**Lab Cycle – I (Wine quality dataset)**

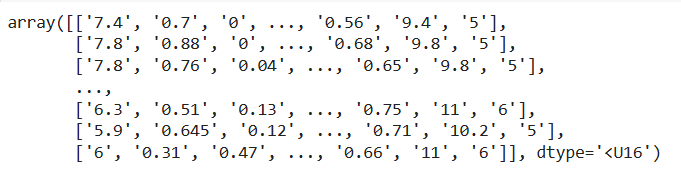
1. Download the dataset winequality-red.csv file( each column is separated by a semicolon (;)) from the UCI Machine Learning Repository.

import numpy as np

arr = np.loadtxt(r"C:\Users\Y20CS68\Downloads\winequality-red.csv",delimiter=";", dtype=str,skiprows=1)

display(arr)

Output:

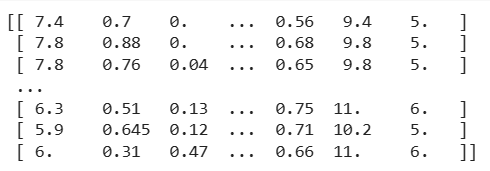


1. Convert it to numPy array, name it as wines (leave the first row of the list) and specify the data type of array as float.

wines=arr.astype('float64')

print(wines)

Output:



1. Identify the shape of the array.

wines.shape

Output:



1. Display the element at row 3 and column 4.

wines[2,3]

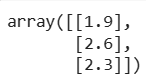
Output:



1. Display the first three items from the fourth column.

wines[:3,[3]]

Output:



1. Display third column from each row.

wines[:,2]

Output:



1. Display fourth row.

wines[3]

Output:



1. Assign value 10 to 2nd row and 6th column element.

wines[1,5]=10

print(wines[1,5])

Output:



1. Take the 10th column from wines array and name that slice as slice\_new and assign value 666 to all elements of slice\_new.

slice\_new=wines[:,9]

slice\_new[:]=666

print(slice\_new)

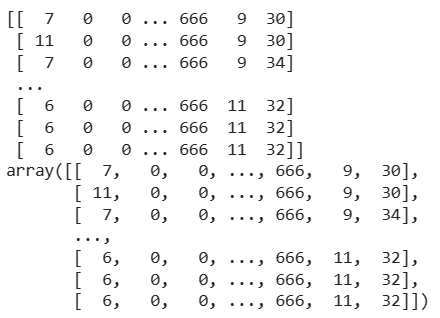
Output:



1. Display wines array.

print(wines)

Output:



1. Find the data type of wines array and Change the data type to int.

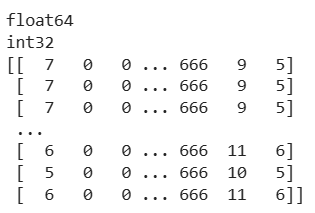
print(wines.dtype)

wines=wines.astype('int')

print(wines.dtype)

print(wines)

Output:



1. Add 10 points to each quality score.

wines[:,-1]+=10

print(wines[:,-1])

Output:



1. Find the sum of all the elements in an array

print(sum(sum(wines)))

print(wines.sum())

Output:



1. Find the sum of all the values in every column.

print(wines.sum(axis=0))

Output:



1. Find the sum of all the values in every row.

print(np.sum(wines, axis=1))

Output:



1. Add the quality column to itself.

wines[:,-1]+=wines[:,-1]

wines[:,-1]

Output:



1. Multiply alcohol by quality

print(wines[:,-2]\* wines[:,-1])

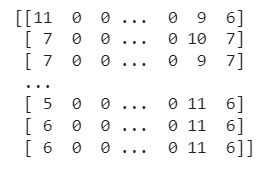
Output:



1. Display which wines have a quality rating higher than 5.

print(wines[wines[:,-1]>5])

Output:



1. Check if any wines have a quality rating equal to 10.

print(wines[wines[:,-1]==10])

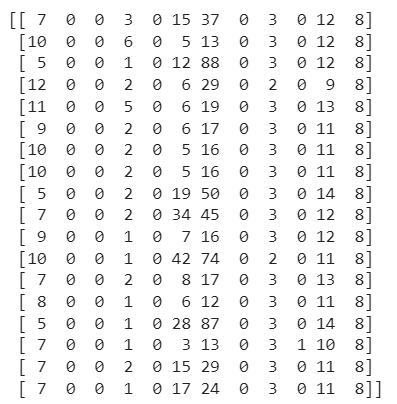
Output:



1. Select rows in wines where the quality is over 7

print(wines[wines[:,-1]>7])

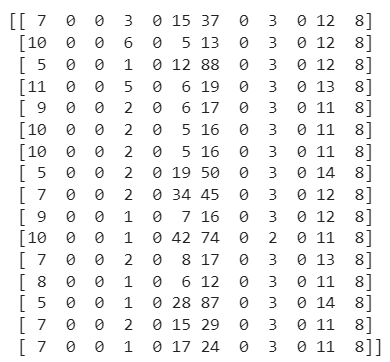
Output:



1. Display wines with alcohol greater than 10 and quality greater than 7.

print(wines[(wines[:,-2]>10) & (wines[:,-1]>7)])

Output:



1. Change the shape of wines array.

wines = wines.reshape((533,36))

print(wines.shape)

Output:



**Lab Cycle – II (Iris dataset)**

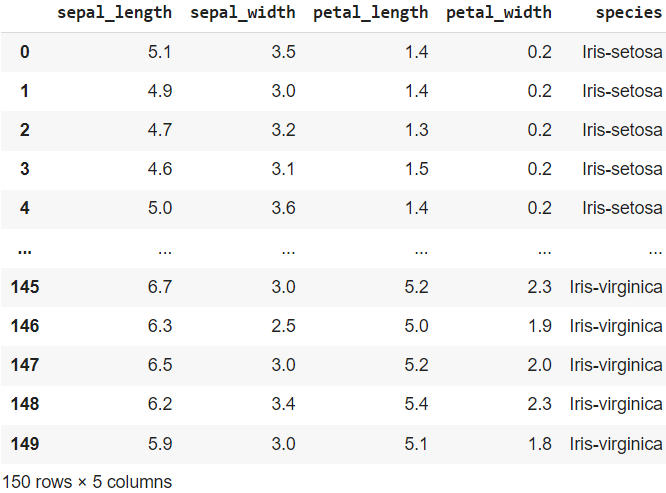
1. Print the dataset iris.

import pandas as pd

file = pd.read\_csv("/content/IRIS.csv")

file

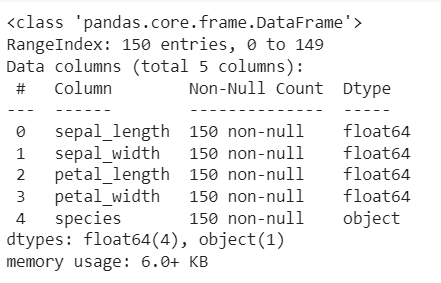
Output:



1. Print the structure of the dataset iris.

file.info()

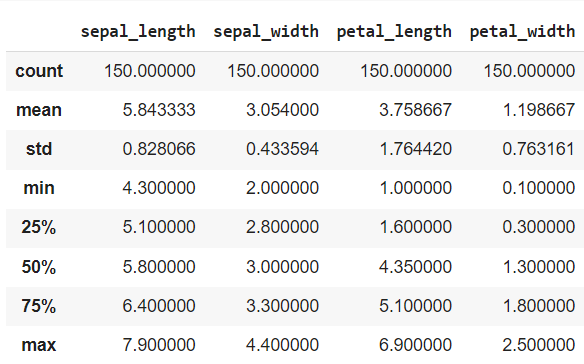
Output:



1. Print the summary of all the variables of the dataset iris.

file.describe()

Output:



1. How many of the variables (columns) are in the dataset iris.

len(file.keys())

Output:



1. How many observations (rows) are in the dataset iris.

len(file)

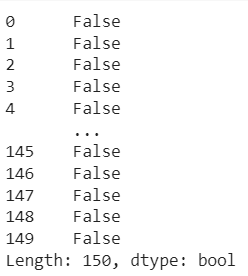
Output:



1. Use *duplicated()* function to print the logical vector indicating the duplicate values present in the dataset iris.

file.duplicated()

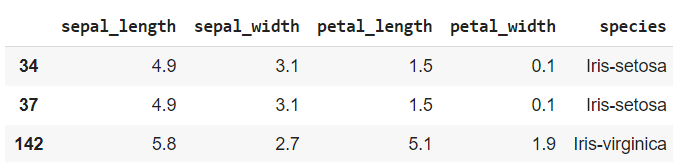
Output:



1. Extract duplicate elements from the dataset iris.

file[file.duplicated()]

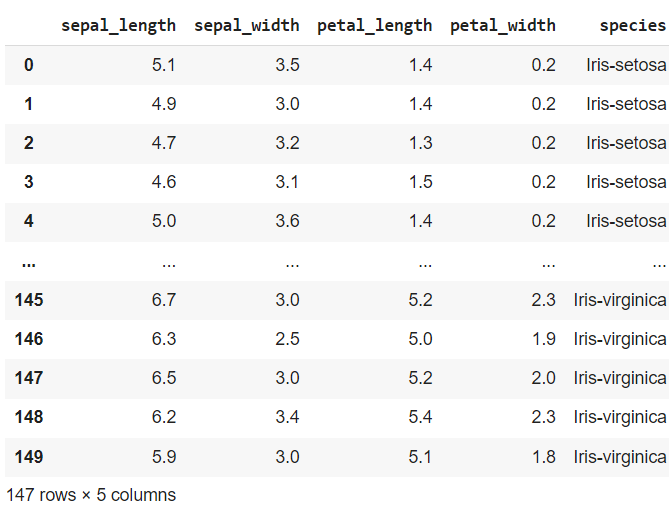
Output:



1. Extract unique elements from the dataset iris.

file.drop\_duplicates()

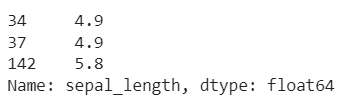
Output:



1. Print the indices of duplicate elements in the dataset iris.

file[file.duplicated()].index.tolist()

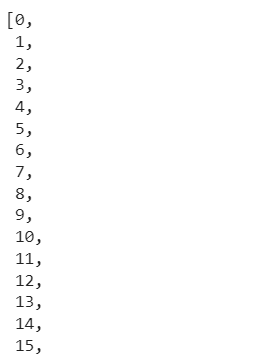
Output:



1. Print the indices of unique elements in the dataset iris.

file.drop\_duplicates().index.tolist()

Output:



1. How many unique elements are in the dataset iris.

len(file.drop\_duplicates())

Output:



1. How many duplicate elements are in the dataset iris.

len(file[file.duplicated()])

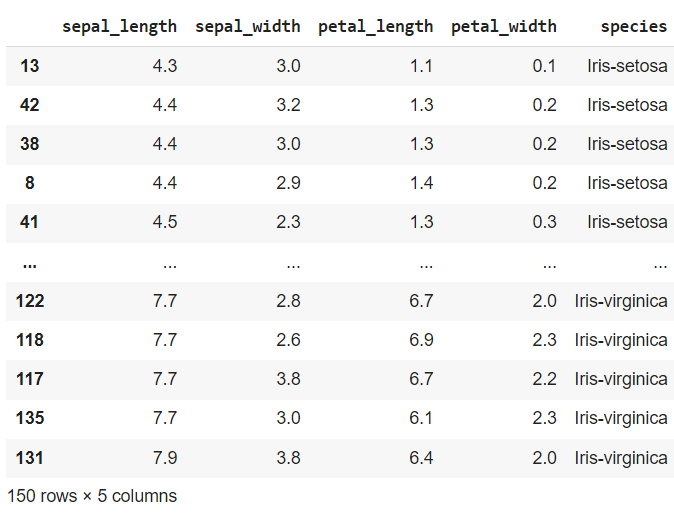
Output:



1. Print the sorted elements in the dataset iris(Ascending  order).

file.sort\_values('sepal\_length', axis=0)

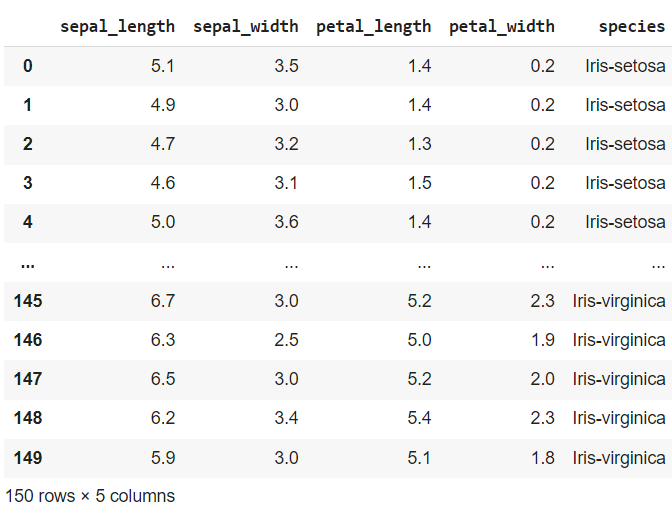
Output:



1. Find whether any missing values are in the dataset iris.

file.dropna()

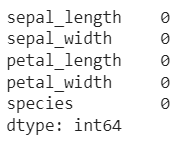
Output:



1. Display how many missing values are present in each column.

file.isnull().sum()

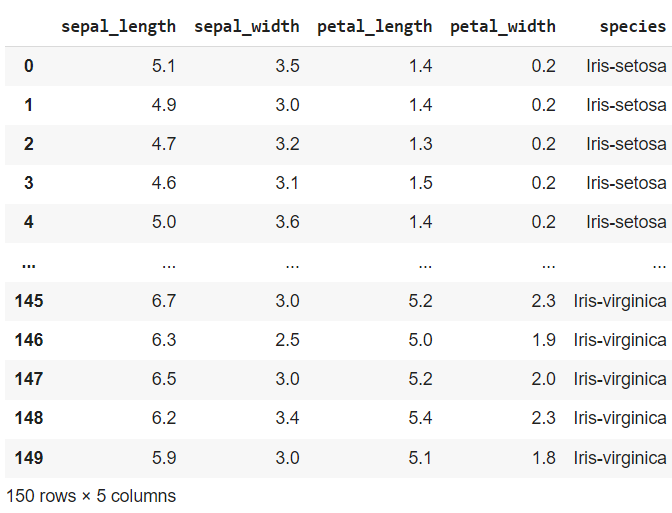
Output:



1. Replace all missing values with zero.

file.fillna(0)

Output:



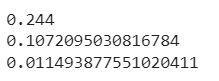
1. Calculate Petal width mean ,median ,SD,Variance  for the species setosa.

print(file[file['species']=='Iris-setosa']['petal\_width'].mean())

print(file[file['species']=='Iris-setosa']['petal\_width'].std())

print(file[file['species']=='Iris-setosa']['petal\_width'].var())

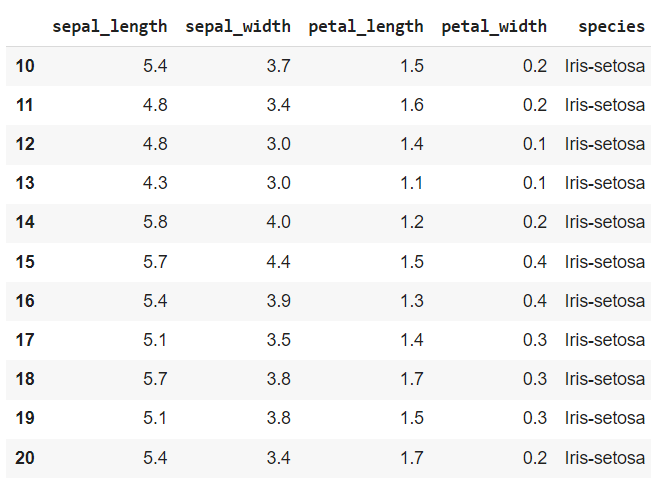
Output:



1. Print from 10th row to 20th row of iris dataset.

file[10:21]

Output:



1. Print Species and its corresponding Petal length and Width.

file[['species', 'petal\_length', 'petal\_width']]

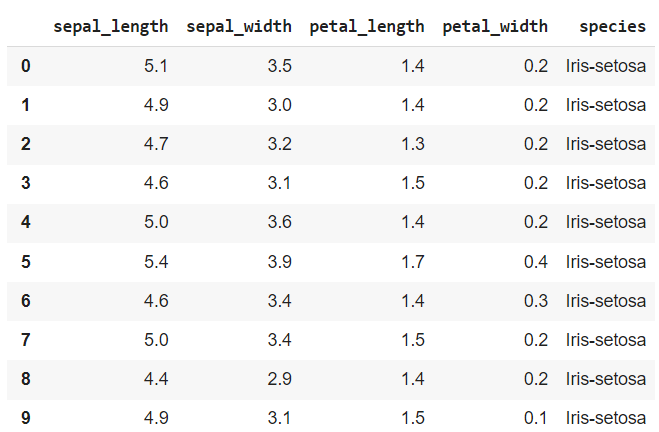
Output:



1. Display records only with species "Iris-setosa".

file[file['species']=="Iris-setosa"]

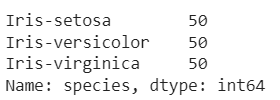
Output:



1. Count number of times a particular species has occurred.

file['species'].value\_counts()

Output:



1. Identifying minimum and maximum Value of Sepal width.

print(min(file['sepal\_width']))

print(max(file['sepal\_width']))

Output:



1. Add new column to store sum of  first four column values

file['Total']=file[file.columns[0:4]].sum(axis=1)

file

Output:



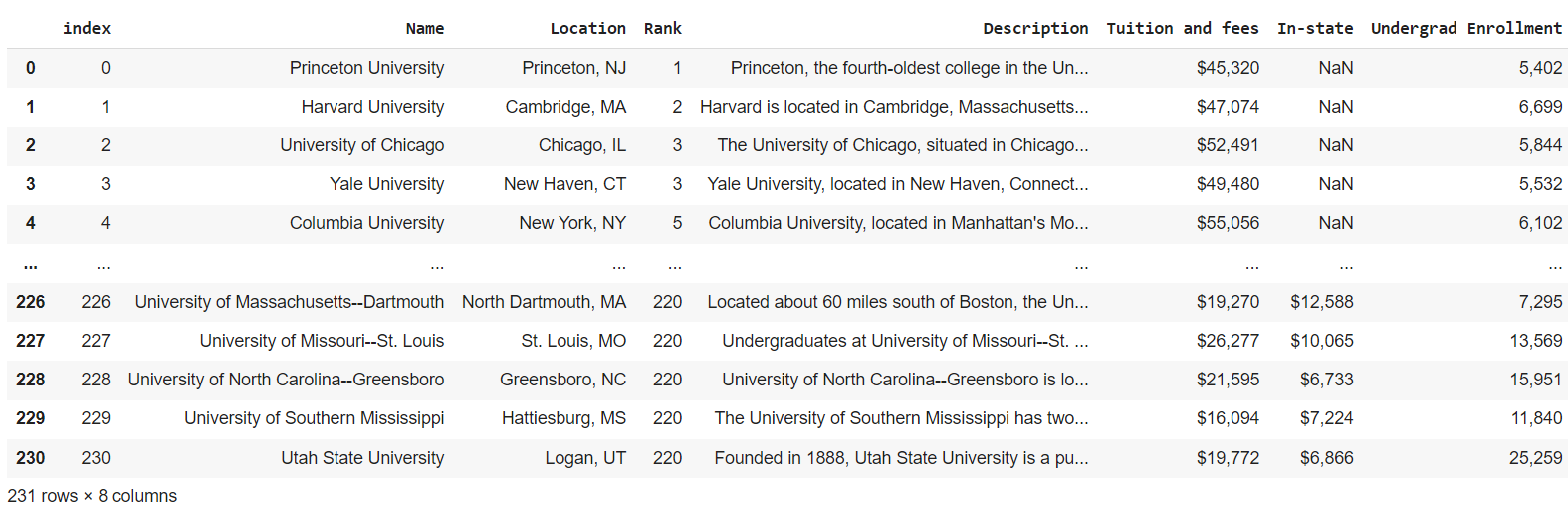
**Lab Cycle – III (National Universities Rankings dataset)**

import pandas as pd

file = pd.read\_csv("/content/National\_Universities\_Rankings.csv")

file

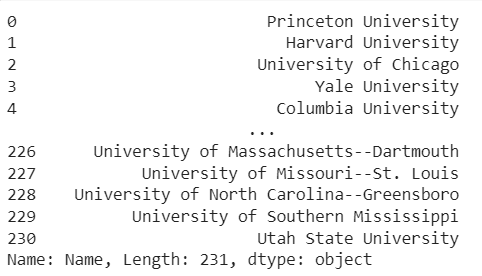
Output:



1. Find the universities in which undergraduate students were admitted.

file['Name'][file['Undergrad Enrollment'].notnull()]

Output:



1. List the states along with the cities in which universities located.

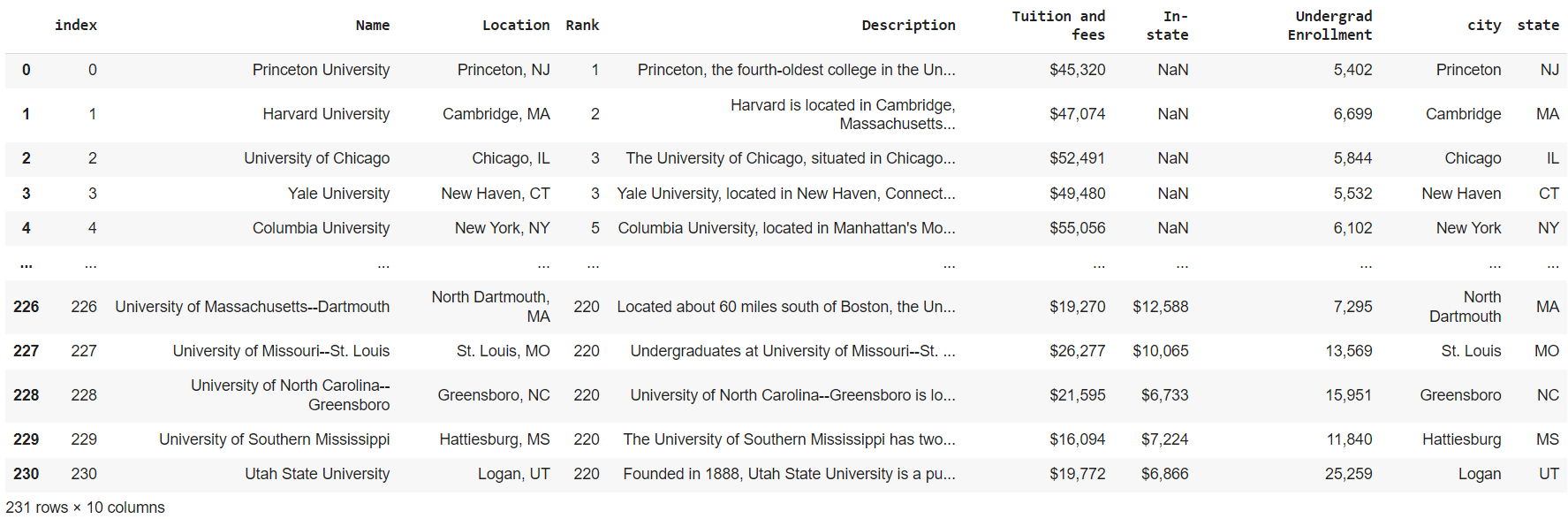
a = file['Location'].str.split(', ', expand=True)

file['city']=a[0]

file['state']=a[1]

file

Output:



1. List the cities &universities under each state.

list(file.groupby(['state', 'city', 'Name']).groups.keys())

Output:



1. How many universities in each state have ranking<50? Print them along with ranking.

list(file[file['Rank']<50].groupby(['state','Rank','Name']).groups)

Output:



1. How many universities have both out-of state and in-state students?

len(file[(file['Tuition and fees'].notnull()) & (file['In-state'].notnull())])

Output:



1. How many universities have marginal difference <=$5000 in in-state &out-of state tuition fees.

fee\_Tu = file['Tuition and fees'].str.replace('\W', '' ,regex=True)

fee\_Tu = fee\_Tu.astype(int)

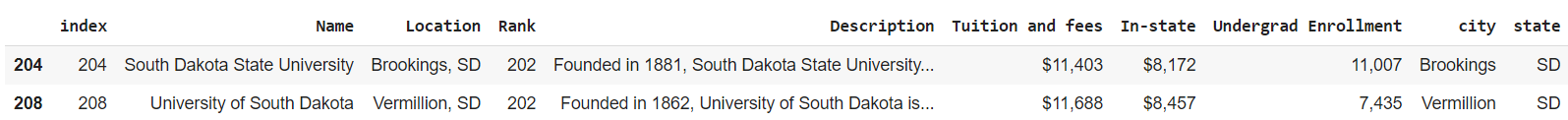
fee\_In = file['In-state'].fillna('0')

fee\_In = fee\_In.str.replace('\W', '' ,regex=True)

fee\_In = fee\_In.astype(int)

file[abs(fee\_Tu-fee\_In)<=5000]

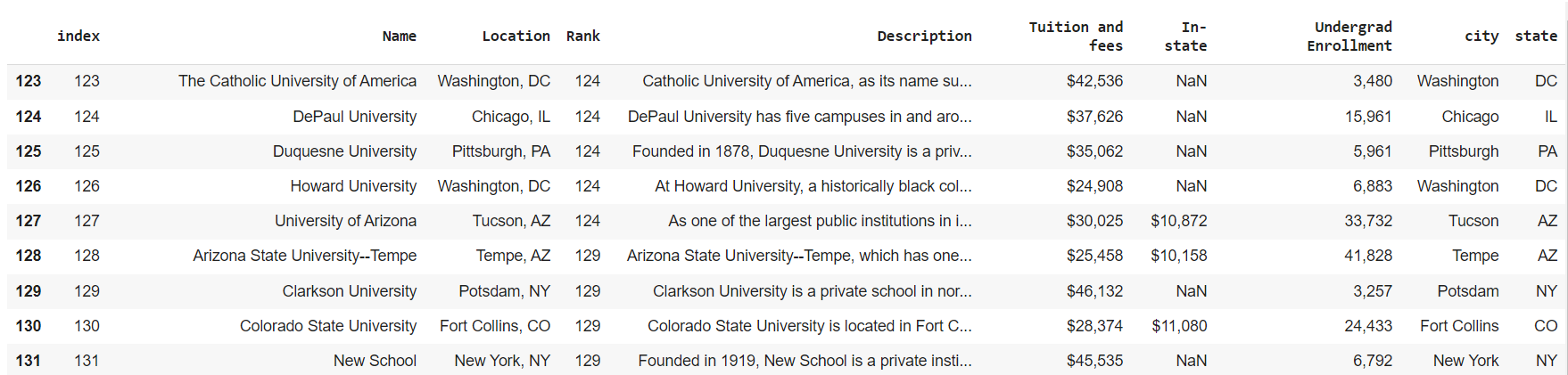
Output:



1. List the universities having tuition fee>15000$ and rank between 120 to 170.

file[(file['Tuition and fees'].str.replace('\W', '' ,regex=True).astype(int)>15000)&(file['Rank']>120)& (file['Rank']<170)]

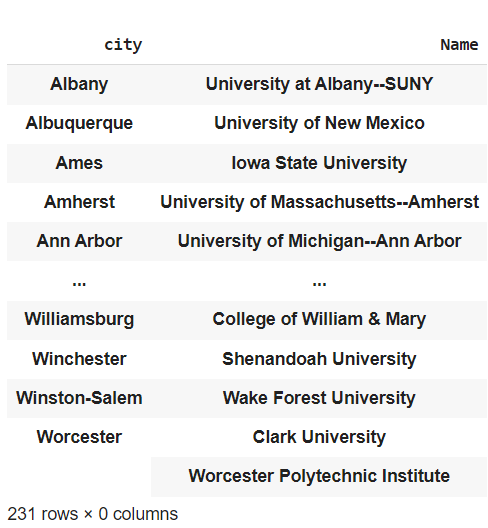
Output:



1. Find the campuses of universities located in different cities (multiple cities).

file[['city', 'Name']].groupby(['city', 'Name']).first()

Output:



1. Mention the states where out-of state fee is more than in-state students. Print Minimum and Maximum fees.

ft = file['Tuition and fees'].str.replace('\W', '',regex=True).astype(int)

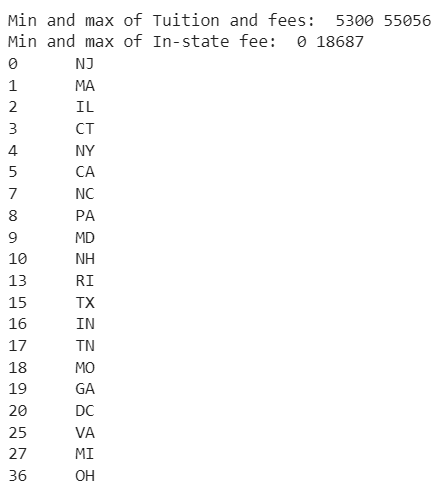
fi = file['In-state'].fillna('0').str.replace('\W', '' ,regex=True).astype(int)

print("Min and max of Tuition and fees: " ,min(ft), max(ft))

print("Min and max of In-state fee: ", min(fi), max(fi))

file[ft>fi].drop\_duplicates(['state'])['state']

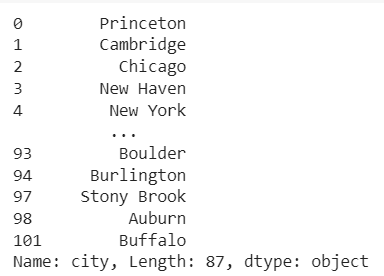
Output:



1. Find the cities locating top 100 universities.

file[file['Rank']<=100].drop\_duplicates(['city'])['city']

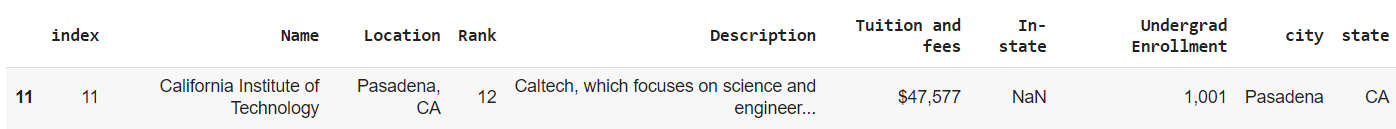
Output:



1. Find universities with least no of undergraduate students.

file[file['Undergrad Enrollment']==file['Undergrad Enrollment'].min()]

Output:



1. Identifying correlations between enrollment numbers and university rank.

file['Rank'].corr(file['Undergrad Enrollment'].apply(lambda x:int(x.replace(',', ""))))

Output:

