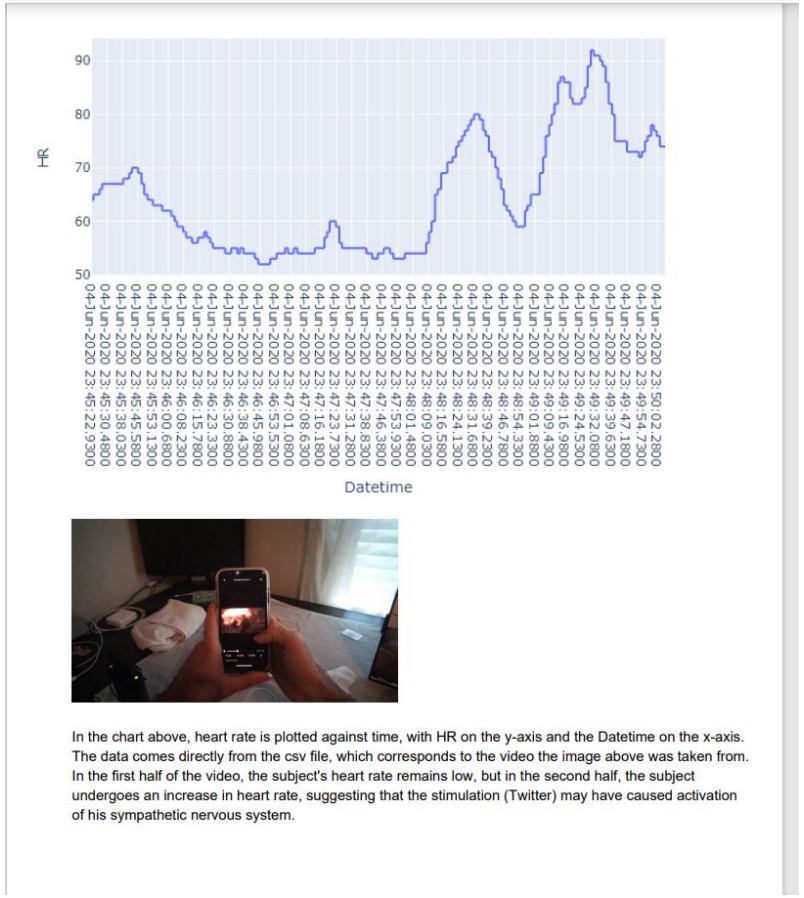
WEEK #1 DELIVERABLES

Cristian Garces

Objectives

- Get familiar with the dataset
- Get more comfortable with Python and the libraries used in the project
- Use dataset to generate plots with the biometric data
- Isolate and extract frames from video
- Generate a PDF file and insert the plots and the extracted frames

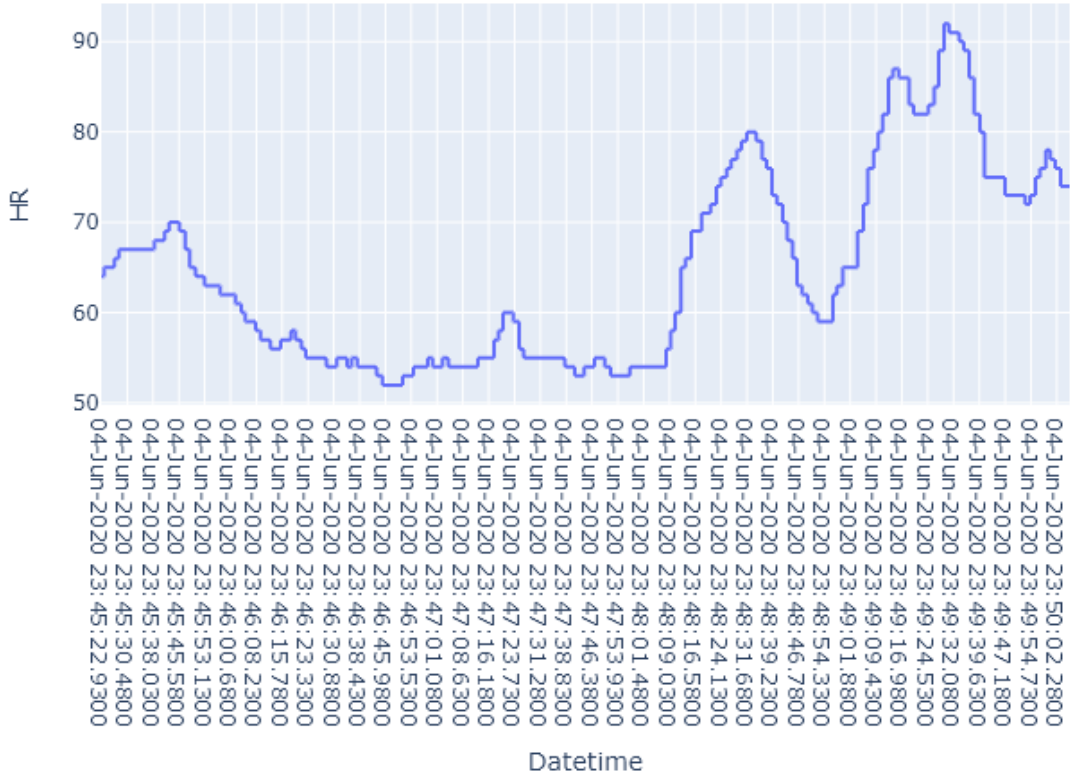
Outcomes



In the chart above, heart rate is plotted against time, with HR on the y-axis and the Datetime on the x-axis. The data comes directly from the csv file, which corresponds to the video the image above was taken from. In the first half of the video, the subject's heart rate remains low, but in the second half, the subject undergoes an increase in heart rate, suggesting that the stimulation (Twitter) may have caused activation of his sympathetic nervous system.

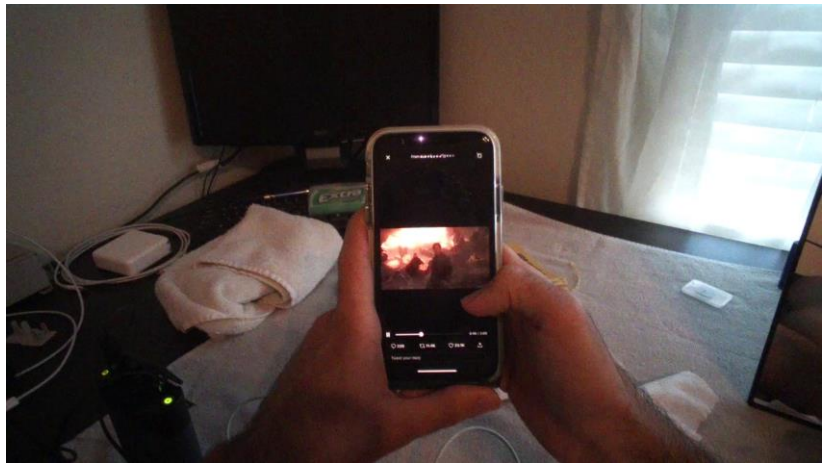
- Screenshot of generated PDF
- PDF was generated and formatted using the PyMuPDF library
- Generated PDF contains the biometric plot, the extracted frame of the video, and a paragraph about the chosen biometric.
- Chosen biometric: Heart Rate
  - In the plot, heart rate was plotted against time

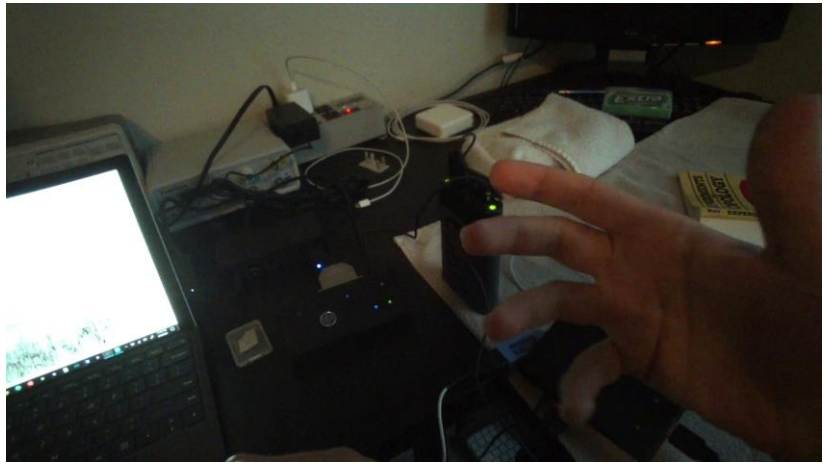
Outcomes



- Closer view of the generated biometric plot
  - HR vs. Datetime
- Plot generated with Plotly Express, taking data directly from the csv file using Pandas

Outcomes





- Various frames extracted from the provided video
- Frames extracted using the cv2 library

Outcomes

```python
import pandas as pd
import fitz
import cv2
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import plotly.express as px
from fitz import Rect

# Imports the dataset and isolates the HR column as a Pandas Series.
dataset = pd.read_csv('2020_06_04_T05_U00T_ADELE.csv')
heartRate = dataset["HR"]

# Creates a plot of HR vs. Datetime and saves it as a png
fig = px.line(dataset, x='Datetime', y='HR')
#fig.show()
fig.write_image("testPlot.png")

# Processes the video frame by frame and saves every 25th frame as a jpg (frame rate ~ 25fps)
# This ensures that every frame corresponds to roughly one second of video.
vidcap = cv2.VideoCapture('fullstream.mp4')
readSuccess, image = vidcap.read()
counter = 0
while readSuccess:
    if ( (counter % 25) == 0 ): #framerate roughly 25 fps, this saves one frame per second of video
        cv2.imwrite("frame%d.jpg" % counter, image)
    readSuccess, image = vidcap.read()
    print('Read a new frame: ', readSuccess)
    counter += 1

# Creates a new blank PDF
doc = fitz.open()
generatedPage = doc.newPage()
```

- First half of code
- Contains all library calls
- Imports dataset and creates the plot with the chosen parameters
- Extracts the chosen frames from the video and saves the images
- Generates a new PDF

Outcomes

```
34
35  # Inserts both the images of the biometric plot and the specified frame from the video
36  bioPlotImage = Rect(0, -800, 570, 1230)
37  generatedPage.insertImage(bioPlotImage, filename="testPlot.png")
38  vidFrameImage = Rect(50, 375, 300, 625)
39  generatedPage.insertImage(vidFrameImage, filename="frame5600.jpg")
40
41  # Inserts the text below the images
42  text = "In the chart above, heart rate is plotted against time, with HR on the y-axis and the Datetime on the x-axis.\n" \
43         "The data comes directly from the csv file, which corresponds to the video the image above was taken from.\n" \
44         "In the first half of the video, the subject's heart rate remains low, but in the second half, the subject\n" \
45         "undergoes an increase in heart rate, suggesting that the stimulation (Twitter) may have caused activation\n" \
46         "of his sympathetic nervous system."
47  startPoint = fitz.Point(50, 600)
48  generatedPage.insertText(startPoint, text, fontname= "helv", fontsize = 11, rotate = 0)
49
50  # Saves the PDF
51  doc.save("biometricPlots.pdf")
```

- Second half of code
- Inserts the plot, an extracted frame, and the paragraph about the biometric into the generated PDF
- Saves the PDF with the specified name

Difficulties

- Python learning curve
  - Installing new libraries
    - “pip install *library name*” = easiest way to install new libraries
  - Syntax
    - Python does not have data types
    - Semicolons are NOT needed after every line
  - Using libraries
    - Most of time spent was reading through library documentation