Open in Colab

After clicking the "Open in Colab" link, copy the notebook to your own Google Drive before getting started, or it will not save your work

BYU CS 180 Lab 5: Cereal Data

Introduction:

Everyone loves cereal. But have you ever thought deeply about your cereal? Well now is your chance to take a data driven view of your breakfast.

Getting Started:

Download the data from github. Run the code below to download the data that you'll be using in this lab.

You may use pandas, numpy, matplotlib and/or seaborn for these excercises.

You can use/read their respective documentation in the links below (only if you need too, it's not required for the lab):

- Seaborn Documentation
- Matplotlib Documentation
- Numpy Documentation
- Pandas Documentation

In [36]:
```
import pandas as pd
import seaborn as sns
cereal = pd.read_csv('https://raw.githubusercontent.com/porterjenkins/cs180-intro-d
```

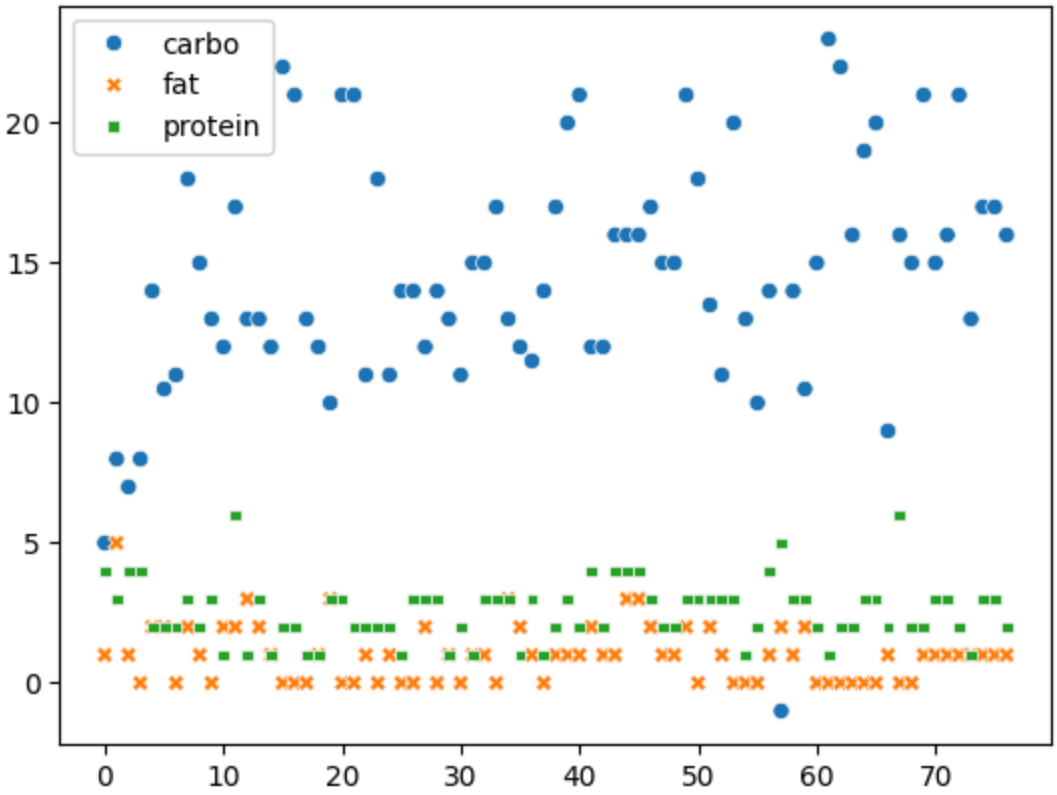
Exercise 1: Protein Powder

Carbs, fats and proteins are the three primary macro nutrients. Create a figure plotting the distribution of each of these macro nutrients together (i.e., three distributions on a single plot). Make sure to provide a legend.

In [37]:
```
# Enter all of your code for exercise 1 here. Feel free to add more cells if you ne

macroCereal = cereal[['carbo','fat','protein']]
sns.scatterplot(data=macroCereal)
```

Out[37]: <Axes: >



I chose to go with a scatterplot because I was not sure how we were supposed to differentiate this data. In this graph the x-axis is the instance of cereal (which type of cereal it is) and the y-axis is the amount of the nutrient specified in the legend. There were no units of measurement in the dataset for the different nutrients so I assume that they were all measured with the same amount of units.

Exercise 2: Sugar Daddy

Get a list of the top 5 most sugary cereals and the 5 least sugary cereals.

In [38]:
```
# Enter all of your code for exercise 2 here. Feel free to add more cells if you ne
sugarSortCereal = cereal.sort_values(by='sugars', ascending=False)
sugarSortCereal = sugarSortCereal[['name', 'sugars']]
print(sugarSortCereal.iloc[:5])
print(sugarSortCereal.iloc[-5:])
```

```
                    name  sugars
30           Golden Crisp      15
66                 Smacks      15
52  Post Nat. Raisin Bran      14
70      Total Raisin Bran      14
6             Apple Jacks      14
                        name  sugars
55              Puffed Wheat       0
63            Shredded Wheat       0
65  Shredded Wheat spoon size       0
64     Shredded Wheat 'n'Bran       0
57             Quaker Oatmeal      -1
```

I am keeping Quaker Oatmeal because I think it is funny that the data resulted in a negative number. If I were cleaning this data for an actual graph I would set the Quaker Oatmeal sugar value to 0 because if Quaker Oatmeal were able to suck sugar out of us it would be much more popular for very different reasons.

Exercise 3: Cereal Killer

Get a list of the top 5 highest rated and lowest rated cereals.

In [39]:
```
# Enter all of your code for exercise 3 here. Feel free to add more cells if you ne
ratingSortCereal = cereal.sort_values(by='rating', ascending=False)
ratingSortCereal = ratingSortCereal[['name', 'rating']]
print(ratingSortCereal.iloc[:5])
print(ratingSortCereal.iloc[-5:])
```

```
                        name     rating
3   All-Bran with Extra Fiber  93.704912
64     Shredded Wheat 'n'Bran  74.472949
65  Shredded Wheat spoon size  72.801787
0                   100% Bran  68.402973
63             Shredded Wheat  68.235885
                     name     rating
14            Cocoa Puffs  22.736446
18          Count Chocula  22.396513
35       Honey Graham Ohs  21.871292
12  Cinnamon Toast Crunch  19.823573
10            Cap'n'Crunch  18.042851
```

Alright who is doing the ratings? If this is a health rating then I understand, but I would rate these in a very opposite order based on tastiness!

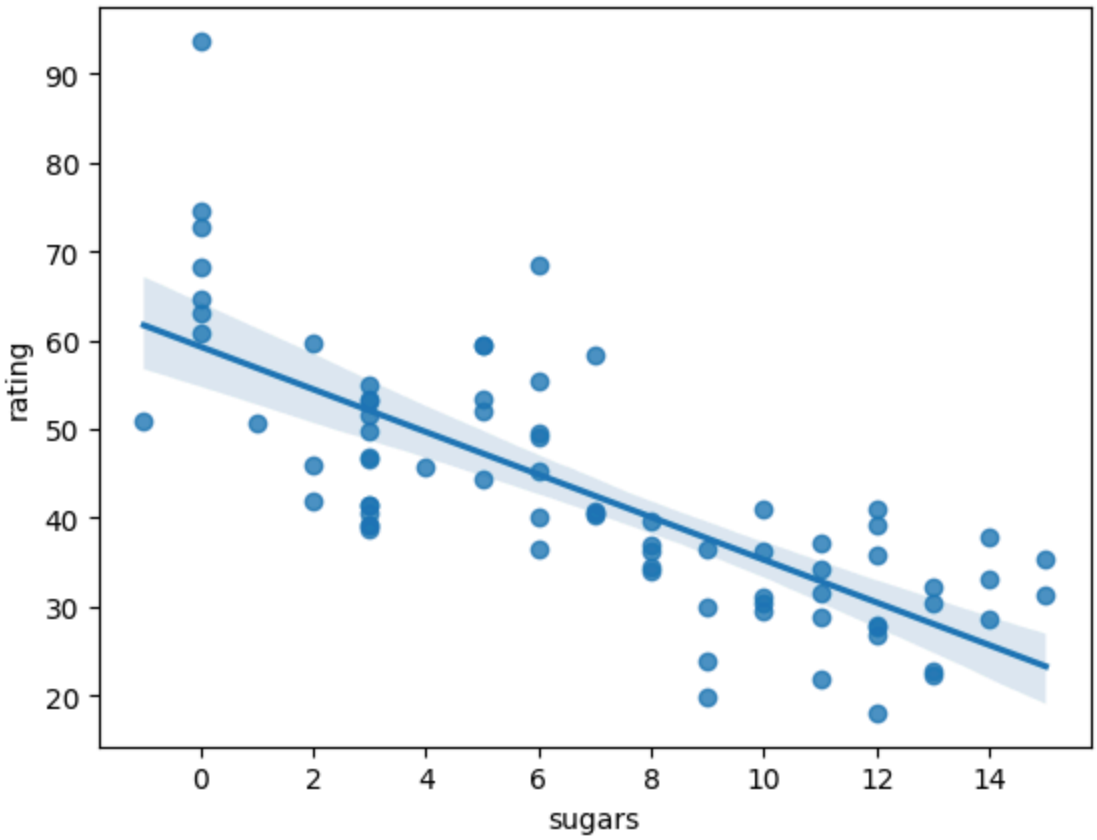
Exercise 4: America

Quantify the relationship between sugar and ratings.

Make a plot to visualize this relationship. Superimpose a best fit line (with seaborn) to describe the relationship. It may be helpful to look at the seaborn regplot documentation.

In [40]:
```
# Make the plot for the data visualization and line of best fit here:
sns.regplot(data=cereal, x='sugars', y='rating')
```

Out[40]:
```
<Axes: xlabel='sugars', ylabel='rating'>
```



Calculate a correlation statistic describing the relationship between sugar and ratings (i.e., r or r squared).

In [41]:
```
# Calculate the statistic using this cell:
print(cereal[['sugars', 'rating']].corr())
```

```
            sugars    rating
sugars    1.000000 -0.759675
rating   -0.759675  1.000000
```

Write a statement in plain English interpreting this statistic.

(Write your statement here)

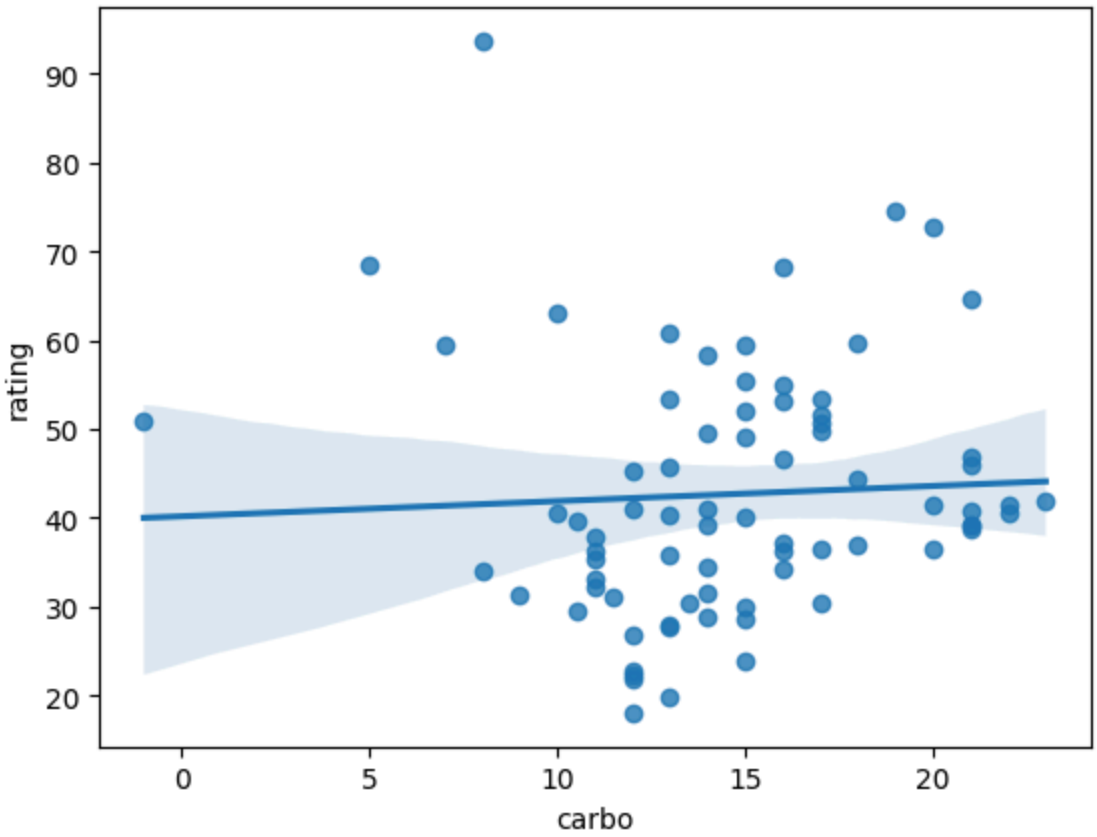
Sugar and rating have a strong (yet negative) correlation. This means that knowing the amount of sugar gives you a 75% chance of knowing what the rating is.

