Session 1 Manual

Topic: *Introduction to Pandas – Data Types, Cleaning, Missing Values, & Duplicates*

Focus: Getting started with Pandas, exploring data, cleaning operations, handling data types, missing values, and duplicates.

1. What is Pandas?

Pandas is a Python library built on top of NumPy that simplifies working with structured data. It turns raw, messy datasets into structured tables (rows & columns), making analysis and cleaning easier.

Main Data Structures:

Series → A single column (1D).

DataFrame → A full table (2D).

Installation: `pip install pandas`

Importing: `import pandas as pd`

2. Exploring DataFrames

When you first load data (usually from `.csv`, `.xlsx`, or `.json`), exploration helps you "get to know your data."

Load data: `df = pd.read_csv("data.csv")`

Quick overview:

- `df.head()` → First 5 rows
- `df.tail()` → Last 5 rows
- `df.shape` → (rows, columns)
- `df.columns` → List of column names
- `df.index` → Row index labels
- `df.info()` → Column types, null counts, memory usage
- `df.describe(include="all")` → Summary stats (numerical + categorical if `include="all"`)

3. Selecting Data with `loc` and `iloc`

Two main ways to slice and select data:

1. **`loc` (label-based)** – Uses row labels and column names.

```
df.loc[0, "Name"]                    # First row, Name column
df.loc[0:5, ["Name", "Age"]]         # Rows 0–5, Name & Age columns
df.loc[df["Age"] > 30, "Name"]       # Filter names where Age > 30
```

2. **`iloc` (integer-based)** – Uses row/column positions.

```
df.iloc[0, 1]                        # First row, second column
df.iloc[0:5, 0:3]                    # First 5 rows, first 3 columns
df.iloc[:, -1]                       # All rows, last column
```

4. Cleaning Operations

Raw data is often messy. Pandas gives powerful tools to clean it.

a) Checking for nulls

```
df.isnull().sum()                    # Count nulls per column
df.notnull().sum()                   # Count non-nulls
```

b) Dropping

Drop rows with nulls: `df.dropna()`

Drop columns with nulls: `df.dropna(axis=1)`

c) Filling

Fill with constant: `df.fillna(0)`

Fill with mean/median/mode: `df["Age"].fillna(df["Age"].mean(), inplace=True)`

d) Removing Duplicates

```
df.duplicated().sum()                # Count duplicates
df.drop_duplicates()                 # Remove them
```

e) Dropping Columns/Rows

```
df.drop("column_name", axis=1, inplace=True)   # Drop the named column
df.drop([0, 1], axis=0, inplace=True)   # Drop rows 0 and 1
```

f) Text Cleaning

Convert case: `df["Name"].str.lower()`

Strip spaces: `df["Name"].str.strip()`

Replace characters: `df["Email"].str.replace("@gmail.com", "@outlook.com")`

5. Data Types

Every column in Pandas has a dtype.

Common dtypes:

- `object` → text/strings
- `int64` → whole numbers
- `float64` → decimals
- `datetime64` → dates/times
- `bool` → True/False

Check types: `df.dtypes`

Convert types:

```
df["Age"] = df["Age"].astype(int)
df["Salary"] = df["Salary"].astype(float)
df["JoinDate"] = pd.to_datetime(df["JoinDate"])
```

6. Handling Missing Values

Missing data is inevitable. Some approaches to handle them:

Drop rows/columns – Good for small datasets with lots of nulls.

```
df.dropna()
```

Fill with fixed value – For placeholders (e.g., `Unknown`, `0`).

```
df.fillna("Unknown")
```

Fill with statistics – Mean/Median/Mode for numeric values.

```
df["Age"].fillna(df["Age"].median())
```

Forward Fill (ffill): `df.fillna(method="ffill")`

Backward Fill (bfill): `df.fillna(method="bfill")`

Interpolation – Estimates values between existing data points (time series).

```
df["Sales"].interpolate(method="linear")
```

Note: *Choose based on dataset size, column type, and analysis needs.*

7. Duplicate Handling

Duplicates can happen due to human error or repeated imports.

Check: `df.duplicated().sum()`

Remove: `df.drop_duplicates(inplace=True)`

Keep first/last:

```
df.drop_duplicates(keep="first")    # Keep first occurrence
df.drop_duplicates(keep="last")     # Keep last occurrence
```

8. Practical Mini-Workflow

1. Load data → `pd.read_csv()`
2. Inspect → `head()`, `info()`, `describe()`
3. Select slices → `loc`, `iloc`
4. Detect & fix nulls → `isnull()`, `fillna()`
5. Handle dtypes → `astype()`, `to_datetime()`
6. Remove duplicates → `drop_duplicates()`
7. Clean text → `str.lower()`, `str.strip()`
8. Save → `df.to_csv("cleaned_data.csv", index=False)`