Time Series Analysis in Health Research

Lecture 1: Introduction to Forecasting

Erfan Hoque

Dept. of Community Health & Epidemiology, University of Saskatchewan

SEA 24th Annual Symposium Workshop, September 26, 2025

Land Acknowledgement

I would like to acknowledge that the Saskatoon campus of the University of Saskatchewan is on Treaty 6 Territory and the Homeland of the Métis. We pay our respect to the First Nation and Métis ancestors of this place and reaffirm our relationship with one another.

Acknowledgement

These documents were prepared with the help of my research assistant, *Syed Jafar Rizvi*, MSc Student, Dept. of Community Health and Epidemiology, University of Saskatchewan.

NSERC CRSNG

UNIVERSITY OF SASKATCHEWAN
College of Medicine
MEDICINE.USASK.CA

Contact details

Dr. Erfanul Hoque (Call me Erfan)

Assistant Professor of Biostatistics & Data Science

Email: erfan.hoque@usask.ca

Office: Room E3222, Health Sciences Building

My Research Areas

Complex and Correlated data Analysis

Longitudinal Data Analysis

Statistical/Machine Learning

Time Series Analysis

Survival Analysis

Biostatistics

Meta-Analysis

Missing Data / Measurement Error

Details: `https://medicine.usask.ca/profiles/che/erfanul-hoque.php`

GitHub link for Materials:

You can download the materials used in this workshop form my GitHub:

Details: `https://github.com/ErfanHoque49/Time-Series-Analysis-in-Health-Research`

Objectives

1. To obtain an understanding of common time series methods used in epidemiological and health data forecasting.
2. To gain insights into the problems of implementing and operating large scale forecasting systems for use in health research.
3. To develop the computer skills required to implement the topics using statistical software R.

R

Available for download from CRAN:

`https://cran.csiro.au/`

Available for download from RStudio:

`https://www.rstudio.com/products/rstudio/download/`

How familiar are you with R and Time Series?

https://www.menti.com/al9vp34p3wk6

Code: 8336 1628

Main packages

Main packages

```r
# Data manipulation and plotting functions
library(tidyverse)
# Time series manipulation
library(tsibble)
# Tidy time series data
library(tsibbledata)
# Time series graphics and statistics
library(feasts)
# Forecasting functions
library(fable)
```

Main packages

```r
# Data manipulation and plotting functions
library(tidyverse)
# Time series manipulation
library(tsibble)
# Tidy time series data
library(tsibbledata)
# Time series graphics and statistics
library(feasts)
# Forecasting functions
library(fable)

# All of the above
library(fpp3)
```

Install required packages

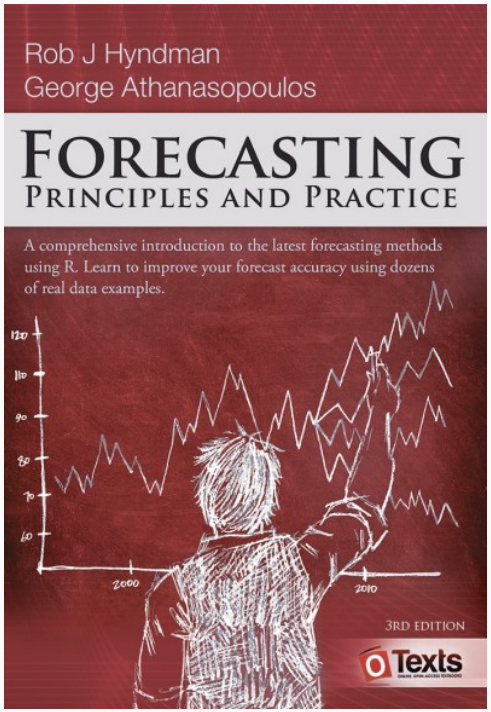
```
install.packages(c(
  "tidyverse",
  "fpp3"
))
```

Schedule for this workshop

	Topic
Lecture 1	Introduction to forecasting, graphics
Lecture 2	Time Series toolbox and Models
Lecture 3	Time Series Regression Models
Lecture 4	Hands-on session in R
Lecture 5	Advanced Forecasting Methods

Resources

Organization and presentation of material in these lectures will largely pull from



- Free and online (`https://otexts.com/fpp3/`)
- Data sets in associated R packages, R code for examples

Forecast is difficult

Forecast is difficult

A Timeline of

Very Bad Future Predictions

1800

"Rail travel at high speed is not possible, because passengers, unable to breathe, would die of asphyxia."

Dr. Dionysys Larder, Professor of Natural Philosophy & Astronomy, University College London

1859

"Drill for oil? You mean drill into the ground to try and find oil? You're crazy!"

Associates of Edwin L. Drake refusing his suggestion to drill for oil in 1859 (Later that year, Drake succeeded in drilling the first oil well.)

1876

"This telephone has too many shortcomings to be seriously considered as a means of communication."

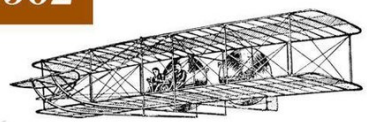
Western Union internal memo

1880

"Everyone acquainted with the subject will recognize it as a conspicuous failure."

Henry Morton, president of the Stevens Institute of Technology, on Edison's light bulb

1902



"Flight by machines heavier than air is unpractical and insignificant, if not utterly impossible."

Simon Newcomb, Canadian-American astronomer and mathematician, 18 months before the Wright Brothers' flight at Kittyhawk

1903

"The horse is here to stay, but the automobile is only a novelty, a fad."

The president of the Michigan Savings Bank, advising Henry Ford's lawyer not to invest in the Ford Motor Company

1916

"The idea that cavalry will be replaced by these iron coaches is absurd. It is little short of treasonous."

Comment of Aide-de-camp to Field Marshal Haig, at tank demonstration

1916

"The cinema is little more than a fad. It's canned drama. What audiences really want to see is flesh and blood on the stage."

Charlie Chaplin, actor, producer, director, and studio founder

1921

"The wireless music box has no imaginable commercial value. Who would pay for a message sent to no one in particular?"

Associates of commercial radio and television pioneer, David Sarnoff, responding to his call for investment in the radio

1946

"Television won't last because people will soon get tired of staring at a plywood box every night."

Darryl Zanuck, movie producer, 20th Century Fox

1977

"There is no reason for any individual to have a computer in his home."

Ken Olson, president, chairman and founder of Digital Equipment Corporation

1995

"The truth is no online database will replace your daily newspaper…"

Clifford Stoll, Newsweek article entitled ***The Internet? Bah!***

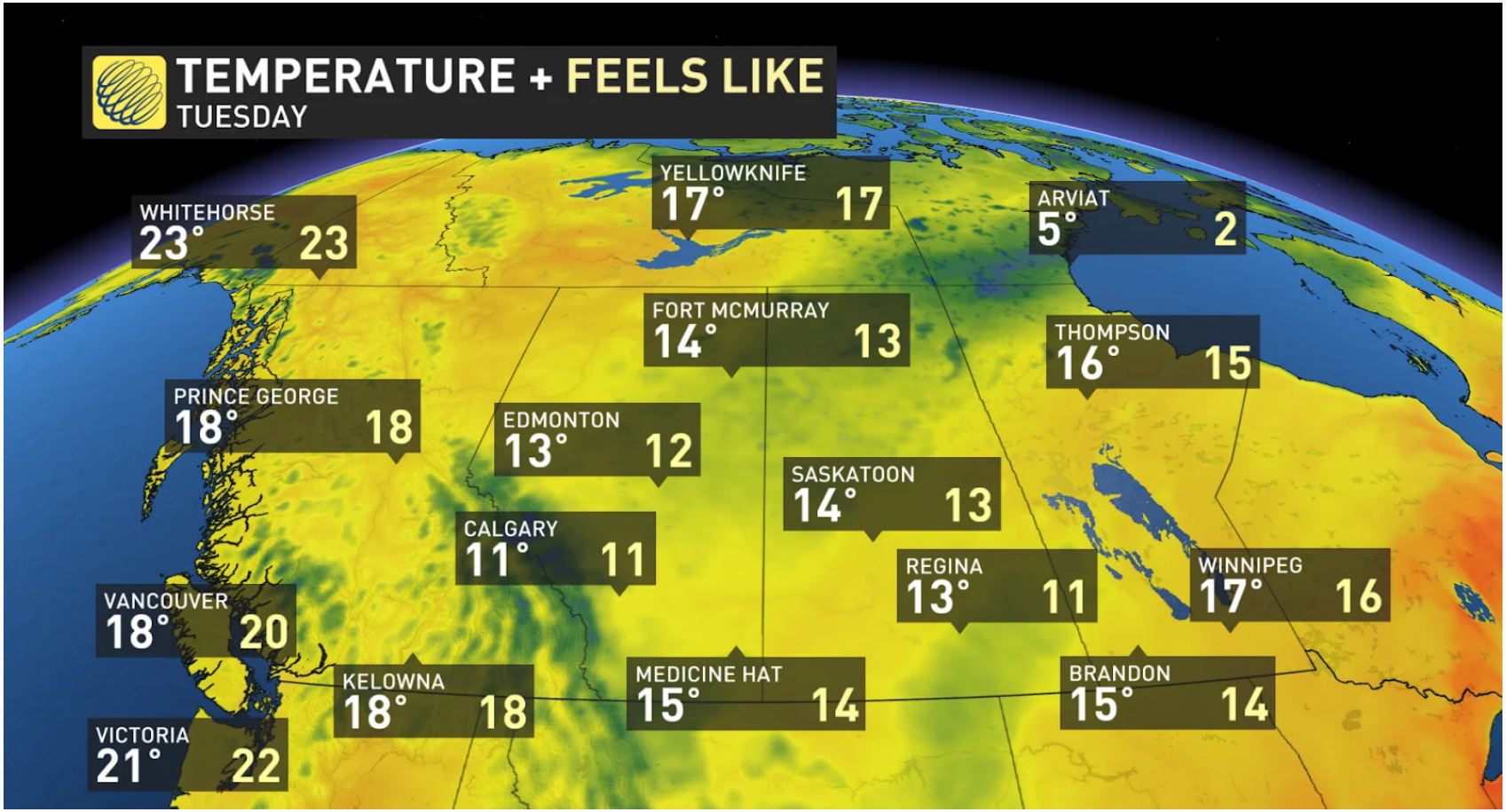
What can we forecast

What can we forecast– Stock-market

What can we forecast– Drug sales

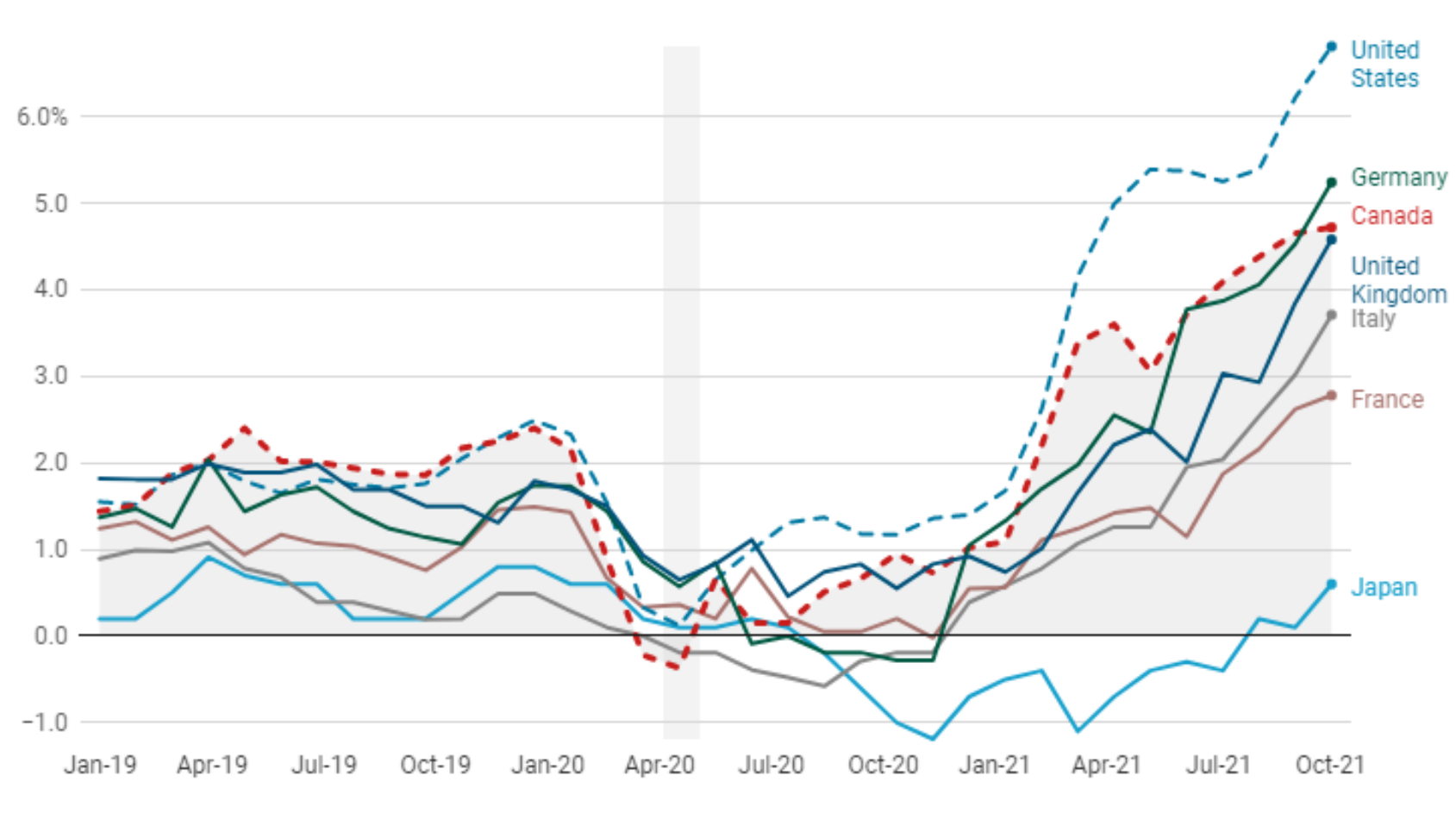
What can we forecast– Electricity demand

What can we forecast– Temperature

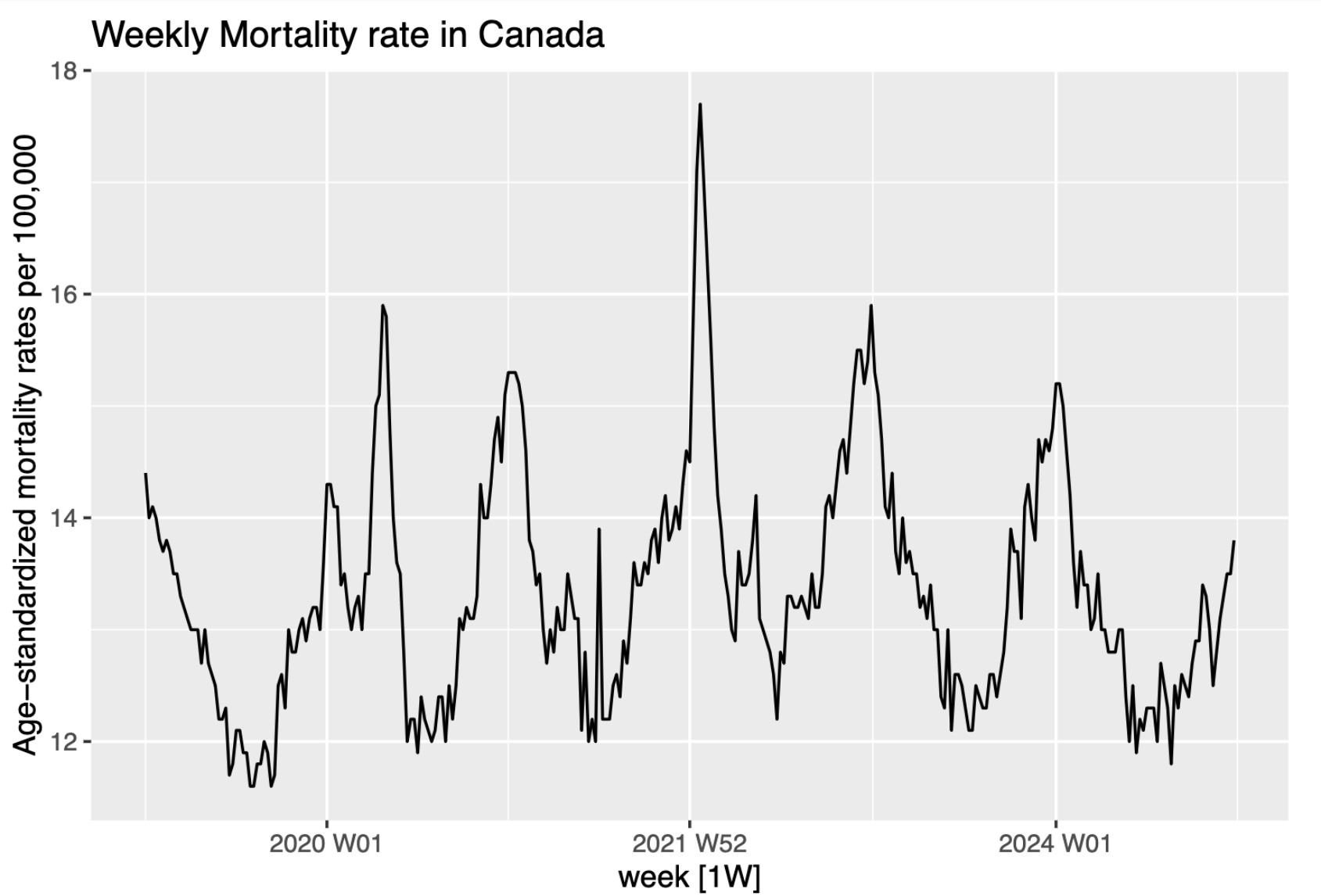


What can we forecast– Traffic flow prediction

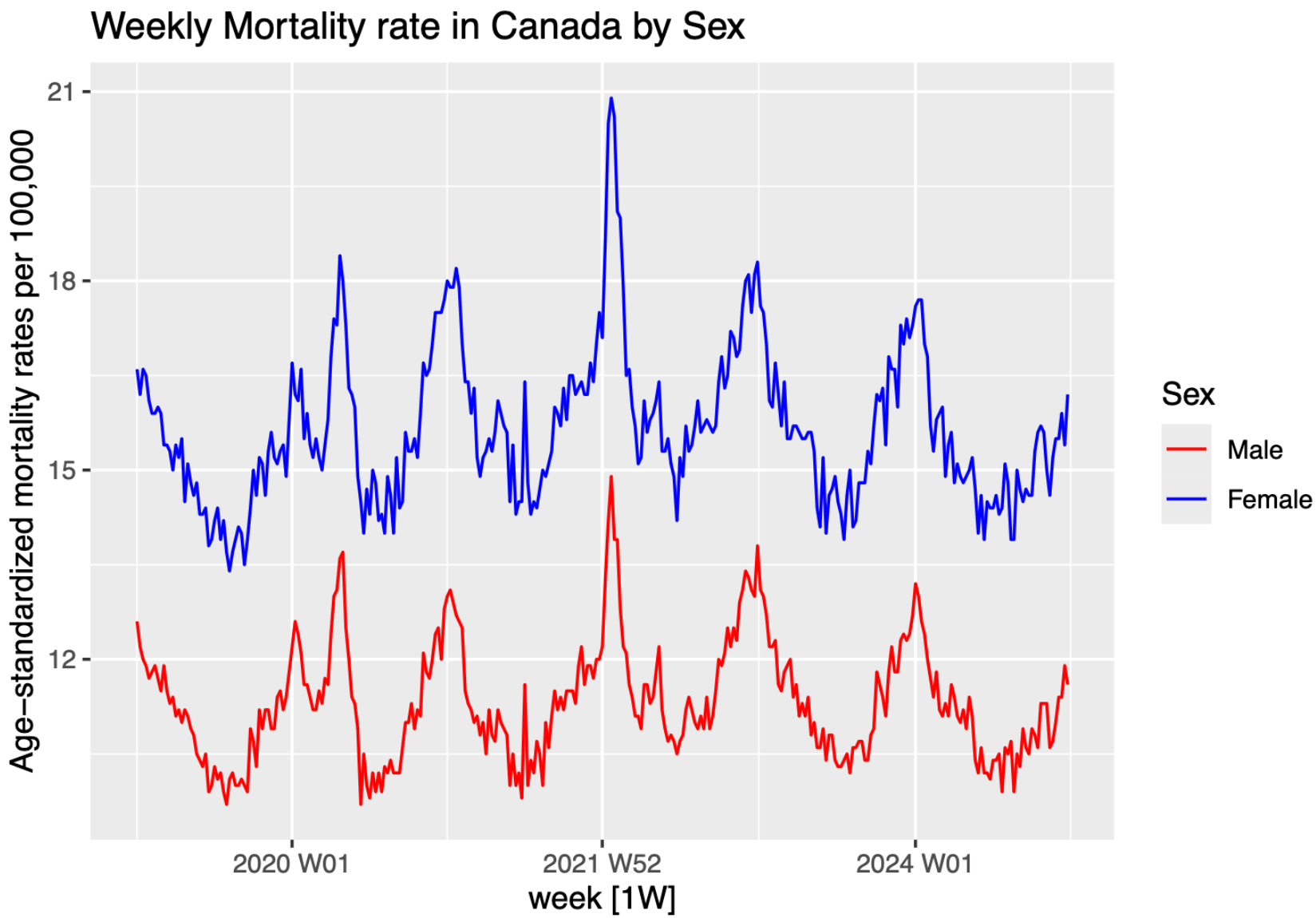
What can we forecast– Inflation rate



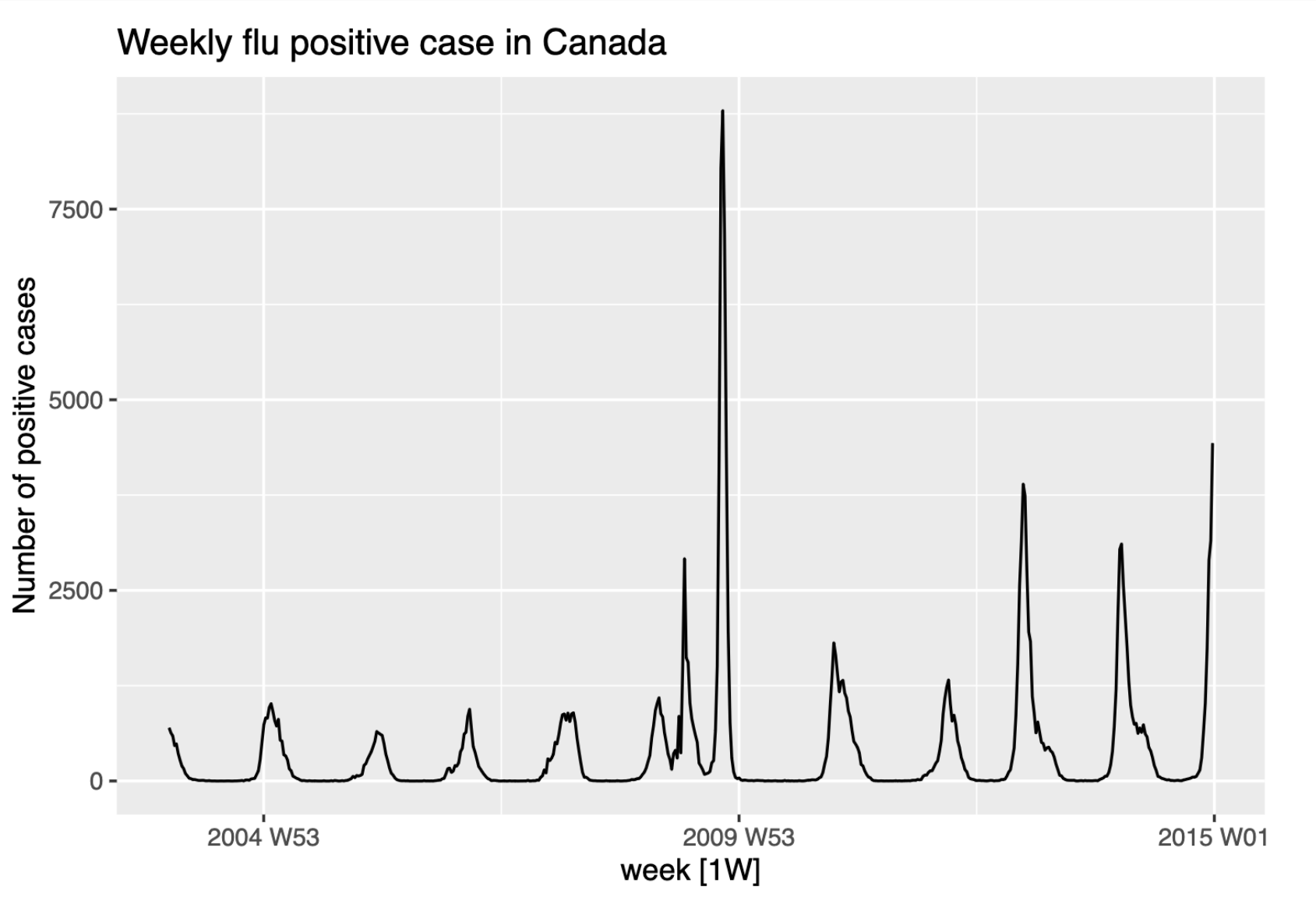
What can we forecast– Morality rate



What can we forecast– Morality rate by Sex



What can we forecast– Flu rate



Factors affecting forecastability

- what makes something easy/difficult to forecast?

Something is easier to forecast if:

1. we have a good understanding of the factors that contribute to it
2. there is lots of data available
3. the future is somewhat similar to the past
4. the forecasts cannot affect the thing we are trying to forecast.

Predictor variables and time series forecasting

- Predictor variables are often useful in time series forecasting. For example, suppose we wish to forecast the hourly electricity demand (ED) of Saskatoon during the summer period. A model with predictor variables can be written as:

$ED_{sk} = f(\text{temperature, population, time of day, day of week, error})$.

- Because the electricity demand data form a time series, we could also use a time series model for forecasting as:

$ED_{t+1} = f(ED_t, ED_{t-1}, ED_{t-2}, ED_{t-3}, \ldots, error)$,

- where t is the present hour, $t+1$ is the next hour, $t-1$ is the previous hour, $t-2$ is two hours ago, and so on.
- Another model can combines the features of the above two models:

$ED_{t+1} = f(ED_t, \text{current temperature, time of day, day of week, error})$.

The forecasting perspective

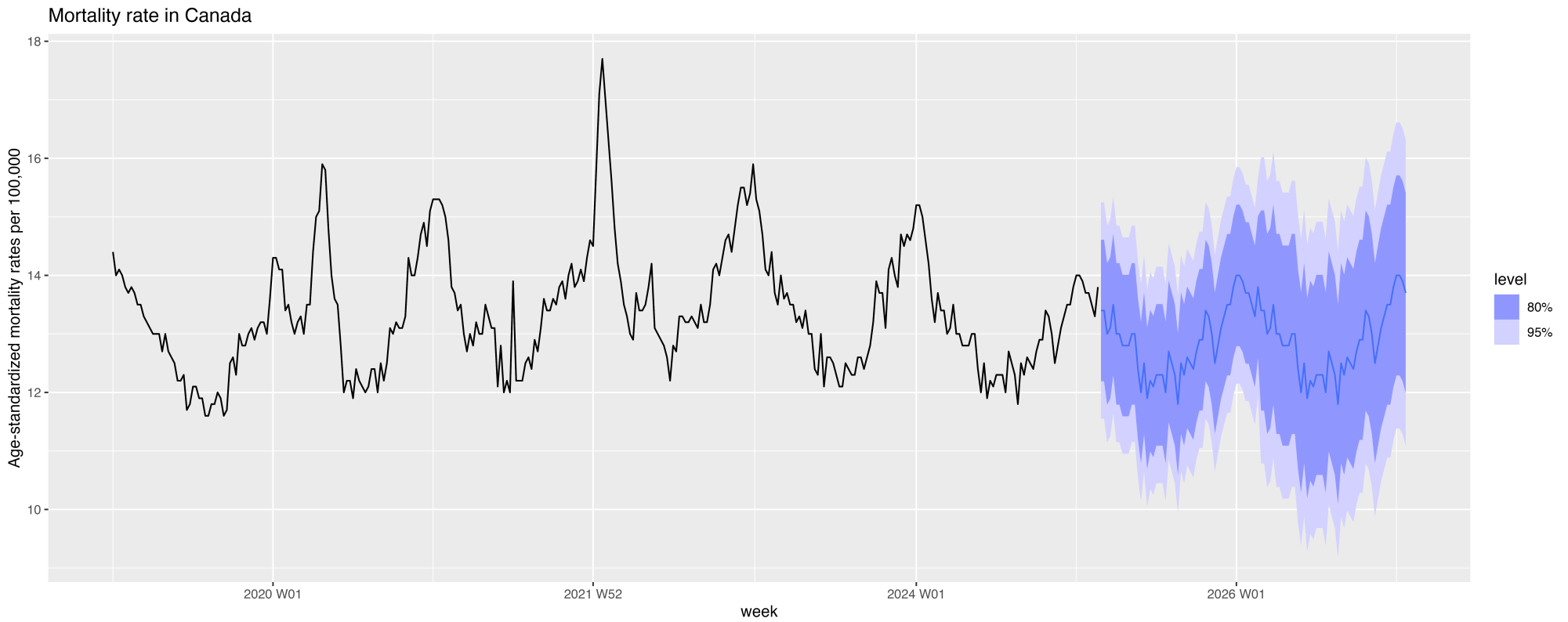
Forecast is a random variable!

- The thing we are trying to forecast is unknown (or we would not be forecasting it), and so we can think of it as a random variable.
- For example, the total sales for next month could take a range of possible values, and until we add up the actual sales at the end of the month, we don't know what the value will be.
- So until we know the sales for next month, it is a random quantity.

Uncertainty in forecast:

- As next month is relatively close, we have a good idea what the likely sales values could be.
- If we are forecasting the sales for the same month next year, the possible values it could take are much more variable.
- Forecast variation will be small as the event approaches. The further ahead we forecast, the more uncertain we are.

Forecast intervals (prediction interval)



Forecasting is estimating how the sequence of observations will continue into the future.

Time series in R

tsibble objects

```
global_economy
```

```
## # A tsibble: 15,150 x 9 [1Y]
## # Key:       Country [263]
##    Country   Code   Year    GDP Growth   CPI Imports Exports
##    <fct>     <fct> <dbl>  <dbl>  <dbl> <dbl>   <dbl>   <dbl>
##  1 Afghanis~ AFG    1960 5.38e8     NA    NA    7.02    4.13
##  2 Afghanis~ AFG    1961 5.49e8     NA    NA    8.10    4.45
##  3 Afghanis~ AFG    1962 5.47e8     NA    NA    9.35    4.88
##  4 Afghanis~ AFG    1963 7.51e8     NA    NA   16.9     9.17
##  5 Afghanis~ AFG    1964 8.00e8     NA    NA   18.1     8.89
##  6 Afghanis~ AFG    1965 1.01e9     NA    NA   21.4    11.3
##  7 Afghanis~ AFG    1966 1.40e9     NA    NA   18.6     8.57
##  8 Afghanis~ AFG    1967 1.67e9     NA    NA   14.2     6.77
##  9 Afghanis~ AFG    1968 1.37e9     NA    NA   15.2     8.90
## 10 Afghanis~ AFG    1969 1.41e9     NA    NA   15.0    10.1
## # i 15,140 more rows
## # i 1 more variable: Population <dbl>
```

tsibble objects

For observations more frequent than once per year, we need to use a time class function on the index.

```
z
```

```
## # A tibble: 5 x 2
##   Month    Observation
##   <chr>          <dbl>
## 1 2019 Jan          50
## 2 2019 Feb          23
## 3 2019 Mar          34
## 4 2019 Apr          30
## 5 2019 May          25
```

The tsibble index

For observations more frequent than once per year, we need to use a time class function on the index.

```
z %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)
```

```
## # A tsibble: 5 x 2 [1M]
##      Month Observation
##      <mth>       <dbl>
## 1 2019 Jan          50
## 2 2019 Feb          23
## 3 2019 Mar          34
## 4 2019 Apr          30
## 5 2019 May          25
```

The `tsibble` index

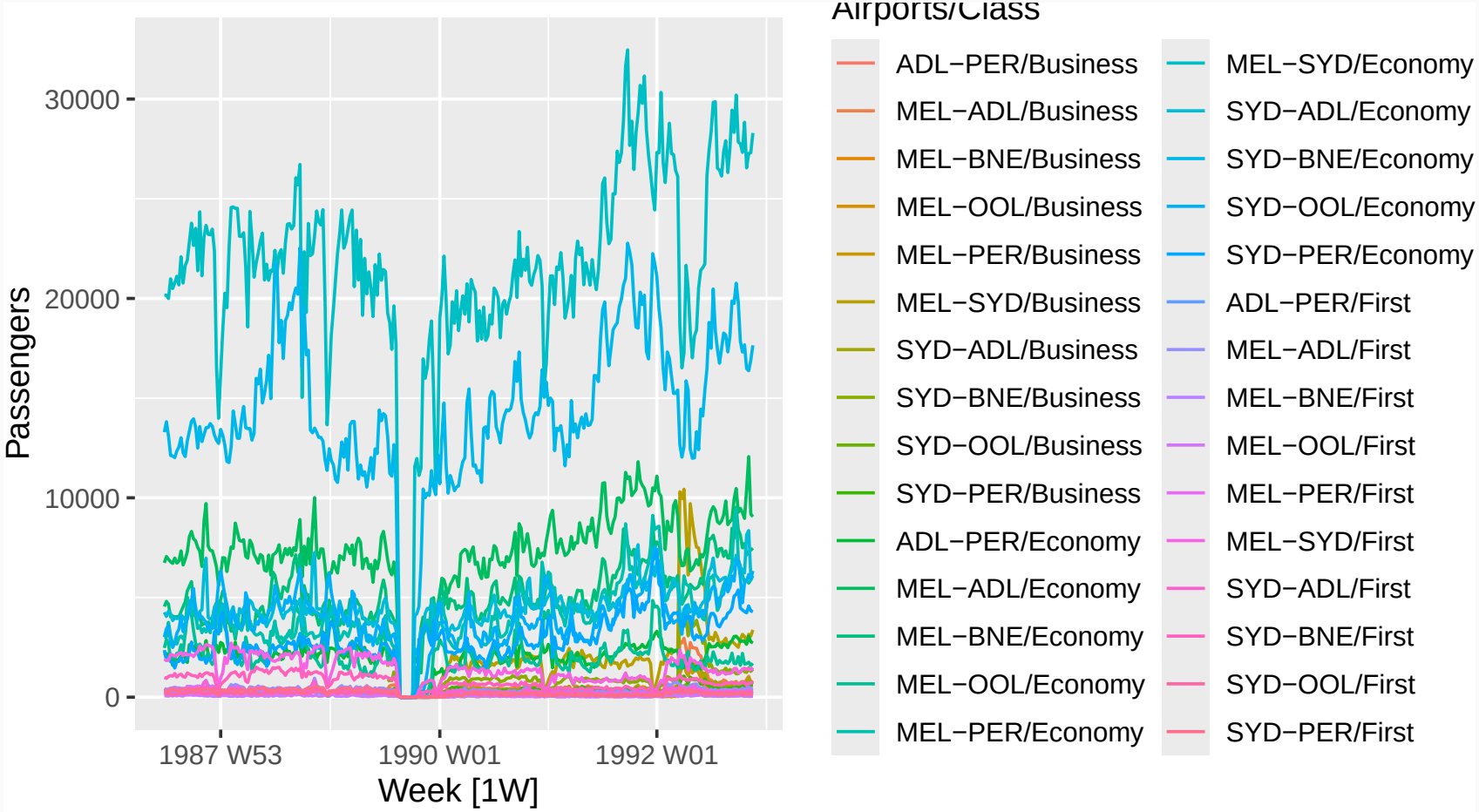
Common time index variables can be created with these functions:

Frequency	Function
Annual	`start:end`
Quarterly	`yearquarter()`
Monthly	`yearmonth()`
Weekly	`yearweek()`
Daily	`as_date()`, `ymd()`
Sub-daily	`as_datetime()`

Time plots

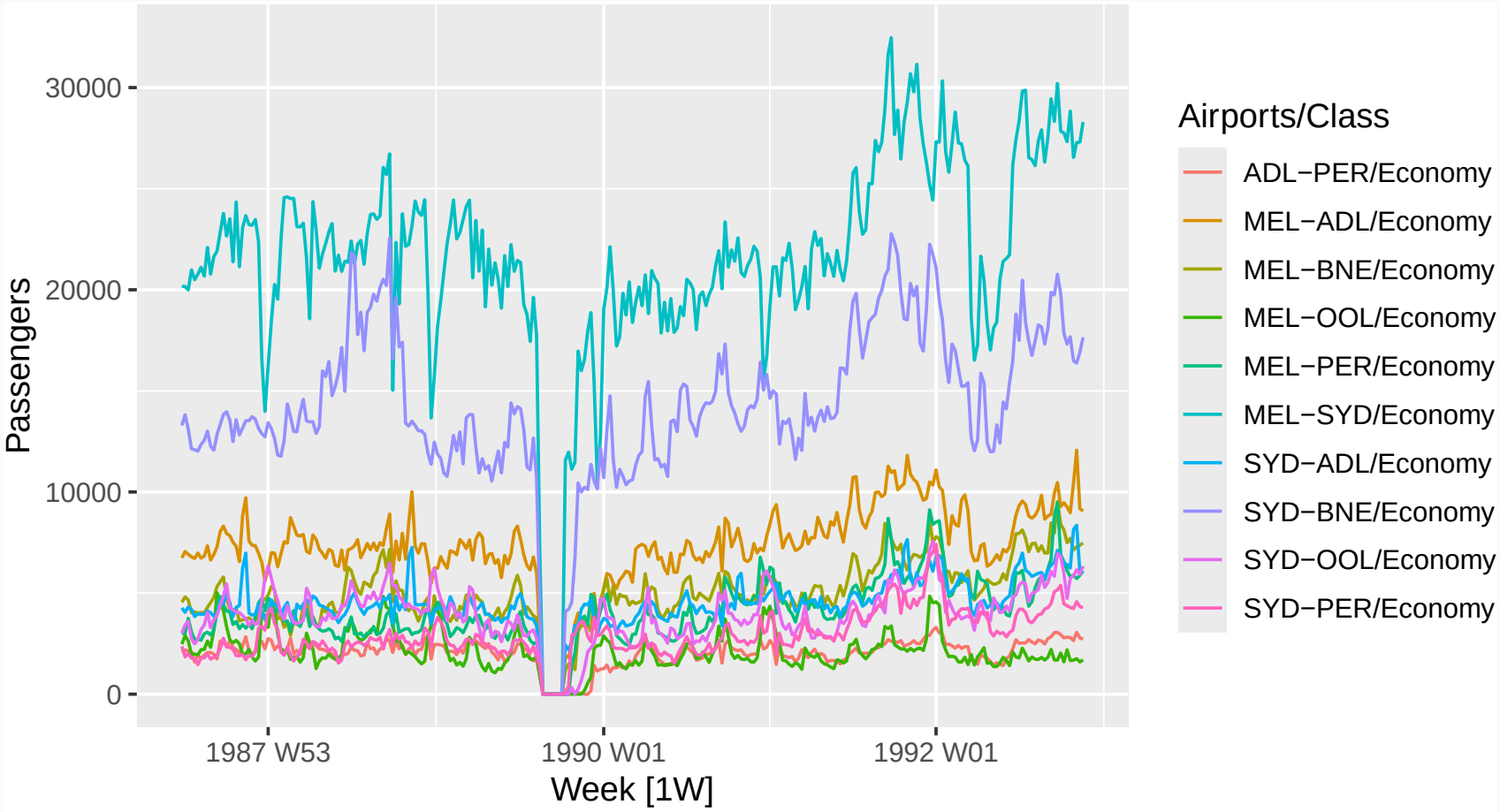
Ansett airlines

```
ansett %>%
  autoplot(Passengers)
```



Ansett airlines

```
ansett %>%
  filter(Class == "Economy") %>%
  autoplot(Passengers)
```



Time series patterns

Time series patterns

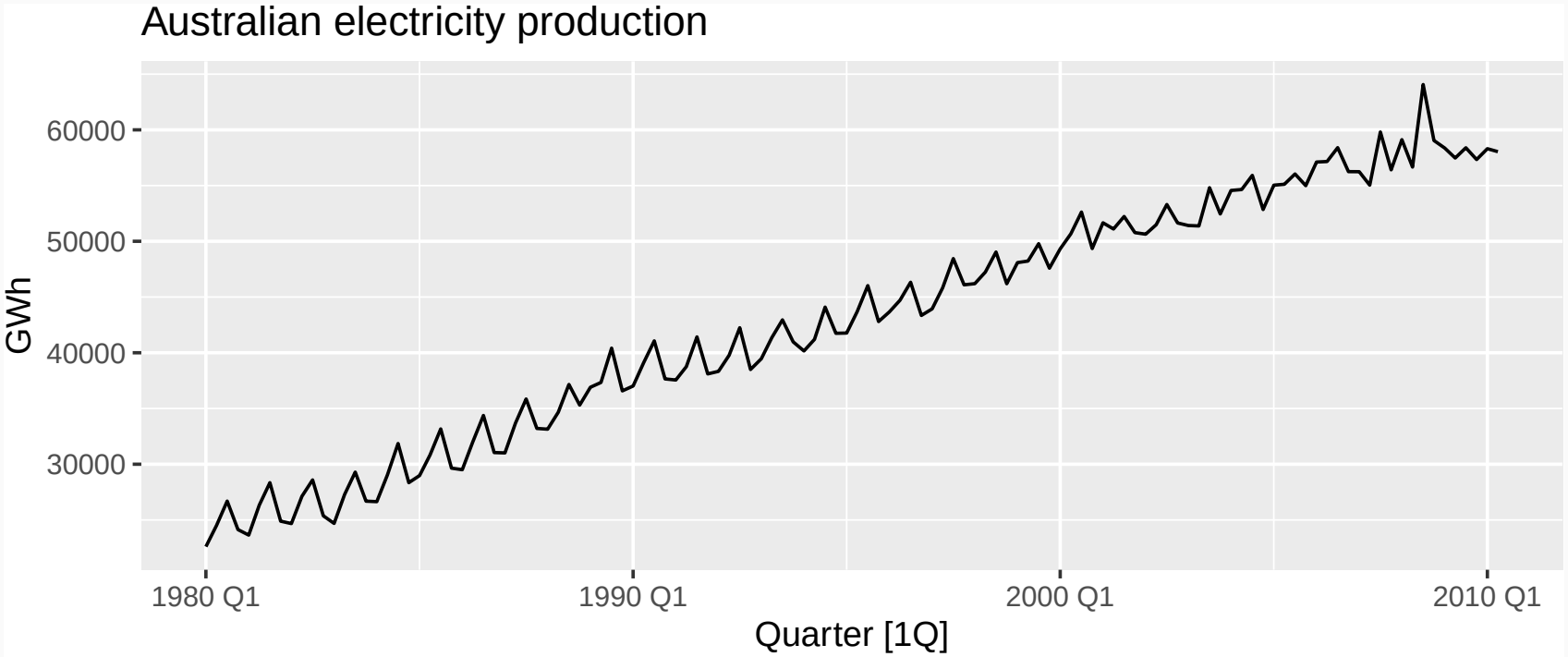
Trend pattern exists when there is a long-term increase or decrease in the data.

Seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year, the month, or day of the week).

Cyclic pattern exists when data exhibit rises and falls that are *not of fixed period* (duration usually of at least 2 years).

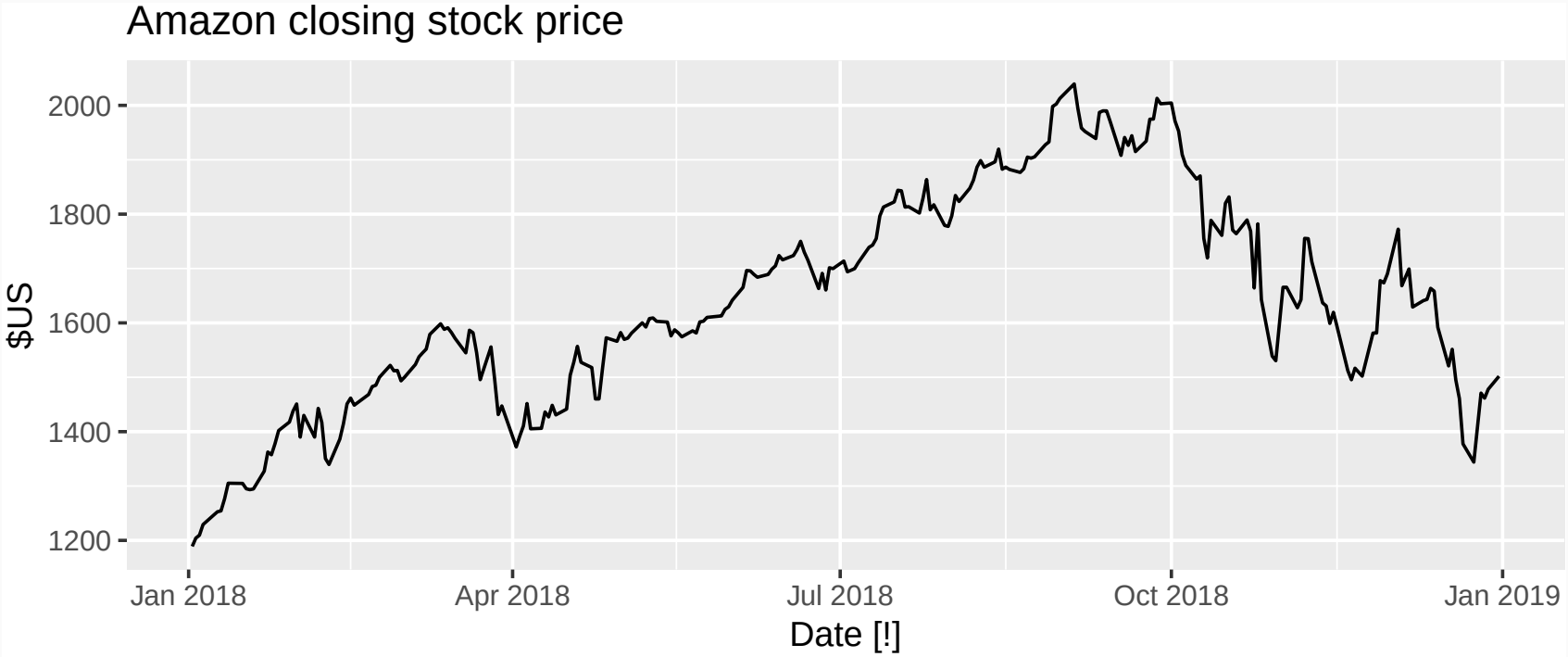
Time series patterns

```
aus_production %>%
  filter(year(Quarter) >= 1980) %>%
  autoplot(Electricity) +
  labs(y = "GWh",
       title = "Australian electricity production")
```



Time series patterns

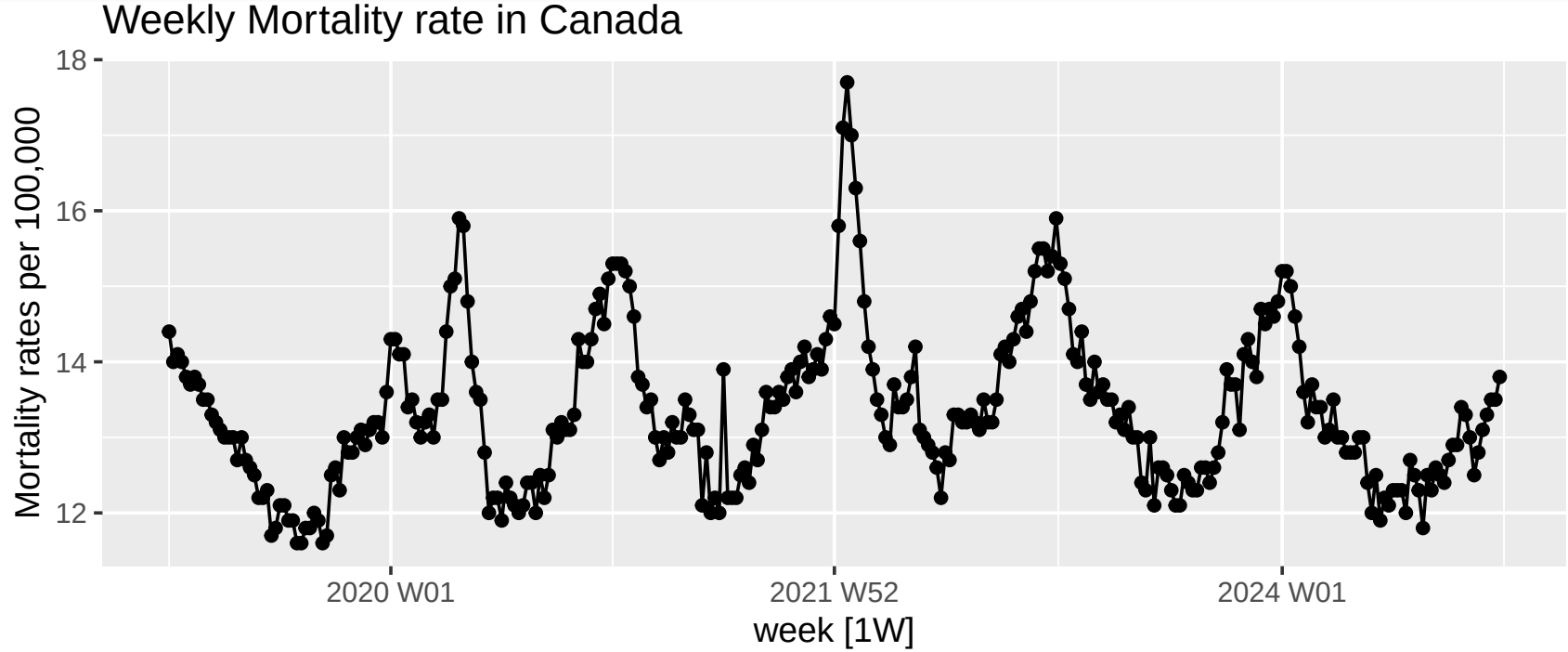
```
gafa_stock %>%
  filter(Symbol == "AMZN", year(Date) >= 2018) %>%
  autoplot(Close) +
  labs(y = "$US",
       title = "Amazon closing stock price")
```



Seasonal plots

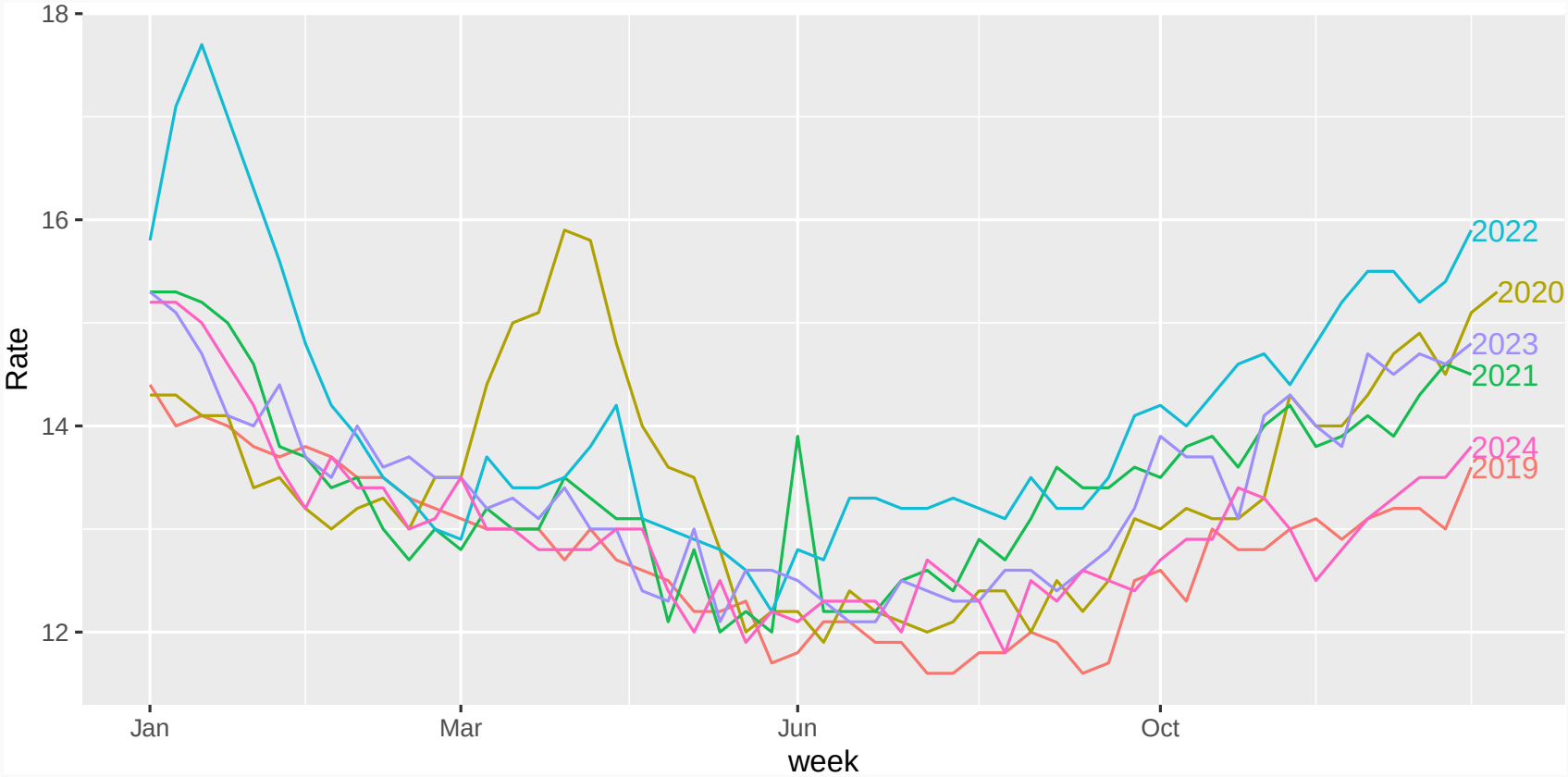
Example: Weekly Mortality Rate in Canada

```
mortality_ts %>% autoplot(Rate) + geom_point() +
  labs(title = "Weekly Mortality rate in Canada",
       y="Mortality rates per 100,000")
```



Mortality Rate in Canada

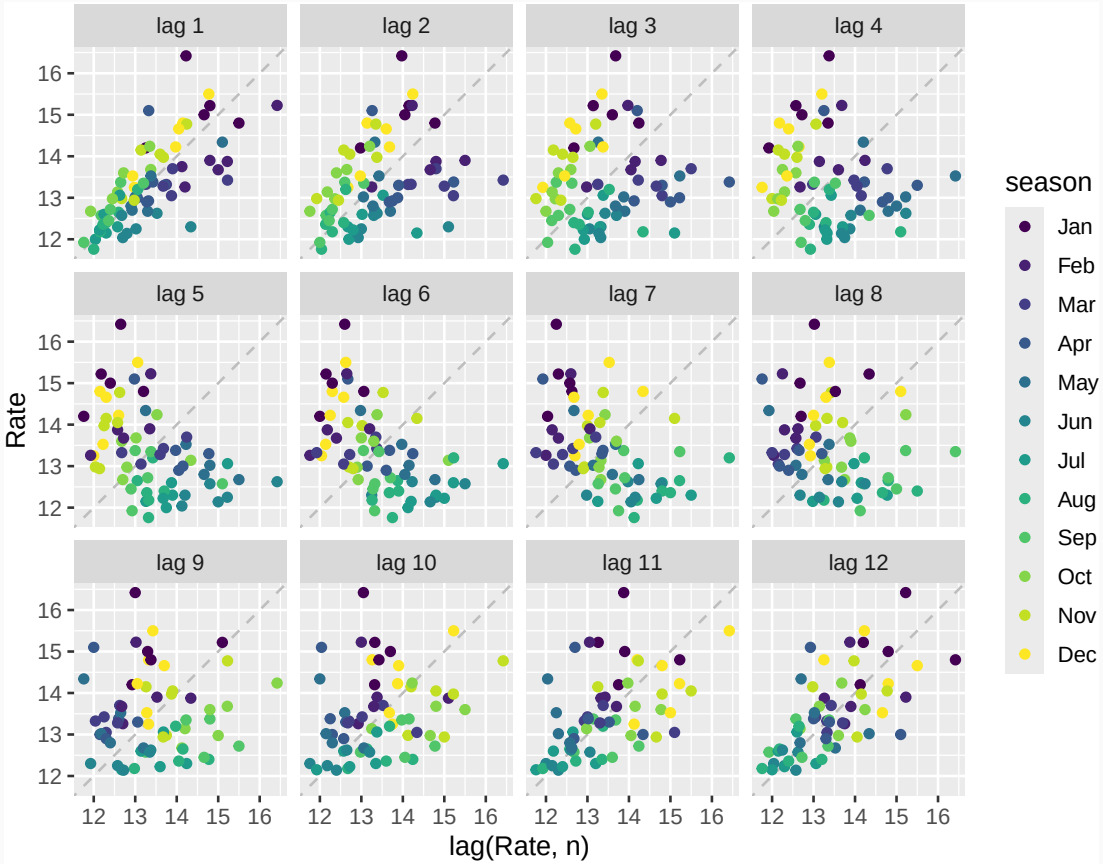
```
mortality_ts %>% gg_season(Rate, labels="right")
```



Lag plots and autocorrelation

Example: Monthly Mortality Rate in Canada

```
mortality_month_ts %>% gg_lag(Rate, geom = "point", lags=1:12)
```



Each graph shows y_t plotted against y_{t-k} for different values of k.

Lagged scatterplots

- Each graph shows y_t plotted against y_{t-k} for different values of k.
- The autocorrelations are the correlations associated with these scatterplots.
- ACF (autocorrelation function):
  - $\rho_1 = \text{Correlation}(y_t, y_{t-1})$
  - $\rho_2 = \text{Correlation}(y_t, y_{t-2})$
  - $\rho_3 = \text{Correlation}(y_t, y_{t-3})$
  - etc.

Autocorrelation

Covariance and **correlation**: measure extent of **linear relationship** between two variables (y and X).

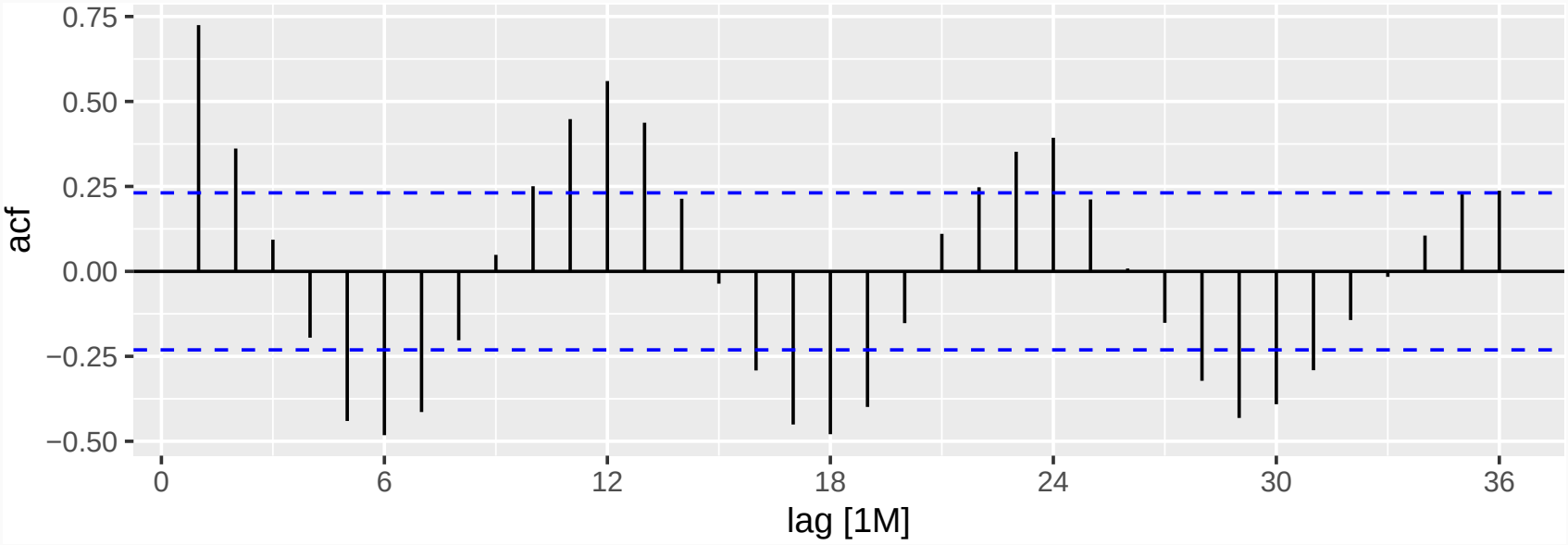
Autocovariance and **autocorrelation**: measure linear relationship between **lagged values** of a time series y.

We measure the relationship between:

- y_t and y_{t-1}
- y_t and y_{t-2}
- y_t and y_{t-3}
- etc.

Autocorrelation: Monthly mortality rate in Canada

```
mortality_month_ts %>% ACF(Rate, lag_max = 36) %>% autoplot()
```



- The plot is known as a **correlogram**
- ρ_1 higher than for the other lags. This is due to **the seasonal pattern in the data**: the peaks tend to be **12 months** apart and the downturns tend to be **6 months** apart.
- ρ_6 is more negative than for the other lags because downturns tend to be 6 months behind peaks.

Trend and seasonality in ACF plots

- When data have a trend, the autocorrelations for small lags tend to be large and positive.
- When data are seasonal, the autocorrelations will be larger at the seasonal lags (i.e., at multiples of the seasonal frequency)
- When data are trended and seasonal, you see a combination of these effects.

Time series decomposition

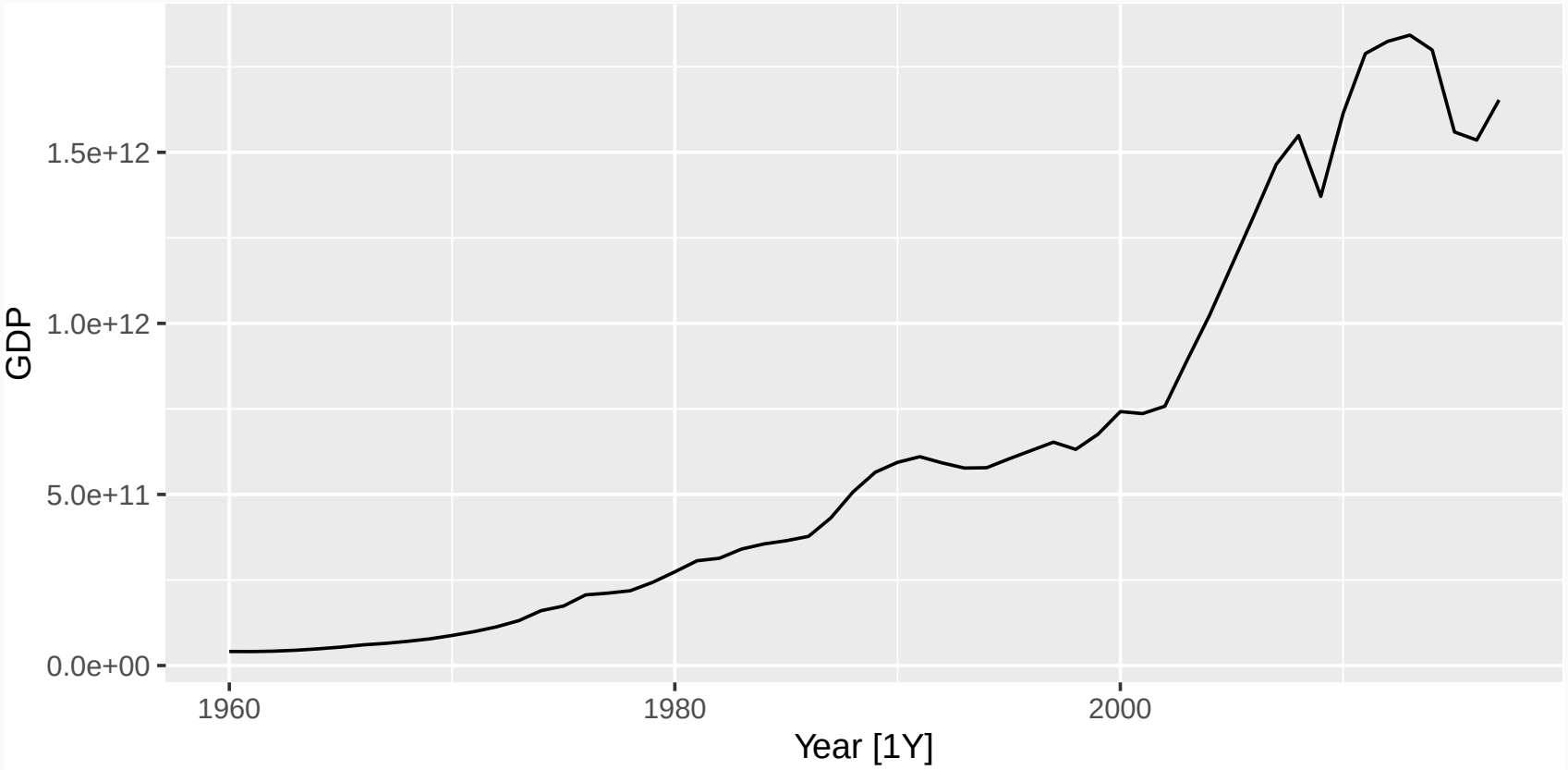
Time series decomposition

- We can think of a time series as comprising three components:
  - a trend-cycle component,
  - a seasonal component, and
  - a remainder component (containing anything else in the time series).
- Here, we consider the most common methods for extracting these components from a time series. Often this is done to help improve understanding of the time series, but it can also be used to improve forecast accuracy.
- When decomposing a time series, it is sometimes helpful to first transform or adjust the series in order to make the decomposition (and later analysis) as simple as possible.

Transformations and adjustments

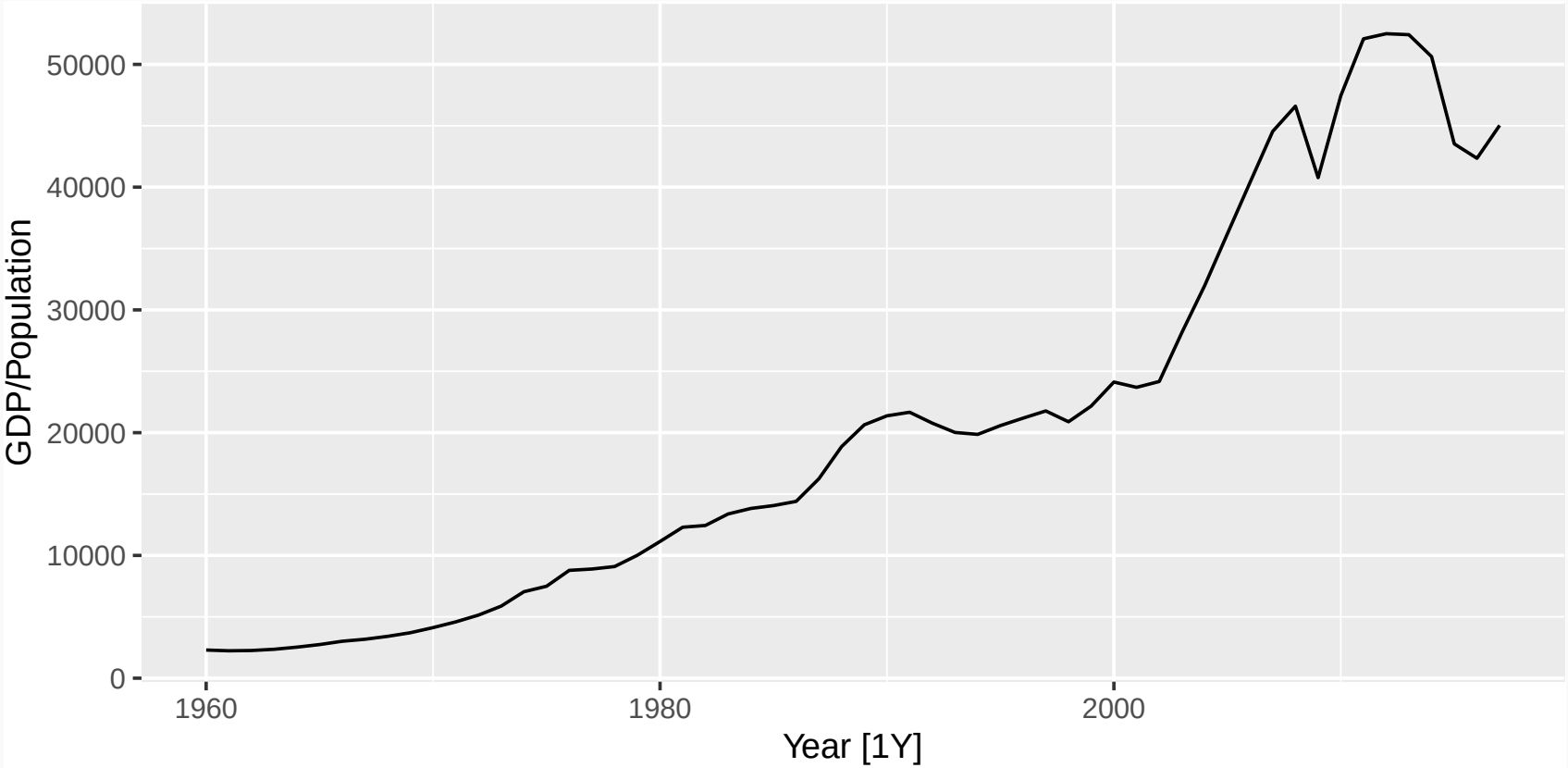
Population Adjustement/Per capita adjustments

```
global_economy %>%
  filter(Country == "Canada") %>%
  autoplot(GDP)
```



Per capita adjustments

```
global_economy %>%
  filter(Country == "Canada") %>%
  autoplot(GDP / Population)
```



Mathematical transformations

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

Mathematical transformations for stabilizing variation

Square root	$w_t = \sqrt{y_t}$	$\downarrow$
Cube root	$w_t = \sqrt[3]{y_t}$	Increasing
Logarithm	$w_t = \log(y_t)$	strength

Logarithms, in particular, are useful because they are more interpretable: changes in a log value are **relative (percent) changes on the original scale**.

Box-Cox transformations

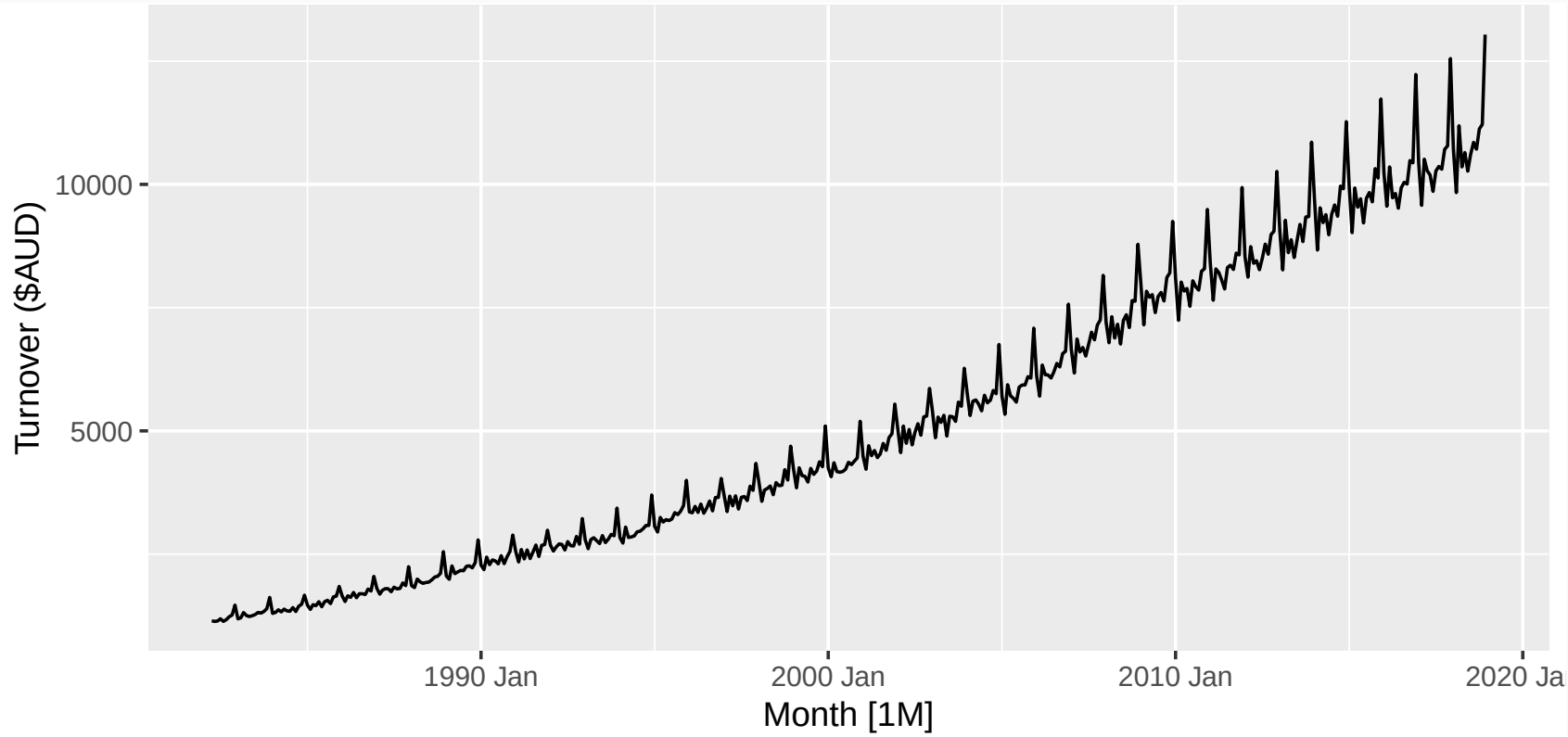
Each of these transformations is close to a member of the family of **Box-Cox transformations**:

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

- $\lambda = 1$: (No substantive transformation)
- $\lambda = \frac{1}{2}$: (Square root plus linear transformation)
- $\lambda = 0$: (Natural logarithm)
- $\lambda = -1$: (Inverse plus 1)

Example: Box-Cox transformations

```
food <- aus_retail %>%
  filter(Industry == "Food retailing") %>%
  summarise(Turnover = sum(Turnover))
```



Box-Cox transformations

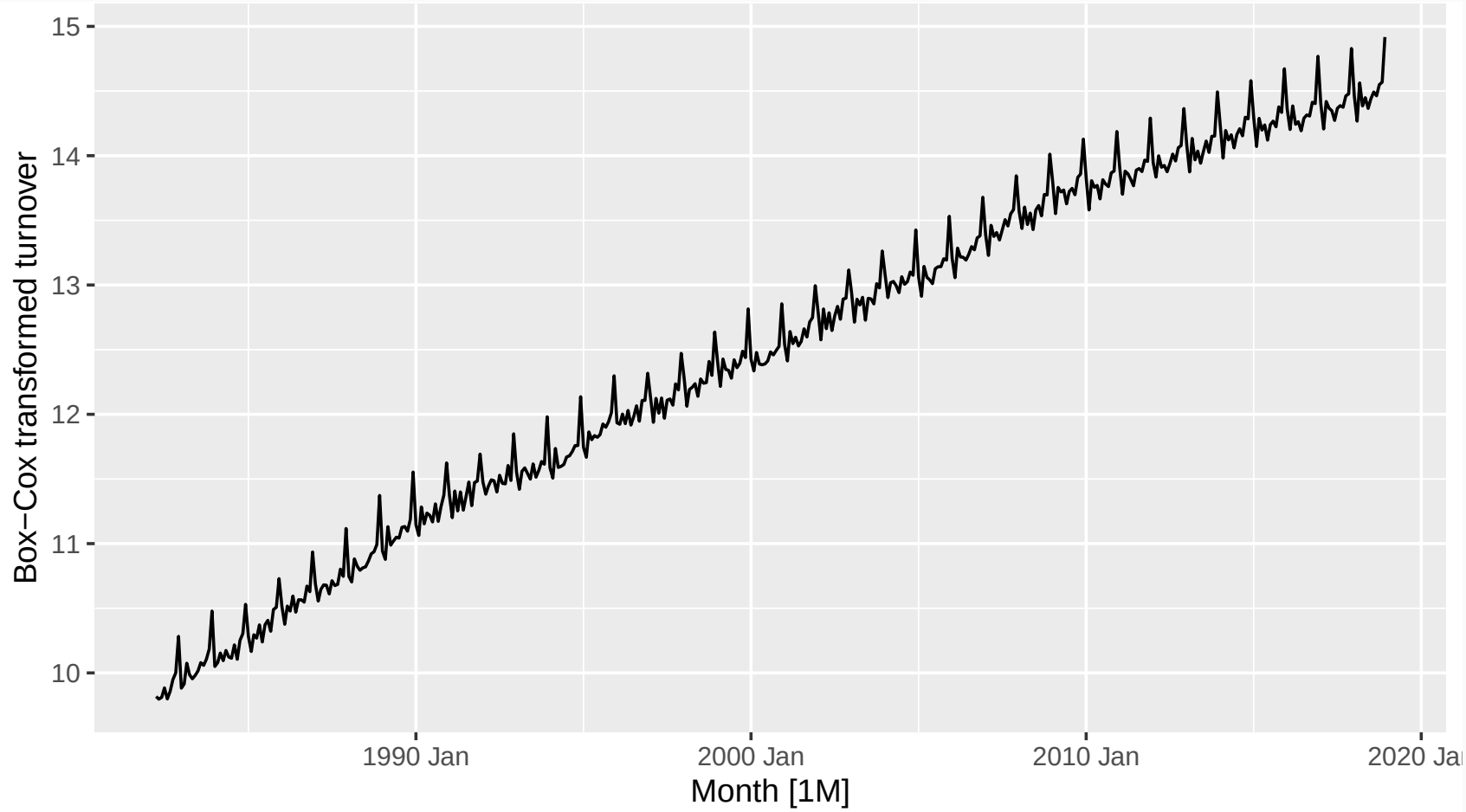
```
food %>%
  features(Turnover, features = guerrero)
```

```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1          0.0895
```

- This attempts to balance the seasonal fluctuations and random variation across the series.
- Always check the results.
- A low value of λ can give extremely large prediction intervals.

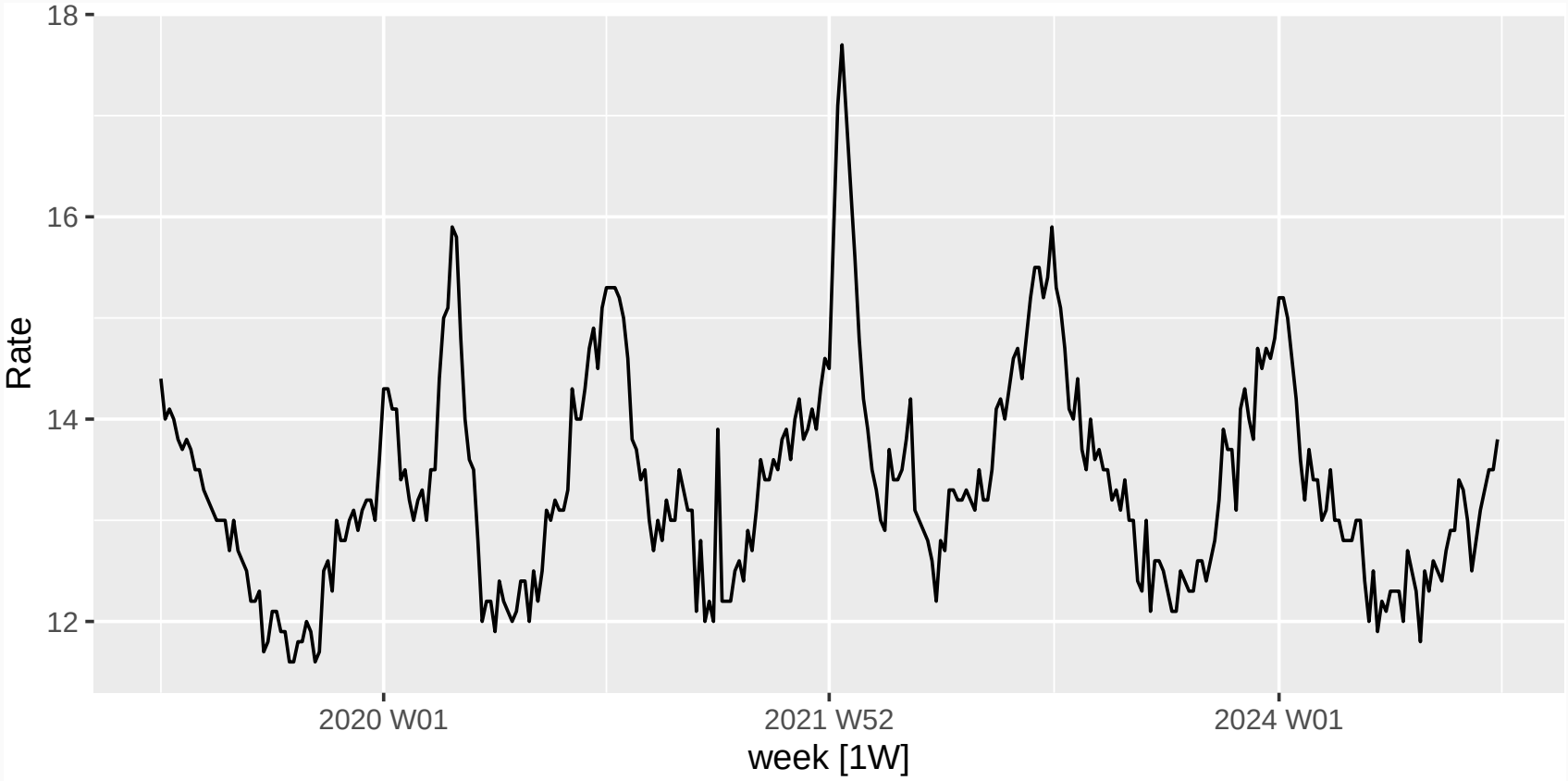
Box-Cox transformations

```
food %>% autoplot(box_cox(Turnover, 0.0895)) +
  labs(y = "Box-Cox transformed turnover")
```



Box-Cox transformations: Mortality rate in Canada

```
mortality_ts %>% autoplot(Rate)
```



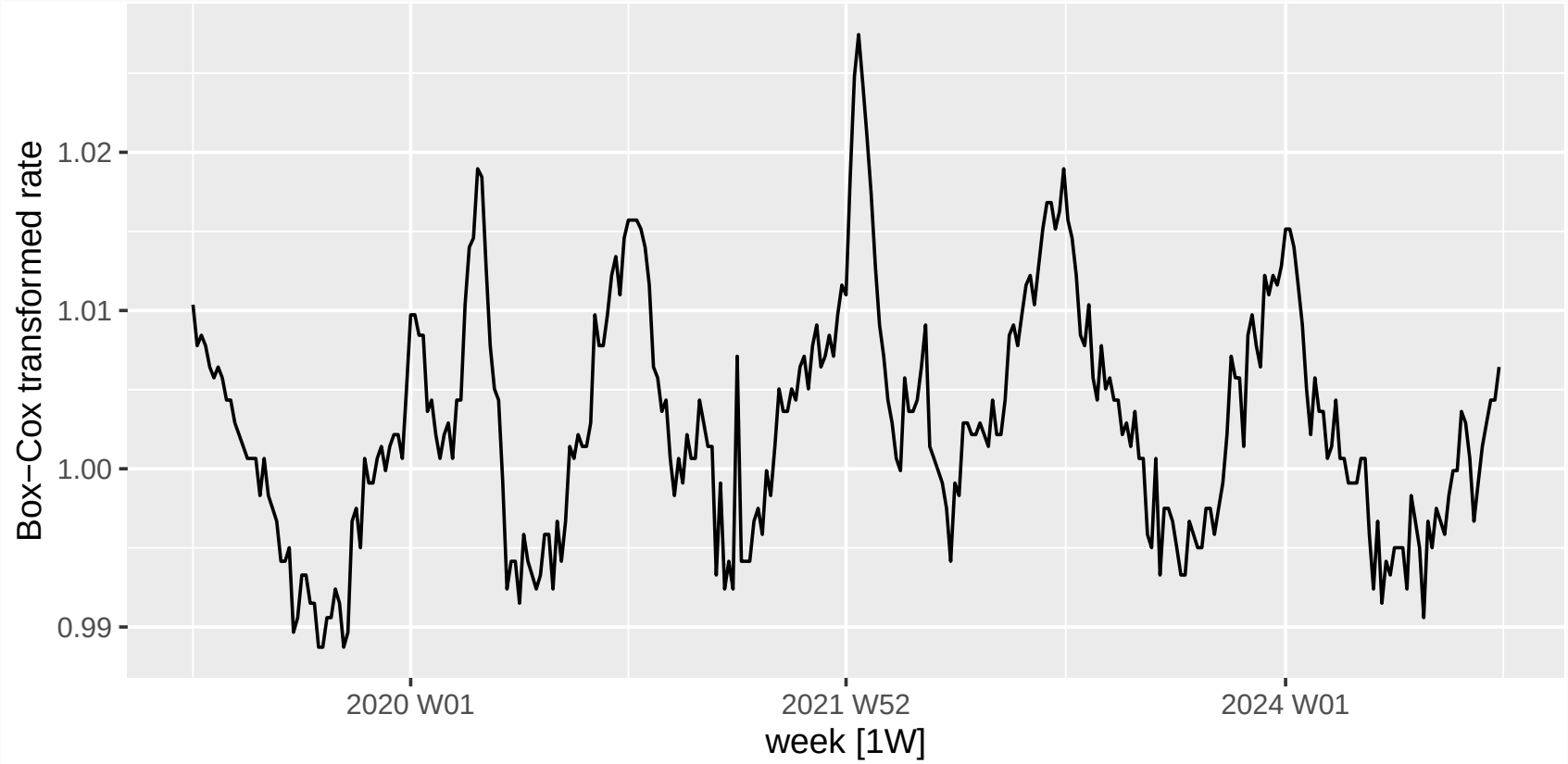
Box-Cox transformations: Mortality rate in Canada

```
mortality_ts %>%
  features(Rate, features = guerrero)
```

```
## # A tibble: 1 x 3
##   GEO    Sex         lambda_guerrero
##   <chr>  <chr>                 <dbl>
## 1 Canada Both sexes           -0.900
```

Box-Cox transformations: Mortality rate in Canada

```
mortality_ts %>% autoplot(box_cox(Rate, -0.900)) +
  labs(y = "Box-Cox transformed rate")
```



Transformations

- Often no transformation needed.
- Simple transformations are easier to explain and work well enough.
- Transformations can have very large effect on PI.
- If some data are zero or negative, then use $\lambda > 0$.
- Choosing logs is a simple way to force forecasts to be positive
- Transformations must be reversed to obtain forecasts on the original scale. (Handled automatically by `fable`.)

Time series decomposition

$$y_t = f(S_t, T_t, R_t)$$

where
$y_t =$ data at period t
$T_t =$ trend-cycle component at period t
$S_t =$ seasonal component at period t
$R_t =$ remainder component at period t

Additive decomposition: $y_t = S_t + T_t + R_t$.

