Forecasting Ecuador's Oil Production: Assessing the impact of halting exploitation in Block 43-ITT

Leines Nicole & Martinez Sayra

2025-04-25

Introduction

Ecuador's economy has been heavily reliant on oil exploitation for over five decades. As is shown in (**garcia-alban__good__2021?**) a result, the oil revenue is the most important driver of the national GDP.

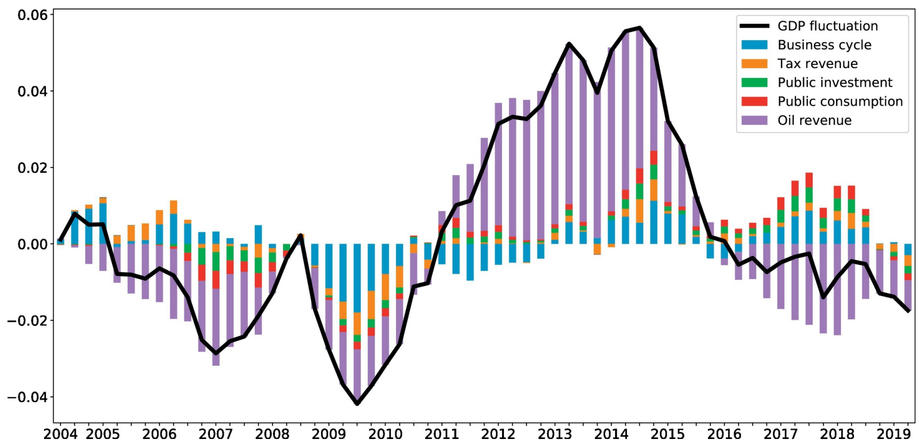


Figure 1: GDP fluctuations vs oil revenue between 2004-2019

Motivation

- The oil well known as Block 43-ITT is located within Ecuador's Yasuní National Park—one of the most biodiverse places on Earth and home to Indigenous communities (UNESCO, 2024).
- Oil exploitation in that well began in 2016 as part of efforts to boost fiscal revenues (Banco Central del Ecuador, 2023).
- In the 2023 national referendum, the Ecuadorian population voted to halt extraction in that well (Corte Consitutional del Ecuador, 2023).
- The decision was driven by the growing environmental and Indigenous rights movement and marked a significant shift in Ecuador's natural resource policy.

Relevance

The government is now responsible for phasing out extraction while addressing the economic implications—especially those related to oil production levels and public revenues. Evaluating how reduced production affects overall output is critical for policy and planning future decisions on resource management.

Objectives

- This final project aims to forecast oil production in Ecuador for the forthcoming years, following the halt of extraction in Block 43-ITT, which raises questions about future national income.

Dataset information

- Our dataset has monthly information from 2007-2024 for oil production:
  - Total and disaggregated for Block 43-ITT (from 2016 to 2023) and for the rest of the wells (data provided by the Government of Ecuador).
- Annual oil barrel production for 1972-2024 + 2025-2029 expected production (public data).
- WTI monthly prices for 2007-2024.

Analysis (Methods and Models)

- **Stage A** (Annual-Level Analysis):
  - We use an annual series (1972–2024) to analyze the long-run production trend.
- **Stage B** (Monthly-Level Analysis)
  - We use monthly dataset (2007–2024) for a more detailed (higher-frequency) forecast.
  - Additional variables:
    * Monthly WTI prices
    * Monthly block-level production of Block 43 ITT.
- **Stage C** (Scenario analysis)

  The idea is that if we trust the long-run historical trend from the annual model, we can ensure that the sum our monthly forecasts matches the trend predicted by the annual model.
  - **Baseline forecast:** assuming Block 43 ITT continues as historical.
  - **Shutdown Scenario:** set Block 43 ITT output to zero in 2024.

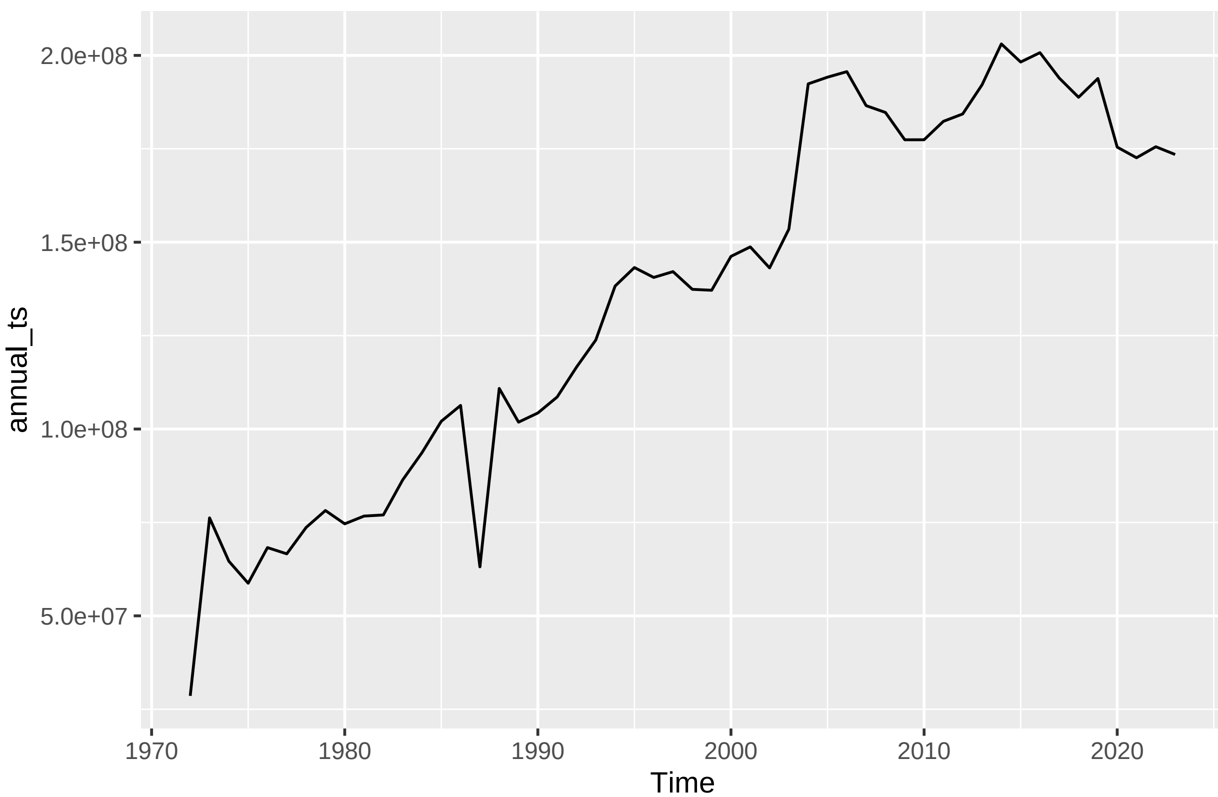
The difference in total production between the baseline and shutdown forecasts is the gap that other blocks must fill to maintain the same output level.

Stage A (Annual-Level Analysis):

We used an annual series (1972–2024) to analyze the long-run production trend.

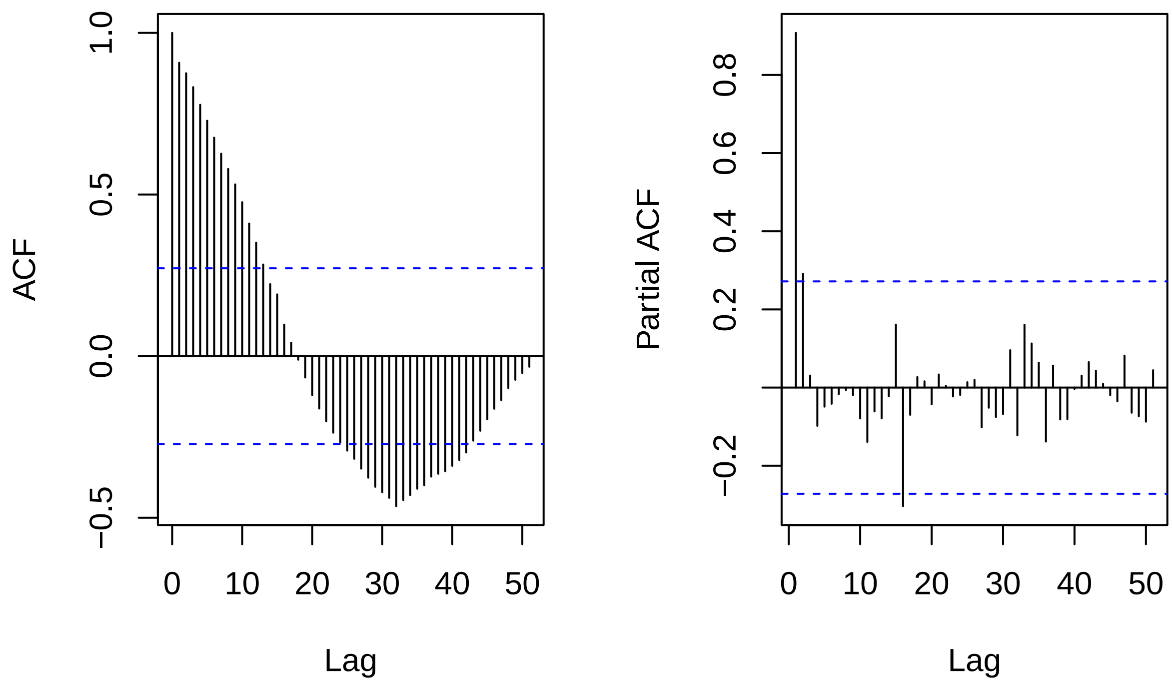
Annual Data The chart below illustrates the trajectory of Ecuador's annual oil output, which surged dramatically from the 1970s through the early 2000s. Following this period of rapid growth, production plateaued but remained substantially higher than pre-2000 levels. By the early 2020s, output had gradually declined to around 170 million barrels, possibly influenced by aging fields, constrained investment, the effects of the pandemic, or a combination of all.

The solely visualization may suggest that including data from before 2000 —when output was only a fraction of its subsequent levels— could distort our model's parameters. In contrast, restricting the sample to the period from 2000 onward, when production stabilized at its modern scale, is likely to yield a more accurate and relevant time series and forecasts. Considering this, analyzing the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) could provide valuable insights for determining the most appropriate research period, helping to identify patterns and lags in the data.



The sample ACF for the full series reveals strong autocorrelation extending up to approximately the 15 lag, beyond which the correlations sharply diminish, falling within the significance bounds for several years. This decline signals that the pre-2000 data may not exhibit meaningful memory. Similarly, the PACF presents a single significant spike at lag 1, which may suggest an AR(1) structure for the series.

From that information and given that pre-2000 output levels are an order of magnitude lower than post-2000 production and introduce disruptive long-lag noise, we confined our model to the 2000–2023 period, aiming at the model to gain precision and isolating the data's most relevant structural characteristics.



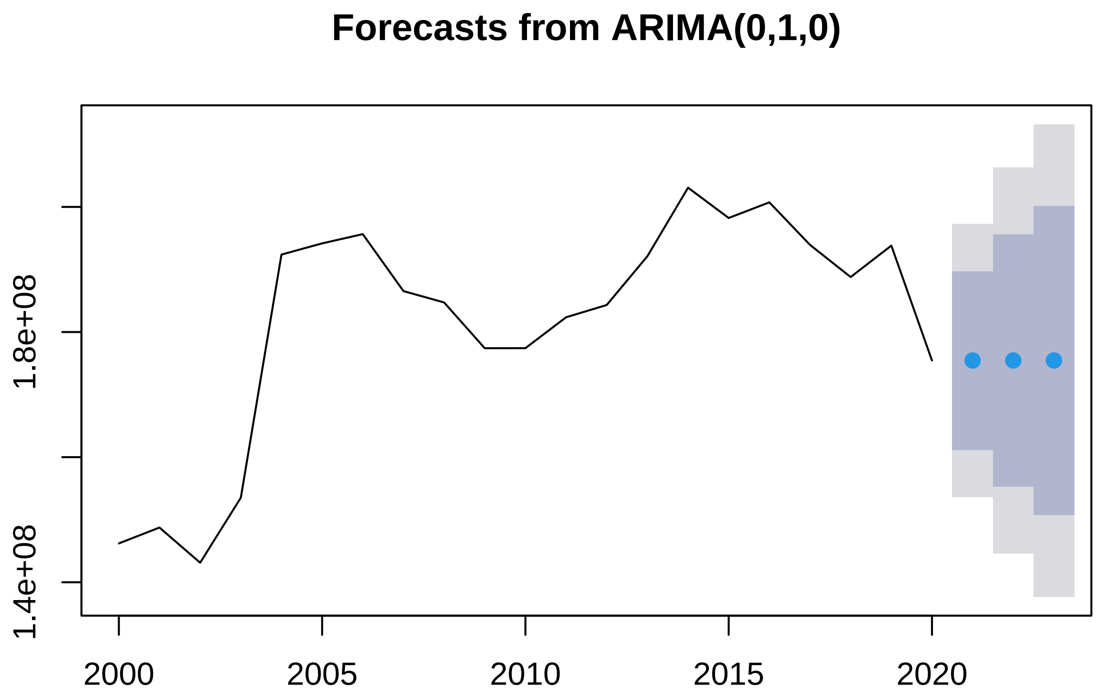
All the annual forecasting models were trained using data up to the year 2020. Because when using the pre-pandemic period, forecast performed poorly (see Annex).

Model 1: ARIMA The "auto.arima" in the training time series, suggests using the ARIMA(0,1,0) model captures the general trend of Ecuador's oil production over time but demonstrates moderate accuracy when handling the data's inherent volatility (See Table 1). With a mean absolute percent error (MAPE) of 0.94 (94% error) and RMSE of approximately 2 million units, the model's performance is acceptable but not exceptional. The forecast shows relatively stable future production levels, though the wide confidence intervals (gray bands) indicate substantial uncertainty in these predictions. The Theil's U value of 0.54 suggests that while the model outperforms naive forecasting approaches, there remains considerable room for improvement in capturing the time series' complex patterns and fluctuations.

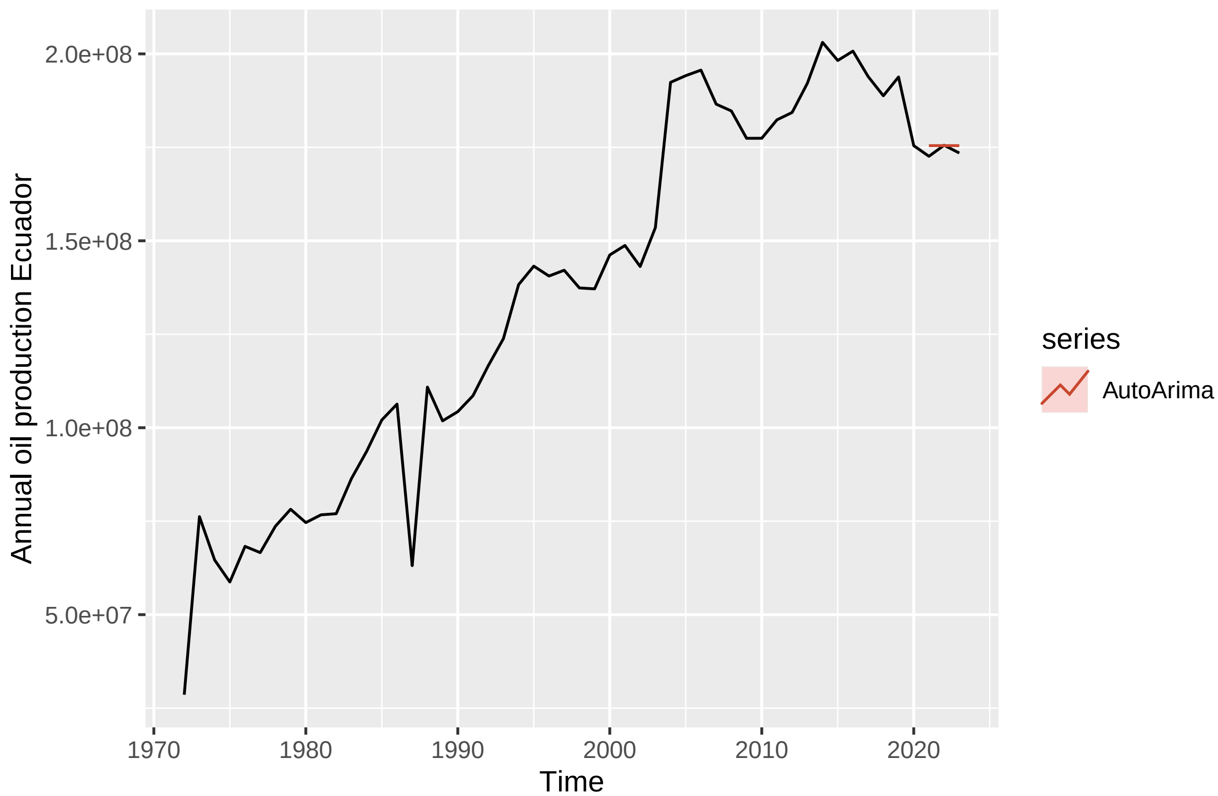
```r
#Model 1: ARIMA
# Fit an ARIMA model to the annual time series and forecast for 3 years
model_arima <- auto.arima(annual_ts_train)
forecast_arima <- forecast(model_arima, h = 3)
print(forecast_arima)
```

```
##      Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2021      175449722 161191369 189708074 153643453 197255990
## 2022      175449722 155285366 195614077 144611001 206288442
## 2023      175449722 150753530 200145913 137680157 213219286
```

```r
# Plot the forecast
plot(forecast_arima)
```




```r
#Plot model + observed data
autoplot(annual_ts) +
  autolayer(forecast_arima, series="AutoArima",PI=FALSE) +
  ylab("Annual oil production Ecuador")
```

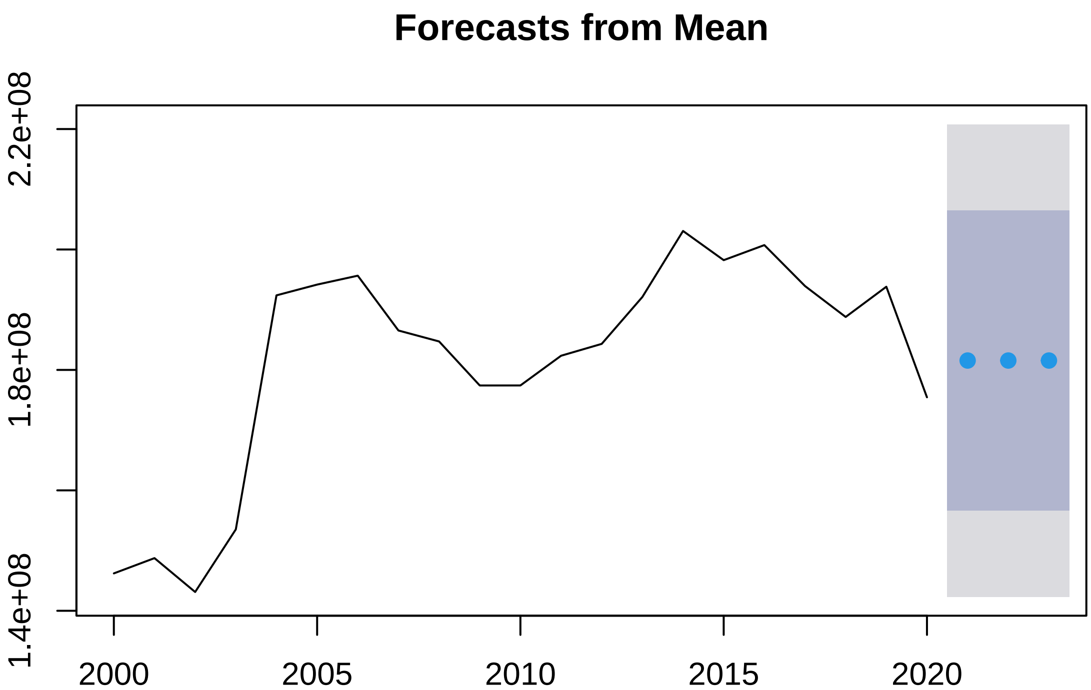


Testing Model 2: MEAN The Mean model employs a much simpler approach than ARIMA, that generates a flat forecast (blue dots) at approximately 181 million barrels with a wide confidence intervals, indicating high uncertainty. Besides, its performance metrics (see Table 1) reveal significant weaknesses, with a much higher RMSE (7,781,977) compared to ARIMA and a concerning MAPE of 4.42 (442% error). Moreover, according to the model's Theil's U value of 2.77 indicates it performs worse than naive forecasting methods, essentially failing to capture any of the time series' patterns or fluctuations.

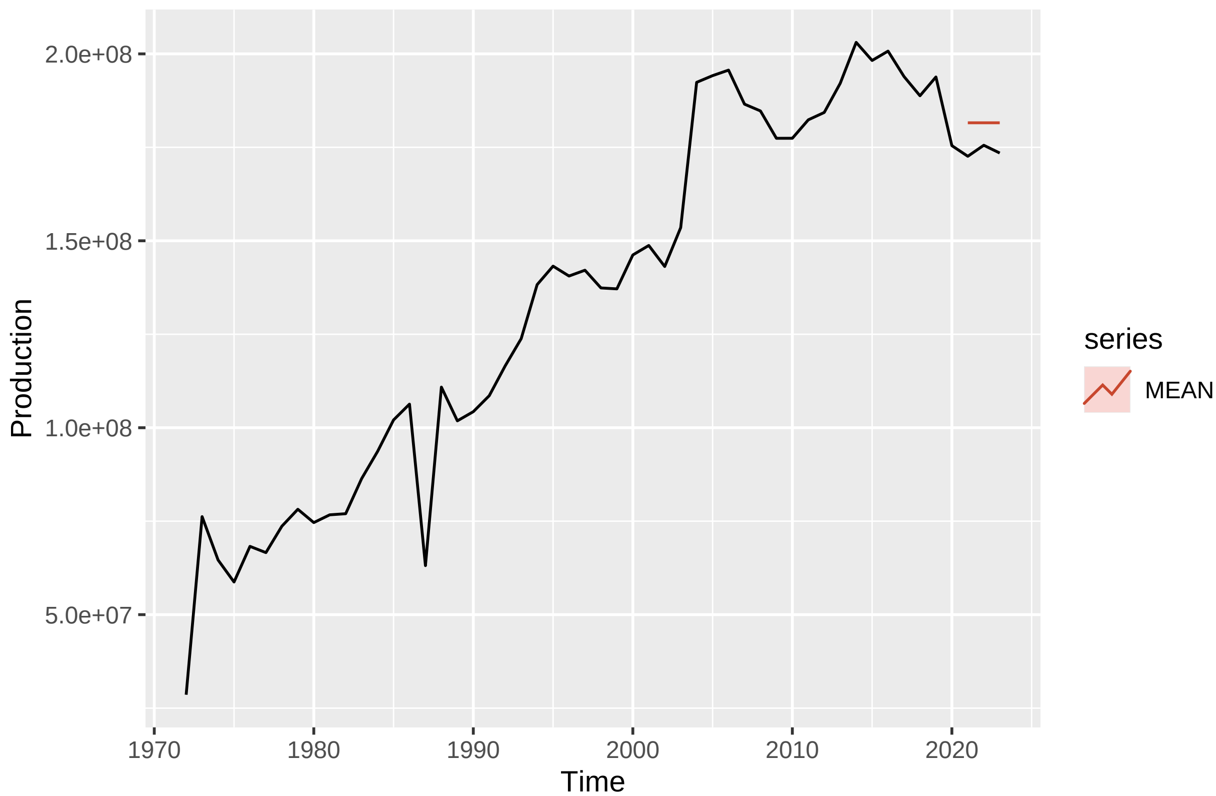
```
#Model 2: Arithmetic mean on original data
MEAN_seas <- meanf(y = annual_ts_train, h = 3)
print(MEAN_seas)
```

```
##      Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2021      181558473 156628140 206488806 142320439 220796506
## 2022      181558473 156628140 206488806 142320439 220796506
## 2023      181558473 156628140 206488806 142320439 220796506
```

```
plot(MEAN_seas)
```




```
autoplot(annual_ts) +
  autolayer(MEAN_seas, series="MEAN",PI=FALSE) +
  ylab("Production")
```



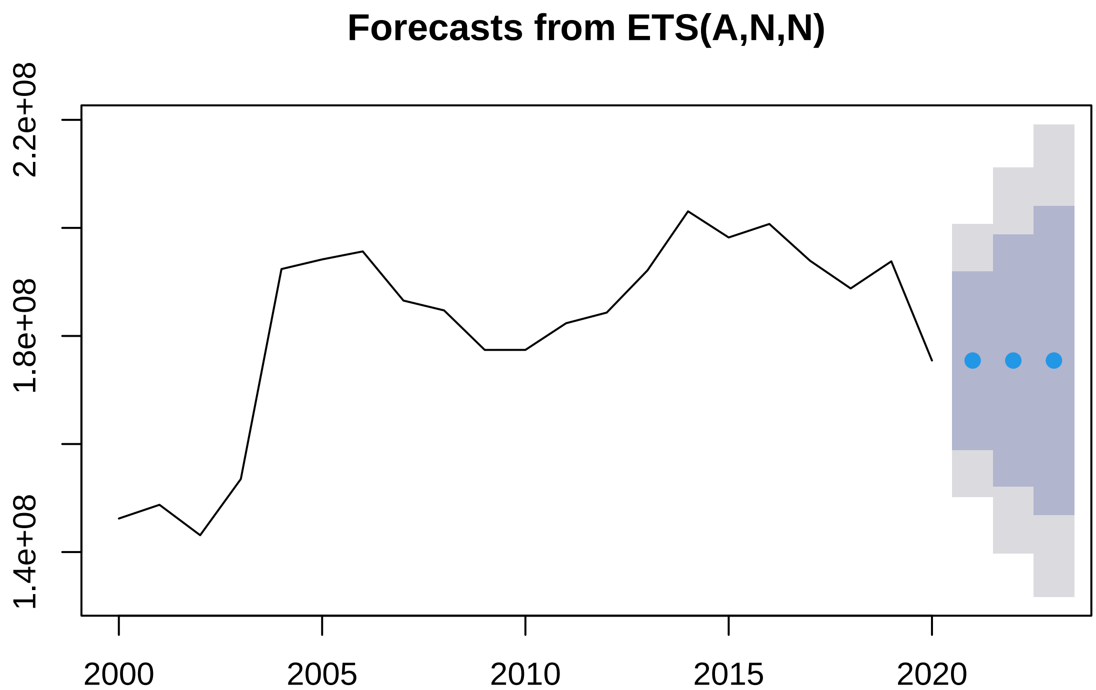
Testing Model 3: ETS The ETS model effectively "locks in" the most recent observed level (approximately 175 million barrels) and extrapolates it forward, producing a flat forecast line characterized by moderately narrow confidence bands. This tighter band of uncertainty, compared to the mean model's wider fan, reflects ETS's ability to adapt to the stable, modern production regime rather than being swayed by earlier, lower historical levels.

In-sample (see Table 1), the model under-forecasts by an average of 1.6 million barrels (ME), achieving a MAPE below 1 percent (around 0.95%). A Theil's U statistic of 0.54 confirms that it outperforms a naive "no-change" forecast. However, the pronounced negative autocorrelation at lag 1 indicates that the ETS model struggles to capture some of the smoother, year-over-year momentum inherent in the data.

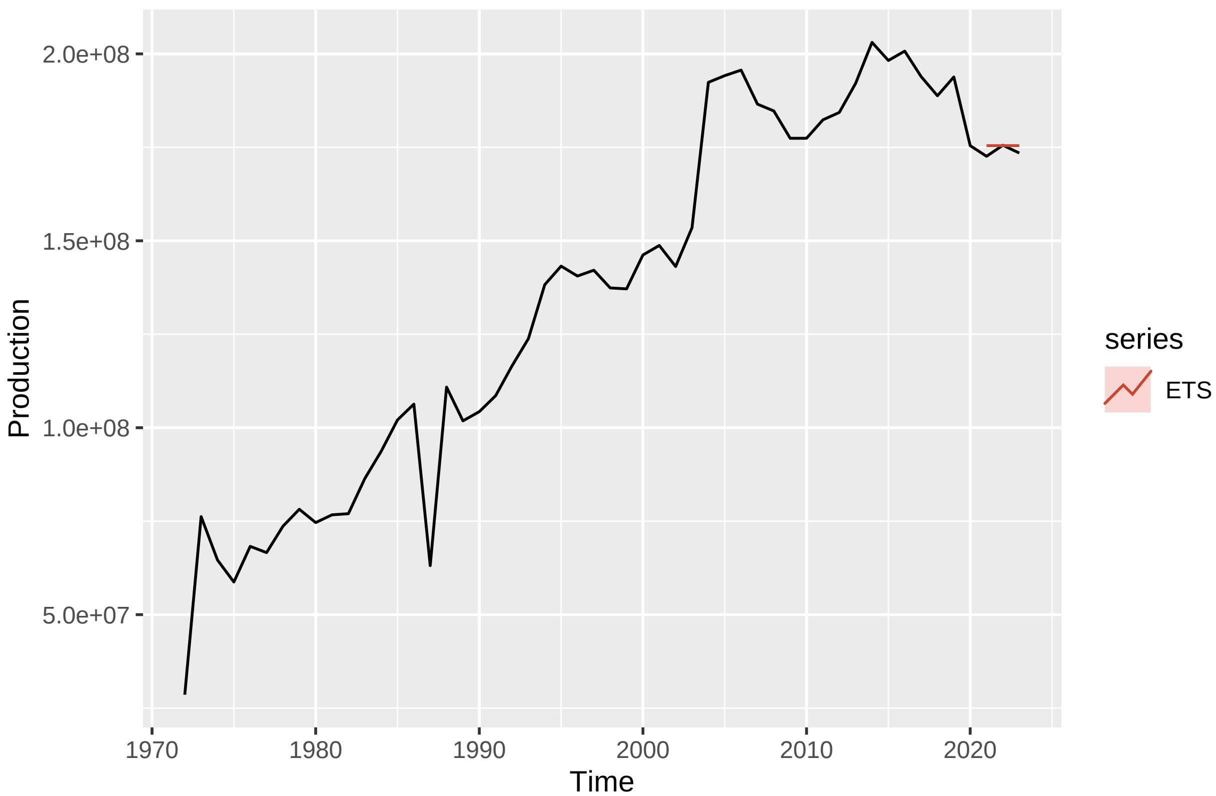
```r
# Model 3: ETS (Exponential Smoothing without seasonality)
model_ets <- ets(annual_ts_train)
forecast_ets <- forecast(model_ets, h = 3)
print(forecast_ets)
```

```
##      Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2021      175451620 158940493 191962746 150200030 200703209
## 2022      175451620 152102567 198800672 139742325 211160914
## 2023      175451620 146855480 204047760 131717598 219185642
```

```r
plot(forecast_ets)
```




```
autoplot(annual_ts) +
  autolayer(forecast_ets, series="ETS",PI=FALSE) +
  ylab("Production")
```

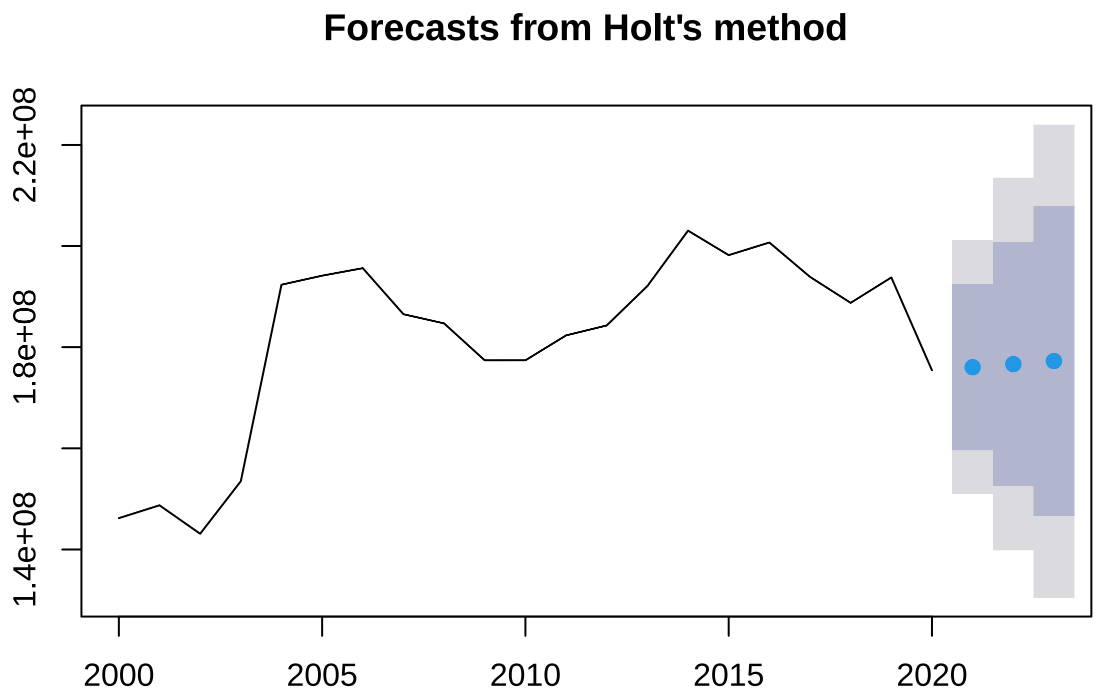


Testing Model 4: HOLT Holt's method augments simple exponential smoothing with a linear trend, and its forecast barely moves from the last observed level (around 175 million barrels), producing an almost flat-looking line with even wider uncertainty bands than ETS. It stands out that its Theil's U is 1.09, which would suggests it actually performs worse than a naïve method.

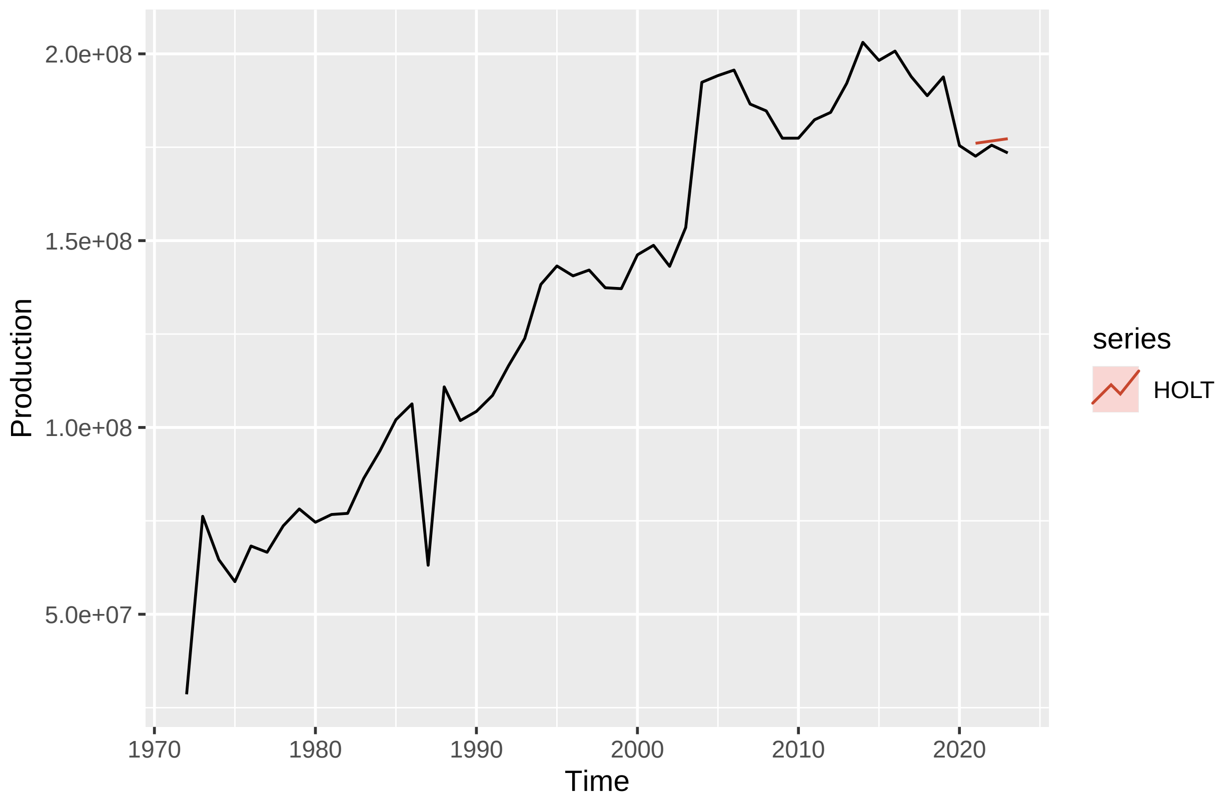
```
# Model 4: Holt's Linear Trend method
model_holt <- holt(annual_ts_train, h = 3)
forecast_holt <- forecast(model_holt, h = 3)
print(forecast_holt)
```

```
##      Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2021      176061114 159675163 192447065 151000965 201121263
## 2022      176670451 152596519 200744383 139852550 213488352
## 2023      177279788 146679865 207879711 130481244 224078332
```

```
plot(forecast_holt)
```




```r
autoplot(annual_ts) +
  autolayer(forecast_holt, series="HOLT",PI=FALSE) +
  ylab("Production")
```



Compare performance metrics of all models for the annual analysis

```
#Model 1: ARIMA
ARIMA_scores <- accuracy(forecast_arima$mean,ts_daily_test)  #store the performance metrics

#Model 2: Arithmetic mean
MEAN_scores <- accuracy(MEAN_seas$mean,ts_daily_test)

# Model 3:  ETS
ETS_scores <- accuracy(forecast_ets$mean,ts_daily_test)

# Model 4:  HOLT
HOLT_scores <- accuracy(forecast_holt$mean,ts_daily_test)

#create data frame
models_scores <- as.data.frame(rbind(ARIMA_scores, MEAN_scores,ETS_scores,HOLT_scores ))
row.names(models_scores) <- c("ARIMA", "MEAN","ETS","HOLT")
```

The following table compares the mentioned models accuracy, and shows how ARIMA beats the rest of the models, while ETS is the second best model

```
kbl(models_scores,
      caption = "Table 1. Forecast Accuracy for Annual Data",
      digits = array(5,ncol(models_scores))) %>%
```

Table 1: Table 1. Forecast Accuracy for Annual Data

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
ARIMA	-1574422	2001707	1640694	-0.91057	0.94832	-0.61118	0.54238
MEAN	-7683173	7781977	7683173	-4.42404	4.42404	-0.61118	2.77997
ETS	-1576320	2003200	1641327	-0.91166	0.94870	-0.61118	0.54288
HOLT	-2795151	3038681	2795151	-1.61209	1.61209	-0.65735	1.08959

```
  kable_styling(full_width = FALSE, position = "center") %>%
  #highlight model with lowest RMSE
  kable_styling(latex_options="striped", stripe_index = which.min(models_scores[,"RMSE"]))
```

```
#choose model with lowest RMSE
best_model_index <- which.min(models_scores[,"RMSE"])
cat("The best model by RMSE is:", row.names(models_scores[best_model_index,]))
```

```
## The best model by RMSE is: ARIMA
```

```
#choose model with lowest RMSE
best_model_index2 <- which.min(models_scores[,"MAPE"])
cat("The best model by MAPE is:", row.names(models_scores[best_model_index,]))
```

```
## The best model by MAPE is: ARIMA
```

Thus, we combined the two best models in aiming to have a more accurate model. By feeding the ETS errors into a simple AR(1), this hybrid forecast (red shading) sits almost exactly on today's production level (around 175 million barrels) and produces the tightest uncertainty "cone" of all models. In back-testing against 2021–2023 actuals (see Table 2), it under-forecasted by only 0.66 million barrels on average (ME around –0.66 m), cutting its RMSE from ~2.0 m (pure ETS or ARIMA) down to 1.17 m and halving the MAPE to 0.54 %. The dramatic drop in MAE (to 0.93 m) and MAPE shows that capturing the year-to-year autocorrelation in the residuals yields materially more accurate point forecasts, while the narrower fan reflects increased confidence in the short-term outlook.

```
# 1) Fit the base ETS
ets_fit <- ets(annual_ts_train)

# 2) Extract residuals and fit an AR(1) (no constant) to them
resid_ets <- residuals(ets_fit)
ar1_fit   <- Arima(resid_ets, order = c(1,0,0), include.mean = FALSE)

# 3) Forecast both models out h steps
h <- 3
ets_fc   <- forecast(ets_fit, h = h)
resid_fc <- forecast(ar1_fit, h = h)

# 4) Combine the forecasts
hybrid_fc        <- ets_fc
#colnames(hybrid_fc$lower)
#colnames(hybrid_fc$upper)
hybrid_fc$mean   <- ets_fc$mean   + resid_fc$mean
```

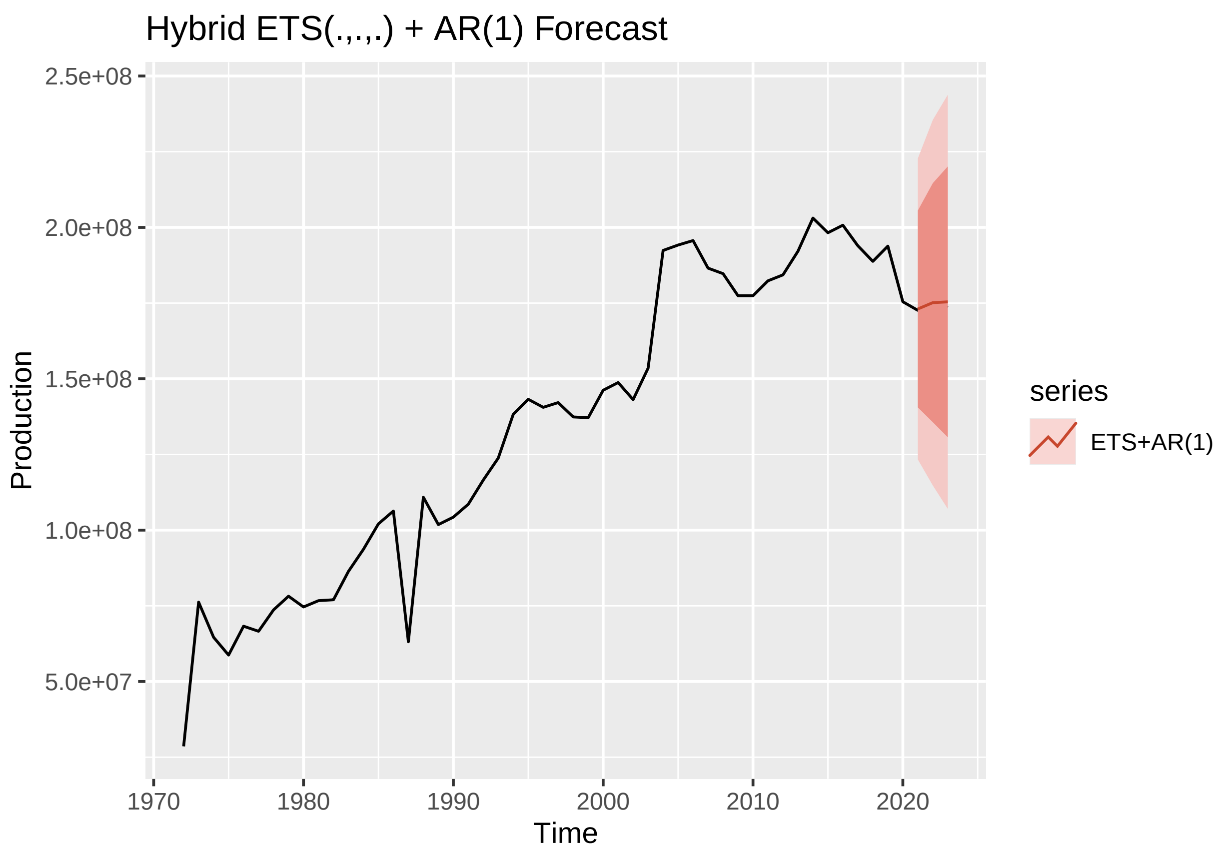
```r
hybrid_fc$lower  <- ets_fc$lower  + resid_fc$lower
hybrid_fc$upper  <- ets_fc$upper  + resid_fc$upper

# 5) Or extract a neat table of point-forecasts + 95% intervals:
print(colnames(hybrid_fc$lower))  # e.g. "80%" or c("80%", "95%")
```

```
## [1] "ets_fc$lower.80%" "ets_fc$lower.95%"
```

```r
# 6) Build a table by position
hybrid_df <- data.frame(
  Year     = time(hybrid_fc$mean),
  Forecast = as.numeric(hybrid_fc$mean),
  Lo80     = hybrid_fc$lower[,1],
  Hi80     = hybrid_fc$upper[,1],
  Lo95     = if(ncol(hybrid_fc$lower)>=2) hybrid_fc$lower[,2] else NA,
  Hi95     = if(ncol(hybrid_fc$upper)>=2) hybrid_fc$upper[,2] else NA
)
print(hybrid_df)
```

```
##   Year  Forecast      Lo80      Hi80      Lo95      Hi95
## 1 2021 173051133 140553653 205548612 123350527 222751738
## 2 2022 175137867 135666488 214609245 114771603 235504131
## 3 2023 175410611 130689832 220131390 107016082 243805140
```

```r
# 6) Plot the result
autoplot(annual_ts) +
  autolayer(hybrid_fc, series="ETS+AR(1)", PI=TRUE) +
  ylab("Production") +
  ggtitle("Hybrid ETS(.,.,.) + AR(1) Forecast")
```




```r
# 1) Compute hybrid accuracy
Hyb_scores <- accuracy(hybrid_fc$mean,ts_daily_test)

# 1) bind all five score-objects into one data.frame
models_scores2 <- as.data.frame(rbind(
  ARIMA                     = ARIMA_scores,
  MEAN                      = MEAN_scores,
  ETS                       = ETS_scores,
  HOLT                      = HOLT_scores,
  `Hybrid ETS & AR(1)`      = Hyb_scores
))

# 2) (re)name the rows for display
rownames(models_scores2) <- c(
  "ARIMA", "MEAN", "ETS", "HOLT", "Hybrid ETS & AR(1)"
)

# 3) render the table; this must be the last expression in the chunk
models_scores2 %>%
  kbl(
    caption = "Table 2. Forecast Accuracy for Annual Data",
    digits  = array(5, ncol(models_scores2)),
    row.names = TRUE
  ) %>%
  kable_styling(full_width = FALSE, position = "center") %>%
  kable_styling(
```

Table 2: Table 2. Forecast Accuracy for Annual Data

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
ARIMA	-1574422.0	2001707	1640693.8	-0.91057	0.94832	-0.61118	0.54238
MEAN	-7683173.0	7781977	7683173.0	-4.42404	4.42404	-0.61118	2.77997
ETS	-1576320.2	2003200	1641326.5	-0.91166	0.94870	-0.61118	0.54288
HOLT	-2795151.5	3038681	2795151.5	-1.61209	1.61209	-0.65735	1.08959
Hybrid ETS & AR(1)	-657903.9	1171499	932078.9	-0.38062	0.53680	-0.40555	0.54320

```
    latex_options = "striped",
    stripe_index  = which.min(models_scores2$RMSE)
  )
```

```
print(models_scores2)
```

```
##                             ME    RMSE       MAE        MPE      MAPE       ACF1
## ARIMA               -1574422.0 2001707 1640693.8 -0.9105732 0.9483243 -0.6111825
## MEAN                -7683173.0 7781977 7683173.0 -4.4240445 4.4240445 -0.6111825
## ETS                 -1576320.2 2003200 1641326.5 -0.9116649 0.9486952 -0.6111825
## HOLT                -2795151.5 3038681 2795151.5 -1.6120878 1.6120878 -0.6573494
## Hybrid ETS & AR(1)   -657903.9 1171499  932078.9 -0.3806204 0.5368018 -0.4055451
##                     Theil's U
## ARIMA               0.5423828
## MEAN                2.7799717
## ETS                 0.5428761
## HOLT                1.0895856
## Hybrid ETS & AR(1) 0.5432020
```

```
# 4) now print out which model is best by RMSE and MAPE
best_rmse <- rownames(models_scores2)[which.min(models_scores2$RMSE)]
best_mape <- rownames(models_scores2)[which.min(models_scores2$MAPE)]

cat("The best model by RMSE is:", best_rmse, "\n")
```

```
## The best model by RMSE is: Hybrid ETS & AR(1)
```

```
cat("The best model by MAPE is:", best_mape, "\n")
```

```
## The best model by MAPE is: Hybrid ETS & AR(1)
```

Now we use the hybrid model for our data from 2000 to 2023. This model captured the long-term level and then added an AR(1) on its one-step residuals to restore the small year-to-year momentum that pure ETS missed. The outcome is a flat forecast of about 173 million barrels per year from 2024 through 2027, with an 80 % confidence band narrowing to roughly 128–219 million and a 95 % band of 103–244 million barrels.

```
# Filter the original from 2000 to 2023
annual_ts_2023 <- window(annual_ts, start = c(2000, 1), end = c(2023, 1))

# 1) Fit the base ETS on 2000-2023
```

```r
ets_fit2   <- ets(annual_ts_2023)

# 2) Extract one-step residuals and fit AR(1) to them
resid_ets2 <- residuals(ets_fit2)
ar1_fit2   <- Arima(resid_ets2, order=c(1,0,0), include.mean=FALSE)

# 3) Forecast each component h years ahead
h2 <- 4
ets_fc2   <- forecast(ets_fit2,   h = h2, level = c(80, 95))
resid_fc2 <- forecast(ar1_fit2,   h = h2, level = c(80, 95))
colnames(ets_fc2$lower)
```

```
## [1] "80%" "95%"
```

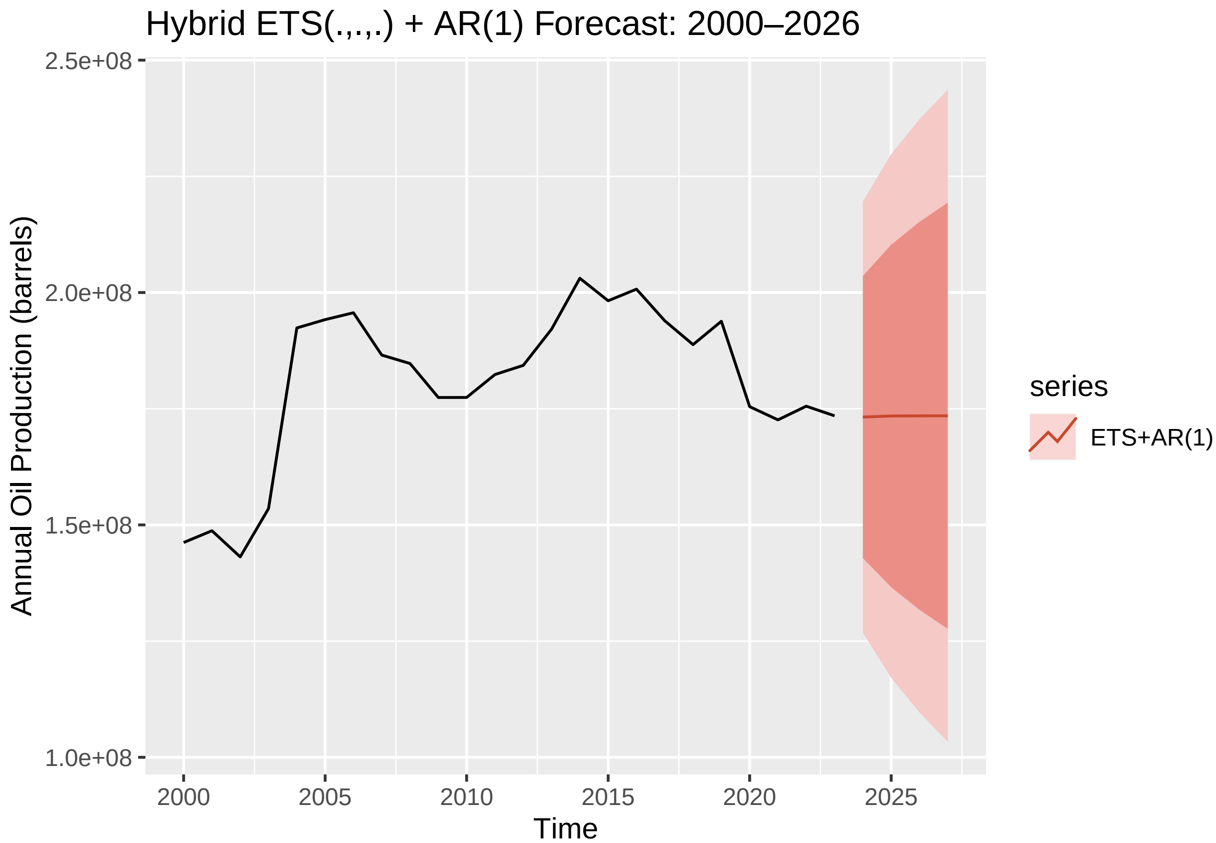
```r
colnames(resid_fc2$lower)
```

```
## [1] "80%" "95%"
```

```r
# 4) Build the hybrid forecast object
hybrid_fc2        <- ets_fc2
hybrid_fc2$mean   <- ets_fc2$mean   + resid_fc2$mean
hybrid_fc2$lower  <- ets_fc2$lower  + resid_fc2$lower
hybrid_fc2$upper  <- ets_fc2$upper  + resid_fc2$upper

# 5) Print the 80% and 95% intervals
hybrid_df2 <- data.frame(
  Year     = time(hybrid_fc2$mean),
  Forecast = as.numeric(hybrid_fc2$mean),
  Lo80     = hybrid_fc2$lower[, 1],  # first column = 80%
  Hi80     = hybrid_fc2$upper[, 1],
  Lo95     = hybrid_fc2$lower[, 2],  # second column = 95%
  Hi95     = hybrid_fc2$upper[, 2]
)
print(hybrid_df2)
```

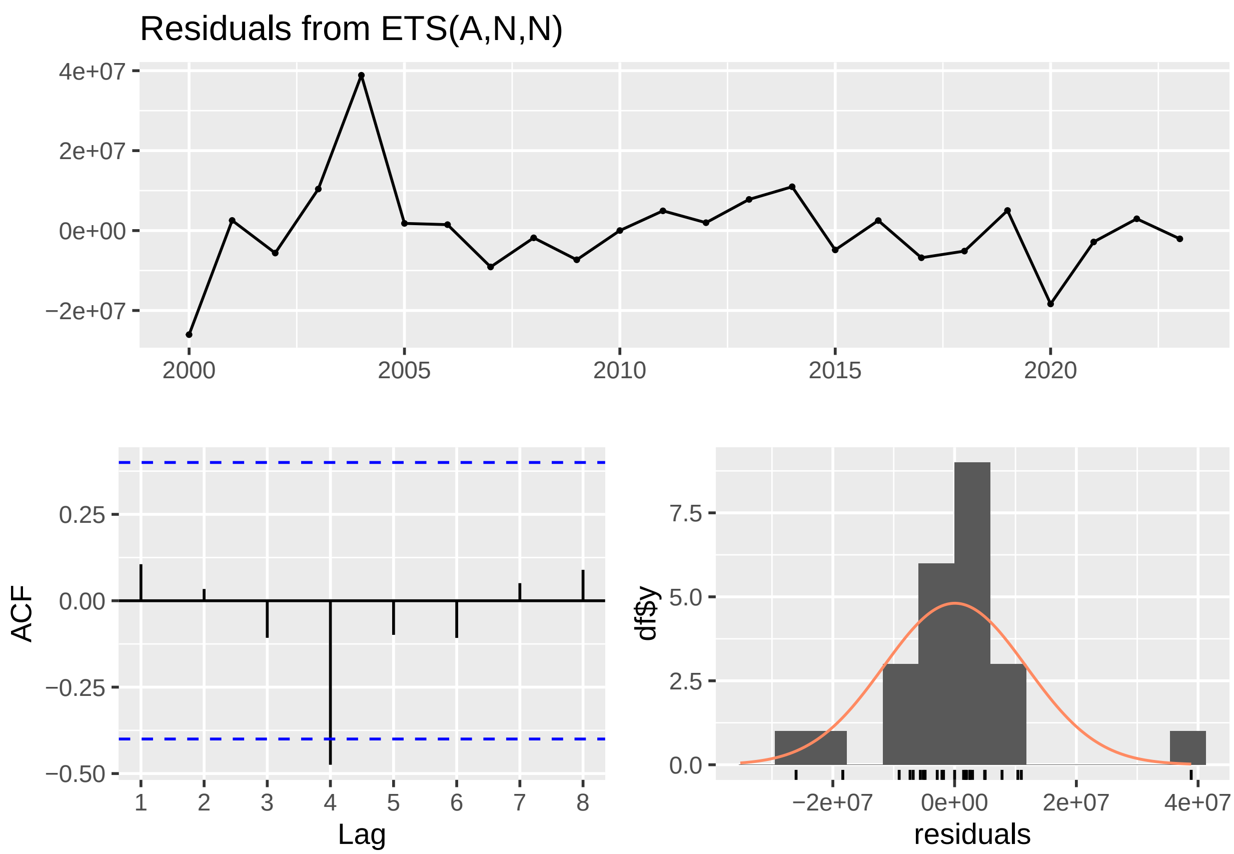
```
##   Year  Forecast      Lo80      Hi80      Lo95      Hi95
## 1 2024 173209118 142864921 203553314 126801674 219616561
## 2 2025 173441171 136598394 210283947 117095006 229787335
## 3 2026 173470963 131733639 215208287 109639234 237302692
## 4 2027 173474788 127612784 219336792 103334906 243614670
```

```r
# 6) Plot: historical 2000-2023 + 2024-2026 hybrid forecast
autoplot(annual_ts_2023) +
  autolayer(hybrid_fc2, series="ETS+AR(1)", PI=TRUE) +
  ylab("Annual Oil Production (barrels)") +
  ggtitle("Hybrid ETS(.,.,.) + AR(1) Forecast: 2000-2026")
```



The residuals fluctuate randomly around zero with no obvious drift or changing variance, and—aside from a single large error in the mid-2000s—stay within about ±20 million barrels. Moreover, the ACF shows all lags inside the 95 % confidence bounds (lag 4 is barely crossing the bounds, but we would say there is no meaningful serial correlation). The histogram of errors looks symmetric (with slightly tails from that outlier). In brief, they behave like white noise, suggesting our hybrid ETS+AR(1) captured the main dynamics of Ecuador's oil-production series.

```
checkresiduals(hybrid_fc2)
```

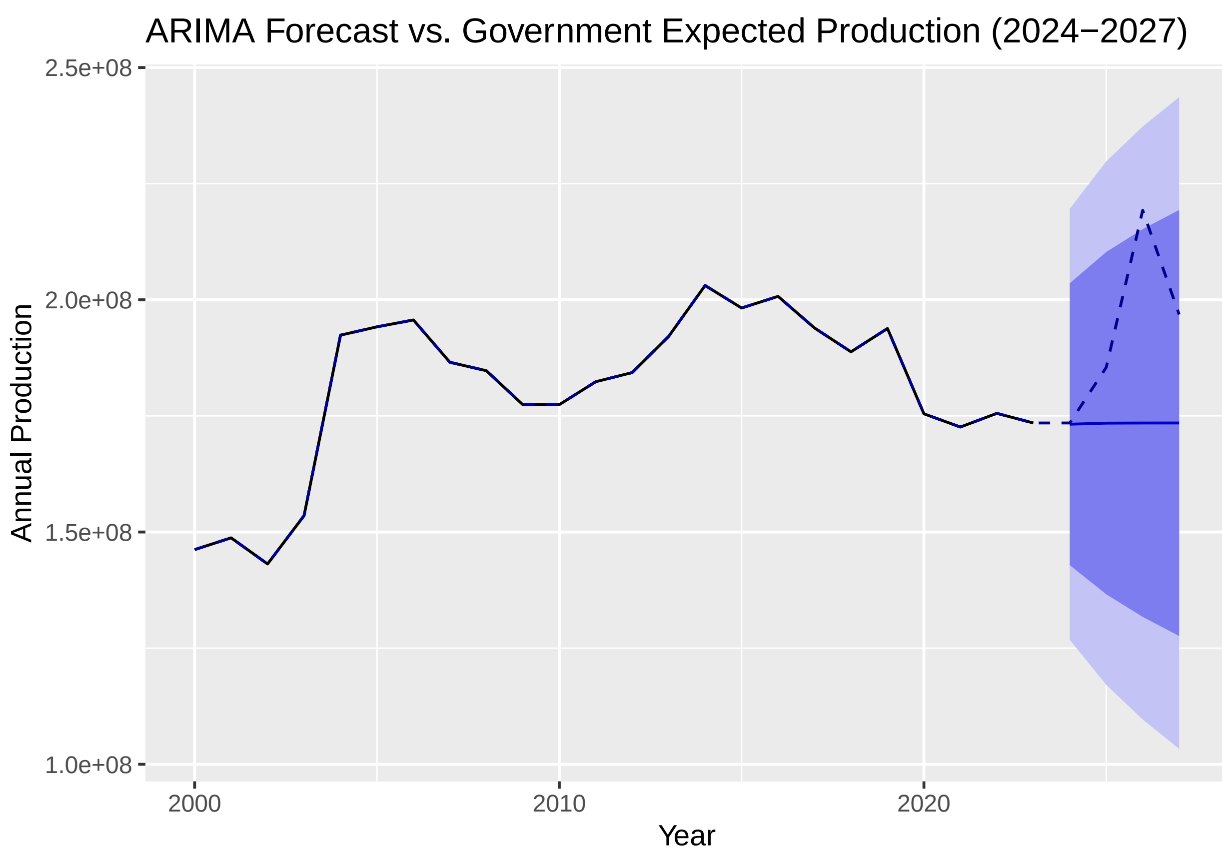



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 8.0225, df = 5, p-value = 0.155
##
## Model df: 0.   Total lags used: 5
```

Finally, we observed that Ecuador's projected a higher production for 2026 & 2027, however, there was no information on the additional data they used for their forecasting. However it is worth noting that projections for 2026 would be historic volumes as is slightly above annual production in previous years.

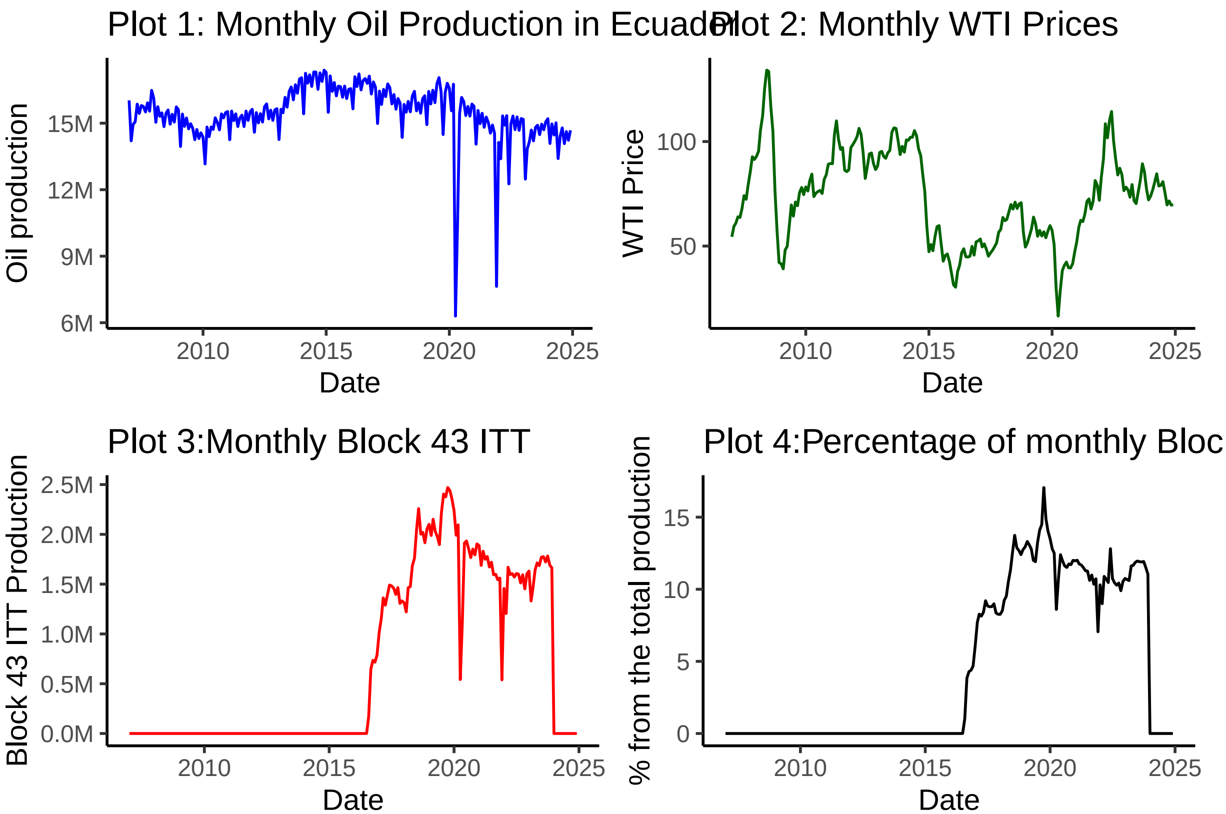
```
# 1. Filter existing data from 2000 to 2023
expected_production <- annual_data_72_2023 %>%
  filter(year >= 2000, year <= 2023)

# 2. Create a data frame for 2024-2027: data from https://www.primicias.ec/economia/plan-hidrocarburife
daily_values <- c(475.27, 508.09, 600.72, 539.252)  # daily production according to Ecuador's gov
future_years <- 2024:2027

df_future <- data.frame(
  year = future_years,
  annual_production = daily_values * 1000 * 365
)
```

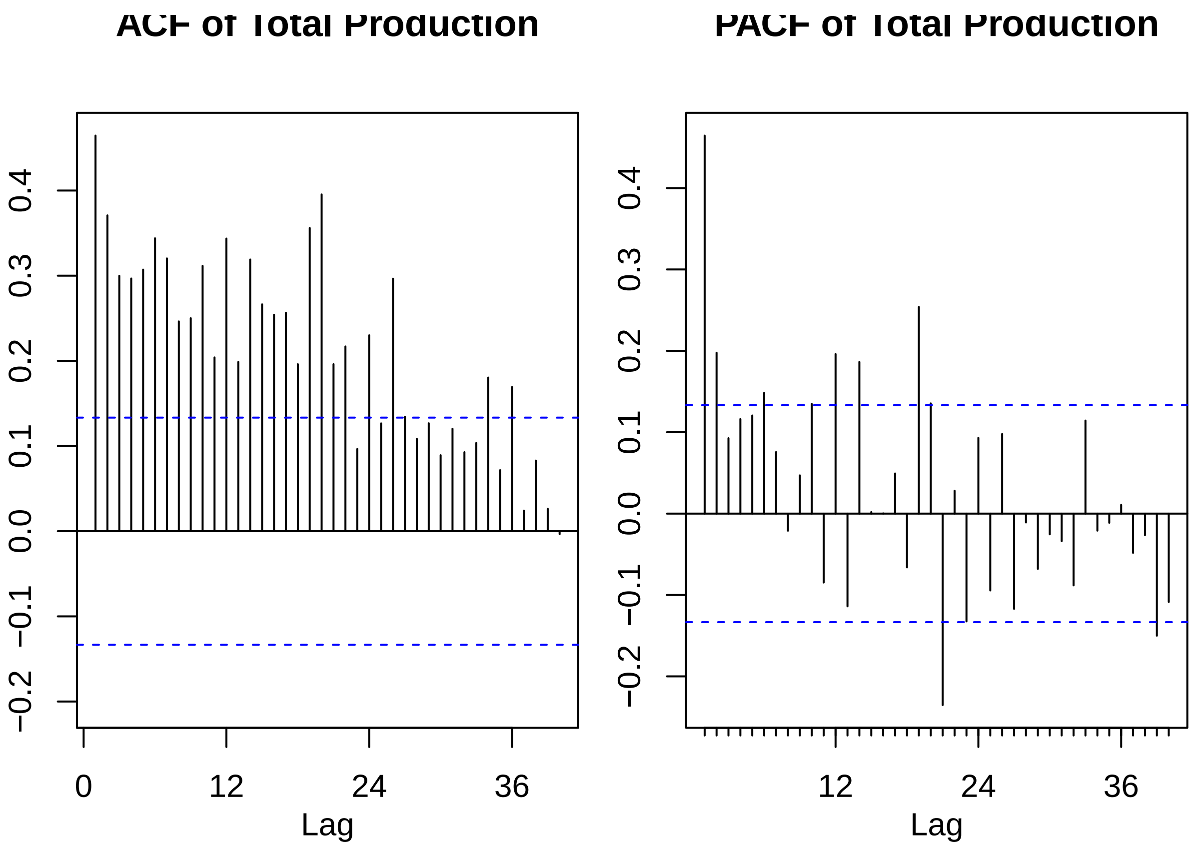
```r
# 3. Combine the filtered data with the new rows
expected_production <- bind_rows(expected_production, df_future)

annual_exp_ts <- ts(expected_production[,2],
                start = c(2000, 1),
                frequency = 1)


