Can Clustering Algorithms Discover New Star Clusters?

Dan Feldman - 3/25/19

Image Credit: NASA/ESA

Project Goal

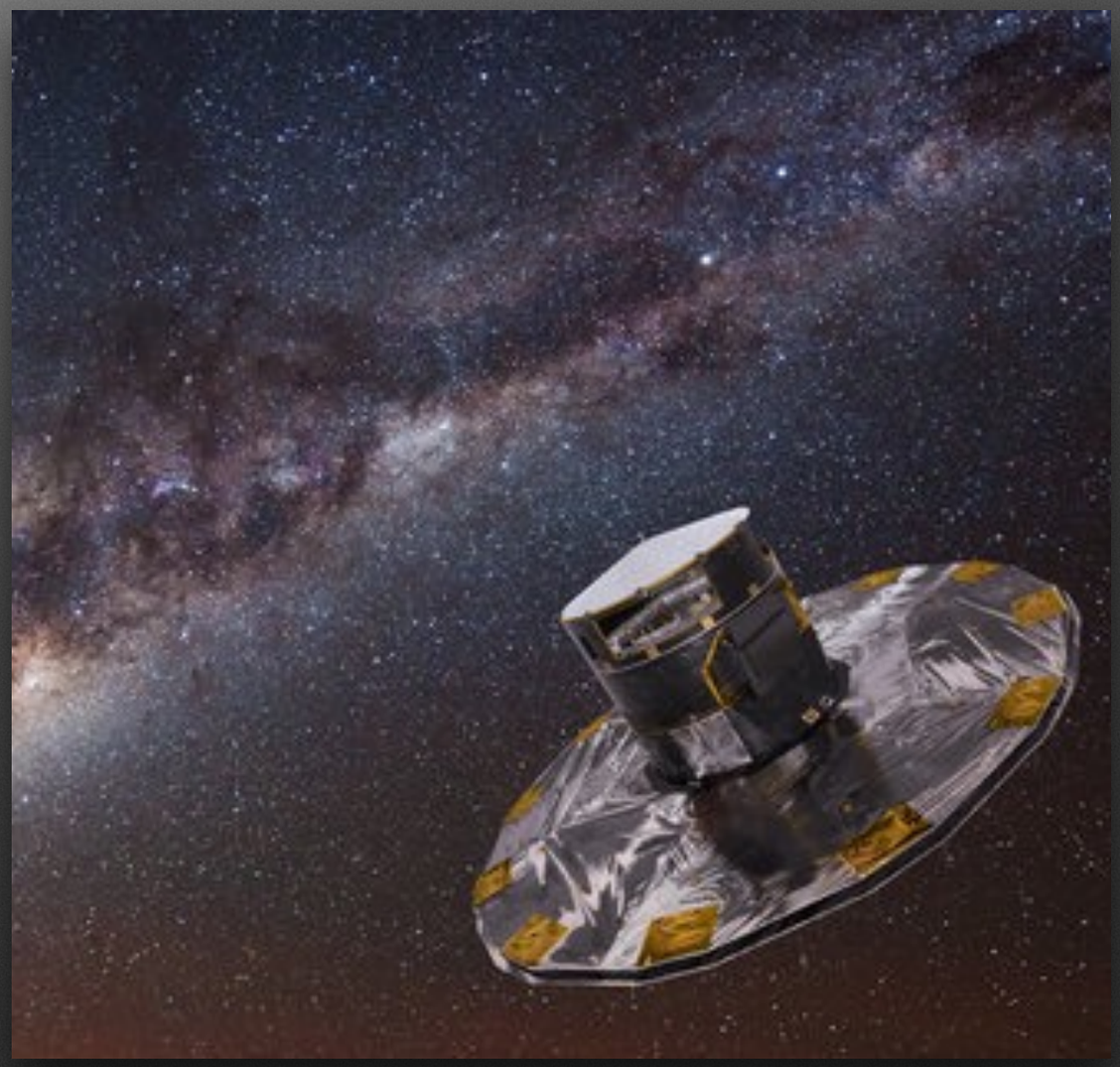
- Determine if realistic star clusters can be found in five-dimensional stellar data using the unsupervised clustering algorithms KMeans and DBSCAN.

- Potential clients for this work are scientific researchers trying to find and study open and globular stellar clusters, which are important for understanding the origins of stellar birth and evolution, galaxy evolution, and planet formation.

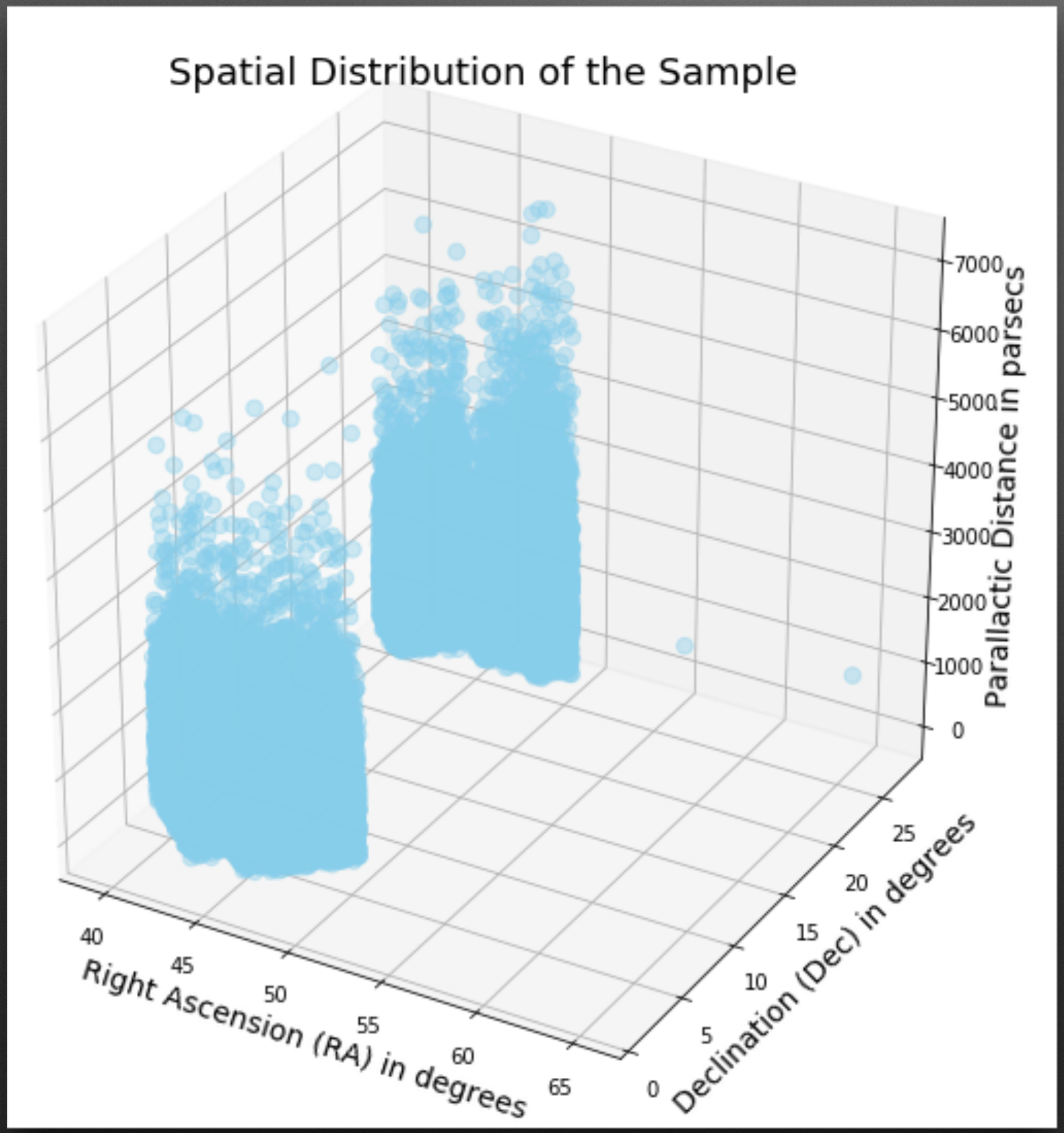
Credit: NASA/ESA

The Dataset: Gaia DR2

- Gaia Data Release 2 (DR2) Source Catalog contains 1.3 billion sources in the galaxy.
- Data includes three spatial dimensions (Right Ascension, Declination, and Parallax), and two velocity dimensions (Proper Motions in RA and Dec directions).
- Smaller subsets contain additional data, like stellar properties and radial velocity.
- Dataset is offered publicly via the ESA's Gaia Archive, using PostgreSQL queries.



