In [47]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [48]:
```
df = pd.read_csv('C:/Users/admin/Downloads/Nutrical Dataset.csv')
```

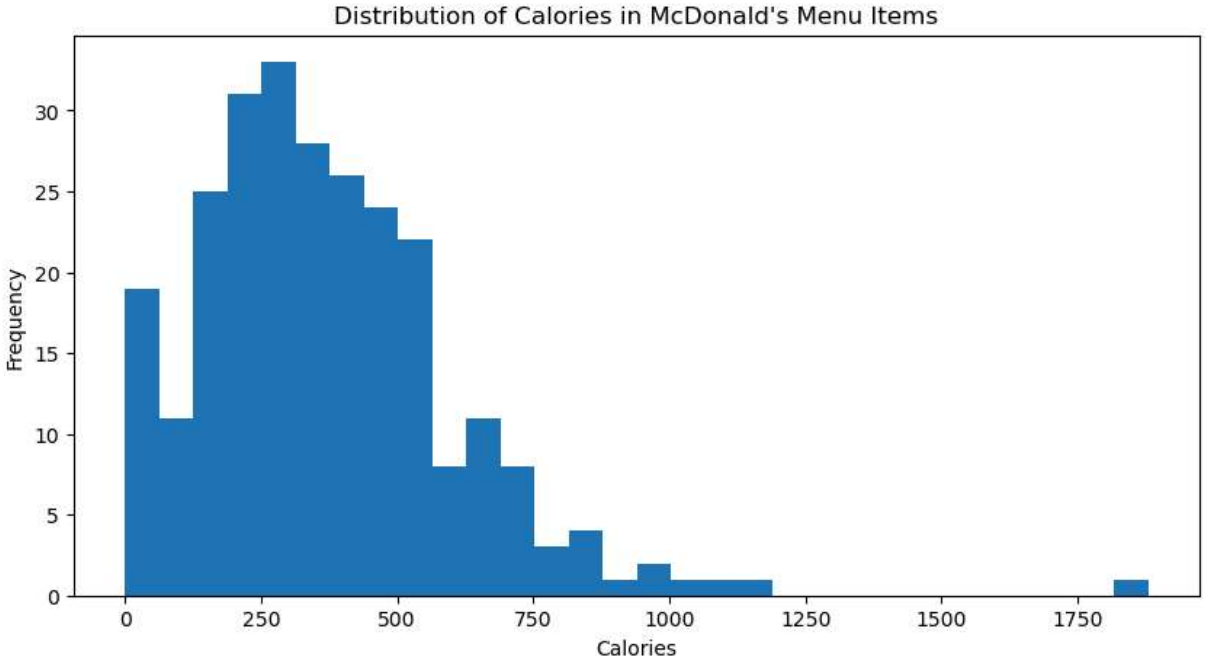
In [49]:
```
# Data preprocessing
print(df.head())  # Inspect the first few rows
print(df.info())  # Check for missing values and data types
```

```
    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]
<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
None
```

In [50]:
```
# Handle missing values (if necessary)
# df.fillna(method='ffill', inplace=True)  # Forward fill missing values

# Exploratory Data Analysis (EDA)
# Analyze calorie distribution
print(df['Calories'].describe())
plt.figure(figsize=(10, 5))
plt.hist(df['Calories'], bins=30)
plt.xlabel('Calories')
plt.ylabel('Frequency')
plt.title('Distribution of Calories in McDonald\'s Menu Items')
plt.show()
```

```
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
```



