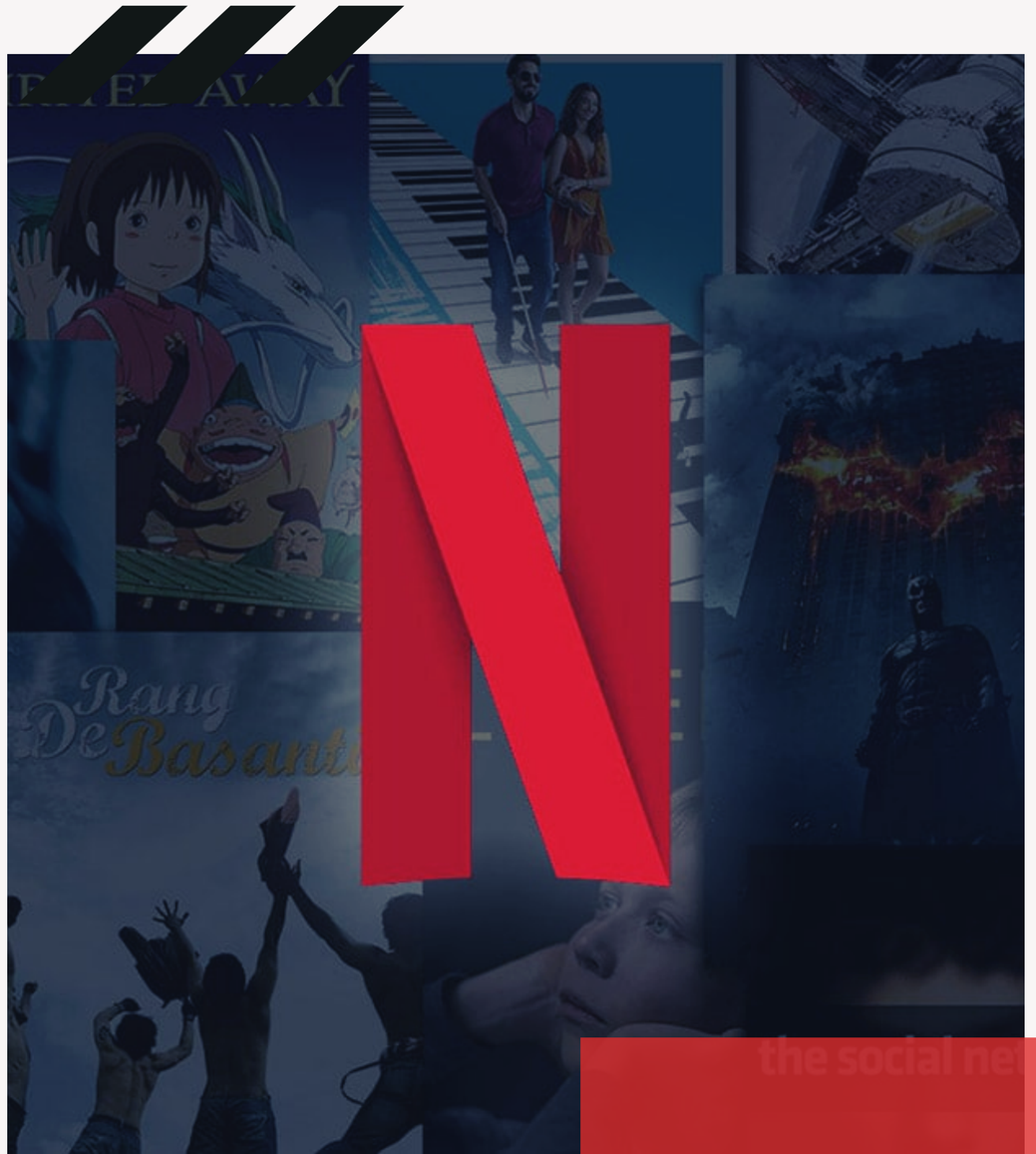
Netflix & Movies ETL

Presented by ***The Conservatory*** *-*
Leonard Paul Kamara, Dilia Yunusova,
Jessica Kwon, Sapiesi Tupou, Wynne Zhang



What it's about...

The purpose of the project is to extract movie data from various sources, transform them into data frames and load in PostgreSQL. Does high budget mean greater revenue?

