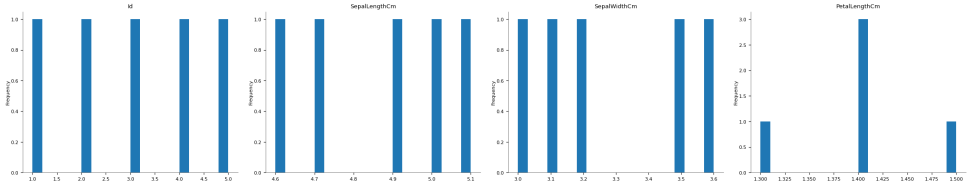
```
import numpy as np
import pandas as pd
#Load Iris.csv into a Pandas data frame
iris = pd.read_csv('/content/Iris.csv')
```

```
iris = pd.read_csv('/content/Iris.csv', header = None)
col_names = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width','Species']
```

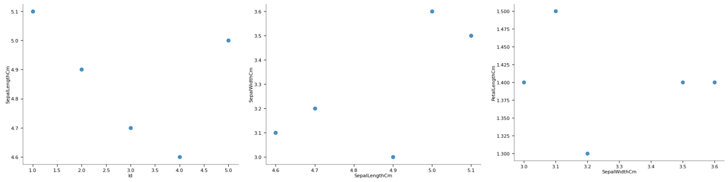
```
iris.head(n=5)
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

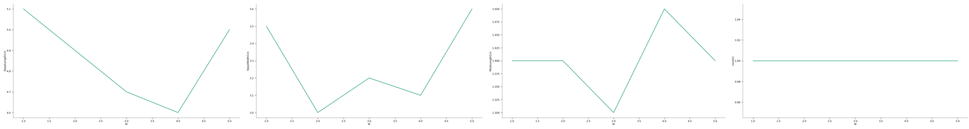
Distributions



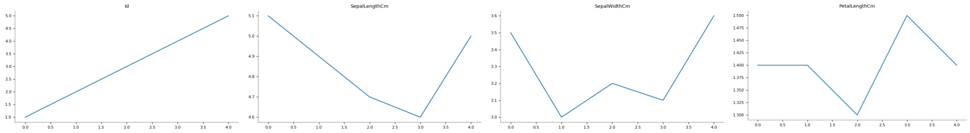
2-d distributions



Time series



Values



```
iris.head(n=11) #Return the first n rows.
iris.tail(n=2)  #Return the last n rows.
```

	0	1	2	3	4	5
149	149	6.2	3.4	5.4	2.3	Iris-virginica
150	150	5.9	3.0	5.1	1.8	Iris-virginica

```
#The index (row labels) of the Dataset.
iris.index
```

```
RangeIndex(start=0, stop=151, step=1)
```

```
#The column labels of the Dataset.
iris.columns
```

```
Int64Index([0, 1, 2, 3, 4, 5], dtype='int64')
```

```
#Return a tuple representing the dimensionality of the Dataset
iris.shape
```

```
(151, 6)
```

```
#Return the dtypes in the Dataset.
iris.dtypes
```

```
0    object
1    object
2    object
3    object
4    object
5    object
dtype: object
```

```
#Return the columns values in the Dataset in array format
iris.columns.values
```

```
array([0, 1, 2, 3, 4, 5])
```

```
#Generate descriptive statistics.
iris.describe(include='all')
```

	0	1	2	3	4	5
count	151	151	151	151	151	151
unique	151	36	24	44	23	4
top	Id	5.0	3.0	1.5	0.2	Iris-setosa
freq	1	10	26	14	28	50

```
iris = pd.read_csv('/content/Iris.csv', names = col_names)
iris[col_names] #Read the Data Column wise.
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
...	...	...	...	...	...
146	6.7	3.0	5.2	2.3	Iris-virginica
147	6.3	2.5	5.0	1.9	Iris-virginica
148	6.5	3.0	5.2	2.0	Iris-virginica
149	6.2	3.4	5.4	2.3	Iris-virginica
150	5.9	3.0	5.1	1.8	Iris-virginica

151 rows × 5 columns

```
#Sort object by labels (along an axis).
iris.sort_index(axis=1,ascending=False)
```

	Species	Sepal_Width	Sepal_Length	Petal_Width	Petal_Length
Id	Species	SepalWidthCm	SepalLengthCm	PetalWidthCm	PetalLengthCm
1	Iris-setosa	3.5	5.1	0.2	1.4
2	Iris-setosa	3.0	4.9	0.2	1.4
3	Iris-setosa	3.2	4.7	0.2	1.3
4	Iris-setosa	3.1	4.6	0.2	1.5
...	...	...	...	...	...
146	Iris-virginica	3.0	6.7	2.3	5.2
147	Iris-virginica	2.5	6.3	1.9	5.0
148	Iris-virginica	3.0	6.5	2.0	5.2
149	Iris-virginica	3.4	6.2	2.3	5.4
150	Iris-virginica	3.0	5.9	1.8	5.1

151 rows × 5 columns

```
iris.sort_index(axis=1,ascending=True)
```

	Petal_Length	Petal_Width	Sepal_Length	Sepal_Width	Species
Id	PetalLengthCm	PetalWidthCm	SepalLengthCm	SepalWidthCm	Species
1	1.4	0.2	5.1	3.5	Iris-setosa
2	1.4	0.2	4.9	3.0	Iris-setosa
3	1.3	0.2	4.7	3.2	Iris-setosa
4	1.5	0.2	4.6	3.1	Iris-setosa
...	...	...	...	...	...
146	5.2	2.3	6.7	3.0	Iris-virginica
147	5.0	1.9	6.3	2.5	Iris-virginica
148	5.2	2.0	6.5	3.0	Iris-virginica
149	5.4	2.3	6.2	3.4	Iris-virginica
150	5.1	1.8	5.9	3.0	Iris-virginica

151 rows × 5 columns

```
import numpy as np
import pandas as pd
iris = pd.read_csv('/content/Iris.csv')
col_names = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width','Species']
iris = pd.read_csv('/content/Iris.csv', names = col_names)
iris.sort_index(axis=1,ascending=True)
```

```
      Petal_Length  Petal_Width  Sepal_Length  Sepal_Width          Species
 Id  PetalLengthCm PetalWidthCm SepalLengthCm SepalWidthCm          Species
  1            1.4          0.2           5.1          3.5      Iris-setosa
  2            1.4          0.2           4.9          3.0      Iris-setosa
  3            1.3          0.2           4.7          3.2      Iris-setosa
  4            1.5          0.2           4.6          3.1      Iris-setosa
...            ...          ...           ...          ...              ...
146            5.2          2.3           6.7          3.0   Iris-virginica
147            5.0          1.9           6.3          2.5   Iris-virginica
148            5.2          2.0           6.5          3.0   Iris-virginica
149            5.4          2.3           6.2          3.4   Iris-virginica
150            5.1          1.8           5.9          3.0   Iris-virginica

151 rows × 5 columns
```

```
#Sort values by column name.
iris.sort_values(by="Sepal_Length")
```

```
      Sepal_Length  Sepal_Width  Petal_Length  Petal_Width          Species
 14            4.3          3.0           1.1          0.1      Iris-setosa
 39            4.4          3.0           1.3          0.2      Iris-setosa
 43            4.4          3.2           1.3          0.2      Iris-setosa
  9            4.4          2.9           1.4          0.2      Iris-setosa
 42            4.5          2.3           1.3          0.3      Iris-setosa
...            ...          ...           ...          ...              ...
119            7.7          2.6           6.9          2.3   Iris-virginica
118            7.7          3.8           6.7          2.2   Iris-virginica
136            7.7          3.0           6.1          2.3   Iris-virginica
132            7.9          3.8           6.4          2.0   Iris-virginica
 Id  SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm          Species

151 rows × 5 columns
```

```
iris.sort_values(by="Petal_Length")
```

```
      Sepal_Length  Sepal_Width  Petal_Length  Petal_Width          Species
 23            4.6          3.6           1.0          0.2      Iris-setosa
 14            4.3          3.0           1.1          0.1      Iris-setosa
 36            5.0          3.2           1.2          0.2      Iris-setosa
 15            5.8          4.0           1.2          0.2      Iris-setosa
 42            4.5          2.3           1.3          0.3      Iris-setosa
...            ...          ...           ...          ...              ...
106            7.6          3.0           6.6          2.1   Iris-virginica
123            7.7          2.8           6.7          2.0   Iris-virginica
118            7.7          3.8           6.7          2.2   Iris-virginica
119            7.7          2.6           6.9          2.3   Iris-virginica
 Id  SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm          Species

151 rows × 5 columns
```

```
#Purely integer-location based indexing for selection by position.
iris.iloc[5]
```

```
Sepal_Length            5.0
Sepal_Width             3.6
Petal_Length            1.4
Petal_Width             0.2
Species         Iris-setosa
Name: 5, dtype: object
```

```
iris.iloc[10]
```

```
Sepal_Length            4.9
Sepal_Width             3.1
Petal_Length            1.5
Petal_Width             0.1
Species         Iris-setosa
Name: 10, dtype: object
```

```
iris.iloc[134]
```

```
Sepal_Length               6.3
Sepal_Width                2.8
Petal_Length               5.1
Petal_Width                1.5
Species         Iris-virginica
Name: 134, dtype: object
```

```
#Selecting via [], which slices the rows.
iris[0:3]
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa

```
iris[10:38]
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1.0	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa
25	4.8	3.4	1.9	0.2	Iris-setosa
26	5.0	3.0	1.6	0.2	Iris-setosa
27	5.0	3.4	1.6	0.4	Iris-setosa
28	5.2	3.5	1.5	0.2	Iris-setosa
29	5.2	3.4	1.4	0.2	Iris-setosa
30	4.7	3.2	1.6	0.2	Iris-setosa
31	4.8	3.1	1.6	0.2	Iris-setosa
32	5.4	3.4	1.5	0.4	Iris-setosa
33	5.2	4.1	1.5	0.1	Iris-setosa
34	5.5	4.2	1.4	0.2	Iris-setosa
35	4.9	3.1	1.5	0.1	Iris-setosa
36	5.0	3.2	1.2	0.2	Iris-setosa
37	5.5	3.5	1.3	0.2	Iris-setosa

```
#Selection by label
iris.loc[:, ["Sepal_Length", "Sepal_Width"]]
```

	Sepal_Length	Sepal_Width
Id	SepalLengthCm	SepalWidthCm
1	5.1	3.5
2	4.9	3.0
3	4.7	3.2
4	4.6	3.1
...	...	...
146	6.7	3.0
147	6.3	2.5
148	6.5	3.0
149	6.2	3.4
150	5.9	3.0

151 rows × 2 columns

```
iris.loc[:, ["Sepal_Length", "Sepal_Width","Species"]]
```

	Sepal_Length	Sepal_Width	Species
Id	SepalLengthCm	SepalWidthCm	Species
1	5.1	3.5	Iris-setosa
2	4.9	3.0	Iris-setosa
3	4.7	3.2	Iris-setosa
4	4.6	3.1	Iris-setosa
...	...	...	...
146	6.7	3.0	Iris-virginica
147	6.3	2.5	Iris-virginica
148	6.5	3.0	Iris-virginica
149	6.2	3.4	Iris-virginica
150	5.9	3.0	Iris-virginica

151 rows × 3 columns

```
#a subset of the first n rows of the original data
iris.iloc[:23, :]
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa

```
#a subset of the first n columns of the original data
iris.iloc[:, :4]
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
...	...	...	...	...
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9
148	6.5	3.0	5.2	2.0
149	6.2	3.4	5.4	2.3
150	5.9	3.0	5.1	1.8

151 rows × 4 columns

```
#a subset of the first m rows and the first n columns
iris.iloc[:23, :4]
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width
Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1
11	5.4	3.7	1.5	0.2
12	4.8	3.4	1.6	0.2
13	4.8	3.0	1.4	0.1
14	4.3	3.0	1.1	0.1
15	5.8	4.0	1.2	0.2
16	5.7	4.4	1.5	0.4
17	5.4	3.9	1.3	0.4
18	5.1	3.5	1.4	0.3
19	5.7	3.8	1.7	0.3
20	5.1	3.8	1.5	0.3
21	5.4	3.4	1.7	0.2
22	5.1	3.7	1.5	0.4

```
#slice the data
iris.iloc[3:5, 0:2]
```

	Sepal_Length	Sepal_Width
3	4.7	3.2
4	4.6	3.1

```
iris.iloc[0:2, 3:5]
```

	Petal_Width	Species
Id	PetalWidthCm	Species
1	0.2	Iris-setosa

```
iris.iloc[3:10, 3:15]
```

	Petal_Width	Species
3	0.2	Iris-setosa
4	0.2	Iris-setosa
5	0.2	Iris-setosa
6	0.4	Iris-setosa
7	0.3	Iris-setosa
8	0.2	Iris-setosa
9	0.2	Iris-setosa

```
#By lists of integer position locations, similar to the NumPy/Python style:
iris.iloc[[1, 2,4], [0, 2]]
```

	Sepal_Length	Petal_Length
1	5.1	1.4
2	4.9	1.4
4	4.6	1.5

```
iris.iloc[[11, 23,4], [0, 2,3]]
```

	Sepal_Length	Petal_Length	Petal_Width
11	5.4	1.5	0.2
23	4.6	1.0	0.2
4	4.6	1.5	0.2

```
#For slicing rows explicitly:
iris.iloc[1:21, :]
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa

```
#For slicing Column explicitly:
iris.iloc[:, 1:4]
```

	Sepal_Width	Petal_Length	Petal_Width
Id	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	3.5	1.4	0.2
2	3.0	1.4	0.2
3	3.2	1.3	0.2
4	3.1	1.5	0.2
...	...	...	...
146	3.0	5.2	2.3
147	2.5	5.0	1.9
148	3.0	5.2	2.0
149	3.4	5.4	2.3
150	3.0	5.1	1.8

151 rows × 3 columns

```
#For getting a value explicitly:
iris.iloc[1, 1]
```

```
'3.5'
```

```
iris.iloc[3, 4]
```

```
'Iris-setosa'
```

```
#Accessing Column and Rows by position
iris['Petal_Length'].iloc[1]
```

```
'1.4'
```

```
iris['Species'].iloc[10]
```

```
'Iris-setosa'
```

```
#Get Column Name then get data from column
cols_2_4=iris.columns[1]
iris[cols_2_4]
```

```
Id     SepalWidthCm
1               3.5
2               3.0
3               3.2
4               3.1
           ...
146             3.0
147             2.5
148             3.0
149             3.4
150             3.0
Name: Sepal_Width, Length: 151, dtype: object
```

```
cols_1_4=iris.columns[1:4]
iris[cols_1_4]
```

	Sepal_Width	Petal_Length	Petal_Width
Id	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	3.5	1.4	0.2
2	3.0	1.4	0.2
3	3.2	1.3	0.2
4	3.1	1.5	0.2
...	...	...	...
146	3.0	5.2	2.3
147	2.5	5.0	1.9
148	3.0	5.2	2.0
149	3.4	5.4	2.3
150	3.0	5.1	1.8

151 rows × 3 columns

```
#in one Expression answer for the above two commands
iris[iris.columns[2:4]].iloc[5:10]
```

	Petal_Length	Petal_Width
5	1.4	0.2
6	1.7	0.4
7	1.4	0.3
8	1.5	0.2
9	1.4	0.2

```
iris[iris.columns[1:4]].iloc[5]
```

```
Sepal_Width     3.6
Petal_Length    1.4
Petal_Width     0.2
Name: 5, dtype: object
```

```
#Checking of Missing Values in Dataset:
iris.isnull()
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
Id	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False
150	False	False	False	False	False

151 rows × 5 columns

```
#to get the count of missing values of column and row wise count of missing values
iris.isna()
```

	Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
Id	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False
150	False	False	False	False	False

151 rows × 5 columns

```
#is there any missing values across each column
iris.isnull().any()
```

```
Sepal_Length    False
Sepal_Width     False
Petal_Length    False
Petal_Width     False
Species         False
dtype: bool
```

```
#count of missing values across each column using isna() and isnull()
iris.isnull().sum().sum()
```

```
0
```

```
#count row wise missing value using isnull()
iris.isnull().sum(axis = 1)
```

```
Id     0
1      0
2      0
3      0
4      0
      ..
146    0
147    0
148    0
149    0
```

```
    150    0
    Length: 151, dtype: int64
```

```
#count Column wise missing value using isnull()
#Method 1:
iris.isnull().sum()
```

```
    Sepal_Length    0
    Sepal_Width     0
    Petal_Length    0
    Petal_Width     0
    Species         0
    dtype: int64
```

```
#count Column wise missing value using isnull()
#Method 2:
iris.isna().sum()
```

```
    Sepal_Length    0
    Sepal_Width     0
    Petal_Length    0
    Petal_Width     0
    Species         0
    dtype: int64
```

```
#count of missing values of a specific column.
iris.Sepal_Width.isnull().sum()
```

```
    0
```

```
#groupby count of missing values of a column.
iris.groupby(['Petal_Length'])['Petal_Width'].apply(lambda x:x.isnull().sum())
```

```
    Petal_Length
    1.0             0
    1.1             0
    1.2             0
    1.3             0
    1.4             0
    1.5             0
    1.6             0
    1.7             0
    1.9             0
    3.0             0
    3.3             0
    3.5             0
    3.6             0
    3.7             0
    3.8             0
    3.9             0
    4.0             0
    4.1             0
    4.2             0
    4.3             0
    4.4             0
    4.5             0
    4.6             0
    4.7             0
    4.8             0
    4.9             0
    5.0             0
    5.1             0
    5.2             0
    5.3             0
    5.4             0
    5.5             0
    5.6             0
    5.7             0
    5.8             0
    5.9             0
    6.0             0
    6.1             0
    6.3             0
    6.4             0
    6.6             0
    6.7             0
    6.9             0
    PetalLengthCm    0
    Name: Petal_Width, dtype: int64
```

```
iris.dtypes
```

```
Sepal_Length    object
Sepal_Width     object
Petal_Length    object
Petal_Width     object
Species         object
dtype: object
```

```
#To change the data type (data type of ‘petal length'changed to int)
#iris['Petal_Length']= iris['Petal_Length'].astype("int")
```

Data normalization

```
#Data normalization:

#Import pandas and sklearn library for preprocessing
from sklearn import preprocessing

#Load the iris dataset in dataframe object df
df = pd.read_csv('/content/Iris.csv')

#Print iris dataset.
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
#Create a minimum and maximum processor object
min_max_scaler = preprocessing.MinMaxScaler()

#Separate the feature from the class label
x=df.iloc[:,:4]


#Create an object to transform the data to fit minmax processor
x_scaled = min_max_scaler.fit_transform(x)


#Run the normalizer on the dataframe
df_normalized = pd.DataFrame(x_scaled)

#View the dataframe
df_normalized
```

	0	1	2	3
0	0.000000	0.222222	0.625000	0.067797

Encoding

2 0.013423 0.111111 0.500000 0.050847

```
#Label encoding
#Import pandas and sklearn library for preprocessing
#Load the iris dataset in dataframe object df

#Observe the unique values for the Species column.
df['Species'].unique()
```

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```
#define label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
```

149 1.000000 0.444444 0.416667 0.694915

```
#Encode labels in column 'species'.
df['Species']= label_encoder.fit_transform(df['Species'])
```

```
#Observe the unique values for the Species column.
df['Species'].unique()
```

```
array([0, 1, 2])
```

```
#One-Hot Encoding
#Import pandas and sklearn library for preprocessing

#Load the iris dataset in dataframe object df
```

```
#Observe the unique values for the Species column.
```

```
#Remove the target variable from dataset
features_df=df.drop(columns=['Species'])
```

```
#Apply one_hot encoder for Species column.
enc = preprocessing.OneHotEncoder()
enc_df=pd.DataFrame(enc.fit_transform(df[['Species']]).toarray(), columns = [0, 1, 2])
```

```
#Join the encoded values with Features variable
df_encode = features_df.join(enc_df)
```

```
#Observe the merge dataframe
df_encode
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	0	1	2
0	1	5.1	3.5	1.4	0.2	1.0	0.0	0.0
1	2	4.9	3.0	1.4	0.2	1.0	0.0	0.0
2	3	4.7	3.2	1.3	0.2	1.0	0.0	0.0