pronóstico_arima

Pronóstico & filtro ARIMA

```
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
```

```
library(fpp3)
```

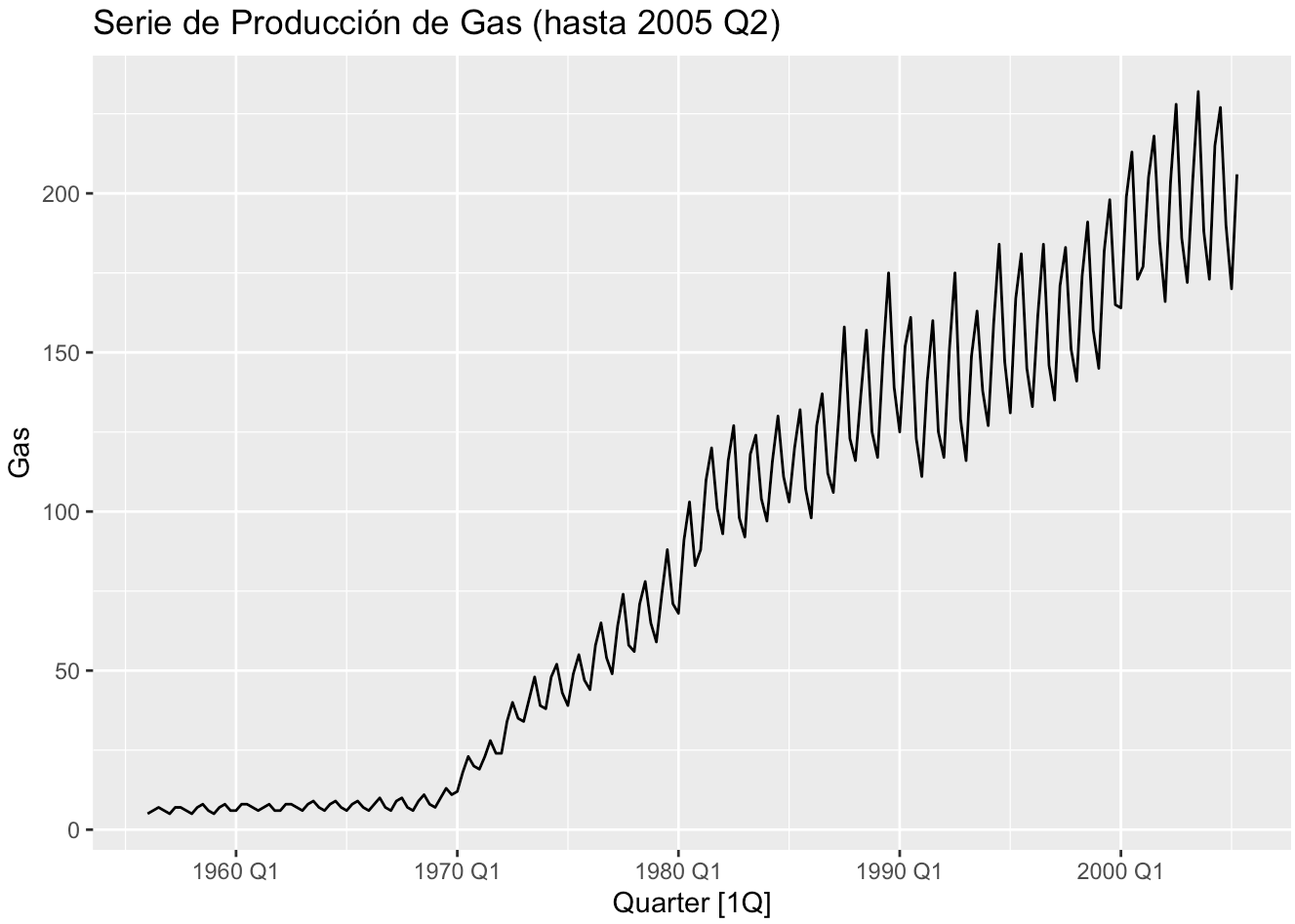
```
Registered S3 method overwritten by 'tsibble':
  method               from
  as_tibble.grouped_df dplyr
── Attaching packages ─────────────────────────────────────────── fpp3 1.0.1 ──
✔ tsibble     1.1.6     ✔ feasts      0.4.1
✔ tsibbledata 0.4.1     ✔ fable       0.4.1
── Conflicts ────────────────────────────────────────── 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()
```

Preparar set de train

```
# Preparar datos de entrenamiento hasta 2005 Q2
gas_train <- aus_production |>
  filter_index(. ~ "2005 Q2")
```

```
# Calcular lambda para transformación Box-Cox
gas_lambda <- gas_train |>
  features(Gas, features = guerrero) |>
  pull()
```

```
# Visualizar serie original
gas_train |>
  autoplot(Gas) +
  labs(title = "Serie de Producción de Gas (hasta 2005 Q2)")
```




```
# ==========================================
# MODELOS MEJORADOS
# ==========================================

gas_fit_mejorado <- gas_train |>
  model(
    # Modelos básicos para comparación
    snaive = SNAIVE(box_cox(Gas, gas_lambda)),
    drift = RW(box_cox(Gas, gas_lambda) ~ drift()),

    # ETS automático (encuentra el mejor modelo)
    ets_auto = ETS(box_cox(Gas, gas_lambda)),

    # ETS específicos
    ets_aaa = ETS(box_cox(Gas, gas_lambda) ~ error("A") + trend("A") + season("A")),
    ets_ada = ETS(box_cox(Gas, gas_lambda) ~ error("A") + trend("Ad") + season("A")),
    ets_mam = ETS(Gas ~ error("M") + trend("A") + season("M")),
    ets_mad = ETS(Gas ~ error("M") + trend("Ad") + season("M")),
```

```
    # ARIMA automático
    arima_auto = ARIMA(box_cox(Gas, gas_lambda)),

    # ARIMA manual con estacionalidad
    arima_manual = ARIMA(box_cox(Gas, gas_lambda) ~ pdq(1,1,1) + PDQ(1,1,1)),

    # Descomposición STL con ETS
    stl_ets = decomposition_model(
      STL(box_cox(Gas, gas_lambda) ~ season(window = "periodic"), robust = TRUE),
      ETS(season_adjust ~ error("A") + trend("Ad") + season("N"))
    ),

    # Descomposición STL con ARIMA
    stl_arima = decomposition_model(
      STL(box_cox(Gas, gas_lambda) ~ season(window = "periodic"), robust = TRUE),
      ARIMA(season_adjust)
    )
  )
```

Mostrar Resumen de Modelos mejorados

```
# Mostrar resumen de modelos
print(gas_fit_mejorado)
```

```
# A mable: 1 x 11
    snaive         drift     ets_auto      ets_aaa       ets_ada      ets_mam
   <model>       <model>      <model>      <model>       <model>      <model>
1 <SNAIVE> <RW w/ drift> <ETS(A,A,A)> <ETS(A,A,A)> <ETS(A,Ad,A)> <ETS(M,A,M)>
# ℹ 5 more variables: ets_mad <model>, arima_auto <model>,
#   arima_manual <model>, stl_ets <model>, stl_arima <model>
```

Diagnóstico de Residuales

```
cat("\n\nREALIZANDO DIAGNÓSTICO DE LOS MODELOS PRINCIPALES...\n")
```

```


REALIZANDO DIAGNÓSTICO DE LOS MODELOS PRINCIPALES...
```

```
# ETS automático
cat("\n", strrep("=", 60), "\n")
```

```

 ============================================================
```

```
cat("DIAGNÓSTICO: ETS Automático\n")
```

```
DIAGNÓSTICO: ETS Automático
```

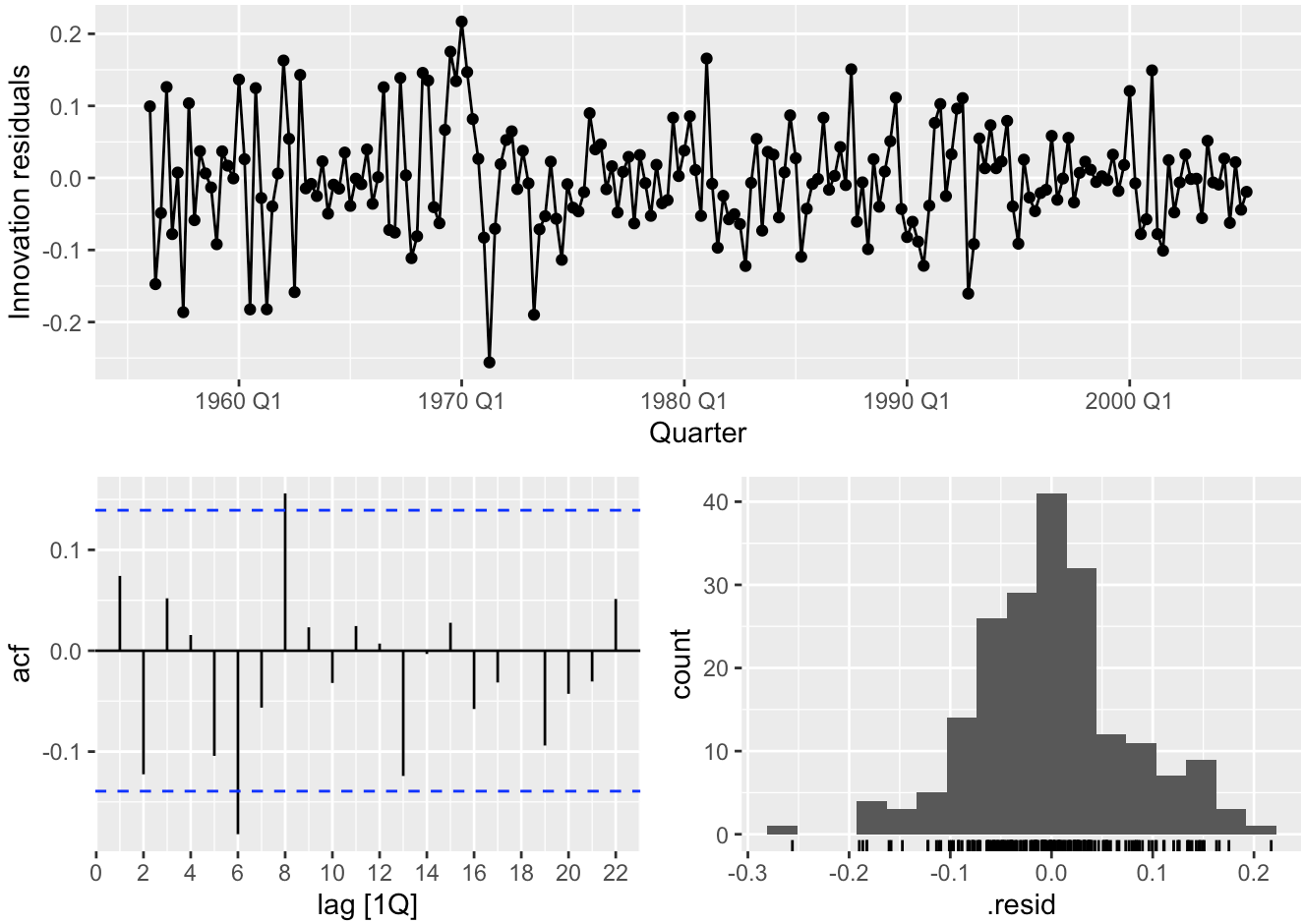
```
cat(strrep("=", 60), "\n\n")
```

```
============================================================
```

```
print(report(gas_fit_mejorado |> select(ets_auto)))
```

```
Series: Gas
Model: ETS(A,A,A)
Transformation: box_cox(Gas, gas_lambda)
  Smoothing parameters:
    alpha = 0.6713216
    beta  = 0.1742178
    gamma = 0.0001002953

  Initial states:
    l[0]       b[0]       s[0]     s[-1]      s[-2]      s[-3]
 1.85807 0.04008532 -0.1015126 0.2512357 0.09531945 -0.2450425

  sigma^2:  0.006

     AIC     AICc      BIC
42.67553 43.63298 72.26994
# A mable: 1 x 1
      ets_auto
       <model>
1 <ETS(A,A,A)>
```

```
gas_fit_mejorado |>
  select(ets_auto) |>
  gg_tsresiduals()
```




```
lb_test_ets <- gas_fit_mejorado |>
  select(ets_auto) |>
  augment() |>
  features(.innov, ljung_box, lag = 8)
cat("\nTest de Ljung-Box:\n")
```

```
Test de Ljung-Box:
```

```
print(lb_test_ets)
```

```
# A tibble: 1 × 3
  .model   lb_stat lb_pvalue
  <chr>      <dbl>     <dbl>
1 ets_auto    19.5    0.0123
```

```
# ARIMA automático
cat("\n", strrep("=", 60), "\n")
```

```

 ============================================================
```

```
cat("DIAGNÓSTICO: ARIMA Automático\n")
```

```
DIAGNÓSTICO: ARIMA Automático
```

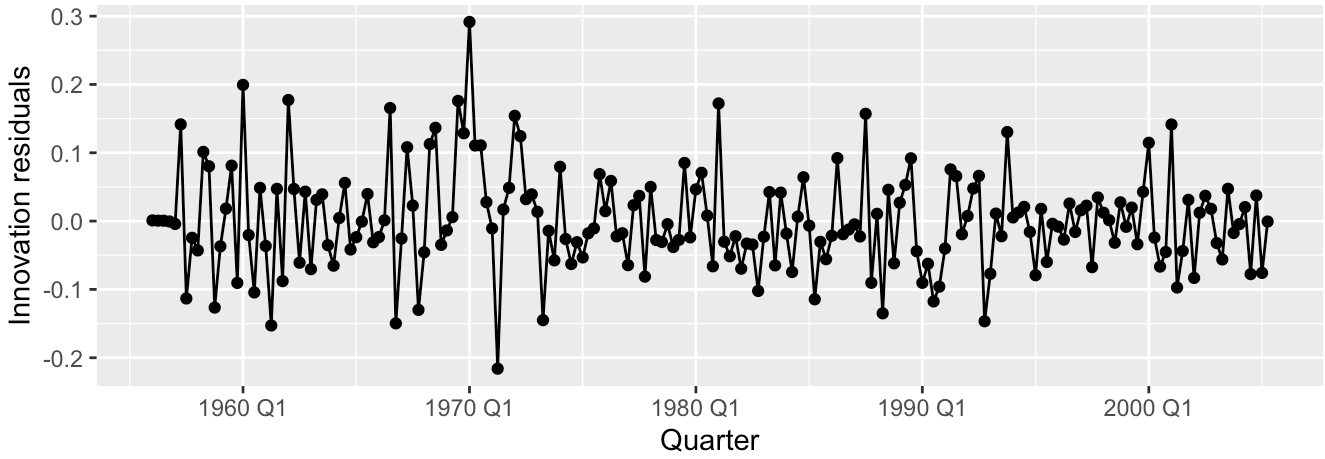
```
cat(strrep("=", 60), "\n\n")
```

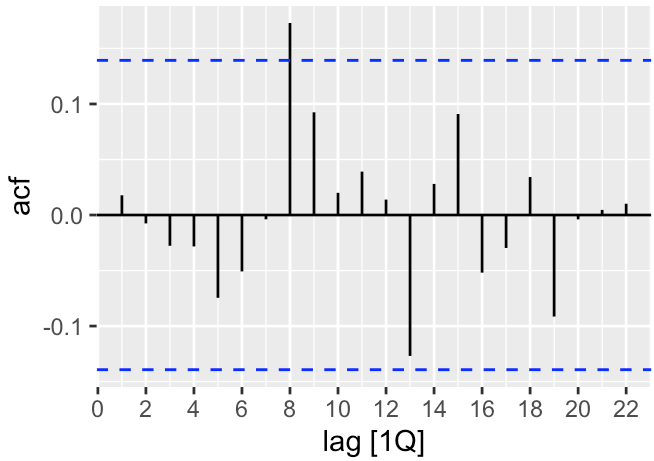
```
============================================================
```

