

File: hwk8.ipynb

Name: Dyab Asdi


EID: DA32435

Date: 3/31/2025

▼ Import Libraries

```
import pandas as pd
import numpy as np
from astropy.coordinates import Angle
import astropy.units as u
import plotly.express as px
```

```
from google.colab import drive
drive.mount('/content/drive')
```

 Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

▼ 1) Read Data

```
df = pd.read_csv("/content/sample_data/bsc5.csv")
```

▼ 2) Drop all objects without a trigonometric parallax

```
df = df.dropna(subset=['Parallax'])
df = df[df['Parallax'] > 0]
df = df[df['Parallax'] > 0]
df = df[df['n_Parallax'] != 'D']
```

▼ 3) Create a column called Distance with the object's distance in parsec

```
df['Distance'] = 1 / df['Parallax']
```

▼ 4) RA and DEC conversion + conversion to Cartesian Coordinate system

```
# convert RA to deg
df['RA_deg'] = Angle(df['RAh']*15 + df['RAm']*15/60 + df['RAs']*15/3600, unit=u.deg).degree
```

```
# convert dec to deg
df['Dec_deg'] = Angle(df['DEd'] + df['DEm']/60 + df['DEs']/3600, unit=u.deg).degree
```

```
# cartesian coordinates
ra_rad = np.deg2rad(df['RA_deg'])
dec_rad = np.deg2rad(df['Dec_deg'])
distance = df['Distance']
```

```
df['x'] = distance * np.cos(dec_rad) * np.cos(ra_rad)
df['y'] = distance * np.cos(dec_rad) * np.sin(ra_rad)
df['z'] = distance * np.sin(dec_rad)
```

5) Create a column called Color in your data frame. The colors could be named colors like red or blue. Or, the colors could be in hexadecimal format. Here is the color scheme according to spectral types - O (deepest blue), B (medium blue), A (light blue), F (green), G (yellow), K (orange), M (red), and any other star not having a spectral type black.

```
# function for assigning color based on sp type
def assign_color(sp_type):
    color_map = {
        'O': '#0000FF', # deepest blue
        'B': '#4169E1', # medium blue
        'A': '#87CEEB', # light blue
        'F': '#008000', # green
        'G': '#FFD700', # yellow
        'K': '#FFA500', # orange
        'M': '#FF0000', # red
        'Other': '#000000' # black
    }
    if pd.isnull(sp_type) or not isinstance(sp_type, str):
        return '#000000'
    first_char = sp_type[0]
    if first_char in 'OBAFGKM':
        return color_map[first_char]
    return '#000000'

df['Color'] = df['SpType'].apply(assign_color)
```

6) Use Plotly to obtain the 3-D distribution of the stars.

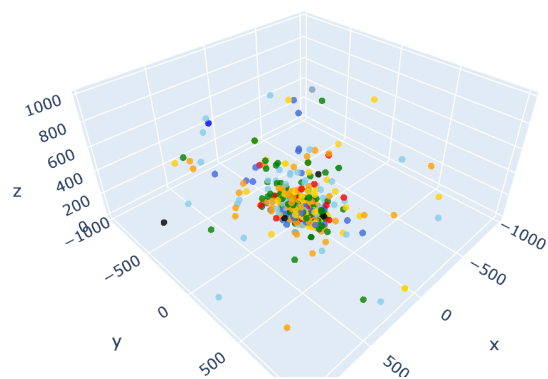
```
fig = px.scatter_3d(
    df, x='x', y='y', z='z',
    color='Color',
    hover_data={
        'HR Number ': df['HR'],
        'Distance (parsec) ': df['Distance'],
        'Radial Velocity (km/s) ': df['RadVel']
    },
    title="3-D Distribution of Stars",
)

# change the makers
fig.update_traces(
    marker=dict(color=df['Color'],size=3, opacity=0.8)
)

fig.show()
```



3-D Distribution of Stars



Start coding or [generate](#) with AI.

