Time Series S05

Group 3

6/23/2020

## DATA 624 Summer 2020, Project #1

Format: Group Effort, Group Representative will turn in your assignment. No conversations across groups regarding this project. DUE: 6/27/20 by Midnight ET Submission: Via Email – [scott.burk@sps.cuny.edu](mailto:scott.burk@sps.cuny.edu) Submission: Word Readable Document for Report (all in one), Excel Readable (all in one, separate sheets). File Convention: Group#\_Project1\_Summer624, example Group1\_Project1\_Summer624 GRADE: 70% Report, 30% Forecast Accuracy

### Overview

Your data is a de-identified Excel spreadsheet. Your assignment is to perform the appropriate analysis to forecast several series for 140 periods. You will have 1622 periods for your analysis. See Requirement #2 for more details.

### Requirement #1

You will turn in a written report. You need to write this report as if it the report will be routed in an office to personnel of vary different backgrounds. You need to be able to reach readers that have no data science background to full fledge data scientists. So, you need to explain what you have done, why and how you did it with both layman and technical terminology. Do not simply write this with me in mind. Visuals and output are expected, but it is not necessary to include every bit of analysis. In fact, a terse report with simple terminology is preferred to throwing in everything into a long, ad nausem report. Story telling is really taking on for data science, so please flex your muscles here. The report is part 1 of 2 requirements.

NOTE: We have covered a lot of material. I don’t want you to try every method to create your forecast. But you should perform fundamentals, visualize your data, and perform exploratory data analysis as appropriate.

### Requirement #2

Your second requirement is to produce forecasts. This workbook will contain at least 6 sheets where I will calculate your error rates. There will be one sheet (tab) for each Group – S01, S02, S03, SO4, S05, S06. You should order each sheet by the variable SeriesIND (low to high). Your source data is sorted this way, except there are all 6 groups present in one sheet which you must break out into 6 tabs. You will submit the data I sent AND the forward forecast for 140 periods. I want you to forecast the following

S01 – Forecast Var01, Var02 S02 – Forecast Var02, Var03 S03 – Forecast Var05, Var07 S04 – Forecast Var01, Var02 S05 – Forecast Var02, Var03 S06 – Forecast Var05, Var07

### Group S05

This group S05 is constituated of variables Var02 and Var03. Our goal is to find the best forecast for the variables Var02 and Var03 in S05. For that, we are going to process the dataset to change missing values and ouliers. After some statistic analysis, we can apply several models and check for the accuaracy of those models.

library(fpp2)

## Warning: package 'fpp2' was built under R version 3.6.3

## Loading required package: ggplot2

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.6.3

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

## Loading required package: fma

## Warning: package 'fma' was built under R version 3.6.3

## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 3.6.3

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(imputeTS)

## Warning: package 'imputeTS' was built under R version 3.6.3

### Load the dataset

full\_data <- readxl::read\_excel("Data Set for Class.xls")  
head(full\_data)

## # A tibble: 6 x 7  
## SeriesInd group Var01 Var02 Var03 Var05 Var07  
## <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 40669 S03 30.6 123432400 30.3 30.5 30.6  
## 2 40669 S02 10.3 60855800 10.0 10.2 10.3  
## 3 40669 S01 26.6 10369300 25.9 26.2 26.0  
## 4 40669 S06 27.5 39335700 26.8 27.0 27.3  
## 5 40669 S05 69.3 27809100 68.2 68.7 69.2  
## 6 40669 S04 17.2 16587400 16.9 16.9 17.1

### Subset the dataset

dt\_s05 <- subset(full\_data, group == 'S05', select = c(SeriesInd, Var02, Var03))  
summary(dt\_s05)

## SeriesInd Var02 Var03   
## Min. :40669 Min. : 4156600 Min. : 55.94   
## 1st Qu.:41304 1st Qu.: 11201200 1st Qu.: 77.21   
## Median :41946 Median : 14578800 Median : 84.87   
## Mean :41945 Mean : 16791350 Mean : 82.97   
## 3rd Qu.:42586 3rd Qu.: 20021200 3rd Qu.: 89.74   
## Max. :43221 Max. :118023500 Max. :103.95   
## NA's :141 NA's :145

#### Get the subsets Var02 and Var03

dt\_s05\_v2 <- dt\_s05 %>% filter(SeriesInd <= 43021) %>% select(Var02)

dt\_s05\_v3 <- dt\_s05 %>% filter(SeriesInd <= 43021) %>% select(Var03)

Explore the subsets Var02

summary(dt\_s05\_v2)

## Var02   
## Min. : 4156600   
## 1st Qu.: 11201200   
## Median : 14578800   
## Mean : 16791350   
## 3rd Qu.: 20021200   
## Max. :118023500   
## NA's :1

Var02 has 1 missing value

Explore the subsets Var03

summary(dt\_s05\_v3)

## Var03   
## Min. : 55.94   
## 1st Qu.: 77.21   
## Median : 84.87   
## Mean : 82.97   
## 3rd Qu.: 89.74   
## Max. :103.95   
## NA's :5

Var03 has 5 missing values Median and mean are in the same order.

#### Imputing missing values

dt\_s05\_v2 <- na\_interpolation(dt\_s05\_v2)  
summary(dt\_s05\_v2)

## Var02   
## Min. : 4156600   
## 1st Qu.: 11206550   
## Median : 14575700   
## Mean : 16788872   
## 3rd Qu.: 20014325   
## Max. :118023500

dt\_s05\_v3 <- na\_interpolation(dt\_s05\_v3)  
summary(dt\_s05\_v3)

## Var03   
## Min. : 55.94   
## 1st Qu.: 77.25   
## Median : 84.88   
## Mean : 82.97   
## 3rd Qu.: 89.71   
## Max. :103.95

#### converse Var02 and Var03 to time series

dt\_s05\_v2 <- ts(dt\_s05\_v2, frequency = 1)  
str(dt\_s05\_v2)

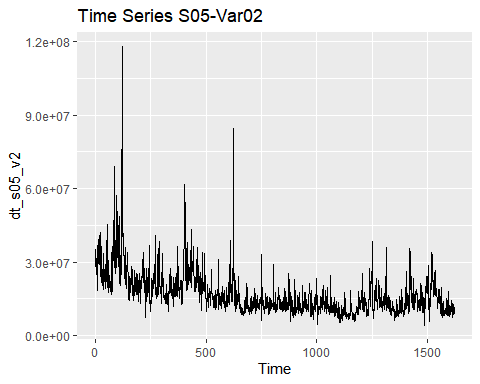
## Time-Series [1:1622, 1] from 1 to 1622: 27809100 30174700 35044700 27192100 24891800 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "Var02"

dt\_s05\_v3 <- ts(dt\_s05\_v3)  
str(dt\_s05\_v3)

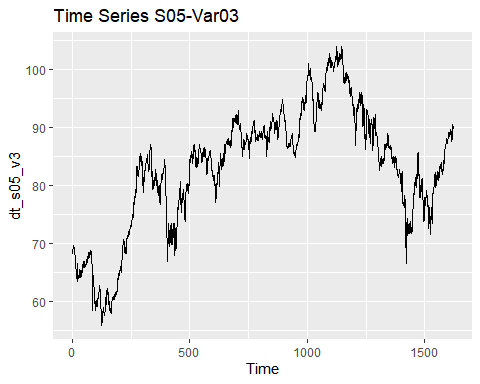
## Time-Series [1:1622, 1] from 1 to 1622: 68.2 68.8 69.3 69.4 69.2 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "Var03"

### Visualization

autoplot(dt\_s05\_v2) + ggtitle("Time Series S05-Var02")

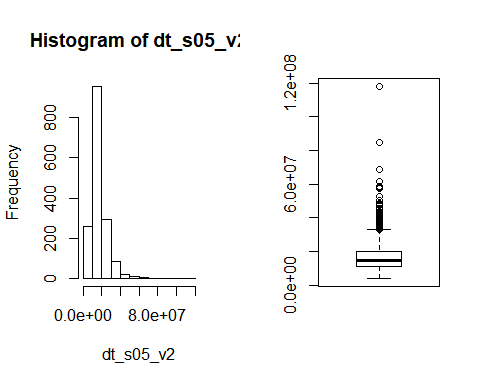
 Trend non seasonal time series

autoplot(dt\_s05\_v3) + ggtitle("Time Series S05-Var03")

 Trend non seasonal time serie. It can also be cyclic time series

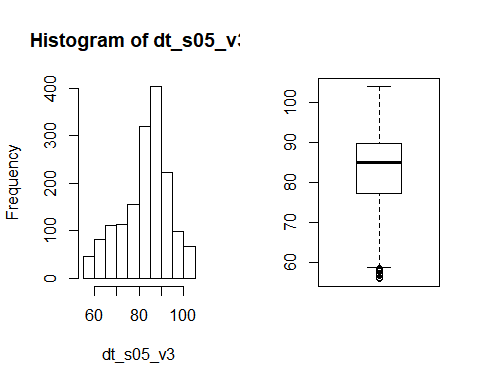
#### The distribution of data

par(mfrow= c(1,2))  
hist(dt\_s05\_v2)  
boxplot(dt\_s05\_v2)



Va02 is right skewed. We can supress the ultimate outliers that skew the distribution then transform the data.

par(mfrow= c(1,2))  
hist(dt\_s05\_v3)  
boxplot(dt\_s05\_v3)

 Var03 is nearly normal distributed and has outliers at the left.

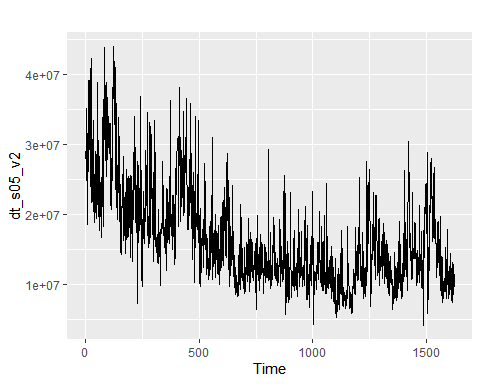
#### Removing the oulier

# check outlier(s)  
dt\_s05\_v2\_out <- tsoutliers(dt\_s05\_v2)  
dt\_s05\_v3\_out <- tsoutliers(dt\_s05\_v3)

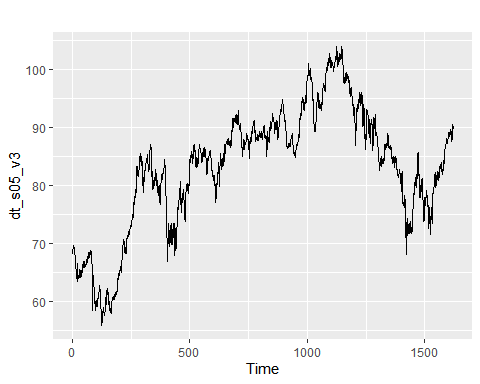
We have 31 extreme value in the time series V02 and only 1 in the time series V03. We usetsclean to replace outlier Cleaning the time series

dt\_s05\_v2 <- tsclean(dt\_s05\_v2)  
dt\_s05\_v3 <- tsclean(dt\_s05\_v3)

autoplot(dt\_s05\_v2)



autoplot(dt\_s05\_v3)



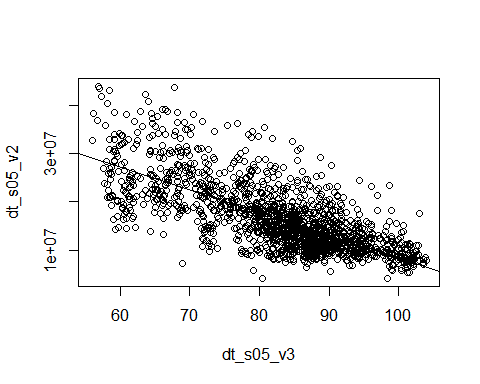
#### Correlation between V02 and V03 variables

