Gaia Project

Presented by Zhiyan Wang

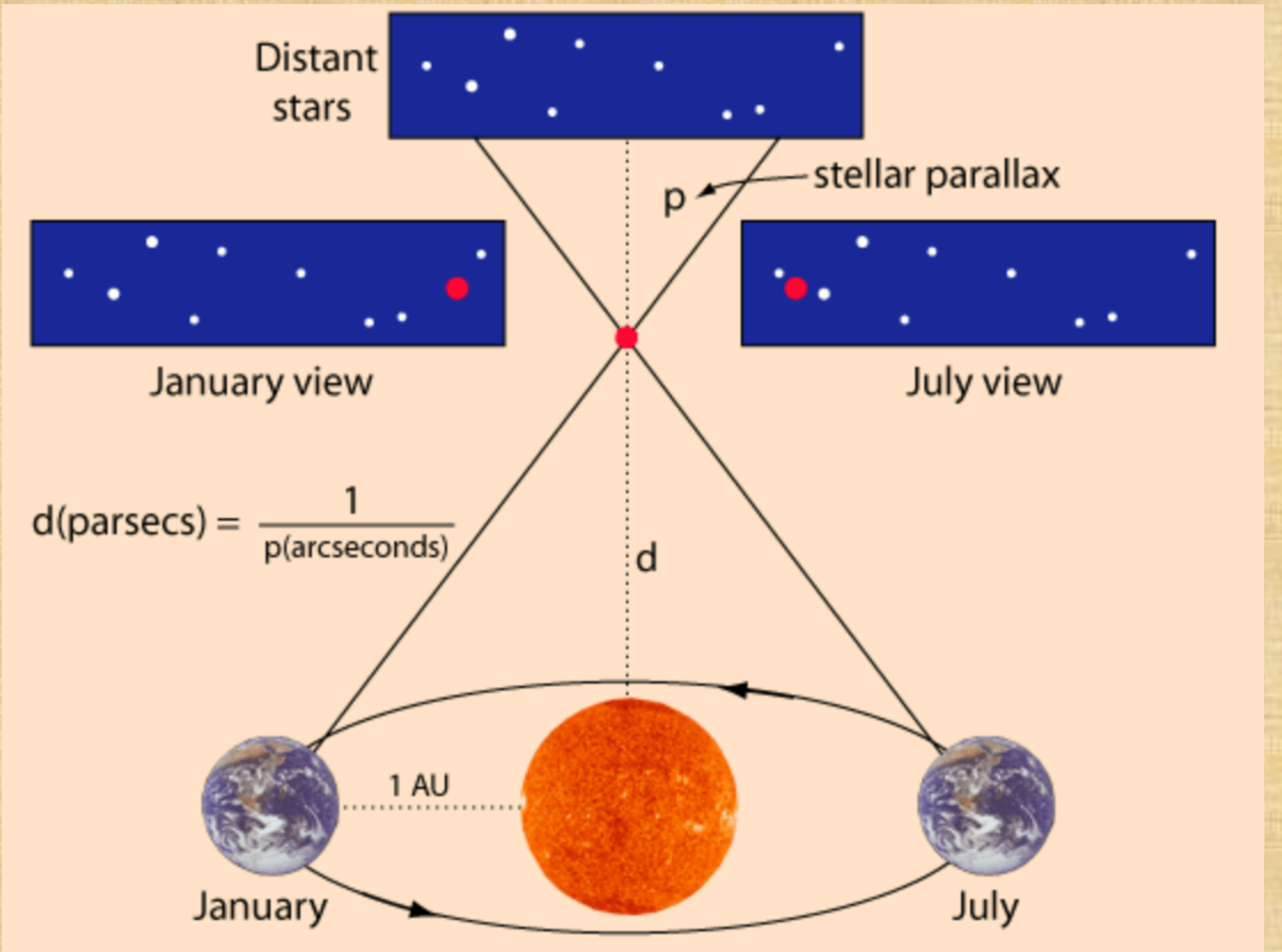
Analysis of Tycho-Gaia Astrometric Solution (TGAS) data and its matched data with 2MASS

- Gaia: Astrometry of stars (Position, distance, proper motion, etc.)
- 2Mass: Astronomical survey of sky in the infrared spectrum

Presentations:

- Transverse velocity statistics
- Density distribution and transverse velocity distribution
- Hertzsprung–Russell diagram

Background: Parallax

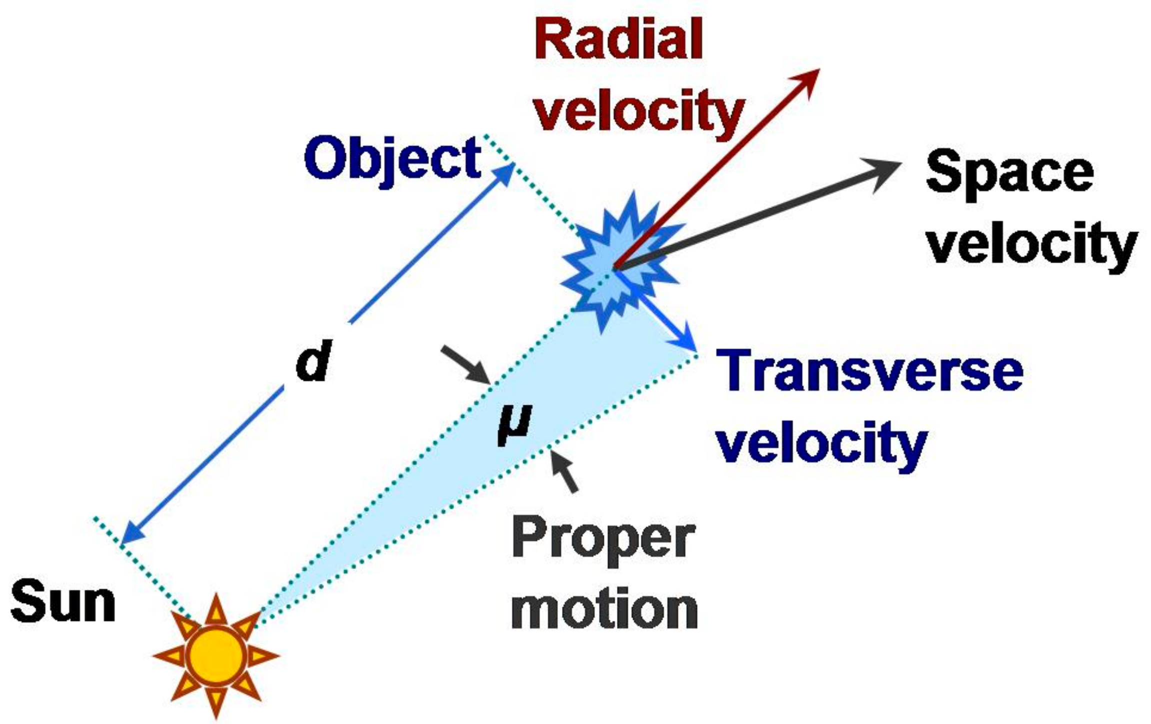


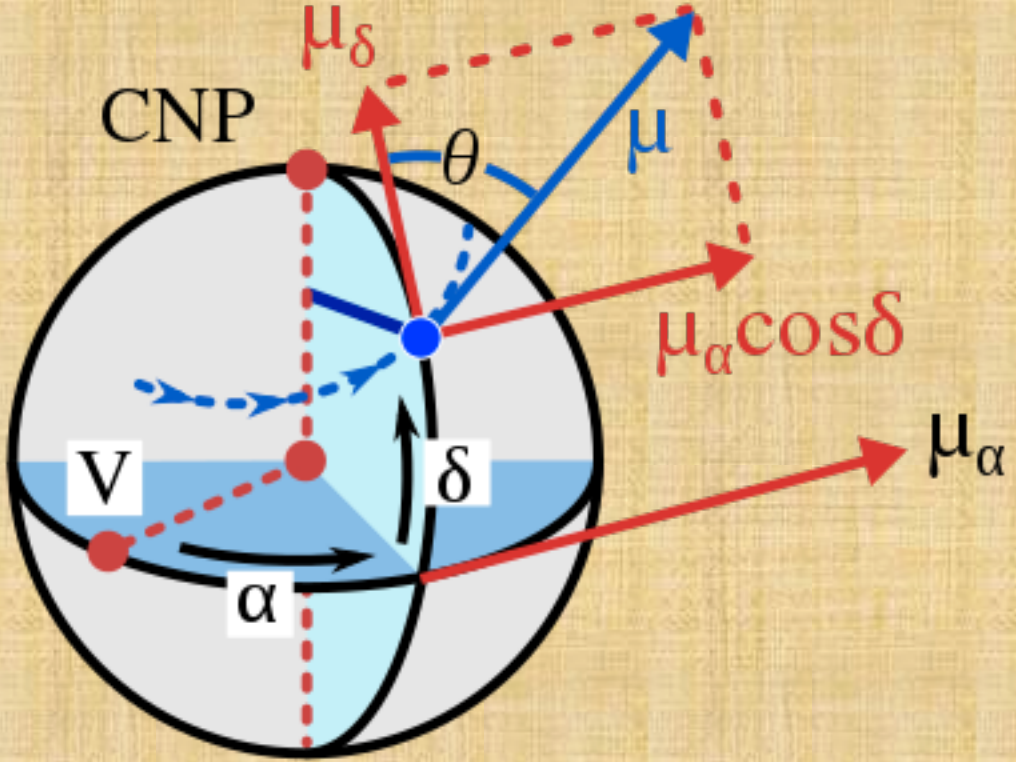
$$\text{distance} = \frac{1}{\text{parallax}}$$

