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 [37]:
```
# 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 [38]:
```
# Read the Mapbox API key
load_dotenv()
```

Out[38]: True

In [39]:
```
map_box_api = os.getenv('mapbox')
px.set_mapbox_access_token(map_box_api)
```

Load Data

In [40]:
```
# 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[40]:

year	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detached_house	row_house	dupl
2001	Agincourt North	3715	1480	0	1055	1295	1
2001	Agincourt South-Malvern West	3250	1835	0	545	455	1
2001	Alderwood	3175	315	0	470	50	1
2001	Annex	1060	6090	5	1980	605	2
2001	Banbury-Don Mills	3615	4465	0	240	380	

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 [41]:
```
# Calculate the sum number of dwelling types units per year (hint: use groupby)
num_dwellings_per_year = to_data[['single_detached_house',
                                  'apartment_five_storeys_plus',
                                  'movable_dwelling',
                                  'semi_detached_house',
                                  'row_house',
                                  'duplex',
                                  'apartment_five_storeys_less',
                                  'other_house']].groupby('year').sum()
num_dwellings_per_year
```

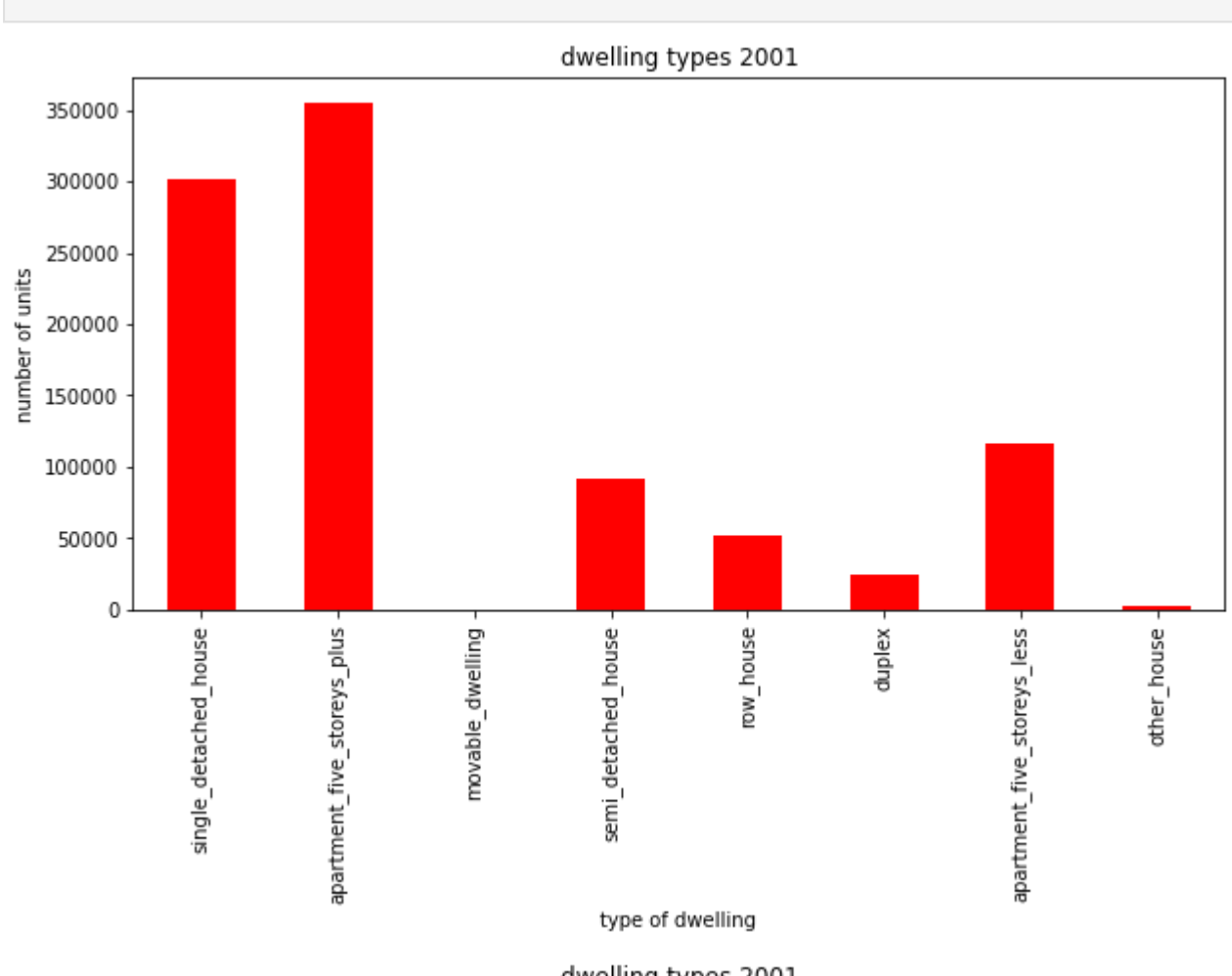
Out[41]:

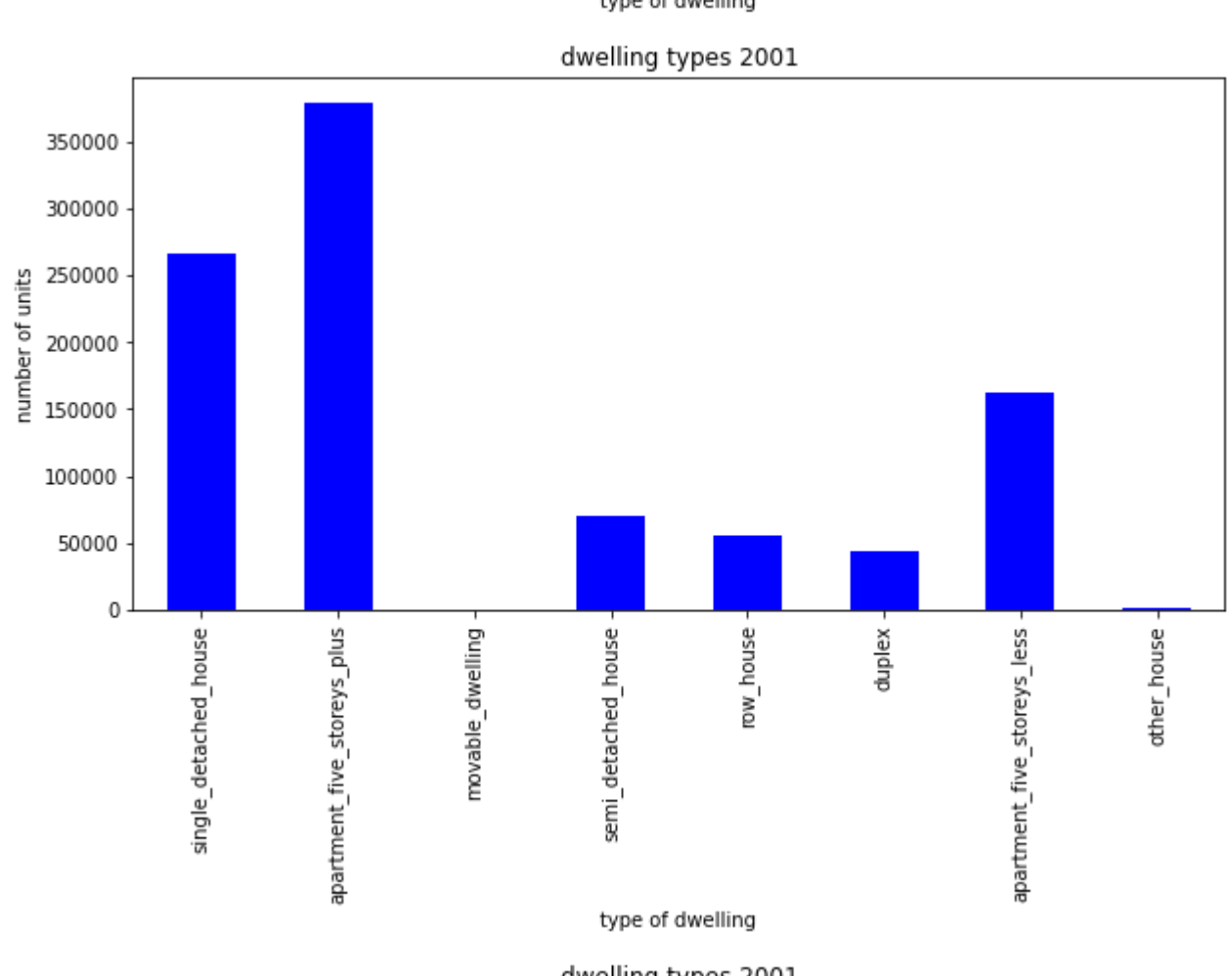
year	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detached_house	row_house	duplex	apartment_f
2001	300930	355015	75	90995	52355	23785	
2006	266860	379400	165	69430	54690	44095	
2011	274940	429220	100	72480	60355	44750	
2016	269680	493270	95	71200	61565	48585	

