Activity

In a script called **A17-cones.py**

Array operations are useful for repetitive calculations.

For example, suppose you collect the volume (V) of various cones by recording their diameter (D) and height (H). Let's imagine you collect information on 100 cones, and let's imagine that lists d (containing the dimeters) and h (containing the heights) each contain 100 elements, and you want to calculate 100 volumes.

Here, we use less data.

```
d = [0.2,1.0,1.5,3.0,-1.0,4.2,3.14]
h = [0.0400,1.0000,2.2500,9.0000,1.0000,17.6400,9.8596]
```

1. Use NumPy vectorization to calculate the volume V of the cones and print V to screen.
   Keep in mind that element by element multiplication is different than matrix multiplication, and so you do not need to transpose.

   $V = 1/12*pi*(D^2)*H$

2. Calculate the mean and standard deviation of V, and print results to screen with 2 decimal precision

```
The average is 18.57 and std is 27.57
```

Activity: Pegasus

Download data file drop.dat from Canvas folder DATA.

The data contained in drop.dat was taken during the launch of a Pegasus missile. The data is the longitudinal (sideways) acceleration just before ignition up to after launch. The first field of the data corresponds to the time (in seconds) and the second field is acceleration (in G's)

Make a script called **A17-Pegasus.py** and in it do the following. Import NumPy.

1. Use loadtext() function to read in the data file drop.dat

2. How many recorded lines there are ?
   Use an array attribute to find it out, and print the result to screen

```
There are 4380 lines
```

3. Store the first field (first column) in variable time and second field (second column) in variable acceleration.

4. Use Boolean indexing to make a new vector called timep, which stores only positive values of time, and make a variable called accelerationp, which stores corresponding accelerations for positive values of time.

Activity: Pegasus

5. Calculate the maximum acceleration (amax) , and the time when it occurred (tmax).
Use max to find the maximum. Use argmax to get the indices of positive times and use them to extract corresponding acceleration values. Use accelerationp and timep.

6. Print this formatted text, which reports the max acceleration and the time it occurred.
Use variables amax and tmax, add text

```
Maximum acceleration was 1.031 G and it occurred at time 0.125 s
```

7. Sort acceleration and store the first 5 largest acceleration values in variable acc5, and corresponding times in variable t5. Use the sort and argsort functions. Do not hardcode the last 5 elements.

8. Use acc5 and t5, and display to screen. Use a for loop. You can zip ndarrays.

```
0.128529 1.0049
0.122128 1.0211
0.126929 1.0223
0.123728 1.0305
0.125329 1.0309
```

Activity: Pegasus

9. Use acc5 and t5 and write their values in a file called file5.csv. The field separator should be a comma, and the the numbers should be float with 2 decimal precision.

Use np.savetxt. Hint: You should concatenate/stack and transpose.

```
0.13,1.00
0.12,1.02
0.13,1.02
0.12,1.03
0.13,1.03
```

Submit to A17:

- **A17-cones.py**
- **A17-Pegasus.py**