```
# Import libraries
import pandas as pd
import numpy as np
```

```
# Load files into Pandas dataframes:
weather_data_01 = pd.read_fwf("/content/drive/MyDrive/Colab Notebooks/weather_data/weather01.txt", colspecs='infer', widths=None,
weather_data_02 = pd.read_fwf("/content/drive/MyDrive/Colab Notebooks/weather_data/weather02.txt", colspecs='infer', widths=None,
weather_data_03 = pd.read_fwf("/content/drive/MyDrive/Colab Notebooks/weather_data/weather03.txt", colspecs='infer', widths=None,
weather_data_04 = pd.read_fwf("/content/drive/MyDrive/Colab Notebooks/weather_data/weather04.txt", colspecs='infer', widths=None,
weather_data_05 = pd.read_fwf("/content/drive/MyDrive/Colab Notebooks/weather_data/weather05.txt", colspecs='infer', widths=None,
weather_data_06 = pd.read_fwf("/content/drive/MyDrive/Colab Notebooks/weather_data/weather06.txt", colspecs='infer', widths=None,

# Join Dataframes: weather_data_merged
weather_data_merged = pd.concat([weather_data_01.iloc[:286,2:],weather_data_02.iloc[:286,2:],weather_data_03.iloc[:286,2:],weather
weather_data_merged
```

	year	month	measure	X1	X2	X4	X5	X6	X7	X9	...	X20
0	2014	12	Max.TemperatureF	64	42	43	42	45	38	49	...	36
1	2014	12	Mean.TemperatureF	52	38	37	34	42	30	39	...	32
2	2014	12	Min.TemperatureF	39	33	30	26	38	21	29	...	27
3	2014	12	Max.Dew.PointF	46	40	24	37	45	36	49	...	30
4	2014	12	MeanDew.PointF	40	27	21	25	40	20	41	...	24
...	...	...	...	...	...	...	...	...	...	...	...	...
281	2015	12	Max.Gust.SpeedMPH	17	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
282	2015	12	PrecipitationIn	0.14	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
283	2015	12	CloudCover	7	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
284	2015	12	Events	Rain	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
285	2015	12	WindDirDegrees	109	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN

286 rows × 29 columns

```
# Convert days to rows:
data_transform = pd.melt(weather_data_merged, id_vars=['year','month','measure'],var_name='day',value_name='value')
data_transform
```

	year	month	measure	day	value
0	2014	12	Max.TemperatureF	X1	64
1	2014	12	Mean.TemperatureF	X1	52
2	2014	12	Min.TemperatureF	X1	39
3	2014	12	Max.Dew.PointF	X1	46
4	2014	12	MeanDew.PointF	X1	40
...	...	...	...	...	...
7431	2015	12	Max.Gust.SpeedMPH	X31	NaN
7432	2015	12	PrecipitationIn	X31	NaN
7433	2015	12	CloudCover	X31	NaN
7434	2015	12	Events	X31	NaN
7435	2015	12	WindDirDegrees	X31	NaN

7436 rows × 5 columns

```
# Replace 'X' with an empty space ''
data_transform.day = data_transform.day.replace('X','',regex=True)
data_transform
```

	year	month	measure	day	value
0	2014	12	Max.TemperatureF	1	64
1	2014	12	Mean.TemperatureF	1	52
2	2014	12	Min.TemperatureF	1	39
3	2014	12	Max.Dew.PointF	1	46
4	2014	12	MeanDew.PointF	1	40
...	...	...	...	...	...
7431	2015	12	Max.Gust.SpeedMPH	31	NaN
7432	2015	12	PrecipitationIn	31	NaN
7433	2015	12	CloudCover	31	NaN

```
# Creating column date using year, month and day.
data_transform['date'] = pd.to_datetime(data_transform[['year','month','day']],errors='coerce',format='%Y-%m-%d')
print(data_transform.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7436 entries, 0 to 7435
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   year     7436 non-null   int64
 1   month    7436 non-null   int64
 2   measure  7436 non-null   object
 3   day      7436 non-null   object
 4   value    6553 non-null   object
 5   date     7282 non-null   datetime64[ns]
dtypes: datetime64[ns](1), int64(2), object(3)
memory usage: 348.7+ KB
None
```

```
# Delete colums year, month and day
data_transform.drop(['year','month','day'], axis=1,inplace=True)
data_transform
```

	measure	value	date
0	Max.TemperatureF	64	2014-12-01
1	Mean.TemperatureF	52	2014-12-01
2	Min.TemperatureF	39	2014-12-01
3	Max.Dew.PointF	46	2014-12-01
4	MeanDew.PointF	40	2014-12-01
...	...	...	...
7431	Max.Gust.SpeedMPH	NaN	2015-12-31
7432	PrecipitationIn	NaN	2015-12-31
7433	CloudCover	NaN	2015-12-31
7434	Events	NaN	2015-12-31
7435	WindDirDegrees	NaN	2015-12-31

```
# Pivot measures and reset index:
data_master = data_transform.pivot_table(index='date', columns='measure',values='value', aggfunc='first').reset_index()
# Remove NaN values:
data_master.dropna(inplace=True)
data_master
```

measure	date	CloudCover	Events	Max.Dew.PointF	Max.Gust.SpeedMPH
0	2014-12-01	6	Rain	46	29
1	2014-12-02	7	Rain-Snow	40	29
3	2014-12-05	5	Rain	37	26
4	2014-12-06	8	Rain	45	25
5	2014-12-07	6	Rain	36	32
...	...	...	...	...	...
295	2015-11-19	8	Rain	50	20
296	2015-11-20	6	Rain	58	29
297	2015-11-22	8	Rain	44	25
298	2015-11-23	3	Rain	36	31

```
# Convert -Rain to Rain
data_master['Events'] = data_master['Events'].replace('-Rain','Rain')
```

```
129 rows x 23 columns
```

```
# Count Events
print(data_master['Events'].value_counts())
```

```
Rain             77
Snow             24
Rain-Snow         6
Fog-Rain          6
Fog-Snow          5
Fog               5
Fog-Rain-Snow     2
derstorm          2
storm             2
Name: Events, dtype: int64
```

```
data_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129 entries, 0 to 305
Data columns (total 23 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   date                       129 non-null    datetime64[ns]
 1   CloudCover                 129 non-null    object
 2   Events                     129 non-null    object
 3   Max.Dew.PointF             129 non-null    object
 4   Max.Gust.SpeedMPH          129 non-null    object
 5   Max.Humidity               129 non-null    object
 6   Max.Sea.Level.PressureIn   129 non-null    object
 7   Max.TemperatureF           129 non-null    object
 8   Max.VisibilityMiles        129 non-null    object
 9   Max.Wind.SpeedMPH          129 non-null    object
 10  Mean.Humidity              129 non-null    object
 11  Mean.Sea.Level.PressureIn  129 non-null    object
 12  Mean.TemperatureF          129 non-null    object
 13  Mean.VisibilityMiles       129 non-null    object
 14  Mean.Wind.SpeedMPH         129 non-null    object
 15  MeanDew.PointF             129 non-null    object
 16  Min.DewpointF              129 non-null    object
 17  Min.Humidity               129 non-null    object
 18  Min.Sea.Level.PressureIn   129 non-null    object
 19  Min.TemperatureF           129 non-null    object
 20  Min.VisibilityMiles        129 non-null    object
 21  PrecipitationIn            129 non-null    object
 22  WindDirDegrees             129 non-null    object
dtypes: datetime64[ns](1), object(22)
memory usage: 24.2+ KB
```

```
# Let's convert datatypes from Objects to Numeric except for Events and Date, so we can do some math fuctions.
num_data_master = list(data_master.loc[:, ~data_master.columns.isin(['date', 'Events'])])
data_master[num_data_master] = data_master[num_data_master].apply(pd.to_numeric, errors='coerce').fillna(0)
data_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129 entries, 0 to 305
Data columns (total 23 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   date                       129 non-null    datetime64[ns]
 1   CloudCover                 129 non-null    int64
 2   Events                     129 non-null    object
 3   Max.Dew.PointF             129 non-null    int64
 4   Max.Gust.SpeedMPH          129 non-null    int64
 5   Max.Humidity               129 non-null    int64
 6   Max.Sea.Level.PressureIn   129 non-null    float64
 7   Max.TemperatureF           129 non-null    int64
 8   Max.VisibilityMiles        129 non-null    int64
 9   Max.Wind.SpeedMPH          129 non-null    int64
 10  Mean.Humidity              129 non-null    int64
 11  Mean.Sea.Level.PressureIn  129 non-null    float64
 12  Mean.TemperatureF          129 non-null    int64
 13  Mean.VisibilityMiles       129 non-null    int64
 14  Mean.Wind.SpeedMPH         129 non-null    int64
 15  MeanDew.PointF             129 non-null    int64
 16  Min.DewpointF              129 non-null    int64
 17  Min.Humidity               129 non-null    int64
 18  Min.Sea.Level.PressureIn   129 non-null    float64
 19  Min.TemperatureF           129 non-null    int64
 20  Min.VisibilityMiles        129 non-null    int64
 21  PrecipitationIn            129 non-null    float64
 22  WindDirDegrees             129 non-null    int64
dtypes: datetime64[ns](1), float64(4), int64(17), object(1)
memory usage: 24.2+ KB
```

```
data_master['Events'].describe()
```

```
count      129
unique       9
top       Rain
freq        77
Name: Events, dtype: object
```

```
# Display rows where Events is equal to 'Snow'
(data_master[data_master['Events'] == 'Snow']).head()
```

x.TemperatureF	Max.VisibilityMiles	Max.Wind.SpeedMPH	...	Mean.Visi
39	10	16	...	
36	10	21	...	
18	10	22	...	
26	10	31	...	
30	10	25	...	

```
# Display the amount of events per month:
data_master_month = pd.DatetimeIndex(data_master['date']).month
data_master.groupby(data_master_month)['Events'].count()
```

```
date
1     13
2     16
3     13
4     10
5      7
6      9
```

```
7     11
8      8
9      7
10     8
11     9
12    18
Name: Events, dtype: int64
```

```
# Calculate the mean value for Max.TemperatureF and print the amunt of
max_temp_f = data_master['Max.TemperatureF']
max_temp_f_greater_mean = max_temp_f[max_temp_f > max_temp_f.mean()]
print('Greater than the mean value: ', + max_temp_f_greater_mean.count(),'\n', max_temp_f_greater_mean)
```

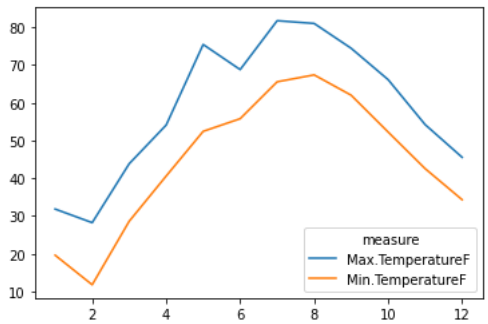
```
Greater than the mean value:  61
 0      64
20     59
83     57
96     56
108    55
       ..
280    62
289    57
290    59
295    55
296    61
Name: Max.TemperatureF, Length: 61, dtype: int64
```

```
# For each Event and month, what was the avarege Max.TemperatureF and Min.TemperatureF?
data_master.groupby([data_master.date.dt.month, 'Events'])['Max.TemperatureF','Min.TemperatureF'].mean().round(2).reset_index()
```

0	1	Fog-Rain	52.00	35.00
1	1	Fog-Rain-Snow	34.00	31.00
2	1	Fog-Snow	24.00	12.50
3	1	Rain	44.50	26.50
4	1	Snow	27.29	15.86
5	2	Fog-Rain-Snow	36.00	7.00
6	2	Fog-Snow	28.67	13.33
7	2	Rain-Snow	39.00	28.00
8	2	Snow	26.45	10.36

```
# What's the average for Max.TemperatureF and Min.TemperatureF by month?
import matplotlib.pyplot as plt
mean_temp_by_month = data_master.groupby(data_master.date.dt.month)['Max.TemperatureF','Min.TemperatureF'].mean().round(2)
mean_temp_by_month.plot()
plt.show()
```

```
<ipython-input-137-01b459b86b4a>:3: FutureWarning: Indexing with mul
  mean_temp_by_month = data_master.groupby(data_master.date.dt.month
```



daideas para graficas:

- build a chart by day of the week to look for a corralation
- build a chart to display the average by month para mostrar los max y min
- make a prediccion for the next month
- cuales son los eventos mas comunes por mes.
- que es el valor percentil? vale la pena incluirlo.
- Usar lanfuncion describe() para mostrar una descripcion general
- usar algo como esto para mostrar valores arriba de la media: print(ages[ages > ages.mean()])
- Antes de los calculos creo que debo convertir mis datos a numeros. No he podido hacerlo ya que al parecer hay texto en algunos valores.. :/

✓ 0s completed at 12:35 PM