Project1\_S03

Lwin\_Shwe

2023-06-16

library(fpp2)  
library(dplyr)  
library(imputeTS)  
library(readxl)  
library(tsoutliers)  
library(forecast)

library(readxl)  
DATA624\_Project1\_Data\_S03 <- read\_excel("~/DATA624\_Project1\_Data\_S03.xlsx")  
View(DATA624\_Project1\_Data\_S03)

data\_S03 <- subset(DATA624\_Project1\_Data\_S03, category == 'S03', select = c(SeriesInd, Var05, Var07)) %>%  
 mutate(date = as.Date(SeriesInd, origin = "1900-01-01"))  
summary(data\_S03)

## SeriesInd Var05 Var07 date   
## Min. :40669 Min. :2.700e+01 Min. :2.700e+01 Min. :2011-05-08   
## 1st Qu.:41304 1st Qu.:5.500e+01 1st Qu.:5.500e+01 1st Qu.:2013-01-31   
## Median :41946 Median :7.900e+01 Median :7.800e+01 Median :2014-11-05   
## Mean :41945 Mean :7.964e+08 Mean :7.964e+08 Mean :2014-11-03   
## 3rd Qu.:42586 3rd Qu.:1.070e+02 3rd Qu.:1.070e+02 3rd Qu.:2016-08-06   
## Max. :43221 Max. :1.000e+10 Max. :1.000e+10 Max. :2018-05-03   
## NA's :4 NA's :4

str(data\_S03)

## tibble [1,762 × 4] (S3: tbl\_df/tbl/data.frame)  
## $ SeriesInd: num [1:1762] 40669 40670 40671 40672 40673 ...  
## $ Var05 : num [1:1762] 30.5 30.7 30.6 30.2 30 ...  
## $ Var07 : num [1:1762] 30.6 30.6 30.1 30.1 30.3 ...  
## $ date : Date[1:1762], format: "2011-05-08" "2011-05-09" ...

data\_S03 <- filter(data\_S03, SeriesInd <= 43221)  
summary(data\_S03)

## SeriesInd Var05 Var07 date   
## Min. :40669 Min. :2.700e+01 Min. :2.700e+01 Min. :2011-05-08   
## 1st Qu.:41304 1st Qu.:5.500e+01 1st Qu.:5.500e+01 1st Qu.:2013-01-31   
## Median :41946 Median :7.900e+01 Median :7.800e+01 Median :2014-11-05   
## Mean :41945 Mean :7.964e+08 Mean :7.964e+08 Mean :2014-11-03   
## 3rd Qu.:42586 3rd Qu.:1.070e+02 3rd Qu.:1.070e+02 3rd Qu.:2016-08-06   
## Max. :43221 Max. :1.000e+10 Max. :1.000e+10 Max. :2018-05-03   
## NA's :4 NA's :4

str(data\_S03)

## tibble [1,762 × 4] (S3: tbl\_df/tbl/data.frame)  
## $ SeriesInd: num [1:1762] 40669 40670 40671 40672 40673 ...  
## $ Var05 : num [1:1762] 30.5 30.7 30.6 30.2 30 ...  
## $ Var07 : num [1:1762] 30.6 30.6 30.1 30.1 30.3 ...  
## $ date : Date[1:1762], format: "2011-05-08" "2011-05-09" ...

data\_S03\_v5 <- data\_S03 %>% select(Var05)  
data\_S03\_v5 <- data\_S03\_v5[1:1622,]  
summary(data\_S03\_v5)

## Var05   
## Min. : 27.48   
## 1st Qu.: 53.30   
## Median : 75.59   
## Mean : 76.90   
## 3rd Qu.: 98.55   
## Max. :134.46   
## NA's :4

str(data\_S03\_v5)

## tibble [1,622 × 1] (S3: tbl\_df/tbl/data.frame)  
## $ Var05: num [1:1622] 30.5 30.7 30.6 30.2 30 ...

data\_S03\_v7 <- data\_S03 %>% select(Var07)  
data\_S03\_v7 <- data\_S03\_v7[1:1622,]  
summary(data\_S03\_v7)

## Var07   
## Min. : 27.44   
## 1st Qu.: 53.46   
## Median : 75.71   
## Mean : 76.87   
## 3rd Qu.: 98.61   
## Max. :133.00   
## NA's :4

str(data\_S03\_v7)

## tibble [1,622 × 1] (S3: tbl\_df/tbl/data.frame)  
## $ Var07: num [1:1622] 30.6 30.6 30.1 30.1 30.3 ...

data\_S03\_v5 <- na\_interpolation(data\_S03\_v5)  
summary(data\_S03\_v5)

## Var05   
## Min. : 27.48   
## 1st Qu.: 53.34   
## Median : 75.66   
## Mean : 76.95   
## 3rd Qu.: 98.53   
## Max. :134.46

str(data\_S03\_v5)

## tibble [1,622 × 1] (S3: tbl\_df/tbl/data.frame)  
## $ Var05: num [1:1622] 30.5 30.7 30.6 30.2 30 ...

data\_S03\_v7 <- na\_interpolation(data\_S03\_v7)  
summary(data\_S03\_v7)

## Var07   
## Min. : 27.44   
## 1st Qu.: 53.53   
## Median : 75.76   
## Mean : 76.91   
## 3rd Qu.: 98.51   
## Max. :133.00

str(data\_S03\_v7)

## tibble [1,622 × 1] (S3: tbl\_df/tbl/data.frame)  
## $ Var07: num [1:1622] 30.6 30.6 30.1 30.1 30.3 ...

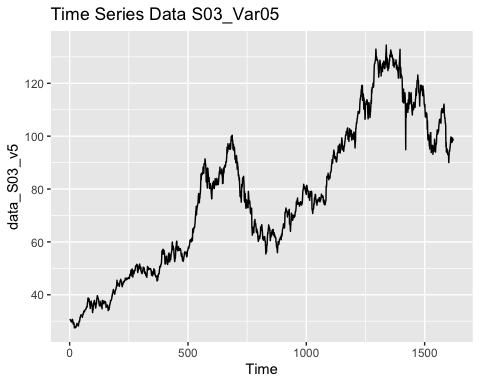
data\_S03\_v5 <- ts(data\_S03\_v5)  
str(data\_S03\_v5)

## Time-Series [1:1622, 1] from 1 to 1622: 30.5 30.7 30.6 30.2 30 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "Var05"

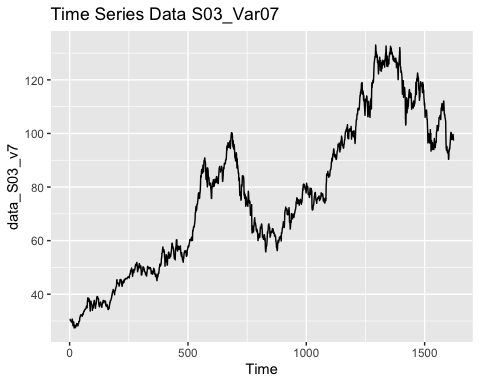
data\_S03\_v7 <- ts(data\_S03\_v7)  
str(data\_S03\_v7)

## Time-Series [1:1622, 1] from 1 to 1622: 30.6 30.6 30.1 30.1 30.3 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "Var07"

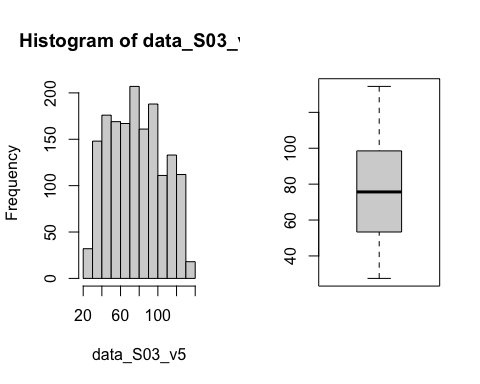
autoplot(data\_S03\_v5) + ggtitle("Time Series Data S03\_Var05")



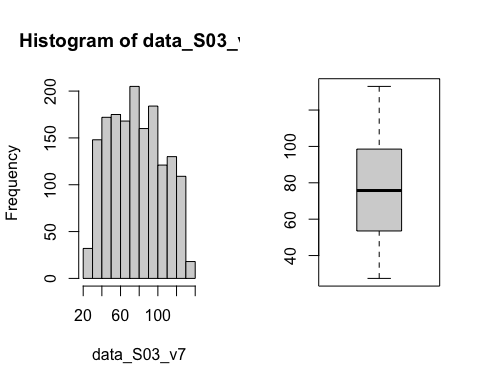
autoplot(data\_S03\_v7) + ggtitle("Time Series Data S03\_Var07")



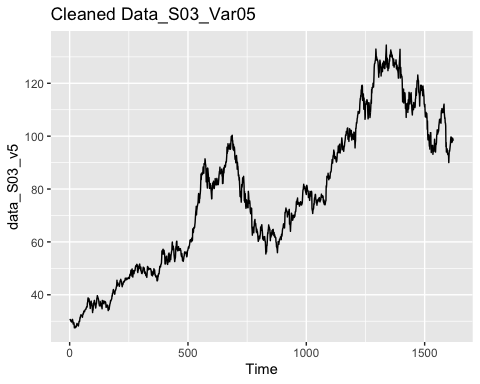
par(mfrow= c(1,2))  
hist(data\_S03\_v5)  
boxplot(data\_S03\_v5)



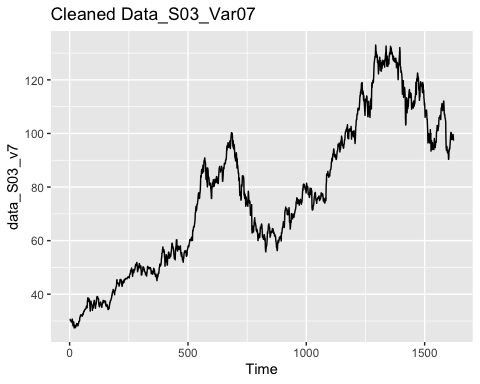
hist(data\_S03\_v7)  
boxplot(data\_S03\_v7)



data\_S03\_v5\_out <- tsoutliers(data\_S03\_v5)  
data\_S03\_v7\_out <- tsoutliers(data\_S03\_v7)  
data\_S03\_v5 <- tsclean(data\_S03\_v5)  
data\_S03\_v7 <-tsclean(data\_S03\_v7)  
autoplot(data\_S03\_v5) + ggtitle("Cleaned Data\_S03\_Var05")



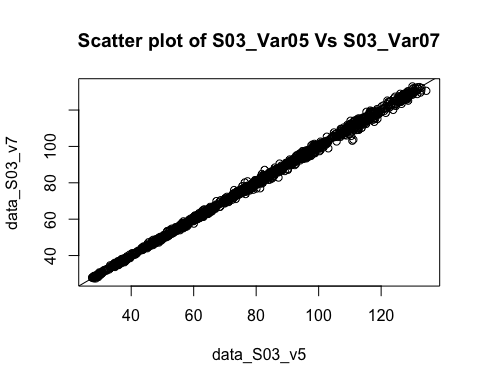
autoplot(data\_S03\_v7) + ggtitle("Cleaned Data\_S03\_Var07")



# correlation test & Extract correlation coefficient and p-value  
correlation\_test <- cor.test(data\_S03\_v5, data\_S03\_v7)  
cor\_coefficient <- correlation\_test$estimate  
p\_value <- correlation\_test$p.value  
print(correlation\_test)

