STA457 - Gorup Project

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
##
##
## Attaching package: 'xgboost'
##
##
## The following object is masked from 'package:dplyr':
##
##     slice
##
##
## Loading required package: lattice
##
##
## Attaching package: 'caret'
##
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
Cocoa_prices <- read_csv("Daily Prices_ICCO.csv",show_col_types = FALSE)
Ghana_data <- read_csv("Ghana_data.csv",show_col_types = FALSE)
```

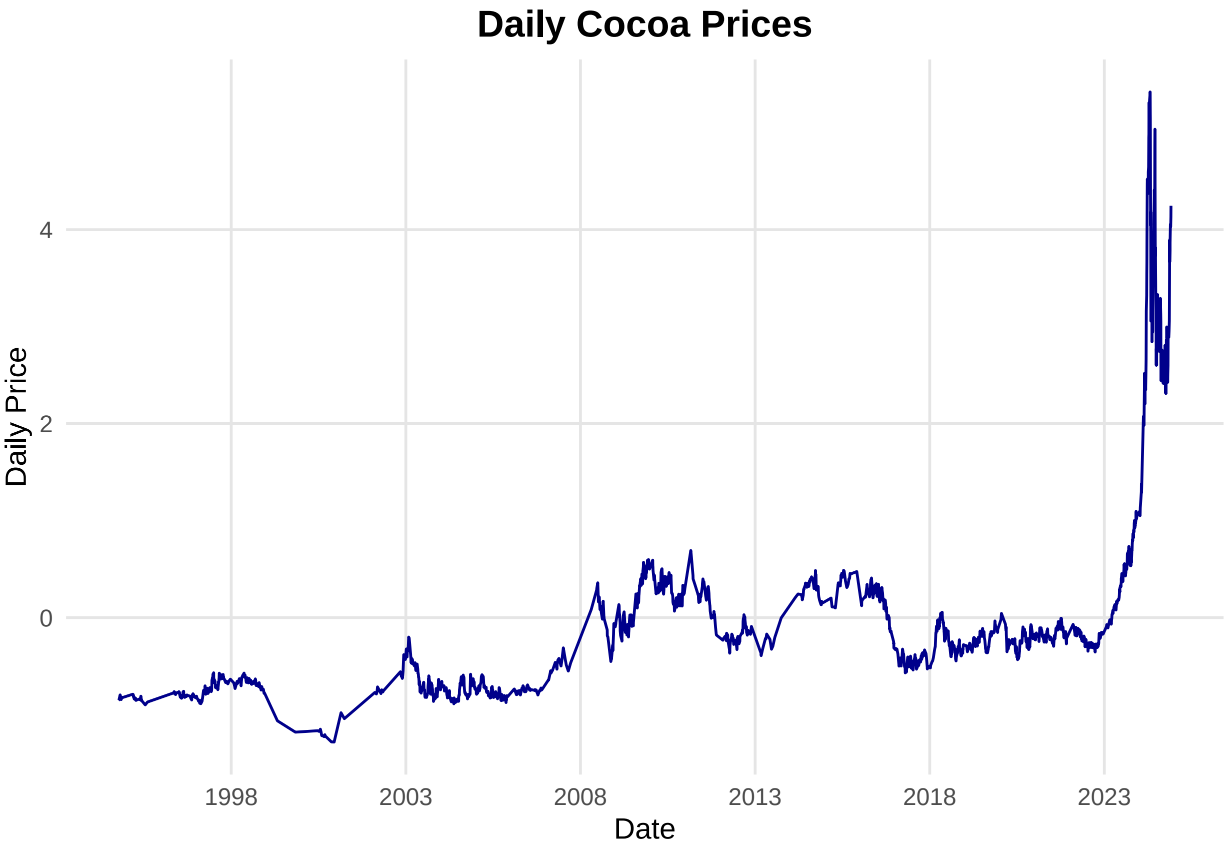
```r
#Data Cleaning
Cocoa_prices_clean <- Cocoa_prices %>%
  mutate(Date = dmy(Date),
         ICCO_price = as.numeric(gsub("/", "", `ICCO daily price (US$/tonne)`))) %>%
  select(Date, Daily_price = ICCO_price) %>%
  arrange(Date)

Ghana_data_clean <- Ghana_data %>%
  mutate(DATE = ymd(DATE)) %>%
  select(Date = DATE, PRCP, TAVG, TMAX, TMIN)

cocoa_data <- inner_join(Cocoa_prices_clean, Ghana_data_clean, by = "Date") %>%
  mutate(log_price = log(Daily_price),
         diff_log_price = c(NA, diff(log_price))) %>%
  drop_na()
```

