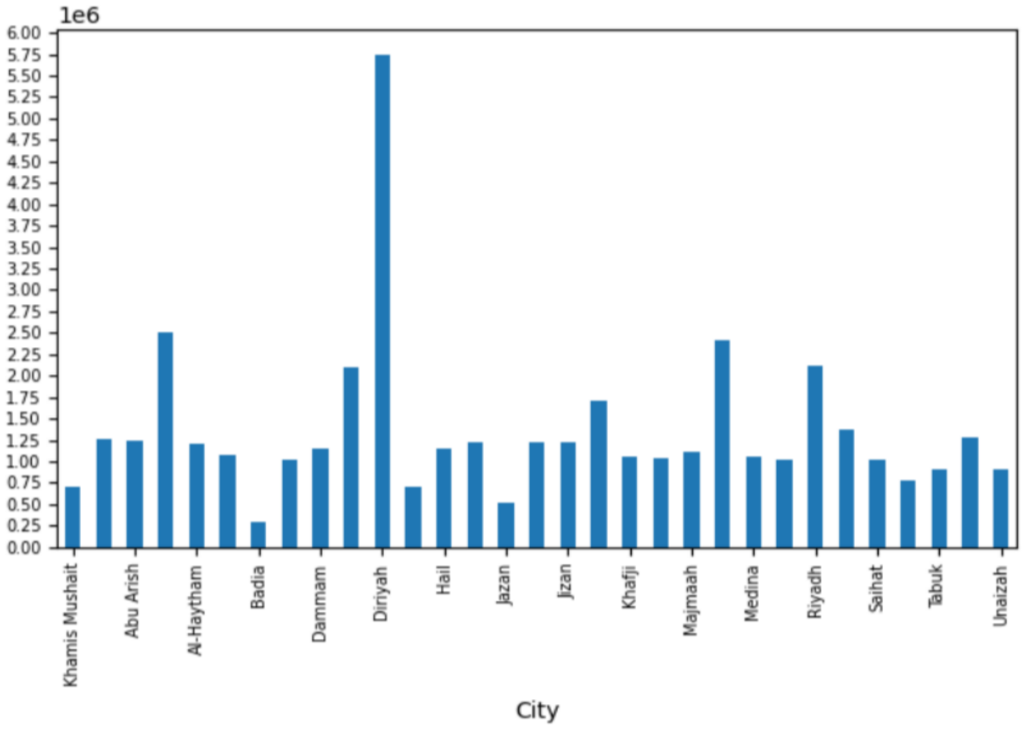
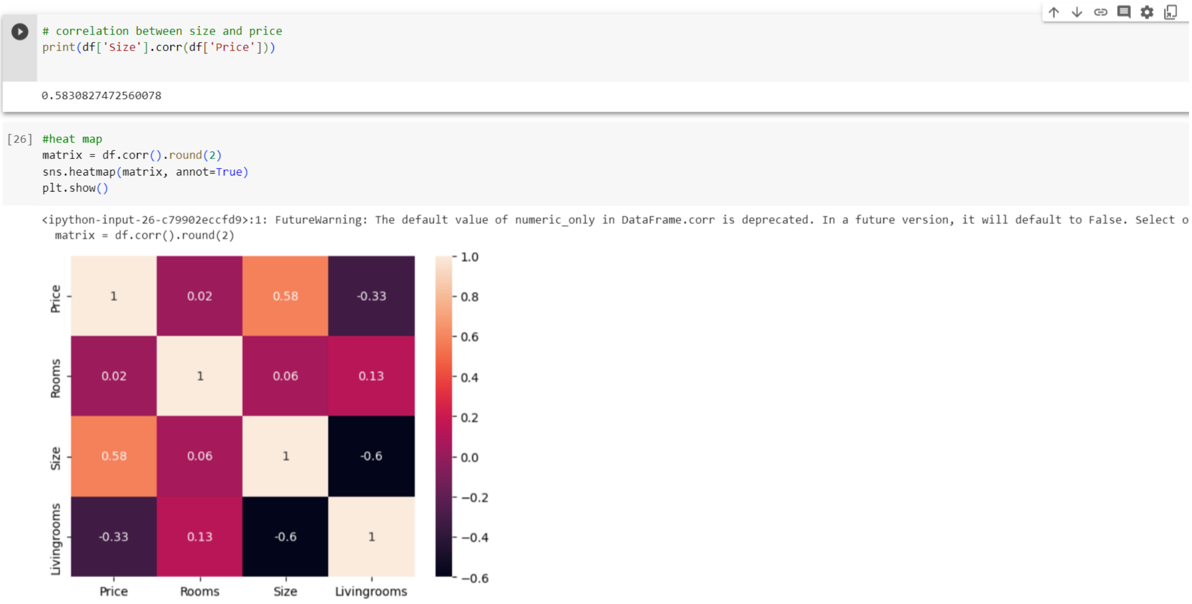


Figure 6 bar chart for average prices

## 2. Is there a correlation between the size of a property and its price?

Yes there is 0.58 correlation between the size and the price of property in Saudi Arabia real estates and after plotting the correlation heat map for all the attributes we found our prediction is true.



```
# correlation between size and price
print(df['Size'].corr(df['Price']))
```

```
0.5830827472560078
```

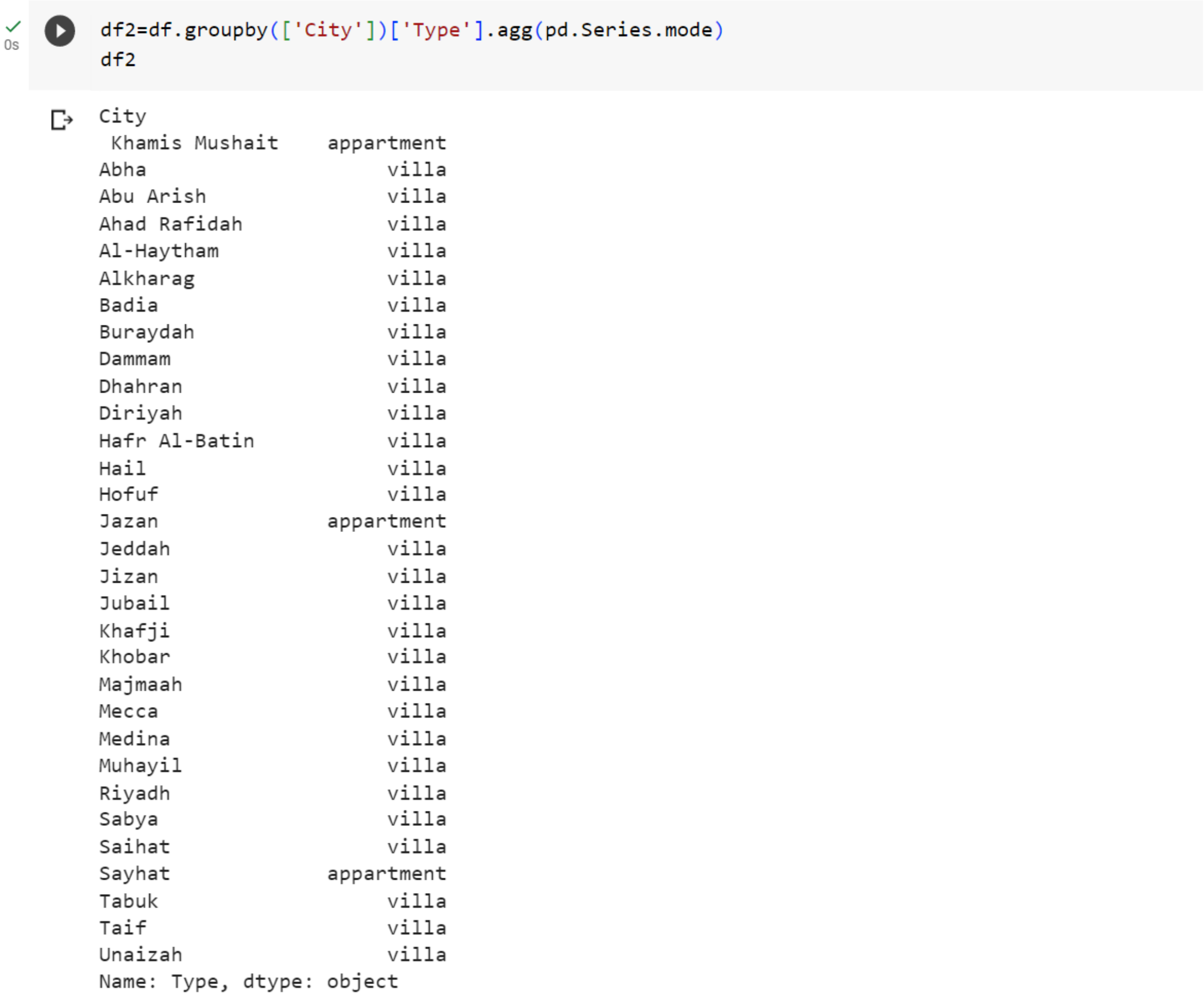
```
[26] #heat map
     matrix = df.corr().round(2)
     sns.heatmap(matrix, annot=True)
     plt.show()
```

```
<ipython-input-26-c79902eccfd9>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select o
  matrix = df.corr().round(2)
```