There exists a linear correlation between V02 and V03 This is prove by the correlation test below

cor.test(dt\_s05\_v2, dt\_s05\_v3)

##   
## Pearson's product-moment correlation  
##   
## data: dt\_s05\_v2 and dt\_s05\_v3  
## t = -40.859, df = 1620, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.7355716 -0.6875738  
## sample estimates:  
## cor   
## -0.7124049

lmodel <- lm(dt\_s05\_v2~dt\_s05\_v3)  
plot(x = dt\_s05\_v3, y = dt\_s05\_v2)+abline(lmodel)



## integer(0)

(cor(dt\_s05\_v2,dt\_s05\_v3))^2

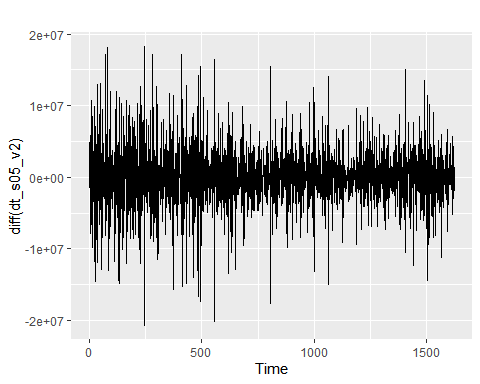
## Var03  
## Var02 0.5075207

V02 can explain V03 by 50.75% and vice-versa

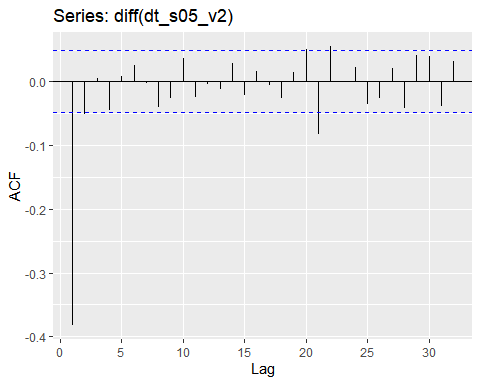
#### Verify the seasonalities and trends

ACF of Var02 difference

autoplot(diff(dt\_s05\_v2))

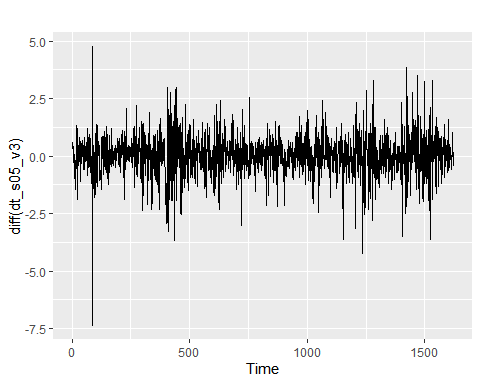


ggAcf(diff(dt\_s05\_v2))

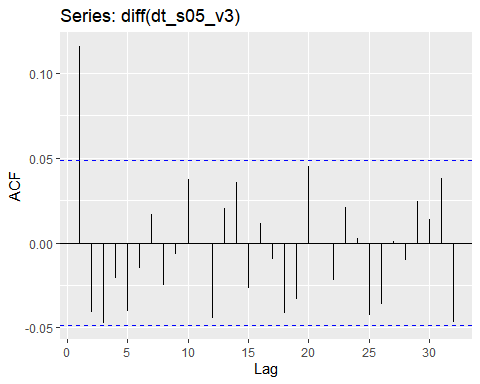
 There is one order autocorrelation This data has no seasonality

ACF of Var03 difference

autoplot(diff(dt\_s05\_v3))



ggAcf(diff(dt\_s05\_v3))

 The ACF shows one order correlation seasonality is insignificant.

