Robert Sandor

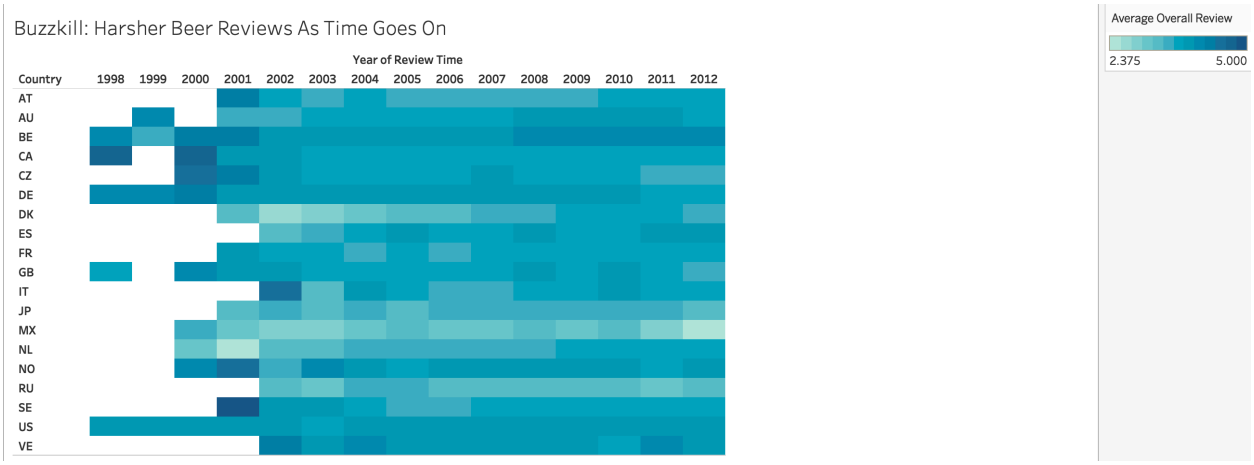
Data Visualization - Beer Reviews

This dataset consists of beer reviews from beeradvocate.com. The data span a period of more than 10 years, including all ~1.5 million reviews up to 2012. Each review includes ratings in terms of five "aspects": appearance, aroma, palate, taste, and overall impression. Reviews include product and user information, followed by each of these five ratings. In addition, I used the vaderSentiment library to generate sentiment scores for both the beer names and brewery names, the Google Places API to acquire location data, and Tableau-generated latitude and longitude that correspond to the Google Places location data. I acquired it from the website data.world (https://data.world/socialmediadata/beeradvocate). I chose this particular dataset because I thought beer would be an interesting subject to investigate and this was one of the few datasets related to beer that had numerous rows. I was hoping to explore and find interesting relationships between consumers' opinions across time and geography.

Summary

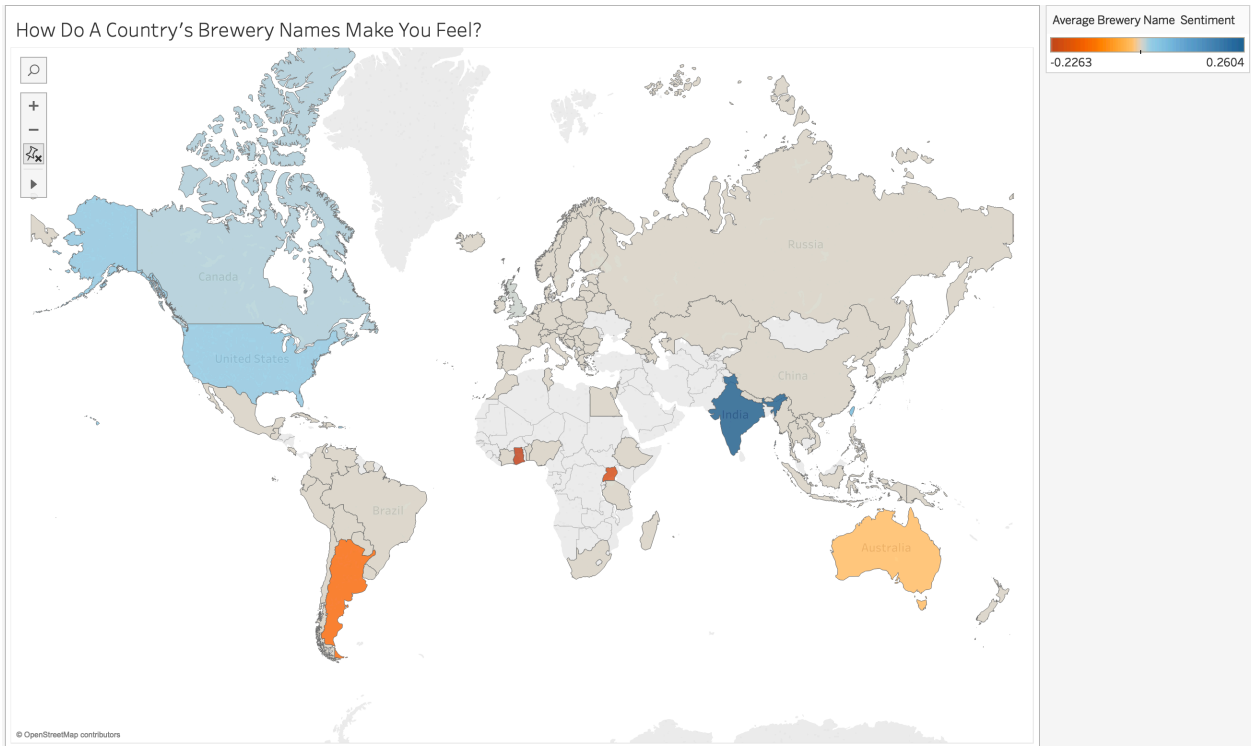
Buzzkill: Harsher Beer Reviews As Time Goes On

Here I used a heat map from Tableau to look at the average overall reviews of the top 19 countries from between 1998 and 2012. Many countries' average overall review scores declined over time which seems to imply that either the first reviewers were generally more lenient upon the beer ratings or reviewers became more critical as time went on.