Distance: In unit of parsec (1 parsec = 3.26 light years)

Parallax: In unit of arcsec (1 arcsec = 1/3600 degree)

Background: Proper Motion





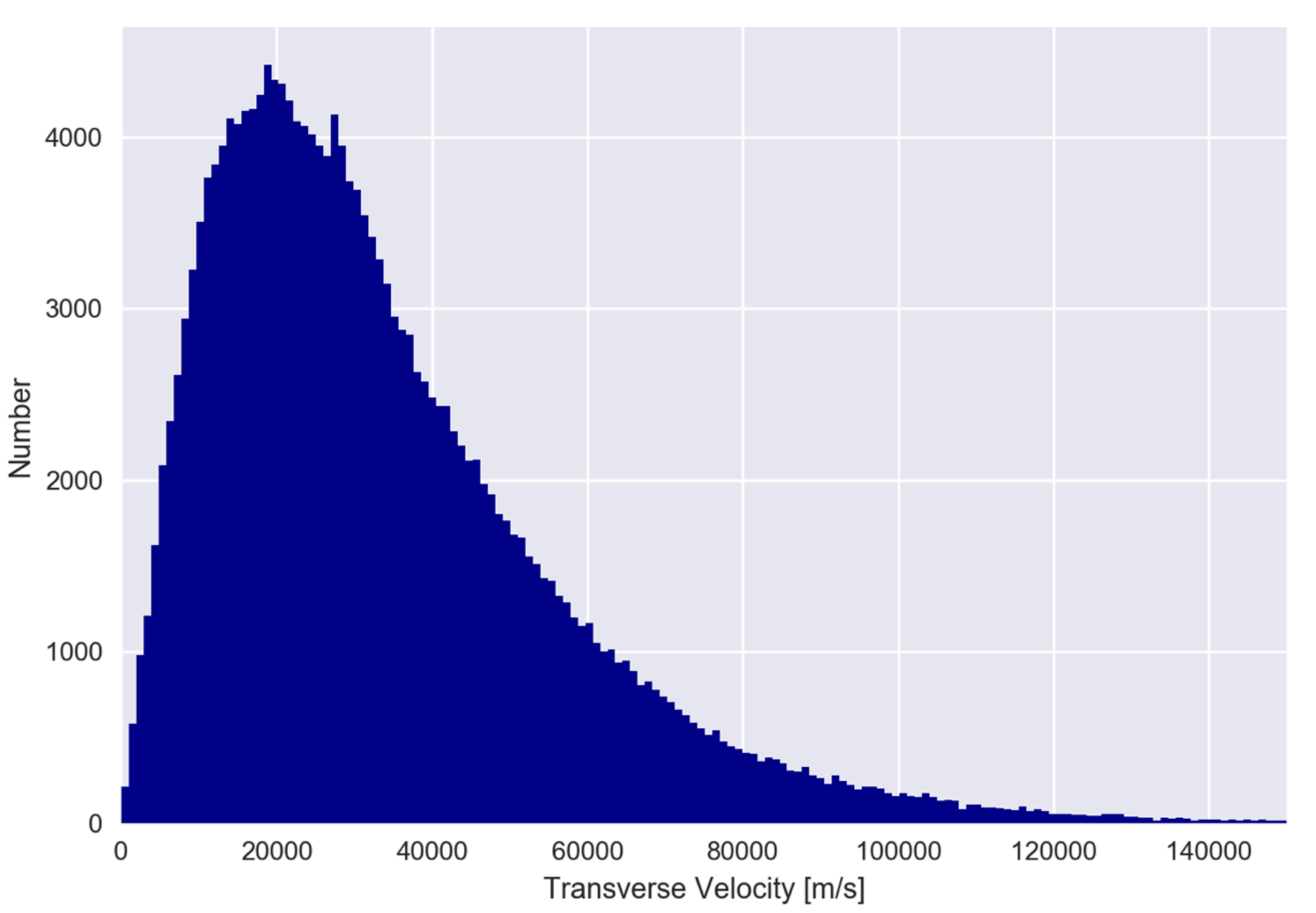
μ_α: proper motion right ascension, μ_δ: proper motion declination. Unit: milliarcsec/year

