

Examen 1

Luis Eduardo Jiménez del Muro

Librerías

```
library(tidyverse)
library(fpp3)
```

Graficar datos

```
data <- tidyquant::tq_get(
  x = "RIVN",
  from = today() - 365,
  to = today() - 1
) |>
  select(symbol, adjusted) |>
  mutate(
    t = seq_len(n())
  ) |>
  as_tsibble(index = t, key = symbol) |>
  relocate(symbol, t)
```

Registered S3 method overwritten by 'quantmod':

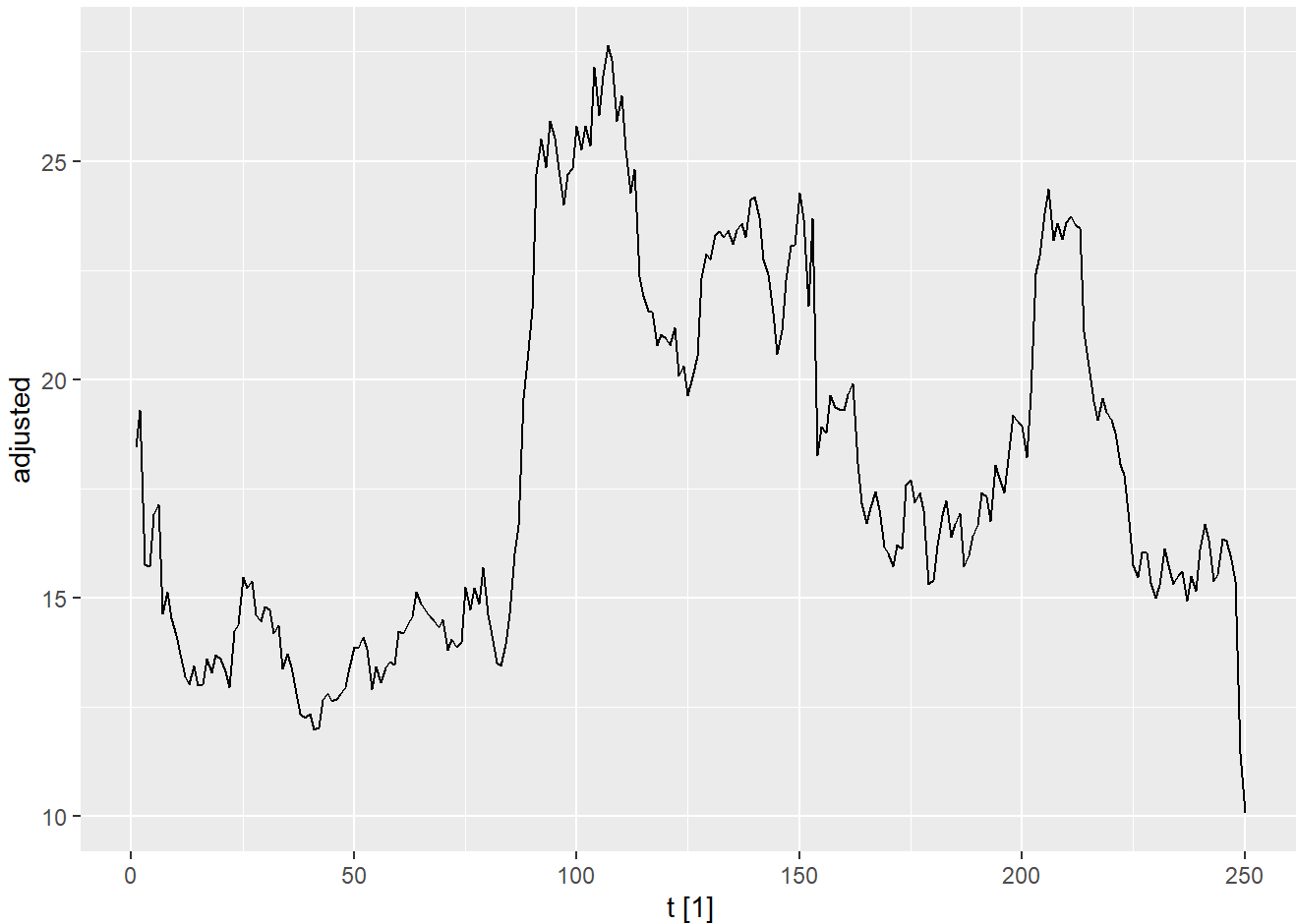
```
method      from
as.zoo.data.frame zoo
```

```
data
```

```
# A tsibble: 250 x 3 [1]
# Key:      symbol [1]
  symbol    t adjusted
  <chr>  <int>   <dbl>
1 RIVN      1    18.5
2 RIVN      2    19.3
3 RIVN      3    15.8
4 RIVN      4    15.7
5 RIVN      5    16.9
6 RIVN      6    17.1
7 RIVN      7    14.6
8 RIVN      8    15.1
9 RIVN      9    14.5
10 RIVN     10    14.2
# i 240 more rows
```

```
autoplot(data)
```

Plot variable not specified, automatically selected ``vars = adjusted``



Obtener Lambda

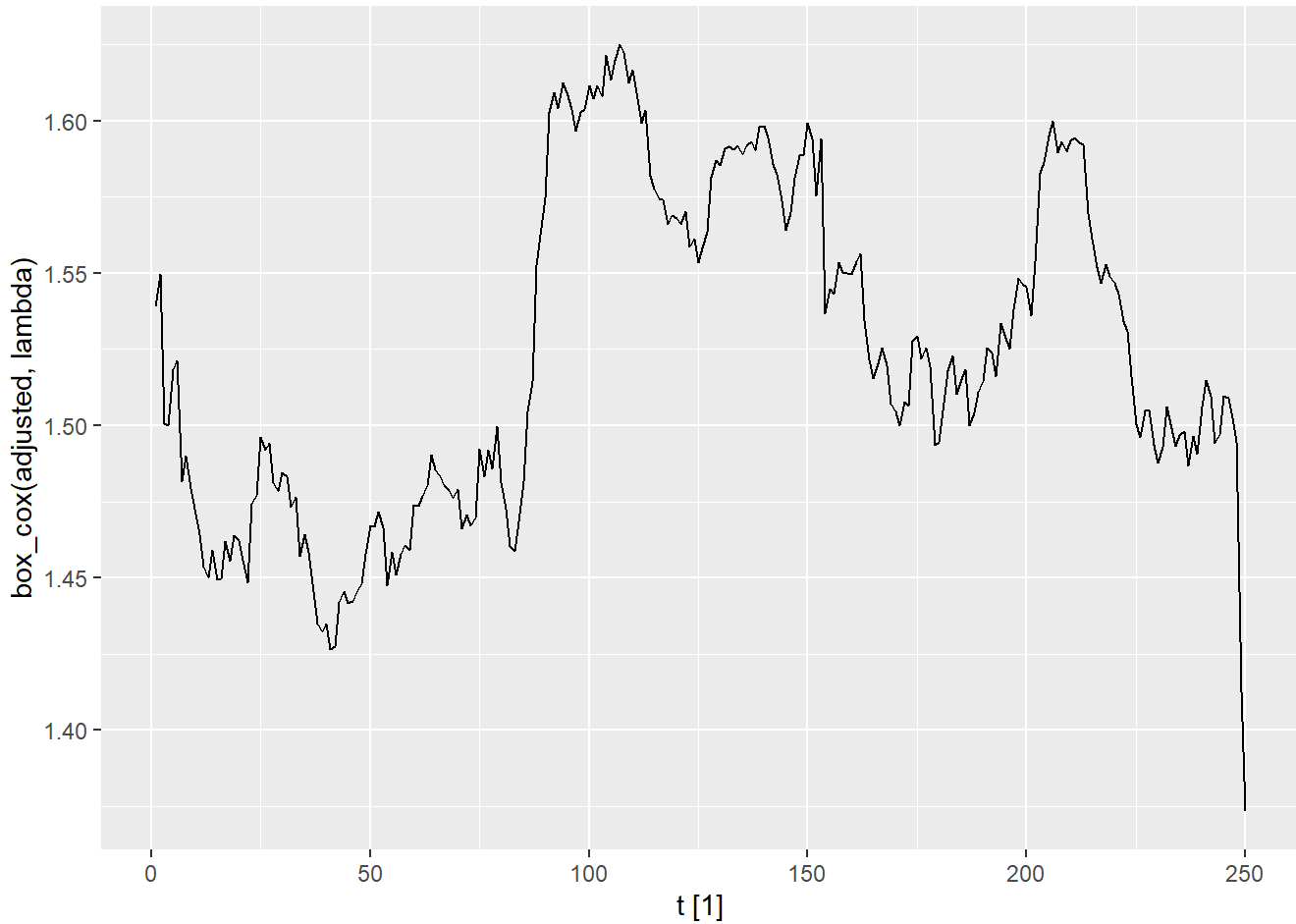
```
data |>  
  features(adjusted, features=guerrero)
```

```
# A tibble: 1 × 2  
  symbol lambda_guerrero  
  <chr>      <dbl>  
1 RIVN      -0.497
```

Grafica con box-cox

```
lambda <- -0.4972675
```

```
data |>  
  autoplot(box_cox(adjusted, lambda))
```

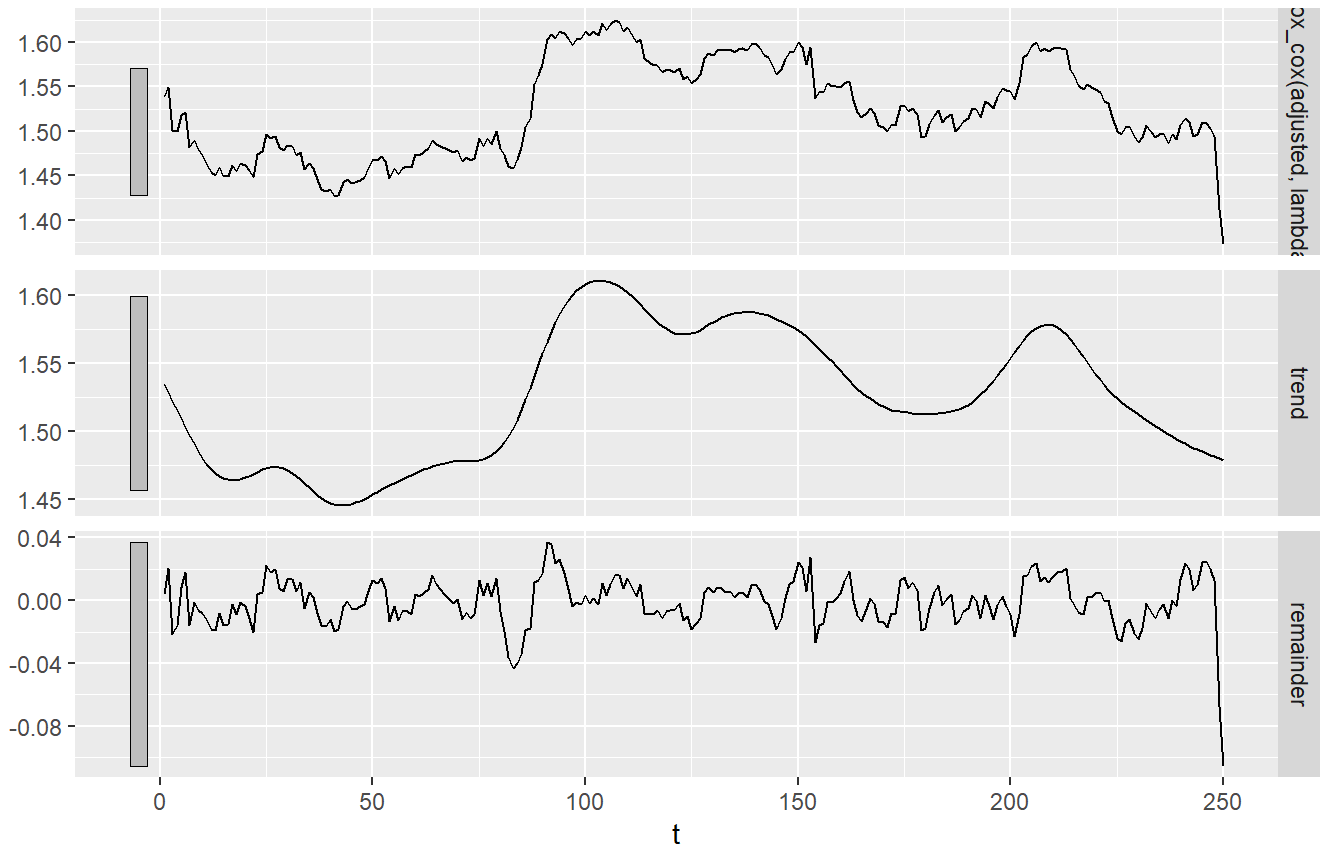


Descomponer para ver si vale la pena

```
data |>  
  model(  
    stl = STL(box_cox(adjusted, lambda), robust = TRUE)  
  ) |>  
  components() |>  
  autoplot()
```

STL decomposition

`box_cox(adjusted, lambda)` = trend + remainder



Seleccionar datos de entrenamiento

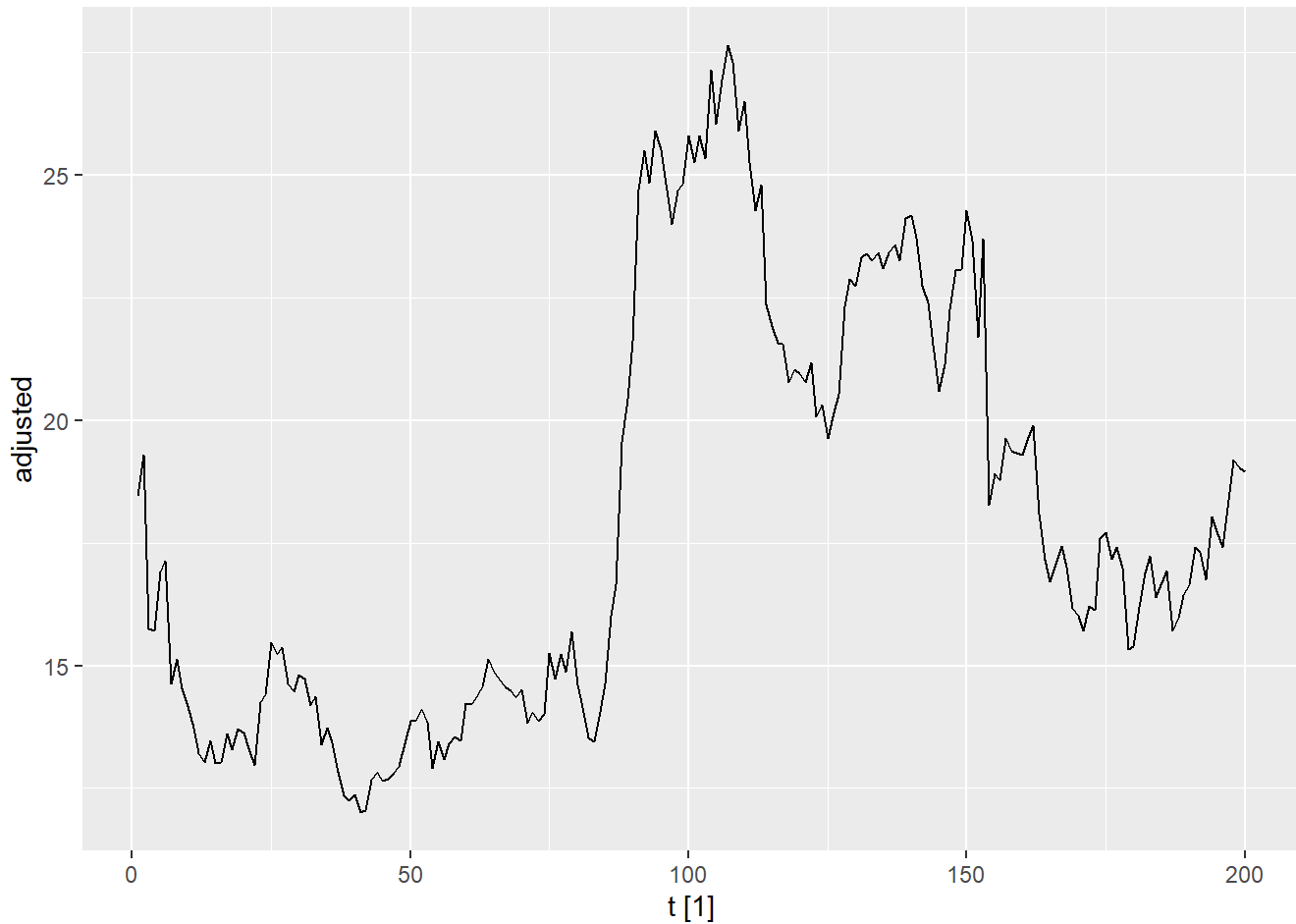
```
data_train <- data |>
  filter_index(. ~ "200")
```

```
data_train
```

```
# A tsibble: 200 x 3 [1]
# Key:      symbol [1]
#   symbol    t adjusted
#   <chr> <int>   <dbl>
1 RIVN      1    18.5
2 RIVN      2    19.3
3 RIVN      3    15.8
4 RIVN      4    15.7
5 RIVN      5    16.9
6 RIVN      6    17.1
7 RIVN      7    14.6
8 RIVN      8    15.1
9 RIVN      9    14.5
10 RIVN     10    14.2
# i 190 more rows
```

```
data_train |>  
  autoplot()
```

