# Assignment 4 Forecasting

12 November 2025

```
options(warn = -1)
library(quantmod)
library(fpp3)
library(readr)
library(broom)
library(tsibble)
library(dplyr)
library(stringr)
library(lubridate)
```

## 1 soybean futures time series analysis

This document is an investigation into time series forecasting of Soybean futures closing price from the Chicago Board of Trade (CBOT). I will be using the Close price.

The futures prices are quoted as “front month” prices which means that the prices for “rolls” from one contract to the next as they expire. When this rollover happens it is due to different expiration months, storage costs, contango, backwardation. In our case we will verify this by looking at how similar Adjusted and Close prices are to each other

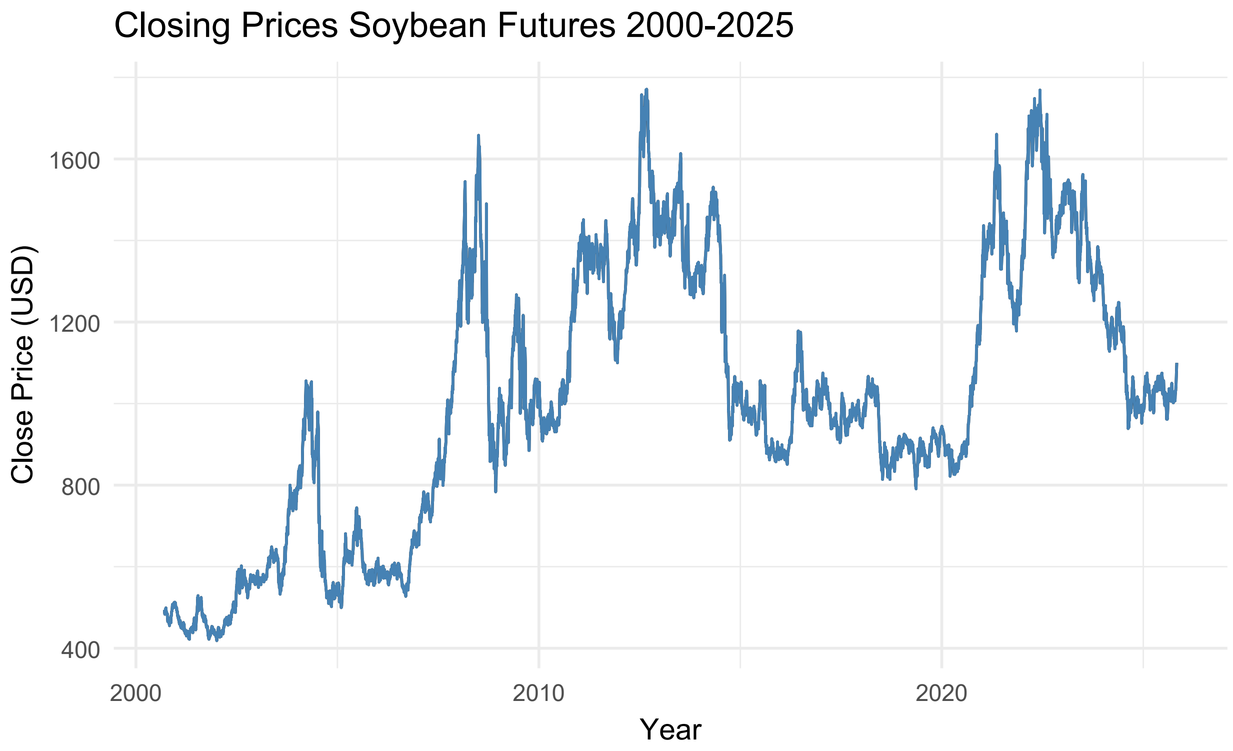
As our data is economic data, the possibility of adjusting for inflation arises. Fortunately due to the nature of how futures contracts are priced, forward looking interest changes, storage costs and inflation are already priced into the price when they are calculated. We care about nominal prices which reflect market behavior. For missing data, fill the previous data

```
getSymbols("ZS=F", src = "yahoo", from = "2000-01-01", to = "2025-11-01", auto.assign = TR
```

```
## [1] "ZS=F"
```

