Seattle AirBnb Data Engineering

Kylie Li, Kerim Celik, Elizabeth Morgan, Mari Awaisi

WHY USE THIS DATASET?

*Optimizing listings and forecasting demand

*Market trends analysis

*Host performance and customer satisfaction insights

*Pricing insights

DATA ETHICS CONSIDERATIONS

*Protect the privacy of hosts and guests names, addresses, etc. Limit access or remove personal contact information (host names, latitude/longitude, etc.)

*Be aware of biases in guest reviews and ensure that predictive modeling does not inadvertently perpetuate biases or disadvantage certain hosts.

*Consider the impact of AirBnB on the local community - from noise complaints to housing availability and affordability - when analyzing data such as price trends and review comments.

ETL: EXTRACT

WHY WE CHOSE THE COLUMNS WE DID

*We eliminated extraneous and redundant columns (city, country, etc.) as well as some personal information (names, latitude/longitude, etc)

*Unique identifiers (host_id, etc.)

*Temporal analysis of pricing trends over different months (month and price for calendar_average_pricing table)

*Enable analysis of pricing trends based on property types and neighborhoods (property_average_pricing)

*Capture guest satisfaction and listing engagement and popularity (listing_scores)

*Provide detailed information about each listing's characteristics, pricing, and policies (listings_cleaned)

ETL: TRANSFORM

WHY DID WE MAKE CERTAIN TRANSFORMATIONS? : Enhancing data usability and quality

Object => Float

```
sorted_hosts_df['host_response_rate'].fillna(-1.0,inplace=True)
sorted_hosts_df['host_response_rate'] = sorted_hosts_df['host_response_rate'].astype('float64')
sorted_hosts_df['host_acceptance_rate'].fillna(-1.0,inplace=True)
sorted_hosts_df['host_acceptance_rate'] = sorted_hosts_df['host_acceptance_rate'].astype('float64')
```

Object => Boolean

```
base_listings_df['instant_bookable'] = base_listings_df['instant_bookable'].str.replace('t','True')
base_listings_df['instant_bookable'] = base_listings_df['instant_bookable'].str.replace('f','False').astype('bool')
```

Object => Integer

host_verifications
['email', 'phone', 'facebook']
['email', 'phone', 'reviews', 'kba']
['email', 'phone', 'linkedin', 'reviews', 'kba']

host_verifications_count
3
4
5

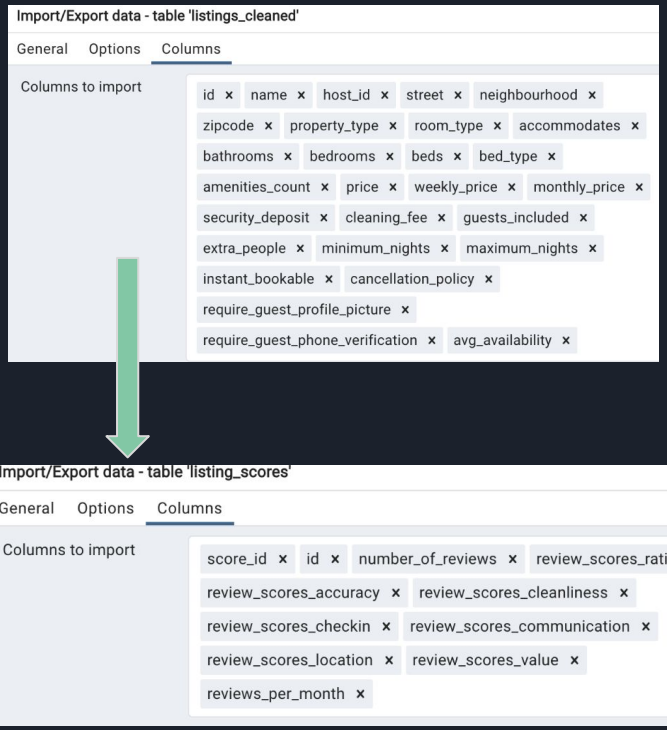
```
sorted_hosts_df['host_verifications'] = sorted_hosts_df['host_verifications'].apply(lambda v: str(v.count(',') + 1))
sorted_hosts_df['host_verifications'] = sorted_hosts_df['host_verifications'].astype('int')
sorted_hosts_df = sorted_hosts_df.rename(columns={ 'host_verifications': 'host_verifications_count'})
```

ETL: Load (Listings tables)

- Listings and listing score tables

```
CREATE TABLE listings_cleaned (
    id INT   PRIMARY KEY,
    name VARCHAR   NOT NULL,
    host_id INT   NOT NULL,
    street VARCHAR   NOT NULL,
    neighbourhood VARCHAR   NOT NULL,
    zipcode VARCHAR   NOT NULL,
    property_type VARCHAR   NOT NULL,
    room_type VARCHAR   NOT NULL,
    accommodates INT   NOT NULL,
    bathrooms FLOAT   NOT NULL,
    bedrooms FLOAT   NOT NULL,
    beds FLOAT   NOT NULL,
    bed_type VARCHAR   NOT NULL,
    amenities_count INT   NOT NULL,
    price FLOAT   NOT NULL,
    weekly_price FLOAT   NOT NULL,
    monthly_price FLOAT   NOT NULL,
    security_deposit FLOAT   NOT NULL,
    cleaning_fee FLOAT   NOT NULL,
    guests_included INT   NOT NULL,
    extra_people FLOAT   NOT NULL,
    minimum_nights INT   NOT NULL,
    maximum_nights INT   NOT NULL,
    instant_bookable BOOLEAN   NOT NULL,
    cancellation_policy VARCHAR   NOT NULL,
    require_guest_profile_picture BOOLEAN   NOT NULL,
    require_guest_phone_verification BOOLEAN   NOT NULL,
    avg_availability FLOAT   NOT NULL,
);
```

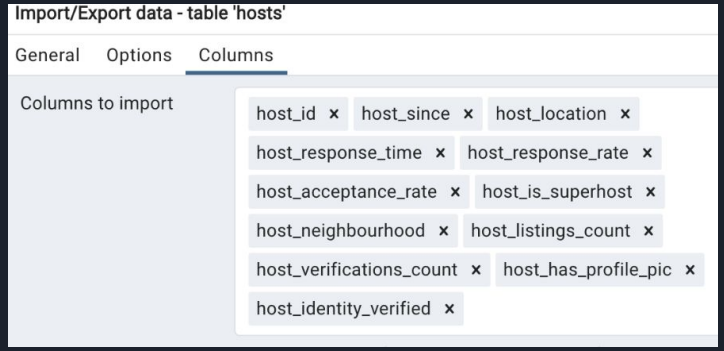
```
CREATE TABLE listing_scores (
    score_id SERIAL PRIMARY KEY,
    id INT NOT NULL,
    number_of_reviews INT   NOT NULL,
    review_scores_rating INT   NOT NULL,
    review_scores_accuracy INT   NOT NULL,
    review_scores_cleanliness INT   NOT NULL,
    review_scores_checkin INT   NOT NULL,
    review_scores_communication INT   NOT NULL,
    review_scores_location INT   NOT NULL,
    review_scores_value INT   NOT NULL,
    reviews_per_month FLOAT   NOT NULL,

    FOREIGN KEY (id) REFERENCES listings_cleaned(id)
);
```



ETL: Load (Hosts and Calendar tables)

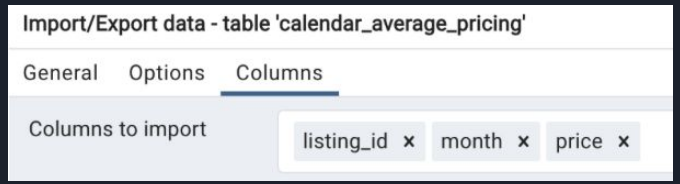
- Hosts - Data related to hosts and their properties

```
CREATE TABLE hosts (
    host_id INT  PRIMARY KEY,
    host_since DATE  NOT NULL,
    host_location VARCHAR   NOT NULL,
    host_response_time VARCHAR   NOT NULL,
    host_response_rate FLOAT   NOT NULL,
    host_acceptance_rate FLOAT   NOT NULL,
    host_is_superhost BOOLEAN   NOT NULL,
    host_neighbourhood VARCHAR   NOT NULL,
    host_listings_count FLOAT   NOT NULL,
    host_verifications_count INT   NOT NULL,
    host_has_profile_pic BOOLEAN   NOT NULL,
    host_identity_verified BOOLEAN   NOT NULL
);
```



- Calendar table - Data for listing fields that change over time

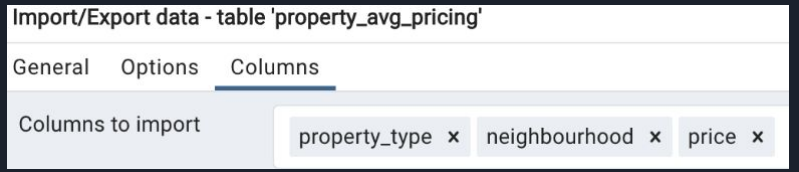
```
CREATE TABLE calendar_average_pricing (

    listing_id INT NOT NULL,
    month INT   NOT NULL,
    price FLOAT   NOT NULL,

    FOREIGN KEY (listing_id) REFERENCES listings_cleaned(id)
);
```



ETL: Load (Property Table)

- Property table- Data for average price for each combination of property type and neighborhood

```
CREATE TABLE property_avg_pricing (
    property_type VARCHAR   NOT NULL,
    neighbourhood VARCHAR   NOT NULL,
    price FLOAT  NOT NULL,

    PRIMARY KEY (property_type,  neighbourhood)-- Composite primary key
);
```



- Challenges encountered

❗ ERROR: null value in column "neighbourhood" of relation "listings_cleaned" violates not-null constraint

❗ ERROR: insert or update on table "listings_cleaned" violates foreign key constraint "listings_cleaned_host_id_fkey"
DETAIL: Key (host_id)=(42515980) is not present in table "hosts".

DATA DISPLAY

Original Listings DataFrame

	id	listing_url	scrape_id	last_scraped	name	summary	space	description	experiences_offered	neighborhood_
0	241032	https://www.airbnb.com/rooms/241032	20160104002432	2016-01-04	Stylish Queen Anne Apartment	NaN	Make your self at home in this charming one-be...	Make your self at home in this charming one-be...	none	

1 rows × 92 columns

Extracted reviews

score_id	listing_id	number_of_reviews	review_scores_rating	review_scores_accuracy	review_scores_cleanliness	review_scores
0	241032	207	95	10	10	
1	953595	43	96	10	10	
2	3308979	20	97	10	10	

Created Sub Dataset to list out Hosts

	host_id	host_url	host_name	host_since	host_location	host_about	host_response_tim
7754	4962900	https://www.airbnb.com/users/show/4962900	Jordan	2013-02-04	Spokane, Washington, United States	Stay Alfred was created based on the idea of o...	within an hou
7755	4962900	https://www.airbnb.com/users/show/4962900	Jordan	2013-02-04	Spokane, Washington, United States	Stay Alfred was created based on the idea of o...	within an hou
7756	4962900	https://www.airbnb.com/users/show/4962900	Jordan	2013-02-04	Spokane, Washington, United States	Stay Alfred was created based on the idea of o...	within an hou

DATA DISPLAY

Original Calendar DataFrame

	listing_id	date	available	price
0	241032	2016-01-04	t	$85.00
1	241032	2016-01-05	t	$85.00
2	241032	2016-01-06	f	NaN
3	241032	2016-01-07	f	NaN
4	241032	2016-01-08	f	NaN
...	...	...	...	...
1393565	10208623	2016-12-29	f	NaN
1393566	10208623	2016-12-30	f	NaN
1393567	10208623	2016-12-31	f	NaN
1393568	10208623	2017-01-01	f	NaN
1393569	10208623	2017-01-02	f	NaN

Transformed into DataFrame, that is grouped by property type / neighborhood, displaying average pricing

		price
property_type	**neighbourhood**	
Apartment	**Alki**	163.9
	Atlantic	89.4
	Ballard	119.8
	Belltown	206.6
	Bitter Lake	82.7
...	**...**	...
Townhouse	**University District**	79.5
	Wedgewood	95.7
Treehouse	**Dunlap**	48.0
	Montlake	200.0
Yurt	**North Admiral**	105.4

Flask App

Run the App in the Terminal and Open the given URL

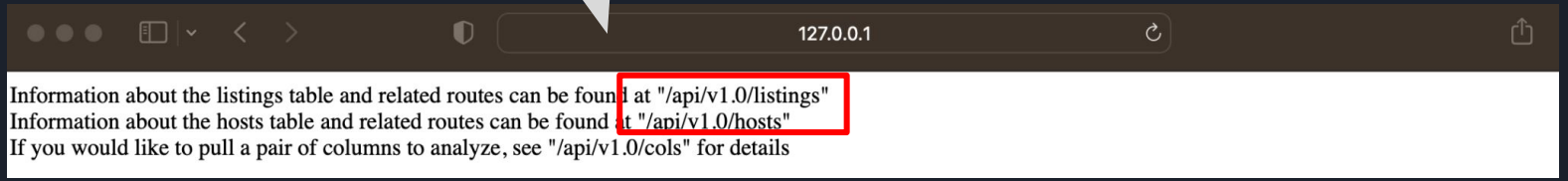
```
(base) mariaskarova@maris-air Project 3 % python flask_app.py
 * Serving Flask app 'flask_app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (fsevents)
 * Debugger is active!
 * Debugger PIN: 134-114-614
127.0.0.1 - - [23/Mar/2024 12:44:13] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Mar/2024 12:44:14] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [23/Mar/2024 12:44:33] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Mar/2024 12:44:33] "GET /favicon.ico HTTP/1.1" 404 -
```

flask_app.py 2

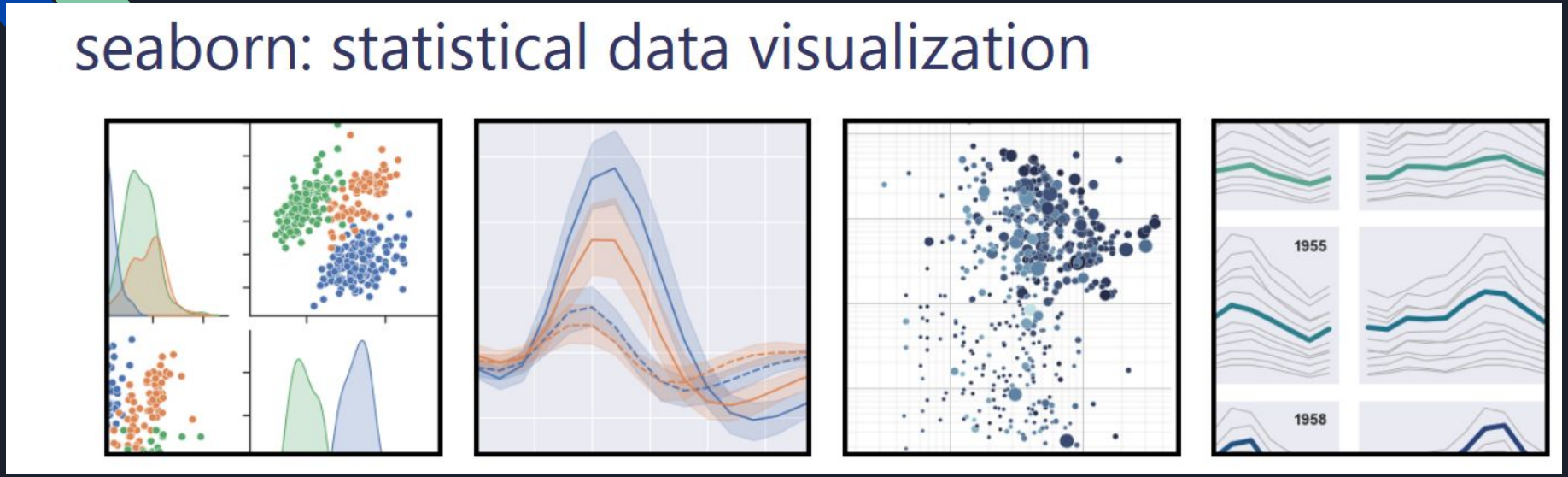
flask_app.py > ...

```python
from flask import Flask, jsonify
import pandas as pd
import json

# ===== SETUP =====
app = Flask(__name__)

listings = pd.read_csv('./Resources/listings_cleaned.csv')
hosts = pd.read_csv('./Resources/hosts.csv')

listing_cols = listings.columns.to_list()
host_cols = hosts.columns.to_list()

listing_dict = dict(enumerate(listing_cols))
host_dict = dict(enumerate(host_cols))
```

The content is served on the web page



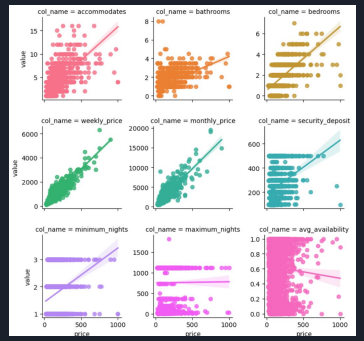
New data library: Seaborn

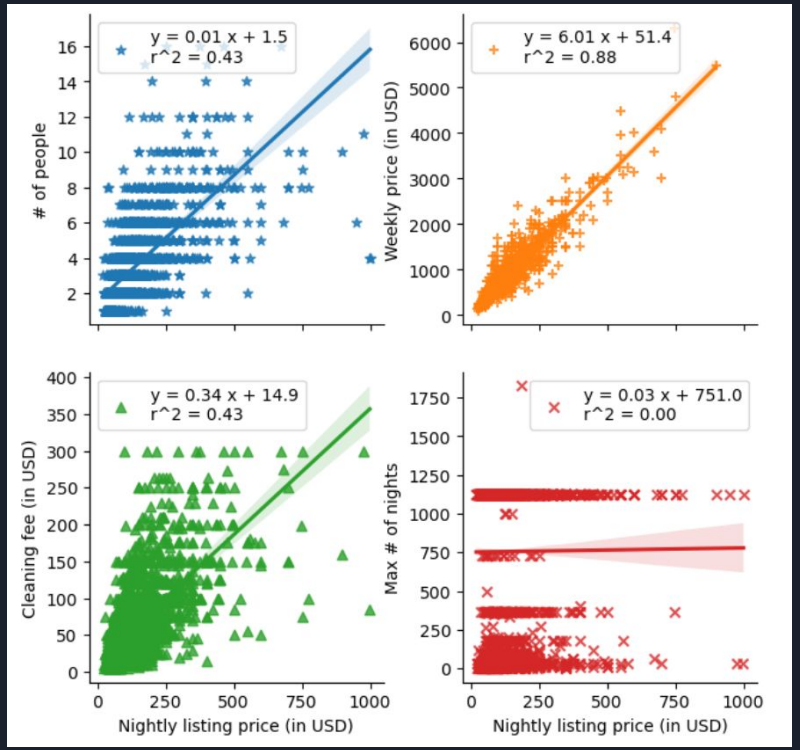


- Powerful data visualization library

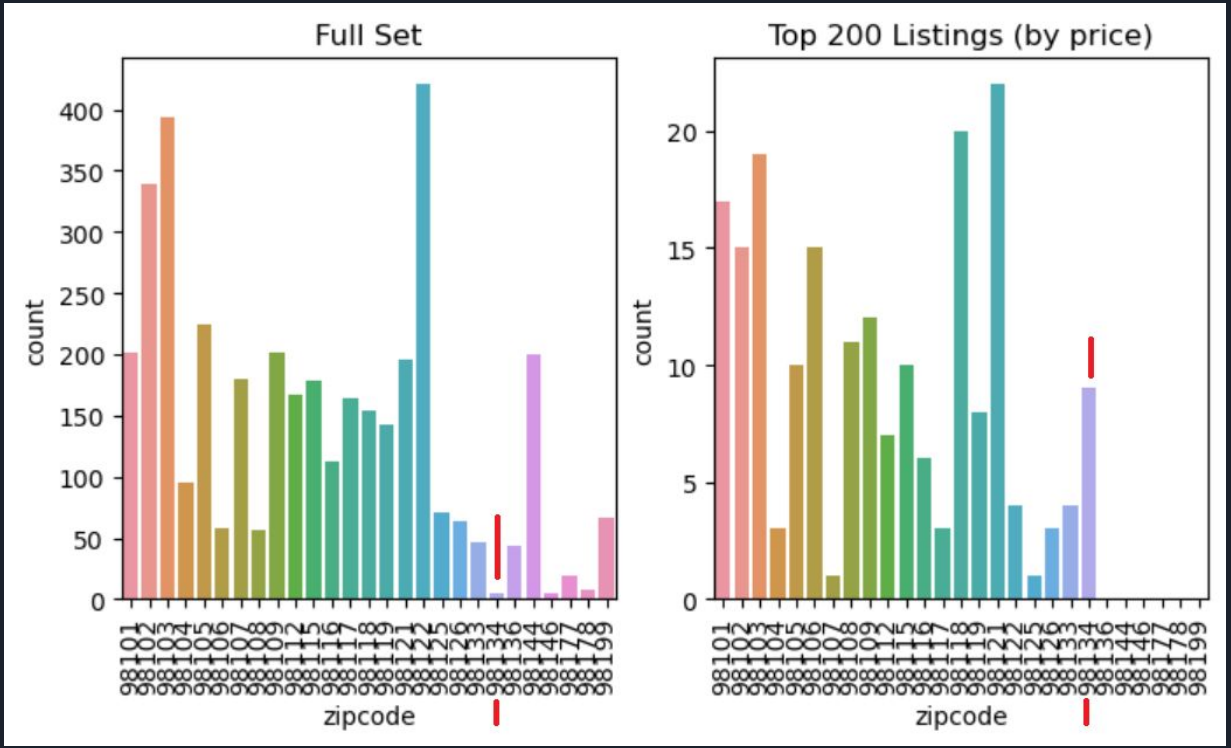
- Utilizes close functional integration with Pandas

Seaborn: continuous

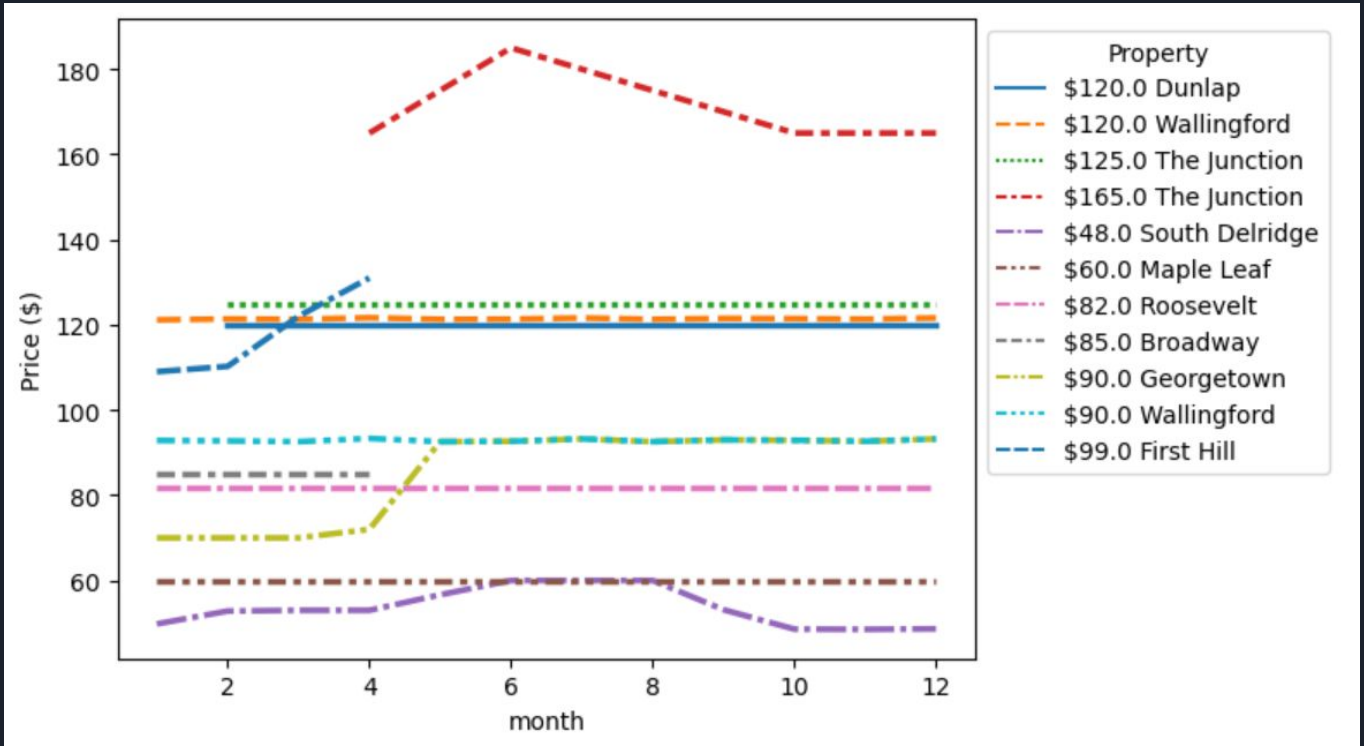




Seaborn: non-continuous



Seaborn: merged



Flask Application

- Allows access to tables via web in JSON format

- Can customize the table data you want to work with using the route
  - Subset of certain tables
  - Pair of columns from any table(s)

Flask demo

Extensibility

- Use Inside Airbnb data to bring in data related to other cities

  - Compare to old Seattle data to most recent data for the city
  - Create datasets to compare between cities
  - Combine cities' sets and find most reliable correlations

Thank you!

QUESTIONS?