A Night To Remember

CINEMA VS STREAMING

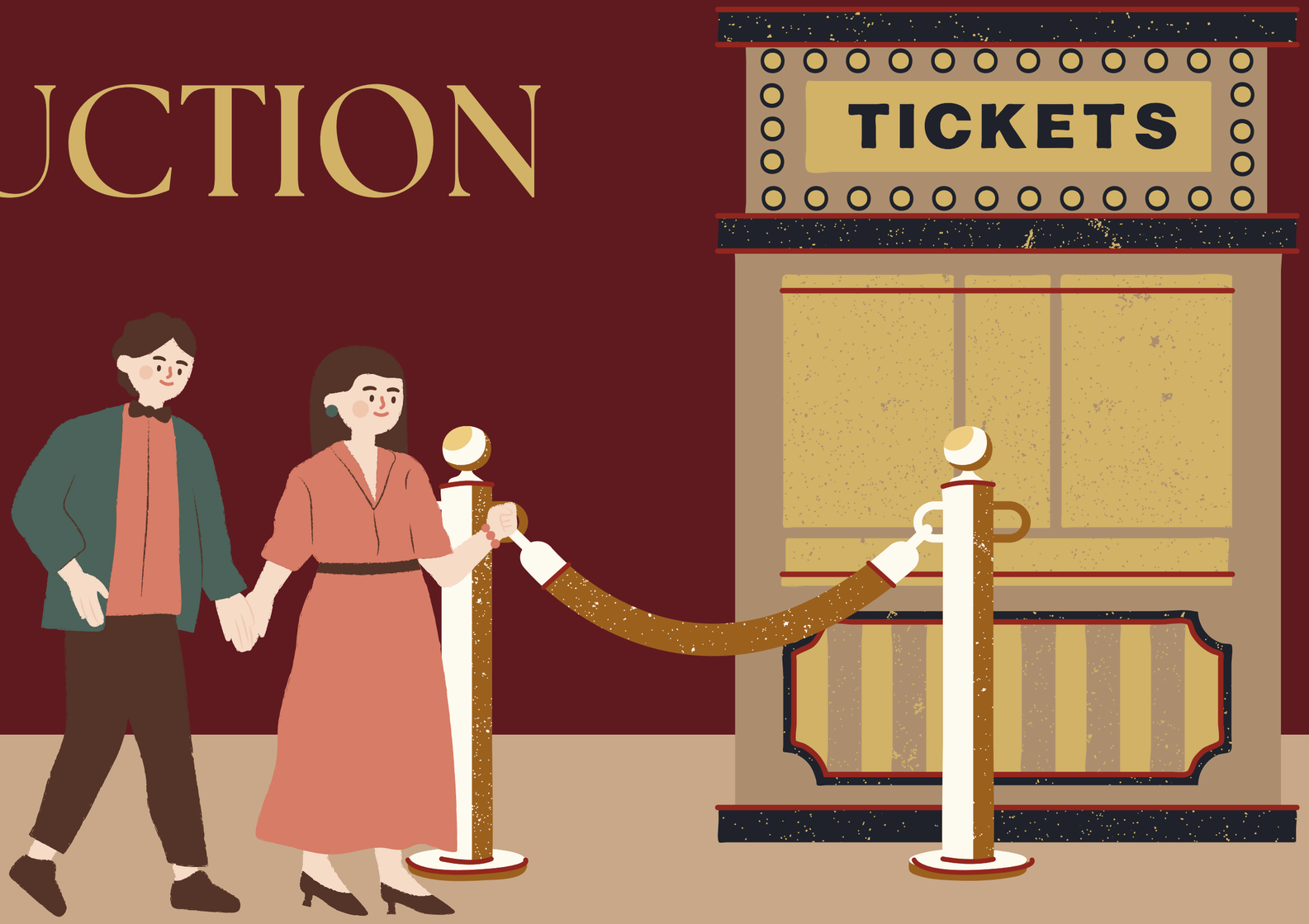
Are good old cinemas still relevant today?

Presented by: Aline Hornoff, Camille Evangelista, Edbert Widjaja & Carmen Sin

TOPIC OUTLINE

- Introduction
- Cinema Location on Map (Leaflet)
- Number of Exhibitors, screens and seats through the years
- Price comparison between cinema ticket price and streaming fee
- Number of Box office movie production
- Movie goers' attendance
- Netflix subscribers numbers
- Conclusion

INTRODUCTION



INTRODUCTION

- Back in the day if people wanted to watch the latest movie everyone else was talking about, they go to the cinema with their family or friends. In case they did not make it on the movie schedule, they had to wait for it to be available in their nearest video rental shop.
- Nowadays due to technological advancement we have the option to use streaming services in the comfort of our own homes to watch movies. This option gained more popularity when Covid happened as most people chose to stay home and also restrictions were put in place on public spaces. This made us ask the question, are cinemas still relevant?

- Are cinemas still have the demand?
  We intend to answer the question viewing from different angles.
- Are we seeing the contraction of cinemas' footprint?
- How often movie goer pay and see movies in the cinema over time?
- Streaming platforms – how competitive are different streaming offerings against cinema ticket price? (excluding free for view platforms)
- In the case of giant of streaming provider, Netflix. How many subscribers over the years?
- Number of BoxOffice production over the years?

METHODS ADOPTED FOR THIS PROJECT

- (1) Web scraping data for all Australian cinema location, clean up data using ETL and using leaflet to map out all their location with markers.

- (2) Read in data and use d3 to produce interactive line chart to display the changes in theater, screen and seating capacity over time

- (3) Web scraping streaming pricing comparison webpage to get data, perform ETL to data and use d3 to map out scatterplot (relationship chart)

- (4)Web scraping webpage to get box office production number/ import csv file for Netflix subscriber numbers and cinema attendance number to make plotly chart

METHODS ADOPTED FOR THIS PROJECT

- Python Flask- powered API, HTML/ CSS, JS and MongoDB for the database

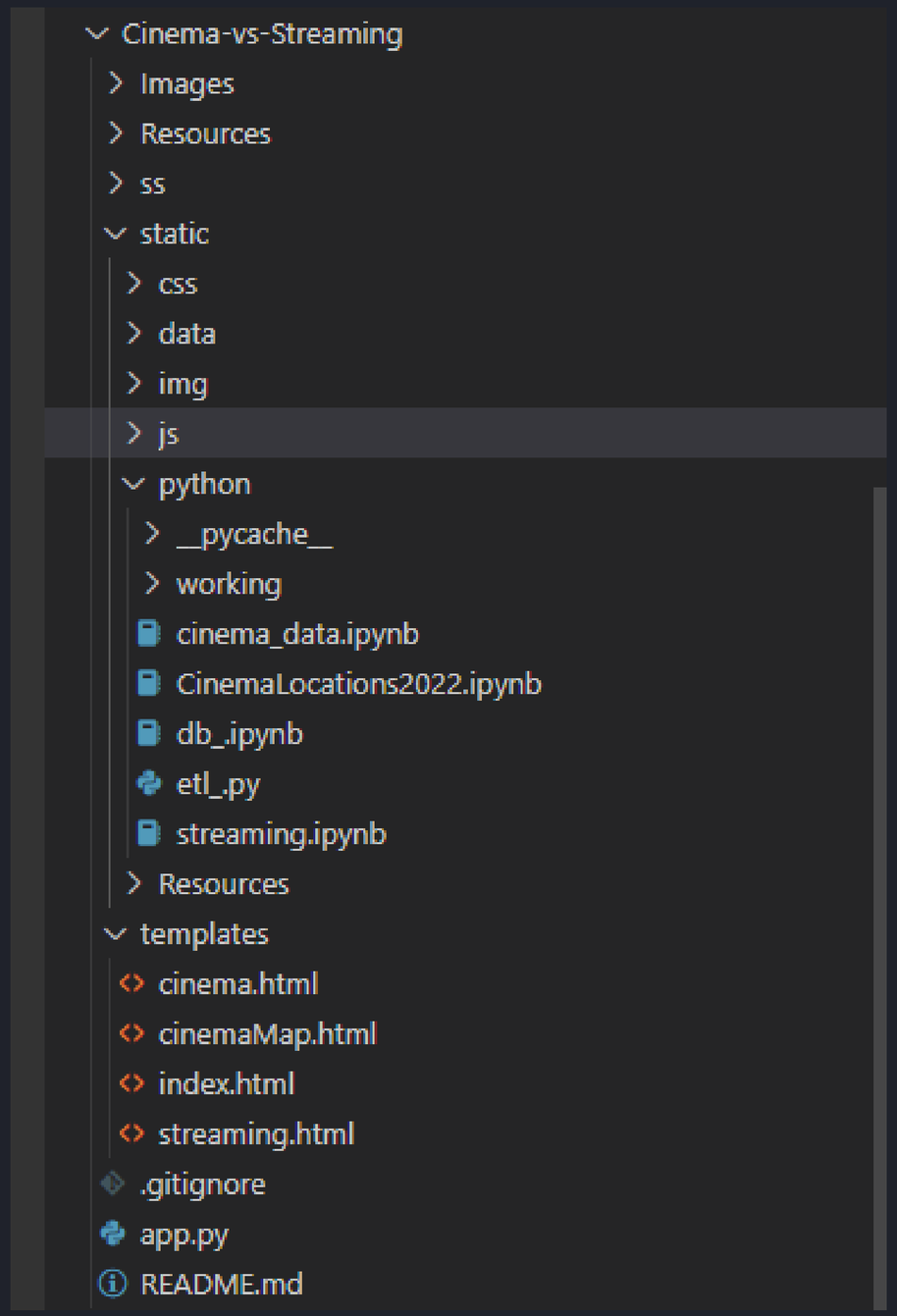
- JS library - Cloudflare - Animate.css

https://cdnjs.cloudflare.com/ajax/libs/animejs/2.0.2/anime.min.js

- Interactive views

Control layer for the map
User select text to view line chart
User interact with with scatter plot chart to display info
User interact with the bar chart to display info

