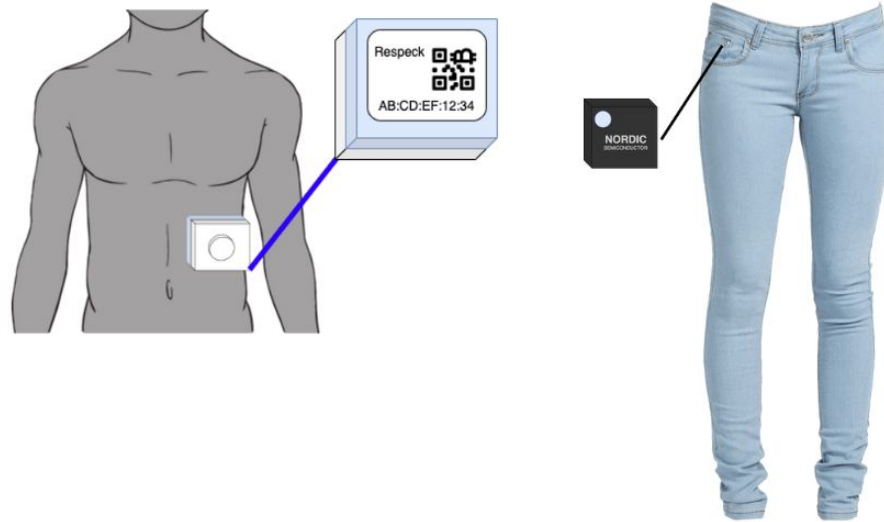


Data Collection and Coursework 1

Principles and Design of IoT Systems

Coursework 1 Task Summary

In this coursework, you will use the wearable devices Thingy and RESpeck (which are connected to the PDIoT app) while engaging in various physical activities and run code to preprocess the data you have collected so that it is ready to be used in Coursework 3.



Activities to be recorded

You will have to perform a series of activities while wearing the sensor. Each activity should be performed for **30(\pm 2) seconds**.

14 Physical activities - done while wearing the RESpeck and Thingy

Sitting, standing, lying down on your left side, lying down on your right side, lying down on your back, lying down on your stomach, walking normally, ascending stairs, descending stairs, shuffle walking, running/jogging, and miscellaneous movements while *breathing normally*.

Activities to be recorded

12 Stationary activities with respiratory responses - done while wearing the RESpeck only

Sitting, standing, lying down on your left side, lying down on your right side, lying down on your back, and lying down on your stomach while *coughing* and *hyperventilating*.

20 Stationary activities with other behaviors - done while wearing the RESpeck

Sitting and standing while *talking*, *eating*, *singing*, and *laughing*.

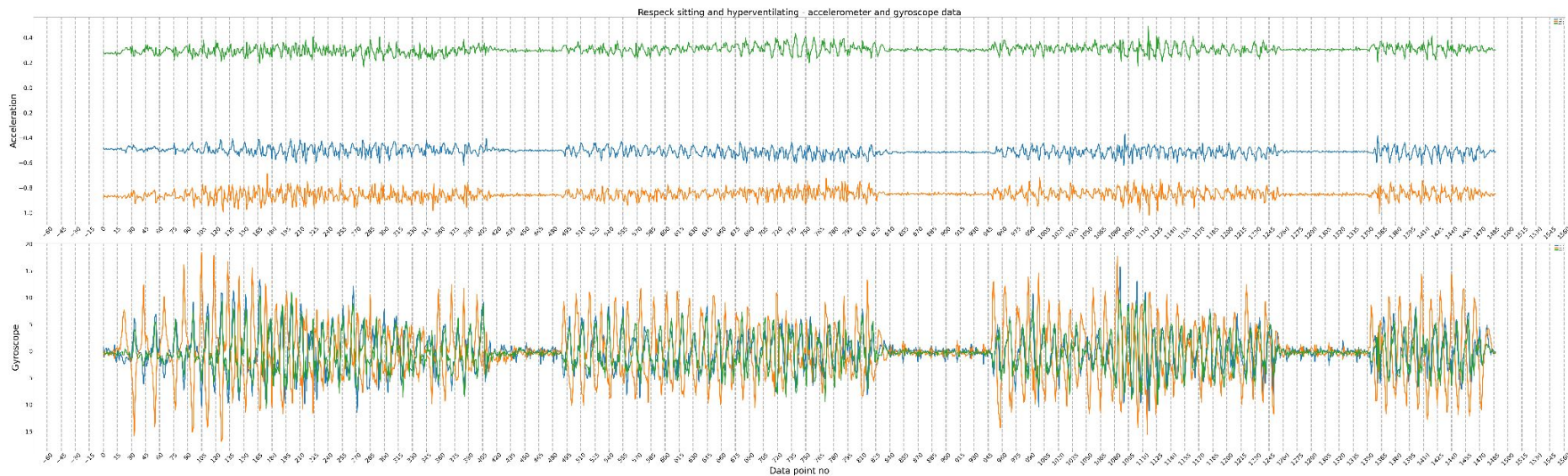
Lying down on back, lying down on left side, lying down on right side, and lying down on stomach while *talking*, *singing*, and *laughing*.

Data Preprocessing and Cleaning

- Ensure that you have a total of 46 recorded activities (60 recorded data files)
 - 28 data files for the physical activities
 - 12 data files for the stationary activities with respiratory responses
 - 20 data files for stationary activities with other behaviors
- Ensure that each of your data files is 30(\pm 2) seconds long/
- Ensure that each of your data files has a sampling rate of 25 Hz.
- Ensure that there are no unnecessary gaps of inactivity in your data recordings.
- Ensure that only ONE activity is being performed in each of your data recordings.
- Visually verify your recordings so that activity starts at the very beginning and stops at the very end of the recording

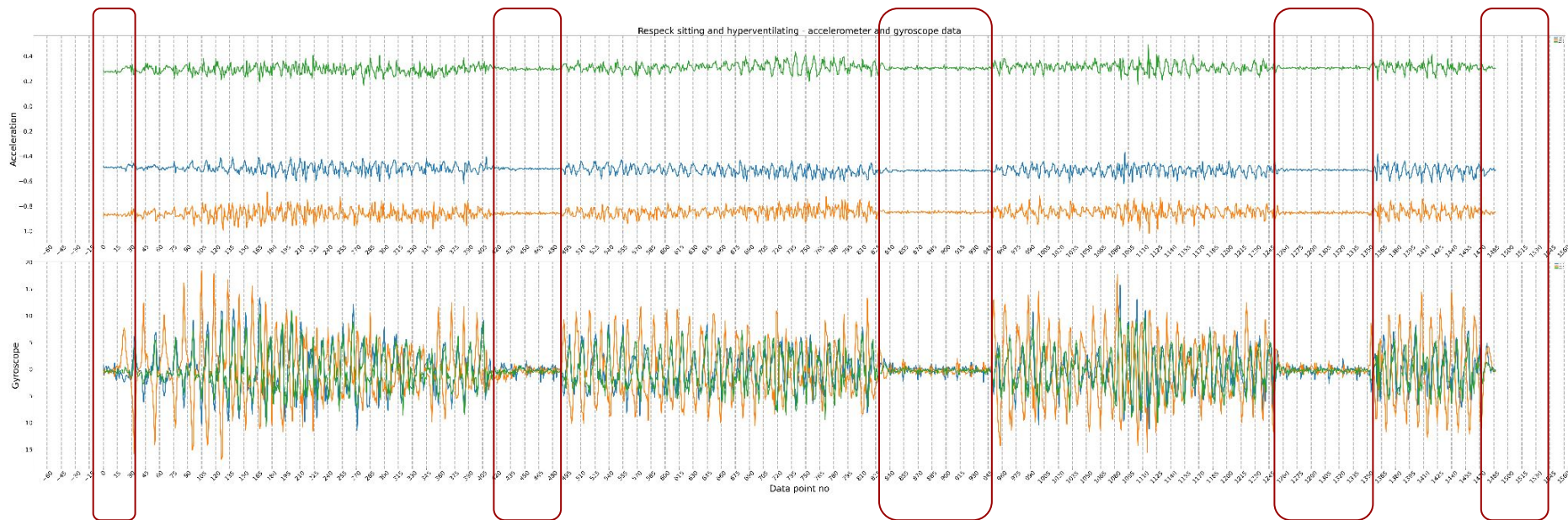
Example - raw unprocessed data

Respeck sitting and hyperventilating



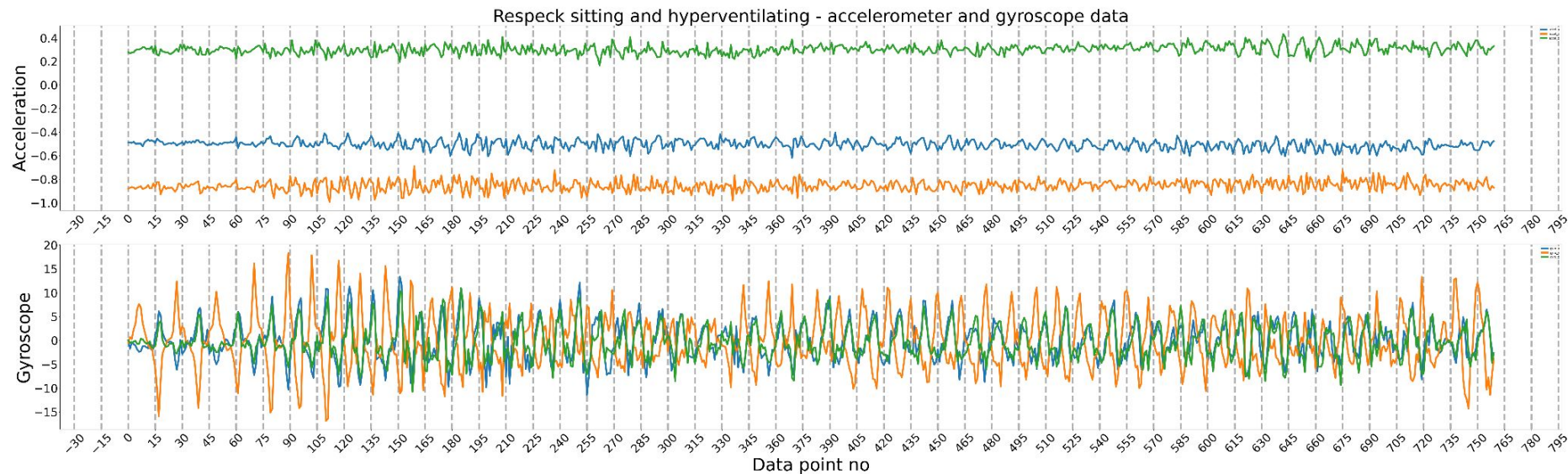
Example - raw unprocessed data

Respeck sitting and hyperventilating



Example - cleaned and trimmed data

Respeck sitting and hyperventilating



Data cleaning process

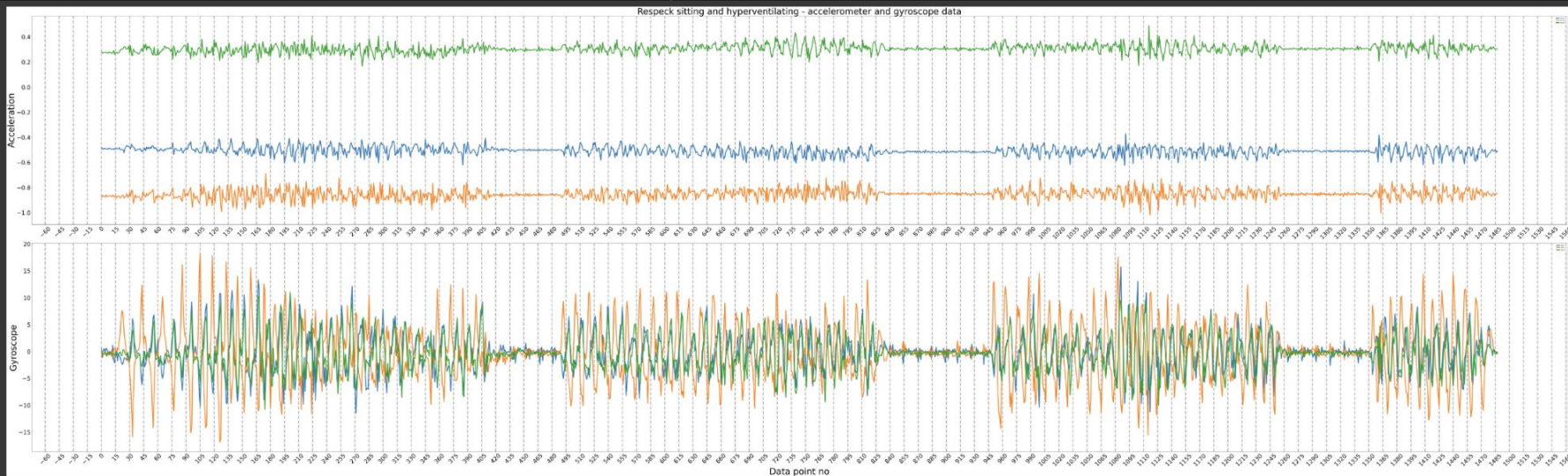
Filtering out gaps of inactivity in data

```
[ ] filename_respeck = "/content/gdrive/SharedDrives/Ink/PDIOI/Respeck_s2255740_Sitting_06-09-2023_21-26-26.csv"  
df_respeck = pd.read_csv(filename_respeck, header=5)
```

Plotting raw and unprocessed data

Note: This may take a while!

```
[ ] plot_data(df_respeck, "Respeck sitting and hyperventilating - accelerometer and gyroscope data")
```



▾ Deleting gaps

```
df_respeck['ind'] = df_respeck.index

to_trim = input("How many data ranges would you like to trim? ")

print("\n")

for i in range(int(to_trim)):

    print(i+1, "Specify the range of the indexes that you would like to delete -----")

    range_trim_start = int(input("Starting at index: "))
    range_trim_end = int(input("Ending at index: "))

    df_respeck = df_respeck[~((df_respeck['ind'] >= range_trim_start) & (df_respeck['ind'] <= range_trim_end))]

    print("\n")
```

☞ How many data ranges would you like to trim? 5

1 Specify the range of the indexes that you would like to delete -----
Starting at index: 0
Ending at index: 15

2 Specify the range of the indexes that you would like to delete -----
Starting at index: 410
Ending at index: 490

3 Specify the range of the indexes that you would like to delete -----
Starting at index: 830
Ending at index: 945

4 Specify the range of the indexes that you would like to delete -----
Starting at index: 1250
Ending at index: 1350

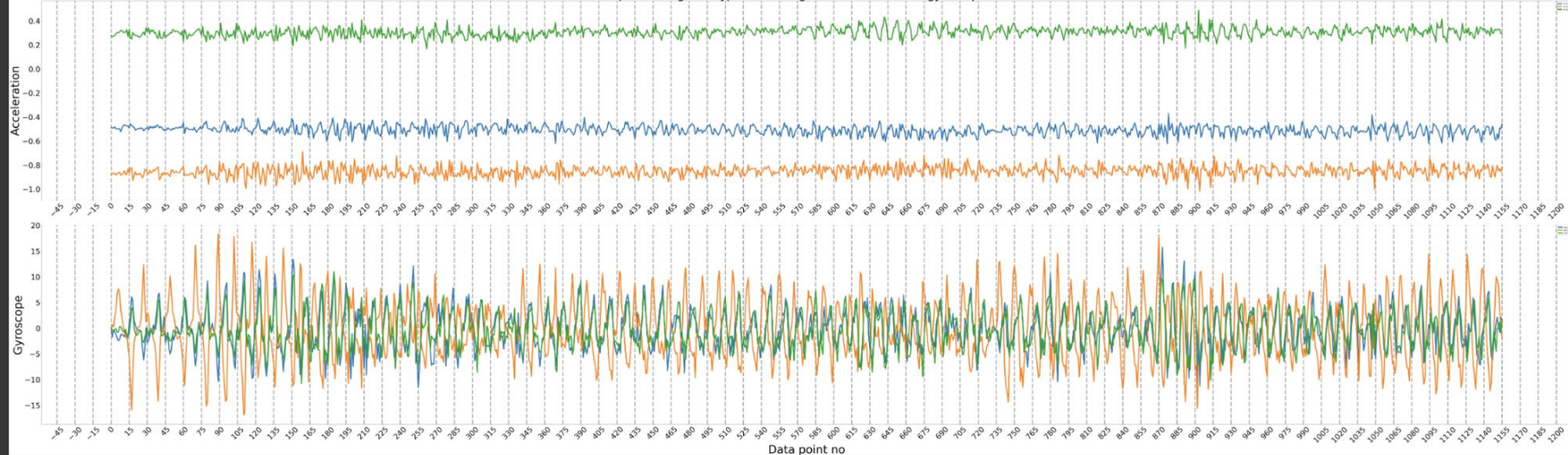
5 Specify the range of the indexes that you would like to delete -----
Starting at index: 1470
Ending at index: 1500

Plot resulting graph

```
print("RESULTING GRAPH -----")  
df_respeak.reset_index(inplace=True, drop=True)  
plot_data(df_respeak, "Respeak sitting and hyperventilating - accelerometer and gyroscope data")
```

RESULTING GRAPH -----

Respeak sitting and hyperventilating - accelerometer and gyroscope data



▾ Readjusting the timestamps to get 25Hz frequency data

```
▶ get_frequency(df_respeck)
```

