FCI Fall 2013 Assignment V. 50 pts.
NAME(s): **Francisco Nardi** (16163892) and **Paulo Silva** (16164638).

Electronic submission on Blackboard is due latest by 11 pm on Monday, Oct 21st. Submissions received after the deadline will be graded only for effort for a maximum of 70% of the total grade (Refer to class syllabus for detailed grading policy). **State any assumptions you make, justify your answers, show intermediate steps and explain your results for maximum credit**. You may leave quantitative answers in the form of expressions unless a numeric value is required to address the question. All answers should be in your own words with any sources you refer to cited at the appropriate places. Any knowledge you acquire from the Internet should be written in your own words and be appropriately referenced. Copying and pasting from the Internet, each other or any other source will not count as your effort (Refer to class syllabus for detailed policy on plagiarism).
You may submit this assignment in groups of two each. Write your names on this sheet and include it as the cover page for your submission. You are required to use MATLAB for this assignment. You may base your code on the samples provided on the textbook website (http://www.dcs.gla.ac.uk/~srogers/firstcourseml/); clearly indicate which sections of your answer are not original.

Q1. (25) Modify the MATLAB script olympbayes.m to predict the 2016 winning time distribution (figure similar to 3.23) for your favorite Olympic event. How does this change by changing the Gaussian prior or sigma?
Q2. (25) Create an artificial noisy dataset from a pre-specified polynomial regression model for an imaginary Olympic event (e.g., time t taken to eat x Ostrich eggs). Use your model and a customized version of the MATLAB script margpoly.m to recreate figures like 3.24 and 3.25.

Homework 5
Francisco Nardi and Paulo Silva

01.

<u>Input:</u>

‘100mWomen.txt’.

<u>Output:</u>

‘Predicted winning time: 10.5401
New variance: 0.0621005

Let us change the prior for 100 and 100
Predicted winning time: 10.5506
New variance: 0.0621005

1-We realized just a tiny difference in the pred. win. time

Let us change the covariance matrix for 200 and 200
Predicted winning time: 10.5398
New variance: 0.0621008

2-Again, just a tiny difference

Now, let us change the variance for 4
Predicted winning time: 10.5995
New variance: 4.96467’

3-We finally observed relevant changes (shape and time).’

<u>Observation:</u>

Using the formulas provided by the book and the implementation provided by the website, we could reach a conclusion that, as there is a lot of data, change in the prior would not affect the predicted winning time. But when we changed the variance, the percentage chance varied, as well as the shape.

02.

Input:

‘Ostrich.txt’.

Output:

‘Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 1.146727e-016.
> In hw52FranciscoPaulo at 44
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 7.168988e-017.
> In hw52FranciscoPaulo at 61’

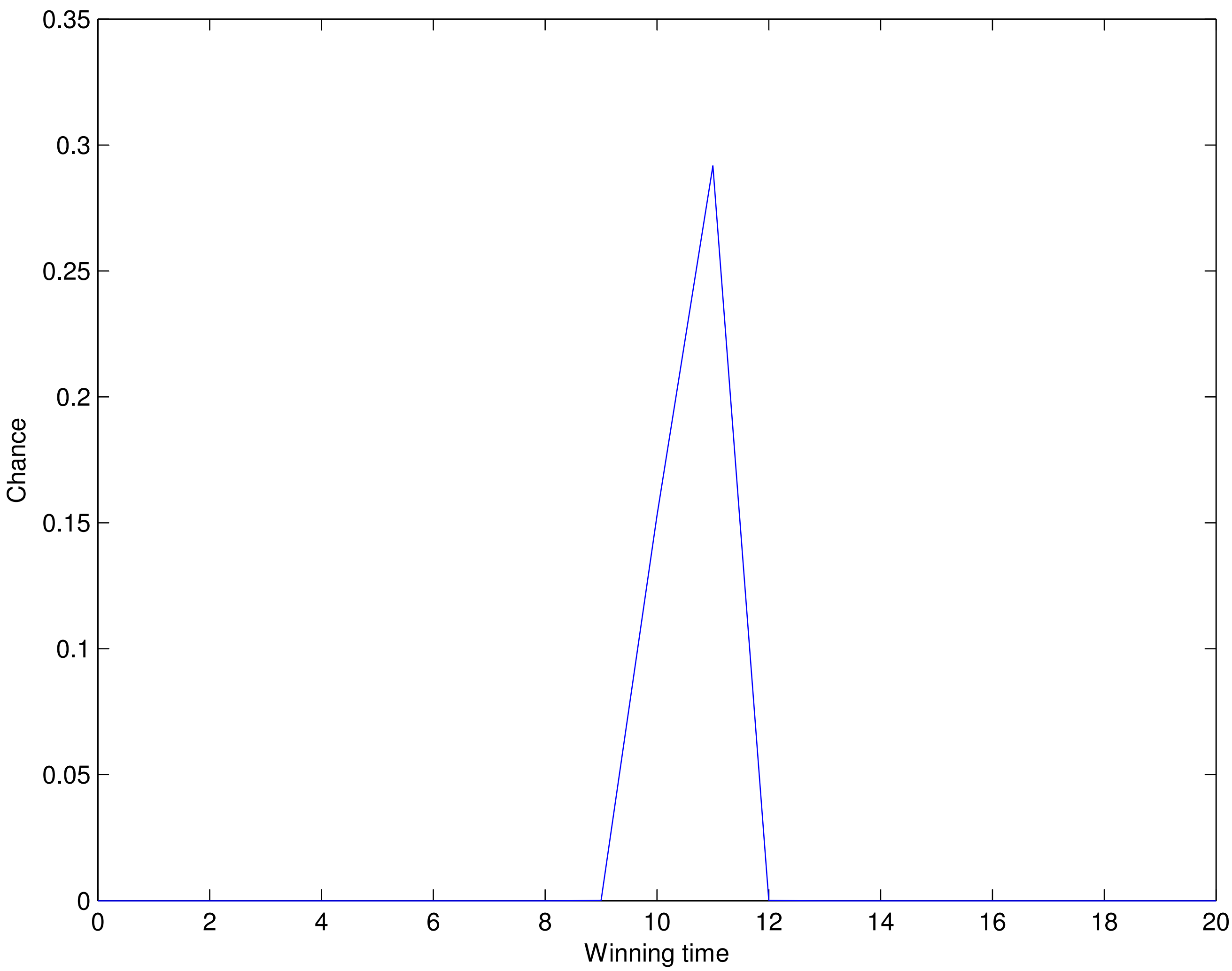
Observation:

Using the formulas provided by the book and the implementation provided by the website, we could reach a conclusion that, changing the variance changed the best model selected. Using noise = 10, the best was model of order 7 (over fit). Using noise = 200, the best was model of order 6 (under fit).

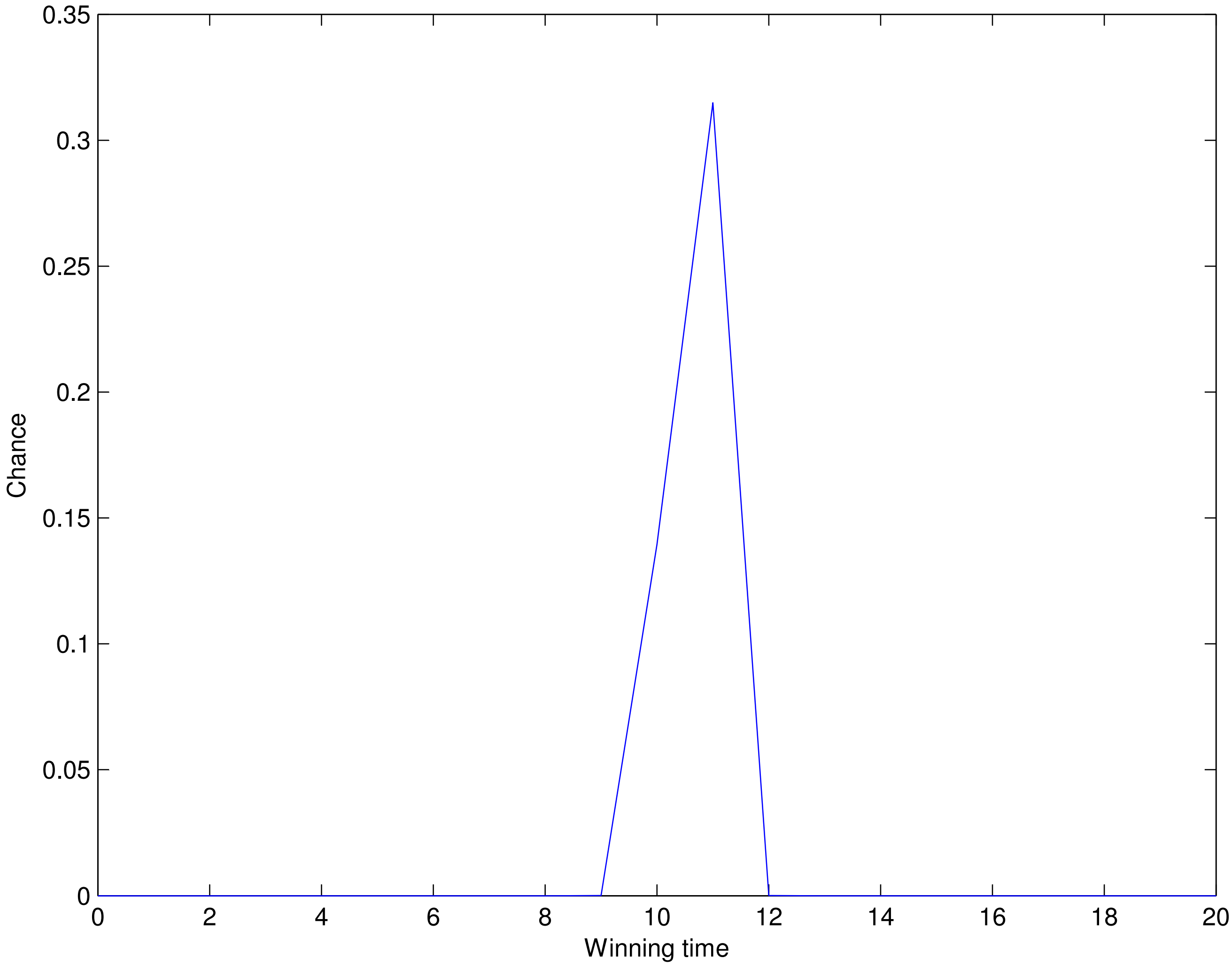
```
% Homework 5.1
% Francisco Nardi and Paulo Silva

% Cleaning the screen and variables as usual
clear all;
close all;

% Here we load the data from the file
% Olympic Women 100m
dataset=importdata('100mWomen.txt');

% x will be the years
% t will be the winning times
x = dataset(:,1);
t = dataset(:,2);

% xdata will be used to calculate the prediction
xdata = x;

% here the events become a sequence starting in 0
% until "the last year", with pace of four years [0 4 8 ...]
x = x - x(1);

% here we divide the sequence by four, turning it into a sequence
% [1 2 3 ...], which mean as well that the graph can be better exhibited
x = x./4;

X = [x.^0 x.^1];

% the questions asks for the prediction for the year of 2016
pred = 2016;

% based on the previous idea, we rescale the year which is going
% to have the winning time predicted
pred = pred - xdata(1);
pred = pred./4;


% here are the parameters used in the calculation of the predicted time
% by changing them, we perceive possible changes in the results of
% the function

% considering w's independents, we need to set u as 0 and 0
u0 = [0;0];

% considering previous observations, we noticed a greater w0 values
% comparing to w1 values in the covariance matrix
s0 = [100 0;0 5];

% ss will affect our posterior
ss = 0.05;

% s is the standard deviation, root square of the variance
s = sqrt(ss);
```

```
hw51(ss,X,s0,u0,t,pred,x);

fprintf('Let us change the prior for 100 and 100\n');
hw51(ss,X,s0,[100;100],t,pred,x);
fprintf('1-We realized just a tiny difference in the pred. win. time\n\n');

fprintf('Let us change the covariance matriz for 200 and 200\n');
hw51(ss,X,[200 0; 0 200],u0,t,pred,x);
fprintf('2-Again, just a tiny difference\n\n');

fprintf('Now, let us change the variance for 4\n');
hw51(4,X,s0,u0,t,pred,x);
fprintf('3-We finally observed relevant changes (shape and time).\n\n');
```

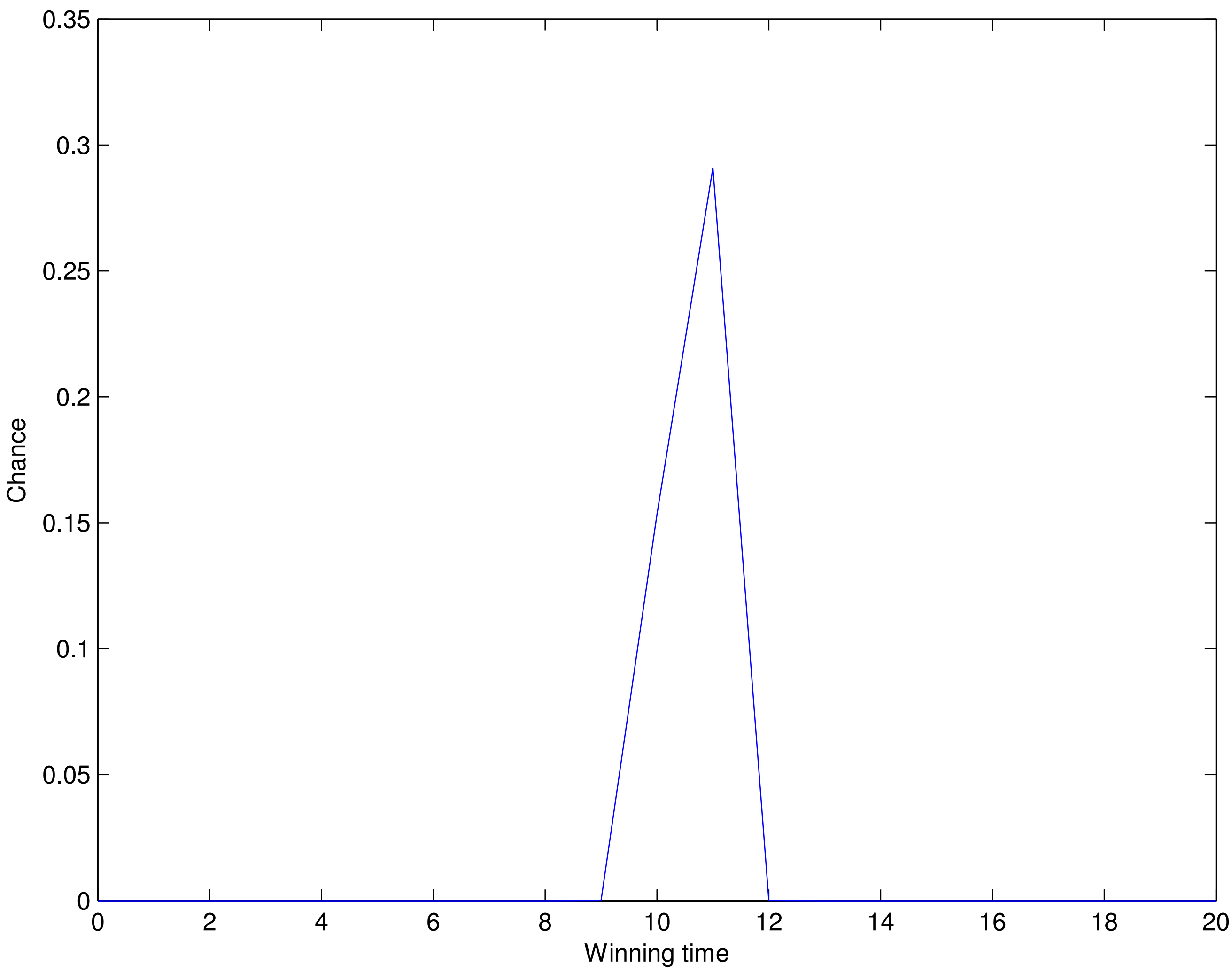
```
function hw51(ss,X,s0,u0,t,pred,x)

% referenced function from the book
sw = inv((1/ss)*X'*X + inv(s0));
uw = sw*((1/ss)*X'*t + inv(s0)*u0);

% the new x, given the year which we want to make a prediction
xnew = [1; pred];

% predicted winning time
u = xnew' * uw;

% variance
sig2 = ss + xnew' * sw * xnew;

% standard deviation
sig = sqrt(sig2);

% plotting of the gauss distribution
figure();
plot(x,normpdf(x,u,sig));


xlabel('Winning time');
ylabel('Chance');

fprintf('Predicted winning time: %g\n',u);
fprintf('New variance: %g\n\n', sig2);
```



