Ian Daniher
2010-10-21

reddit/r/wikipedia: article post to content edit correlation

Background:

This project was designed and executed from the ground up over the past week. As I transitioned from some semblance of balance in my personal and scholastic life to a state I affectionately refer to as "probstataholic," it quickly became apparent that the torrent-upload data I had so carefully recruited from friends and family was going to be insufficient for my analytical needs. Due to difficulty in normalizing metadata and recruiting datasets from elusive private torrent trackers[1][2][3], I abandoned the project in favor of something more accessible.

Over the past few years, I have worked on and off with individuals associated with the Web Ecology Project[4], a consortium of independent social media analysts, primarily based out of Harvard. While "Researching Quantized Social Interaction," the W.E.P. has had its papers including "*Detecting Sadness in 140 Characters: Sentiment Analysis and Mourning Michael Jackson on Twitter*"[5] and "*The Iranian Election on Twitter: The First Eighteen Days*"[6] discussed everywhere from Times Online to Forbes Magazine. For this project, I sought to leverage there formula for success: open information + innovative analysis + CPU time = success.

Reddit is a popular link-sharing website where individuals can post links and upvote, downvote, or comment on them. Divided into topic-and-site-specific subreddits, it's the center of a lively internet community. Wikipedia, a staple of the daily life of so many web-goers, has its own subreddit. For this project, I sought to explore the correlation of the posting of a wikipedia article to reddit.com/r/wikipedia to an change in the number of article edits.

Methodology and Tools:

Data Gathering:

This project consisted of a series of highly-interconnected mathematical explorations, all firmly rooted in the a complex dataset, programmatically gathered from Wikipedia and Reddit. The Wi

My primary metric wikipedia-side was edit count over a defined time period. Optimally, I sought to find a library or API which would allow me to query wikipedia with an article title and a time period. I located Duesentrieb's "Contributors" tool[8] which allowed me to get the complete edit history in CSV format for any en.wikipedia.org page over any time block. Unfortunately, this tool was written in PHP and heavily dependent upon server-side software, leaving my only use route to query his install. The python module urllib made this process feasible and fairly

---

[1] http://www.passthepopcorn.org/
[2] https://hdbits.org/login.php?returnto=/
[3] http://www.demonoid.com/
[4] http://www.webecologyproject.org/
[5] http://www.webecologyproject.org/2009/08/detecting-sadness-in-140-characters/
[6] http://www.webecologyproject.org/2009/06/iran-election-on-twitter/
[7] http://github.com/mellort/reddit_api
[8] http://toolserver.org/~daniel/WikiSense/Contributors.php

seamless. Ten or so hours and a hundred lines of code later, I had constructed a data format providing all of the needed information, and populated with the results of web queries. The result were three lists of dictionaries, one for each Reddit category. I queried for 1000 data points and procured the following:

controversial	new	top-scoring
877	886	929

Of the 1000 queries, responses were discarded if they were not on the English Wikipedia or if their contents were unparsable.

Data Processing:

After the aforementioned data had been gathered, I dug in using an esoteric collection of tools, scripts, and libraries. PyLab, most notably the SciPy library, was used extensively. The PyLab interface to matplotlib was used for graphing. iPython was used as an interactive terminal to facilitate improvisation and on-the-fly programming. Various scripts from class were used to facilitate simple data organization and exploration. Many of the more complicated analyses were pulled from the scipy.stats module.

Explorations:

Normalization Attempts:

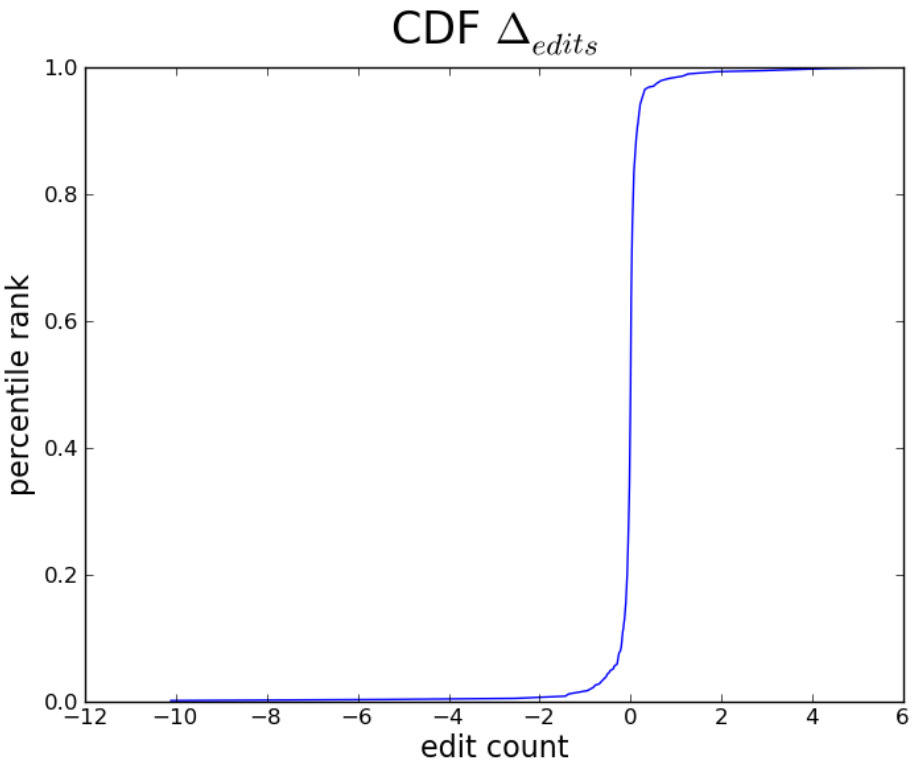
Analyzing the effects and properties of various datapoints required a significant amount of normalization to account for the absurdly high variance of changes in edits between the time period before being posted and after. As the dataset consists of a cross section of Wikipedia pages with varying independent popularity and as the edit history extends varying lengths of time, the number of factors to take into account were prohibitive. To address this, the change in edits was divided by the time distance, giving a positive or negative float equal to the rate of edits per day. In retrospect, using a static time window probably would have been more statistically appropriate.

Prior to deciding upon time-based normalization, I attempted to normalize the edit-delta by dividing through the number of edits prior to posting. This, however, eliminated the utility of our dataset by convoluting our independent and dependent variables.

CDF and PMF explorations:

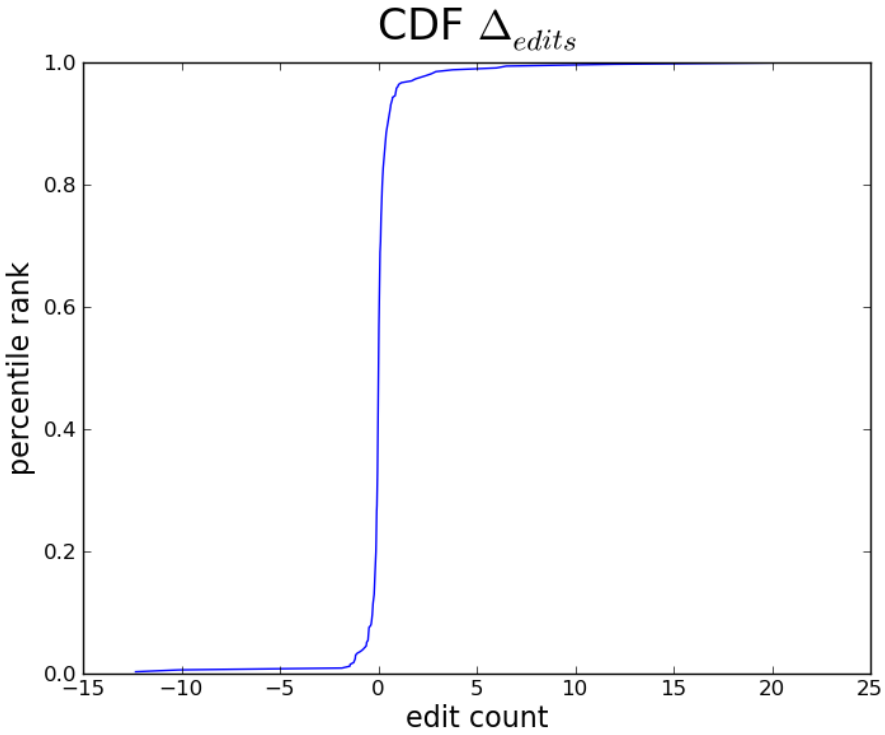
After normalizing our dataset, I plotted the CDF and PMF for my edit-delta statistics in each Reddit category.

This chart shows the edit-delta CDF for popular articles.

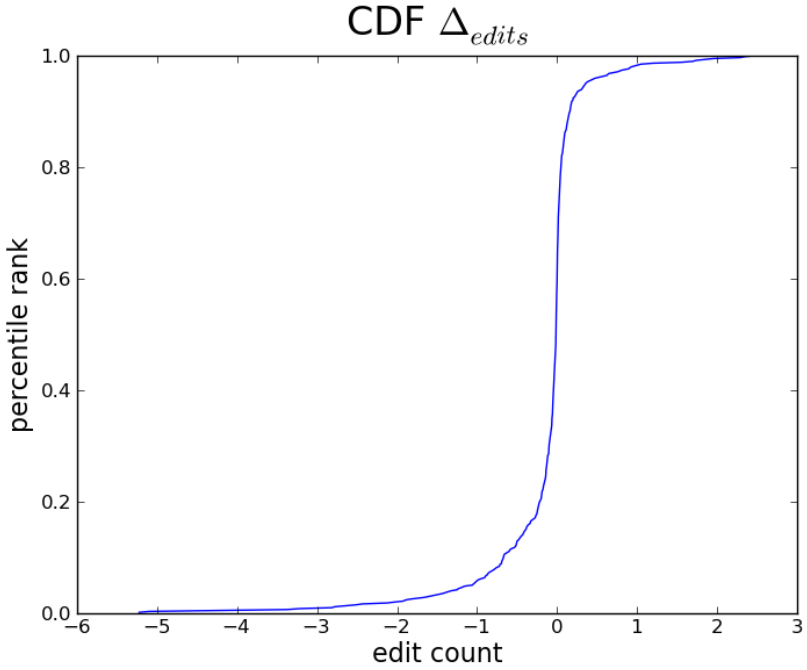


When I first saw this image, I was incredibly disheartened. This CDF is representative of a normal data distribution with a mean at -.02, and a vast majority of points clustered near zero. Zero on this graph is indicative of no change in edits per day after an article was posted to Reddit.

The CDF for new articles looked nearly identical but with a significantly greater spread.

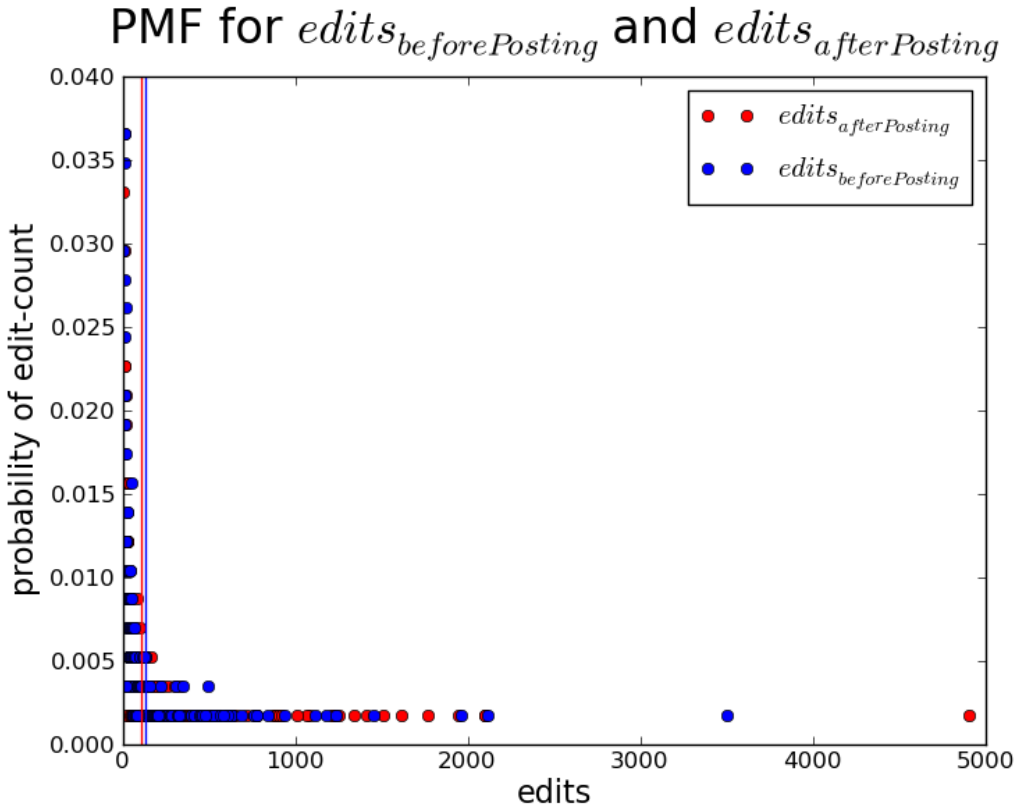


The CDF for controversial articles looked fairly similar, as well:



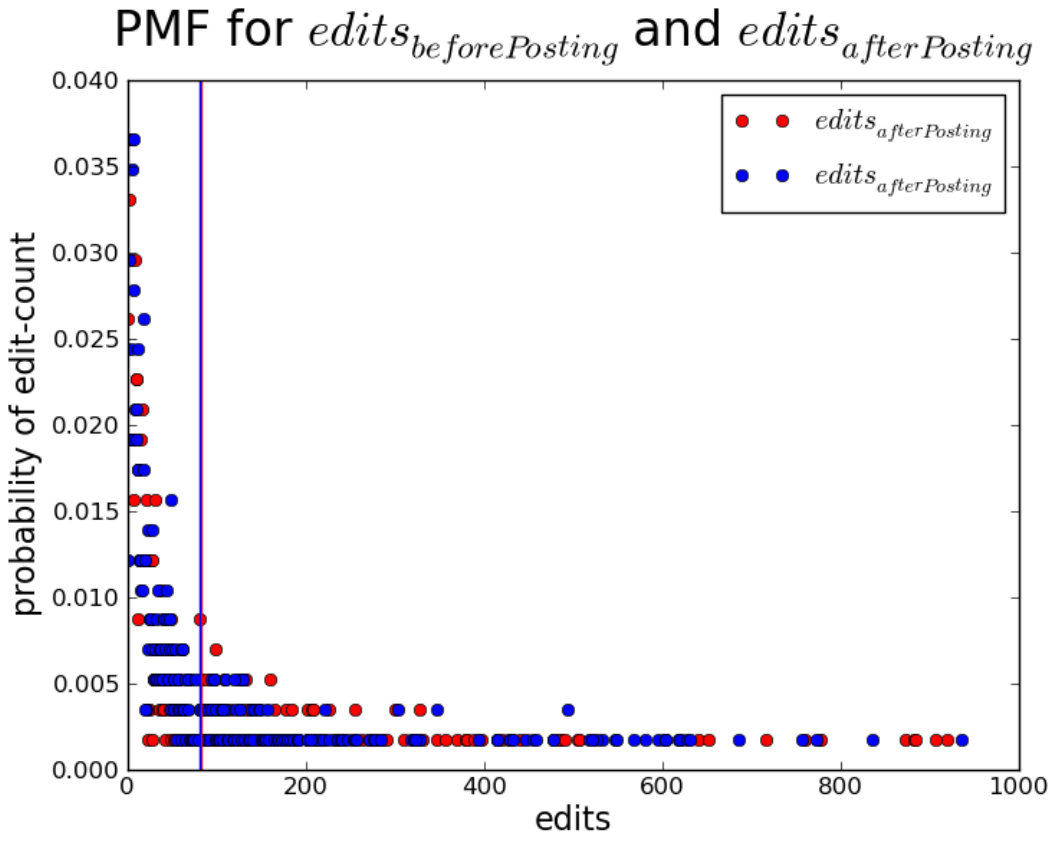
After all this Cumulative-Density-Function heartbreak, it dawned on me that the diversity and depth of my data was quite possibly occluding a trend.

Plotting the PMF for popular articles shed more light on this.



N.B. The vertical lines show means. In this case, there were, on average, fewer edits made after the article was submitted.

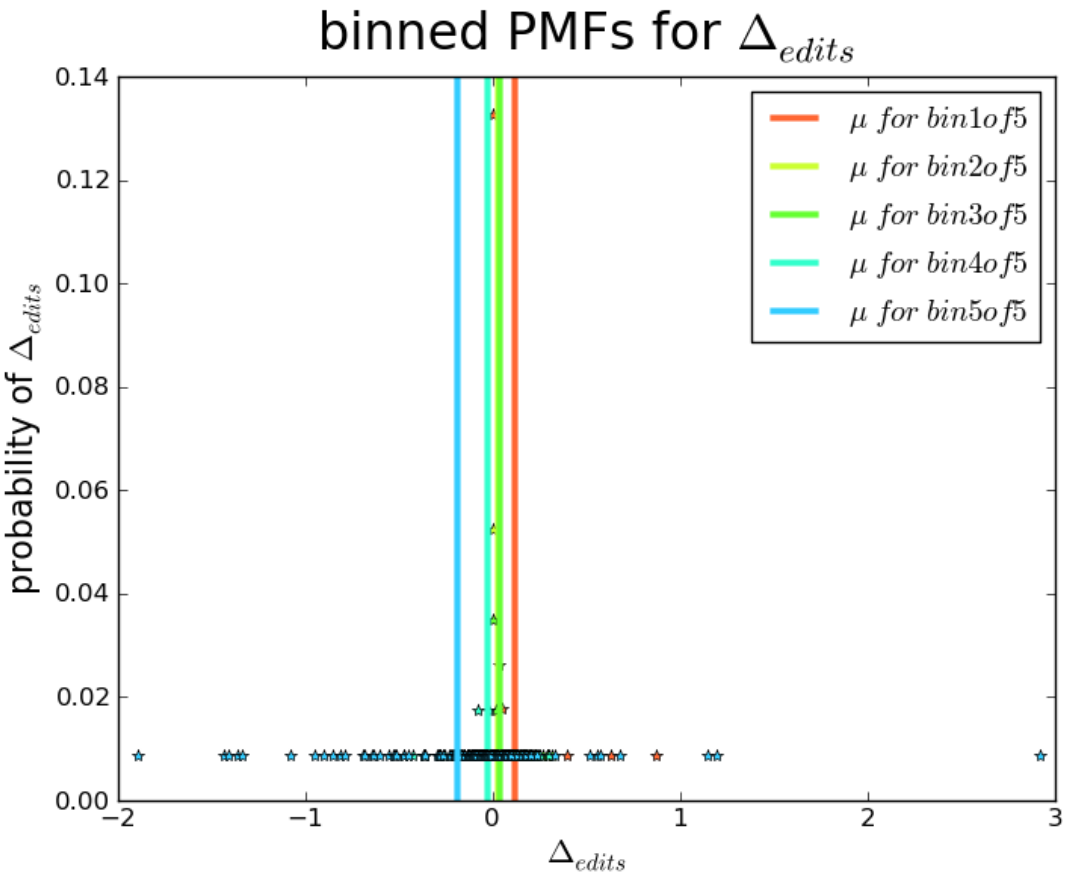
I noticed that there was a significant number of articles with edit counts in the few thousands. It seemed logical to me that the effect of linking from reddit would have a near-negligible effect on such articles. I wrote a script to modify the PMF to only show edit-counts less than a thousand.



The effect was immediately apparent. The mean edit-count after posting grew to be larger than the mean edit-count before posting. This plotting of a conditional PMF suggested that due to the variety of data I gathered, in order to discover any effect, I would need to look at the effect article popularity, shown by edit-count before posting, effected the difference between edits before and after the article was submitted.

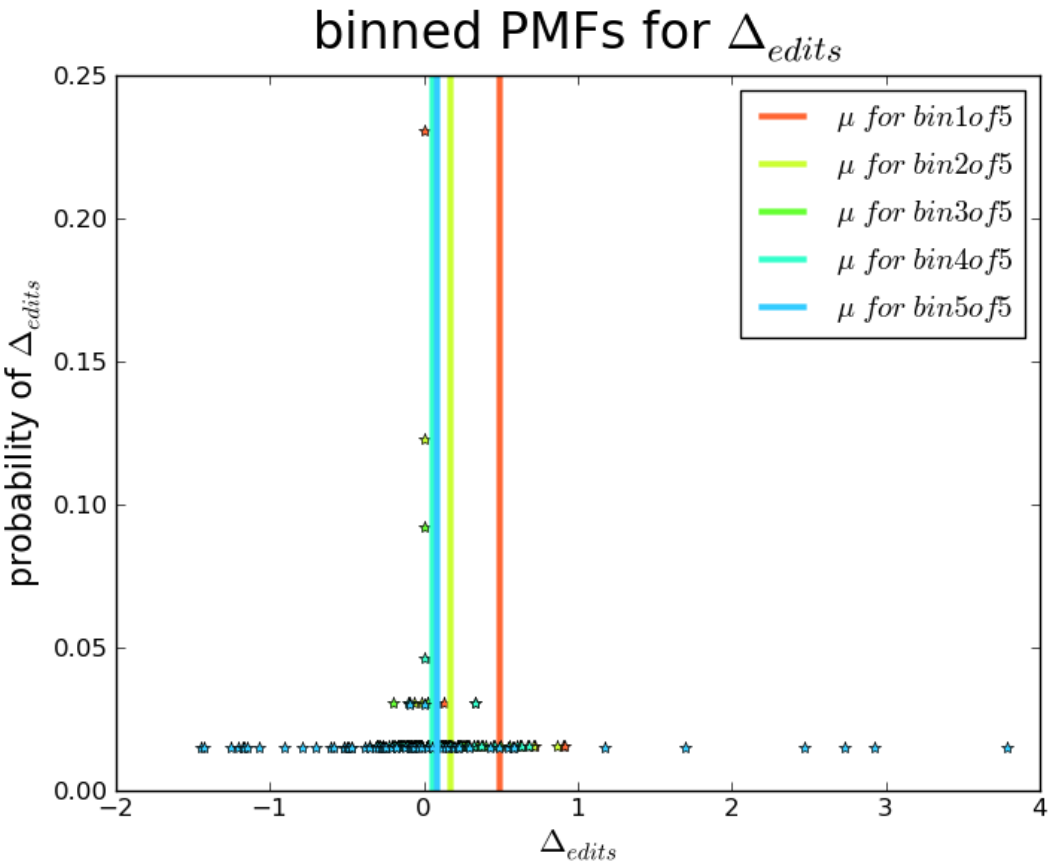
Binning:

The next exploration I carried out was a more complicated version of the above. Rather than merely trim off outliers, I split the dataset into five evenly sized bin by looking at the number of edits the article had prior to being posted. The resultant PMF was incredibly enlightening.

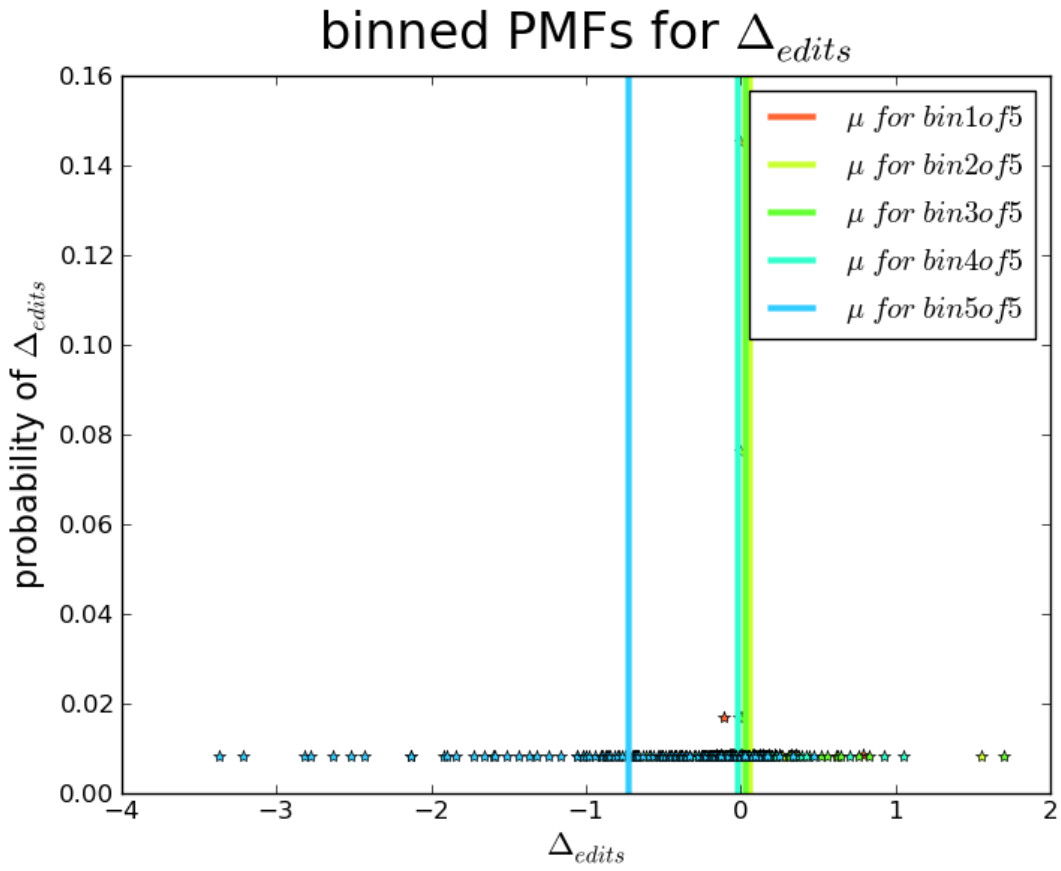


This plot is looking at popular articles. The vertical lines indicate the mean of that particular dataset. The first bin, containing the lowest-edit-count articles, clearly showed the greatest increase after being posted. The last bin, containing the highest-edit-count articles, clearly showed the least increase.

This was also shown in the dataset of new articles:



And, much more subtly, in the dataset of controversial articles:

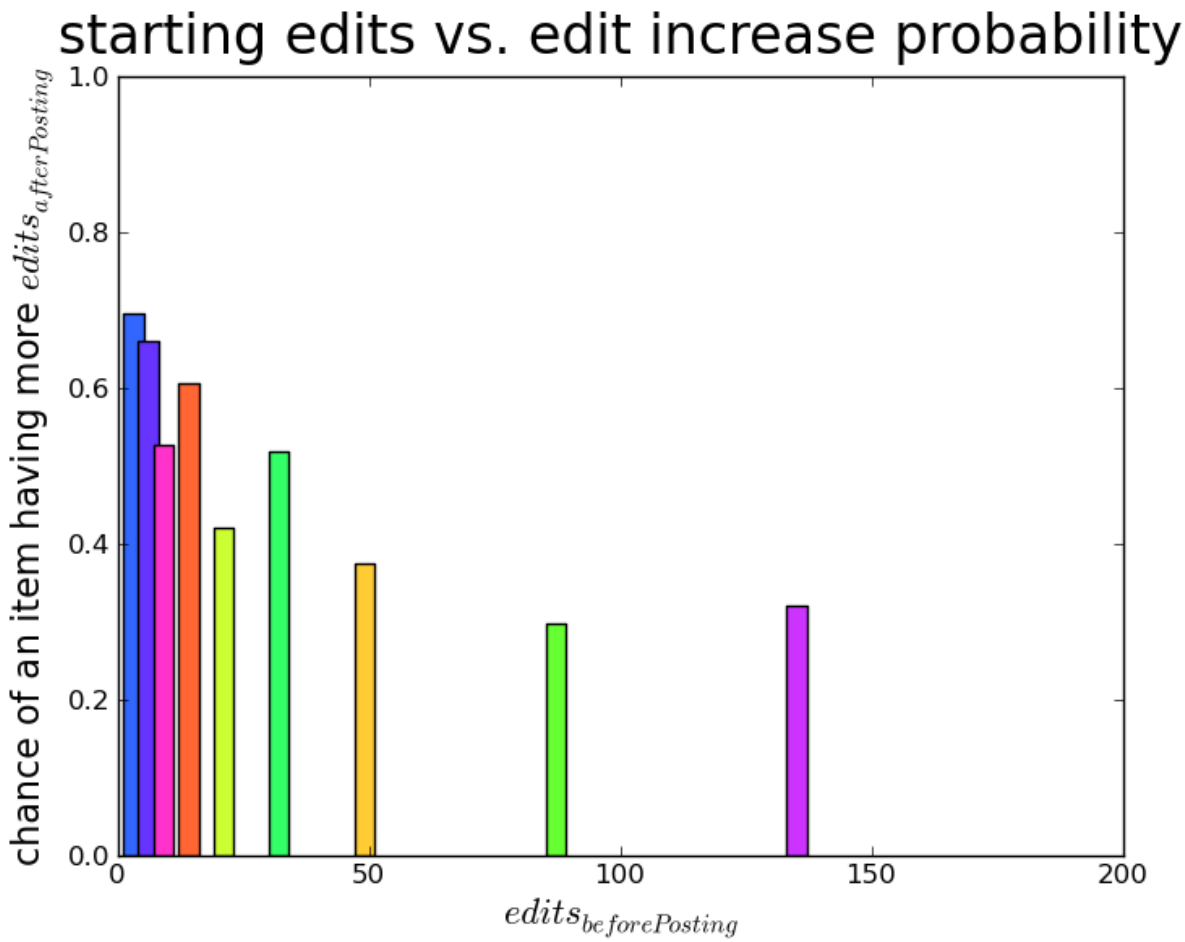


This exploration seemed to strongly indicate a correlation between the probability of an increase of edits after being posted and the 'popularity' of the article in question.

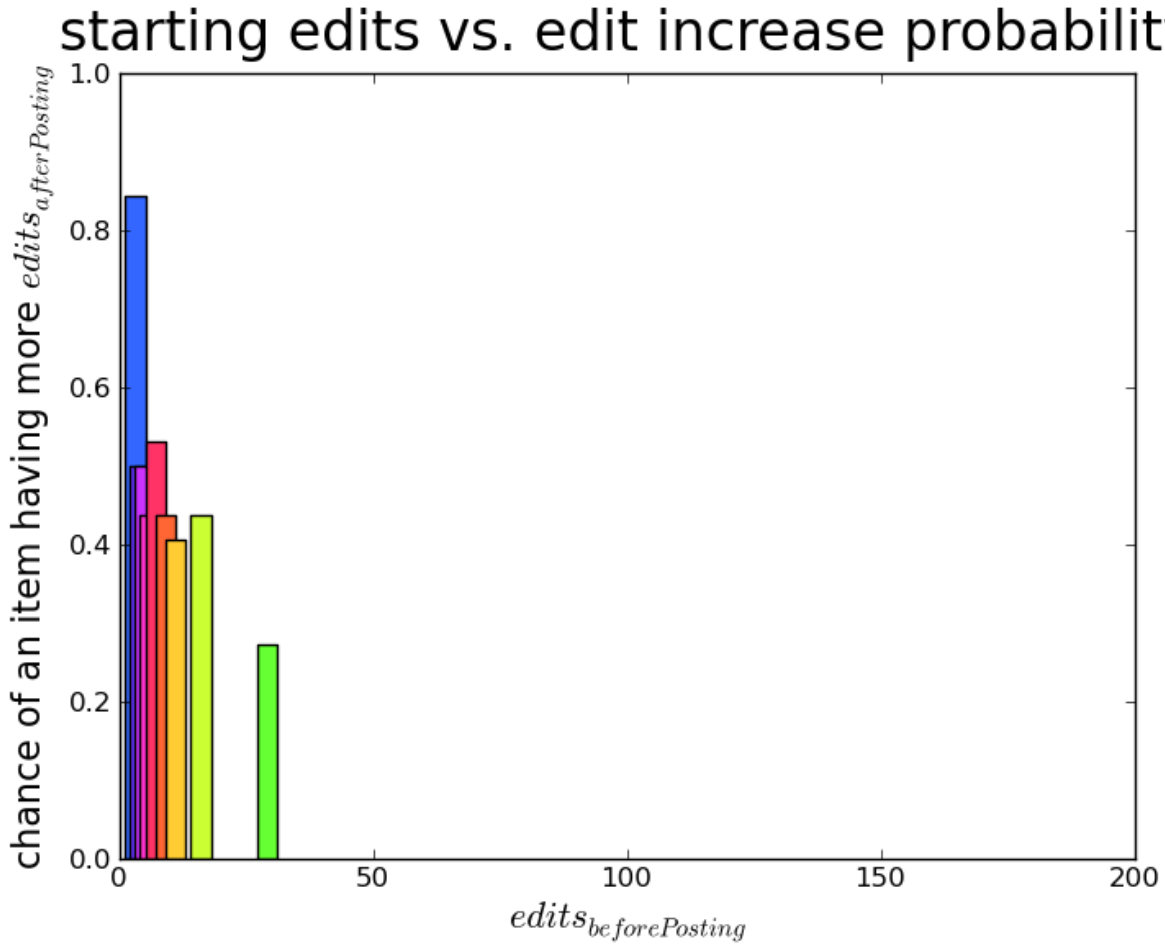
Chance of increase of edits vs. starting edits:

To explore this further, I decided that I wanted to plot the relationship between the probability of an article having more edits after being posted and the number of starting edits. I wrote a function to assign a binary value to the relationship between edits before and after, split the dataset into ten evenly sized groups, and plotted the mean of the relationship-value against starting edits.

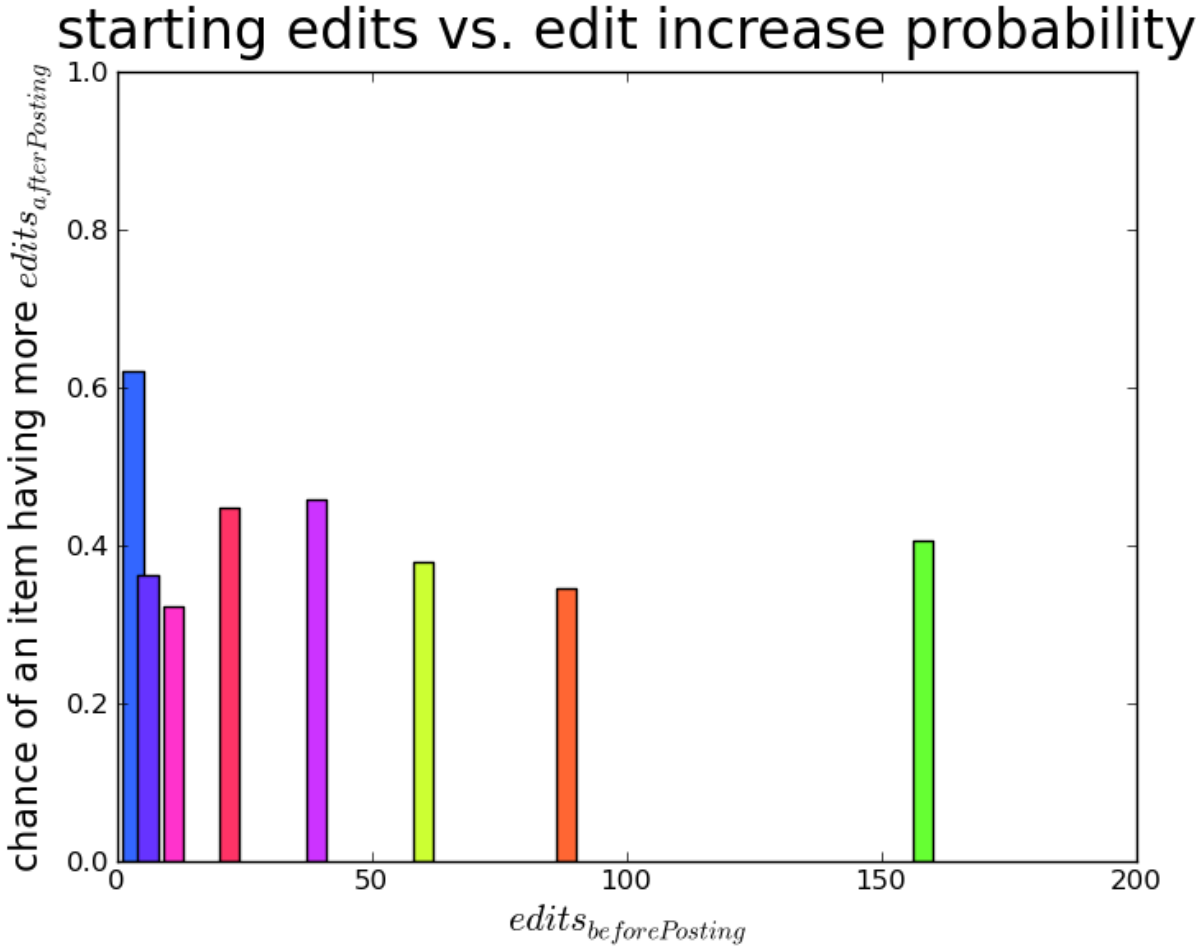
For popular Reddit articles, a strong correlation was shown.



For new Reddit articles, an even stronger correlation was shown.



For controversial articles, a slightly weaker correlation was shown.



The difference between new articles$(\mu = 34.78 days)$, popular articles$(\mu = 357.72 days)$, and the slightly older controversial articles$(\mu = 369.31 days)$ indicates that time also plays a large roll.

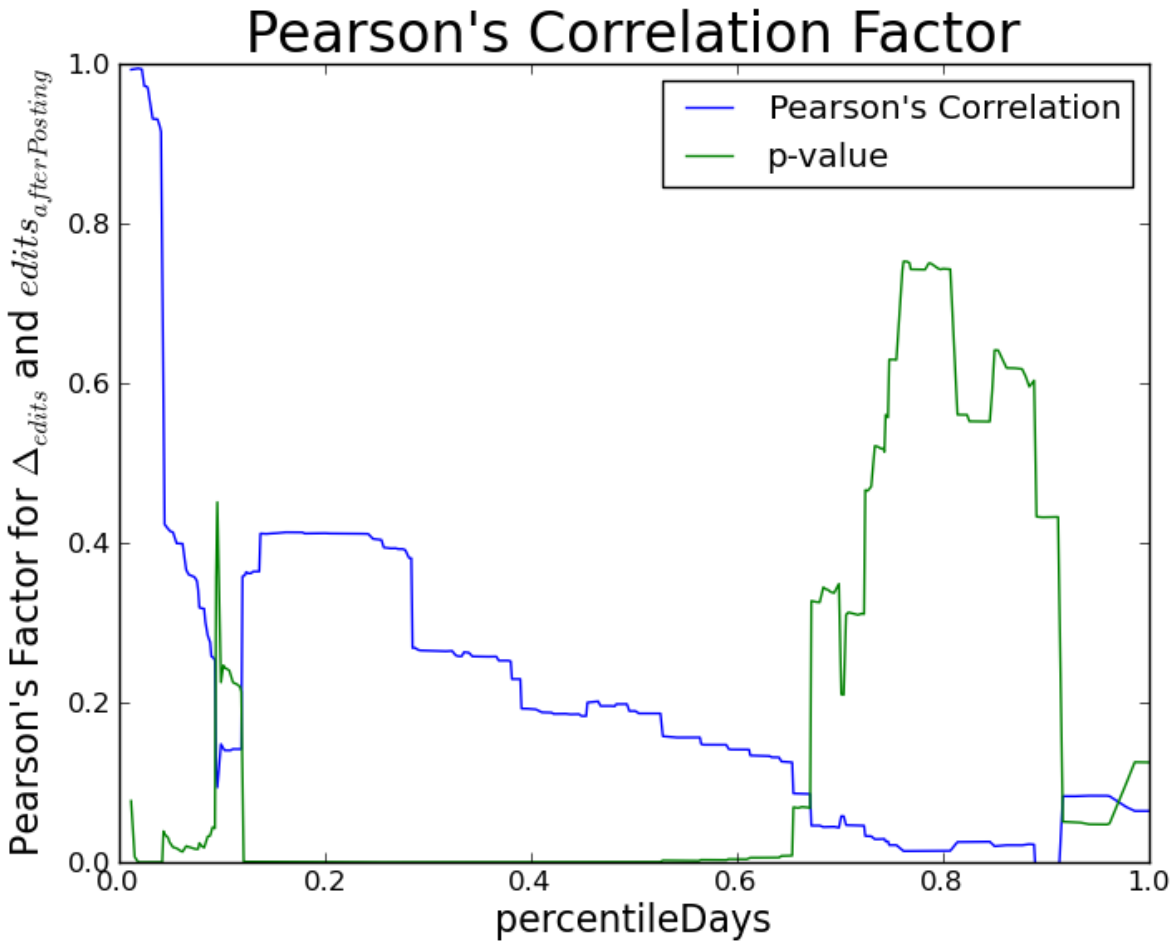
Pearson's Correlation Factor

To explore the roll that duration-of-life plays on the correlation between edits before and after, I utilized Pearson's correlation factor, as implemented by SciPy. SciPy also provides an approximate p-value for the correlation factor.[9] To add a time-variant attribute, I iterated through the dataset, computed the Pearson correlation factor for all datapoints younger than the max age(normalized to 1.0) minus a decrementing variable. Plotting this slightly unusual metric is highly illuminating.

For the 'popular' dataset:

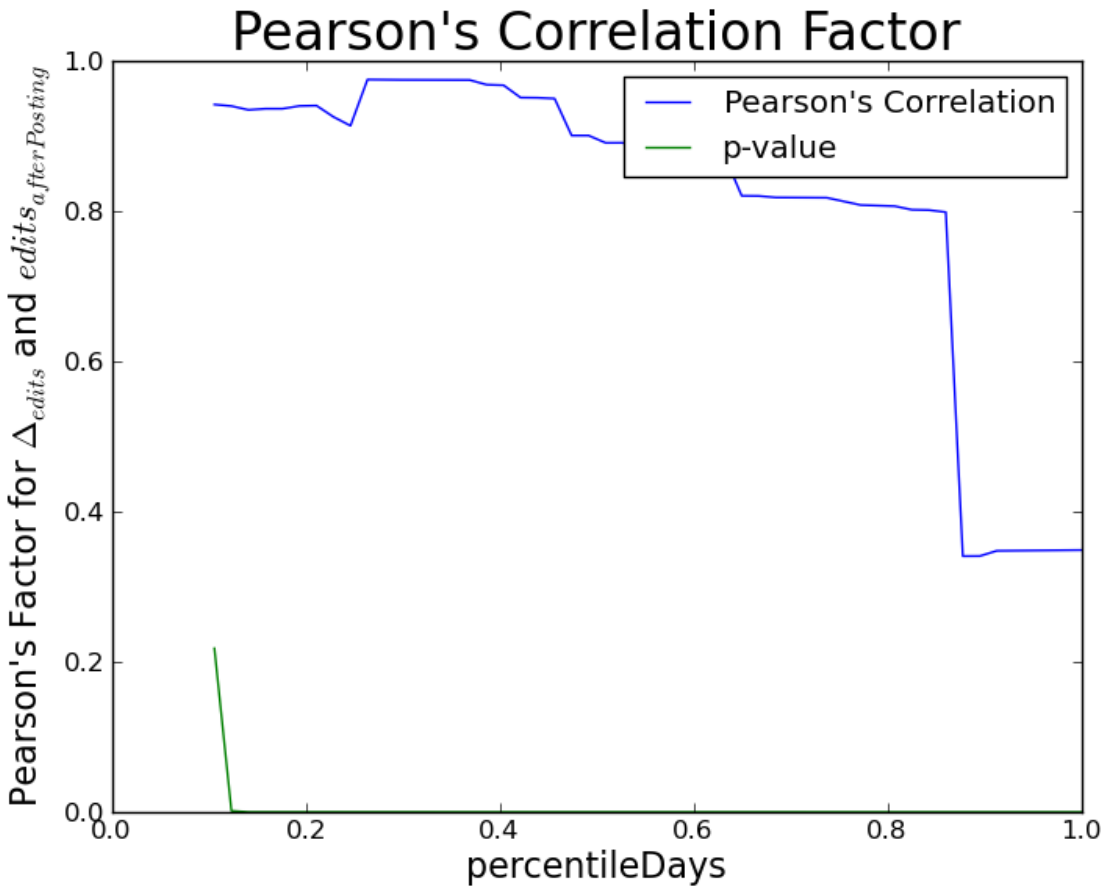
---

[9]See /usr/lib/python2.6/dist-packages/scipy/stats/stats.py - in the future, perhaps the probstat final could be to redo the scipy stats library?
Or at least redocument it?



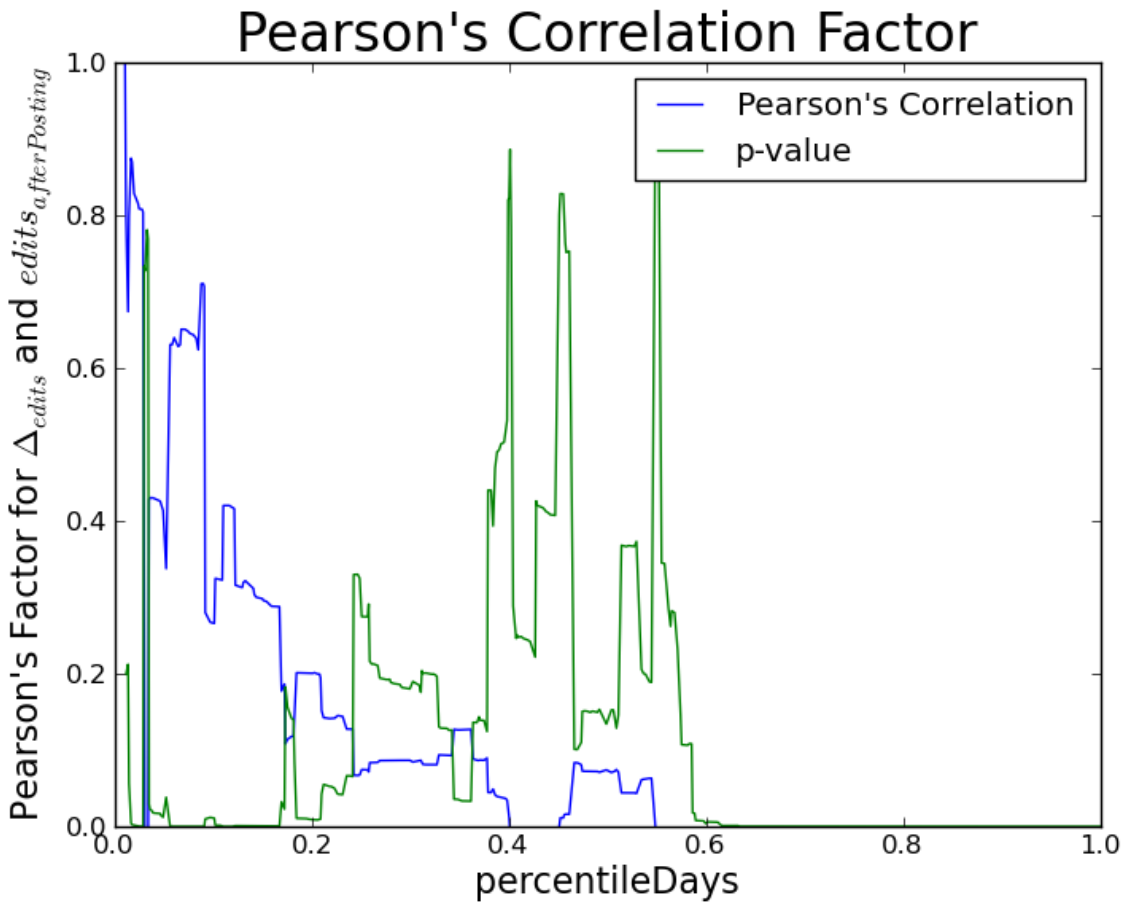
This graph shows how the Pearson correlation value decreases nearly exponentially as the lifespan increases. Meanwhile, the p-value, the chance that the apparent effect is due entirely to random chance, increases significantly.

This general trend also holds true for the other datasets.
New:



The correlation here is much stronger than in the previous dataset. As a result, the p-value is significantly lower, to the point of being unable to be clearly shown on this scale.

Controversial:



This dataset, as always, is troublesome. Even so, Pearson's Correlation Value decreases reliably and the P-value increases significantly.

Conclusions from Explorations:

This exercise was interesting in that utilizing the brute-force iterative abilities of a computer, coupled with the subjects covered in class, I was able to tease both temporal and scalar correlations out of a seemingly perfectly normal $(in\ the\ \aleph - sense!)$ dataset.

All three independent datasets were shown to vary in a possibly predictable manner with regards to time and starting edit-count. With further development, numerical parameters could be estimated for the correlation factors, potentially providing forewarning to Wikipedia sysadmins by monitoring /r/wikipedia. Generalizations of this effect could be applied to analyze and predict other forms of aggregator / community-contributed-media interactions, including blog comments linked from RSS feeds and bug reports on projects linked to off popular websites.

As for the noted lack of Bayesian in the above... it didn't quite "fit" with my line of inquiry, and I wasn't going to spend the time trying to force it.project

Closing Notes:

This quarter has been one heck of a rollercoaster in my personal life. I got behind early, and, for all practical purposes, have stayed behind. ProbStat, being a two-credit course, was deprioritized in comparison to Orgo, Mechanics, et alia. However, I'm quite pleased with the outcome of this project, with my learning in the class, and will quite possibly continue this line of inquiry in collaboration with Seth Woodworth[10] and other members of the Web Ecology Project.

[10]twitter.com/sethish