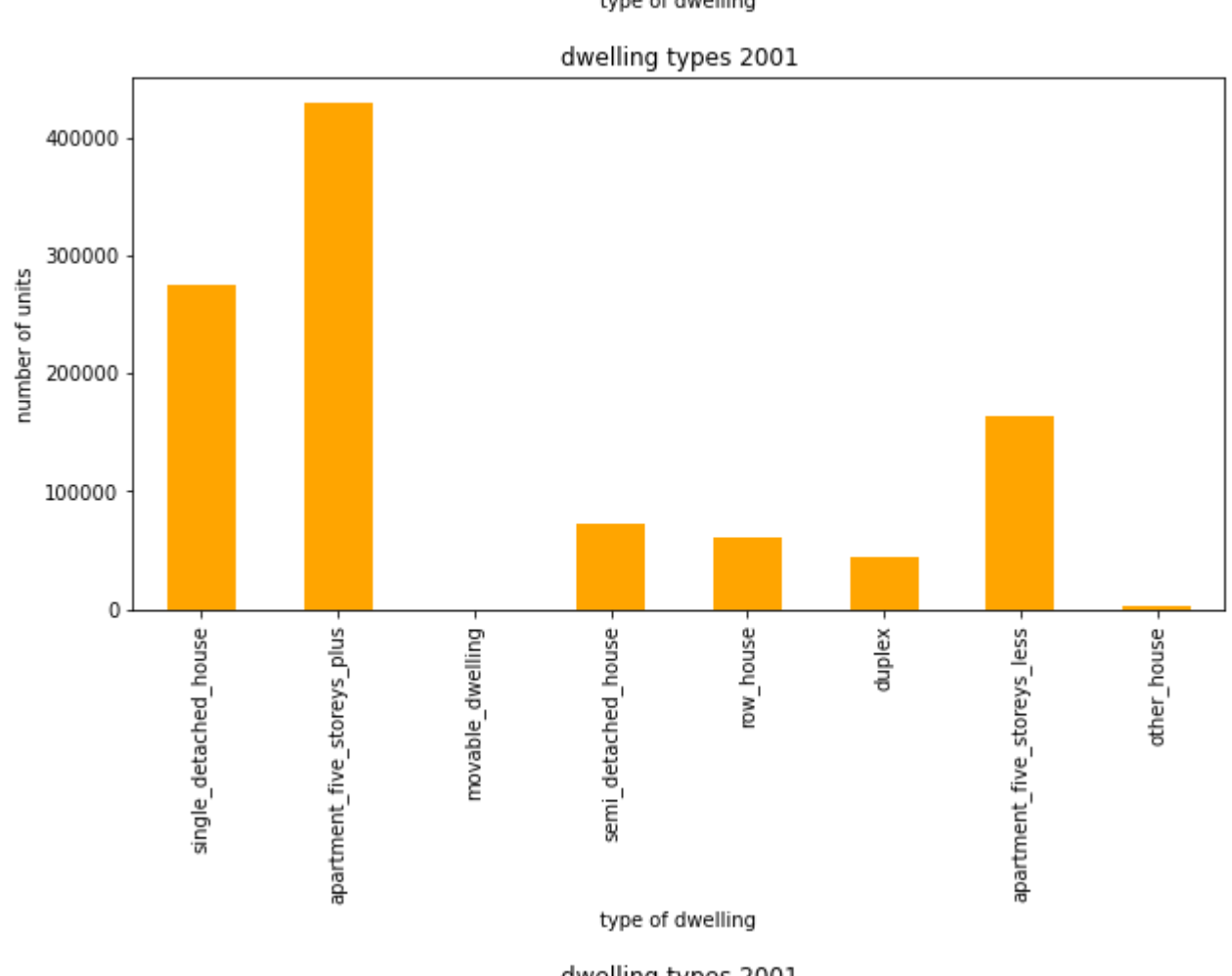
In [42]:
```
# Save the dataframe as a csv file
num_dwellings_per_year.to_csv('dwelling_per_year.csv')
```

