# Screen Time and Yoga

Christophe Al Haddad

## 1. Motivation

I am passionate about technology, how it changes the world, and how we interact with it. Some claim that technology is negatively impacting our lives and replacing us; I disagree because its true purpose is to push humanity forward. It is the way we use technology that dictates its effects on us. In particular, in the context of smartphone usage, the amount of time spent on our phones or screen time is an indicator of our diverse interactions with technology. A high daily screen time, an indicator of smartphone addiction, may hinder social interactions and induce fatigue; on the other hand, if properly balanced, smartphone usage would smoothly complement our daily lives. Recently, the major smartphone companies implemented screen time in their mobile OS to help users control their smartphone usage: Apple introduced Screen Time (iOS) and Google launched Digital Wellbeing (Android). Through this project, I showcase how we can improve our interactions with technology through visualizations of smartphone screen time evolutions.

## 2. Data Preprocessing

The Screen Time data shows the smartphone usage (in minutes) of an individual throughout the days spanning 4 weeks. The usage is divided into 8 categories characterizing the different app categories, including the Total Screen Time. Interestingly, the user started doing yoga (meditation, breath control) at the end of the second week, indicated in a binary column (*yoga*); starting this discipline changed the smartphone usage.

Data preprocessing was a crucial part for the visualization, since it allowed to showcase different viewpoints of the dataset:

First, since I wanted to encode the category using color hue (explained later in 3. Design Decisions), the number of categories (8) was greater than the recommended limit one for selection and association (7). Also, the *creativity* category had mostly zero values (with the exception of only 4 positive ones). Therefore, I decided to merge *creativity* with *entertainment* since both are fairly similar and that would lead to 7 categories.

The evolution of screen time from the initial weeks to the yoga weeks is the most important dataset feature to showcase. For this reason, I divided the data into two structurally identical groups, the *before* and *after* groups, containing the rows where *yoga* is 0 and 1, respectively. Each group was represented as a dictionary storing:

- dates: a list of the dates (mm/dd/yyyy)
- week_dates: a list of the weekdays (Monday, Tuesday…)
- rawCategories: a dictionary mapping a category to the list of its corresponding screen time values throughout the days
- avgCategories: a dictionary mapping a category to the list of its corresponding average screen time value for a weekday (average Monday values, average Tuesday values…). I wanted to analyze the screen times for each weekday, instead of just averaging them for each category, since it might be insightful to see the screen times throughout the weekdays

- avgCategoriesTotal: a dictionary mapping a category to its corresponding overall average screen time value

Moreover, I create a *dif* dictionary where I calculated for each category the percentage change (positive or negative) of the total average screen time. It was very useful for clearly showing the impact of yoga on each screen time category.

# 3. Design Decisions

The visualization is divided into three parts: month view, detailed view, and insights.

## Month View (Tiles)

The Month View is the main visualization of the screen times. It is a handcrafted temporal visualization inspired from a calendar-based visualization: tile maps. Using colored tiles, arranged like a calendar, the screen time evolution for each category is shown (Figure 1).

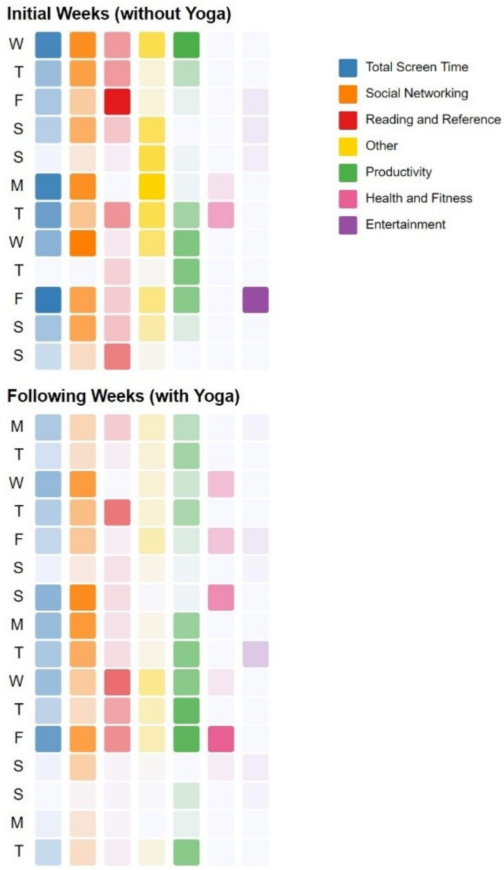


**Figure 1: Month View**

As the dataset is divided into *before* and *after*, the tiles are grouped according to their group (before or after yoga). All the dimensions in the project (alignment, spacing) are based on the dimension of tiles (tileDim). I encoded three data attributes in this visualization: app category (categorical), screen time in minutes (ordered), and date or time (ordered).

