Astronomy is the science that deals with the study of celestial bodies in the universe, including planets and their satellites, comets and meteorites, stars and interstellar matter, systems of dark matter, gas and dust called galaxies and clusters of galaxies. Most of the information used by astronomers is collected by remote observation, although in some cases it has been possible to reproduce the execution of celestial phenomena in the laboratory. It is also one of the few sciences in which amateurs can still play an active role, especially in the discovery and monitoring of phenomena such as light curves of variable stars, discovery of asteroids and comets, etc.

In this workshop, you will work with data from all confirmed extrasolar planets (called exoplanets), discovered between 1988 and 2018 (more than 3000). This information is compiled in the database called Open Exoplanet Catalogue (https://www.kaggle.com/mrisdal/open-exoplanet-catalogue). In the file "exoplanets.csv" you will find a simplified version of the original data. We removed from this catalog planets that have not been confirmed as such, eliminated some columns, and modified classifications from numerical values to categorical values.

The data fields include attributes of stars and planets, discovery methods, and (of course) discovery dates.

Part 1: Loading the Data

Load data from a CSV file into a DataFrame. The user should be asked for the name of the file. The function that implements this option should receive the file name as a parameter and return a DataFrame.

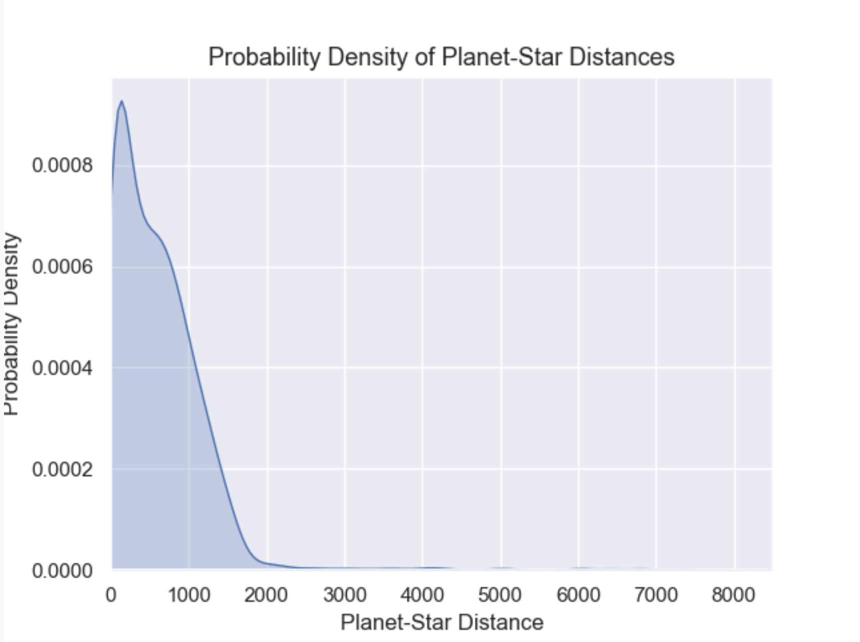
The columns of the file and their meanings are as follows:

- **NAME**: name of the discovered planet
- **MASS**: mass of the planet
- **DISCOVERY**: year of discovery of the planet
- **UPDATE**: last time the information about the planet was updated
- **PUBLICATION_STATUS**: medium in which the discovery was published: Announced on a professional conference, Announced on a website, Published in a refereed paper, Submitted to a professional journal
- **DETECTION_TYPE**: type of detection used to discover the planet: Astrometry, Imaging, Microlensing, Other, Primary Transit, Primary Transit, TTV
- **RA**: refers to Right Ascension, which is one of the coordinates of the planet as seen from Earth
- **DEC**: refers to Declination, which is the other coordinate of the planet as seen from Earth
- **STAR_DISTANCE**: distance to the nearest star
- **STAR_MASS**: mass of the nearest star

Help: When studying the problem and the file, remember that the functions `describe()` and `unique()` can be useful. The `describe()` function applied to a DataFrame returns statistical information about all numerical columns. The `unique()` function, applied to a column, returns a list of values that appear in that column.

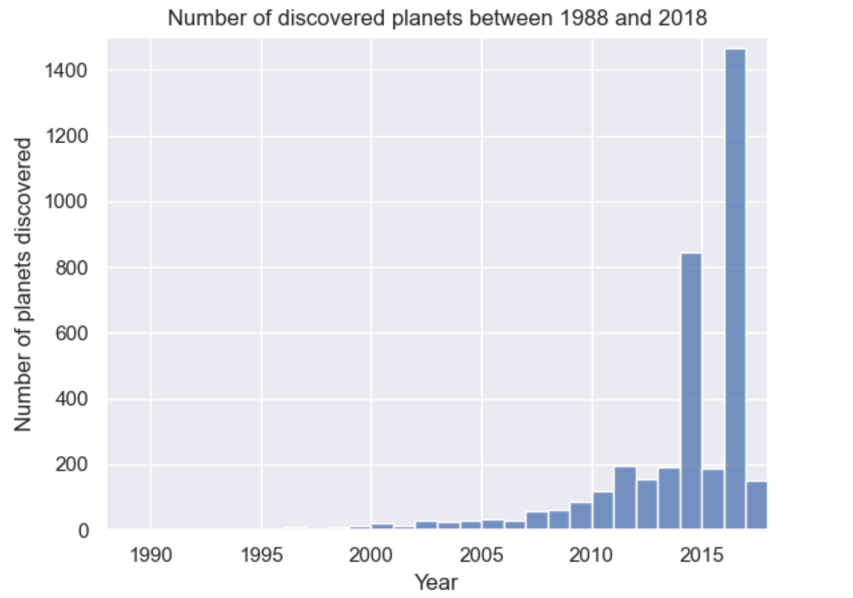
Part 2: Studying the Distances Between Planets and Their Nearest Star

In this part of the application, the distribution of the distances of each planet to its nearest star should be shown. To do this, a KDE (kernel density estimation) graph should be generated using the column `STAR_DISTANCE`. KDE graphs are a way to estimate the probability density function of a random variable, which in this case is the distance between a planet and its nearest star. These graphs are based on curves and can be seen as a smoothed version of a histogram that attempts to fill in the gaps in the sample. Due to the effect of introducing these curves, the default graphs show that it is possible for a planet to have a negative distance to its nearest star. To avoid this problem, it is necessary to define explicit limits for the graph using the parameter `xlim`, which corresponds to a tuple with the minimum value (0) and the maximum expected distance value (8500). The following figure shows the appearance of this graph.

