80-cereals

January 13, 2024

1 COGNORISE INFOTECH _ 80 CEREALS _ TASK 3

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

[27]:
```
# Provide the full path to the CSV file
file_path = r"C:\Users\KARTHIK\OneDrive\Desktop\CognoRise Intern\Task 3\cereal.
 ↪csv"

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)
```

[28]:
```
df.shape
```

[28]: (77, 16)

[29]:
```
df.describe()
```

[29]:

	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

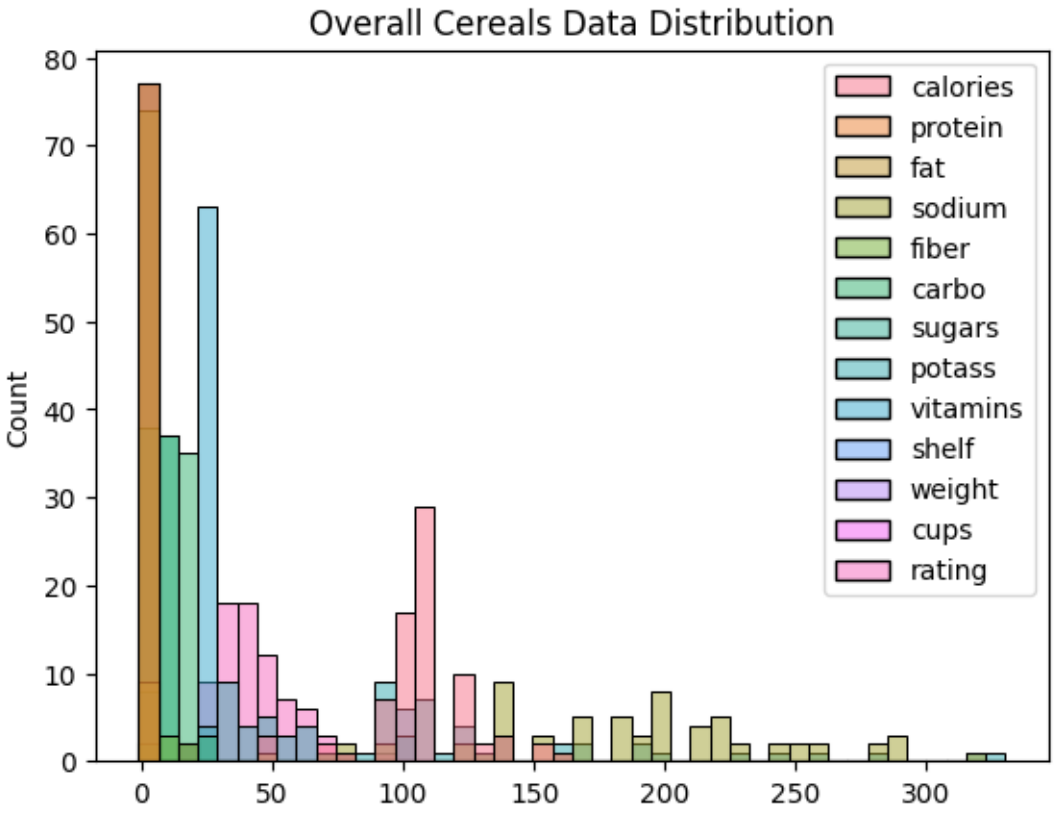
```
75%    11.000000  120.000000   25.000000   3.000000   1.000000   1.000000
max    15.000000  330.000000  100.000000   3.000000   1.500000   1.500000

          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
```

[30]:
```
df.isnull().sum().sum()
```

[30]:
```
0
```

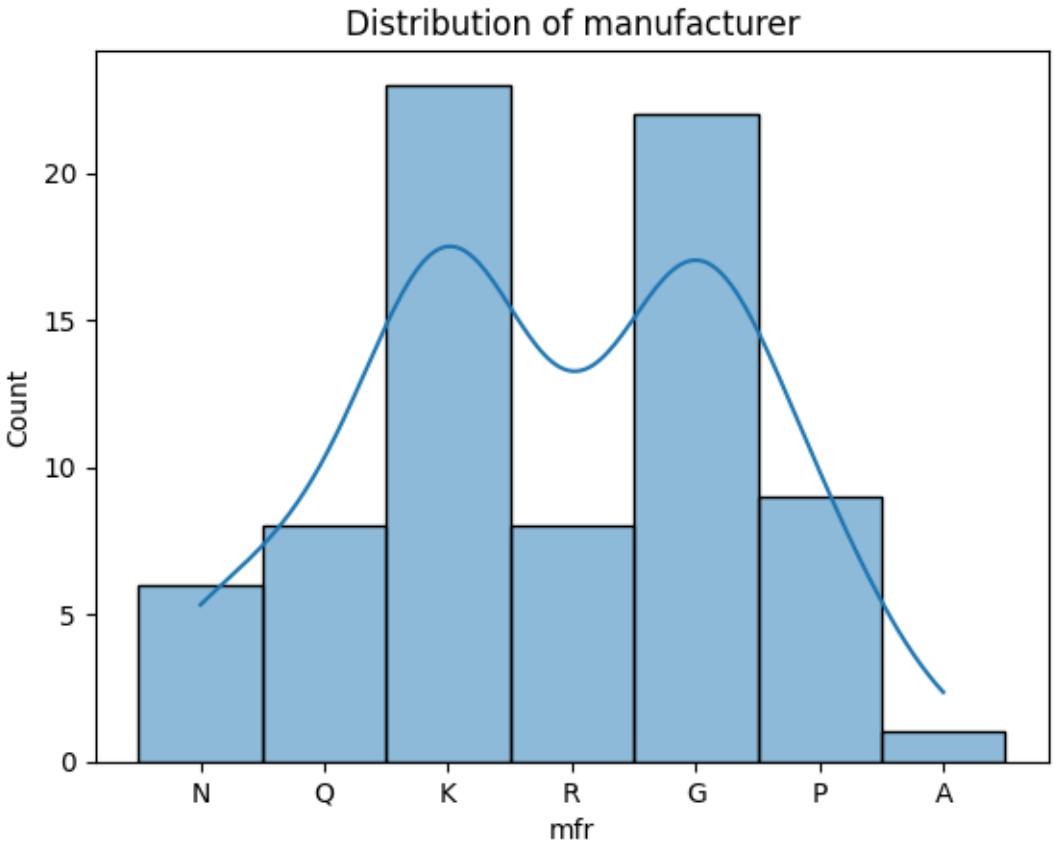
[31]:
```
plt.title("Overall Cereals Data Distribution")
sns.histplot(data = df)
```

[31]:
```
<Axes: title={'center': 'Overall Cereals Data Distribution'}, ylabel='Count'>
```



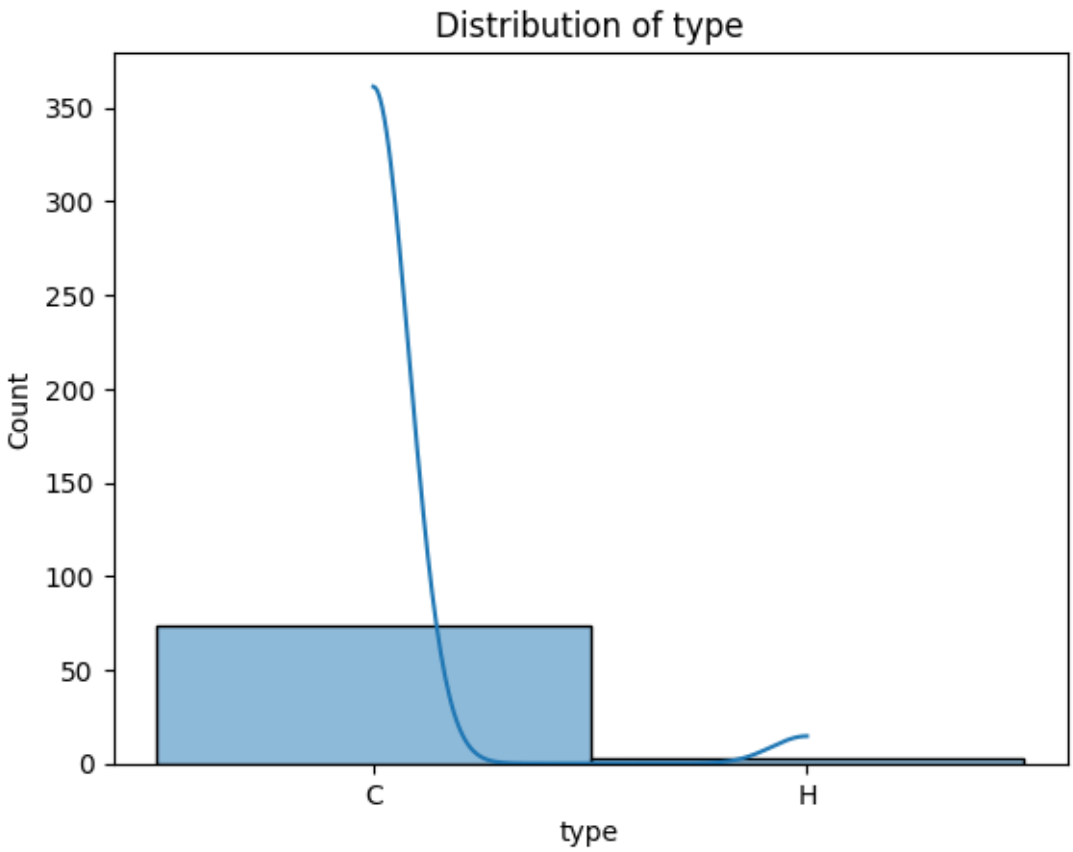
[32]:
```
sns.histplot(data=df,x='mfr', kde=True)
plt.title('Distribution of manufacturer')
```

