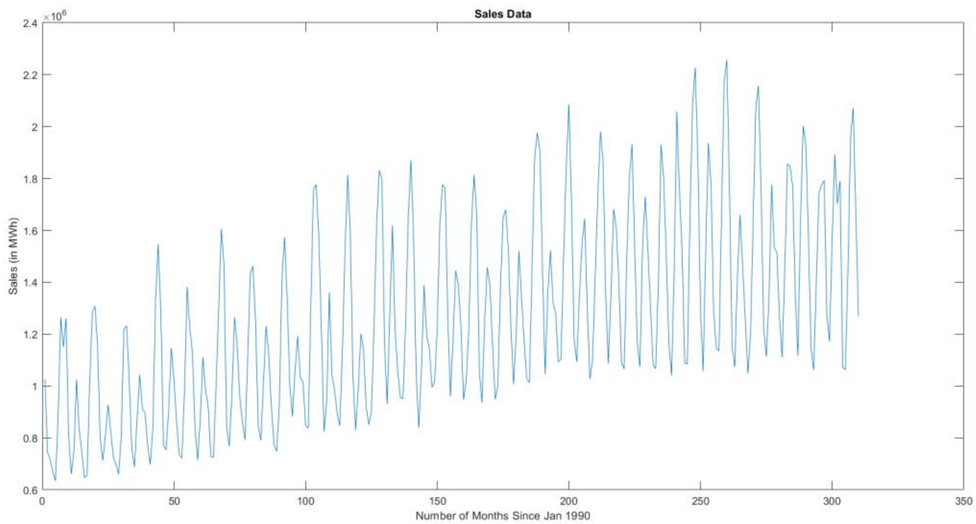
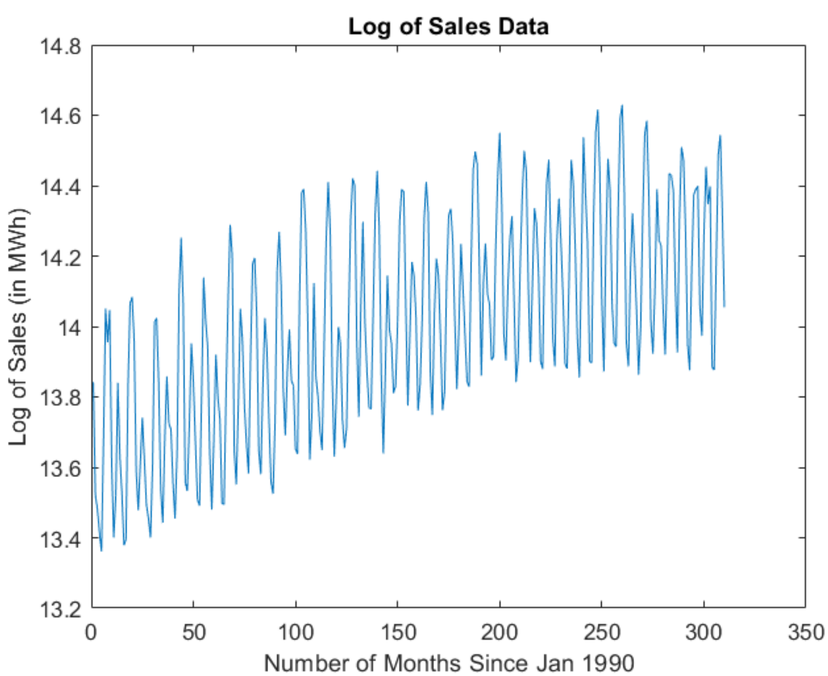
Task 2:

Below is a plot of the regular sales data. It is evident that the variance is increasing with time. Therefore, even if we removed the trend and seasonality, we would not have a weakly stationary time series.

