# Data Science & AI
# &
# NIC - Param

## Python-For Data Science

## Pandas

One Shot

By- Pankaj Sharma Sir

## Recap of Previous Lecture

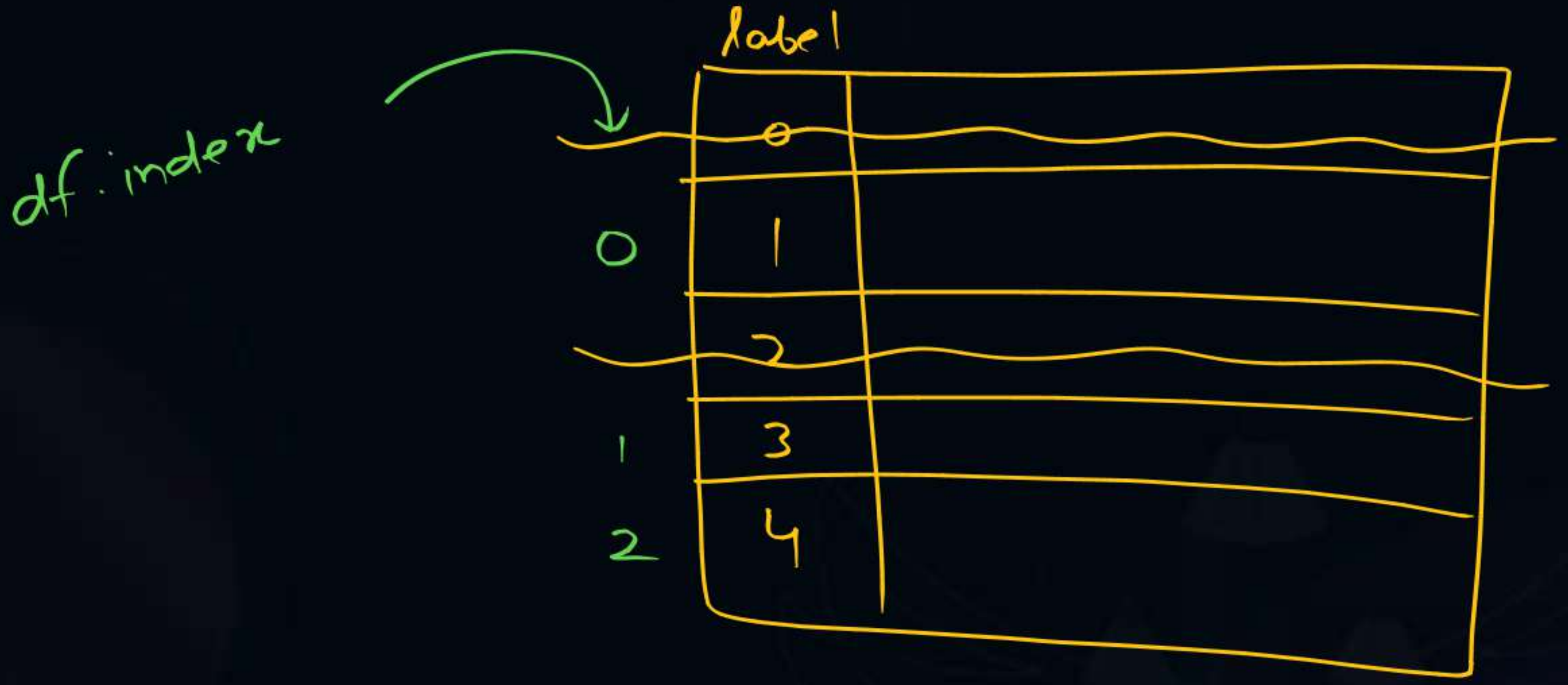
Topic NumPy Part 03

## Topics to be Covered

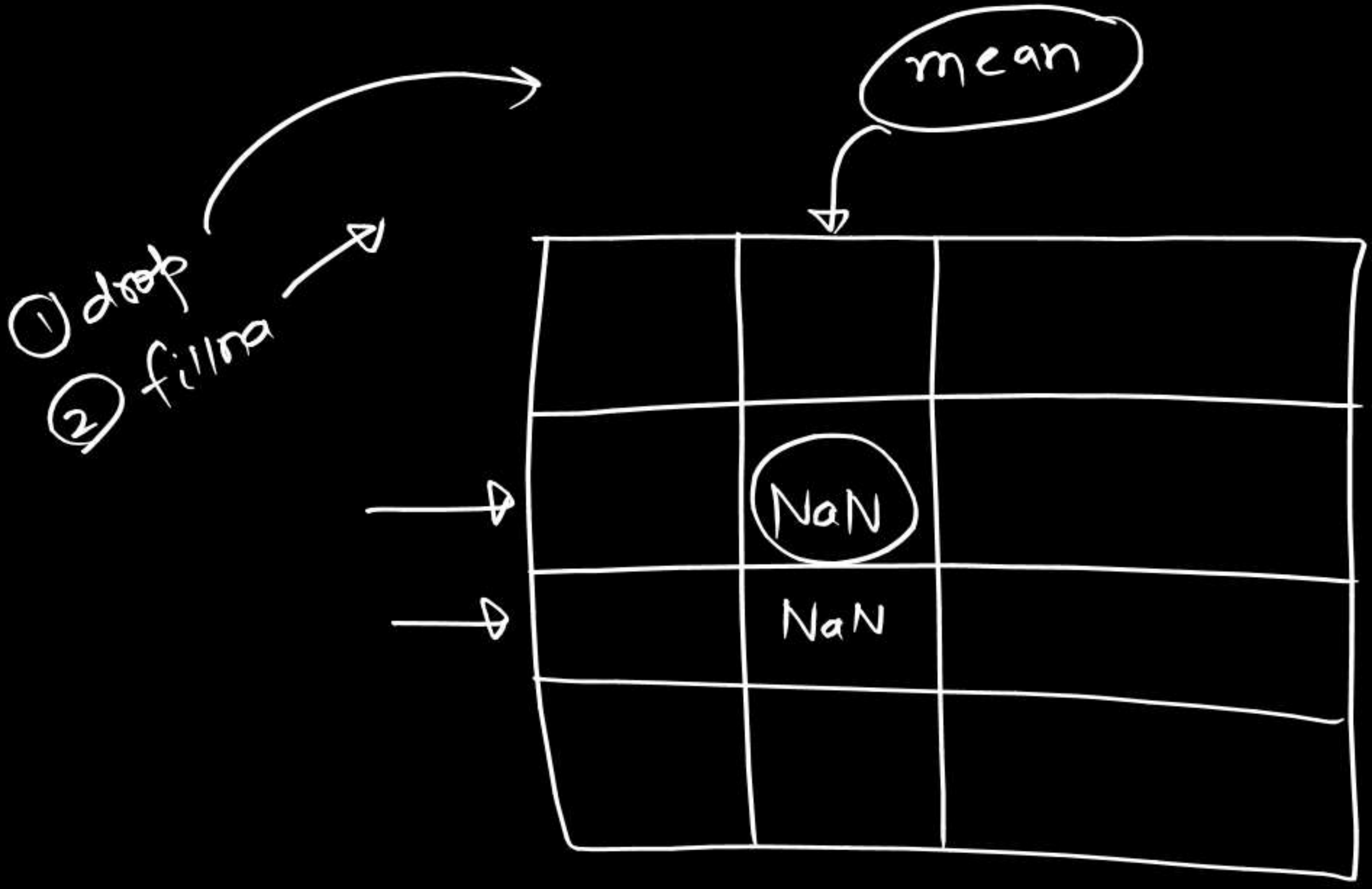
Topic Pandas

Topic : NumPy

# Pandas



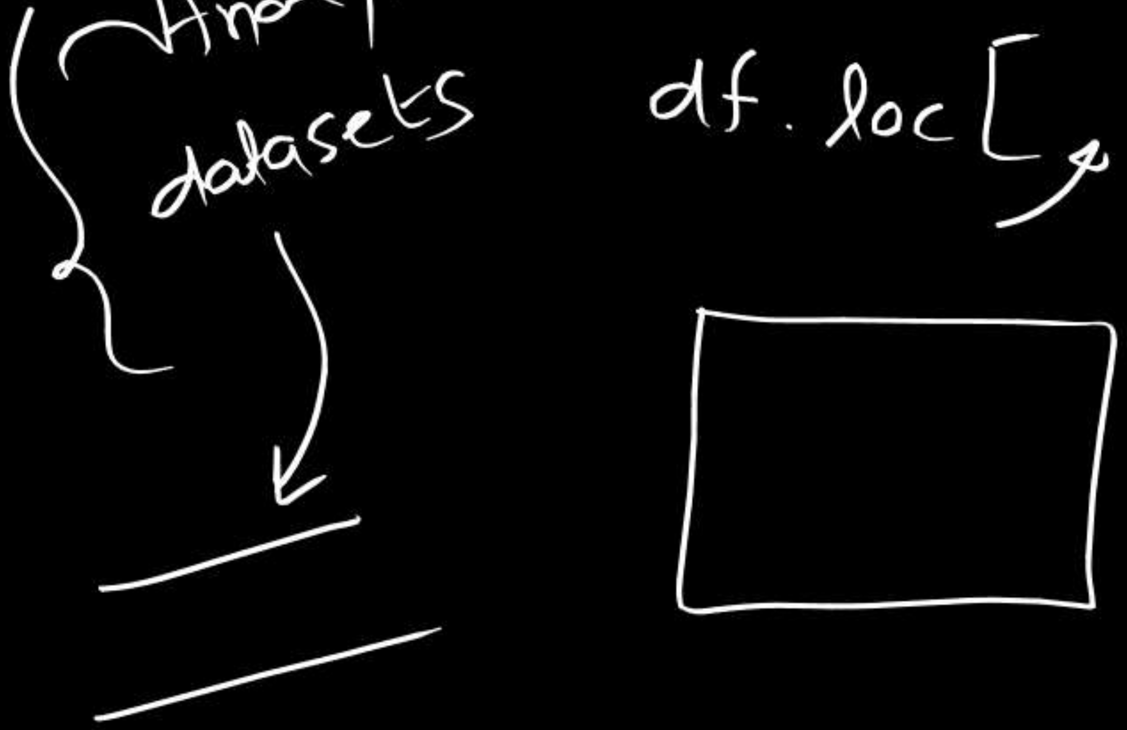
Slide 4


mean
① drop
② fillna
NaN
NaN

Analysis

datasets

df.loc[ $\emptyset$ , $\emptyset$ ] = np.nan



①

read_csv

read_~~~

$\rightarrow$ dataframe

df.head( ) → 5 rows

① df.columns = [ _ _ _ ]

② df.drop(0, inplace = True)

③ df.iloc[0] = [ _ _ ]

④ df['col'] = 1

⑤ df.drop(col_name, axis=1, inplace = True)

⑥ del df[col_name]

Handle NaN

→ drop →

→ fillna

→ meaningful data

Numpy

① Memory
② Time
③ Easy

Broadcasting

slicing, indexing

File handling

zip( )
enumerate( )

The End

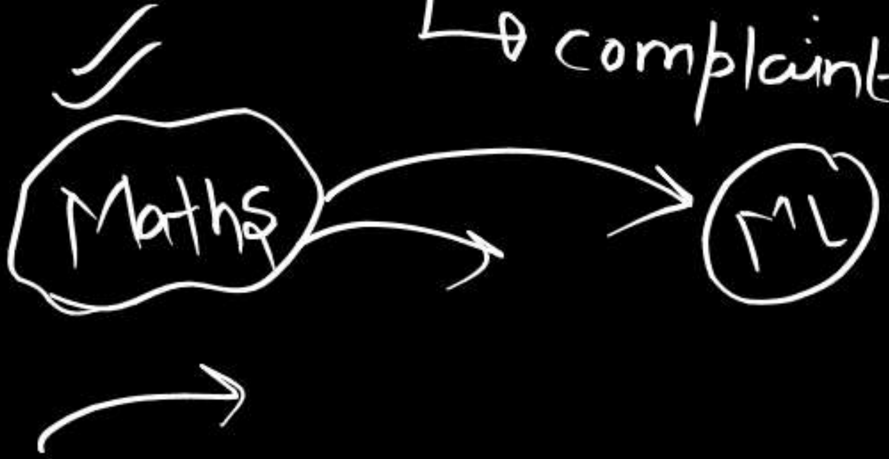
All the best

38 classes

- → review
- → complaint :

ML

90% →



ML

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
iris=pd.read_csv('Desktop\petals.csv')
```

In [3]:
```
df=iris
```

In [4]:
```
df.head()
```

Out[4]:

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | Nan | 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 |

In [5]:
```
df.columns=['sl','sw','pl','pw','kind']
```

In [6]:
```
df.head()
```

Out[6]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | Nan | 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 |

In [7]:
```
df.describe()
```

Out[7]:

| | sl | sw | pw |
|---|---|---|---|
| **count** | 150.000000 | 150.000000 | 150.000000 |
| **mean** | 5.843333 | 3.054000 | 1.198667 |
| **std** | 0.828066 | 0.433594 | 0.763161 |
| **min** | 4.300000 | 2.000000 | 0.100000 |
| **25%** | 5.100000 | 2.800000 | 0.300000 |
| **50%** | 5.800000 | 3.000000 | 1.300000 |
| **75%** | 6.400000 | 3.300000 | 1.800000 |
| **max** | 7.900000 | 4.400000 | 2.500000 |

In [8]:
```
df.iloc[0]
```

Out[8]:
```
sl          5.1
sw          3.5
pl          Nan
pw          0.2
kind     setosa
Name: 0, dtype: object
```

In [9]:
```
df.head()
```

Out[9]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | Nan | 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 |

In [10]:
```
df.iloc[0] #index
```

Out[10]:
```
sl          5.1
sw          3.5
pl          Nan
pw          0.2
kind     setosa
Name: 0, dtype: object
```

In [11]:
```
#delete a row
df.loc[0] #label
```

Out[11]:
```
sl          5.1
sw          3.5
pl          Nan
pw          0.2
kind     setosa
Name: 0, dtype: object
```

In [12]:
```
#index vs label
```

In [13]:
```
df.head()
```

Out[13]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | Nan | 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 |

In [14]:
```
df.drop(0)
```

Out[14]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **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 |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **...** | ... | ... | ... | ... | ... |
| **145** | 6.7 | 3.0 | 5.2 | 2.3 | virginica |
| **146** | 6.3 | 2.5 | 5 | 1.9 | virginica |
| **147** | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| **148** | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| **149** | 5.9 | 3.0 | 5.1 | 1.8 | virginica |

149 rows × 5 columns

In [15]:
```
df.head()
```

Out[15]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | Nan | 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 |

In [16]:
```
df.drop(0,inplace=True)
```

In [17]:
```
df.head()
```

Out[17]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **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 |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |

In [18]:
```
df.drop(0,inplace=True)  #no row whose label is 0
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[18], line 1
----> 1 df.drop(0,inplace=True)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\frame.py:5258, in DataFra
me.drop(self, labels, axis, index, columns, level, inplace, errors)
   5110 def drop(
   5111     self,
   5112     labels: IndexLabel = None,
   (...)
   5119     errors: IgnoreRaise = "raise",
   5120 ) -> DataFrame | None:
   5121     """
   5122     Drop specified labels from rows or columns.
   5123 
   (...)
   5256             weight  1.0     0.8
   5257     """
-> 5258     return super().drop(
   5259         labels=labels,
   5260         axis=axis,
   5261         index=index,
   5262         columns=columns,
   5263         level=level,
   5264         inplace=inplace,
   5265         errors=errors,
   5266     )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:4549, in NDFra
me.drop(self, labels, axis, index, columns, level, inplace, errors)
   4547 for axis, labels in axes.items():
   4548     if labels is not None:
-> 4549         obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   4551 if inplace:
   4552     self._update_inplace(obj)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:4591, in NDFra
me._drop_axis(self, labels, axis, level, errors, only_slice)
   4589         new_axis = axis.drop(labels, level=level, errors=errors)
   4590     else:
-> 4591         new_axis = axis.drop(labels, errors=errors)
   4592     indexer = axis.get_indexer(new_axis)
   4594 # Case for non-unique axis
   4595 else:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6699, in 
Index.drop(self, labels, errors)
   6697 if mask.any():
   6698     if errors != "ignore":
-> 6699         raise KeyError(f"{list(labels[mask])} not found in axis")
   6700     indexer = indexer[~mask]
   6701 return self.delete(indexer)

KeyError: '[0] not found in axis'
```

In [21]:
```
df.drop(1,inplace=True) #label 1 wali row ko delete kr dia
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[21], line 1
----> 1 df.drop(1,inplace=True)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\frame.py:5258, in DataFra
me.drop(self, labels, axis, index, columns, level, inplace, errors)
   5110 def drop(
   5111     self,
   5112     labels: IndexLabel = None,
   (...)
   5119     errors: IgnoreRaise = "raise",
   5120 ) -> DataFrame | None:
   5121     """
   5122     Drop specified labels from rows or columns.
   5123 
   (...)
   5256             weight  1.0     0.8
   5257     """
-> 5258     return super().drop(
   5259         labels=labels,
   5260         axis=axis,
   5261         index=index,
   5262         columns=columns,
   5263         level=level,
   5264         inplace=inplace,
   5265         errors=errors,
   5266     )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:4549, in NDFra
me.drop(self, labels, axis, index, columns, level, inplace, errors)
   4547 for axis, labels in axes.items():
   4548     if labels is not None:
-> 4549         obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   4551 if inplace:
   4552     self._update_inplace(obj)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:4591, in NDFra
me._drop_axis(self, labels, axis, level, errors, only_slice)
   4589         new_axis = axis.drop(labels, level=level, errors=errors)
   4590     else:
-> 4591         new_axis = axis.drop(labels, errors=errors)
   4592     indexer = axis.get_indexer(new_axis)
   4594 # Case for non-unique axis
   4595 else:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6699, in
Index.drop(self, labels, errors)
   6697 if mask.any():
   6698     if errors != "ignore":
-> 6699         raise KeyError(f"{list(labels[mask])} not found in axis")
   6700     indexer = indexer[~mask]
   6701 return self.delete(indexer)

KeyError: '[1] not found in axis'
```

```
In [22]: df.head()
```

Out[22]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **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 |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |

In [23]:
```
df.drop(3,inplace=True)
```

In [24]:
```
df.head()
```

Out[24]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |

In [25]:
```
df.index
```

Out[25]:
```
Index([  2,    4,    5,    6,    7,    8,    9,   10,   11,   12,
       ...
       140, 141, 142, 143, 144, 145, 146, 147, 148, 149],
      dtype='int64', length=147)
```

In [27]:
```
df.index[0],df.index[1]
```

Out[27]:
```
(2, 4)
```

In [29]:
```
df.drop(df.index[0],inplace=True)
```

In [30]:
```
df.head()
```

Out[30]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **8** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |

In [31]:
```
df.index
```

```
Out[31]: Index([  4,   5,   6,   7,   8,   9,  10,  11,  12,  13,
                ...
                140, 141, 142, 143, 144, 145, 146, 147, 148, 149],
               dtype='int64', length=146)
```

```
In [32]: df.head()
```

Out[32]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **8** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |

```
In [34]: df[df.sw>3]
```

Out[34]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **9** | 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| **...** | ... | ... | ... | ... | ... |
| **140** | 6.7 | 3.1 | 5.6 | 2.4 | virginica |
| **141** | 6.9 | 3.1 | 5.1 | 2.3 | virginica |
| **143** | 6.8 | 3.2 | 5.9 | 2.3 | virginica |
| **144** | 6.7 | 3.3 | 5.7 | 2.5 | virginica |
| **148** | 6.2 | 3.4 | 5.4 | 2.3 | virginica |

64 rows × 5 columns

```
In [35]: df[df.kind=='setosa']
```

Out[35]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **8** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| **9** | 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| **10** | 5.4 | 3.7 | 1.5 | 0.2 | setosa |
| **11** | 4.8 | 3.4 | 1.6 | 0.2 | setosa |
| **12** | 4.8 | 3.0 | 1.4 | 0.1 | setosa |
| **13** | 4.3 | 3.0 | 1.1 | 0.1 | setosa |
| **14** | 5.8 | 4.0 | 1.2 | 0.2 | setosa |
| **15** | 5.7 | 4.4 | 1.5 | 0.4 | setosa |
| **16** | 5.4 | 3.9 | 1.3 | 0.4 | setosa |
| **17** | 5.1 | 3.5 | 1.4 | 0.3 | setosa |
| **18** | 5.7 | 3.8 | 1.7 | 0.3 | setosa |
| **19** | 5.1 | 3.8 | 1.5 | 0.3 | setosa |
| **20** | 5.4 | 3.4 | 1.7 | 0.2 | setosa |
| **21** | 5.1 | 3.7 | 1.5 | 0.4 | setosa |
| **22** | 4.6 | 3.6 | 1 | 0.2 | setosa |
| **23** | 5.1 | 3.3 | 1.7 | 0.5 | setosa |
| **24** | 4.8 | 3.4 | 1.9 | 0.2 | setosa |
| **25** | 5.0 | 3.0 | 1.6 | 0.2 | setosa |
| **26** | 5.0 | 3.4 | 1.6 | 0.4 | setosa |
| **27** | 5.2 | 3.5 | 1.5 | 0.2 | setosa |
| **28** | 5.2 | 3.4 | 1.4 | 0.2 | setosa |
| **29** | 4.7 | 3.2 | 1.6 | 0.2 | setosa |
| **30** | 4.8 | 3.1 | 1.6 | 0.2 | setosa |
| **31** | 5.4 | 3.4 | 1.5 | 0.4 | setosa |
| **32** | 5.2 | 4.1 | 1.5 | 0.1 | setosa |
| **33** | 5.5 | 4.2 | 1.4 | 0.2 | setosa |
| **34** | 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| **35** | 5.0 | 3.2 | 1.2 | 0.2 | setosa |
| **36** | 5.5 | 3.5 | 1.3 | 0.2 | setosa |
| **37** | 4.9 | 3.1 | 1.5 | 0.1 | setosa |

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **38** | 4.4 | 3.0 | 1.3 | 0.2 | setosa |
| **39** | 5.1 | 3.4 | 1.5 | 0.2 | setosa |
| **40** | 5.0 | 3.5 | 1.3 | 0.3 | setosa |
| **41** | 4.5 | 2.3 | 1.3 | 0.3 | setosa |
| **42** | 4.4 | 3.2 | 1.3 | 0.2 | setosa |
| **43** | 5.0 | 3.5 | 1.6 | 0.6 | setosa |
| **44** | 5.1 | 3.8 | 1.9 | 0.4 | setosa |
| **45** | 4.8 | 3.0 | 1.4 | 0.3 | setosa |
| **46** | 5.1 | 3.8 | 1.6 | 0.2 | setosa |
| **47** | 4.6 | 3.2 | 1.4 | 0.2 | setosa |
| **48** | 5.3 | 3.7 | 1.5 | 0.2 | setosa |
| **49** | 5.0 | 3.3 | 1.4 | 0.2 | setosa |

In [36]:
```
df.head()
```

Out[36]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **8** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |

In [37]:
```
df.iloc[0]
```

Out[37]:
```
sl          5.0
sw          3.6
pl          1.4
pw          0.2
kind     setosa
Name: 4, dtype: object
```

In [38]:
```
df.loc[6]
```

Out[38]:
```
sl          4.6
sw          3.4
pl          1.4
pw          0.3
kind     setosa
Name: 6, dtype: object
```

In [39]:
```
#new row add
df.loc[0]=[2.3,3.5,4.2,1.4,'setosa']
```

In [40]:
```
df.head()
```

Out[40]:

|   | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **8** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |

In [41]:
```python
df.tail()
```

Out[41]:

|   | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **146** | 6.3 | 2.5 | 5 | 1.9 | virginica |
| **147** | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| **148** | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| **149** | 5.9 | 3.0 | 5.1 | 1.8 | virginica |
| **0** | 2.3 | 3.5 | 4.2 | 1.4 | setosa |

In [42]:
```python
df.reset_index()
```

Out[42]:

|   | index | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|---|
| **0** | 4 | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **1** | 5 | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **2** | 6 | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **3** | 7 | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **4** | 8 | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| **...** | ... | ... | ... | ... | ... | ... |
| **142** | 146 | 6.3 | 2.5 | 5 | 1.9 | virginica |
| **143** | 147 | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| **144** | 148 | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| **145** | 149 | 5.9 | 3.0 | 5.1 | 1.8 | virginica |
| **146** | 0 | 2.3 | 3.5 | 4.2 | 1.4 | setosa |

147 rows × 6 columns

In [44]:
```python
df.reset_index(drop=True,inplace=True)
```

In [45]:
```python
df
```

Out[45]:

|  | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **1** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **2** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **3** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **4** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| **...** | ... | ... | ... | ... | ... |
| **142** | 6.3 | 2.5 | 5 | 1.9 | virginica |
| **143** | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| **144** | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| **145** | 5.9 | 3.0 | 5.1 | 1.8 | virginica |
| **146** | 2.3 | 3.5 | 4.2 | 1.4 | setosa |

147 rows × 5 columns

In [46]:
```
df.head()
```

Out[46]:

|  | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **1** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **2** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **3** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **4** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |

In [47]:
```
df.drop('sl')  #no such row  ==>for columns axis =1
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[47], line 1
----> 1 df.drop('sl')

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\frame.py:5258, in DataFra
me.drop(self, labels, axis, index, columns, level, inplace, errors)
   5110 def drop(
   5111     self,
   5112     labels: IndexLabel = None,
   (...)
   5119     errors: IgnoreRaise = "raise",
   5120 ) -> DataFrame | None:
   5121     """
   5122     Drop specified labels from rows or columns.
   5123 
   (...)
   5256             weight  1.0     0.8
   5257     """
-> 5258     return super().drop(
   5259         labels=labels,
   5260         axis=axis,
   5261         index=index,
   5262         columns=columns,
   5263         level=level,
   5264         inplace=inplace,
   5265         errors=errors,
   5266     )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:4549, in NDFra
me.drop(self, labels, axis, index, columns, level, inplace, errors)
   4547 for axis, labels in axes.items():
   4548     if labels is not None:
-> 4549         obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   4551 if inplace:
   4552     self._update_inplace(obj)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:4591, in NDFra
me._drop_axis(self, labels, axis, level, errors, only_slice)
   4589         new_axis = axis.drop(labels, level=level, errors=errors)
   4590     else:
-> 4591         new_axis = axis.drop(labels, errors=errors)
   4592     indexer = axis.get_indexer(new_axis)
   4594 # Case for non-unique axis
   4595 else:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6699, in
Index.drop(self, labels, errors)
   6697 if mask.any():
   6698     if errors != "ignore":
-> 6699         raise KeyError(f"{list(labels[mask])} not found in axis")
   6700     indexer = indexer[~mask]
   6701 return self.delete(indexer)

KeyError: "['sl'] not found in axis"
```

In [48]:
```
df.head()
```

Out[48]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **1** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **2** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **3** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **4** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |

In [49]:
```
df.drop('sl',axis=1) #delete col with name as 'sl'
```

Out[49]:

| | sw | pl | pw | kind |
|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa |
| **1** | 3.9 | 1.7 | 0.4 | setosa |
| **2** | 3.4 | 1.4 | 0.3 | setosa |
| **3** | 3.4 | 1.5 | 0.2 | setosa |
| **4** | 2.9 | 1.4 | 0.2 | setosa |
| **...** | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica |
| **143** | 3.0 | 5.2 | 2.0 | virginica |
| **144** | 3.4 | 5.4 | 2.3 | virginica |
| **145** | 3.0 | 5.1 | 1.8 | virginica |
| **146** | 3.5 | 4.2 | 1.4 | setosa |

147 rows × 4 columns

In [50]:
```
df
```

Out[50]:

| | sl | sw | pl | pw | kind |
|---|---|---|---|---|---|
| **0** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **1** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **2** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **3** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **4** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| **...** | ... | ... | ... | ... | ... |
| **142** | 6.3 | 2.5 | 5 | 1.9 | virginica |
| **143** | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| **144** | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| **145** | 5.9 | 3.0 | 5.1 | 1.8 | virginica |
| **146** | 2.3 | 3.5 | 4.2 | 1.4 | setosa |

147 rows × 5 columns

In [51]:
```
df.drop('sl',axis=1,inplace=True)
```

In [52]:
```
df
```

Out[52]:

| | sw | pl | pw | kind |
|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa |
| **1** | 3.9 | 1.7 | 0.4 | setosa |
| **2** | 3.4 | 1.4 | 0.3 | setosa |
| **3** | 3.4 | 1.5 | 0.2 | setosa |
| **4** | 2.9 | 1.4 | 0.2 | setosa |
| **...** | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica |
| **143** | 3.0 | 5.2 | 2.0 | virginica |
| **144** | 3.4 | 5.4 | 2.3 | virginica |
| **145** | 3.0 | 5.1 | 1.8 | virginica |
| **146** | 3.5 | 4.2 | 1.4 | setosa |

147 rows × 4 columns

In [58]:
```
df["sum"]=df['sw']+df['pw'] #add a new col
```

In [59]:
```
df
```

Out[59]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | 3.9 | 1.7 | 0.4 | setosa | 4.3 |
| **2** | 3.4 | 1.4 | 0.3 | setosa | 3.7 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |
| **...** | ... | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **143** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **144** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **145** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **146** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

147 rows × 5 columns

In [60]:
```
df.head()
```

Out[60]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | 3.9 | 1.7 | 0.4 | setosa | 4.3 |
| **2** | 3.4 | 1.4 | 0.3 | setosa | 3.7 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |

In [61]:
```
df['col']=1
```

In [62]:
```
df
```

Out[62]:

|  | sw | pl | pw | kind | diff | col |
|---|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 | 1 |
| **1** | 3.9 | 1.7 | 0.4 | setosa | 4.3 | 1 |
| **2** | 3.4 | 1.4 | 0.3 | setosa | 3.7 | 1 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 | 1 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 | 1 |
| **...** | ... | ... | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica | 4.4 | 1 |
| **143** | 3.0 | 5.2 | 2.0 | virginica | 5.0 | 1 |
| **144** | 3.4 | 5.4 | 2.3 | virginica | 5.7 | 1 |
| **145** | 3.0 | 5.1 | 1.8 | virginica | 4.8 | 1 |
| **146** | 3.5 | 4.2 | 1.4 | setosa | 4.9 | 1 |

147 rows × 6 columns

In [63]:
```
del df['col']
```

In [64]:
```
df
```

Out[64]:

|  | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | 3.9 | 1.7 | 0.4 | setosa | 4.3 |
| **2** | 3.4 | 1.4 | 0.3 | setosa | 3.7 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |
| **...** | ... | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **143** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **144** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **145** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **146** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

147 rows × 5 columns

In [65]:
```
df.describe()
```

Out[65]:

| | sw | pw | diff |
|---|---|---|---|
| **count** | 147.000000 | 147.000000 | 147.000000 |
| **mean** | 3.053061 | 1.227211 | 4.280272 |
| **std** | 0.437819 | 0.752672 | 0.725715 |
| **min** | 2.000000 | 0.100000 | 2.600000 |
| **25%** | 2.800000 | 0.350000 | 3.700000 |
| **50%** | 3.000000 | 1.300000 | 4.200000 |
| **75%** | 3.300000 | 1.800000 | 4.800000 |
| **max** | 4.400000 | 2.500000 | 6.100000 |

In [66]:
```
df.head()
```

Out[66]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | 3.9 | 1.7 | 0.4 | setosa | 4.3 |
| **2** | 3.4 | 1.4 | 0.3 | setosa | 3.7 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |

In [67]:
```
df.iloc[1:3,0:2]=np.nan
```

In [68]:
```
df
```

Out[68]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | NaN | NaN | 0.4 | setosa | 4.3 |
| **2** | NaN | NaN | 0.3 | setosa | 3.7 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |
| **...** | ... | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **143** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **144** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **145** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **146** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

147 rows × 5 columns

In [69]:
```
df.describe()
```

Out[69]:

|  | sw | pw | diff |
|---|---|---|---|
| **count** | 145.000000 | 147.000000 | 147.000000 |
| **mean** | 3.044828 | 1.227211 | 4.280272 |
| **std** | 0.434123 | 0.752672 | 0.725715 |
| **min** | 2.000000 | 0.100000 | 2.600000 |
| **25%** | 2.800000 | 0.350000 | 3.700000 |
| **50%** | 3.000000 | 1.300000 | 4.200000 |
| **75%** | 3.300000 | 1.800000 | 4.800000 |
| **max** | 4.400000 | 2.500000 | 6.100000 |

In [70]:
```
df.dropna()
```

Out[70]:

|  | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |
| **5** | 3.1 | 1.5 | 0.1 | setosa | 3.2 |
| **6** | 3.7 | 1.5 | 0.2 | setosa | 3.9 |
| **...** | ... | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **143** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **144** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **145** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **146** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

145 rows × 5 columns

In [71]:
```
df
```

Out[71]:

|  | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | NaN | NaN | 0.4 | setosa | 4.3 |
| **2** | NaN | NaN | 0.3 | setosa | 3.7 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |
| **...** | ... | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **143** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **144** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **145** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **146** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

147 rows × 5 columns

In [72]:
```
df.dropna(inplace=True)
```

In [73]:
```
df
```

Out[73]:

|  | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **3** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **4** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |
| **5** | 3.1 | 1.5 | 0.1 | setosa | 3.2 |
| **6** | 3.7 | 1.5 | 0.2 | setosa | 3.9 |
| **...** | ... | ... | ... | ... | ... |
| **142** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **143** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **144** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **145** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **146** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

145 rows × 5 columns

In [75]:
```
df.reset_index(drop=True,inplace=True)
```

In [76]:
```
df
```

Out[76]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | 3.4 | 1.5 | 0.2 | setosa | 3.6 |
| **2** | 2.9 | 1.4 | 0.2 | setosa | 3.1 |
| **3** | 3.1 | 1.5 | 0.1 | setosa | 3.2 |
| **4** | 3.7 | 1.5 | 0.2 | setosa | 3.9 |
| **...** | ... | ... | ... | ... | ... |
| **140** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **141** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **142** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **143** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **144** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

145 rows × 5 columns

In [78]:
```
df.iloc[1:4,0:2]=np.nan
```

In [79]:
```
df
```

Out[79]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.6 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | NaN | NaN | 0.2 | setosa | 3.6 |
| **2** | NaN | NaN | 0.2 | setosa | 3.1 |
| **3** | NaN | NaN | 0.1 | setosa | 3.2 |
| **4** | 3.7 | 1.5 | 0.2 | setosa | 3.9 |
| **...** | ... | ... | ... | ... | ... |
| **140** | 2.5 | 5 | 1.9 | virginica | 4.4 |
| **141** | 3.0 | 5.2 | 2.0 | virginica | 5.0 |
| **142** | 3.4 | 5.4 | 2.3 | virginica | 5.7 |
| **143** | 3.0 | 5.1 | 1.8 | virginica | 4.8 |
| **144** | 3.5 | 4.2 | 1.4 | setosa | 4.9 |

145 rows × 5 columns

In [80]:
```
df.sw.mean()
```

Out[80]:
```
3.0429577464788733
```

In [81]:
```
df.fillna()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[81], line 1
----> 1 df.fillna()

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\frame.py:5493, in DataFra
me.fillna(self, value, method, axis, inplace, limit, downcast)
   5482 @doc(NDFrame.fillna, **_shared_doc_kwargs)
   5483 def fillna(
   5484     self,
   (...)
   5491     downcast: dict | None = None,
   5492 ) -> DataFrame | None:
-> 5493     return super().fillna(
   5494         value=value,
   5495         method=method,
   5496         axis=axis,
   5497         inplace=inplace,
   5498         limit=limit,
   5499         downcast=downcast,
   5500     )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:6859, in NDFra
me.fillna(self, value, method, axis, inplace, limit, downcast)
   6746 """
   6747 Fill NA/NaN values using the specified method.
   6748
   (...)
   6856 Note that column D is not affected since it is not present in df2.
   6857 """
   6858 inplace = validate_bool_kwarg(inplace, "inplace")
-> 6859 value, method = validate_fillna_kwargs(value, method)
   6861 # set the default here, so functions examining the signaure
   6862 # can detect if something was set (e.g. in groupby) (GH9221)
   6863 if axis is None:

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\util\_validators.py:288, in va
lidate_fillna_kwargs(value, method, validate_scalar_dict_value)
    285 from pandas.core.missing import clean_fill_method
    287 if value is None and method is None:
--> 288     raise ValueError("Must specify a fill 'value' or 'method'.")
    289 if value is None and method is not None:
    290     method = clean_fill_method(method)

ValueError: Must specify a fill 'value' or 'method'.
```

In [82]:
```
df.sw.fillna(df.sw.mean(),inplace=True)
```

In [83]:
```
df
```

Out[83]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.600000 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | 3.042958 | NaN | 0.2 | setosa | 3.6 |
| **2** | 3.042958 | NaN | 0.2 | setosa | 3.1 |
| **3** | 3.042958 | NaN | 0.1 | setosa | 3.2 |
| **4** | 3.700000 | 1.5 | 0.2 | setosa | 3.9 |
| **...** | ... | ... | ... | ... | ... |
| **140** | 2.500000 | 5 | 1.9 | virginica | 4.4 |
| **141** | 3.000000 | 5.2 | 2.0 | virginica | 5.0 |
| **142** | 3.400000 | 5.4 | 2.3 | virginica | 5.7 |
| **143** | 3.000000 | 5.1 | 1.8 | virginica | 4.8 |
| **144** | 3.500000 | 4.2 | 1.4 | setosa | 4.9 |

145 rows × 5 columns

In [84]:
```
df.pl.mean()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[84], line 1
----> 1 df.pl.mean()

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:11556, in NDFrame._add_numeric_operations.<locals>.mean(self, axis, skipna, numeric_only, **kwargs)
  11539 @doc(
  11540     _num_doc,
  11541     desc="Return the mean of the values over the requested axis.",
   (...)
  11554     **kwargs,
  11555 ):
> 11556     return NDFrame.mean(self, axis, skipna, numeric_only, **kwargs)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:11201, in NDFrame.mean(self, axis, skipna, numeric_only, **kwargs)
  11194 def mean(
  11195     self,
  11196     axis: Axis | None = 0,
   (...)
  11199     **kwargs,
  11200 ) -> Series | float:
> 11201     return self._stat_function(
  11202         "mean", nanops.nanmean, axis, skipna, numeric_only, **kwargs
  11203     )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:11158, in NDFrame._stat_function(self, name, func, axis, skipna, numeric_only, **kwargs)
  11154     nv.validate_stat_func((), kwargs, fname=name)
  11156 validate_bool_kwarg(skipna, "skipna", none_allowed=False)
> 11158 return self._reduce(
  11159     func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_only
  11160 )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\series.py:4670, in Series._reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4665     raise TypeError(
   4666         f"Series.{name} does not allow {kwd_name}={numeric_only} "
   4667         "with non-numeric dtypes."
   4668     )
   4669 with np.errstate(all="ignore"):
-> 4670     return op(delegate, skipna=skipna, **kwds)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:96, in disallow.__call__.<locals>._f(*args, **kwargs)
     94 try:
     95     with np.errstate(invalid="ignore"):
---> 96         return f(*args, **kwargs)
     97 except ValueError as e:
     98     # we want to transform an object array
     99     # ValueError message to the more typical TypeError
    100     # e.g. this is normally a disallowed function on
    101     # object arrays that contain strings
    102     if is_object_dtype(args[0]):

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:158, in bottleneck_switch.__call__.<locals>.f(values, axis, skipna, **kwds)
    156         result = alt(values, axis=axis, skipna=skipna, **kwds)
    157 else:
```

```
--> 158     result = alt(values, axis=axis, skipna=skipna, **kwds)
    160 return result

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:421, in _dateti
melike_compat.<locals>.new_func(values, axis, skipna, mask, **kwargs)
    418 if datetimelike and mask is None:
    419     mask = isna(values)
--> 421 result = func(values, axis=axis, skipna=skipna, mask=mask, **kwargs)
    423 if datetimelike:
    424     result = _wrap_results(result, orig_values.dtype, fill_value=iNaT)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:727, in nanmean
(values, axis, skipna, mask)
    724     dtype_count = dtype
    726 count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 727 the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    729 if axis is not None and getattr(the_sum, "ndim", False):
    730     count = cast(np.ndarray, count)

File C:\ProgramData\anaconda3\Lib\site-packages\numpy\core\_methods.py:49, in _sum(a,
axis, dtype, out, keepdims, initial, where)
     47 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
     48          initial=_NoValue, where=True):
---> 49     return umr_sum(a, axis, dtype, out, keepdims, initial, where)

TypeError: can only concatenate str (not "int") to str
```

In [85]:
```
df.columns
```

Out[85]:
```
Index(['sw', 'pl', 'pw', 'kind', 'diff'], dtype='object')
```

In [86]:
```
df.pl.describe()
```

Out[86]:
```
count     142
unique     44
top       1.5
freq       11
Name: pl, dtype: object
```

In [87]:
```
df.pl.mean()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[87], line 1
----> 1 df.pl.mean()

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:11556, in NDFr
ame._add_numeric_operations.<locals>.mean(self, axis, skipna, numeric_only, **kwargs)
  11539 @doc(
  11540     _num_doc,
  11541     desc="Return the mean of the values over the requested axis.",
   (...)
  11554     **kwargs,
  11555 ):
> 11556     return NDFrame.mean(self, axis, skipna, numeric_only, **kwargs)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:11201, in NDFr
ame.mean(self, axis, skipna, numeric_only, **kwargs)
  11194 def mean(
  11195     self,
  11196     axis: Axis | None = 0,
   (...)
  11199     **kwargs,
  11200 ) -> Series | float:
> 11201     return self._stat_function(
  11202         "mean", nanops.nanmean, axis, skipna, numeric_only, **kwargs
  11203     )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\generic.py:11158, in NDFr
ame._stat_function(self, name, func, axis, skipna, numeric_only, **kwargs)
  11154     nv.validate_stat_func((), kwargs, fname=name)
  11156 validate_bool_kwarg(skipna, "skipna", none_allowed=False)
> 11158 return self._reduce(
  11159     func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_only
  11160 )

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\series.py:4670, in Serie
s._reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4665     raise TypeError(
   4666         f"Series.{name} does not allow {kwd_name}={numeric_only} "
   4667         "with non-numeric dtypes."
   4668     )
   4669 with np.errstate(all="ignore"):
-> 4670     return op(delegate, skipna=skipna, **kwds)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:96, in disallo
w.__call__.<locals>._f(*args, **kwargs)
     94 try:
     95     with np.errstate(invalid="ignore"):
---> 96         return f(*args, **kwargs)
     97 except ValueError as e:
     98     # we want to transform an object array
     99     # ValueError message to the more typical TypeError
    100     # e.g. this is normally a disallowed function on
    101     # object arrays that contain strings
    102     if is_object_dtype(args[0]):

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:158, in bottlen
eck_switch.__call__.<locals>.f(values, axis, skipna, **kwds)
    156         result = alt(values, axis=axis, skipna=skipna, **kwds)
    157 else:
```

```
--> 158     result = alt(values, axis=axis, skipna=skipna, **kwds)
    160 return result

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:421, in _dateti
melike_compat.<locals>.new_func(values, axis, skipna, mask, **kwargs)
    418 if datetimelike and mask is None:
    419     mask = isna(values)
--> 421 result = func(values, axis=axis, skipna=skipna, mask=mask, **kwargs)
    423 if datetimelike:
    424     result = _wrap_results(result, orig_values.dtype, fill_value=iNaT)

File C:\ProgramData\anaconda3\Lib\site-packages\pandas\core\nanops.py:727, in nanmean
(values, axis, skipna, mask)
    724     dtype_count = dtype
    726 count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 727 the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    729 if axis is not None and getattr(the_sum, "ndim", False):
    730     count = cast(np.ndarray, count)

File C:\ProgramData\anaconda3\Lib\site-packages\numpy\core\_methods.py:49, in _sum(a,
axis, dtype, out, keepdims, initial, where)
     47 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
     48          initial=_NoValue, where=True):
---> 49     return umr_sum(a, axis, dtype, out, keepdims, initial, where)

TypeError: can only concatenate str (not "int") to str
```

In [89]:
```
df.pl.fillna(0,inplace=True)
```

In [90]:
```
df
```

Out[90]:

| | sw | pl | pw | kind | diff |
|---|---|---|---|---|---|
| **0** | 3.600000 | 1.4 | 0.2 | setosa | 3.8 |
| **1** | 3.042958 | 0 | 0.2 | setosa | 3.6 |
| **2** | 3.042958 | 0 | 0.2 | setosa | 3.1 |
| **3** | 3.042958 | 0 | 0.1 | setosa | 3.2 |
| **4** | 3.700000 | 1.5 | 0.2 | setosa | 3.9 |
| **...** | ... | ... | ... | ... | ... |
| **140** | 2.500000 | 5 | 1.9 | virginica | 4.4 |
| **141** | 3.000000 | 5.2 | 2.0 | virginica | 5.0 |
| **142** | 3.400000 | 5.4 | 2.3 | virginica | 5.7 |
| **143** | 3.000000 | 5.1 | 1.8 | virginica | 4.8 |
| **144** | 3.500000 | 4.2 | 1.4 | setosa | 4.9 |

145 rows × 5 columns

In [91]:
```
#either drop rows ==>rows are less
#fillna ==>other meaningful values
```

```
In [ ]:
```

THANK - YOU