Ryan Chappell
UW Data Science 350, Methods for Data Analysis

Reddit Front Page and New Page Posts

Introduction

My project was to collect data about Reddit posts on the front page and new page. With this data, I wanted to apply statistical tests and create models for predictions.

ETL

Both the data retrieval and data loading ended up taking a large chunk of time. I quickly ran into issues with the amount of data I was gathering and needed to find better ways of organizing and loading the data into R more efficiently.

Data Retrieval

The application for retrieving the Reddit data was built in Node. I decided on Node since the Reddit API returns data in JSON format where JSON is a natural part of Node/javascript. The source code is here:

https://github.com/ryanchappell/uw-ds-350-reddit-retriever

Initially, I constructed the application to make requests in a loop (one request per minute). I found, though, that it would occassionally fail due to network connectivity or Reddit API availability. My experience with Node is limited so I did not want to spend time figuring out retry mechanisms. Node was made for asynchronous tasks, so implementing synchronous functionality is sometimes a pain. I ended up using Jenkins to run the Node application every minute. This ensured that the data request would occur about every minute and failed requests would be logged.

Data Transformation and Loading

Another issue with the ETL process was that reading each Reddit API result (a JSON file containing 100 posts) into R was taking too long. Roughly 300 JSON files could be read into R per minute. If ~50,000 files were to be read, it would take at least 2.5 hours each time I wanted to load the data (50,000 / 300 / 60 = 2.777 hours). I ended up writing an R script (write-results-to-csv.R) to write the relevant columns to csv files that could be loaded into R more quickly.

Data Analysis

Hypothesis: /r/aww on the front page

- H0 /r/aww composes less than 10% of the subreddits on the front page

- Ha /r/aww composes 10% or more of the subreddits on the front page

The data for this hypothesis comprises a sample of the front page (100 reddit posts or observations) every 15 minutes for a little over 29 days (2016-04-27 19:18:12 to 2016-05-26 21:44:48). This totals 2,796 samples (2,796,000 observations). According to the data, aww subreddit makes up 4.3% of the subreddits on the front page. This is illustrated in in figure 1.

