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 [92]:   # imports
           import panel as pn
           pn.extension('plotly')
           import plotly.express as px
           import pandas as pd
           import hvplot.pandas
           import matplotlib.pyplot as plt
           import os
           from pathlib import Path
           from dotenv import load_dotenv
```

```
In [145…   # Read the Mapbox API key
           load_dotenv()
           map_box_api = os.getenv("mapbox")
```

Load Data

```
In [94]:   # 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.head()
```

Out[94]:

	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling	s
year					
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.

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

In [95]:
```python
# Calculate the sum number of dwelling types units per year (hint: use groupb
# YOUR CODE HERE!

df_dwelling_types = to_data.groupby(to_data.index).sum()
df_dwelling_types = df_dwelling_types.iloc[:,0:8]
df_dwelling_types
```

Out[95]:

year	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detached_h
2001	300930	355015	75	90
2006	266860	379400	165	69
2011	274940	429220	100	72
2016	269680	493270	95	7

In [96]:
```python
# Save the dataframe as a csv file
# YOUR CODE HERE!
df_dwelling_types.to_csv('Data/dwelling_types_per_year.csv')
```

In [97]:
```python
# Helper create_bar_chart function
def create_bar_chart(data, title, xlabel, ylabel, color):
    """
    Create a barplot based in the data argument.
    """

    fig_dwelling_types = plt.figure()
    plot_dwelling_types = data.plot.bar(color=color)
    plot_dwelling_types.set_xlabel(xlabel)
    plot_dwelling_types.set_ylabel(ylabel)
    plot_dwelling_types.set_title(title)
    plt.show()
```

In [98]:
```python
# Create a bar chart per year to show the number of dwelling types

# Bar chart for 2001
# YOUR CODE HERE!
create_bar_chart(df_dwelling_types.loc[2001],"Dwelling Types on 2001","Dwelli

# Bar chart for 2006
create_bar_chart(df_dwelling_types.loc[2006],"Dwelling Types on 2006","Dwelli

# Bar chart for 2011
create_bar_chart(df_dwelling_types.loc[2011],"Dwelling Types on 2011","Dwelli

# Bar chart for 2016
create_bar_chart(df_dwelling_types.loc[2016],"Dwelling Types on 2016","Dwelli

##If want to include line charts for the similar visualization
#create_line_chart(df_dwelling_types.loc[2001],"Dwelling Types on 2001","Dwel