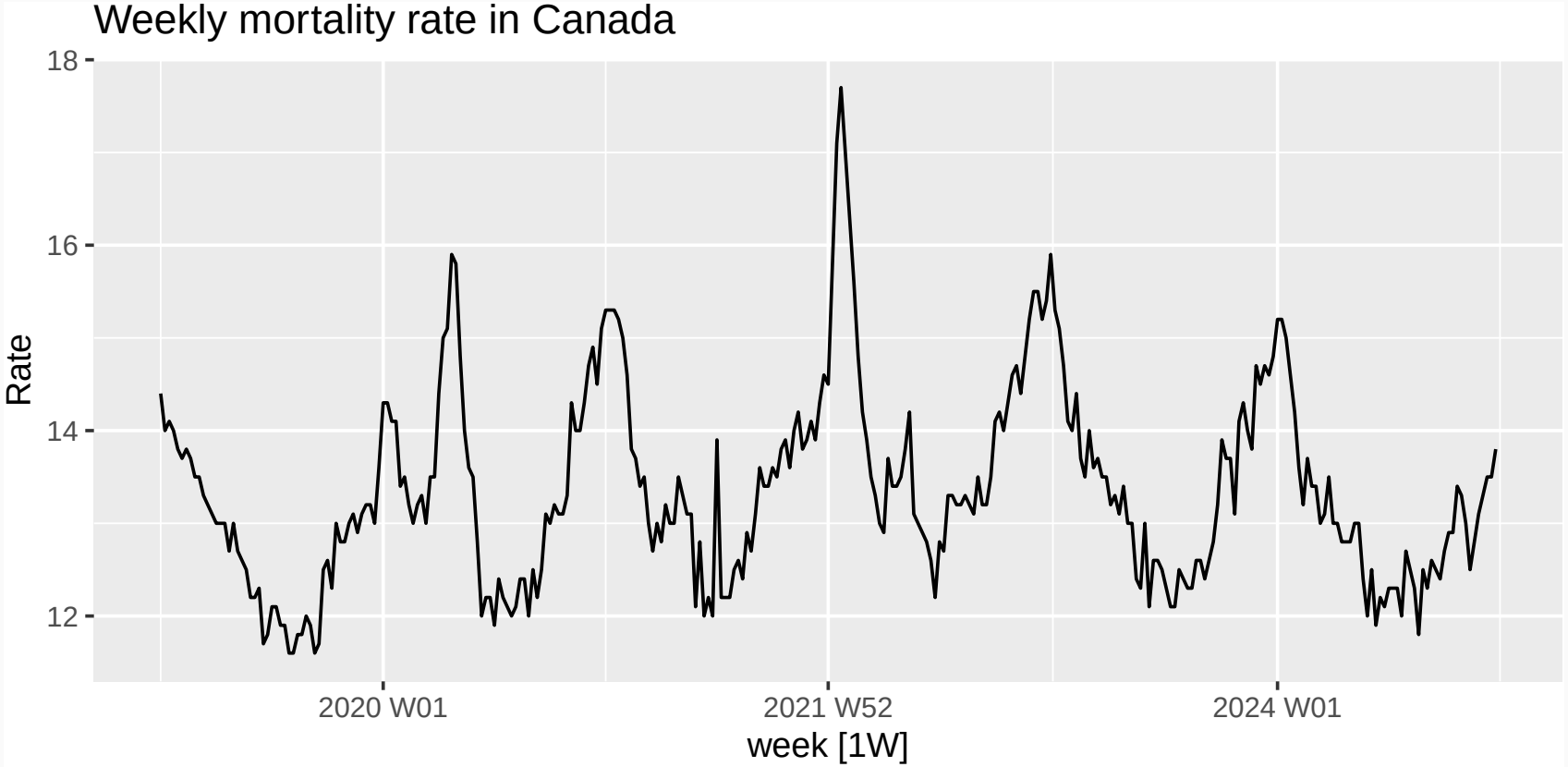
Multiplicative decomposition: $y_t = S_t \times T_t \times R_t$.

- Logs turn multiplicative relationship into an additive relationship:

$$y_t = S_t \times T_t \times R_t \quad \Rightarrow \quad \log y_t = \log S_t + \log T_t + \log R_t.$$

Mortality rate in Canada

```
mortality_ts %>%
  autoplot(Rate) +
  labs(y = "Rate",
       title = "Weekly mortality rate in Canada")
```



Mortality rate in Canada

```
mortality_ts %>%
  model(stl = STL(Rate))
```

```
## # A mable: 1 x 3
## # Key:     GEO, Sex [1]
##   GEO    Sex            stl
##   <chr>  <chr>      <model>
## 1 Canada Both sexes   <STL>
```

- STL (Seasonal and Trend decomposition using Loess) is a versatile and robust method for decomposing time series.

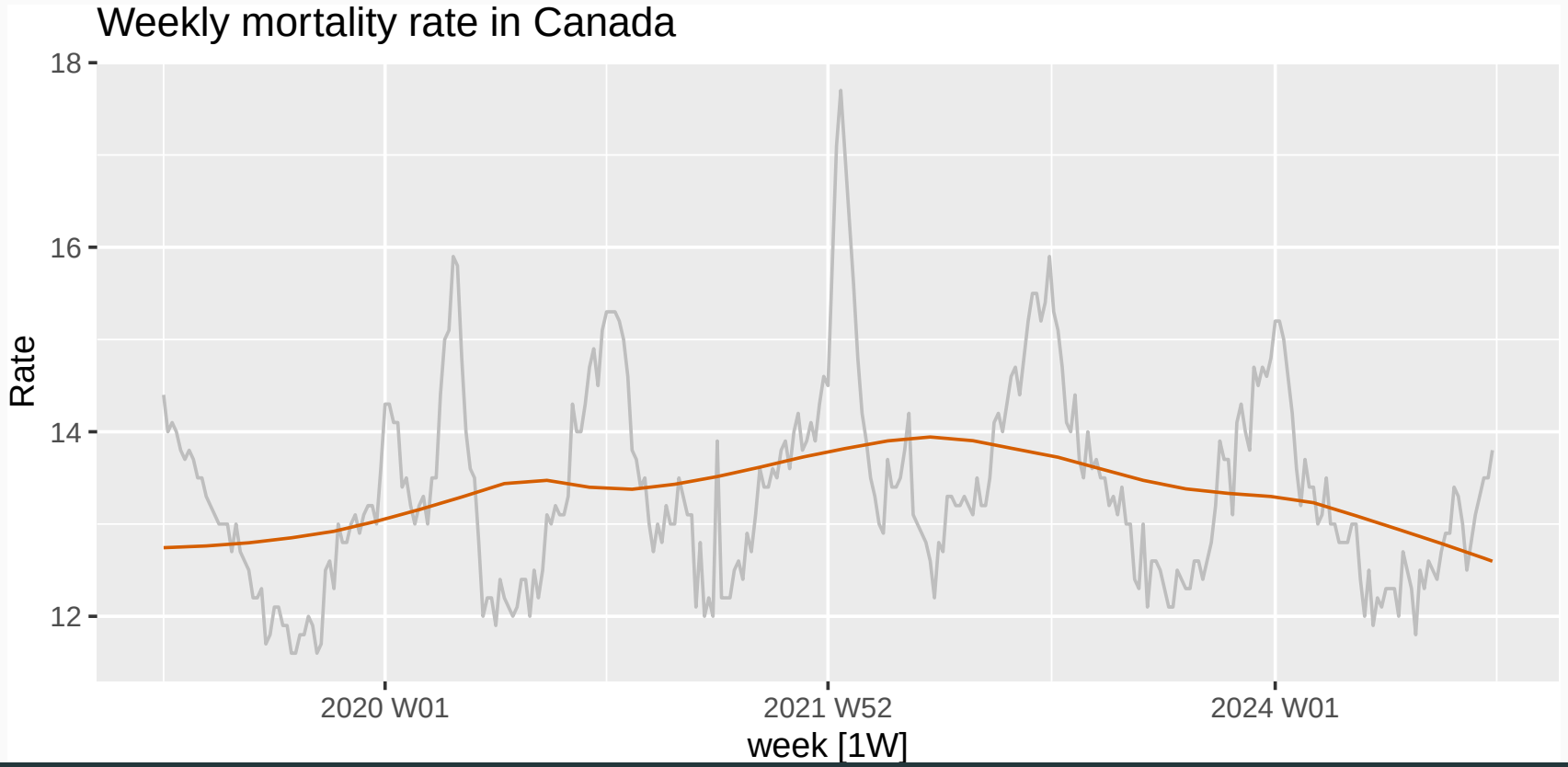
Mortality rate in Canada

```
dcmp <- mortality_ts %>%
  model(stl = STL(Rate))
components(dcmp)
```

```
## # A dable: 313 x 9 [1W]
## # Key:     GEO, Sex, .model [1]
## # :        Rate = trend + season_year + remainder
##    GEO    Sex         .model     week  Rate trend season_year
##    <chr>  <chr>       <chr>     <week> <dbl> <dbl>       <dbl>
##  1 Canada Both sexes stl    2019 W01  14.4  12.7       1.49
##  2 Canada Both sexes stl    2019 W02  14    12.7       1.58
##  3 Canada Both sexes stl    2019 W03  14.1  12.7       1.76
##  4 Canada Both sexes stl    2019 W04  14    12.7       1.76
##  5 Canada Both sexes stl    2019 W05  13.8  12.8       1.30
##  6 Canada Both sexes stl    2019 W06  13.7  12.8       1.06
##  7 Canada Both sexes stl    2019 W07  13.8  12.8       0.737
##  8 Canada Both sexes stl    2019 W08  13.7  12.8       0.400
##  9 Canada Both sexes stl    2019 W09  13.5  12.8       0.234
## 10 Canada Both sexes stl    2019 W10  13.5  12.8       0.264
## # i 303 more rows
```

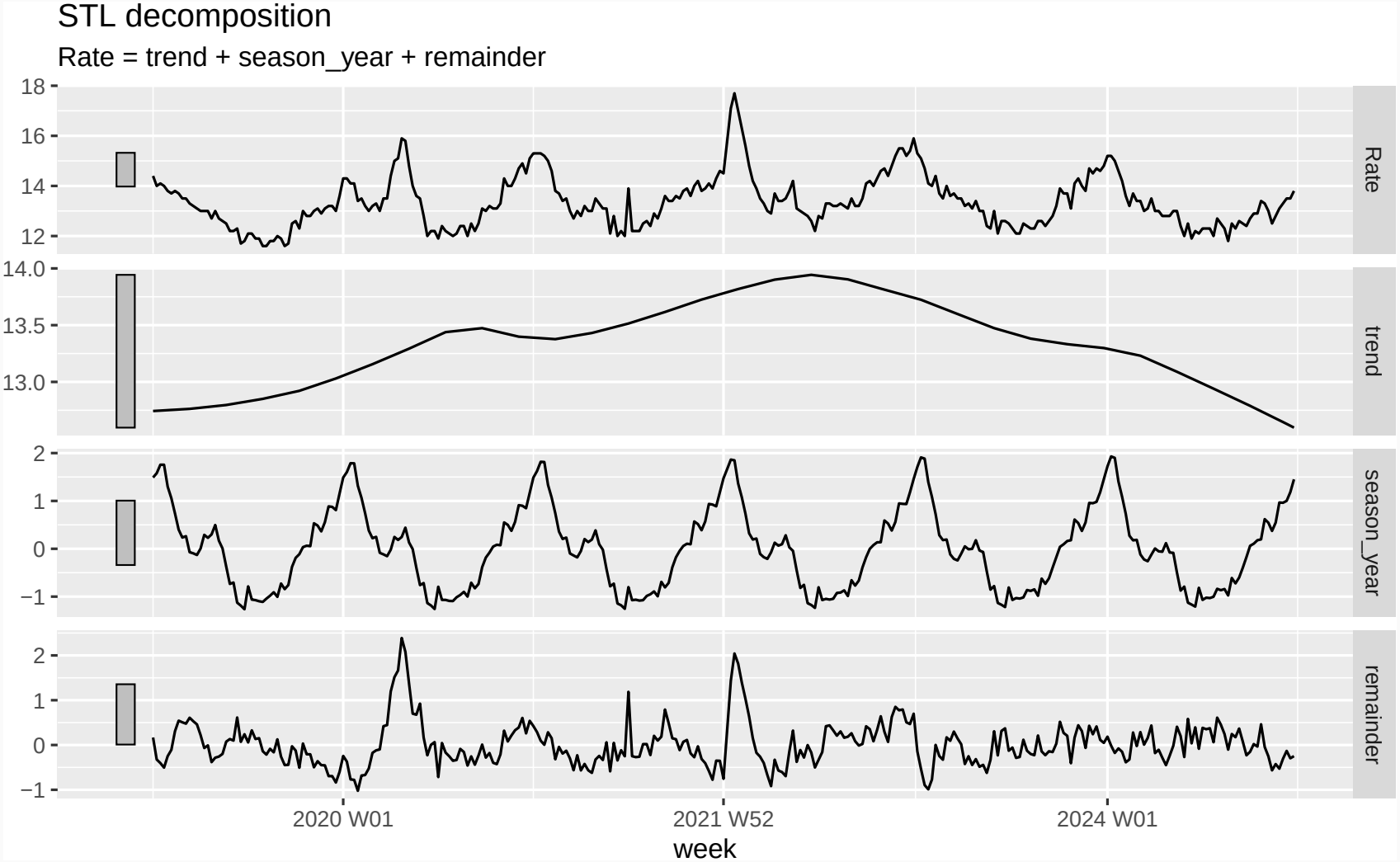
Mortality rate in Canada

```
mortality_ts %>%
  autoplot(Rate, color='gray') +
  autolayer(components(dcmp), trend, color='#D55E00') +
  labs(y = "Rate",
       title = "Weekly mortality rate in Canada")
```



Mortality rate in Canada

```
components(dcmp) %>% autoplot()
```



Seasonal adjustment

- Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.
- Additive decomposition: seasonally adjusted data given by

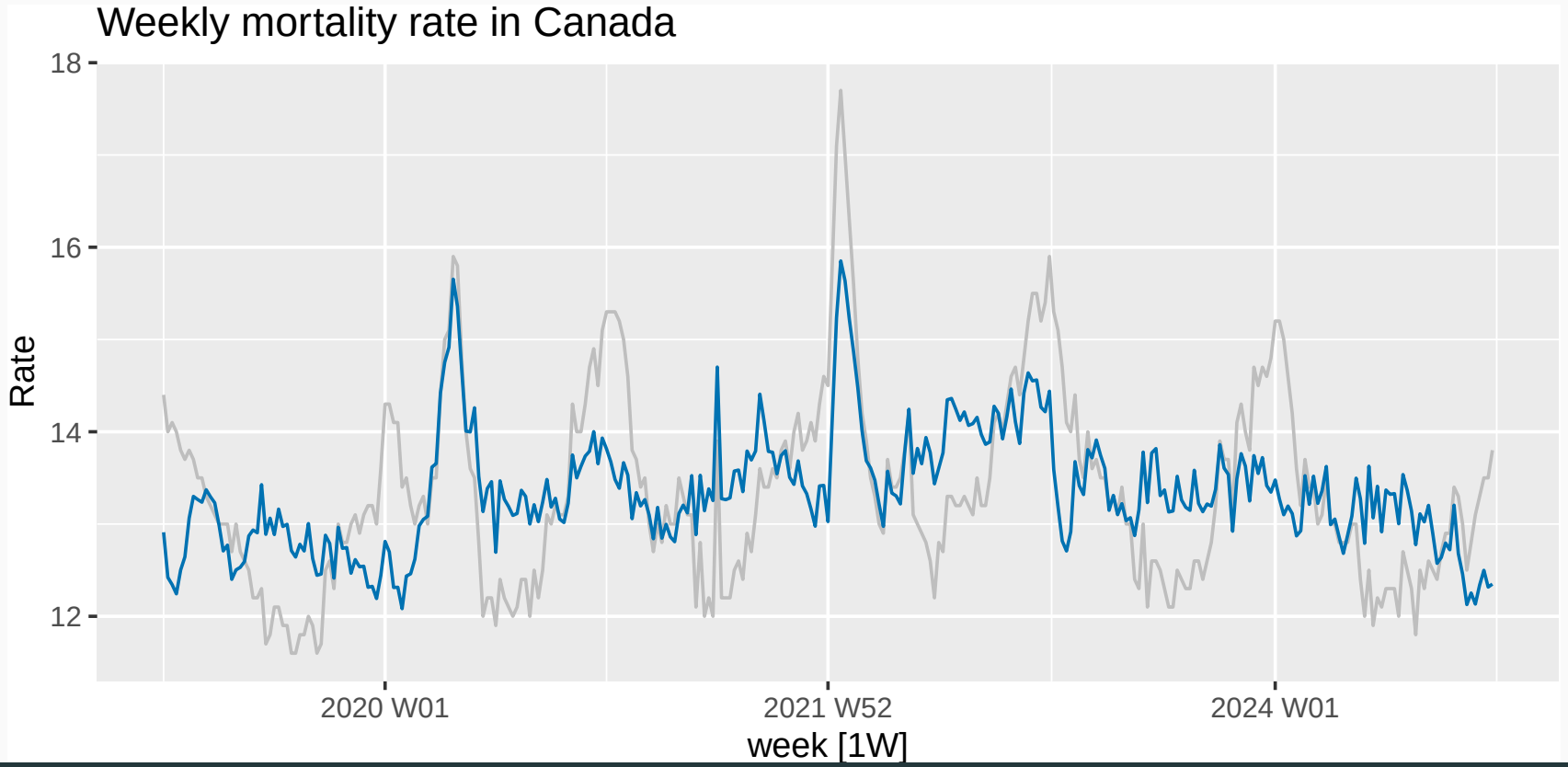
$$y_t - S_t = T_t + R_t$$

- Multiplicative decomposition: seasonally adjusted data given by

$$y_t / S_t = T_t \times R_t$$

Mortality rate in Canada

```
mortality_ts %>%
  autoplot(Rate, color='gray') +
  autolayer(components(dcmp), season_adjust, color='#0072B2') +
  labs(y = "Rate",
       title = "Weekly mortality rate in Canada")
```



Classical Decomposition

- The traditional way to do time series decomposition is called `Classical decomposition`.
- The first step in a classical decomposition is to use a moving average method.
- The simplest estimate of the trend-cycle uses moving averages.
- A moving average of order m can be written as

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j}, \quad \text{where} \quad m = 2k + 1$$

Moving Average Smoothing

So a moving average is an average of nearby points

- observations nearby in time are also likely to be close in value.
- average eliminates some randomness in the data, leaving a smooth trend-cycle component.

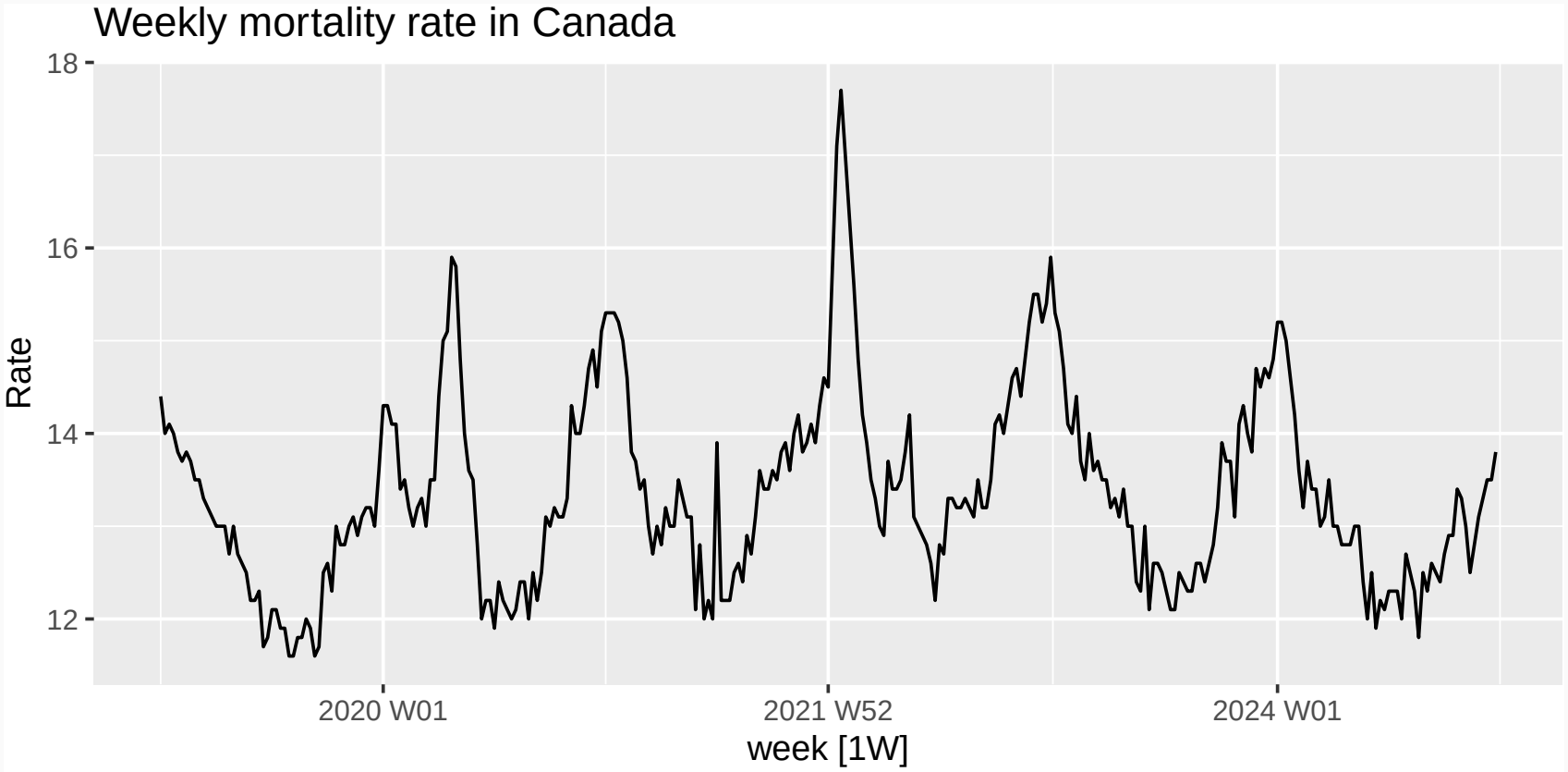
$$\text{3-MA:} \quad \hat{T}_t = (y_{t-1} + y_t + y_{t+1})/3$$

$$\text{5-MA:} \quad \hat{T}_t = (y_{t-2} + y_{t-1} + y_t + y_{t+1} + y_{t+2})/5$$

- each average computed by dropping oldest observation and including next observation.
- averaging moves through time series until trend-cycle computed at each observation possible.

Moving averages: example

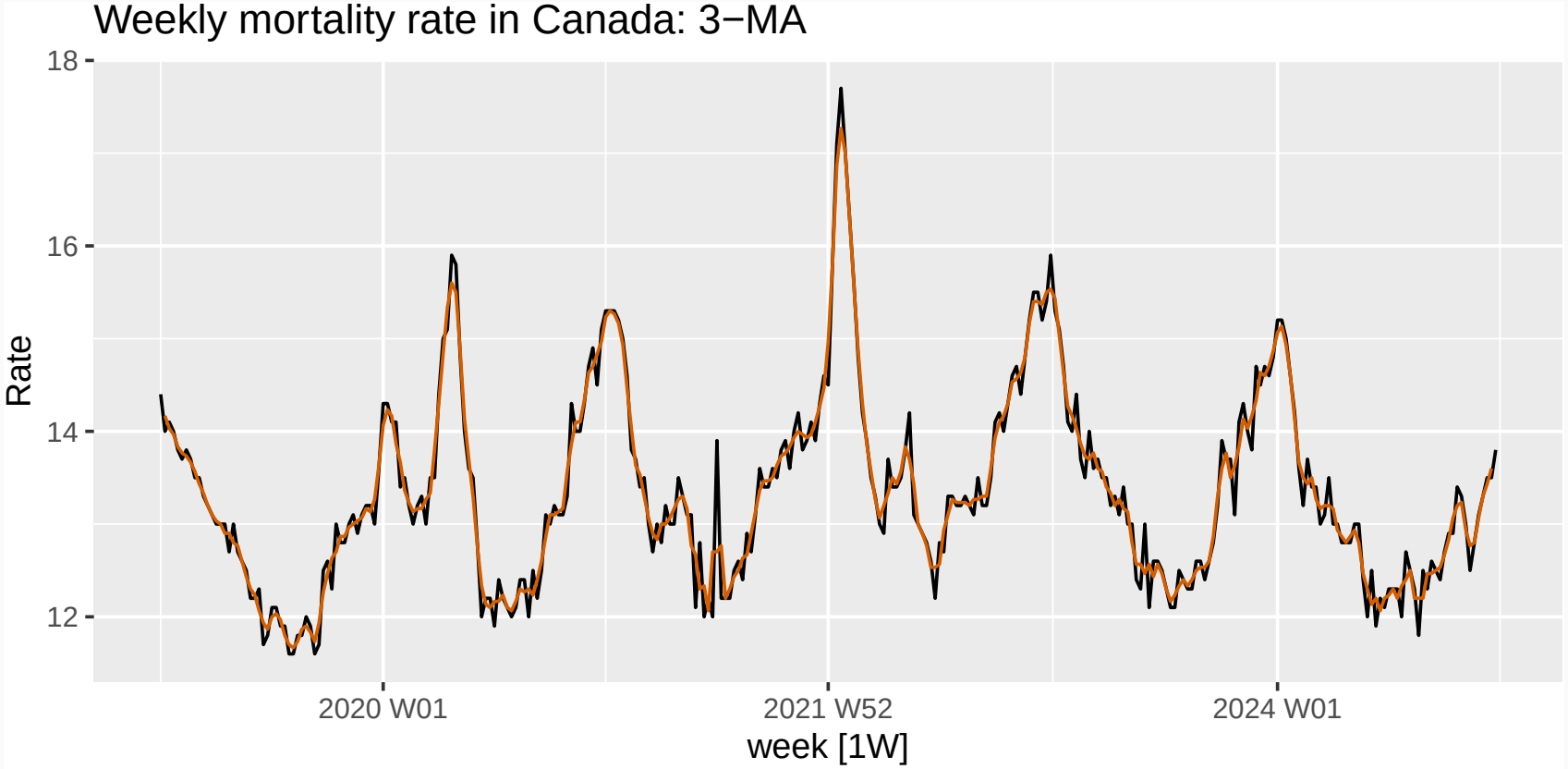
```
mortality_ts |> autoplot(Rate) +
  labs(y = "Rate", title = "Weekly mortality rate in Canada")
```



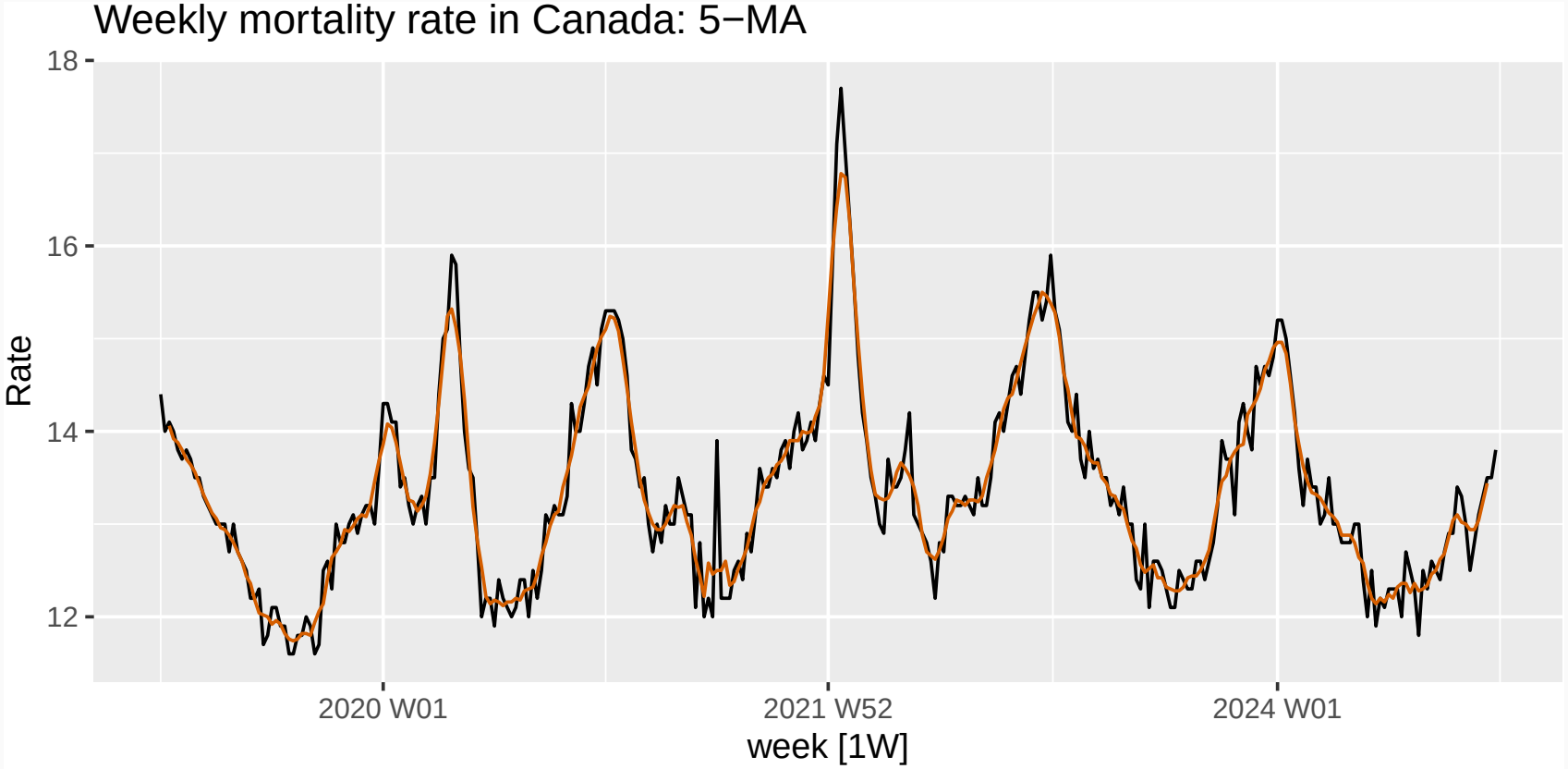
Moving average smoothing

week	Rate	5-MA
2019 W01	14.40	
2019 W02	14.00	
2019 W03	14.10	14.06
2019 W04	14.00	13.92
2019 W05	13.80	13.88
...	...	...
2024 W47	12.80	12.94
2024 W48	13.10	13.04
2024 W49	13.30	13.24
2024 W50	13.50	13.44
2024 W51	13.50	
2024 W52	13.80	

Moving average smoothing



Moving average smoothing



Final thoughts!

This first lecture provides the introduction and background of time series features to have a good understanding of our time series, their patterns and characteristics, before we attempt to build any models and produce any forecasts.

More details can be found in Chapters 1, 2 and 3 of Forecasting: Principles and Practice (3rd ed, `https://otexts.com/fpp3/`), as well as in many other time series forecasting resources.