Predictive Analysis Project1 : Forecasting

Jered Ataky, Matthew Baker, Christopher Bloome, David Blumenstiel, Dhairav Chhatbar

6/25/2021

## Introduction

## Data Exploration & Preparation

### The data

Read the excel file and get the insight on the data

url<-"https://github.com/jnataky/Predictive\_Analytics/raw/main/Project1/Data\_Set\_for\_Class.xls"  
temp.file <- paste(tempfile(),".xls",sep = "")  
download.file(url, temp.file, mode = "wb")  
  
dataset <- read\_excel(temp.file, sheet = 1)  
  
str(dataset)

## tibble[,7] [10,572 x 7] (S3: tbl\_df/tbl/data.frame)  
## $ SeriesInd: num [1:10572] 40669 40669 40669 40669 40669 ...  
## $ group : chr [1:10572] "S03" "S02" "S01" "S06" ...  
## $ Var01 : num [1:10572] 30.6 10.3 26.6 27.5 69.3 ...  
## $ Var02 : num [1:10572] 1.23e+08 6.09e+07 1.04e+07 3.93e+07 2.78e+07 ...  
## $ Var03 : num [1:10572] 30.3 10.1 25.9 26.8 68.2 ...  
## $ Var05 : num [1:10572] 30.5 10.2 26.2 27 68.7 ...  
## $ Var07 : num [1:10572] 30.6 10.3 26 27.3 69.2 ...

Converting the series index to date format: Assuming that the data represent stock data, we are going to convert the series index in date using 1990-01-02 as reference to allow us to have only week days which will make sense for stock data.

df <- dataset %>%  
 mutate(SeriesInd = as.Date(SeriesInd, origin="1900-01-02"))  
head(df)

## # A tibble: 6 x 7  
## SeriesInd group Var01 Var02 Var03 Var05 Var07  
## <date> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2011-05-09 S03 30.6 123432400 30.3 30.5 30.6  
## 2 2011-05-09 S02 10.3 60855800 10.0 10.2 10.3  
## 3 2011-05-09 S01 26.6 10369300 25.9 26.2 26.0  
## 4 2011-05-09 S06 27.5 39335700 26.8 27.0 27.3  
## 5 2011-05-09 S05 69.3 27809100 68.2 68.7 69.2  
## 6 2011-05-09 S04 17.2 16587400 16.9 16.9 17.1

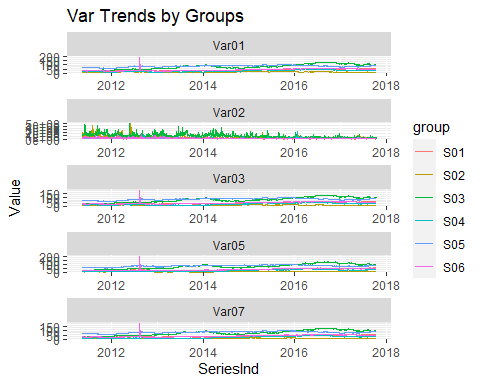
Missing data are distributed within the same observations for the most part. Group has negligeable effect on missing values.  
There are about 140 missing observations per group which are the number to forecast.

Drop NA Values

df <- df %>% gather("Var", "Value", 3:7) %>% drop\_na()

### Visualizations

df$Day <- weekdays(df$SeriesInd)  
df %>% ggplot( aes(x=SeriesInd, y=Value, color=group)) + facet\_wrap(~Var, scales = "free", ncol = 1) + geom\_line() +  
 labs(title = "Var Trends by Groups")



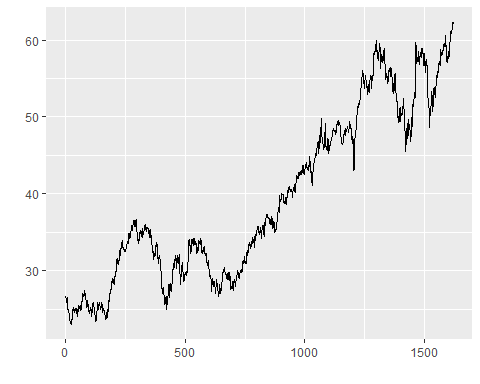
Nearly uniform but has consistent slight differences across variables

### Time series

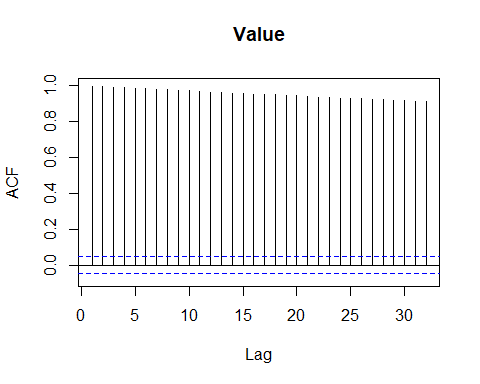
Separating data by group to make different time series

#### S01 Var01

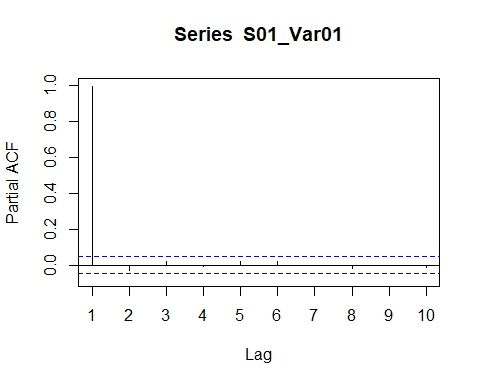
S01\_Var01 <- ts(df %>% filter(group == "S01", Var=="Var01") %>% select(Value), frequency = 1)  
autoplot(S01\_Var01)



Acf(S01\_Var01)



Pacf(S01\_Var01, lag.max = 10)



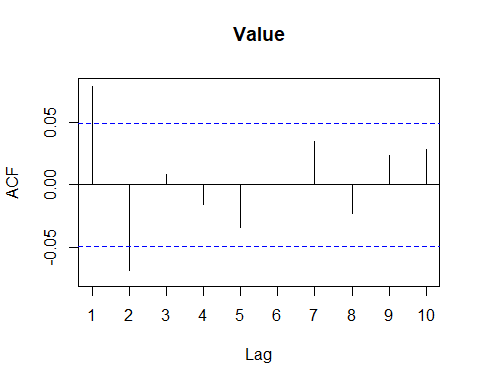
adf.test(S01\_Var01) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 1.62 0.975  
## [2,] 1 1.47 0.964  
## [3,] 2 1.63 0.975  
## [4,] 3 1.59 0.973  
## [5,] 4 1.62 0.975  
## [6,] 5 1.67 0.977  
## [7,] 6 1.68 0.977  
## [8,] 7 1.64 0.976  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -0.148 0.940  
## [2,] 1 -0.303 0.919  
## [3,] 2 -0.187 0.934  
## [4,] 3 -0.235 0.928  
## [5,] 4 -0.168 0.937  
## [6,] 5 -0.115 0.944  
## [7,] 6 -0.147 0.940  
## [8,] 7 -0.246 0.926  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -2.42 0.401  
## [2,] 1 -2.60 0.321  
## [3,] 2 -2.40 0.407  
## [4,] 3 -2.45 0.388  
## [5,] 4 -2.42 0.401  
## [6,] 5 -2.35 0.427  
## [7,] 6 -2.34 0.434  
## [8,] 7 -2.37 0.419  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

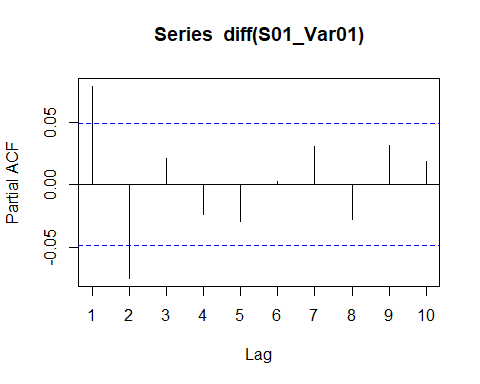
ndiffs(S01\_Var01) #Determines how many first differences are nessicary

## [1] 1

#Differenced acf/pacf plots  
  
Acf(diff(S01\_Var01), lag.max = 10)



Pacf(diff(S01\_Var01), lag.max = 10)



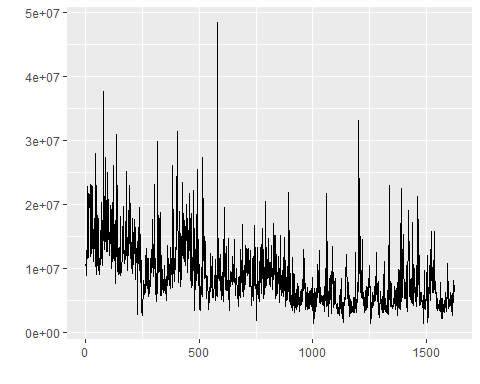
non-stationary; trend needs 1 first differencing

suggestive of autoregressive process because the PACF becomes insignificant much earlier than the ACF

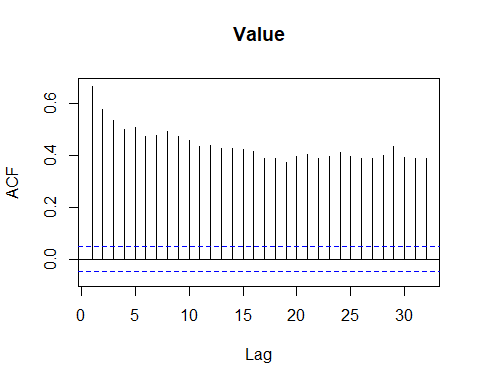
Both the ACF and PACF of the differenced series start positive, and become insignificant after lag 2. Could be two MA and two AR terms; maybe more AR terms because the PACF starts positive.

