Statistics for Decision Making: Broad Introduction

A Naive Approach for Forecasting Time Series

Le Wang

2018-08-27

Forecasting: What are we trying to do?

We first have to accept the fact that what we are trying to forecast or predict is a random variable subject to random disturbances.

There is a part that is not predictable at all, no matter how hard we try.

$$\text{Series} = \textbf{Signal} + \textbf{Noise}$$

1. **Signal** A collection of systematic, predictable components}
2. **Noise** Unpredictable, erratic, abrupt changes/random components

Forecasting: What are we trying to do?

How we treat the signal part leads to two different approaches in practice. The first approach considers the signal as the product of a *gradually* evolving trend and *smoothly* varying cycle. Whatever is erratic or unsystematic is treated as noise (or residual). This approach is called **smoothing**.

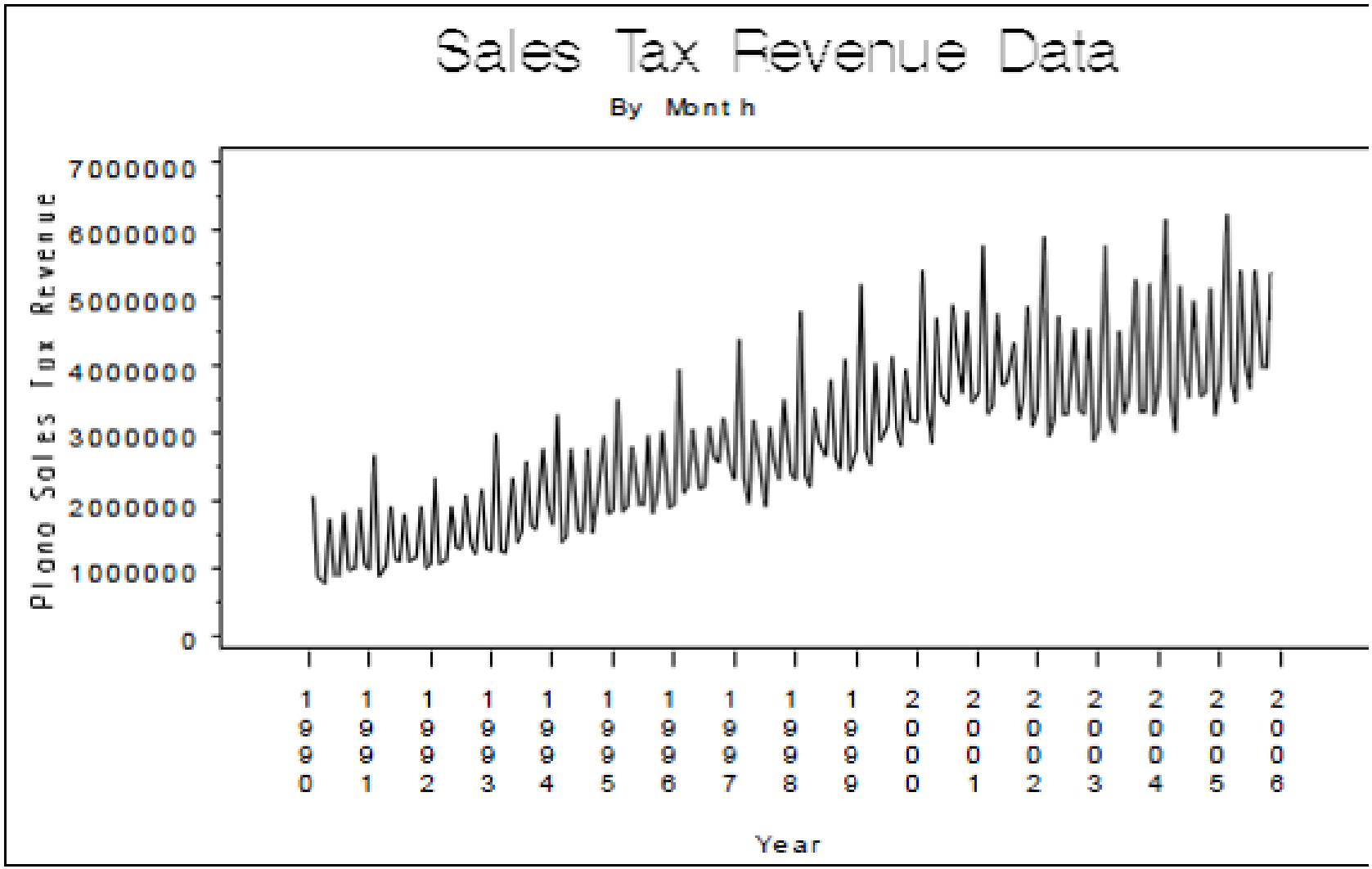
$$\textbf{Smoothing}: \quad \text{Series} = \text{Fit} + \text{Residual}$$

The second approach formally models the relation between the outcome of interest and its predictors such as past outcomes and other covariates.

$$\textbf{Modeling}: \quad \text{Series} = \text{Model} + \text{Error}$$

Forecasting: A Naive Way to Decompose a Variable

What will next period's sales tax revenue (quarterly data) be? **Use your common sense!**



Forecasting: A Naive Way to Decompose a Variable

1. Trend
2. Seasonality
3. Something else

Forecasting: A Way to Decompose a (time series) Variable

$$Y = T + C + S + I$$

or

$$Y = T \times C \times S \times I$$

Equivalent:

$$\log(Y) = \log(T) + \log(C) + \log(S) + \log(I)$$

1. Trend
2. Seasonality
3. Something else
   - 3.1 Cycles
   - 3.2 Irregular component

Forecasting: A Naive Way to Decompose a Variable

Let's define each component (Chapter 6.1 in Forecasting: Principles and Practice)

Trend A **long-term** increase or decrease in the data, as opposed to *local*, *current*, *recent* trend. (It does not have to be linear)

Question: Is this a trend? A recent trend in stock prices when prices trend up in a "bull" market or down in a "bear" market.

Forecasting: A Naive Way to Decompose a Variable

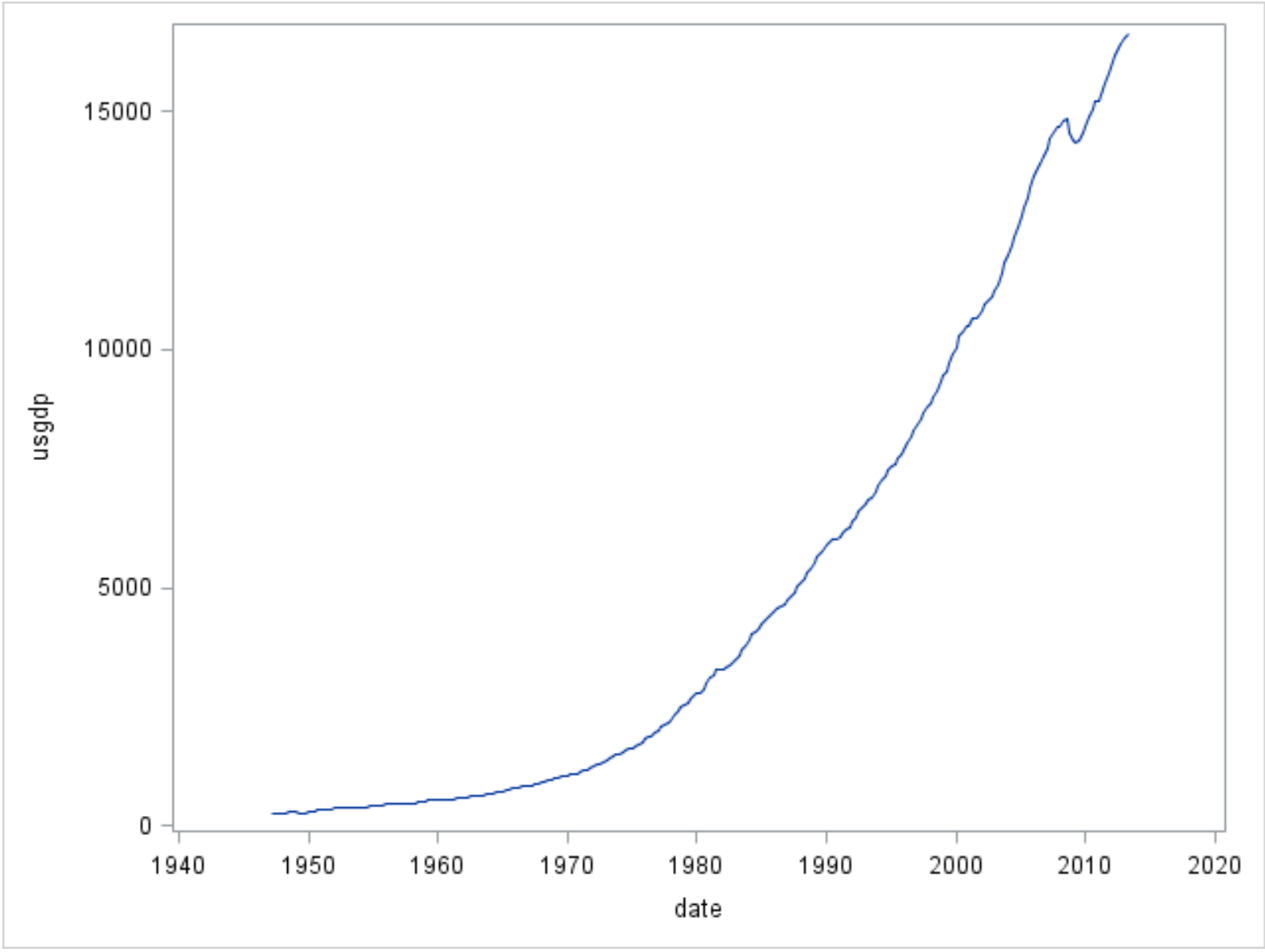


Figure 1:

Forecasting: A Naive Way to Decompose a Variable

Cycle The data exhibit rises and falls that are *not of a fixed period*. (Roughly speaking)

The term is a bit "misleading": Does not resemble the regular periodic ossillation of a function such as sine and cosine.

Business cycles

Cycles

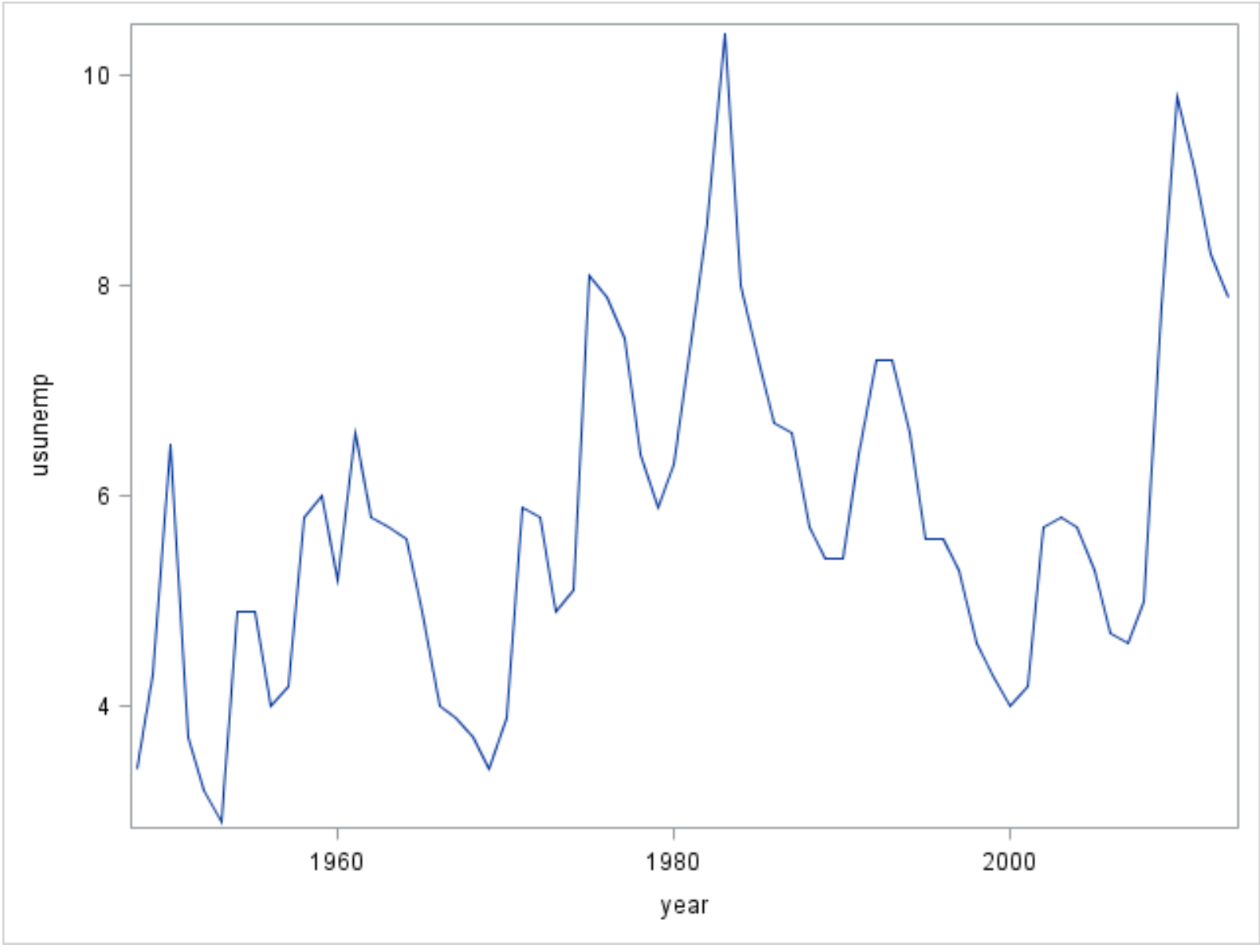


Figure 2:

Statistical Cycles vs. Deterministic Cycles

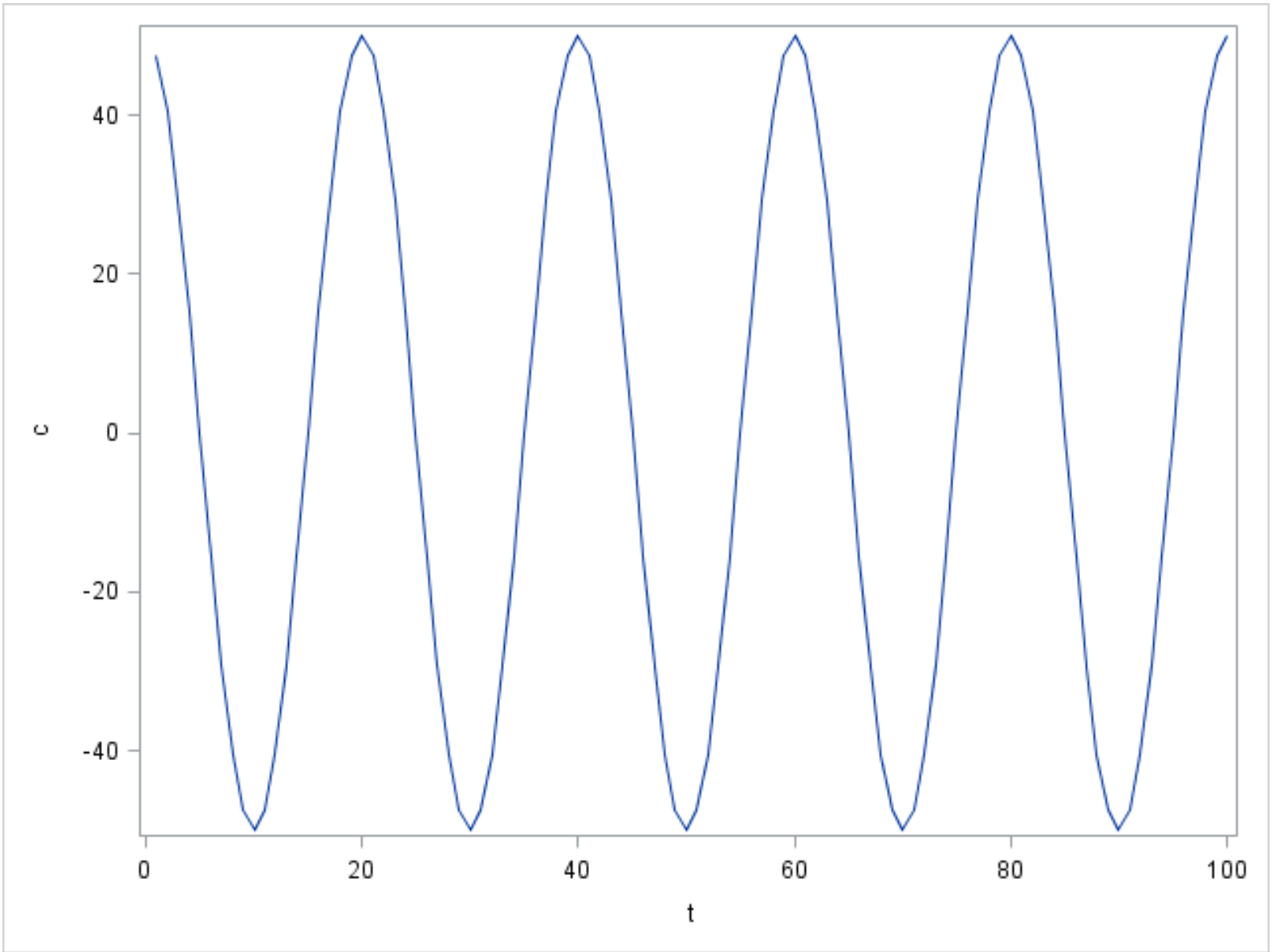


Figure 3:

Seasonality

Seasonality A time series is affected by seasonal factors such as the time of year, or the day of week.

Seasonality

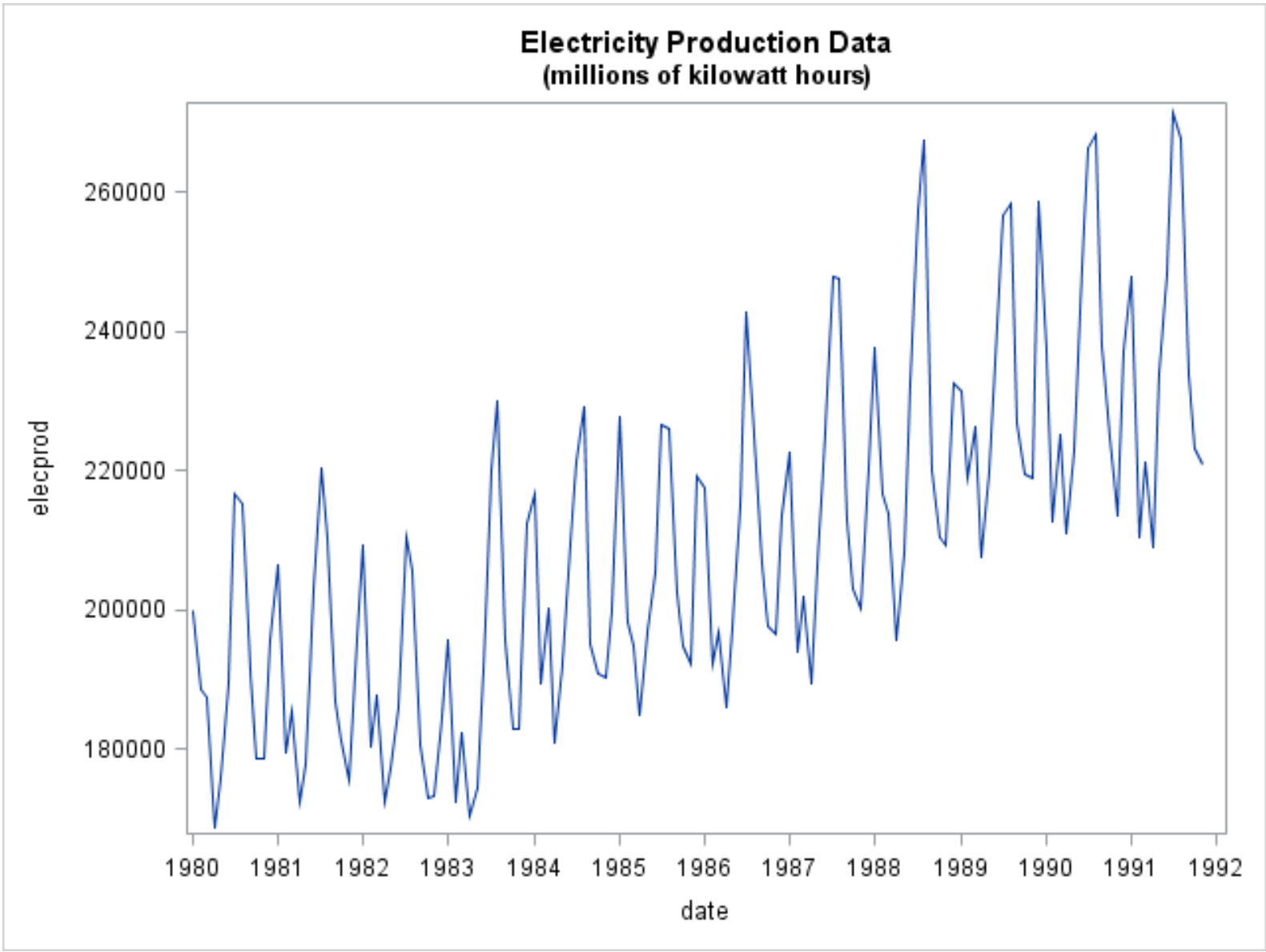


Figure 4:

Working with Time Series in R

Many of you are more familiar with macro type of data such as inflation and interest rates for a particular country or region. This type of data are time series.

Such data requires some specical handling in order to fully expolit the time series structure. It is thus important to learn how to work with time series data in R.

Working with Time Series in R

In what follows, we will learn two basic things:

1. Time Series Plots
    - 1.1 `quantmod()`
    - 1.2 `plot()`
    - 1.3 `ggplot2()`
2. Classical Decomposition of Time Series (Not particularly recommended, but sufficient for illustration).

Working with Time Series in R

Let's obtain the data first using `quantmod()`.

Step 1. We need to let R know where this function (or tool) is located.

Step 2. We can grab the data using `getSymbols()`

Working with Time Series in R

```
library(quantmod)
getSymbols("UNRATE", src = "FRED")
## [1] "UNRATE"
```

1. The first argument is **variable name**, which can also be found at the end of the website link,
2. The second argument `src = 'FRED'` instructs R to obtain the data from St. Louis Fed.

Working with Time Series in R

```
head(UNRATE)
```

```
##              UNRATE
## 1948-01-01      3.4
## 1948-02-01      3.8
## 1948-03-01      4.0
## 1948-04-01      3.9
## 1948-05-01      3.5
## 1948-06-01      3.6
```

Note: only one column of data, which is called `UNRATE`. The first column is not actual data, but row names.

Working with Time Series in R

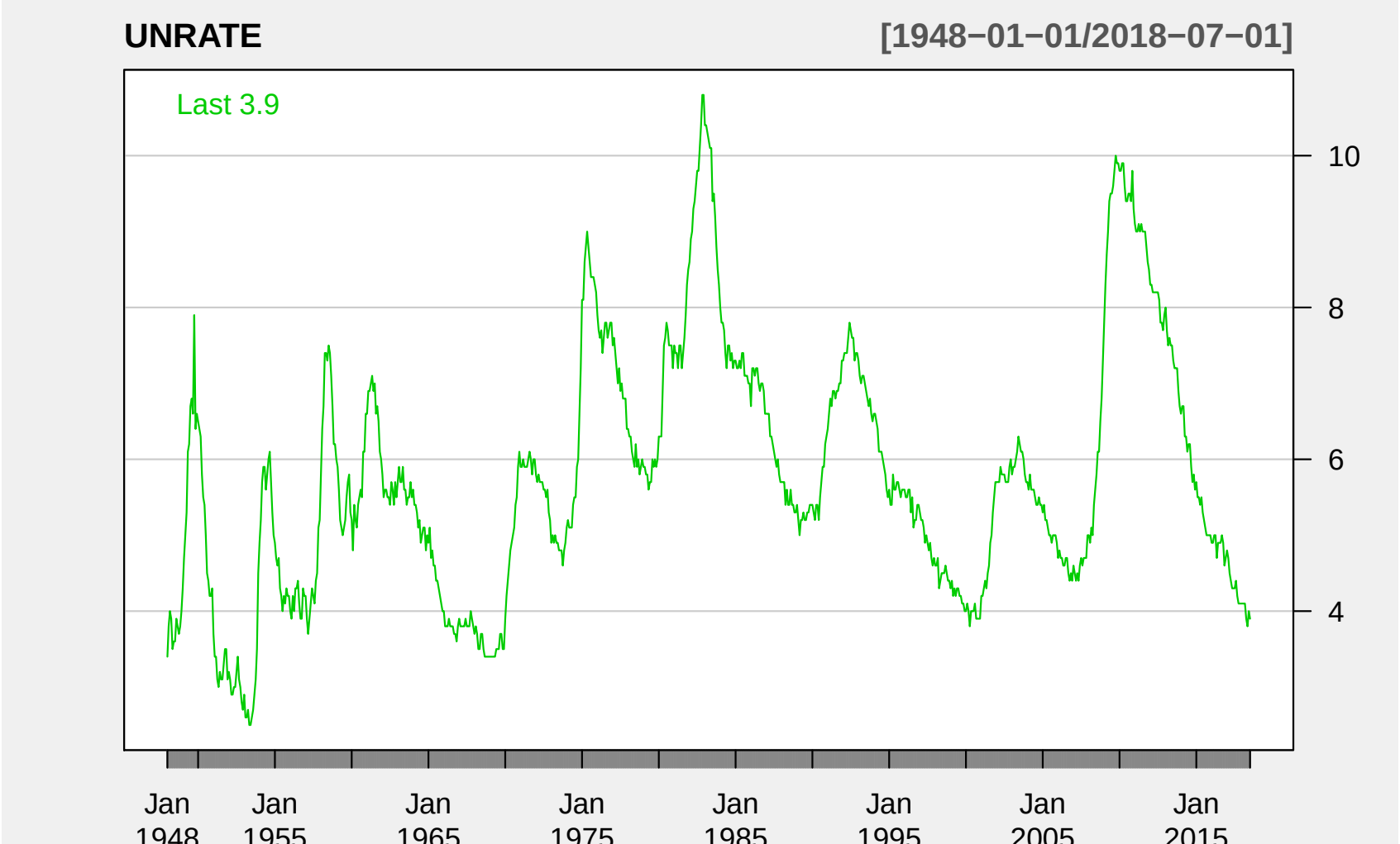
```
tail(UNRATE)
```

```
##            UNRATE
## 2018-02-01    4.1
## 2018-03-01    4.1
## 2018-04-01    3.9
## 2018-05-01    3.8
## 2018-06-01    4.0
## 2018-07-01    3.9
```

Working with Time Series in R (Approach 1)

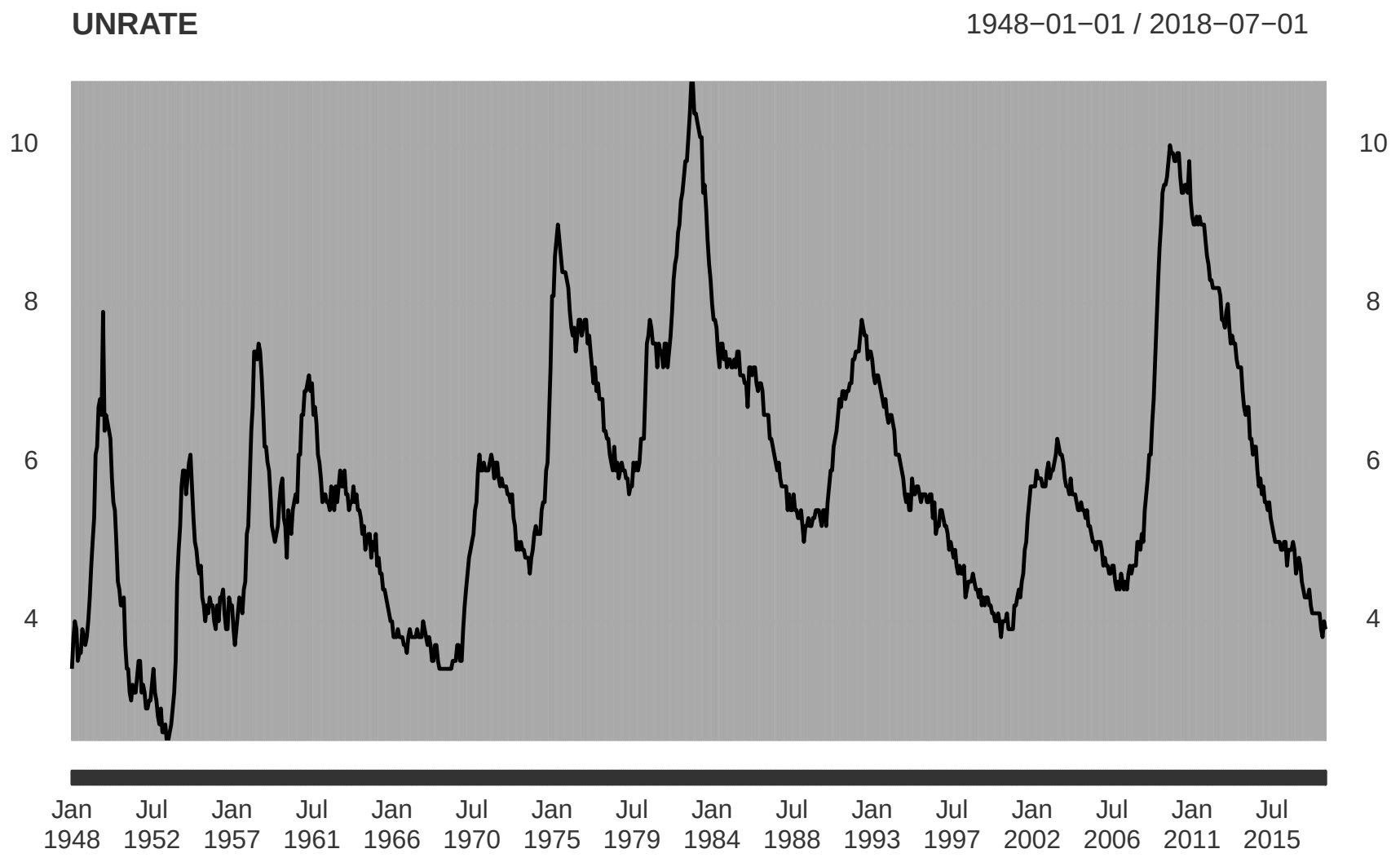
First approach to plotting the data is to use `chartSeries()`, a built-in command that comes with the package `quantmod`.

```
chartSeries(UNRATE, theme="white")
```



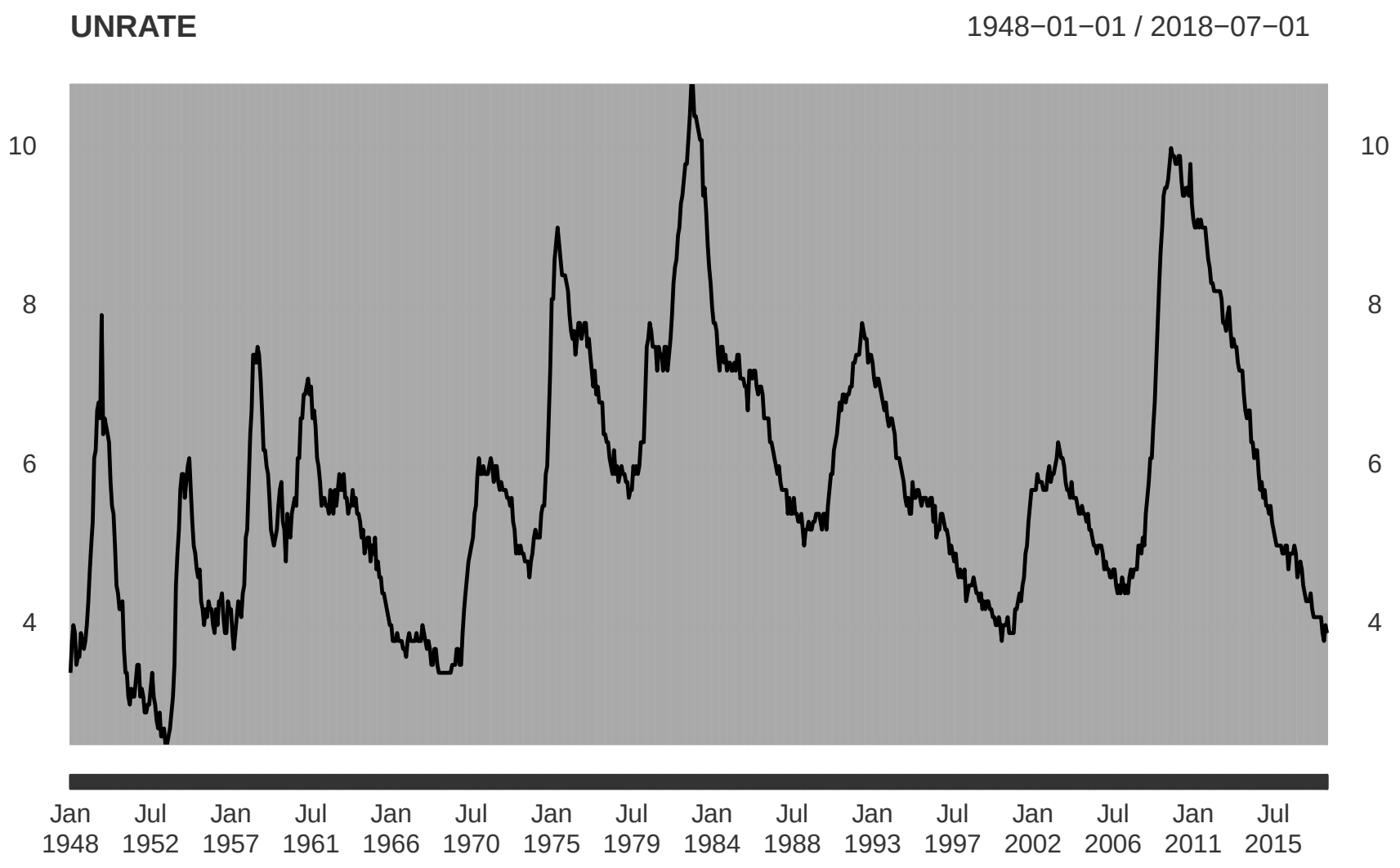
Working with Time Series in R (Approach 2)

```
plot(UNRATE)
```



Working with Time Series in R (Approach 2)

```
plot(UNRATE)
abline(h = 10, col = "salmon")
```



Working with Time Series in R (Approach 3)

The **third** approach is to use **ggplot2**, one of the most elegant and most versatile systems for making graphs.

1. Implements the *grammar of graphics*, a coherent system for describing and building graphs
2. More faster by learning one system and applying it in many places.

Working with Time Series in R (Approach 3)

Here we need a dataset (in the data frame) with two variables, one on the x axis and the other y-axis.

Working with Time Series in R (Approach 3)

```r
# 1. Convert it to a data frame
data.unrate <- data.frame(UNRATE)
data.unrate$date <- as.Date(rownames(data.unrate))
  colnames(data.unrate)[1] = 'unrate'
  head(data.unrate)
```

```
##              unrate       date
## 1948-01-01      3.4 1948-01-01
## 1948-02-01      3.8 1948-02-01
## 1948-03-01      4.0 1948-03-01
## 1948-04-01      3.9 1948-04-01
## 1948-05-01      3.5 1948-05-01
## 1948-06-01      3.6 1948-06-01
```

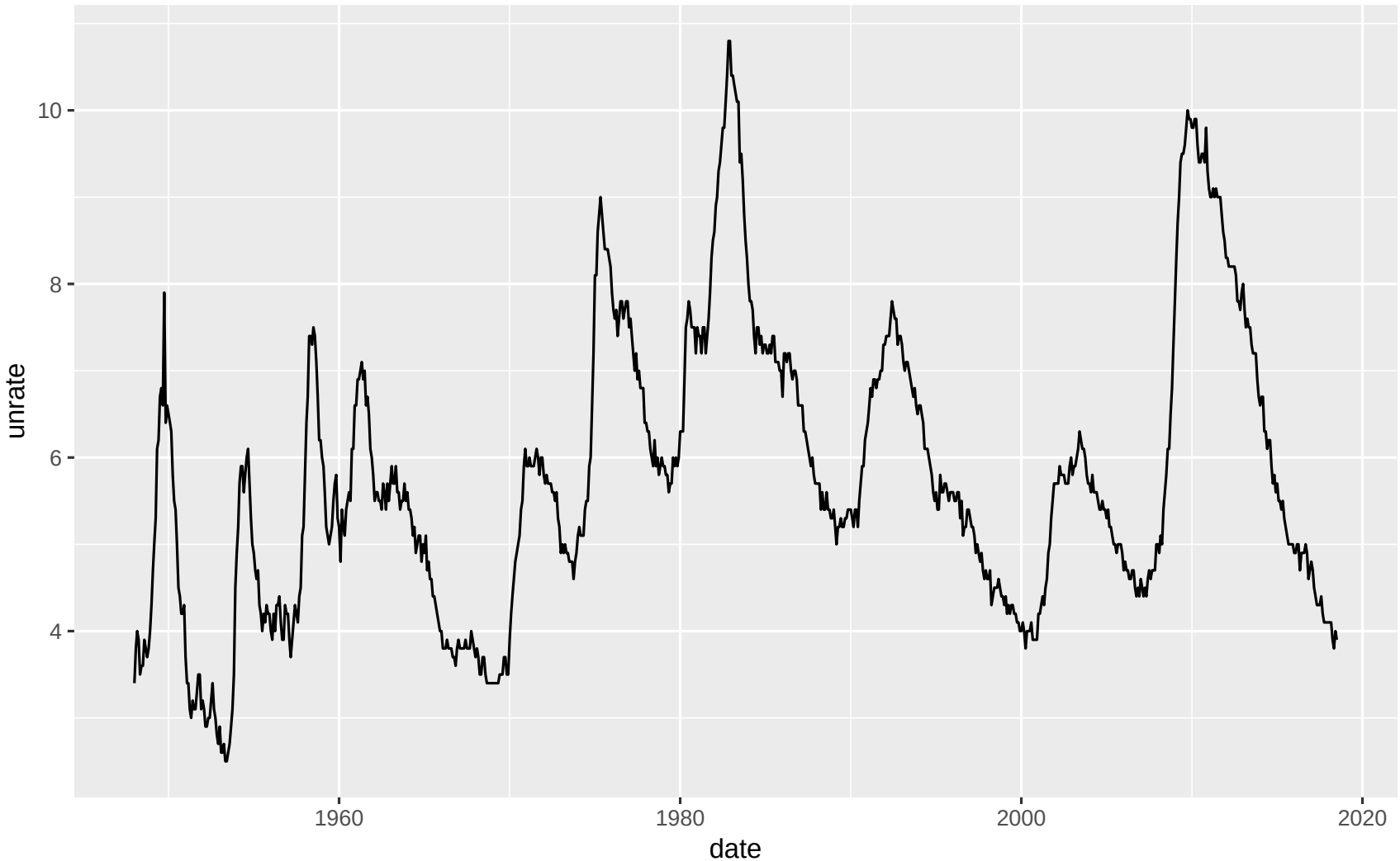
Working with Time Series in R (Approach 3)

A Graphing Template

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

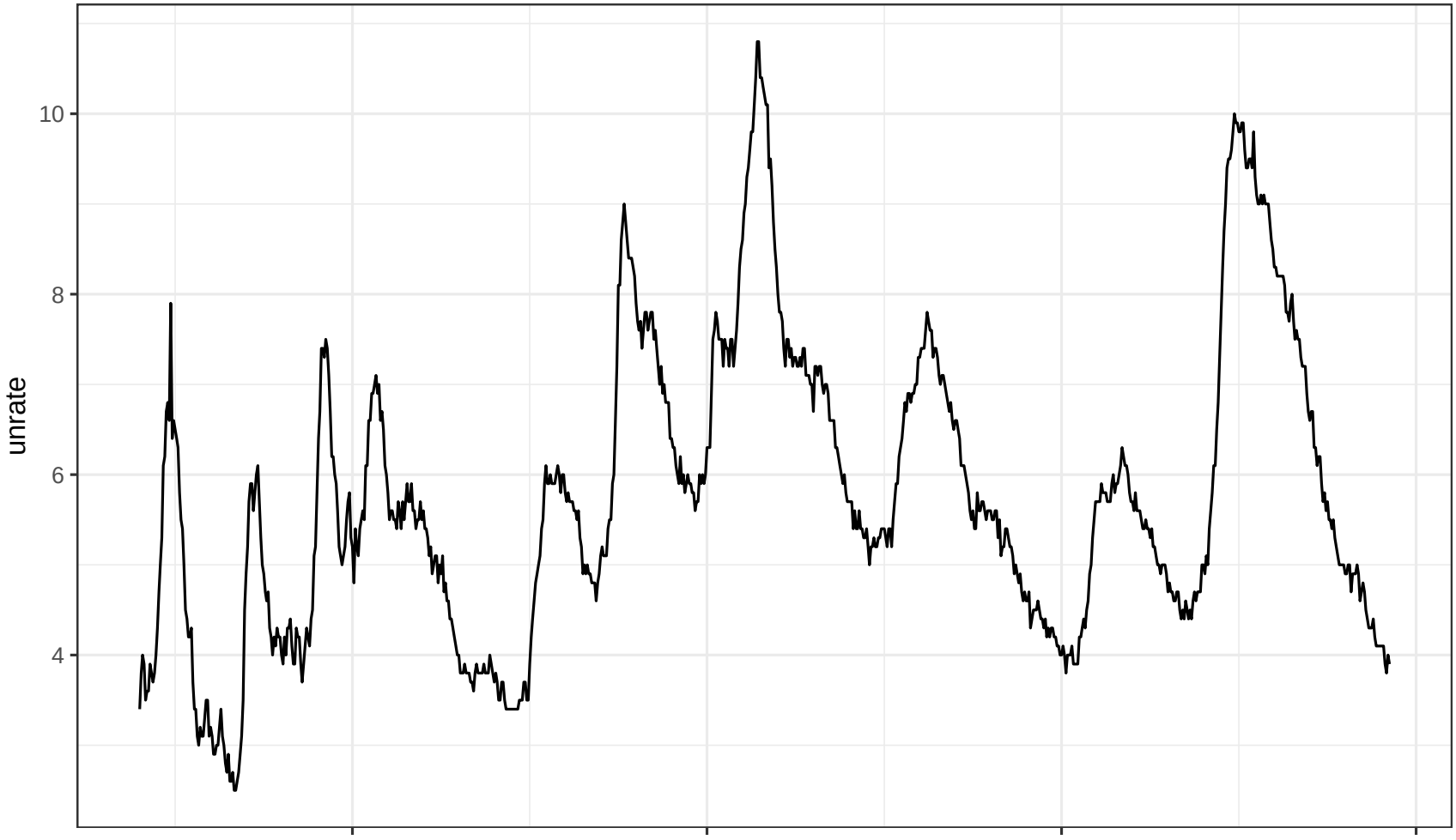
Working with Time Series in R (Approach 3)

```
library(ggplot2)
ggplot(data=data.unrate) +
  geom_line(mapping = aes(x=date,y=unrate))
```



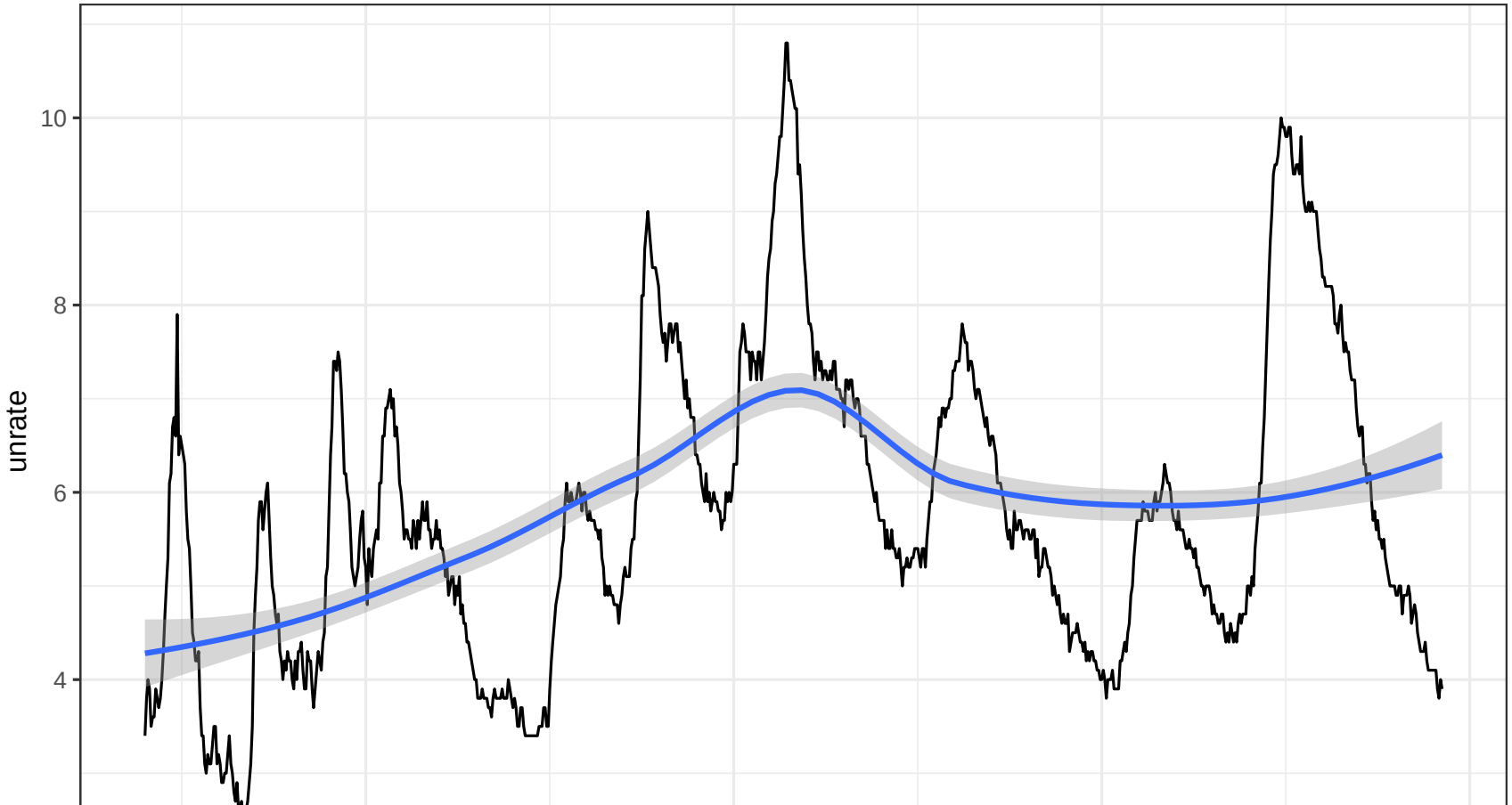
Working with Time Series in R (Approach 3)

```
library(ggplot2)
ggplot(data=data.unrate) +
  geom_line(mapping = aes(x=date,y=unrate)) +
  theme_bw()
```



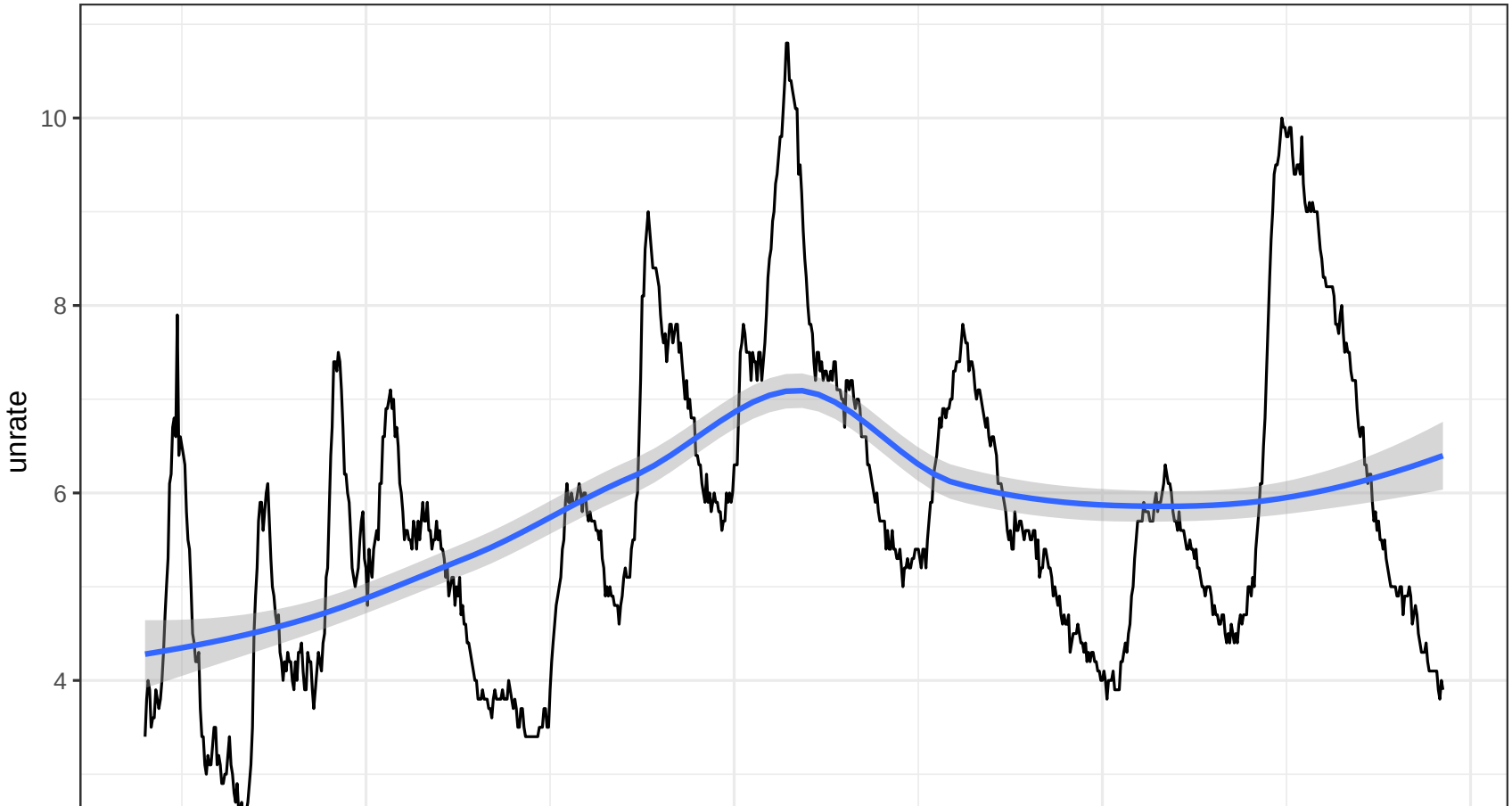
Working with Time Series in R (Approach 3)

```
library(ggplot2)
ggplot(data=data.unrate) +
  geom_line(mapping = aes(x=date,y=unrate)) +
  geom_smooth(mapping = aes(x=date,y=unrate)) +
  theme_bw()
```



Working with Time Series in R (Efficient Approach 3)

```
library(ggplot2)
ggplot(data=data.unrate, mapping = aes(x=date,y=unrate))
  geom_line() +
  geom_smooth() +
  theme_bw()
```



Working with Time Series in R (Efficient Approach 3)

```
ggplot(data=data.unrate, mapping = aes(x=date,y=unrate))
  geom_line() +
  geom_smooth() +
  theme_bw()
```

Painting on a Canvas

1. Creates an empty canvas
2. Adds a layer of connected line (geometric objects **GEOM** functions)
   - 2.1 The `mapping` argument is always paired with `aes()`.
   - 2.2 An *aesthetic* is a visual property of the objects in your plot. You can change, e.g., the shape or the size of your line.
3. Adds another layer of smoothed line
4. Replace the background with black and white canvas.