MATH36032 Project 3

Rudi Agnew 1013652

1 The data

The data set was a table consisting of close to 12000 entries and was given in the form of a *.csv* file. The columns of the table gave information about the bananas, that is their country of origin, the date the information was gathered, the price of the banana and the units of that price.

	Origin	Date	Price	Units
1	'belize'	15/01/2021	0.8600	'£/kg'
2	'costa_rica'	15/01/2021	0.8700	'£/kg'
3	'dominican_...	15/01/2021	0.5500	'£/kg'
4	'ecuador'	15/01/2021	0.8300	'£/kg'

Figure 1: Figure showing the first 4 entries of the data set.

The table was arranged in chronological order and there were many entries for each of the different countries.

2 Question 1: Unique origins

My first task was to produce a list of distinct entries in the 'Origin' column. This could be done in one line of code. I first used the **readtable()** function to put the data file into MATLAB. I then used dot notation to access the 'Origin' column. Finally I called the **unique()** function to only return the distinct entries of that column and transposed the whole thing to give a row vector. I chose to have the data as an input parameter rather than hard code it in to give me more flexibility in using this function, for example if I had a bigger data set with the same table structure the function would still work.

Listing 1: MATLAB code for unique_origins

```
1 function [unique_origins] = unique_origins(data)
2 % Function that returns the distinct origins of the data
3 unique_origins = unique(readtable(data).Origin)';
4 end
```

This unique list of origins served as a sort of library for me to access all of the different origins that existed in my data. This was useful later when I was asked to compare data from different origins.

The output was 27 distinct origins, since this is a large amount I shall include but a sample of the output.

```
>> unique_origins('bananas-18jan21.csv')
ans =
  1×27 cell array
  Columns 1 through 11
    {'acp_bananas'}    {'all_bananas'}    {'belize'}    {'brazil'}    {'cameroon'}
```

This could be further formatted for purely aesthetic reasons using the in built function **cell2table()** on the function I created, however the functionality was fine so I left it as it is.

3 Question 2: Comparing price fluctuations

3.1 My task

The question was to find the 3 countries with the highest mean price of bananas for the last 5 years of data for that specific country and also the 3 countries with the lowest mean price. My interpretation of this was to take the most recent data entry for a country and subtract 5 years to get my timescale. For example, Belize had 340 entries running from 15/01/2021 to 07/01/2000, so I only included entries from 15/01/2021 to 15/01/2016 when calculating the mean. However Malaysia only has three entries, one from 1997 and two from 1995 so I used all 3 of these entries when calculating the mean.

3.2 My Approach

First I used the **readtable()** function to store the data set as a variable called *dat*. I then called my **unique_origins()** function to generate a list of distinct origins, this was stored as the *unique_origin* variable. Using a **For** loop, I looped through all elements of *unique_origin*. In this loop I used the inbuilt function **strcmp()**, which generates a logical array, to create a table of just the origin we are on in the loop. Calling **sortrows()** on this table allowed me to sort by date with the most recent entry at the top. This was stored as the variable T.

Next I had to deal with the date restrictions. Since the dates were all of the structure being DD/MM/YYYY, I was able to manipulate them using MATLAB's inbuilt date and time functions [2]. To access the dates I used dot notation on T to obtain the first and thus most recent date. I then used the **days()** and **calyears()** functions to create upper and lower bounds x and y. Using the **isbetween()** function I generated a logical array determining what dates of T fell within the bounds x and y. I then applied this logical matrix to T giving me the data entries for the time period. I stored this as a new

table *T1*. To calculate the mean I simply used the **mean()** function the price column of *T1*.

These last bits of code are just picking out the top and bottom 3 and arranging them neatly in a table. While still in the loop I created a new structure which has the origin and mean price as its fields. Outside of the loop I convert my new structure into a table purely so I could call **sortrows()** on it and get the prices in order of highest to lowest. I then returned a table of the top 3 entries and a table of the bottom 3 entries.

Listing 2: MATLAB code for meanprice

```
function [f1,f2] = meanprice(data,t,n)
% Calculates the mean price of data over a duration of time
% f1 top 'n', f2 bottom 'n'
% t is how many years in the past you wish to go
% n represents the top 'n' and bottom 'n' countries
dat = readtable(data); % reads file
unique_origin = unique_origins(data);
for i = 1:length(unique_origin)
    % splits dat into different tables depending on origin and sorts
    T = sortrows(dat(strcmp(dat.Origin, unique_origin(i)),:),'Date', '
       descend');
    x = T.Date(1);
    y = T.Date(1) - calyears(t); % takes away t calendar years from the
    m = isbetween(T.Date,y,x); % x,y are closed intervals
    T1 = T(m,:);
    mean_price = mean(T1.Price);
    struc(i).Origin = unique_origin{i};
    struc(i).Mean_Price = mean_price;
end
Y = struct2table(struc);
final = sortrows(Y,'Mean_Price', 'descend');
if n >= length(unique_origin)
    n = length(unique_origin);
end
f1 = final(1:n,1:2);
f2 = final(height(final)-(n-1):end,1:2);
```

```
26 end
```

I've set this function up so you can change the data set as long as it retains the same table structure, for example you could increase the amount of countries of origin or collect more data from the same countries etc. You can also change the time scale, i.e. how far back from the most recent year you want to go. You can also view the top and bottom 'n' countries, rather than just the top 3. To do this last bit I included an **If** statement to check that the 'n' selected was less than or equal to the amount of unique origins, otherwise there would be an indexing error. If the n chosen was too large I just set it to the max value it could be, i.e. the length of the *unique_origin* variable.

3.3 Results

My resutls were Somalia, Panama and Costa rica for my top 3, with mean prices 0.83£/kg, 0.79£/kg and 0.77£/kg respectively and rounded up to the nearest pound. My bottom three were the Windward Isles, Surinam and Venezuela with means of 0.47£/kg, 0.50£/kg and 0.52£/kg respectively.

```
>> [top,bottom] = meanprice('bananas-18jan21.csv',5,3)
top =                                bottom =
  3×2 table                            3x2 table
       Origin        Mean_Price    |        Origin            Mean_Price
    --------------    ----------    |    ------------------    ----------
    {'somalia'   }        0.83      |    {'venezuela'     }      0.5212
    {'panama'    }     0.78896      |    {'surinam'       }     0.50714
    {'costa_rica'}     0.76873      |    {'windward_isles'}     0.47026
```

4 Question 3 and 4: Plotting price variations

4.1 My task

For Q3 I was to compare three specific origins by plotting the price fluctuations over time. These origins were Costa Rica, The Windward Isles and Ecuador. I was also to plot the origin 'all_bananas' with the other three. Q4 was the same but with the restricted time frame of 01/01/2016 to 31/12/2020. Because of the similarities, I did both the questions in one function.

4.2 My Approach and result

For Q3 I defined *dat* as before, used dot notation and then **strcmp()** to retrieve tables of the specified origins. I then sorted these tables by date with the oldest entry at the top. Using the **plot()** function, I plotted the dates and prices of each origin. For Q4 I introduced upper and lower time limits. I did this using **datetime()** [2]. This function creates a variable which was of the same format as the date column in my data, therefore I was able to use the **isbetween()** function with the time limits as new parameters to find entries only within the specified time.

Listing 3: MATLAB code for dataplot3

```
function [plt1,plt2] = dataplot3()
% plots date vs price for 3 origins over a specified period
dat = readtable('bananas-18jan21.csv'); % reads file
cr = sortrows(dat(strcmp(dat.Origin, 'costa_rica'),:),'Date','ascend');
   % gets costa rica entries and sorts by date
wi = sortrows(dat(strcmp(dat.Origin, 'windward_isles'),:),'Date','ascend
   ');
ec = sortrows(dat(strcmp(dat.Origin, 'ecuador'),:),'Date','ascend');
ab = sortrows(dat(strcmp(dat.Origin, 'all_bananas'),:),'Date','ascend');
t1 = datetime(2016,01,01); % year/month/day
t2 = datetime(2020,12,31);
cr1 = cr(isbetween(cr.Date,t1,t2),:);
wi1 = wi(isbetween(wi.Date,t1,t2),:);
ec1 = ec(isbetween(ec.Date,t1,t2),:);
ab1 = ab(isbetween(ab.Date,t1,t2),:);
figure()
plt1 = plot(cr.Date,cr.Price,wi.Date,wi.Price,ec.Date,ec.Price,ab.Date,
   ab.Price);
legend({'costa\_rica','windward\_isles','ecuador','all\_bananas'},'
   FontSize',20); % \ for formatting
title('Plot of banana price fluctations over time','FontSize',20)
xlabel('Date','FontSize',20)
ylabel('Price','FontSize',20)
figure()
plt2 = plot(cr1.Date,cr1.Price,wi1.Date,wi1.Price,ec1.Date,ec1.Price,ab1
```

```
    .Date,ab1.Price);
22 legend('costa\_rica','windward\_isles','ecuador','all\_bananas', '
    FontSize', 20,'Location','northwest');
23 title('Plot of banana price fluctations over time', 'FontSize',20)
24 xlabel('Date','FontSize',20)
25 ylabel('Price','FontSize',20)
26 end
```

