**Pandas**

# Creation of DataFrames in Pandas

Data = pd.read\_csv(path/file\_name.csv)

Data = pd.read\_csv(file\_name.csv)

Data = pd.read\_excel(file\_name.xlsx)

A = pd.DataFrames(Data)

# Exploring Data in Pandas

a = data.head(10) show first ten row

a = data.tail(10) show last ten row

a = data.info() show information about data

a = data.describe() describe the data

a = data.isnull().sum() check the null values in all data

# Handling Duplicate values in Pandas

a = data.duplicated().sum() check duplicate values in all data

a = data[“ID”].duplicated().sum() check duplicate values in one column

a = data.drop\_duplicates() drop the duplicates values in data

a = data.drop\_duplicates(“column\_name”)

# Working with Missing Data in Pandas

a = data.dropna() drop null values in data  
data["Salary"]= data["Salary"].replace(np.nan, 81543) to fill duplicate values using numeric

data["Gender"] = data["Gender"].fillna(method = "ffill" or “bfill”) to fill data by up and down values this method use for string

# Columns Transformation in Pandas

1. **if I want to create new column there I can check that which values True and False**

data.loc[(data["column\_name"] >= 30), "new\_column\_name"] = "enter 1st value"

data.loc[(data["column\_name "] < 30), " new\_column\_name "] = “enter 2nd value "

1. **if I want to create new column there I can combine two columns**

data["Create\_new\_column"] = data["1st\_column\_name"].str.capitalize() + " " + data["2nd\_column\_name"]

1. **if I want to create new column for bouns**

data["create\_new\_column"] = data["column\_name"]/100\*20

1. **if I want to create short column**

def extract(value):

return value[0:3]

data["create\_new\_column"] = data["column\_name"].map(extract)

# GroupBy in Pandas

gb = data.groupby(["Department", "Gender"]).agg({"Salary":"max",})

# Merge and Concatenate in Pandas

**Merge** use for merge data like merge columns

print(pd.merge(1st\_column,2nd\_column, on = "same\_column\_name"))

print(pd.merge(1st\_column,2nd\_column, on = "same\_column\_name" , how = "left")) use if NaN

**Concatenate** use for add two data all data should be same

print(pd.concat([1st\_column,2nd\_column]))

# Compare DataFrames

Print(1st\_data.compare(2nd\_data, keep\_shape = True))

# Pivoting and Melting DataFrames

**Pivoting**

dic = {"keys": ["k1", "k2", "k1", "k2"],

"Names": ["Haroon", "Sajjad", "Ali", "Amjad"],

"Houses": ["Green", "Red", "Blue", "Black"],

"Grades": ["3rd","8th", "9th", "5th"]}

df = pd.DataFrame(dic)

print(df)

print(df.pivot(index="keys",columns="Names",values=["Houses","Grades"]))

**Melting**

above

print(pd.melt(df, id\_vars=["Names"],value\_vars=["Houses","Grades"]))

print(pd.melt(df,id\_vars=["Names"],value\_vars=["Houses","Grades"],var\_name="Houses&Grades",value\_name="values"))