ENG06 Project #2 (Final Project)

Collaboration Policy:

You need to form a team of three to complete one of the project options. You are only allowed to talk and collaborate with members within your team. Team members are expected to equally participate, and collaboratively work towards the completion of the project. ***Other than contacting the teaching assistants for clarification, you may not seek the assistance of other persons.***

Logistics

For this project, you will form a team of three. There are four project options to select from. Each team will select one topic to work on. **Team and topic cannot be changed after 2/22/2013.**

Item 1: One individual (each student will submit their own) preliminary report is required. It will be due on Friday 3/1/2013 by 5PM on SmartSite. This is a 1 to 2 page report where information on you and your team's plan to complete the project is outlined. This report must contain two things, as numbered below:

1*) Team's plan.* Consider using a table similar to the one below (add more rows as needed). However, your team may use any other method, as long as the information is clear.

Milestone description	Tasks that need to be completed	Responsible team members	Date of completion
GUI and IO interface	Draw grid, user input interface, GUI page transitions	Alice	3/8/2013
…	…	…	…
Integrate all parts	Integrate all components together into a single program	Alice and Bob	3/13/2013
Test and debug program	Test program using different test cases. Fix bugs.	Alice, Bob, and Carl	3/18/2013

2) *Your plan* (not your team mate's plan). Write a few sentences detailing how you will achieve your assigned milestones and the associated tasks. Consider using tables to help you explain.

Item 2: The completed project will be due on Tuesday, 3/19/2013 at 5PM on SmartSite. Submit the following:

- A ZIP package containing:
  - All files required to run your program,
  - README file containing Instruction on how to run your program (which file does what)
  - The report (.PDF) – don't forget to include the YouTube link in your report.

- In addition, e-mail the YouTube video link to the TA responsible for your team. E-mail title "ENG6: Final Project YouTube Link - <team name>"

It is you and your team's responsibility to provide a complete set of files to run and test your final program. It will also be important to give clear instructions on how to run your program. This could be done in various ways, depending on the problem that you choose. One good way to document how your program executes is to give an example case and the results.

Grading Criteria:

<u>The projects are open ended. As long as your program can perform the assigned tasks, there will be no correct or incorrect approaches.</u> Certainly there will be more acceptable and attractive solutions, and that will be judged in comparison with competing solutions submitted in the same category. Projects will be judged against the others project in the same group that have been submitted.

The expectation is that each team member must take responsibility for a specific aspect of the project and being able to explain what their contributions to the project have been.

The grade of each team member will be adjusted according to how the project tasks were delegated and who was responsible for what aspects of the project. Each project will allocate at least 10% of the grade to a section that must be included as Appendix A. **Appendix A must contain:**

- A table with the breakdown of the tasks to complete the project, and who was responsible for what part of the project. The intent here is to determine who did what to implement the project. While it is perfectly reasonable that some tasks can be completed jointly, it is unrealistic to claim that that everyone worked together *on all aspects* of the project equally.
- Each member must provide a brief personal summary of what the person's involvement and contributions. Before the project is submitted, the summaries must be provided to all members for review and comment. Appendix A must conclude with the following statement:

"All team members have read the task summaries contained in this report and have been given an opportunity to comment. "

Project Report Requirements:

Each project submission must have a project report that contains any relevant material deemed essential, and it must contain an Appendix A as described previously. The first page of the report must contain the team name and the names of all members. This file must be saved a PDF document.

The report must provide clear explanations on how your program works, and how problems encountered in the project are solved. Use flowcharts or diagrams to help illustrate the program's functionality.

The length of your report is what your team feel is required to fully document and explain your team's work. It is good to keep the report to within a reasonable length, while being concise.

In the event that you have used external resources, you must provide appropriate credit in the project report or in your program code. If it is discovered that you have borrowed or used material from a source that is not credited, it will be consider plagiarism and the case will be turned over to Student Judicial Affairs.