# 6. Plot the ARIMA forecast and government expected production together
autoplot(hybrid_fc2) +
  autolayer(annual_exp_ts, series = "Government Expected", linetype = "dashed", color = "darkblue") +
  xlab("Year") +
  ylab("Annual Production") +
  ggtitle("ARIMA Forecast vs. Government Expected Production (2024-2027)") +
  guides(colour = guide_legend(title = "Series"))
```

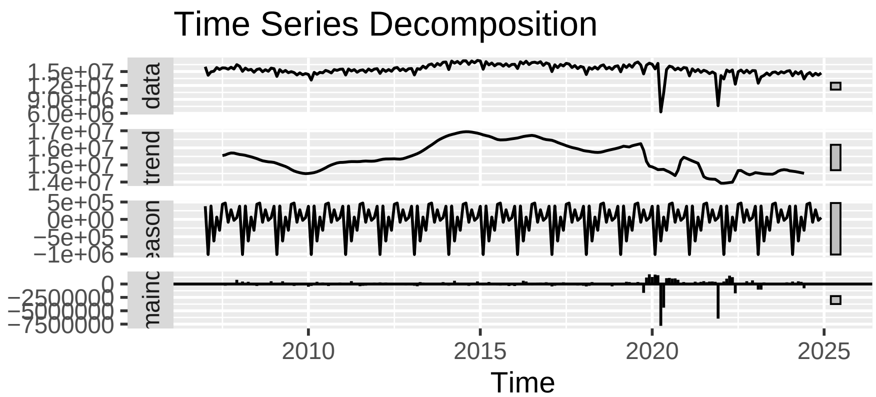


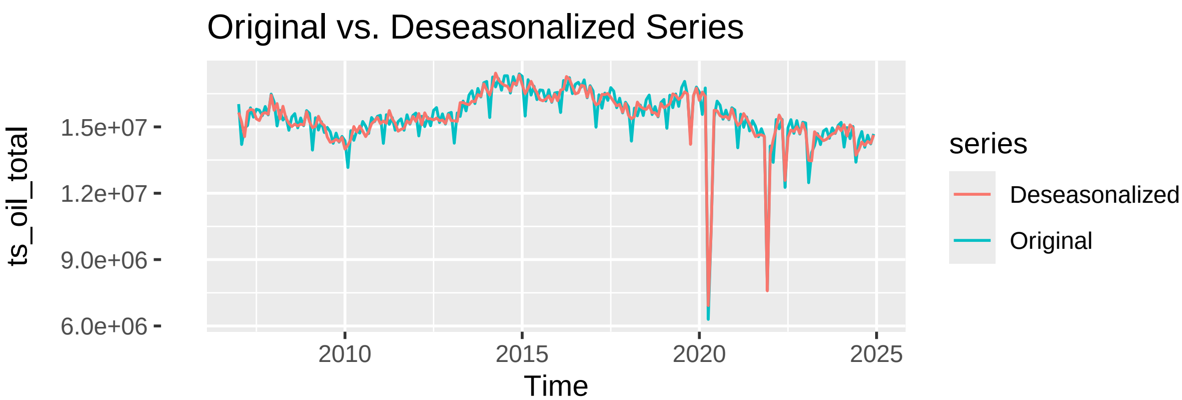
Stage B (Month-Level Analysis):

This is a more detailed monthly analysis from 2007–2024 using monthly WTI prices and Block 43 production.The following graphs shows oil production in Ecuador has been decreasing. Oil extraction in Block 43-ITT started in 2016 and has boosted the economy. Plot 4 shows that oil exploitation on Block 43-ITT has increased production from 2016 to 2023, reaching up to 17% of the total oil production.


Plot 1: Monthly Oil Production in Ecuador
Oil production
6M
9M
12M
15M
2010
2015
2020
2025
Date
Plot 2: Monthly WTI Prices
WTI Price
50
100
2010
2015
2020
2025
Date
Plot 3:Monthly Block 43 ITT
Block 43 ITT Production
0.0M
0.5M
1.0M
1.5M
2.0M
2.5M
2010
2015
2020
2025
Date
Plot 4:Percentage of monthly Bloc
% from the total production
0
5
10
15
2010
2015
2020
2025
Date


ACF of Total Production
PACF of Total Production
Lag
Lag
0
12
24
36
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4






```
##
##  Augmented Dickey-Fuller Test
##
## data:  deseasonal_prod
## Dickey-Fuller = -2.5608, Lag order = 5, p-value = 0.3407
## alternative hypothesis: stationary
```

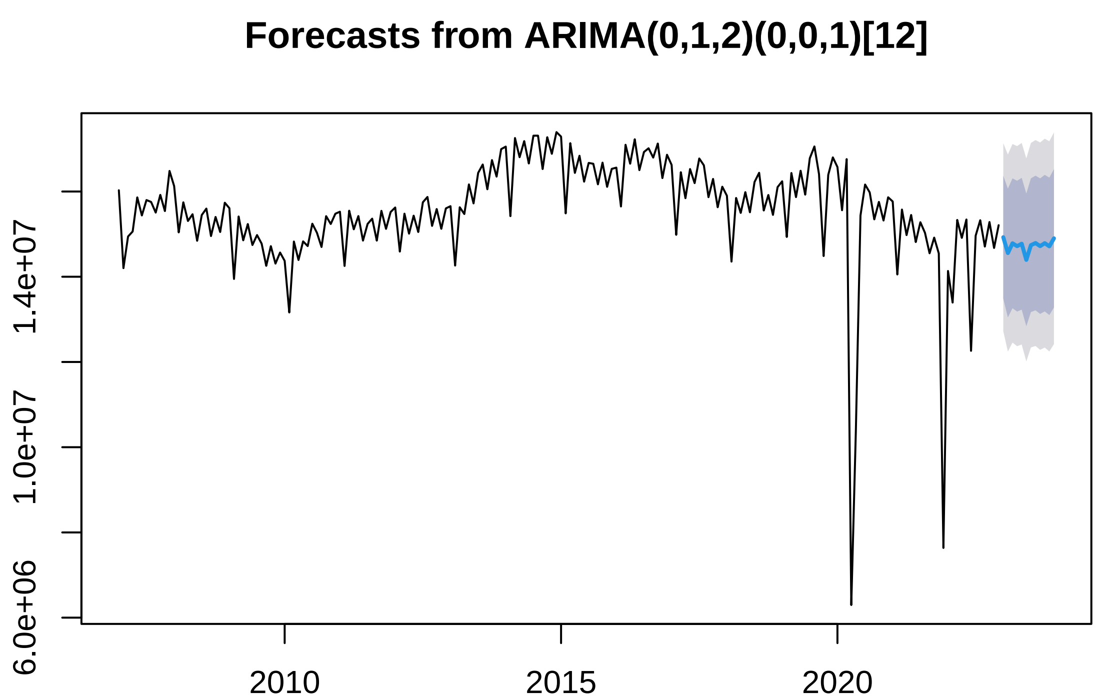
```
# Model A: Baseline SARIMA on total production
model_1_train <- auto.arima(ts_train_A, seasonal = TRUE)

# Forecast for Model A
forecast_1 <- forecast(model_1_train, h = h)

print(forecast_1)
```

Model 1

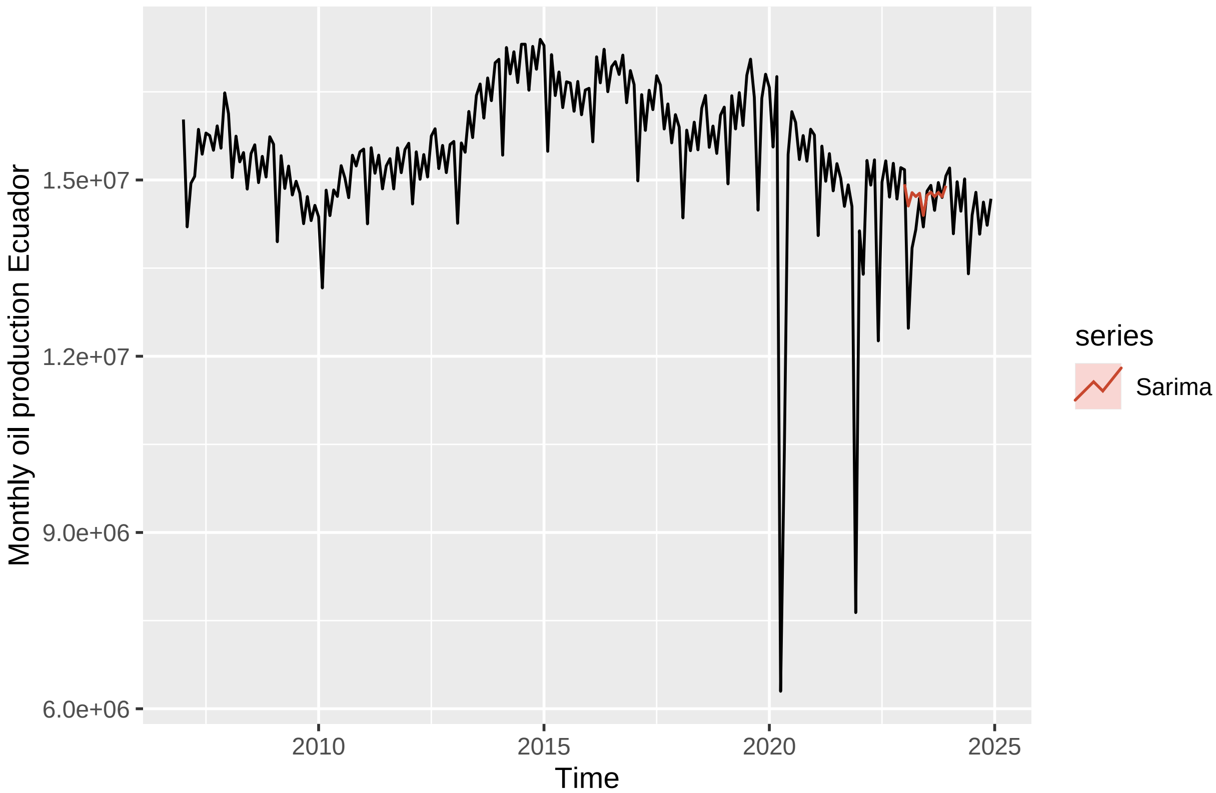
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2023       14928191 13486338 16370045 12723067 17133316
## Feb 2023       14556697 13045060 16068334 12244847 16868546
## Mar 2023       14785338 13262131 16308544 12455794 17114881
## Apr 2023       14717948 13183259 16252637 12370844 17065052
```

```
## May 2023       14774850 13228764 16320936 12410315 17139384
## Jun 2023       14395989 12838589 15953389 12014152 16777826
## Jul 2023       14740911 13172280 16309543 12341896 17139926
## Aug 2023       14791884 13212100 16371668 12375813 17207955
## Sep 2023       14718428 13127570 16309286 12285421 17151436
## Oct 2023       14788555 13186700 16390411 12338729 17238382
## Nov 2023       14714125 13101347 16326903 12247594 17180656
## Dec 2023       14901745 13278118 16525372 12418621 17384868
```

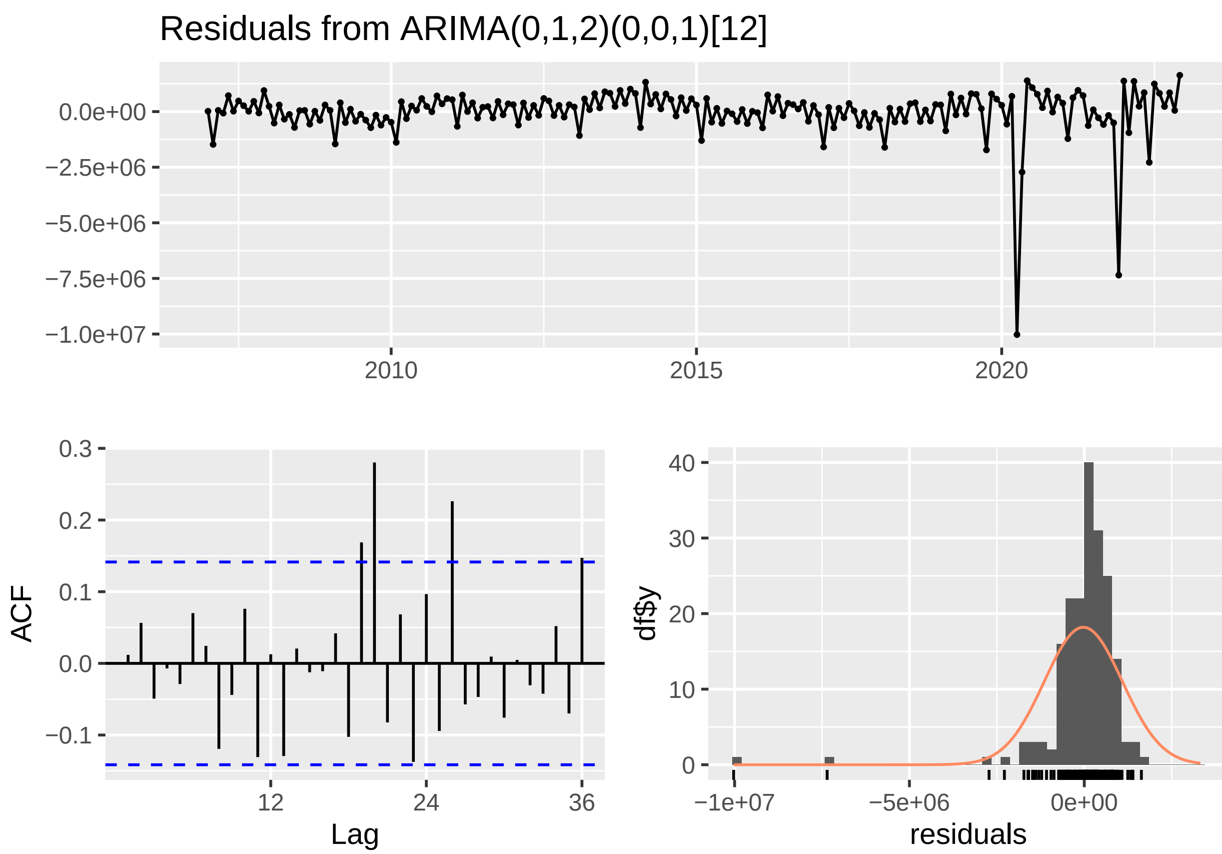
```
# Plot the forecast
plot(forecast_1)
```




```
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_1, series="Sarima",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```r
checkresiduals(model_1_train)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)(0,0,1)[12]
## Q* = 48.566, df = 21, p-value = 0.0005756
##
## Model df: 3.   Total lags used: 24
```

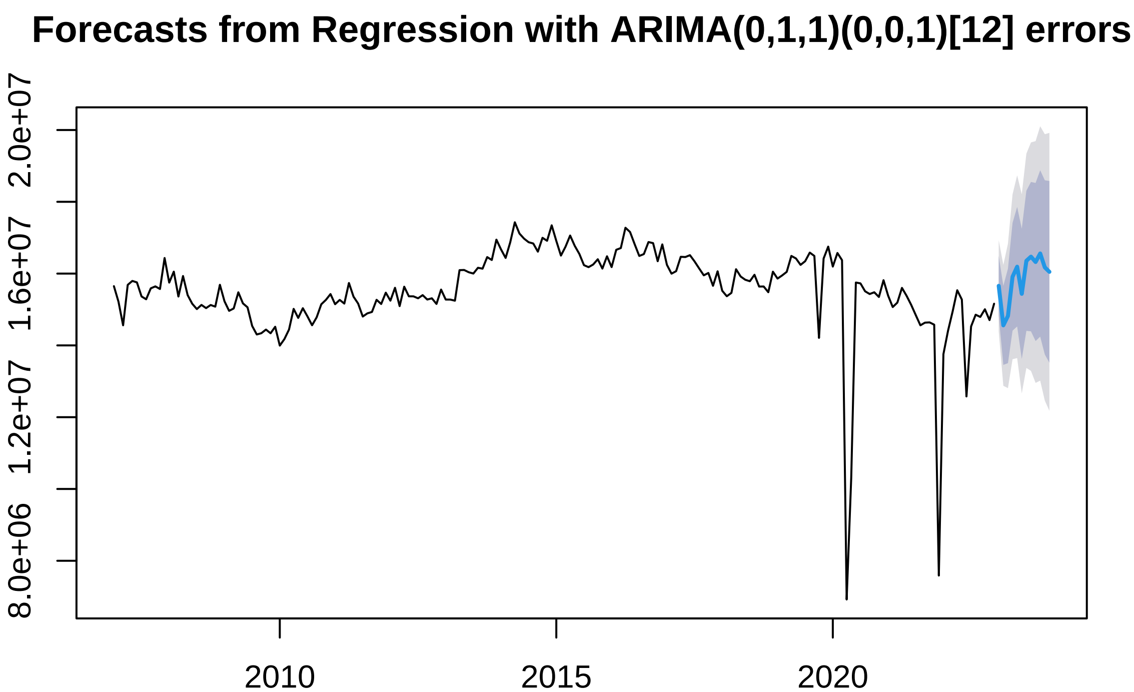
#Model 2:

```r
# Model B: SARIMAX on deseasonalized production with regressors
model_2_train <- auto.arima(ts_train_B, xreg = xreg_train, seasonal = TRUE)

# Forecast for Model B (with xreg)
forecast_2 <- forecast(model_2_train, xreg = xreg_test, h = h)

print(forecast_2)
```

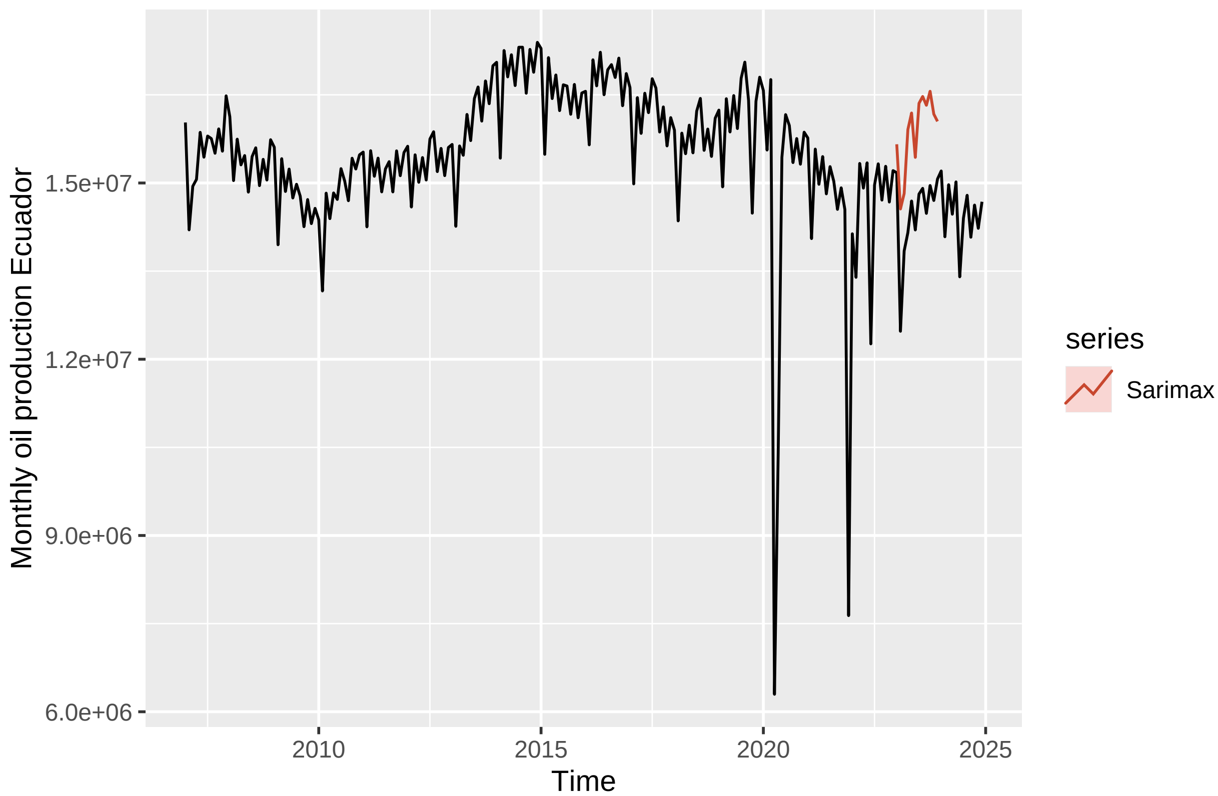
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2023       15659239 14827461 16491017 14387144 16931334
## Feb 2023       14557776 13457145 15658408 12874506 16241047
## Mar 2023       14821011 13505365 16136657 12808904 16833118
## Apr 2023       15909960 14409807 17410113 13615674 18204246
## May 2023       16191741 14527412 17856071 13646369 18737114
## Jun 2023       15436036 13622331 17249742 12662213 18209859
```