Step 1

Extract

Data Sets and Sources
Kaggle - Netflix Movie dataset
API call from The Movie Database (TMDb)
The Numbers Website

Step 2

Transform

Merge Netflix dataset with TMDb data
Clean datasets and investigate null values

Step 3

Load

Create connection in pandas
Loaded into Postgres
Query tables

Extract

Our Sources

The Numbers: run by Nash Information Services, premier provider of movie industry data.

Kaggle - Netflix Movies and TV Shows: Dataset consisting of TV shows and movies available on Netflix as of 2019. Arranged by Shivam Bansal and data collected from Flixable, a third-party Netflix search engine.

The Movie Database (TMDb) API: A community built movie and TV database with extensive breadth of data and strong international focus.

Web Scraping & API

```
# The budget
# Extract first 6000 movies from The Numbers website
# Use splinter and beautiful soup to extract data from the first 60 pages
# Save data csv
```

```
# Range of first 60 pages to loop through
urls = np.arange(1, 6001, 100)
```

```
# create empty list to hold the tables
movies = []
for url in urls:

    print('Extracting data')
    print('.................')

    # pages to loop through
    page = 'https://www.the-numbers.com/movie/budgets/all' + '/' + str(url)

    # Browser instance t visit pages
    browser.visit(page)

    # Extract htlp of pages
    html = browser.html

    # Create a Soup instance
    soup = BeautifulSoup(html, 'html.parser')

    # Extract the tables and allow minimum 2 seconds wait time to sleep
    data = sou
    sleep(rand:

    # Create a
    film = pd.

    # Append t
    movies.app
```

```
# Inspect movies_num_df
movies_num_df.head()
```

	ReleaseDate	Movie	ProductionBudget	DomesticGross	WorldwideGross
0	Apr 23, 2019	Avengers: Endgame	$400,000,000	$858,373,000	$2,797,800,564
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$379,000,000	$241,063,875	$1,045,663,875
2	Apr 22, 2015	Avengers: Age of Ultron	$365,000,000	$459,005,868	$1,396,099,202
3	Dec 16, 2015	Star Wars Ep. VII: The Force Awakens	$306,000,000	$936,662,225	$2,065,478,084
4	Apr 25, 2018	Avengers: Infinity War	$300,000,000	$678,815,482	$2,044,540,523

```
# Export movies_num_df as csv movies_num_final.csv
movies_num_df.to_csv('movies_num_final.csv', index = False)
```

Using Splinter & BeautifulSoup to scrape table data from The Numbers website.

We utilized TMDb's API to extract more extensive information about each movie - beginning with minimum information such as movie title, and retrieving information such as movie budget, revenue, runtime, and popularity.

```
1 | Extracting Revenue Data for Norm of the North: King Sized Adventure
.....................
2 | Extracting Revenue Data for Jandino: Whatever it Takes
.....................
3 | Extracting Revenue Data for #realityhigh
.....................
#realityhigh not found.....Skipping!
4 | Extracting Revenue Data for Automata
.....................
5 | Extracting Revenue Data for Fabrizio Copano: Solo pienso en mi
.....................
6 | Extracting Revenue Data for Good People
.....................
7 | Extracting Revenue Data for Joaquín Reyes: Una y no más
.....................
8 | Extracting Revenue Data for Kidnapping Mr. Heineken
.....................
9 | Extracting Revenue Data for Krish Trish and Baltiboy
.....................
Krish Trish and Baltiboy not found.....Skipping!
10 | Extracting Revenue Data for Krish Trish and Baltiboy: Battle of Wits
.....................
Krish Trish and Baltiboy: Battle of Wits not found.....Skipping!
11 | Extracting Revenue Data for Krish Trish and Baltiboy: Best Friends Forever
.....................
Krish Trish and Baltiboy: Best Friends Forever not found.....Skipping!
12 | Extracting Revenue Data for Krish Trish and Baltiboy: Comics of India
.....................
Krish Trish and Baltiboy: Comics of India not found.....Skipping!
```