Youtube Video Requirements:

The format of the video is entirely up to your team as long as the following criteria are met:

- Maximum length of the video is 10 minutes
- Each team member must be seen in the video to present their part of the work and contributions
- A clear and easy to follow demonstration that shows the correct functionality of your program

Use visual aides to help explain your steps. (white board, markers, poster, etc)

Project Option #1: Sustainable Urban Planning

For this project, you will be making a program that helps engineers and architects to develop and deploy renewable energy for a small area, in particular solar and wind.

Getting started:

Select one of the cities in the database used in Project 1. Then go onto Google maps. Go to that city, find an "appropriately sized" (see details below on choosing the area) area, and take a snapshot of the aerial image of the area. Save the snapshot as an image.

Selecting an "appropriately sized" area:

- The area you select should be appropriately sized such that solar panels or wind turbines can be realistically placed in a box on the grid. The size of the area you choose also impacts the grid size: too large an area results in very small boxes on the grid while too small an area results in large boxes on the grid.
- Each box on the grid can represent one solar panel, or multiple. It is up to your team. But should choose the dimension of the box considering the panel/turbine dimensions.
- Figure below is a good example. It is part of downtown Davis and the street block labeled is approx. 400 feet. The length of each box is around 20 feet (there are about 20 boxes). Each 20ft x 20ft box in this case contains approx. 16 to 25 solar panels (ignoring the dimensions shown in table below).
- Experiment with different area/box sizes! However, large area results in large grid with too many small boxes, which can slow down your program and potentially exhaust the available memory.

Task: Your GUI should meet the following criteria:

1) Allow user to select from ***three*** different cities (from the cities in database from project 1).
2) Select the type of solar panel or wind turbine to place (each item here has a color. Assign the color/shading of your own choosing) Consider only the solar panels and wind turbines in two tables on the next page.
    a. Operation: The user first selects an item (solar panel or wind turbine), then clicks anywhere inside a box on the grid to place that item. Your program should automatically fill in the box with a corresponding color. (operation is similar to MS Paint where you first select a tool, then use that tool to perform an operation on the canvas)
3) Repeat step 2) as long as the user wishes. There should be a clear way for the user to tell the program that he/she has finished.
4) When the user finishes, your program should present the following options to the user:
    a. Save as a picture (a .png picture of the result)
    b. Save as text output (a N by M text file containing the content of each box in the grid. - N and M are the dimension of your grid. You decide the text file format.)
    c. Create BOM. (BOM - "Bill Of Materials" contains the quantity and types of panels, turbines, and total cost.)
5) Solar Energy Estimate: using the total area occupied by the solar panels together with database from Project 1, obtain an estimate for the total amount of potential solar power available. Assume solar panels in table below are capable of producing the amount supplied by the sun. Solar data is in kW per square meter.

6) Wind Energy Estimate: using the total number of wind turbines with database from Project 1, obtain an estimate for the total amount of potential wind power available. Assume for every 1mph of wind, 500 kW of power is produced.
7) Add a feature into your program that allows the text file saved in 4b) to be loaded into your program (with the proper image, of course). In a way, this feature allows users to save their work and continue working on it later.

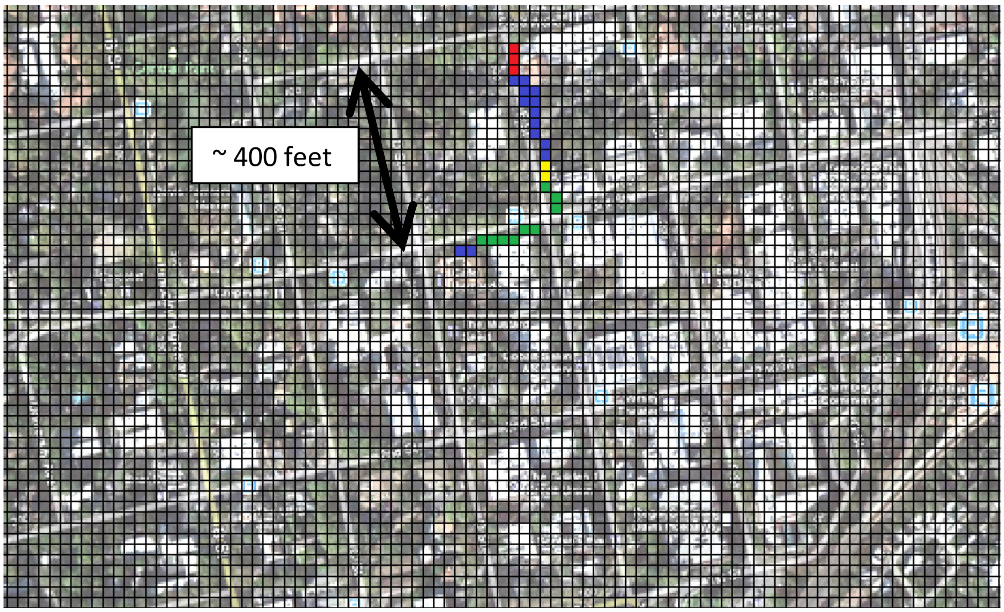
Table of Solar Panels. (Data from www.amazon.com)

Brand	Model	Type	Dimension (inch x inch x inch)	Weight (lbs)	Price ($)
Sunforce	50048	Amorphous Silicon	42.5 x 1.5 x 16	11	279.95
Sunforce	39810	Polycristalline	21 x 48 x 2	22	499.95
Instapark	SPCC-5W	Mono-crystalline	11 x 8 x 1	2.8	34.95
Instapark	SP-100W	Mono-crystalline	45 x 1.5 x 26	21	319.99
Instapark	SPCC-30W	Mono-crystalline	21.5 x 1.1 x 17.2	7.2	114.70
Instapark	SP-10W	Mono-crystalline	14 x 11 x 1	2.8	39.95
Ramsond	100SP	Mono-crystalline	47 x 1.5 x 21.8	12	245.99
Epcom	WK50-12	Polycrystalline	32 x 22 x 1.4	12	99.99
Sun Power	E18	Mono-crystalline	41.18 x 81.36 x 2.13	56	249.5
Sun Power	T5	Mono-crystalline	43.06 x 75.13 x 8.37	47	199.99

Table of Wind Generators/Turbines. (Data from www.amazon.com)

Brand	Model	Type	Diameter (feet)	Price ($)
Windmax	HY 1000-5	Wind Generator	15	999.99
Windmax	HY400	Wind Generator	13	686.40
GudCraft	WG400	Wind Generator	13	399.00
GudCraft	WG700	Wind Generator	13	449.00
All Power America	APWT400A	Wind Generator	10	476.93
Sunforce	45444	Wind Turbine	10	749.99
Sunforce	44444	Wind Generator	10	474.34
WindyNation	WCK-750	Wind Turbine	15	999.98

Figure: Part of downtown Davis overlay with grid. Filled boxes indicate solar panels. Colors indicate solar panel/wind generator type.



Project Option #2: Solar Survey

Use the Arduino hardware module to perform a solar survey of the UC Davis Campus. Each team member will receive an Arduino Module.

TASK 1: Create a GUI to help with the survey process. Your GUI should:

- Displays a map of the area (screenshot from Google maps is OK)
- Allow the user to indicate their location, and the time/date (see 'ginput()'). Store this information.
- After the user has finished recording the location, and time/date, the Arduino hardware should start the trials. Be sure to allow the user to reposition the hardware module in between trials. See below for what should be recorded in each trial.

Select a set of locations/buildings on campus. Find a spot that seems to have the most sunlight and perform a measurement.

Measurement Guidelines:

- ***At least 18 buildings/locations*** (6 buildings/locations per team member)
- For each building/location, perform a measurement at ***three different times of the day***: morning, noon, and late afternoon.
- For each measurement, pick a spot where you think receives the most sunlight at the location. Be sure the module is positioned so the solar cell receives the most sunlight. Then perform the three trials. (to minimize measurement/positioning errors)
  For each trial, perform:
  1) Load sweep: sweep the variable resistor 0 from resistance code 0 to 255, while keeping variable resistor 1 at resistance code of 128. Record the voltage at each resistance code.
  2) Open circuit voltage sweep: Remove the open circuit jumper. Set both variable resistors to resistance code of 255. Record 256 data points.

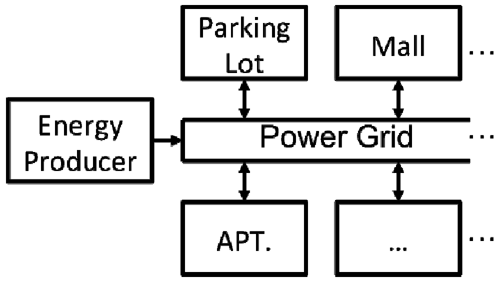
TASK 2: Create a GUI to plot (see notes on making contour plots) the data measured in Task 1. Your GUI should allow the user to:

- Select the data to plot:
  1) raw data from the three measurement times you took the data, or
  2) average across the three measurement times.
- Select whether to display the plot using two methods:
  1) Maximum Power Point: calculate using the data from "Load sweep" trial. For each data value, calculate the power by squaring the data value, then dividing the result by the corresponding resistance value. Calculate the power of all data points. The maximum power point is simply the largest power value.
     Maximum power point = max(data point^2/load resistance)
  2) Open Circuit Voltage

Note: For both plot methods, you will need to interpolate the data at locations where data is not available.

Project Option #3: Sustainable Transportation System Simulator

Just about everything that uses electricity (Energy Consumers) needs to be connected to the power grid at some point. The power grid is also connected to the energy production plant (Energy Producer), which supplies electricity to the grid. See figure below.



In your simulator, the power grid is connected to various energy consumers similar to the figure above. You will design a simulator that simulates the EV traffic at each energy consumer location, and estimate the total power demand from the power grid, and finally compute and display the required scale of the renewable energy source to meet the demand. Below are the energy producers and consumers, and their consumption rates (shown in corresponding tables) you will use.

ENERGY PRODUCERS:

- Hydroelectric Dams.
- Solar Panel Array
- Wind Farm

Energy Producer	6am-11am	11am-3pm	3pm-7pm	7pm-12am	12am-6am
Solar	20% of average yearly value (AYV).	50% of AYV	90% of AYV	20% of AYV	10% of AYV
Wind	10% of AYV	20% of AYV	40% of AYV	60% of AYV	70% of AYV

ENERGY CONSUMERS:

- Parking Lot
- Shopping Mall
- Apartment Building (APT.)

Energy Consumer	6am-11am	11am-3pm	3pm-7pm	7pm-12am	12am-6am
Parking Lot (500 EV spaces)	60% EVs charging	80% EVs charging	30% EVs charging	10% EVs charging	5% EVs charging
Shopping Mall (1000 EV spaces)	20% EVs charging	60% EVs charging	70% EVs charging	60% EVs charging	5% EVs charging
Apartment Building (200 EV spaces)	20% EVs charging	10% EVs charging	60% EVs charging	70% EVs charging	90% EVs charging

Tasks: In this project, you will create a simulator that

1) Estimates the number of electric vehicles that can be supported by a particular type of renewable energy producer (solar, wind, hydroelectric),
    - To estimate the power capable of being produced by each energy production mode (solar, wind, hydroelectric), research (Wikipedia is fine) the top energy plant for each EP mode, and use the numbers you found.
2) Simulates the (random) activities of the electric vehicles moving in between locations (mall, parking lot, apartment building) throughout the day,
3) Outputs (as charts, wrt. time, as the simulator is running):
    - Real-time total power demanded by the EVs, and
    - Real-time Solar/wind farm size, or hydroelectric dam height.
4) Displays a warning if the power demanded by the EVs increases to within 10% of the power being supplied.
    - If the above condition occurs, your GUI should automatically increase the scale of the solar/wind farm, or the hydroelectric dam height.
5) While the simulator is running, the user should have the option to
    - increase or decrease the number of EVs.
    - Add or take away an energy consumer. (there must be at least two energy consumers in the simulator)

