McDonald's Menu Nutritional Analysis - Project

github link - https://github.com/nehakholia/McDonald-s-Menu-Nutritional-Analysis---Project

---

Problem Statement

McDonald's is a global fast-food chain known for its diverse menu offerings. As a data analyst, your task is to analyze the nutritional content of the menu items available at McDonald's outlets. This analysis will provide valuable insights into the calorie count and nutrition facts of various menu items.

Objectives:

1. Extract meaningful information from the McDonald's menu nutritional dataset.
2. Perform exploratory data analysis to understand the nutritional distribution and trends.
3. Create visualizations to present the calorie count and nutrition facts of different menu items.
4. Identify healthy and less healthy menu options based on nutritional content.

Data

- **Category**: Category of manu item
- **Item**: name of the menu item
- **Serving size**: serving size of the manu item in grams and ounces
- **calories**: calorie count of the menu item
- **calories from fat**: calorie count from fat of the menu item
- **Total fat**: total fat of the menu item
- **Saturated Fat**: total saturated fat of the menu item
- **Saturated Fat (% Daily Value)**: daily % of saturated fat
- **Cholesterol**: total cholestrol value of the menu item
- **Cholesterol (% Daily Value):** daily % value of cholestrol
- **Sodium:** total sodium count in the menu item
- **Sodium (% Daily Value):** daily % value of sodium in the menu item
- **Carbohydrates:** total carbohydrates count in the menu item
- **Carbohydrates (% Daily Value)**: daily % value of carbohydrates
- **Dietary Fiber:** total dietry fibre in the menu item
- **Dietary Fiber (% Daily Value):** daily % value of fibre
- **Sugars:** total sugar count in the menu item
- **Protein:** total protein in the menu item
- **Vitamin A (% Daily Value):** daily % of vitamin A
- **Vitamin C (% Daily Value):** daily % of vitamin C
- **Calcium (% Daily Value):** daily % of calcium
- **Iron (% Daily Value):** daily % of iron

In [1]:

```
# import libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Data Collection

In [25]:

```
df = pd.read_csv(r"D:\data_analytics\Projects\Nutrical Dataset.csv")
```

Data Preprocessing

Understanding the data

In [3]:

```
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
```

```
In [4]: len(df)
```

```
Out[4]: 260
```

```
In [5]: df.columns
```

```
Out[5]: 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')
```

```
In [6]: df.head(5)
```

Out[6]:

	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)
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0.0	...	31	10
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0.0	...	30	10
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0	...	29	10
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0.0	...	30	10
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23.0	35	8.0	42	0.0	...	30	10

5 rows × 24 columns

```
In [7]: df.describe()
```

Out[7]:

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Cholesterol	Cholesterol (% Daily Value)	Sodiu
count	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.00000
mean	368.269231	127.096154	14.165385	21.815385	6.007692	29.965385	0.203846	54.942308	18.392308	495.75000
std	240.269886	127.875914	14.205998	21.885199	5.321873	26.639209	0.429133	87.269257	29.091653	577.02632
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	210.000000	20.000000	2.375000	3.750000	1.000000	4.750000	0.000000	5.000000	2.000000	107.50000
50%	340.000000	100.000000	11.000000	17.000000	5.000000	24.000000	0.000000	35.000000	11.000000	190.00000
75%	500.000000	200.000000	22.250000	35.000000	10.000000	48.000000	0.000000	65.000000	21.250000	865.00000
max	1880.000000	1060.000000	118.000000	182.000000	20.000000	102.000000	2.500000	575.000000	192.000000	3600.00000

8 rows × 21 columns

Missing value Treatment

In [53]:

```
## Identifying missising value
df.isnull().sum()
```

Out[53]:

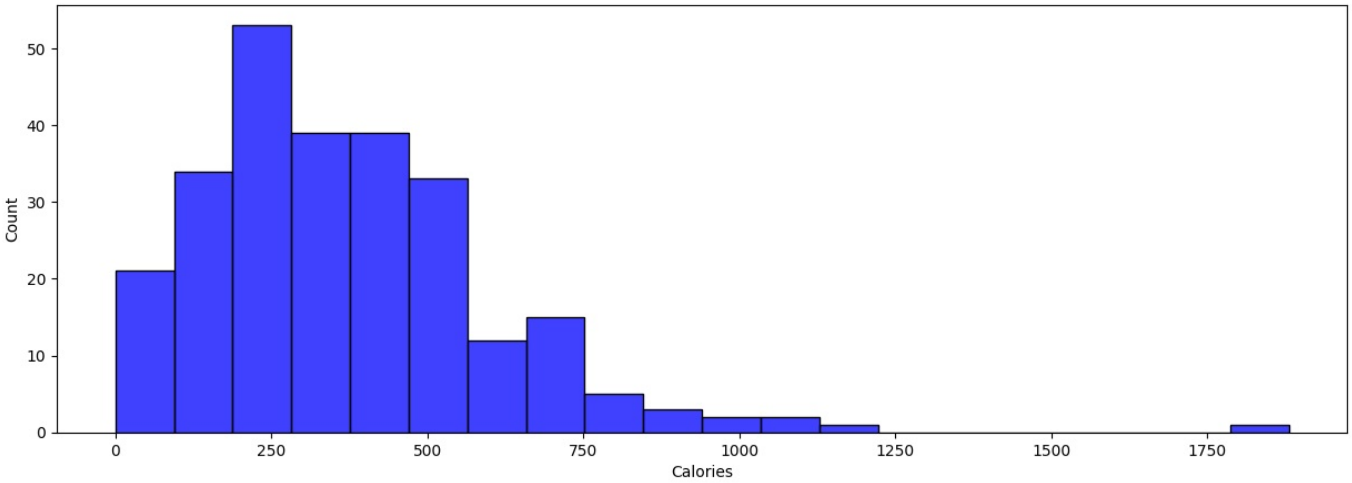
```
Category                         0
Item                             0
Serving Size                     0
Calories                         0
Calories from Fat                0
Total Fat                        0
Total Fat (% Daily Value)        0
Saturated Fat                    0
Saturated Fat (% 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
```

Conclusion: There is no missing values in the dataset

Analyze the distribution of calorie counts across menu items

In [8]:

```
plt.figure(figsize = (15,5))
sns.histplot(x = 'Calories', data = df, color = 'blue', bins = 20)
plt.show()
```



Conclusion:

Most menu items have a Calories between 180 and 280 calorie range and the distribution is **positive skewed.**

In [26]:

```
Calories = df['Calories'].describe()
Calories
```

Out[26]:

```
count     260.000000
mean      368.269231
std       240.269886
min         0.000000
25%       210.000000
50%       340.000000
75%       500.000000
max      1880.000000
Name: Calories, dtype: float64
```

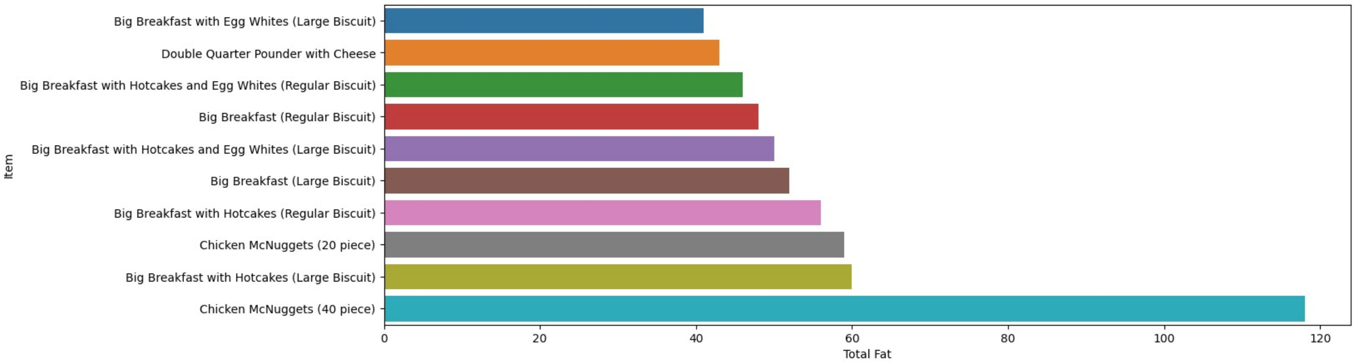
Conclusion:

The mean calorie count is 368.27 calories. The standard deviation is 240 calories, indicating the widespread in the calorie counts. 25% of the items have 210 or less calories, 50% of items have 340 or less calories & 75% of items have 500 or less calories.

Explore the nutritional content (e.g., fat, protein, carbohydrates) of different items.

In [27]:

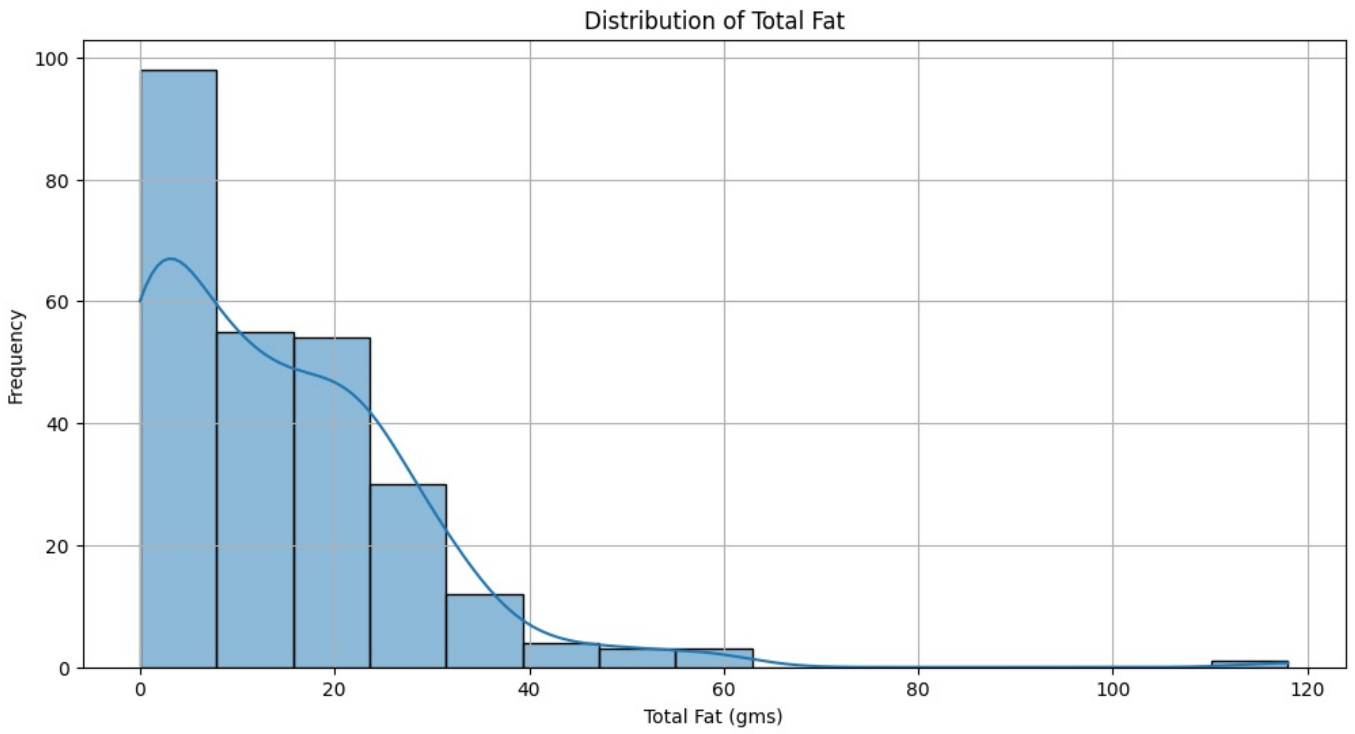
```
## Total Fat