[32]: Text(0.5, 1.0, 'Distribution of manufacturer')



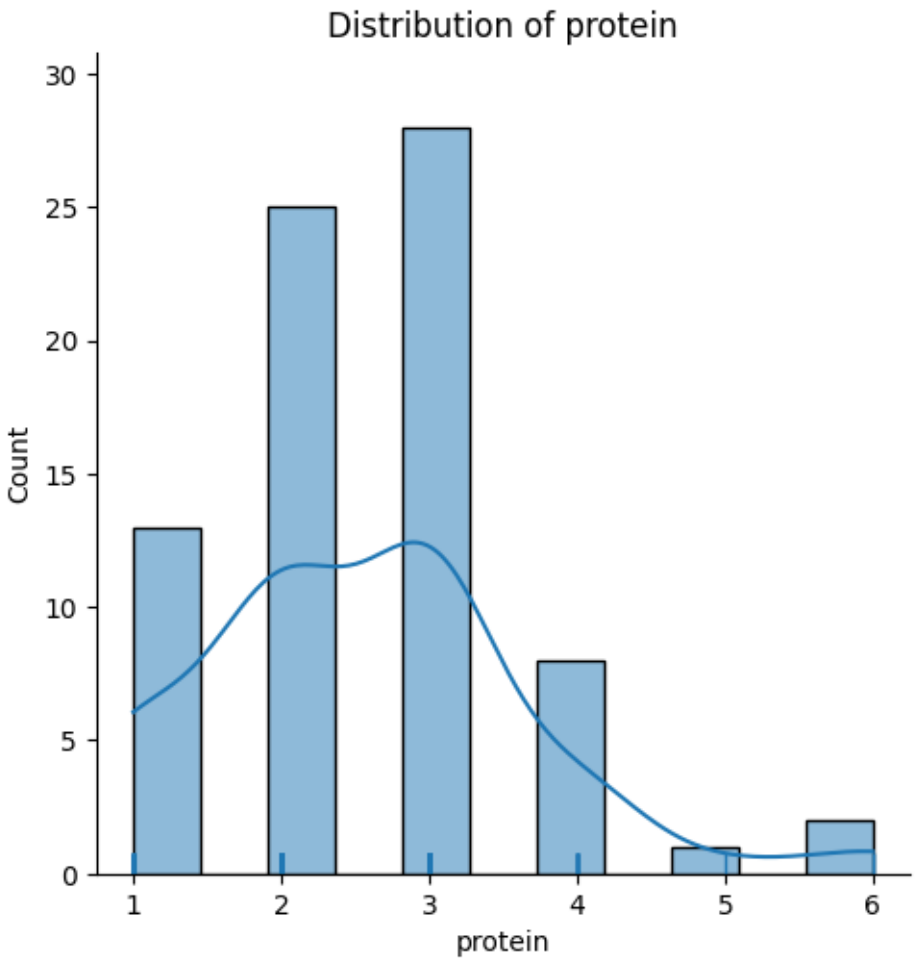
[33]:
```
sns.histplot(x='type',data=df,kde=True)
plt.title('Distribution of type')
```

[33]: Text(0.5, 1.0, 'Distribution of type')



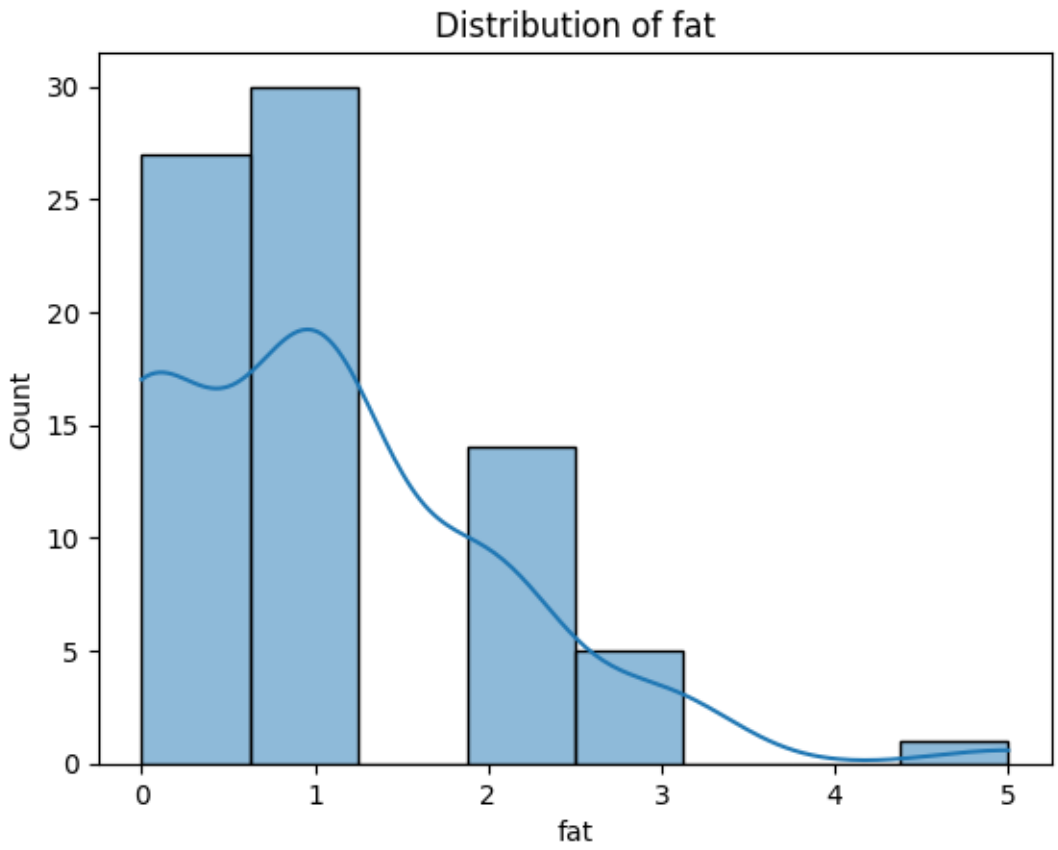
[34]:
```
sns.displot(df['protein'],rug=True,kde=True)
plt.title('Distribution of protein')
```

