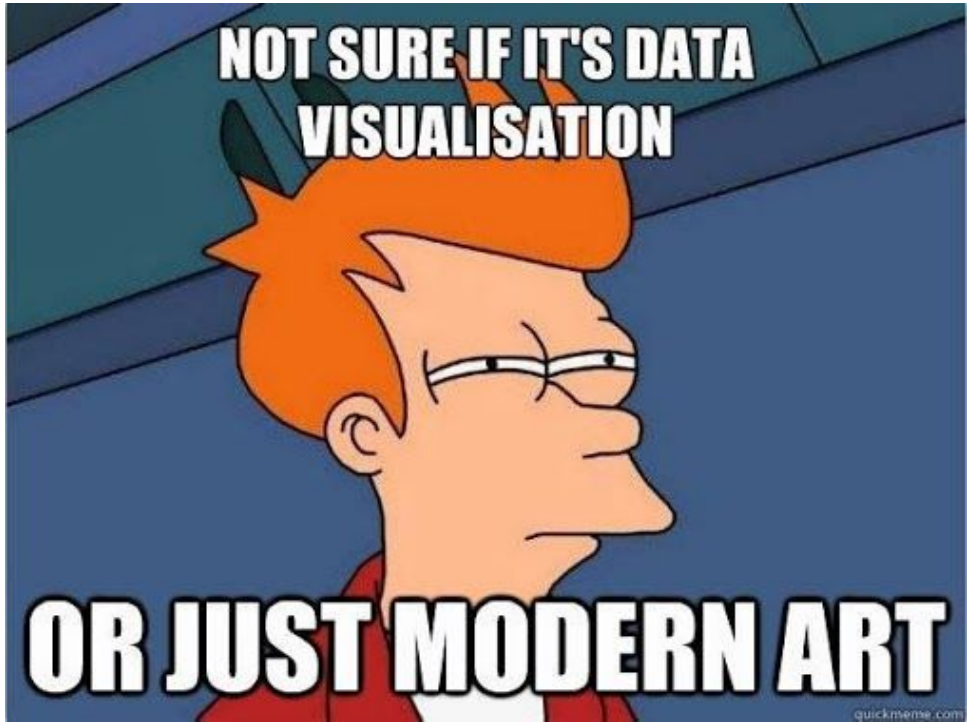

NOT SURE IF IT'S DATA VISUALISATION
OR JUST MODERN ART
quickmeme.com

Source

Welcome to Week 3 Lecture 2!

Data Science in Python & Machine Learning

Week 3 CORE Assignments

These must be submitted by <u>Sunday February 13th</u>:

- **Belt exam takes priority, though!!!**

1) Average Height Exercise (Core)
2) Histograms & Boxplots (Core)
3) Project 1 - Part 3 (Core)

🥋 Stack 1 Belt Exam Eligibility 🥋

- **Belt exam is this weekend!**

- Eligibility:
  - 90 % of Week 1 & 2 Assignments (Including resubmits)
    - **No more than 1 missing/pending resubmissions**
    - Due: 9 AM PST on Thursday (02/10/22)
  - 80% attendance (no more than 1 missed lecture)

- Unlock Codes:
  - **Sent via email after class tonight**
  - Note: once unlocked, you have **up to 24 hours to complete** exam
  - You **may wait until Sunday** morning to start exam
    - but **need to submit it by 11:59 PM PST Sunday** (so <24 hours)

Belt Exam Rules and Policies

- You can **unlock the exam any time after you receive the unlock code provided in your email**. Once you unlock the exam, you will have **24 hours** to submit your work. So make sure not to unlock the exam until you are ready to start working.
  - **Recommend doing at least 2 of Week 3's assignments BEFORE unlocking exam!**

- The belt exam must be submitted by end of day **Sunday on 02/13/21.**

- You **may** use any materials on the exam (notes, classwork, Google, Stackoverflow, etc.) BUT YOU **MUST CITE YOUR SOURCES**!!! Include a comment or a text cell with a link to any sources used.
  - Example:
    - "# Code adapted from: https://stackoverflow.com/questions/19966018/pandas-filling-missing-values-by-mean-in-each-group"

- You must complete the belt exam e**ntirely on your own.**

- You may **NOT** work with anyone else on this exam. **Any** collaboration is cause for expulsion.

- Do **NOT post anything related to the belt exam on Github or anywhere on the internet.** This will be interpreted as collaboration and is cause for expulsion.

