Name: Oswaldo Andres Celi Vega

ID: 43717921

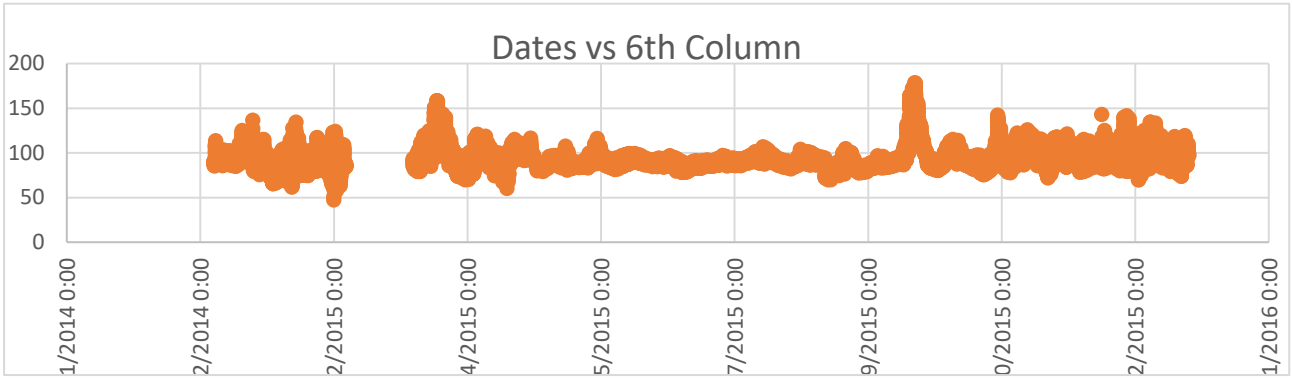
1.2)The data inside “Mystery.cvs” represents a whole year of data collected every 30 minutes.

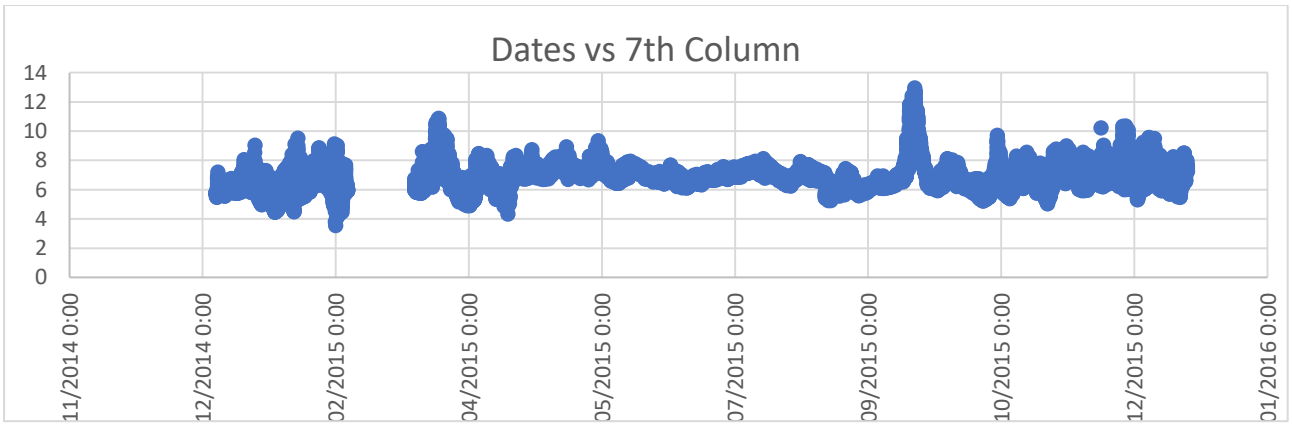
There are some empty dates though, such as:

- From 20/02/2015 to 15/03/2015

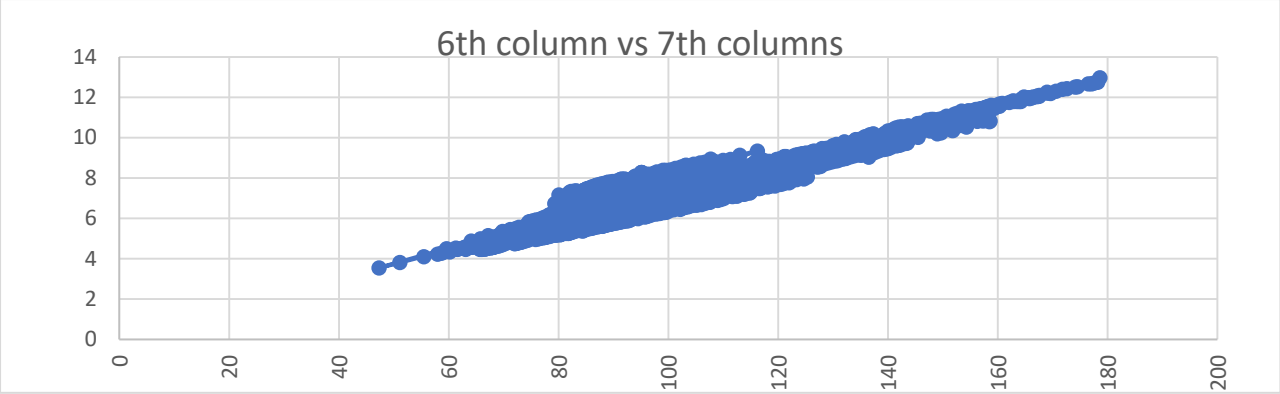
The second column of data ascends by units with each new entry. It might just be a count for the number of new entries obtained. Infact, after the break in entries the count in this column continues from where it left.

The columns 6th and 7th column seem very different. However, if plotted against the dates they show a correlation. As shown in figure 1 and 2.