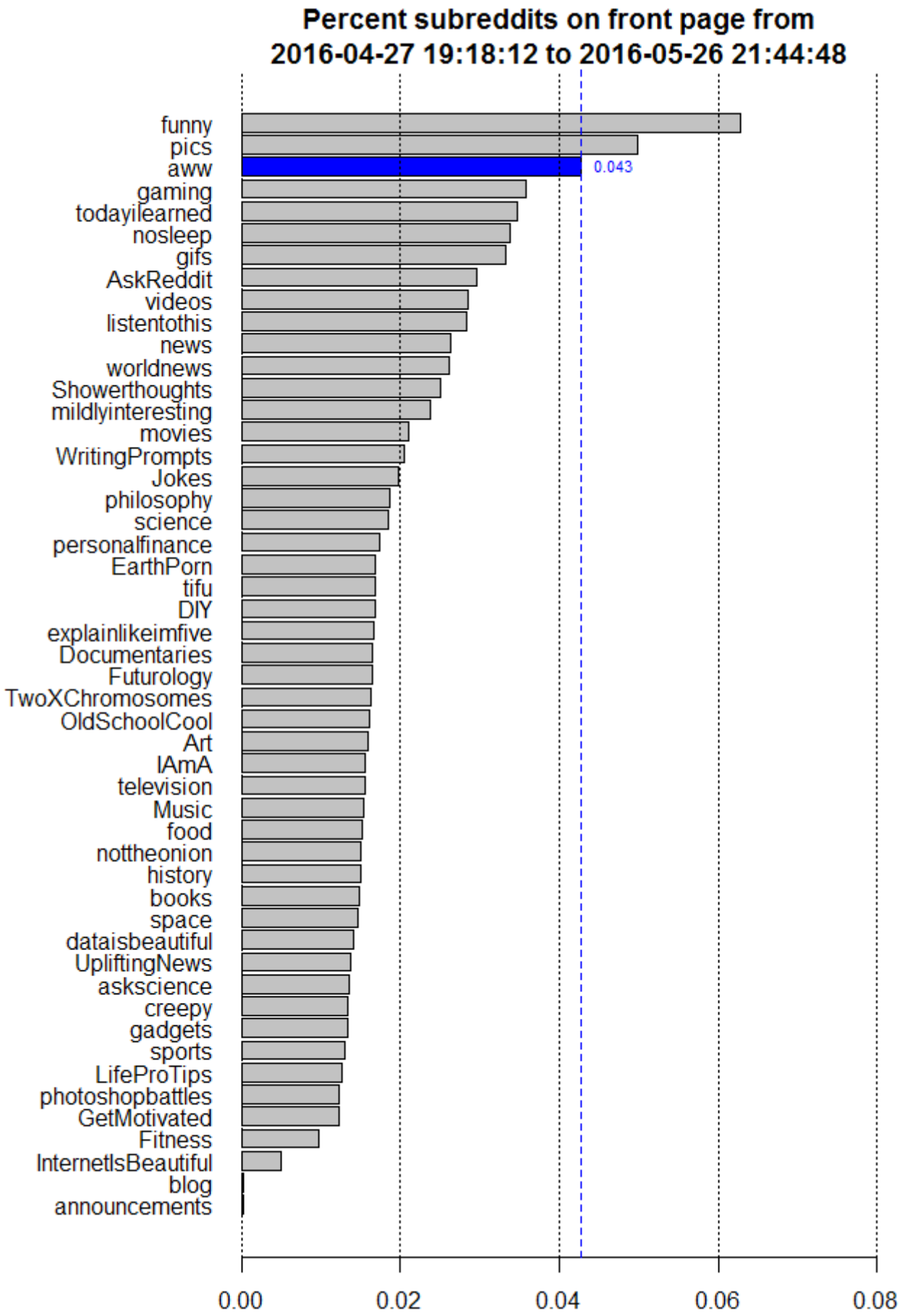
Percent subreddits on front page from
2016-04-27 19:18:12 to 2016-05-26 21:44:48
funny
pics
aww
0.043
gaming
todayilearned
nosleep
gifs
AskReddit
videos
listentothis
news
worldnews
Showerthoughts
mildlyinteresting
movies
WritingPrompts
Jokes
philosophy
science
personalfinance
EarthPorn
tifu
DIY
explainlikeimfive
Documentaries
Futurology
TwoXChromosomes
OldSchoolCool
Art
IAmA
television
Music
food
nottheonion
history
books
space
dataisbeautiful
UpliftingNews
askscience
creepy
gadgets
sports
LifeProTips
photoshopbattles
GetMotivated
Fitness
InternetIsBeautiful
blog
announcements
0.00
0.02
0.04
0.06
0.08

figure 1

In order to confirm statistical significance of the sample, it was tested against a theoretical Poisson distribution. As in figure 2, creating a histogram of the k-s statistic values from 500 theoretical Poisson distribution samples, using the empirical sample mean, provides visual context for the comparison.

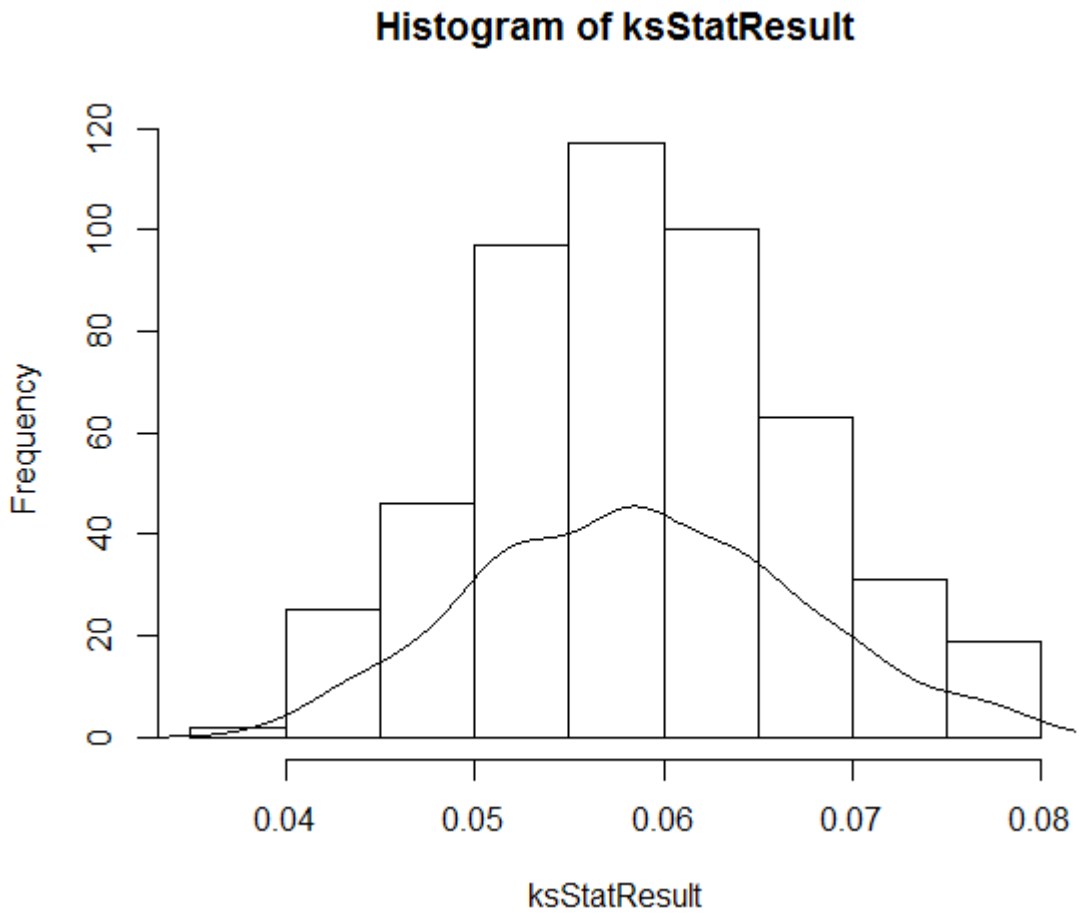

figure 2

The p-value for aww making up 4.27% of the subreddits on the front page is 0.004563. Therefore, we fail to reject the null hypothesis that /r/aww composes less than 10% of the subreddits on the front page.

Predicting Reddit Post Transition with Naive Bayes

In an attempt to predict whether a post on the new page would make it to the front page, I used NLP with Naive Bayes to create a model. I started by finding the intersection of posts on the new page and the front page. This was generally around 2% (new posts that made it to the front page) depending on the fidelity of data I was using. Figure 3 shows the predictive result in a confusion

matrix. The predictive power is quite minimal.