max_fat = df.sort_values('Total Fat').tail(10)

plt.figure(figsize = (15,5))
sns.barplot(y = 'Item', x = 'Total Fat', data = max_fat, hue = 'Item')
plt.show()
```



Conclusion: Chicken McNuggets(40 piece) has highest fat value

In [28]:

```
plt.figure(figsize=(12,6))
sns.histplot(data=df, x='Total Fat',bins=15, kde = True)
plt.title('Distribution of Total Fat')
plt.xlabel('Total Fat (gms)')
plt.ylabel('Frequency')
plt.grid()
plt.show()
```



In [29]:

```
Total_fat = df['Total Fat'].describe()
Total_fat
```

Out[29]:

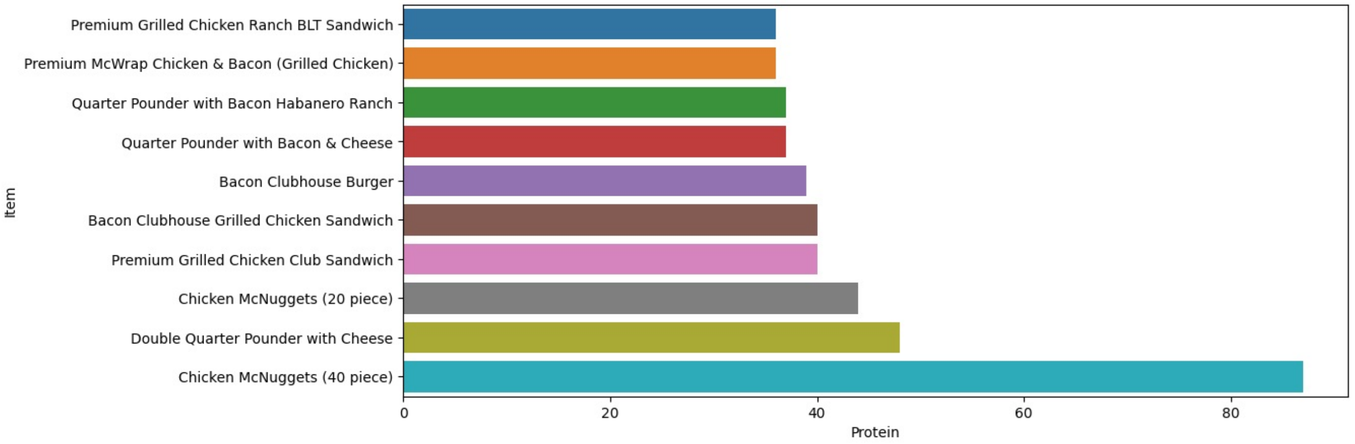
```
count    260.000000
mean      14.165385
std       14.205998
min        0.000000
25%        2.375000
50%       11.000000
75%       22.250000
max      118.000000
Name: Total Fat, dtype: float64
```

Conclusion:

The mean of the Total fat content is 14.16 grams. The standard deviation is 14.2 grams, indicating the moderate spread in the total fat values. 25% of the items have 2.37 grams or less total fat, 50% of items have 11 grams or less total fat & 75% of items have 22.25 grams or less total fat.

In [30]:

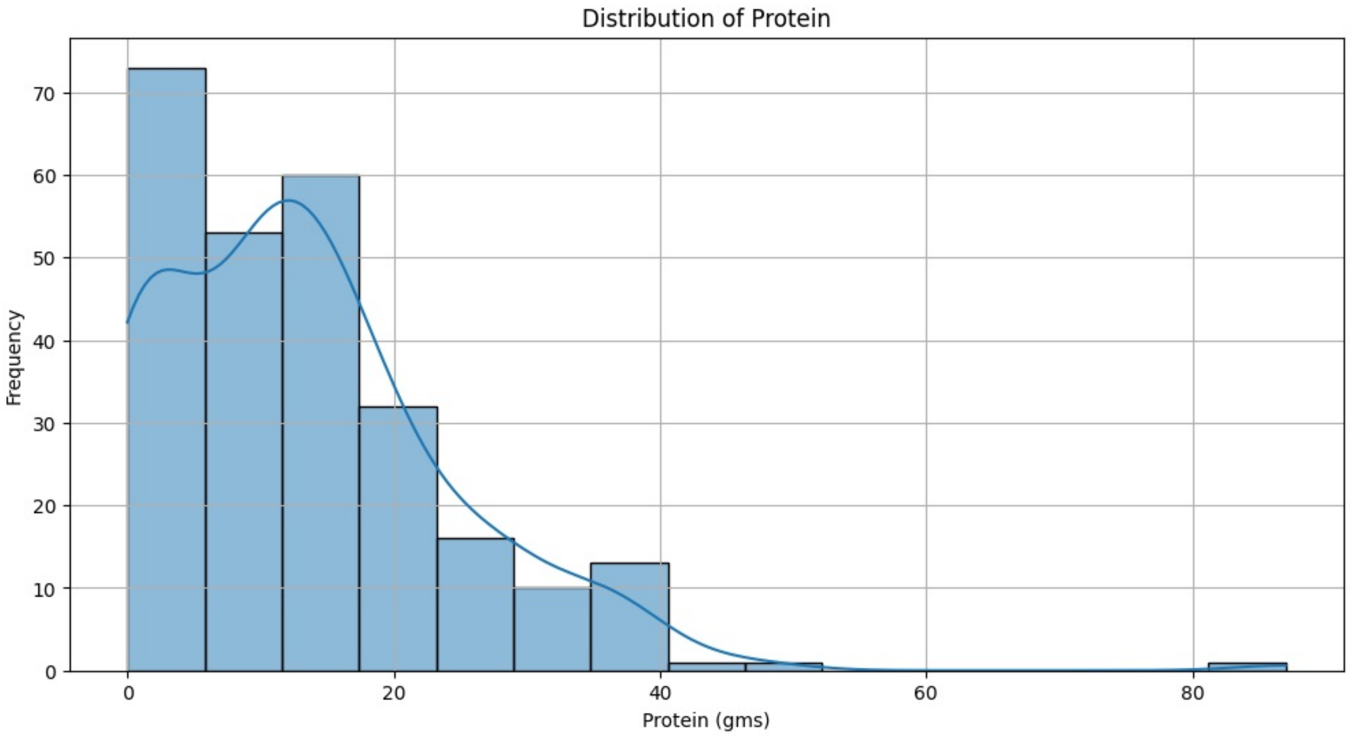
```
## Protein

max_Protein = df.sort_values('Protein').tail(10)

plt.figure(figsize = (12,5))
sns.barplot(y = 'Item', x = 'Protein', data = max_Protein, hue = 'Item')
plt.show()
```



Conclusion: Chicken McNuggets(40 piece) has highest Protein value

In [31]:

```
plt.figure(figsize=(12,6))
sns.histplot(data=df, x='Protein',bins=15, kde = True )
plt.title('Distribution of Protein')
plt.xlabel('Protein (gms)')
plt.ylabel('Frequency')
plt.grid()
plt.show()
```



In [32]:

```
Proteins =df['Protein'].describe()
Proteins
```

Out[32]:

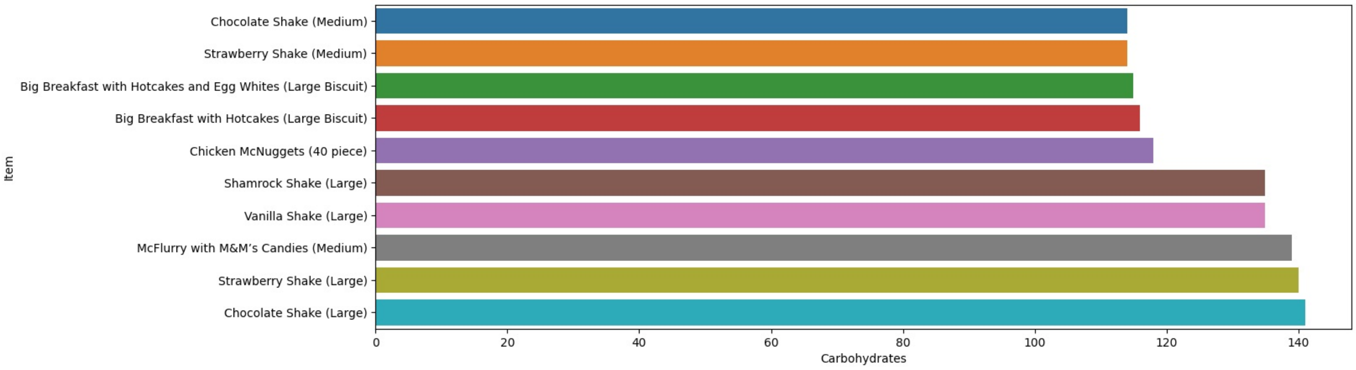
```
count    260.000000
mean      13.338462
std       11.426146
min        0.000000
25%        4.000000
50%       12.000000
75%       19.000000
max       87.000000
Name: Protein, dtype: float64
```

Conclusion:

The mean of the Protein content is 13.33 grams. The standard deviation is 11.43 grams, indicating the moderate spread in the protein values. 25% of the items have 12 grams or less proteins, 50% of items have 19 grams or less proteins & 75% of items have 87 grams or less proteins.

In [33]:

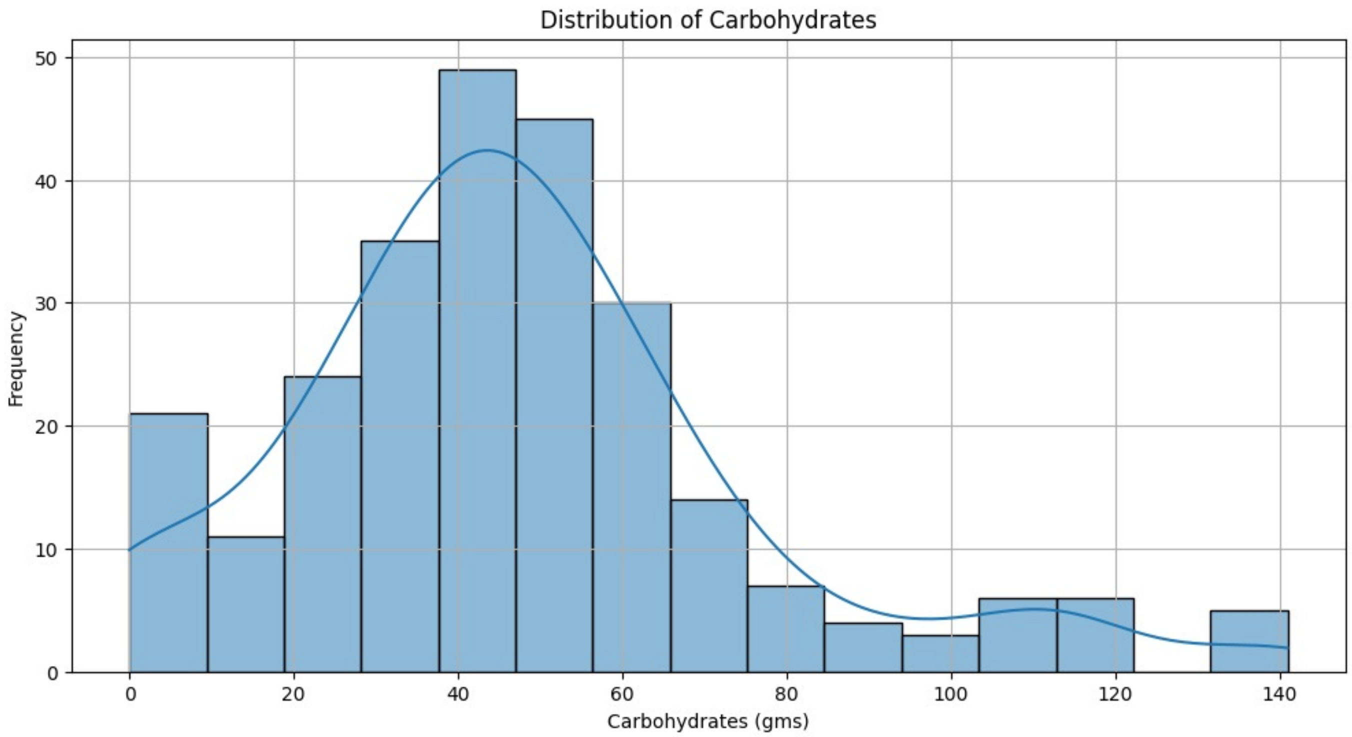
```
## Carbohydrates

max_Carbohydrates = df.sort_values('Carbohydrates').tail(10)

plt.figure(figsize = (15,5))
sns.barplot(y = 'Item', x = 'Carbohydrates', data = max_Carbohydrates, hue = 'Item')
plt.show()
```



Conclusion: Choclate Shake (Large) has highest carbohydrates

In [34]:

```
plt.figure(figsize = (12,6))
sns.histplot(data = df, x='Carbohydrates',bins = 15, kde = True )
plt.title('Distribution of Carbohydrates')
plt.xlabel('Carbohydrates (gms)')
plt.ylabel('Frequency')
plt.grid()
plt.show()
```



```
In [35]: Carbohydrates = df['Carbohydrates'].describe()
         Carbohydrates
```

```
Out[35]: count    260.000000
         mean      47.346154
         std       28.252232
         min        0.000000
         25%       30.000000
         50%       44.000000
         75%       60.000000
         max      141.000000
         Name: Carbohydrates, dtype: float64
```

Conclusion:

The mean of the Carbohydrates content is 47.34 grams. The standard deviation is 28.25 grams, indicating the widespread in the carbohydrates values. 25% of the items have 30 grams or less carbohydrates, 50% of items have 44 grams or less carbohydrates & 75% of items have 60 grams or less carbohydrates.

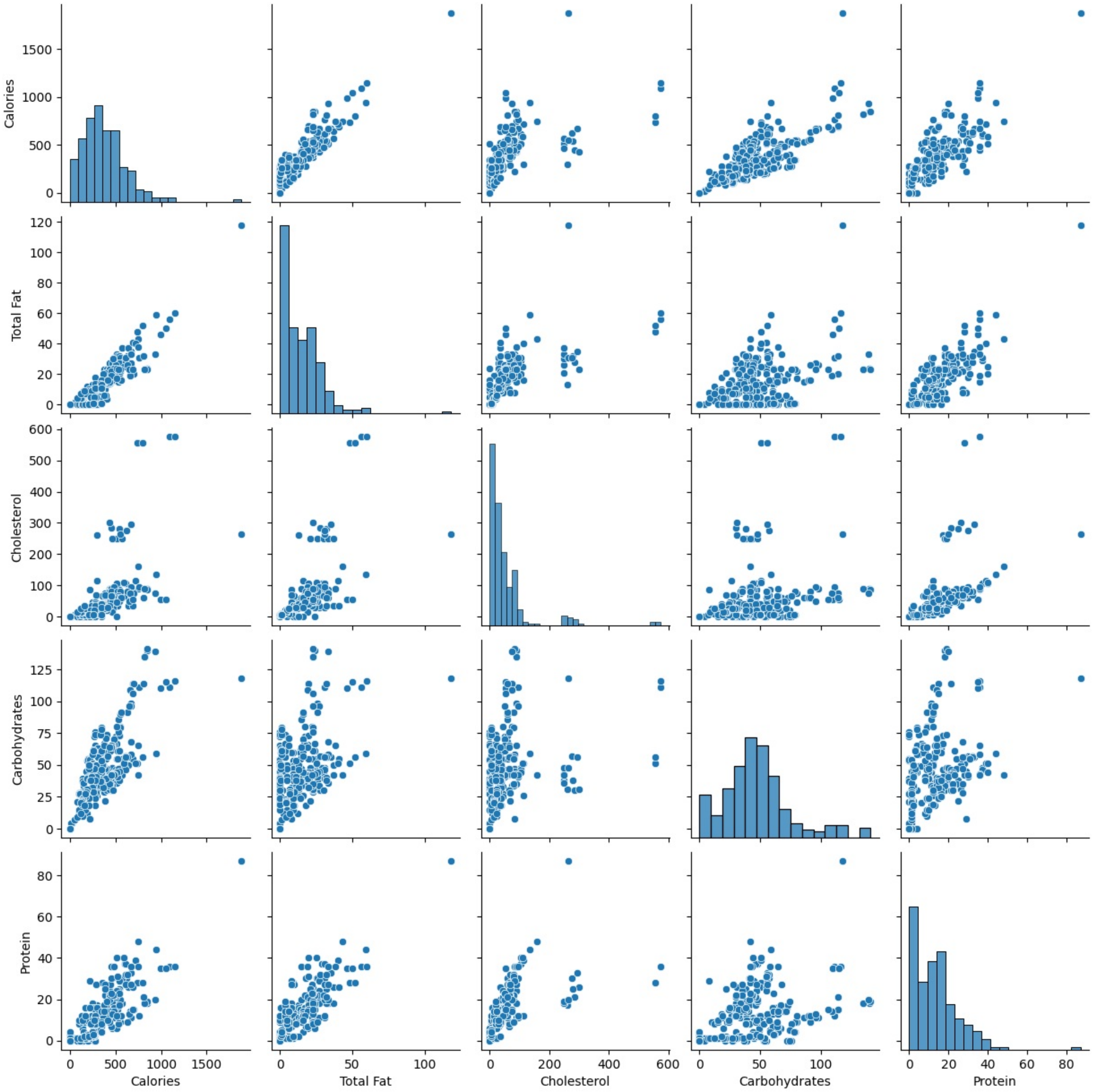
Identify trends and patterns in the dataset.

```
In [36]: nutritional_vars = ['Calories','Total Fat', 'Cholesterol','Carbohydrates','Proteins']
         nutritional_vars
```

```
Out[36]: ['Calories', 'Total Fat', 'Cholesterol', 'Carbohydrates', 'Proteins']
```

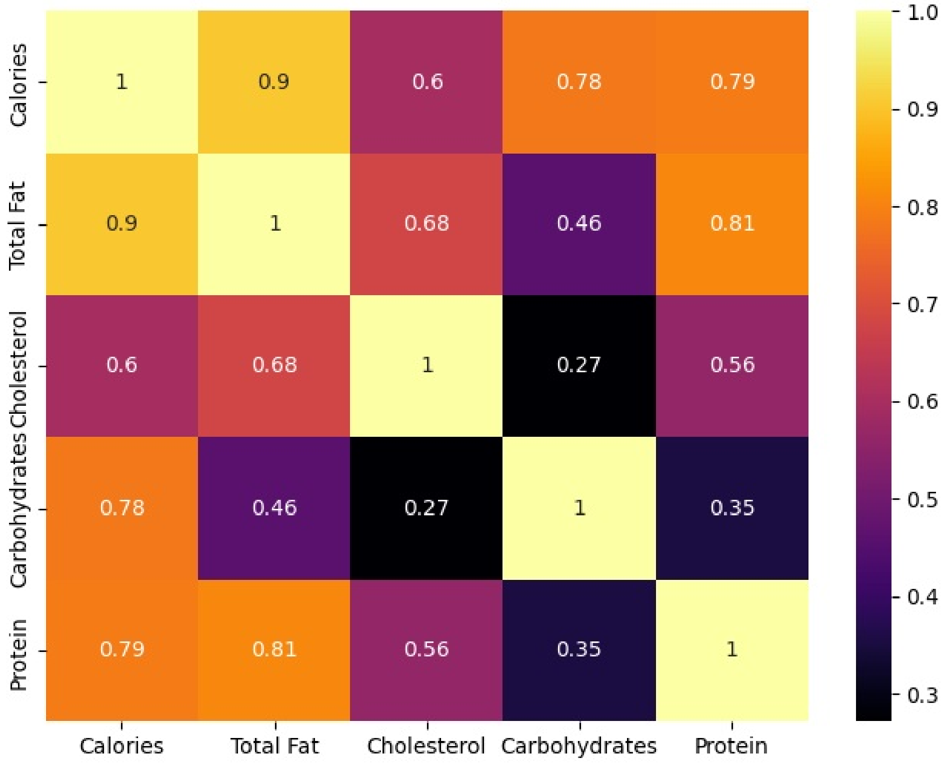
```
In [37]: sns.pairplot(df[['Calories','Total Fat', 'Cholesterol','Carbohydrates','Protein']])
```

```
Out[37]: <seaborn.axisgrid.PairGrid at 0x1c9c1e26d10>
```



In [40]:

```python
calories_vs_nutrients = df[['Calories','Total Fat', 'Cholesterol','Carbohydrates','Protein']].corr()
plt.figure(figsize=(8,6))
sns.heatmap(calories_vs_nutrients, annot=True, cmap='inferno')
plt.show()
```



Conclusion:

1. Calories and Total Fat: A strong positive correlation (0.904) indicates that items with higher calorie counts typically have higher total fat content.

2. Total Fat and Protein: A high correlation (0.807) shows that protein-rich items are also likely to be higher in total fat.

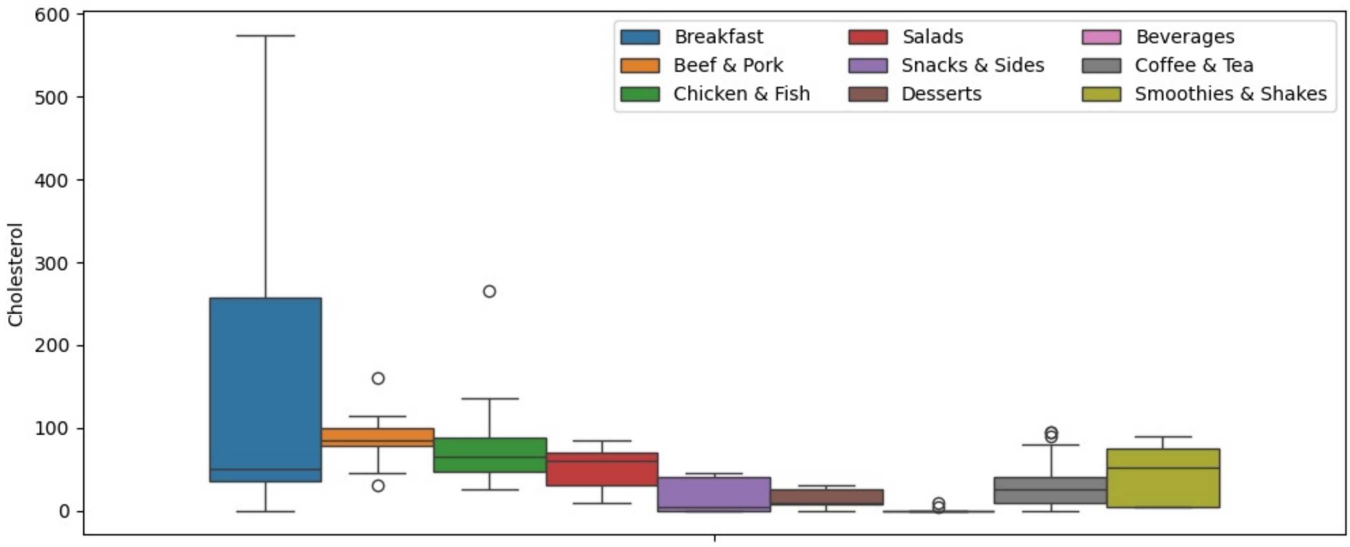
3. Calories and Protein: The positive correlation (0.787) suggests that items with more calories generally have higher protein content.

4. Calories and Carbohydrates: A positive correlation (0.781) means that items with higher calorie counts often have more carbohydrates.

Lastly, I included a heatmap to visually represent the correlation matrix, making it easier to understand how these nutrients relate to each other.ate.

In [19]:

```
plt.figure(figsize = (12,5))
sns.boxplot(y = 'Cholesterol', hue = 'Category', data = df)
plt.legend(ncols = df['Category'].nunique()/3)
plt.show()
```



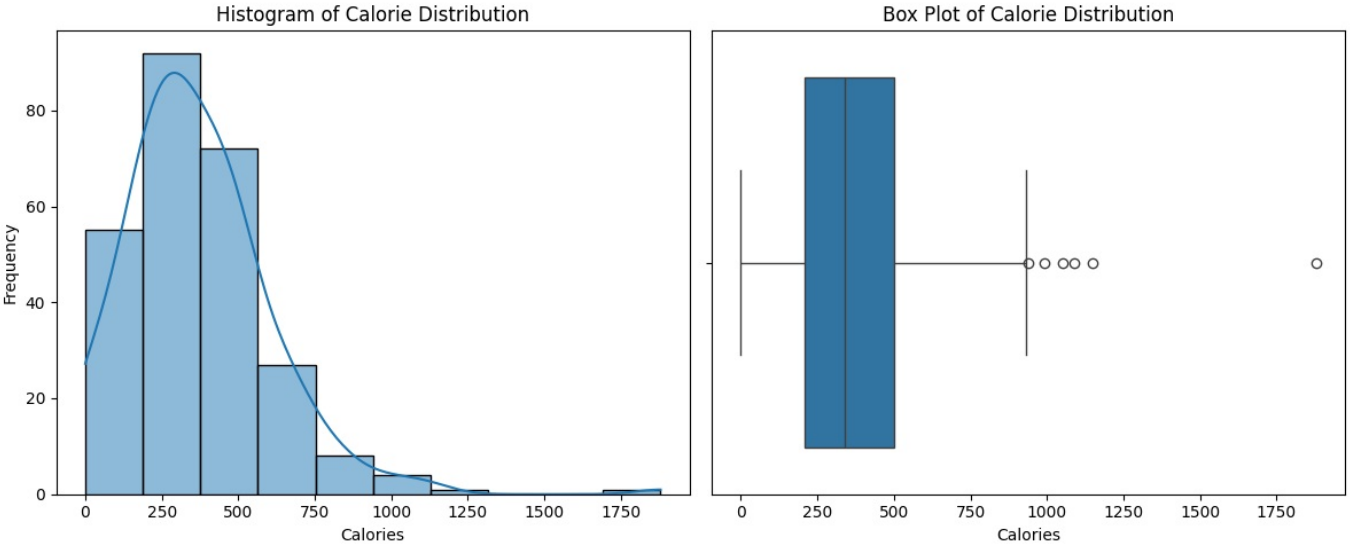
Data Visualization

To understand the calorie distribution and nutritional content of the menu items, I will create several visualizations. First, a histogram and a box plot will show the range and distribution of calorie counts, highlighting any outliers or skewness. Next, bar charts will compare the nutritional content (total fat, saturated fat, carbohydrates, and protein) of different food categories like burgers, salads, and desserts. These visuals will help identify differences in nutrient profiles, providing insights for consumer choices and menu development.

Distribution of Calories

In [42]:

```
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.histplot(df['Calories'], bins=10, kde=True)
plt.title('Histogram of Calorie Distribution')
plt.xlabel('Calories')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
sns.boxplot(x=df['Calories'])
plt.title('Box Plot of Calorie Distribution')
plt.xlabel('Calories')

plt.tight_layout()
plt.show()
```



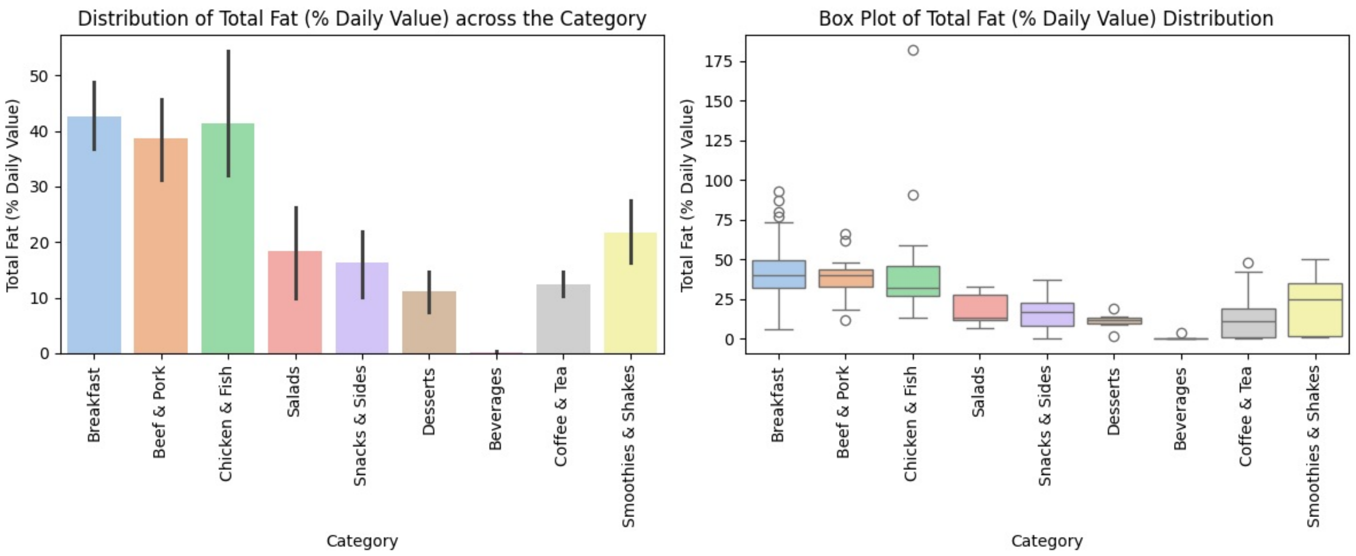
Conclusion: The histogram is right-skewed, peaking around 125-500 calories, with a long tail towards higher calorie values. Most menu items have lower calorie counts, with a few high-calorie outliers. The box plot supports this, showing a median of around 340 calories, with the 25th and 75th percentiles at 210 and 500, respectively. Several outliers are present, including high-calorie items.

Using bar charts and box plots, we compare the nutritional characteristics of various food categories. Instead of visualizing all 22 nutrients

In [43]:

```
# Total Fat (% Daily Value)

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.barplot(data= df, x= 'Category', y= 'Total Fat (% Daily Value)', palette= 'pastel')
plt.title('Distribution of Total Fat (% Daily Value) across the Category')
plt.xlabel('Category')
plt.ylabel('Total Fat (% Daily Value)')
plt.xticks(rotation= 90)

plt.subplot(1, 2, 2)
sns.boxplot(data= df, x= 'Category', y='Total Fat (% Daily Value)',palette= 'pastel')
plt.title('Box Plot of Total Fat (% Daily Value) Distribution')
plt.xlabel('Category')
plt.ylabel('Total Fat (% Daily Value)')
plt.xticks(rotation= 90)
plt.tight_layout()
plt.show()
```

```
C:\Users\nehak\AppData\Local\Temp\ipykernel_18608\3764116270.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(data= df, x= 'Category', y= 'Total Fat (% Daily Value)', palette= 'pastel')
C:\Users\nehak\AppData\Local\Temp\ipykernel_18608\3764116270.py:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.boxplot(data= df, x= 'Category', y='Total Fat (% Daily Value)',palette= 'pastel')
```



Conclusion:

Bar Plot: The bar plot shows that the Breakfast category has the highest average Total Fat (% Daily Value) at about 43%, followed by the Chicken & Fish category at around 40%. The Beverages category has the lowest average Total Fat (% Daily Value), with negligible values.

Box Plot (Top Two Categories with Highest Average Total Fat % Daily Value):

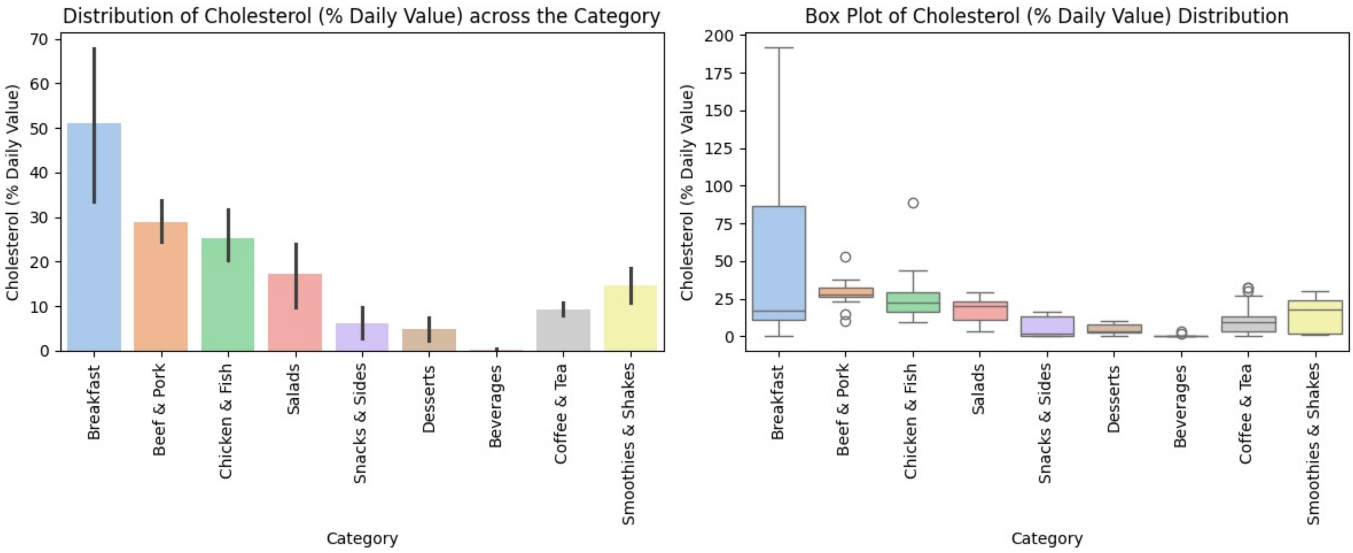
Breakfast: The box plot reveals a relatively even distribution of Total Fat (% Daily Value) in breakfast items, with a median of 43%. The 25th percentile is around 30%, and the 75th percentile is 50%, with several high-value outliers.

Chicken & Fish: The box plot shows a skewed distribution for Chicken & Fish items, with a median of about 40%. The 25th percentile is around 28%, and the 75th percentile is about 48%, with a few high-value outliers.

In [44]:

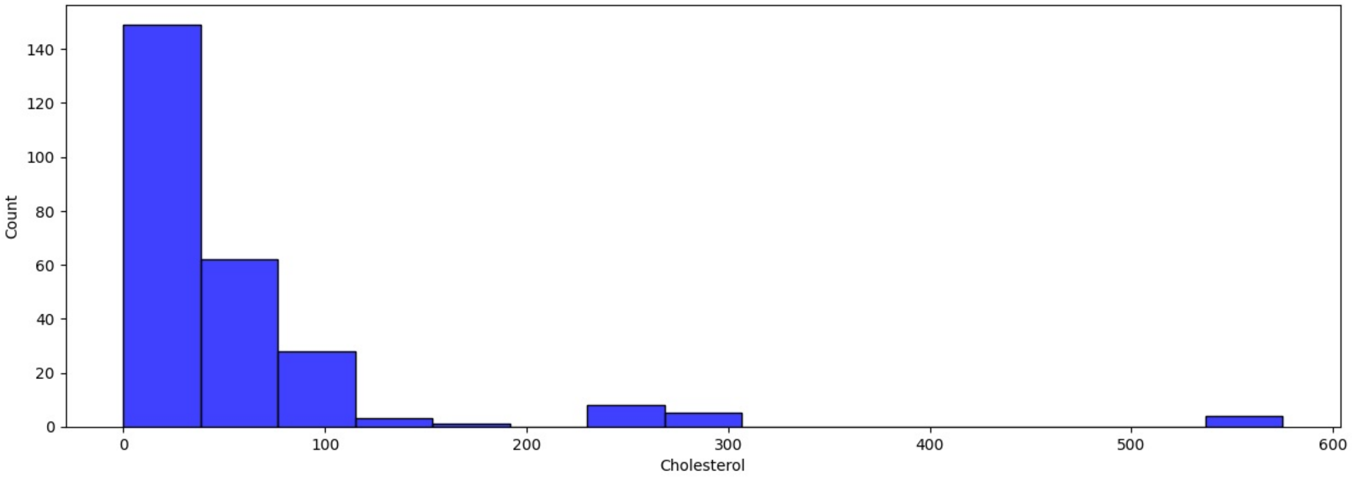
```python
# Cholesterol (% Daily Value)

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.barplot(data= df, x= 'Category', y= 'Cholesterol (% Daily Value)', palette= 'pastel')
plt.title('Distribution of Cholesterol (% Daily Value) across the Category')
plt.xlabel('Category')
plt.ylabel('Cholesterol (% Daily Value)')
plt.xticks(rotation= 90)

plt.subplot(1, 2, 2)
sns.boxplot(data= df, x= 'Category', y='Cholesterol (% Daily Value)',palette= 'pastel')
plt.title('Box Plot of Cholesterol (% Daily Value) Distribution')
plt.xlabel('Category')
plt.ylabel('Cholesterol (% Daily Value)')
plt.xticks(rotation= 90)
plt.tight_layout()
plt.show()
```

```
C:\Users\nehak\AppData\Local\Temp\ipykernel_18608\3763559197.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(data= df, x= 'Category', y= 'Cholesterol (% Daily Value)', palette= 'pastel')
C:\Users\nehak\AppData\Local\Temp\ipykernel_18608\3763559197.py:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.boxplot(data= df, x= 'Category', y='Cholesterol (% Daily Value)',palette= 'pastel')
```



In [52]:

```
plt.figure(figsize = (15,5))
sns.histplot(x = 'Cholesterol', data = df, color = 'blue', bins = 15)
plt.show()
```



Conclusion

Histogram: Most menu items have a Cholestrol within the range between 0 and 40

Bar Plot: The Breakfast category has the highest average Cholesterol (% Daily Value) at around 50%, followed by the Beef & Pork category at 30% and the Chicken & Fish category at 25%. The Beverages category has negligible cholesterol content.

Box Plot (Top Two Categories with Highest Average Cholesterol % Daily Value): Breakfast: The Breakfast category shows a wide range of cholesterol values, with some items reaching up to 90% of the daily value. The median cholesterol level is also relatively high.

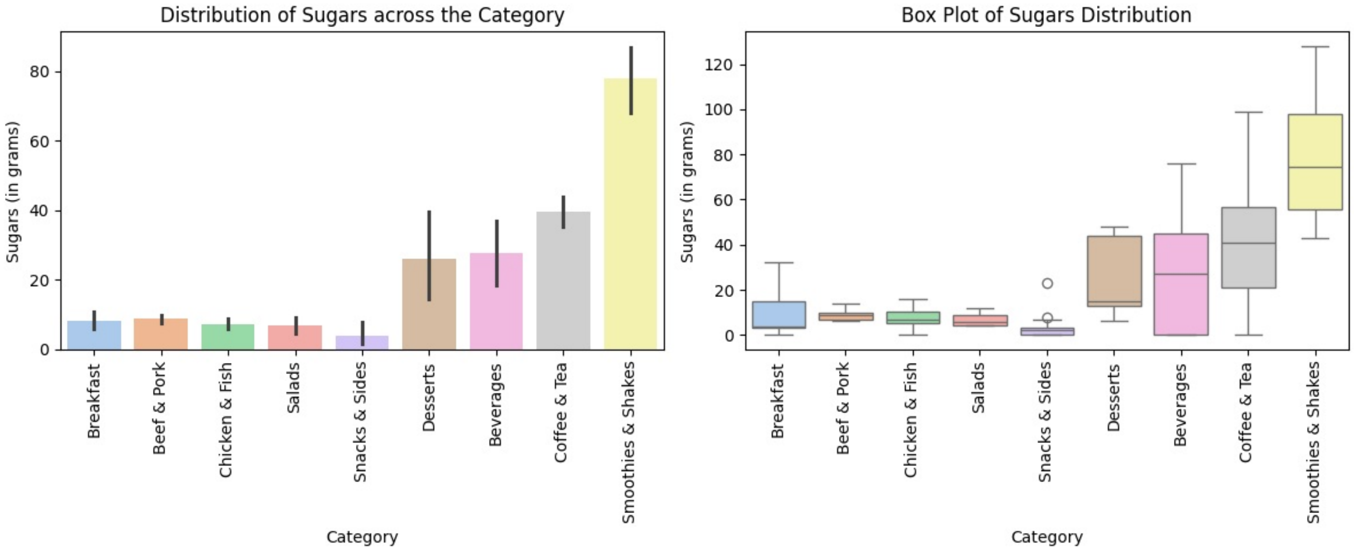
Beef & Pork: The Beef & Pork category has a more compact distribution, with most values between 25-35% of the daily cholesterol value. There are a few high-value outliers in both the Chicken & Fish and Beef & Pork categories, indicating some items have exceptionally high cholesterol content.

In [50]:

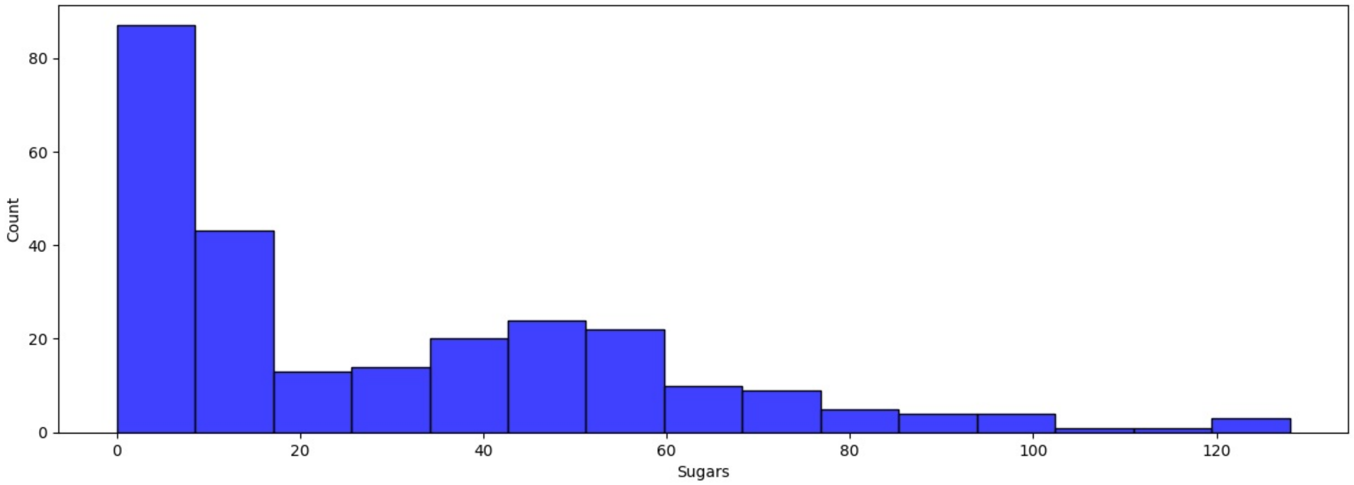
```
# Sugars (in grams)

plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.barplot(data= df, x= 'Category', y= 'Sugars', palette= 'pastel')
plt.title('Distribution of Sugars across the Category')
plt.xlabel('Category')
plt.ylabel('Sugars (in grams)')
plt.xticks(rotation= 90)

plt.subplot(1, 2, 2)
```

```
sns.boxplot(data= df, x= 'Category', y='Sugars',palette= 'pastel')
plt.title('Box Plot of Sugars Distribution')
plt.xlabel('Category')
plt.ylabel('Sugars (in grams)')
plt.xticks(rotation= 90)
plt.tight_layout()
plt.show()
```

```
C:\Users\nehak\AppData\Local\Temp\ipykernel_18608\269524182.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(data= df, x= 'Category', y= 'Sugars', palette= 'pastel')
C:\Users\nehak\AppData\Local\Temp\ipykernel_18608\269524182.py:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.boxplot(data= df, x= 'Category', y='Sugars',palette= 'pastel')
```



In [51]:
```
plt.figure(figsize = (15,5))
sns.histplot(x = 'Sugars', data = df, color = 'blue', bins = 15)
plt.show()
```



Conclusion

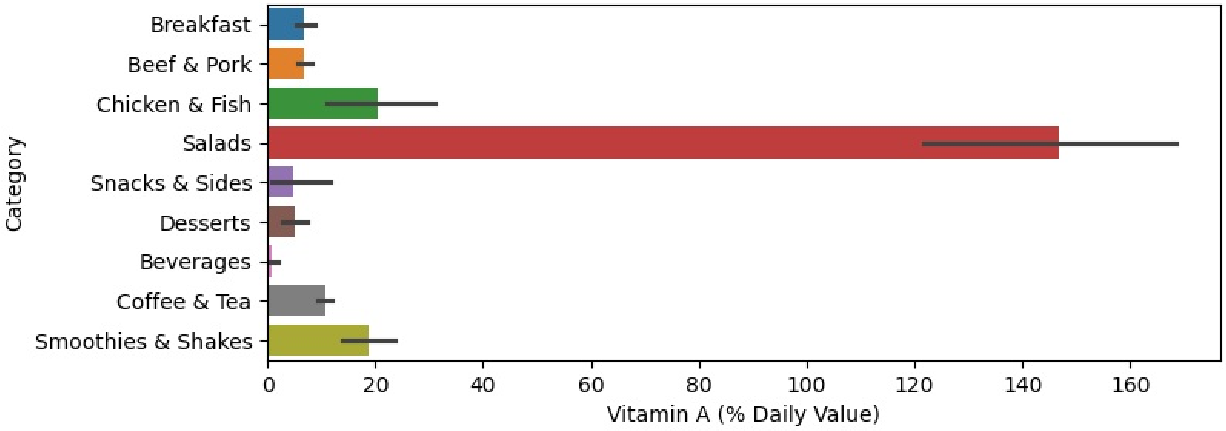
Histogram: Most menu items have a sugar level between 0 and 10

Bar Plot: The Smoothies & Shakes category has the highest average sugar content, with around 80 grams per serving. The Coffee & Tea category follows, with an average of 40 grams of sugar per serving. The Snacks & Sides category has the lowest average sugar content, with around 8 grams per serving.

Box Plot (Top Two Categories with Highest Average Sugars in Grams): Smoothies & Shakes: This category has the broadest range of sugar content, with some items reaching up to 100 grams per serving. Coffee & Tea: This category also shows a wide range, with some items containing up to 60 grams of sugar per serving. Overall, apart from the Snacks & Sides category, which has the lowest sugar content, there are no significant outliers in sugar content across all categories

In [45]:
```
plt.figure(figsize = (8,3))
```

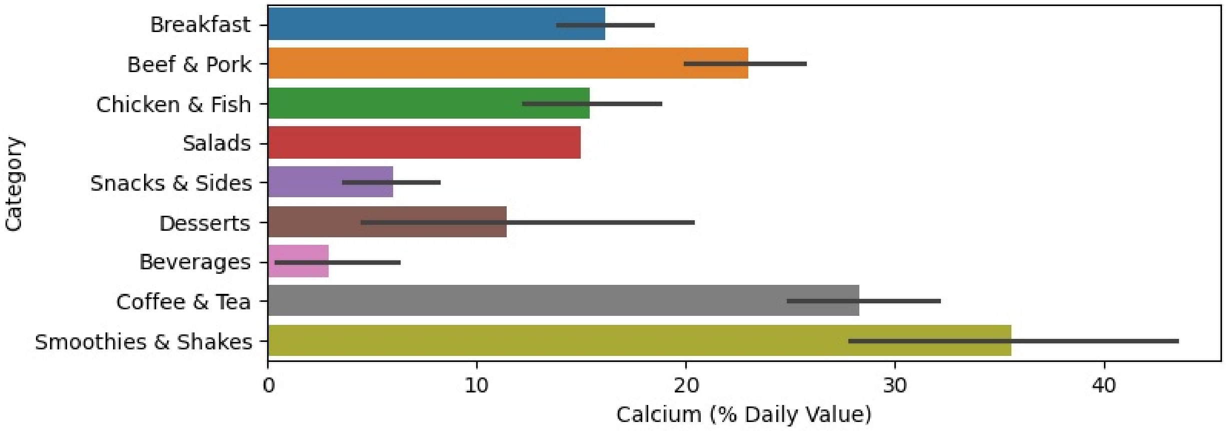
```python
sns.barplot(y = 'Category', x = 'Vitamin A (% Daily Value)', data = df, hue = 'Category')
plt.show()
```



Conclusion: Beverages are poor source and **Salads** are rich source of Vitamin A

In [46]:

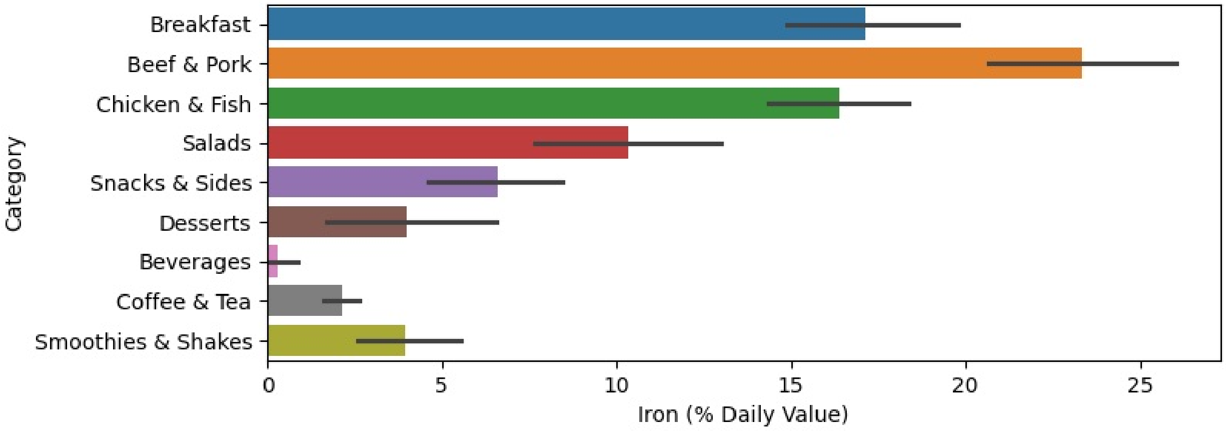
```python
plt.figure(figsize = (8,3))
sns.barplot(y = 'Category', x = 'Calcium (% Daily Value)', data = df, hue = 'Category')
plt.show()
```



Conclusion: Beverages are poor source and **Smoothies & Shakes** are rich source of Calcium

In [47]:

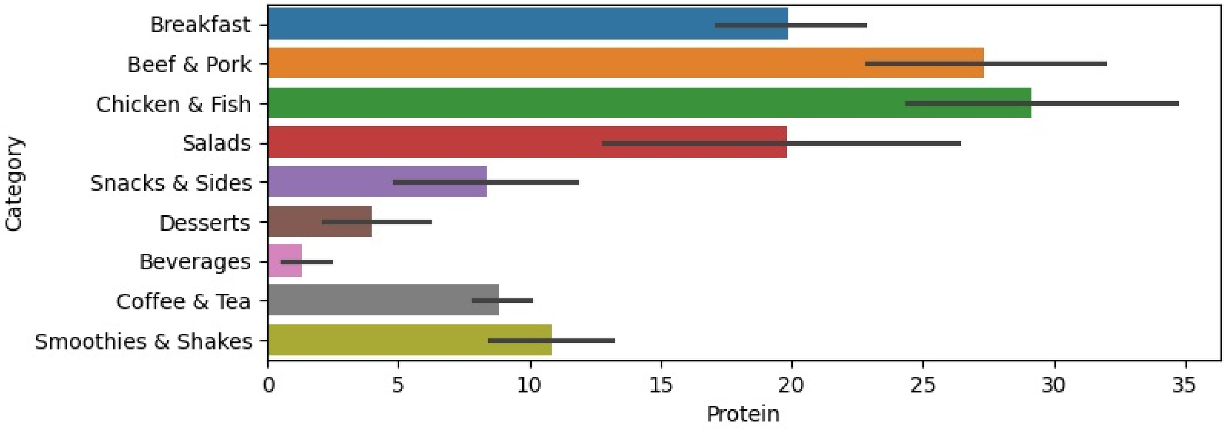
```python
plt.figure(figsize = (8,3))
sns.barplot(y = 'Category', x = 'Iron (% Daily Value)', data = df, hue = 'Category')
plt.show()
```



Conclusion: Beverages are poor source and **Beef & Pork** are rich source of Iron

In [48]:

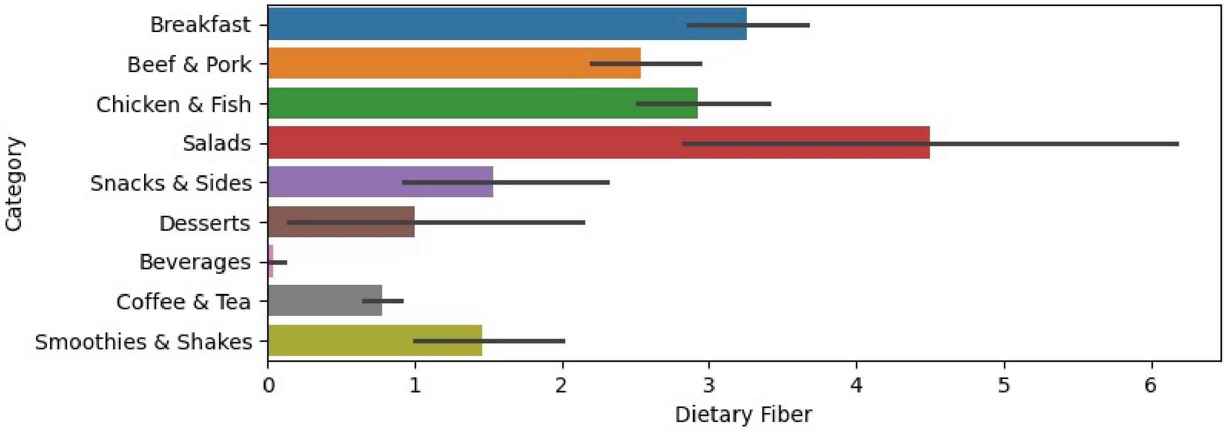
```python
plt.figure(figsize = (8,3))
sns.barplot(y = 'Category', x = 'Protein', data = df, hue = 'Category')
plt.show()
```



Conclusion: Beverages are poor source and **Chicken and Fish** are rich source of Protien

In [49]:

```
plt.figure(figsize = (8,3))
sns.barplot(y = 'Category', x = 'Dietary Fiber', data = df, hue = 'Category')
plt.show()
```



Conclusion: Beverages are poor source and **Salads** are rich source of Fiber

Identify menu items with the highest and lowest calorie counts.

In [20]:

```
highest_calories = df.loc[df['Calories'].idxmax()]
print('Item with the highest calorie count: ')
print(highest_calories)
```

```
Item with the highest calorie count:
Category                                       Chicken & Fish
Item                                 Chicken McNuggets (40 piece)
Serving Size                                   22.8 oz (646 g)
Calories                                                 1880
Calories from Fat                                        1060
Total Fat                                               118.0
Total Fat (% Daily Value)                                 182
Saturated Fat                                            20.0
Saturated Fat (% Daily Value)                             101
Trans Fat                                                 1.0
Cholesterol                                               265
Cholesterol (% Daily Value)                                89
Sodium                                                   3600
Sodium (% Daily Value)                                    150
Carbohydrates                                             118
Carbohydrates (% Daily Value)                              39
Dietary Fiber                                               6
Dietary Fiber (% Daily Value)                              24
Sugars                                                      1
Protein                                                    87
Vitamin A (% Daily Value)                                   0
Vitamin C (% Daily Value)                                  15
Calcium (% Daily Value)                                     8
Iron (% Daily Value)                                       25
Name: 82, dtype: object
```

Conclusion: Chicken McNuggets (40 piece) has the highest calorie count

In [21]:

```
lowest_calories = df.loc[df['Calories'].idxmin()]
print('Item with the lowest calorie count: ')
print(lowest_calories)
```

```
Item with the lowest calorie count:
Category                                   Beverages
Item                                Diet Coke (Small)
Serving Size                             16 fl oz cup
Calories                                            0
Calories from Fat                                   0
Total Fat                                         0.0
Total Fat (% Daily Value)                           0
Saturated Fat                                     0.0
Saturated Fat (% Daily Value)                       0
Trans Fat                                         0.0
Cholesterol                                         0
Cholesterol (% Daily Value)                         0
Sodium                                             10
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
Name: 114, dtype: object
```

Conclusion: Diet Coke (Small) has the lowest calorie count

Determine the average nutritional content of popular menu categories.

In [54]:

```
columns_to_drop = ['Item', 'Serving Size']
df_new = df.drop(columns_to_drop, axis=1)
df_new.head()
```

Out[54]:

	Category	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Cholesterol	Cholesterol (% Daily Value)	...	Carbohydrates	Carbohydi (% I Va
0	Breakfast	300	120	13.0	20	5.0	25	0.0	260	87	...	31	
1	Breakfast	250	70	8.0	12	3.0	15	0.0	25	8	...	30	
2	Breakfast	370	200	23.0	35	8.0	42	0.0	45	15	...	29	
3	Breakfast	450	250	28.0	43	10.0	52	0.0	285	95	...	30	
4	Breakfast	400	210	23.0	35	8.0	42	0.0	50	16	...	30	

5 rows × 22 columns

In [55]:

```
average_nutritional_content = df_new.groupby('Category').mean()
average_nutritional_content
```

Out[55]:

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Cholesterol	Cholesterol (% Daily Value)	Sodium	.
Category											
Beef & Pork	494.000000	224.666667	24.866667	38.600000	10.466667	52.000000	1.100000	87.333333	28.933333	1020.666667	.
Beverages	113.703704	0.740741	0.092593	0.148148	0.055556	0.296296	0.000000	0.555556	0.185185	41.481481	.
Breakfast	526.666667	248.928571	27.690476	42.666667	10.654762	53.428571	0.107143	152.857143	50.952381	1211.071429	.
Chicken & Fish	552.962963	242.222222	26.962963	41.333333	6.166667	31.111111	0.129630	75.370370	25.222222	1257.777778	.
Coffee & Tea	283.894737	71.105263	8.021053	12.357895	4.921053	24.368421	0.142105	27.263158	9.378947	136.894737	.
Desserts	222.142857	64.285714	7.357143	11.142857	4.285714	21.285714	0.000000	15.000000	4.857143	117.142857	.
Salads	270.000000	108.333333	11.750000	18.333333	3.750000	18.500000	0.000000	51.666667	17.333333	588.333333	.
Smoothies & Shakes	531.428571	127.678571	14.125000	21.714286	8.375000	41.785714	0.535714	45.000000	14.714286	183.571429	.
Snacks & Sides	245.769231	94.615385	10.538462	16.230769	2.692308	13.384615	0.000000	18.461538	6.230769	395.769231	.

9 rows × 21 columns

Findings and Insights:

Menu Items Analysis:

The dataset includes various categories such as Breakfast, Beef & Pork, Chicken & Fish, Coffee & Tea, etc. "Chicken McNuggets (40 pieces)" in the Chicken & Fish category has the highest calorie count, while "Diet Coke (small)" in Beverages has the lowest. Beef & Pork items tend to have higher protein content, while Breakfast items often have more fat and carbohydrates. Beverages, Smoothies & Shakes, and Coffee & Tea categories, although lower in calories, fat, and carbohydrates, show a significant spike in sugar levels and very low average protein content. Average Nutritional Content:

Calories:Chicken & Fish items are high in calories, averaging 552 calories, making them a calorie-rich option. Total Fat:Breakfast and Chicken & Fish categories have high total fat values, averaging 27.69 and 26.9 grams respectively. Protein:Chicken & Fish items have the highest average protein content, at 29.11 grams. Cholesterol:Breakfast items have a high average cholesterol level, at 153 grams. Carbohydrates:

While protein-rich categories like Chicken, Fish, Beef, and Pork are notable, the Smoothies & Shakes category has significantly higher carbohydrate levels, averaging 90.5 grams.

Beverages are poor source and Salads are rich source of **Vitamin A.** Beverages are poor source and Smoothies & Shakes are rich source of **Calcium.** Beverages are poor source and Beef & Pork are rich source of **Iron.** Beverages are poor source and Salads are rich **source of Fiber.**

Conclusions:

Healthier Options: Customers can identify healthier choices like the Salads, Egg White Delight, Premium Grilled Chicken Classic Sandwich, and Fruit & Maple Oatmeal without Brown Sugar, which are lower in calories, fat, and sodium, supporting balanced diets.

Nutrient Sources:

Vitamin A: Salads are a rich source, while beverages are a poor source. Calcium: Smoothies & Shakes are rich in calcium, whereas beverages are poor sources. Iron: Beef & Pork are rich sources of iron, with beverages being poor sources. Fiber: Salads are rich in fiber, whereas beverages are poor sources. Advice for Customers:

Breakfast Items: Opt for lighter choices like "Fruit & Maple Oatmeal" or "Hash Brown." Beef & Pork: Choose leaner options like "Hamburger" or "Cheeseburger." Chicken & Fish: Better choices include "Premium Crispy Chicken Classic Sandwich" and "Premium Grilled Chicken Ranch BLT Sandwich." Portion Sizes: Choose smaller portions to limit intake of unhealthy nutrients. Side Items: Add healthier sides like "Hash Brown" or "Fruit & Maple Oatmeal" to balance meals. Recommendations to Improve McDonald's Menu Nutritional Profile:

Increase Healthy Options: Add more low-calorie, low-fat, and low-sodium items like salads, grilled chicken, and fruit-based sides. Nutritional Information Transparency: Display nutritional and allergen information prominently on menus and packaging. Reduce Added Sugars: Lower added sugars in beverages, desserts, and breakfast items. Promote Balanced Meals: Create meal deals with lean protein, whole grains, and vegetables. Offer combo meals with side salads or fruit. Benefit of Nutritional Analysis:

Benefit for Customers:

Informed Choices: Helps customers select options that fit their dietary preferences and health goals. Health Conscious Decisions: Identifies healthier options with lower calories, fat, and sodium. Dietary Restrictions and Preferences: Empowers customers to choose items that meet their nutritional needs. Benefit for McDonald's:

Menu Development: Guides the creation of a balanced menu to meet customer preferences. Customer Satisfaction: Enhances customer experience with transparency in nutritional information. Health and Wellness Initiatives: Aligns McDonald's with health trends and promotes healthier options, attracting health-conscious customers. In conclusion, nutritional analysis benefits both McDonald's and its customers by supporting informed food choices, health-conscious decisions, and menu development strategies for diverse dietary needs.

In [ ]: