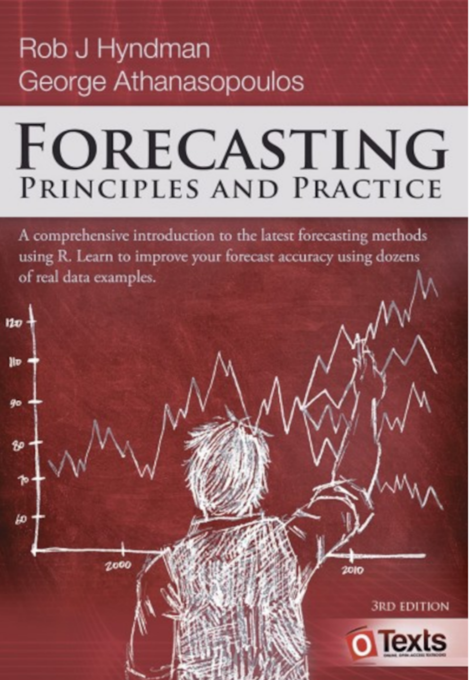


9. ARIMA models

9.1 Stationary and differencing

OTexts.org/fpp3/

Stationarity

Definition

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \ldots, y_{t+s})$ does not depend on t.

Stationarity

Definition

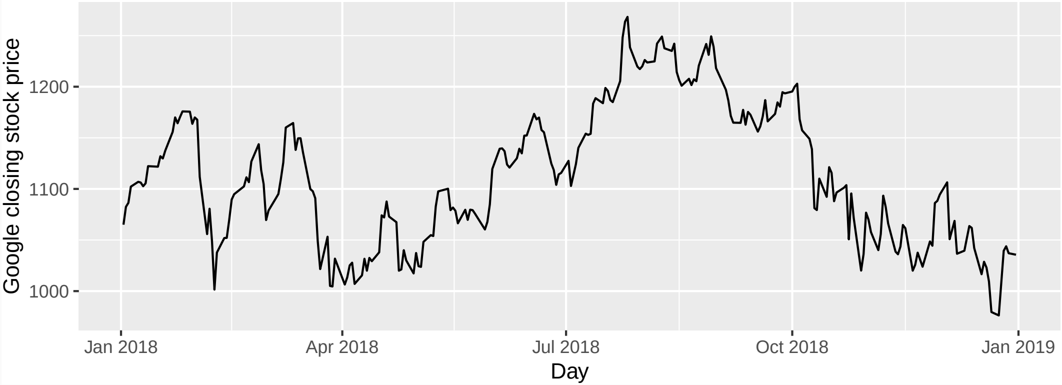
If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \dots, y_{t+s})$ does not depend on t.

A **stationary series** is:

- roughly horizontal
- constant variance
- no patterns predictable in the long-term

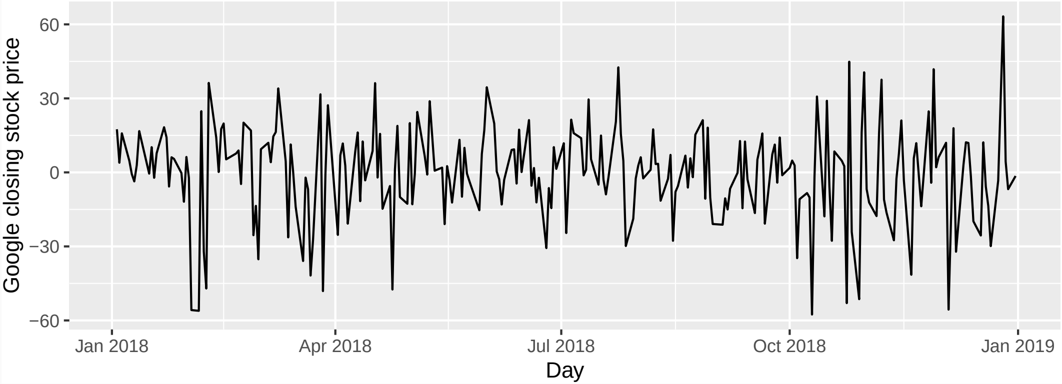
Stationary?

```r
gafa_stock |>
  filter(Symbol == "GOOG", year(Date) == 2018) |>
  autoplot(Close) +
  labs(y = "Google closing stock price", x = "Day")
```



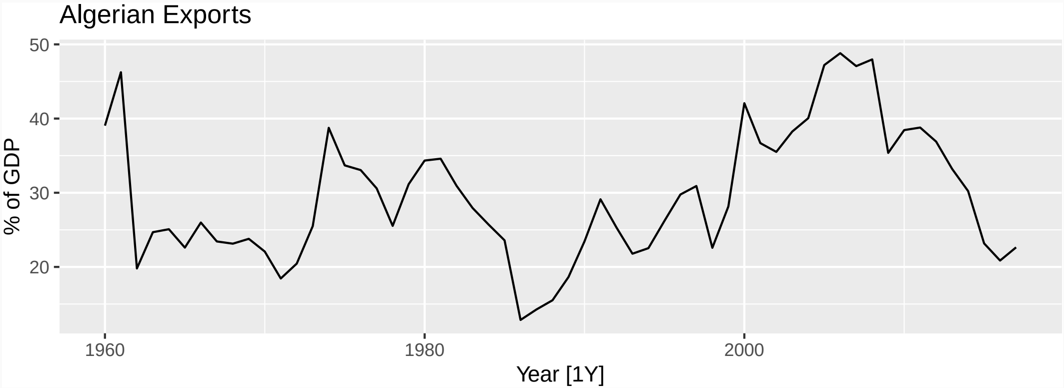
Stationary?

```
gafa_stock |>
  filter(Symbol == "GOOG", year(Date) == 2018) |>
  autoplot(difference(Close)) +
  labs(y = "Google closing stock price", x = "Day")
```



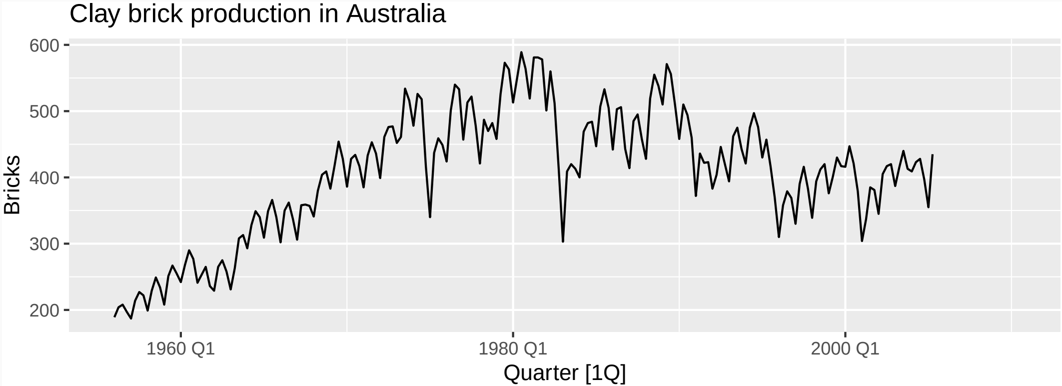
Stationary?

```r
global_economy |>
  filter(Country == "Algeria") |>
  autoplot(Exports) +
  labs(y = "% of GDP", title = "Algerian Exports")
```



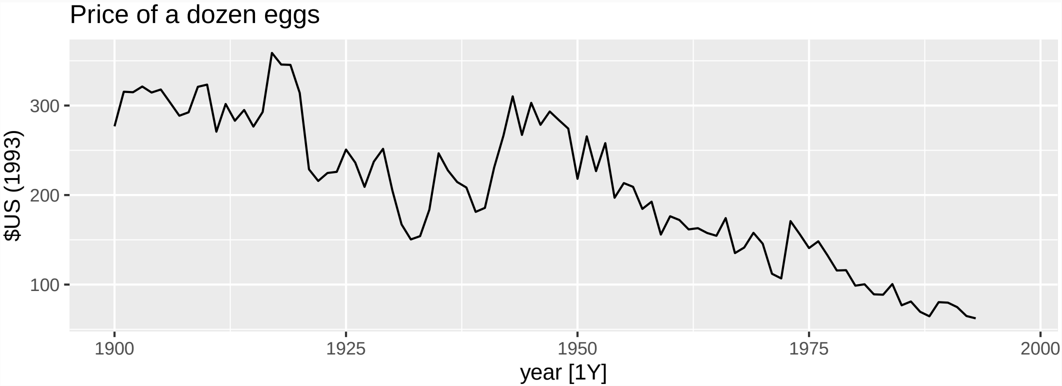
Stationary?

```
aus_production |>
  autoplot(Bricks) +
  labs(title = "Clay brick production in Australia")
```



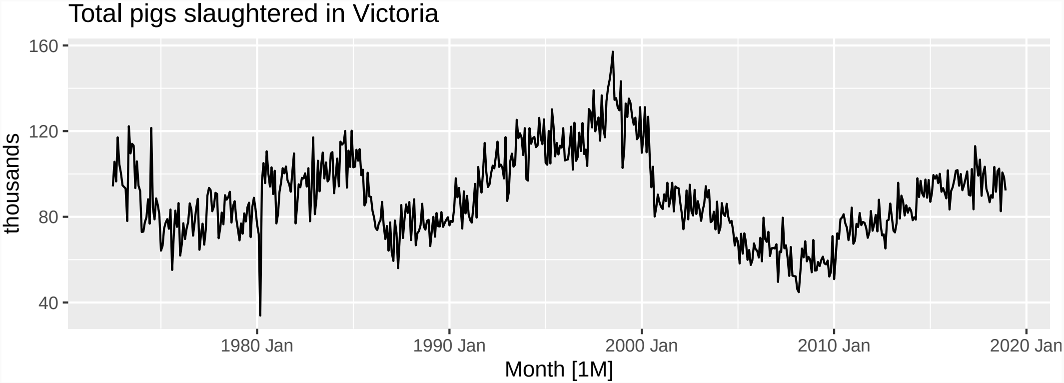
Stationary?

```
prices |>
  filter(year >= 1900) |>
  autoplot(eggs) +
  labs(y = "$US (1993)", title = "Price of a dozen eggs")
```



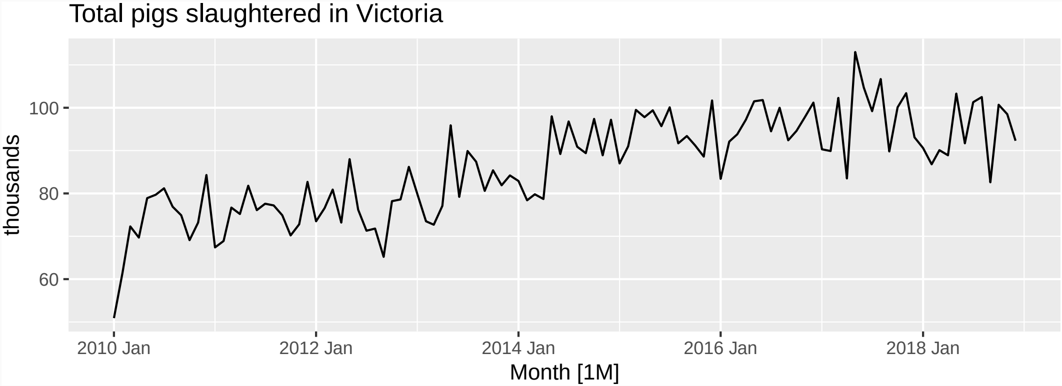
Stationary?

```r
aus_livestock |>
  filter(Animal == "Pigs", State == "Victoria") |>
  autoplot(Count / 1e3) +
  labs(y = "thousands", title = "Total pigs slaughtered in Victoria")
```



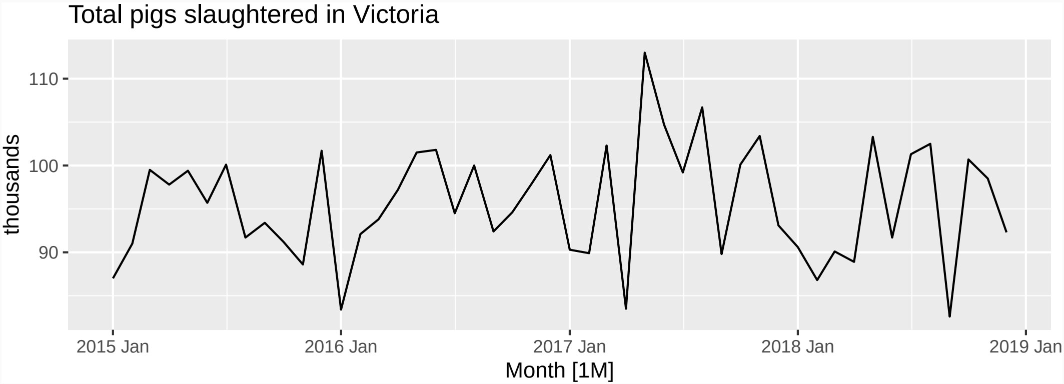
Stationary?

```
aus_livestock |>
  filter(Animal == "Pigs", State == "Victoria", year(Month) >= 2010) |>
  autoplot(Count / 1e3) +
  labs(y = "thousands", title = "Total pigs slaughtered in Victoria")
```



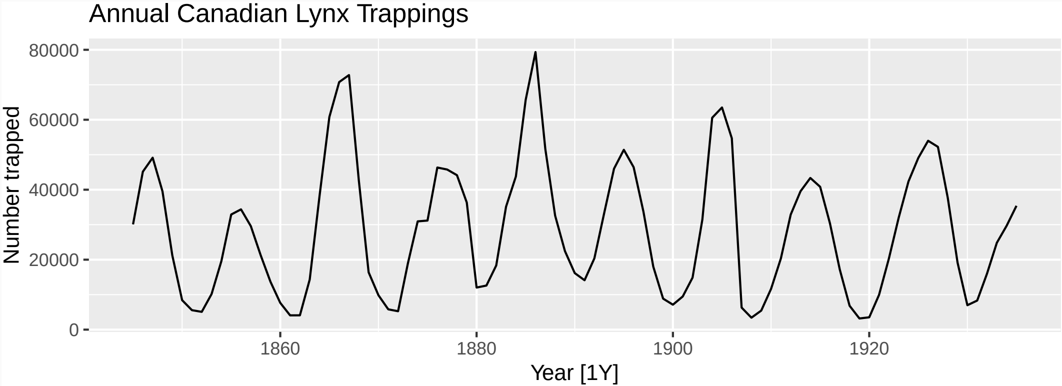
Stationary?

```
aus_livestock |>
  filter(Animal == "Pigs", State == "Victoria", year(Month) >= 2015) |>
  autoplot(Count / 1e3) +
  labs(y = "thousands", title = "Total pigs slaughtered in Victoria")
```



Stationary?

```
pelt |>
  autoplot(Lynx) +
  labs(y = "Number trapped", title = "Annual Canadian Lynx Trappings")
```



Stationarity

Definition

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \ldots, y_{t+s})$ does not depend on t.

Stationarity

Definition

If $\{y_t\}$ is a stationary time series, then for all s, the distribution of $(y_t, \ldots, y_{t+s})$ does not depend on t.

Transformations help to **stabilize the variance**.

For ARIMA modelling, we also need to **stabilize the mean**.

Non-stationarity in the mean

Identifying non-stationary series

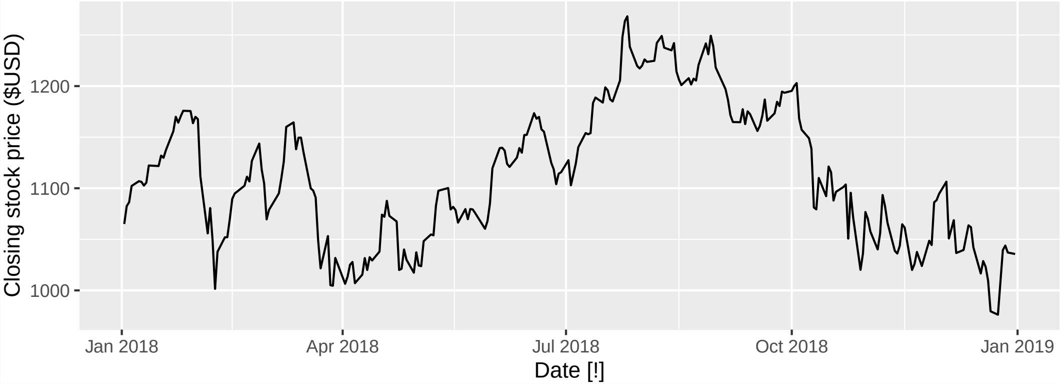
- time plot.
- The ACF of stationary data drops to zero relatively quickly
- The ACF of non-stationary data decreases slowly.
- For non-stationary data, the value of r_1 is often large and positive.

Example: Google stock price

```
google_2018 <- gafa_stock |>
  filter(Symbol == "GOOG", year(Date) == 2018)
```

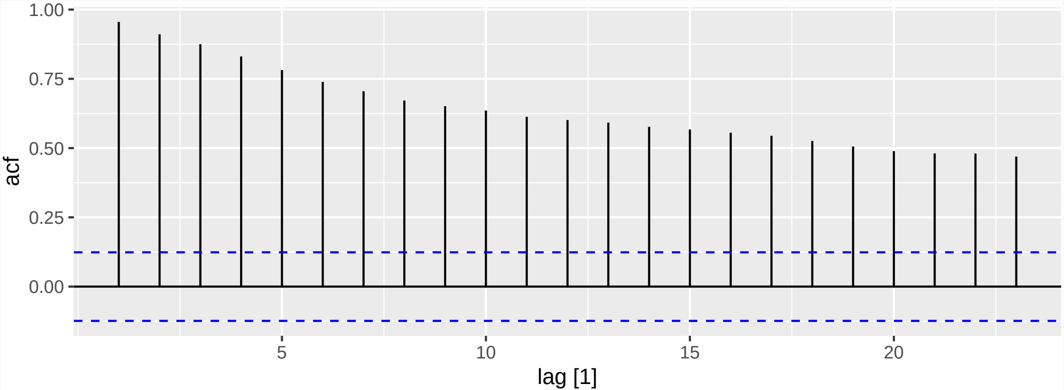
Example: Google stock price

```r
google_2018 |>
  autoplot(Close) +
  labs(y = "Closing stock price ($USD)")
```