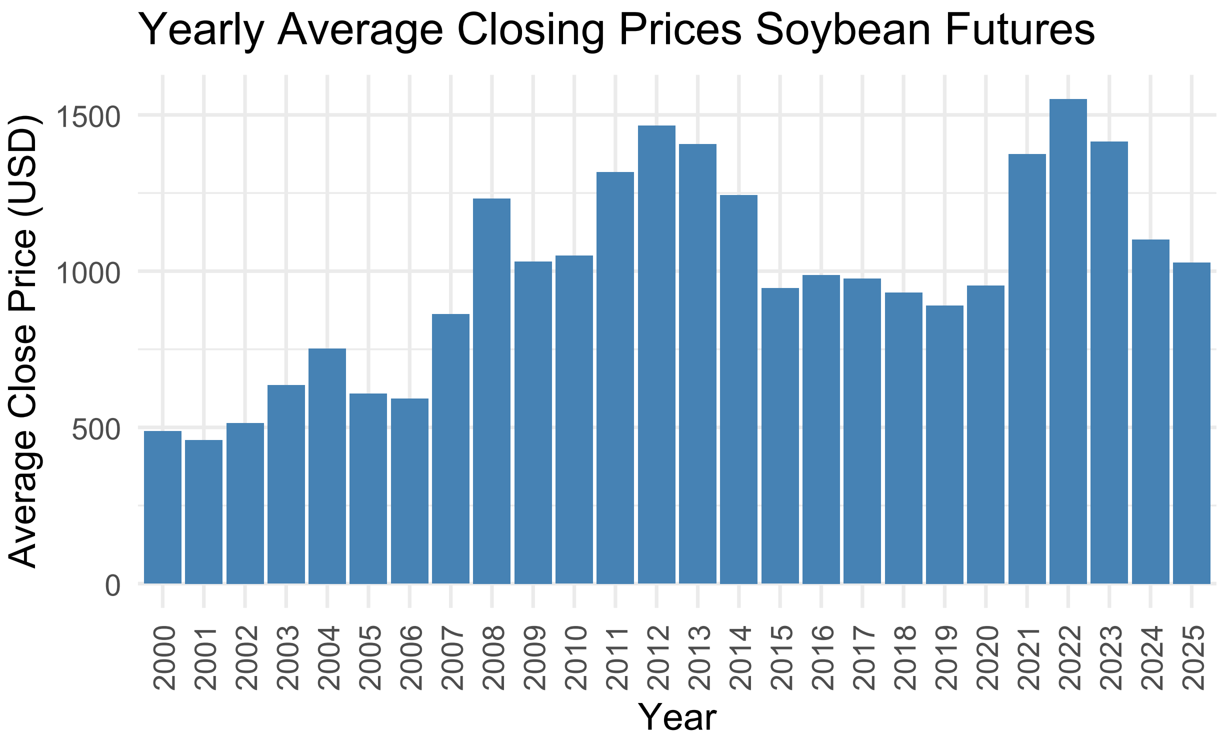
```
soybean_tibble <- tidy(`ZS=F`)

soybean_wide <- soybean_tibble %>%
  mutate(index = as.Date(index)) %>%
  pivot_wider(
    names_from = series,
    values_from = value
  )


soybean_tsibble <- as_tsibble(soybean_wide, index = index, regular = FALSE)


soybean_tsibble <- soybean_tsibble %>%
  rename_with(~ str_replace(., "ZS=F\\.", ""), .cols = -index)

soybean_tsibble <- soybean_tsibble |>
  fill(Open:Adjusted, .direction = "down")

write_csv(soybean_tsibble, "soybean_data.csv")
```

```
soybean_tsibble |>
  autoplot(Close) +
  geom_line(color  = "steelblue")+
  labs(
    title = "Closing Prices Soybean Futures 2000-2025",
    x = "Year",
    y = "Close Price (USD)"
  ) +
  theme_minimal()
```



## 2 Average yearly prices

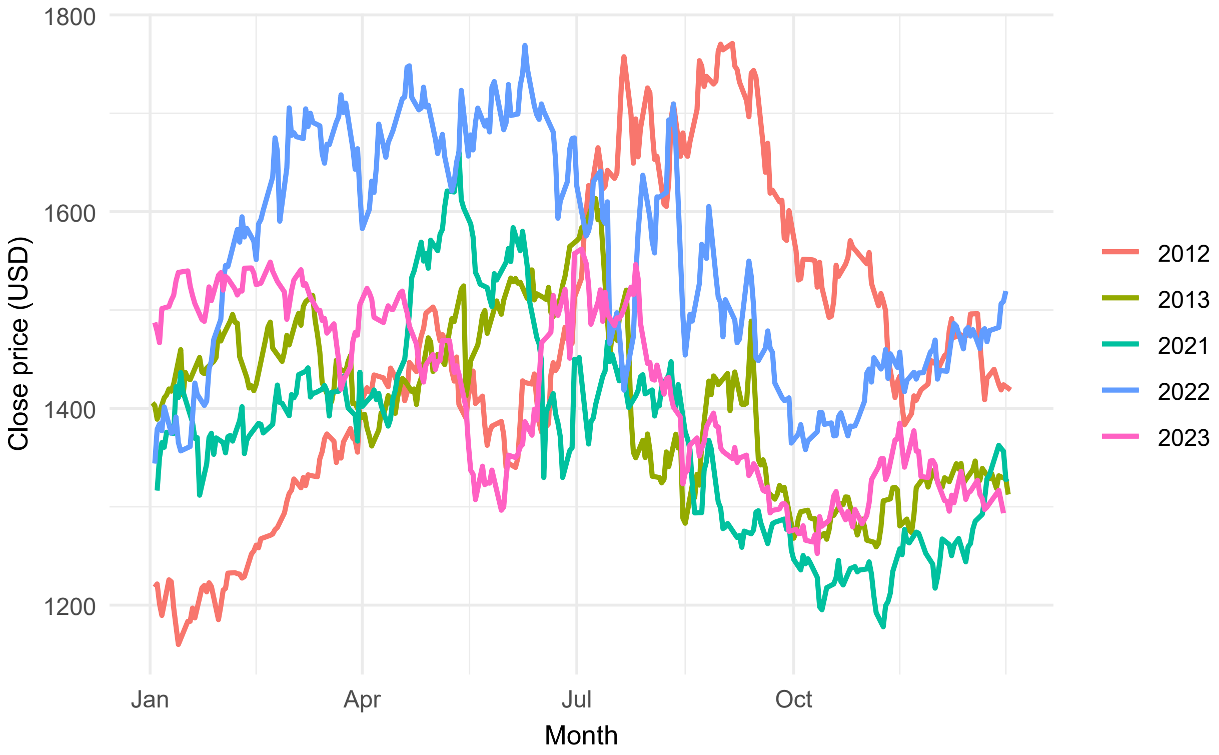
```
yearly_soy <- soybean_tsibble |>
  index_by(Year = year(index)) |>
  summarise(across(where(is.numeric), \(x) mean(x, na.rm = TRUE)))

ggplot(yearly_soy, aes(x = factor(Year), y = Close)) +
  geom_col(fill = "steelblue") +
  labs(
    title = "Yearly Average Closing Prices Soybean Futures",
    x = "Year",
    y = "Average Close Price (USD)"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5))
```



## 3 Top 5 years by average price

```
# Find the top 5 years
top_years <- yearly_soy |>
```

```
  slice_max(order_by = Close, n =5) |>
  pull(Year)

# Based on the top_years vector, pass in the tsibble and filter the index for years that m
best_years <- soybean_tsibble %>%
  filter(year(index) %in% top_years)

# Plot it
best_years %>%
  gg_season(Close) +
  geom_line(size = 0.9) +
   xlab("Month") +
  ylab("Close price (USD)") +
  ggtitle("Soybean Closing price 2000-2025 best performing years")
```



```
# Aggregated to monthly
soybean_monthly <- soybean_tsibble %>%
  index_by(month = ~ yearmonth(.)) %>%
  summarise(Close = mean(Close, na.rm = TRUE))

# Compute features / diagnostics dynamically
spectral_entropy <- feat_spectral(soybean_monthly$Close)
bp_stat <- box_pierce(soybean_monthly$Close)[1]      # Box-Pierce statistic
bp_p <- box_pierce(soybean_monthly$Close)[2]         # p-value
kpss_stat <- unitroot_kpss(soybean_monthly$Close)[1]
```

