Facebook Interaction Trends - FreeCodeCamp

Lacey Jeroue

11/6/2019

## Introduction

FreeCodeCamp provided data from posts on their facebook page. The dataset includes date and time the post was made, the type of post (whether it was a link, video, photo, or status), number of users reached, and user interactions (clicks or reactions). Matthew Barlowe provided the initial dive into cleaning and exploratory analysis. Here, I go a step further and examine the user reactions by facebook post topics.

How, if at all, are interactions towards Free Code Camp’s Facebook posts changing over the year? Is the content of Free Code Camp’s posts driving any of those changes?

Check out the dataset and documentation here: <https://github.com/freeCodeCamp/open-data/tree/master/facebook-fCC-data>

## Prepare the workspace

# Load Packages  
library(tidyverse)  
library(RCurl)  
library(lubridate)  
library(pander)  
library(tm)  
library(wordcloud)  
  
panderOptions('table.split.table', Inf) # Do not split table  
  
# Load the Facebook dataset from FreeCodeCamp on github  
gitURL <- getURL("https://raw.githubusercontent.com/freeCodeCamp/open-data/master/facebook-fCC-data/data/freeCodeCamp-facebook-page-activity.csv")  
fb <- read\_csv(gitURL)  
  
  
# Address (convert) variable class  
fb <- fb %>%   
 mutate\_each(mdy, "date") %>%  
 mutate\_each(as.numeric, "reactions")  
  
  
# Take a look at the dataset  
str(fb)

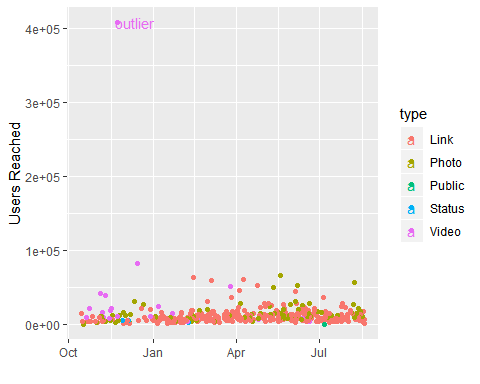
## Classes 'spec\_tbl\_df', 'tbl\_df', 'tbl' and 'data.frame': 420 obs. of 7 variables:  
## $ date : Date, format: "2017-08-18" "2017-08-18" ...  
## $ time : 'hms' num 15:45:00 11:27:00 19:18:00 16:55:00 ...  
## ..- attr(\*, "units")= chr "secs"  
## $ title : chr "The origins of t-distributions and how they can help you make accurate estimates from small sample sizes." "How one camper got his developer dream job" "Trying to code when chat's open" "An interaction designer explains how a \"homeless iPhone\" might work." ...  
## $ type : chr "Link" "Link" "Video" "Link" ...  
## $ reach : num 1768 6941 17399 3751 18248 ...  
## $ clicks : num 44 536 2236 167 1946 ...  
## $ reactions: num 21 99 750 10 474 56 466 403 217 109 ...

## Add Variables

# Add proportion of clicks per users reached & proporiton of reactions per user clicks  
fb <- mutate(fb,  
 propClicks = clicks / reach,  
 propReactions = reactions / reach)  
  
  
# Get sample size by post type  
type\_n <- fb %>%   
 group\_by(type) %>%   
 summarize(  
 Qty\_PostType = n()  
 )  
  
  
# Add label including sample size for plotting later  
fb <- left\_join(fb, type\_n)  
fb <- fb %>%   
 mutate(Type = paste0(type, "\nn = ", Qty\_PostType))  
  
  
# Add date & time variables  
fb <- fb %>%   
 mutate(week = week(date),  
 year = year(date),  
 month = month(date, label = T, abbr = T),  
 datetime = make\_datetime(year, month,   
 day = day(date),   
 hour = as.numeric(substr(time, 1, 2)),   
 min = as.numeric(substr(time, 4, 5))))

## Remove data

Remove the outlier post with over 100,000 reactions. Reduce to three types of posts because there are too few observations for public (n = 1) and status (n = 3) post types.

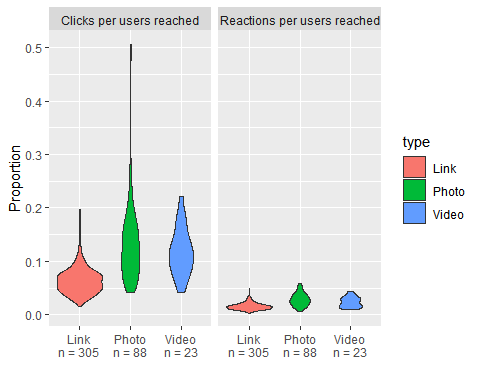


fbdat <- fb %>%   
 filter(reach < 100000 &   
 !type %in% c("Public", "Status")   
 )

## User Interactions

Users more often clicked a link when the post additionally had a photo or a video. Over the course of the year, 12% of users on average clicked a post containing a photo or video compared to 7% for posts containing only a link. Users were also more likely to react to a post when the post contained a photo or video.

|  |  |  |
| --- | --- | --- |
| Type | Average Clicks | Average Reactions |
| Link | 7.4% | 1.9% |
| Photo | 11.8% | 3.2% |
| Video | 11.8% | 2.4% |

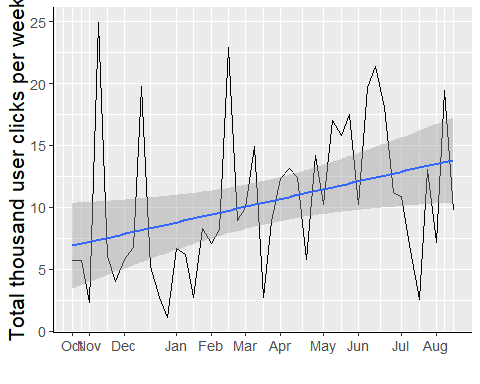


## Posts over time

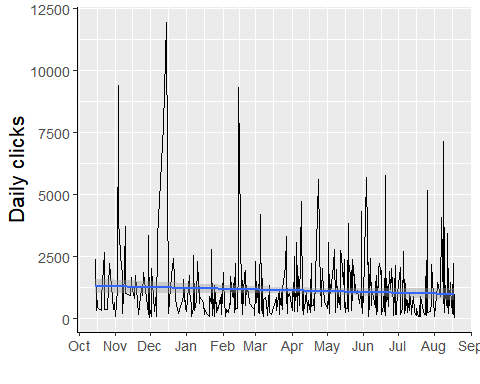
Now we are going to look at total posts by week to reduce the noice in daily posts and get a better picture of the annual pattern. From the graphic below, it is clear that clicks increased over the study year.

# Aggregate by week for a summarized dataset  
fb\_week <- fbdat %>%   
 group\_by(week, year) %>%   
 summarize(n\_posts = n(),  
 sum\_reach = sum(reach),  
 sum\_click = sum(clicks),  
 mean\_clicks = mean(clicks),  
 mean\_propClicks = mean(propClicks),  
 mean\_propReaction = mean(propReactions),  
 mean\_reachedPerPost = sum\_reach / n\_posts) %>%   
 arrange(year, week)  
  
  
# Standardize the week number to begin with one  
# (this works becuase posts were made each week and none were missing)  
fb\_week$weekNo <- seq(1, nrow(fb\_week))  
  
  
# Get more meaningful x labels for graphing (month rather than week number)  
labels <- unique(fbdat[,c("week","month")])  
labels <- left\_join(labels, fb\_week[,c("week", "weekNo")])  
labels <- labels[order(labels$weekNo), ]  
label <- NULL  
for(i in unique(labels$month)){ # some weeks span two months  
 labelx <- labels[labels$month == i, ] # choose earliest week for each month  
 labelx <- labelx[labelx$weekNo == min(labelx$weekNo), ]  
 label <- rbind(label, labelx)  
}

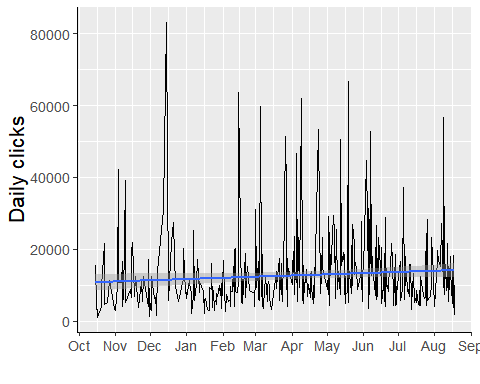
# The plot  
fb\_week %>%   
 mutate(clicked1000 = sum\_click/1000) %>%   
 ggplot(aes(weekNo, clicked1000)) +   
 geom\_line() +   
 geom\_smooth(method = "lm") +   
 labs(y = "Total thousand user clicks per week", x = "") +  
 scale\_x\_continuous(breaks = label$weekNo,  
 labels = label$month) +  
 theme(axis.line = element\_line(),  
 text = element\_text(size=rel(4)))



fbdat %>%   
ggplot(aes(date, clicks)) +   
 geom\_line() +   
 geom\_smooth(method = "lm") +   
 labs(y = "Daily clicks", x = "") +  
 scale\_x\_date(date\_breaks = "1 month",  
 date\_labels = "%b") +  
 theme(axis.line = element\_line(),  
 text = element\_text(size=rel(4)))



fbdat %>%   
ggplot(aes(date, reach)) +   
 geom\_line() +   
 geom\_smooth(method = "lm") +   
 labs(y = "Daily clicks", x = "") +  
 scale\_x\_date(date\_breaks = "1 month",  
 date\_labels = "%b") +  
 theme(axis.line = element\_line(),  
 text = element\_text(size=rel(4)))



### User click rates increased over the year!

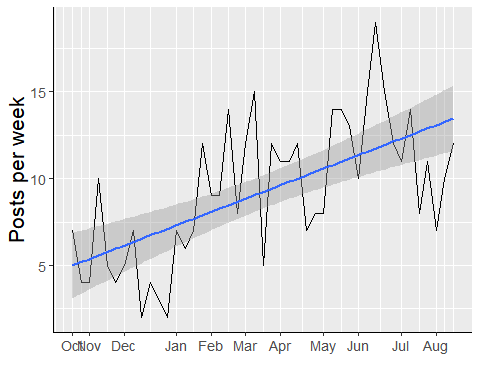
### Well, not so fast!!

It is important to take into account the number of posts from which a user was inspired to click. When we look at the number of posts over time, it is clear that Free Code Camp has increased it’s social media presense on Facebook over the year. By just looking at the total clicks it is impossible to see if clicks are increasing because posts are increasing.

We need to look at the proportion of clicks per post to understand whether user interactions are increasing due to post content rather than the increased Facebook posts.

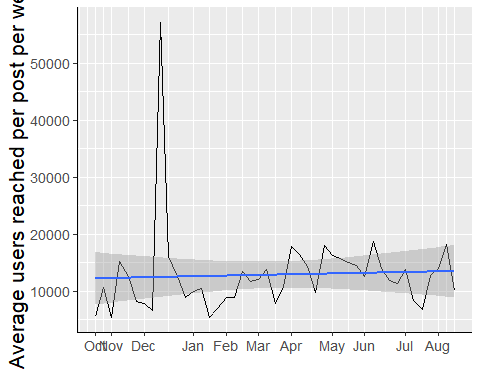
#### First, check out how facebook posts have increased

ggplot(fb\_week, aes(weekNo, n\_posts)) +   
 geom\_line() +   
 geom\_smooth(method = "lm") +   
 labs(y = "Posts per week", x = "") +  
 scale\_x\_continuous(breaks = label$weekNo,  
 labels = label$month) +  
 theme(axis.line = element\_line(),  
 text = element\_text(size=rel(4)))



