2024-1

# Multivariate Data Analysis

Youjin Shin
(yj.shinn@catholic.ac.kr)

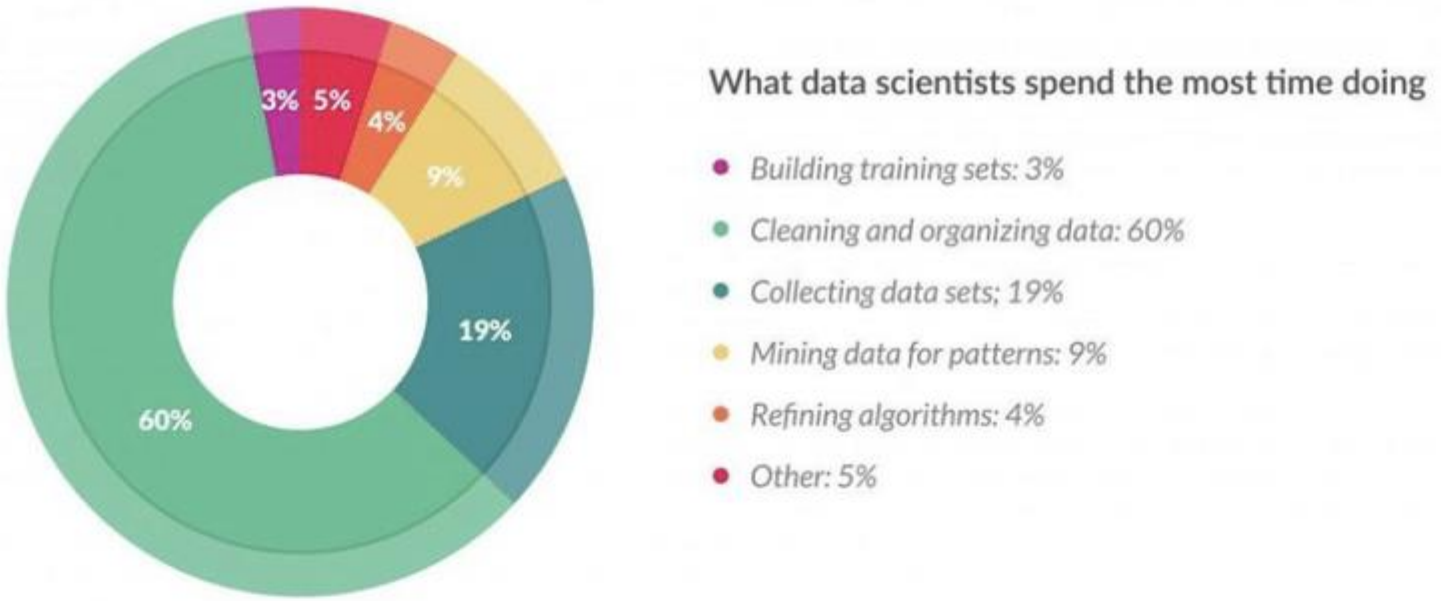
# Data Wrangling

## ◆ Data wrangling

- Data wrangling (Data manipulation) often referred to as data munging, is a critical step in the data science process. It involves cleaning, structuring, and enriching raw data into a desired format for better decision making in less time. This process is fundamental because data is rarely in a form that is immediately optimal for obtaining insights. Data wrangling makes data more usable and accessible.

# Data Wrangling

***Data preparation** accounts for about 80% of the work of data scientists*



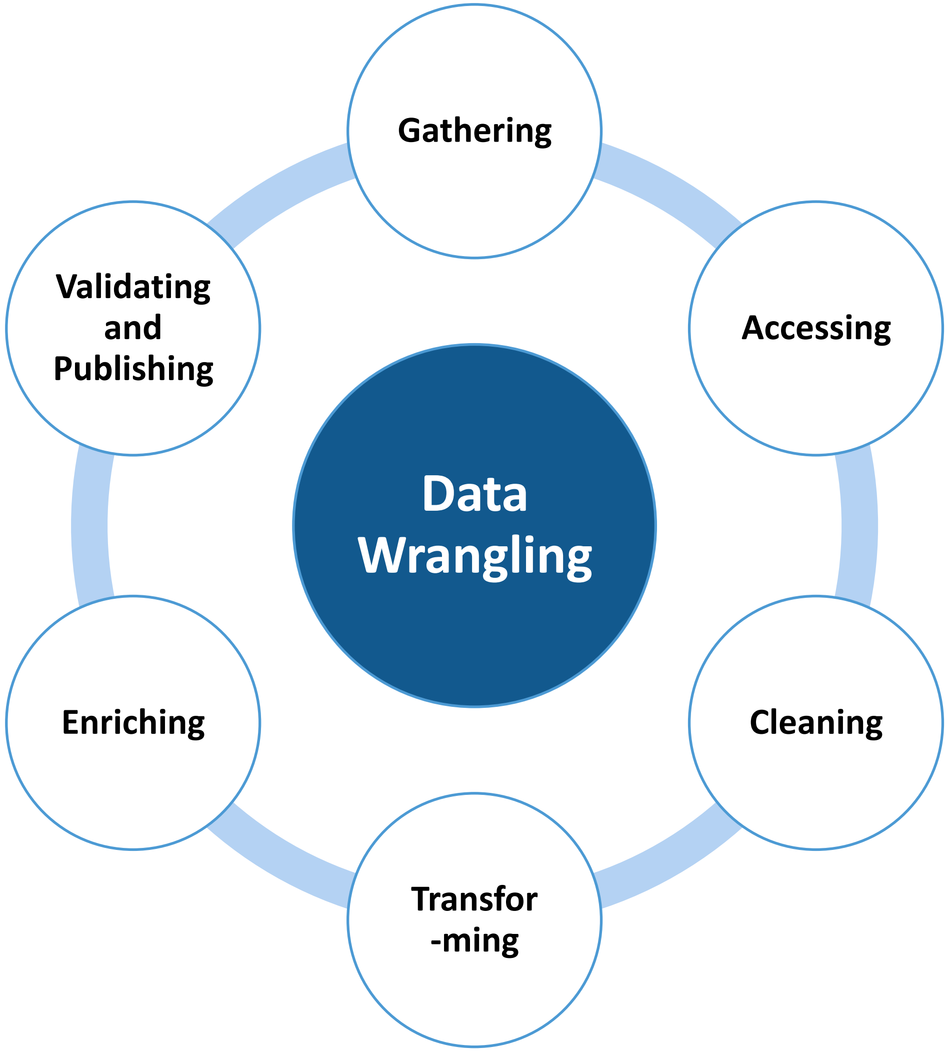
What data scientists spend the most time doing

- ● *Building training sets: 3%*
- ● *Cleaning and organizing data: 60%*
- ● *Collecting data sets; 19%*
- ● *Mining data for patterns: 9%*
- ● *Refining algorithms: 4%*
- ● *Other: 5%*

# Data Wrangling

## ◆ Key Steps in Data Wrangling

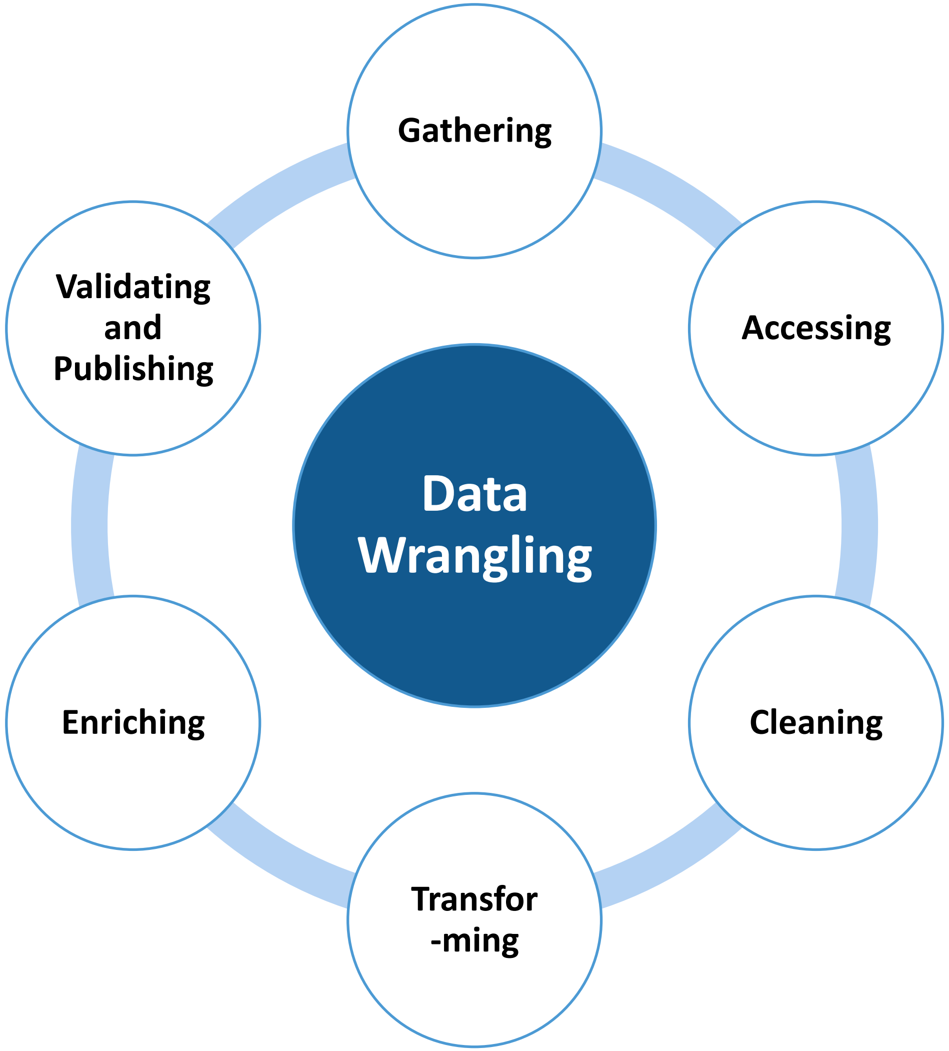
1. **Gathering Data**: Collecting data from various sources such as files, databases, web servers, APIs, and more.
2. **Assessing Data**: Evaluating the data for quality and structure issues. This includes identifying and diagnosing data inconsistencies, missing values, and the need for restructuring.
3. **Cleaning Data**: Addressing the issues identified in the assessment phase. This involves correcting, imputing, or removing erroneous data. Techniques include deduplication, filling in missing values, correcting errors, and more.



# Data Wrangling

## ◆ Key Steps in Data Wrangling

4. **Transforming Data**: Changing the format or structure of the data to make it more suitable for analysis. This could involve normalizing data, aggregating information, pivoting tables, and more.
5. **Enriching Data**: Enhancing data by merging it with other data sources, adding derived attributes, or performing calculations that make the data more informative.
6. **Validating and Publishing Data**: Ensuring that the data meets the criteria and standards for the analytics purposes. This could involve consistency checks, quality assurance, and finally making the data available for analysis or reporting.



# Lab: Data Wrangling

- **NYC weather data**
  - we will using NYC daily weather data that was taken from the National Centers for Environmental Information (NCEI) API
  - 1-Year data, 1 station gathers the data once a day

- **Data meanings:**
  - PRCP: precipitation in millimeters
  - SNOW: snowfall in millimeters
  - SNWD: snow depth in millimeters
  - TMAX: maximum daily temperature in Celsius
  - TMIN: minimum daily temperature in Celsius
  - TOBS: temperature at time of observation in Celsius
  - WESF: water equivalent of snow in millimeters

- **Some important facts to get our bearings:**
  - According to the National Weather Service, the coldest temperature ever recorded in Central Park was -15°F (-26.1°C) on February 9, 1934: source
  - The temperature of the Sun's photosphere is approximately 5,505°C: source

# Lab: Data Wrangling

```
# Understanding data -  head()

import pandas as pd
import numpy as np

pd.set_option('display.max_columns', None)

df = pd.read_csv('data/MDA_04_dirty_data.csv')
print(df.head())
```

```
 date                           station             PRCP   SNOW   SNWD   TMAX    TMIN    TOBS   WESF  inclement_weather
0  2018-01-01T00:00:00              ?                0.0    0.0   -inf   5505.0  -40.0   NaN    NaN       NaN
1  2018-01-01T00:00:00              ?                0.0    0.0   -inf   5505.0  -40.0   NaN    NaN       NaN
2  2018-01-01T00:00:00              ?                0.0    0.0   -inf   5505.0  -40.0   NaN    NaN       NaN
3  2018-01-02T00:00:00  GHCND:USC00280907            0.0    0.0   -inf   -8.3    -16.1   -12.2  NaN       False
4  2018-01-03T00:00:00  GHCND:USC00280907            0.0    0.0   -inf   -4.4    -13.9   -13.3  NaN       False
```

# Lab: Data Wrangling

```
# Understanding data -  head()

import pandas as pd
import numpy as np

pd.set_option('display.max_columns', None)

df = pd.read_csv('data/MDA_04_dirty_data.csv')
print(df.head())
```

<Need to check>

- Station : ?
- SNWD : -inf
- TMAX 5505.0
- Some NaN s

```
 date                           station          PRCP   SNOW   SNWD    TMAX    TMIN   TOBS  WESF  inclement_weather
0  2018-01-01T00:00:00            ?               0.0    0.0   -inf   5505.0   -40.0   NaN   NaN        NaN
1  2018-01-01T00:00:00            ?               0.0    0.0   -inf   5505.0   -40.0   NaN   NaN        NaN
2  2018-01-01T00:00:00            ?               0.0    0.0   -inf   5505.0   -40.0   NaN   NaN        NaN
3  2018-01-02T00:00:00  GHCND:USC00280907         0.0    0.0   -inf   -8.3     -16.1   -12.2  NaN       False
4  2018-01-03T00:00:00  GHCND:USC00280907         0.0    0.0   -inf   -4.4     -13.9   -13.3  NaN       False
```

# Lab: Data Wrangling

```
# Understanding data - describe()


print(df.describe())
```

```
              PRCP        SNOW   SNWD         TMAX        TMIN        TOBS       WESF
count   765.000000  577.000000  577.0   765.000000  765.000000  398.000000  11.000000
mean      5.360392    4.202773    NaN  2649.175294  -15.914379    8.632161  16.290909
std      10.002138   25.086077    NaN  2744.156281   24.242849    9.815054   9.489832
min       0.000000    0.000000   -inf   -11.700000  -40.000000  -16.100000   1.800000
25%       0.000000    0.000000    NaN    13.300000  -40.000000    0.150000   8.600000
50%       0.000000    0.000000    NaN    32.800000  -11.100000    8.300000  19.300000
75%       5.800000    0.000000    NaN  5505.000000    6.700000   18.300000  24.900000
max      61.700000  229.000000    inf  5505.000000   23.900000   26.100000  28.700000
```

# Lab: Data Wrangling

```
# Understanding data - info()

print(df.info())
```

<Need to check>

- 765 entries (we have 1-year data)
- # of Non-null are different
- Inclement_weather is 'object', not 'boolean'

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 765 entries, 0 to 764
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   date               765 non-null    object
 1   station            765 non-null    object
 2   PRCP               765 non-null    float64
 3   SNOW               577 non-null    float64
 4   SNWD               577 non-null    float64
 5   TMAX               765 non-null    float64
 6   TMIN               765 non-null    float64
 7   TOBS               398 non-null    float64
 8   WESF               11 non-null     float64
 9   inclement_weather  408 non-null    object
dtypes: float64(7), object(3)
```

# Lab: Data Wrangling

```
# Understanding data - info()

print(df.info())
```

<Need to check>

- 765 entries (we have 1-year data)
- # of Non-null are different
- Inclement_weather is ‘object’, not ‘boolean’

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 765 entries, 0 to 764
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   date               765 non-null    object
 1   station            765 non-null    object
 2   PRCP               765 non-null    float64
 3   SNOW               577 non-null    float64
 4   SNWD               577 non-null    float64
 5   TMAX               765 non-null    float64
 6   TMIN               765 non-null    float64
 7   TOBS               398 non-null    float64
 8   WESF               11 non-null     float64
 9   inclement_weather  408 non-null    object
dtypes: float64(7), object(3)
```

# Lab: Data Wrangling

```
# Understanding Nulls : (1) WESF has only 11 non-null values: how do they look like?

print(df[df.WESF.isna()==False])
```

```
                    date  station  PRCP   SNOW  SNWD    TMAX  TMIN  TOBS  WESF  inclement_weather
7    2018-01-04T00:00:00        ?  20.6  229.0   inf  5505.0 -40.0   NaN  19.3               True
8    2018-01-04T00:00:00        ?  20.6  229.0   inf  5505.0 -40.0   NaN  19.3               True
58   2018-01-30T00:00:00        ?   1.5   13.0   inf  5505.0 -40.0   NaN   1.8               True
137  2018-03-08T00:00:00        ?  28.4    NaN   NaN  5505.0 -40.0   NaN  28.7                NaN
146  2018-03-13T00:00:00        ?   3.0   13.0   inf  5505.0 -40.0   NaN   3.0               True
159  2018-03-21T00:00:00        ?   6.6  114.0   inf  5505.0 -40.0   NaN   8.6               True
162  2018-03-21T00:00:00        ?   6.6  114.0   inf  5505.0 -40.0   NaN   8.6               True
186  2018-04-02T00:00:00        ?  14.0  152.0   inf  5505.0 -40.0   NaN  15.2               True
678  2018-11-16T00:00:00        ?  47.0  152.0   inf  5505.0 -40.0   NaN  24.9               True
679  2018-11-16T00:00:00        ?  47.0  152.0   inf  5505.0 -40.0   NaN  24.9               True
680  2018-11-16T00:00:00        ?  47.0  152.0   inf  5505.0 -40.0   NaN  24.9               True
```

# Lab: Data Wrangling

```
# Understanding Nulls : (2) How many entries have at least one null value?

contain_nulls = df[
   df.SNOW.isna() | df.SNWD.isna() | df.TOBS.isna()
   | df.WESF.isna() | df.inclement_weather.isna()
]
print(contain_nulls.shape[0])
print(contain_nulls.head(10))
```

```
765

date                         station          PRCP  SNOW  SNWD   TMAX     TMIN   TOBS   WESF  inclement_weather
0  2018-01-01T00:00:00             ?          0.0   0.0   -inf   5505.0   -40.0  NaN    NaN        NaN
1  2018-01-01T00:00:00             ?          0.0   0.0   -inf   5505.0   -40.0  NaN    NaN        NaN
2  2018-01-01T00:00:00             ?          0.0   0.0   -inf   5505.0   -40.0  NaN    NaN        NaN
3  2018-01-02T00:00:00  GHCND:USC00280907     0.0   0.0   -inf     -8.3   -16.1  -12.2  NaN        False
4  2018-01-03T00:00:00  GHCND:USC00280907     0.0   0.0   -inf     -4.4   -13.9  -13.3  NaN        False
5  2018-01-03T00:00:00  GHCND:USC00280907     0.0   0.0   -inf     -4.4   -13.9  -13.3  NaN        False
6  2018-01-03T00:00:00  GHCND:USC00280907     0.0   0.0   -inf     -4.4   -13.9  -13.3  NaN        False
7  2018-01-04T00:00:00             ?          20.6  229.0  inf   5505.0   -40.0  NaN    19.3       True
8  2018-01-04T00:00:00             ?          20.6  229.0  inf   5505.0   -40.0  NaN    19.3       True
9  2018-01-05T00:00:00             ?          0.3    NaN   NaN   5505.0   -40.0  NaN    NaN        NaN
```

# Lab: Data Wrangling

```
# Understanding inf/-inf: (1) Find the number of inf/-inf values per column in the df

def get_inf_count(df):
    return {
        col: df[df[col].isin([np.inf, -np.inf])].shape[0] for col in df.columns
    }

print(get_inf_count(df))
```

```
{'date': 0, 'station': 0, 'PRCP': 0, 'SNOW': 0, 'SNWD': 577, 'TMAX': 0, 'TMIN': 0, 'TOBS': 0, 'WESF': 0,
'inclement_weather': 0}
```

- (CF) # of null values per column?

➔ We already get information of Nulls from info()

➔ 765-(non-null)

# Lab: Data Wrangling

```
# Understanding inf/-inf: (2) Take a deep look at the 'SNWD(Snow depth)'

# Since snow depth (SNWD) and snowfall amount (SNOW) are closely related, let's interpret SNWD's inf/-inf
information through SNOW.

snow = pd.DataFrame({
    'np.inf Snow Depth': df[df.SNWD == np.inf].SNOW.describe(),
    '-np.inf Snow Depth': df[df.SNWD == -np.inf].SNOW.describe()
})

print(snow)
```

```
       np.inf Snow Depth   -np.inf Snow Depth
count          24.000000                553.0
mean          101.041667                  0.0
std            74.498018                  0.0
min            13.000000                  0.0
25%            25.000000                  0.0
50%           120.500000                  0.0
75%           152.000000                  0.0
max           229.000000                  0.0
```

# Lab: Data Wrangling

```
# Understanding inf/-inf: (2) Take a deep look at the ‘SNWD(Snow depth)’

# Since snow depth (SNWD) and snowfall amount (SNOW) are closely related, let's interpret SNWD's inf/-inf
information through SNOW.

snow = pd.DataFrame({
    'np.inf Snow Depth': df[df.SNWD == np.inf].SNOW.describe(),
    '-np.inf Snow Depth': df[df.SNWD == -np.inf].SNOW.describe()
})

print(snow)
```

```
         np.inf Snow Depth    -np.inf Snow Depth
count          24.000000                 553.0
mean          101.041667                   0.0
std            74.498018                   0.0
min            13.000000                   0.0
25%            25.000000                   0.0
50%           120.500000                   0.0
75%           152.000000                   0.0
max           229.000000                   0.0
```

-inf -> 0 -> NO SNOW

# Lab: Data Wrangling

```
# Understanding 'data' and 'station'

print(df.describe(include='object'))
```

```
                         date                station           inclement_weather
count                    765                 765                         408
unique                   324                   2                           2
top         2018-07-05T00:00:00    GHCND:USC00280907                  False
freq                       8                 398                         384
```

- ‘data’
  - We have 1-year data -> 765 rows : Need to check
  - ‘data’ has 324 unique rows (Some might be duplicated(765:365) and missing(365:324))
  - A specific date appears up to 8 times (2018-07-05T00:00:00 )

- ‘station’
  - We have 1 station, but # of unique data is 2 -> ‘GHCND:USC00280907’ and ‘?’
  - ‘GHCND:USC00280907’ appears 398 times -> ‘?’ appears (756-398) = 367 times

# Lab: Data Wrangling

```
# Understanding 'data' and 'station' : Duplication information
print(df[df.duplicated(keep=False)].head(10))
```

```
                    date              station  PRCP   SNOW  SNWD    TMAX   TMIN   TOBS  WESF  inclement_weather
0   2018-01-01T00:00:00                    ?    0.0    0.0  -inf  5505.0  -40.0    NaN   NaN               NaN
1   2018-01-01T00:00:00                    ?    0.0    0.0  -inf  5505.0  -40.0    NaN   NaN               NaN
2   2018-01-01T00:00:00                    ?    0.0    0.0  -inf  5505.0  -40.0    NaN   NaN               NaN
4   2018-01-03T00:00:00  GHCND:USC00280907    0.0    0.0  -inf    -4.4  -13.9  -13.3   NaN             False
5   2018-01-03T00:00:00  GHCND:USC00280907    0.0    0.0  -inf    -4.4  -13.9  -13.3   NaN             False
6   2018-01-03T00:00:00  GHCND:USC00280907    0.0    0.0  -inf    -4.4  -13.9  -13.3   NaN             False
7   2018-01-04T00:00:00                    ?   20.6  229.0   inf  5505.0  -40.0    NaN  19.3              True
8   2018-01-04T00:00:00                    ?   20.6  229.0   inf  5505.0  -40.0    NaN  19.3              True
9   2018-01-05T00:00:00                    ?    0.3    NaN   NaN  5505.0  -40.0    NaN   NaN               NaN
10  2018-01-05T00:00:00                    ?    0.3    NaN   NaN  5505.0  -40.0    NaN   NaN               NaN
20  2018-01-12T00:00:00  GHCND:USC00280907    1.3    0.0  -inf     9.4    0.6    7.8   NaN             False
21  2018-01-12T00:00:00                    ?    0.5    NaN   NaN  5505.0  -40.0    NaN   NaN               NaN
22  2018-01-12T00:00:00                    ?    0.5    NaN   NaN  5505.0  -40.0    NaN   NaN               NaN
23  2018-01-12T00:00:00  GHCND:USC00280907    1.3    0.0  -inf     9.4    0.6    7.8   NaN             False
24  2018-01-12T00:00:00  GHCND:USC00280907    1.3    0.0  -inf     9.4    0.6    7.8   NaN             False
26  2018-01-14T00:00:00  GHCND:USC00280907    0.0    0.0  -inf     2.2  -11.1  -11.1   NaN             False
27  2018-01-14T00:00:00                    ?    0.0    0.0  -inf  5505.0  -40.0    NaN   NaN               NaN
28  2018-01-14T00:00:00                    ?    0.0    0.0  -inf  5505.0  -40.0    NaN   NaN               NaN
29  2018-01-14T00:00:00  GHCND:USC00280907    0.0    0.0  -inf     2.2  -11.1  -11.1   NaN             False
31  2018-01-16T00:00:00  GHCND:USC00280907    0.0    0.0  -inf    -1.1   -9.4   -4.4   NaN             False
```

Same date
Diff. stations

'?' station
has lots NaN
values,
and 5505s

# Lab: Data Wrangling

- Handling the data!!
- Remove duplications : Combine the row having same date into one row.

BEFORE THAT, Let's consider the values that we should not erase carelessly!

- We will Keep the WESF values first

- QUIZ: Find unique ‘station’ values where the WSF value is not Null

```
date                   station               PRCP  SNOW  SNWD   TMAX      TMIN   TOBS   WESF  inclement_weather
0  2018-01-01T00:00:00         ?               0.0   0.0   -inf   5505.0    -40.0  NaN    NaN    NaN
1  2018-01-01T00:00:00         ?               0.0   0.0   -inf   5505.0    -40.0  NaN    NaN    NaN
2  2018-01-01T00:00:00         ?               0.0   0.0   -inf   5505.0    -40.0  NaN    NaN    NaN
3  2018-01-02T00:00:00  GHCND:USC00280907      0.0   0.0   -inf   -8.3      -16.1  -12.2  NaN    False
4  2018-01-03T00:00:00  GHCND:USC00280907      0.0   0.0   -inf   -4.4      -13.9  -13.3  NaN    False
X  2018-01-03T00:00:00  GHCND:USC00280907      0.0   0.0   -inf   -4.4      -13.9  -13.3  13     False
X  2018-01-03T00:00:00  GHCND:USC00280907      0.0   0.0   -inf   -4.4      -13.9  -13.3  13     False
X  2018-01-04T00:00:00         ?               20.6  229.0  inf   5505.0    -40.0  NaN    19.3   True
X  2018-01-04T00:00:00         ?               20.6  229.0  inf   5505.0    -40.0  NaN    19.3   True
9  2018-01-05T00:00:00         ?               0.3   NaN   NaN    5505.0    -40.0  NaN    NaN    NaN
```

For example:

GHCND:USC00280907, ?

# Lab: Data Wrangling

- QUIZ: Find unique ‘station’ values where the WSF value is not Null

```
# Understanding ‘data’ and ‘station’ : Unique value of the station whose WESF value is not null

print(df[df.WESF.notna()].station.unique())
```

```
['?']
```

# Lab: Data Wrangling

- QUIZ: Find unique ‘station’ values where the WSF value is not Null

```
# Understanding ‘data’ and ‘station’ : Unique value of the station whose WESF value is not null

print(df[df.WESF.notna()].station.unique())
```

```
['?']
```

- we only have values for the WESF column when the station is ‘?’

# Lab: Data Wrangling

- Handling duplications : Combine the row having same date into one row.
  - We will drop rows based on date.
  - Keep the WESF values first. (index = date)
  - If there are two types of station information for the same date, remove ‘?’
  - Delete the 'station’ column as it is no longer needed.
  - Combine the WESF values based on date.

```
# 1. make the date a datetime (We don't need time (hour/minutes…) information.)

# 2. save the WEST information

# 3. sort ‘?’ to the bottom

# 4. drop duplicates based on the date column keeping the first occurrence

# 5. remove the station column because we are done with it

# 6. Insert the saved WESF value.
```

# Lab: Data Wrangling

```
# 1. make the date a datetime (We don't need time (hour/minutes…) information.)
df.date = pd.to_datetime(df.date)

# 2. save the WEST information
station_qm_wesf = df[df.station == '?'].drop_duplicates('date').set_index('date').WESF
```

```
  date station  PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF  …
0  2018-01-01      ?   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN
1  2018-01-01      ?   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN
2  2018-01-01      ?   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN
7  2018-01-04      ?  20.6  229.0   inf  5505.0 -40.0   NaN  19.3
8  2018-01-04      ?  20.6  229.0   inf  5505.0 -40.0   NaN  19.3
9  2018-01-05      ?   0.3    NaN   NaN  5505.0 -40.0   NaN   NaN
10 2018-01-05      ?   0.3    NaN   NaN  5505.0 -40.0   NaN   NaN
21 2018-01-12      ?   0.5    NaN   NaN  5505.0 -40.0   NaN   NaN
22 2018-01-12      ?   0.5    NaN   NaN  5505.0 -40.0   NaN   NaN
27 2018-01-14      ?   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN
28 2018-01-14      ?   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN
```

df[df.station == '?']

```
  date station  PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF  …
0  2018-01-01      ?   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN
7  2018-01-04      ?  20.6  229.0   inf  5505.0 -40.0   NaN  19.3
9  2018-01-05      ?   0.3    NaN   NaN  5505.0 -40.0   NaN   NaN
21 2018-01-12      ?   0.5    NaN   NaN  5505.0 -40.0   NaN   NaN
27 2018-01-14      ?   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN
```

drop_duplicates('date')

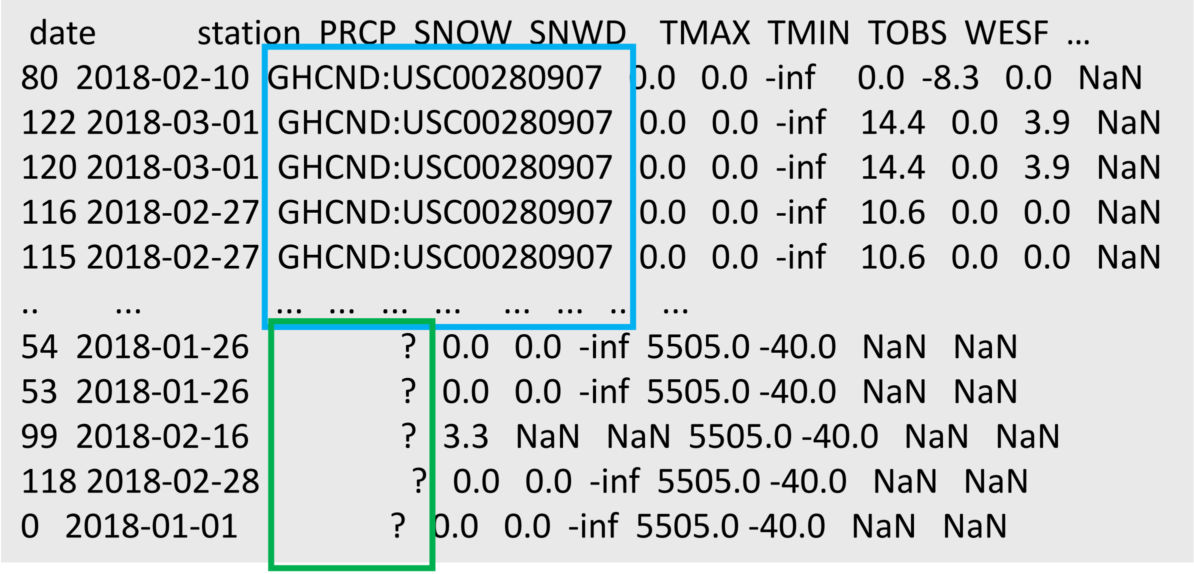
```
date
2018-01-01     NaN
2018-01-04    19.3
2018-01-05     NaN
2018-01-12     NaN
2018-01-14     NaN
Name: WESF, dtype: float64
```

set_index(‘data’).WESF

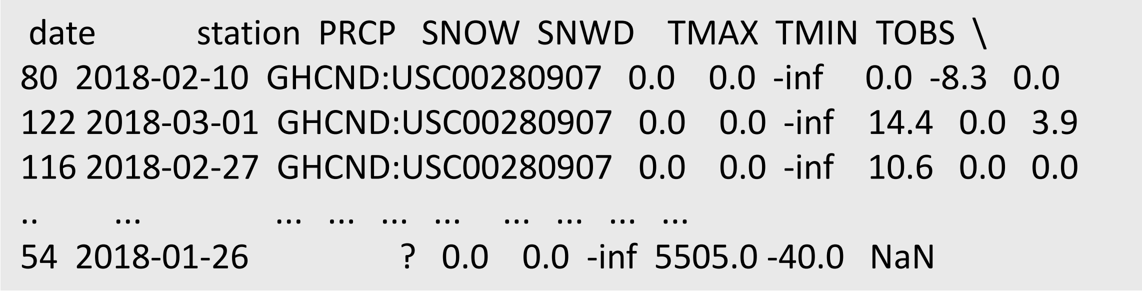
# Lab: Data Wrangling

```
# 3. sort '?' to the bottom
df.sort_values('station', ascending=False, inplace=True)

# 4. drop duplicates based on the date column keeping the first occurrence
# which will be the valid station if it has data
df_deduped = df.drop_duplicates('date')
```



```
 date          station  PRCP  SNOW  SNWD   TMAX  TMIN  TOBS  WESF  ...
80  2018-02-10 GHCND:USC00280907  0.0   0.0  -inf    0.0  -8.3   0.0   NaN
122 2018-03-01 GHCND:USC00280907  0.0   0.0  -inf   14.4   0.0   3.9   NaN
120 2018-03-01 GHCND:USC00280907  0.0   0.0  -inf   14.4   0.0   3.9   NaN
116 2018-02-27 GHCND:USC00280907  0.0   0.0  -inf   10.6   0.0   0.0   NaN
115 2018-02-27 GHCND:USC00280907  0.0   0.0  -inf   10.6   0.0   0.0   NaN
..         ...               ...  ...   ...   ...    ...   ...   ..   ...
54  2018-01-26                 ?  0.0   0.0  -inf  5505.0 -40.0  NaN   NaN
53  2018-01-26                 ?  0.0   0.0  -inf  5505.0 -40.0  NaN   NaN
99  2018-02-16                 ?  3.3   NaN   NaN  5505.0 -40.0  NaN   NaN
118 2018-02-28                 ?  0.0   0.0  -inf  5505.0 -40.0  NaN   NaN
0   2018-01-01                 ?  0.0   0.0  -inf  5505.0 -40.0  NaN   NaN
```



```
 date          station  PRCP  SNOW  SNWD   TMAX  TMIN  TOBS  \
80  2018-02-10 GHCND:USC00280907  0.0   0.0  -inf    0.0  -8.3   0.0
122 2018-03-01 GHCND:USC00280907  0.0   0.0  -inf   14.4   0.0   3.9
116 2018-02-27 GHCND:USC00280907  0.0   0.0  -inf   10.6   0.0   0.0
..         ...               ...  ...   ...   ...    ...   ...   ...
54  2018-01-26                 ?  0.0   0.0  -inf  5505.0 -40.0  NaN
```

# Lab: Data Wrangling

```
# 5. remove the station column because we are done with it
df_deduped = df_deduped.drop(columns='station').set_index('date').sort_index()

# 6. Insert the saved WESF value.
# combine_first(): Combine two DataFrame objects by filling null values in one DataFrame with non-null values

df_deduped = df_deduped.assign(
   WESF=lambda x: x.WESF.combine_first(station_qm_wesf)
)

print(df_deduped.shape)
```

```
            PRCP   SNOW  SNWD    TMAX  TMIN  TOBS  WESF inclement_weather
date
…
2018-03-19   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN               NaN
2018-03-21   0.0    0.0  -inf     2.8  -2.8   0.6   NaN             False
2018-03-22  17.3  178.0   inf     1.7  -1.7   0.0   NaN              True
2018-03-23   0.0    0.0  -inf     8.9   0.0   1.1   NaN             False
2018-03-25   0.0    0.0  -inf     8.3  -1.1  -0.6   NaN             False
…
```

```
(324, 8)
```

```
            PRCP   SNOW  SNWD    TMAX  TMIN  TOBS  WESF inclement_weather
date
…
2018-03-19   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN               NaN
2018-03-21   0.0    0.0  -inf     2.8  -2.8   0.6   8.6             False
2018-03-22  17.3  178.0   inf     1.7  -1.7   0.0   NaN              True
2018-03-23   0.0    0.0  -inf     8.9   0.0   1.1   NaN             False
2018-03-25   0.0    0.0  -inf     8.3  -1.1  -0.6   NaN             False
…
```

# Lab: Data Wrangling

## ◆Handling nulls - Remove

① Remove if at least one null exist

```
print(df_deduped.dropna().shape)
Print(df_deduped)
```

Do not apply

```
(4, 8)
            PRCP   SNOW  SNWD  TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-30   0.0    0.0  -inf   6.7  -1.7  -0.6   1.8             False
2018-03-13   4.1   51.0   inf   5.6  -3.9   0.0   3.0              True
2018-03-21   0.0    0.0  -inf   2.8  -2.8   0.6   8.6             False
2018-04-02   9.1  127.0   inf  12.8  -1.1  -1.1  15.2              True
```

# Lab: Data Wrangling

## ◆Handling nulls - Remove

② Remove if there are nulls in certain columns (all)

Do not apply

```
df_deduped =df_deduped.dropna(
   how='all', subset=['inclement_weather', 'SNOW', 'SNWD’]
)
print(df_deduped)
```

```
 PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN              NaN
2018-01-02   0.0    0.0  -inf    -8.3 -16.1 -12.2   NaN            False
2018-01-03   0.0    0.0  -inf    -4.4 -13.9 -13.3   NaN            False
2018-01-04  20.6  229.0   inf  5505.0 -40.0   NaN  19.3             True
2018-01-05  14.2  127.0   inf    -4.4 -13.9 -13.9   NaN             True
2018-01-06   0.0    0.0  -inf   -10.0 -15.6 -15.0   NaN            False
2018-01-07   0.0    0.0  -inf   -11.7 -17.2 -16.1   NaN            False
2018-01-08   0.0    0.0  -inf    -7.8 -16.7  -8.3   NaN            False
2018-01-10   0.0    0.0  -inf     5.0  -7.8  -7.8   NaN            False
2018-01-11   0.0    0.0  -inf     4.4  -7.8   1.1   NaN            False
2018-01-12   1.3    0.0  -inf     9.4   0.6   7.8   NaN            False
2018-01-13  17.5    NaN   NaN  5505.0 -40.0   NaN   NaN              NaN
2018-01-14   0.0    0.0  -inf     2.2 -11.1 -11.1   NaN            False
2018-01-15   0.0    0.0  -inf    -5.0 -11.7  -8.9   NaN            False
2018-01-16   0.0    0.0  -inf    -1.1  -9.4  -4.4   NaN            False
```

```
 PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0  -inf  5505.0 -40.0   NaN   NaN              NaN
2018-01-02   0.0    0.0  -inf    -8.3 -16.1 -12.2   NaN            False
2018-01-03   0.0    0.0  -inf    -4.4 -13.9 -13.3   NaN            False
2018-01-04  20.6  229.0   inf  5505.0 -40.0   NaN  19.3             True
2018-01-05  14.2  127.0   inf    -4.4 -13.9 -13.9   NaN             True
2018-01-06   0.0    0.0  -inf   -10.0 -15.6 -15.0   NaN            False
2018-01-07   0.0    0.0  -inf   -11.7 -17.2 -16.1   NaN            False
2018-01-08   0.0    0.0  -inf    -7.8 -16.7  -8.3   NaN            False
2018-01-10   0.0    0.0  -inf     5.0  -7.8  -7.8   NaN            False
2018-01-11   0.0    0.0  -inf     4.4  -7.8   1.1   NaN            False
2018-01-12   1.3    0.0  -inf     9.4   0.6   7.8   NaN            False
2018-01-14   0.0    0.0  -inf     2.2 -11.1 -11.1   NaN            False
2018-01-15   0.0    0.0  -inf    -5.0 -11.7  -8.9   NaN            False
2018-01-16   0.0    0.0  -inf    -1.1  -9.4  -4.4   NaN            False
```

# Lab: Data Wrangling

## ◆Handling nulls - Remove

③ Remove only if there are nulls in all columns

```
df_deduped =df_deduped.dropna(how=‘all’)
print(df_deduped.shape)
```

```
(324, 8)
```

# Lab: Data Wrangling

## ◆Handling nulls - Fill

① Filling nulls with a constant value

```
print(df_deduped['WESF'].fillna(0, inplace=True))
print(df_deduped)
```

```
            PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0   -inf 5505.0 -40.0  NaN   0.0         NaN
2018-01-02   0.0    0.0   -inf   -8.3 -16.1 -12.2  0.0         False
2018-01-03   0.0    0.0   -inf   -4.4 -13.9 -13.3  0.0         False
2018-01-04  20.6  229.0    inf 5505.0 -40.0  NaN  19.3         True
2018-01-05  14.2  127.0    inf   -4.4 -13.9 -13.9  0.0         True
```

```
df_deduped = df_deduped.assign(
   TMAX=lambda x: x.TMAX.replace(5505, np.nan),
   TMIN=lambda x: x.TMIN.replace(-40, np.nan),
)
print(df_deduped)
```

```
            PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0   -inf   NaN   NaN   NaN   0.0         NaN
2018-01-02   0.0    0.0   -inf  -8.3 -16.1 -12.2   0.0         False
2018-01-03   0.0    0.0   -inf  -4.4 -13.9 -13.3   0.0         False
2018-01-04  20.6  229.0    inf   NaN   NaN   NaN  19.3         True
2018-01-05  14.2  127.0    inf  -4.4 -13.9 -13.9   0.0         True
```

# Lab: Data Wrangling

## ◆Handling nulls - Fill

② Filling nulls with the preceding (forward fill) or succeeding (backward fill) values.

```
df_deduped=df_deduped.assign(
   TMAX=lambda x: x.TMAX.fillna(method='ffill’),                    #backward fill : ‘bfill’
   TMIN=lambda x: x.TMIN.fillna(method='ffill')
)

print(df_deduped)
```

```
 PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0    -inf   NaN     NaN   NaN     0.0           NaN
2018-01-02   0.0    0.0    -inf   -8.3   -16.1  -12.2    0.0          False
2018-01-03   0.0    0.0    -inf   -4.4   -13.9  -13.3    0.0          False
2018-01-04  20.6  229.0   inf    NaN     NaN   NaN   19.3          True
2018-01-05  14.2  127.0   inf    -4.4    -13.9  -13.9   0.0          True
```

```
 PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0    -inf   NaN   NaN     NaN     0.0           NaN
2018-01-02   0.0    0.0    -inf  -8.3   -16.1   -12.2    0.0           False
2018-01-03   0.0    0.0    -inf  -4.4 -  13.9   -13.3    0.0           False
2018-01-04  20.6  229.0   inf  -4.4    -13.9    NaN    19.3           True
2018-01-05  14.2  127.0   inf  -4.4    -13.9   -13.9    0.0           True
```

# Lab: Data Wrangling

## ◆Handling nulls - Fill

③ Filling nulls with the clip() method to cap values at a specific minimum and/or maximum threshold.

```
df_deduped=df_deduped.assign(
   SNWD=lambda x: x.SNWD.clip(0, x.SNOW)
)

print(df_deduped)
```

```
 PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0   -inf  NaN    NaN   NaN    0.0          NaN
2018-01-02   0.0    0.0   -inf  -8.3   -16.1 -12.2   0.0        False
2018-01-03   0.0    0.0   -inf  -4.4   -13.9 -13.3   0.0        False
2018-01-04  20.6  229.0   inf   NaN    NaN   NaN   19.3         True
2018-01-05  14.2  127.0   inf   -4.4   -13.9 -13.9   0.0         True
```

```
 PRCP   SNOW  SNWD   TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0   0.0    0.0    NaN   NaN   NaN    0.0           NaN
2018-01-02   0.0   0.0    0.0    -8.3  -16.1  -12.2  0.0           False
2018-01-03   0.0   0.0    0.0    -4.4  -13.9  -13.3  0.0           False
2018-01-04  20.6  229.0  229.0  -4.4  -13.9   NaN   19.3          True
2018-01-05  14.2  127.0  127.0  -4.4  -13.9  -13.9   0.0          True
```

# Lab: Data Wrangling

## ◆Handling nulls - Impute

① Use fillna() with other types of calculations.

➢ We replace missing values of TMAX with the median of all TMAX values, TMIN with the median of all TMIN values, and TOBS to the average of the TMAX and TMIN values. Since we place TOBS last, we have access to the imputed values for TMIN and TMAX in the calculation

```
df_deduped=df_deduped.assign(
    TMAX=lambda x: x.TMAX.fillna(x.TMAX.median()),
    TMIN=lambda x: x.TMIN.fillna(x.TMIN.median()),
    # average of TMAX and TMIN
    TOBS=lambda x: x.TOBS.fillna((x.TMAX + x.TMIN) / 2)
)
print(df_deduped)
```

```
            PRCP   SNOW   SNWD  TMAX  TMIN  TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0    0.0   NaN   NaN   NaN   0.0               NaN
2018-01-02   0.0    0.0    0.0  -8.3 -16.1 -12.2   0.0             False
2018-01-03   0.0    0.0    0.0  -4.4 -13.9 -13.3   0.0             False
2018-01-04  20.6  229.0  229.0  -4.4 -13.9   NaN  19.3              True
2018-01-05  14.2  127.0  127.0  -4.4 -13.9 -13.9   0.0              True
```

```
            PRCP   SNOW   SNWD  TMAX  TMIN   TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0    0.0  16.1   6.7  11.40   0.0               NaN
2018-01-02   0.0    0.0    0.0  -8.3 -16.1 -12.20   0.0             False
2018-01-03   0.0    0.0    0.0  -4.4 -13.9 -13.30   0.0             False
2018-01-04  20.6  229.0  229.0  -4.4 -13.9  -9.15  19.3              True
2018-01-05  14.2  127.0  127.0  -4.4 -13.9 -13.90   0.0              True
```

# Lab: Data Wrangling

## ◆Handling nulls - Impute

### ② Interpolate with the interpolate() method

➢ We specify the method parameter with the interpolation strategy to use. There are many options, but we will stick with the default of 'linear', which will treat values as evenly spaced and place missing values in the middle of existing ones. We have some missing data, so we will reindex first. Look at January 9th, which we didn't have before—the values for TMAX, TMIN, and TOBS are the average of values the day prior (January 8th) and the day after (January 10th):

```
print(df_deduped.head(10))
df_deduped=df_deduped.reindex(
   pd.date_range('2018-01-01', '2018-12-31', freq='D')).apply(lambda x: x.interpolate())
print(df_deduped.head(10))
```

```
PRCP   SNOW   SNWD  TMAX  TMIN   TOBS  WESF inclement_weather
date
2018-01-01   0.0    0.0    0.0  16.1   6.7  11.40   0.0            NaN
2018-01-02   0.0    0.0    0.0  -8.3 -16.1 -12.20   0.0          False
2018-01-03   0.0    0.0    0.0  -4.4 -13.9 -13.30   0.0          False
2018-01-04  20.6  229.0  229.0  -4.4 -13.9  -9.15  19.3           True
2018-01-05  14.2  127.0  127.0  -4.4 -13.9 -13.90   0.0           True
2018-01-06   0.0    0.0    0.0 -10.0 -15.6 -15.00   0.0          False
2018-01-07   0.0    0.0    0.0 -11.7 -17.2 -16.10   0.0          False
2018-01-08   0.0    0.0    0.0  -7.8 -16.7  -8.30   0.0          False
2018-01-10   0.0    0.0    0.0   5.0  -7.8  -7.80   0.0          False
2018-01-11   0.0    0.0    0.0   4.4  -7.8   1.10   0.0          False
```

```
PRCP   SNOW   SNWD  TMAX   TMIN   TOBS  WESF inclement_weather
2018-01-01   0.0    0.0    0.0  16.1   6.70  11.40   0.0            NaN
2018-01-02   0.0    0.0    0.0  -8.3 -16.10 -12.20   0.0          False
2018-01-03   0.0    0.0    0.0  -4.4 -13.90 -13.30   0.0          False
2018-01-04  20.6  229.0  229.0  -4.4 -13.90  -9.15  19.3           True
2018-01-05  14.2  127.0  127.0  -4.4 -13.90 -13.90   0.0           True
2018-01-06   0.0    0.0    0.0 -10.0 -15.60 -15.00   0.0          False
2018-01-07   0.0    0.0    0.0 -11.7 -17.20 -16.10   0.0          False
2018-01-08   0.0    0.0    0.0  -7.8 -16.70  -8.30   0.0          False
2018-01-09   0.0    0.0    0.0  -1.4 -12.25  -8.05   0.0            NaN
2018-01-10   0.0    0.0    0.0   5.0  -7.80  -7.80   0.0          False
```

*Default: interpolate missing values linearly

Bye Bye