In [1]:
```
import pandas as pd
import numpy as np
import warnings as w
w.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv("C:\\Users\\Admin\\Downloads\\India_Menu.csv")
```

In [3]:
```
data.dtypes
```

Out[3]:
```
Menu Category              object
Menu Items                 object
Per Serve Size             object
Energy (kCal)             float64
Protein (g)               float64
Total fat (g)             float64
Sat Fat (g)               float64
Trans fat (g)             float64
Cholesterols (mg)         float64
Total carbohydrate (g)    float64
Total Sugars (g)          float64
Added Sugars (g)          float64
Sodium (mg)               float64
dtype: object
```

In [4]:
```
data.head()
```

Out[4]:

	Menu Category	Menu Items	Per Serve Size	Energy (kCal)	Protein (g)	Total fat (g)	Sat Fat (g)	Trans fat (g)	Cholesterols (mg)	Total carbohydrate (g)
0	Regular Menu	McVeggie™ Burger	168 g	402.05	10.24	13.83	5.34	0.16	2.49	56.54
1	Regular Menu	McAloo Tikki Burger®	146 g	339.52	8.50	11.31	4.27	0.20	1.47	50.27
2	Regular Menu	McSpicy™ Paneer Burger	199 g	652.76	20.29	39.45	17.12	0.18	21.85	52.33
3	Regular Menu	Spicy Paneer Wrap	250 g	674.68	20.96	39.10	19.73	0.26	40.93	59.27
4	Regular Menu	American Veg Burger	177 g	512.17	15.30	23.45	10.51	0.17	25.24	56.96

In [5]:
```
data.shape
```

Out[5]:
```
(141, 13)
```

In [6]:
```
# Checking for the null values
data.isnull().sum()
```

Out[6]:
```
Menu Category             0
Menu Items                0
Per Serve Size            0
Energy (kCal)             0
Protein (g)               0
Total fat (g)             0
Sat Fat (g)               0
Trans fat (g)             0
Cholesterols (mg)         0
Total carbohydrate (g)    0
Total Sugars (g)          0
Added Sugars (g)          0
Sodium (mg)               1
dtype: int64
```

In [7]:
```
# Replacing the null value
data['Sodium (mg)'].fillna(data['Sodium (mg)'].mean(),inplace = True)
```

In [8]:
```
data.isnull().sum().sum()
```

Out[8]:
```
0
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 141 entries, 0 to 140
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Menu Category           141 non-null    object 
 1   Menu Items              141 non-null    object 
 2   Per Serve Size          141 non-null    object 
 3   Energy (kCal)           141 non-null    float64
 4   Protein (g)             141 non-null    float64
 5   Total fat (g)           141 non-null    float64
 6   Sat Fat (g)             141 non-null    float64
 7   Trans fat (g)           141 non-null    float64
 8   Cholesterols (mg)       141 non-null    float64
 9   Total carbohydrate (g)  141 non-null    float64
 10  Total Sugars (g)        141 non-null    float64
 11  Added Sugars (g)        141 non-null    float64
 12  Sodium (mg)             141 non-null    float64
dtypes: float64(10), object(3)
memory usage: 14.4+ KB
```

In [10]:
```
data
```

Out[10]:

	Menu Category	Menu Items	Per Serve Size	Energy (kCal)	Protein (g)	Total fat (g)	Sat Fat (g)	Trans fat (g)	Cholesterols (mg)	To carbohydr
0	Regular Menu	McVeggie™ Burger	168 g	402.05	10.24	13.83	5.34	0.16	2.49	56
1	Regular Menu	McAloo Tikki Burger®	146 g	339.52	8.50	11.31	4.27	0.20	1.47	50
2	Regular Menu	McSpicy™ Paneer Burger	199 g	652.76	20.29	39.45	17.12	0.18	21.85	52
3	Regular Menu	Spicy Paneer Wrap	250 g	674.68	20.96	39.10	19.73	0.26	40.93	59
4	Regular Menu	American Veg Burger	177 g	512.17	15.30	23.45	10.51	0.17	25.24	56
...	...	...	...	...	...	...	...	...	...	
136	Condiments Menu	Tomato Ketchup Sachets	8 g	11.23	0.08	23.45	0.00	0.01	0.08	2
137	Condiments Menu	Maple Syrup	30 g	86.40	0.00	0.00	0.00	0.00	0.30	21
138	Condiments Menu	Cheese Slice	14 g	51.03	3.06	3.99	2.89	0.01	13.43	0
139	Condiments Menu	Sweet Corn	40 g	45.08	1.47	1.00	0.22	0.04	2.00	7
140	Condiments Menu	Mixed Fruit Beverage	180 ml	72.25	0.65	0.02	0.02	0.02	0.01	18

141 rows × 13 columns

In [11]:
```
# group by menu item and calulate mean energy for each item
menu_items_energy = data.groupby('Menu Items')['Energy (kCal)'].mean()
```

In [12]:
```
print(menu_items_energy)
```

```
Menu Items
2 piece Chicken Strips          164.44
3 piece Chicken Strips          246.65
4 piece Chicken McNuggets       169.68
5 piece Chicken Strips          411.09
6 piece Chicken McNuggets       254.52
                                 ...
Tomato Ketchup Sachets           11.23
Vanilla Chocochips Muffin       329.29
Vedica Natural Mineral Water      0.00
Veg Maharaja Mac                832.67
Veg McMuffin                    309.35
Name: Energy (kCal), Length: 141, dtype: float64
```

In [13]:
```
# select the top 15 menu items
top_cal_items = menu_items_energy.nlargest(15)
```

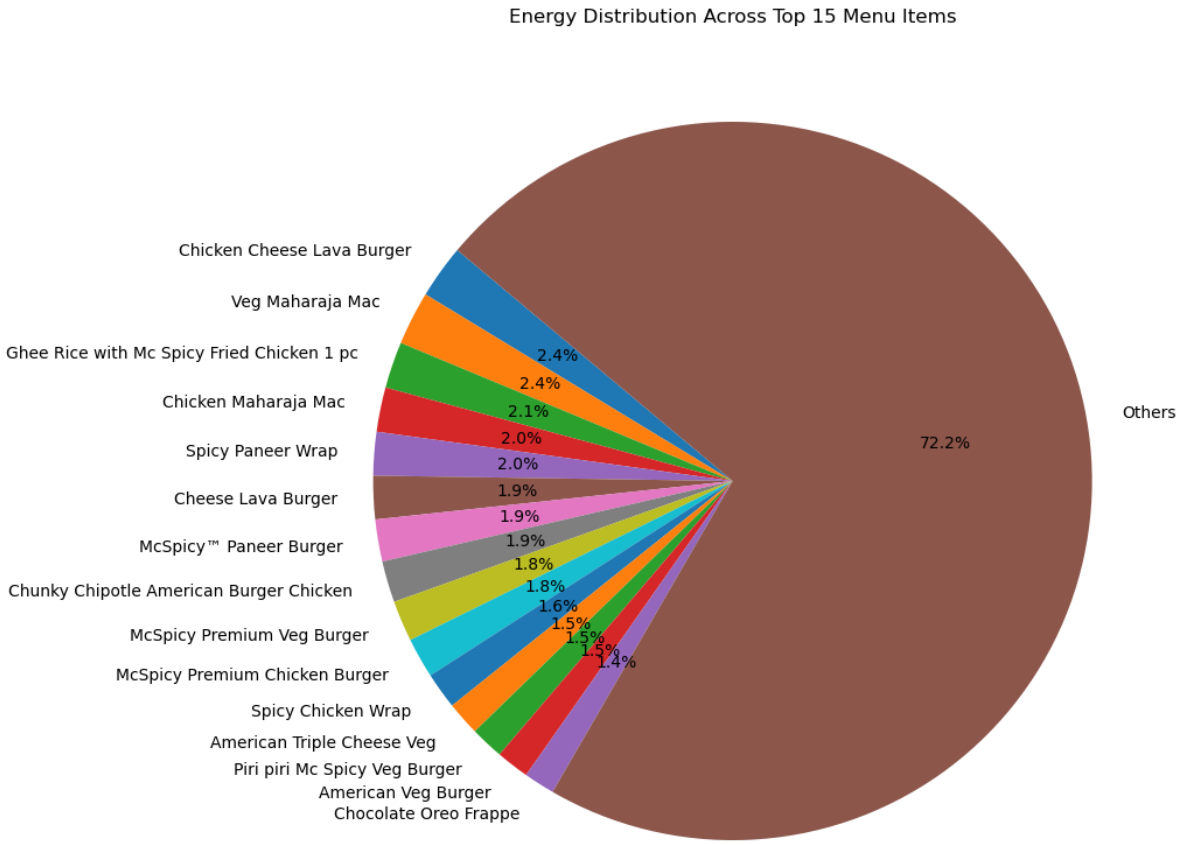
In [14]:
```
# caluculating the total energy of other menu items
other_energy = menu_items_energy.drop(top_cal_items.index).sum()
top_cal_items['Others'] = other_energy
```

In [15]:
```
top_cal_items
```

Out[15]:
```
Menu Items
Chicken Cheese Lava Burger                      834.36
Veg Maharaja Mac                                832.67
Ghee Rice with Mc Spicy Fried Chicken 1 pc      720.30
Chicken Maharaja Mac                            689.12
Spicy Paneer Wrap                               674.68
Cheese Lava Burger                              671.06
McSpicy™ Paneer Burger                          652.76
Chunky Chipotle American Burger Chicken         641.36
McSpicy Premium Veg Burger                      634.71
McSpicy Premium Chicken Burger                  622.25
Spicy Chicken Wrap                              567.19
American Triple Cheese Veg                      524.69
Piri piri Mc Spicy Veg Burger                   517.98
American Veg Burger                             512.17
Chocolate Oreo Frappe                           481.11
Others                                        24917.19
Name: Energy (kCal), dtype: float64
```

In [16]:
```
plt.figure(figsize=(10,10))
plt.pie(top_cal_items, labels=top_cal_items.index, autopct='%1.1f%%',startangle=140
plt.title("Energy Distribution Across Top 15 Menu Items")
```

Out[16]:
```
Text(0.5, 1.0, 'Energy Distribution Across Top 15 Menu Items')
```



In [17]:
```
# Descriptive Statistics
data.describe
```

Out[17]:

```
<bound method NDFrame.describe of        Menu Category              Menu Items Per
Serve Size  Energy (kCal)  \
0       Regular Menu        McVeggie™ Burger          168 g         402.05
1       Regular Menu    McAloo Tikki Burger®          146 g         339.52
2       Regular Menu  McSpicy™ Paneer Burger          199 g         652.76
3       Regular Menu       Spicy Paneer Wrap          250 g         674.68
4       Regular Menu     American Veg Burger          177 g         512.17
..               ...                     ...            ...           ...
136  Condiments Menu  Tomato Ketchup Sachets            8 g          11.23
137  Condiments Menu             Maple Syrup           30 g          86.40
138  Condiments Menu            Cheese Slice           14 g          51.03
139  Condiments Menu              Sweet Corn           40 g          45.08
140  Condiments Menu    Mixed Fruit Beverage         180 ml          72.25

     Protein (g)  Total fat (g)  Sat Fat (g)  Trans fat (g)  \
0          10.24          13.83         5.34           0.16
1           8.50          11.31         4.27           0.20
2          20.29          39.45        17.12           0.18
3          20.96          39.10        19.73           0.26
4          15.30          23.45        10.51           0.17
..           ...            ...          ...            ...
136         0.08          23.45         0.00           0.01
137         0.00           0.00         0.00           0.00
138         3.06           3.99         2.89           0.01
139         1.47           1.00         0.22           0.04
140         0.65           0.02         0.02           0.02

     Cholesterols (mg)  Total carbohydrate (g)  Total Sugars (g)  \
0                 2.49                   56.54              7.90
1                 1.47                   50.27              7.05
2                21.85                   52.33              8.35
3                40.93                   59.27              3.50
4                25.24                   56.96              7.85
..                 ...                     ...               ...
136               0.08                    2.63              2.33
137               0.30                   21.60             16.20
138              13.43                    0.72              0.54
139               2.00                    7.55              2.54
140               0.01                   18.00             16.83

     Added Sugars (g)  Sodium (mg)
0                4.49       706.13
1                4.07       545.34
2                5.27      1074.58
3                1.08      1087.46
4                4.76      1051.24
..                ...          ...
136              1.64        71.05
137              5.34        15.00
138              0.00       178.95
139              0.00         0.04
140              0.00        10.80

[141 rows x 13 columns]>
```

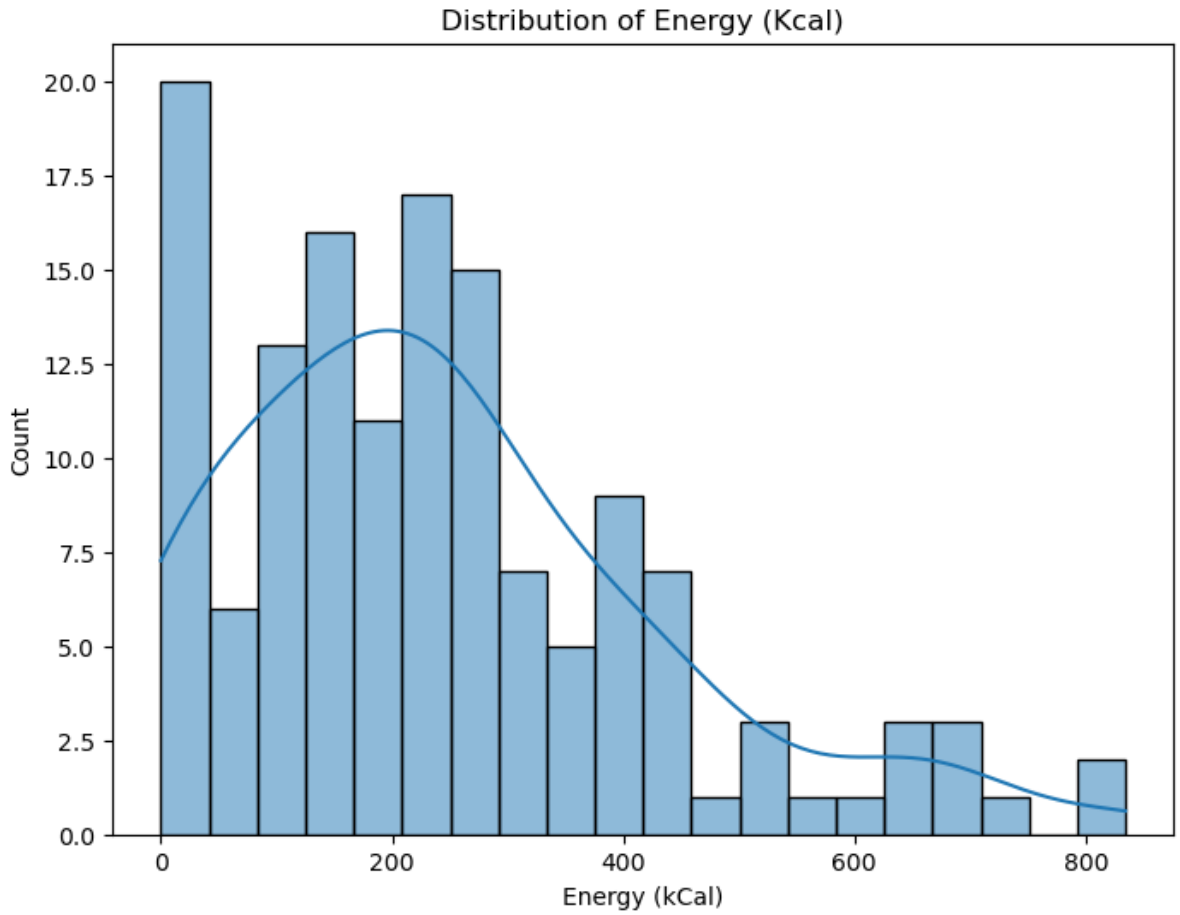
In [18]:

```
data.tail()
```

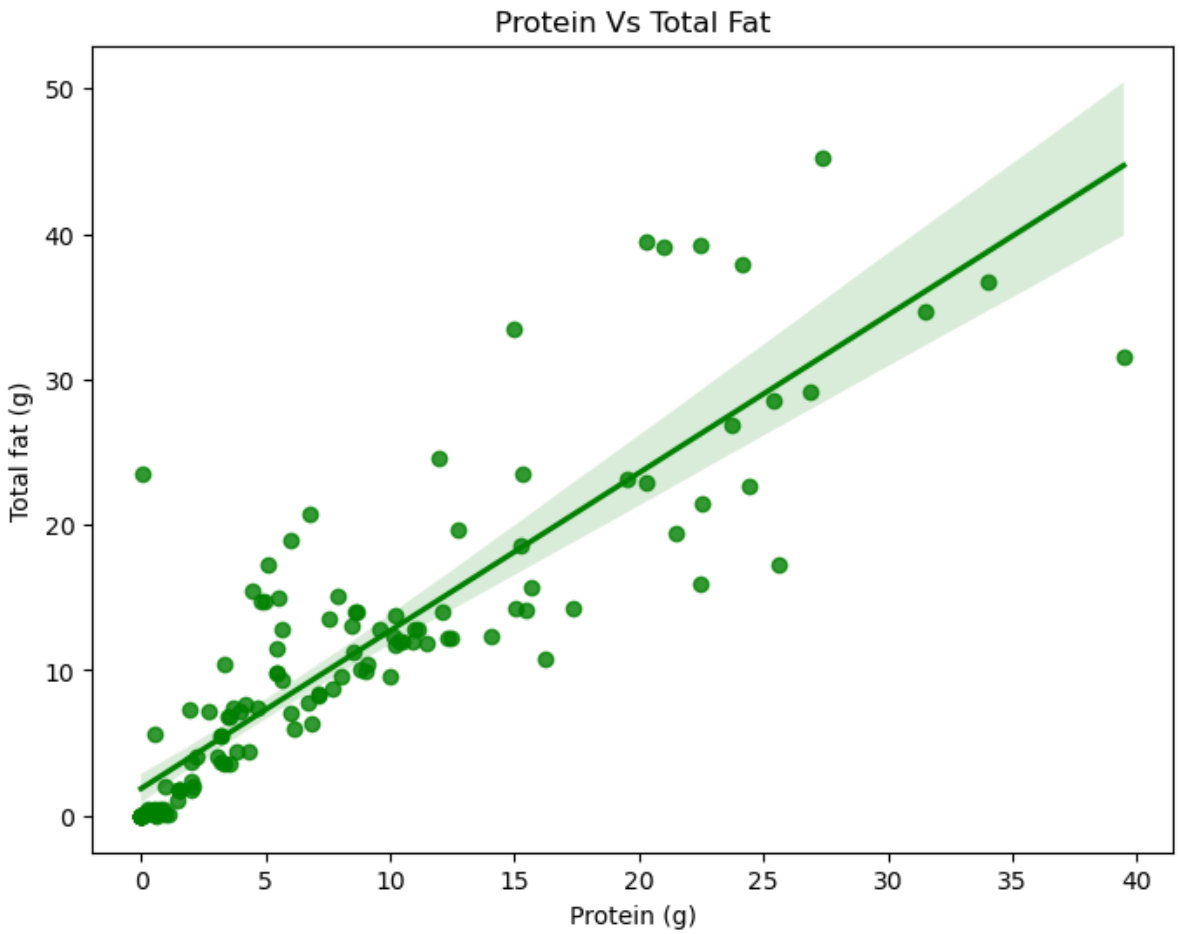
Out[18]:

	Menu Category	Menu Items	Per Serve Size	Energy (kCal)	Protein (g)	Total fat (g)	Sat Fat (g)	Trans fat (g)	Cholesterols (mg)	Total carbohydrate (g)
136	Condiments Menu	Tomato Ketchup Sachets	8 g	11.23	0.08	23.45	0.00	0.01	0.08	2.63
137	Condiments Menu	Maple Syrup	30 g	86.40	0.00	0.00	0.00	0.00	0.30	21.60
138	Condiments Menu	Cheese Slice	14 g	51.03	3.06	3.99	2.89	0.01	13.43	0.72
139	Condiments Menu	Sweet Corn	40 g	45.08	1.47	1.00	0.22	0.04	2.00	7.55
140	Condiments Menu	Mixed Fruit Beverage	180 ml	72.25	0.65	0.02	0.02	0.02	0.01	18.00

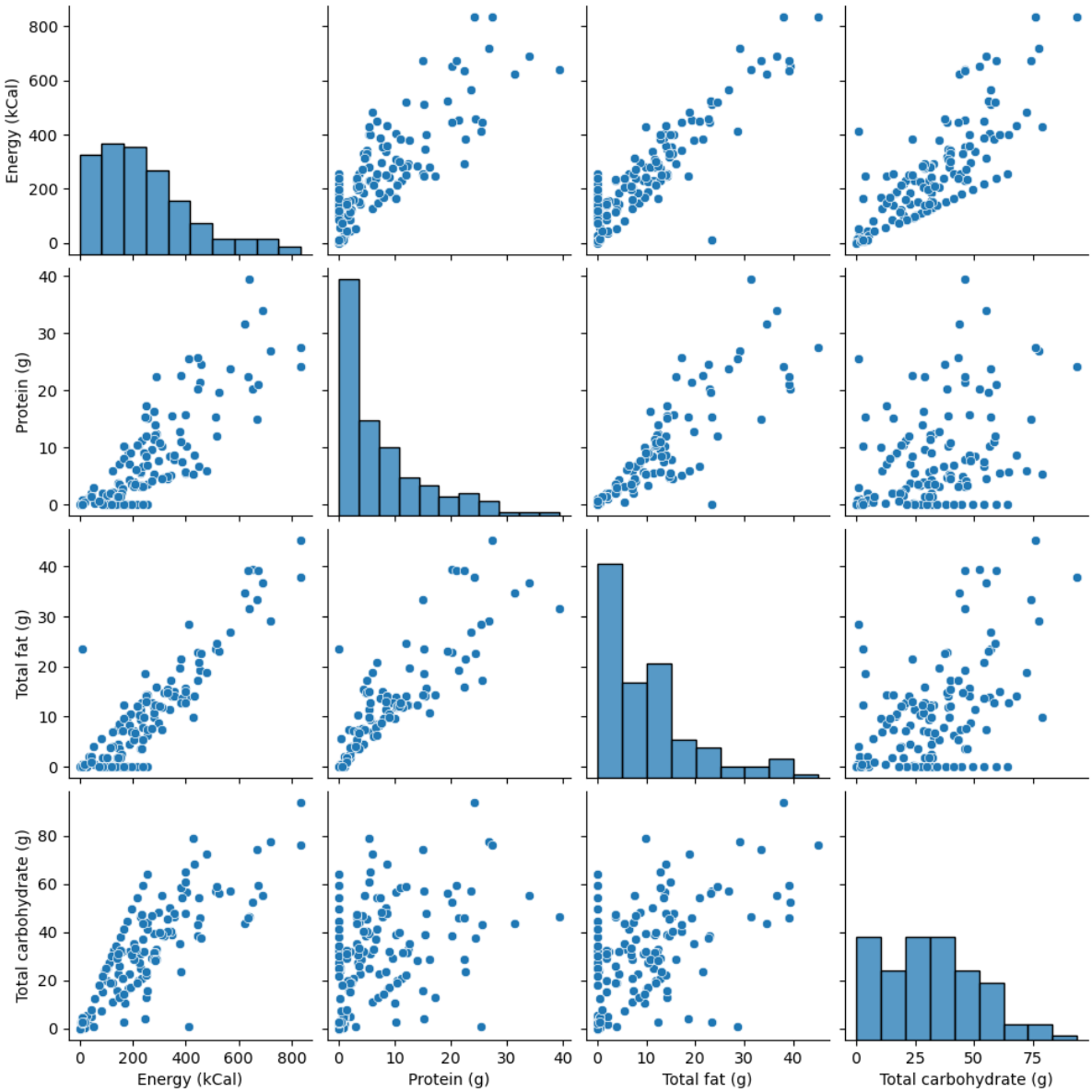
In [19]:
```
# univariate Ananlysis
# Visualize the distribution of Energy Kcal
plt.figure(figsize=(8,6))
sns.histplot(data['Energy (kCal)'],bins = 20, kde = True)
plt.title("Distribution of Energy (Kcal)")
plt.show()
```



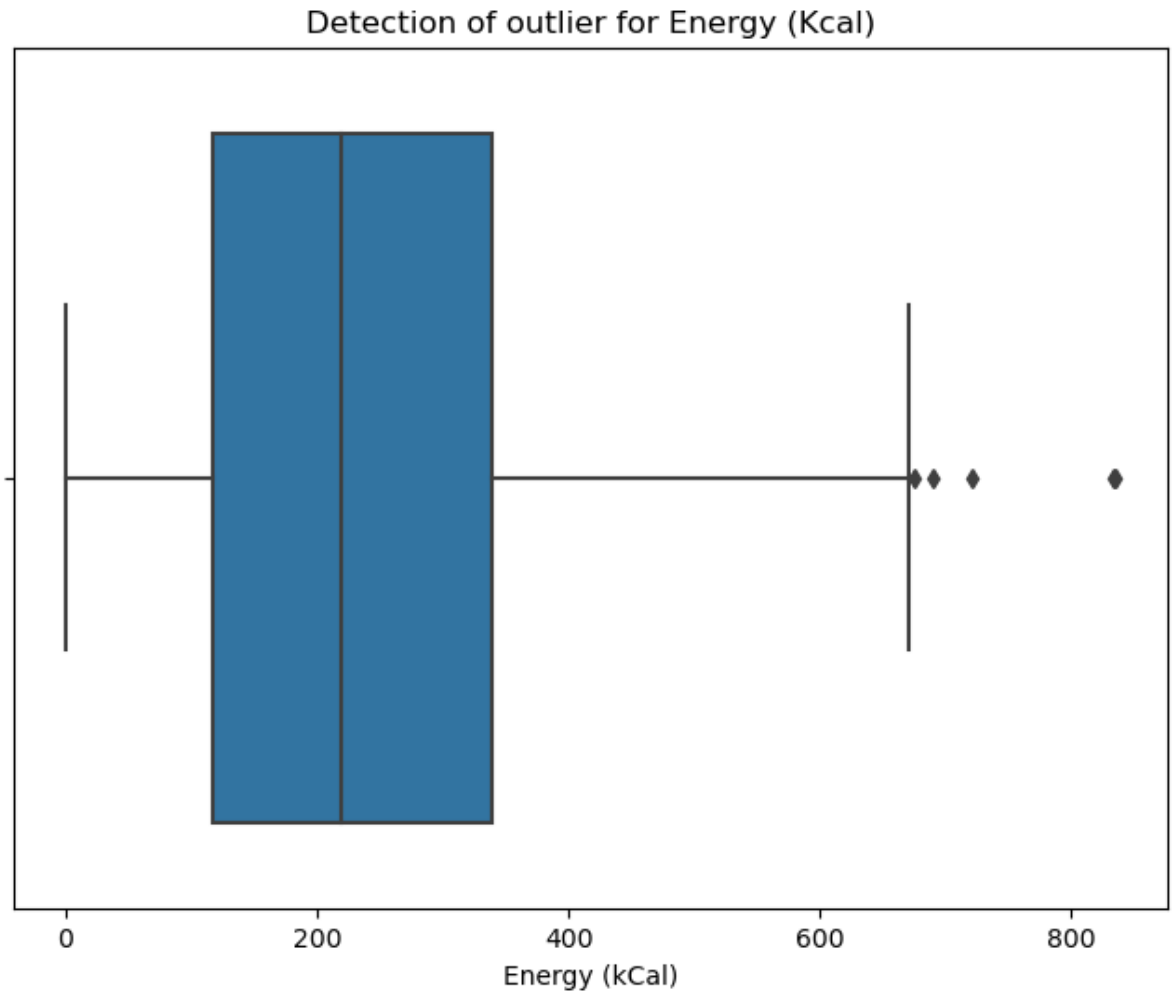
In [20]:
```
# Bivariate Analysis
# sactter plot for protein (g) vs fats (g)
plt.figure(figsize=(8,6))
sns.regplot(x=data["Protein (g)"], y=data["Total fat (g)"],color = "green")
plt.title("Protein Vs Total Fat")
plt.show()
```



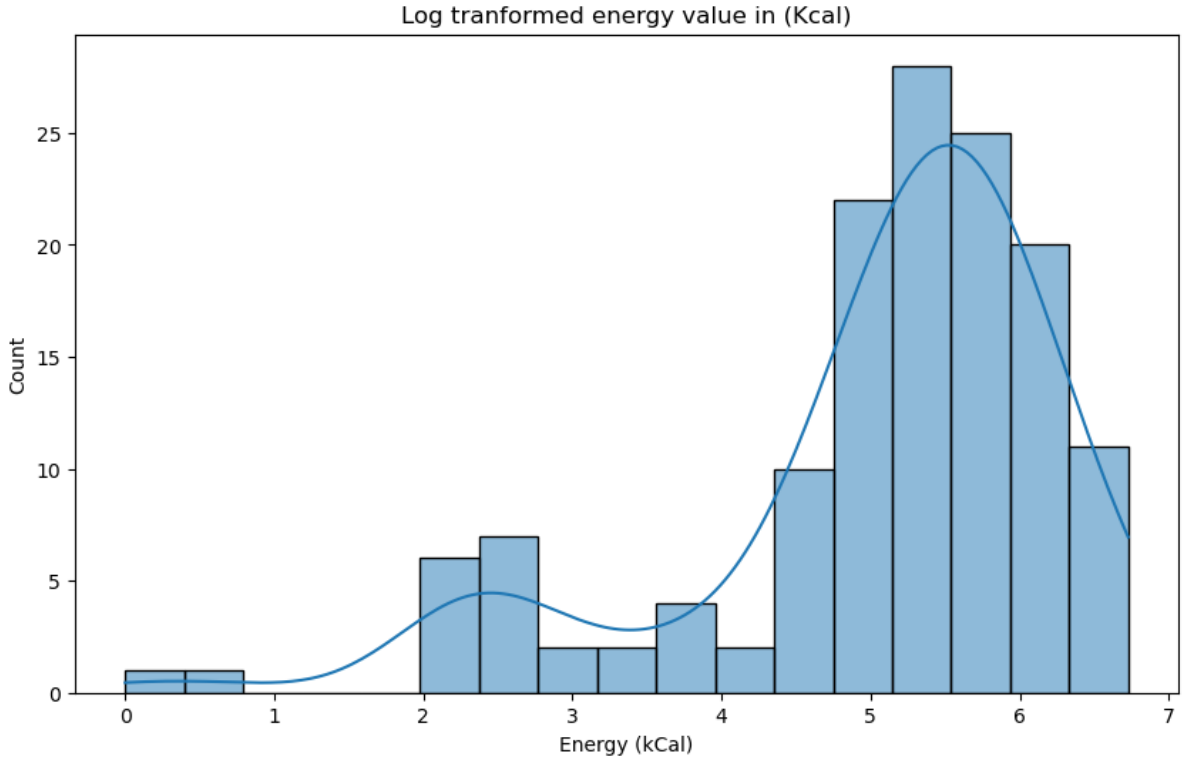
In [21]:
```
# Multivariate Analysis
# Pairplot for selected numerical values
selected_data =['Energy (kCal)', 'Protein (g)', 'Total fat (g)', 'Total carbohydrat
sns.pairplot(data[selected_data])
plt.show()
```



In [22]:
```
# Outlier Detection
# Box plot for outlier detection of Energy (Kcal)
plt.figure(figsize=(8,6))
sns.boxplot(x=data['Energy (kCal)'])
plt.title("Detection of outlier for Energy (Kcal)")
plt.show()
```



In [23]:
```
# visualize the transformed Distribution
log_energy = np.log1p(data["Energy (kCal)"])

plt.figure(figsize=(10,6))
sns.histplot(x=log_energy,kde= True)
plt.title("Log tranformed energy value in (Kcal)")
plt.show()
```



In [24]:
```
from scipy.stats import ttest_ind
# hypothesis Testing
Category_1 = data[data["Menu Items"]=='Burger']["Energy (kCal)"]
Category_2 = data[data["Menu Items"]=='Salads']['Energy (kCal)']

# Check if both have data points
if not Category_1.empty and not Category_2.empty:
    # Perform t-test
    t_stat, p_value_t = ttest_ind(Category_1,Category_2)
    print(f"T-Test: t_stat = {t_stat}, p_value = {p_value_t}")

    plt.figure(figsize=(10, 6))
    sns.boxplot(x='Menu Category', y='Energy (kCal)', data=data[data['Menu Category
    plt.title('Boxplot of Energy Content for Burgers and Salads')
    plt.show()
else:
    print("One or both categories have no data points. Unable to perform tests.")
```

```
One or both categories have no data points. Unable to perform tests.
```

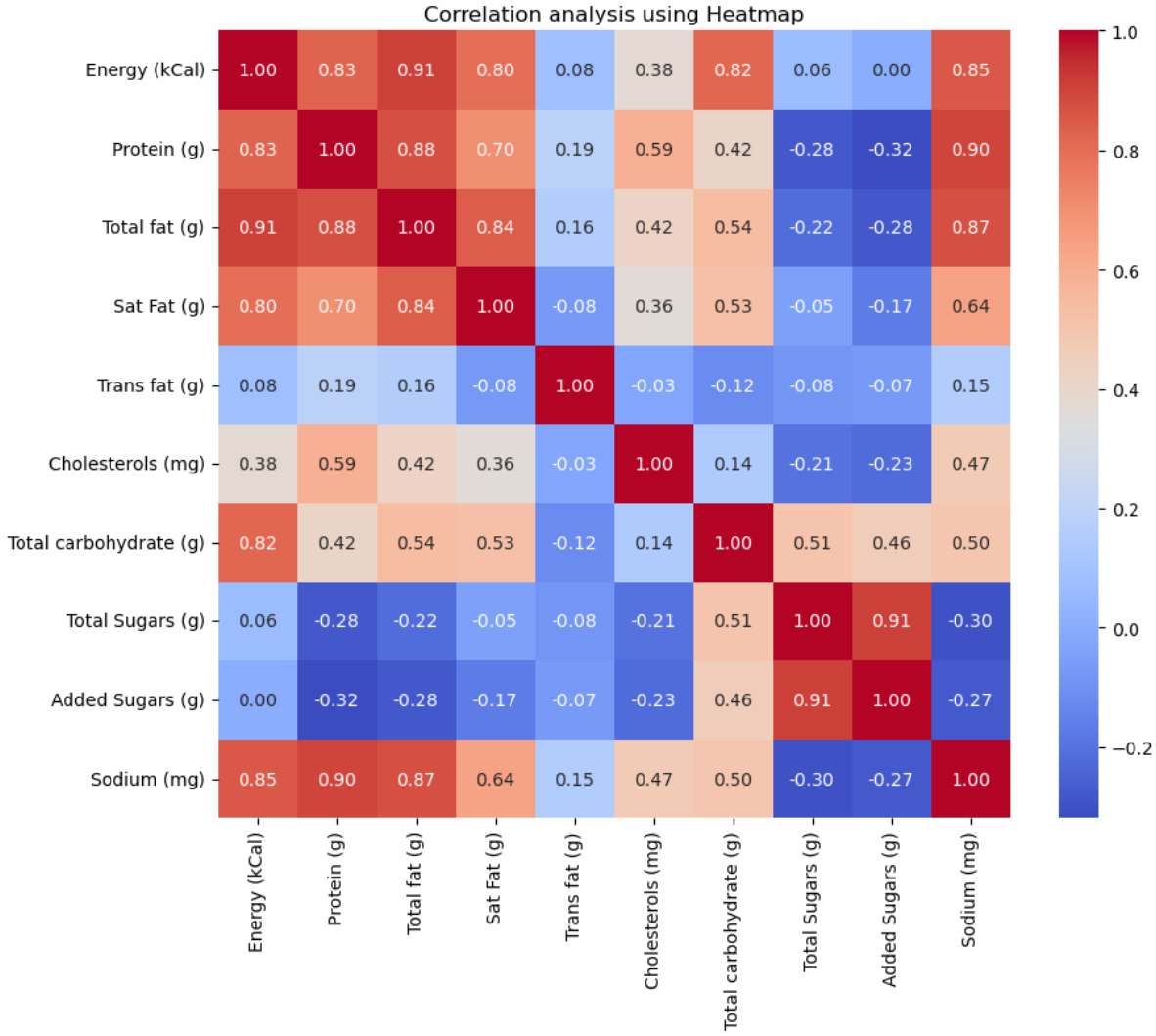
In [25]:
```
# Correlation Analysis
numeric_col = data[data.select_dtypes(include=['float64','int64']).columns]
correlation_matrix = numeric_col.corr()
print(correlation_matrix)

plt.figure(figsize=(10,8))
sns.heatmap(correlation_matrix, annot= True , cmap='coolwarm', fmt='.2f')
plt.title("Correlation analysis using Heatmap")
plt.show()
```

	Energy (kCal)	Protein (g)	Total fat (g)
Energy (kCal)	1.000000	0.826833	0.908642
Protein (g)	0.826833	1.000000	0.875594
Total fat (g)	0.908642	0.875594	1.000000
Sat Fat (g)	0.798445	0.702715	0.843381
Trans fat (g)	0.081401	0.189194	0.158400
Cholesterols (mg)	0.379387	0.590031	0.424339
Total carbohydrate (g)	0.815603	0.415217	0.538478
Total Sugars (g)	0.063306	-0.282875	-0.220125
Added Sugars (g)	0.003639	-0.319231	-0.280462
Sodium (mg)	0.851195	0.899282	0.873337

	Sat Fat (g)	Trans fat (g)	Cholesterols (mg)
Energy (kCal)	0.798445	0.081401	0.379387
Protein (g)	0.702715	0.189194	0.590031
Total fat (g)	0.843381	0.158400	0.424339
Sat Fat (g)	1.000000	-0.076431	0.363135
Trans fat (g)	-0.076431	1.000000	-0.029681
Cholesterols (mg)	0.363135	-0.029681	1.000000
Total carbohydrate (g)	0.525837	-0.123237	0.142834
Total Sugars (g)	-0.050434	-0.082297	-0.205699
Added Sugars (g)	-0.174230	-0.067124	-0.225601
Sodium (mg)	0.637510	0.154134	0.474205

	Total carbohydrate (g)	Total Sugars (g)
Energy (kCal)	0.815603	0.063306
Protein (g)	0.415217	-0.282875
Total fat (g)	0.538478	-0.220125
Sat Fat (g)	0.525837	-0.050434
Trans fat (g)	-0.123237	-0.082297
Cholesterols (mg)	0.142834	-0.205699
Total carbohydrate (g)	1.000000	0.508707
Total Sugars (g)	0.508707	1.000000
Added Sugars (g)	0.455049	0.912168
Sodium (mg)	0.498462	-0.299005

	Added Sugars (g)	Sodium (mg)
Energy (kCal)	0.003639	0.851195
Protein (g)	-0.319231	0.899282
Total fat (g)	-0.280462	0.873337
Sat Fat (g)	-0.174230	0.637510
Trans fat (g)	-0.067124	0.154134
Cholesterols (mg)	-0.225601	0.474205
Total carbohydrate (g)	0.455049	0.498462
Total Sugars (g)	0.912168	-0.299005
Added Sugars (g)	1.000000	-0.272978
Sodium (mg)	-0.272978	1.000000

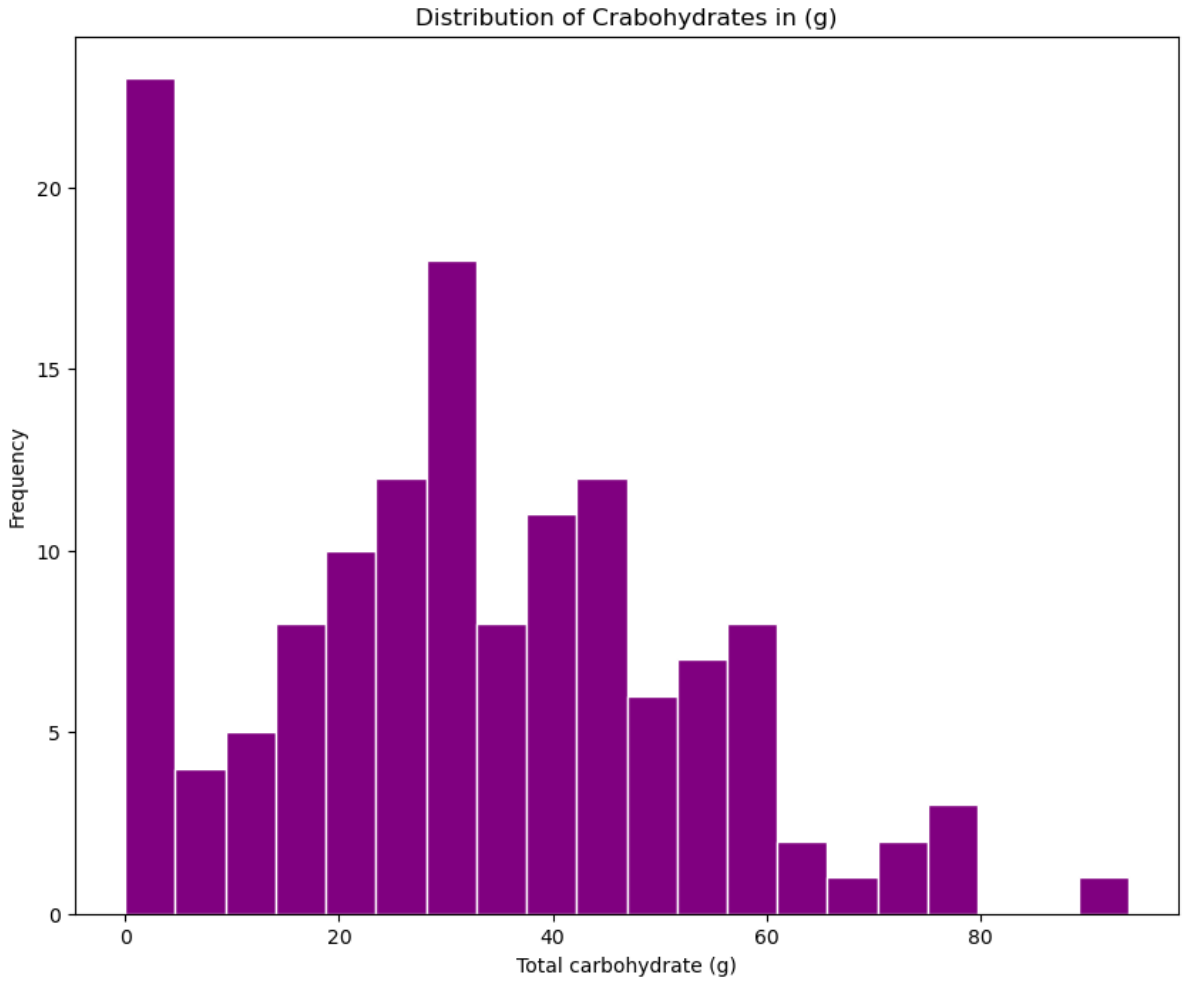


In [26]:
```
# average nutrition value
data['per serve size (g)'] = data['Per Serve Size'].str.extract('(\d+)').astype(flo
data['per serve size (g)']
data.drop(columns=['Per Serve Size'],inplace = True)
```

In [27]:
```
# highest Energy (Kcal) item in the menu
high_Kcal = data[data['Energy (kCal)']== data['Energy (kCal)'].max()]['Menu Items']
print("The menu item with higest Energy:",high_Kcal)
```

```
The menu item with higest Energy: Chicken Cheese Lava Burger
```

In [28]:
```
# Distribution of total carbohydrates across the Menu Items
plt.figure(figsize=(10,8))
plt.hist(data["Total carbohydrate (g)"],bins=20,color= 'Purple',edgecolor='white')
plt.title("Distribution of Crabohydrates in (g)")
plt.xlabel("Total carbohydrate (g)")
plt.ylabel("Frequency")
plt.show()
```



In [29]:
```
# what is the average protein content for menu category
avg_pro_by_cat= data.groupby("Menu Category")["Protein (g)"].mean()
print("The average protein content for menu category:",avg_pro_by_cat)
```

```
The average protein content for menu category: Menu Category
Beverages Menu      0.268235
Breakfast Menu      7.636667
Condiments Menu     0.731111
Desserts Menu       2.815000
Gourmet Menu       21.684545
McCafe Menu         4.295490
Regular Menu       12.990833
Name: Protein (g), dtype: float64
```

In [30]:
```
# which category has the highest energy content ?
hig_energy_content = data.groupby('Menu Category')['Energy (kCal)'].mean().idxmax()
print("The Category with highest energy content is:",hig_energy_content)
```

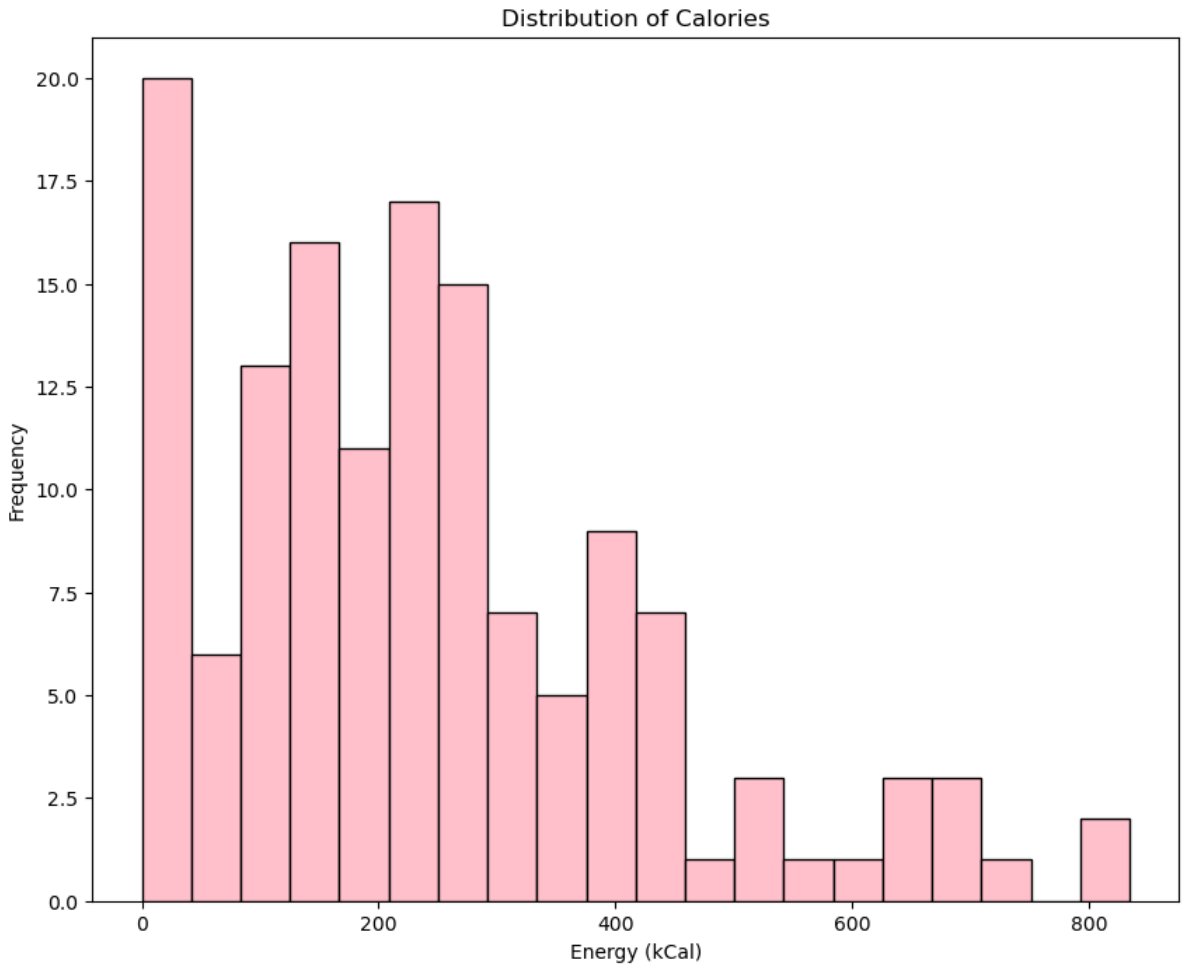
```
The Category with highest energy content is: Gourmet Menu
```

In [31]:
```
# Correlation between Total Fat and Protein
Corr=data['Protein (g)'].corr(data['Total fat (g)'])
print("Correlation between Total Fat and Protein:",Corr)
```

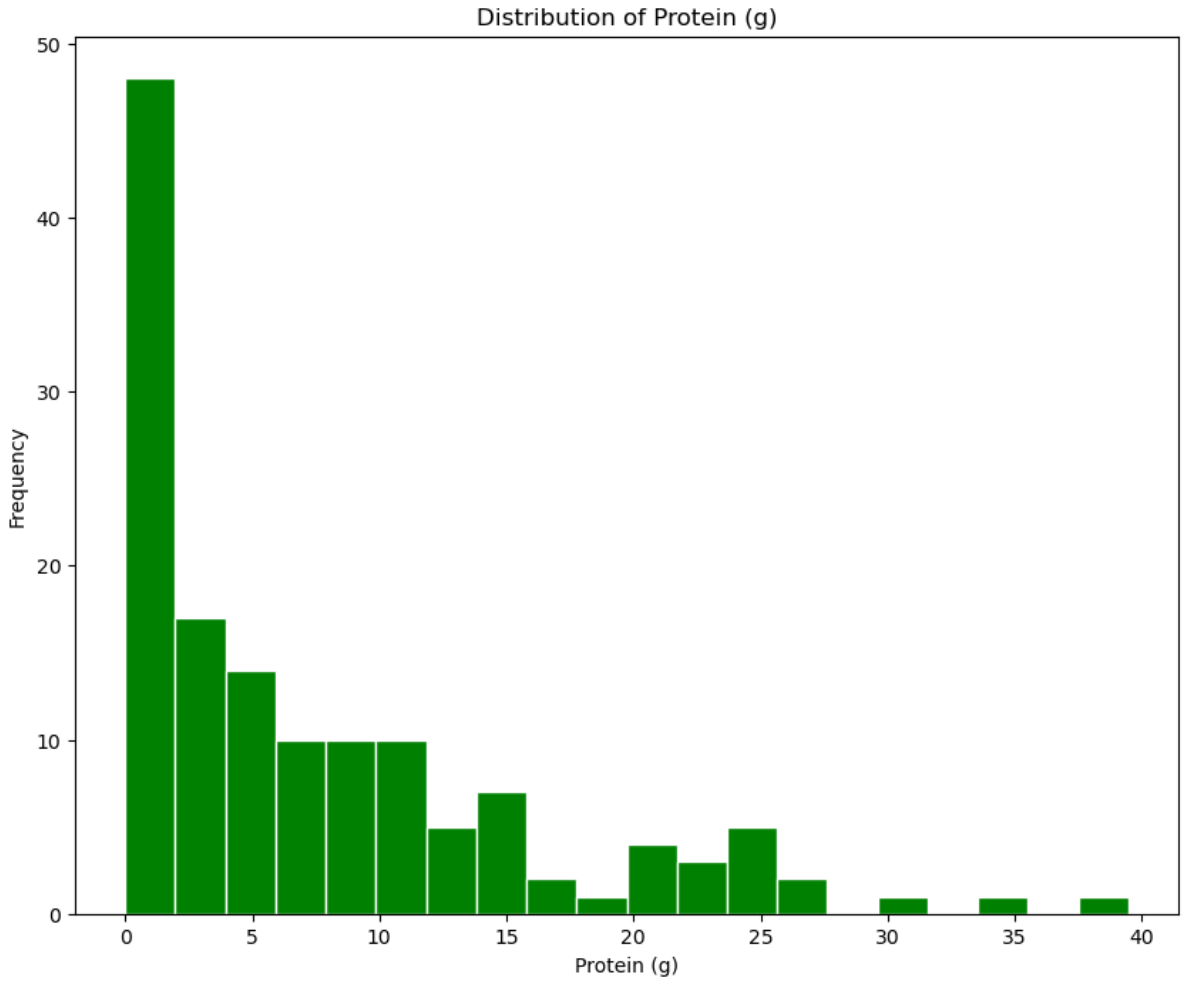
```
Correlation between Total Fat and Protein: 0.8755938053642127
```

In [32]:
```
# Distribution of calorie values across menu items
print(data['Energy (kCal)'].describe())
plt.figure(figsize=(10,8))
plt.hist(data['Energy (kCal)'],bins=20,color="Pink",edgecolor='Black')
plt.title("Distribution of Calories")
plt.xlabel("Energy (kCal)")
plt.ylabel("Frequency")
plt.show()
```

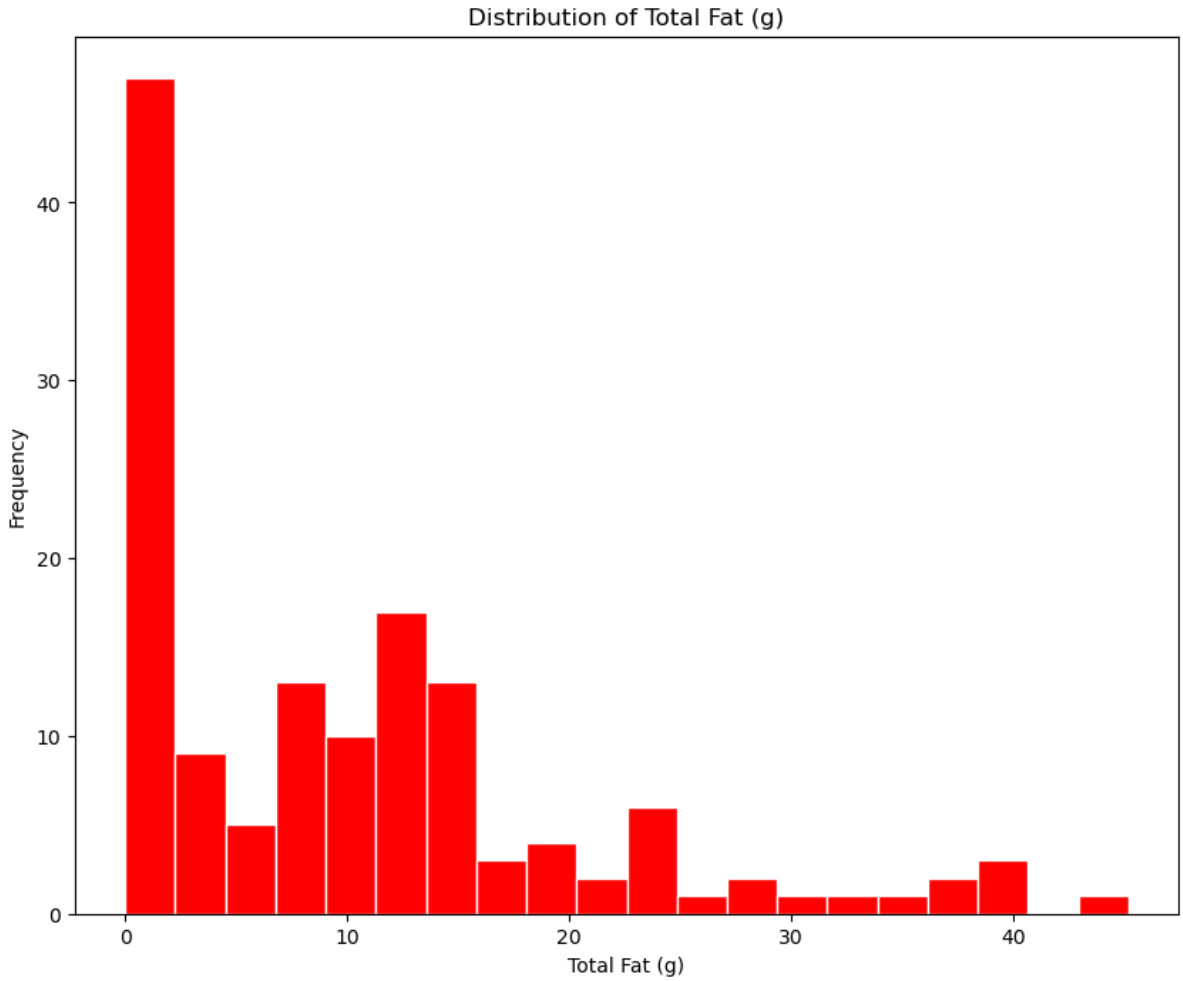
```
count    141.000000
mean     244.635461
std      185.554837
min        0.000000
25%      116.360000
50%      219.360000
75%      339.520000
max      834.360000
Name: Energy (kCal), dtype: float64
```



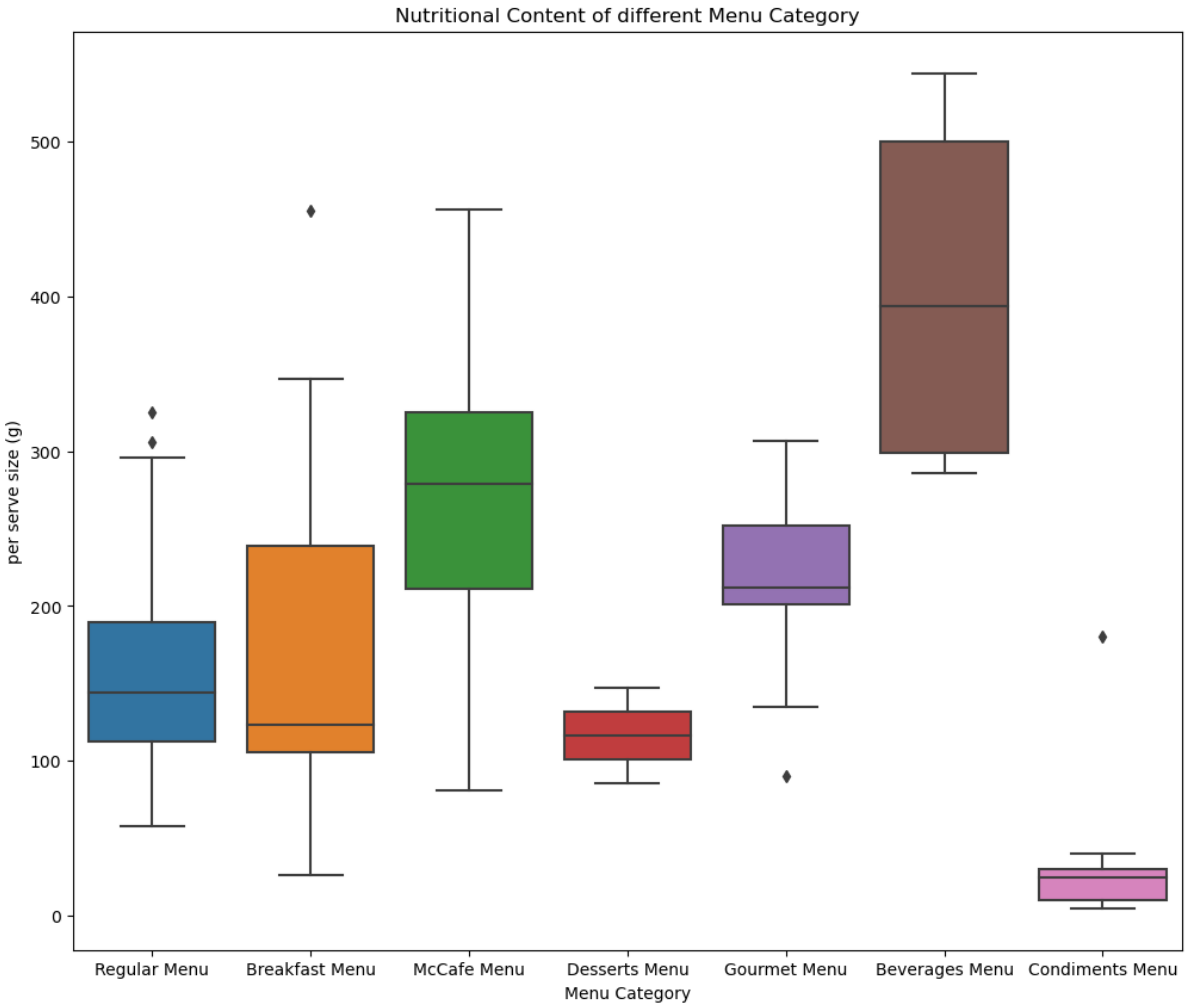
In [33]:
```
# Distribution of Protein Content
plt.figure(figsize=(10,8))
plt.hist(data['Protein (g)'],bins=20,color='green',edgecolor='white')
plt.title("Distribution of Protein (g)")
plt.xlabel("Protein (g)")
plt.ylabel("Frequency")
plt.show()
```



In [34]:
```
# Distribution of Total Fat Content
plt.figure(figsize=(10,8))
plt.hist(data['Total fat (g)'],bins=20,color='red',edgecolor='white')
plt.title("Distribution of Total Fat (g)")
plt.xlabel("Total Fat (g)")
plt.ylabel("Frequency")
plt.show()
```

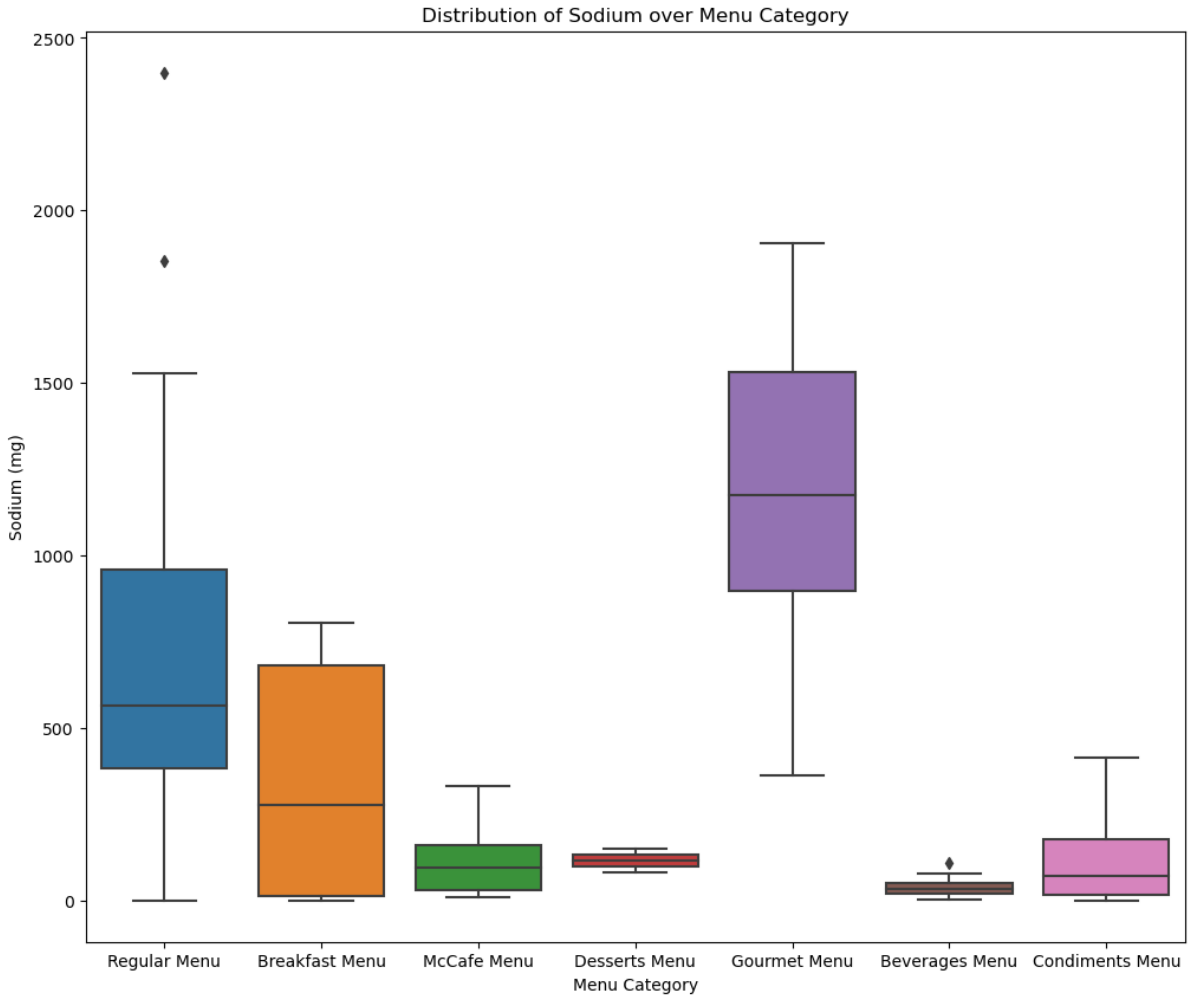


In [35]:
```
# comparing the nutritional content of different menu categories
plt.figure(figsize=(12,10))
sns.boxplot(x=data['Menu Category'],y=data['per serve size (g)'])
plt.title("Nutritional Content of different Menu Category")
plt.show()
```

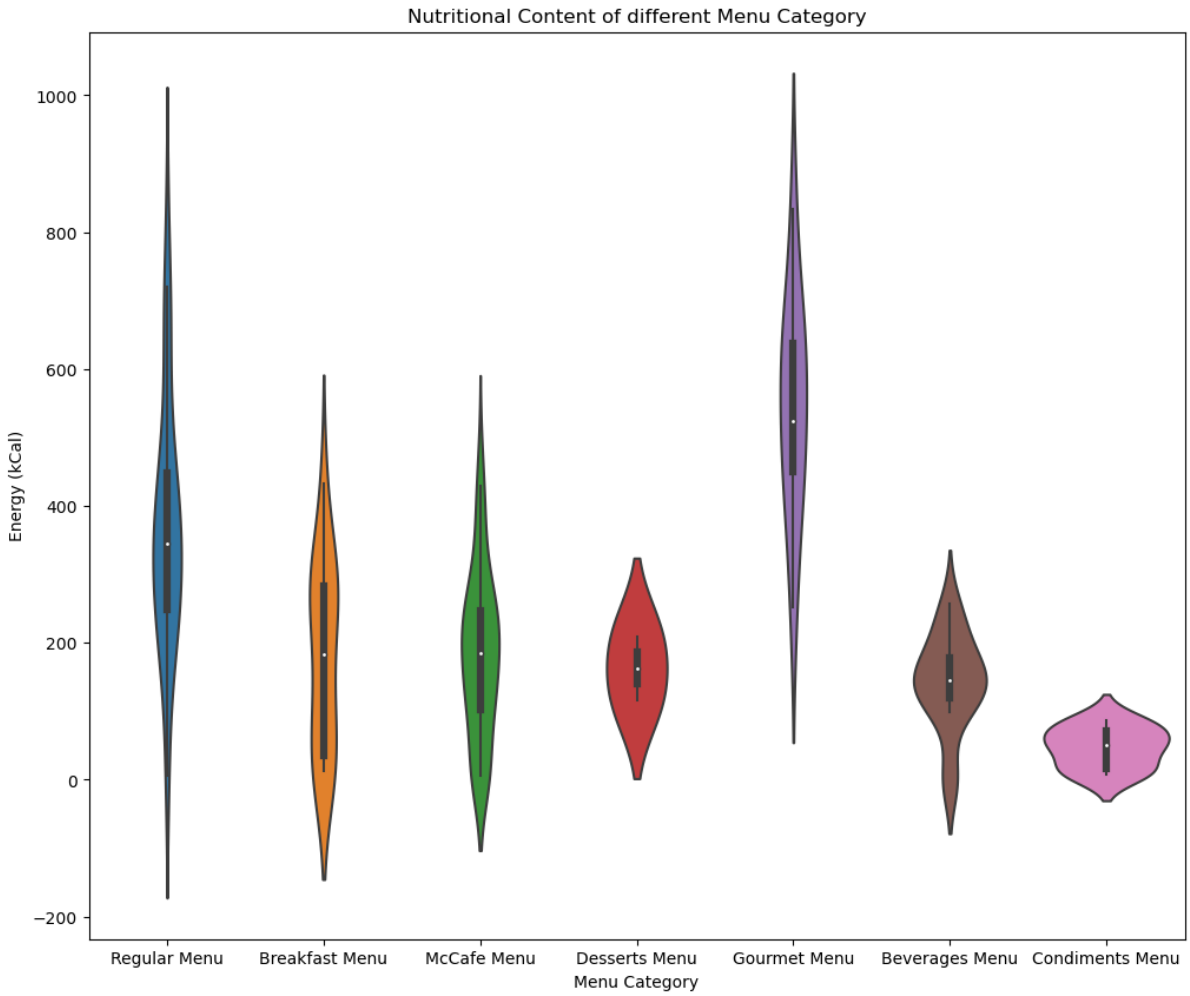


In [36]:
```
# sodium content vary across menu categories
plt.figure(figsize=(12,10))
sns.boxplot(x=data['Menu Category'],y=data['Sodium (mg)'])
plt.title("Distribution of Sodium over Menu Category")
```

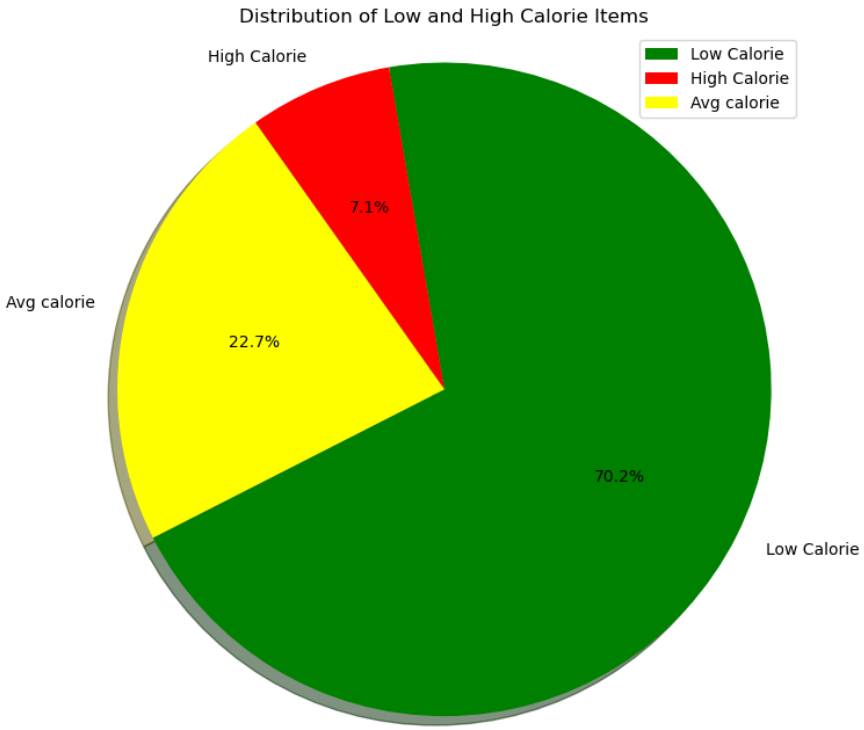
Out[36]:
```
Text(0.5, 1.0, 'Distribution of Sodium over Menu Category')
```



In [37]:
```
# the distribution of calories for different menu categories
plt.figure(figsize=(12,10))
sns.violinplot(x=data['Menu Category'],y=data['Energy (kCal)'])
plt.title("Nutritional Content of different Menu Category")
plt.show()
```

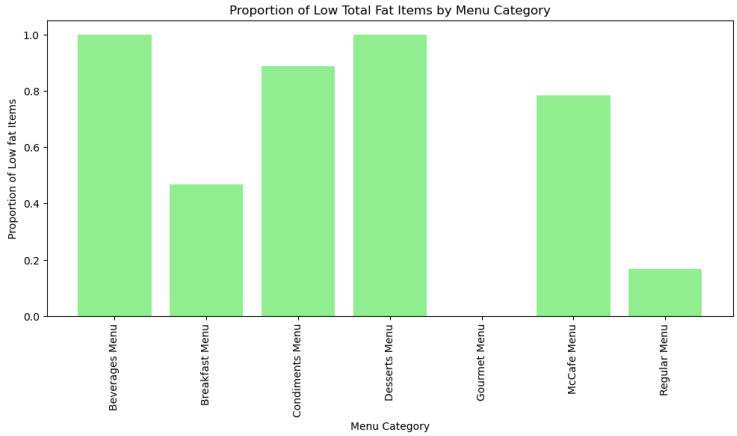

Nutritional Content of different Menu Category
Energy (kCal)
1000
800
600
400
200
0
−200
Regular Menu
Breakfast Menu
McCafe Menu
Desserts Menu
Gourmet Menu
Beverages Menu
Condiments Menu
Menu Category

In [38]:
```
# How do the distribution of menu items with low calorie content compare to those w
# What conclusions can we draw for health-conscious individuals

# Threshold values for Calories
low_calorie_threshold=300
high_calorie_threshold=600

# filtered Calories data
low_calorie_item = data[data['Energy (kCal)']<low_calorie_threshold].shape[0]
high_calorie_item = data[data['Energy (kCal)']>high_calorie_threshold].shape[0]
avg_calorie_item = data[(data['Energy (kCal)']>low_calorie_threshold) & (data['Ener

# Creating Data for Pie-Chart Analysis
labels=['Low Calorie','High Calorie','Avg calorie']
sizes=[low_calorie_item,high_calorie_item,avg_calorie_item]
colors=['green','red','yellow']

# Data Visualization
plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=207,shad
plt.title('Distribution of Low and High Calorie Items')
plt.legend(loc='best')
plt.axis('equal')

# Conclusion
if low_calorie_item > high_calorie_item and low_calorie_item >avg_calorie_item :
    conclusion="There are more menu items with low calorie content, which is benefi
elif high_calorie_item > low_calorie_item and high_calorie_item > avg_calorie_item:
    conclusion="There are more menu items with high calorie content, indicating a n
else:
    conclusion="There is a balanced distribution of menu items with low and high ca

# Display the Conclusion
plt.text(0.5, -1.2, conclusion, ha='center', va='center', fontsize=12,fontweight='b
plt.show()
```



There are more menu items with low calorie content, which is beneficial for health-conscious individuals.

In [39]:
```
# Which menu category offers the highest proportion of items with low total fat con
# How can this information guide individuals towards healthier menu options.
Low_total_fat_threshold = 10
low_fat_Items = data[data['Total fat (g)']<Low_total_fat_threshold]
category_low_portion = low_fat_Items.groupby('Menu Category').size()/data.groupby('
Healthiest_Category= category_low_portion.idxmax()

plt.figure(figsize=(10,6))
plt.bar(category_low_portion.index,category_low_portion.values,color='lightgreen')
plt.title('Proportion of Low Total Fat Items by Menu Category')
plt.xlabel('Menu Category')
plt.ylabel('Proportion of Low fat Items')
plt.xticks(rotation=90)
plt.tight_layout()

conclusion = f"The '{Healthiest_Category}' category offers the highest proportion o
plt.text(0.5, -0.6, conclusion, ha='center', va='center', fontsize=12,fontweight='b
plt.show()
```



The 'Beverages Menu' category offers the highest proportion of items with low total fat content, making it a healthier choice for individuals concerned about fat intake.