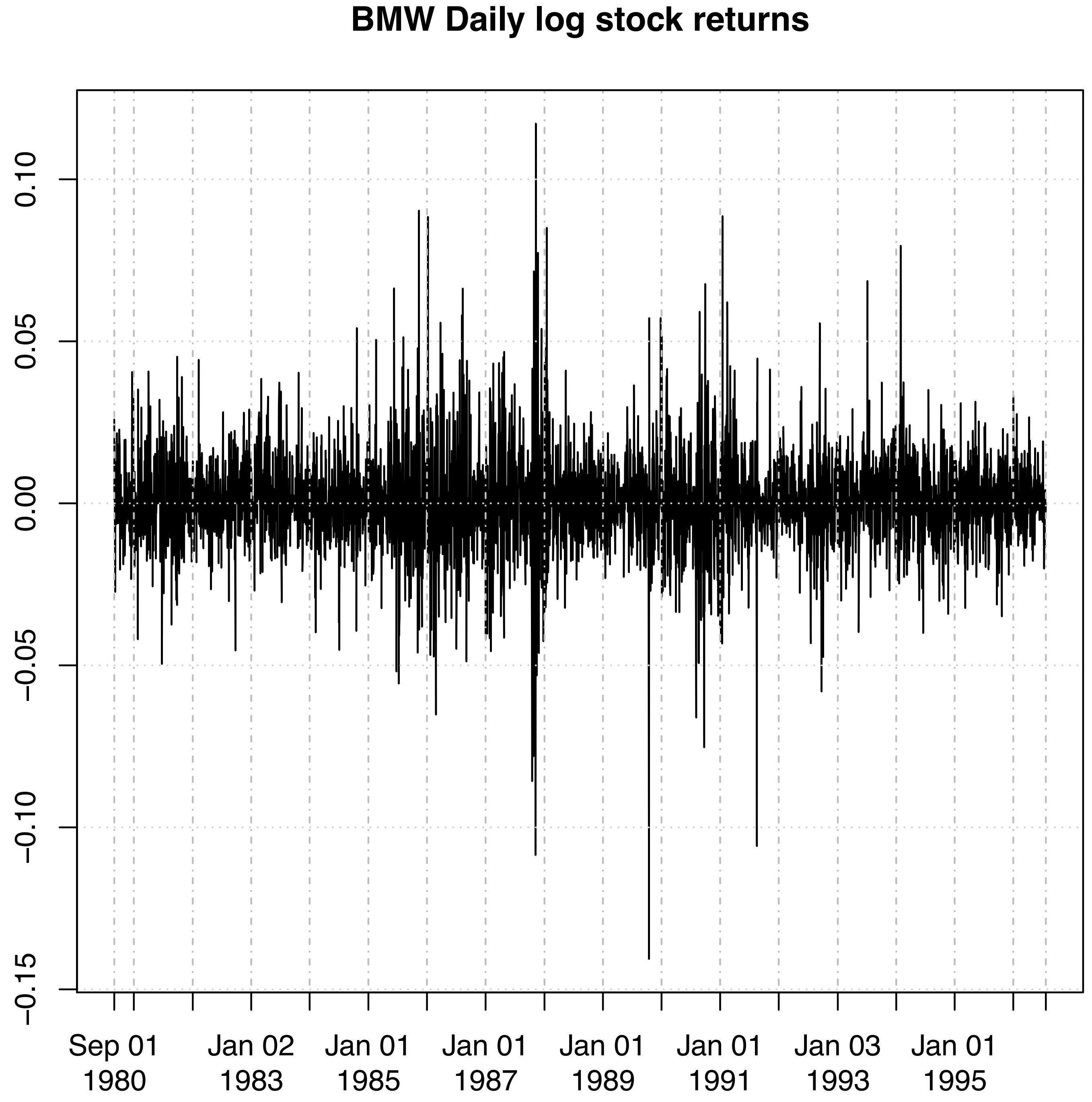
INTRODUCTION TO TIME SERIES ANALYSIS

Welcome to the Course!

