Toronto Dwellings Analysis Dashboard In this notebook, you will compile the visualizations from the previous analysis into functions to create a Panel dashboard. # 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 Bad key savefig.frameon in file /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplot lib/mpl-data/stylelib/ classic_test.mplstyle, line 421 ('savefig.frameon : True') You probably need to get an updated matplotlibrc file from https://github.com/matplotlib/matplotlib/blob/v3.3.4/matplotlibrc.template or from the matplotlib source distribution Bad key verbose.level in file /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotli b/mpl-data/stylelib/ classic test.mplstyle, line 472 ('verbose.level : silent # one of silent, helpful, d ebug, debug-annoying') You probably need to get an updated matplotlibrc file from https://github.com/matplotlib/matplotlib/blob/v3.3.4/matplotlibrc.template or from the matplotlib source distribution Bad key verbose.fileo in file /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotli b/mpl-data/stylelib/_classic_test.mplstyle, line 473 ('verbose.fileo : sys.stdout # a log filename, sys.stdou t or sys.stderr') You probably need to get an updated matplotlibrc file from https://github.com/matplotlib/matplotlib/blob/v3.3.4/matplotlibrc.template or from the matplotlib source distribution In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/ classi c test.mplstyle: The text.latex.preview rcparam was deprecated in Matplotlib 3.3 and will be removed two minor releases later. In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/_classi c test.mplstyle: The mathtext.fallback to cm rcparam was deprecated in Matplotlib 3.3 and will be removed two minor releases lat In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/_classi c test.mplstyle: Support for setting the 'mathtext.fallback to cm' rcParam is deprecated since 3.3 and will be removed two minor releases later; use 'mathtext.fallback : 'cm' instead. In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/_classi c test.mplstyle: The validate_bool_maybe_none function was deprecated in Matplotlib 3.3 and will be removed two minor releases 1 In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/ classi c test.mplstyle: The savefig.jpeg quality rcparam was deprecated in Matplotlib 3.3 and will be removed two minor releases later. In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/ classi c test.mplstyle: The keymap.all axes rcparam was deprecated in Matplotlib 3.3 and will be removed two minor releases later. In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/ classi The animation.avconv path rcparam was deprecated in Matplotlib 3.3 and will be removed two minor releases late In /Users/yandomingos/opt/anaconda3/envs/pyviz/lib/python3.7/site-packages/matplotlib/mpl-data/stylelib/ classi The animation.avconv args rcparam was deprecated in Matplotlib 3.3 and will be removed two minor releases late # Initialize the Panel Extensions (for Plotly) import panel as pn pn.extension("plotly") # Read the Mapbox API key load dotenv() map box api = os.getenv("mapbox") px.set_mapbox_access_token(map_box_api) **Import Data** In [4]: # Import the CSVs to Pandas DataFrames file path = Path("Data/toronto neighbourhoods census data.csv") to_data = pd.read_csv(file_path, index_col="year") file path = Path("Data/toronto neighbourhoods coordinates.csv") df_neighbourhood_locations = pd.read_csv(file_path) **Panel Visualizations** In this section, you will copy the code for each plot type from your analysis notebook and place it into separate functions that Panel can use to create panes for the dashboard. These functions will convert the plot object to a Panel pane. Be sure to include any DataFrame transformation/manipulation code required along with the plotting code. Return a Panel pane object from each function that can be used to build the dashboard. Note: Remove any .show() lines from the code. We want to return the plots instead of showing them. The Panel dashboard will then display the plots. Global available data # Getting the data from the top 10 expensive neighbourhoods # YOUR CODE HERE! 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 # Calculate the mean number of dwelling types units per year # YOUR CODE HERE! 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').mean() num_dwellings_per_year # Calculate the average monthly shelter costs for owned and rented dwellings # YOUR CODE HERE! avg shelter costs = to data[['shelter costs owned', 'shelter costs rented']].groupby('year').mean() avg shelter costs shelter_costs_owned shelter_costs_rented year 846.878571 2001 1085.935714 2006 1316.800000 925.414286 2011 1448.214286 1019.792857 2016 1761.314286 1256.321429 **Panel Visualization Functions** In [14]: # Define Panel visualization functions def neighbourhood map(): """Neighbourhood Map""" to neighbourhood = to data.groupby('neighbourhood').mean() to neighbourhood.reset index(inplace =True) to neighbourhood.head() conc df = pd.concat([df neighbourhood locations, to