

# corn.ab.R

harsh

2023-07-24

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.1
```

```
## Warning: package 'forcats' was built under R version 4.3.1
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.2      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2    3.4.2      ✓ tibble     3.2.1
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ purrr      1.0.1
## — 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(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.3.1
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.1
```

```
## Registered S3 method overwritten by 'quantmod':  
##   method           from  
##   as.zoo.data.frame zoo
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.3.1
```

```
library(gridExtra)
```

```
##  
## Attaching package: 'gridExtra'  
##  
## The following object is masked from 'package:dplyr':  
##  
##   combine
```

```
library(rugarch)
```

```
## Warning: package 'rugarch' was built under R version 4.3.1
```

```
## Loading required package: parallel  
##  
## Attaching package: 'rugarch'  
##  
## The following object is masked from 'package:purrr':  
##  
##   reduce  
##  
## The following object is masked from 'package:stats':  
##  
##   sigma
```

```
setwd("C:\\Users\\harsh\\OneDrive\\Desktop\\SCMA")
getwd()
```

```
## [1] "C:/Users/harsh/OneDrive/Desktop/SCMA"
```

```
df = read.csv("CORN.csv")

names(df)
```

```
## [1] "Date"      "Price"     "Open"      "High"      "Low"       "Vol."      "Change.."
```

```
head(df)
```

```
##           Date Price  Open  High   Low Vol. Change..
## 1 10-06-2010 25.46 25.46 25.46 25.46 0.20K   1.23%
## 2 11-06-2010 25.79 25.88 25.88 25.79 0.50K   1.30%
## 3 14-06-2010 26.11 25.99 26.11 25.99 2.25K   1.24%
## 4 15-06-2010 25.97 26.24 26.24 25.97 6.96K  -0.54%
## 5 16-06-2010 26.32 26.26 26.44 26.20 2.40K   1.35%
## 6 17-06-2010 26.08 26.20 26.20 25.82 1.60K  -0.91%
```

```
tail(df)
```

```
##           Date Price  Open  High   Low Vol. Change..
## 3322 14-07-2023 23.16 22.91 23.17 22.60 172.71K   2.34%
## 3323 17-07-2023 22.80 23.28 23.28 22.72  79.77K  -1.55%
## 3324 18-07-2023 23.84 23.15 24.12 23.07 210.36K   4.56%
## 3325 19-07-2023 24.59 24.79 24.98 24.10 211.42K   3.15%
## 3326 20-07-2023 24.35 24.75 24.82 24.19  83.99K  -0.98%
## 3327 21-07-2023 23.96 23.77 24.32 23.71 131.97K  -1.60%
```

```
df$Date <- as.Date(df$Date, format = '%m-%d-%Y')
df= df[order(df$Date),]
head(df)
```

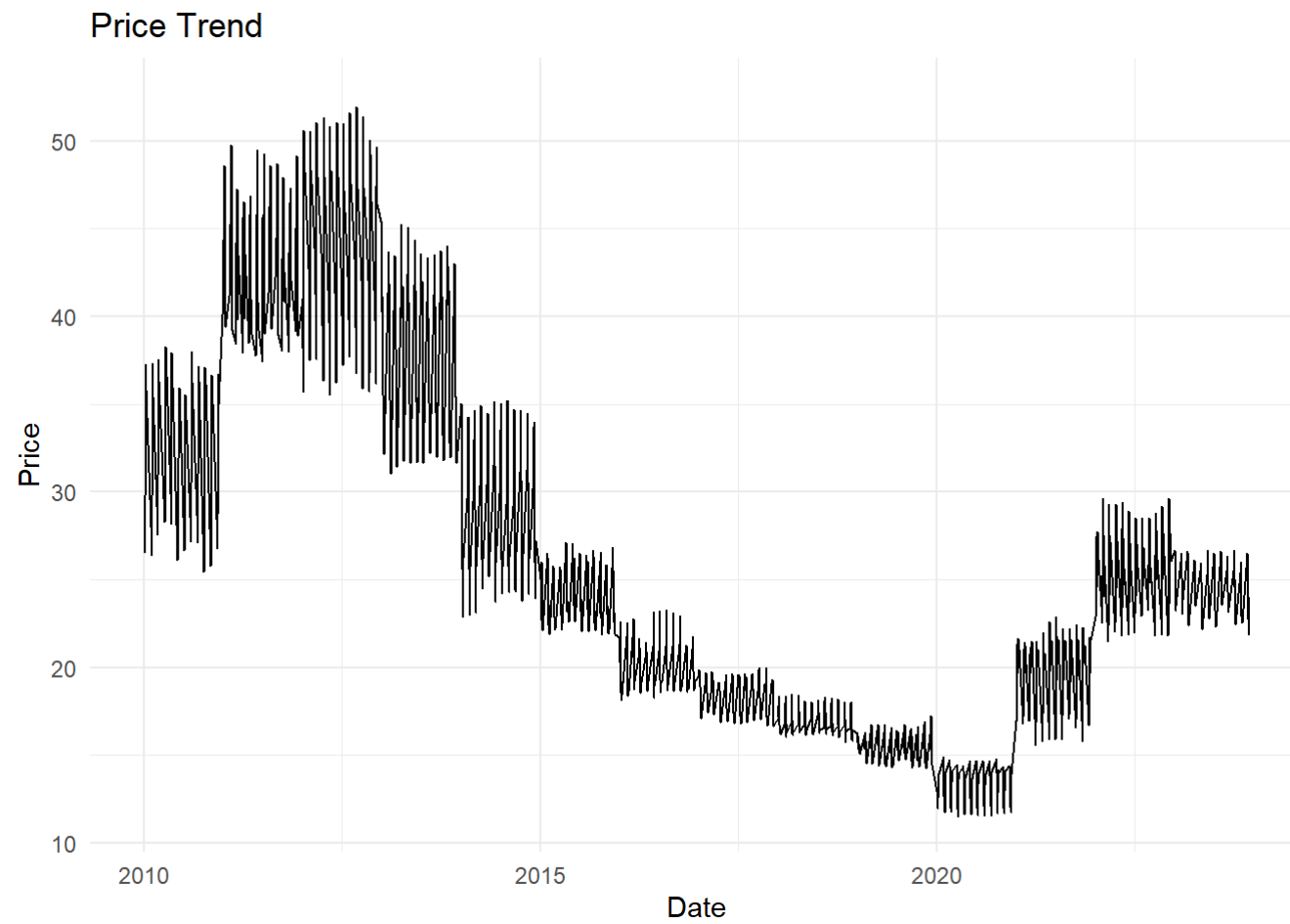
```
##           Date Price  Open  High   Low  Vol. Change..
## 16  2010-01-07 26.51 25.77 26.57 25.77 31.10K    2.20%
## 59  2010-01-09 29.80 29.70 30.02 29.50 25.73K    1.12%
## 80  2010-01-10 30.67 32.08 32.44 30.50 75.58K   -5.83%
## 101 2010-01-11 37.25 37.75 37.81 37.07 48.12K   -0.61%
## 122 2010-01-12 35.80 35.00 35.93 34.63 48.41K    3.74%
## 17  2010-02-07 26.39 26.52 26.58 26.17  6.40K   -0.45%
```

```
View(df)