GUI Design - Inputs:

Design a GUI that allows the user to input the following:

- City (one of the cities from the database used in Project 1)
- Month (to start simulation with)
- Desired mode(s) of renewable energy producer (solar, wind, or hydroelectric). User can specify more than one.
    - If the user specified more than one, you need to ask the user how to allocate the demand between the different EP modes

After obtaining the above from the user, your GUI should estimate and display the number of electric vehicles that can be supported by the selected EP mode(s). You should underestimate (by 15%) the number of EVs, in case the power demand increases slightly. This estimate is the baseline.

Now the GUI should have some mechanism that allows the user to start the simulation. As the simulation runs, charts will be displayed, see the next section for more detail.

GUI Design – Outputs:

Your simulator will display a chart plotting the real-time total power demanded from the power grid as a function of time (y-axis is the total power available, and x-axis is time), and another chart plotting the scale/size of the solar/wind farm or the height of the hydroelectric plant. Both of the above depends on the number of EVs in the simulation, and their current location. Follow the guidelines below.

1. The simulation starts with the baseline estimate of EVs.
2. Randomly decide the initial speed and the location of each EV.
3. Randomly assign a duration that each EV remains at its present location.

4. Each EV randomly moves to its next location. Again, there should be some mechanism on the GUI that allows the user should be able to increase/decrease the total number of EVs in the simulation, while the simulation is running.
5. When the power output of the EP reaches within 10 percent of the energy being consumed by the EVs, your simulator should present a warning to the user, and automatically increase the scale of the solar/wind farm size, hydroelectric plant height, volume, or decrease the number of vehicles. (Bullet No. 4 under ***Tasks***)
6. As the simulator is running, you should allow the user to (Bullet No. 5 under ***Tasks***)
    a. increase/decrease the number of EVs in the simulation.
    b. Add or take away an energy consumer. (there must be at least two energy consumers in the simulator)

General Hint: You should base your approach on the bouncing ball simulator discussed during the object-oriented lecture. Edit the greenball.m and greenball2.m files to instead simulate the random traffic at each energy consumer locations.

Project Option #4: Commodities Market Analysis

For this project, you will be making a program to a user to select and display historical information from the commodities market. If the user select from a given set of commodities, the program will analyze data from the commodities market, and then make predictions based on past performance. "Past performance" will be based on data that is made available on a monthly basis by the International Monetary Fund (IMF). The IMF monthly publishes a chart containing the commodities indices and prices for every month from January 1980 until the past month. For example, let us take Sept 14, 2012. Your program will use information in the IMF database[1] up to August 2012, perform an analysis, and predict what a user selected commodity price or commodity price index will be on Sept 14, 2012. You program will then also retrieve the latest commodity price from a website, and compare your predicted value with the actual price. The expectation is not that you will be able to predict the price with great accuracy (and if you can, that will be great.......), however you will be required to implement a rational approach using a moving average.

THERE WILL BE MANY POSSIBLE WAYS THAT INFORMATION CAN BE COLLECTED FROM THE USER, AND MANY WAYS THE INFORMATION CAN BE PROCESSED, AND PRESENTED. THERE IS NOT ONE CORRECT ANSWER. VIEW IT AS YOUR TEAM PRODUCING A PRODUCT THAT WILL BE COMPETING WITH A SIMILAR PRODUCT PRODUCED BY A COMPETING COMPANY. IF THE USER HAS A "GOOD" EXPERIENCE, YOUR

---

[1] The IMF database is in .csv (comma separated value) format and can be downloaded from the following website:

IMF Commodities Database

It can also be found at www.imf.org/external/np/res/commod/index.aspx (it is at the bottom of the table, you should see the link: monthly data (CSV file).)

The data is formatted as follows (as seen if you open the file with Microsoft Excel):

- Column A, rows 10 – 400 contain the year and month for the data in the corresponding rows. The date is in the format yyyyMmm –> 2012M07 means July 2012.
- Row 6, columns B – BL contain the units of measurement – Index number, US Dollars per metric ton, etc.
- Row 9, columns B – BL contain the description of the commodity index (a group of similar commodities, i.e. the metals index represents gold, silver and all other metals) or the individual commodity (i.e. Aluminum, 99.5% minimum purity, LME spot price, CIF UK ports).
- Rows 10 – 400, columns B – K contain the price for commodity indices by year and month.
- Rows 10 – 400, columns L – BL contain the prices of individual commodities by year and month.

COMPANY WILL BE SUCCESFULL, IF NOT YOU WILL GO OUT OF BUSINESS. IT WILL ALWAYS REMAIN UNKNOWN WHAT QUALIFIES AS A "GOOD" EXPERIENCE UNTIL PRODUCTS ARE AVAILABLE.

Tasks – Your team must incorporate all the following options into a GUI:

1. A menu that gives the user the option of analyzing either commodity price indices or individual commodity prices.
2. A menu that lists all the options based on the choice the user made in step 1. For example, if the user chose "Commodity Price Indices", this menu should list "All Commodity Price Index", "Non-Fuel Price Index", "Food and Beverage Price Index", "Food Price Index", "Beverage Price Index",..., up to and including "Crude Oil (petroleum), Price Index". If the user enters "Individual Commodity Prices" then the menu should list every individual commodity in the menu. You do not have to list the entire title, i.e. "Coffee, Other Mild Arabicas, International Coffee Organization New York cash price, ex-dock New York, US cents per pound" or "Coffee, Robusta, International Coffee Organization New York cash price, ex-dock New York, US cents per pound"; you can simply list enough to distinguish what the commodity is – "Coffee, Arabicas" and "Coffee, Robusta".
3. Based off the user's selection from the menu in step 2, report to the user the years of data available for the user's choice (not all options have data for all months and years).
4. Present the user with a menu of types of graphs to choose from. Your program must offer all of the following (at a minimum):
   - The option to plot the price from a single month, i.e. January, for every year in the database that contains a January price. You will need to present a sub-menu for the user to choose which month he or she wants plotted.
   - The option to plot the price for every month over a range of years specified by the user. You will need to find a way to ask the user the years she/he wants plotted and make sure the user knows which years of data are available for the given commodity price index or commodity price.
   - The option to create a bar chart for every quarter (Jan-Mar, Apr-June, July-Sept, Oct-Dec) for a range of years specified by the user. Calculate the average, max and min for each quarter to create the "bars" on the bar chart. You don't need to list all the stuff you see on the graph displayed in the hyperlink above, but you will need to add a title, etc., see #5 below. You will need to find a way to ask the user the years she/he wants plotted and make sure the user knows which years of data are available for the given commodity price index or commodity price. You may find the MatLab function `boxplot` useful for making the plot.
   - The option to create a bar chart for every year over a range of years specified by the user. Again, you will need to find a way to ask the user the years he/she wants plotted and make sure the user knows which years of data are available for the given commodity price index or commodity price.
5. Every plot MUST have a title listing the commodity price index or commodity price that is being plotted and the type of plot (those listed in #4 above) that is being shown to the user (i.e. Soybean Prices, Month = January, Years = 1980 – 2012). It must also have axis labels (with

appropriate units of dimension on the y-axis and clear labels on the x-axis that list month, quarter, year, etc.)

