# Analysis and Visualization of Astrometric Mission's Data

Instructor: Dr. Parthiban Srinivasan
Department of Data Science and Engineering
Indian Institute of Science Education and Research, Bhopal

Divyesh Krishna Lagisetty
Department of Physics
Indian Institute of Science Education and Research, Bhopal

19178

Data Science in Practice DSE 315

## Abstract:

This project tries to address the Hipparcos missions data which is very historical. This project deals with the very cluttered data and try to de-clutter the data by useful methods of data cleaning. This project tries to give the physical emphasis of the data that is present in the enormous amount and try to visualize the stars in the night sky that is captured by this astro-metric mission. This project tries to calculate the distances of the stars from the earth and also gives the information of the nearest star that is present in the particular inclination and declination.

## Research Question:

Main objective of this project is to visualize the Hipparcos data on stars and their distances from the Earth and calculating the number of stars that are visible from the data that is present in theThe Hipparcos and Tycho Catalogues.

## Background & Prior work

Prior to this Project, one must go through the basics of Data-Analysis that were discussed by Dr. Parthiban Srinivasan such as Descriptive Analysis, Exploratory analysis, Inferential data analysis, Pandas, Numpy, Scipy. This kind of data analysis's were done on different kinds of data by Kaggle. Little knowledge of Astro-Physics topics like parllax, inclination angle.. are required .

## Introduction:

By the second half of the 20th century, the accurate measurement of star positions from the ground was running into essentially insurmountable barriers to improvements in accuracy, especially for large-angle measurements and systematic terms. Problems were dominated by the effects of the Earth's atmosphere, but were compounded by complex optical terms, thermal and gravitational instrument flexures, and the absence of all-sky visibility. A formal proposal to make these exacting observations from space was first put forward in 1967.

The mission was originally proposed to the French space agency CNES, which considered it too complex and expensive for a single national programme and recommended that it be proposed in a multinational context. Its acceptance within the European Space Agency's scientific programme, in 1980, was the result of a lengthy process of study and lobbying. The underlying scientific motivation was to determine the physical properties of the stars through the measurement of their distances and space motions, and thus to place theoretical studies of stellar structure and evolution, and studies of galactic structure and kinematics, on a more secure empirical basis. Observationally, the objective was to provide the positions, parallaxes, and annual proper motions for some 100,000 stars with an unprecedented accuracy of 0.002 arcseconds, a target in practice eventually surpassed by a factor of two. The name of the space telescope, ”Hipparcos”, was an acronym for High Precision Parallax Collecting Satellite, and it also reflected the name of the ancient Greek astronomer Hipparchus, who is considered the founder of trigonometry and the discoverer of the precession of the equinoxes (due to the Earth wobbling on its axis).

Hipparcos or the High Precision Parallax Collecting Satellite was a scientific satellite of the European Space Agency (ESA), launched in 1989 and operated until 1993. It was the first space experiment devoted to precision astrometry, the accurate measurement of the positions of celestial objects on the sky. This permitted the first high-precision measurements of the intrinsic brightnesses (compared to the less precise apparent brightness), proper motions, and parallaxes of stars, enabling better calculations of their distance and tangential velocity.When combined with radial velocity measurements from spectroscopy, astrophysicists were able to finally measure all six quantities needed to determine the motion of stars. The resulting Hipparcos Catalogue, a highprecision catalogue of more than 118,200 stars, was published in 1997.

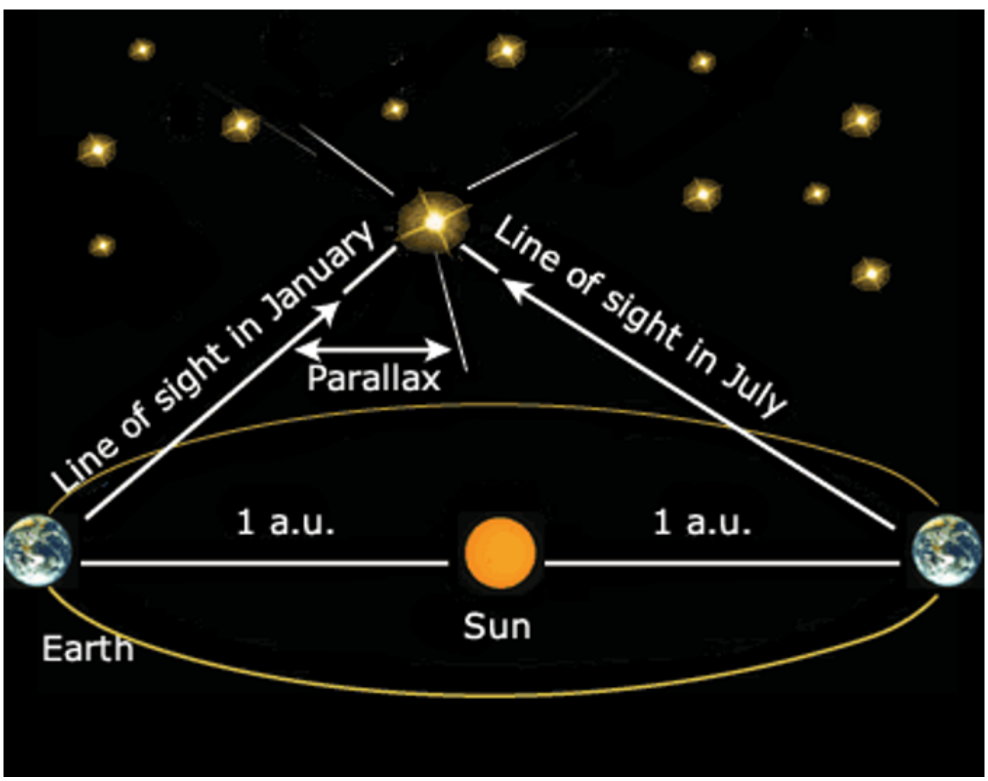
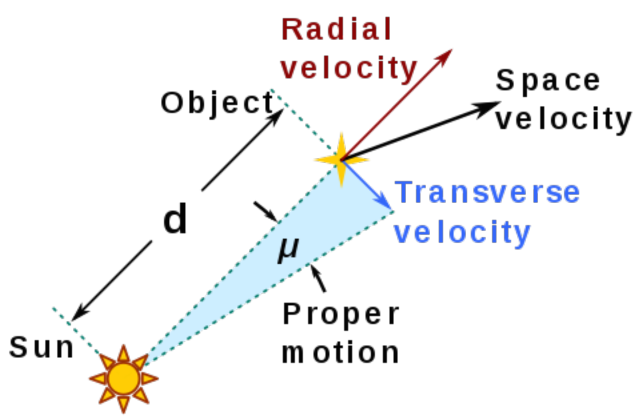



FIGURE : *Parallax*

We will be using that data to generate plots of night sky and other inferences in to stars. We will first clean the data and convert it into usable form and then perform visualisation on it using different python libraries like NumPy, SciPy, Mathplotlib, etc.. Though the catalogue has vast data, we are choosing only a particular parts of it which will be needed to do the necessary plotting. The physical phenomenon which, in normal words, is related to the fact that objects appear to be in different places when observed from different angles. Parallax, in astronomy, the difference in direction of a celestial object as seen by an observer from two widely separated points. The measurement of parallax is used directly to find the distance of the body from Earth (geocentric parallax) and from the Sun (heliocentric parallax). Hipparocs measured parallax by direct triangulation using observations of parallax from either side of Earth's orbit around the Sun.
So, distance can be obtained simply from the parllax using $d = tan^{-1}p$.




Astronomers define star brightness in terms of apparent magnitude how bright the star appears from Earth and absolute magnitude how bright the star appears at a standard distance of 32.6 light-years, or 10 parsecs. (A light-year is the distance light travels in one year,about 6 trillion miles, or 10 trillion kilometers.) Astronomers also measure luminosity is the amount of energy (light) that a star emits from its surface. Apparent magnitude (m) is a measure of the brightness of a star or other astronomical object observed from Earth.The faintest stars visible with the naked eye on the darkest night have apparent magnitudes of about +6.5, though this varies depending on a person's eyesight and with altitude and atmospheric conditions.

Proper Motion is the apparent angular motion of a star across the sky with respect to more distant stars. Distant stars on the celestial sphere appear to remain fixed, however, most of the stars in the solar neighborhood appear to move across the celestial sphere since these stars are closer. This apparent motion of stars with respect to the fixed stars in the background is called the proper motion.Typical proper motion is 0.1 arcsec/year. The effect of proper motions build up over time. The longer you wait, the greater the apparent angular motion is.

## Data Cleaning:

The data is filled with '|'. So, we need to use a delimiter to fix it. As the first 11 rows are useless, we skip those. We also give header a number and skip the footer.

```
    e|    H|_Hp|a|      |      |        |p|a|t|          |D|Ns|co|a|c|a|_H| th|       |      |      |  e|e|r|e|       |           |           |
        |(V-I|             |r
g|   HIP|y|RAhms       DEdms       | Vmag|g|g|       RAdeg        DEdeg|f|     Plx|    pmRA|    pmDE|_RAdeg|_DEdeg|
e_Plx|e_pmRA|e_pmDE|DE:RA|1x:RA|1x:DE|RA:RA|RA:DE|A:Plx|DE:RA|DE:DE|E:Plx|:pmRA| F1|   F2       BTmag|BTmag| VTmag|VTmag|g|    B-V|e_B-
V|V| V-I|_V-I|I|g|  Hpmag|_Hpmag|pscat|mag|g|Hpmax|HPmin| Period|e|r|o|CCDM      |M|ys|mp|g|e|l|IP|eta|    rho|e_rho|  dHp|_dHp|y|t|s|
HD|BD         |CoD        |CPD        |)red|SpType       |_
-|------|-|-----------------------|-----|-|-|-------------------------|-|-------|--------|--------|------|------|------|------|------|---
--|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|----------------|-----|------|-----|-|------|-----|-|----|----|-|-|-------
|------|-----|---|-|-----|-----|-------|-|-|-|----------|-|--|--|-|-|-|--|---|-------|-----|-----|----|-|-|-|------|----------|----------
|----------|----|------------|-
H|     1| |00 00 00.22 +01 05 20.4| 9.10| |H|000.00091185 +01.08901332| |   3.54|   -5.20|   -1.88|  1.32|  0.74|  1.39|  1.36|  0.81|
0.32|-0.07|-0.11|-0.24| 0.09|-0.01| 0.10|-0.01| 0.01| 0.34|  0| 0.74     1 9.643|0.020| 9.130|0.019| | 0.482|0.025|T|0.55|0.03|L| |
9.2043|0.0020|0.017| 87| | 9.17| 9.24|       | | | |            | |  | 1| | | |  |    |       |     |      |   |S| | |224700|B+00 5077 |
     |         |0.66|F5          |S
H|     2| |00 00 00.91 -19 29 55.8| 9.27| |G|000.00379737 -19.49883745|+|  21.90|  181.21|   -0.93|  1.28|  0.70|  3.10|  1.74|  0.92|
0.12|-0.14|-0.24|-0.29| 0.01| 0.21|-0.02|-0.19|-0.28| 0.14|  2| 1.45     210.519|0.033| 9.378|0.021| | 0.999|0.002|G|1.04|0.00|I| |
9.4017|0.0017|0.015|120| | 9.37| 9.44|       |C| | |            | |  | 1|O| | |  |    |       |     |      |   | | | |224690|B-20 6688 |
     |         |1.04|K3V         |4
H|     3| |00 00 01.20 +38 51 33.4| 6.61| |G|000.00500795 +38.85928608| |   2.81|    5.24|   -2.91|  0.53|  0.40|  0.63|  0.57|  0.47|
0.06| 0.09| 0.04| 0.43|-0.01|-0.06| 0.03| 0.24| 0.07| 0.21|  0|-0.45     3 6.576|0.004| 6.621|0.005| |-0.019|0.004|G|0.00|0.00|H| |
6.6081|0.0007|0.008|127| | 6.60| 6.62|       |C| | |00000+3852|I| 1| 1| | | |  |    |       |     |      |   |S| | |224699|B+38 5108 |
     |         |0.00|B9          |S
H|     4| |00 00 02.01 -51 53 36.8| 8.06| |H|000.00838170 -51.89354612| |   7.75|   62.85|    0.16|  0.53|  0.59|  0.97|  0.65|
0.65|-0.22|-0.09|-0.03| 0.24| 0.20| 0.08| 0.18| 0.08|-0.31|-0.18|  0|-1.46     4 8.471|0.007| 8.092|0.007| | 0.370|0.009|T|0.43|0.01|L| |
8.1498|0.0011|0.015|201| | 8.12| 8.18|       | | | |            | |  | 1| | | |  |    |       |     |      |   |S| | |224707|          |
     |P-52 12237|0.43|F0V         |2
H|     5| |00 00 02.39 -40 35 28.4| 8.55| |H|000.00996534 -40.59122440| |   2.87|    2.53|    9.07|  0.64|  0.61|  1.11|  0.67|  0.74|
0.10| 0.24| 0.06| 0.26|-0.10| 0.20|-0.16|-0.30|-0.19| 0.06|  0|-1.24     5 9.693|0.014| 8.656|0.010| | 0.902|0.013|T|0.90|0.01|L| |
8.7077|0.0018|0.019|161| | 8.68| 8.74|       | | | |            | |  | 1| | | |  |    |       |     |      |   | | | |224705|          |C-41
15372|P-41  9991|0.95|G8III       |2
```

FIGURE : *Data Cleaning*

Here, we just need few columns from the data namely 'magnitude', 'raw position', 'raw position parallax', 'proper motion alpha', 'proper motion delta', 'error alpha', 'error delta', 'error parallax', 'error motion alpha', 'error motion delta'. So, we will select particular columns and give own index starting from 0

## Setup:

Firstly, we import all the necessary libraries as we need such as NumPy, SciPy, Mathplotlib, etc...

Our next step is to load the data set from the text file to .csv file and only required columns must be loaded as mentioned earlier and name those columns with the appropriate names which were discussed in the introduction section.

```
data= pd.read_csv('I_239_hip_main',delimiter='|',engine='python',skiprows=11,
                    skipfooter=1,
                    index_col=0,
                    header=0,usecols=[1, 4, 7, 9, 10, 11, 12, 13, 14, 15, 16],
                    names=['column', 'magnitude', 'raw position', 'raw position parallax',
                           'proper motion alpha', 'proper motion delta',
                           'error alpha', 'error delta', 'error parallax',
                           'error motion alpha', 'error motion delta'])
```

FIGURE : *Importing required libraries*

We mention python engine to avoid errors. Now our data frame looks like this.

```
data.head()
```

| column | magnitude | raw position | raw position parallax | proper motion alpha | proper motion delta | error alpha | error delta | error parallax | error motion alpha | error motion delta | sp_type |
|---|---|---|---|---|---|---|---|---|---|---|---|
| H | 1 | 9.10 | 000.00091185 +01.08901332 | 3.54 | -5.20 | -1.88 | 1.32 | 0.74 | 1.39 | 1.36 | 0.81 |
| H | 2 | 9.27 | 000.00379737 -19.49883745 | 21.90 | 181.21 | -0.93 | 1.28 | 0.70 | 3.10 | 1.74 | 0.92 |
| H | 3 | 6.61 | 000.00500795 +38.85928608 | 2.81 | 5.24 | -2.91 | 0.53 | 0.40 | 0.63 | 0.57 | 0.47 |
| H | 4 | 8.06 | 000.00838170 -51.89354612 | 7.75 | 62.85 | 0.16 | 0.53 | 0.59 | 0.97 | 0.65 | 0.65 |
| H | 5 | 8.55 | 000.00996534 -40.59122440 | 2.87 | 2.53 | 9.07 | 0.64 | 0.61 | 1.11 | 0.67 | 0.74 |

FIGURE : *Data Frame*

Here, the raw position has two parts in it which are alpha and delta. We split that and add two new columns resulting in following. We drop empty columns using dropna.

```
data[['alpha', 'delta']] = (
        data['raw position']
        .str.split(expand=True)
        .astype(float))
#splits raw position strings into alpha and delta and make them floats
```

```
data.head()
```

| column | magnitude | raw position | raw position parallax | proper motion alpha | proper motion delta | error alpha | error delta | error parallax | error motion alpha | error motion delta | alpha | delta |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 9.10 | 000.00091185 +01.08901332 | 3.54 | -5.20 | -1.88 | 1.32 | 0.74 | 1.39 | 1.36 | 0.81 | 0.000912 | 1.089013 |
| 2 | 9.27 | 000.00379737 -19.49883745 | 21.90 | 181.21 | -0.93 | 1.28 | 0.70 | 3.10 | 1.74 | 0.92 | 0.003797 | -19.498837 |
| 3 | 6.61 | 000.00500795 +38.85928608 | 2.81 | 5.24 | -2.91 | 0.53 | 0.40 | 0.63 | 0.57 | 0.47 | 0.005008 | 38.859286 |
| 4 | 8.06 | 000.00838170 -51.89354612 | 7.75 | 62.85 | 0.16 | 0.53 | 0.59 | 0.97 | 0.65 | 0.65 | 0.008382 | -51.893546 |
| 5 | 8.55 | 000.00996534 -40.59122440 | 2.87 | 2.53 | 9.07 | 0.64 | 0.61 | 1.11 | 0.67 | 0.74 | 0.009965 | -40.591224 |

FIGURE : *splitted raw position strings into alpha and delta and make them floats*

```
data.dropna(inplace=True)
#drops empty columns
data.drop('raw position', axis=1, inplace=True)
#drops raw position column
```

```
data.head()
```

| column | magnitude | raw position parallax | proper motion alpha | proper motion delta | error alpha | error delta | error parallax | error motion alpha | error motion delta | alpha | delta |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 9.10 | 3.54 | -5.20 | -1.88 | 1.32 | 0.74 | 1.39 | 1.36 | 0.81 | 0.000912 | 1.089013 |
| 2 | 9.27 | 21.90 | 181.21 | -0.93 | 1.28 | 0.70 | 3.10 | 1.74 | 0.92 | 0.003797 | -19.498837 |
| 3 | 6.61 | 2.81 | 5.24 | -2.91 | 0.53 | 0.40 | 0.63 | 0.57 | 0.47 | 0.005008 | 38.859286 |
| 4 | 8.06 | 7.75 | 62.85 | 0.16 | 0.53 | 0.59 | 0.97 | 0.65 | 0.65 | 0.008382 | -51.893546 |
| 5 | 8.55 | 2.87 | 2.53 | 9.07 | 0.64 | 0.61 | 1.11 | 0.67 | 0.74 | 0.009965 | -40.591224 |

FIGURE : *Dropping empty columns*

Now, we need to convert the values in the columns. But, they are in strings, first we use for loop through a list of columns and make all the series elements into float. Then, we the columns are divided by 1000 to go from miliarcseconds to arcseconds, and then by another factor of 3600 to obtain the values in degrees. This will lead to the following dataframe

| column | raw position parallax | proper motion alpha | proper motion delta | error alpha | error delta | error parallax | error motion alpha | error motion delta |
|---|---|---|---|---|---|---|---|---|
| 1 | 9.833333e-07 | -1.444444e-06 | -5.222222e-07 | 3.666667e-07 | 2.055556e-07 | 3.861111e-07 | 3.777778e-07 | 2.250000e-07 |
| 2 | 6.083333e-06 | 5.033611e-05 | -2.583333e-07 | 3.555556e-07 | 1.944444e-07 | 8.611111e-07 | 4.833333e-07 | 2.555556e-07 |
| 3 | 7.805556e-07 | 1.455556e-06 | -8.083333e-07 | 1.472222e-07 | 1.111111e-07 | 1.750000e-07 | 1.583333e-07 | 1.305556e-07 |
| 4 | 2.152778e-06 | 1.745833e-05 | 4.444444e-08 | 1.472222e-07 | 1.638889e-07 | 2.694444e-07 | 1.805556e-07 | 1.805556e-07 |
| 5 | 7.972222e-07 | 7.027778e-07 | 2.519444e-06 | 1.777778e-07 | 1.694444e-07 | 3.083333e-07 | 1.861111e-07 | 2.055556e-07 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 118318 | 5.333333e-07 | -6.000000e-07 | 5.805556e-07 | 2.166667e-07 | 1.388889e-07 | 2.527778e-07 | 2.444444e-07 | 1.666667e-07 |
| 118319 | 2.952778e-06 | 4.131667e-05 | 7.647222e-06 | 2.694444e-07 | 1.611111e-07 | 3.250000e-07 | 3.055556e-07 | 1.444444e-07 |
| 118320 | 1.388889e-06 | 5.811111e-06 | -9.794444e-06 | 2.638889e-07 | 1.472222e-07 | 2.805556e-07 | 2.472222e-07 | 1.500000e-07 |
| 118321 | 5.338889e-06 | 6.027500e-05 | 2.957222e-05 | 2.055556e-07 | 2.111111e-07 | 2.777778e-07 | 2.333333e-07 | 2.250000e-07 |
| 118322 | 2.419444e-06 | 1.350833e-05 | -6.202778e-06 | 1.222222e-07 | 1.166667e-07 | 1.583333e-07 | 1.333333e-07 | 1.305556e-07 |

117955 rows × 8 columns

FIGURE : *After converting the elemnts in columns to float and dividing with 3600000.0 to convert to degrees*

## Visualization:

Now, Plotting the positions we found of alpha and delta are basically elevation and declination. We, plot the stars based on that and get a star map on alpha vs delta graph for a section we want. I chose [0,360] and [0,90] for elevation and declination and got the plot of stars in that part of sky. The image generated is,

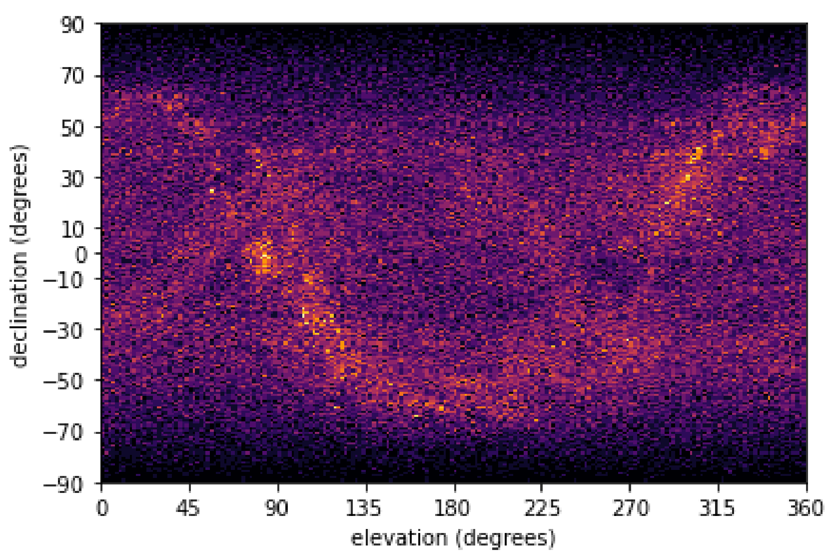



FIGURE : *Visualization of stars*

Now, to plot the distance of stars we need to take tan inverse of raw position parallax and do some adjustments to match the units and this will be a new column. We, plot number of stars vs their distance in light years. The following is the plot generated.

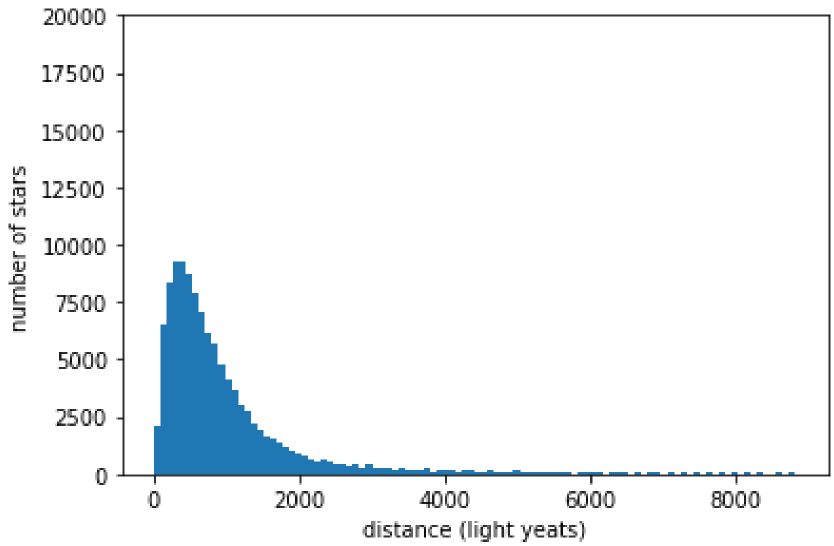



FIGURE : *Number of stars and their distances*

Now, we can use this data and find the closest stars to earth, like following: We can also arrange

```
data.iloc[data['distance'].argmin()]

magnitude                      11.01
raw position parallax       0.000215
proper motion alpha        -0.001049
proper motion delta         0.000213
error alpha                      0.0
error delta                      0.0
error parallax              0.000001
error motion alpha               0.0
error motion delta          0.000001
alpha                     217.448948
delta                     -62.681352
distance                    4.223016
Name: 70890, dtype: object
```

FIGURE

them in order, in anyway we like. Now, using magnitude of star to plot them in some particular range of elevation and declination, let's take [0,360] and [0,90] for elevation and declination respectively. We first convert magnitude to float, we didn't convert earlier because then we converted them into degrees right inside the for loop. Here, we don't need to do that. The magnitude used to measure brightness of stars is a logarithmic scale, and is set up such that a difference of 5 magnitudes corresponds to a factor of 100 in the brightness of the objects. Moreover, somewhat confusingly, brighter objects have a smaller magnitude. And so, in order to filter only the visible stars, all we need to do is to isolate those around some magnitude, consider it 6.5 From the above

```
data['magnitude']=data['magnitude'].astype(float)
data['magnitude']
visible = data[data['magnitude'] < 6.5]
#here cutoff magnitude is 6.5
```

```
brightness = np.power(10, -(visible['magnitude'] - visible['magnitude'].min()) / 10)
brightness = (1 - brightness).to_numpy()
plt.scatter(visible['alpha'], visible['delta'], c=brightness, s=brightness * 4,
            cmap='Greys', marker='.', vmin=0.5, vmax=1.0)
plt.xlim([0, 360])
plt.ylim([0, 100])
plt.xlabel('inclination (degrees)')
plt.ylabel('declination (degrees)')
ax = plt.gca()
ax.set_facecolor((0.0, 0.0, 0.0))
plt.show()
```

visible stars set we can easily get the number of stars that are visible in the night sky from the following command.

```
visible['magnitude'].count()
#this gives number of stars in night skyD
```

```
8785
```

Finally, We get the following plot depicting the nearest stars to the earth.

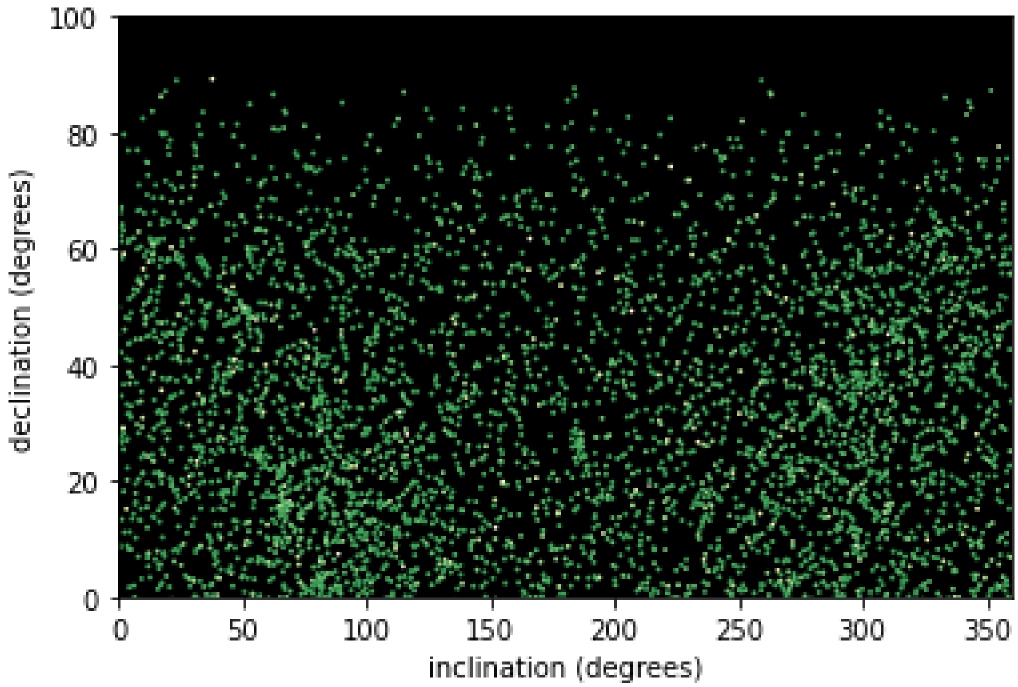



Figure : *Result*

## Summary:

This project tried to address the research problem that was posed in the start of this report that is to visualize the Hipparcos missions data and by following the steps like data cleaning as the data has the enormous amount of data, crasting the set-up and finally visualising data using important libraries. This can be also used to identify star constellations, predict comet and asteroid paths. With the relative velocities of stars given, we can even predict star locations and identify where they will be almost predicting the future.