df$Price <- as.numeric(df$Price)
df$Open <- as.numeric(df$Open)
df$High <- as.numeric(df$High)
df$Low <- as.numeric(df$Low)
missing_values <- is.na(df$Price)
missing_values_open <- is.na(df$Open)
missing_values_high <- is.na(df$High)
missing_values_low <- is.na(df$Low)
df$Price[missing_values] <- mean(df$Price, na.rm = TRUE)
df$Open[missing_values_open] <- mean(df$Open, na.rm = TRUE)
df$High[missing_values_high] <- mean(df$High, na.rm = TRUE)
df$Low[missing_values_low] <- mean(df$Low, na.rm = TRUE)
```

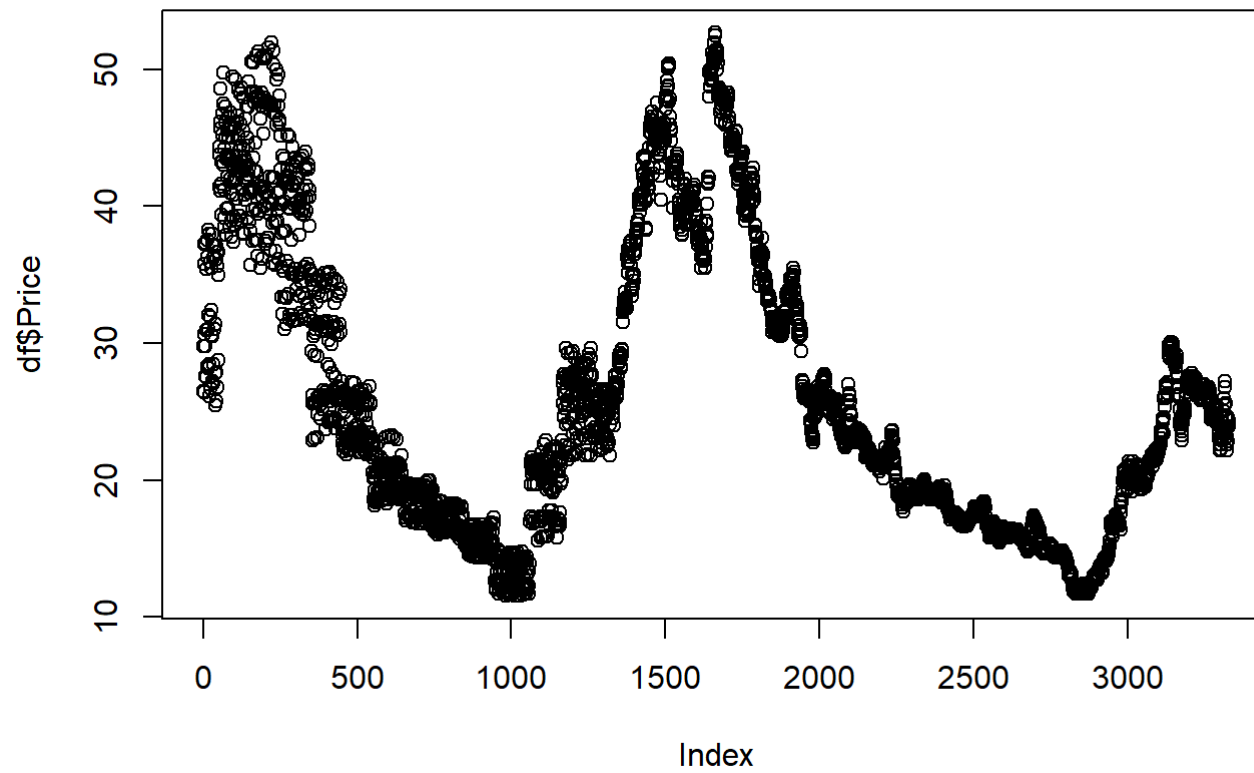
```
library(ggplot2)
```

