Q6. Consider any sales training/ weather forecasting dataset

In [1]:
```
import pandas as pd
import numpy as np
from random import sample
import os
from dateutil.parser import parse

df = pd.read_csv("F:\CS\sem 5\Data Analysis and Visualisation\Practicals\weather
df
```

Out[1]:

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	W Bear (degr
0	2006-04-01 00:00:00+02:00	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	25
1	2006-04-01 01:00:00+02:00	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	25
2	2006-04-01 02:00:00+02:00	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	20
3	2006-04-01 03:00:00+02:00	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	20
4	2006-04-01 04:00:00+02:00	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	25
...	...	...	...	...	...	...	...	
96448	2016-09-09 19:00:00+02:00	Partly Cloudy	rain	26.016667	26.016667	0.43	10.9963	3
96449	2016-09-09 20:00:00+02:00	Partly Cloudy	rain	24.583333	24.583333	0.48	10.0947	2
96450	2016-09-09 21:00:00+02:00	Partly Cloudy	rain	22.038889	22.038889	0.56	8.9838	3
96451	2016-09-09 22:00:00+02:00	Partly Cloudy	rain	21.522222	21.522222	0.60	10.5294	2
96452	2016-09-09 23:00:00+02:00	Partly Cloudy	rain	20.438889	20.438889	0.61	5.8765	3

96453 rows × 12 columns

a. Compute mean of a series grouped by another series

In [2]:
```
df[['Humidity']].groupby(df['Summary']).mean()
```

Out[2]:

Summary	Humidity
Breezy	0.637778
Breezy and Dry	0.260000
Breezy and Foggy	0.938571
Breezy and Mostly Cloudy	0.637054
Breezy and Overcast	0.763144
Breezy and Partly Cloudy	0.545803
Clear	0.729708
Dangerously Windy and Partly Cloudy	0.490000
Drizzle	0.867949
Dry	0.230294
Dry and Mostly Cloudy	0.242143
Dry and Partly Cloudy	0.240814
Foggy	0.950765
Humid and Mostly Cloudy	0.874250
Humid and Overcast	0.881429
Humid and Partly Cloudy	0.848824
Light Rain	0.888095
Mostly Cloudy	0.725069
Overcast	0.837232
Partly Cloudy	0.648571
Rain	0.947000
Windy	0.572500
Windy and Dry	0.240000
Windy and Foggy	0.900000
Windy and Mostly Cloudy	0.600000
Windy and Overcast	0.708667
Windy and Partly Cloudy	0.528806

b. Fill an intermittent time series to replace all missing dates with values of previous non-missing date.

In [3]:
```
df['Formatted Date'].isnull().sum()
data = df.copy()

random_indices = data['Formatted Date'].sample(1000).index
data.loc[random_indices,'Formatted Date'] = np.nan

data['Formatted Date'].isna().value_counts()
data.fillna(method = 'ffill').loc[random_indices, 'Formatted Date'].head(5)
```

Out[3]:
```
69306    2013-10-04 17:00:00+02:00
71530    2014-08-06 09:00:00+02:00
52037    2011-09-15 04:00:00+02:00
51522    2011-10-23 18:00:00+02:00
28474    2009-12-07 09:00:00+01:00
Name: Formatted Date, dtype: object
```

c. Perform appropriate year-month string to dates conversion.

In [5]:
```
temp = df.copy()
dummy_yearMonths = ['Jan 2005', 'Feb 2016', 'Mar 2017', 'Apr 2018', 'May 2019',
temp_size = len(temp)
temp['Dummy_Date'] = [sample(dummy_yearMonths,1)[0]

for i in range(temp_size)]
temp
```

Out[5]:

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	W Bear (degr
0	2006-04-01 00:00:00+02:00	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	25
1	2006-04-01 01:00:00+02:00	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	25
2	2006-04-01 02:00:00+02:00	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	20
3	2006-04-01 03:00:00+02:00	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	20
4	2006-04-01 04:00:00+02:00	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	25
...	...	...	...	...	...	...	...	
96448	2016-09-09 19:00:00+02:00	Partly Cloudy	rain	26.016667	26.016667	0.43	10.9963	
96449	2016-09-09 20:00:00+02:00	Partly Cloudy	rain	24.583333	24.583333	0.48	10.0947	
96450	2016-09-09 21:00:00+02:00	Partly Cloudy	rain	22.038889	22.038889	0.56	8.9838	
96451	2016-09-09 22:00:00+02:00	Partly Cloudy	rain	21.522222	21.522222	0.60	10.5294	
96452	2016-09-09 23:00:00+02:00	Partly Cloudy	rain	20.438889	20.438889	0.61	5.8765	

96453 rows × 13 columns

In [6]:
```
temp['Dummy_Date'].map(lambda d: parse(d))
```

Out[6]:
```
0       2019-05-09
1       2016-02-09
2       2021-06-09
3       2019-05-09
4       2019-05-09
           ...
96448   2017-03-09
96449   2017-03-09
96450   2005-01-09
96451   2005-01-09
96452   2021-06-09
Name: Dummy_Date, Length: 96453, dtype: datetime64[ns]
```

d. Split a dataset to group by two columns and then sort the aggregated results within the groups.

In [9]:
```
def sort_values(df, column='Summary', ascending = True):
    return df.sort_values(by=column, ascending = ascending)

grouped = df.groupby(['Summary', 'Precip Type'])
temp = grouped.apply(sort_values, column = 'Temperature (C)', ascending = False)
temp['Temperature (C)']
```

Out[9]:
```
Summary                   Precip Type
Breezy                    rain         12805    37.588889
                                       30183    33.888889
                                       53776    30.900000
                                       53777    29.938889
                                       45035    29.738889
                                                   ...
Windy and Partly Cloudy   rain         63542     3.983333
                                       63540     3.911111
                                       63539     2.872222
                                       31833     2.222222
                                       31832     1.111111
Name: Temperature (C), Length: 95936, dtype: float64
```

e. Split a given dataframe into groups with bin counts.

In [10]:
```
df.groupby(['Summary']).count()
df.groupby(['Summary']).size()
```

Out[10]:

Summary	
Breezy	54
Breezy and Dry	1
Breezy and Foggy	35
Breezy and Mostly Cloudy	516
Breezy and Overcast	528
Breezy and Partly Cloudy	386
Clear	10890
Dangerously Windy and Partly Cloudy	1
Drizzle	39
Dry	34
Dry and Mostly Cloudy	14
Dry and Partly Cloudy	86
Foggy	7148
Humid and Mostly Cloudy	40
Humid and Overcast	7
Humid and Partly Cloudy	17
Light Rain	63
Mostly Cloudy	28094
Overcast	16597
Partly Cloudy	31733
Rain	10
Windy	8
Windy and Dry	1
Windy and Foggy	4
Windy and Mostly Cloudy	35
Windy and Overcast	45
Windy and Partly Cloudy	67

dtype: int64