```
% MatLab_Project_Semester_Banks.m    This program was created to aid in the statistical
analysis of
%numerical data using basic statistical analysis.

%   MatLab_Project_Semester_Banks.m overwrites these variables:
%%%%%%%%%%%%%%%%%%%%%%%%
%      list of global variables that the program creates or changes
%%%%%%%%%%%%%%%%%%%%%%%%
%   MatLab_Project_Semester_Banks.m prompts the user to input values for
%      list the variables that correspond to input prompts, if any


% By submitting this assignment, I agree to the following:
%  "Aggies do not lie, cheat, or steal, or tolerate those who do"
%  "I have not given or received any unauthorized aid on this assignment"
%
% Name: Aaron Banks
% Section: 536
% Team:        Team 11
% Assignment: Matlab Semester Porject 3
% Date:        04/23/2017
% Initial settings for username, data, and filename as blank arrays to
    % alllow data to be inserted into them in every ineration of the
    % application without being added to previously stored values
 %%% COMMENT %%% = scrape code/comment
 run = true;
username = [];
data = [];
filename = [];
% X and Y equallying zero are just place holders for linear
    % regression/statistics
X=0;
Z=0;
Probability=0;
while run == true
    %While loop is set to run as long as the statment is true, if false the
        % the loop will terminate itself, saving all priorly recorded values
    Application = 0;
    while Application == 0
```

```
    Application = 0;
    % Application is a hybrid GUI-based/Text-based application. Useers
      % will have menus to click in external windows, however user
      % promt's will have to be inserted into the command window
    Application = menu('Main Menu','Enter Username','Load Data File','Clear Data From
Memory','Set Output Destination','Plot Histogram','Plot Probability Plots','Plot Probability',' Find
probability using x or z values','Find x and z','Data Collection','Factory Reset','Exit');
  end
  % switch output between 1st input and 3rd input based on value of second
  switch Application
    case 1
      %Enter User Name
      % The purpose of this case is to prompt the user to enter a
        % customizable self generated username.
      %determines if username array is empty true = 1, false = 0
      if isempty(username)== 0
        disp('You Already Have a Username, clear data to input a new username. \n');
      else
        %Prompts user to enter customized username to personalize
          %relation with applicaion
        username = input('Please input a username ','s');
      end
      fprintf('Howdy! I am Cotana your organizational assistant, nice to meet you %s. Today you
will be stress testing my GUI, I hope all is well \n',username)
    case 2
      % Load Data File
      % The purpose ofthis case is to prompt the user to enter the
        % name of a file that needs to be uploaded, without the
        % file type extension.

      %determines if data array is empty true = 1, false = 0
      if isempty(data) == 0
        % Used as a preventative measure against inccorectly
          % inserting multiple filenames.
        disp('You already have the data file, clear data to input new data. \n');
      else
        % User must enter a file name without data type extension
          % for simplicity of use
        data = input('Input a data file. ','s');
```

```
            fprintf('%s your data has been successfully loaded. \n','username');
            LoadDataFile = 0;
            % while there is a data file loaded in array the user can
               % specify what type of file is being uploaded
            while LoadDataFile == 0
               % .txt = text file, %.dat = % .xlsx = excel file
               % User chooses from a list of preselected/accepted file types to upload their data as.
               % Increased variation in accepted files coming at a
                  % not yet determined later date
               LoadDataFile = menu('What type of file is being Uploaded? ','.txt file','.dat file','.xlsx
file');
            end

            switch LoadDataFile
               % Creating cases for varying/acceptable data types.
               % Prevents user from uploading an unsupported file type
               case 1
                  % strcat = horizontally concatenates character arrays to
                  % combine filename with file type extension
                  data = strcat(data,'.txt');
               case 2
                  data = strcat(data,'.dat');
               case 3
                  data = strcat(data,'.xlsx');
                  data = xlsread(data);
            end

            data = importdata(data);

         end
      case 3
         % The purpose of this case is to delete uploaded data files and usernames from the
application workspace.
         username = [];
         data = [];

      case 4
         % The purpose of this case is to output uploaded file
         % statistical data into another external file.
```

```
        % Set Output Destination
        if isempty(filename) == 0
           disp('A Data file has already been inputed, clear data to input new data.');
        elseif isempty(data) == 1
           disp('Input a data file. \n');

        else
           % Purpose of this else condition is to calculate statisical
              % values for outputed file.
           filename = input('Input a new file. \n','s');
           Mean_newdoc = mean(data);
           Median_newdoc = median(data);
           Mode_newdoc = mode(data);
           Variance_newdoc = var(data);
           StandardDeviation_newdoc = std(data);
           Minimum_newdoc = min(data);
           Maximum_newdoc = max(data);
           filename = fopen(filename,'w');
%%% fprintf(filename,input('Enter OutPut File Title \n','s'));%%%
           OutPutTitleMenu = menu('OutPut Title','.txt','.dat','.xlsx');
           OutPutTitle = input('Input title name: \n','s');
           switch OutPutTitleMenu
              case 1
                 % strcat = horizontally concatenates character arrays to
                 % combine filename with file type extension
                 newdata = strcat(filename,'.txt');
              case 2
                 newdata = strcat(filename,'.dat');
              case 3
                 newdata = strcat(filename,'.xlsx');
                 newdata = xlsread(filename);
           end
           fprintf(filename,OutPutTitle);

           fprintf(filename,'Mean = %.2f  \n',Mean_newdoc);
           fprintf(filename,'Median = %.2f  \n',Median_newdoc);
           fprintf(filename,'Mode = %.2f  \n',Mode_newdoc);
           fprintf(filename,'Var = %.2f  \n',Variance_newdoc);
           fprintf(filename,'Stdev = %.2f  \n',StandardDeviation_newdoc);
```

```
    fprintf(filename,'Min = %.2f  \n',Minimum_newdoc);
    fprintf(filename,'Max = %.2f  \n',Maximum_newdoc);
    fclose(filename);
  end

case 5
  % Plot Histogram
  % The purpose of this case is to plot the column data of a
    % matrix in a selectable number of bins
  if isempty(data) == 1
    disp('You need to input a data file \n','s');
  else
    % binnum = bin number
    binnum = input('How many bins/sectors do you want in your histogram? \n')
    hist(data,binnum);
    xlabel(input('Enter x-axis label. \n','s'));
    ylabel(input('Enter y-axis label. \n','s'));
    title(input('Enter Histogram Title. \n','s'));
  end
case 6
  % Plot Probability Plots (histfit)
  % The purpose of this case is to normally distribute data, for
    % statistical use pertaining to confidence intervals
  if isempty(data) == 1
    disp('You need to input a data file. \n');
  elseif isempty(data) == 0
    % dependent of whether the data has 1 or 2 colomns of data
    [m,n] = size(data)
    while n == 1;
      % using normcdf to calculate the confidence interval
       normdata = normpdf(data);
  histfit(normdata(n));
  histfit(normdata)
    end
    while n == 2;
      normdata = normpdf(data)
  histfit(normdata(n))
  histfit(normdata)
    end
```

```
        end
 % possible alternative way to plot histogram
%         [m,n] = size(data);
%
%         if n == 1
%            hold on
%            histfit(data(1));
%            histfit(data);
%         elseif n == 2
%            histfit(data(2));
%
%         elseif n == 3
%            histfit(data(3));
%
%
%         end

      case 7
         % Plot Probability (probplot)
         % The purpose of this case is to make comparisons between the
         % normally distributed data from the previous case to imputed z
            % values
         if isempty(data) == 1
            disp('You need to input a data file.');
         else
            figure
            normplot(data);
         end
      case 8
         % find (regression) probability using x or z values
         % the purpose of this case is to create a linear regression
            % slope line of the data points before being normally
            % distrubuted
         if isempty(data) == 1
            disp('You need to input a data file.\n');
         else
            Regression = 0;
            while Regression == 0
               Regression = menu('Choose one','X','Z.\n');
```

```
        end
        switch Regression
            case 1
                x = input('input in X value: \n');
                mu = mean(data);
                sigma = std(data);
                Probability = normcdf(x,mu,sigma);
            case 2
                z = input('input a Z value: \n');
                Probability = normcdf(x,0,1);
            otherwise
                disp('You done messed up A-ARON!!! \n');
        end
    end
case 9
    % find x and z
    % The purpose of this case is to calculate confidence intervals
    % based off of inserted Z, mu, standard deviation, mean, etc.
    % values
    if isempty(data) == 1
        % prevents user from entering data in a full array
        fprintf('You need to input a data file. \n');
    else
        % defining variables for confidence interval equation
        mu = mean(data);
        sigma = std(data);
        % User inputs desired probability to output the according
            % z-value
        p = input('Enter probability: \n');
        Z = norminv(p,0,1);
        % x normally distributes data, to accurrately determine
            % condifence interval
        X = norminv(p,mu,sigma);
    end

case 10
    % Data Collection
    % The purpose of this case is to simply comllect the components
```

```
        % of data used thoughout this application, and within the
        % statistical analyses
    % Purpose of this case is to simply compile a list of
        % statisical data values, for the user to see
    % Case serves a a glorified calculator for specific functions
        % essentially
    Count = sum(data(:));
    StandardDeviation_newdoc = std(data);
    % Checks that wheather the matrix is empty or not
    if isempty(data) == 0
        % basic statistical analysis
        mean = mean(data);
        median = median(data);
        mode = mode(data);
        variation = var(data);
        StandardDeviation_newdoc = std(data);
        MinimumValue = min(data);
        MaximumValue = max(data);
        Count = sum(data(:));
    else
        fprintf('Load Data Prior to Data Collection. \n')
    end
    if Count >= 30
        % changes the type of deviation from standard to population
            % standard if data size has more than 30 subject
        %relative deviation (population standard deviation)
        disp(std(data,1));
    else
        %standard deviation
        disp(StandardDeviation_newdoc);
        pause;
    end
case 11
    % Delete Every variable
    clear all;
    clc;
case 12
    % exits the script
    fprintf('Thanks %s and Gig Em for testing this application. \n','username')
```

```
        run = false;

    end
end
```