Squiggles are Data, Squiggles are Vectors

Recall the squiggles example

In [9]:

```
%matplotlib inline

import numpy as np
import matplotlib.pyplot as pt
```

Here are two 'squiggles' represented as a bunch of numbers:

In [10]:

```
squiggle_1 = "141.03 291.04 141.28 291.50 141.92 291.50 142.67 291.04 143.94 290
.13 145.58 288.32 147.22 285.25 149.62 281.27 152.40 276.15 155.31 270.24 158.21
264.10 161.37 258.19 164.02 252.96 166.67 248.41 168.69 244.89 170.72 241.82 172
.48 239.55 174.25 237.84 175.52 236.59 176.27 236.13 176.65 236.48 176.65 237.50
177.03 240.00 177.41 244.21 177.79 250.12 178.04 257.39 178.04 265.12 178.42 273
.08 178.67 280.93 179.05 288.43 179.56 295.71 179.94 302.64 180.69 309.01 181.71
314.35 182.72 318.10 183.85 320.03 184.86 321.06 186.00 320.72 187.52 319.35 189
.28 316.40 191.31 311.39 193.33 304.35 195.60 295.93 198.00 286.84 200.65 277.86
203.56 269.67 206.21 262.62 208.61 257.17 210.51 253.07 212.02 250.23 213.03 248
.64 213.79 247.84 214.17 247.73 214.17 248.53 214.17 250.35 214.04 253.53 213.92
258.19 213.66 263.99 213.54 270.13 213.41 276.38 213.41 282.18 213.92 287.41 214
.80 291.84 216.32 295.48 218.21 298.09 220.11 299.80 222.13 300.59 224.27 300.71
226.55 300.03 228.95 298.43 231.22 295.71 233.24 292.07 235.26 287.07 237.16 280
.81 239.31 273.76 241.83 265.81 244.86 257.39 248.15 249.21 251.68 241.14 255.09
233.63 258.00 227.27 260.40 222.49 261.92 219.54 262.80 218.63 263.18 218.63 263
.18 219.88 263.05 222.49 262.55 226.93 261.92 233.75 261.03 242.16 260.40 251.82
260.27 262.40 260.53 273.08 260.91 284.11 261.79 294.91 262.80 304.35 264.44 312
.30 266.72 317.99 269.12 321.63 271.64 323.56 274.17 324.69 276.95 324.47 279.98
322.42 283.26 318.56 286.42 311.96 289.83 303.21 293.24 292.98 296.78 282.18 300
.32 272.17 303.60 263.42 306.38 256.60 308.65 251.82 310.04 248.75 310.93 247.50
311.18 247.96 311.05 248.98 310.93 250.91 310.80 253.98 310.55 258.76 310.42 265
.12 310.17 272.74 310.55 280.81 311.31 289.00 312.57 296.50 314.46 302.64 316.86
307.41 319.77 310.37 322.80 311.73 325.96 312.08 328.86 311.17 331.64 309.01 334
.29 304.91 336.82 299.00 339.47 291.73 342.38 283.43 345.54 275.13 348.95 267.51
352.36 260.69 355.14 255.57 357.03 252.16 358.17 250.46 358.55 250.23 358.55 250
.69 358.80 251.94 359.43 254.32 360.06 258.19 360.69 264.33 361.20 271.95 361.71
280.59 362.59 289.23 363.98 296.84 366.00 303.44 368.53 308.44 371.56 311.73 375
.22 313.33 379.26 312.64 383.81 310.26 388.74 306.62 394.29 301.50 "
squiggle_2 = "243.60 219.20 243.60 218.06 242.84 216.92 242.21 215.67 241.07 214
.19 239.56 212.60 237.66 211.01 235.26 209.65 232.23 208.62 228.82 207.60 225.03
206.92 220.86 206.46 216.57 206.12 212.27 206.80 207.60 208.05 202.80 210.33 198
.13 213.74 193.20 217.72 188.78 222.15 184.74 226.59 181.45 231.13 178.93 235.91
177.28 241.02 176.27 246.59 175.89 252.62 176.27 258.99 177.28 265.81 178.93 272
.85 181.33 279.79 184.74 286.38 188.91 292.64 193.83 298.32 199.52 303.78 205.71
308.67 212.40 312.99 219.60 316.62 226.80 319.69 234.00 322.19 241.20 323.90 248
.27 324.81 255.60 325.26 262.93 324.81 270.25 323.90 277.96 322.53 285.54 320.60
293.12 318.56 300.32 316.17 306.63 313.33 312.44 310.14 317.75 306.39 322.80 301
.96 327.60 296.73 332.15 290.82 336.44 284.11 340.48 276.83 344.15 269.22 347.05
261.60 349.07 253.98 350.34 246.59 350.59 239.20 349.83 231.59 348.19 223.63 345
.41 215.67 341.75 208.17 337.07 201.35 331.77 195.66 326.21 191.12 320.65 187.48
315.09 184.75 309.16 182.93 302.84 182.02 296.02 181.91 288.82 182.36 281.87 183
.27 274.93 184.52 268.11 185.89 261.54 187.25 255.35 188.61 249.66 190.21 244.86
191.68 240.82 193.28 237.03 194.87 233.87 196.57 230.97 198.28 228.44 199.98 226
.42 201.69 224.65 203.51 223.52 205.55 222.51 207.94 221.75 210.78 221.24 213.97
220.74 217.72 220.48 221.24 220.23 224.43 219.85 227.04 219.22 229.20 218.72 230
.91 218.21 232.04 217.83 232.72 217.71 233.07 217.33 233.07 216.95 232.50 216.06
231.13 215.43 228.86 215.05 225.56 "
```

"Parsing" those turns them from strings into arrays of numbers:

In [11]:

```
def parse_squiggle(s):
    numbers = [float(num) for num in s.split()] #Convert strings to floats. s.sp
lit() ignores whitespace.
    a = np.array(numbers) #convert to numpy array
    return a.reshape(-1, 2).T #reshape into array with 2 rows (2 columns, then t
ranspose)
```

In [12]:

```
s1 = parse_squiggle(squiggle_1)
s2 = parse_squiggle(squiggle_2)
```

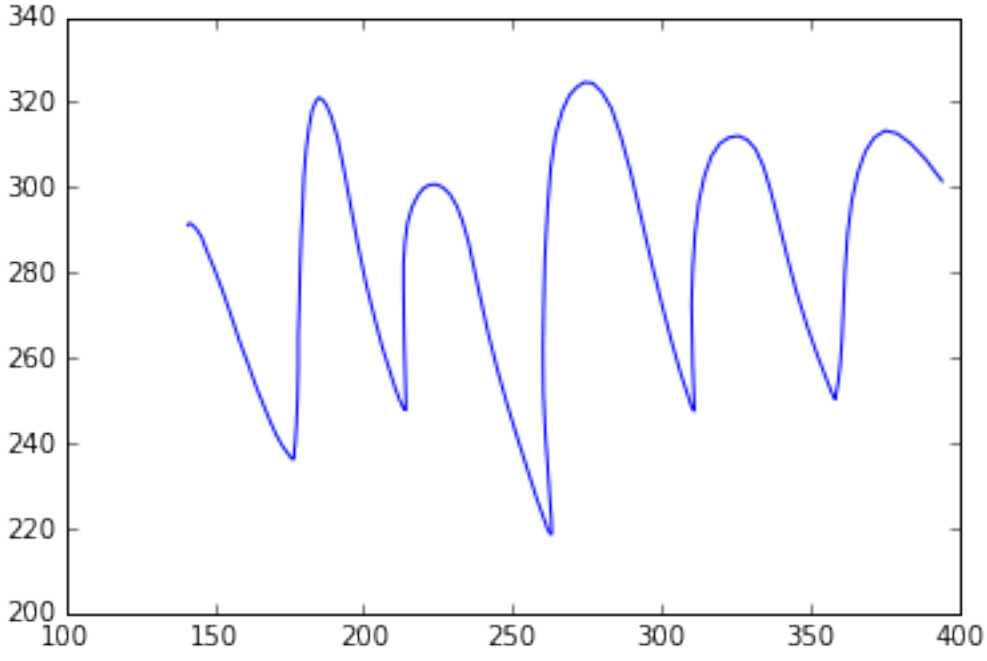
Let's plot both squiggles.

In [13]:

```
pt.plot(s1[0], s1[1]) #plot first row on x-axis, second on row on y-axis
```

Out[13]:

```
[<matplotlib.lines.Line2D at 0x10d566470>]
```

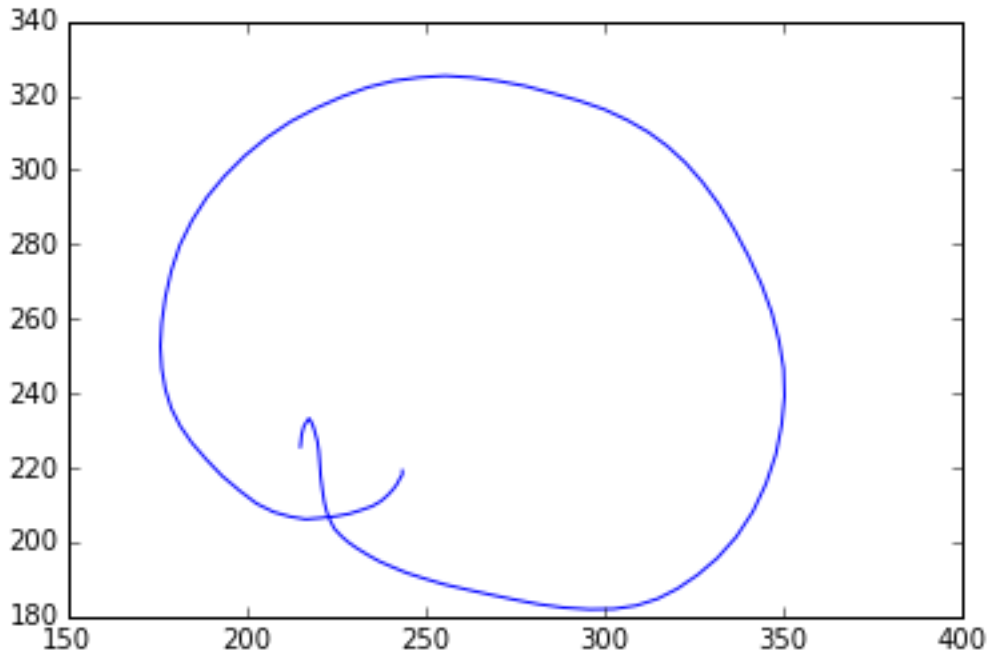


In [14]:

```
pt.plot(s2[0], s2[1])
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x10d6243c8>]
```



Will this work?

In [15]:

```
s1 + s2
```

```
---------------------------------------------------------------------
-------
ValueError                                Traceback (most recent cal
l last)
<ipython-input-15-1659087814e1> in <module>()
----> 1 s1 + s2

ValueError: operands could not be broadcast together with shapes (2,
159) (2,105)
```

So we'll need to do something.

In [16]:

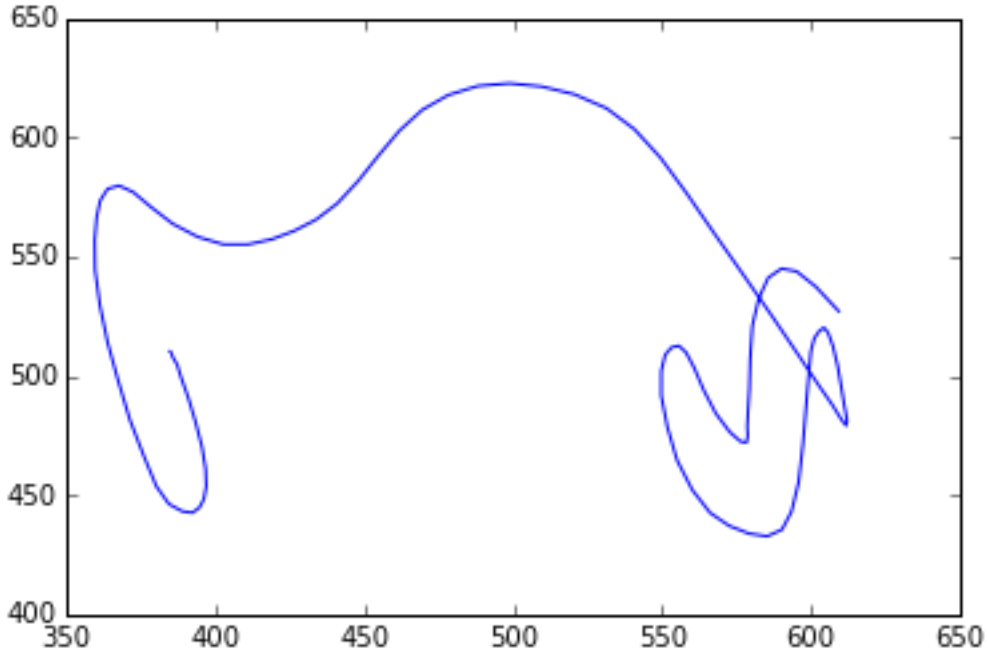
```
from scipy.interpolate import interp1d #this interpolates between data points (r
eturns function that "fills" in holes)

_, ns1 = s1.shape #save number of columns of s1
_, ns2 = s2.shape

s1x_interp = interp1d(np.linspace(0, 1, ns1), s1[0]) #interpolate first row over
interval [0, 1]
s1y_interp = interp1d(np.linspace(0, 1, ns1), s1[1])

s1_new = np.array([
    s1x_interp(np.linspace(0, 1, ns2)), #evaluate interpolated function over sam
e interval with smaller spacing
    s1y_interp(np.linspace(0, 1, ns2))]) #the two calls extend the length of s1,
so we can add it to s2
```

In [32]:

```
s3 = s1_new + s2
pt.plot(s3[0], s3[1])
```

Out[32]:

```
[<matplotlib.lines.Line2D at 0x10ed8b6d8>]
```



In [29]:

```
print(s1[0][:10])
print(s1[1][:10])
```

```
[ 141.03  141.28  141.92  142.67  143.94  145.58  147.22  149.62  15
2.4
  155.31]
[ 291.04  291.5   291.5   291.04  290.13  288.32  285.25  281.27  27
6.15
  270.24]
```

In [ ]:

```

```