Belt Exam Rules and Policies

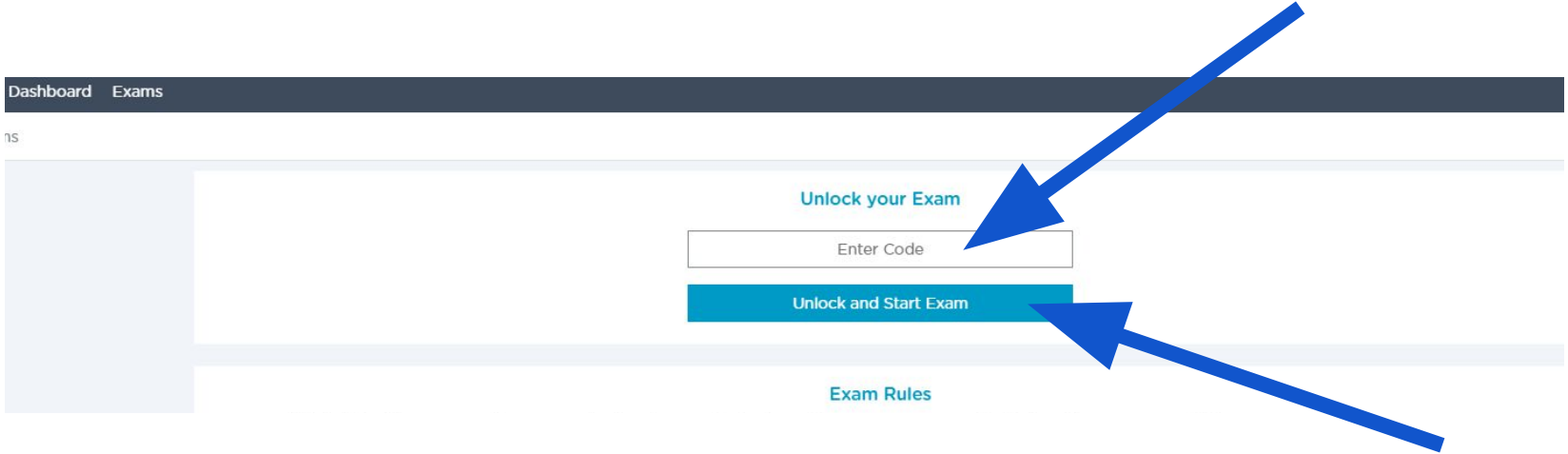
- I will check **email** on Friday in case you have issues on the platform (not content or coding questions). I will NOT check Discord. (jirving@codingdojo.com)
- NOTE: NO exam codes will be sent from Friday 02/11/22 through Sunday 02/13/22.
- You **can contact a TA through direct message** on Discord if you have **process** questions! (TAs can help with your week 3 assignments, just not the belt exam)
- **Do NOT contact any other students** regarding the belt exam, even if they are questions about instructions or policies.
- **Do NOT post questions or comments about the belt exam in our Discord Channel**
- If something goes wrong on the platform when trying to submit, don't stress, just email me your completed materials as a backup within the 24 hour time period (jirving@codingdojo.com).

Accessing the Belt Exam:

1) To take your Belt exam, go to https://login.codingdojo.com/exams.

2) Enter in your code:(To be emailed to you)

3) Then select “Unlock and Start Exam.”

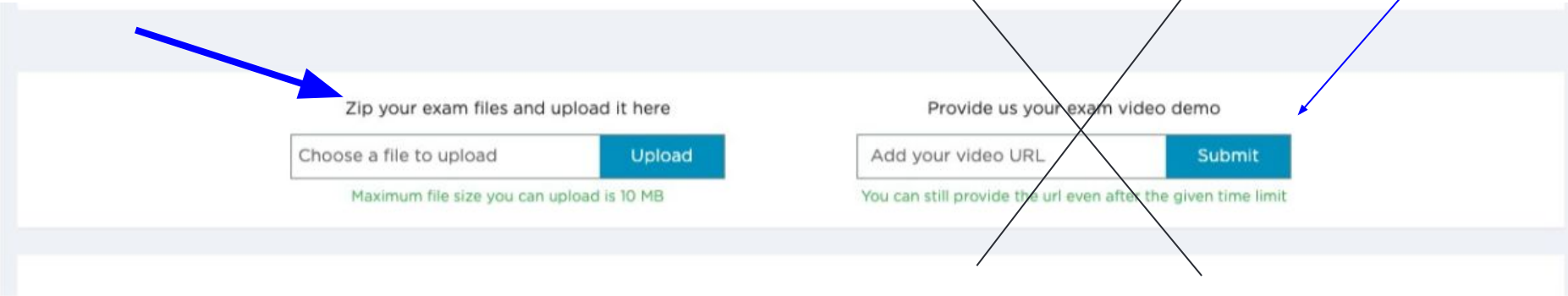


Belt Exam:

- Your first belt exam consists of **one** part:
  1) Submitting your commented code file

- *Optional* Starter Notebook with Checklist rubric!
  - In our notes repo (https://github.com/sensei-jirving/Online-DS-PT-01.24.22-cohort-notes)
    - StarterNotebooks folder > [Belt_Exam_1_Starter_Notebook_01-24-22.ipynb]

No video!

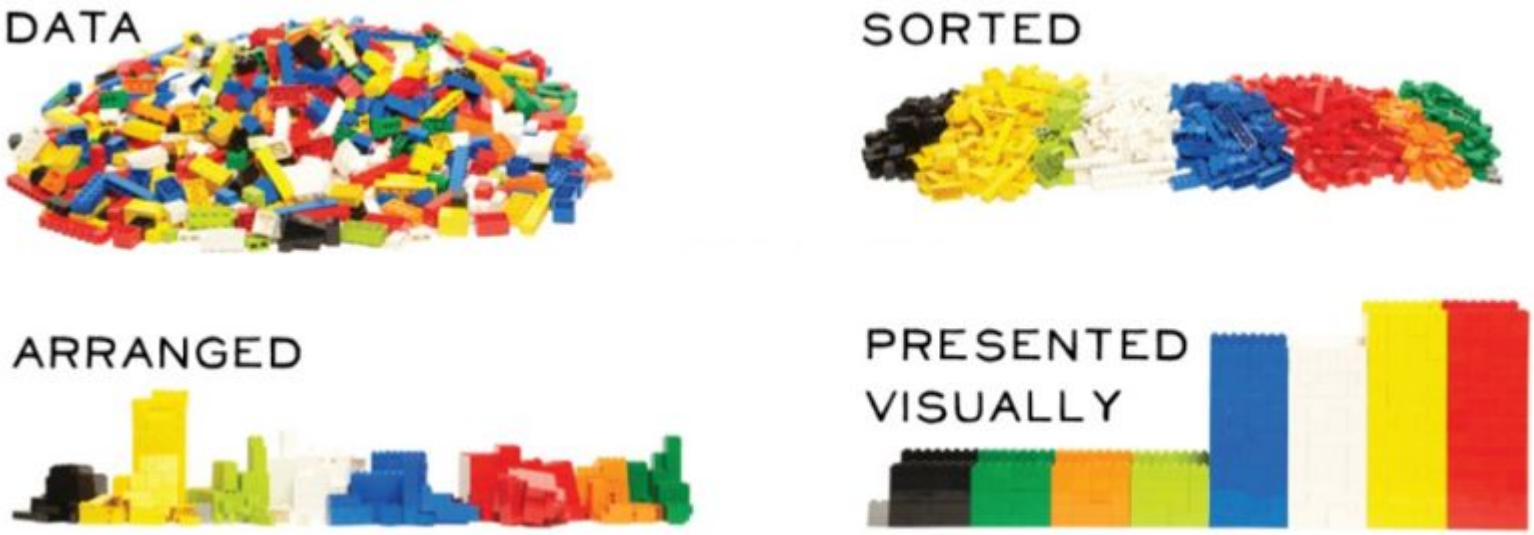


- Please be aware that every data set and problem is different.

- The sample solutions for this mock belt exam cannot be applied in the same way to other data sets/problems.

- **Attempting to copy/paste this code or any other code into your belt exam without understanding what you are doing and why you are doing it will lead to bad outcomes!**

Exploratory Data Analysis: Understanding your Data



Source

Overview of The EDA Process (Suggestion)

1) **Initial Exploration:** Get a feel for your data set
   - What are the columns?
   - What is the type of data for each column
   - How much data do you have?
   - Are you missing data points?
2) **Clean:** Prepare the data for analysis
   - Make sure you have the appropriate data type (are numerical values actually numerical data types?)
   - Make sure your categorical data is consistent
   - Address missing values
   - Address any other obvious errors in your data (Example: Age can't be 230!)
3) **Univariate Visuals:** Explore the distribution of each column of data
   - Histograms and boxplots
   - Bar charts
4) **Multivariate Visuals:** Explore relationships between variables and differences in groups
   - Scatterplots
   - Correlation Heatmaps
   - Multivariable bar charts or boxplots

Go back and repeat any step as needed

Learning Goals

By the end of this lesson you will be able to:

1. Multivariate Data Visualizations
2. Interpret Correlation Plots
3. Identify Correlations with Scatter Plots
4. Create a Three Variable Plot using Seaborn

Warm up:

Why would we do univariate exploratory analysis on our data?

Multivariate Analysis

- Exploring multiple variables at the same time
- We mostly look at just two variables which is also known as <u>bivariate</u> analysis
- Consider relationships or patterns

Correlation: Related Variables

When two or more variables tend to change together:

- When one goes up, the other tends to go up.

OR

- When one goes up the other tends to go down.

Correlation Coefficient (r)

- We can quantify the extent to which two variables are correlated by using a correlation coefficient (r)
- The sign (+ or -) indicates the direction of correlation
- The numerical quantity indicates the extent of correlation
- r can range from -1 to 1
- The closer the value is to 0, the weaker the correlation

-1 — 0 — +1

Strong negative | Weak negative | Weak positive | Strong positive

A correlation of 0.56 is considered moderate

Find the correlation coefficient between the two variables.

```
correlation = cereal['calories per serving'].corr(cereal['grams of sugars'])
correlation
```

```
0.5623402898034883
```

Correlation Values

Low Correlation: 0.3 to 0.5

Moderate Correlation: 0.5 to 0.7

Strong Correlation: Greater than 0.7

Be Careful:
Correlation does NOT mean causation!

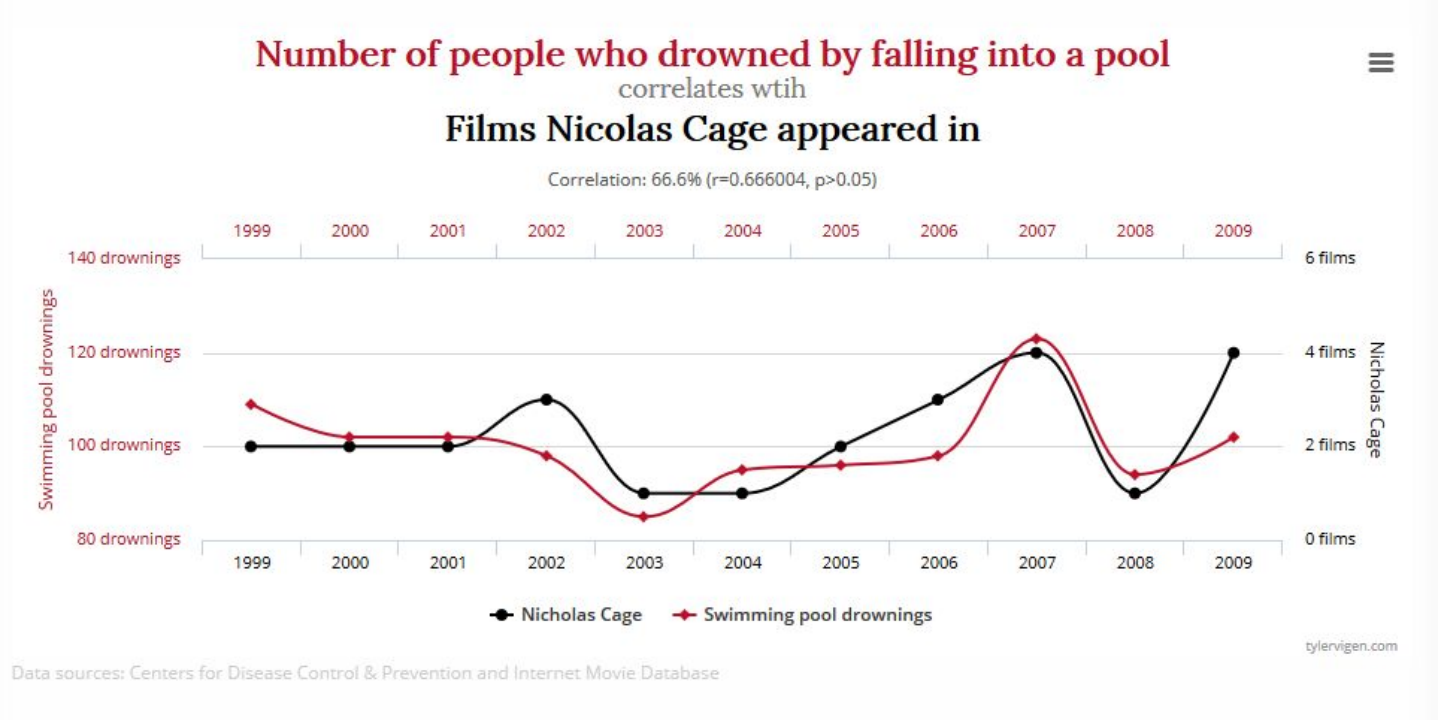


Image source: https://mstranslate.com.au/correlation-vs-causation/

Looking for Correlations

```
[65] cereal.corr()
```

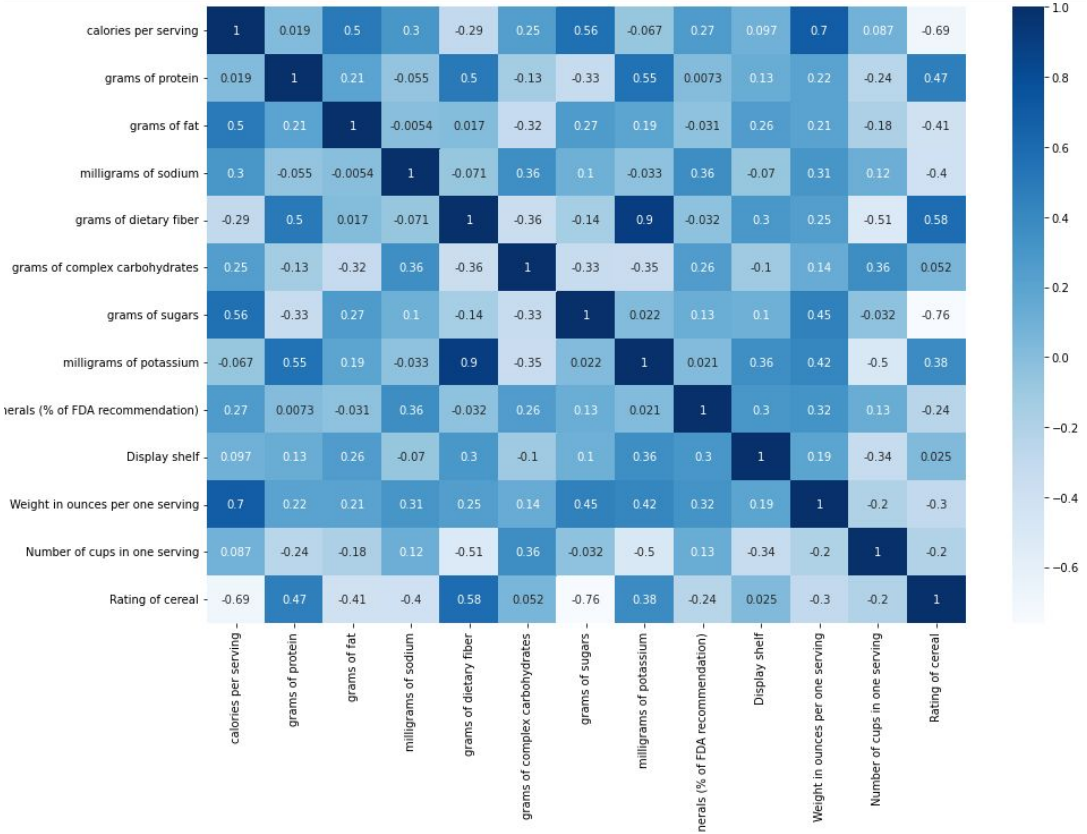
	calories per serving	grams of protein	grams of fat	milligrams of sodium
calories per serving	1.000000	0.019066	0.498610	0.300649
grams of protein	0.019066	1.000000	0.208431	-0.054674
grams of fat	0.498610	0.208431	1.000000	-0.005407
milligrams of sodium	0.300649	-0.054674	-0.005407	1.000000
grams of dietary fiber	-0.293413	0.500330	0.016719	-0.070675
grams of complex carbohydrates	0.250681	-0.130864	-0.318043	0.355983
grams of sugars	0.562340	-0.329142	0.270819	0.101451

Shows r for every combination of numerical value...this is only a section of the output!

Visualizing correlation with a heat map

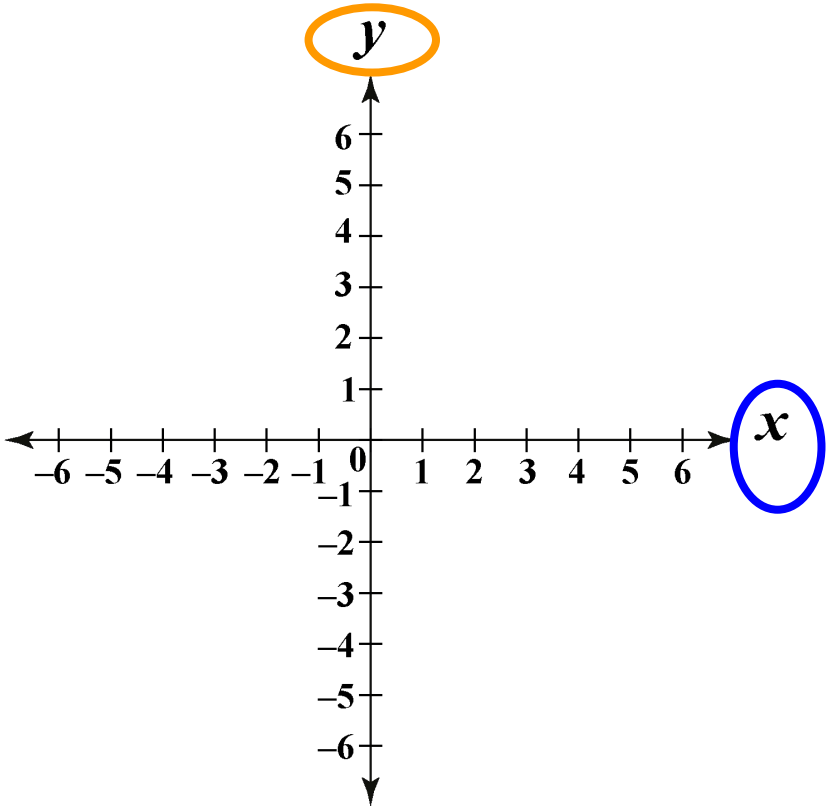
```
import seaborn as sns

corr = cereal.corr()
plt.figure(figsize=(10,10))
sns.heatmap(corr, cmap='Blues', annot=True)
```

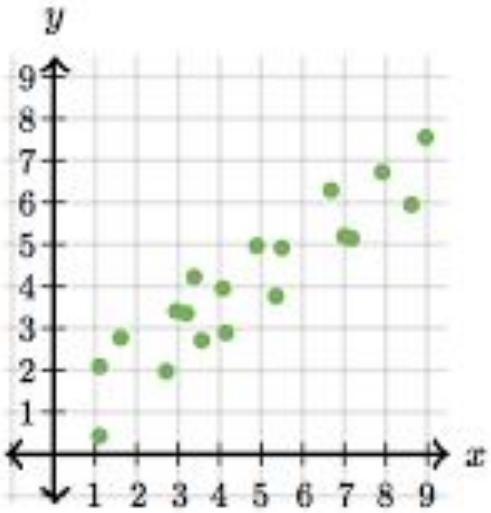


Scatter Plots

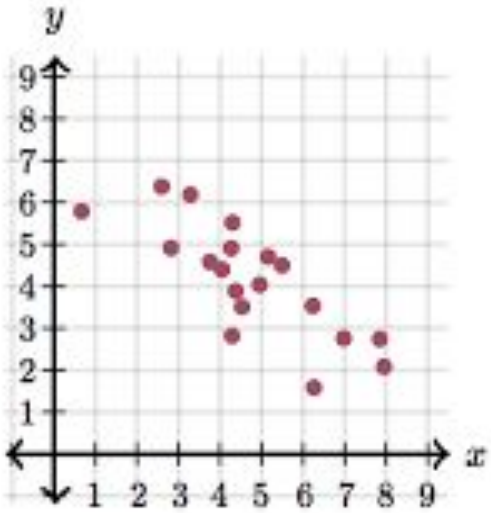
(x, y)



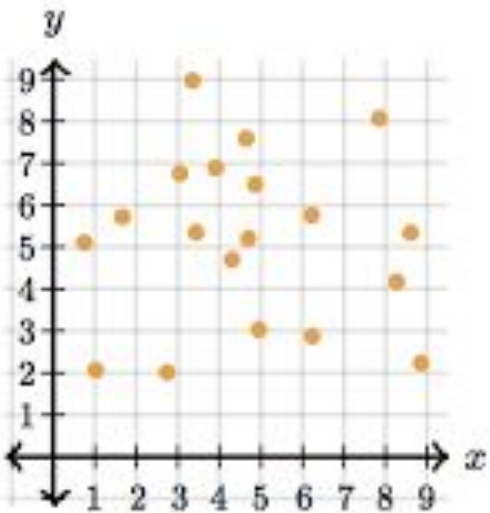
Correlation with Scatter Plots



Positive Correlation



Negative Correlation



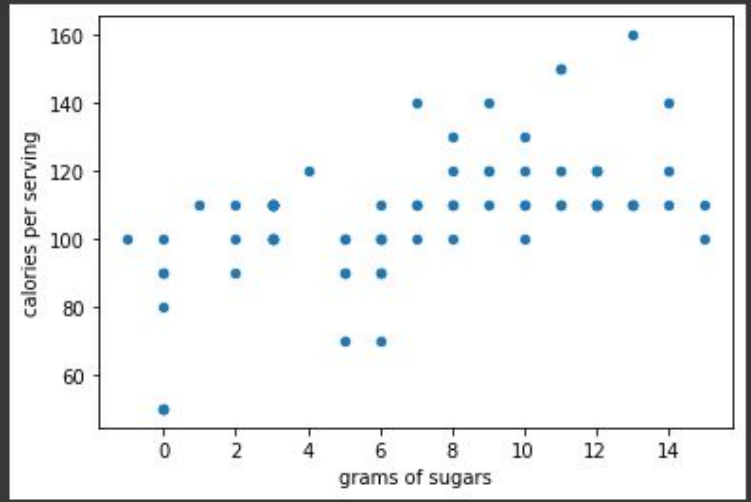
No Correlation

Source

Scatter Plot in Python

Scatterplot of calories per serving vs grams of sugar

```
cereal.plot.scatter(x = 'grams of sugars', y = 'calories per serving');
```



This is a positive correlation.

On average, cereals with more sugar also have more calories and cereals with more calories also have more sugar.

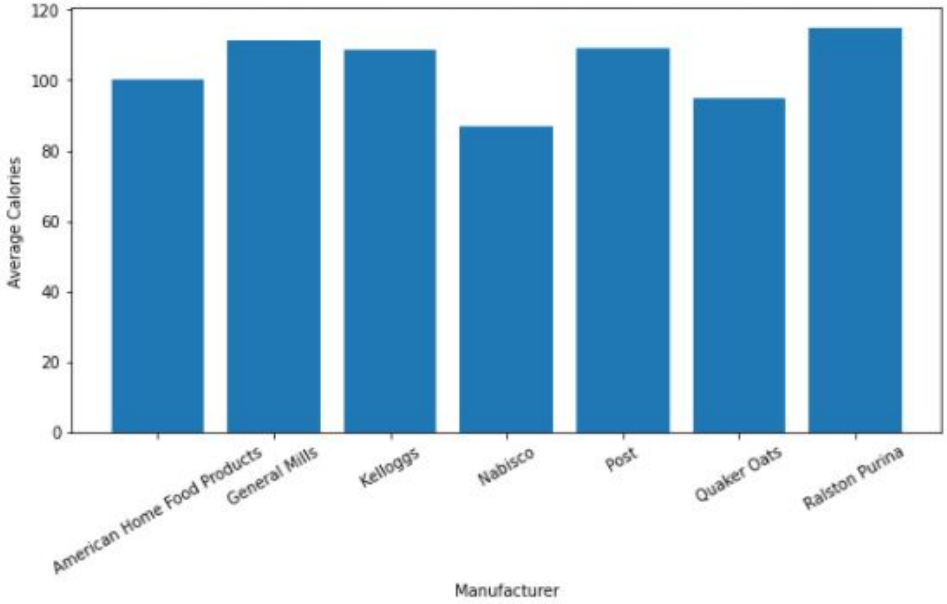
Use Groupby to make bar charts

```
manu_cal = cereal.groupby('Manufacturer')['calories per serving'].mean()
manu_cal
```

```
Manufacturer
American Home Food Products    100.000000
General Mills                  111.363636
Kelloggs                       108.695652
Nabisco                         86.666667
Post                           108.888889
Quaker Oats                     95.000000
Ralston Purina                 115.000000
Name: calories per serving, dtype: float64
```

Vertical Bar Chart

```
plt.bar(manu_cal.index, manu_cal.values)
plt.ylabel('Average Calories')
plt.xlabel('Manufacturer')
plt.xticks(rotation=30)
plt.show()
```

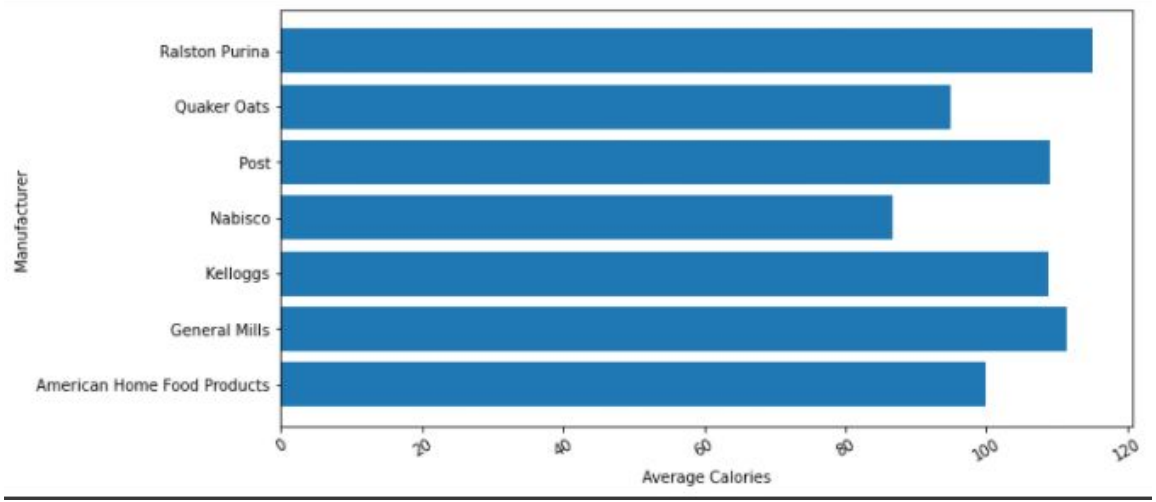


Horizontal bar chart

Notice that plt.barh is used to create a horizontal barchart.

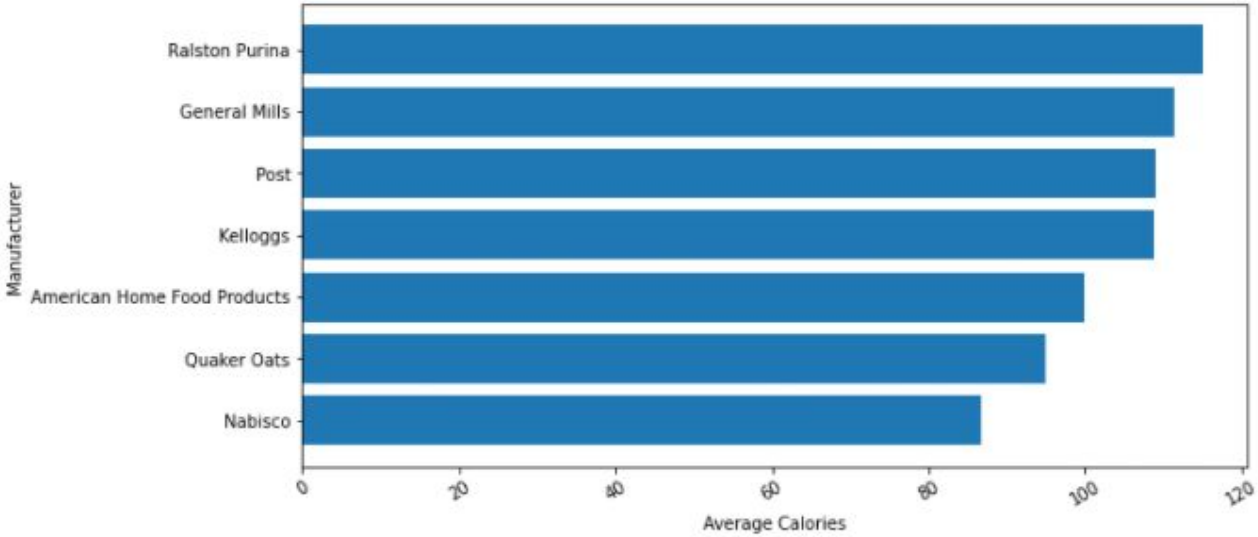
Horizontal bar charts are especially useful when the labels are long

```
plt.barh(manu_cal.index, manu_cal.values)
plt.ylabel('Manufacturer')
plt.xlabel('Average Calories')
plt.xticks(rotation=30)
plt.show()
```



Sorted Vertical Bar Chart

```
manu_cal = manu_cal.sort_values()
plt.figure(figsize=(10,5))
plt.barh(manu_cal.index, manu_cal.values)
plt.ylabel('Manufacturer')
plt.xlabel('Average Calories')
plt.xticks(rotation=30)
plt.show()
```



Seaborn

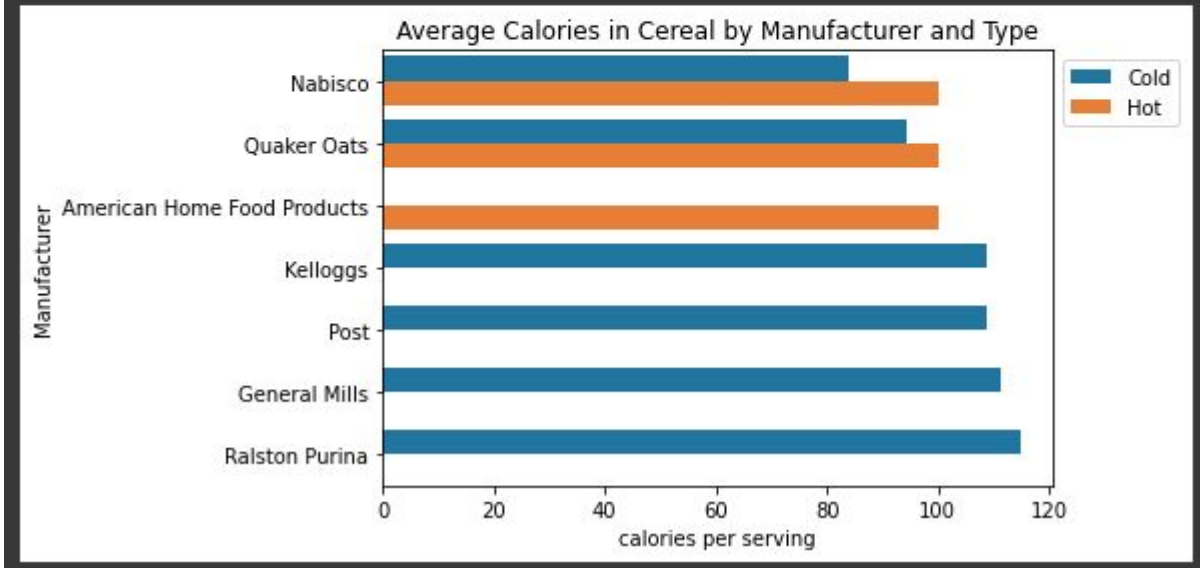
More advanced plots
and trivariate analysis!!

```
import seaborn as sns
```

Comparing 3 variables in a barplot:

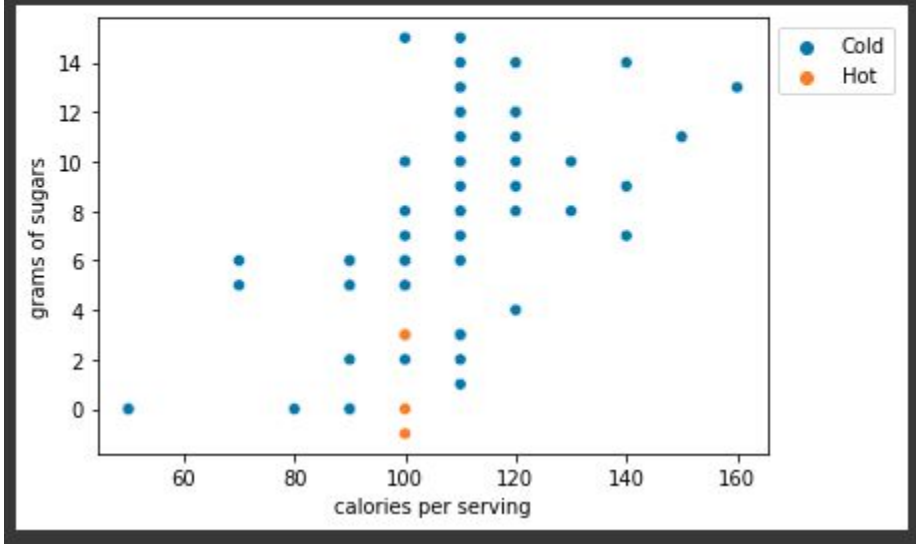
```
manufact = cereal.groupby(['Manufacturer', 'type']).mean().reset_index()
manufact = manufact.sort_values(by='calories per serving')
sns.barplot(data=manufact, x='calories per serving', y='Manufacturer', hue='type')
plt.title('Average Calories in Cereal by Manufacturer and Type')
plt.legend(bbox_to_anchor=(1, 1))
plt.show()
```

3rd Variable



Color Coding Categories in a Scatterplot

```
sns.scatterplot(data=cereal, x='calories per serving', y='grams of sugars',
                hue='type')
plt.legend(bbox_to_anchor=(1,1));
```



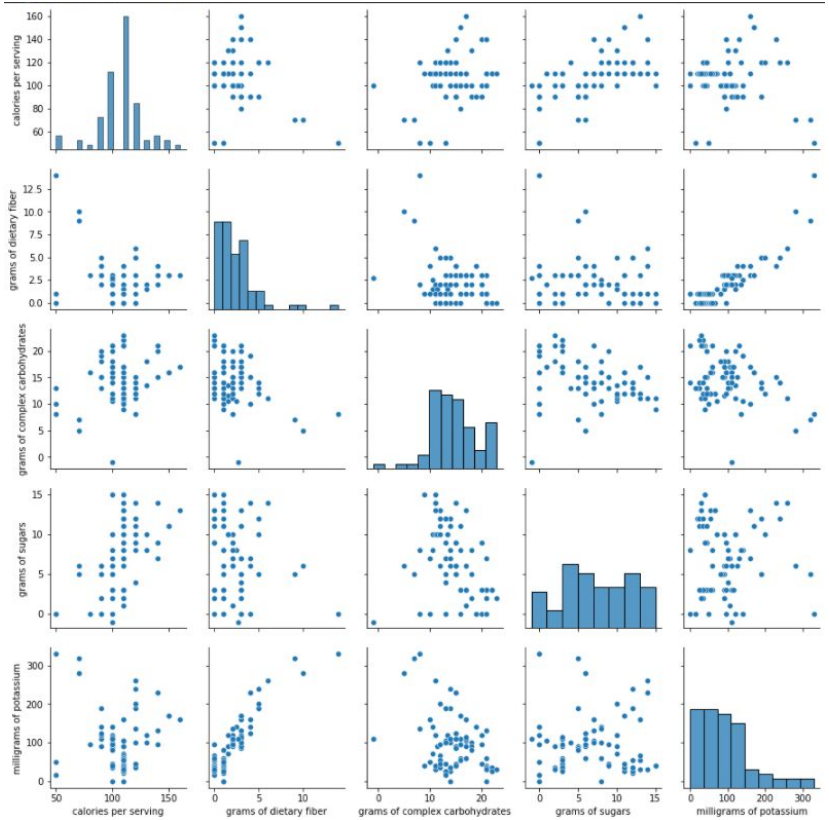
Pair Plot

Distributions

```
import seaborn as sns

sns.pairplot(cereal.iloc[:, [3,7,8,9,10]])

plt.show()
```



Correlations

Visualization Topics Saved for Next Week

- We will be **covering Explanatory Visualizations next week** (visualizations intended to tell a story for a larger/broader audience)
    - We are **saving most of the visualization aesthetics concepts**/questions until then.

- <u>Topics saved for next week:</u>
    - Formatting ticks
    - Matplotlib styles/seaborn themes
    - Font customization (titles/axis labels, etc)
    - Figures with multiple subplots
        - Multiple subplots with DIFFERENT figure sizes.

- In the meantime, I am happy to answer any questions about these in office hours after class today.

Multivariate Codealong!

Notebook

Cleaned Data