Pronóstico

Forecasts

```r
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.2     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.1.0
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
```

```r
library(fpp3)
```

```
Registered S3 method overwritten by 'tsibble':
  method               from
  as_tibble.grouped_df dplyr
── Attaching packages ──────────────────────────────────────────── fpp3 1.0.1 ──
✔ tsibble     1.1.6     ✔ feasts      0.4.1
✔ tsibbledata 0.4.1     ✔ fable       0.4.1
── Conflicts ───────────────────────────────────────────────── fpp3_conflicts ──
✖ lubridate::date()    masks base::date()
✖ dplyr::filter()      masks stats::filter()
✖ tsibble::intersect() masks base::intersect()
✖ tsibble::interval()  masks lubridate::interval()
✖ dplyr::lag()         masks stats::lag()
✖ tsibble::setdiff()   masks base::setdiff()
✖ tsibble::union()     masks base::union()
```

Let’s review this time series for Australian production. You can see that the gas production doesn’t have constant variance and needs a mathematical transformation to stabilize it.

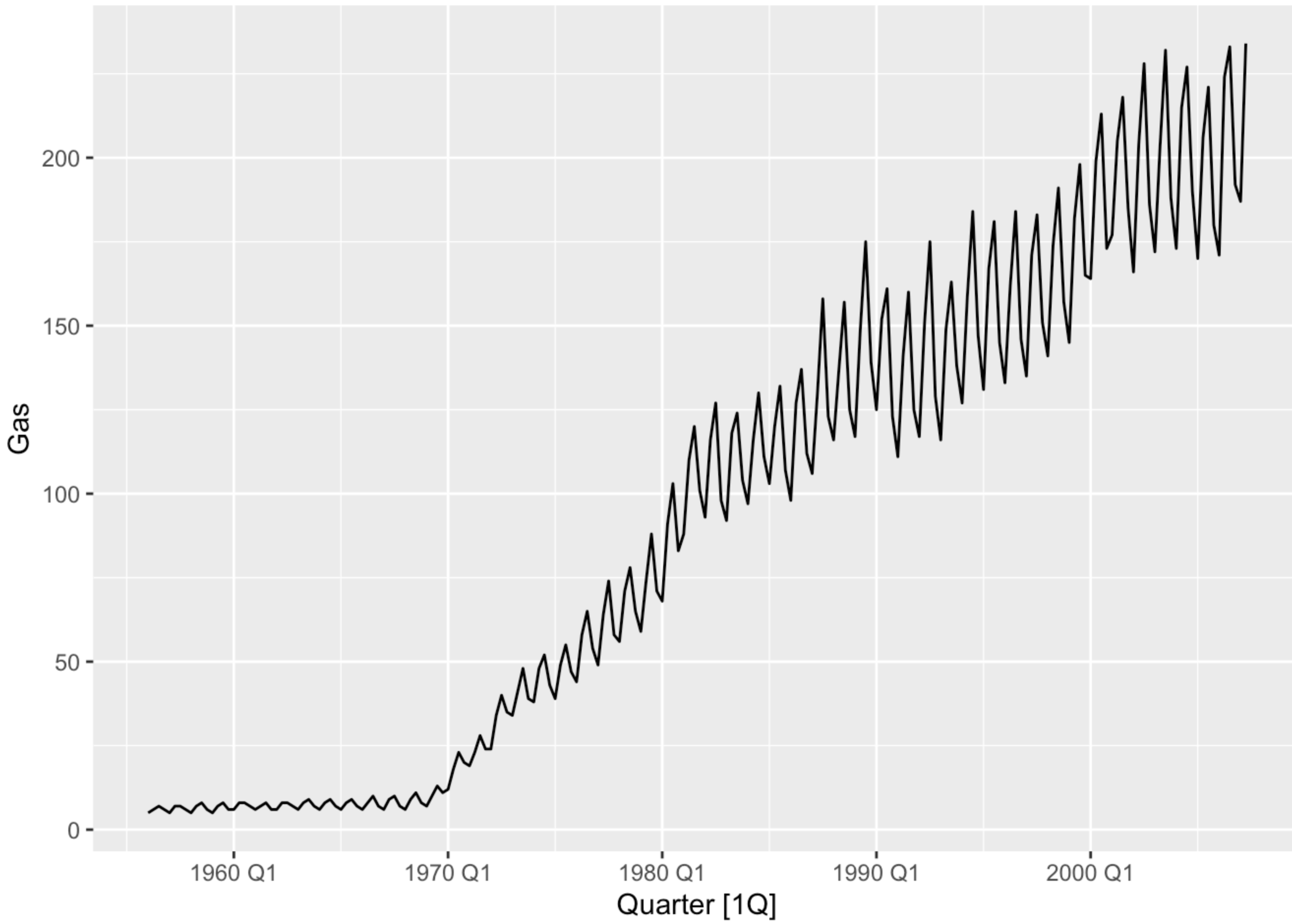
Assuming we’ll make a forecast for 3 years of Aus production, let’s separate our data in train and test.

```r
gas_train <- aus_production |>
  filter_index(. ~ "2007 Q2")
```

Now let’s visualize the time series:

```r
gas_train |>
```

```
  autoplot(Gas)
```



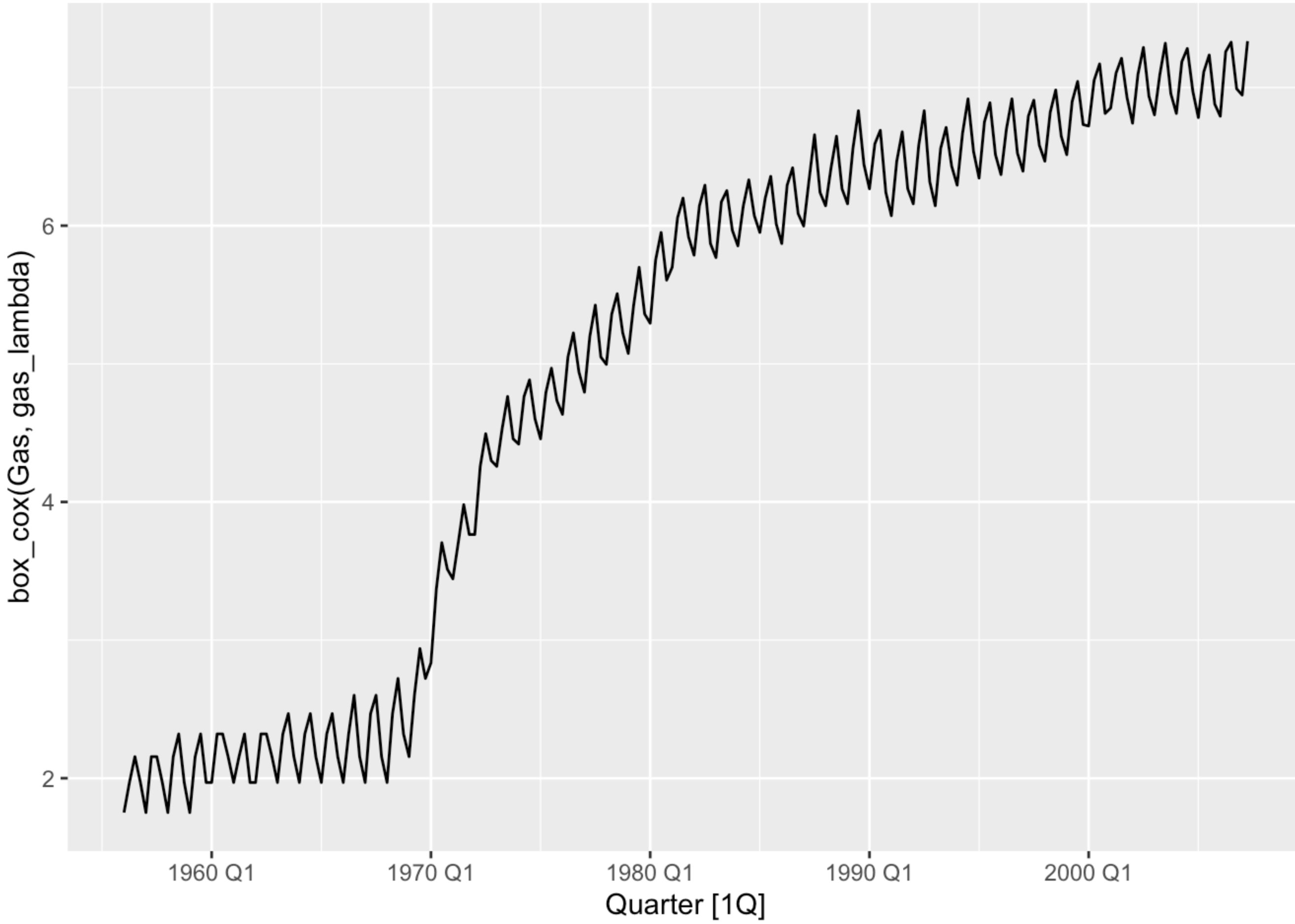
Lets use box-cox with Guerrero feature and see the changes.

```
gas_lambda <- gas_train |>
  features(Gas, features= guerrero) |>
  pull()

gas_lambda
```

```
[1] 0.1037006
```

```
gas_train |>
  autoplot(box_cox(Gas, gas_lambda))
```



¿Which forecast method is the most appropriate for this time series?

Note that all the mathematical transformations have to be done inside the fitting function, not before.

```
#tabla con los modelos ajustados
gas_fit <- gas_train |>
  model(
    drift = RW(box_cox(Gas, gas_lambda) ~ drift()),
    snaive = SNAIVE(box_cox(Gas, gas_lambda)),
    media = MEAN(box_cox(Gas, gas_lambda))
  )

gas_fit
```

```
# A mable: 1 x 3
          drift   snaive   media
        <model>  <model> <model>
1 <RW w/ drift> <SNAIVE>  <MEAN>
```

Lets make a diagnosis of the models residuals. The function `augment()`, it allows us to obtain the residuals and other adjusted values of the models.

```
gas_aug <- gas_fit |>
  augment()
```

```r
gas_aug
```

```
# A tsibble: 618 x 6 [1Q]
# Key:       .model [3]
   .model Quarter   Gas .fitted .resid  .innov
   <chr>    <qtr> <dbl>   <dbl>  <dbl>   <dbl>
 1 drift  1956 Q1  5      NA     NA     NA
 2 drift  1956 Q2  6.00    5.12  0.884  0.190
 3 drift  1956 Q3  7.00    6.14  0.863  0.160
 4 drift  1956 Q4  6.00    7.16 -1.16  -0.214
 5 drift  1957 Q1  5       6.14 -1.14  -0.245
 6 drift  1957 Q2  7.00    5.12  1.88   0.377
 7 drift  1957 Q3  7.00    7.16 -0.157 -0.0272
 8 drift  1957 Q4  6.00    7.16 -1.16  -0.214
 9 drift  1958 Q1  5       6.14 -1.14  -0.245
10 drift  1958 Q2  7.00    5.12  1.88   0.377
# ℹ 608 more rows
```

Residual Diagnosis

A good forecast model will produce residuals with the following characteristics:

1. **Residuals are not autocorrelated**: if correlations are detected between residuals, there's still useful information yet to be modeled.
2. **The residuals average is zero:** If the average is different than zero, then the forecast is biased.

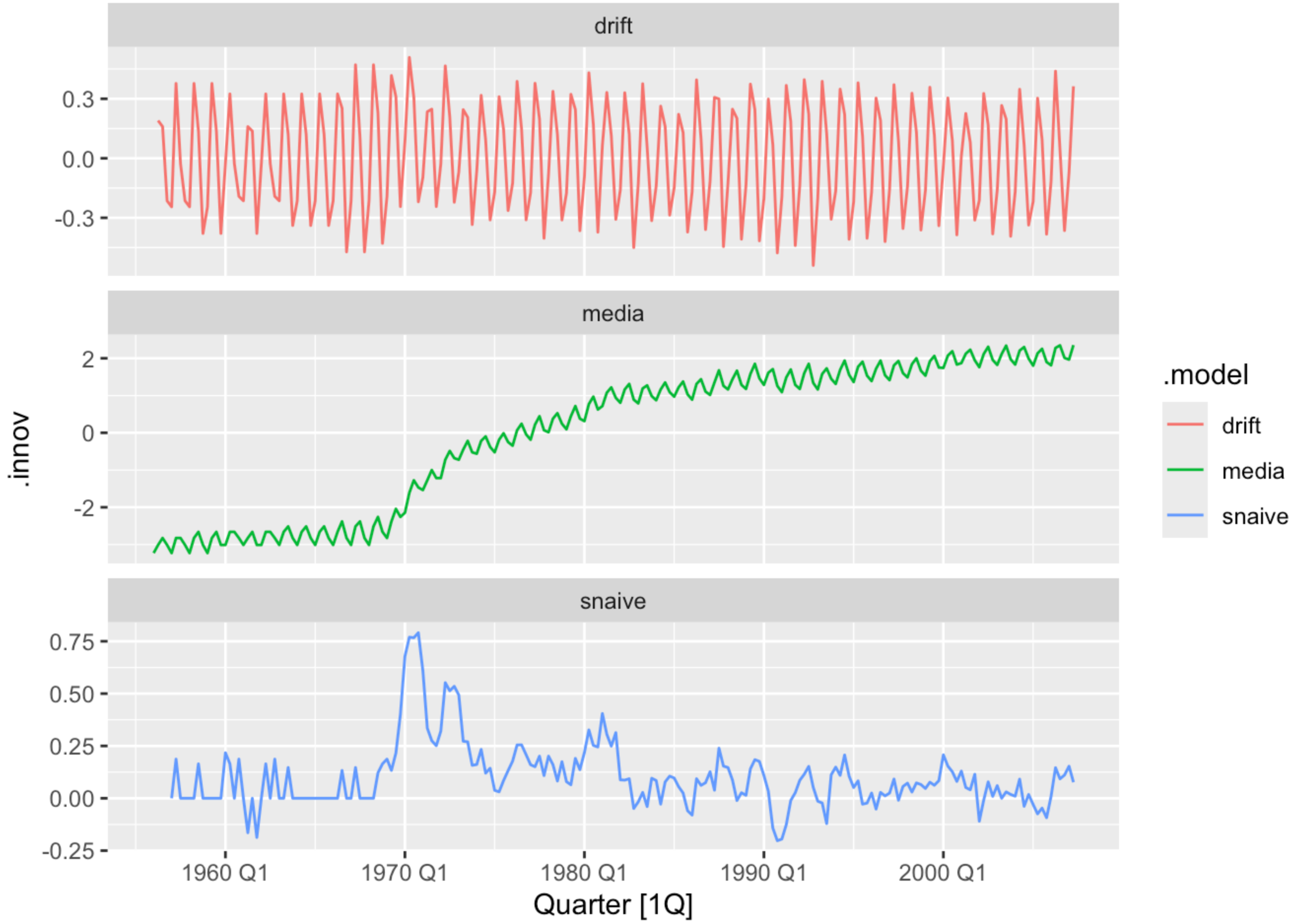
Additional to this, the following characteristics are useful, not determinant:

3. The residuals have a constant variance
4. The residuals are *normally* distributed

Gráfica de los residuos de los modelos:

```r
gas_aug |>
  autoplot(.innov) +
  facet_wrap(~.model, ncol=1, scales = "free_y")
```

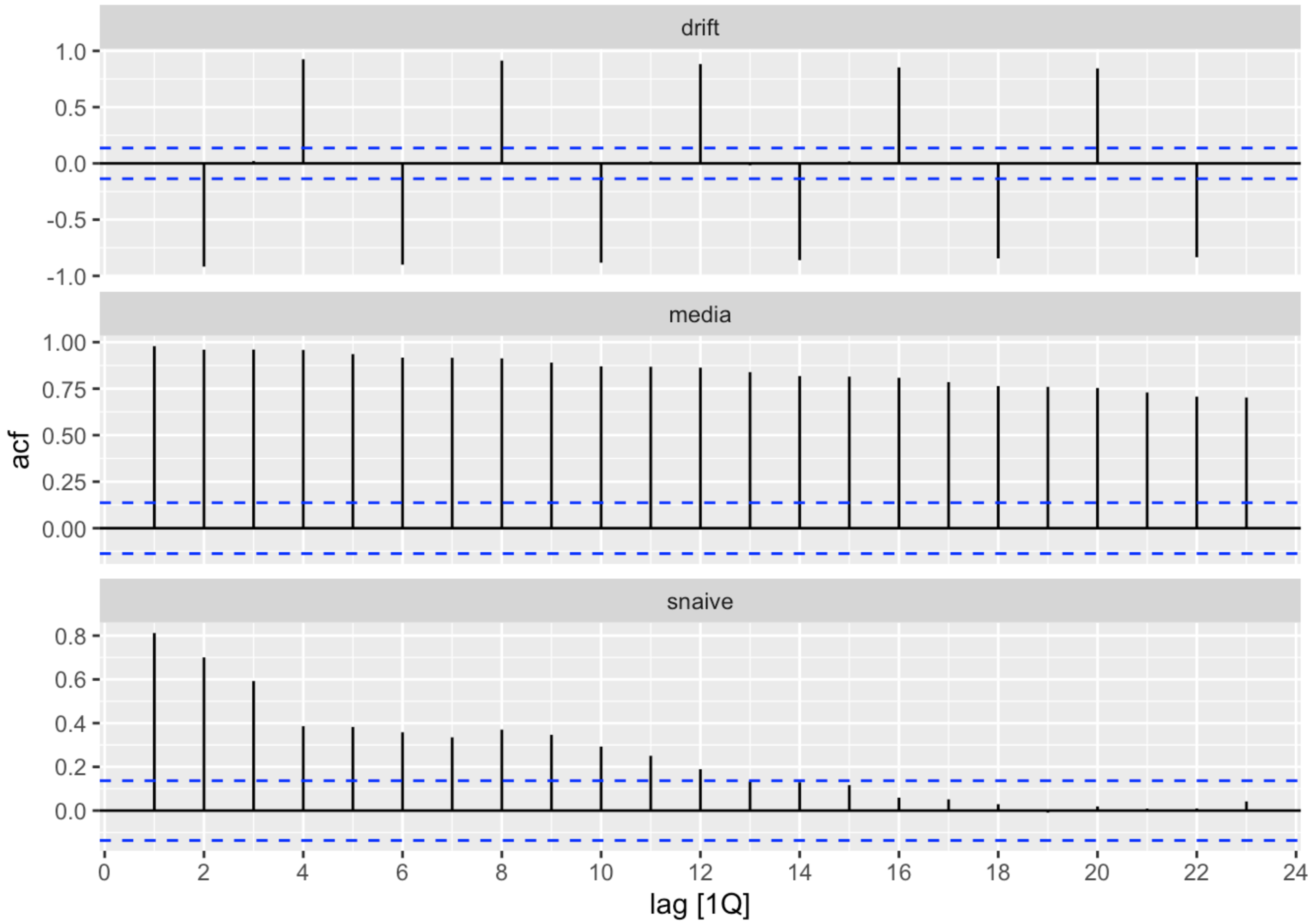
```
Warning: Removed 5 rows containing missing values or values outside the scale range
(`geom_line()`).
```



Gráfica edel ACF del residuo de los modelos:

```
gas_aug <- gas_fit |>
  augment()

gas_aug |>
  ACF(.innov) |>
  autoplot() +
  facet_wrap(~.model, ncol = 1, scale = "free_y")
```