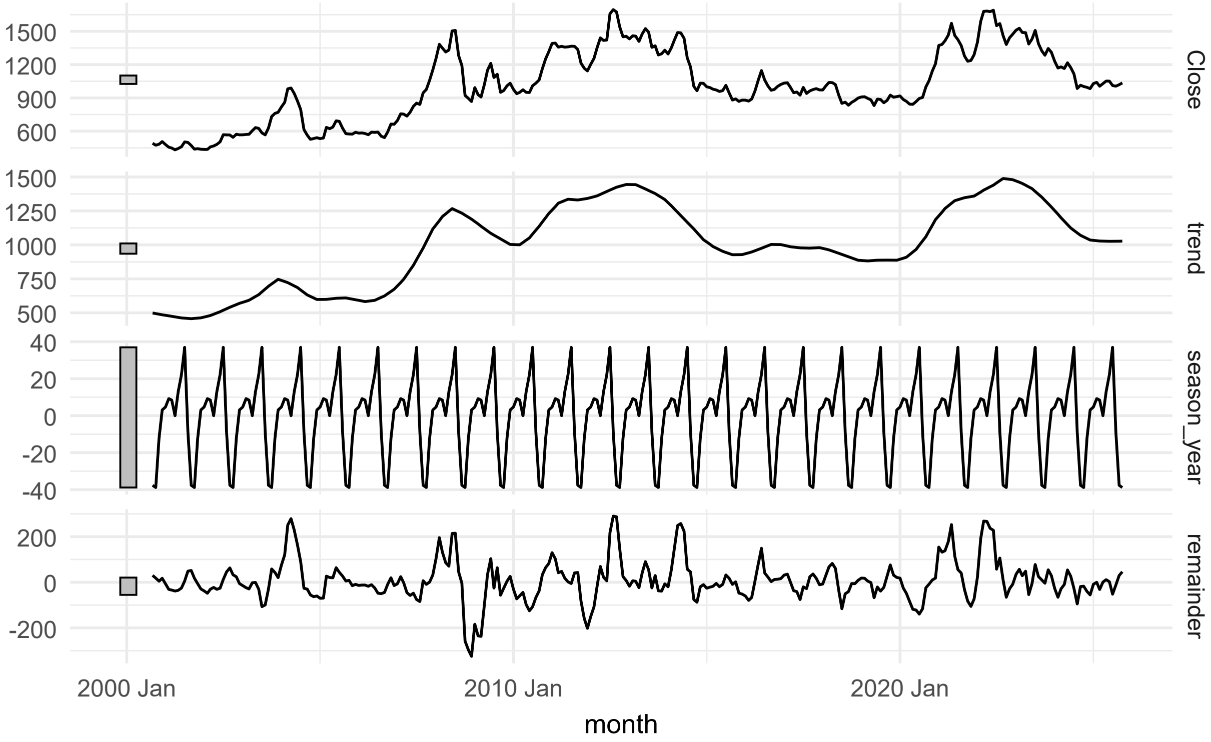
```r
kpss_p <- unitroot_kpss(soybean_monthly$Close)[2]
nd <- unitroot_ndiffs(soybean_monthly$Close)
nsd <- unitroot_nsdiffs(soybean_monthly$Close)
lambda <- guerrero(soybean_monthly$Close)
lb_stat <- ljung_box(soybean_monthly$Close)[1]
lb_p <- ljung_box(soybean_monthly$Close)[2]

# Combine into a tidy table
soybean_diagnostics <- tibble(
  Metric = c(
    "Spectral Entropy",
    "Breusch-Pagan Statistic",
    "Breusch-Pagan p-value",
    "KPSS Statistic",
    "KPSS p-value",
    "Number of differences (ndiffs)",
    "Number of seasonal differences (nsdiffs)",
    "Box-Cox lambda (Guerrero)",
    "Ljung-Box Statistic",
    "Ljung-Box p-value"
  ),
  Value = c(
    spectral_entropy,
    bp_stat,
    bp_p,
    kpss_stat,
    kpss_p,
    nd,
    nsd,
    lambda,
    lb_stat,
    lb_p
  )
)

# Display as a neat kable
kable(soybean_diagnostics,
      caption = "Diagnostics for Monthly Soybean Futures Prices",
      digits = 4,
      format = "markdown")
```

