20250616_01

June 16, 2025

[19]:
```
# Get a basic idea about time series
import pandas as pd
```

[20]:
```
# Create a little dataframe
data = pd.DataFrame({'Date':['2025-06-01', '2025-06-02', '2025-06-03',␣
 ↪'2025-06-04', '2025-06-05'],
                     'Value':[100, 110, 105, 95, 100]})
```

[21]:
```
# Convert to datetime
data['Date'] = pd.to_datetime(data['Date'])
```

[22]:
```
# Extract components
data['Year'] = data['Date'].dt.year
data['Month'] = data['Date'].dt.month
data['DayofWeek'] = data['Date'].dt.day_name()
```

[23]:
```
print(data)
```

```
        Date  Value  Year  Month  DayofWeek
0 2025-06-01    100  2025      6     Sunday
1 2025-06-02    110  2025      6     Monday
2 2025-06-03    105  2025      6    Tuesday
3 2025-06-04     95  2025      6  Wednesday
4 2025-06-05    100  2025      6   Thursday
```

[36]:
```
# Get a hand on resample
weekly = data.resample('W', on = 'Date')['Value'].mean()
print(weekly)
```

```
Date
2025-06-01    100.0
2025-06-08    102.5
Freq: W-SUN, Name: Value, dtype: float64
```

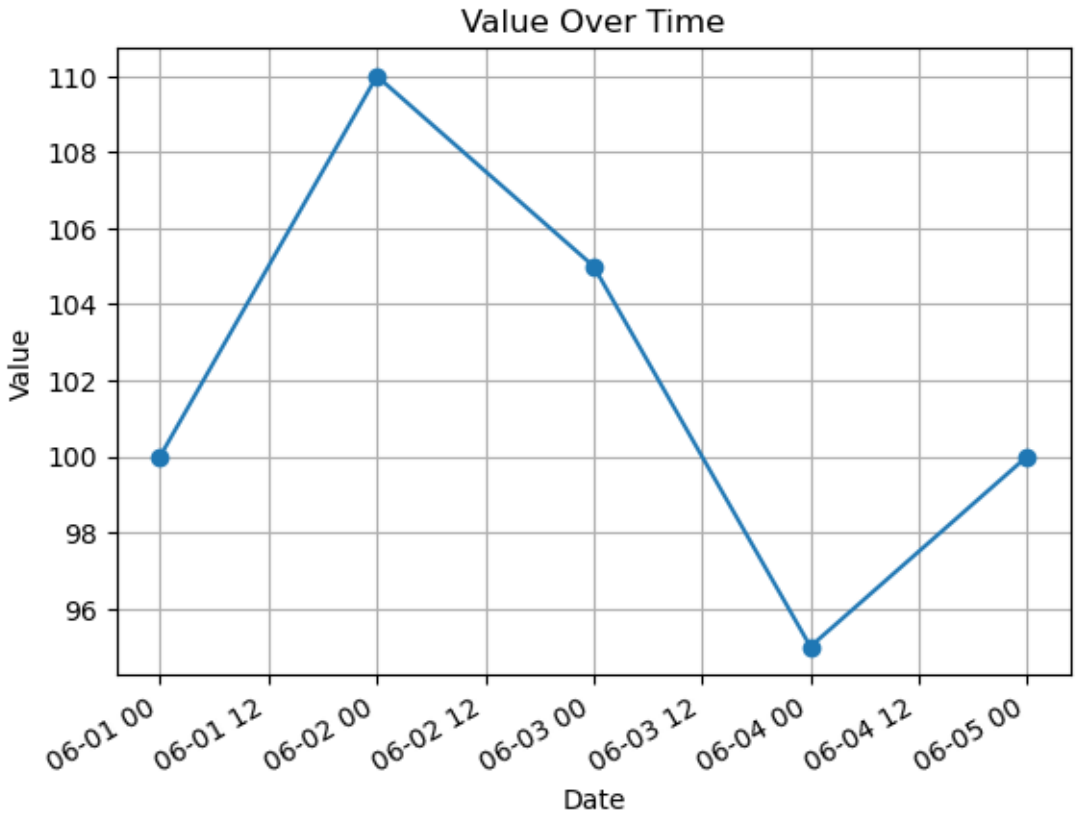
[38]:
```
# Because resample 'W' is default to count to every Sunday.
# So 2025-06-01 itself is a week, hence 100.
# And 2025-06-08 only contains 4 input and average to 102.5
# Can also use 'W-MON' to force every week end on Monday etc.
```

[45]:
```
# Visualize
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

plt.plot(data['Date'], data['Value'], marker = 'o')
plt.title("Value Over Time")
plt.xlabel("Date")
plt.ylabel("Value")
plt.grid(True)

# gcf = Get Current Figure (so it has to be after) (inculde in matplotlib.
 ↪pyplot)
# autofmt = auto format (quite self-explanatory)
# xdate = x-axis date
plt.gcf().autofmt_xdate()  # Rotate date labels

plt.show()
```



[ ]: