Load Forecast Modeling Code Project

The goal of this project is to create a load forecast model based on 2 years of data (provided). The project contains two parts:

1. An importer that will import zonal load and temperature data into a SQLite database.
2. A model that will forecast load based on temperature from the data stored in that database.

The project will be coded in Python 2 and any library installable with pip can be used. Your code can be organized as you wish, any number of files or modules can be created apart from the two deliverables described hereunder. You should have received some data files part of the package that has been sent to you. The folder structure looks like this

```
project

 system_load_by_region

 weather_data
```

system_load_by_region contains hourly ERCOT load data by region in CSV format

weather_data contains Wunderground weather station data for Houston, Dallas and San Antonio in CSV format

1. Data Importer

The goal of the importer is to structure the data that has been provided and insert it into a SQLite database so it can easily be queried. The importer should import the temperature and load data into a normalized form in the database.

1. The importer will be named ***importer.py*** and placed at the root of the *project* folder.
2. ***importer.py*** will be run from the command line and take a single argument: the path of the SQLite file to create and write into
3. The importer can assume finding the data relative to the root of the *project* folder.

> **Example**
> `python importer.py test_project.db`

2. Forecast Model

All data for modeling should be at a “Daily” aggregation, both input and output. The load forecast should be forecasting Daily Peak Load.

Relevant Time Periods:

- Calendar year 2014 – This is to be used for estimation
- Calendar year 2015 – Do not use for any model estimation / training, this is the “test” period

Rules for the load forecast model:

1. The forecaster will be called ***model.py*** and placed at the root of the *project* folder.
2. It can use temperature data from a single weather station, and “calendar” variables, as long as the calendar variable cannot be construed as a proxy for weather behavior. (So, a Boolean variable such as IsJuly is not acceptable).
3. Do not use any “autoregressive” (such as previous day load) terms for this project
4. Once the forecast model is estimated, it should be able to take an observation of temperature data, along with calendar variables, and produce the forecasted load.
5. ***model.py*** will be run from the command line with the following arguments:
    a. The SQLite file generated by the importer
    b. A weather station (a string in the list [‘KHOU’, ‘KDAL’, ‘KSAT’])
    c. Load Zone name (a string in the list [‘COAST’, ‘EAST’, ‘FAR_WEST’, ‘NORTH’, …, ‘WEST’, ‘TOTAL’])

> **Example**
> `python model.py test_project.db KHOU EAST`

This script should estimate a load forecast model (using the indicated weather station and load zone), and then run the model for all of 2015, and write a CSV file with the time series of daily predicted peak load.

Example CSV result

```
Date,Forecasted_Peak_Load

01/01/2015,14242

01/02/2015,14095

...
```

3. Evaluation

We will be evaluating:

- the organization and design of the code
- how the data is structured in the database
- the method and design of the load forecast model

We will look at the accuracy of the forecast model in the testing period, in order to understand if it is performing as intended, but the actual accuracy of the load forecast model is not part of the evaluation (although a good design and good forecasting method would tend to produce a more accurate model). We would advise not to focus on the model accuracy during the project, rather we would like you to focus on best practices of writing code, structuring a SQL database, and creating a basic but reasonable load forecast model.

Good Coding !