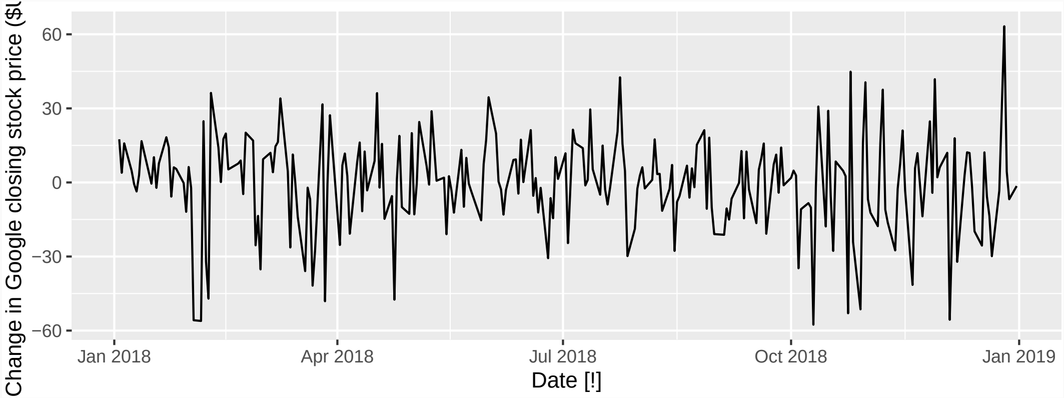
Example: Google stock price

```
google_2018 |>
  ACF(Close) |>
  autoplot()
```



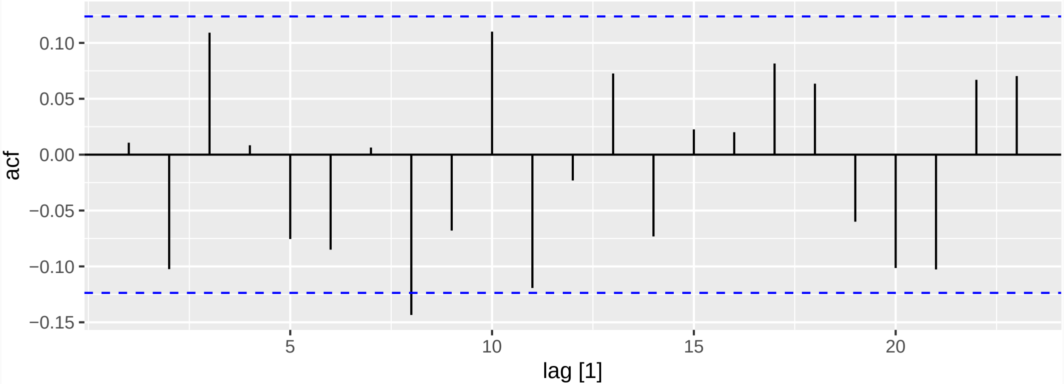
Example: Google stock price

```
google_2018 |>
  autoplot(difference(Close)) +
  labs(y = "Change in Google closing stock price ($USD)")
```



Example: Google stock price

```
google_2018 |>
  ACF(difference(Close)) |>
  autoplot()
```



Differencing

- Differencing helps to **stabilize the mean**.
- The differenced series is the *change* between each observation in the original series: $y'_t = y_t - y_{t-1}$.
- The differenced series will have only $T - 1$ values since it is not possible to calculate a difference y'_1 for the first observation.

Random walk model

If differenced series is white noise with zero mean:

$$y_t - y_{t-1} = \varepsilon_t \qquad \text{or} \qquad y_t = y_{t-1} + \varepsilon_t$$

where $\varepsilon_t \sim NID(0, \sigma^2)$.

- Very widely used for non-stationary data.
- This is the model behind the **naïve method**.
- Random walks typically have:
  - long periods of apparent trends up or down
  - Sudden/unpredictable changes in direction
- Forecast are equal to the last observation
  - future movements up or down are equally likely.

Random walk with drift model

If differenced series is white noise with non-zero mean:

$$y_t - y_{t-1} = c + \varepsilon_t \quad \text{or} \quad y_t = c + y_{t-1} + \varepsilon_t$$

where $\varepsilon_t \sim NID(0, \sigma^2)$.

- c is the **average change** between consecutive observations.
- If $c > 0$, y_t will tend to drift upwards and vice versa.
- This is the model behind the **drift method**.

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2}. \end{aligned}$$

Second-order differencing

Occasionally the differenced data will not appear stationary and it may be necessary to difference the data a second time:

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2}. \end{aligned}$$

- y''_t will have $T - 2$ values.
- In practice, it is almost never necessary to go beyond second-order differences.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

$$y'_t = y_t - y_{t-m}$$

where m = number of seasons.

Seasonal differencing

A seasonal difference is the difference between an observation and the corresponding observation from the previous year.

$$y'_t = y_t - y_{t-m}$$

where m = number of seasons.

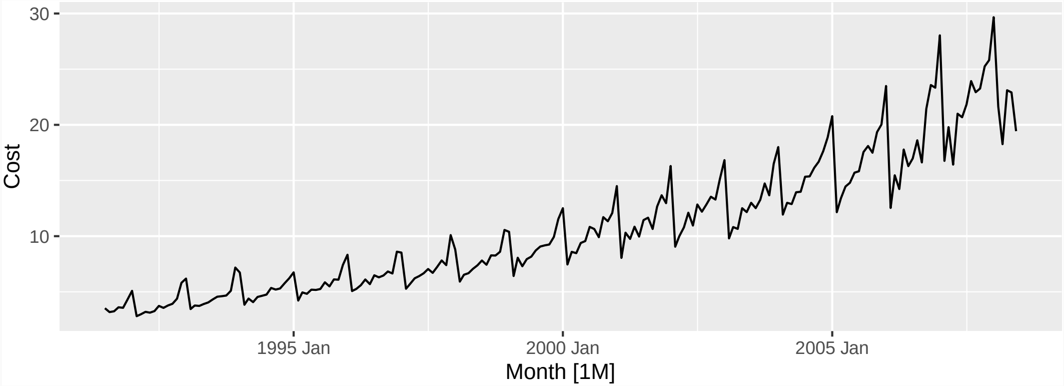
- For monthly data $m = 12$.
- For quarterly data $m = 4$.

Antidiabetic drug sales

```
a10 <- PBS |>
  filter(ATC2 == "A10") |>
  summarise(Cost = sum(Cost) / 1e6)
```

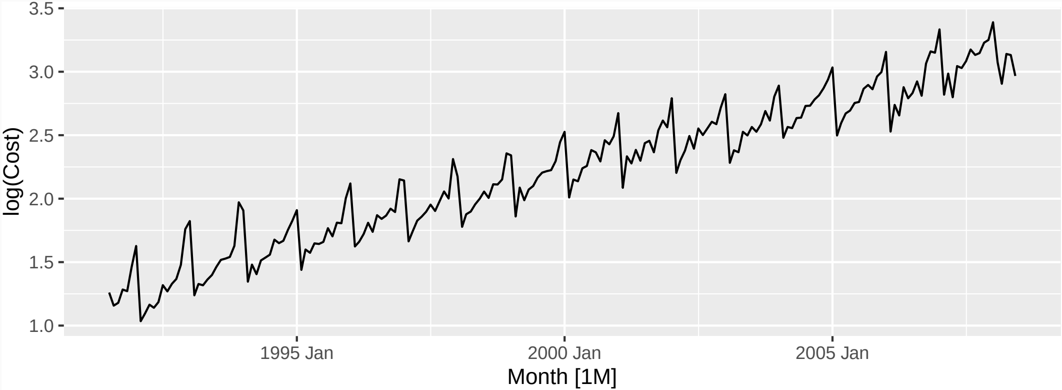
Antidiabetic drug sales

```
a10 |> autoplot(
  Cost
)
```



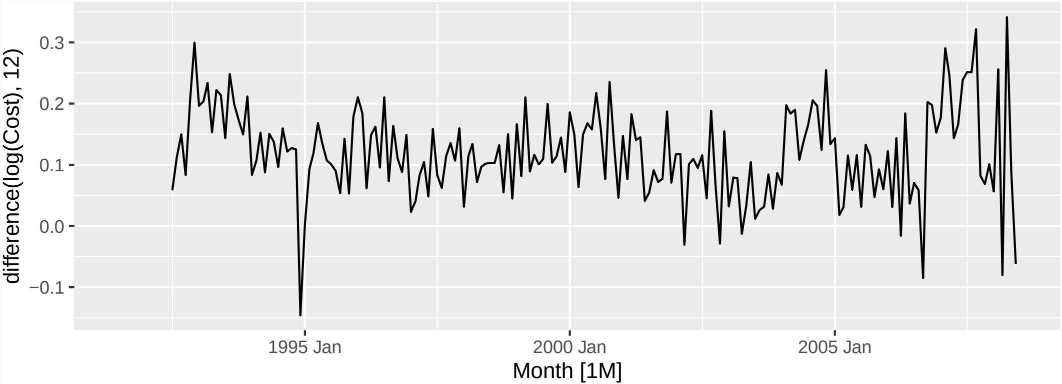
Antidiabetic drug sales

```
a10 |> autoplot(
  log(Cost)
)
```



Antidiabetic drug sales

```
a10 |> autoplot(
  log(Cost) |> difference(12)
)
```

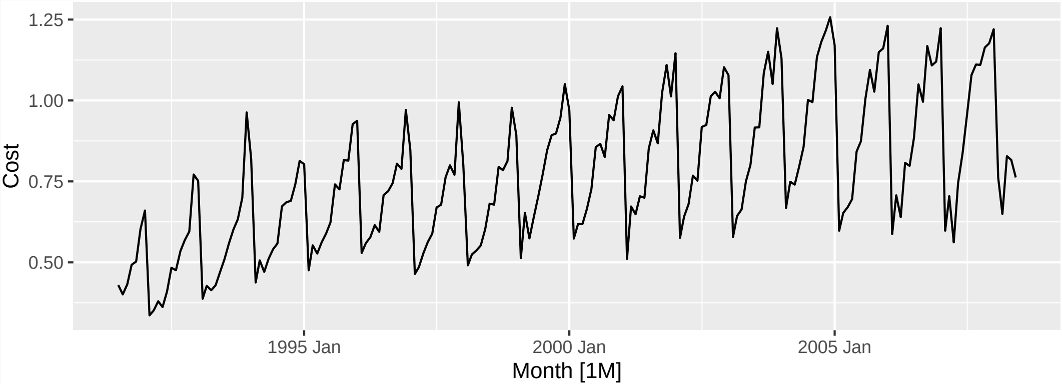


Cortecosteroid drug sales

```
h02 <- PBS |>
  filter(ATC2 == "H02") |>
  summarise(Cost = sum(Cost) / 1e6)
```

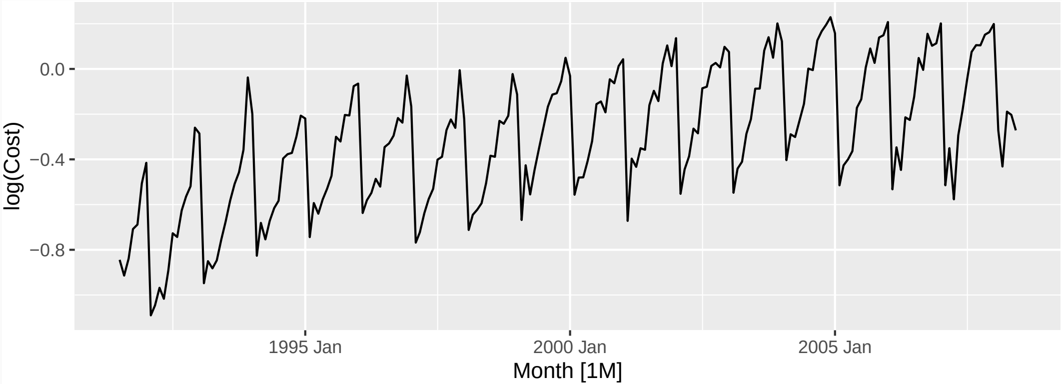
Cortecosteroid drug sales

```r
h02 |> autoplot(
  Cost
)
```



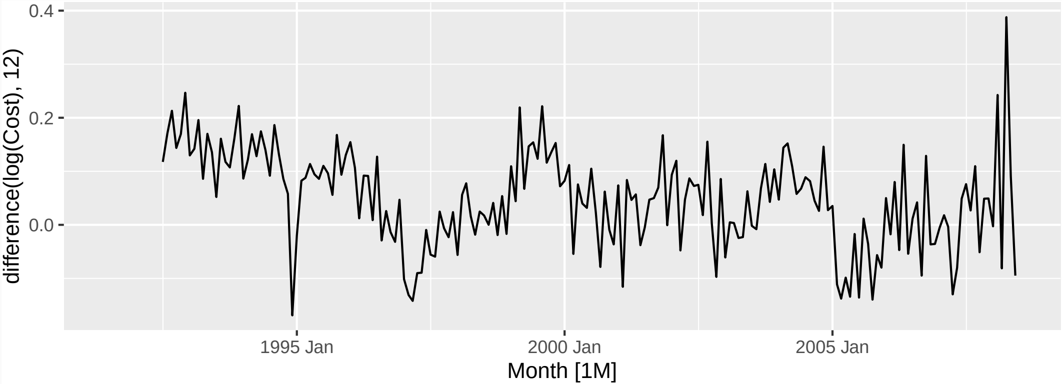
Cortecosteroid drug sales

```
h02 |> autoplot(
  log(Cost)
)
```