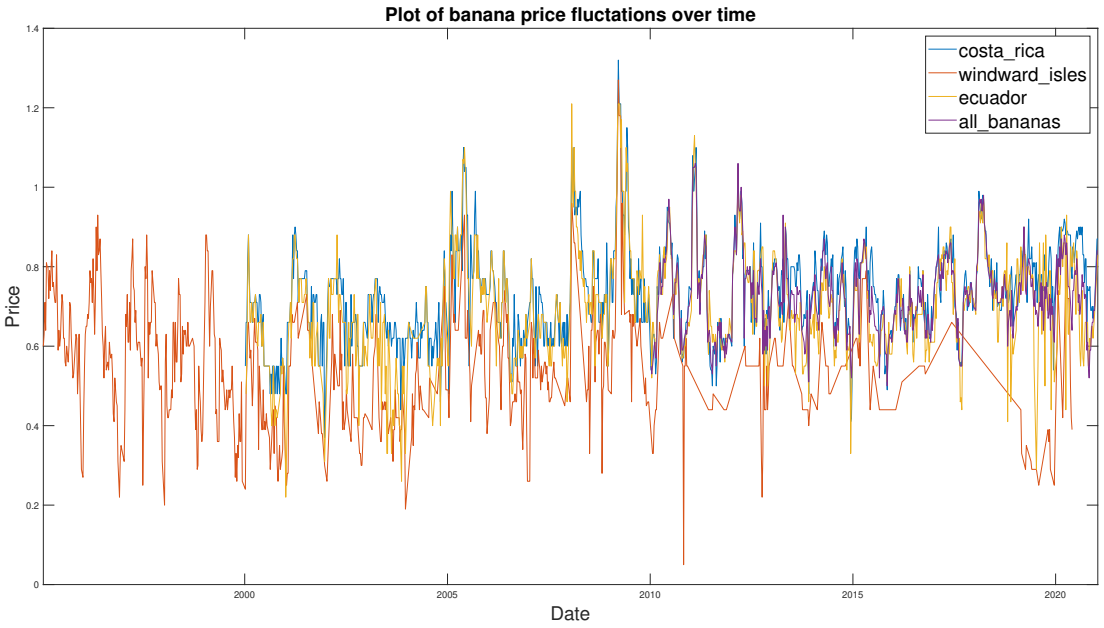

Figure 2: Graph showing the fluctuations of prices of bananas over the years.

4.3 Results and analysis

Price are lowest during winter up to January where they then skyrocket and peak, plateauing throughout summer and then decrease again. According to this website [4], the low period is during the banana growing country's summertime, since bananas are easier to grow during summer there is have a surplus of bananas causing the price to drop. A paper [3] I found online has similar findings to me but from a data set going from 1980 to 1997. It also agrees with me in why the banana prices vary and also offers another explanation which is that people in the northern hemisphere tend to eat bananas in their summer causing an increase in demand and thus an increase in price. This is during the winter time of the banana growing countries. It should be noted that it appears the Windward Isles only starts following the trend of the other origins in January 2020 but this is only because of the lack of data for the years before.

