

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

# Read the UScereal.csv dataset
cereal_df = pd.read_csv('UScereal.csv')

# Display necessary details about the dataset
print(cereal_df.info())
print(cereal_df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 65 entries, 0 to 64
```

```
Data columns (total 12 columns):
```

#	Column	Non-Null Count	Dtype
0	Name	65 non-null	object
1	mfr	65 non-null	object
2	calories	65 non-null	float64
3	protein	65 non-null	float64
4	fat	65 non-null	float64
5	sodium	65 non-null	float64
6	fibre	65 non-null	float64
7	carbo	65 non-null	float64
8	sugars	65 non-null	float64
9	shelf	65 non-null	int64
10	potassium	65 non-null	float64
11	vitamins	65 non-null	object

```
dtypes: float64(8), int64(1), object(3)
```

```
memory usage: 6.2+ KB
```

```
None
```

	Name	mfr	calories	protein	fat	sodium	fibre	\
0	100% Bran	N	212.12	12.12	3.03	393.94	30.30	
1	All-Bran	K	212.12	12.12	3.03	787.88	27.27	
2	All-Bran with Extra Fiber	K	100.00	8.00	0.00	280.00	28.00	
3	Apple Cinnamon Cheerios	G	146.67	2.67	2.67	240.00	2.00	
4	Apple Jacks	K	110.00	2.00	0.00	125.00	1.00	

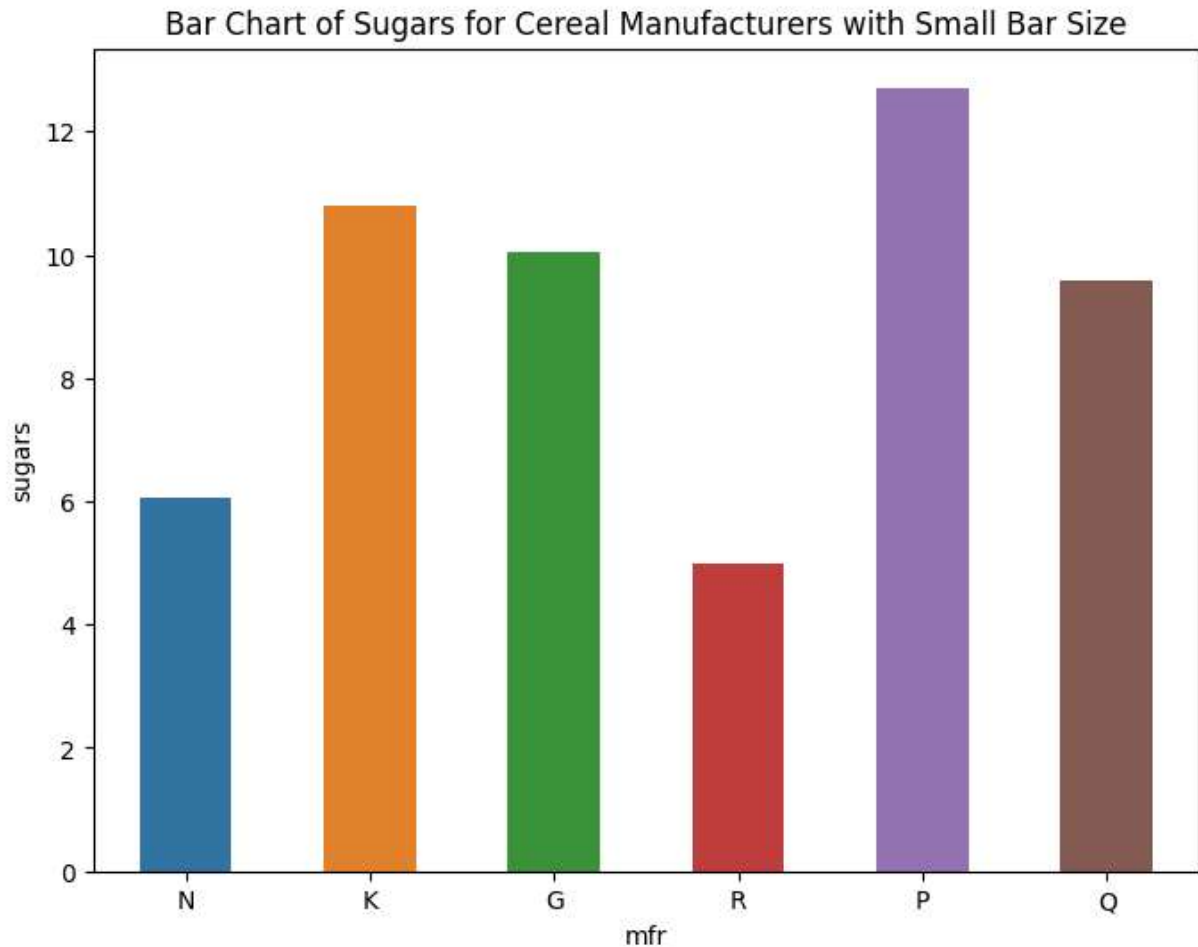
	carbo	sugars	shelf	potassium	vitamins
0	15.15	18.18	3	848.48	enriched
1	21.21	15.15	3	969.70	enriched
2	16.00	0.00	3	660.00	enriched
3	14.00	13.33	1	93.33	enriched
4	11.00	14.00	2	30.00	enriched

```
In [ ]: # Bar Chart with 2 variables (using 'sugars' column) and smaller bar size
plt.figure(figsize=(8, 6))
sns.barplot(x='mfr', y='sugars', data=cereal_df, ci=None, width=0.5)
plt.title('Bar Chart of Sugars for Cereal Manufacturers with Small Bar Size')
plt.show()
```

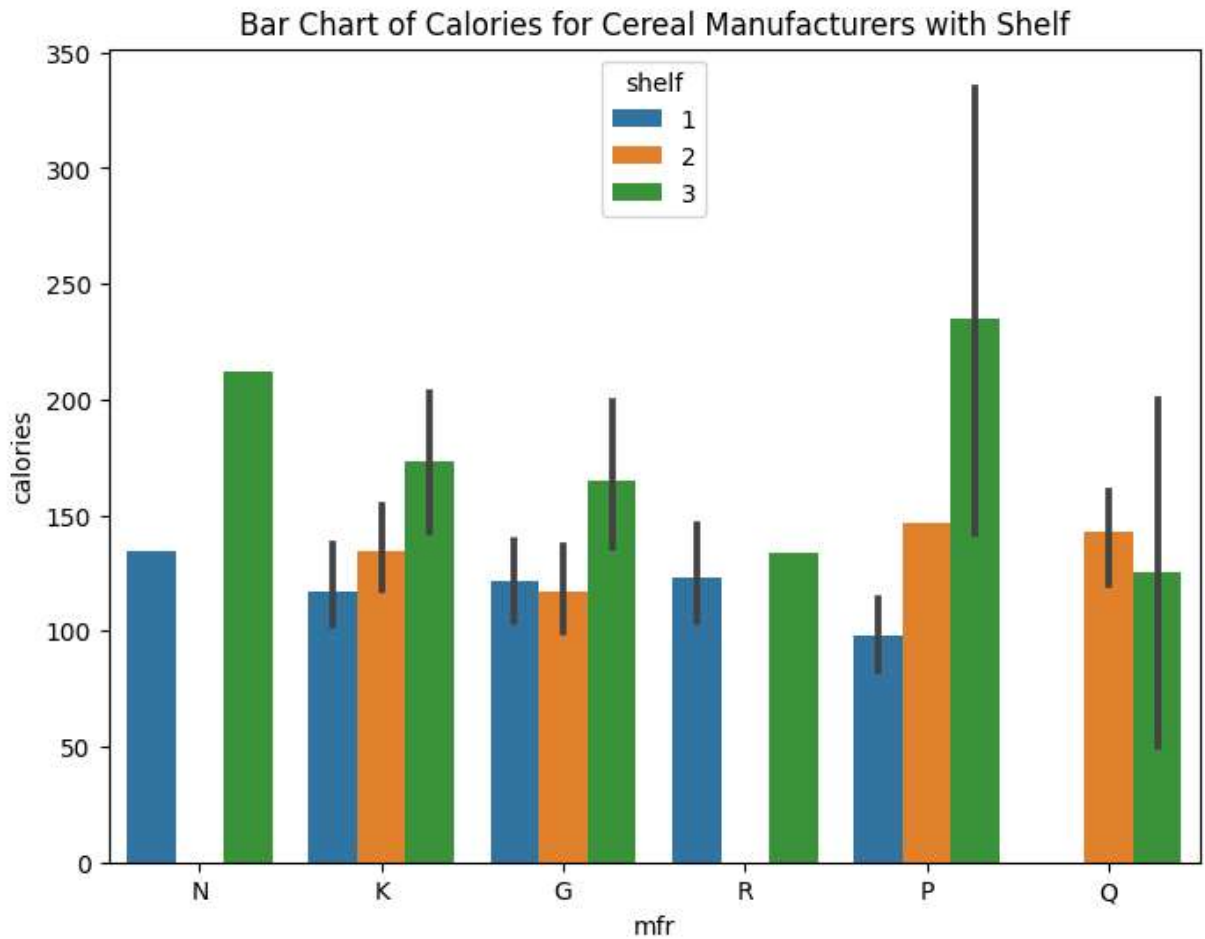
C:\Users\Kalpana\AppData\Local\Temp\ipykernel_2120\1180265433.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

