Python Project 5 - K-Means Clustering

In this project we want to explore a dataset which contains information on breakfast cereals. We are only going to consider the manufacturers, cereal ratings, sugar and carbohydrate levels. Some of the values in sugars and carbohydrate features have invalid data which we must address.

We want to answer the following questions.

1. What cereal manufacturers have the most/least varieties?
2. What cereal has the highest rating?
3. Investigate sugar levels in cereals

- Does each manufacturer provide cereals equally among the sugar levels?
- How are the cereals clustered by sugar levels (low, medium, high)
- What are the cereals which have the highest sugar level? lowest sugar levels?
- How does your favorite breakfast cereal rank?

1. Investigate carbohydrate levels

- Cluster cereals in low, medium and high level of carbs
- What cereals are low sugar and low carbs?
- What cereals are high sugar and high carbs?

Steps:

- Input libraries and K-Means model from scikit-learn
- Create dataframe and print out a few lines
- Drop all features in dataframe except those we are using: Name, Manufacturer, Sugars and Carbohydrates, and Rating; alternately you can create a new dataframe with just the features you want
- Check to see if there are any NaN values in sugars or carbohydrates features
- Check to see if there are any invalid values (that is values <0) in sugars or carbohydrates features; if there are 2 or less cereals with these invalid values drop the cereal(s) from dataframe; otherwise replace negative values with 0.
- Plot number of cereals for each manufacturer; if any manufacturer has only 1 cereal, drop that data instance. Which manufacturer has the most varieties of cereals.
- In dataframe the manufacturer is listed by a single letter; for plotting purposes add a column giving the actual name and delete feature 'mfr'
- Use a plot to determine which manufacturer has cereals with the highest ratings.
- What cereal has the highest rating?
- Plot sugars levels vs manufacturer and determine which brand has lowest sugar levels.
- Cluster data using K-Means and the sugars feature with clusters low, middle and high sugar levels; print out cluster centroids; add cluster as feature in dataframe.
- Determine which cluster is associated with low, mid or high sugar levels; add feature giving sugar level corresponding to cluster
- Create a plot to show how cereals are distributed among sugar levels; what cereals have highest sugar levels? lowest sugar levels?
- Repeat the steps you did for investigating sugar levels using carbohydrate levels.
- Print out the cereals that are low sugars and low carbohydrates.
- Print out the cereals that are low sugars and high carbohydrates.

This dataset is contained in the file `cereal.csv'

Data Set Description

This dataset contains information on 77 different breakfast cereals. The features are

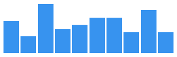
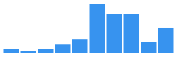
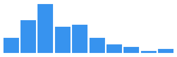
1. Cereal Name
2. Manufacturer

- A → American Home Food Products
- G → General Mills
- K → Kelloggs
- N → Nabisco
- P → Post
- Q → Quaker Oats
- R → Ralston Purina

1. Type (C → Cold, H → Hot)
2. Calories (per serving)
3. Protein (in grams)
4. Fat (in grams)
5. Sodium (in milligrams)
6. Fiber (in grams of dietary fiber)
7. Carbo (grams of complex carbohydrates)
8. Sugars (in grams)
9. Potass (milligrams of potassium)
10. Vitamins (possible values are 0, 25 or 100 indicating percent of FDA recommended vitamins and minerals)
11. Shelf (display shelf with possible values 1, 2, or 3 counting from floor)
12. Weight (weight of one serving in ounces)
13. Cups (number of cups in one serving)
14. Rating (Consumer ratings of cereal from 0 to 100)

```
# import libraries  and K-Means function

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.cluster import KMeans
```

```
# Create dataframe for data
df = pd.read_csv('cereal.csv')
df = df[['name', 'mfr', 'sugars', 'carbo', 'rating']]
df
```

	name object	mfr object	sugars int64	carbo float64	rating float64
	100% Bran .... 1.3% 100% Natura... 1.3% 75 others 97.4%	K .... 29.9% G .... 28.6% 5 others 41.6%	-1 - 15 	-1.0 - 23.0 	18.042851 - 93.704... 
0	100% Bran	N	6	5.0	68.402973
1	100% Natural Bran	Q	8	8.0	33.983679
2	All-Bran	K	5	7.0	59.425505
3		K	0	8.0	93.704912

