



Daily Weather

Madrid



Team 3

Rachata Thananchai 650510638

Mesani Laihueang 650510676

Warisa Narata 650510683





01

Data Preparation



Tidy data

Before

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6812 entries, 0 to 6811
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CET              6812 non-null    object  
 1   Max TemperatureC 6810 non-null    float64
 2   Mean TemperatureC 6809 non-null    float64
 3   Min TemperatureC  6810 non-null    float64
 4   Dew PointC       6810 non-null    float64
 5   MeanDew PointC   6810 non-null    float64
 6   Min DewPointC   6810 non-null    float64
 7   Max Humidity     6810 non-null    float64
 8   Mean Humidity    6810 non-null    float64
 9   Min Humidity     6810 non-null    float64
 10  Max Sea Level PressurehPa 6812 non-null int64  
 11  Mean Sea Level PressurehPa 6812 non-null int64  
 12  Min Sea Level PressurehPa 6812 non-null int64  
 13  Max VisibilityKm  5872 non-null    float64
 14  Mean VisibilityKm 5872 non-null    float64
 15  Min VisibilityKM 5872 non-null    float64
 16  Max Wind SpeedKm/h 6812 non-null    int64  
 17  Mean Wind SpeedKm/h 6812 non-null    int64  
 18  Max Gust SpeedKm/h 3506 non-null    float64
 19  Precipitationmm   6812 non-null    float64
 20  CloudCover        5440 non-null    float64
 21  Events            1798 non-null    object  
 22  WindDirDegrees   6812 non-null    int64  
dtypes: float64(15), int64(6), object(2)
memory usage: 1.2+ MB
```

After

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6812 entries, 0 to 6811
Data columns (total 23 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CET              6812 non-null    object  
 1   Max TemperatureC 6810 non-null    float64
 2   Mean TemperatureC 6809 non-null    float64
 3   Min TemperatureC  6810 non-null    float64
 4   Max Dew PointC   6810 non-null    float64
 5   Mean Dew PointC  6810 non-null    float64
 6   Min Dew PointC   6810 non-null    float64
 7   Max Humidity     6810 non-null    float64
 8   Mean Humidity    6810 non-null    float64
 9   Min Humidity     6810 non-null    float64
 10  Max Sea Level PressurehPa 6812 non-null int64  
 11  Mean Sea Level PressurehPa 6812 non-null int64  
 12  Min Sea Level PressurehPa 6812 non-null int64  
 13  Max VisibilityKm  5872 non-null    float64
 14  Mean VisibilityKm 5872 non-null    float64
 15  Min VisibilityKM 5872 non-null    float64
 16  Max Wind SpeedKm/h 6812 non-null    int64  
 17  Mean Wind SpeedKm/h 6812 non-null    int64  
 18  Max Gust SpeedKm/h 3506 non-null    float64
 19  Precipitationmm   6812 non-null    float64
 20  CloudCover        5440 non-null    float64
 21  Events            1798 non-null    object  
 22  WindDirDegrees   6812 non-null    int64  
dtypes: float64(15), int64(6), object(2)
memory usage: 1.2+ MB
```

Remove duplicates

No duplicate data

```
[30]: data[data['CET'].duplicated()]
```

CET	Max TemperatureC	Mean TemperatureC	Min TemperatureC	Max Dew PointC	Mean Dew PointC	Min Dew PointC	Max Humidity	Mean Humidity	Min Humidity	...	Max VisibilityKm	Mean VisibilityKm	Min VisibilitykM	Max Wind SpeedKm/h	M

0 rows × 23 columns



Data type

Convert the data type of the 'CET' column from Object to Datetime.

```
# convert object type to datetimes type  
data['CET'] = pd.to_datetime(data['CET'])
```

Before

```
data.dtypes
```

CET	object
-----	--------

After

```
data.dtypes
```

CET	datetime64[ns]
-----	----------------

Missing value

Populate the computable data and remove a significant amount of missing data.

Before

```
# fill missing value
data.isna().sum()

CET          0
Max TemperatureC    2
Mean TemperatureC   3
Min TemperatureC    2
Max Dew PointC     2
Mean Dew PointC    2
Min Dew PointC     2
Max Humidity        2
Mean Humidity       2
Min Humidity        2
Max Sea Level PressurehPa 0
Mean Sea Level PressurehPa 0
Min Sea Level PressurehPa 0
Max VisibilityKm    940
Mean VisibilityKm   940
Min VisibilitykM    940
Max Wind SpeedKm/h  0
Mean Wind SpeedKm/h 0
Max Gust SpeedKm/h  3306
Precipitationmm     0
CloudCover         1372
Events             5014
WindDirDegrees     0
dtype: int64

data.drop(['Events', 'Max Gust SpeedKm/h'], axis='columns', inplace=True)
```

After

```
data.isna().sum()

CET          0
Max TemperatureC    0
Mean TemperatureC   0
Min TemperatureC    0
Max Dew PointC     0
Mean Dew PointC    0
Min Dew PointC     0
Max Humidity        0
Mean Humidity       0
Min Humidity        0
Max Sea Level PressurehPa 0
Mean Sea Level PressurehPa 0
Min Sea Level PressurehPa 0
Max VisibilityKm    938
Mean VisibilityKm   938
Min VisibilitykM    938
Max Wind SpeedKm/h  0
Mean Wind SpeedKm/h 0
Precipitationmm     0
CloudCover         1370
WindDirDegrees     0
dtype: int64
```

Missing value

Remove rows with missing data.

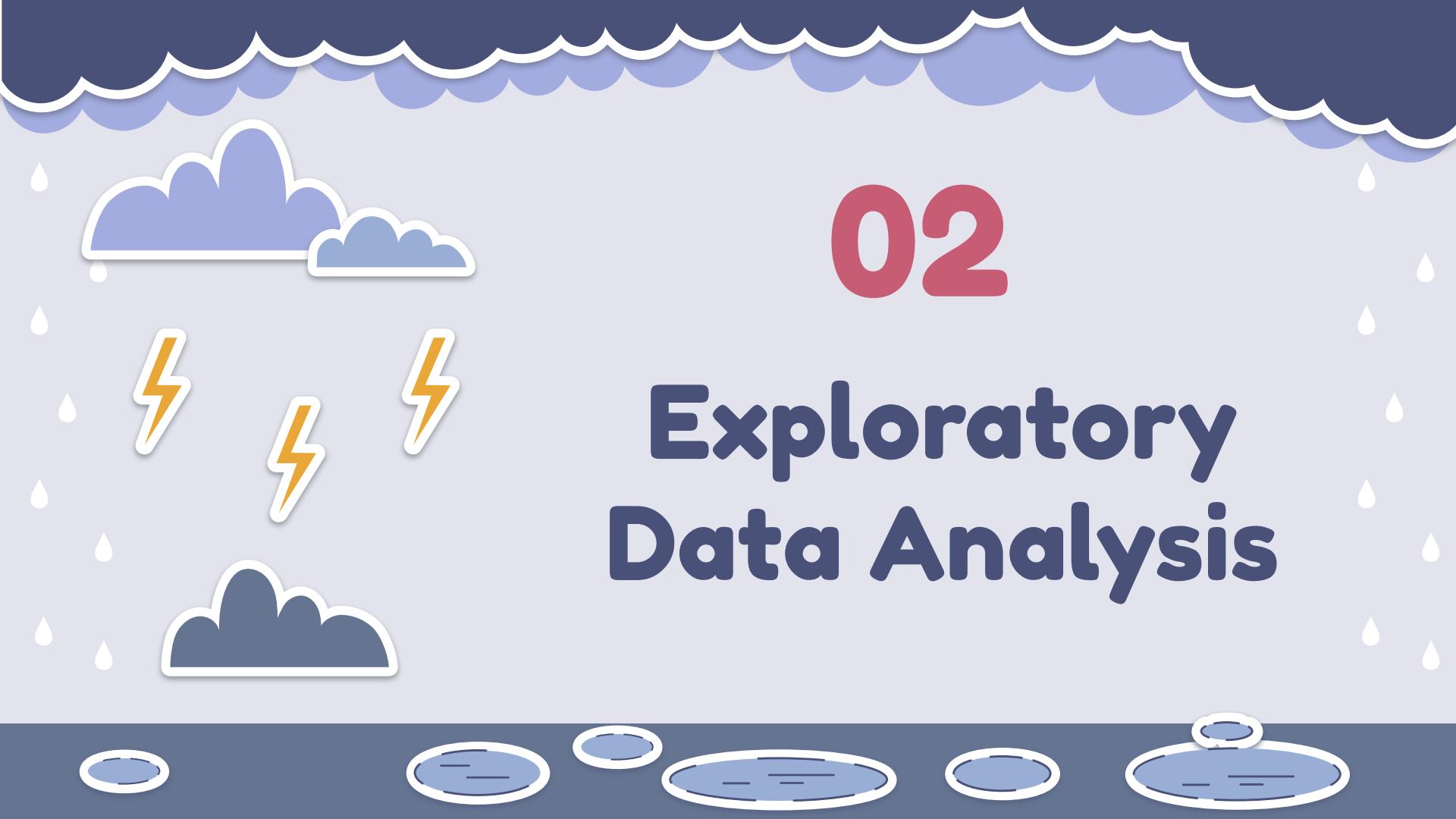
CET	Max TemperatureC	Mean TemperatureC	Min TemperatureC	Max Dew PointC	Mean Dew PointC	Min Dew PointC	Max Humidity	Mean Humidity	Min Humidity	...	Mean Sea Level PressurehPa	Min Sea Level PressurehPa	Max VisibilityKm	Visibil
1539	2001-06-25	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1019	1018	NaN	
1540	2001-06-26	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1019	1016	NaN	

2 rows × 21 columns

```
data.drop([1539, 1540], axis='index', inplace=True)
```

Populate fields with computable data.

```
temp = pd.DataFrame({'Mean TemperatureC': (data['Max TemperatureC'] + data['Min TemperatureC']) // 2})
data.fillna(value=temp, inplace=True)
```



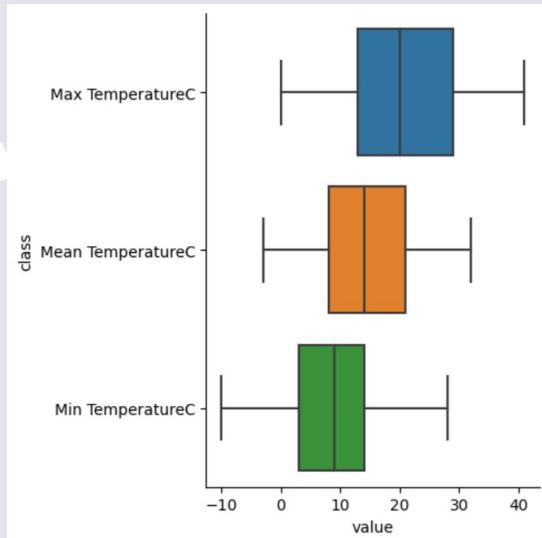
A decorative background featuring stylized white clouds at the top, followed by a layer of blue clouds. Small white raindrops fall from the sky. In the center-left, there are three yellow lightning bolts. At the bottom, there are five small blue circles with white outlines, resembling bubbles or water droplets.

02

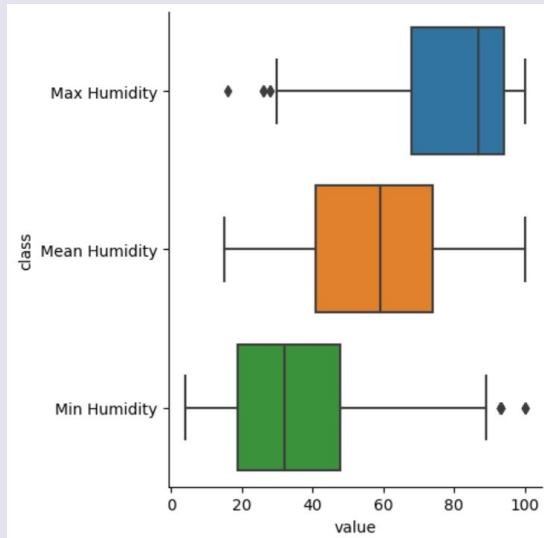
Exploratory Data Analysis

Outlier

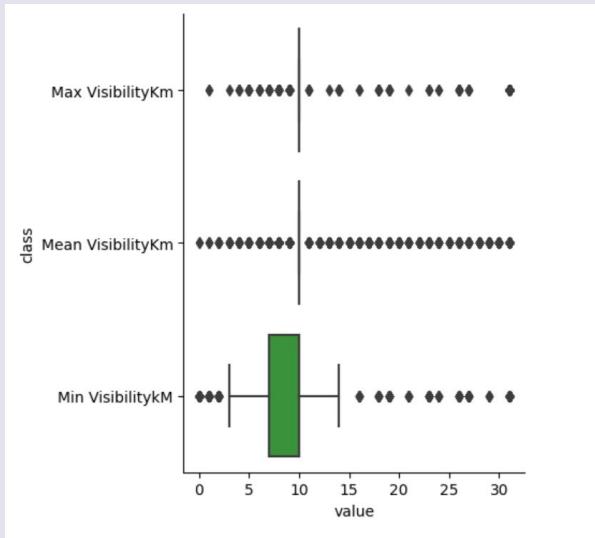
Temperature



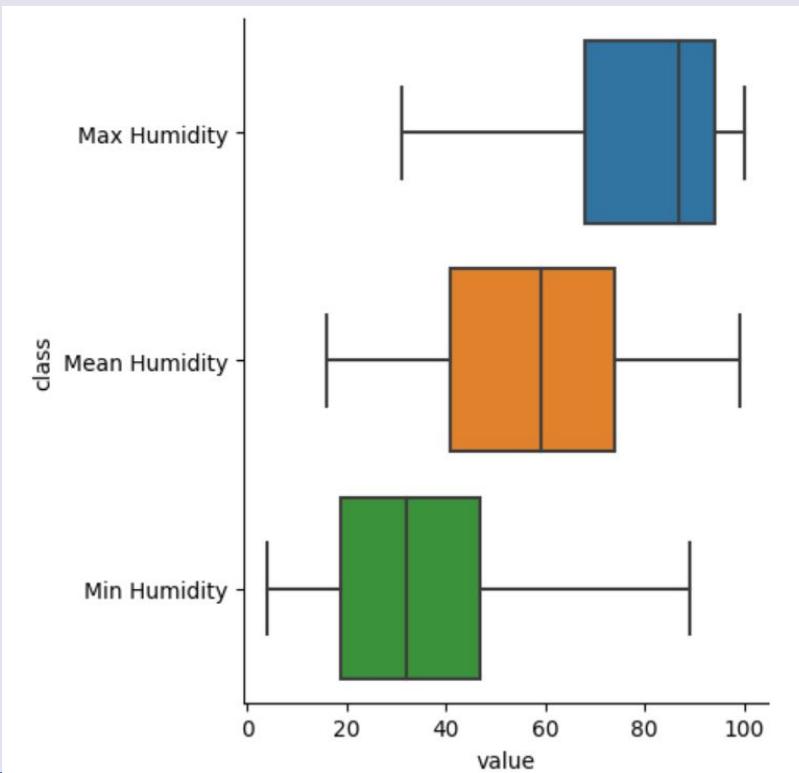
Humidity



Visibility

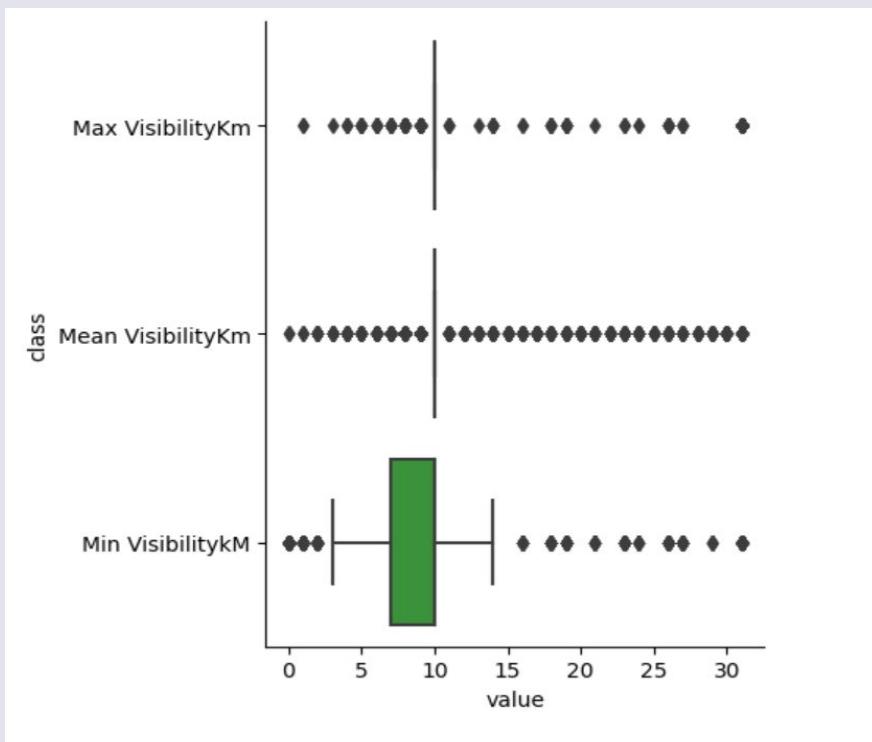


```
data = data[(data['Min Humidity'] < 90) & (data['Max Humidity'] > 30)]
humi = pd.melt(data, value_vars=data.columns[7:10], var_name='class', value_name='value')
sns.catplot(data=humi, x="value", y="class", kind="box")
```

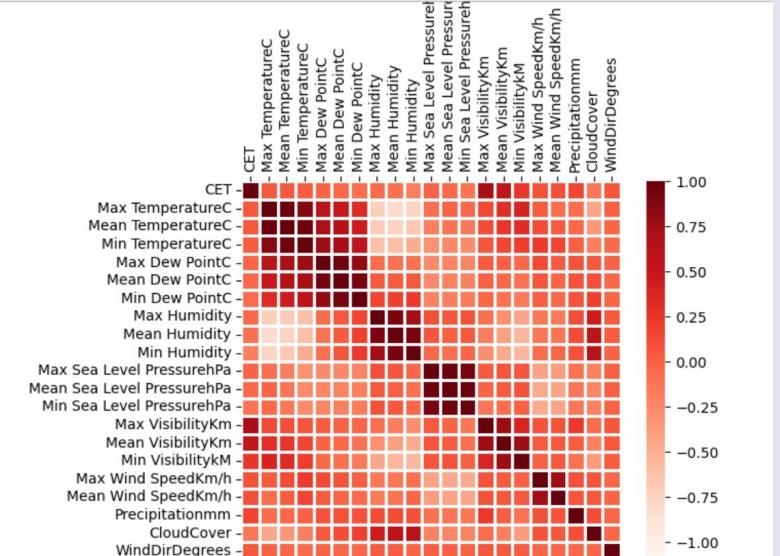


Drop outlier

Visibility



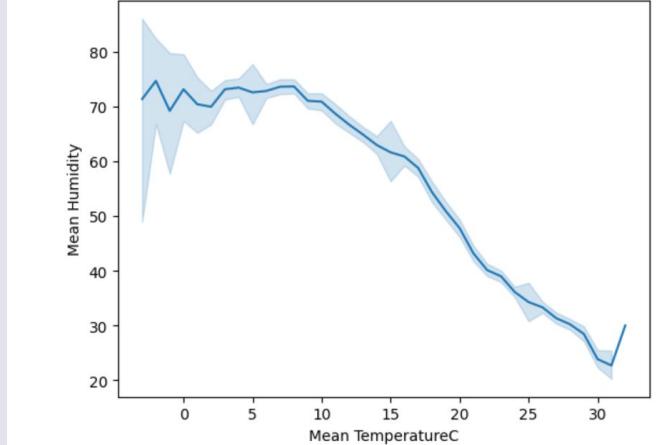
Analysis



```
pd.set_option('display.max_columns', None)
plt.tick_params(axis='both', which='major', labelsize=10, labelbottom = False, bottom=False, top = False, labeltop=True)
sns.heatmap(data.corr(), linewidths=2,vmax=1, vmin=-1, square=True, cmap='Reds').xaxis.tick_top()
```

[23]: `temp_and_humidity = data[['Mean TemperatureC', 'Mean Humidity']]
sns.lineplot(data=temp_and_humidity, x='Mean TemperatureC', y='Mean Humidity')`

[23]: <Axes: xlabel='Mean TemperatureC', ylabel='Mean Humidity'>



03

Visualization



Visualization

Examine the annual temperature data from 1997 to 2015.

```
# group data by year
focus_data_year = focus_data.groupby(focus_data.CET.dt.year)

# find mean of temperature in a month
min_temp_y = pd.DataFrame(focus_data_year['Min TemperatureC'].mean())
max_temp_y = pd.DataFrame(focus_data_year['Max TemperatureC'].mean())
mean_temp_y = pd.DataFrame(focus_data_year['Mean TemperatureC'].mean())
```

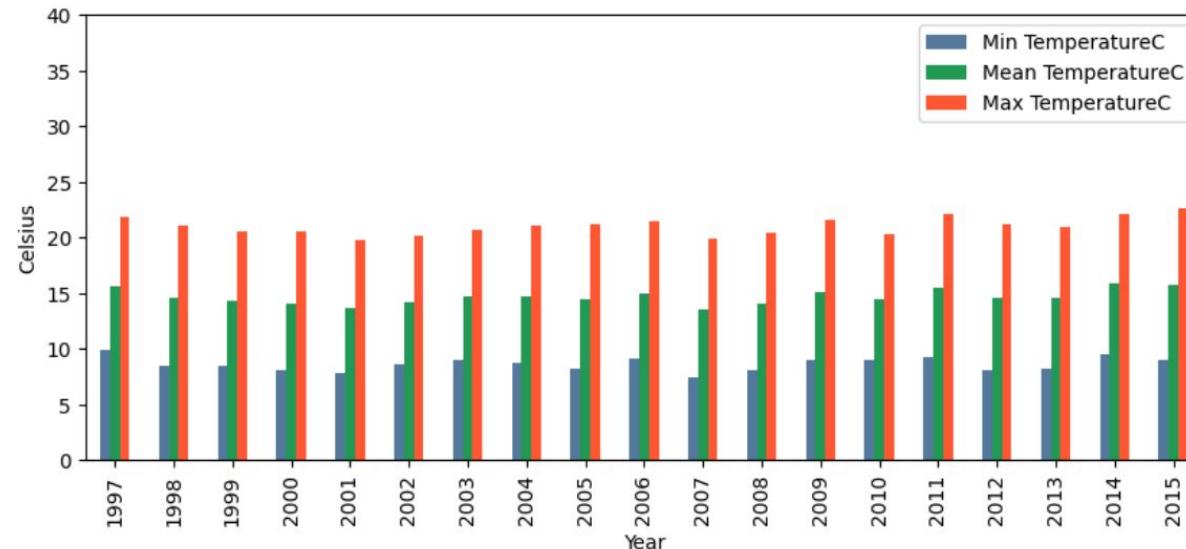
CET	Min TemperatureC	Mean TemperatureC	Max TemperatureC
1997	9.854795	15.610959	21.838356
1998	8.465753	14.556164	21.106849
1999	8.427397	14.252055	20.512329
2000	8.130112	14.104089	20.553903
2001	7.884298	13.606061	19.809917
2002	8.550685	14.169863	20.178082
2003	9.035616	14.652055	20.673973
2004	8.773810	14.744048	21.083333
2005	8.156164	14.465753	21.194521
2006	9.178082	15.027397	21.413699
2007	7.457534	13.515068	19.961644
2008	8.122951	14.073770	20.360656
2009	8.980822	15.115068	21.652055
2010	9.005479	14.380822	20.284932
2011	9.221918	15.454795	22.076712
2012	8.046448	14.527322	21.240437
2013	8.191781	14.539726	20.939726
2014	9.556164	15.808219	22.109589
2015	9.005479	15.802740	22.632877

Visualization

Presents an overview of temperature data from 1997 to 2015 in a bar chart.

```
# temperature per year
temp_y.plot(kind="bar", figsize = (10, 4), color={'Max TemperatureC': '#FF5733', 'Mean TemperatureC': '#229954', 'Min TemperatureC': '#55799D'})
plt.xlabel("Year")
plt.ylabel("Celsius")
plt.ylim(0,40)
```

(0.0, 40.0)



Visualization

A tabular overview of monthly temperatures for each year.

```
[29]: # group data by month  
focus_data_month = data.groupby(data.CET.dt.month)
```

```
[30]: # mean of max, min and mean value  
min_temp_m = pd.DataFrame(focus_data_month['Min TemperatureC'].mean())  
max_temp_m = pd.DataFrame(focus_data_month['Max TemperatureC'].mean())  
mean_temp_m = pd.DataFrame(focus_data_month['Mean TemperatureC'].mean())
```

```
[31]: # group three data  
temp_m = pd.concat([min_temp_m, mean_temp_m, max_temp_m], axis=1)  
temp_m
```

	Min TemperatureC	Mean TemperatureC	Max TemperatureC
--	------------------	-------------------	------------------

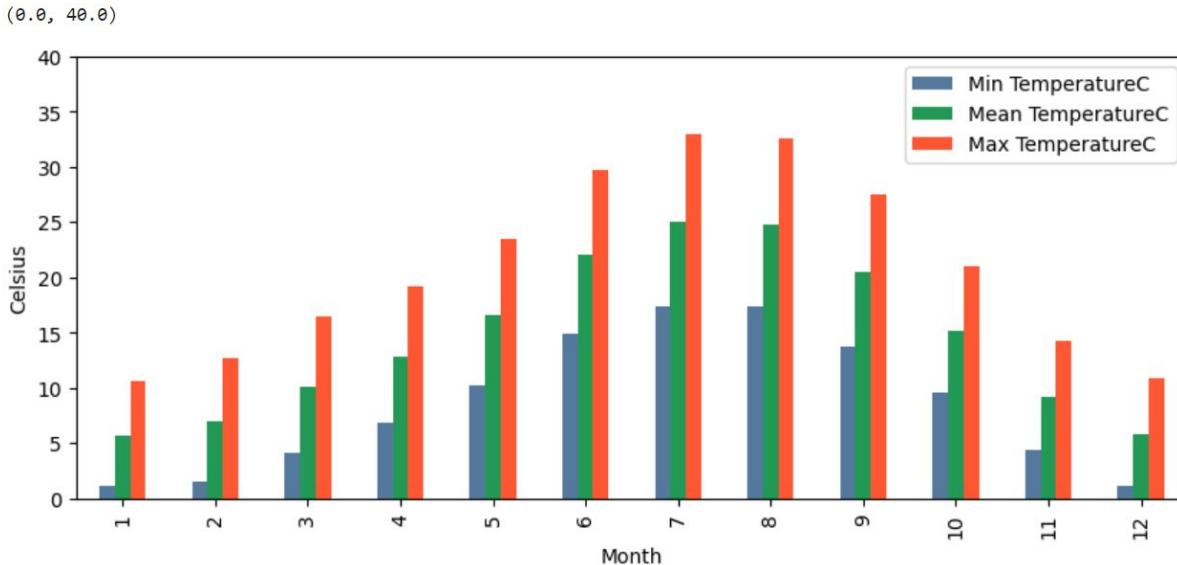
CET

1	1.107335	5.688730	10.588551
2	1.523629	6.947070	12.752363
3	4.207885	10.148746	16.526882
4	6.931481	12.859259	19.168519
5	10.220137	16.667235	23.455631
6	14.898214	22.116071	29.723214
7	17.455008	25.074703	33.003396
8	17.395797	24.789842	32.572680
9	13.826316	20.494737	27.550877
10	9.614601	15.152801	21.032258
11	4.463158	9.196491	14.270175
12	1.174873	5.859083	10.845501

Visualization

A bar chart illustrating the monthly temperature averages for each year.

```
# temperature in latest year
temp_m.plot(kind="bar", figsize = (10, 4), color=['Max TemperatureC': '#FF5733', 'Mean TemperatureC': '#229954', 'Min TemperatureC': '#55799D'])
# temp_m.plot(kind="bar", figsize = (10, 4), color=['blue', 'orange'])
plt.xlabel("Month")
plt.ylabel("Celsius")
plt.ylim(0,40)
```



Visualization

```
vis_analyze = data[(data['CET'] >= '2013-01-01')]

# mean_fog = vis_analyze[vis_analyze['Mean VisibilityKm'] <= 5]
min_fog = vis_analyze[vis_analyze['Min VisibilitykM'] <= 5]
min_fog
```

```
# mean_fog = mean_fog.groupby(mean_fog.CET.dt.month).count()
min_fog = min_fog.groupby(min_fog.CET.dt.month)['Min VisibilitykM'].count()
min_fog
```

```
CET
1    36
2    21
3    22
4    14
5     2
6     2
7     2
8     2
9     6
10   19
11   25
12   30
```

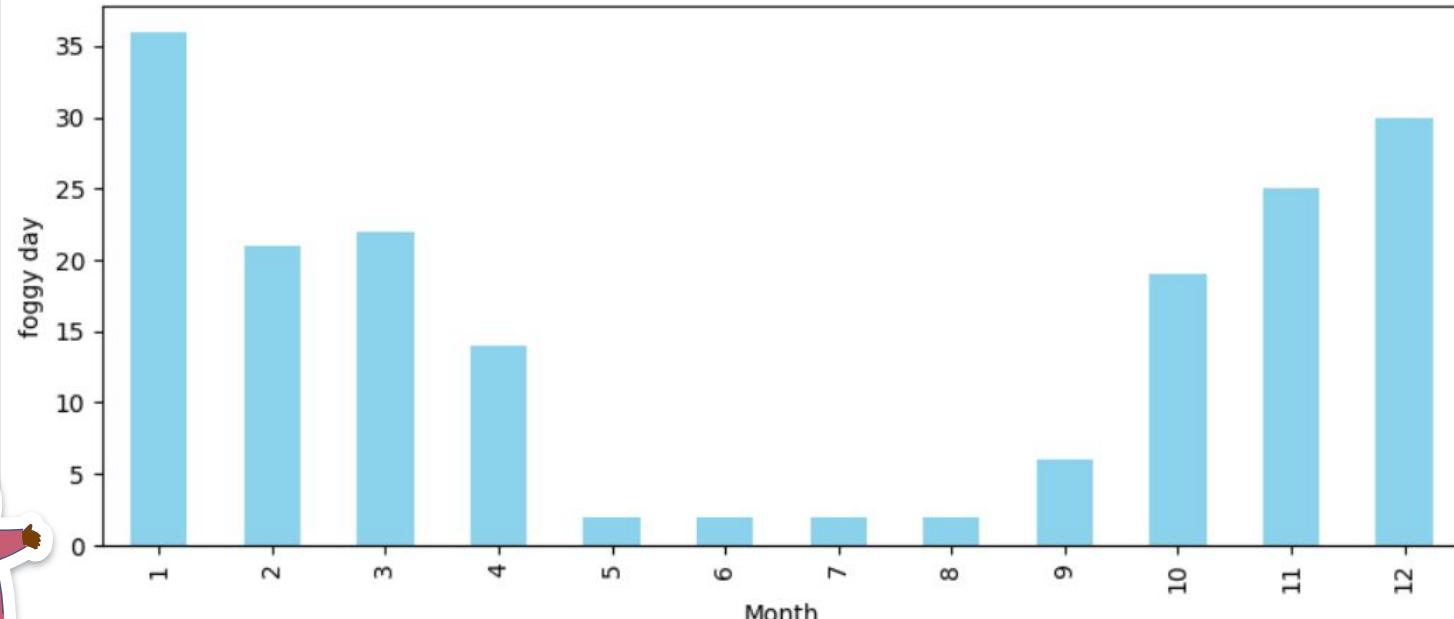
```
Name: Min VisibilitykM, dtype: int64
```

Visualization

This presents an overview of visibility from 2013 to 2015 displayed in a bar chart.

```
min_fog.plot(kind="bar", figsize = (10, 4), color="#8BD2EC")
plt.xlabel("Month")
plt.ylabel("foggy day")

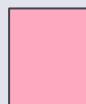
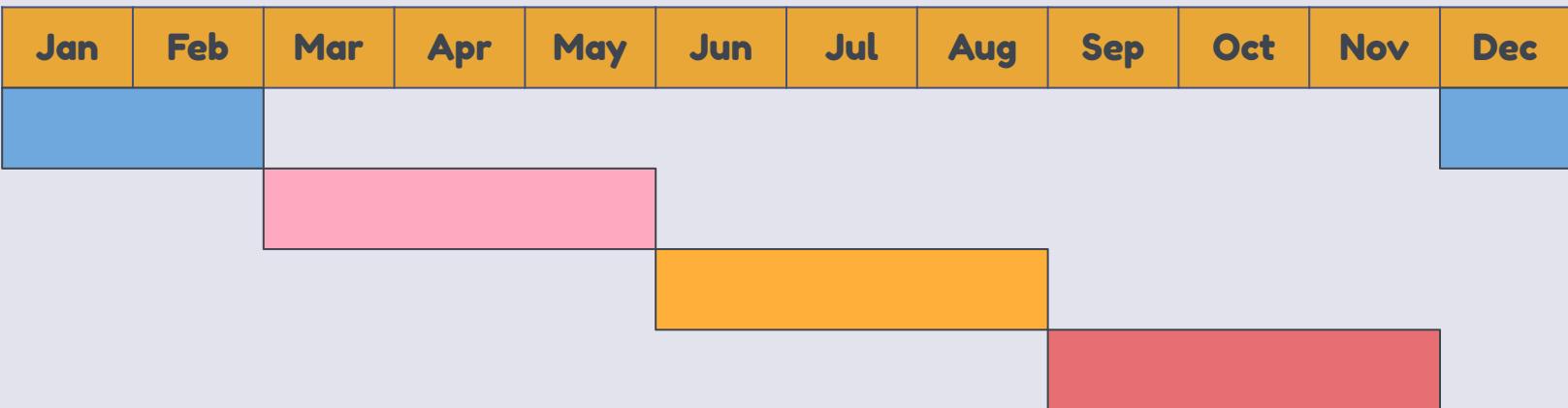
Text(0, 0.5, 'foggy day')
```



Climates of the year

January	February	March	April
Frosty	Frosty	Cold	Cold
May	June	July	August
Warm	Warm	Hot	Hot
September	October	November	December
Warm	Warm	Cold	Frosty

Seasons graph



Spring

Venus is hot



Autumn

Jupiter is big



Summer

Mars is cold



Winter

Earth has life



Recommend place during each season in Madrid



Winter



Naviluz Bus

Spring



El Retiro Park

Summer



Terraza Florida Retiro

Autumn



Museo Nacional Del Prado

**THANK YOU FOR WATCHING AND
LISTENING.**

