

Nutritional Insights: A Data Analysis of Diet Patterns and Health Metrics

Importing the Python libraries.

```
In [8]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn as sk
```

1. The 1st is to load our dataset and Read the data

```
In [13]: data = pd.read_csv("C:\\Users\\ganga\\OneDrive\\Desktop\\Nutritional diet plan.csv")
print("Dataset loaded Successfully.Shape:", data.shape)
print(data)
```

Dataset loaded Successfully.Shape: (120, 18)

	Date	Meal	Time	Food Item	Quantity (g)	\		
0	01-06-2025	Breakfast	7:15 AM	Poha	172			
1	01-06-2025	Lunch	13:45 PM	Palak Curry	170			
2	01-06-2025	Snacks	18:30 PM	Apple	150			
3	01-06-2025	Dinner	19:30 PM	Chapati	101			
4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151			
..			
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149			
116	30-06-2025	Breakfast	9:00 AM	Dosa	168			
117	30-06-2025	Lunch	14:00 PM	White Rice	132			
118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211			
119	30-06-2025	Dinner	20:45 PM	Puri with Potato	246			
	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Meal Type	Homemade	\
0	250	5.0	45	5.0	2	Veg	No	
1	120	5.0	10	6.0	3	Veg	Yes	
2	95	0.5	25	0.3	19	Vegan	Yes	
3	260	8.0	42	7.0	1	Veg	No	
4	220	10.0	30	5.0	2	Vegan	No	
..	
115	265	8.0	38	6.0	2	Vegan	Yes	
116	230	6.0	30	8.0	2	Veg	Yes	
117	260	4.5	56	0.4	0	Veg	Yes	
118	150	2.0	30	2.0	18	Vegan	Yes	
119	340	5.0	45	15.0	4	Veg	No	
	Water Intake (ml)	Mood	After Meal	Physical Activity (steps)	\			
0	300	Energetic		1464				
1	300	Sluggish		927				
2	100	Energetic		1985				
3	100	Okay		1309				
4	250	Sluggish		1627				
..				
115	200	Satisfied		537				
116	150	Tired		655				
117	100	Sluggish		1305				
118	100	Sleepy		929				
119	250	Okay		1587				
	Notes	Exercise Time (minutes)	Hospital Visit					
0	Heavy	30.0	Yes - Routine Checkup					
1	No issues	60.0	No					
2	Felt light	30.0	No					
3	Loved it	60.0	No					
4	No issues	60.0	No					
..					
115	Heavy	60.0	Yes - Stomach issue					
116	Felt light	15.0	Yes - Fever					
117	Tasted good	30.0	Yes - Stomach issue					
118	Felt light	0.0	Yes - Stomach issue					
119	Loved it	45.0	Yes - Fever					

[120 rows x 18 columns]

Data Exploration

#1.Display basic information about the dataset and display the first 5 Rows

```
In [7]: print("Dataset Shape:", data.shape)
print("First 5 rows:")
display(data.head())
```

Dataset Shape: (120, 18)

First 5 rows:

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Meal Type
0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0	2	Veg
1	01-06-2025	Lunch	13:45 PM	Palak Curry	170	120	5.0	10	6.0	3	Veg
2	01-06-2025	Snacks	18:30 PM	Apple	150	95	0.5	25	0.3	19	Vegar
3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0	1	Veg
4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151	220	10.0	30	5.0	2	Vegar

2. Describe function summarizes the statistics for numerical data

```
In [9]: data.describe()
```

Out[9]:

	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Wa Inta (t)
count	120.000000	120.000000	120.000000	120.000000	120.000000	120.000000	120.000000
mean	176.750000	207.541667	5.138333	32.900000	5.276667	5.016667	197.083333
std	45.436862	73.221191	2.541725	13.054121	3.267531	5.233993	71.094667
min	101.000000	90.000000	0.500000	10.000000	0.300000	0.000000	100.000000
25%	142.000000	140.000000	2.750000	25.000000	3.000000	2.000000	150.000000
50%	172.500000	230.000000	6.000000	36.000000	5.000000	3.000000	200.000000
75%	219.000000	260.000000	7.000000	42.000000	7.000000	8.000000	250.000000
max	250.000000	340.000000	10.000000	56.000000	15.000000	19.000000	300.000000

3.To display the list of column names

```
In [11]: data.columns
```

```
Out[11]: Index(['Date', 'Meal', 'Time', 'Food Item', 'Quantity (g)', 'Calories',  
              'Protein (g)', 'Carbs (g)', 'Fat (g)', 'Sugar (g)', 'Meal Type',  
              'Homemade', 'Water Intake (ml)', 'Mood After Meal',  
              'Physical Activity (steps)', 'Notes', 'Exercise Time (minutes)',  
              'Hospital Visit'],  
              dtype='object')
```

4.To print the types of data

```
In [13]: print(data.dtypes)
```

Date	object
Meal	object
Time	object
Food Item	object
Quantity (g)	int64
Calories	int64
Protein (g)	float64
Carbs (g)	int64
Fat (g)	float64
Sugar (g)	int64
Meal Type	object
Homemade	object
Water Intake (ml)	int64
Mood After Meal	object
Physical Activity (steps)	int64
Notes	object
Exercise Time (minutes)	float64
Hospital Visit	object
dtype:	object

5.To print the Last 5 rows

```
In [75]: print("\n5. Last 5 rows:\n", data.tail())
```

5. Last 5 rows:

	Date	Meal	Time	Food Item	Quantity (g)	\
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	
117	30-06-2025	Lunch	14:00 PM	White Rice	132	
118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211	
119	30-06-2025	Dinner	20:45 PM	Puri with Potato	246	

	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Meal Type	Homemade	\
115	265	8.0	38	6.0	2	Vegan	Yes	
116	230	6.0	30	8.0	2	Veg	Yes	
117	260	4.5	56	0.4	0	Veg	Yes	
118	150	2.0	30	2.0	18	Vegan	Yes	
119	340	5.0	45	15.0	4	Veg	No	

	Water Intake (ml)	Mood After Meal	Physical Activity (steps)	\
115	200	Satisfied	537	
116	150	Tired	655	
117	100	Sluggish	1305	
118	100	Sleepy	929	
119	250	Okay	1587	

	Notes	Exercise Time (minutes)	Hospital Visit
115	Heavy	60.0	Yes - Stomach issue
116	Felt light	15.0	Yes - Fever
117	Tasted good	30.0	Yes - Stomach issue
118	Felt light	0.0	Yes - Stomach issue
119	Loved it	45.0	Yes - Fever

6. Summary the data Set

In [49]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120 entries, 0 to 119
Data columns (total 19 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Date                                120 non-null    object
1   Meal                                120 non-null    object
2   Time                                120 non-null    object
3   Food Item                            120 non-null    object
4   Quantity (g)                        120 non-null    int64
5   Calories                            120 non-null    int64
6   Protein (g)                         120 non-null    float64
7   Carbs (g)                           120 non-null    int64
8   Fat (g)                             120 non-null    float64
9   Sugar (g)                           120 non-null    int64
10  Meal Type                            120 non-null    object
11  Homemade                             120 non-null    object
12  Water Intake (ml)                    120 non-null    int64
13  Mood After Meal                      120 non-null    object
14  Physical Activity (steps)            120 non-null    int64
15  Notes                                120 non-null    object
16  Exercise Time (minutes)              119 non-null    float64
17  Hospital Visit                       120 non-null    object
18  Calories_diff                        119 non-null    float64
dtypes: float64(4), int64(6), object(9)
memory usage: 17.9+ KB
```

7.Value counts in 'meal'-value counts means how many times the value appears

```
In [204... print("\n7. Value Counts in 'meal':\n", data['Meal'].value_counts())
```

```
7. Value Counts in 'meal':  
Meal  
Breakfast    30  
Lunch        30  
Snacks       30  
Dinner       30  
Name: count, dtype: int64
```

8.To find the Average Calories

```
In [208... print("\n8. Average Calories:", data['Calories'].mean())
```

```
8. Average Calories: 207.54166666666666
```

9.Meal Proportions

```
In [211... print("\n0.Meal Proportions:\n", data['Meal'].value_counts(normalize=True))
```

```
0.Meal Proportions:  
Meal  
Breakfast    0.25  
Lunch        0.25  
Snacks       0.25  
Dinner       0.25  
Name: proportion, dtype: float64
```

10.Filter the column rows based on a partial matches

```
In [14]: data[data['Food Item'].str.contains("Chapati", case=False, na=False)]
```

Out[14]:

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Mez Typ
3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0	1	Ve
11	03-06-2025	Dinner	19:15 PM	Chapati with Dal	246	265	8.0	38	6.0	2	Vega
15	04-06-2025	Dinner	21:15 PM	Chapati with Dal	133	265	8.0	38	6.0	2	Ve
31	08-06-2025	Dinner	19:30 PM	Chapati with Dal	240	265	8.0	38	6.0	2	Ve
35	09-06-2025	Dinner	21:30 PM	Chapati with Dal	225	265	8.0	38	6.0	2	Vega
39	10-06-2025	Dinner	19:00 PM	Chapati	125	260	8.0	42	7.0	1	Vega
43	11-06-2025	Dinner	21:30 PM	Chapati	239	260	8.0	42	7.0	1	Vega
51	13-06-2025	Dinner	19:30 PM	Chapati	103	260	8.0	42	7.0	1	Vega
67	17-06-2025	Dinner	21:00 PM	Chapati with Dal	213	265	8.0	38	6.0	2	Vega
83	21-06-2025	Dinner	19:45 PM	Chapati with Dal	160	265	8.0	38	6.0	2	Ve
91	23-06-2025	Dinner	21:45 PM	Chapati	211	260	8.0	42	7.0	1	Ve
95	24-06-2025	Dinner	20:15 PM	Chapati with Dal	237	265	8.0	38	6.0	2	Ve
99	25-06-2025	Dinner	21:15 PM	Chapati	179	260	8.0	42	7.0	1	Vega
103	26-06-2025	Dinner	20:45 PM	Chapati with Dal	219	265	8.0	38	6.0	2	Vega
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	265	8.0	38	6.0	2	Vega

11. To print values between range

In [63]: `data[(data['Protein (g)']>=4) & (data['Protein (g)']<=8)]`

Out[63]:

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	
0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0	2	
1	01-06-2025	Lunch	13:45 PM	Palak Curry	170	120	5.0	10	6.0	3	
3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0	1	
7	02-06-2025	Dinner	19:45 PM	Idli with Sambar	144	240	7.0	36	3.0	3	V
8	03-06-2025	Breakfast	9:45 AM	Upma	149	240	6.0	42	5.0	1	
...	
113	29-06-2025	Lunch	14:00 PM	Sambar Rice	181	310	7.0	50	9.0	3	
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	265	8.0	38	6.0	2	V
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	230	6.0	30	8.0	2	
117	30-06-2025	Lunch	14:00 PM	White Rice	132	260	4.5	56	0.4	0	
119	30-06-2025	Dinner	20:45 PM	Puri with Potato	246	340	5.0	45	15.0	4	

77 rows × 19 columns

12.To view the unique Data

In [27]: `data.reset_index()`

Out[27]:

	index	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)
0	0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0
1	1	01-06-2025	Lunch	13:45 PM	Palak Curry	170	120	5.0	10	6.0
2	2	01-06-2025	Snacks	18:30 PM	Apple	150	95	0.5	25	0.3
3	3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0
4	4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151	220	10.0	30	5.0
...
115	115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	265	8.0	38	6.0
116	116	30-06-2025	Breakfast	9:00 AM	Dosa	168	230	6.0	30	8.0
117	117	30-06-2025	Lunch	14:00 PM	White Rice	132	260	4.5	56	0.4
118	118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211	150	2.0	30	2.0
119	119	30-06-2025	Dinner	20:45 PM	Puri with Potato	246	340	5.0	45	15.0

120 rows × 20 columns



13. Removes the Rows with missing values

In [58]:

```
data.dropna()
```

Out[58]:

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0	2
1	01-06-2025	Lunch	13:45 PM	Palak Curry	170	120	5.0	10	6.0	3
2	01-06-2025	Snacks	18:30 PM	Apple	150	95	0.5	25	0.3	19
3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0	1
4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151	220	10.0	30	5.0	2
...
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	265	8.0	38	6.0	2
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	230	6.0	30	8.0	2
117	30-06-2025	Lunch	14:00 PM	White Rice	132	260	4.5	56	0.4	0
118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211	150	2.0	30	2.0	18
119	30-06-2025	Dinner	20:45 PM	Puri with Potato	246	340	5.0	45	15.0	4

119 rows × 18 columns



Data Cleaning

14.To check the missing values

```
In [15]: data.isnull().sum()
```

```
Out[15]: Date          0
        Meal          0
        Time          0
        Food Item      0
        Quantity (g)   0
        Calories       0
        Protein (g)    0
        Carbs (g)      0
        Fat (g)        0
        Sugar (g)      0
        Meal Type      0
        Homemade       0
        Water Intake (ml) 0
        Mood After Meal 0
        Physical Activity (steps) 0
        Notes          0
        Exercise Time (minutes) 1
        Hospital Visit 0
        dtype: int64
```

15.To Remove the duplicates

```
In [218... print("\n.15 To Remove the duplicates :\n",data.drop_duplicates())
```

.15 To Remove the duplicates :

	Date	Meal	Time	Food Item	Quantity (g)	\
0	01-06-2025	Breakfast	7:15 AM	Poha	172	
1	01-06-2025	Lunch	13:45 PM	Palak Curry	170	
2	01-06-2025	Snacks	18:30 PM	Apple	150	
3	01-06-2025	Dinner	19:30 PM	Chapati	101	
4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151	
..	
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	
117	30-06-2025	Lunch	14:00 PM	White Rice	132	
118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211	
119	30-06-2025	Dinner	20:45 PM	Puri with Potato	246	

	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Meal Type	Homemade	\
0	250	5.0	45	5.0	2	Veg	No	
1	120	5.0	10	6.0	3	Veg	Yes	
2	95	0.5	25	0.3	19	Vegan	Yes	
3	260	8.0	42	7.0	1	Veg	No	
4	220	10.0	30	5.0	2	Vegan	No	
..	
115	265	8.0	38	6.0	2	Vegan	Yes	
116	230	6.0	30	8.0	2	Veg	Yes	
117	260	4.5	56	0.4	0	Veg	Yes	
118	150	2.0	30	2.0	18	Vegan	Yes	
119	340	5.0	45	15.0	4	Veg	No	

	Water Intake (ml)	Mood After Meal	Physical Activity (steps)	\
0	300	Energetic	1464	
1	300	Sluggish	927	
2	100	Energetic	1985	
3	100	Okay	1309	
4	250	Sluggish	1627	
..	
115	200	Satisfied	537	
116	150	Tired	655	
117	100	Sluggish	1305	
118	100	Sleepy	929	
119	250	Okay	1587	

	Notes	Exercise Time (minutes)	Hospital Visit
0	Heavy	30.0	Yes - Routine Checkup
1	No issues	60.0	No
2	Felt light	30.0	No
3	Loved it	60.0	No
4	No issues	60.0	No
..
115	Heavy	60.0	Yes - Stomach issue
116	Felt light	15.0	Yes - Fever
117	Tasted good	30.0	Yes - Stomach issue
118	Felt light	0.0	Yes - Stomach issue
119	Loved it	45.0	Yes - Fever

[120 rows x 18 columns]

16.To identify the duplicate rows

```
In [221... print("\n.16 has Rows Count:", data.duplicated())
```

```
.16 has Rows Count: 0      False
1      False
2      False
3      False
4      False
...
115    False
116    False
117    False
118    False
119    False
Length: 120, dtype: bool
```

To save the cleaned data

```
In [87]: data.to_csv('cleaned_data.csv', index=False)
```

Data Filtering

1. Filter by Specific Meal Type

```
In [101... data[data['Meal'] == 'Breakfast']
```

Out[101...

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0	2
4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151	220	10.0	30	5.0	2
8	03-06-2025	Breakfast	9:45 AM	Upma	149	240	6.0	42	5.0	1
12	04-06-2025	Breakfast	8:45 AM	Dosa	135	230	6.0	30	8.0	2
16	05-06-2025	Breakfast	9:45 AM	Poha	213	250	5.0	45	5.0	2
20	06-06-2025	Breakfast	7:15 AM	Upma	220	240	6.0	42	5.0	1
24	07-06-2025	Breakfast	9:00 AM	Bread with Jam	130	200	4.0	32	6.0	12
28	08-06-2025	Breakfast	7:00 AM	Dosa	142	230	6.0	30	8.0	2
32	09-06-2025	Breakfast	8:00 AM	Bread with Jam	103	200	4.0	32	6.0	12
36	10-06-2025	Breakfast	9:15 AM	Pesarattu	237	220	10.0	30	5.0	2
40	11-06-2025	Breakfast	8:00 AM	Upma	231	240	6.0	42	5.0	1
44	12-06-2025	Breakfast	9:00 AM	Idli	182	210	6.0	40	2.0	1
48	13-06-2025	Breakfast	8:15 AM	Bread with Jam	204	200	4.0	32	6.0	12
52	14-06-2025	Breakfast	8:45 AM	Poha	218	250	5.0	45	5.0	2
56	15-06-2025	Breakfast	8:30 AM	Idli	229	210	6.0	40	2.0	1

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
60	16-06-2025	Breakfast	7:15 AM	Upma	175	240	6.0	42	5.0	1
64	17-06-2025	Breakfast	8:45 AM	Idli	104	210	6.0	40	2.0	1
68	18-06-2025	Breakfast	7:45 AM	Idli	171	210	6.0	40	2.0	1
72	19-06-2025	Breakfast	9:45 AM	Poha	124	250	5.0	45	5.0	2
76	20-06-2025	Breakfast	9:00 AM	Poha	170	250	5.0	45	5.0	2
80	21-06-2025	Breakfast	9:00 AM	Dosa	237	230	6.0	30	8.0	2
84	22-06-2025	Breakfast	8:45 AM	Poha	238	250	5.0	45	5.0	2
88	23-06-2025	Breakfast	9:00 AM	Poha	147	250	5.0	45	5.0	2
92	24-06-2025	Breakfast	7:15 AM	Idli	219	210	6.0	40	2.0	1
96	25-06-2025	Breakfast	7:45 AM	Pesarattu	181	220	10.0	30	5.0	2
100	26-06-2025	Breakfast	9:45 AM	Upma	102	240	6.0	42	5.0	1
104	27-06-2025	Breakfast	9:45 AM	Idli	136	210	6.0	40	2.0	1
108	28-06-2025	Breakfast	7:15 AM	Pesarattu	222	220	10.0	30	5.0	2
112	29-06-2025	Breakfast	9:45 AM	Pesarattu	209	220	10.0	30	5.0	2
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	230	6.0	30	8.0	2



2.Filter by Calories or Nutrients (Range-based)

```
In [104... data[(data['Calories'] >= 200) & (data['Calories'] <= 300)]
```

Out[104...

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0	2
3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0	1
4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151	220	10.0	30	5.0	2
7	02-06-2025	Dinner	19:45 PM	Idli with Sambar	144	240	7.0	36	3.0	3
8	03-06-2025	Breakfast	9:45 AM	Upma	149	240	6.0	42	5.0	1
...
111	28-06-2025	Dinner	21:15 PM	Idli with Sambar	108	240	7.0	36	3.0	3
112	29-06-2025	Breakfast	9:45 AM	Pesarattu	209	220	10.0	30	5.0	2
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	265	8.0	38	6.0	2
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	230	6.0	30	8.0	2
117	30-06-2025	Lunch	14:00 PM	White Rice	132	260	4.5	56	0.4	0

64 rows × 18 columns



3.Filter Homemade vs Outside Food

```
In [111... data[data['Homemade'] == 'Yes']
```


Out[111...

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
1	01-06-2025	Lunch	13:45 PM	Palak Curry	170	120	5.0	10	6.0	3
2	01-06-2025	Snacks	18:30 PM	Apple	150	95	0.5	25	0.3	19
6	02-06-2025	Snacks	16:00 PM	Banana	217	105	1.3	27	0.3	14
7	02-06-2025	Dinner	19:45 PM	Idli with Sambar	144	240	7.0	36	3.0	3
12	04-06-2025	Breakfast	8:45 AM	Dosa	135	230	6.0	30	8.0	2
16	05-06-2025	Breakfast	9:45 AM	Poha	213	250	5.0	45	5.0	2
19	05-06-2025	Dinner	21:45 PM	Dosa	158	230	6.0	30	8.0	2
22	06-06-2025	Snacks	16:45 PM	Coffee	106	100	2.0	10	5.0	8
23	06-06-2025	Dinner	19:00 PM	Puri with Potato	170	340	5.0	45	15.0	4
25	07-06-2025	Lunch	14:30 PM	Brinjal Curry	231	140	3.0	15	7.0	2
29	08-06-2025	Lunch	12:15 PM	Dal	204	180	9.0	22	3.0	1
31	08-06-2025	Dinner	19:30 PM	Chapati with Dal	240	265	8.0	38	6.0	2
32	09-06-2025	Breakfast	8:00 AM	Bread with Jam	103	200	4.0	32	6.0	12
33	09-06-2025	Lunch	12:45 PM	Sambar Rice	110	310	7.0	50	9.0	3
34	09-06-2025	Snacks	17:15 PM	Fruit Salad	178	150	2.0	30	2.0	18

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
38	10-06-2025	Snacks	16:15 PM	Coffee	161	100	2.0	10	5.0	8
41	11-06-2025	Lunch	14:15 PM	Sambar Rice	172	310	7.0	50	9.0	3
42	11-06-2025	Snacks	18:45 PM	Apple	245	95	0.5	25	0.3	19
46	12-06-2025	Snacks	17:30 PM	Tea with Sugar	152	90	2.0	12	4.0	10
48	13-06-2025	Breakfast	8:15 AM	Bread with Jam	204	200	4.0	32	6.0	12
52	14-06-2025	Breakfast	8:45 AM	Poha	218	250	5.0	45	5.0	2
53	14-06-2025	Lunch	14:30 PM	Sambar Rice	208	310	7.0	50	9.0	3
54	14-06-2025	Snacks	16:45 PM	Banana	128	105	1.3	27	0.3	14
56	15-06-2025	Breakfast	8:30 AM	Idli	229	210	6.0	40	2.0	1
57	15-06-2025	Lunch	14:45 PM	Vegetable Rice	187	270	6.0	48	5.0	4
58	15-06-2025	Snacks	17:15 PM	Coffee	166	100	2.0	10	5.0	8
61	16-06-2025	Lunch	13:45 PM	Sambar Rice	126	310	7.0	50	9.0	3
62	16-06-2025	Snacks	16:15 PM	Banana	245	105	1.3	27	0.3	14
63	16-06-2025	Dinner	19:00 PM	Dosa	250	230	6.0	30	8.0	2
65	17-06-2025	Lunch	13:00 PM	Sambar Rice	120	310	7.0	50	9.0	3

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
67	17-06-2025	Dinner	21:00 PM	Chapati with Dal	213	265	8.0	38	6.0	2
68	18-06-2025	Breakfast	7:45 AM	Idli	171	210	6.0	40	2.0	1
69	18-06-2025	Lunch	12:45 PM	Brinjal Curry	189	140	3.0	15	7.0	2
72	19-06-2025	Breakfast	9:45 AM	Poha	124	250	5.0	45	5.0	2
73	19-06-2025	Lunch	13:30 PM	White Rice	159	260	4.5	56	0.4	0
76	20-06-2025	Breakfast	9:00 AM	Poha	170	250	5.0	45	5.0	2
77	20-06-2025	Lunch	12:15 PM	Palak Curry	134	120	5.0	10	6.0	3
83	21-06-2025	Dinner	19:45 PM	Chapati with Dal	160	265	8.0	38	6.0	2
88	23-06-2025	Breakfast	9:00 AM	Poha	147	250	5.0	45	5.0	2
90	23-06-2025	Snacks	17:30 PM	Banana	201	105	1.3	27	0.3	14
95	24-06-2025	Dinner	20:15 PM	Chapati with Dal	237	265	8.0	38	6.0	2
96	25-06-2025	Breakfast	7:45 AM	Pesarattu	181	220	10.0	30	5.0	2
97	25-06-2025	Lunch	14:00 PM	Vegetable Rice	220	270	6.0	48	5.0	4
98	25-06-2025	Snacks	17:30 PM	Biscuit	190	180	2.0	25	9.0	10
100	26-06-2025	Breakfast	9:45 AM	Upma	102	240	6.0	42	5.0	1

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
101	26-06-2025	Lunch	12:00 PM	Brinjal Curry	222	140	3.0	15	7.0	2
104	27-06-2025	Breakfast	9:45 AM	Idli	136	210	6.0	40	2.0	1
106	27-06-2025	Snacks	18:45 PM	Banana	249	105	1.3	27	0.3	14
107	27-06-2025	Dinner	19:30 PM	Dosa	154	230	6.0	30	8.0	2
109	28-06-2025	Lunch	14:30 PM	Vegetable Rice	108	270	6.0	48	5.0	4
110	28-06-2025	Snacks	17:00 PM	Tea with Sugar	226	90	2.0	12	4.0	10
114	29-06-2025	Snacks	16:45 PM	Coffee	166	100	2.0	10	5.0	8
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	265	8.0	38	6.0	2
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	230	6.0	30	8.0	2
117	30-06-2025	Lunch	14:00 PM	White Rice	132	260	4.5	56	0.4	0
118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211	150	2.0	30	2.0	18

4.Filter by Mood After Meal

In [114... data[data['Mood After Meal'] == 'Energetic']

Out[114...

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Me Ty
0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0	2	V
2	01-06-2025	Snacks	18:30 PM	Apple	150	95	0.5	25	0.3	19	Veg
14	04-06-2025	Snacks	16:30 PM	Apple	105	95	0.5	25	0.3	19	V
20	06-06-2025	Breakfast	7:15 AM	Upma	220	240	6.0	42	5.0	1	V
26	07-06-2025	Snacks	17:45 PM	Tea with Sugar	146	90	2.0	12	4.0	10	V
39	10-06-2025	Dinner	19:00 PM	Chapati	125	260	8.0	42	7.0	1	Veg
44	12-06-2025	Breakfast	9:00 AM	Idli	182	210	6.0	40	2.0	1	V
46	12-06-2025	Snacks	17:30 PM	Tea with Sugar	152	90	2.0	12	4.0	10	Veg
59	15-06-2025	Dinner	21:15 PM	Idli with Sambar	219	240	7.0	36	3.0	3	V
65	17-06-2025	Lunch	13:00 PM	Sambar Rice	120	310	7.0	50	9.0	3	V
69	18-06-2025	Lunch	12:45 PM	Brinjal Curry	189	140	3.0	15	7.0	2	V
98	25-06-2025	Snacks	17:30 PM	Biscuit	190	180	2.0	25	9.0	10	Veg

5.To Filter the data based on water in take

In [119...

```
data[data['Water Intake (ml)']== 100]
```

Out[119...

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
2	01-06-2025	Snacks	18:30 PM	Apple	150	95	0.5	25	0.3	19
3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0	1
17	05-06-2025	Lunch	12:00 PM	Vegetable Rice	250	270	6.0	48	5.0	4
28	08-06-2025	Breakfast	7:00 AM	Dosa	142	230	6.0	30	8.0	2
31	08-06-2025	Dinner	19:30 PM	Chapati with Dal	240	265	8.0	38	6.0	2
45	12-06-2025	Lunch	14:15 PM	Dal	173	180	9.0	22	3.0	1
48	13-06-2025	Breakfast	8:15 AM	Bread with Jam	204	200	4.0	32	6.0	12
52	14-06-2025	Breakfast	8:45 AM	Poha	218	250	5.0	45	5.0	2
59	15-06-2025	Dinner	21:15 PM	Idli with Sambar	219	240	7.0	36	3.0	3
60	16-06-2025	Breakfast	7:15 AM	Upma	175	240	6.0	42	5.0	1
62	16-06-2025	Snacks	16:15 PM	Banana	245	105	1.3	27	0.3	14
63	16-06-2025	Dinner	19:00 PM	Dosa	250	230	6.0	30	8.0	2
64	17-06-2025	Breakfast	8:45 AM	Idli	104	210	6.0	40	2.0	1
69	18-06-2025	Lunch	12:45 PM	Brinjal Curry	189	140	3.0	15	7.0	2
77	20-06-2025	Lunch	12:15 PM	Palak Curry	134	120	5.0	10	6.0	3

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
79	20-06-2025	Dinner	19:45 PM	Dosa	178	230	6.0	30	8.0	2
87	22-06-2025	Dinner	21:00 PM	Idli with Sambar	208	240	7.0	36	3.0	3
92	24-06-2025	Breakfast	7:15 AM	Idli	219	210	6.0	40	2.0	1
93	24-06-2025	Lunch	13:45 PM	Vegetable Rice	175	270	6.0	48	5.0	4
97	25-06-2025	Lunch	14:00 PM	Vegetable Rice	220	270	6.0	48	5.0	4
98	25-06-2025	Snacks	17:30 PM	Biscuit	190	180	2.0	25	9.0	10
99	25-06-2025	Dinner	21:15 PM	Chapati	179	260	8.0	42	7.0	1
117	30-06-2025	Lunch	14:00 PM	White Rice	132	260	4.5	56	0.4	0
118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211	150	2.0	30	2.0	18

6.To Filter the data by means of physical activity

In [129...

```
data[data['Physical Activity (steps)'] > 500]
```

Out[129...

	Date	Meal	Time	Food Item	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)
0	01-06-2025	Breakfast	7:15 AM	Poha	172	250	5.0	45	5.0	2
1	01-06-2025	Lunch	13:45 PM	Palak Curry	170	120	5.0	10	6.0	3
2	01-06-2025	Snacks	18:30 PM	Apple	150	95	0.5	25	0.3	19
3	01-06-2025	Dinner	19:30 PM	Chapati	101	260	8.0	42	7.0	1
4	02-06-2025	Breakfast	7:15 AM	Pesarattu	151	220	10.0	30	5.0	2
...
115	29-06-2025	Dinner	20:00 PM	Chapati with Dal	149	265	8.0	38	6.0	2
116	30-06-2025	Breakfast	9:00 AM	Dosa	168	230	6.0	30	8.0	2
117	30-06-2025	Lunch	14:00 PM	White Rice	132	260	4.5	56	0.4	0
118	30-06-2025	Snacks	18:45 PM	Fruit Salad	211	150	2.0	30	2.0	18
119	30-06-2025	Dinner	20:45 PM	Puri with Potato	246	340	5.0	45	15.0	4

120 rows × 18 columns



Data Grouping

1.Total Protein per Day

In [133...

```
data.groupby('Date')['Protein (g)'].sum()
```



```
Out[133... Date
01-06-2025    18.5
02-06-2025    21.3
03-06-2025    25.0
04-06-2025    21.5
05-06-2025    18.3
06-06-2025    20.0
07-06-2025    14.0
08-06-2025    25.0
09-06-2025    21.0
10-06-2025    25.0
11-06-2025    21.5
12-06-2025    22.0
13-06-2025    17.0
14-06-2025    18.3
15-06-2025    21.0
16-06-2025    20.3
17-06-2025    23.0
18-06-2025    18.0
19-06-2025    18.5
20-06-2025    18.0
21-06-2025    19.0
22-06-2025    17.5
23-06-2025    18.8
24-06-2025    21.3
25-06-2025    26.0
26-06-2025    19.0
27-06-2025    18.3
28-06-2025    25.0
29-06-2025    27.0
30-06-2025    17.5
Name: Protein (g), dtype: float64
```

2.Average protein and fat grouped by meal type.

```
In [139... data.groupby('Meal')[['Protein (g)', 'Fat (g)']].mean()
```

```
Out[139...      Protein (g)    Fat (g)
Meal
Breakfast    6.233333    4.900000
Dinner       7.033333    7.366667
Lunch        5.700000    5.753333
Snacks       1.586667    3.086667
```

3.Water intake grouped by meal

```
In [146... data.groupby('Meal')['Water Intake (ml)'].mean()
```

```
Out[146... Meal
Breakfast      200.000000
Dinner         181.666667
Lunch          208.333333
Snacks         198.333333
Name: Water Intake (ml), dtype: float64
```

4. Calories grouped by Food_item

```
In [149... data.groupby('Food Item')['Calories'].sum()
```

```
Out[149... Food Item
Apple          475
Banana         735
Biscuit        540
Bread with Jam 600
Brinjal Curry  700
Chapati       1560
Chapati with Dal 2385
Coffee         800
Dal           540
Dosa          1840
Fruit Salad   450
Idli          1260
Idli with Sambar 1440
Palak Curry   600
Pesarattu     1100
Poha          1750
Puri with Potato 1700
Sambar Rice   2480
Tea with Sugar 360
Upma          1200
Vegetable Rice 1350
White Rice    1040
Name: Calories, dtype: int64
```

5. Exercise item was grouped by meal

```
In [154... data.groupby('Meal')['Exercise Time (minutes)'].mean()
```

```
Out[154... Meal
Breakfast      17.000000
Dinner         21.833333
Lunch          24.333333
Snacks         22.931034
Name: Exercise Time (minutes), dtype: float64
```

6. Food item was grouped by Hospital visit

```
In [156... data.groupby('Food Item')['Hospital Visit'].value_counts()
```

```
Out[156... Food Item      Hospital Visit
Apple      No          2
           Yes - Stomach issue 2
           Yes - Fever        1
Banana     No          3
           Yes - Fever        2
           ..
Vegetable Rice  Yes - Stomach issue 3
              No          1
              Yes -Routine checkup 1
White Rice     Yes - Fever        2
              Yes - Stomach issue 2
Name: count, Length: 69, dtype: int64
```

7.data was grouped by the date, protein in (gm), carbs(gm),fat(gm)

```
In [160... data.groupby('Date')[['Protein (g)', 'Carbs (g)', 'Fat (g)']].sum()
```

Out[160...

	Protein (g)	Carbs (g)	Fat (g)
Date			
01-06-2025	18.5	122	18.3
02-06-2025	21.3	108	15.3
03-06-2025	25.0	132	16.0
04-06-2025	21.5	143	23.3
05-06-2025	18.3	150	18.3
06-06-2025	20.0	147	34.0
07-06-2025	14.0	104	32.0
08-06-2025	25.0	100	22.0
09-06-2025	21.0	150	23.0
10-06-2025	25.0	92	23.0
11-06-2025	21.5	159	21.3
12-06-2025	22.0	119	24.0
13-06-2025	17.0	114	29.0
14-06-2025	18.3	167	29.3
15-06-2025	21.0	134	15.0
16-06-2025	20.3	149	22.3
17-06-2025	23.0	153	26.0
18-06-2025	18.0	101	17.0
19-06-2025	18.5	149	12.4
20-06-2025	18.0	95	24.0
21-06-2025	19.0	149	14.7
22-06-2025	17.5	116	14.3
23-06-2025	18.8	170	12.7
24-06-2025	21.3	153	13.3
25-06-2025	26.0	145	26.0
26-06-2025	19.0	105	23.0
27-06-2025	18.3	107	16.3
28-06-2025	25.0	126	17.0
29-06-2025	27.0	128	25.0
30-06-2025	17.5	161	25.4

8. Returns the size (count) of each group

```
In [10]: data.groupby('Meal').size()
```

```
Out[10]: Meal
Breakfast    30
Dinner       30
Lunch        30
Snacks       30
dtype: int64
```

Data Aggregation

1.Returns the largest value in each column

```
In [14]: data.min()
```

```
Out[14]: Date                01-06-2025
Meal                Breakfast
Time                12:00 PM
Food Item            Apple
Quantity (g)         101
Calories              90
Protein (g)          0.5
Carbs (g)            10
Fat (g)              0.3
Sugar (g)            0
Meal Type            Veg
Homemade             No
Water Intake (ml)    100
Mood After Meal      Energetic
Physical Activity (steps)  504
Notes                Digestive discomfort
Exercise Time (minutes)  0.0
Hospital Visit       No
dtype: object
```

2.Returns the largest value in each column

```
In [16]: data.max()
```

```
Out[16]: Date          30-06-2025
Meal          Snacks
Time          9:45 AM
Food Item     White Rice
Quantity (g)  250
Calories      340
Protein (g)   10.0
Carbs (g)     56
Fat (g)       15.0
Sugar (g)     19
Meal Type     Vegan
Homemade      Yes
Water Intake (ml) 300
Mood After Meal Tired
Physical Activity (steps) 1989
Notes         Very filling
Exercise Time (minutes) 60.0
Hospital Visit Yes- Stomach issue
dtype: object
```

3. Mean of Numeric Columns

```
In [18]: num_col = data.select_dtypes(include='number').columns
mean_v = data[num_col].mean()
mean_v
```

```
Out[18]: Quantity (g)          176.750000
Calories          207.541667
Protein (g)        5.138333
Carbs (g)          32.900000
Fat (g)            5.276667
Sugar (g)          5.016667
Water Intake (ml)  197.083333
Physical Activity (steps) 1249.816667
Exercise Time (minutes) 21.512605
dtype: float64
```

4. To Find the Median in a specific column or a number columns

```
In [34]: median_v = data[['Calories', 'Protein (g)']].median()
median_v
```

```
Out[34]: Calories          230.0
Protein (g)           6.0
dtype: float64
```

5. To Calculate the variance for a given data

```
In [40]: var_v = data[['Carbs (g)', 'Exercise Time (minutes)']].var()
var_v
```

```
Out[40]: Carbs (g)          170.410084
Exercise Time (minutes)  269.302806
dtype: float64
```

Calculate the variance for numbers contain columns

```
In [43]: var_v = data[num_col].var()
var_v
```

```
Out[43]: Quantity (g)          2064.508403
Calories          5361.342787
Protein (g)        6.460367
Carbs (g)         170.410084
Fat (g)           10.676762
Sugar (g)          27.394678
Water Intake (ml)  5054.446779
Physical Activity (steps) 172552.117367
Exercise Time (minutes) 269.302806
dtype: float64
```

6. Relationship between the variables can be concluded by corr()

```
In [50]: corr_v = data[num_col].corr()
corr_v
```

```
Out[50]:
```

	Quantity (g)	Calories	Protein (g)	Carbs (g)	Fat (g)	Sugar (g)	Water Intake (ml)	
Quantity (g)	1.000000	0.036653	0.063599	0.000609	0.040543	-0.089593	0.012129	-(
Calories	0.036653	1.000000	0.709482	0.866198	0.558238	-0.637097	0.007086	(
Protein (g)	0.063599	0.709482	1.000000	0.484513	0.300600	-0.775931	0.008065	-(
Carbs (g)	0.000609	0.866198	0.484513	1.000000	0.159877	-0.402524	-0.017068	(
Fat (g)	0.040543	0.558238	0.300600	0.159877	1.000000	-0.378669	0.093576	(
Sugar (g)	-0.089593	-0.637097	-0.775931	-0.402524	-0.378669	1.000000	-0.024710	-(
Water Intake (ml)	0.012129	0.007086	0.008065	-0.017068	0.093576	-0.024710	1.000000	(
Physical Activity (steps)	-0.037684	0.138503	-0.012677	0.161829	0.118594	-0.013882	0.000679	·
Exercise Time (minutes)	-0.041890	-0.028982	-0.018867	-0.063677	0.101650	-0.002511	0.101186	-(

Correlation between the specified variables

```
In [58]: corr_v = data[['Water Intake (ml)', 'Physical Activity (steps)', 'Sugar (g)']].corr()
corr_v
```

```
Out[58]: <bound method DataFrame.corr of
s) Sugar (g)
0          300          1464          2
1          300           927          3
2          100          1985         19
3          100          1309          1
4          250          1627          2
..          ...          ...          ...
115         200           537          2
116         150           655          2
117         100          1305          0
118         100           929         18
119         250          1587          4

[120 rows x 3 columns]>
```

Multiple Aggregation of Table

```
In [81]: data.agg({'Calories': ['sum', 'mean'], 'Protein (g)': ['max', 'min'],
                  'Fat (g)': ['sum'], 'Carbs (g)': ['mean', 'sum'], 'Sugar (g)': ['mean', 'sum'],
                  'Water Intake (ml)': ['sum', 'max'] })
```

```
Out[81]:
```

	Calories	Protein (g)	Fat (g)	Carbs (g)	Sugar (g)	Exercise Time (minutes)	Water Intake (ml)
sum	24905.000000	NaN	633.2	3948.0	602.000000	2560.000000	23650.0
mean	207.541667	NaN	NaN	32.9	5.016667	21.512605	NaN
max	NaN	10.0	NaN	NaN	NaN	NaN	300.0
min	NaN	0.5	NaN	NaN	NaN	NaN	NaN

Diet/Nutrition Rating Analysis

```
In [111... data['health_score'] = (data['Protein (g)'] * 2) - (data['Fat (g)'] * 1.5) - (data['Carbs (g)'] * 1)
data.groupby('Meal')['health_score'].mean()
```

```
Out[111... Meal
Breakfast    2.840000
Dinner       0.348333
Lunch        0.533333
Snacks       -2.576667
Name: health_score, dtype: float64
```

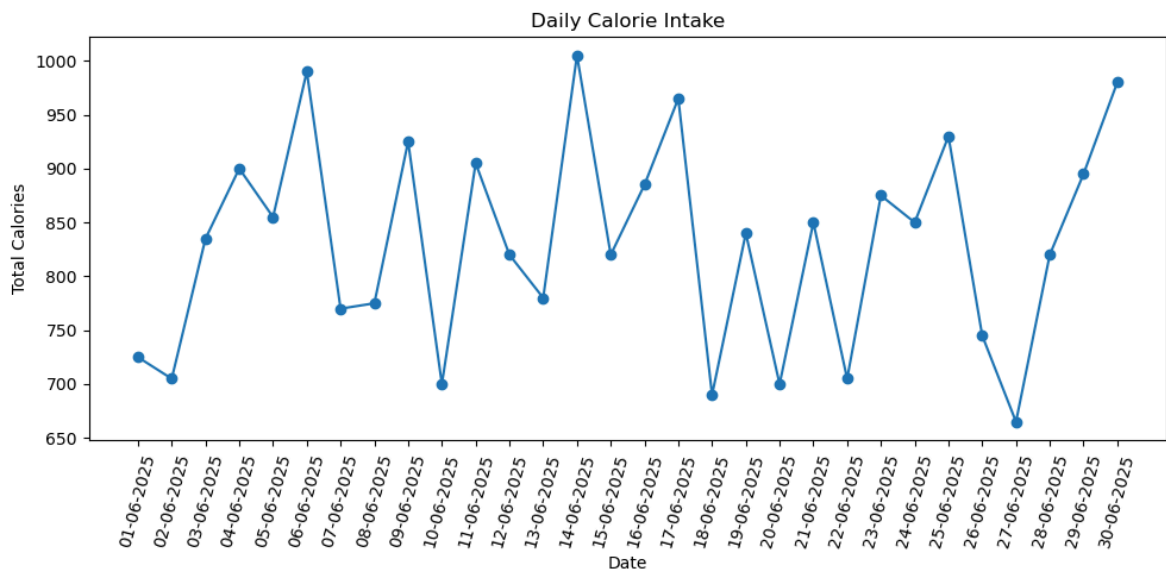
Data Visualization

1.Line plot

```
In [75]: daily_calories = data.groupby('Date')['Calories'].sum()
plt.figure(figsize=(10,5))
plt.plot(daily_calories.index, daily_calories.values, marker='o')
plt.title('Daily Calorie Intake')
plt.xlabel('Date')
```



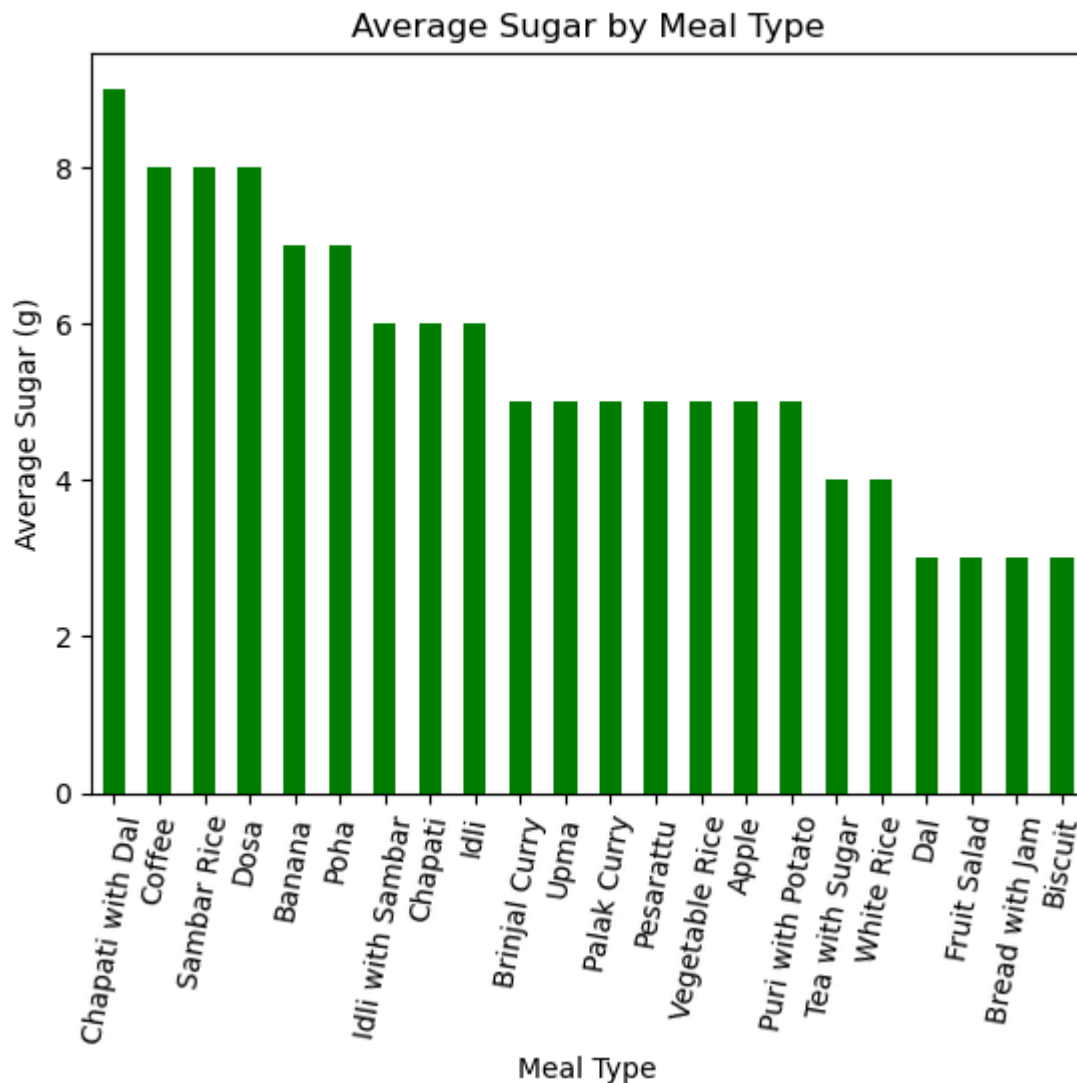
```
plt.ylabel('Total Calories')
plt.xticks(rotation=75)
plt.tight_layout()
plt.show()
```



2.Bar Plot:- Average Sugar by Meal Type

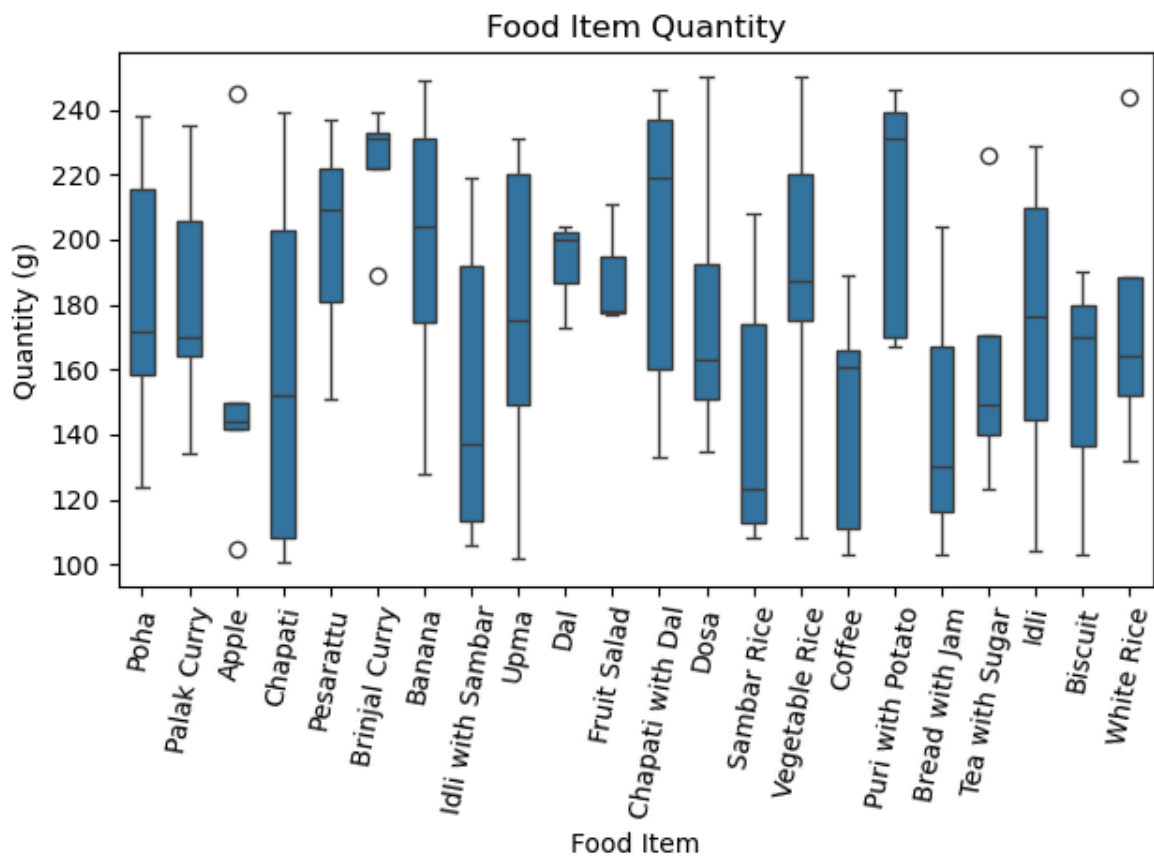
In [201...

```
data['Food Item'].value_counts().plot(kind='bar', color='green',width=0.5)
avg_sugar = data.groupby('Meal')['Sugar (g)'].mean().sort_values()
plt.title('Average Sugar by Meal Type')
plt.xlabel('Meal Type')
plt.ylabel('Average Sugar (g)')
plt.xticks(rotation=80)
plt.show()
```



3.Box plot :- Food item Quantity