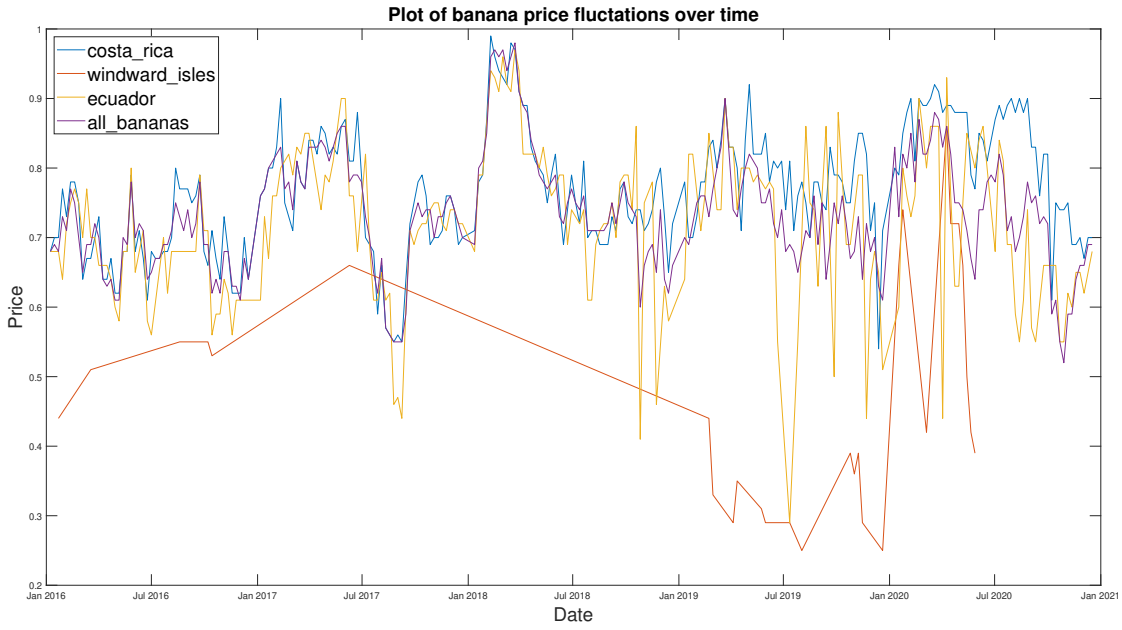

Figure 3: Graph showing the fluctuations of prices of bananas over the time interval 01/01/2016 to 31/12/2020.

5 Fast Fourier Transform

To really highlight the cyclical behaviour of the banana prices I decided to do a Fast Fourier Transform analysis of the 'all_bananas' origin, which had a data plot every week, over the span of 5 years. This was similar to the example done in the week 9 review session, and is documented on the Mathworks website [1], but I will be looking at weeks per cycle instead of years per cycle due to the structure of the data.

Listing 4: MATLAB code for fftdata

```
function [cycle] = fftdata()
dat = readtable('bananas-18jan21.csv');
absort = sortrows(dat(strcmp(dat.Origin, 'all_bananas'),:),'Date','
    ascend');
ab = absort(isbetween(absort.Date,datetime(2016,01,01),datetime
    (2020,12,31)),:);
y = fft(ab.Price); y(1)=[];
n = length(y);
power = abs(y(1:floor(n/2))).^2; % need first half of data due to
    symmetry
```

```
freq = (1:n/2)/(n/2)*1/2; % equally spaced frequency grid, 1/2 maximum
    frequency
cycle = 1./freq(find(power==max(power))); % cycle length, 1/freq=period
end
```

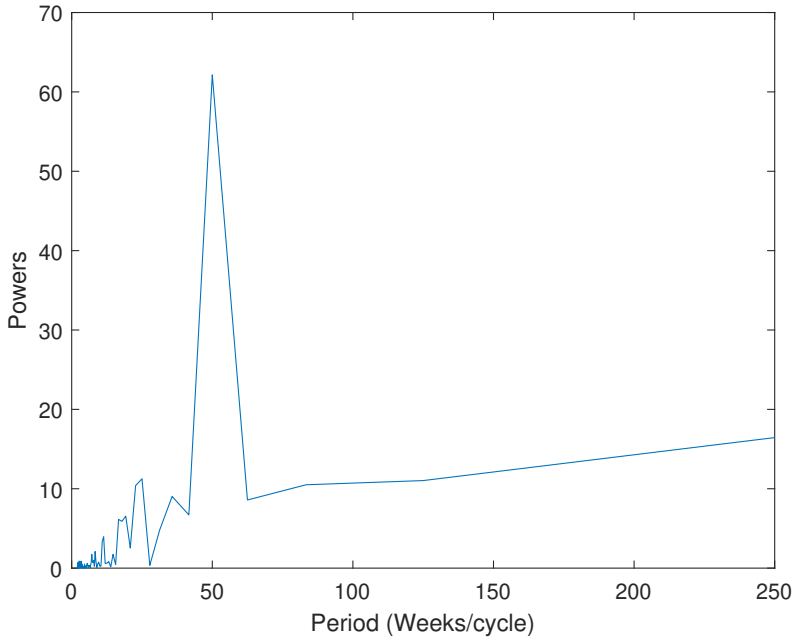

Figure 4: Plot of weeks per cycle vs power for the FFT.

In my code I found the cyclic behaviour to occur every 50 weeks, this is also clear from the graph. This shows that the banana prices follow the same pattern essentially every year.

References

[1] The MathWorks Inc. *Analyzing Cyclical Data with FFT*. 2021. URL: `https://uk.mathworks.com/help/matlab/math/using-fft.html` (visited on 05/03/2021).

[2] The MathWorks Inc. *Dates and Time*. 2021. URL: `https://uk.mathworks.com/help/matlab/date-and-time-operations.html` (visited on 05/03/2021).

[3] K Peterson. *Price Analysis of Bananas: Supply Demand and Elasticity*. 1997. URL: `http://eweb.furman.edu/~kpeterso/peterson/ecn22banana1.htm` (visited on 05/03/2021).

[4] Heather Y Wheeler. *South America*. 2015. URL: `http://www.naturalhistoryonthenet.com/Continents/south_america.htm.` (visited on 05/03/2021).