Transform

Inspect merged data-frames
Check for null values
Resolved null values via:
- API for countries
- Google search for Ratings
- Google search for average run time
- Dropped duplicate rows with duplicate values in ID column
- Convert Date column to datetime object
- Clean Budget Data Frame by stripping '$' and commas (,)
- Save the cleaned dataframes and export as csv

```
# Inspect values and data types
netflix_revenue_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3217 entries, 0 to 3216
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   title         3217 non-null   object 
 1   country       3107 non-null   object 
 2   release_year  3217 non-null   int64  
 3   rating        3215 non-null   object 
 4   id            3217 non-null   int64  
 5   revenue       3217 non-null   int64  
 6   vote_count    3217 non-null   int64  
 7   vote_average  3217 non-null   float64
 8   runtime       3188 non-null   float64
 9   popularity    3217 non-null   float64
dtypes: float64(3), int64(4), object(3)
memory usage: 276.5+ KB
```

Initial inspection of the merged Netflix datasets

```
country_null_values.head()
```

	title	country	release_year	rating	id	revenue	vote_count	vote_average	runtime	popularity
0	Joaquín Reyes: Una y no más	NaN	2017	TV-MA	474599	0	1	6.0	77.0	0.600
1	Out of Thin Air	NaN	2017	TV-14	455479	0	35	6.3	85.0	3.476
2	Cultivating the Seas: History and Future of th...	NaN	2019	TV-G	648825	0	0	0.0	45.0	0.634
3	Jugaad	NaN	2017	TV-14	15646	0	3	5.3	114.0	0.751
4	Evelyn	NaN	2019	TV-MA	21868	0	61	6.5	92.0	8.942

```
.....................
88 | Extracting country information for Movie-ID: 230015
.....................
89 | Extracting country information for Movie-ID: 724061
.....................
90 | Extracting country information for Movie-ID: 750813
.....................
91 | Extracting country information for Movie-ID: 752925
.....................
92 | Extracting country information for Movie-ID: 567524
.....................
93 | Extracting country information for Movie-ID: 623618
.....................
94 | Extracting country information for Movie-ID: 568030
.....................
95 | Extracting country information for Movie-ID: 567788
.....................
96 | Extracting country information for Movie-ID: 623976
```

Cleaning & Wrangling

```
# Import movies_num_final.csv
movies_num_final = pd.read_csv('movies_num_final.csv')
movies_num_final.head()
```

	ReleaseDate	Movie	ProductionBudget	DomesticGross	WorldwideGross
0	Apr 23, 2019	Avengers: Endgame	$400,000,000	$858,373,000	$2,797,800,564
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$379,000,000	$241,063,875	$1,045,663,875
2	Apr 22, 2015	Avengers: Age of Ultron	$365,000,000	$459,005,868	$1,396,099,202
3	Dec 16, 2015	Star Wars Ep. VII: The Force Awakens	$306,000,000	$936,662,225	$2,065,478,084
4	Apr 25, 2018	Avengers: Infinity War	$300,000,000	$678,815,482	$2,044,540,523

```
movies_num_final.head()
```

	ReleaseDate	Movie	ProductionBudget	DomesticGross	WorldwideGross
0	Apr 23, 2019	Avengers: Endgame	400000000	858373000	2797800564
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	379000000	241063875	1045663875
2	Apr 22, 2015	Avengers: Age of Ultron	365000000	459005868	1396099202
3	Dec 16, 2015	Star Wars Ep. VII: The Force Awakens	306000000	936662225	2065478084
4	Apr 25, 2018	Avengers: Infinity War	300000000	678815482	2044540523

Final dataframe of the cleaned movies budget data set

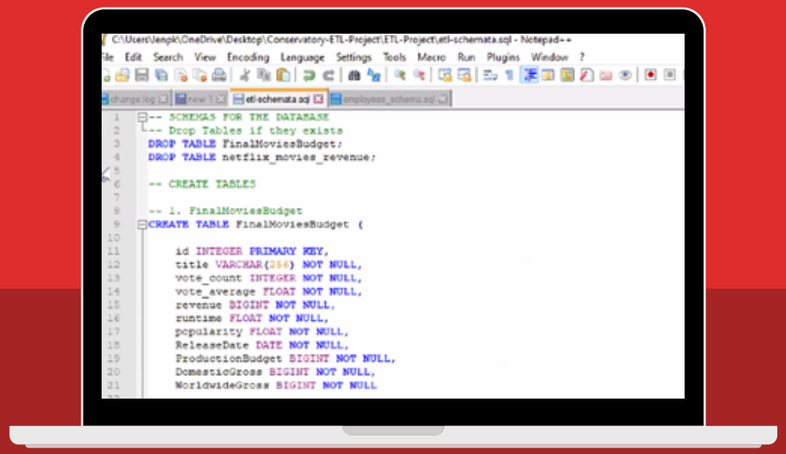
Final merged dataframe with no null values

```
MoviesBudget.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4739 entries, 0 to 4819
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   id                4739 non-null   int64
 1   title             4739 non-null   object
 2   vote_count        4739 non-null   int64
 3   vote_average      4739 non-null   float64
 4   revenue           4739 non-null   int64
 5   runtime           4739 non-null   float64
 6   popularity        4739 non-null   float64
 7   ReleaseDate       4739 non-null   object
 8   ProductionBudget  4739 non-null   int64
 9   DomesticGross     4739 non-null   int64
 10  WorldwideGross    4739 non-null   int64
dtypes: float64(3), int64(6), object(2)
memory usage: 444.3+ KB
```

Load

- Create schemata
- Load tables into PostgreSQL & pandas
- Inspect data
- One of the reasons we chose Postgres is because our data is mostly tabular



