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
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 [2]:
```
# Read the Mapbox API key
load_dotenv()
map_box_api = os.getenv("MAP_BOX_API")
px.set_mapbox_access_token(map_box_api)
type(map_box_api)
```

Out[2]: str

Load Data

In [3]:
```
# 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[3]:

year	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detac
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 [4]:
```
# Calculate the sum number of dwelling types units per year (hint: use groupby)
dwelling_units_sum = to_data.groupby('year').sum()

# Remove excess columns to obtain the desired output
dwelling_units_sum = dwelling_units_sum.drop(["average_house_value", "shelter_costs_own
dwelling_units_sum.head()
```

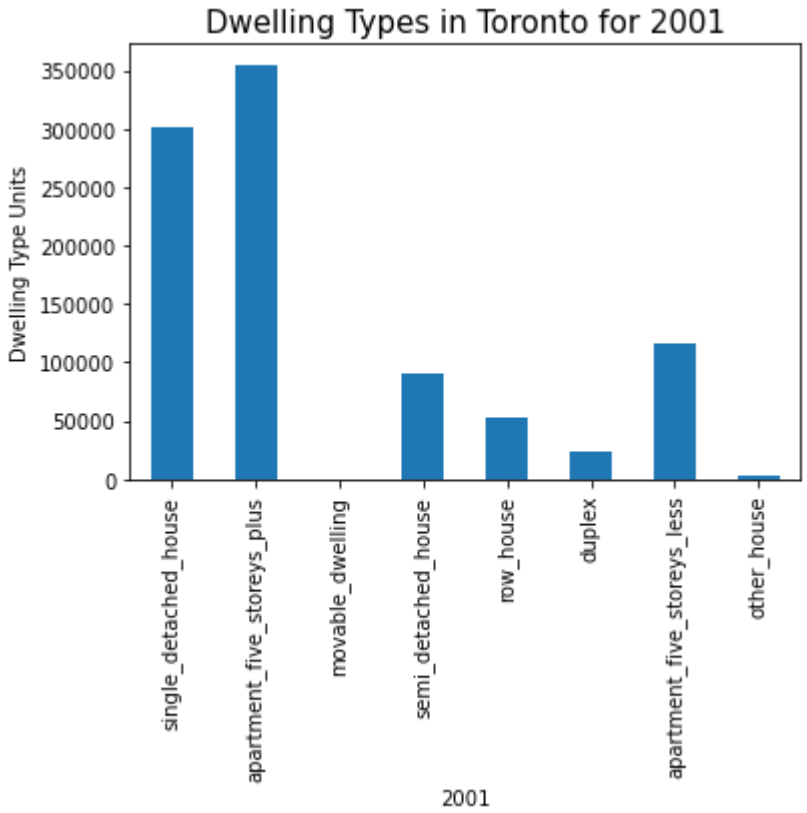
Out[4]:

year	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detached_house	row_
2001	300930	355015	75	90995	
2006	266860	379400	165	69430	
2011	274940	429220	100	72480	
2016	269680	493270	95	71200	

In [5]:
```
# Save the dataframe as a csv file
dwelling_units = dwelling_units_sum.to_csv("dwelling_units.csv")
```

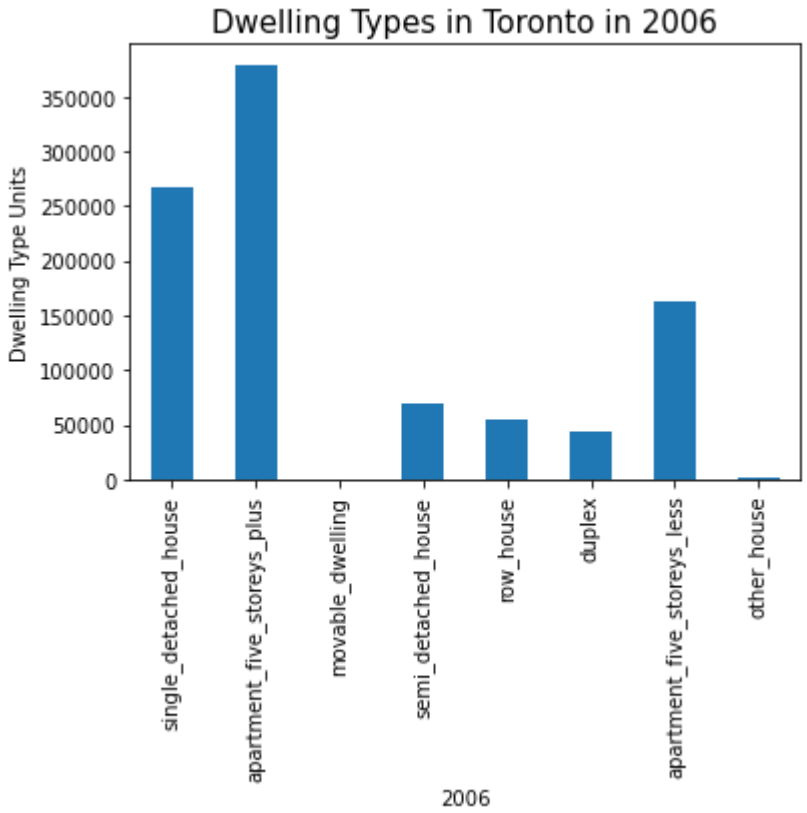
In [6]:
```
# Helper create_bar_chart function
def create_bar_chart(data, title, xlabel, ylabel, color):
    chart = data.plot.bar()
    chart.set_title(title, fontsize=15)
    chart.set_xlabel(xlabel, fontsize=10)
    chart.set_ylabel(ylabel, fontsize=10)
    figure = plt.figure()
    plt.show()
    return create_bar_chart
```

In [7]:
```
# Create a bar chart per year to show the number of dwelling types
# Bar chart for 2001
dwellings_2001 = create_bar_chart(dwelling_units_sum.loc[2001], "Dwelling Types in Toro
```




