ENG06 Project #1

Collaboration Policy:

You may talk at a high level about the project with up to 2 other students. They are your collaborators. By high level discussions, we mean that you can say things like `We need to use a For Loop to accomplish this task'. Once you start `speaking in Matlab code', you've gone too far. Collaborators may help each other debug their code, but the only hands that should ever touch the keyboard for your project are your own. Transferring even small portions of code to each other is certainly not allowed. You must submit your own solution to the project. At the top of your submission, state the names of all your collaborators. It will be your responsibility to make sure that all your collaborators are listed.[1] ***Other than contacting the teaching assistants for clarification, you may not seek the assistance of other persons.***

Grading Criteria:

Your computer program will be tested with an input file similar to the format used in "weather.mat" provided with the assignment.

- Tasks 1 to Task 24
  - 5 points each task, for correct solution and approach.
- Task 25 and Task 26
  - 10 points each.
- For any task that does not execute, or executes with error, the maximum grade you can receive for that task is 20% of the maximum points (see above) for that task.
  - Tasks 1 to 24: maximum of 1 point, if task does not run/run with error.
  - Tasks 25 and 26: maximum of 2 points, if task does not run/run with error.

No late submissions will be accepted and no resubmissions will be accepted for credit.

[1] In the grading of the assignment the listed names may be cross checked. So, if you assist someone, make sure that that person agrees to list you as one of the collaborators, and that you list the person on your list. For this reason, it may be the easiest to form collaborative teams of three, with the understanding that discussions outside of the group will not be allowed.

Project Statement:

Renewable energy is available in many types: wind, rain, solar, bio-fuel, ocean tides, and many more. The effectiveness of a particular type of renewable energy depends on many factors, and one big factor is the geographical location. For example, installing a wind farm at a city with a daily average wind speed of 1 mph is not nearly as effective as installing a wind farm at a city with daily average wind speed of 20 mph. Another example is solar panels. Using solar panel arrays at a cloudy city as the main source of renewable energy is not cost effective.

Your task is to investigate the feasibility of using wind versus sunlight as the main source of renewable energy in various cities in the United States. In particular, you'll be classifying each city according to whether solar or wind energy should be used. Topics related to precipitation will also be explored.

In order to develop your program, weather data for the United States will be used. The information that you will process is already read into Matlab format. The steps to access the information are:

Step 1:
Place the file "weather.mat" from ZIP package, and place it in your *current directory*. Otherwise, Matlab won't know where to find the file.

Step 2:
In Matlab, type "load weather.mat". You'll now have five matrices, **solar, wind, precip, city, and state. solar**, **wind**, and **precip** contain solar insolation, wind speed, and precipitation data for select cities in the United State, respectively. Each of the rows represents a different city. The city and state name can be found in the matrices **city** and **state**. See more information about each of the matrices below:

solar, wind, precip:
- Column 1 to Column 12: data for January, February, …, and December

city, state:
- **city** contains city names. Each row is a string (character array) with length 14. For city names shorter than 14 characters (including spaces), the remaining elements will be empty spaces.
- **state** contains state abbreviations. Each row is a string with length 2.

Example: The 13th row in **city** is 'Dubuque', so the 13th row in **state** is 'IA'. In addition, the 13th row in each **solar, wind,** and **precip** has the solar, wind, and precipitation data for Dubuque, respectively.

Submission Requirements:

You must follow the following submission requirements closely.

- Submit ONE script for Tasks 1 to 24. Name this script **Analyze_Weather.m**
- Submit one script for Task 25, and one script for Task 26. You may assume they will be executed in the following order: Task 25 script, then Task 26 script. Name the scripts **task25.m and task26.m, respectively.**
- Make sure to submit the function (as a .m file, in the same directory as all other files) for Task 23. The file should be called **sum_square.m.**

- **ALL files above** must be submitted in one "zipped" folder. Name it as "Proj1_XXX.zip" where "XXX" are your initials.
- Include myWeather.mat inside the zipped folder.
- The "zipped" folder must be submitted through SmartSite.

