taipei_shop_rent_cost

October 6, 2020

1 Taipei Shop Rent Cost

The cost aspect would be one of the things that could be good to consider. In this list, we would use the data we scrape from 591.com (one of the biggest property listing platfrom in Taiwan).

[1]:

```
# initial setup, import packages, path, and config
from typing import Any
import json
import os

import pandas as pd
import geopandas as gpd
import plotly.express as px
from shapely.geometry import MultiPoint
pd.options.mode.chained_assignment = None  # not show dataframe copy slice␣
 ↪warning

from lib import shared_lib
from shared_lib import data_processor
from data_processor.lib.geolib_helper import get_shp_filepath,␣
 ↪load_normalize_gov_shp_data

from lib.plotly_helper import add_chart_title, add_chart_annotation

# setup path
ANALYSIS_NAME = 'taipei_shop_rent_cost'

CURRENT_DIR = os.path.dirname(os.path.abspath('__file__'))
BASE_DIR = os.path.dirname(CURRENT_DIR)
ANALYSIS_DIR = os.path.join(BASE_DIR, 'analysis', ANALYSIS_NAME)

plotly_default_config_chart = dict(
    displayModeBar=True,
    responsive=False,
    modeBarButtonsToRemove=['zoomIn2d', 'zoomOut2d', 'select2d', 'lasso2d',␣
 ↪'toggleSpikelines'],
    displaylogo=False
)
```

```
plotly_default_config_geo = dict(
    displayModeBar=True,
    responsive=False,
    scrollZoom=False,
    modeBarButtonsToRemove=['select2d', 'lasso2d'])
```

1.1 Get the average rent price per village

We would try to know how expensive each area is. To make things equal, would use the rent cost of 1st floor, store front, and get the price per area (ping).

[2]:

```
# setup output filepath
data_dir = os.path.join(BASE_DIR, 'data')
data_mart_dir = os.path.join(data_dir, 'aggregated-data_mart')

save_taipei_shop_rent_cost_filepath = os.path.join(data_mart_dir,␣
 ↪ANALYSIS_NAME+'.csv')

# setup data source
data_warehouse_dir = os.path.join(data_dir, 'normalized-data_warehouse')

# - taipei shop rent price
taipei_shop_rent_price_filepath = os.path.join(data_warehouse_dir,␣
 ↪'taipei_shop_rent_price.csv')
taipei_shop_rent_price_df = pd.read_csv(taipei_shop_rent_price_filepath)

# - area dimension table
area_dimension_table = pd.read_csv('../data/normalized-data_warehouse/
 ↪area_dimension_table.csv')
area_dimension_table = area_dimension_table.astype({'village_code':str})
area_dimension_table.set_index('village_code', inplace=True)

# - taipei area data, village detail
village_shp_path = get_shp_filepath(os.path.join(BASE_DIR, 'data',␣
 ↪'taiwan_twd97_map_data_village'))
village_gpd = load_normalize_gov_shp_data(village_shp_path)

taipei_village_gpd = village_gpd[village_gpd['county_chinese_name'] == '   ']
taipei_village_gpd.set_index('village_code', drop=False, inplace=True)

taipei_village_gpd = pd.merge(
    taipei_village_gpd, area_dimension_table[['township_english_name']],
    left_index=True, right_index=True
)
```

1.1.1 Calculate the data

Calulate the average rent price and average them based on location, on village detail level.

[3]:

```
# Calculate Taipei average shop rent (store front, first floor) price per month
taipei_shop_rent_price_df = taipei_shop_rent_price_df\
    [taipei_shop_rent_price_df['floorInfo'].apply(lambda x: '1/' in str(x))]

taipei_shop_rent_price_df['village_code'] = \
    taipei_shop_rent_price_df['village_code'].apply(lambda x: str(x).split('.
 ↪')[0])

taipei_shop_rent_price_df['price_per_ping'] = \
    taipei_shop_rent_price_df['price'] / taipei_shop_rent_price_df['area']

village_area_rent_price_average = taipei_shop_rent_price_df\
    .groupby(['village_code'])['price_per_ping'].mean().to_dict()

def dict_helper(lookup_dict: dict, key: Any) -> Any:
    if key in lookup_dict:
        return lookup_dict.get(key)
    else:
        return 0

taipei_village_gpd['store_price_per_ping_average'] = \
    taipei_village_gpd['village_code'].apply(lambda x:␣
 ↪dict_helper(village_area_rent_price_average, x))
```

