McDonald's Menu Nutritional Analysis – Project

2.1 Data Preprocessing: Load and inspect the dataset.

[1]:
```
import pandas as pd

# Load the dataset

df =pd.read_csv(r"C:\Users\ravi.kumar\Desktop\Upgrad Class\Final Projects\Nutrical Dataset.csv")

# Display the first five rows of the dataset
df.head()
```

[1]:

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...	Carbohydrates	Carbohydrates (% Daily Value)	Dietary Fiber	Dietary Fiber (% Daily Value)	Sugars	Protein	Vitamin A (% Daily Value)	Vita C D Va
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0.0	...	31	10	4	17	3	17	10	
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0.0	...	30	10	4	17	3	18	6	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0	...	29	10	4	17	2	14	8	
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0.0	...	30	10	4	17	2	21	15	
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	42	0.0	...	30	10	4	17	2	21	6	

5 rows × 24 columns

[2]:
```
print(df.head())
```

```
   Category                         Item    Serving Size  Calories  \
0  Breakfast                Egg McMuffin  4.8 oz (136 g)       300
1  Breakfast            Egg White Delight  4.8 oz (135 g)       250
2  Breakfast             Sausage McMuffin  3.9 oz (111 g)       370
3  Breakfast      Sausage McMuffin with Egg  5.7 oz (161 g)       450
4  Breakfast  Sausage McMuffin with Egg Whites  5.7 oz (161 g)       400

   Calories from Fat  Total Fat  Total Fat (% Daily Value)  Saturated Fat  \
0                120       13.0                         20            5.0
1                 70        8.0                         12            3.0
2                200       23.0                         35            8.0
3                250       28.0                         43           10.0
4                210       23.0                         35            8.0

   Saturated Fat (% Daily Value)  Trans Fat  ...  Carbohydrates  \
0                             25        0.0  ...             31
1                             15        0.0  ...             30
2                             42        0.0  ...             29
3                             52        0.0  ...             30
4                             42        0.0  ...             30

   Carbohydrates (% Daily Value)  Dietary Fiber  \
0                             10              4
1                             10              4
2                             10              4
3                             10              4
4                             10              4

   Dietary Fiber (% Daily Value)  Sugars  Protein  Vitamin A (% Daily Value)  \
0                             17       3       17                         10
1                             17       3       18                          6
2                             17       2       14                          8
3                             17       2       21                         15
4                             17       2       21                          6

   Vitamin C (% Daily Value)  Calcium (% Daily Value)  Iron (% Daily Value)
0                          0                       25                    15
1                          0                       25                     8
2                          0                       25                    10
3                          0                       30                    15
4                          0                       25                    10

[5 rows x 24 columns]
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 260 entries, 0 to 259
Data columns (total 24 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Category                        260 non-null    object
 1   Item                            260 non-null    object
 2   Serving Size                    260 non-null    object
 3   Calories                        260 non-null    int64
 4   Calories from Fat               260 non-null    int64
 5   Total Fat                       260 non-null    float64
 6   Total Fat (% Daily Value)       260 non-null    int64
 7   Saturated Fat                   260 non-null    float64
 8   Saturated Fat (% Daily Value)   260 non-null    int64
 9   Trans Fat                       260 non-null    float64
 10  Cholesterol                     260 non-null    int64
 11  Cholesterol (% Daily Value)     260 non-null    int64
 12  Sodium                          260 non-null    int64
 13  Sodium (% Daily Value)          260 non-null    int64
 14  Carbohydrates                   260 non-null    int64
 15  Carbohydrates (% Daily Value)   260 non-null    int64
 16  Dietary Fiber                   260 non-null    int64
 17  Dietary Fiber (% Daily Value)   260 non-null    int64
 18  Sugars                          260 non-null    int64
 19  Protein                         260 non-null    int64
 20  Vitamin A (% Daily Value)       260 non-null    int64
 21  Vitamin C (% Daily Value)       260 non-null    int64
 22  Calcium (% Daily Value)         260 non-null    int64
 23  Iron (% Daily Value)            260 non-null    int64
dtypes: float64(3), int64(18), object(3)
memory usage: 48.9+ KB
```

[7]:
```
len(df)
```

[7]:
```
260
```

[8]:
```
df.columns
```

[8]:
```
Index(['Category', 'Item', 'Serving Size', 'Calories', 'Calories from Fat',
       'Total Fat', 'Total Fat (% Daily Value)', 'Saturated Fat',
       'Saturated Fat (% Daily Value)', 'Trans Fat', 'Cholesterol',
       'Cholesterol (% Daily Value)', 'Sodium', 'Sodium (% Daily Value)',
       'Carbohydrates', 'Carbohydrates (% Daily Value)', 'Dietary Fiber',
       'Dietary Fiber (% Daily Value)', 'Sugars', 'Protein',
       'Vitamin A (% Daily Value)', 'Vitamin C (% Daily Value)',
       'Calcium (% Daily Value)', 'Iron (% Daily Value)'],
      dtype='object')
```

[9]:
```
len(df.columns)
```

[9]:
```
24
```

2.2 Data Preprocessing: Handle missing values and data cleaning if necessary.

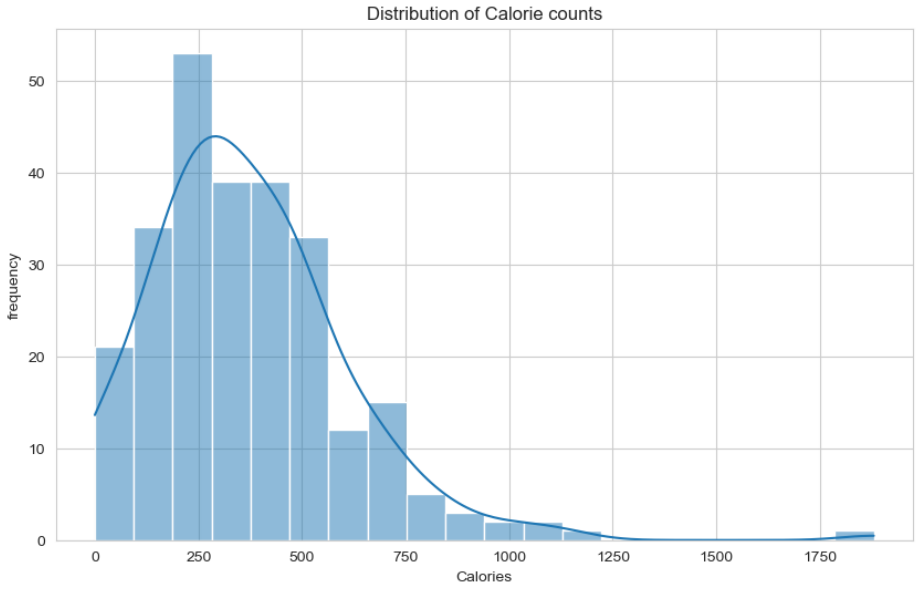
[11]:
```
# check for the missing values
df.isnull().sum()
```

[11]:
```
Category                        0
Item                            0
Serving Size                    0
Calories                        0
Calories from Fat               0
Total Fat                       0
Total Fat (% Daily Value)       0
Saturated Fat                   0
Saturated Fat (% Daily Value)   0
Trans Fat                       0
Cholesterol                     0
Cholesterol (% Daily Value)     0
Sodium                          0
Sodium (% Daily Value)          0
Carbohydrates                   0
Carbohydrates (% Daily Value)   0
Dietary Fiber                   0
Dietary Fiber (% Daily Value)   0
Sugars                          0
Protein                         0
Vitamin A (% Daily Value)       0
Vitamin C (% Daily Value)       0
Calcium (% Daily Value)         0
Iron (% Daily Value)            0
dtype: int64
```

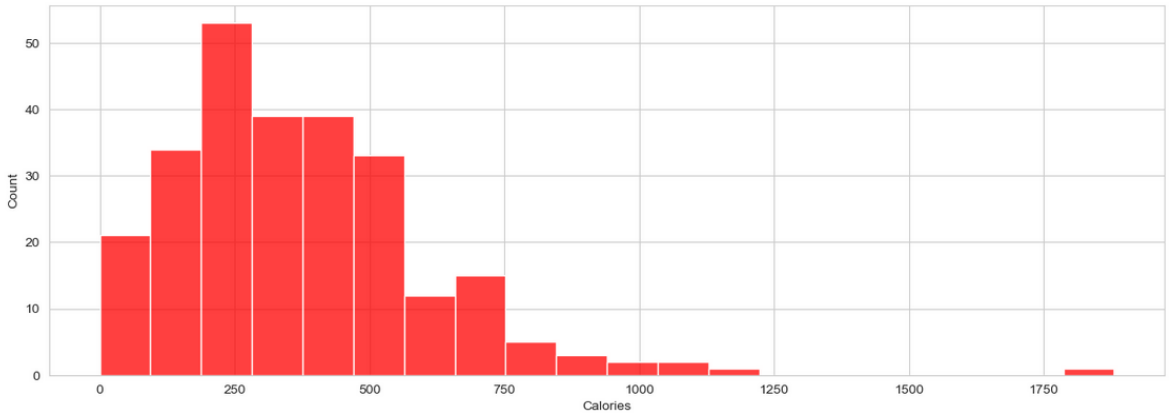
No null values found.

3.1 Exploratory Data Analysis (EDA): Analyze the distribution of calorie counts across menu items.

[29]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# distribusion of calories counts
plt.figure(figsize=(10, 6))
sns.histplot(df['Calories'], bins=20, kde=True)
plt.title('Distribution of Calorie counts')
plt.xlabel('Calories')
plt.ylabel('frequency')
plt.show
```

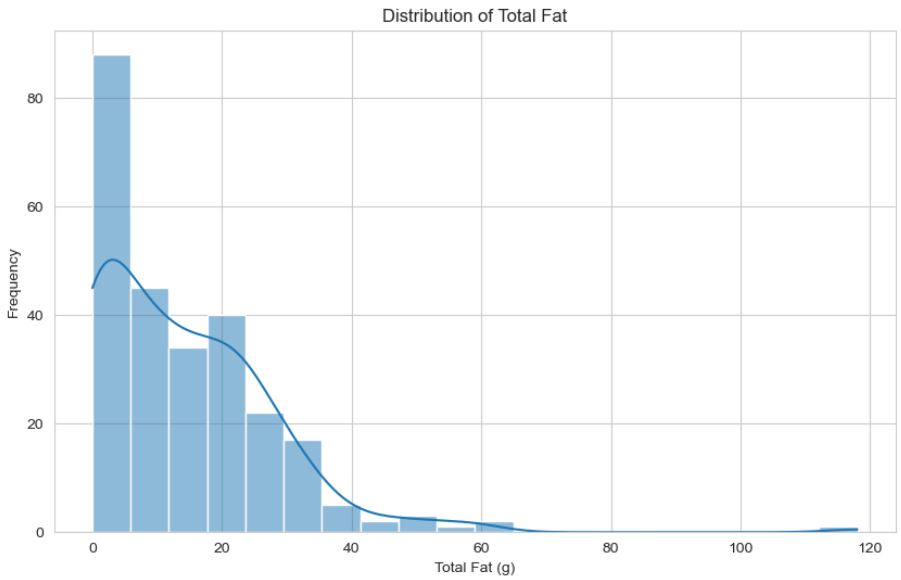


[30]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create a figure and axis object
fig, ax = plt.subplots(figsize=(15, 5))

# Create a histogram using seaborn's histplot function
sns.histplot(x='Calories', data=df, ax=ax, color='red', bins=20)

# Show the plot
plt.show()
```

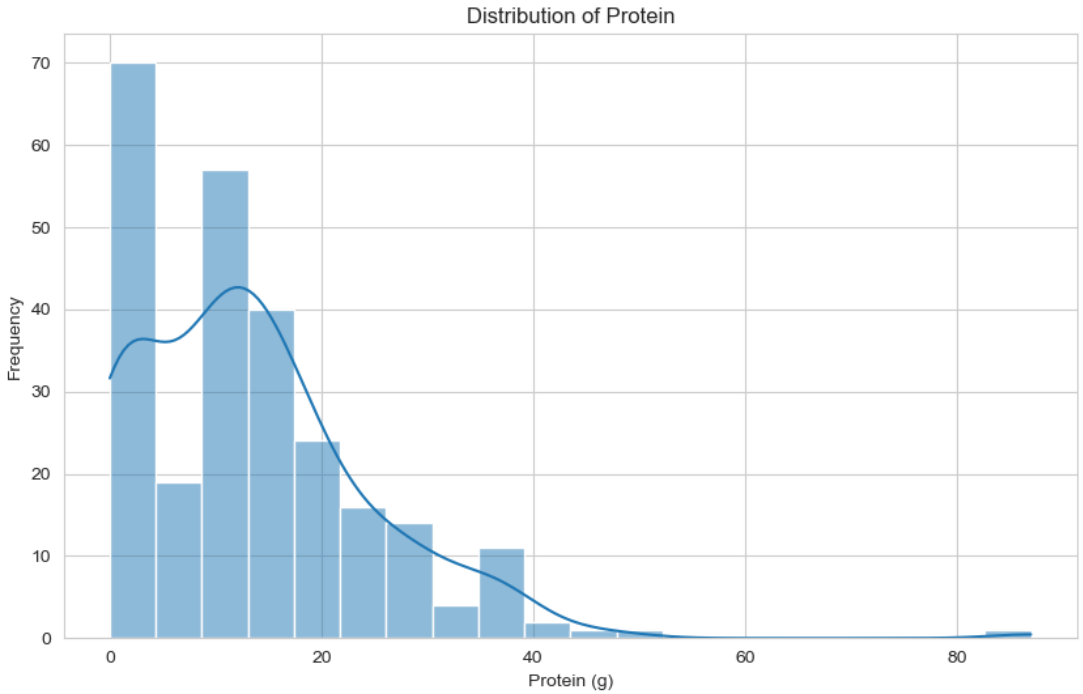


3.2 Exploratory Data Analysis (EDA): Explore the nutritional content (e.g., fat, protein, carbohydrates) of different items.

:
```
# Distribution of fat content
plt.figure(figsize=(10, 6))
sns.histplot(df['Total Fat'], bins=20, kde=True)
plt.title('Distribution of Total Fat')
plt.xlabel('Total Fat (g)')
plt.ylabel('Frequency')
plt. show()
```

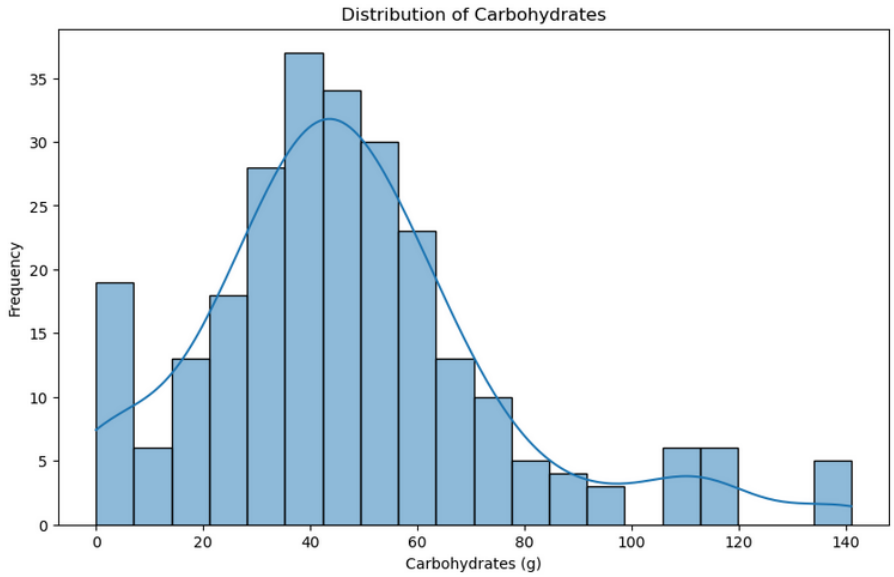


[32]:
```
# Distribution of Protein content
plt.figure(figsize=(10, 6))
sns.histplot(df['Protein'], bins=20, kde=True)
plt.title('Distribution of Protein')
plt.xlabel('Protein (g)')
plt.ylabel('Frequency')
plt. show()
```



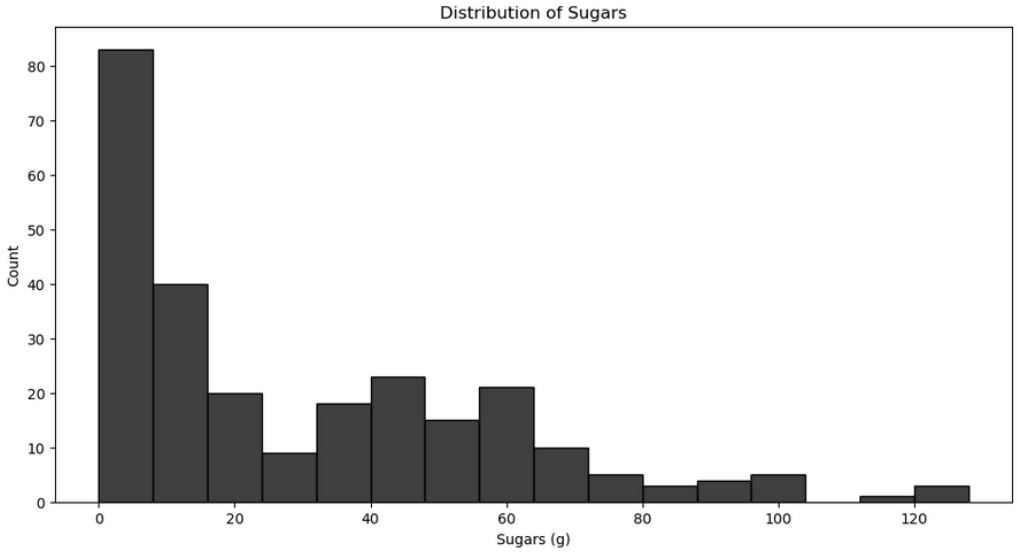
```
# Distribution of Carbohydrates content
plt.figure(figsize=(10, 6))
sns.histplot(df['Carbohydrates'], bins=20, kde=True)
plt. title('Distribution of Carbohydrates')
plt.xlabel('Carbohydrates (g)')
plt.ylabel('Frequency')
plt.show()
```

```
C[redacted]FutureWarning: use_inf_as_na option is deprecated and
will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

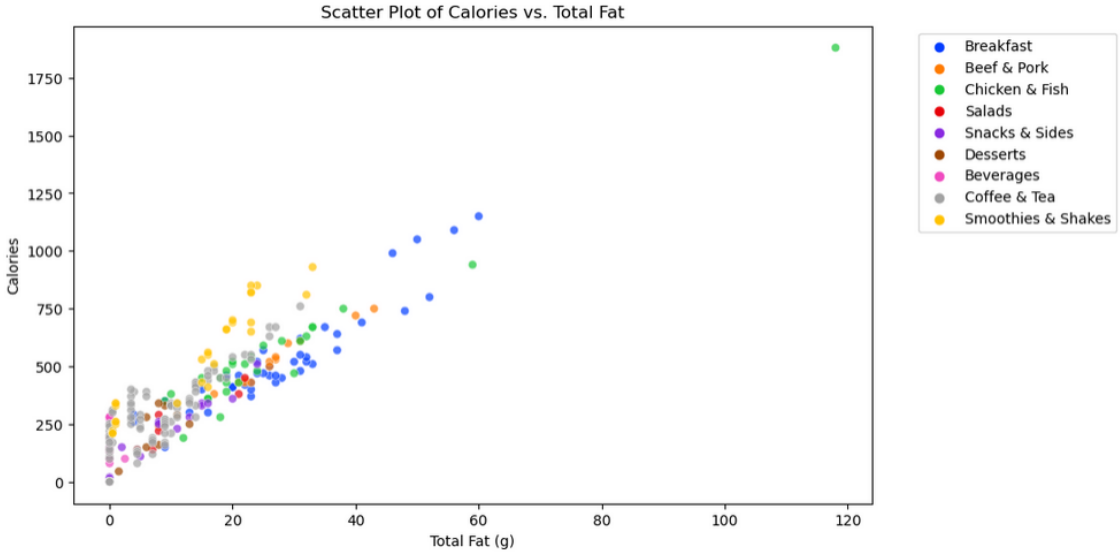


[14]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create a figure and axis object
fig, ax = plt.subplots(figsize=(12, 6))

# Create a histogram using seaborn's histplot function
sns.histplot(x='Sugars', data=df, ax=ax, color='black', bins=16)

ax.set_title('Distribution of Sugars')
ax.set_xlabel('Sugars (g)')
ax.set_ylabel('Count')
plt.show()
```

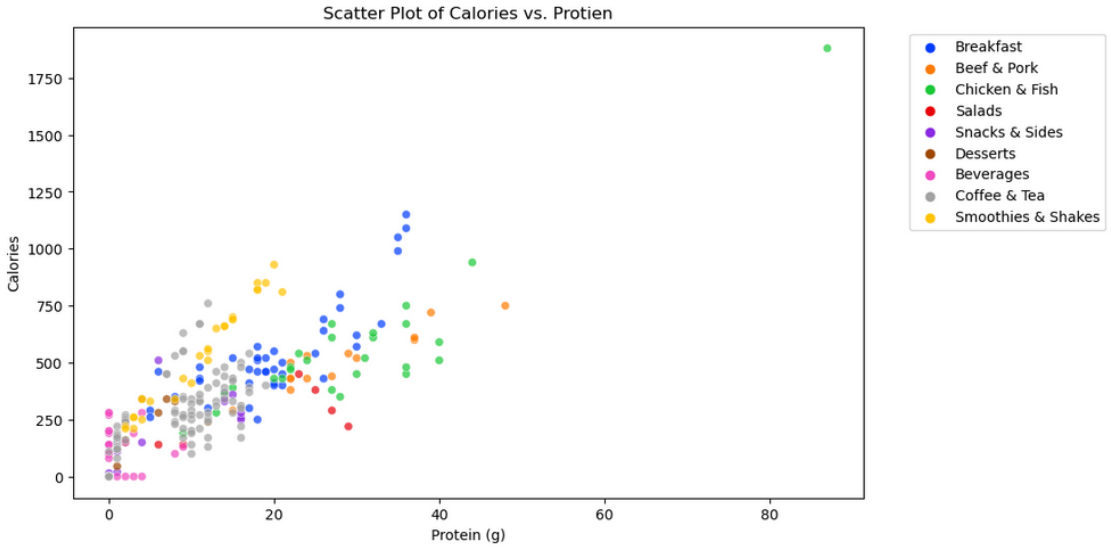
```
[redacted]FutureWarning: use_inf_as_na option is deprecated and
will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



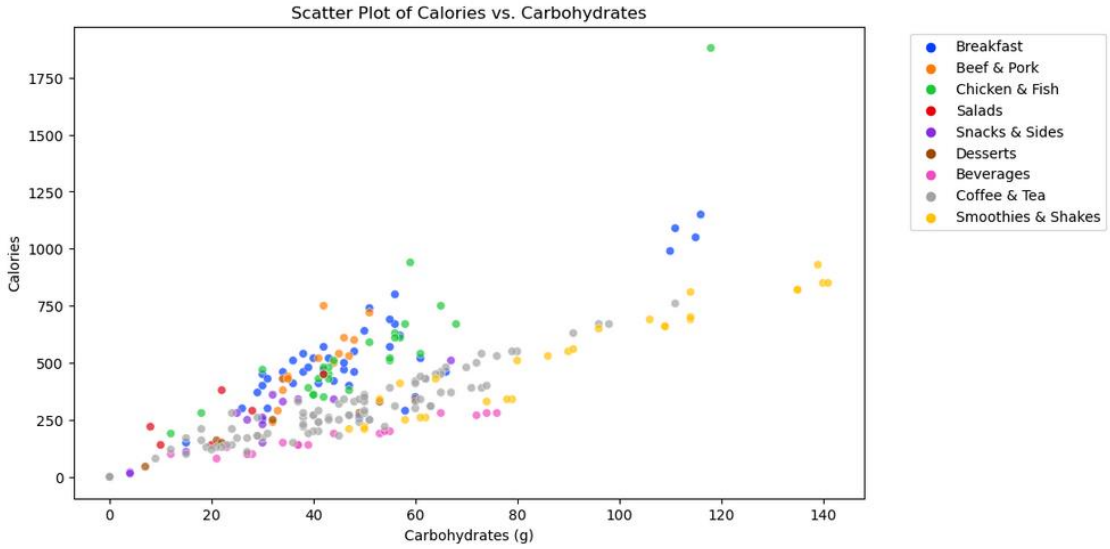
[15]:
```
# Scatter Plot of Calories vs. Total Fat
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Total Fat', y='Calories', data=df, hue='Category', palette='bright', alpha=0.7)
plt. title('Scatter Plot of Calories vs. Total Fat')
plt.xlabel('Total Fat (g)')
plt.ylabel('Calories')
plt. legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



[16]:
```
# Scatter Plot of Calories vs. Protein
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Protein', y='Calories', data=df, hue='Category', palette='bright', alpha=0.7)
plt.title('Scatter Plot of Calories vs. Protien')
plt.xlabel('Protein (g)')
plt.ylabel('Calories')
plt. legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

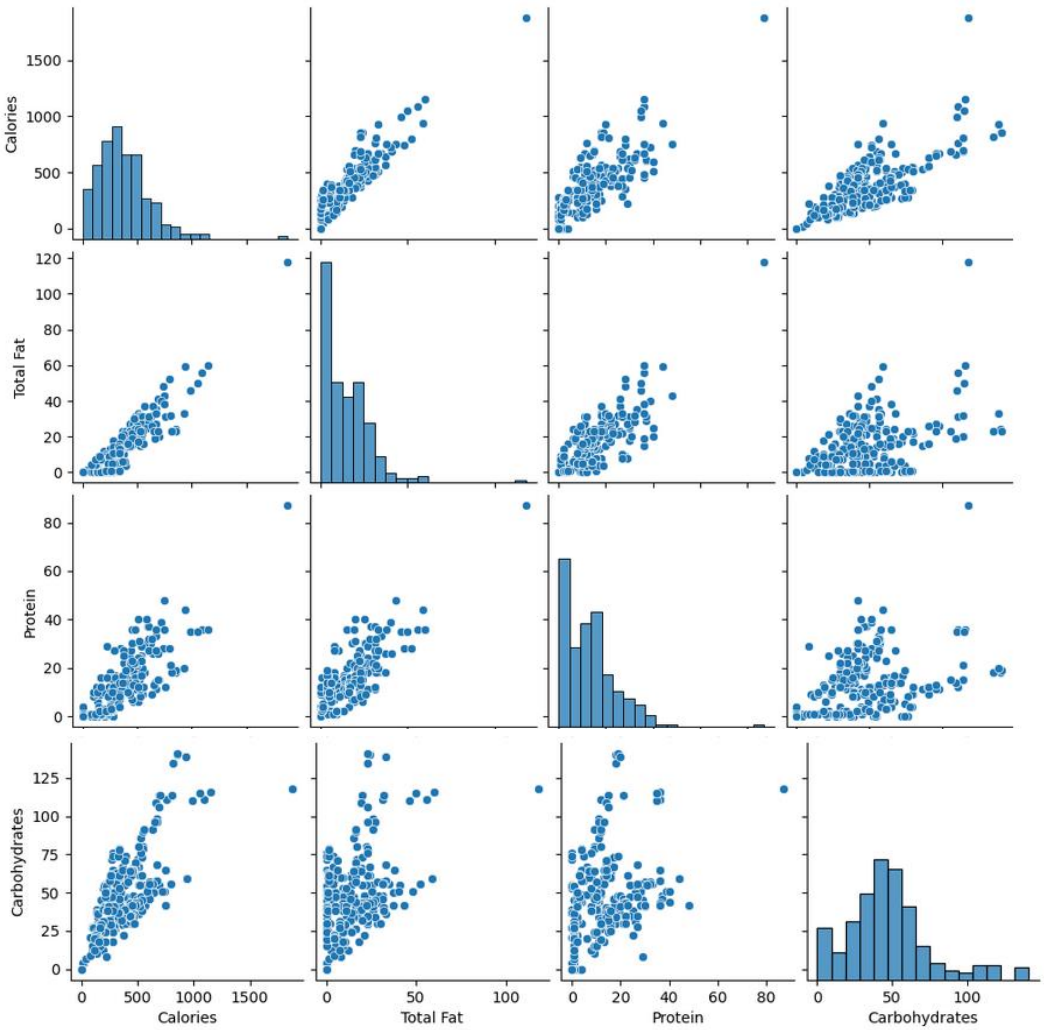


[17]:
```
    # Scatter Plot of Calories vs. Carbohydrate
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Carbohydrates', y='Calories', data=df, hue='Category', palette='bright', alpha=0.7)
plt. title('Scatter Plot of Calories vs. Carbohydrates')
plt.xlabel('Carbohydrates (g)')
plt.ylabel('Calories')
plt. legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



3.3 Exploratory Data Analysis (EDA): Identify trends and patterns in the dataset.

[18]:
```
# Pairplot to identify relationships between different nutritional components

sns.pairplot(df[['Calories', 'Total Fat', 'Protein', 'Carbohydrates' ]])
plt.show()
```



[19]:
```
df.corr(numeric_only=True).round(2)
```

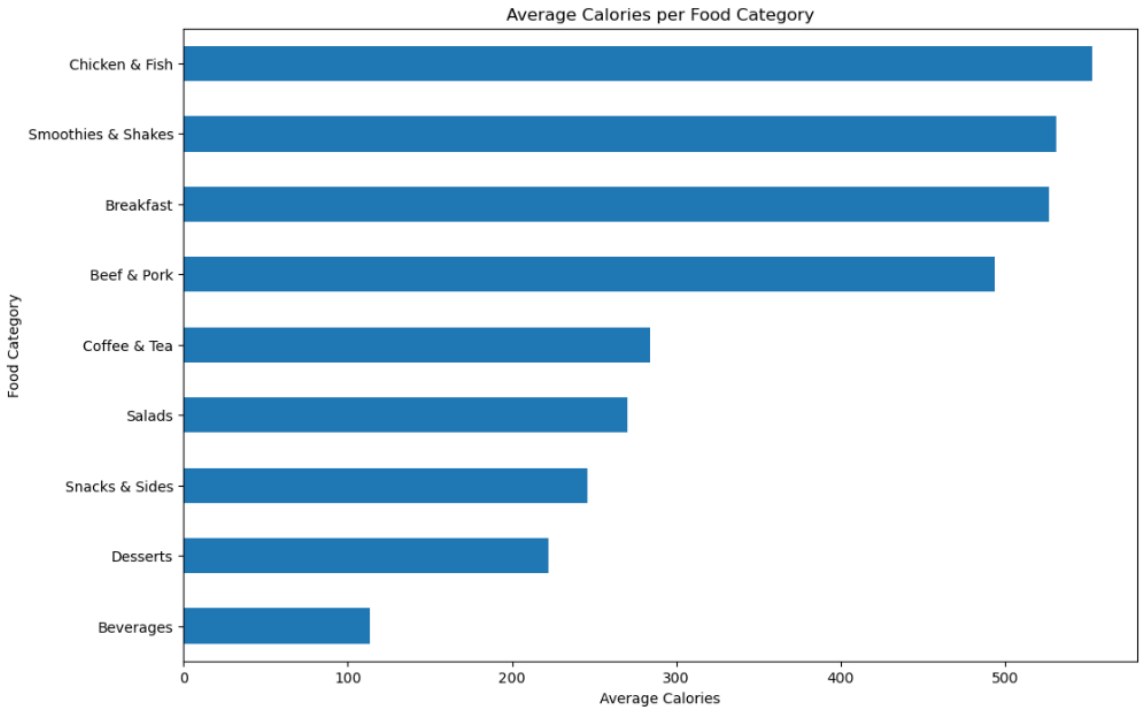
[19]:

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Cholesterol	Cholesterol (% Daily Value)	Sodium	...	Carbohydrates	Carbohydrates (% Daily Value)	Dietary Fiber	Die F [V
Calories	1.00	0.90	0.90	0.90	0.85	0.85	0.52	0.60	0.60	0.71	...	0.78	0.78	0.54	
Calories from Fat	0.90	1.00	1.00	1.00	0.85	0.85	0.43	0.68	0.68	0.85	...	0.46	0.46	0.58	
Total Fat	0.90	1.00	1.00	1.00	0.85	0.85	0.43	0.68	0.68	0.85	...	0.46	0.46	0.58	
Total Fat (% Daily Value)	0.90	1.00	1.00	1.00	0.85	0.85	0.43	0.68	0.68	0.85	...	0.46	0.46	0.58	
Saturated Fat	0.85	0.85	0.85	0.85	1.00	1.00	0.62	0.63	0.63	0.58	...	0.59	0.59	0.35	
Saturated Fat (% Daily Value)	0.85	0.85	0.85	0.85	1.00	1.00	0.62	0.63	0.63	0.59	...	0.59	0.59	0.36	
Trans Fat	0.52	0.43	0.43	0.43	0.62	0.62	1.00	0.25	0.25	0.19	...	0.46	0.46	0.05	
Cholesterol	0.60	0.68	0.68	0.68	0.63	0.63	0.25	1.00	1.00	0.62	...	0.27	0.27	0.44	
Cholesterol (% Daily Value)	0.60	0.68	0.68	0.68	0.63	0.63	0.25	1.00	1.00	0.62	...	0.27	0.27	0.43	
Sodium	0.71	0.85	0.85	0.85	0.58	0.59	0.19	0.62	0.62	1.00	...	0.20	0.20	0.69	
Sodium (% Daily Value)	0.71	0.85	0.85	0.85	0.59	0.59	0.19	0.62	0.62	1.00	...	0.20	0.20	0.69	
Carbohydrates	0.78	0.46	0.46	0.46	0.59	0.59	0.46	0.27	0.27	0.20	...	1.00	1.00	0.22	
Carbohydrates (% Daily Value)	0.78	0.46	0.46	0.46	0.59	0.59	0.46	0.27	0.27	0.20	...	1.00	1.00	0.22	
Dietary Fiber	0.54	0.58	0.58	0.58	0.35	0.36	0.05	0.44	0.43	0.69	...	0.22	0.22	1.00	
Dietary Fiber (% Daily Value)	0.54	0.58	0.58	0.58	0.35	0.35	0.06	0.44	0.44	0.69	...	0.23	0.23	0.99	
Sugars	0.26	-0.12	-0.12	-0.12	0.20	0.20	0.33	-0.14	-0.14	-0.43	...	0.76	0.76	-0.30	
Protein	0.79	0.81	0.81	0.81	0.60	0.61	0.39	0.56	0.56	0.87	...	0.35	0.35	0.64	
Vitamin A (% Daily Value)	0.11	0.06	0.05	0.05	0.06	0.07	0.08	0.08	0.08	0.08	...	0.08	0.08	0.34	
Vitamin C (% Daily Value)	-0.07	-0.09	-0.09	-0.09	-0.18	-0.18	-0.08	-0.08	-0.08	-0.03	...	-0.03	-0.04	0.14	
Calcium (% Daily Value)	0.43	0.16	0.16	0.16	0.40	0.40	0.39	0.13	0.13	-0.02	...	0.59	0.59	0.03	
Iron (% Daily Value)	0.64	0.74	0.73	0.74	0.58	0.58	0.33	0.65	0.65	0.87	...	0.21	0.21	0.74	

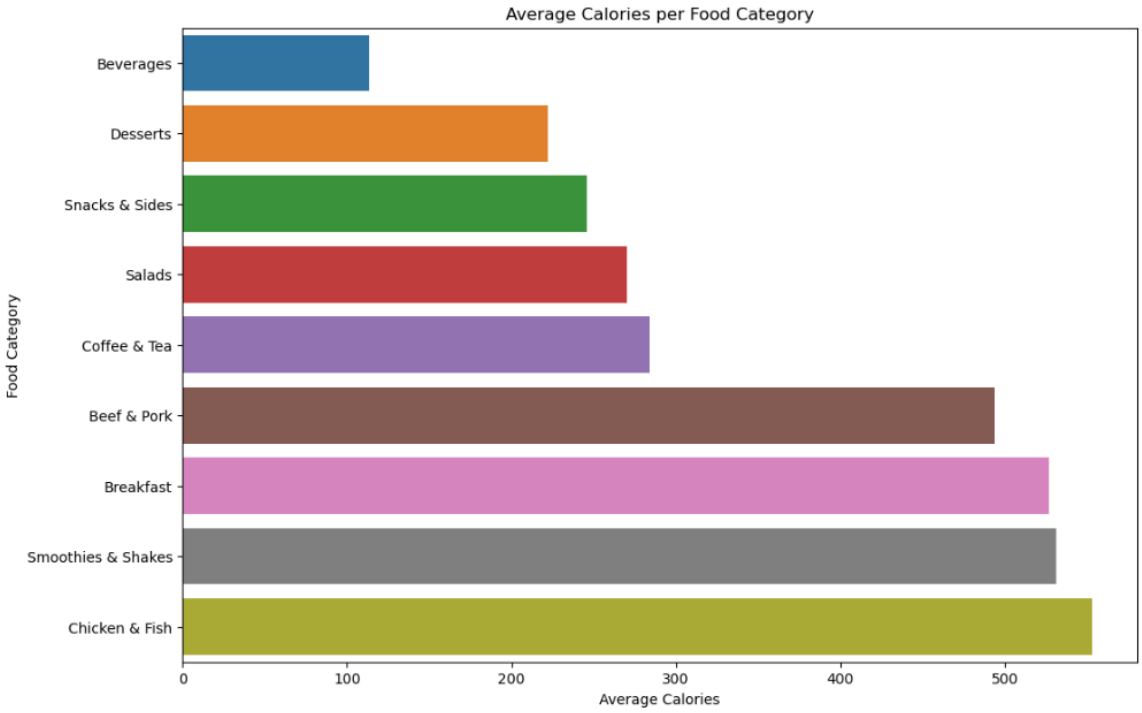
21 rows × 21 columns

4 Data Visualization:

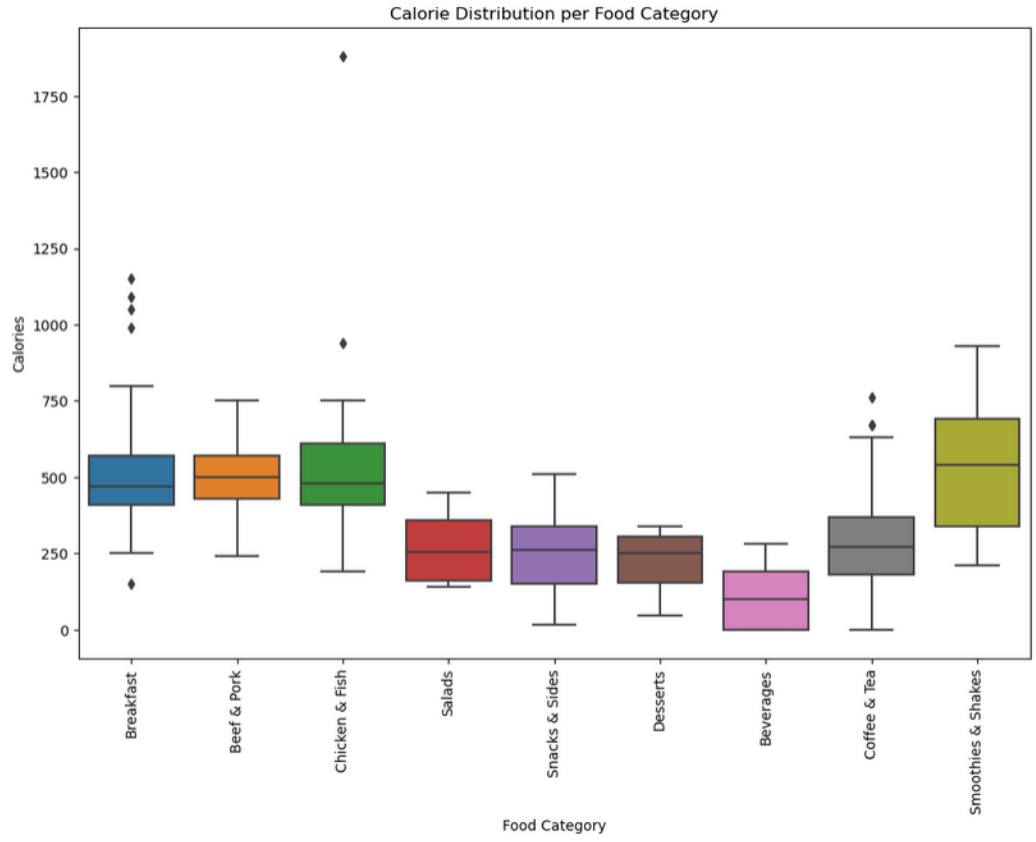
[20]:
```
# Bar chart for average calories per food category
avg_calories_per_category = df.groupby('Category') ['Calories' ].mean().sort_values()
plt.figure(figsize=(12, 8))
avg_calories_per_category.plot(kind='barh')
plt.title('Average Calories per Food Category')
plt.xlabel('Average Calories')
plt.ylabel('Food Category')
plt. show()
```



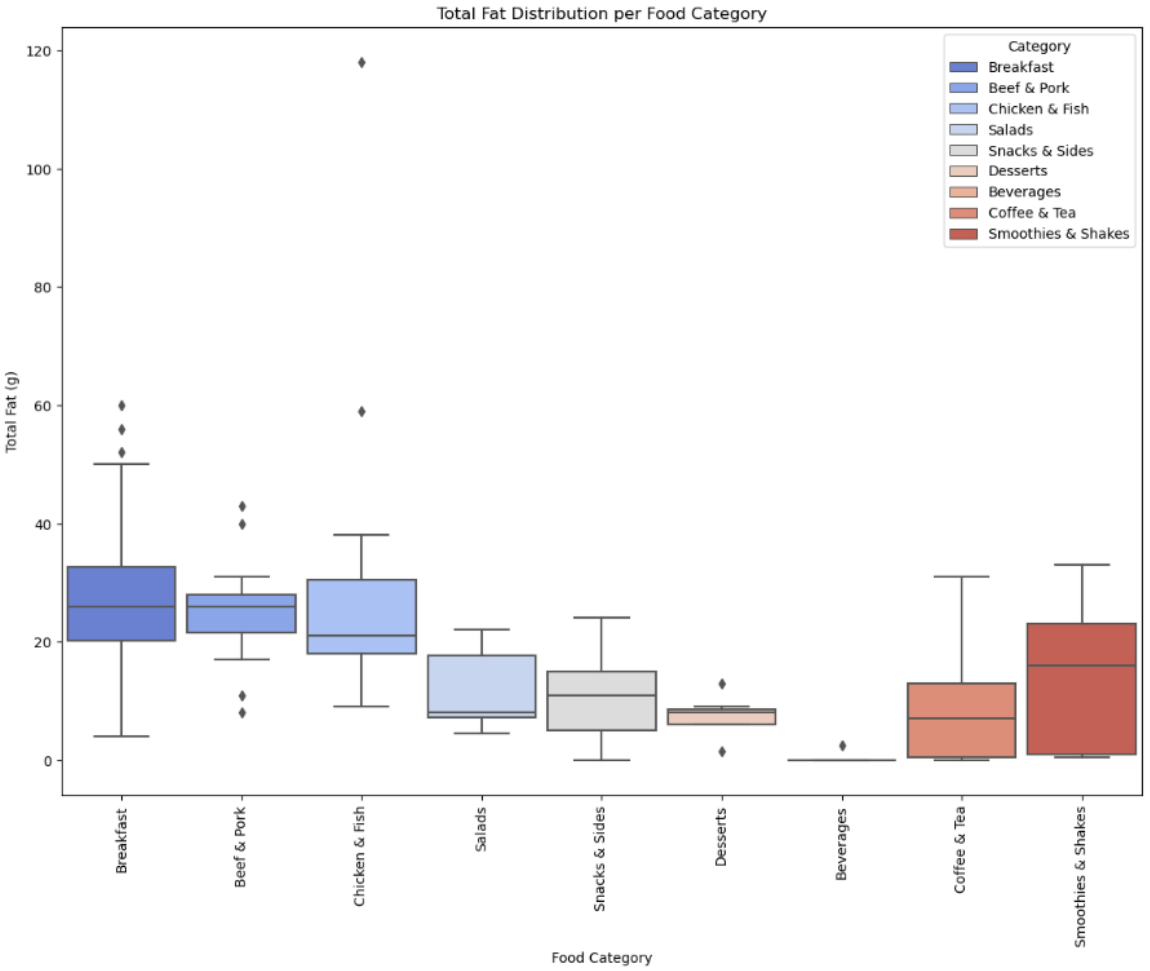
[21]:
```
# Plot the bar chart
plt.figure(figsize=(12, 8))
sns.barplot(x=avg_calories_per_category.values, y=avg_calories_per_category.index)
plt. title('Average Calories per Food Category')
plt.xlabel('Average Calories')
plt.ylabel('Food Category')
plt. show()
```



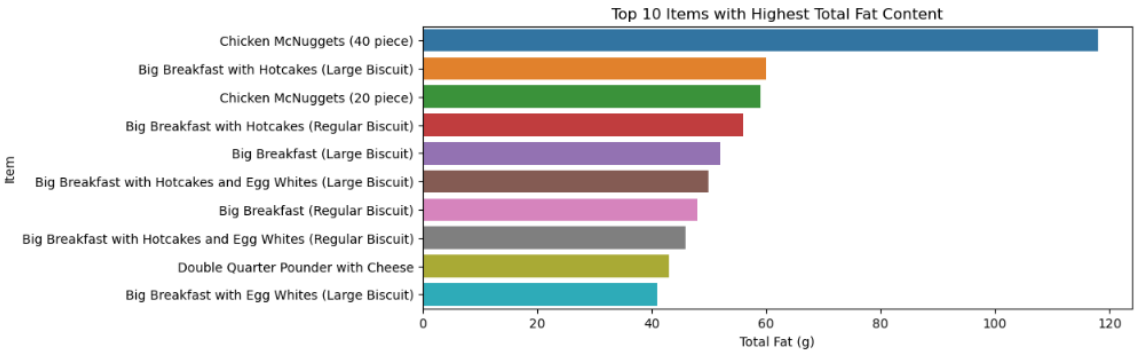
[22]:
```
# Box plot for calorie distribution per category
plt.figure(figsize=(12, 8))
sns.boxplot(x='Category', y='Calories', data=df)
plt.xticks(rotation=90)
plt.title('Calorie Distribution per Food Category')
plt.xlabel('Food Category')
plt.ylabel('Calories')
plt.show()
```



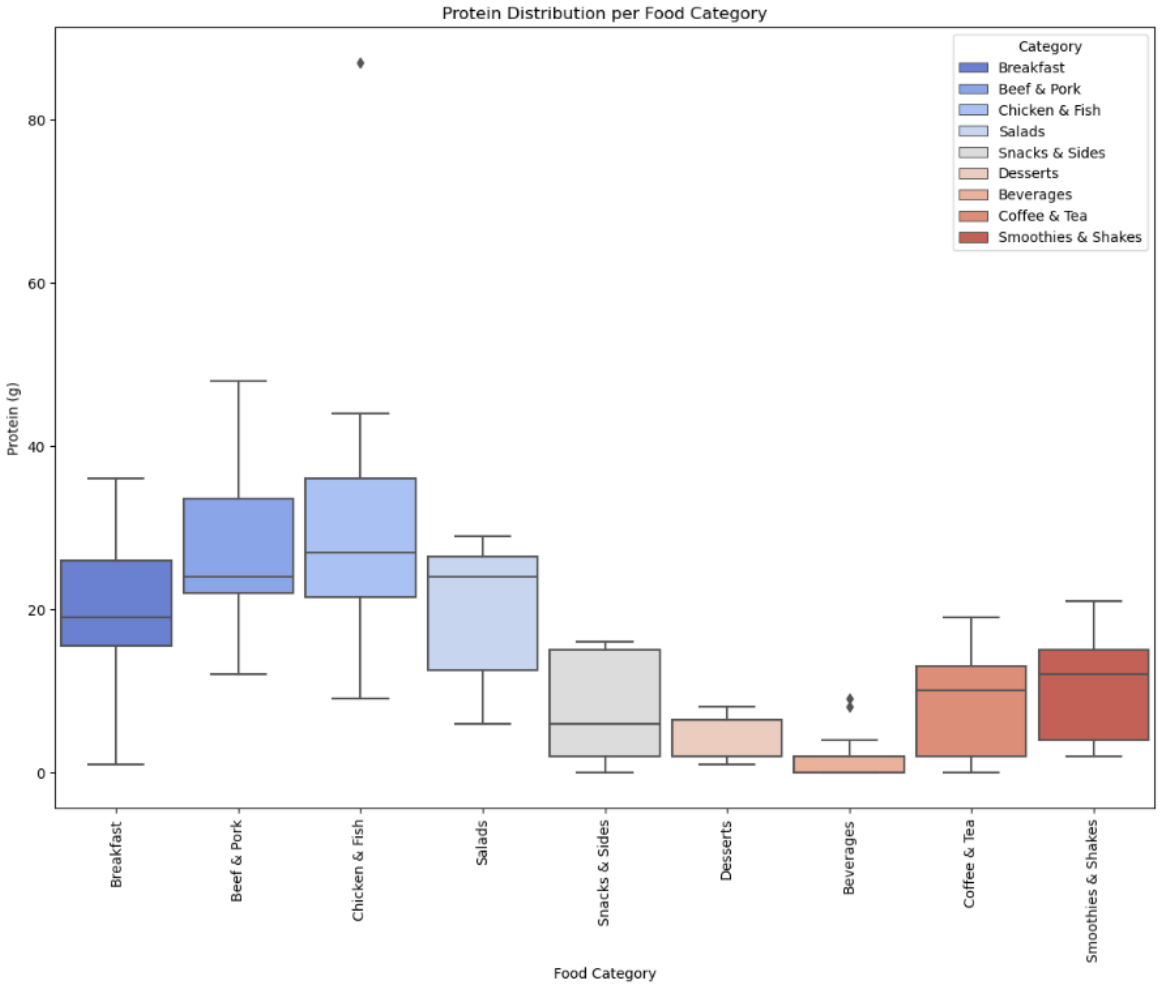
[34]:
```
# Box plot for total fat distribution per category
plt.figure(figsize=(14, 10))
sns.boxplot(x='Category', y='Total Fat', data=df, hue='Category', palette='coolwarm', dodge=False, width=0.9)
plt.xticks(rotation=90)
plt.title('Total Fat Distribution per Food Category')
plt.xlabel('Food Category')
plt.ylabel('Total Fat (g)')
plt. show()
```



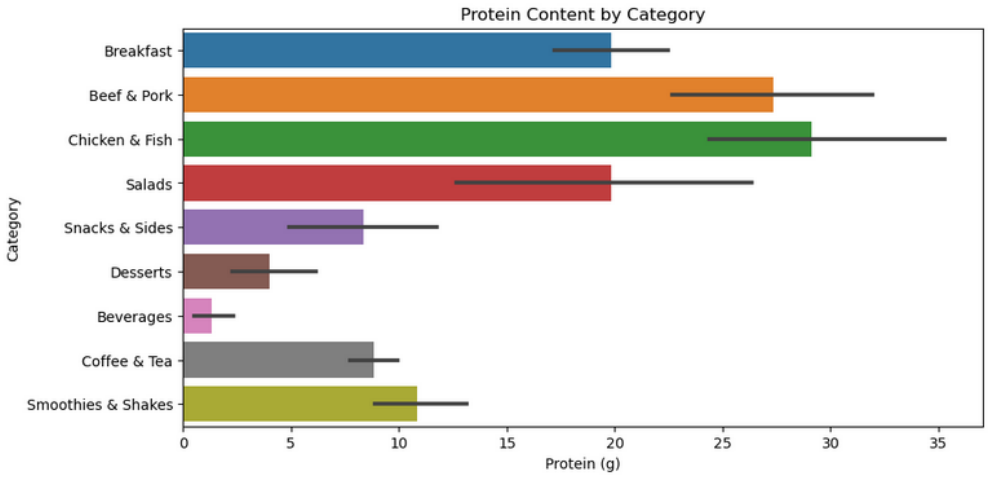
[53]:
```
# Top 10 Items with Highest Total Fat Content
max_fat = df.sort_values('Total Fat', ascending=False).head(10)

fig, ax = plt.subplots(figsize=(10, 4))
sns.barplot(x='Total Fat', y='Item', data=max_fat, ax=ax, dodge=False)

ax.set_title('Top 10 Items with Highest Total Fat Content')
ax.set_xlabel('Total Fat (g)')
ax.set_ylabel('Item')

plt.show()
```



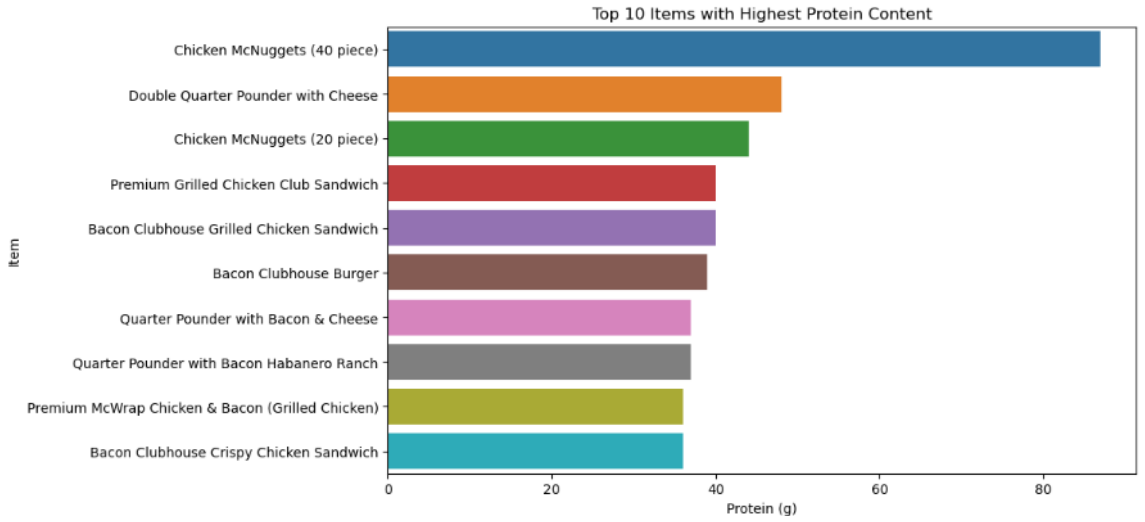
[37]:
```
# Box plot for protein distribution per category
plt.figure(figsize=(14, 10))
sns . boxplot (x='Category', y='Protein', data=df, hue='Category', palette='coolwarm', dodge=False, width=0.9)
plt.xticks(rotation=90)
plt. title('Protein Distribution per Food Category')
plt.xlabel ('Food Category')
plt.ylabel('Protein (g)')
plt. show()
```



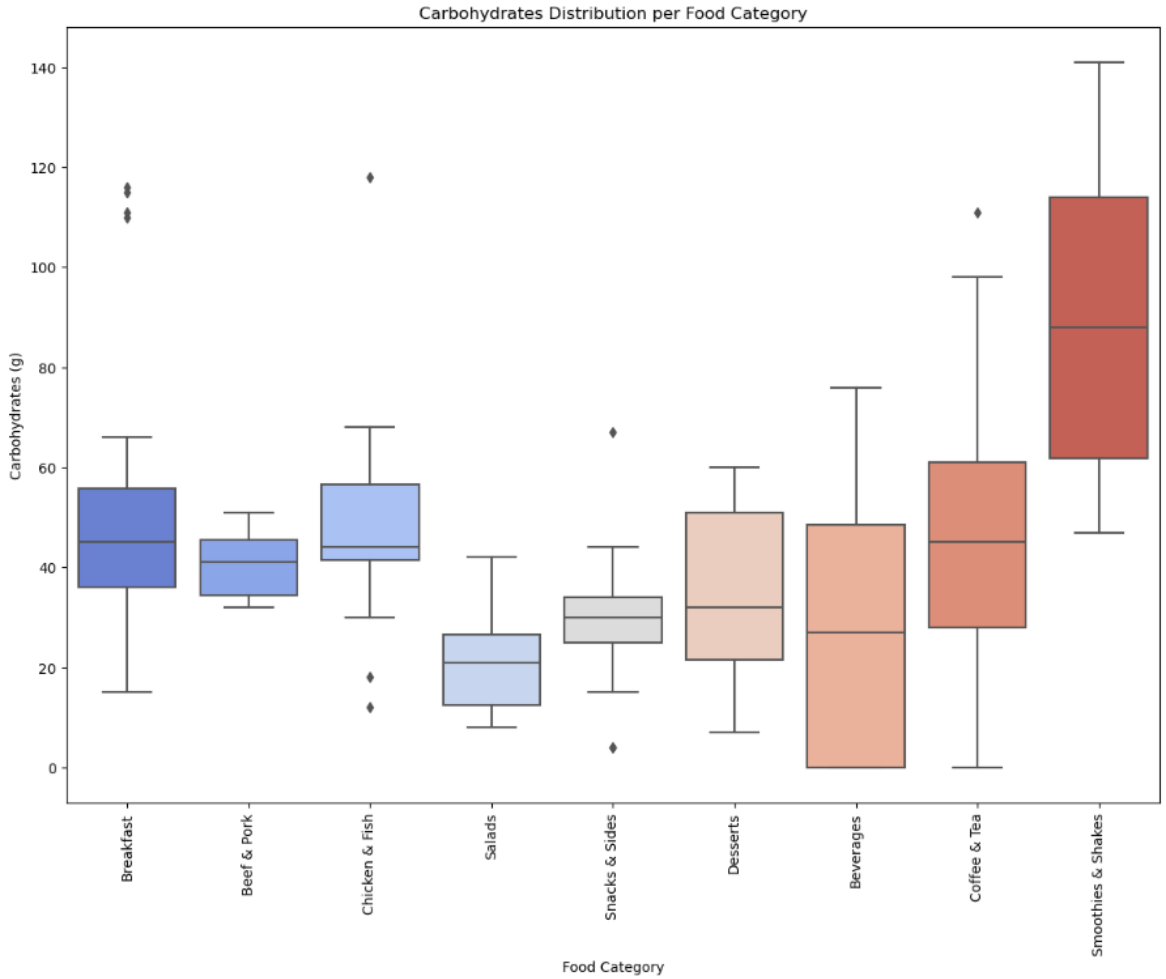
[52]:
```
# Protein Content by Category
fig, ax = plt.subplots(figsize=(10, 5))

sns.barplot(x='Protein', y='Category', data=df, ax=ax,)

ax.set_title('Protein Content by Category')
ax.set_xlabel('Protein (g)')
ax.set_ylabel('Category')

plt. show()
```



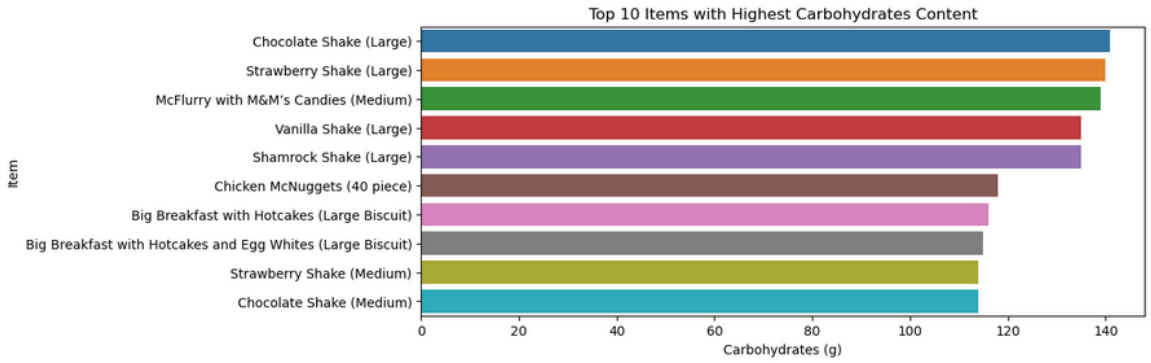
[51]:
```
# Top 10 items with the highest Protein content
max_protein = df.sort_values('Protein', ascending=False).head(10)

fig, ax = plt.subplots(figsize=(10, 6))

sns.barplot(x='Protein', y='Item', data=max_protein, ax=ax,)

ax.set_title('Top 10 Items with Highest Protein Content')
ax.set_xlabel('Protein (g)')
ax.set_ylabel('Item')
plt.show()
```



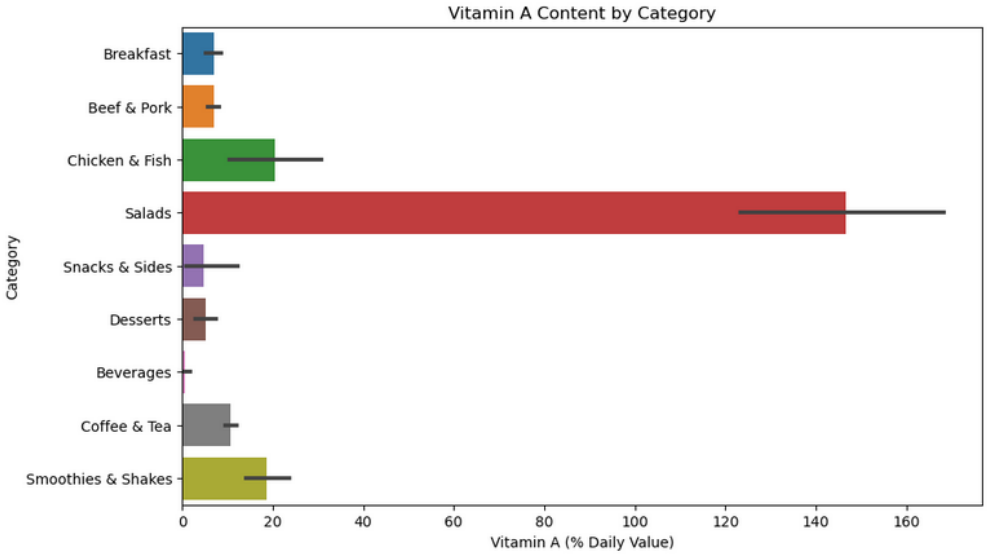
[55]:
```
# Box plot for carbohydrate distribution per category
plt.figure(figsize=(14, 10))
sns.boxplot(x='Category', y='Carbohydrates', data=df, palette='coolwarm', dodge=False)
plt.xticks(rotation=90)
plt.title('Carbohydrates Distribution per Food Category')
plt.xlabel('Food Category')
plt.ylabel('Carbohydrates (g)')
plt.show()
```



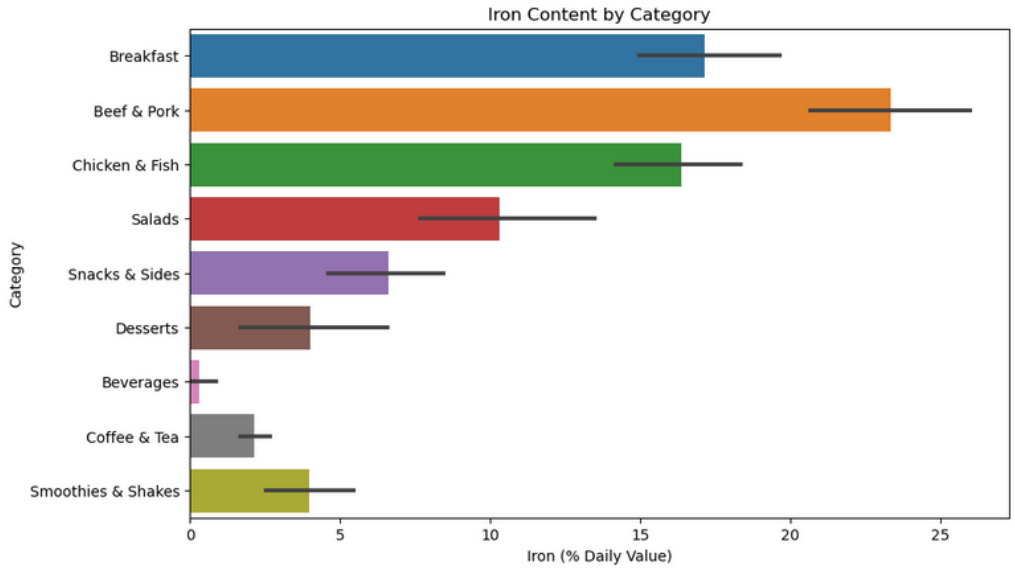
[56]:
```
# The top 10 items with the highest Carbohydrates content
max_carb = df.sort_values('Carbohydrates', ascending=False).head(10)

fig, ax = plt.subplots(figsize=(10, 4))
sns.barplot(x='Carbohydrates', y='Item', data=max_carb, ax=ax,)

ax.set_title('Top 10 Items with Highest Carbohydrates Content')
ax.set_xlabel('Carbohydrates (g)')
ax.set_ylabel('Item')
plt. show()
```



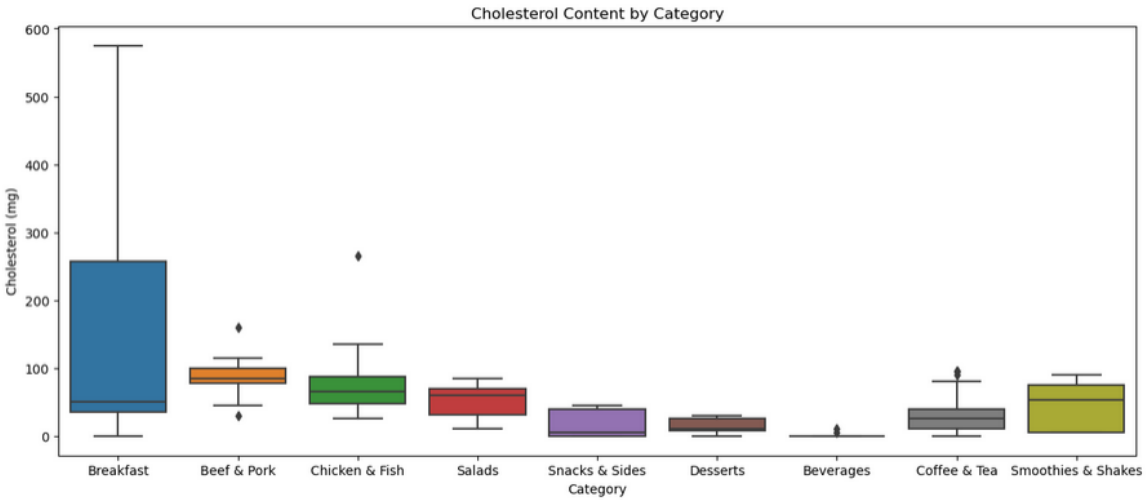
[57]:
```
# Vitamin A Content by Category
fig, ax = plt.subplots(figsize=(10, 6))

sns.barplot(x='Vitamin A (% Daily Value)', y='Category', data=df, ax=ax)

ax.set_title('Vitamin A Content by Category')
ax.set_xlabel('Vitamin A (% Daily Value)')
ax.set_ylabel('Category')

plt.show()
```



[58]:
```
# Iron Content by Category
fig, ax = plt.subplots(figsize=(10, 6))

sns.barplot(x='Iron (% Daily Value)', y='Category', data=df, ax=ax)

ax.set_title('Iron Content by Category')
ax.set_xlabel('Iron (% Daily Value)')
ax.set_ylabel('Category')
plt.show()
```



[59]:
```
# Cholesterol Content by Category
fig, ax = plt.subplots(figsize=(15, 6))

sns.boxplot(x='Category', y='Cholesterol', data=df, ax=ax)

# Add a title and labels
ax.set_title('Cholesterol Content by Category')
ax.set_xlabel('Category')
ax.set_ylabel('Cholesterol (mg)')

plt.show()
```



5.1 Nutrition-Based Insights: Identify menu items with the highest and lowest calorie counts.

[60]:
```
# Highest calorie items
df.nlargest(10, 'Calories')[['Item', 'Category', 'Calories']]
```

[60]:

	Item	Category	Calories
82	Chicken McNuggets (40 piece)	Chicken & Fish	1880
32	Big Breakfast with Hotcakes (Large Biscuit)	Breakfast	1150
31	Big Breakfast with Hotcakes (Regular Biscuit)	Breakfast	1090
34	Big Breakfast with Hotcakes and Egg Whites (La...	Breakfast	1050
33	Big Breakfast with Hotcakes and Egg Whites (Re...	Breakfast	990
81	Chicken McNuggets (20 piece)	Chicken & Fish	940
253	McFlurry with M&M's Candies (Medium)	Smoothies & Shakes	930
246	Strawberry Shake (Large)	Smoothies & Shakes	850
249	Chocolate Shake (Large)	Smoothies & Shakes	850
243	Vanilla Shake (Large)	Smoothies & Shakes	820

[61]:
```
# Lowest calorie items
df.nsmallest(10, 'Calories')[['Item', 'Category', 'Calories']]
```

[61]:

	Item	Category	Calories
114	Diet Coke (Small)	Beverages	0
115	Diet Coke (Medium)	Beverages	0
116	Diet Coke (Large)	Beverages	0
117	Diet Coke (Child)	Beverages	0
122	Diet Dr Pepper (Small)	Beverages	0
123	Diet Dr Pepper (Medium)	Beverages	0
124	Diet Dr Pepper (Large)	Beverages	0
125	Diet Dr Pepper (Child)	Beverages	0
136	Dasani Water Bottle	Beverages	0
137	Iced Tea (Small)	Coffee & Tea	0

5.2 Nutrition-Based Insights: Determine the average nutritional content of popular menu categories.

[62]:
```
# Average nutritional content per category
numeric_columns = df.select_dtypes(include='number').columns
numeric_df = df[numeric_columns]
df.groupby('Category') [numeric_columns].mean().round(2)
```

[62]:

Category	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Cholesterol	Cholesterol (% Daily Value)	Sodium	...	Carbohydrates	Carbohydrates (% Daily Value)	Dietary Fiber	Dietary Fiber (% Daily Value)	Sugars
Beef & Pork	494.00	224.67	24.87	38.60	10.47	52.00	1.10	87.33	28.93	1020.67	...	40.13	13.47	2.53	9.87	8.80
Beverages	113.70	0.74	0.09	0.15	0.06	0.30	0.00	0.56	0.19	41.48	...	28.81	9.59	0.04	0.07	27.85
Breakfast	526.67	248.93	27.69	42.67	10.65	53.43	0.11	152.86	50.95	1211.07	...	49.76	16.57	3.26	12.83	8.26
Chicken & Fish	552.96	242.22	26.96	41.33	6.17	31.11	0.13	75.37	25.22	1257.78	...	49.07	16.33	2.93	11.81	7.33
Coffee & Tea	283.89	71.11	8.02	12.36	4.92	24.37	0.14	27.26	9.38	136.89	...	44.53	14.86	0.78	3.18	39.61
Desserts	222.14	64.29	7.36	11.14	4.29	21.29	0.00	15.00	4.86	117.14	...	34.86	11.57	1.00	3.43	26.14
Salads	270.00	108.33	11.75	18.33	3.75	18.50	0.00	51.67	17.33	588.33	...	21.67	7.17	4.50	18.50	6.83
Smoothies & Shakes	531.43	127.68	14.12	21.71	8.38	41.79	0.54	45.00	14.71	183.57	...	90.43	30.14	1.46	5.75	77.89
Snacks & Sides	245.77	94.62	10.54	16.23	2.69	13.38	0.00	18.46	6.23	395.77	...	29.15	9.62	1.54	7.08	4.08

9 rows × 21 columns

Analysis and Reporting

Summary of Findings and Insights from the Analysis

1. Calorie Distribution:

General Trends: Most McDonald's menu items fall within a specific calorie range, with some items being notable outliers.

High Calorie Items: Desserts and burgers generally have higher calorie counts compared to beverages and salads.

2. Nutritional Content Analysis:

Total Fat: Burgers and breakfast items have higher fat content, while salads and beverages typically have lower fat levels.

Protein: Protein content is highest in burgers and chicken items, making them good protein sources.

Carbohydrates: Desserts and beverages are high in carbohydrates, mainly due to their sugar content.

Sugars: Desserts and beverages have the highest sugar levels, aligning with their carbohydrate content.

Sodium: High sodium levels are common in burgers, chicken items, and breakfast options.

3. Correlation Analysis:

Calories and Total Fat: A strong positive correlation exists between calories and total fat, indicating that higher calorie items tend to have more fat.

Calories and Other Nutrients: Calories also show a positive correlation with protein and sodium, suggesting that high calorie items are often rich in these nutrients.

Carbohydrates and Sugars: A strong correlation exists between carbohydrates and sugars, as sugars contribute significantly to the carbohydrate content.

4. Category wise Trends:

Burgers: High in calories, total fat, protein, and sodium; energy dense but rich in nutrients needing moderation.

Salads: Generally lower in calories and fat but can vary significantly based on dressings and addons.

Beverages: Wide calorie range, with sugary drinks contributing high sugar and carbohydrate content.

Desserts: High in sugar and carbohydrates, moderate in calories, and low in protein and fat.

Breakfast Items: High in calories, total fat, and sodium, with moderate protein content.

Benefits of Nutritional Analysis for McDonald's Customers and the Organization

Benefits for Customers:

1. Informed Choices: Nutritional transparency allows customers to make informed food choices. Understanding nutritional content helps customers choose meals that align with their dietary goals and health needs.

2. Healthier Alternatives: Clear information helps customers identify healthier options. Customers might opt for salads or grilled chicken items over higher calorie burgers and fried foods.

3. Dietary Management: Customers with specific dietary requirements can select appropriate menu items, aiding in better health management.

4. Portion Control: Knowing calorie content helps customers practice portion control, avoiding excessive calorie intake.

Benefits for the Organization:

1. Enhanced Customer Trust: Providing detailed nutritional information builds transparency and trust, demonstrating McDonald's commitment to customer health and wellbeing.

2. Market Differentiation: Offering detailed nutritional information can differentiate McDonald's from competitors and attract health-conscious consumers.

3. Menu Optimization: Nutritional analysis helps identify areas for menu improvement, such as reducing sodium in high sodium items or offering lower calorie versions of popular items.

4. Targeted Marketing: Understanding nutritional profiles enables targeted marketing efforts, promoting items that align with popular dietary trends like high protein or low carb diets.

5. Regulatory Compliance: Detailed nutritional information ensures compliance with regulatory requirements, avoiding legal issues and enhancing corporate responsibility.

6. Customer Feedback and Innovation: Analyzing customer preferences based on nutritional data provides insights into consumer trends, enabling McDonald's to innovate and introduce new items that meet evolving customer demands.

Recommendations for McDonald's

1. Introduce Healthier Options: Expand the menu to include lower calorie, low fat, and low sodium items, such as more salads, grilled options, and fruit-based desserts.

2. Nutritional Labelling: Ensure all menu items have clear and accessible nutritional labelling both instore and online to help customers make informed choices.

3. Reformulate Recipes: Consider reducing sodium, fat, and sugar in recipes without compromising taste.

4. Portion Sizes: Offer smaller portion sizes for high calorie items to provide more choices and control over calorie intake.

5. Customer Education: Launch campaigns to educate customers on balanced nutrition and making healthier choices from the McDonald's menu.

6. Sustainability and Sourcing: Emphasize sustainability and healthier sourcing practices, such as using organic ingredients, reducing additives, and ensuring high nutritional standards.

By implementing these recommendations, McDonald's can improve the nutritional profile of its menu, cater to the demand for healthier food options, and enhance its brand image as a responsible and customer focused organization.