It will be completely your responsibility to make sure that your zipped folder contains ***all the supporting functions,*** and are all in the same folder. NOTE THAT THIS IS VERY IMPORTANT: **IF ANY OF THE SUPPORTING FUNCTIONS USED IN THE SCRIPT ARE NOT PRESENT IN THE FOLDER, YOU WILL AT MOST BE ABLE TO GET 25% OF THE GRADE FOR THE TASK THAT REQUIRES THE MISSING FUNCTION. In the event that a supporting function is not present, you will be give ONE opportunity to resubmit within 24 hours after receiving notice that there is a problem with your submission. You will not be allowed to submit corrections to your original submission, just correct the submission error.**

Submission Checklist: (Into a single .zip file, Proj1_XXX.zip)

- Analyze_Weather.m
- sum_square.m
- task25.m
- task26.m
- importfile.m
- myWeather.mat

Your Matlab program will need to function correctly (Task 1 to Task 24) using values for ANY DATA SET OF THE SAME FORMAT (i.e. calculate the correct numbers for your data and fill them in).

Your submissions will be graded by machine using a different data set from the one provided.

- You may assume that the number of columns and the designation of each will be the same.
- You may also assume that the names of the variables (**solar, wind, precip, city** and **state**) remain the same.
- You may not assume that the number of rows will stay the same.
  For example, the number of cities in **precip**, **solar** or **wind** can increase. Thus, your program will have to find which row contains a particular city. To do this, try using the following command: find(strcmp(states,'Dubuque')). For all outputs, use fprintf and make sure you display the result with 0 places after the decimal point. I suggest copying and pasting the outputs so that you don't spell anything incorrectly. Below, are sample outputs with fake numbers; your text should be the same, the numbers (or sometimes state names) different!
- In producing output you must use the fprintf (and not a series of disp(x) functions). The output to each task should start off with an output that indicates which task it is fulfilling, and then text should follow as described below. For clarity, between the different tasks output at least 3 blank lines.

<u>Important: you may not use loop constructs unless otherwise noted.</u>

<u>For Task 1 to Task 24 below, create a new MATLAB script.</u> (Analyze_Weather.m)

The database "weather.xlsx" contains solar, wind, and precipitation data for 54 cities in the United States, which is the same data contained in "weather.mat". Open "weather.xlsx" (or "weather_excel97_03.xls", or "weather_openDoc.ods". All contain the same data, just different format.) using spreadsheet software such as Microsoft Excel or OpenOffice (free online download) and familiarize yourself with the database before moving on. <u>In Task 1, we will learn how to import the Excel file into Matlab.</u>

Each type of data is stored as a worksheet in the database. Each worksheet has 12 columns where column 1 is January, and column 12 is December. Worksheet 1 contains city information and worksheet 2 contains state information. As previously described, rows in the city and state worksheets correspond to rows in other worksheets. For example, row 20 in city and state worksheets is 'Boston' and 'MA', respectively**,** so row 20 in the solar, wind, and precip worksheets contains data for Boston.

Task 1: Follow steps below to import the database (weather.xlsx) into MatLab using the Data Import Wizard).

Step 1: In Matlab, go to File-> Import Data. In the window that shows up, navigate to and select the file, "weather.xlsx". Click Open. A window will show up. Move on to Step 2.

Step 2: In this window, you can select which worksheet from the Excel file to import. Select the "city" worksheet. The right side of the window shows the content of the worksheet. Move on to Step 3, but don't click "Next" yet.

Step 3: Check the checkbox, "Generate MATLAB code". This tells MATLAB to generate code (a function) that will perform the task you are currently completing. Click "Next". The code will appear in a pop-up window at the end of all these steps.

Step 4: In this window, the left side shows information (Name, Size, Bytes, and Type/Class) about the variables that will be imported into the MATLAB workspace. The right side shows the contents of the variable. Again, check the checkbox, "Generate MATLAB code". Click "Finish".

In the MatLab workspace, you now have a variable "textdata" that contains the information in worksheet 'city. Type 'textdata' at the command prompt to see what's in the variable. This variable is a cell array, we will convert this back to an array of strings in Step 6.

Note: if numerical data is contained in the worksheet you wish to import (i.e. solar/wind/precip), Matlab will store the imported data into a variable 'data', instead of 'textdata'.

Step 5: The MATLAB Editor window now pops up, containing the function "importfile()" which imports one worksheet, the 'city' worksheet from the Excel file into MatLab. Save this function as a .m file with the same name of the function.
Note: the input to this function is the name of the database file. The name of the worksheet to import is assigned to the variable 'sheetName'.

Step 6: Your job in this step is to modify the "importfile()" function such that

1) The name of the worksheet (as a string) is passed into the function as an input argument, in addition to the file name, and
2) convert text data from cell arrays into character arrays. See sub-bullet below.
    a. In the if statement that checks for non-empty strings, replace the variable 'strings' by 'char(strings)'". This converts the contents of 'strings' into a character type. In this case, character arrays, or strings.