# Bar chart for 2006
#create_line_chart(df_dwelling_types.loc[2006],"Dwelling Types on 2006","Dwel

# Bar chart for 2011
#create_line_chart(df_dwelling_types.loc[2011],"Dwelling Types on 2011","Dwel
```
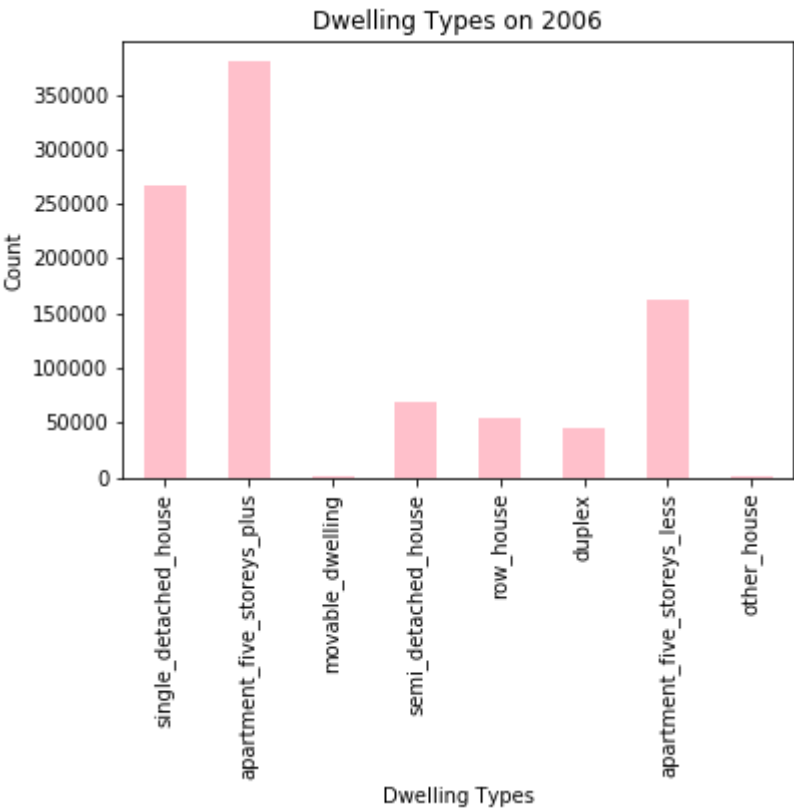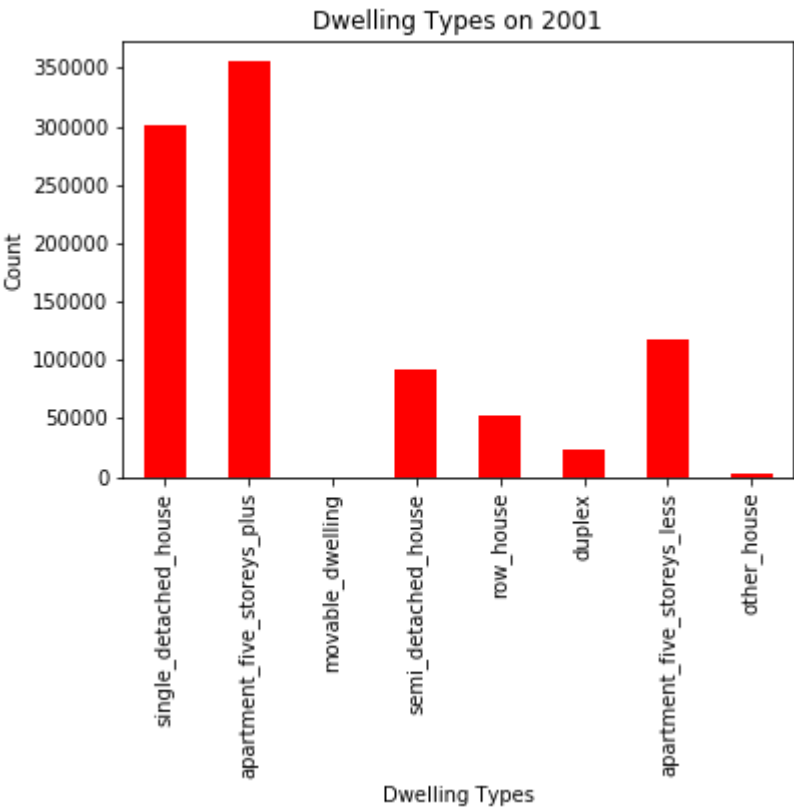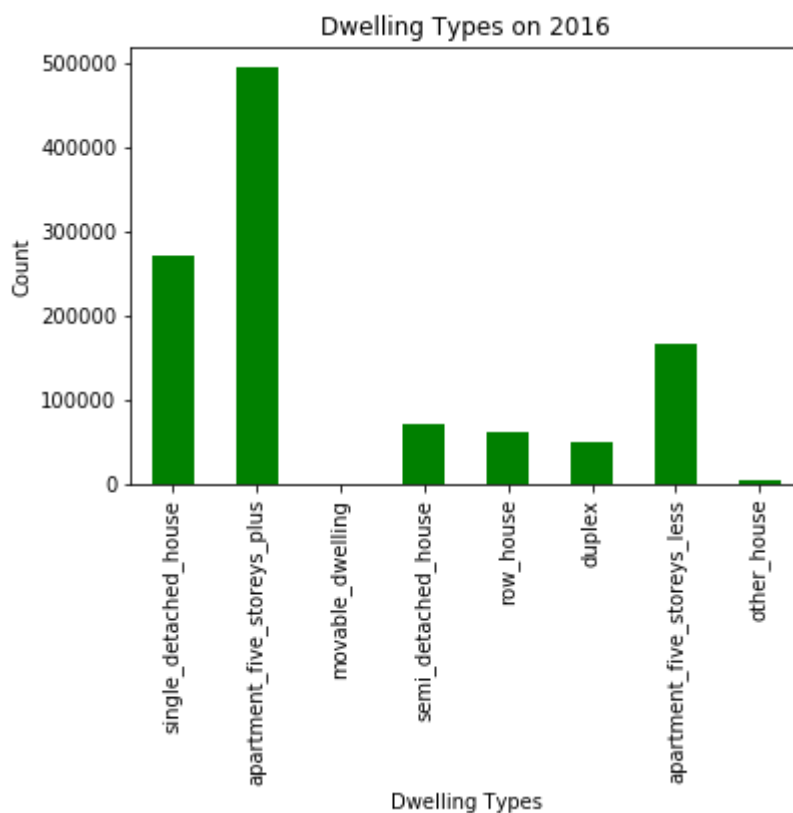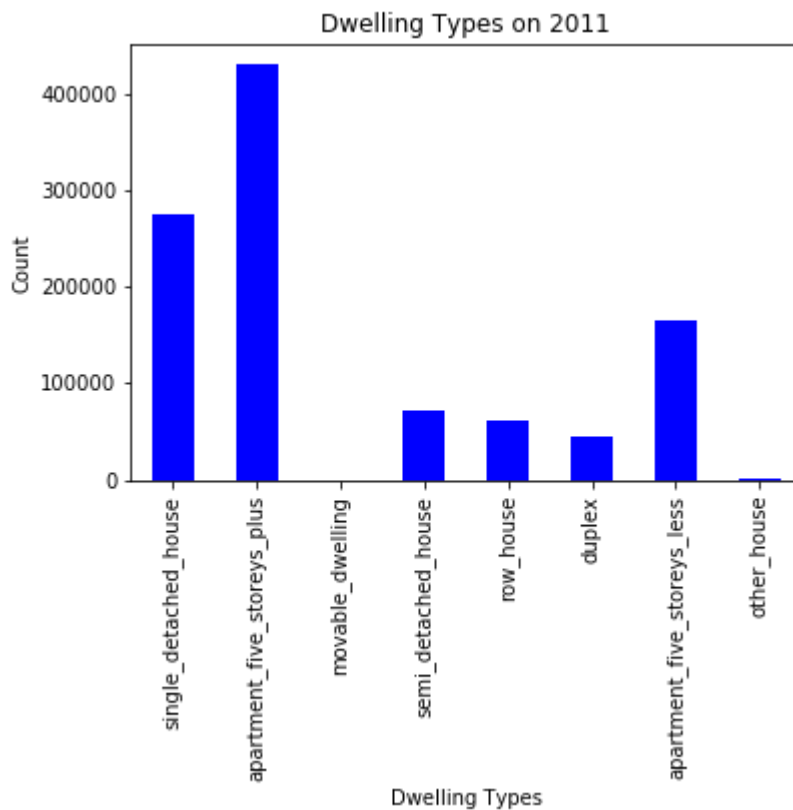
```
# Bar chart for 2016
#create_line_chart(df_dwelling_types.loc[2016],"Dwelling Types on 2016","Dwel
```

Dwelling Types on 2001

Dwelling Types on 2006

Dwelling Types on 2011

Dwelling Types on 2016

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 [99]:    # Calculate the average monthly shelter costs for owned and rented dwellings
```

```
# YOUR CODE HERE!

df_costs = to_data[["shelter_costs_owned", "shelter_costs_rented"]].groupby(t
df_costs.head()
```

