Upload the data

In [5]:
```
!pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\users\rh\anaconda3\lib\site-packages
(3.1.5)
Requirement already satisfied: et-xmlfile in c:\users\rh\anaconda3\lib\site-packages
(from openpyxl) (1.1.0)
```

In [3]:
```
import pandas as pd
df=pd.read_excel('cereal.xlsx',engine='openpyxl')
df.head(10)
```

Out[3]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vita
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	
5	Apple Cinnamon Cheerios	G	C	110	2	2	180	1.5	10.5	10	70	
6	Apple Jacks	K	C	110	2	0	125	1.0	11.0	14	30	
7	Basic 4	G	C	130	3	2	210	2.0	18.0	8	100	
8	Bran Chex	R	C	90	2	1	200	4.0	15.0	6	125	
9	Bran Flakes	P	C	90	3	0	210	5.0	13.0	5	190	

In [5]:
```
df.tail(10)
```

Out[5]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vit
67	Special K	K	C	110	6	0	230	1.0	16.0	3	55	
68	Strawberry Fruit Wheats	N	C	90	2	0	15	3.0	15.0	5	90	
69	Total Corn Flakes	G	C	110	2	1	200	0.0	21.0	3	35	
70	Total Raisin Bran	G	C	140	3	1	190	4.0	15.0	14	230	
71	Total Whole Grain	G	C	100	3	1	200	3.0	16.0	3	110	
72	Triples	G	C	110	2	1	250	0.0	21.0	3	60	
73	Trix	G	C	110	1	1	140	0.0	13.0	12	25	
74	Wheat Chex	R	C	100	3	1	230	3.0	17.0	3	115	
75	Wheaties	G	C	100	3	1	200	3.0	17.0	3	110	
76	Wheaties Honey Gold	G	C	110	2	1	200	1.0	16.0	8	60	

In [7]:
```
df.columns
```

Out[7]:
```
Index(['name', 'mfr', 'type', 'calories', 'protein', 'fat', 'sodium', 'fiber',
       'carbo', 'sugars', 'potass', 'vitamins', 'shelf', 'weight', 'cups',
       'rating'],
      dtype='object')
```

In [80]:
```
df.info
```

Out[80]:
```
<bound method DataFrame.info of                         name mfr type  calories
protein  fat  sodium  fiber  \
0                    100% Bran   N    C        70        4    1     130   10.0
1            100% Natural Bran   Q    C       120        3    5      15    2.0
2                     All-Bran   K    C        70        4    1     260    9.0
3   All-Bran with Extra Fiber   K    C        50        4    0     140   14.0
4               Almond Delight   R    C       110        2    2     200    1.0
..                         ...  ..  ...       ...      ...  ...     ...    ...
72                    Triples   G    C       110        2    1     250    0.0
73                       Trix   G    C       110        1    1     140    0.0
74                 Wheat Chex   R    C       100        3    1     230    3.0
75                    Wheaties   G    C       100        3    1     200    3.0
76         Wheaties Honey Gold   G    C       110        2    1     200    1.0

    carbo  sugars  potass  vitamins  shelf  weight  cups              rating
0     5.0       6     280        25      3     1.0  0.33  684029730000.0%%%%%
1     8.0       8     135         0      3     1.0  1.00  339836790000.0%%%%%
2     7.0       5     320        25      3     1.0  0.33  594255050000.0%%%%%
3     8.0       0     330        25      3     1.0  0.50  937049120000.0%%%%%
4    14.0       8      -1        25      3     1.0  0.75  343848430000.0%%%%%
..    ...     ...     ...       ...    ...     ...   ...                 ...
72   21.0       3      60        25      3     1.0  0.75  391061740000.0%%%%%
73   13.0      12      25        25      2     1.0  1.00  277533010000.0%%%%%
74   17.0       3     115        25      1     1.0  0.67  497874450000.0%%%%%
75   17.0       3     110        25      1     1.0  1.00  515921930000.0%%%%%
76   16.0       8      60        25      1     1.0  0.75  361875590000.0%%%%%

[77 rows x 16 columns]>
```

View the rating as %

In [16]:
```
import matplotlib.pyplot as plt
```

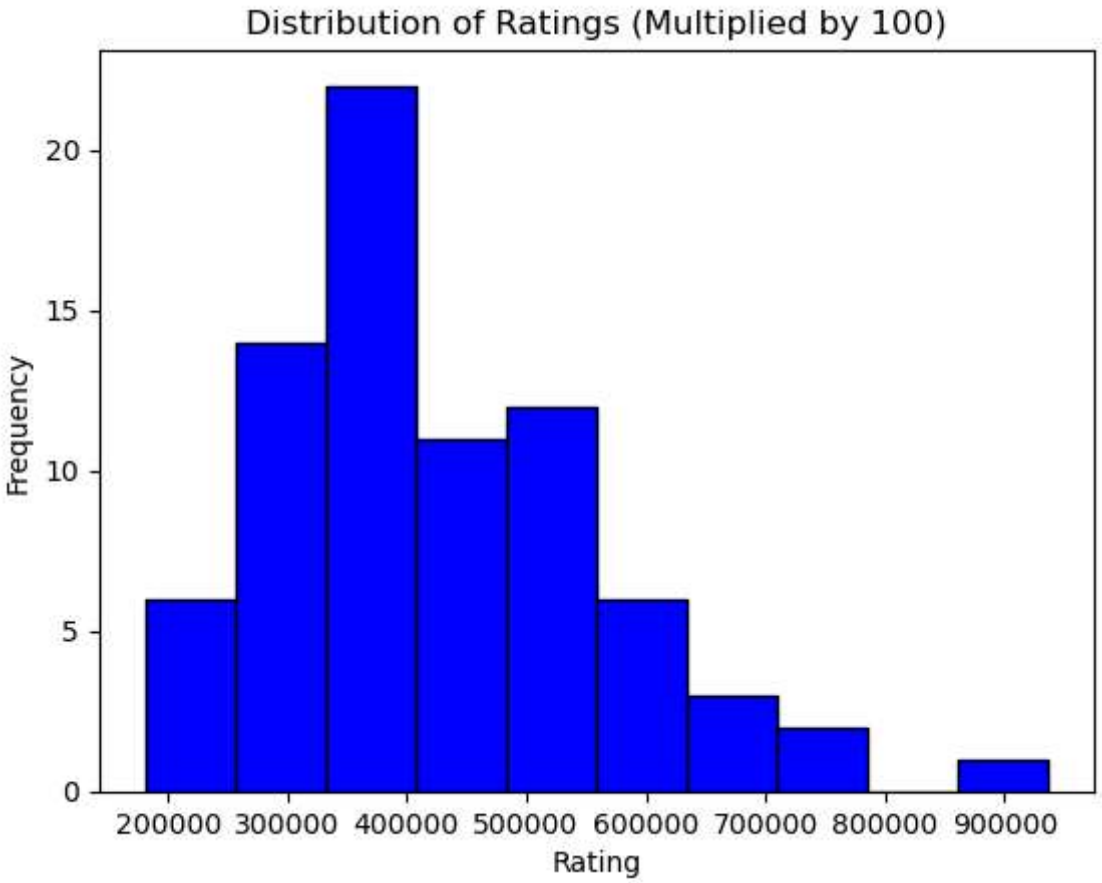
In [9]:
```
df.describe()
```

Out[9]:

	calories	protein	fat	sodium	fiber	carbo	sugars	
count	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.(
mean	106.883117	2.545455	1.012987	159.675325	2.151948	14.597403	6.922078	96.(
std	19.484119	1.094790	1.006473	83.832295	2.383364	4.278956	4.444885	71.2
min	50.000000	1.000000	0.000000	0.000000	0.000000	-1.000000	-1.000000	-1.(
25%	100.000000	2.000000	0.000000	130.000000	1.000000	12.000000	3.000000	40.(
50%	110.000000	3.000000	1.000000	180.000000	2.000000	14.000000	7.000000	90.(
75%	110.000000	3.000000	2.000000	210.000000	3.000000	17.000000	11.000000	120.(
max	160.000000	6.000000	5.000000	320.000000	14.000000	23.000000	15.000000	330.(

In [13]:
```
import matplotlib.pyplot as plt
df['rating'] =df['rating'] *100

# Create a simple plot to visualize the updated 'rating' values
plt.hist(df['rating'], bins=10, color='blue', edgecolor='black')
plt.title('Distribution of Ratings (Multiplied by 100)')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.show()
```



Calories and Healthiness Analysis

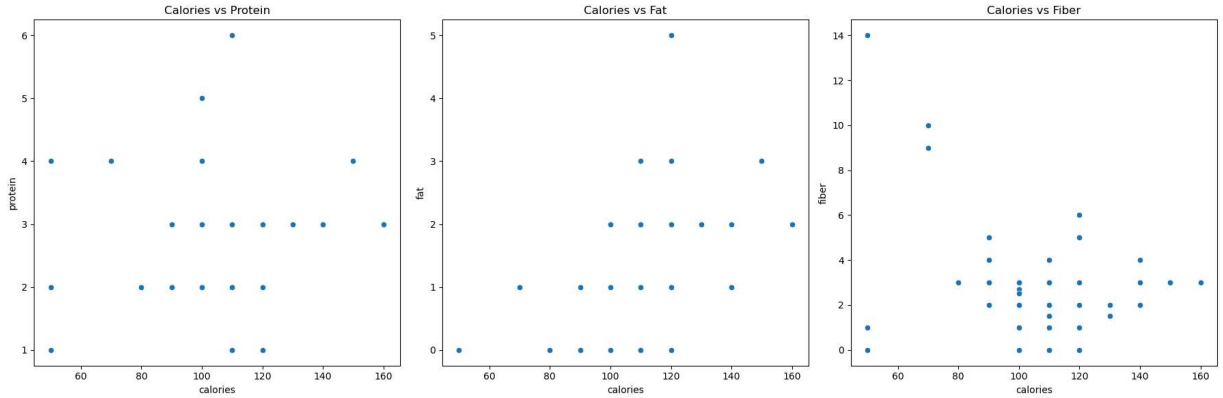
In [17]:
```
# Plot calories vs. protein, fat, fiber
import matplotlib.pyplot as plt
import seaborn as sns
fig, axes = plt.subplots(1, 3, figsize=(18, 6))
# Calories vs Protein
sns.scatterplot(x='calories', y='protein', data=df, ax=axes[0])
axes[0].set_title('Calories vs Protein')

# Calories vs Fat
sns.scatterplot(x='calories', y='fat', data=df, ax=axes[1])
axes[1].set_title('Calories vs Fat')

# Calories vs Fiber
sns.scatterplot(x='calories', y='fiber', data=df, ax=axes[2])
```

```
axes[2].set_title('Calories vs Fiber')

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



Nutrient Efficiency

In [21]:
```
# Create a new column for protein-to-calories ratio
df['protein_to_calories'] = df['protein'] / df['calories']

# Plot protein-to-calories ratio
sns.histplot(df['protein_to_calories'], kde=True)
plt.title('Protein-to-Calories Ratio Distribution')
plt.show()
```