

① convert time series operations into datetime object

df = pd.read\_csv("weather.csv", parse\_dates=["day"])

② df.set\_index("day") → index set करा ③ df.reset\_index(inplace=True)

④ df.isna().sum()

⑤ Given column शरीर change करा

⑥ fill NaN value with 0

df.temp = pd.Series([5, 6, np.nan, 3])

tf = df.fillna(0)

⑦ tf = df.fillna({

"temp": 0,

"wind": 1,

"even": "NO"

})

⑧ नये value द्या जाण value fill करा →

tf = df.fillna(method="bfill")

⑨ set limit

tf = df.fillna(method="bfill", limit=3)

⑩ जाण value द्या fill करा →

↳ forward fill → all

tf = df.fillna(method="ffill")

⑪ limit द्या करीर fill करा →

tf = df.fillna(method="ffill", limit=1)

• 1st nan → always NaN  
करा

⑫ Normal interpolate → value द्या करीर  
observation

df.interpolate()

parse-date

⑬ Time interpolation

↓  
time interpolation दिता  
करा शत index कर  
Datetime 20 270

tf = df.set\_index("day")  
gf = tf.interpolate  
(method='time')

[14] drop rows that contain any missing value  
↳ bad

```
tf = df.dropna()
```

len(tf) → row size.

[15] drop rows if all value are missing

```
tf = df.dropna(how = "all")
```

[16] minimum row value ko shamil drop kr

↳ must 3 se row shamil ho.

```
tf = df.dropna(thresh = 3)
```

→ minimum 3 se  
shamil ho.

[17] ty. dtypes.

[18] replace kr.

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

[19] multiple value ko replace kr.

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

[20]

```
of = df.replace(
```

```
{ 'temp' : -9,
```

```
  'wind' : -8,
```

```
  'even' : '0' },
```

```
np.nan)
```

[21] જરૂર નહીં હોય તો 'no event' કે 'sunny' થાય

```
df = df.replace(
```

```
{ -1 : np.nan,
```

```
"no event" : 'sunny'
```

```
}
```

```
)
```

[22] Regex → extra unit (F, amp) વાદ પડે તો

'[A-Za-z]' → A to z or a to z

```
df = df.replace(
```

```
{ 'temp' : r "[A-Za-z]",
```

```
  'wind' : r "[A-Za-z]"
```

```
}, "", regex = True
```

```
)
```

↳ space વાદ પડે તો & numeric બનાવો

```
df["temp"] = pd.to_numeric(df["temp"],
```

```
  astype(str).str.strip(), errors =
```

```
  "coerce")
```

coerce → ના પૂર્ણ નાન કરે

②③ replace list with another list,

```
grades = pd.DataFrame({
```

```
    "score": ['exceptional', 'average', 'good', 'poor', 'average'],
```

```
    "student": ['nob', 'maya', 'po', 'tom', 'julia', 'eric']
```

```
})
```

→

<u>score</u>	<u>student</u>
exceptional	nob
average	maya
good	po
poor	tom
average	julia

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

```
    ["poor", "average", "good", "exp"],
```

```
    [1, 2, 3, 4]
```

```
)
```

→

<u>score</u>	<u>student</u>
1	nob
2	maya
3	po
4	tom
3	julia



concat & Merge : →

```
x_train = pd.DataFrame(  
    {  
        'ID': [1, 2, 3],  
        'age': [20, 21, 22],  
        'bp': [120, 100, 90],  
    }  
)
```

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

```
x_test = pd.DataFrame(  
    {  
        'ID': [5, 6],  
        'age': [20, 25],  
        'bp': [100, 90],  
    }  
)
```

	ID	age	bp
0	5	20	100
1	6	25	90

```
y_train = pd.DataFrame(  
    {  
        'ID': [1, 2, 3, 4],  
        'target': [0, 1, 0, 1],  
    }  
)
```

	ID	target
0	1	0
1	2	1
2	3	0
3	4	1

① concat vertically (axis=0) -

$X_{all} = pd.concat([X_{train}, X_{test}], axis=0)$

↳

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

$X_{all} = pd.concat([X_{train}, X_{test}], axis=0, ignore_index=True)$

↳

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

$X_{all} = pd.concat([X_{train}, X_{test}],$   
~~keys =~~ keys = ["train", "test"])

↳

train	0	1	2	3	4
test	5	6			

• access train block

$X_{all}.loc["train"]$

↳

train	0	1	2	3
-------	---	---	---	---

concat variable : →

```
X-all = pd.concat([X-train, X-test],  
axis = 1)
```

13 row गुला shuffle करवा फंग index को मरो  
(मरो)

```
X-all-new = X-all.sample(frac=1,  
random_state=0).reset_index(drop=  
True)
```

14 ID show n error index version cancel  
→ Bad

```
bad-with = pd.concat([X-train, X-test],  
axis = 1)
```



①5 ID show ~~add~~ add

`x_new = x_train.set_index("id")`

`x_new1 = x_new1.assign {`

`id = x_train["ID"].set-`

`index("ID")`

`id = x_train["ID"].set-`

`good = pd.concat([x_new, x_new1], axis=1)`

`x.merge()`

Merge :  $\rightarrow$

merge

`inner = x(y, on = "id", how = "inner")`

`left = x(y, on = "id", how = "left")`

`outer = x(y, on = "id", how = "outer")`

$\rightarrow$  all full

`indicator=True`

$\rightarrow$  check on  
column add 25,

# overlapping column name → suffixes

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

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

```
merge = df_a.merge(df_b, on="ID",
```

```
suffixes = ("part1", "part2"))
```



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

④ from collections import Counter

Counter (df['temperature'])

↳ count each value of (temp)

# 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.merge(visits, left\_on="id",

right\_on="patient\_id", how="left")

	<u>id</u>	<u>age</u>	<u>bp</u>	<u>pat</u>	<u>vi</u>
0	1	23	120	1	3
1	2	45	140	2	1
2	2	45	140	2	2
3	4	35	128	4	5

Howe =