import pandas as pd

import numpy as np

#Convert dictionary to DataFrame

Df = pd.DataFrame(dictionary1)

#Write to csv file

Df.to\_csv(‘test1.csv’)

#To remove Index

Df.to\_csv(‘test1.csv’,index=False)

#To show first n rows

Df.head(n)

#To show last n rows

Df.tail(n)

#To statistical analysis of numerical Columns

Df.describe()

#To read from CSV files

Df2 = pd.read\_csv(‘test2.csv)

#To get a particular column (col2)

Df2[‘col2’]

#To get a particular value of a particular column(col2) and particular row(1)

Df2[‘col2’][1]

# To change a particular value to 100 of a particular column(col2) and particular row(1)

Df2[‘col2’][1] = 100 (But with warning)

#Change the Index value

Df2.index = [‘1st’, ’2nd’,’3rd’,’4th’]

# 2 types of data structure i.e. series and DataFrame

#Series – 1D array with index. It stores a single column or row of data in DataFrame

#DataFrame – Collections of Series (2D)

#To create a random series (Column) of 35 numbers

Ser = pd.Series(np.random.rand(35))

#To create a DataFrame

Df3 = pd.DataFrame(np.random.rand(334,5),index = np.arange(334))

#To get the datatype of a DataFrame

Df.3.dtype()

#To change a value of 2nd column , 1st row (Index start from 0)

DF3[1][2] = ‘Hello’

#To get the Index

Df3.index

#To convert to NumPy array

Df3.to\_numpy()

#To transpose

Df3.T

#To sort with respect to index in descending order

Df3.sort\_index(axis=1 “””(for column)”””,ascending = False) # axis = 0 for rows

#To sort by a column

Df3.sort\_values(by=’B’)

#To get 0th column

Df3[0]

# To copy a DataFrame

Df4 = Df3.copy()

#To set a new value

Df3.loc[0,0] = 235 #row, column

#To set a column name

Df3.columns(list(“ABCDE”)

#To set a new value

Df3.loc[0, ’A’] = 354

#To Drop a column

Df3.drop(column\_name, axis=1 “””For column”””, inplace = True “””To change the original Df3 DataFrame”””)

#To Drop a columns

Df3.drop([column1\_name, column2\_name], axis=1 “””For column”””, inplace = True “””To change the original Df3 DataFrame”””)

#To Drop a row

Df3.drop(row\_name, axis = 0 “””For Row”””, inplace = True “””To change the original Df3 DataFrame”””)

#To Drop a rows

Df3.drop([row1\_name, row2\_name], axis=0 “””For Row”””, inplace = True “””To change the original Df3 DataFrame”””)

# To get a range of rows and columns

Df3.loc[[1,2] “””Row 1 and 2”””, [‘C’,’D’] “””Column C and D”””] #Df3.at[1,2]

#To get all rows of column C , D

Df3.loc[:,[‘C’,’D’]]

#To get all columns of Row 1,2

Df3.loc[[1,2],:]

#To get all data where A column having value 0.3

Df3.loc[(Df3[‘A’] < 0.3)]

#To get all data where A column having value < 0.3 and Column C > 0.2

Df3.loc[(Df3[‘A’] < 0.3) & (Df3[‘C’] > 0.i)]

#To get values with row and column index

Df3.iloc[1,2]

Df3iloc[[1,2],[3,4]]

#loc[] for row name and column name, iloc for row index and column index staring from 0

#To reset indexing

Df3.reset\_index(drop=True, inplace=True)

Using the [isin()](https://pandas.pydata.org/docs/reference/api/pandas.Series.isin.html#pandas.Series.isin) method for filtering:

df2 = df.copy()

df2["E"] = ["one", "one", "two", "three", "four", "three"]

df2

Out[43]:

A B C D E

2013-01-01 0.469112 -0.282863 -1.509059 -1.135632 one

2013-01-02 1.212112 -0.173215 0.119209 -1.044236 one

2013-01-03 -0.861849 -2.104569 -0.494929 1.071804 two

2013-01-04 0.721555 -0.706771 -1.039575 0.271860 three

2013-01-05 -0.424972 0.567020 0.276232 -1.087401 four

2013-01-06 -0.673690 0.113648 -1.478427 0.524988 three

df2[df2["E"].isin(["two", "four"])]

Out[44]:

A B C D E

2013-01-03 -0.861849 -2.104569 -0.494929 1.071804 two

2013-01-05 -0.424972 0.567020 0.276232 -1.087401 four