

Pay To Pee: The Impact of Accessible Bathrooms on Tourist Attraction Ratings

By: Sahithi Rampally & Megan Gao

Inspiration

We both want to visit Europe, but we've heard about the difficulty of accessing public restrooms. This inspired us to explore whether bathroom availability affects how tourists rate attractions.

What it does

Our project analyzes user ratings of various tourist attractions, comparing average ratings between attractions likely to have accessible bathrooms and those without. It shows a clear correlation between bathroom access and higher user satisfaction.

How we built it

We cleaned and prepared a large dataset of tourist attraction ratings, categorizing locations by the likelihood of having bathrooms. Using Python, we calculated the average ratings for both groups and visualized the results with clear graphs.

Challenges we ran into

A key challenge was determining which attractions likely had bathrooms, as the data didn't explicitly state this. We also had to deal with missing data and experiment with visualizations to best convey our findings.

Accomplishments that we're proud of

We're proud of transforming a simple idea into a data-backed analysis that reveals clear insights about tourist satisfaction. Our visualizations clearly highlight how bathroom availability impacts ratings.

What we learned

We learned how to clean and analyze large datasets, and how simple amenities like bathrooms can significantly influence visitor satisfaction. The project also sharpened our

skills in data visualization and storytelling with data.

What's next for Pay to Pee

We plan to expand this analysis to specific cities and regions, and potentially incorporate other amenities like parking or Wi-Fi. Ultimately, we aim to build a tool that helps tourists find well-equipped attractions.

```
In [305... from datascience import Table
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = Table.read_table('Social_Science_Dataset.csv')

# changes all the labels for readability (original columns had arbitrary names)
data.relabel('Unnamed: 1', 'Churches')
data.relabel('Unnamed: 2', 'Resorts')
data.relabel('Unnamed: 3', 'Beaches')
data.relabel('Unnamed: 4', 'Parks')
data.relabel('Unnamed: 5', 'Theatres')
data.relabel('Unnamed: 6', 'Museums')
data.relabel('Unnamed: 7', 'Malls')
data.relabel('Unnamed: 8', 'Zoos')
data.relabel('Unnamed: 9', 'Restaurants')
data.relabel('Unnamed: 10', 'Pubs/Bars')
data.relabel('Unnamed: 11', 'Local Services')
data.relabel('Unnamed: 12', 'Burger/Pizza Shops')
data.relabel('Unnamed: 13', 'Hotels/Other Lodgings')
data.relabel('Unnamed: 14', 'Juice Bars')
data.relabel('Unnamed: 15', 'Art Galleries')
data.relabel('Unnamed: 16', 'Dance Clubs')
data.relabel('Unnamed: 17', 'Swimming Pools')
data.relabel('Unnamed: 18', 'Gyms')
data.relabel('Unnamed: 19', 'Bakeries')
data.relabel('Unnamed: 20', 'Beauty & Spas')
data.relabel('Unnamed: 21', 'Cafes')
data.relabel('Unnamed: 22', 'View Points')
data.relabel('Unnamed: 23', 'Monuments')
data.relabel('Unnamed: 24', 'Gardens')

data = data.exclude(0)
data = data.drop('Unnamed: 25')
data
```

Out[305]: **google_review_ratings**
2

	Churches	Resorts	Beaches	Parks	Theatres	Museums	Malls	Zoos
User 1	0	0	3.63	3.65	5	2.92	5	2.35
User 2	0	0	3.63	3.65	5	2.92	5	2.64
User 3	0	0	3.63	3.63	5	2.92	5	2.64
User 4	0	0.5	3.63	3.63	5	2.92	5	2.35
User 5	0	0	3.63	3.63	5	2.92	5	2.64
User 6	0	0	3.63	3.63	5	2.92	5	2.63
User 7	0	5	3.63	3.63	5	2.92	3.03	2.35
User 8	0	5	3.63	3.63	5	2.92	5	2.63
User 9	0	5	3.64	3.64	5	2.92	3.03	2.62
User 10	0	5	3.64	3.64	5	2.92	5	2.35