For encoding the category, I relied on color hue, the 2$^{nd}$ most effective identity channel. I chose the color hue channel for its selectivity (stand out) and mostly for its associativity to group the categories. As mentioned in 2. Data Preprocessing, I had to merge 2 categories to obtain an optimal discriminability of 7; the initial length of 8 would have been practically cumbersome for the eye. I created a custom color palette inspired from [ColorBrewer: Color Advice for Maps (colorbrewer2.org)](colorbrewer2.org) with 7 qualitative data classes. I replaced the yellow color (which was too bright) with *cyber yellow* (#FFD300) which is much more appropriate.

Then, I used the color saturation of each tile to encode the screen time value. Although it is not the best channel for ordered attributes, it is the most convenient for this current visualization. The scale ranges from *ghostwhite* to the category color; I explicitly chose *ghostwhite* instead of *white* for a better look and for avoiding blank tiles. Also, it is a single scale having the values of both week groups as extent. Initially, when the page is loaded, the tiles are all *ghostwhite*. Then, a transition with increasing delays successively colors each tile row; again, this is to incite the viewer to look at the evolution of the screen times throughout the days.

Furthermore, the time evolution was encoded by using two of the Gestalt Laws of Perceptual Organization: the laws of similarity and proximity. The former, stating that similar objects are perceived as groups, was implicitly applied with the unique color of each category. The latter more powerful one, stating that objects close to each other are perceived as groups, contributed the most to the encoding of time. I specifically decreased the vertical margin between tiles (those in the same category are closer to each other) compared to the horizontal margin between tiles of the same row. So, since tiles are vertically closer, they would be more perceived as a group and the viewer would clearly notice the evolution as they look from top to bottom.

Other subtle details include rounding the tile corners; they look nicer and the shape reminds us of application icons. Also, I included the first letter of the weekday (M, T, W, etc.) next to each tile row with the text-anchor as "end", inspired from the calendar-based tile map. And a tooltip was added to show the screen time value and the specific calendar date of a tile.

Finally, when the mouse hovers on a tile, it is smoothly shrunk by 20% to indicate its focus and to subtly incite clicking. In fact, a mouse click triggers a transition to the detailed view; the transition is used only to change views, as a narrative, not just to animate.

## Detailed View

This view provides more analysis on the screen times. It consists of two graphs: a histogram (Figure 2) and a horizontal bar chart (Figure 3).

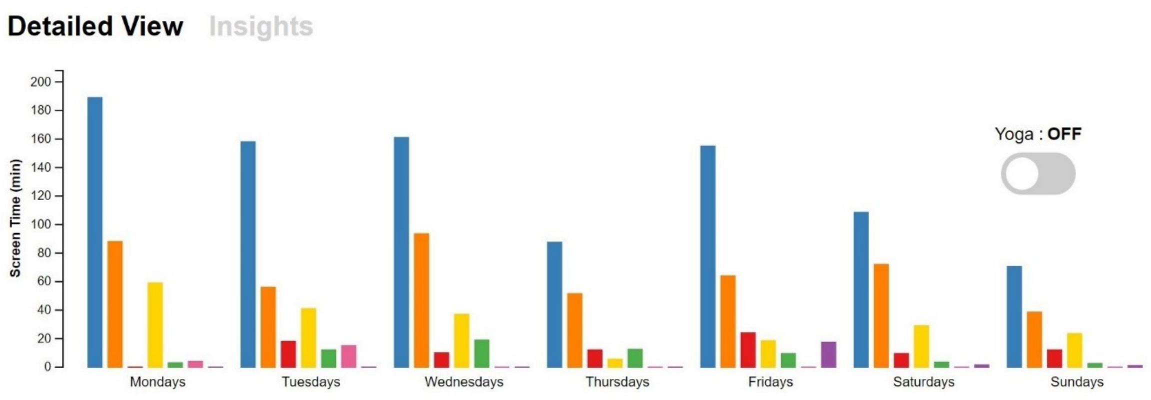


**Figure 2: Histogram**

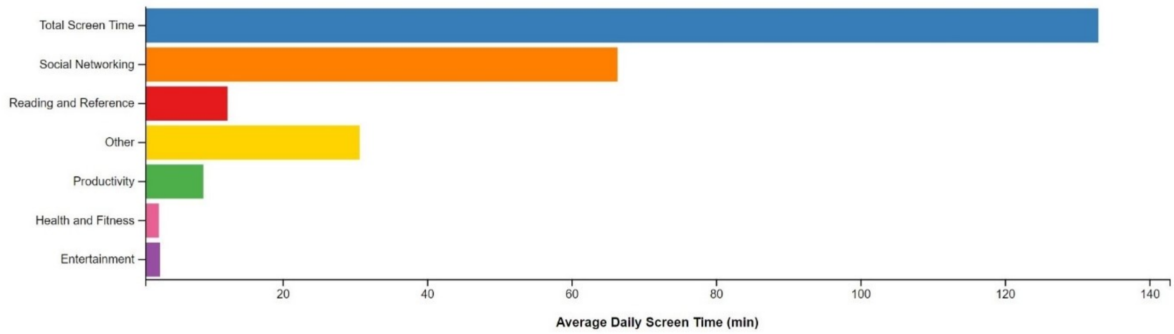


**Figure 3: Bar Chart**

The histogram shows the data of the avgCategories dictionary; the x-axis is the weekdays with their categories. Again, I used the law of proximity by making bars of a same day closer to each other and increasing the horizontal margin between each weekday group. Also, I added an “s” at the end of each weekday (Monday”s”, Tuesday”s”, etc.) to subtly imply that those values are averages over all the Mondays or Tuesdays, etc. Also, I added tooltips when hovering the mouse for viewing precise values.

As for the bar chart, it visualizes the data of the avgCategoriesTotal dictionary: the screen time of each category averaged over the days of the corresponding group (*before* or *after*).

For both graphs, I adjusted the minimum size of a bar to avoid getting tiny bars (or even imperceptible ones) when the screen time value is small. I made sure the bar lengths still respect their relative values: *value = value < 20 ? 5 + value*(3.0/5) : value*. In addition, when creating the scales for the graphs, I extended the domain of the axis values to add a small margin to the axis; otherwise, the bar with maximal length would coincide with the axis endpoint.

Most importantly, for interactively showing the evolution of screen time, I added a button that switches yoga ON/OFF for both graphs. It triggers a smooth transition to the values of the other group (*before* or *after* yoga). Surely, I kept the same axis scale when transitioning the group, allowing for better comparison and perspective. In fact, the position on a common scale is the best magnitude channel for encoding ordered attributes (here the screen time).

For adding interactivity, I added a selection feature: clicking on any bar (histogram or bar chart) adds the selected bar’s category to a selection list. The bars related to the selected categories are still colored in both graphs, but the non-selected ones become out of focus and are colored in a very light shade of grey (#F0F0F0). This allows the user to focus on specific categories for inspecting their change throughout time.

## Insights View

Clicking on the *Insights* title transitions to the insights view (Figure 4), containing both the differential graph and some insight texts.

The differential graph accurately shows the % change in screen time (both positive and negative) for each category, using the *dif* dictionary. It gave an interesting perspective to the visualization: it turns out an average daily increase of 3 min of *health and fitness* screen time corresponds to a +110.5% change. I also used the same principles as the Detailed View graphs (axis scale, selection feature, minimum bar length, etc.).

The insight texts are colored according to the category represented.

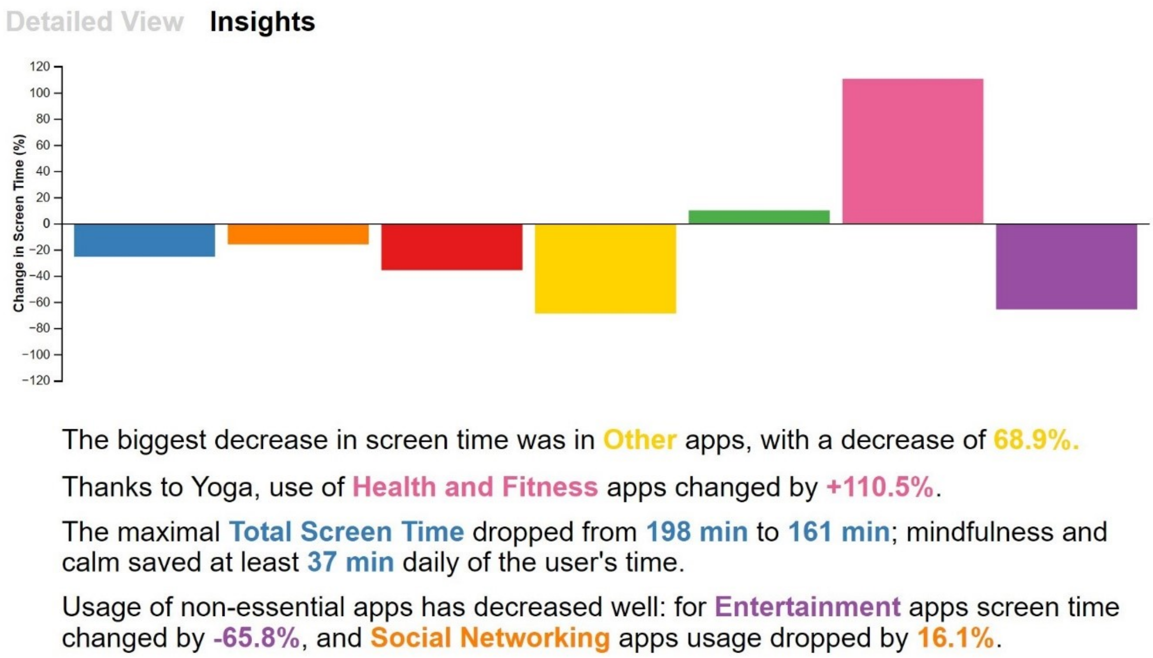


**Figure 4: Insights View**

# 4. Insights from Data

Here are additional insights of the impact of yoga on screen time and smartphone usage for the individual not mentioned in Figure 4:

- The *health and fitness* screen time (after yoga) is noticeable on Monday, Wednesday, Friday, Sunday, indicating that the days in between are rest days
- *Productivity* screen time values are apparent mostly in weekdays, which is logical since those apps are used for work
- The minimal average *total* screen time dropped from 87.5 min on Thursdays to 58 min on Saturdays

# 5. Implementation: Challenges and Achievements

I used JavaScript and d3 to code this visualization. Every element was hand-made; for instance, the bars in graphs are custom rectangles, to add more control and flexibility on the design (like adding the selection feature).

Challenges faced:

- When switching views, I could still click on a hidden element (bar or button); solved this by changing the visibility attribute
- References copies often caused bugs; fixed with occasional deep copies

Achievements:

- ✓ All the design parameters are dependent on hyperparameters (tileDim, spacing)
- ✓ Keyboard shortcuts: *R* resets selection, *T* transitions views, *Y* switches the yoga button
- ✓ Created my own wrap function for fitting blocks of texts (automatically adds new lines, support for styles)
- ✓ Added transition guards (several ctx variables to avoid mixing transitions)

# 6. Appendix: Additional Screenshots

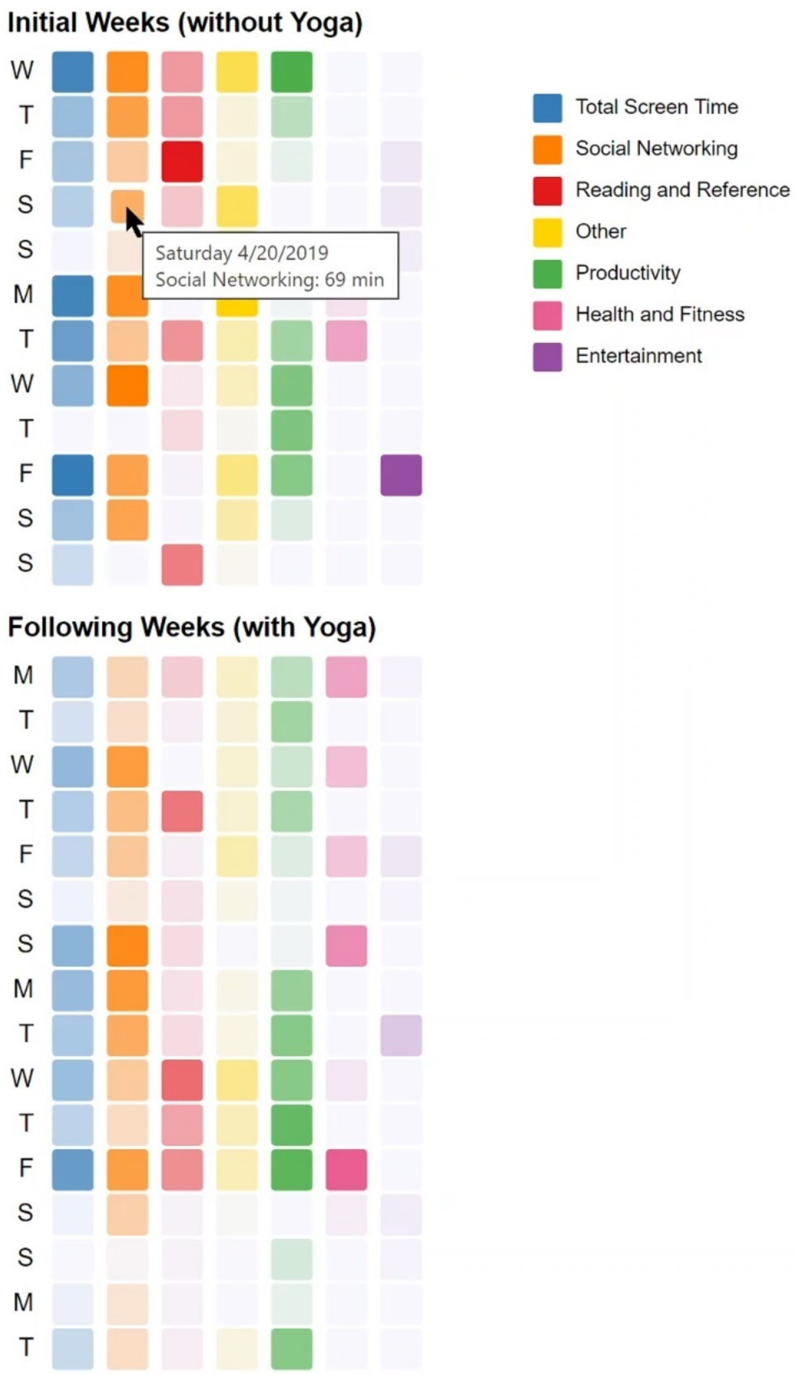


**Figure 5: Month View with Selected Tile**

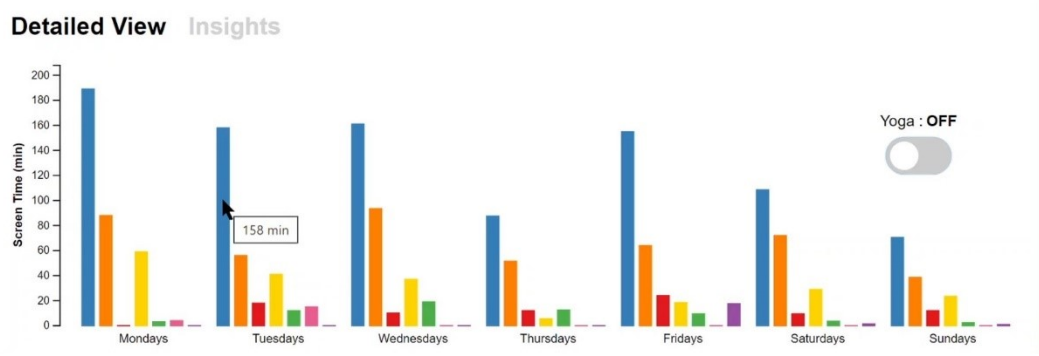


**Figure 6: Histogram Tooltip**

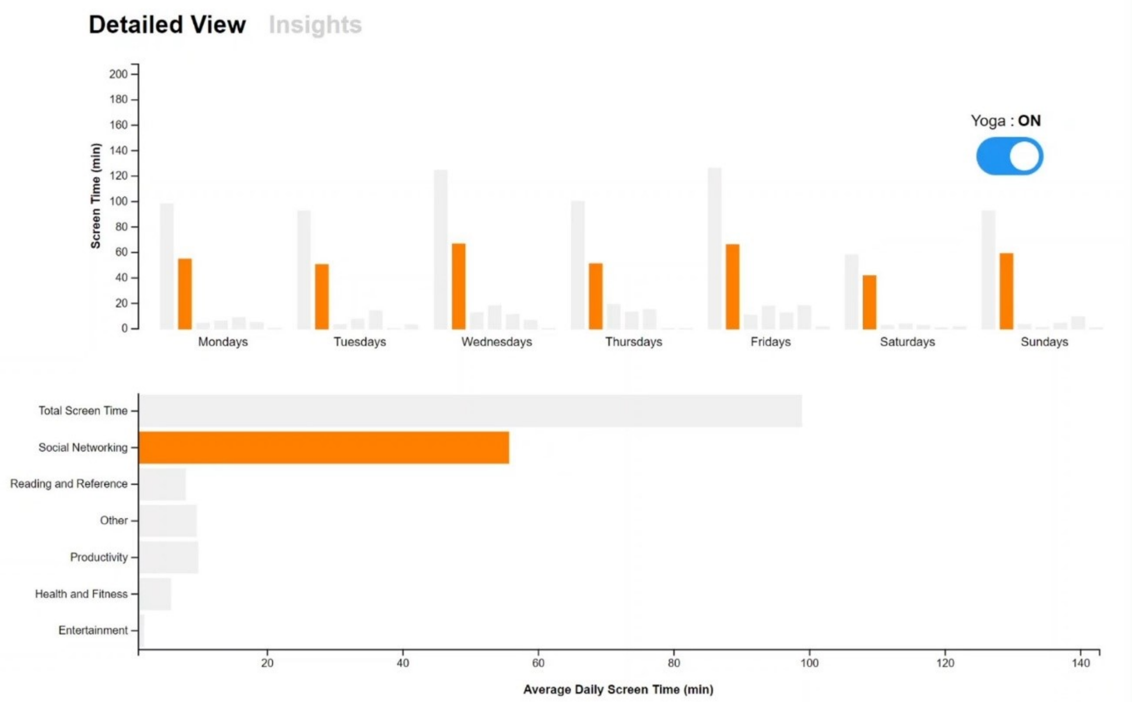


**Figure 7: Selected Category and Focus**

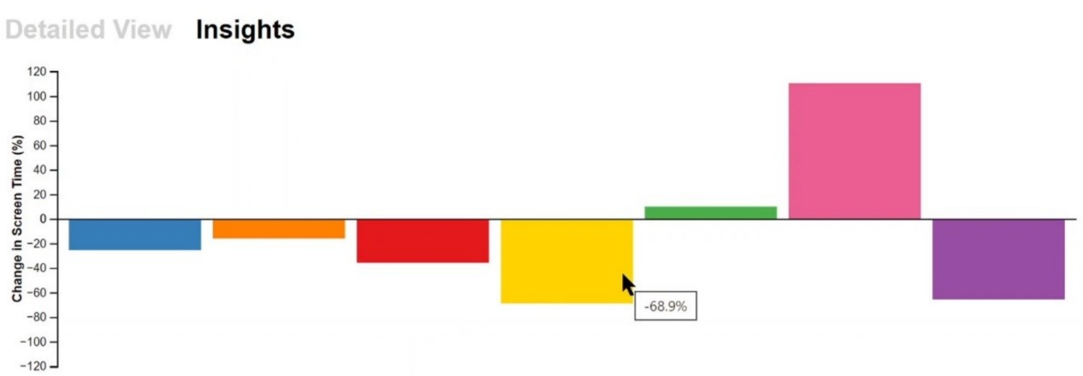


**Figure 8: Insights Tooltip**