```
## Jul 2023       16356481 14404799 18308162 13371641 19341320
## Aug 2023       16472308 14391780 18552836 13290415 19654201
## Sep 2023       16322671 14120824 18524518 12955236 19690105
## Oct 2023       16561109 14244287 18877931 13017836 20104383
## Nov 2023       16172619 13746264 18598974 12461829 19883409
## Dec 2023       16048412 13517260 18579565 12177349 19919476
```

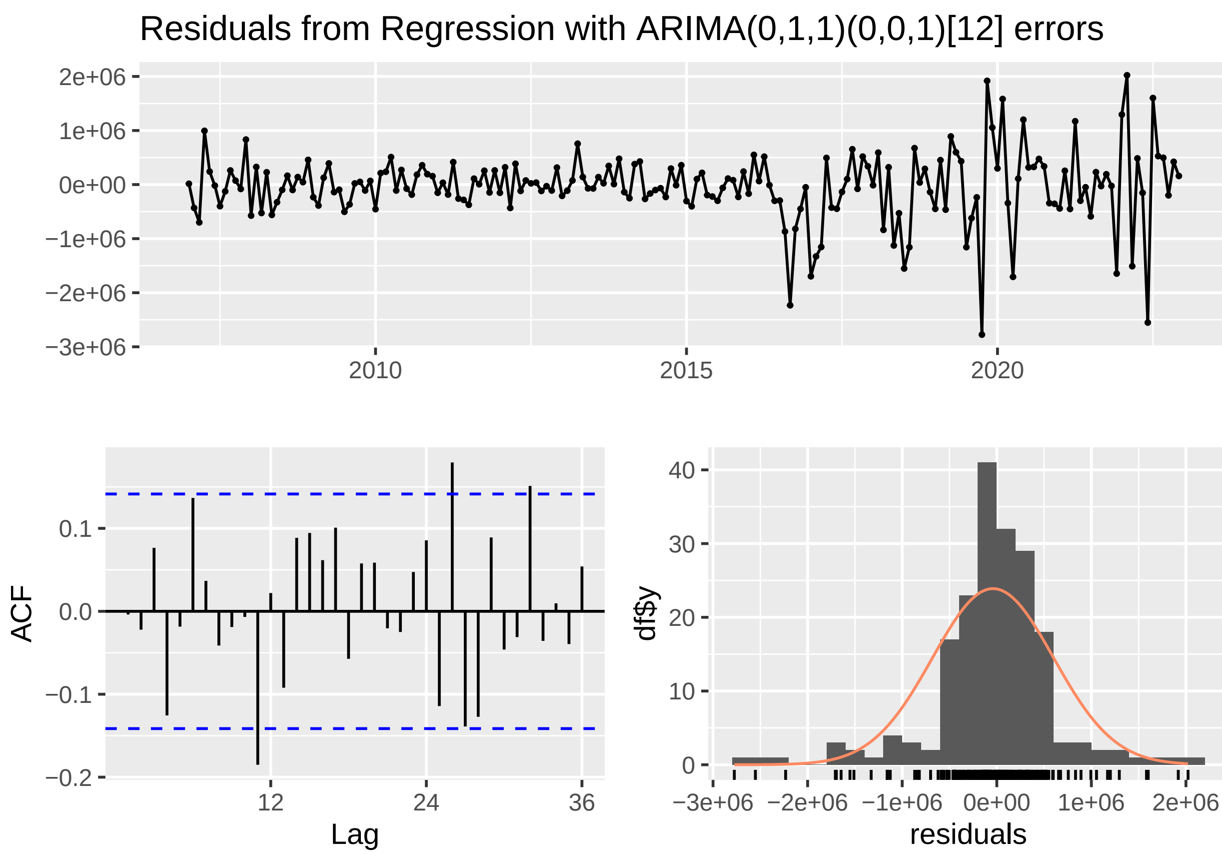
```r
# Plot the forecast
plot(forecast_2)
```




```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_2, series="Sarimax",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```
checkresiduals(model_2_train)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,1,1)(0,0,1)[12] errors
## Q* = 28.794, df = 22, p-value = 0.1508
##
## Model df: 2.   Total lags used: 24
```

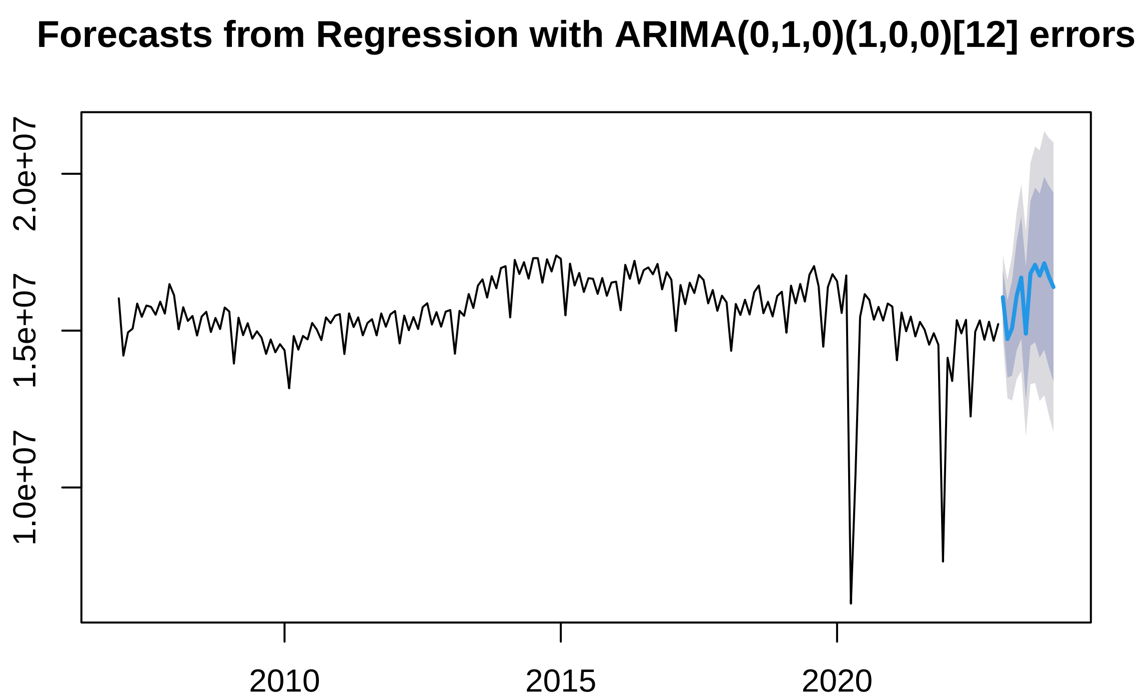
#Model 3

```
# Model 3: ARIMAX on deseasonalized production with regressors
model_3_train <- auto.arima(ts_train_A, xreg = xreg_train, seasonal = TRUE)

# Forecast for Model C (with xreg)
forecast_3 <- forecast(model_3_train, xreg = xreg_test, h = h)

print(forecast_3)
```

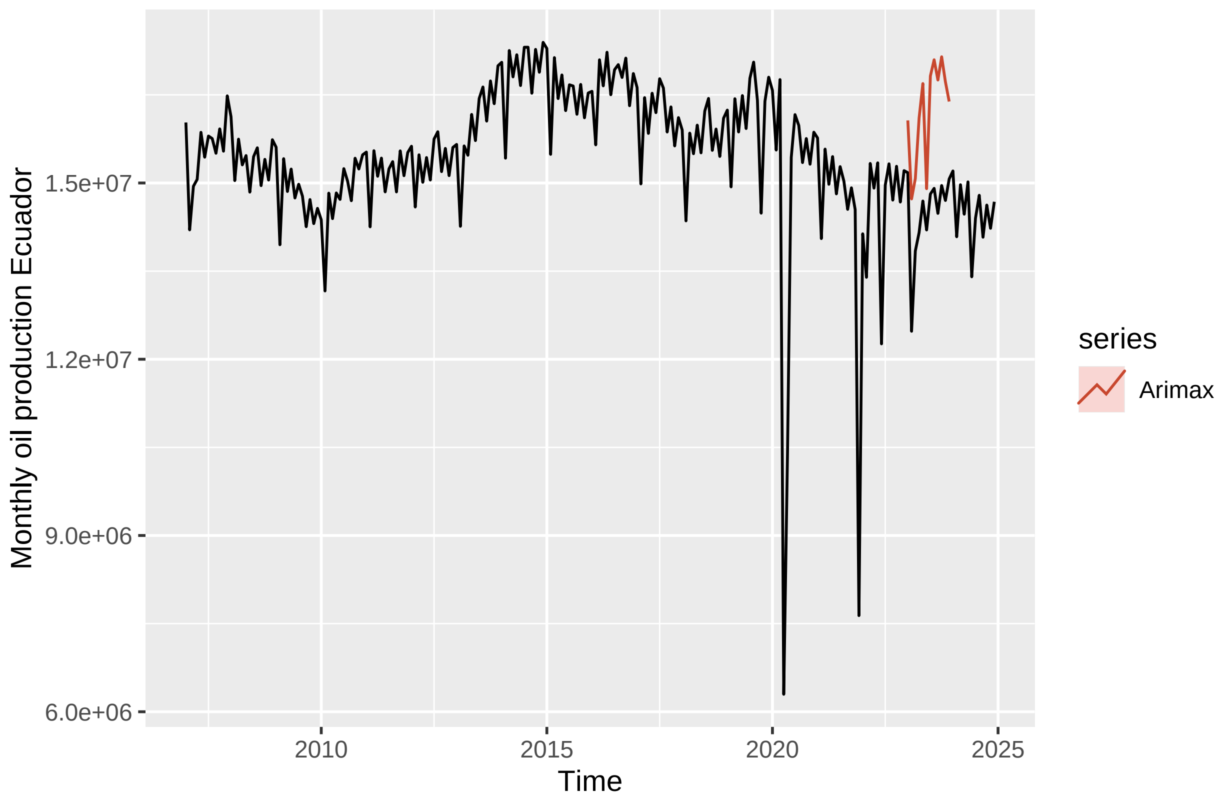
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2023       16065577 15194474 16936679 14733340 17397813
## Feb 2023       14728591 13496666 15960516 12844524 16612658
## Mar 2023       15077204 13568410 16585998 12769703 17384705
## Apr 2023       16106479 14364274 17848684 13442006 18770952
## May 2023       16692823 14744978 18640667 13713852 19671794
## Jun 2023       14902773 12769016 17036530 11639473 18166072
```

```
## Jul 2023       16821560 14516839 19126280 13296793 20346326
## Aug 2023       17097588 14633738 19561438 13329455 20865722
## Sep 2023       16752588 14139280 19365895 12755878 20749297
## Oct 2023       17148163 14393495 19902831 12935262 21361064
## Nov 2023       16722107 13832987 19611227 12303579 21140635
## Dec 2023       16386630 13369042 19404217 11771627 21001632
```

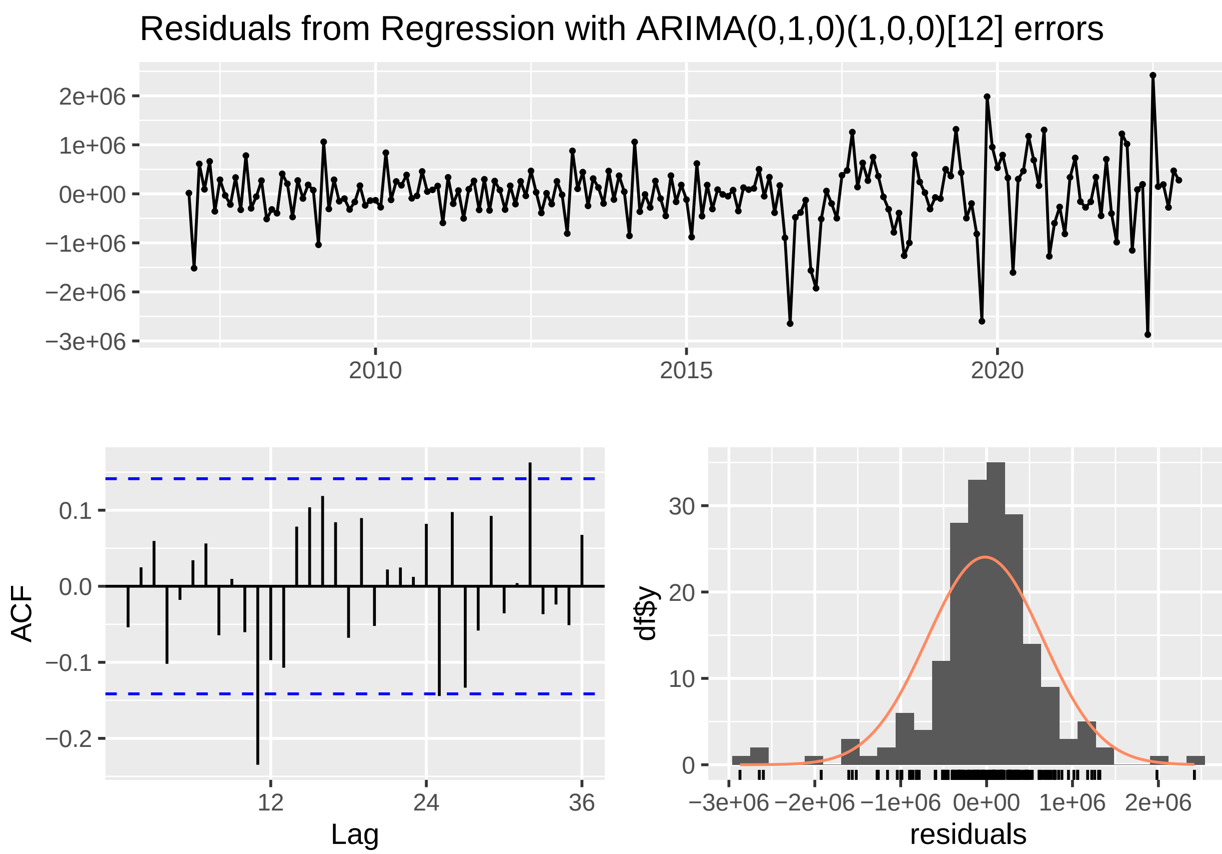
```r
# Plot the forecast
plot(forecast_3)
```




```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_3, series="Arimax",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```



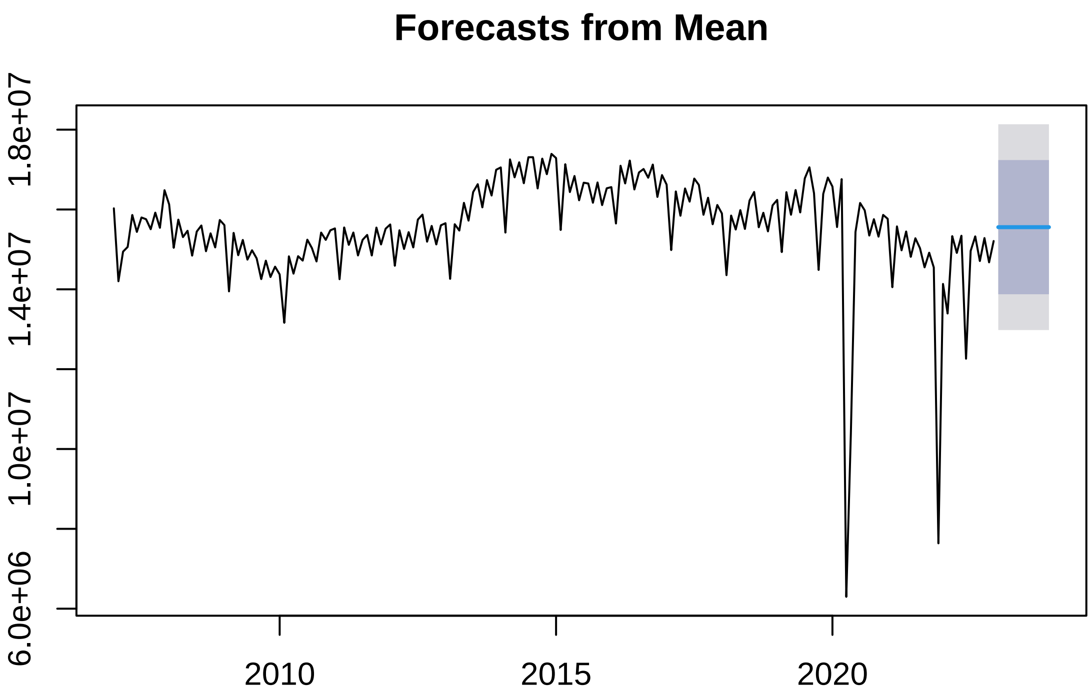

```
checkresiduals(model_3_train)
```



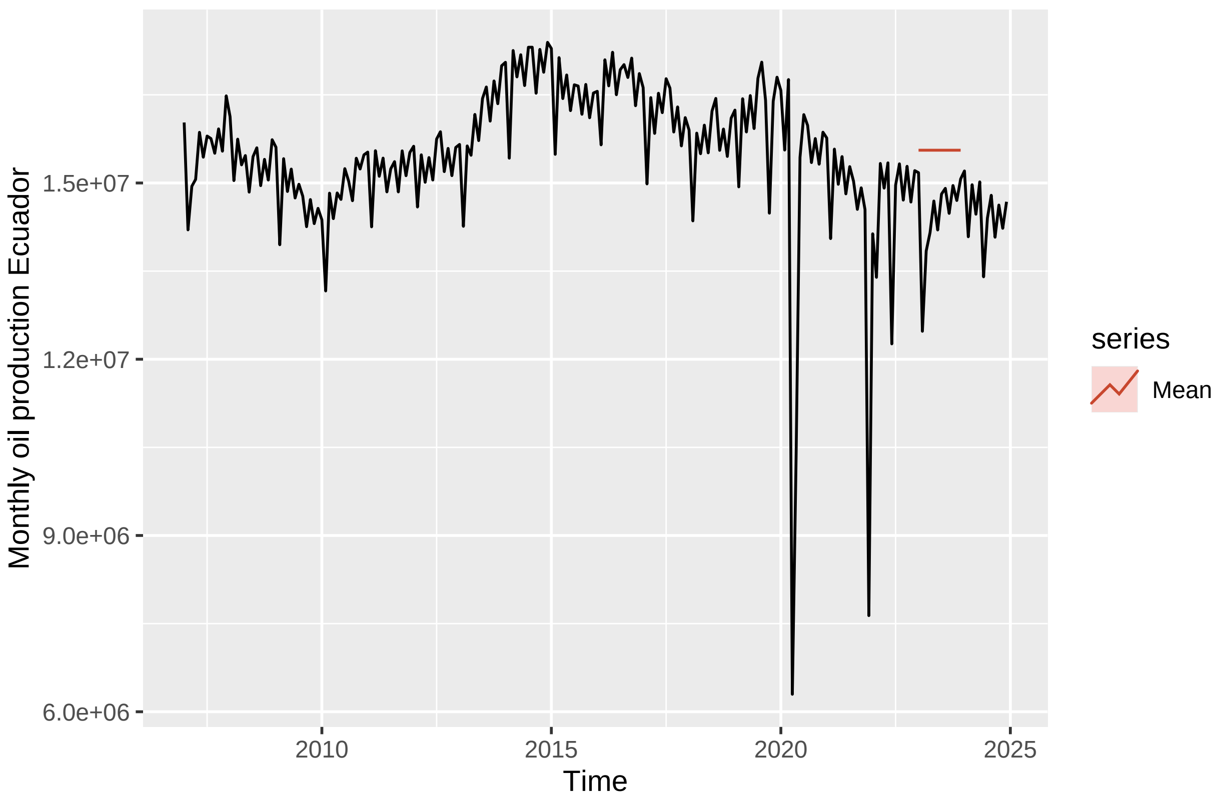

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,1,0)(1,0,0)[12] errors
## Q* = 34.78, df = 23, p-value = 0.05469
##
## Model df: 1.   Total lags used: 24
```

#Model 4

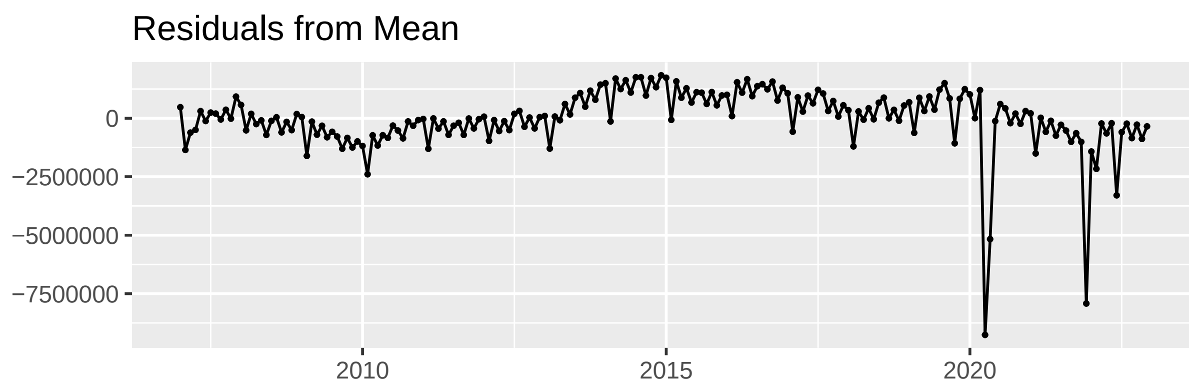
```r
# Model 4
model_4_train <- meanf(ts_train_A, h = h)

# Plot the forecast
plot(model_4_train)
```

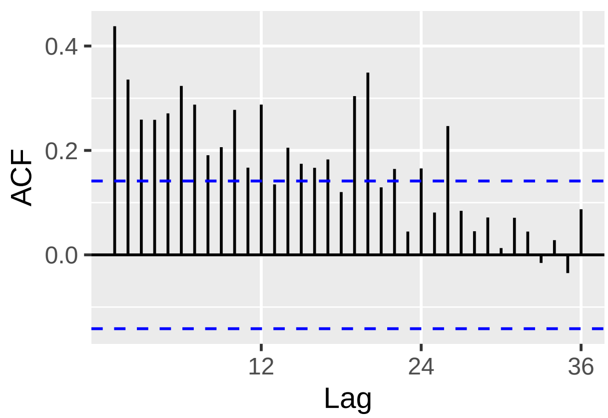


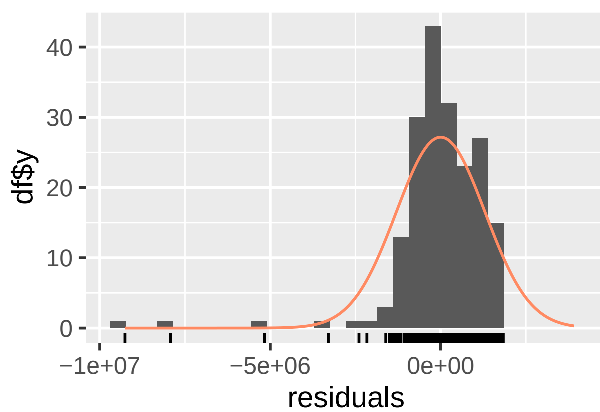

```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(model_4_train, series="Mean",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```
checkresiduals(model_4_train)
```







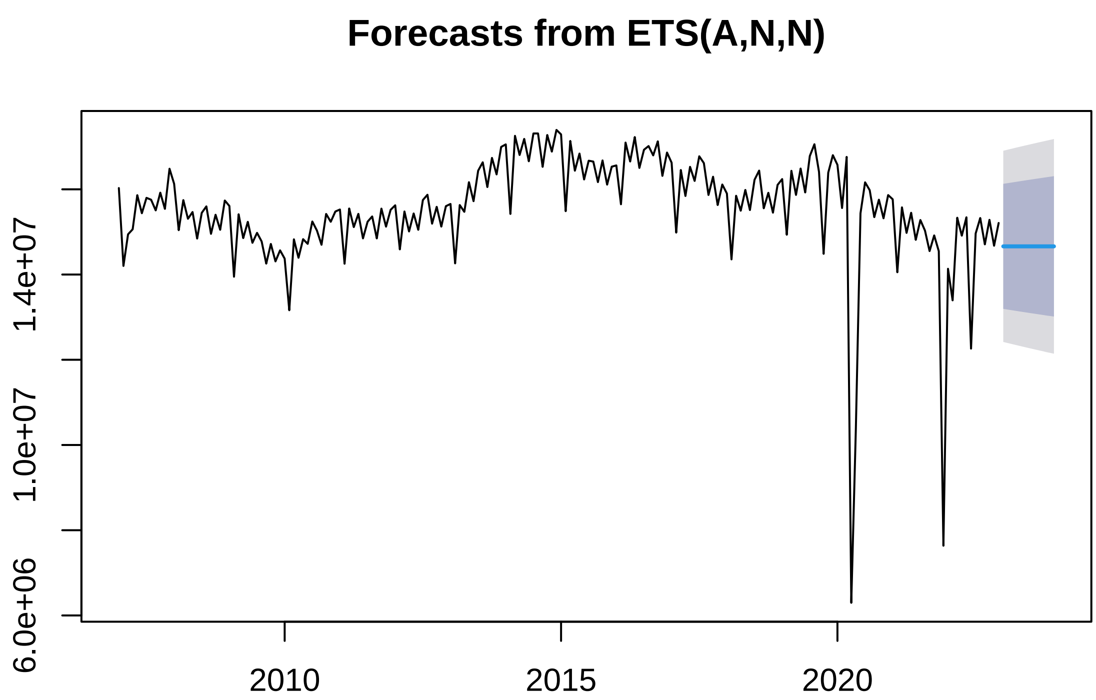

```
##
##  Ljung-Box test
##
## data:  Residuals from Mean
## Q* = 290.4, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

```r
# Model 5:
model_5_train <- ets(ts_train_A, model = "ANN")

# Forecast for Model 5
forecast_5 <- forecast(model_5_train, h = h)

print(forecast_5)
```

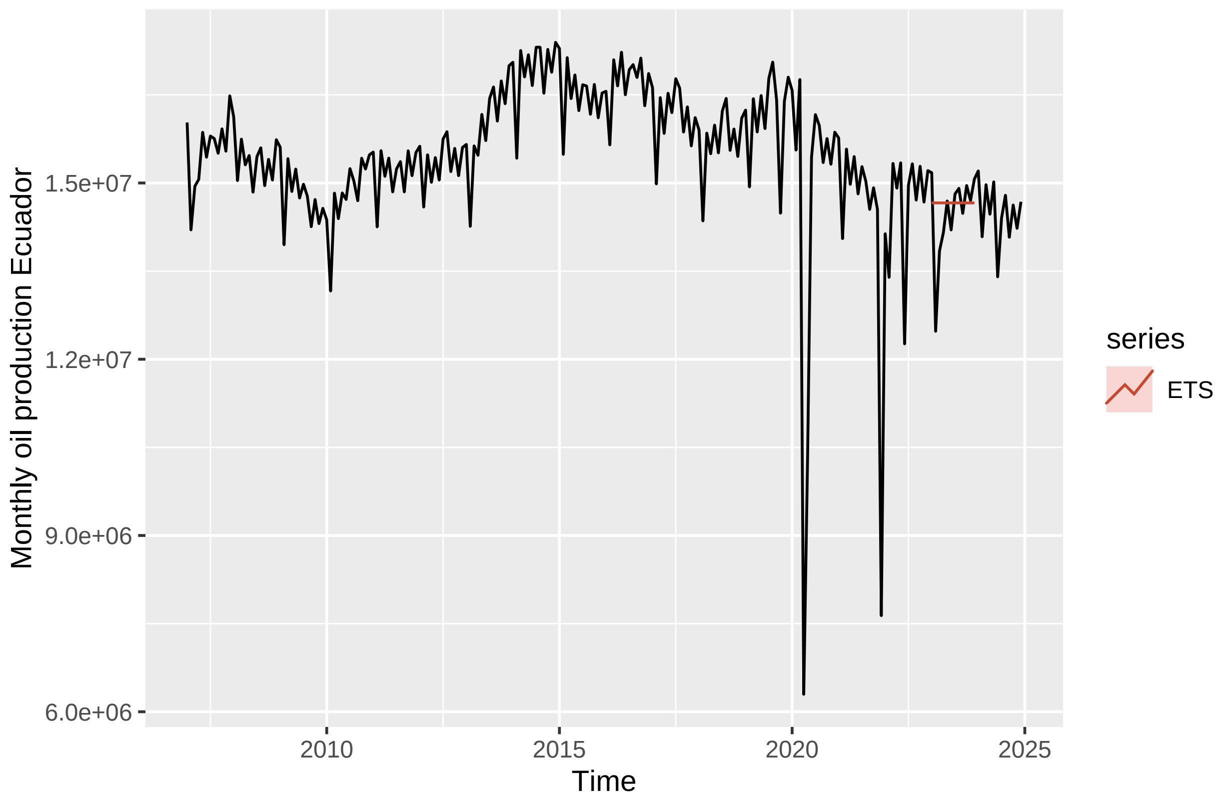
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2023       14661274 13194535 16128013 12418090 16904458
## Feb 2023       14661274 13177242 16145306 12391643 16930905
## Mar 2023       14661274 13160149 16162399 12365501 16957047
## Apr 2023       14661274 13143248 16179300 12339654 16982895
## May 2023       14661274 13126533 16196015 12314090 17008458
## Jun 2023       14661274 13109998 16212550 12288803 17033746
## Jul 2023       14661274 13093638 16228910 12263782 17058767
## Aug 2023       14661274 13077447 16245102 12239019 17083529
```

```
## Sep 2023       14661274 13061419 16261129 12214507 17108041
## Oct 2023       14661274 13045551 16276998 12190238 17132310
## Nov 2023       14661274 13029836 16292712 12166205 17156343
## Dec 2023       14661274 13014272 16308276 12142402 17180146
```

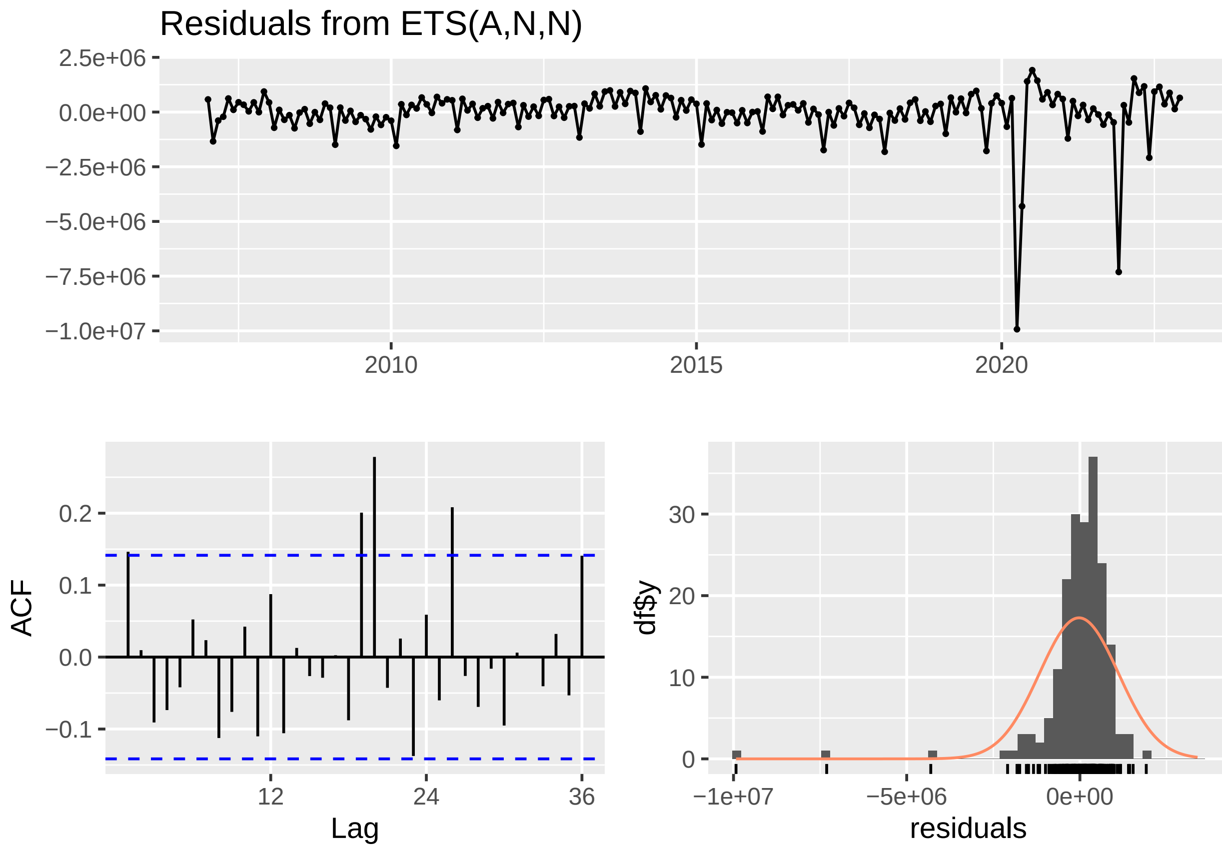
```r
# Plot the forecast
plot(forecast_5)
```




```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_5, series="ETS",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```r
checkresiduals(model_5_train)
```



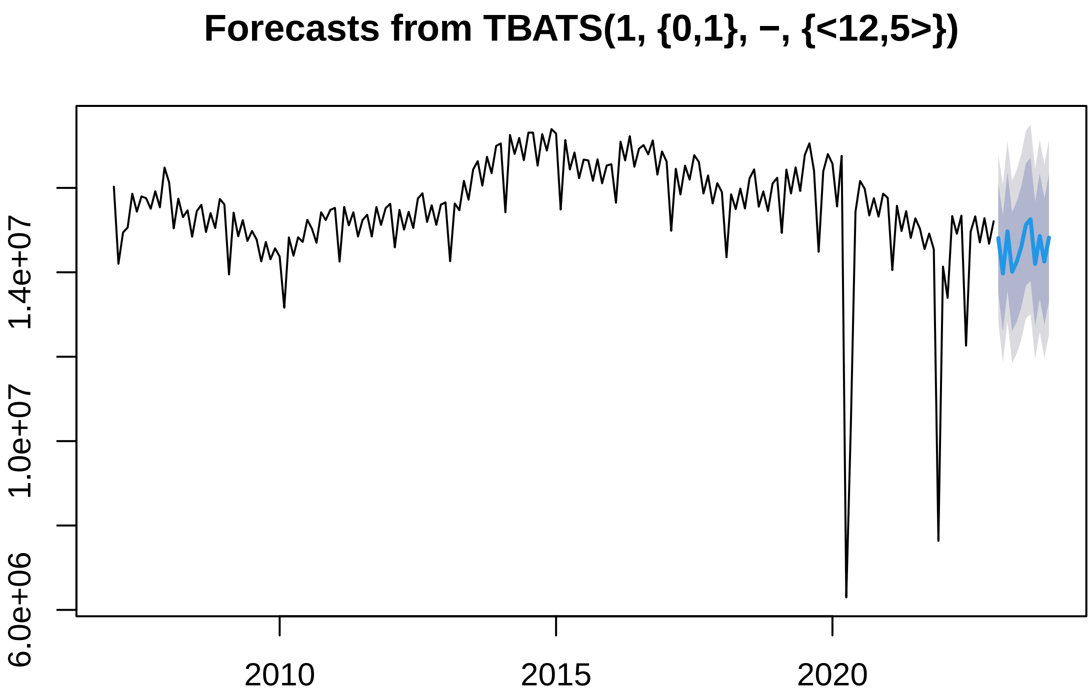

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 51.347, df = 24, p-value = 0.0009511
##
## Model df: 0.   Total lags used: 24
```

```
# Model 6:
model_6_train <- tbats(ts_train_A)

# Forecast for Model 6
forecast_6 <- forecast(model_6_train, h = h)

print(forecast_6)
```

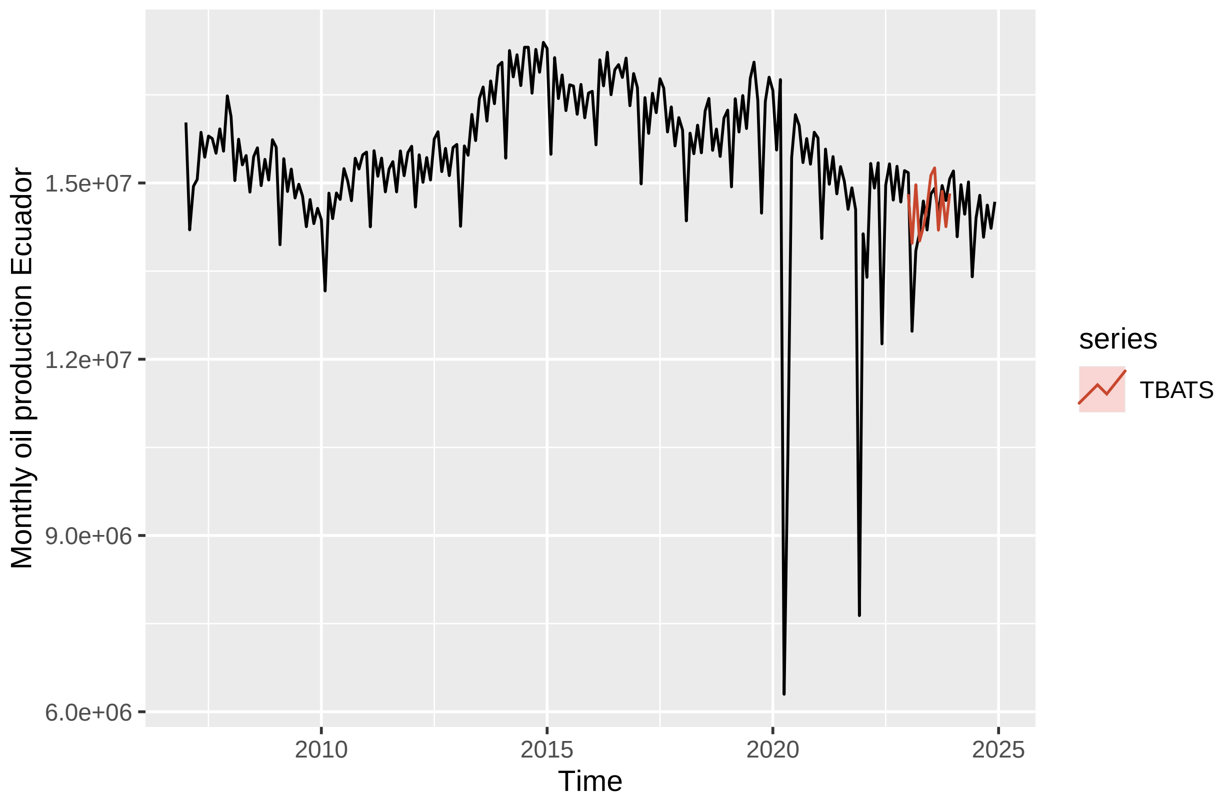
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2023       14810030 13517782 16102277 12833708 16786352
## Feb 2023       13972931 12595158 15350705 11865808 16080055
## Mar 2023       14970236 13573103 16367369 12833506 17106966
## Apr 2023       14012209 12600357 15424061 11852968 16171451
## May 2023       14248425 12822516 15674335 12067685 16429166
## Jun 2023       14606624 13167780 16045469 12406102 16807147
## Jul 2023       15129277 13678049 16580505 12909816 17348739
## Aug 2023       15256989 13793429 16720549 13018667 17495311
```

```
## Sep 2023       14197049 12722283 15671815 11941589 16452509
## Oct 2023       14860281 13373648 16346915 12586671 17133891
## Nov 2023       14256346 12759921 15752771 11967761 16544931
## Dec 2023       14821151 13313387 16328916 12515225 17127078
```

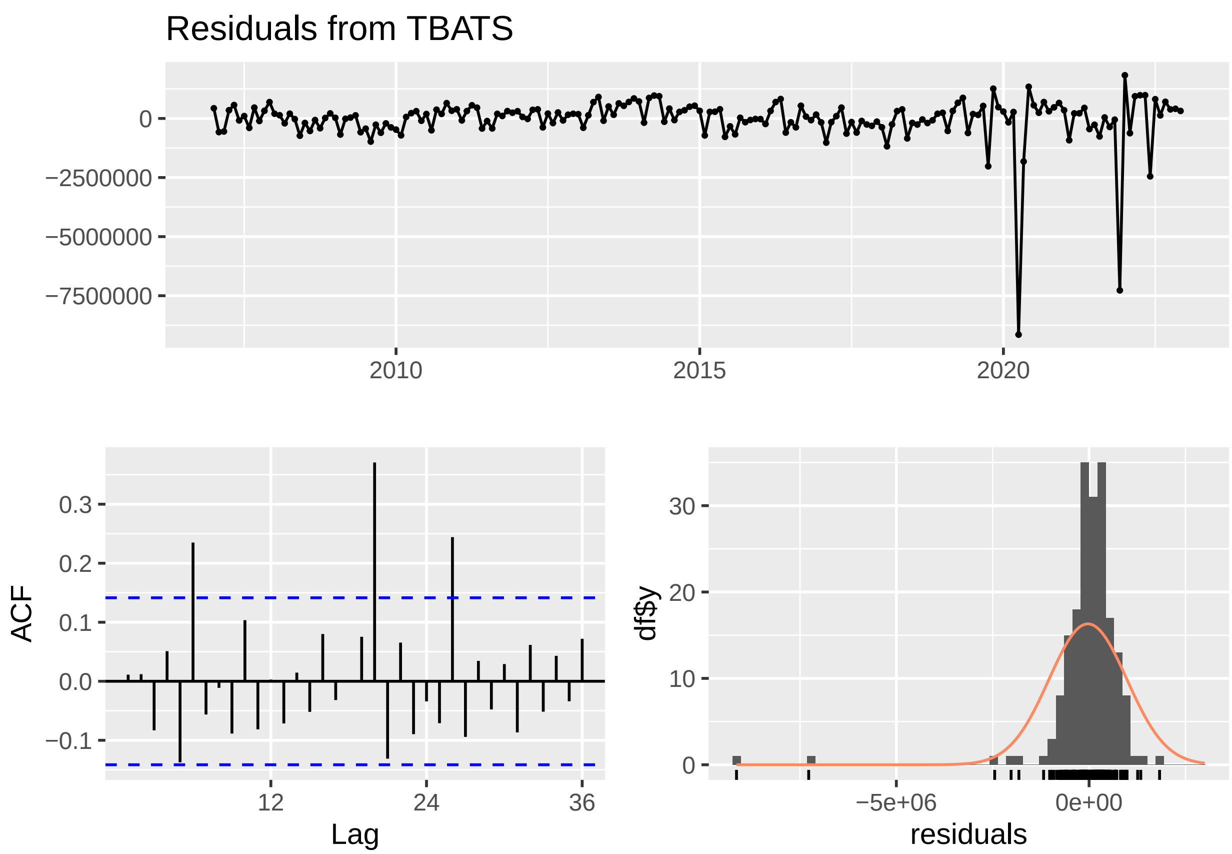
```r
# Plot the forecast
plot(forecast_6)
```




```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_6, series="TBATS",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```r
checkresiduals(model_6_train)
```

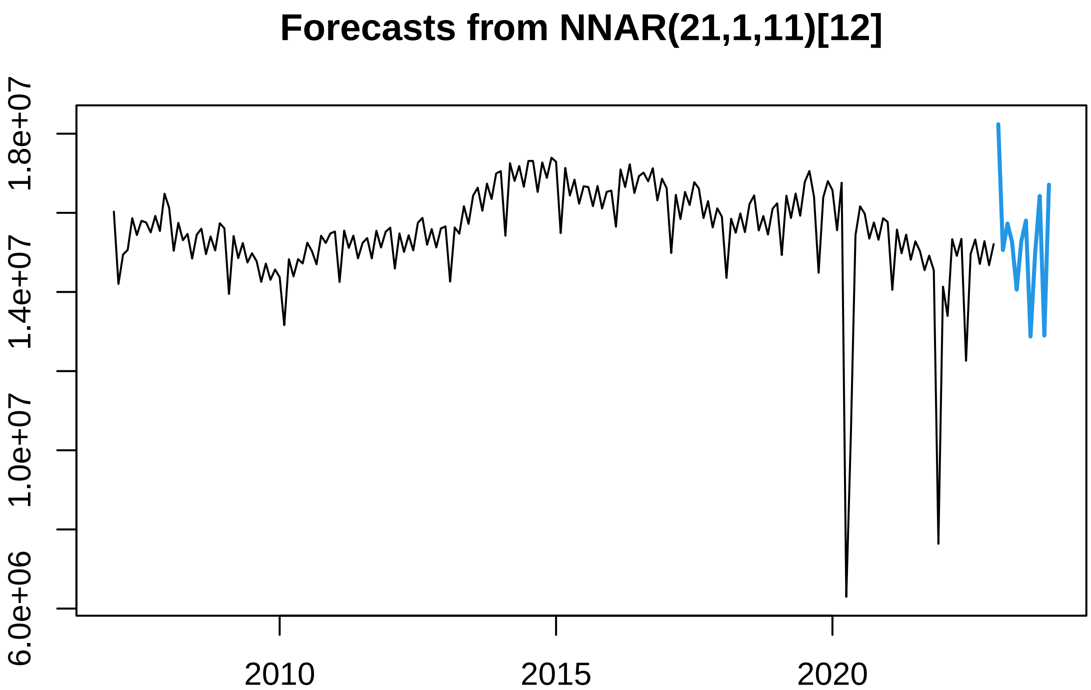



```
##
##  Ljung-Box test
##
## data:  Residuals from TBATS
## Q* = 63.515, df = 24, p-value = 2.005e-05
##
## Model df: 0.   Total lags used: 24
```

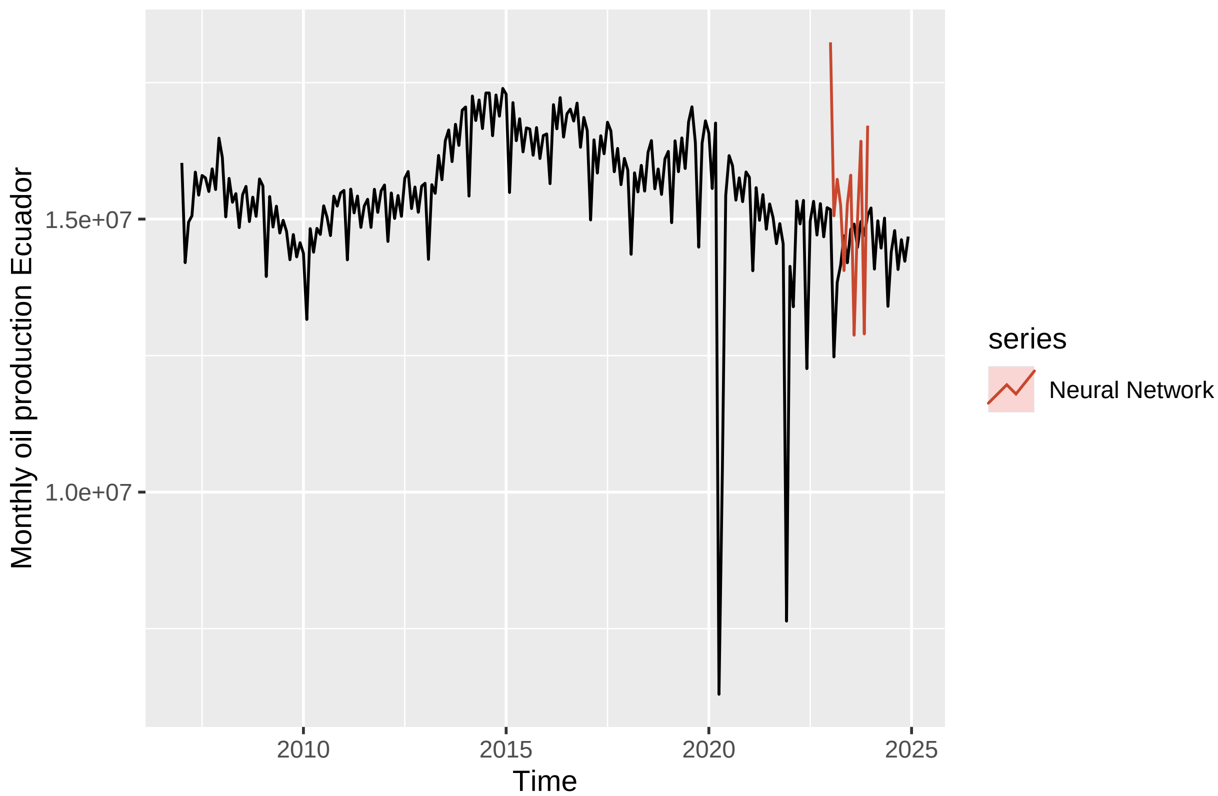
```
# Model 7:
model_7_train <- nnetar(ts_train_A)

# Forecast for Model 7
forecast_7 <- forecast(model_7_train, h = h)

print(forecast_7)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2023 18237959 15060722 15729977 15255879 14058706 15283193 15805240 12874039
##           Sep      Oct      Nov      Dec
## 2023 14980066 16426564 12896969 16712268
```