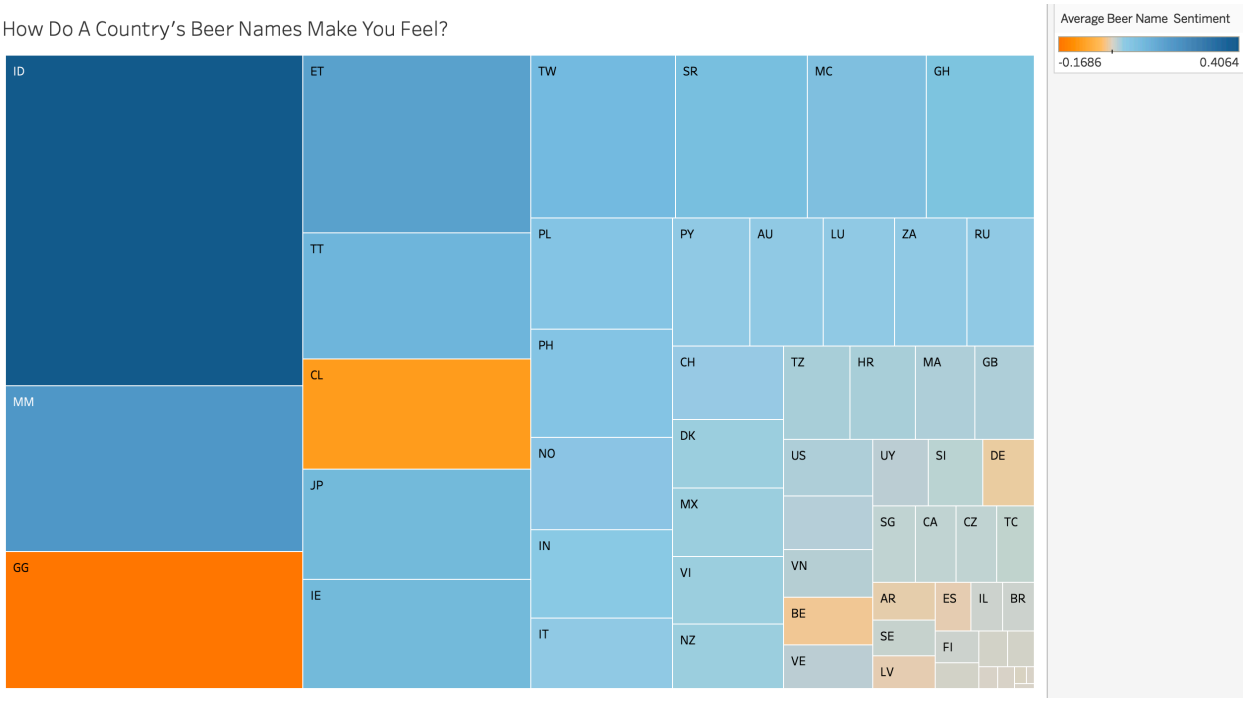
How Do A Country's Brewery Names Make You Feel?

Here I used a choropleth from Tableau to look at an average sentiment score of brewery names by country. While most countries and brewery names ended up having a neutral score, I found it interesting that the countries with more positive sentiment ended up being in the northern hemisphere and the countries with more negative sentiment ended up in the southern hemisphere.



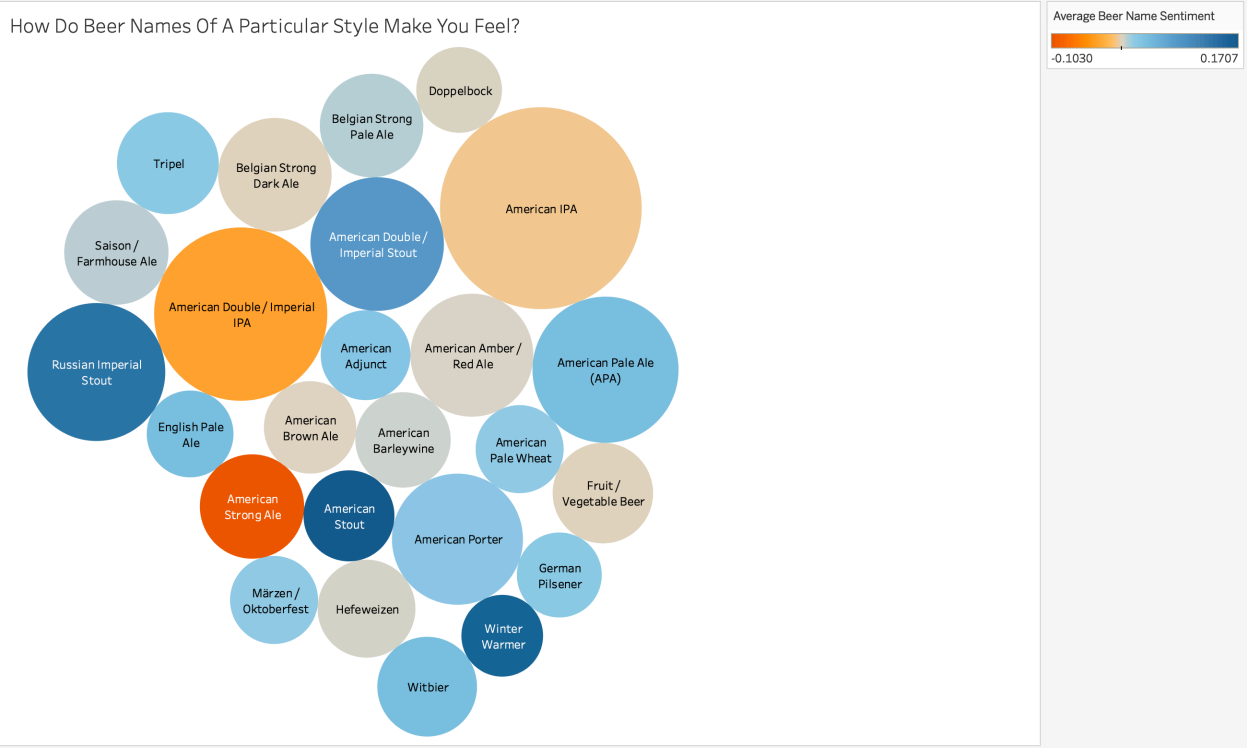
How Do A Country's Beer Names Make You Feel?