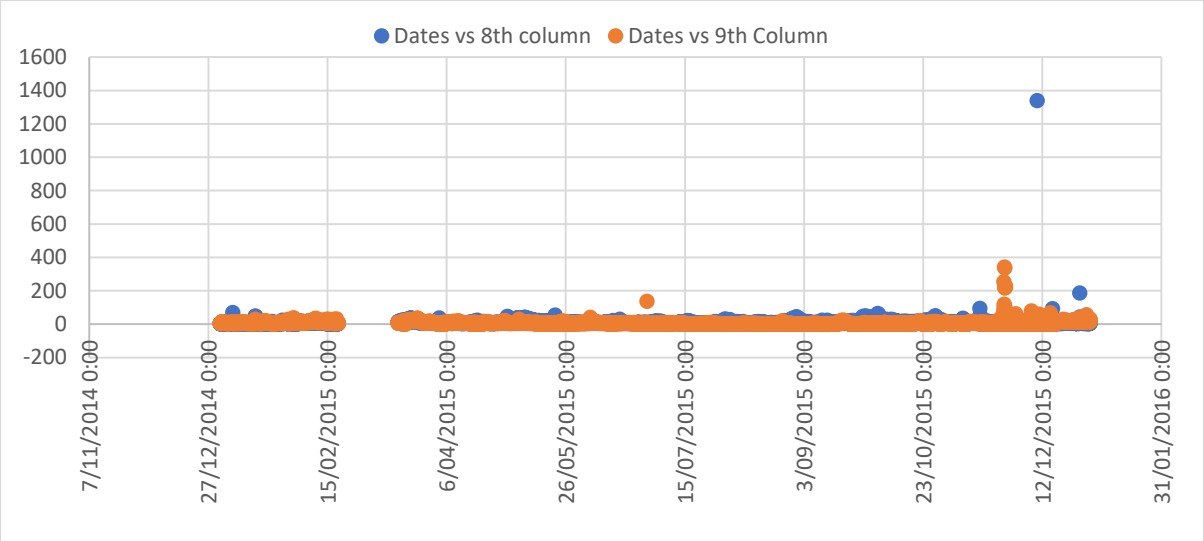




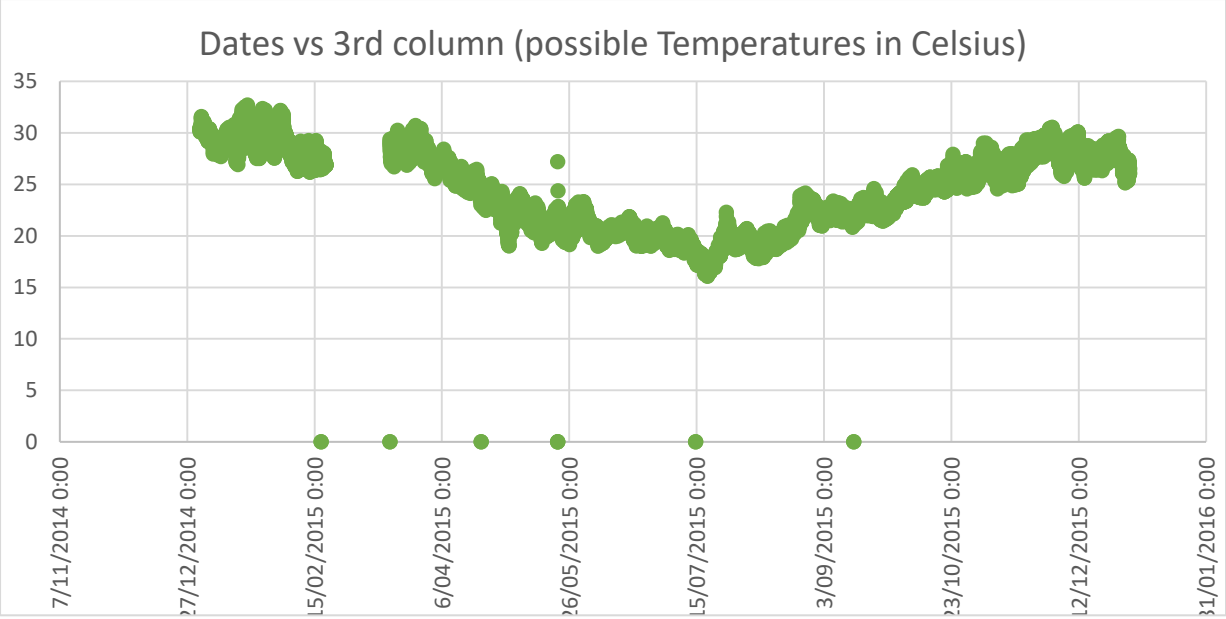
The shape of their distribution is the same and indeed if plotted against each other, they show a rather linear relationship. As demostrated in figure n.



The final colums of the data are hard to decipher. Moreover, towards the end of the year very high values are recorded which might actually be abberrant data.