```
In [98]: df = pd.read_csv("C:\\Users\\ganga\\OneDrive\\Desktop\\Nutritional diet plan.csv")
sns.boxplot(x='Food Item',y='Quantity (g)',width=0.5,data=df)
plt.title('Food Item Quantity')
plt.xticks(rotation=80)
plt.tight_layout()
plt.show()
```

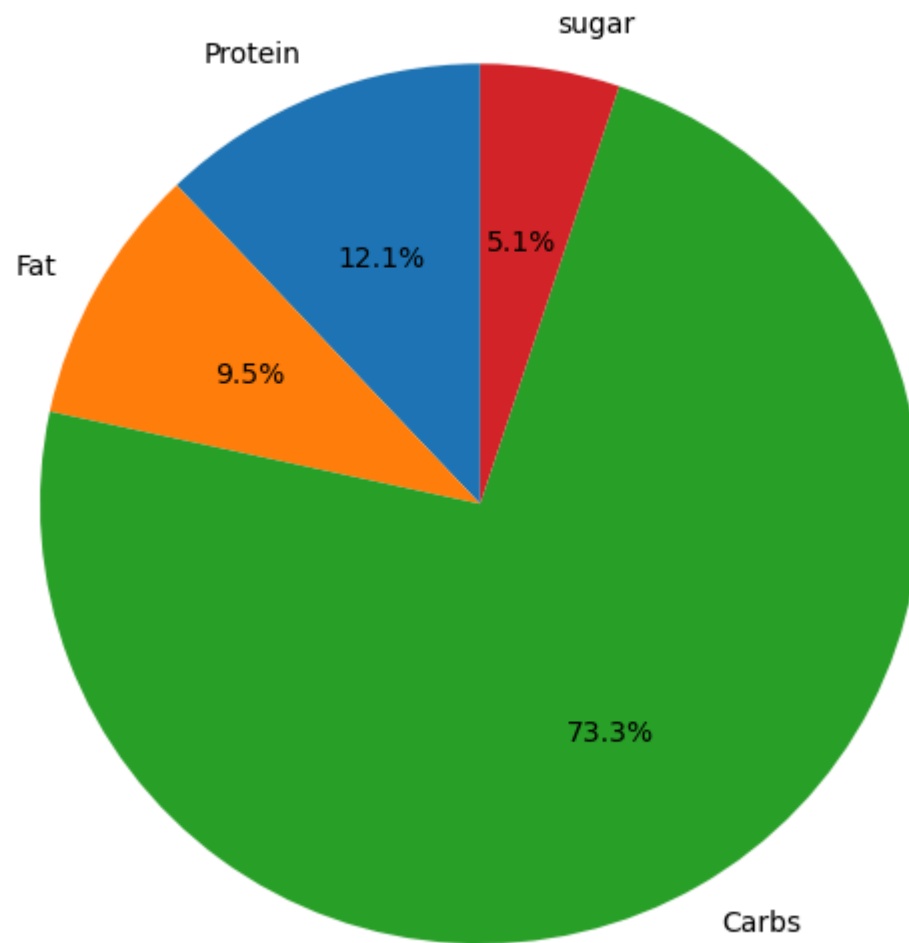


4. Piechart : Nutrient Share in Breakfast

In [122...

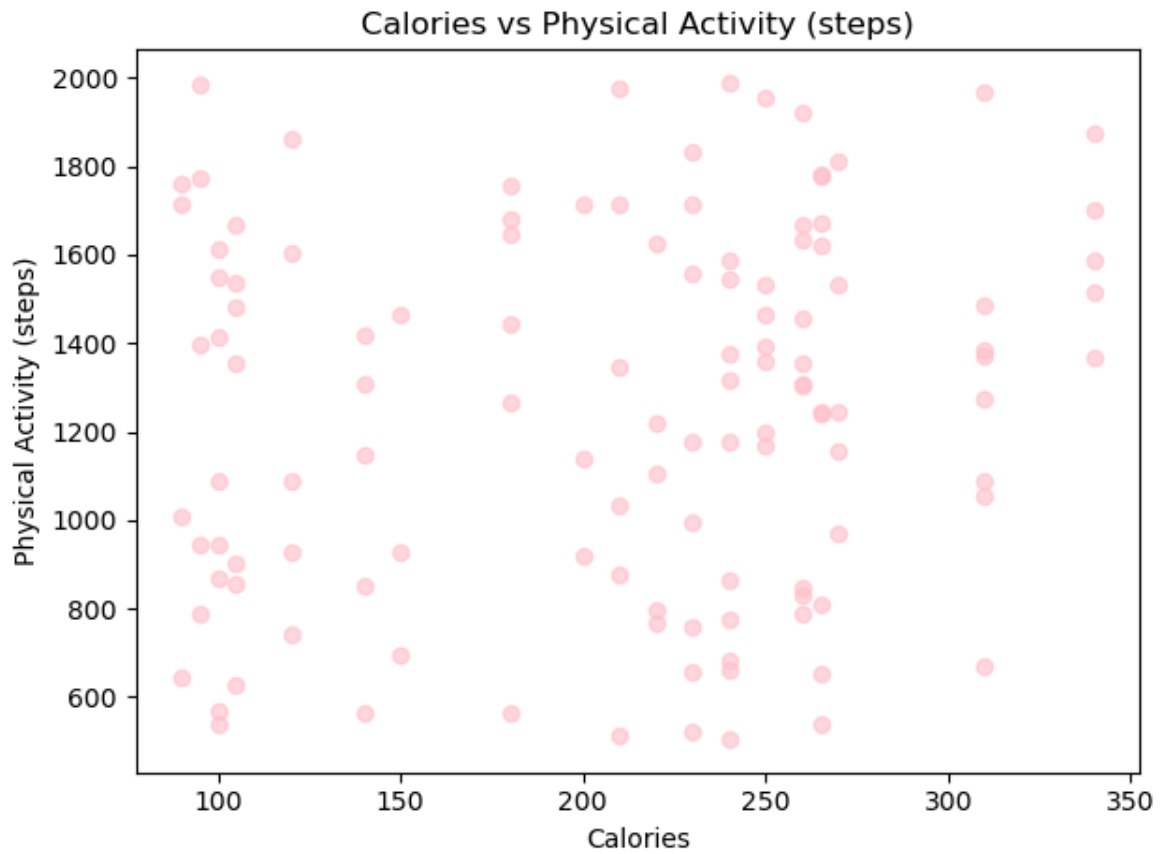
```
meal_type = 'Breakfast'
meal_data = df[df['Meal'] == meal_type]
Protein = meal_data['Protein (g)'].sum()
Fat = meal_data['Fat (g)'].sum()
Carbs = meal_data['Carbs (g)'].sum()
Sugar = meal_data['Sugar (g)'].sum()
labels = ['Protein', 'Fat', 'Carbs', 'sugar']
values = [Protein, Fat, Carbs, Sugar]
plt.figure(figsize=(6,6))
plt.pie(values, labels=labels, autopct='%1.1f%%', startangle=90)
plt.title(f'Nutrient Share in {meal_type}')
plt.tight_layout()
plt.show()
```

Nutrient Share in Breakfast



5.Scatter Plot:- Calories vs Quantity

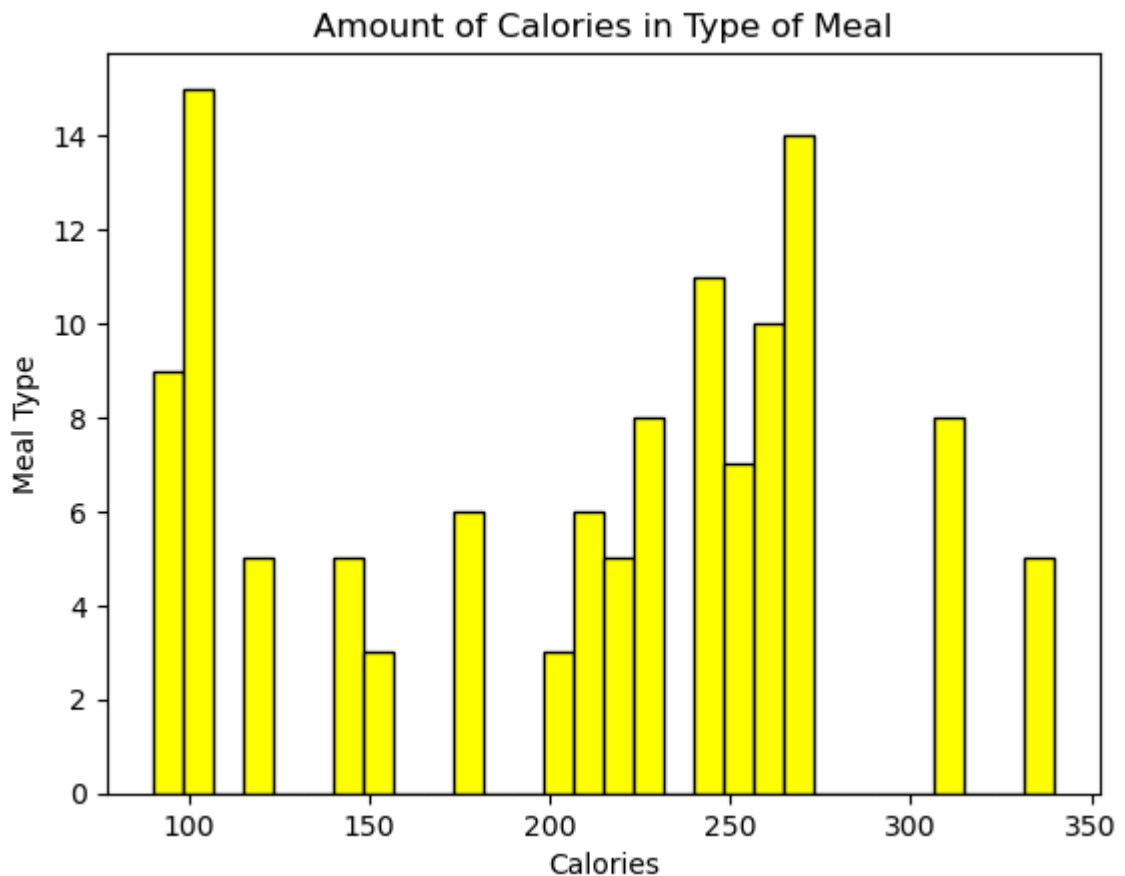
```
In [131... plt.scatter(df['Calories'], df['Physical Activity (steps)'], alpha=0.6, color='p
plt.title('Calories vs Physical Activity (steps)')
plt.xlabel('Calories')
plt.ylabel('Physical Activity (steps)')
plt.tight_layout()
plt.show()
```



6.Histogram:- Amount of Calories in Type of Meal

In [144...

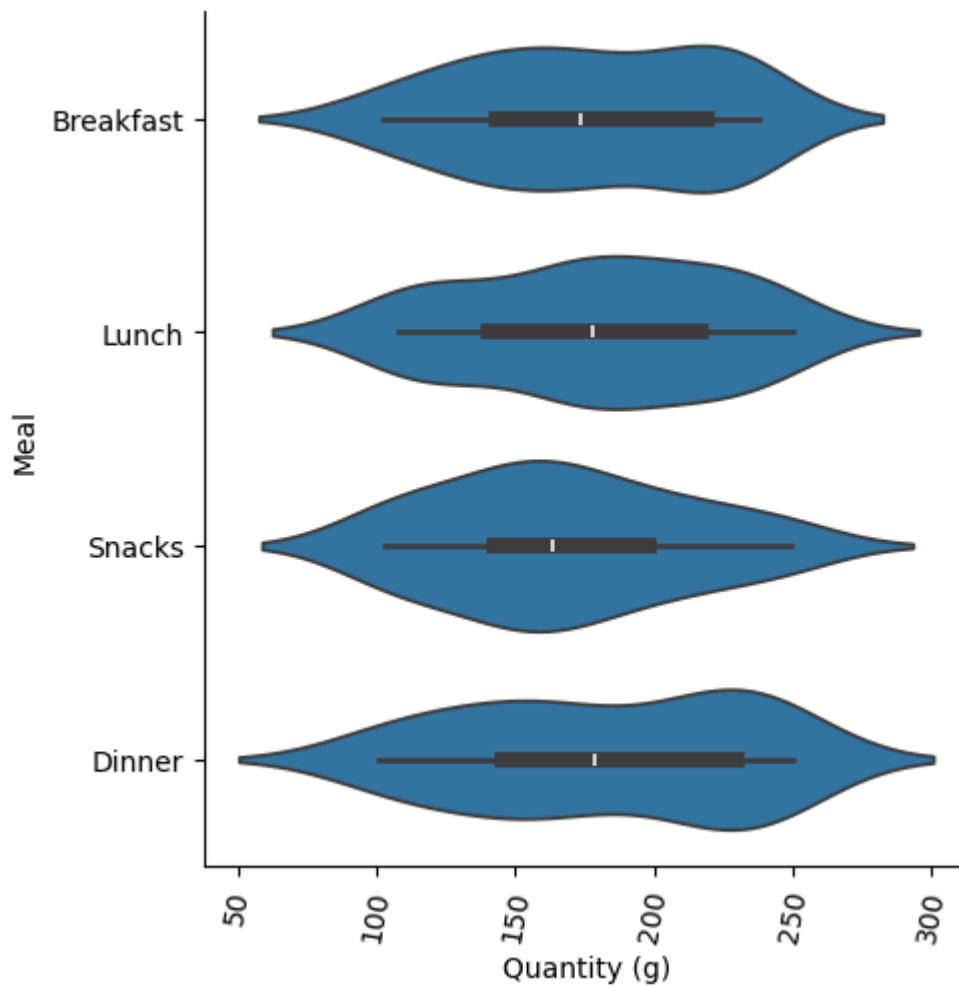
