

In []:

```

1 Topics :
2     1. Fillna to fill missing values using different ways
3     2. interpolate to make a guess on missing values
4     3. dropna to drop the null values

```

In [5]:

```

1 import pandas as pd
2 df = pd.read_csv("weather_data.csv", parse_dates=['day'])
3 # type(df.day[0])
4 df.head()

```

Out[5]:

	day	temperature	windspeed	event
0	2017-01-01	32.0	6.0	Rain
1	2017-01-04	NaN	9.0	Sunny
2	2017-01-05	28.0	NaN	Snow
3	2017-01-06	NaN	7.0	NaN
4	2017-01-07	32.0	NaN	Rain

In [6]:

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

In [7]:

```
1 df.head()
```

Out[7]:

	temperature	windspeed	event
day			
2017-01-01	32.0	6.0	Rain
2017-01-04	NaN	9.0	Sunny
2017-01-05	28.0	NaN	Snow
2017-01-06	NaN	7.0	NaN
2017-01-07	32.0	NaN	Rain

In [9]:

```
1 df.isna().sum() # Checking for null values
```

Out[9]:

```
temperature    4
windspeed      4
event          2
dtype: int64
```

In [12]:

```
1 new_df = df.fillna(0) # Filling the null values with 0
2 new_df
```

Out[12]:

	temperature	windspeed	event
day			
2017-01-01	32.0	6.0	Rain
2017-01-04	0.0	9.0	Sunny
2017-01-05	28.0	0.0	Snow
2017-01-06	0.0	7.0	0
2017-01-07	32.0	0.0	Rain
2017-01-08	0.0	0.0	Sunny
2017-01-09	0.0	0.0	0
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

In [13]:

```
1 # Filling the specific values to the particular column
2
3 new_df = df.fillna({
4     'temperature' : 0,
5     'windspeed' :0,
6     'event' : 'No event'
7 })
8
9 new_df
```

Out[13]:

	temperature	windspeed	event
day			
2017-01-01	32.0	6.0	Rain
2017-01-04	0.0	9.0	Sunny
2017-01-05	28.0	0.0	Snow
2017-01-06	0.0	7.0	No event
2017-01-07	32.0	0.0	Rain
2017-01-08	0.0	0.0	Sunny
2017-01-09	0.0	0.0	No event
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

In [16]:

```
1 new_df = df.fillna(method='ffill', limit=1) # Forward fill and backward fill and can l
2 new_df
```

Out[16]:

	temperature	windspeed	event
day			
2017-01-01	32.0	6.0	Rain
2017-01-04	32.0	9.0	Sunny
2017-01-05	28.0	9.0	Snow
2017-01-06	28.0	7.0	Snow
2017-01-07	32.0	7.0	Rain
2017-01-08	32.0	NaN	Sunny
2017-01-09	NaN	NaN	Sunny
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

In [18]:

```
1 new_df = df.interpolate()  
2 new_df
```

Out[18]:

	temperature	windspeed	event
day			
2017-01-01	32.000000	6.00	Rain
2017-01-04	30.000000	9.00	Sunny
2017-01-05	28.000000	8.00	Snow
2017-01-06	30.000000	7.00	NaN
2017-01-07	32.000000	7.25	Rain
2017-01-08	32.666667	7.50	Sunny
2017-01-09	33.333333	7.75	NaN
2017-01-10	34.000000	8.00	Cloudy
2017-01-11	40.000000	12.00	Sunny

In [19]:

```
1 new_df = df.interpolate(method='time') # Considering the time it relates it b/w the d  
2 new_df
```

Out[19]:

	temperature	windspeed	event
day			
2017-01-01	32.000000	6.00	Rain
2017-01-04	29.000000	9.00	Sunny
2017-01-05	28.000000	8.00	Snow
2017-01-06	30.000000	7.00	NaN
2017-01-07	32.000000	7.25	Rain
2017-01-08	32.666667	7.50	Sunny
2017-01-09	33.333333	7.75	NaN
2017-01-10	34.000000	8.00	Cloudy
2017-01-11	40.000000	12.00	Sunny

In [21]:

```
1 new_df = df.dropna()  
2 new_df
```

Out[21]:

	temperature	windspeed	event
day			
2017-01-01	32.0	6.0	Rain
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

In [22]:

```
1 new_df = df.dropna(how='all') # how="all" drops the row which has all na's  
2 new_df
```

Out[22]:

	temperature	windspeed	event
day			
2017-01-01	32.0	6.0	Rain
2017-01-04	NaN	9.0	Sunny
2017-01-05	28.0	NaN	Snow
2017-01-06	NaN	7.0	NaN
2017-01-07	32.0	NaN	Rain
2017-01-08	NaN	NaN	Sunny
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny

In [23]:

```
1 new_df = df.dropna(thresh=1) # Keeping the row that has atleast one non NA value
2 new_df
```

Out[23]:

	temperature	windspeed	event
day			
2017-01-01	32.0	6.0	Rain
2017-01-04	NaN	9.0	Sunny
2017-01-05	28.0	NaN	Snow
2017-01-06	NaN	7.0	NaN
2017-01-07	32.0	NaN	Rain
2017-01-08	NaN	NaN	Sunny
2017-01-10	34.0	8.0	Cloudy
2017-01-11	40.0	12.0	Sunny