Proper motion= $\sqrt{(\text{proper motion right ascension})^2 + (\text{propermotion declination})^2}$

Transverse velocity = 4.7 × proper motion × distance

```python
import numpy as np
import matplotlib.pyplot as plt
import astropy.io.fits as fits
import seaborn as sns
sns.set(color_codes=True)
import scipy

# ================ Import data using class ==================

class DATA(object):
    def __init__(self):
        # read in data from Gaia file
        self.file=fits.open('tgas-source.fits')
        self.data_list=self.file[1]

# ================ Select data with low noise ==================

# Choose parallax data
data=DATA()
parallax=data.data_list.data['parallax'] # in mas(milliarcsecond) = 0.001 arcsecond = 1/3600000 degree
parallax_error=data.data_list.data['parallax_error'] # error

# Calculate signal to noise ratio
ratio=parallax/parallax_error

# Select data that we want
highSNindices = ratio > 16. # The ones with high signal to noise ratio

# Calculate distances of the stars from valid data
distance=1./parallax[highSNindices] # in Kpc = 1000 parsecs = 3262 light-years

# ================ Calculate velocity from proper motion ==================

# Calculate proper motion right ascension and declination
pmra = data.data_list.data['pmra'] # in mas/year
pmdec = data.data_list.data['pmdec'] # in mas/year

# Calculate transverse velocity using: v = 4.74*(proper motion angular velocity[arcsec/year])*distance[parsec]*10**3 in m/s
transv_ra = 4.74*pmra[highSNindices]*distance*10**3 # in m/s
transv_dec = 4.74*pmdec[highSNindices]*distance*10**3 # in m/s
transverse_vsqared = transv_ra**2+transv_dec**2 # in (m/s)^2
transverse_v = transverse_vsqared**(.5) # in m/s
```

Distribution of Transverse Velocity



This is the histogram of all transverse velocities calculated from proper motion.

The data is measured relatively to the detector.

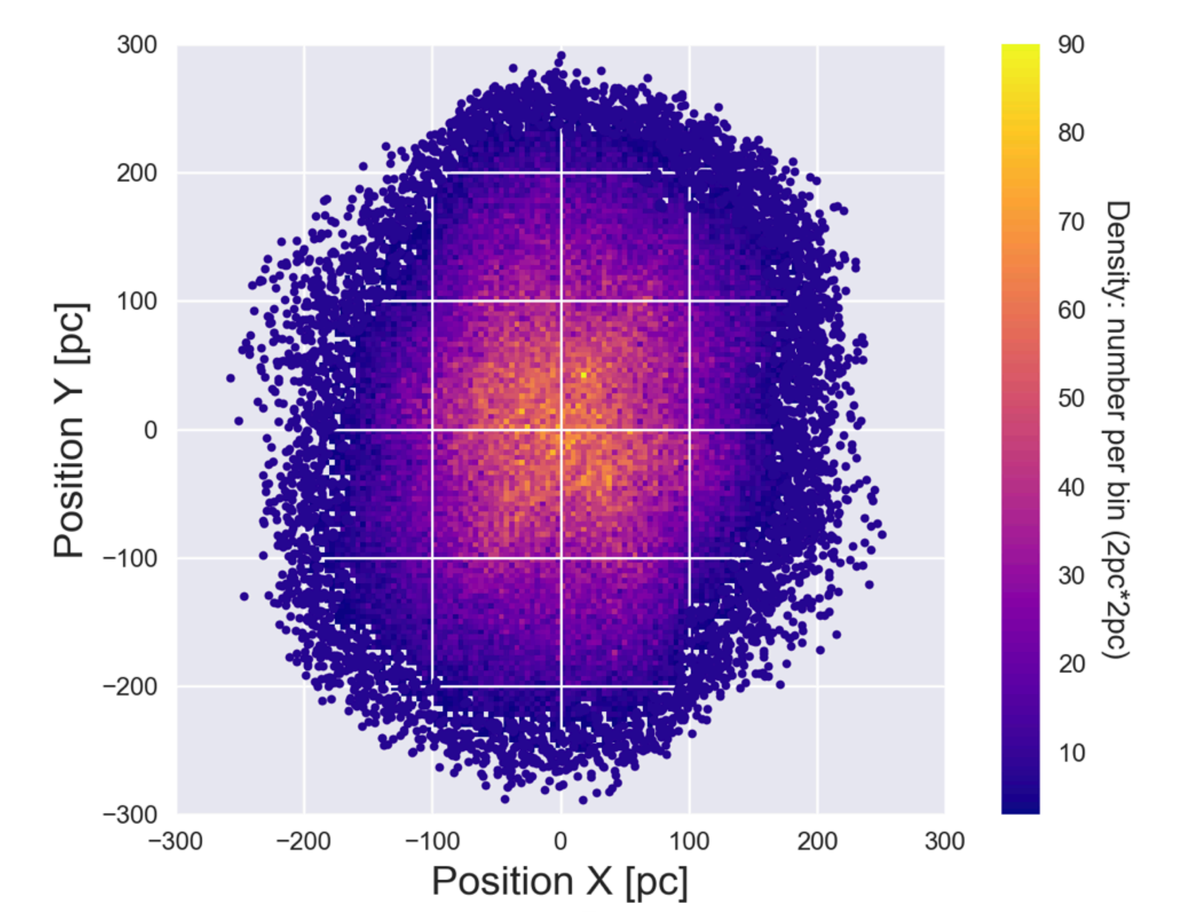
The peak value is around 18 km/s.

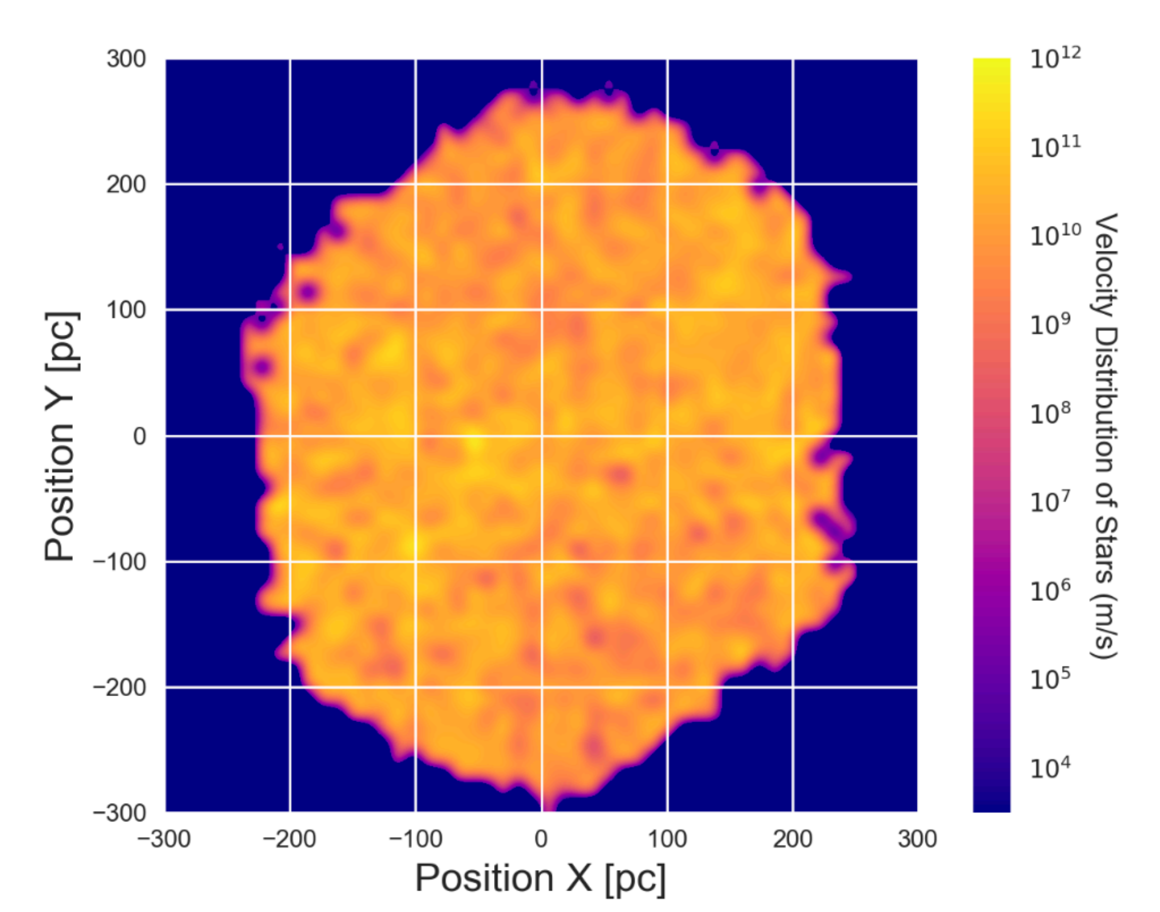
In comparison, the Sun travels around the center of the Milky Way at a speed of about 220 km/s at a radius of 8±0.65 kPc from the center.

```python
# ================ Visualize stars' 2D density distribution ===================

# Get the coordinates for plot
par = parameters()

# Print maximum and minimum values of position and velocity
print ("Max value on X: ", par.X.max())
print ("Min value on X: ", par.X.min())
print ("Max value on Y: ", par.Y.max())
print ("Min value on Y: ", par.Y.min())
print ("Min value on V: ", par.V.min())

# Set the plot's parameters
xyrange = [[-300,300],[-300,300]] # data range
bins = [150,150] # number of bins
thresh = 3  # density threshold

# Select points above density threshold within the plot
hh, locx, locy = scipy.histogram2d(par.X, par.Y, range=xyrange, bins=bins)
posx = np.digitize(par.X, locx)
posy = np.digitize(par.Y, locy)
ind = (posx > 0) & (posx <= bins[0]) & (posy > 0) & (posy <= bins[1])
hhsub = hh[posx[ind], posy[ind]] # values of the histogram where the points are
xdat1 = par.X[ind][hhsub < thresh] # low density points
ydat1 = par.Y[ind][hhsub < thresh] # low density points
hh[hh < thresh] = np.nan # Fill the areas with low density by NaNs

# Make the 2D plot
plt.figure(2)
plt.imshow(np.flipud(hh.T),cmap='plasma',extent=np.array(xyrange).flatten(), interpolation='none', origin='upper')
clb=plt.colorbar()
clb.ax.set_yticklabels(['10','20','30','40','50','60','70','80','90'],fontsize=10)
clb.set_label('Density: number per bin (2pc*2pc)', rotation=270, fontsize=12)
clb.ax.xaxis.set_label_coords(3.3, 0.5)
clb.ax.yaxis.set_label_coords(3.3, 0.5)
plt.plot(xdat1, ydat1, '.',color=sns.color_palette(palette="plasma", n_colors=20)[0])
plt.xlabel('Position X [pc]', fontsize = 16)
plt.ylabel('Position Y [pc]', fontsize = 16)
```

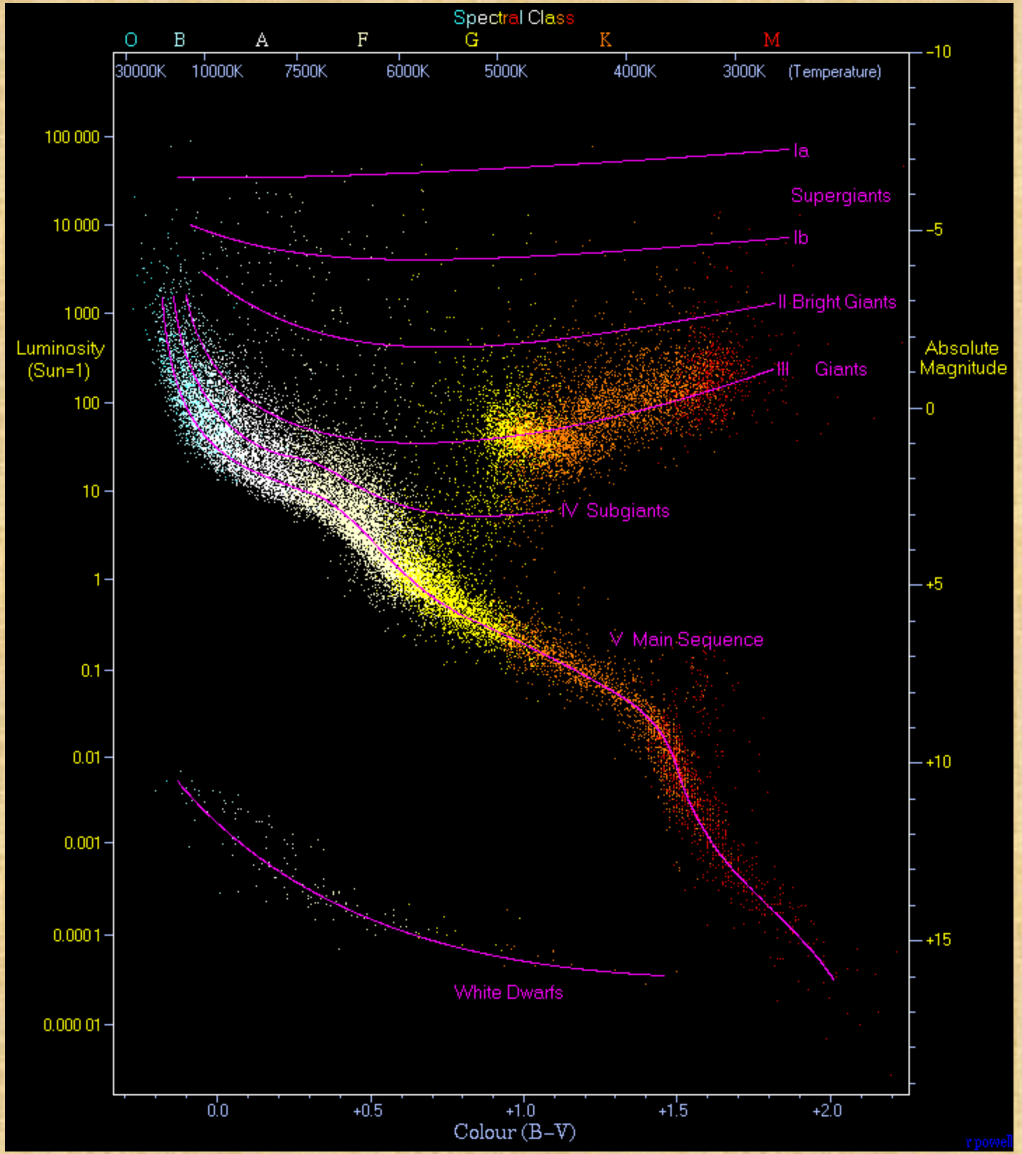
Density Distribution VS. Velocity Distribution

- Density of detected stars is higher near the center because of detector resolution
- There is no apparent correlation between relative positions and transverse velocities.





Hertzsprung–Russell Diagram

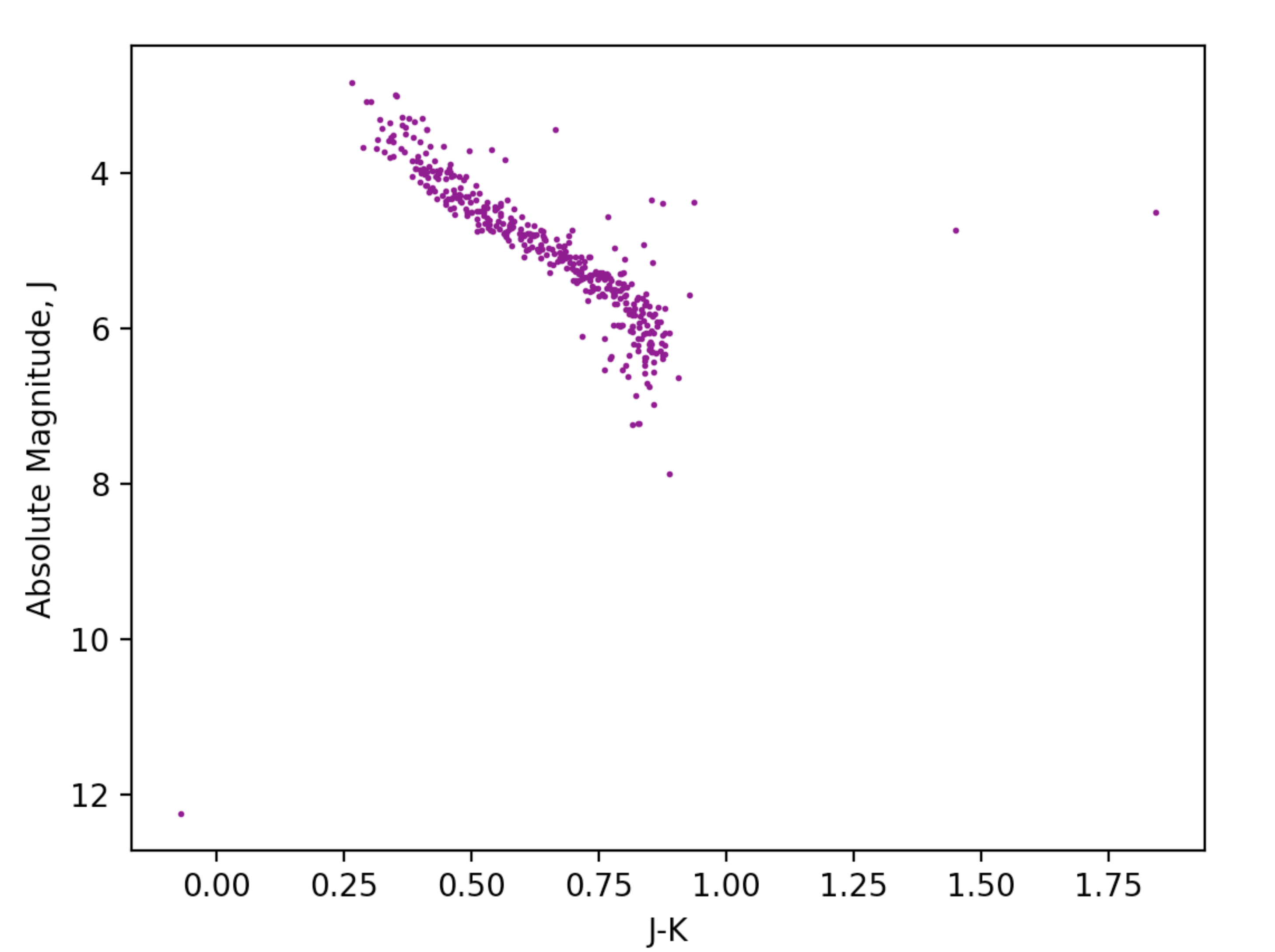


Scatter plot of stars showing the relationship between the stars' absolute magnitudes and their effective temperature.

Because 2MASS is measuring spectrum around 2-micrometer wavelength, the observed sources, such as brown dwarfs, are expected to have low temperatures.

An observational Hertzsprung–Russell diagram with 22,000 stars plotted from the Hipparcos Catalogue and 1,000 from the Gliese Catalogue of nearby stars.

Hertzsprung–Russell Diagram



J-K band spans from 1.25 µm to 2.15 µm, representing the peak wavelength of the measured stars.

According to blackbody radiation, we have:

$$\lambda_{peak}T = 2.898 \times 10^{-3}\ \text{m}\cdot\text{K}$$

The temperature of the selected stars (with signal to noise ratio > 128), spans from 1751 K to 2127 K

Thank you!