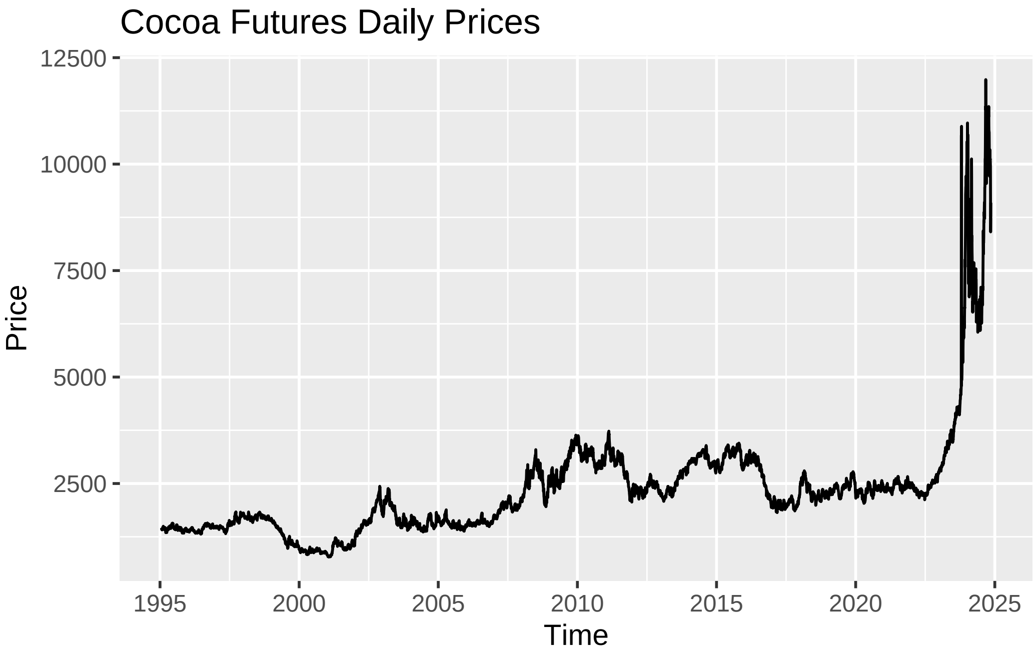
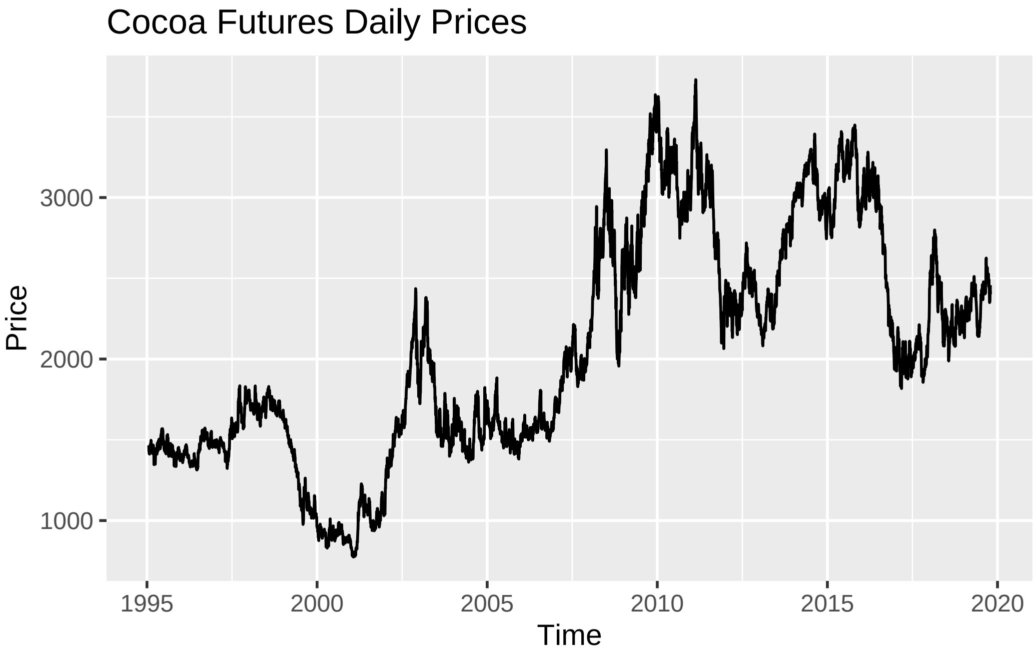
457finalunscaled

```r
# Load libraries
library(readr)
library(lubridate)
library(forecast)
library(ggplot2)
library(tseries)
library(tsoutliers)
library(vars)
```

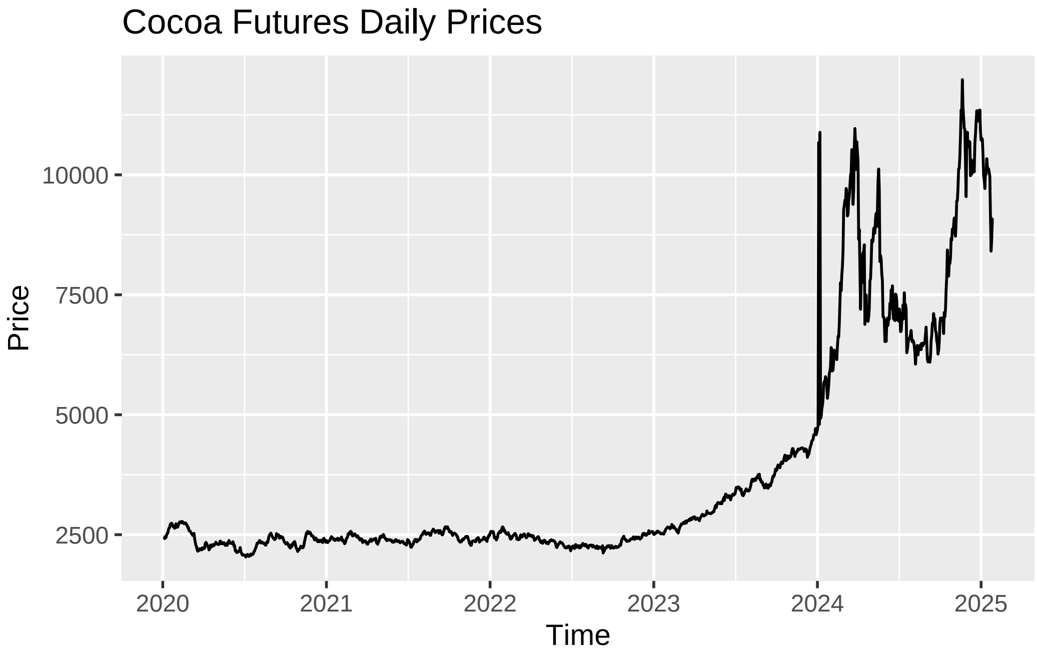
```r
# 1. Data Import and Preprocessing

# Read the CSV file (ensure your working directory is set correctly)
data <- read_csv("Daily Prices_ICCO (1).csv", show_col_types = FALSE)
data$Date <- as.Date(data$Date, format = "%d/%m/%Y")

# Make sure the data is in ascending order by Date
data <- data[order(data$Date), ]


# Calculate start_year and start_doy before creating the time series object
start_year <- year(min(data$Date))
start_doy  <- yday(min(data$Date))

# Create a time series object for daily data
price_ts <- ts(data$`ICCO daily price (US$/tonne)`, frequency = 262, start = c(start_year, s

# Plot the raw time series
autoplot(price_ts) +
