Лабораторная работа №1

Выполнена студентом группы ИВТ-б-о-22-1: Репкин Александр

Исследование набора данных - Небесные Тела.

Подключение библиотеки NumPy и загрузка данных

In [ ]:
```
!wget https://raw.githubusercontent.com/AlexRepkin/Machine-Learning/main/Act-1-Data%20Analysis/Cosmic%20bodies.data
# Загрузка данных из своего репозитория GitHub.
```

```
--2024-02-21 08:04:26--  https://raw.githubusercontent.com/AlexRepkin/Machine-Learning/main/Act-1-Data%20Analysis/Cosmic%20bodies.data
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3379 (3.3K) [text/plain]
Saving to: ‘Cosmic bodies.data’

Cosmic bodies.data  100%[===================>]   3.30K  --.-KB/s    in 0s

2024-02-21 08:04:26 (27.9 MB/s) - ‘Cosmic bodies.data’ saved [3379/3379]
```

In [11]:
```
import numpy as np
data_path = "Cosmic bodies.data"
data = np.genfromtxt(data_path, delimiter=",")
print(data)
```

```
[[ 2.43970e+03 -1.00000e+02  0.00000e+00  3.30000e+02          nan]
 [ 6.05180e+03  1.50000e+01  2.00000e+00  4.86800e+03          nan]
 [ 6.37100e+03  1.50000e+01  1.00000e+00  5.97360e+03          nan]
 [ 3.38950e+03 -6.30000e+01  2.00000e+00  6.41710e+02          nan]
 [ 6.99110e+04 -1.45000e+02  8.20000e+01  1.89819e+03          nan]
 [ 5.79600e+02  4.62000e+02  0.00000e+00  7.30000e-02          nan]
 [ 7.14920e+04 -1.45000e+02  7.90000e+01  5.68340e+02          nan]
 [ 6.02680e+04 -2.18000e+02  6.20000e+01  8.68130e+01          nan]
 [ 2.55590e+04  5.20000e+01  2.70000e+01  1.02413e+02          nan]
 [ 5.43640e+04 -2.14000e+02  1.40000e+01  5.68340e+02          nan]
 [ 2.37000e+03 -5.30000e+01  0.00000e+00  3.30000e-01          nan]
 [ 4.86140e+03  1.67000e+02  0.00000e+00  6.42000e-01          nan]
 [ 2.50000e+01  1.00000e+02  0.00000e+00  2.00000e-06          nan]
 [ 1.27560e+04  1.50000e+01  1.00000e+00  5.97200e+24          nan]
 [ 4.95280e+04 -2.24000e+02  2.70000e+01  4.86700e+24          nan]
 [ 1.42984e+05 -2.34000e+02  7.90000e+01  5.97200e+24          nan]
 [ 1.20536e+05 -2.45000e+02  6.20000e+01  5.97200e+24          nan]
 [ 5.11180e+04 -2.11000e+02  2.70000e+01  5.97200e+24          nan]
 [ 4.95280e+04 -2.13000e+02  1.40000e+01  5.97200e+24          nan]
 [ 4.92440e+04 -2.34000e+02  6.20000e+01  5.97200e+24          nan]
 [ 5.24140e+04 -2.21000e+02  2.70000e+01  5.97200e+24          nan]
 [ 6.02680e+04 -2.45000e+02  2.70000e+01  5.97200e+24          nan]
 [ 6.96340e+05 -2.43000e+02  8.20000e+01  5.97200e+24          nan]
```

```
 [ 7.14920e+05 -2.30000e+02  7.90000e+01  5.97200e+24          nan]
 [ 4.95280e+05 -2.15000e+02  6.20000e+01  5.97200e+24          nan]
 [ 5.41080e+04 -2.15000e+02  2.70000e+01  5.97200e+24          nan]
 [ 6.02680e+04 -2.36000e+02  1.40000e+01  5.97200e+24          nan]
 [ 4.95280e+05 -2.22000e+02  8.20000e+01  5.97200e+24          nan]
 [ 7.14920e+05 -2.30000e+02  7.90000e+01  5.97200e+24          nan]
 [ 5.41080e+04 -2.15000e+02  6.20000e+01  5.97200e+24          nan]
 [ 6.04320e+04 -2.15000e+02  2.70000e+01  5.97200e+24          nan]
 [ 6.02680e+04 -2.36000e+02  1.40000e+01  5.97200e+24          nan]
 [ 2.54820e+04 -2.56000e+02  2.70000e+01  5.97200e+24          nan]
 [ 6.59232e+05 -2.78000e+02  8.20000e+01  5.97200e+24          nan]
 [ 7.14920e+05 -2.81000e+02  7.90000e+01  5.97200e+24          nan]
 [ 5.89520e+04 -2.92000e+02  6.20000e+01  5.97200e+24          nan]
 [ 4.95280e+05 -2.76000e+02  2.70000e+01  5.97200e+24          nan]
 [ 6.02680e+04 -2.98000e+02  1.40000e+01  5.97200e+24          nan]
 [ 6.94100e+03 -2.20000e+02  0.00000e+00  1.07000e-01          nan]
 [ 1.02000e+04 -2.01000e+02  0.00000e+00  8.15000e-01          nan]
 [ 6.95700e+05  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.96340e+05  5.50500e+03  9.00000e+00  1.98900e+30          nan]
 [ 7.14920e+05  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 6.94100e+03  5.77800e+03  0.00000e+00  2.00000e+30          nan]
 [ 1.02000e+04  5.77800e+03  0.00000e+00  2.00000e+30          nan]
 [ 1.27560e+04  5.77800e+03  0.00000e+00  2.00000e+30          nan]
 [ 4.95280e+04  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 5.41080e+04  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.02680e+04  5.77800e+03  9.00000e+00  1.98900e+30          nan]
 [ 6.96340e+05  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 7.14920e+05  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.94100e+03  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.02000e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.27560e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 4.95280e+04  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 5.41080e+04  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.02680e+04  5.77800e+03  9.00000e+00  1.98900e+30          nan]
