

Exercise 1 – Stream Discharge Analysis

From the class website, obtain the file `white_clay_discharge_2015.txt` and use a text editor (Emacs) to look at it. The first few lines in the file look like this:

```
USGS      01479000      1960-01-01      98      A
USGS      01479000      1960-01-02      91      A
USGS      01479000      1960-01-03     446      A
USGS      01479000      1960-01-04     169      A
USGS      01479000      1960-01-05     124      A
```

indicating that these data are from US Geological Survey stream gauge 01479000, which is on White Clay Creek where it passes through Delaware Park (the horse racing and casino area downstream of Newark). The third column indicates a date, with data beginning on January 1, 1960, the fourth column is a discharge in cubic feet per second (cfs), and the fifth is a status code we will not do anything with.

A stream gage at or near this site has been operated intermittently since 1931. Uninterrupted monitoring began in late 1959, and the file described here begins with January 1, 1960, and ends with December 31, 2016. Detailed information about USGS stream gage 01479000 can be found at

http://waterdata.usgs.gov/nwis/dv?referred_module=sw&site_no=01479000

You can look at another local gage by changing the site number to 1478650, which is the gage north of town just off Paper Mill Road that is actually used for evaluating Newark's water supply – it has a much shorter period of record. The USGS maintains similar gages all over the country, with archived and real-time data available from their web site.

When a water supply comes from a river, a minimum flow is required at all times to preserve the fish and the ecological balance in general. The city is never allowed to pump so much water that the river discharge falls below some regulatory minimum, and if the discharge falls below this minimum naturally, the city is not allowed to pump water at all. In equation form, this could look like

$$Q_{\text{available}} = \max (Q - Q_{\text{fish}}, 0)$$

where $Q_{\text{available}}$ is the amount regulations would allow to be pumped from the river, Q is the actual river discharge (the numbers in the data file), and Q_{fish} is the regulatory minimum discharge.

A second consideration is that pumping capacity to remove water from the river is limited. High capacity pumps are expensive, and a city maintains just enough pumping capacity to refill all of its water towers and recharge the reservoir at night. Call this Q_{pump} , and the amount of potential water available in a day is

$$Q_{\text{potential}} = \min (Q_{\text{available}}, Q_{\text{pump}})$$

(In Python, it would be possible, and preferable, to combine both equations into one and never actually calculate $Q_{\text{available}}$.)

Phase I. Write a Python program that reads the discharge data file and calculates the *average daily potential water supply* over the entire period of record. For each day, you will need to find the discharge from the data line, calculate the potential water supply for the day, sum this discharge to a variable that accumulates the total potential, and also maintain a separate day counter that counts up how many days are in the file. After completing the loop for each day, you will need to calculate the average daily supply potential from the accumulated potential and the number of days. Print to the screen your average daily amount and how many days were involved, with enough additional character strings to identify what the numbers mean.

Using $Q_{\text{fish}} = 100$ cfs and $Q_{\text{pump}} = 5$ cfs, your answers for Phase I should be 1.810 cfs averaged over 20820 days. (You do not need to turn in Phase I when completed– these are just steps along the way to Phase II.)

Phase II. Add code to your program that prints out the average daily potential water supply individually for *each complete year* of the record. A couple of Python hints:

- You will need to extract the month, day, and, year from the date, and that requires use of *split* in a more sophisticated manner than previously. Remember that the space character is the *default* splitting character, but you can put any character in as the splitting character, including “-”. Also, once a line of the file has been split into substrings using one splitting character, those substrings can be further split using another splitting character.
- As with the discharge in Phase I, you need to convert date components from strings to numbers. If variables are months, days, and years, probably *int* is a better choice than *float*.
- There are at least two ways to determine you are at the end of the year. The easiest is probably to test if the month number is 12 and the day number is 31.
- When you have accumulated the total discharge and number of days for a single year (and you do have to either count number of days or find out how to determine if this is a leap year), the set of steps is roughly:
 - calculate the average daily potential water supply by dividing accumulated potential water supply by the day count,
 - print the year and the average daily potential water supply to the screen (without extra text strings – it can look like a table – and
 - reset the variables into which you have been accumulating daily water supply and counting days back to zero.

With those resets, the average daily water supply potential should be calculated independently for each year. The average for 2016 was 1.33 cfs.

Hand-in for this exercise is just the Python program that solves Phase II. The program file should be named *lastname_ex1.py*, where *lastname* is your actual last name. The first line or two in the file should be comments (lines starting with #) with your full name and the exercise number. (Please put the comments in your file right away, or at least make sure you test the program after adding the comment lines.) Your program file should be sent to the instructor as an email attachment. 10 points.