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



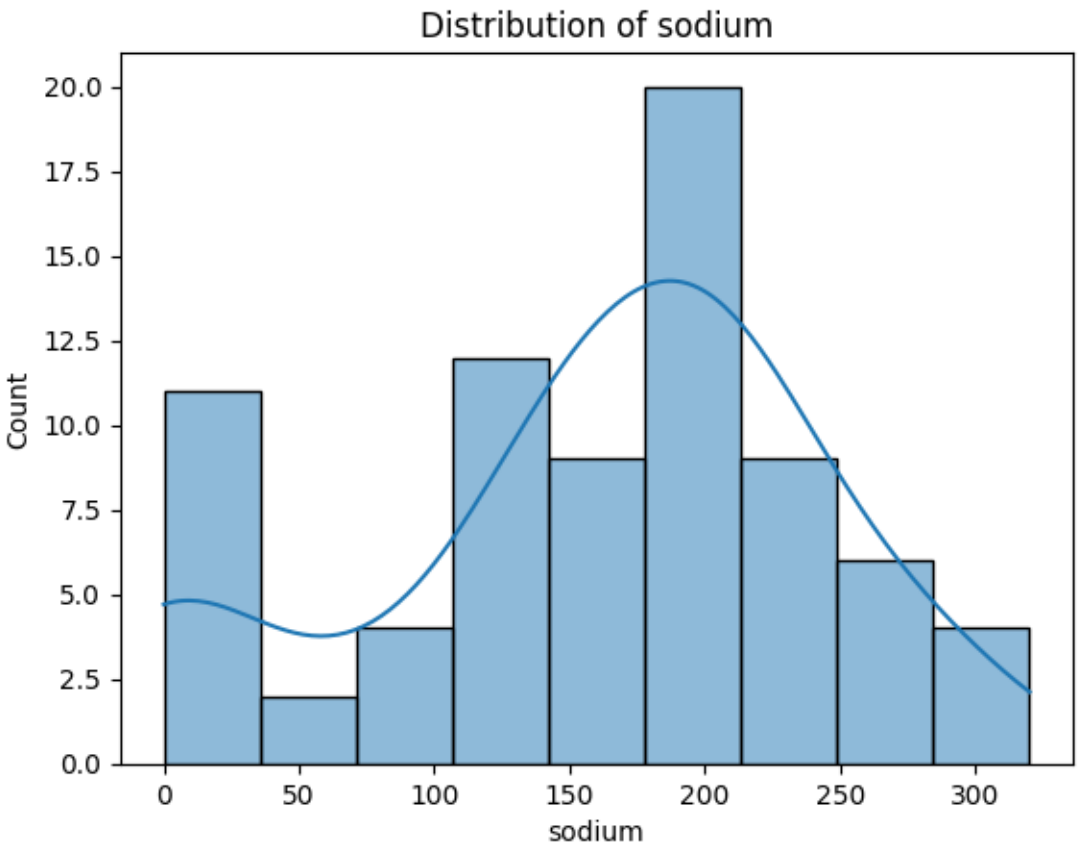
[35]:
```
sns.histplot(df['fat'],kde=True)
plt.title('Distribution of fat')
```

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



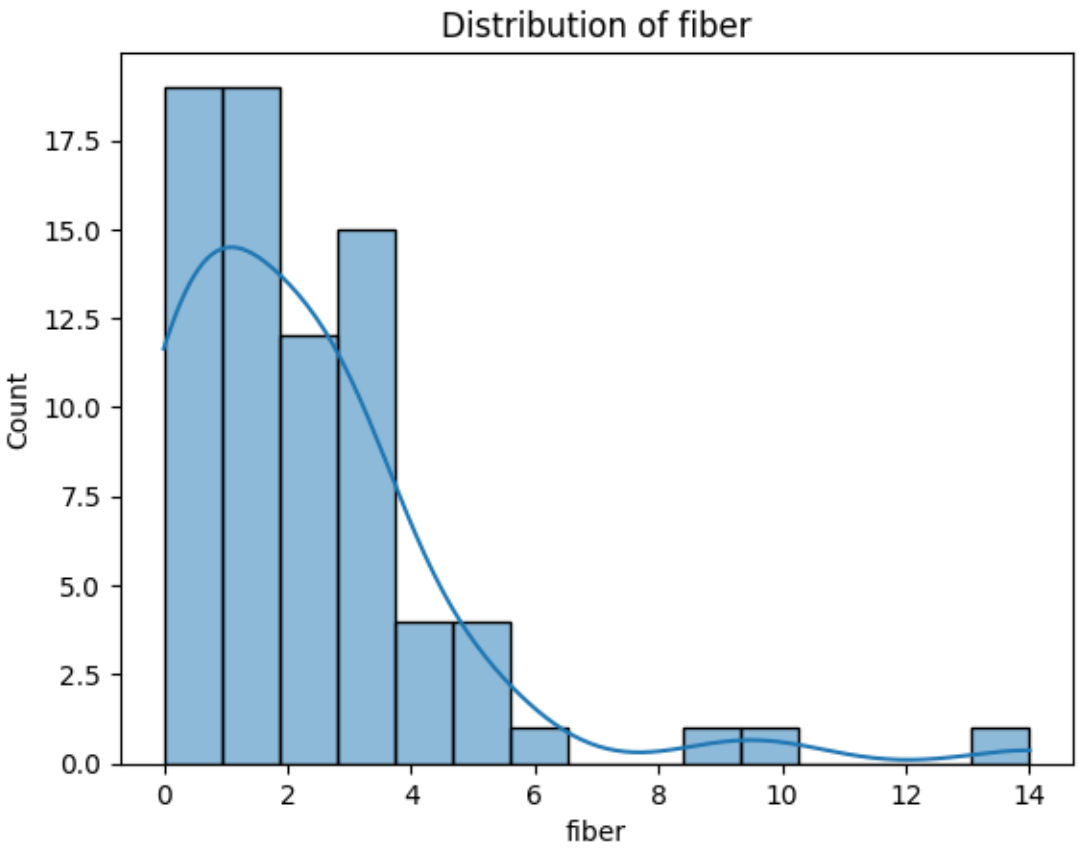
[36]:
```
sns.histplot(df['sodium'],kde=True)
plt.title('Distribution of sodium')
```

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



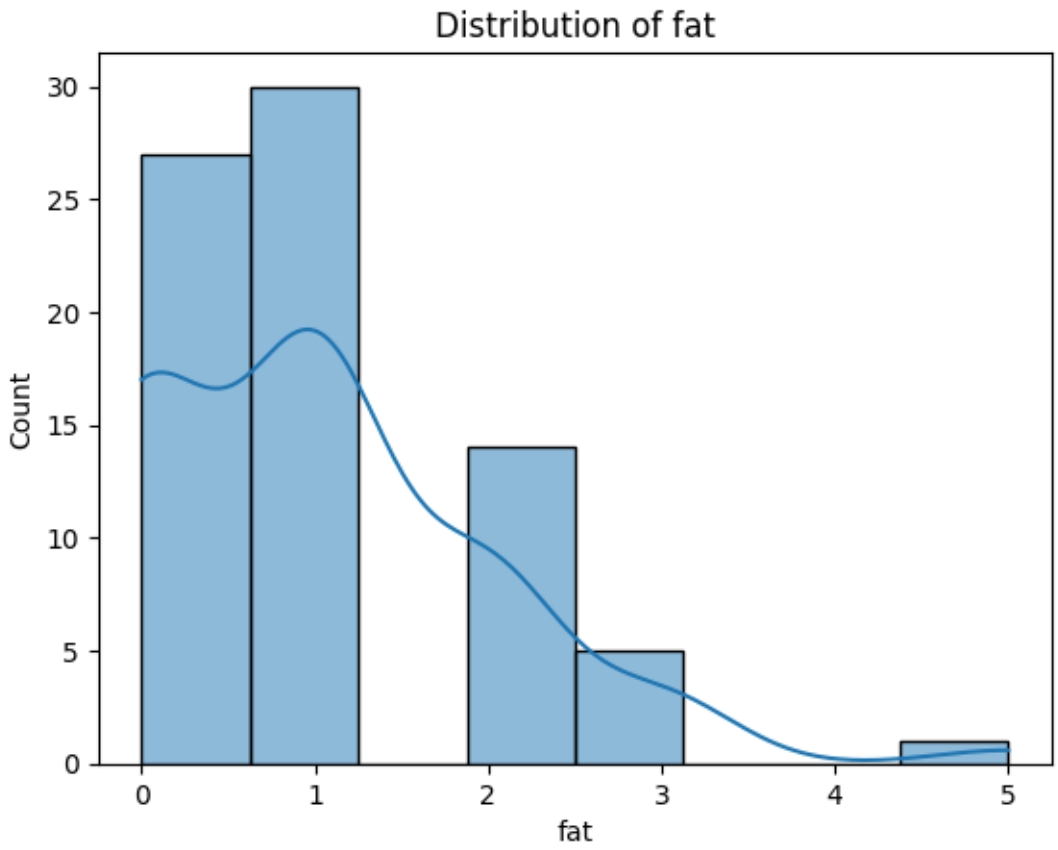
[37]:
```
sns.histplot(df['fiber'],kde=True)
plt.title('Distribution of fiber')
```