 [ 6.96340e+05  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 7.14920e+05  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.94100e+03  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.02000e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.27560e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 4.95280e+04  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 5.41080e+04  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.02680e+04  5.77800e+03  9.00000e+00  1.98900e+30          nan]
 [ 6.96340e+05  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 7.14920e+05  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.94100e+03  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.02000e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.27560e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 4.95280e+04  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 5.41080e+04  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.02680e+04  5.77800e+03  9.00000e+00  1.98900e+30          nan]
 [ 6.96340e+05  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 7.14920e+05  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 6.94100e+03  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.02000e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 1.27560e+04  5.77800e+03  0.00000e+00  1.98900e+30          nan]
 [ 4.95280e+04  5.77800e+03  7.00000e+00  1.98900e+30          nan]
 [ 5.41080e+04  5.77800e+03  8.00000e+00  1.98900e+30          nan]
 [ 5.00000e+00  5.00000e+02  0.00000e+00  1.00000e+09          nan]
 [ 4.50000e+00  6.00000e+02  0.00000e+00  5.00000e+08          nan]
 [ 3.00000e+00  7.00000e+02  0.00000e+00  3.00000e+08          nan]
 [ 2.20000e+00  8.00000e+02  0.00000e+00  2.00000e+08          nan]
```

```
 [ 6.70000e+00  4.00000e+02  0.00000e+00  1.50000e+09           nan]
 [ 4.00000e+00  5.50000e+02  0.00000e+00  4.00000e+08           nan]
 [ 3.50000e+00  6.50000e+02  0.00000e+00  2.50000e+08           nan]
 [ 2.80000e+00  7.50000e+02  0.00000e+00  2.20000e+08           nan]
 [ 6.50000e+00  4.20000e+02  0.00000e+00  1.70000e+09           nan]
 [ 5.20000e+00  4.80000e+02  0.00000e+00  6.00000e+08           nan]
 [ 4.80000e+00  5.80000e+02  0.00000e+00  4.50000e+08           nan]
 [ 3.90000e+00  6.80000e+02  0.00000e+00  3.50000e+08           nan]
 [ 2.90000e+00  7.80000e+02  0.00000e+00  2.30000e+08           nan]
 [ 7.20000e+00  4.30000e+02  0.00000e+00  1.80000e+09           nan]
 [ 6.00000e+00  5.20000e+02  0.00000e+00  7.00000e+08           nan]
 [ 5.50000e+00  6.00000e+02  0.00000e+00  5.50000e+08           nan]
 [ 4.60000e+00  6.70000e+02  0.00000e+00  3.60000e+08           nan]
 [ 3.80000e+00  7.20000e+02  0.00000e+00  2.80000e+08           nan]
 [ 2.50000e+00  7.90000e+02  0.00000e+00  2.50000e+08           nan]
 [ 7.80000e+00  4.40000e+02  0.00000e+00  1.90000e+09           nan]
 [ 6.30000e+00  5.10000e+02  0.00000e+00  8.00000e+08           nan]
 [ 5.80000e+00  5.90000e+02  0.00000e+00  5.70000e+08           nan]
 [ 4.90000e+00  6.60000e+02  0.00000e+00  4.70000e+08           nan]
 [ 4.20000e+00  7.30000e+02  0.00000e+00  3.20000e+08           nan]
 [ 3.30000e+00  7.70000e+02  0.00000e+00  2.40000e+08           nan]
 [ 8.50000e+00  4.50000e+02  0.00000e+00  2.00000e+09           nan]
 [ 7.00000e+00  5.40000e+02  0.00000e+00  8.50000e+08           nan]
 [ 6.40000e+00  6.10000e+02  0.00000e+00  6.00000e+08           nan]
 [ 5.60000e+00  6.80000e+02  0.00000e+00  5.80000e+08           nan]
 [ 4.70000e+00  7.40000e+02  0.00000e+00  3.80000e+08           nan]
 [ 3.70000e+00  7.80000e+02  0.00000e+00  2.90000e+08           nan]
 [ 8.90000e+00  4.60000e+02  0.00000e+00  2.10000e+09           nan]
 [ 7.40000e+00  5.30000e+02  0.00000e+00  8.70000e+08           nan]
 [ 6.60000e+00  6.20000e+02  0.00000e+00  6.20000e+08           nan]
 [ 5.90000e+00  6.90000e+02  0.00000e+00  5.90000e+08           nan]
 [ 4.50000e+00  7.50000e+02  0.00000e+00  3.90000e+08           nan]
 [ 3.60000e+00  7.90000e+02  0.00000e+00  3.00000e+08           nan]
 [ 9.50000e+00  4.70000e+02  0.00000e+00  2.20000e+09           nan]
 [ 7.80000e+00  5.40000e+02  0.00000e+00  8.80000e+08           nan]
 [ 6.80000e+00  6.30000e+02  0.00000e+00  6.30000e+08           nan]]
```

Тип переменной и форма (shape)

In [12]:
```
print ( "Data type : ", type(data) )
print ( "Data shape : ", data.shape )
print ( data[-4:] )
```

```
Data type :  <class 'numpy.ndarray'>
Data shape :  (120, 5)
[[3.6e+00 7.9e+02 0.0e+00 3.0e+08     nan]
 [9.5e+00 4.7e+02 0.0e+00 2.2e+09     nan]
 [7.8e+00 5.4e+02 0.0e+00 8.8e+08     nan]
 [6.8e+00 6.3e+02 0.0e+00 6.3e+08     nan]]
```

Получение типа набора данных, строки, элемента

In [13]:
```
data1 = np.genfromtxt(data_path, delimiter=",", dtype=None)
print('Shape of the dataset:', data1.shape)
print('Dataset type:', type(data1))
print('A single row of the dataset is type of:', type(data1[0]))
print('Types of elements:', type(data1[0][1]), type(data1[0][4]))
print('Dataset:')
print(data1)
```

```
Shape of the dataset: (120,)
Dataset type: <class 'numpy.ndarray'>
```

```
A single row of the dataset is type of: <class 'numpy.void'>
Types of elements: <class 'numpy.int64'> <class 'numpy.bytes_'>
Dataset:
[(2.43970e+03, -100,  0, 3.30000e+02, b'Planet')
 (6.05180e+03,   15,  2, 4.86800e+03, b'Planet')
 (6.37100e+03,   15,  1, 5.97360e+03, b'Planet')
 (3.38950e+03,  -63,  2, 6.41710e+02, b'Planet')
 (6.99110e+04, -145, 82, 1.89819e+03, b'Planet')
 (5.79600e+02,  462,  0, 7.30000e-02, b'Planet')
 (7.14920e+04, -145, 79, 5.68340e+02, b'Planet')
 (6.02680e+04, -218, 62, 8.68130e+01, b'Planet')
 (2.55590e+04,   52, 27, 1.02413e+02, b'Planet')
 (5.43640e+04, -214, 14, 5.68340e+02, b'Planet')
 (2.37000e+03,  -53,  0, 3.30000e-01, b'Planet')
 (4.86140e+03,  167,  0, 6.42000e-01, b'Planet')
 (2.50000e+01,  100,  0, 2.00000e-06, b'Planet')
 (1.27560e+04,   15,  1, 5.97200e+24, b'Planet')
 (4.95280e+04, -224, 27, 4.86700e+24, b'Planet')
 (1.42984e+05, -234, 79, 5.97200e+24, b'Planet')
 (1.20536e+05, -245, 62, 5.97200e+24, b'Planet')
 (5.11180e+04, -211, 27, 5.97200e+24, b'Planet')
 (4.95280e+04, -213, 14, 5.97200e+24, b'Planet')
 (4.92440e+04, -234, 62, 5.97200e+24, b'Planet')
 (5.24140e+04, -221, 27, 5.97200e+24, b'Planet')
 (6.02680e+04, -245, 27, 5.97200e+24, b'Planet')
 (6.96340e+05, -243, 82, 5.97200e+24, b'Planet')
 (7.14920e+05, -230, 79, 5.97200e+24, b'Planet')
 (4.95280e+05, -215, 62, 5.97200e+24, b'Planet')
 (5.41080e+04, -215, 27, 5.97200e+24, b'Planet')
 (6.02680e+04, -236, 14, 5.97200e+24, b'Planet')
 (4.95280e+05, -222, 82, 5.97200e+24, b'Planet')
 (7.14920e+05, -230, 79, 5.97200e+24, b'Planet')
 (5.41080e+04, -215, 62, 5.97200e+24, b'Planet')
 (6.04320e+04, -215, 27, 5.97200e+24, b'Planet')
 (6.02680e+04, -236, 14, 5.97200e+24, b'Planet')
 (2.54820e+04, -256, 27, 5.97200e+24, b'Planet')
 (6.59232e+05, -278, 82, 5.97200e+24, b'Planet')
 (7.14920e+05, -281, 79, 5.97200e+24, b'Planet')
 (5.89520e+04, -292, 62, 5.97200e+24, b'Planet')
 (4.95280e+05, -276, 27, 5.97200e+24, b'Planet')
 (6.02680e+04, -298, 14, 5.97200e+24, b'Planet')
 (6.94100e+03, -220,  0, 1.07000e-01, b'Planet')
 (1.02000e+04, -201,  0, 8.15000e-01, b'Planet')
 (6.95700e+05, 5778,  8, 1.98900e+30, b'Star')
 (6.96340e+05, 5505,  9, 1.98900e+30, b'Star')
 (7.14920e+05, 5778,  7, 1.98900e+30, b'Star')
 (6.94100e+03, 5778,  0, 2.00000e+30, b'Star')
 (1.02000e+04, 5778,  0, 2.00000e+30, b'Star')
 (1.27560e+04, 5778,  0, 2.00000e+30, b'Star')
 (4.95280e+04, 5778,  7, 1.98900e+30, b'Star')
 (5.41080e+04, 5778,  8, 1.98900e+30, b'Star')
 (6.02680e+04, 5778,  9, 1.98900e+30, b'Star')
 (6.96340e+05, 5778,  7, 1.98900e+30, b'Star')
 (7.14920e+05, 5778,  8, 1.98900e+30, b'Star')
 (6.94100e+03, 5778,  0, 1.98900e+30, b'Star')
