Appendix E: Journey of Data Analysis

D.1 First Attempt (no aggregation)

After mapping the data, Nathan's data looked like Figure E.01.

Then, we tried only looking at the data after Sept 19 and set the productivity level to -1 if there was no manual rating, but no obvious pattern showed up (see Figure E.02).

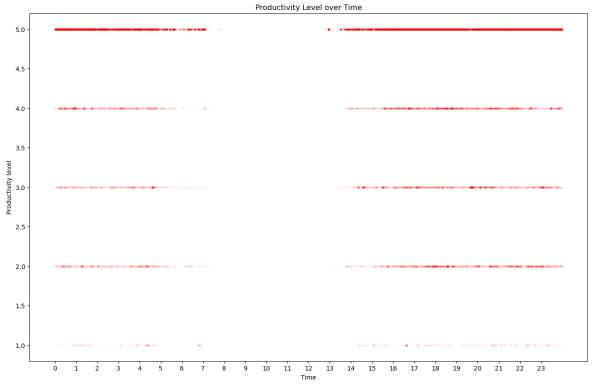

Figure E.01. Plot all the productivity data on a scatter plot. Opacity = 0.005 (#01)

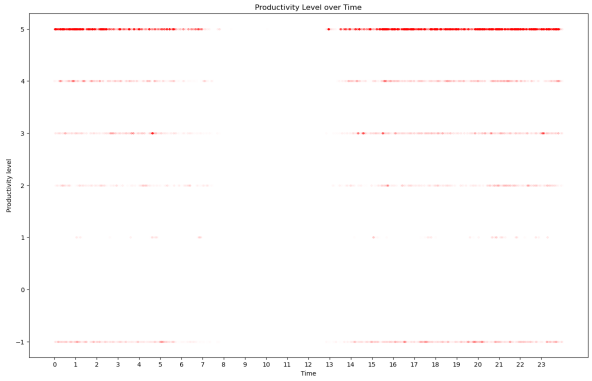

Figure E.02. Filter data only after Sept 19. Opacity = 0.005 (#02)

It was odd that there was data missing in the middle. After some more investigation, we realized that we needed to apply a timezone (see Figure E.03).

Even then, the data was incomprehensible. We tried making fewer categories "Very Productive" (aka "Very Intentinal") on Nathan's data, but the data still appeared quite messy.

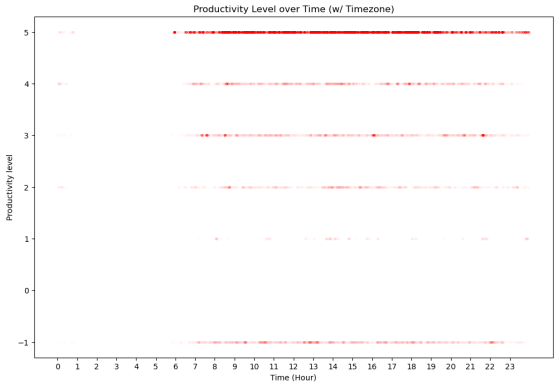

Figure E.03. Plot productivity levels after applying a timezone. Opacity = 0.005 (#03)

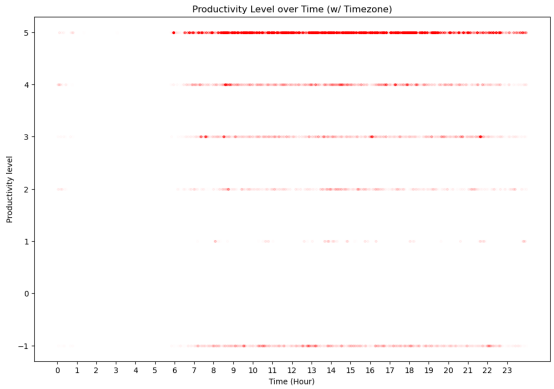

Figure E.04. After making fewer categories "Very Productive". Opacity = 0.005 (#04)

Because there was so much data, we thought about only analyzing a week's worth, e.g. Sept 26 - 30, 2022. (See Figure E.05)

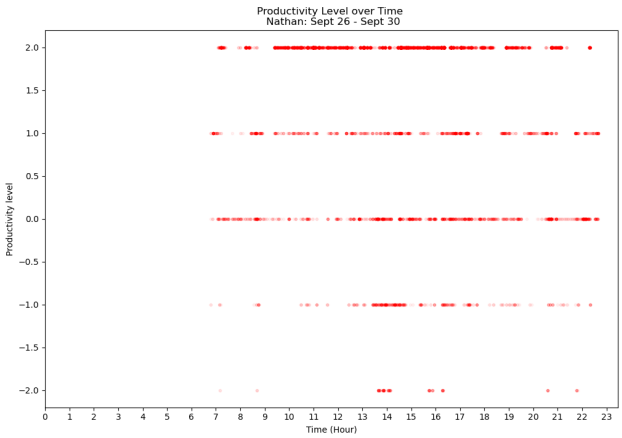

Figure E.05. Looking at only a week's worth of data. ***Opacity = 0.05*** *(#05)*

Still no luck. It seemed like scatterplots were not the best way to analyze the data.

D.2 Second Approach: Only Distracting Data, Bin Visits into Half Hours

Based on the results from the scatterplots, we realized it might be more important to just look at the distracting data (e.g. productivity levels of -1 and -2).

We visualized the data by grouping all the data points by the half-hour (e.g. round to the nearest half hour[16]), then summing the scores of each half-hour.

And the results were fascinating.

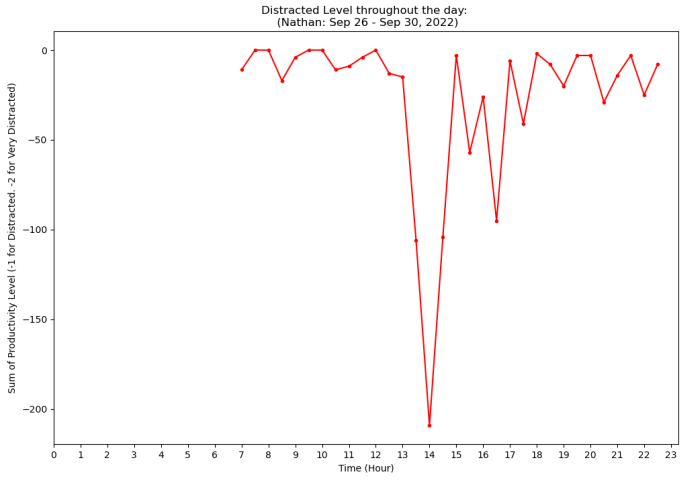

Figure E.06. Looking at only a week's worth of data, with the opacity of the dots = 0.05 (#06)

[16] We used: only_distracted['half_hour'] = (only_distracted['time_of_day'] /(30 * 60)).round().astype(np.int32)

Before the experiment, Nathan had tracked his energy levels and found empirically that he would naturally become distracted at 12:30 pm[17].

Now, he had some data to show for it. There was only one problem: the valley was at 2 pm.

We inspected other weeks, but there appeared to be too much noise.

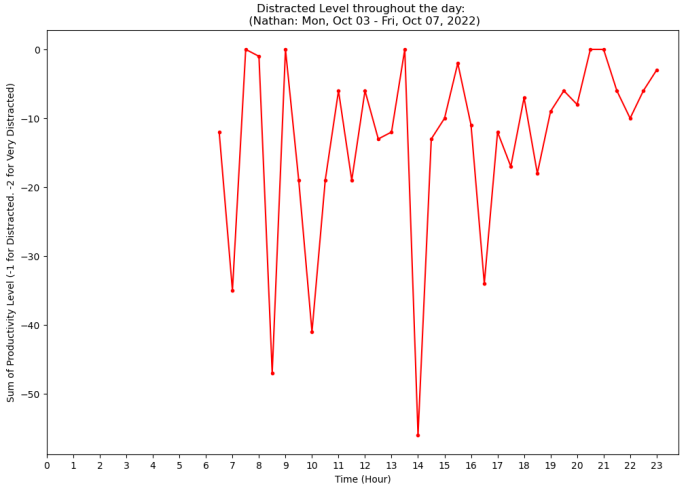

Figure E.07. Distracted data for the Oct 3-7, 2022 (#7)

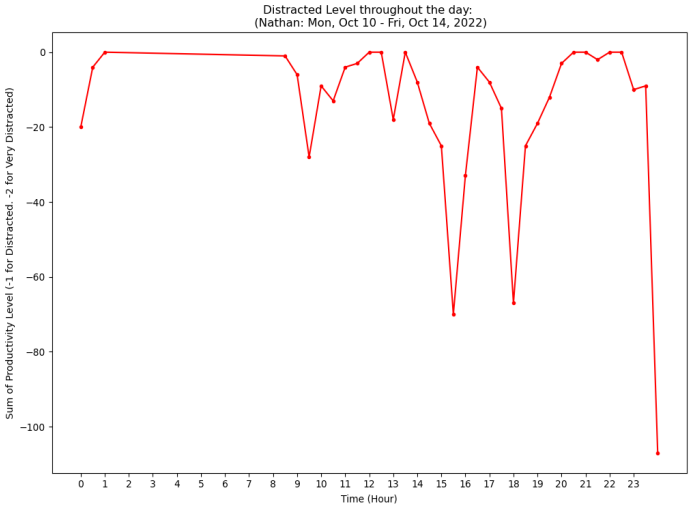

Figure E.08. Distracted data for the Oct 10-14, 2022[18] (#08)

