SectionD_tln32_HW8

HW 8

1

Copy

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Trevor Nichols             %
% ENGR 130 Module HW 8       %
% Section D                  %
% 2024-10-15T00:50:36-04:00  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%bring data in from file
load("HW8_vectors.mat", "pricesPer", "qtyBought")

% set up variables to appropriate initial values
itemPrice = pricesPer .* qtyBought;
totalNoSales =  sum(itemPrice);   % will eventually be the total paid
without the sale
total = sum(itemPrice .* (-0.1 * (itemPrice > 12) + 1));  % will
eventually be the total paid with the sale discount

% Apply tax
totalNoSales = totalNoSales * 1.08;
total = total * 1.08;

% print the total cost for both prices (with/without sales)
fprintf('The total cost is $%.2f with sales applied\n',  total   );
fprintf('The total cost is $%.2f without sales applied\n', totalNoSales
);

% print the total amount of money saved by the sale
fprintf('She saved $%.2f with the sales!\n', totalNoSales - total);
```

2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Trevor Nichols             %
% ENGR 130 Module HW 8       %
% Section D                  %
% 2024-10-15T00:50:36-04:00  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%% Question 2 %%

clear; clc; close all;

% Load the input data
load("HW8_Cuyahoga_Contaminants.mat", "contaminants")

range = 1:365;

% Find the highest contamination value
[maxv, maxi] = max(contaminants);

% Begin the first figure
figure(1)

% Start the first graph
hold on

% Plot the original data
plot(range, contaminants);
% Plot the max
plot(maxi, maxv, "Marker", "x");
% Add the reference line
yline(50);

% Label the graph
xlabel('Day of year');
ylabel('Contaminant density (NTU)');
title('Contaminant density over time');
legend('Contaminant density over time', 'Maximum contamination')

hold off

%%% Seasonal trends %%%

% Begin the second figure
figure(2)

% Define our subplots as well as season information
subpSize = 2;
seasons = [{1:92} {93:186} {187:274} {275:365}];
seasonNames = ["Winter", "Spring", "Summer", "Fall"];
noop = @(varargin) disp("");

% Define that we want cubic fit for the first season
extraAnalysis = [{@addCubic},{noop},{noop},{noop}];

%%%%%
%
```

```
% Use the cellfun function to apply a function to each cell of the list
% https://www.mathworks.com/help/matlab/ref/cellfun.html
%
%%%%%

% Plot the four seasons in subplots
plots = cellfun( ...
    @(r, i, n, fun) subp( ...
        subpSize, ...                % Size of the subplot
        subpSize, ...
        i, ...                       % Index of the subplot
        r, ...                       % Domain of data to plot
        contaminants(r), ...         % Points to plot
        n, ...                       % Name of the subplot
        "Day", ...                   % xlabel of the subplot
        "Turbidity (NTU)", ...       % ylabel of the subplot
        fun, ...                     % Extra things to plot on the
subplot
        500 ...                      % Density argument for the extra
plotting
    ), ...
    seasons, ...             % Pass in our season definition
    num2cell(1:4), ...       % Pass in the ID of the season
    seasonNames, ...         % Pass in the names of the seasons
    extraAnalysis, ...       % Pass in the extra functions for the extra
analysis
    "UniformOutput", false ...
);

% Calculate CV for each season
CV = cellfun( ...
    @(r) 100 * std(contaminants(r)) / mean(contaminants(r)), ...
    seasons ...
);

% Find max CV season
[maxS, maxSi] = max(CV);

% Find days over 50 NTU per season
overDays = cellfun( ...
    @(r) sum(contaminants(r) > 50), ...
    seasons ...
);

% Format report as specified
fprintf("* CEIA Report - 2023 Cuyahoga River Turbidity *\n");
fprintf("In 2023, there were %i days (%.2f%%) where contaminants
exceeded 50 NTUs:\n", sum(overDays), 100 * sum(overDays)/365);
fprintf("* Winter: %i days\n", overDays(1));
fprintf("* Spring: %i days\n", overDays(2));
```

```
fprintf("* Summer: %i days\n", overDays(3));
fprintf("* Fall: %i days\n", overDays(4));
fprintf("Season with highest CV: %s (CV = %.2f%%)\n",
seasonNames(maxSi), CV(maxSi));

% Define a custom subp function for each season subplot
% m, n, p: Same arguments as @subplot
% x, y: The graph arguments
% extra: A function that takes the x and y arguments as well as the
density argument and plots extra information
% density: The number of points to use for the cubic fit
% xlab, ylab: The x and y labels
%
% Returns:
% o: The handle of the created subplot
function o = subp(m, n, p, x, y, name, xlab, ylab, extra, density)
    subplot(m, n, p);
    hold on
    o = plot(x, y);
    extra(x, y, density);
    yline(50);
    title(name);
    xlabel(xlab);
    ylabel(ylab);
    hold off
end

% Adds a cubic polyfit to specified data
% To be used in the extra argument in @subp
% x, y: The data points to fit
% den: The number of points to use for the cubic fit
%
% Returns:
% o: The handle of the created plot of the cubic fit
function o = addCubic(x, y, den)
    fit = polyfit(x, y, 3);
    mi = min(x);
    ma = max(x);
    ran = linspace(mi, ma, den);
    o = plot(ran, polyval(fit, ran));
    legend("Real Data", "Cubic Fit")
end
```