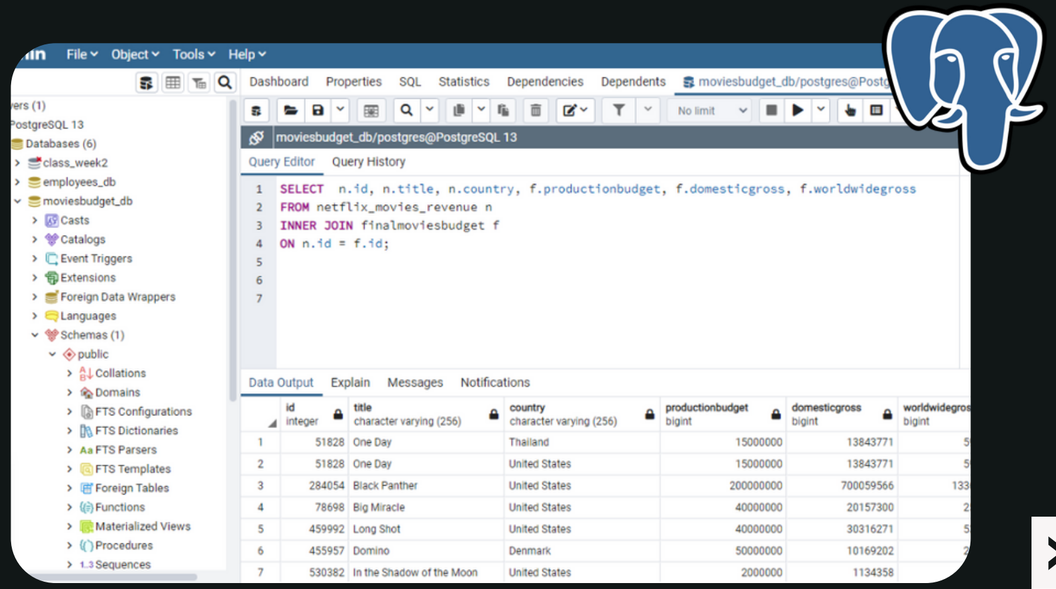
ETL - LOAD

```
%matplotlib inline
```

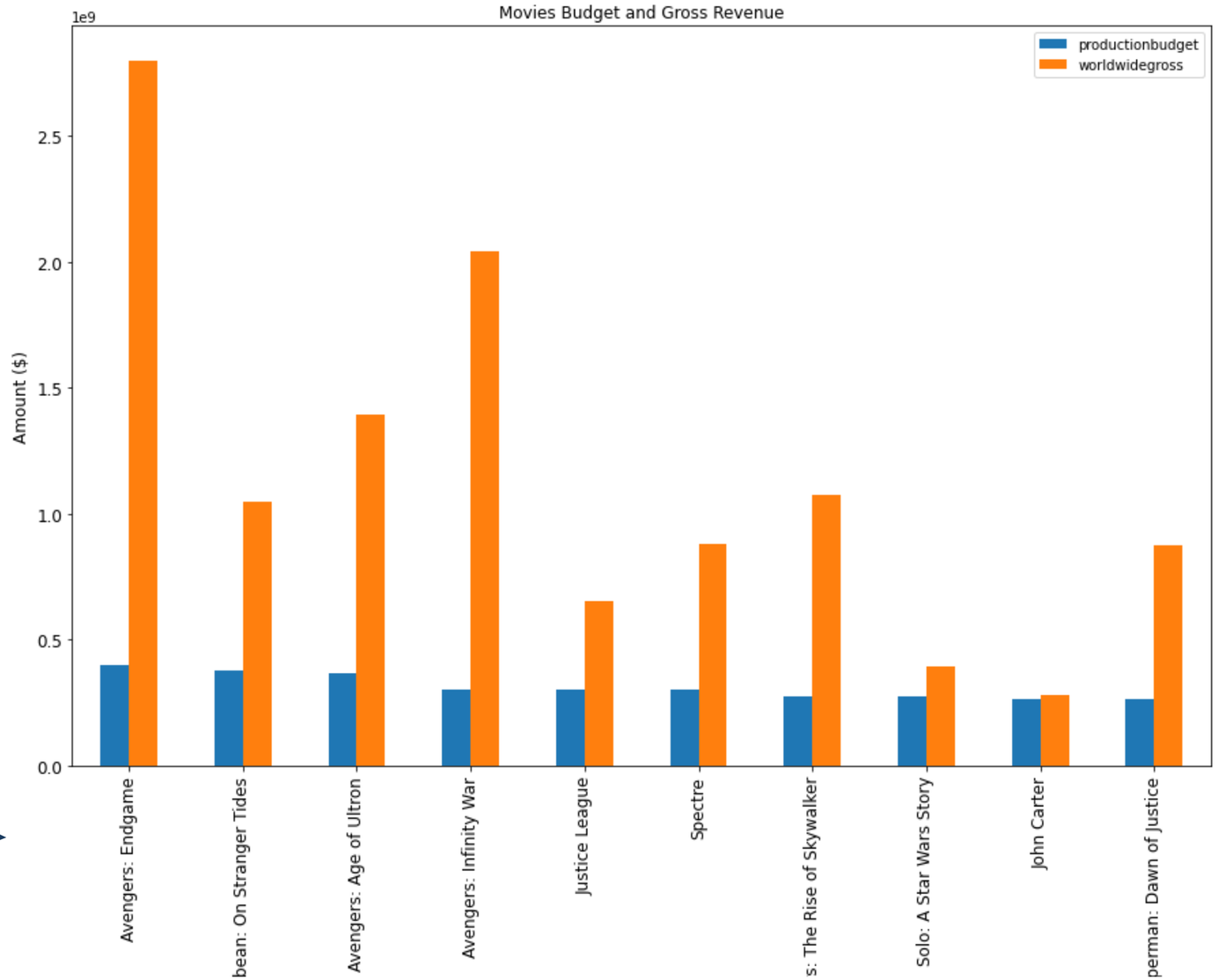
```
# import dependencies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sqlalchemy import create_engine
from config import DATABASE_URI
import datetime
```

```
# Create connection
engine = create_engine(DATABASE_URI)
conn = engine.connect()
```

```
# Import csv files to load in database and inspect
# FinalMoviesBudget
FinalMoviesBudget = pd.read_csv('FinalMoviesBudget.csv')
FinalMoviesBudget.head()
```



Top 10 Movies with Highest Budget and Revenue




THANK
YOU!

Top 10 Movies with Highest Budget and Revenue

0	Avengers: Endgame	400000000	2797800564
1	Pirates of the Caribbean: On Stranger Tides	379000000	1045663875
2	Avengers: Age of Ultron	365000000	1396099202
3	Avengers: Infinity War	300000000	2044540523
4	Justice League	300000000	655945209
5	Spectre	300000000	879500760
6	Star Wars: The Rise of Skywalker	275000000	1072944222
7	Solo: A Star Wars Story	275000000	393151347
8	John Carter	263700000	282778100
9	Batman v Superman: Dawn of Justice	263000000	872395091