Finally, the third column of data seems to represent temperature at any given point in time. In fact, they ressemble the temperatures of areas with latitudes similar to Cairns. As shown in the following figure



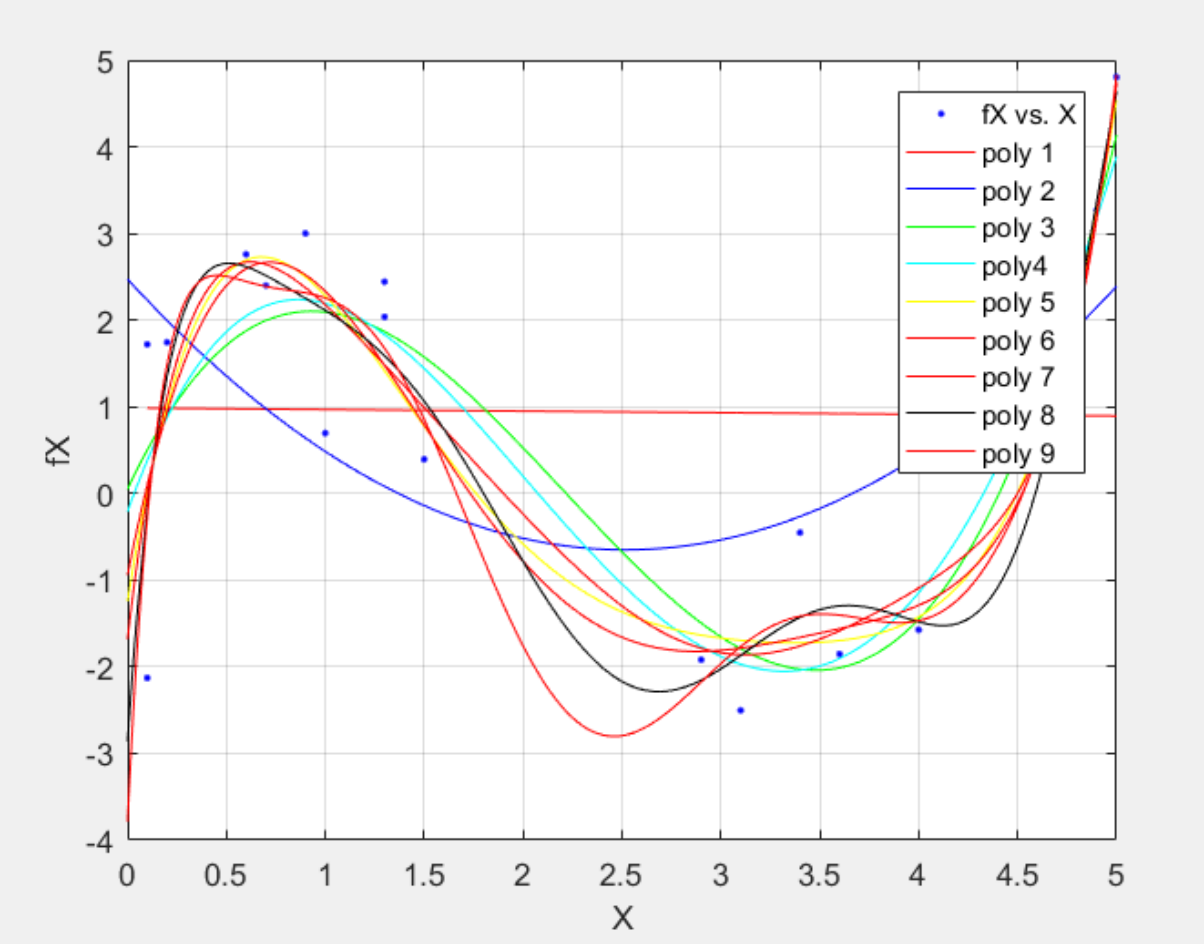
1.6)

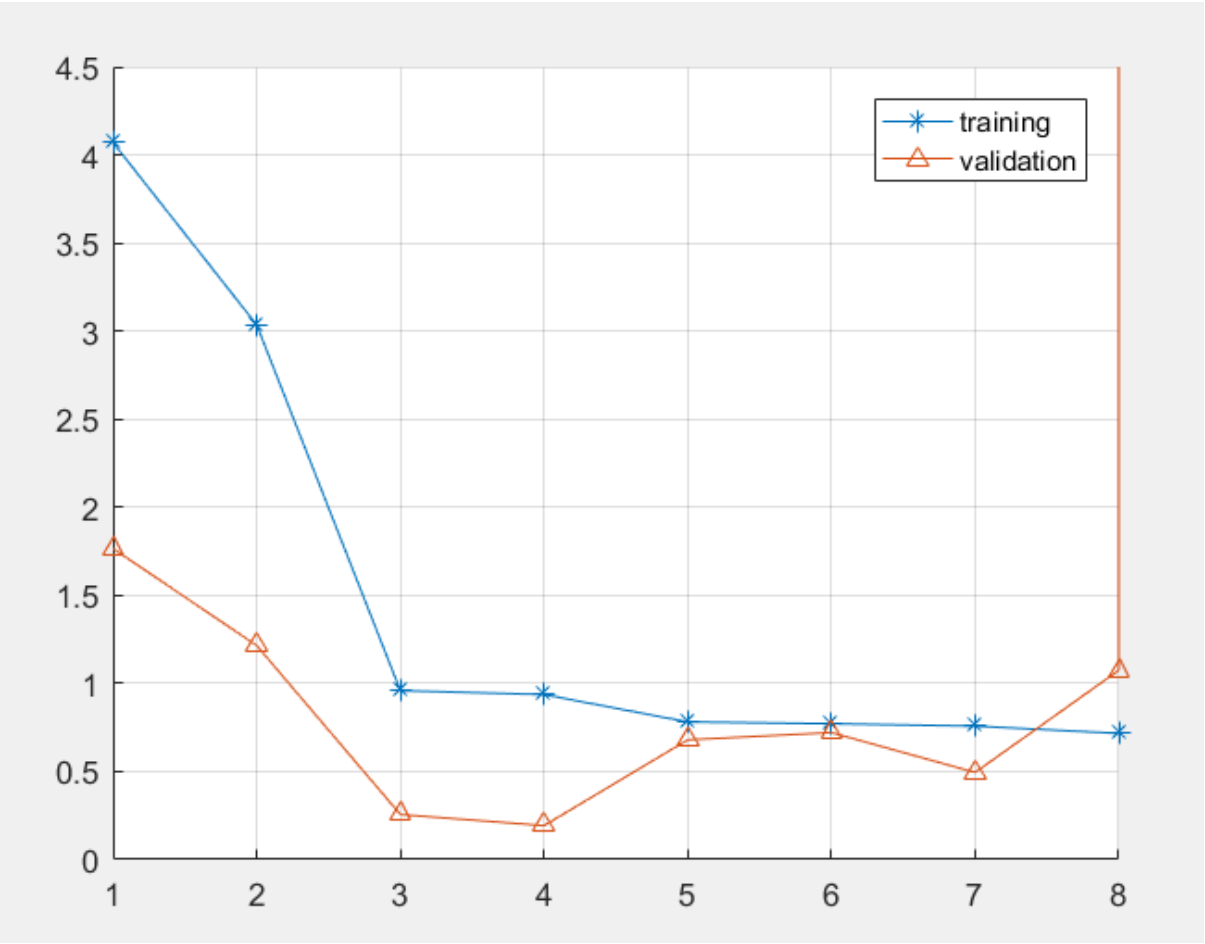
```
function [out] = prac1_6(in,n)
l = length(in);
parts = ceil(l/n);
out = [];
for i = 1:parts
    l = length(in);
    if l-n+1>=1
        new_part=in(l-n+1:l);
        in(l-n+1:l) = [];
    else
        new_part=in(1:l);
        in(1:l) = [];
    end
    out = [out,new_part];
end
end
```

```
>> [out] = prac1_6([1 2 3 4 5 6 7 8],3)

out =

     6     7     8     3     4     5     1     2
```

```
>> [out] = prac1_6([38 6 89 65 38 23 98 43 12],6)

out =

    65    38    23    98    43    12    38     6    89

>> [out] = prac1_6([38 6 89 65 38 23 98 43 12],1)

out =

    12    43    98    23    38    65    89     6    38
```

