**import** pandas **as** pd *#standard opening*

**import** numpy **as** np

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

In [6]:



df **=** pd.read\_csv("iris\_dataset.csv") *# to open a df in the same folder*

In [14]:



df.head(1) *# to see the top (x) rows*

Out[14]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_DColumn** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 | 0 | 12 |

In [23]:



df.tail(1) *# to see the botom x rows*

Out[23]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_DColumn** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **149** | 5.9 | 3.0 | 5.1 | 1.8 | 2 | 0 | 12 |

In [12]:



df['New\_Column']**=**12 *# adds a new column to the df (inspace = True)*

*# would keep the new column in the original df*

In [70]:



​

**---------------------------------------------------------------------------**

**NameError** Traceback (most recent call last)

Cell **In[70], line 1**

**----> 1** df['New\_column']= Col\_data

**NameError**: name 'Col\_data' is not defined

In [15]:



df.head(5)

Out[15]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_DColumn** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 | 0 | 12 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 | 0 | 12 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 | 0 | 12 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 | 0 | 12 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 | 0 | 12 |

In [16]:



df.rank(1) *# ranks the df based on the column I choose*

Out[16]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_DColumn** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 6.0 | 5.0 | 4.0 | 3.0 | 1.5 | 1.5 | 7.0 |
| **1** | 6.0 | 5.0 | 4.0 | 3.0 | 1.5 | 1.5 | 7.0 |
| **2** | 6.0 | 5.0 | 4.0 | 3.0 | 1.5 | 1.5 | 7.0 |
| **3** | 6.0 | 5.0 | 4.0 | 3.0 | 1.5 | 1.5 | 7.0 |
| **4** | 6.0 | 5.0 | 4.0 | 3.0 | 1.5 | 1.5 | 7.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **145** | 6.0 | 4.0 | 5.0 | 3.0 | 2.0 | 1.0 | 7.0 |
| **146** | 6.0 | 4.0 | 5.0 | 2.0 | 3.0 | 1.0 | 7.0 |
| **147** | 6.0 | 4.0 | 5.0 | 2.5 | 2.5 | 1.0 | 7.0 |
| **148** | 6.0 | 4.0 | 5.0 | 3.0 | 2.0 | 1.0 | 7.0 |
| **149** | 6.0 | 4.0 | 5.0 | 2.0 | 3.0 | 1.0 | 7.0 |

150 rows × 7 columns

In [26]:



df.sort\_values(by**=**'sepal width (cm)', ascending **=** **True**).head() *# sorts by the chosen column*

​

Out[26]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_DColumn** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **60** | 5.0 | 2.0 | 3.5 | 1.0 | 1 | 0 | 12 |
| **62** | 6.0 | 2.2 | 4.0 | 1.0 | 1 | 0 | 12 |
| **119** | 6.0 | 2.2 | 5.0 | 1.5 | 2 | 0 | 12 |
| **68** | 6.2 | 2.2 | 4.5 | 1.5 | 1 | 0 | 12 |
| **41** | 4.5 | 2.3 | 1.3 | 0.3 | 0 | 0 | 12 |

In [27]:



df.sort\_values(by**=**'sepal width (cm)', ascending **=** **False**).head() *# sorts as above*

*#but descending*

Out[27]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_DColumn** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | 5.7 | 4.4 | 1.5 | 0.4 | 0 | 0 | 12 |
| **33** | 5.5 | 4.2 | 1.4 | 0.2 | 0 | 0 | 12 |
| **32** | 5.2 | 4.1 | 1.5 | 0.1 | 0 | 0 | 12 |
| **14** | 5.8 | 4.0 | 1.2 | 0.2 | 0 | 0 | 12 |
| **16** | 5.4 | 3.9 | 1.3 | 0.4 | 0 | 0 | 12 |

In [17]:



df1**=**[['row0', 'row1', 'row2']] *# creates a brand new df*

In [18]:



df1

Out[18]:

[['row0', 'row1', 'row2']]

In [19]:



top5\_rows**=**df.head(5) *# creates a new df with the number of rows I*

*# determine in the brackets*

In [20]:



top5\_rows

Out[20]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_DColumn** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 | 0 | 12 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 | 0 | 12 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 | 0 | 12 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 | 0 | 12 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 | 0 | 12 |

In [22]:



top5\_rows.drop('New\_DColumn', axis**=**1) *# removes the selected column*

Out[22]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** | **New\_Column** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 | 12 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 | 12 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 | 12 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 | 12 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 | 12 |

In [31]:



df.columns**=**['Column 1', 'Column 2', 'Column 3', 'Column 4', 'Column 5', 'Column 6', 'Column 7',]

*# changes the names of the columns*

In [32]:



df

Out[32]:

|  | **Column 1** | **Column 2** | **Column 3** | **Column 4** | **Column 5** | **Column 6** | **Column 7** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 | 0 | 12 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 | 0 | 12 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 | 0 | 12 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 | 0 | 12 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 | 0 | 12 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **145** | 6.7 | 3.0 | 5.2 | 2.3 | 2 | 0 | 12 |
| **146** | 6.3 | 2.5 | 5.0 | 1.9 | 2 | 0 | 12 |
| **147** | 6.5 | 3.0 | 5.2 | 2.0 | 2 | 0 | 12 |
| **148** | 6.2 | 3.4 | 5.4 | 2.3 | 2 | 0 | 12 |
| **149** | 5.9 | 3.0 | 5.1 | 1.8 | 2 | 0 | 12 |

150 rows × 7 columns

In [33]:



df **=** pd.read\_csv("iris\_dataset.csv")

In [36]:



df.head(5)

Out[36]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 |

In [37]:



df.shape *# give the overall size of the df = 150 rows, 5 columns*

Out[37]:

(150, 5)

In [38]:



df.columns

Out[38]:

Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',

'petal width (cm)', 'target'],

dtype='object')

In [39]:



df.index

Out[39]:

RangeIndex(start=0, stop=150, step=1)

In [40]:



df.count() *# gives number of non Null values*

Out[40]:

sepal length (cm) 150

sepal width (cm) 150

petal length (cm) 150

petal width (cm) 150

target 150

dtype: int64

In [43]:



​

**Cell In[43], line 1**

**df['New\_Column']=?**

**^**

**SyntaxError:** invalid syntax

In [44]:



df.min() *# finds minimum values from the df*

Out[44]:

sepal length (cm) 4.3

sepal width (cm) 2.0

petal length (cm) 1.0

petal width (cm) 0.1

target 0.0

dtype: float64

In [45]:



df.max()

Out[45]:

sepal length (cm) 7.9

sepal width (cm) 4.4

petal length (cm) 6.9

petal width (cm) 2.5

target 2.0

dtype: float64

In [46]:



df.mean()

Out[46]:

sepal length (cm) 5.843333

sepal width (cm) 3.057333

petal length (cm) 3.758000

petal width (cm) 1.199333

target 1.000000

dtype: float64

In [48]:



df.median()

Out[48]:

sepal length (cm) 5.80

sepal width (cm) 3.00

petal length (cm) 4.35

petal width (cm) 1.30

target 1.00

dtype: float64

In [49]:



df.describe() *# gives all SD values*

Out[49]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **count** | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 |
| **mean** | 5.843333 | 3.057333 | 3.758000 | 1.199333 | 1.000000 |
| **std** | 0.828066 | 0.435866 | 1.765298 | 0.762238 | 0.819232 |
| **min** | 4.300000 | 2.000000 | 1.000000 | 0.100000 | 0.000000 |
| **25%** | 5.100000 | 2.800000 | 1.600000 | 0.300000 | 0.000000 |
| **50%** | 5.800000 | 3.000000 | 4.350000 | 1.300000 | 1.000000 |
| **75%** | 6.400000 | 3.300000 | 5.100000 | 1.800000 | 2.000000 |
| **max** | 7.900000 | 4.400000 | 6.900000 | 2.500000 | 2.000000 |

In [54]:



df[['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)' ]]

*# and so on*

Out[54]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** |
| --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 |
| **1** | 4.9 | 3.0 | 1.4 |
| **2** | 4.7 | 3.2 | 1.3 |
| **3** | 4.6 | 3.1 | 1.5 |
| **4** | 5.0 | 3.6 | 1.4 |
| **...** | ... | ... | ... |
| **145** | 6.7 | 3.0 | 5.2 |
| **146** | 6.3 | 2.5 | 5.0 |
| **147** | 6.5 | 3.0 | 5.2 |
| **148** | 6.2 | 3.4 | 5.4 |
| **149** | 5.9 | 3.0 | 5.1 |

150 rows × 3 columns

In [62]:



df['sepal length (cm)'].where(df['sepal length (cm)'] **>** 6.5)

*# returns rows where the query is satisfied (Boolean)*

Out[62]:

0 NaN

1 NaN

2 NaN

3 NaN

4 NaN

...

145 6.7

146 NaN

147 NaN

148 NaN

149 NaN

Name: sepal length (cm), Length: 150, dtype: float64

In [ ]:



pd.merge(df, dfx, how**=**'left', on 'column')

*# merges 2 dfs with the same column, can be right, inner, outer*

In [66]:



df.target.unique() *#unique values in the column*

Out[66]:

array([0, 1, 2], dtype=int64)

In [67]:



df.target.value\_counts() *# number of unique values in the column*

Out[67]:

0 50

1 50

2 50

Name: target, dtype: int64

In [68]:



df.isnull()

Out[68]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **0** | 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 |
| **...** | ... | ... | ... | ... | ... |
| **145** | 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 rows × 5 columns

In [71]:



grouped**=**df.groupby(['petal width (cm)'])

In [73]:



grouped.head()

Out[73]:

|  | **sepal length (cm)** | **sepal width (cm)** | **petal length (cm)** | **petal width (cm)** | **target** |
| --- | --- | --- | --- | --- | --- |
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **140** | 6.7 | 3.1 | 5.6 | 2.4 | 2 |
| **141** | 6.9 | 3.1 | 5.1 | 2.3 | 2 |
| **142** | 5.8 | 2.7 | 5.1 | 1.9 | 2 |
| **144** | 6.7 | 3.3 | 5.7 | 2.5 | 2 |
| **146** | 6.3 | 2.5 | 5.0 | 1.9 | 2 |

90 rows × 5 columns