```
# Create a line plot for the "Price" column
ggplot(df, aes(x = Date, y = Price)) +
  geom_line() +
  labs(x = "Date", y = "Price", title = "Price Trend") +
  theme_minimal()
```

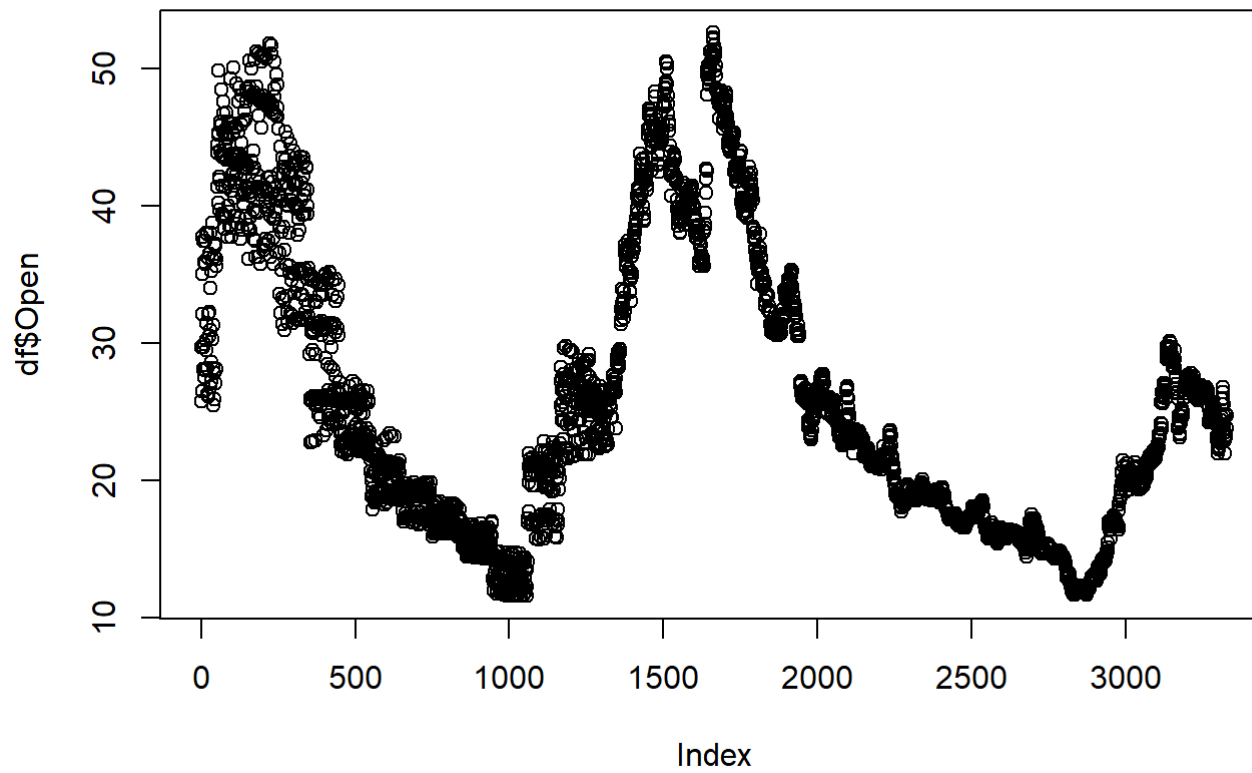
```
## Warning: Removed 2007 rows containing missing values (`geom_line()`).
```



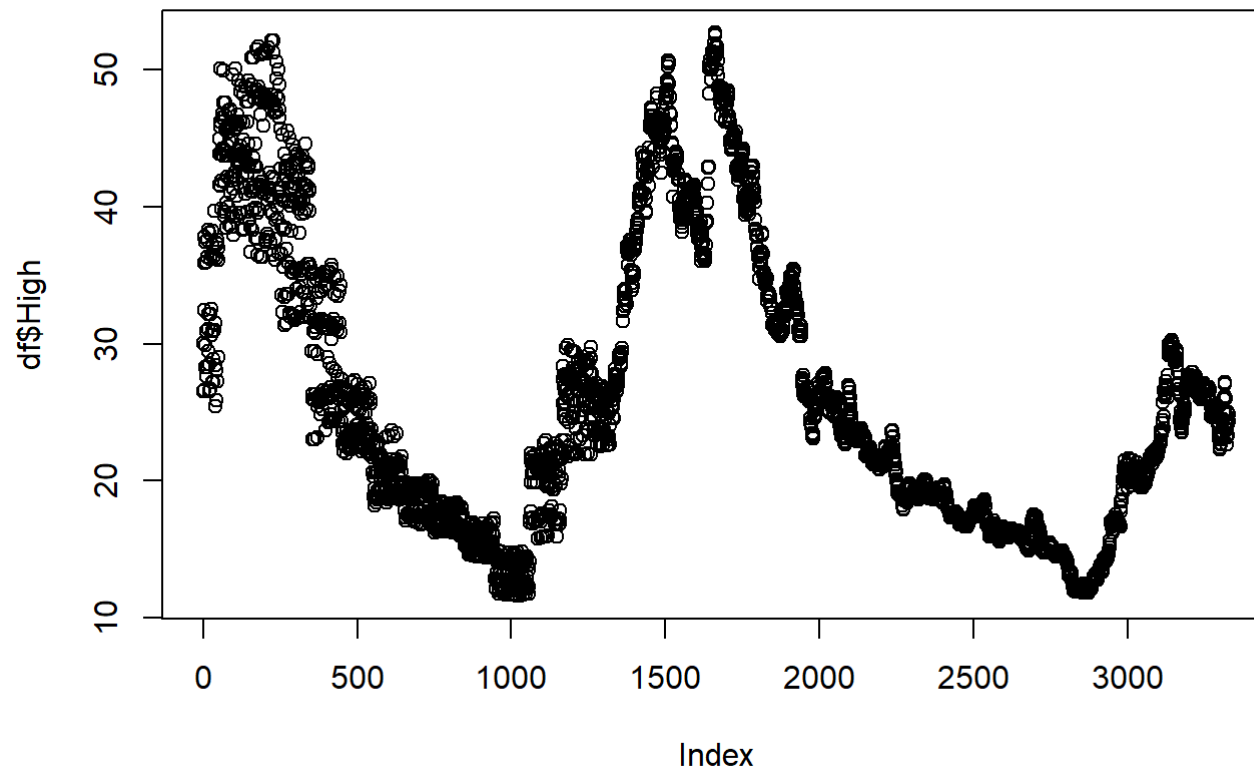
```
plot(df$Price)
```



```
plot(df$Open)
```

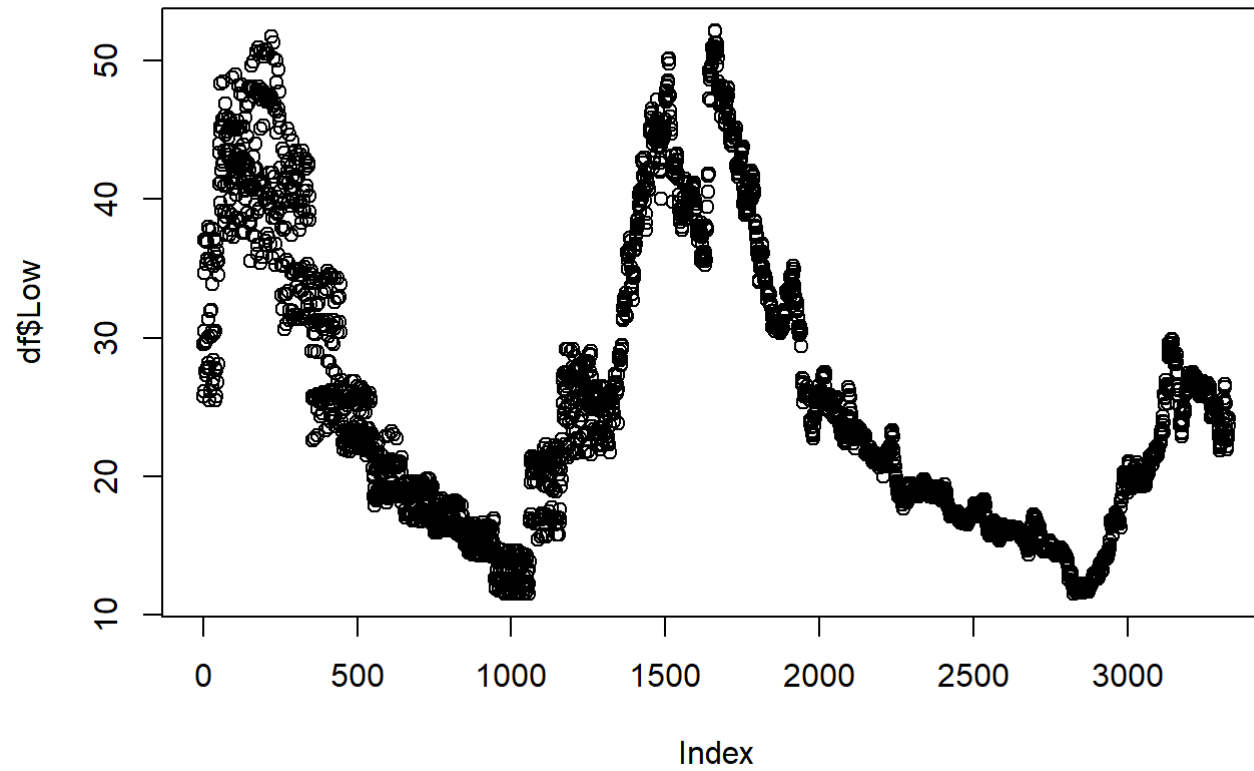


```
plot(df$High)
```



```
plot(df$Low)
```





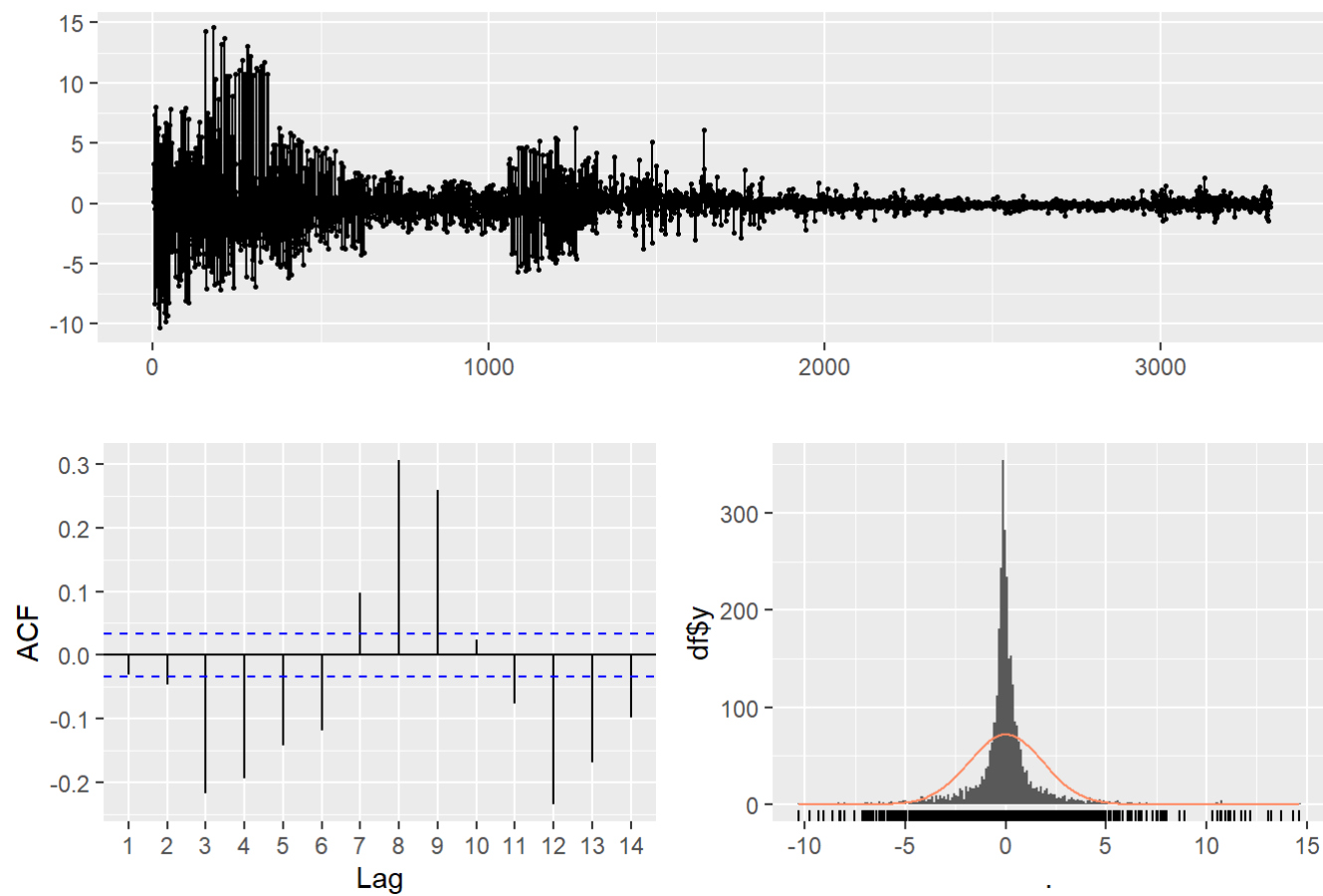
```
model.arima = auto.arima(df$Price , max.order = c(3 , 1 ,3) , stationary = TRUE , trace = T , ic = 'aicc')
```

```
##
## Fitting models using approximations to speed things up...
##
## ARIMA(2,0,2) with non-zero mean : Inf
## ARIMA(0,0,0) with non-zero mean : 24951.44
## ARIMA(1,0,0) with non-zero mean : 13689.71
## ARIMA(0,0,1) with non-zero mean : 21132.43
## ARIMA(0,0,0) with zero mean      : 31563.16
## ARIMA(2,0,0) with non-zero mean : 13661.12
## ARIMA(3,0,0) with non-zero mean : 13561.61
## ARIMA(4,0,0) with non-zero mean : Inf
## ARIMA(3,0,1) with non-zero mean : Inf
## ARIMA(2,0,1) with non-zero mean : Inf
## ARIMA(4,0,1) with non-zero mean : Inf
## ARIMA(3,0,0) with zero mean     : Inf
##
## Now re-fitting the best model(s) without approximations...
##
## ARIMA(3,0,0) with non-zero mean : 13565.58
##
## Best model: ARIMA(3,0,0) with non-zero mean
```

```
model.arima
```

```
## Series: df$Price
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3      mean
##          0.8757 -0.0626  0.1738  25.8052
## s.e.  0.0171   0.0228  0.0171   2.3752
##
## sigma^2 = 3.444: log likelihood = -6777.78
## AIC=13565.56  AICc=13565.58  BIC=13596.11
```

```
model.arima$residuals %>% ggtsdisplay(plot.type = 'hist' , lag.max = 14)
```

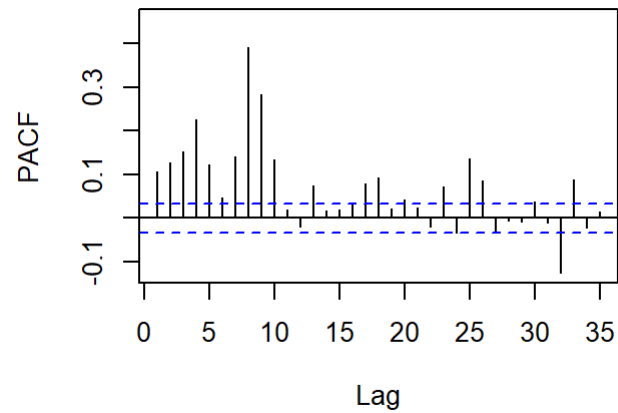
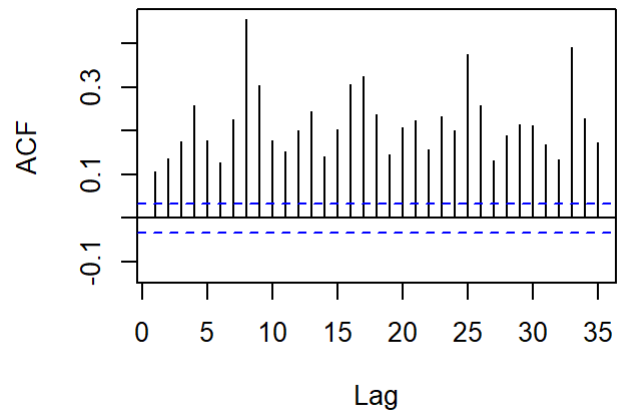
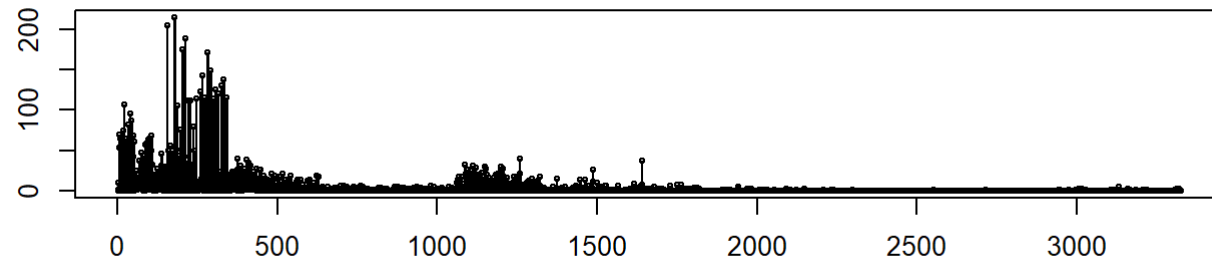


```
ar.res = model.arima$residuals  
Box.test(model.arima$residuals , lag = 14 , fitdf = 2 , type = 'Ljung-Box')
```

```
##  
## Box-Ljung test  
##  
## data: model.arima$residuals  
## X-squared = 1306.2, df = 12, p-value < 2.2e-16
```

```
tsdisplay(ar.res^2 , main = 'Squared Residuals')
```

### Squared Residuals



```
model.spec = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(1, 1)),
                        mean.model = list(armaOrder = c(0, 0)))
model.fit = ugarchfit(spec = model.spec, data = ar.res, solver = 'solnp')

options(scipen = 999)
model.fit@fit$matcoef
```

```
##           Estimate  Std. Error  t value    Pr(>|t|)
## mu      -0.0928408654 0.0070338083 -13.199232 0.0000000000
## omega    0.0005982118 0.0001542643  3.877837 0.0001053892
## alpha1   0.0568117487 0.0052891690 10.741148 0.0000000000
## beta1    0.9421882507 0.0049396698 190.739118 0.0000000000
```

```
jarque.bera.test(ar.res)
```

```
##
## Jarque Bera Test
##
## data:  ar.res
## X-squared = 25018, df = 2, p-value < 0.0000000000000022
```

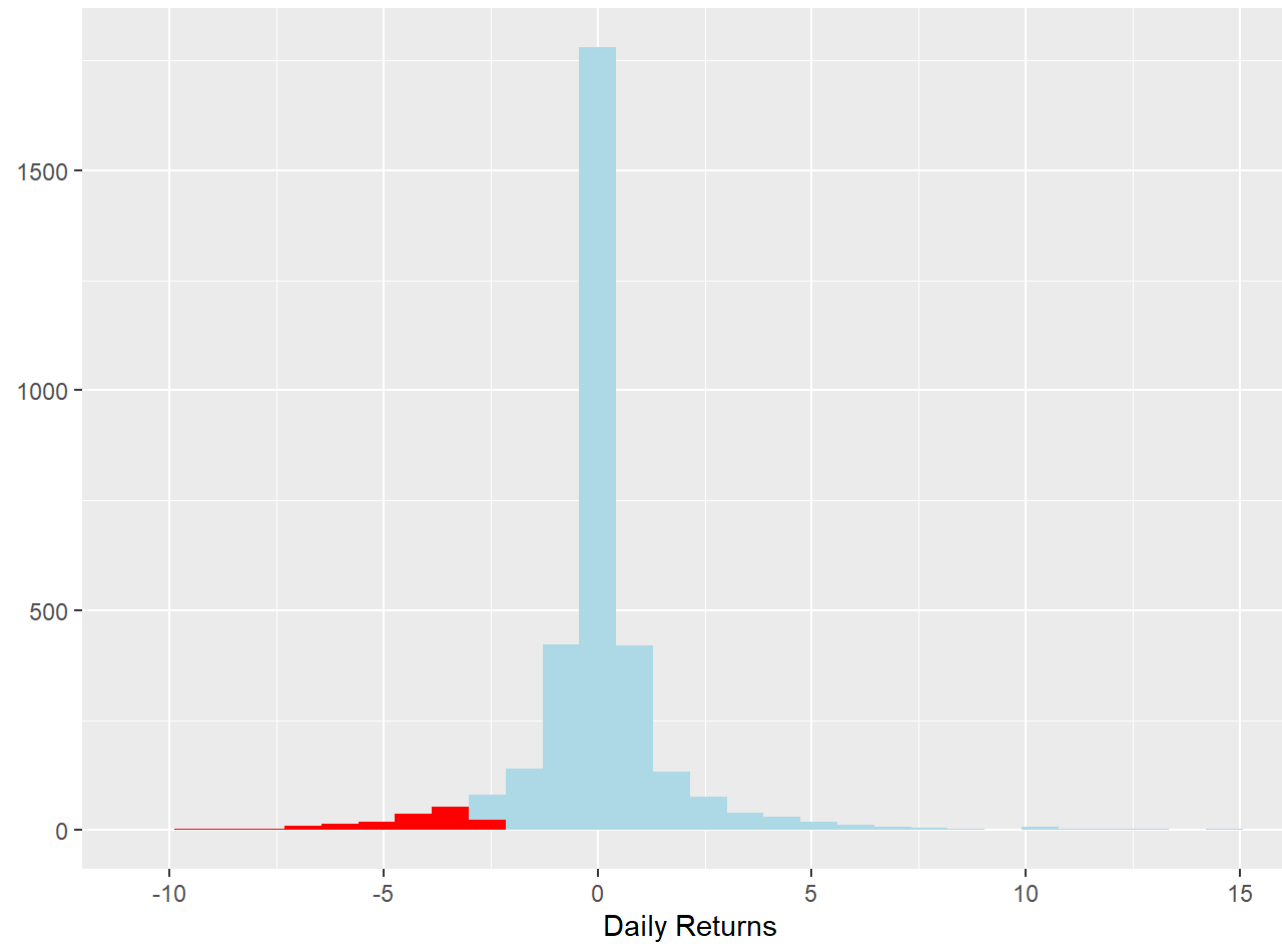
```
quantile(ar.res, 0.05)
```