Credit: ESA

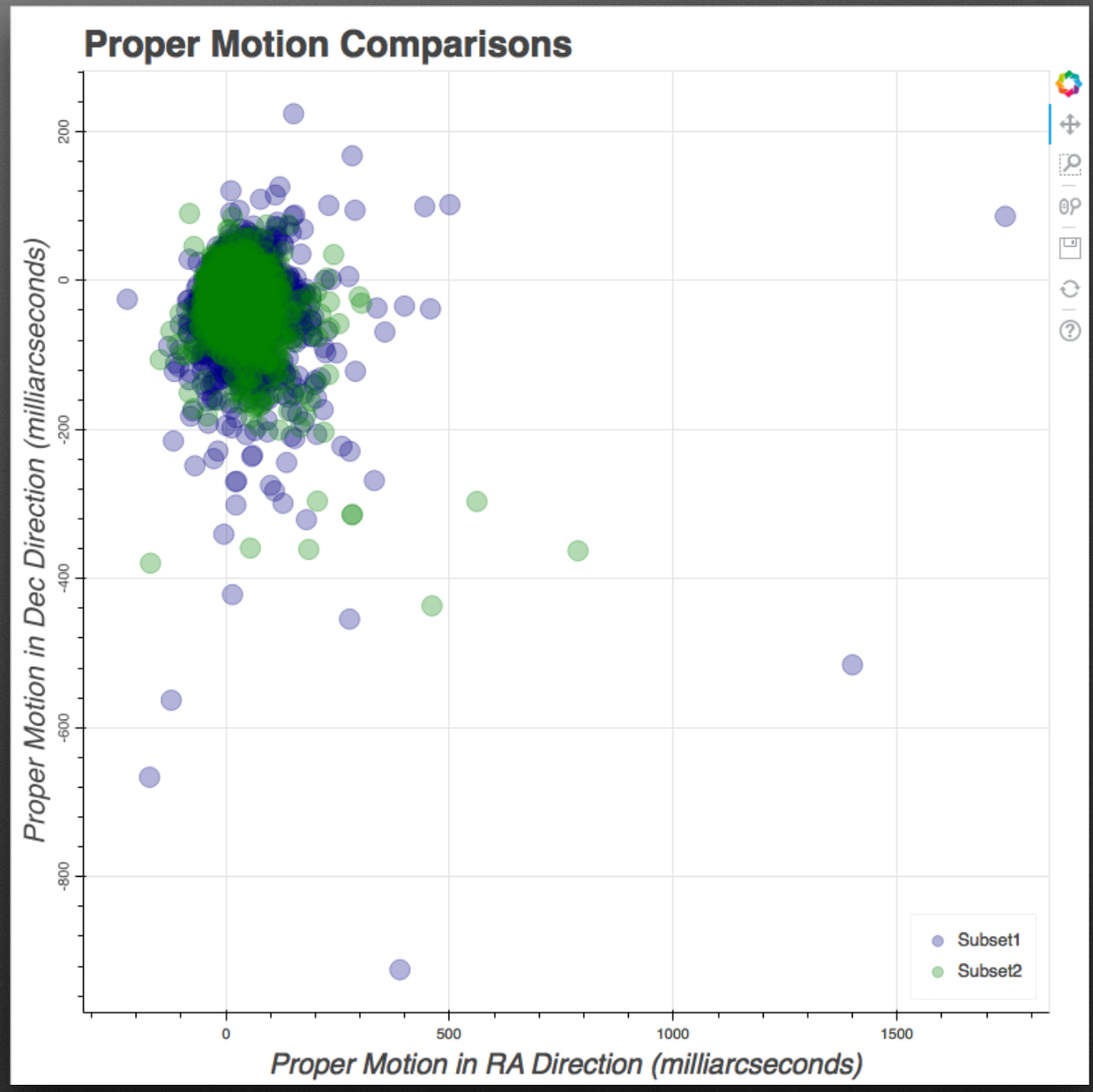


Subsample Details

Due to hardware constraints, I needed to utilize a subset of the DR2 sample. The total number of sources used in this project was 144,806, split into two subsamples in space.

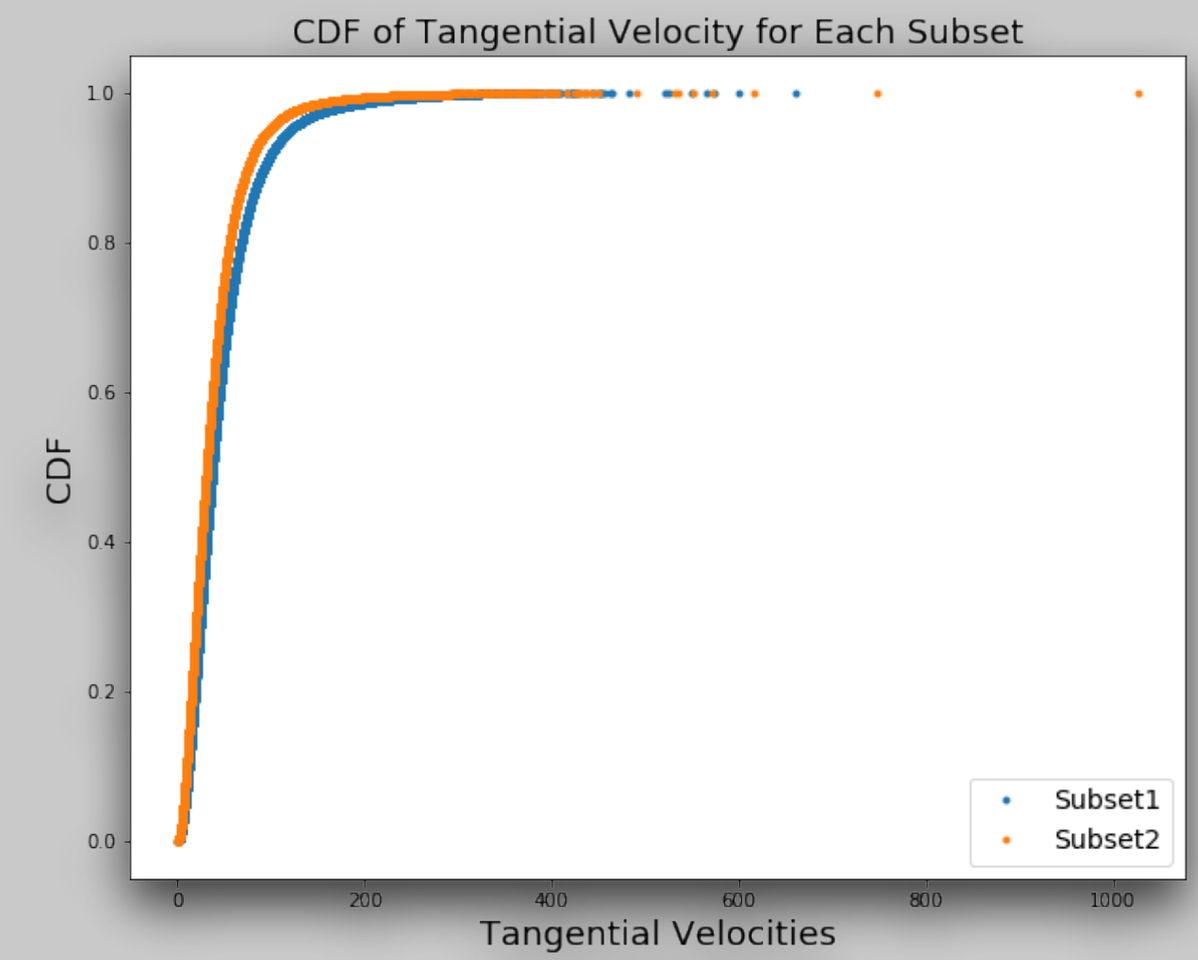
Proper Motion/Velocity Distributions

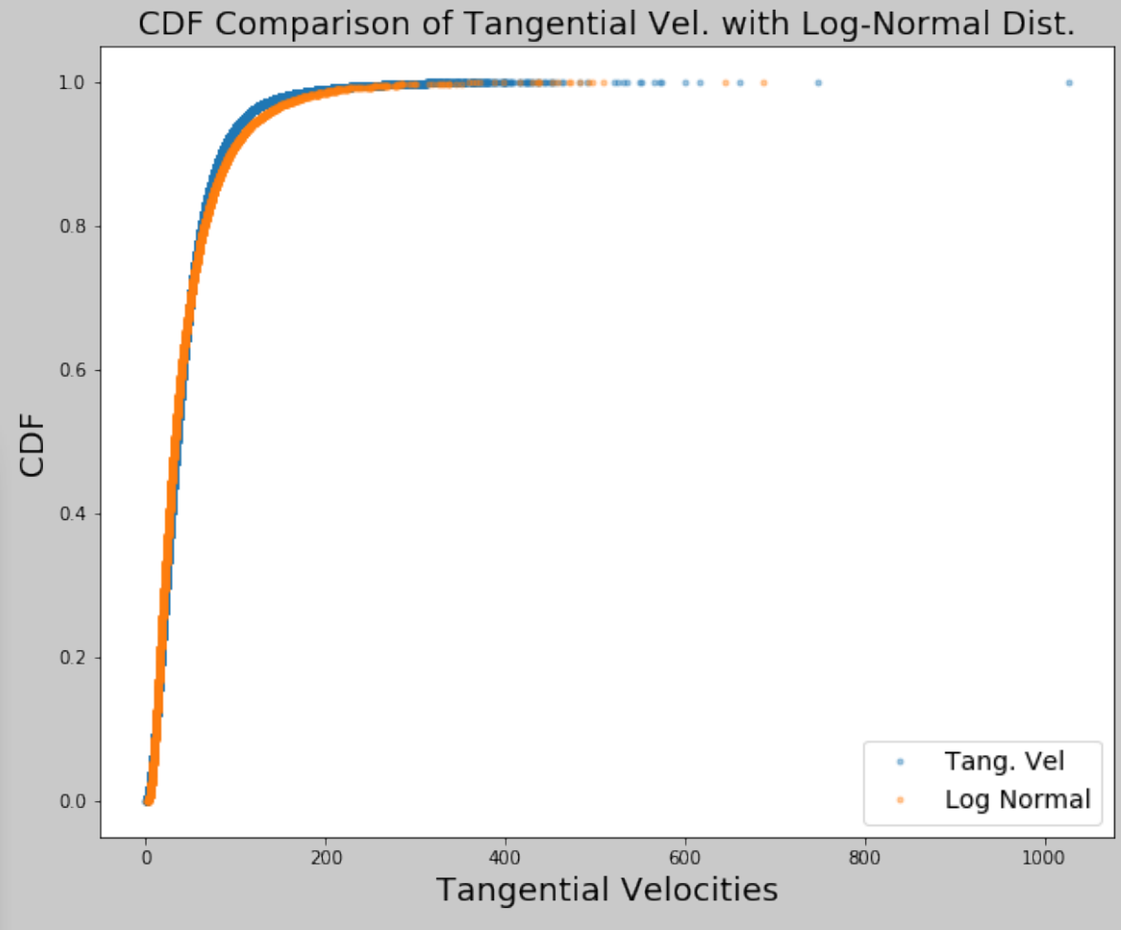
- I looked at the differences between the two subsets within my sample.
- Most of the proper motions were small in each dimension, with some faster moving objects.
- To look at their distributions, I converted these into a one-dimensional “tangential veloctity” value in km/s.

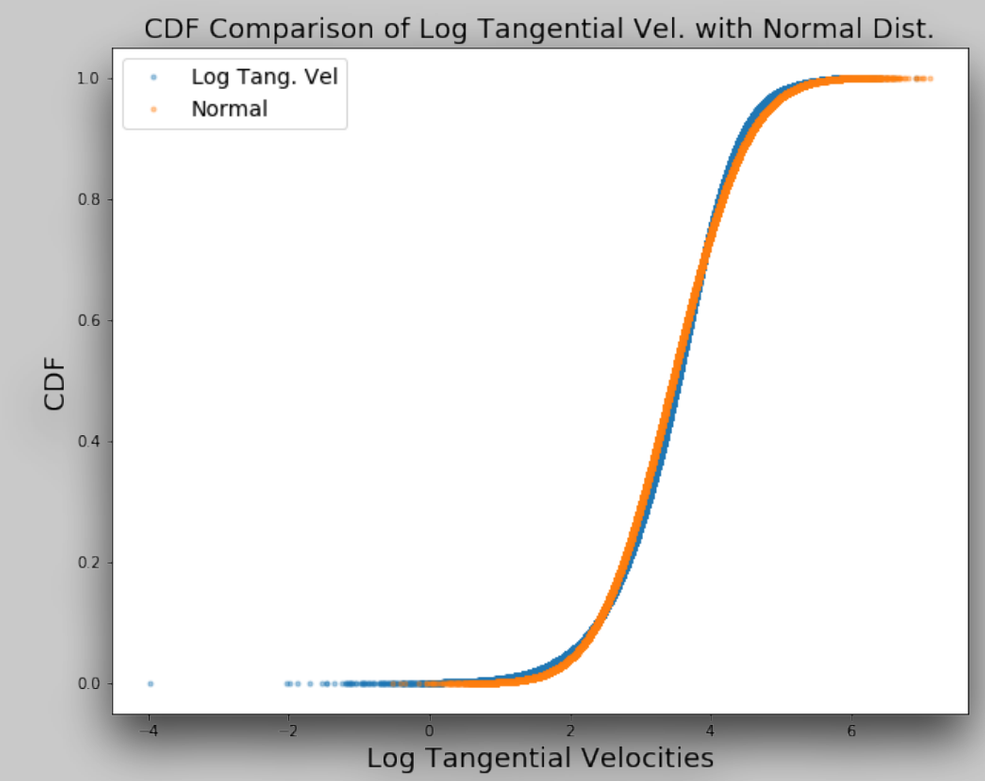


Proper Motions for each subset, in milliarcseconds

Tangential Velocity Distributions: Log-Normal

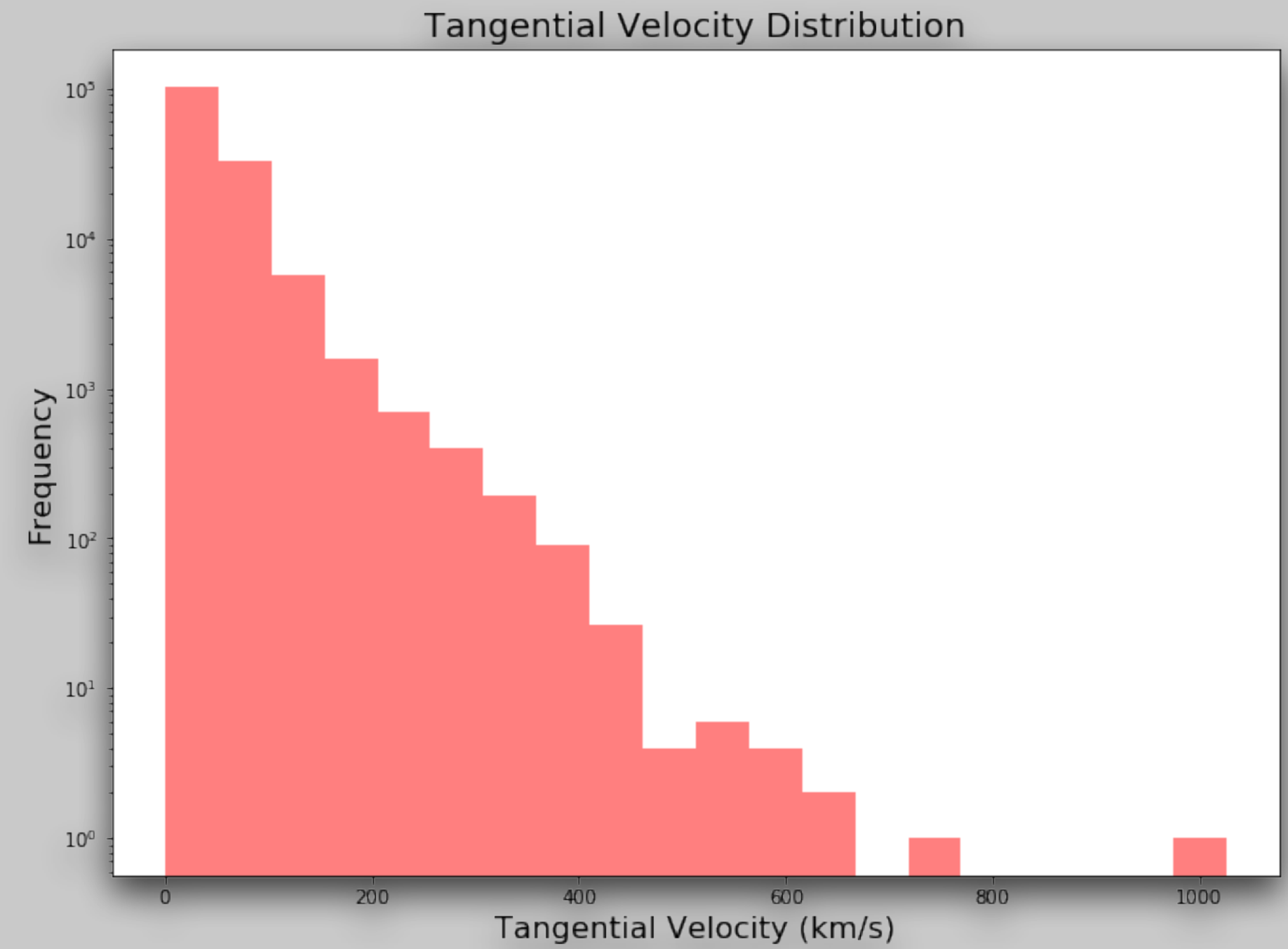






Each subset is log-normal, with statistically significant different μ and σ.

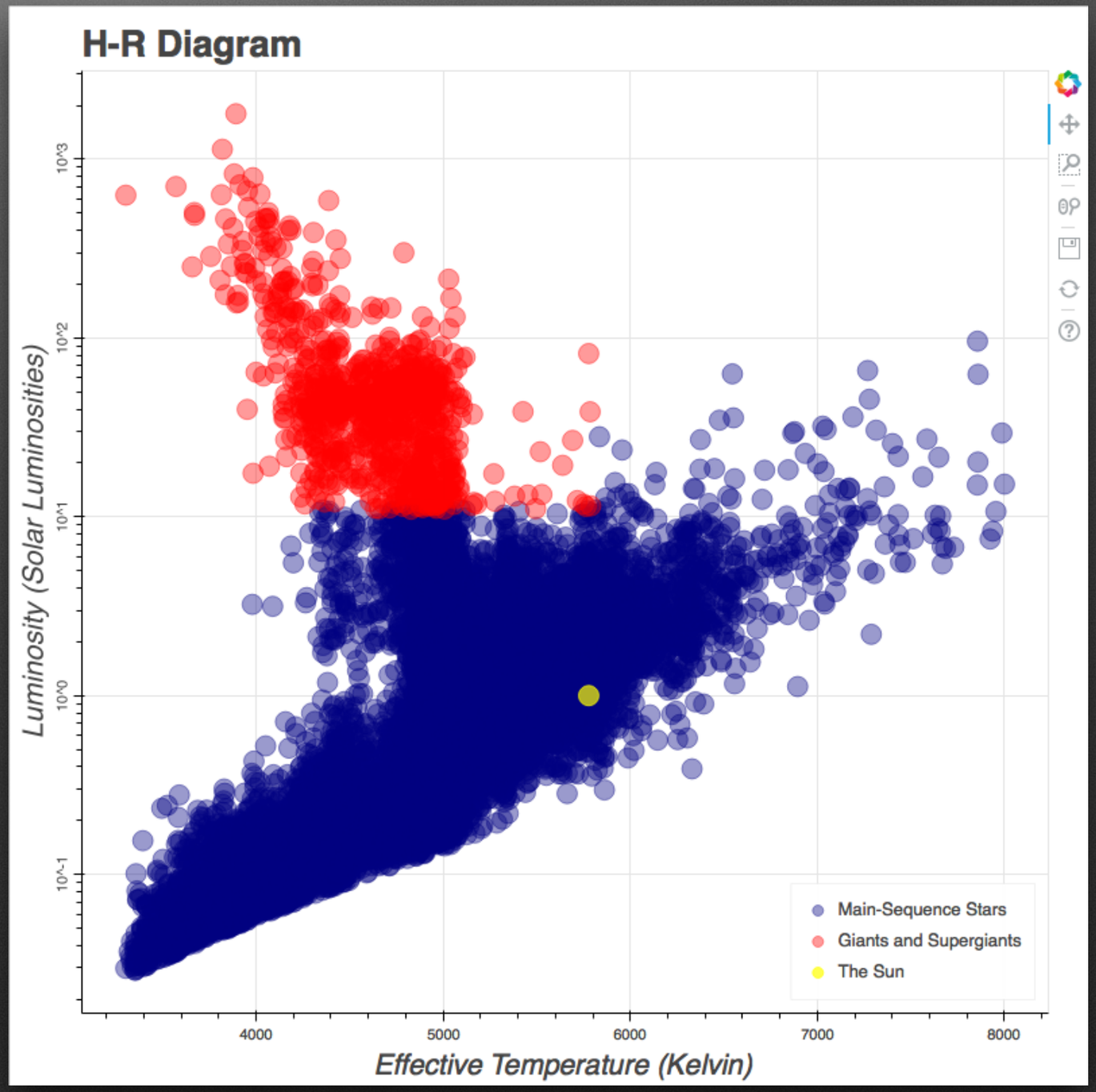
High- and Hyper-velocity Stars



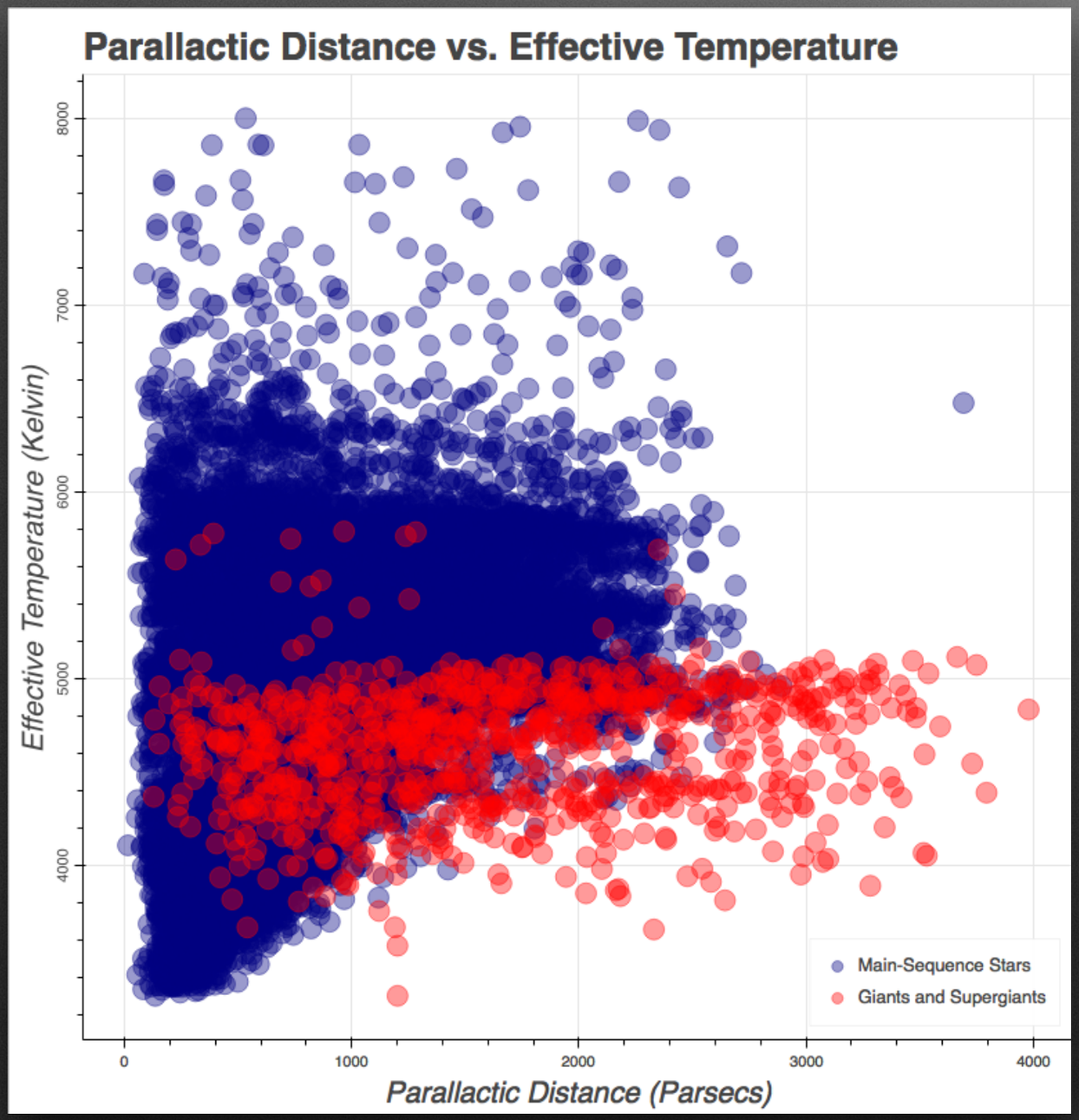
Above is the tangential velocity distribution for my sample. The stars with velocities greater than 400 km/s are high-velocity stars, many of which may be escaping the Milky Way. The highest two are hyper-velocity stars.

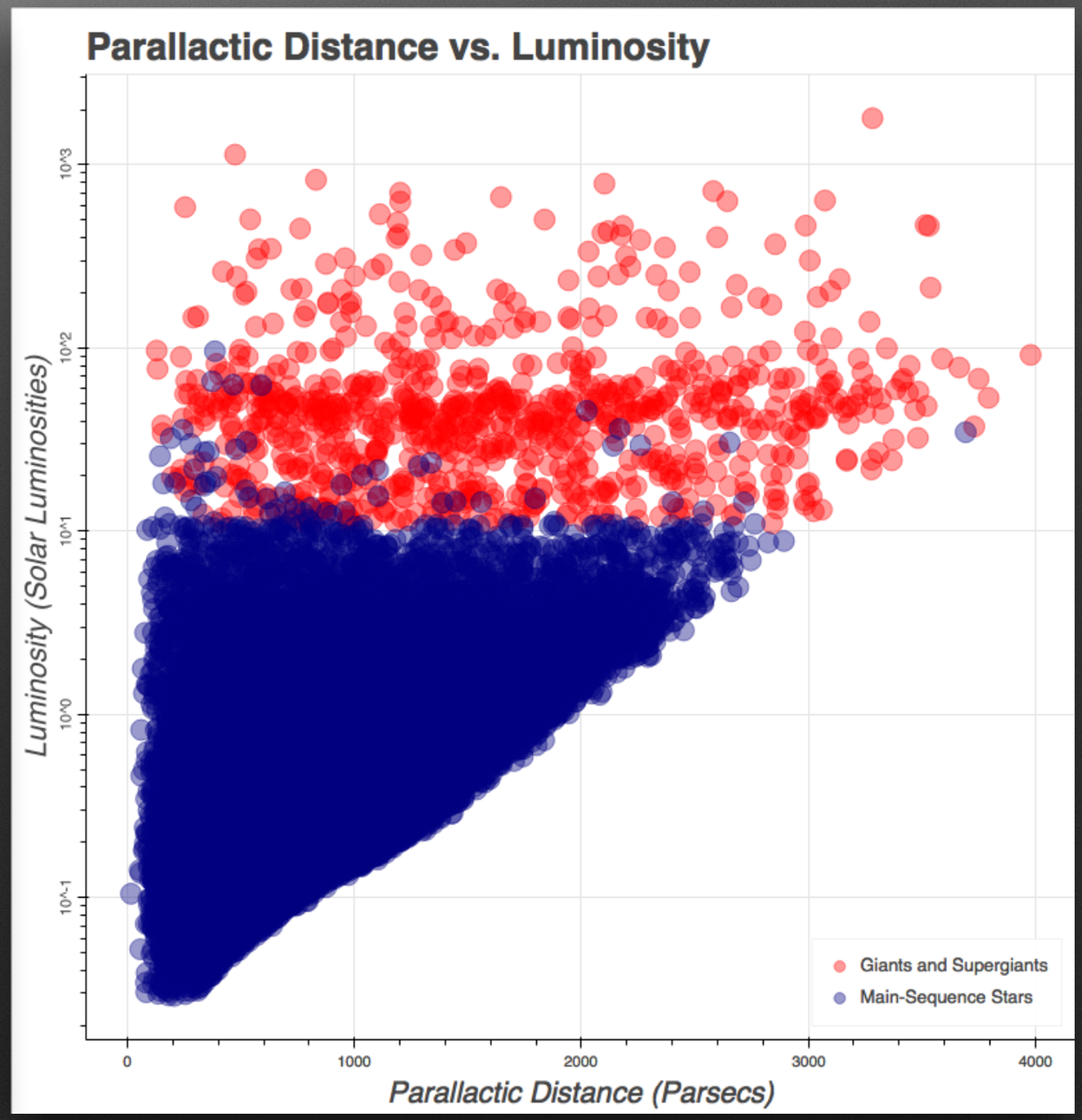
Giant Stars vs. Main-Sequence Stars

- I could look at known stars in the sample, using the 27,605 sources with stellar properties.
- Plotting an H-R Diagram shows that the sample has a number of Giants and Supergiants in the sample, shown in red. Main-sequence (normal) stars are shown in blue, with a yellow dot where the Sun would be. I found 882 giants.
- I looked into how the giants compare with the main-sequence stars using the stars with stellar properties and their tangential velocities.

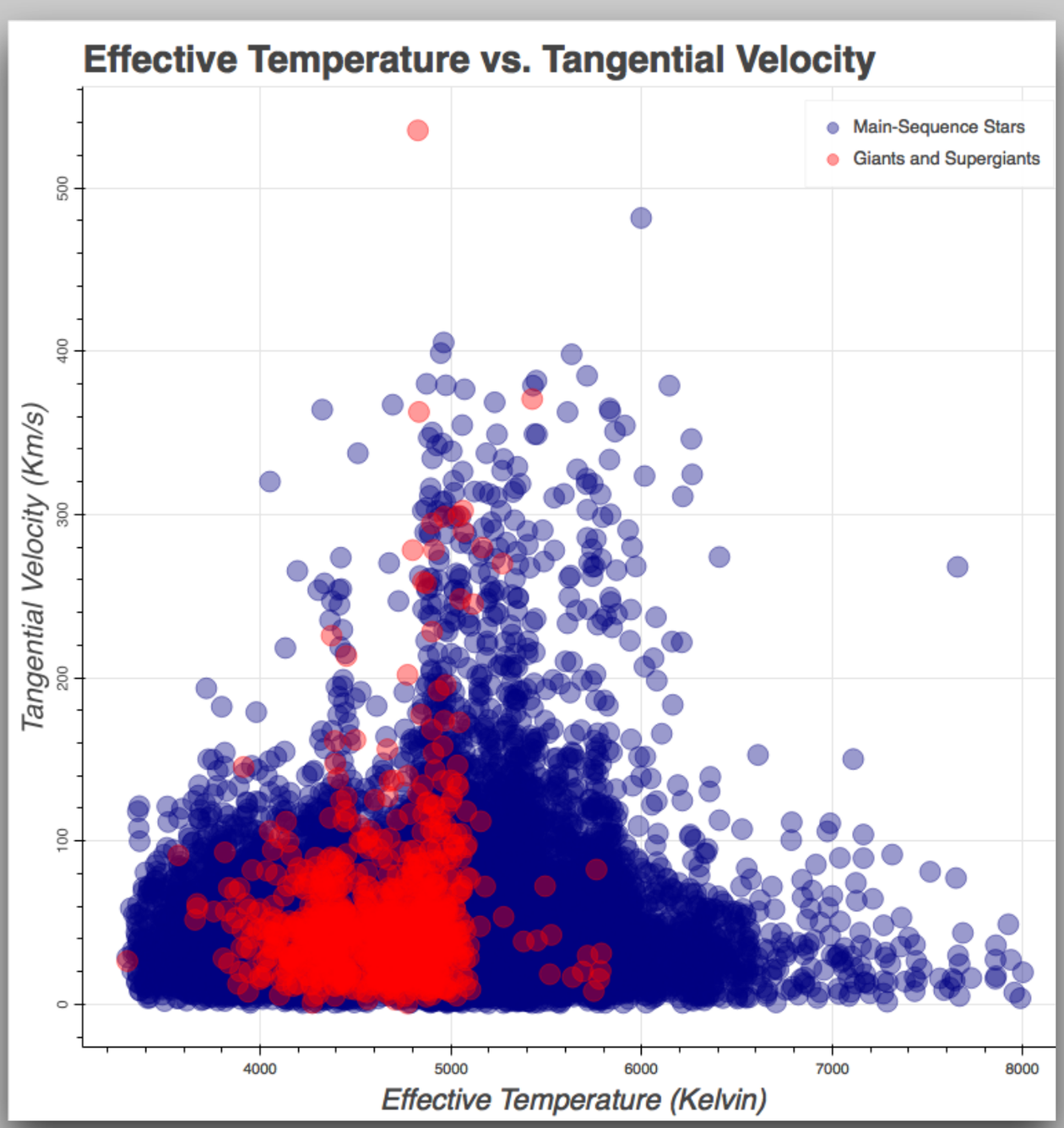


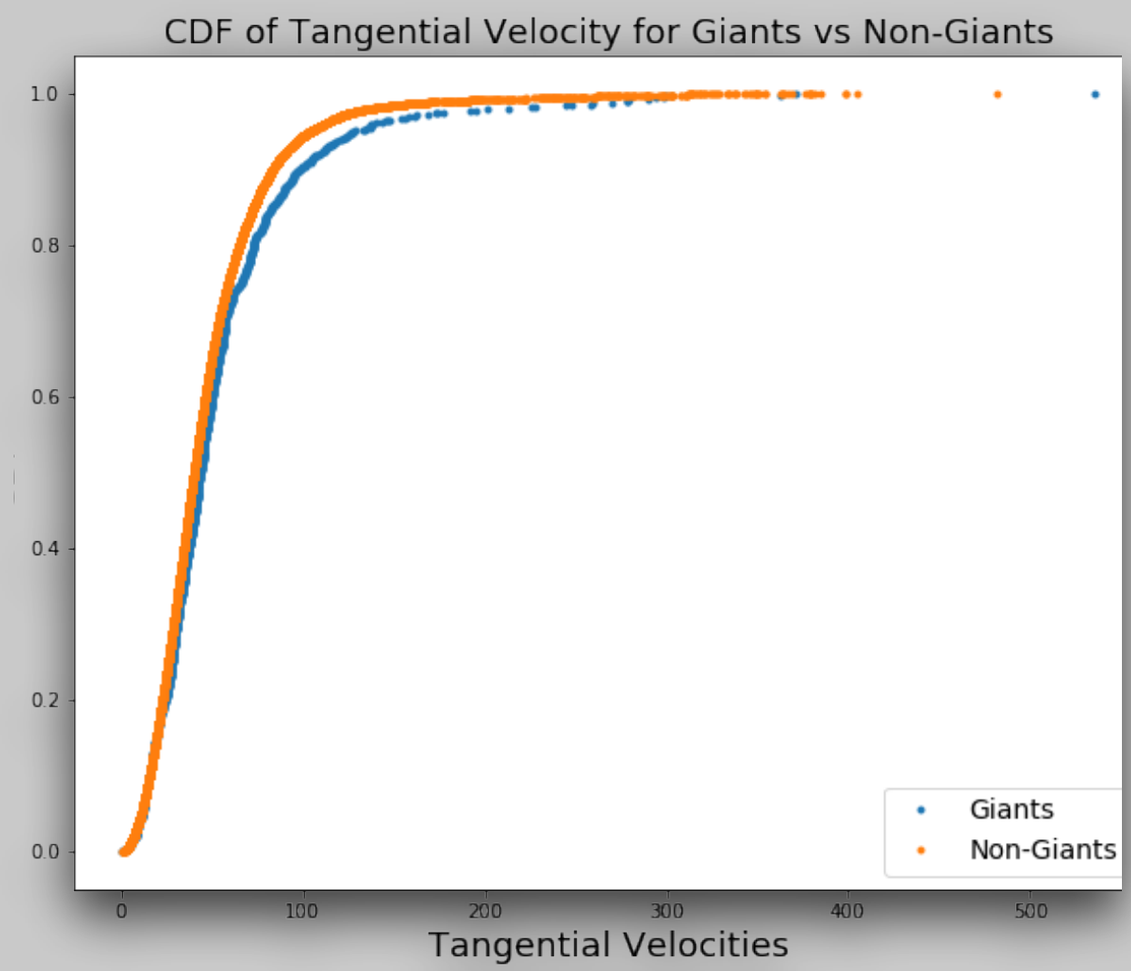
Giants vs. Main-Sequence Stars: Selection Bias towards closer, larger stars





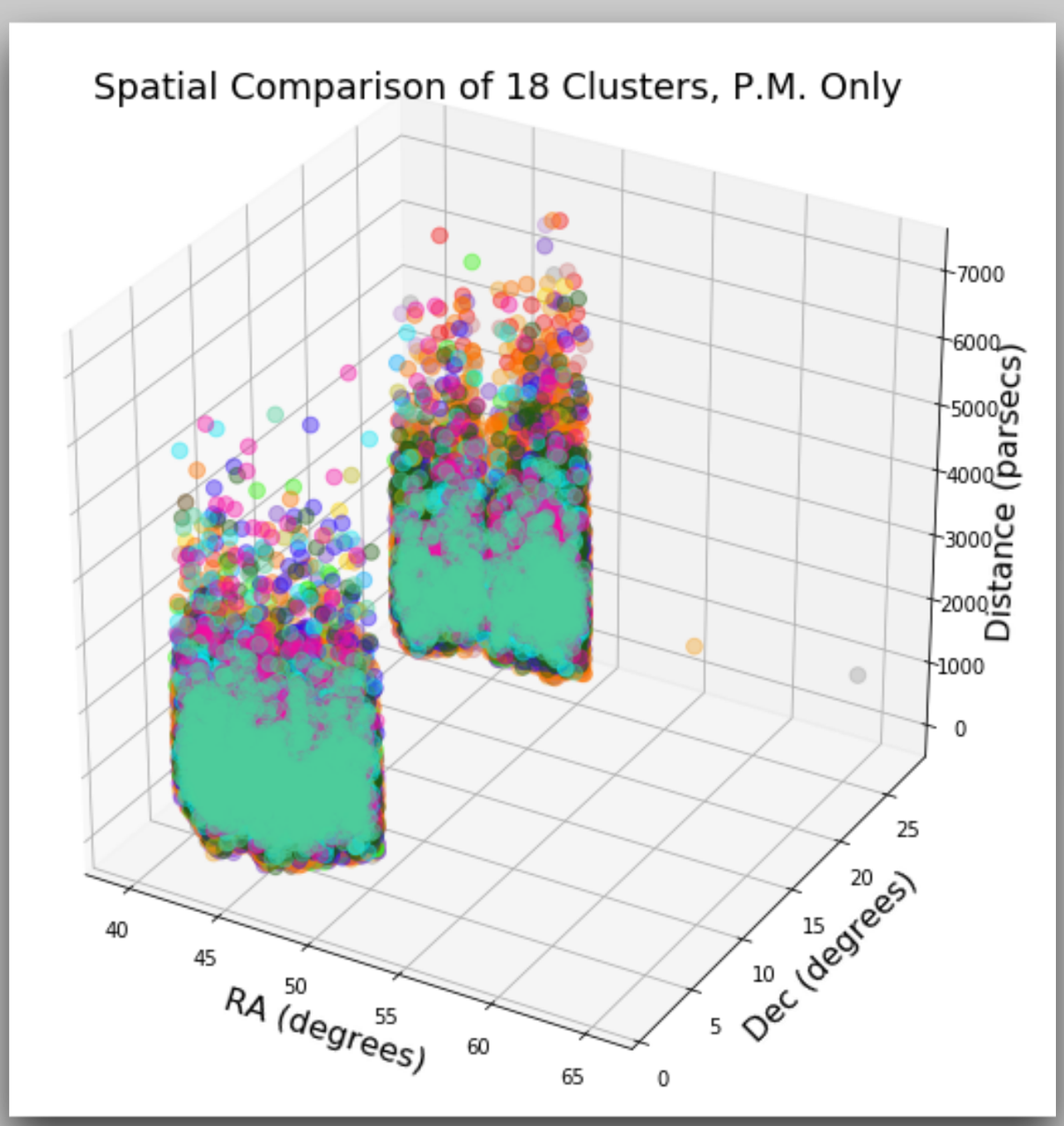
Giants vs. Main-Sequence Stars: Different (Log-Normal) Distributions

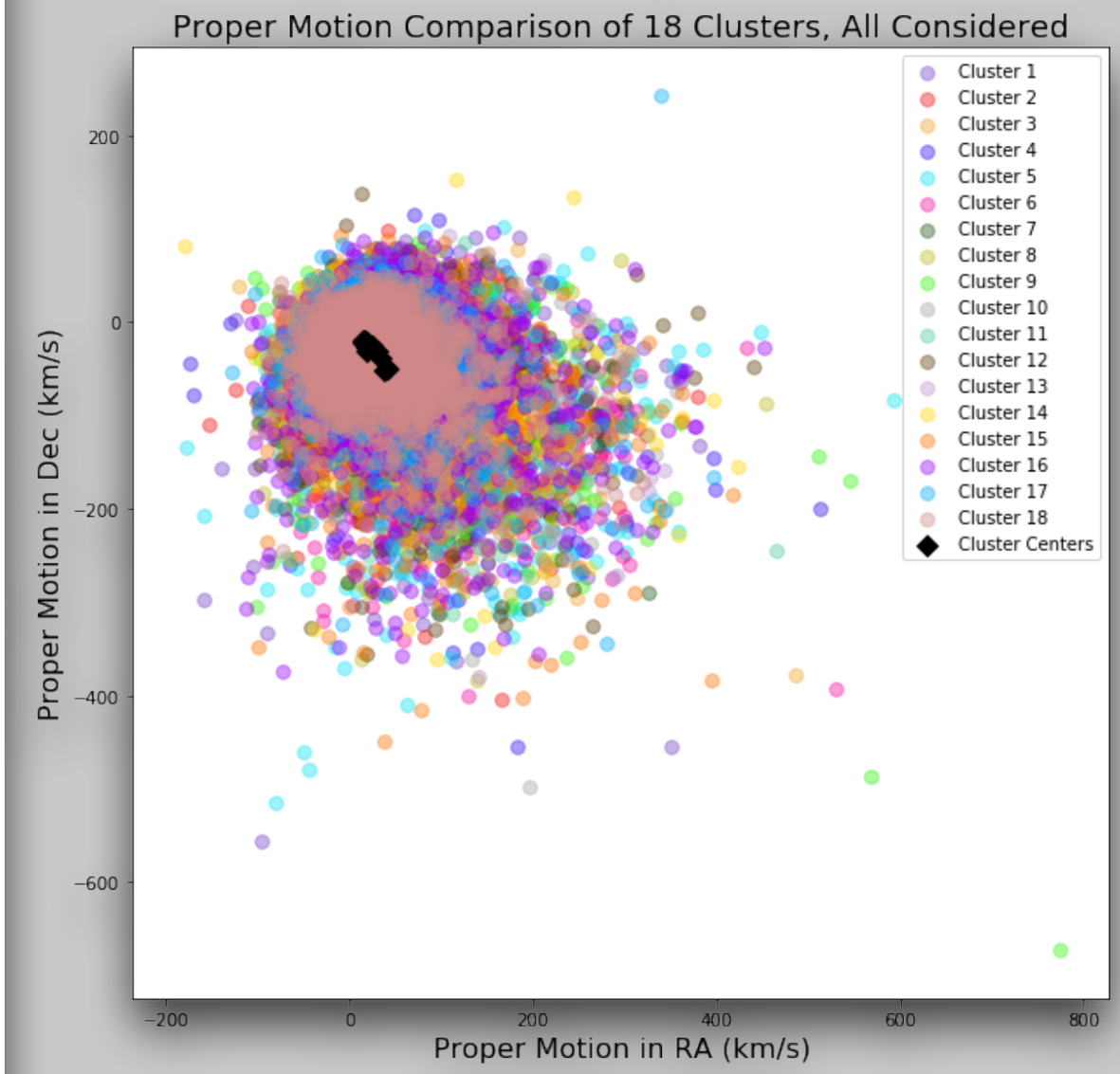




I found that the two tangential velocity distributions were statistically different, similar to the two spatial subsets.

Modeling: KMeans

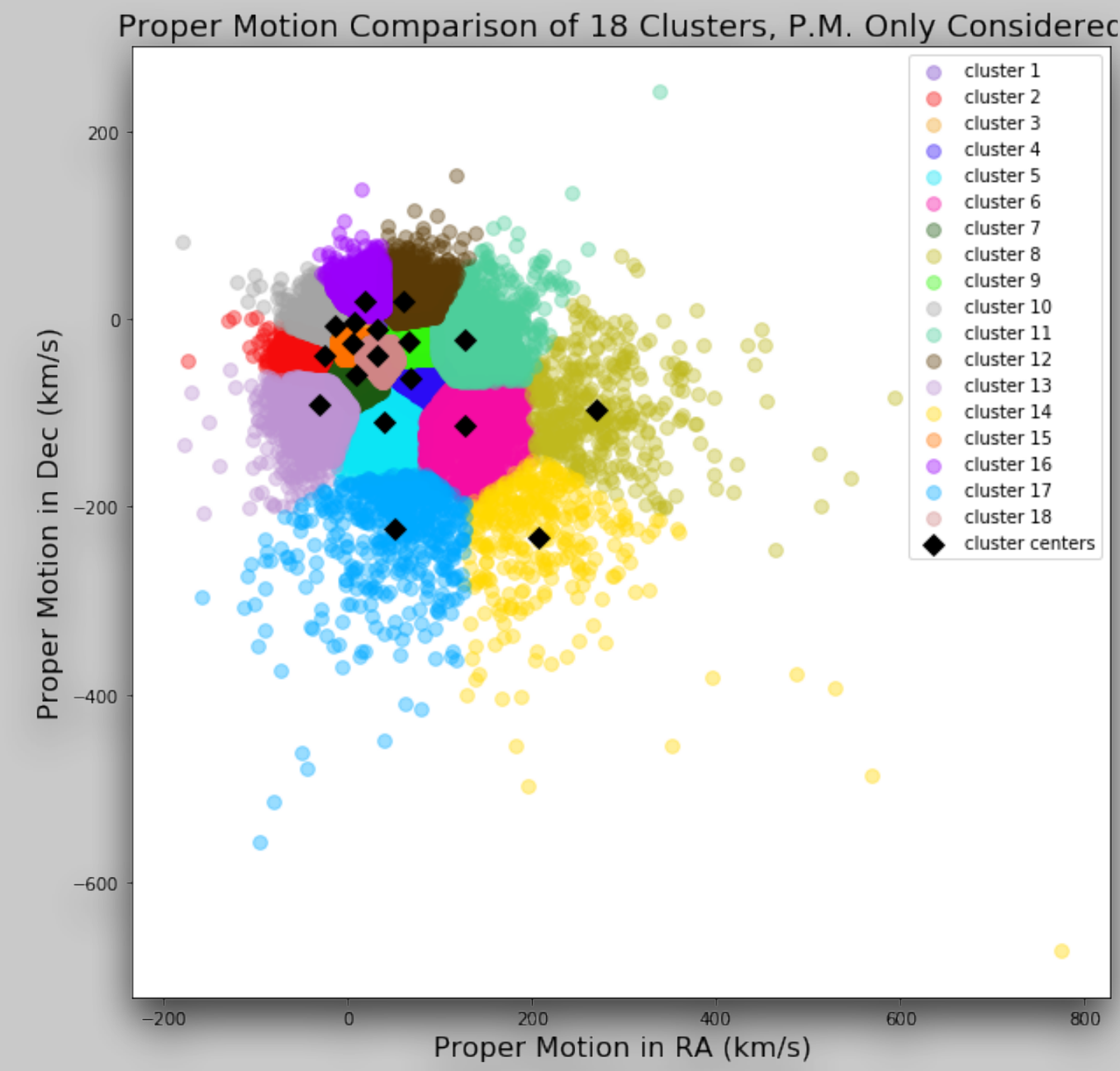




With KMeans, I had issues getting all 5 dimensions to have equal weight, and therefore separating well.

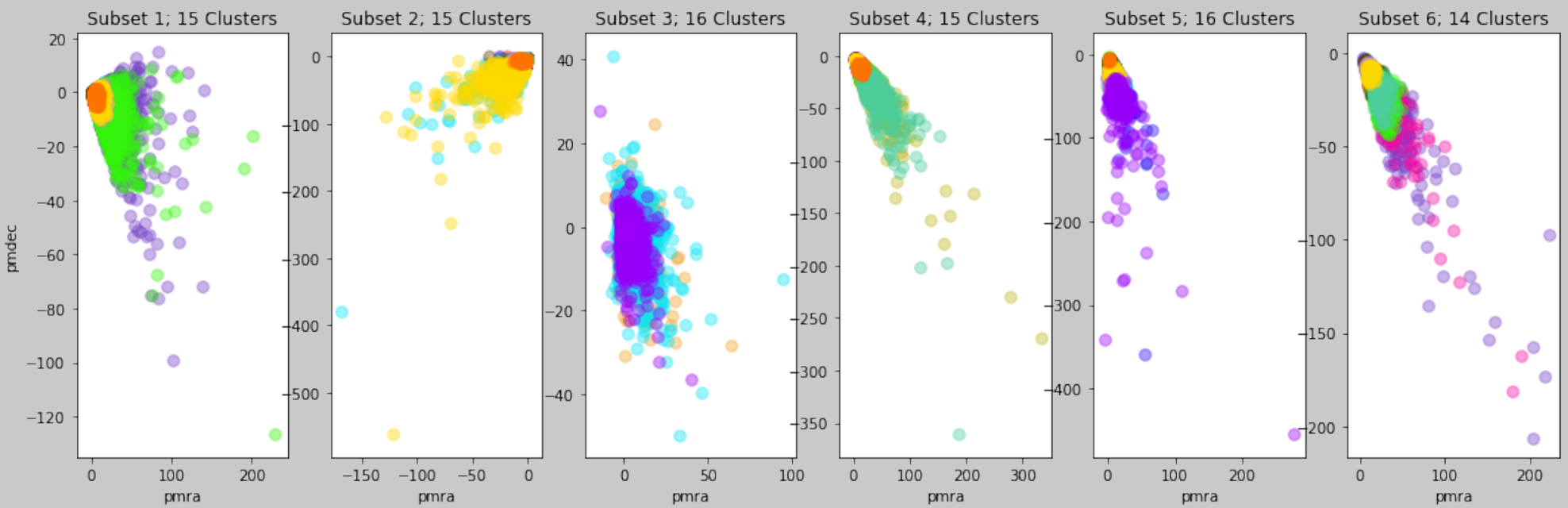
KMeans Step 1: First Pass with Velocity Dimensions Only

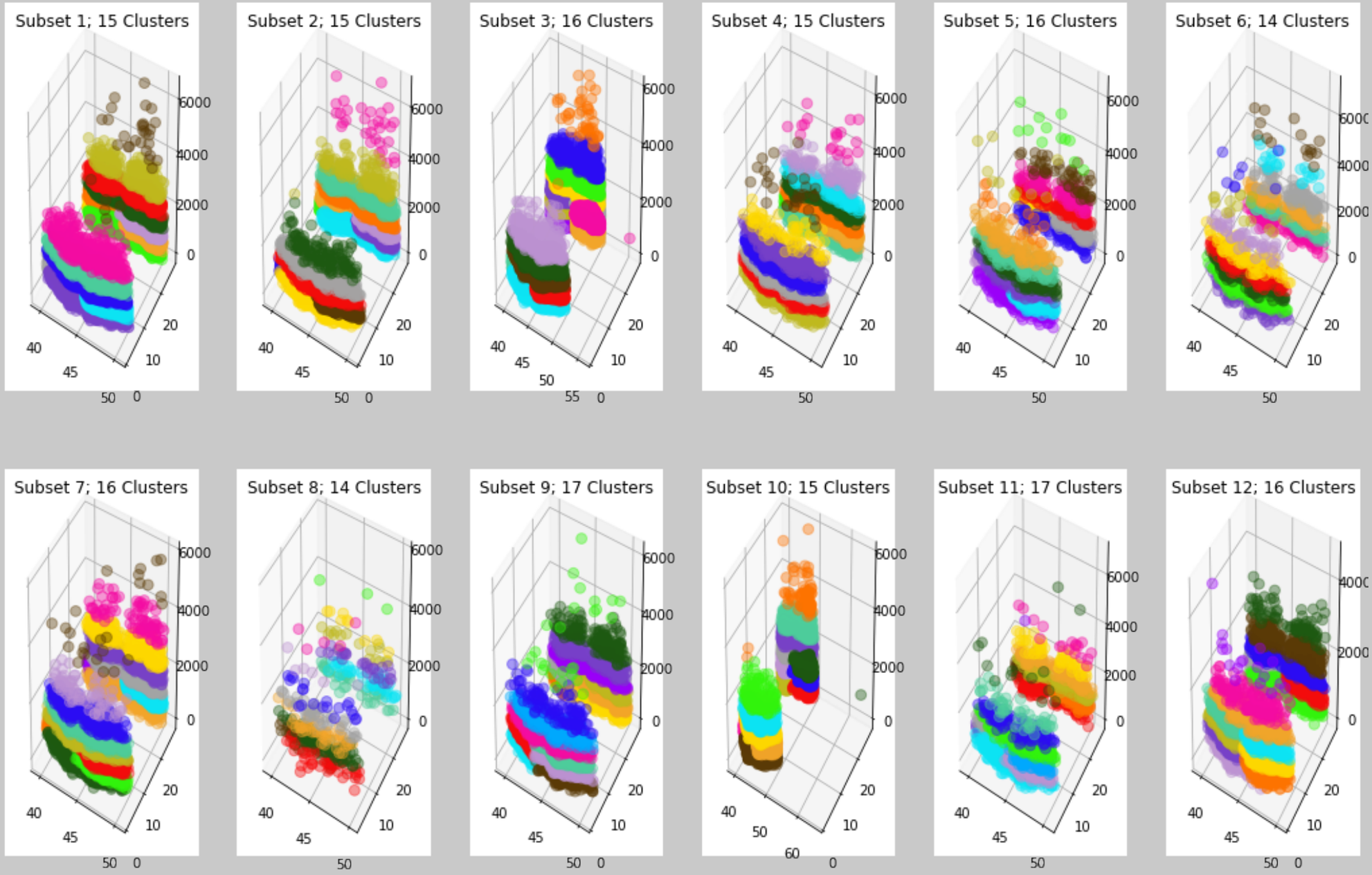
- To fix the issues with KMeans, I decided to break the clustering up into two steps.
- First, run KMeans clustering considering velocity only, using 18 clusters (found via the elbow method).
- Here, the clusters separate well, shown in the adjacent figure.



KMeans Step 2: Second Pass with Spatial Dimensions Only

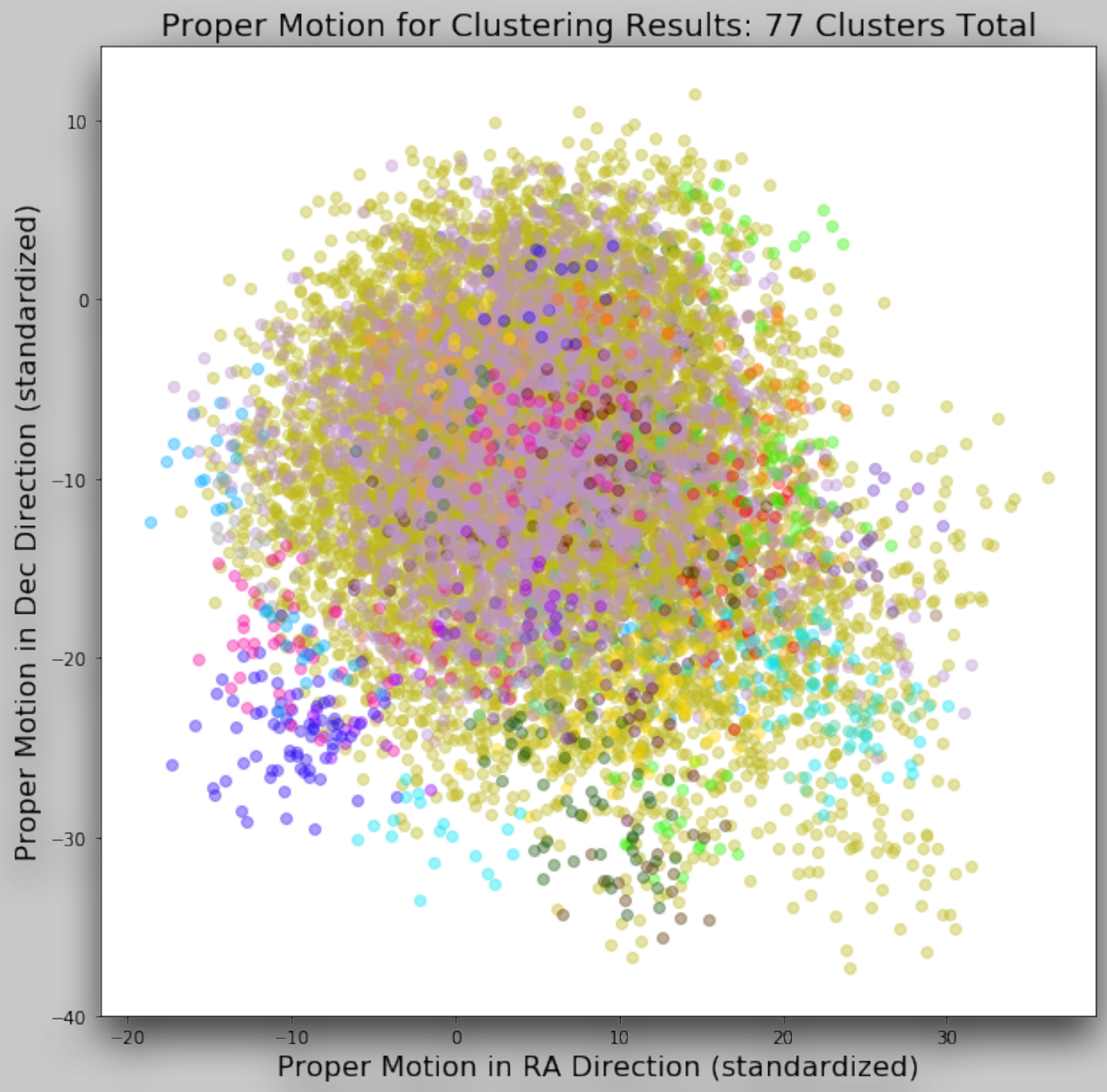
- After the first pass, I took each cluster and ran a second KMeans clustering to get subclusters, using only the 3 spatial dimensions.
- The number of clusters in each second pass was determined with an automated elbow method and was 14-17 depending on the cluster.
- At the end of this modeling process, a total of 281 clusters were found.


Subset 1; 15 Clusters
Subset 2; 15 Clusters
Subset 3; 16 Clusters
Subset 4; 15 Clusters
Subset 5; 16 Clusters
Subset 6; 14 Clusters
Subset 7; 16 Clusters
Subset 8; 14 Clusters
Subset 9; 17 Clusters
Subset 10; 15 Clusters
Subset 11; 17 Clusters
Subset 12; 16 Clusters
Subset 13; 17 Clusters
Subset 14; 15 Clusters
Subset 15; 16 Clusters
Subset 16; 16 Clusters
Subset 17; 15 Clusters
Subset 18; 16 Clusters
pmra
pmdec


Subset 1; 15 Clusters
Subset 2; 15 Clusters
Subset 3; 16 Clusters
Subset 4; 15 Clusters
Subset 5; 16 Clusters
Subset 6; 14 Clusters
Subset 7; 16 Clusters
Subset 8; 14 Clusters
Subset 9; 17 Clusters
Subset 10; 15 Clusters
Subset 11; 17 Clusters
Subset 12; 16 Clusters
Subset 13; 17 Clusters
Subset 14; 15 Clusters
Subset 15; 16 Clusters
Subset 16; 16 Clusters
Subset 17; 15 Clusters
Subset 18; 16 Clusters

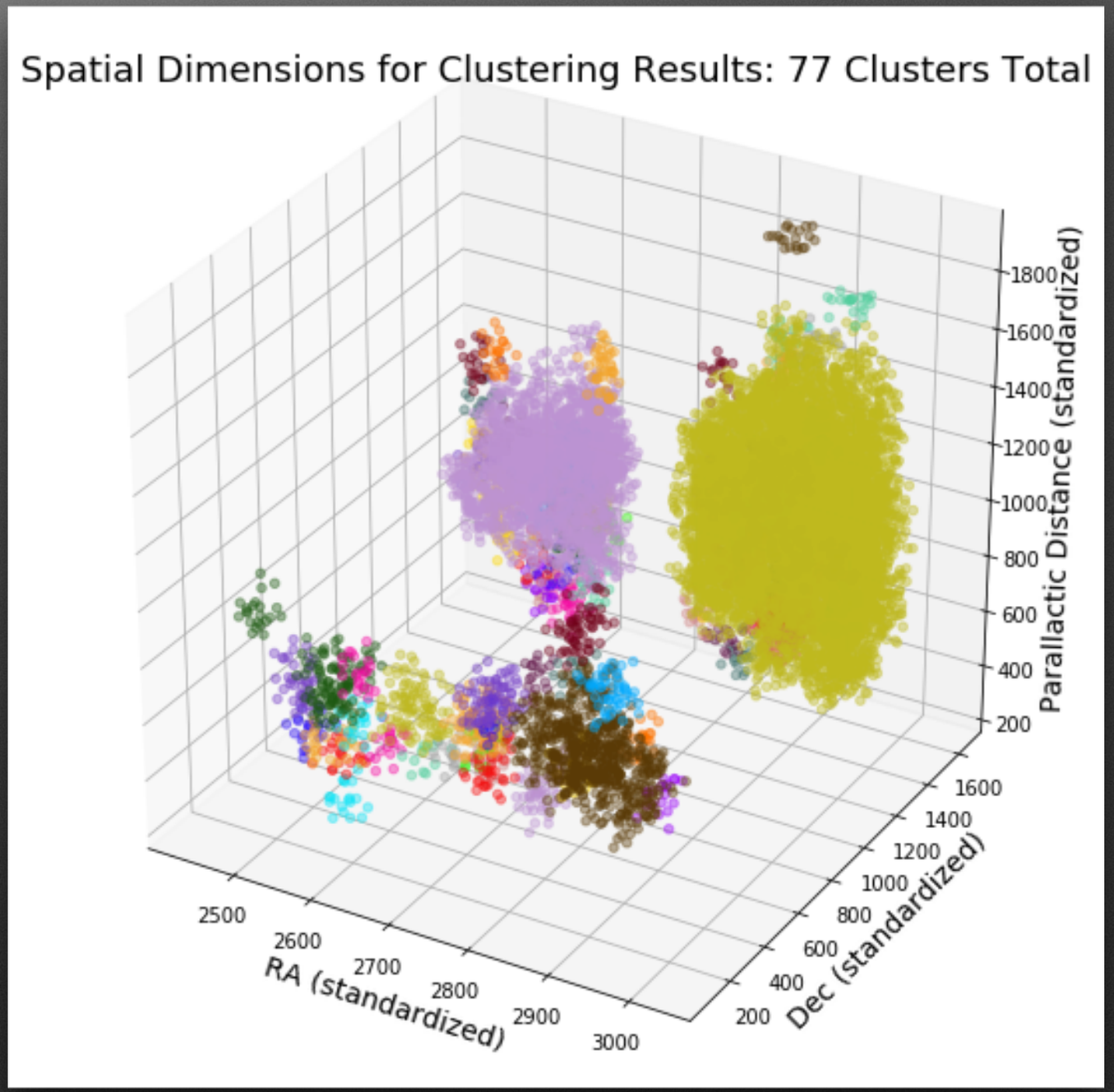
Modeling: DBSCAN

- Unlike KMeans, DBSCAN does not require a priori knowledge of number of clusters.
- Had to standardize all 5 dimensional data in order to get DBSCAN to work properly.
- Generally, DBSCAN was effective at creating smaller clusters, though a few large ones were found. Most stars were lumped into an “outliers” cluster, which was helpful in limiting cluster size.



DBSCAN Results

- A total of 77 clusters were found with DBSCAN, with a 78th being for outliers (not shown in the figure).
- Most clusters were much smaller, and more realistic than for KMeans.
- Hard to say if this is a stable solution, as the subsetting in space introduced fake boundaries in the data; full dataset needed for more tests.



General Results/Trends

- KMeans and DBSCAN both had clusters that were spatially too large, though DBSCAN was much closer to realistic cluster sizes.
- KMeans had clusters with too high of a velocity dispersion, where DBSCAN's clusters were realistic or only slightly too high.
- In both cases, velocity dispersions in RA and Dec directions were highly correlated.
- For KMeans, cluster sizes (number of stars) were anti-correlated with core size (spatial dispersion) and velocity dispersions. For DBSCAN, they were all positively correlated, which is more realistic. This is likely due to KMeans not handling outliers.

Conclusions/Future Work

- DBSCAN was more effective than KMeans at producing realistic clusters, but both algorithms need to be improved upon. Overall, though not perfect, these methods do seem promising for discovering new clusters in the Gaia data, when improved with future work.

- For KMeans, I need to find a way to deal with outliers (same as identifying field stars).

- For DBSCAN, I need to test with full dataset and fine-tune hyper-parameters more for more stable results.

- Testing on known clusters could help add additional assessment to whether or not these algorithms can effectively find clusters.

Thank You For Listening

Credit: NASA APOD