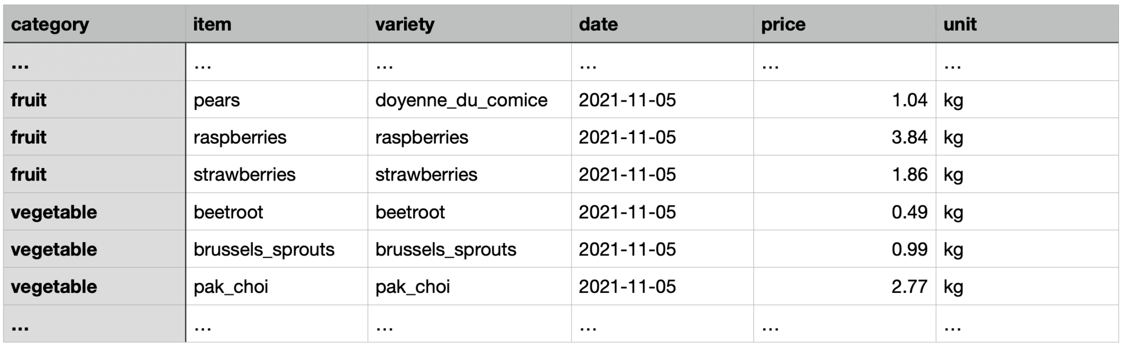
Project 3 Report - Data Analysis

UID: 10450192

December 17, 2021

1 Introduction to the problem

The aim of the script is to perform a series of analyses on a data set called *fruitvegprices-15nov21.csv*. It contains information about various natural products like ’apples’ and ’carrots’. Most notably it contains the prices of certain varieties of these on different dates between the years 2017 and 2022. The currency is not specified but the prices will be signified with “$” in this report, just to distinguish them from other values. What follows is a couple of rows from the middle of the data set which show all columns and some possible entries:



category	item	variety	date	price	unit
...	...	...	...	...	...
fruit	pears	doyenne_du_comice	2021-11-05	1.04	kg
fruit	raspberries	raspberries	2021-11-05	3.84	kg
fruit	strawberries	strawberries	2021-11-05	1.86	kg
vegetable	beetroot	beetroot	2021-11-05	0.49	kg
vegetable	brussels_sprouts	brussels_sprouts	2021-11-05	0.99	kg
vegetable	pak_choi	pak_choi	2021-11-05	2.77	kg
...	...	...	...	...	...

Figure 1: A small section of *fruitvegprices-15nov21.csv*. The full data set contains 9148 rows of 6 columns.

The 5 tasks to be accomplished are:

1. List all distinct “items” in the data set and all distinct varieties for apples, carrots, pears and cabbage.
2. Find the mean price for each apple variety.
3. Create a box plot of prices for each apple variety and analyse the result.
4. Find the seasonal trend for the apple variety with the smallest mean by analysing its time series.
5. Find the correlation coefficient between carrots and a chosen apple variety by comparing their time series.

Since the entire project is centred around data analysis, the live script (“notebook”) editor was used so that each task can be neatly contained in its code cell while still having access to all previously defined variables.

2 Brief overview of the theory

2.1 Box plots

This section aims to provide a quick introduction to the construction of box plots (a diagram of which can be seen in Figure 2). The main body of a box plot is the box itself. It is defined to run from the 1st to 3rd quartile of a sample, meaning it spans 50% of the sample's data points by definition. The length of the box is known as the interquartile range (IQR). Inside of it, there is a single vertical line that symbolises the median of the sample, hence dividing the box into two parts containing 25% of the sample each. The "whiskers" on two sides are supposed to show the minimum and maximum values of the sample. However, if there are points that are far away from the box (usually more than 1.5IQR away from the box's boundary in the given direction), they are often treated as outliers and not included by the whiskers. In a more realistic case than the one depicted in Figure 2, the median will often be skewed to either side of the box and the whiskers will have uneven lengths, signifying the last point in the given direction which is not far enough to be treated as an outlier. However, if the sample is distributed close to normally, then the ends of the two whiskers cover 99.3% of all data points[1].

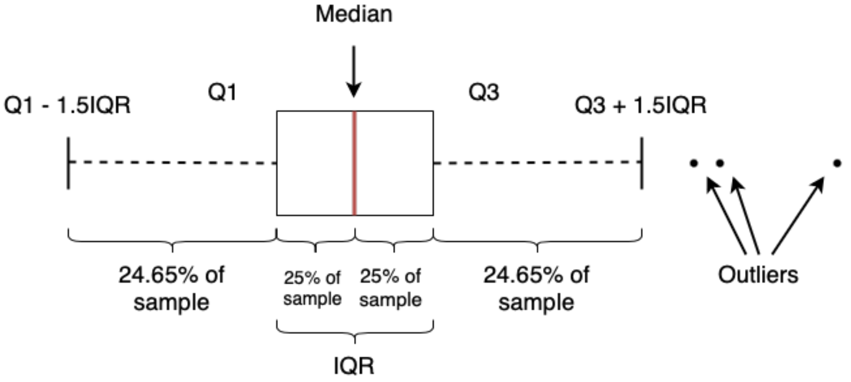


Figure 2: An example of a box plot for a normally distributed sample.

2.2 Correlation coefficient

The correlation coefficient measures the linear association between two variables[1]. In practice, this means that the coefficient is 1 or close to 1 when the two variables are closely related and tend to increase and decrease together (if its value is -1, then they increase and decrease exactly opposite of one another). On the other hand, if the coefficient is 0, then there is no relation whatsoever. In practice, any coefficient whose absolute value is bigger than 0.7 is considered to show a significant correlation. Anything less shows at best a low correlation between the two variables in question. Mathematically, the correlation coefficient ρ can be expressed as:

$$\rho(X, Y) = \frac{1}{N-1} \sum_{n=1}^{N} (\frac{X_n - \mu_X}{\sigma_X})(\frac{Y_n - \mu_Y}{\sigma_Y}) \,,$$

where X and Y are the observed variables, each with N observations. μ_i and σ_i correspond to the average and standard deviation of the i-th observable[2].

3 Task 1 - distinct items and varieties

To begin, the function `readtable()` has been used which imported the data into a `table` variable that can be readily used in MATLAB. Then, `groupsummary()` was called to extract group counts sorted by headers 'item' and 'variety' simultaneously. The actual values of group counts are not very important but the function is a neat way to quickly obtain a smaller table where each variety appears only once but is still connected to its root 'item'. With that, an empty `containers.Map` was created and then filled with 'items' as keys and 'varieties' as values using a `for` loop iterating over `groupsummary()` table. This `Map` (called `products`) contains all information needed to complete task 1. Distinct items can be extracted by calling `products.keys` and distinct varieties for a given item by calling `products(given_item_name)`. The full code cell for task 1 can be seen below:

```matlab
% TASK 1
data = readtable('fruitvegprices-15nov21.csv');
unique_products = groupsummary(data,{'item','variety'});
products = containers.Map();

for i = 1:height(unique_products)
    item = string(table2array(unique_products(i, 'item')));
    variety = string(table2array(unique_products(i, 'variety')));
    if ¬isKey(products, item)
        products(item) = [variety];
    else
        products(item) = [products(item); variety];
    end
end

distinct_items = products.keys
Apples = products('apples')
Pears = products('pears')
Carrots = products('carrots')
Cabbage = products('cabbage')
```

For the sake of formatting, the output of this task has been placed in appendix A.

4 Task 2 - mean prices of apples

```matlab
% TASK 2
prices_apples = [];
varieties_apples = [];
cheapest_variety = [inf, "_"];

for i = 1:length(Apples)
    Variety = Apples(i);
    reduced_data = table2array(data(strcmp(data.variety, Variety), 'price'));
    mean_price = mean(reduced_data);
    if mean_price < str2double(cheapest_variety(1))
        cheapest_variety = [mean_price, Variety];
    end
    disp(Variety + " mean price = $" + mean_price)
    prices_apples = [prices_apples; reduced_data];
    varieties_apples = [varieties_apples, repmat(Variety, 1, length(reduced_data))];
end

disp(cheapest_variety)
```

The code in this section (seen above) focuses on task 2 but also prepares variables for task 3 since to calculate the mean price and create a price box plot, the same array has to be used. Firstly, two empty variables are created: prices_apples and varieties_apples which will be used for the box plot. At the same time, the variable cheapest_variety is created and set equal to [inf, "_"] where the first entry is the mean price and second the name of variety. The inf represents infinity and is used as a placeholder since any mean calculated will be less than it. Thus, it simplifies the code needed in the for loop.