#### S01 Var02

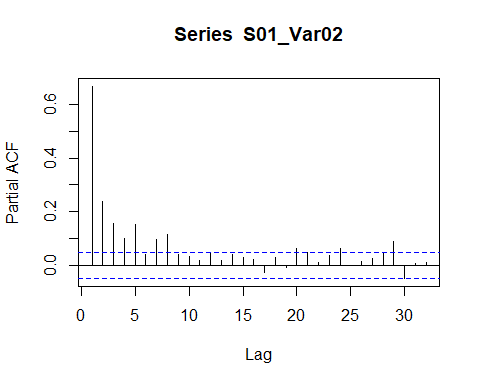
S01\_Var02 <- ts(df %>% filter(group == "S01", Var=="Var02") %>% select(Value), frequency = 1)  
autoplot(S01\_Var02)



Acf(S01\_Var02)



pacf(S01\_Var02)



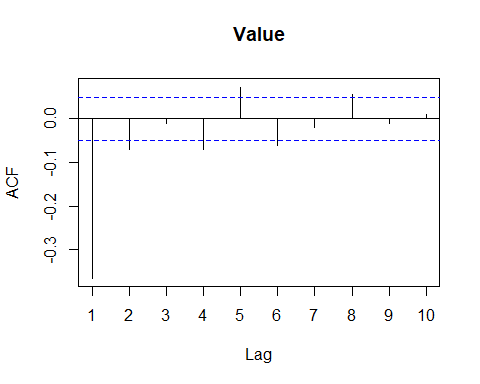
adf.test(S01\_Var02) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 -7.96 0.0100  
## [2,] 1 -5.49 0.0100  
## [3,] 2 -4.35 0.0100  
## [4,] 3 -3.75 0.0100  
## [5,] 4 -3.13 0.0100  
## [6,] 5 -2.92 0.0100  
## [7,] 6 -2.69 0.0100  
## [8,] 7 -2.41 0.0172  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -17.95 0.01  
## [2,] 1 -12.84 0.01  
## [3,] 2 -10.43 0.01  
## [4,] 3 -9.12 0.01  
## [5,] 4 -7.63 0.01  
## [6,] 5 -7.20 0.01  
## [7,] 6 -6.47 0.01  
## [8,] 7 -5.71 0.01  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -23.26 0.01  
## [2,] 1 -17.25 0.01  
## [3,] 2 -14.44 0.01  
## [4,] 3 -12.94 0.01  
## [5,] 4 -11.01 0.01  
## [6,] 5 -10.57 0.01  
## [7,] 6 -9.51 0.01  
## [8,] 7 -8.44 0.01  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

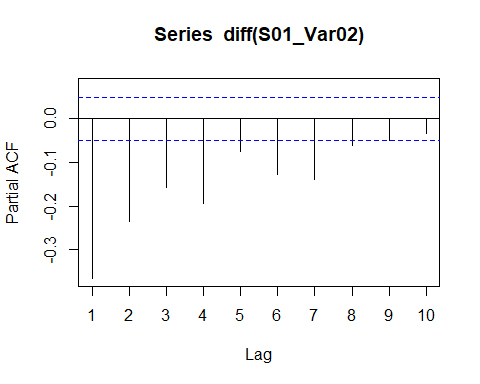
ndiffs(S01\_Var02) #Determines how many first differences are nessicary

## [1] 1

#Differenced acf/pacf plots  
  
Acf(diff(S01\_Var02), lag.max = 10)



Pacf(diff(S01\_Var02), lag.max = 10)



stationary

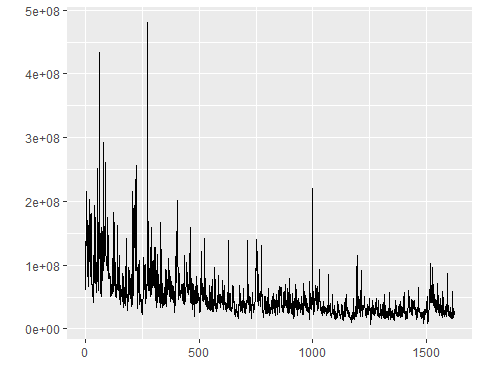
needs 1 first differencing

suggestive of autoregressive process

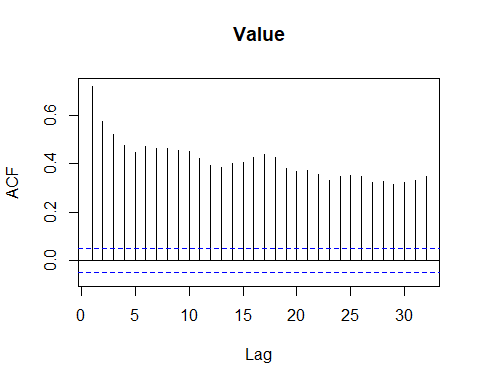
The ACF of the differenced series the ACF of the differenced series cuts off more quickly than the PACF, indicating primarily AR. The ACF cuts off past lag 2, and the PACF takes a while. Could try several AR terms and fewer MA terms

#### S02 Var02

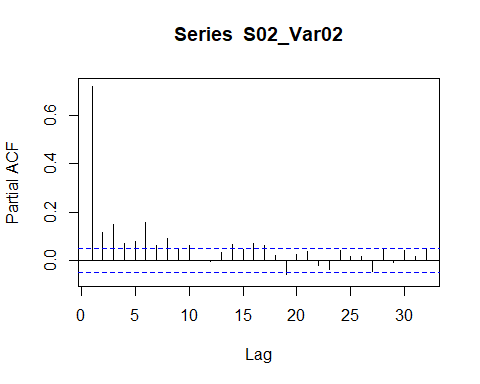
S02\_Var02 <- ts(df %>% filter(group == "S02", Var=="Var02") %>% select(Value), frequency = 1)  
autoplot(S02\_Var02)



Acf(S02\_Var02)



Pacf(S02\_Var02)



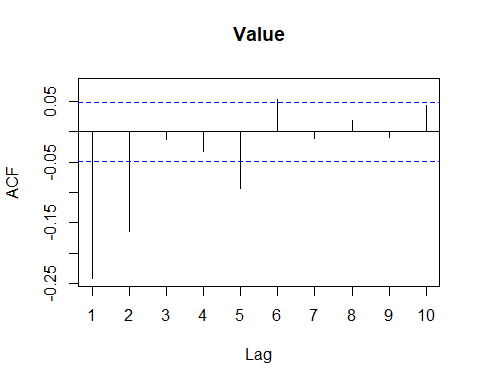
adf.test(S02\_Var02) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 -9.34 0.01  
## [2,] 1 -7.65 0.01  
## [3,] 2 -6.20 0.01  
## [4,] 3 -5.44 0.01  
## [5,] 4 -4.85 0.01  
## [6,] 5 -4.18 0.01  
## [7,] 6 -3.93 0.01  
## [8,] 7 -3.59 0.01  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -16.19 0.01  
## [2,] 1 -13.51 0.01  
## [3,] 2 -11.10 0.01  
## [4,] 3 -9.93 0.01  
## [5,] 4 -8.95 0.01  
## [6,] 5 -7.61 0.01  
## [7,] 6 -7.14 0.01  
## [8,] 7 -6.48 0.01  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -20.40 0.01  
## [2,] 1 -17.39 0.01  
## [3,] 2 -14.57 0.01  
## [4,] 3 -13.35 0.01  
## [5,] 4 -12.24 0.01  
## [6,] 5 -10.44 0.01  
## [7,] 6 -9.84 0.01  
## [8,] 7 -8.97 0.01  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

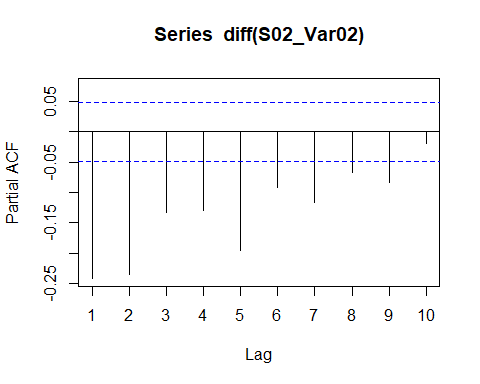
ndiffs(S02\_Var02) #Determines how many first differences are nessicary

## [1] 1

#Differenced acf/pacf plots  
  
Acf(diff(S02\_Var02), lag.max = 10)



Pacf(diff(S02\_Var02), lag.max = 10)



stationary

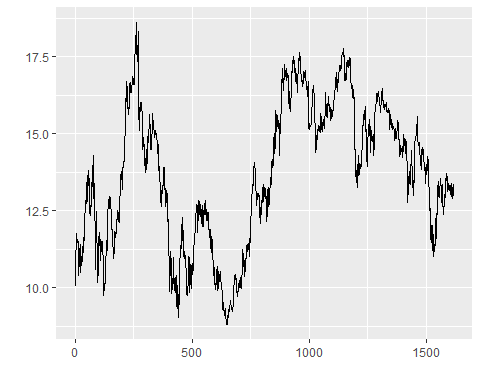
needs 1 first differencing

suggestive of autoregressive process

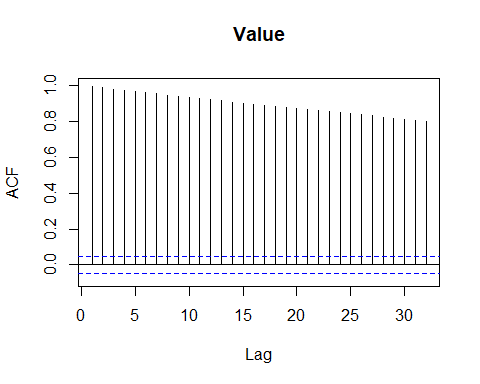
The ACF of the differenced series the ACF of the differenced series cuts off more quickly than the PACF, indicating primarily AR. The ACF cuts off past lag 2, and the PACF takes a while. Could try several AR terms and fewer MA terms