Plot variable not specified, automatically selected ``.vars = adjusted``



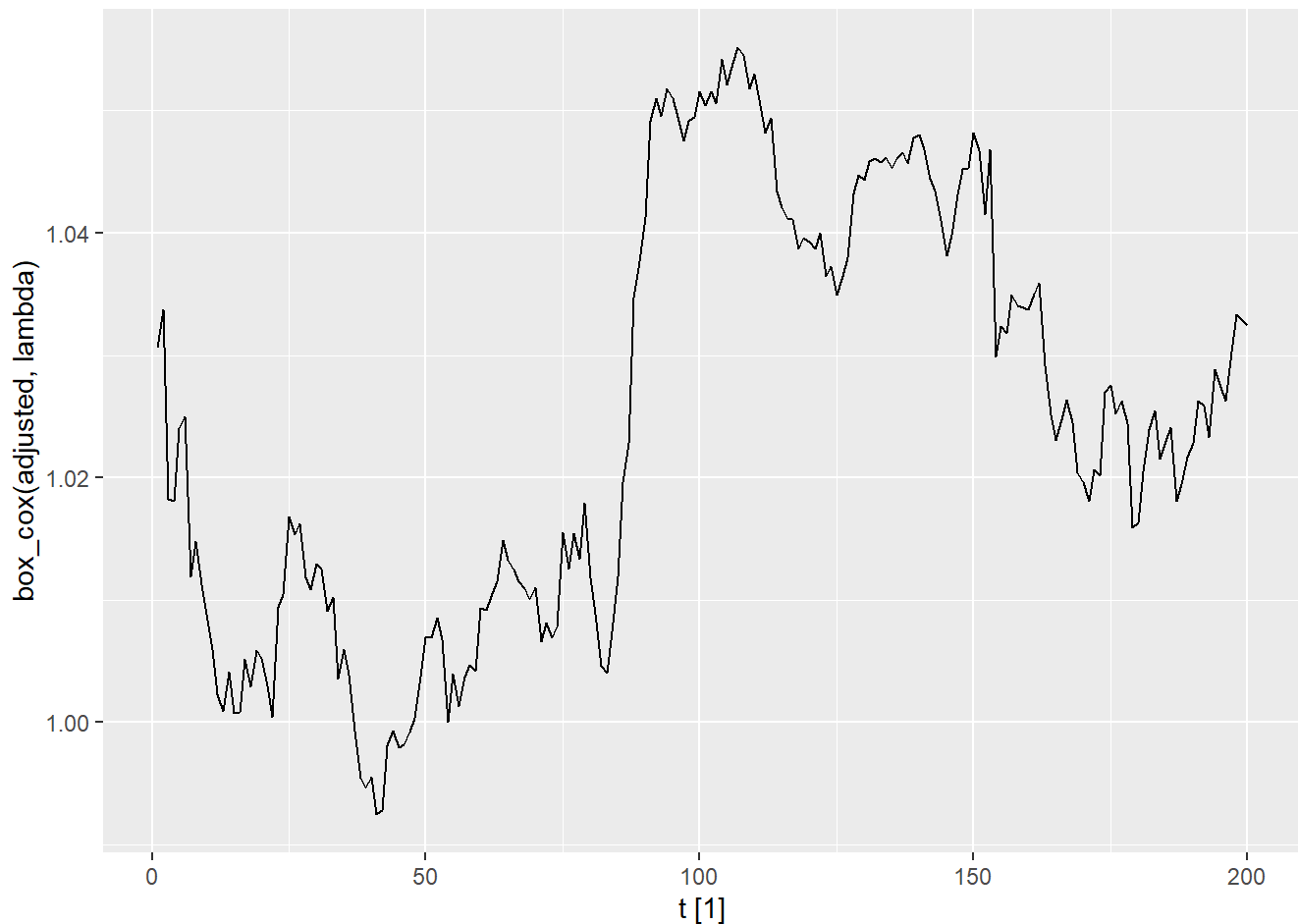
Datos de entrenamiento con box-cox

```
lambda <- data_train |>  
  features(adjusted, features = guerrero) |>  
  pull(lambda_guerrero)
```

```
lambda
```

```
[1] -0.8999268
```

```
data_train |>  
  autoplot(box_cox(adjusted, lambda))
```



Hacer los modelos para comparar cual vale mas la pena

```
data_fit <- data_train |>
  model(
    media = MEAN(box_cox(adjusted, lambda)),
    naive = NAIVE(box_cox(adjusted, lambda)),
    drift = RW(box_cox(adjusted, lambda) ~ drift()),
    # No se usa snaive porque no tiene estacionalidad
  )

data_fit
```

```
# A mable: 1 x 4
# Key:      symbol [1]
  symbol  media  naive      drift
  <chr>  <model> <model>    <model>
1 RIVN   <MEAN> <NAIVE> <RW w/ drift>
```

```
accuracy(data_fit)
```

```
# A tibble: 3 × 11
  symbol .model .type      ME  RMSE  MAE    MPE  MAPE  MASE  RMSSE  ACF1
  <chr>  <chr>  <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
1 RIVN  media  Training 0.892    4.37  3.61 -0.254 19.5   5.98  4.92  0.978
2 RIVN  naive  Training 0.00251  0.889 0.604 -0.100  3.34   1     1    -0.0811
3 RIVN  drift  Training 0.0000728 0.889 0.604 -0.113  3.34   1.00  1.00 -0.0811
```

El que tiene menos error es naive con mae

Preparar datos para el pronóstico

```
data_fit <- data_train |>
  model(
    naive = NAIVE(box_cox(adjusted, lambda)),
  )

data_fit
```

```
# A mable: 1 × 2
# Key:      symbol [1]
  symbol    naive
  <chr>    <model>
1 RIVN    <NAIVE>
```

P-value

```
data_fit |>
  augment() |>
  features(.innov, ljung_box, lag = 10, dof = 0)
```

```
# A tibble: 1 × 4
  symbol .model lb_stat lb_pvalue
  <chr>  <chr>    <dbl>    <dbl>
1 RIVN  naive      10.8      0.376
```

Como el p-values es mayor a α ; $p_{value} = 0.05$, **NO RECHAZAMOS** H_0 : "Los residuos son ruido blanco"

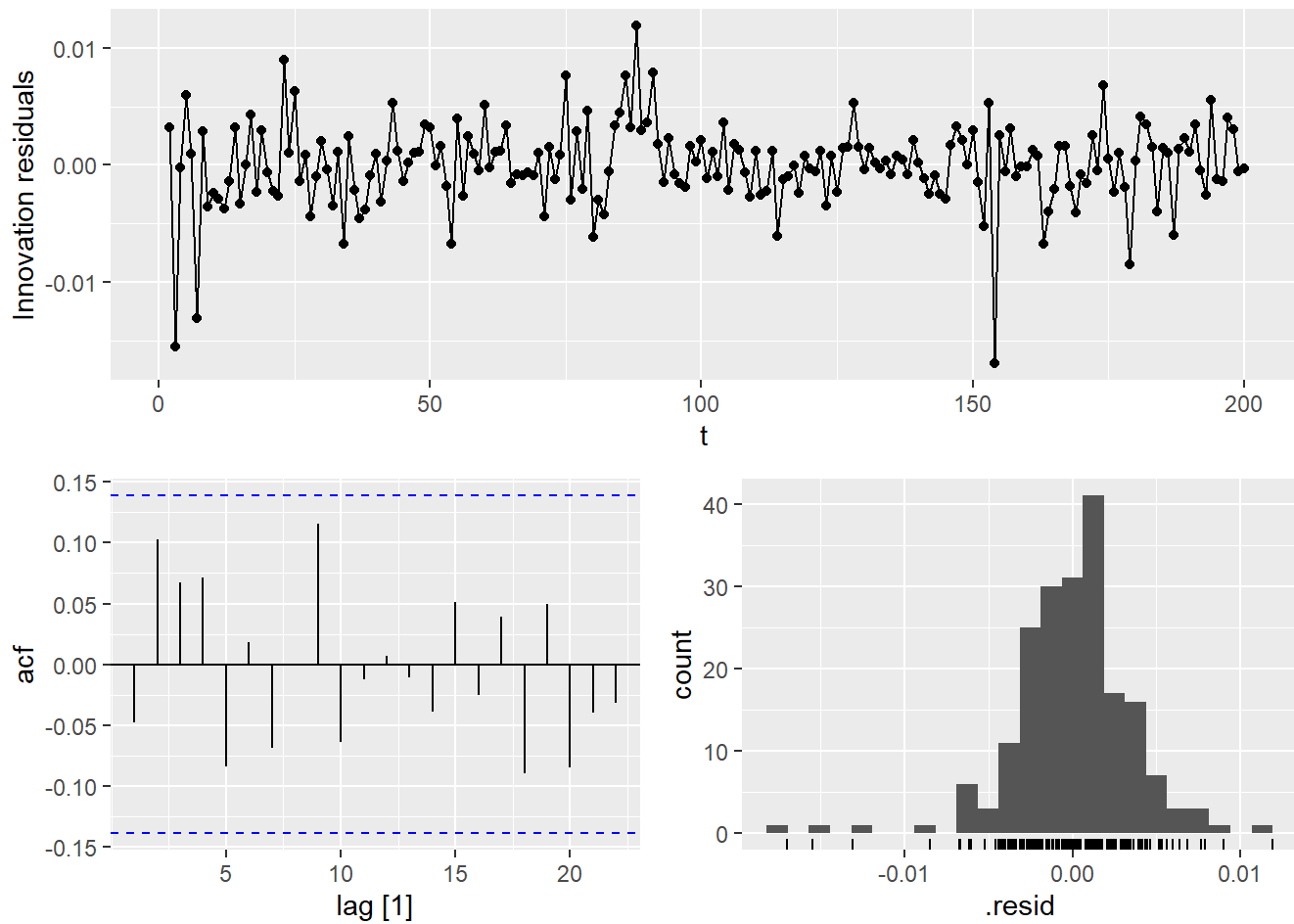
Gráficas de residuos

```
data_fit |>
  select(naive) |>
  gg_tsresiduals()
```

Warning: Removed 1 row containing missing values (`geom_line()`).

Warning: Removed 1 rows containing missing values (`geom_point()`).

Warning: Removed 1 rows containing non-finite values (`stat_bin()`).



Pronóstico

```
data_fc <- data_fit |>
  forecast(h = 50)

data_fc |>
  autoplot(data) +
  facet_wrap(~ .model) +
  theme(legend.position = "none")
```