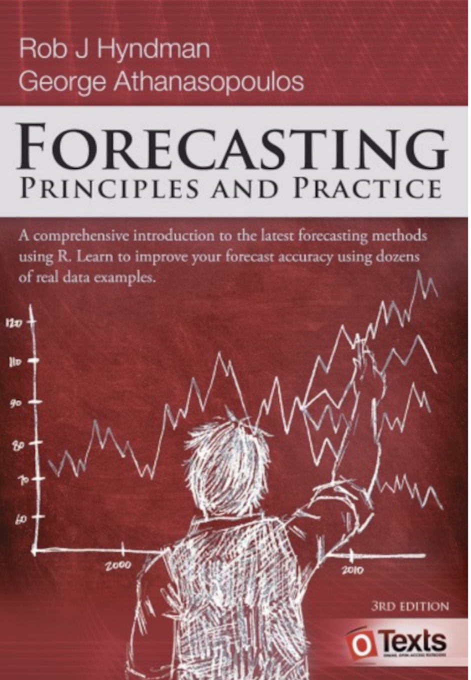


9. ARIMA models

9.1 Unit root tests

OTexts.org/fpp3/

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.

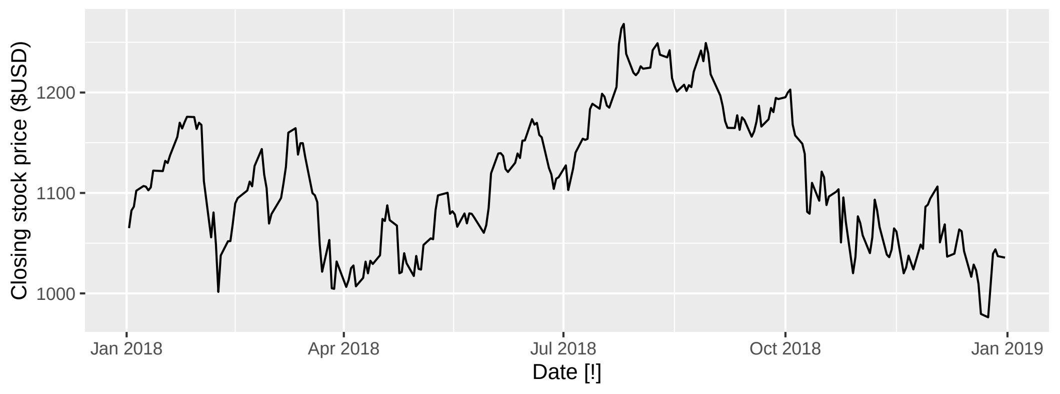
Unit root tests

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. H_0: non-stationary
2. Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test: null hypothesis is that the data are stationary and non-seasonal. H_0: stationary
3. Other tests available for seasonal data.

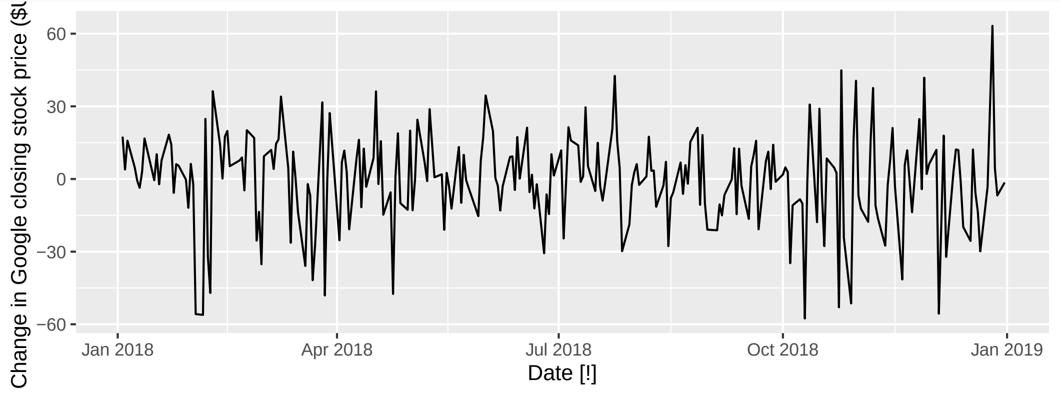
Example: Google stock price

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



Example: Google stock price

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



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)
```

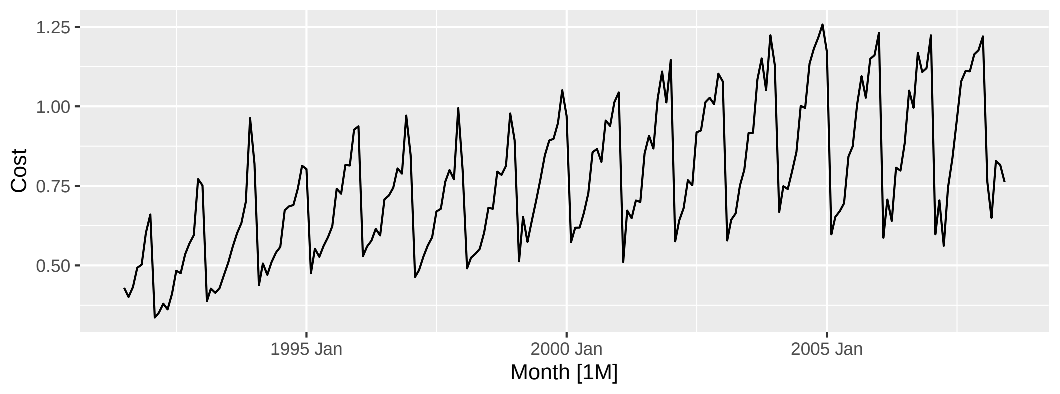
```
## # A tibble: 1 x 3
##   Symbol kpss_stat kpss_pvalue
##   <chr>      <dbl>       <dbl>
## 1 GOOG       0.573      0.0252
```

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

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

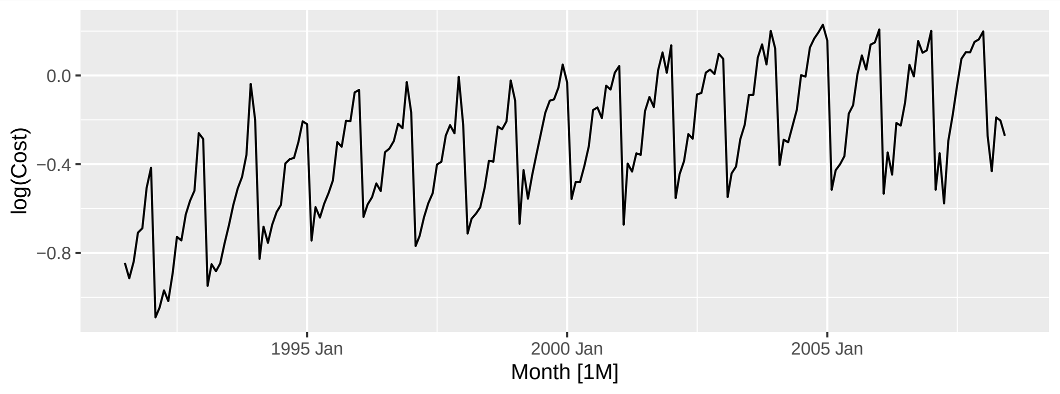
Corticosteroid drug sales

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



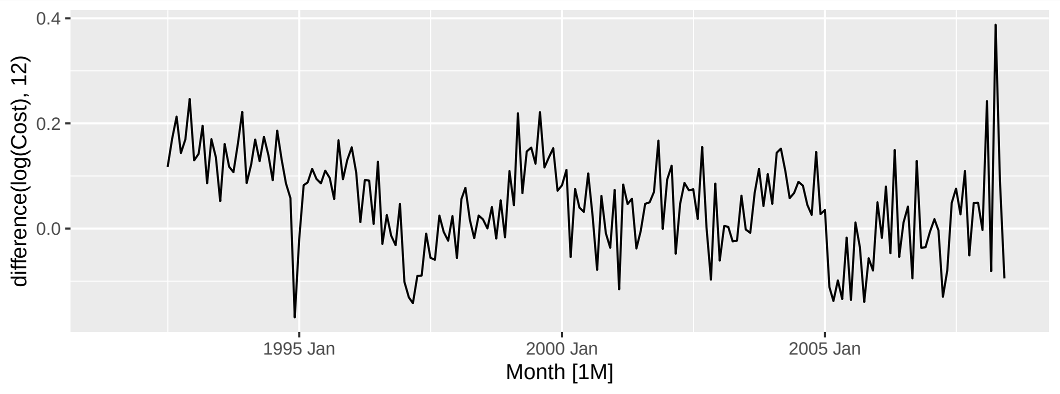
Corticosteroid drug sales

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



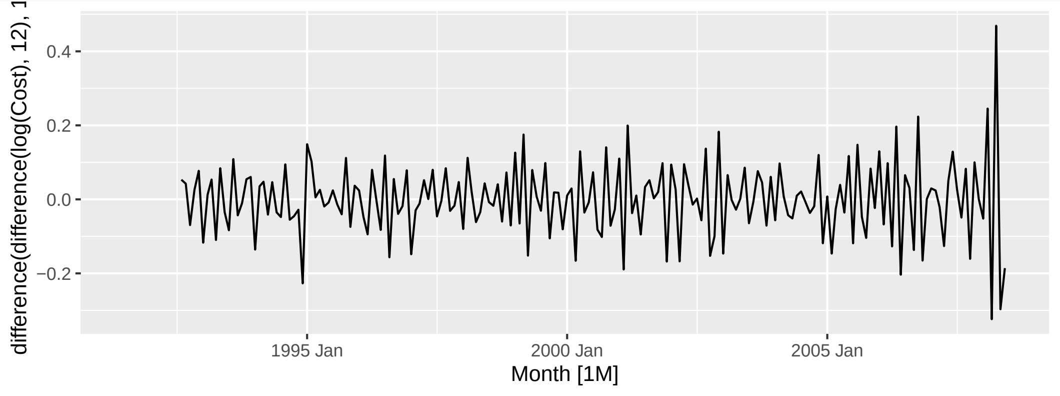
Corticosteroid drug sales

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



Corticosteroid drug sales

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

```
## # A tibble: 1 x 9
##   trend_~1 seaso~2 seaso~3 seaso~4 spikin~5 linea~6 curva~7 stl_e~8 stl_e~9
##      <dbl>   <dbl>   <dbl>   <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1    0.957   0.955       6       8 3.78e-10    2.92  -0.831  -0.237   0.220
## # ... with abbreviated variable names 1: trend_strength,
## #   2: seasonal_strength_year, 3: seasonal_peak_year,
## #   4: seasonal_trough_year, 5: spikiness, 6: linearity, 7: curvature,
## #   8: stl_e_acf1, 9: stl_e_acf10
```

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
```