Out[99]:

|	shelter_costs_owned	shelter_costs_rented
year		
2001	846.878571	1085.935714
2006	1316.800000	925.414286
2011	1448.214286	1019.792857
2016	1761.314286	1256.321429

In [100…

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

    fig_dwelling_types = plt.figure()
    plot_dwelling_types = data.plot(color=color)
    plot_dwelling_types.set_xlabel(xlabel)
    plot_dwelling_types.set_ylabel(ylabel)
    plot_dwelling_types.set_title(title)
    plt.xticks(rotation=90)
    plt.show()
```

In [101…

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

# Line chart for owned dwellings
# YOUR CODE HERE!
create_line_chart(df_costs.shelter_costs_owned,"Average Monthly Shelter Costs
# Line chart for rented dwellings
# YOUR CODE HERE!
create_line_chart(df_costs.shelter_costs_rented,"Average Monthly Shelter Cost
```

Average Monthly Shelter Costs For Rentend Dwelleing
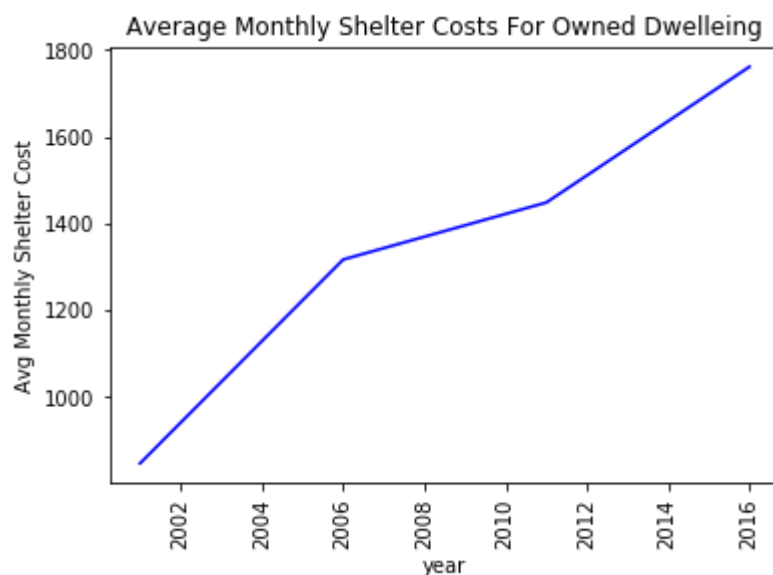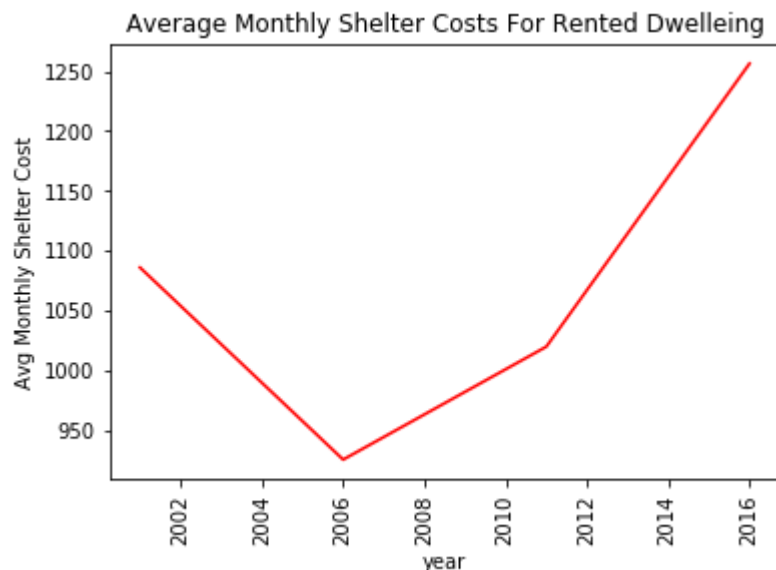
Average House Value per Year

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 [102…
```
# Calculate the average house value per year
# YOUR CODE HERE!

average_house_value = to_data["average_house_value"].groupby(to_data.index).me
average_house_value
```

Out[102…
```
year
2001    289882.885714
2006    424059.664286
2011    530424.721429
2016    664068.328571
Name: average_house_value, dtype: float64
```

In [103…
```
# Plot the average house value per year as a line chart
# YOUR CODE HERE!

create_line_chart(average_house_value, "Average House Value", "Year","Average
```