#### Subset the train set

dt\_s05\_v2\_train <- window(dt\_s05\_v2, end = as.integer(length(dt\_s05\_v2)\*0.8))

dt\_s05\_v3\_train <- window(dt\_s05\_v3, end = as.integer(length(dt\_s05\_v3)\*0.8))

#### Look for lambda transformation

lambda2 <- BoxCox.lambda(dt\_s05\_v2)

lambda3 <- BoxCox.lambda(dt\_s05\_v3)

#### Apply the models to the train sets

The forecast horizon here is the length of test set

h\_test <- length(dt\_s05\_v2)- as.integer(length(dt\_s05\_v2)\*0.8)  
h\_test

## [1] 325

Get the arima model for Var02

dt\_s05\_v2\_farima\_fit <- dt\_s05\_v2\_train %>% auto.arima(lambda = lambda2, stepwise = FALSE) %>% forecast(h = h\_test)

Get the arima model for Var03

dt\_s05\_v3\_farima\_fit <- dt\_s05\_v3\_train %>% auto.arima(lambda = lambda3, stepwise = FALSE) %>% forecast(h = h\_test)

#### ETS model

Get the ets model for Var02

dt\_s05\_v2\_fets\_fit <- dt\_s05\_v2\_train %>% ets(lambda = lambda2) %>% forecast(h = h\_test)

Get the arima ets for Var03

dt\_s05\_v3\_fets\_fit <- dt\_s05\_v3\_train %>% ets(lambda = lambda3) %>% forecast(h = h\_test)

#### Naive method

Naive method for Var02

dt\_s05\_v2\_naive\_fit <- naive(dt\_s05\_v2\_train, h=h\_test)

Naive method for Var03

dt\_s05\_v3\_naive\_fit <- naive(dt\_s05\_v3\_train, h=h\_test)

### Accuracy compare to the naive model

#### MAPE accuracy of Var02 models

arima model

s05\_v2\_farima\_ac <- accuracy(dt\_s05\_v2\_farima\_fit, dt\_s05\_v2)["Test set", "MAPE"]

ets model

s05\_v2\_ets\_ac <- accuracy(dt\_s05\_v2\_fets\_fit, dt\_s05\_v2)["Test set", "MAPE"]

naive model

s05\_v2\_naive\_ac <- accuracy(dt\_s05\_v2\_naive\_fit, dt\_s05\_v2)["Test set", "MAPE"]

Comparing the MAPEs, the arima model for Var02 is the best.

#### MAPE accuracy for Var03 models

arima model

s05\_v3\_farima\_ac <- accuracy(dt\_s05\_v3\_farima\_fit, dt\_s05\_v3)["Test set", "MAPE"]

ets model

s05\_v3\_ets\_ac <- accuracy(dt\_s05\_v3\_fets\_fit, dt\_s05\_v3)["Test set", "MAPE"]

naive method

s05\_v3\_naive\_ac <- accuracy(dt\_s05\_v3\_naive\_fit, dt\_s05\_v3)["Test set", "MAPE"]

Using MAPE for accuracy, the naive method for Var03 is better than the arima of the same data. ETS model is the best. ### MAPE ACCURACY

s05\_v2\_MAPE <- c(s05\_v2\_farima\_ac, s05\_v2\_ets\_ac, s05\_v2\_naive\_ac)  
s05\_v3\_MAPE <- c(s05\_v3\_farima\_ac, s05\_v3\_ets\_ac, s05\_v3\_naive\_ac)  
s05\_MAPE <- matrix(rbind(s05\_v2\_MAPE, s05\_v3\_MAPE), nrow = 2 )  
rownames(s05\_MAPE) <- c("S05\_V02", "S05\_V03")  
colnames(s05\_MAPE) <- c("Arima", "ETS", "Naive")  
data.frame(s05\_MAPE)

