Miles Waugh

Mr. Palm

AP Statistics C°

December 5 2022

# Do More Popular Scratchers Receive More Loves per View on Average?

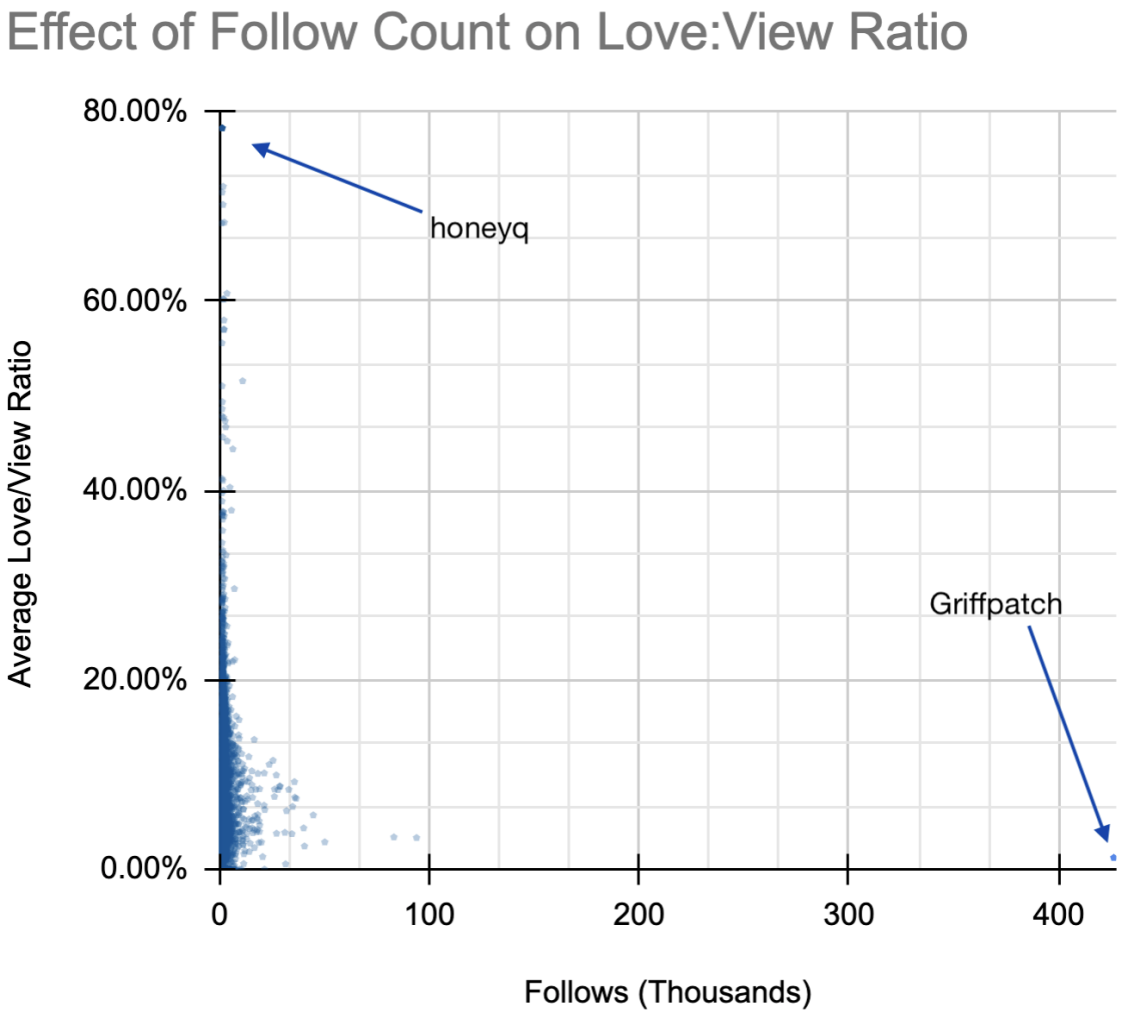
Scratch is a high-level block-based visual programming language and website aimed primarily at children as an educational tool for programming, with a target audience of ages 8 to 16 (though many believe the actual audience consists of older individuals). Users on the site, called Scratch users, Scratchers, or users, can create projects on the website using a block-like interface. Publicly-posted projects may receive views, loves, and favorites from other users. Views are counted as the number of times a user runs a project for the first time after loading the project, and each user can choose to love and/or favorite a project once. Users can also follow each other, and it does not have to be a mutual connection.

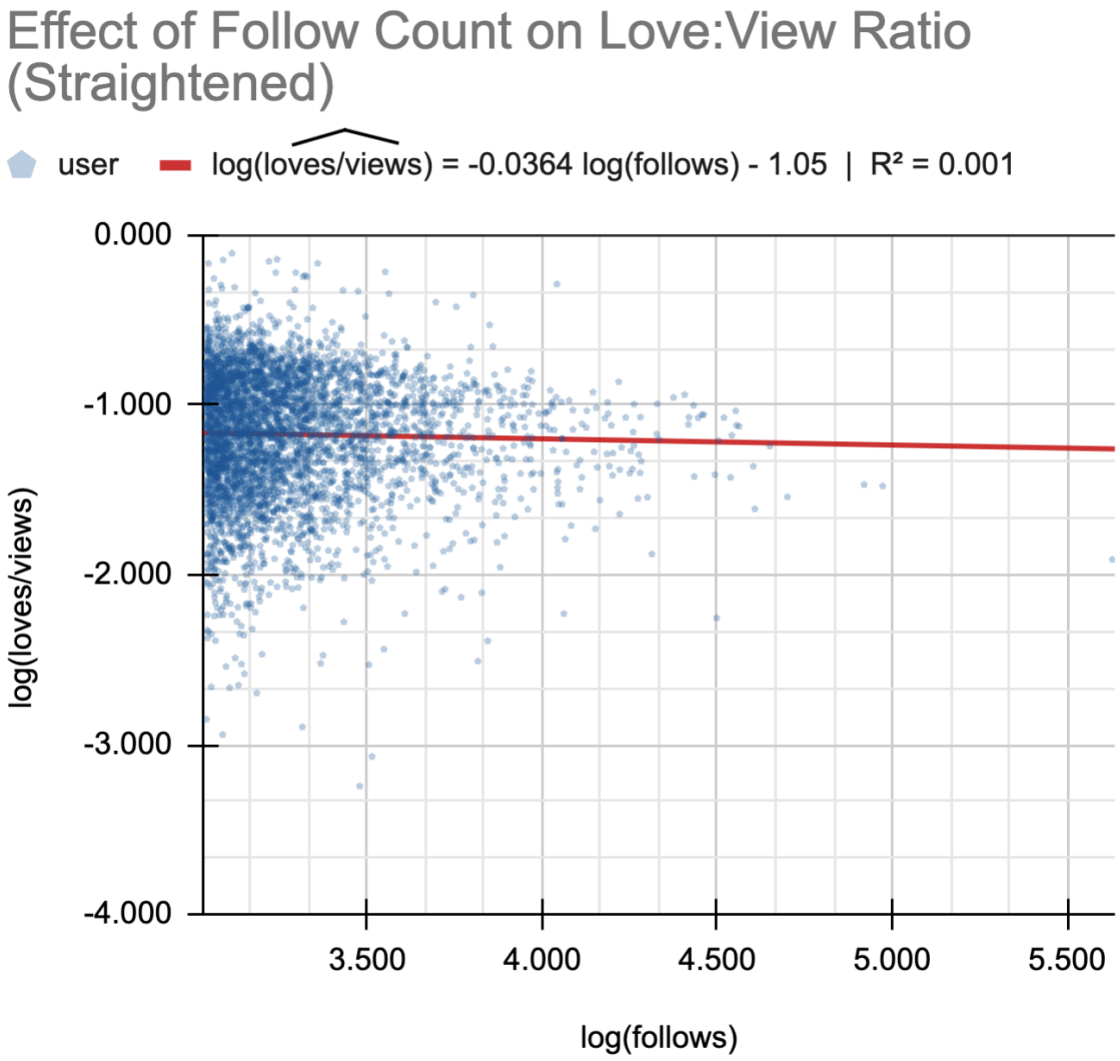
Scratch is very dear to my heart, because it was one of the first programming languages I ever used. I established my first account on the platform 8 years and 4 months ago, and it enabled me to express creative problem-solving outside the apparent rigid nature of elementary school. Because of this, I wanted to do my AP Statistics project on data collected from the Scratch website. Through this project, I hope to learn more about the relationships between different project statistics on Scratch.

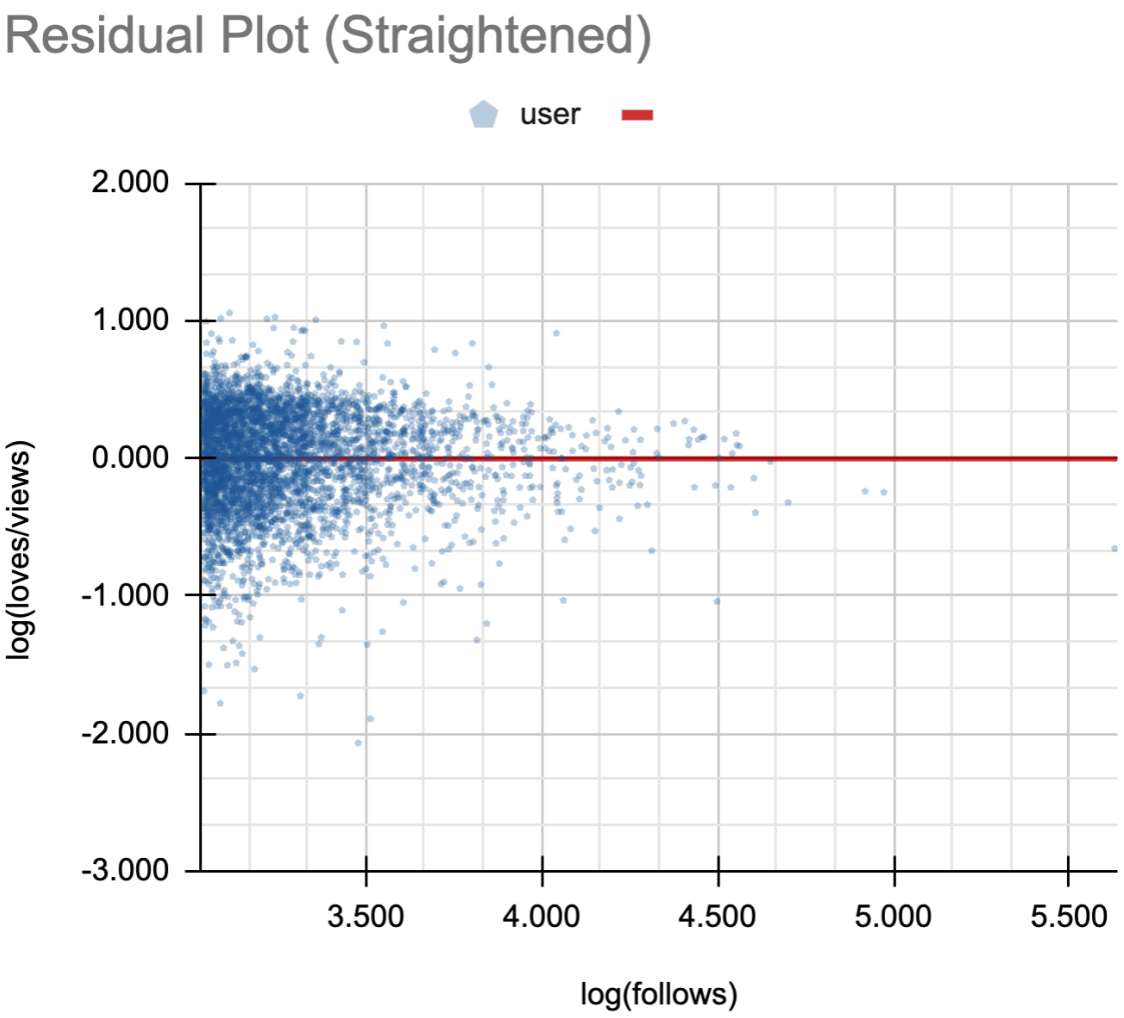
I will be investigating the relationship between Scratcher popularity and how likely their projects are to be ‘loved’ by a viewer. I will measure the Scratcher’s popularity by the number of followers they have, and I will measure ‘love’ likelihood as the average ratio between total loves and views on all public, published, and visible projects that the Scratcher has produced.

I predict that more popular Scratchers will receive fewer loves per view on average than less popular Scratchers, largely due to the demographics of viewers on various types of projects. Scratchers with small follow counts likely do not produce projects favored by the general public, or they would likely gain more exposure and followers as a result. I would therefore expect that group to have a small love-to-view ratio. Younger Scratchers also commonly participate in a practice called 'F4F' (follow-for-follow), where both users agree to follow each other. The Scratchers then 'love' each other's projects in order to mutually boost each others' popularity. This would cause an increase in the love-to-view ratio in users with a decently large following. Users with a massive following usually gain their follower base through either well-made video games, or through direct or indirect affiliation with Scratch (e.g., directly by being on the Scratch Team, or indirectly by being a senior developer for Scratch Addons). Blindly following members of the Scratch Team despite their lack of quality projects on the platform, or avoiding internet filters for online gaming websites by playing Scratch remakes, tend to be the defining characteristics of the largest crowd of users on the platform: the youngest kids who do not know or want to learn to code, and who likely do not tend to love and favorite projects since they have a less genuine appreciation for them. I would expect this to reduce the love-to-view ratio in Scratchers with massive follow bases, hence making the overall association negative.

To extract and collect data from the Scratch website, I used the Scratch API. My code is posted publicly on GitHub (github.com/piano-miles/Scratch-Userdata), and the data can be collected by running `./local/main.py` with Python 3. I suggest leaving at least 4 GB of RAM available for running the full data collection process.


Effect of Follow Count on Love:View Ratio
80.00%
60.00%
40.00%
20.00%
0.00%
Average Love/View Ratio
honeyq
Griffpatch
0
100
200
300
400
Follows (Thousands)


Effect of Follow Count on Love:View Ratio (Straightened)
user
log(loves/views) = -0.0364 log(follows) - 1.05 | R² = 0.001
0.000
-1.000
-2.000
-3.000
-4.000
log(loves/views)
3.500
4.000
4.500
5.000
5.500
log(follows)


Residual Plot (Straightened)
user
2.000
1.000
0.000
-1.000
-2.000
-3.000
log(loves/views)
3.500
4.000
4.500
5.000
5.500
log(follows)

The data in the original scatter plot is highly concentrated in the bottom-left corner of the graph (a right skew in the distribution for loves/views and an even heavier right skew in the distribution for follows). I, therefore, chose to re-express the data by taking the log of both distributions in an attempt to make the distributions more uniform. The equation of the regression line is log(predicted loves/views) = –0.0364 log(follows) – 1.05. The slope is -0.0364, meaning that the love/view ratio will tend to decrease by $1 - 10^{-0.0364} = 8.03978\ldots\%$ for a 10-fold increase in follower count, according to the model. The y-intercept is -1.05, meaning the love-to-view ratio is approximately $10^{-1.05} = 8.9125\ldots\%$ when the follower count is 1, according to the model.

The $R^2$ value is extremely low, at just 0.001. That means that only about 0.1% of the variation in log(loves/views) can be attributed to log(follows) in the model. The residual plot is also not uniformly distributed, so there may be some other more complex association between the parameters that is outside the scope of the AP Statistics course.

In conclusion, there is no clear association between a Scratcher's popularity (measured by follower count) and the ratio between the number of loves and views they receive. The $R^2$ value of 0.1% means that only about 0.1% of the variation in log(loves/views) is accounted for by the model, which is statistically insignificant. There also appears to be little association between the two parameters, visually.