A10 - 3D Model of the Universe

In [335…
```
# Imports
import pandas as pd
import plotly
import plotly.express as px
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode
import astropy
from astropy import units as u
from astropy.coordinates import SkyCoord
import numpy as np
```

In [336…
```
# Read in dataset
df = pd.read_csv("Data/bsc5.csv")
```

In [337…
```
# Drop all objects that don't have trigonometric parallax
df = df[df['n_Parallax'] != 'D']
df = df[df['Parallax'].notnull()]
df = df.reset_index()
```

In [338…
```
# Add distance in parsecs (d_pc = 1 / p")
df['Distance'] = 1 / df['Parallax']
```

In [339…
```
# Convert RA to numbers and drop all NaN values
df['RAh'] = pd.to_numeric(df['RAh'], errors='coerce').dropna()
df['RAm'] = pd.to_numeric(df['RAm'], errors='coerce').dropna()
df['RAs'] = pd.to_numeric(df['RAs'], errors='coerce').dropna()
```

In [340…
```
# Convert DE to numbers and drop all NaN values
df['DEd'] = pd.to_numeric(df['DEd'], errors='coerce').dropna()
df['DEm'] = pd.to_numeric(df['DEm'], errors='coerce').dropna()
df['DEs'] = pd.to_numeric(df['DEs'], errors='coerce').dropna()
df = df.dropna(subset=['DE'])
```

In [341…
```
# Reset the indices so no error
df = df.reset_index()
```

In [342…
```
# Make the signs into "+1" or "-1" for easier multiplication
df['DE'] = df['DE'] + "1"
```

In [343…
```
# Convert Right Ascension and Declination into degrees
df['RA'] = 15 * (df['RAh'] + df['RAm'] / 60 + df['RAs'] / 3600)
df['DEC'] = pd.to_numeric(df['DE'], errors='coerce')\
            * (df['DEd'] + df['DEm'] / 60 + df['DEs'] / 3600)
```

In [344…
```
# Calculate the x, y, and z coordinates of each star using algebra
tan = lambda angle: np.tan(angle * np.pi/180)
df['x'] = df['Distance'] / np.sqrt(tan(df['DEC']) ** 2 + (tan(df['RA'])\
            * tan(df['DEC'])) ** 2 + 1 + tan(df['RA']) ** 2)
df['y'] = df['x'] * tan(df['RA'])
df['z'] = np.sqrt(df['Distance'] ** 2 - df['x'] ** 2 - df['y'] ** 2)
```

In [345…
```
# Make colors
color_dict = {"O": "#0000ff", "B": "#5959ff", "A": "#8f8fff", "F": "#00ff00",
              "G": "#fff200", "K": "#ff8800", "M": "#ff0000"}
df['Sp'] = df['SpType'].apply(lambda x: \
            x[1:] if x.startswith("d") or\
            x.startswith("g") else x).astype(str).str[0]
```

```
df['Color'] = df['Sp'].apply(lambda sp: \
            [color_dict[sp] for c in sp\
            if c in color_dict]).astype(str).str[2:-2]
```

In [346…

```
# Formats a number as x * 10^y
def format_exp(x):
    return (x[0] if x[0] != "-" else x[1])  + "."\
    + (x[1:4].rstrip('0') if x[0] != "-" else x[2:5])\
    + f" x 10<sup>{len(x[1:])}</sup>"
```

In [350…

```
# Draw the Plotly Graph
init_notebook_mode(connected=True)
df["Distance"] = df["Distance"].apply(abs)
df["Miles"] = df["Distance"] * 1.917e+13
df["LTYR"] = df["Distance"] * 3.26
df["Miles"] = df["Miles"].astype(str).apply(format_exp)
df["LTYR"] = df["LTYR"].astype(str).apply(format_exp)
df["DistanceNice"] = df["Distance"].astype(str).apply(format_exp)
replace_whitespace = lambda x: " ".join(x.split())
df["Name"] = df["Name"].astype(str).apply(replace_whitespace)
df.replace("nan", "<i>Unknown</i>", inplace=True)
fig = px.scatter_3d(
        df,
        x='x', y='y', z='z',
        color='Color',
        hover_name="Name",
        hover_data={
            "HR": True,
            "DistanceNice": True,
            "RadVel": True,
            "Sp": True,
            "x": False,
            "y": False,
            "z": False,
            "Color": False
        },
        custom_data=["Name", "HR", "DistanceNice", "LTYR", "Miles", "RadVel", "Sp"]
    )
hovertemplate=\
"""
<b>Star Info</b><br>
<i>Name</i>: %{customdata[0]}<br>
Bright Star Number: %{customdata[1]}<br>
Distance (parsecs): %{customdata[2]}<br>
Distance (lt yr): %{customdata[3]}<br>
Distance (miles): %{customdata[4]}<br>
Radial Velocity (km/s): %{customdata[5]}<br>
Spectral Type: %{customdata[6]}<extra></extra>
"""
fig.update_traces(hovertemplate=hovertemplate)
fig.update_layout(
    title=go.layout.Title(
        text="<b>3D Model of the Stars in the Universe<br><br><sup>Data from the Yale
    ), showlegend=False)
fig.show()
```

3D Model of the Stars in the Universe

Data from the Yale Bright Star Catalog