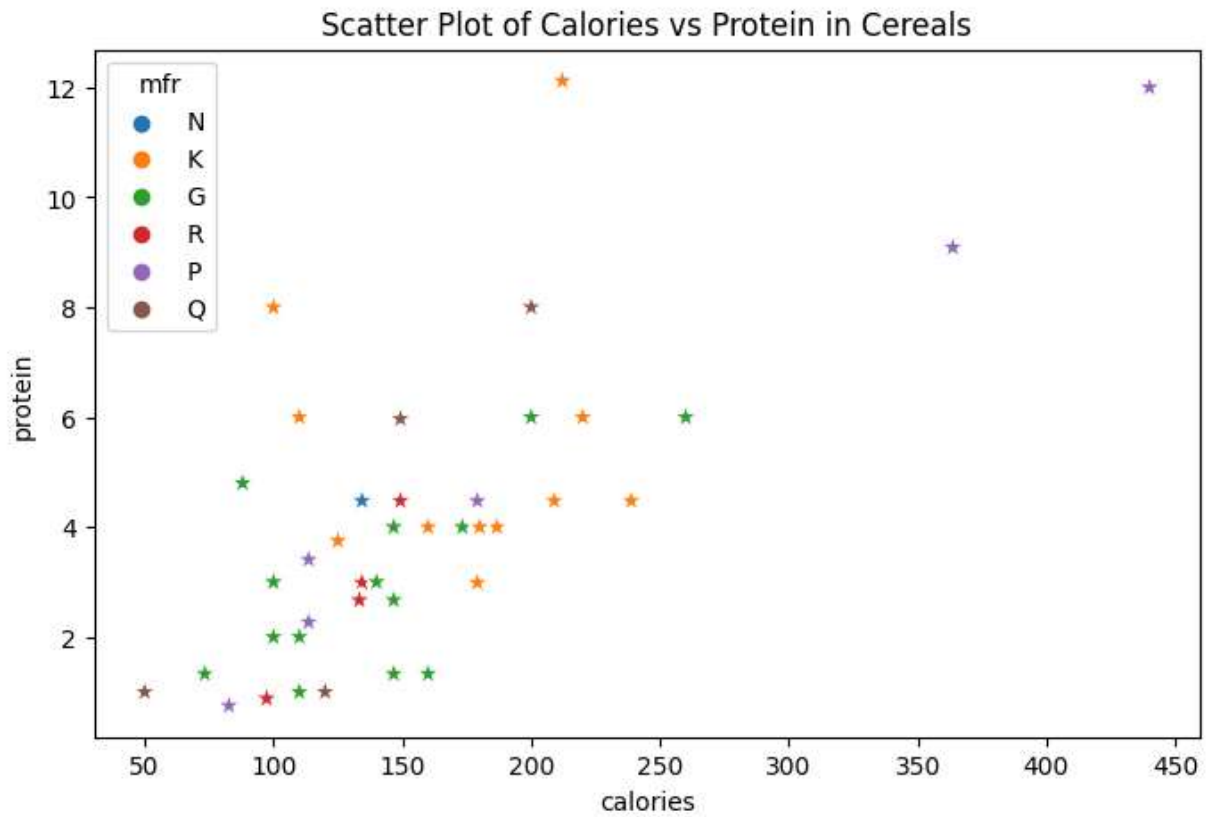
```
sns.barplot(x='mfr', y='sugars', data=cereal_df, ci=None, width=0.5)
```



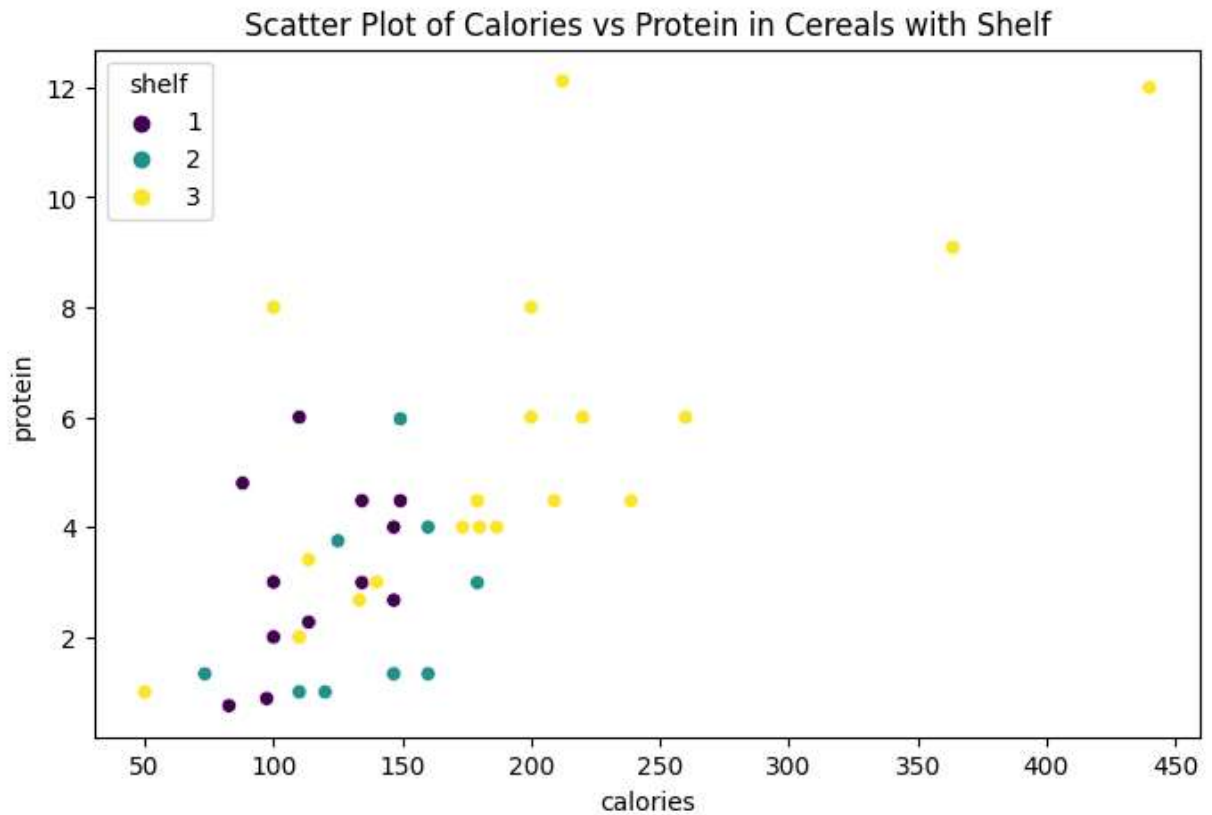
```
In [ ]: # Bar Chart with 3 variables
plt.figure(figsize=(8, 6))
sns.barplot(x='mfr', y='calories', hue='shelf', data=cereal_df)
plt.title('Bar Chart of Calories for Cereal Manufacturers with Shelf')
plt.show()
```



```
In [ ]: # Scatter Plot with 2 variables
plt.figure(figsize=(8, 5))
sns.scatterplot(x='calories', y='protein', data=cereal_df, hue='mfr', marker='*', s=
plt.title('Scatter Plot of Calories vs Protein in Cereals')
plt.show()
```



```
In [ ]: # Scatter Plot with 3 variables and custom colors using 'shelf' column
plt.figure(figsize=(8,5))
sns.scatterplot(x='calories', y='protein', hue='shelf', data=cereal_df, palette='vi
plt.title('Scatter Plot of Calories vs Protein in Cereals with Shelf')
plt.show()
```

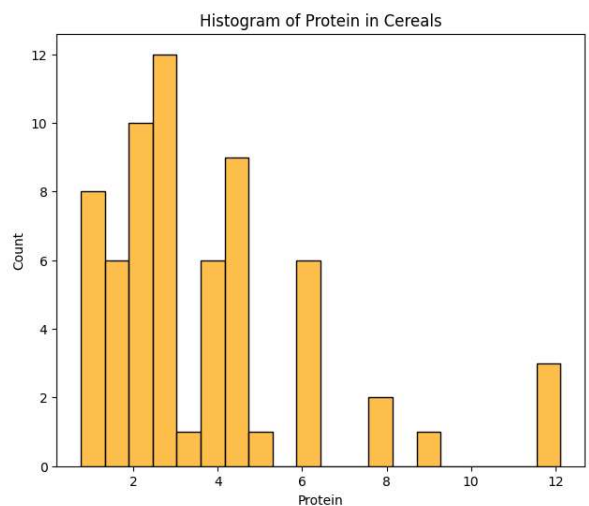
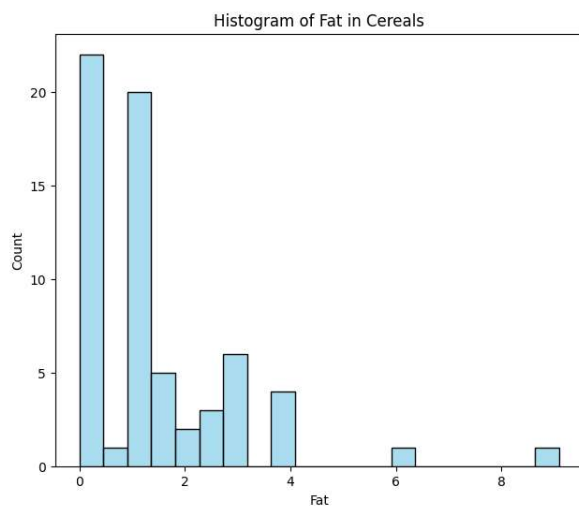


```
In [ ]: # Separate Histograms with 2 variables using 'fat' and 'protein' columns
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

sns.histplot(data=cereal_df, x='fat', bins=20, color='skyblue', alpha=0.7, ax=axes[0])
axes[0].set_title('Histogram of Fat in Cereals')
axes[0].set_xlabel('Fat')

sns.histplot(data=cereal_df, x='protein', bins=20, color='orange', alpha=0.7, ax=axes[1])
axes[1].set_title('Histogram of Protein in Cereals')
axes[1].set_xlabel('Protein')

plt.show()
```



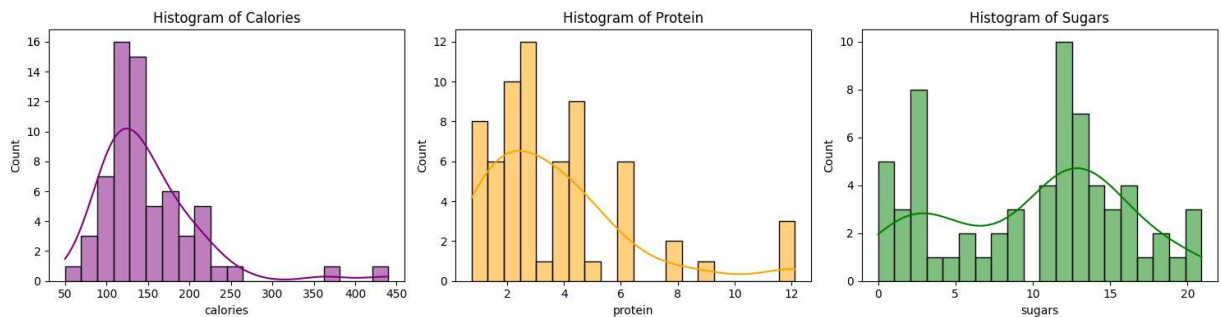
```
In [ ]: # Histograms for 'calories', 'protein', and 'sugars'
plt.figure(figsize=(15, 4))

plt.subplot(1, 3, 1)
sns.histplot(cereal_df['calories'], bins=20, kde=True, color='purple')
plt.title('Histogram of Calories')

plt.subplot(1, 3, 2)
sns.histplot(cereal_df['protein'], bins=20, kde=True, color='orange')
plt.title('Histogram of Protein')

plt.subplot(1, 3, 3)
sns.histplot(cereal_df['sugars'], bins=20, kde=True, color='green')
plt.title('Histogram of Sugars')

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

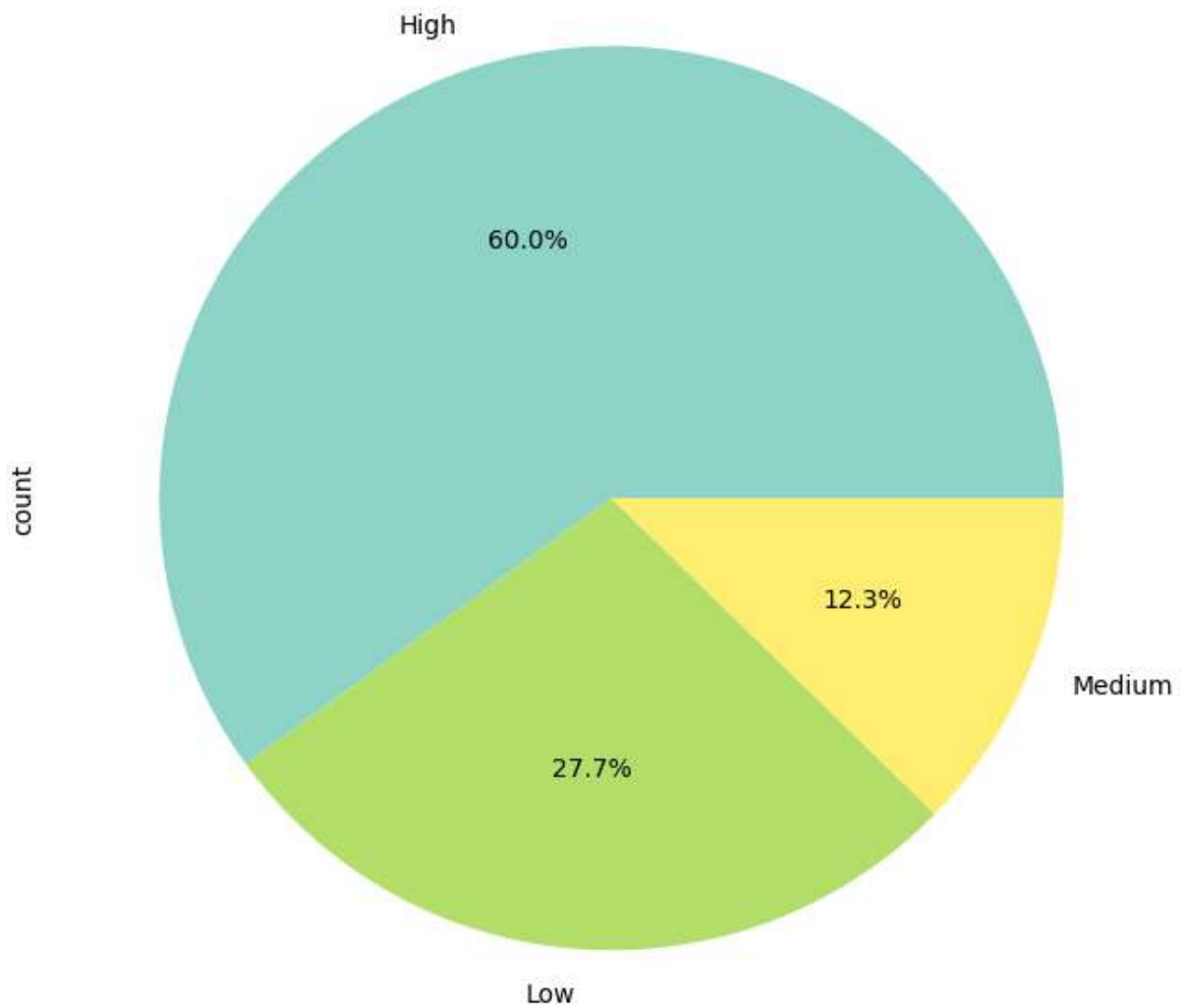


```
In [ ]: # Grouping 'sugars' into categories (e.g., low, medium, high)

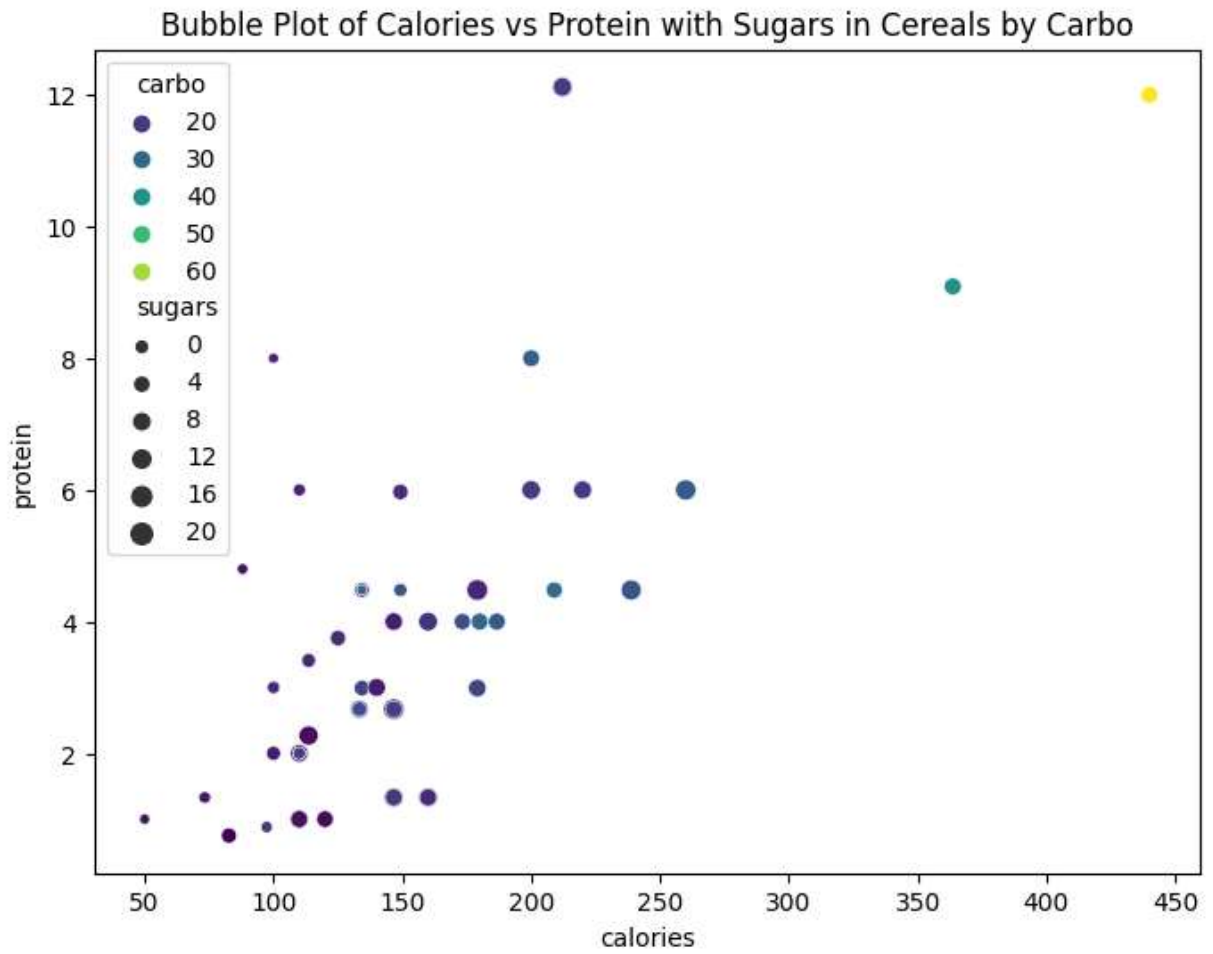
bins = [0, 5, 10, np.inf]
labels = ['Low', 'Medium', 'High']
cereal_df['sugar_category'] = pd.cut(cereal_df['sugars'], bins=bins, labels=labels,

# Pie chart using the sugar categories
plt.figure(figsize=(8, 8))
cereal_df['sugar_category'].value_counts().plot.pie(autopct='%1.1f%%', labels=cerea
plt.title('Pie Chart of Cereal Sugars Distribution')
plt.show()
```

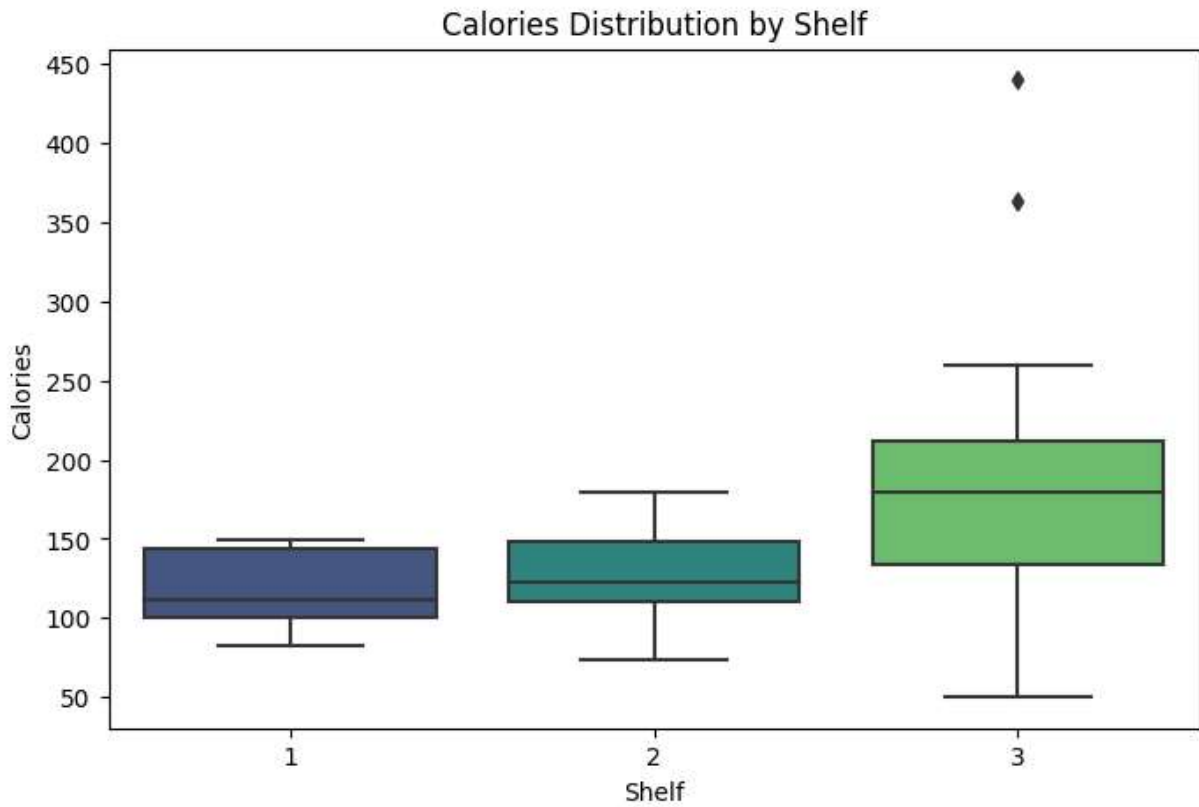
Pie Chart of Cereal Sugars Distribution



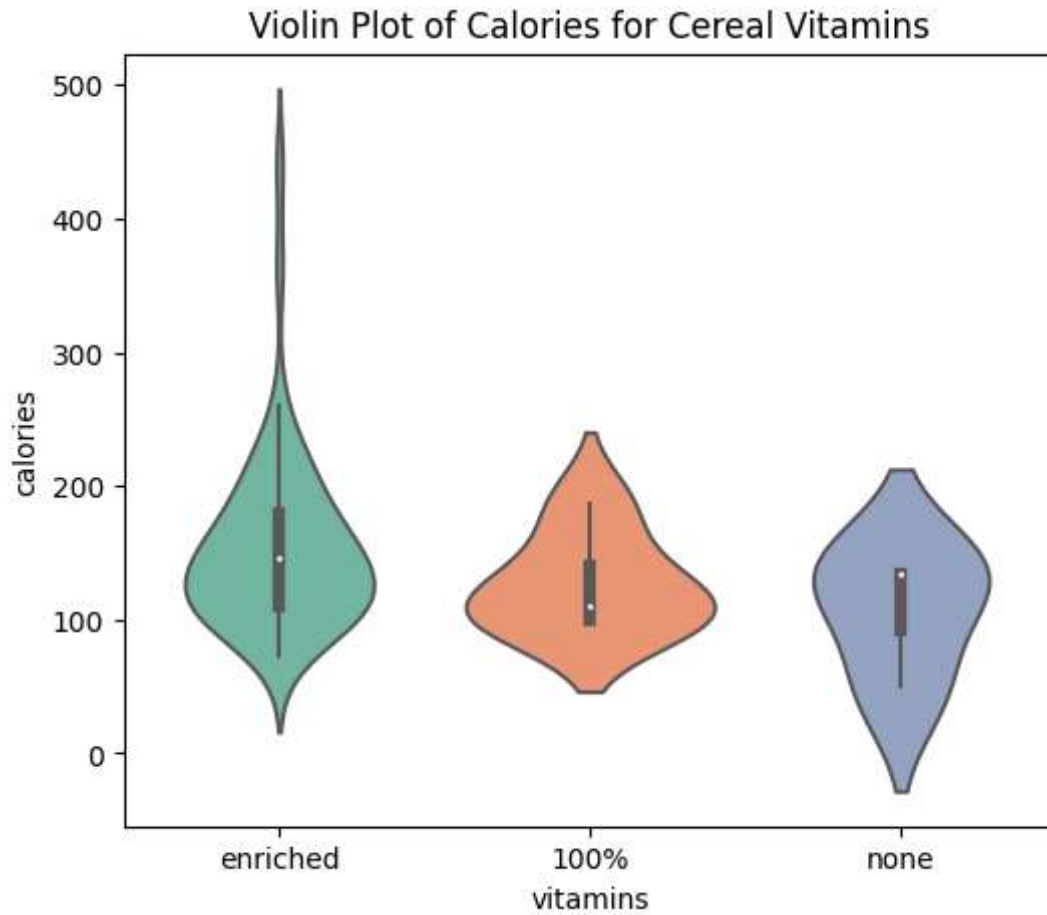
```
In [ ]: # Bubble Plot using 'carbo' column with different colors
plt.figure(figsize=(8, 6))
sns.scatterplot(x='calories', y='protein', size='sugars', data=cereal_df, hue='carb
plt.title('Bubble Plot of Calories vs Protein with Sugars in Cereals by Carbo')
plt.show()
```



```
In [ ]: # Assuming 'shelf' and 'calories' are columns in your DataFrame
plt.figure(figsize=(8, 5))
sns.boxplot(x='shelf', y='calories', data=cereal_df, palette='viridis')
plt.title('Calories Distribution by Shelf')
plt.xlabel('Shelf')
plt.ylabel('Calories')
plt.show()
```

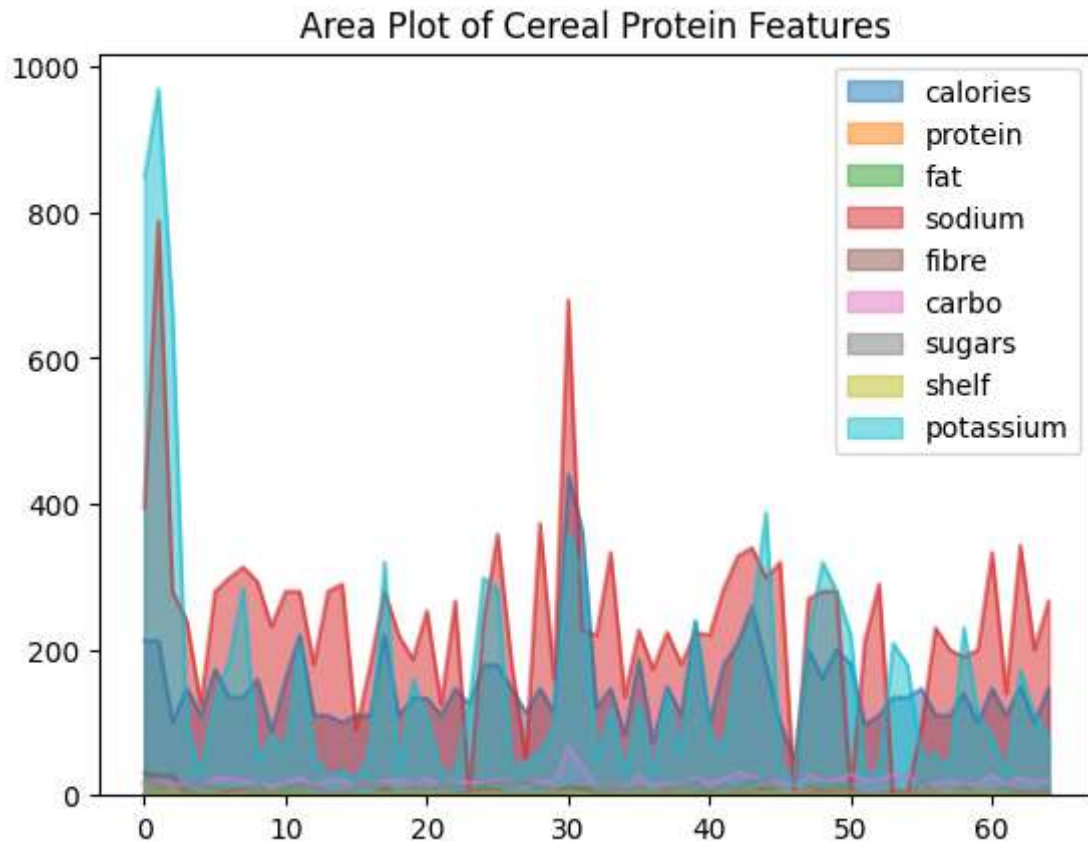



```
In [ ]: # Violin Plot using 'vitamins' column
plt.figure(figsize=(6,5))
sns.violinplot(x='vitamins', y='calories', data=cereal_df, palette='Set2')
plt.title('Violin Plot of Calories for Cereal Vitamins')
plt.show()
```

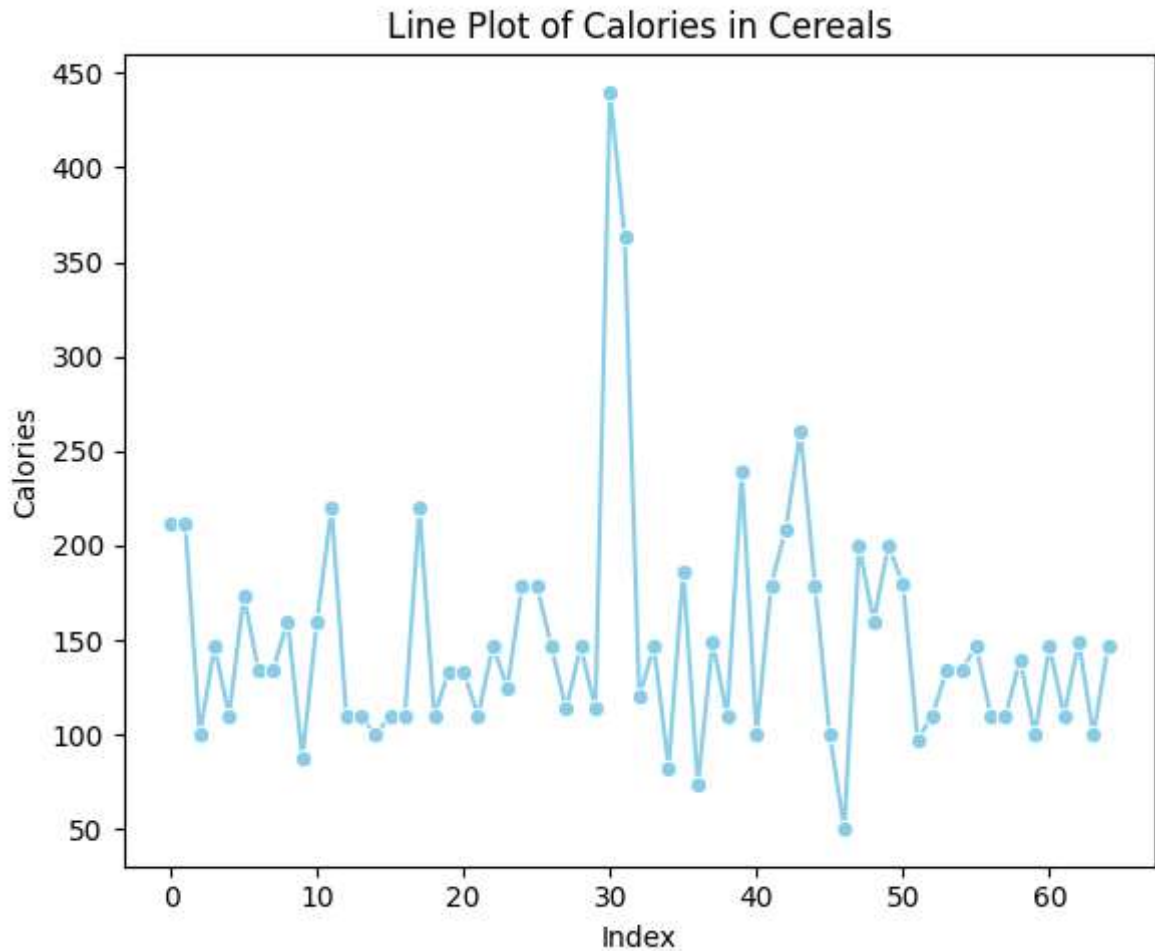


```
In [ ]: # Area Plot using 'protein' column
plt.figure(figsize=(8, 5)) # Adjust the figure size here
cereal_df.drop('vitamins', axis=1).plot.area(stacked=False, colormap='tab10')
plt.title('Area Plot of Cereal Protein Features')
plt.show()
```

<Figure size 800x500 with 0 Axes>



```
In [ ]: # Line plot using the index and 'calories' column
plt.figure(figsize=(6, 5))
sns.lineplot(x=cereal_df.index, y='calories', data=cereal_df, marker='o', color='sk
plt.title('Line Plot of Calories in Cereals')
plt.xlabel('Index')
plt.ylabel('Calories')
plt.tight_layout()
plt.show()
```



```
In [ ]: # Assuming 'calories', 'protein', 'fat', 'sodium', 'fibre', 'carbo', 'sugars', 'pot
numeric_columns = ['calories', 'protein', 'fat', 'sodium', 'fibre', 'carbo', 'sugar

# Create a smaller pair plot
sns.pairplot(cereal_df[numeric_columns], height=1.5)
plt.suptitle('Pair Plot of Numerical Variables', y=1.02)
plt.show()
```

```
C:\Users\Kalpana\AppData\Roaming\Python\Python311\site-packages\seaborn\axisgrid.py:
118: UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)
```