 (1.02000e+04, 5778,  0, 1.98900e+30, b'Star')
 (1.27560e+04, 5778,  0, 1.98900e+30, b'Star')
 (4.95280e+04, 5778,  7, 1.98900e+30, b'Star')
 (5.41080e+04, 5778,  8, 1.98900e+30, b'Star')
 (6.02680e+04, 5778,  9, 1.98900e+30, b'Star')
 (6.96340e+05, 5778,  7, 1.98900e+30, b'Star')
```

```
(7.14920e+05, 5778,  8, 1.98900e+30, b'Star')
(6.94100e+03, 5778,  0, 1.98900e+30, b'Star')
(1.02000e+04, 5778,  0, 1.98900e+30, b'Star')
(1.27560e+04, 5778,  0, 1.98900e+30, b'Star')
(4.95280e+04, 5778,  7, 1.98900e+30, b'Star')
(5.41080e+04, 5778,  8, 1.98900e+30, b'Star')
(6.02680e+04, 5778,  9, 1.98900e+30, b'Star')
(6.96340e+05, 5778,  7, 1.98900e+30, b'Star')
(7.14920e+05, 5778,  8, 1.98900e+30, b'Star')
(6.94100e+03, 5778,  0, 1.98900e+30, b'Star')
(1.02000e+04, 5778,  0, 1.98900e+30, b'Star')
(1.27560e+04, 5778,  0, 1.98900e+30, b'Star')
(4.95280e+04, 5778,  7, 1.98900e+30, b'Star')
(5.41080e+04, 5778,  8, 1.98900e+30, b'Star')
(6.02680e+04, 5778,  9, 1.98900e+30, b'Star')
(6.96340e+05, 5778,  7, 1.98900e+30, b'Star')
(7.14920e+05, 5778,  8, 1.98900e+30, b'Star')
(6.94100e+03, 5778,  0, 1.98900e+30, b'Star')
(1.02000e+04, 5778,  0, 1.98900e+30, b'Star')
(1.27560e+04, 5778,  0, 1.98900e+30, b'Star')
(4.95280e+04, 5778,  7, 1.98900e+30, b'Star')
(5.41080e+04, 5778,  8, 1.98900e+30, b'Star')
(5.00000e+00,  500,  0, 1.00000e+09, b'Meteor')
(4.50000e+00,  600,  0, 5.00000e+08, b'Meteor')
(3.00000e+00,  700,  0, 3.00000e+08, b'Meteor')
(2.20000e+00,  800,  0, 2.00000e+08, b'Meteor')
(6.70000e+00,  400,  0, 1.50000e+09, b'Meteor')
(4.00000e+00,  550,  0, 4.00000e+08, b'Meteor')
(3.50000e+00,  650,  0, 2.50000e+08, b'Meteor')
(2.80000e+00,  750,  0, 2.20000e+08, b'Meteor')
(6.50000e+00,  420,  0, 1.70000e+09, b'Meteor')
(5.20000e+00,  480,  0, 6.00000e+08, b'Meteor')
(4.80000e+00,  580,  0, 4.50000e+08, b'Meteor')
(3.90000e+00,  680,  0, 3.50000e+08, b'Meteor')
(2.90000e+00,  780,  0, 2.30000e+08, b'Meteor')
(7.20000e+00,  430,  0, 1.80000e+09, b'Meteor')
(6.00000e+00,  520,  0, 7.00000e+08, b'Meteor')
(5.50000e+00,  600,  0, 5.50000e+08, b'Meteor')
(4.60000e+00,  670,  0, 3.60000e+08, b'Meteor')
(3.80000e+00,  720,  0, 2.80000e+08, b'Meteor')
(2.50000e+00,  790,  0, 2.50000e+08, b'Meteor')
(7.80000e+00,  440,  0, 1.90000e+09, b'Meteor')
(6.30000e+00,  510,  0, 8.00000e+08, b'Meteor')
(5.80000e+00,  590,  0, 5.70000e+08, b'Meteor')
(4.90000e+00,  660,  0, 4.70000e+08, b'Meteor')
(4.20000e+00,  730,  0, 3.20000e+08, b'Meteor')
(3.30000e+00,  770,  0, 2.40000e+08, b'Meteor')
(8.50000e+00,  450,  0, 2.00000e+09, b'Meteor')
(7.00000e+00,  540,  0, 8.50000e+08, b'Meteor')
(6.40000e+00,  610,  0, 6.00000e+08, b'Meteor')
(5.60000e+00,  680,  0, 5.80000e+08, b'Meteor')
(4.70000e+00,  740,  0, 3.80000e+08, b'Meteor')
(3.70000e+00,  780,  0, 2.90000e+08, b'Meteor')
(8.90000e+00,  460,  0, 2.10000e+09, b'Meteor')
(7.40000e+00,  530,  0, 8.70000e+08, b'Meteor')
(6.60000e+00,  620,  0, 6.20000e+08, b'Meteor')
(5.90000e+00,  690,  0, 5.90000e+08, b'Meteor')
(4.50000e+00,  750,  0, 3.90000e+08, b'Meteor')
(3.60000e+00,  790,  0, 3.00000e+08, b'Meteor')
(9.50000e+00,  470,  0, 2.20000e+09, b'Meteor')
(7.80000e+00,  540,  0, 8.80000e+08, b'Meteor')
```

```
 (6.80000e+00,  630,  0, 6.30000e+08, b'Meteor')]
<ipython-input-13-be1180784c4e>:1: VisibleDeprecationWarning: Reading unicode strings wit
hout specifying the encoding argument is deprecated. Set the encoding, use None for the s
ystem default.
  data1 = np.genfromtxt(data_path, delimiter=",", dtype=None)
```