#### S02 Var03

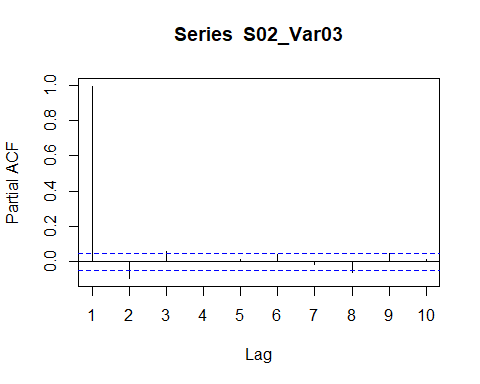
S02\_Var03 <- ts(df %>% filter(group == "S02", Var=="Var03") %>% select(Value), frequency = 1) %>% tsclean()  
autoplot(S02\_Var03)



Acf(S02\_Var03)



Pacf(S02\_Var03, lag.max = 10)



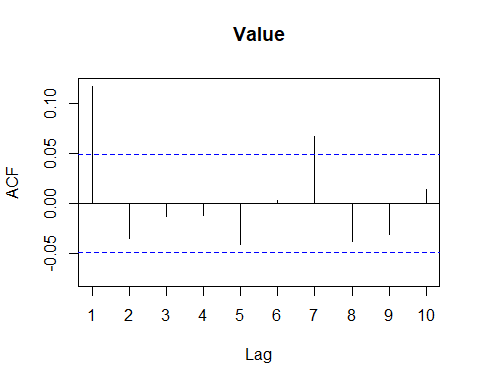
adf.test(S02\_Var03) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 -0.116 0.611  
## [2,] 1 -0.214 0.582  
## [3,] 2 -0.235 0.576  
## [4,] 3 -0.242 0.574  
## [5,] 4 -0.246 0.573  
## [6,] 5 -0.247 0.573  
## [7,] 6 -0.248 0.572  
## [8,] 7 -0.272 0.566  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -2.46 0.1448  
## [2,] 1 -2.65 0.0868  
## [3,] 2 -2.45 0.1490  
## [4,] 3 -2.43 0.1561  
## [5,] 4 -2.39 0.1727  
## [6,] 5 -2.27 0.2196  
## [7,] 6 -2.31 0.2053  
## [8,] 7 -2.47 0.1417  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -2.35 0.430  
## [2,] 1 -2.61 0.320  
## [3,] 2 -2.43 0.396  
## [4,] 3 -2.41 0.401  
## [5,] 4 -2.38 0.417  
## [6,] 5 -2.26 0.466  
## [7,] 6 -2.30 0.450  
## [8,] 7 -2.48 0.376  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

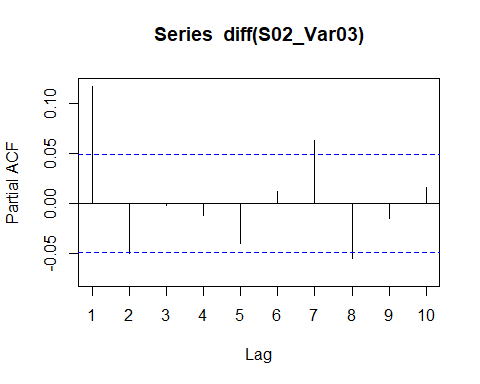
ndiffs(S02\_Var03) #Determines how many first differences are nessicary

## [1] 1

#Differenced acf/pacf plots  
  
Acf(diff(S02\_Var03), lag.max = 10)



Pacf(diff(S02\_Var03), lag.max = 10)



A nice big outlier here. It was removed with tsclean()

Not stationary; maybe not trend?

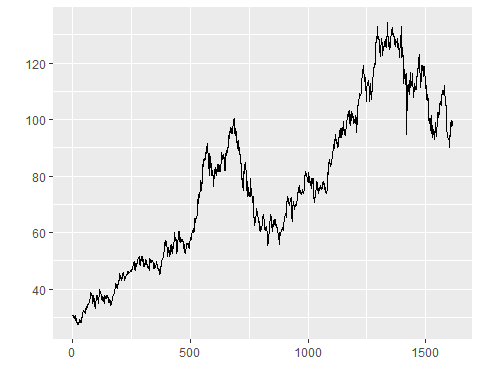
needs 1 first differencing

suggestive of autoregressive process

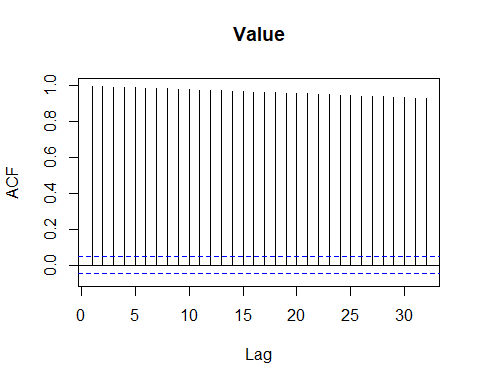
ACF of the differenced series cuts off from a negative value very quickly, indicating AR process. It takes a while for the PACF of the differenced series to drop off, so a higher order AR process. ACF is negative, so maybe an MA term would work well in addition.

#### S03 Var05

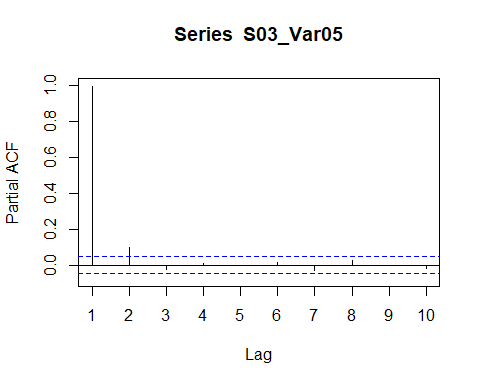
S03\_Var05 <- ts(df %>% filter(group == "S03", Var=="Var05") %>% select(Value), frequency = 1)  
autoplot(S03\_Var05)



Acf(S03\_Var05)



Pacf(S03\_Var05, lag.max = 10)



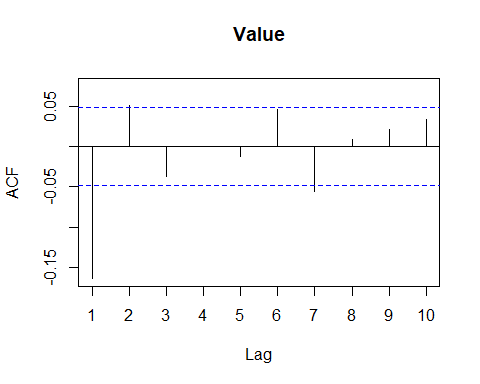
adf.test(S03\_Var05) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 0.506 0.789  
## [2,] 1 0.717 0.850  
## [3,] 2 0.683 0.840  
## [4,] 3 0.718 0.851  
## [5,] 4 0.733 0.855  
## [6,] 5 0.747 0.859  
## [7,] 6 0.692 0.843  
## [8,] 7 0.749 0.859  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -1.58 0.498  
## [2,] 1 -1.49 0.527  
## [3,] 2 -1.51 0.523  
## [4,] 3 -1.51 0.522  
## [5,] 4 -1.51 0.520  
## [6,] 5 -1.50 0.524  
## [7,] 6 -1.53 0.514  
## [8,] 7 -1.53 0.515  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -2.13 0.521  
## [2,] 1 -1.71 0.702  
## [3,] 2 -1.77 0.675  
## [4,] 3 -1.71 0.700  
## [5,] 4 -1.69 0.710  
## [6,] 5 -1.66 0.723  
## [7,] 6 -1.76 0.678  
## [8,] 7 -1.66 0.722  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

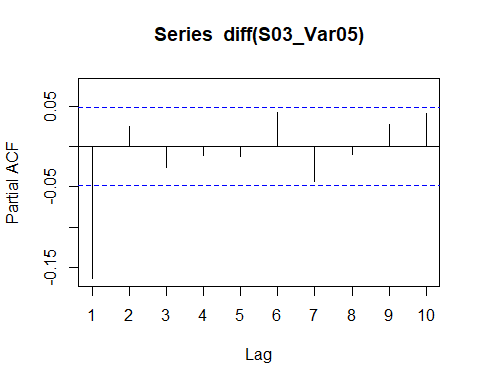
ndiffs(S03\_Var05) #Determines how many first differences are nessicary

## [1] 1

#Differenced acf/pacf plots  
  
Acf(diff(S03\_Var05), lag.max = 10)



Pacf(diff(S03\_Var05), lag.max = 10)



non-stationary; trend

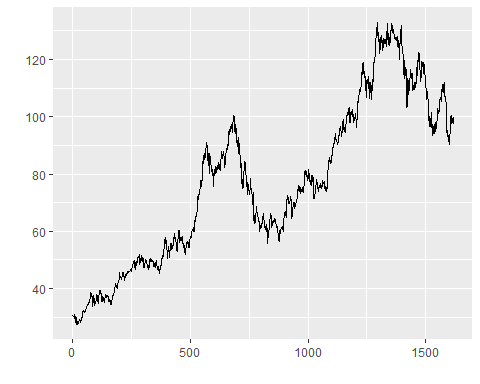
needs 1 first differencing

suggestive of autoregressive process

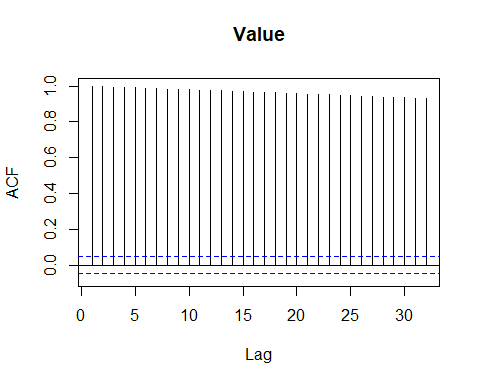
Both PACF and ACF of the differenced series cut off after 1 lag. Maybe one MA and AR term