## Arima ETS Naive  
## S05\_V02 29.847124 26.074663 26.807647  
## S05\_V03 9.104011 9.057531 9.057559

### Forecast the time series

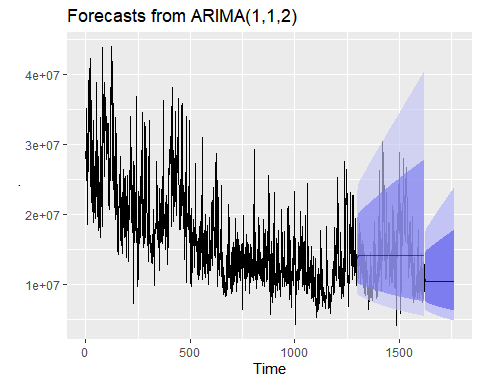
### Arima model

dt\_s05\_v2\_farima <- dt\_s05\_v2 %>% auto.arima(lambda = lambda2, stepwise = FALSE) %>% forecast(h = 140)

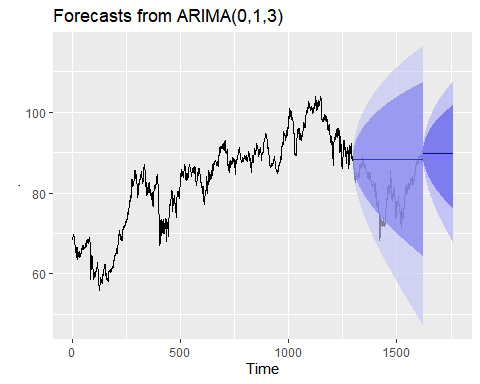
dt\_s05\_v3\_farima <- dt\_s05\_v3 %>% auto.arima(lambda = lambda3, stepwise = FALSE) %>% forecast(h = 140)

##### Var02 forecast

autoplot(dt\_s05\_v2\_farima)+autolayer(dt\_s05\_v2\_farima\_fit, alpha = 0.65)

 ##### Var03 forecast

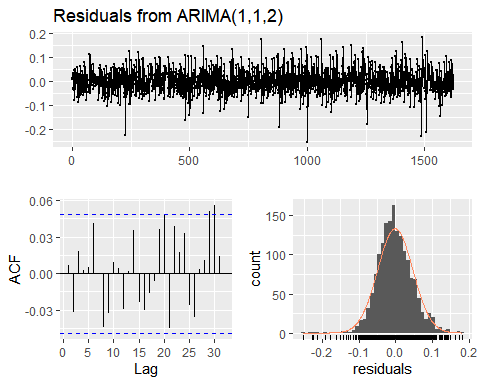
autoplot(dt\_s05\_v3\_farima) + autolayer(dt\_s05\_v3\_farima\_fit, alpha = 0.65)



#### Check the residuals if the model is valid

##### Var02 residuals

checkresiduals(dt\_s05\_v2\_farima)

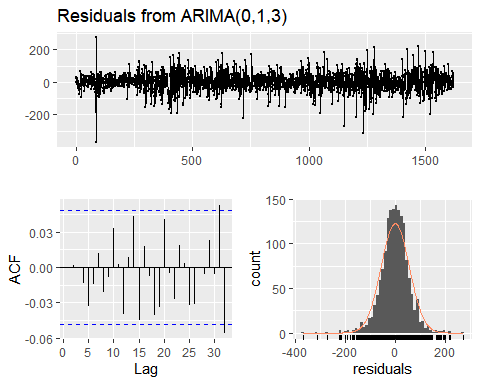


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,2)  
## Q\* = 10.064, df = 7, p-value = 0.185  
##   
## Model df: 3. Total lags used: 10

with p-value greater than 0.05, there is convaincing evidence that residuals for Var02 are white noise. On ACF, the residuals are uncorrelated. The histogram shows that the residuals are normal distributed.

##### Var03 residuals

checkresiduals(dt\_s05\_v3\_farima)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,3)  
## Q\* = 5.2616, df = 7, p-value = 0.6281  
##   
## Model df: 3. Total lags used: 10

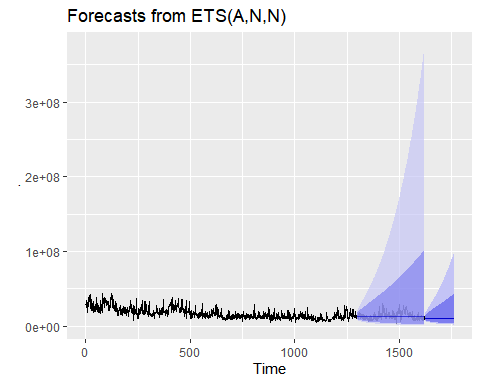
with p-value greater than 0.05, there is convaincing evidence that residuals for Var03 are white noise. On ACF, the residuals are uncorrelated. The histogram shows that the residuals are normal distributed.

### ETS model

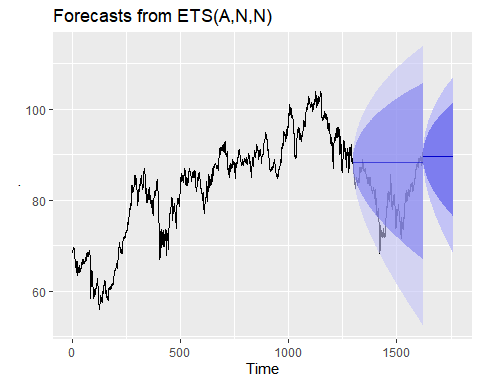
dt\_s05\_v2\_fets <- dt\_s05\_v2 %>% ets(lambda = lambda2) %>% forecast(h = 140)

dt\_s05\_v3\_fets <- dt\_s05\_v3 %>% ets(lambda = lambda3) %>% forecast(h = 140)

autoplot(dt\_s05\_v2\_fets) + autolayer(dt\_s05\_v2\_fets\_fit, alpha = 0.65)



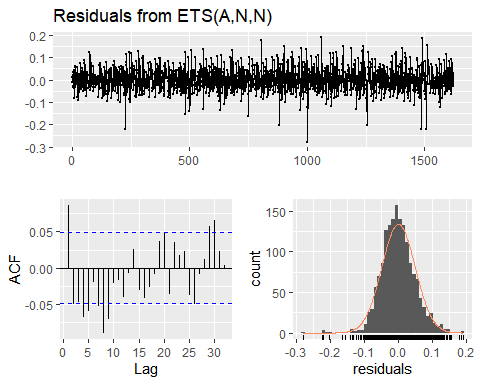
autoplot(dt\_s05\_v3\_fets) + autolayer(dt\_s05\_v3\_fets\_fit, alpha = 0.60)



#### Check the residuals if the model is valid

##### Var02 residuals

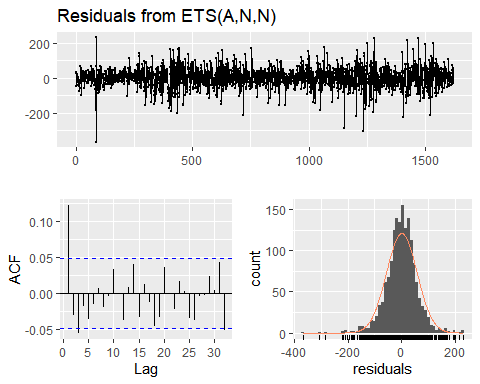
checkresiduals(dt\_s05\_v2\_fets)



##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,N,N)  
## Q\* = 60.114, df = 8, p-value = 4.427e-10  
##   
## Model df: 2. Total lags used: 10

##### Var03 residuals

checkresiduals(dt\_s05\_v3\_fets)



##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,N,N)  
## Q\* = 36.383, df = 8, p-value = 1.494e-05  
##   
## Model df: 2. 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 accuracies with the test set.

This is the metric MAPE on the entire dataset S05\_V02 or S05\_V03 using the ETS model

# MAPE  
print(paste0("MAPE for S05 Var02 is ", MLmetrics::MAPE(dt\_s05\_v2\_fets\_fit$fitted, dt\_s05\_v2)))

## [1] "MAPE for S05 Var02 is 0.17961672058084"

print(paste0("MAPE for S05 Var03 is ", MLmetrics::MAPE(dt\_s05\_v3\_fets\_fit$fitted, dt\_s05\_v3)))

## [1] "MAPE for S05 Var03 is 0.00779389269457009"