Exercise 5: America Part 2

Make five plots comparing the relationships of carbo, sugars, calories, protein, and fat with rating.

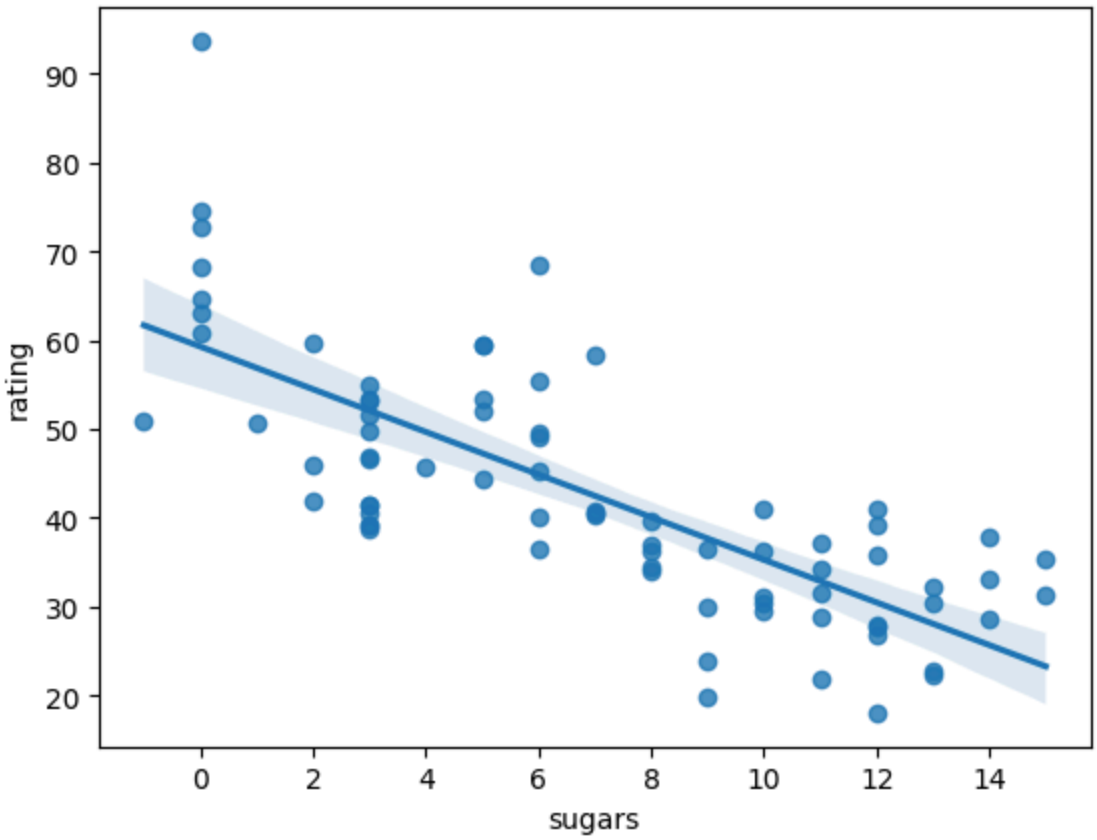
In [42]:
```
# Write your code to compare the various variables with rating below:
print(sns.regplot(data=cereal, x='carbo', y='rating'))
print(cereal[['carbo', 'rating']].corr())
```

```
Axes(0.125,0.11;0.775x0.77)
           carbo    rating
carbo   1.000000  0.052055
rating  0.052055  1.000000
```



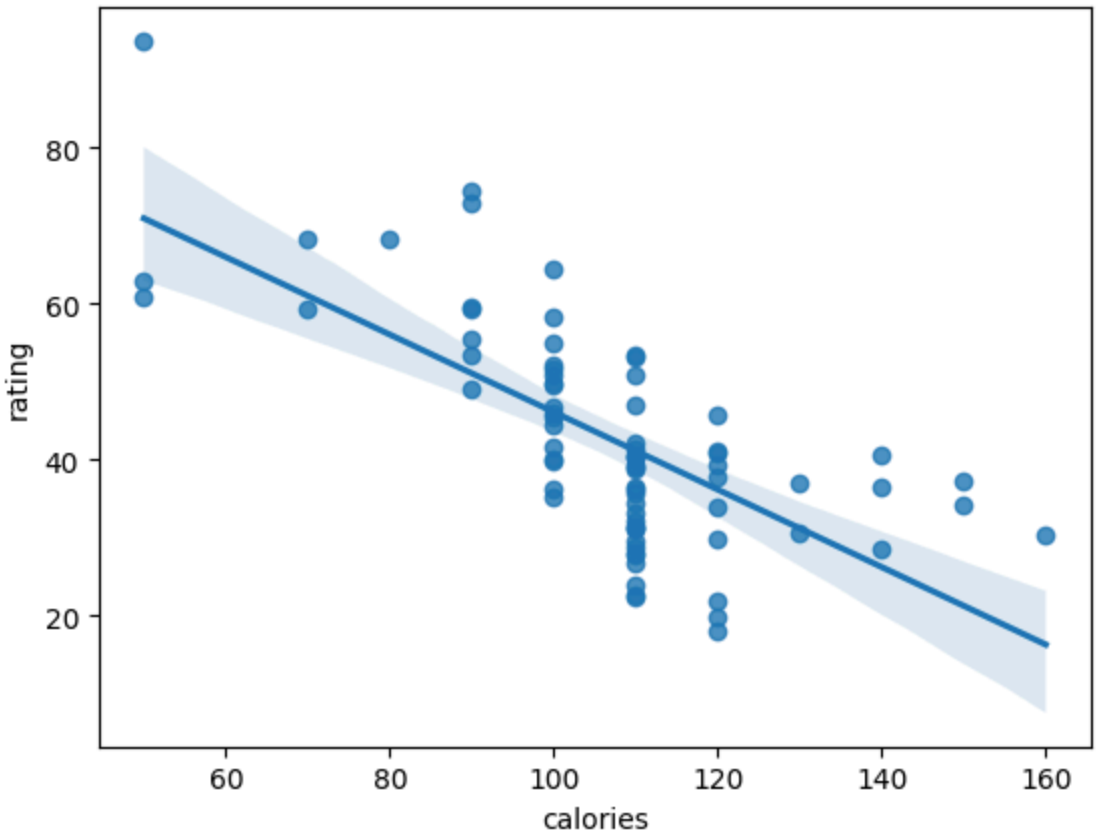
In [43]:
```
sns.regplot(data=cereal, x='sugars', y='rating')
print(cereal[['sugars', 'rating']].corr())
```

```
           sugars    rating
sugars   1.000000 -0.759675
rating  -0.759675  1.000000
```



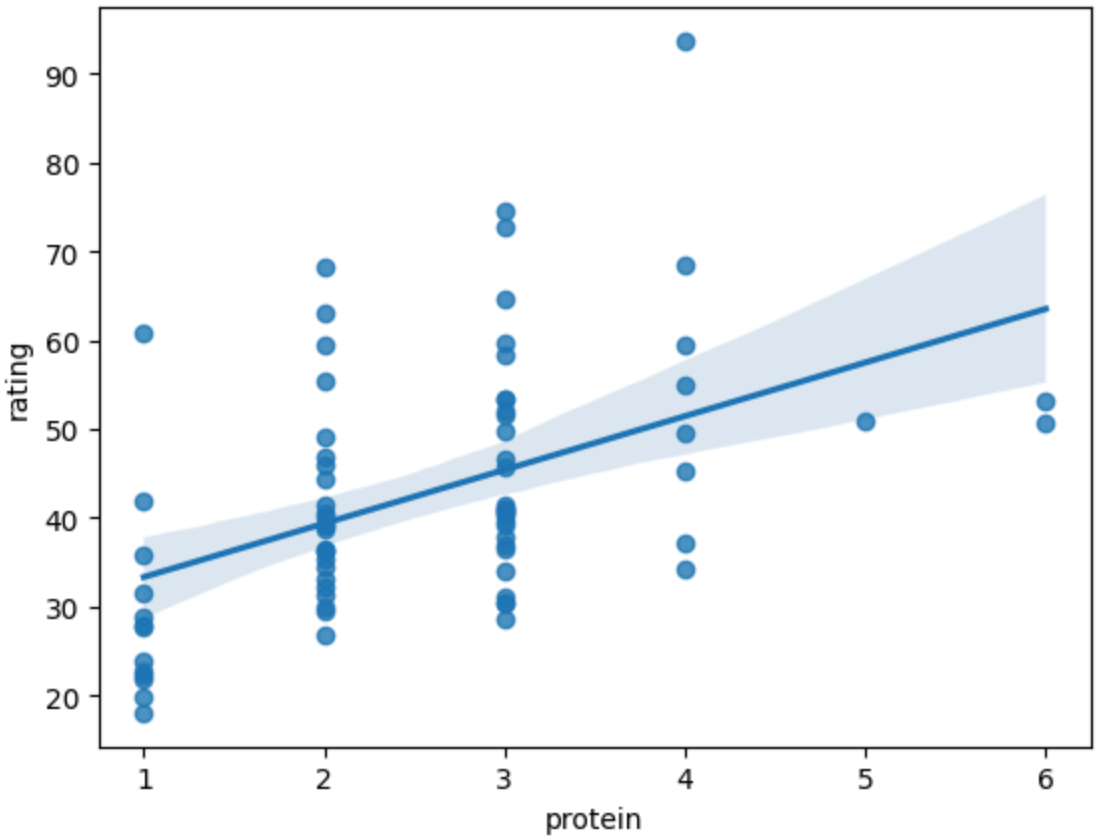
In [44]:
```
sns.regplot(data=cereal, x='calories', y='rating')
print(cereal[['calories', 'rating']].corr())
```

```
            calories    rating
calories    1.000000 -0.689376
rating     -0.689376  1.000000
```



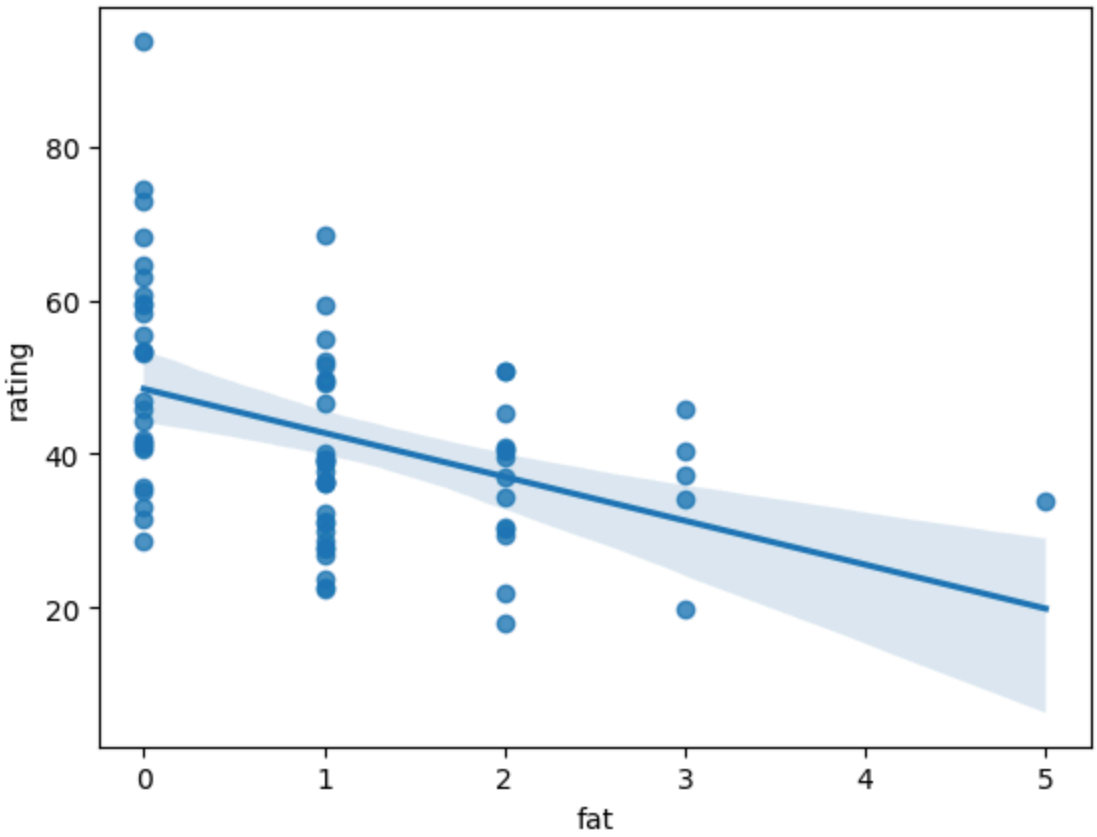
In [45]:
```
sns.regplot(data=cereal, x='protein', y='rating')
print(cereal[['protein', 'rating']].corr())
```

```
           protein    rating
protein   1.000000  0.470618
rating    0.470618  1.000000
```



In [46]:
```
sns.regplot(data=cereal, x='fat', y='rating')
print(cereal[['fat', 'rating']].corr())
```

```
             fat    rating
fat     1.000000 -0.409284
rating -0.409284  1.000000
```



Of the variables carbo, sugars, calories, protein, and fat, which has the strongest relationship with rating? Justify your answer.

(Write your statement here)

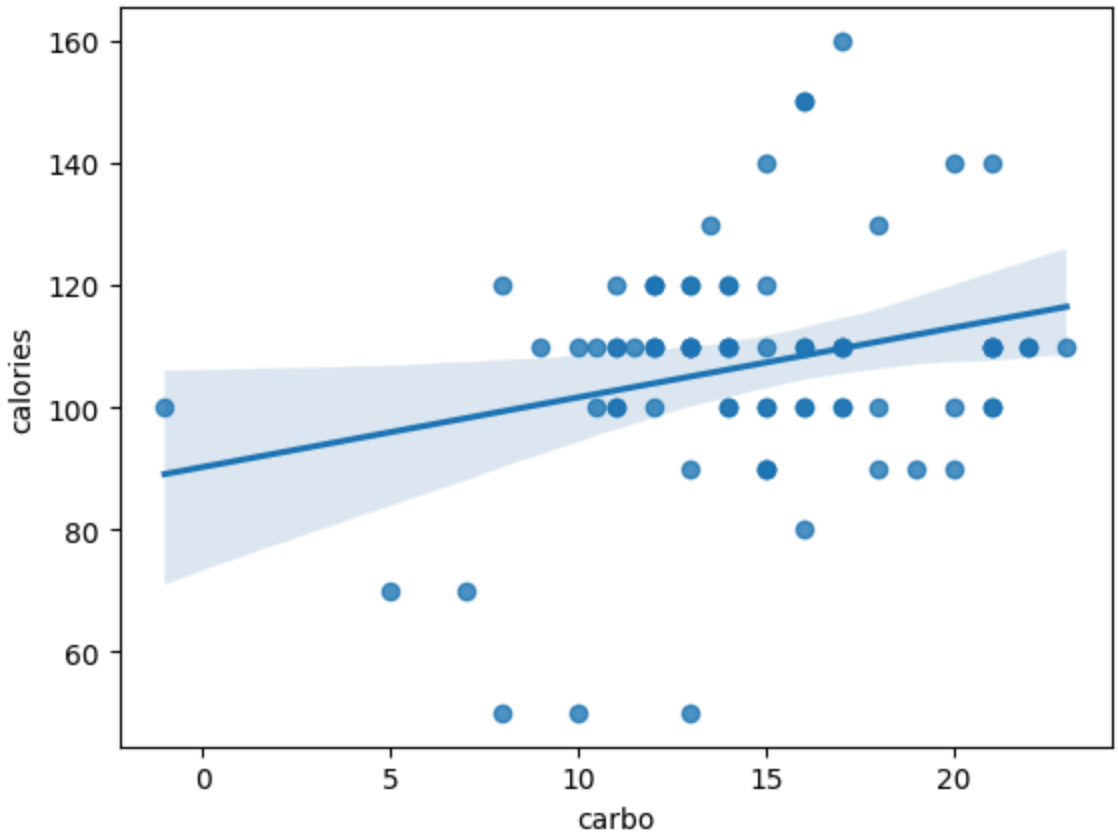
I would answer that the strongest relationship depends on your definition. If you are defining strongest relationship to mean most positive then protein has the strongest relationship with rating. If you are defining strongest relationship as largest slope value (positive or negative), then sugar has the largest (absolute value) correlation coefficient.

Exercise 6: Preparing for Mt. Everest

Do the same as you did with exercise 5, but instead compare carbo, sugars, protein, and fat with calories.

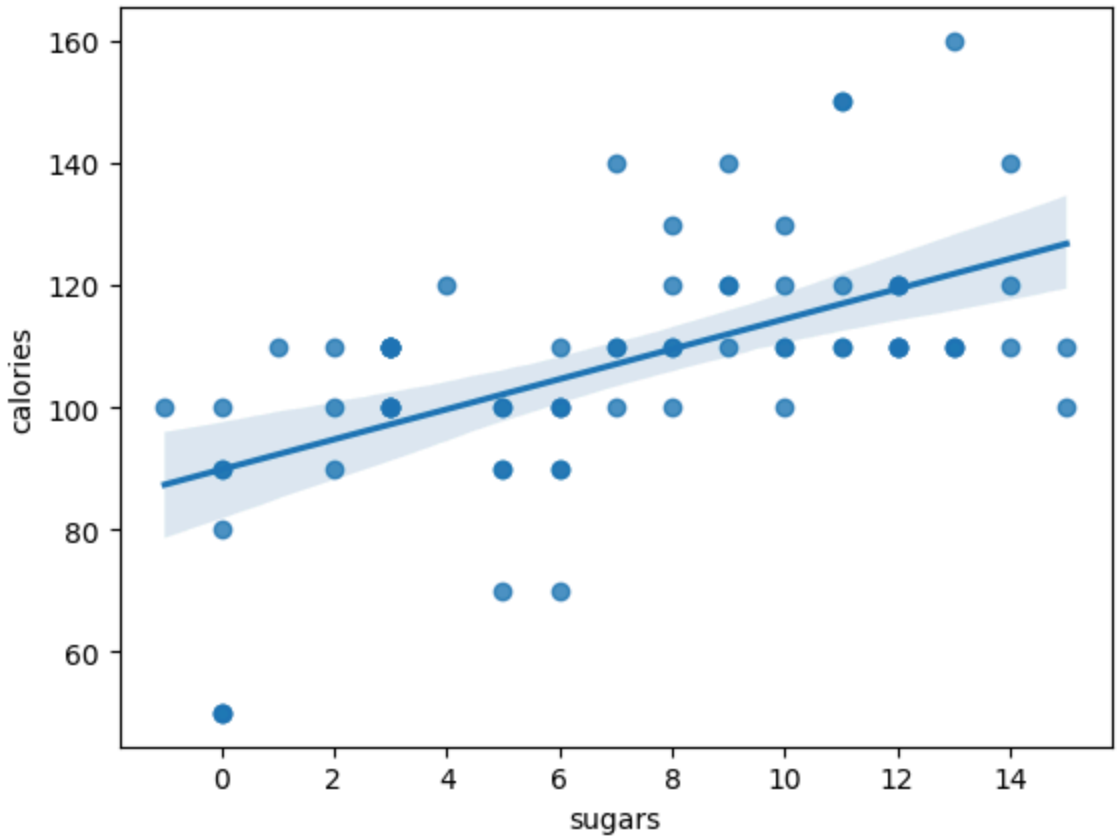
In [47]:
```
# Write your code to compare the various variables with calories below:
sns.regplot(data=cereal, x='carbo', y='calories')
print(cereal[['carbo', 'calories']].corr())
```

```
             carbo  calories
carbo     1.000000  0.250681
calories  0.250681  1.000000
```



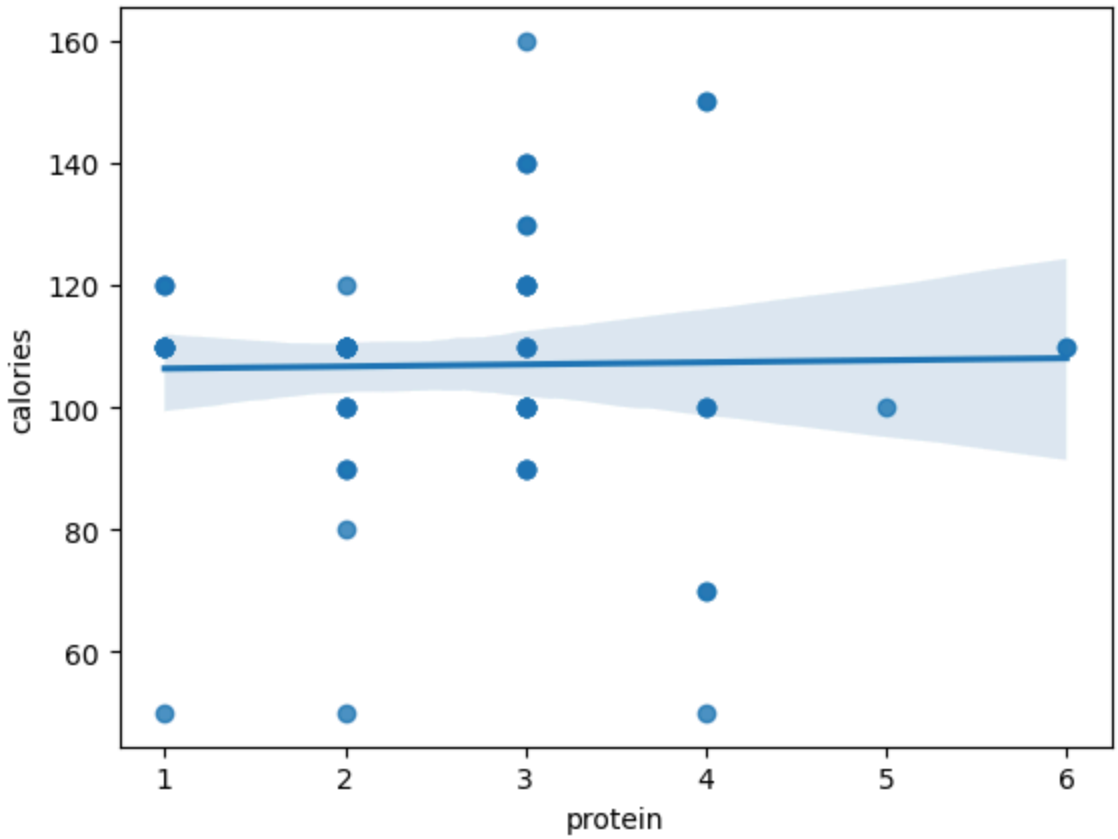
In [48]:
```
sns.regplot(data=cereal, x='sugars', y='calories')
print(cereal[['sugars', 'calories']].corr())
```

```
           sugars  calories
sugars    1.00000   0.56234
calories  0.56234   1.00000
```



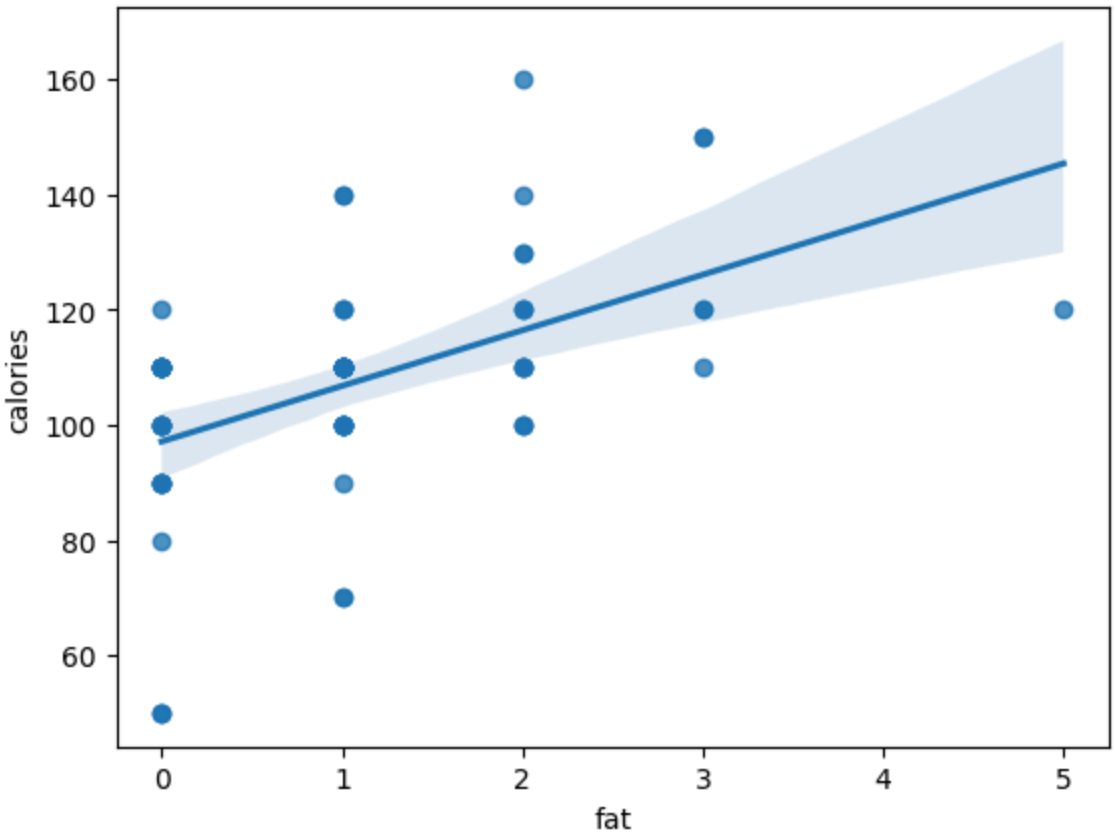
In [49]:
```
sns.regplot(data=cereal, x='protein', y='calories')
print(cereal[['protein', 'calories']].corr())
```

```
            protein  calories
protein    1.000000  0.019066
calories   0.019066  1.000000
```



In [50]:
```
sns.regplot(data=cereal, x='fat', y='calories')
print(cereal[['fat', 'calories']].corr())
```

```
                fat  calories
fat         1.00000   0.49861
calories    0.49861   1.00000
```



Of the variables carbo, sugars, protein, and fat, which has the strongest relationship with calories? Justify your answer.

(Write your statement here)

Sugar and calories has the strongest relationship because it has the largest correlation coefficient. It also makes sense because calories is a measure of energy and human bodies are really good at turn sugar into energy.

Exercise 7: It's Hot and It's Cold

The type column has two values: H='hot' and C='cold'. What is the average rating of each type?

In [51]:
```
# Write your code to get the average rating by type of cereal below:
tempCereal = cereal[['type', 'rating']]
print(tempCereal.groupby('type').mean())
```

```
        rating
type
C    42.095218
H    56.737708
```

