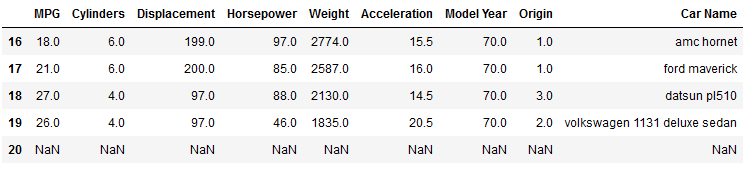
**Program: 14 Numpy Functions on Datasets (Query)**

import pandas as pd

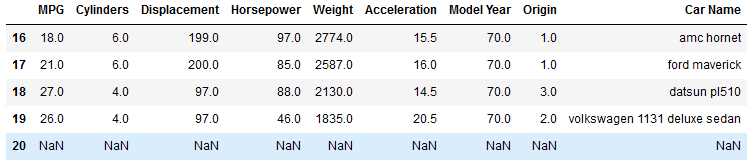
import numpy as np

data=pd.read\_csv("C:\\Users\\BASAPPA SIR\\Desktop\\Dataset\\auto-mpg.csv")

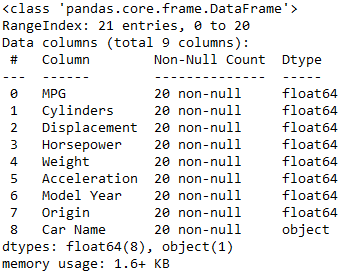
data.head()

****

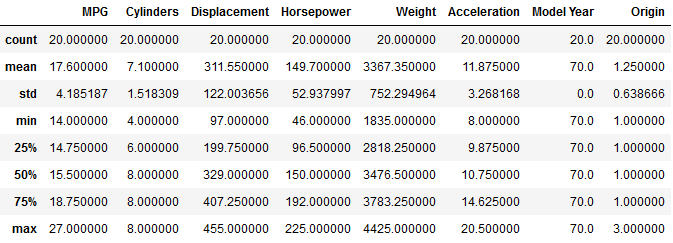
**data.tail()**



**data.info()**



**data.describe()**



data.size

189

data.shape

(21, 9)

**Program: 15 Numpy Functions on Datasets (Aggregate functions)**

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

iris=pd.read\_csv("C:\\Users\\BASAPPA SIR\\Desktop\\Dataset\\iris1.csv")

display(iris)

**#Plot scatter by comparing**

fig=iris[iris.variety=='Setosa'].plot.scatter(x='petal.length',y='petal.width',color='orange',label='Setosa')

iris[iris.variety=='Versicolor'].plot.scatter(x='petal.length',y='petal.width',color='blue',label='Versicolor',ax=fig)

iris[iris.variety=='Virginica'].plot.scatter(x='petal.length',y='petal.width',color='green',label='Virginica',ax=fig)

fig.set\_xlabel("petal.length")

fig.set\_ylabel("petal.width")

fig.set\_title("petal length and width")

fig=plt.gcf()

fig.set\_size\_inches(8,8)

plt.show()

**#To display all data whose values ==1.4**

petal\_width=iris[iris['petal.length']==1.4]

display(petal\_width)

**#To known the how many times each value in petal length is given**

count=iris.groupby('petal.length')['petal.length'].count()

display(count)

**#To find the sum of petal length column**

sum=iris['petal.length'].sum()

display(sum)

**#To known the max size of sepal width column**

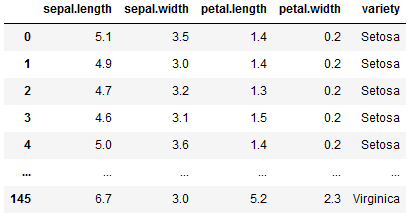
max=iris['sepal.width'].max()

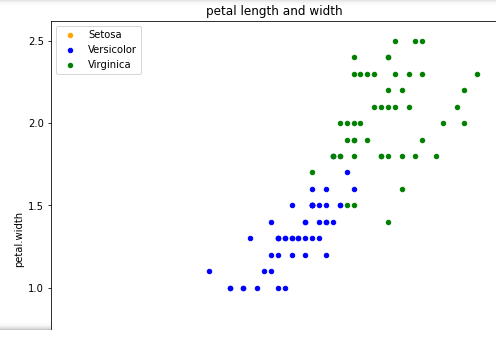
display(max)

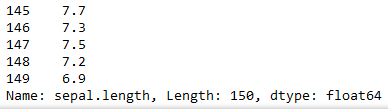
**#To add 1.0 to all the inputs in sepal length column**

add=iris['sepal.length']+1.0

display(add)







**Program: 16 Dimension Reductions**

**# Import necessary libraries**

from sklearn import datasets # to retrieve the iris Dataset

import pandas as pd # to load the dataframe

from sklearn.preprocessing import StandardScaler # to standardize the features

from sklearn.decomposition import PCA # to apply PCA

import seaborn as sns # to plot the heat maps

**#Load the Dataset**

iris = datasets.load\_iris()

df = pd.DataFrame(iris['data'], columns = iris['feature\_names'])

df.head ()

**#Standardize the features**

scalar = StandardScaler()

scaled\_data = pd.DataFrame (scalar.fit\_transform (df)) #scaling the data

scaled\_data

**#Check the Co-relation between features without PCA**

sns.heatmap (scaled\_data.corr())

**# Applying PCA Taking no. of Principal Components as 3**

pca = PCA(n\_components = 3)

pca.fit(scaled\_data)

data\_pca = pca.transform(scaled\_data)

data\_pca = pd.DataFrame(data\_pca,columns=['PC1','PC2','PC3'])

**#Checking Co-relation between features after PCA**

sns.heatmap (data\_pca.corr())

**OUTPUT**

