Ananya Godse - 60009220161, Dhruvi Mehta - 60009220204, Richa Patel - 60009230202, Asma Shaikh - 60009230209

Importing Common Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
sns.set_palette("pastel")
```

In [3]:
```
df = pd.read_csv("food_choices.csv")
df
```

Out[3]:

	GPA	Gender	breakfast	calories_chicken	calories_day	calories_scone	coffee	comfort_food	comfort_food_reasor
0	2.4	2	1	430	NaN	315.0	1	none	we dont have comfc
1	3.654	1	1	610	3.0	420.0	2	chocolate, chips, ice cream	Stress, bored, ang
2	3.3	1	1	720	4.0	420.0	2	frozen yogurt, pizza, fast food	stress, sadne
3	3.2	1	1	430	3.0	420.0	2	Pizza, Mac and cheese, ice cream	Boredo
4	3.5	1	1	720	2.0	420.0	2	Ice cream, chocolate, chips	Stress, boredor cravin
...	...	...	...	...	...	...	...	...	
120	3.5	1	1	610	4.0	420.0	2	wine. mac and cheese, pizza, ice cream	boredom and sadne
121	3	1	1	265	2.0	315.0	2	Pizza / Wings / Cheesecake	Loneliness / Homesi / Sadne
122	3.882	1	1	720	NaN	420.0	1	rice, potato, seaweed soup	sadne
123	3	2	1	720	4.0	420.0	1	Mac n Cheese, Lasagna, Pizza	happiness, they a some of my favori foo
124	3.9	1	1	430	NaN	315.0	2	Chocolates, pizza, and Ritz.	hormone Premenstru syndrom

125 rows × 61 columns

In [4]:
```
df.shape
```

Out[4]:
```
(125, 61)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 125 entries, 0 to 124
Data columns (total 61 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   GPA                             123 non-null    object 
 1   Gender                          125 non-null    int64  
 2   breakfast                       125 non-null    int64  
 3   calories_chicken                125 non-null    int64  
 4   calories_day                    106 non-null    float64
 5   calories_scone                  124 non-null    float64
 6   coffee                          125 non-null    int64  
 7   comfort_food                    124 non-null    object 
 8   comfort_food_reasons            123 non-null    object 
 9   comfort_food_reasons_coded      106 non-null    float64
 10  cook                            122 non-null    float64
 11  comfort_food_reasons_coded.1    125 non-null    int64  
 12  cuisine                         108 non-null    float64
 13  diet_current                    124 non-null    object 
 14  diet_current_coded              125 non-null    int64  
 15  drink                           123 non-null    float64
 16  eating_changes                  122 non-null    object 
 17  eating_changes_coded            125 non-null    int64  
 18  eating_changes_coded1           125 non-null    int64  
 19  eating_out                      125 non-null    int64  
 20  employment                      116 non-null    float64
 21  ethnic_food                     125 non-null    int64  
 22  exercise                        112 non-null    float64
 23  father_education                124 non-null    float64
 24  father_profession               122 non-null    object 
 25  fav_cuisine                     123 non-null    object 
 26  fav_cuisine_coded               125 non-null    int64  
 27  fav_food                        123 non-null    float64
 28  food_childhood                  124 non-null    object 
 29  fries                           125 non-null    int64  
 30  fruit_day                       125 non-null    int64  
 31  grade_level                     125 non-null    int64  
 32  greek_food                      125 non-null    int64  
 33  healthy_feeling                 125 non-null    int64  
 34  healthy_meal                    124 non-null    object 
 35  ideal_diet                      124 non-null    object 
 36  ideal_diet_coded                125 non-null    int64  
 37  income                          124 non-null    float64
 38  indian_food                     125 non-null    int64  
 39  italian_food                    125 non-null    int64  
 40  life_rewarding                  124 non-null    float64
 41  marital_status                  124 non-null    float64
 42  meals_dinner_friend             122 non-null    object 
 43  mother_education                122 non-null    float64
 44  mother_profession               123 non-null    object 
 45  nutritional_check               125 non-null    int64  
 46  on_off_campus                   124 non-null    float64
 47  parents_cook                    125 non-null    int64  
 48  pay_meal_out                    125 non-null    int64  
 49  persian_food                    124 non-null    float64
 50  self_perception_weight          124 non-null    float64
 51  soup                            124 non-null    float64
 52  sports                          123 non-null    float64
 53  thai_food                       125 non-null    int64  
 54  tortilla_calories               124 non-null    float64
 55  turkey_calories                 125 non-null    int64  
 56  type_sports                     99 non-null     object 
 57  veggies_day                     125 non-null    int64  
 58  vitamins                        125 non-null    int64  
 59  waffle_calories                 125 non-null    int64  
 60  weight                          123 non-null    object 
dtypes: float64(20), int64(27), object(14)
memory usage: 59.7+ KB
```

In [6]:
```
null_values = pd.DataFrame(df.isnull().sum())
print(null_values.to_markdown())
```

	0
GPA	2
Gender	0
breakfast	0
calories_chicken	0
calories_day	19
calories_scone	1
coffee	0
comfort_food	1
comfort_food_reasons	2
comfort_food_reasons_coded	19
cook	3
comfort_food_reasons_coded.1	0
cuisine	17
diet_current	1
diet_current_coded	0
drink	2
eating_changes	3
eating_changes_coded	0
eating_changes_coded1	0
eating_out	0
employment	9
ethnic_food	0
exercise	13
father_education	1
father_profession	3
fav_cuisine	2
fav_cuisine_coded	0
fav_food	2
food_childhood	1
fries	0
fruit_day	0
grade_level	0
greek_food	0
healthy_feeling	0
healthy_meal	1
ideal_diet	1
ideal_diet_coded	0
income	1
indian_food	0
italian_food	0
life_rewarding	1
marital_status	1
meals_dinner_friend	3
mother_education	3
mother_profession	2
nutritional_check	0
on_off_campus	1
parents_cook	0
pay_meal_out	0
persian_food	1
self_perception_weight	1
soup	1
sports	2
thai_food	0
tortilla_calories	1
turkey_calories	0
type_sports	26
veggies_day	0
vitamins	0
waffle_calories	0
weight	2

In [7]:
```
percent_missing = pd.DataFrame(df.isnull().sum() * 100 / len(df))
print(percent_missing.to_markdown())
```

	0
GPA	1.6
Gender	0
breakfast	0
calories_chicken	0
calories_day	15.2
calories_scone	0.8
coffee	0
comfort_food	0.8
comfort_food_reasons	1.6
comfort_food_reasons_coded	15.2
cook	2.4
comfort_food_reasons_coded.1	0
cuisine	13.6
diet_current	0.8
diet_current_coded	0
drink	1.6
eating_changes	2.4
eating_changes_coded	0
eating_changes_coded1	0
eating_out	0
employment	7.2
ethnic_food	0
exercise	10.4
father_education	0.8
father_profession	2.4
fav_cuisine	1.6
fav_cuisine_coded	0
fav_food	1.6
food_childhood	0.8
fries	0
fruit_day	0
grade_level	0
greek_food	0
healthy_feeling	0
healthy_meal	0.8
ideal_diet	0.8
ideal_diet_coded	0
income	0.8
indian_food	0
italian_food	0
life_rewarding	0.8
marital_status	0.8
meals_dinner_friend	2.4
mother_education	2.4
mother_profession	1.6
nutritional_check	0
on_off_campus	0.8
parents_cook	0
pay_meal_out	0
persian_food	0.8
self_perception_weight	0.8
soup	0.8
sports	1.6
thai_food	0
tortilla_calories	0.8
turkey_calories	0
type_sports	20.8
veggies_day	0
vitamins	0
waffle_calories	0
weight	1.6

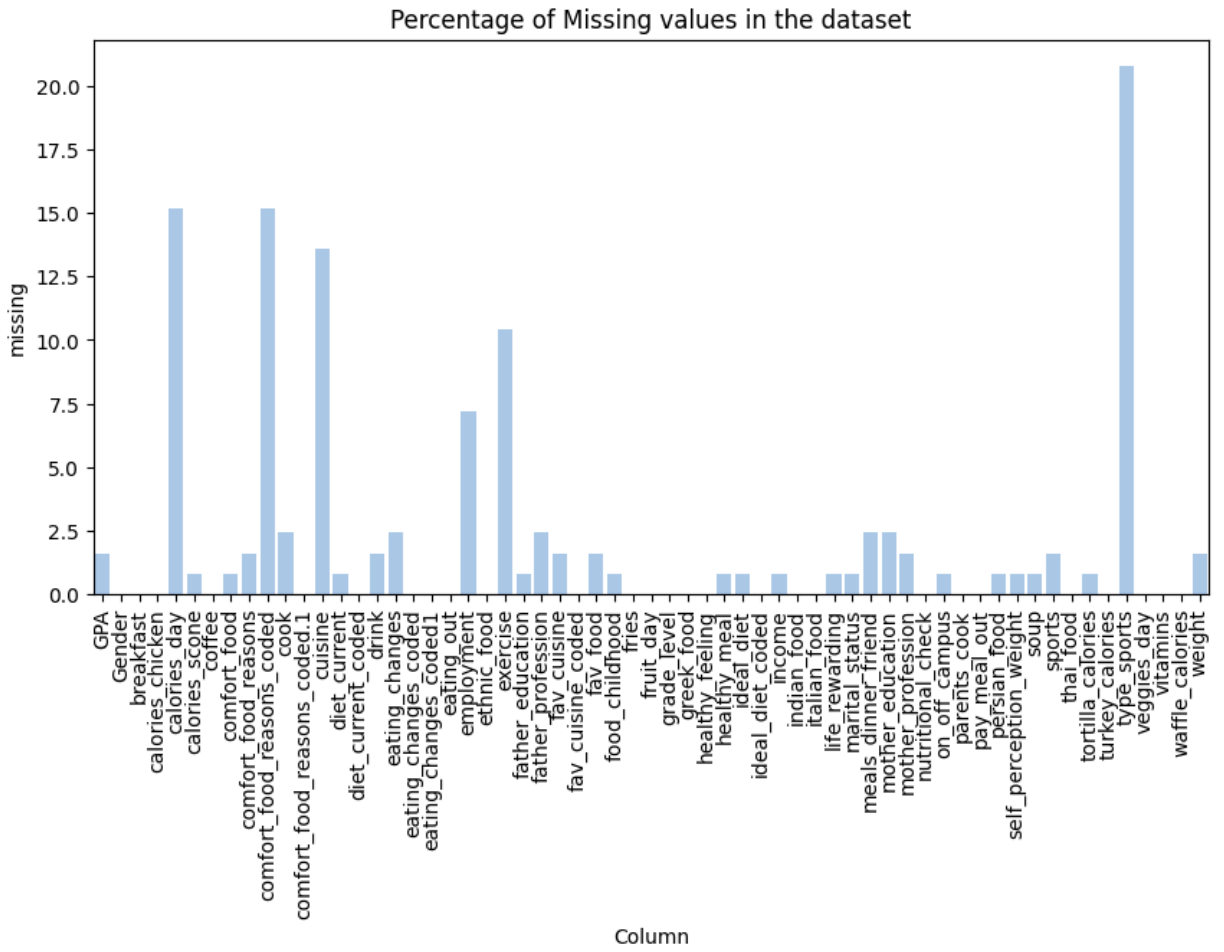
In [8]:
```
percent_missing = percent_missing.rename(columns={0:'missing'})
percent_missing
```

Out[8]:

	missing
GPA	1.6
Gender	0.0
breakfast	0.0
calories_chicken	0.0
calories_day	15.2
...	...
type_sports	20.8
veggies_day	0.0
vitamins	0.0
waffle_calories	0.0
weight	1.6

61 rows × 1 columns

In [9]:
```
plt.figure(figsize=(10, 5))
sns.barplot(data=percent_missing, x=percent_missing.index, y="missing").set(title="Percentage of Mis
plt.xticks(rotation=90)
plt.show()
```



In [10]:
```
df.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
df.describe()
```

Out[11]:

	Gender	breakfast	calories_chicken	calories_day	calories_scone	coffee	comfort_food_reasons_coded	12
count	125.000000	125.000000	125.000000	106.000000	124.000000	125.00000	106.000000	
mean	1.392000	1.112000	577.320000	3.028302	505.241935	1.75200	2.698113	
std	0.490161	0.316636	131.214156	0.639308	230.840506	0.43359	1.972042	
min	1.000000	1.000000	265.000000	2.000000	315.000000	1.00000	1.000000	
25%	1.000000	1.000000	430.000000	3.000000	420.000000	2.00000	2.000000	
50%	1.000000	1.000000	610.000000	3.000000	420.000000	2.00000	2.000000	
75%	2.000000	1.000000	720.000000	3.000000	420.000000	2.00000	3.000000	
max	2.000000	2.000000	720.000000	4.000000	980.000000	2.00000	9.000000	

8 rows × 47 columns

In [12]:
```
df['calories_day'].fillna((df['calories_day'].mean()), inplace=True)
df.isnull().sum()
```

Out[12]:
```
GPA                  2
Gender               0
breakfast            0
calories_chicken     0
calories_day         0
                   ..
type_sports         26
veggies_day          0
vitamins             0
waffle_calories      0
weight               2
Length: 61, dtype: int64
```

In [13]:
```
df['exercise'].fillna((df['exercise'].mode()[0]), inplace=True)
df['cuisine'].fillna((df['cuisine'].mode()[0]), inplace=True)
df.isnull().sum()
```

Out[13]:
```
GPA                  2
Gender               0
breakfast            0
calories_chicken     0
calories_day         0
                   ..
type_sports         26
veggies_day          0
vitamins             0
waffle_calories      0
weight               2
Length: 61, dtype: int64
```

In [14]:
```
df.drop(['comfort_food_reasons','diet_current', 'eating_changes', 'father_education', 'father_profes
df.shape
```

Out[14]:
```
(125, 47)
```

In [15]:
```
df.rename(columns={'comfort_food_reasons_coded.1': 'comfort_food_reasons_coded'}, inplace=True)
```

In [16]:
```
df.isnull().sum()
```

Out[16]:
```
GPA                           2
Gender                        0
breakfast                     0
calories_chicken              0
calories_day                  0
calories_scone                1
coffee                        0
comfort_food                  1
cook                          3
comfort_food_reasons_coded    0
cuisine                       0
diet_current_coded            0
drink                         2
eating_changes_coded          0
eating_out                    0
employment                    9
ethnic_food                   0
exercise                      0
fav_cuisine_coded             0
fav_food                      2
food_childhood                1
fries                         0
fruit_day                     0
grade_level                   0
greek_food                    0
healthy_feeling               0
ideal_diet_coded              0
income                        1
indian_food                   0
italian_food                  0
life_rewarding                1
marital_status                1
nutritional_check             0
on_off_campus                 1
parents_cook                  0
pay_meal_out                  0
persian_food                  1
self_perception_weight        1
soup                          1
sports                        2
thai_food                     0
tortilla_calories             1
turkey_calories               0
veggies_day                   0
vitamins                      0
waffle_calories               0
weight                        2
dtype: int64
```

In [17]:
```
# Convert all items to lowercase and split by comma, then stack them into individual rows
comfort_foods_series = df['comfort_food'].str.lower().str.strip().str.split(',\s*').explode()

# Count the occurrences of each item
comfort_food_item_counts = comfort_foods_series.value_counts()

# Display the most popular items
most_popular_items = pd.DataFrame(comfort_food_item_counts.head(10))  # Displaying the top 10 most p
print(most_popular_items)
```

```
               count
comfort_food
ice cream         42
pizza             37
chocolate         25
chips             22
cookies           17
mac and cheese    11
pasta              8
cake               7
popcorn            7
french fries       6
```

In [18]:
```
# Convert all items to lowercase and split by comma, then stack them into individual rows
childhood_food_series = df['food_childhood'].str.lower().str.strip().str.split(',\s*').explode()

# Count the occurrences of each item
```

```
childhood_food_item_counts = childhood_food_series.value_counts()

# Display the most popular items
most_popular_childhood_items = pd.DataFrame(childhood_food_item_counts.head(10))  # Displaying the t
print(most_popular_childhood_items)
```

```
                 count
food_childhood
pizza               31
pasta               18
spaghetti           11
chicken             11
steak               10
mac and cheese       9
chicken nuggets      6
tacos                5
lasagna              4
mashed potatoes      4
```

In [19]:
```
df['weight'] = pd.to_numeric(df['weight'], errors='coerce')
df['GPA'] = pd.to_numeric(df['GPA'], errors='coerce')
df.isnull().sum()
```

Out[19]:
```
GPA                          5
Gender                       0
breakfast                    0
calories_chicken             0
calories_day                 0
calories_scone               1
coffee                       0
comfort_food                 1
cook                         3
comfort_food_reasons_coded   0
cuisine                      0
diet_current_coded           0
drink                        2
eating_changes_coded         0
eating_out                   0
employment                   9
ethnic_food                  0
exercise                     0
fav_cuisine_coded            0
fav_food                     2
food_childhood               1
fries                        0
fruit_day                    0
grade_level                  0
greek_food                   0
healthy_feeling              0
ideal_diet_coded             0
income                       1
indian_food                  0
italian_food                 0
life_rewarding               1
marital_status               1
nutritional_check            0
on_off_campus                1
parents_cook                 0
pay_meal_out                 0
persian_food                 1
self_perception_weight       1
soup                         1
sports                       2
thai_food                    0
tortilla_calories            1
turkey_calories              0
veggies_day                  0
vitamins                     0
waffle_calories              0
weight                       5
dtype: int64
```

In [20]:
```
df['parents_cook'] = df['parents_cook'].astype(float)
```

In [21]:
```
df.dropna(inplace=True)
```

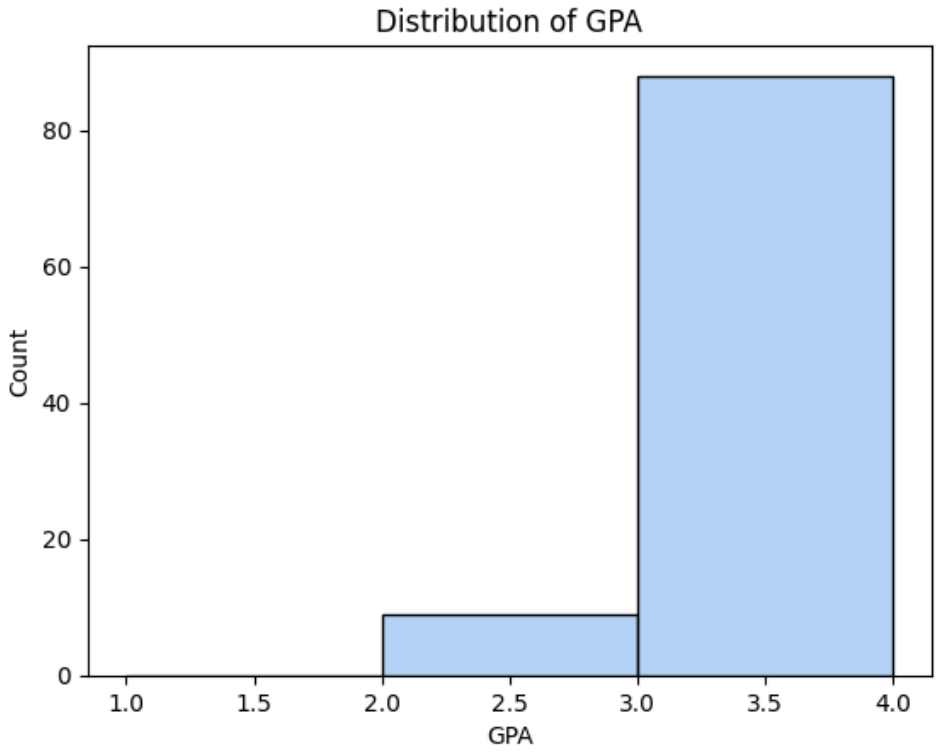
In [22]:
```
df.isnull().sum()
```

Out[22]:
```
GPA                          0
Gender                       0
breakfast                    0
calories_chicken             0
calories_day                 0
calories_scone               0
coffee                       0
comfort_food                 0
cook                         0
comfort_food_reasons_coded   0
cuisine                      0
diet_current_coded           0
drink                        0
eating_changes_coded         0
eating_out                   0
employment                   0
ethnic_food                  0
exercise                     0
fav_cuisine_coded            0
fav_food                     0
food_childhood               0
fries                        0
fruit_day                    0
grade_level                  0
greek_food                   0
healthy_feeling              0
ideal_diet_coded             0
income                       0
indian_food                  0
italian_food                 0
life_rewarding               0
marital_status               0
nutritional_check            0
on_off_campus                0
parents_cook                 0
pay_meal_out                 0
persian_food                 0
self_perception_weight       0
soup                         0
sports                       0
thai_food                    0
tortilla_calories            0
turkey_calories              0
veggies_day                  0
vitamins                     0
waffle_calories              0
weight                       0
dtype: int64
```

In [23]:
```
df.shape
```

Out[23]:
```
(97, 47)
```

In [24]:
```
sns.histplot(df, x='GPA', bins=[1, 2, 3, 4]).set(title='Distribution of GPA')
plt.show()
```



Observation:

The histogram illustrates the Distribution of Student GPAs, ranging from 1 to 4.

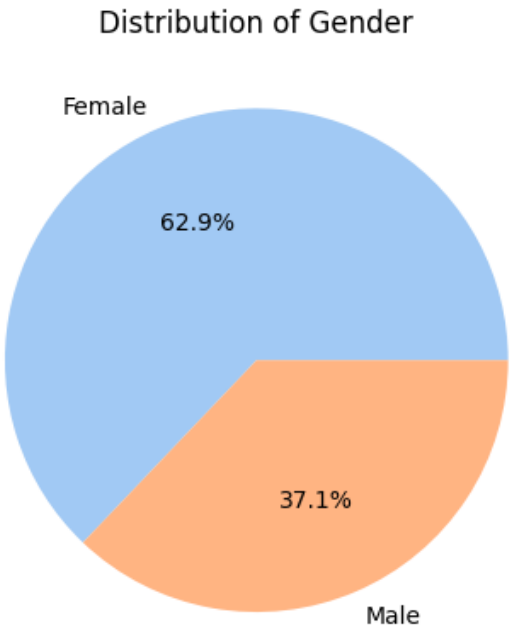
We can conclude that:

Most students fall in the range of 2 to 4, greater concerntration of students is observed in the range 3 to 4 and fewer students from 2 to 3.

No student falls in the range of 1 to 2.

In [25]:
```
gender_dist = df['Gender'].value_counts()
```

In [26]:
```
plt.pie(gender_dist, labels=['Female', 'Male'], autopct='%1.1f%%')
plt.title('Distribution of Gender')
plt.show()
```



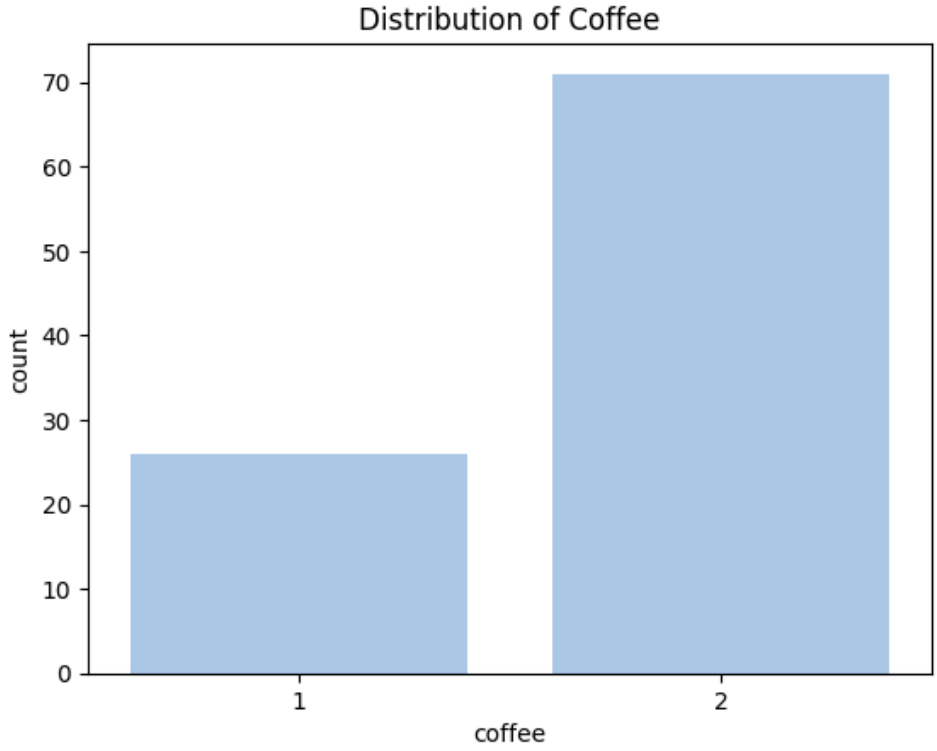
Observation:

The above pie chart illustrates the distribution of students based gender.

We can conlude that:

The percentage of female students i.e 62.9% is higher than the male students i.e 37.1%.

In [27]:
```
sns.countplot(data=df, x='coffee').set(title='Distribution of Coffee')
plt.show()
```



Observation:

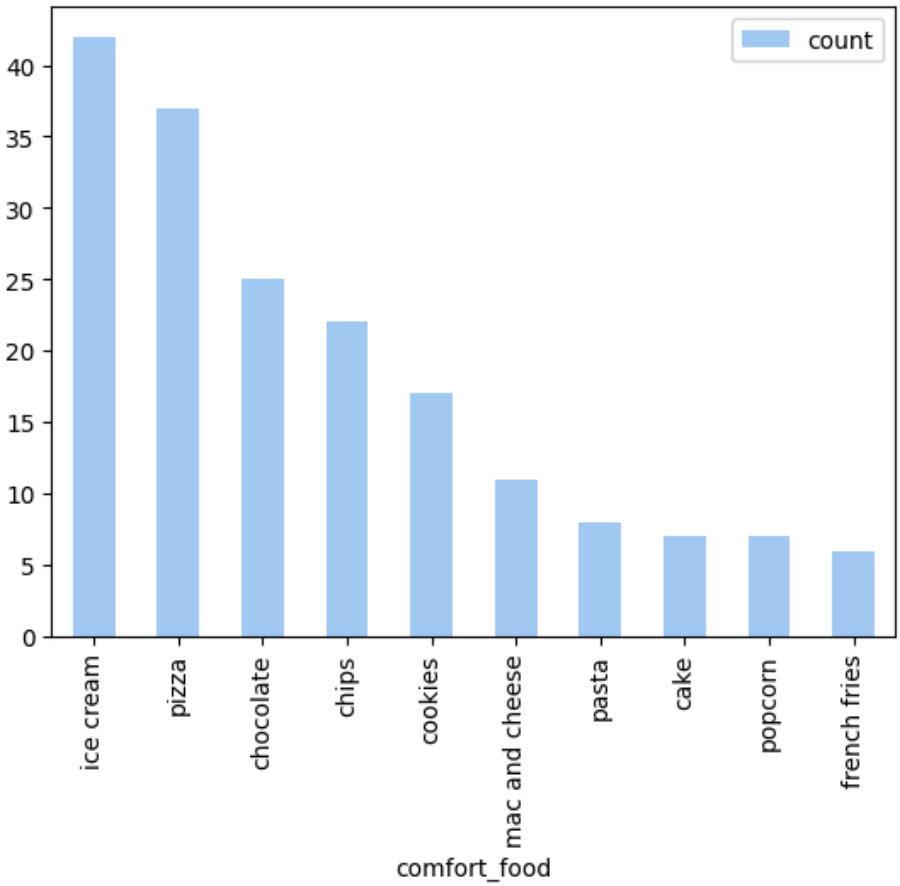
The above countplot illustrates the Distribution of Coffee consumed by the students.

We can conclude that:

Consumption of Expresso is much higher than Frapuccino.

Students prefer drinking expresso over frapaccuino.

In [28]:
```
most_popular_items.plot(kind='bar')
plt.show()
```



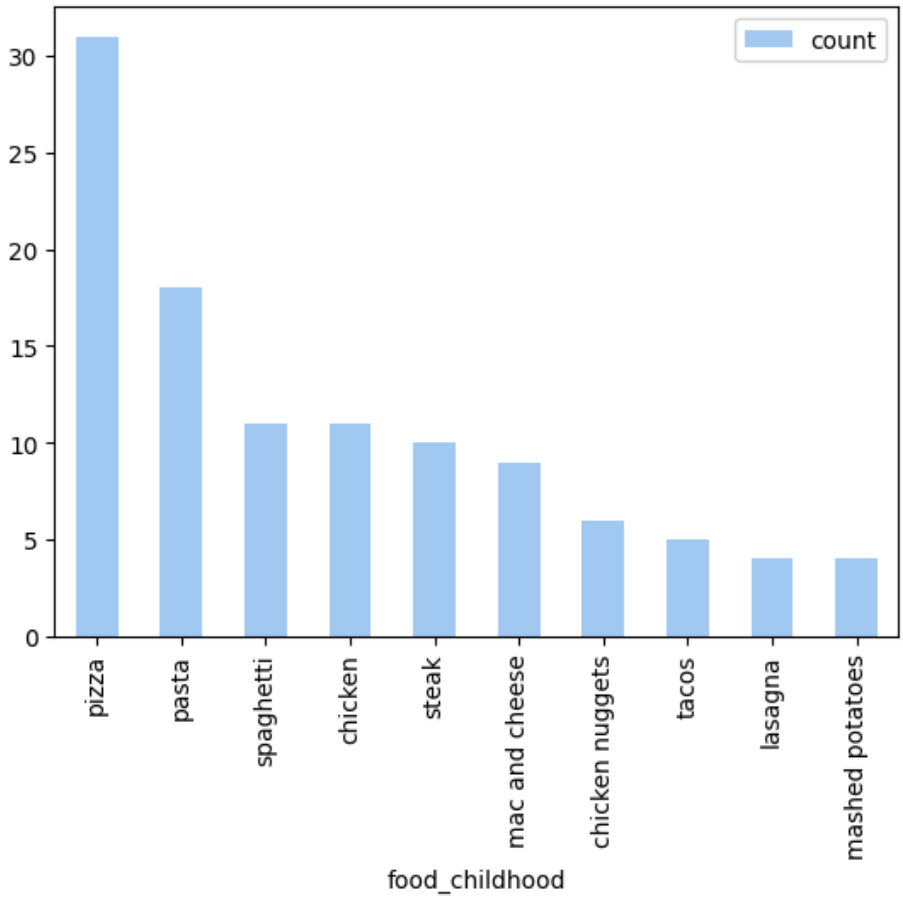
Obsevation:

The above bar graph illustrates the most popular items amongst the students.

We can conclude that:

Students of fond of fast food items, ice cream consumption being the most followed by pizza, chocolate chips etc

In [29]:
```
most_popular_childhood_items.plot(kind='bar')
plt.show()
```



Obsevation:

The above bar graph illustrates the most popular items amongst the students during their childhood.

We can conclude that:

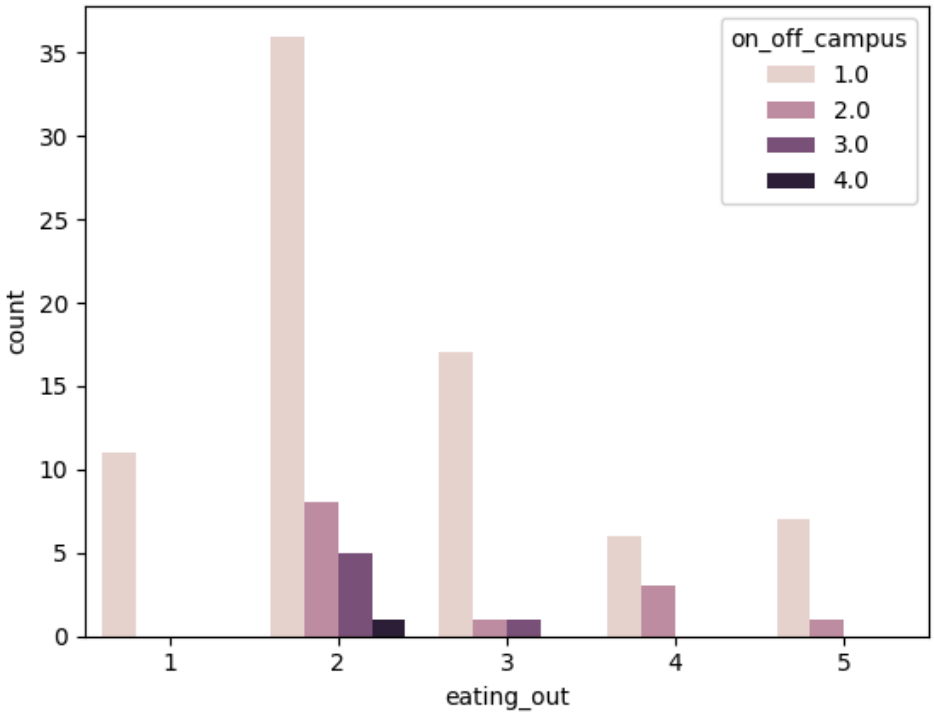
Students of fond of fast food items, pizza consumption being the most followed by pasta, spaghetti etc

From the above two graphs we can conclude that the students liking hasn't changed much. They like eating Fast Food

In [30]:
```
df.columns
```

Out[30]:
```
Index(['GPA', 'Gender', 'breakfast', 'calories_chicken', 'calories_day',
       'calories_scone', 'coffee', 'comfort_food', 'cook',
       'comfort_food_reasons_coded', 'cuisine', 'diet_current_coded', 'drink',
       'eating_changes_coded', 'eating_out', 'employment', 'ethnic_food',
       'exercise', 'fav_cuisine_coded', 'fav_food', 'food_childhood', 'fries',
       'fruit_day', 'grade_level', 'greek_food', 'healthy_feeling',
       'ideal_diet_coded', 'income', 'indian_food', 'italian_food',
       'life_rewarding', 'marital_status', 'nutritional_check',
       'on_off_campus', 'parents_cook', 'pay_meal_out', 'persian_food',
       'self_perception_weight', 'soup', 'sports', 'thai_food',
       'tortilla_calories', 'turkey_calories', 'veggies_day', 'vitamins',
       'waffle_calories', 'weight'],
      dtype='object')
```

In [31]:
```
sns.countplot(data=df, x='eating_out', hue='on_off_campus')
plt.show()
```



Observation:

The above countplot represents the relationship between living situation of the student and how frequently the student eats out.

We can conclude that:

Eating out

1 --> Never eats out

2 --> 1-2 a week

3 --> 2-3 a week

4 --> 3-5 a week

5 --> Everyday eats out

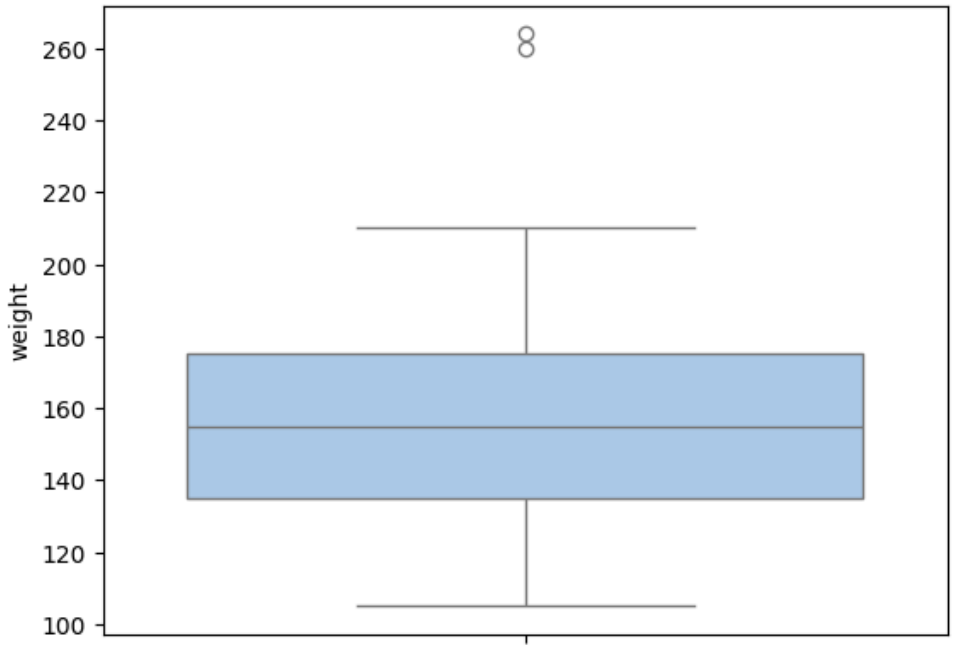
Mostly all students espeacially students living on campus eat out 2-3 a week .

Students living with their family or have their own house prefer eating home cooked food they eat out very less.

In [32]:
```
df['weight'].describe()
```

Out[32]:
```
count     97.000000
mean     157.010309
std       29.807890
min      105.000000
25%      135.000000
50%      155.000000
75%      175.000000
max      264.000000
Name: weight, dtype: float64
```

In [33]:
```
sns.boxplot(data=df, y='weight')
plt.show()
```



Observation:

The above box plot illustrates the weight distribution of the students.

We can conclude that:

The mean weight of the students lies around at 160.

The first quartile of weight of the students lies around at 140.

The third quartile of weight of the students lies around at 180.

The range of the weight is from 100 to 220.

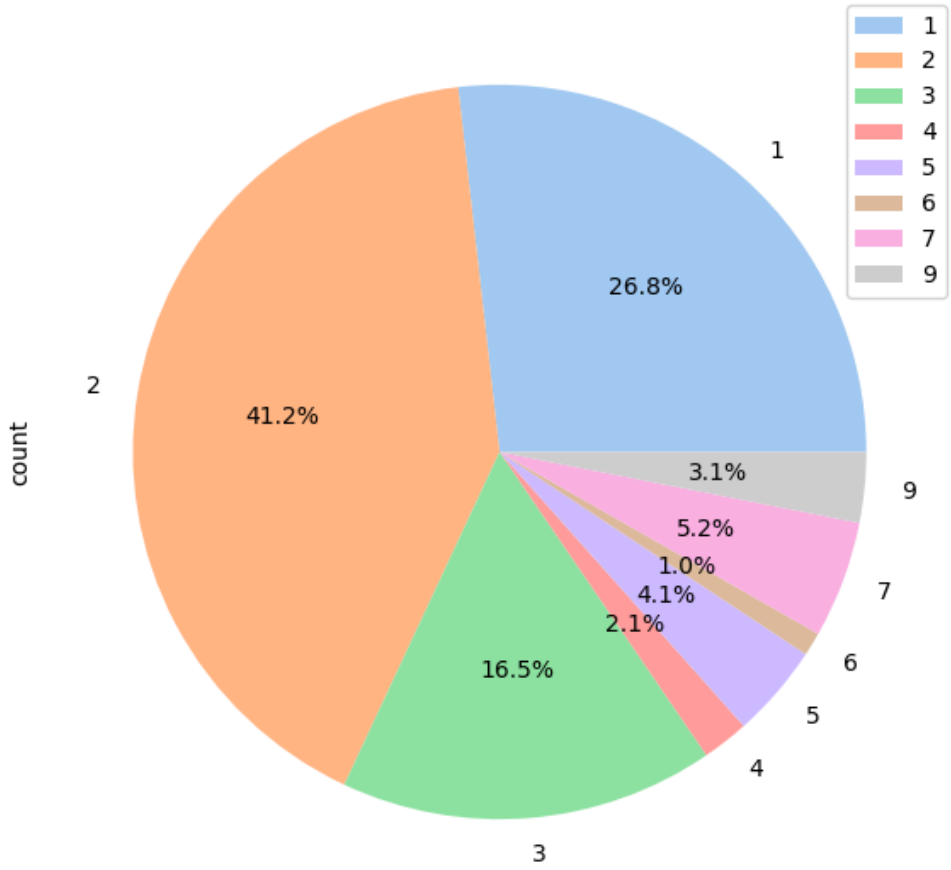
It consist of two ouliers.

In [34]:
```
comfort_food_reasons = pd.DataFrame(df['comfort_food_reasons_coded'].value_counts())
comfort_food_reasons = comfort_food_reasons.sort_values(by=['comfort_food_reasons_coded'], ascending
comfort_food_reasons
```

Out[34]:

comfort_food_reasons_coded	count
1	26
2	40
3	16
4	2
5	4
6	1
7	5
9	3

In [35]:
```
comfort_food_reasons.plot(kind='pie', x='comfort_food_reasons_coded', y='count', autopct='%1.1f%%',
plt.show()
```



Observation: The above pie chart shows the distribution of the reasons why a student eats their preferred comfort food.

1 - stress

2 - boredom

3 - depression/sadness

4 - hunger

5 - laziness

6 - cold weather

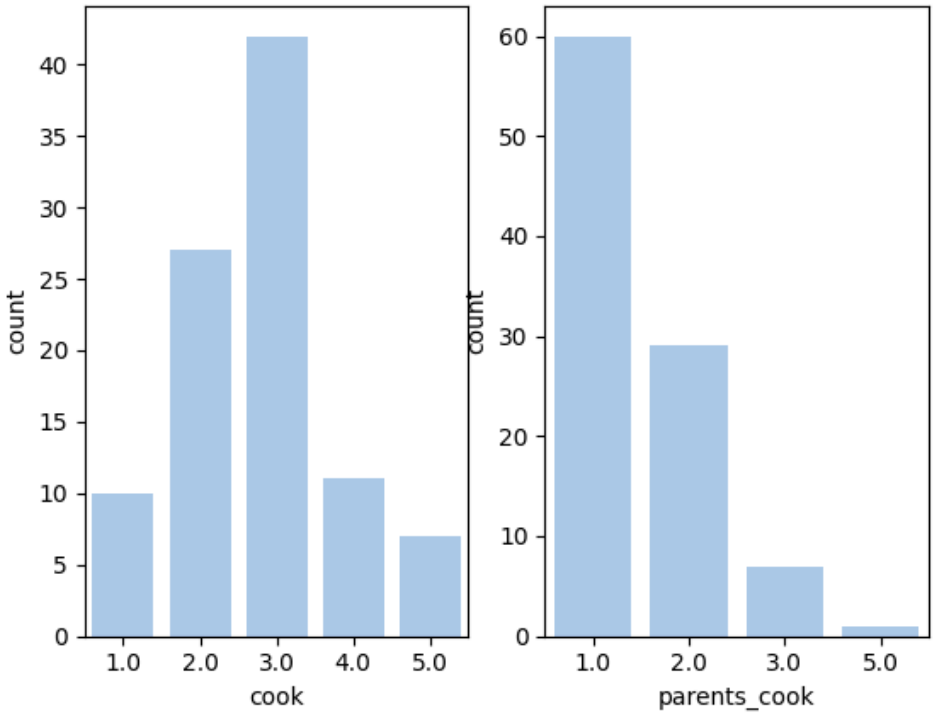
7 - happiness

8 - watching tv

9 - none

The most common reason (41.2%) is boredom. The next most common reason is stress (26.8%), followed by depression/sadness (16.5%)

In [36]:
```
fig, ax =plt.subplots(1,2)
sns.countplot(data=df, x='cook', ax=ax[0])
sns.countplot(data=df, x='parents_cook', ax=ax[1])
fig.show()
```



Observation: The above figure shows two countplots side by side - illustrating the distribution of how often the students cook and how often their cooked.

1 - Almost everyday

2 - 2-3 times a week

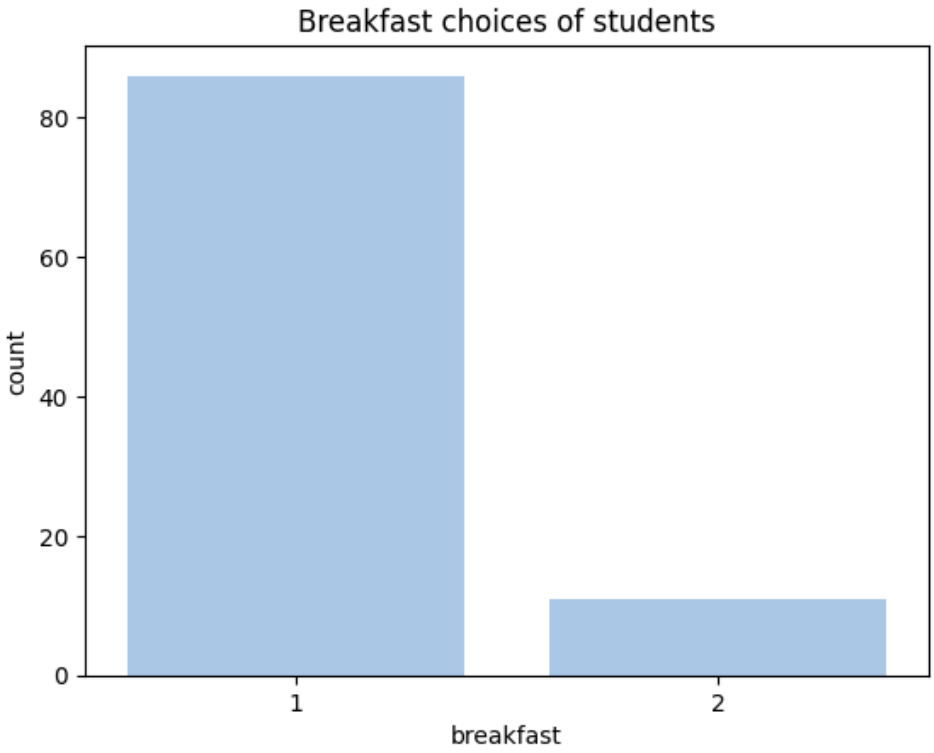
3 - 1-2 times a week

4 - on holidays only

5 - never

It is apparent that most students cook only 1-2 times a week while most parents cooked almost everyday.

In [37]:
```
sns.countplot(data=df, x="breakfast").set(title="Breakfast choices of students")
plt.show()
```

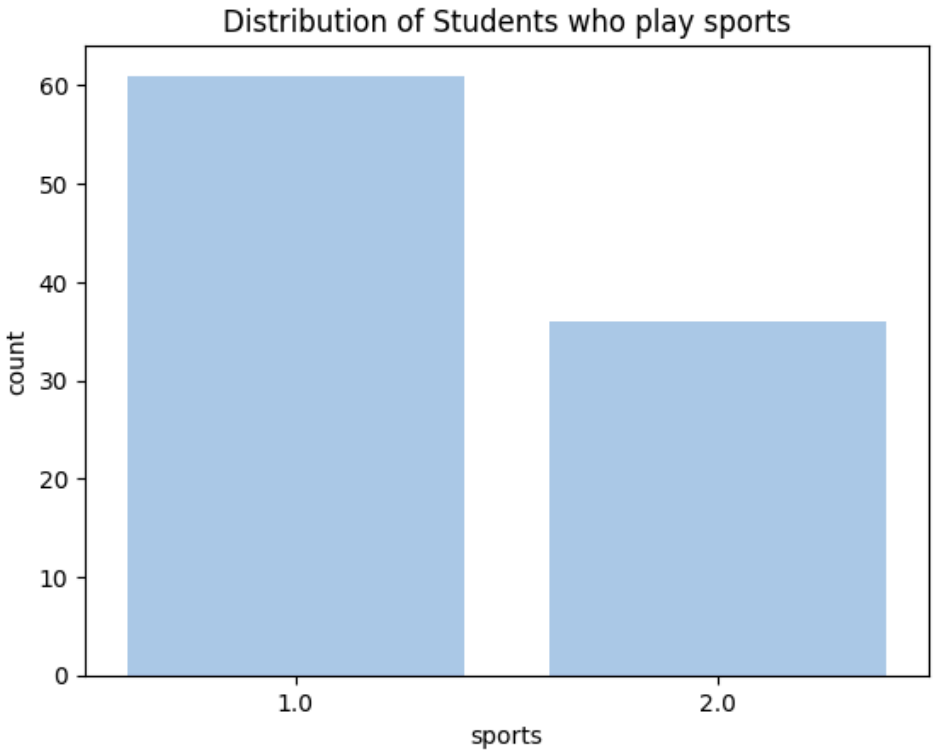


Observation: The above plot shows what option students associate with breakfast.

1 - cereal 2 - donuts

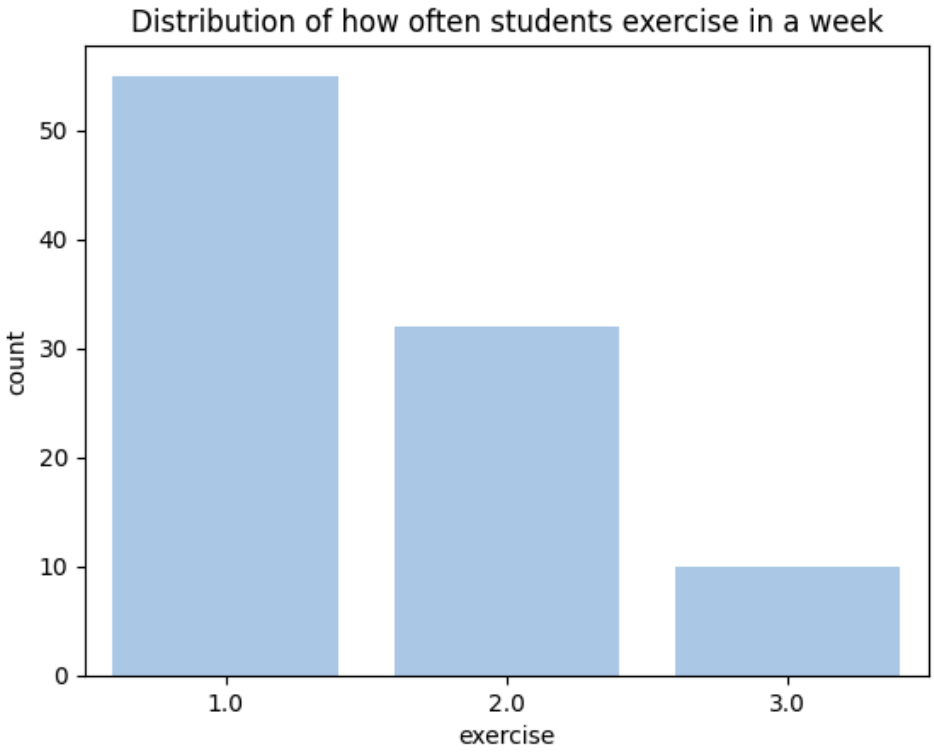
It is clear that an overwhelming amount of students associate the healthier option of cereal with breakfast.

In [38]:
```
sns.countplot(data=df, x="sports").set(title="Distribution of Students who play sports")
plt.show()
```



Observation: Out of the 97 students, 60 said that they do a sporting activity and 37 said they do not performing a sporting activity.

In [39]:
```
sns.countplot(data=df, x="exercise").set(title="Distribution of how often students exercise in a wee
plt.show()
```



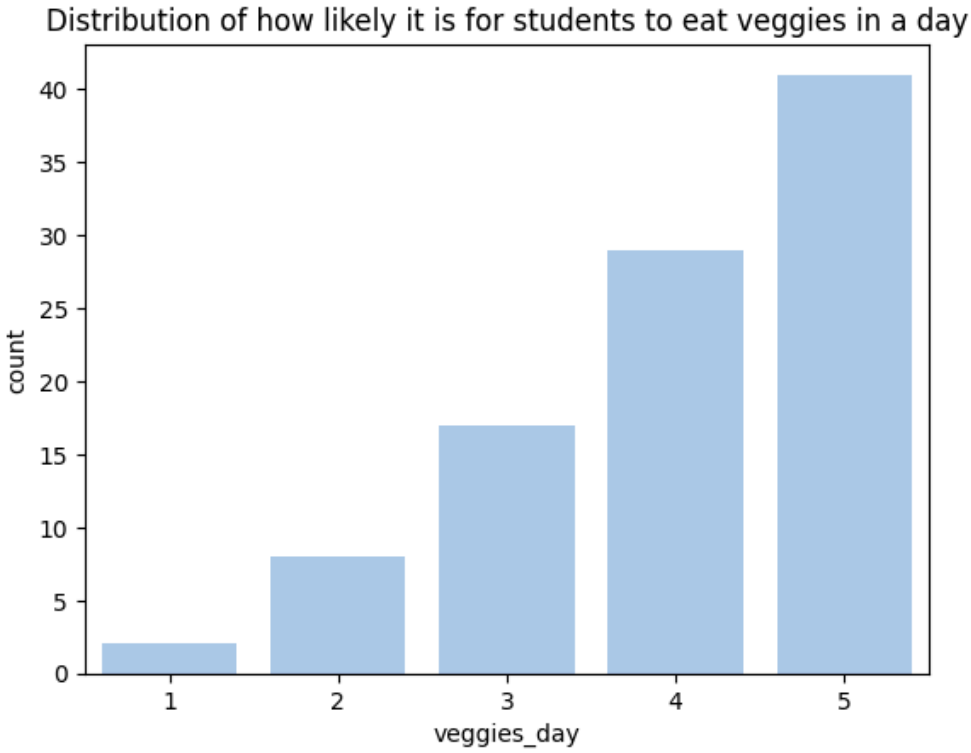
Observation: Majority of students say that they exercise everyday.

1 - Everyday

2 - Two or three times a week

3 - Once a week

In [40]:
```
sns.countplot(data=df, x="veggies_day").set(title="Distribution of how likely it is for students to
plt.show()
```



Observation: Majority of students said that they are very likely to eat veggies in a day.

1 - very unlikely

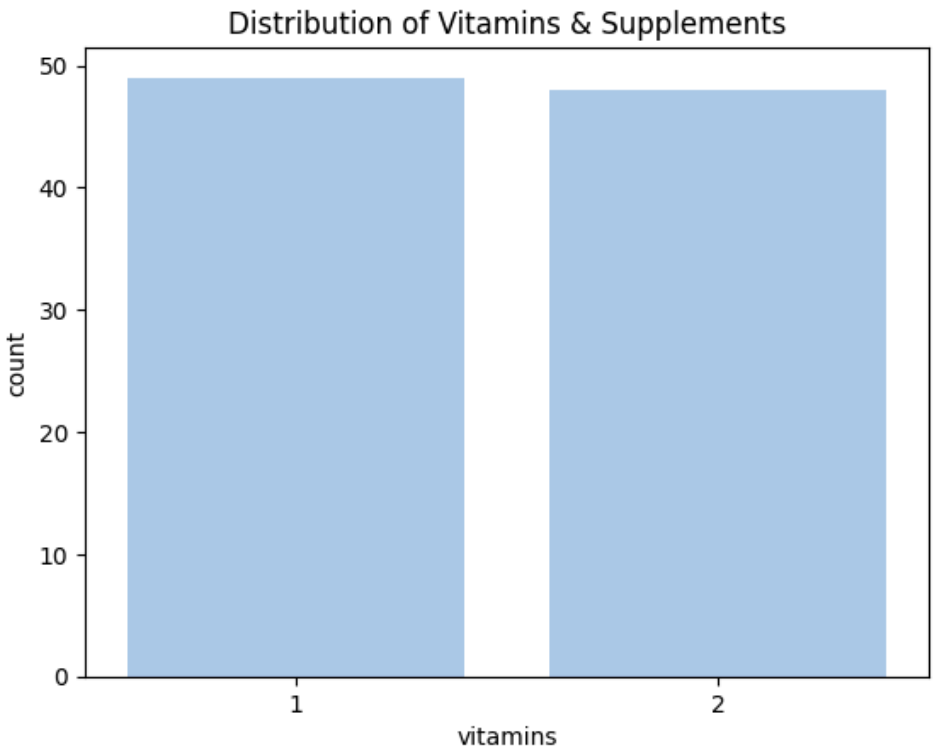
2 - unlikely

3 - neutral

4- likely

5 - very likely

In [41]:
```
sns.countplot(data=df, x="vitamins").set(title="Distribution of Vitamins & Supplements")
plt.show()
```

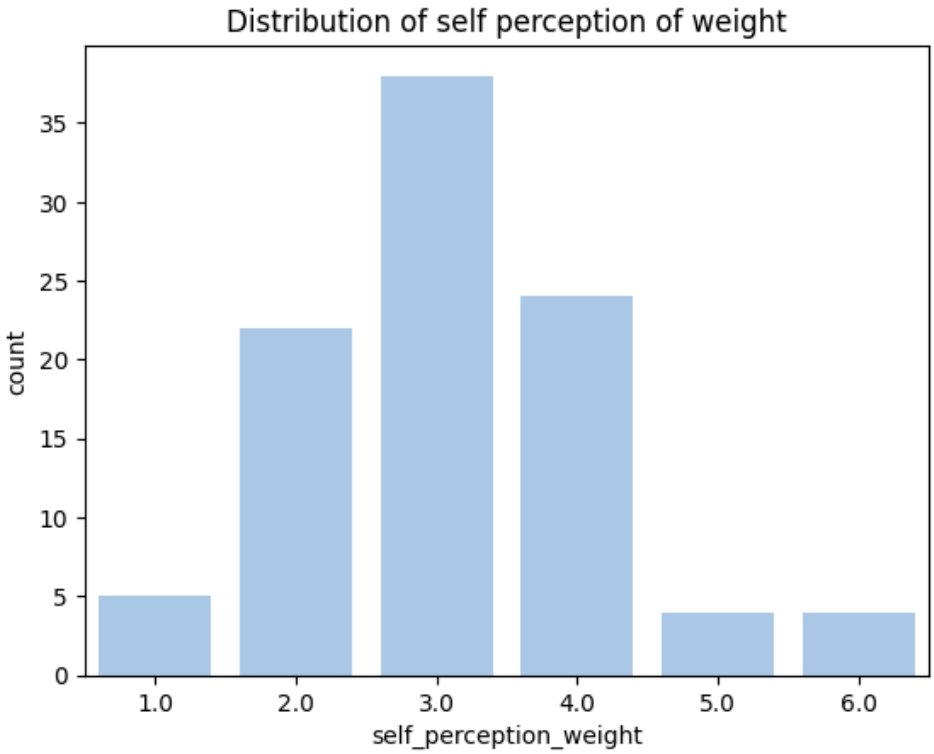


Observation: The distribution of how many students take supplements is 50-50.

1 - yes

2 - no

In [42]:
```
sns.countplot(data=df, x="self_perception_weight").set(title="Distribution of self perception of wei
plt.show()
```



Observation: Most of the students think that they are at just the right weight.

6 - i dont think myself in these terms

5 - overweight

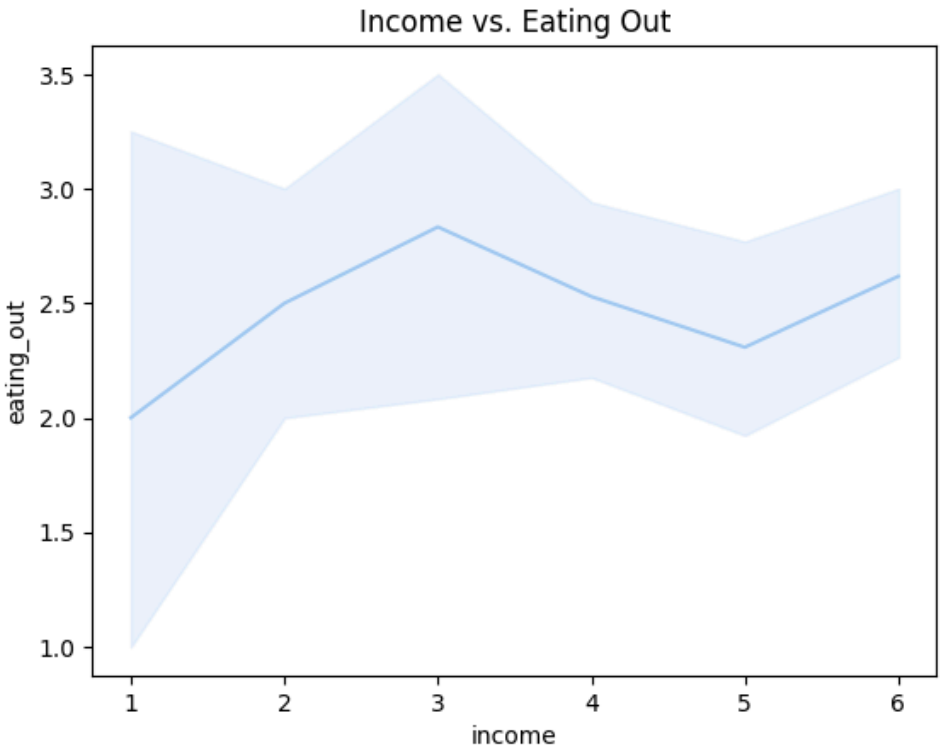
4 - slightly overweight

3 - just right

2 - very fit

1 - slim

In [43]:
```
sns.lineplot(data=df, x="income", y="eating_out").set(title="Income vs. Eating Out")
plt.show()
```



Observation: Students with the income between 30000 to 50000 USD eat out the most frequently.

Income (in dollars)

1 - less than 15,000

2 - 15,001 to 30,000

3 - 30,001 to 50,000

4 - 50,001 to 70,000

5 - 70,001 to 100,000

6 - higher than 100,000

Eating Out

1 - Never

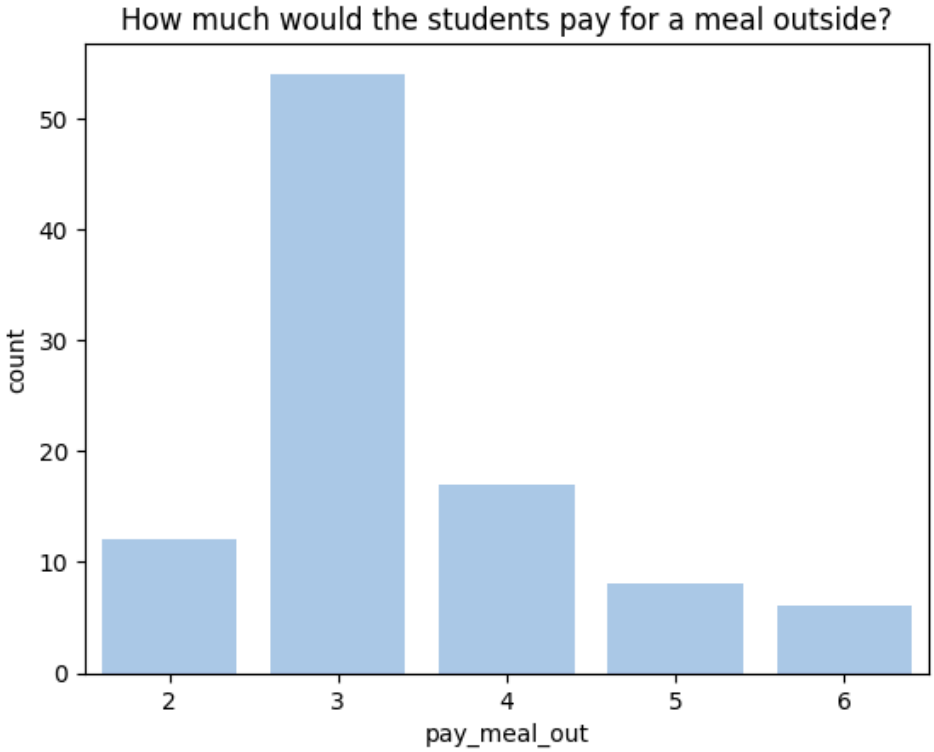
2 - 1-2 times

3 - 2-3 times

4 - 3-5 times

5 - every day

In [44]:
```
sns.countplot(data=df, x="pay_meal_out").set(title="How much would the students pay for a meal outsi
plt.show()
```



Observation: A vast majority of students said they'll pay between 10 to 20 dollars for a meal out.

Pay for a meal outside (in dollars):

1 - up to 5.00

2 - 5.01 to 10.00

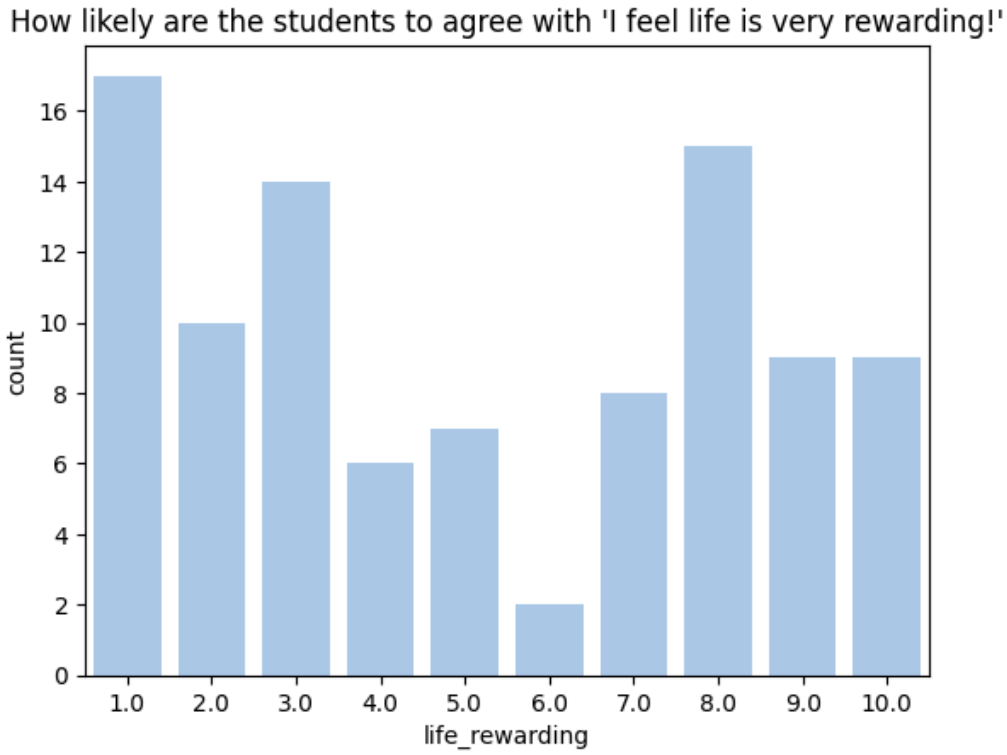
3 - 10.01 to 20.00

4 - 20.01 to 30.00

5 - 30.01 to 40.00

6 - more than 40.01

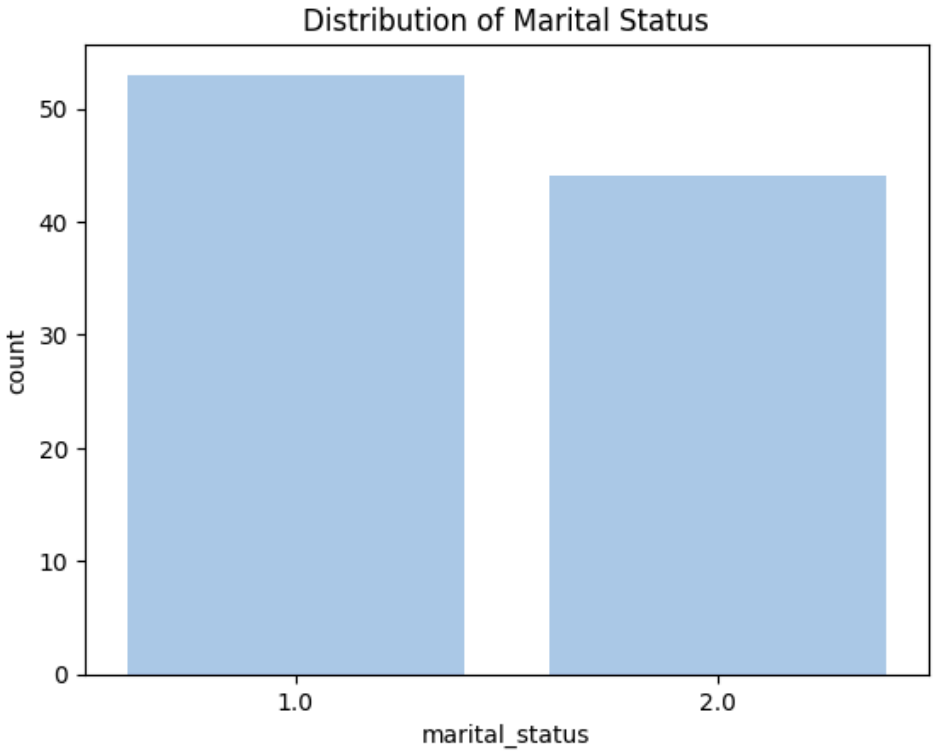
In [45]:
```
sns.countplot(data=df, x="life_rewarding").set(title="How likely are the students to agree with 'I f
plt.show()
```



Observation: Majority of students strongly agree that life is very rewarding.

1 to 10 where 1 is strongly agree and 10 is strongly disagree - scale

In [46]:
```
sns.countplot(data=df, x="marital_status").set(title="Distribution of Marital Status")
plt.show()
```



Observation: Over 50 students are single and around 45 are in a relationship.

1 - Single

2 - In a relationship

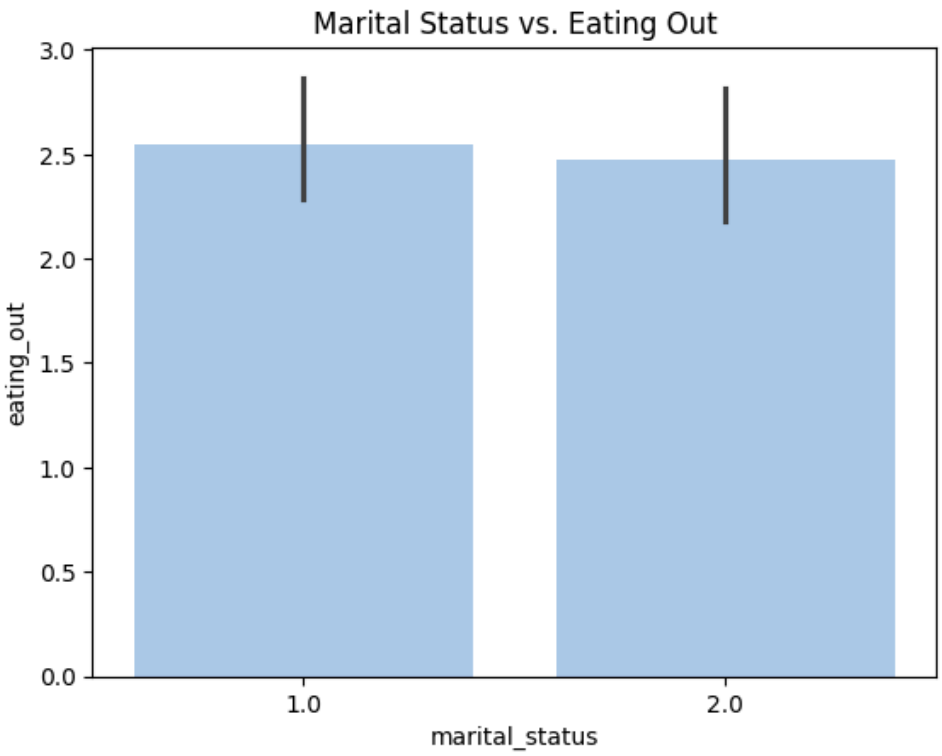
3 - Cohabiting

4 - Married

5 - Divorced

6 - Widowed

In [47]:
```
sns.barplot(data=df, x="marital_status", y="eating_out").set(title="Marital Status vs. Eating Out")
plt.show()
```



Observation: Students who are either single or in a relationship eat out 1-3 times a week.

In [48]:
```
df.to_csv("new_food_choices.csv")
```

In [49]:
```
most_popular_items.to_csv("most_popular_items.csv")
```