Time Series

Sample Data

Sample Data

```
In [36]: df = pd.read_csv('data/cafe.csv', parse_dates=['date'])
         df.head(4)
```

Out[36]:

	date	temperature	sold_icecream	sold_cups_coffee	sold_coke
0	2018-06-29	28	40	57	44
1	2018-06-30	25	36	61	19
2	2018-07-01	31	45	53	15
3	2018-07-02	31	47	52	26

Set Time Index

Set Time Index

In [38]:

```
ts = df.set_index('date')b
ts.sample(4)
```

Out[38]:

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-07-01	31	45	53	15
2018-07-13	29	44	55	35
2018-07-05	28	42	56	22
2018-08-01	27	41	59	23

Resample

- Downsample and/or unify the frequency of the time steps
- Need to aggregate the values (sum, mean)

Resample

- Downsample and/or unify the frequency of the time steps
- Need to aggregate the values (sum, mean)

```
In [40]: ts.resample('W').mean()
```

```
Out[40]:
```

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-07-01	28.000000	40.333333	57.000000	26.000000
2018-07-08	27.142857	40.714286	56.428571	27.000000
2018-07-15	28.285714	42.571429	56.000000	35.428571
2018-07-22	26.142857	39.142857	58.714286	32.857143
2018-07-29	29.000000	44.285714	55.142857	28.285714
2018-08-05	28.000000	42.000000	57.250000	28.500000

Resample

- Downsample and/or unify the frequency of the time steps
- Need to aggregate the values (sum, mean)

```
In [40]: ts.resample('W').mean()
```

Out[40]:

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-07-01	28.000000	40.333333	57.000000	26.000000
2018-07-08	27.142857	40.714286	56.428571	27.000000
2018-07-15	28.285714	42.571429	56.000000	35.428571
2018-07-22	26.142857	39.142857	58.714286	32.857143
2018-07-29	29.000000	44.285714	55.142857	28.285714
2018-08-05	28.000000	42.000000	57.250000	28.500000

- Many frequency codes available
  - e.g. D: Calendar day, B: Business day, W: Weekly, M: Month end, Q: Quarter end, A: Year end, H: Hours, T: Minutes

In [41]:

```
ts.resample('W-MON').mean()
```

Out[41]:

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-07-02	28.750000	42.000000	55.750000	26.000000
2018-07-09	26.428571	39.714286	57.285714	26.000000
2018-07-16	29.000000	43.571429	55.285714	36.857143
2018-07-23	25.857143	38.857143	59.000000	34.142857
2018-07-30	29.142857	44.142857	55.000000	27.285714
2018-08-06	27.333333	41.333333	58.333333	27.666667

Filtering

Filtering

```
In [42]: ts['2018-08']
```

Out[42]:

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-08-01	27	41	59	23
2018-08-02	26	39	61	44

Filtering

In [42]:
```
ts['2018-08']
```

Out[42]:

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-08-01	27	41	59	23
2018-08-02	26	39	61	44

In [43]:
```
ts['2018-07-13':'2018-07-14']
```

Out[43]:

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-07-13	29	44	55	35
2018-07-14	29	43	56	44

In [44]:

```
# Monday=0, Sunday=6
ts[ts.index.weekday==6]
```

Out[44]:

	temperature	sold_icecream	sold_cups_coffee	sold_coke
date				
2018-07-01	31	45	53	15
2018-07-08	24	35	61	19
2018-07-15	28	42	56	36
2018-07-22	25	36	60	33
2018-07-29	27	42	57	38

Exercise 7 - Time Series

Goals:

- Work with time series data

Tasks:

- Use the Rossmann dataset
- Set the 'Date' column as index
- Determine mean sales on Mondays and compare with mean sales on Sundays
- Plot sales per month
- Plot sales per quarter, grouped by store type