Name: Sonali Shintre
Roll No: 29

Assignment: Exp 13 — Pandas DataFrames and CSV Operations

Question 1: Create a CSV file named employees.csv with columns Name, Age, Department, Salary. Load the data, filter employees older than 30, group by Department, and calculate the average, minimum, and maximum salary for each department.

Code:

```
import pandas as pd
import numpy as np data
= {
    'Name': [f'Emp{i}' for i in range(1,11)],
    'Age': np.random.randint(22,60,10),
    'Department': np.random.choice(['HR','Sales','IT','Finance'],10),
    'Salary': np.random.randint(30000,120001,10)
} df = pd.DataFrame(data) print('Employees DataFrame:')
print(df.to_string(index=False)) filtered = df[df['Age'] > 30]
print('\nFiltered (Age > 30):') print(filtered.to_string(index=False))
grp =
filtered.groupby('Department')['Salary'].agg(['mean','min','max']).rese
t_index() print('\nSalary stats by Department (for Age>30):')
print(grp.to_string(index=False))
```

Output:

```
Employees DataFrame:
 Name  Age Department  Salary
 Emp1   22          HR  112457
 Emp2   25          HR   36921
 Emp3   25          IT   68804
 Emp4   31       Sales   97699
 Emp5   41          IT  100608
 Emp6   43     Finance   67619
 Emp7   58     Finance   37877
 Emp8   45          IT  113966
 Emp9   28          HR   31871
Emp10   46       Sales  103135

Filtered (Age > 30):
 Name  Age Department  Salary
 Emp4   31       Sales   97699
 Emp5   41          IT  100608
 Emp6   43     Finance   67619
 Emp7   58    Finance   37877
 Emp8   45         IT  113966
Emp10   46      Sales  103135

Salary stats by Department (for Age>30):
Department     mean    min    max
   Finance  52748.0  37877  67619
        IT 107287.0 100608 113966
     Sales 100417.0  97699 103135
```

Question 2: Create a CSV file named sales.csv with columns SaleID, Product, Amount, Customer. Load the data, filter sales with amount greater than 50,000, group by Product, and calculate total and average sales per product.

Code:

```
import pandas as pd, numpy as np data
= {
    'SaleID': range(1,13),
    'Product': np.random.choice(['A','B','C','D'],12),
    'Amount': np.random.randint(10000,100001,12),
    'Customer': np.random.choice(['Cust1','Cust2','Cust3','Cust4'],12)
} df = pd.DataFrame(data) print('Sales DataFrame:')
print(df.to_string(index=False)) filtered = df[df['Amount'] > 50000]
print('\nFiltered (Amount > 50000):')
print(filtered.to_string(index=False)) grp =
filtered.groupby('Product')['Amount'].agg(['sum','mean']).reset_index()
print('\nTotal and Average sales per Product:')
print(grp.to_string(index=False))
```

Output:

```
Sales DataFrame:
 SaleID Product  Amount Customer
1        A   91857    Cust2
2        C   62489    Cust3
3        D   94665    Cust1
4        A   51504    Cust1
5        B   59866    Cust2
6        D   94212    Cust4
```

```
7        B   21723    Cust1
8        D   53890    Cust2
9        D   27591    Cust3
10       C   68146    Cust3
11       D   69300    Cust4
12       A   99141    Cust1

Filtered (Amount > 50000):
 SaleID Product  Amount Customer
1       A   91857    Cust2
2       C   62489    Cust3
3       D   94665    Cust1
4       A   51504    Cust1
5       B   59866    Cust2
6       D   94212    Cust4
      8        D   53890    Cust2
10       C   68146    Cust3
11       D   69300    Cust4
12       A   99141    Cust1

Total and Average sales per Product:
Product    sum     mean
A  242502 80834.00
B  59866 59866.00
C  130635 65317.50
D  312067 78016.75
```

Question 3: Create a CSV file named student_marks.csv with columns StudentID, Name, Maths, Science, English. Load the data, filter students with Maths marks above 80, group by student name, and calculate total and average marks per student.

Code:

```
import pandas as pd, numpy as np data
= {
    'StudentID': range(1,11),
    'Name': [f'Stud{i}' for i in range(1,11)],
    'Maths': np.random.randint(40,101,10),
    'Science': np.random.randint(35,101,10),
    'English': np.random.randint(30,101,10)
} df = pd.DataFrame(data) print('Student Marks:')
print(df.to_string(index=False)) filtered = df[df['Maths'] > 80]
print('\nFiltered (Maths > 80):')
print(filtered[['StudentID','Name','Maths']].to_string(index=False))
grp =
df.groupby('Name')[['Maths','Science','English']].agg(['sum','mean'])
print('\nTotal and Average marks per student:')
# flatten columns for printing grp2 = grp.copy() grp2.columns =
['_'.join(col).strip() for col in grp2.columns.values]
print(grp2.reset_index().to_string(index=False))
```

Output:

```
Student Marks:
 StudentID   Name  Maths  Science  English
1  Stud1     41       35       50
2  Stud2     41       49       88
3  Stud3     79       88       53
4  Stud4     81       47       43
5  Stud5     97       77       78
6  Stud6     75       41       79
7  Stud7     78       82       99
8  Stud8     95       38       71
9  Stud9     51       87       65
10 Stud10     86       50       94

Filtered (Maths > 80):
 StudentID   Name  Maths
4  Stud4     81
5  Stud5     97
        8  Stud8     95
       10 Stud10     86

Total and Average marks per student:
  Name  Maths_sum  Maths_mean  Science_sum  Science_mean  English_sum
English_mean
 Stud1         41        41.0           35          35.0           50
50.0 Stud10         86        86.0           50          50.0
94
94.0  Stud2         41        41.0           49          49.0
88
88.0  Stud3         79        79.0           88          88.0
53
53.0  Stud4         81        81.0           47          47.0
43
43.0
 Stud5         97        97.0           77          77.0           78
78.0  Stud6         75        75.0           41          41.0
79
```

```
79.0  Stud7         78        78.0            82           82.0
99
99.0  Stud8         95        95.0            38           38.0
71
71.0  Stud9         51        51.0            87           87.0
65
65.0
```

Question 4: Create a CSV file named products.csv with columns ProductID, ProductName, Category, Stock. Load the data, filter products with stock less than 50, group by Category, and calculate the total and average stock per category.

Code:

```
import pandas as pd, numpy as np data
= {
    'ProductID': range(1,13),
    'ProductName': [f'Prod{i}' for i in range(1,13)],
    'Category':
np.random.choice(['Electronics','Clothing','Grocery'],12),
    'Stock': np.random.randint(0,200,12)
} df = pd.DataFrame(data)
print('Products:')
print(df.to_string(index=False))
filtered = df[df['Stock'] < 50]
print('\nFiltered (Stock <
50):')
print(filtered.to_string(index=F
alse)) grp =
filtered.groupby('Category')['St
ock'].agg(['sum','mean']).reset_
index() print('\nTotal and
Average stock per Category (for
low-stock products):')
print(grp.to_string(index=False))
```

Output:

```
Products:
 ProductID ProductName    Category  Stock
```

```
1       Prod1    Clothing    112
2       Prod2     Grocery    149
3       Prod3 Electronics    127
4       Prod4    Clothing      0
5       Prod5     Grocery    138
6       Prod6 Electronics    114
7       Prod7     Grocery     43
8       Prod8 Electronics    186
9       Prod9    Clothing    127
10      Prod10     Grocery     23
11      Prod11     Grocery    187
12      Prod12    Clothing    130

Filtered (Stock < 50):
 ProductID ProductName Category  Stock
         4       Prod4 Clothing      0
         7       Prod7  Grocery     43
        10      Prod10  Grocery     23

Total and Average stock per Category (for low-stock products):
Category  sum  mean
Clothing    0   0.0
 Grocery   66  33.0
```

Question 5: Create a CSV file named transactions.csv with columns TransactionID, Customer, Amount, Date. Load the data, filter transactions with amount greater than 1000, group by Customer, and calculate total and average spending per customer.

Code:

```
import pandas as pd, numpy as np data
= {
    'TransactionID': range(1,13),
    'Customer': np.random.choice(['C1','C2','C3','C4'],12),
    'Amount': np.random.randint(100,5000,12),
    'Date': pd.date_range(end=pd.Timestamp('2025-11-02'),
periods=12).strftime('%Y-%m-%d')
} df = pd.DataFrame(data) print('Transactions:')
print(df.to_string(index=False)) filtered = df[df['Amount'] > 1000]
print('\nFiltered (Amount > 1000):')
print(filtered.to_string(index=False)) grp =
filtered.groupby('Customer')['Amount'].agg(['sum','mean']).reset_index(
) print('\nTotal and Average spending per Customer:')
print(grp.to_string(index=False))
```

Output:

```
Transactions:
 TransactionID Customer  Amount       Date
1        C2    2710 2025-10-22
2        C3    4282 2025-10-23
3        C3    2159 2025-10-24
4        C4     407 2025-10-25
5        C3    1252 2025-10-26
6        C4     375 2025-10-27
7        C4    1308 2025-10-28
8        C3    2592 2025-10-29
9        C3    2636 2025-10-30
10       C4     737 2025-10-31
11       C1     870 2025-11-01
12       C2     194 2025-11-02

Filtered (Amount > 1000):
 TransactionID Customer  Amount       Date
1       C2    2710 2025-10-22
2       C3    4282 2025-10-23
3       C3    2159 2025-10-24
             5       C3    1252 2025-10-26
7       C4    1308 2025-10-28
8       C3    2592 2025-10-29
9       C3    2636 2025-10-30

Total and Average spending per Customer:
Customer   sum   mean
      C2  2710 2710.0
      C3 12921 2584.2
      C4  1308 1308.0
```

Question 6: Create a CSV file named employee_salary.csv with columns EmpID, Name, Department, Salary. Load the data, filter employees with salary above 60,000, group by Department, and calculate mean, max, and min salary per department.

Name: Sonali Shintre

Roll No: 29

Code:

```
import pandas as pd, numpy as np data
= {
    'EmpID': range(1,11),
    'Name': [f'E{i}' for i in range(1,11)],
    'Department': np.random.choice(['HR','IT','Sales','Finance'],10),
    'Salary': np.random.randint(30000,120001,10)
} df = pd.DataFrame(data) print('Employee Salary Data:')
print(df.to_string(index=False)) filtered = df[df['Salary'] > 60000]
print('\nFiltered (Salary > 60000):')
print(filtered.to_string(index=False)) grp =
filtered.groupby('Department')['Salary'].agg(['mean','max','min']).rese
t_index() print('\nMean, Max, Min salary per Department:')
print(grp.to_string(index=False))
```

Output:

```
Employee Salary Data:
 EmpID Name Department  Salary
1   E1      Sales  102130
2   E2    Finance   51752
3   E3         IT   89715
4   E4         HR  119068
5   E5         HR   37997
6   E6         HR   83006
7   E7    Finance   70800
8   E8    Finance   45620
9   E9         HR  100381
10  E10      Sales   84268

Filtered (Salary > 60000):
 EmpID Name Department  Salary
     1   E1      Sales  102130
3   E3         IT   89715
4   E4         HR  119068
6   E6         HR   83006
7   E7    Finance   70800
9   E9         HR  100381
10  E10      Sales   84268

Mean, Max, Min salary per Department:
Department          mean    max   min
   Finance  70800.000000  70800 70800
        HR 100818.333333 119068 83006
        IT  89715.000000  89715 89715
     Sales  93199.000000 102130 84268
```

Question 7: Create a CSV file named orders.csv with columns OrderID, Customer, Product, Quantity, OrderDate. Load the data, filter orders with quantity greater than 5, group by Product, and calculate total quantity sold per product.

Code:

```
import pandas as pd, numpy as np data
= {
    'OrderID': range(1,13),
    'Customer': np.random.choice(['CustA','CustB','CustC'],12),
    'Product': np.random.choice(['P1','P2','P3','P4'],12),
    'Quantity': np.random.randint(1,15,12),
    'OrderDate': pd.date_range(end=pd.Timestamp('2025-11-02'),
periods=12).strftime('%Y-%m-%d')
} df = pd.DataFrame(data)
print('Orders:')
print(df.to_string(index=False))
filtered = df[df['Quantity'] > 5]
print('\nFiltered (Quantity >
5):')
print(filtered.to_string(index=Fa
lse)) grp =
filtered.groupby('Product')['Quan
tity'].sum().reset_index().rename
(colu
mns={'Quantity':'TotalQuantity'})
print('\nTotal quantity sold per
Product:')
print(grp.to_string(index=False))
```

Output:

```
Orders:
 OrderID Customer Product  Quantity  OrderDate
1     CustC      P1        10 2025-10-22
2     CustB      P2         7 2025-10-23
3     CustB      P3         6 2025-10-24
4     CustB      P2        14 2025-10-25
5     CustA      P2         8 2025-10-26
```

```
6     CustA      P4         9 2025-10-27
7     CustA      P2         9 2025-10-28
8     CustB      P1        10 2025-10-29
9     CustB      P1         3 2025-10-30
10    CustC      P1         9 2025-10-31
11    CustB      P4         7 2025-11-01
12    CustA      P4        12 2025-11-02

Filtered (Quantity > 5):
 OrderID Customer Product  Quantity  OrderDate
1    CustC      P1        10 2025-10-22
2    CustB      P2         7 2025-10-23
3    CustB      P3         6 2025-10-24
4    CustB      P2        14 2025-10-25
5    CustA      P2         8 2025-10-26
6    CustA      P4         9 2025-10-27
7    CustA      P2         9 2025-10-28
8    CustB      P1        10 2025-10-29
10    CustC      P1         9 2025-10-31
11    CustB      P4         7 2025-11-01
12    CustA      P4        12 2025-11-02

Total quantity sold per Product:
Product  TotalQuantity
     P1             29
     P2             38
     P3              6
     P4             28
```

Question 8: Create a CSV file named movies.csv with columns MovieID, Title, Genre, Rating. Load the data, filter movies with rating above 8, group by Genre, and calculate average, maximum, and minimum rating per genre.

Code:

```
import pandas as pd, numpy as np data
= {
    'MovieID': range(1,13),
    'Title': [f'Movie{i}' for i in range(1,13)],
    'Genre':
np.random.choice(['Action','Drama','Comedy','Thriller'],12),
    'Rating': np.round(5 + 5*np.random.rand(12),1)
```

Name: Sonali Shintre
Roll No: 29

```
} df = pd.DataFrame(data) print('Movies:')
print(df.to_string(index=False)) filtered = df[df['Rating'] > 8]
print('\nFiltered (Rating > 8):')
print(filtered.to_string(index=False)) grp =
filtered.groupby('Genre')['Rating'].agg(['mean','max','min']).reset_ind
ex() print('\nRating stats per Genre (for Rating>8):')
print(grp.to_string(index=False))
```

Output:

```
Movies:
 MovieID   Title    Genre  Rating
1  Movie1   Comedy     6.1
2  Movie2 Thriller     5.7
3  Movie3    Drama     5.5
4  Movie4    Drama     9.9
5  Movie5    Drama     6.3
6  Movie6 Thriller     7.7
7  Movie7   Comedy     7.2
8  Movie8   Action     5.5
9  Movie9   Action     6.8
10 Movie10    Drama     7.3
11 Movie11   Action     9.2
12 Movie12 Thriller     9.5

Filtered (Rating > 8):
 MovieID   Title    Genre  Rating
       4  Movie4    Drama     9.9
11 Movie11   Action     9.2
12 Movie12 Thriller     9.5

Rating stats per Genre (for Rating>8):
   Genre  mean  max  min
  Action   9.2  9.2  9.2
   Drama   9.9  9.9  9.9 Thriller
9.5  9.5  9.5
```

Question 9: Create a CSV file named weather.csv with columns Date, City, Temperature, Humidity. Load the data, filter days with temperature above 35°C, group by City, and calculate maximum, minimum, and average temperature for each city.

Name: Sonali Shintre

Roll No: 29

Code:

```
import pandas as pd, numpy as np dates =
pd.date_range(end=pd.Timestamp('2025-11-02'), periods=10) data =
{
    'Date': dates.strftime('%Y-%m-%d'),
    'City': np.random.choice(['CityA','CityB','CityC'],10),
    'Temperature': np.round(25 + 15*np.random.rand(10),1),
    'Humidity': np.random.randint(30,90,10)
} df = pd.DataFrame(data) print('Weather:')
print(df.to_string(index=False)) filtered = df[df['Temperature'] > 35]
print('\nFiltered (Temperature > 35°C):')
print(filtered.to_string(index=False)) grp =
filtered.groupby('City')['Temperature'].agg(['max','min','mean']).reset
_index() print('\nTemperature stats per City (for days >35°C):')
print(grp.to_string(index=False))
```

Output:

```
Weather:
      Date  City  Temperature  Humidity
2025-10-24 CityA         35.5        64
2025-10-25 CityC         31.8        54
2025-10-26 CityB         35.8        63
2025-10-27 CityB         38.0        35
2025-10-28 CityC         39.6        70
2025-10-29 CityA         37.8        66
2025-10-30 CityB         25.2        30
2025-10-31 CityB         30.4        41
2025-11-01 CityB         35.9        64
2025-11-02 CityB         27.6        35

Filtered (Temperature > 35°C):
      Date  City  Temperature  Humidity
2025-10-24 CityA         35.5        64
2025-10-26 CityB         35.8        63
2025-10-27 CityB         38.0        35
2025-10-28 CityC         39.6        70
2025-10-29 CityA         37.8        66
2025-11-01 CityB         35.9        64

Temperature stats per City (for days >35°C):
 City  max  min      mean
CityA 37.8 35.5 36.650000
CityB 38.0 35.8 36.566667 CityC
39.6 39.6 39.600000
```

Name: Sonali Shintre
Roll No: 29

Question 10: Create a CSV file named bank transactions.csv with columns TransactionID, Customer, Type, Amount. Load the data, filter deposits, group by Customer, and calculate total, maximum, and average deposit amount per customer.

Code:

```
import pandas as pd, numpy as np data
= {
    'TransactionID': range(1,13),
    'Customer': np.random.choice(['Cust1','Cust2','Cust3'],12),
'Type': np.random.choice(['Deposit','Withdrawal'],12, p=[0.6,0.4]),
    'Amount': np.random.randint(100,5000,12)
} df = pd.DataFrame(data)
print('Bank Transactions:')
print(df.to_string(index=Fa
lse)) filtered =
df[df['Type'] == 'Deposit']
print('\nFiltered
(Deposits):')
print(filtered.to_string(in
dex=False)) grp =
filtered.groupby('Customer'
)['Amount'].agg(['sum','max
','mean']).reset_ index()
print('\nDeposit stats per
Customer:')
print(grp.to_string(index=F
alse))
```

Output:

```
Bank Transactions:
 TransactionID Customer       Type  Amount
1     Cust3 Withdrawal    2913
2     Cust2 Withdrawal    3795
3     Cust3    Deposit    3513
4     Cust1 Withdrawal     523
5     Cust3 Withdrawal    3723
6     Cust1 Withdrawal    3834
7     Cust2 Withdrawal    3586
```

```
8     Cust2    Deposit    3829
9     Cust2    Deposit    1067
10    Cust1    Deposit    4470
11    Cust1    Deposit     960
12    Cust1    Deposit     533

Filtered (Deposits):
 TransactionID Customer    Type  Amount
             3    Cust3 Deposit    3513
8     Cust2 Deposit    3829
9     Cust2 Deposit    1067
10    Cust1 Deposit    4470
11    Cust1 Deposit     960
12    Cust1 Deposit     533

Deposit stats per Customer:
Customer  sum  max        mean
   Cust1 5963 4470 1987.666667
   Cust2 4896 3829 2448.000000
   Cust3 3513 3513 3513.000000
```

Question 11: Create a CSV file named hospital.csv with columns PatientID, Name, Department, Admission Date. Load the data, filter patients admitted in the last month, group by Department, and calculate the total number of patients per department.

Code:

```
import pandas as pd, numpy as np today = pd.Timestamp('2025-11-02')
dates = pd.to_datetime(pd.date_range(end=today, periods=60))  # 60 days
pool data = {
    'PatientID': range(1,21),
    'Name': [f'P{i}' for i in range(1,21)],
    'Department':
np.random.choice(['Cardiology','Neurology','Orthopedics','General'],20
) ,
    'Admission Date': np.random.choice(dates,20)
} df = pd.DataFrame(data) print('Hospital
Records (sample):')
print(df.head(10).to_string(index=False))
# filter last 30 days cutoff = today - pd.Timedelta(days=30) filtered
= df[df['Admission Date'] >= cutoff] print('\nFiltered (Admitted in
last 30 days):') print(filtered.to_string(index=False)) grp =
filtered.groupby('Department').size().reset_index(name='Count')
```

```
print('\nTotal patients per Department (last 30 days):')
print(grp.to_string(index=False))
```

Output:

```
Hospital Records (sample):
 PatientID Name  Department Admission Date
1   P1   Neurology     2025-10-10
2   P2   Neurology     2025-10-04
3   P3 Orthopedics     2025-09-28
4   P4   Neurology     2025-10-26
5   P5   Neurology     2025-09-07
6   P6   Neurology     2025-10-25
7   P7 Orthopedics     2025-09-12
8   P8  Cardiology     2025-10-30
9   P9     General     2025-10-01
10  P10   Neurology     2025-10-03

Filtered (Admitted in last 30 days):
 PatientID Name  Department Admission Date
1   P1   Neurology     2025-10-10
2   P2   Neurology     2025-10-04
         4   P4   Neurology     2025-10-26
         6   P6   Neurology     2025-10-25
         8   P8  Cardiology     2025-10-30
10  P10   Neurology     2025-10-03
11  P11  Cardiology     2025-11-02
12  P12     General     2025-10-20
        14  P14   Neurology     2025-10-06
        20  P20 Orthopedics     2025-11-01

Total patients per Department (last 30 days):
 Department  Count
 Cardiology      2
    General      1
  Neurology      6 Orthopedics
1
```

Question 12: Create a CSV file named online_store.csv with columns OrderID, Customer, Product, Amount. Load the data, filter orders with amount greater than 200, group by Customer, and calculate average, total, and maximum order amount per customer.

Name: Sonali Shintre
Roll No: 29

Code:

```
import pandas as pd, numpy as np data
= {
    'OrderID': range(1,16),
    'Customer': np.random.choice(['U1','U2','U3','U4'],15),
    'Product': np.random.choice(['X','Y','Z'],15),
    'Amount': np.random.randint(50,1001,15)
} df = pd.DataFrame(data) print('Online Orders:')
print(df.to_string(index=False)) filtered = df[df['Amount'] > 200]
print('\nFiltered (Amount > 200):')
print(filtered.to_string(index=False)) grp =
filtered.groupby('Customer')['Amount'].agg(['mean','sum','max']).reset_
index()
print('\nAvg, Total, Max order amount per Customer:')
print(grp.to_string(index=False))
```

Output:

```
Online Orders:
 OrderID Customer Product  Amount
1        U2       Z     636
2        U1       X     870
3        U1       Z     483
4        U4       Z     269
5        U1       X     869
6        U4       Y     789
7        U1       Y     923
8        U1       Y     836
9        U2       X     423
10       U1       X     340
11       U3       Y     613
12       U3       X     720
13       U3       X     487
14       U1       Y     876
15       U1       Z     989

Filtered (Amount > 200):
 OrderID Customer Product  Amount
1        U2       Z     636
2        U1       X     870
3        U1       Z     483
4        U4       Z     269
5        U1       X     869
```

```
6        U4       Y     789
7        U1       Y     923
8        U1       Y     836
9        U2       X     423
10       U1       X     340
11       U3       Y     613
12       U3       X     720
13       U3       X     487
14       U1       Y     876
15       U1       Z     989

Avg, Total, Max order amount per Customer:
Customer       mean  sum  max
      U1 773.250000 6186  989
      U2 529.500000 1059  636
      U3 606.666667 1820  720
      U4 529.000000 1058  789
```

Question 13: Create a CSV file named flights.csv with columns FlightID, Airline, Source, Destination, Delay. Load the data, filter flights with delay above 30 minutes, group by Airline, and calculate total flights, average, and maximum delay per airline.

Code:

```
import pandas as pd, numpy as np data
= {
    'FlightID': range(1,21),
    'Airline': np.random.choice(['AirA','AirB','AirC'],20),
    'Source': np.random.choice(['S1','S2','S3'],20),
    'Destination': np.random.choice(['D1','D2','D3'],20),
    'Delay': np.random.randint(0,120,20)
} df = pd.DataFrame(data) print('Flights:
(sample)')
print(df.head(10).to_string(index=False))
filtered = df[df['Delay'] > 30]
print('\nFiltered (Delay > 30 mins):')
print(filtered.to_string(index=False))
grp =
filtered.groupby('Airline').agg(TotalFlights=('FlightID','count'),
AverageDelay=('Delay','mean'), MaxDelay=('Delay','max')).reset_index()
print('\nDelay stats per Airline:') print(grp.to_string(index=False))
```

Name: Sonali Shintre

Roll No: 29

Output:

```
Flights: (sample)
 FlightID Airline Source Destination  Delay
1     AirA     S3          D2      9
2     AirA     S3          D1     54
3     AirC     S1          D1      0
4     AirB     S2          D1     83
5     AirB     S1          D1     36
6     AirC     S1          D2     81
7     AirC     S1          D3     20
8     AirB     S3          D3    117
9     AirA     S1          D2    112
10    AirC     S2          D1      3

Filtered (Delay > 30 mins):
 FlightID Airline Source Destination  Delay
        2    AirA     S3          D1     54
4    AirB     S2          D1     83
5    AirB     S1          D1     36
6    AirC     S1          D2     81
8    AirB     S3          D3    117
9    AirA     S1          D2    112
11    AirA     S1          D2     42
12    AirC     S3          D2     65
14    AirC     S1          D2     36
15    AirB     S2          D2     68
16    AirC     S3          D2     80
17    AirB     S1          D3    112
18    AirA     S1          D3     47
       20    AirB     S3          D2    109

Delay stats per Airline:
Airline  TotalFlights  AverageDelay  MaxDelay
   AirA            4         63.75       112
   AirB            6         87.50       117
   AirC            4         65.50        81
```

Question 14: Create a CSV file named inventory.csv with columns ProductID, ProductName, Supplier, Stock. Load the data, filter out-of-stock products, group by Supplier, and calculate total out-of-stock products per supplier.

Name: Sonali Shintre

Roll No: 29

Code:

```
import pandas as pd, numpy as np data
= {
    'ProductID': range(1,16),
    'ProductName': [f'Item{i}' for i in range(1,16)],
    'Supplier': np.random.choice(['SupA','SupB','SupC'],15),
    'Stock': np.random.randint(0,30,15)
} df = pd.DataFrame(data)
print('Inventory:')
print(df.to_string(index=False))
filtered = df[df['Stock'] == 0]
print('\nFiltered (Out of stock):')
print(filtered.to_string(index=False))
grp =
filtered.groupby('Supplier').size().re
set_index(name='OutOfStockCount')
print('\nTotal out-of-stock products
per Supplier:')
print(grp.to_string(index=False))
```

Output:

```
Inventory:
 ProductID ProductName Supplier  Stock
1       Item1     SupA      1
2       Item2     SupC     17
3       Item3     SupA      0
4       Item4     SupC      7
5       Item5     SupB     24
6       Item6     SupA     26
7       Item7     SupA      4
8       Item8     SupC      3
9       Item9     SupB      5
10      Item10     SupC      6
11      Item11     SupC      3
12      Item12     SupB      2
13      Item13     SupA     25
14      Item14     SupA      8
15      Item15     SupC     28

Filtered (Out of stock):
 ProductID ProductName Supplier  Stock
         3       Item3     SupA      0
```

Name: Sonali Shintre
Roll No: 29

```
Total out-of-stock products per Supplier:
Supplier  OutOfStockCount
    SupA                1
```

Question 15: Create a CSV file named ecommerce.csv with columns OrderID, Category, SubCategory, Sales, Load the data, filter sales above 500, group by Category and SubCategory, and calculate total, average, and maximum sales per combination.

Code:

```
import pandas as pd, numpy as np data
= {
    'OrderID': range(1,21),
    'Category': np.random.choice(['Electronics','Home','Fashion'],20),
    'SubCategory': np.random.choice(['A','B','C'],20),
    'Sales': np.random.randint(50,2001,20)
} df = pd.DataFrame(data) print('Ecommerce Sales:')
print(df.to_string(index=False)) filtered = df[df['Sales'] > 500]
print('\nFiltered (Sales > 500):')
print(filtered.to_string(index=False)) grp =
filtered.groupby(['Category','SubCategory'])['Sales'].agg(['sum','mean'
,'max']).reset_index() print('\nTotal, Avg, Max sales per
Category-SubCategory:') print(grp.to_string(index=False))
```

Output:

```
Ecommerce Sales:
 OrderID    Category SubCategory  Sales
1         Home           A   1939
2         Electronics           B    478
3         Electronics           C    877
4         Fashion           B   1899
5         Electronics           B    740
6         Electronics           B    223
7         Home           A   1819
8         Home           A   1206
9         Home           C   1515
10        Home           A    143
11        Home           A   1421
```

Name: Sonali Shintre

Roll No: 29

```
12        Home           C   1475
13        Fashion           B    469
14        Electronics           C   1780
15        Fashion           C    454
16        Electronics           A    991
17        Home           C   1259
18        Electronics           C   1025
19        Fashion           A    425
20        Electronics           C   1750

Filtered (Sales > 500):
 OrderID    Category SubCategory  Sales
       1        Home           A   1939
3     Electronics           C    877
4     Fashion           B   1899
5     Electronics           B    740
7        Home           A   1819
8        Home           A   1206
9        Home           C   1515
11        Home           A   1421
12        Home           C   1475
      14 Electronics           C   1780
16        Electronics           A    991
17        Home           C   1259
18        Electronics           C   1025
      20 Electronics           C   1750

Total, Avg, Max sales per Category-SubCategory:
   Category SubCategory  sum        mean  max
Electronics           A  991  991.000000  991
Electronics           B  740  740.000000  740
Electronics           C 5432 1358.000000 1780
    Fashion           B 1899 1899.000000 1899
       Home           A 6385 1596.250000 1939
       Home           C 4249 1416.333333 1515
```