Table 1: Diagnostics for Monthly Soybean Futures Prices

| Metric | Value |
|---|---:|
| Spectral Entropy | 0.4914 |
| Breusch-Pagan Statistic | 288.9028 |
| Breusch-Pagan p-value | 0.0000 |
| KPSS Statistic | 2.1572 |
| KPSS p-value | 0.0100 |
| Number of differences (ndiffs) | 1.0000 |
| Number of seasonal differences (nsdiffs) | 0.0000 |
| Box-Cox lambda (Guerrero) | -0.0016 |
| Ljung-Box Statistic | 291.7822 |
| Ljung-Box p-value | 0.0000 |

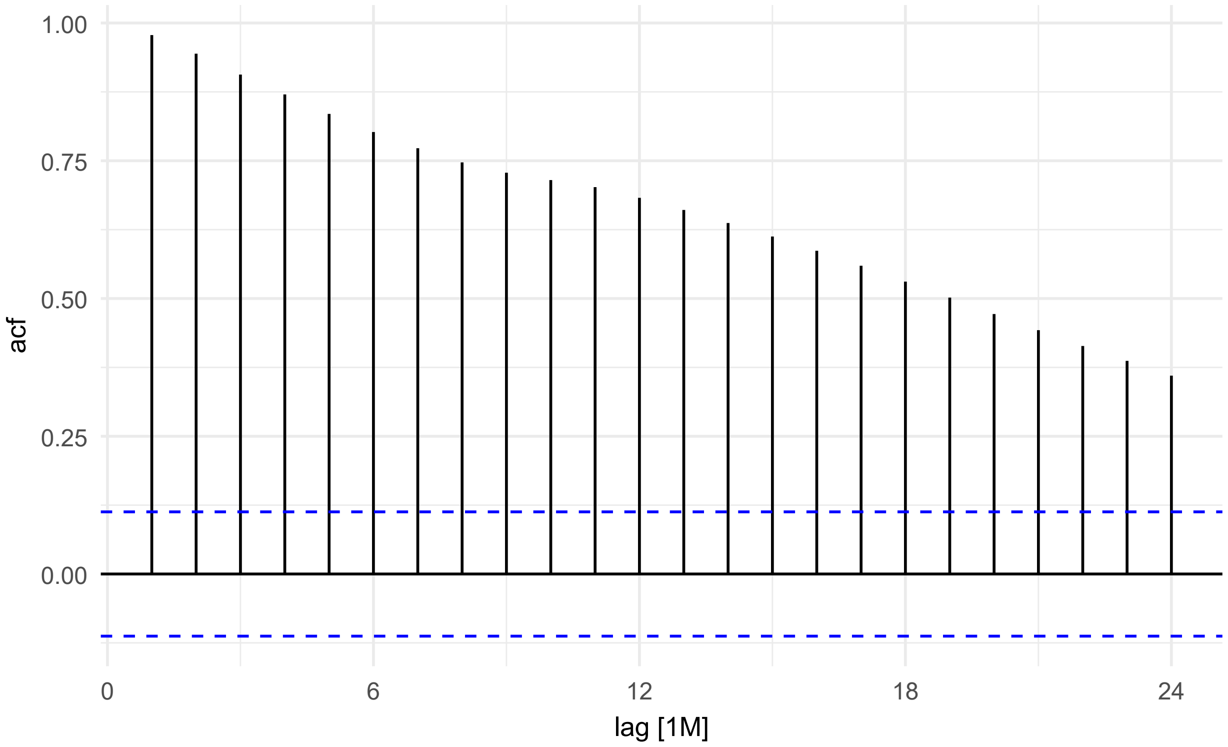
We can see there needs to be differencing on the data because its not stationary # Time series decomposition

```r
soybean_monthly  |>
  model(
    STL(Close ~ trend(window=21) +
                  season(window='periodic'),
    robust = TRUE)) |>
  components() |>
  autoplot()
```

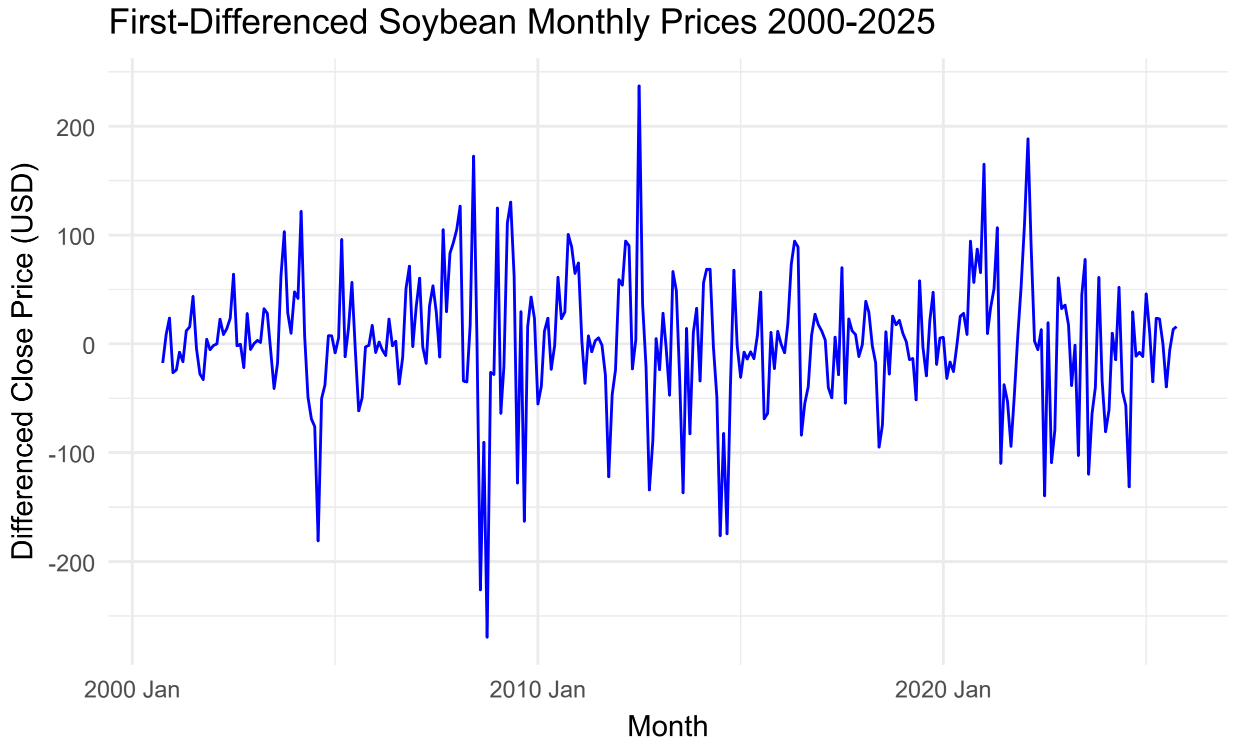


We can see the series is non stationary

```
soybean_monthly %>%
  ACF(Close) %>%
  autoplot()
```



```
soybean_diff <- soybean_monthly %>%
  mutate(Close_diff = difference(Close, lag = 1))

ggplot(soybean_diff, aes(x = month, y = Close_diff)) +
  geom_line(color = "blue") +
  labs(title = "First-Differenced Soybean Monthly Prices 2000-2025",
       x = "Month",
       y = "Differenced Close Price (USD)") +
  theme_minimal()
```



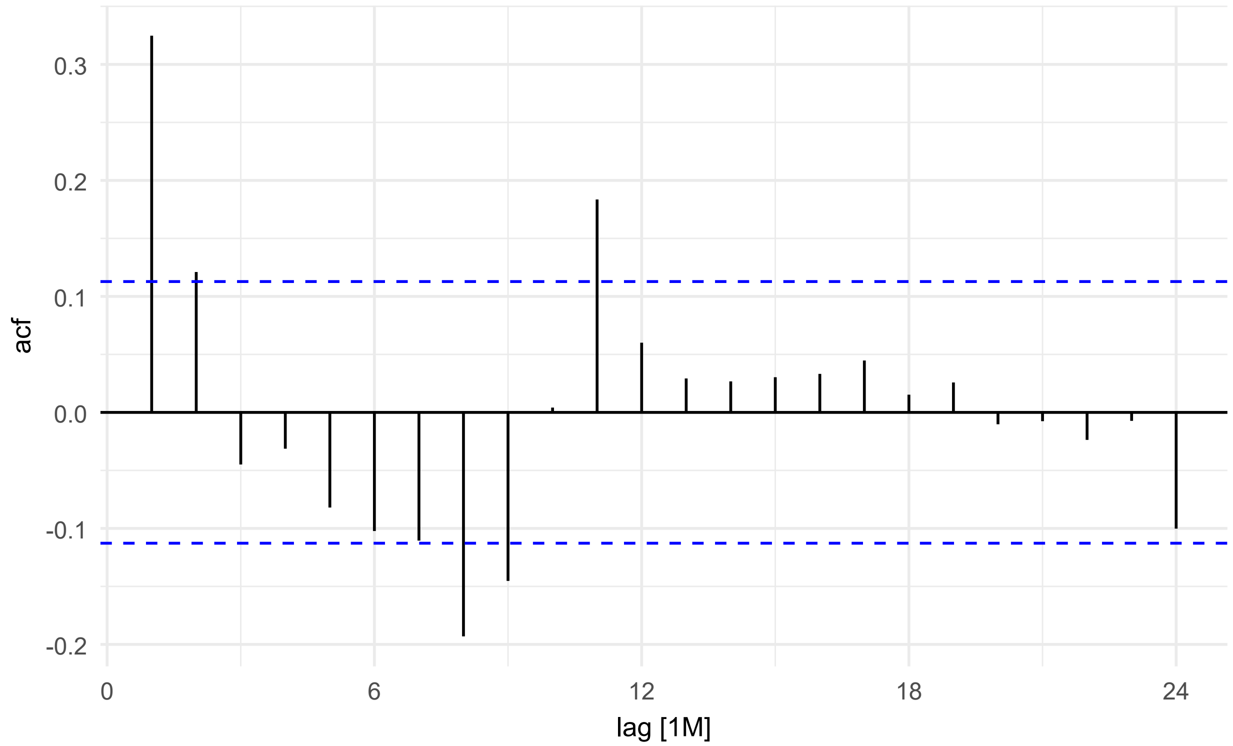
```
unitroot_kpss(soybean_diff$Close_diff)
```

```
##   kpss_stat kpss_pvalue
##  0.06976841  0.10000000
```

```
unitroot_ndiffs(soybean_diff$Close_diff)
```

```
## ndiffs
##      0
```

```
soybean_diff %>%
  ACF(Close_diff) %>%
  autoplot()
```



```
models <- soybean_monthly %>%
  model(
    SNAIVE = SNAIVE(Close),
    ETS    = ETS(Close),
    ARIMA  = ARIMA(Close),NNETAR(sqrt(Close))

  )
model_names <- names(models)

#models <- it_trans %>%
#  model(
#    ARIMA1 = ARIMA(Employed_tr ~ pdq(0,2,1)),
#    ARIMA2 = ARIMA(Employed_tr ~ pdq(0,2,2)),
#    ARIMA3 = ARIMA(Employed_tr ~ pdq(1,2,1)),
#    ARIMA4 = ARIMA(Employed_tr ~ pdq(2,2,2)))

models %>%
  glance() %>%
  arrange(AICc)
```

```
## # A tibble: 4 x 11
##   .model          sigma2 log_lik   AIC  AICc   BIC   MSE   AMSE     MAE ar_roots
##   <chr>            <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl> <list>
## 1 ARIMA          3.38e+3  -1649. 3304. 3304. 3315.    NA     NA  NA     <cpl>
## 2 ETS            3.22e-3  -2063. 4137. 4138. 4160. 3575. 10996.  0.0420 <NULL>
## 3 SNAIVE         5.80e+4     NA    NA    NA    NA    NA     NA  NA      <NULL>
## 4 NNETAR(sqrt(C~ 7.17e-1     NA    NA    NA    NA    NA     NA  NA      <NULL>
## # i 1 more variable: ma_roots <list>
```