6. The program must offer the option of predicting the future price of the individual commodity <u>if the user has chosen one of the following</u>:
    - Crude Oil
    - Copper
    - Aluminum (LME)
    - Cotton
    - Cocoa
7. Now your program will start making a prediction. To carry out the prediction use an exponential moving average (EMA) on the data (here are some other links about EMA and its use in the stock market: 1, 2, 3, you can find many others using Google.) How the EMA can be used is described in the footnote two.[2]

---

[2] The EMA works in the following way. The rules you should apply when using it on the data:

$$EMA_i = EMA_{i-1} + k * (price_i - EMA_{i-1})$$

where EMA_i is the current (or future) "average" price, $EMA_{i-1}$is the previous "average" price, $price_i$ is the current (or latest) price and k is a weighting factor limited to the range $0 < k \leq 1$.

There are a couple of key things to keep in mind when trying to carry out an EMA. First, you must choose what value to use for EMA_0. You will need to put some thought into this. If you use $EMA_0 = 0$, here is how the EMA evolves (assuming by data by month):

$$EMA_{Jan} = 0 + k * (price_{Jan} - 0)$$
$$EMA_{Feb} = EMA_{Jan} + k * (price_{Feb} - EMA_{Jan})$$

and so forth. For a good choice of k, you should see that the change from $EMA_i$to EMA_{i+1} will get smaller, and smaller as i increases.

The value of *k* that you choose will have a big influence on how accurate your EMA will be. You will need to figure out a method of optimizing the value of *k* that you use in order to get your EMA to be as accurate as possible.

The optimum value of *k* will depend on the number of data points you have to work with before being required to predict a price. A larger value of *k* (close to 1), will be more reflective of short term trends (if you have only a few data points); a small value of *k* will do a better job of picking out long term trends, but it will require a lot of data points to converge to a relatively stable average price.

You will also need to devise a strategy to deal with abrupt changes in price trends, i.e. a very sudden and dramatic increase or decrease in price. You may find that with an optimized value of *k* that these abrupt changes have very little influence, or you might decide to restart the EMA after coming across one of these abrupt changes.

8. From the EMA processed data, a prediction is made. You will need to use curve-fitting in order to obtain a "predicted" price. You may find the functions `polyfit(x,y,n)` and `polyval(x,y)` to be helpful. Polyfit provides the coefficients for an nth order polynomial. You will need to find a way to pick the n that provides the best correlation between the curve-fit polynomial and the EMA.
9. Use your curve-fitting results to generate the "predicted" price for the day the program is running.
10. Now, the program needs to query the internet (check out `urlread`) and use regular expressions to get the "current" actual commodity price index or commodity price on the day the program is running. You can get the current prices from the Wall Street Journal. On this website there will be five columns that list prices; you are to use the price in in bold font printed in the center column. Depending on the day your program queries the website, the name of the column will change, for example on Wednesday the column will be called "Wed Price" and on Friday it will be called "Fri Price". Make sure the units of measurement on the website match those of your data, if not you need to convert the unit of measurement from the website to match the units in your data. Use the specific listings below to compare to your "predicted" price:
    - Crude Oil: compare your predicted price to each of the four domestic crude oils listed.
    - Copper: Copper, high grade: Comex spot price $ per lb.
    - Aluminum: Aluminum, LME, $ per metric ton.
    - Cotton: Cotton, 1 1/16 strand lw-md Mmphs, per lb-U
    - Cocoa: Cocoa, Ivory Coast, $ per metric ton-W

Comments: This project is a team effort. Your team will need to decide who will be responsible for what. As a team you need to identify the subtasks, identify who takes responsibility, how communication within the team will take place, etc. Here are some subtasks for this project:

a) GUI layout
b) GUI programming
c) Reading and extracting information from the latest version of the IMF Commodities Database
d) Data analysis and prediction
e) Retrieving the current prices from the Wall Street Journal website by using regular expressions
f) Making the YouTube video
g) Writing the report
etc........