# PROGRAM NO: 4.1

**AIM :** Programs to handle data using pandas.

**DATE:** 2.09.2022

# SOURCE CODE :

1. import pandas as pd

orders = pd.read\_table('[http://bit.ly/movieusers')](http://bit.ly/movieusers%27)) print("Overview of dataframe : ") print(orders.head())

print("Shape : ",orders.shape) print()

user\_cols = ['user\_id', 'age', 'gender', 'occupation', 'zip\_code']

users = pd.read\_table('[http://bit.ly/movieusers',](http://bit.ly/movieusers%27) sep='|', header=None, names=user\_cols)

print("Dataframe after modifying the default parameter values for read\_table: ") print(users.head())

# OUTPUT :

1. import pandas as pd **#read a csv file**

ufo = pd.read\_csv('[http://bit.ly/uforeports')](http://bit.ly/uforeports%27)) print("Overview of UFO data reports: ") print(ufo.head())

print()

**#series**

**print("Cityseries(sorted):")** print(ufo.City.sort\_values()) print()

**ufo['Location'] = ufo.City + ', ' + ufo.State** print("After creating a new 'Location' Series : ") print(ufo.head())

print()

print("Calculate summary statistics : ") **print(ufo.describe())**

print()

print("Column names of ufo dataframe : ",ufo.columns) print()

# rename two of the columns by using the 'rename' method **ufo.rename(columns={'Colors Reported':'Colors\_Reported', 'Shape Reported':'Shape\_Reported'},inplace=True)**

print("Column name of ufo dataframe after renaming two column names : ",ufo.columns)

print()

**# remove multiple columns at once** ufo.drop(['City', 'State'], axis=1, inplace=True)

print("Column name of ufo dataframe after removing two columns(city,state) : ",ufo.columns)

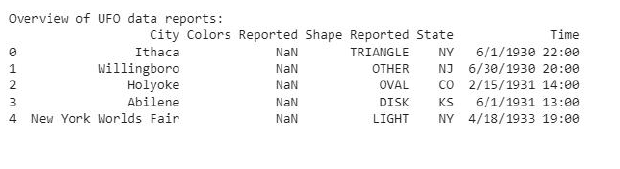
print()

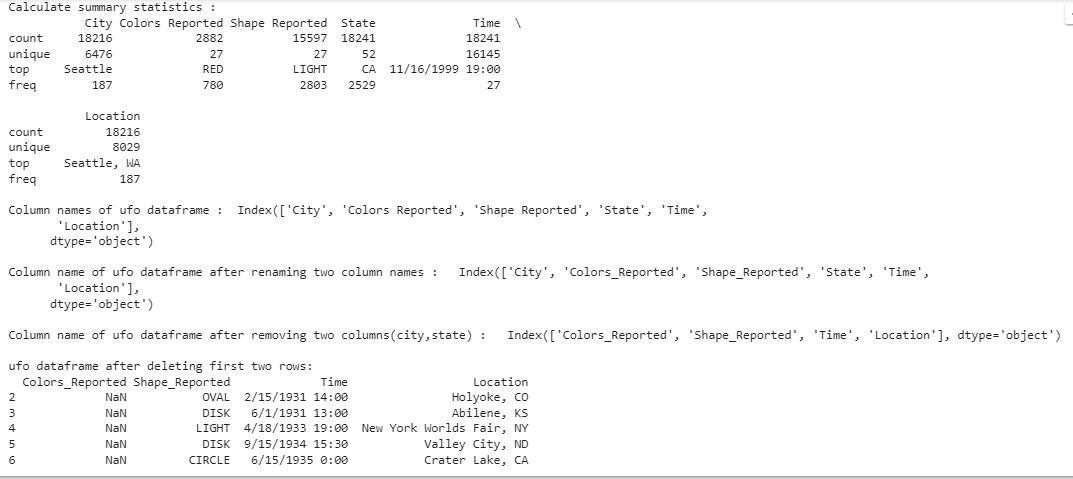
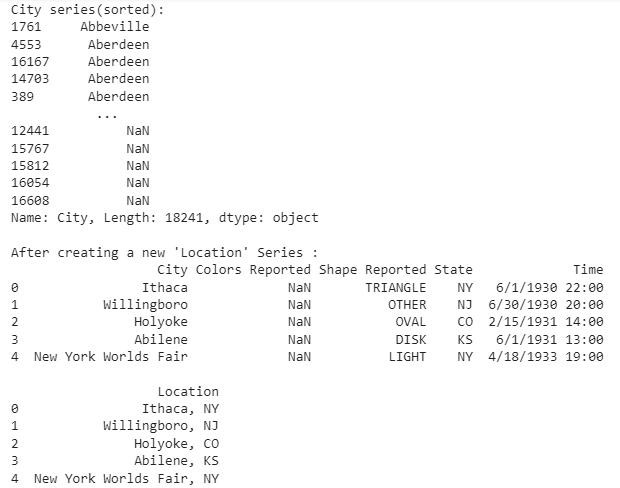
**# remove multiple rows at once (axis=0 refers to rows)**

ufo.drop([0, 1], axis=0, inplace=True)

print("ufo dataframe after deleting first two rows: ") print(ufo.head())

# OUTPUT :





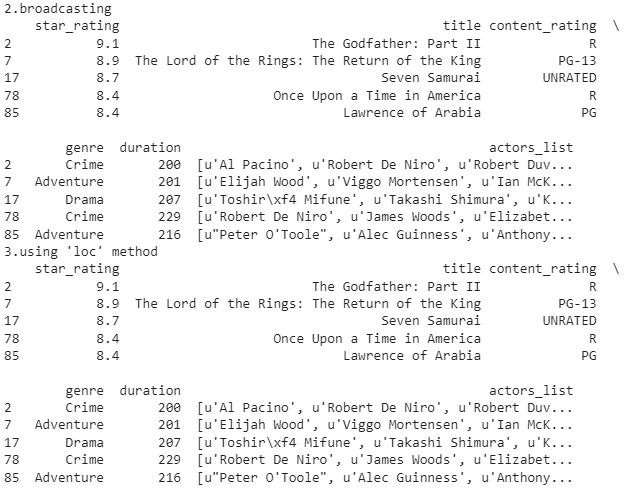
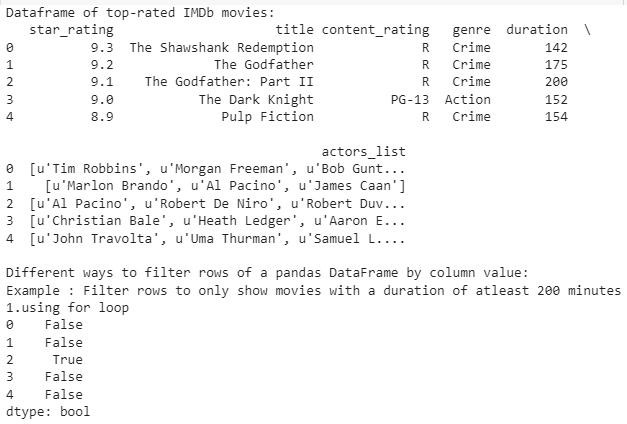
1. import pandas as pd

# read a dataset of top-rated IMDb movies into a DataFrame movies = pd.read\_csv('[http://bit.ly/imdbratings')](http://bit.ly/imdbratings%27)) print("Dataframe of top-rated IMDb movies: ") print(movies.head())

print()

print("**Different ways to filter rows of a pandas DataFrame by column value**: ") print("Example : Filter rows to only show movies with a duration of atleast 200 minutes") print("1.using for loop")

**# create a list in which each element refers to a DataFrame row: True if the row satisfies the condition,False otherwise**



booleans = []

for length in movies.duration:

if length >= 200:

**booleans.append(True)**

else:

**booleans.append(False)**

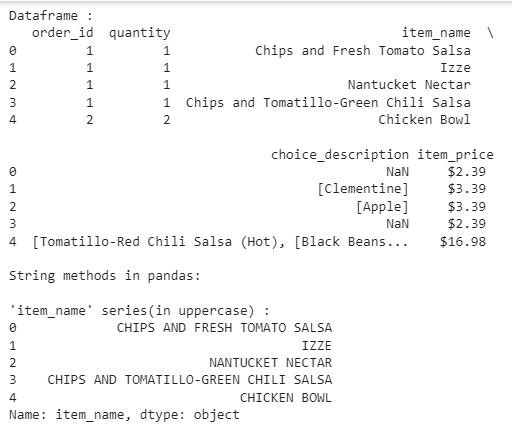
is\_long = pd.Series(booleans) print(is\_long.head())

print() print("2.broadcasting")

print(movies[movies.duration >= 200]) print()

print("3.using 'loc' method") print(movies.loc[movies.duration >= 200])

# OUTPUT :



1. import pandas as pd

# read a dataset of Chipotle orders into a DataFrame orders = pd.read\_table('[http://bit.ly/chiporders')](http://bit.ly/chiporders%27)) print("Dataframe : ")

print(orders.head()) print()

**print("String methods in pandas: ")** print()

print("'item\_name' series(in uppercase) : ") print(orders.item\_name.**str.upper().**head()) print()

print("Checks for a substring 'Chicken' in the given dataframe: ") print(orders[orders.item\_name.**str.contains('Chicken'**)].head()) print()

# many pandas string methods support regular expressions (regex) print(orders.choice\_description.**str.replace('[\[\]]', '').**head()) print()

print("Examine the data type of each Series: ") print(**orders.dtypes)**

print()

print("Dataframe after replacing '$' and converting string to float of 'item\_price' series: ") print(orders.item\_price.**str.replace('$', '').astype(float))**

print()

# OUTPUT :