  ggtitle("Cocoa Futures Daily Prices") +
  xlab("Time") +
  ylab("Price")
```



```r
# 2. Split the Data by Date (Training: Before 2020-01-01, Test: From 2020-01-01 Onward)

# Create training and test subsets from the original data frame
train_data <- data[data$Date < as.Date("2020-01-01"), ]
test_data  <- data[data$Date >= as.Date("2020-01-01"), ]

# Convert the training subset into a time series
train_start_year <- year(min(train_data$Date))
train_start_doy  <- yday(min(train_data$Date))
train_ts <- ts(train_data$`ICCO daily price (US$/tonne)`, frequency = 262, start = c(train_s
autoplot(train_ts)+
  ggtitle("Cocoa Futures Daily Prices") +
  xlab("Time") +
  ylab("Price")
```



```
# Convert the test subset into a time series
test_start_year <- year(min(test_data$Date))
test_start_doy  <- yday(min(test_data$Date))
test_ts <- ts(test_data$`ICCO daily price (US$/tonne)`, frequency = 262, start = c(test_star

autoplot(test_ts)+
  ggtitle("Cocoa Futures Daily Prices") +
  xlab("Time") +
  ylab("Price")
```



```r
# Check for variance stabilization need using a Box-Cox transformation
lambda <- BoxCox.lambda(train_ts)
cat("Estimated Box-Cox Lambda:", lambda, "\n")
```

```
Estimated Box-Cox Lambda: 0.4277683
```

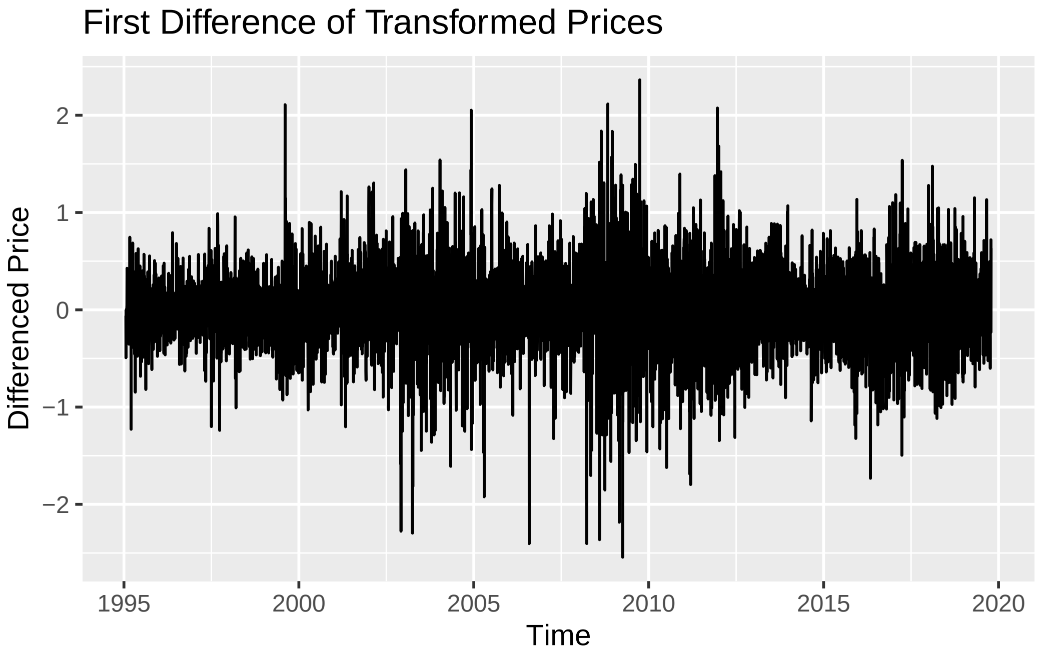
```r
# Apply Box-Cox transformation if lambda is not 1 (i.e., non-linear variance)
if(abs(lambda - 1) > 0.1){
  train_ts <- BoxCox(train_ts, lambda)
} else {
  train_ts <- price_ts
}
```

```r
# 3. Stationarity Check

# ADF Test on the transformed series
adf_result <- adf.test(train_ts)
print(adf_result)
```

```

    Augmented Dickey-Fuller Test
```

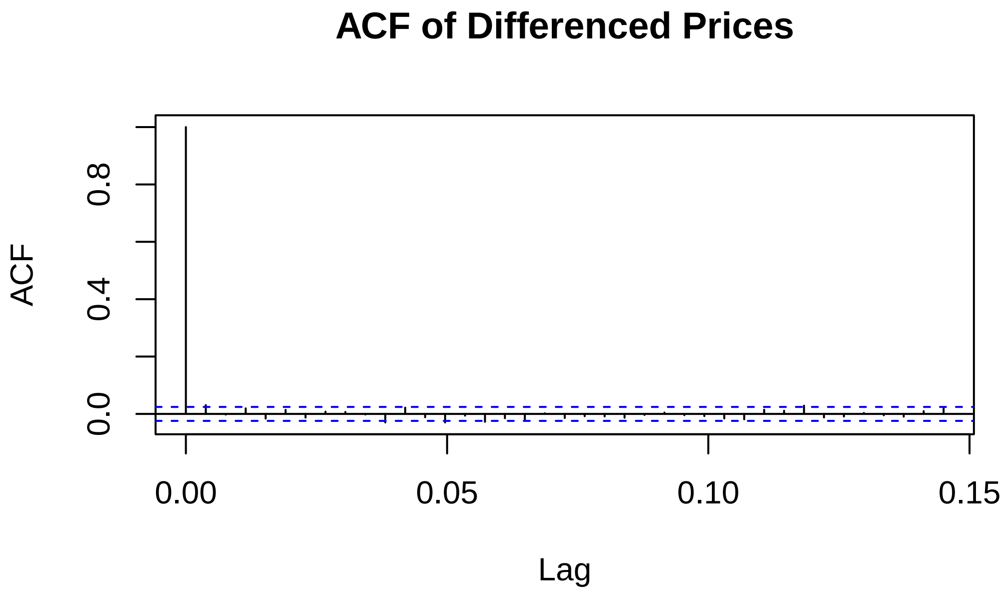
```
data:  train_ts
Dickey-Fuller = -2.4771, Lag order = 18, p-value = 0.3763
alternative hypothesis: stationary
```

```
# Compute the differenced series if the ADF test suggests non-stationarity
if(adf_result$p.value > 0.05){