```
plt.hist(df['Calories'], bins=30, color='yellow', edgecolor='black')
plt.title('Amount of Calories in Type of Meal')
plt.xlabel('Calories')
plt.ylabel('Meal Type')
plt.show()
```



7.Violin plot

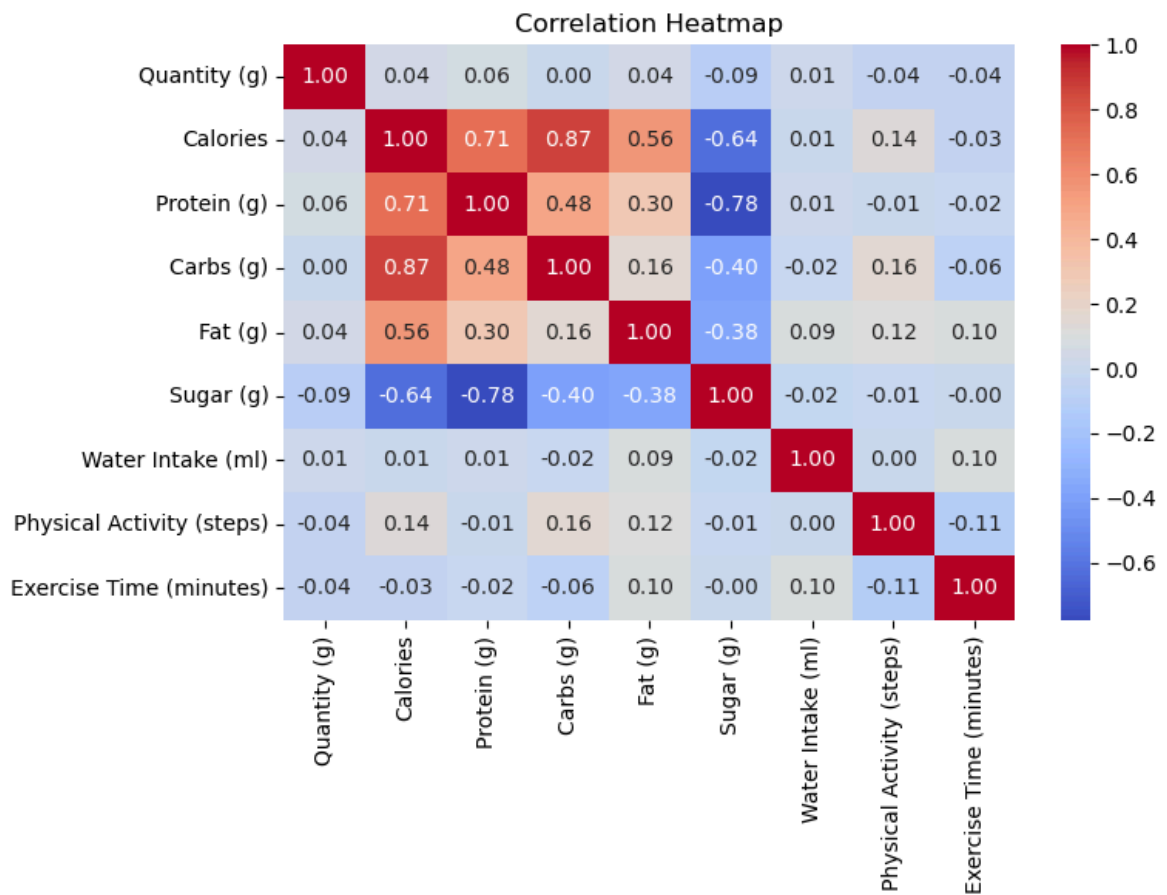
```
In [169... df = pd.read_csv("C:\\Users\\ganga\\OneDrive\\Desktop\\Nutritional diet plan.csv")
sns.catplot(x='Quantity (g)',y='Meal',data=df,kind='violin')
plt.xticks(rotation=80)
```

```
Out[169... (array([ 0., 50., 100., 150., 200., 250., 300., 350.]),
 [Text(0.0, 0, '0'),
  Text(50.0, 0, '50'),
  Text(100.0, 0, '100'),
  Text(150.0, 0, '150'),
  Text(200.0, 0, '200'),
  Text(250.0, 0, '250'),
  Text(300.0, 0, '300'),
  Text(350.0, 0, '350')])
```



8. Heatmap:-Correlation matrix for the selected Numeric columns.

```
In [173... numeric_cols = df.select_dtypes(include='number')
corr = numeric_cols.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```



9.Barplot For the food is homemade or not

```
In [18]: df = pd.read_csv("C:\\Users\\ganga\\OneDrive\\Desktop\\Nutritional diet plan.csv")
plt.figure(figsize=(6, 4))
sns.barplot(x='Homemade', y='Calories', hue='Homemade', data=avg_cal, palette='pastel')
plt.title('Average Calories: Homemade vs Non-Homemade')
plt.xlabel('Homemade')
plt.ylabel('Average Calories')
plt.tight_layout()
plt.show()
```