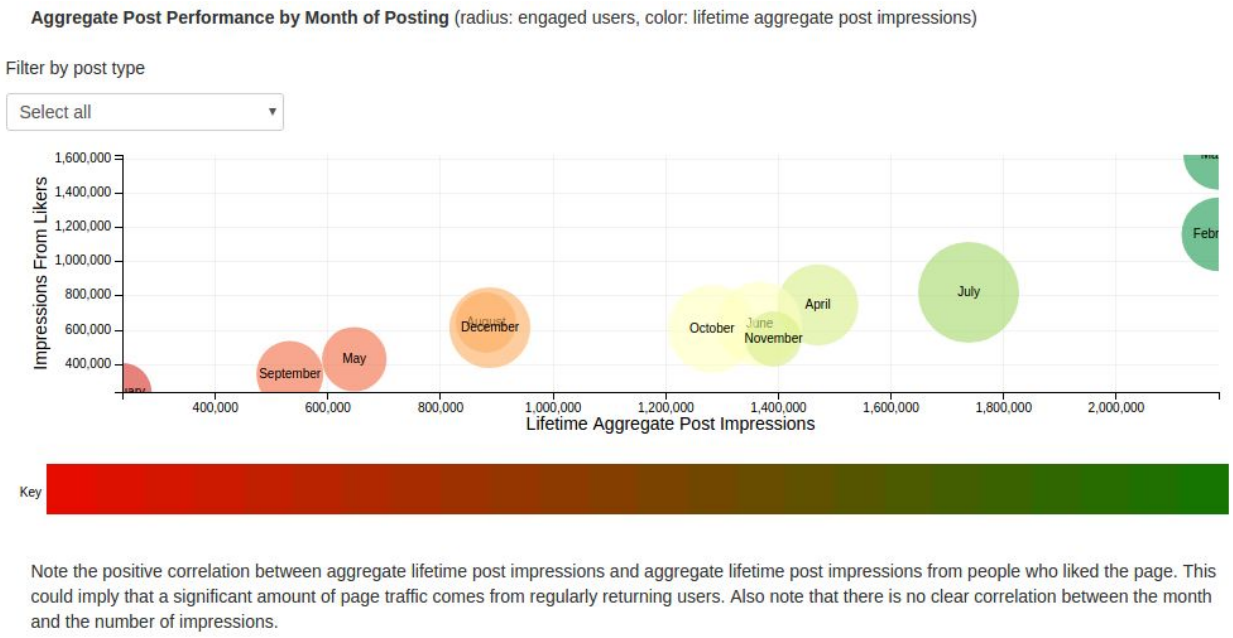
Colin Stern
29622919



Note the positive correlation between aggregate lifetime post impressions and aggregate lifetime post impressions from people who liked the page. This could imply that a significant amount of page traffic comes from regularly returning users. Also note that there is no clear correlation between the month and the number of impressions.

Wondering which month is most likely to see the creation of a popular post, I made this chart to plot relationships between which posts are seen the most, the month they are posted in, and how many users become engaged with the post. With the visualization in hand, it became clear that the month does not clearly correlate with the popularity of the post. However, the amount of impressions correlates with the amount of impressions from page likes, suggesting that most posts are seen by returning users.