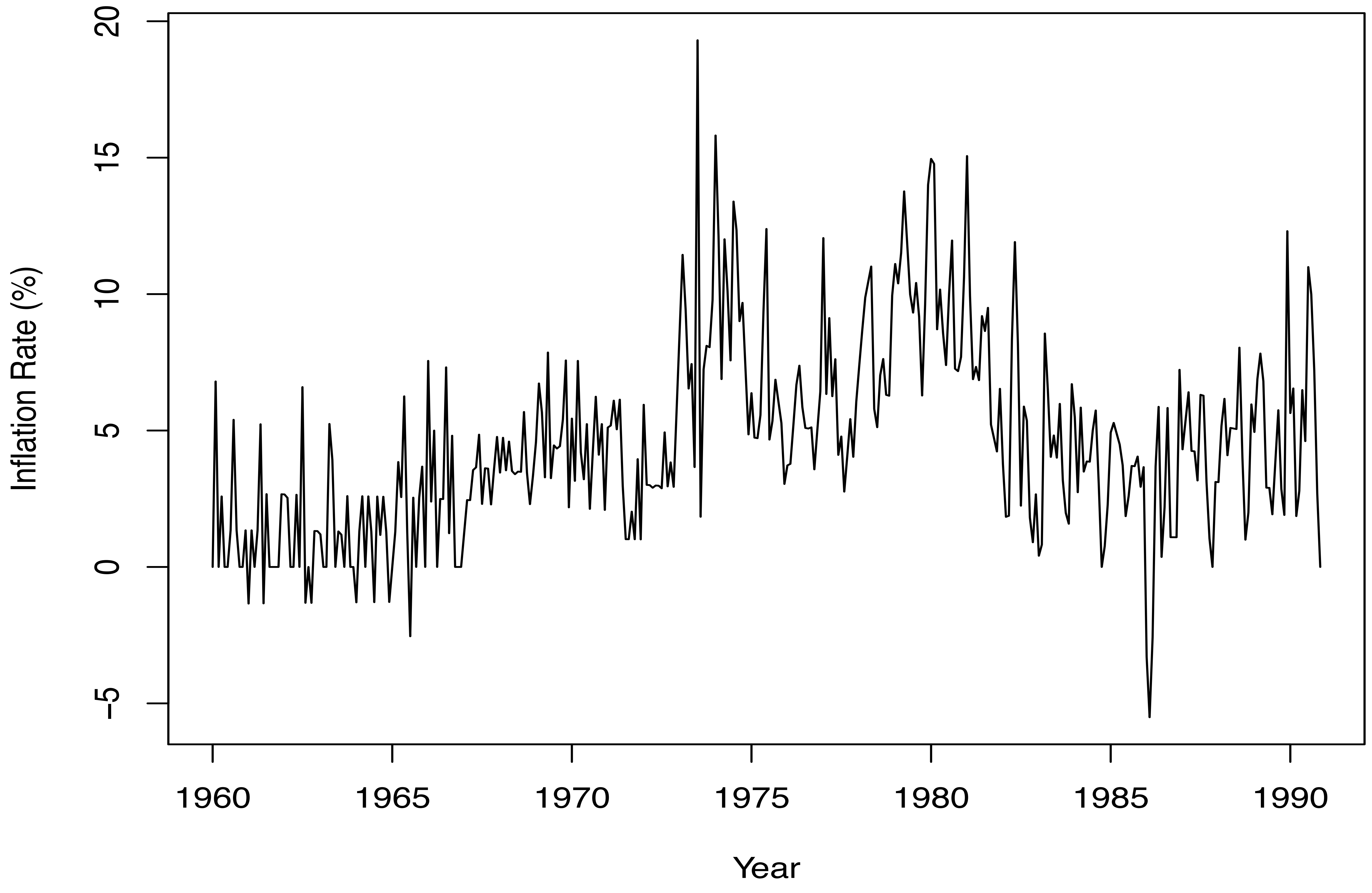
Introduction

- **Time Series:** A sequence of data in chronological order
- Data is commonly recorded sequentially, over time
- Time series data is everywhere



Time Series Example

Monthly values of the Consumer Price Index (CPI):



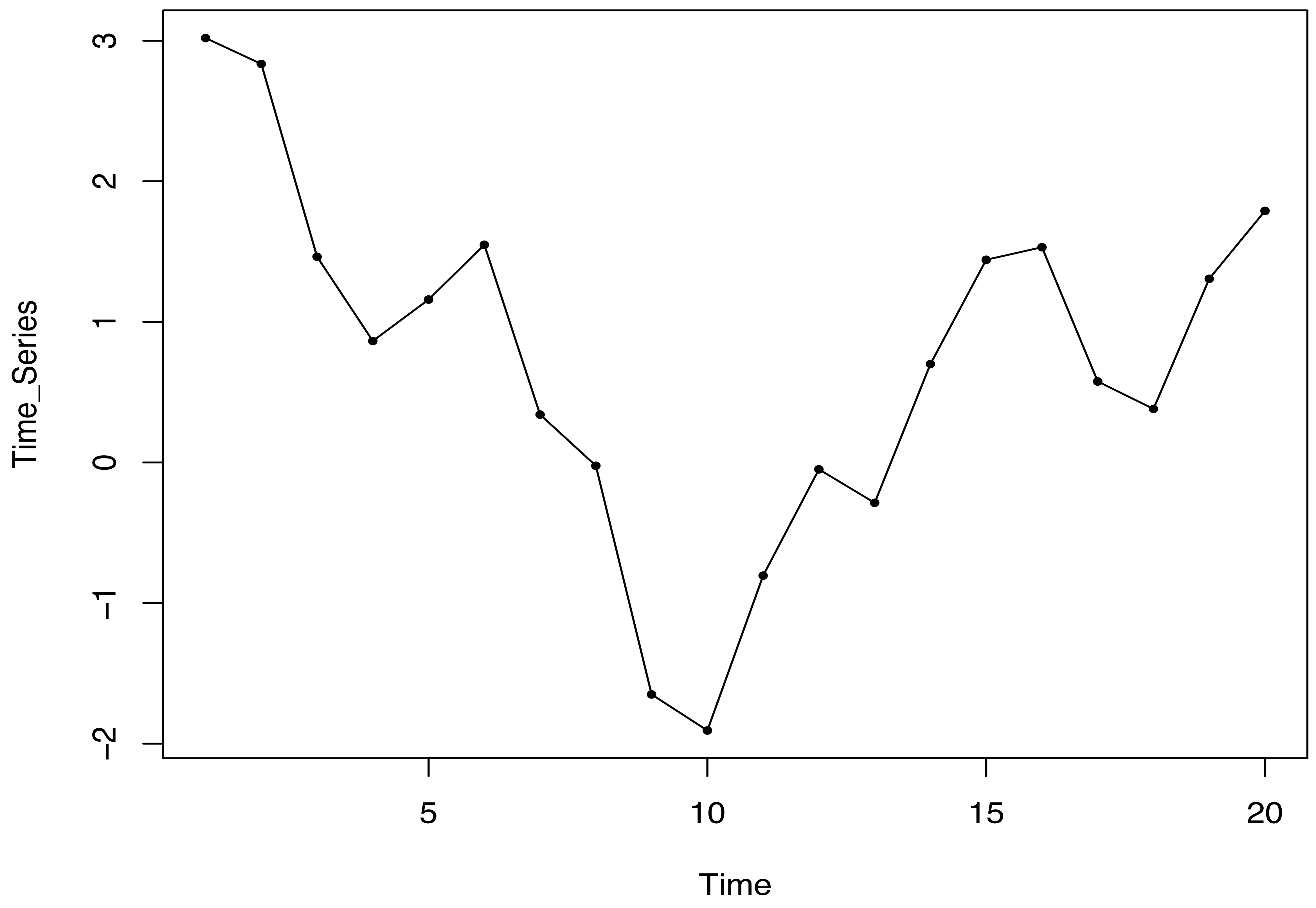
Time Series Data

- Time series data is dated or time stamped in R

```
> print(BMW_data)
...
1996-07-08   0.002
1996-07-09  -0.006
1996-07-10  -0.016
1996-07-11  -0.020
1996-07-14  -0.006
1996-07-15  -0.014
1996-07-16   0.002
1996-07-17  -0.001
...
```

Time Series Plots

```
> plot(Time_Series)
```



Basic Time Series Models

- White Noise (WN)
- Random Walk (RW)
- Autoregression (AR)
- Simple Moving Average (MA)

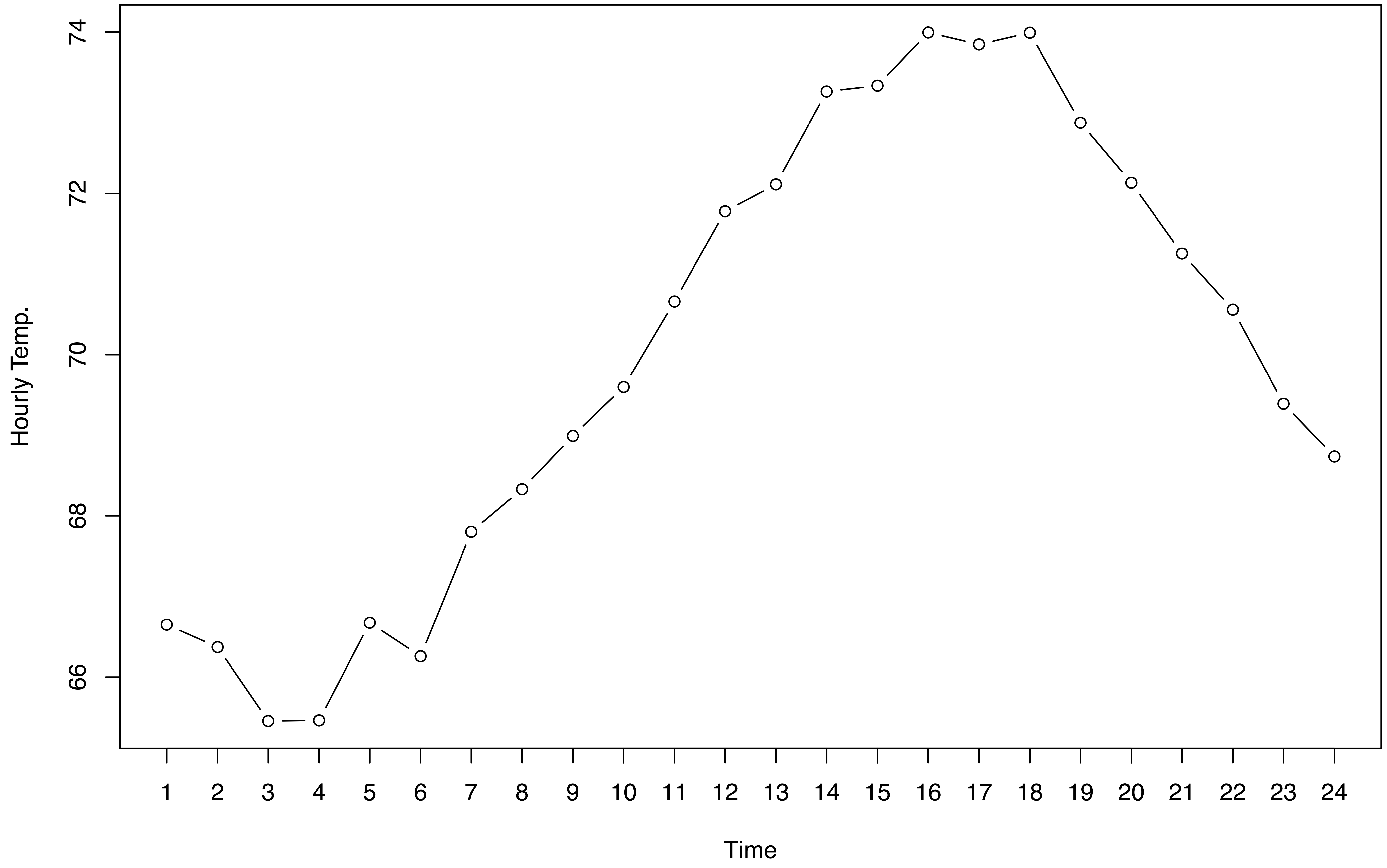
Let’s practice!

INTRODUCTION TO TIME SERIES ANALYSIS

Sampling Frequency

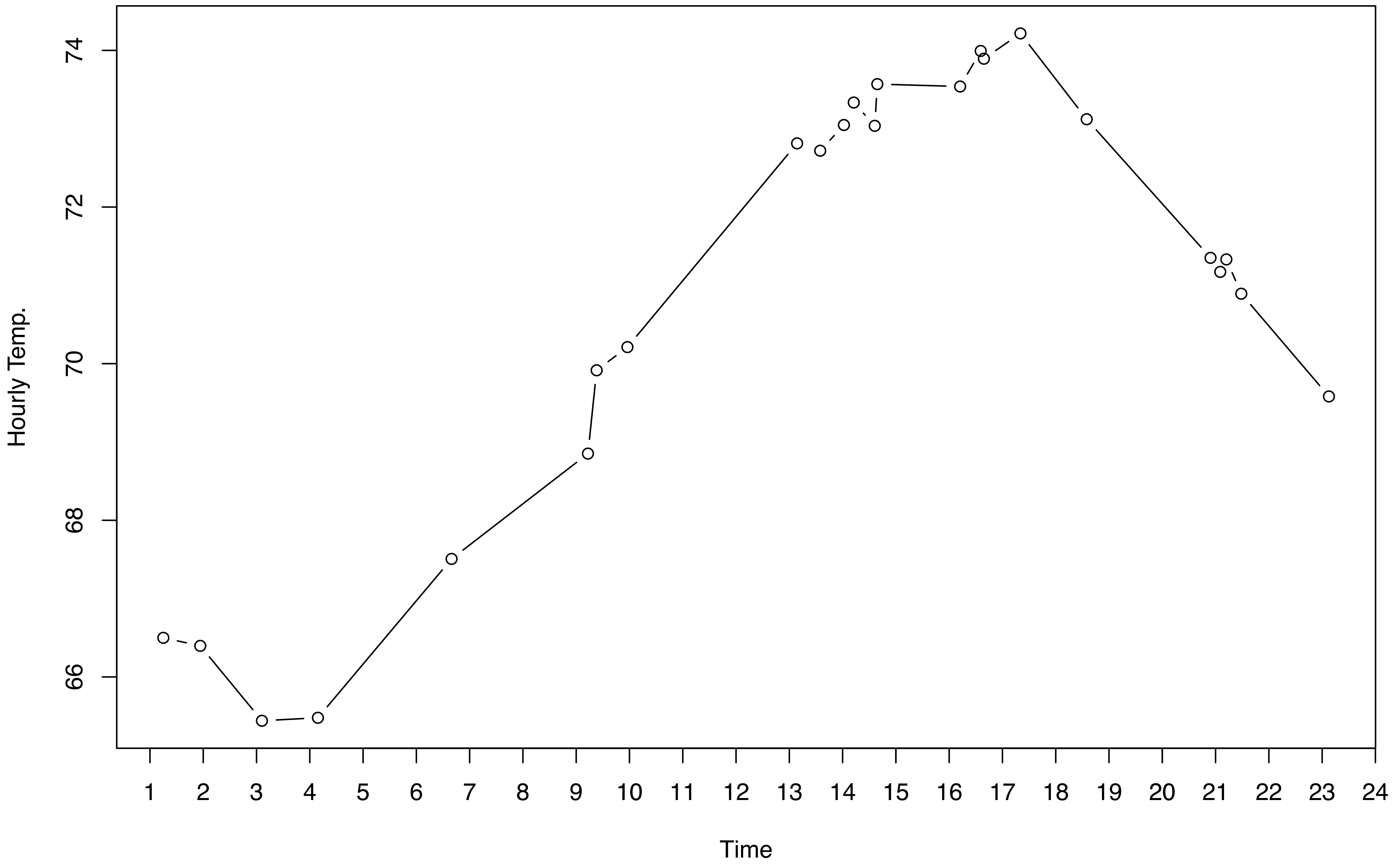
Sampling Frequency: Exact

- Some time series data is *exactly* evenly spaced



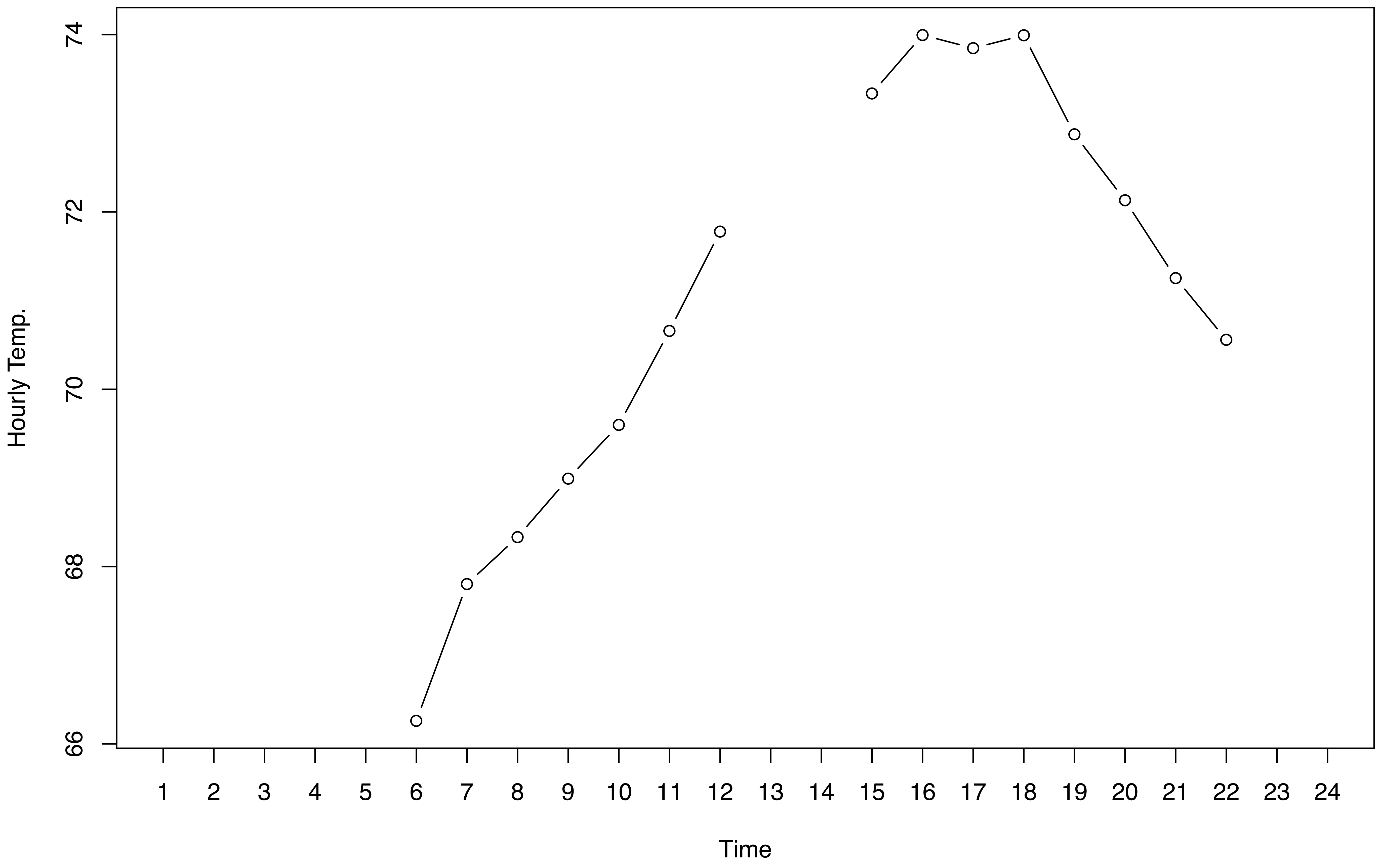
Sampling Frequency: Approximate

- Some time series data is only *approximately* evenly spaced



Sampling Frequency: Missing Values

- Some time series data is evenly spaced, but with missing values



Basic Assumptions

Simplifying assumptions for time series:

- Consecutive observations are equally spaced
- Apply a discrete-time observation index
- This may only hold approximately

Ex. Daily log returns on stock may only be available for weekdays.

Ex. Monthly CPI values are equally spaced by month, not by day.

Sampling Frequency: R Functions

- R functions: `start()`, `end()`, `frequency()`, `deltat()`

```
> start(Hourly_series)
[1] 1 1
```

```
> end(Hourly_series)
[1] 1 24
```

```
> frequency(Hourly_series)
[1] 24
```

```
> deltat(Hourly_series)
[1] 0.0417
```

Let’s practice!

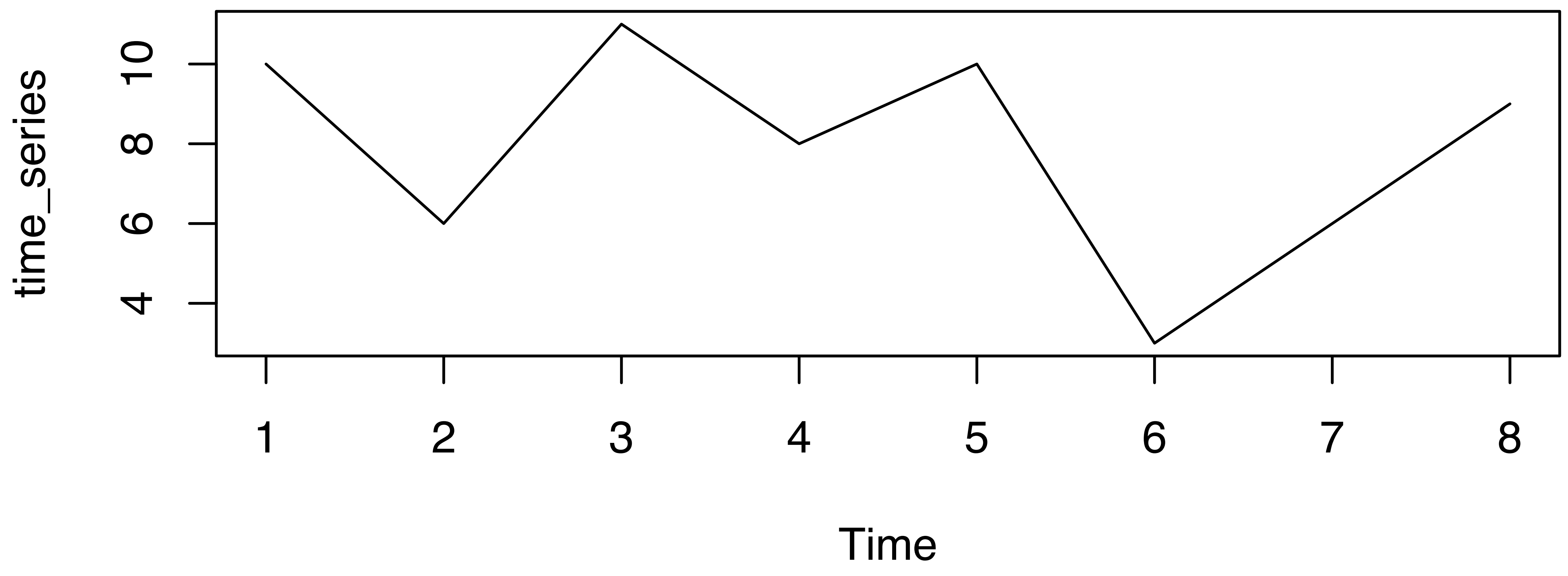
Basic Time Series Objects

Building ts() Objects - I

- Start with a vector of data
- Apply the ts() function

```
> data_vector
[1] 10 6 11 8 10 3 6 9
```

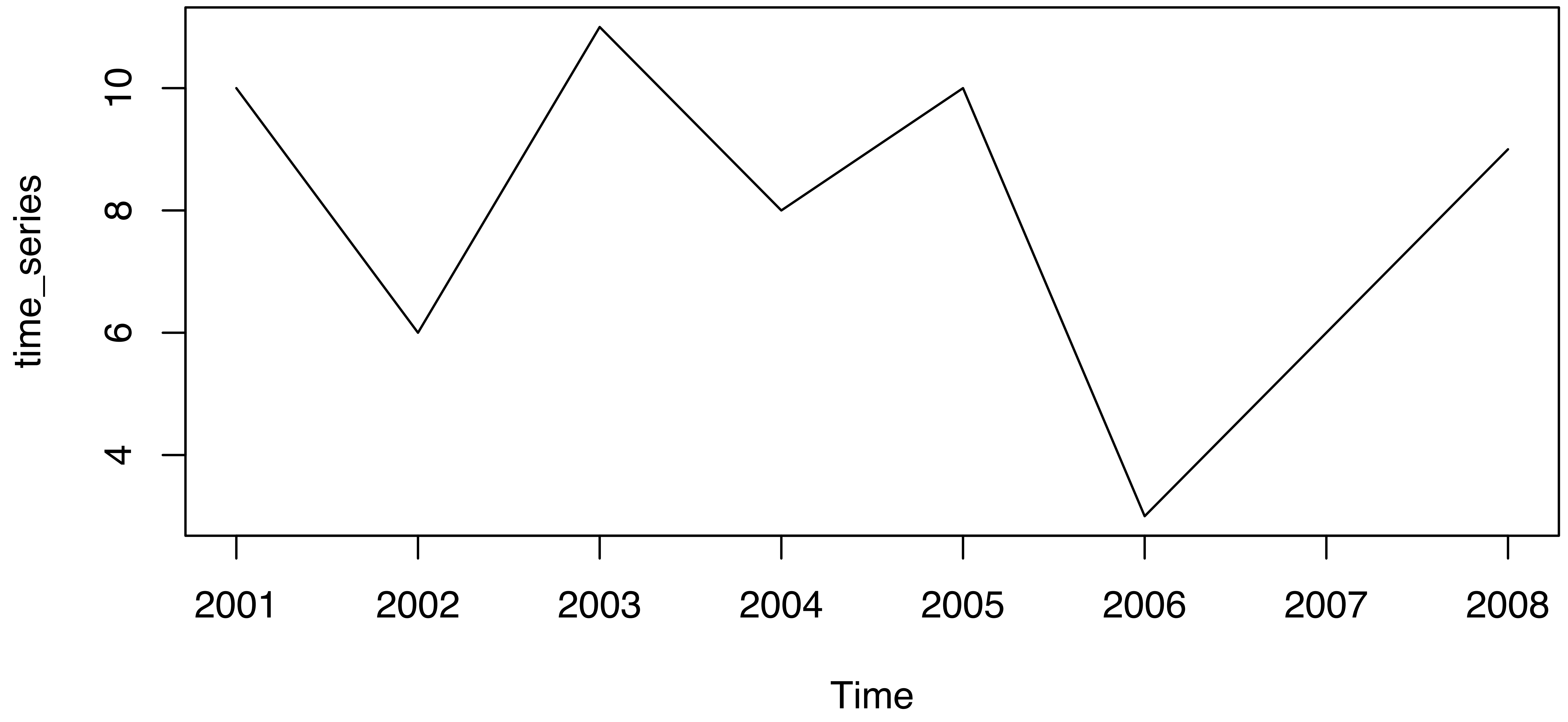
```
> time_series <-  ts(data_vector)
> plot(time_series)
```



Building `ts()` Objects - II

- Specify the start date and observation frequency:

```
> time_series <- ts(data_vector, start = 2001, frequency = 1)
> plot(time_series)
```



Using `is.ts()`

- The `is.ts()` function checks whether an object is of the `ts()` class:

```
> is.ts(data_vector)
[1] FALSE

> is.ts(time_series)
[1] TRUE
```

Why `ts()` Objects?

Why create and use time series objects of the `ts()` class?

- Improved plotting
- Access to time index information
- Model estimation and forecasting (later chapters)

Let’s practice!