Initial Version: 1/17/2013

HW2: CS 110X C 2013

Note: This homework (and all remaining homework assignments) is a **partner homework** and must be completed by each partner pair. When you complete this assignment, you must not share your answers with any other student. Only one person from a partner pair needs to submit the assignment.

Q1	Demonstrate input abilities

Skills
PM-1
CS-9
CS-5

Lecture Dependency
Jan-15

Newton's method computes approximations to the roots of a real-valued function. Given a real-valued function f(x) and its derivate f'(x) you start by "guessing" a value, x_0, which is your initial approximation of a root. From this initial starting point, a better approximation, x_1, is computed by:

$$x = x - \frac{f(x)}{f'(x)}$$

You can then iterate this process a number of times, computing increasingly accurate approximations. This method rapidly converges on roots of the given function. For this question, iterate 10 times, printing the computed approximation with each pass.

For this exercise, use $f(x) = 3x^6 - 5x^3 + 13x - 17$ and $f'(x) = 18x^5 - 15x^2 + 13$. You must define a function, `newtonMethod()`, for this problem, as shown below:

Sample Output

```
>>> newtonMethod()
Enter initial guess for root -2
-1.69662921348
-1.49566634169
-1.40634825341
-1.39048574903
-1.39003925848
-1.39003891405
-1.39003891405
-1.39003891405
-1.39003891405
-1.39003891405
```

The above is the sample output when starting with `-2` as the initial guess.

Ungraded: Feel free to experiment with different initial guess, and indeed, different functions f() and derivatives f'().

Q2	Plot Function Values

Skills
PM-7
CS-9
CS-5
DT-6

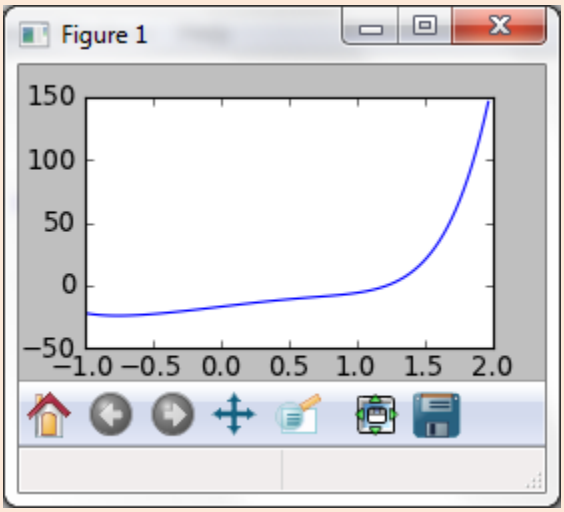
Lecture Dependency
Jan-17

Scientists often plot real-valued functions. Write a Python function `plotFunction()` that allows the user to enter in an `xleft` and an `xright` value.

Use $f(x) = 3x^6 - 5x^3 + 13x - 17$.

Construct two lists: `xValues` contains 100 evenly spaced floating point values between `xleft` and `xright`; `yValues` evaluates f(x) at each of these 100 `x` values.

With this information, you can then plot the f(x) function as shown here.



Sample Output

```
>>> plotFunction()
Enter left X value -1
Enter right X value 2
```

Q3	If statement

Skills
CS-1

Lecture Dependency
Jan-18

Write a Python function `guessingGame()`. In this function, the user is trying to "guess" a secret integer value known to the program. The user has **five** tries to guess the number. If the user guesses too high, the program responds "Too High". If the user guesses too low, the program responds "Too Low". If they guess correctly, the program says "You Win".

If the user doesn't guess the answer in 5 tries, then the program says "You Lose! My number was N"

Sample Output

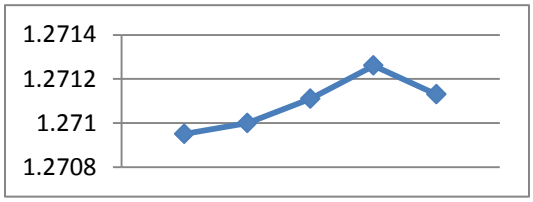
```
>>> guessingGame()
Can you guess my number? 3
Your guess is too low
Can you guess my number? 9
Your guess is too high
Can you guess my number? 5
You Win
```

Initial Version: 1/17/2013

The context for the following questions on this homework is the Foreign Exchange Market (**forex** or **fx**) which is the global decentralized trading of international currencies. This market determines the relative values of different currencies using symbols, such as EURUSD or USDJPY. I have simplified many elements of FX for the purpose of this assignment.

An fx trader reviews the value of a currency symbol at regular one minute intervals. At the end of each minute, there is a posted **close** value that determines the currency conversion. For example, in a five minute period, one might see the following **close** values for EURUSD, which determines the conversion rate between Euros (€) and US Dollars ($):

- 00:01 EURUSD is 1.27095
- 00:02 EURUSD is 1.271
- 00:03 EURUSD is 1.27111
- 00:04 EURUSD is 1.27126
- 00:05 EURUSD is 1.27113



As you can see, EURUSD was initially rising (which means that the EUR currency was gaining strength relative to the US Dollar), but in the fifth minute it fell back. When the value of EURUSD is 1.27095, it means that one Euro can be exchanged for 1.27095 US Dollars.

Fx traders can only make money because of leverage; for the purpose of this assignment, however, I am simplifying the discussion. A **pip** is defined as 0.0001 and is a convenient unit to use when comparing the rise and fall of a currency symbol. For example, if EURUSD increases from 1.271 to 1.2711, you could say that EURUSD rose by 1.1 pips. Alternatively, if you were to say that EURUSD had fallen 3 pips, you know that the absolute value of EURUSD had fallen by 0.0003.

An *FX Trader* wants to open two kinds of positions on a currency symbol to make money.

- *Long* – When a trader opens a LONG position (also known as a Buy) with a given LOT SIZE on a given currency, he will only make money if the currency rises in value.
- *Short* – When a trader opens a SHORT position (also known as a Sell) with a given LOT SIZE on a given currency, he will only make money if the currency falls in value.

After a position has been opened, the trader can close it, which realizes a profit or a loss.

Assume the trader had opened a LONG position with lotSize = 1.0 after the first **close** (that is, when EURUSD=1.27095 and then closed that position after the 5th **close** (when EURUSD=1.27113). As you can

verify, the currency has risen 1.8 pips. In this case, the profit can be computed to be 1.8*1.0*100000 = 18 USD$. It's because of leverage that you need to multiply by 100,000 to compute the accurate profit.

If the trader had opened a SHORT position with lotSize = 1.0 after the first **close**, then after the 5th **close** he would have realized a loss of 1.8 pips, which would translate into a loss of 18 USD$.

Q4	Plot Currency

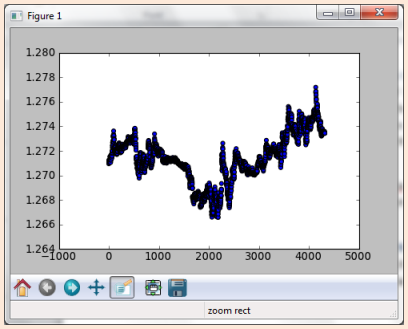
Skills
PM-7

Lecture Dependency
Jan-17

Lectures contain code examples that use the **pylab** module.

Write a function `plotCurrencySymbol()` that opens a window with a scatter plot of the data. *Hint: range() will be useful*

Sample output appears on the right. The HW2.py template file contains the `currencyValues` list.



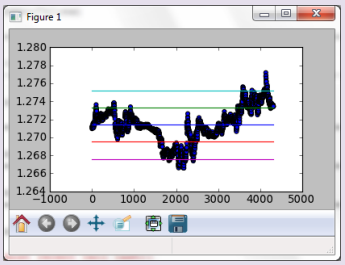
Q5	Demonstrate computational ability and plotting

Skills
PM-7
DT-6

Lecture Dependency
Jan-18

Given the `currencyValues` list, compute and plot some statistical information. Write a `plotCurrencyStats`() function for this question. You want to find the 'outliers' in this data set. To do so, you must determine (a) the average currency value of the symbol; and (b) the standard deviation of the currency value.

With this information, you can plot 5 horizontal lines. (1) the average; (2) average + stdev; (3) average + 2*stdev; (4) average – stdev; (5) average – 2*stdev. On the right, these appear on the plot.



You are then to count the number of value, v, in `currencyValues` that fall within these ranges. Specifically: (a) values within one stdev; (b) values within two stdev; (c) values within three stdev.

Hint: Be sure to print the statistics to the console window before you show the plot window, otherwise the output won't appear until you close the plot window.

Sample Output

```
>>> plotCurrencyStats ()
Within one stdev: 2906 or 67.2685185185 percent
Within two stdev: 4154 or 96.1574074074 percent
Within three stdev: 4319 or 99.9768518519 percent
```

Compare your answers to "Standard deviation and tolerance intervals" as found in any statistics reference.

Q6	Trading Strategies

Skills

Lecture Dependency

Jan-18

Write a function `tradingBackTest()` that allows the user to "test" a basic trading strategy that "Buys Low" and "Sells High". Specifically, your **fx** trader only wants to wait for the currency to fall below a certain value, at which time he opens a BUY position. Then he will wait for the currency to rise to a certain higher level, at which point he will SELL and take a profit. The trader will never have more than one position open at a time. The following is sample input for a given low and high threshold.

Sample Output

```
>>> tradingBackTest()
Enter in low threshold to buy: 1.270
Enter in high threshold to sell: 1.276
Open BUY position at 1.26978
Close BUY position at 1.27718 with profit = 740.0
Total Profit=740.0
```

Your program must iterate over every value in `currencyValues`, and determine whether to open a BUY position, or close an existing BUY position for profit. As you can see above, the BUY trade was executed when the value fell below 1.270 and was closed when the value became greater than 1.276.

Note: If you get to the end of the data and you have not yet closed a position, you must then do so at STOP, as shown below:

Sample Output When Reaching End Of Data

```
>>> tradingBackTest()
Enter in low threshold to buy: 1.270
Enter in high threshold to sell: 1.3
Open BUY position at 1.26978
Close BUY position at STOP 1.27344 with profit = 366.0
Total Profit=366.0
```

Naturally, if you are too conservative, you may never open any trades:

Sample Output With No Trades

```
>>> tradingBackTest()
Enter in low threshold to buy: 1.1
Enter in high threshold to sell: 2.0
Total Profit=0
```

You must answer the following two questions for credit:

1. If you were to open (and close) a single BUY position, what is the maximum profit that you can make?
2. Find a (low, high) threshold pairing that would allow you to make multiple trades during this time period.

Hint: Use the "zoom" capability of the plot window in question 4 to locate two currency values that will allow you to open (and close) two separate positions.

Initial Version: 1/17/2013

How To Get Started On This Assignment

A template HW2.py file is provided to you with EURUSD financial data sampled at one minute intervals over a three-day period. As you can see, this **list** contains 1440 * 3 = 4320 individual values.

Much of the work for this assignment will be spent trying to understand the **fx** domain and writing the appropriate Python code. In many ways, that is as it should be! The job of a programmer is more than learning a particular syntax. You need to know how to produce code relevant for a specific problem. Sometimes the code you write is only 5 lines of code (but it will be just the right five lines of code).

Submit your HW2.py file using the web-based turnin system. As we have mentioned in class, only one of the team members needs to submit the assignment. But just make sure that something gets submitted!