In [51]:
```
print(df.columns.tolist())
```

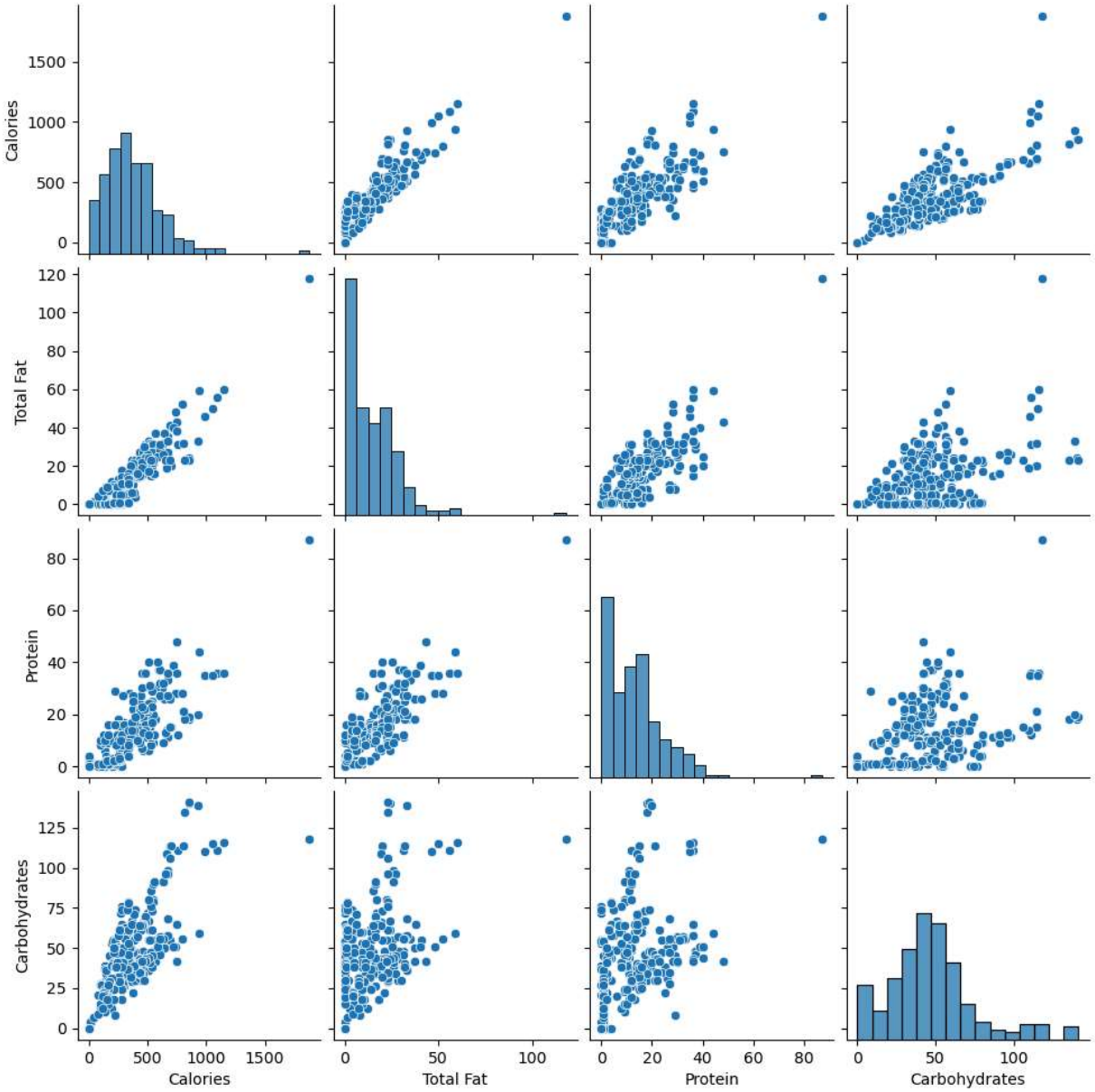
```
['Category', 'Item', 'Serving Size', 'Calories', 'Calories from Fat', 'Total Fat',
'Total Fat (% Daily Value)', 'Saturated Fat', 'Saturated Fat (% Daily Value)', 'Tran
s Fat', 'Cholesterol', 'Cholesterol (% Daily Value)', 'Sodium', 'Sodium (% Daily Val
ue)', 'Carbohydrates', 'Carbohydrates (% Daily Value)', 'Dietary Fiber', 'Dietary Fi
ber (% Daily Value)', 'Sugars', 'Protein', 'Vitamin A (% Daily Value)', 'Vitamin C
(% Daily Value)', 'Calcium (% Daily Value)', 'Iron (% Daily Value)']
```

