1. What is Dask?

- **Dask** is a parallel computing library that integrates well with **Pandas** and **NumPy** for scaling computations to larger-than-memory datasets.
- It operates by dividing data into smaller chunks that can be processed in parallel on a single machine or distributed across multiple machines.

2. Sample Data:
Pase and create your own CSV files. Here is a simple CSV file named `sample_data.csv`:

```
id,name,age,department,salary
1,John,28,HR,50000
2,Jane,32,Engineering,70000
3,Mark,45,HR,60000
4,Emily,29,Engineering,75000
5,Anna,35,HR,55000
6,Luke,40,Engineering,80000
7,Mia,22,Marketing,45000
8,James,33,Marketing,48000
9,Grace,38,HR,57000
10,David,41,Engineering,85000
```

Install Dask:
pip install dusk

An Example of Data Wrangling with Dask:

```
import dask.dataframe as dd

# Load the sample data into a Dask DataFrame
df = dd.read_csv('sample_data.csv')

# Display the first few rows of the dataframe
print("First 5 rows of the data:")
print(df.head())

# Compute the average salary by department
avg_salary_by_department = df.groupby('department')['salary'].mean().compute()
print("\nAverage Salary by Department:")
print(avg_salary_by_department)

# Filter out people aged over 30
filtered_df = df[df['age'] > 30]
print("\nPeople older than 30:")
print(filtered_df.compute())

# Apply a function to each row (for example, categorize salary)
def categorize_salary(salary):
```

```
    if salary > 70000:
        return 'High'
    elif salary > 50000:
        return 'Medium'
    else:
        return 'Low'

df['salary_category'] = df['salary'].apply(categorize_salary, meta=('x', 'object'))

# Show the DataFrame with the new salary category
print("\nData with salary category:")
print(df.compute())

# Save the resulting DataFrame to a new CSV file
df.to_csv('output.csv', index=False, single_file=True)
```

3. Expected Output:

- The **first few rows** of the dataset will be printed.
- The **average salary by department** will be computed.
- The **filtered dataset** for people older than 30 will be displayed.
- The **new DataFrame** with salary categories will be shown.
- A new **CSV file (`output.csv`)** will be generated with the new data.

4. Scaling Dask for Big Data:

- Dask operates on **multiple cores** on a single machine or can be run on a **cluster** with multiple nodes. You can scale your computation as needed by setting up a **Dask cluster**.
- For larger datasets, Dask will break them into partitions, so you can operate on each partition in parallel, making the operations faster than using Pandas for very large datasets.

```
from dask.distributed import Client

# Start a local Dask client to monitor computations

client = Client()

# Now execute the same code as above, and Dask will distribute the work
```

Summary:

This is a basic introduction to using Dask for **data wrangling** tasks. You can easily scale up operations to handle large datasets, use **parallel computations**, and make your workflows efficient when dealing with **big data**.