CSE 231 **Fall 2016**

Computer Project #05

(Sample output fixed on October 13 with a missing parameter added to the `month_average` function)

Assignment Overview

This assignment focuses on the implementation of Python programs to read files and process data by using lists and functions.

It is worth 45 points (4.5% of course grade) and must be completed no later than 11:59 PM on Monday, October 17.

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj05.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **handin system** before the project deadline.

Assignment Background

U.S. Geological Survey (USGS) provides scientific information to understand Earth, manage resources and minimize loss from natural disasters. You can download interesting data from the website https://www.usgs.gov/

We have downloaded the flow rate data for the Red Cedar River in East Lansing starting from 1932 . Your task is to design and implement a Python program that draws two plots from the data (and displays a table of data for each plot).

Here is the first line of the file. The numbers that we are interested are the *last three numbers* which are year, month and flow rate in cubic feet per second (CFS). Notice that the year and month are ints and the flow rate is float:

```
USGS 04112500  00060     70495     1932 1    215.5
```

Assignment Specifications

1. The program must provide following functions to extract some statistics.
   a) `open_file()` prompts the user to enter a file name. The program will try to open the data file. Appropriate error message should be shown if the data file cannot be opened. This function will loop until it receives proper input and successfully opens the file. It returns a file pointer.

b) `read_file()` calls the `open_file()` function and uses the returned file pointer to read the data file. This function returns a list of your choosing containing data you need for other parts of this project.

c) `draw_plot( x, y, plt_title, x_label, y_label)` provided by us takes two equal-length lists of numbers and plots them. You need to pass the label strings and plot title to the function. Include the range of years in the title (Hint: use string concatenation and check sample output).

d) `annual_average(L)` takes a list as an argument and returns a list of tuples in the form (year, average_flow).

e) `month_average(L,M)` takes a list L and month M as arguments and returns a list of tuples in the form (year, month_flow). (Note: this originally left out the month M parameter—if you figured out how to solve the problem using the previous specification, you will get full credit.)

f) You may use extra functions, if you wish.

2. The program should read the file only once.
3. The program should plot the average flow for each year. That is, find the average flow for each year and then generate a plot with years on the x-axis and average flow for that year on the y-axis. Then generate a table that has the year and average flow on each line. Put a title on the table and label each column.
4. Next, the program should prompt the user for a number in the range 1-12 and should plot the flow rates for only that month. For example if user enters 1, the program should extract the flow rate of Jan 1932, Jan 1933, … , Jan 2015 and plot them. (Note: do not loop and ask for more.)
5. The program should re-prompt the user if an invalid input (either wrong type or wrong value), was entered for the month. If something other than an integer is input, your error message should include the phrase "not an integer." If an integer is input but it is not in the proper range, your error message should include the phrase "integer out of range." Note: use exceptions.
6. The month name should be displayed in the title of the second plot instead of the month number. For example, in the sample below you see "May" in the title. (Hint: use the month number as an index into a list of strings.) As with the previous plot, display a table of values that you plotted.

Assignment Notes

1. Items 1-9 of the Coding Standard will be enforced for this project.

2. Note that data are separated using spaces. You can use the list method .split() to split the line into a list of data.

3. It is much easier to convert the input data to int and float when the program reads the file and creates the list.

4. To create a list `L` of data begin with an empty list (e.g. `L = []` before the loop begins) and within the loop append to the list one item at a time, e.g. `L.append(item)`. The data item you append may be a collection such as a tuple or another list.

Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step may be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.

- Construct the program one function at a time—testing before moving on.

- Use the **handin system** to turn in the first version of your solution.
  Cycle through the steps to incrementally develop your program:

  - Edit your program to add new capabilities.
  - Run the program and fix any errors.
  - Use the **handin system** to submit the current version of your solution.

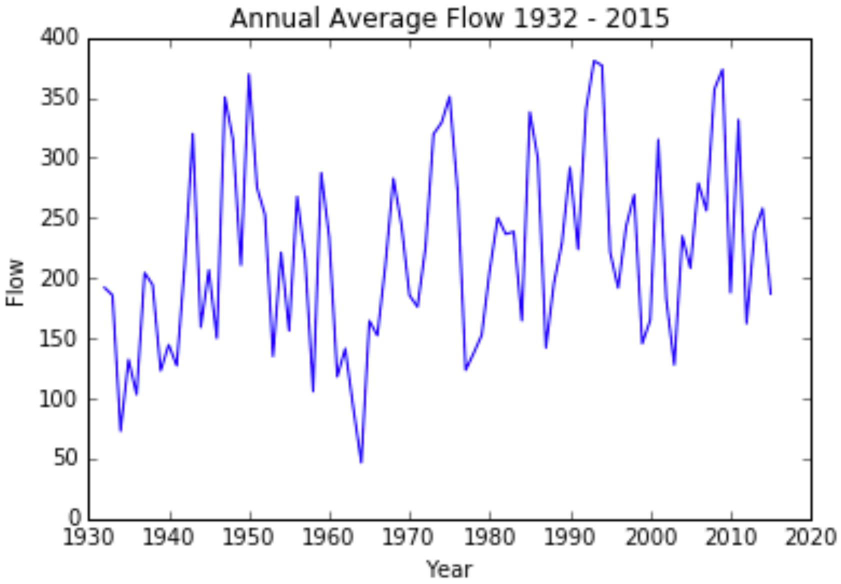
- Be sure to log out when you leave the room, if you're working in a public lab.

Sample Output

```
runfile('/Users/enbody/Documents/cse231/FS16/Projects/Project05/proj
05.py',
wdir='/Users/enbody/Documents/cse231/FS16/Projects/Project05')

Input a file name: BadFileName

Error: Input a file name: RedCedarRiver.txt
```

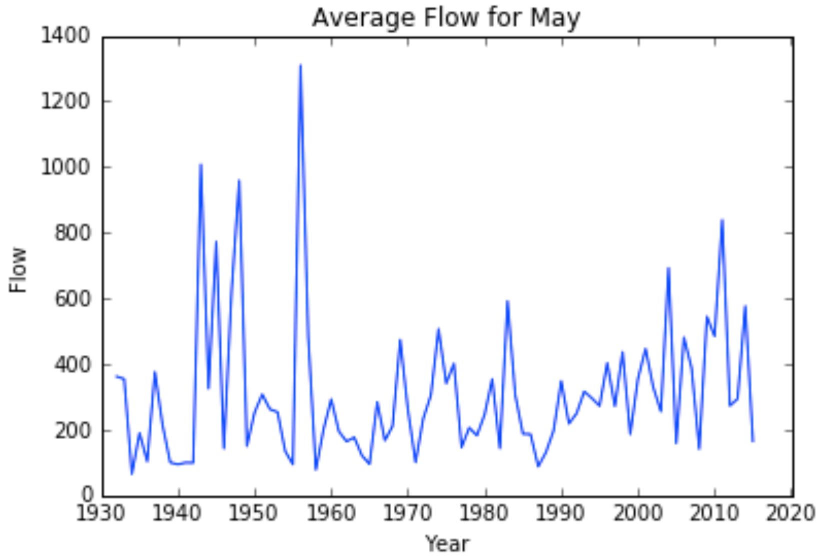


```
Annual Average Flow
Year            Flow
1932          192.10
1933          185.74
1934           72.88
1935          132.31
1936          103.25
```

1937	204.47
1938	194.36
1939	123.07
1940	144.65
1941	127.25
1942	211.36
1943	319.97
1944	159.20
1945	206.77
1946	150.00
1947	350.34
1948	315.48
1949	210.72
1950	369.48
1951	275.35
1952	252.57
1953	135.01
1954	221.32
1955	156.29
1956	267.83
1957	216.15
1958	105.79
1959	287.53
1960	234.02
1961	118.08
1962	141.41
1963	92.31
1964	46.62
1965	164.56
1966	152.28
1967	210.72
1968	282.76
1969	244.50
1970	185.69
1971	176.13
1972	225.50
1973	319.70
1974	329.13
1975	350.86
1976	272.38
1977	123.58
1978	137.61
1979	152.61
1980	206.37
1981	250.23
1982	236.56
1983	238.59
1984	164.65
1985	337.97

```
1986        299.21
1987        142.12
1988        195.68
1989        229.19
1990        292.00
1991        223.88
1992        340.07
1993        380.49
1994        376.62
1995        221.57
1996        191.86
1997        243.13
1998        269.27
1999        145.65
2000        164.97
2001        315.15
2002        182.42
2003        127.92
2004        234.91
2005        208.33
2006        278.90
2007        256.09
2008        356.99
2009        373.35
2010        187.87
2011        331.62
2012        162.27
2013        238.76
2014        257.95
2015        186.90

Enter a month (1-12): xxxx
Error. Not an integer
Enter a month (1-12): 13
Error. Integer out of range.
Enter a month (1-12): -5
Error. Integer out of range.
Enter a month (1-12): 5
```



```
Average Flow for May
Year        Flow
1932      362.40
1933      354.50
1934       65.40
1935      190.00
1936      103.60
1937      376.30
1938      214.60
1939      100.40
1940       94.50
1941       99.90
1942       99.30
1943     1008.00
1944      325.30
1945      772.40
1946      143.00
1947      619.50
1948      959.30
1949      150.20
1950      252.50
1951      307.90
1952      262.70
1953      253.70
1954      134.70
1955       94.60
1956     1310.00
1957      482.80
1958       78.30
1959      201.70
1960      293.00
```

1961	195.00
1962	164.50
1963	176.80
1964	121.60
1965	95.50
1966	284.40
1967	167.20
1968	211.40
1969	474.10
1970	261.20
1971	101.10
1972	234.10
1973	305.80
1974	507.10
1975	341.10
1976	401.70
1977	146.60
1978	206.30
1979	182.30
1980	246.20
1981	354.40
1982	143.60
1983	591.90
1984	306.20
1985	187.60
1986	186.50
1987	87.90
1988	131.20
1989	197.30
1990	348.40
1991	218.50
1992	249.80
1993	316.60
1994	296.00
1995	272.60
1996	403.40
1997	271.80
1998	436.80
1999	186.40
2000	354.90
2001	446.70
2002	329.50
2003	255.80
2004	692.80
2005	158.90
2006	481.40
2007	387.70
2008	141.30
2009	544.70

```
2010      485.00
2011      839.20
2012      273.10
2013      294.20
2014      576.70
2015      165.10
```

Educational Research

When you have completed the project insert the 5-line comment specified below.

For each of the following statements, please respond with how much they apply to your experience completing the programming project, on the following scale:

1 = Strongly disagree / Not true of me at all
2
3
4 = Neither agree nor disagree / Somewhat true of me
5
6
7 = Strongly agree / Extremely true of me

****Please note that your responses to these questions will not affect your project grade, so please answer as honestly as possible.****

Q1: Upon completing the project, I felt proud/accomplished

Q2: While working on the project, I often felt frustrated/annoyed

Q3: While working on the project, I felt inadequate/stupid

Q4: Considering the difficulty of this course, the teacher, and my skills, I think I will do well in this course.

Please insert your answers into the <u>bottom</u> of your project program as a <u>comment</u>, formatted exactly as follows (so we can write a program to extract them).

```
# Questions
# Q1: 5
# Q2: 3
# Q3: 4
# Q4: 6
```