Chris Huang, Sarah Gonzalez, Ava Doyle

Spring 2020 Research

Disk Star Fitting

Abstract:

We wish to map stellar density across the Milky Way disk using data collected from stellar surveys such as SDD and PanStarrs. The goal of this project is to create a program which when given stars as a function of distance we can find a best fit for stars as a function of distance when we account for uncertainty and completeness. The stars we give it will be 2.5 degree by 5 degree or 5 degree by 5 degree “lines” that point to the galactic anti-center.

The Data:

The majority of data collection was handled by Sarah G, and Ava D. This is an excerpt from their summer paper discussing the data.

Observational star data was obtained from SDSS Data Release 14. For each star data point, the apparent magnitude, extinction, magnitude error, and galactic coordinates were recorded. More specifically, the magnitude, extinction, and error were extracted for three color filters: ultraviolet, green, and red. The query requesting this data was processed through the online CasJobs personal-database collector. Stars were selected with galactic longitude between 5◦< l <210◦and galactic latitude 10◦<|b|<30◦. Additional color conditions were applied; 0.1<(g−r)0<0.3 color was selected to isolate main sequence turn-off (MSTO) stars. Isolated MSTO stars are less likely to have stars move in and out of the color selection based on high extinction values. The (u−g)0>0.4 color cut eliminates quasars and other extragalactic bodies. With this observational star data, a scatter plot of the galactic longitude versus latitude was created to find the regions where the most stars were located. The plot identified eleven main vertical bars of data, with some disorganized data surrounding them. The longitudinal locations of these vertical bars were found by isolating the data from each bar, finding the minimum and maximum longitudes from this data, then centering a 2.5◦ section between the minimum and maximum. The disorganized surroundings were removed for more consistent data processing. The vertical bar centered at l= 10◦ is only present in the galactic north.

The sweeps of data in SDSS were conducted in 2.5◦ wide bars and our pencil beams were chosen to be 2.5◦ in longitude and 5◦ in latitude to match the data distribution. For each pencil beam, the stars within it were corrected for reddening. The extinction value used was taken from the SDSS database. The apparent magnitudes and extinction values used for data processing had the green color filter. These corrected stars were organized in a histogram of apparent magnitude versus the number of stars. Each bin was a half-magnitude wide and the histogram had magnitude limits zero to thirty. The height of each bin was recorded and exported into a CSV file for interpretation by statistical photometric parallax. Observational star data was obtained from Pan-STARRS Data Release 2. The apparent magnitude, magnitude error, and galactic coordinates were recorded for each star. Two magnitude color filters were used: green and red. The SQL query requesting this data was processed through the online CasJobs personal-database collector. Stars were selected with galactic latitude $10^\circ<|b|<30^\circ$. In the galactic north, longitudes $0^\circ< l <250^\circ$ were selected. In the galactic south, longitudes $0^\circ< l <230^\circ$ were selected, in accordance with the scope of Pan-STARRS in this region. Preliminary color conditions were applied: $a-0.2<(g-r)0<0.8$ color was selected to isolate the region around main sequence turn-off (MSTO) stars. Approximately 1.6 billion stars fulfilled these location and color conditions and were pulled for processing. The Pan-STARRS database does not include extinction values. We found these values manually through the dustmaps module of Python, which used the corrected SFD two-dimensional dustmap (Schlegel, Finkbeiner, and Davis, 1998; Schlafly and Finkbeiner, 2011). Once the extinction values were found for each star, we made a more selective color cut to match the MSTO star population from SDSS data: $0.1<(g-r)0<0.3$. Unlike the SDSS data, Pan-STARRS offers continuous data throughout all of the selected regions of the sky. This data was then split into 5° by 5° pencil beams and each pencil beam had a corresponding histogram of apparent magnitudes using the green-filter corrected magnitudes. Similar to the SDSS data, the bin heights of these histograms were formatted for processing in the density mapping program.

The Algorithm:

The algorithm builds of previous work from both myself, Rae Helmreich, Sarah Gonzalez, Ava Doyle, and Jake Weiss. A more thorough examination of how the algorithm works can be read from either my or Rae's older papers.

In order to avoid overfitting to the observed data, an interpolation first happens. The interpolation acts as a linear interpolation. In layman's terms, all it does is add a "data point" in between each pair of data points. These data points take the average value of the neighbors where they were inserted in. This will increase a data set of size n to an interpolated data set of size 2n-1.

Next the model is convolved by a Gaussian. However, the convolution has been changed to a pseudo-convolution, and the Gaussian used has become a special 2-sigma Gaussian, with a special Gaussian for each half magnitude. The 2-sigma Gaussian was used in order to better describe the distribution of stars at each half magnitude. An example Gaussian is shown with the 2-sigma Gaussian equations:

$$F(M_g) = \frac{2}{(\sigma_l+\sigma_r)\sqrt{2\pi}} e^{\frac{-(M_g-4.2)^2}{\sigma_i^2}}$$

Where:

$$\mu = 4.2$$

$$\sigma_i = \begin{cases} \sigma_l = \ .36\ if\ M_g \leq \mu \\ \sigma_r = \frac{\alpha}{1+e^{-(d_{eff}-\beta)}} + \gamma\ if\ M_g > \mu \end{cases}$$

$$\alpha = .52; \beta = 12.0; \gamma = .76$$

Fig 1) The equations describing the shape of the Gaussian. An example plot is shown. Note that in the program the Gaussians used are 11 bins wide—the ends that look like zero are just very small.

