Лабораторная работа Nº1

Исследование набора данных "Диаметр и средние значения RGB для цитрусовых: апельсины, лимоны, грейпфруты".

```
!wget https://raw.githubusercontent.com/WaisShams/ai_ml_1/main/data
%20set/citrus.data
```

```
--2024-05-13 19:49:07--
https://raw.githubusercontent.com/WaisShams/ai_ml_1/main/data%20set/
citrus.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: 3353 (3.3K) [text/plain]
Saving to: ‘citrus.data’

 citrus.data           0%[                   ]       0  --.-KB/s
citrus.data          100%[===================>]   3.27K  --.-KB/s    in
0s

2024-05-13 19:49:07 (50.3 MB/s) - ‘citrus.data’ saved [3353/3353]
```

Подключение библиотеки NumPy и загрузка данных

```
import numpy as np
data_path = "citrus.data"
data = np.genfromtxt(data_path, delimiter=",")
print(data)
```

```
[[  2.96 172.    85.     2.      nan]
 [  3.91 166.    78.     3.      nan]
 [  4.42 156.    81.     2.      nan]
 [  4.47 163.    81.     4.      nan]
 [  4.48 161.    72.     9.      nan]
 [  4.59 142.   100.     2.      nan]
 [  4.64 156.    85.     2.      nan]
 [  4.65 142.    74.     2.      nan]
 [  4.68 159.    90.    16.      nan]
 [  4.69 161.    76.     6.      nan]
 [  4.73 148.    88.     2.      nan]
 [  4.74 166.    69.     2.      nan]
 [  4.75 166.    98.    13.      nan]
 [  4.77 166.    86.     6.      nan]
 [  4.78 170.    82.     2.      nan]
 [  4.82 162.    87.    12.      nan]
 [  4.85 166.    78.     5.      nan]
 [  4.91 151.    79.     2.      nan]
 [  4.97 168.    79.    22.      nan]
 [  5.02 179.    73.     2.      nan]
 [  5.02 162.    79.     2.      nan]
 [  5.08 156.    94.     2.      nan]
 [  5.1  169.    86.     2.      nan]
 [  5.11 146.    84.     2.      nan]
 [  5.15 154.    87.     6.      nan]
 [  5.16 168.    82.     2.      nan]
 [  5.22 162.    79.     2.      nan]
 [  5.22 161.    75.    10.      nan]
 [  5.24 153.    90.     2.      nan]
 [  5.24 157.    85.    15.      nan]
 [  5.25 155.    94.     6.      nan]
 [  5.26 155.    93.     2.      nan]
 [  5.3  171.    82.     2.      nan]
 [  5.3  159.    84.    13.      nan]
 [  5.3  162.    93.     2.      nan]
 [  5.32 145.    82.     2.      nan]
 [  5.34 173.    75.     3.      nan]
 [  5.35 158.    97.     2.      nan]
 [  5.36 164.    86.     4.      nan]
 [  5.37 178.    92.     7.      nan]
 [  5.37 177.    76.     2.      nan]
 [  5.37 171.    75.    23.      nan]
 [  5.38 163.    93.     8.      nan]
 [  5.4  148.    81.     2.      nan]
 [  5.43 157.    98.     2.      nan]
 [  5.45 145.    75.    18.      nan]
 [  5.47 162.    90.     2.      nan]
 [  5.47 170.    82.    11.      nan]
 [  5.48 165.    67.     8.      nan]
 [  5.48 146.    72.     2.      nan]
 [  5.48 167.    88.     4.      nan]
 [  5.48 128.    87.     4.      nan]
 [  5.49 177.    80.     2.      nan]
 [  5.51 150.    98.     5.      nan]
 [  5.51 165.    88.     7.      nan]
 [  5.51 157.    82.     3.      nan]
 [  5.52 140.    65.     5.      nan]
 [  5.53 169.    76.     3.      nan]
 [  5.53 150.    87.     2.      nan]
 [  5.57 145.    81.     9.      nan]
 [  5.57 159.    83.     2.      nan]
 [  5.58 160.    73.     3.      nan]
 [  5.59 168.    75.    19.      nan]
 [  5.6  167.    74.     2.      nan]
 [  5.6  171.    72.     2.      nan]
 [  5.61 165.    84.     2.      nan]
 [  5.61 158.    69.     8.      nan]
 [  5.61 167.   104.     2.      nan]
 [  5.63 161.    74.     2.      nan]
 [  5.65 138.    78.    14.      nan]
 [  5.65 155.    70.     2.      nan]
 [  5.65 158.   107.     3.      nan]
 [  5.65 154.    98.    10.      nan]
 [  5.66 163.    86.     3.      nan]
 [  5.66 164.    94.     2.      nan]
 [  5.67 165.    78.     2.      nan]
 [  5.67 150.    85.     6.      nan]
 [  5.68 137.    76.     9.      nan]
 [  5.69 164.    81.     8.      nan]
 [  5.71 168.    78.     2.      nan]
 [  5.71 151.    82.     2.      nan]
 [  5.71 154.    62.    20.      nan]
 [  5.71 156.    91.     2.      nan]
 [  5.72 164.    95.     2.      nan]
 [  5.72 141.    85.     2.      nan]
 [  5.73 169.    66.     2.      nan]
 [  5.73 163.    89.    18.      nan]
 [  5.73 155.    87.     2.      nan]
 [  5.73 151.    86.     2.      nan]
 [  5.73 156.    68.     4.      nan]
 [  5.74 149.    68.     2.      nan]
 [  5.76 168.    94.    30.      nan]
 [  5.76 146.    78.     8.      nan]
 [  5.77 150.    95.     2.      nan]
 [  5.77 142.   101.     9.      nan]
 [  5.77 163.    76.     2.      nan]
 [  5.79 150.    90.     2.      nan]
 [  5.79 144.    73.     2.      nan]
 [  5.8  139.   105.     2.      nan]
 [  5.8  165.    68.     2.      nan]
 [  7.63 150.    74.    24.      nan]
 [  7.69 150.    43.    15.      nan]
 [  7.72 150.    54.    21.      nan]
 [  7.77 150.    75.    15.      nan]
 [  7.84 149.    77.    20.      nan]
 [  7.94 144.    81.    26.      nan]
 [  8.05 134.    85.    26.      nan]
 [  8.11 148.    59.    22.      nan]
 [  8.18 142.    81.     7.      nan]
 [  8.19 143.    65.    30.      nan]
 [  8.19 149.    63.    16.      nan]
 [  8.21 157.    83.    21.      nan]
 [  8.24 142.    84.    10.      nan]
 [  8.24 149.    83.    19.      nan]
 [  8.29 168.    86.    42.      nan]
 [  8.3  158.    71.    16.      nan]
 [  8.36 133.    74.    21.      nan]
 [  8.36 154.    80.    12.      nan]
 [  8.37 137.    71.    18.      nan]
 [  8.37 156.    67.    22.      nan]
 [  8.45 141.    54.     2.      nan]
 [  8.47 148.    88.    23.      nan]
 [  8.5  157.    80.    33.      nan]
 [  8.5  147.    68.    12.      nan]
 [  8.51 148.    91.    28.      nan]
 [  8.51 162.    75.     5.      nan]
 [  8.53 171.    72.     6.      nan]
 [  8.54 153.    87.    11.      nan]
 [  8.59 146.    82.    40.      nan]
 [  8.59 142.    58.    11.      nan]
 [  8.61 137.    70.    18.      nan]
 [  8.64 145.    67.    31.      nan]
 [  8.65 158.    70.    36.      nan]
 [  8.66 153.    74.    17.      nan]
 [  8.66 143.    62.    19.      nan]
 [  8.67 162.    62.    24.      nan]
 [  8.67 152.    63.     6.      nan]
 [  8.68 153.    78.    32.      nan]
 [  8.68 168.    70.    13.      nan]
