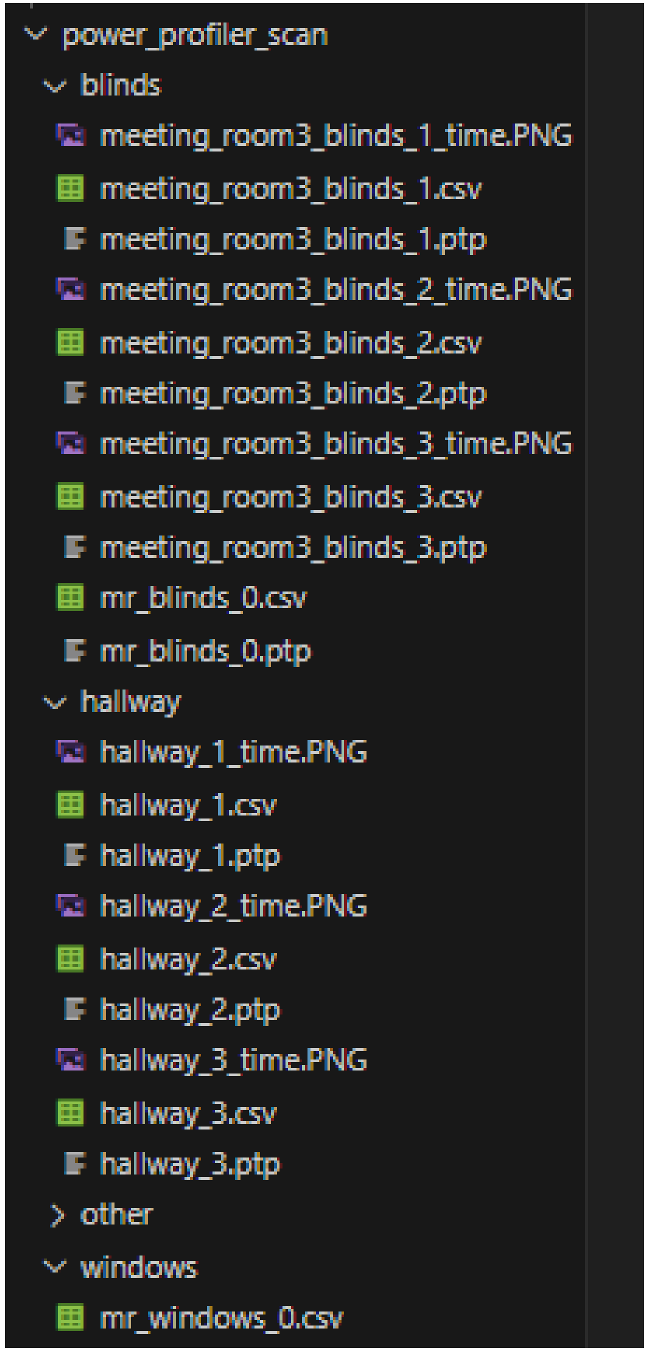
AR Security 10/25

Allie Craddock & Casie Peng

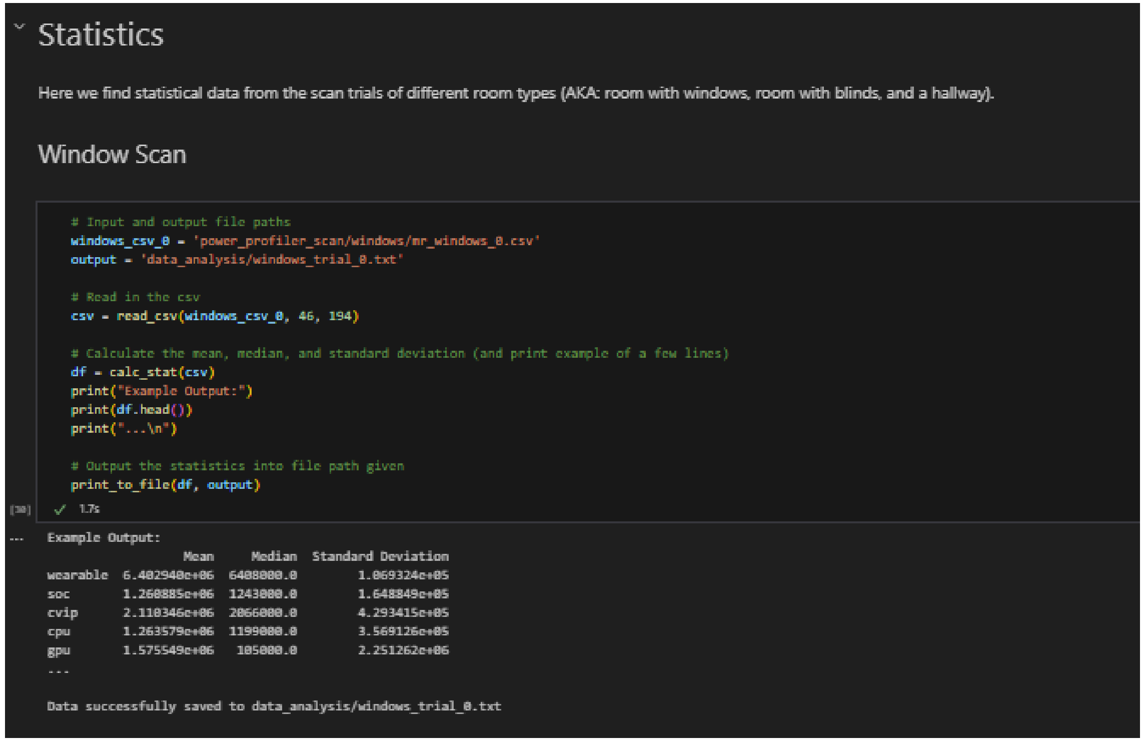
Data Collection

- Headset will still no longer connect to device
- Scanning opportunities halved
- Still need to retake a few more scans
- Focused on:
  - Hallway
  - Meeting Room (Blinds Drawn)
  - Meeting Room (Windows Exposed)
  - Shorter Scans (One Wall Reading)
    - Window vs. Blank Wall
    - Blank Wall vs. Decorated Wall



```
v power_profiler_scan
  v blinds
      meeting_room3_blinds_1_time.PNG
      meeting_room3_blinds_1.csv
      meeting_room3_blinds_1.ptp
      meeting_room3_blinds_2_time.PNG
      meeting_room3_blinds_2.csv
      meeting_room3_blinds_2.ptp
      meeting_room3_blinds_3_time.PNG
      meeting_room3_blinds_3.csv
      meeting_room3_blinds_3.ptp
      mr_blinds_0.csv
      mr_blinds_0.ptp
  v hallway
      hallway_1_time.PNG
      hallway_1.csv
      hallway_1.ptp
      hallway_2_time.PNG
      hallway_2.csv
      hallway_2.ptp
      hallway_3_time.PNG
      hallway_3.csv
      hallway_3.ptp
  > other
  v windows
      mr_windows_0.csv
```

Pandas (cont.)