Folder Struture



Database

MongoDB

db_ipynb

Cinema-vs-Streaming > static > python > db_ipynb > # Declare the collection & Insert

+ Code + Markdown | Run All Clear Outputs of All Cells Restart Interrupt | Variables Outline ...

```
# The default port used by MongoDB is 27017
# https://docs.mongodb.com/manual/reference/default-mongodb-port/
conn = 'mongodb://localhost:27017'
client = pymongo.MongoClient(conn)

# Declare the database
db = client.Cace_db
```

[2]

```
#Grab dataframes from our resources
df_boxoffice = etl_.scrape_boxoffice()
df_cinemadata = etl_.df_Cinemadata()
df_nextflixsubs = etl_.df_Nextlixsubs()
df_streamingfee = etl_.df_Streamingfee()
df_cinema1948 = etl_.df_Cinema1948()
df_cinema2022 = etl_.df_Cinema2022()
```

[3]

```
# Clearn up netflixsubs dataframe
df_boxoffice = pd.merge(df_boxoffice,df_nextflixsubs, on="Year")
```

[4]

```
# Declare the collection & Insert
db.boxoffice.insert_many(df_boxoffice.to_dict('records'))
db.cinemadata.insert_many(df_cinemadata.to_dict('records'))
db.nextflixsubs.insert_many(df_nextflixsubs.to_dict('records'))
db.streamingfee.insert_many(df_streamingfee.to_dict('records'))
db.cinema1948.insert_many(df_cinema1948.to_dict('records'))
db.cinema2022.insert_many(df_cinema2022.to_dict('records'))
```

[5]

```
<pymongo.results.InsertManyResult at 0x2368d5d2670>
```

MongoDB Compass - localhost:27017/Cace_db.boxoffice

Connect View Collection Help

localhost:27017

4 DBS 9 COLLECTIONS

FAVORITE

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 5.0.8 Community

My Queries

Databases

Filter your data

Cace_db

boxoffice

cinema1948

cinema2022

cinemadata

nextflixsubs

streamingfee

Documents
Cace_db.boxoffice

Cace_db.boxoffice

Documents Aggregations Schema Explain Plan

FILTER { field: 'value' }

ADD DATA VIEW

```
_id: ObjectId('62dcb886af0a5a2d2d37f837')
Year: 2022
Total_gross: "$4,540,382,909"
LY: "-"
Releases: 274
Average: "$16,570,740"
No1_release: "Top Gun: Maverick"
Total_Subscriber: 221744000
AsiaPacific_Percentages: 0.15
AsiaPacific_Subscribers: 33261600
Cinema_attendance_perc: 0.45
cinema_freq: 3
```

```
_id: ObjectId('62dcb886af0a5a2d2d37f838')
Year: 2021
Total_gross: "$4,488,914,150"
LY: "+113.4%"
Releases: 438
```