```
<Figure size 432x288 with 0 Axes>
```

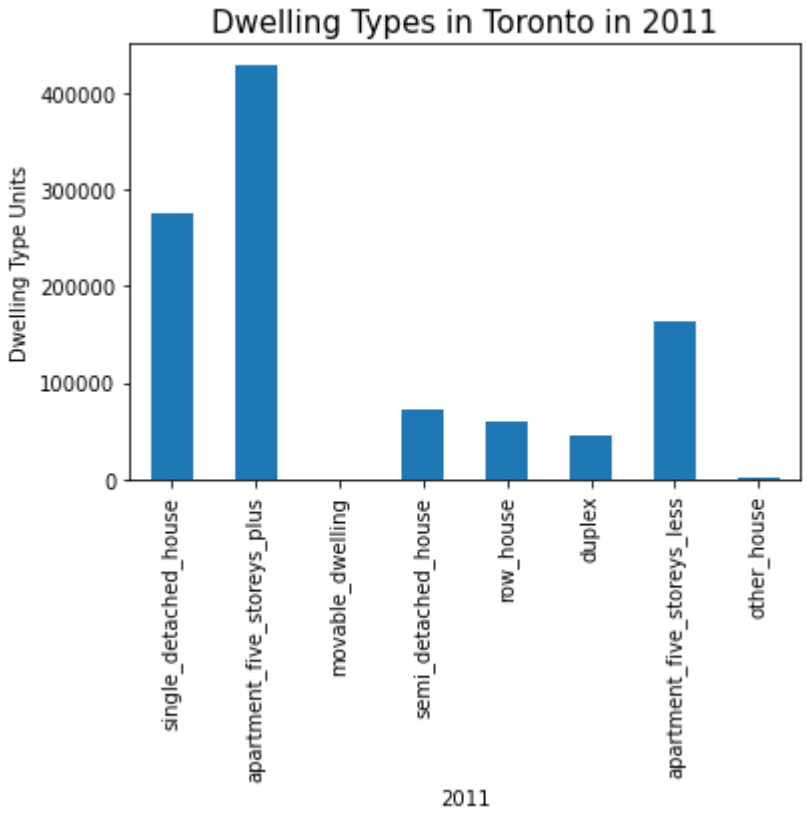
In [8]:
```
# Bar chart for 2006
dwellings_2006 = create_bar_chart(dwelling_units_sum.loc[2006], "Dwelling Types in Toro
```




```
<Figure size 432x288 with 0 Axes>
```

In [9]:
```
# Bar chart for 2011
```

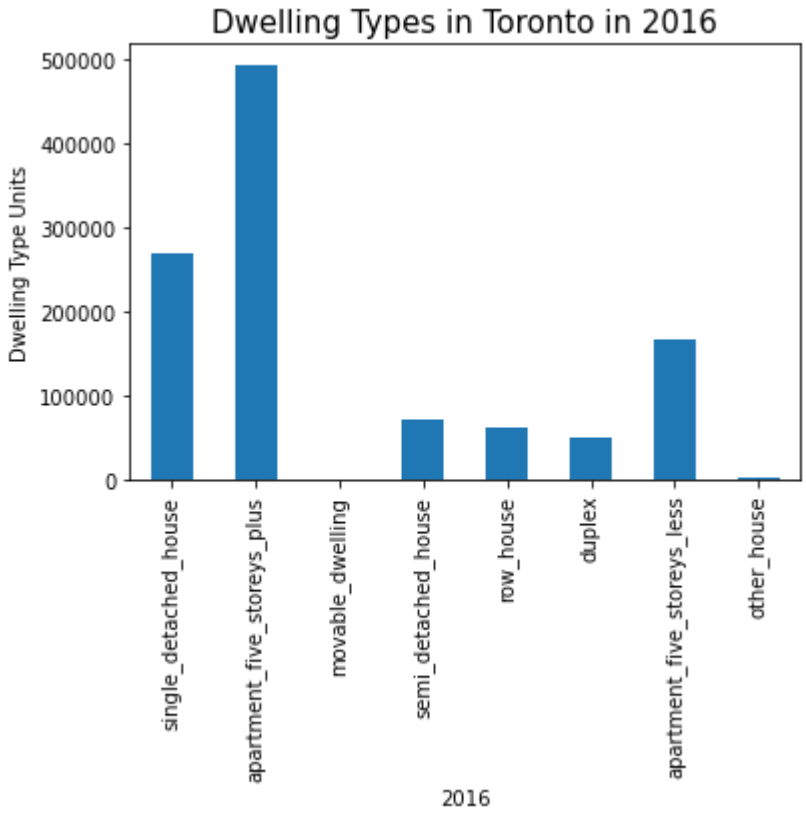
```
create_bar_chart(dwelling_units_sum.loc[2011], "Dwelling Types in Toronto in 2011", "20
```




```
<Figure size 432x288 with 0 Axes>
```

Out[9]:
```
<function __main__.create_bar_chart(data, title, xlabel, ylabel, color)>
```

In [10]:
```
# Bar chart for 2016
create_bar_chart(dwelling_units_sum.loc[2016], "Dwelling Types in Toronto in 2016", "20
```




```
<Figure size 432x288 with 0 Axes>
```

Out[10]:
```
<function __main__.create_bar_chart(data, title, xlabel, ylabel, color)>
```

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 [11]:
```
# Calculate the average monthly shelter costs for owned and rented dwellings
average_costs_all = to_data.groupby("year").mean()

for col in average_costs_all.columns:
    print(col)
```

```
single_detached_house
apartment_five_storeys_plus
movable_dwelling
semi_detached_house
row_house
duplex
apartment_five_storeys_less
other_house
average_house_value
shelter_costs_owned
shelter_costs_rented
```

In [12]:
```
# Remove Columns
average_costs_shelter = average_costs_all.drop(["single_detached_house", "apartment_fiv
```

```
average_costs_shelter.head()
```

Out[12]:

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

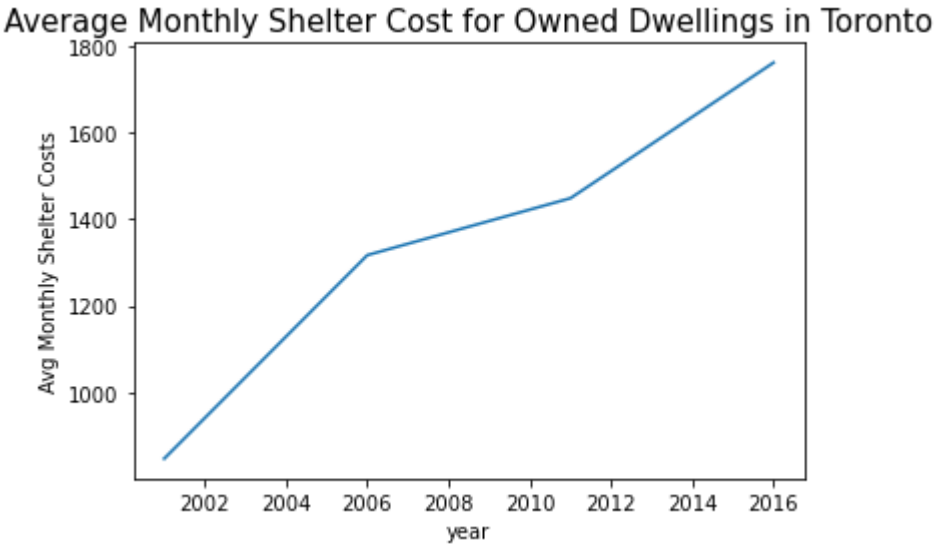
In [13]:

```
# Helper create_line_chart function
def create_line_chart(data, title, xlabel, ylabel, color):
    chart = data.plot.line()
    chart.set_title(title, fontsize=15)
    chart.set_xlabel(xlabel, fontsize=10)
    chart.set_ylabel(ylabel, fontsize=10)
    figure = plt.figure()
    plt.show()
    return create_line_chart
```

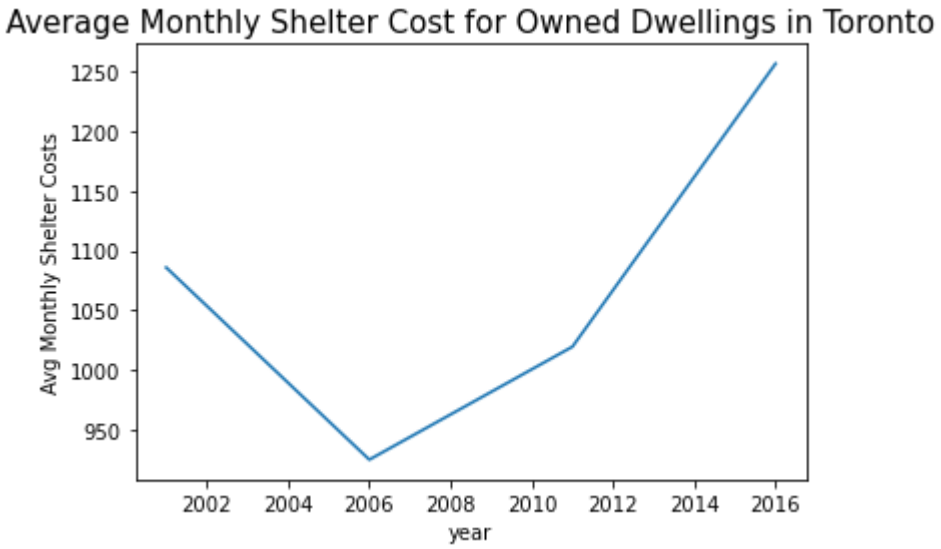
In [14]:

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

# Owned Dwellings
create_line_chart(average_costs_shelter.loc[:, "shelter_costs_owned"], "Average Monthly
# Rented Dwellings
create_line_chart(average_costs_shelter.loc[:, "shelter_costs_rented"], "Average Monthl
```




```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

Out[14]:
```
<function __main__.create_line_chart(data, title, xlabel, ylabel, color)>
```

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 [15]:
```
# Calculate the average house value per year
average_costs_all = to_data.groupby("year").mean()
for col in average_costs_all.columns:
    print(col)
```

```
single_detached_house
apartment_five_storeys_plus
movable_dwelling
semi_detached_house
row_house
duplex
apartment_five_storeys_less
other_house
average_house_value
shelter_costs_owned
shelter_costs_rented
```

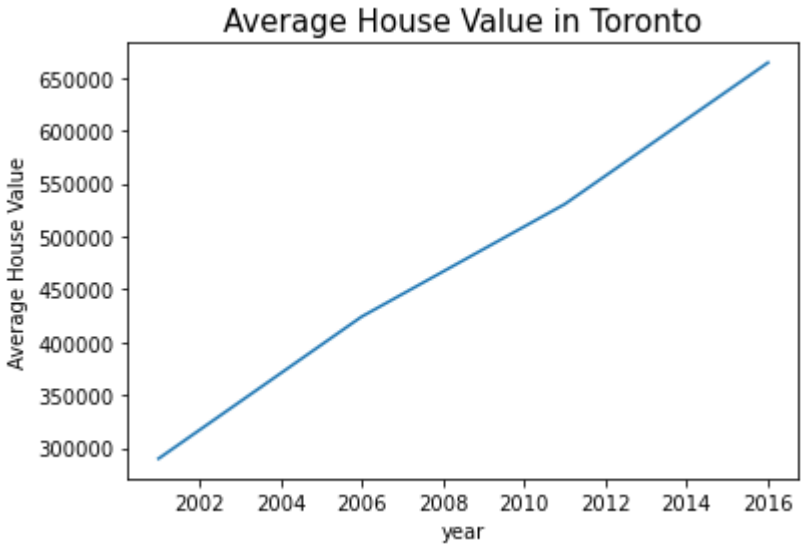
In [16]:
```
# Remove Columns
average_house_yearly = average_costs_all.filter(["average_house_value"], axis = 1)
average_house_yearly.head()
```

Out[16]:

year	average_house_value
2001	289882.885714
2006	424059.664286

year	average_house_value
2011	530424.721429
2016	664068.328571

In [17]:
```
# Plot the average house value per year as a line chart
create_line_chart(average_house_yearly.loc[:, "average_house_value"], "Average House Va
```