  train_diff <- diff(train_ts)
} else {
  train_diff <- train_ts
}

autoplot(train_diff) +
  ggtitle("First Difference of Transformed Prices") +
  xlab("Time") +
  ylab("Differenced Price")
```

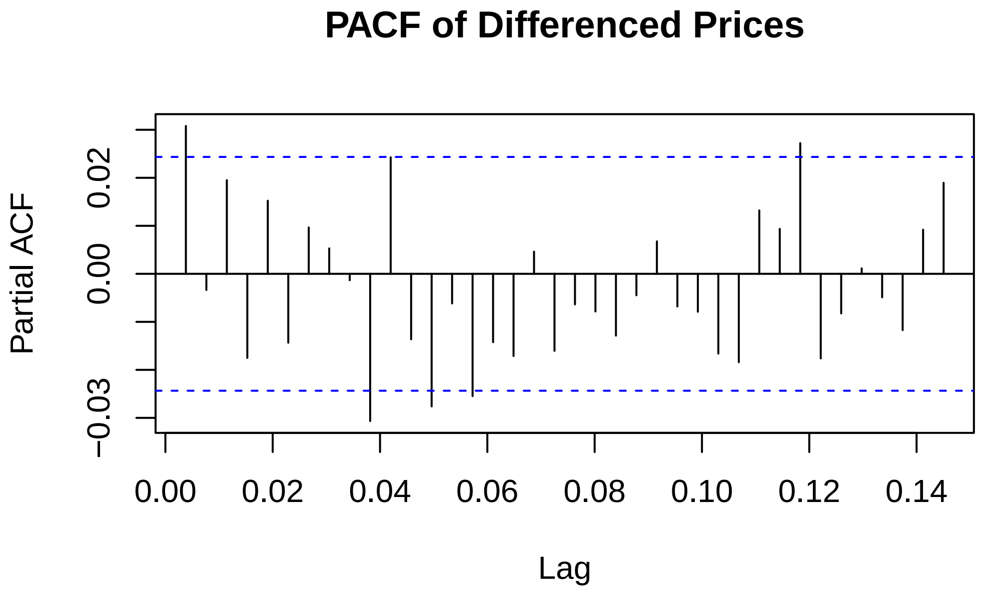


```
train_ts_acf <- ts(train_data$`ICCO daily price (US$/tonne)`, frequency = 1, start = c(train_
if(adf_result$p.value > 0.05){
  train_diff_acf <- diff(train_ts_acf)
} else {
  train_diff_acf <- train_ts_acf
}
```

```r
acf(train_diff, main = "ACF of Differenced Prices")
```



```r
pacf(train_diff, main = "PACF of Differenced Prices")
```



```
# Manual candidate models based on ACF/PACF interpretation
# ARIMA(1,1,0), ARIMA(0,1,1), ARIMA(1,1,1)
candidate_415 <- arima(train_ts, order = c(1, 1, 0))
candidate_410 <- arima(train_ts, order = c(4, 1, 0))
candidate_015 <- arima(train_ts, order = c(0, 1, 5))
candidate_212 <- arima(train_ts, order = c(2, 1, 2))
```