1. Download historical daily weather data for France from https://canvas.cmu.edu/files/5908496/download?download_frd=1 . Load the data into your environment for use. Fill any gaps in the data using linear interpolation.

The Paris weather dataset has been loaded into the ipynb where linear interpolation has been observed. After analyzing columns of High gust wind and Events aren't fully fit with variables, therefore are dropped.

```
Paris_weather.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 19 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Date                 365 non-null    datetime64[ns]
 1   high Temp            365 non-null    int64
 2   avg Temp             365 non-null    int64
 3   low Temp             365 non-null    int64
 4   high Dew Point       365 non-null    int64
 5   av Dew Point         365 non-null    int64
 6   low Dew Point        365 non-null    int64
 7    high Humidity       365 non-null    int64
 8   avg Humidity         365 non-null    int64
 9   low Humidity         365 non-null    int64
 10  High sea level       365 non-null    int64
 11  avg Sea Level Press  365 non-null    int64
 12  low Sea Level Press  365 non-null    int64
 13  high Visibility      365 non-null    float64
 14  avg Visibility�(km)  365 non-null    float64
 15  low Visibility�(km)  365 non-null    float64
 16  high Wind�(km/h)     365 non-null    int64
 17  avg Wind             365 non-null    int64
 18  sum Precip           365 non-null    int64
dtypes: datetime64[ns](1), float64(3), int64(15)
memory usage: 54.3 KB
```

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

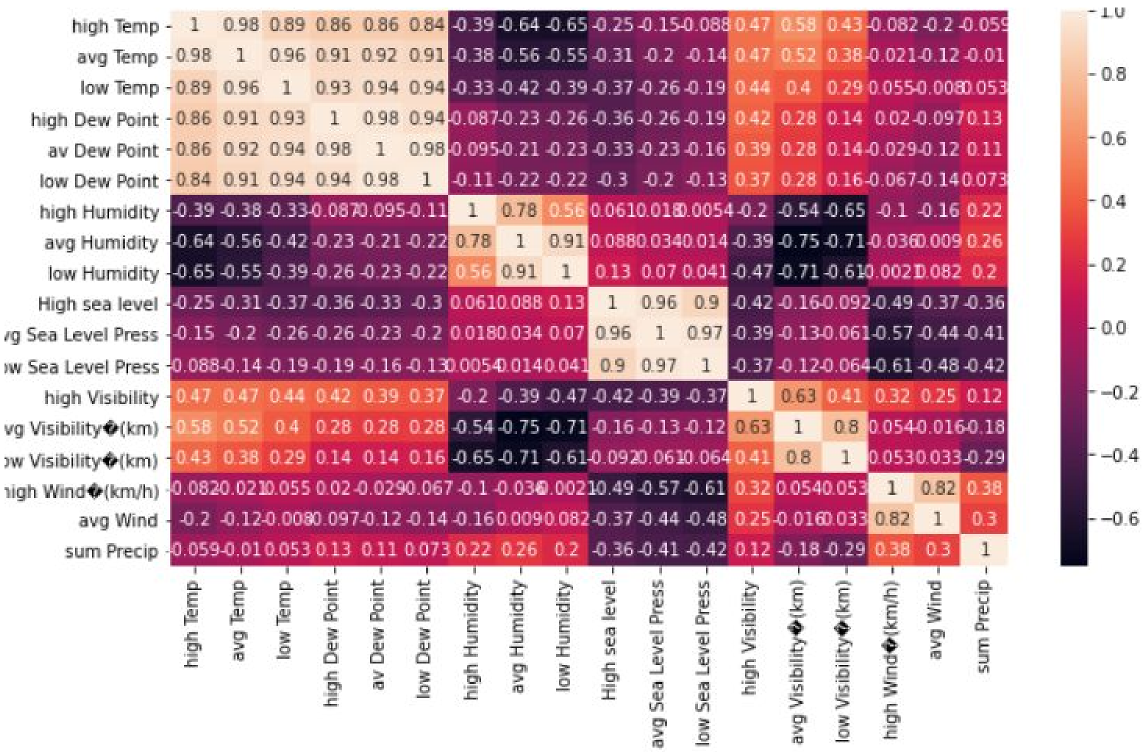
The correlation observed by variable below, does individually reflect on the variables with themselves and on others. I.e High Temp 1 → High Temp 1

High Temp 0.98 → Avg Temp

High Temp -0.65 → Low Humidity

High Temp -0.059 → Sum Precip

From the described data High Temp does strongly correlate with itself (1), and Avg Temp 0.98. The opposite is true for Low Humidity -0.65. Sum Precip -0.059



3. Download historical daily electricity consumption data for France from: https://canvas.cmu.edu/files/5908494/download?download_frd=1
Save it as a csv file and load it into your computer.

```
History={'Date':Historiqu['Date'],'Energy':Historiqu['Energie journalière (MWh)']}


Historica=pd.DataFrame(data=History)
Historica.to_csv('EnergyConsumption.csv')
```

```
Energy_Consumption=pd.read_csv('EnergyConsumption.csv')
Energy_Consumption.head()
```

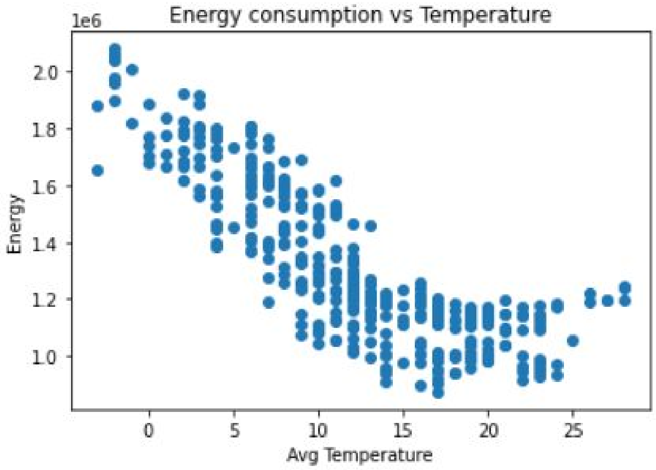
Loading in data. Saving it as Energy_Consumption. Finally, reading it.

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

```
:
plt.scatter(dataset['avg Temp'],dataset.Energy)

plt.xlabel('Avg Temperature')
plt.ylabel('Energy')
plt.title('Energy consumption vs Temperature')
```

```
: Text(0.5, 1.0, 'Energy consumption vs Temperature')
```

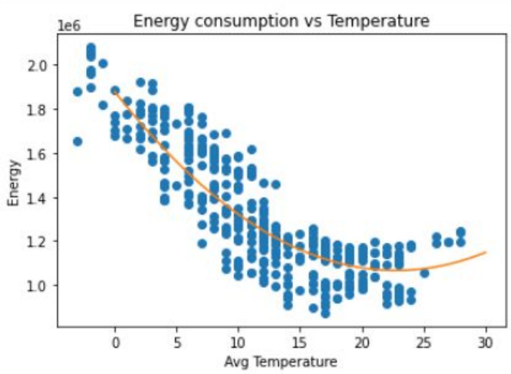


The energy is placed on the y-axis, where by Av Temperature is placed on the x-axis after synchronization the scatter plot is represented above.

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

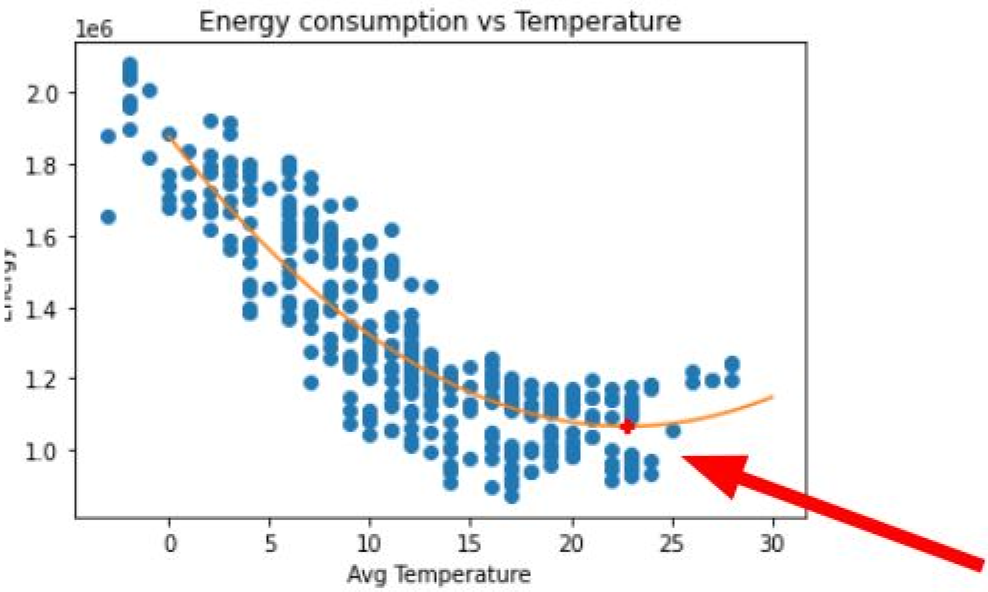
```
In [19]: p=np.poly1d(np.polyfit(dataset['avg Temp'],dataset.Energy,2))

         xp=np.linspace(0,30,200)
         plt.plot(dataset['avg Temp'],dataset.Energy,'o',xp,p(xp),'-')
         plt.xlabel('Avg Temperature')
         plt.ylabel('Energy')
         plt.title('Energy consumption vs Temperature')
         plt.show()
```



The fitted quadratic model is shown above, from the way the line is passing in the middle of the scatter.

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



7. Use a stepwise approach to find an optimal multivariate linear regression model using the
weather variables to forecast consumption. Which variables are selected? What is the coefficient of determination, R 2 ?
Forward regression is used to check which variables are closely correlated to use in the model. After checking in the data set: 'high Temp','high Visibility',' high Humidity','avg Temp','low Humidity','av Dew Point','low Sea Level Press' are provided.
With an rsquared of 0.7506437341112865
.

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 R 2 value and is this an improvement?
After squaring each variable for the forward regression does provide the following output.

'high Temp', 'high Temp^2', 'high Visibility^2', 'high Visibility'
With an improved rsquared of 0.8068265031072407

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 R 2 value and does this improve the model?
Creating Week days on the dataset from which dummies were generated, a new dataset was made. Forward regression was applied on the columns/ variables of the dataset, the results are displayed below. The days of week which are selected are Sunday, Saturday, and Monday.

'high Temp','high Temp^2','Sunday','Saturday','avg Temp','low Humidity','high Wind�(km/h)^2','Monday','sum Precip','avg Temp^2','high Dew Point^2']
The rsquared rocked to 0.8945057843312311

10. Can you be sure that this modeling approach is not over-fitting? Describe two approaches that could be used to prevent over-fitting?
The ability of a model mastering noise over signal is often found in machine learning, thus the model utilized in this homework improved overtime as more data points were added, and squared along the way. I generally don't trust it since numerous data wasn't tested with it. Two approaches to avoid overfitting are cross-validation: splitting your data into numerous data points to train your model, from this the model won't overfit while it's trained on diverse data points. The other point is, removing features which are irrelevant or hold less correlation to your model. Forward regression of a great tool selecting/removing data variables for tuning the model in order to provide better rsquared results. This forward regression does work similar to feature selection which removes and select features to use individually