... (5446 rows omitted)

Average per Category Data (Unsorted, Unaesthetic)

```
In [306... categories_with_bathrooms = ['Beaches', 'Parks', 'Theatres',
                             'Museums', 'Malls', 'Zoos',
                             'Restaurants', 'Hotels/Other Lodgings']
categories_without_bathrooms = ['Churches', 'Resorts', 'Pubs/Bars', 'Local Services',
                                'Burger/Pizza Shops', 'Juice Bars', 'Art Galleries',
                                'Dance Clubs', 'Swimming Pools', 'Gyms', 'Beauty & Spas', 'Cafes', 'View Points', 'Monuments', 'Gardens']

# converts table to pandas
with_bathrooms = data.select(categories_with_bathrooms).to_df().apply(pd.to_numeric)
without_bathrooms = data.drop(categories_with_bathrooms).to_df().apply(pd.to_numeric)

avg_ratings_with_bathrooms = with_bathrooms.mean(axis=0)
avg_ratings_without_bathrooms = without_bathrooms.mean(axis=0)

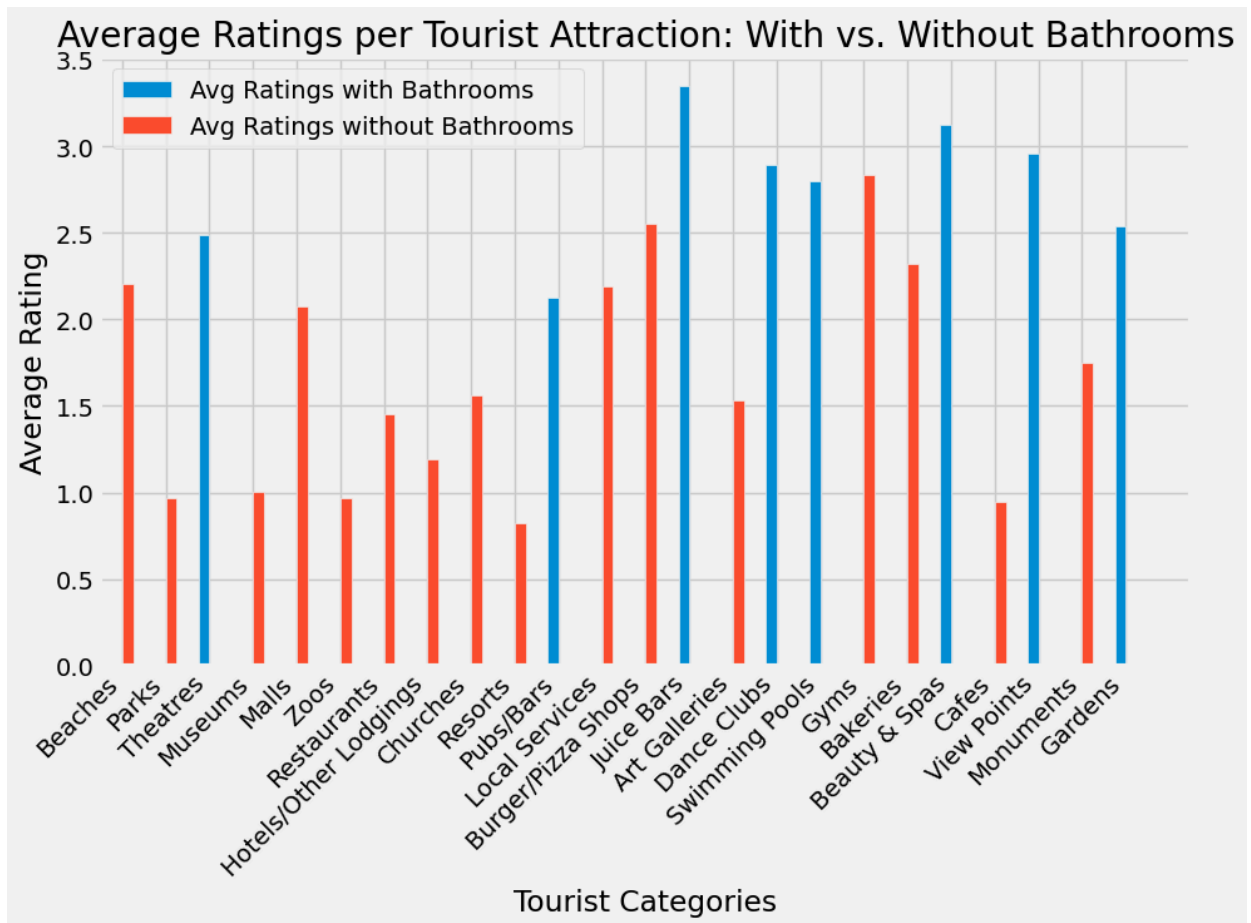
# used to plot and compare the results
comparison_df = pd.DataFrame({'Avg Ratings with Bathrooms': avg_ratings_with_bathrooms,
                              'Avg Ratings without Bathrooms': avg_ratings_without_bathrooms})

# plotting & adjusting visuals
xtick_labels = ['Beaches', 'Parks', 'Theatres', 'Museums', 'Malls', 'Zoos',
                'Restaurants', 'Hotels/Other Lodgings', 'Churches', 'Resorts',
                'Pubs/Bars', 'Local Services', 'Burger/Pizza Shops', 'Juice Bars',
                'Art Galleries', 'Dance Clubs', 'Swimming Pools', 'Gyms', 'Beauty & Spas', 'Cafes', 'View Points', 'Monuments', 'Gardens']

ax = comparison_df.plot(kind='bar', figsize=(10, 6))

ax.set_xlabel('Tourist Categories')
ax.set_ylabel('Average Rating')
ax.set_xticks(range(len(xtick_labels)))
ax.set_xticklabels(xtick_labels, rotation=45, ha='right')

plt.title('Average Ratings per Tourist Attraction: With vs. Without Bathrooms')
plt.show()
```