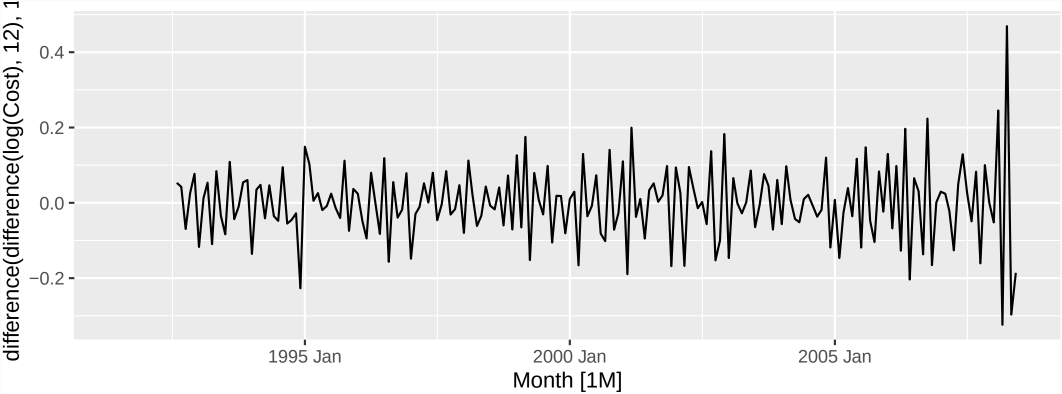
Cortecosteroid drug sales

```
h02 |> autoplot(
  log(Cost) |> difference(12)
)
```



Cortecosteroid drug sales

```r
h02 |> autoplot(
  log(Cost) |> difference(12) |> difference(1)
)
```



Cortecosteroid drug sales

- Seasonally differenced series is closer to being stationary.
- Remaining non-stationarity can be removed with further first difference.

If $y'_t = y_t - y_{t-12}$ denotes seasonally differenced series, then twice-differenced series is

$$\begin{aligned} y^*_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \\ &= y_t - y_{t-1} - y_{t-12} + y_{t-13}\,. \end{aligned}$$

Seasonal differencing

When both seasonal and first differences are applied…

Seasonal differencing

When both seasonal and first differences are applied…

- it makes no difference which is done first—the result will be the same.
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference.

Seasonal differencing

When both seasonal and first differences are applied…

- it makes no difference which is done first—the result will be the same.
- If seasonality is strong, we recommend that seasonal differencing be done first because sometimes the resulting series will be stationary and there will be no need for further first difference.

It is important that if differencing is used, the differences are interpretable.

Interpretation of differencing

- first differences are the change between **one observation and the next**;
- seasonal differences are the change between **one year to the next**.

Interpretation of differencing

- first differences are the change between **one observation and the next**;
- seasonal differences are the change between **one year to the next**.

But taking lag 3 differences for yearly data, for example, results in a model which cannot be sensibly interpreted.

Unit root tests

Statistical tests to determine the required order of differencing.

1. Augmented Dickey Fuller test: null hypothesis is that the data are non-stationary and non-seasonal.
2. Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test: null hypothesis is that the data are stationary and non-seasonal.
3. Other tests available for seasonal data.

KPSS test

```
google_2018 |>
  features(Close, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   Symbol kpss_stat kpss_pvalue
##   <chr>      <dbl>       <dbl>
## 1 GOOG       0.573      0.0252
```

KPSS test

```r
google_2018 |>
  features(Close, unitroot_kpss)
```

```
## # A tibble: 1 x 3
##   Symbol kpss_stat kpss_pvalue
##   <chr>      <dbl>       <dbl>
## 1 GOOG       0.573      0.0252
```

```r
google_2018 |>
  features(Close, unitroot_ndiffs)
```

```
## # A tibble: 1 x 2
##   Symbol ndiffs
##   <chr>   <int>
## 1 GOOG        1
```

Automatically selecting differences

STL decomposition: $y_t = T_t + S_t + R_t$

Seasonal strength $F_s = \max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t + R_t)}\right)$

If $F_s > 0.64$, do one seasonal difference.

```
h02 |>
  mutate(log_sales = log(Cost)) |>
  features(log_sales, list(unitroot_nsdiffs, feat_stl)) |>
  select(1:3)
```

```
## # A tibble: 1 x 3
##   nsdiffs trend_strength seasonal_strength_year
##     <int>          <dbl>                  <dbl>
## 1       1          0.957                  0.955
```

Automatically selecting differences

```r
h02 |>
  mutate(log_sales = log(Cost)) |>
  features(log_sales, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       1
```

```r
h02 |>
  mutate(d_log_sales = difference(log(Cost), 12)) |>
  features(d_log_sales, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```