The loop itself iterates over the apple varieties and simultaneously calculates and displays the mean price of each variety, saves the cheapest variety to the cheapest_variety variable and appends all prices to prices_apples. The function repmat is used when appending each variety's name to varieties_apples so that this name appears in the array as many times as there are prices recorded for it. The output for this task shows that the cheapest variety is the "other mid season" with a mean price of $0.80 ($0.80344). The rest of the output can be found in appendix A.

5 Task 3 - box plots of apples' prices

Since the variables prices_apples and varieties_apples have been filled in such that they have the same length, the code needed to create a box plot is very simple:

```
% TASK 3
boxplot(prices_apples, varieties_apples)
title("Box plot of apple varieties' prices")
ylabel('Price ($)')
xlabel('Variety')
```

The resulting plot looks as follows:

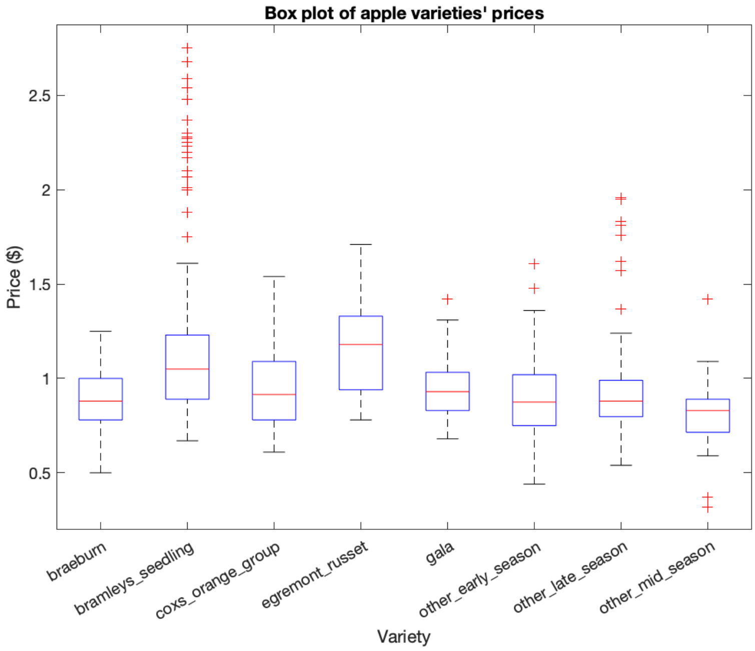


Figure 3: The parallel box plots for all apple varieties.

As can easily be seen, most of the prices for the different varieties follow slightly skewed normal distributions, where "Braeburn" variety is almost perfectly symmetric. Most of the apples' prices appear to be in the same range and with similar variation over the measured period. Notably, as found in task 2 the "other mid season" variety appears to be the cheapest and does not vary significantly in price. That makes sense as this variety is most likely a collection of unnamed or less known apple types at the peak of apple season. The two other varieties which stand out are "Bramley's seedling" and "other late season". These show a big number of outliers a significant distance away from the central box. The most likely explanation for "Bramley's" would be temporary trends which made the variety more in-demand. Outliers for "other late season" are harder to explain; perhaps there was a smaller supply of apples in the late season of one year which increased their price. However, all these are just hypotheses - the data here cannot provide the full answer. Finally, we can notice that the "Egremont russet" variety is typically the most expensive one, ignoring the outliers. Its median price is somewhere around $1.2 and it is not uncommon to see apples of this variety sell for as much as $1.5. In comparison, other varieties do not usually sell for this much.