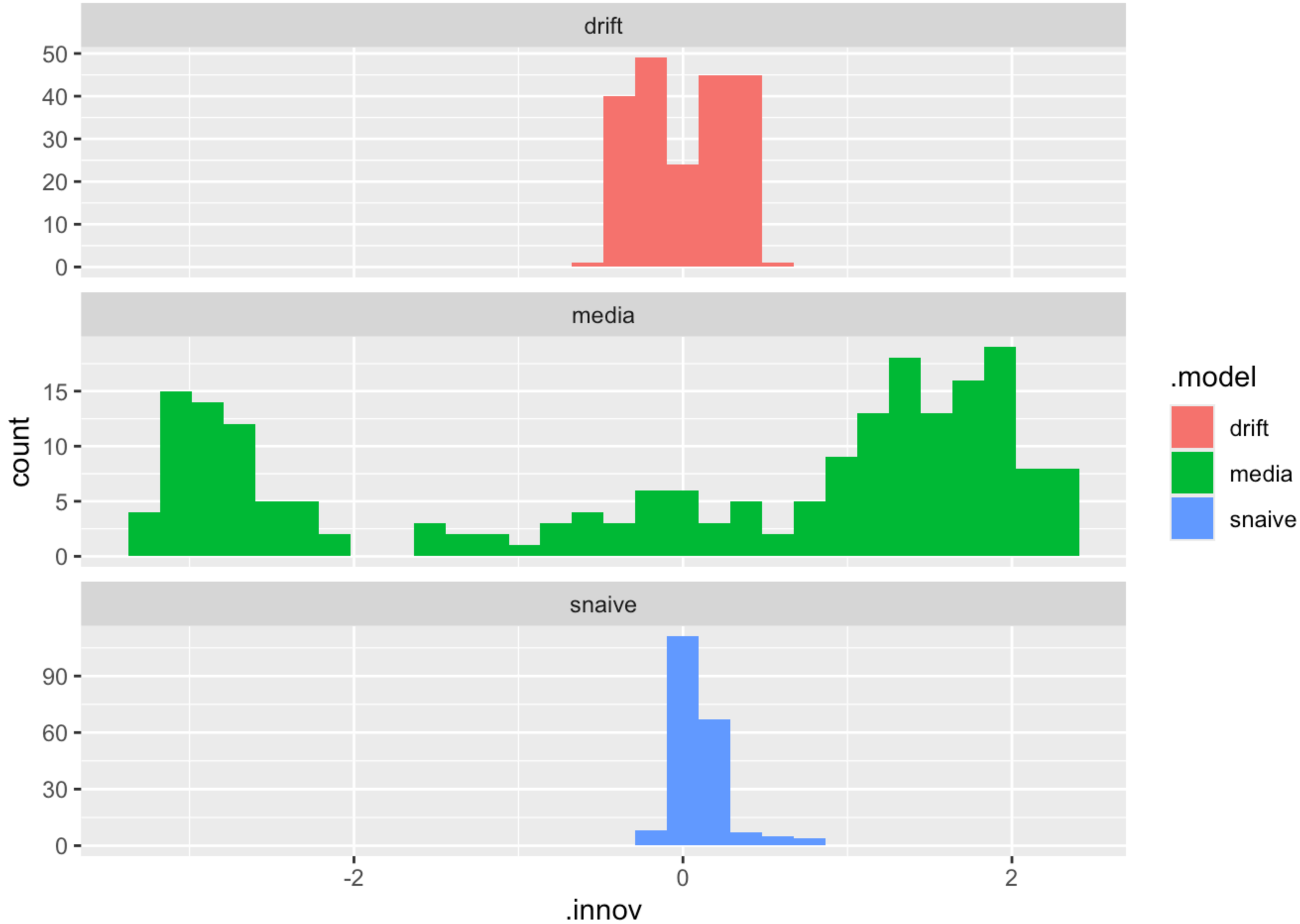


Gráfica del histograma de los residuos

```
gas_aug |>
  ggplot(aes(x = .innov, fill = .model)) +
  geom_histogram() +
  facet_wrap(~.model, ncol=1, scales = "free_y")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 5 rows containing non-finite outside the scale range
(`stat_bin()`).
```



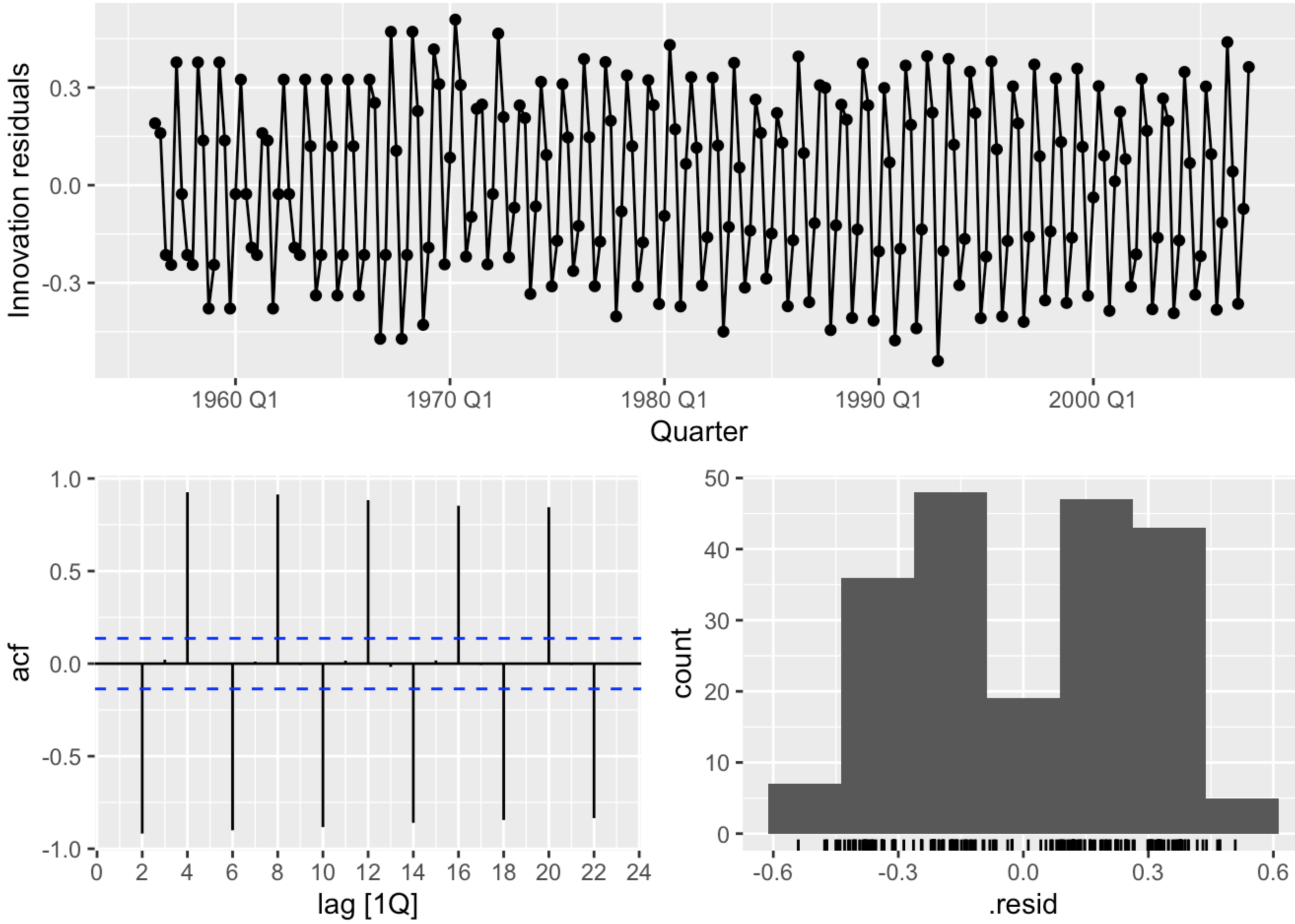
Otra opción

```
gas_fit |>
  select(drift) |>
  gg_tsresiduals()
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_line()`).

Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_point()`).

Warning: Removed 1 row containing non-finite outside the scale range
(`stat_bin()`).
```




```r
aus_prod_recent <- aus_production |>
  filter_index("1999 Q1" ~ .)
```

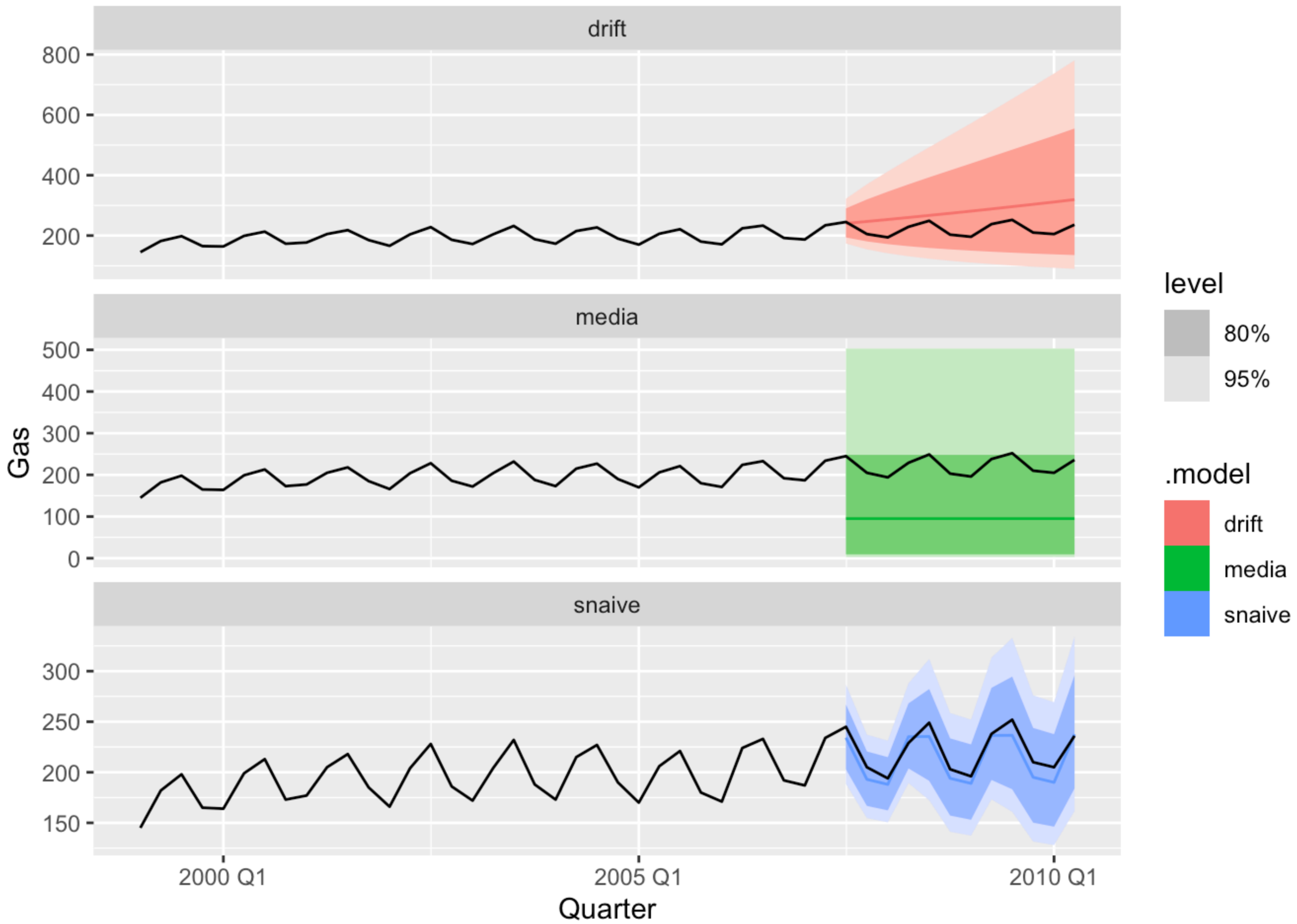
```r
gas_fc <- gas_fit |>
  forecast(h="3 years") # 12 Q es 3 años, se puede escribir solo 12

gas_fc
```

```
# A fable: 36 x 4 [1Q]
# Key:     .model [3]
   .model Quarter              Gas .mean
   <chr>    <qtr>           <dist> <dbl>
 1 drift  2007 Q3 t(N(7.4, 0.077))  240.
 2 drift  2007 Q4  t(N(7.4, 0.15))  247.
 3 drift  2008 Q1  t(N(7.4, 0.23))  253.
 4 drift  2008 Q2  t(N(7.4, 0.31))  260.
 5 drift  2008 Q3  t(N(7.5, 0.39))  267.
 6 drift  2008 Q4  t(N(7.5, 0.47))  274.
 7 drift  2009 Q1  t(N(7.5, 0.55))  281.
 8 drift  2009 Q2  t(N(7.6, 0.64))  288.
 9 drift  2009 Q3  t(N(7.6, 0.72))  296.
```

```
10 drift  2009 Q4   t(N(7.6, 0.8))  303.
# ℹ 26 more rows
```

```
gas_fc |>
  autoplot(aus_prod_recent) +
  facet_wrap(vars(.model), scale = "free_y", ncol = 1)
```



El modelo seasonal naïve copia y pega el último año, su pronóstico es el igual al último año y no capturó la tendencia de la serie.

$$MAE = mean(|e_t|)$$

Estas medidas no pueden usarse para comparar pronósticos entre distintas series por la escala de los datos. Sin embargo, se puede usar el error porcentual como el MAPE, el error absoluto medio porcentual:

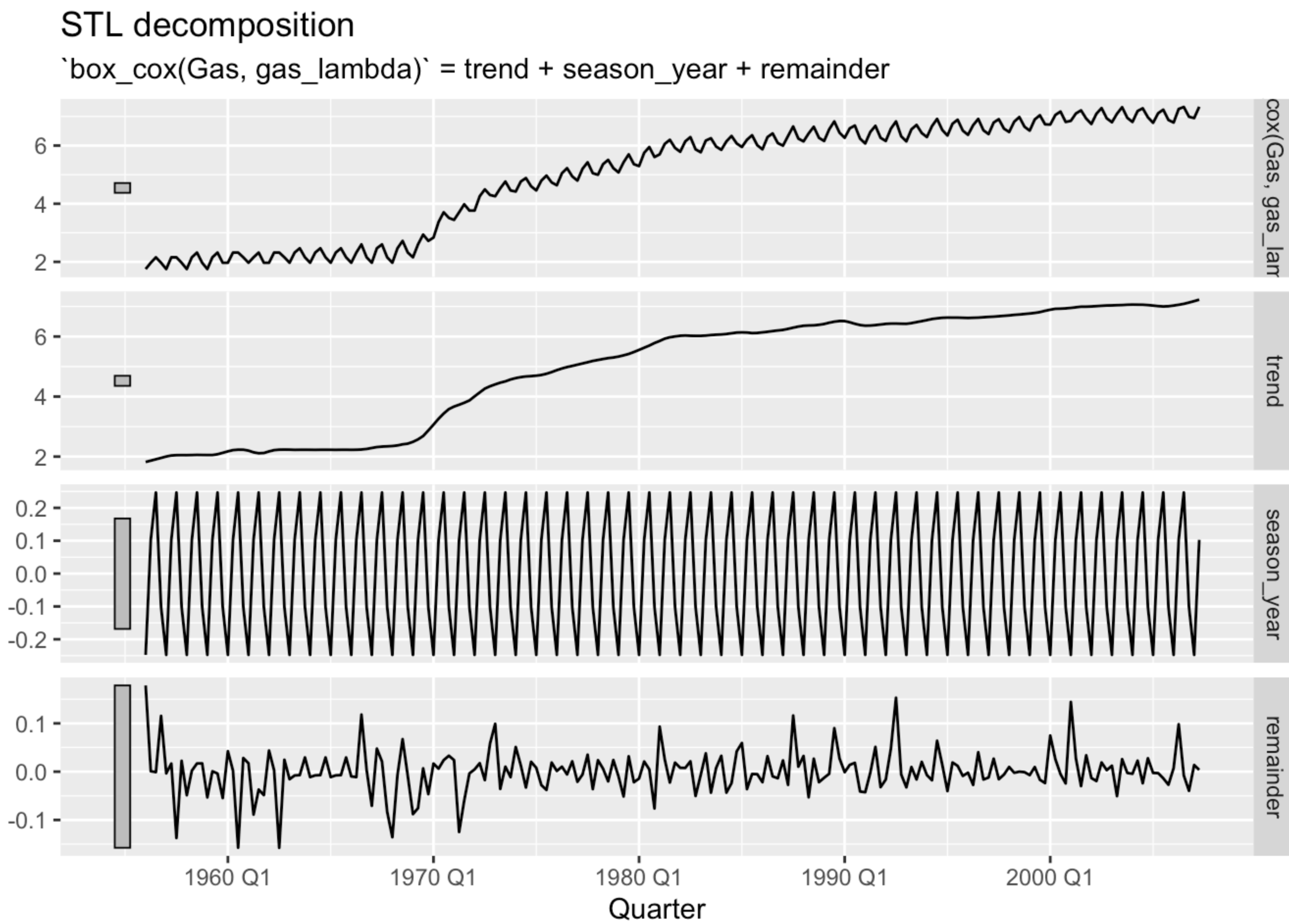
$$MAPE = mean(|P_t|)$$

```
gas_fc |>
  accuracy(aus_production) |>
  arrange(RMSE)
```

```
# A tibble: 3 × 10
  .model .type     ME  RMSE    MAE    MPE  MAPE  MASE RMSSE    ACF1
  <chr>  <chr>  <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
1 snaive Test    8.12  10.6   9.42   3.72  4.28  1.70  1.39 -0.0968
2 drift  Test  -56.5   64.7  57.3  -26.5  26.8  10.4   8.51  0.357
3 media  Test  127.   128.  127.    56.7  56.7  22.9  16.9  -0.0534
```

Pronóstico por descomposición

```
gas_train |>
  model(
    stl = STL(box_cox(Gas, gas_lambda) ~ season(window= "periodic"), robust = TRUE)
  ) |> #robust = true es para que los outliers el efecto se vaya al componente residual
  components() |>
  autoplot()
```




```
gas_dcmp <- gas_train |>
  model(
    dcmp = decomposition_model(
      STL(box_cox(Gas, gas_lambda) ~ season(window = "periodic"), robust = TRUE),
      RW(season_adjust ~ drift()),
      SNAIVE(season_year)
```

```
    )
  )

gas_dcmp
```

1. `decomposition_model()` define que se realizará un pronóstico a partir de una descomposición.
2. Primero se define cómo se realizará la descomposición. En este caso, con STL, con ajuste robusto y con componente estacional periódica.
3. Luego se define el modelo para la serie desestacionalizada, `season_adjust`.
4. Finalmente, se define el modelo para la serie estacional, `season_year`. Si este componente no se especifica, R va a utilizar `SNAIVE()` por default.

```
# A mable: 1 x 1
                           dcmp
                        <model>
1 <STL decomposition model>
```

