Solution Review: Reading Auto MPG Dataset

This lesson provides the solution to the previous challenge.

WE'LL COVER THE FOLLOWING

- Reading the Dataset

Reading the Dataset

```
import pandas as pd # calling pandas module

def read_csv():

    # Define the column names as a list
    names = ["mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "mod
    # Read in the CSV file from the webpage using the defined column names
    df = pd.read_csv("auto-mpg.data", header=None, names=names, delim_whitespace=True)
    return df.shape

# Calling the function and printing the result
print(read_csv())
```

According to the problem statement, we need to get the **shape** of **Auto MPG Dataset** as an output. In the code above, at **line 1**, we imported the `pandas` module for reading the dataset. Next, we implemented the function `read_csv()`.

Look at its header at **line 3**. At **line 6**, we set the name of the *columns* in a list called `names`. We get the name of the columns from the documentation. Notice that `names` hold **9** string values:

- `mpg`
- `cylinders`
- `displacement`

- `horsepower`

- `weight`
- `acceleration`
- `model_year`
- `origin`
- `car_name`

Notice that we replace whitespace with `_` in `model year` and `car name`.

Line 8 is the most important line. We are using a built-in function `read_csv()` from `pandas`. As you are familiar, it takes some arguments as follows:

- `path`: The path to the dataset
- `header`: In case the columns are provided explicitly, it's set to `None`.
- `names`: List of columns' names to be used
- `delim_whitespace`: Specifies whether or not whitespace should be used as a sep

So for `path`, we give the name of the file: **auto-mpg.data**. As we know the columns' names and provided them explicitly to the function as `names=names`, we set `header` as `None`. It was mentioned that there are whitespaces, and we have to use them as sep, so `delim_whitespace` is set to `true`. This built-in function returns a *dataframe* `df` as a result. At **line 9**, we are returning `shape`, a property of the dataframe `df`.

Now, look at **line 12**, where we are calling the function `read_csv()`. It returns a tuple in the form of `(rows, columns)`, which is printed at the end. For this dataset, the function will return `(398, 9)` as there are **398** rows and **9** columns.

That's it about the basics of reading the dataset of different formats using Pandas. The next chapter explains how to describe any dataset in statistical terms.