# Case Study Questions

## Question 1: Fuel Economy

Using the following dataset, please answer the following questions. The dataset in question comes from <https://www.fueleconomy.gov/feg/download.shtml>. The particular file is <https://www.fueleconomy.gov/feg/epadata/vehicles.csv.zip> “Datasets for All Model Years (1984 - 2019)”. The data dictionary is here: <https://www.fueleconomy.gov/feg/ws/index.shtml#vehicle>

1. Which manufacturer produces the most fuel efficient fleet of cars?
2. How has fuel economy changed over time? Are there any other interesting insights or trends?
3. Prepare the dataset to be used in a predictive model, in particular, handle any missing data.

## Question 2: Anagram finder

Use anagram\_finder.py file to build an anagram finder. An anagram finder, given a word, finds all words in a dictionary that can be constructed by re-arranging the word's letters. By word, we mean a sequence of characters that does not contain whitespace. The letters are not case sensitive in the input, but the case should be maintained in the output. The dictionary is specified as a string which contains arbitrary number of words.

## Question 3: Time series analysis

Use “question3.xlsx” file to answer the following questions:

1. Given a time series of daily returns for one ticker (see FOR\_PROBLEM\_1\_2), calculate the month to date and year to date returns.  Your result should be a dataframe with date, ticker, daily returns, month to date returns, and year to date returns.
2. If you were given a time series of daily prices instead of daily returns, what would your time series look like?  Adjust your output from (1) to include the daily price as well. Use the INITIAL\_PRICES data to help you.
3. Redo (1) and (2) for a time series of multiple tickers. Use the data in FOR\_PROBLEM\_3\_4.
4. Suppose a portfolio P is created using some linear combination of the tickers’ returns from (2).  Assume that each ticker is equally weighted at the beginning of each month (monthly rebalanced), and that their weights sum to 1. Using the data from FOR\_PROBLEM\_3\_4, get the daily / month to date / year to date returns for P.  Like in (1), your result should be a dataframe with date, daily returns, month to date returns, and year to date returns.