Pandas

Pandas is a Python open-source library that provides various datastructures to perform effective operations on the data.

Pandas provides two data structures such as:

- Series
- DataFrame

```
import numpy as np
import pandas as pd
```

```
"""
  A Pandas Series is like a column in a table.
  It is a one-dimensional array holding data of any type.
"""

series = pd.Series([1,2,3,4,5])  #creating a pandas series using python list
series
```

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

```
a = np.arange(1, 10) #creating a numpy array
a
```

```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
ser = pd.Series() #creating an empty pandas series
ser
```

```
Series([], dtype: object)
```

```
ser1 = pd.Series(a) #creating a pandas series using numpy array
ser1
```

```
0    1
1    2
2    3
3    4
4    5
5    6
6    7
7    8
8    9
dtype: int64
```

```
ser1[8] #accesing elements using index
```

```
9
```

```
ser1[5]
```

```
6
```

```
ser2 = pd.Series(a, index=['a','b','c','d','e','f','g','h','i']) #creating a pandas series using numpy array with user-defined indexes
ser2
```

```
a    1
b    2
c    3
d    4
e    5
f    6
g    7
h    8
i    9
dtype: int64
```

```
ser2['a']
```

```
1
```

```
ser2[0]
```

```
1
```

```
ser2['e'] = 10 #re-initializing item using indesing
ser2
```

```
a     1
b     2
c     3
d     4
e    10
f     6
g     7
h     8
i     9
dtype: int64
```

```
ser3 = pd.Series(4, index=['a','b','c']) #creating a pandas series using a scalar value
ser3
```

```
a    4
b    4
c    4
dtype: int64
```

```
ser3['c']
```

```
4
```

```
"""
  A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.
  A DataFrame is a collection of Pandas Series objects, all sharing the same index.
"""
```

```
l = [1,2,3,4,5]
df = pd.DataFrame(l)
df
```

	0
0	1
1	2
2	3
3	4
4	5

```
weather_data = {
    'day': ['1/1/2017','1/2/2017','1/3/2017','1/4/2017','1/5/2017','1/6/2017'],
    'temperature': [32,35,28,24,32,31],
    'windspeed': [6,7,2,7,4,2],
    'event': ['Rain', 'Sunny', 'Snow','Snow','Rain', 'Sunny']
}
df = pd.DataFrame(weather_data)
df
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow
3	1/4/2017	24	7	Snow
4	1/5/2017	32	4	Rain
5	1/6/2017	31	2	Sunny

```
df.shape
```

```
(6, 4)
```

```
df1 = pd.read_csv("/content/weather_data.csv")
df1
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow
3	1/4/2017	24	7	Snow
4	1/5/2017	32	4	Rain
5	1/6/2017	31	2	Sunny

```
df1.shape
```

```
(6, 4)
```

```
df1.head(2)
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny

```
df1.tail(2)
```

	day	temperature	windspeed	event
4	1/5/2017	32	4	Rain
5	1/6/2017	31	2	Sunny

```
df1.describe()
```

	temperature	windspeed
count	6.000000	6.000000
mean	30.333333	4.666667
std	3.829708	2.338090
min	24.000000	2.000000
25%	28.750000	2.500000
50%	31.500000	5.000000
75%	32.000000	6.750000
max	35.000000	7.000000

```
df1[2:4]
```

	day	temperature	windspeed	event
2	1/3/2017	28	2	Snow
3	1/4/2017	24	7	Snow

```
df.event
```

```
0     Rain
1    Sunny
2     Snow
3     Snow
4     Rain
5    Sunny
Name: event, dtype: object
```

```
type(df.event)
```

```
pandas.core.series.Series
def __init__(data=None, index=None, dtype: Dtype | None=None, name=None, copy: bool |
None=None, fastpath: bool=False) -> None

One-dimensional ndarray with axis labels (including time series).

Labels need not be unique but must be a hashable type. The object
supports both integer- and label-based indexing and provides a host of
methods for performing operations involving the index. Statistical
methods from ndarray have been overridden to automatically exclude
```

```
df[['day','event']]
```

	day	event
0	1/1/2017	Rain
1	1/2/2017	Sunny
2	1/3/2017	Snow
3	1/4/2017	Snow
4	1/5/2017	Rain
5	1/6/2017	Sunny

```
df.temperature
```

```
0    32
1    35
2    28
3    24
4    32
5    31
Name: temperature, dtype: int64
```

```
df.temperature.max()
```

```
35
```

```
df['day'].min()
```

```
'1/1/2017'
```

```
df['windspeed'].mean()
```

```
4.666666666666667
```

```
df['windspeed'].std()
```

```
2.3380903889000244
```

```
df['windspeed'].median()
```

```
5.0
```

```
df['windspeed'].sum()
```

```
28
```

```
df['windspeed'].count()
```

```
6
```

```
df.index
```

```
RangeIndex(start=0, stop=6, step=1)
```

```
df.iloc[2]
```

```
day            1/3/2017
temperature          28
windspeed             2
event              Snow
Name: 2, dtype: object
```

```
df.set_index('day', inplace=True)
df
```

day	temperature	windspeed	event
1/1/2017	32	6	Rain
1/2/2017	35	7	Sunny
1/3/2017	28	2	Snow
1/4/2017	24	7	Snow
1/5/2017	32	4	Rain
1/6/2017	31	2	Sunny

```
df.loc['1/3/2017']
```

```
temperature      28
windspeed         2
event          Snow
Name: 1/3/2017, dtype: object
```

```
df.reset_index(inplace=True)
df
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow
3	1/4/2017	24	7	Snow
4	1/5/2017	32	4	Rain
5	1/6/2017	31	2	Sunny

```
data = pd.read_csv("/content/nyc_weather.csv")
data.head()
```

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedMPH	Precip
0	1/1/2016	38	23	52	30.03	10	8.0	
1	1/2/2016	36	18	46	30.02	10	7.0	
2	1/3/2016	40	21	47	29.86	10	8.0	
3	1/4/2016	25	9	44	30.05	10	9.0	

```
data['Temperature'].max()
```

```
50
```

```
r = data["Events"] == "Rain"
r
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8      True
9      True
10    False
11    False
12    False
13    False
14    False
15     True
16    False
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
26     True
27    False
28    False
29    False
30    False
Name: Events, dtype: bool
```

```
data["EST"][r]
```

```
8      1/9/2016
9     1/10/2016
15    1/16/2016
26    1/27/2016
Name: EST, dtype: object
```

```
data.fillna(0, inplace = True) #replacig null values with 0
data.head()
```

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedMPH	Precip
0	1/1/2016	38	23	52	30.03	10	8.0	
1	1/2/2016	36	18	46	30.02	10	7.0	
2	1/3/2016	40	21	47	29.86	10	8.0	
3	1/4/2016	25	9	44	30.05	10	9.0	

```
data["WindSpeedMPH"].mean()
```

```
6.225806451612903
```

Data Munging or Data Wrangling: The process of cleaning data is called as data munging or data wrangling.

```
#creating dataframe using dictionaries
weather_data = {
    'day': ['1/1/2017','1/2/2017','1/3/2017'],
    'temperature': [32,35,28],
    'windspeed': [6,7,2],
    'event': ['Rain', 'Sunny', 'Snow']
}
df = pd.DataFrame(weather_data)
df
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow

```
#creating dataframe using tuples
weather_data = [
    ('1/1/2017',32,6,'Rain'),
    ('1/2/2017',35,7,'Sunny'),
    ('1/3/2017',28,2,'Snow')
]
df = pd.DataFrame(data=weather_data, columns=['day','temperature','windspeed','event'])
df
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow

```
#creating dataframe from a set of dictionaries
weather_data = [
    {'day': '1/1/2017', 'temperature': 32, 'windspeed': 6, 'event': 'Rain'},
    {'day': '1/2/2017', 'temperature': 35, 'windspeed': 7, 'event': 'Sunny'},
    {'day': '1/3/2017', 'temperature': 28, 'windspeed': 2, 'event': 'Snow'},

]
df = pd.DataFrame(data=weather_data, columns=['day','temperature','windspeed','event'])
df
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow

```
#creating dataframe from csv file
df = pd.read_csv("/content/weather_data.csv")
df
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	35	7	Sunny
2	1/3/2017	28	2	Snow
3	1/4/2017	24	7	Snow
4	1/5/2017	32	4	Rain
5	1/6/2017	31	2	Sunny

```
#creating dataframe from excel file
df = pd.read_excel("/content/weather_data.xlsx", "Sheet1")
df
```

	day	temperature	windspeed	event
0	2017-01-01	32	6	Rain
1	2017-01-02	35	7	Sunny
2	2017-01-03	28	2	Snow

```
#reading and writing to csv file
df = pd.read_csv("/content/stock_data.csv")
df
```

	tickers	eps	revenue	price	people
0	GOOGL	27.82	87	845	larry page
1	WMT	4.61	484	65	n.a.
2	MSFT	-1	85	64	bill gates
3	RIL	not available	50	1023	mukesh ambani
4	TATA	5.6	-1	n.a.	ratan tata

```
df = pd.read_csv("/content/stock_data.csv", skiprows = 1)
df
```

	GOOGL	27.82	87	845	larry page
0	WMT	4.61	484	65	n.a.
1	MSFT	-1	85	64	bill gates
2	RIL	not available	50	1023	mukesh ambani
3	TATA	5.6	-1	n.a.	ratan tata

```
df = pd.read_csv("/content/stock_data.csv", header = None)
df
```

	0	1	2	3	4
0	tickers	eps	revenue	price	people
1	GOOGL	27.82	87	845	larry page
2	WMT	4.61	484	65	n.a.
3	MSFT	-1	85	64	bill gates
4	RIL	not available	50	1023	mukesh ambani
5	TATA	5.6	-1	n.a.	ratan tata

```
df = pd.read_csv("/content/stock_data.csv", nrows = 4)
df
```

	tickers	eps	revenue	price	people
0	GOOGL	27.82	87	845	larry page
1	WMT	4.61	484	65	n.a.
2	MSFT	-1	85	64	bill gates
3	RIL	not available	50	1023	mukesh ambani

```
df = pd.read_csv("/content/stock_data.csv", na_values = ["not available","n.a."])
df
```

	tickers	eps	revenue	price	people
0	GOOGL	27.82	87	845.0	larry page
1	WMT	4.61	484	65.0	NaN
2	MSFT	-1.00	85	64.0	bill gates
3	RIL	NaN	50	1023.0	mukesh ambani
4	TATA	5.60	-1	NaN	ratan tata

```
df = pd.read_csv("stock_data.csv",  na_values={
        'eps': ['not available'],
        'revenue': [-1],
        'people': ['not available','n.a.']
    })
df
```

	tickers	eps	revenue	price	people
0	GOOGL	27.82	87.0	845	larry page
1	WMT	4.61	484.0	65	NaN
2	MSFT	-1.00	85.0	64	bill gates
3	RIL	NaN	50.0	1023	mukesh ambani
4	TATA	5.60	NaN	n.a.	ratan tata

```
#writing to csv
df.to_csv("new.csv", index = False)
```

```
df2 = pd.read_csv("new.csv")
df2
```

	tickers	eps	revenue	price	people
0	GOOGL	27.82	87.0	845	larry page
1	WMT	4.61	484.0	65	NaN
2	MSFT	-1.00	85.0	64	bill gates
3	RIL	NaN	50.0	1023	mukesh ambani
4	TATA	5.60	NaN	n.a.	ratan tata

```
df.to_csv('new1.csv', columns=['tickers','people'])
```

```
df3 =pd.read_csv("new1.csv")
df3
```

	Unnamed: 0	tickers	people
0	0	GOOGL	larry page
1	1	WMT	NaN
2	2	MSFT	bill gates
3	3	RIL	mukesh ambani
4	4	TATA	ratan tata

```
df3.drop('Unnamed: 0', axis = 1, inplace = True)
df3
```

	tickers	people
0	GOOGL	larry page
1	WMT	NaN
2	MSFT	bill gates
3	RIL	mukesh ambani
4	TATA	ratan tata

```
#reading excel file
df4 = pd.read_excel("/content/stock_data.xlsx", "Sheet1")
df4
```

	tickers	eps	revenue	price	people
0	GOOGL	27.82	87	845	larry page
1	WMT	4.61	484	65	n.a.
2	MSFT	-1	85	64	bill gates
3	RIL	not available	50	1023	mukesh ambani
4	TATA	5.6	-1	n.a.	ratan tata

```
#writing to excel file
df4.to_excel("new.xlsx", sheet_name = "stocks")
```

```
df5 = pd.read_excel("new.xlsx", "stocks")
df5
```

	Unnamed: 0	tickers	eps	revenue	price	people
0	0	GOOGL	27.82	87	845	larry page
1	1	WMT	4.61	484	65	n.a.
2	2	MSFT	-1	85	64	bill gates
3	3	RIL	not available	50	1023	mukesh ambani
4	4	TATA	5.6	-1	n.a.	ratan tata

```
df4.to_excel("new1.xlsx", sheet_name = "stocks", index = False)
```

```
df6 = pd.read_excel("new1.xlsx", "stocks")
df6
```

	tickers	eps	revenue	price	people
0	GOOGL	27.82	87	845	larry page
1	WMT	4.61	484	65	n.a.
2	MSFT	-1	85	64	bill gates
3	RIL	not available	50	1023	mukesh ambani
4	TATA	5.6	-1	n.a.	ratan tata

```
df7 = pd.read_csv('weather_data (1).csv')
df7
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	NaN	9.0	Sunny
2	1/5/2017	28.0	NaN	Snow
3	1/6/2017	NaN	7.0	NaN
4	1/7/2017	32.0	NaN	Rain
5	1/8/2017	NaN	NaN	Sunny
6	1/9/2017	NaN	NaN	NaN
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df8 = df7.fillna(df7.temperature.median())
df8
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	32.0	9.0	Sunny
2	1/5/2017	28.0	32.0	Snow
3	1/6/2017	32.0	7.0	32.0
4	1/7/2017	32.0	32.0	Rain
5	1/8/2017	32.0	32.0	Sunny
6	1/9/2017	32.0	32.0	32.0
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df9 = df7.fillna(method = 'ffill') #null values will be replaced with the last observed non-null value
df9
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	32.0	9.0	Sunny
2	1/5/2017	28.0	9.0	Snow
3	1/6/2017	28.0	7.0	Snow
4	1/7/2017	32.0	7.0	Rain
5	1/8/2017	32.0	7.0	Sunny
6	1/9/2017	32.0	7.0	Sunny
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
#different values for different column null values
df10 = df7.fillna({
    'temperature': df7.temperature.median(),
    'windspeed': df7.windspeed.median(),
    'event' : 'no event'
})
df10
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	32.0	9.0	Sunny
2	1/5/2017	28.0	8.0	Snow
3	1/6/2017	32.0	7.0	no event
4	1/7/2017	32.0	8.0	Rain
5	1/8/2017	32.0	8.0	Sunny
6	1/9/2017	32.0	8.0	no event
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df11 = df7.fillna(method = 'bfill') #null values will be replaced with the next observed non-null value
df11
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	28.0	9.0	Sunny
2	1/5/2017	28.0	7.0	Snow
3	1/6/2017	32.0	7.0	Rain
4	1/7/2017	32.0	8.0	Rain
5	1/8/2017	34.0	8.0	Sunny
6	1/9/2017	34.0	8.0	Cloudy
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df7
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	NaN	9.0	Sunny
2	1/5/2017	28.0	NaN	Snow
3	1/6/2017	NaN	7.0	NaN
4	1/7/2017	32.0	NaN	Rain
5	1/8/2017	NaN	NaN	Sunny
6	1/9/2017	NaN	NaN	NaN
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df7.fillna(method = 'ffill', inplace = True) #using inplace = True modifies the existing dataframe
                                             #by default inplace = False, which returns a new modified dataframe
df7
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	32.0	9.0	Sunny
2	1/5/2017	28.0	9.0	Snow
3	1/6/2017	28.0	7.0	Snow
4	1/7/2017	32.0	7.0	Rain
5	1/8/2017	32.0	7.0	Sunny
6	1/9/2017	32.0	7.0	Sunny
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df12 = df7.fillna(method='bfill', limit = 1)
df12
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/4/2017	28.0	9.0	Sunny
2	1/5/2017	28.0	7.0	Snow
3	1/6/2017	32.0	7.0	Rain
4	1/7/2017	32.0	NaN	Rain
5	1/8/2017	NaN	NaN	Sunny
6	1/9/2017	34.0	8.0	Cloudy
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df13 = df7.interpolate()
df13
```

	day	temperature	windspeed	event
0	1/1/2017	32.000000	6.00	Rain
1	1/4/2017	30.000000	9.00	Sunny
2	1/5/2017	28.000000	8.00	Snow
3	1/6/2017	30.000000	7.00	NaN
4	1/7/2017	32.000000	7.25	Rain
5	1/8/2017	32.666667	7.50	Sunny
6	1/9/2017	33.333333	7.75	NaN
7	1/10/2017	34.000000	8.00	Cloudy
8	1/11/2017	40.000000	12.00	Sunny

```
df14 = df7.dropna()
df14
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
7	1/10/2017	34.0	8.0	Cloudy
8	1/11/2017	40.0	12.0	Sunny

```
df7.set_index('day', inplace = True)
df7
```

day	temperature	windspeed	event
1/1/2017	32.0	6.0	Rain
1/4/2017	NaN	9.0	Sunny
1/5/2017	28.0	NaN	Snow
1/6/2017	NaN	7.0	NaN
1/7/2017	32.0	NaN	Rain
1/8/2017	NaN	NaN	Sunny
1/9/2017	NaN	NaN	NaN
1/10/2017	34.0	8.0	Cloudy
1/11/2017	40.0	12.0	Sunny

```
df15 = df7.dropna(how="all")
df15
```

day	temperature	windspeed	event
1/1/2017	32.0	6.0	Rain
1/4/2017	NaN	9.0	Sunny
1/5/2017	28.0	NaN	Snow
1/6/2017	NaN	7.0	NaN
1/7/2017	32.0	NaN	Rain
1/8/2017	NaN	NaN	Sunny
1/10/2017	34.0	8.0	Cloudy
1/11/2017	40.0	12.0	Sunny

```
df16 = df7.dropna(thresh=1)
df16
```

day	temperature	windspeed	event
1/1/2017	32.0	6.0	Rain
1/4/2017	NaN	9.0	Sunny
1/5/2017	28.0	NaN	Snow
1/6/2017	NaN	7.0	NaN
1/7/2017	32.0	NaN	Rain
1/8/2017	NaN	NaN	Sunny
1/10/2017	34.0	8.0	Cloudy
1/11/2017	40.0	12.0	Sunny

```
df17 = df7.dropna(thresh=2)
df17
```

day	temperature	windspeed	event
1/1/2017	32.0	6.0	Rain
1/4/2017	NaN	9.0	Sunny
1/5/2017	28.0	NaN	Snow
1/7/2017	32.0	NaN	Rain
1/10/2017	34.0	8.0	Cloudy
1/11/2017	40.0	12.0	Sunny

```
import numpy as np
import pandas as pd
```

```
df = pd.read_csv('weather_data (2).csv')
df
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	-99999	7	Sunny
2	1/3/2017	28	-99999	Snow
3	1/4/2017	-99999	7	0
4	1/5/2017	32	-99999	Rain
5	1/6/2017	31	2	Sunny
6	1/6/2017	34	5	0

Next steps: Generate code with df View recommended plots

```
df['temperature'][1] = -88888
df
```

```
<ipython-input-14-4e0e231e82f0>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
  df['temperature'][1] = -88888
```

	day	temperature	windspeed	event
0	1/1/2017	32	6	Rain
1	1/2/2017	-88888	7	Sunny
2	1/3/2017	28	-99999	Snow
3	1/4/2017	-99999	7	0
4	1/5/2017	32	-99999	Rain
5	1/6/2017	31	2	Sunny
6	1/6/2017	34	5	0

Next steps: Generate code with df View recommended plots

```
new_df = df.replace(-99999,np.NaN)
new_df
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/2/2017	-88888.0	7.0	Sunny
2	1/3/2017	28.0	NaN	Snow
3	1/4/2017	NaN	7.0	0
4	1/5/2017	32.0	NaN	Rain
5	1/6/2017	31.0	2.0	Sunny
6	1/6/2017	34.0	5.0	0

Next steps: Generate code with new_df View recommended plots

```
new_df1 = df.replace([-99999,-88888],np.NaN)
new_df1
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/2/2017	NaN	7.0	Sunny
2	1/3/2017	28.0	NaN	Snow
3	1/4/2017	NaN	7.0	0
4	1/5/2017	32.0	NaN	Rain
5	1/6/2017	31.0	2.0	Sunny
6	1/6/2017	34.0	5.0	0

Next steps: Generate code with new_df1 View recommended plots

```
new_df2 = df.replace(
    {
        'temperature' : [-99999, -88888],
        'windspeed' : -99999,
    }, np.NaN
)
new_df2
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/2/2017	NaN	7.0	Sunny
2	1/3/2017	28.0	NaN	Snow
3	1/4/2017	NaN	7.0	0
4	1/5/2017	32.0	NaN	Rain
5	1/6/2017	31.0	2.0	Sunny
6	1/6/2017	34.0	5.0	0

Next steps: Generate code with new_df2 View recommended plots

```
new_df3 = df.replace(
    {
        -99999 : np.NaN,
        -88888 : np.NaN,
        '0' : "No Event"
    }
)
new_df3
```

	day	temperature	windspeed	event
0	1/1/2017	32.0	6.0	Rain
1	1/2/2017	NaN	7.0	Sunny
2	1/3/2017	28.0	NaN	Snow
3	1/4/2017	NaN	7.0	No Event
4	1/5/2017	32.0	NaN	Rain
5	1/6/2017	31.0	2.0	Sunny
6	1/6/2017	34.0	5.0	No Event

Next steps: Generate code with new_df3 View recommended plots

```
df = pd.DataFrame({
    'score': ['exceptional','average', 'good', 'poor', 'average', 'exceptional'],
    'student': ['rob', 'maya', 'parthiv', 'tom', 'julian', 'erica']
})
df
```

	score	student
0	exceptional	rob
1	average	maya
2	good	parthiv
3	poor	tom
4	average	julian
5	exceptional	erica

Next steps: Generate code with df View recommended plots

```
df.replace(['exceptional','good','average','poor'],[4,3,2,1])
```

	score	student
0	4	rob
1	2	maya
2	3	parthiv
3	1	tom
4	2	julian
5	4	erica

```
df = pd.read_csv('weather_by_cities.csv')
df
```

	day	city	temperature	windspeed	event
0	1/1/2017	new york	32	6	Rain
1	1/2/2017	new york	36	7	Sunny
2	1/3/2017	new york	28	12	Snow
3	1/4/2017	new york	33	7	Sunny
4	1/1/2017	mumbai	90	5	Sunny
5	1/2/2017	mumbai	85	12	Fog
6	1/3/2017	mumbai	87	15	Fog
7	1/4/2017	mumbai	92	5	Rain
8	1/1/2017	paris	45	20	Sunny
9	1/2/2017	paris	50	13	Cloudy
10	1/3/2017	paris	54	8	Cloudy
11	1/4/2017	paris	42	10	Cloudy

Next steps: Generate code with df View recommended plots

```
df1 = df.groupby('city')
df1
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x788d3f7bf3a0>
```

```
for city, data in df1:
  print(city)
  print(data)
```

```
mumbai
        day    city  temperature  windspeed  event
4  1/1/2017  mumbai           90          5  Sunny
5  1/2/2017  mumbai           85         12    Fog
6  1/3/2017  mumbai           87         15    Fog
7  1/4/2017  mumbai           92          5   Rain
new york
        day      city  temperature  windspeed  event
0  1/1/2017  new york           32          6   Rain
1  1/2/2017  new york           36          7  Sunny
2  1/3/2017  new york           28         12   Snow
3  1/4/2017  new york           33          7  Sunny
paris
         day   city  temperature  windspeed   event
8   1/1/2017  paris           45         20   Sunny
9   1/2/2017  paris           50         13  Cloudy
10  1/3/2017  paris           54          8  Cloudy
11  1/4/2017  paris           42         10  Cloudy
```

```
df1.get_group('paris')
```

	day	city	temperature	windspeed	event
8	1/1/2017	paris	45	20	Sunny
9	1/2/2017	paris	50	13	Cloudy
10	1/3/2017	paris	54	8	Cloudy
11	1/4/2017	paris	42	10	Cloudy

```
df1.max()
```

city	day	temperature	windspeed	event
mumbai	1/4/2017	92	15	Sunny
new york	1/4/2017	36	12	Sunny
paris	1/4/2017	54	20	Sunny

```
df1.describe()
```

	temperature								windspeed							
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
city																
mumbai	4.0	88.50	3.109126	85.0	86.50	88.5	90.50	92.0	4.0	9.25	5.057997	5.0	5.00	8.5	12.75	15.0
new york	4.0	32.25	3.304038	28.0	31.00	32.5	33.75	36.0	4.0	8.00	2.708013	6.0	6.75	7.0	8.25	12.0
paris	4.0	47.75	5.315073	42.0	44.25	47.5	51.00	54.0	4.0	12.75	5.251984	8.0	9.50	11.5	14.75	20.0

```
india_weather = pd.DataFrame({
    "city": ["mumbai","delhi","banglore"],
    "temperature": [32,45,30],
    "humidity": [80, 60, 78]
})
india_weather
```

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78

Next steps: Generate code with india_weather | View recommended plots

```
us_weather = pd.DataFrame({
    "city": ["new york","chicago","orlando"],
    "temperature": [21,14,35],
    "humidity": [68, 65, 75]
})
us_weather
```

	city	temperature	humidity
0	new york	21	68
1	chicago	14	65
2	orlando	35	75

Next steps: Generate code with us_weather | View recommended plots

```
combined_df = pd.concat([india_weather,us_weather])
combined_df
```

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78
0	new york	21	68
1	chicago	14	65
2	orlando	35	75

Next steps: Generate code with combined_df | View recommended plots

```
combined_df1 = pd.concat([india_weather, us_weather], ignore_index = True)
combined_df1
```

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78
3	new york	21	68
4	chicago	14	65
5	orlando	35	75

Next steps: Generate code with combined_df1 | View recommended plots

```
combined_df2 = pd.concat([india_weather, us_weather], keys = ['India', 'US'])
combined_df2
```

		city	temperature	humidity
India	0	mumbai	32	80
	1	delhi	45	60
	2	banglore	30	78
US	0	new york	21	68
	1	chicago	14	65
	2	orlando	35	75

Next steps: Generate code with combined_df2 View recommended plots

```
combined_df2.loc['India']
```

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78

```
temperature_df = pd.DataFrame({
    "city": ["mumbai","delhi","banglore"],
    "temperature": [32,45,30],
}, index=[0,1,2])
temperature_df
```

	city	temperature
0	mumbai	32
1	delhi	45
2	banglore	30

Next steps: Generate code with temperature_df View recommended plots

```
windspeed_df = pd.DataFrame({
    "city": ["delhi","mumbai"],
    "windspeed": [7,12],
}, index=[1,0])
windspeed_df
```

	city	windspeed
1	delhi	7
0	mumbai	12

Next steps: Generate code with windspeed_df View recommended plots

```
combined_df3 = pd.concat([temperature_df, windspeed_df], axis = 1)
combined_df3
```

	city	temperature	city	windspeed
0	mumbai	32	mumbai	12.0
1	delhi	45	delhi	7.0
2	banglore	30	NaN	NaN

Next steps: Generate code with combined_df3 View recommended plots

```
s = pd.Series(["Humid","Dry","Rain"], name="event")
s
```

```
0    Humid
1      Dry
2     Rain
Name: event, dtype: object
```

```
combined_df4 = pd.concat([temperature_df, s], axis = 1)
combined_df4
```

	city	temperature	event
0	mumbai	32	Humid
1	delhi	45	Dry
2	banglore	30	Rain

Next steps: Generate code with combined_df4 View recommended plots

```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando"],
    "temperature": [21,14,35],
})
df1
```

	city	temperature
0	new york	21
1	chicago	14
2	orlando	35

Next steps: Generate code with df1 View recommended plots

```
df2 = pd.DataFrame({
    "city": ["chicago","new york","orlando"],
    "humidity": [65,68,75],
})
df2
```

	city	humidity
0	chicago	65
1	new york	68
2	orlando	75

Next steps: Generate code with df2 View recommended plots

```
df3 = pd.merge(df1, df2, on="city")
df3
```

	city	temperature	humidity
0	new york	21	68
1	chicago	14	65
2	orlando	35	75

Next steps: Generate code with df3 View recommended plots

```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando", "baltimore"],
    "temperature": [21,14,35, 38],
})
df1
```

	city	temperature
0	new york	21
1	chicago	14
2	orlando	35
3	baltimore	38

Next steps: Generate code with df1 View recommended plots

```
df2 = pd.DataFrame({
    "city": ["chicago","new york","san diego"],
    "humidity": [65,68,71],
})
df2
```

	city	humidity
0	chicago	65
1	new york	68
2	san diego	71

Next steps: Generate code with df2 View recommended plots

```
df3=pd.merge(df1,df2,on="city",how="inner")
df3
```

	city	temperature	humidity
0	new york	21	68
1	chicago	14	65

Next steps: Generate code with df3 View recommended plots

```
df3=pd.merge(df1,df2,on="city",how="outer")
df3
```

	city	temperature	humidity
0	new york	21.0	68.0
1	chicago	14.0	65.0
2	orlando	35.0	NaN
3	baltimore	38.0	NaN
4	san diego	NaN	71.0

Next steps: Generate code with df3 View recommended plots

```
df3=pd.merge(df1,df2,on="city",how="right")
df3
```

	city	temperature	humidity
0	chicago	14.0	65
1	new york	21.0	68
2	san diego	NaN	71

Next steps: Generate code with df3 View recommended plots

```
df3=pd.merge(df1,df2,on="city",how="left")
df3
```

	city	temperature_x	humidity_x	temperature_y	humidity_y
0	new york	21	65	14.0	68.0
1	chicago	14	68	21.0	65.0
2	orlando	35	71	NaN	NaN
3	baltimore	38	75	NaN	NaN

Next steps: Generate code with df3 View recommended plots

```
df3=pd.merge(df1,df2,on="city",how="outer", indicator = True)
df3
```

	city	temperature	humidity	_merge
0	new york	21.0	68.0	both
1	chicago	14.0	65.0	both
2	orlando	35.0	NaN	left_only
3	baltimore	38.0	NaN	left_only
4	san diego	NaN	71.0	right_only

Next steps: Generate code with df3 View recommended plots

```
df1 = pd.DataFrame({
    "city": ["new york","chicago","orlando", "baltimore"],
    "temperature": [21,14,35,38],
    "humidity": [65,68,71, 75]
})
df1
```

	city	temperature	humidity
0	new york	21	65
1	chicago	14	68
2	orlando	35	71
3	baltimore	38	75

Next steps: Generate code with df1 View recommended plots

```
df2 = pd.DataFrame({
    "city": ["chicago","new york","san diego"],
    "temperature": [21,14,35],
    "humidity": [65,68,71]
})
df2
```

	city	temperature	humidity
0	chicago	21	65
1	new york	14	68
2	san diego	35	71

Next steps: Generate code with df2 View recommended plots

```
df3=pd.merge(df1,df2,on="city",how="outer", suffixes = ('_first', '_second'), indicator = True)
df3
```

	city	temperature_first	humidity_first	temperature_second	humidity_second	_merge
0	new york	21.0	65.0	14.0	68.0	both
1	chicago	14.0	68.0	21.0	65.0	both
2	orlando	35.0	71.0	NaN	NaN	left_only
3	baltimore	38.0	75.0	NaN	NaN	left_only
4	san diego	NaN	NaN	35.0	71.0	right_only

Next steps: Generate code with df3 View recommended plots

```
df = pd.read_csv('weather.csv')
df
```

	date	city	temperature	humidity
0	5/1/2017	new york	65	56
1	5/2/2017	new york	66	58
2	5/3/2017	new york	68	60
3	5/1/2017	mumbai	75	80
4	5/2/2017	mumbai	78	83
5	5/3/2017	mumbai	82	85
6	5/1/2017	beijing	80	26
7	5/2/2017	beijing	77	30
8	5/3/2017	beijing	79	35

Next steps: Generate code with df View recommended plots

```
df1 = df.pivot_table(index = 'city', columns = 'date')
df1
```