##   
## Pearson's product-moment correlation  
##   
## data: data\_S03\_v5 and data\_S03\_v7  
## t = 1027.5, df = 1620, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9991552 0.9993047  
## sample estimates:  
## cor   
## 0.9992336

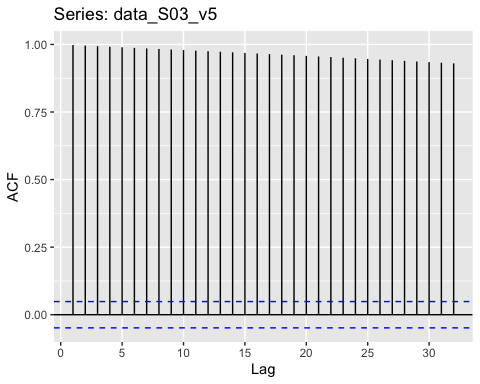
lmodel <- lm(data\_S03\_v5 ~ data\_S03\_v7)  
plot(data\_S03\_v5, data\_S03\_v7)   
abline(lmodel)  
title(main = "Scatter plot of S03\_Var05 Vs S03\_Var07")



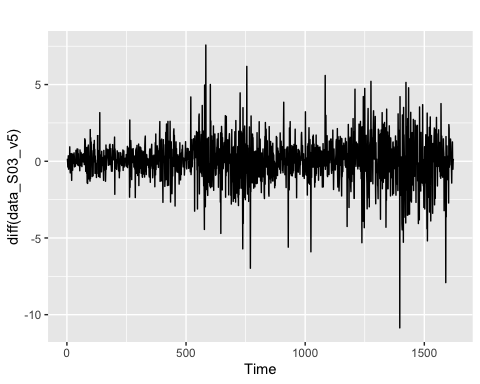
correlation <- cor(data\_S03\_v5, data\_S03\_v7)  
squared\_correlation <- correlation^2  
print(squared\_correlation)

## Var07  
## Var05 0.9984678

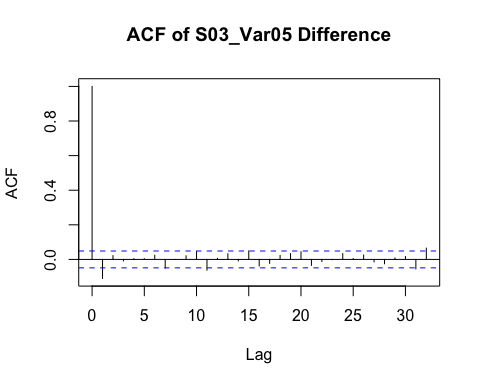
# Plot the ACF of the differences of S03\_Var05  
ggAcf(data\_S03\_v5)



autoplot(diff(data\_S03\_v5))

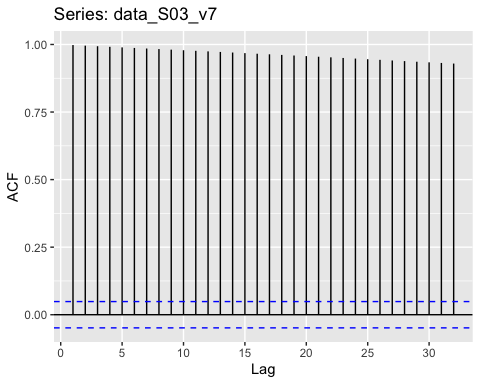


acf(diff(data\_S03\_v5), main = "ACF of S03\_Var05 Difference")

