Solution Review: Cleaning Auto MPG Dataset

This lesson provides the solution to the previous challenge.

WE'LL COVER THE FOLLOWING

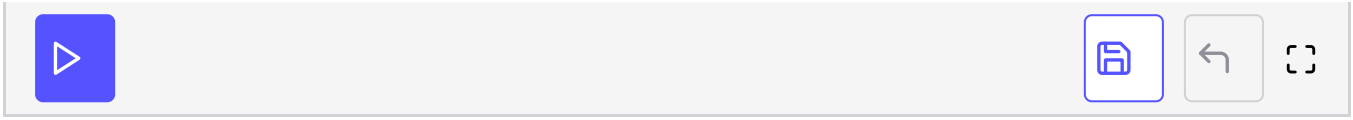
Cleaning the dataset

```
import pandas as pd

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

# Remving outliers from the data
def outlier_detection(df):
    df = df.quantile([.90, .10])
    return df

print(outlier_detection(read_csv()))
```



According to the problem statement, we need to find percentile from the data **Auto MPG Dataset** of all columns. Before doing it, we have to read the data first. There is no need to explain how to read the data, as we studied that in detail previously. Dataset is read from **line 4** to **line 8**.

Moving towards the main implementation, look at the header of the `outlier_detection(df)` function at **line 11**. It takes *one* arguments as input:

- `df`: A dataframe containing the dataset in the form of a matrix.

Line 12 is the most important line. We are using a built-in function

`quantile()` on `df` which takes *one* argument: `[.90, .10]`. We are specifying the range for quartile that it should be between **0.90** and **0.10**. It will return *two* numbers in a list for each column.

At **line 15** we are calling the function `outlier_detection(read_csv())`. First control will transfer to `read_csv()` at **line 3** and we'll get a dataframe. Then control will go to **line 11** and 90^{th} and 10^{th} percentile will be returned for each column.

For the result, you will notice that percentiles are printed. For example, the 90^{th} and 10^{th} percentile for `mpg` is **34.33** and **14.00** respectively.

That's it about the main concepts regarding cleaning the dataset using Pandas. The next chapter explains how to visualize a dataset using the `seaborn` library.