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



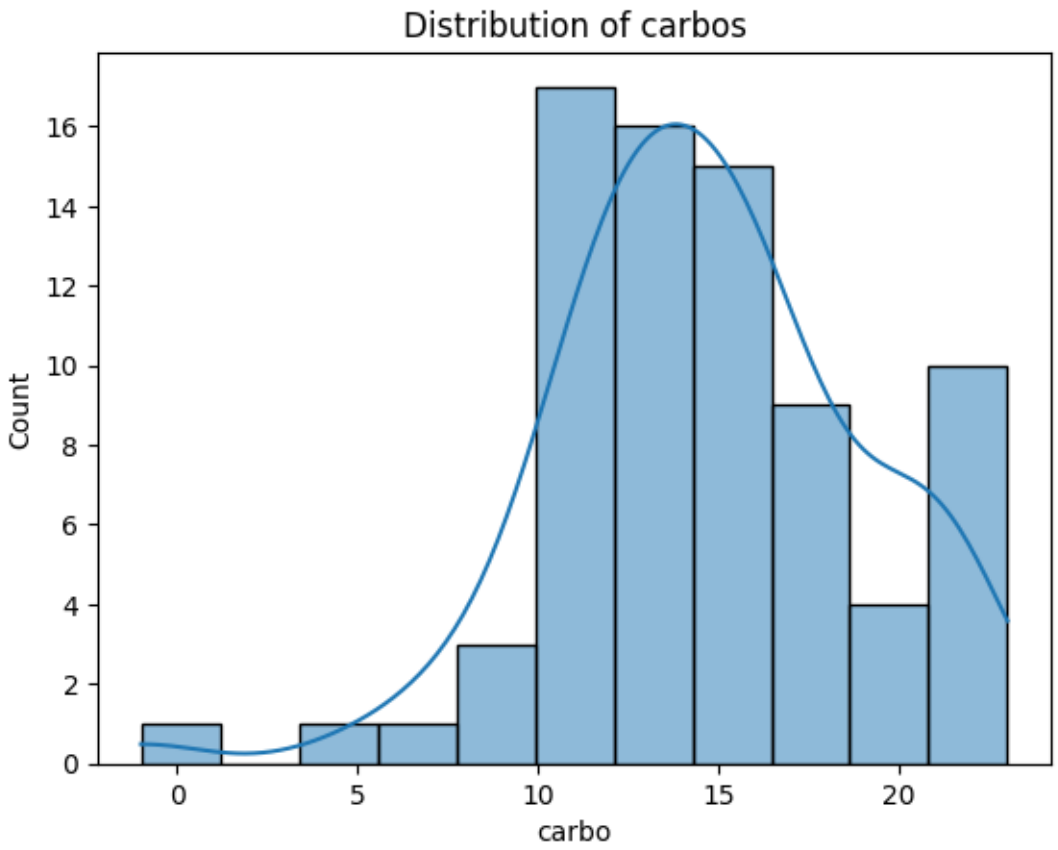
[38]:
```
sns.histplot(df['fat'],kde=True)
plt.title('Distribution of fat')
```

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



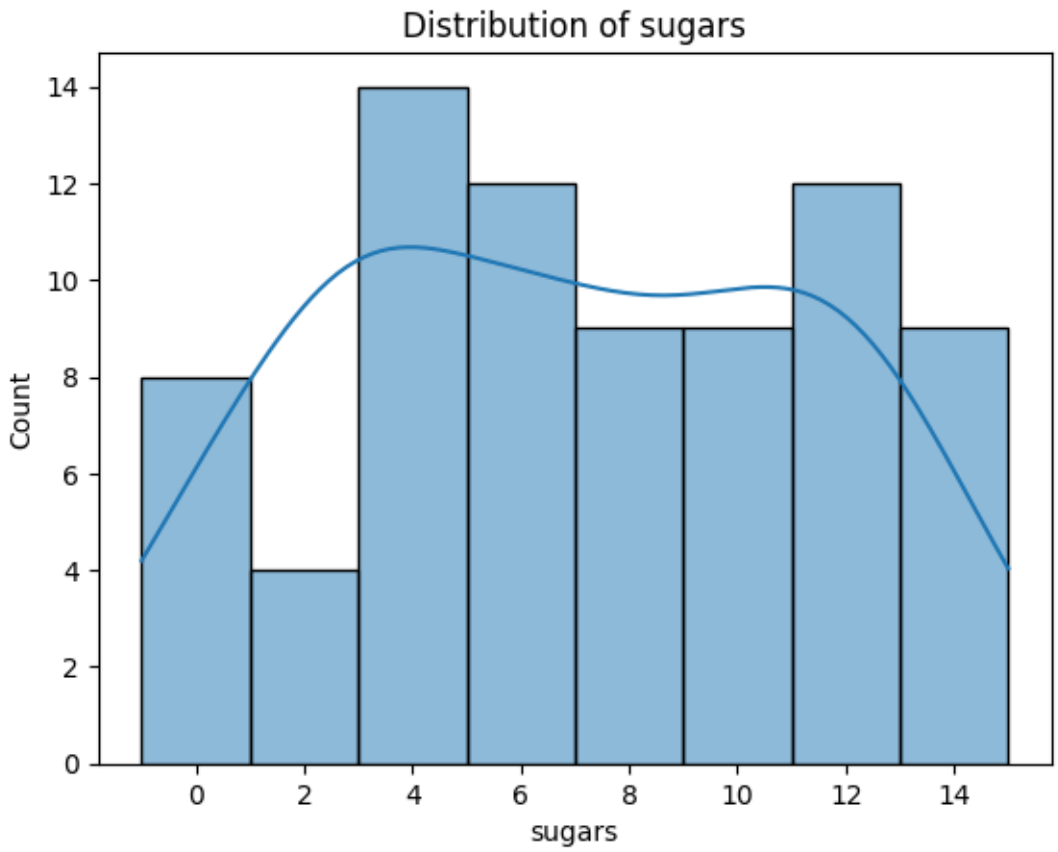
[39]:
```
sns.histplot(df['carbo'],kde=True)
plt.title('Distribution of carbos')
```

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



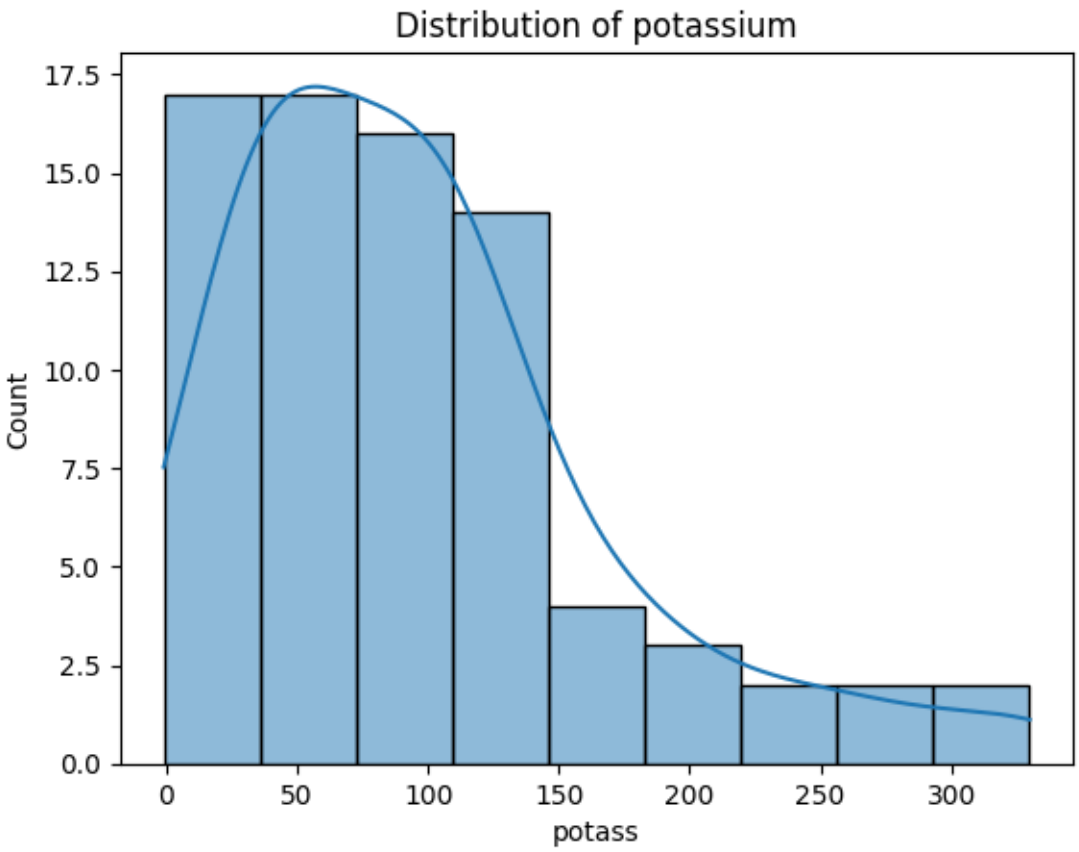
[40]:
```
sns.histplot(df['sugars'],kde=True)
plt.title('Distribution of sugars')
```

[40]:
```
Text(0.5, 1.0, 'Distribution of sugars')
```



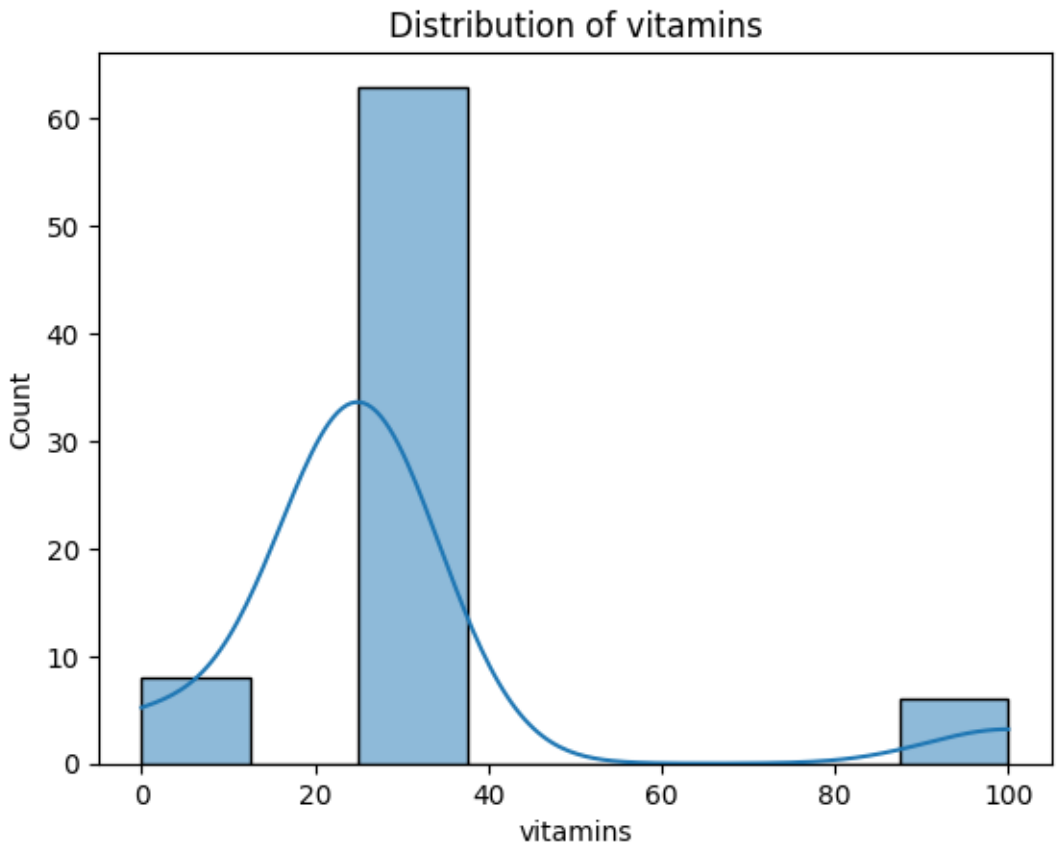
[41]:
```
sns.histplot(df['potass'],kde=True)
plt.title('Distribution of potassium')
```

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



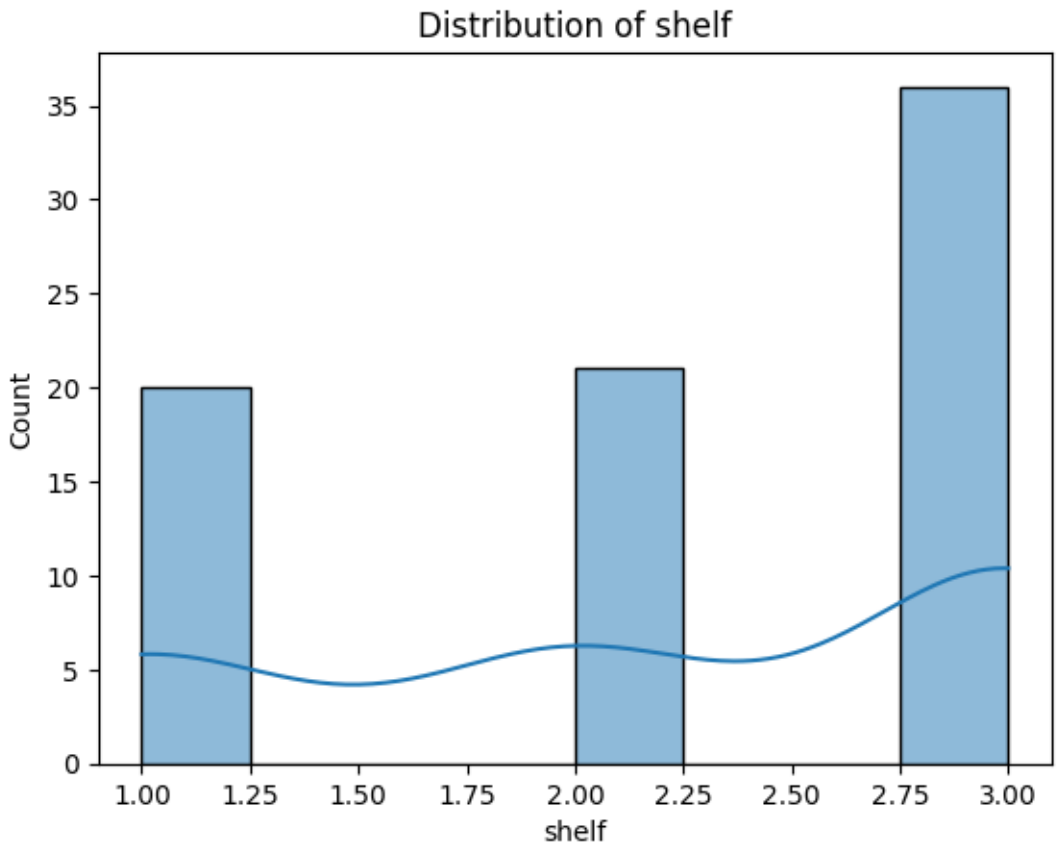
[42]:
```
sns.histplot(df['vitamins'],kde=True)
plt.title('Distribution of vitamins')
```

[42]:
```
Text(0.5, 1.0, 'Distribution of vitamins')
```



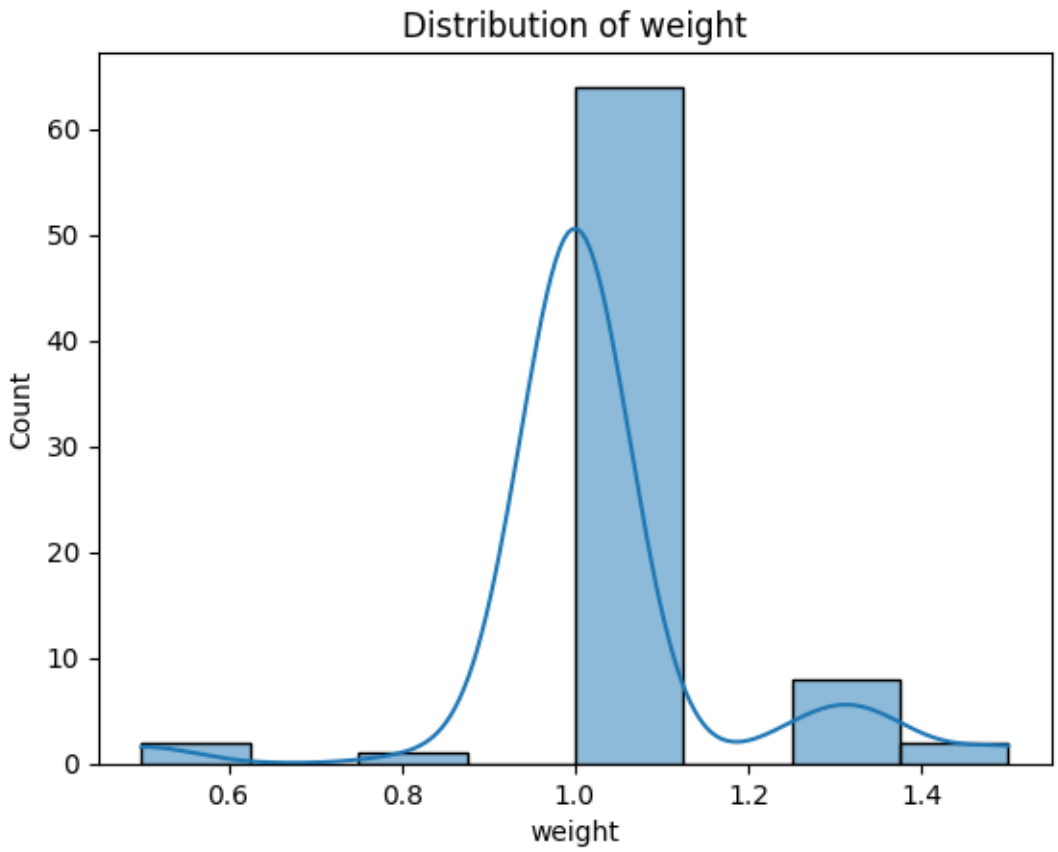
[43]:
```
sns.histplot(df['shelf'],kde=True)
plt.title('Distribution of shelf')
```

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



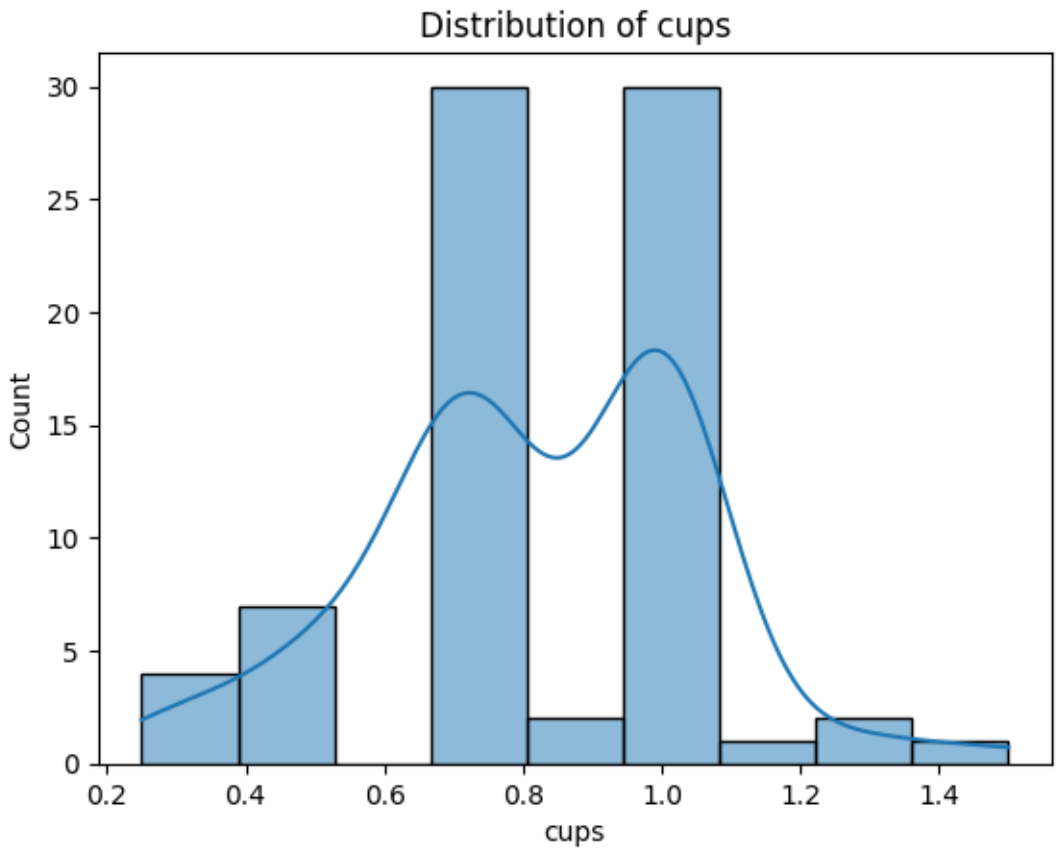
[44]:
```
sns.histplot(df['weight'],kde=True)
plt.title('Distribution of weight')
```

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



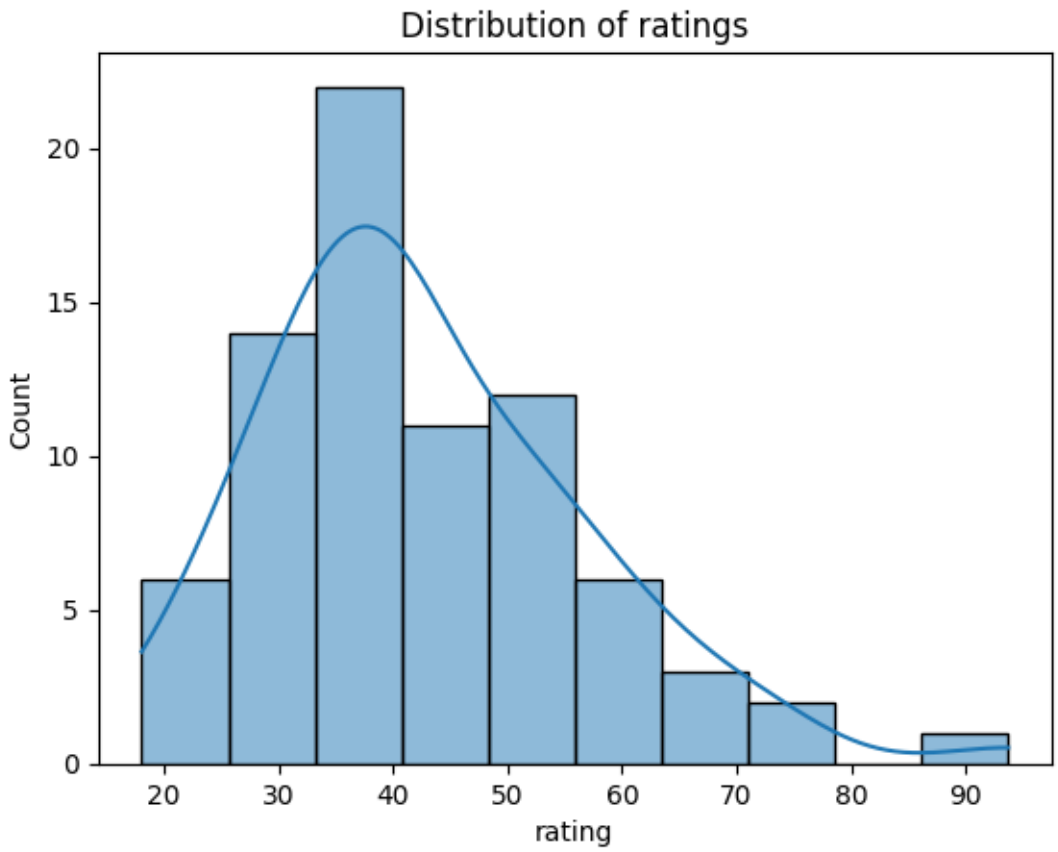
[45]:
```
sns.histplot(df['cups'],kde=True)
plt.title('Distribution of cups')
```

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



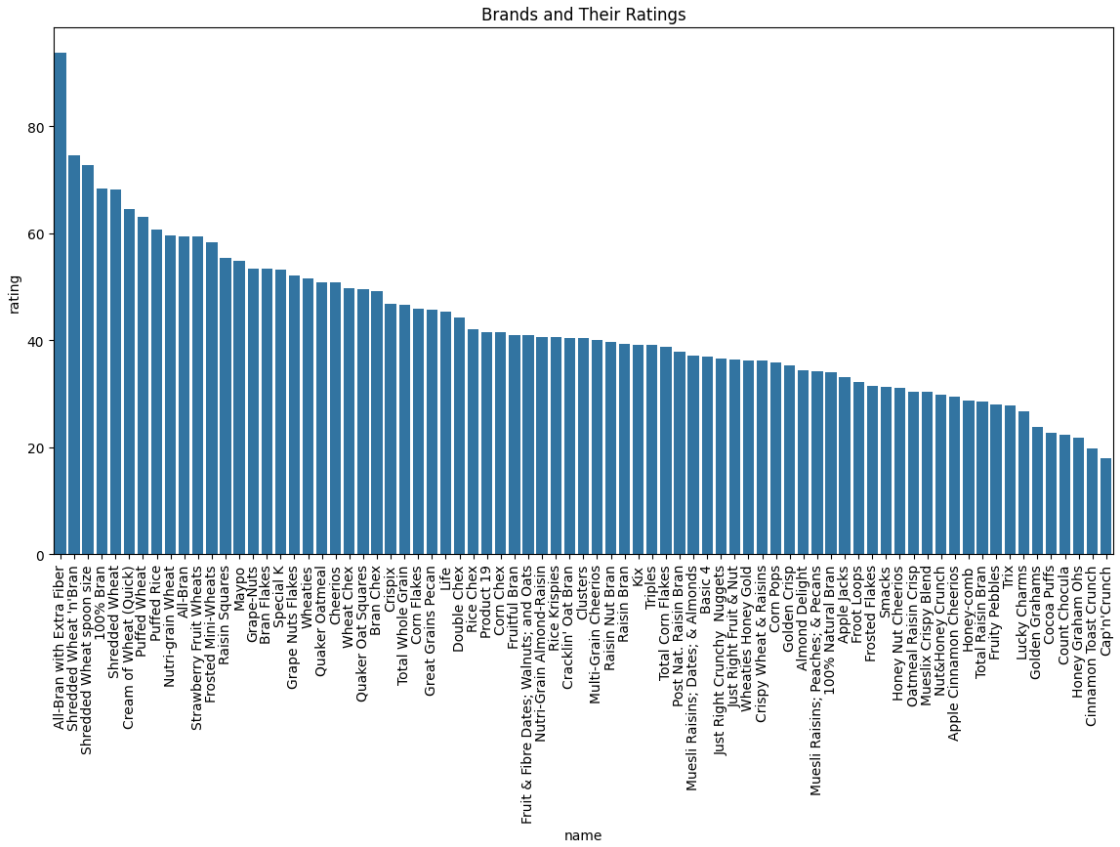
[46]:
```
sns.histplot(df['rating'],kde=True)
plt.title('Distribution of ratings')
```

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



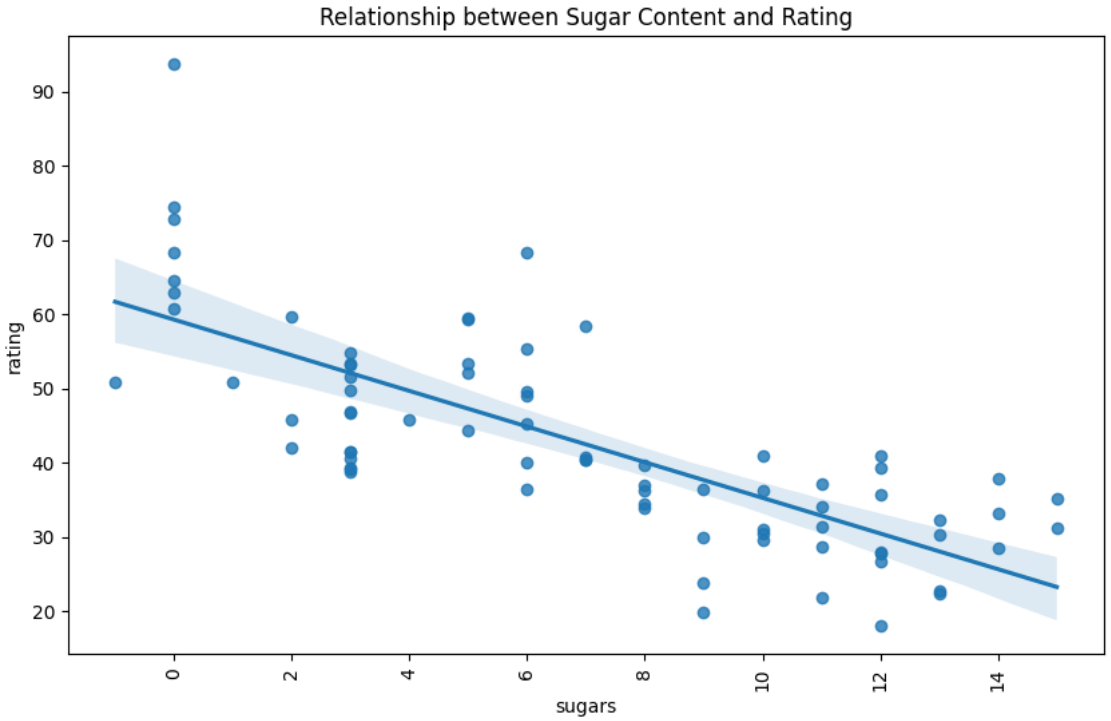
[47]:
```
# Sort the DataFrame by rating in descending order
cereals_sorted = df.sort_values(by='rating', ascending=False)

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'])
```

[47]:
```
<Axes: title={'center': 'Brands and Their Ratings'}, xlabel='name',
ylabel='rating'>
```



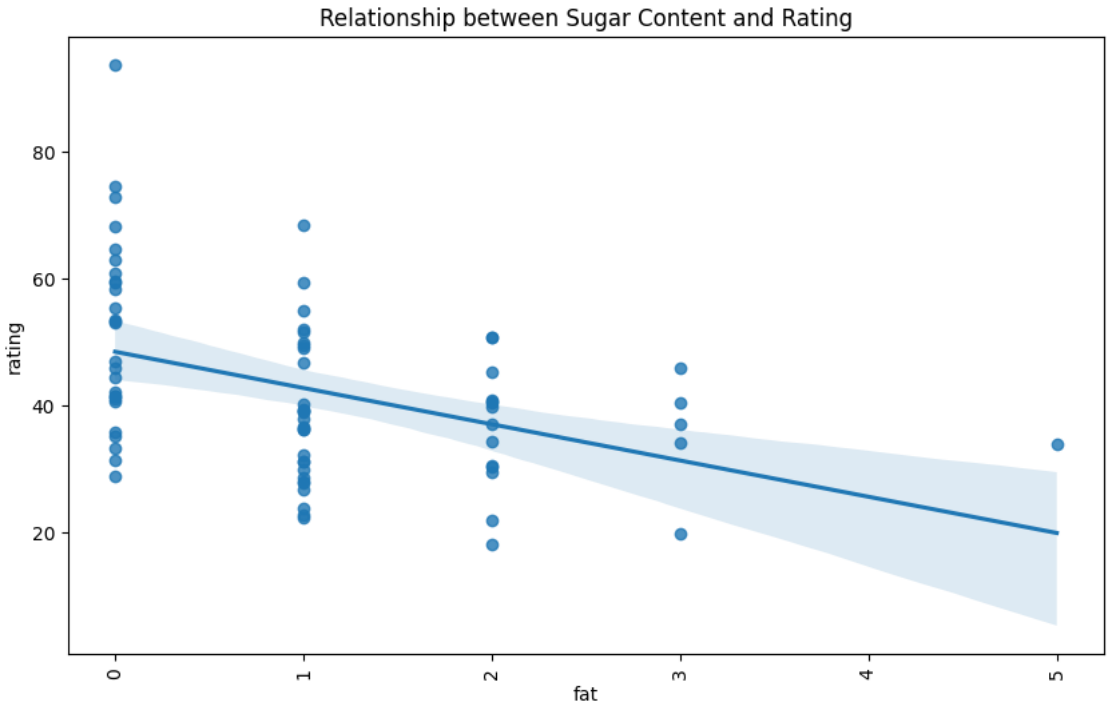
[48]:
```
plt.figure(figsize=(10, 6))
plt.title('Relationship between Sugar Content and Rating')
plt.xticks(rotation=90)
sns.regplot(data=df, x=df['sugars'], y=df['rating'])
```

[48]:
```
<Axes: title={'center': 'Relationship between Sugar Content and Rating'},
xlabel='sugars', ylabel='rating'>
```



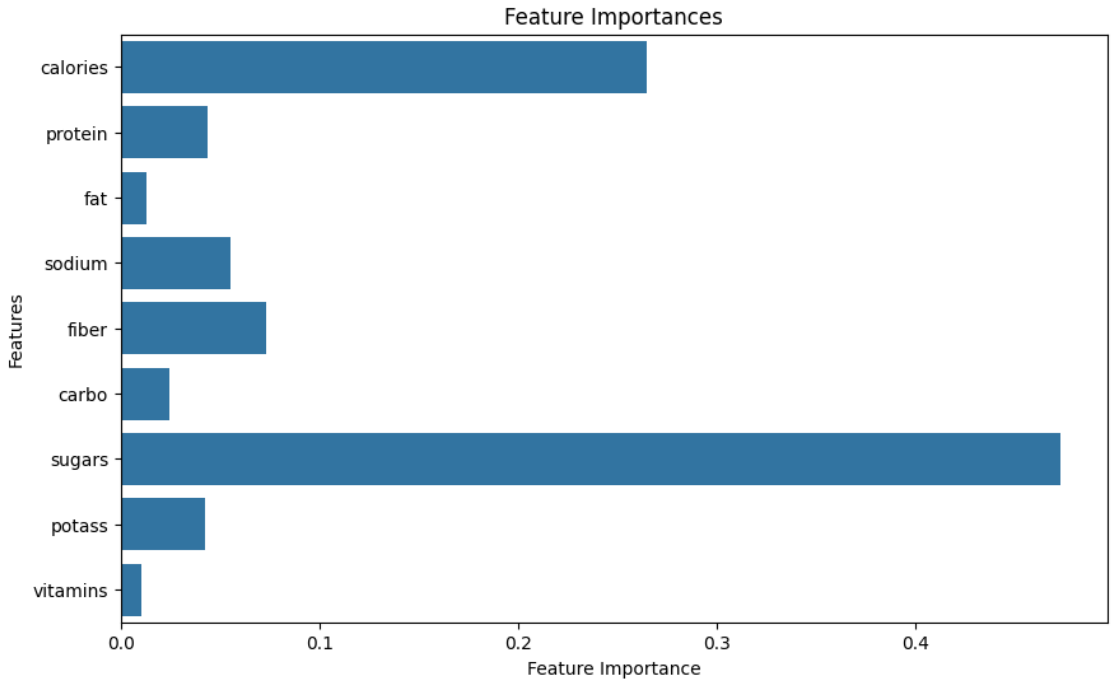
[49]:
```
plt.figure(figsize=(10, 6))
plt.title('Relationship between Sugar Content and Rating')
plt.xticks(rotation=90)
sns.regplot(data=df, x=df['fat'], y=df['rating'])
```

[49]:
```
<Axes: title={'center': 'Relationship between Sugar Content and Rating'},
xlabel='fat', ylabel='rating'>
```

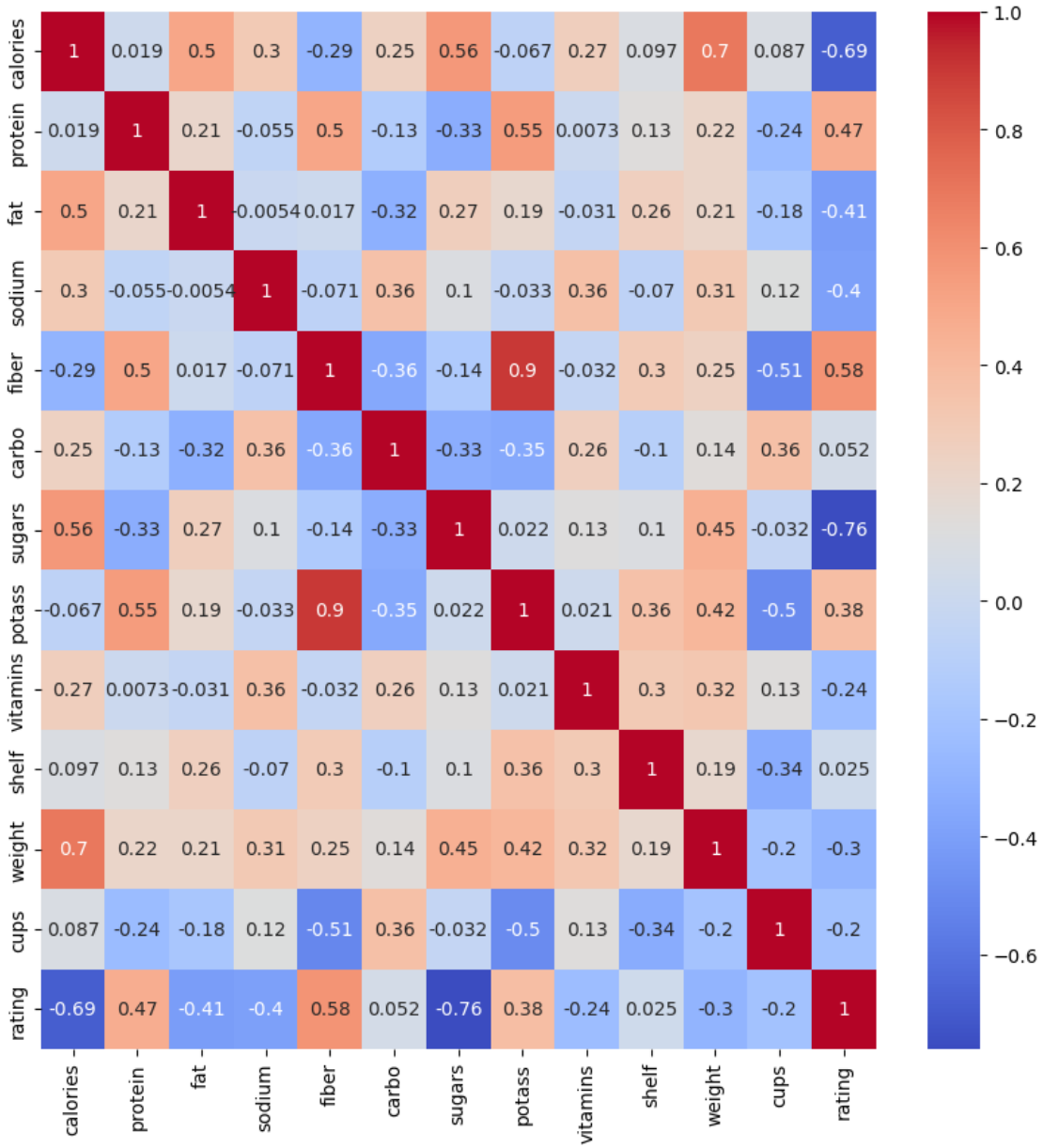


[51]:
```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor

# 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_
```

[52]:
```
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()
```

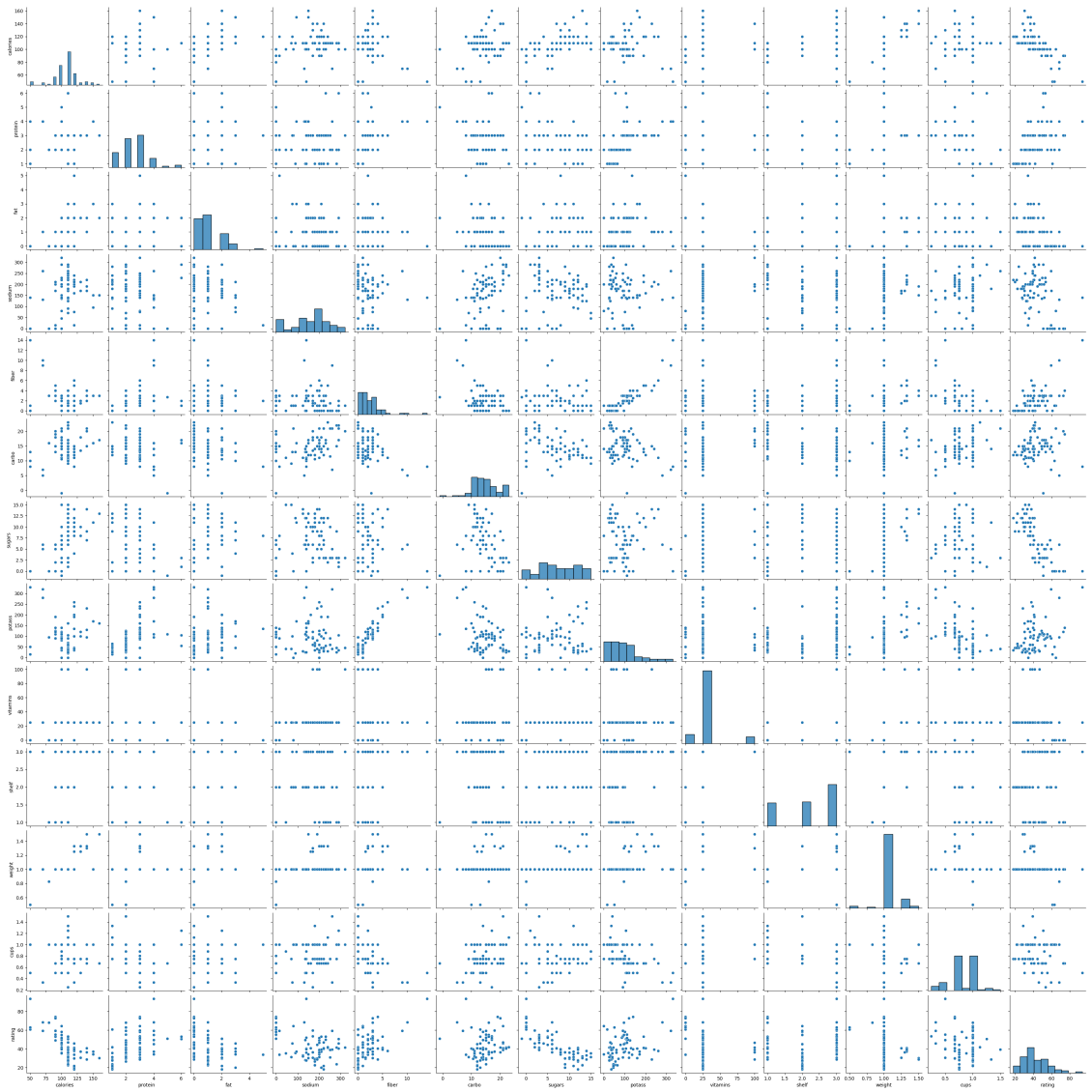


[53]:
```
fig, ax = plt.subplots(figsize=(10, 10))
num=['calories','protein','fat','sodium','fiber','carbo','sugars','potass','vitamins','shelf',
sns.heatmap(df[num].corr(), annot=True, cmap='coolwarm',ax=ax)
plt.show()
```



[54]:
```
sns.pairplot(df)
```

[54]: <seaborn.axisgrid.PairGrid at 0x17d8ebfe690>


calories
protein
fat
sodium
fiber
carbo
sugars
potass
vitamins
shelf
weight
cups
rating

[ ]: