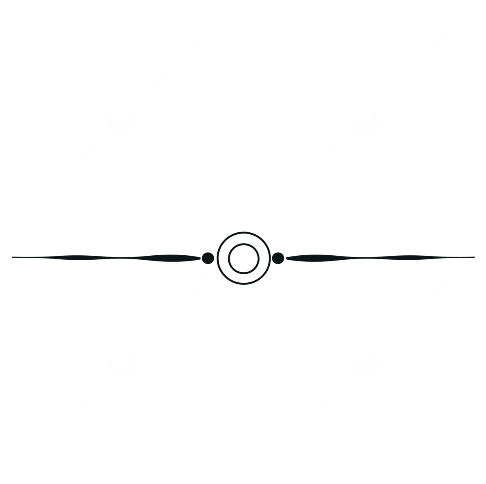
**Table of Contents**



1. **Analysis 1: Identifying the Issue - Are the Subway Ratings Improving?**
2. **Analysis 2: Competitive Analysis – Is This Performance Usual for Competitors, Or Is It Only A Subway Issue?**
3. **Analysis 3: Analyzing Market Position - Do National Chains Usually Face Lower Ratings Compared to Small/Local/Boutique Restaurants?**
4. **Analysis 4: Testing the Reliability of Reviews – Do Customers Only Post Reviews When They Are Either Very Happy or Very Angry with The Service, Not In Between?**
5. **Analysis 5: How Are the Ratings like For Local Businesses VS. National Chains?**
6. **Analysis 6: Comparing Average Ratings for Subway VS. Competitors Over Time**

## ##Analysis 1: Identifying the Issue - Are the Subway Ratings Improving?

# Take year from review date

reviews\_data['year'] = pd.to\_datetime(reviews\_data['date']).dt.year

# Group by year – > avg rating & count of reviews

yearly\_summary = reviews\_data.groupby('year').agg(

avg\_rating=('stars', 'mean'),

num\_ratings=('stars', 'count')

).reset\_index()

import matplotlib.pyplot as plt

fig, ax1 = plt.subplots(figsize=(10, 6)) #Plotting & adjusting scale

# Primary y-axis: Average rating over time

ax1.plot(yearly\_summary['year'], yearly\_summary['avg\_rating'], color='blue', label='Average Rating', marker='o')

ax1.set\_xlabel('Year')

ax1.set\_ylabel('Average Rating', color='blue')

ax1.tick\_params(axis='y', labelcolor='blue')

ax1.set\_title('Average Rating and Number of Ratings Over Time')

# Secondary y-axis: Number of ratings

ax2 = ax1.twinx()

ax2.bar(yearly\_summary['year'], yearly\_summary['num\_ratings'], color='gray', alpha=0.5, label='Number of Ratings')

ax2.set\_ylabel('Number of Ratings', color='gray')

ax2.tick\_params(axis='y', labelcolor='gray')

# Legends and grid

fig.tight\_layout()

ax1.legend(loc='upper left')

ax2.legend(loc='upper right')

plt.grid()

plt.show()

Chart, line chart

Description automatically generated

~ Summarized Findings ~

As shown in the graph, we see a negative trend in average ratings over the years, indicating no improvement in ratings.

## ##Analysis 2: Competitive Analysis – Is This Performance Usual for Competitors, Or Is It Only A Subway Issue?

# Subway and competitors using categories

subway\_data = restaurants\_data[restaurants\_data['name'].str.contains('Subway', case=False)]

competitor\_names = ['Jimmy John', 'Jersey Mike'] # Competitors

competitor\_data = restaurants\_data[restaurants\_data['name'].str.contains('|'.join(competitor\_names), case=False)]

# Review filtering

subway\_reviews = reviews\_with\_state[reviews\_with\_state['business\_id'].isin(subway\_data['business\_id'])]

competitor\_reviews = reviews\_with\_state[reviews\_with\_state['business\_id'].isin(competitor\_data['business\_id'])]

# Mean and std dev for ratings

comparison\_stats = pd.concat([

subway\_reviews.assign(brand='Subway'),

competitor\_reviews.assign(brand='Competitor')

]).groupby('brand').agg(

mean\_rating=('stars', 'mean'),

std\_rating=('stars', 'std')

).reset\_index()

# Plotting

import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(8, 6))

# Bar plot for mean

ax.bar(comparison\_stats['brand'], comparison\_stats['mean\_rating'], color='blue', alpha=0.7, label='Mean Rating')

# Error bars for standard deviation

ax.errorbar(comparison\_stats['brand'], comparison\_stats['mean\_rating'],

yerr=comparison\_stats['std\_rating'], fmt='o', color='black', label='Standard Deviation')

#Labels

ax.set\_ylabel('Rating')

ax.set\_title('Comparison of Ratings: Subway vs Competitors')

ax.legend()

plt.grid()

plt.show()

Chart, box and whisker chart

Description automatically generated

~ Summarized Findings ~

The standard deviation in ratings is similar between Subway and its competitors, but Subway has a lower mean rating (2.57 vs 3.09). Poor ratings might not necessarily mean there is an issue with sandwiches, but it is specific to Subway.

## ##Analysis 3: Analyzing Market Position - Do National Chains Usually Face Lower Ratings Compared to Small/Local/Boutique Restaurants?

# Restaurant presence in cities

city\_presence = restaurants\_data.groupby('name')['city'].nunique().reset\_index()

city\_presence.rename(columns={'city': 'city\_count'}, inplace=True)

# Conditions to determine national, local, and regional

restaurants\_data = restaurants\_data.merge(city\_presence, on='name', how='left')

restaurants\_data['chain\_category'] = restaurants\_data['city\_count'].apply(

lambda x: 'National Chain' if x > 50 else 'Local Chain' if x == 1 else 'Regional Chain'

)

# Add chain data into reviews

reviews\_with\_chains = reviews\_with\_state.merge(

restaurants\_data[['business\_id', 'chain\_category']], on='business\_id', how='left'

)

# Comparing averages

chain\_comparison = reviews\_with\_chains.groupby('chain\_category').agg(

avg\_rating=('stars', 'mean'),

std\_rating=('stars', 'std')

).reset\_index()

# Plot – Does average rating decrease as restaurant size increases?

import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(8, 6))

# Bar plot

ax.bar(chain\_comparison['chain\_category'], chain\_comparison['avg\_rating'], color='blue', alpha=0.7, label='Mean Rating')

# Error bars

ax.errorbar(chain\_comparison['chain\_category'], chain\_comparison['avg\_rating'],

yerr=chain\_comparison['std\_rating'], fmt='o', color='black', label='Standard Deviation')

#Labels

ax.set\_ylabel('Rating')

ax.set\_title('Comparison of Ratings: National vs Local Chains')

ax.legend()

plt.grid()

Chart, box and whisker chart

Description automatically generatedplt.show()

~ Summarized Findings ~

Local chains have average rating of 3.96 and national chains have 2.46. As a result, local chains tend to get higher ratings on average. National chains have higher standard deviation, which shows the wider range of user reviews compared to local ones. Therefore, the data supports the claim that smaller restaurants get higher ratings. Average rating decreases as the restaurant size increases. In order from highest average ratings to lowest, it goes from local chains, smaller regional chains, and then national chains.

## ##Analysis 4: Testing the Reliability of Reviews – Do Customers Only Post Reviews When They Are Either Very Happy or Very Angry with The Service, Not In Between?

# Count rating freq

rating\_distribution = reviews\_data['stars'].value\_counts().sort\_index().reset\_index()

rating\_distribution.columns = ['rating', 'count']

# Plotting

import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(10, 6))

# Bar plot

ax.bar(rating\_distribution['rating'], rating\_distribution['count'], color='blue', alpha=0.7)

ax.set\_xlabel('Ratings')

ax.set\_ylabel('Number of Reviews')

ax.set\_title('Distribution of Ratings')

plt.grid()

plt.show()

# Filter for any year range

# Filter data for 2018-2021

filtered\_data = reviews\_data[reviews\_data['year'].between(2018, 2021)]

# Count ratings freq per year

yearly\_rating\_distribution = filtered\_data.groupby(['year', 'stars']).size().unstack(fill\_value=0)

# Normal dist

yearly\_rating\_percentage = yearly\_rating\_distribution.div(yearly\_rating\_distribution.sum(axis=1), axis=0) \* 100

fig, ax = plt.subplots(figsize=(12, 8))

yearly\_rating\_percentage.plot(kind='bar', stacked=True, ax=ax, colormap='viridis', alpha=0.8) #Color coded for each rating

ax.set\_xlabel('Year')

ax.set\_ylabel('Percentage of Reviews')

ax.set\_title('Yearly Distribution of Ratings (2018-2021)')

plt.legend(title='Rating')

plt.grid()

Chart, bar chart

Description automatically generatedChart, bar chart, histogram

Description automatically generatedplt.show()

~ Summarized Findings ~

Data slightly supports the statement. The bar plot shows how skewed the data is towards the extreme ratings (1 or 5 stars). This could either mean that customers tend to report only the extreme ends of their experiences, or that the service is mainly 1- or 5-star worthy.

## ##Analysis 5: How Are the Ratings like For Local Businesses VS. National Chains?

chain\_rating\_distribution = reviews\_with\_chains.groupby(['chain\_category', 'stars']).size().unstack(fill\_value=0)

chain\_rating\_percentage = chain\_rating\_distribution.div(chain\_rating\_distribution.sum(axis=1), axis=0) \* 100

#Plot

fig1, ax1 = plt.subplots(figsize=(10, 6))

chain\_rating\_percentage.T.plot(kind='bar', stacked=True, ax=ax1, colormap='coolwarm', alpha=0.8)

ax1.set\_xlabel('Rating')

ax1.set\_ylabel('Percentage of Reviews')

ax1.set\_title('Distribution of Ratings by Chain Category')

plt.legend(title='Chain Category', bbox\_to\_anchor=(1.05, 1))

plt.grid()

Chart, bar chart

Description automatically generated

~ Summarized Findings ~

Like most national chains, Subway experiences extreme ratings (primarily 1-star) compared to local chains. This comes down to demographics and the target customers for each location of a national chain restaurant. For instance, service from a local small business would be much more prioritized than that of a national chain, which is where the room for improvement comes into play.

## ##Analysis 6: Comparing Average Ratings for Subway VS. Competitors Over Time

comparison\_over\_time = pd.concat([

subway\_reviews.assign(brand='Subway'),

competitor\_reviews.assign(brand='Competitor')

]).groupby(['year', 'brand'])['stars'].mean().unstack()

# Plot

fig2, ax2 = plt.subplots(figsize=(10, 6))

comparison\_over\_time.plot(ax=ax2, marker='o')

ax2.set\_xlabel('Year')

ax2.set\_ylabel('Average Rating')

ax2.set\_title('Rating Trends Over Time: Subway vs Competitors')

plt.grid()

Chart, line chart

Description automatically generated

~ Summarized Findings ~

We see here that Subway has gotten worse reviews than its Competitors for every year except 2008. This shows that its nothing to do with the sandwich business being difficult, but rather an issue with Subway service.