Указание типа столбцов при загрузке данных

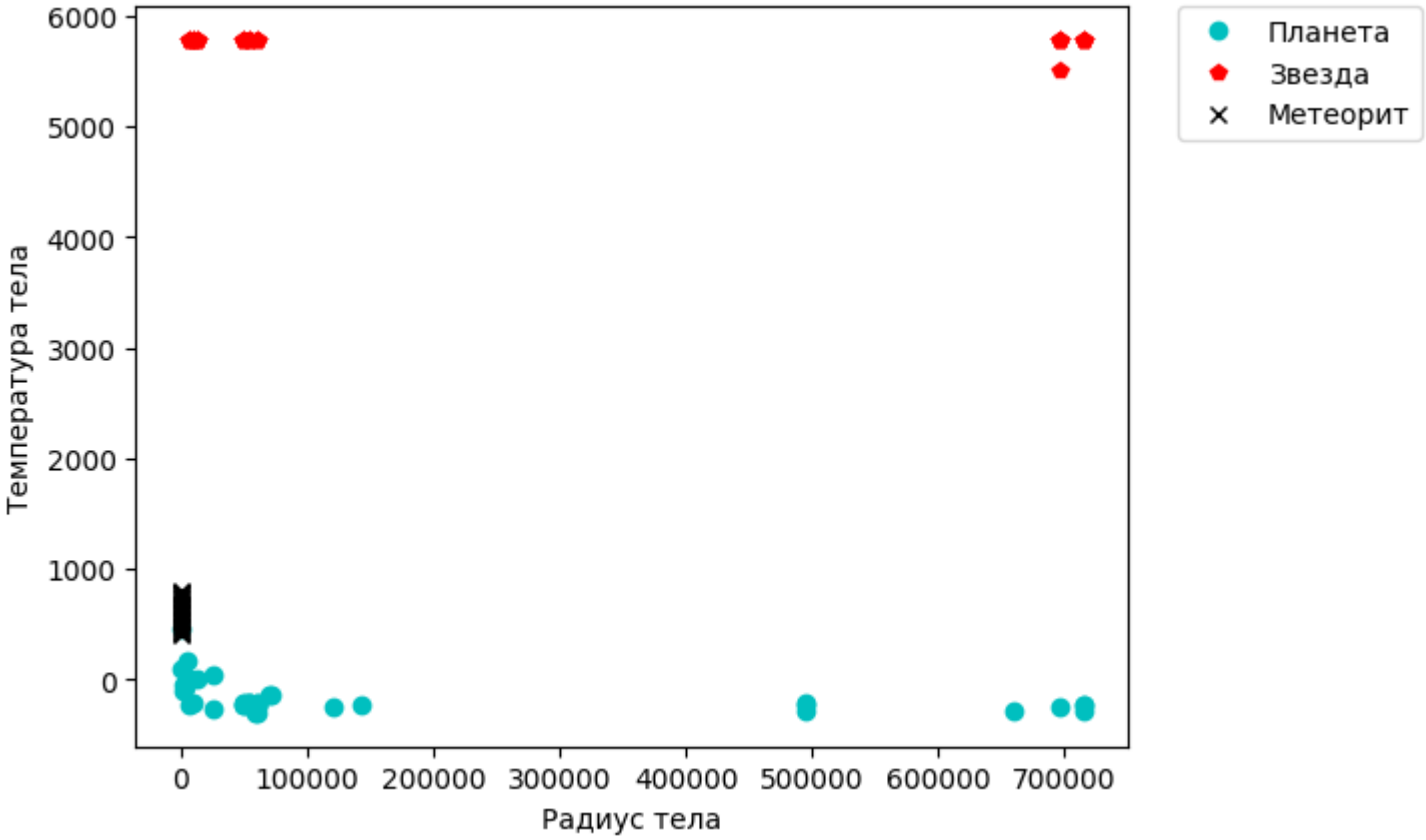
In [14]:
```
dt = np.dtype("f8, f8, f8, f8, U30")
data2 = np.genfromtxt(data_path, delimiter=",", dtype=dt)
print('Shape of the dataset:', data2.shape)
print('Dataset type:', type(data2))
print('A single row of the dataset is type of:', type(data2[0]))
print('Types of elements:', type(data2[0][1]), type(data2[0][4]))
print('Dataset slice:')
print(data2[:10])
```

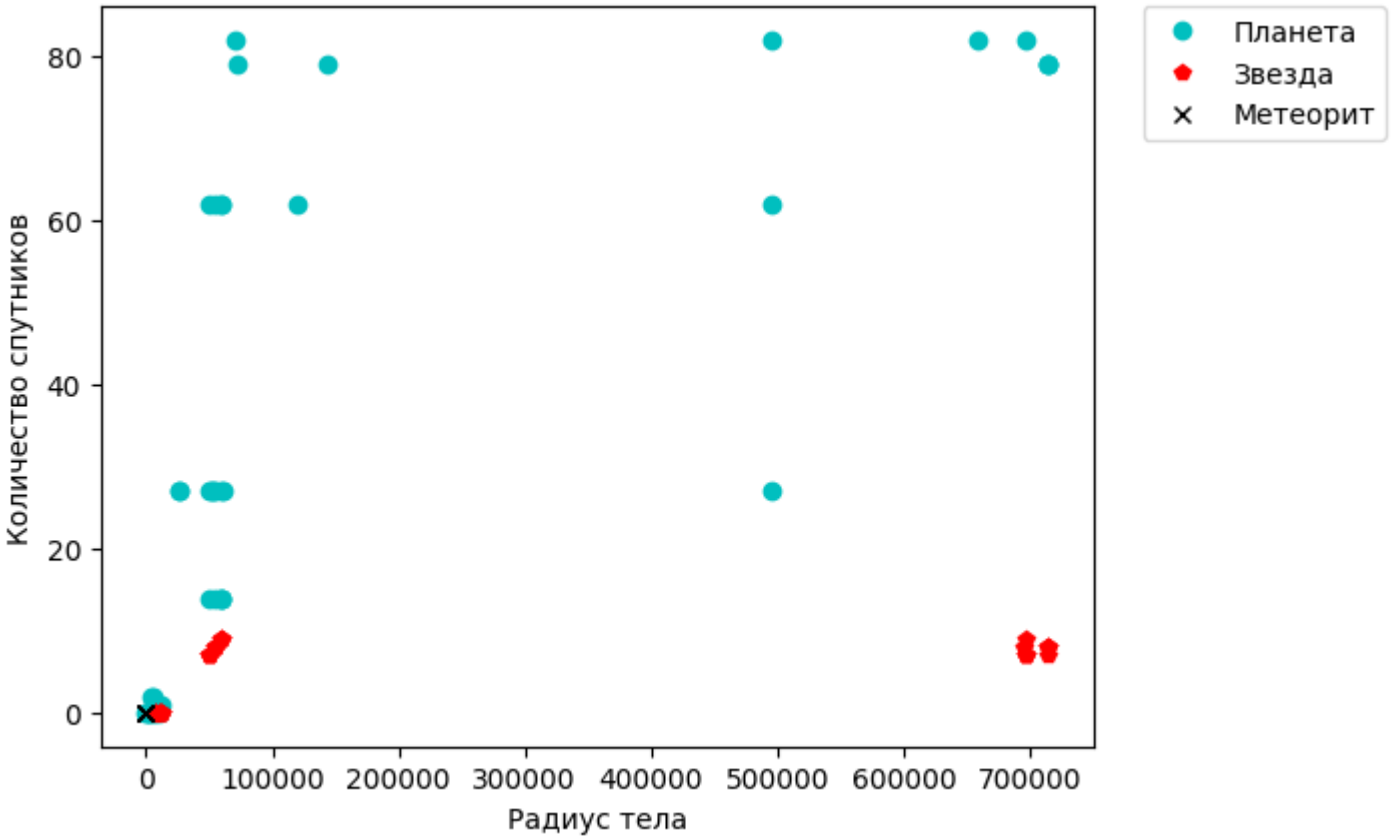
```
Shape of the dataset: (120,)
Dataset type: <class 'numpy.ndarray'>
A single row of the dataset is type of: <class 'numpy.void'>
Types of elements: <class 'numpy.float64'> <class 'numpy.str_'>
Dataset slice:
[( 2439.7, -100.,  0., 3.30000e+02, 'Planet')
 ( 6051.8,   15.,  2., 4.86800e+03, 'Planet')
 ( 6371. ,   15.,  1., 5.97360e+03, 'Planet')
 ( 3389.5,  -63.,  2., 6.41710e+02, 'Planet')
 (69911. , -145., 82., 1.89819e+03, 'Planet')
 (  579.6,  462.,  0., 7.30000e-02, 'Planet')
 (71492. , -145., 79., 5.68340e+02, 'Planet')
 (60268. , -218., 62., 8.68130e+01, 'Planet')
 (25559. ,   52., 27., 1.02413e+02, 'Planet')
 (54364. , -214., 14., 5.68340e+02, 'Planet')]
```

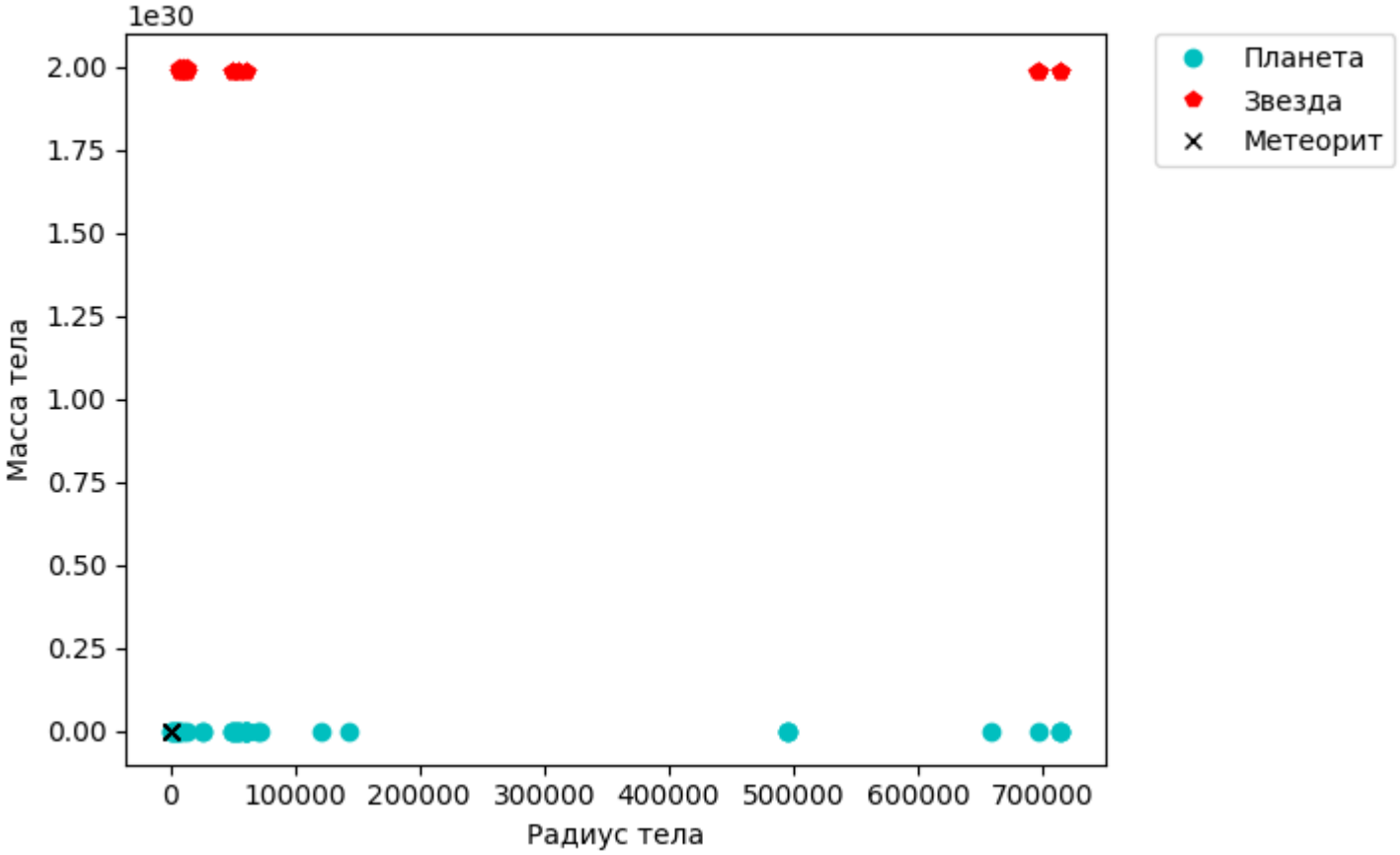
Построение графиков с использованием Matplotlib

In [23]:
```
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

# Данные из отдельных столбцов
radius = [] # Радиус тела
temperature = [] # Температура тела
satellites = [] # КОличество спутников
mass = [] # Масса тела

# Заметка о маркерах!

# Первы символ - цвет:
# 'b' синий.
# 'r' красный.
# 'g' зелёный.
# 'c' зеленовато-голубой.
# 'm' лиловато-малиновый.
# 'y' жёлтый.
# 'k' чёрный.
# 'w' белый.

# Второй символ - тип отображения:
# '-' сплошная линия.
# '--' штриховая линия.
# '-.' штрих-пунктирная линия.
```

```
    # ':' пунктирная линия.
    # '.' маркеры.
    # ',' пиксели.
    # 'o' круги.
    # 'v' треугольники, смотрящие вниз.
    # '^' треугольники, смотрящие вверх.
    # '<' треугольники, смотрящие влево.
    # '>' треугольники, смотрящие вправо.
    # 's' квадраты.
    # 'p' пятиугольники.
    # '+' плюс.
    # 'x' крест.
    # 'D' ромб.
    # 'd' тонкий ромб.
    # 'h' шестиугольник (1).
    # 'H' шестиугольник (2).
    # '8' восьмиугольник.

    # Выполняется обход всей коллекции data2
    for dot in data2:
        radius.append(dot[0])
        temperature.append(dot[1])
        satellites.append(dot[2])
        mass.append(dot[3])

    # Строим графики по проекциям данных
    # Учитывается, что каждые 40 типов тел идут последовательно
    plt.figure(1)
    planet, = plt.plot(radius[:40], temperature[:40], 'co', label='Планета')
    star, = plt.plot(radius[40:80], temperature[40:80], 'rp', label='Звезда')
    meteor, = plt.plot(radius[80:120], temperature[80:120], 'kx', label='Метеорит')
    plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
    plt.xlabel('Радиус тела')
    plt.ylabel('Температура тела')

    plt.figure(2)
    planet, = plt.plot(radius[:40], satellites[:40], 'co', label='Планета')
    star, = plt.plot(radius[40:80], satellites[40:80], 'rp', label='Звезда')
    meteor, = plt.plot(radius[80:120], satellites[80:120], 'kx', label='Метеорит')
    plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
    plt.xlabel('Радиус тела')
    plt.ylabel('Количество спутников')

    plt.figure(3)
    planet, = plt.plot(radius[:40], mass[:40], 'co', label='Планета')
    star, = plt.plot(radius[40:80], mass[40:80], 'rp', label='Звезда')
    meteor, = plt.plot(radius[80:120], mass[80:120], 'kx', label='Метеорит')
    plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
    plt.xlabel('Радиус тела')
    plt.ylabel('Масса тела')

    plt.show()
```


Планета
Звезда
Метеорит
Температура тела
Радиус тела
6000
5000
4000
3000
2000
1000
0
0
100000
200000
300000
400000
500000
600000
700000


Планета
Звезда
Метеорит
Количество спутников
Радиус тела
80
60
40
20
0
0
100000
200000
300000
400000
500000
600000
700000


1e30
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Масса тела
0
100000
200000
300000
400000
500000
600000
700000
Радиус тела
Планета
Звезда
Метеорит

In [ ]: