

读取iris数据集中鸢尾花的萼片、花瓣长度数据，并对其进行排序、去重，并求出和、累积和、均值、标准差、方差、最小值、最大值。

1、导入模块

```
In [ ]: import pandas as pd
import numpy as np
```

2、获取数据

```
In [ ]: data=pd.read_csv('iris.csv')
data.head()
```

```
Out [ ]:   Unnamed: 0  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width  Species
0           1           5.1           3.5           1.4           0.2    setosa
1           2           4.9           3.0           1.4           0.2    setosa
2           3           4.7           3.2           1.3           0.2    setosa
3           4           4.6           3.1           1.5           0.2    setosa
4           5           5.0           3.6           1.4           0.2    setosa
```

3、数据清洗

```
In [ ]: data=data.iloc[:,1:]
data.head()
```

```
Out [ ]:   Sepal.Length  Sepal.Width  Petal.Length  Petal.Width  Species
0           5.1           3.5           1.4           0.2    setosa
1           4.9           3.0           1.4           0.2    setosa
2           4.7           3.2           1.3           0.2    setosa
3           4.6           3.1           1.5           0.2    setosa
4           5.0           3.6           1.4           0.2    setosa
```

4、数据统计

(1)、创建数据类型（字符串类型）

```
In [ ]: stringData=np.array(data,dtype=str)
stringData[:,5,:]
```

```
Out [ ]: array([[ '5.1', '3.5', '1.4', '0.2', 'setosa'],
                [ '4.9', '3.0', '1.4', '0.2', 'setosa'],
                [ '4.7', '3.2', '1.3', '0.2', 'setosa'],
                [ '4.6', '3.1', '1.5', '0.2', 'setosa'],
                [ '5.0', '3.6', '1.4', '0.2', 'setosa']], dtype='<U10')
```

(2)、创建二维数组

```
In [ ]: two_dimensional_array=np.array([[1,2],[3,4]])
two_dimensional_array,two_dimensional_array.shape
```

```
Out [ ]: (array([[1, 2],
                [3, 4]]),
          (2, 2))
```

(3)、将待处理数据的类型转化为float类型

```
In [ ]: data.iloc[:, :-1] = pd.DataFrame(data.iloc[:, :-1], dtype=float)
data.head()
```

```
Out [ ]:   Sepal.Length  Sepal.Width  Petal.Length  Petal.Width  Species
0          5.1           3.5           1.4           0.2    setosa
1          4.9           3.0           1.4           0.2    setosa
2          4.7           3.2           1.3           0.2    setosa
3          4.6           3.1           1.5           0.2    setosa
4          5.0           3.6           1.4           0.2    setosa
```

(4)、排序(按照Sepal.Length这一列排序)

```
In [ ]: data.sort_values('Sepal.Length').head()
```

```
Out [ ]:   Sepal.Length  Sepal.Width  Petal.Length  Petal.Width  Species
13          4.3           3.0           1.1           0.1    setosa
42          4.4           3.2           1.3           0.2    setosa
38          4.4           3.0           1.3           0.2    setosa
8           4.4           2.9           1.4           0.2    setosa
41          4.5           2.3           1.3           0.3    setosa
```

(5)、数据去重(对Petal.Length列去重，只保留第一次出现的行)

```
In [ ]: data.drop_duplicates(subset=['Petal.Length'], keep='first').head()
```

```
Out [ ]:   Sepal.Length  Sepal.Width  Petal.Length  Petal.Width  Species
0          5.1           3.5           1.4           0.2    setosa
2          4.7           3.2           1.3           0.2    setosa
3          4.6           3.1           1.5           0.2    setosa
5          5.4           3.9           1.7           0.4    setosa
11         4.8           3.4           1.6           0.2    setosa
```

(6)、对指定列（萼片的长度、宽度，或花瓣的长度、宽度，选一个，不用都做）求和、均值、标准差、方差、最小值及最大值

对Sepal.Width进行求解

```
In [ ]: #求和
print("求和", data.iloc[:, 1].sum())

#求均值
print("求均值", data.iloc[:, 1].mean())

#标准差
```

```
print("标准差",data.iloc[:,1].std())

print("求方差",data.iloc[:,1].var())

print("求最小值",data.iloc[:,1].min())

print("求最大值",data.iloc[:,1].max())
```

求和 458.6

求均值 3.0573333333333337

标准差 0.4358662849366982

求方差 0.189979418344519

求最小值 2.0

求最大值 4.4