Here I used a treemap from Tableau to explore the relationship between the average sentiment score of the beer names within a country. Most countries' beers ended up having positive sentiment scores with a few notable exceptions. I found it interesting that there was no significant relationship between the sentiment score of a country's brewery names (like from the How Do A Country's Brewery Names Make You Feel? visualization) and the sentiment score of the beer names within the same country.



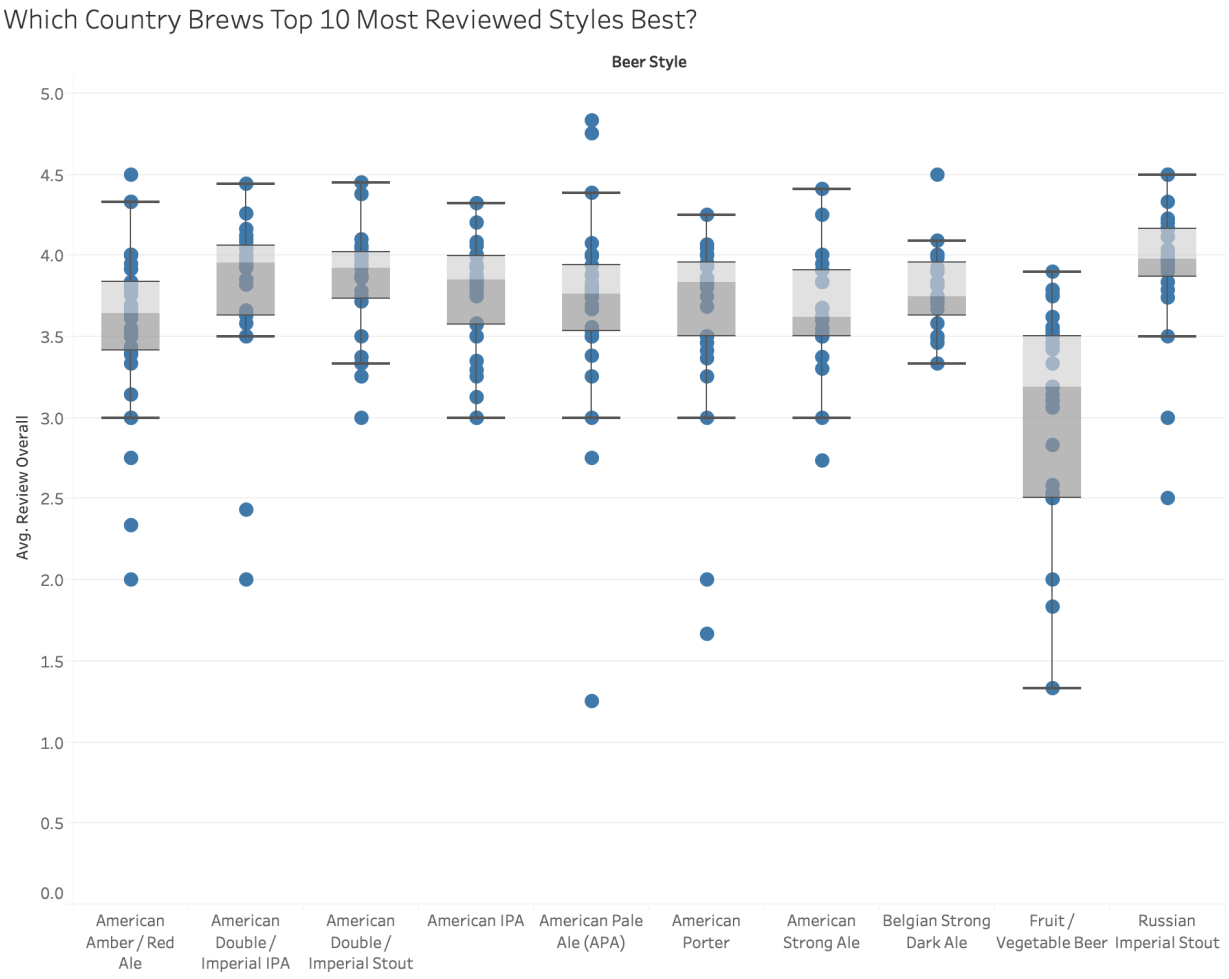
How Do Beer Names Of A Particular Style Make You Feel?

Here I used a bubble plot from Tableau to show the relationships between the style of beer, how reviewed the beer was, and the sentiment score of the beer names within the same category. The most reviewed beer styles seem to have lower sentiment scores (American IPA and American Double/Imperial IPA) but the styles of the most reviewed beers generally had positive sentiment scores.



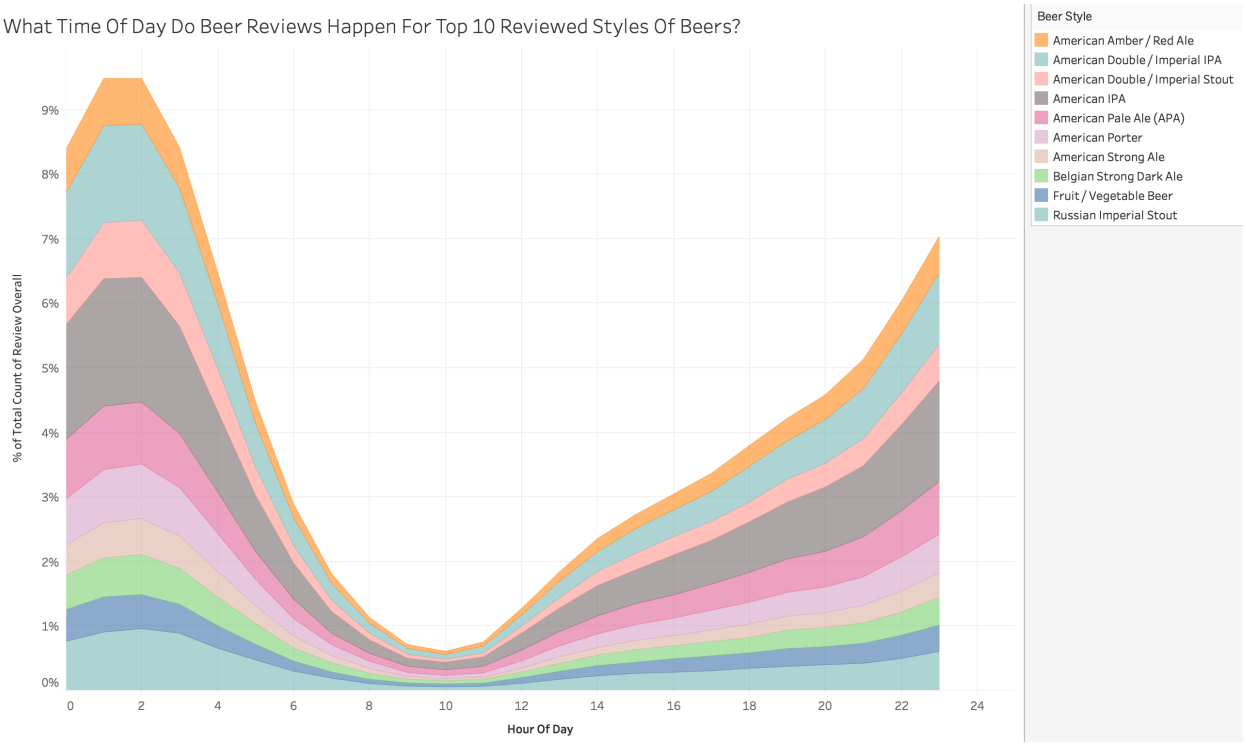
Which Country Brews Top 10 Most Reviewed Styles Best?

Here I used a box plot from Tableau to find out which countries make the best version of the top 10 reviewed styles of beers and the overall distribution of scores by country. What I found surprising was the number of countries outside of America that brewed “American” styles even better than the USA.



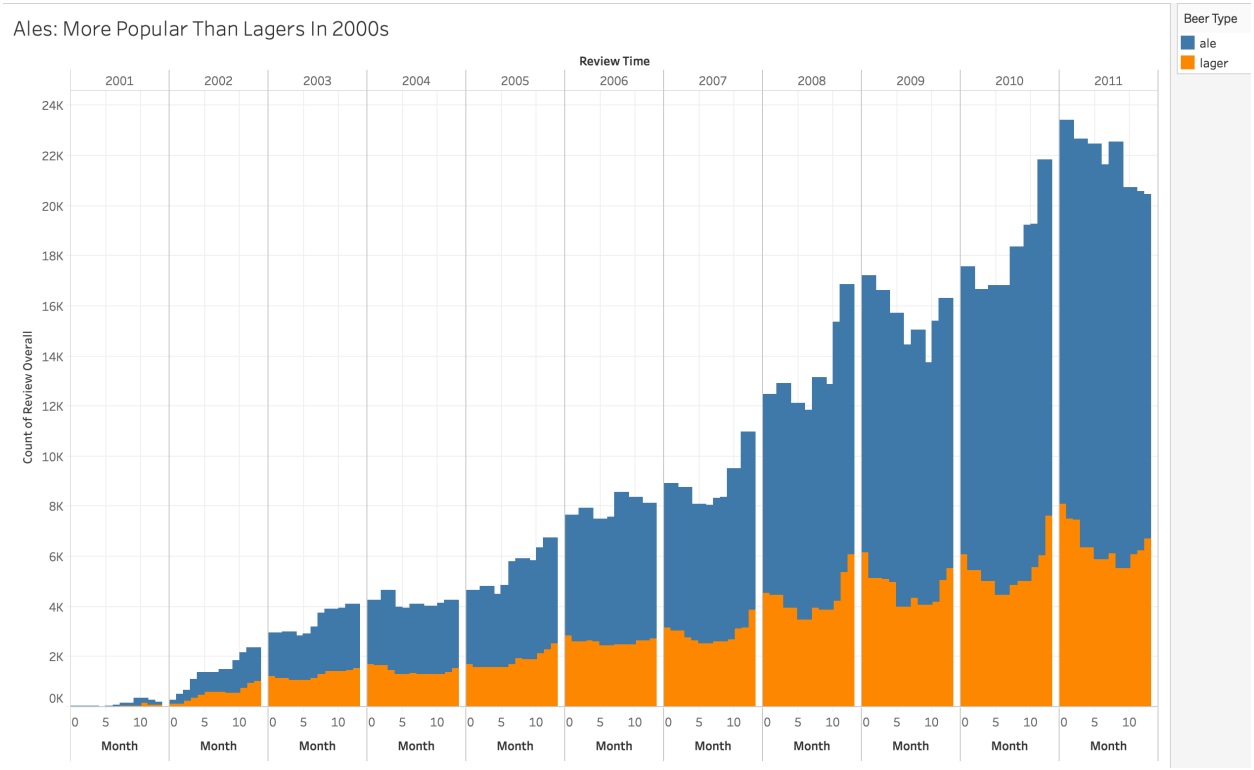
What Time Of Day Do Beer Reviews Happen For Top 10 Reviewed Styles Of Beers?

Here I used a stacked area graph from Tableau to find out the time of day at which these reviews were made. Naturally, a majority of the reviews were after work hours. I expected a somewhat bimodal distribution (for “wet lunches” and after work hours), but that wasn’t quite the case. The lowest activity was between 8am and 11am after which activity rose until the peak hours between 1 and 2am and then steadily declined.



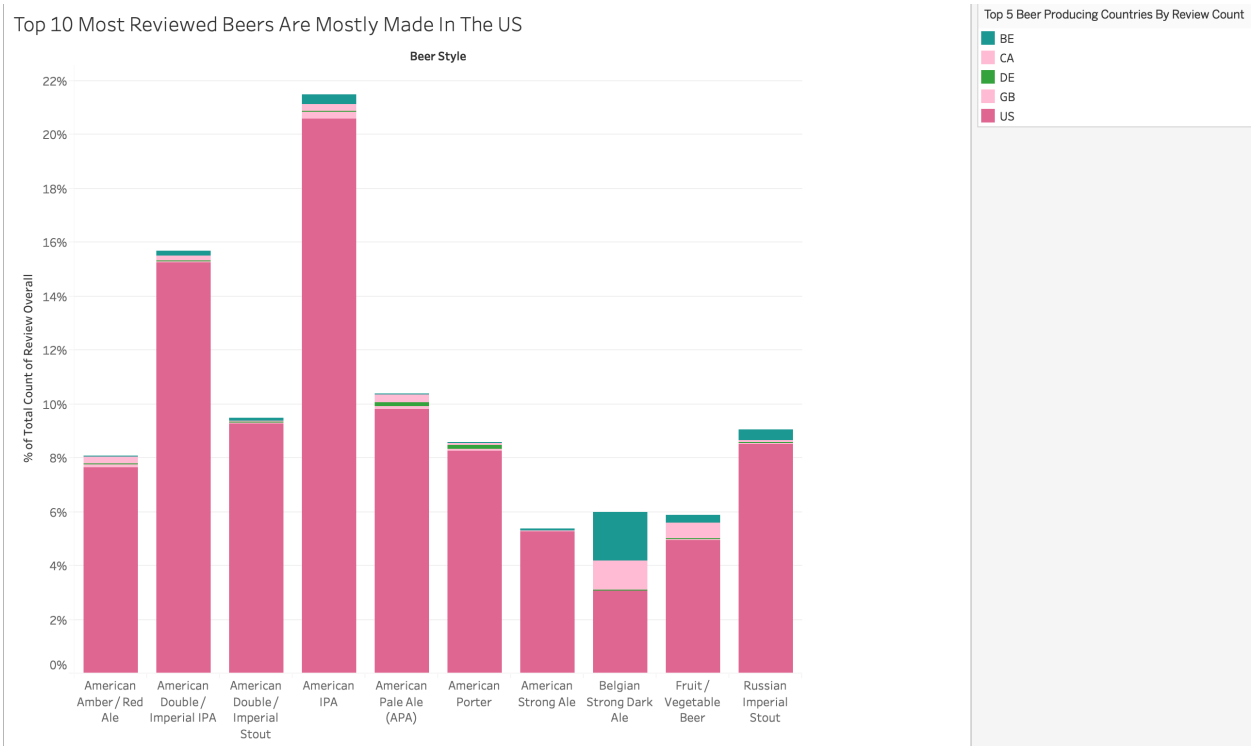
Ales: More Popular Than Lagers In 2000s

Here I used a bar graph from Tableau to compare the number of beer reviews between the beers that were aggregated as ales and lagers between 2001 and 2011. While the number of beer reviews for each type increased over time, it seems that ales were reviewed more over the same period of time, especially during the months of November, December, and January.



Top 10 Most Reviewed Beers Are Mostly Made In The US

Here I used a histogram from Tableau to look at the 10 most reviewed beer styles to find out what proportion of each style was made in the top 5 countries by reviews. Of the top 10 most reviewed beer styles, the US was the major producer of those styles, followed by Belgium, Canada, Germany, and Great Britain.