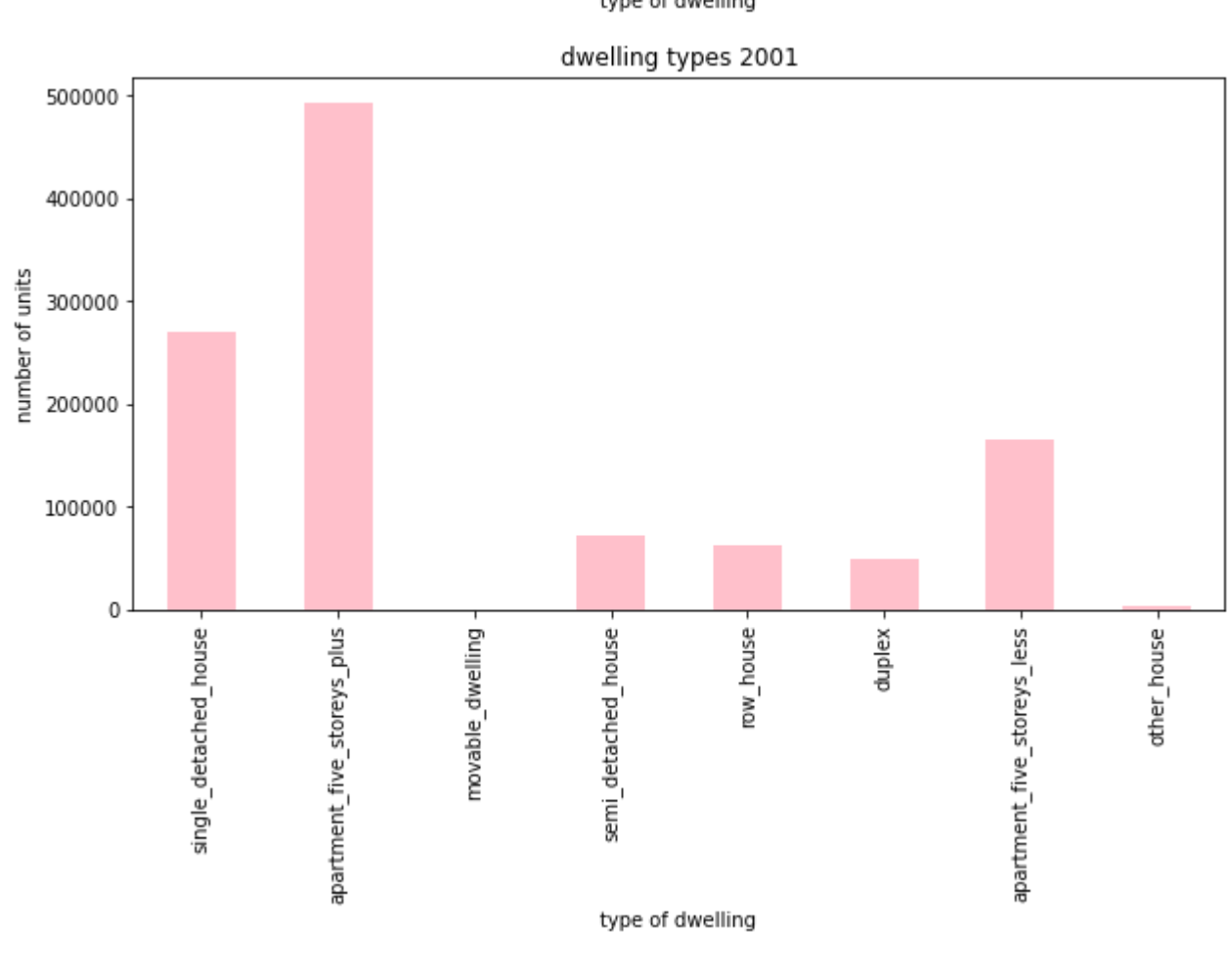
In [43]:
```
# Helper create_bar_chart function
def create_bar_chart(data, title, xlabel, ylabel, color):
    plt.figure(figsize = (10,5))
    data.plot.bar(color=color)
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.show()
```

In [44]:
```
# Create a bar chart per year to show the number of dwelling types

#Bar chart for 2001
create_bar_chart(data=num_dwellings_per_year.loc[2001], title='dwelling types 2001', xlabel='type of dwelling',

# Bar chart for 2006
create_bar_chart(data=num_dwellings_per_year.loc[2006], title='dwelling types 2001', xlabel='type of dwelling',

# Bar chart for 2011
create_bar_chart(data=num_dwellings_per_year.loc[2011], title='dwelling types 2001', xlabel='type of dwelling',

# Bar chart for 2016
create_bar_chart(data=num_dwellings_per_year.loc[2016], title='dwelling types 2001', xlabel='type of dwelling',
```









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 [45]:
```
# Calculate the average monthly shelter costs for owned and rented dwellings
avg_shelter_costs = to_data[['shelter_costs_owned', 'shelter_costs_rented']].groupby('year').mean()
avg_shelter_costs
```

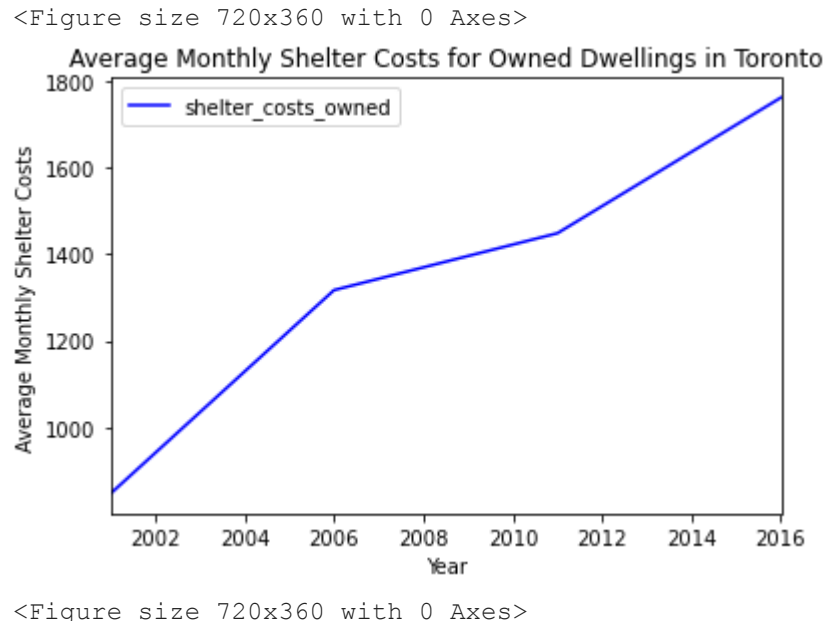
Out[45]:

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

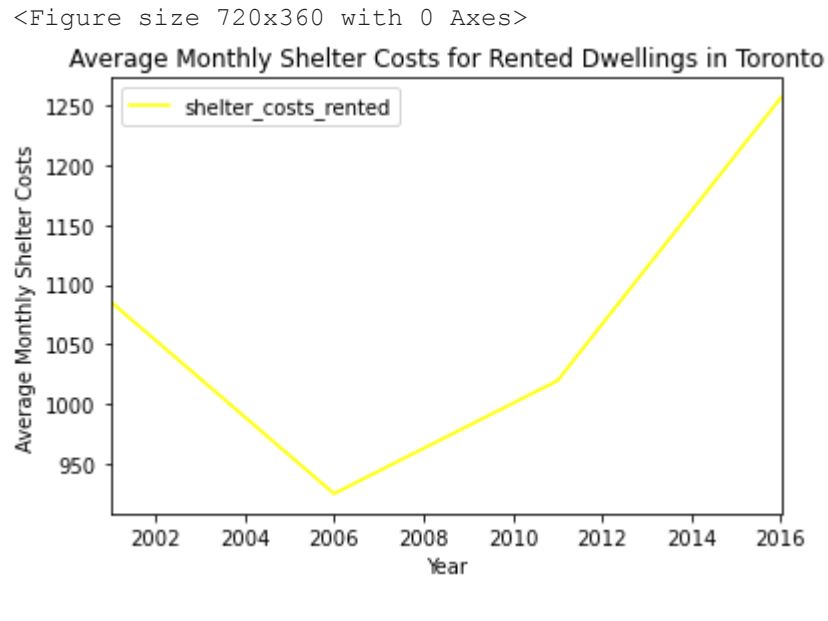
In [46]:
```
# Helper create_line_chart function
def create_line_chart(data, title, xlabel, ylabel, color):
    """
    Create a line chart based in the data argument.
    """
    plt.figure(figsize = (10,5))
    data.plot.line(color=color)
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.show()
```

In [47]:
```
# Create two line charts, one to plot the monthly shelter costs for owned dwelleing and other for rented dwell

# Line chart for owned dwellings
create_line_chart(data=avg_shelter_costs[['shelter_costs_owned']], title='Average Monthly Shelter Costs for Own

# Line chart for rented dwellings
create_line_chart(data=avg_shelter_costs[['shelter_costs_rented']], title='Average Monthly Shelter Costs for Re
```

<Figure size 720x360 with 0 Axes>



<Figure size 720x360 with 0 Axes>



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 [48]:
```
# Calculate the average house value per year
avg_house_value = to_data[['average_house_value']].groupby('year').mean()
avg_house_value
```

Out[48]:

year	average_house_value
2001	289882.885714
2006	424059.664286
2011	530424.721429
2016	664068.328571

In [49]:
```
# Plot the average house value per year as a line chart
avg_house_value.hvplot.line(title='average house value').opts(yformatter="%.0f")
```

Out[49]:

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 [50]:
```
# Create a new DataFrame with the mean house values by neighbourhood per year
to_data_new = to_data[['neighbourhood', 'average_house_value']].groupby([to_data.index, "neighbourhood"]).mean
to_avg_sales_neighbourhood = to_data_new.reset_index()
to_avg_sales_neighbourhood.head(10)
```

Out[50]:

	year	neighbourhood	average_house_value
0	2001	Agincourt North	200388
1	2001	Agincourt South-Malvern West	203047
2	2001	Alderwood	259998
3	2001	Annex	453850
4	2001	Banbury-Don Mills	371864
5	2001	Bathurst Manor	304749
6	2001	Bay Street Corridor	257404
7	2001	Bayview Village	327644
8	2001	Bayview Woods-Steeles	343535
9	2001	Bedford Park-Nortown	565304

In [51]:
```
# Use hvplot to create an interactive line chart of the average house value per neighbourhood
# The plot should have a dropdown selector for the neighbourhood
sales_plot = to_avg_sales_neighbourhood.hvplot.line(
             x="year",
             y="average_house_value",
             xlabel='year',
             ylabel='Avr. House Value',
             colormap="viridis",
             title="Neighbourhood",
        groupby = "neighbourhood"
    )
sales_plot
```

Out[51]:

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 [52]:
```
# Fetch the data of all dwelling types per year
dwelling_new = to_data[['neighbourhood',
                        'single_detached_house',
                        'apartment_five_storeys_plus',
                        'movable_dwelling',
                        'semi_detached_house',
                        'row_house', 'duplex',
                        'apartment_five_storeys_less',
                        'other_house']]
dwelling_avg = dwelling_new.groupby([to_data.index, 'neighbourhood']).mean()
dwelling_avg.reset_index(inplace=True)
```

In [53]:
```
# Use hvplot to create an interactive bar chart of the number of dwelling types per neighbourhood
# The plot should have a dropdown selector for the neighbourhood
dwelling_avg.hvplot.bar(stacked=False,
                        x='year',
                        groupby = 'neighbourhood',
                        height=500,
                        rot=90)
```

Out[53]:

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 [54]:
```
# Getting the data from the top 10 expensive neighbourhoods
top_10_most_expensive = to_data.groupby("neighbourhood").mean().sort_values(by='average_house_value', ascending
top_10_most_expensive.reset_index(inplace=True)
top_10_most_expensive
```

Out[54]:

	neighbourhood	single_detached_house	apartment_five_storeys_plus	movable_dwelling	semi_detached_house	row_house	duplex
0	Bridle Path-Sunnybrook-York Mills	2260.00	331.25	0.00	36.25	90.00	25.0
1	Forest Hill South	1742.50	2031.25	1.25	61.25	45.00	75.0
2	Lawrence Park South	3472.50	773.75	0.00	126.25	38.75	225.0
3	Rosedale-Moore Park	2498.75	4641.25	0.00	486.25	245.00	327.5
4	St.Andrew-Windfields	3225.00	1670.00	0.00	185.00	552.50	97.5
5	Casa Loma	916.25	2310.00	0.00	288.75	201.25	162.5
6	Bedford Park-Nortown	4865.00	1981.25	0.00	43.75	57.50	287.5
7	Forest Hill North	1488.75	3392.50	0.00	12.50	16.25	82.5
8	Kingsway South	2326.25	576.25	0.00	66.25	48.75	20.0
9	Yonge-St.Clair	565.00	3948.75	0.00	425.00	212.50	172.5

In [55]:
```
# Plotting the data from the top 10 expensive neighbourhoods
expensive_plot = top_10_most_expensive.hvplot.bar(
        x="neighbourhood",
        y="average_house_value",
        colormap="viridis",
        title="Top 10 Expensive Neighborhoods in TO",
    height=500,
    rot=90,
    xlabel="Neighborhood",
    ylabel="Avg. Sale Price per Square Foot"
).opts(yformatter="%.0f")
expensive_plot
```

Out[55]:

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 [56]:
```
# 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[56]:

	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 [57]:
```
# Calculate the mean values for each neighborhood
all_neighbourhoods = to_data[['neighbourhood', 'average_house_value']].groupby([to_data.index, "neighbourhood"]
all_neighbourhoods.reset_index(inplace=True)
all_neighbourhoods.head()
```

Out[57]:

	year	neighbourhood	average_house_value
0	2001	Agincourt North	200388
1	2001	Agincourt South-Malvern West	203047
2	2001	Alderwood	259998
3	2001	Annex	453850
4	2001	Banbury-Don Mills	371864

In [58]:
```
# Join the average values with the neighbourhood locations
combined_df = pd.merge(df_neighbourhood_locations, all_neighbourhoods, on='neighbourhood')
combined_df.head()
```

Out[58]:

	neighbourhood	lat	lon	year	average_house_value
0	Agincourt North	43.805441	-79.266712	2001	200388
1	Agincourt North	43.805441	-79.266712	2006	293140
2	Agincourt North	43.805441	-79.266712	2011	366667
3	Agincourt North	43.805441	-79.266712	2016	459051
4	Agincourt South-Malvern West	43.788658	-79.265612	2001	203047

Mapbox Visualization

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

In [59]:
```
map_plot = px.scatter_mapbox(
    combined_df,
    lat="lat",
    lon="lon",
    size="average_house_value",
    color="neighbourhood",
    zoom=10,
    title="Average Home Value"
)

map_plot.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 [60]:
```
# YOUR CODE HERE!


plot_1 = px.bar(
    to_avg_sales_neighbourhood,
    x="neighbourhood",
    y="average_house_value",
    title="Top 10 Expensive Neighborhoods in TO",
    facet_row="year",
    width=800,
    height= 900,
    labels={'average_house_value':'Avg. Value'}
    )
plot_1.show()
```