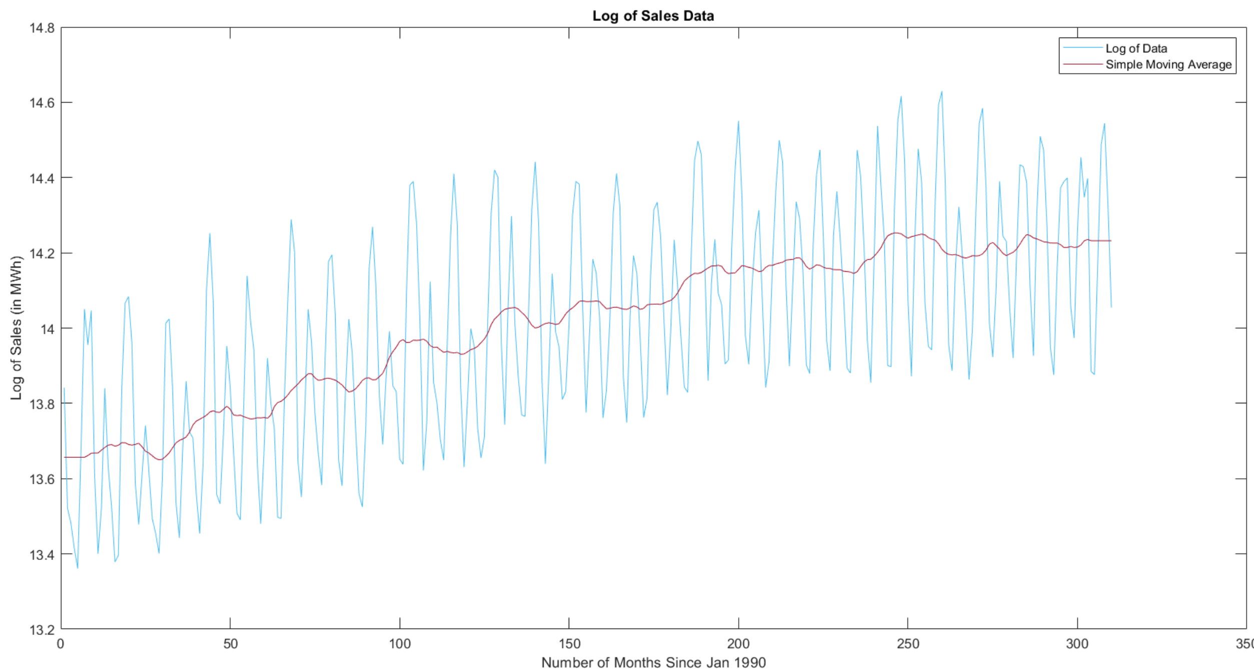


We plot the log of the data to check if the variance stays relatively constant:



This variance looks a lot better, and therefore we shall use it as the basis for our analysis.

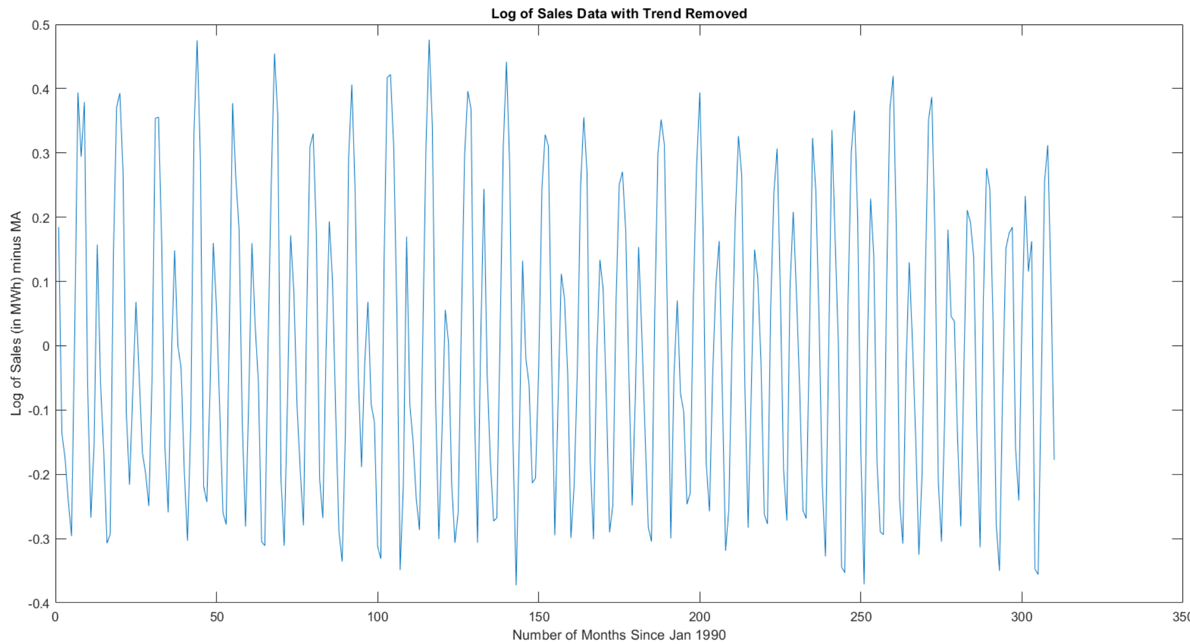
Task 3:



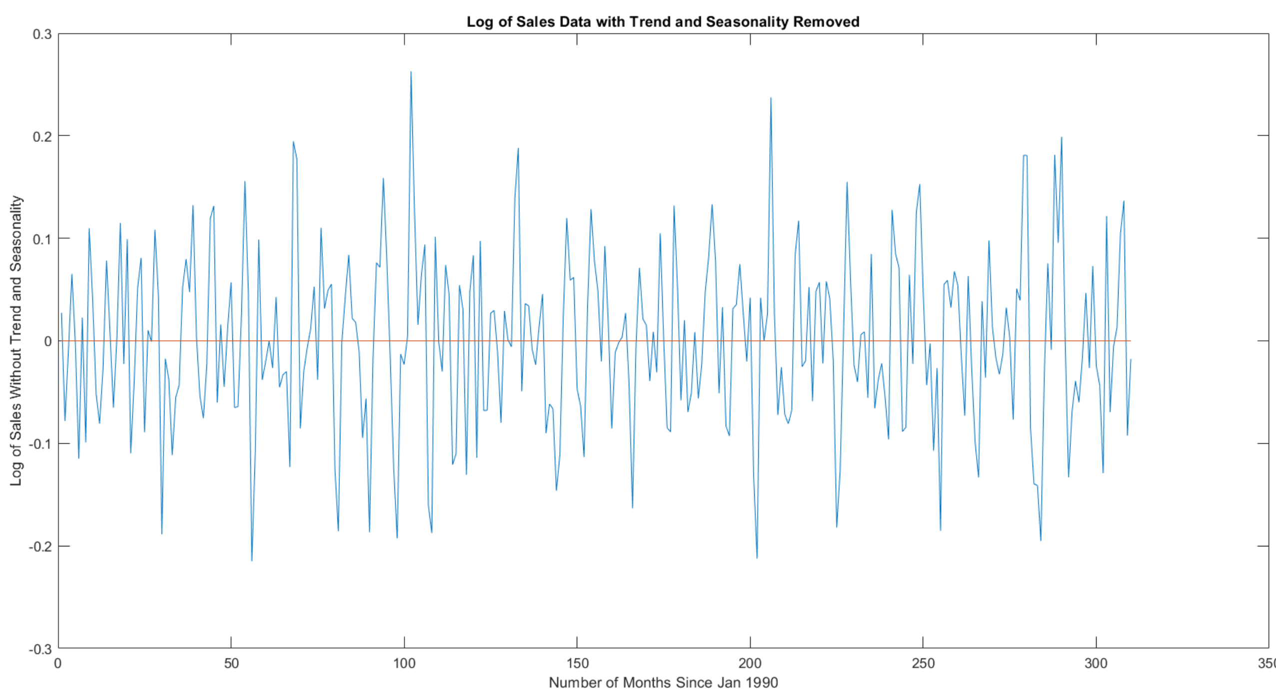
The red line is a 12- term simple moving average. It was calculated as the average of two asymmetric moving averages of length 11 (see code)

Task 4 AND Task 5

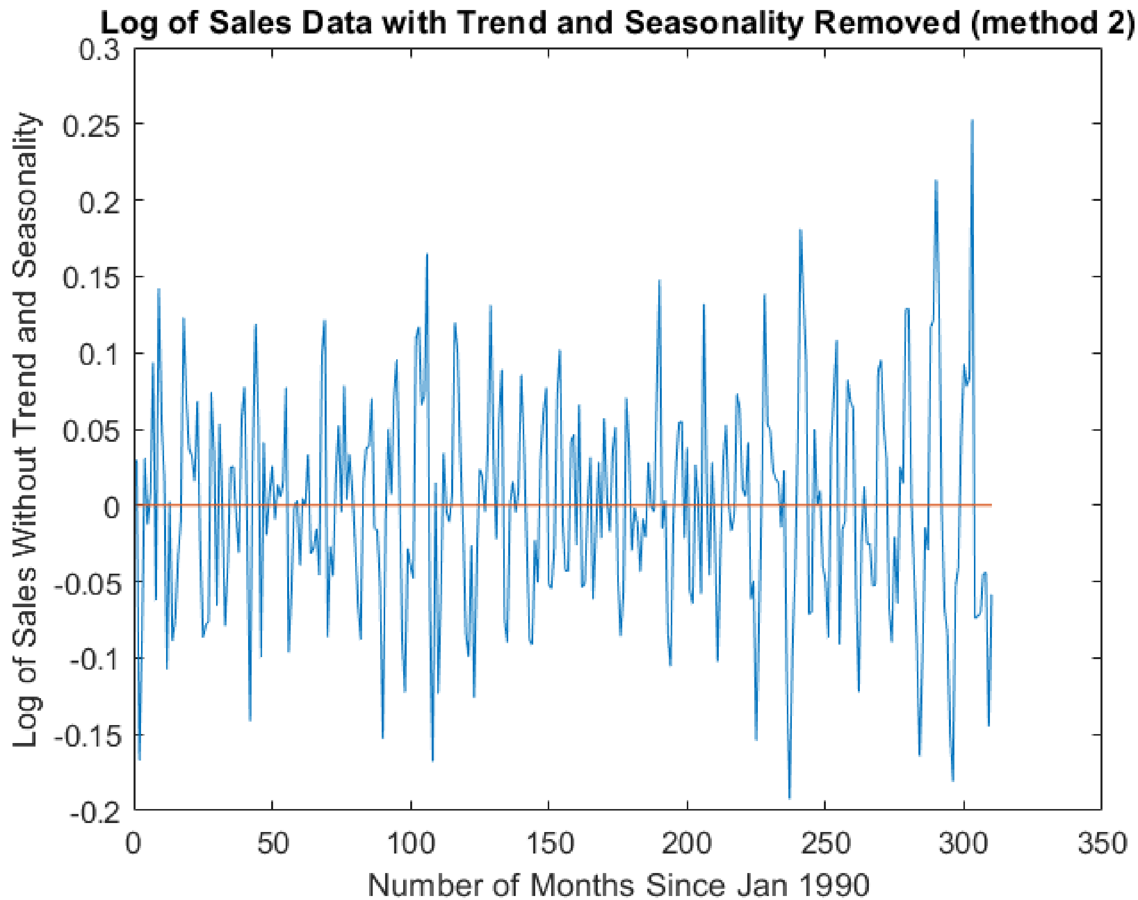
Below is a plot of the data with the trend removed:



Next, we remove the seasonality by using the equation seasonless (i) = trendless (i) – trendless (i – 12). Below is the plot:



This gives us a stationary time series, but does not give us the seasonal adjustments for our table. We therefore try a second method: take the average for each month and consider that to be the seasonal adjustment. Then subtract this average for the corresponding months from the original data to get the seasonless data. This procedure resulted in the following graph:



This appears to be stationary too, but looks a bit more wild compared to the previous method. The seasonal adjustments are:

Month	Seasonal Adjustment	Seasonal Adjustment (shifted so sum = 0)
January	14.1642	0.1550
Feb	14.0404	0.0312
March	13.9188	-0.0904

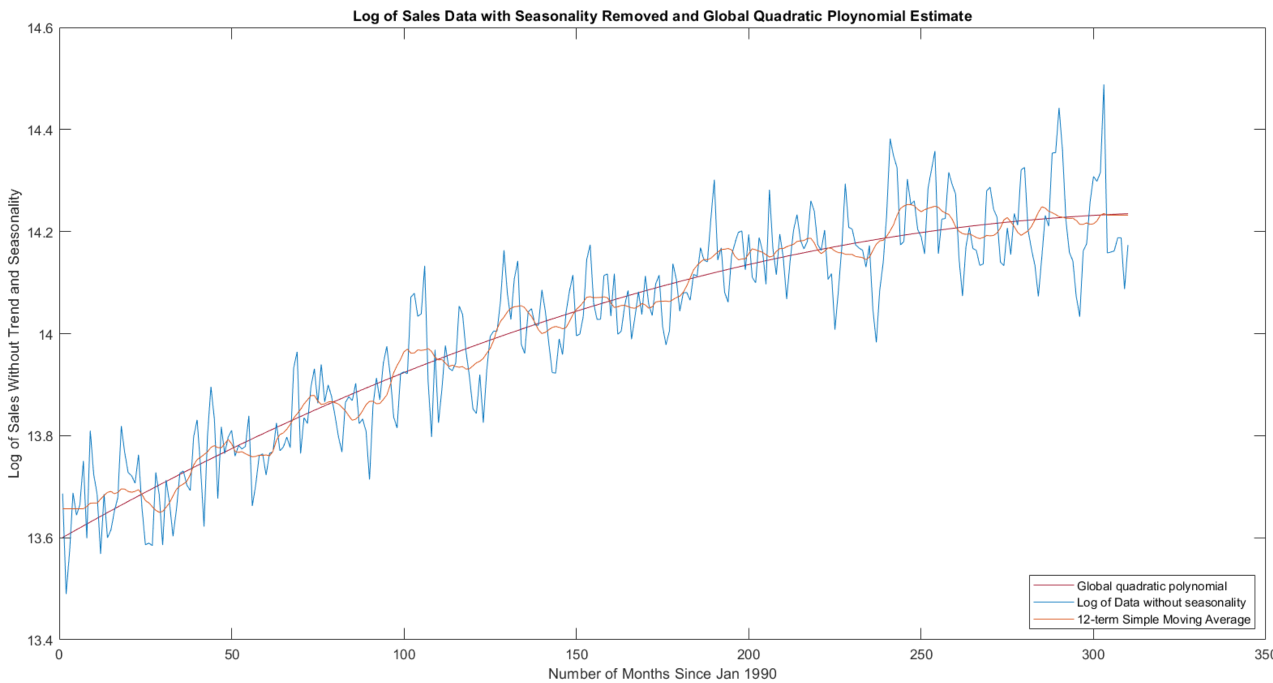
April	13.7361	-0.2731
May	13.7258	-0.2834
June	14.0267	0.0175
July	14.3095	0.3003
August	14.3655	0.3562
Sept	14.2461	0.2369
Oct	13.8902	-0.1190
Nov	13.7249	-0.2843
Dec	13.9622	-0.0470
Sum	168.11	0

Task 6:

We set M to be the nx3 Vandermonde matrix, and then calculate the coefficients of our quadratic using the fact that this matrix is full rank, i.e., coeff = (M'*M)^(-1) *(M'*Y), where M' is the transpose of M, and Y is our deseasonalized time series. The coeffiients we get are

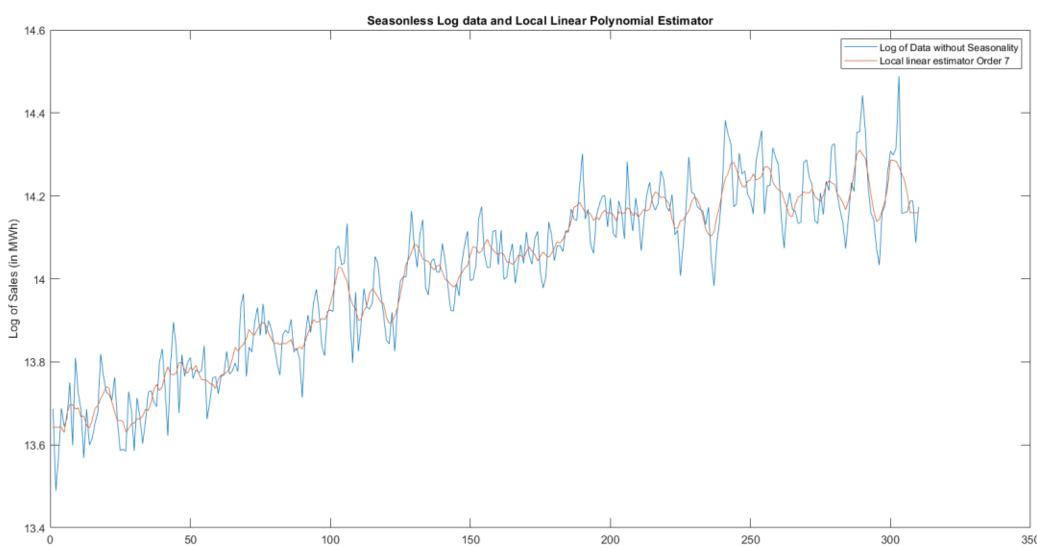
- alpha1 = 13.5966
- alpha2 = 0.0038
- alpha3 = -5.767e-06

We get the plot below:



The quadratic polynomial seems to be a smoothed version of the moving average, as it takes into account all data points, not just the points closest to the one in consideration. The global polynomial is therefore more useful in predicting future values.

Task 7



Farhad de Sousa 545 Data Assignment 1 Feb 27, 2023

We calculate the local polynomial estimator using the formula

$$[\beta_0, \beta_1] = (M^T * M)\text{^}(-1) * (M^T * X)$$

Where X = [Xt-3, Xt-2, …, Xt+2, Xt+3].

M = [1, -3; 1, -2; 1, -1; 1, 0; 1, 1; 1,2; 1,3]

We get $(M^T * M)^{-1} * (M^T) =$

0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429
-0.1071	-0.0714	-0.0357	0	0.0357	0.0714	0.1071

Therefore we just end up taking a moving average of order 7.

Compared to the moving average of order 12, this one has more variability, and follows the data more closely.

```
%% Plot the data and its log, to see which better fits the classical decomposition of ↙
a time series
figure(1)
plot(Sales)
title ('Sales Data')
xlabel('Number of Months Since Jan 1990')
ylabel('Sales (in MWh)')
logsales = log(Sales)
figure(2)
plot(logsales)
title ('Log of Sales Data')
xlabel('Number of Months Since Jan 1990')
ylabel('Log of Sales (in MWh)')
%% Calculate the moving average (for logsales)
movingavg1 = zeros(310,1);
movingavg2 = zeros(310,1);
movingavg = zeros(310,1); % we take the average of two asymmetric windows
for i = 7:(size(logsales)-6)
    sum = 0;
    for j = i - 5: i + 5
        sum = sum + logsales(j);
    end
    movingavg1 (i) = (sum + logsales(i - 6))/12;
    movingavg2 (i) = (sum + logsales(i + 6))/12;
end
movingavg = (movingavg1 + movingavg2)/2;
for i = 1:6
    movingavg(i) = movingavg(7);
    movingavg(length(logsales) + 1 - i) = movingavg(length(logsales)- 6);
end
%% Plot the log data with the moving average
p1 = plot(logsales);
title ('Log of Sales Data')
xlabel('Number of Months Since Jan 1990')
ylabel('Log of Sales (in MWh)')
legend('Log of Data')
hold on
p2 = plot(movingavg);
legend ([p1, p2],'Log of Data', 'Simple Moving Average')
%% Remove the trend from the data
trendless = logsales - movingavg
figure(2)
plot(trendless)
title ('Log of Sales Data with Trend Removed')
xlabel('Number of Months Since Jan 1990')
ylabel('Log of Sales (in MWh) minus MA')
%% Remove seasonality (Method 1)
seasonless = zeros(length(logsales));
```

```
for i = 13:length(logsales)
    seasonless(i) = trendless(i) - trendless(i-12);
end
for i = 1:12
    seasonless(i) = trendless(i) - trendless(i+12);
end
figure(3)
plot(seasonless)
title ('Log of Sales Data with Trend and Seasonality Removed')
xlabel('Number of Months Since Jan 1990')
ylabel('Log of Sales Without Trend and Seasonality')
%% Calculate seasonal adjustment and remove seasonality (Method 2)
seasonadj = zeros(12,1);
for i = 1:12
    monthsum = 0;
    count = 0;
    for j = i:12:310
        monthsum = monthsum + logsales(j);
        count = count + 1;
    end
    seasonadj(i) = monthsum/count;
end
seasonadj = seasonadj - mean(seasonadj)
newseasonless = zeros(length(logsales),1);
onlyseasonless = zeros(length(logsales),1);
for i = 1:12
    for j = i:12:310
        newseasonless(j) = trendless(j)-seasonadj(i);
        onlyseasonless(j) = logsales(j)-seasonadj(i);
    end
end
figure (4)
plot(newseasonless)
title ('Log of Sales Data with Trend and Seasonality Removed (method 2)')
xlabel('Number of Months Since Jan 1990')
ylabel('Log of Sales Without Trend and Seasonality')
%% Fitting a Global Polynomial to the seasonless data
M = zeros(length(logsales),3);
for i = 1:length(logsales) %build Vandermonde matrix
    M(i,1) = 1;
    M(i,2) = i;
    M(i,3) = i^2;
end
coeff = inv(M'* M) * M' * onlyseasonless

x = (1:0.1:length(logsales));
y = coeff(1) + coeff(2)*x + coeff(3)* x.^2;
figure (5)
p3 = plot(x,y);
```

```
hold on
p4 = plot(onlyseasonless);
ylim([13.4,14.6])
hold on
p5 = plot(movingavg)
title ('Log of Sales Data with Seasonality Removed and Global Quadratic Ploynomial
Estimate')
xlabel('Number of Months Since Jan 1990')
ylabel('Log of Sales Without Trend and Seasonality')
legend ([p3, p4, p5],'Global quadratic polynomial','Log of Data without seasonality',
'12-term Simple Moving Average')
%% Local Polynomial Estimate
A = zeros(7,2);
for i = 1:7 %build local Vandermonde matrix
    A(i,1) = 1;
    A(i,2) = i-4;
end
inv(A'* A)* A'
% Since the first row of the matrix is just 1/7 in each entry, we just need
% to take a moving average of order 7
movingavg7 = zeros(310,1); % we take the average of two asymmetric windows
for i = 4:(size(onlyseasonless)-3)
    sum = 0;
    for j = i - 3: i + 3
        sum = sum + onlyseasonless(j);
    end
    movingavg7 (i) = sum/7;
end
for i = 1:3
    movingavg7(i) = movingavg7(4);
    movingavg7(length(logsales) + 1 - i) = movingavg7(length(logsales)- 3);
end
p5 = plot(onlyseasonless);
title ('Seasonless Log data and Local Linear Polynomial Estimator')
xlabel('Number of Months Since Jan 1990')
ylabel('Log of Sales (in MWh)')
hold on
p6 = plot(movingavg7);
legend ([p5, p6],'Log of Data without Seasonality', 'Local linear estimator Order 7')
```