HW 5

1

```
% Normally housekeeping commands would be here.
% Omit them for MATLAB Grader.

%vector with grades from Hogwarts
OWL_grades = ['D' 'P' 'T' 'P' 'D' 'E' 'P'];

OWL_scheme = ['O' 'E' 'A' 'P' 'D' 'T'];
HHS_scheme = ['A' 'B' 'C' 'D' 'F' 'F'];
Point_scheme = [4 3 2 1 0 0];

HSS_grades = arrayfun(@(a) HHS_scheme(find(OWL_scheme == a)),
OWL_grades)

% calculate the GPA.  Store result in GPA.

Points_grades = arrayfun(@(a) Point_scheme(find(OWL_scheme == a)),
OWL_grades);
GPA = mean(Points_grades)
```

2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Trevor Nichols              %
% ENGR 130 Module HW 5        %
% Section D                   %
% 2024-09-22T19:04:05-04:00   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Question 2 %%%
clear; clc; close all;

% Input Data

horiz_dist = [0 1.9 3.3 4.2 6.3 7 7.7 9.8 10.9 12.5 14 14.9 16.7 17.5];
vert_dist = [0 5.2 6.6 10 13.3 14.8 15.6 15 12.9 9.4 6.4 5.3 4.2 0];

% Calculate the polynomial fit and evaluate it at 20ppu

fit = polyfit(horiz_dist, vert_dist, 2);
fit_x = linspace(0, 17.5, 20);
```

```
fit_y = polyval(fit, fit_x);

% Begin the figure

figure
hold on

% Plot the original data
scatter(horiz_dist, vert_dist);
% Plot the polyfit
plot(fit_x, fit_y);

% Label graph
legend("Exp. Data", "Quadratic fit");
title("Vertical vs. Horizontal distance for tossing of a cat toy");
xlabel("Horizontal Distance (in)");
ylabel("Vertical Distance (in)");

hold off
```

3

```
%%% Question 3 %%%

% Initialize the graph
figure
hold on

% Input data
data = [
  struct( ...
      "name", ...
      "Paws & Claws Cuisine", ...
      "price", ...
      [ 10.05 10.00, 9.95, 9.90 ], ...
      "order", ...
      1 ...
  )
  struct( ...
      "name", ...
      "Purrfect Pate", ...
      "price", ...
      [ 8.99 9.05, 9.50, 10.05 ], ...
      "order", ...
      2 ...
  )
  struct( ...
      "name", ...
      "Whisker Licks", ...
      "price", ...
      [ 9.40 9.50 9.55 10.00 ], ...
      "order", ...
      3 ...
  )
];

x_pos = 1:1:4;

% Polyfit domain
x_pos_interp = linspace(1, 6, 10);

% Initialize legend
leg = [];

% Plot original data and append labels to legend
arrayfun(@(a) scatter(x_pos, a.price), data)
leg = cat(2, leg, arrayfun(@(a) sprintf("%s historical cost", a.name),
data));

% Calculate respective polyfits for input data
fits = arrayfun(@(a) struct("poly", polyfit(x_pos, a.price, a.order)),
data);

% Plot input data over the polyfit domain and append labels to legend
fit_graphs = arrayfun(@(a) plot(x_pos_interp, polyval(a.poly,
x_pos_interp)), fits);
leg = cat(2, leg, arrayfun(@(a) sprintf("Order %i polynomial fit for
%s", a.order, a.name), data));

% Initialize domain for extrapolation; and extrapolate
pred_x = [ 5 6 ];
pred_y = arrayfun(@(a) struct("y", polyval(a.poly, pred_x)), fits);

% Plot extrapolations and append to the legend
predictions = arrayfun(@(a) scatter(pred_x, a.y, Marker="x"), pred_y);
leg = cat(2, leg, arrayfun(@(a) sprintf("Projected cost for %s",
a.name), data));

% Display the legend and label the graph
legend(leg, Location="northwest");

title("Price vs. Year for 3 cat food brands");
xlabel("Year (year)");
ylabel("Price ($)");

hold off
```

```
% Print results based on already calculated projections
results = arrayfun(@(a) fprintf("%s projected cost: $%.2f\n",
data(a).name, pred_y(a).y(1)), 1:length(data));

% Find the minimum and print
[m, minI] = min(arrayfun(@(a) a.y(1), pred_y));

fprintf("\nThe lowest projected cost is $%.2f\n", pred_y(minI).y(1));
```

4

✓ **Answer**

Innovation really is a melting pot of the ideas that surround you, particularly in environments with people of different perspectives. Also connecting is more innovative than protecting your own ideas.

Participate and talk out ideas in my team, whether it be a project team or just a my group of friends.