80-cereals

March 2, 2024

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

[2]:
```
df = pd.read_csv("cereals.csv")
```

[3]:
```
df.shape
```

[3]:
```
(77, 16)
```

[4]:
```
df.describe
```

[4]:
```
<bound method NDFrame.describe 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  68.402973
1     8.0       8     135         0      3     1.0  1.00  33.983679
2     7.0       5     320        25      3     1.0  0.33  59.425505
3     8.0       0     330        25      3     1.0  0.50  93.704912
4    14.0       8      -1        25      3     1.0  0.75  34.384843
..      …       …       …         …      …       …     …          …
72   21.0       3      60        25      3     1.0  0.75  39.106174
73   13.0      12      25        25      2     1.0  1.00  27.753301
74   17.0       3     115        25      1     1.0  0.67  49.787445
75   17.0       3     110        25      1     1.0  1.00  51.592193
```

```
76   16.0       8      60         25       1      1.0  0.75  36.187559

[77 rows x 16 columns]>
```

[5]:

```
df.describe()
```

[5]:

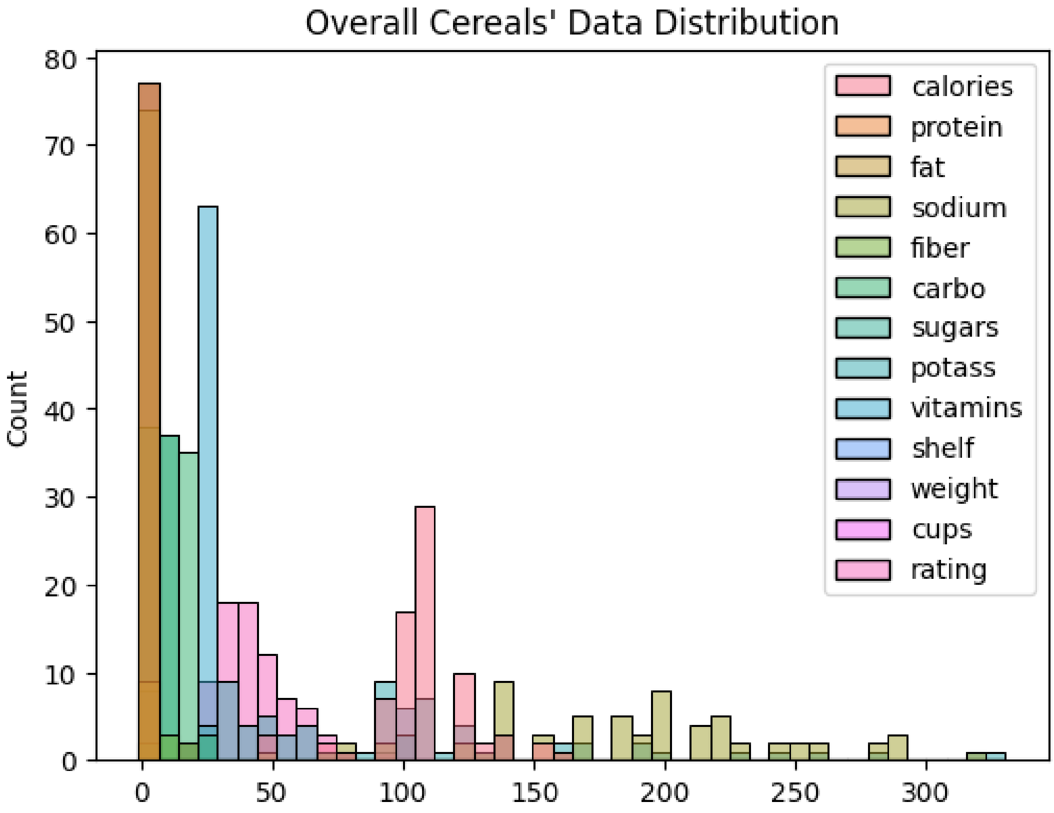
```
         calories    protein        fat      sodium      fiber      carbo  \
count   77.000000  77.000000  77.000000   77.000000  77.000000  77.000000   
mean   106.883117   2.545455   1.012987  159.675325   2.151948  14.597403   
std     19.484119   1.094790   1.006473   83.832295   2.383364   4.278956   
min     50.000000   1.000000   0.000000    0.000000   0.000000  -1.000000   
25%    100.000000   2.000000   0.000000  130.000000   1.000000  12.000000   
50%    110.000000   3.000000   1.000000  180.000000   2.000000  14.000000   
75%    110.000000   3.000000   2.000000  210.000000   3.000000  17.000000   
max    160.000000   6.000000   5.000000  320.000000  14.000000  23.000000   

          sugars      potass    vitamins      shelf     weight       cups  \
count  77.000000   77.000000   77.000000  77.000000  77.000000  77.000000   
mean    6.922078   96.077922   28.246753   2.207792   1.029610   0.821039   
std     4.444885   71.286813   22.342523   0.832524   0.150477   0.232716   
min    -1.000000   -1.000000    0.000000   1.000000   0.500000   0.250000   
25%     3.000000   40.000000   25.000000   1.000000   1.000000   0.670000   
50%     7.000000   90.000000   25.000000   2.000000   1.000000   0.750000   
75%    11.000000  120.000000   25.000000   3.000000   1.000000   1.000000   
max    15.000000  330.000000  100.000000   3.000000   1.500000   1.500000   

          rating  
count  77.000000  
mean   42.665705  
std    14.047289  
min    18.042851  
25%    33.174094  
50%    40.400208  
75%    50.828392  
max    93.704912  
```

[6]:

```
plt.title("Overall Cereals' Data Distribution")
sns.histplot(data = df)
```

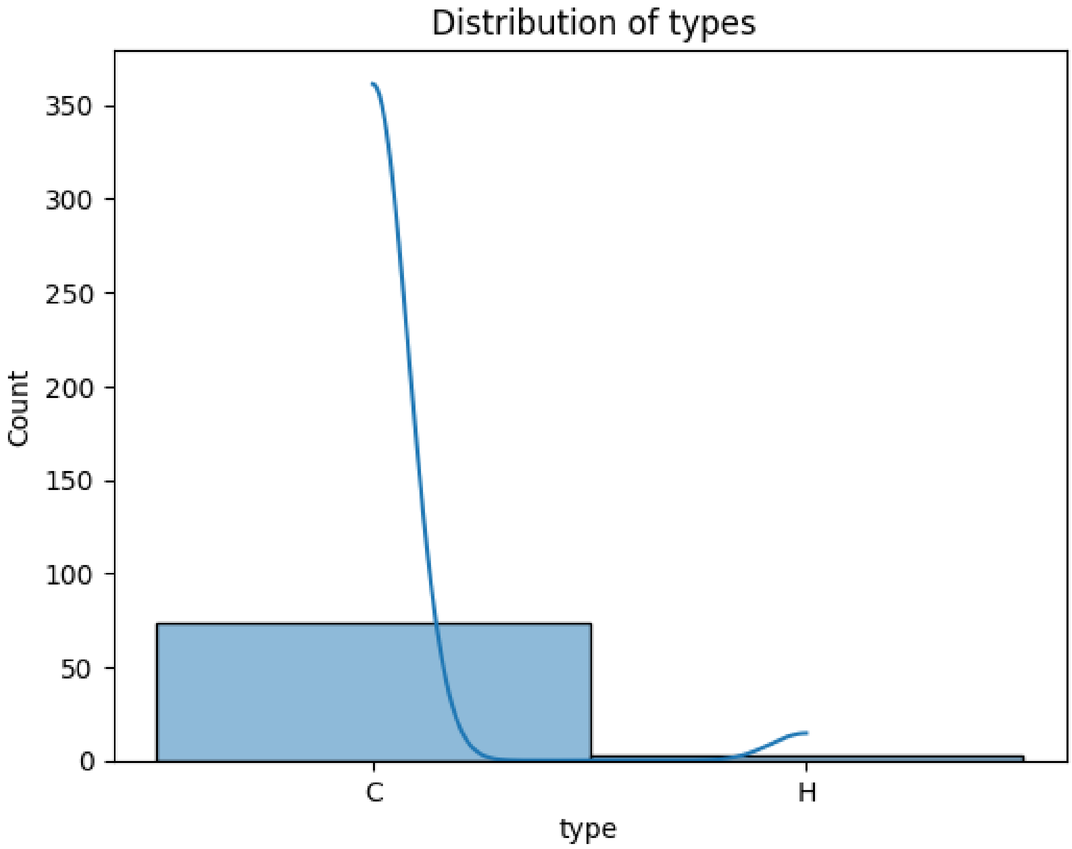
[6]:

```
<Axes: title={'center': "Overall Cereals' Data Distribution"}, ylabel='Count'>
```



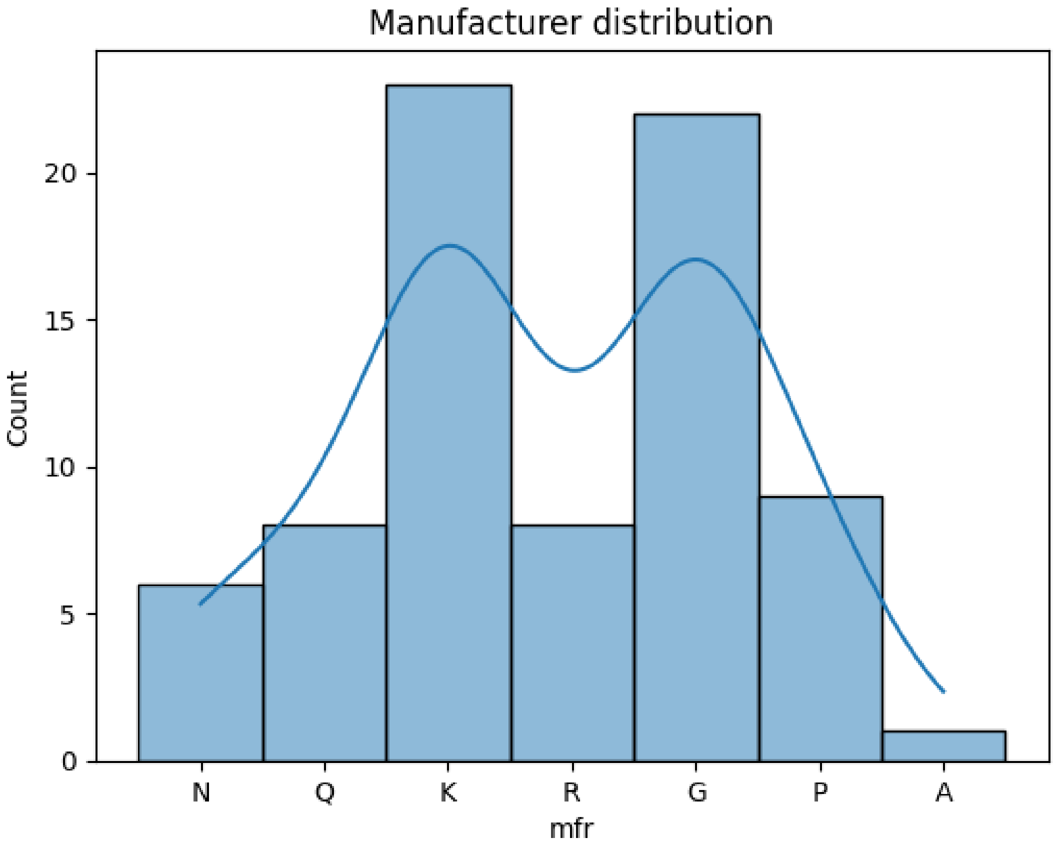
```
[7]: sns.histplot(x='type', data=df,kde=True)
     plt.title(' Distribution of types')
```

```
[7]: Text(0.5, 1.0, ' Distribution of types')
```



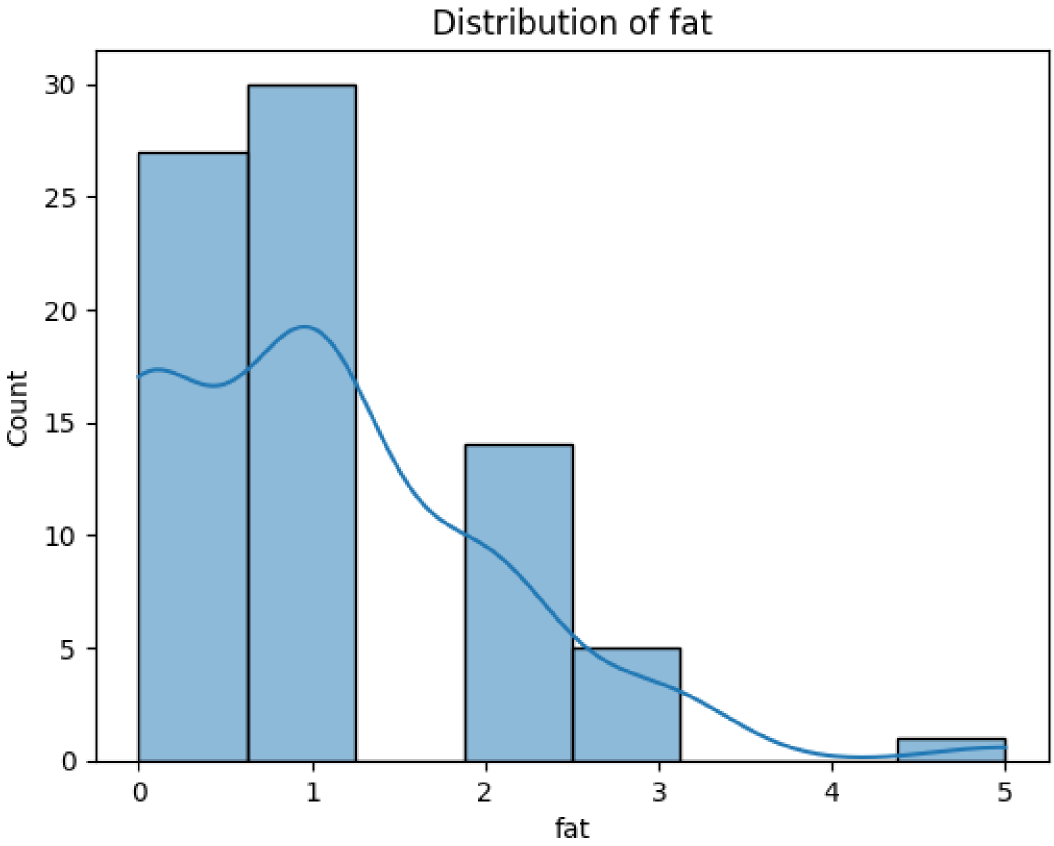
```
[8]: sns.histplot(data=df,x='mfr', kde=True)
     plt.title('Manufacturer distribution')
```

```
[8]: Text(0.5, 1.0, 'Manufacturer distribution')
```



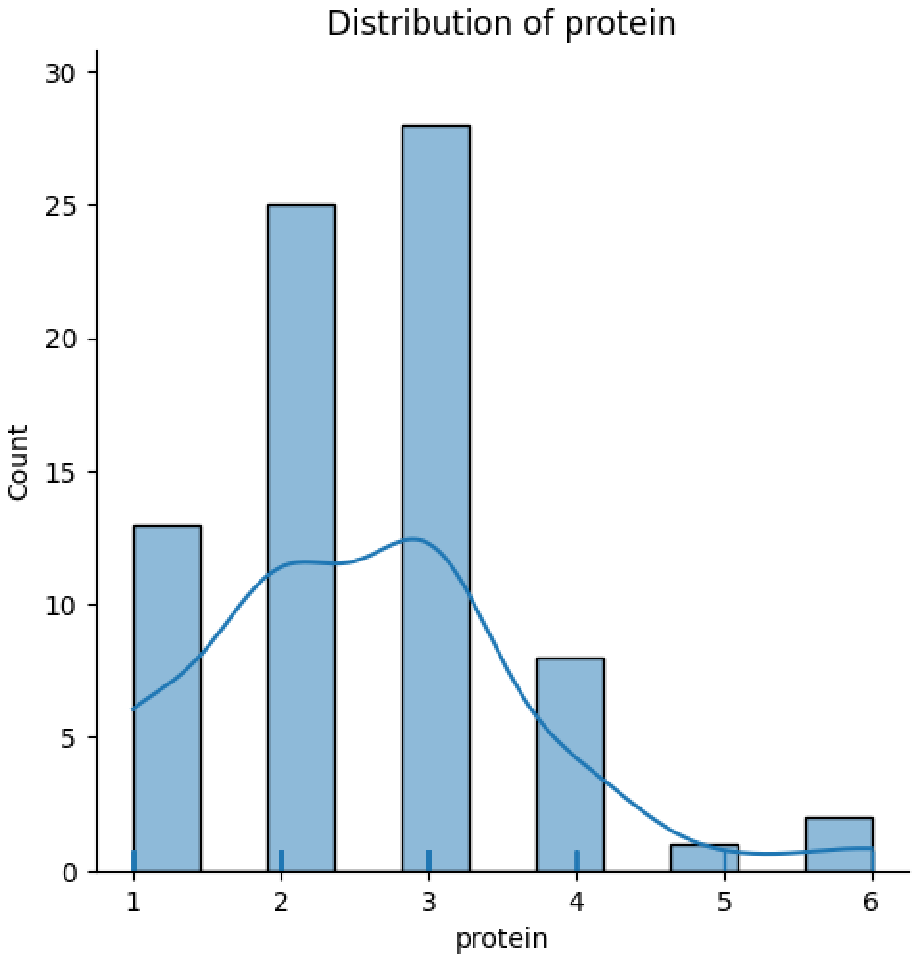
[9]:
```
sns.histplot(df['fat'],kde=True)
plt.title('Distribution of fat')
```

[9]:
```
Text(0.5, 1.0, 'Distribution of fat')
```



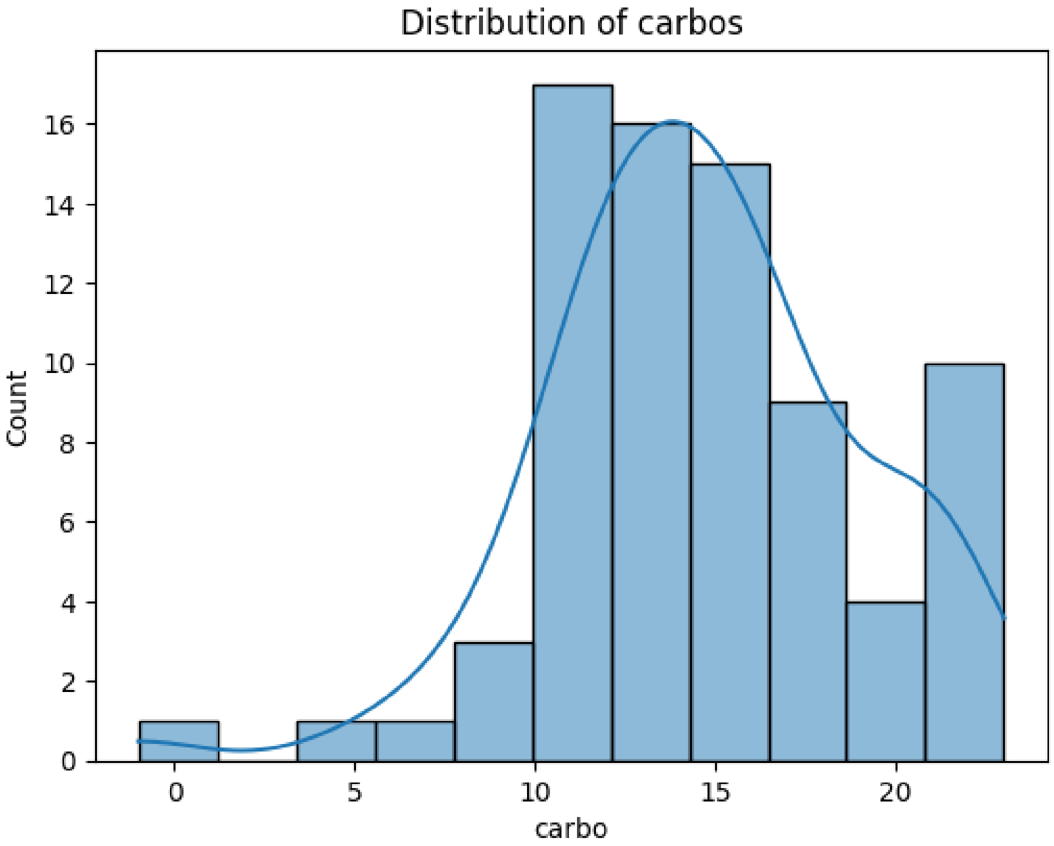
```
[10]: sns.displot(df['protein'],rug=True,kde=True)
      plt.title('Distribution of protein')
```

```
[10]: Text(0.5, 1.0, 'Distribution of protein')
```



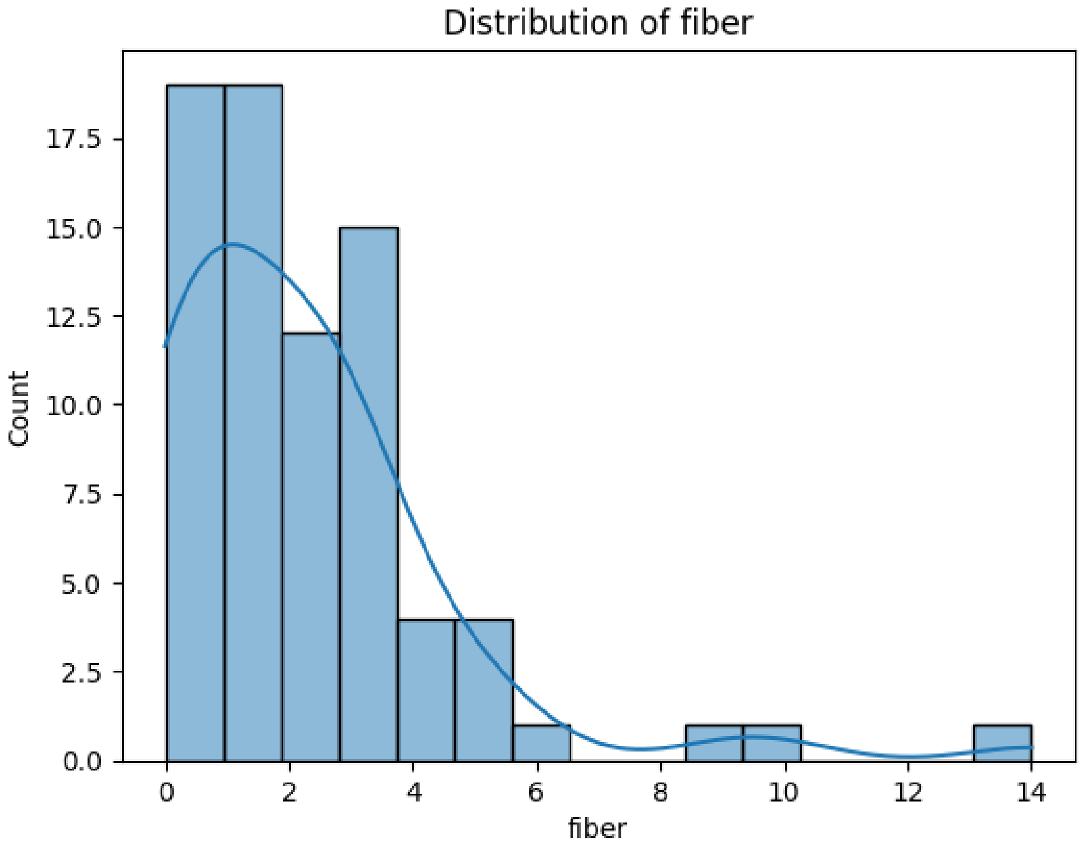
```
[11]: sns.histplot(df['carbo'],kde=True)
      plt.title('Distribution of carbos')
```

```
[11]: Text(0.5, 1.0, 'Distribution of carbos')
```



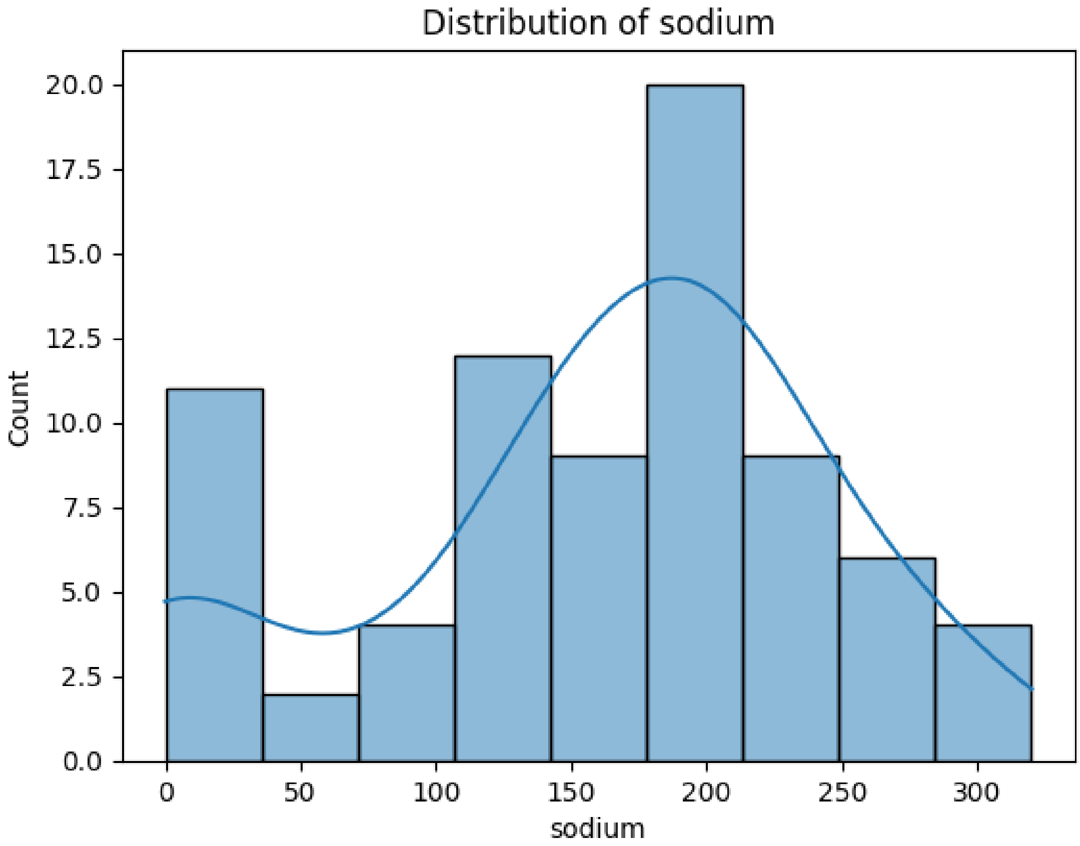
[12]:
```
sns.histplot(df['fiber'],kde=True)
plt.title('Distribution of fiber')
```

[12]:
```
Text(0.5, 1.0, 'Distribution of fiber')
```



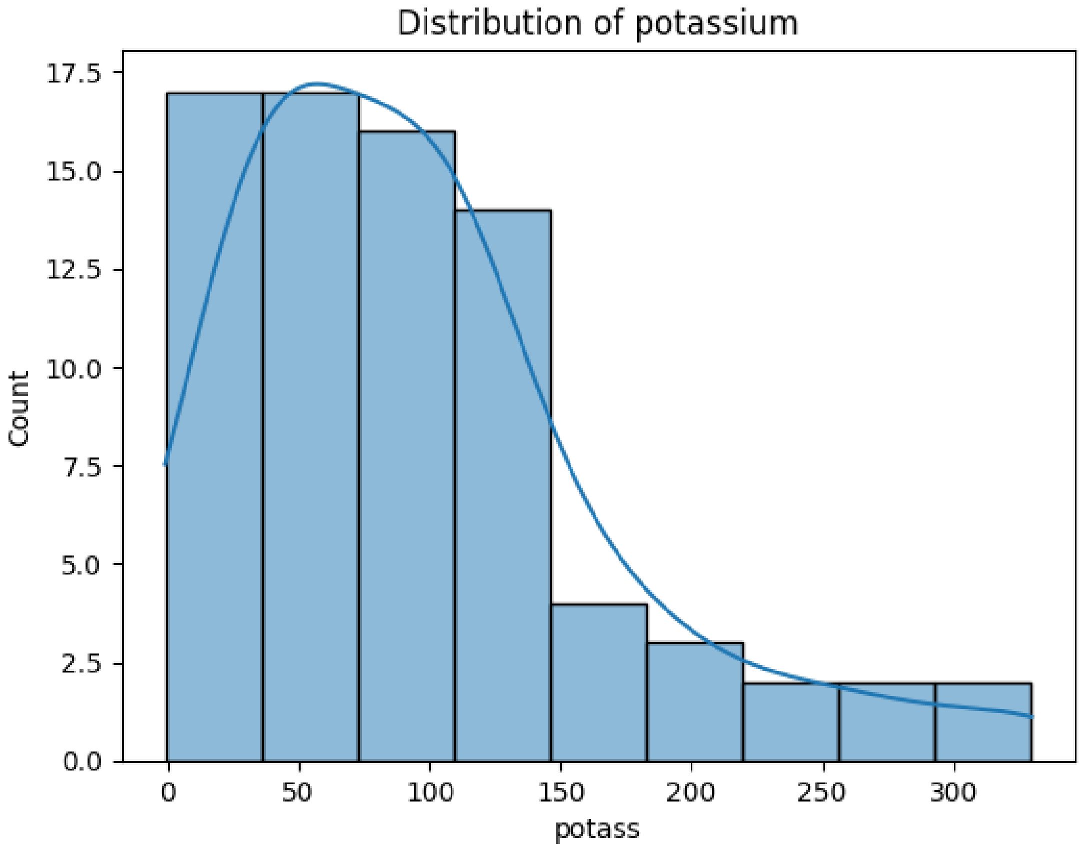
```
[13]: sns.histplot(df['sodium'],kde=True)
      plt.title('Distribution of sodium')
```

```
[13]: Text(0.5, 1.0, 'Distribution of sodium')
```



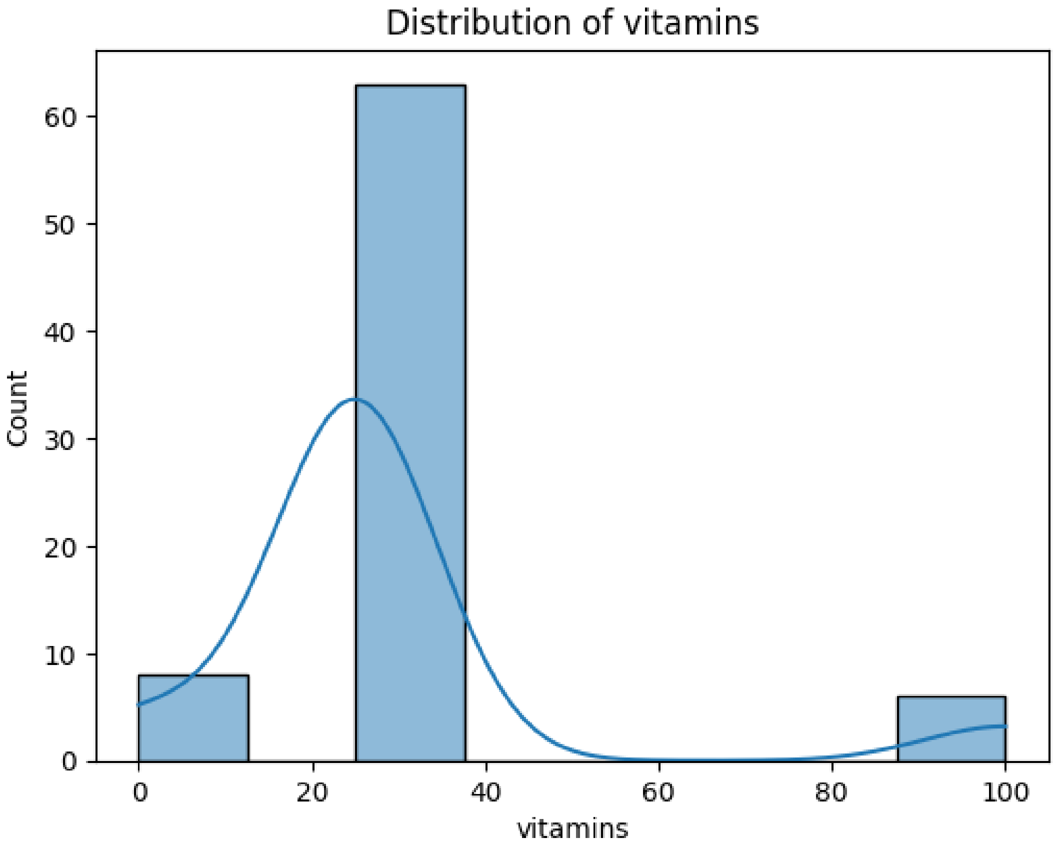
```
[14]: sns.histplot(df['potass'],kde=True)
      plt.title('Distribution of potassium')
```

```
[14]: Text(0.5, 1.0, 'Distribution of potassium')
```



```
[15]: sns.histplot(df['vitamins'],kde=True)
      plt.title('Distribution of vitamins')
```

```
[15]: Text(0.5, 1.0, 'Distribution of vitamins')
```

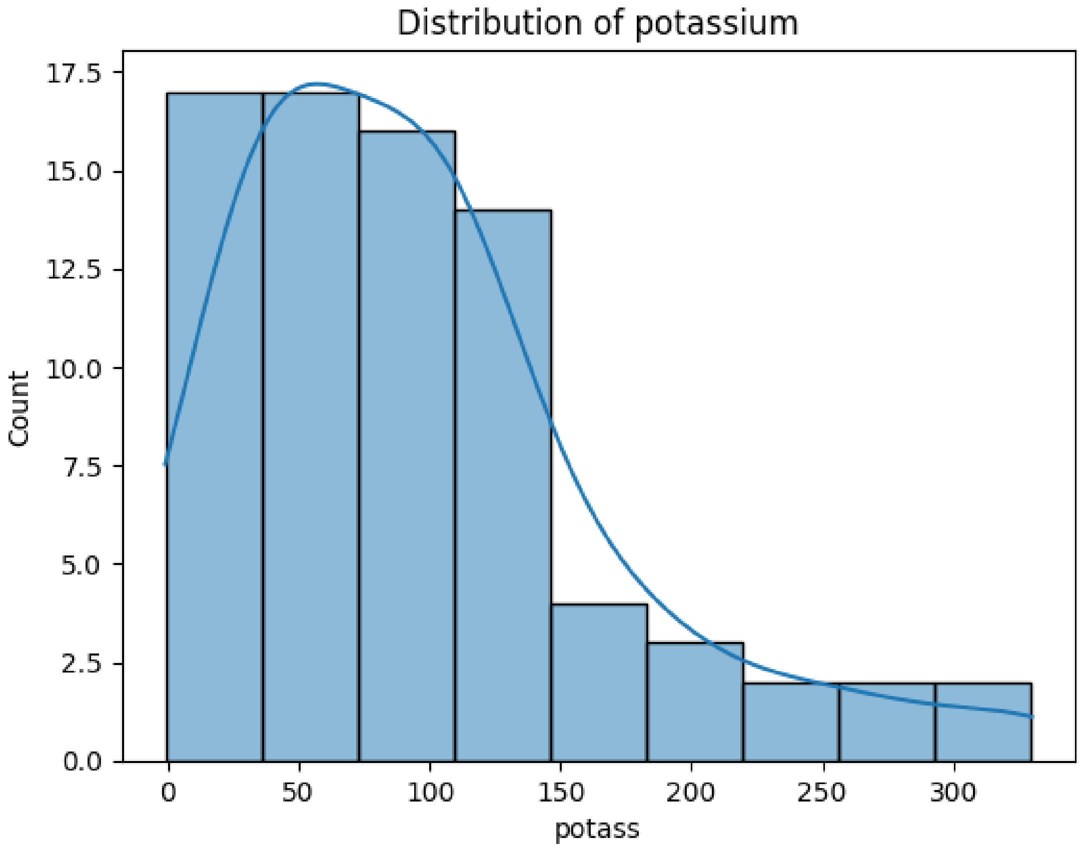


[16]:

```
sns.histplot(df['potass'],kde=True)
plt.title('Distribution of potassium')
```

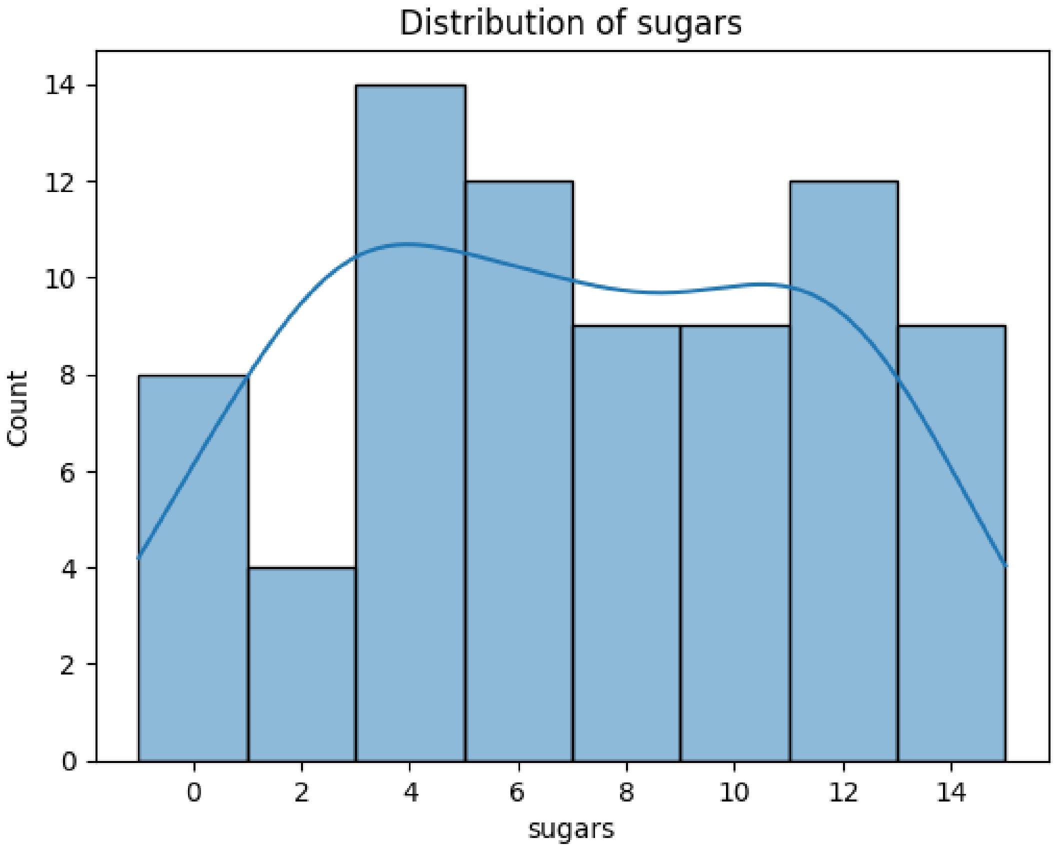
[16]:

```
Text(0.5, 1.0, 'Distribution of potassium')
```



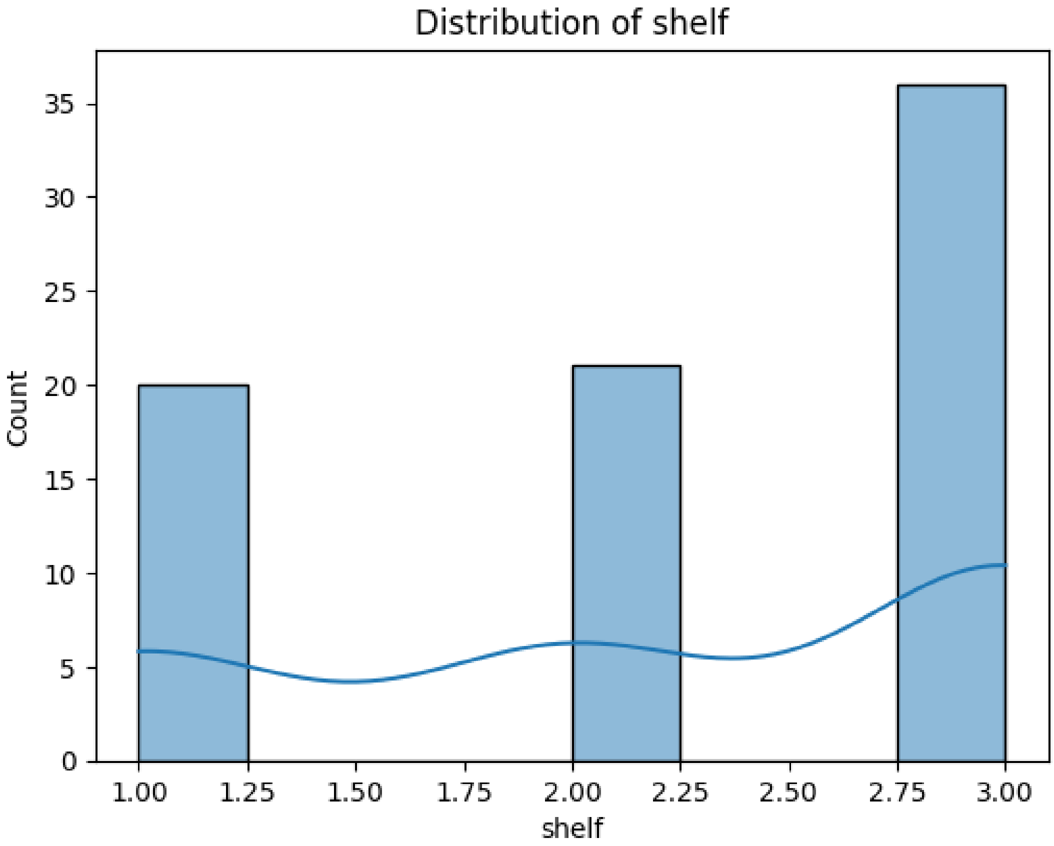
[17]:
```
sns.histplot(df['sugars'],kde=True)
plt.title('Distribution of sugars')
```

[17]:
```
Text(0.5, 1.0, 'Distribution of sugars')
```



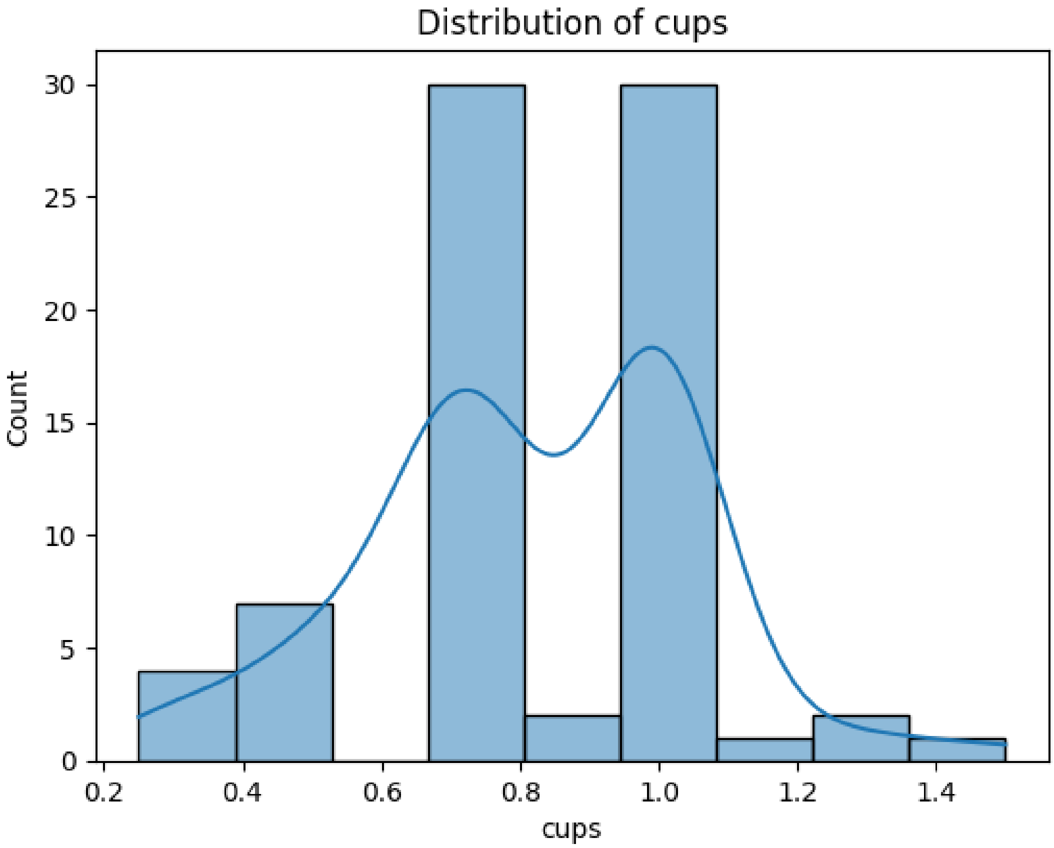
```
[18]: sns.histplot(df['shelf'],kde=True)
      plt.title('Distribution of shelf')
```

```
[18]: Text(0.5, 1.0, 'Distribution of shelf')
```



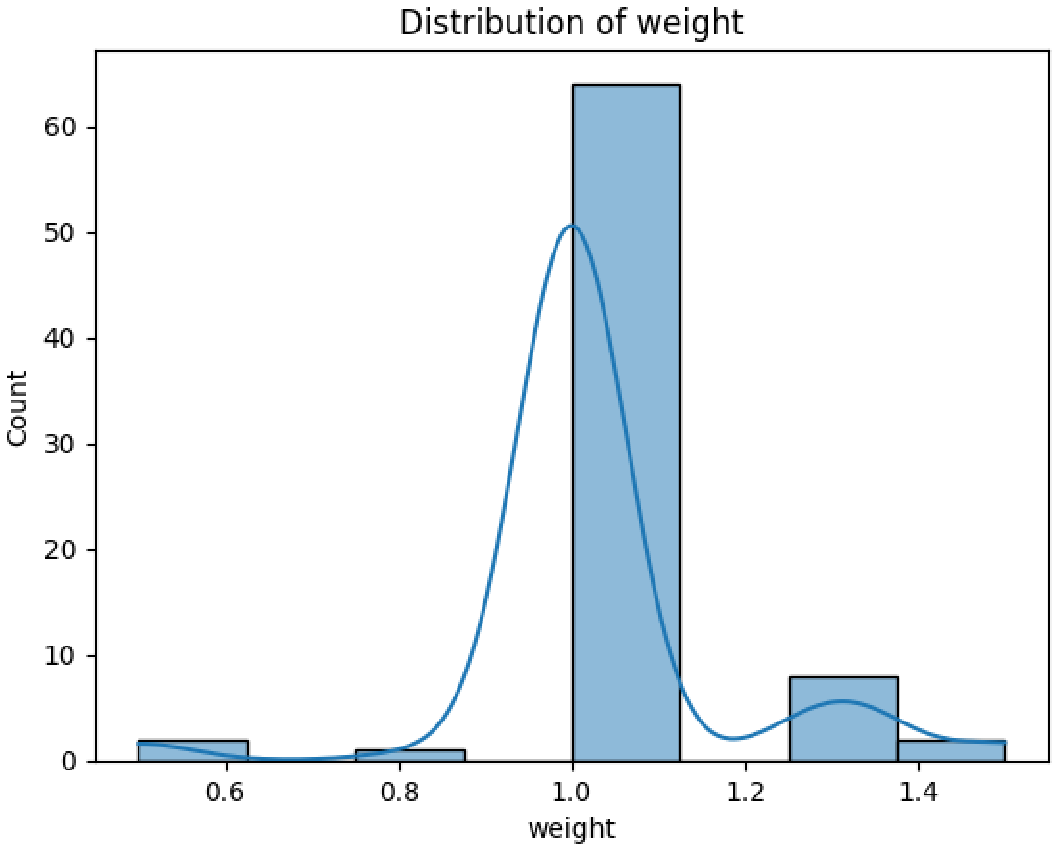
```
[19]: sns.histplot(df['cups'],kde=True)
      plt.title('Distribution of cups')
```

```
[19]: Text(0.5, 1.0, 'Distribution of cups')
```



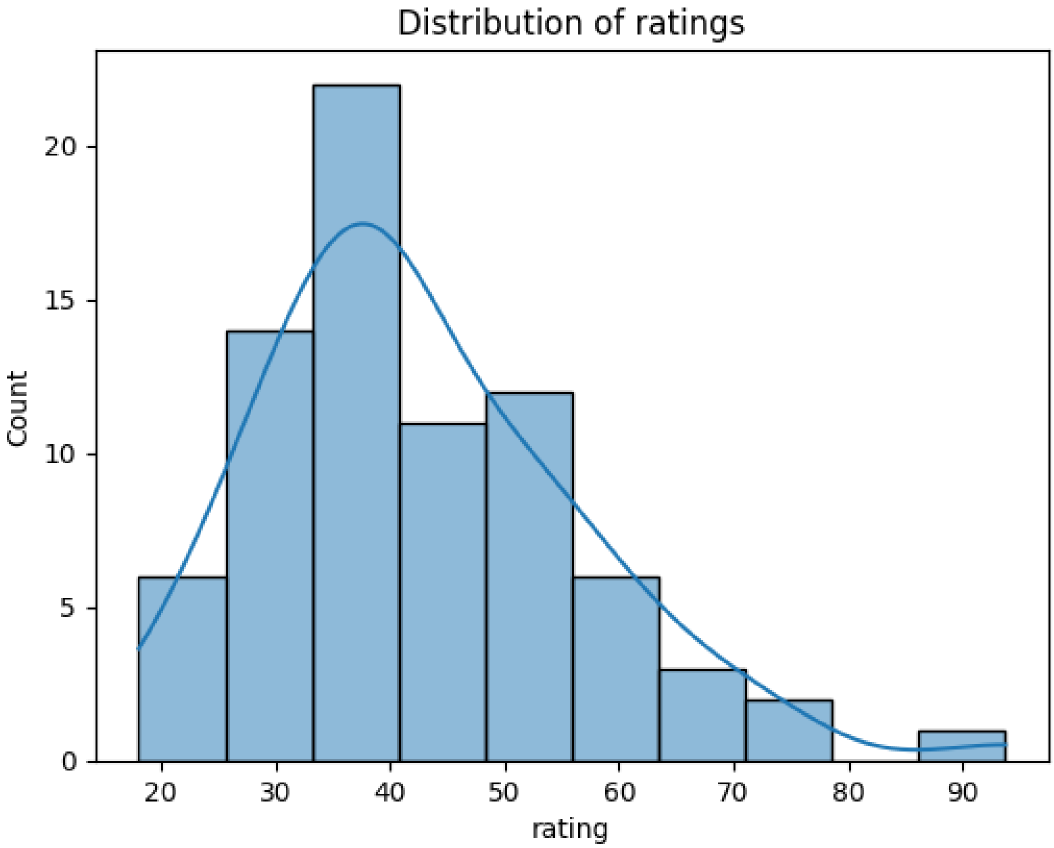
```
[20]: sns.histplot(df['weight'],kde=True)
      plt.title('Distribution of weight')
```

```
[20]: Text(0.5, 1.0, 'Distribution of weight')
```



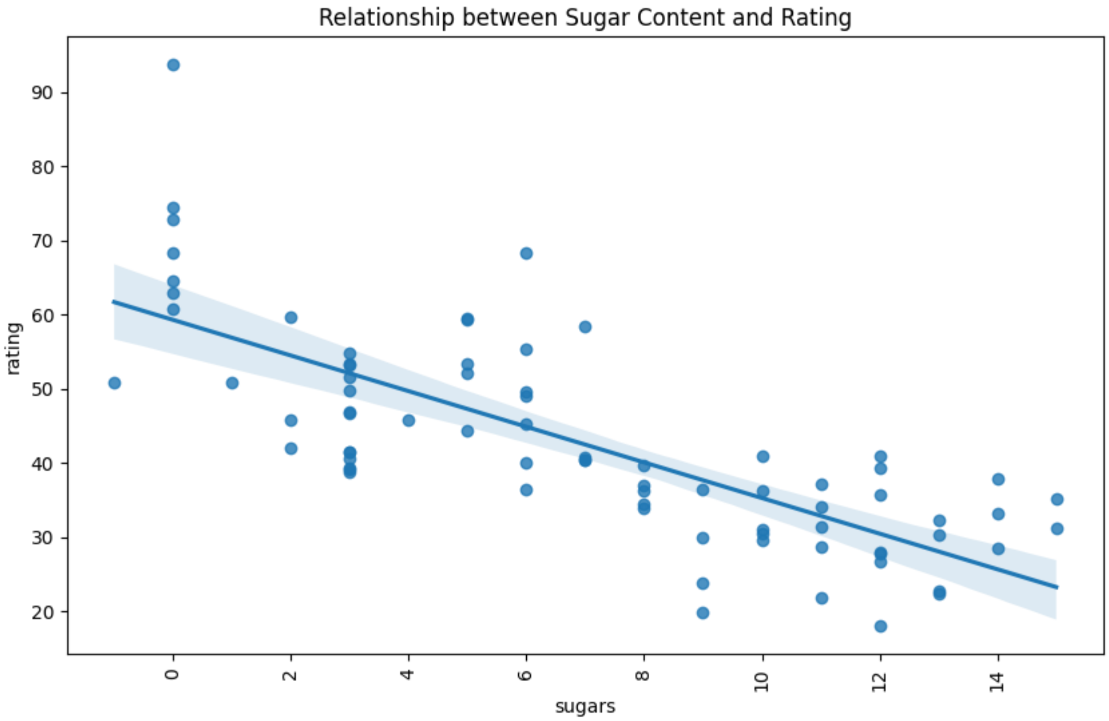
```
[21]: sns.histplot(df['rating'],kde=True)
      plt.title('Distribution of ratings')
```

```
[21]: Text(0.5, 1.0, 'Distribution of ratings')
```



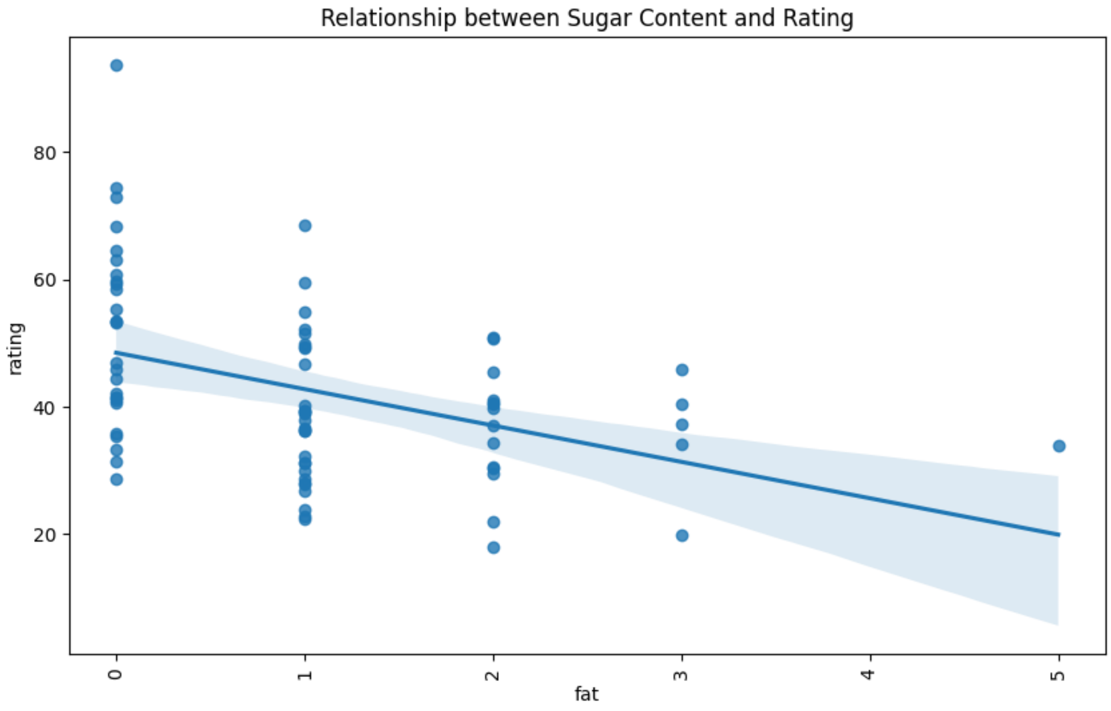
```
[22]: plt.figure(figsize=(10, 6))
      plt.title('Relationship between Sugar Content and Rating')
      plt.xticks(rotation=90)
      sns.regplot(data=df, x=df['sugars'], y=df['rating'])
```

```
[22]: <Axes: title={'center': 'Relationship between Sugar Content and Rating'},
      xlabel='sugars', ylabel='rating'>
```



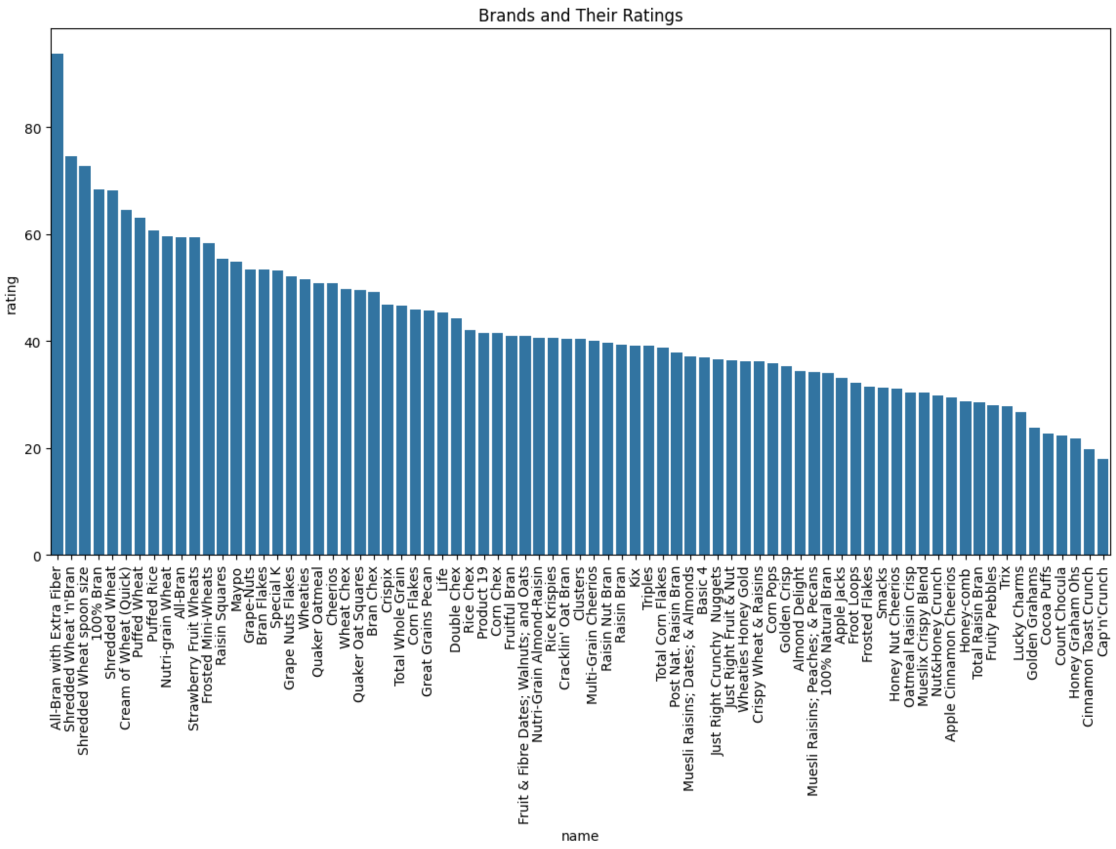
```
[23]: plt.figure(figsize=(10, 6))
      plt.title('Relationship between Sugar Content and Rating')
      plt.xticks(rotation=90)
      sns.regplot(data=df, x=df['fat'], y=df['rating'])
```

```
[23]: <Axes: title={'center': 'Relationship between Sugar Content and Rating'},
      xlabel='fat', ylabel='rating'>
```



[24]:
```
 # Sorting the DataFrame by rating in descending order
cereals_sorted = df.sort_values(by='rating', ascending=False)
plt.figure(figsize=(14, 7))
plt.title("Brands and Their Ratings")
plt.xticks(rotation=90)
sns.barplot(data=cereals_sorted, x=cereals_sorted['name'],
y=cereals_sorted['rating'])
```

[24]:
```
<Axes: title={'center': 'Brands and Their Ratings'}, xlabel='name',
ylabel='rating'>
```

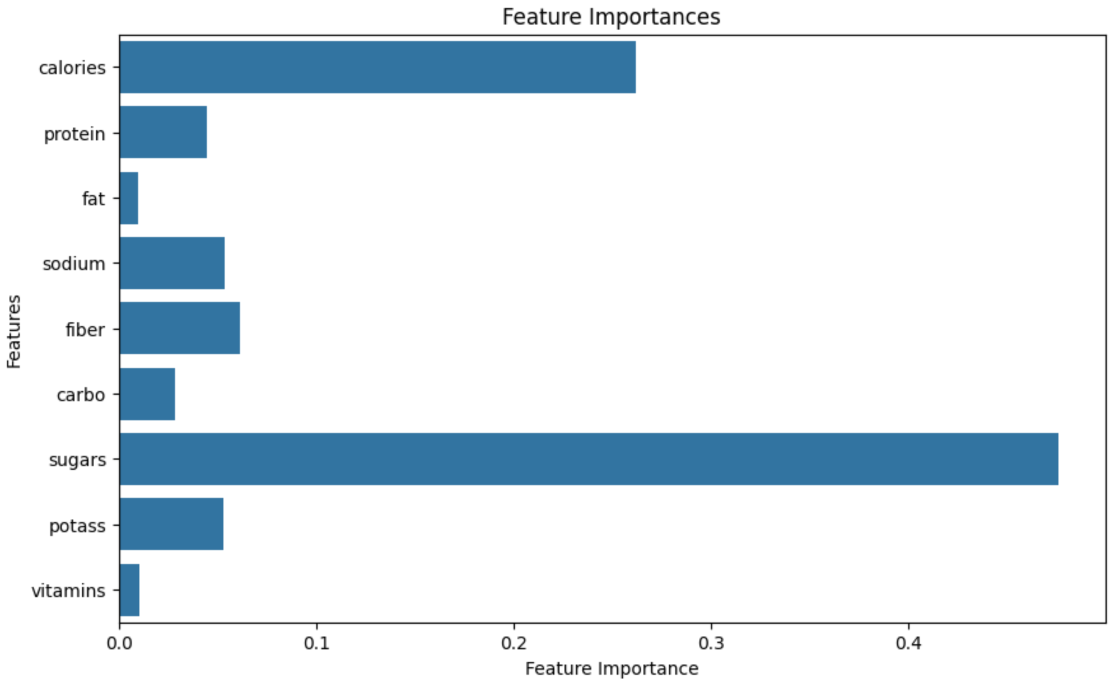


[25]:

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor

# Assuming you have already loaded your DataFrame 'df'
# Drop non-numerical features for X
X = df.drop(columns=['name', 'type', 'mfr', 'rating', 'shelf', 'cups','weight'])
# Assign the target variable y as 'rating'
y = df['rating']
# Fit the RandomForestRegressor model
model = RandomForestRegressor()
model.fit(X, y)
# Extract feature importances
feature_importances = model.feature_importances_
```

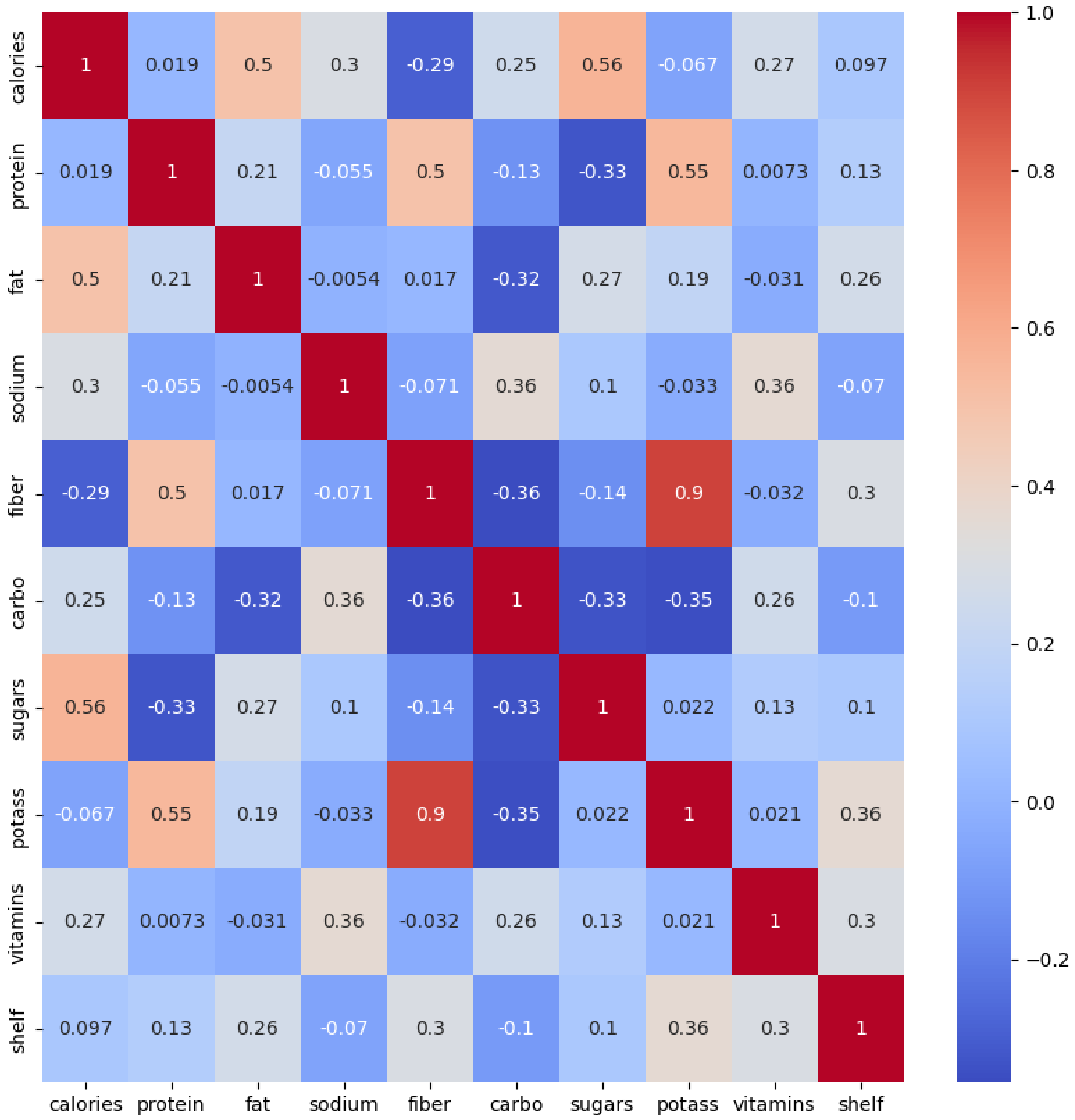
[26]:

```
plt.figure(figsize=(10, 6))
sns.barplot(x=feature_importances, y=X.columns)
plt.xlabel("Feature Importance")
plt.ylabel("Features")
plt.title("Feature Importances")
plt.show()
```



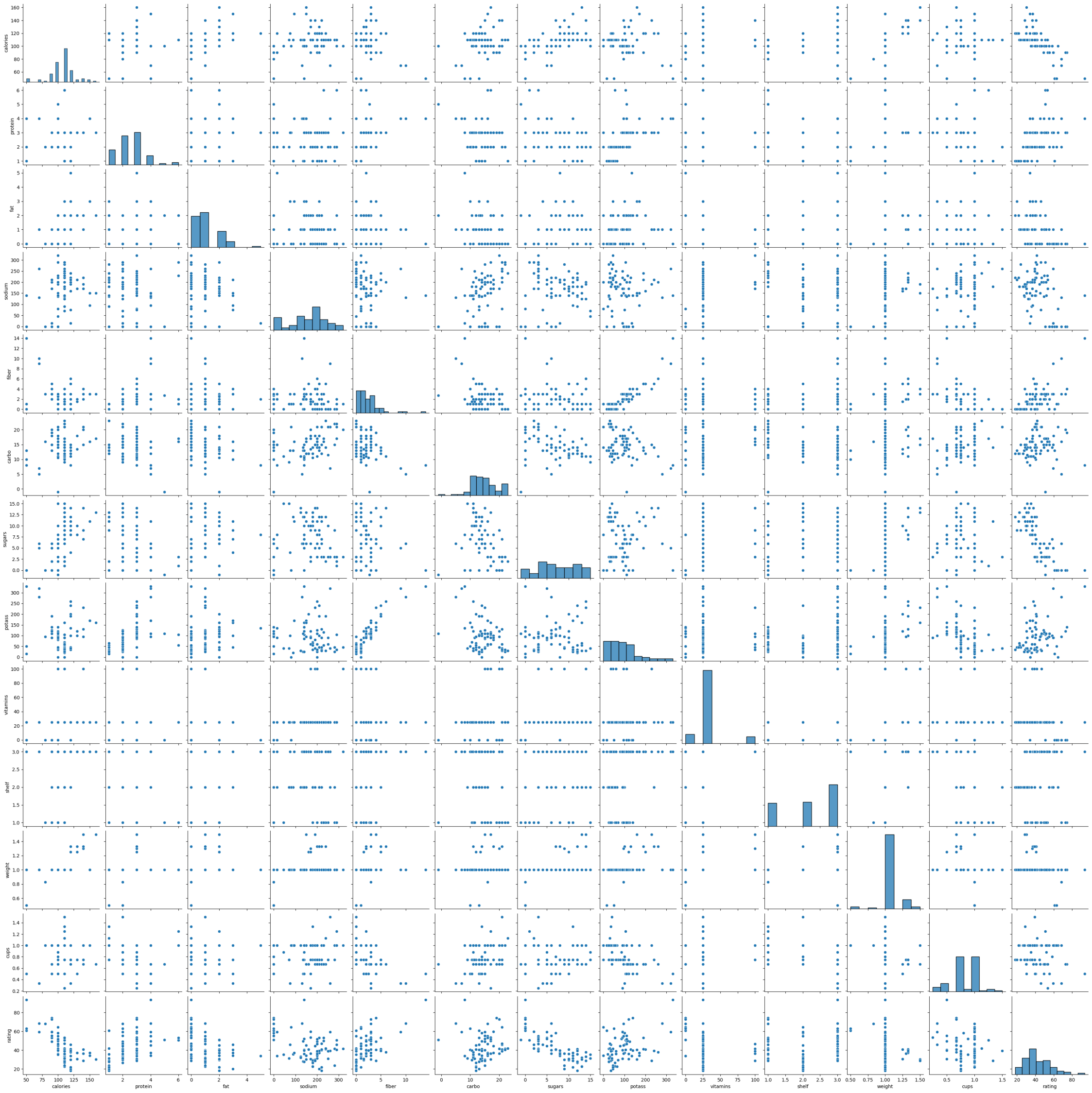
[27]:

```
fig, ax = plt.subplots(figsize=(10, 10))
num=['calories','protein','fat','sodium','fiber','carbo','sugars','potass','vitamins','shelf']
sns.heatmap(df[num].corr(), annot=True, cmap='coolwarm',ax=ax)
plt.show()
```



```
[29]: sns.pairplot(df)
```

```
[29]: <seaborn.axisgrid.PairGrid at 0x1d52b4118b0>
```


calories
protein
fat
sodium
fiber
carbo
sugars
potass
vitamins
shelf
weight
cups
rating
50 75 100 125 150
calories
2 4 6
protein
0 2 4
fat
0 100 200 300
sodium
0 5 10
fiber
0 10 20
carbo
0 5 10 15
sugars
0 100 200 300
potass
0 25 50 75 100
vitamins
1.0 1.5 2.0 2.5 3.0
shelf
0.50 0.75 1.00 1.25 1.50
weight
0.5 1.0 1.5
cups
20 40 60 80
rating