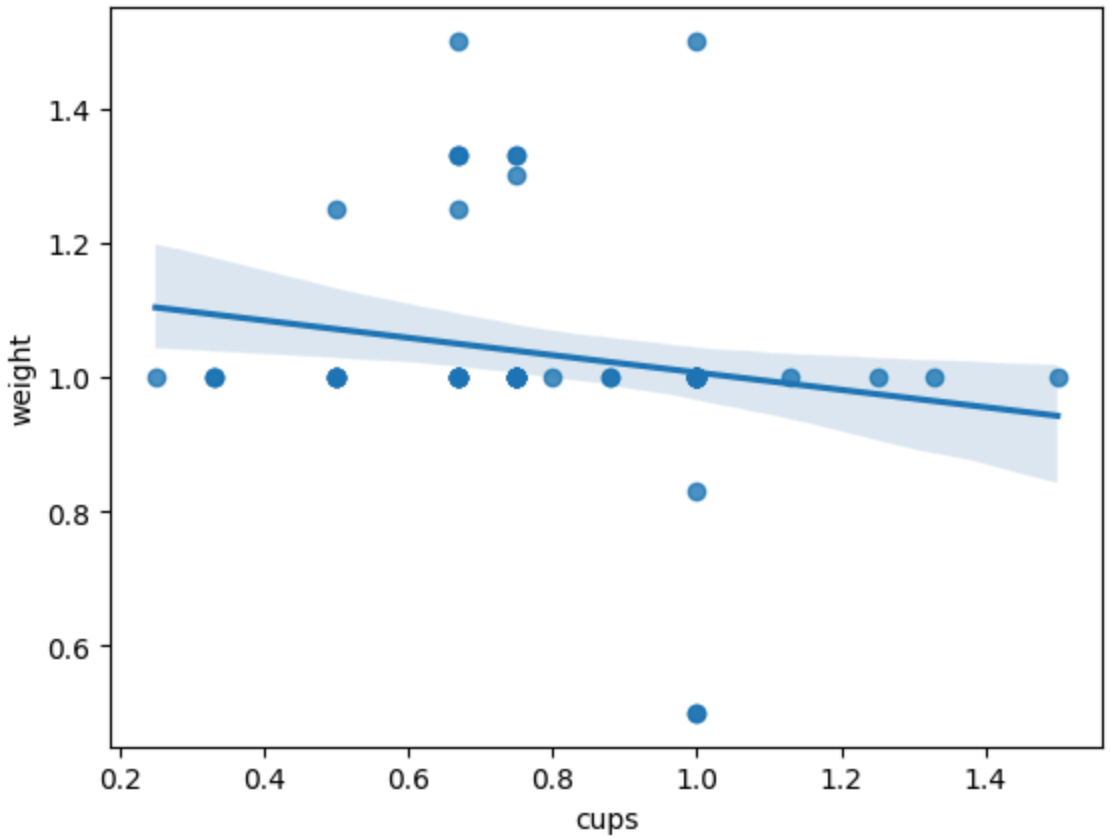
I guess I am just really unhealthy because I really like the stuff that is rated low. I prefer cold cereal to hot cereal, and I prefer sugar to protein.

Exercise 8: Captain Crunch the Numbers

Provide one additional insight from this dataset that you found interesting. Create at least one figure and explain why the figure was interesting to you.

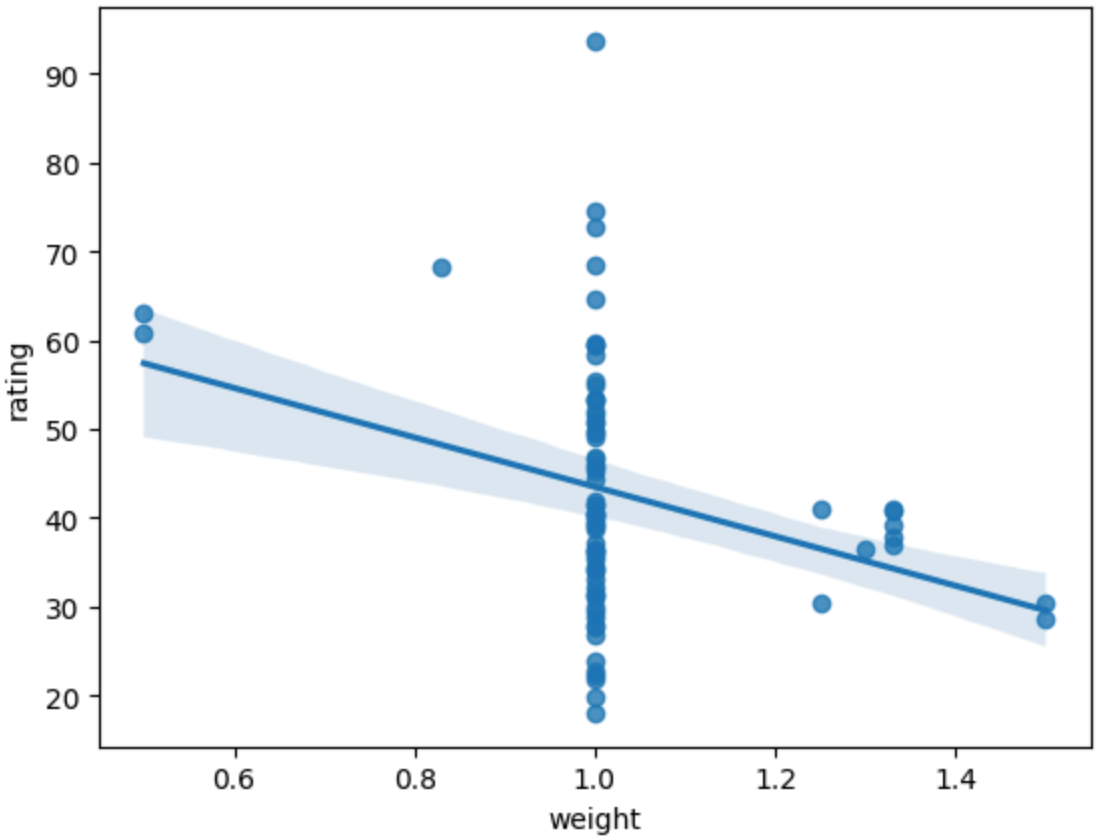
In [52]:
```
# Create the extra plot below:
sns.regplot(data=cereal, x='cups', y='weight')
print(cereal[['cups', 'weight']].corr())
```

```
           cups    weight
cups    1.000000 -0.199583
weight -0.199583  1.000000
```



In [53]:
```
sns.regplot(data=cereal, x='weight', y='rating')
print(cereal[['weight', 'rating']].corr())
```

```
          weight    rating
weight  1.000000 -0.298124
rating -0.298124  1.000000
```



(Write why it was interesting here)

I just thought this was interesting because I would have initially theorized that more cups would have increased the weight. Unless I am misunderstanding what cups and weights measures. Also the rating does not care about the weight of the cereal which means that how dense it is does not affect the final rating.