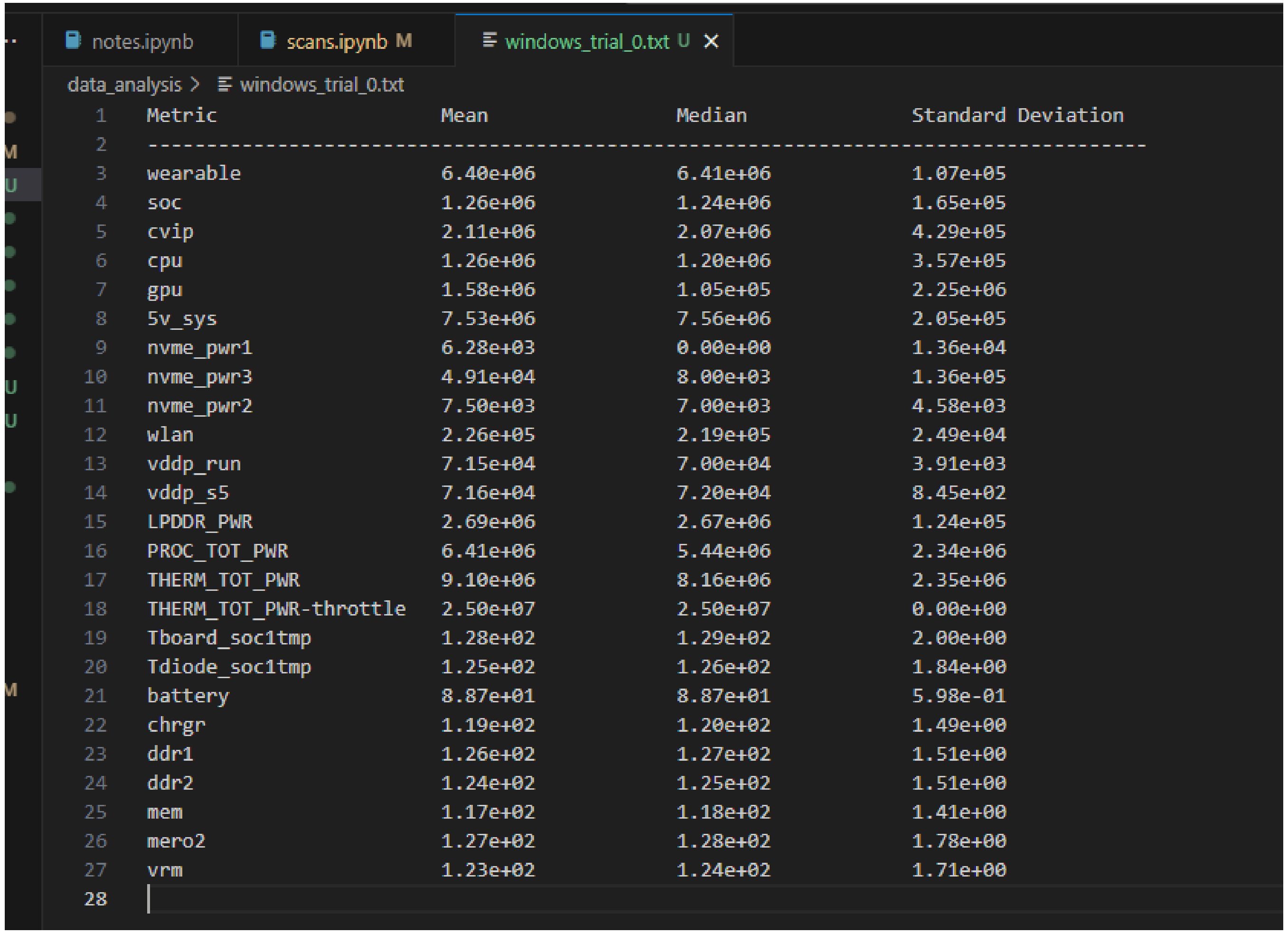
Statistics

Here we find statistical data from the scan trials of different room types (AKA: room with windows, room with blinds, and a hallway).

Window Scan

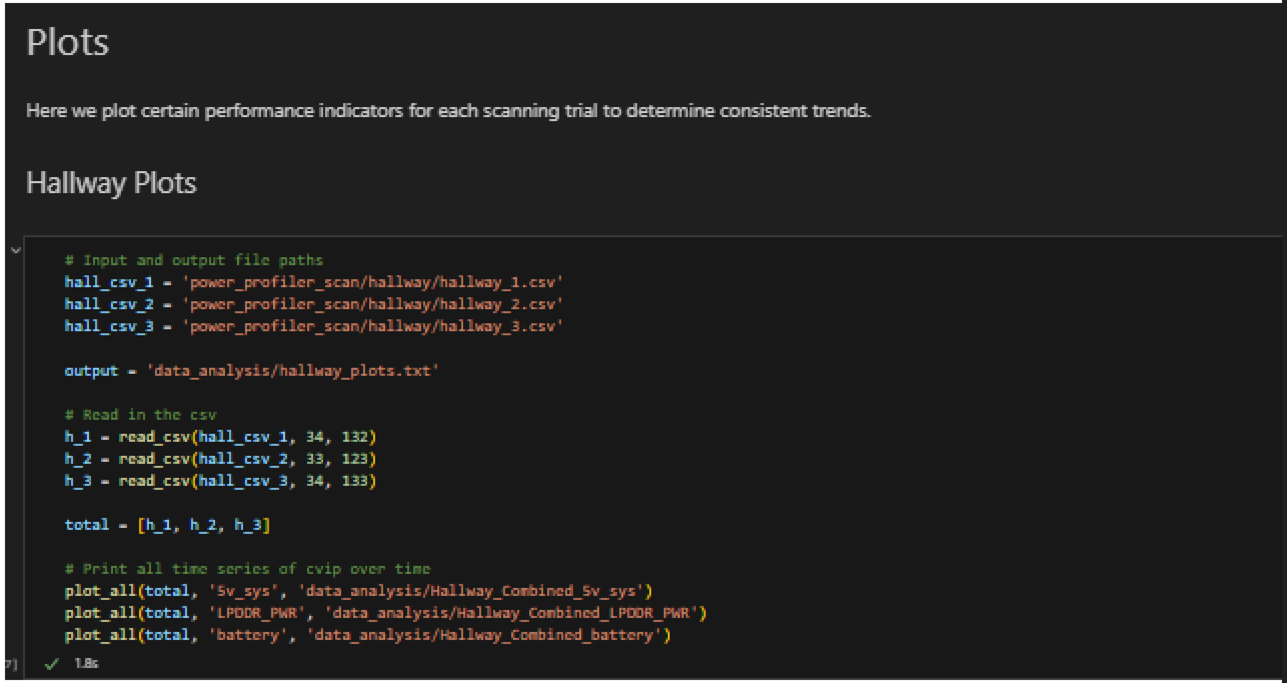
```python
# Input and output file paths
windows_csv_0 = 'power_profiler_scan/windows/mr_windows_0.csv'
output = 'data_analysis/windows_trial_0.txt'

# Read in the csv
csv = read_csv(windows_csv_0, 46, 194)

# Calculate the mean, median, and standard deviation (and print example of a few lines)
df = calc_stat(csv)
print("Example Output:")
print(df.head())
print("...\n")

# Output the statistics into file path given
print_to_file(df, output)
```

```
Example Output:
                 Mean     Median  Standard Deviation
wearable  6.402940e+06  6408000.0        1.069324e+05
soc       1.260885e+06  1243000.0        1.648849e+05
cvip      2.110346e+06  2066000.0        4.293415e+05
cpu       1.263579e+06  1199000.0        3.569126e+05
gpu       1.575549e+06   105000.0        2.251262e+06
...

Data successfully saved to data_analysis/windows_trial_0.txt
```



```
Metric                  Mean        Median      Standard Deviation
------------------------------------------------------------------------------------------
wearable                6.40e+06    6.41e+06    1.07e+05
soc                     1.26e+06    1.24e+06    1.65e+05
cvip                    2.11e+06    2.07e+06    4.29e+05
cpu                     1.26e+06    1.20e+06    3.57e+05
gpu                     1.58e+06    1.05e+05    2.25e+06
5v_sys                  7.53e+06    7.56e+06    2.05e+05
nvme_pwr1               6.28e+03    0.00e+00    1.36e+04
nvme_pwr3               4.91e+04    8.00e+03    1.36e+05
nvme_pwr2               7.50e+03    7.00e+03    4.58e+03
wlan                    2.26e+05    2.19e+05    2.49e+04
vddp_run                7.15e+04    7.00e+04    3.91e+03
vddp_s5                 7.16e+04    7.20e+04    8.45e+02
LPDDR_PWR               2.69e+06    2.67e+06    1.24e+05
PROC_TOT_PWR            6.41e+06    5.44e+06    2.34e+06
THERM_TOT_PWR           9.10e+06    8.16e+06    2.35e+06
THERM_TOT_PWR-throttle  2.50e+07    2.50e+07    0.00e+00
Tboard_soc1tmp          1.28e+02    1.29e+02    2.00e+00
Tdiode_soc1tmp          1.25e+02    1.26e+02    1.84e+00
battery                 8.87e+01    8.87e+01    5.98e-01
chrgr                   1.19e+02    1.20e+02    1.49e+00
ddr1                    1.26e+02    1.27e+02    1.51e+00
ddr2                    1.24e+02    1.25e+02    1.51e+00
mem                     1.17e+02    1.18e+02    1.41e+00
mero2                   1.27e+02    1.28e+02    1.78e+00
vrm                     1.23e+02    1.24e+02    1.71e+00
```

Matplotlib

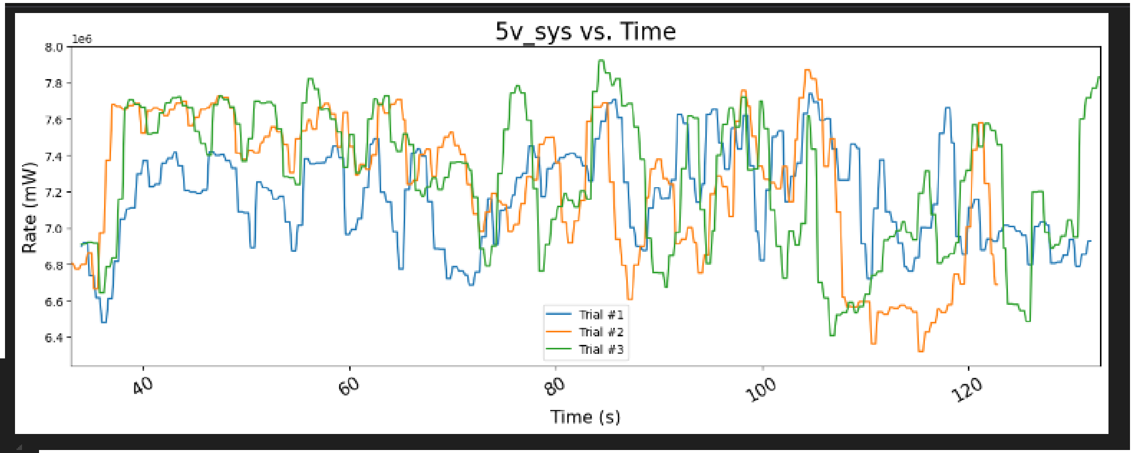


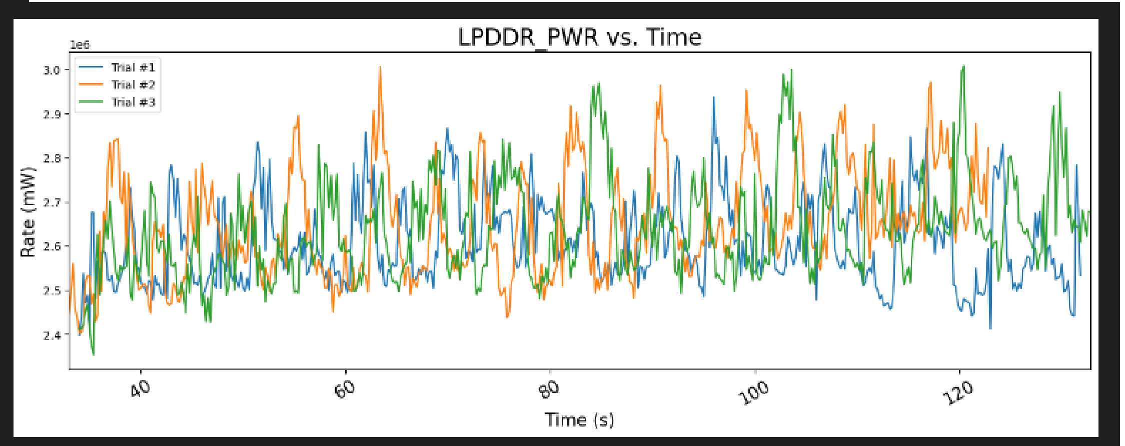
Plots

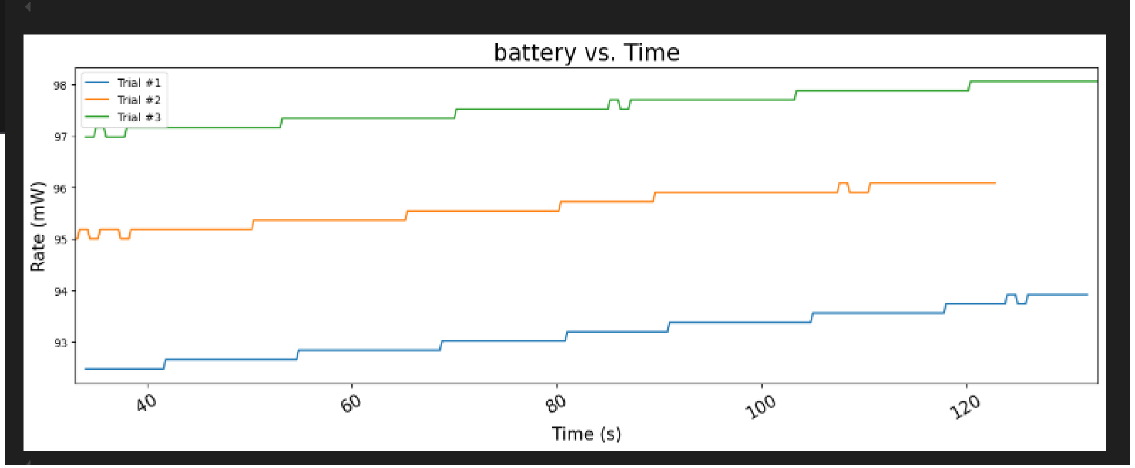
Here we plot certain performance indicators for each scanning trial to determine consistent trends.

Hallway Plots

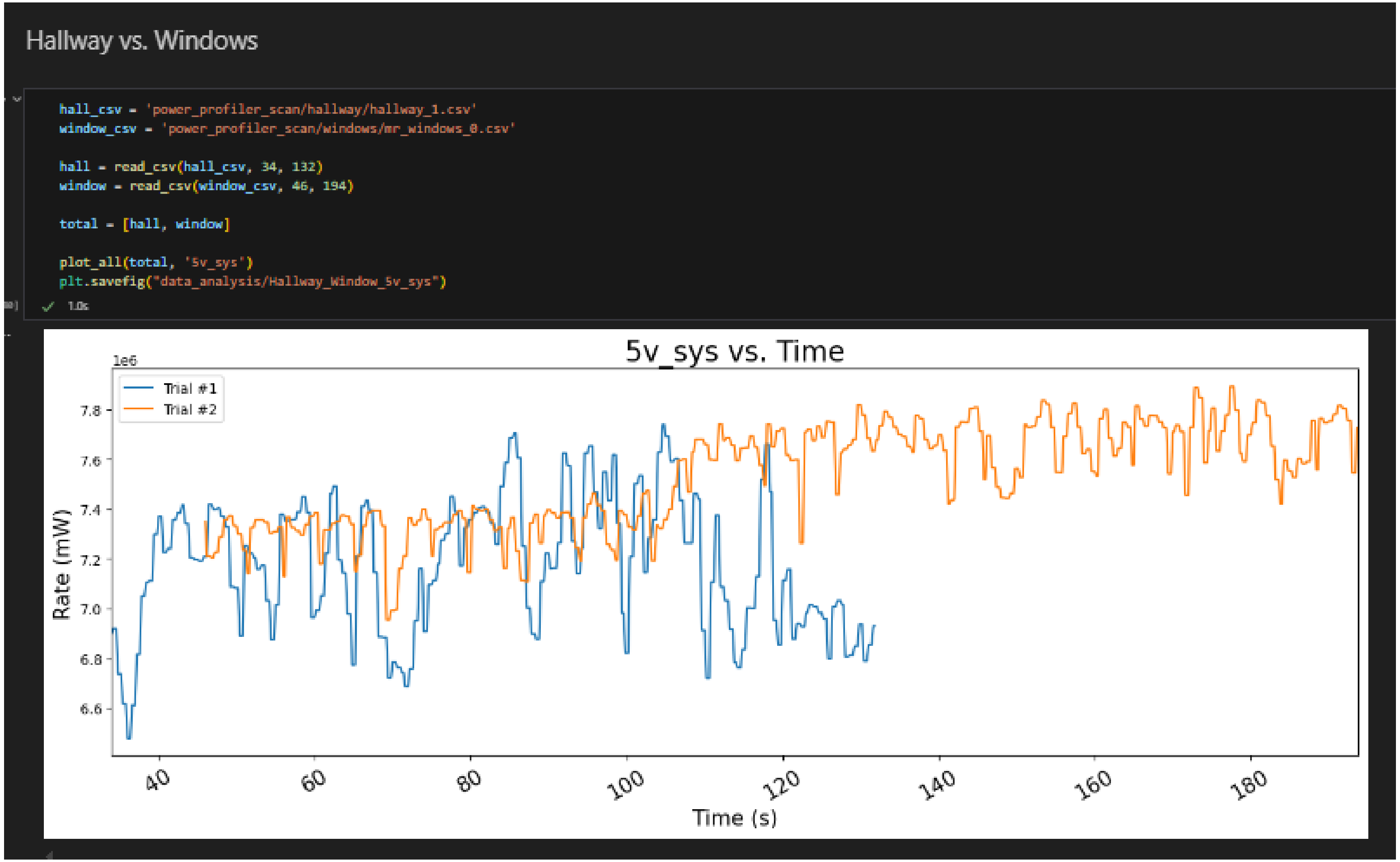
```
# Input and output file paths
hall_csv_1 = 'power_profiler_scan/hallway/hallway_1.csv'
hall_csv_2 = 'power_profiler_scan/hallway/hallway_2.csv'
hall_csv_3 = 'power_profiler_scan/hallway/hallway_3.csv'

output = 'data_analysis/hallway_plots.txt'

# Read in the csv
h_1 = read_csv(hall_csv_1, 34, 132)
h_2 = read_csv(hall_csv_2, 33, 123)
h_3 = read_csv(hall_csv_3, 34, 133)

total = [h_1, h_2, h_3]

# Print all time series of cvip over time
plot_all(total, '5v_sys', 'data_analysis/Hallway_Combined_5v_sys')
plot_all(total, 'LPDDR_PWR', 'data_analysis/Hallway_Combined_LPDDR_PWR')
plot_all(total, 'battery', 'data_analysis/Hallway_Combined_battery')
```







Matplotlib (cont.)



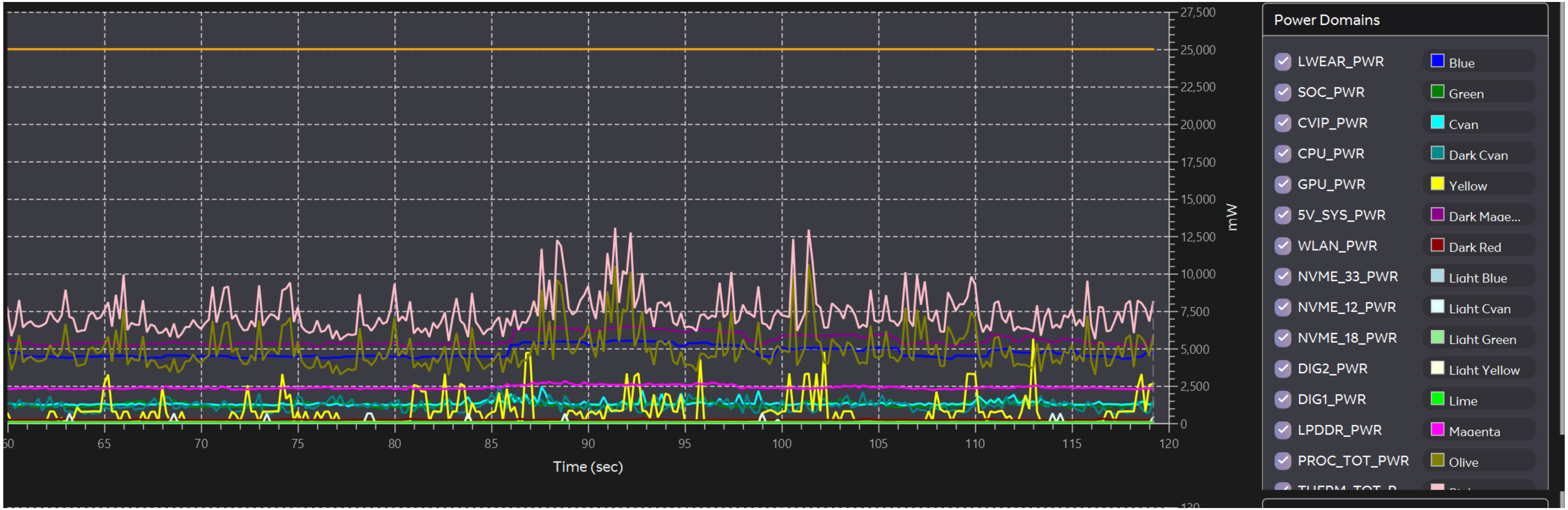
Hallway vs. Windows

```
hall_csv = 'power_profiler_scan/hallway/hallway_1.csv'
window_csv = 'power_profiler_scan/windows/mr_windows_0.csv'

hall = read_csv(hall_csv, 34, 132)
window = read_csv(window_csv, 46, 194)

total = [hall, window]

plot_all(total, '5v_sys')
plt.savefig("data_analysis/Hallway_Window_5v_sys")
```

✓ 1.0s

Unity – Library Consultant

- Information Observed:
  - Renders and uses power more when close to things
  - When in a large room, less render/power compared to smaller room
  - Possibility: Can see spikes when person turns head to unrendered space
- Plan: Use ML power profiler while running unity meshing project
  - Need to create working meshing project
  - Will switch to Unreal if truly unsuccessful



Power Domains

☑ LWEAR_PWR	🟦 Blue
☑ SOC_PWR	🟩 Green
☑ CVIP_PWR	■ Cyan
☑ CPU_PWR	■ Dark Cyan
☑ GPU_PWR	🟨 Yellow
☑ 5V_SYS_PWR	🟪 Dark Mage...
☑ WLAN_PWR	🟥 Dark Red
☑ NVME_33_PWR	■ Light Blue
☑ NVME_12_PWR	■ Light Cyan
☑ NVME_18_PWR	■ Light Green
☑ DIG2_PWR	■ Light Yellow
☑ DIG1_PWR	🟩 Lime
☑ LPDDR_PWR	■ Magenta
☑ PROC_TOT_PWR	■ Olive
☑ THERM_TOT_P	■

Current Questions