```
<Figure size 432x288 with 0 Axes>
```

Out[17]:
```
<function __main__.create_line_chart(data, title, xlabel, ylabel, color)>
```

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
neighbourhood_value = to_data.filter(["neighbourhood", "average_house_value"], axis = 1
neighbourhood_value.head()
```

Out[18]:

year	neighbourhood	average_house_value
2001	Agincourt North	200388
2001	Agincourt South-Malvern West	203047
2001	Alderwood	259998
2001	Annex	453850

year	neighbourhood	average_house_value
2001	Banbury-Don Mills	371864

In [19]:
```
# Use hvplot to create an interactive line chart of the average house value per neighbo
# The plot should have a dropdown selector for the neighbourhood
neighbourhood_value.hvplot.line(title = "Average Value of Dwellings Types per year", gr
```

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
average_dwelling_types = to_data.drop(['average_house_value','shelter_costs_owned', 'sh
average_dwelling_types.head()
```

Out[20]:

year	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detac
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	

In [21]:
```
# Use hvplot to create an interactive bar chart of the number of dwelling types per nei
# The plot should have a dropdown selector for the neighbourhood
average_dwelling_types.hvplot.bar(title = "Number of Dwelling Types per year", groupby=
```

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
top_10 = (to_data[["average_house_value"]].groupby(to_data["neighbourhood"]).mean())
top_10 = top_10.sort_values(by="average_house_value", ascending = False)
top_10_sliced = top_10.head(10)
top_10_sliced.head(11)
```

Out[22]:

neighbourhood	average_house_value
Bridle Path-Sunnybrook-York Mills	1526485.75
Forest Hill South	1195992.50
Lawrence Park South	1094027.75
Rosedale-Moore Park	1093640.00
St.Andrew-Windfields	999107.00
Casa Loma	981064.25
Bedford Park-Nortown	930415.25
Forest Hill North	851680.50
Kingsway South	843234.25
Yonge-St.Clair	813220.25

In [23]:
```
# Plotting the data from the top 10 expensive neighbourhoods
top_10_sliced.hvplot.bar(title = "Top 10 Expensive Neighbourhoods in Toronto", ylabel =
```

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

	neighbourhood	lat	lon
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
average_value_neighbourhood = (to_data[['average_house_value']].groupby(to_data["neighb
average_value_neighbourhood = average_value_neighbourhood.reset_index()
average_value_neighbourhood = average_value_neighbourhood.drop (columns = "neighbourhoo
average_value_neighbourhood.head()
```

Out[25]:

	average_house_value
0	329811.5
1	334189.0
2	427922.5
3	746977.0
4	612039.0

In [26]:
```
# Join the average values with the neighbourhood locations
joined_df = pd.concat([df_neighbourhood_locations, average_value_neighbourhood], join =
joined_df.head(10)
```

Out[26]:

	neighbourhood	lat	lon	average_house_value
0	Agincourt North	43.805441	-79.266712	329811.50
1	Agincourt South-Malvern West	43.788658	-79.265612	334189.00
2	Alderwood	43.604937	-79.541611	427922.50
3	Annex	43.671585	-79.404001	746977.00
4	Banbury-Don Mills	43.737657	-79.349718	612039.00
5	Bathurst Manor	43.764813	-79.456055	501576.75
6	Bay Street Corridor	43.657511	-79.385721	423653.50
7	Bayview Village	43.776361	-79.377117	539258.75

	neighbourhood	lat	lon	average_house_value
8	Bayview Woods-Steeles	43.796802	-79.382118	565413.50
9	Bedford Park-Nortown	43.731486	-79.420227	930415.25

Mapbox Visualization

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

In [27]:

```
# Create a scatter mapbox to analyze neighbourhood info
map_visualization = px.scatter_mapbox(
    joined_df,
    lat="lat",
    lon="lon",
    color="average_house_value",
    size="average_house_value",
    color_continuous_scale = "Icefire",
    title="Average House Values in Toronto",
    zoom = 9
)
map_visualization.show()
```