EXP NO:2 Write a program to import and export different categories of data using relevant data structures.

AIM:

To import and export different categories of data using relevant data structures.

DESCRIPTION:

CSV

CSV stands for Comma-separated values. Each line in a CSV file represents a data record, and each record has one or more data fields, which are separated by commas.

XLSX

The XLSX file is a spreadsheet in the Open XML Format of Microsoft Excel. This is used to store any kind of data, although it's primarily used to develop mathematical models and store financial data, among other things.

JSON

JSON stands for JavaScript Object Notation. It is a lightweight format for storing and transporting data. It is often used when data is sent from a server to a web page. It is "self-describing" and easy to understand

HDF

The Hierarchical Data Format (HDF) is a data model, file format and I/O library designed for storing, exchanging, managing and archiving complex data including scientific, engineering, and remote sensing data. The latest version of HDF, HDF5 allows users to read only the data that they need, not the whole file.

CODE:

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

READ CSV FILE using pandas

In [2]:
```
df=pd.read_csv('rainfall in india 1901-2015.csv',index_col=False)
df.head
#read file using pandas read_csv method
```

Out[2]:
```
<bound method NDFrame.head of                     SUBDIVISION  YEAR   JAN    FEB   MA
R    APR    MAY    JUN  \
0     ANDAMAN & NICOBAR ISLANDS  1901  49.2   87.1  29.2    2.3  528.8  517.5
1     ANDAMAN & NICOBAR ISLANDS  1902   0.0  159.8  12.2    0.0  446.1  537.1
2     ANDAMAN & NICOBAR ISLANDS  1903  12.7  144.0   0.0    1.0  235.1  479.9
3     ANDAMAN & NICOBAR ISLANDS  1904   9.4   14.7   0.0  202.4  304.5  495.1
4     ANDAMAN & NICOBAR ISLANDS  1905   1.3    0.0   3.3   26.9  279.5  628.7
...                         ...   ...   ...    ...   ...    ...    ...    ...
4111                LAKSHADWEEP  2011   5.1    2.8   3.1   85.9  107.2  153.6
4112                LAKSHADWEEP  2012  19.2    0.1   1.6   76.8   21.2  327.0
4113                LAKSHADWEEP  2013  26.2   34.4  37.5    5.3   88.3  426.2
4114                LAKSHADWEEP  2014  53.2   16.1   4.4   14.9   57.4  244.1
4115                LAKSHADWEEP  2015   2.2    0.5   3.7   87.1  133.1  296.6

        JUL    AUG    SEP    OCT    NOV    DEC  ANNUAL  Jan-Feb  Mar-May  \
0     365.1  481.1  332.6  388.5  558.2   33.6  3373.2    136.3    560.3
1     228.9  753.7  666.2  197.2  359.0  160.5  3520.7    159.8    458.3
2     728.4  326.7  339.0  181.2  284.4  225.0  2957.4    156.7    236.1
3     502.0  160.1  820.4  222.2  308.7   40.1  3079.6     24.1    506.9
4     368.7  330.5  297.0  260.7   25.4  344.7  2566.7      1.3    309.7
...     ...    ...    ...    ...    ...    ...     ...      ...      ...
4111  350.2  254.0  255.2  117.4  184.3   14.9  1533.7      7.9    196.2
4112  231.5  381.2  179.8  145.9   12.4    8.8  1405.5     19.3     99.6
4113  296.4  154.4  180.0   72.8   78.1   26.7  1426.3     60.6    131.1
4114  116.1  466.1  132.2  169.2   59.0   62.3  1395.0     69.3     76.7
4115  257.5  146.4  160.4  165.4  231.0  159.0  1642.9      2.7    223.9

      Jun-Sep  Oct-Dec
0      1696.3    980.3
1      2185.9    716.7
2      1874.0    690.6
3      1977.6    571.0
4      1624.9    630.8
...       ...      ...
4111   1013.0    316.6
4112   1119.5    167.1
4113   1057.0    177.6
4114    958.5    290.5
4115    860.9    555.4

[4116 rows x 19 columns]>
```

CHECK FOR NULL VALUES

In [3]:
```
df.info()

"""
```

```
The info() method prints information about the DataFrame.
The information contains the number of columns, column labels, column data types, memc
range index, and the number of cells in each column (non-null values

"""
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4116 entries, 0 to 4115
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   SUBDIVISION  4116 non-null   object 
 1   YEAR         4116 non-null   int64  
 2   JAN          4112 non-null   float64
 3   FEB          4113 non-null   float64
 4   MAR          4110 non-null   float64
 5   APR          4112 non-null   float64
 6   MAY          4113 non-null   float64
 7   JUN          4111 non-null   float64
 8   JUL          4109 non-null   float64
 9   AUG          4112 non-null   float64
 10  SEP          4110 non-null   float64
 11  OCT          4109 non-null   float64
 12  NOV          4105 non-null   float64
 13  DEC          4106 non-null   float64
 14  ANNUAL       4090 non-null   float64
 15  Jan-Feb      4110 non-null   float64
 16  Mar-May      4107 non-null   float64
 17  Jun-Sep      4106 non-null   float64
 18  Oct-Dec      4103 non-null   float64
dtypes: float64(17), int64(1), object(1)
memory usage: 611.1+ KB
```

Out[3]:
```
'\nThe info() method prints information about the DataFrame. \nThe information contai
ns the number of columns, column labels, column data types, memory usage, \nrange ind
ex, and the number of cells in each column (non-null values\n\n'
```

In [4]:
```
df.isnull().sum()  #find the null values in each column
```

Out[4]:
```
SUBDIVISION     0
YEAR            0
JAN             4
FEB             3
MAR             6
APR             4
MAY             3
JUN             5
JUL             7
AUG             4
SEP             6
OCT             7
NOV            11
DEC            10
ANNUAL         26
Jan-Feb         6
Mar-May         9
Jun-Sep        10
Oct-Dec        13
dtype: int64
```

In [5]:
```
from sklearn.impute import SimpleImputer
# Specify the column to exclude (string type)

column_to_exclude = 'SUBDIVISION'

# Create a list of numerical columns
numerical_columns = [col for col in df.columns if col != column_to_exclude and df[col]

# Create the imputer, excluding the string column
imputer = SimpleImputer(strategy='mean', copy=False)  # avoid unnecessary data copy
imputer.fit(df[numerical_columns])

# Transform the data, excluding the string column
df[numerical_columns] = imputer.transform(df[numerical_columns])

print(df)
```

```
                  SUBDIVISION    YEAR   JAN    FEB   MAR    APR    MAY  \
0     ANDAMAN & NICOBAR ISLANDS  1901.0  49.2   87.1  29.2    2.3  528.8
1     ANDAMAN & NICOBAR ISLANDS  1902.0   0.0  159.8  12.2    0.0  446.1
2     ANDAMAN & NICOBAR ISLANDS  1903.0  12.7  144.0   0.0    1.0  235.1
3     ANDAMAN & NICOBAR ISLANDS  1904.0   9.4   14.7   0.0  202.4  304.5
4     ANDAMAN & NICOBAR ISLANDS  1905.0   1.3    0.0   3.3   26.9  279.5
...                         ...     ...   ...    ...   ...    ...    ...
4111                LAKSHADWEEP  2011.0   5.1    2.8   3.1   85.9  107.2
4112                LAKSHADWEEP  2012.0  19.2    0.1   1.6   76.8   21.2
4113                LAKSHADWEEP  2013.0  26.2   34.4  37.5    5.3   88.3
4114                LAKSHADWEEP  2014.0  53.2   16.1   4.4   14.9   57.4
4115                LAKSHADWEEP  2015.0   2.2    0.5   3.7   87.1  133.1

        JUN    JUL    AUG    SEP    OCT    NOV    DEC  ANNUAL  Jan-Feb  \
0     517.5  365.1  481.1  332.6  388.5  558.2   33.6  3373.2    136.3
1     537.1  228.9  753.7  666.2  197.2  359.0  160.5  3520.7    159.8
2     479.9  728.4  326.7  339.0  181.2  284.4  225.0  2957.4    156.7
3     495.1  502.0  160.1  820.4  222.2  308.7   40.1  3079.6     24.1
4     628.7  368.7  330.5  297.0  260.7   25.4  344.7  2566.7      1.3
...     ...    ...    ...    ...    ...    ...    ...     ...      ...
4111  153.6  350.2  254.0  255.2  117.4  184.3   14.9  1533.7      7.9
4112  327.0  231.5  381.2  179.8  145.9   12.4    8.8  1405.5     19.3
4113  426.2  296.4  154.4  180.0   72.8   78.1   26.7  1426.3     60.6
4114  244.1  116.1  466.1  132.2  169.2   59.0   62.3  1395.0     69.3
4115  296.6  257.5  146.4  160.4  165.4  231.0  159.0  1642.9      2.7

      Mar-May  Jun-Sep  Oct-Dec
0       560.3   1696.3    980.3
1       458.3   2185.9    716.7
2       236.1   1874.0    690.6
3       506.9   1977.6    571.0
4       309.7   1624.9    630.8
...       ...      ...      ...
4111    196.2   1013.0    316.6
4112     99.6   1119.5    167.1
4113    131.1   1057.0    177.6
4114     76.7    958.5    290.5
4115    223.9    860.9    555.4

[4116 rows x 19 columns]
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
SUBDIVISION    0
YEAR           0
JAN            0
FEB            0
MAR            0
APR            0
MAY            0
JUN            0
JUL            0
AUG            0
SEP            0
OCT            0
NOV            0
DEC            0
ANNUAL         0
Jan-Feb        0
Mar-May        0
Jun-Sep        0
Oct-Dec        0
dtype: int64
```

EXPORT THE PROCESSED DATAFRAME TO OTHER FORMATS - CSV,JSON,EXCEL,HDF

In [7]:
```
df.to_excel('./ProcessedDataset/ExportEXCEL.xlsx',index=False)
df.head

#dataframe to excel format
```

Out[7]:
```
<bound method NDFrame.head of                    SUBDIVISION    YEAR   JAN    FEB
MAR    APR    MAY  \
0     ANDAMAN & NICOBAR ISLANDS  1901.0  49.2   87.1  29.2    2.3  528.8
1     ANDAMAN & NICOBAR ISLANDS  1902.0   0.0  159.8  12.2    0.0  446.1
2     ANDAMAN & NICOBAR ISLANDS  1903.0  12.7  144.0   0.0    1.0  235.1
3     ANDAMAN & NICOBAR ISLANDS  1904.0   9.4   14.7   0.0  202.4  304.5
4     ANDAMAN & NICOBAR ISLANDS  1905.0   1.3    0.0   3.3   26.9  279.5
...                         ...     ...   ...    ...   ...    ...    ...
4111                LAKSHADWEEP  2011.0   5.1    2.8   3.1   85.9  107.2
4112                LAKSHADWEEP  2012.0  19.2    0.1   1.6   76.8   21.2
4113                LAKSHADWEEP  2013.0  26.2   34.4  37.5    5.3   88.3
4114                LAKSHADWEEP  2014.0  53.2   16.1   4.4   14.9   57.4
4115                LAKSHADWEEP  2015.0   2.2    0.5   3.7   87.1  133.1

        JUN    JUL    AUG    SEP    OCT    NOV    DEC  ANNUAL  Jan-Feb  \
0     517.5  365.1  481.1  332.6  388.5  558.2   33.6  3373.2    136.3
1     537.1  228.9  753.7  666.2  197.2  359.0  160.5  3520.7    159.8
2     479.9  728.4  326.7  339.0  181.2  284.4  225.0  2957.4    156.7
3     495.1  502.0  160.1  820.4  222.2  308.7   40.1  3079.6     24.1
4     628.7  368.7  330.5  297.0  260.7   25.4  344.7  2566.7      1.3
...     ...    ...    ...    ...    ...    ...    ...     ...      ...
4111  153.6  350.2  254.0  255.2  117.4  184.3   14.9  1533.7      7.9
4112  327.0  231.5  381.2  179.8  145.9   12.4    8.8  1405.5     19.3
4113  426.2  296.4  154.4  180.0   72.8   78.1   26.7  1426.3     60.6
4114  244.1  116.1  466.1  132.2  169.2   59.0   62.3  1395.0     69.3
4115  296.6  257.5  146.4  160.4  165.4  231.0  159.0  1642.9      2.7

      Mar-May  Jun-Sep  Oct-Dec
0       560.3   1696.3    980.3
1       458.3   2185.9    716.7
2       236.1   1874.0    690.6
3       506.9   1977.6    571.0
4       309.7   1624.9    630.8
...       ...      ...      ...
4111    196.2   1013.0    316.6
4112     99.6   1119.5    167.1
4113    131.1   1057.0    177.6
4114     76.7    958.5    290.5
4115    223.9    860.9    555.4

[4116 rows x 19 columns]>
```

In [8]:
```
df.to_json('./ProcessedDataset/ExportJSON.json')
#dataframe to json format
```

In [9]:
```
df.to_csv('./ProcessedDataset/ExportCSV.csv',index=False)

#dataframe to csv format
```

In [10]:
```
df.to_hdf('./ProcessedDataset/ExportHDF.h5',key='df',mode='a')

#dataframe to hdf format
```

In [11]:
```
df=pd.read_csv('./ProcessedDataset/ExportCSV.csv')
df.head(5)

#reading processed csv
```

Out[11]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DE
0	ANDAMAN & NICOBAR ISLANDS	1901.0	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33
1	ANDAMAN & NICOBAR ISLANDS	1902.0	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160
2	ANDAMAN & NICOBAR ISLANDS	1903.0	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225
3	ANDAMAN & NICOBAR ISLANDS	1904.0	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40
4	ANDAMAN & NICOBAR ISLANDS	1905.0	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344

In [12]:
```
df=pd.read_excel('./ProcessedDataset/ExportEXCEL.xlsx')
df.head(5)
```

Out[12]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DE(
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.(
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.(
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.1

In [13]:
```
df=pd.read_hdf('./ProcessedDataset/ExportHDF.h5')
df.head(5)
```

Out[13]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DE
0	ANDAMAN & NICOBAR ISLANDS	1901.0	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33
1	ANDAMAN & NICOBAR ISLANDS	1902.0	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160
2	ANDAMAN & NICOBAR ISLANDS	1903.0	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225
3	ANDAMAN & NICOBAR ISLANDS	1904.0	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40
4	ANDAMAN & NICOBAR ISLANDS	1905.0	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344

In [14]:
```
df=pd.read_json('./ProcessedDataset/ExportJSON.json')
df.head(5)
```

Out[14]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DE(
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.(
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.(
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.1

OUTPUT:

PROCESSED DATSETS IN VARIOUS FORMATS