

# DataFrame operations 3

---

UTKARSH GAIKWAD

CLASS STARTING SHARP AT 3:05 PM

# Non harmful warnings ignored

---

```
from warnings import filterwarnings
```

```
filterwarnings('ignore')
```

# Basic functions

---

- `df.head()`
- `df.tail()`
- `df.nunique()`
- `df['column'].unique()`
- `df.isna().sum()`

# Some Problem statements from Cars93

---

- Find out the car with maximum weight on entire data.
- Find out cars with price is greater than average price
- Find out car with maximum weight in compact category
- Find out the car with minimum price and belongs to Van Type
- Find out top7 cars with highest mileage In city

# Create a column based on conditions

---

Price < 17 --> Cheap

17 - 22 --> Moderate|

22+ --> Expensive

# Creating new columns from other columns

---

➤ `df['new_column_name'] = df operations`

# Create price column showing price in USD

---

Price of cars is in 1 Lakhs INR

Check current usd price

# Replacing missing values in DataFrame

---

- `df.isna().sum()`
- Separate categorical and numeric values with filter operations
- categorical ----> `mode()` Most frequently occurring values
- continuous -----> mean or median
- Function to replace missing values `df.fillna(value)`

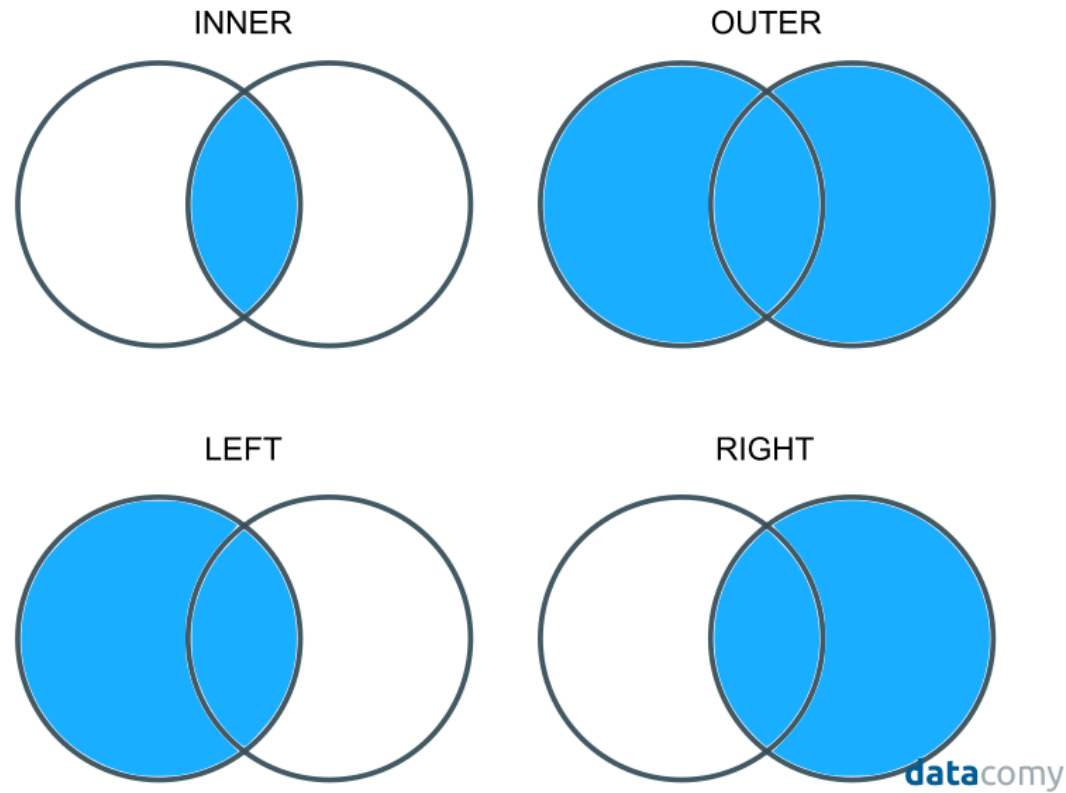
Continuous (Numeric) Values Missing  
Mean, Median

Categorical Values Missing  
Mode



# Combining 2 Dataframes with merge

---



# Inner Join

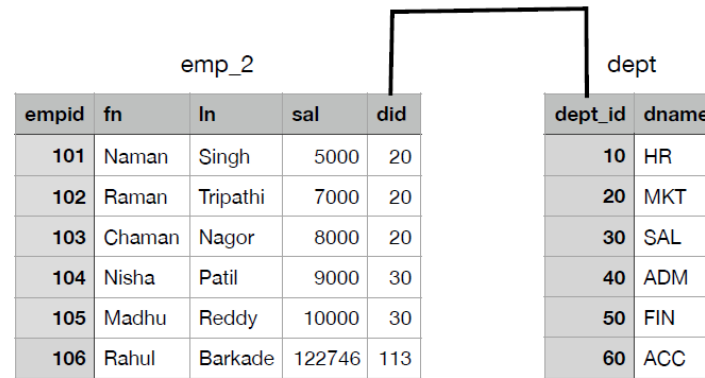
---

emp_2					dept	
empid	fn	ln	sal	did	dept_id	dname
101	Naman	Singh	5000	20	10	HR
102	Raman	Tripathi	7000	20	20	MKT
103	Chaman	Nagor	8000	20	30	SAL
104	Nisha	Patil	9000	30	40	ADM
105	Madhu	Reddy	10000	30	50	FIN
106	Rahul	Barkade	122746	113	60	ACC

INNER_JOIN_OP						
empid	fn	ln	sal	did	dept_id	dname
101	Naman	Singh	5000	20	20	MKT
102	Raman	Tripathi	7000	20	20	MKT
103	Chaman	Nagor	8000	20	20	MKT
104	Nisha	Patil	9000	30	30	SAL
105	Madhu	Reddy	10000	30	30	SAL

# Left Join

---



LO\_JOIN\_OP

empid	fn	ln	sal	did	dept_id	dname
101	Naman	Singh	5000	20	20	MKT
102	Raman	Tripathi	7000	20	20	MKT
103	Chaman	Nagor	8000	20	20	MKT
104	Nisha	Patil	9000	30	30	SAL
105	Madhu	Reddy	10000	30	30	SAL
106	Rahul	Barkade	122746	113	NULL	NULL

# Right Join

---

emp_2					dept	
empid	fn	ln	sal	did	dept_id	dname
101	Naman	Singh	5000	20	10	HR
102	Raman	Tripathi	7000	20	20	MKT
103	Chaman	Nagor	8000	20	30	SAL
104	Nisha	Patil	9000	30	40	ADM
105	Madhu	Reddy	10000	30	50	FIN
106	Rahul	Barkade	122746	113	60	ACC

RO\_JOIN\_OP

empid	fn	ln	sal	did	dept_id	dname
NULL	NULL	NULL	NULL	NULL	10	HR
102	Raman	Tripathi	7000	20	20	MKT
103	Chaman	Nagor	8000	20	20	MKT
101	Naman	Singh	5000	20	20	MKT
104	Nisha	Patil	9000	30	30	SAL
105	Madhu	Reddy	10000	30	30	SAL
NULL	NULL	NULL	NULL	NULL	40	ADM
NULL	NULL	NULL	NULL	NULL	50	FIN
NULL	NULL	NULL	NULL	NULL	60	ACC

# Full Outer

empid	fn	ln	sal	did
101	Naman	Singh	5000	20
102	Raman	Tripathi	7000	20
103	Chaman	Nagor	8000	20
104	Nisha	Patil	9000	30
105	Madhu	Reddy	10000	30
106	Rahul	Barkade	122746	113

dept_id	dname
10	HR
20	MKT
30	SAL
40	ADM
50	FIN
60	ACC

empid	fn	ln	sal	did	dept_id	dname
NULL	NULL	NULL	NULL	NULL	10	HR
102	Raman	Tripathi	7000	20	20	MKT
103	Chaman	Nagor	8000	20	20	MKT
101	Naman	Singh	5000	20	20	MKT
104	Nisha	Patil	9000	30	30	SAL
105	Madhu	Reddy	10000	30	30	SAL
NULL	NULL	NULL	NULL	NULL	40	ADM
NULL	NULL	NULL	NULL	NULL	50	FIN
NULL	NULL	NULL	NULL	NULL	60	ACC
106	Rahul	Barkade	122746	113	NULL	NULL

# Pandas merge function

---

- `pd.merge(emp,dept,how='inner',left_on="did", right_on="dept_id")`
- `pd.merge(emp,dept,how='left',left_on="did", right_on="dept_id")`
- `pd.merge(emp,dept,how='right',left_on="did", right_on="dept_id")`
- `pd.merge(emp,dept,how='outer',left_on="did", right_on="dept_id")`

# Thank you

---

PLEASE PING ME ON SKYPE GROUP IF YOU HAVE ANY QUERIES