Goal:
This project aims to find the strong relationship between MRI features and the smartwatch optical signal (PPG or photoplethysmography) signals to predict MRI features using the smartwatch PPG data.

Why:
If we can predict some MRI-derived variability from smartwatch signals alone, it may be possible to get early warning signals about health conditions through smartwatch data**.**

Data Provided:
A total of 7 subjects' data has been given by the professor using funds. Below is the data are given for each subject

- **MRI:** Blood oxygen level-dependent MRI, flow (carotid), cardiac left ventricle and qflow(aorta).
- **Smart Watch:** Optical sensor (PPG) and accelerometer X, Y, Z

Process:
- Preprocessing
- Feature Extraction
- Correlations

  (1) Smartwatch vs MRI correlations
  (2) Smartwatch vs smartwatch correlations
  (3) MRI vs MRI correlations
  (4) Scatter pair plot of smartwatch vs MRI correlation
  (5) P-value with pair plot for correlations
  (6) Bonferroni correction

NOTE:
- We cannot calculate the correlation between the 5x4 (5x7 + 4x7) matrix. Even though if we make it 5x7 and 7x4 because the number of columns and rows should be equal to perform correlation. To solve it we added an extra column named the Third derivative to make the size equal.
- Subject 1 doesn't have a left ventricle image. To compensate for the value, we took the mean of the value of all other results.

Results and Detailed process by code:
Below is a quick view of the results. Step-by-step code and images had been included after this:

Smartwatch vs MRI correlations:

	dmn_mean	peak_carotid	left_ven_area	area_outerring	peak_aorta	bpm	hrv	first_deriv	second_deriv	Third_derv
dmn_mean	1.000000	-0.440068	-0.906138	0.147759	0.147759	0.285891	-0.188012	-0.413345	0.562216	0.265830
peak_carotid	-0.440068	1.000000	0.233296	-0.420569	-0.420569	-0.560409	0.357666	0.372687	-0.785594	-0.554263
left_ven_area	-0.906138	0.233296	1.000000	0.149593	0.149593	-0.058215	-0.139515	0.335791	-0.198993	0.157259
area_outerring	0.147759	-0.420569	0.149593	1.000000	1.000000	0.611130	-0.806657	0.233597	0.517820	0.556553
peak_aorta	[illegible]	[illegible]	0.149593	1.000000	1.000000	0.611130	-0.806657	0.233597	0.517820	0.556553
bpm	0.285891	-0.560409	-0.058215	0.611130	0.611130	1.000000	-0.674872	0.407983	0.460246	0.458000
hrv	-0.188012	0.357666	-0.139515	-0.806657	-0.806657	-0.674872	1.000000	-0.092260	-0.507665	-0.683849
first_deriv	-0.413345	0.372687	0.335791	0.233597	0.233597	0.407983	-0.092260	1.000000	-0.499053	-0.355316
second_deriv	0.562216	-0.785594	-0.198993	0.517820	0.517820	0.460246	-0.507665	-0.499053	1.000000	0.891600
Third_derv	0.265830	-0.554263	0.157259	0.556553	0.556553	0.458000	-0.683849	-0.355316	0.891600	1.000000

- We can see many useful findings here but below are 3 main findings
- Then there is a great inverse correlation between dmn_mean and the left ventricle area
- **Note:** These are inaccurate results because a lot of time should be taken to properly preprocess and select an accurate region of interest.
- Many features have inverse relationships so in-depth research will surely bring a fair amount of results

Smartwatch vs smartwatch correlations

	dmn_mean	peak_carotid	left_ven_area	area_outerring	peak_aorta
dmn_mean	1.000000	-0.674872	0.407983	0.460246	0.458000
peak_carotid	-0.674872	1.000000	-0.092260	-0.507665	-0.683849
left_ven_area	0.407983	-0.092260	1.000000	-0.499053	-0.355316
area_outerring	0.460246	-0.507665	-0.499053	1.000000	0.891600
peak_aorta	0.458000	-0.683849	-0.355316	0.891600	1.000000

- As you can see the results are quite accurate because there is a 0.89 correlation between peak value aorta and area altering which makes sense.

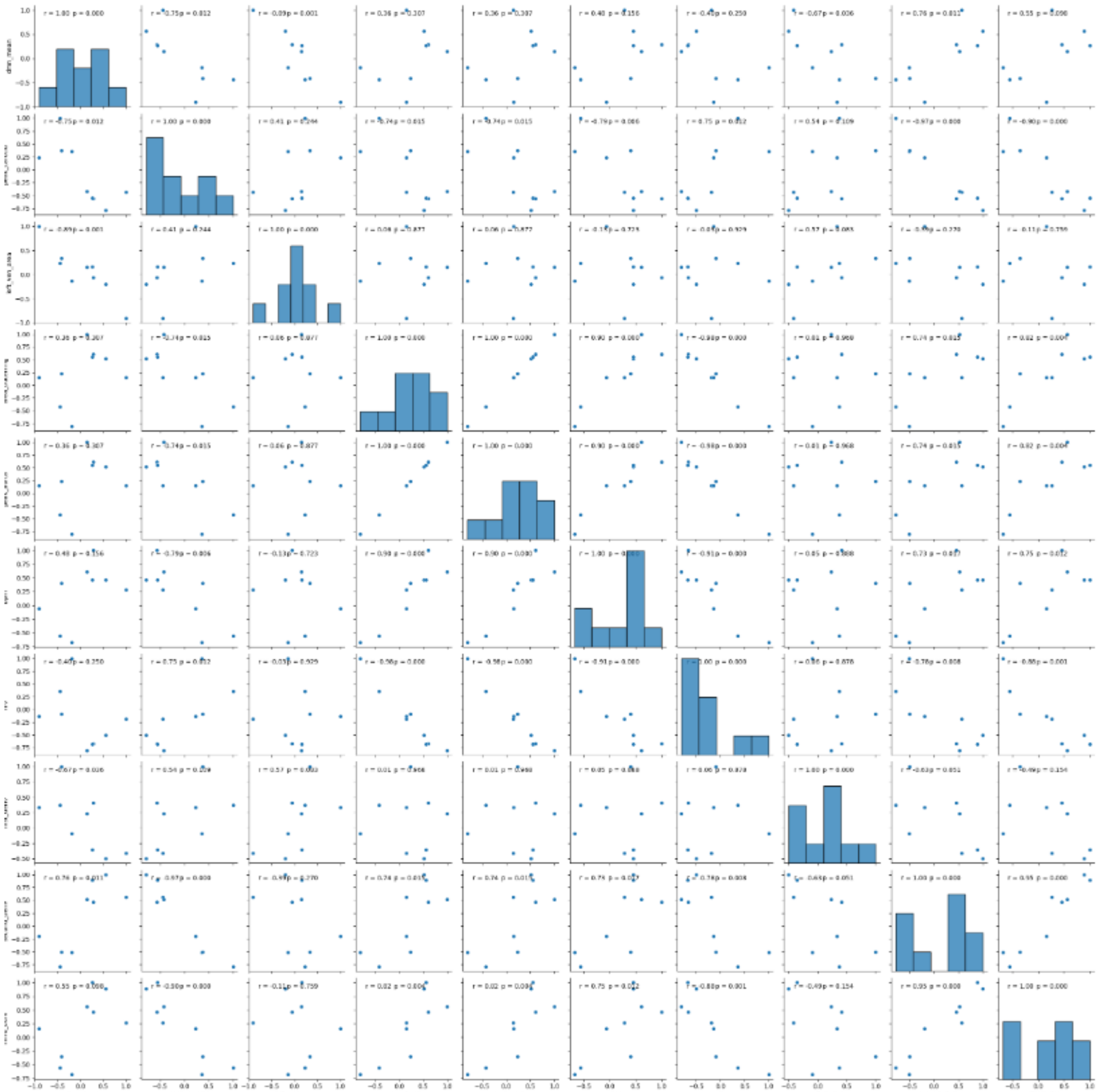
MRI vs MRI correlations

	bpm	hrv	first_deriv	second_deriv	Third_derv
bpm	1.000000	-0.440068	-0.906138	0.147759	0.147759
hrv	-0.440068	1.000000	0.233296	-0.420569	-0.420569
first_deriv	-0.906138	0.233296	1.000000	0.149593	0.149593
second_deriv	0.147759	-0.420569	0.149593	1.000000	1.000000
Third_derv	0.147759	-0.420569	0.149593	1.000000	1.000000

- The above shows the correlation between features of MRI

Scatter pairplot of smartwatch vs MRI correlation

- The image will not be clear, so we attached the below image with the submission. (Pairplot.png)



- The above scatter plots show the p-value for each scatter plot. After Bonferroni's correction, we concluded that the result is significant. As there are 61 p values to display, I included the p values for all the combinations in the **p-values.txt** file

BONUS:

- We did add a third derivative column as an extra column to compensate for the size of the matrix for correlation calculation

Preprocessing and Feature Extraction

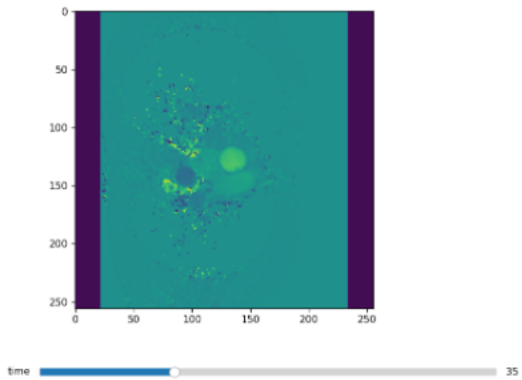
- Below results are on subject 4.
- Created high pass and normalized functions

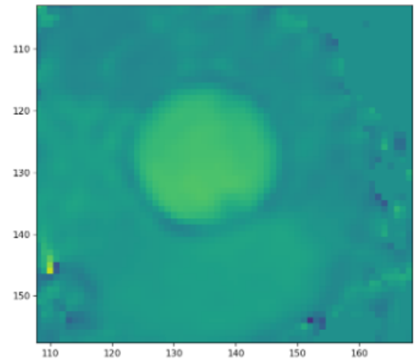
```python
import numpy as np
import nibabel as nib
import matplotlib.pyplot as plt
from matplotlib.widgets import Slider

from scipy import signal
from scipy.signal import resample
import pandas as pd
from datetime import datetime

def butter_highpass(cutoff, fs, order=5):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    b, a = signal.butter(order, normal_cutoff, btype='highpass', analog=False)
    return b, a


def butter_highpass_filter(data, cutoff, fs, order=5):
    b, a = butter_highpass(cutoff, fs, order=order)
    y = signal.filtfilt(b, a, data)
    return y

def normalize(arr, t_min=0, t_max=1):
    norm_arr = []
    diff = t_max - t_min
    diff_arr = max(arr) - min(arr)
    for i in arr:
        temp = (((i - min(arr))*diff)/diff_arr) + t_min
        norm_arr.append(temp)
    return norm_arr
```

- Import all the required subjects and looking the best slice for finding ROI for aorta

```python
%matplotlib notebook
import numpy as np

bold = nib.load('./subject_4/fmri.nii.gz').get_fdata()
# (80, 80, 42, 350)
lv = nib.load('./subject_4/left_ventricle_4d.nii.gz').get_fdata()
# (240, 7, 240, 28)
aorta  = nib.load('./subject_4/qflow_aorta.nii.gz').get_fdata()
# (256, 256, 120)
carotid = nib.load('./subject_4/qflow_carotid.nii.gz').get_fdata()
# (192, 192, 120)


mri = aorta[:,:,:]

fig, ax = plt.subplots()
plt.subplots_adjust(bottom=0.25)
ax.imshow(mri[:,:,0])
zmax = mri.shape[2]

ax_time = plt.axes([0.25, 0.1, 0.65, 0.03])
stime = Slider(
    ax=ax_time,
    label="time",
    valmin=0,
    valmax=zmax - 1,
    valstep=1,
    valinit=0
)

def update(val):
    time = int(stime.val)
    ax.imshow(mri[:,:,time])
    fig.canvas.draw_idle()


stime.on_changed(update)
plt.show()
```

- Aorta and Selected ROI

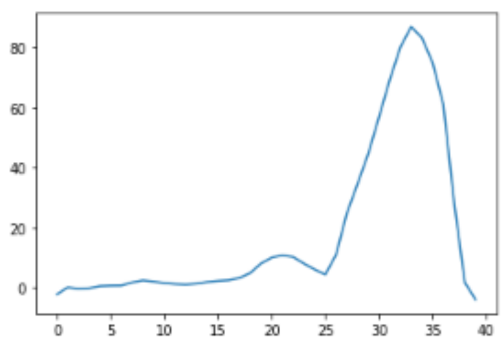




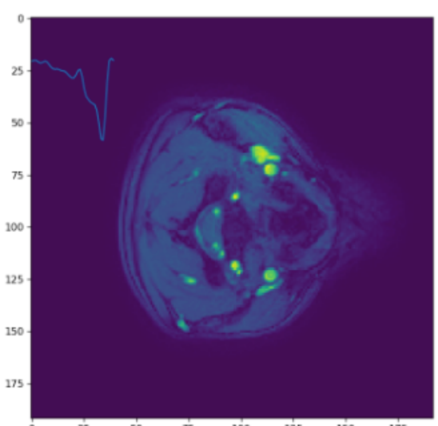
- Calculated the aorta peak

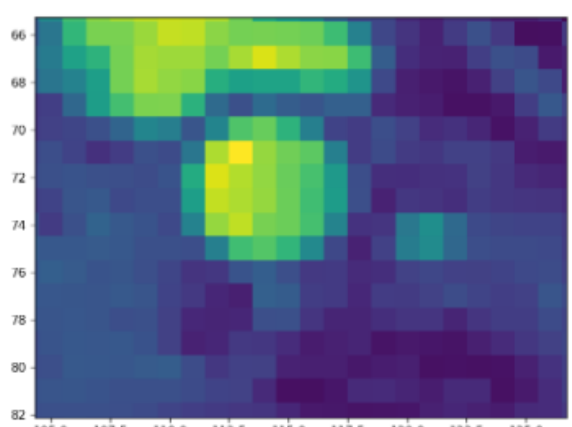
```
%matplotlib inline
aorta_timeline = np.mean(np.mean(aorta[121:132,128:145,0:40], axis=0), axis=0)
plt.plot(aorta_timeline)
peak_aorta = np.max(aorta_timeline)
print(peak_aorta)
```

```
86.87443770954316
```



- Using Carotid and selecting ROI



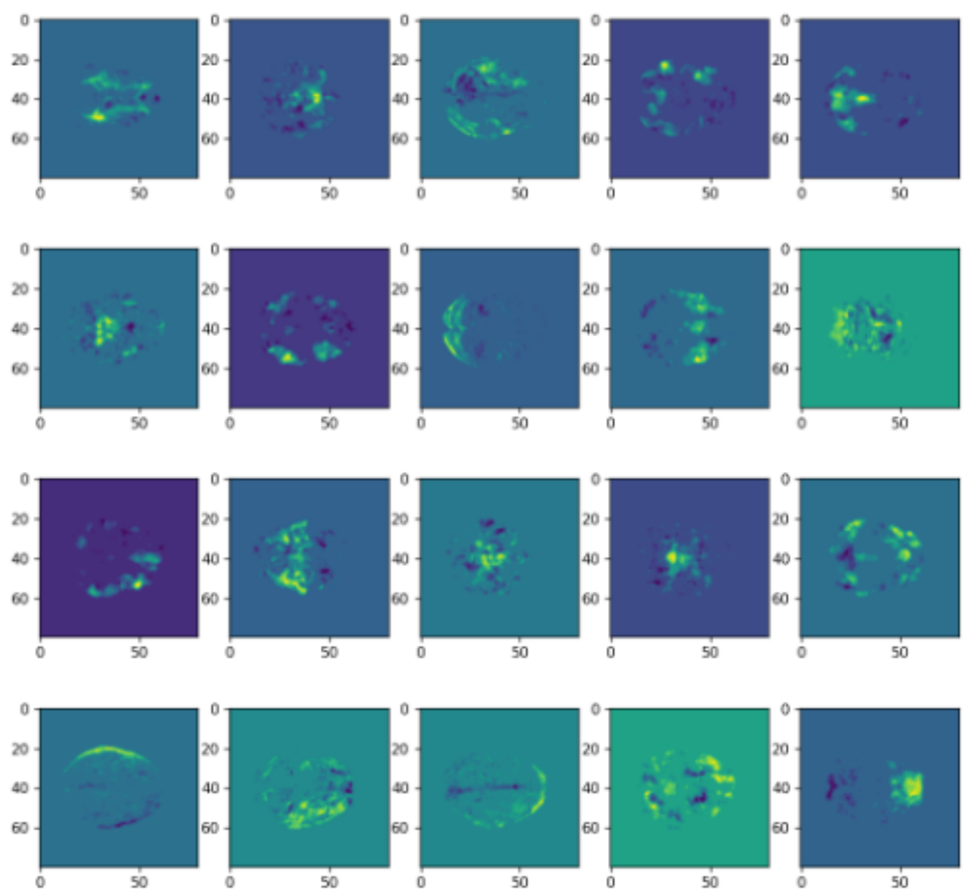


- Calculating the peak carotid

```
# sub1 [71:75, 106:108, 0:40]
# sub2 [67:70, 93:96, 0:40]
# sub3 [66:70, 94:97, 0:40]
# sub4 [72:74,111:116,0:40]
# sub5 [63:68,122:128,0:40]
# sub6 [67:71,97:102,0:40]
# sub7 [63:66,133:135,0:40]

carotid_timeline = np.mean(np.mean(carotid[72:74,111:116,0:40], axis=0), axis=0)
plt.plot(carotid_timeline)
peak_carotid = np.max(carotid_timeline)
print(peak_carotid)
```

```
58.373627471923825
```

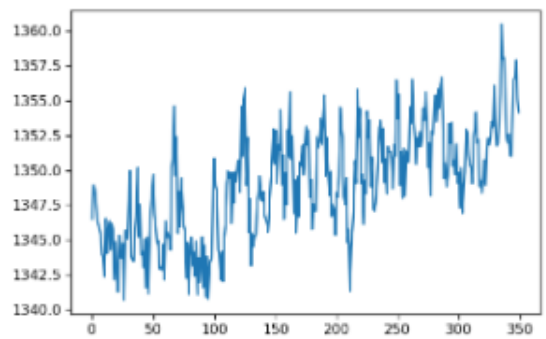
- Using CanICA and plotting for all the slices

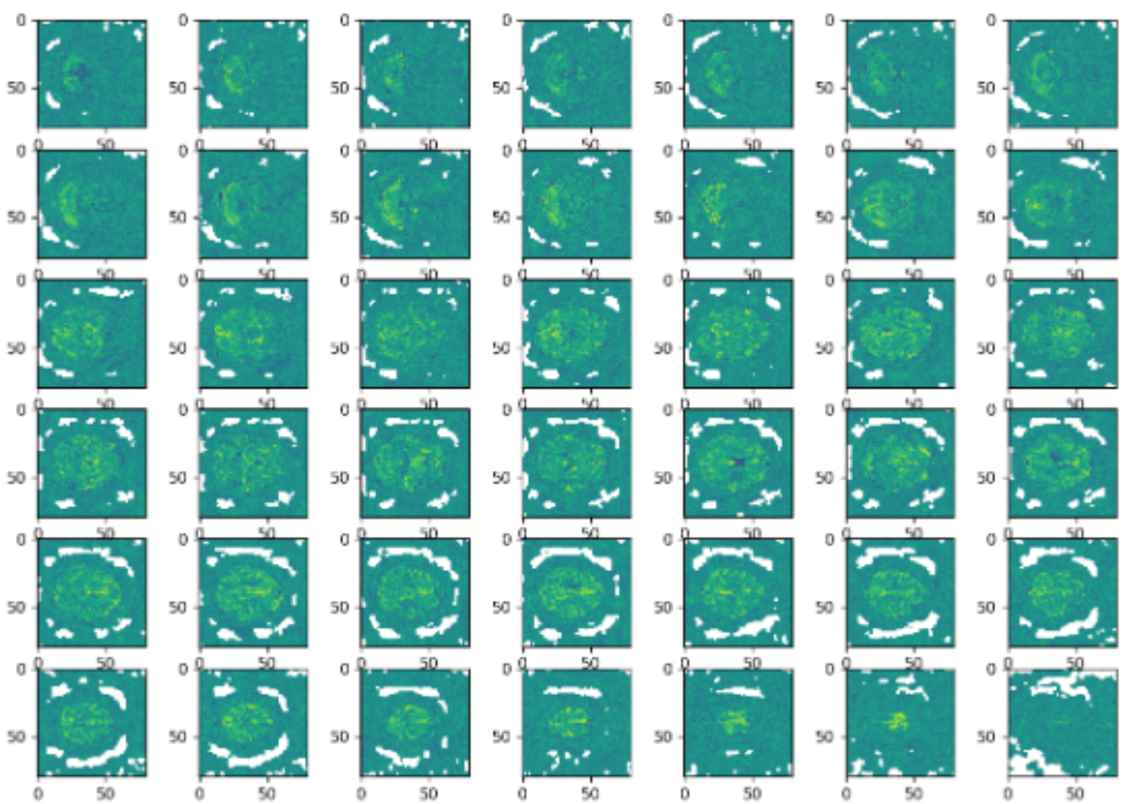


- Selecting the best image, plotting it, taking the best ROI and plotting the time-series data

```
%matplotlib notebook
# 1: [39:41,24:26,24:26,:]
# 2: [39:41,31:33,25:27,:]
# 3: [38:41,24:27,24:27,:]
# 4: [32:45,39:57,14:32,:]
# 5: [34:40,25:29,28:32,:]
# 6: [37:40,22:25,28:30,:]
# 7: [37:40,30:32,7:11,:]
plt.plot(np.mean(img[32:45,39:57,14:32,:],axis=(0,1,2)))
```

<IPython.core.display.Javascript object>

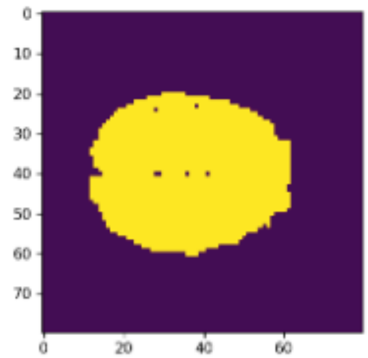




- Creating the best mask using the above results

```
%matplotlib notebook
mask = np.mean(img,axis=3)>900
plt.imshow(mask[:,:,21])
```

```
<IPython.core.display.Javascript object>
```



- Calculating the dmn_mean

```
dmn_mean = np.mean(r[:,:,21][mask[:,:,21]])
print(dmn_mean)
```

```
0.0951058450232986
```

- Now using the watch data, selecting the best one and calculating the bpm and hrv

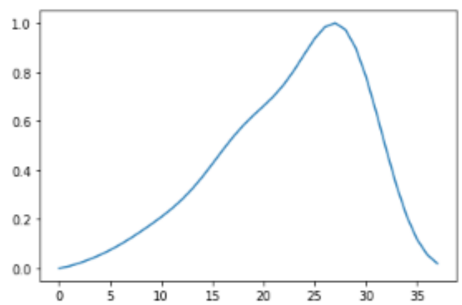
```
# ./subject_1/watch/_watch_01_1656214721.csv
# ./subject_2/watch/_watch_01_1647574027.csv
# ./subject_3/watch/_watch_01_1654928721.csv
# ./subject_4/watch/_watch_01_1655600680.csv
# ./subject_5/watch/_watch_01_1656835054.csv
# ./subject_6/watch/_watch_01_1657093094.csv
# ./

csv = pd.read_csv('./subject_4/watch/_watch_01_1655600680.csv', delimiter='\t')
table = csv.to_numpy()

print(table.shape)
plt.plot(table[:,1])

filt = butter_highpass_filter(table[:, 1], 0.6, 10, order=5)
res = resample(filt[-250:], 1000)
sample_rate = 40
plt.plot(res)

from scipy.signal import find_peaks

peaks, _ = find_peaks(res, distance=20)
bpm = peaks.size / (1000 / sample_rate) * 60
print(bpm)
hrv = np.std(np.diff(peaks))
print(hrv)
```

```
(599, 14)
62.400000000000006
2.645448922205832
```

- Finding the average of peaks and calculating the first and second derivative

```
%matplotlib inline
interval = sample_rate - 2
hbs = np.zeros([peaks.size - 2, interval])
bias = -8
for i in range(peaks.size - 2):
    hbs[i,:] = res[peaks[i+1] - interval // 2 + bias: peaks[i+1] + interval // 2 + bias]
hbs_mean = np.mean(hbs, axis=0)
hbs_norm = normalize(hbs_mean)
plt.plot(hbs_norm)
peak_index = np.argmax(hbs_norm)
first_deriv = np.mean(np.diff(hbs_norm)[0:peak_index])
second_deriv = np.mean(np.diff(hbs_norm, n=2)[0:peak_index])
print(first_deriv, second_deriv)
```

```
0.037037037037037035 -0.0012974344994988393
```



- Created a library named leftventricearea. With a single run, it calculates the inner and outer ring area. The input will be thresholds, seed point, closing, and dilation values for all the time slices.

```
%matplotlib inline
import leftventiclearea
area_outerring, area = leftventiclearea.left_ventricle(lv,2,
                      [300,300,300,300,300,
                       300,300,300,300,300,200,
                      300,300,200,200,200,200,200,
                      200,200,200,200,200,200,200],[(121,121)],[(5,5)],
                       [8,8,8,8,9,9,10,10,8,8,10,10,8,8,8,8,6,6,6,7,7,6,6,5])
```

- Now gathering all the features in one place to find a correlation

```
features_mri = np.array([dmn_mean , peak_carotid, min(area),max(area_outerring),peak_aorta])
features_sw = np.array([bpm, hrv, first_deriv, second_deriv])
```

```
features_mri
```

```
array([9.51058450e-02, 5.83736275e+01, 2.86000000e+02, 9.87314957e+00,
       8.68744377e+01])
```

```
features_sw
```

```
array([ 6.24000000e+01,  2.64544892e+00,  3.70370370e-02, -1.29743450e-03])
```

- Correlation results had already been included

Conclusion:
Much more research needs to be done to find accurate values but overall, this will be a great way to start and find approximate relationships.