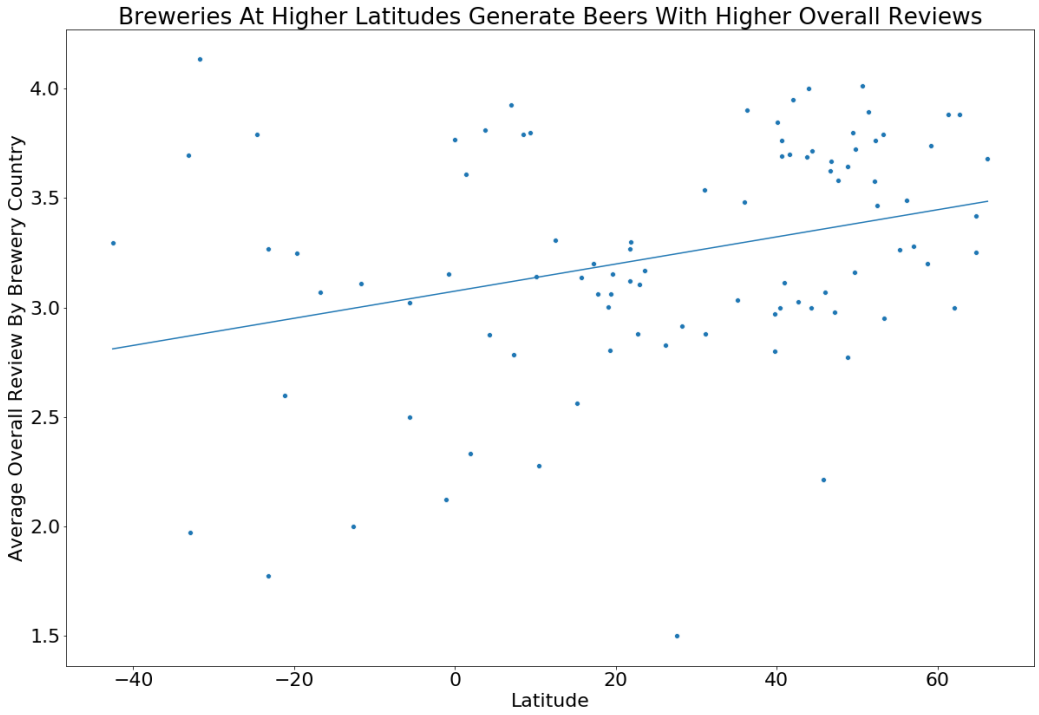


Path From Brewery to Consumer

Here I used a Plotly graph to look at the paths from brewing countries to reviewer countries.

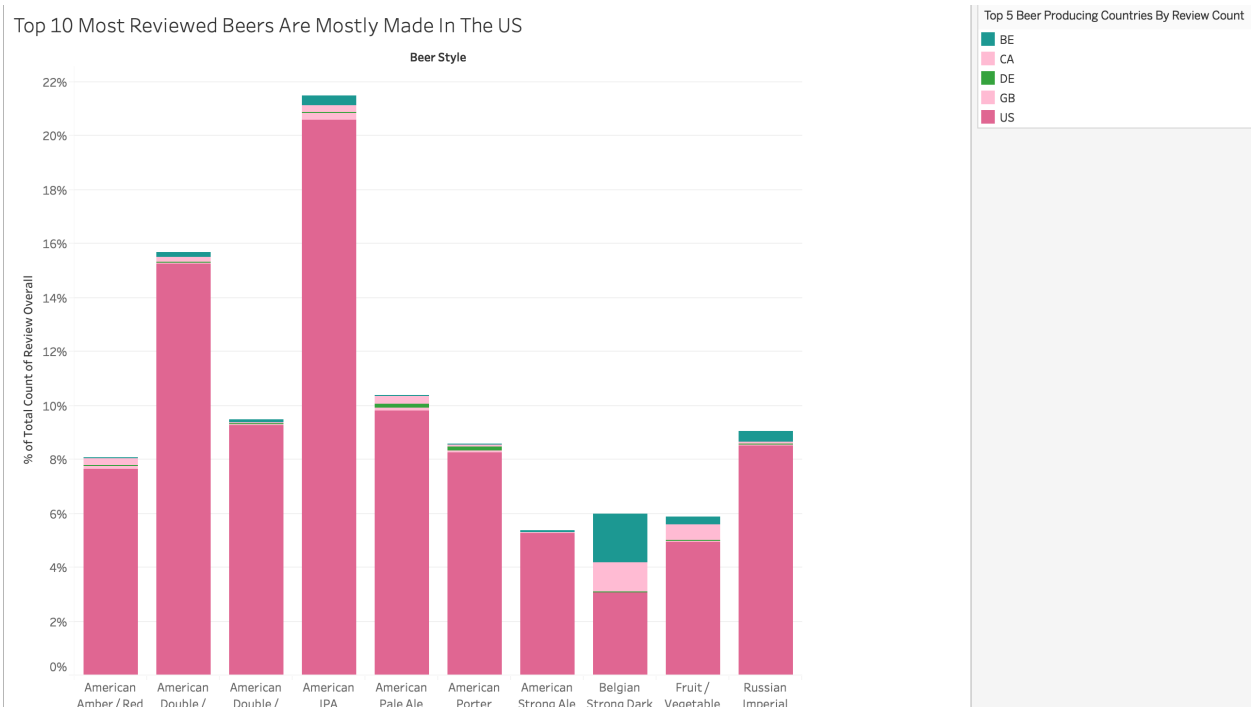
Breweries At Higher Latitudes Generate Beers With Higher Overall Reviews

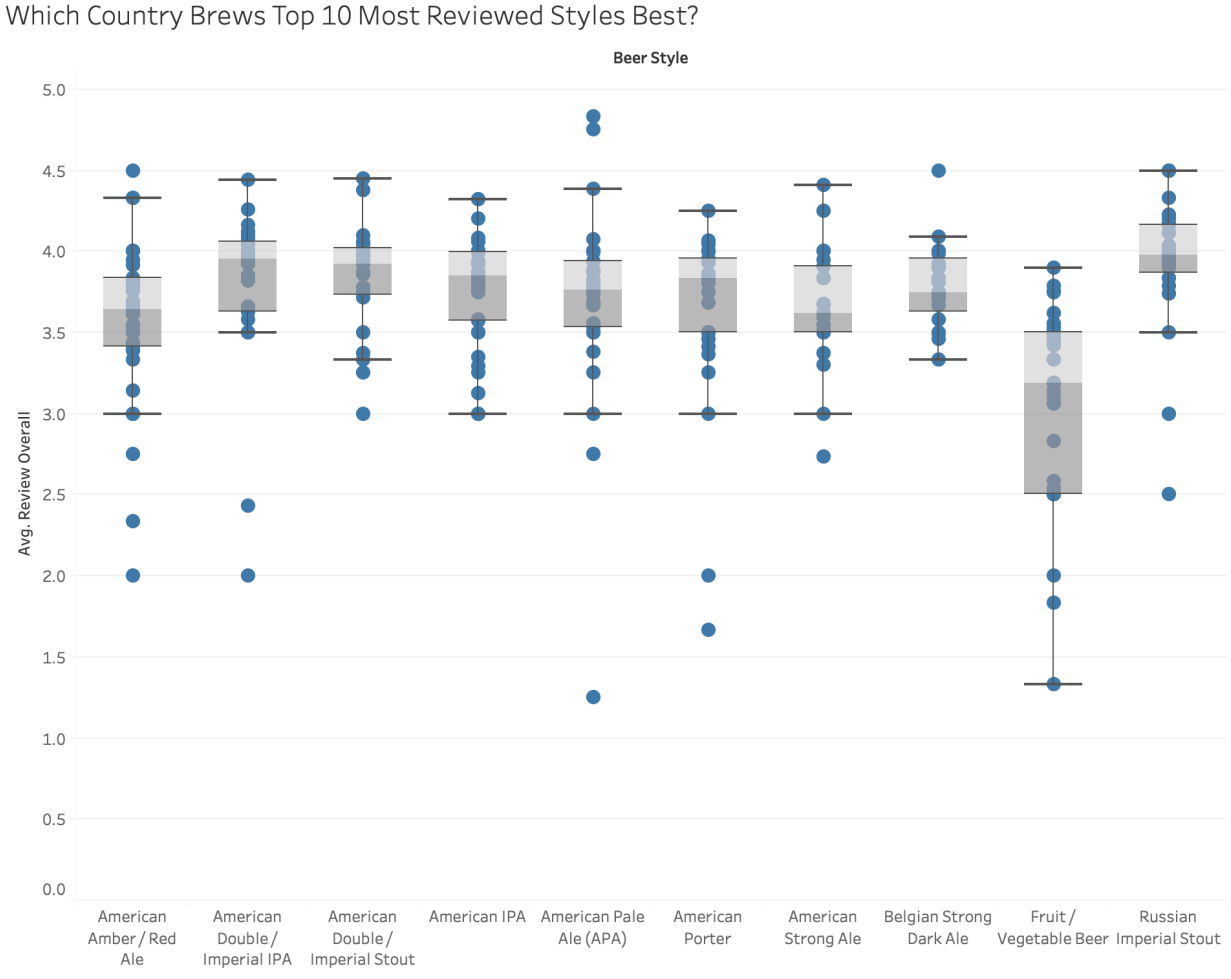
Here I used a scatterplot from matplotlib to look at the relationship between the average overall review score of a beer and the latitude of the brewery that produced it. In general, breweries with higher latitudes produced better reviewed beer.



Storyline

While the most highly reviewed beers are generally American styles that are mostly produced in America, countries outside of America tend to make better versions of these American styles.





To fully understand the point, you should check out the interactive version of the Which Country Brews Top 10 Most Reviewed Styles Best? Visualization.

Appendix

All code will also be on GitHub.

Code to get location data through Google Places API

```
import pandas as pd
import requests
import json

brewery_names = pd.read_csv('brewery_names.csv', index_col=0)
print(len(brewery_names))
# print(brewery_names.head())
remove = ['Company', '.']

api_key = my_api_key
location_data = []
counter = 0
start = counter
for brewery in brewery_names.iloc[start:start+1, 0]:
    if counter % 25 == 0:
        print(counter)

    if isinstance(brewery, str):
        place_name = brewery.replace(' ', '%20')
    else:
        location_data.append([counter, "None", "None", "None", "None", "None"])
        continue

    for item in remove:
        place_name = place_name.replace(item, '')

    maps_api_url = f'https://maps.googleapis.com/maps/api/place/autocomplete/json?
input={place_name}&key={api_key}'
    location_exists_response = requests.get(maps_api_url)
    location_json = json.loads(location_exists_response.text)

    if len(location_json['predictions']) > 0:
        place_id = location_json['predictions'][0]['place_id']
        place_api_url = f'https://maps.googleapis.com/maps/api/place/details/json?
placeid={place_id}&key={api_key}'
        location_details_response = requests.get(place_api_url)
        location_details_json = json.loads(location_details_response.text)
        if 'result' not in location_details_json.keys() or \
           'address_components' not in location_details_json['result'].keys():
            location_data.append({'index': counter,
                                  'administrative_area_level_1': "None",
                                  'administrative_area_level_2':"None",
```

```
                        'country': "None",
                        'route':"None",
                        'street_number':"None"})
            continue
        location_details = location_details_json['result']['address_components']
        location_data_details = {'index': counter,
                        'administrative_area_level_1': "None",
                        'administrative_area_level_2':"None",
                        'country': "None",
                        'route':"None",
                        'street_number':"None"}
        for component in location_details:
            if 'administrative_area_level_1' in component['types']:
                location_data_details['administrative_area_level_1'] = component['short_name']

            if 'administrative_area_level_2' in component['types']:
                location_data_details['administrative_area_level_2'] = component['short_name']

            if 'country' in component['types']:
                location_data_details['country'] = component['short_name']

            if 'route' in component['types']:
                location_data_details['route'] = component['short_name']

            if 'street_number' in component['types']:
                location_data_details['street_number'] = component['short_name']
    else:
        location_data_details = {'index': counter,
                        'administrative_area_level_1': "None",
                        'administrative_area_level_2':"None",
                        'country': "None",
                        'route':"None",
                        'street_number':"None"}
    location_data.append(location_data_details)
    counter += 1

loc_data_df = pd.DataFrame(location_data, columns=['index',
                                    'street_number',
                                    'route',
                                    'administrative_area_level_2',
                                    'administrative_area_level_1',
                                    'country'])
```

