

Beertopia

Scope:

For beer enthusiasts, the craft beer segment of the alcohol and beverages industry has a ton of options. Some of the questions that are relevant for beer enthusiasts are as follows:

1. How are craft beers differentiated?
2. What are the most common beer styles consumed in the US?
3. What are the states with the most craft beer consumption?

To answer these questions, we cleaned our dataset and later built a web application that shows various beer styles, a map showing the state with most beer consumption, a data table, and a map showing brewery ratings.

Data sources:

Beers and Breweries – 2 datasets from Kaggle were used. One for US craft breweries and another for beers: <https://www.kaggle.com/ritesaluja/the-beer/data>

Google Places – API interactions to retrieve geolocation data and brewery ratings.

Web Scrape – Scrape data for beer consumption by state:
<https://www.thrillist.com/drink/nation/beer-consumption-by-state/3973432>

Beer Styles - The sources to map and define the family tree used were:

- www.beeradvocate.com
- www.brewersassociation.org
- www.beerspot.com
- www.ratebeer.com

API Keys Required:

- Google API (console.developers.google.com/getting-started)
- Mapbox API (mapbox.com)

Technologies used:

- Python/Pandas
- Javascript

- PHP
- HTML
- CSS/BootStrap

Data Munging:

The dataset had 12 number of columns and 2410 number of rows. Since our dataset was so large, we had to remove unnecessary data by using Pandas. Our final dataset which only consists of breweries located in US resulting in 9 columns and 2255 rows of data.

Open Git bash, activate your python environment and install libraries mentioned below:

- pandas
- sqlalchemy
- psycopg2
- bs4
- datetime

Sign up for Google API and acquire key to copy. Create “config.py” file in the root directory and assign Google API key as “gkey” in the config file.

Create 2 Jupyter Notebooks:-

1. Load breweries and beers CSVs into jupyter notebook. Make Google Place API calls for each brewery to extract latitude, longitude, and ratings data. Munge data to create a final DataFrame. Export to CSV for data visualization.
2. Scrape beer consumption data using BeautifulSoup and import into jupyter notebook. Data munge and format the data into a pandas DataFrame. Export to CSV for data visualization.

Local Database

PostgreSQL to store data from Jupyter Notebook. Two tables created :-

- i. Table with breweries and beers data from the first notebook.
- ii. Table with beer consumption per state from the second notebook

Data Visualization (JavaScript libraries used)

1. Beer style and aroma wheel (Sunburst):

A research was made to create a family tree of beer styles, defining the parent for each beer style, ABV, IBU and a description of the style.

2. Consumption map (Choropleth):

Choropleth map to show Average annual consumption per capita. Plotly was used to create this chart as well as mapbox api for location mode. The color range describes the average beer consumed by each state in Gallons. When hovered over the map it shows average consumption and state name.

3. Beer Style Frequency (Interactive Word cloud)

- Word cloud shows thirty most used beer style in the United States. The font size is dictated by the number of beer types associated with that particular beer style. When hovered over it shows beer style and number of beers associated with it.
- Data for word cloud was modified from beers_df.csv. Only beer style name and beer counts were needed. To get the desired top thirty beer style with beer counts, jupyter notebook was used to munge the data.

```
df_style=beer_data[["style","beer_name"]]  
style_df=df_style.groupby(['style']).count()  
style_df= style_df.sort_values(['beer_name'], ascending=False)  
final_style= style_df.iloc[0:30]
```

- Anychart library was used to create the word cloud.

Data Table

The data table section (data.html) shows the entire brewery dataset. The records are filterable by 4 fields:

- Brewery Names
- State
- Beer Style
- Brewery Ratings

Data table is being displayed by calling a .js file with json entries for each record which was created by running a simple csv to json conversion using python.

```

import csv
import json

csvfile = open('final_beer.csv', 'r')
jsonfile = open('final_beer.json', 'w')

fieldnames =
("index", "brewery_name", "city", "state", "beer_name", "style", "lati
tude", "longitude", "rating", "abv_percent")
reader = csv.DictReader( csvfile, fieldnames)
for row in reader:
    json.dump(row, jsonfile)
    jsonfile.write(',\n')

```

The purpose of showing a data table is to have users answer the following questions:

- Which breweries are located near me (or to a particular place)?
- Which beers are produced by these breweries (with ratings and abv info)?
- Which breweries have the best rating?

Deployment to Heroku

Heroku is a hosting platform where you can deploy dynamic applications in Rails, PHP, Node.js and Python web applications.

Prerequisites

Steps to setup Heroku account:

1. Have git installed
2. Sign up for Heroku Account

Deploying Your Site

- First, you need to navigate to your project in the git bash/command prompt.

```
cd Final
```

- Create an index.php file. The purpose of this file is to trick Heroku to deploy a static site by including 1 dynamic file.
- This is a simple way to masquerade the HTML app you build as a PHP app. Next include one line of PHP code into the index.php.

```
<?php header( 'Location: /index.html' ) ; ?>
```

- Open git bash or command prompt. Run the command:

```
git init
```

```
git add .
```

The add . means add all the files to the git repository.

- Then commit

```
git commit -m "Ready for deployment."
```

- Login into Heroku site before you run the next command

```
heroku apps:create my-website-name
```

- Insert your desired name instead of my-website-name.

```
git push heroku master
```

- Once you see “remote: Verifying deploy. done.” It means the site is ready to view.
- You can now visit your site at <https://my-website-name.herokuapp.com/>.

If you need to make changes to your site after deployment, please follow below steps:

- Add the changes...

```
git add .
```

- Save the changes...

```
git commit -am "updated files"
```

- Then deploy...

```
git push heroku master
```

Challenges/Limitations

- For Breweries with multiple locations - Google API provided the first address it was able to find even though that might not be the actual one.
- The rating for individual beers were not available, the indicated rating in data table is for brewery as a whole.
- Using multiple data sources
- We had to create a family tree to connect beers and their style.

Next steps:

- Make the application dynamic to make it interactive with users (i.e users could rate beers or add brewery information)
- Add beer rating information.
- Provide an accurate street address for each brewery.
- Evaluate adding more fun charts.
- Dictionary of terms

Contributors

- Sai
- Dan
- Deepen
- Pratikchhya
- Pavana

How to run the application on live server

- Git clone the final folder
- Open Git bash
- Find the folder and run the following command

```
cd final
```

```
python -m http.server
```

- Open web browser and access your localhost i.e <http://localhost:8000/>

Production Website

<https://visualizing-doubles.herokuapp.com/index.html>

Data sources

<https://www.kaggle.com/ritesaluja/the-beer/data>

<https://www.thrillist.com/drink/nation/beer-consumption-by-state/3973432>

beerAdvocate.com

beersport.com

Ratebeer.com,

BJCP.org,

Brewersassociation.org,

learn.kegerator.com