

PSBC Project 3

May 8, 2019

1 Introduction

This report details the approach and implementation relating to the second coursework project of MATH36032. Please note that all programming and testing was carried out using GNU Octave rather than MATLAB.

2 Context

The goal of this project was to examine a dataset concerning the sales figures of three products; Bread, Salad and Lettuce. Our main aim was to determine whether there exists a correlation between the temperature of a given day, and the amount of each product was sold. In subsequent sections, we will describe our implementation using the “Bread” dataset for reference, then run the code over the other two.

3 Preprocessing the data

??

The first, simplest preprocessing of the data performed was to avoid working with date strings since this involves working with cell arrays, which are harder to manipulate than vectors and matrices. Instead, each date d was be represented as the number of days between 31/12/2014 and d .

Let us plot a few weeks worth of data and examine the results (shown in Figure ??). We can immediately observe that the data exhibits a periodic trend, with each period being seven days long. Since we are aiming to discover a correlation between temperature and sales, we will need to filter out this periodic effect. For this reason, our analysis will focus heavily on these periods.

To simplify our data analysis, apply a transformation to each relevant column from the initial data. This transform transforms the flat list of values to a matrix whose rows represent periods. Since our periods are seven days long, our matrices each have seven columns.

For example, consider a column vector

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{7m} \end{bmatrix} \tag{1}$$

Under this transform v becomes

$$\begin{bmatrix} v_1 & v_2 & v_3 & \dots & v_7 \\ v_8 & v_9 & v_{10} & \dots & v_{14} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{7m-6} & v_{7m-5} & v_{7m-4} & \dots & v_{7m} \end{bmatrix}$$

Of course, this assumes that the number of data entries is a multiple of seven. By examining the data, we see the first period begins on day 4 and the end of the last period is on the penultimate day, therefore the first three days worth of data, as well as the last day were discarded. The implementation of this is listed in Listing 1.

Listing 1: A function to transform a flat data column into a matrix whose rows represent periods.

```
function [datesPeriods, ...
        salesPeriods, ...
        maxTempPeriods] = preprocessData(data, salesData)

% Trim the start and end of the data set
dates = data.Days(4:end-1);
sales = salesData(4:end-1);
maxTemp = data.MaxTemp(4:end-1);

% Transform the data into a matrix who's rows are periods
datesPeriods = reshape(dates, 7, [])';
salesPeriods = reshape(sales, 7, [])';
maxTempPeriods = reshape(maxTemp, 7, [])';

end
```

4 Examining the period slope

The first step towards computing the correlation we seek is to examine the slope of the periods in the data. For this analysis, we will assume that the slope of each period is equal (for each product), though the average for this period may vary. In order to compute the common slope, we will compute the slope of each period and take their average after eliminating any outliers.

For each product, we have a matrix representing its sales data which we will denote P , defined as follows.

$$v = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \quad (2)$$

Where p_i is a vector of the seven sales data points for period i . Similarly, we have a matrix D , where each row d_i contains date values for the period i , indexed as described in Section ??.

Least square fitting was used to compute the slope of the data for each period. However, there are certain days throughout the dataset whose sales figures are zero; these must be omitted

from our calculations. For ease of notation let $p = [p_1 \dots p_k]$ be the product sales data for an arbitrary period and $d = [d_1 \dots d_k]$ the corresponding dates after dates whose sales data is zero have been removed (so $k \leq 7$). This approach uses the following two equations to compute the slope m and intercept c .

$$m \sum_{j=1}^k d_j^2 + c \sum_{j=1}^k d_j = m \sum_{j=1}^k d_j \times p_j$$

$$m \sum_{j=1}^k d_j + c \sum_{j=1}^k 1 = m \sum_{j=1}^k p_j$$

These equations are derived from equations described by S.J Miller [?]. Notice that to compute of m and c we can express the above equation in matrix form as follows. This allows the system to be solved easily in Octave.

$$\begin{bmatrix} \sum_{j=1}^7 d_j^2 & \sum_{j=1}^7 d_j \\ \sum_{j=1}^7 d_j & \sum_{j=1}^7 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^7 d_j \times p_j \\ \sum_{j=1}^7 p_j \end{bmatrix}$$

Figure ?? shows the results of this computation for the Bread data, performed on the first few periods which shows promising results for estimating the slopes. Figure ?? shows all the slopes and reveals evident outliers highlighted in red. We will discuss the approach to identifying outliers next.

