S06\_Var05\_Var07

Lwin Shwe

2023-06-18

You can embed an R code chunk like this:

library(fpp2)  
library(dplyr)  
library(imputeTS)  
library(readxl)  
library(forecast)  
library(ggplot2)  
library(ggfortify)  
library(caret)

Project1\_DataS06 <- read\_excel("Project1\_DataS06.xlsx", skip = 1)  
View(Project1\_DataS06)

data\_S06 <- subset(Project1\_DataS06, category == 'S06', select = c(SeriesInd, Var05, Var07)) %>%  
 mutate(date = as.Date(SeriesInd, origin = "1900-01-01"))  
data\_S06 <- filter(data\_S06, SeriesInd <= 43221)  
summary(data\_S06)

## SeriesInd Var05 Var07 date   
## Min. :40669 Min. :2.300e+01 Min. :2.300e+01 Min. :2011-05-08   
## 1st Qu.:41304 1st Qu.:3.100e+01 1st Qu.:3.100e+01 1st Qu.:2013-01-31   
## Median :41946 Median :4.100e+01 Median :4.100e+01 Median :2014-11-05   
## Mean :41945 Mean :7.968e+08 Mean :7.968e+08 Mean :2014-11-03   
## 3rd Qu.:42586 3rd Qu.:5.200e+01 3rd Qu.:5.200e+01 3rd Qu.:2016-08-06   
## Max. :43221 Max. :1.000e+10 Max. :1.000e+10 Max. :2018-05-03   
## NA's :5 NA's :5

str(data\_S06)

## tibble [1,762 × 4] (S3: tbl\_df/tbl/data.frame)  
## $ SeriesInd: num [1:1762] 40669 40670 40671 40672 40673 ...  
## $ Var05 : num [1:1762] 27 27.3 28 28.1 28.9 ...  
## $ Var07 : num [1:1762] 27.3 28.1 28.1 29.1 28.9 ...  
## $ date : Date[1:1762], format: "2011-05-08" "2011-05-09" ...

data\_S06\_v5 <- data\_S06 %>% select(Var05)  
data\_S06\_v5 <- data\_S06\_v5[1:1622,]  
summary(data\_S06\_v5)

## Var05   
## Min. : 22.91   
## 1st Qu.: 30.32   
## Median : 36.87   
## Mean : 39.85   
## 3rd Qu.: 50.47   
## Max. :195.00   
## NA's :5

str(data\_S06\_v5)

## tibble [1,622 × 1] (S3: tbl\_df/tbl/data.frame)  
## $ Var05: num [1:1622] 27 27.3 28 28.1 28.9 ...

data\_S06\_v7 <- data\_S06 %>% select(Var07)  
data\_S06\_v7 <- data\_S06\_v7[1:1622,]  
summary(data\_S06\_v7)

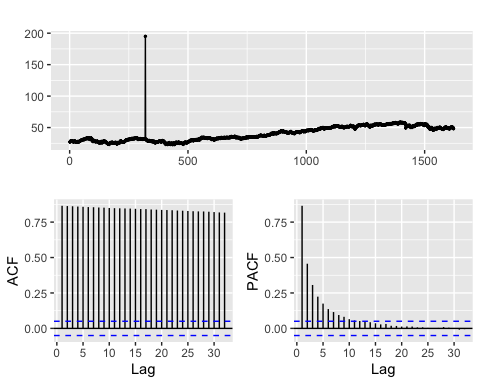
## Var07   
## Min. : 22.88   
## 1st Qu.: 30.26   
## Median : 36.97   
## Mean : 39.85   
## 3rd Qu.: 50.45   
## Max. :189.72   
## NA's :5

str(data\_S06\_v7)

## tibble [1,622 × 1] (S3: tbl\_df/tbl/data.frame)  
## $ Var07: num [1:1622] 27.3 28.1 28.1 29.1 28.9 ...

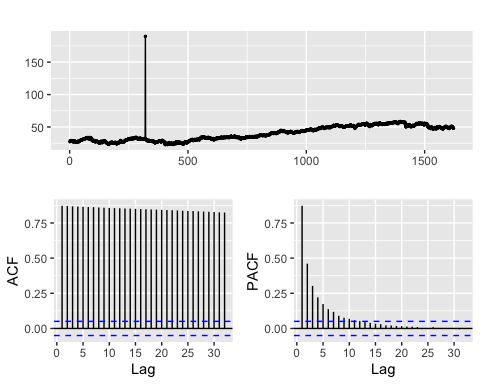
data\_S06\_Var05 <- ts(data\_S06\_v5)  
ggtsdisplay(data\_S06\_Var05)

