CARNEGIE MELLON UNIVERSITY
DATA ANALYTICS (COURSE 18-899)
ASSIGNMENT 1

Name: GretaYu

ID: kaiweiy

Date: January.23.2019

Language: Python3

Main code in Jupyter notebook: ./HW1.ipynb

Non-built-in libraries for Python3 codes:

- pandas
- sympy
- datetime
- statsmodels
- numpy
- matplotlib

1. Download historical daily weather data for France. For example, the analysis could be based on the weather in Paris by using:
http://www.wunderground.com/history/airport/LFPO/
Organize the data so that it can be saved as a csv file and load it into your computer. Fill any gaps in the data using linear interpolation.

Drop both 'Events' and 'high Gust Wind (km/h)' column, there are still 3 columns with NaN values, as shown in output 1.

```
Q1
Before filling missing value
<class 'pandas.core.frame.DataFrame'>
Index: 365 entries, 0 to 364
Data columns (total 19 columns):
Date                           365 non-null object
high Temp. (°C)                365 non-null int64
avg Temp. (°C)                 365 non-null int64
low Temp. (°C)                 365 non-null int64
high Dew Point (°C)            365 non-null int64
avg Dew Point (°C)             365 non-null int64
low Dew Point (°C)             365 non-null int64
high Humidity (%)              365 non-null int64
avg Humidity (%)               365 non-null int64
low Humidity (%)               365 non-null int64
high Sea Level Press. (hPa)    365 non-null int64
avg Sea Level Press. (hPa)     365 non-null int64
low Sea Level Press. (hPa)     365 non-null int64
high Visibility (km)           363 non-null float64
avg Visibility (km)            363 non-null float64
low Visibility (km)            363 non-null float64
high Wind (km/h)               365 non-null int64
avg Wind (km/h)                365 non-null int64
sum Precip. (mm)               365 non-null float64
dtypes: float64(4), int64(14), object(1)
memory usage: 57.0+ KB
None
```

output 1

Run linear interpolation with pd.interpolate() to fill all the Gaps. After interpolation, all gaps are filled (output 2).

```
After filling missing value
<class 'pandas.core.frame.DataFrame'>
Index: 365 entries, 0 to 364
Data columns (total 19 columns):
Date                           365 non-null object
high Temp. (°C)                365 non-null int64
avg Temp. (°C)                 365 non-null int64
low Temp. (°C)                 365 non-null int64
high Dew Point (°C)            365 non-null int64
avg Dew Point (°C)             365 non-null int64
low Dew Point (°C)             365 non-null int64
high Humidity (%)              365 non-null int64
avg Humidity (%)               365 non-null int64
low Humidity (%)               365 non-null int64
high Sea Level Press. (hPa)    365 non-null int64
avg Sea Level Press. (hPa)     365 non-null int64
low Sea Level Press. (hPa)     365 non-null int64
high Visibility (km)           365 non-null float64
avg Visibility (km)            365 non-null float64
```

```
low Visibility (km)            365 non-null float64
high Wind (km/h)               365 non-null int64
avg Wind (km/h)                365 non-null int64
sum Precip. (mm)               365 non-null float64
dtypes: float64(4), int64(14), object(1)
memory usage: 57.0+ KB
None
```

output 2

2. Calculate the correlation matrix between all the weather variables. Make a graphic to show the correlation matrix as a heat-map.

corrMatrix

	high Temp. (°C)	avg Temp. (°C)	low Temp. (°C)	high Dew Point (°C)	avg Dew Point (°C)	low Dew Point (°C)	high Humidity (%)	avg Humidity (%)	low Humidity (%)	high Sea Level Press. (hPa)	avg Sea Level Press. (hPa)	low Sea Level Press. (hPa)	high Visibility (km)	avg Visibility (km)	low Visibility (km)	high Wind (km/h)	avg Wind (km/h)	sum Precip. (mm)
high Temp. (°C)	1.0000	0.9767	0.8881	0.8561	0.8581	0.8432	-0.3944	-0.6426	-0.6530	-0.2536	-0.1509	-0.0885	0.4726	0.5834	0.4336	-0.0817	-0.2000	-0.0530
avg Temp. (°C)	0.9767	1.0000	0.9621	0.9134	0.9214	0.9098	-0.3783	-0.5623	-0.5531	-0.3090	-0.2033	-0.1353	0.4707	0.5161	0.3829	-0.0205	-0.1189	-0.0051
low Temp. (°C)	0.8881	0.9621	1.0000	0.9298	0.9447	0.9383	-0.3331	-0.4246	-0.3898	-0.3655	-0.2612	-0.1900	0.4404	0.3962	0.2943	0.0549	-0.0080	0.0552
high Dew Point (°C)	0.8561	0.9134	0.9298	1.0000	0.9786	0.9407	-0.0874	-0.2336	-0.2623	-0.3558	-0.2647	-0.1927	0.4239	0.2843	0.1373	0.0200	-0.0966	0.1363
avg Dew Point (°C)	0.8581	0.9214	0.9447	0.9786	1.0000	0.9796	-0.0949	-0.2146	-0.2312	-0.3293	-0.2291	-0.1584	0.3911	0.2756	0.1410	-0.0286	-0.1210	0.1130
low Dew Point (°C)	0.8432	0.9098	0.9383	0.9407	0.9796	1.0000	-0.1117	-0.2184	-0.2214	-0.3035	-0.1996	-0.1336	0.3709	0.2836	0.1578	-0.0673	-0.1388	0.0781
high Humidity (%)	-0.3944	-0.3783	-0.3331	-0.0874	-0.0949	-0.1117	1.0000	0.7850	0.5649	0.0609	0.0182	0.0054	-0.1972	-0.5377	-0.6506	-0.1048	-0.1623	0.2178
avg Humidity (%)	-0.6426	-0.5623	-0.4246	-0.2336	-0.2146	-0.2184	0.7850	1.0000	0.9131	0.0885	0.0342	0.0143	-0.3939	-0.7512	-0.7051	-0.0356	0.0090	0.2600
low Humidity (%)	-0.6530	-0.5531	-0.3898	-0.2623	-0.2312	-0.2214	0.5649	0.9131	1.0000	0.1252	0.0696	0.0412	-0.4700	-0.7138	-0.6081	-0.0021	0.0819	0.1967
high Sea Level Press. (hPa)	-0.2536	-0.3090	-0.3655	-0.3558	-0.3293	-0.3035	0.0609	0.0885	0.1252	1.0000	0.9646	0.8993	-0.4204	-0.1606	-0.0919	-0.4856	-0.3712	-0.3598
avg Sea Level Press. (hPa)	-0.1509	-0.2033	-0.2612	-0.2647	-0.2291	-0.1996	0.0182	0.0342	0.0696	0.9646	1.0000	0.9738	-0.3934	-0.1256	-0.0608	-0.5658	-0.4431	-0.4114
low Sea Level Press. (hPa)	-0.0885	-0.1353	-0.1900	-0.1927	-0.1584	-0.1336	0.0054	0.0143	0.0412	0.8993	0.9738	1.0000	-0.3723	-0.1168	-0.0640	-0.6071	-0.4846	-0.4213
high Visibility (km)	0.4726	0.4707	0.4404	0.4239	0.3911	0.3709	-0.1972	-0.3939	-0.4700	-0.4204	-0.3934	-0.3723	1.0000	0.6281	0.4140	0.3217	0.2514	0.1216
avg Visibility (km)	0.5834	0.5161	0.3962	0.2843	0.2756	0.2836	-0.5377	-0.7512	-0.7138	-0.1606	-0.1256	-0.1168	0.6281	1.0000	0.7990	0.0545	-0.0161	-0.1779
low Visibility (km)	0.4336	0.3829	0.2943	0.1373	0.1410	0.1578	-0.6506	-0.7051	-0.6081	-0.0919	-0.0608	-0.0640	0.4140	0.7990	1.0000	0.0527	0.0327	-0.2863
high Wind (km/h)	-0.0817	-0.0205	0.0549	0.0200	-0.0286	-0.0673	-0.1048	-0.0356	-0.0021	-0.4856	-0.5658	-0.6071	0.3217	0.0545	0.0527	1.0000	0.8179	0.3761
avg Wind (km/h)	-0.2000	-0.1189	-0.0080	-0.0966	-0.1210	-0.1388	-0.1623	0.0090	0.0819	-0.3712	-0.4431	-0.4846	0.2514	-0.0161	0.0327	0.8179	1.0000	0.2927
sum Precip. (mm)	-0.0530	-0.0051	0.0552	0.1363	0.1130	0.0781	0.2178	0.2600	0.1967	-0.3598	-0.4114	-0.4213	0.1216	-0.1779	-0.2863	0.3761	0.2927	1.0000

table 1. Correlation matrix

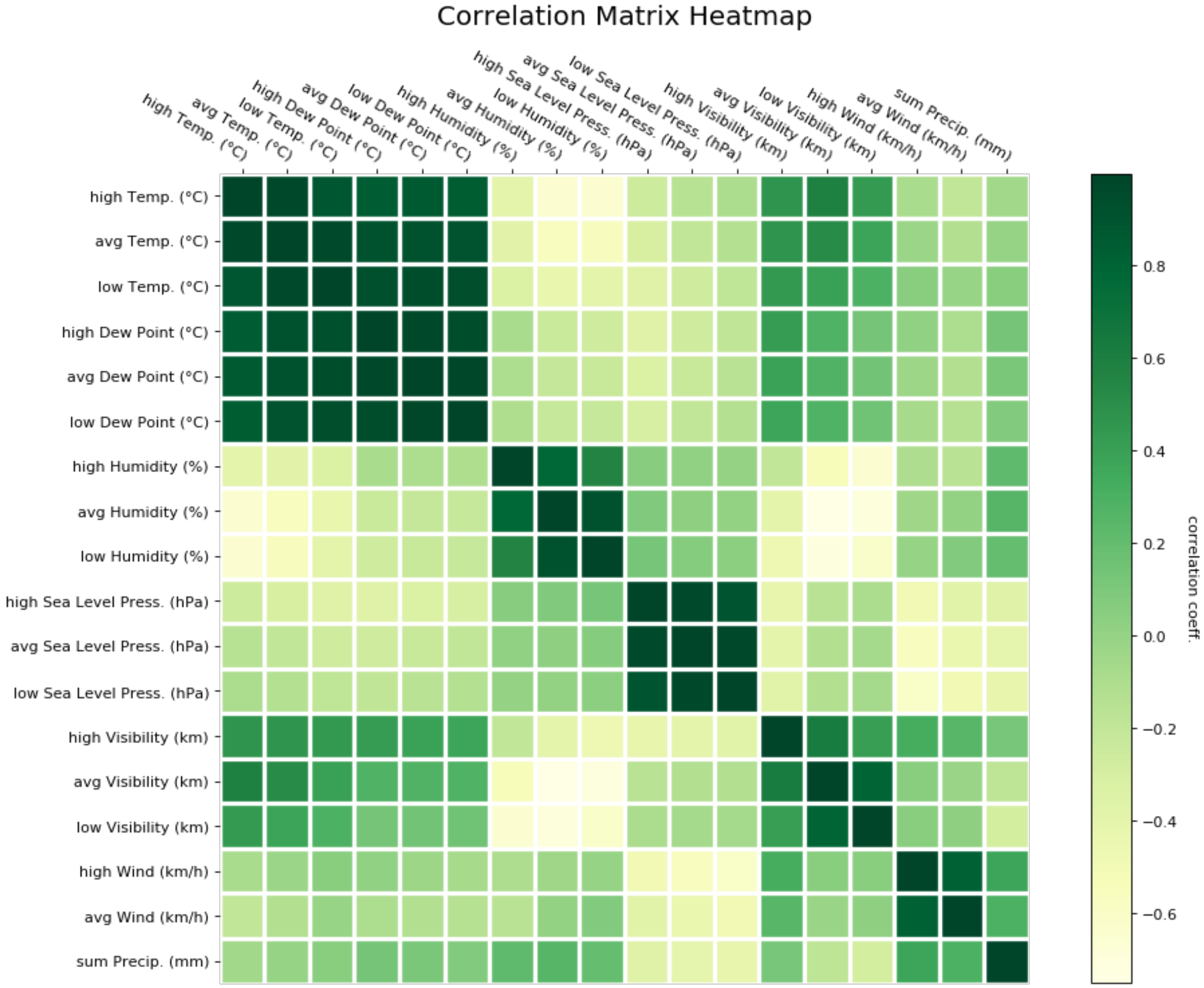


figure 1

3. Download historical daily electricity consumption data for France from: http://clients.rte-france.com/lang/an/visiteurs/vie/vie_stats_conso_jour.jsp Save it as a csv file and load it into your computer.

Daily electricity data is cleaned and saved as "**electricity_consumption.csv**".

4. Synchronize the dates corresponding to both time series and make a scatter plot of energy consumption against mean temperature.

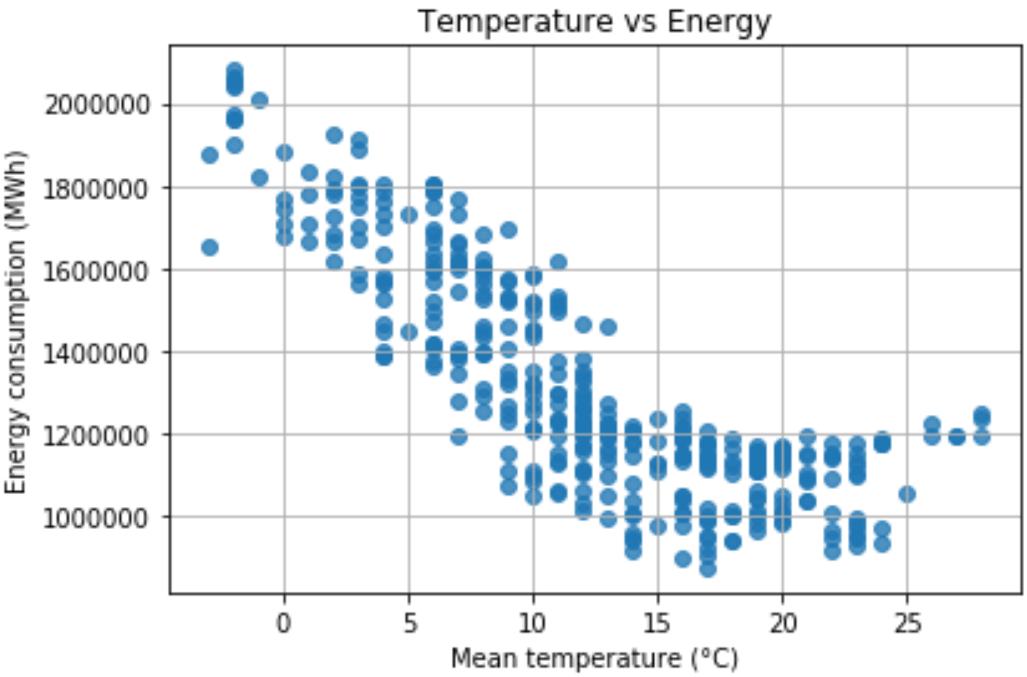


figure 2

5. Fit a quadratic model to the energy versus temperature. Plot the quadratic fit as a line on top of the scatter plot.

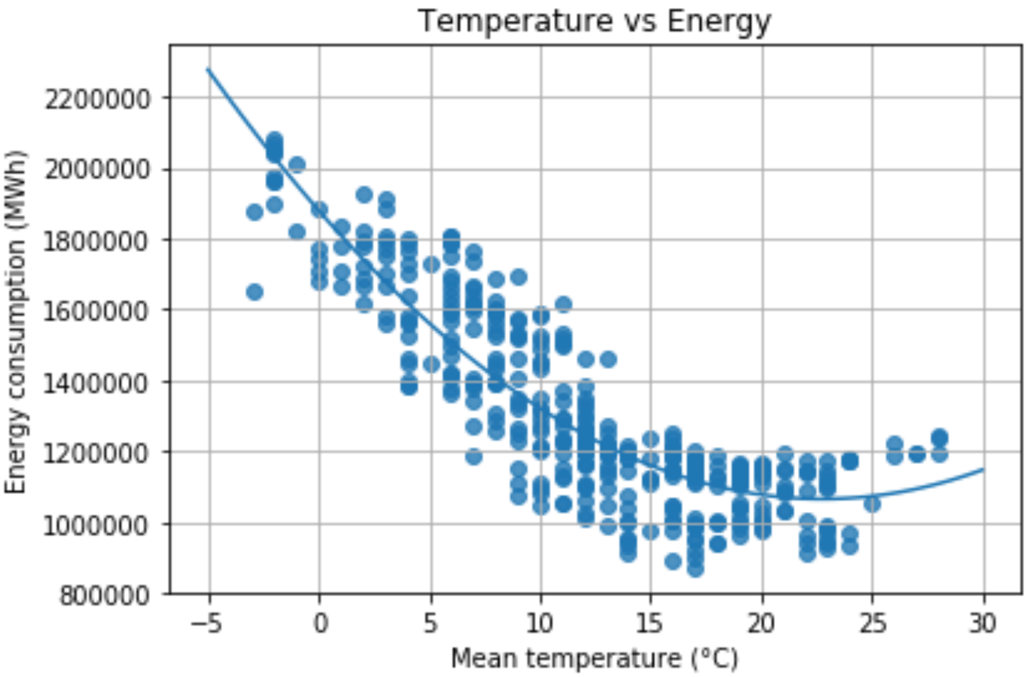


figure 3

Quadratic equation: y = 1.8791e+06 * x^2 - 7.1407e+04 * x + 1.5675e+03

6. Based on the empirical analysis, what is the optimal temperature coinciding with minimal consumption? Use the quadratic fit and verify visually.

Optimal temperature calculated by quadratic fit is 22.77°C. However, minimum consumption in data happens at temperature=17°C.
For the minimum consumption, the optimal temperature calculated by quadratic fit is offset from the real value by 5.7778°C

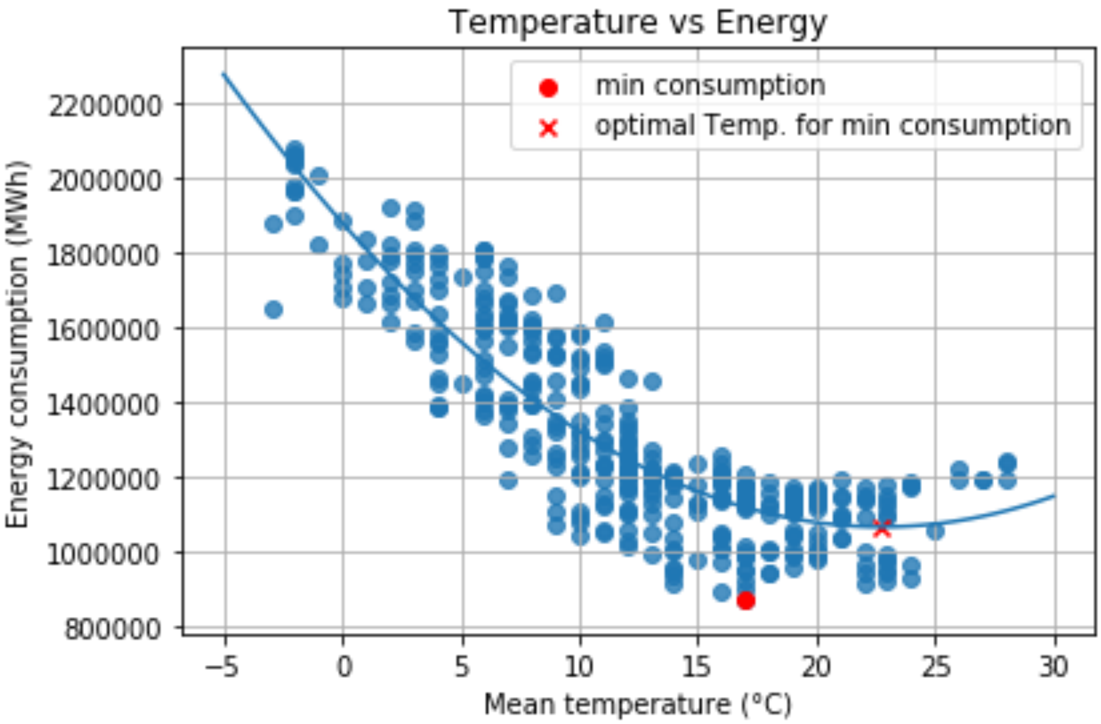


figure 4

7. Use a stepwise approach to find an optimal multivariate linear regression model using the weather variables to forecast the consumption. Which variables are selected? What is the coefficient of determination, R2?

The steps, selected variables, coefficient of determination and R-squared are listed in output below.

```
Iteration 1    Add  high Temp. (°C)            with p-value 1.82151e-93 R-Squared=0.00147621
Iteration 2    Add  high Visibility (km)        with p-value 9.72513e-08       R-Squared=0.687399
Iteration 3    Add  high Humidity (%)           with p-value 0.000846698       R-Squared=0.711618
Iteration 4    Add  avg Temp. (°C)             with p-value 0.014433   R-Squared=0.719564
Iteration 5    Add  low Humidity (%)            with p-value 7.79481e-06       R-Squared=0.725656
Iteration 6    Drop high Temp. (°C)            with p-value 0.834719   R-Squared=0.738679
Iteration 7    Add  avg Dew Point (°C)         with p-value 0.0019757  R-Squared=0.741084
Iteration 8    Drop avg Temp. (°C)             with p-value 0.849876   R-Squared=0.745533
Iteration 9    Add  low Sea Level Press. (hPa)  with p-value 0.0085081  R-Squared=0.747962

================================================================================
Model_1 results:
Selected variables: ['high Visibility (km)', 'high Humidity (%)', 'low Humidity (%)', 'avg Dew Point
(°C)', 'low Sea Level Press. (hPa)']
R-Squared = 0.7503746824703877

const                          3.931924e+06
high Visibility (km)          -5.777534e+03
high Humidity (%)             -2.764919e+03
low Humidity (%)               6.860939e+03
avg Dew Point (°C)            -3.192404e+04
low Sea Level Press. (hPa)    -2.265546e+03
dtype: float64
```

output 3

8. Increase the number of explanatory variables by also considering squared terms for each weather variable. Use a stepwise approach to obtain a new model. Which variables are selected? What is the new R2 value and is this an improvement?

The steps, selected variables, coefficient of determination and R-squared are listed in output below. The performance of new model performs better, the R-Squared improved by 0.0565.

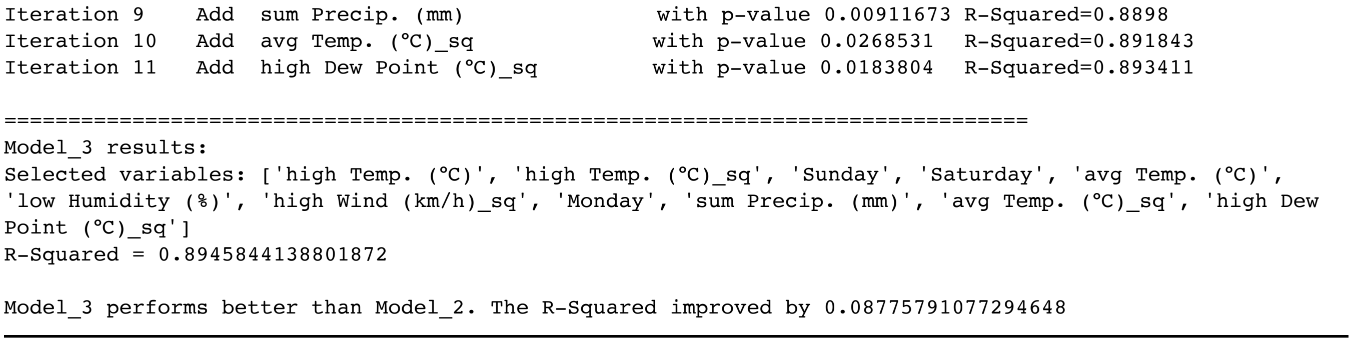
```
Iteration 1    Add  high Temp. (°C)                with p-value 1.82151e-93 R-Squared=0.00147621
Iteration 2    Add  high Temp. (°C)_sq             with p-value 1.18332e-36 R-Squared=0.687399
Iteration 3    Add  high Visibility (km)_sq         with p-value 0.0300572  R-Squared=0.79882
Iteration 4    Add  high Visibility (km)            with p-value 0.00131632 R-Squared=0.802565

================================================================================
Model_2 results:
Selected variables: ['high Temp. (°C)', 'high Temp. (°C)_sq', 'high Visibility (km)_sq', 'high Visibility
(km)']
R-Squared = 0.8068265031072407

Model_2 performs better than Model_1. The R-Squared improved by 0.056451820636852945
```

output 4

9. Consider the day of the week effect by including dummy variables for the day of the week in the multivariate regression. Which days of the week are selected for the new model? What is the new R2 value and does this improve the model?

Seven dummy variables were created for "the day of the week", in order to include the feature of day-of-the-week into the multivariate regression model. Sunday, Saturday and Monday were selected for the new model. This may due to the unusual behavior and activities on weekends and Monday. The performance of new model performs even better than the previous model (Q8), the R-Squared improved by 0.0878.

```
Iteration 1    Add  high Temp. (°C)                with p-value 1.82151e-93 R-Squared=0.00147621
Iteration 2    Add  high Temp. (°C)_sq             with p-value 1.18332e-36 R-Squared=0.687399
Iteration 3    Add  Sunday                          with p-value 2.02543e-18        R-Squared=0.79882
Iteration 4    Add  Saturday                        with p-value 1.51634e-18        R-Squared=0.837335
Iteration 5    Add  avg Temp. (°C)                 with p-value 2.22261e-05 R-Squared=0.868733
Iteration 6    Add  low Humidity (%)                with p-value 2.29513e-06        R-Squared=0.876072
Iteration 7    Add  high Wind (km/h)_sq             with p-value 0.000399405        R-Squared=0.88579
Iteration 8    Add  Monday                          with p-value 0.00398247 R-Squared=0.887122
```



```
Iteration 9    Add  sum Precip. (mm)              with p-value 0.00911673 R-Squared=0.8898
Iteration 10   Add  avg Temp. (°C)_sq            with p-value 0.0268531  R-Squared=0.891843
Iteration 11   Add  high Dew Point (°C)_sq       with p-value 0.0183804  R-Squared=0.893411

================================================================================
Model_3 results:
Selected variables: ['high Temp. (°C)', 'high Temp. (°C)_sq', 'Sunday', 'Saturday', 'avg Temp. (°C)',
'low Humidity (%)', 'high Wind (km/h)_sq', 'Monday', 'sum Precip. (mm)', 'avg Temp. (°C)_sq', 'high Dew
Point (°C)_sq']
R-Squared = 0.8945844138801872

Model_3 performs better than Model_2. The R-Squared improved by 0.08775791077294648
```

output 5

10. Can you be sure that this modeling approach is not over-fitting? Describe two approaches that could be used to prevent over-fitting?

Unless we have more weather data and electricity consumption data, let's say data from 2018, to validate the built model, there is no clear answer of whether this model is over-fitting or not. Although for the training data, the model might seems fitting well by judging from it's R-squared value, it might be over-fitted and when plotted with other data (ie. data in 2018), it will not perform that well.

There are two approach to prevent over-fitting. The first one is to choose the threshold for variables entering and the threshold for variables removing by trial and error. The second approach to prevent over-fitting can be done by partitioning the data we have into testing group and validating group to validate the model performance.