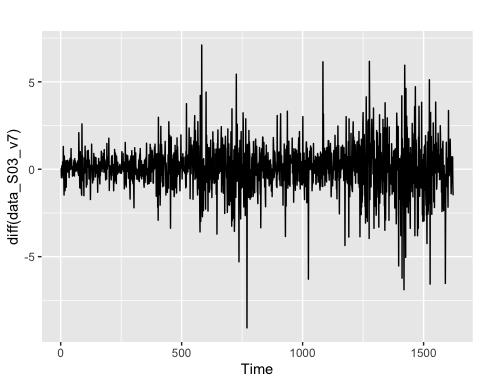


# There is no seasonality due to the presence of one order autocorrelation.

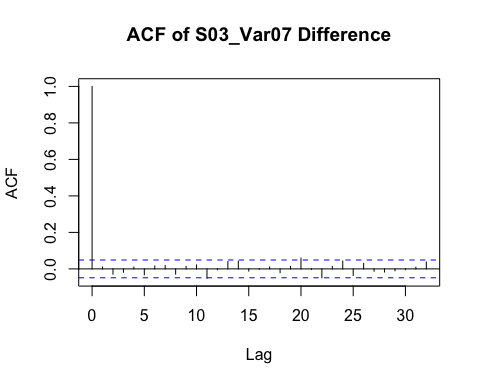
# Plot the ACF of the differences of S03\_Var07  
ggAcf(data\_S03\_v7)



autoplot(diff(data\_S03\_v7))



acf(diff(data\_S03\_v7), main = "ACF of S03\_Var07 Difference")



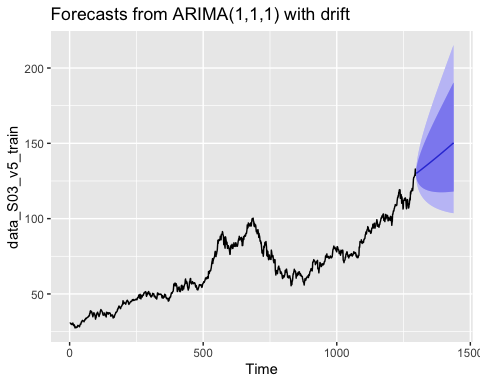
# There is no seasonality due to the presence of one order autocorrelation.

data\_S03\_v5\_train <- window(data\_S03\_v5, end = as.integer(length(data\_S03\_v5) \* 0.80))  
data\_S03\_v7\_train <- window(data\_S03\_v7, end = as.integer(length(data\_S03\_v7) \* 0.85))  
  
# Find lambda value  
lambda5 <- BoxCox.lambda(data\_S03\_v5)  
lambda7 <- BoxCox.lambda(data\_S03\_v7)  
# Apply the selected model to training Set  
f\_horizon <- length(data\_S03\_v5) - as.integer(length(data\_S03\_v5) \* 0.914)  
f\_horizon

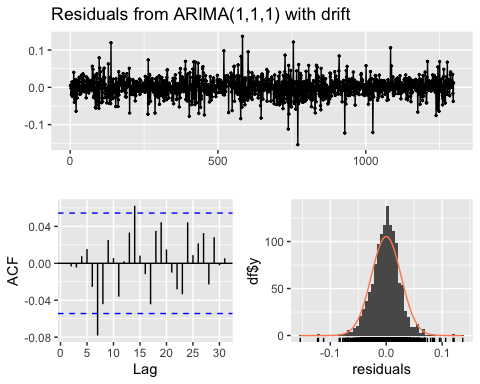
## [1] 140

# There are 140 periods in forecasting horizon.

# Fit the ARIMA model  
data\_S03\_v5\_farima\_fit <- auto.arima(data\_S03\_v5\_train, lambda = lambda5, stepwise = FALSE)  
# Forecast time series  
fresult\_arima\_V05 <- forecast(data\_S03\_v5\_farima\_fit, h = f\_horizon)  
# visualize ARIMA model of foresting   
autoplot(fresult\_arima\_V05)+autolayer(fresult\_arima\_V05, alpha = 0.65)