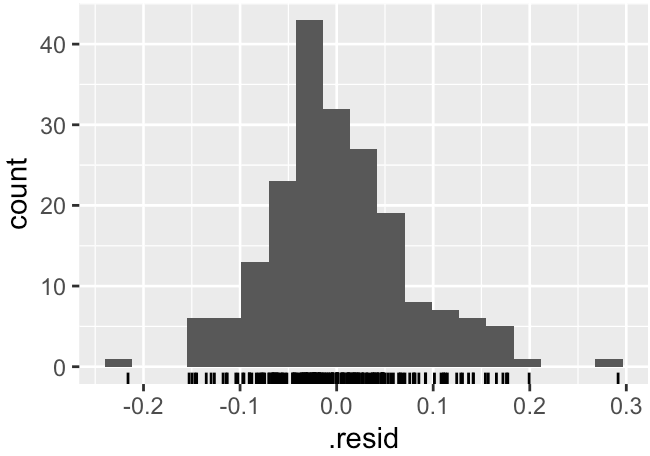
```
print(report(gas_fit_mejorado |> select(arima_auto)))
```

```
Series: Gas
Model: ARIMA(3,1,1)(1,1,1)[4]
Transformation: box_cox(Gas, gas_lambda)

Coefficients:
          ar1      ar2     ar3     ma1    sar1     sma1
      -0.6459  -0.0677  0.2500  0.5819  0.3118  -0.8907
s.e.   0.1605   0.0888  0.0812  0.1562  0.1138   0.0732

sigma^2 estimated as 0.005654:  log likelihood=226.02
AIC=-438.04   AICc=-437.43   BIC=-415.2
# A mable: 1 x 1
                arima_auto
                   <model>
1 <ARIMA(3,1,1)(1,1,1)[4]>
```

```
gas_fit_mejorado |>
  select(arima_auto) |>
  gg_tsresiduals()
```








```
lb_test_arima <- gas_fit_mejorado |>
  select(arima_auto) |>
  augment() |>
  features(.innov, ljung_box, lag = 8)
cat("\nTest de Ljung-Box:\n")
```

```
Test de Ljung-Box:
```

```
print(lb_test_arima)
```

```
# A tibble: 1 × 3
  .model     lb_stat lb_pvalue
  <chr>        <dbl>     <dbl>
1 arima_auto    8.31     0.404
```

```
# STL con ARIMA
cat("\n", strrep("=", 60), "\n")
```

```
 ============================================================
```

```
cat("DIAGNÓSTICO: STL + ARIMA\n")
```

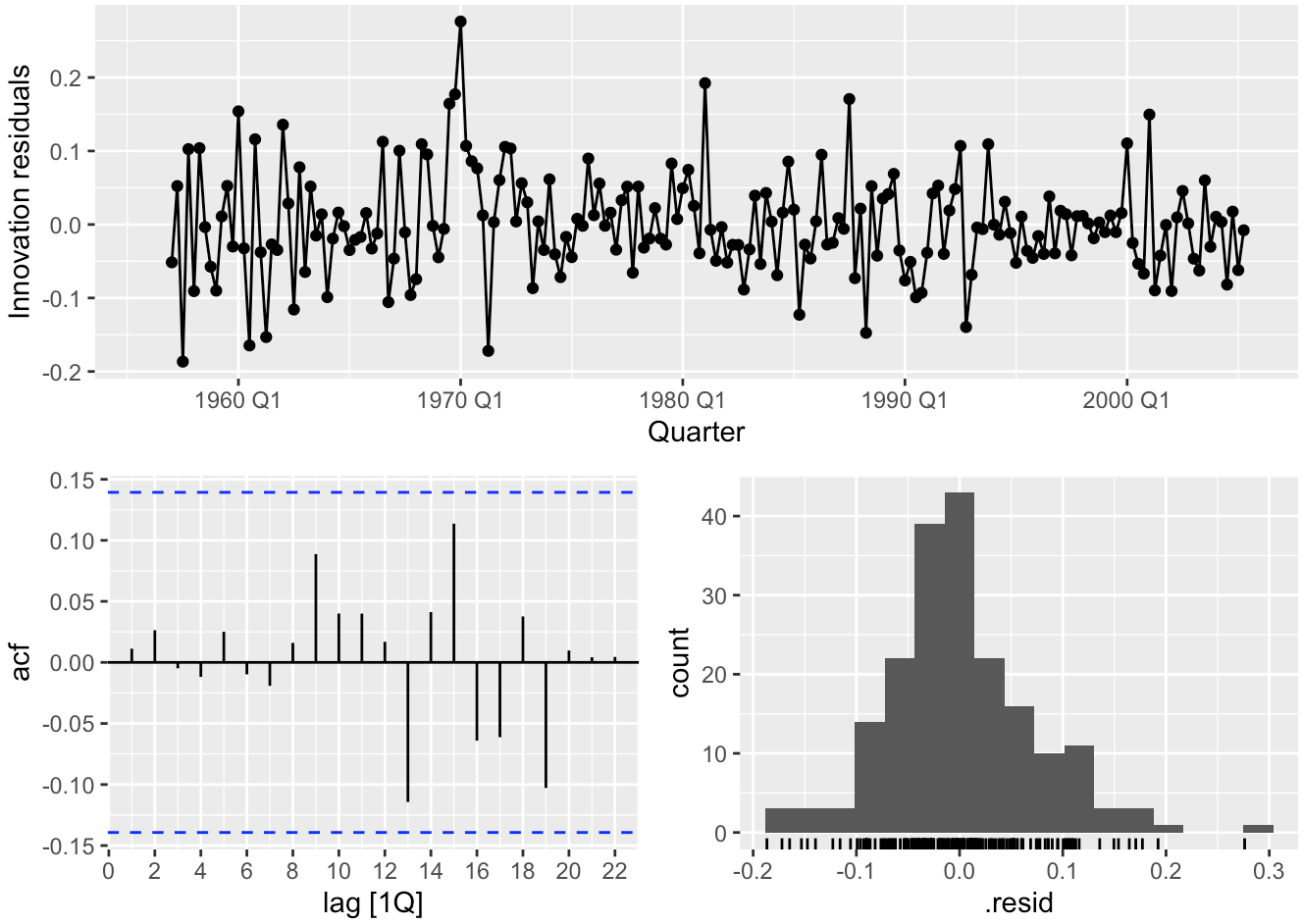
```
DIAGNÓSTICO: STL + ARIMA
```

```
cat(strrep("=", 60), "\n\n")
```

```
============================================================
```

```
gas_fit_mejorado |>
  select(stl_arima) |>
  gg_tsresiduals()
```

```
Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_line()`).

Warning: Removed 4 rows containing missing values or values outside the scale range
(`geom_point()`).

Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_bin()`).
```



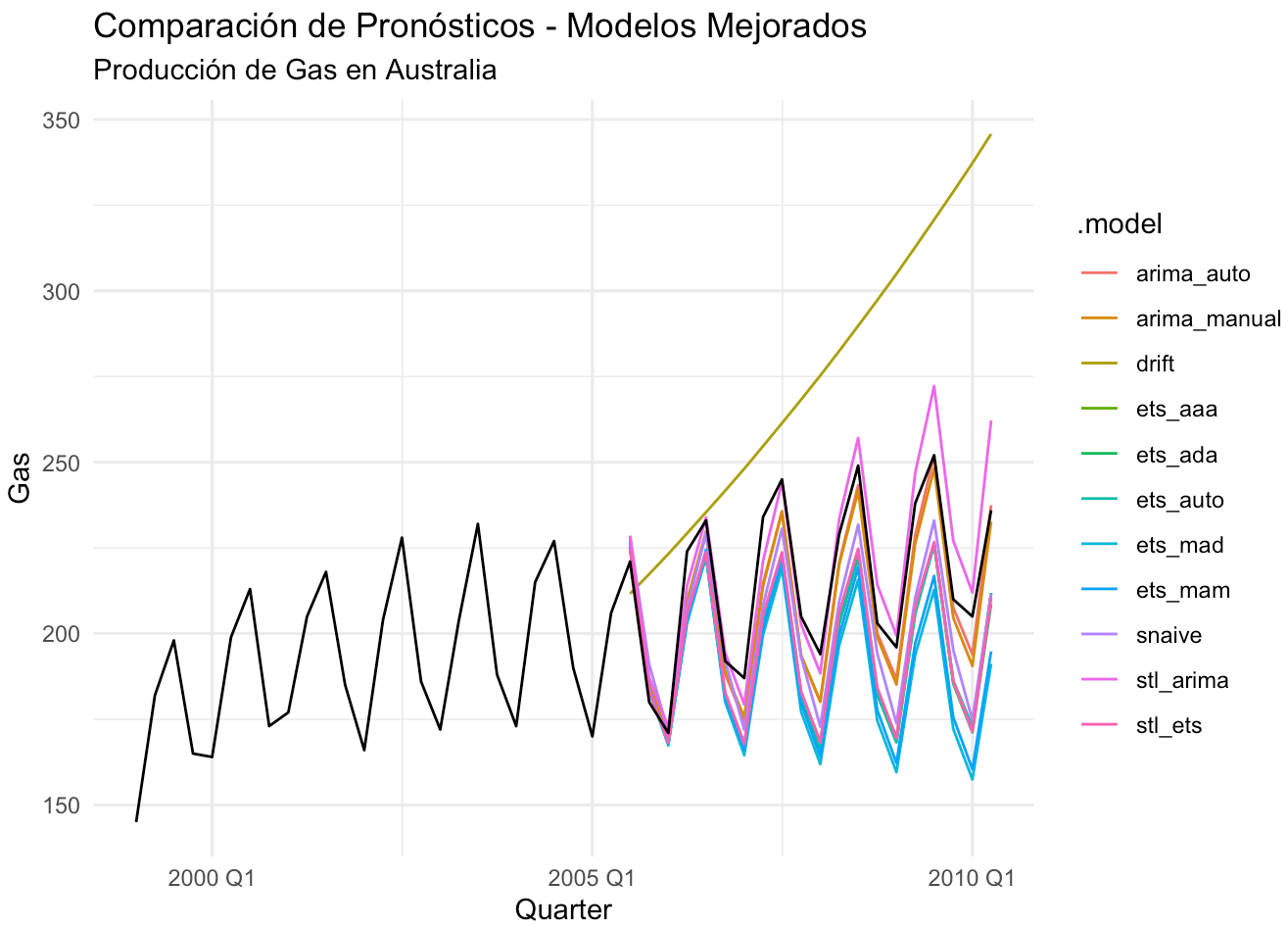
```
lb_test_stl <- gas_fit_mejorado |>
  select(stl_arima) |>
  augment() |>
  features(.innov, ljung_box, lag = 8)
cat("\nTest de Ljung-Box:\n")
```

```
Test de Ljung-Box:
```

```
print(lb_test_stl)
```

```
# A tibble: 1 × 3
  .model    lb_stat lb_pvalue
  <chr>       <dbl>     <dbl>
1 stl_arima   0.470     1.000
```

```
# Generar pronósticos para 3 años (hasta 2008 Q2)
gas_fc_mejorado <- gas_fit_mejorado |>
  forecast(h = "5 years")

# Visualizar pronósticos
aus_prod_recent <- aus_production |>
  filter_index("1999 Q1" ~ .)

gas_fc_mejorado |>
  autoplot(aus_prod_recent, level = NULL) +
  labs(
    title = "Comparación de Pronósticos - Modelos Mejorados",
    subtitle = "Producción de Gas en Australia",
    y = "Gas"
  ) +
  theme_minimal()
```




```
cat("\n", strrep("=", 60), "\n")
```

```
 ============================================================
```

```
cat("EVALUACIÓN DE PRECISIÓN EN CONJUNTO DE PRUEBA\n")
```

```
EVALUACIÓN DE PRECISIÓN EN CONJUNTO DE PRUEBA
```

```
cat(strrep("=", 60), "\n\n")
```

```
============================================================
```

```
accuracy_results <- gas_fc_mejorado |>
  accuracy(aus_production) |>
  arrange(RMSE)

print(accuracy_results)
```

