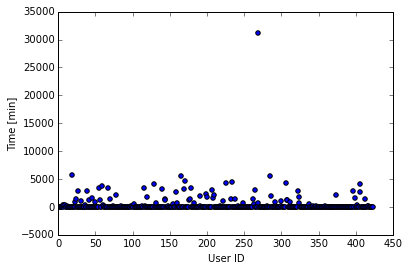
**Discovery Analytics Challenge I Submission**

**Stewart DeSoto**

We are provided with a dataset consisting of n=423 unique user IDs, and two corresponding fields, START TIME and END TIME, which encode the beginning and ending times (in seconds since midnight on Jan. 1, 1970) for users completing an online activity. The desired statistical result is the best estimate of the average Time on Task (ToT) value, which will be provided to classroom teachers and enable them to schedule the task as a homework assignment appropriately.

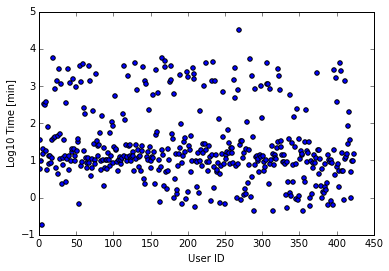
At first sight, the problem may seem trivial, as one can simply subtract the starting times from the ending times to find the ToT for each user, and calculate an average. However, the challenge in the data set is that users complete the task “in the wild”. In this case, some people may submit their response immediately after receipt, likely without careful consideration of the problem and without actually doing the required work. In addition, other users may look at the question, and then proceed to do other activities for hours, or even days, before returning to the web site to submit their response. It is required then, to filter out invalid data points, of which there may be many.

As a first look at the data, consider the plot shown in figure 1. Here we show the raw data on a linear plot. We can make several observations, including the fact that most users have ToT times close to zero on this scale, while a small fraction range up to 5000 minutes (i.e., several days), and one extreme outlier had a ToT of close to 1 month! Clearly, we need to rescale the data in order to understand the detailed structure of the task times, which are currently clustered on the x-axis in this plot.



**Figure 1. Raw ToT times for all n=423 users.**

Given that there is such a large dynamic range of ToT times, ranging from 0.18 min (about 12 seconds) to 31,103 min (21.6 days), or about 5 orders of magnitude, it is helpful to consider taking the logarithm of the task times to better visualize the shorter recorded times.



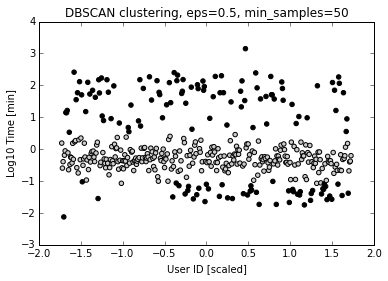
**Figure 2. Semi-log plot of ToT times vs. User ID**

In figure 2, we are now able to easily visualize the distribution of the entire range of recorded times. It is intriguing that there is a heavy concentration of times near Log10(ToT) = 1, corresponding to about 10 minutes. There also appears to be a substantial number (albeit at a much lower density) of ToT times above a horizontal line near Log10(ToT) = 2 (about 100 minutes). These apparently result from users who participated in non-task focused activities between the START TIME and END TIME values. There is also a region of reported user times below about Log10(ToT) = 0.5 (about 3 minutes). Given that there are relatively few such reported times, and also considering the very sparse region separating them from the heavy concentration at Log10(ToT) = 1, it is probable that these very short times result from users who did not fully participate in the task and perhaps clicked the submit button prematurely.

In summary, there appears to be a trimodal distribution of user times, with very short values from users who likely didn’t adequately complete the task, a majority of users with ostensibly valid task times, and a group of users with very long recorded ToT times, who may or may not have completed the task, but for whom we have no reliable estimate of the actual time they worked on the task.

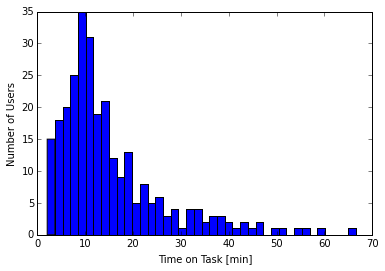
Since the goal of this investigation is to report best estimate of the mean time taken by people to actually and fully complete the task, it seems desirable to find the average of the high density core of user times centered around Log10(ToT) = 1. One way of doing this would be to visually select cut-off times to separate the valid times in the middle. Good choices would perhaps be Log10(T) = 0.5 and Log10(ToT) = 2.0. But a more reliable way to automate this would be to try various clustering algorithms which seek to use unsupervised learning methods to identify groups of close or “similar” data points, as defined by some distance metric.

In this case, a classic clustering algorithm like k-means may not be appropriate since the data visually present wide, adjacent clusters, rather than the tight and compact clusters expected by a centroid-based metric. Instead, the defining characteristic of our clusters are the densities of the various groupings. The outlying top and bottom clusters (corresponding to excessively long and short task times, respectively) have somewhat low density, while the center region of valid times has a high density of data points. An appropriate choice of clustering techniques here is the DBSCAN (Density Based Spatial Clustering of Applications with Noise.) We were, in fact, able to easily separate the data according to the anticipated clusters using this approach with the Scikit Learn package in Python. In many machine learning algorithms, an import pre-processing step is standardizing and scaling data for the various features, to better estimate distance metrics, and not have one feature dominate over others. We performed this scaling step on both features used by our machine learning algorithm. The results are displayed in figure 3.



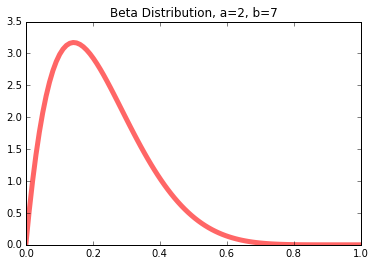
**Figure 3. DBSCAN clustering showing Time on Task assignment to one of two clusters, indicated by the open and closed circles.**

Intriguingly, both groups of points which we considered to be suspect (for being too short, or too long to be trusted as valid task-only times) have been classified by DBSCAN to be members of the same cluster. This is consistent with our assumption that the central cluster (open circles) have a greater probability of representing valid ToT times. The number of task times assigned to the high density central group is n\_valid = 280, which is 66% of the entire data set. Following is a histogram showing the distribution of ToT times for users considered by our DBSCAN clustering to have yielded reliable task times.



**Figure 4. Distribution of Time on Task times for the valid subset (n\_valid = 280) as determined by our DBSCAN clustering assignments.**

We have arrived at a set of data that appear to draw from a single probability distribution. The probability density function is clearly not a normal distribution as it not symmetric and instead has positive skew. It is thought that task completion times may often be represented by a beta probability distribution, with a fraction of users taking much longer to complete the assigned task than most.



**Figure 5. Beta Distribution with parameters a = 2, b = 7.**

In figure 5, we show a beta distribution with parameters a and b selected to yield the approximate shape of our empirical ToT distribution. There is indeed a close similarity and it is plausible that the underlying task times do in fact follow a beta distribution.

Finally, we provide the desired result, namely the value and confidence interval for our estimate of the average Time on Task for individuals actively working to complete the task. We will compute this by finding the mean and variance of the n\_valid = 280 data points. We find that the mean of the times for the valid data cluster is , with standard deviation . This corresponds to a standard error of the mean given by .

As our data set is large (n\_valid >> 30), we can use a z-statistic rather than a t-statistic to estimate a 95% confidence interval. We thus determine that the best estimate of the mean time on task is. Note that this is much shorter than the naïve mean of the entire data set of task times, which is 435 minutes, due to the large number of extremely long reported (and considered invalid in our analysis) raw task times.