In [52]:
```
# Explore nutritional content
print(df.groupby('Category')[['Calories', 'Total Fat', 'Protein', 'Carbohydrates']]
```

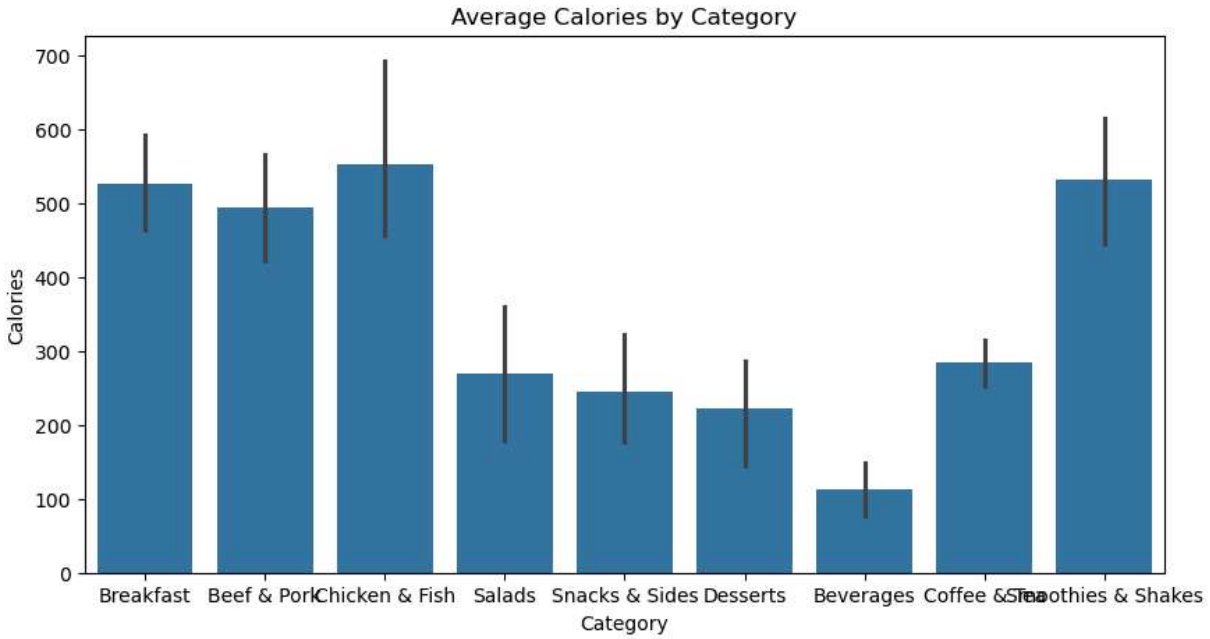
Category	Calories	Total Fat	Protein	Carbohydrates
Beef & Pork	494.000000	24.866667	27.333333	40.133333
Beverages	113.703704	0.092593	1.333333	28.814815
Breakfast	526.666667	27.690476	19.857143	49.761905
Chicken & Fish	552.962963	26.962963	29.111111	49.074074
Coffee & Tea	283.894737	8.021053	8.863158	44.526316
Desserts	222.142857	7.357143	4.000000	34.857143
Salads	270.000000	11.750000	19.833333	21.666667
Smoothies & Shakes	531.428571	14.125000	10.857143	90.428571
Snacks & Sides	245.769231	10.538462	8.384615	29.153846

In [53]:
```
# Identify trends and patterns
sns.pairplot(df[['Calories', 'Total Fat', 'Protein', 'Carbohydrates']])
```

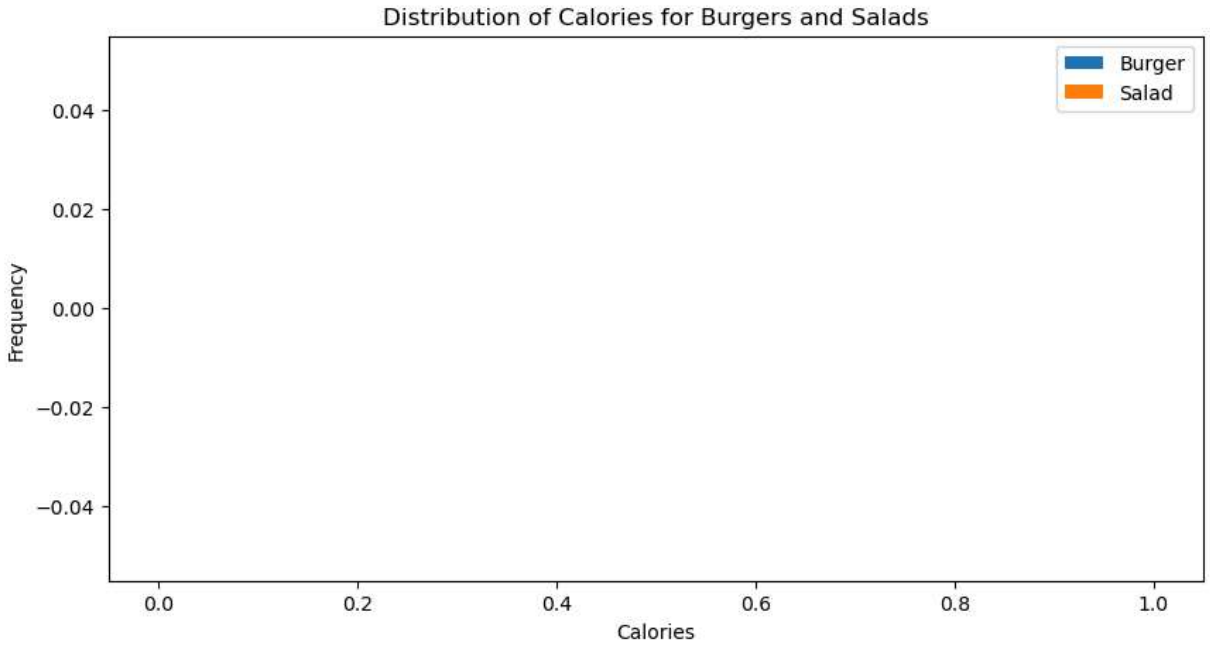
Out[53]:
```
<seaborn.axisgrid.PairGrid at 0x1d9a61d5bb0>
```



In [54]:
```
# Data visualization
# Create bar charts
plt.figure(figsize=(10, 5))
sns.barplot(x='Category', y='Calories', data=df)
plt.xlabel('Category')
plt.ylabel('Calories')
plt.title('Average Calories by Category')
plt.show()
```

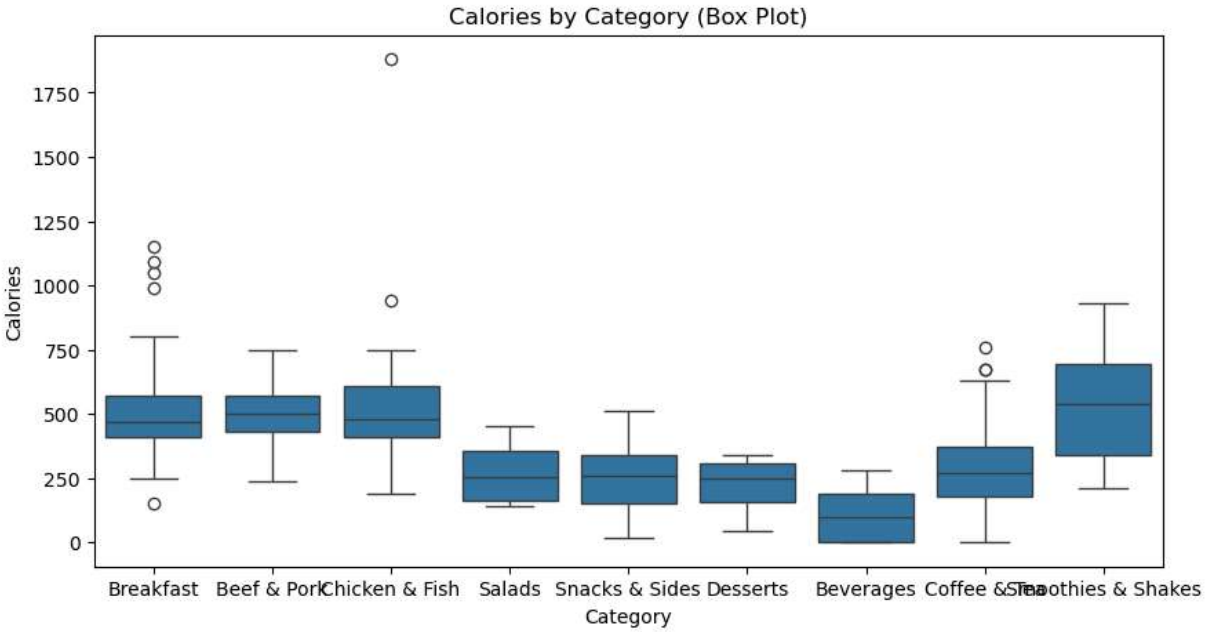


In [55]:
```
# Create histograms
plt.figure(figsize=(10, 5))
plt.hist(df[df['Category'] == 'Burger']['Calories'], bins=30, label='Burger')
plt.hist(df[df['Category'] == 'Salad']['Calories'], bins=30, label='Salad')
plt.legend()
plt.xlabel('Calories')
plt.ylabel('Frequency')
plt.title('Distribution of Calories for Burgers and Salads')
plt.show()
```



In [56]:
```
# Create box plots
plt.figure(figsize=(10, 5))
sns.boxplot(x='Category', y='Calories', data=df)
plt.xlabel('Category')
plt.ylabel('Calories')
```

```
plt.title('Calories by Category (Box Plot)')
plt.show()
```



In [57]:
```
# Nutrition-based insights
highest_calories_item = df.loc[df['Calories'].idxmax()]
print("Item with the highest calories:", highest_calories_item)
```

```
Item with the highest calories: Category                                Chick
en & 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
```

In [58]:
```
lowest_calories_item = df.loc[df['Calories'].idxmin()]
print("Item with the lowest calories:", lowest_calories_item)
```

```
Item with the lowest calories: 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
```

In [59]:
```
print(df.columns.tolist())
```

```
['Category', 'Item', 'Serving Size', 'Calories', 'Calories from Fat', 'Total Fat',
'Total Fat (% Daily Value)', 'Saturated Fat', 'Saturated Fat (% Daily Value)', 'Tran
s Fat', 'Cholesterol', 'Cholesterol (% Daily Value)', 'Sodium', 'Sodium (% Daily Val
ue)', 'Carbohydrates', 'Carbohydrates (% Daily Value)', 'Dietary Fiber', 'Dietary Fi
ber (% Daily Value)', 'Sugars', 'Protein', 'Vitamin A (% Daily Value)', 'Vitamin C
(% Daily Value)', 'Calcium (% Daily Value)', 'Iron (% Daily Value)']
```

In [60]:
```
# Average nutritional content of popular categories
average_nutrition = df.groupby('Category')[['Calories', 'Total Fat', 'Protein', 'Ca
```

In [44]:
```
print(average_nutrition)
```

```
                    Calories  Total Fat    Protein  Carbohydrates
Category
Beef & Pork       494.000000  24.866667  27.333333      40.133333
Beverages         113.703704   0.092593   1.333333      28.814815
Breakfast         526.666667  27.690476  19.857143      49.761905
Chicken & Fish    552.962963  26.962963  29.111111      49.074074
Coffee & Tea      283.894737   8.021053   8.863158      44.526316
Desserts          222.142857   7.357143   4.000000      34.857143
Salads            270.000000  11.750000  19.833333      21.666667
Smoothies & Shakes 531.428571 14.125000  10.857143      90.428571
Snacks & Sides    245.769231  10.538462   8.384615      29.153846
```

In [40]:
```
# Documentation and reporting
# Summarize findings and insights here
# Explain how the analysis could benefit McDonald's customers and the organization
```

In [40]:
```
# Documentation and reporting
# Summarize findings and insights here
# Explain how the analysis could benefit McDonald's customers and the organization
```

In [ ]:
```

```