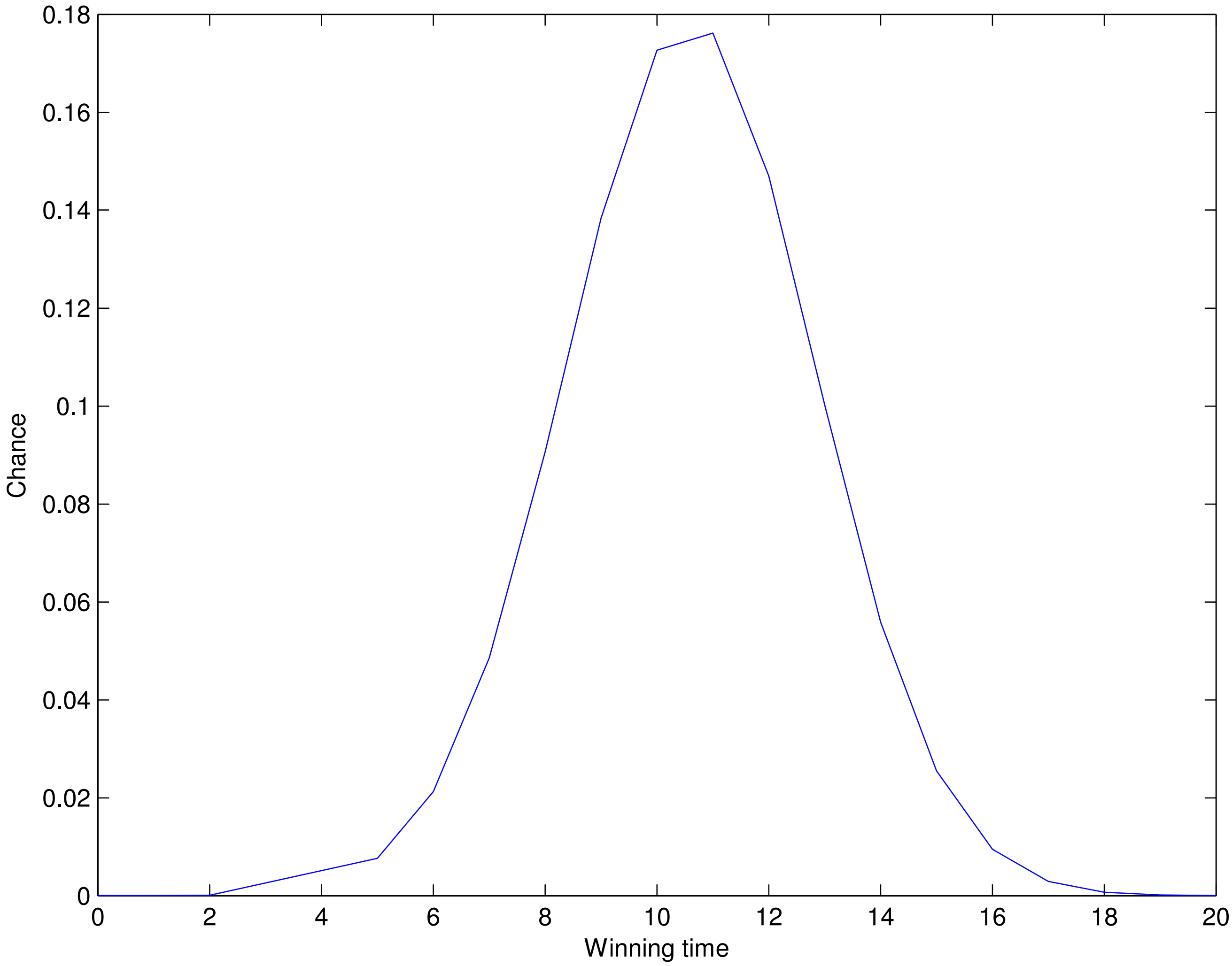
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Chance
0
2
4
6
8
10
12
14
16
18
20
Winning time



0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Chance
0
2
4
6
8
10
12
14
16
18
20
Winning time



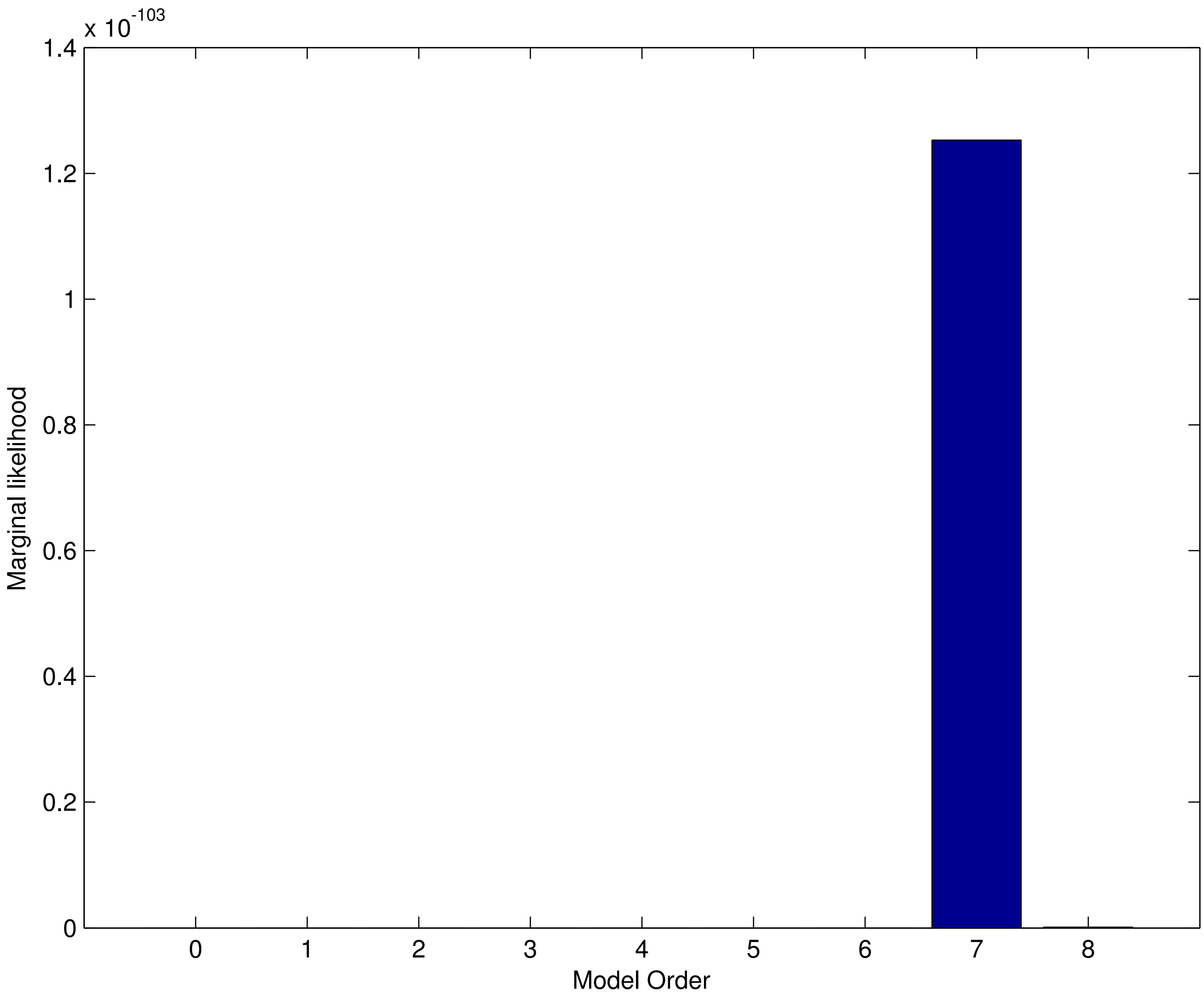
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Chance
0
2
4
6
8
10
12
14
16
18
20
Winning time



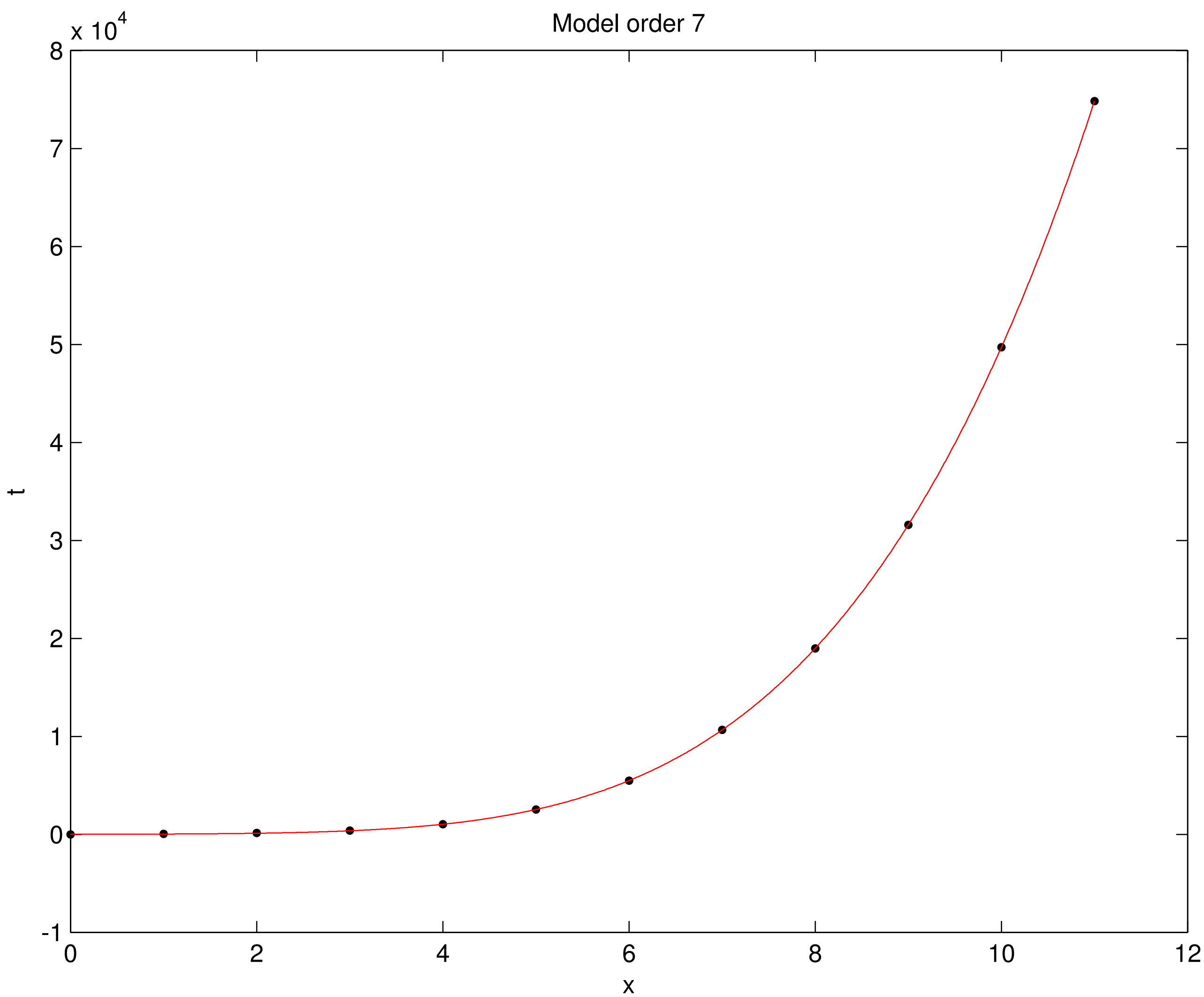
Chance
Winning time
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
2
4
6
8
10
12
14
16
18
20

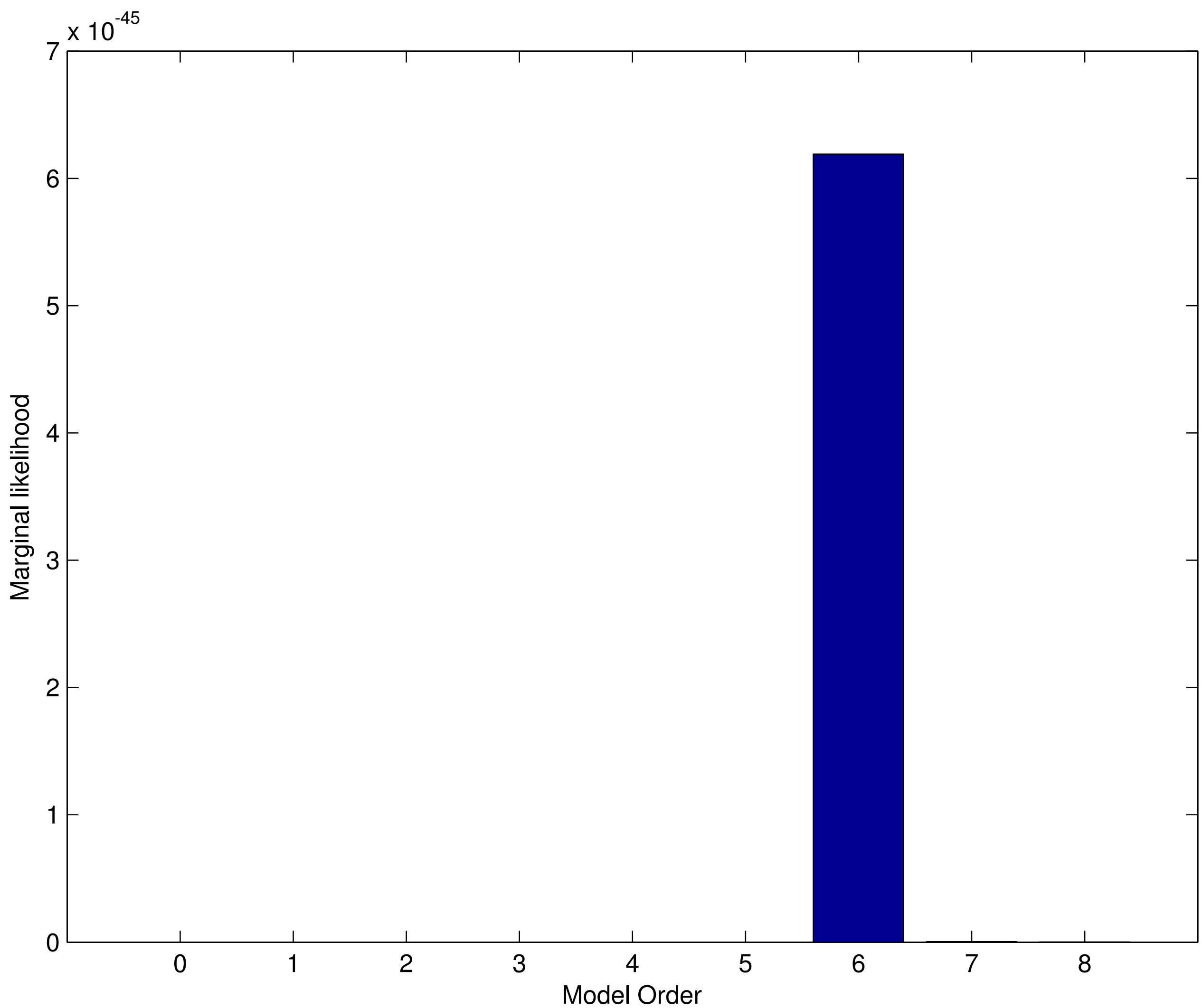
```
% Homework 5.2
% Francisco Nardi and Paulo Silva

% Cleaning the screen and variables as usual
clear all;
close all;

% Here we load the data from the file
% Ostrich
dataset = importdata('Ostrich.txt');

x = dataset(:,1);
y = dataset(:,2);
N = length(x);

noise_var = 10;

% function which describes the behavior the data
t = 5*x + 53.6*x.^2 - 25.7*x.^3 + 7*x.^4;

% adding noise to the function
t = t + randn(size(x)).*sqrt(noise_var);

% plotting the graph
plot(x,t,'k.','markersize',10);
xlabel('x');
ylabel('t');

% many models for diverse orders
orders = [0:8];
testx = [0:0.01:11]';

X = [];
testX = [];
for i = 1:length(orders)

    si0 = eye(orders(i)+1);
    mu0 = repmat(0,orders(i)+1,1);
    X = [X x.^orders(i)];
    testX = [testX testx.^orders(i)];
    siw = inv((1/noise_var)*X'*X + inv(si0));
    muw = siw*((1/noise_var)*X'*t + inv(si0)*mu0);

    figure();hold off;
    plot(x,t,'k.','markersize',10);
    xlabel('x');
    ylabel('t');
    hold on
    plot(testx,testX*muw,'r');
    ti = sprintf('Model order %g',orders(i));
    title(ti);

    % Marginal likelihood
    margcov = noise_var*eye(N) + X*si0*X';
    margmu = X*mu0;
```

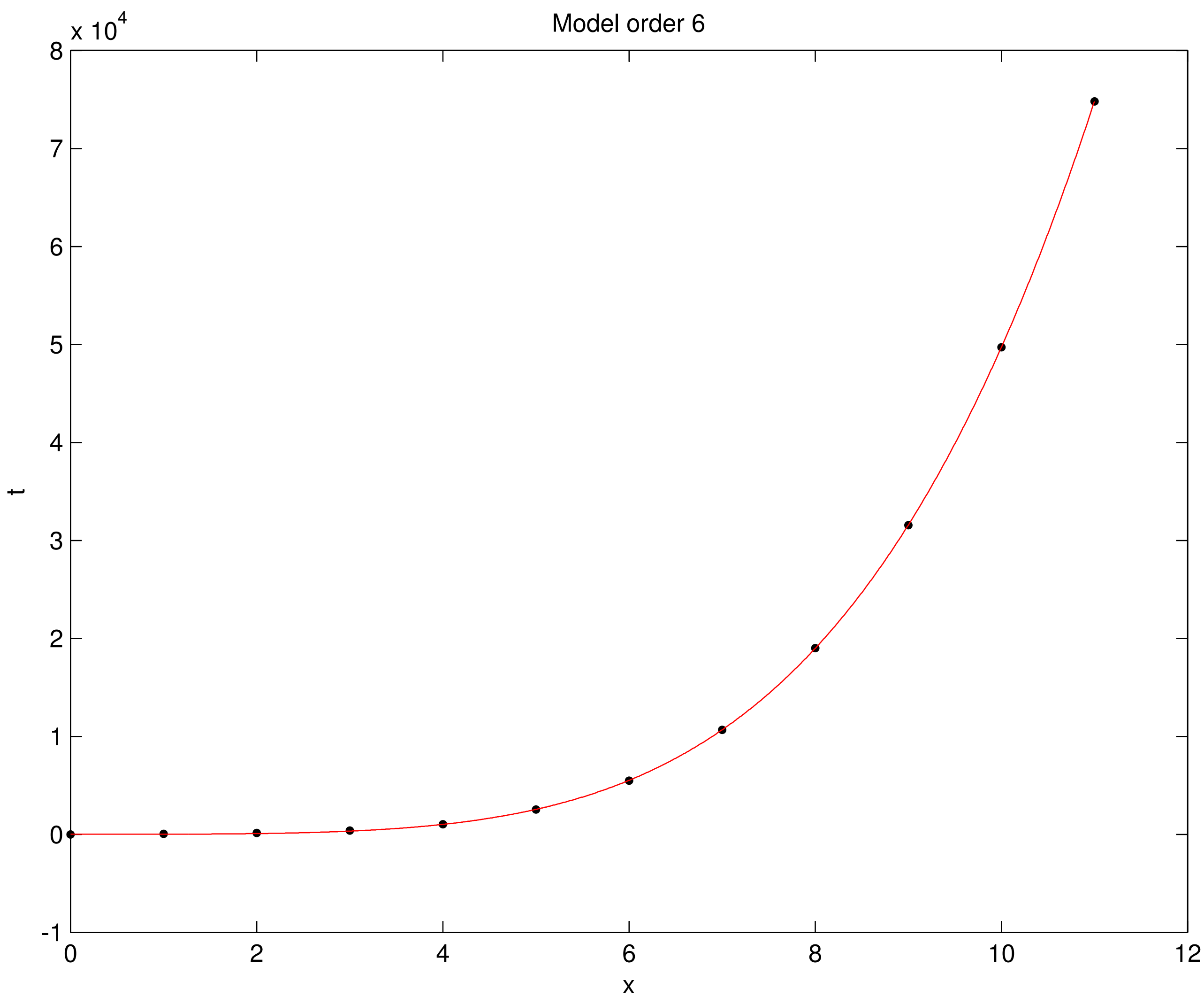
```
    D = length(margmu);
    log_marg(i) = -(D/2)*log(2*pi) - 0.5*log(det(margcov));
    log_marg(i) = log_marg(i) - 0.5*(t-margmu)'*inv(margcov)*(t-margmu);

end


% plotting them
figure(); hold off
bar(orders,exp(log_marg));
xlabel('Model Order');
ylabel('Marginal likelihood');
```



x 10^-103
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Marginal likelihood
0
1
2
3
4
5
6
7
8
Model Order


Model order 7
x 10^4
8
7
6
5
4
3
2
1
0
-1
t
0
2
4
6
8
10
12
x


x 10^-45
Marginal likelihood
Model Order
0
1
2
3
4
5
6
7
8



Model order 6
x 10^4
8
7
6
5
4
3
2
1
0
-1
t
0
2
4
6
8
10
12
x