```
## Warning in inner_join(Cocoa_prices_clean, Ghana_data_clean, by = "Date"): Detected an unexpected many
## i Row 3 of `x` matches multiple rows in `y`.
## i Row 10557 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

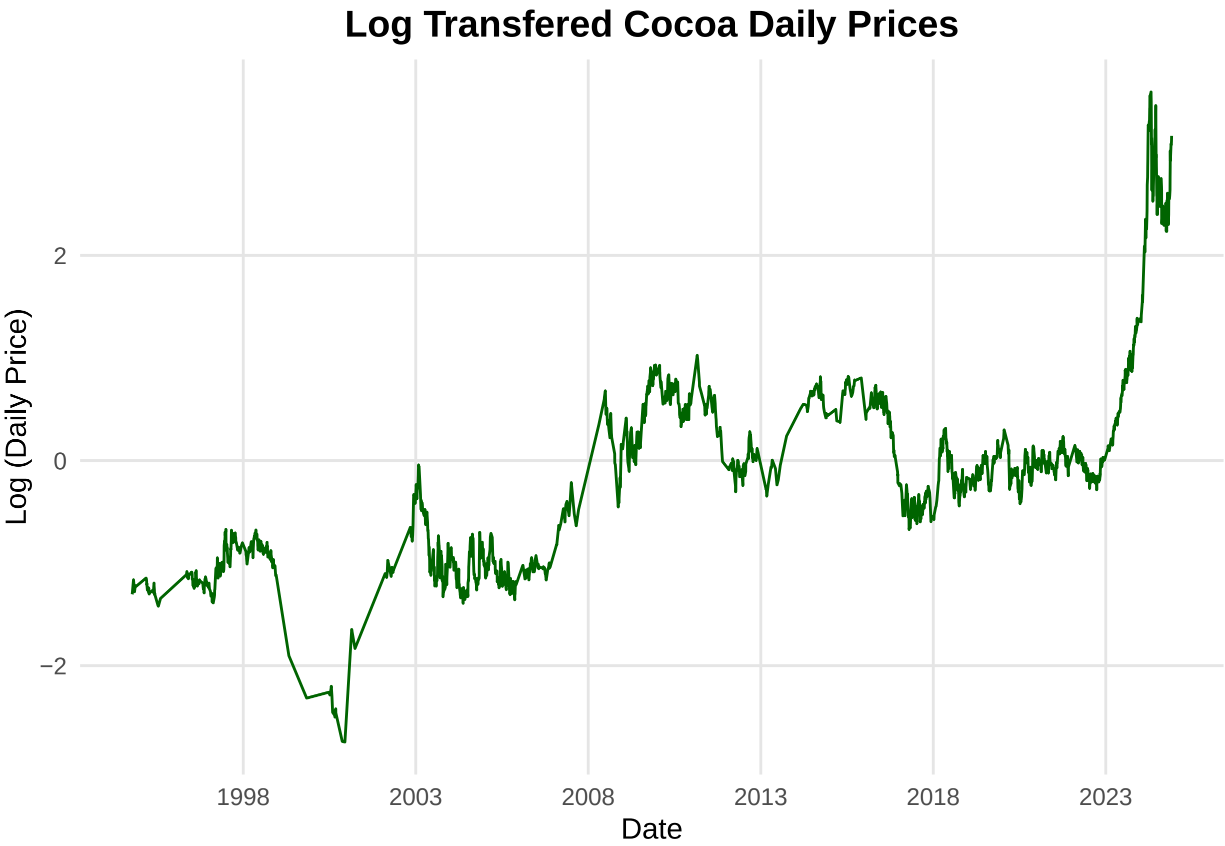
```
cocoa_data
```

```
## # A tibble: 6,527 x 8
##    Date       Daily_price  PRCP  TAVG  TMAX  TMIN log_price diff_log_price
##    <date>           <dbl> <dbl> <dbl> <dbl> <dbl>     <dbl>          <dbl>
##  1 1994-10-12       1412.  0.94    75    82    69      7.25         0     
##  2 1994-10-14       1416.  0.55    82    90    69      7.26         0     
##  3 1994-10-27       1497.  0.04    79    87    74      7.31         0.0136
##  4 1994-10-27       1497.  0.51    77    84    65      7.31         0     
##  5 1994-10-27       1497.  0.04    80    84    74      7.31         0     
##  6 1994-10-27       1497.  0.55    83    90    73      7.31         0     
##  7 1994-11-01       1465.  0.12    78    87    71      7.29         0     
##  8 1994-11-01       1465.  0.39    81    88    69      7.29         0     
##  9 1994-11-07       1426.  0       81    97    71      7.26        -0.0158
## 10 1994-11-07       1426.  0       75    96    71      7.26         0     
## # i 6,517 more rows
```

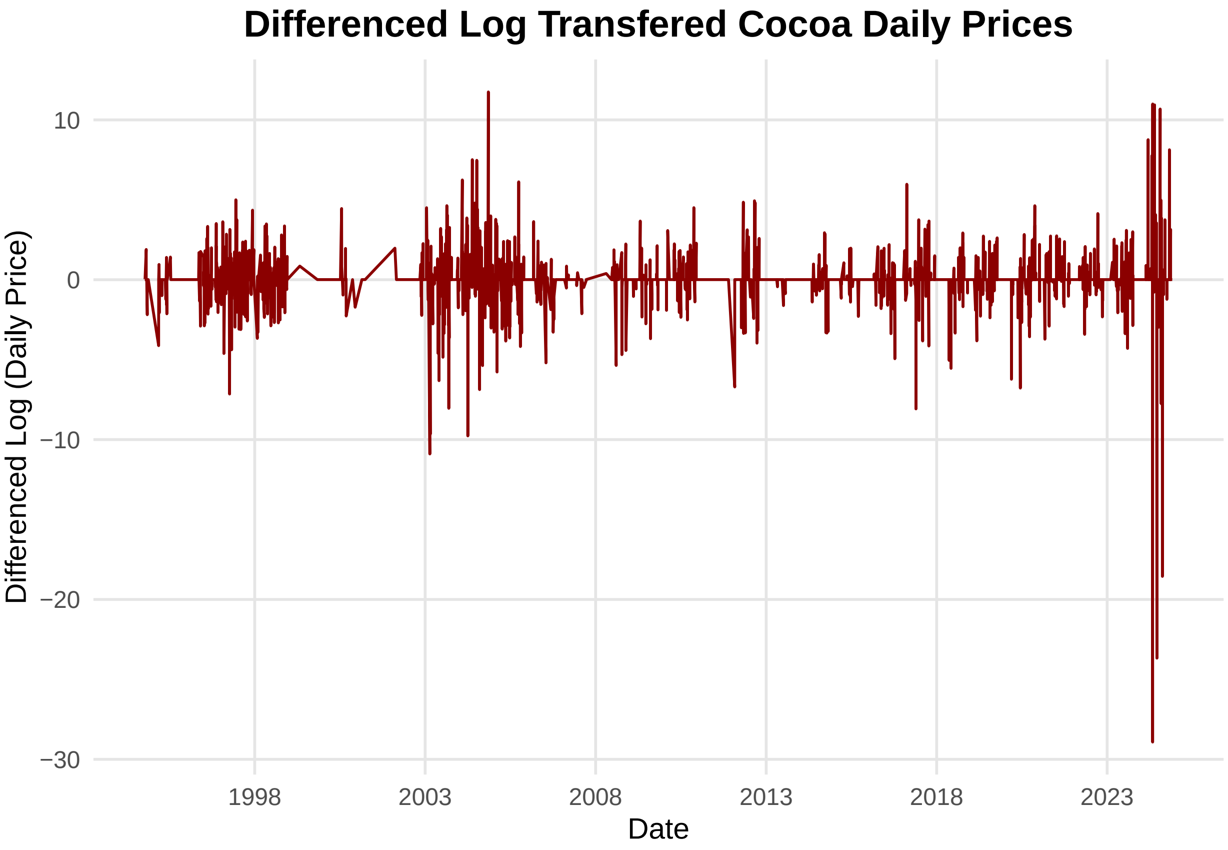
```
#plots
plot1 <- ggplot(cocoa_data, aes(x = Date)) +
  geom_line(aes(y = scale(Daily_price)), color = "darkblue") +
  labs(title = "Daily Cocoa Prices", x = "Date", y = "Daily Price") +
  scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    panel.background = element_rect(fill = "white", color = NA),
    plot.background = element_rect(fill = "white", color = NA),
