This almost exactly follows SWC lesson 1, just replacing inflammation data with auto spectra.

I'm leaving out all the words/explanations in this notebook and just documenting the actual code typed into the notebook, so this notebook can only be understood in concert with SWC lesson 1.

In [ ]:
```
import numpy
```

In [ ]:
```
numpy.loadtxt(fname='spectra-01.csv', delimiter=',')
```

In [ ]:
```
weight_kg = 55
```

In [ ]:
```
print(weight_kg)
```

In [ ]:
```
print('weight in pounds:', 2.2 * weight_kg)
```

In [ ]:
```
weight_kg = 57.5
print('weight in kilograms is now:', weight_kg)
```

In [ ]:
```
weight_lb = 2.2 * weight_kg
print('weight in kilograms:', weight_kg, 'and in pounds:', weight_lb)
```

In [ ]:
```
weight_kg = 100.0
print('weight in kilograms is now:', weight_kg, 'and weight in pounds is sti
```

In [ ]:
```
%whos
```

In [ ]:
```
data = numpy.loadtxt(fname='spectra-01.csv', delimiter=',')
```

In [ ]:
```
print(data)
```

In [ ]:
```
print(type(data))
```

In [ ]:
```
print(data.dtype)
```

In [ ]:
```
print(data.shape)
```

In [ ]:
```
print('first value in data:', data[0, 0])
```

In [ ]:
```
print('middle value in data:', data[63, 192])
```

In [ ]:
```
print(data[0:4, 0:3])
```

In [ ]:
```
print(data[4:8, 0:3])
```

In [ ]:

```
small = data[:4, 381:]
print('small is:')
print(small)
```

In [ ]:

```
doubledata = data * 2.0
```

In [ ]:

```
print('original:')
print(data[:4, 381:])
print('doubledata:')
print(doubledata[:4, 381:])
```

In [ ]:

```
tripledata = doubledata + data
```

In [ ]:

```
print('tripledata:')
print(tripledata[:4, 381:])
```

In [ ]:

```
print(numpy.mean(data))
```

In [ ]:

```
import time
print(time.ctime())
```

In [ ]:

```
maxval, minval, stdval = numpy.max(data), numpy.min(data), numpy.std(data)

print('maximum power:', maxval)
print('minimum power:', minval)
print('standard deviation:', stdval)
```

In [ ]:

```
antenna_0 = data[0, :] # 0 on the first axis, everything on the second
print('maximum inflammation for antenna 0:', antenna_0.max())
```

In [ ]:

```
print('maximum inflammation for antenna 2:', numpy.max(data[2, :]))
```

In [ ]:

```
print(numpy.mean(data, axis=0))
```

In [ ]:

```
print(numpy.mean(data, axis=0).shape)
```

In [ ]:

```
print(numpy.mean(data, axis=1))
```

In [ ]:

```
print(numpy.mean(data, axis=1)).shape
```

In [ ]:

```
import matplotlib.pyplot
image = matplotlib.pyplot.imshow(data)
matplotlib.pyplot.show()
```

In [ ]:

```
%matplotlib inline
```

In [ ]:

```
image = matplotlib.pyplot.imshow(data)
matplotlib.pyplot.show()
```

In [ ]:

```
ave_spectrum = numpy.mean(data, axis=0)
ave_plot = matplotlib.pyplot.plot(ave_spectrum)
matplotlib.pyplot.show()
```

In [ ]:

```
max_plot = matplotlib.pyplot.plot(numpy.max(data, axis=0))
matplotlib.pyplot.show()
```

In [ ]:

```
min_plot = matplotlib.pyplot.plot(numpy.min(data, axis=0))
matplotlib.pyplot.show()
```

In [ ]:

```
import numpy
import matplotlib.pyplot

data = numpy.loadtxt(fname='spectra-01.csv', delimiter=',')

fig = matplotlib.pyplot.figure(figsize=(10.0, 3.0))

axes1 = fig.add_subplot(1, 3, 1)
axes2 = fig.add_subplot(1, 3, 2)
axes3 = fig.add_subplot(1, 3, 3)

axes1.set_ylabel('average')
axes1.plot(numpy.mean(data, axis=0))

axes2.set_ylabel('max')
axes2.plot(numpy.max(data, axis=0))

axes3.set_ylabel('min')
axes3.plot(numpy.min(data, axis=0))

fig.tight_layout()

matplotlib.pyplot.show()
```

End of lesson 1

Lessons 2 & 3 exactly follows SWC, no inflammation data is invoked.

Begin Lesson 4. Unlikely to actually get to this point, but better to be prepared.

In [ ]:

```
import glob
```

In [ ]:

```
print(glob.glob('spectra*.csv'))
```

```
In [ ]: import numpy
        import matplotlib.pyplot

        filenames = sorted(glob.glob('data/inflammation*.csv'))
        filenames = filenames[0:3]
        for f in filenames:
            print(f)

            data = numpy.loadtxt(fname=f, delimiter=',')

            fig = matplotlib.pyplot.figure(figsize=(10.0, 3.0))

            axes1 = fig.add_subplot(1, 3, 1)
            axes2 = fig.add_subplot(1, 3, 2)
            axes3 = fig.add_subplot(1, 3, 3)

            axes1.set_ylabel('average')
            axes1.plot(numpy.mean(data, axis=0))

            axes2.set_ylabel('max')
            axes2.plot(numpy.max(data, axis=0))

            axes3.set_ylabel('min')
            axes3.plot(numpy.min(data, axis=0))

            fig.tight_layout()
            matplotlib.pyplot.show()
```