AEMO Analysis

Oliver Sheridan-Methven

15th May 2019

Abstract

In this report we investigate the relationship between electricity prices and daytime temperatures in Sydney Australia during 2013. We build a second order non-linear model based on the minimum daytime temperature, and predict the prices for the first week in January. This model is able to capture weekend and weekday differences.

The Raw Data

The data available is the RRP (the electricity price measured in AUD A$), and the daily maximum and minimum temperature values, where some example data is shown in Table 1.

Table 1: Some example data illustrating the fields/features available for fitting.

Date	Maximum Temperature (C)	Minimum Temperature (C)	RRP (A$)
2013-01-01	26.2	20.2	46.15
2013-01-02	22.9	20.3	49.94
2013-01-03	24.8	18.4	50.78
2013-01-04	26.6	18.3	49.41
2013-01-05	28.3	20.9	57.72

Outliers

There is a notable outlier in the electricity prices for 20$^{\text{th}}$ December 2013, (caused by an electrical storm). Due to the esoteric nature of such an event, we do not want to consider such a point for use in everyday price predictions, so we exclude this from our considerations. The resultant dependence between the electricity prices and the minimum daily temperature is shown in Figure 1.

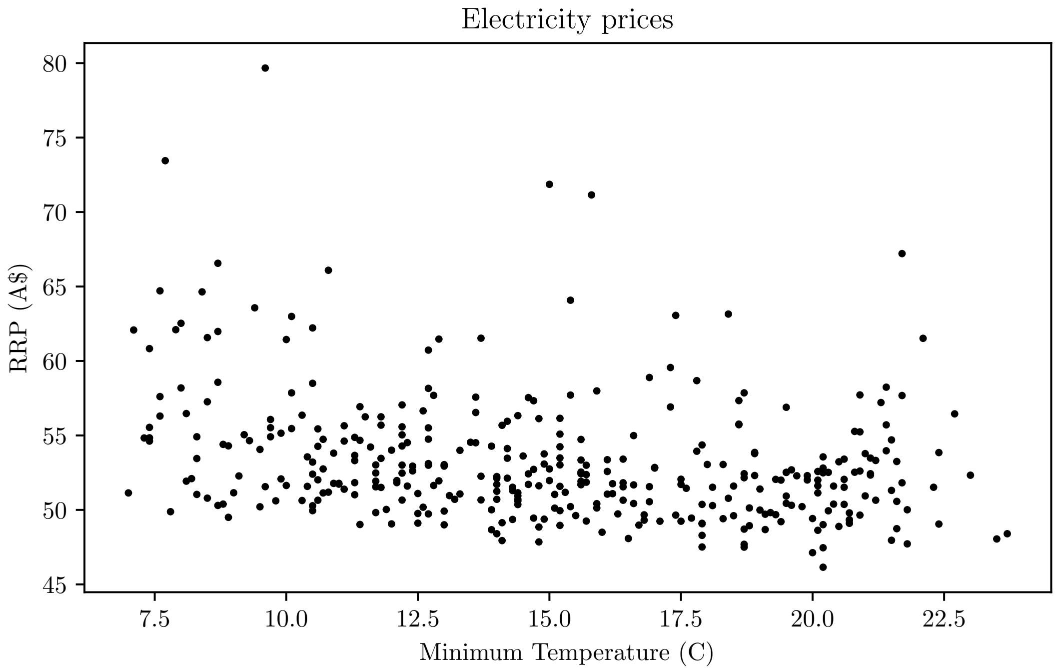

Figure 1: The dependence on the electricity prices with variations in the daily temperature minimum.

Non-linear temperature effects

We recall the linear model

$$Y = \beta X + \epsilon. \tag{1}$$

To assess whether non-linear temperature effects are significant, we fit a linear model and a second order polynomial model to the data. We find the linear model (1) has the p-value 7.1×10^{-9}, which is significant. Furthermore, trying a higher order non-linear model with quadratic terms achieves p-values of $8.1 \times 10^{-81}, 1.6 \times 10^{-5}, 3.7 \times 10^{-4}$, for the intercept, linear, and quadratic terms respectively. We can see these p-values are less than 0.05, and so we believe they are statistically significant to a 95% confidence level (approximately a 2σ confidence).

Building model matrices

We can build model matrices using the following R-function:

```
def pie_linear(x, k):
    """
    Description:
      Produces a model matrix for linear models using knots.
    Input:
      x: Array, sample points.
      k: Array, knot points.
    Return:
      Multidimensional array, model matrix.
    """
    positive = lambda y: (y + np.fabs(y)) * 0.5
    model_matrix = np.zeros([len(x), 1 + len(k)])
    model_matrix[:, 0] = x
    for i in range(len(k)):
        model_matrix[:, i + 1] = positive(x - k[i])
    return model_matrix
```

Which when given $\boldsymbol{x} = (1, 2, \dots, 10)$ and knot points $(2, 6)$ will output the model matrix:

```
x = np.linspace(1, 10, 10)
knots = np.array([2, 6])
X = pie_linear(x, knots)
print(X)
```

```
## [[ 1.  0.  0.]
##  [ 2.  0.  0.]
##  [ 3.  1.  0.]
##  [ 4.  2.  0.]
##  [ 5.  3.  0.]
##  [ 6.  4.  0.]
##  [ 7.  5.  1.]
##  [ 8.  6.  2.]
##  [ 9.  7.  3.]
##  [10.  8.  4.]]
```