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_blank()
  )
plot1
```



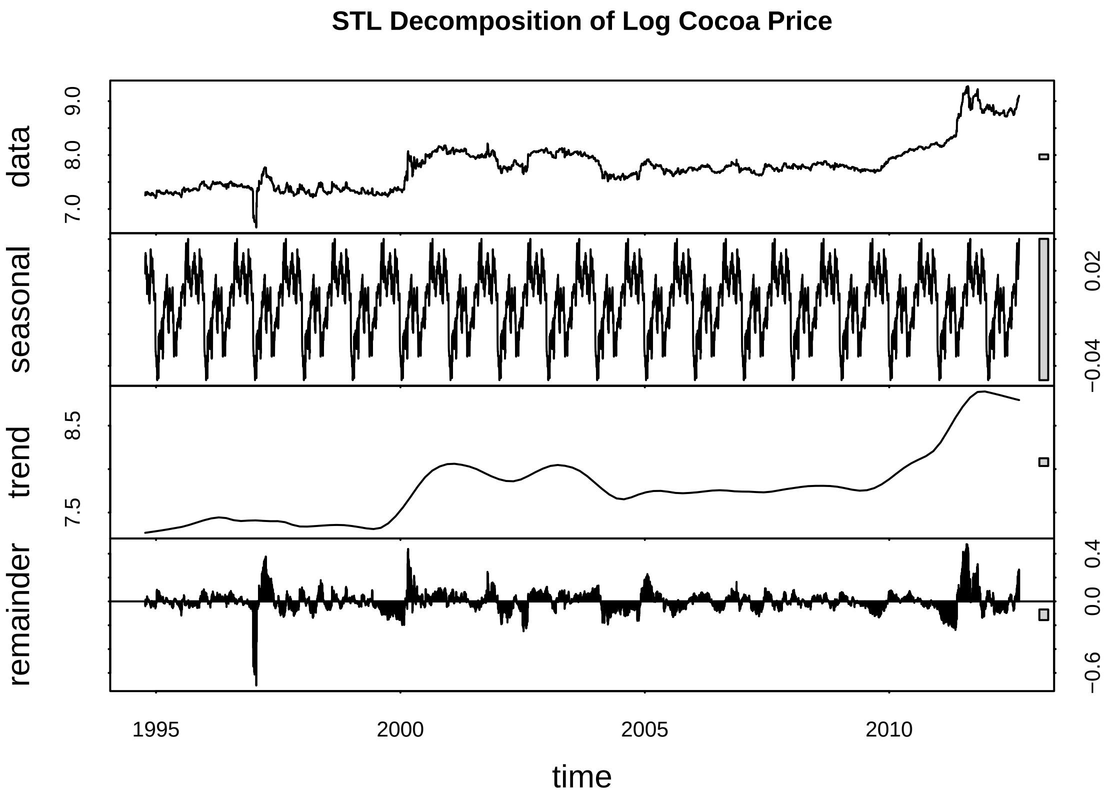
```
plot2 <- ggplot(cocoa_data, aes(x = Date)) +
  geom_line(aes(y = scale(log_price)), color = "darkgreen") +
  labs(title = "Log Transfered Cocoa Daily Prices", x = "Date", y = "Log (Daily Price)") +
  scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    panel.background = element_rect(fill = "white", color = NA),
    plot.background = element_rect(fill = "white", color = NA),
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_blank()
  )
plot2
```



```
plot3 <- ggplot(cocoa_data, aes(x = Date)) +
  geom_line(aes(y = scale(diff_log_price)), color = "darkred") +
  labs(title = "Differenced Log Transfered Cocoa Daily Prices", x = "Date", y = "Differenced Log (Daily
  scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    panel.background = element_rect(fill = "white", color = NA),
    plot.background = element_rect(fill = "white", color = NA),
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_blank()
  )
plot3
```


Differenced Log Transfered Cocoa Daily Prices

```
ts_log_price <- ts(cocoa_data$log_price, frequency = 365, start = c(year(min(cocoa_data$Date)), yday(mi
decomp <- stl(ts_log_price, s.window = "periodic")
plot(decomp, main = "STL Decomposition of Log Cocoa Price")
```



```
#Training data & Testing data
train_size <- floor(0.8 * nrow(cocoa_data))
train_data <- cocoa_data[1:train_size, ]
test_data <- cocoa_data[(train_size + 1):nrow(cocoa_data), ]
```

```
#ETS Model
ets_model_1 <- ets(train_data$diff_log_price)
ets_model_2 <- ets(train_data$diff_log_price, model = "ZZZ")
ets_model_1
```