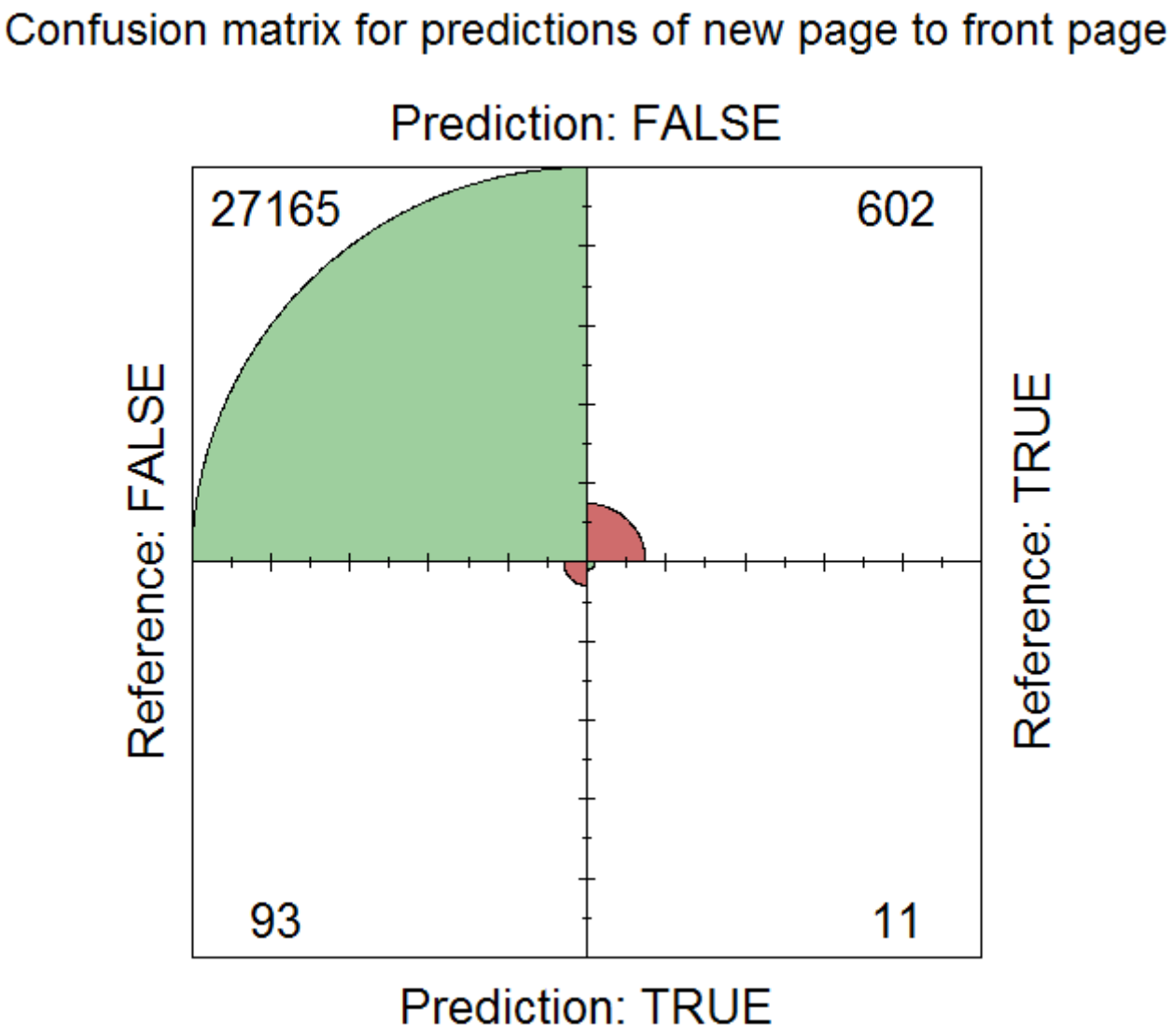

figure 3

Given more time, additional features may have improved the predictive power of the model. For instance, the R script I wrote did not lend itself to describing media posts well. Image recognition could have added additional meta data to each posts (e.g. classifying a post as a picture of a cat or a dog). Using the post comment data may have helped as well.

Predicting Duration on Front Page with Linear Regression

In an attempt to predict the duration a post would remain on the front page, I used a simple linear model. I started by aggregating the total time each post was on the front page to find the target duration. Then, a model was created that included factors like the hour and day of week the post was created. I also used Principal Components to produce a second model for which to compare against the full linear model. After creating both models, I compared the adjusted r squared of the two models. The models ended up not very successful at prediction. As in figure 4, the predicted

duration in hours against the test set is not close to the actual hours.

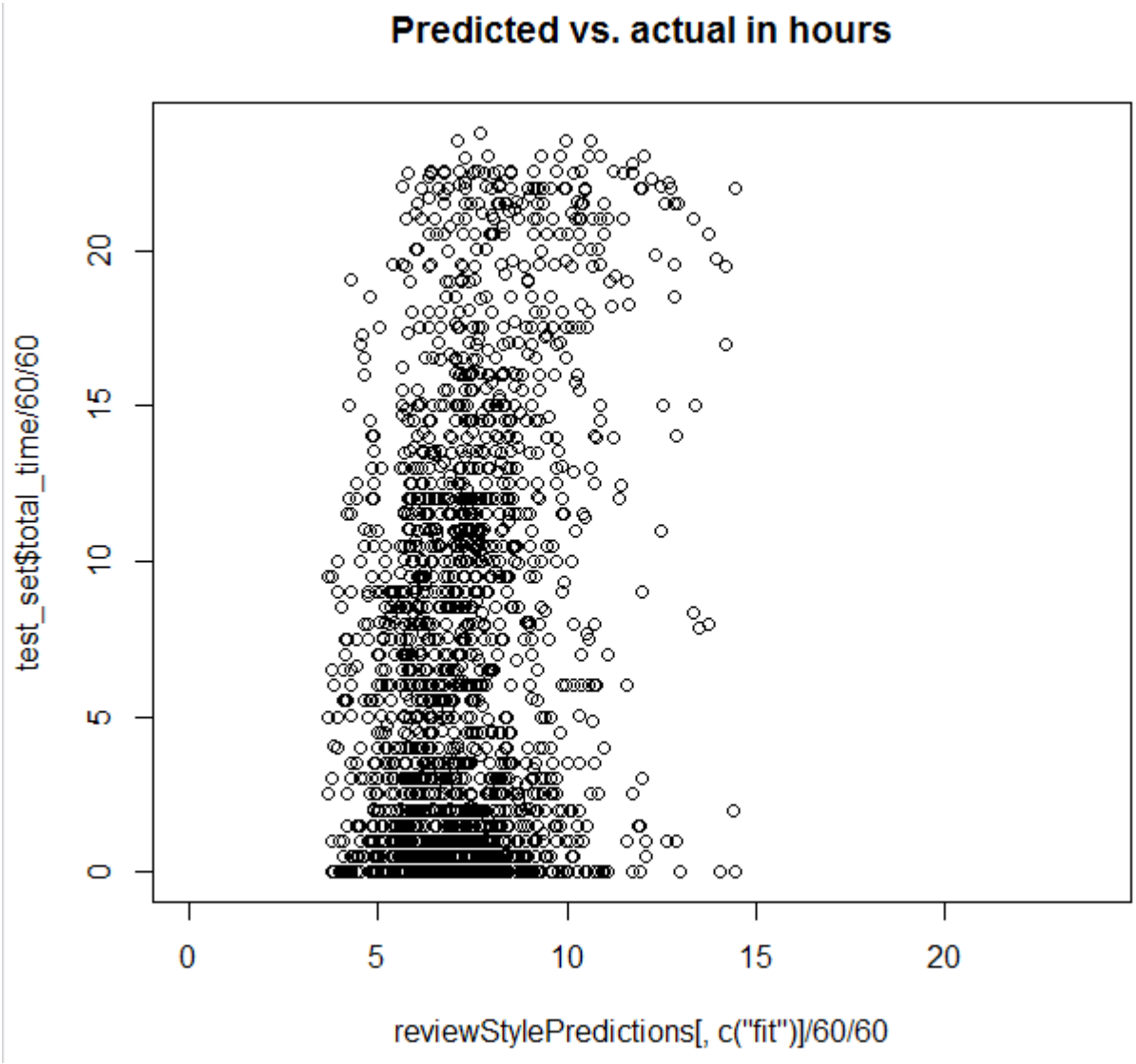

figure 4

Further illustrating in figure 5, the difference in the predicted and the actual value in hours has a mean of around 8 hours.

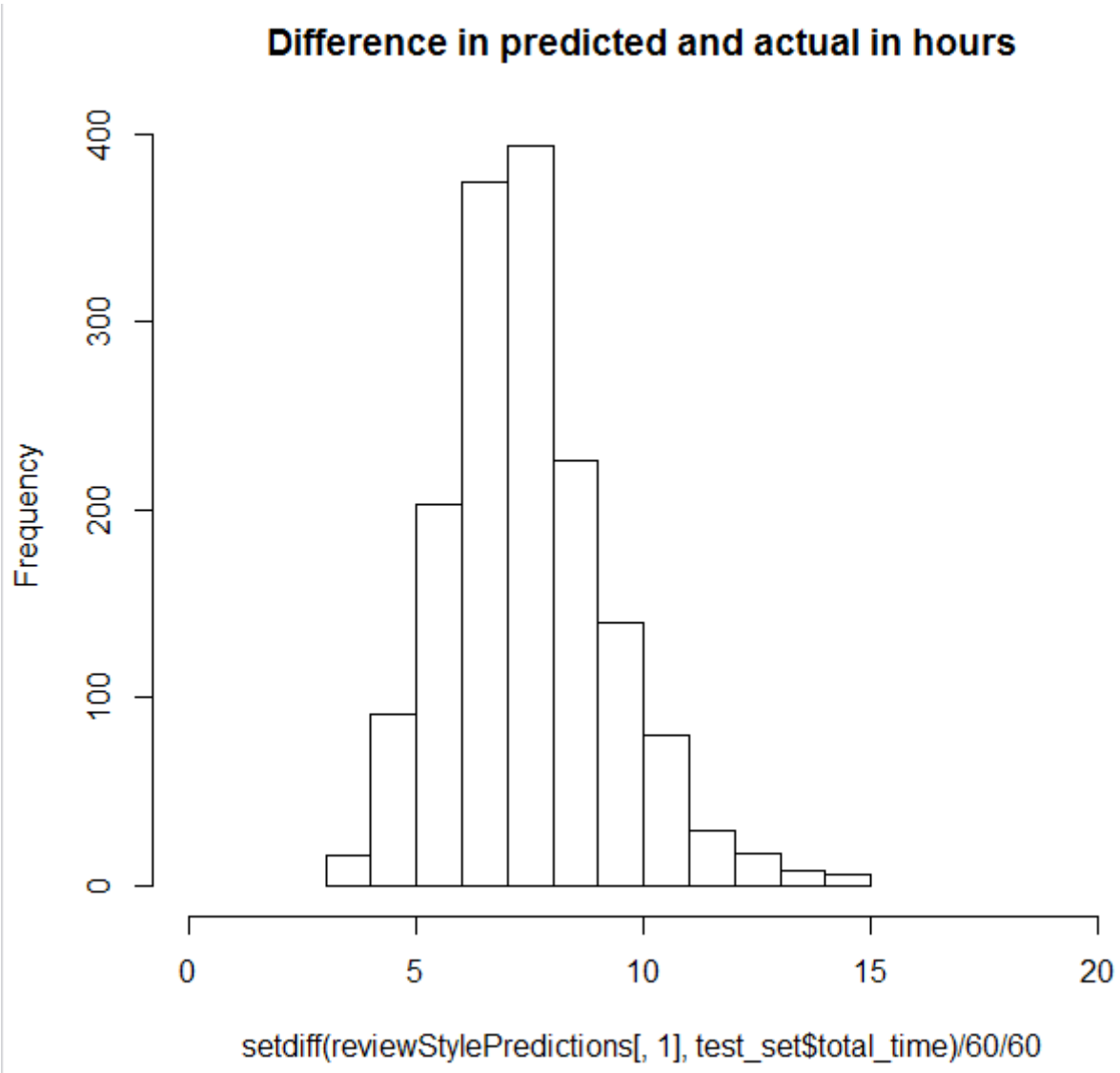

figure 5

I suspect that there is more predictive power to be found in the seasonal nature of the post creation dates. In order to visualize this, I increased the fidelity of the sample rate (to one every 5 minutes) and plotted upvotes over time. Seasonality can clearly be seen in figure 6. This also led me to the discover that there seems to be a cap of 24 hours that a post will remain on the front page.

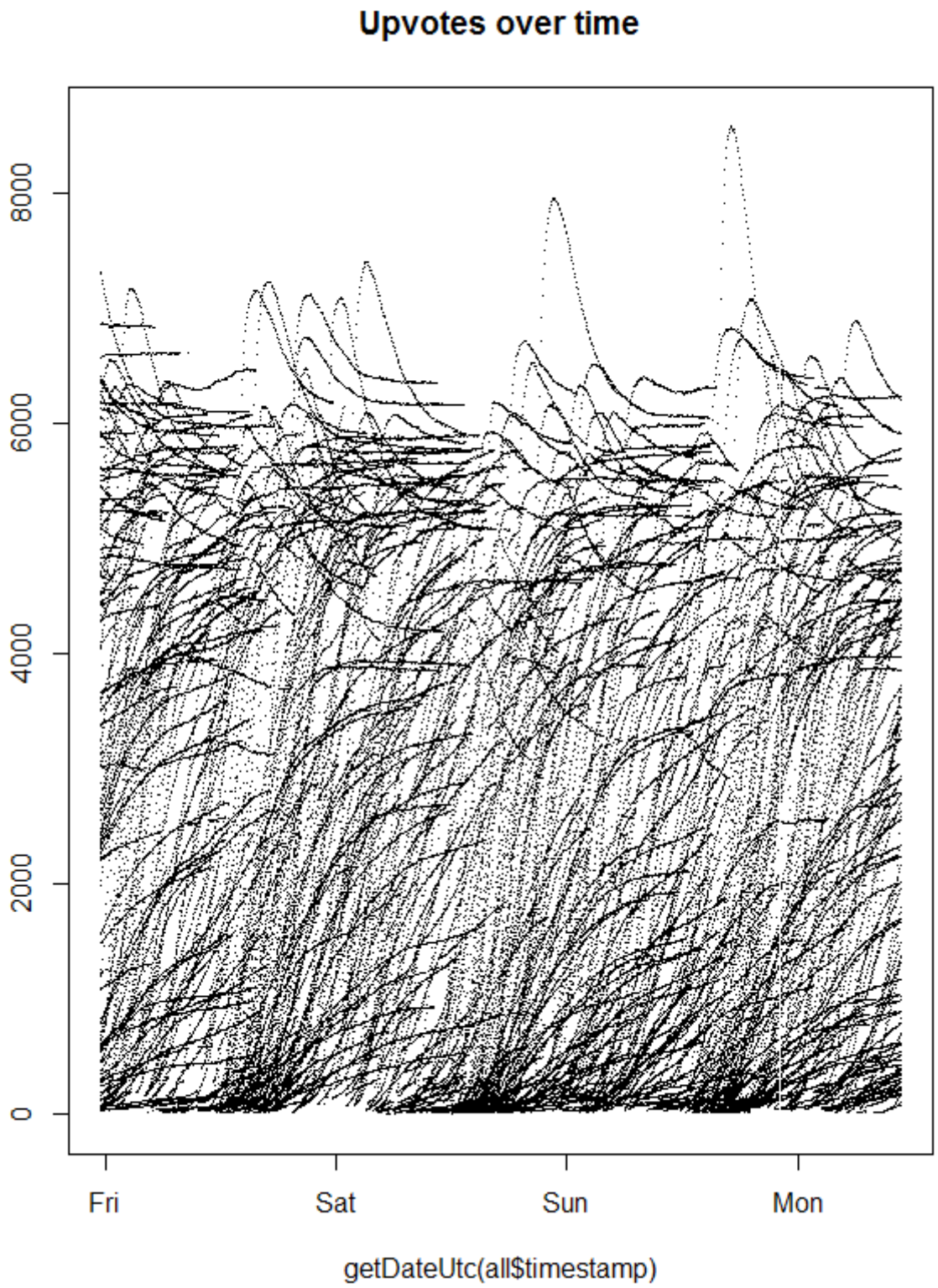

figure 6

Conclusion

In conclusion, I'm satisfied with the result of the aww hypothesis test but the predictive models I created were not very successful. There were parts of the data I did not have time to dig into that may have improved the models. Were I to do this again, I would likely find a data source that did not require as much time to retrieve and organize.