Figure 7 heat map for all attributes correlation

## 3. What is the most common property type in each city?

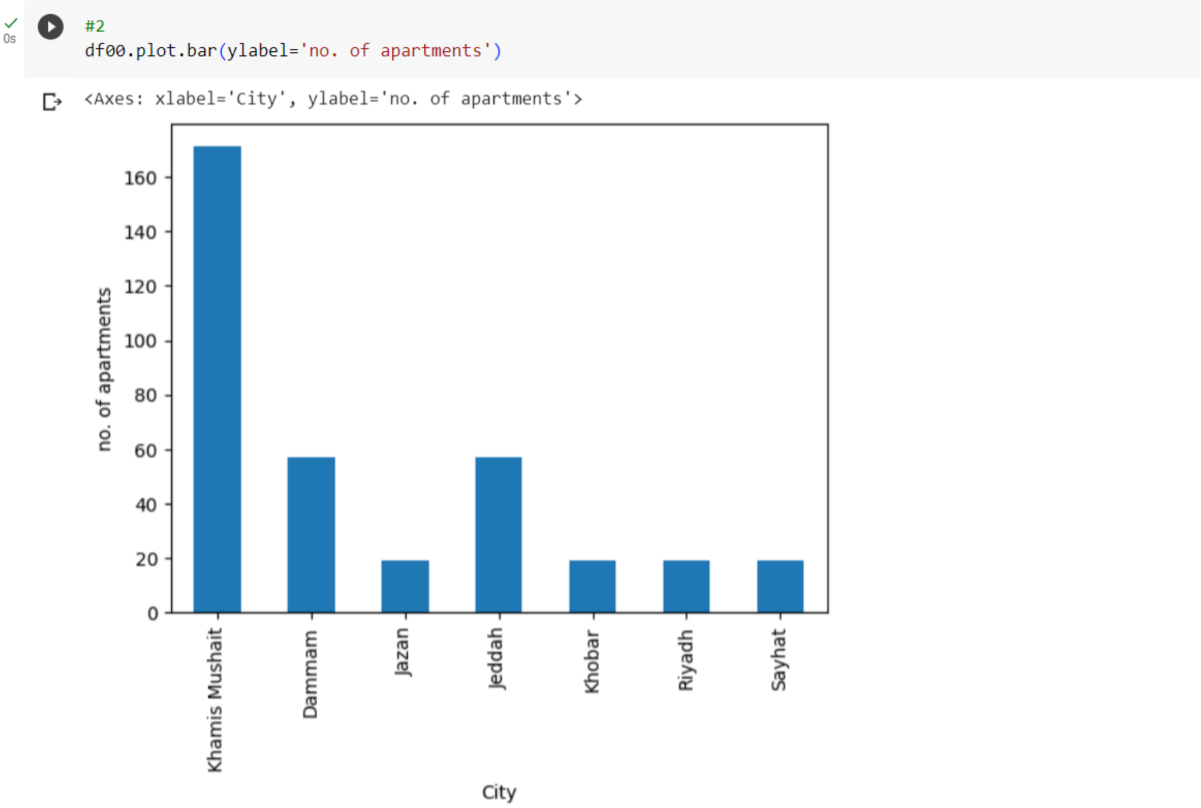
After analyzing the most common property for all the cities we found villa properties are the most common except three cities that had apartments which are Khamis Mushit, Jazan and Sayhat.



0s

```
df2=df.groupby(['City'])['Type'].agg(pd.Series.mode)
df2
```

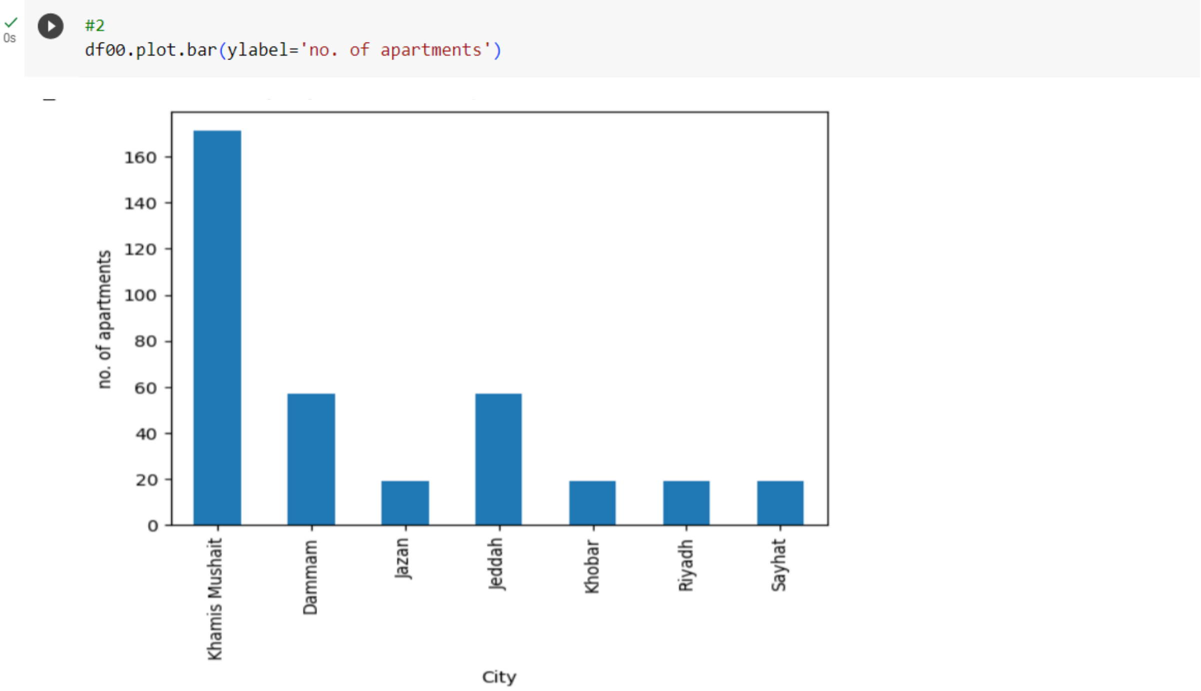
```
City
 Khamis Mushait    appartment
Abha                    villa
Abu Arish               villa
Ahad Rafidah            villa
Al-Haytham              villa
Alkharag                villa
Badia                   villa
Buraydah                villa
Dammam                  villa
Dhahran                 villa
Diriyah                 villa
Hafr Al-Batin           villa
Hail                    villa
Hofuf                   villa
Jazan              appartment
Jeddah                  villa
Jizan                   villa
Jubail                  villa
Khafji                  villa
Khobar                  villa
Majmaah                 villa
Mecca                   villa
Medina                  villa
Muhayil                 villa
Riyadh                  villa
Sabya                   villa
Saihat                  villa
Sayhat             appartment
Tabuk                   villa
Taif                    villa
Unaizah                 villa
Name: Type, dtype: object
```



```
#2
df00.plot.bar(ylabel='no. of apartments')
```

```
<Axes: xlabel='City', ylabel='no. of apartments'>
```

**Figure 8(a) number of villa for each city**

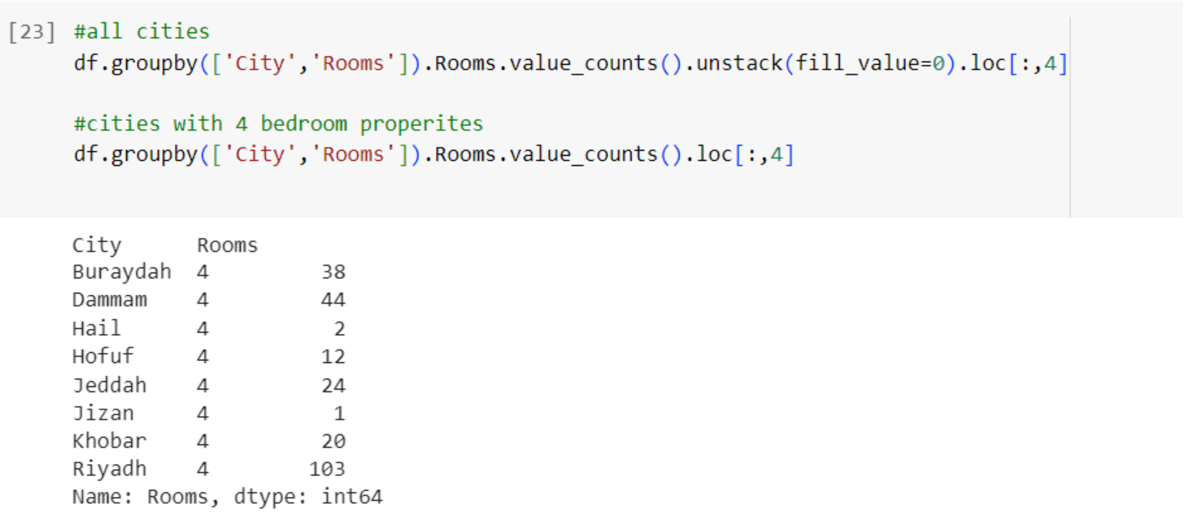


```
#2
df00.plot.bar(ylabel='no. of apartments')
```

**Figure 8(b) number of apartments for each city**

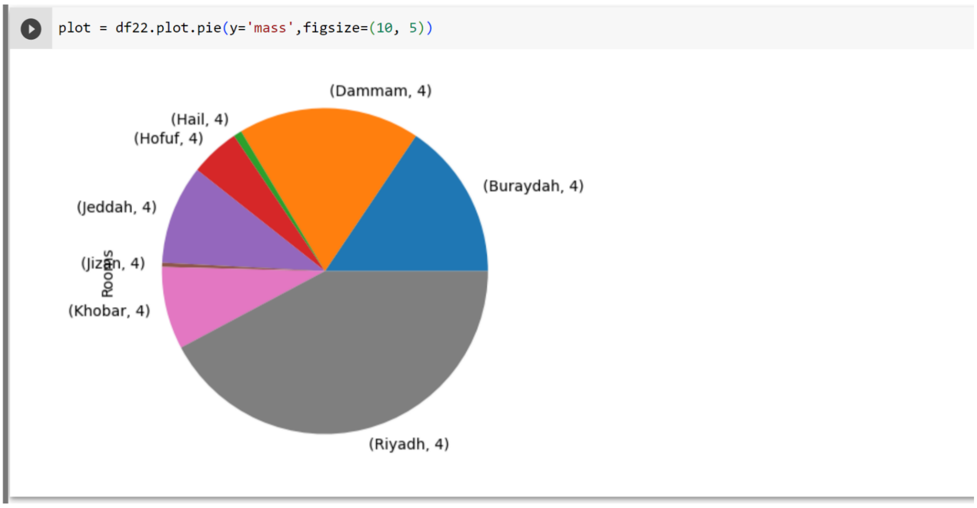
## 4. How many properties have 4 bedrooms per city?

We found 8 cities that have 4 bedroom properties which are Riyadh with 103 properties , Dammam have 44, Buraydah have 38 ,Jeddah have 24 ,Khobar have 20 , Hofouf have 12, Hail have 2 and Jizan have 1.



```
[23] #all cities
     df.groupby(['City','Rooms']).Rooms.value_counts().unstack(fill_value=0).loc[:,4]

     #cities with 4 bedroom properites
     df.groupby(['City','Rooms']).Rooms.value_counts().loc[:,4]
```

```
City      Rooms
Buraydah  4         38
Dammam    4         44
Hail      4          2
Hofuf     4         12
Jeddah    4         24
Jizan     4          1
Khobar    4         20
Riyadh    4        103
Name: Rooms, dtype: int64
```

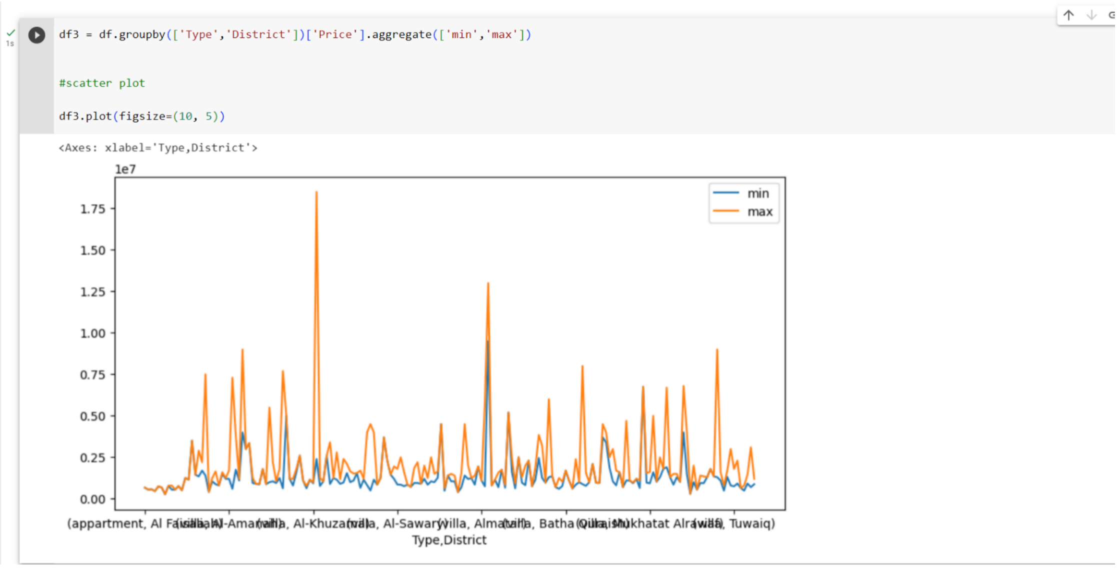


```
plot = df22.plot.pie(y='mass',figsize=(10, 5))
```

**Figure 9 pie chart for 4 bedroom properties**

## 5. What is the most expensive and least expensive district for each property type?

After finding The most and least expensive district for each property type we can see that most of the districts have a close price range in the same district.



```
df3 = df.groupby(['Type','District'])['Price'].aggregate(['min','max'])


#scatter plot

df3.plot(figsize=(10, 5))
```

```
<Axes: xlabel='Type,District'>
```

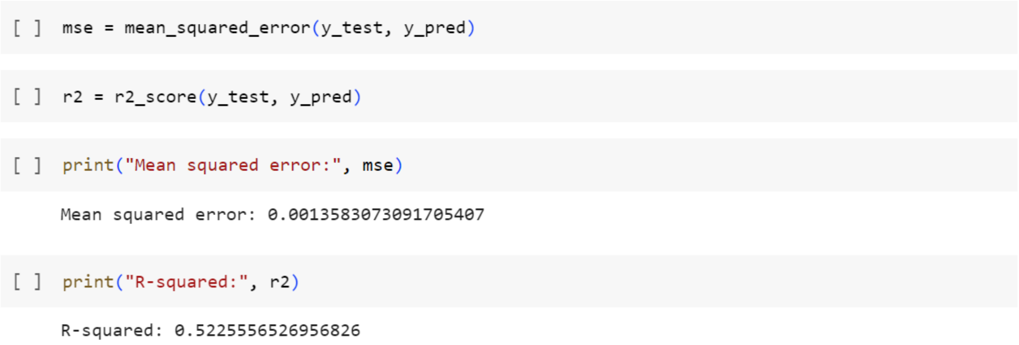
**Figure 10  min max price scatter plot**

## Classifier Model

We made a classifier model that predicts the price of a property based on the following attributes: Number of rooms, Size , Number of livingrooms, City , District and Type of property. For our classifier type, we decided to use Linear regression since the target variable is (price) which is a numeric value.

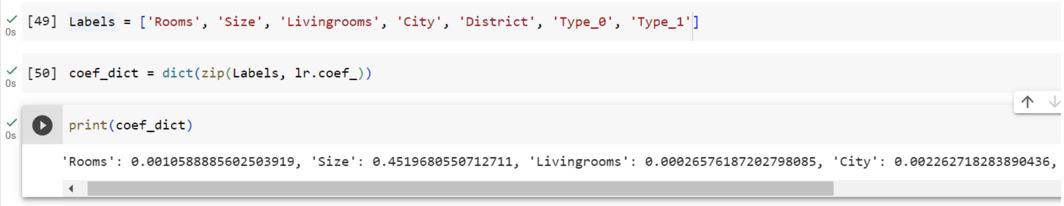
To evaluate the accuracy of our model we used two measures: Mean squared error and R-squared. Our mean squared error was: 0.0013 and R-squared was: 0.522 **Figure 11** which indicates that our model accuracy is really high.

Finally, to take a better look on the relationships between the dependent variables and the target variable we used the coefficients  of our model and found that the strongest relation the target has with a variable was with the Variable: Size with coefficient 0.45, while the weakest relation was with the variable: District with coefficient 0.0001 **Figure 12** and **Figure 13**.



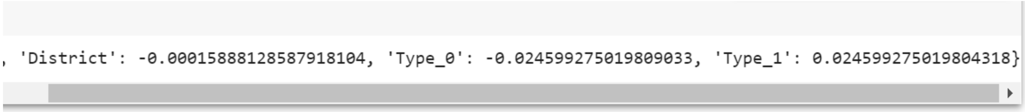
```
[ ]  mse = mean_squared_error(y_test, y_pred)

[ ]  r2 = r2_score(y_test, y_pred)

[ ]  print("Mean squared error:", mse)

     Mean squared error: 0.0013583073091705407

[ ]  print("R-squared:", r2)

     R-squared: 0.5225556526956826
```

Figure 11 mean square error & R squared



```
[49] Labels = ['Rooms', 'Size', 'Livingrooms', 'City', 'District', 'Type_0', 'Type_1']

[50] coef_dict = dict(zip(Labels, lr.coef_))

     print(coef_dict)

     'Rooms': 0.0010588885602503919, 'Size': 0.4519680550712711, 'Livingrooms': 0.00026576187202798085, 'City': 0.002262718283890436,
```

Figure 12 model coefficient



```
, 'District': -0.00015888128587918104, 'Type_0': -0.024599275019809033, 'Type_1': 0.024599275019804318}
```

Figure 13 model coefficient