6 Task 4 - time series for the cheapest apple variety on average

To construct the time series for the cheapest variety, the corresponding data has to be first extracted from the full data table. For that, the second entry of `cheapest_variety` variable has been used with `strcmp()` - a function which compares two strings. Used within the table variable it extracts only those rows which have the required cheapest variety. Having done that, the plotting can be performed right away. The code cell can be seen below:

```
% TASK 4
cheap_apple_data = data(strcmp(data.variety, cheapest_variety(2)), {'price', 'date'});
plot(table2array(cheap_apple_data(:, 'date')), table2array(cheap_apple_data(:,'price')))
title(cheapest_variety(2)+' variety time series', 'Interpreter', 'none')
xlabel('Date')
ylabel('Price')
```

The produced time series is:

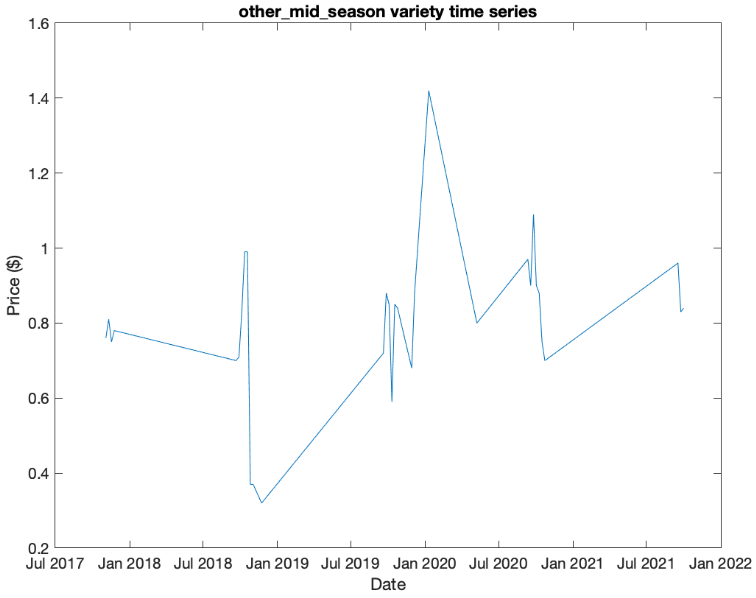


Figure 4: The time series for the cheapest apple variety on average.

From the time series, we can see a big jump in price from January 2019 to January 2020, likely meaning that less known apple types (maybe local varieties) or apples in general were more in-demand in 2020. Alternatively, it could just mean a smaller supply of apples in 2020 as compared to 2019 due to some supply shock. Outside these two years, the price variation is much less significant but that could simply be due to the short period recorded in the data set. Perhaps on a longer time scale, there may be a clearer trend wherein this variation is more/less popular every couple of years.

7 Task 5 - price correlation for carrots and "Bramley's seedling"

For the calculation of the correlation coefficient, the "Bramley's seedling" variety of apples has been chosen. The reason is that it contains around the same number of data points (203) as the single variety of carrots (199), meaning the correlation coefficient will be more significant (for comparison, the number of data points of the variety from the last task is just 32). First, all relevant rows have been extracted from the main table using `strcmp()`. Next, the `dates` variable was created from the dates recorded for carrots and `carrot_prices` variable created from corresponding prices. Using the `dates` vector, the prices of "Bramley's seedling" only on the same dates as the ones recorded for carrots have been saved into `bramleys_prices`. At this point, we are finished with this specific example since the two price arrays and the dates array have the same length. If it turned out that the price of "Bramley's seedling" was not recorded on some of the dates in `dates`, then we would need to implement another round of reduction to delete those data points in `carrot_prices` and `dates` which do not share dates with "Bramley's seedling".

What remains to do is calling the function `corrcoef()` to calculate the correlation matrix. Since only two series are compared, the required correlation coefficient lies on the off-diagonal of this matrix and is equal to 0.5034. The code cell can be seen below:

```
% TASK 5
bramleys_data = data(strcmp(data.variety, 'bramleys_seedling'), {'price', 'date'});
carrot_data = data(strcmp(data.item, 'carrots'), {'price', 'date'});

dates = table2array(carrot_data(:, 'date'));
carrot_prices = table2array(carrot_data(:, 'price'));
bramleys_prices = table2array(bramleys_data(ismember(dates, ...
    table2array(bramleys_data(:,'date'))), 'price'));

coeff = corrcoef(bramleys_prices, carrot_prices);
correlation_coefficient = coeff(2,1)
plot(dates, bramleys_prices, 'g-', dates, carrot_prices, 'r-');
legend(["Bramley's seedling prices", "Carrot prices"], 'Location','best')
title("Comparison of price time series for carrots and Bramsley's seedling apples")
xlabel('Date')
ylabel("Price")
```

This code also plots the two series against each other so that the correlation can be seen more easily:

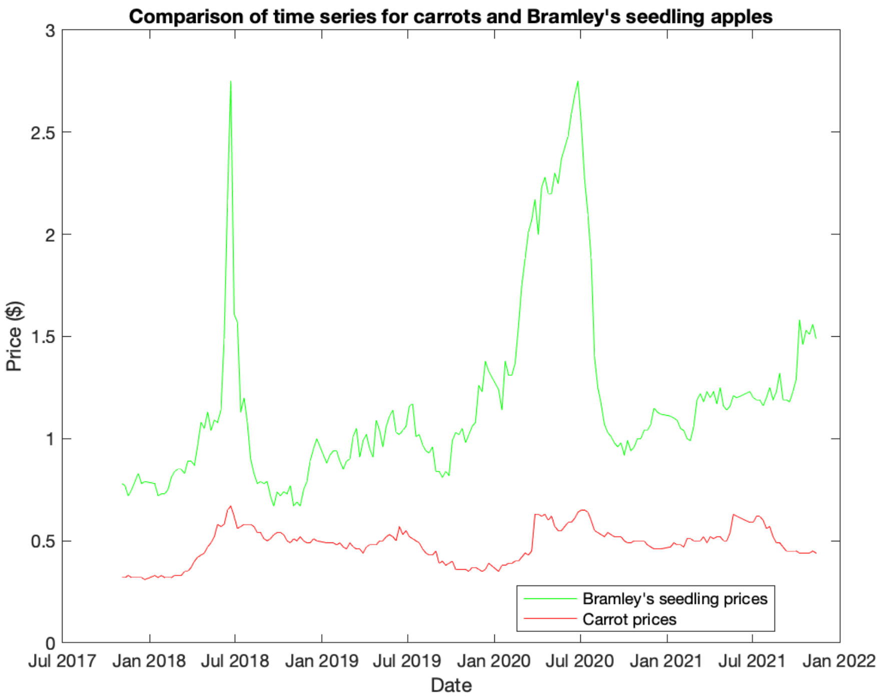


Figure 5: The time series for carrots and Bramley's seedling.