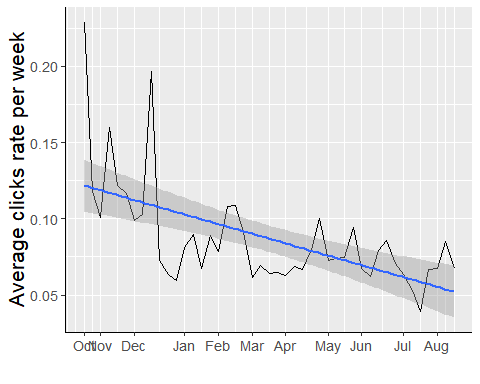
#### Now, see how the relative number of users reached hasn’t changed over the year

fb\_week %>%   
 ggplot(aes(weekNo, mean\_reachedPerPost)) +   
 geom\_line() +   
 geom\_smooth(method = "lm") + # second in a week  
 labs(y = "Average users reached per post per week", x = "") +  
 scale\_x\_continuous(breaks = label$weekNo,  
 labels = label$month) +  
 theme(axis.line = element\_line(),  
 text = element\_text(size=rel(4)))



#### Last, check out how the click rate is actually falling!

fb\_week %>%   
 ggplot(aes(weekNo, mean\_propClicks)) +   
 geom\_line() +   
 geom\_smooth(method = "lm") +   
 labs(y = "Average clicks rate per week", x = "") +  
 scale\_x\_continuous(breaks = label$weekNo,  
 labels = label$month) +  
 theme(axis.line = element\_line(),  
 text = element\_text(size=rel(4)))

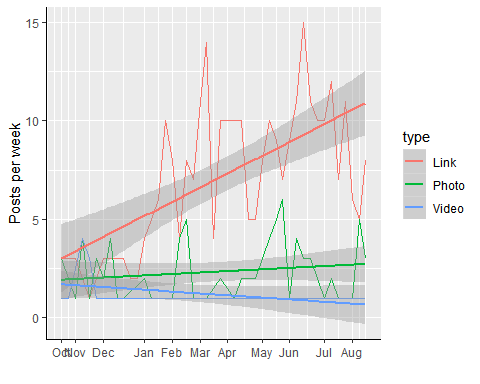


### Take a look at post type

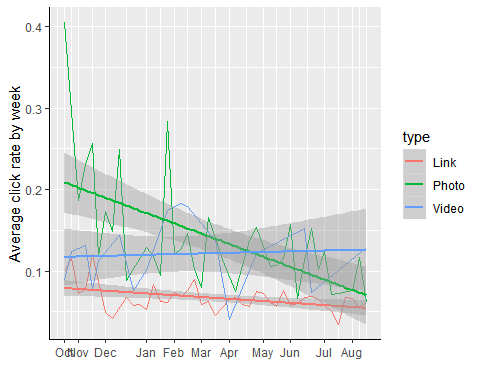
Posts with just a link increased throughout the year while posts with photos stayed about the same and posts with videos stopped early in the year. Only that but the click rate for photo posts deacreased as did single link posts. Recal that post with links garner a lower click rate than those paired with a photo or video.

# Aggregate by week for a summarized dataset  
fb\_week\_type <- fbdat %>%   
 group\_by(type, week, year) %>%   
 summarize(n\_posts = n(),  
 sum\_reach = sum(reach),  
 mean\_propClicks = mean(propClicks),  
 mean\_propReaction = mean(propReactions),  
 mean\_reachedPerPost = sum\_reach / n\_posts) %>%   
 arrange(year, week)  
  
  
# Standardize the week number to begin with one  
fb\_week\_type <- left\_join(fb\_week\_type, fb\_week[,c("week", "weekNo")])

ggplot(fb\_week\_type) +   
 geom\_line(aes(weekNo, n\_posts, col = type)) +   
 geom\_smooth(method = "lm", aes(weekNo, n\_posts, col = type)) +  
 labs(y = "Posts per week", x = "") +  
 scale\_x\_continuous(breaks = label$weekNo,  
 labels = label$month) +  
 theme(axis.line = element\_line())



ggplot(fb\_week\_type) +   
 geom\_line(aes(weekNo, mean\_propClicks, col = type)) +   
 geom\_smooth(method = "lm", aes(weekNo, mean\_propClicks, col = type)) +  
 labs(y = "Average click rate by week", x = "") +  
 scale\_x\_continuous(breaks = label$weekNo,  
 labels = label$month) +  
 theme(axis.line = element\_line())



## Post Topics & “buzzwords”

### Top 5 Titles

Let’s take a look at the top 5 most clicked posts for links and photos

clickable\_Titles <- rbind(fbdat %>%   
 filter(type == "Link") %>%   
 select(type, title, propClicks) %>%   
 arrange(desc(propClicks)) %>%   
 top\_n(5),  
 fbdat %>%   
 filter(type == "Photo") %>%   
 select(type, title, propClicks) %>%   
 arrange(desc(propClicks)) %>%   
 top\_n(5)  
)  
   
names(clickable\_Titles) <- c("Type", "Post Title", "Click Rate")

|  |  |  |
| --- | --- | --- |
| Type | Post Title | Click Rate |
| Link | Freecodecamp shared your post. | 0.1966 |
| Link | Before you drop $2000 on a MacBook read this. | 0.1831 |
| Link | Freecodecamp shared your post. | 0.1559 |
| Link | Stay safe out there. | 0.1468 |
| Link | You can’t make this stuff up. | 0.1447 |
| Photo | Only 5 hours left! Get your low-effort Halloween costume: <https://teespring.com/low-effort-coder-halloween#pid=2&cid=2397&sid=front> | 0.506 |
| Photo | Only one day left to get this low effort Halloween costume: freecodecamp.com/shop | 0.4817 |
| Photo | Timeline Photos | 0.3661 |
| Photo | This subway map shows how agile software development methods relate to one another. | 0.2841 |
| Photo | We either need to significantly improve security or keep some of these devices offline. | 0.257 |

### Buzzwords

Now let’s pull out common words from the post titles and look at post click rates for posts containing those common words.

# Create a 'corpus' object via the text mining, tm package  
post\_titles <- str\_to\_lower(fbdat$title) #pulls the titles into a chr vector  
post\_corpus <- post\_titles %>%  
 VectorSource() %>%   
 Corpus() %>%   
 tm\_map(removePunctuation) %>%   
 tm\_map(removeWords, stopwords('english')) %>%   
 tm\_map(removeNumbers) %>%   
 tm\_map(stemDocument) %>%   
 tm\_map(stripWhitespace)   
   
  
# Make a word cloud  
wordcloud::wordcloud(post\_corpus, max.words = 80, random.order = FALSE)



# Isolate common words into a Text documet matrix  
commonWords\_tdm <- post\_corpus %>%   
 TermDocumentMatrix() %>%   
 removeSparseTerms(0.99)  
   
# Convert TDM to tibble  
commonWords <- commonWords\_tdm %>% as.matrix()  
commonWords <- tibble(Terms = rownames(commonWords),  
 freq = rowSums(commonWords)) %>%   
 arrange(desc(freq))  
  
  
# Get the average proportion of clicks for articals containing each buzzword  
wordClicks <- NULL  
for(word in findFreqTerms(commonWords\_tdm, 12)){ # only terms that occur 12+ times  
 modeldf <- fbdat %>%   
 filter(str\_detect(str\_to\_lower(title), word))  
 wordclicksx <- tibble(Buzzword = word,  
 avg.prop.clicks = mean(modeldf$propClicks),  
 sd = sd(modeldf$propClicks),  
 n = nrow(modeldf))  
 wordClicks <- rbind(wordClicks, wordclicksx)  
}  
  
  
# Visualize clicks by buzzword  
ggplot(wordClicks, aes(x = reorder(Buzzword, avg.prop.clicks), y =avg.prop.clicks)) +  
 geom\_point() +  
 coord\_flip() +   
 labs(y = "Average click rate for articals containing term", x = "Term")