Joining brewery name information with location information

```
brewery_names = pd.read_csv('brewery_names.csv', index_col=0)
print(len(brewery_names))
print(brewery_names.head())

new_df = brewery_names.join(total_df.set_index('index'), how='inner').drop('Unnamed: 0',
axis=1)
print(len(new_df))
print(new_df.head())
```

```
total_df = df.set_index('brewery_name').join(new_df.set_index('0'), how='inner').reset_index()
print(len(total_df))
print(total_df.head())
```

After exporting Tableau latitude and longitude data from locations.

```
with open('tableau_exported_data.csv', encoding="utf16", errors='ignore') as f:
    data = f.read()

new_data = data.split('\n')
cols = [line.split('\t') for line in new_data][0]
new_df = pd.DataFrame([line.split('\t') for line in new_data[1:]], columns=cols)
print(new_df)

random_df = pd.DataFrame(total_df.groupby('country')['beer_abv']\
                    .mean().dropna()).join(new_df.set_index('Country'), how='inner')
print(random_df.head())


def float_cast(item):
    try:
        return float(item)
    except ValueError:
        return 0

random_df['Latitude'] = [float_cast(num) for num in random_df['Latitude (generated)']]
random_df['Average Overall Review By Brewery Country'] = [float_cast(num) for num in
random_df['Avg. Review Overall']]
```

To generate scatterplot.

```
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams
import numpy as np

plt.plot(np.unique(random_df['Latitude']),
        np.poly1d(np.polyfit(random_df['Latitude'],
                             random_df["Average Overall Review By Brewery Country"], 1))
(np.unique(random_df['Latitude'])))


# figure size in inches
plt.rcParams.update({'font.size': 22})
rcParams['figure.figsize'] = 20, 13.5
plt.title('Breweries At Higher Latitudes Generate Beers With Higher Overall Reviews')

plot_ = sns.scatterplot(x='Latitude', y="Average Overall Review By Brewery Country",
data=random_df)
fig = plot_.get_figure()
fig.savefig('average_overall_review_by_brewery_county.png')
```

To generate Plotly graph

```
new_destination_df = destination_df.sample(frac=1).reset_index().drop('index', axis=1)
new_destination_df.columns = ['endpoint_latitude', 'endpoint_longitude']
print(new_destination_df.head())

connections_df = pd.concat([destination_df, new_destination_df], axis=1, sort=False)
print(connections_df.head())

import plotly.plotly as py
import plotly.graph_objs as go

import pandas as pd


airports = [go.Scattergeo(
    locationmode = 'ISO-3',
    lon = random_df['Longitude'],
    lat = random_df['Latitude'],
    mode = 'markers',
    marker = go.scattergeo.Marker(
        size = 2,
        color = 'rgb(255, 0, 0)',
        line = go.scattergeo.marker.Line(
            width = 3,
            color = 'rgba(68, 68, 68, 0)'
        )
    ))]

flight_paths = []
for i in range(len(connections_df)):
    flight_paths.append(
        go.Scattergeo(
            locationmode = 'USA-states',
            lon = [connections_df['Longitude'][i], connections_df['endpoint_longitude'][i]],
            lat = [connections_df['Latitude'][i], connections_df['endpoint_latitude'][i]],
            mode = 'lines',
            line = go.scattergeo.Line(
                width = 1,
                color = 'red',
            ),
            opacity = float(df_flight_paths['cnt'][i]) / float(df_flight_paths['cnt'].max()),
        )
    )

layout = go.Layout(
    title = go.layout.Title(
        text = 'Path From Brewery to Consumer'
    ),
    showlegend = False,
    geo = go.layout.Geo(
        scope = 'world',
        projection = go.layout.geo.Projection(type = 'azimuthal equal area'),
```

```
        showland = True,
        landcolor = 'rgb(243, 243, 243)',
        countrycolor = 'rgb(204, 204, 204)',
    ),
)

fig = go.Figure(data = flight_paths + airports, layout = layout)
py.iplot(fig, filename = 'path_from_brewery_to_consumer')
```

Summary

Making beer is an artistic craft. Despite the origins of a particular style, artisans from different countries can find and make interesting takes that end up being better than the originals. Great beer is affected by but transcends geography.

GitHub link: https://github.com/robertisandor/BeerAdvocate_Beer_Reviews_Analysis