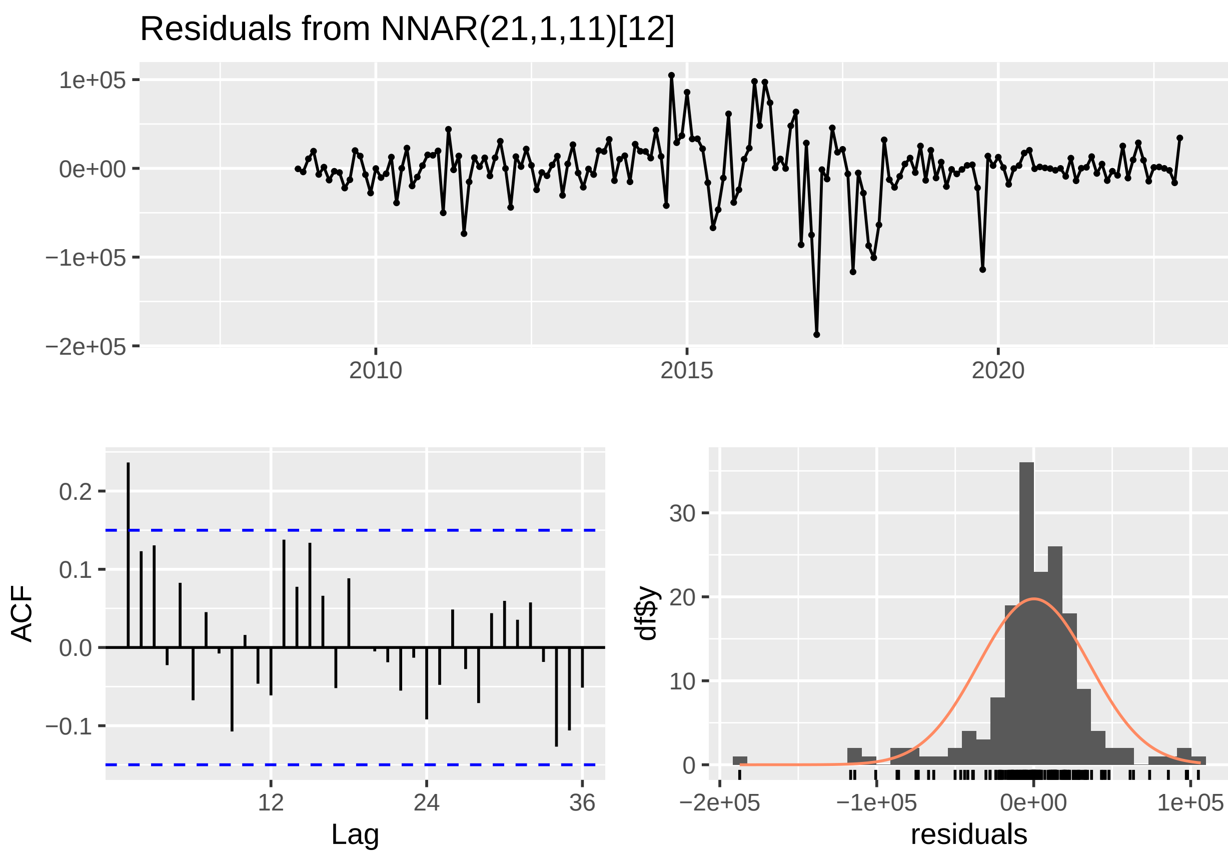
```
# Plot the forecast
plot(forecast_7)
```




```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_7, series="Neural Network",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```r
checkresiduals(model_7_train)
```



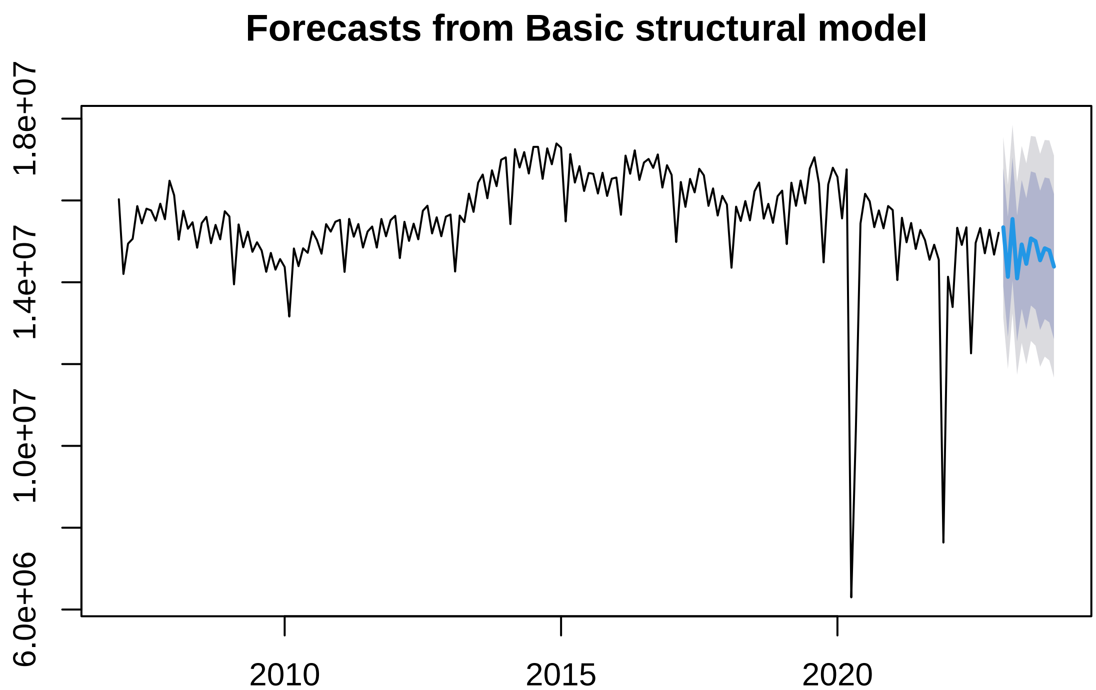

```
##
##  Ljung-Box test
##
## data:  Residuals from NNAR(21,1,11)[12]
## Q* = 34.503, df = 24, p-value = 0.07612
##
## Model df: 0.   Total lags used: 24
```

```
# Model 8:
model_8_train <- StructTS(ts_train_A, type = "BSM")

# Forecast for Model 8
forecast_8 <- forecast(model_8_train, h = h)

print(forecast_8)
```

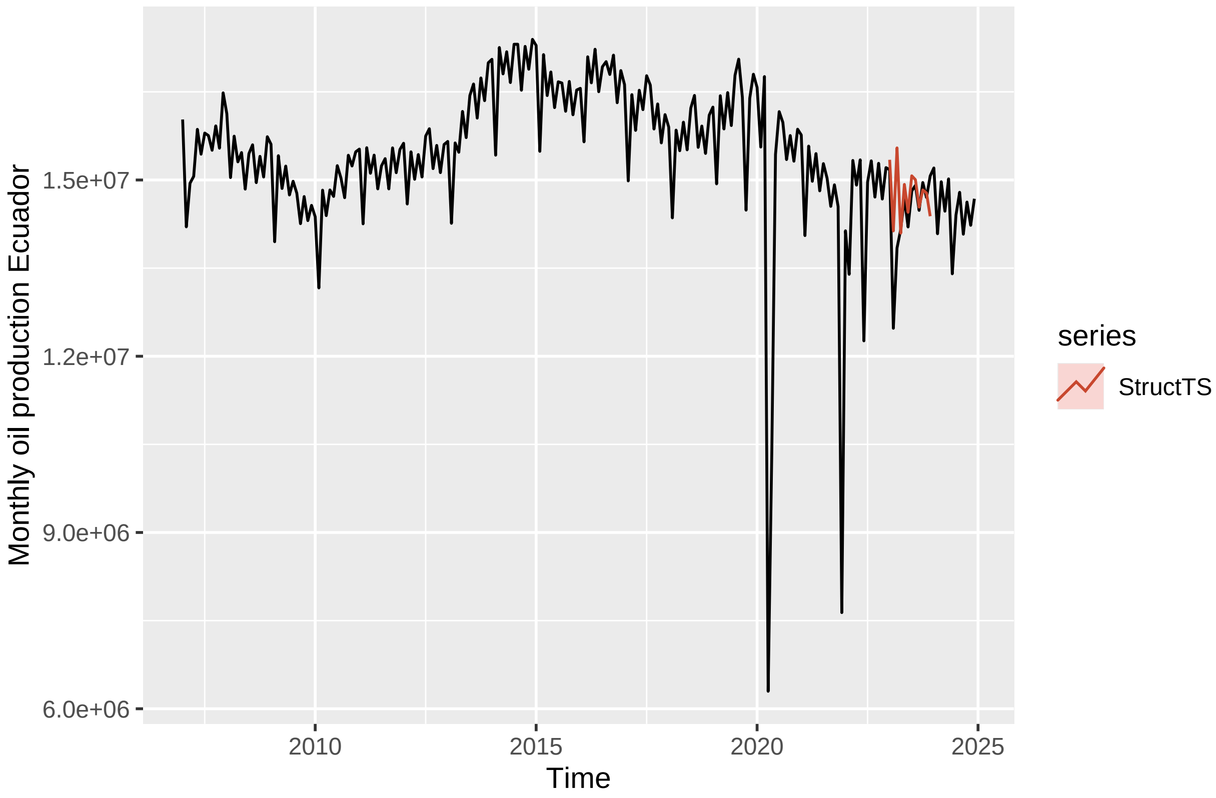
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2023       15343823 13899091 16788556 13134296 17553351
## Feb 2023       14132564 12661291 15603838 11882446 16382683
## Mar 2023       15546772 14041285 17052258 13244328 17849215
## Apr 2023       14094481 12554828 15634135 11739785 16449178
## May 2023       14924654 13351495 16497813 12518715 17330593
## Jun 2023       14450412 12844335 16056490 11994129 16906696
## Jul 2023       15071488 13433006 16709971 12565646 17577331
## Aug 2023       15004106 13333748 16674464 12449514 17558699
```

```
## Sep 2023       14537057 12835545 16238570 11934818 17139296
## Oct 2023       14830601 13099271 16561932 12182760 17478443
## Nov 2023       14778439 13020417 16536462 12089776 17467103
## Dec 2023       14383272 12608586 16157957 11669124 17097419
```

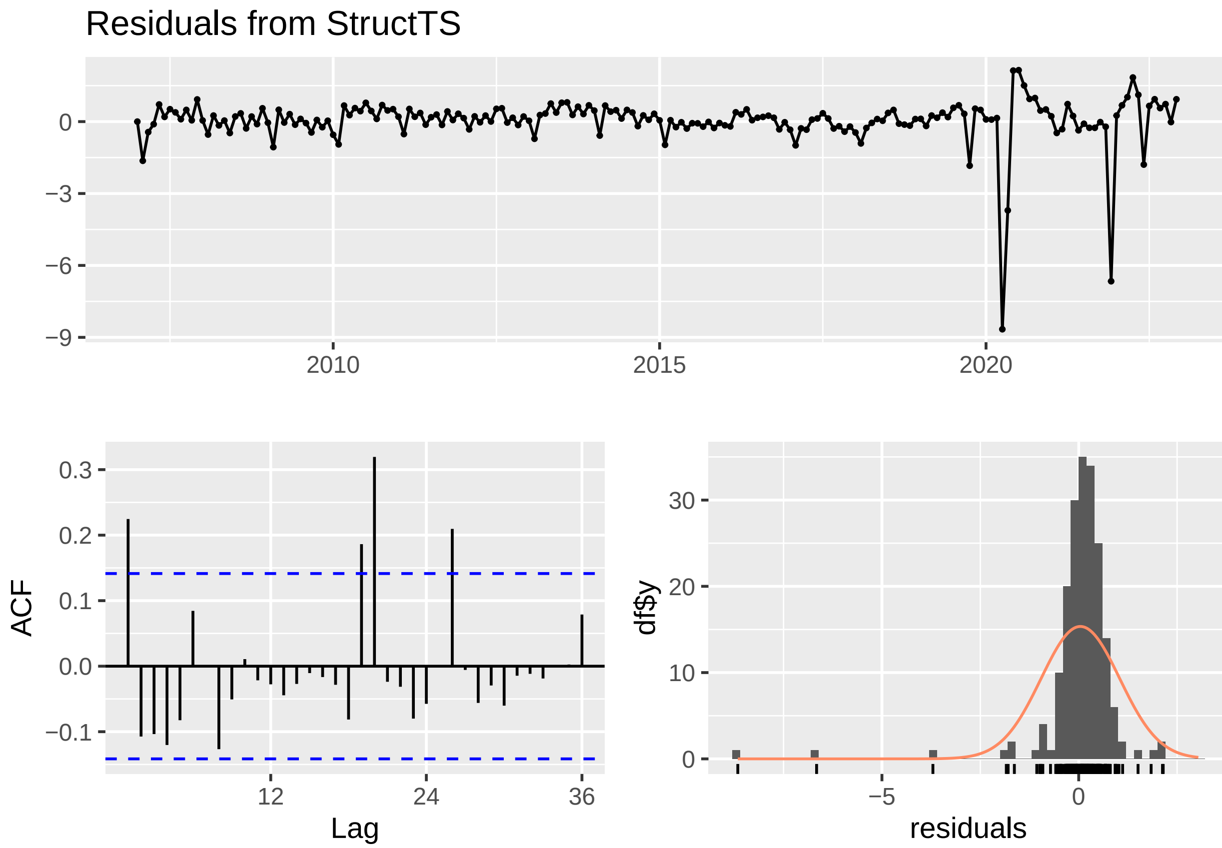
```r
# Plot the forecast
plot(forecast_8)
```




```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_8, series="StructTS",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```
checkresiduals(model_8_train)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from StructTS
## Q* = 58.197, df = 24, p-value = 0.0001143
##
## Model df: 0.   Total lags used: 24
```

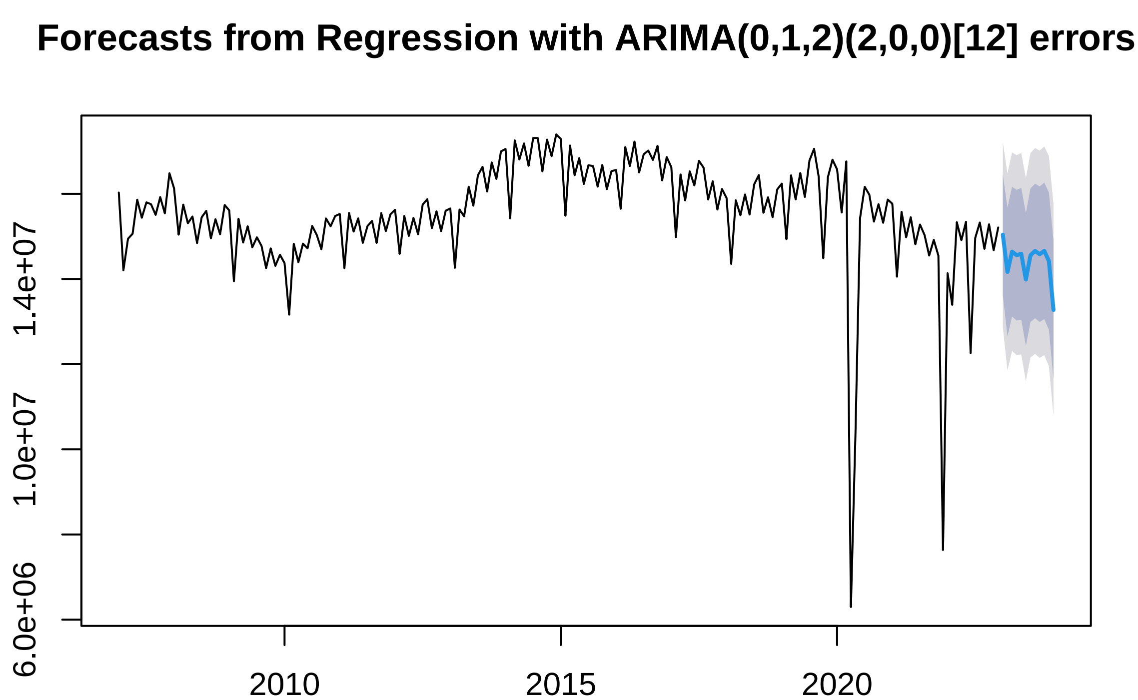
#Model 9

```
# Model 8:
model_9_train <- auto.arima(ts_train_A,
                              xreg = price_train, seasonal = TRUE, stepwise = FALSE, approximation = FALS

# Forecast for Model 8
forecast_9 <- forecast(model_9_train, h = h, xreg = price_test)

print(forecast_9)
```

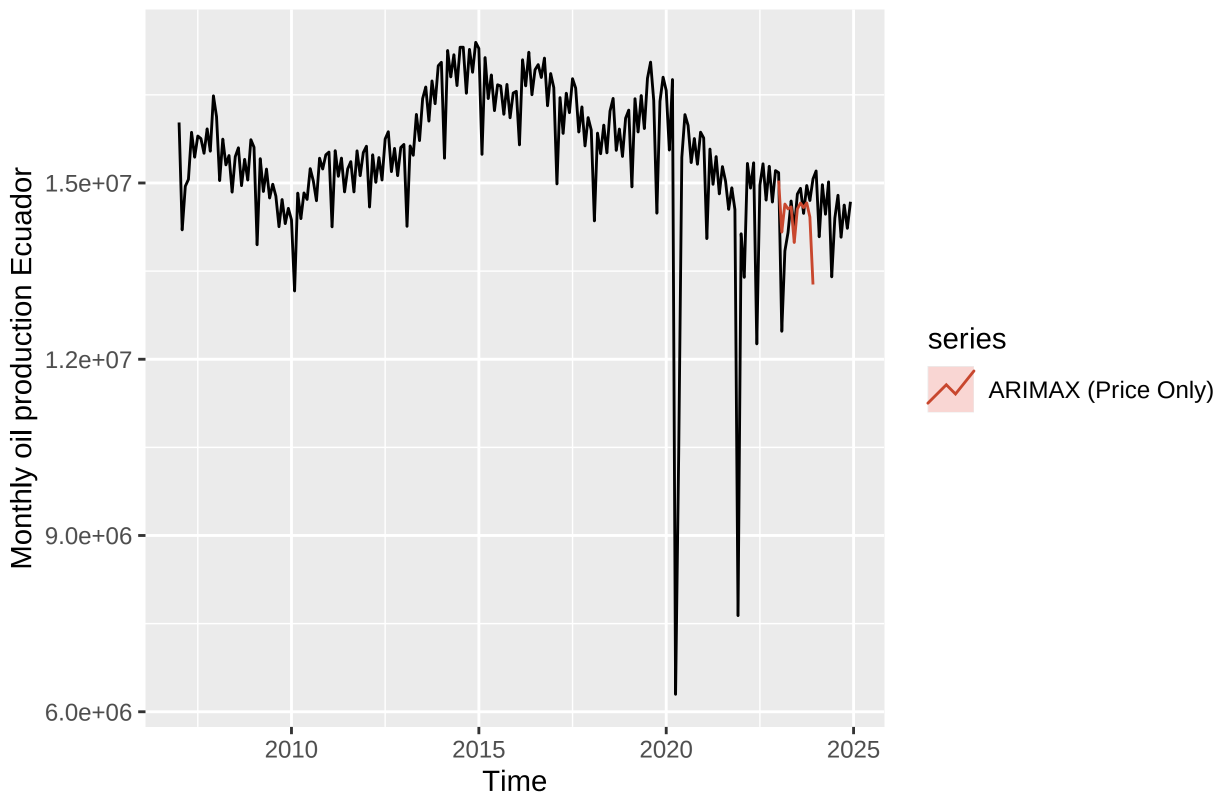
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2023       15040482 13623432 16457533 12873291 17207674
## Feb 2023       14164622 12652105 15677139 11851427 16477817
## Mar 2023       14641464 13117464 16165464 12310707 16972221
## Apr 2023       14558761 13023363 16094158 12210573 16906949
## May 2023       14592767 13046056 16139478 12227276 16958258
```

```
## Jun 2023       13987292 12429349 15545234 11604624 16369959
## Jul 2023       14557435 12988341 16126529 12157714 16957157
## Aug 2023       14658232 13078066 16238398 12241577 17074887
## Sep 2023       14581468 12990307 16172629 12147997 17014939
## Oct 2023       14661034 13058953 16263115 12210863 17111206
## Nov 2023       14420684 12807757 16033611 11953925 16887442
## Dec 2023       13272391 11648691 14896091 10789156 15755626
```

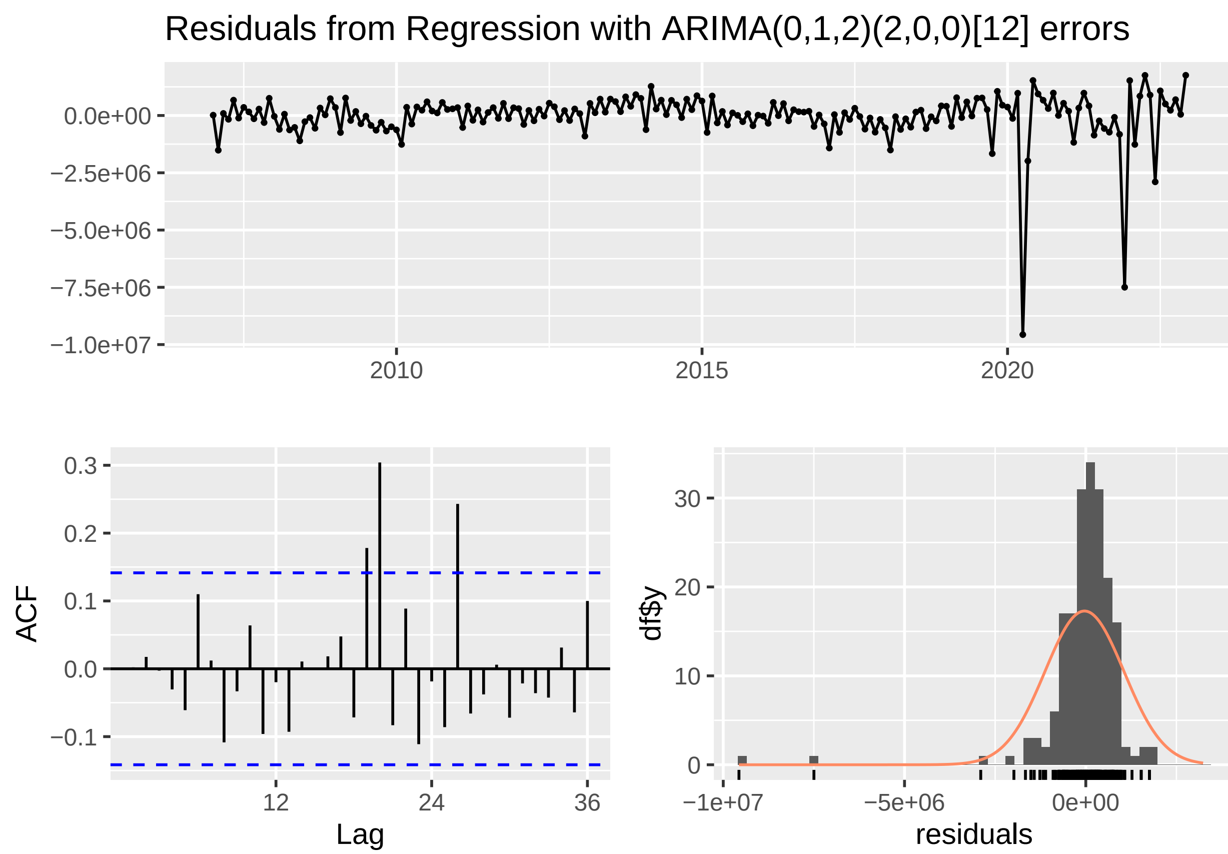
```r
# Plot the forecast
plot(forecast_9)
```




```r
#Plot model + observed data
autoplot(ts_oil_total) +
  autolayer(forecast_9, series="ARIMAX (Price Only)",PI=FALSE) +
  ylab("Monthly oil production Ecuador")
```




```r
checkresiduals(model_9_train)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,1,2)(2,0,0)[12] errors
## Q* = 45.22, df = 20, p-value = 0.00103
##
## Model df: 4.   Total lags used: 24
```

```
#Model 1
SARIMA_scores <- accuracy(forecast_1$mean, ts_test_A)

#Model 2
SARIMAX_scores <- accuracy(forecast_2$mean, ts_test_B)

#Model 3
ARIMAX_scores <- accuracy(forecast_3$mean, ts_test_A)

#Model 4
Mean_scores <- accuracy(model_4_train$mean, ts_test_A)

#Model 5
ETS_scores <- accuracy(forecast_5$mean, ts_test_A)

#Model 6
TBATS_scores <- accuracy(forecast_6$mean, ts_test_A)
```

```r
#Model 7
NN_scores <- accuracy(forecast_7$mean, ts_test_A)

#Model 8
StructTS_scores <- accuracy(forecast_8$mean, ts_test_A)

#Model 9
Arimax_p_scores <- accuracy(forecast_9$mean, ts_test_A)
```

```r
# Combine in a table for easy comparison
models_scores <- as.data.frame(rbind(SARIMA_scores, SARIMAX_scores,ARIMAX_scores,
                                     Mean_scores,ETS_scores, TBATS_scores,
                                     NN_scores,StructTS_scores, Arimax_p_scores ))
row.names(models_scores) <- c("SARIMA", "SARIMAX", "ARIMAX",
                              "Mean","ETS", "TBATS",
                                     "NN","StructTS", "Arimax_p")
print(models_scores)
```

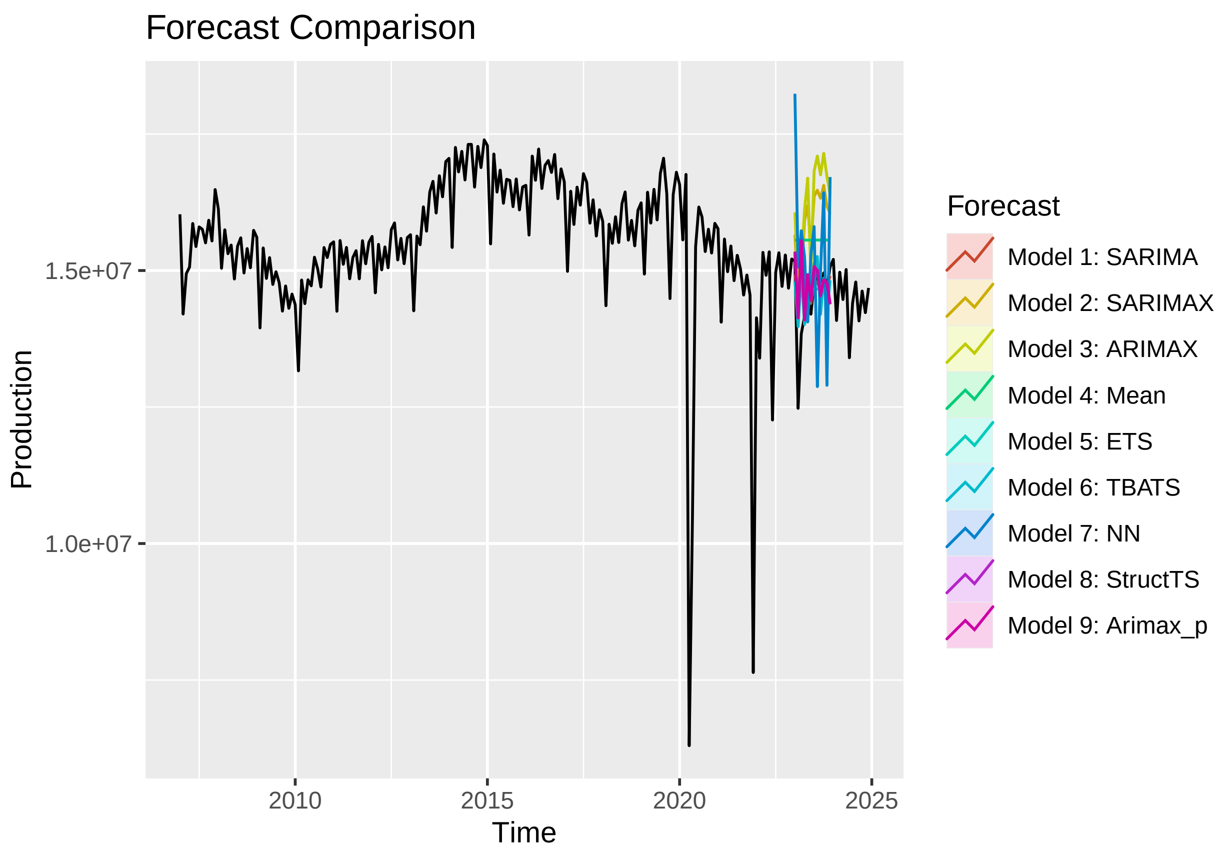
```
##                   ME      RMSE       MAE          MPE      MAPE        ACF1
## SARIMA    -278293.15  693303.0  406561.4  -2.16415262  3.017604  0.17326075
## SARIMAX  -1419518.45 1476086.4 1419518.5  -9.82953050  9.829530  0.27825205
## ARIMAX   -1752244.91 1832260.0 1752244.9 -12.18567542 12.185675 -0.20539000
## Mean     -1101461.92 1310253.5 1101461.9  -7.90358838  7.903588  0.03492328
## ETS       -205012.18  738631.1  486261.3  -1.68608859  3.558863  0.03492328
## TBATS     -138867.29  619140.1  477062.0  -1.14463527  3.434871  0.19380236
## NN        -820536.63 1729652.8 1566298.3  -5.87943939 10.920379  0.06269289
## StructTS  -301877.35  728247.7  447376.6  -2.29073820  3.261772  0.32730412
## Arimax_p    28209.18  777434.1  526220.8  -0.04530021  3.758984  0.29258131
##          Theil's U
## SARIMA   0.7240965
## SARIMAX  2.5875701
## ARIMAX   1.9128261
## Mean     1.3849179
## ETS      0.7520508
## TBATS    0.6598076
## NN       1.5770627
## StructTS 0.8107510
## Arimax_p 0.8122824
```

```r
#choose model with lowest RMSE
best_model_index <- which.min(models_scores[,"RMSE"])
cat("The best model by RMSE is:", row.names(models_scores[best_model_index,]))
```

```
## The best model by RMSE is: TBATS
```

```r
autoplot(ts_oil_total) +
  autolayer(forecast_1, series = "Model 1: SARIMA", PI = FALSE) +
  autolayer(forecast_2, series = "Model 2: SARIMAX", PI = FALSE) +
  autolayer(forecast_3, series = "Model 3: ARIMAX", PI = FALSE) +
  autolayer(model_4_train, series = "Model 4: Mean", PI = FALSE) +
  autolayer(forecast_5, series = "Model 5: ETS", PI = FALSE) +
```

```r
  autolayer(forecast_6, series = "Model 6: TBATS", PI = FALSE) +
  autolayer(forecast_7, series = "Model 7: NN", PI = FALSE) +
  autolayer(forecast_8, series = "Model 8: StructTS", PI = FALSE) +
  autolayer(forecast_8, series = "Model 9: Arimax_p", PI = FALSE) +
  ggtitle("Forecast Comparison") +
  xlab("Time") + ylab("Production")+
  guides(colour=guide_legend(title="Forecast"))
```




```r
kbl(models_scores,
      caption = "Forecast Accuracy for Monthly Data",
      digits = array(9,ncol(models_scores))) %>%
  kable_styling(full_width = FALSE, position = "center") %>%
  #highlight model with lowest RMSE
  kable_styling(latex_options="striped", stripe_index = which.min(models_scores[,"RMSE"])) %>%
  kable_styling(full_width = FALSE) %>%
  row_spec(6, bold = TRUE, background = "#F0F0F0")  # highlight best MAPE
```

#Scenario Analysis

TBATS—the best-performing model among the nine—to conduct the scenario analysis. Because TBATS is a univariate model, we estimated the Block 43 contribution separately. Our approach compute the average monthly production from Block 43 during a recent period and then “remove” that contribution from the TBATS baseline forecast to simulate a shutdown. In other words, the shutdown scenario forecast equals the TBATS baseline forecast minus the estimated Block 43 production.

Table 3: Forecast Accuracy for Monthly Data

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
SARIMA	-278293.15	693303.0	406561.4	-2.16415262	3.017604	0.17326075	0.7240965
SARIMAX	-1419518.45	1476086.4	1419518.5	-9.82953050	9.829530	0.27825205	2.5875701
ARIMAX	-1752244.91	1832260.0	1752244.9	-12.18567542	12.185675	-0.20539000	1.9128261
Mean	-1101461.92	1310253.5	1101461.9	-7.90358838	7.903588	0.03492328	1.3849179
ETS	-205012.18	738631.1	486261.3	-1.68608859	3.558863	0.03492328	0.7520508
TBATS	**-138867.29**	**619140.1**	**477062.0**	**-1.14463527**	**3.434871**	**0.19380236**	**0.6598076**
NN	-820536.63	1729652.8	1566298.3	-5.87943939	10.920379	0.06269289	1.5770627
StructTS	-301877.35	728247.7	447376.6	-2.29073820	3.261772	0.32730412	0.8107510
Arimax_p	28209.18	777434.1	526220.8	-0.04530021	3.758984	0.29258131	0.8122824

```r
#The code proceeds as follows:

#Fit a TBATS model to the total production series through 2023.
#Generate a baseline forecast using TBATS.
#Compute the average Block 43 production over the last 12 months (or use a ramp-down vector).
#Create a "shutdown scenario" forecast by subtracting that average from the TBATS forecast.
#Compute and plot the production gap.
```

```r
#Fit TBATS Model on Total Production
tbats_model <- tbats(ts_oil_total_2023)
forecast_baseline <- forecast(tbats_model, h = h)
print(forecast_baseline)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2024       14741703 13478080 16005326 12809158 16674247
## Feb 2024       13860366 12514021 15206710 11801309 15919422
## Mar 2024       14906029 13541316 16270742 12818880 16993178
## Apr 2024       14067001 12687870 15446132 11957802 16176200
## May 2024       14292902 12900140 15685663 12162857 16422946
## Jun 2024       14542883 13137448 15948317 12393456 16692309
## Jul 2024       15099255 13681801 16516708 12931446 17267063
## Aug 2024       15282824 13853287 16712362 13096536 17469113
## Sep 2024       14192595 12752172 15633017 11989659 16395531
## Oct 2024       14840792 13388740 16292843 12620071 17061513
## Nov 2024       14207417 12745843 15668991 11972132 16442702
## Dec 2024       14813145 13340459 16285831 12560866 17065423
```

```r
# Estimate Block 43 Contribution
# Here, we compute the average monthly production from Block43 over the last 12 months.
average_block43 <- mean(tail(oil_data_2023$barrels_b043, 12))
cat("Average monthly Block 43 production:", average_block43, "\n")
```

```
## Average monthly Block 43 production: 1656682
```

```r
# Alternatively, you could create a ramp-down vector if you expect a gradual shutdown.
# For a simple case, we use a constant value:
block43_shutdown <- rep(average_block43, h)  # this will be subtracted from the baseline
```

```r
# Create Shutdown Scenario Forecast
# The shutdown scenario forecast is computed by subtracting Block43's contribution.
forecast_shutdown <- forecast_baseline
forecast_shutdown$mean <- forecast_baseline$mean - block43_shutdown

# Compute Production Gap
production_gap <- forecast_baseline$mean - forecast_shutdown$mean
cat("Production gap (per month):\n")
```

```
## Production gap (per month):
```

```r
print(production_gap)
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 2024 1656682 1656682 1656682 1656682 1656682 1656682 1656682 1656682 1656682
##          Oct     Nov     Dec
## 2024 1656682 1656682 1656682
```

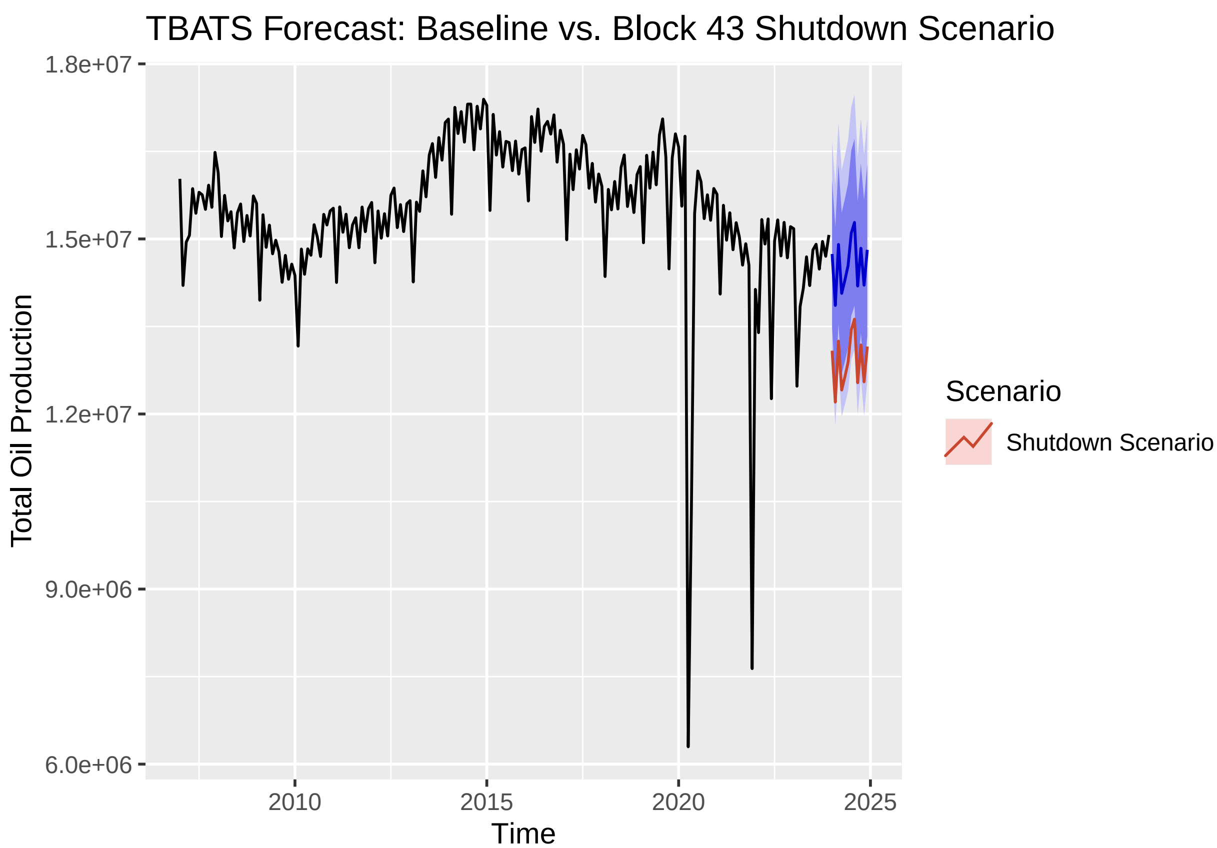
```r
total_gap <- sum(production_gap)  # Sum of all monthly losses
avg_gap   <- mean(production_gap) # Mean monthly loss
cat("Average monthly production gap:", avg_gap, "\n")
```

```
## Average monthly production gap: 1656682
```

```r
cat("Total production gap over the forecast period:", total_gap, "\n")
```

```
## Total production gap over the forecast period: 19880180
```

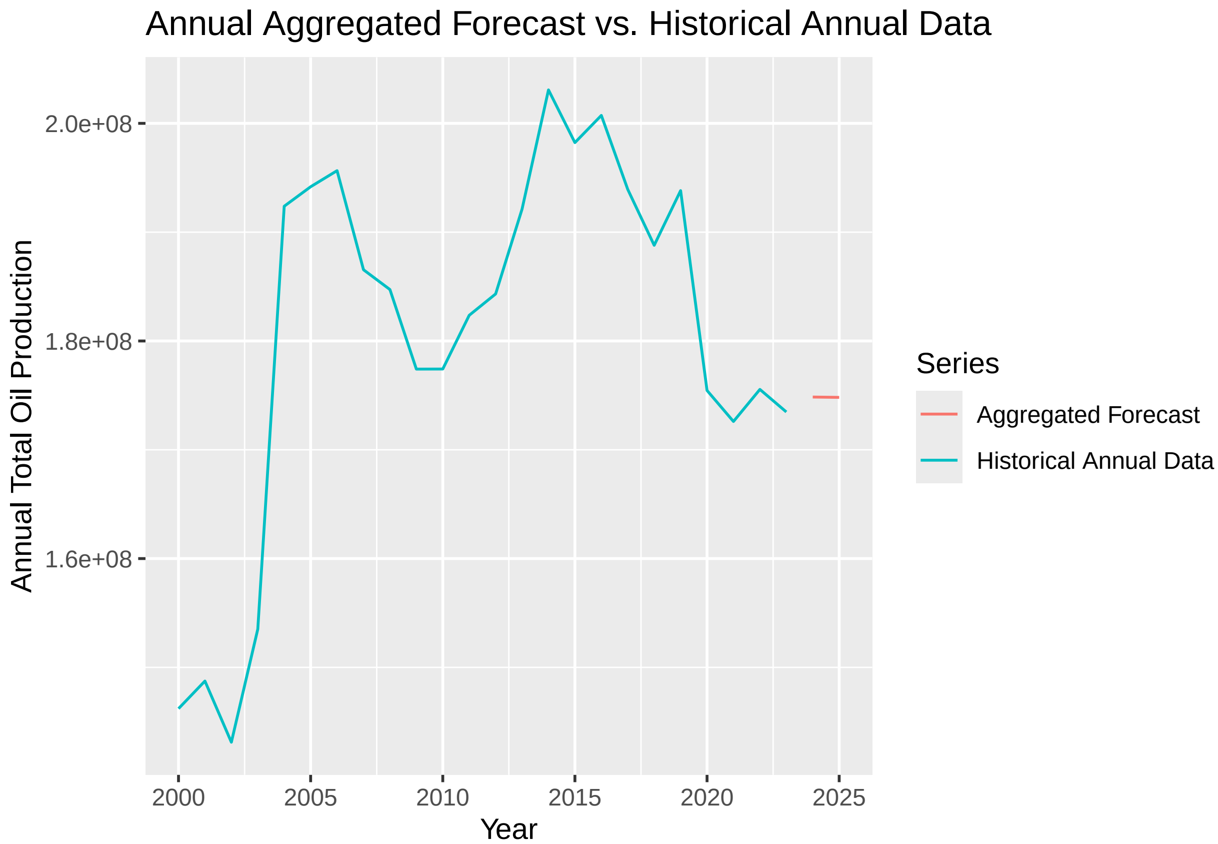
```r
# Plot the Forecast Scenarios
autoplot(forecast_baseline) +
  autolayer(forecast_shutdown, series = "Shutdown Scenario", PI = FALSE) +
  ggtitle("TBATS Forecast: Baseline vs. Block 43 Shutdown Scenario") +
  xlab("Time") + ylab("Total Oil Production") +
  guides(colour = guide_legend(title = "Scenario"))
```




```r
forecast_obj <- forecast(tbats_model, h = 24)  # forecast 2 years ahead, for example
# Aggregate the monthly forecast to annual totals.
# 'nfrequency = 1' converts the series to annual frequency.
annual_forecast <- aggregate(forecast_obj$mean, nfrequency = 1, FUN = sum)

print(annual_forecast)
```

```
## Time Series:
## Start = 2024
## End = 2025
## Frequency = 1
## [1] 174846909 174814020
```

```r
autoplot(annual_forecast, series = "Aggregated Forecast") +
  autolayer(annual_ts_2023, series = "Historical Annual Data", PI = FALSE) +
  ggtitle("Annual Aggregated Forecast vs. Historical Annual Data") +
  xlab("Year") + ylab("Annual Total Oil Production") +
  guides(colour = guide_legend(title = "Series"))
```



Summary and Conclusions

References

Banco Central del Ecuador. (2023). Estudio de los impactos macroeconómicos de mantener el crudo del Bloque 43-ITT indefinidamente en el subsuelo. *74*. https://contenido.bce.fin.ec/documentos/PublicacionesNotas/Catalogo/Apuntes/ae74.pdf

Corte Consitutional del Ecuador. (2023). *Case no. 6-22-CP*. http://esacc.corteconstitucional.gob.ec/storage/api/v1/10_DWL_FL/e2NhcnBldGE6J3RyYW1pdGUnLHV1aWQ6JzYwMjJlYzc1LWViYzctNDNjYi05MjJjLW
=

UNESCO. (2024). *Main initiatives in the Yasuní Biosphere Reserve, Ecuador | UNESCO*. https://www.unesco.org/en/amazon
biosphere-reserves-project/yasuni.