FCI Fall 2013 Assignment 11. 50 pts.
NAME(s): **Francisco Nardi** (16163892) e **Paulo Silva** (16164638).

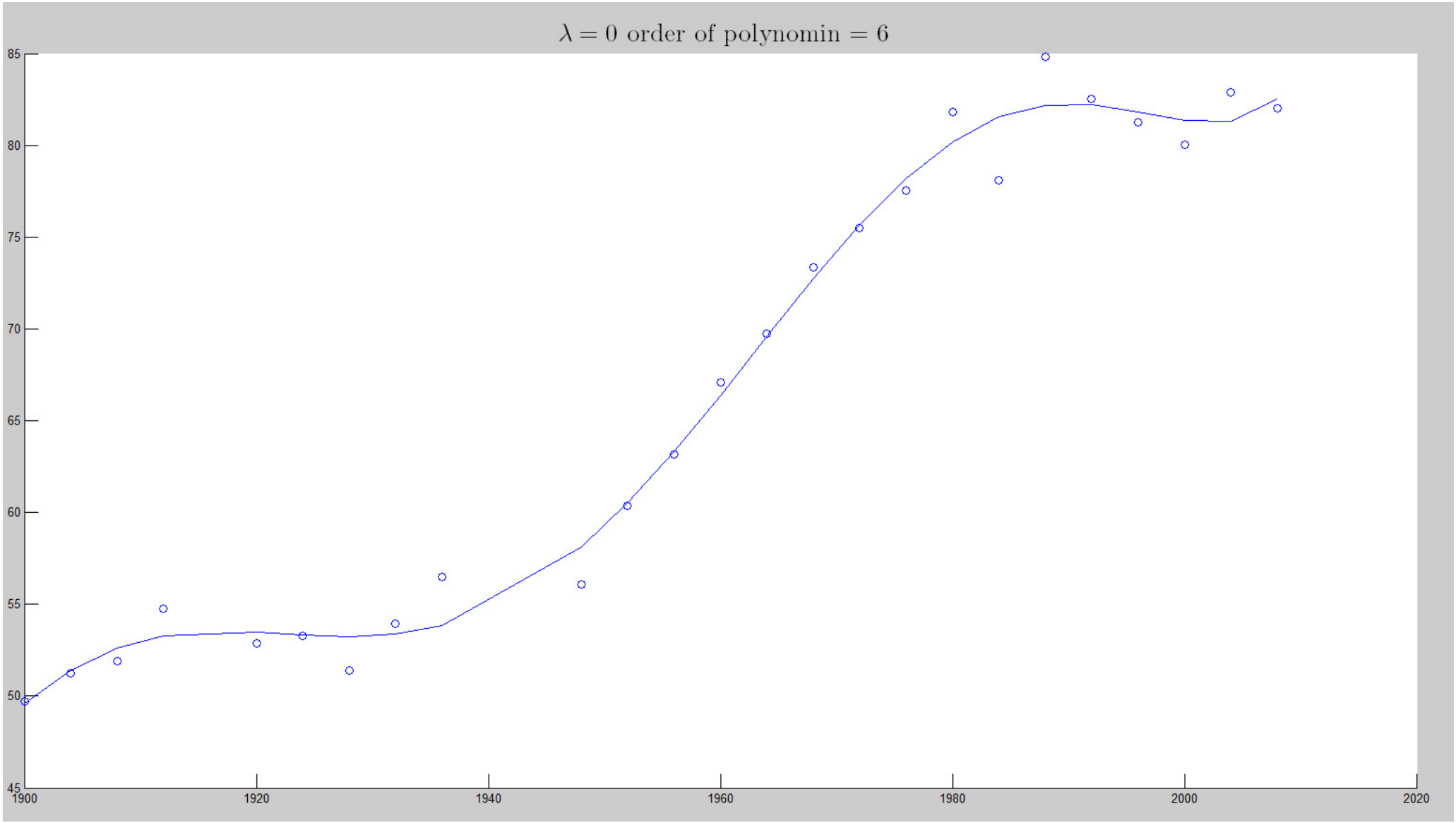
Electronic submission on Blackboard is due latest by 11 pm on Monday, Sep 16th. Submissions received after the deadline will be graded only for effort for a maximum of 70% of the total grade (Refer to class syllabus for detailed grading policy). **State any assumptions you make, justify your answers, 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 encouraged 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. (35) Use any Olympic event to create and test polynomial regression models of progressive complexity for predicting Olympic winning times or distances as a function of calendar year. (see http://www.databaseolympics.com/sport/sporteventlist.htm?sp=ATH for data)
Carry out n-way training/validation with regularization to select the best model. Finally, calculate the loss on a test set based on the best model.