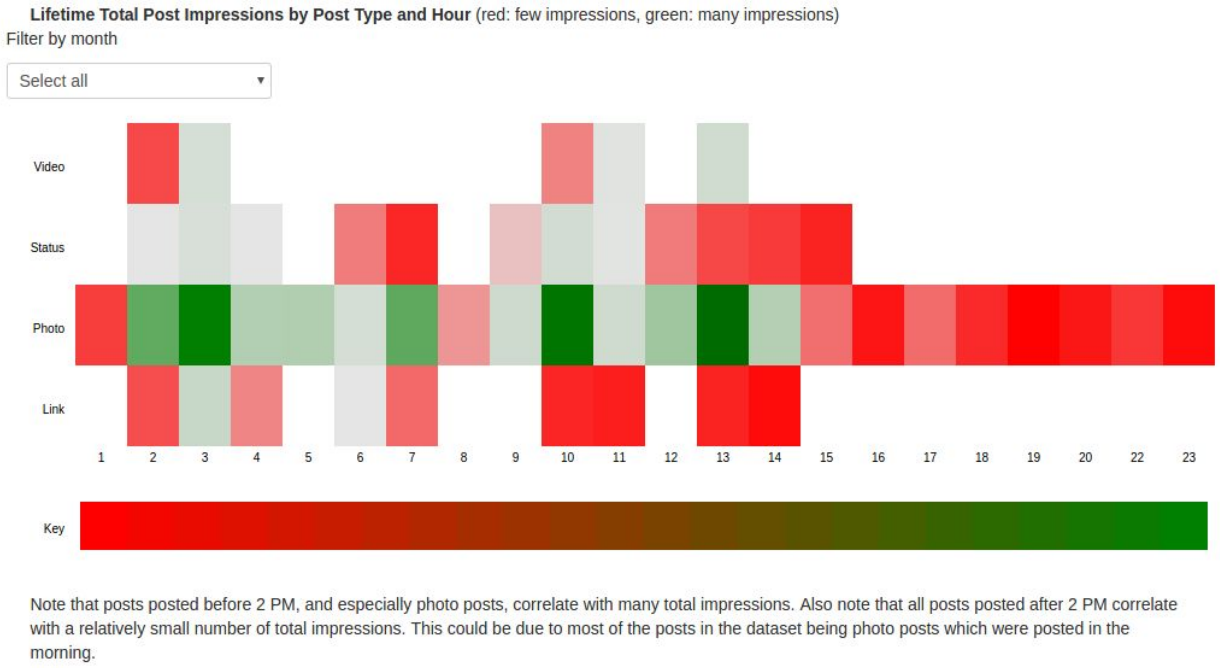
To simplify calculations, I used a total instead of an average so I would not need to account for very large outliers caused by a single popular post being compared with a group of many average posts, which would appear very small. The drawback of this decision is that there is no normalization, which makes the visualization more susceptible to biases in the dataset. In the next project I will account for this by normalizing the data and instituting a non-linear scale if necessary.

To aid the user, I redundantly encoded the x-axis variable, lifetime aggregate post impressions, with color. This makes trends easier to spot. A drawback of my color choice is that red-green colorblind people may have difficulty perceiving differences in the color scale. Luckily, by redundantly encoding the x-axis variable, a color-blind user can still understand the visualization.

Additionally, a legend helps map color to the value it represents. By hovering their mouse over a part of the legend, the value represented by that color is displayed in a popup box.

By hovering their mouse over a bubble, a user is shown a box listing the value of each variable encoded in that circle. This is my implementation of probing. By clicking on a bubble, that bubble is selected and corresponding linked visualizations update. This is my implementation of selection.

To give the user more options in selection, a dropdown allows them to filter by post type.



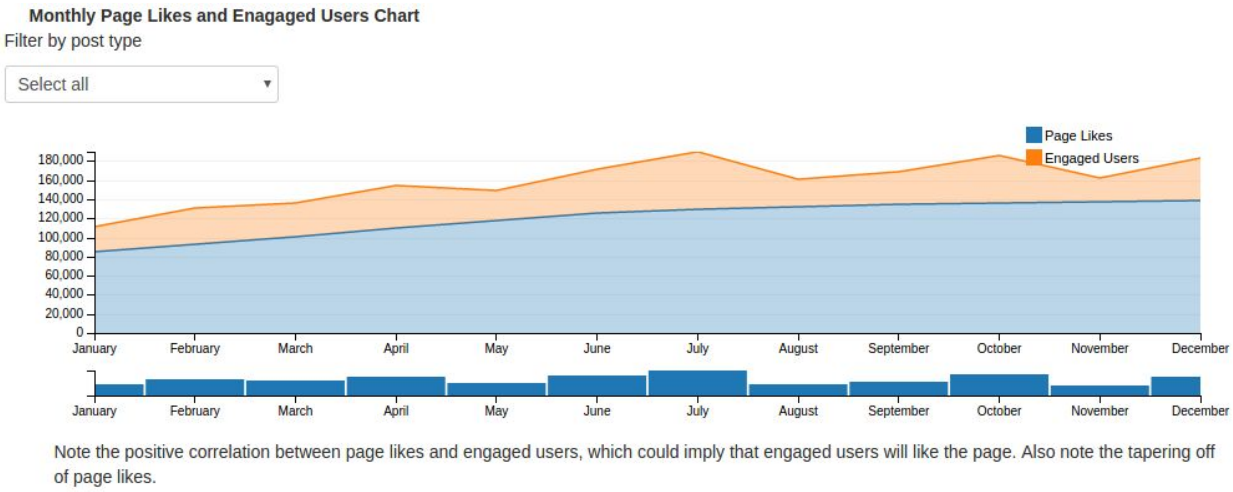
Note that posts posted before 2 PM, and especially photo posts, correlate with many total impressions. Also note that all posts posted after 2 PM correlate with a relatively small number of total impressions. This could be due to most of the posts in the dataset being photo posts which were posted in the morning.

Curious to see when popular posts are born, I made this heatmap to explore which posts get popular and what time of day they are posted. It is clear that photo posts, and especially the ones posted in the morning and early afternoon, take the cake in total impressions, However, this could be due to the dataset being biased by including more photo posts than any other type. It could also be due to most posts being made in the morning, when the employees of the company that run this cosmetics page are at work.

To simplify calculations, I used a total instead of an average so I would not need to account for very large outliers caused by a single popular post being compared with a group of many average posts, which fall off the color scale. The drawback of this decision is that there is no normalization, which makes the visualization more susceptible to biases in the dataset. In the next project I will account for this by normalizing the data and instituting a non-linear scale if necessary.

I added a key which aids the user in determining relative values of the heatmap. The user can hover their mouse over a color box to determine what value that color corresponds to. While the legend renders darker colors toward the middle rather than lighter colors, as I had intended, this can be fixed with a different color scale in the next project.

Hovering your mouse over a box will display the hour, type, and total number of impressions represented by that box; this implements probing. Clicking on a box highlights the records represented by that box in all of the other visualizations; this is selection. A dropdown allows the user to filter values by month.



Note the positive correlation between page likes and engaged users, which could imply that engaged users will like the page. Also note the tapering off of page likes.

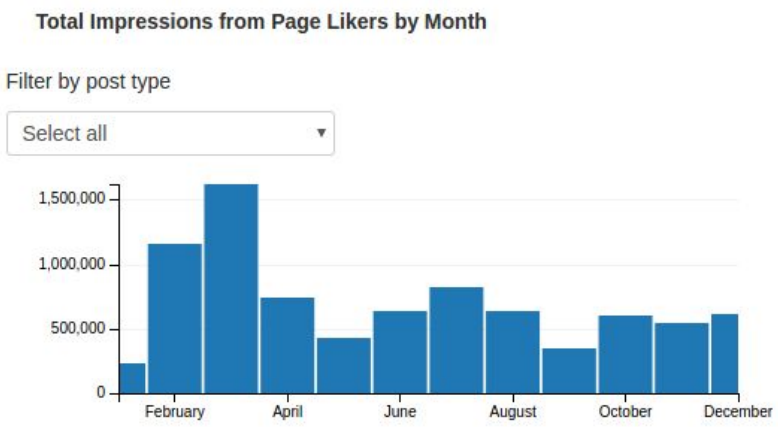
How did the page grow in popularity over time? To answer this perplexing question, I made a stacked line chart which displays the total page likes along with the number of engaged users in that month. It became clear that while the page collected more likes over time, growth tapered toward the end of the year as the amount of engaged users stopped growing.

To compute the amount of page likes per month, I averaged the total number of page likes associated with each record of the dataset over each month. This binning was necessary because each record did not have a date associated with it.

I added a legend which helps the user identify which line means which. The colors are chosen to minimize the effect of color blindness - orange and blue are sufficiently separable by most sufferers of color blindness.

Hovering your mouse over either line will display the month and value associated with that data point; this is probing. Using the selector below the chart, we can zoom in on selections of data and watch them change in the linked visualizations; this is selection.

A dropdown allows the user to filter by type of post, further enabling data exploration.

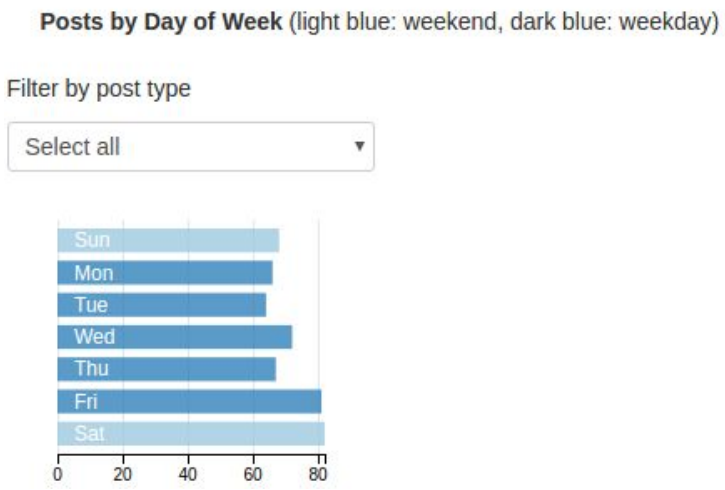


How many total impressions per month come from people who have liked the page? The chart suggests that this group was very active in February and March and less active throughout the rest of the year. Comparing this chart with the bubble chart, we can see that most impressions come from users who have liked the page, reinforcing the claim that most page traffic comes from returning users.

To simplify calculations, I used a total instead of an average so I would not need to account for very large outliers caused by a single month with a few very popular posts being compared with months of many average posts, which would appear very small. The drawback of this decision is that there is no normalization, which makes the visualization more susceptible to biases in the dataset. In the next project I will account for this by normalizing the data and instituting a non-linear scale if necessary.

We can select ranges of the data by clicking and dragging on the chart; this is selection.

A dropdown allows the user to filter by type of post, further enabling data exploration.



I was curious to see which day of the week most posts were made, so I made this histogram. It indicates that most posts were made on Friday or Saturday, which would make sense - people are free from work during the weekend and sometimes get out of work early on Friday. However, filtering the data by type suggests that Friday is a more popular day for link posts, and filtering the data by month suggests that Wednesday is a more popular day for making posts in July. The reasons behind this are not immediately clear, and would require domain knowledge to understand, but it could be due to an attempt to maximize views by the cosmetics company which operates the page.

To make the calculation, I summed up the number of posts by weekday which are currently selected.

To aid in detecting trends, I set the bars representing weekdays to have the same color, while the weekend bars have a different color. This comes from the observation that posting volume seems to correlate with the day of the week.

Hovering your mouse over a bar will display the number of posts associated with bar's weekday; this is probing. Clicking on a box highlights the records from that weekday in all of the other visualizations; this is selection.

A dropdown allows the user to filter by type of post, further enabling data exploration.

Works Cited

dc.js Example Page. Gordon Woodhull, et. al., 2016, https://dc-js.github.io/dc.js/css/dc.css

dc.js Getting Started and How-To Guide. Gordon Woodhull, et. al., 2016, https://dc-js.github.io/dc.js/docs/stock.html

Heatmap Legend. Ben Leitner, 2015, https://stackoverflow.com/questions/31441536/is-there-a-way-to-make-a-key-or-legend-for-a-heat-map-with-dc-js

dc.js - Select Menu Example. Gordon Woodhull, et. al., 2016, https://dc-js.github.io/dc.js/examples/select.html