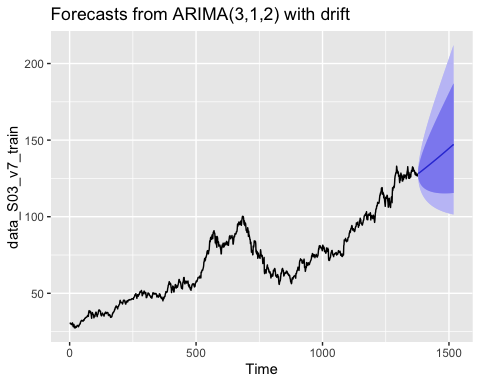


#Check Var05 residuals if the model is valid  
checkresiduals(fresult\_arima\_V05)

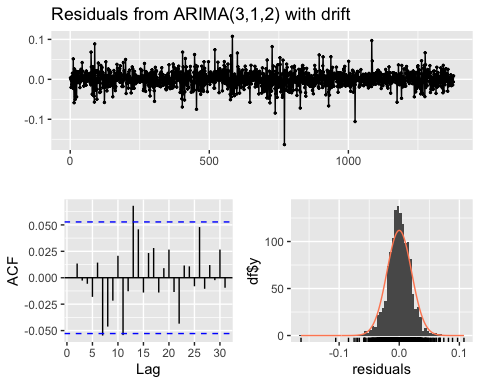


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1) with drift  
## Q\* = 12.718, df = 8, p-value = 0.1219  
##   
## Model df: 2. Total lags used: 10

#with p-value greater than 0.05, there is convincing evidence that residuals for Var05 are white noise. On ACF, the residuals are uncorrelated. The histogram shows that the residuals are normal distributed.  
  
data\_S03\_v7\_farima\_fit <- auto.arima(data\_S03\_v7\_train, lambda = lambda7, stepwise = FALSE)  
fresult\_arima\_V07 <- forecast(data\_S03\_v7\_farima\_fit, h = f\_horizon)  
autoplot(fresult\_arima\_V07)+autolayer(fresult\_arima\_V07, alpha = 0.65)

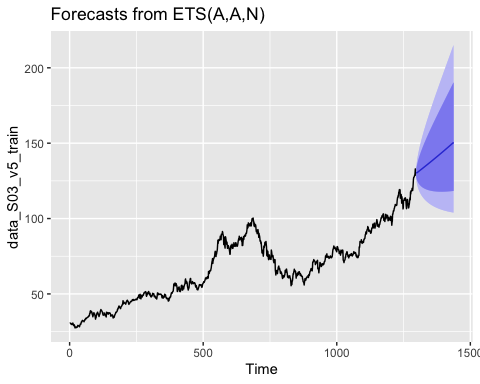


checkresiduals(fresult\_arima\_V07)

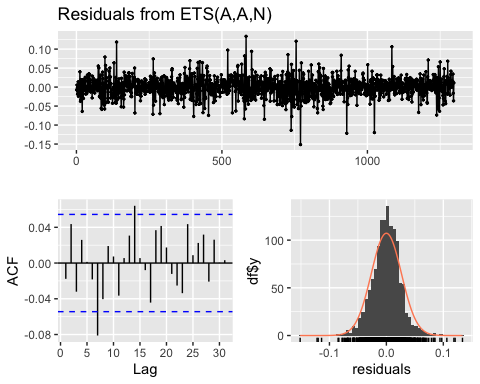


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,2) with drift  
## Q\* = 9.4061, df = 5, p-value = 0.09392  
##   
## Model df: 5. Total lags used: 10

# Fit the ETS model  
data\_S03\_v5\_fets\_fit <- ets(data\_S03\_v5\_train, lambda = lambda5)  
# Forecast the result  
fresult\_fets\_V05<- forecast(data\_S03\_v5\_fets\_fit, h = f\_horizon)  
# visualize ETS model of foresting   
autoplot(fresult\_fets\_V05)+autolayer(fresult\_fets\_V05, alpha = 0.65)

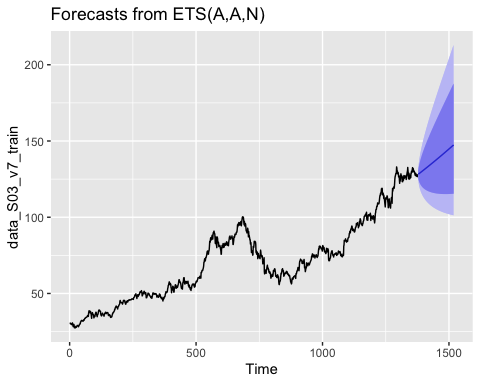


#Check Var05 residuals if the model is valid  
checkresiduals(fresult\_fets\_V05)

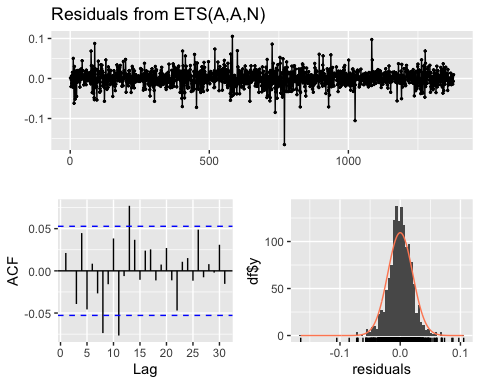


##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,A,N)  
## Q\* = 16.813, df = 10, p-value = 0.07861  
##   
## Model df: 0. Total lags used: 10

#The p values for the ETS models residuals are less than 0.05.The residuals are not white noise. They ETS models have prediction interval to wide. The ETS models have the best accuracy with the test set  
  
# Fit the ETS model for S03\_Var07  
data\_S03\_v7\_fets\_fit <- ets(data\_S03\_v7\_train, lambda = lambda7)  
# Forecast the result of S03\_Var07  
fresult\_fets\_V07<- forecast(data\_S03\_v7\_fets\_fit, h = f\_horizon)  
# visualize ETS model of foresting for variables S03\_Var07  
autoplot(fresult\_fets\_V07)+autolayer(fresult\_fets\_V07, alpha = 0.65)

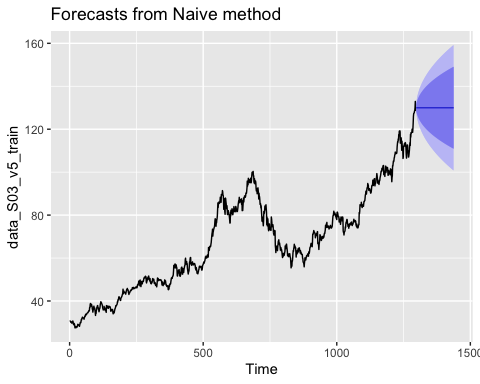


#Check Var07 residuals if the model is valid  
checkresiduals(fresult\_fets\_V07)

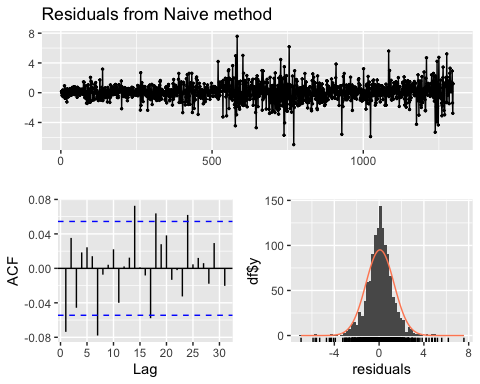


##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,A,N)  
## Q\* = 19.454, df = 10, p-value = 0.03486  
##   
## Model df: 0. Total lags used: 10

# Forecast the result   
data\_S03\_v5\_naive\_fit <- naive(data\_S03\_v5\_train, h = f\_horizon)  
# visualize Naive model of foresting   
autoplot(data\_S03\_v5\_naive\_fit)+autolayer(data\_S03\_v5\_naive\_fit, alpha = 0.65)

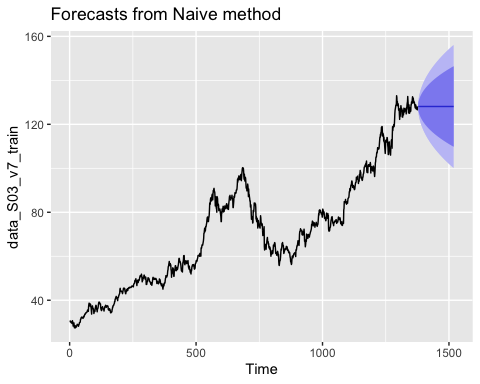


#Check Var05 residuals if the model is valid  
checkresiduals(data\_S03\_v5\_naive\_fit)

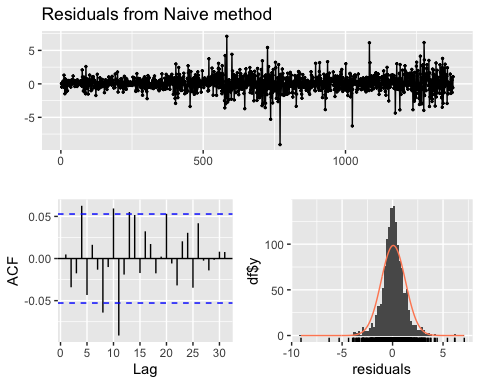


##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 21.548, df = 10, p-value = 0.01758  
##   
## Model df: 0. Total lags used: 10

data\_S03\_v7\_naive\_fit <- naive(data\_S03\_v7\_train, h = f\_horizon)  
autoplot(data\_S03\_v7\_naive\_fit)+autolayer(data\_S03\_v7\_naive\_fit, alpha = 0.65)



checkresiduals(data\_S03\_v7\_naive\_fit)



##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 21.442, df = 10, p-value = 0.01821  
##   
## Model df: 0. Total lags used: 10

# ARIMA model  
DS03\_Var05\_farima\_ac <- accuracy(fresult\_arima\_V05, data\_S03\_v5)["Test set", "MAPE"]  
  
# ETS model  
DS03\_Var05\_fets\_ac <- accuracy(fresult\_fets\_V05, data\_S03\_v5)["Test set", "MAPE"]  
  
# Naive model  
DS03\_Var05\_fnaive\_ac <- accuracy(data\_S03\_v5\_naive\_fit, data\_S03\_v5)["Test set", "MAPE"]

# ARIMA model  
DS03\_Var07\_farima\_ac <- accuracy(fresult\_arima\_V07, data\_S03\_v7)["Test set", "MAPE"]  
  
# ETS model  
DS03\_Var07\_fets\_ac <- accuracy(fresult\_fets\_V07, data\_S03\_v7)["Test set", "MAPE"]  
  
# Naive model  
DS03\_Var07\_fnaive\_ac <- accuracy(data\_S03\_v7\_naive\_fit, data\_S03\_v7)["Test set", "MAPE"]

DS03\_Var05\_MAPE <- c(DS03\_Var05\_fnaive\_ac, DS03\_Var05\_farima\_ac, DS03\_Var05\_fets\_ac)  
DS03\_Var07\_MAPE <- c(DS03\_Var07\_fnaive\_ac, DS03\_Var07\_farima\_ac, DS03\_Var07\_fets\_ac)  
  
S03\_MAPE <- matrix(rbind(DS03\_Var05\_MAPE, DS03\_Var07\_MAPE), nrow = 2, ncol = 3)  
rownames(S03\_MAPE) <- c("S03\_Var05", "S03\_Var07")  
colnames(S03\_MAPE) <- c("Naive", "ARIMA", "ETS")  
data.frame(S03\_MAPE)

## Naive ARIMA ETS  
## S03\_Var05 5.331602 13.33630 13.42142  
## S03\_Var07 11.746943 20.03622 20.13147

### According to the results of MAPE (Mean Absolute Percentage Error), the Naive model has the best accuracy with the least error compared to ETS and ARIMA methods for the variables S03\_Var05 and S03\_Var07.