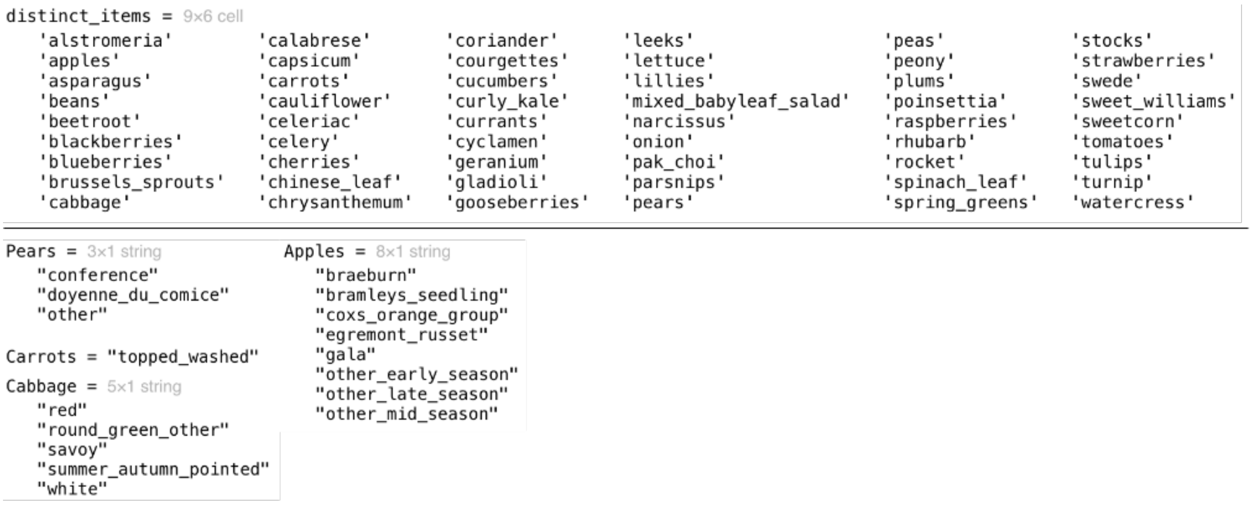
The output correlation coefficient suggests a low level of correlation between the two prices and this can be seen visually as the peaks and valleys of the two time series seem to only vaguely align. It is difficult to decisively say how this correlation arises. Perhaps the prices of both rise when there is a bigger societal trend of healthy eating. However, the more likely explanation is that these two are simply impacted by the natural variations in the economy at large. The deviation from stronger correlation makes sense in this case as the actual shape of the time series will be dictated by supply and demand dynamics for that specific product. For example, the two peaks for "Bramley's seedling" which are much more extreme than the other in the two series do not seem likely to be caused by the same factors which produced the peaks in carrots prices at around the same time.

References

[1] Sarah Boslaugh. *Statistics in a Nutshell: A Desktop Quick Reference*. O'Reilly Media; Second edition, 2012.

[2] *corrcoef*. MathWorks. https://www.mathworks.com/help/matlab/ref/corrcoef.html.

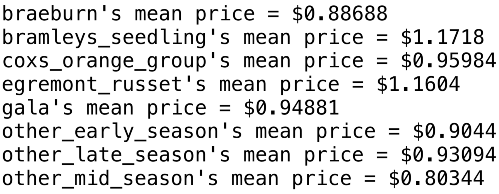
Appendix A



```
distinct_items = 9×6 cell
    'alstromeria'        'calabrese'        'coriander'       'leeks'                    'peas'            'stocks'
    'apples'             'capsicum'         'courgettes'      'lettuce'                  'peony'           'strawberries'
    'asparagus'          'carrots'          'cucumbers'       'lillies'                  'plums'           'swede'
    'beans'              'cauliflower'      'curly_kale'      'mixed_babyleaf_salad'     'poinsettia'      'sweet_williams'
    'beetroot'           'celeriac'         'currants'        'narcissus'                'raspberries'     'sweetcorn'
    'blackberries'       'celery'           'cyclamen'        'onion'                    'rhubarb'         'tomatoes'
    'blueberries'        'cherries'         'geranium'        'pak_choi'                 'rocket'          'tulips'
    'brussels_sprouts'   'chinese_leaf'     'gladioli'        'parsnips'                 'spinach_leaf'    'turnip'
    'cabbage'            'chrysanthemum'    'gooseberries'    'pears'                    'spring_greens'   'watercress'
```

```
Pears = 3×1 string                Apples = 8×1 string
    "conference"                      "braeburn"
    "doyenne_du_comice"               "bramleys_seedling"
    "other"                           "coxs_orange_group"
                                      "egremont_russet"
Carrots = "topped_washed"             "gala"
                                      "other_early_season"
Cabbage = 5×1 string                  "other_late_season"
    "red"                             "other_mid_season"
    "round_green_other"
    "savoy"
    "summer_autumn_pointed"
    "white"
```

Figure 6: Output for Task 1



```
braeburn's mean price = $0.88688
bramleys_seedling's mean price = $1.1718
coxs_orange_group's mean price = $0.95984
egremont_russet's mean price = $1.1604
gala's mean price = $0.94881
other_early_season's mean price = $0.9044
other_late_season's mean price = $0.93094
other_mid_season's mean price = $0.80344
```

Figure 7: Output for Task 2