NOTE: One important change was that a bug was found in the Gaussian equation. Before, there was no sigma squared in the equation but that was a bug, and fixed. This is one major and important change to the code. Make sure to use the latest version of the code.

Due to the fact that each half bin had its own unique Gaussian functions, some of the test cases written discovered that the number of total stars was not conserved before and after convolution. As a result, the convolution kernel was rewritten and now runs like a pseudo convolution kernel. It works by taking the bins height and redistributing said bin into neighboring bins by multiplying the bin height by the Gaussian value relative to the original bin.

Errors:

One of the main things added was the code to calculate the once we had a guess of the number of stars in each bin. This was achieved by calculating the hessian matrix for the output variables. The partial derivatives were found numerically. Once the hessian matrix was calculated, the inverse of said matrix was calculated using NumPy. The error for each variable would be equal to the square root of the diagonal terms of the inverse matrix. This would be done for each pencil bean individually, then it would be plotted together. All in all, the errors were from a tenth to a hundredth the size of the bins for the variables it represented.

$$H\ f\ (x_1, x_2, \ldots, x_n) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\, \partial x_2} & \frac{\partial^2 f}{\partial x_1\, \partial x_3} & \cdots & \frac{\partial^2 f}{\partial x_1\, \partial x_n} \\ \frac{\partial^2 f}{\partial x_2\, \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2\, \partial x_3} & \cdots & \frac{\partial^2 f}{\partial x_2\, \partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\, \partial x_1} & \frac{\partial^2 f}{\partial x_n\, \partial x_2} & \frac{\partial^2 f}{\partial x_n\, \partial x_3} & \cdots & \frac{\partial^2 f}{\partial x_n^2}. \end{bmatrix}$$

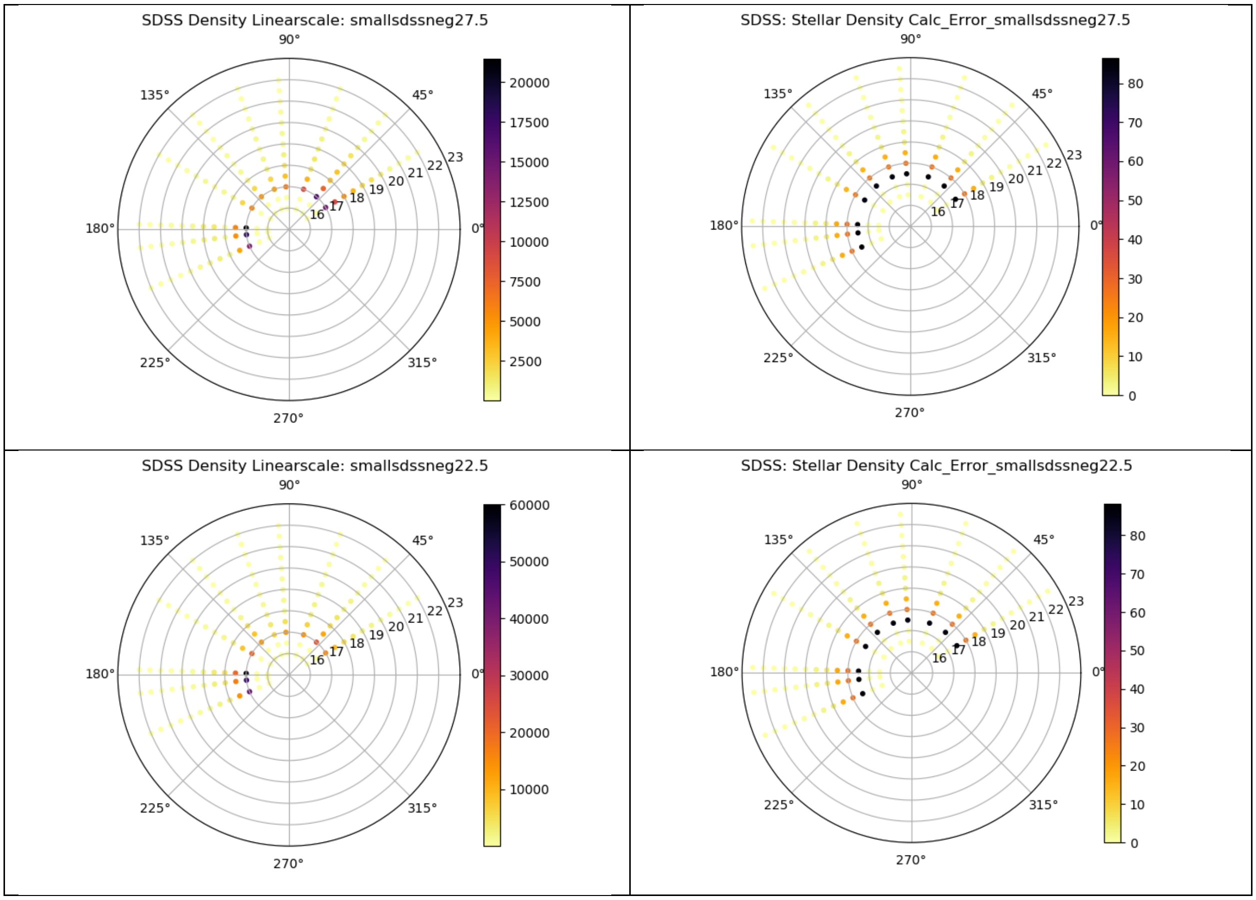
$$\frac{\partial^2 f}{\partial x \partial y}(a,b) \approx \frac{f(a+h_1,b+h_2) - f(a+h_1,b-h_2) - f(a-h_1,b+h_2) + f(a-h_1,b-h_2)}{4h_1h_2}.$$

Fig 2,3) A hessian matrix and the equation used to calculate the partial derivatives numerically. Then this matrix is inverted and the diagonal terms of this inverted matric are taken to determine errors in each parameter.

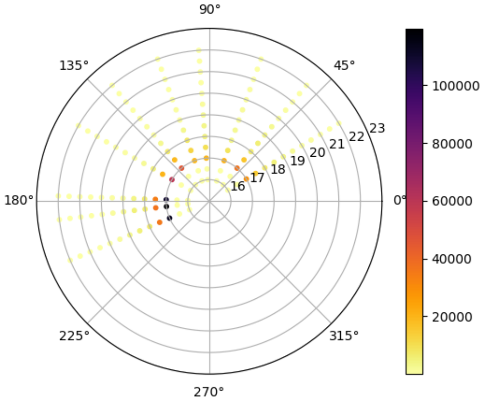
Data and Runs:

Compared to last semester, there was a small bug when plotting many of the bins. This bug had been fixed, so that the plots show the correct density of stars in each magnitude bin. This change had affected both the SDSS and the PanStar plots.

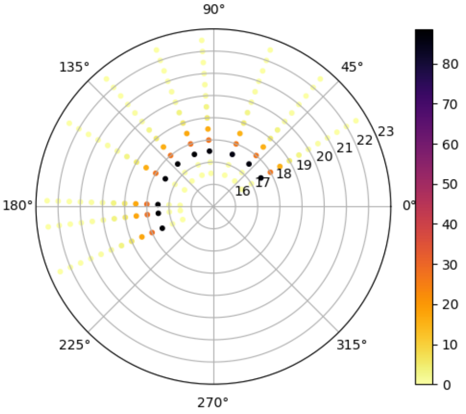
For the SDSS data, we need to account for the 16 magnitude cutoff, so we exclude that bin. This means that the inner magnitudes 16 and 16.5 are not included. Note that the 16.5 bin is excluded because it derives from a linear interpolation from the 16 and 17 magnitude bins. There are issues with the error bins for the SDSS data, which stems from some bug in its code this has yet to be resolved. The SDSS plots start from most negative l and end at most positive b. There are many plots with many variations from density to linear counts, to linear scale vs log scale. A full selection of each combination of plots can be found in the PowerPoints in the folder for Density Comparison.



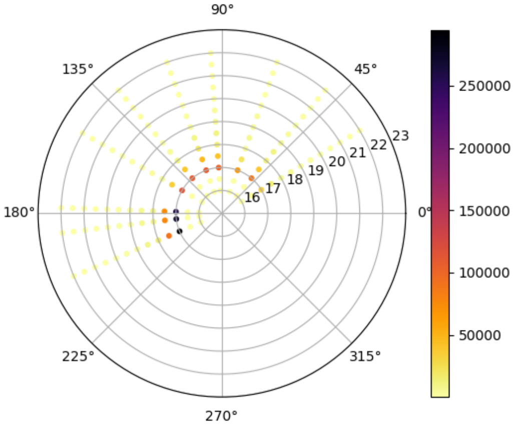
SDSS Density Linearscale: smallsdssneg17.5



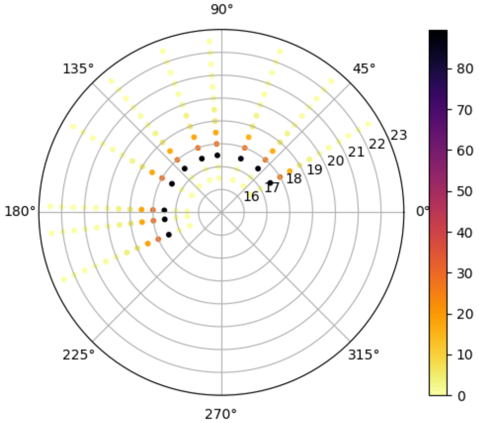
SDSS: Stellar Density Calc_Error_smallsdssneg17.5



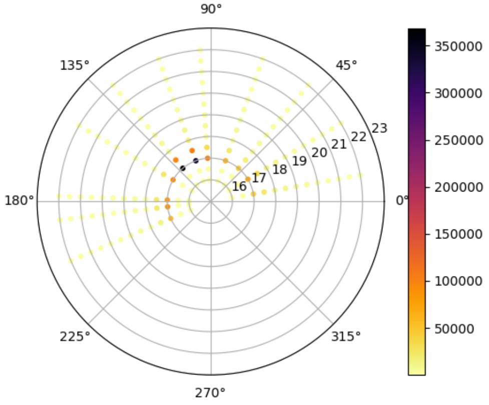
SDSS Density Linearscale: smallsdssneg12.5



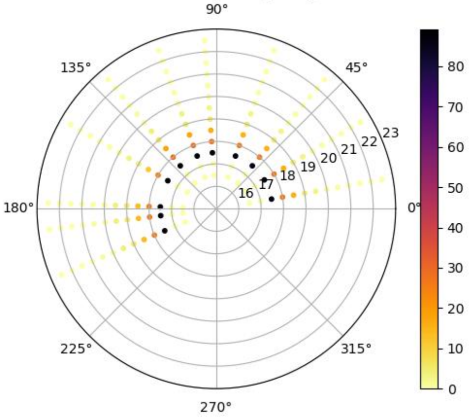
SDSS: Stellar Density Calc_Error_smallsdssneg12.5

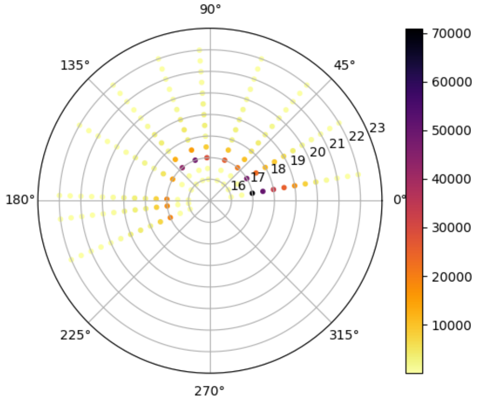


SDSS Density Linearscale: smallsdss12.5

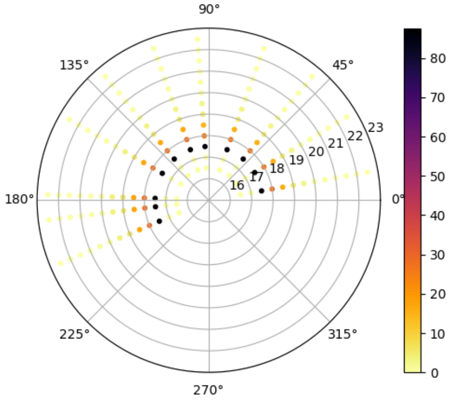


SDSS: Stellar Density Calc_Error_smallsdss12.5

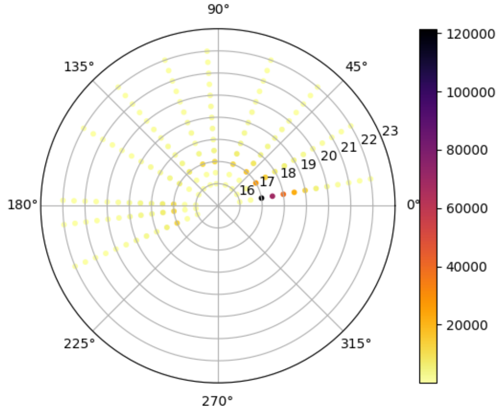




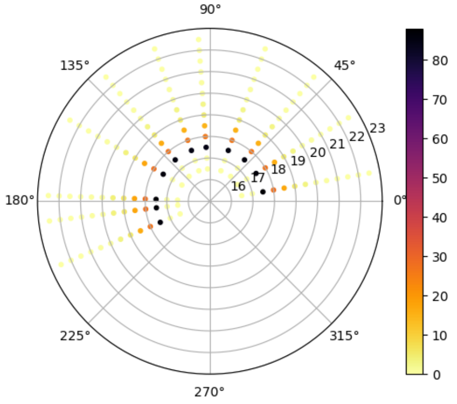
SDSS Density Linearscale: smallsdss17.5



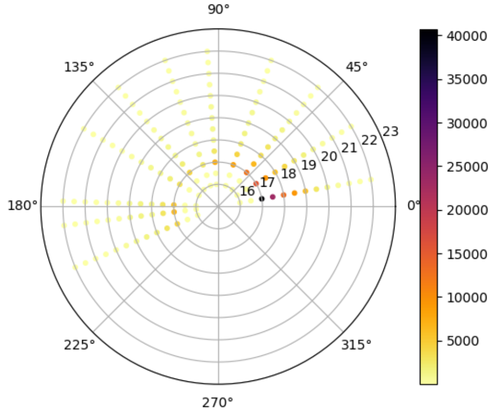
SDSS: Stellar Density Calc_Error_smallsdss17.5



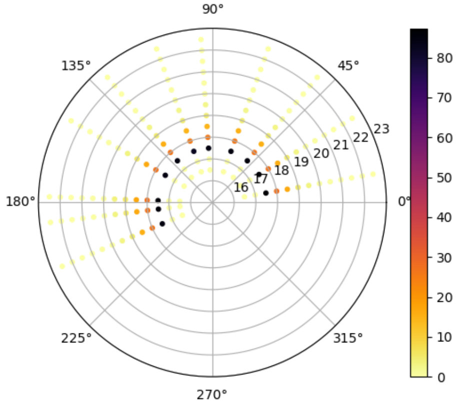
SDSS Density Linearscale: smallsdss22.5



SDSS: Stellar Density Calc_Error_smallsdss22.5

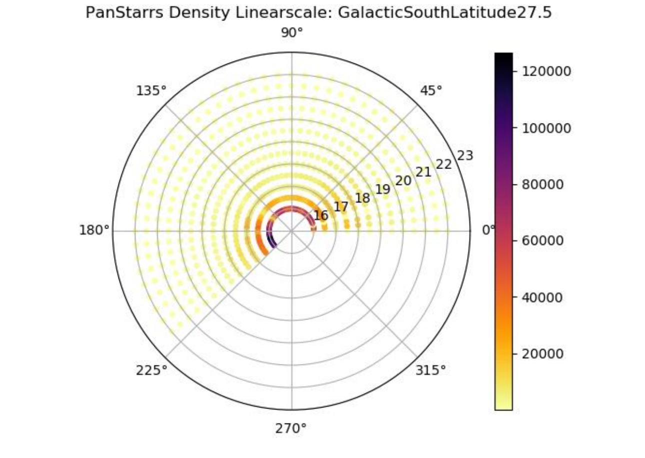


SDSS Density Linearscale: smallsdss27.5

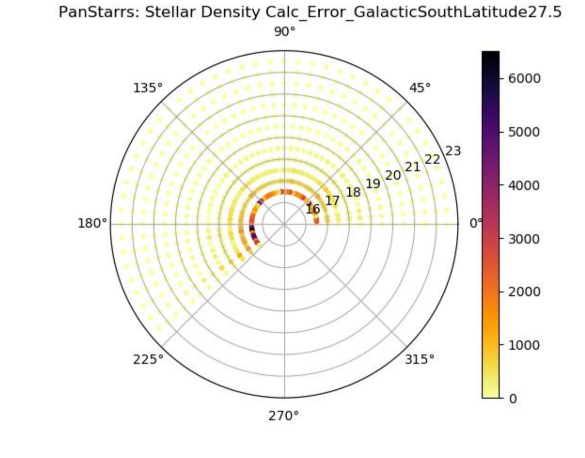


SDSS: Stellar Density Calc_Error_smallsdss27.5

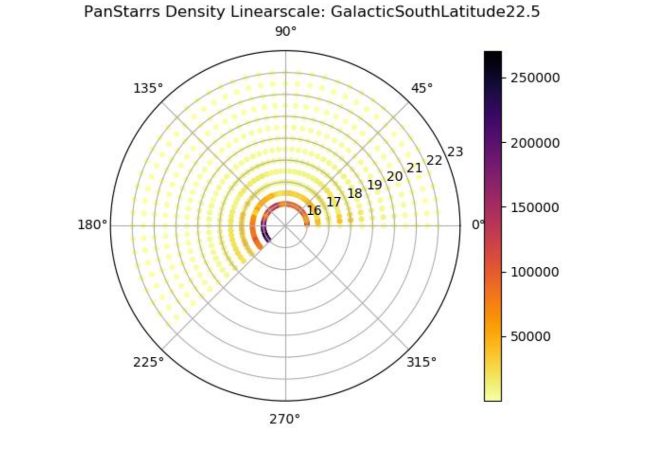
The PanStarrs runs include the innermost bin, so it has results for magnitudes 16 and 16.5. These results look more reasonable in that the errors are what looks normal and that the densities near the disk are about what is expected. Just like the SDSS set, the units are stars/kpc^3 and the plots start from the most negative l to the most positive l.



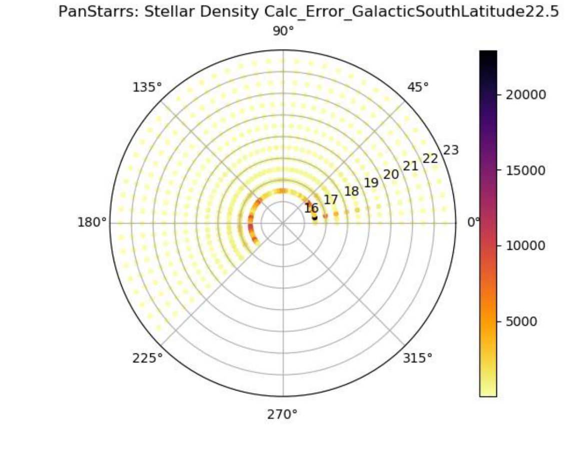
PanStarrs Density Linearscale: GalacticSouthLatitude27.5



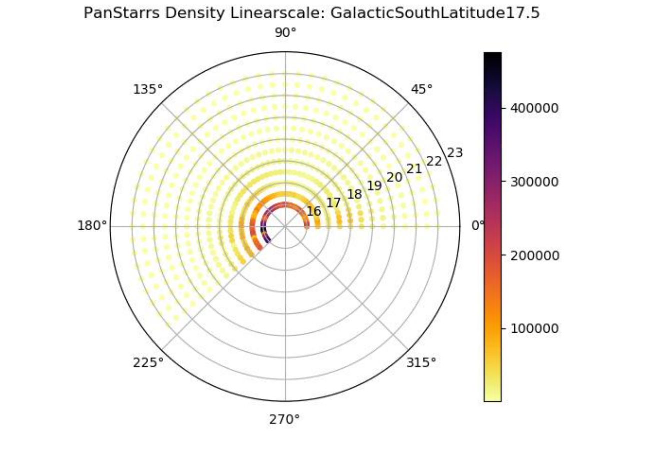
PanStarrs: Stellar Density Calc_Error_GalacticSouthLatitude27.5



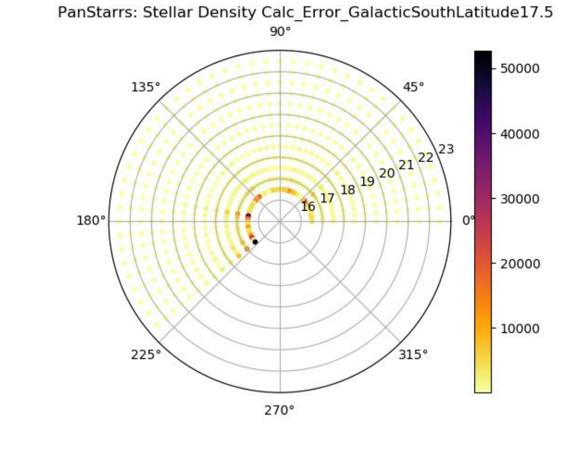
PanStarrs Density Linearscale: GalacticSouthLatitude22.5



PanStarrs: Stellar Density Calc_Error_GalacticSouthLatitude22.5

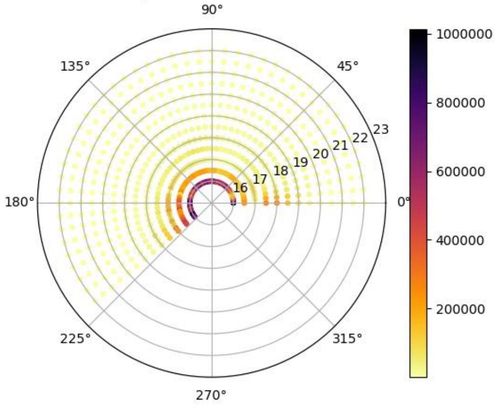


PanStarrs Density Linearscale: GalacticSouthLatitude17.5

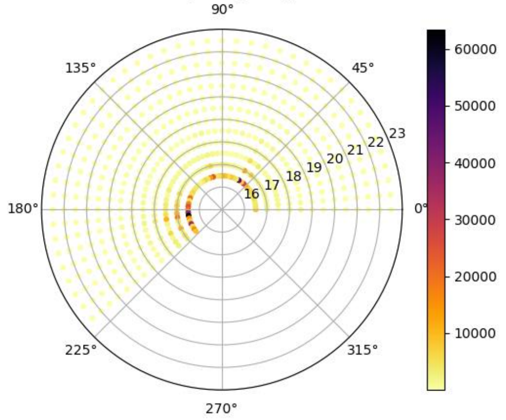


PanStarrs: Stellar Density Calc_Error_GalacticSouthLatitude17.5

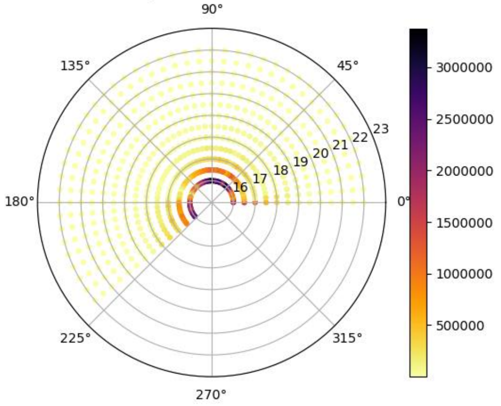
PanStarrs Density Linearscale: GalacticSouthLatitude12.5



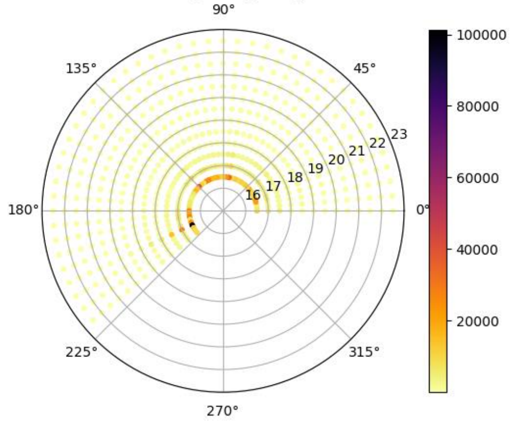
PanStarrs: Stellar Density Calc_Error_GalacticSouthLatitude12.5



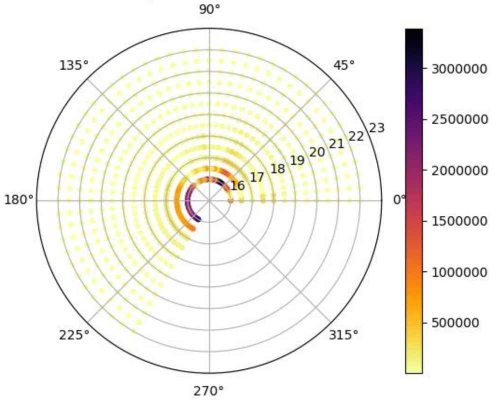
PanStarrs Density Linearscale: GalacticSouthLatitude7.5



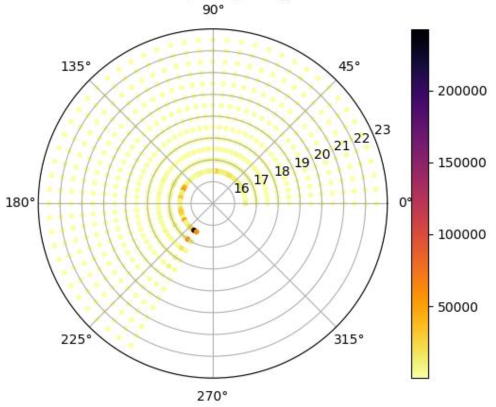
PanStarrs: Stellar Density Calc_Error_GalacticSouthLatitude7.5

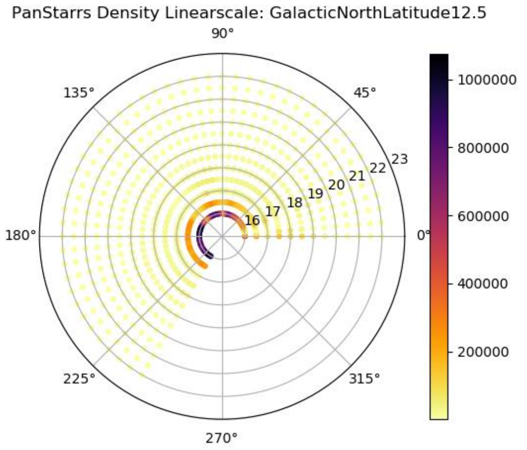


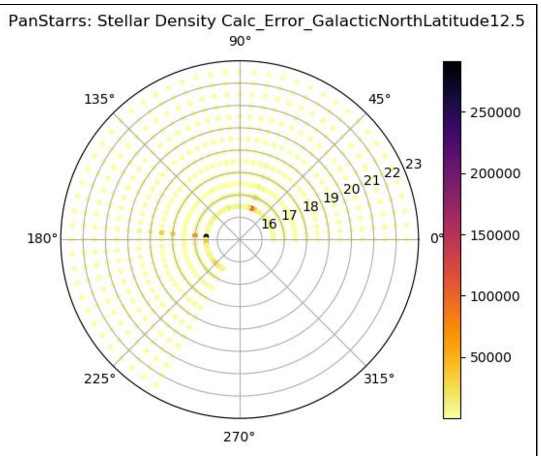
PanStarrs Density Linearscale: GalacticNorthLatitude7.5

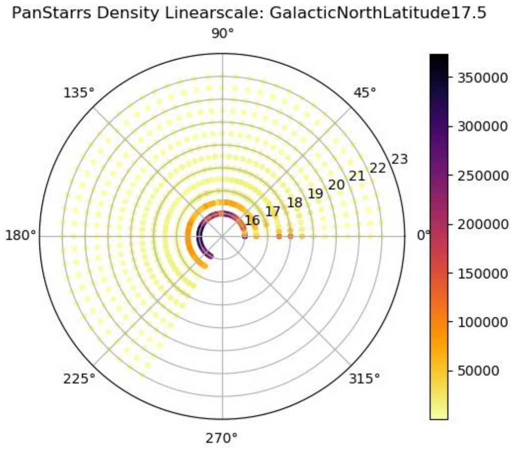


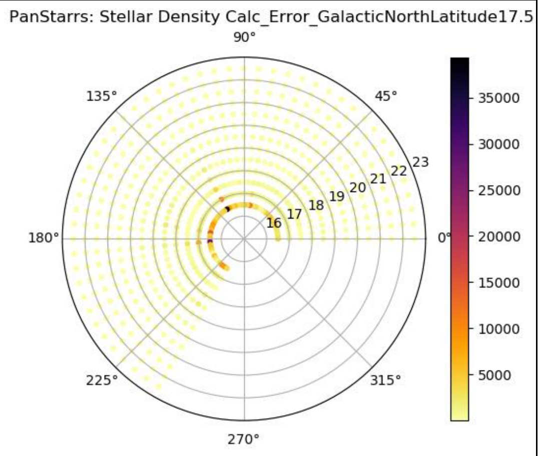
PanStarrs: Stellar Density Calc_Error_GalacticNorthLatitude7.5

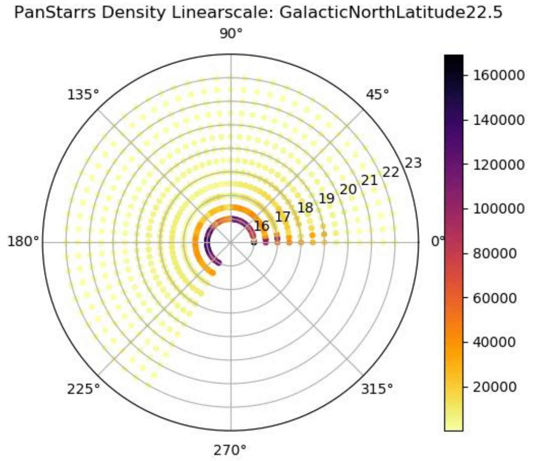


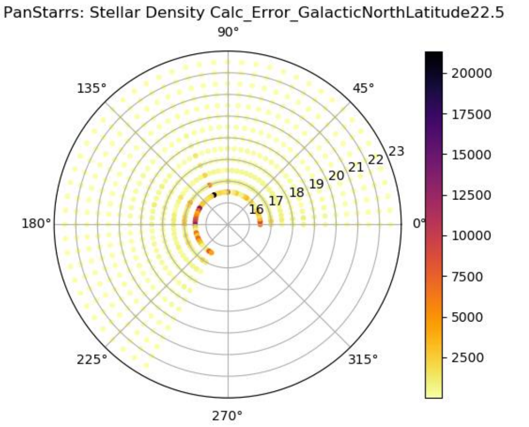

PanStarrs Density Linearscale: GalacticNorthLatitude12.5

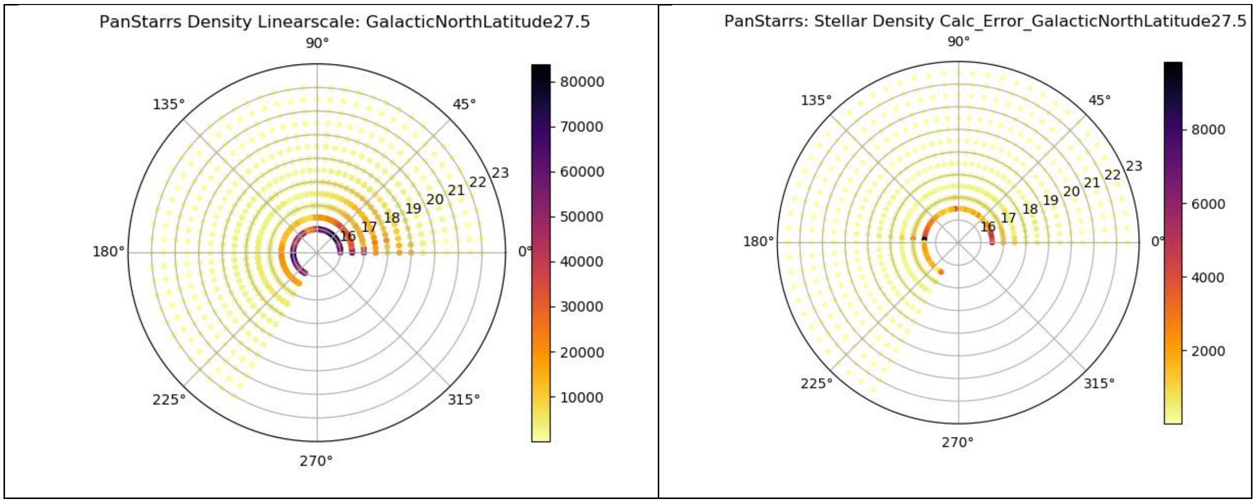

PanStarrs: Stellar Density Calc_Error_GalacticNorthLatitude12.5


PanStarrs Density Linearscale: GalacticNorthLatitude17.5


PanStarrs: Stellar Density Calc_Error_GalacticNorthLatitude17.5


PanStarrs Density Linearscale: GalacticNorthLatitude22.5


PanStarrs: Stellar Density Calc_Error_GalacticNorthLatitude22.5



Future Work:

Currently, to the best of my knowledge, the researching part of this project is done. What is needed to be worked is a methodical way to make sense out of and conclusions out of the data. What has been worked on is the creation of a disk stellar model from the paper Rings and Radial Waves in the Disk of the Milky Way. A link to the paper is found here: "https://iopscience.iop.org/article/10.1088/0004-637X/801/2/105". In the Density Comparison folder lies disk_model.py. This attempts to plot the structure described in the paper. It is still a work in progress. Note that there have not only been both bugs in the code, but also in the paper. One such typo in the paper is that the disk density terms for the disk and thin disk have e^-r/l. The geometry is not as trivial as expected.

Some Notes:

build- You need to probably create this folder. Create a "build" folder, and in it run "cmake <path to source>". The path should lead to the outer most CMakeLists.txt which resides in the DiskDensityFit file. Make sure you have a good c++ compiler and you might need the boost packages. Something like "sudo apt-get install libboost-all-dev" should work, but you it might vary based on your terminal. After you get cmake working, in the future all you need to do is type "make" in the build file. To run the program you just compiled, go into /build/Density_Program. type /.Diskfit and it should run. It needs input file names and output file names as command line arguments to crunch data, however, without any command line arguments it will just run the test cases which are also worth looking over.

DiskDensityFit- contains the meat of the program. It has all the c++ files you would ever need to edit. All you should need is DiskDensityFit\Density_Program\includes "Density_to_Star_Counts.h", the DiskDensityFit\Density_Program\src "Density_to_StarCounts_TrimmedDown.cpp", and you are free to look over some of the test cases as well in DiskDensityFit\Density_Program\Tests\src "Tests_main.cpp".

Density Comparison- Contains python scripts to model the disk for a comparison. This is where the files that "Future Work" references resides.

PANNSTARRS- Where pannstarrs raw data and results live.

SDSS- Where SDSS raw data and results live.

I know I had trouble setting up the program to compile and run. I hope the information is at least a little helpful; feel free to ask me—you can always reach me at: huang.chris.me@gmail.com

Make sure to take a look at the readmes~

All files can be found at: https://github.com/ChrisHuang-git/Spring20DiskFitting

All files can also be found: (just a layer of redundancy if github is odd)
https://drive.google.com/file/d/11yjVNN3ZXo7Ik2psu7avHJ8fgZ1iWv_B/view?usp=sharing