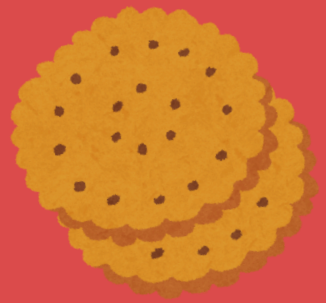
what makes the PERFECT chocolate chip cookie?

Meher Singh 24/838

OBJECTIVE:



To explore how standardised ingredient ratios, taking flour as the base ingredient, influence the ratings of a chocolate chip cookie.

By analyzing trends in highly-rated recipes, this project aims to uncover data-backed insights into what makes the “perfect” cookie.

FACTORS TO CONSIDER:

1

RICHNESS

2

SWEETNESS QUOTIENT

3

STRUCTURE

4

CHOCOLATE DENSITY

5

RISE & SPREAD

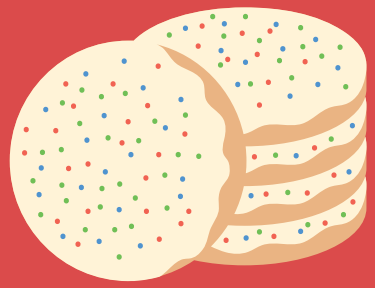
6

FLAVOUR DEPTH

ABOUT THE DATA

index	Ingredient	Text	Recipe_Index	Rating	Quantity	Unit
0	all purpose flour	3.0 cups all purpose flour	AR_1	0.920725	3.000000	cup
1	all purpose flour	2.8000000000000003 cups all purpose flour	AR_10	0.905162	2.800000	cup
2	all purpose flour	1.1076923076923078 cups all purpose flour	AR_101	0.600000	1.107692	cup
3	all purpose flour	3.333333333333333 cups sifted all purpose flour	AR_102	0.937500	3.333333	cup

COLUMNS:



index unique identifier for each row in the dataset

Ingredient the specific ingredient mentioned

Text the full ingredient line from the recipe

Recipe_Index ID linking this ingredient to a specific recipe

Rating the rating (0.4-1.0) that users gave the full recipe

Quantity numeric amount of the ingredient

Unit measurement unit associated with the quantity

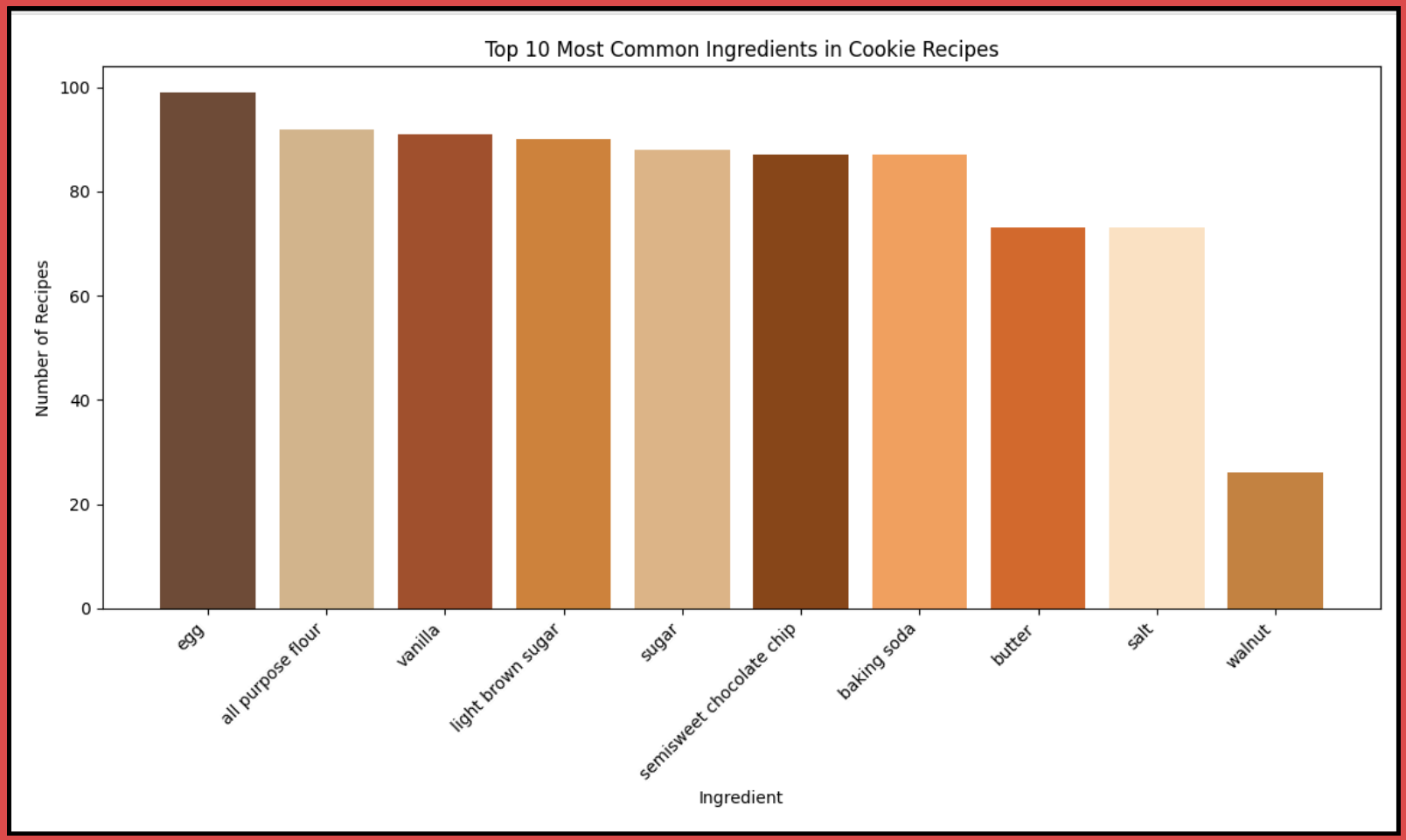
CLEANING THE DATA

Removing all:

- **duplicates**
- **missing values**
- **unnamed columns**

```
# Chocolate Chip Cookie ingredient analysis and rating insights
""" analysing the relationship between ingredient quantity and cookie recipe
ratings using a public dataset """

import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv(r'/Users/jasleenkaur/Desktop/choc_chip_cookie_ingredients.csv', index_col = 0)

# dropping unnecessary columns
df = df.drop(columns=["Unnamed: 0"])

#df = df.drop_duplicates(subset='Ingredient')

pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)

print(df.head())
# done



# checking for missing values
print(df.isnull().sum())

# checking column data types to analyse properly
print(df.dtypes)

# dropping rows with missing ratings
df = df.dropna(subset=["Rating"])

print(df.head())
# done
```

Top 10 Most Common Ingredients

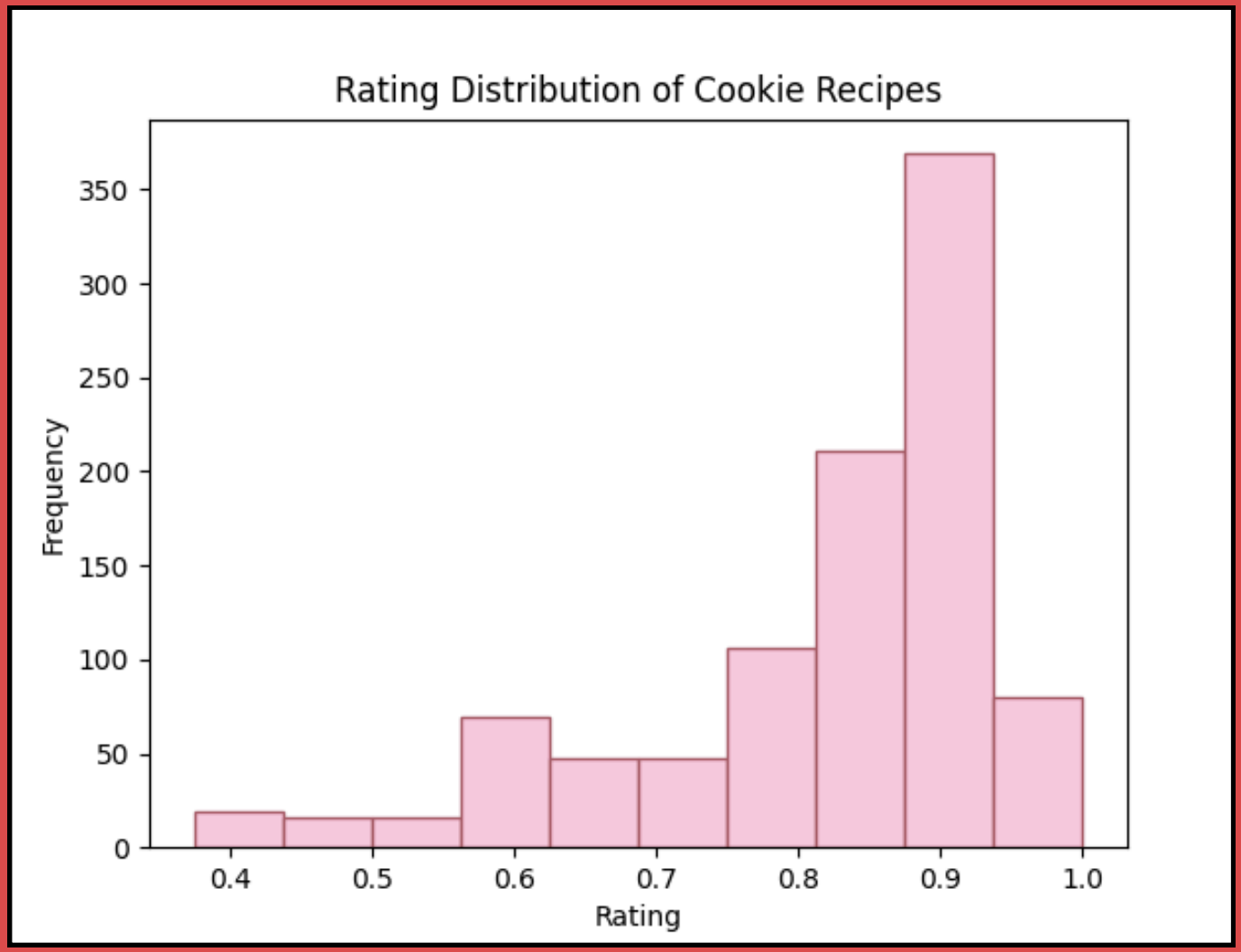


Egg, All Purpose Flour, Vanilla, Light Brown Sugar, Sugar, Semisweet Chocolate Chip, Baking Soda, Butter, Salt, Walnut

