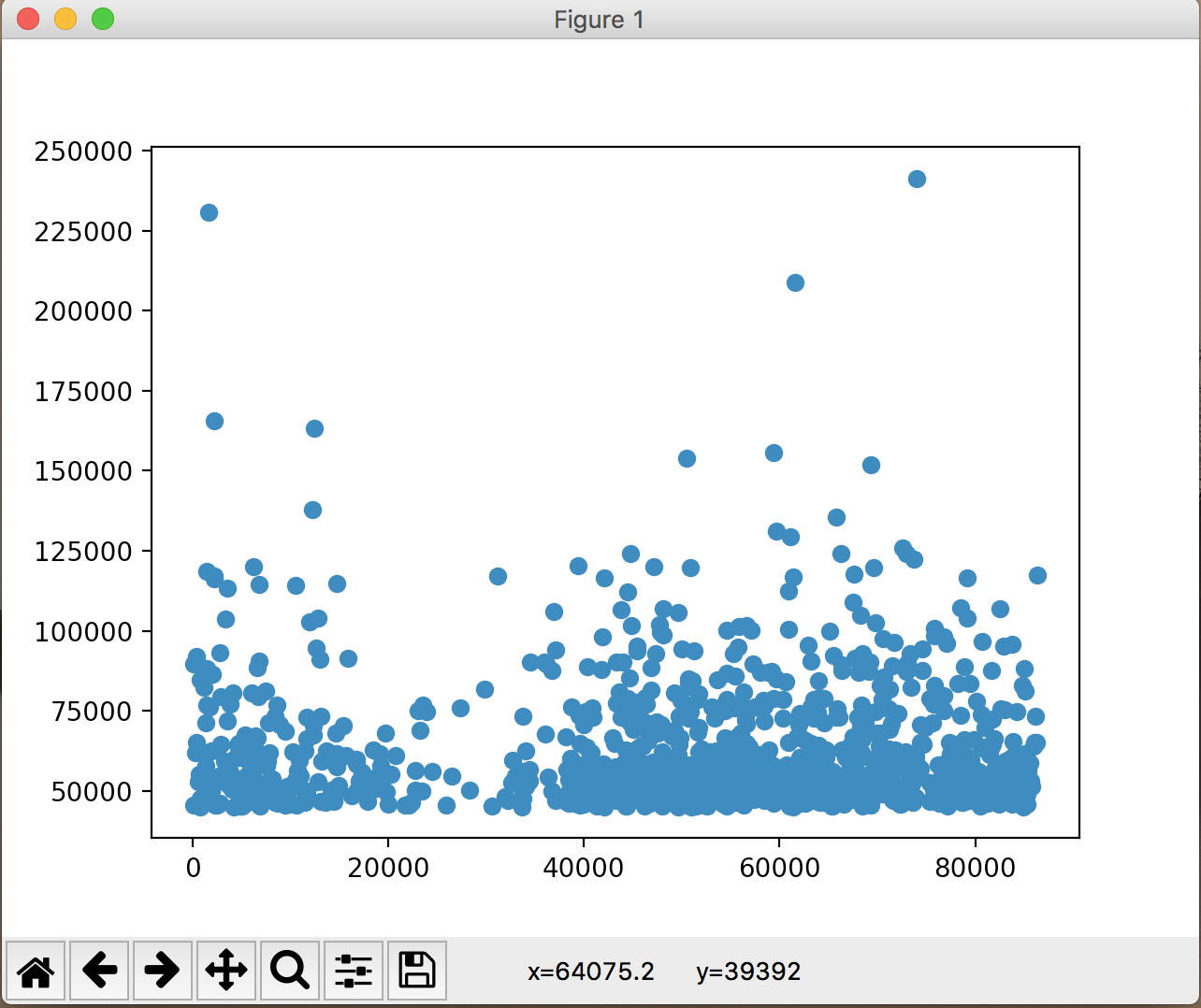
A brief but complete description of your project ( see the Gallery page for examples from last year) and all participants

3. Your deliverable, final version of your project.

For our project, we hypothesized that there would be linear trend between the length of a post in r/Jokes, and the number of up votes where the shorter the post, the more up votes there would be because none of us liked reading long jokes on r/Jokes. We also thought there might be some kind of parabolic trend between the time of day posted and up votes, however, after looking at that data we deemed there was no trend and focused solely on the the relation between up votes and length.

Fig 1: the trend between time of day and up votes, there wasn’t any strong connection here, but there was one gap at t=30000.

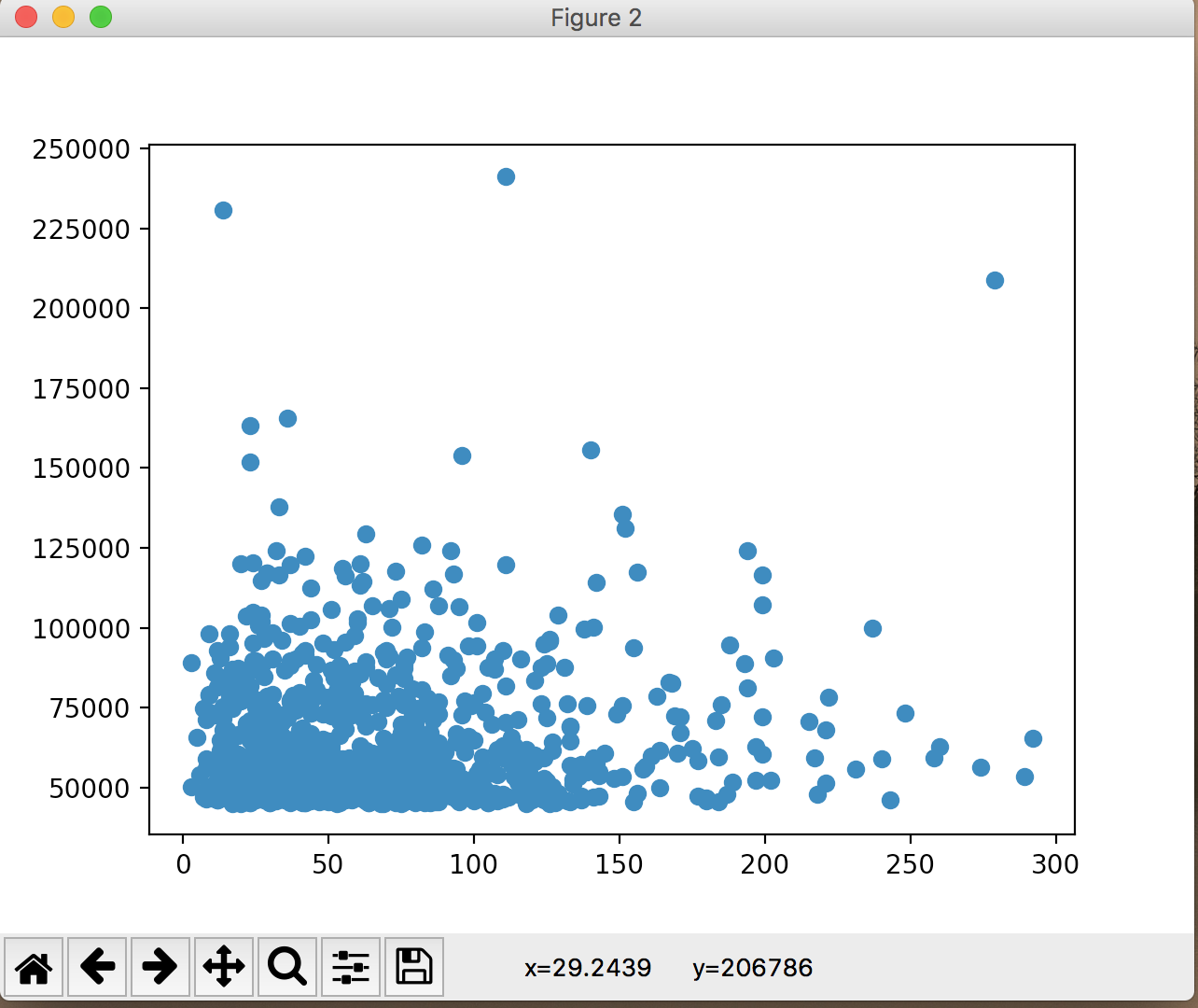
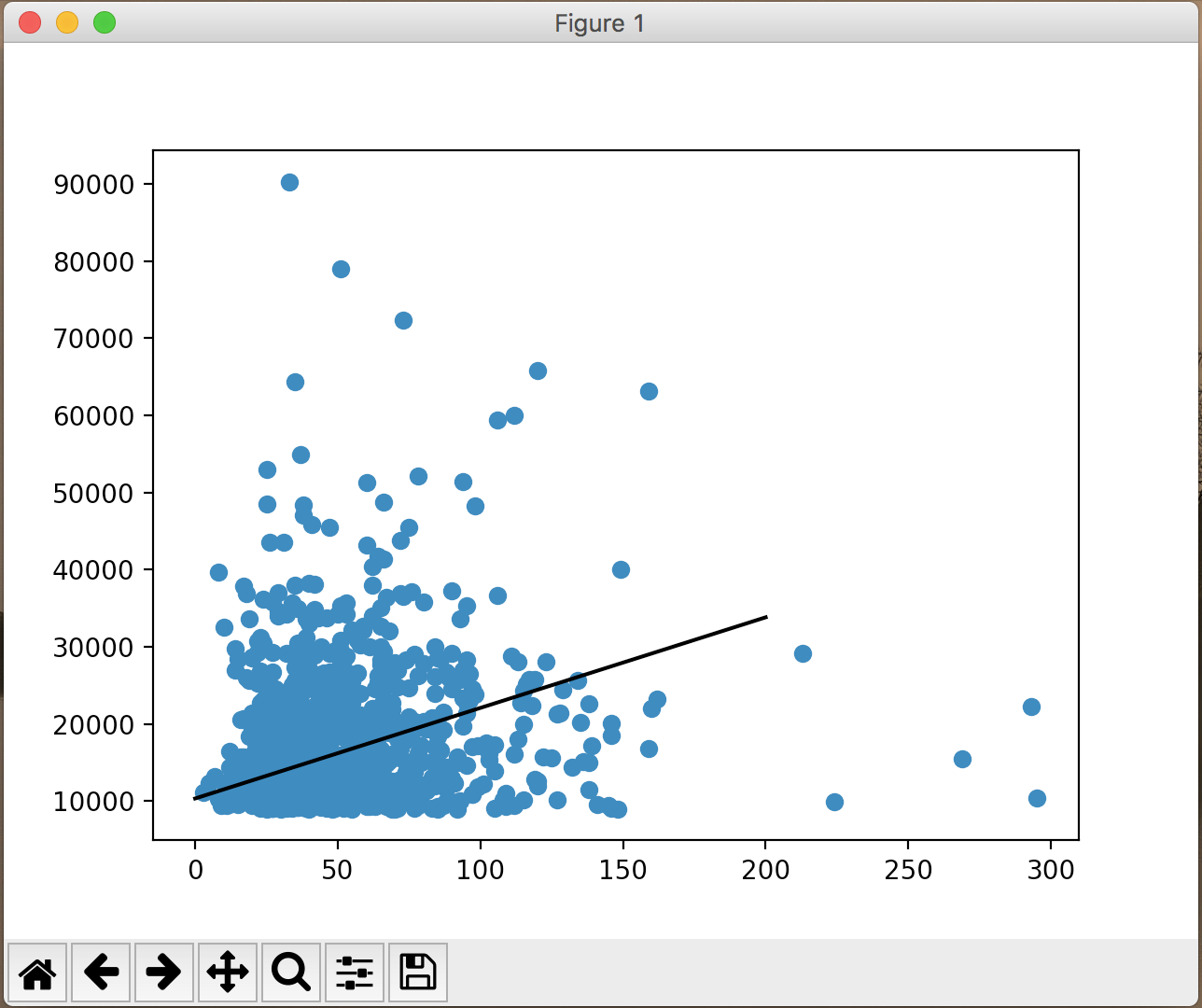


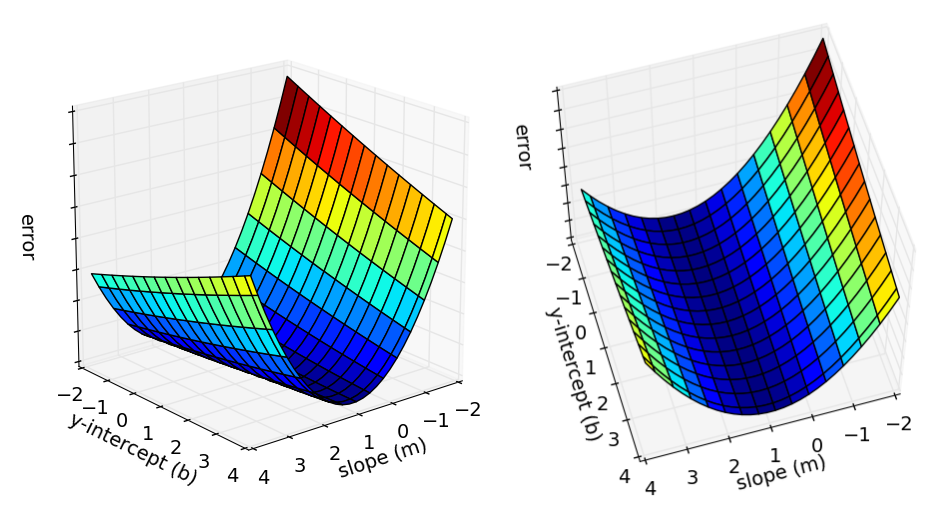
Fig 2: Relationship between length of post and up votes, slight linear trend

After performing some machine learning – specifically linear regression – on the other set of data, we surprisingly found that the trend line did the opposite of what we expected!



As you can see, the trend line goes up, indicating that as the post lengths get longer, the amount of up votes decreases. However, this line may be a result of the fact that we have very few data points for posts with a length of greater than 150 characters. This suggests that the actual trend line might be a parabola, and reaches some peak at somewhere between 100 and 200 characters, and then decreases later on.

The calculus involved was during the machine learning. The algorithm that I used, minimizes the squared error by using partial differentiation on the error function. My error graph was a 3d graph that would probably look something like this (this is just a generic graph for most linear regression models, but mine was probably different due to the fact that my data probably had many different best fit lines).



I partially differentiated the squared error with respect to each data point which just made the function 2 times the original function.