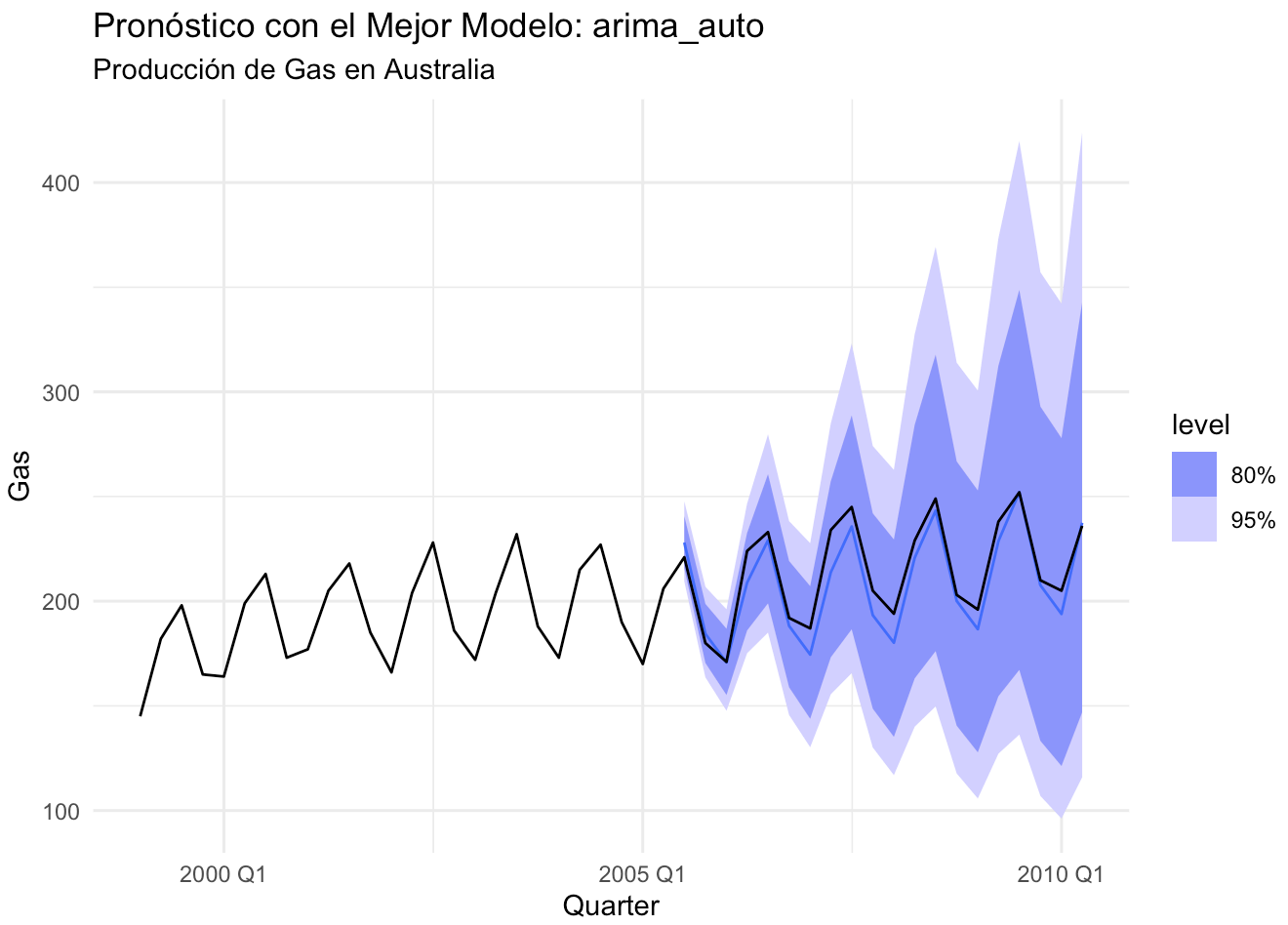
```
# A tibble: 11 × 10
   .model       .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
```

	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	arima_auto	Test	6.34	9.28	7.63	2.94	3.57	1.43	1.26	0.295
2	arima_manual	Test	7.28	9.65	8.31	3.34	3.87	1.56	1.31	0.296
3	stl_arima	Test	-4.35	10.5	8.27	-1.92	3.76	1.55	1.43	0.486
4	snaive	Test	13.7	17.7	15.6	6.19	7.13	2.92	2.40	0.524
5	stl_ets	Test	19.0	21.6	19.7	8.69	9.04	3.70	2.93	0.661
6	ets_ada	Test	19.4	22.1	20.1	8.88	9.24	3.78	3.00	0.656
7	ets_aaa	Test	20.2	22.9	21.0	9.24	9.62	3.93	3.10	0.655
8	ets_auto	Test	20.2	22.9	21.0	9.24	9.62	3.93	3.10	0.655
9	ets_mam	Test	23.9	27.8	24.9	10.9	11.4	4.67	3.77	0.718
10	ets_mad	Test	26.2	30.3	27.1	12.0	12.4	5.10	4.11	0.735
11	drift	Test	-59.1	71.1	60.0	-28.3	28.7	11.3	9.63	0.569

```
mejor_modelo <- accuracy_results$.model[1]
cat("\n\nMEJOR MODELO:", mejor_modelo, "\n")
```

MEJOR MODELO: arima_auto

```
gas_fc_mejorado |>
  filter(.model == mejor_modelo) |>
  autoplot(aus_prod_recent, level = c(80, 95)) +
  labs(
    title = paste("Pronóstico con el Mejor Modelo:", mejor_modelo),
    subtitle = "Producción de Gas en Australia",
    y = "Gas"
  ) +
  theme_minimal()
```




```
top3_models <- accuracy_results$.model[1:3]

gas_fit_ensemble <- gas_train |>
  model(
    ets_auto = ETS(box_cox(Gas, gas_lambda)),
    arima_auto = ARIMA(box_cox(Gas, gas_lambda)),
    stl_arima = decomposition_model(
      STL(box_cox(Gas, gas_lambda) ~ season(window = "periodic"), robust = TRUE),
      ARIMA(season_adjust)
    )
  ) |>
  mutate(
    combinado = (ets_auto + arima_auto + stl_arima) / 3
  )
```

```
gas_fc_ensemble <- gas_fit_ensemble |>
  forecast(h = "5 years")

cat("\n\nEVALUACIÓN DEL MODELO COMBINADO (ENSEMBLE):\n")
```

EVALUACIÓN DEL MODELO COMBINADO (ENSEMBLE):

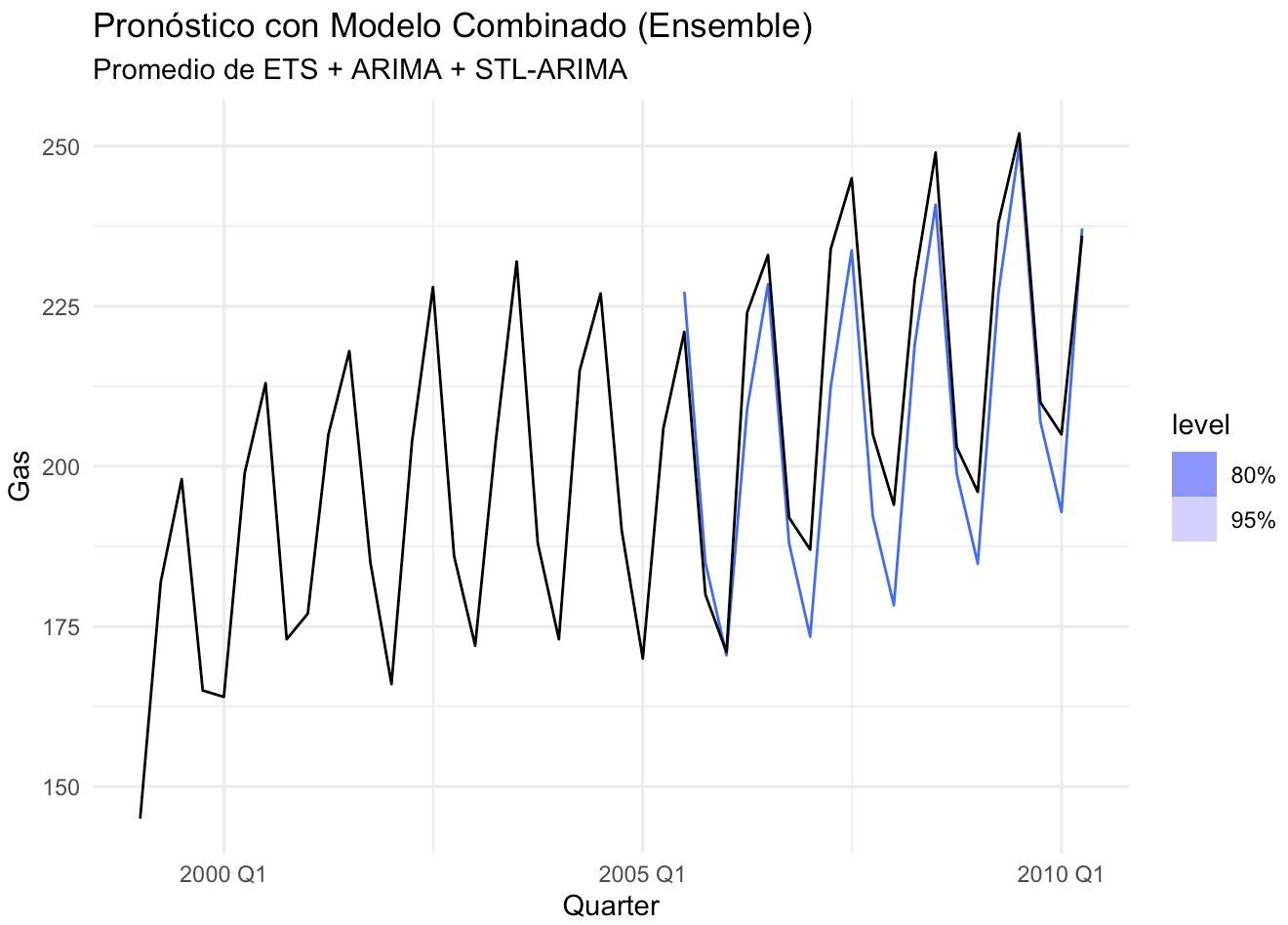
```
cat(strrep("-", 60), "\n")
```

```
------------------------------------------------------------
```

```
accuracy_ensemble <- gas_fc_ensemble |>
  accuracy(aus_production) |>
  arrange(RMSE)

print(accuracy_ensemble)
```

```
# 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 arima_auto Test   6.34  9.28  7.63  2.94  3.57  1.43  1.26 0.295
2 combinado  Test   7.40 10.2   8.63  3.42  4.03  1.62  1.39 0.340
3 stl_arima  Test  -4.35 10.5   8.27 -1.92  3.76  1.55  1.43 0.486
4 ets_auto   Test  20.2  22.9  21.0   9.24  9.62  3.93  3.10 0.655
```

```
gas_fc_ensemble |>
  filter(.model == "combinado") |>
  autoplot(aus_prod_recent, level = c(80, 95)) +
  labs(
    title = "Pronóstico con Modelo Combinado (Ensemble)",
    subtitle = "Promedio de ETS + ARIMA + STL-ARIMA",
    y = "Gas"
  ) +
  theme_minimal()
```




```
cat("\n", strrep("=", 60), "\n")
```

```
============================================================
```

```
cat("COMPARACIÓN FINAL: MODELOS BÁSICOS VS MEJORADOS\n")
```

```
COMPARACIÓN FINAL: MODELOS BÁSICOS VS MEJORADOS
```

```
cat(strrep("=", 60), "\n\n")
```

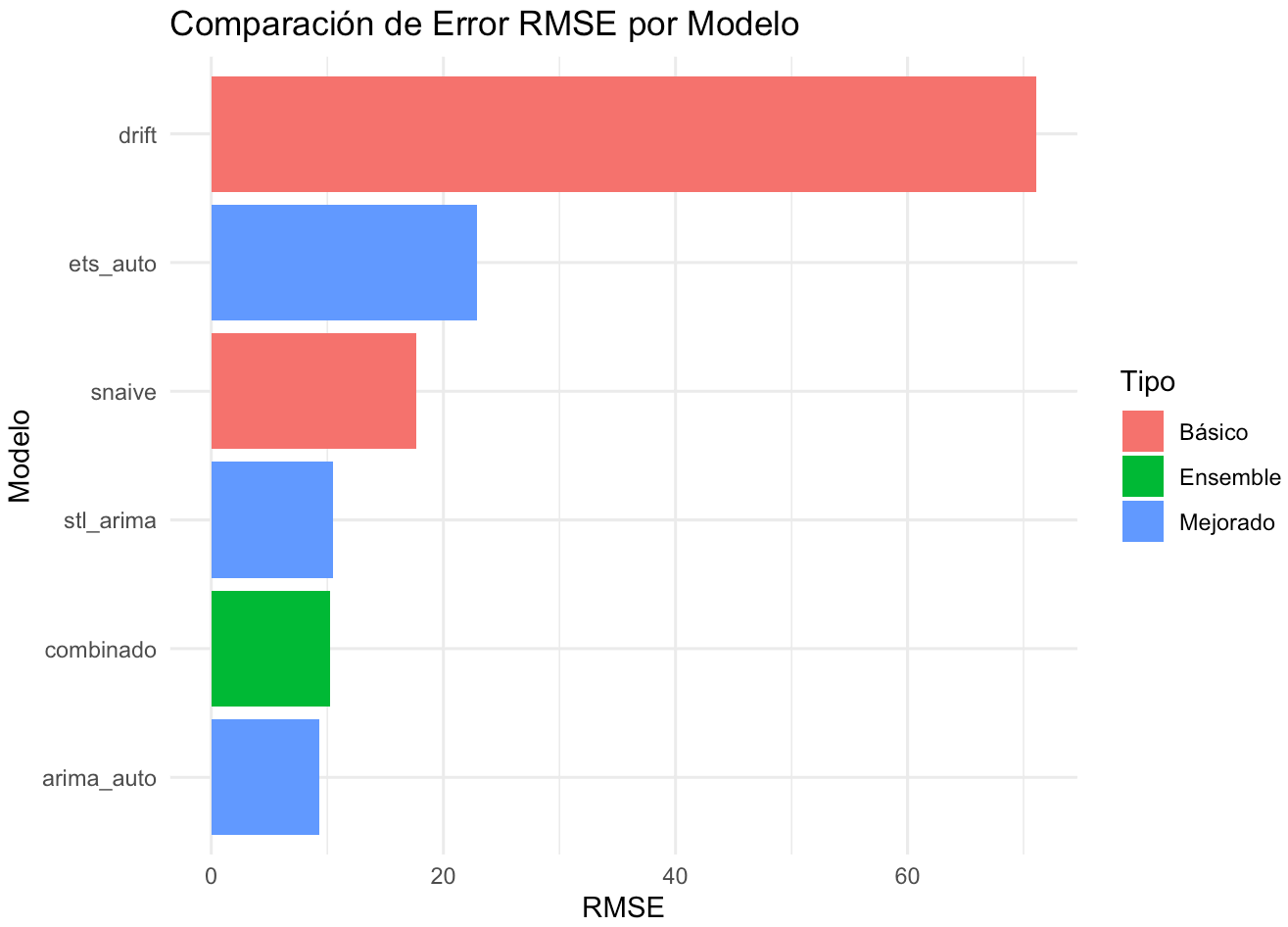
```
============================================================
```

```
comparacion_final <- bind_rows(
  accuracy_results |>
    filter(.model %in% c("drift", "snaive")) |>
    mutate(tipo = "Básico"),
  accuracy_results |>
    filter(.model %in% c("ets_auto", "arima_auto", "stl_arima")) |>
    mutate(tipo = "Mejorado"),
  accuracy_ensemble |>
    filter(.model == "combinado") |>
```

```
    mutate(tipo = "Ensemble")
)

print(comparacion_final |> select(tipo, .model, RMSE, MAE, MAPE))
```

```
# A tibble: 6 × 5
  tipo     .model      RMSE   MAE  MAPE
  <chr>    <chr>      <dbl> <dbl> <dbl>
1 Básico   snaive     17.7  15.6   7.13
2 Básico   drift      71.1  60.0  28.7
3 Mejorado arima_auto  9.28  7.63  3.57
4 Mejorado stl_arima  10.5   8.27  3.76
5 Mejorado ets_auto   22.9  21.0   9.62
6 Ensemble combinado  10.2   8.63  4.03
```

```
# Gráfico comparativo
comparacion_final |>
  ggplot(aes(x = reorder(.model, RMSE), y = RMSE, fill = tipo)) +
  geom_col() +
  coord_flip() +
  labs(
    title = "Comparación de Error RMSE por Modelo",
    x = "Modelo",
    y = "RMSE",
    fill = "Tipo"
  ) +
  theme_minimal()
```


Comparación de Error RMSE por Modelo
drift
ets_auto
snaive
stl_arima
combinado
arima_auto
Modelo
0
20
40
60
RMSE
Tipo
Básico
Ensemble
Mejorado

Conclusión