## Warning: Removed 5 rows containing missing values (`geom\_point()`).

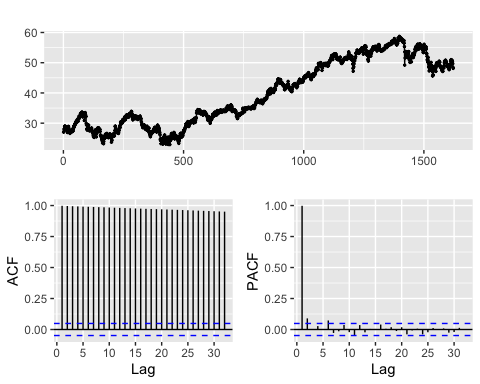


data\_S06\_Var07 <- ts(data\_S06\_v7)  
ggtsdisplay(data\_S06\_Var07)

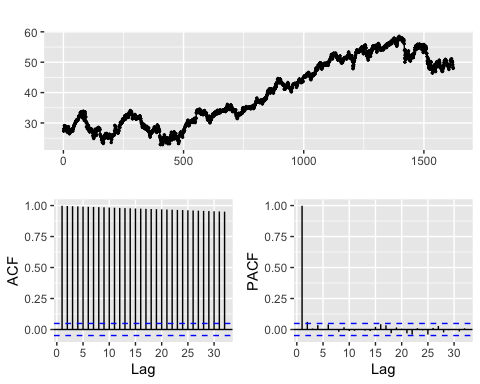
## Warning: Removed 5 rows containing missing values (`geom\_point()`).



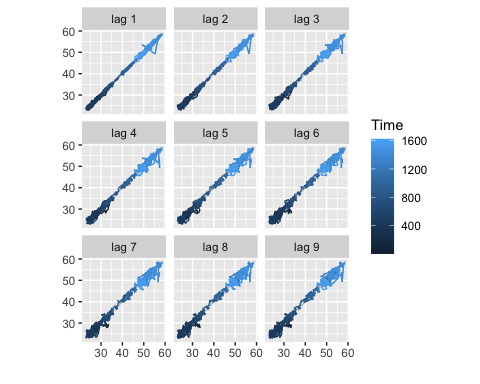
data\_S06\_Var05 <- data\_S06\_Var05 |>  
 tsclean()   
ggtsdisplay(data\_S06\_Var05)



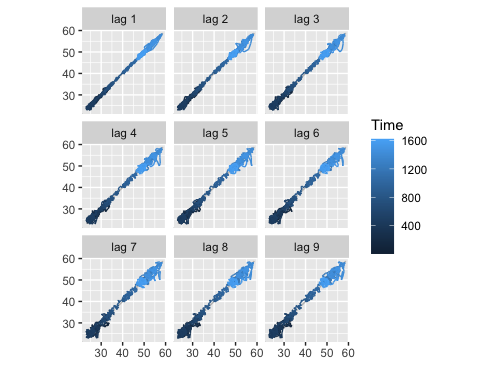
data\_S06\_Var07 <- data\_S06\_Var07 |>  
 tsclean()   
ggtsdisplay(data\_S06\_Var07)

 ### The data look getting better, appears to be some seaonality, upward trends via time. No significant variabilities but lags are so obvious over time in the ACF and cutoff after lag2 in PACE.

gglagplot(data\_S06\_Var05)



gglagplot(data\_S06\_Var07)



train\_index <- createDataPartition(data\_S06\_Var05, times = 1, p = 0.912, list = FALSE)  
train <- data\_S06\_Var05[train\_index]  
test <- data\_S06\_Var05[-train\_index]  
horizon <- length(test)  
print(dim(train))

## NULL

print(dim(test))

## NULL

print(horizon)

## [1] 140

train\_index <- createDataPartition(data\_S06\_Var07, times = 1, p = 0.912, list = FALSE)  
train <- data\_S06\_Var07[train\_index]  
test <- data\_S06\_Var07[-train\_index]  
horizon <- length(test)  
print(dim(train))

## NULL

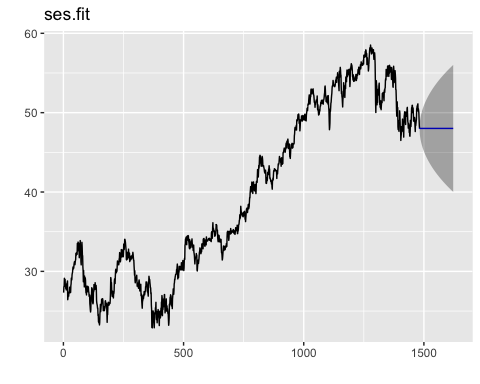
print(dim(test))

## NULL

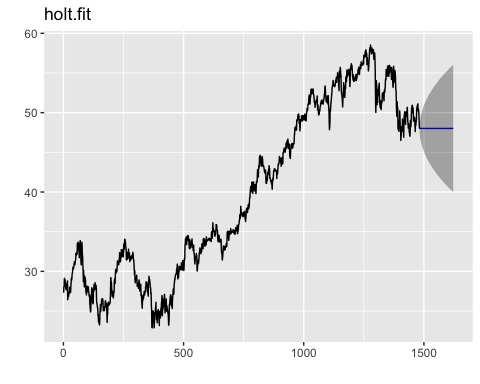
print(horizon)

## [1] 140

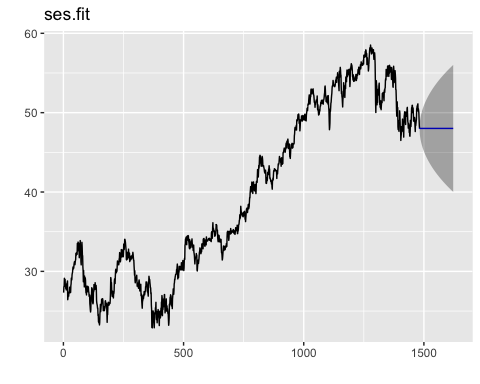
# Simple Exponential Smoothing Method and # Holt's Method  
set.seed(123)  
ses.fit <- ses(train, h = horizon)  
ses\_accuracy\_Var05 <- c(data\_S06\_Var05 = accuracy(ses.fit, test)['Test set', 'MAPE'])  
holt.fit <- holt(train, damped=TRUE, h = horizon)  
holt\_accuracy\_Var05 <- c(data\_S06\_Var05 = accuracy(holt.fit, test)['Test set', 'MAPE'])  
  
autoplot(ses.fit) + ggtitle("ses.fit")



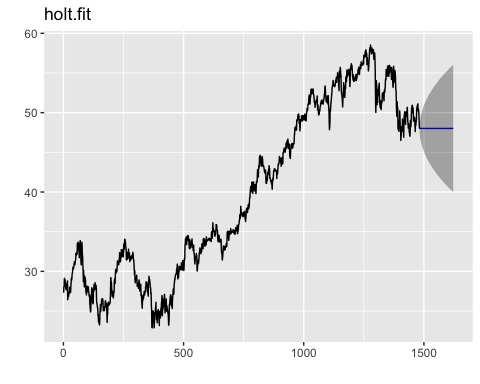
autoplot(holt.fit) + ggtitle("holt.fit")



ses.fit <- ses(train, h = horizon)  
ses\_accuracy\_Var07<- c(data\_S06\_Var07 = accuracy(ses.fit, test)['Test set', 'MAPE'])  
holt.fit <- holt(train, damped=TRUE, h = horizon)  
holt\_accuracy\_Var07 <- c(data\_S06\_Var07 = accuracy(holt.fit, test)['Test set', 'MAPE'])  
  
autoplot(ses.fit) + ggtitle("ses.fit")



autoplot(holt.fit) + ggtitle("holt.fit")



rbind(holt\_pval = Box.test(residuals(holt.fit))$p.value,  
 ses\_pval = Box.test(residuals(ses.fit))$p.value)

## [,1]  
## holt\_pval 0.9882207  
## ses\_pval 0.9884219

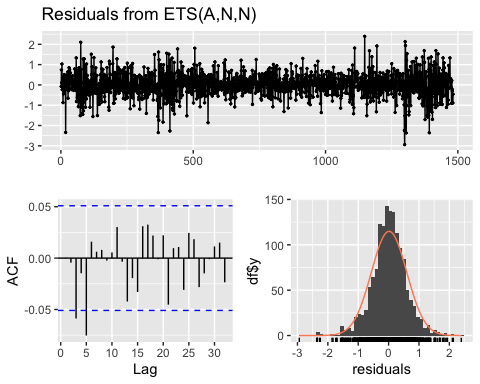
set.seed(123)  
ets.fit <- ets(train)  
ets.fit

## ETS(A,N,N)   
##   
## Call:  
## ets(y = train)   
##   
## Smoothing parameters:  
## alpha = 0.902   
##   
## Initial states:  
## l = 27.3948   
##   
## sigma: 0.5848  
##   
## AIC AICc BIC   
## 9233.986 9234.002 9249.889

ets.fc <- forecast(ets.fit, h = horizon)  
ets\_accuray\_Var05 <- c(data\_S06\_Var05 = accuracy(ets.fc, test)['Test set', 'MAPE'])  
ets\_accuray\_Var05

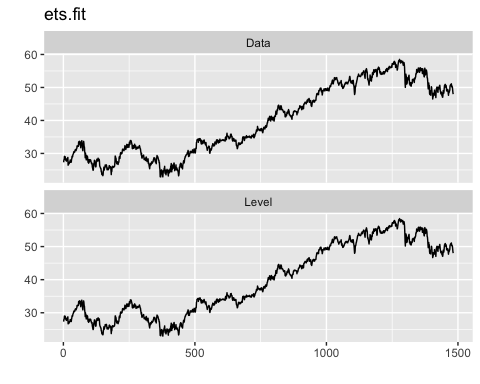
## data\_S06\_Var05   
## 35.46384

checkresiduals(ets.fc)

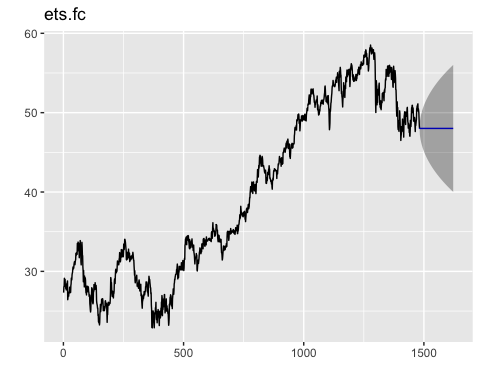


##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,N,N)  
## Q\* = 14.534, df = 10, p-value = 0.15  
##   
## Model df: 0. Total lags used: 10

autoplot(ets.fit) + ggtitle("ets.fit")



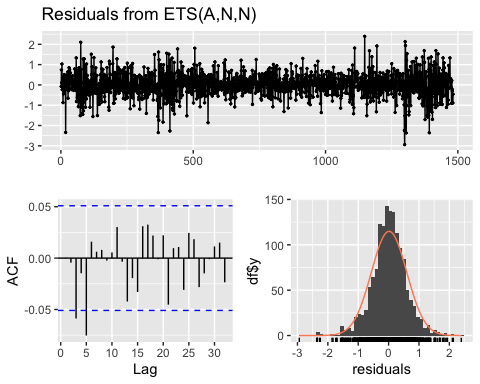
autoplot(ets.fc) + ggtitle("ets.fc")



ets.fc <- forecast(ets.fit, h = horizon)  
ets\_accuray\_Var07 <- c(data\_S06\_Var07 = accuracy(ets.fc, test)['Test set', 'MAPE'])  
ets\_accuray\_Var07

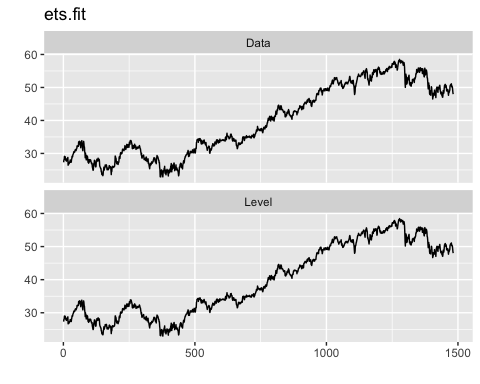
## data\_S06\_Var07   
## 35.46384

checkresiduals(ets.fc)

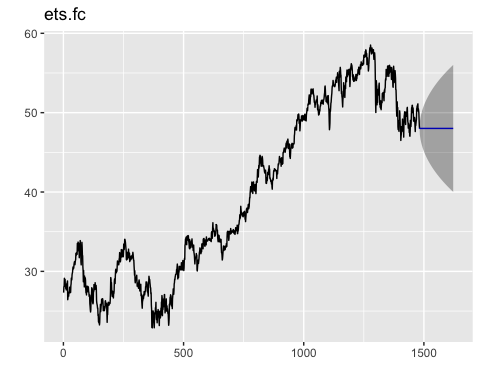


##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,N,N)  
## Q\* = 14.534, df = 10, p-value = 0.15  
##   
## Model df: 0. Total lags used: 10

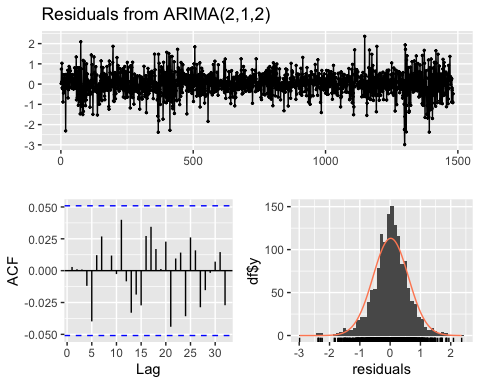
autoplot(ets.fit) + ggtitle("ets.fit")



autoplot(ets.fc) + ggtitle("ets.fc")

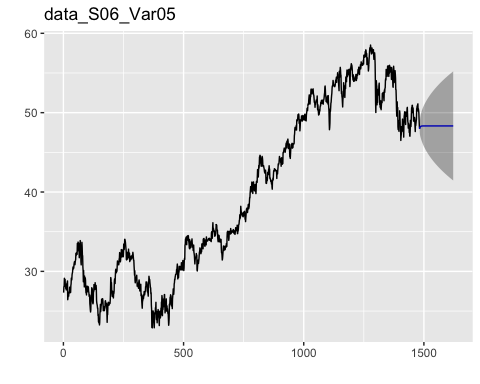


set.seed(123)  
train |>  
 auto.arima() |>  
 forecast(h = horizon) |>  
 checkresiduals()

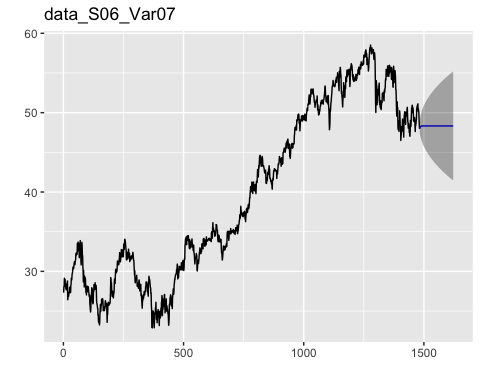


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2)  
## Q\* = 4.1021, df = 6, p-value = 0.6629  
##   
## Model df: 4. Total lags used: 10

arima.fc <- train |>  
 auto.arima() |>  
 forecast(h=horizon)  
arima\_accuray\_Var05 <- c(data\_S06\_Var05= accuracy(arima.fc, test)['Test set', 'MAPE'])  
  
autoplot(arima.fc) + ggtitle("data\_S06\_Var05")



arima\_accuray\_Var07 <- c(data\_S06\_Var07= accuracy(arima.fc, test)['Test set', 'MAPE'])  
  
autoplot(arima.fc) + ggtitle("data\_S06\_Var07")



S06\_Var05.result <- cbind(ses = ses\_accuracy\_Var05, holt = holt\_accuracy\_Var05, ets = ets\_accuray\_Var05, arima = arima\_accuray\_Var05)  
cat("\tMAPE results by model - S06\_Var05\n\n")

## MAPE results by model - S06\_Var05

S06\_Var05.result

## ses holt ets arima  
## data\_S06\_Var05 35.46384 35.46193 35.46384 35.89939

S06\_Var07.result <- cbind(ses = ses\_accuracy\_Var07, holt = holt\_accuracy\_Var07, ets = ets\_accuray\_Var07, arima = arima\_accuray\_Var07)  
cat("\tMAPE results by model - S06\_Var07\n\n")

## MAPE results by model - S06\_Var07

S06\_Var07.result

## ses holt ets arima  
## data\_S06\_Var07 35.46384 35.46193 35.46384 35.89939

### Conclusions

It is found that MAPE accuracy measures of the forecasting models are the same. So, it might be similar patterns, have less significant variability over time and finally limitation of forecasting models.