The Code

```
# getting top 10 ingredients
top_ingredients = df["Ingredient"].value_counts().head(10)

# plotting a bar chart

colors = ['#6F4E37', '#D2B48C', '#A0522D', '#CD853F', '#DEB887',
          '#8B4513', '#F4A460', '#D2691E', '#FFE4C4', '#C68642']

plt.figure(figsize=(10, 6))
plt.bar(top_ingredients.index, top_ingredients.values, color=colors)
plt.xticks(rotation=45, ha='right')
plt.xlabel("Ingredient")
plt.ylabel("Number of Recipes")
plt.title("Top 10 Most Common Ingredients in Cookie Recipes")
plt.tight_layout()
plt.show()
```

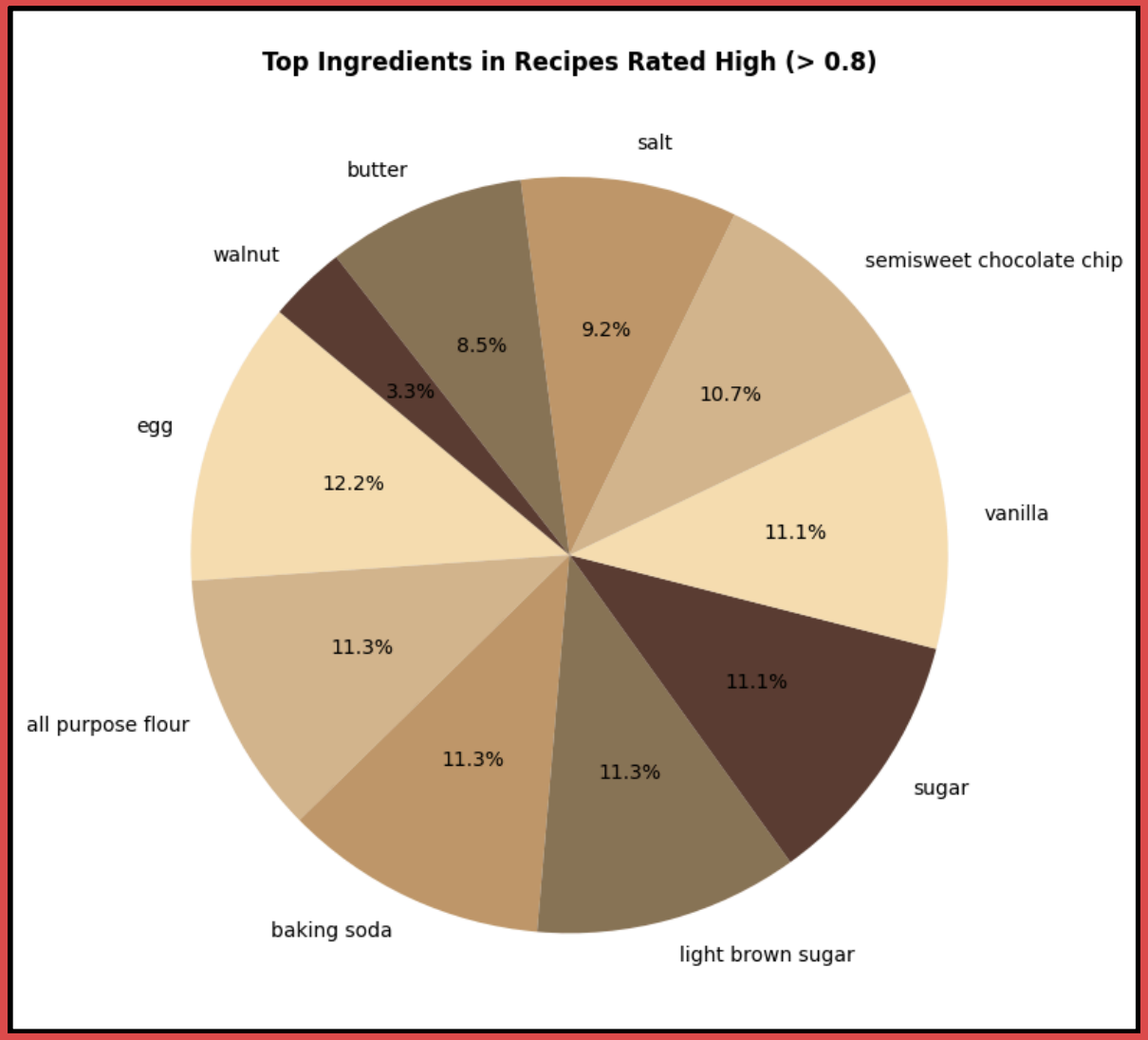
Rating Distribution of Cookie Recipes



The Code

```
# histogram to understand general distribution of recipe ratings

df['Rating'].plot(kind='hist', bins=10, color='#F8C8DC', edgecolor='#A95C68')
plt.title('Rating Distribution of Cookie Recipes')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```

Top Ingredients in Recipes Rated High (> 0.8)



The Code

```
# top ingredients in high rated recipes (>0.8)
high_rated_df = df[df['Rating'] > 0.8]
top_ingredients_high_rated = high_rated_df['Ingredient'].value_counts().head(10)

# Plotting the pie chart
colors1 = ['#F5DEB3', '#D2B48C','#C19A6B', '#8B7355', '#5C4033']
plt.figure(figsize=(8, 8))
plt.pie(
    top_ingredients_high_rated.values,
    labels=top_ingredients_high_rated.index,
    colors=colors1,
    autopct='%1.1f%%',
    startangle=140
)
plt.title('Top Ingredients in Recipes Rated High (> 0.8)', fontweight='bold')
plt.tight_layout()
plt.show()
```

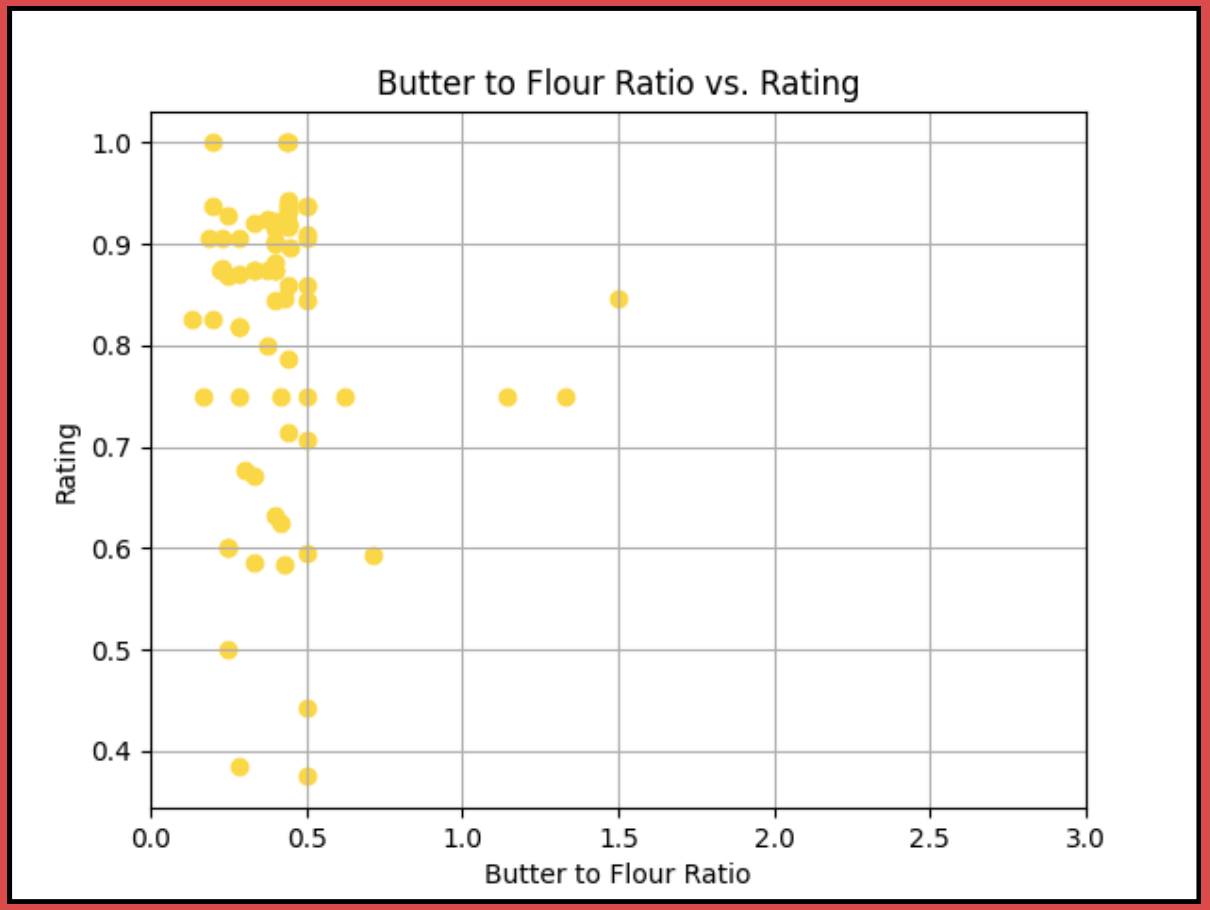
Ingredient Ratios

Richness: Butter vs Flour Ratio

Most high-rated recipes cluster between a butter-to-flour ratio of **0.4** to **0.6**.

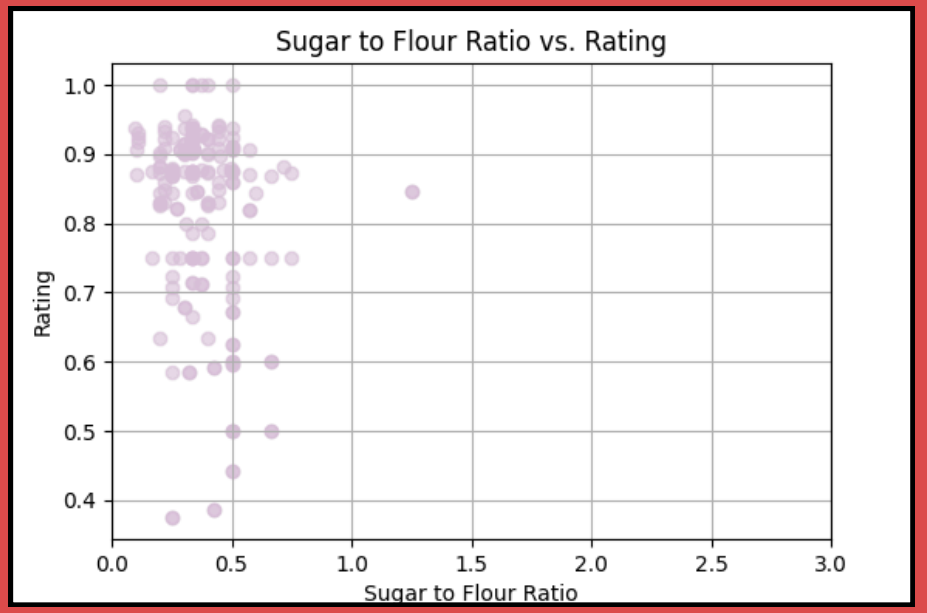
A few outliers exist above 1.0, but they don't show exceptionally high ratings.

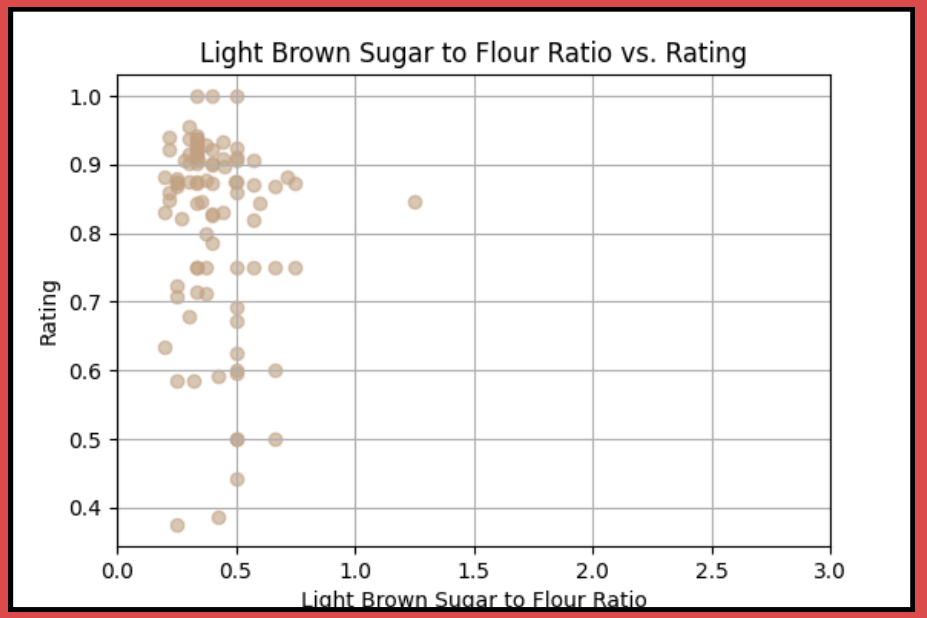
Very low ratios (under 0.3) tend to correlate with lower ratings.



2

Sweetness: Sugar vs Flour & Light Brown Sugar vs Flour





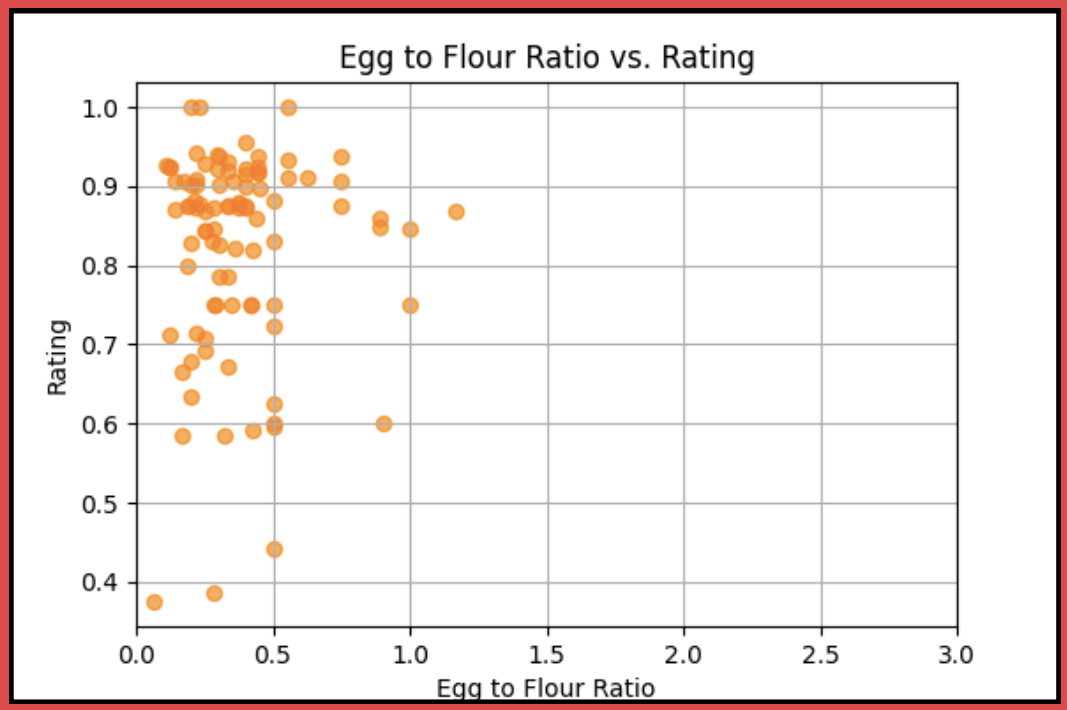
Highest density of well-rated recipes again hovers around a sugar-to-flour ratio of **0.4** to **0.6**.

A few higher sugar ratios (above 1.0) do exist but aren't consistently highly rated.

Light brown sugar seems slightly more forgiving; high-rated recipes range wider from **0.3** to **0.6**, some even up to 0.9+.

3

Structure: Egg vs Flour Ratio



Highly rated recipes mostly fall in the 0.25 to 0.5 range for egg-to-flour.

There's a drop in rating beyond 0.75 egg ratio, possibly due to cakiness.

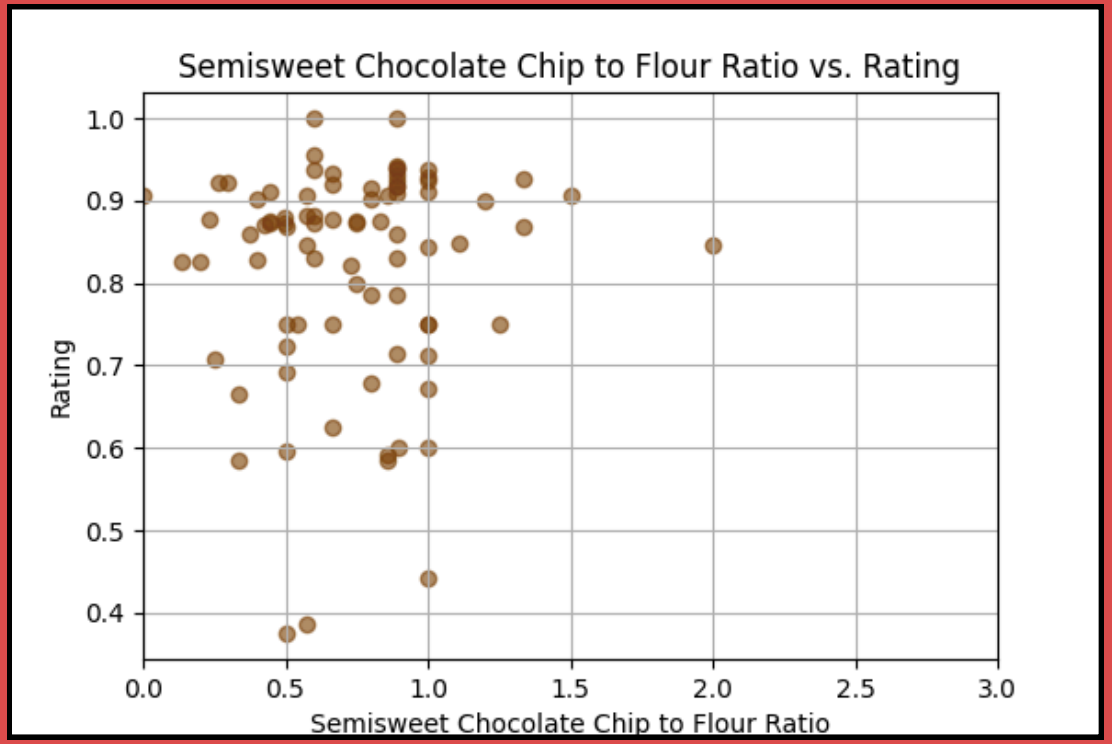
4

Chocolate Density: Semisweet Chocolate Chip vs Flour

Most recipes fall between a **0.5–1.0** chip-to-flour ratio.

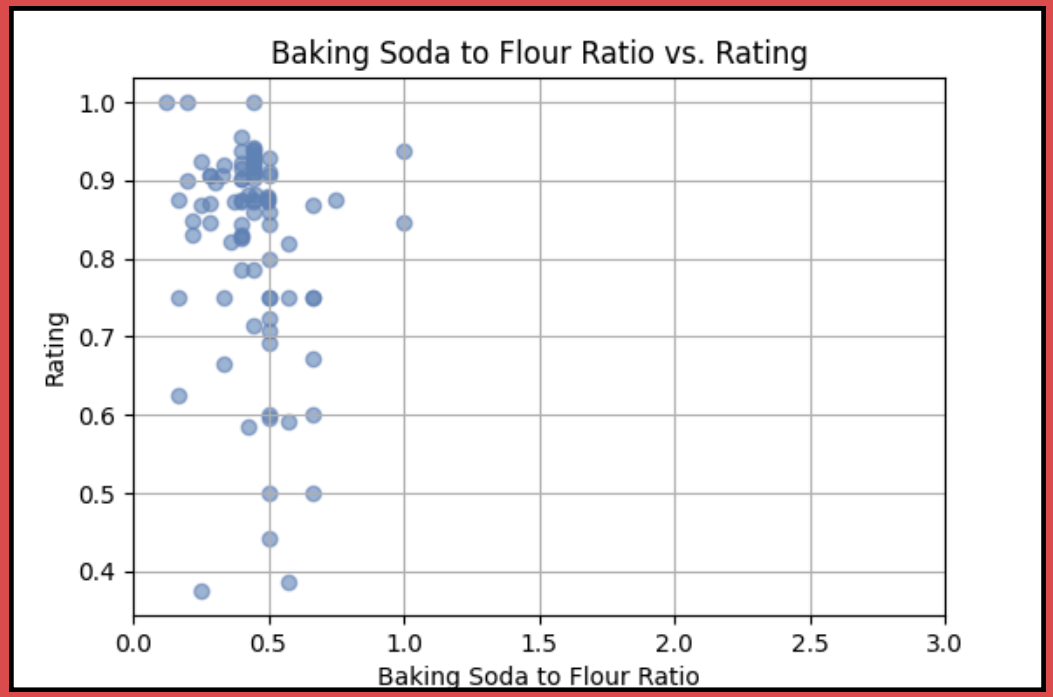
High ratings cluster heavily around **0.7–1.0**. A few outliers with very high chip ratios don't seem to perform better.

0.7–1.0 ratio = best-rated cookies



5

Rise & Spread: Baking Soda vs Flour



Concentrated around **0.3–0.5** range.

Baking soda is tricky - too little, the cookie falls flat, too much, and it tastes off.

0.35–0.45 ratio = optimal leavening and rating

6

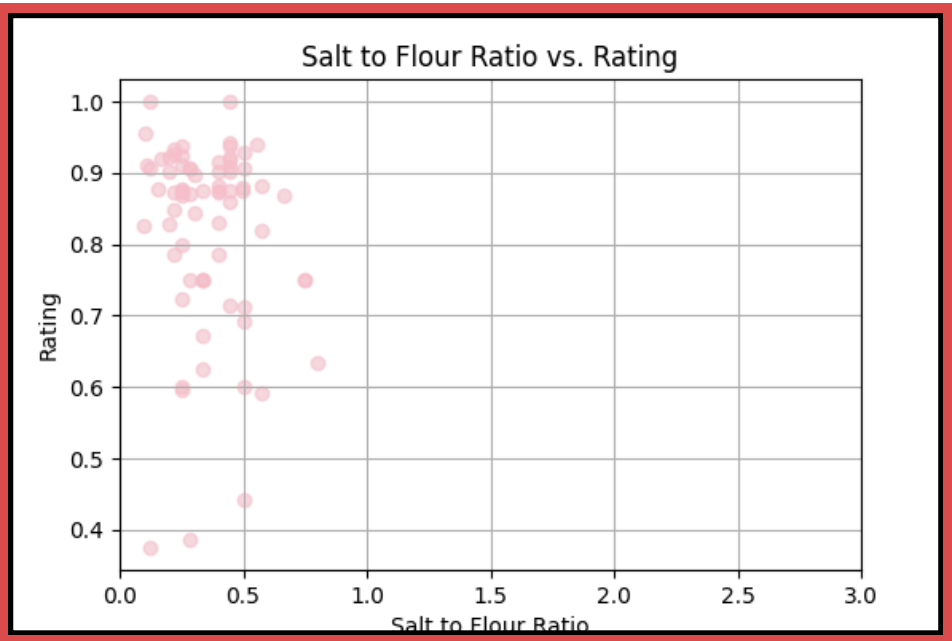
Flavour Depth: Salt vs Flour & Vanilla vs Flour

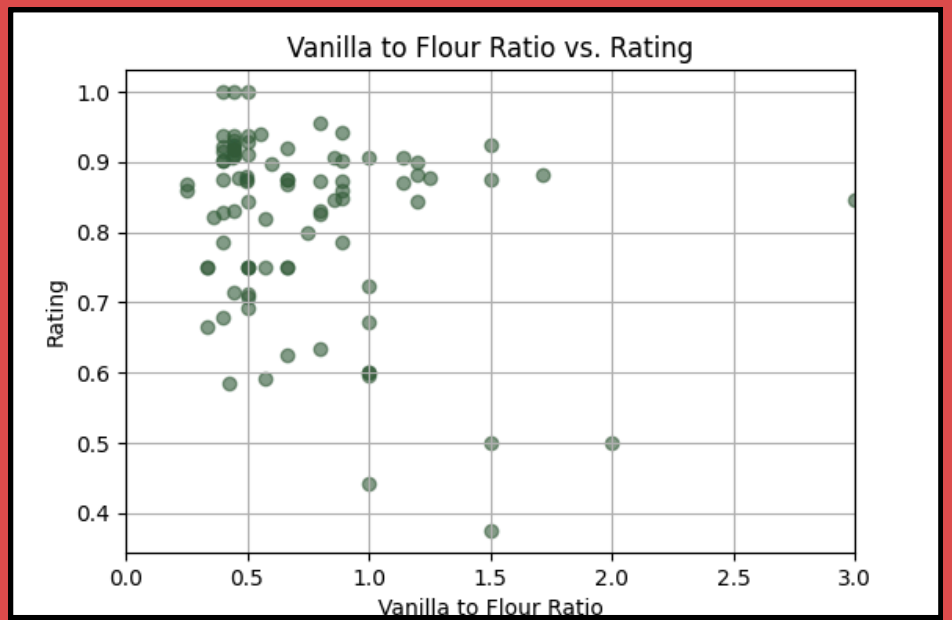
Salt - Tightly clustered around 0.1–0.4. Higher ratios are rare and often dip in ratings.

0.2–0.3 ratio = flavorful without overpowering

Vanilla - Most fall under 1.0, with high ratings between 0.3–0.8.

0.5–0.8 ratio = balance of aroma and richness





The Code

```
# Reset index if needed
df = df.reset_index()

# Convert flour to grams (1 cup = 120g)
df.loc[df['Ingredient'].str.contains('flour', case=False, na=False) &
       (df['Unit'].str.lower() == 'cups'), 'Quantity'] *= 120

# Convert ingredients to grams based on their units
df.loc[df['Ingredient'].str.contains('butter', case=False, na=False) &
       (df['Unit'].str.lower() == 'grams'), 'Quantity'] = df['Quantity']

# Convert sugar & light brown sugar (1 cup = 200g)
df.loc[df['Ingredient'].str.contains('sugar', case=False, na=False) &
       (df['Unit'].str.lower() == 'cups'), 'Quantity'] *= 200
df.loc[df['Ingredient'].str.contains('light brown sugar', case=False, na=False) &
       (df['Unit'].str.lower() == 'cups'), 'Quantity'] *= 200

# Convert semi-sweet chocolate chips (1 cup = 170g)
df.loc[df['Ingredient'].str.contains('semisweet chocolate chip', case=False, na=False) &
       (df['Unit'].str.lower() == 'cups'), 'Quantity'] *= 170

# Convert salt, baking soda, and vanilla (1 tsp = 4.9g)
df.loc[df['Ingredient'].str.contains('salt', case=False, na=False) &
       (df['Unit'].str.lower() == 'tsp'), 'Quantity'] *= 4.9
df.loc[df['Ingredient'].str.contains('baking soda', case=False, na=False) &
       (df['Unit'].str.lower() == 'tsp'), 'Quantity'] *= 4.9
df.loc[df['Ingredient'].str.contains('vanilla', case=False, na=False) &
       (df['Unit'].str.lower() == 'tsp'), 'Quantity'] *= 4.9

# Convert eggs
df.loc[df['Ingredient'].str.contains('egg', case=False, na=False), 'Quantity'] = 1  # Treat each egg as 1
```

```
# Quantity is numeric
df['Quantity'] = pd.to_numeric(df['Quantity'], errors='coerce')
df = df.dropna(subset=['Quantity'])

# Get flour rows for merging
df_flour = df[df['Ingredient'].str.contains('flour', case=False, na=False)]
df_flour = df_flour[['Recipe_Index', 'Quantity']].rename(columns={'Quantity': 'Quantity_Flour'})

# Merge with all ingredients to get flour amount per recipe
df_ingredients = df.merge(df_flour, on='Recipe_Index')

# Calculate ratio
df_ingredients['Ratio'] = df_ingredients['Quantity'] / df_ingredients['Quantity_Flour']
```

```
# Calculate ratio
df_ingredients['Ratio'] = df_ingredients['Quantity'] / df_ingredients['Quantity_Flour']

# Plotting each ingredient
ingredients_to_plot = {
    'butter': '#FFD700',
    'sugar': '#D8BFD8',
    'light brown sugar': '#C4A484',
    'vanilla': '#355E3B',
    'semisweet chocolate chip': '#7B3F00',
    'baking soda': '#6082B6',
    'salt': '#FFC0CB',
    'egg': '#F08000',
}

for ingredient, color in ingredients_to_plot.items():
    df_plot = df_ingredients[df_ingredients['Ingredient'].str.contains(ingredient, case=False, na=False)]

    if not df_plot.empty:
        plt.figure(figsize=(6, 4))
        plt.scatter(df_plot['Ratio'], df_plot['Rating'], color=color, alpha=0.6)
        plt.title(f'{ingredient.title()} to Flour Ratio vs. Rating')
        plt.xlabel(f'{ingredient.title()} to Flour Ratio')
        plt.ylabel('Rating')
        plt.grid(True)
        plt.xlim(0, 3)
        plt.show()
```

Conclusions

Butter: Too little = dry, too much = greasy — 0.5 is the sweet spot for richness and shape.

Sugar: Impacts browning, flavor, and spread — balance is key for chew and caramelization.

Eggs: Add moisture and structure — too many = cakey, too few = crumbly.

Chocolate: Semisweet offers balance — enough for flavor, but excess affects texture.

Baking Soda: Essential for rise and spread — must be used moderately to avoid overpowering.

Salt & Vanilla: Enhance depth — subtle use lifts flavor without overwhelming.

The Recipe Framework

For 1 cup (120g) of all purpose flour

113g butter

¾ cup of white sugar

⅔ to ¾ cup of brown sugar

1/2 an egg (1 whole egg per 2 cups flour)

1/4 tsp of baking soda

1 cup of semisweet chocolate chips

1 tsp of vanilla extract

1/2 tsp of salt

THANK YOU!