Average House Value
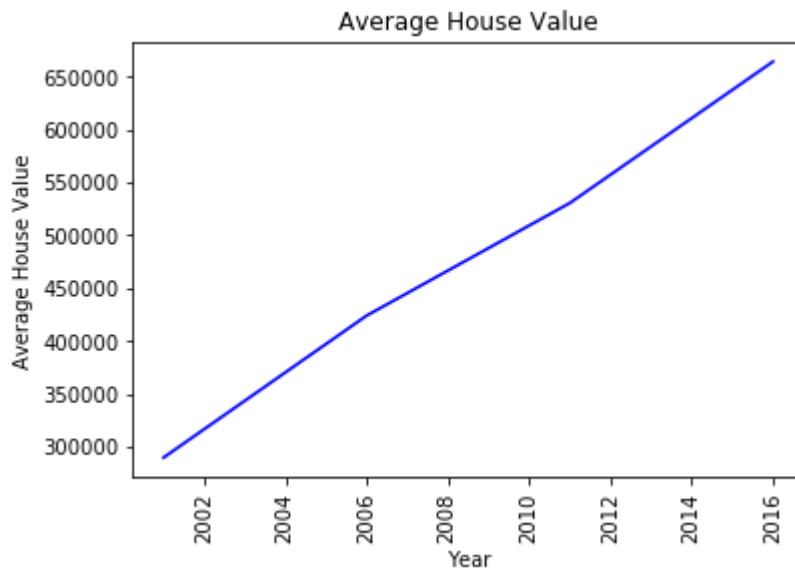
Average House Value by Neighbourhood

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 [104…
```python
# Create a new DataFrame with the mean house values by neighbourhood per year
# YOUR CODE HERE!

avg_HV_df = to_data.reset_index()
avg_HV_df = avg_HV_df[["year","average_house_value","neighbourhood"]]
avg_HV_df = avg_HV_df.groupby(['neighbourhood','year']).mean()
avg_HV_df.reset_index(inplace=True)
avg_HV_df.head()
```

Out[104…

	neighbourhood	year	average_house_value
0	Agincourt North	2001	200388
1	Agincourt North	2006	293140
2	Agincourt North	2011	366667
3	Agincourt North	2016	459051
4	Agincourt South-Malvern West	2001	203047

In [105…
```python
# Use hvplot to create an interactive line chart of the average house value p
# The plot should have a dropdown selector for the neighbourhood
# YOUR CODE HERE!

avg_HV_df.hvplot.line(
    'year',
    'average_house_value',
    xlabel = "Year",
    ylabel = "Average House Value",
    groupby='neighbourhood',
)
```

Out[105…

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 [106…
```python
# Fetch the data of all dwelling types per year
# YOUR CODE HERE!

dwe_svg_df = to_data.groupby([to_data.index,'neighbourhood']).mean().reset_in
dwe_svg_df.head()
```

Out[106…

	year	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling
0	2001	Agincourt North	3715	1480	0
1	2001	Agincourt South-Malvern West	3250	1835	0
2	2001	Alderwood	3175	315	0
3	2001	Annex	1060	6090	5
4	2001	Banbury-Don Mills	3615	4465	0

In [107…
```python
# Use hvplot to create an interactive bar chart of the number of dwelling typ
# The plot should have a dropdown selector for the neighbourhood
# YOUR CODE HERE!

dwe_svg_df.hvplot.bar(
    'year',
    ['single_detached_house',
     'apartment_five_storeys_plus',
     'movable_dwelling',
     'semi_detached_house',
     'row_house',
     'duplex',
     'apartment_five_storeys_less',
     'other_house'],
    xlabel = "Year",
    ylabel = "Dwelling Units",
    groupby='neighbourhood',
    height=500,
    rot=125
)
```

Out[107…

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 [108…
```python
# Getting the data from the top 10 expensive neighbourhoods
# YOUR CODE HERE!

most_exp_df = to_data.groupby('neighbourhood').mean()
most_exp_df = most_exp_df.sort_values(by='average_house_value',ascending=False
most_exp_df
```

