Lab Assignment 3

MIT's Karl Ulrich donated a dataset titled Servo Data Set to the UCI Machine Learning Repository in the 1980's. The dataset has been described as "an interesting collection of data that covers an extremely non-linear phenomenon - predicting the rise time of a servomechanism in terms of two (continuous) gain settings and two (discrete) choices of mechanical linkages."

As noted on the dataset website above, the column names are defined in order as:

```
['motor', 'screw', 'pgain', 'vgain', 'class']
```

Your mission, should you choose to accept, is to figure out a few stats about this dataset, which has been conveniently copied to your Module2/Datasets/**servo.data**. You can get started by opening up the assignment starter code, saved to Module2/**assignment3.py**.

Note: Before submitting, double check your work. Peek at the first few entries of your dataset, by opening up servo.data with a text editor. After that, use the appropriate command to look at the first few entries of your dataframe; do they match? If it's not a precise match, there might be a few useful parameters in the read_csv() api documentation that will fix your issue!

Lab Questions

- 3. Exploring Data
- 4. Transforming Data
- 5. Data Modeling

(3/3 points)

Please enter a numeric value (e.g. 0, 1, 10.5, etc) which correctly answers the question(s) below:

How many samples in this dataset have a **vgain** feature value equal to 5?

22 ✔ **Answer:** 22

How many samples in this dataset contain the value **E** for both motor and screw features?

6 ✔ **Answer:** 6

What is the **mean vgain** value of those samples that have a **pgain** feature value equal to 4?

2.060606 ✔ **Answer:** 2.060606

EXPLANATION

Before anything else, take a look at the *names* parameter of read_csv(). It's quite useful.

For the first problem, you need to filter by the vgain column, e.g.: **df = df[df.vgain==5]**. Then you can simply count the number of results, or use **.describe()**, or **len()**.

The second problem is similar, but you have to make use of two filters. You can either do this successively, or combine them and use the bitwise **df = df[(df.motor=='E') & (df.screw=='E')]** to concatenate the separate filters. It's important to encapsulate the two conditionals within

(brackets).

For the last problem, you must first filter by **pgain**, then you can use either **.describe()**, or **.vgain.mean()** to find out the mean value of vgain.

You have used 1 of 2 submissions