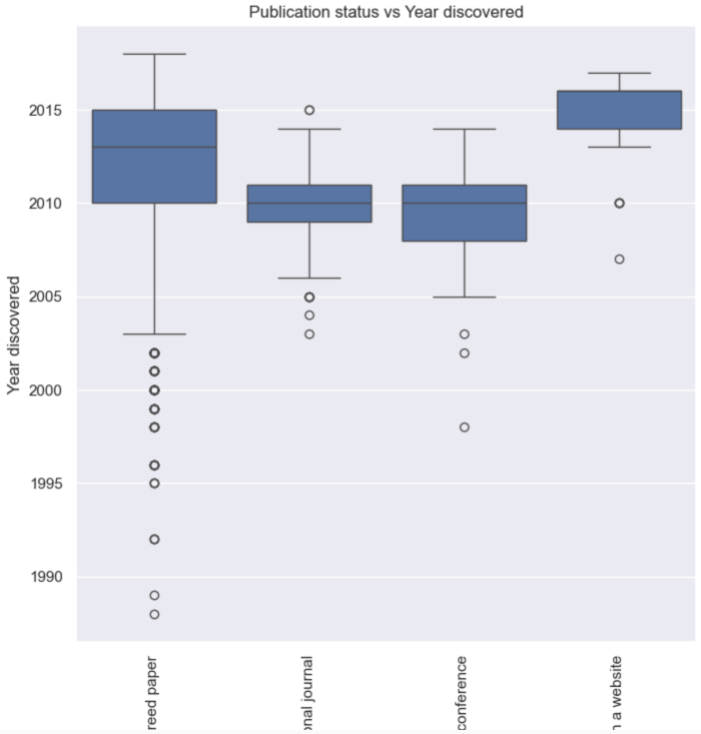


Part 3: Analyzing the Discovery of Planets

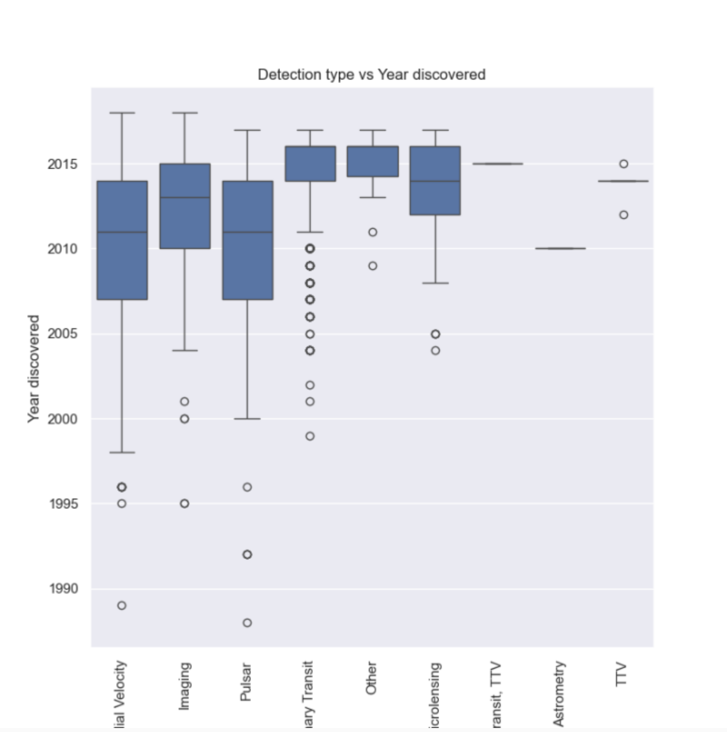
The first option of this part of the application should calculate and graph a histogram showing how many planets were discovered over the years. To do this, a histogram graph should be created using the values from the `DISCOVERY` column. The histogram should separate the data into 30 groups (bins). This way, the number of planets discovered each year between 1998 and 2018 can be appreciated. The following figure shows the appearance of this graph.



The second option of this part should show the relationship between the years of discovery of the planets and the types of publication used to report the discoveries. To do this, a boxplot graph should be created using the **`DISCOVERY`** column and grouping the data according to the **`PUBLICATION_STATUS`** column. The following figure shows the appearance of this graph.



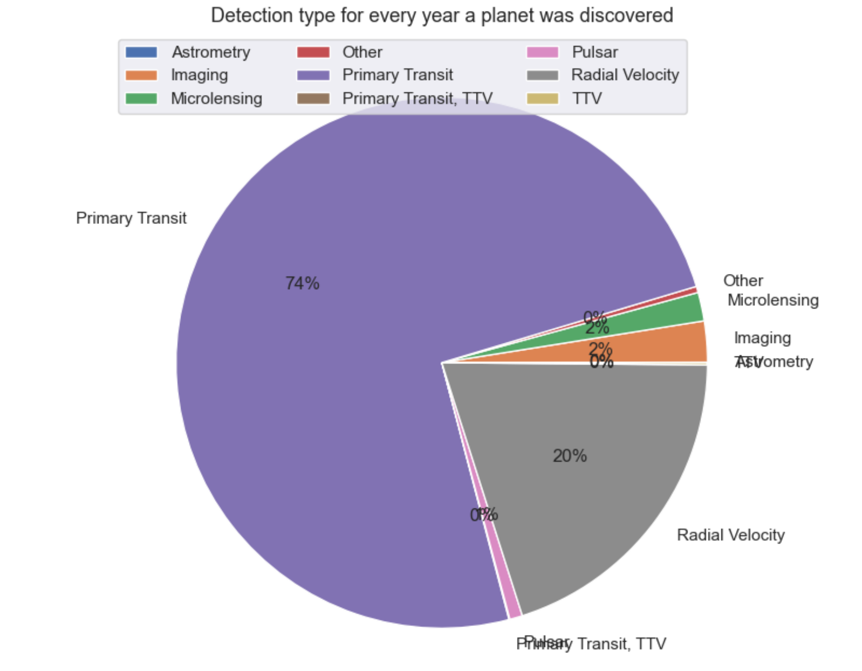
The third option of this part should show the relationship between the years of discovery of the planets and the types of detections used. To do this, a second boxplot graph should be created using the **DISCOVERY** column and grouping the data according to the **DETECTION_TYPE** column. The following figure shows the appearance of this graph.



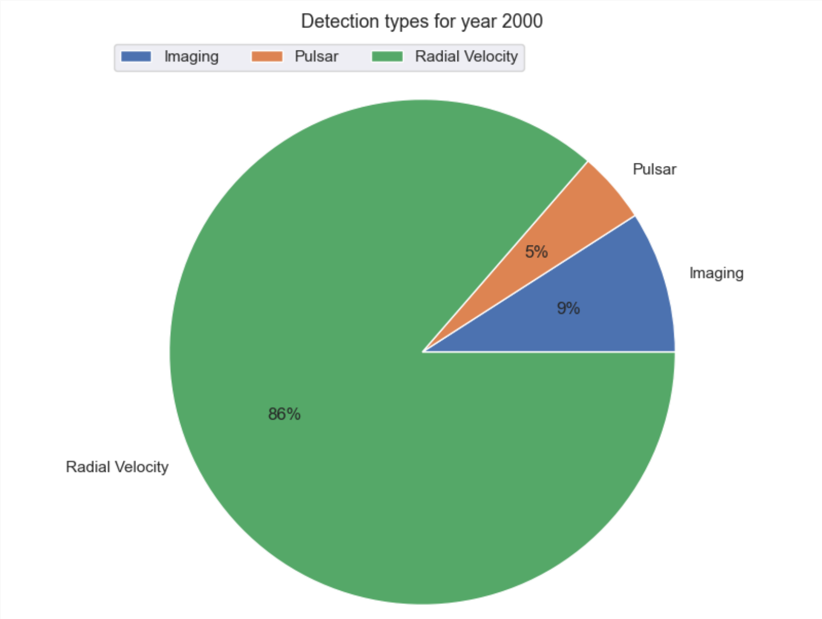
The fourth option of this part of the program should generate two pie charts that show:

- The types of detection for all the years in which a planet was discovered, and
- The types of detection for a specific year in which a planet was discovered.

The user should be given the option to graph all years or to enter a specific year of their choice into the program (e.g., input 0 to plot all years). The following figure shows the appearance of the pie chart considering all the years:



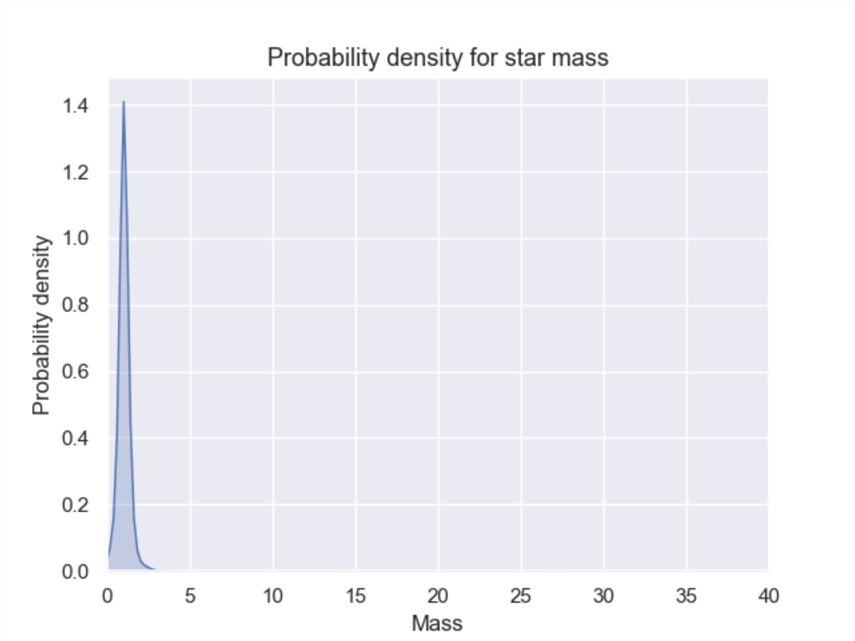
The following figure shows the appearance of the pie chart considering only the year 2000:

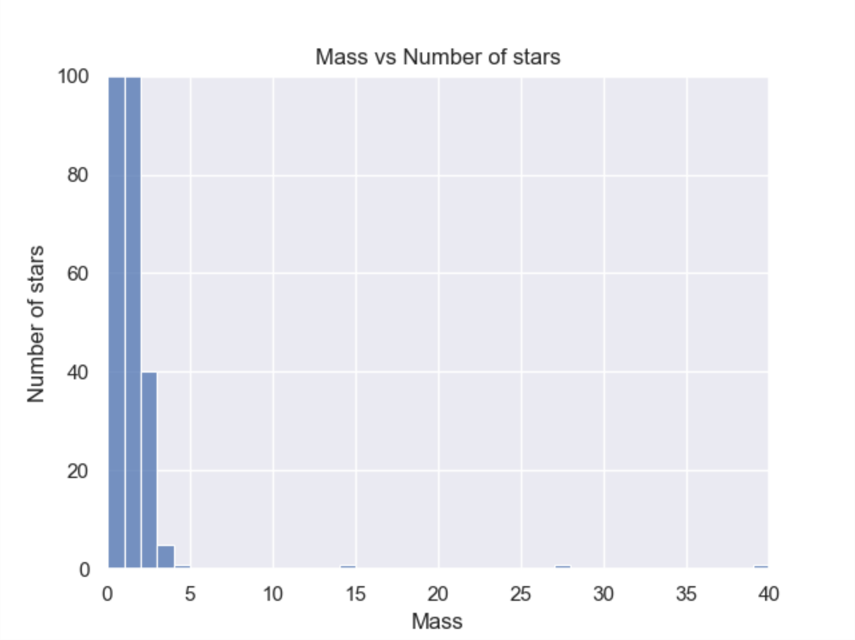


Help: You can configure the size of the figure with the parameter figsize=(8,8) when creating the figure. For example: plt.figure(figsize=(8, 8)). To ensure the values shown in the pie chart correspond to percentages, you can use the parameter: autopct='%1.1f%%' when creating the pie chart.

Part 4: Studying the Mass of the Stars Closest to the Planets

In this part, a KDE graph should be generated to analyze the mass of the stars closest to the planets and compare this graph with a histogram of the same information, with the purpose of corroborating the similarity of both graphs. The following figure shows the appearance of the two graphs:





Help:

- Configure the range of the x-axis of the two figures between 0 and 40 (where 40 is the maximum mass value). To do this, use the `xlim` method like this: `plt.xlim(0,40)`.
- Configure the range of the y-axis of the histogram between 0 and 100. To do this, use the `ylim` method like this: `plt.ylim(0,100)`.