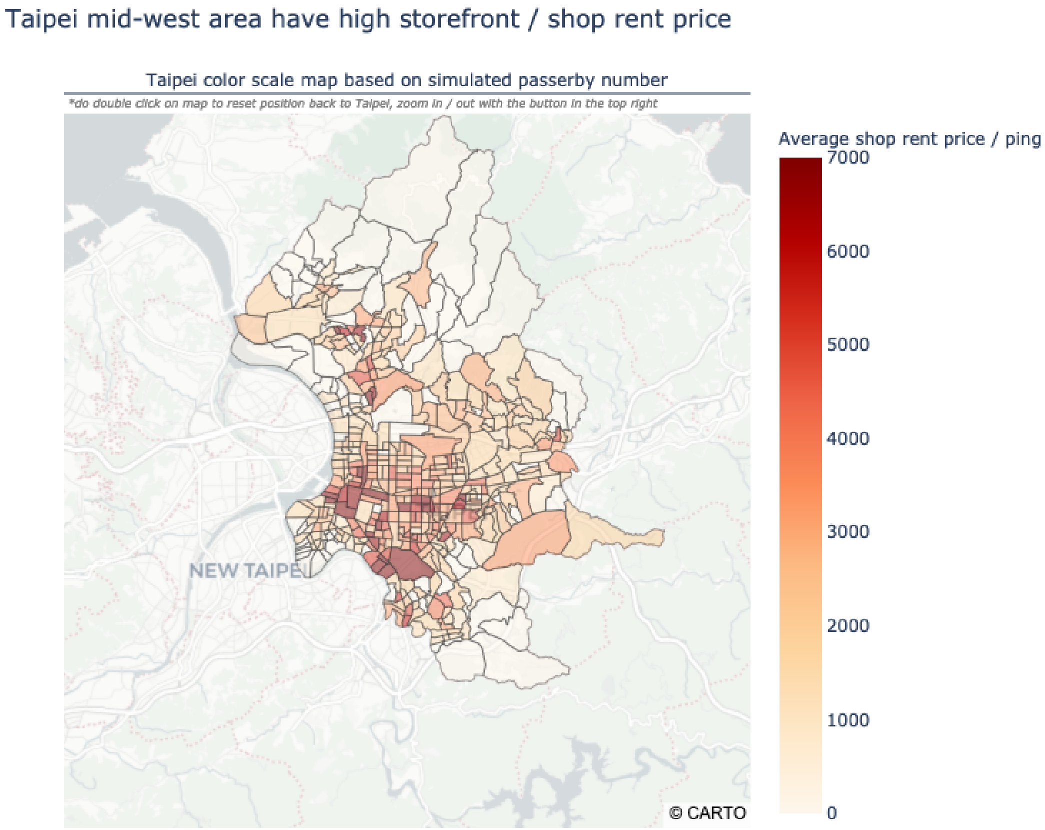
1.1.2 Save and visualize the data

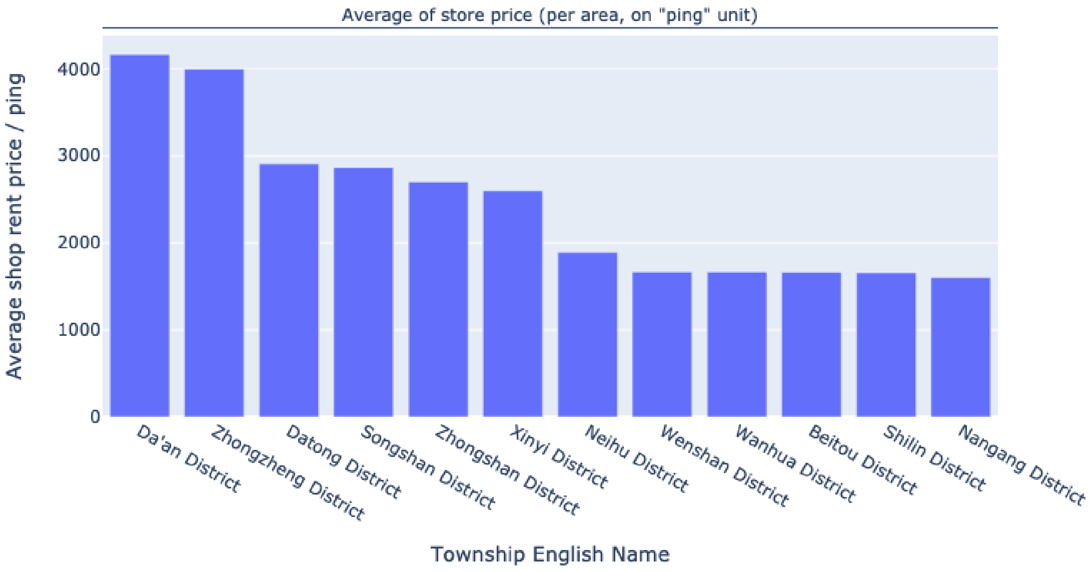
Would save and visualize how the data is

[4]:

```
# prepare the geojson data
taipei_village_geojson = json.loads(taipei_village_gpd.geometry.to_json())
center_point = MultiPoint(taipei_village_gpd['geometry'].apply(lambda x: x.
 ↪centroid)).centroid

save_df = taipei_village_gpd.loc[:, taipei_village_gpd.columns != 'geometry']
save_df.to_csv(save_taipei_shop_rent_cost_filepath, index=False)

taipei_township_shop_price_agg = taipei_village_gpd.groupby(['township_code',␣
 ↪'township_english_name'])['store_price_per_ping_average'].mean().
 ↪reset_index()

taipei_township_shop_price_agg.sort_values('store_price_per_ping_average',␣
 ↪ascending=False, inplace=True)
```

```
[5]: # draw first chart the map
fig = px.choropleth_mapbox(taipei_village_gpd, geojson=taipei_village_geojson,
                           locations='village_code',
                           color='store_price_per_ping_average',
                           hover_name='village_english_name',
                           hover_data=['township_english_name'],
                           labels={'township_english_name': 'Township English␣
 ↪Name',
                                   'store_price_per_ping_average' : 'Average␣
 ↪shop rent price / ping'},
                           color_continuous_scale='OrRd',
                           range_color=(0,7000),
                           opacity=0.5,
                           mapbox_style='carto-positron',
                           center={'lon':center_point.x, 'lat':center_point.y},
                           zoom=10)

fig.update_traces(hovertemplate=fig['data'][-1]['hovertemplate']\
                  .replace('village_code=%{location}<br>','')\
                  .replace('=',' = ')\
                  .replace('{z}','{z:,.2r}')
                 )

add_chart_title(fig, "Taipei color scale map based on simulated passerby␣
 ↪number", 1.2)

add_chart_annotation(fig,
                     '<i>*do double click on map to reset position back to␣
 ↪Taipei, '
                     'zoom in / out with the button in the top right</i>')

fig.update_layout(
    title='Taipei mid-west area have high storefront / shop rent price',
    margin={'t':120},
    height=700
)

fig.show(config=plotly_default_config_geo)
fig.write_image(os.path.join(ANALYSIS_DIR, 'taipei_shop_rent_price-1.png'))

# draw second chart, bar chart of average
fig = px.bar(taipei_township_shop_price_agg,
             x='township_english_name',
             y='store_price_per_ping_average',
             labels={'township_english_name': 'Township English Name',
                 'store_price_per_ping_average': 'Average shop rent price /␣
 ↪ping'},
            )

fig.update_layout(showlegend=False)
fig.update_xaxes(fixedrange=True)
fig.update_yaxes(fixedrange=True)

add_chart_title(fig, 'Average of store price (per area, on "ping" unit)')

fig.show(config=plotly_default_config_chart)
fig.write_image(os.path.join(ANALYSIS_DIR, 'taipei_shop_rent_price-2.png'))
```


Taipei mid-west area have high storefront / shop rent price
Taipei color scale map based on simulated passerby number
*do double click on map to reset position back to Taipei, zoom in / out with the button in the top right
NEW TAIPEI
© CARTO
Average shop rent price / ping
7000
6000
5000
4000
3000
2000
1000
0


Average of store price (per area, on "ping" unit)
Average shop rent price / ping
0
1000
2000
3000
4000
Da'an District
Zhongzheng District
Datong District
Songshan District
Zhongshan District
Xinyi District
Neihu District
Wenshan District
Wanhua District
Beitou District
Shilin District
Nangang District
Township English Name