

- 1 convert time series operations into datetime object
- `df = pd.read_csv("weather.csv", parse_dates=["day"])`
- 2 df.set\_index("day") → index set করা      3 df.reset\_index(inplace=True)
- 4 df.isna().sum()      5 কোন column পরে change করা
- 6 fill NaN value with 0      df.temp = pd.Series([5, 6, np.nan, 3])
- `tf = df.fillna(0)`
- 7 `{ "temp": 0, "wind": 1, "even": "NO" }`      8 পরের value টাকা আগে value fill করা → `tf = df.fillna(method="ffill")`
- 9 forward fill → all
- 10 set limit  
`tf = df.fillna(method="ffill", limit=3)`
- 11 last nan → always  
`tf = df.fillna(method="ffill", limit=1)`
- 12 Normal interpolate  
`df.interpolate()`      parse\_date=True
- 13 Time interpolation  
 time interpolation দিয়ে  
 একটি শাল index টি  
 Datetime 2020-01-01
- `+f = df.set_index("day")`  
`gf = +f.interpolate(method='time')`

14 drop rows that contain any missing value

↳ bad

$tf = df.dropna()$

$len(tf) \rightarrow$  new size.

15 drop rows if all value are missing

$tf = df.dropna(how='all')$

16 minimum number value ते अंतर drop २०

↳ must ले कर्यावाचक अंतर २०

$tf = df.dropna(thresh=3) \rightarrow$  minimum ३ ते

→ २० ८२ २०

17 ty. dtyps -

18 replace ले  
→ -9 or nan ले

$of = df.replace(-9, np.nan)$

19 multiple value ले replace ले

20

$of = df.replace({$

'temp': -9,

'wind': -8,

'even': '0'

$, np.nan})$

$of = df.replace(to_replace=[-8, -9], value=0)$

21 यह  $\rightarrow$  np.nan एवं 'no event' के sunny वर्ग  
 $df = df.replace($   
 $\{$   
 $-\emptyset : np.nan,$   
 $"no event" : "sunny"$   
 $\})$

22 Regex  $\rightarrow$  extra unit (F1 amp) या दृश्या

'[A-zA-Z]'  $\rightarrow$  A to Z or a to Z

$df = df.replace($   
 $\{$   
 $'temp' : re '[A-zA-Z]',$   
 $'wind' : re '[A-zA-Z]'$   
 $"", regex = True$   
 $)$

space या दृश्या & numeric बनाना

$df["temp"] = pd.to_numeric(df["temp"],$   
 $astype(str), str.strip(), errors =$   
 $"coerce")$

coerce  $\rightarrow$  ए पूर्ण नान बदले

② replace list with another list.

```
grades = pd.DataFrame({  
    "score": ["exceptional", "average", "good", "poor", "average"],  
    "student": ["rob", "maya", "paul", "tom", "julian", "eric"]  
})
```

score	student
exceptional	rob
average	maya
good	paul
poor	tom
average	julian

Score: good

poor

average

ng

on

julin

```
grades_num = grades.replace({
```

```
    "poor": "average", "good": "exp."},
```

[1, 2, 3, 4]

score	student
1	rob
2	maya
3	paul
4	tom

1	rob
2	maya
3	tom
4	julin

## Scikit & Menge : →

x\_train = pd.DataFrame({

ID	age	bp	
0	1	20	120
1	2	21	100
2	3	22	90

{'ID': [1, 2, 3],  
 'age': [20, 21, 22],  
 'bp': [120, 100, 90]},

{}

x\_test = pd.DataFrame({

IP	age	bp	
0	5	20	150
1	6	25	90

{'IP': [5, 6],  
 'age': [20, 25],  
 'bp': [150, 90]},

{}

y\_train = pd.DataFrame({

FD	target
0	1
1	2
2	3
3	4

{'FD': [1, 2, 3, 4],  
 'target': [0, 1, 0, 1]},

{}

FD	target
0	0
1	1
2	0
3	1

① concat vertically (axis=0) :- several bases

```
X_all = pd.concat([X_train, X_test], axis=0)
```

		IP	age	bp
0	1			
1	2			
2	3			
3	4			
0	5			
0	0			
1				

$X\_all = pd.concat([X\_{train}, X\_{test}], axis=0, ignore\_index=True)$

$$\begin{array}{c}
 \text{L} \\
 \left( \begin{array}{ccc|c}
 & 0 & \frac{10}{11} & \\
 1 & & & \\
 \\ 
 2 & 2 & 3 & \\
 & 3 & 9 & \\
 3 & 9 & 5 & \\
 & 5 & 6 & \\
 \hline
 \end{array} \right) \\
 \left\{ \begin{array}{l}
 \text{Row 3 - Row 1} \\
 \text{Row 4 - Row 1} \\
 \text{Row 5 - Row 1} \\
 \text{Row 6 - Row 1}
 \end{array} \right.
 \end{array}$$

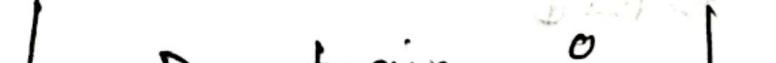
`x-all = pd.concat([x-train, x-test], keys = ["train", "test"])`

	train	test	
0	1	0	1
1	2	1	2
2	3	2	3
3	4	3	4
4	5	4	5
5	6	5	6

- access train block

三

X-all = loc [ "train" ]

•  $[f(x) \text{ and } x]$  )  $\rightarrow$   $\text{train}$  

(Endings)

• concat variants : →

$X\_all = pd.concat([X\_train, X\_test], axis=1)$

13 now शाला suffel करते हुए index भी बदलो  
(एग्जेम्पल)

$X\_all\_new = X\_all.sample(frac=1, random_state=0).reset_index(drop=True)$

14 ID share के बारे में index विद्युतीय concat

→ Bad

bad\_with =  $pd.concat([X\_train, X\_test], axis=1)$

⑯ ID share ~~to~~ add  
new row by filling values of

$x_{\text{new}} = x_{\text{train}}. \text{set\_index}(\text{"ID"})$

$x_{\text{new1}} = x_{\text{new}}. \text{assign}($

$\text{id} = x_{\text{train}}[\text{"ID"}]. \text{set\_}$

$\text{index}(\text{"ID"})$

$(\text{good} = \text{pd.concat}([x_{\text{new}}, x_{\text{new1}}], \text{axis=1}),$

Merge :  $\rightarrow$  inner  $\rightarrow$  left  $\rightarrow$  outer

$\text{inner} = X(y, \text{on} = \text{"id"}, \text{how} = \text{"inner"})$

$\text{left} = X(y, \text{on} = \text{"left"})$

$\text{outer} = X(y, \text{on} = \text{"outer"})$

$(\text{left}, \text{right}, \text{on}) \rightarrow \text{indicator=True}$

all full

current on  
count add 25,

# overlapping column name → suffixes

("ID") → ID → ID + part 1 → age → age - part 1

df\_a = pd.DataFrame({ 'ID': [1, 2, 3], 'age': [21, 22, 30] })

df\_b = pd.DataFrame({ 'ID': [1, 2, 3], 'age': [25, 26, 27] })

munge = df\_a.merge(df\_b, on='ID',

left\_on=..., right\_on=..., suffixes=("-part1", "-part2"))

		age-part1	age-part2
0	1	21	25
1	2	22	26
2	3	30	27

④ from collections import Counter

Counters ( df.temperature )

↳ count → value of int.

# left-on , right-on



<u>X</u>	<u>id</u>	<u>age</u>	<u>bp</u>
0	1	23	120
1	2	45	140
2	3	31	130
3	4	35	128

visits

	<u>patient-id</u>	<u>visit-count</u>
0	1	3
1	2	1
2	2	2
3	4	5

join = X.menge(visits, left-on = "id",  
right-on = "patient-id", how = left)

	<u>id</u>	<u>age</u>	<u>bp</u>	<u>pat</u>	<u>vis</u>
0	1	23	120	1	3
1	2	45	140	2	1
2	3	31	130	2	2
3	4	35	128	None	5