Example: typing the command "importfile('weather.xlsx', 'solar')" in the command window will import the worksheet "solar" from the Excel file "weather.xlsx" into a variable named 'data'. Typing the command "importfile('weather.xlsx', 'state')" will import the "state" worksheet into MatLab, into a variable named 'textdata'.

IMPORTANT: The Excel file and "importfile.m" must be in the Current Folder of MatLab.

Step 7: Use the modified function to import the worksheets 'city', 'state' 'solar', 'precip', and 'wind' from "weather.xlsx" into MatLab, and assign the imported variables to **city, state**, **solar**, **precip**, and **wind**, respectively.

Note: Numerical data is always imported into 'data' while text data is always imported into 'textdata'. When importing consecutive worksheets with numerical data, you must assign 'data' to another variable before importing the next worksheet, otherwise the previously imported data will be over written by the newly imported data.

Step 8: Select the five variables **city**, **state**, **solar**, **precip**, and **wind** by using control+click. After selecting all variables, right click, and select "Save As". Save it as "myWeather.mat". This file should be identical to "weather.mat". You will submit "myWeather.mat".

You will use the five variables inside myWeather.mat to complete all remaining tasks.

Task 2: Include the lines of code below at the top of your script.

```
% Your Name
 % Your UC Davis ID Number
% Your Section Number
% Collaborators (List Full Names, maximum of two)
%
load myWeather.mat % You may use the provided Matlab database, "weather.mat" if you are
                   % having trouble completing Task 0, and would like to go ahead and work
                   % on remaining tasks.
```

Task 3: In total, how many <u>states (not cities!)</u> exist in the database? You may use a loop in this task. However, 2 points extra credit for solution that does not use loop in this task.
Display your result as:
"In total, 53 states exist in the database"

Task 4: How many cities in the database have a name longer than 10 characters (A city name can only include one blank space. It's name can have upper, lower, or mixed case letters)? You may use a loop in this task.
Display your result as:
"34 cities in the database have names longer than 10 characters."

Wind energy problems

Task 5: Which city experiences the most wind throughout the 12 months? The least? (Calculate the average yearly wind speed)
Display your result as:
"Seattle experiences the most wind while Boise experiences the least wind."

Task 6: What percentage of cities has average yearly wind speed greater than 8 miles per hour?
Display your result as:
"35 percent of cities have yearly wind speed greater than 8 mph."

Task 7: Which city has the highest percentage of months that have wind speed greater than 8 miles per hour? What is that percentage? You may use loops to help you solve this problem.
Display your result as:
"Billings has the highest percentage of months with wind speed greater than 8 miles per hour. 48 percent."

Solar energy problems

Task 8: Which city has the greatest monthly solar radiation? During which month?
Display your result as:
"Washington has the greatest monthly solar insolation in July."

Task 9: How many cities have average yearly solar insolation greater than 4 kilo Watts per square meter?
Display your result as:
"19 cities have solar insolation greater than 4 kilo Watts per square meter."

Task 10: Which city has the largest solar insolation during July? How about December?
Display your result as:
"Anchorage has the largest solar insolation in July, while Sacramento has the largest in December."

Task 11: Strong wind (wind speed greater than 3mph) damages solar panels. List the top three cities where solar panels are most likely to be damaged. Assume only cities with solar insolation greater than 4 kilo Watts per square meter have solar panels installed.
Display your result as:
"In order of likeliness to damage solar panels: Birmingham, Anchorage, and Little Rock."

Rain water related problems

Task 12: A rain water collection system with a surface area of 10 square meters is used in each city. Using this rain water collection system, which three cities can collect the most rain during February? How about in August?
Display your result as:
"San Francisco, San Jose, and Sacramento experience the most rain in February while New Orleans, Honolulu, and Austin experience the most rain in August."

Task 13: A rain water collection system with a surface area of 10 square meters is used in each city. Assuming wind speed below 3mph reduces the collection amount by 5%, speed from 3 to 5 mph reduces the collection amount by 10%, and speed above 5 mph reduces the collection amount by 30%. Which city collect the most rain water in a year?
Display your result as:
"Accounting for loss due to strong wind, Seattle collects the most rainwater in a year"

Task 14: In addition to the strong wind problem in Task 11, cities with monthly precipitation above 3 inches are also likely to have its solar panels damaged. (Again assume only cities with solar insolation greater than 4 kilo Watt per meter square have solar panels installed)
Taking rain and wind into account, list the top three cities where solar panels are most likely to be damaged.
Display your result as:
"In order to likeliness to have solar panels damaged due to rain: Birmingham, Anchorage, and Little Rock"

Solar to Wind Energy Comparison

Task 15: Assume 1.8 kilo Watts of power is available from a wind turbine that experiences a wind speed of 3 miles per hour. Cities with wind power greater than solar power should install wind turbines. How many cities should install wind turbines?
Display your result as:
"32 cities should install wind turbines."

Task 16: Cities with average yearly solar insolation greater than 4 kilo Watt per square meter should make use of solar power. How many cities should use both solar and wind power?
Display your result as:
"32 cities should install both solar panels and wind turbines."

For tasks 17 to21, consider the below: (Precipitation data is not used in these tasks)

A city is considered
1) **SOLAR SUITABLE if the total solar insolation in a year can generate more than 55 kW of power (55 kW is the upper threshold for solar power).**
2) **WIND SUITABLE if its total wind speed in a year can generate more than 70kW of power (70 kW is the upper threshold for wind power).**
3) **SOLAR UNSUITABLE if the yearly output solar power is below 40 kW (lower threshold).**
4) **WIND UNSUITABLE if the yearly output wind power is below 40 kW (lower threshold).**

For either power source, if the yearly output falls in between the upper and lower thresholds, the city is classified as ADEQUATE.

Below are general equations for tasks 17 to 21:
Solar power in one year = yearly average solar insolation*12months
Wind power in one year = yearly average wind speed*12months*1.8kW/3mph

Task 17: Using conditions described above, find the number of cities that are UNSUITABLE to use both wind and solar power. Use a loop to list those cities (city).
Display your result as:
"3 city (cities) are UNSUITABLE to use both wind and solar power. San Francisco, Chicago, New York."

Task 18: Using conditions described above, find the number of cities that are SUITABLE for at least one type of renewable energy source.
Display your result as:
"24 cities are SUITABLE for at least one type of renewable energy."

Task 19: Using conditions described above, find the number of cities that are SUITABLE for ONLY ONE type of renewable energy source. (Hint: which logical operator only allows one operand to be true?)
Display your result as:
"15 cities are SUITABLE for ONLY ONE type of renewable energy."

Task 20: Using conditions described above, find the number of cities that are ADEQUATE for at least one type of renewable energy source.
Display your result as:
"26 cities are ADEQUATE for at least one type of renewable energy."

Task 21: Using conditions described above, find the number of cities that are ADEQUATE for both solar and wind power.
Display your result as:
"23 cities are ADEQUATE for both solar and wind power."

Relationship Between Solar and Precipitation

Dataset obtained from performing an experiment tells us how the experiment behaves in the past, during measurement. By calculating the mean and standard deviation, which are statistical properties, we can better understand the experiment.

Another important statistical property is the correlation. Correlation tells us how one dataset behaves relative to another dataset. For example, if data values in both datasets increase with time, by the same amount, then the correlation between the two datasets is 1. On the other hand, if data values in one dataset increase in time while data values in the second dataset decrease in time, the correlation between the two datasets is negative. You will learn how to calculate correlation in the remaining tasks.

Example, dataset1=[1 2 3] and dataset2=[9 10 11] have a correlation of 1, and the two datasets c=[8 9 10] and d=[5 4 3] have a correlation of -1.

Task 22: **Write code to plot the solar and precipitation data for the 12 months, for the following three cities on the same plot. Cities: Miami, Seattle, and Boston. Your plot should be formatted using the following specifications:**
1) plot the solar data using dashdot style lines, and the precipitation data using dotted style lines,
2) plot data for San Francisco using the color Red, Seattle using Blue, and Boston using Magenta, and
3) for all, use line width of 2.5.

HINT: use MatLab's help command to find information on 'plot'.

Task 23: **Create a function called 'sum_square', that takes two 1-D vectors with equal lengths as input arguments, and returns a scalar as output. This function implements the equation below. We will use this function in the next task to compute correlation.**

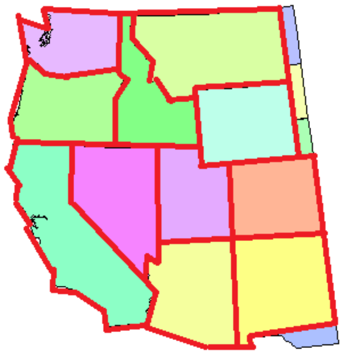
$$out = \sum(\mathrm{x} * \mathrm{y}) - \frac{(\sum \mathrm{x})(\sum \mathrm{y})}{\mathrm{n}} \quad where\ n\ is\ the\ number\ of\ elements\ in\ vector\ x$$

The notation, $\sum \mathrm{x}$ means the sum of all elements in vector x.
The notation, $\sum(\mathrm{x} * \mathrm{y})$ is the sum of the element-wise product of vectors x and y.

Task 24: **The correlation between two datasets, x and y can be computed using the formula below. Write code to find the top three cities whose correlation between solar and precipitation data is closest to -1.**

$$correlation, r = \frac{\mathrm{sum\_square}(\mathrm{x},\mathrm{y})}{\sqrt{\mathrm{sum\_square}(\mathrm{x},\mathrm{x}) * \mathrm{sum\_square}(\mathrm{y},\mathrm{y})}}$$

In tasks 25 and 26 (exercises on graphical input), we will display a map of the Western United States (see below), wait for the user to click on any state (we will only allow the user to select the 11 states highlighted in red below) on the map, and display the solar radiation for the selected state.



Task 25: In this task, we will first write a script (task25.m) to create a database of coordinates for the 11 states. Follow the steps below to complete this task. You may use loops here.

Step 1: Download and place the image 'PartUSA.tif' in your MATLAB's working directory.
Step 2: Use' imread()' to load the image, and 'imshow()' to display the image.
Step 3: Now write a loop and use 'ginput()' to create bounding boxes for the 11 states in the image. <u>Obtain at least 6 coordinates for each bounding box.</u>

Definition: A bounding box is a polygon that surrounds, as close as possible, to the area the box encloses. Example, to completely enclose an octagon, we only need to use a bounding box with 8 points/sides/coordinates. To completely enclose an irregular area (such as California) with N sides, we need a bounding box with at least N points/coordinates. For this task, obtain at least 6 coordinates for each bounding box. Note that a square, 4 sided polygon, can also be enclosed by a bounding box with more than 4 coordinates – some coordinates will simply lie on the same line.

Step 4: Store the x and y coordinates of the bounding boxes returned by 'ginput()' into two matrices called **Xstate** and **Ystate**. The size of **Xstate** and **Ystate** should be 11 rows (states) by N columns, where N is the number of coordinates.
Step 5: You will now create a vector call **myStates.** **myStates** will have 11 rows by 1 column. Each row in **myStates** will contain a state name, the state name in each row should correspond to the coordinates in **Xstate** and **Ystate**.
For example, if the 3rd row in **myStates** is 'CA', then **Xstate**(3) and **Ystate**(3) should contain the coordinates for California's bounding box.
Step 5: Control+click on **Xstate, Ystate,** and **myStates** in the MATLAB workspace and select "Save As". Save the variables as StateInfo.mat.
This completes Task 25.

Task 26: For this task, open a new MATLAB script and save it as task26.m. In your script, we will 1) display the map, 2) obtain an input from the user, 3) check if the input lies inside the boundary of any of the 11 states, and 4) displays the results accordingly in the MATLAB command window.
If the user's input is invalid, continue until a valid input is entered. Once a valid input is obtained, display the state name. Follow the steps below to get started. You may use loops for this exercise.

Step 1: Load StateInfo.mat created from **Task 25**.
Step 1.5: Load the provided database, weather.mat.
Step 2: Use 'imread()' and 'imshow()' to load and display the map.
Step 3: Use 'ginput()' to obtain a single coordinate input from the user.
Step 4: Use 'inpolygon()' to check if the input from user in step 3 lies inside any one of the 11 states. Note: The function 'inpolygon(x,y,xb,yb)' returns 1 if points in (x,y) lie inside the bounding box given by points in (xb,yb).
Step 5: If 'inpolygon()' from Step 4 returns 0, display an error message and ask the user to click again.
Step 6: If 'inpolygon()' from Step 4 returns 1. Look up and display the state name, and the yearly average solar insolation and wind speed for each city in that state.

See the sample output below for this task, which shows the case for two cities in California.

```
"You selected CA."
"Los Angeles, CA has 3.4 Watt-hour per square meter of yearly average solar insolation."
"Los Angeles, CA has 2.1 miles per hour of yearly average wind speed."
"San Francisco, CA has 7.2 Watt-hour per square meter of yearly average solar insolation."
"San Francisco, CA has 4.9 miles per hour of yearly average wind speed."
```