	All-Bran with Extra					
4	Almond Delight	R	8	14.0	34.384843	
5	Apple Cinnamon Cheerios	G	10	10.5	29.509541	
6	Apple Jacks	K	14	11.0	33.174094	
7	Basic 4	G	8	18.0	37.038562	
8	Bran Chex	R	6	15.0	49.120253	
9	Bran Flakes	P	5	13.0	53.313813	

This chart is empty

```
# drop features we aren't using or create a new dataframe with only the features we want
```

```
# Check to see if any instances have NaN for entries using .info
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77 entries, 0 to 76
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   name    77 non-null     object 
 1   mfr     77 non-null     object 
 2   sugars  77 non-null     int64  
 3   carbo   77 non-null     float64
 4   rating  77 non-null     float64
dtypes: float64(2), int64(1), object(2)
memory usage: 3.1+ KB
None
```

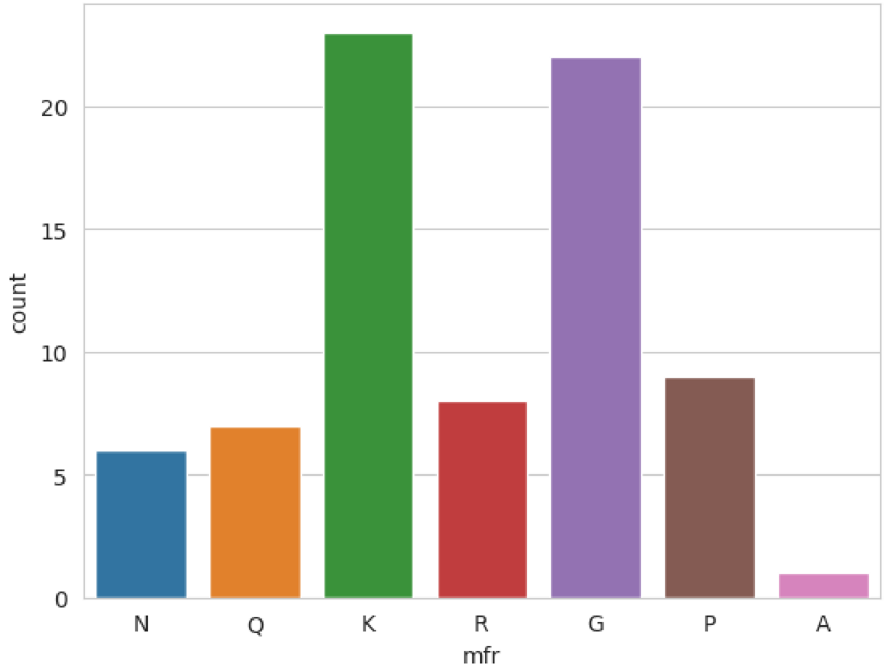
```
# Check to see if any cereals have negative values for sugars or carbohydrates; if 2 or less cereals, drop those
# data instances; otherwise replace negative values with 0
neg_sugars = df[df['sugars'] < 0]
neg_carbs = df[df['carbo'] < 0]

if len(neg_sugars) <= 2:
    df = df[df['sugars'] >= 0]
if len(neg_carbs) <= 2:
    df = df[df['carbo'] >= 0]
```

```
# Address negative values
df.loc[df['sugars'] < 0, 'sugars'] = 0
df.loc[df['carbo'] < 0, 'carbo'] = 0
```

```
# set background grid for plots
sns.set_style('whitegrid')
```

```
# Plot number of products per manufacturer
sns.countplot(x='mfr', data=df)
```

```
<AxesSubplot:xlabel='mfr', ylabel='count'>
```

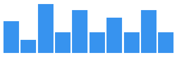
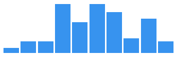
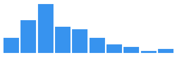


```
# There is only one cereal from American Home Foods Company so we drop that data sample
#
counts = df.mfr.value_counts()

# Type df[df.mfr == 'A']  and it will give you all data from this manufacturer.
# Make sure you set df = drop command; use command to make sure it was deleted
#

drop_mfrs = counts[counts == 1].index
df = df[~df.mfr.isin(drop_mfrs)]

df[df.mfr == 'A']
```

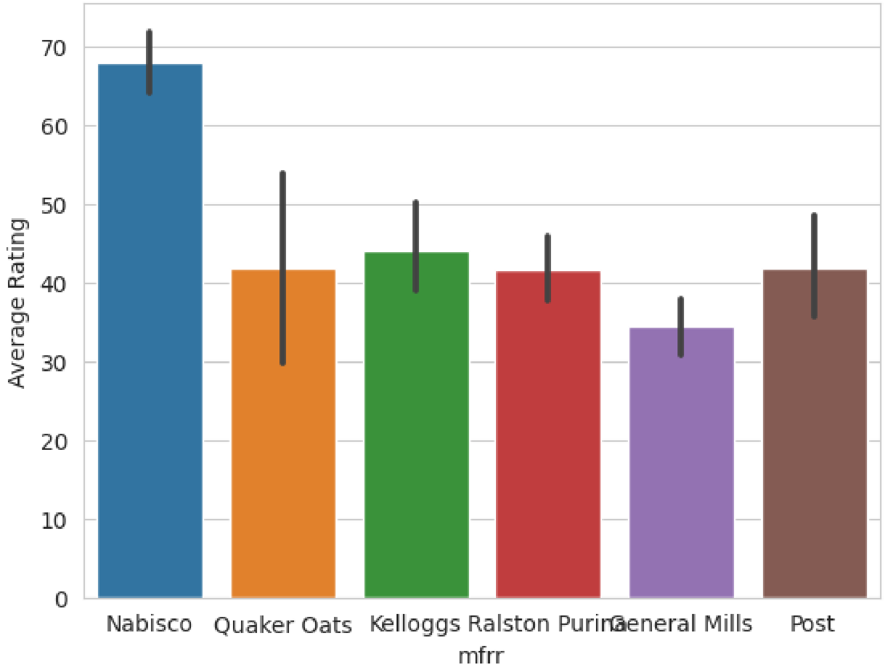
	name object	mfr object	sugars int64	carbo float64	rating float64

```


# For plots we would like the name of manufacturer instead of just "N" or "Q"
# Use .map or .apply
#

mfr_names = {'A': 'American Home Food Products', 'G': 'General Mills', 'K': 'Kelloggs',
             'N': 'Nabisco', 'P': 'Post', 'Q': 'Quaker Oats', 'R': 'Ralston Purina'}


df['mfrr'] = df['mfr'].map(mfr_names)

# Drop the 'mfr' column
```

```
df = df.drop('mfr', axis=1)
df
```

	name object 100% Bran .... 1.3% 100% Natura... . 1.3% 73 others 97.3%	sugars int64 0 - 15 	carbo float64 5.0 - 23.0 	rating float64 18.042851 - 93.704... 	mfrr object Kelloggs .... 30.7% General Mills .. 29.3% 4 others 40%
0	100% Bran	6	5.0	68.402973	Nabisco
1	100% Natural Bran	8	8.0	33.983679	Quaker Oats
2	All-Bran	5	7.0	59.425505	Kelloggs
3	All-Bran with Extra Fiber	0	8.0	93.704912	Kelloggs
4	Almond Delight	8	14.0	34.384843	Ralston Purina
5	Apple Cinnamon Cheerios	10	10.5	29.509541	General Mills
6	Apple Jacks	14	11.0	33.174094	Kelloggs
7	Basic 4	8	18.0	37.038562	General Mills
8	Bran Chex	6	15.0	49.120253	Ralston Purina
9	Bran Flakes	5	13.0	53.313813	Post

```
# Plot to see which manufacturer  has highest rated cereals
sns.barplot(x='mfrr', y='rating', data=df)
plt.ylabel('Average Rating')
plt.show()
```