```
## ETS(A,N,N)
##
## Call:
## ets(y = train_data$diff_log_price)
##
##   Smoothing parameters:
##     alpha = 1e-04
##
##   Initial states:
##     l = -1e-04
##
##   sigma:  0.0061
##
##       AIC      AICc       BIC
## -8470.236 -8470.231 -8450.555
```

```
ets_model_2
```

```
## ETS(A,N,N)
##
```

```
## Call:
## ets(y = train_data$diff_log_price, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 1e-04
##
##   Initial states:
##     l = -1e-04
##
##   sigma:  0.0061
##
##       AIC      AICc       BIC
## -8470.236 -8470.231 -8450.555
```

```
external_regressors_train <- train_data %>% select(PRCP, TAVG, TMAX, TMIN) %>% as.matrix()
external_regressors_test  <- test_data  %>% select(PRCP, TAVG, TMAX, TMIN) %>% as.matrix()

# ARIMAX
arimax_model <- auto.arima(train_data$diff_log_price,
                           xreg = external_regressors_train,
                           seasonal = FALSE)

# SARIMAX
sarimax_model <- auto.arima(train_data$diff_log_price,
                            xreg = external_regressors_train,
                            seasonal = TRUE)
arimax_model
```

```
## Series: train_data$diff_log_price
## Regression with ARIMA(2,0,2) errors
##
## Coefficients:
##          ar1      ar2      ma1     ma2   PRCP   TAVG  TMAX  TMIN
##       0.0667  -0.9651  -0.0766  0.9575  1e-04  0e+00     0     0
## s.e.  0.0206   0.0248   0.0229  0.0271  1e-04  1e-04     0     0
##
## sigma^2 = 3.773e-05:  log likelihood = 19183.84
## AIC=-38349.68   AICc=-38349.64   BIC=-38290.63
```

```
sarimax_model
```

```
## Series: train_data$diff_log_price
## Regression with ARIMA(2,0,2) errors
##
## Coefficients:
##          ar1      ar2      ma1     ma2   PRCP   TAVG  TMAX  TMIN
##       0.0667  -0.9651  -0.0766  0.9575  1e-04  0e+00     0     0
## s.e.  0.0206   0.0248   0.0229  0.0271  1e-04  1e-04     0     0
##
## sigma^2 = 3.773e-05:  log likelihood = 19183.84
## AIC=-38349.68   AICc=-38349.64   BIC=-38290.63
```

```
#forecast
ets_forecast_1 <- forecast(ets_model_1, h = nrow(test_data))
ets_forecast_2 <- forecast(ets_model_2, h = nrow(test_data))
forecast_arimax <- forecast(arimax_model, xreg = external_regressors_test)
```

```
forecast_sarimax <- forecast(sarimax_model, xreg = external_regressors_test)

ets_acc1 <- accuracy(ets_forecast_1, test_data$diff_log_price)
ets_acc2 <- accuracy(ets_forecast_2, test_data$diff_log_price)
arimax_acc <- accuracy(forecast_arimax, test_data$diff_log_price)
sarimax_acc <- accuracy(forecast_sarimax, test_data$diff_log_price)
```

```
print("EST Model 1 Performance:")
```

```
## [1] "EST Model 1 Performance:"
```

```
ets_acc1
```

```
##                         ME        RMSE         MAE MPE MAPE      MASE
## Training set 1.443680e-05 0.006145921 0.001883802 Inf  Inf 0.5380097
## Test set     9.943775e-05 0.010355974 0.001578565 Inf  Inf 0.4508348
##                     ACF1
## Training set -0.01163916
## Test set              NA
```

```
print("EST Model 2 Performance:")
```

```
## [1] "EST Model 2 Performance:"
```

```
ets_acc2
```

```
##                         ME        RMSE         MAE MPE MAPE      MASE
## Training set 1.443680e-05 0.006145921 0.001883802 Inf  Inf 0.5380097
## Test set     9.943775e-05 0.010355974 0.001578565 Inf  Inf 0.4508348
##                     ACF1
## Training set -0.01163916
## Test set              NA
```

```
print("ARIMAX Model Performance:")
```

```
## [1] "ARIMAX Model Performance:"
```

```
arimax_acc
```

```
##                         ME        RMSE         MAE MPE MAPE      MASE
## Training set 1.812991e-06 0.006137704 0.001994736 NaN  Inf 0.5696924
## Test set     1.001817e-04 0.010358191 0.001604912 NaN  Inf 0.4583595
##                     ACF1
## Training set -0.00193678
## Test set              NA
```

```
print("SARIMAX Model Performance:")
```

```
## [1] "SARIMAX Model Performance:"
```

```
sarimax_acc
```

```
##                         ME        RMSE         MAE MPE MAPE      MASE
## Training set 1.812991e-06 0.006137704 0.001994736 NaN  Inf 0.5696924
## Test set     1.001817e-04 0.010358191 0.001604912 NaN  Inf 0.4583595
##                     ACF1
## Training set -0.00193678
## Test set              NA
```

```r
models <- list("ETS Model 1" = ets_acc1,
               "ETS Model 2" = ets_acc2,
               "ARIMAX" = arimax_acc,
               "SARIMAX" = sarimax_acc)
best_model_name <- names(which.min(sapply(models, function(x) x[2])))
```

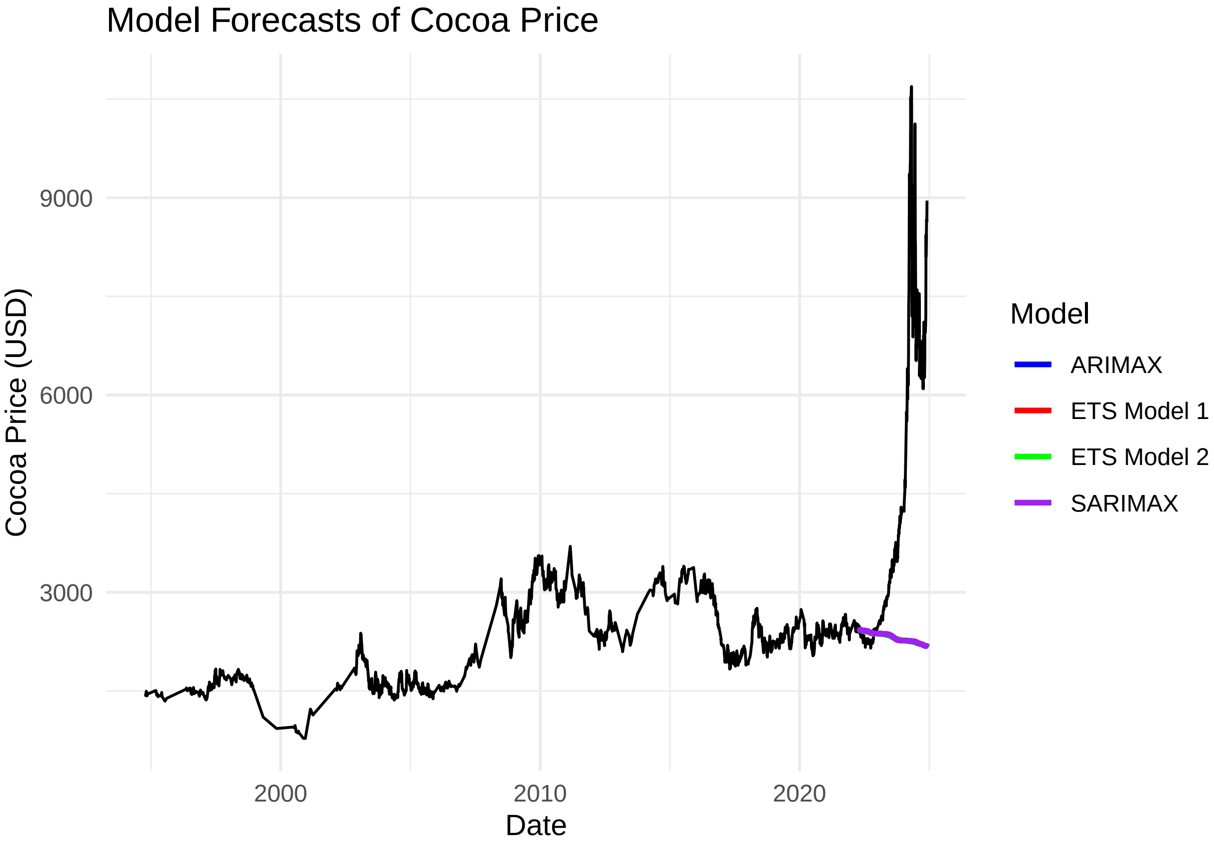
Best Model Based on RMSE:**`best_model_name`**

```r
re_log_prices <- function(last_log_price, diffs) {cumsum(c(last_log_price, diffs))[-1]}

last_log_price <- tail(train_data$log_price, 1)
n <- nrow(test_data)
forecast_dates <- test_data$Date

# Reconstruct log forecasts
ets1_log_forecast    <- re_log_prices(last_log_price, ets_forecast_1$mean)
ets2_log_forecast    <- re_log_prices(last_log_price, ets_forecast_2$mean)
arimax_log_forecast  <- re_log_prices(last_log_price, forecast_arimax$mean)
sarimax_log_forecast <- re_log_prices(last_log_price, forecast_sarimax$mean)

# Exponentiate to get actual price forecasts
ets1_price_forecast    <- exp(ets1_log_forecast)
ets2_price_forecast    <- exp(ets2_log_forecast)
arimax_price_forecast  <- exp(arimax_log_forecast)
sarimax_price_forecast <- exp(sarimax_log_forecast)

# Combine into dataframe
forecast_df <- tibble(Date = rep(forecast_dates, 4),
                      Forecast = c(ets1_price_forecast, ets2_price_forecast,
                                   arimax_price_forecast, sarimax_price_forecast),
                      Model = rep(c("ETS Model 1", "ETS Model 2", "ARIMAX", "SARIMAX"), each = n))
forecast_df
```

```
## # A tibble: 5,224 x 3
##    Date       Forecast Model
##    <date>        <dbl> <chr>
##  1 2022-04-25    2429. ETS Model 1
##  2 2022-04-26    2429. ETS Model 1
##  3 2022-04-27    2429. ETS Model 1
##  4 2022-04-28    2429. ETS Model 1
##  5 2022-04-28    2429. ETS Model 1
##  6 2022-04-28    2428. ETS Model 1
##  7 2022-04-28    2428. ETS Model 1
##  8 2022-04-29    2428. ETS Model 1
##  9 2022-04-29    2428. ETS Model 1
## 10 2022-04-29    2428. ETS Model 1
## # i 5,214 more rows
```

```r
plot4 <- ggplot(forecast_df, aes(x = Date, y = Forecast, color = Model)) +
  geom_line(data = cocoa_data, aes(x = Date, y = Daily_price), color = "black") +
  geom_line(data = forecast_df, aes(x = Date, y = Forecast, color = Model), linewidth = 1) +
  labs(title = "Model Forecasts of Cocoa Price",
       y = "Cocoa Price (USD)", x = "Date") +
  theme_minimal() +
  scale_color_manual(values = c("blue","red","green","purple"))
```

```
plot4
```


Model Forecasts of Cocoa Price
Cocoa Price (USD)
9000
6000
3000
2000
2010
2020
Date
Model
ARIMAX
ETS Model 1
ETS Model 2
SARIMAX