Do Higher Calories Result in Higher Ratings?

Name(s): Nic Colebank

Website Link: ncolebank12.github.io/recipe-ratings/

Code

In [ ]:
```
import pandas as pd
import numpy as np
import os

import plotly.express as px
pd.options.plotting.backend = 'plotly'
```

In [ ]:
```
recipes_raw = pd.read_csv('data/RAW_recipes.csv')
interactions_raw = pd.read_csv('data/RAW_interactions.csv')
```

In [ ]:
```
interactions_raw
```

Out[ ]:

	user_id	recipe_id	date	rating	review
0	1293707	40893	2011-12-21	5	So simple, so delicious! Great for chilly fall...
1	126440	85009	2010-02-27	5	I made the Mexican topping and took it to bunk...
2	57222	85009	2011-10-01	5	Made the cheddar bacon topping, adding a sprin...
3	124416	120345	2011-08-06	0	Just an observation, so I will not rate. I fo...
4	2000192946	120345	2015-05-10	2	This recipe was OVERLY too sweet. I would sta...
...	...	...	...	...	...
731922	2002357020	82303	2018-12-05	5	Delicious quick thick chocolate sauce with ing...
731923	583662	386618	2009-09-29	5	These were so delicious! My husband and I tru...
731924	157126	78003	2008-06-23	5	WOW! Sometimes I don't take the time to rate ...
731925	53932	78003	2009-01-11	4	Very good! I used regular port as well. The ...
731926	2001868099	78003	2017-12-18	5	I am so glad I googled and found this here. Th...

731927 rows × 5 columns

In [ ]:
```
recipes_raw
```

```
Out[ ]:
```

	name	id	minutes	contributor_id	submitted	tags	nutrition	n_ste
0	1 brownies in the world best ever	333281	40	985201	2008-10-27	['60-minutes-or-less', 'time-to-make', 'course...	[138.4, 10.0, 50.0, 3.0, 3.0, 19.0, 6.0]	
1	1 in canada chocolate chip cookies	453467	45	1848091	2011-04-11	['60-minutes-or-less', 'time-to-make', 'cuisin...	[595.1, 46.0, 211.0, 22.0, 13.0, 51.0, 26.0]	
2	412 broccoli casserole	306168	40	50969	2008-05-30	['60-minutes-or-less', 'time-to-make', 'course...	[194.8, 20.0, 6.0, 32.0, 22.0, 36.0, 3.0]	
3	millionaire pound cake	286009	120	461724	2008-02-12	['time-to-make', 'course', 'cuisine', 'prepara...	[878.3, 63.0, 326.0, 13.0, 20.0, 123.0, 39.0]	
4	2000 meatloaf	475785	90	2202916	2012-03-06	['time-to-make', 'course', 'main-ingredient', ...	[267.0, 30.0, 12.0, 12.0, 29.0, 48.0, 2.0]	
...	...	...	...	...	...	...	...	
83777	zydeco soup	486161	60	227978	2012-08-29	['ham', '60-minutes-or-less', 'time-to-make', ...	[415.2, 26.0, 34.0, 26.0, 44.0, 21.0, 15.0]	
83778	zydeco spice mix	493372	5	1500678	2013-01-09	['15-minutes-or-less', 'time-to-make', 'course...	[14.8, 0.0, 2.0, 58.0, 1.0, 0.0, 1.0]	
83779	zydeco ya ya deviled	308080	40	37779	2008-06-07	['60-minutes-	[59.2, 6.0, 2.0, 3.0,	

	name	id	minutes	contributor_id	submitted	tags	nutrition	n_ste
	eggs					or-less', 'time-to-make', 'course...	6.0, 5.0, 0.0]	
83780	cookies by design cookies on a stick	298512	29	506822	2008-04-15	['30-minutes-or-less', 'time-to-make', 'course...	[188.0, 11.0, 57.0, 11.0, 7.0, 21.0, 9.0]	
83781	cookies by design sugar shortbread cookies	298509	20	506822	2008-04-15	['30-minutes-or-less', 'time-to-make', 'course...	[174.9, 14.0, 33.0, 4.0, 4.0, 11.0, 6.0]	

In [ ]:
```
#merging datasets, creating avg_rating column, replacing 0 ratings with NaN
merged = recipes_raw.merge(interactions_raw, how='left', left_on='id', right_on='re
merged['rating'] = merged['rating'].replace(0, np.nan)
avg_rating = merged.groupby('recipe_id')['rating'].mean()
recipes = merged.merge(avg_rating, left_on='id', right_index=True, how='left').rena
```

Possible Questions: -Do less healthy recipes tend to have more engagement? -Do longer recipes have lower ratings? -What kinds of ingredients are most common in the highest rated recipes?

Research Question: Do recipes with an average rating of 3+ tend to have more calories than recipes that do not?

Cleaning and EDA

In [ ]:
```
recipes['calories'] = recipes['nutrition'].apply(lambda x: pd.to_numeric(x[1:-1].sp
recipes = recipes[recipes['calories'] <= 5000].copy()  #gets rid of recipes over 50
recipes['ratings_missing'] = recipes['rating_x'].isna() #column for shuffling in mi
recipes = recipes.drop(columns=['steps', 'description', 'tags', 'ingredients', 'rev
unique_recipes = recipes.drop_duplicates(subset=['id']) #creates additional datafra
```

In [ ]:
```
#categorizes a given rating into one of 2 groups, or NaN
def categorize_ratings(rating):
    if rating < 3:
        return '1.0-2.9'
    if rating <= 5:
        return '3.0-5.0'
    else:
        return np.NAN
```

```
In [ ]: #creates categorical column from avg_rating column
        unique_recipes = unique_recipes.assign(avg_rating_categorized=unique_recipes['avg_r
```

Univariate Analysis

```
In [ ]: #summary statistics for calories column
        calories = unique_recipes['calories']
        print(calories.mean(), calories.median(), calories.std(ddof=0), calories.min(), cal

        409.5323995069588 304.6 432.64657397391903 0.0 4967.6
```

```
In [ ]: #summary statistics for avg_rating column
        avg_rating = unique_recipes['avg_rating']
        print(avg_rating.mean(), avg_rating.median(), avg_rating.std(ddof=0), avg_rating.mi

        4.625376837549059 5.0 0.6405490727866491 1.0 5.0
```

```
In [ ]: #histogram of calories
        calories_hist = px.histogram(unique_recipes, x='calories', title='Distribution of C
        calories_hist.show()
```

```
In [ ]: #histogram of ratings
        ratings_hist = px.histogram(unique_recipes, x='avg_rating', nbins=10, title='Distri
        ratings_hist.show()
```

Bivariate Analysis

```
In [ ]: scatter = px.scatter(unique_recipes, x='calories', y='avg_rating', title='Relation
        scatter
```

```
In [ ]: boxplot = px.box(unique_recipes, x='avg_rating_categorized', y='calories', title='B
        boxplot.update_xaxes(categoryorder='array', categoryarray=['1.0-2.9', '3.0-5.0'])
        boxplot.show()
```

Interesting Aggregates

```
In [ ]: by_rating_mean = unique_recipes.groupby('avg_rating_categorized').mean() #gets mean
        by_rating_mean
```

Out[ ]:

	id	minutes	contributor_id	n_steps	n_ingredients
avg_rating_categorized					
1.0-2.9	389378.85633	96.989331	2.914267e+07	10.530583	9.113087
3.0-5.0	380569.88617	111.441897	1.282545e+07	10.036108	9.203806

```
In [ ]: by_rating_count = unique_recipes.groupby('avg_rating_categorized').count() #gets co
        by_rating_count
```

Out[ ]:

	name	id	minutes	contributor_id	submitted	nutrition	n_step
avg_rating_categorized							
1.0-2.9	1406	1406	1406	1406	1406	1406	140
3.0-5.0	79566	79566	79566	79566	79566	79566	7956

Assessment of Missingness

Column to assess missingness: rating_x

In [ ]:
```
def diff_of_means(df, col):
    return df.groupby('ratings_missing')[col].mean().diff().abs().iloc[-1] #calcula
```

In [ ]:
```
observed_min = diff_of_means(recipes, 'minutes') #getting observed difference betwe
observed_min
```

Out[ ]:
```
51.68377697446657
```

In [ ]:
```
observed_calories = diff_of_means(unique_recipes, 'calories') #getting observed dif
observed_calories
```

Out[ ]:
```
52.05524092086563
```

In [ ]:
```
def determine_missingness(col, df): #runs a permutation test for a given column, re
    n_repetitions = 1000
    diffs = []
    for _ in range(n_repetitions):
        shuffled = df.assign(ratings_missing=np.random.permutation(df['ratings_miss
        diffs.append(diff_of_means(shuffled, col))
    return diffs
```

In [ ]:
```
minutes_diffs = determine_missingness('minutes', recipes) #running permutation test
```

In [ ]:
```
calories_diffs = determine_missingness('calories', unique_recipes) #running permuta
```

In [ ]:
```
(minutes_diffs >= observed_min).mean() #p-value for minutes missingness test, p-val
```

Out[ ]:
```
0.106
```

In [ ]:
```
(calories_diffs >= observed_calories).mean() #p-value for calories missingness test
```

Out[ ]:
```
0.0
```

In [ ]:
```
diffs_plot = px.histogram(pd.DataFrame(minutes_diffs), x=0, nbins=50, histnorm='pro
diffs_plot.add_vline(x=observed_min, line_color='red') #adding line for the observe
diffs_plot.add_annotation(text=f'<span style="color:red">Observed Abs Diff of Means
diffs_plot.show()
```

Hypothesis Testing

Null Hypothesis: The median number of calories of recipes with an average rating of 3+ is the same as the median number of calories of all recipes

Alternative Hypothesis: The median number of calories of recipes with an average rating of 3+ is greater than the median number of calories of all recipes

In [ ]:
```
#test statistic - signed difference in median between all recipes and recipes with
by_rating_observed = unique_recipes.groupby('avg_rating_categorized')['calories'].m
observed_diff = by_rating_observed['3.0-5.0'] - by_rating_observed['1.0-2.9']
observed_diff
```

Out[ ]:
```
8.649999999999977
```

In [ ]:
```
diffs = []
for _ in range(1000): #generates difference in medians, shuffling the categories ea
    shuffled = unique_recipes.assign(avg_rating_categorized=np.random.permutation(u
    by_rating = shuffled.groupby('avg_rating_categorized')['calories'].median()
    diffs.append(by_rating['3.0-5.0'] - by_rating['1.0-2.9'])
```

In [ ]:
```
(diffs >= observed_diff).mean() #calculating p-value
```

Out[ ]:
```
0.12
```