Average per Category Data (Sorted, Aesthetic) ★

```
In [307... # cleans & makes lengths match for comparison to work
avg_ratings_without_bathrooms = avg_ratings_without_bathrooms.drop('google_rev:

comparison_df = pd.DataFrame({'Category': categories_with_bathrooms + categorie
                              'Avg Ratings': list(avg_ratings_with_bathrooms) +
                              'Has Bathrooms': ['With Bathrooms'] * len(catego

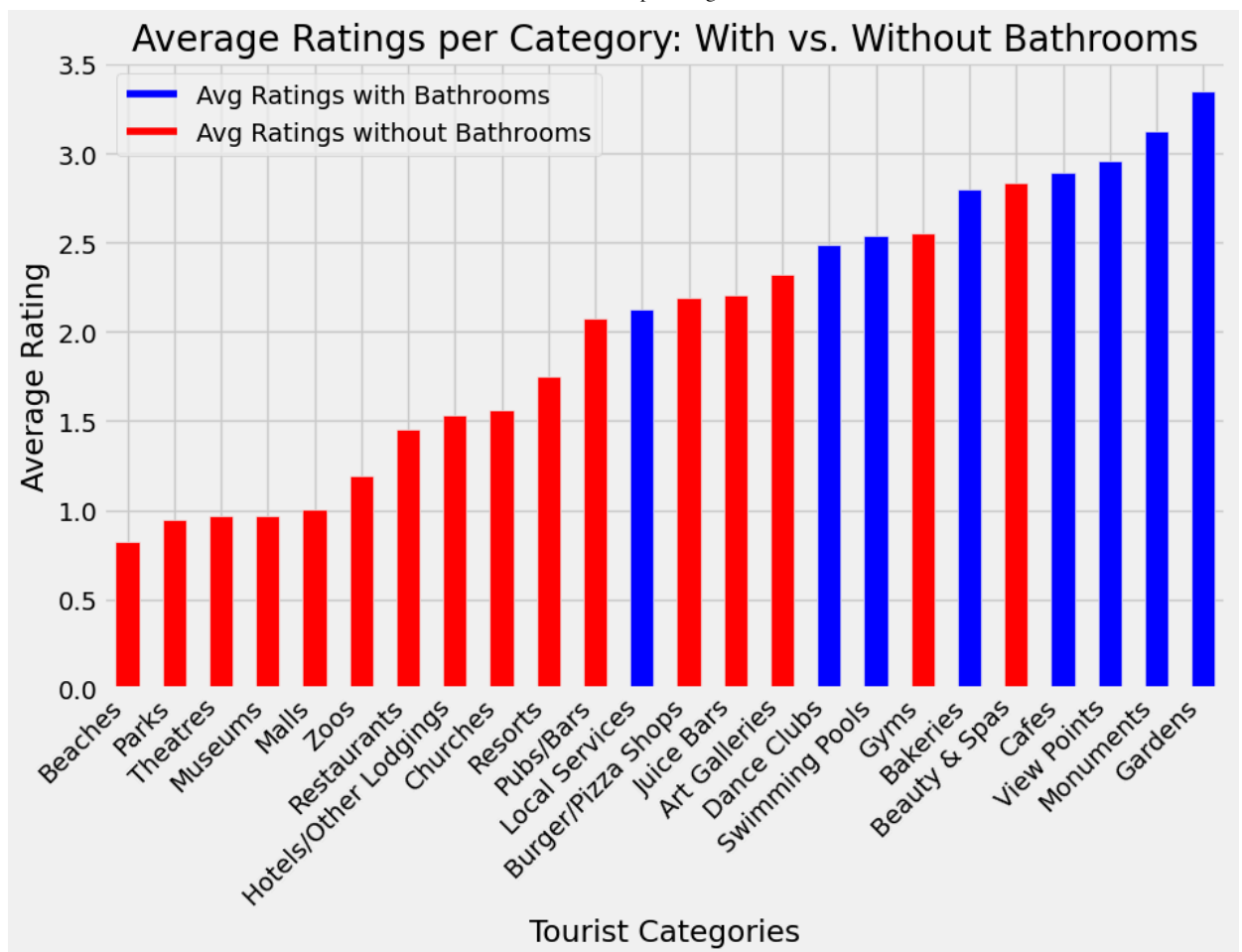
# sorts in ascending order
comparison_df_sorted = comparison_df.sort_values(by='Avg Ratings')

# plotting & adjusting visuals (but better this time)!
fig, ax = plt.subplots(figsize=(10, 6))

colors = {'With Bathrooms': 'blue', 'Without Bathrooms': 'red'}
comparison_df_sorted['Color'] = comparison_df_sorted['Has Bathrooms'].map(colo

comparison_df_sorted.plot.bar(x='Category', y='Avg Ratings', ax=ax, color=compa
new_legend = [plt.Line2D([0], [0], color='blue', lw=4), plt.Line2D([0], [0], co
ax.legend(new_legend, ['Avg Ratings with Bathrooms', 'Avg Ratings without Bath

ax.set_xlabel('Tourist Categories')
ax.set_ylabel('Average Rating')
ax.set_xticklabels(xtick_labels, rotation=45, ha='right')
plt.title('Average Ratings per Category: With vs. Without Bathrooms')
plt.show()
```



In [308...

```

avg_ratings_with_bathrooms = data_df[categories_with_bathrooms].mean(axis=1).mean()
avg_ratings_without_bathrooms = data_df[categories_without_bathrooms].mean(axis=1).mean()

print(f"Average rating for attractions with bathrooms: {avg_ratings_with_bathrooms}")
print(f"Average rating for attractions without bathrooms: {avg_ratings_without_bathrooms}")

```

Average rating for attractions with bathrooms: 2.79
 Average rating for attractions without bathrooms: 1.65