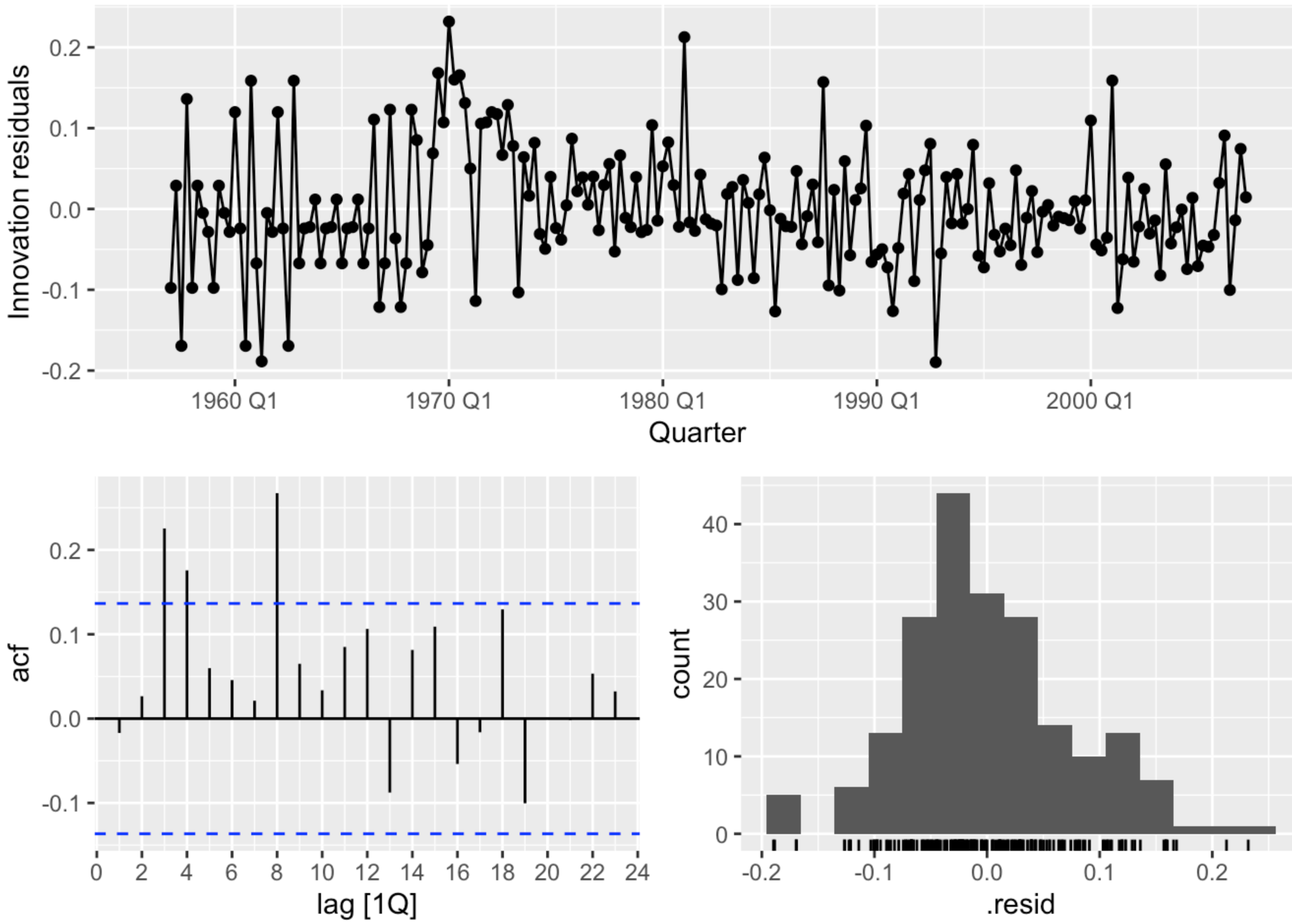
$H_0 : Q = 0$ ➤ No autocorrelación

$H_i : Q \neq 0$ ➤ Sí hay autocorrelación

Rechazar o no una hipotesis H_0 depende de una α, donde α es la probabilidad de cometer un error de tipo I (Probabilidad de rechazar H_0 cuando es verdadera). El error de tipo II es No rechazar la H_0 cuando es falsa.

si p-value < α rechazo H_0

```
gas_dcmp |>
  gg_tsresiduals()
```




```
gas_dcmp |>
  augment() |>
  features(.innov, ljung_box, lag = 8)
```

```
# A tibble: 1 × 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 dcmp      33.6 0.0000473
```

```
gas_dcmp |>
  accuracy()
```

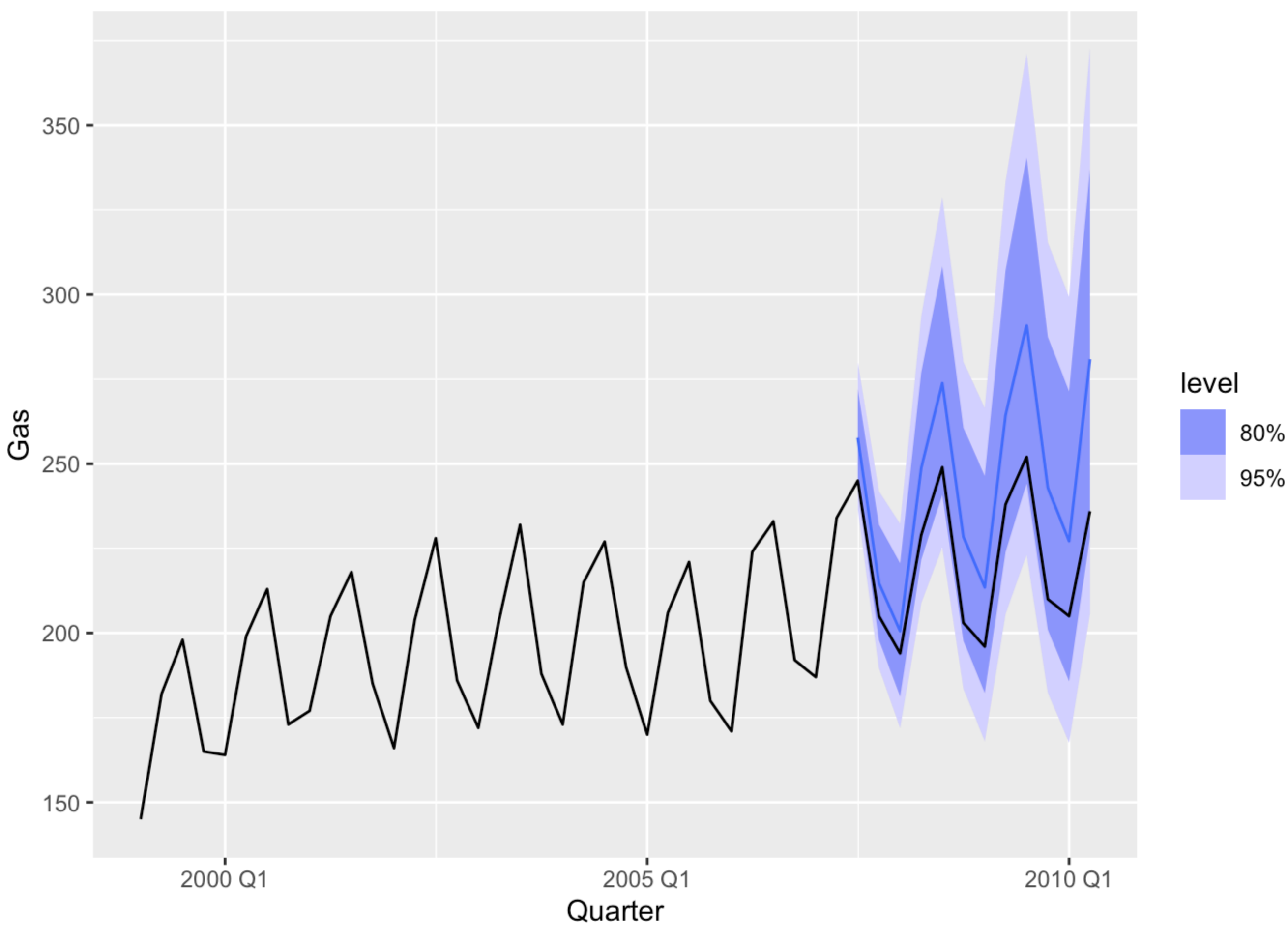
```
# A tibble: 1 × 10
  .model .type        ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
  <chr>  <chr>     <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
1 dcmp   Training -0.381  4.28  2.83 -0.108  4.02 0.512 0.563 -0.184
```

```
gas_dcmp_fc <- gas_dcmp |>
  forecast(h = "3 years")

gas_dcmp_fc
```

```
# A fable: 12 x 4 [1Q]
# Key:     .model [1]
   .model Quarter                Gas .mean
   <chr>    <qtr>             <dist> <dbl>
 1 dcmp   2007 Q3 t(N(7.5, 0.0058))  258.
 2 dcmp   2007 Q4  t(N(7.2, 0.012))  215.
 3 dcmp   2008 Q1  t(N(7.1, 0.018))  201.
 4 dcmp   2008 Q2  t(N(7.4, 0.024))  249.
 5 dcmp   2008 Q3   t(N(7.6, 0.03))  274.
 6 dcmp   2008 Q4  t(N(7.3, 0.036))  228.
 7 dcmp   2009 Q1  t(N(7.2, 0.042))  213.
 8 dcmp   2009 Q2  t(N(7.5, 0.048))  264.
 9 dcmp   2009 Q3  t(N(7.7, 0.054))  291.
10 dcmp   2009 Q4  t(N(7.4, 0.061))  243.
11 dcmp   2010 Q1  t(N(7.3, 0.067))  227.
12 dcmp   2010 Q2  t(N(7.6, 0.074))  281.
```

```r
gas_dcmp_fc |>
  autoplot(aus_production |> filter_index("1999 Q1" ~ .))
```




```r
gas_fc_full <- gas_fc |>
  full_join(gas_dcmp_fc)
```

```
Joining with `by = join_by(.model, Quarter, Gas, .mean)`
```

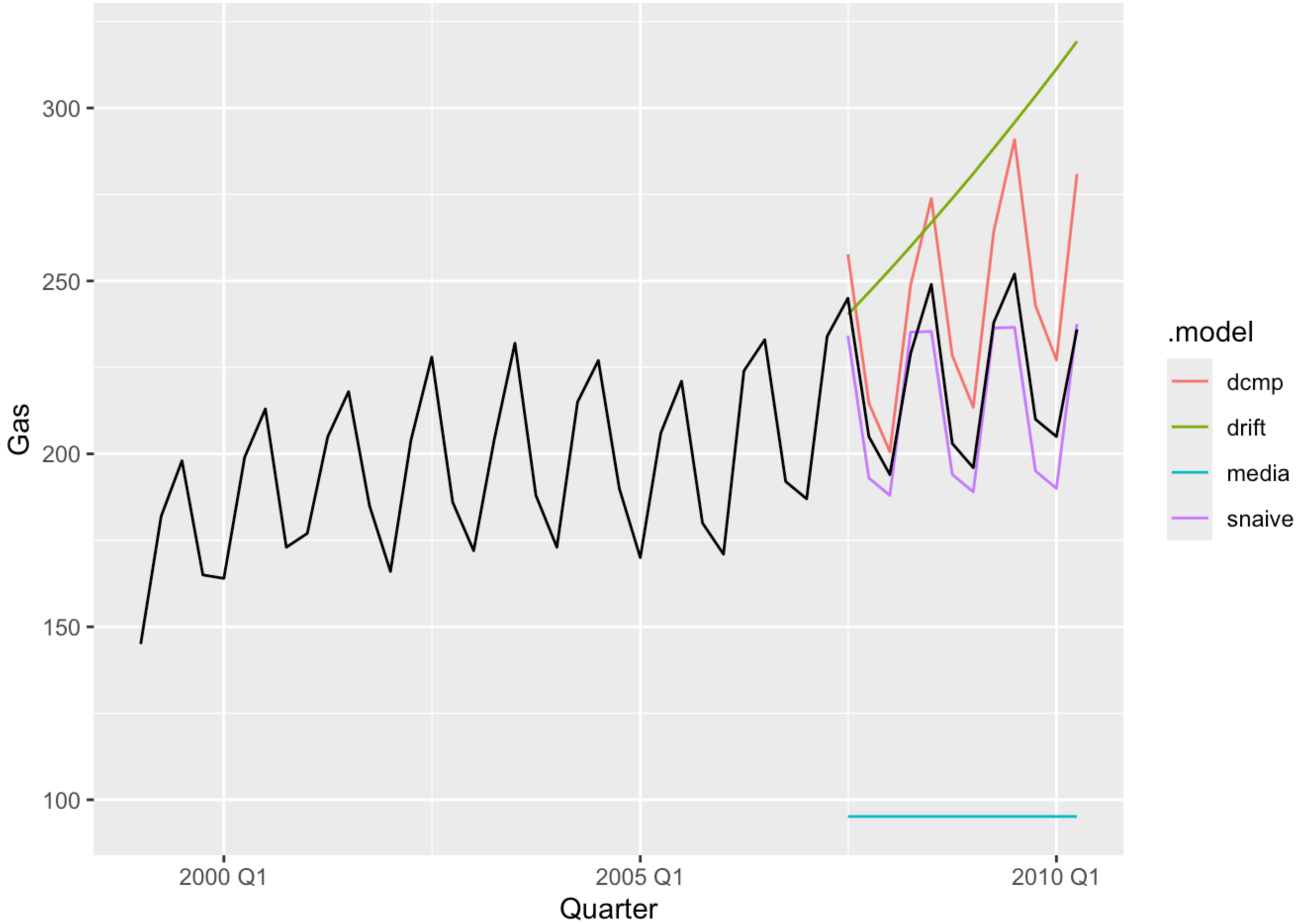
```r
gas_fc_full
```

```
# A fable: 48 x 4 [1Q]
# Key:     .model [4]
   .model Quarter               Gas .mean
   <chr>    <qtr>            <dist> <dbl>
 1 dcmp   2007 Q3 t(N(7.5, 0.0058))  258.
 2 dcmp   2007 Q4  t(N(7.2, 0.012))  215.
 3 dcmp   2008 Q1  t(N(7.1, 0.018))  201.
 4 dcmp   2008 Q2  t(N(7.4, 0.024))  249.
 5 dcmp   2008 Q3   t(N(7.6, 0.03))  274.
 6 dcmp   2008 Q4  t(N(7.3, 0.036))  228.
 7 dcmp   2009 Q1  t(N(7.2, 0.042))  213.
 8 dcmp   2009 Q2  t(N(7.5, 0.048))  264.
 9 dcmp   2009 Q3  t(N(7.7, 0.054))  291.
10 dcmp   2009 Q4  t(N(7.4, 0.061))  243.
# ℹ 38 more rows
```

```r
gas_fc_full |>
  accuracy(aus_production) |>
  arrange(RMSE)
```

```
# A tibble: 4 × 10
  .model .type     ME  RMSE    MAE    MPE  MAPE  MASE RMSSE    ACF1
  <chr>  <chr>  <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
1 snaive Test    8.12  10.6   9.42   3.72  4.28  1.70  1.39 -0.0968
2 dcmp   Test  -23.5   25.9  23.5  -10.4  10.4   4.24  3.41  0.389
3 drift  Test  -56.5   64.7  57.3  -26.5  26.8  10.4   8.51  0.357
4 media  Test  127.   128.  127.    56.7  56.7  22.9  16.9  -0.0534
```

```r
gas_fc_full |>
  autoplot(aus_production |> filter_index("1999 Q1" ~ .), level = NULL)
```



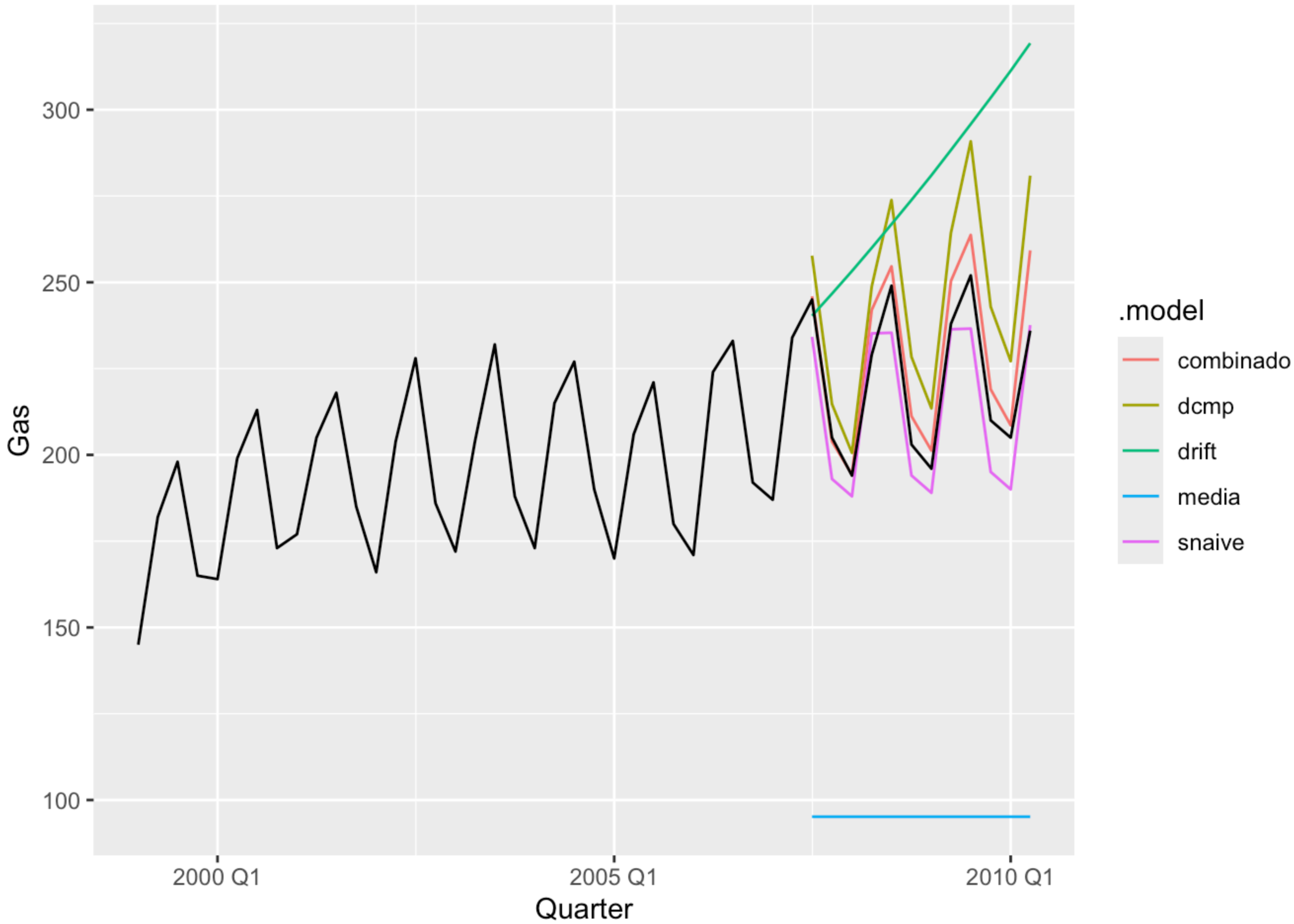

```
gas_fit_full <- gas_fit |>
  cross_join(gas_dcmp) |>
  mutate(combinado = (snaive + dcmp)/2)

gas_fit_full
```

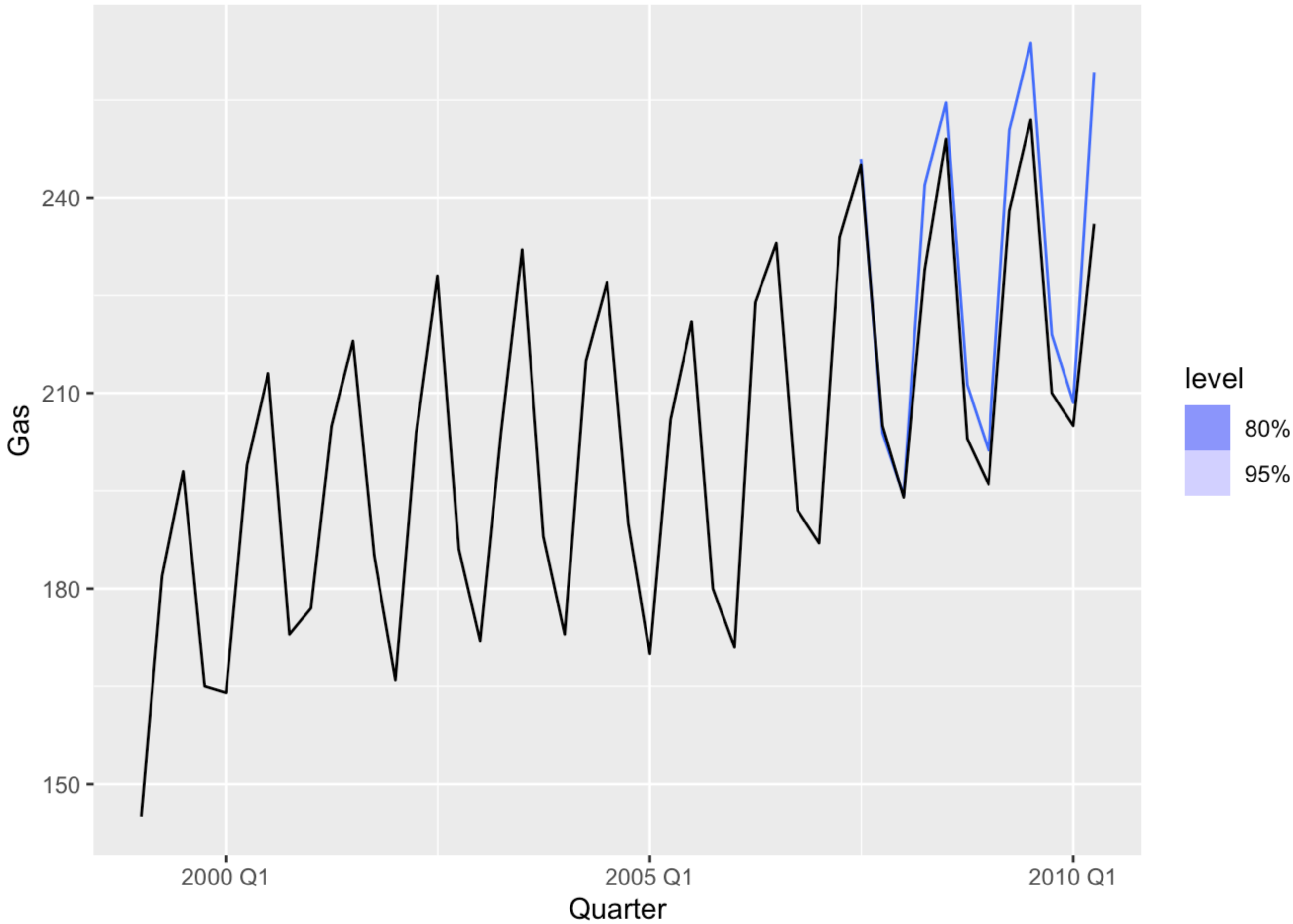
```
# A mable: 1 x 5
          drift   snaive   media                          dcmp     combinado
        <model>  <model> <model>                       <model>       <model>
1 <RW w/ drift> <SNAIVE>  <MEAN> <STL decomposition model> <COMBINATION>
```

```
gas_fcst_full <- gas_fit_full |>
  forecast(h = "3 years")

gas_fcst_full |>
  autoplot(aus_production |> filter_index("1999 Q1" ~ .), level = NULL)
```




```
gas_fcst_full |>
  filter(.model == "combinado") |>
  autoplot(aus_production |> filter_index("1999 Q1" ~ .))
```




```
gas_fcst_full |>
  accuracy(aus_production) |>
  arrange(RMSE)
```

```
# A tibble: 5 × 10
  .model    .type     ME  RMSE    MAE    MPE  MAPE  MASE RMSSE    ACF1
  <chr>     <chr>  <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
1 combinado Test   -7.67  10.1   7.86  -3.36  3.46  1.42  1.33  0.0294
2 snaive    Test    8.12  10.6   9.42   3.72  4.28  1.70  1.39 -0.0968
3 dcmp      Test  -23.5   25.9  23.5  -10.4  10.4   4.24  3.41  0.389
4 drift     Test  -56.5   64.7  57.3  -26.5  26.8  10.4   8.51  0.357
5 media     Test  127.   128.  127.    56.7  56.7  22.9  16.9  -0.0534
```

ETS

```
ses <- gas_train |>
  model(
    ses = ETS(Gas ~ error("A") + trend("N") + season("N"))
  )

ses
```

```
# A mable: 1 x 1
          ses
      <model>
1 <ETS(A,N,N)>
```

1. `ETS()` es la función para estimar modelos de suavización exponencial y se deben definir 3 argumentos: error, tendencia y estacionalidad.
2. Tenemos dos opciones: “A” para error aditivo, y “M” para multiplicativo
3. Para especificar que no queremos ni tendencia, ni estacionalidad, ponemos “N” en ambos casos.

```r
report(ses)
```

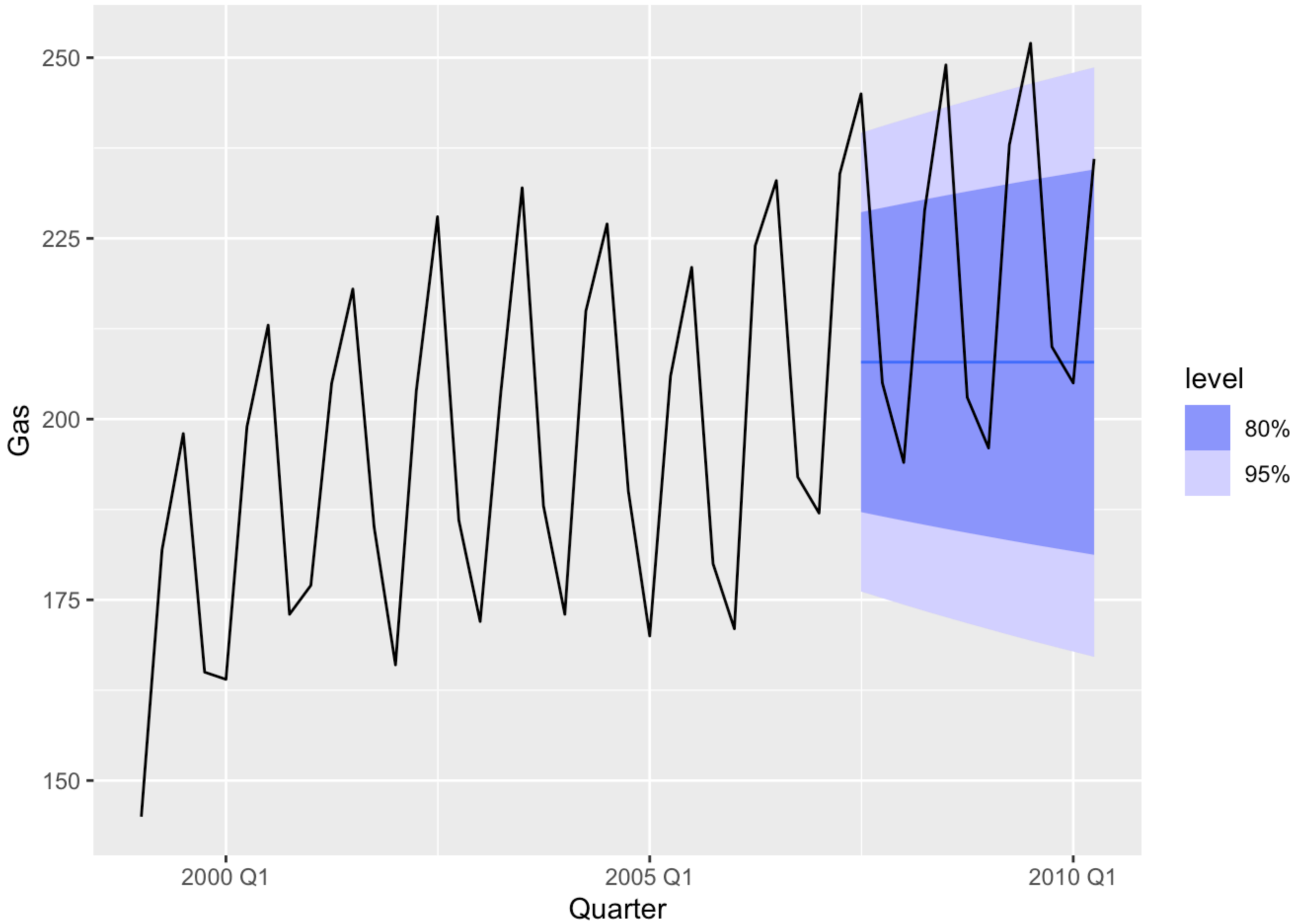
```
Series: Gas
Model: ETS(A,N,N)
  Smoothing parameters:
    alpha = 0.2434761

  Initial states:
   l[0]
 5.9707

  sigma^2:  261.8623

     AIC     AICc      BIC
2248.503 2248.622 2258.487
```

```r
ses |>
  forecast(h = "3 years") |>
  autoplot(aus_prod_recent)
```



Tendencia lineal de HOLT

```
holt <- gas_train |>
  model(
    holt = ETS(Gas ~ error("A") + trend("A") + season("N"))
  )

holt
```

```
# A mable: 1 x 1
          holt
       <model>
1 <ETS(A,A,N)>
```

```
report(holt)
```

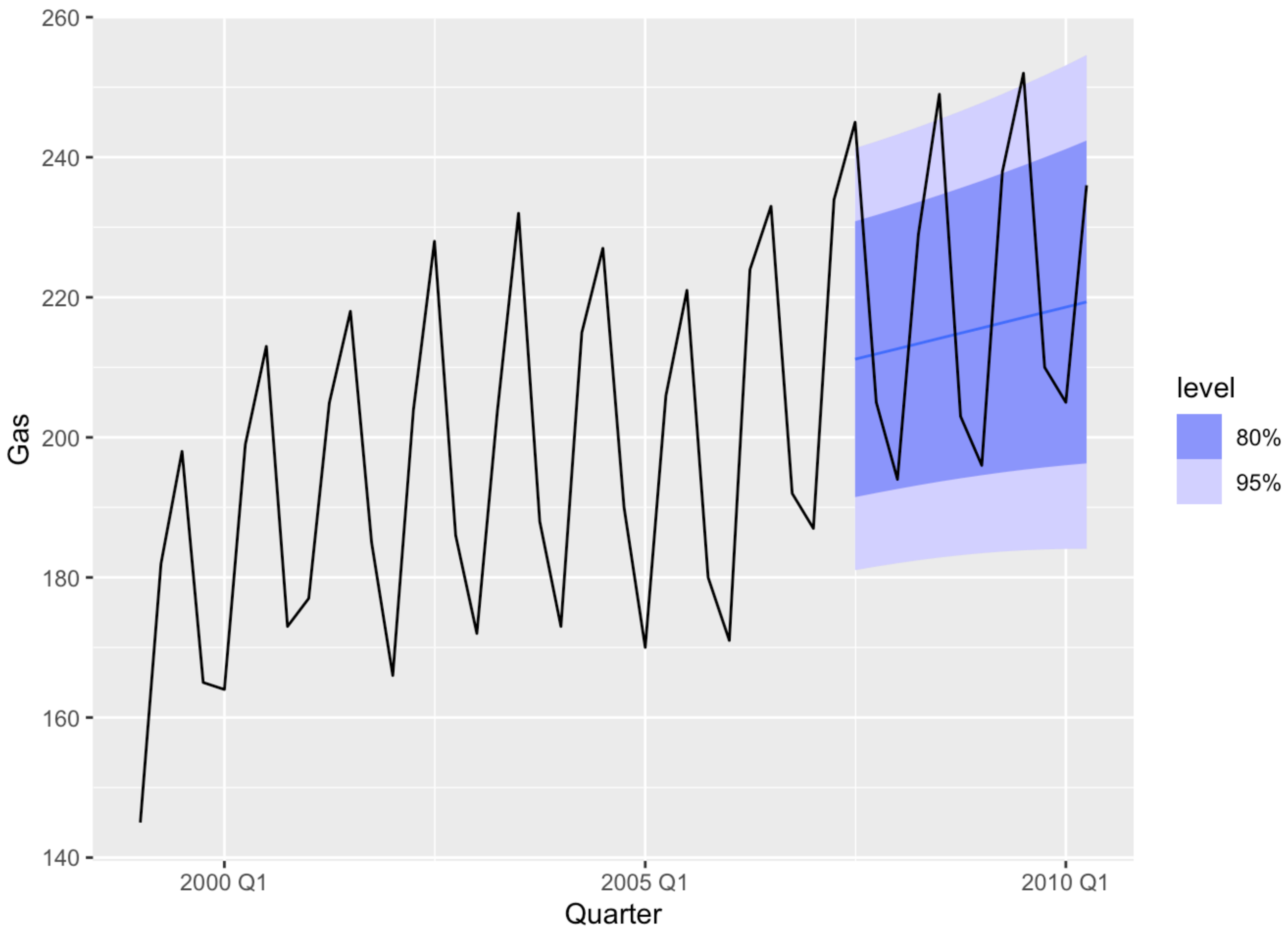
```
Series: Gas
Model: ETS(A,A,N)
  Smoothing parameters:
    alpha = 0.1117699
    beta  = 0.01133185
```

```
  Initial states:
     l[0]       b[0]
 6.681936 0.01735882

  sigma^2:  236.0876

     AIC     AICc      BIC
2229.129 2229.429 2245.768
```

```
holt |>
  forecast(h = "3 years") |>
  autoplot(aus_prod_recent)
```



Tendencia amortiguada

Holt winters

```
hw <- gas_train |>
  model(
    hw = ETS(Gas ~ error("M") + trend("Ad", phi = 0.8) + season("M")) #poner error = esta
```

```r
  )

report(hw)
```

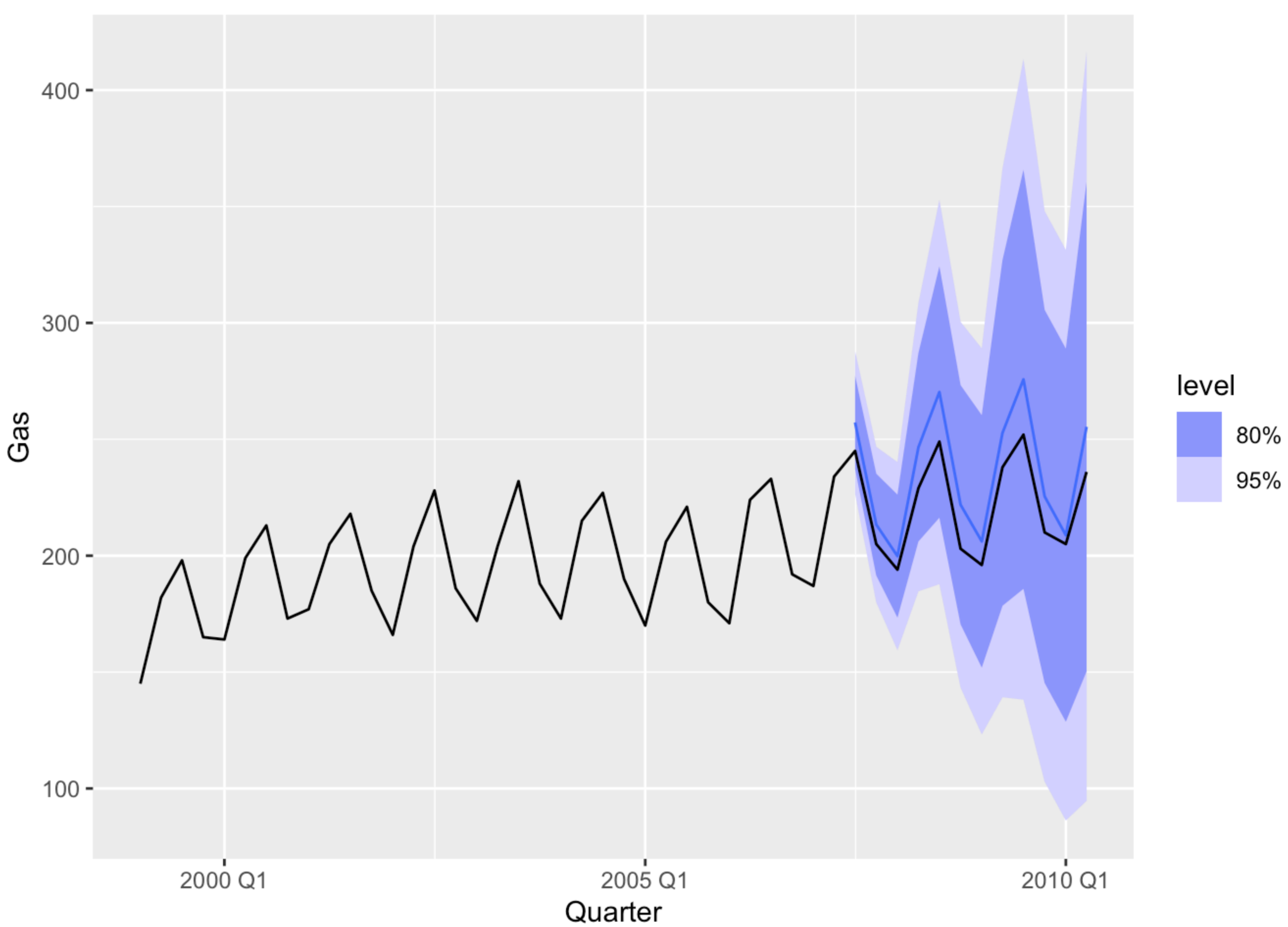
```
Series: Gas
Model: ETS(M,Ad,M)
  Smoothing parameters:
    alpha = 0.5788133
    beta  = 0.2907041
    gamma = 0.0736501
    phi   = 0.8

  Initial states:
     l[0]       b[0]      s[0]    s[-1]    s[-2]     s[-3]
 5.837584 0.08005418 0.9305848 1.182244 1.069461 0.8177104

  sigma^2:  0.0037

     AIC     AICc      BIC
1565.797 1566.715 1595.748
```

```r
hw |>
  forecast(h = "3 years") |>
  autoplot(aus_prod_recent)
```



Comparando modelos

```
STLF <- decomposition_model(
  STL(box_cox(Gas, gas_lambda) ~ season(window = "periodic"), robust=TRUE),
  RW(season_adjust ~ drift())
)

STLF_ets <- decomposition_model(
  STL(box_cox(Gas, gas_lambda) ~ season(window = "periodic"), robust=TRUE),
  ETS(season_adjust ~ error("A") + trend("Ad") + season("N"))
)



gas_fit_todos <- gas_train |>
  model(
    hw = ETS(Gas ~ error("M") + trend("Ad", phi = 0.8) + season("M")),
    snaive = SNAIVE(box_cox(Gas, gas_lambda)),
    hw_boxcox = ETS(box_cox(Gas, lambda = gas_lambda) ~ error("A") + trend("Ad") + season
    stlf = STLF,
    stlf_ets = STLF_ets
  )

gas_fit_todos
```

```
# A mable: 1 x 5
             hw   snaive     hw_boxcox                     stlf
        <model>  <model>       <model>                  <model>
1 <ETS(M,Ad,M)> <SNAIVE> <ETS(A,Ad,A)> <STL decomposition model>
# ℹ 1 more variable: stlf_ets <model>
```

```
gas_fc_todos <- gas_fit_todos |>
  forecast(h = "3 years")

gas_fc_todos |>
  accuracy(aus_production) |>
  arrange(RMSE)
```

```
# A tibble: 5 × 10
  .model    .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE    ACF1
  <chr>     <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
1 snaive    Test    8.12  10.6  9.42   3.72  4.28  1.70  1.39 -0.0968
2 hw        Test  -14.3   15.5 14.3   -6.32  6.32  2.58  2.04  0.0430
3 stlf_ets  Test  -16.2   18.0 16.2   -7.19  7.19  2.93  2.37  0.201
4 hw_boxcox Test  -16.6   18.3 16.6   -7.37  7.37  3.00  2.40  0.231
5 stlf      Test  -23.5   25.9 23.5  -10.4  10.4   4.24  3.41  0.389
```

```
p <- gas_fc_todos |>
  autoplot(aus_prod_recent, level = NULL)
```

```
plotly::ggplotly(p)
```