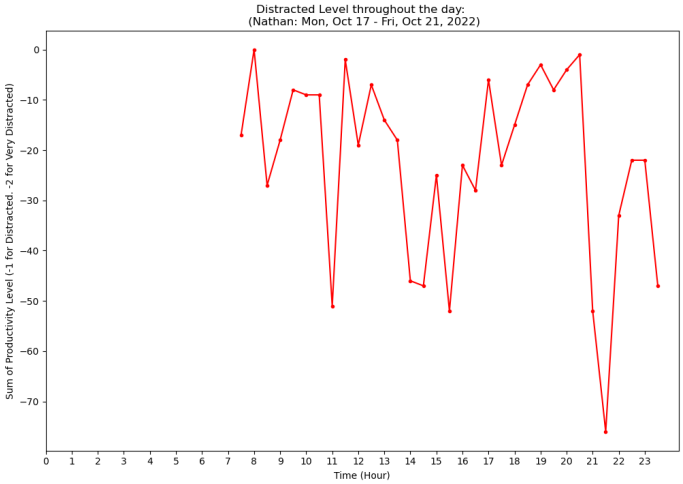

Figure E.09. Distracted data for the Oct 17-21, 2022 (#09)

Juan's data

[17] Nathan Tsai: One time I caught myself becoming distracted checked my browsing history for when I openned the first distracting website. It was at 12:31 pm!
[18] Nathan Tsai: Apparently, I stayed up late a couple of nights until ~2 am on distracting websites

At this point, we also uploaded Juan's data and began analyzing it along Nathan's data.

Juan's data is also quite noisy, but it appears the distraction peaks around 8 pm (with a more subtle peak around 4 pm - 4:30 pm).

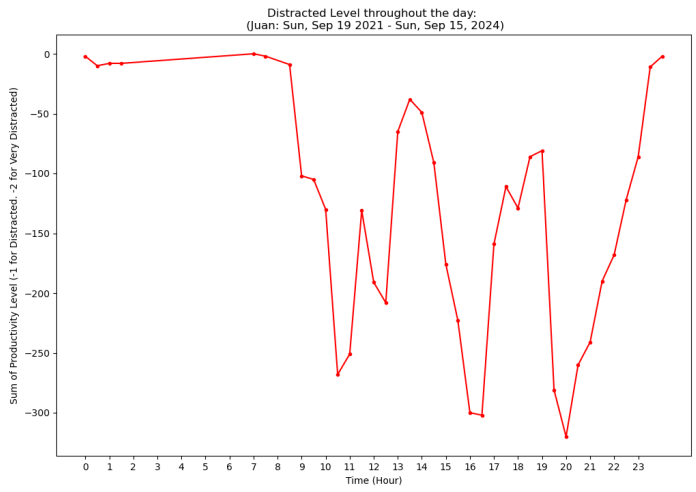

Figure E.10. Juan's Data for the entire length all the history (e.g. Sept 2021 - Present) (#19)

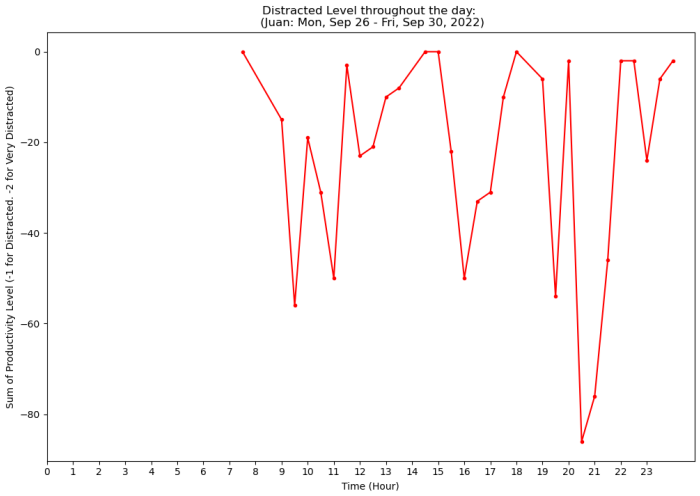

Figure E.11. Juan's Data for a week; Sept 26-30, 2022 (#15)

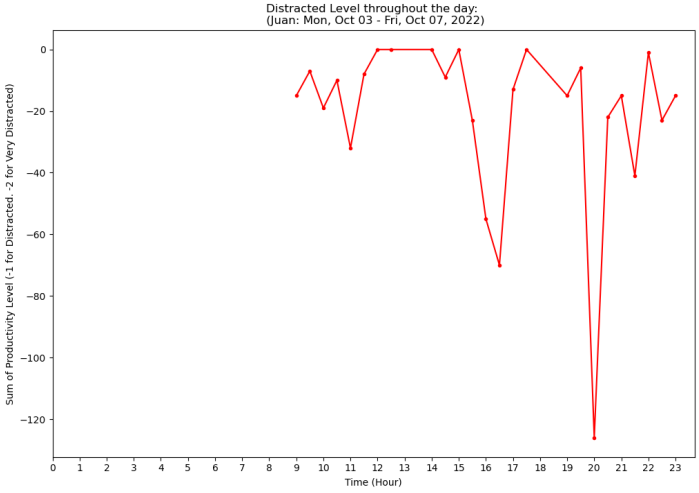

Figure E.12. Juan's Data for a week; Oct 3-7, 2022 (#16)

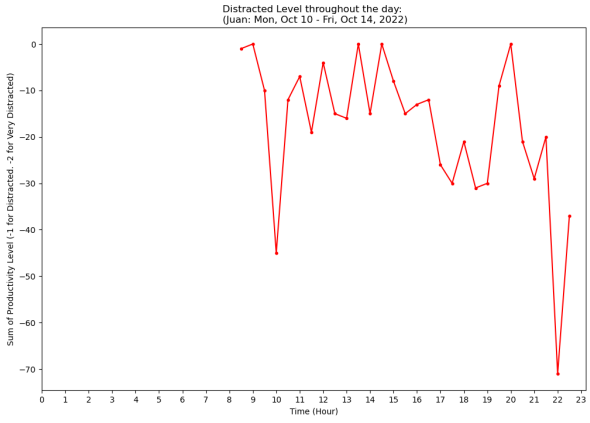

Figure E.13. Juan's Data for a week; Oct 10-14, 2022 (#17)

Weekends

We also analyzed the weekend data, but quickly realized that Nathan and Juan did not use their computers very often on the weekends.

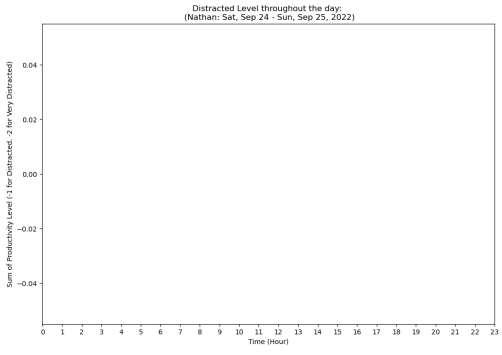

Figure E.14. Nathan's Weekend Data: Sept 24-25, 2022 (#10)

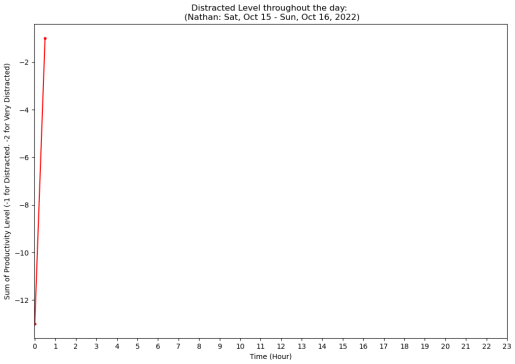

Figure E.15. Nathan's Weekend Data: Sept 24-25, 2022 (#11)

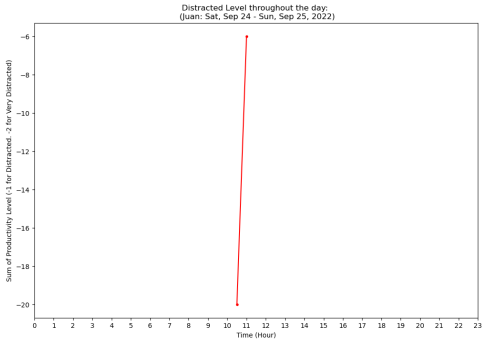

Figure E.16. Juan's Data for a weekend; Sept 24-25, 2022 (#14)

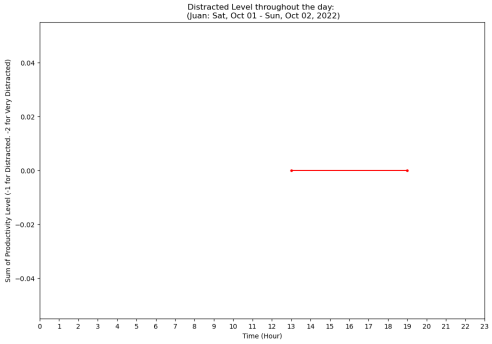

Figure E.17. Juan's Data for a weekend; Oct 1-2, 2022 (#13)

Let's look for all-time

After analyzing the weekly data, we figured there was too much noise. We need to explain our scope a little bit to average out the noise. So, we looked aggregating all of the data, but this interval seemed too wide, removing the trough at 2 pm for example.

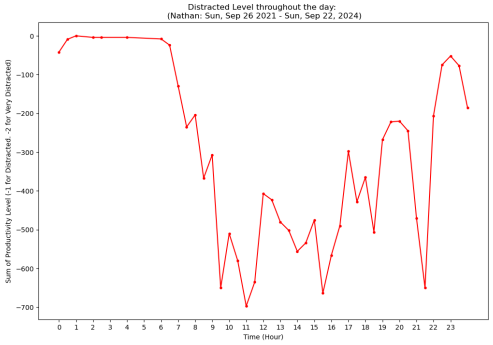

Figure E.18. Nathan's Distracted data for all time, Sept 2021 - Present (#12)

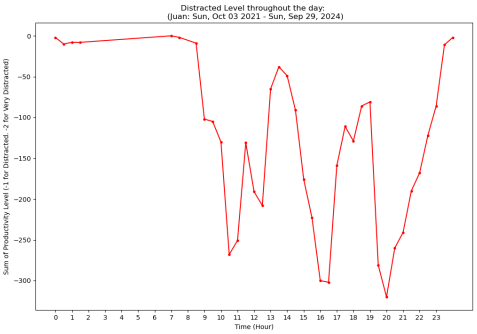

Figure E.19. Nathan's Distracted data for all time, Sept 2021 - Present (#18)

Viewing Distracted Sums by Month

Since the week was too small and all time was too large, we tried viewing the data by month to smooth out some of the noise.

When viewing Nathan's data for a month, we were able to see the peak arise again (see Figure E.20).

And we could also see three peaks for Juan's data as well (see Figure E.21).

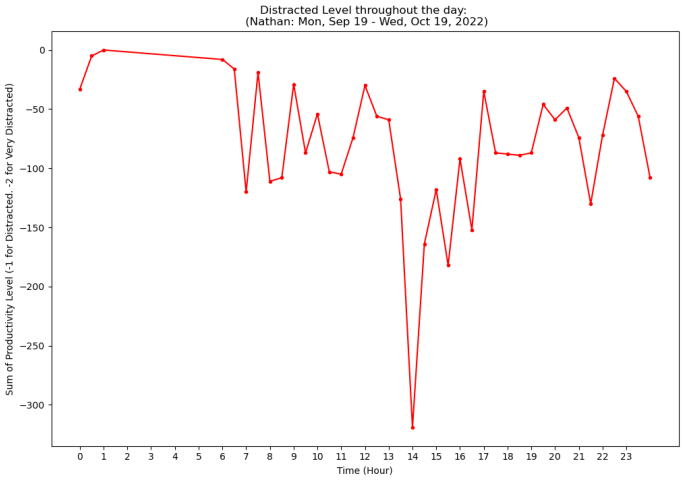

Figure E.20. Nathan's distracted data from Sept 19 - Oct 19 (#20)

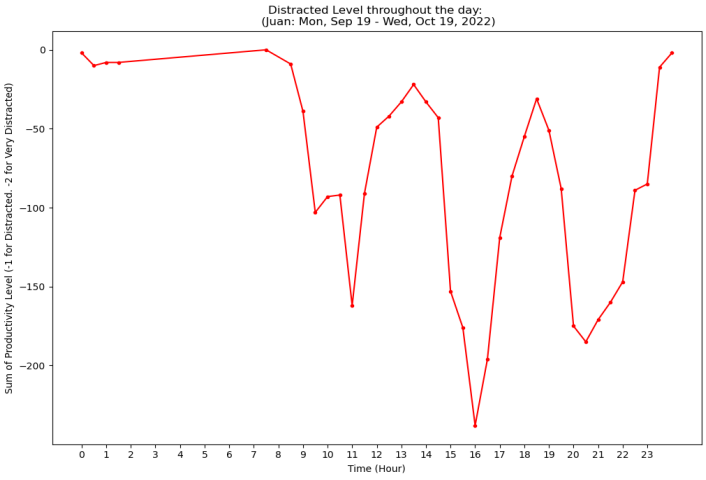

Figure E.21. Juan's distracted data from Sept 19 - Oct 19 (#21)

Using Safari Data

At this point, we started using Sanyam's data (who primarily used Safari) and analyzed his data by month.