2.1) In the following 3 Figures is possible to see the behaviour of the polynomials as well as the error produced by each polynomial degree in fitting the given data.

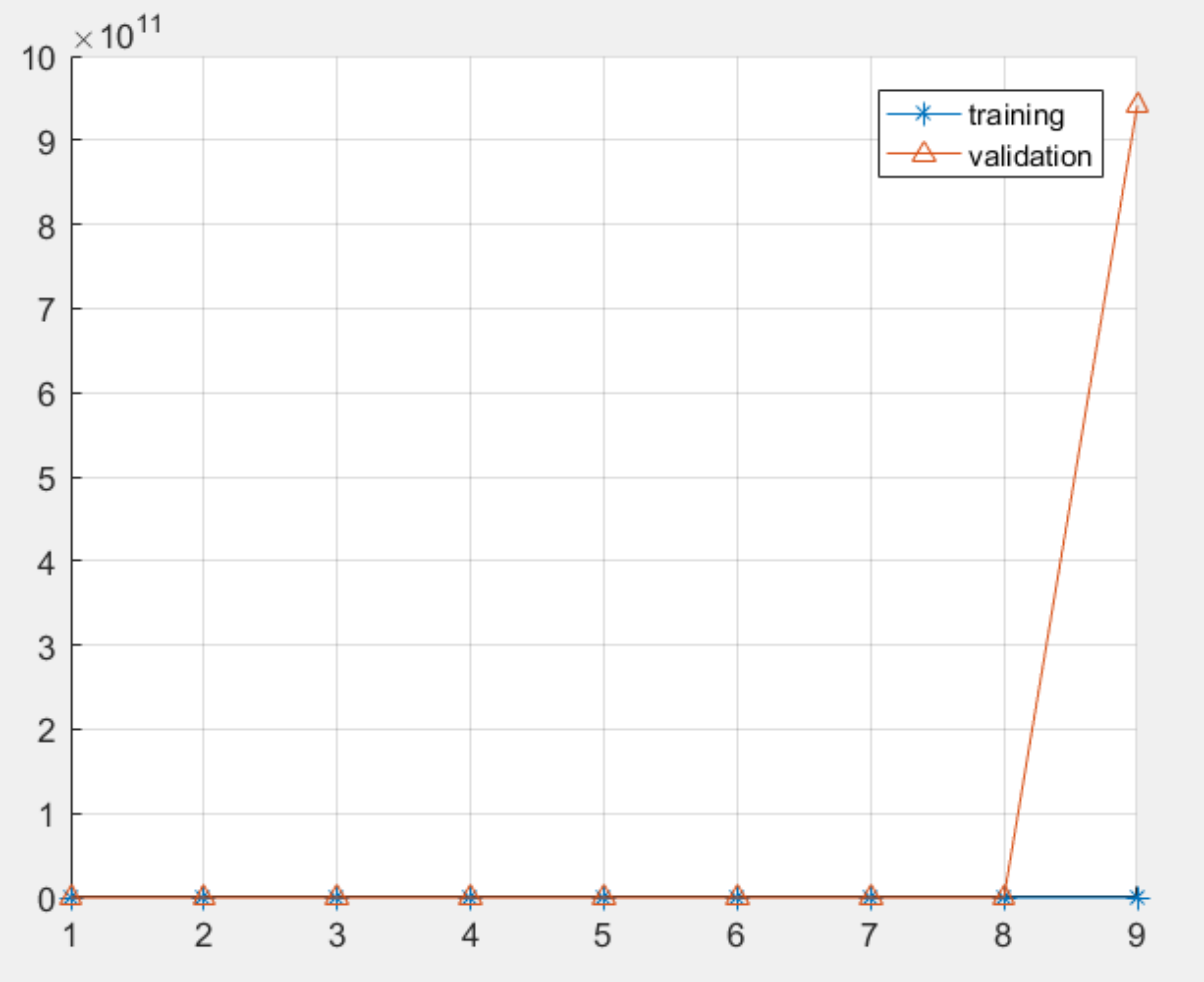




The previous plot does not include the 9th polynomial function since the error at that point spikes and makes impossible to appreciate the scale of the other points.

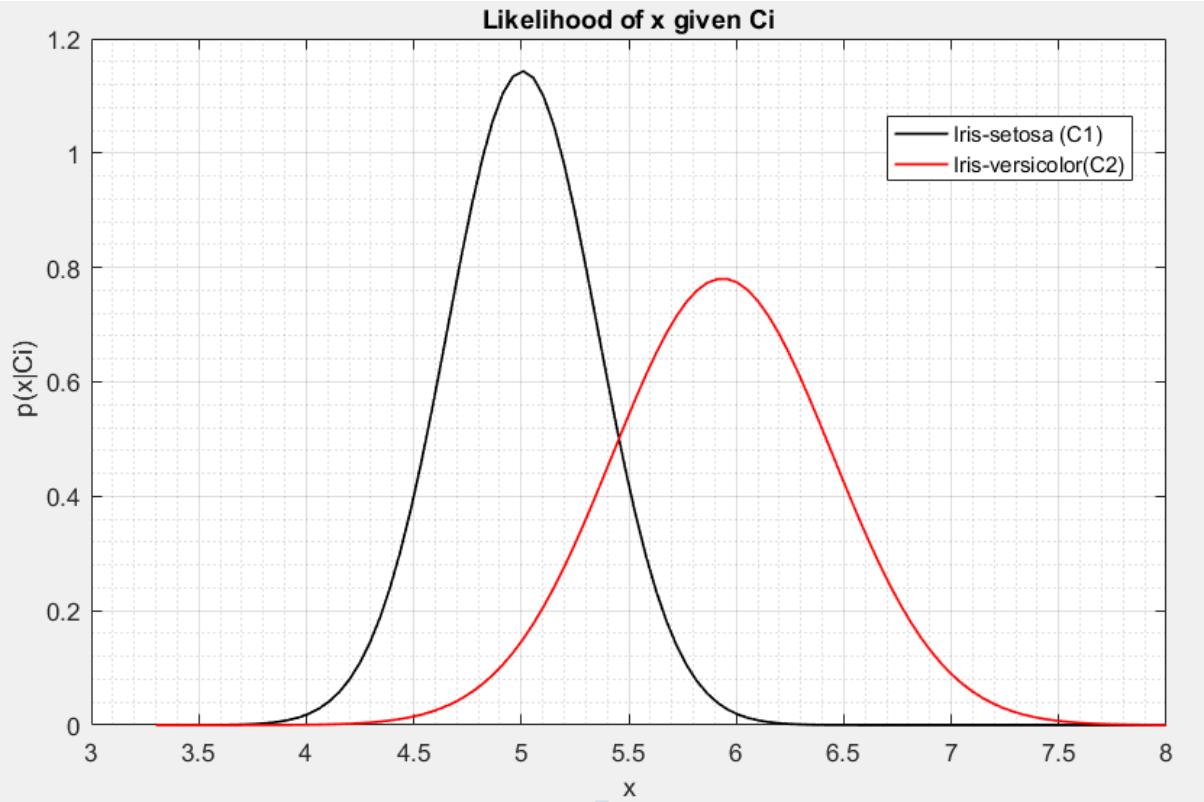
While using Matlab to create the previous graphs, the variance for the noise applied was 1. While, the value used in the book is 0.1. Therefore, overfitting is a catalyst for the increase of the variance and if the variance in the data is high then the error will increase in an exponential manner.

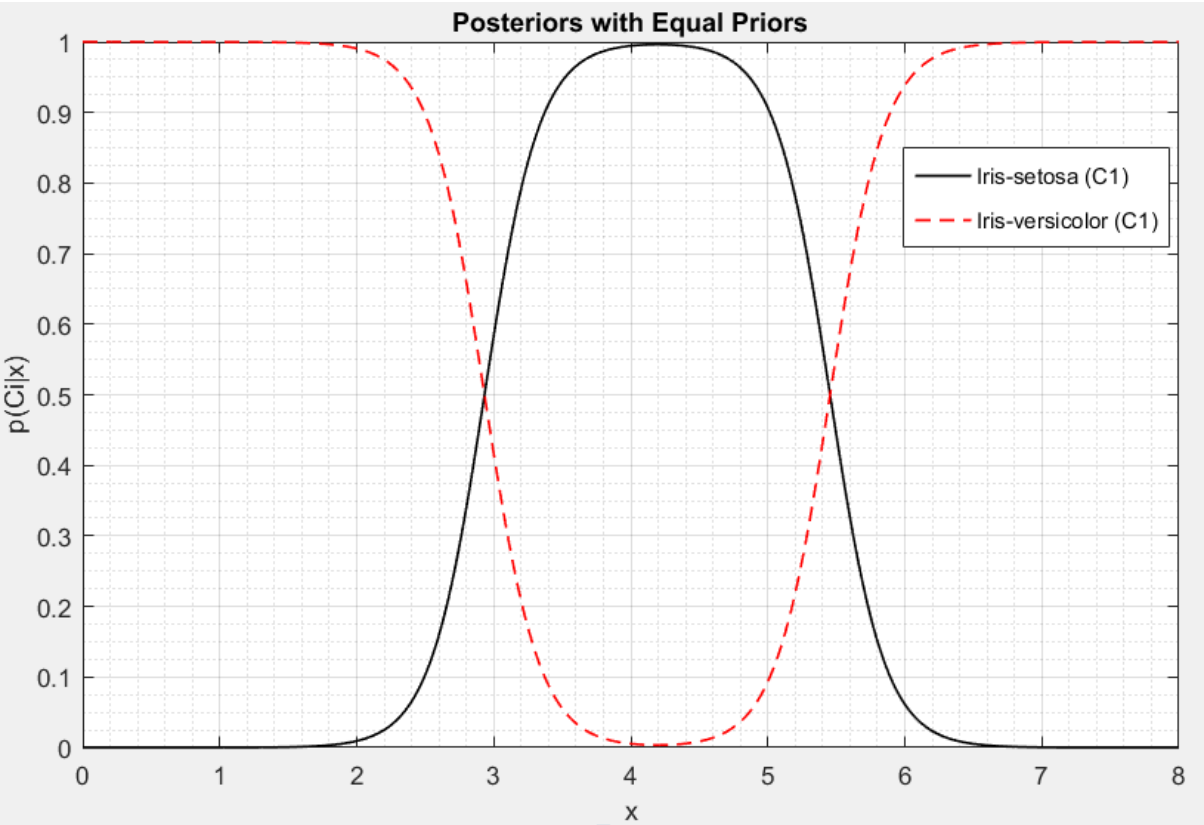
Finally, reproducing this result was troublesome since running randn() different times produces of course new random variables each time.

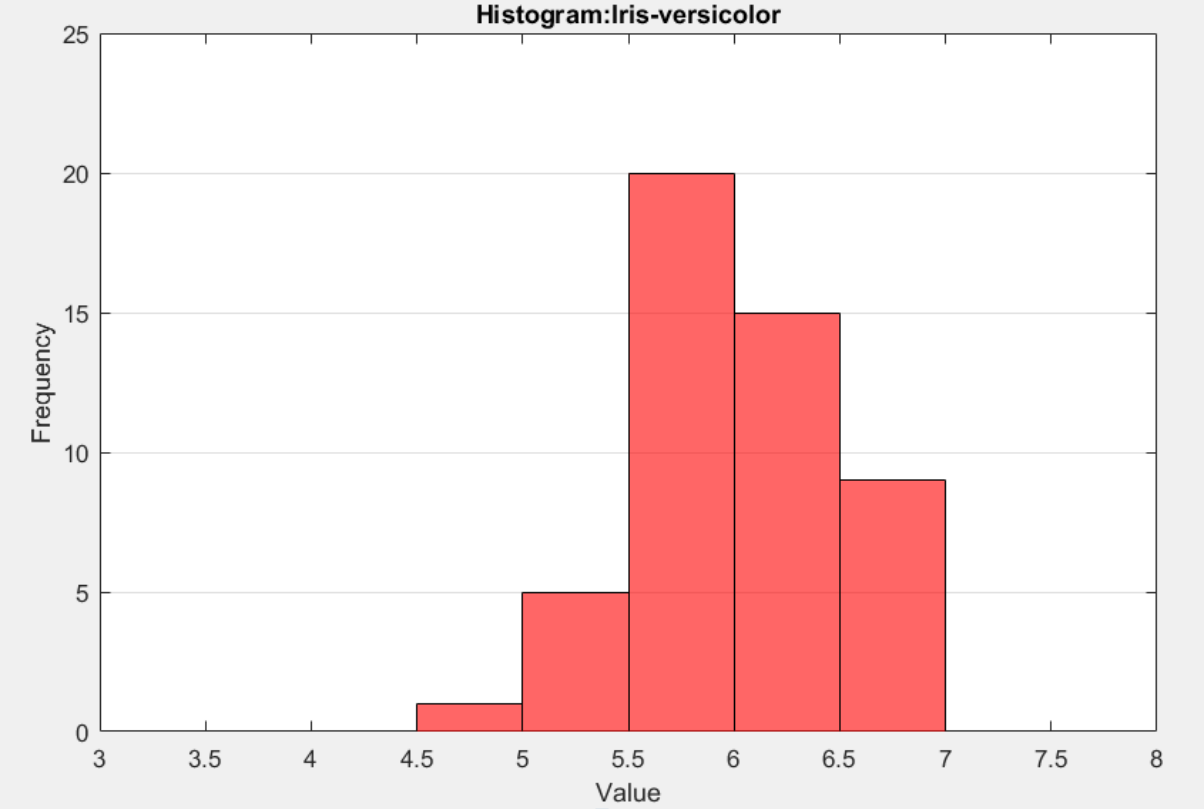


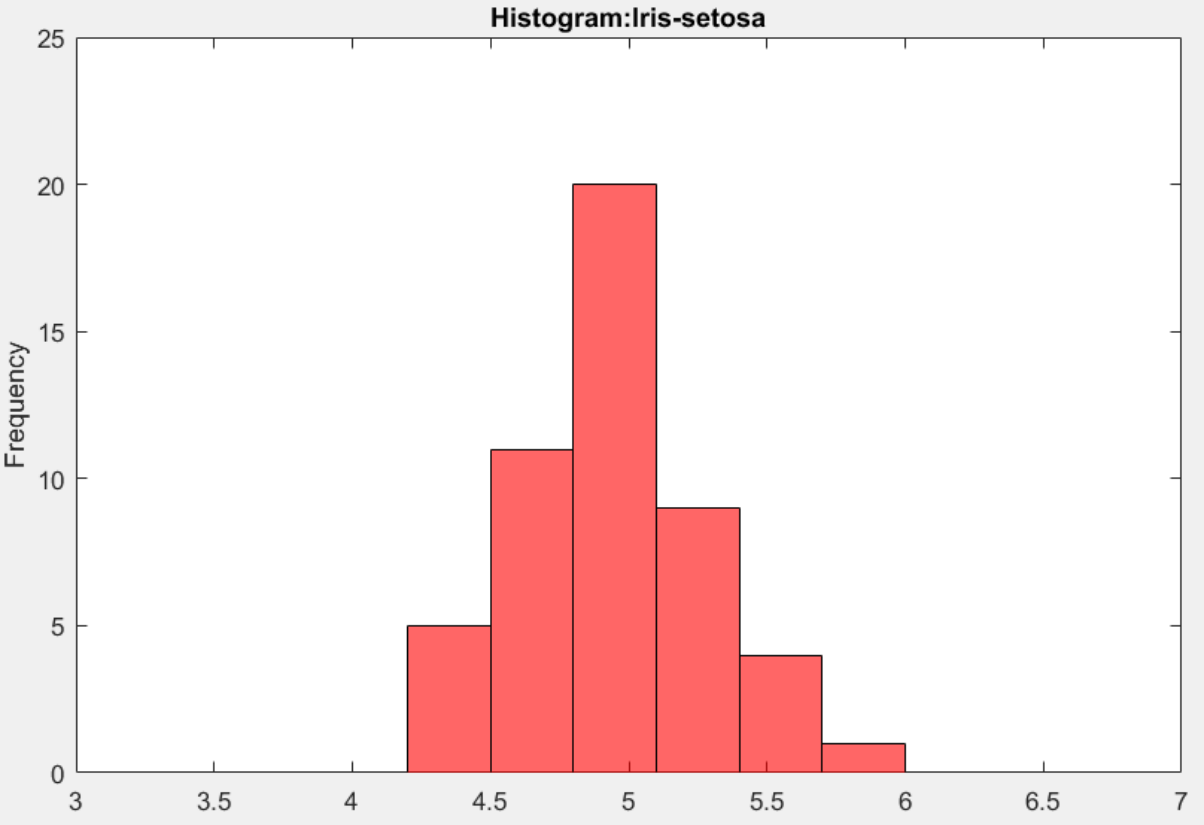
In here it is possible to observe the effects of overfitting. The plot demonstrate that the accuracy of the polynomial regression decreases after a certain amount of degrees.

2.4)


Likelihood of x given Ci
Iris-setosa (C1)
Iris-versicolor(C2)
p(x|Ci)
x
1.2
1
0.8
0.6
0.4
0.2
0
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8


Posteriors with Equal Priors
Iris-setosa (C1)
Iris-versicolor (C1)
p(Ci|x)
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8


Histogram:Iris-versicolor
Frequency
Value
0
5
10
15
20
25
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8


Histogram:Iris-setosa
Frequency
0
5
10
15
20
25
3
3.5
4
4.5
5
5.5
6
6.5
7

3.1)

Split sets and creat P(Ci)s:

```
%% Create Train matrix
Train = [];
Train = pimaindiansdiabetes(1:500,1:9);
%% Create a Test Matrix
Test = [];
Test = pimaindiansdiabetes(501:768,1:9);
%% Calculate probability of C1
P_C1 = sum(Train(:,9))/length(Train);
%% Calculate probability of C2
P_C0 = (length(Train)-sum(Train(:,9)))/length(Train);
```

```
P_C1 =

    0.3640
```

Prior P(C1)

```
P_C0 =

    0.6360
```

Prior(C0)

a) Error training set:

```
errorTest =

   25.2000
```

```
errorTest =

   22.0149
```

b) Error test set

c) Covariance of set1

```
S1 =

   1.0e+04 *

    0.0014    0.0008    0.0010   -0.0005   -0.0029   -0.0003   -0.0000    0.0018
    0.0008    0.0963    0.0039    0.0003    0.1129    0.0006    0.0001    0.0060
    0.0010    0.0039    0.0491    0.0078    0.0269    0.0007   -0.0000    0.0059
   -0.0005    0.0003    0.0078    0.0295    0.1232    0.0035    0.0002   -0.0030
   -0.0029    0.1129    0.0269    0.1232    1.9654    0.0031    0.0005    0.0162
   -0.0003    0.0006    0.0007    0.0035    0.0031    0.0056    0.0000   -0.0017
   -0.0000    0.0001   -0.0000    0.0002    0.0005    0.0000    0.0000   -0.0000
    0.0018    0.0060    0.0059   -0.0030    0.0162   -0.0017   -0.0000    0.0114
```

Mean of set 1

```
m1 =

    4.7802
  140.4890
   69.7253
   21.7143
  102.4286
   35.3231
    0.5672
   36.2692
```

Covariance set 0

```
S0 =

1.0e+04 *

 0.0009    0.0011    0.0006   -0.0004   -0.0033    0.0001   -0.0000    0.0019
 0.0011    0.0773    0.0072   -0.0002    0.1058    0.0038    0.0001    0.0098
 0.0006    0.0072    0.0311    0.0050    0.0141    0.0056    0.0000    0.0037
-0.0004   -0.0002    0.0050    0.0217    0.0611    0.0055    0.0000   -0.0025
-0.0033    0.1058    0.0141    0.0611    1.0829    0.0242    0.0009   -0.0166
 0.0001    0.0038    0.0056    0.0055    0.0242    0.0063    0.0000    0.0007
-0.0000    0.0001    0.0000    0.0000    0.0009    0.0000    0.0000    0.0000
 0.0019    0.0098    0.0037   -0.0025   -0.0166    0.0007    0.0000    0.0138
```

Mean set 0

```
m0 =

    3.2642
  110.5189
   68.2107
   19.9717
   68.1447
   30.0780
    0.4522
   31.2956
```

3.2)Shared covariance matrix

```
S10 =

   1.0e+04 *

    0.0011    0.0010    0.0007   -0.0004   -0.0032   -0.0001   -0.0000    0.0019
    0.0010    0.0842    0.0060   -0.0000    0.1084    0.0026    0.0001    0.0084
    0.0007    0.0060    0.0376    0.0060    0.0188    0.0039   -0.0000    0.0045
   -0.0004   -0.0000    0.0060    0.0246    0.0837    0.0048    0.0001   -0.0027
   -0.0032    0.1084    0.0188    0.0837    1.4042    0.0165    0.0008   -0.0047
   -0.0001    0.0026    0.0039    0.0048    0.0165    0.0061    0.0000   -0.0002
   -0.0000    0.0001   -0.0000    0.0001    0.0008    0.0000    0.0000    0.0000
    0.0019    0.0084    0.0045   -0.0027   -0.0047   -0.0002    0.0000    0.0129

errorSTrain =

   25.6000
```

shared covariance error for the Train set

```
errorSTest =

   19.4030
```

shared covariance error for test set

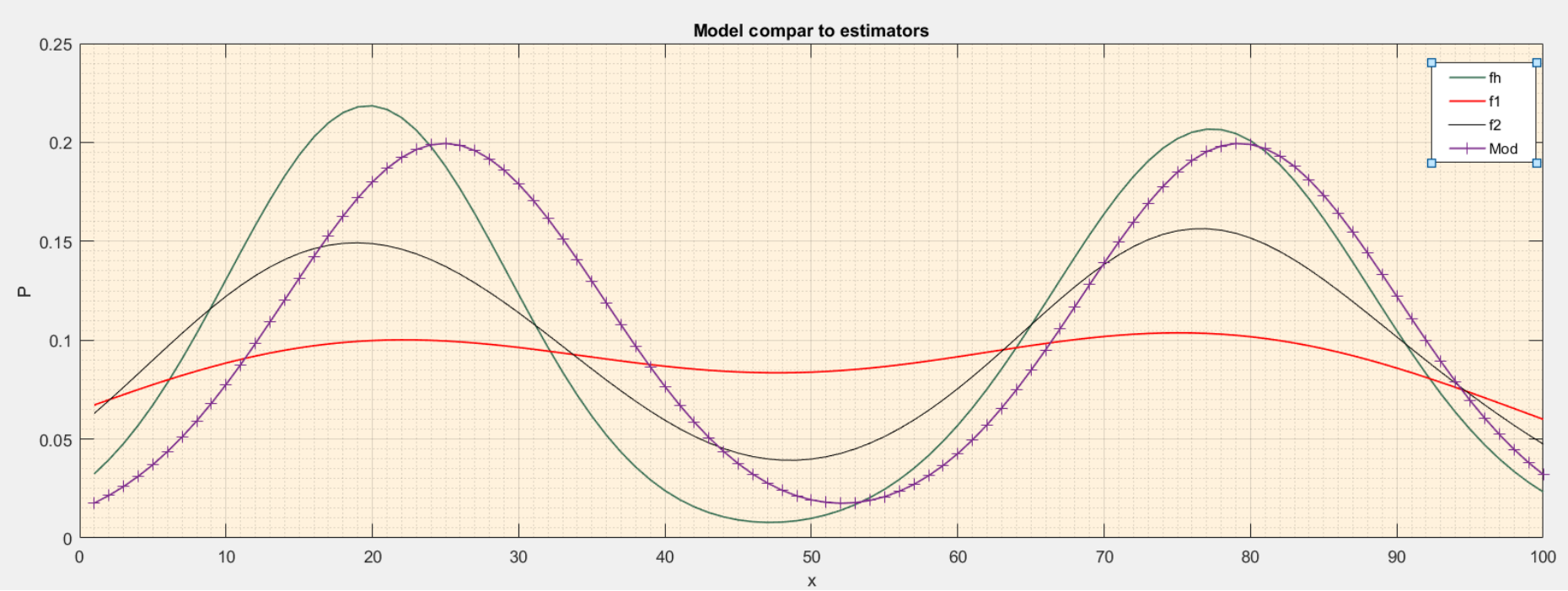
3.5)

```
X=[randn(30,1);5+randn(30,1)];
H1=histogram(X,20,'Normalization','pdf');
[H12,LIMS]=hist(X,20);
H12=H12/max(H12);
%Gaussian Mixture Model
GaussMixMod = fitgmdist(X,2);
%test set
test= linspace(min(X),max(X),100);
test=transpose(test);
 %parameters of first gaussian
m_1=0;
sigma_1=1;
 %parameters of second gaussian
m_2=5;
sigma_2=1;
%density function for first gaussian
P1 = exp(-(test-m_1).^2./(2*sigma_1^2))./(sigma_1*sqrt(2*pi));
 %density function for second gaussian
P2 = exp(-(test-m_2).^2./(2*sigma_2^2))./(sigma_2*sqrt(2*pi));

Mod=0.5*P1+0.5*P2;

fh=pdf(GaussMixMod,test);
[f1,xk1,width1]= ksdensity(X,test);
[f2,xk2,width2]=ksdensity(X,test,'width',0.5*width1);
```

```
%% Calculate KL divergences

count=0;
for n=1:length(Mod)
    contri=fh(n)*log(fh(n)/Mod(n));
    count=count+contri;
end
KL1=count;

count=0;
for n=1:length(Mod)
    contri=f1(n)*log(f1(n)/Mod(n));
    count=count+contri;
end
KL2=count;

count=0;
for n=1:length(Mod)
    contri=f2(n)*log(f2(n)/Mod(n));
    count=count+contri;
end
KL3=count;
```

```
figure
plot(fh');hold on
plot(f1');
plot(f2');
plot(Mod');
legend('fh','f1','f2','Mod');
grid on; grid minor;
xlabel('x');
ylabel('P');
title('Model compared to estimators');
```

The smaller the bandwidth of the kernel, the higher and closer the approximation will get to the model. In the following graph is possible to see an approximation to the distribution by histogram as well as different kernel types.


Model compar to estimators
fh
f1
f2
Mod
P
x
0.25
0.2
0.15
0.1
0.05
0
0
10
20
30
40
50
60
70
80
90
100