PRACTICAL-7

Perform following operations on a CSV file a. Create a data frame from csv file, dictionary, List of tuples
b. Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data frame
d. Finding maximum and minimum values
e. Displaying statistical information
f. Performing queries
g. Data Analysis using groupby()
h: Handling missing data(NA, Missing,Null)
Use Kaggle Datasets

CODE:

```
import pandas as pd

# creating a data frame
df = pd.read_csv("CardioGoodFitness.csv")
print("DataFrame\n" + str(df.head()))

# dictionary
import csv

filename = "CardioGoodFitness.csv"
print("Dictionary")
with open(filename, 'r') as data:
    for line in csv.reader(data):
        print(line)

# List of tuples
import pandas as pd

df = pd.read_csv('CardioGoodFitness.csv', delimiter=',')
list_of_tuples = [tuple(row) for row in df.values]

print("List of tupple\n" + str(list_of_tuples))

# Operations on Data Frame Shape, head, tail c. Retrieving rows / columns from data framedf =
pd.read_csv("CardioGoodFitness.csv")
import pandas as pd

# creating a data frame
df = pd.read_csv("CardioGoodFitness.csv")
shape = df.shape
print("Use of Shape" + str(shape))

# head
import pandas as pd
```

```
df = pd.read_csv("CardioGoodFitness.csv")
print("DataFrame using head\n" + str(df.head()))

# tail
import pandas as pd

data = pd.read_csv("CardioGoodFitness.csv")
print("Use of Tail\n" + str(data.tail()))

# fetch raw/column
# column
import pandas as pd

df = pd.read_csv("CardioGoodFitness.csv")
column = df[["Age", "Gender", "Fitness"]]
print("Show column\n" + str(column))
# row
import numpy as np
import pandas as pd

data = pd.read_csv("CardioGoodFitness.csv", index_col="Gender")
result = data.loc["Male"]
print("Show row\n" + str(result))

# Finding maximum and minimum values
import numpy as np
import pandas as pd

df = pd.read_csv("CardioGoodFitness.csv")
maxValues = df.max()
minValues = df.min()
print("Max Values\n" + str(maxValues))
print("Min Values\n" + str(minValues))

# Displaying statistical information.
import pandas as pd

data = pd.read_csv("CardioGoodFitness.csv")
stats_numeric = data['Income'].describe()
print("Statestical information on Income\n" + str(stats_numeric))

# Run a query
import pandas as pd

data = pd.read_csv("CardioGoodFitness.csv")
result = data.query('Fitness > 3')
```

```
result1 = data.query('MaritalStatus == "Single"')
print("Print data where fitness is greater than 3\n" + str(result))
print("Print data where marital status is single\n" + str(result1))
# Data Analysis using groupby()
import pandas as pd

df = pd.read_csv("CardioGoodFitness.csv")
gb = df.groupby('Age')
print("Groupby" + str(gb))

# Handling missing data(NA, Missing,Null  )

import pandas as pd

data = pd.read_csv("CardioGoodFitness.csv")
bool_series = pd.isnull(data["Usage"])
print("Missing data\n" + str(data[bool_series]))
```

OUTPUT:

```
C:\Users\Admin\PycharmProjects\Python\venv\Scripts\python.exe C:/Users/Admin/PycharmProjects/Python/Practical7.py
DataFrame
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
0    P101   18    Male         14        Single      3        4   29562    112
1    P102   19    Male         15        Single      2        3   31836     75
2    P103   19  Female         14       Married      4        3   30699     66
3    P104   19    Male         12        Single      3        3   32973     85
4    P105   20    Male         13       Married      4        2   35247     47
Dictionary
['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage', 'Fitness', 'Income', 'Miles']
['P101', '18', 'Male', '14', 'Single', '3', '4', '29562', '112']
['P102', '19', 'Male', '15', 'Single', '2', '3', '31836', '75']
['P103', '19', 'Female', '14', 'Married', '4', '3', '30699', '66']
['P104', '19', 'Male', '12', 'Single', '3', '3', '32973', '85']
['P105', '20', 'Male', '13', 'Married', '4', '2', '35247', '47']
List of tupple
[('P101', 18, 'Male', 14, 'Single', 3, 4, 29562, 112), ('P102', 19, 'Male', 15, 'Single', 2, 3, 31836, 75), ('P103', 19, 'Female', 14, 'Married', 4, 3,
 30699, 66), ('P104', 19, 'Male', 12, 'Single', 3, 3, 32973, 85), ('P105', 20, 'Male', 13, 'Married', 4, 2, 35247, 47)]
Use of Shape(5, 9)
DataFrame using head
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
0    P101   18    Male         14        Single      3        4   29562    112
1    P102   19    Male         15        Single      2        3   31836     75
2    P103   19  Female         14       Married      4        3   30699     66
3    P104   19    Male         12        Single      3        3   32973     85
4    P105   20    Male         13       Married      4        2   35247     47
```

```
Use of Tail
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
0    P101   18    Male         14        Single      3        4   29562    112
1    P102   19    Male         15        Single      2        3   31836     75
2    P103   19  Female         14       Married      4        3   30699     66
3    P104   19    Male         12        Single      3        3   32973     85
4    P105   20    Male         13       Married      4        2   35247     47
Show column
   Age  Gender  Fitness
0   18    Male        4
1   19    Male        3
2   19  Female        3
3   19    Male        3
4   20    Male        2
Show row
         Product  Age  Education MaritalStatus  Usage  Fitness  Income  Miles
Gender
Male        P101   18         14        Single      3        4   29562    112
Male        P102   19         15        Single      2        3   31836     75
Male        P104   19         12        Single      3        3   32973     85
Male        P105   20         13       Married      4        2   35247     47
Max Values
Product    P105
Age          20
```

```
Product            P105
Age                  20
Gender             Male
Education            15
MaritalStatus    Single
Usage                 4
Fitness               4
Income            35247
Miles               112
dtype: object
Min Values
Product            P101
Age                  18
Gender           Female
Education            12
MaritalStatus    Married
Usage                 2
Fitness               2
Income            29562
Miles                47
dtype: object
Statestical information on Income
count        5.000000
mean     32063.400000
std       2187.063168
min      29562.000000
25%      30699.000000
50%      31836.000000
75%      32973.000000
max      35247.000000
Name: Income, dtype: float64
Print data where fitness is greater than 3
  Product  Age Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
0    P101   18   Male         14        Single      3        4   29562    112
Print data where marital status is single
  Product  Age Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
0    P101   18   Male         14        Single      3        4   29562    112
1    P102   19   Male         15        Single      2        3   31836     75
3    P104   19   Male         12        Single      3        3   32973     85
Groupby<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000216F7733D90>
Missing data
Empty DataFrame
Columns: [Product, Age, Gender, Education, MaritalStatus, Usage, Fitness, Income, Miles]
Index: []
```