
Example Script by Jason Yalim

Table of Contents

Example 1	1
Example 2	5

Publishing Matlab code:

- [my_advice](#)
- [mathworks_code](#)
- [specifying_output_preferences_for_publishing](#)

Publish doc:

- [mathworks_publish](#)

Publish Markdown doc:

- [mathworks_markup](#)

Example 1

External code called by script NOTE THAT THERE MUST BE LINE BREAKS BEFORE AND AFTER include TAG LINE AND NO SPACES WITHIN TAG.

```
%% Jason Yalim
%
=====
% my_ls_fit.m
% least squares fitting algo
%
-----
% The Vandermonde matrix,  $X$ , is defined as,
%
%  $X_{jk} = x_j^k$ ,
%
% thus the input data and output coefficients have the following
% relationship,
%
%  $y_j = \sum_{k=0}^N c_k x_j^k$ ,
%
% which expressed with matrices and performing a least squares,
%
%  $Xc = y \implies X'Xc = X'y$ ,
%  $QRc = y \implies R'Q'QRc = R'Q'y$ ,
%  $c = \text{inv}(R)\text{inv}(R')R'Q'y$ ,
```

```
% $$                                c = inv(R)Q'y.                $$
%
-----

function [c] = my_ls_fit(x,y,order)
% generating Vandermonde matrix
X = ones(length(x),order+1);
for k = 0:order
    X(:,k+1) = x.^k;
end
% performing tall factorization of $X$, such that $R$ is square.
[Q,R] = qr(X,0);
% Ensure that $c$ is a column vector
[m,n] = size(y);
if m > n
    c = inv(R)*Q'*y;
else
    c = inv(R)*Q'*y';
end
end
% [x] Party button

%% Jason Yalim
%
=====
% my_ls_eval.m
% least squares eval algo
%
-----
% Assumes coefficients indices are increasing,
%
%     $$ y_j = y(x_j;c) = \sum_{k=0}^{N-1} c_k x_j^k, $$
%
% where $N$ is the length of the coefficient vector, $c$.
%
-----

function [y] = my_ls_eval(c,x)
X = zeros(length(x),length(c));
for k = 0:length(c)-1
    X(:,k+1) = x.^k;
end
% $c$ must be a column vector
[m,n] = size(c);
if m > n
    y = X*c;
else
    y = X*c';
end
end
% [x] Party Button
```

```

%% figprops.m
%
=====
% A script to adjust the look of plots, in an attempt to improve
% formatting.
%
% Finish plotting, then call figprops.m (this script). E.g.,
%
%     x = linspace(-1,1,101)'; y = x.^2; figure(1), clf
%     plot(x,y), title('example'), xlabel('x'), ylabel('x^2'),
%     figprops
%
% Modified by Jason Yalim, from APM 505, Professor Rosie Renaut, F13.
%
-----
% text properties:
%   https://www.mathworks.com/help/matlab/ref/text-properties.html
% axes properties:
%   https://www.mathworks.com/help/matlab/ref/axes-properties.html
%
-----
%

opts = {
    { 'line', {
        'MarkerSize', 9,
        'LineWidth', 1,
    },
    },
    { 'text', {
        'FontWeight', 'normal',
        'FontAngle', 'normal',
        'Color', 'k',
    },
    },
    { 'axes', {
        'FontSize', 16,
        'FontWeight', 'normal',
        'FontAngle', 'normal',
        'Color', 'w',
    },
    },
    { 'title', {
        'FontSize', 18,
        'FontWeight', 'bold',
        'FontAngle', 'normal',
        'Color', 'k',
    },
    },
};

gca_FontWeight = 'normal'; % bold
gca_FrameWidth = 1.2;

```

```

if exist('a'), aold=a; end
if exist('h'), ho=h; end

%axis tight
axis(axis);

h=gcf; %handles of the figure
a=gca; %axes handle

% This makes the text on the axis bold and the x or y label bold and
the title
set(gca,'FontWeight',gca_FontWeight);
set( a, 'LineWidth',gca_FrameWidth); % This makes the width of the
axis box wider

% note that it seems to matter that we do titles etc after setting to
bold
for k = 1:length(opts)
    names = {}; values = {}; count = 1;
    for j = 1:2:length(opts{k}{2})
        names{count} = opts{k}{2}{j};
        values{count} = opts{k}{2}{j+1};
        count = count + 1;
    end
    set(findobj('Type',opts{k}{1}),names,values);
end

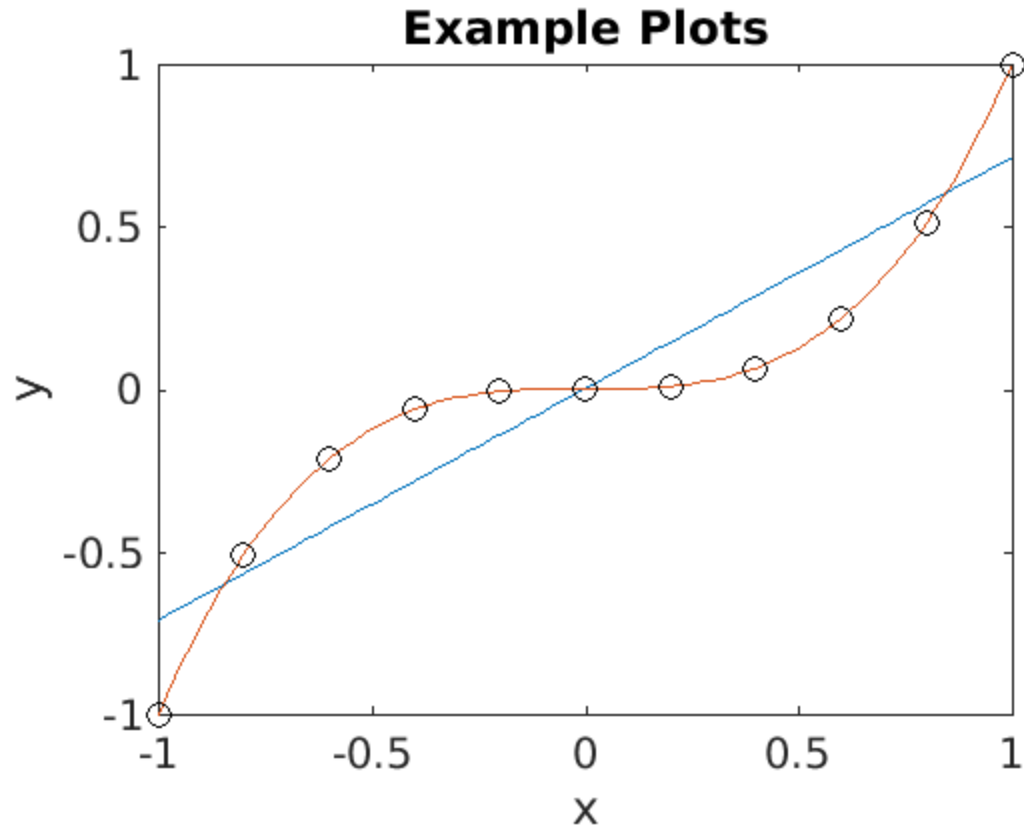
if exist('aold'), a=aold; end
if exist('ho' ), h=ho; end
hold off

% Section init
clear all, close all

% Problem init
x = linspace(-1,1,11)'; % transpose to column vectors
q = linspace(-1,1,101)';
y = x.^3;
c1= my_ls_fit(x,y,1);
c3= my_ls_fit(x,y,3);
y1= my_ls_eval(c1,q);
y3= my_ls_eval(c3,q);

figure(1), clf
plot(q,y1,'-'), hold on
plot(q,y3,'-')
plot(x,y,'ko')
title('Example Plots')
xlabel('x')
ylabel('y')
figprops

```



Example 2

```
% Section init
clear all, close all

whos          % Show workspace
a = 5;
whos          % Show workspace

ls            % Display files/directories from working directory

a = { ones(5), 2, 'three' } % Cell Array
a(1)          % Object is shown
a{1}          % Value is shown
% Note where the output from the code is placed in the publish
```

Name	Size	Bytes	Class	Attributes
a	1x1	8	double	

```
src

a =

    1x3 cell array
```

```
[5×5 double]    [2]    'three'
```

```
ans =
```

```
cell
```

```
[5×5 double]
```

```
ans =
```

```
1      1      1      1      1
1      1      1      1      1
1      1      1      1      1
1      1      1      1      1
1      1      1      1      1
```

Published with MATLAB® R2016b