SCE is a power utilities participant in the California deregulated energy market (called the CAISO), meaning that everyday SCE buys and sells energy in the market in order to meet the demand of its customers. Having accurate load forecasts for SCE. Most of the forecasted load is bought 14 hours before the start of the "flow date" in the "day- ahead" energy market. The flow date is defined as the day the energy is consumed by our customers. The rest of the load is bought in the "real time" energy market in order to hit real time fluxes of demand.
The load forecast accuracy is crucial for SCE in order to keep costs down for SCE customers and to be able to balance supply and demand on the grid. SCE aims to predict the Load with a model yielding a mean absolute percentage error (MAPE) of under 4% for our 14-hour forecast in order to accurately purchase gas in the "day-ahead" market.

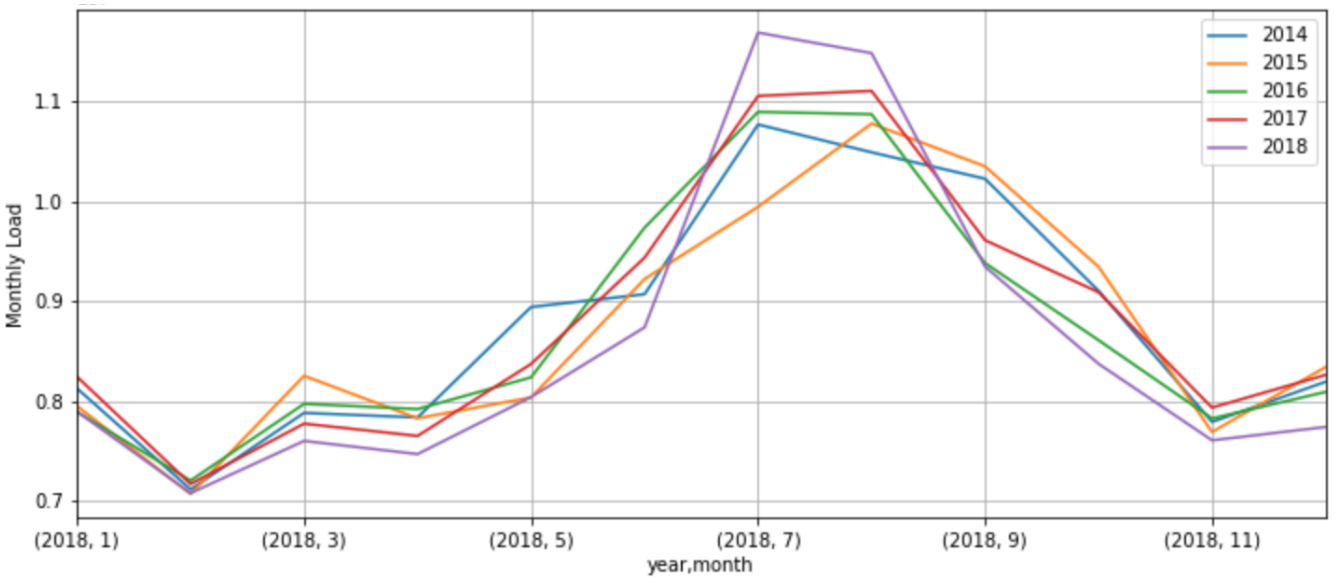
The `sce.xlsx` file contains the load (measured in MWhs) in hour intervals for every day from 2014 to 2019. Consider using the following steps. Run the following function right after opening the libraries (numpy, pandas, and any other of your choice)

```
def mean_absolute_percentage_error(y,y_pred):
    y = np.array(y)
    y_pred = np.array(y_pred)
    return 100*np.mean(np.abs((y-y_pred)/y))
```

Read the datafile into a dataframe `df`.

1. (10 pts.) Use `df['year'] = df['Date'].dt.year` to create a column for the `year`. Repeat this step to create columns for the `month, day,` and `hour`. Also,
   use `df['day of week'] = df['Date'].dt.dayofweek` to create a column for the day of the week.
   Fit a multiple regression model with all predictors (new created columns) and with
   `month, day, hour, day of week` as categorical variables. Report the MAPE and r-square.
2. (10 pts.) Plot of `Load` vs `temp` with the x-axis displaying the years.
3. (20 pts.) Construct a scatterplot of `Load` vs `temp` with the x-axis displaying the `temp`. Add a quadratic least squares (red) line on the plot. This plot should suggest that the square of `temp` is a good predictor of the `Load` and that it should be included in the following models.
4. (10 pts.) Fit a multiple regression model by adding to the first model the temperature squared, the interaction of temperature and hour, and, the interaction of squared temperature and hour. Report the MAPE and r-square.
5. (10 pts.) Use `df['lag24'] = df['Load'].shift(24)` to add the `Load` shifted 24 hours as an additional predictor. Fit the MLR model and report the MAPE and r-square.
6. (10 pts.) Use `sm.graphics.tsa.plot_pacf(load, lags = 60)` to plot partial autocorrelations (adjust the argument `lags = 60` as needed). Lags with a large partial autocorrelations (+ or -) should be good predictors. Fit a MLR model with the additional lags found and report the MAPE and r-square.
7. (20 pts.) Split the data into a test set (2019 load values) and a train set (all other years). Fit the best model found and report the MAPE and r-square for the test set.
8. (10 pts.) Plot the cumulative load by year and month. This is called a *seasonal chart* (see below). It is useful to display the seasonality of the data. Adjust the x-axis with your choice.

Submit your report with your name and USC ID as a pdf file online (no screen captures).


2014
2015
2016
2017
2018
1.1
1.0
0.9
0.8
0.7
Monthly Load
(2018, 1)
(2018, 3)
(2018, 5)
(2018, 7)
(2018, 9)
(2018, 11)
year,month