Out[108…

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

In [109…
```python
# Plotting the data from the top 10 expensive neighbourhoods
# YOUR CODE HERE!
most_exp_df.hvplot.bar(
    'neighbourhood',
    'average_house_value',
    xlabel = "Neighbourhood",
    ylabel = "Average House Value",
    title='Top 10 Expensive Neighbourhoods in Toronto',
    height=500,
    rot=125,
)
```

Out[109…

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 [110…    # 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[110…

	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 [111…    # Calculate the mean values for each neighborhood
            # YOUR CODE HERE!

            mean_neigh_df = to_data.groupby('neighbourhood').mean().reset_index()
            mean_neigh_df.head()
```

Out[111…

	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi
0	Agincourt North	3435.00	1947.50	2.50	
1	Agincourt South-Malvern West	2897.50	2180.00	1.25	
2	Alderwood	2903.75	302.50	1.25	
3	Annex	751.25	7235.00	1.25	
4	Banbury-Don Mills	3572.50	5388.75	1.25	

```
In [112…    # Join the average values with the neighbourhood locations
            # YOUR CODE HERE!

            neigh_loc_combo_df = pd.merge(
                df_neighbourhood_locations,
                mean_neigh_df,
                on='neighbourhood',
                how='inner')
            neigh_loc_combo_df.head()
```

Out[112…

neighbourhood	lat	lon	single_detached_house	apartment_five_storeys_plus

	neighbourhood	lat	lon	single_detached_house	apartment_five_storeys_plus
0	Agincourt North	43.805441	-79.266712	3435.00	1947.50
1	Agincourt South-Malvern West	43.788658	-79.265612	2897.50	2180.00
2	Alderwood	43.604937	-79.541611	2903.75	302.50
3	Annex	43.671585	-79.404001	751.25	7235.00
4	Banbury-Don Mills	43.737657	-79.349718	3572.50	5388.75

Mapbox Visualization

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

In [113…
```python
# Create a scatter mapbox to analyze neighbourhood info
# YOUR CODE HERE!


px.set_mapbox_access_token(map_box_api)
neighbourhood_map = px.scatter_mapbox(
neigh_loc_combo_df,
lat="lat",
lon="lon",
size='average_house_value',
color='average_house_value',
title='Average House Value in Toronto',
size_max=30,
zoom=10
)
neighbourhood_map.show()
```

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 [114…
```python
# YOUR CODE HERE!
```

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

In [142…
```python
# Fetch the data from all expensive neighbourhoods per year.
# YOUR CODE HERE!

most_exp_neigh = to_data.groupby([to_data.index,'neighbourhood']).mean()
most_exp_neigh = most_exp_neigh.sort_values(by='average_house_value',ascendin
most_exp_neigh
```

Out[142…

	year	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling
0	2016	Bridle Path–Sunnybrook–York Mills	2275	590	0
1	2011	Bridle Path–Sunnybrook–York Mills	2285	480	0
2	2016	Forest Hill South	1685	2025	0
3	2016	Lawrence Park South	3420	925	0
4	2016	Rosedale–Moore Park	2450	4990	0

In [131…
```python
# Create the sunburst chart
# YOUR CODE HERE!


#data = dict(
    #character=["Eve", "Cain", "Seth", "Enos", "Noam", "Abel", "Awan", "Enoch
    #parent=["", "Eve", "Eve", "Seth", "Seth", "Eve", "Eve", "Awan", "Eve" ],
    #value=[10, 14, 12, 10, 2, 6, 6, 4, 4])

#fig =px.sunburst(
```

```python
    #data,
    #names='character',
    #parents='parent',
    #values='value',
#)
#fig.show()

import plotly.express as px

fig = px.sunburst(
    most_exp_neigh,
    names = 'neighbourhood',
    parents = 'year',
    values = 'average_house_value',
    title='Cost Analysis of Most Expensive Neighbourhoods in Toronto per Year
)
fig.show()
```

In [ ]: