Toronto Dwellings Analysis

In this assignment, you will perform fundamental analysis for the Toronto dwellings market to allow potential real estate investors to choose rental investment properties.

In [1]:

```
# imports
import panel as pn
import plotly.express as px
import pandas as pd
import matplotlib.pyplot as plt
import os
from pathlib import Path
from dotenv import load_dotenv
from panel.interact import interact
from panel import widgets
```

In [2]:

```
# Set up Panel Plotly extension
pn.extension('plotly')
```

In [3]:

```
# Import hvplot.pandas after pn.extension
# This avoids plotly initialization failure
import hvplot.pandas
```

In [4]:

```
# Read the Mapbox API key
load_dotenv()
map_box_api = os.getenv("mapbox")
px.set_mapbox_access_token(map_box_api)
```

Load Data

In [5]:

```
# Read the census data into a Pandas DataFrame
file_path = Path("Data/toronto_neighbourhoods_census_data.csv")
to_data = pd.read_csv(file_path, index_col="year")
to_data.drop(columns=["average_house_value"],inplace=True)
to_data.head()
```

Out[5]:

year	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling
2001	Agincourt North	3715	1480	0
2001	Agincourt South-Malvern West	3250	1835	0
2001	Alderwood	3175	315	0
2001	Annex	1060	6090	5
2001	Banbury-Don Mills	3615	4465	0

Dwelling Types Per Year

In this section, you will calculate the number of dwelling types per year. Visualize the results using bar charts and the Pandas plot function.

Hint: Use the Pandas `groupby` function.

In [6]:

```
# Calculate the sum number of dwelling types units per year (hint: use groupb
to_data_per_year = to_data.groupby('year').sum()
to_data_per_year.head()
```

Out[6]:

year	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detached_
2001	300930	355015	75	
2006	266860	379400	165	
2011	274940	429220	100	
2016	269680	493270	95	

Optional challenge: Plot each bar chart in a different color.

In [7]:

```
dwelling_types  = [x for x in to_data_per_year.columns]
plot = to_data_per_year[dwelling_types[0]].hvplot(kind="bar",yformatter='%f')
for i in range(1,len(dwelling_types)):
    plot = plot * to_data_per_year[dwelling_types[i]].hvplot(kind="bar",yform
```

In [8]:

```
plot
```

Out[8]:

In [9]:

```
# Save the dataframe as a csv file
to_data_per_year.to_csv("Data/Export_to_data_per_year.csv")
```

In [10]:

```
# Helper create_bar_chart function
def create_bar_chart(data, title, xlabel, ylabel, color):
    """
    Create a barplot based in the data argument.
    """
    plot = data.hvplot(kind="bar", rot=90,yformatter='%f')
    plot.opts(title=title)
    plot.opts(xlabel=xlabel)
    plot.opts(ylabel=ylabel)
    plot.opts(color=color)
    return plot
```

In [ ]:

```

```

In [11]:

```
# Create a bar chart per year to show the number of dwelling types


# Bar chart for 2001
plots_to_show2001 =  create_bar_chart(to_data_per_year.iloc[0],"Dwelling Type

# Bar chart for 2006
plots_to_show2006 =  create_bar_chart(to_data_per_year.iloc[1],"Dwelling Type


# Bar chart for 2011
plots_to_show2011 =  create_bar_chart(to_data_per_year.iloc[2],"Dwelling Type

# Bar chart for 2016
plots_to_show2016 =  create_bar_chart(to_data_per_year.iloc[3],"Dwelling Type
```

In [12]:

```
pn_yearly_column = pn.Column("## Dwelling Types in Toronto", plots_to_show200
pn_yearly_column
```

Out[12]:

Average Monthly Shelter Costs in Toronto Per Year

In this section, you will calculate the average monthly shelter costs for owned and rented dwellings and the average house value for each year. Plot the results as a line chart.

Optional challenge: Plot each line chart in a different color.

In [13]:

```
# Calculate the average monthly shelter costs for owned and rented dwellings
df_monthly_shelter_costs = pd.DataFrame(round(to_data[["shelter_costs_owned",
df_monthly_shelter_costs
```

Out[13]:

	shelter_costs_owned	shelter_costs_rented
year		
2001	846.88	1085.94
2006	1316.80	925.41
2011	1448.21	1019.79
2016	1761.31	1256.32

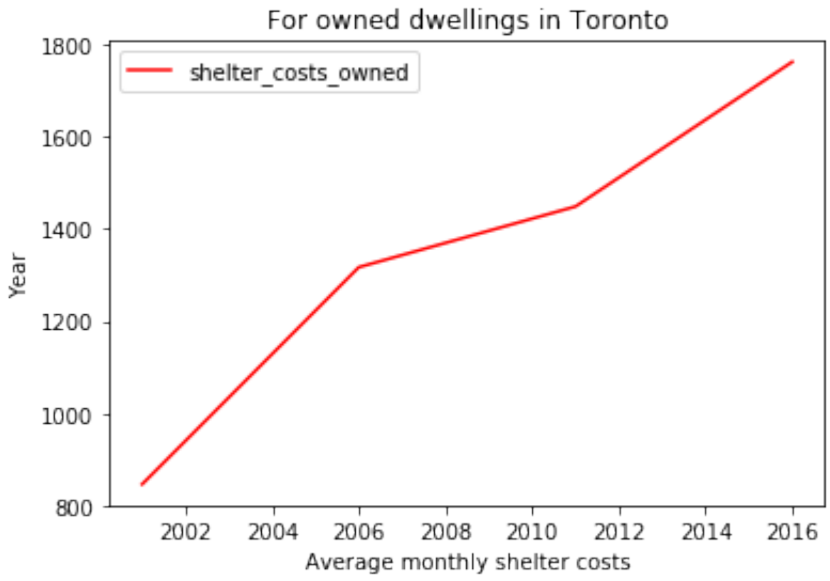
In [14]:

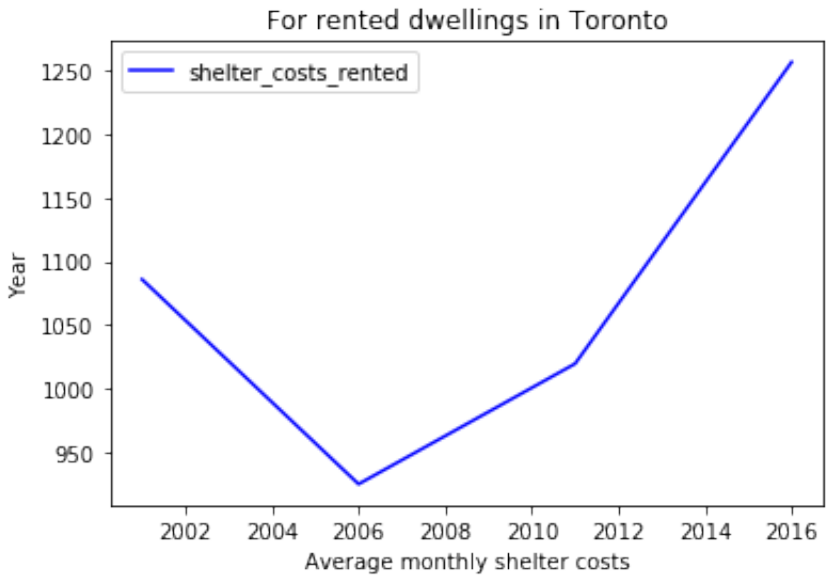
```
# Helper create_line_chart function
def create_line_chart(data, title, xlabel, ylabel, color):
    """
    Create a line chart based in the data argument.
    """
    return data.plot(title=title, ylabel=ylabel,xlabel=xlabel,color=color)
```

In [15]:

```
# Create two line charts, one to plot the monthly shelter costs for owned dwe

# Line chart for owned dwellings
plot_shelter_costs_owned = create_line_chart(df_monthly_shelter_costs[["shelte

#title="Average monthly shelter costs for owned dwellings in Toronto"
#    elif title=="rented":
#        title="Average monthly shelter costs for rented dwellings in Toronto

# Line chart for rented dwellings
plot_shelter_costs_rented = create_line_chart(df_monthly_shelter_costs[["shel

pn_shelter_costs_column = pn.Column("## Average monthly shelter costs", plot_
pn_shelter_costs_column
```

Out[15]:





Average House Value per Year

In this section, you want to determine the average house value per year. An investor may want to understand better the sales price of the rental property over time. For example, a customer will want to know if they should expect an increase or decrease in the property value over time so they can determine how long to hold the rental property. You will visualize the `average_house_value` per year as a bar chart.

In [16]:

```
# Calculate the average house value per year
to_data = pd.read_csv(file_path, index_col=["year","neighbourhood"])
average_house_value = to_data["average_house_value"].groupby("year").mean()
average_house_value.head()
```

Out[16]:

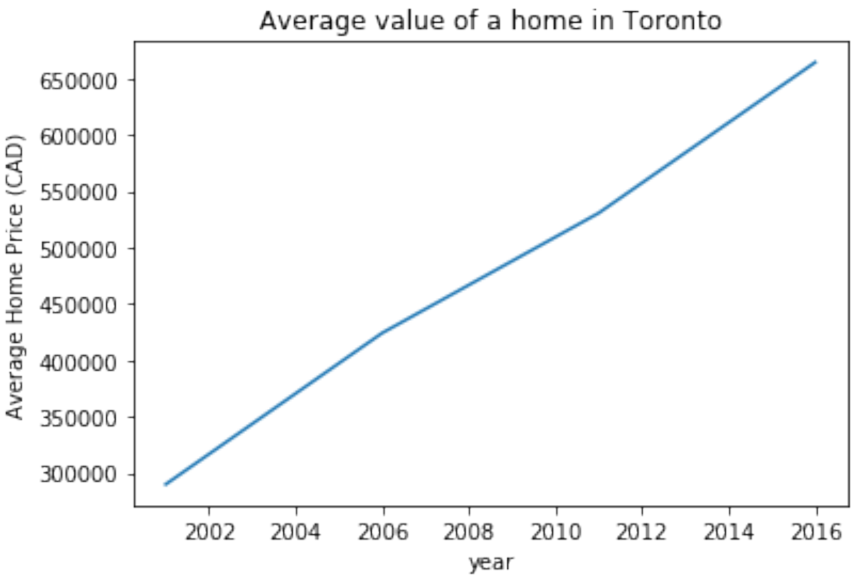
```
year
2001    289882.885714
2006    424059.664286
2011    530424.721429
2016    664068.328571
Name: average_house_value, dtype: float64
```

In [17]:

```
# Plot the average house value per year as a line chart
average_house_value.plot(kind="line", title="Average value of a home in Toron
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb1130a0a10>
```



Average House Value by Neighbourhood

In this section, you will use `hvplot` to create an interactive visualization of the average house value with a dropdown selector for the neighbourhood.

Hint: It will be easier to create a new DataFrame from grouping the data and calculating the mean house values for each year and neighbourhood.

In [18]:

```
# Create a new DataFrame with the mean house values by neighbourhood per year
avg_home_value_per_neighbourhood = to_data["average_house_value"].groupby(by=
avg_home_value_per_neighbourhood.head()
```

Out[18]:

```
year  neighbourhood
2001  Agincourt North                 200388.0
      Agincourt South-Malvern West    203047.0
      Alderwood                       259998.0
      Annex                           453850.0
      Banbury-Don Mills               371864.0
Name: average_house_value, dtype: float64
```

In [19]:

```
# Use hvplot to create an interactive line chart of the average house value p
# The plot should have a dropdown selector for the neighbourhood
avg_home_value_per_neighbourhood.hvplot(kind="line", x="year", y="average_hou
                                         groupby="neighbourhood",widget_location="top_left")
```

Out[19]:

Number of Dwelling Types per Year

In this section, you will use `hvplot` to create an interactive visualization of the average number of dwelling types per year with a dropdown selector for the neighbourhood.

In [20]:

```
# Fetch the data of all dwelling types per year
to_data = pd.read_csv(file_path)
to_data.head()
#removing the average house value for better swelling visulization
avg_dwelling_types = to_data.drop(columns=["average_house_value"])
```

In [21]:

```
# Use hvplot to create an interactive bar chart of the number of dwelling typ
# The plot should have a dropdown selector for the neighbourhood
avg_dwelling_types.hvplot(kind="bar", x='year', rot=90 , xlabel="year",ylabel
```

Out[21]:

The Top 10 Most Expensive Neighbourhoods

In this section, you will need to calculate the house value for each neighbourhood and then sort the values to obtain the top 10 most expensive neighbourhoods on average. Plot the results as a bar chart.

In [22]:

```
# Getting the data from the top 10 expensive neighbourhoods
to_top_10neighbourhoods=to_data.drop(columns=["year"]).groupby(by=["neighbour
to_top_10neighbourhoods.head(10)
```

Out[22]:

	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_
neighbourhood				
Bridle Path-Sunnybrook-York Mills	2260.00	331.25	0.00	
Forest Hill South	1742.50	2031.25	1.25	
Lawrence Park South	3472.50	773.75	0.00	
Rosedale-Moore Park	2498.75	4641.25	0.00	
St.Andrew-Windfields	3225.00	1670.00	0.00	
Casa Loma	916.25	2310.00	0.00	
Bedford Park-Nortown	4865.00	1981.25	0.00	
Forest Hill North	1488.75	3392.50	0.00	
Kingsway South	2326.25	576.25	0.00	
Yonge-St.Clair	565.00	3948.75	0.00	

In [23]:

```
# Plotting the data from the top 10 expensive neighbourhoods
to_top_10neighbourhoods.hvplot(kind="bar",x="neighbourhood", y="average_house_
```

Out[23]:

Neighbourhood Map

In this section, you will read in neighbourhoods location data and build an interactive map with the average house value per neighbourhood. Use a `scatter_mapbox` from Plotly express to create the visualization. Remember, you will need your Mapbox API key for this.

Load Location Data

In [24]:

```
# Load neighbourhoods coordinates data
file_path = Path("Data/toronto_neighbourhoods_coordinates.csv")
df_neighbourhood_locations = pd.read_csv(file_path)
df_neighbourhood_locations.head()
```

Out[24]:

	neighbourhood	lat	lon
0	Agincourt North	43.805441	-79.266712
1	Agincourt South-Malvern West	43.788658	-79.265612
2	Alderwood	43.604937	-79.541611
3	Annex	43.671585	-79.404001
4	Banbury-Don Mills	43.737657	-79.349718

Data Preparation

You will need to join the location data with the mean values per neighbourhood.

1. Calculate the mean values for each neighbourhood.

2. Join the average values with the neighbourhood locations.

In [25]:

```
# Calculate the mean values for each neighborhood
df_to_avg_house_prices= to_data.drop(columns=["year"]).groupby(by=["neighbour
df_to_avg_house_prices.head()
```

Out[25]:

	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_
neighbourhood				
Agincourt North	3435.00	1947.50	2.50	
Agincourt South-Malvern West	2897.50	2180.00	1.25	
Alderwood	2903.75	302.50	1.25	
Annex	751.25	7235.00	1.25	
Banbury-Don Mills	3572.50	5388.75	1.25	

In [26]:

```
# Join the average values with the neighbourhood locations
df_to_avg_house_prices_with_location = pd.merge(df_neighbourhood_locations,df_
#df_to_avg_house_prices_with_location.set_index("neighbourhood", inplace=True
df_to_avg_house_prices_with_location.head()
```

Out[26]:

	neighbourhood	lat	lon	single_detached_house	apartment_five_storeys_plus
0	Agincourt North	43.805441	-79.266712	3435.00	1947.5
1	Agincourt South-Malvern West	43.788658	-79.265612	2897.50	2180.0
2	Alderwood	43.604937	-79.541611	2903.75	302.5
3	Annex	43.671585	-79.404001	751.25	7235.0
4	Banbury-Don Mills	43.737657	-79.349718	3572.50	5388.7

Mapbox Visualization

Plot the average values per neighbourhood using a Plotly express `scatter_mapbox` visualization.

In [36]:

```
# Create a scatter mapbox to analyze neighbourhood info
to_neighbourhoods = px.scatter_mapbox(
    df_to_avg_house_prices_with_location,
    lat="lat",
    lon="lon",
    size="average_house_value",
    color="average_house_value",
    color_continuous_scale=px.colors.cyclical.IceFire,
    zoom=10,
    title="Toronto's neighbourhoods avg. house prices",
    width=1000,
)
```

In [37]:

```
map_column = pn.Column("## Scatter Mapbox To analyze neighbourhood info", to_
```

In [38]:

```
map_column
```

Out[38]:

Cost Analysis - Optional Challenge

In this section, you will use Plotly express to a couple of plots that investors can interactively filter and explore various factors related to the house value of the Toronto's neighbourhoods.

Create a bar chart row facet to plot the average house values for all Toronto's neighbourhoods per year

In [30]:

```
# YOUR CODE HERE!
```

Create a sunburst chart to conduct a costs analysis of most expensive neighbourhoods in Toronto per year

In [31]:

```
# Fetch the data from all expensive neighbourhoods per year.
# YOUR CODE HERE!
```

In [32]:

```
# Create the sunburst chart
# YOUR CODE HERE!
```