```
##           5%
## -2.657457
```

```
qplot(ar.res, geom = 'histogram') + geom_histogram(fill = 'lightblue' , bins = 30) +
  geom_histogram(aes(ar.res[ar.res < quantile(ar.res , 0.05)]), fill = 'red' , bins = 30) +
  labs(x = 'Daily Returns')
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting  
## to continuous.  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
p2_1 = qplot(ar.res , geom = 'density') + geom_density(fill = 'blue' , alpha = 0.4) +  
  geom_density(aes(rnorm(200000 , 0 , sd(ar.res))) , fill = 'red' , alpha = 0.25) +  
  labs(x = 'Daily Returns')  
  
p2_2 = qplot(p2_1 = qplot(ar.res , geom = 'density') + geom_density(fill = 'blue' , alpha = 0.4) +  
  geom_density(aes(rnorm(200000 , 0 , sd(ar.res))) , fill = 'red' , alpha = 0.25) +  
  labs(x = ''), geom = 'density') + geom_density(fill = 'blue' , alpha = 0.4) +  
  geom_density(aes(rnorm(200000 , 0 , sd(ar.res))) , fill = 'red' , alpha = 0.25) +  
  coord_cartesian(xlim = c(-0.07 , -0.02) , ylim = c(0 , 10)) +  
  geom_vline(xintercept = c(qnorm(p = c(0.01 , 0.05) , mean = mean(ar.res) , sd = sd(ar.res))) ,  
    color = c('darkgreen' , 'green') , size = 1) + labs(x = 'Daily Returns')
```

```
## Warning in geom_density(p2_1 = structure(list(data = structure(list(), names =  
## character(0), class = "data.frame", row.names = integer(0)), : Ignoring unknown  
## parameters: `p2_1`
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## i Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```

```
grid.arrange(p2_1 , p2_2 , ncol = 1)
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting  
## to continuous.
```



```
fitdist(distribution = 'std' , x = ar.res)$pars
```

```
##      mu      sigma      shape  
## -0.05603014  8.09222404  2.01000003
```



```
cat("For a = 0.05 the quantile value of normal distribution is: " ,  
    qnorm(p = 0.05) , "\n" ,  
    "For a = 0.05 the quantile value of t-distribution is: " ,  
    qdist(distribution = 'std' , shape = 2.0100001 , p = 0.05) , "\n" , "\n" ,  
    'For a = 0.01 the quantile value of normal distribution is: ' ,  
    qnorm(p = 0.01) , "\n" ,  
    "For a = 0.01 the quantile value of t-distribution is: " ,  
    qdist(distribution = 'std' , shape = 2.0100001 , p = 0.01) , sep = "")
```

```
## For a = 0.05 the quantile value of normal distribution is: -1.644854  
## For a = 0.05 the quantile value of t-distribution is: -0.2052638  
##  
## For a = 0.01 the quantile value of normal distribution is: -2.326348  
## For a = 0.01 the quantile value of t-distribution is: -0.487933
```