Q2. (15) Choose any three Olympic events that are held for both men and women. For each event, decide if the men's record holder or women's record holder is more noteworthy by computing the Z-score.

```
% Homework 2.1
% Francisco Nardi e Paulo Silva
clear all;
close all;

dataset=importdata('datahw21.txt');
numberofdata=size(dataset,1);
figure('Name','Hammer Throw Men','NumberTitle','off')

% initialization of the variables we use for lambda values from the vector lam
lam = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
n=2;

% nmax gives the last order + 1 of the highest polynomin which is in this program 5
nmax=10;

% the vector minimeanpos is used to store  position of values for the which is the best value of
% lambda, the minimean vector stores the values of the best lambda for a given polynomin
minimeanpos=zeros(nmax-n,1);
minimean=zeros(nmax-n,1);

% vectormsev is used to store all the
vectormsev=zeros(numberofdata,1);
vectormean=zeros(11,1);

% the for with the variable z is for generate all the polynomins(degree 1 to 5)
for z = n:nmax
    % Xt is the matrix with all the variables of x in polynomin for example for
    % degree 4 : [ x1^0 x2^0 ... xn^0;...;x1^4 x2^4 ... xn^4]
    Xt=[];

    for k = 0:z-1
        % put all the values of x scaleted to correct the error of singular
        % matrices values by subtract the mean value and divide by the standard deviation
        Xt = [Xt ;((dataset(1:numberofdata)-mean(dataset(1:numberofdata)))./std(dataset(1:numberofdata))).^k];

    end
    % we create a new matrices with the vectors of x values with transpose from Xt
    X=Xt';
    %the next for will generate models for the current model of polynomin
    for l = 1:length(lam)
        lambda = lam(l);
        %the next for will use n cross validation training/validation with method LOOCV
        for i=1:numberofdata
            testingset=zeros(numberofdata,2);
            testingset=dataset(1:numberofdata,1:2);
            testingset(i,:)=[];
            Xtn=[];
            for k = 0:z-1
                Xtn = [Xtn ;((testingset(1:numberofdata-1)-mean(testingset(1:numberofdata-1)))./std(testingset(1:numberofdata-1))).^k];
```

```
            end
            Xn=Xtn';

            % we calculate the values of the parameters w in the vector w using the
            % formula : w = (X'X+N.lambda.I)^-1*X't
            w = inv((Xtn*Xn + (numberofdata)*lambda*eye(size(Xn,2))))*(Xtn*testingset↙
(1:numberofdata-1,2));
            % vectormsev stores the values of all single value of Squared Loss
            % Validation with the model with the training data and the validation
            % data as x
            vectormsev(i)=sum((dataset(i,2)-X(i,1:z)*w).^2);

        end

        % vectormean stores the mean value of the vectorsev giving the output of
        % LOOCV method
        vectormean(l)=mean(vectormsev);

    end

    % the vectors minimean and minimeanpos gives the minimum value of LOOCV of
    % given order of polynomin with the currespond value and position of lambda↙
respectivetely
    [minimean(z-1),minimeanpos(z-1)]=min(vectormean);

    %we set the variable lambda with the minimum value of lambda from the given↙
polynomin
    lambda = lam(minimeanpos(z-1));

    %lambdaminglobal stores the position of lambda with the minimum LOOCV for every↙
order of polynomin
    lambdaminglobal(z-1)=minimeanpos(z-1);
    w = inv((Xt*X + numberofdata*lambda*eye(size(X,2))))*(Xt*dataset(1:numberofdata,↙
2));

    %calculate w with the best model for the current order of polynomin
    figure(z-1);

    %create a new figure which has all the data and the polynomin model with best↙
lambda
    hold off
    scatter(dataset(1:numberofdata,1),dataset(1:numberofdata,2));
    hold on
    plot(dataset(1:numberofdata,1),X*w);
    ti = sprintf('$\\lambda = %g$ order of polynomin = %d',lambda,z-1);
    title(ti,'interpreter','latex','fontsize',20)

end

%minimeanglobalval has the minimum LOOCV of the best model gave by the
%minimeanglobalpos which gives the order of polynomin
[minimeanglobalval,minimeanglobalpos]=min(minimean);
Xt=[];
for k = 0:minimeanglobalpos-1
```

```
    %regenerate Xt matrix with the best model
    Xt = [Xt ;((dataset(1:numberofdata)-dataset(1,1))./(4)).^k];

end
X=Xt';

%recalculate the lambda with the best lambda from the best model
lambda=lam(lambdaminglobal(minimeanglobalpos));

%recalculate w for the best model
w = inv((Xt*X + numberofdata*lambda*eye(size(X,2))))*(Xt*dataset(1:numberofdata,2));
averageloss=sum((dataset(1:numberofdata,2)-X*w).^2)/numberofdata;

%averageloss correspond the MSE for all the test set(all data)
hold off
ti = sprintf('Best Model!polynomin order= %d , average loss : %d',minimeanglobalpos,↙
averageloss);

%retitle the best model
figH= figure(minimeanglobalpos);
set(figH,'Name',ti,'NumberTitle','off')
hold on
```


$\lambda = 0$ order of polynomin $= 6$
85
80
75
70
65
60
55
50
45
1900
1920
1940
1960
1980
2000
2020

```
%Homework 2.2
%Francico Nardi and Paulo

%First, we need to clear the screen and erase the variables
%perhaps already stored in order to start running our program
clear all;
close all;

%Here we load the data
filename1 = '4x100mRelayMen.txt';
data1 = importdata (filename1);

filename2 = '4x100mRelayWomen.txt';
data2 = importdata (filename2);

filename3 = '100mMen.txt';
data3 = importdata (filename3);

filename4 = '100mWomen.txt';
data4 = importdata (filename4);

filename5 = '400mHurdlesMen.txt';
data5 = importdata (filename5);

filename6 = '400mHurdlesWomen.txt';
data6 = importdata (filename6);

%We put each best time in a different vector, since there are
%different years that something was disputed
a = data1 (: , 2);
b = data2 (: , 2);
c = data3 (: , 2);
d = data4 (: , 2);
e = data5 (: , 2);
f = data6 (: , 2);

%We take the zscore of each modality per sex
za = zscore(a);
zb = zscore(b);
zc = zscore(c);
zd = zscore(d);
ze = zscore(e);
zf = zscore(f);

%We take the minimum values of each modality
ma = min(za);
mb = min(zb);
mc = min(zc);
md = min(zd);
me = min(ze);
mf = min(zf);

%Counters used in order to know which gender has more noteworthy records
countmale = 0;
countfemale = 0;
```

```
%
if (ma < mb)
    countmale = countmale +1;
    disp('On modality 4x100m Relay -> Men`s record was better');
elseif (ma > mb)
    countfemale = countfemale + 1;
    disp('On modality 4x100m Relay -> Women`s record was better');
else
    disp('On modality 4x100m Relay -> Record had the same value');
end

if (mc < md)
    countmale = countmale +1;
    disp('On modality 100m -> Men`s record was better');
elseif (mc > md)
    countfemale = countfemale + 1;
    disp('On modality 100m -> Women`s record was better');
else
    disp('On modality 100m -> Record had the same value');
end

if (me < mf)
    countmale = countmale +1;
    disp('On modality 400m Hurdles -> Men`s record was better');
elseif (me > mf)
    countfemale = countfemale + 1;
    disp('On modality 400m Hurdles -> Women`s record was better');
else
    disp('On modality 400m Hurdles -> Record had the same value');
end

fprintf ('\ntherefore,');
if (countmale > countfemale)
    fprintf('\n Men`s records were more noteworthy than women`s');
elseif (countmale < countfemale)
    fprintf('\n Women`s records were more noteworthy than men`s');
else
    fprintf('\n Men`s and women`s quantity of noteworthy records were the same');
end
```