20.02390396840519

```
▶ # Define the starting timestamp in milliseconds
start_timestamp_ms = df_respeck.timestamp[0]

# Define the number of timestamps you want to generate
num_timestamps = len(df_respeck)

# Calculate the time interval between timestamps in microseconds
microseconds_per_timestamp = int(1e6 / 25)

# Initialize a list to store the generated timestamps
timestamps = []

# Generate the timestamps
for i in range(num_timestamps):
    timestamp = start_timestamp_ms + i * microseconds_per_timestamp // 1000 # Convert microseconds to milliseconds
    timestamps.append(timestamp)

# Print the generated timestamps
# for timestamp in timestamps:
#     formatted_time = datetime.datetime.fromtimestamp(timestamp / 1000).strftime('%Y-%m-%d %H:%M:%S.%F')
#     print(timestamp, formatted_time[:-3]) # Print the timestamp with milliseconds

df_respeck['timestamp'] = timestamps
```

```
[ ] get_frequency(df_respeck)
```

25.02164502164502

Trimming data to 30s

```
[ ] get_recording_length(df_respeck)
```

```
46.2
```

```
[ ] get_recording_length(df_respeck[:760])
```

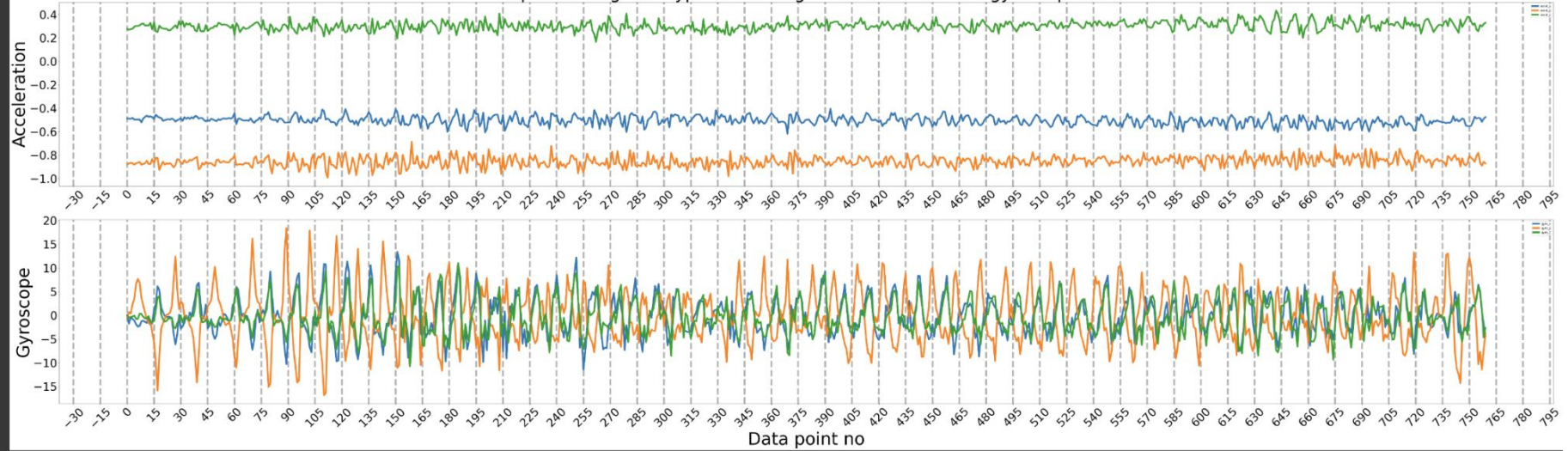
```
30.360000000000003
```

```
[ ] df_respeck = df_respeck[:760]
```

```
[ ] plot_data(df_respeck, "Respeck sitting and hyperventilating - accelerometer and gyroscope data")
```



Respeck sitting and hyperventilating - accelerometer and gyroscope data



Submission

You should submit both your clean and unprocessed (original) data files to the PDIoT GitHub repository.

More details on the submission guidelines can be found in the Coursework 1 document.

Resources

All resources to complete Coursework 1 can be found in:

- [Lab 1.1 Instructions](#) - Connecting sensors to the mobile app
- [Lab 1.2 Instructions](#) - Collecting physical activity data
- [Lab 2 Instructions](#) - Data collection protocol: data cleaning and trimming
- Lab 2 Python Notebooks for visualizing and cleaning data
- [Coursework 1 Document](#)

All materials will be made available on **Learn** and the [PDIoT website.](#)