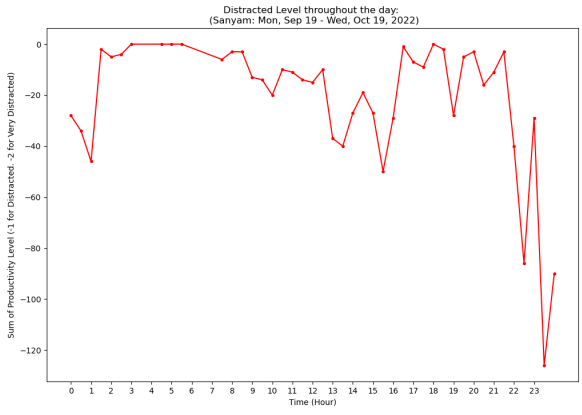

Figure E.22. Data from Safari, using the appropriate Safari database fields (#23)

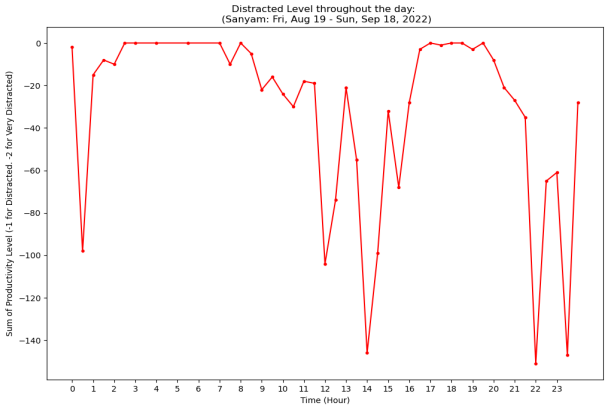

Figure E.23. Sanyam's Data from Safari again, showing the productivity better (#24)

Filtered Data

After analyzing the Chrome history database more, we realized that many of the rows had visit duration zero.

And after visualizing the data with the removed visit durations, the plots followed roughly the same shape. Thus, we concluded the visits < 5 microseconds were noisy and removed them.

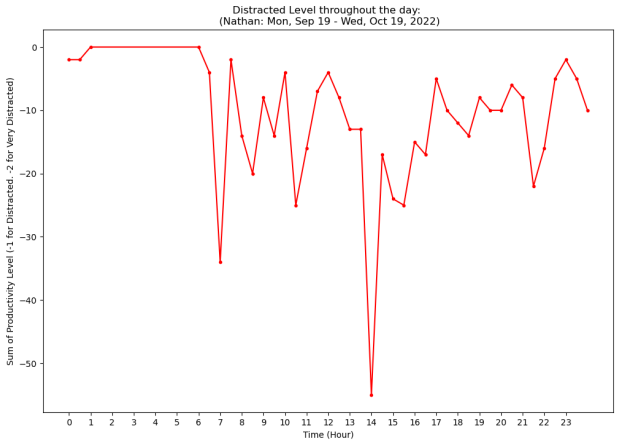

Figure E.24. Nathan's Data, filtering out visitings with duration <= 5 microseconds. (#25)

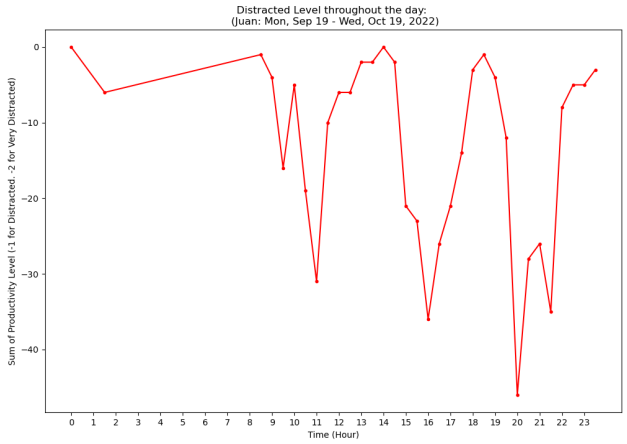

Figure E.25. Juan's Data, filtering out visitings with duration <= 5 microseconds. (#26)

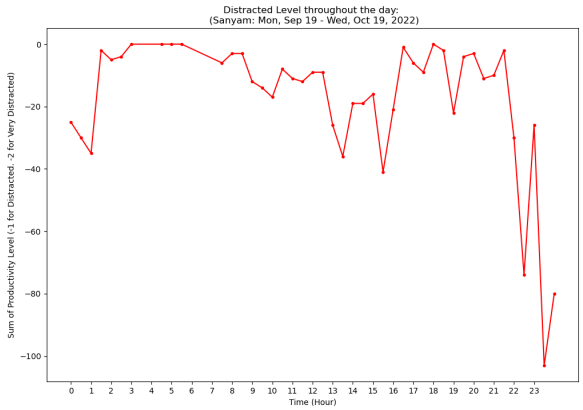

Figure E.26. Sanyam's Data, filtering out visits with score <= 5. (#27)

Include Intentional (aka Productive) Data Again

We then figured that we should not neglect the productive data.

So, we began analyzing the data again (with the intentional data), but divided the intentional scores by the total count of intentional visits, and the same with the distracting scores.

This normalized the values a bit, allowing us to have some preliminary results.

Note: the positive and negative values on the y-axis in the graphs below.

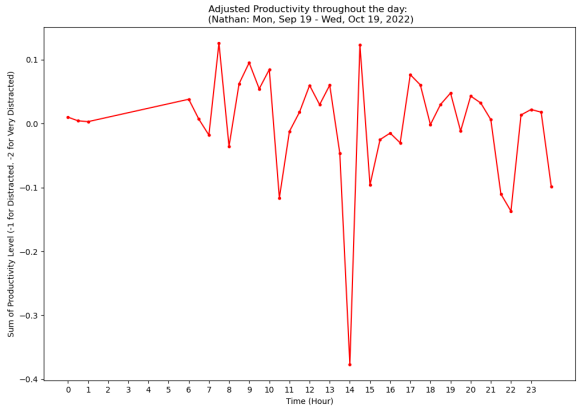

Figure E.27. Nathan's data; Include productive data & divide each score by the total count in that category (#28)

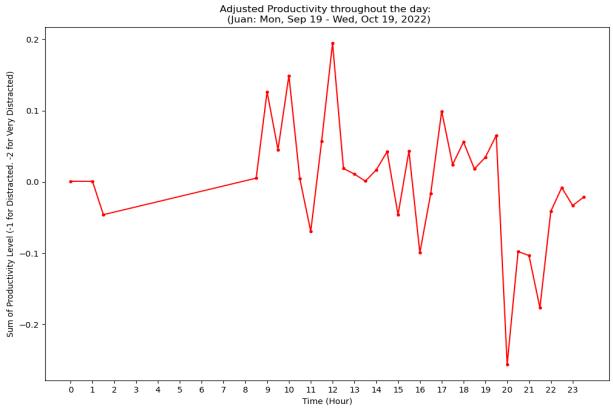

Figure E.28. Same as the left, but for Juan (#29)

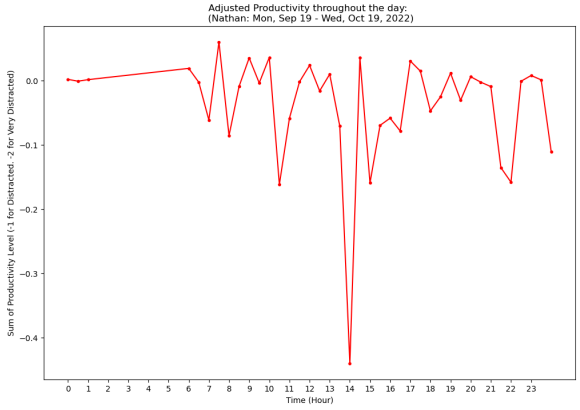

Figure E.29. Same as above (#28), but reduce the productive scores by 50% (#30)

Accounting for Visit Duration

We also considered that we should also look at the visit duraction.

We started by multiply the productivity score by the visit duration (see Figure E.30), but this quickly became problematic, because some visits were more than 24 hours long. Even after removing all visits longer than 8 hours, we realized that we were doing something wrong.

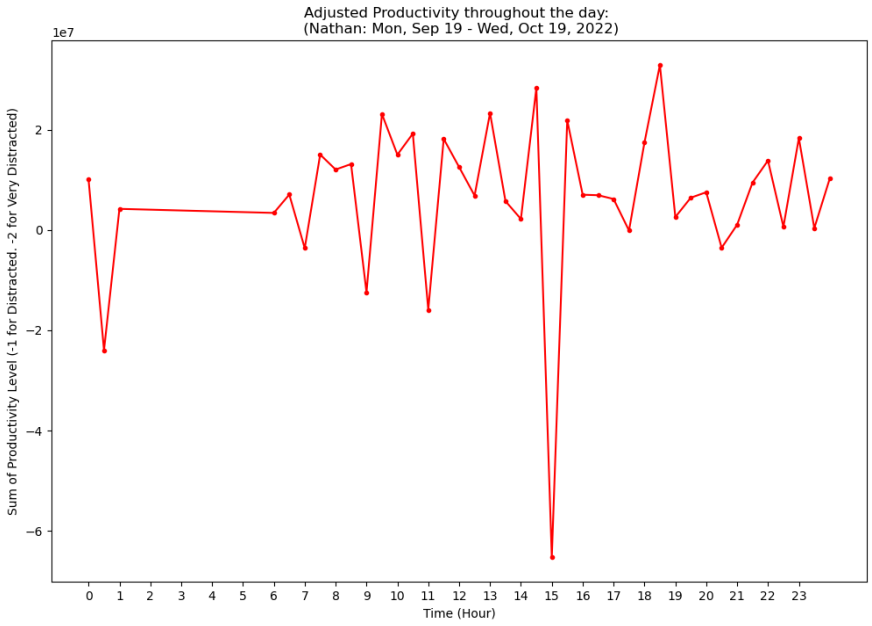

Figure E.30. Multiply the productivity score by the visit duration. (#31)

If I open a distracting website at 2 pm for an hour, I should have two distracted entries: one at 2 pm and one at 2:30 pm. I shouldn't magnify the distracted data at 2 pm by two times.

Note: below, accounting for visit duration means having `floor(visit_duration / 30 min)` entries per row in the visits database table, (e.g. have two rows if 30 min < visit_duration < 60 mins).

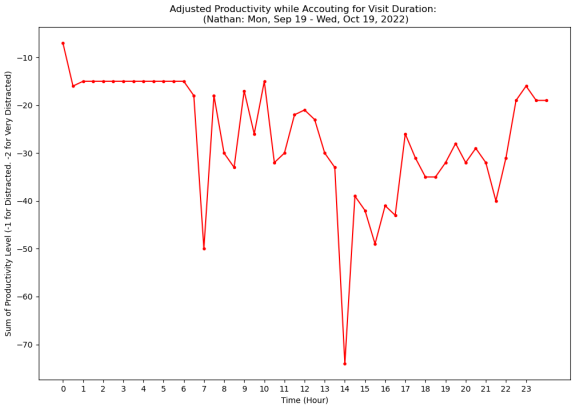

Figure E.31. Nathan's data (with accounting for visit duration) (#32)

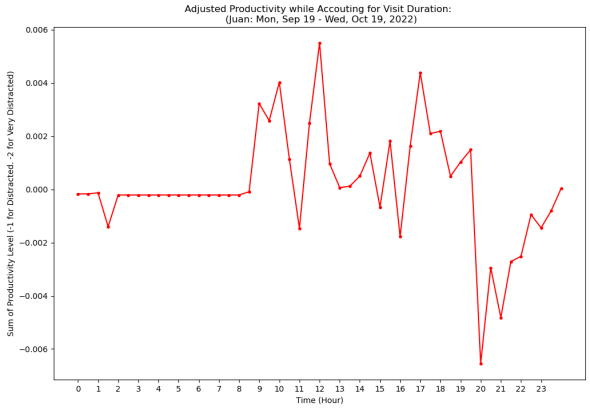

Figure E.32. Juan's Data (with accounting for visit duration) (#33)

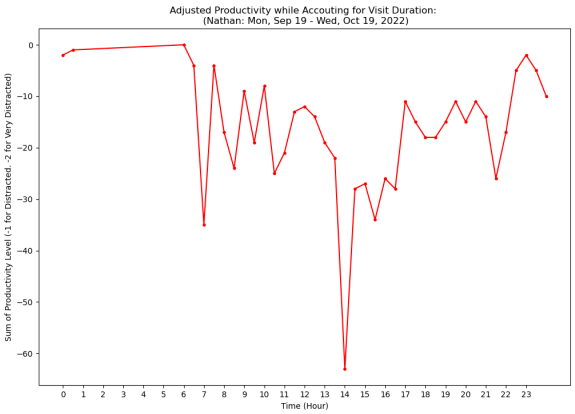

Figure E.33. View Nathan's only distracted data (with accounting for visit duration) (#34)

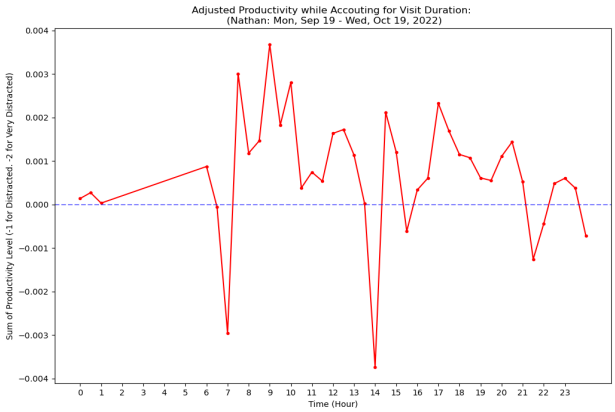

Figure E.34. View all data (with accounting for visit duration) (#35)

D.3 Conclusions

After reaching this point, we came up with normalized productivity (see *3.5 Adjusted & Normalized Productivity*) and using seaborn's **.lineplot()** instead of using **.sum()** to aggregate the half hour bins, producting the plots in *5. Conclusions*.