neighbourhood['average house value']] , axis=1).dropna px.set mapbox access token(map box api) map plot = px.scatter mapbox(conc df , lat='lat' , lon='lon' , size='average house value' , width=1500 , he plotly panel = pn.pane.Plotly(map plot) plotly panel. updates = True return plotly panel def create bar chart(data, title, xlabel, ylabel, color): Create a barplot based in the data argument. fig bar = plt.figure(figsize = (10,5)) data.plot.bar(color=color) plt.title(title) plt.xlabel(xlabel) plt.ylabel(ylabel) plt.close(fig bar) return pn.pane.Matplotlib(fig bar) def create line chart(data, title, xlabel, ylabel, color): Create a line chart based in the data argument. fig line = plt.figure(figsize = (20,5)) data.plot(color=color) plt.title(title) plt.xlabel(xlabel) plt.ylabel(ylabel) plt.close(fig line) return pn.pane.Matplotlib(fig line) def average house value(): """Average house values per year.""" avg house value = to data['average house value'].groupby('year').mean() avg house value.head() fig house = plt.figure(figsize = (20,5)) avg house value.plot.bar(figsize = (20,5)) plt.title('average house value') plt.xlabel('year') plt.ylabel('price') plt.close(fig house) return pn.pane.Matplotlib(fig house) def average value by neighbourhood(): """Average house values by neighbourhood.""" to data new = to data[['neighbourhood' , 'average house value']] fig average = to data new.hvplot.line(x="year", y="average house value", xlabel= "Year", ylabel="Avg. House Value", groupby="neighbourhood", yformatter='%.0f' return fig average def number dwelling types(): """Number of dwelling types per year""" dwelling new = to data.groupby(['year' , 'neighbourhood']).mean() dwelling new.reset index(inplace=True) fig to = dwelling new[['year' , 'neighbourhood', 'single detached house', 'apartment_five_storeys_plus', 'movable dwelling', 'semi detached house', 'row house', 'duplex', 'apartment five storeys less', 'other house']].hvplot.bar(x='year', ylabel= 'average units' , y=['single detached house', 'apartment five storeys plus', 'movable dwelling', 'semi detached house', 'row house', 'duplex', 'apartment five_storeys_less', 'other house'] , width =1000 , height=500 , rot=90, groupby return fig to def average house value snapshot(): """Average house value for all Toronto's neighbourhoods per year.""" average house to = px.bar(to data , x='neighbourhood' , y='average house value' , facet row=to data.index return average house to def top most expensive neighbourhoods(): """Top 10 most expensive neighbourhoods.""" top 10 most expensive = to data.groupby("neighbourhood").mean().sort values(by='average house value', ascer top 10 most expensive.reset index(inplace=True) top 10 most expensive 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") return expensive plot Panel Dashboard In this section, you will combine all of the plots into a single dashboard view using Panel. Be creative with your dashboard design! # Create a Title for the Dashboard title = pn.pane.Markdown(# Real Estate Study of Toronto width=800, welcome = pn.pane.Markdown(This dashboard contains visual data of the real estate market in Toronto for years 2001, 2006, 2011 and 2016. Each tab contains different information on the house market and neighbourhood date. You will be able to interact) tabs = pn.Tabs(("Welcome", pn.Column(welcome, neighbourhood map())), ("Yearly Market Analysis", pn.Column(create_bar_chart(num_dwellings_per_year.loc[2001], "Number of Differer create_bar_chart(num_dwellings_per_year.loc[2006], "Number of Different Dv create_bar_chart(num_dwellings_per_year.loc[2011], "Number of Different create_bar_chart(num_dwellings_per_year.loc[2016], "Number of Different ("Shelter Cost vs House Value", pn.Column(create_line_chart(avg_shelter_costs['shelter_costs_owned'], "Avg create_line_chart(avg_shelter_costs['shelter_costs_rented'], "Avg average house value)), ("Neighbourhood Analysis", pn.Column(average_house_value(), number_dwelling_types(), average_house_value_snapshot())), ("Top Expensive Neighbourhoods", pn.Column(top_most_expensive_neighbourhoods())) dashboard = pn.Column(pn.Row(title), tabs, width=900) Serve the Panel Dashboard dashboard.servable() Real Estate Study of Toronto Welcome Yearly Market Analysis Shelter Cost vs House Value Neighbourhood Analysis Top Expensive Neighbourhoods This dashboard contains visual data of the real estate market in Toronto for years 2001, 2006, 2011 and 2016. Each tab contains different information or Debugging Note: Some of the Plotly express plots may not render in the notebook through the panel functions. However, you can test each plot by uncommenting the following code neighbourhood map() # create bar chart(data, title, xlabel, ylabel, color) # # 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', # create line chart(data, title, xlabel, ylabel, color) # # 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 average house value() average_value_by_neighbourhood() number dwelling types() average_house_value_snapshot() top most expensive neighbourhoods()

sunburts cost analysis()