



Data-Driven Insights Into The Cosmos



DATA ANALYSIS: UNCOVERING THE METHODS OF EXOPLANET DETECTION



```
In [37]: #Importing necessary Libraries
import plotly.express as px
import matplotlib.pyplot as plt
import pandas as pd

In [38]: # Load the data into a pandas DataFrame
df = pd.read_csv("Exoplanets_Tess.csv", header=9)

In [39]: # Check the shape of the DataFrame and number of missing values
print("Shape of the DataFrame:", df.shape)
print("Number of missing values:", df.isnull().sum().sum())

Shape of the DataFrame: (33873, 4)
Number of missing values: 31681

In [40]: # Remove rows with missing values
df = df.dropna()

In [41]: # Check the shape of the cleaned DataFrame
print("Shape of the cleaned DataFrame:", df.shape)

Shape of the cleaned DataFrame: (4506, 4)

In [44]: # Print the unique discovery methods in the dataset
print("Discovery methods:", df["Discovery Method"].unique())

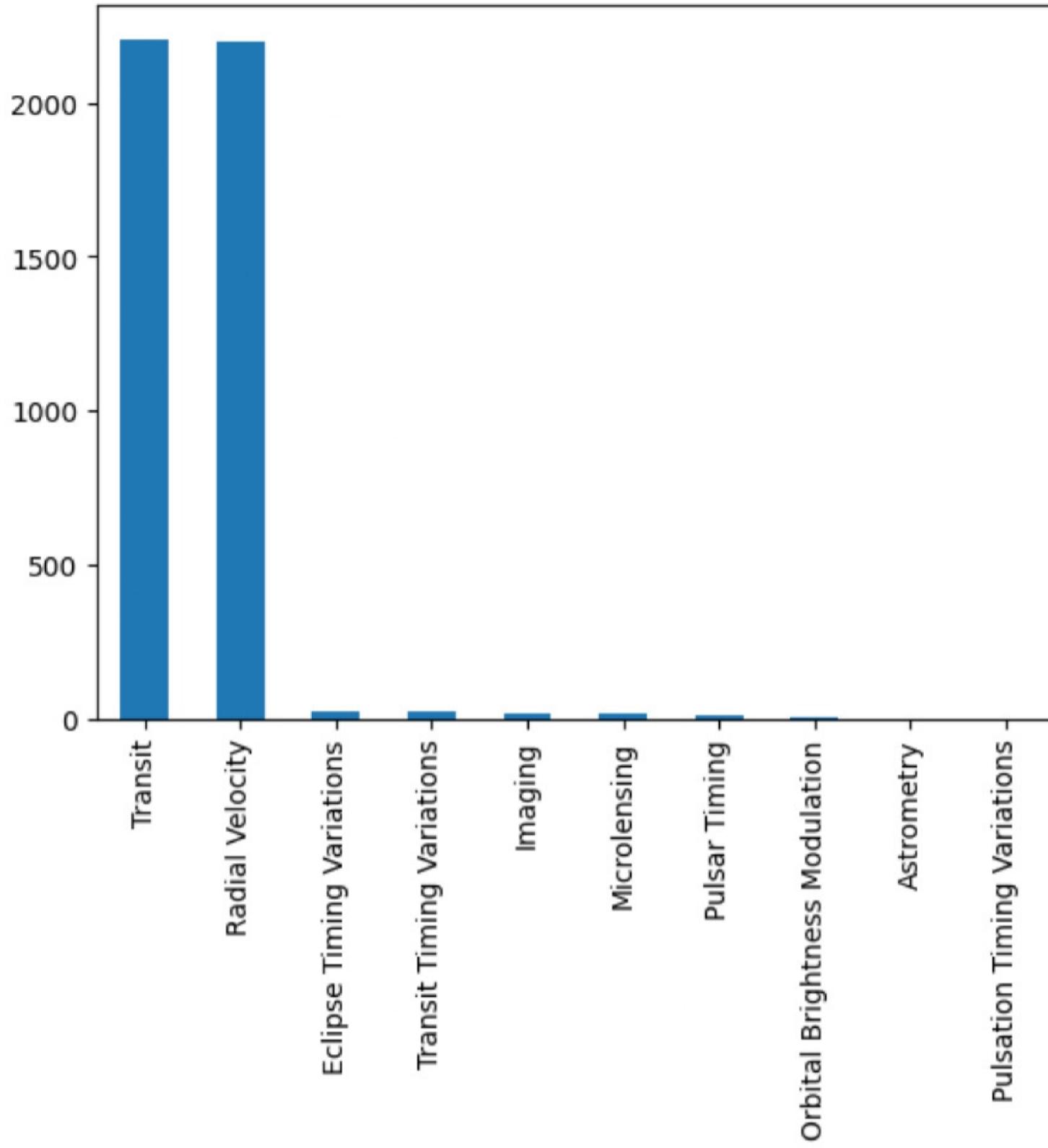
Discovery methods: ['Radial Velocity' 'Eclipse Timing Variations' 'Transit' 'Imaging'
 'Astrometry' 'Orbital Brightness Modulation'
 'Pulsation Timing Variations' 'Transit Timing Variations' 'Microlensing'
 'Pulsar Timing']

In [45]: #Finding max and min values in Orbital Period and Mass of Planet
df['pl_orbper'].max(), df['pl_orbper'].min()
df['pl_bmassj'].max(), df['pl_bmassj'].min()

Out[45]: (80.0, 5e-05)

In [46]: discoverymethod_counts = df['Discovery Method'].value_counts()
discoverymethod_counts.plot(kind='bar')
plt.show()
```



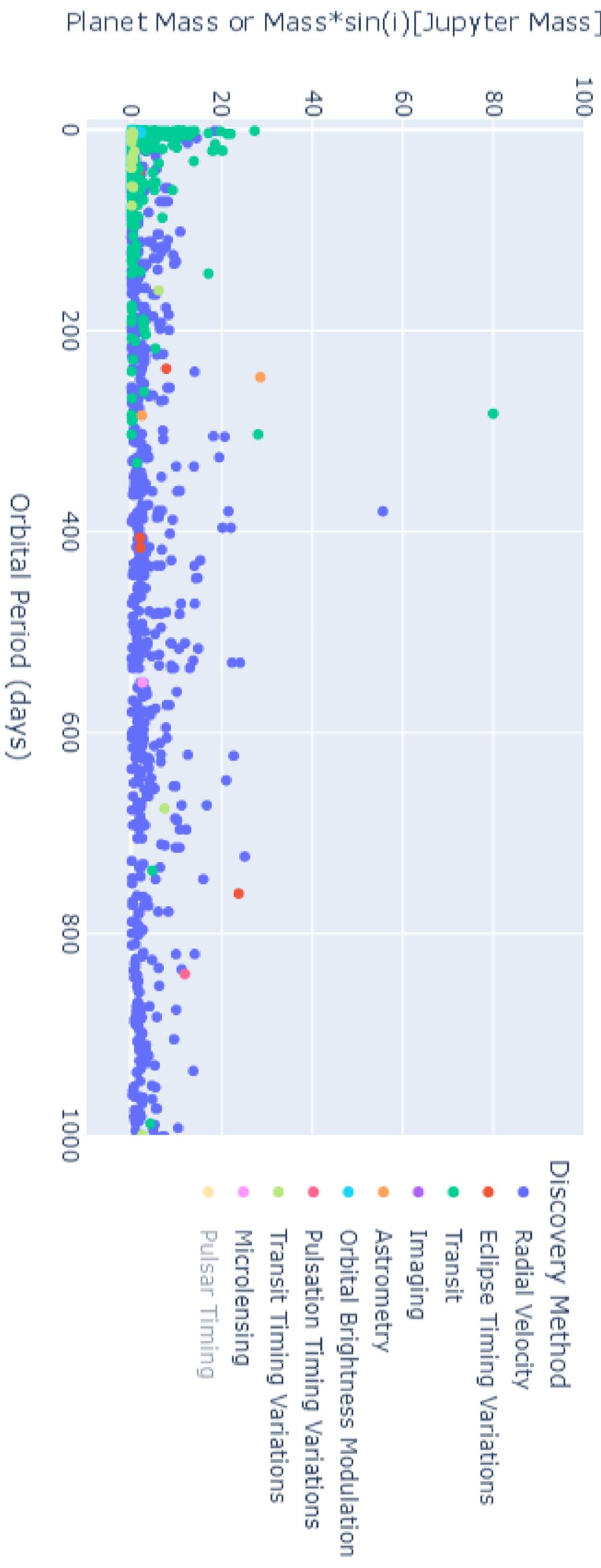


```
In [47]: # Define a dictionary to map each discovery method to a different marker shape
# marker_shapes = {"Method 1": "square", "Method 2": "diamond", "Method 3": "star"}
# df["marker_shape"] = df["discoverymethod"].map(marker_shapes)
```

```
In [49]: # Create the scatter plot
fig = px.scatter(df, x="pl_orbper", y="pl_bmassj", color="Discovery Method",
                  title="Exoplanet Discovery Methods", hover_name="Discovery Method",
                  render_mode="webgl", size_max=15)
```

```
In [51]: # Set the lower and upper bounds of the x-axis and y-axis
fig.update_xaxes(title_text="Orbital Period (days)", range=[-10, 1000])
fig.update_yaxes(title_text="Planet Mass or Mass*sin(i)[Jupyter Mass]", range=[-10, 1000])
# Show the plot
fig.show()
```

Exoplanet Discovery Method





Data-Driven Insights Into The Cosmos