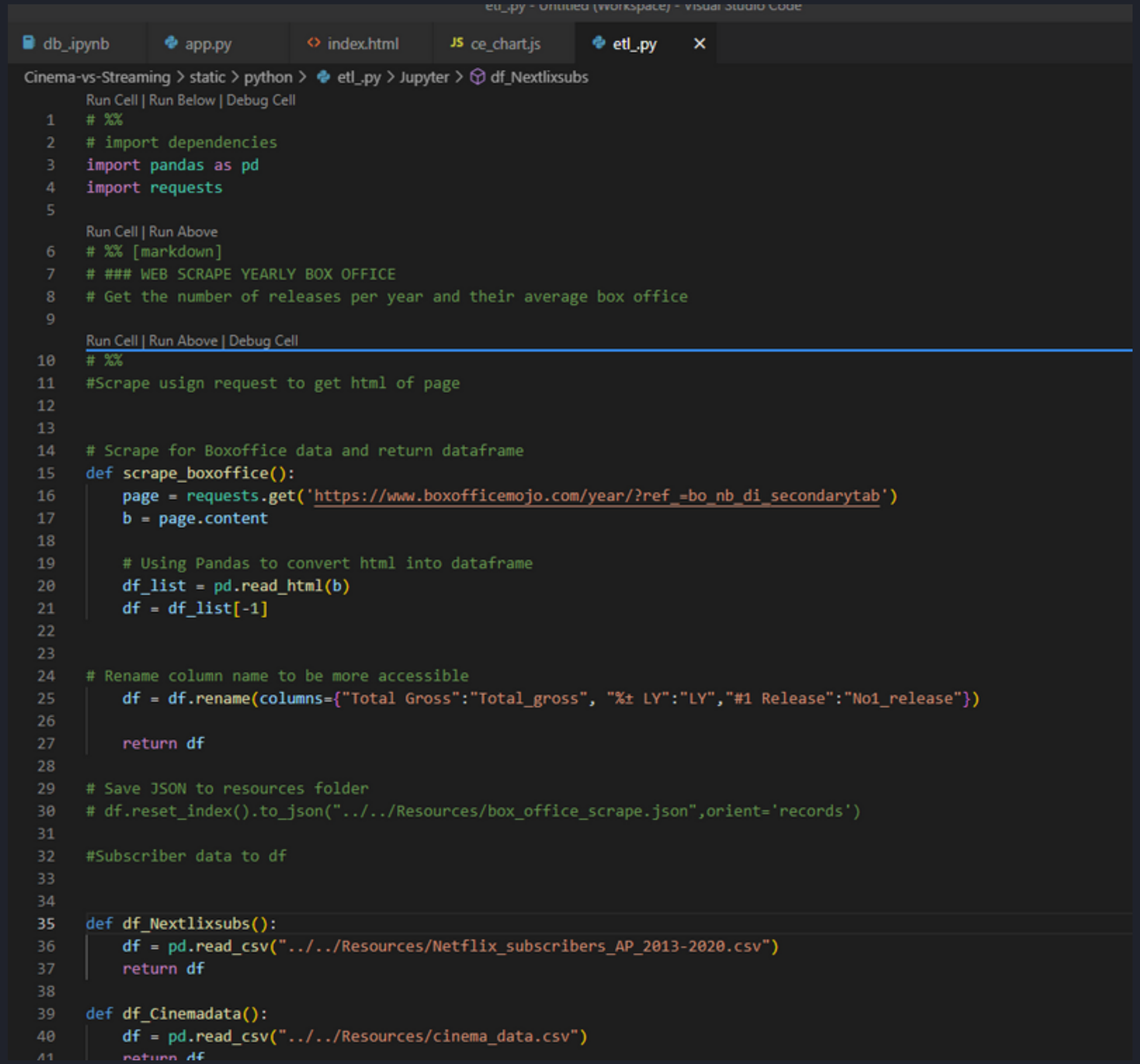
Flask

db_.ipynb | app.py

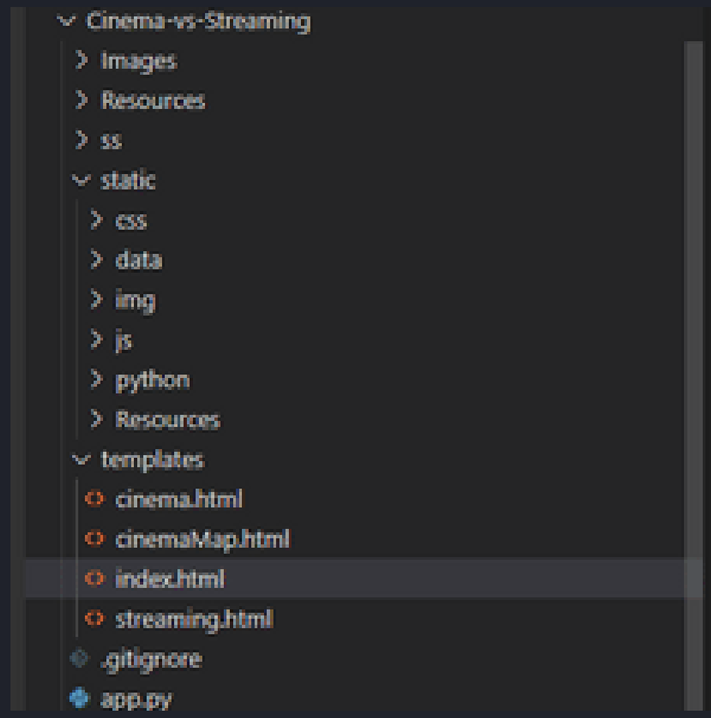
Cinema-vs-Streaming > app.py > ...

```python
from flask import Flask, render_template, redirect
from flask_pymongo import PyMongo
import pymongo
from flask import Response
from bson import json_util
from flask import jsonify
import json
from bson.json_util import ObjectId



# Create an instance of Flask
app = Flask(__name__)

# Use PyMongo to establish Mongo connection
mongo = PyMongo(app, uri="mongodb://localhost:27017/cace_db")

conn = 'mongodb://localhost:27017'
client = pymongo.MongoClient(conn)


# Route to render index.html template using data from Mongo
@app.route("/")
def home():

    # Find one record of data from the mongo database
    _data = mongo.db.collection.find_one()
    a = "static/Resources/box_office_scrape.json"


    # Return template and data
    return render_template("index.html", mars=_data, titles = [0])

@app.route("/streaming")
def streaming():
    # Find one record of data from the mongo database
    _data = mongo.db.collection.find_one()
    a = "static/Resources/box_office_scrape.json"

    return render_template("streaming.html")
```

ETL

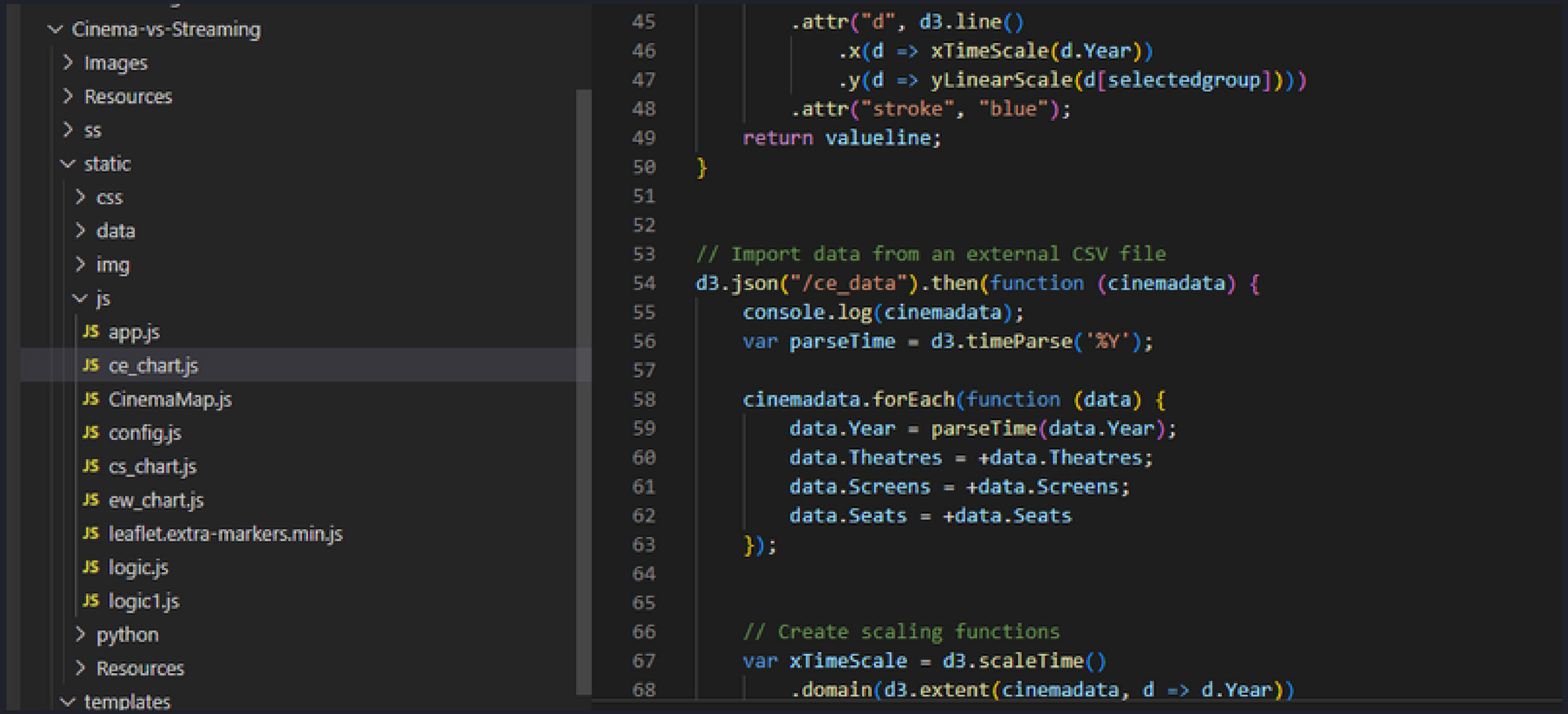

etl_.py - Untitled (Workspace) - Visual Studio Code
db_ipynb
app.py
index.html
ce_chart.js
etl_.py
Cinema-vs-Streaming > static > python > etl_.py > Jupyter > df_Nextlixsubs
Run Cell | Run Below | Debug Cell
# %%
# import dependencies
import pandas as pd
import requests
Run Cell | Run Above
%% [markdown]
WEB SCRAPE YEARLY BOX OFFICE
Get the number of releases per year and their average box office
Run Cell | Run Above | Debug Cell
%%
#Scrape usign request to get html of page
# Scrape for Boxoffice data and return dataframe
def scrape_boxoffice():
page = requests.get('https://www.boxofficemojo.com/year/?ref_=bo_nb_di_secondarytab')
b = page.content
# Using Pandas to convert html into dataframe
df_list = pd.read_html(b)
df = df_list[-1]
# Rename column name to be more accessible
df = df.rename(columns={"Total Gross":"Total_gross", "%± LY":"LY","#1 Release":"No1_release"})
return df
Save JSON to resources folder
df.reset_index().to_json("../../Resources/box_office_scrape.json",orient='records')
#Subscriber data to df
def df_Nextlixsubs():
df = pd.read_csv("../../Resources/Netflix_subscribers_AP_2013-2020.csv")
return df
def df_Cinemadata():
df = pd.read_csv("../../Resources/cinema_data.csv")

Jinja templates




```
                    up the same level as pre-Covid.</p>
                </div>
            </div>
        </div>


        <script src="https://cdnjs.cloudflare.com/ajax/libs/animejs/2.0.2/anime.min.js"></script>

        <script src="https://d3js.org/d3.v5.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/d3-tip/0.7.1/d3-tip.min.js"></script>

        <script src="https://cdnjs.cloudflare.com/ajax/libs/d3/5.5.0/d3.js"></script>
        <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

        <script src="{{ url_for('static', filename='js/app.js') }}"></script>


    </body>
```

js linking to the database



CSS file - styling

style.css

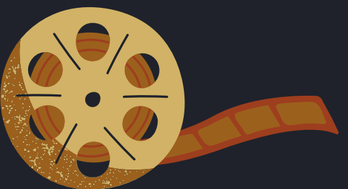
Cinema-vs-Streaming > static > css > # style.css > .axisText

```css
body{
  background-color: black;
}

.container{
  background-color: white;
  margin-top: 50px;
}

p {
  color: rgb(78, 93, 119);
  font-family: sans-serif;
  font-size: 20px;
  text-align: justify;
  margin-left: 30px;
  }
.btn-secondary{
  background-color: #a0353a;
  color: white;
}
.card-title{
  color: rgb(78, 93, 119);
  font-family: sans-serif;
  font-size: 20px;
  font-weight: bold;
  text-align: justify;
}
.card-text{
  color: rgb(78, 93, 119);
  font-family: sans-serif;
  font-size: 20px;
  text-align: justify;
}
```

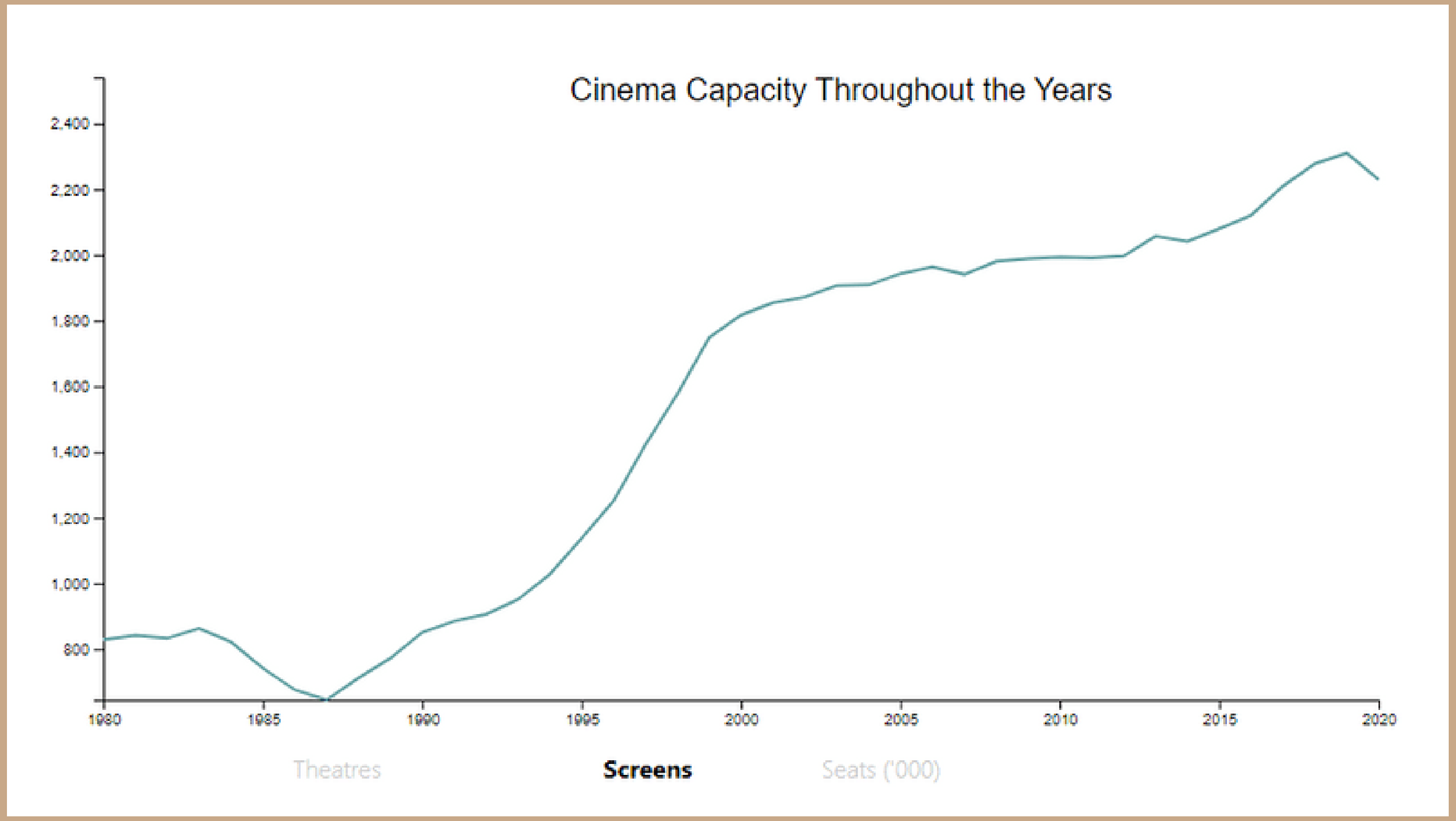
CINEMA FOOTING OVER TIMES (LEAFLET)


Denpasar
Timor Lest
Kupang
Timor Sea
Arafura Sea
Port Moresby
Darwin
Cairns
Townsville
Mackay
NORTHERN TERRITORY
Mount Isa
QUEENSLAND
Alice Springs
WESTERN AUSTRALIA
Australia
Brisbane
SOUTH AUSTRALIA
Kalgoorlie
NEW SOUTH WALES
Bunbury
Mildura
Newcastle
Sydney
A.C.T.
J.B.T.
Great Australian Bight
Melbourne

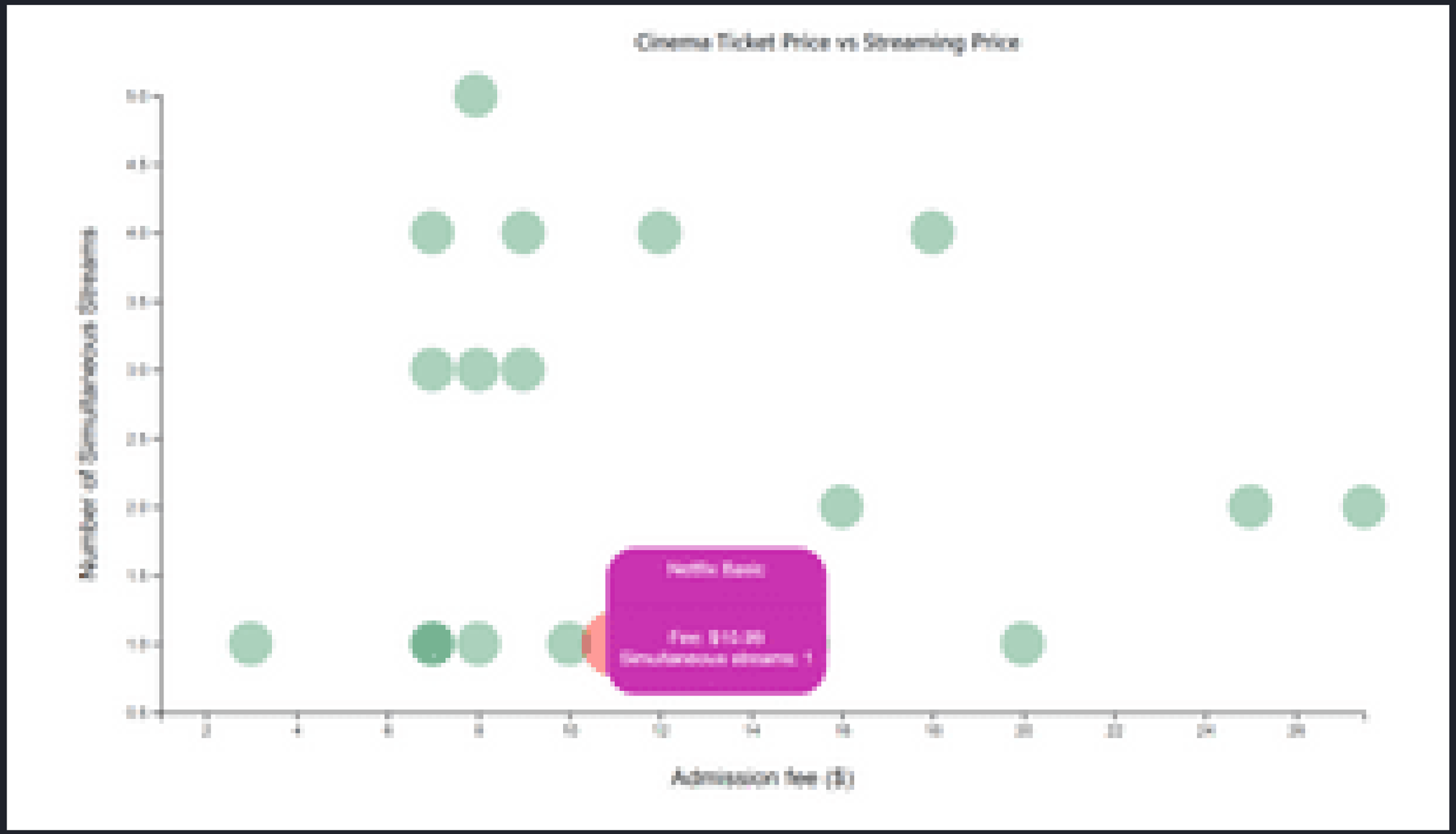
GEO MAPPING

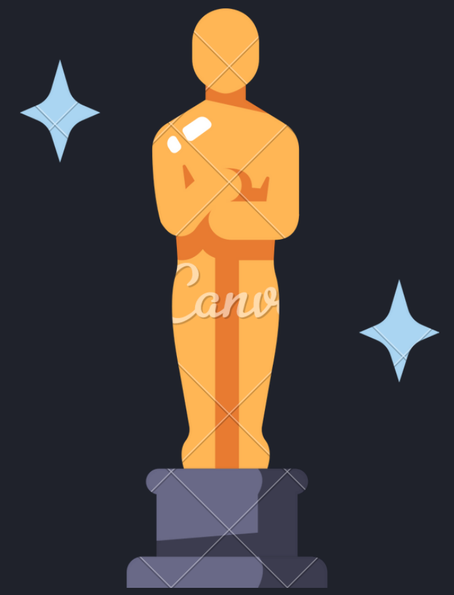


CINEMA SCREENS / SEATING CAPACITY

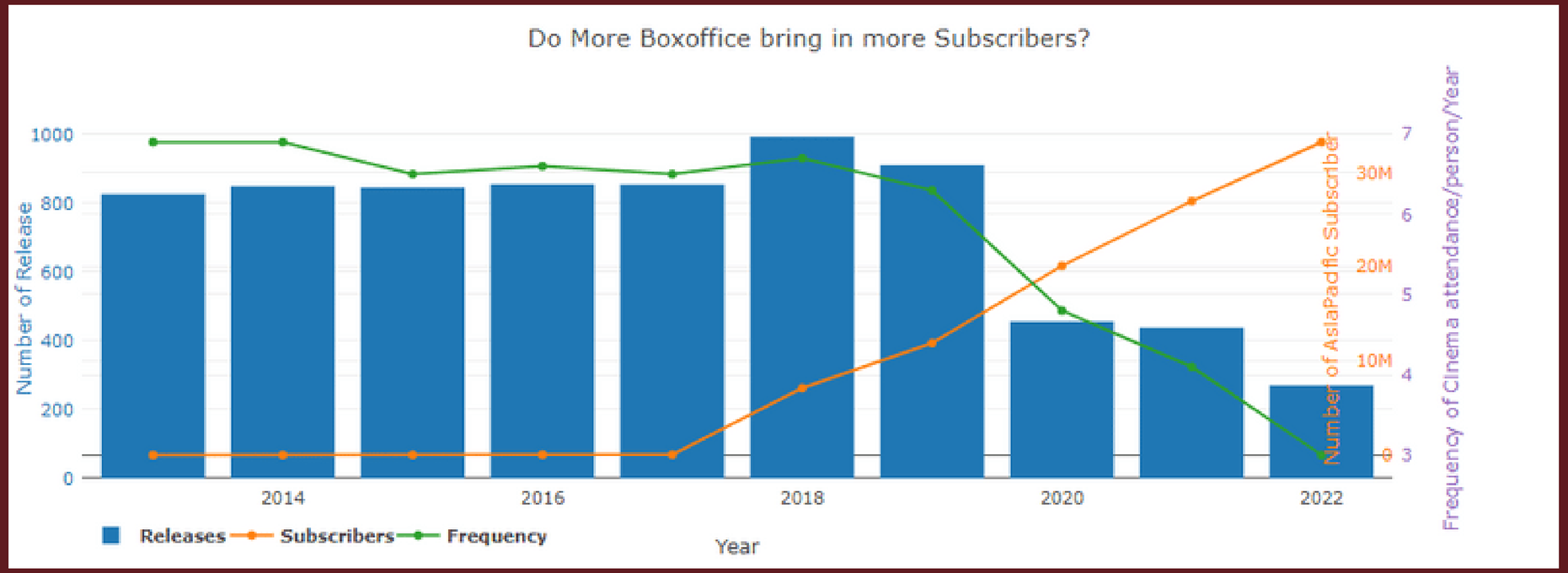


CINEMA VS STREAMING IN PRICING





BOX OFFICE RELEASE / SUBSCRIBERS NUMBERS CINEMA ATTENDENCE FREQ



CINEMA

- Number of movie-goers remain relatively similar even the streaming numbers are growing in 2013-2019
- Data shows that movie-goers continue to see movies in cinema, the frequency was less due to COVID related restriction
- **Cinemas remain relevant**

STREAMING PROVIDER

- Market share will be fragmented as more new providers available
- choices will drive the streaming fee downward
- More free and on-demand streaming provide alternative entertainment

Challenges:

If we have more time:
Finalized the map with layers and markers

If we have more time

***Finalized the map with layers and markers**
***Collect demographic preference data to tell more complete story**
***To expand the map and add in the data for 1948 to 1971...**

THANK YOU FOR LISTENING

Don't ask any questions!