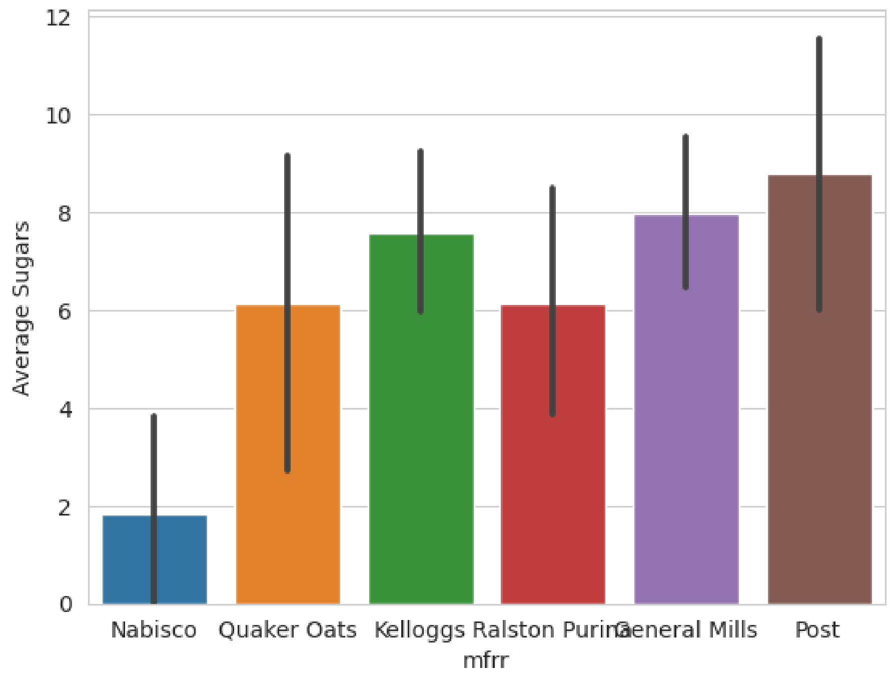


```
# Which cereal is rated highest?
print('Nabisco is rated the highest, above Kelloggs and Post')
```

```
Nabisco is rated the highest, above Kelloggs and Post
```

```
# Look at sugars per brand by plotting
sns.barplot(x='mfrr', y='sugars', data=df)

plt.ylabel('Average Sugars')
```

```
plt.show()
```



Cluster by sugars into highest, middle and lowest levels; random initial guess

```
# Get data for clustering
sugars_df = df[['sugars']]
```

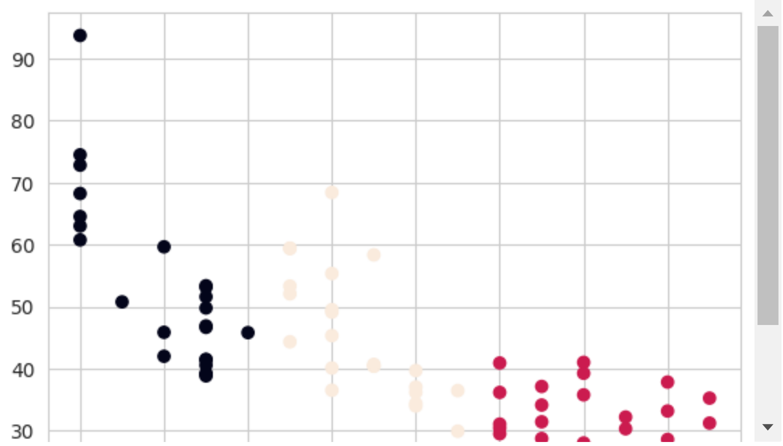
```
# Form model, fit data and print out cluster centers
kmeans = KMeans(n_clusters=3)
kmeans.fit(sugars_df)
print(kmeans.cluster_centers_)
```

```
[[ 1.95833333]
 [12.03846154]
 [ 6.84      ]]
```

```
# Add column to dataframe for this clusters, say sugars_clusters
sugars_clusters = kmeans.predict(sugars_df)


df['sugars_clusters'] = sugars_clusters
```

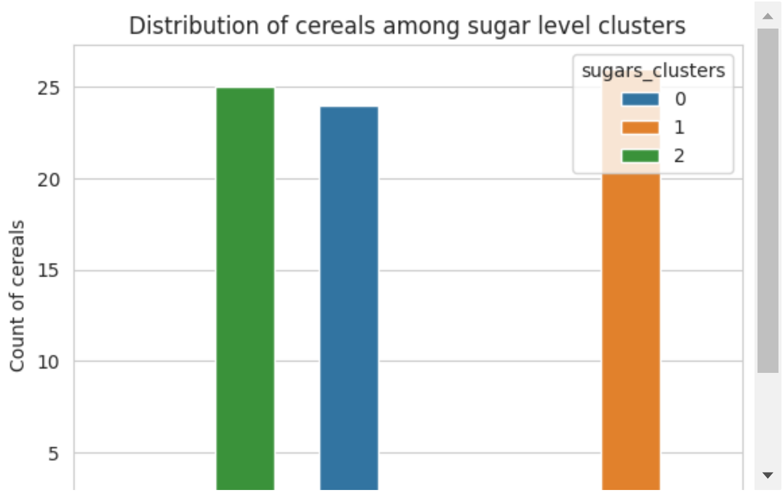
```python
# Plot clusters
plt.scatter(sugars_df, df['rating'], c=df['sugars_clusters'])
```

```
<matplotlib.collections.PathCollection at 0x7f91860d1790>
```



```python
# Determine which cluster number corresponds to lowest, middle and highest level and create a new
# column in dataframe using .map

low_sugar_cluster = 0
middle_sugar_cluster = 1
high_sugar_cluster = 2


df['sugar_level'] = df['sugars_clusters'].map({low_sugar_cluster: 'low',
                                               middle_sugar_cluster: 'middle',
                                               high_sugar_cluster: 'high'})


#
```

```python
# How are cereals distributed among the 3 levels?
sns.countplot(x='sugar_level', hue='sugars_clusters', data=df)
plt.xlabel('Sugars level')
plt.ylabel('Count of cereals')
plt.title('Distribution of cereals among sugar level clusters')
plt.show()

# the most cereals sold on average are the ones sold in the middle sugar levels
```



```
# Which cereals have the highest sugar levels
#

highest_sugar_cereals = df.sort_values('sugars', ascending=False).head(10)


print(highest_sugar_cereals[['name', 'sugars']])

#printed is the top 10 cereals with the highest sugar levels with Smacks, Golden Crips, and Post Nat. Raisin Bran.
```

```
                     name  sugars
66                 Smacks      15
30           Golden Crisp      15
52  Post Nat. Raisin Bran      14
70      Total Raisin Bran      14
6             Apple Jacks      14
24            Froot Loops      13
46   Mueslix Crispy Blend      13
14            Cocoa Puffs      13
18          Count Chocula      13
10           Cap'n'Crunch      12
```

```
# Which cereals have the lowest sugar levels
#

lowest_sugar_cereals = df.sort_values('sugars', ascending=True).head(10)


print(lowest_sugar_cereals[['name', 'sugars']])

#printed are the 10 cereals with the lowest sugar levels with Cream of wheat, All-Bran with extra fiber, and Puffed
```

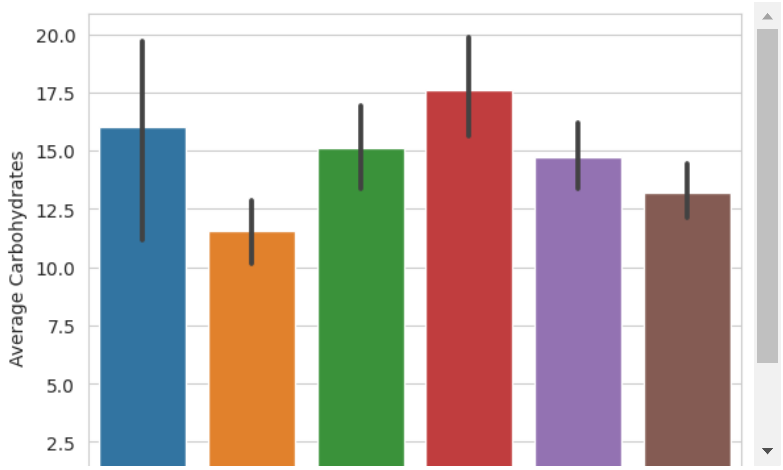
```
                        name  sugars
20      Cream of Wheat (Quick)       0
3   All-Bran with Extra Fiber       0
54                 Puffed Rice       0
55                Puffed Wheat       0
65  Shredded Wheat spoon size       0
63              Shredded Wheat       0
64      Shredded Wheat 'n'Bran       0
11                    Cheerios       1
50           Nutri-grain Wheat       2
61                   Rice Chex       2
```

```python
# If you eat a particular cereal like Apple Jacks, Froot Loops, etc. what cluster is it in?
my_cereal = 'Cheerios'
print(f"The data instance and sugar cluster for {my_cereal} is {df[df.name == my_cereal].sugars_clusters}")
```

```
The data instance and sugar cluster for Cheerios is 11    0
Name: sugars_clusters, dtype: int32
```

Repeat calculations and plots for carbohydrates instead of sugars

```python
 sns.barplot(x='mfrr', y='carbo', data=df)

plt.ylabel('Average Carbohydrates')


plt.show()
```

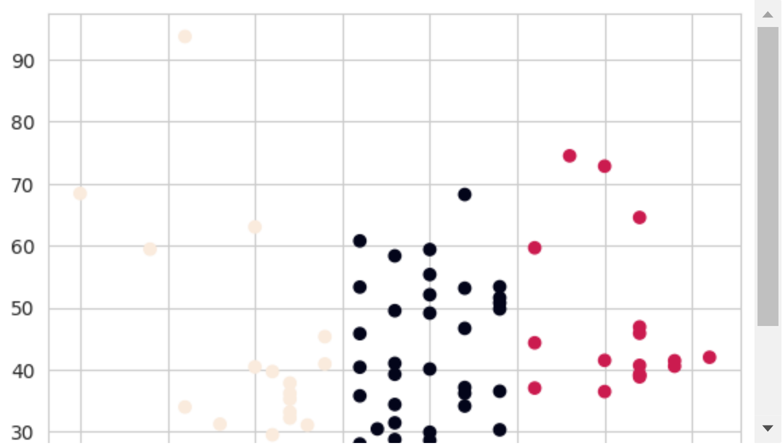


```python
carbo_df = df[['carbo']]
kmeans = KMeans(n_clusters=3)
kmeans.fit(carbo_df)
```

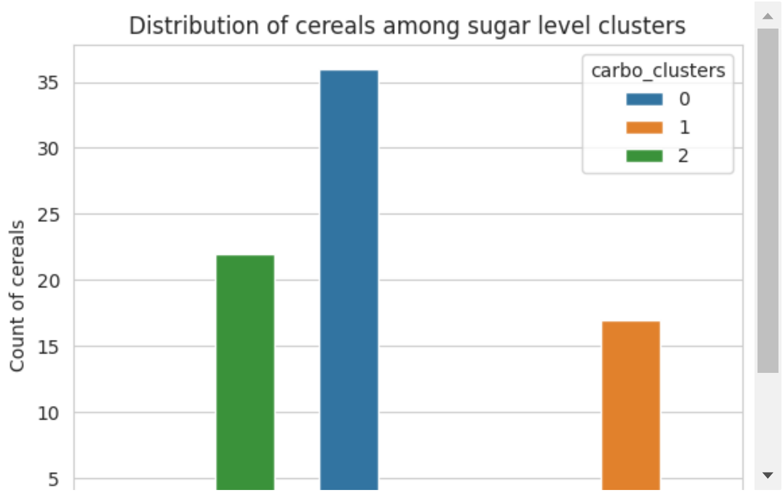
```
print(kmeans.cluster_centers_)
carbo_clusters = kmeans.predict(carbo_df)

df['carbo_clusters'] = carbo_clusters
plt.scatter(carbo_df, df['rating'], c=df['carbo_clusters'])
```

```
[[14.81944444]
 [20.41176471]
 [10.38636364]]
```

```
<matplotlib.collections.PathCollection at 0x7f91803c5210>
```



```
low_carbo_cluster = 0
middle_carbo_cluster = 1
high_carbo_cluster = 2


df['carbo_level'] = df['carbo_clusters'].map({low_carbo_cluster: 'low',
                                              middle_carbo_cluster: 'middle',
                                              high_carbo_cluster: 'high'})

# How are cereals distributed among the 3 levels?
sns.countplot(x='carbo_level', hue='carbo_clusters', data=df)
plt.xlabel('Carbohydrate level')
plt.ylabel('Count of cereals')
plt.title('Distribution of cereals among sugar level clusters')
plt.show()
#Low carbohydrate levels have the highest sales on average and middle with the lowest sales on average

lowest_sugar_cereals = df.sort_values('carbo', ascending=True).head(10)


print(lowest_sugar_cereals[['name', 'carbo']])

highest_sugar_cereals = df.sort_values('carbo', ascending=False).head(10)


print(highest_sugar_cereals[['name', 'carbo']])
```



```
                      name  carbo
0                100% Bran    5.0
2                 All-Bran    7.0
1        100% Natural Bran    8.0
3   All-Bran with Extra Fiber    8.0
66                  Smacks    9.0
19        Cracklin' Oat Bran   10.0
55             Puffed Wheat   10.0
5     Apple Cinnamon Cheerios   10.5
59           Raisin Nut Bran   10.5
30             Golden Crisp   11.0
                      name  carbo
61               Rice Chex   23.0
15               Corn Chex   22.0
62           Rice Krispies   22.0
72                 Triples   21.0
49  Nutri-Grain Almond-Raisin   21.0
21                 Crispix   21.0
40                     Kix   21.0
16             Corn Flakes   21.0
20     Cream of Wheat (Quick)   21.0
69       Total Corn Flakes   21.0
```

What cereals are high carbs and low sugar? What are low carbs and low sugar?

```
high_carbs_low_sugar = df[(df['carbo'] > df['carbo'].mean()) & (df['sugars'] < df['sugars'].mean())]
print(high_carbs_low_sugar[['name', 'carbo', 'sugars']])
```

```
                          name  carbo  sugars
8                    Bran Chex   15.0       6
11                    Cheerios   17.0       1
15                   Corn Chex   22.0       3
16                 Corn Flakes   21.0       2
```

```
20      Cream of Wheat (Quick)   21.0       0
21                     Crispix   21.0       3
23                 Double Chex   18.0       5
32           Grape Nuts Flakes   15.0       5
33                  Grape-Nuts   17.0       3
38  Just Right Crunchy  Nuggets   17.0       6
40                         Kix   21.0       3
47          Multi-Grain Cheerios   15.0       6
49    Nutri-Grain Almond-Raisin   21.0       7
50           Nutri-grain Wheat   18.0       2
53                  Product 19   20.0       3
60              Raisin Squares   15.0       6
61                   Rice Chex   23.0       2
62               Rice Krispies   22.0       3
63              Shredded Wheat   16.0       0
64      Shredded Wheat 'n'Bran   19.0       0
65     Shredded Wheat spoon size   20.0       0
67                   Special K   16.0       3
68      Strawberry Fruit Wheats   15.0       5
69            Total Corn Flakes   21.0       3
71            Total Whole Grain   16.0       3
72                     Triples   21.0       3
74                  Wheat Chex   17.0       3
75                    Wheaties   17.0       3
```

```python
low_carbs_low_sugar = df[(df['carbo'] < df['carbo'].mean()) & (df['sugars'] < df['sugars'].mean())]
print(low_carbs_low_sugar[['name', 'carbo', 'sugars']])
```

```
                        name  carbo  sugars
0                  100% Bran    5.0       6
2                   All-Bran    7.0       5
3  All-Bran with Extra Fiber    8.0       0
9                Bran Flakes   13.0       5
13                  Clusters   13.0       7
19         Cracklin' Oat Bran   10.0       7
26        Frosted Mini-Wheats   14.0       7
34         Great Grains Pecan   13.0       4
41                      Life   12.0       6
54               Puffed Rice   13.0       0
55              Puffed Wheat   10.0       0
56         Quaker Oat Squares   14.0       6
```