4.1 Identifying and eliminating outliers

Tukey fences [?] is the method used to identify the periods which are outliers. It relies on the computation of quartiles in the data, (the data here being the list of slopes).

This is a reasonably simple method; we work out the first and third quartiles in the data and denote them Q_1 and Q_3 respectively, let the interquartile range ($Q_3 - Q_1$) by IQR . Denote the slope for period i as s_i . Now we define outliers to be all the slopes s_i such that either of the following holds:

$$s_i < Q_1 - 1.5 \times IQR,$$

$$s_i > Q_3 + 1.5 \times IQR.$$

The number 1.5 was a number used initially in this context by John Tukey [?] since it was effective in many applications. In our case, the figure 1.5 works well, evident when we plot the results as shown in Figure ?? (this is similarly effective for the other two products). Outliers are identified correctly. Now that this is done, we can take the mean of the remaining values to obtain the value we need. Listing 2 shows the code used to identify outliers. Note, the preprocessing of the data now makes it very easy to eliminate entire periods of data (those with slopes that are outliers) in the next part of our analysis by removing the corresponding rows of the data matrices which is a simple operation.

Listing 2: A function to identify outliers in a column of data

```
function [datesPeriods, ...
        salesPeriods, ...
        maxTempPeriods] = preprocessData(data, salesData)
```

```

% Trim the start and end of the data set
dates = data.Days(4:end-1);
sales = salesData(4:end-1);
maxTemp = data.MaxTemp(4:end-1);

% Transform the data into a matrix who's rows are periods
datesPeriods = reshape(dates, 7, [])';
salesPeriods = reshape(sales, 7, [])';
maxTempPeriods = reshape(maxTemp, 7, [])';

end

```

5 Computing the correlation coefficient

Now that we have a mechanism of computing the slope for a given product we can use it to filter out the periodic effect in order to gain insight into the correlation between temperature and sales. Denote this common slope by μ . Note that from now on we will be working with data that omits the periods whose slopes are outliers.

Our aim now is to define a transformation T on the sales data such that the transformed data exhibits trends unaffected by the day of the week, i.e. data which is independent of the periodic effect. To achieve this we make the following observation. For a given period, the deviation of a specific data point from the mean for that period is dependent on two things: temperature and day of the week.

Let us first define the mean for the period. Here our x-axis is the date and the y-axis is the sales data. Define the sales data \underline{p} and dates \underline{d} as before. We can represent the mean point by (\bar{d}, \bar{p}) . Now define the expected sales value by the line defined by the slope μ and this mean point as illustrated in Figure ???. Using our observation, we now work with the assumption that the difference δ_i between the expected sales value $ex_{p_i}(d_i)$ at date d_i , and the actual value p_i is the influence of the temperature. Therefore we can define our transform as follows.

$$T(p_i) = \bar{p} + \delta_i$$

Figure ?? shows the data for the first few periods as well as the transformed data. It is clear the periodic effect is largely eliminated. Applying the transformation T to each period and concatenating the results now gives a flat list of transformed values. By flattening the temperature matrix M we now have data of a form that can be processed by the `corrcoef` command to compute a correlation coefficient. Flattening the matrices consists of taking each row and forming a vector by concatenating them.

However, Figure ?? shows the results of plotting the transformed data against the temperature data. Clearly, there are two separate correlations, in investigating this, it appears sales data spiked in the third year, a trend not related to the temperature. To mitigate this we compute two correlation coefficient, one for the third year and one for the other two. Of course, in computing their mean it is vital to weight the first coefficient twice as much as the second.

The results of our analysis are below in Listing ??, having run the analysis on each product data.

```

>> main
    Bread: Slope = 17.1303; Correlation Coefficient = 0.769772
    Lettuce: Slope = 23.0191; Correlation Coefficient = 0.7539
    Salad: Slope = 11.8209; Correlation Coefficient = 0.63174

```

Appendix A Distance