 [  8.69 154.    83.    36.      nan]
 [  8.69 147.    58.    10.      nan]
 [  8.71 131.    80.    24.      nan]
 [  8.73 151.    81.     3.      nan]
 [  8.74 148.    74.    22.      nan]
 [  8.74 152.    85.    12.      nan]
 [  8.75 154.    53.    41.      nan]
 [  8.76 146.    78.    13.      nan]
 [  8.76 143.    69.    13.      nan]
 [  8.78 153.    71.    10.      nan]
 [  8.79 150.    72.    24.      nan]]
```

Тип переменной и форма (shape)

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

Data type :  <class 'numpy.ndarray'>
Data shape :  (150, 5)
[[  8.76 146.    78.    13.      nan]
 [  8.76 143.    69.    13.      nan]
 [  8.78 153.    71.    10.      nan]
 [  8.79 150.    72.    24.      nan]]
```

Получение типа набора данных, строки, элемента

```
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: (150,)
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:
[(2.96, 172,  85,  2, b'orange') (3.91, 166,  78,  3, b'orange')
 (4.42, 156,  81,  2, b'orange') (4.47, 163,  81,  4, b'orange')
 (4.48, 161,  72,  9, b'orange') (4.59, 142, 100,  2, b'orange')
 (4.64, 156,  85,  2, b'orange') (4.65, 142,  74,  2, b'orange')
 (4.68, 159,  90, 16, b'orange') (4.69, 161,  76,  6, b'orange')
 (4.73, 148,  88,  2, b'orange') (4.74, 166,  69,  2, b'orange')
 (4.75, 166,  98, 13, b'orange') (4.77, 166,  86,  6, b'orange')
 (4.78, 170,  82,  2, b'orange') (4.82, 162,  87, 12, b'orange')
 (4.85, 166,  78,  5, b'orange') (4.91, 151,  79,  2, b'orange')
 (4.97, 168,  79, 22, b'orange') (5.02, 179,  73,  2, b'orange')
 (5.02, 162,  79,  2, b'orange') (5.08, 156,  94,  2, b'orange')
 (5.1 , 169,  86,  2, b'orange') (5.11, 146,  84,  2, b'orange')
 (5.15, 154,  87,  6, b'orange') (5.16, 168,  82,  2, b'orange')
 (5.22, 162,  79,  2, b'orange') (5.22, 161,  75, 10, b'orange')
 (5.24, 153,  90,  2, b'orange') (5.24, 157,  85, 15, b'orange')
 (5.25, 155,  94,  6, b'orange') (5.26, 155,  93,  2, b'orange')
 (5.3 , 171,  82,  2, b'orange') (5.3 , 159,  84, 13, b'orange')
 (5.3 , 162,  93,  2, b'orange') (5.32, 145,  82,  2, b'orange')
 (5.34, 173,  75,  3, b'orange') (5.35, 158,  97,  2, b'orange')
 (5.36, 164,  86,  4, b'orange') (5.37, 178,  92,  7, b'orange')
 (5.37, 177,  76,  2, b'orange') (5.37, 171,  75, 23, b'orange')
 (5.38, 163,  93,  8, b'orange') (5.4 , 148,  81,  2, b'orange')
 (5.43, 157,  98,  2, b'orange') (5.45, 145,  75, 18, b'orange')
 (5.47, 162,  90,  2, b'orange') (5.47, 170,  82, 11, b'orange')
 (5.48, 165,  67,  8, b'orange') (5.48, 146,  72,  2, b'orange')
 (5.48, 167,  88,  4, b'lemon') (5.48, 128,  87,  4, b'lemon')
 (5.49, 177,  80,  2, b'lemon') (5.51, 150,  98,  5, b'lemon')
 (5.51, 165,  88,  7, b'lemon') (5.51, 157,  82,  3, b'lemon')
 (5.52, 140,  65,  5, b'lemon') (5.53, 169,  76,  3, b'lemon')
 (5.53, 150,  87,  2, b'lemon') (5.57, 145,  81,  9, b'lemon')
 (5.57, 159,  83,  2, b'lemon') (5.58, 160,  73,  3, b'lemon')
 (5.59, 168,  75, 19, b'lemon') (5.6 , 167,  74,  2, b'lemon')
 (5.6 , 171,  72,  2, b'lemon') (5.61, 165,  84,  2, b'lemon')
 (5.61, 158,  69,  8, b'lemon') (5.61, 167, 104,  2, b'lemon')
 (5.63, 161,  74,  2, b'lemon') (5.65, 138,  78, 14, b'lemon')
 (5.65, 155,  70,  2, b'lemon') (5.65, 158, 107,  3, b'lemon')
 (5.65, 154,  98, 10, b'lemon') (5.66, 163,  86,  3, b'lemon')
 (5.66, 164,  94,  2, b'lemon') (5.67, 165,  78,  2, b'lemon')
 (5.67, 150,  85,  6, b'lemon') (5.68, 137,  76,  9, b'lemon')
 (5.69, 164,  81,  8, b'lemon') (5.71, 168,  78,  2, b'lemon')
 (5.71, 151,  82,  2, b'lemon') (5.71, 154,  62, 20, b'lemon')
 (5.71, 156,  91,  2, b'lemon') (5.72, 164,  95,  2, b'lemon')
 (5.72, 141,  85,  2, b'lemon') (5.73, 169,  66,  2, b'lemon')
 (5.73, 163,  89, 18, b'lemon') (5.73, 155,  87,  2, b'lemon')
 (5.73, 151,  86,  2, b'lemon') (5.73, 156,  68,  4, b'lemon')
 (5.74, 149,  68,  2, b'lemon') (5.76, 168,  94, 30, b'lemon')
 (5.76, 146,  78,  8, b'lemon') (5.77, 150,  95,  2, b'lemon')
 (5.77, 142, 101,  9, b'lemon') (5.77, 163,  76,  2, b'lemon')
 (5.79, 150,  90,  2, b'lemon') (5.79, 144,  73,  2, b'lemon')
 (5.8 , 139, 105,  2, b'lemon') (5.8 , 165,  68,  2, b'lemon')
 (7.63, 150,  74, 24, b'grapefruit') (7.69, 150,  43, 15,
b'grapefruit')
 (7.72, 150,  54, 21, b'grapefruit') (7.77, 150,  75, 15,
b'grapefruit')
 (7.84, 149,  77, 20, b'grapefruit') (7.94, 144,  81, 26,
b'grapefruit')
 (8.05, 134,  85, 26, b'grapefruit') (8.11, 148,  59, 22,
b'grapefruit')
 (8.18, 142,  81,  7, b'grapefruit') (8.19, 143,  65, 30,
b'grapefruit')
 (8.19, 149,  63, 16, b'grapefruit') (8.21, 157,  83, 21,
b'grapefruit')
 (8.24, 142,  84, 10, b'grapefruit') (8.24, 149,  83, 19,
b'grapefruit')
 (8.29, 168,  86, 42, b'grapefruit') (8.3 , 158,  71, 16,
b'grapefruit')
 (8.36, 133,  74, 21, b'grapefruit') (8.36, 154,  80, 12,
b'grapefruit')
 (8.37, 137,  71, 18, b'grapefruit') (8.37, 156,  67, 22,
b'grapefruit')
 (8.45, 141,  54,  2, b'grapefruit') (8.47, 148,  88, 23,
b'grapefruit')
 (8.5 , 157,  80, 33, b'grapefruit') (8.5 , 147,  68, 12,
b'grapefruit')
 (8.51, 148,  91, 28, b'grapefruit') (8.51, 162,  75,  5,
b'grapefruit')
 (8.53, 171,  72,  6, b'grapefruit') (8.54, 153,  87, 11,
b'grapefruit')
 (8.59, 146,  82, 40, b'grapefruit') (8.59, 142,  58, 11,
b'grapefruit')
 (8.61, 137,  70, 18, b'grapefruit') (8.64, 145,  67, 31,
b'grapefruit')
 (8.65, 158,  70, 36, b'grapefruit') (8.66, 153,  74, 17,
b'grapefruit')
 (8.66, 143,  62, 19, b'grapefruit') (8.67, 162,  62, 24,
b'grapefruit')
 (8.67, 152,  63,  6, b'grapefruit') (8.68, 153,  78, 32,
b'grapefruit')
 (8.68, 168,  70, 13, b'grapefruit') (8.69, 154,  83, 36,
b'grapefruit')
 (8.69, 147,  58, 10, b'grapefruit') (8.71, 131,  80, 24,
b'grapefruit')
 (8.73, 151,  81,  3, b'grapefruit') (8.74, 148,  74, 22,
b'grapefruit')
 (8.74, 152,  85, 12, b'grapefruit') (8.75, 154,  53, 41,
b'grapefruit')
 (8.76, 146,  78, 13, b'grapefruit') (8.76, 143,  69, 13,
b'grapefruit')
 (8.78, 153,  71, 10, b'grapefruit') (8.79, 150,  72, 24,
b'grapefruit')]

<ipython-input-4-be1180784c4e>:1: VisibleDeprecationWarning: Reading
unicode strings without specifying the encoding argument is
deprecated. Set the encoding, use None for the system default.
  data1 = np.genfromtxt(data_path, delimiter=",", dtype=None)
```

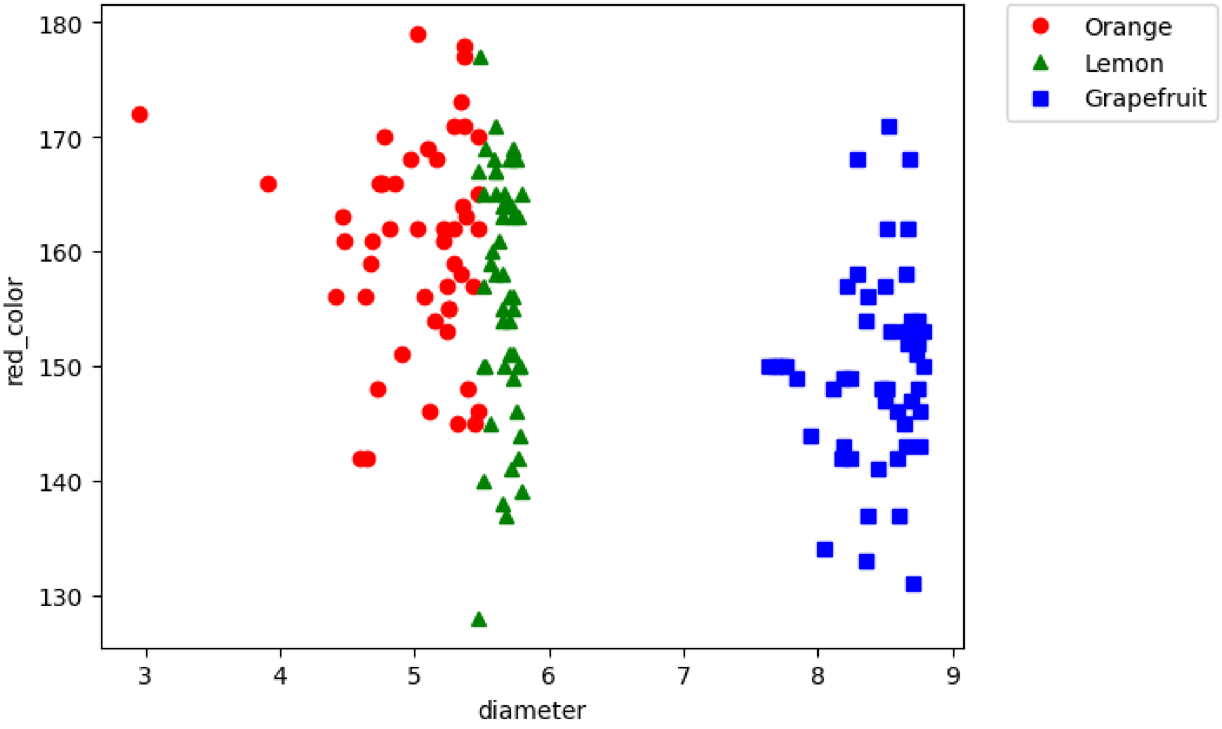
Указание типа столбцов при загрузке данных

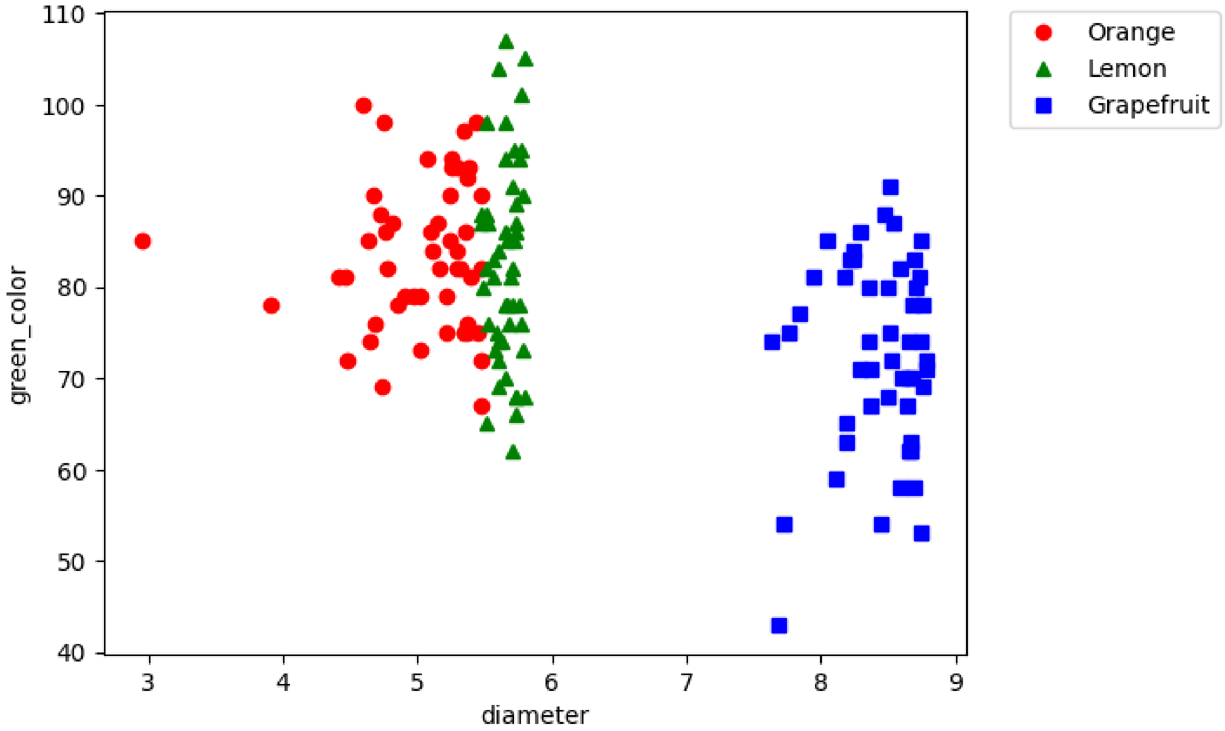
```
dt = np.dtype("f8, f8, f8, f8, U30")
data2 = np.genfromtxt(data_path, delimiter=",", dtype=dt)
print('Shape of the dataset:', data2.shape)
print('Dataset type:', type(data2))
print('A single row of the dataset is type of:', type(data2[0]))
print('Types of elements:', type(data2[0][1]), type(data2[0][4]))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

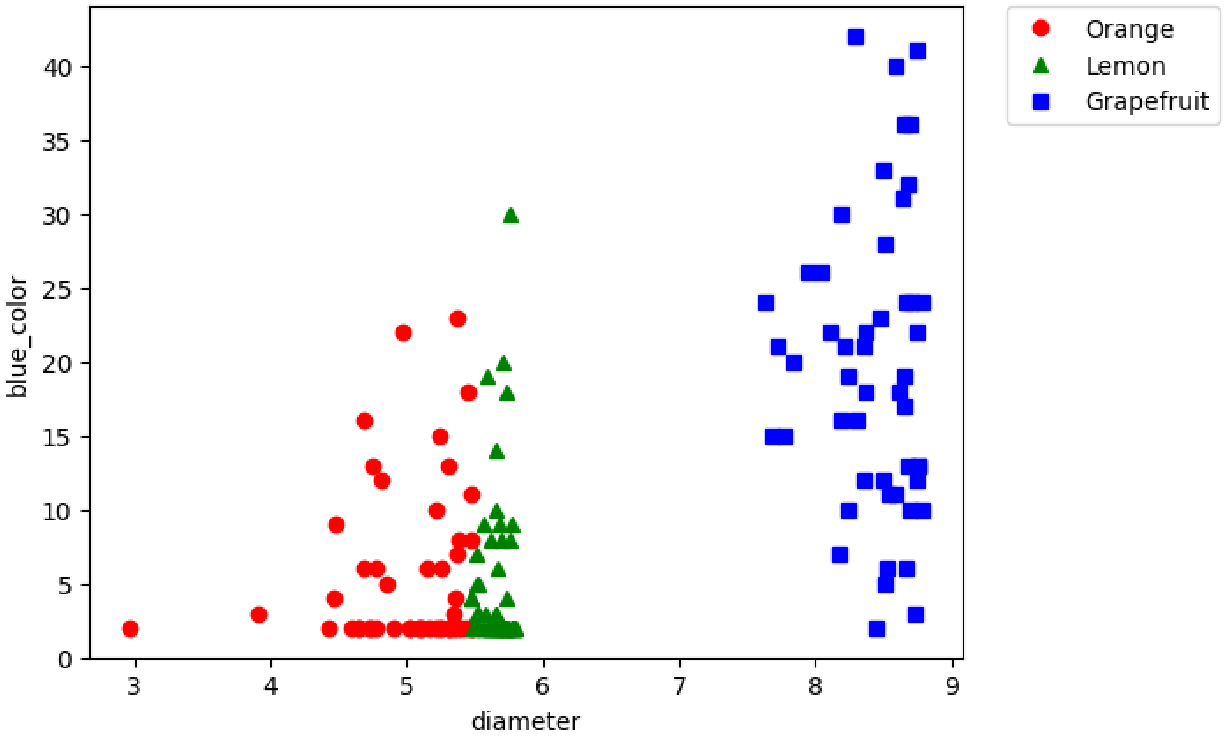
```
Shape of the dataset: (150,)
Dataset type: <class 'numpy.ndarray'>
A single row of the dataset is type of: <class 'numpy.void'>
Types of elements: <class 'numpy.float64'> <class 'numpy.str_'>
Dataset slice:
[(2.96, 172.,  85.,  2., 'orange') (3.91, 166.,  78.,  3., 'orange')
 (4.42, 156.,  81.,  2., 'orange') (4.47, 163.,  81.,  4., 'orange')
 (4.48, 161.,  72.,  9., 'orange') (4.59, 142., 100.,  2., 'orange')
 (4.64, 156.,  85.,  2., 'orange') (4.65, 142.,  74.,  2., 'orange')
 (4.68, 159.,  90., 16., 'orange') (4.69, 161.,  76.,  6., 'orange')]
```

Построение графиков с использованием Matplotlib

```
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

# Данные из отдельных столбцов
diameter = [] # Diameter
red_color = [] # red_color
green_color = [] # green_color
blue_color = [] # blue_color

# Выполняем обход всей коллекции data2
for dot in data2:
    diameter.append(dot[0])
    red_color.append(dot[1])
    green_color.append(dot[2])
    blue_color.append(dot[3])

# Строим графики по проекциям данных
# Учитываем, что каждые 50 типов цитрусовых идут последовательно
plt.figure(1)
orange, = plt.plot(diameter[:50], red_color[:50], 'ro',
label='Orange')
lemon, = plt.plot(diameter[50:100], red_color[50:100], 'g^',
label='Lemon')
grapefruit, = plt.plot(diameter[100:150], red_color[100:150], 'bs',
label='Grapefruit')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('diameter')
plt.ylabel('red_color')
plt.figure(2)
orange, = plt.plot(diameter[:50], green_color[:50], 'ro',
label='Orange')
lemon, = plt.plot(diameter[50:100], green_color[50:100], 'g^',
label='Lemon')
grapefruit, = plt.plot(diameter[100:150], green_color[100:150], 'bs',
label='Grapefruit')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('diameter')
plt.ylabel('green_color')

plt.figure(3)
orange, = plt.plot(diameter[:50], blue_color[:50], 'ro',
label='Orange')
lemon, = plt.plot(diameter[50:100], blue_color[50:100], 'g^',
label='Lemon')
grapefruit, = plt.plot(diameter[100:150], blue_color[100:150], 'bs',
label='Grapefruit')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('diameter')
plt.ylabel('blue_color')

plt.show()
```




110
100
90
80
70
60
50
40
green_color
3
4
5
6
7
8
9
diameter
Orange
Lemon
Grapefruit


40
35
30
25
20
15
10
5
0
blue_color
3
4
5
6
7
8
9
diameter
Orange
Lemon
Grapefruit