

Name:-Mohammad sayeed Kazi

Roll no:- C56

Class:-TY CSE Div: C

Experiment no:-13

Title:- Create Data frame ,load data from csv ,and perform basics data analysis such as filtering grouping and aggregation

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

```
import pandas as pd

data = {
    'Name': ['Amit', 'Ravi', 'Priya', 'Sneha', 'Karan'],
    'Age': [25, 35, 30, 40, 28],
    'Department': ['HR', 'IT', 'Finance', 'IT', 'Finance'],
    'Salary': [40000, 60000, 55000, 75000, 48000]
}
df = pd.DataFrame(data)
df.to_csv('employees.csv', index=False)

df = pd.read_csv('employees.csv')

filtered = df[df['Age'] > 30]

result = filtered.groupby('Department')['Salary'].agg(['mean', 'min', 'max'])
print(result)

          mean      min      max
Department
IT        67500.0  60000  75000
```

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

```
data = {
    'SaleID': [1,2,3,4,5],
    'Product': ['TV', 'Laptop', 'Mobile', 'TV', 'Laptop'],
    'Amount': [45000, 80000, 60000, 30000, 90000],
    'Customer': ['A', 'B', 'C', 'D', 'E']
}
df = pd.DataFrame(data)
df.to_csv('sales.csv', index=False)

df = pd.read_csv('sales.csv')
filtered = df[df['Amount'] > 50000]
result = filtered.groupby('Product')['Amount'].agg(['sum'])
print(result)

          sum      mean
Product
Laptop     170000  85000.0
Mobile      60000  60000.0
```

3. Create a CSV file named student_marks.csv with columns StudentID, Name, Maths, Science, English.
Load the data, filter students with Maths marks

Code:-

```
data = {
    'StudentID': [1,2,3,4],
    'Name': ['Anu','Ravi','Neha','Raj'],
    'Maths': [90,75,85,60],
    'Science': [80,70,95,55],
    'English': [85,65,88,70]
}
df = pd.DataFrame(data)
df.to_csv('student_marks.csv', index=False)

df = pd.read_csv('student_marks.csv')
filtered = df[df['Maths'] > 80]
df['Total'] = df[['Maths','Science','English']].sum(axis=1)
df['Average'] = df[['Maths','Science','English']].mean(axis=1)
result = df.groupby('Name')[['Total','Average']].mean()
print(result)
```

	Total	Average
Name		
Anu	255.0	85.000000
Neha	268.0	89.333333
Raj	185.0	61.666667
Ravi	210.0	70.000000

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

Code:-

```
data = {
    'ProductID':[1,2,3,4,5],
    'ProductName':['Pen','Book','Bag','Pencil','Notebook'],
    'Category':['Stationery','Stationery','Bags','Stationery','Stationery'],
    'Stock':[20,100,40,15,10]
}
df = pd.DataFrame(data)
df.to_csv('products.csv', index=False)
```

```
df = pd.read_csv('products.csv')
filtered = df[df['Stock'] < 50]
result = filtered.groupby('Category')['Stock'].agg(['sum','mean'])
print(result)
```

	sum	mean
Category		
Bags	40	40.0
Stationery	45	15.0

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

```

data = {
    'TransactionID':[1,2,3,4],
    'Customer':['A','B','A','C'],
    'Amount':[1200,800,1500,2000],
    'Date':['2025-10-01','2025-10-05','2025-10-07','2025-10-10']
}
df = pd.DataFrame(data)
df.to_csv('transactions.csv', index=False)

df = pd.read_csv('transactions.csv')
filtered = df[df['Amount'] > 1000]
result = filtered.groupby('Customer')['Amount'].agg(['sum','mean'])
print(result)

```

	sum	mean
Customer		
A	2700	1350.0
C	2000	2000.0

Delimiter: , ▾

	TransactionID	Customer	Type	Amount
1	1	A	Deposit	5000
2	2	B	Withdraw	2000
3	3	A	Deposit	8000
4	4	C	Deposit	10000

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

```
data = {
    'EmpID':[1,2,3,4],
    'Name':['Amit','Ravi','Kiran','Sneha'],
    'Department':['IT','HR','IT','Finance'],
    'Salary':[50000,70000,90000,80000]
}
df = pd.DataFrame(data)
df.to_csv('employee_salary.csv', index=False)

df = pd.read_csv('employee_salary.csv')
filtered = df[df['Salary'] > 60000]
result = filtered.groupby('Department')[['Salary']].agg(['mean','max','min'])
print(result)

          mean     max     min
Department
Finance    80000.0  80000  80000
HR          70000.0  70000  70000
IT          90000.0  90000  90000
```

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

```

data = {
    'OrderID':[1,2,3,4],
    'Customer':['A','B','C','D'],
    'Product':['TV','Laptop','TV','Mobile'],
    'Quantity':[3,6,8,2],
    'OrderDate':['2025-10-01','2025-10-02','2025-10-05','2025-10-08']
}
df = pd.DataFrame(data)
df.to_csv('orders.csv', index=False)

df = pd.read_csv('orders.csv')
filtered = df[df['Quantity'] > 5]
result = filtered.groupby('Product')['Quantity'].sum()
print(result)

Product
Laptop    6
TV        8
Name: Quantity, dtype: int64

```

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.

```

data = {
    'MovieID':[1,2,3,4],
    'Title':['A','B','C','D'],
    'Genre':['Action','Drama','Action','Comedy'],
    'Rating':[9,7,8.5,9.1]
}
df = pd.DataFrame(data)
df.to_csv('movies.csv', index=False)

df = pd.read_csv('movies.csv')
filtered = df[df['Rating'] > 8]
result = filtered.groupby('Genre')['Rating'].agg(['mean','max','min'])
print(result)

      mean   max   min
Genre
Action  8.75  9.0  8.5
Comedy  9.10  9.1  9.1

```

	MovieID	Title	Genre	Rating
1	1	A	Action	9.0
2	2	B	Drama	7.0
3	3	C	Action	8.5
4	4	D	Comedy	9.1

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,

```
data = {
    'Date': ['2025-10-01', '2025-10-02', '2025-10-03', '2025-10-04'],
    'City': ['Mumbai', 'Delhi', 'Mumbai', 'Chennai'],
    'Temperature': [36, 34, 38, 37],
    'Humidity': [70, 50, 80, 60]
}
df = pd.DataFrame(data)
df.to_csv('weather.csv', index=False)

df = pd.read_csv('weather.csv')
filtered = df[df['Temperature'] > 35]
result = filtered.groupby('City')['Temperature'].agg(['max', 'min', 'mean'])
print(result)
```

	max	min	mean
City			
Chennai	37	37	37.0
Mumbai	38	36	37.0

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.

```
data = {
    'TransactionID':[1,2,3,4],
    'Customer':['A','B','A','C'],
    'Type':['Deposit','Withdraw','Deposit','Deposit'],
    'Amount':[5000,2000,8000,10000]
}
df = pd.DataFrame(data)
df.to_csv('bank_transactions.csv', index=False)

df = pd.read_csv('bank_transactions.csv')
filtered = df[df['Type'] == 'Deposit']
result = filtered.groupby('Customer')['Amount'].agg(['sum','max','mean'])
print(result)
```

	sum	max	mean
Customer			
A	13000	8000	6500.0
C	10000	10000	10000.0

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

```
data = {
    'PatientID':[1,2,3,4],
```

```

'Name':['A','B','C','D'],
'Department':['Cardio','Neuro','Ortho','Cardio'],
'AdmissionDate':['2025-10-05','2025-09-20','2025-10-10','2025-10-15']
}
df = pd.DataFrame(data)
df.to_csv('hospital.csv', index=False)

df = pd.read_csv('hospital.csv')
df['AdmissionDate'] = pd.to_datetime(df['AdmissionDate'])
filtered = df[df['AdmissionDate'] > '2025-09-30']
result = filtered.groupby('Department')['PatientID'].count()
print(result)

```

```

Department
Cardio    2
Ortho     1
Name: PatientID, dtype: int64

```

Delimiter: , ▾

	PatientID	Name	Department	AdmissionDate
1	1	A	Cardio	2025-10-05
2	2	B	Neuro	2025-09-20
3	3	C	Ortho	2025-10-10
4	4	D	Cardio	2025-10-15

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.

```

data = {
    'OrderID':[1,2,3,4],
    'Customer':['A','B','A','C'],
    'Product':['Pen','Book','Bag','Laptop'],
    'Amount':[250,150,300,400]
}
df = pd.DataFrame(data)
df.to_csv('online_store.csv', index=False)

df = pd.read_csv('online_store.csv')
filtered = df[df['Amount'] > 200]
result = filtered.groupby('Customer')['Amount'].agg(['mean','sum','max'])
print(result)

```

	mean	sum	max
Customer			
A	275.0	550	300
C	400.0	400	400

Delimiter: , ▾

	OrderID	Customer	Product	Amount
1	1	A	Pen	250
2	2	B	Book	150
3	3	A	Bag	300
4	4	C	Laptop	400

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.

```
data = {
    'FlightID':[1,2,3,4,5],
    'Airline':['Indigo','Air India','Indigo','Vistara','Air India'],
    'Source':['Mumbai','Delhi','Pune','Goa','Delhi'],
    'Destination':['Delhi','Mumbai','Delhi','Pune','Goa'],
    'Delay':[40,20,50,10,60]
}
df = pd.DataFrame(data)
df.to_csv('flights.csv', index=False)

df = pd.read_csv('flights.csv')
filtered = df[df['Delay'] > 30]
result = filtered.groupby('Airline')['Delay'].agg(['count','mean','max'])
print(result)

      count  mean  max
Airline
Air India      1  60.0   60
Indigo         2  45.0   50
```

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.

```
data = {
    'ProductID':[1,2,3,4],
    'ProductName':['Pen','Book','Bag','Pencil'],
    'Supplier':['A','A','B','B'],
    'Stock':[0,100,0,10]
}
df = pd.DataFrame(data)
df.to_csv('inventory.csv', index=False)

df = pd.read_csv('inventory.csv')
filtered = df[df['Stock'] == 0]
result = filtered.groupby('Supplier')['ProductID'].count()
print(result)

Supplier
A    1
B    1
Name: ProductID, dtype: int64
```

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

```
data = {
    'OrderID':[1,2,3,4],
    'Category':['Electronics','Clothing','Electronics','Clothing'],
    'SubCategory':['Mobile','Shirt','Laptop','Pants'],
    'Sales':[600,300,800,700]
}
df = pd.DataFrame(data)
df.to_csv('ecommerce.csv', index=False)

df = pd.read_csv('ecommerce.csv')
filtered = df[df['Sales'] > 500]
result = filtered.groupby(['Category','SubCategory'])['Sales'].agg(['sum','mean','max'])
print(result)
```

		sum	mean	max
Category	SubCategory			
Clothing	Pants	700	700.0	700
Electronics	Laptop	800	800.0	800
	Mobile	600	600.0	600

	SaleID	Product	Amount	Customer
1	1	TV	45000	A
2	2	Laptop	80000	B
3	3	Mobile	60000	C
4	4	TV	30000	D
5	5	Laptop	90000	E