#### S03 Var07

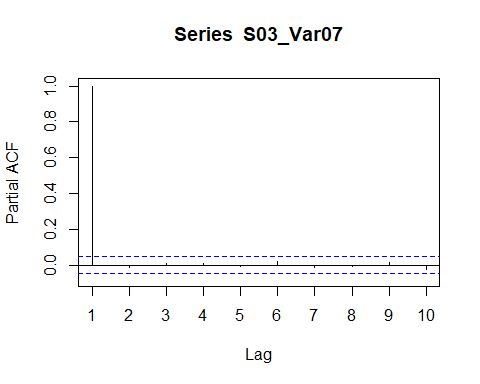
S03\_Var07 <- ts(df %>% filter(group == "S03", Var=="Var07") %>% select(Value), frequency = 1)  
autoplot(S03\_Var07)



Acf(S03\_Var07)



Pacf(S03\_Var07, lag.max = 10)



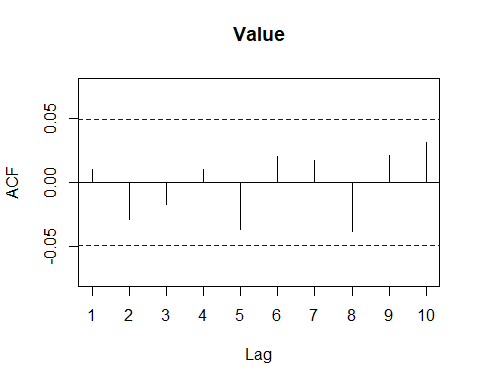
adf.test(S03\_Var07) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 0.632 0.826  
## [2,] 1 0.617 0.821  
## [3,] 2 0.658 0.833  
## [4,] 3 0.679 0.839  
## [5,] 4 0.663 0.835  
## [6,] 5 0.715 0.850  
## [7,] 6 0.688 0.842  
## [8,] 7 0.663 0.835  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -1.53 0.514  
## [2,] 1 -1.53 0.513  
## [3,] 2 -1.54 0.511  
## [4,] 3 -1.54 0.512  
## [5,] 4 -1.54 0.512  
## [6,] 5 -1.53 0.513  
## [7,] 6 -1.55 0.507  
## [8,] 7 -1.55 0.508  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -1.77 0.673  
## [2,] 1 -1.80 0.661  
## [3,] 2 -1.73 0.690  
## [4,] 3 -1.69 0.707  
## [5,] 4 -1.72 0.696  
## [6,] 5 -1.63 0.735  
## [7,] 6 -1.68 0.711  
## [8,] 7 -1.72 0.695  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

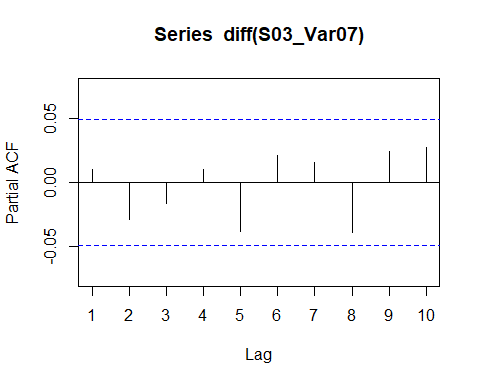
ndiffs(S03\_Var07) #Determines how many first differences are nessicary

## [1] 1

Acf(diff(S03\_Var07), lag.max = 10)



Pacf(diff(S03\_Var07), lag.max = 10)



non-stationary; trend and

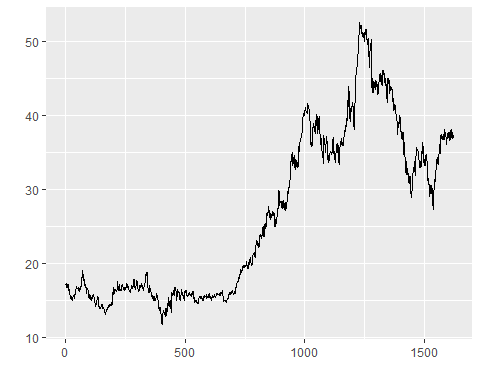
needs 1 first differencing

suggestive of autoregressive process

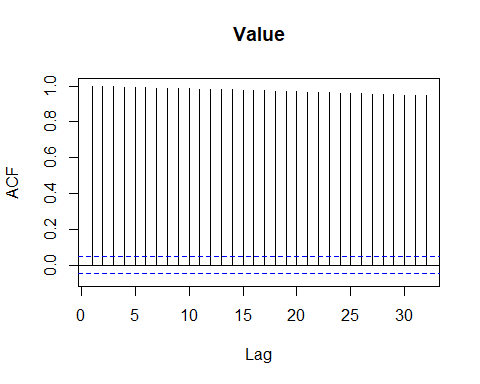
No significant lags in either ACF of PACF of the differenced series. This one has no AR or MA terms

#### S04 Var01

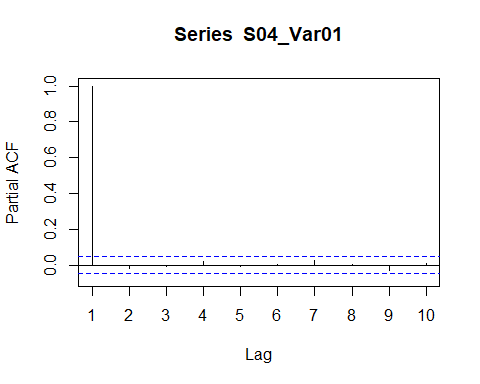
S04\_Var01 <- ts(df %>% filter(group == "S04", Var=="Var01") %>% select(Value), frequency = 1)  
autoplot(S04\_Var01)



Acf(S04\_Var01)



Pacf(S04\_Var01, lag.max = 10)



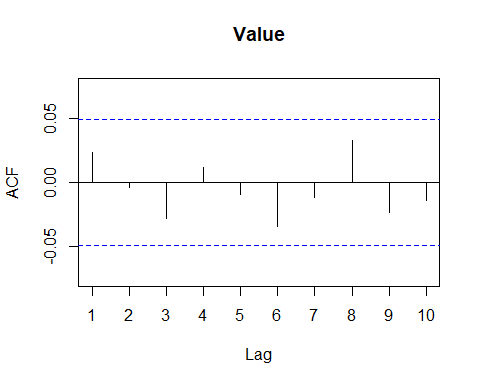
adf.test(S04\_Var01) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 0.557 0.804  
## [2,] 1 0.526 0.795  
## [3,] 2 0.528 0.796  
## [4,] 3 0.575 0.809  
## [5,] 4 0.563 0.806  
## [6,] 5 0.572 0.809  
## [7,] 6 0.616 0.821  
## [8,] 7 0.624 0.823  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -0.850 0.755  
## [2,] 1 -0.872 0.747  
## [3,] 2 -0.865 0.749  
## [4,] 3 -0.855 0.753  
## [5,] 4 -0.873 0.747  
## [6,] 5 -0.860 0.751  
## [7,] 6 -0.827 0.763  
## [8,] 7 -0.814 0.768  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -1.74 0.685  
## [2,] 1 -1.78 0.668  
## [3,] 2 -1.78 0.669  
## [4,] 3 -1.71 0.698  
## [5,] 4 -1.73 0.693  
## [6,] 5 -1.72 0.697  
## [7,] 6 -1.66 0.720  
## [8,] 7 -1.66 0.723  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

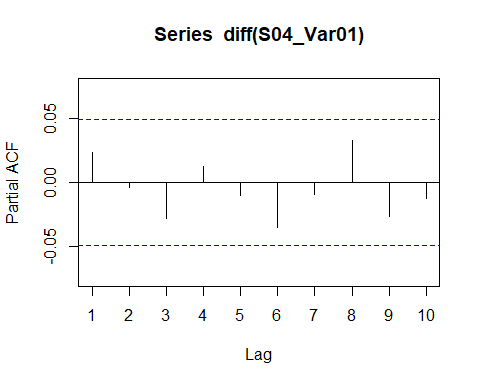
ndiffs(S04\_Var01) #Determines how many first differences are nessicary

## [1] 1

Acf(diff(S04\_Var01), lag.max = 10)



Pacf(diff(S04\_Var01), lag.max = 10)



non-stationary; trend

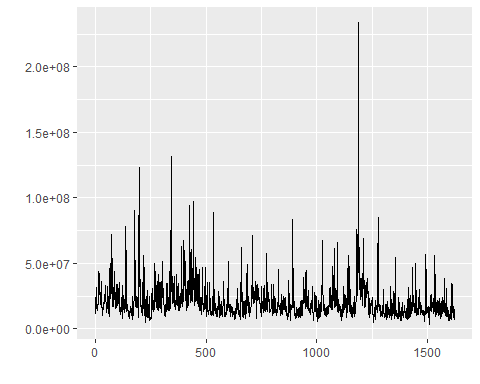
needs 1 first difference

suggestive of autoregressive process

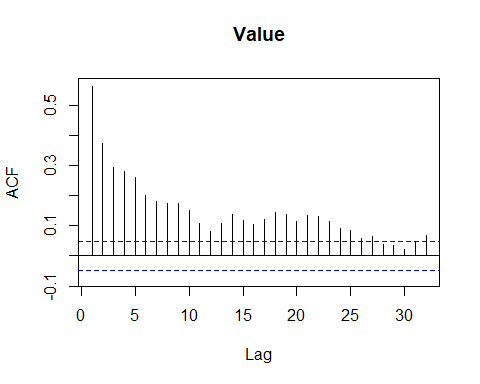
No significant lags in either ACF of PACF of the differenced series. This one has no AR or MA terms

#### S04 Var02

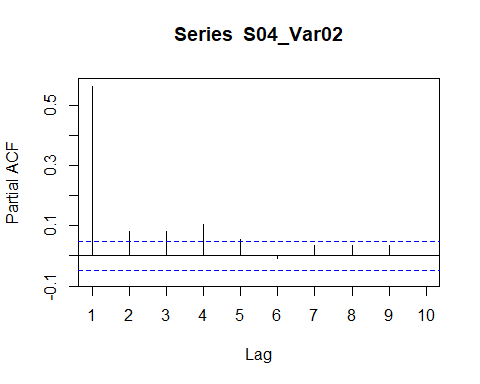
S04\_Var02 <- ts(df %>% filter(group == "S04", Var=="Var02") %>% select(Value), frequency = 1)  
autoplot(S04\_Var02)



Acf(S04\_Var02)



Pacf(S04\_Var02, lag.max = 10)



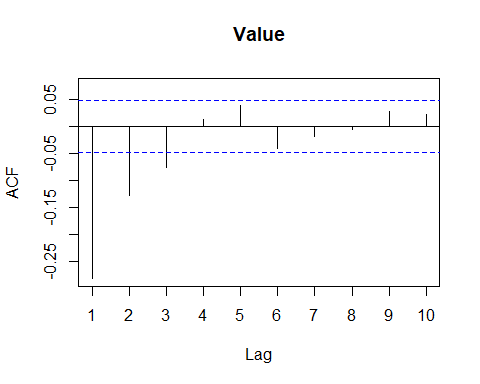
adf.test(S04\_Var02) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 -10.86 0.01  
## [2,] 1 -8.30 0.01  
## [3,] 2 -6.69 0.01  
## [4,] 3 -5.48 0.01  
## [5,] 4 -4.79 0.01  
## [6,] 5 -4.51 0.01  
## [7,] 6 -4.10 0.01  
## [8,] 7 -3.75 0.01  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -21.23 0.01  
## [2,] 1 -17.30 0.01  
## [3,] 2 -14.62 0.01  
## [4,] 3 -12.33 0.01  
## [5,] 4 -11.13 0.01  
## [6,] 5 -10.79 0.01  
## [7,] 6 -10.05 0.01  
## [8,] 7 -9.39 0.01  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -21.62 0.01  
## [2,] 1 -17.68 0.01  
## [3,] 2 -15.00 0.01  
## [4,] 3 -12.67 0.01  
## [5,] 4 -11.47 0.01  
## [6,] 5 -11.14 0.01  
## [7,] 6 -10.41 0.01  
## [8,] 7 -9.75 0.01  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

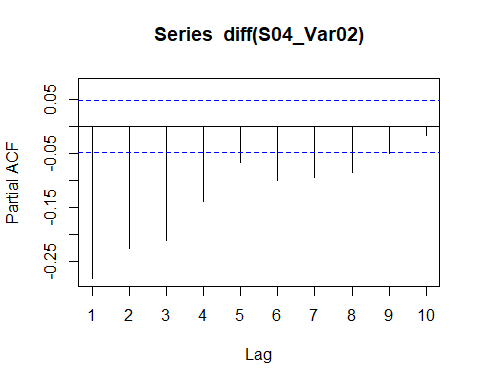
ndiffs(S04\_Var02) #Determines how many first differences are nessicary

## [1] 1

Acf(diff(S04\_Var02), lag.max = 10)



Pacf(diff(S04\_Var02), lag.max = 10)



Stationary

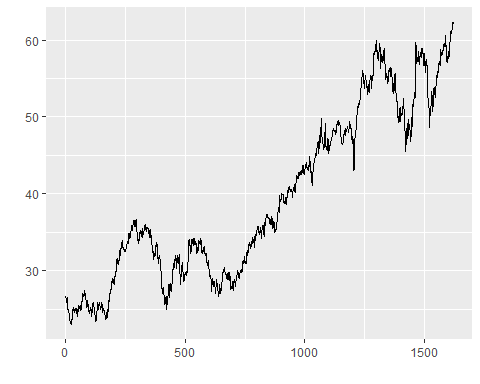
needs 1 first difference

suggestive of autoregressive process, but less so than others

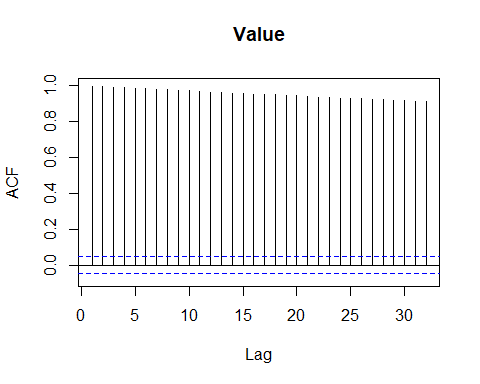
A high number of significant lags in the PACF of the differenced series, and a few significant lags in the ACF. Probably a higher number of AR terms, maybe a couple MA terms

#### S05 Var02

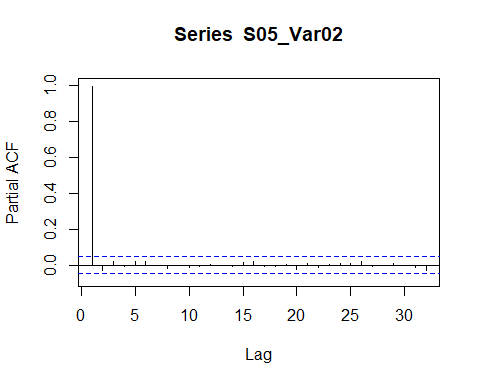
S05\_Var02 <- ts(df %>% filter(group == "S01", Var=="Var01") %>% select(Value), frequency = 1)  
autoplot(S05\_Var02)



Acf(S05\_Var02)



Pacf(S05\_Var02)



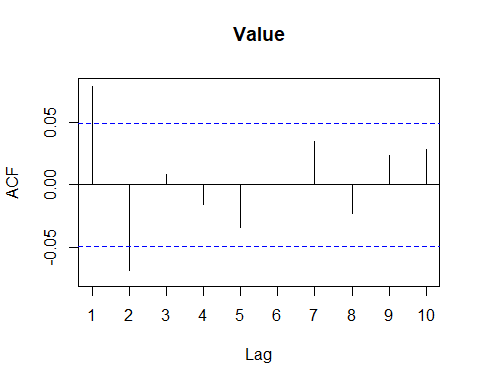
adf.test(S05\_Var02) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 1.62 0.975  
## [2,] 1 1.47 0.964  
## [3,] 2 1.63 0.975  
## [4,] 3 1.59 0.973  
## [5,] 4 1.62 0.975  
## [6,] 5 1.67 0.977  
## [7,] 6 1.68 0.977  
## [8,] 7 1.64 0.976  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -0.148 0.940  
## [2,] 1 -0.303 0.919  
## [3,] 2 -0.187 0.934  
## [4,] 3 -0.235 0.928  
## [5,] 4 -0.168 0.937  
## [6,] 5 -0.115 0.944  
## [7,] 6 -0.147 0.940  
## [8,] 7 -0.246 0.926  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -2.42 0.401  
## [2,] 1 -2.60 0.321  
## [3,] 2 -2.40 0.407  
## [4,] 3 -2.45 0.388  
## [5,] 4 -2.42 0.401  
## [6,] 5 -2.35 0.427  
## [7,] 6 -2.34 0.434  
## [8,] 7 -2.37 0.419  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

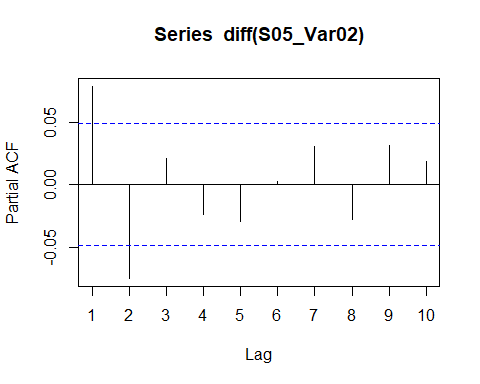
ndiffs(S05\_Var02) #Determines how many first differences are nessicary

## [1] 1

Acf(diff(S05\_Var02), lag.max = 10)



Pacf(diff(S05\_Var02), lag.max = 10)



non-stationary; trend

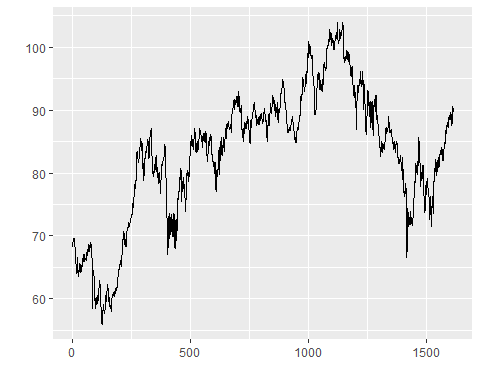
needs 1 first difference

suggestive of autoregressive process

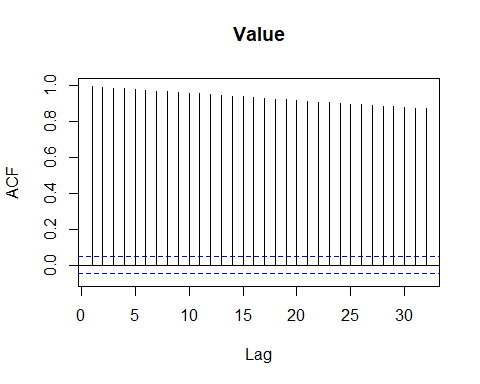
Similar ACF and PACF of differenced series. Maybe one of each term, or two AR because PACF starts positive

#### S05 Var03

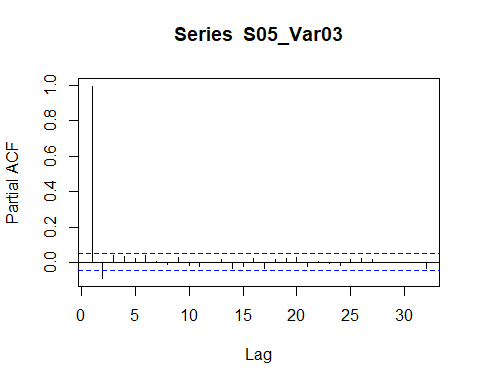
S05\_Var03 <- ts(df %>% filter(group == "S05", Var=="Var03") %>% select(Value), frequency = 1)  
autoplot(S05\_Var03)



Acf(S05\_Var03)



Pacf(S05\_Var03)



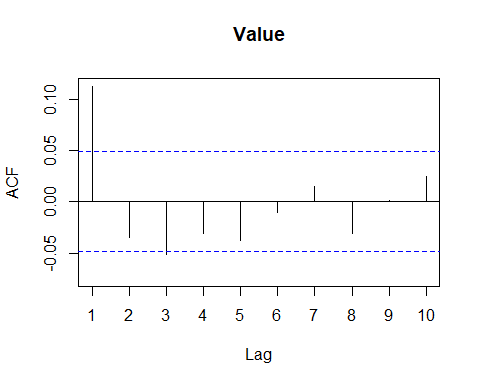
adf.test(S05\_Var03) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 0.342 0.742  
## [2,] 1 0.242 0.713  
## [3,] 2 0.267 0.721  
## [4,] 3 0.300 0.730  
## [5,] 4 0.321 0.736  
## [6,] 5 0.336 0.741  
## [7,] 6 0.344 0.743  
## [8,] 7 0.340 0.742  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -1.94 0.352  
## [2,] 1 -2.10 0.287  
## [3,] 2 -2.01 0.326  
## [4,] 3 -1.94 0.353  
## [5,] 4 -1.91 0.364  
## [6,] 5 -1.84 0.391  
## [7,] 6 -1.84 0.392  
## [8,] 7 -1.87 0.382  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -2.09 0.539  
## [2,] 1 -2.33 0.436  
## [3,] 2 -2.22 0.484  
## [4,] 3 -2.13 0.522  
## [5,] 4 -2.08 0.541  
## [6,] 5 -2.01 0.574  
## [7,] 6 -2.00 0.578  
## [8,] 7 -2.02 0.566  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

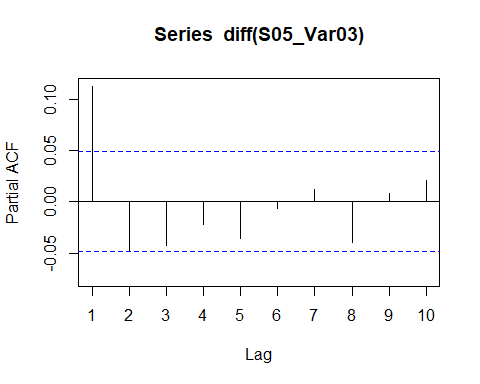
ndiffs(S05\_Var03) #Determines how many first differences are nessicary

## [1] 1

Acf(diff(S05\_Var03), lag.max = 10)



Pacf(diff(S05\_Var03), lag.max = 10)



non-stationary (probably); probably trend

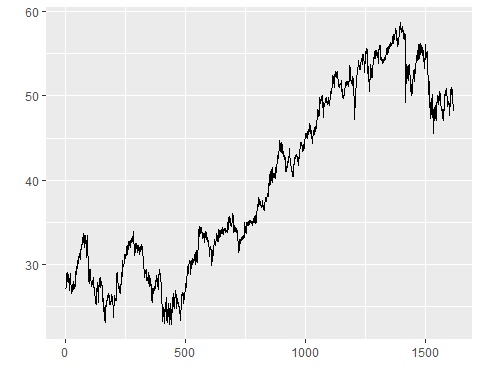
needs 1 first difference

suggestive of autoregressive process

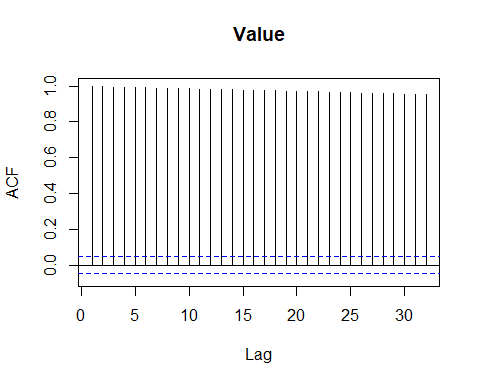
Similar ACF and PACF of differenced series. Maybe one of each term, or two AR because PACF starts positive

#### S06 Var05

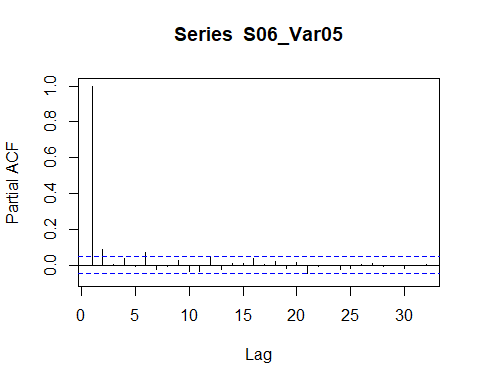
S06\_Var05 <- ts(df %>% filter(group == "S06", Var=="Var05") %>% select(Value), frequency = 1) %>% tsclean()  
autoplot(S06\_Var05)



Acf(S06\_Var05)



Pacf(S06\_Var05)



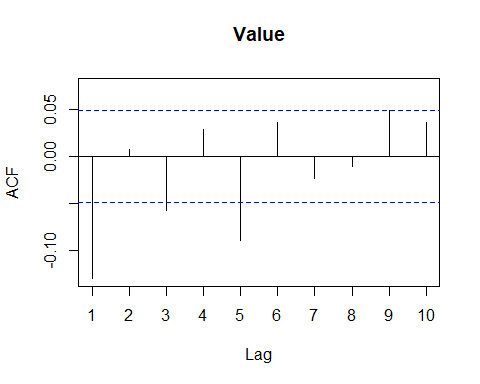
adf.test(S06\_Var05) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 0.561 0.805  
## [2,] 1 0.706 0.847  
## [3,] 2 0.695 0.844  
## [4,] 3 0.769 0.865  
## [5,] 4 0.728 0.853  
## [6,] 5 0.841 0.886  
## [7,] 6 0.839 0.885  
## [8,] 7 0.875 0.896  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -1.258 0.610  
## [2,] 1 -1.139 0.652  
## [3,] 2 -1.092 0.669  
## [4,] 3 -1.038 0.688  
## [5,] 4 -1.009 0.699  
## [6,] 5 -0.930 0.727  
## [7,] 6 -0.965 0.714  
## [8,] 7 -0.975 0.711  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -2.47 0.379  
## [2,] 1 -2.11 0.531  
## [3,] 2 -2.12 0.523  
## [4,] 3 -1.99 0.581  
## [5,] 4 -2.07 0.548  
## [6,] 5 -1.88 0.626  
## [7,] 6 -1.88 0.629  
## [8,] 7 -1.82 0.655  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

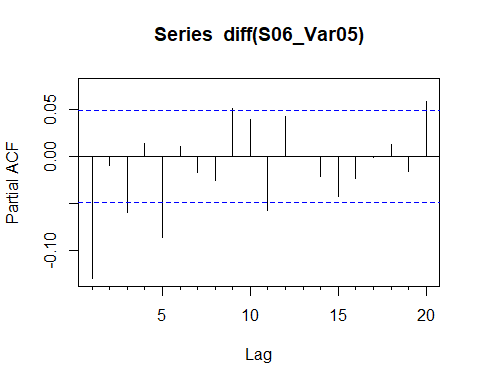
ndiffs(S06\_Var05) #Determines how many first differences are nessicary

## [1] 1

Acf(diff(S06\_Var05), lag.max = 10)



Pacf(diff(S06\_Var05), lag.max = 20)



needed outlier cleaning

non-stationary

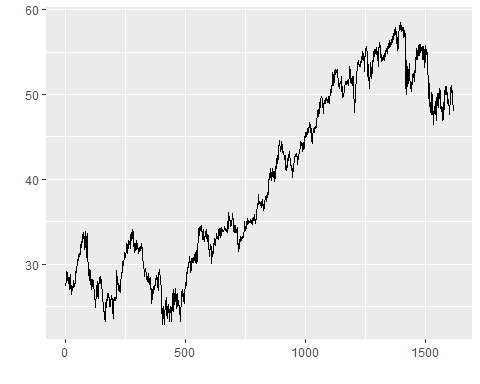
needs 1 first difference

suggestive of autoregressive process

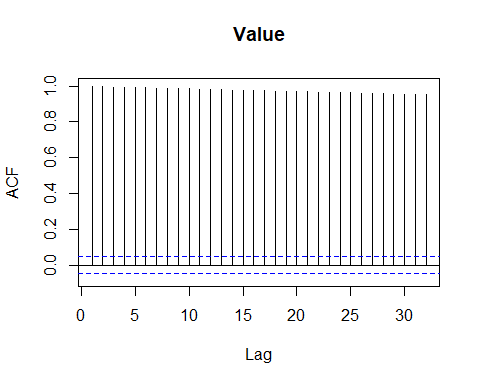
The ACF of the differenced series cut off after 1 lag, while the PACF trailed on for a while. Could be just one MA term, maybey 2

#### S06 Var07

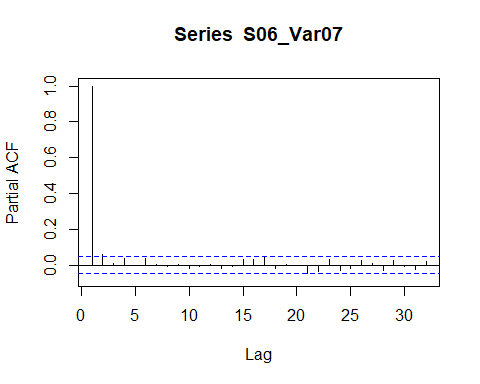
S06\_Var07 <- ts(df %>% filter(group == "S06", Var=="Var07") %>% select(Value), frequency = 1) %>% tsclean()  
autoplot(S06\_Var07)



Acf(S06\_Var07)



Pacf(S06\_Var07)



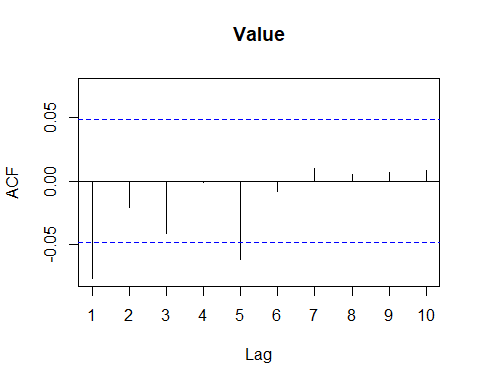
adf.test(S06\_Var07) #Unit root test. This tests if differencing is nessiary

## Augmented Dickey-Fuller Test   
## alternative: stationary   
##   
## Type 1: no drift no trend   
## lag ADF p.value  
## [1,] 0 0.561 0.805  
## [2,] 1 0.628 0.825  
## [3,] 2 0.658 0.833  
## [4,] 3 0.687 0.842  
## [5,] 4 0.702 0.846  
## [6,] 5 0.794 0.872  
## [7,] 6 0.839 0.885  
## [8,] 7 0.820 0.880  
## Type 2: with drift no trend   
## lag ADF p.value  
## [1,] 0 -1.235 0.618  
## [2,] 1 -1.133 0.655  
## [3,] 2 -1.107 0.664  
## [4,] 3 -1.020 0.695  
## [5,] 4 -1.020 0.695  
## [6,] 5 -0.972 0.712  
## [7,] 6 -0.989 0.706  
## [8,] 7 -0.970 0.712  
## Type 3: with drift and trend   
## lag ADF p.value  
## [1,] 0 -2.42 0.400  
## [2,] 1 -2.24 0.476  
## [3,] 2 -2.17 0.504  
## [4,] 3 -2.12 0.527  
## [5,] 4 -2.09 0.539  
## [6,] 5 -1.93 0.606  
## [7,] 6 -1.85 0.641  
## [8,] 7 -1.88 0.626  
## ----   
## Note: in fact, p.value = 0.01 means p.value <= 0.01

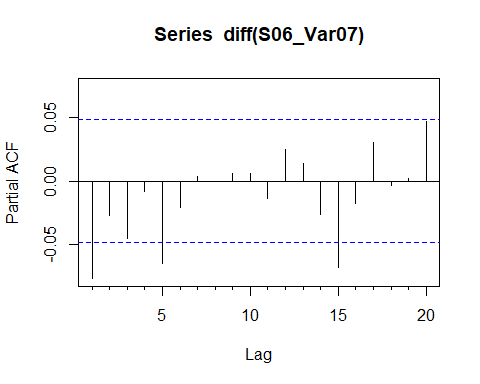
ndiffs(S06\_Var07) #Determines how many first differences are nessicary

## [1] 1

Acf(diff(S06\_Var07), lag.max = 10)



Pacf(diff(S06\_Var07), lag.max = 20)



needed outlier cleaning

non-stationary

needs 1 first difference

suggestive of autoregressive process

The ACF of the differenced series cut off after 1 lag, while the PACF trailed on for a while. Could be just one MA term, maybey 2

## Modeling

This was of great help: <https://www.datalytyx.com/choosing-the-right-forecast-model-for-time-series-data/>

Also this for helping determine the number of AR/MA terms: <https://people.duke.edu/~rnau/arimrule.htm>

We’ll try selecting ARIMA models based on our observations of the data, and by using the auto.arima() function to find one automatically.

When it comes to testing for correlations among residuals, we’ll use Box-Ljung tests. We’ll use ln(n) as the number of lags, where n is the number of residuals. There’s not much consensus out there on how many lags to use (pretty sure R automatically uses the frequency, but this isn’t great), but hopefully this’l do.

Note: because all datasets indicate 1 differencing is appropriate, d (the ‘I’ in arima) is always going to be 1. Could do ARMA models on the differenced timeseries, but ARIMA saves some code.

##### Some common observations:

Everything seems to require differencing 1 time.

No seasonal differencing was nessicariry; likely no seasonal trends

### S01 Var01

Both the ACF and PACF of the differenced series start positive, and become insignificant after lag 2. Could be two MA and two AR terms; maybe more AR terms because the PACF starts positive.

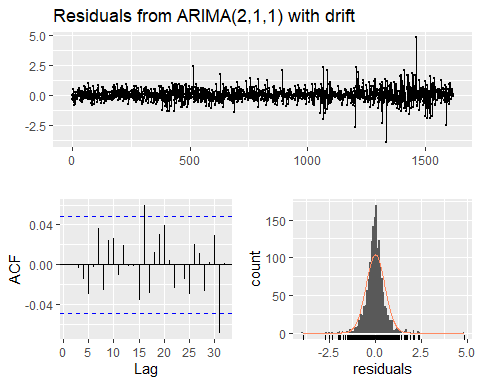
#### Manual

We’ll try out ARIMA 2, 1, 1. There is a trend, so drift should be included

#First a little function. Takes an arima model and does the analyses/residual-visualizations  
arima\_analysis <- function(fit) {  
 print(summary(fit))  
   
 checkresiduals(fit, lag = log(length(fit$residuals)))  
}

S01\_Var01.fit <- Arima(S01\_Var01, order = c(2, 1, 1), include.drift = TRUE)  
  
arima\_analysis(S01\_Var01.fit)

## Series: S01\_Var01   
## ARIMA(2,1,1) with drift   
##   
## Coefficients:  
## ar1 ar2 ma1 drift  
## -0.2298 -0.0500 0.3163 0.0221  
## s.e. 0.2829 0.0386 0.2829 0.0131  
##   
## sigma^2 estimated as 0.2617: log likelihood=-1210.15  
## AIC=2430.31 AICc=2430.34 BIC=2457.25  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 6.623852e-06 0.510797 0.347477 -0.01573551 0.9100188 0.9914794  
## ACF1  
## Training set -8.744703e-06



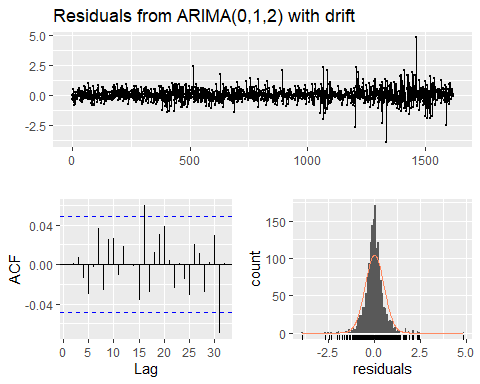
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1) with drift  
## Q\* = 3.9961, df = 3.3902, p-value = 0.3173  
##   
## Model df: 4. Total lags used: 7.39018142822643

The residuals have a mean of approximately 0 and are distributed normally. However they might not vary constantly, and the acf plot indicates some autocorrelation. The Ljung-Box test however failed to find evidence that the observed autocrorrelations did not come from white noise. This is a valid model

#### Auto ARIMA

S01\_Var01.autofit <- auto.arima(S01\_Var01)  
  
arima\_analysis(S01\_Var01.autofit)

## Series: S01\_Var01   
## ARIMA(0,1,2) with drift   
##   
## Coefficients:  
## ma1 ma2 drift  
## 0.0868 -0.0723 0.0221  
## s.e. 0.0248 0.0250 0.0129  
##   
## sigma^2 estimated as 0.2616: log likelihood=-1210.19  
## AIC=2428.38 AICc=2428.4 BIC=2449.94  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.928803e-05 0.5108083 0.3475428 -0.01592269 0.9100563 0.9916669  
## ACF1  
## Training set -0.000460003



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,2) with drift  
## Q\* = 4.0784, df = 4.3902, p-value = 0.4521  
##   
## Model df: 3. Total lags used: 7.39018142822643

the auto.arima function came up with ARIMA 0, 1, 2 with drift; much heavier on moving average than the manually selected model, with no autoregressive term.

The residuals have a mean of approximately 0 and are distributed normally. However they might not vary constantly, and the acf plot indicates some autocorrelation. The Ljung-Box test however failed to find evidence that the observed autocrorrelations did not come from white noise. This is a valid model.

Both models perform similarly, with the manual model barely edging out the automatic one with a log liklihood 0.04 higher, although the AICc for the auto model is slightly smaller. The RMSE for the manual model was also negligably higher. Either model works, but the reasoning behind the manual one makes more sense (should have autoregressive terms according to ACF/PACF of differenced series)

### S01 Var02

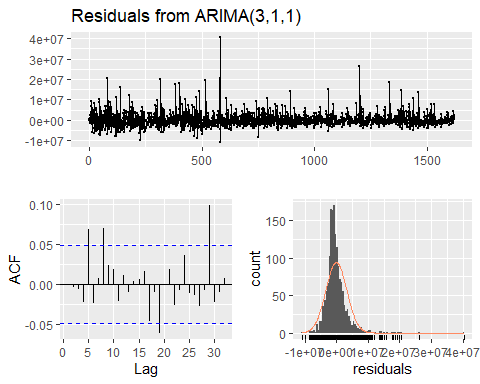
#### Manual

The ACF of the differenced series the ACF of the differenced series cuts off more quickly than the PACF, indicating primarily AR. The ACF cuts off past lag 2, and the PACF takes a while. Could try several AR terms and fewer MA terms

We’ll go with ARIMA 3,1,1 with no drift

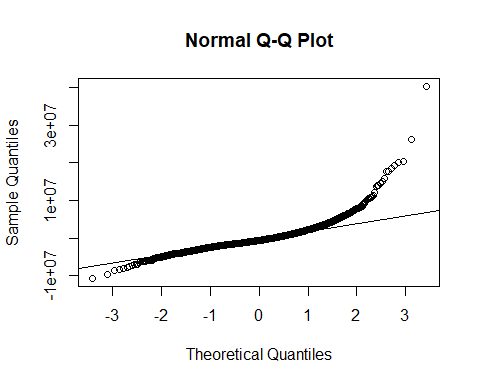
S01\_Var02.fit <- Arima(S01\_Var02, order = c(3, 1, 1), include.drift = FALSE)  
  
arima\_analysis(S01\_Var02.fit)

## Series: S01\_Var02   
## ARIMA(3,1,1)   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1  
## 0.3558 0.0763 0.0434 -0.9641  
## s.e. 0.0272 0.0273 0.0266 0.0107  
##   
## sigma^2 estimated as 1.102e+13: log likelihood=-26638.48  
## AIC=53286.95 AICc=53286.99 BIC=53313.91  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -78217.22 3313805 2191357 -12.08088 27.43747 0.8677296  
## ACF1  
## Training set -0.0008638219



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,1)  
## Q\* = 9.7636, df = 3.3914, p-value = 0.02858  
##   
## Model df: 4. Total lags used: 7.39141523467536

qqnorm(S01\_Var02.fit$residuals)  
qqline(S01\_Var02.fit$residuals)

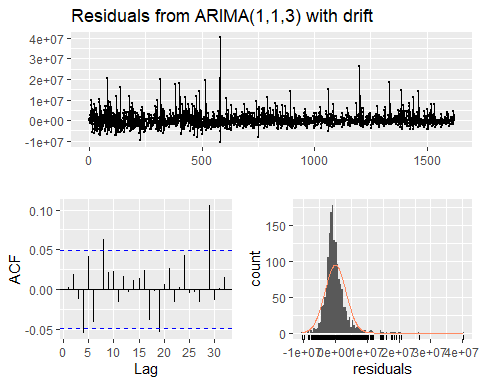


ARIMA 3, 1, 1 yielded some significantly correlated residuals, indicating this model isn’t a great fit. QQplot and the historgram indicate that the residuals are not normally distributed. Let’s see how an automatic fit does.

#### automatic

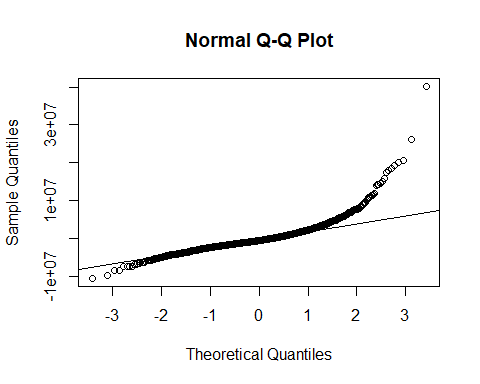
S01\_Var02.autofit <- auto.arima(S01\_Var02)  
  
arima\_analysis(S01\_Var02.autofit)

## Series: S01\_Var02   
## ARIMA(1,1,3) with drift   
##   
## Coefficients:  
## ar1 ma1 ma2 ma3 drift  
## 0.8216 -1.4374 0.3437 0.1010 -6469.450  
## s.e. 0.0717 0.0790 0.0540 0.0436 3543.399  
##   
## sigma^2 estimated as 1.099e+13: log likelihood=-26635.98  
## AIC=53283.96 AICc=53284.02 BIC=53316.31  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 16047.92 3308382 2163914 -10.68699 26.70408 0.8568628 0.00259467



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3) with drift  
## Q\* = 11.476, df = 2.3914, p-value = 0.005083  
##   
## Model df: 5. Total lags used: 7.39141523467536

qqnorm(S01\_Var02.autofit$residuals)  
qqline(S01\_Var02.autofit$residuals)

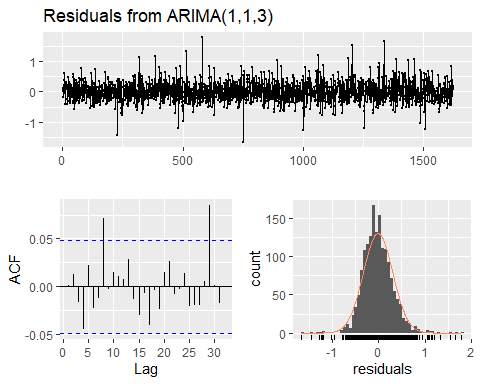


This doesn’t do well either. let’s try a log transform and do again

#### Log transform automatic

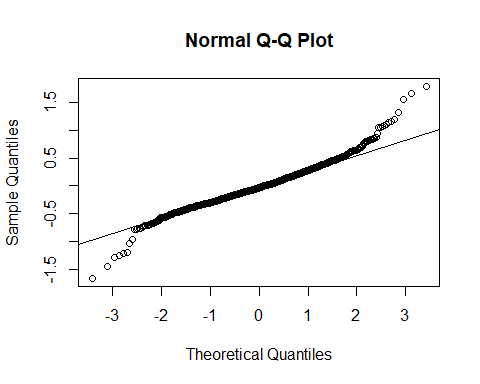
S01\_Var02.autofit <- auto.arima(log(S01\_Var02))  
  
arima\_analysis(S01\_Var02.autofit)

## Series: log(S01\_Var02)   
## ARIMA(1,1,3)   
##   
## Coefficients:  
## ar1 ma1 ma2 ma3  
## 0.8343 -1.3588 0.2603 0.1077  
## s.e. 0.0444 0.0532 0.0438 0.0368  
##   
## sigma^2 estimated as 0.1005: log likelihood=-436.35  
## AIC=882.7 AICc=882.73 BIC=909.65  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.01058189 0.3164588 0.2387578 -0.1051178 1.505845 0.8648944  
## ACF1  
## Training set 0.001403994



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 5.8017, df = 3.3914, p-value = 0.1552  
##   
## Model df: 4. Total lags used: 7.39141523467536

qqnorm(S01\_Var02.autofit$residuals)  
qqline(S01\_Var02.autofit$residuals)



Much better. Residuals here have a normal distribution with a mean of approximately 0. Box-Ljung test found insignificant correlation between residuals. QQ looks acceptable.

Use this one, but make sure to account for the log transform.

### S02 Var02

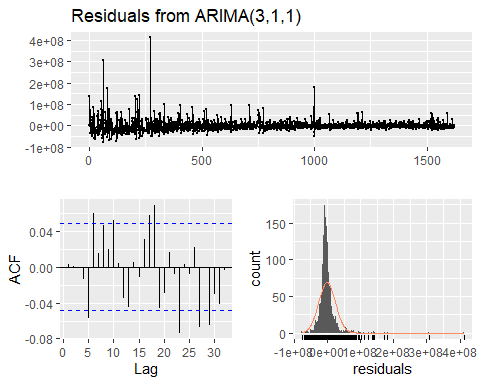
#### Manual

The ACF of the differenced series the ACF of the differenced series cuts off more quickly than the PACF, indicating primarily AR. The ACF cuts off past lag 2, and the PACF takes a while. Could try several AR terms and fewer MA terms

We’ll go with ARIMA 3,1,1 with no drift

S02\_Var02.fit <- Arima(S02\_Var02, order = c(3, 1, 1), include.drift = FALSE)  
  
arima\_analysis(S02\_Var02.fit)

## Series: S02\_Var02   
## ARIMA(3,1,1)   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1  
## 0.5126 -0.0264 0.0556 -0.9745  
## s.e. 0.0263 0.0284 0.0261 0.0081  
##   
## sigma^2 estimated as 6.364e+14: log likelihood=-29926.39  
## AIC=59862.79 AICc=59862.83 BIC=59889.74  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -969941.5 25188066 14202258 -12.8037 28.23752 0.9074204  
## ACF1  
## Training set 0.002722003



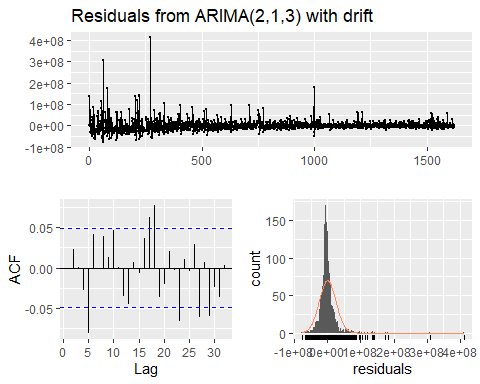
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,1)  
## Q\* = 11.881, df = 3.3914, p-value = 0.01113  
##   
## Model df: 4. Total lags used: 7.39141523467536

Ljung-Box finds the residuals to be correlated. Let’s see what automatic does

#### Automatic

S02\_Var02.autofit <- auto.arima(S02\_Var02)  
  
arima\_analysis(S02\_Var02.autofit)

## Series: S02\_Var02   
## ARIMA(2,1,3) with drift   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 ma3 drift  
## 0.9410 -0.0952 -1.4019 0.2375 0.1698 -52061.64  
## s.e. 1.9259 1.1067 1.9100 1.9357 0.0709 22970.61  
##   
## sigma^2 estimated as 6.352e+14: log likelihood=-29924.02  
## AIC=59862.04 AICc=59862.11 BIC=59899.78  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -106883.4 25148718 14038251 -10.04727 27.21884 0.8969416  
## ACF1  
## Training set -0.001583427



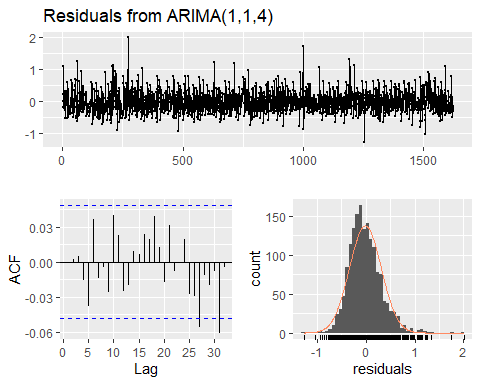
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,3) with drift  
## Q\* = 15.604, df = 1.3914, p-value = 0.000162  
##   
## Model df: 6. Total lags used: 7.39141523467536

Not gonna do it. Let’s try a log transform again.

#### Second manual with log transform

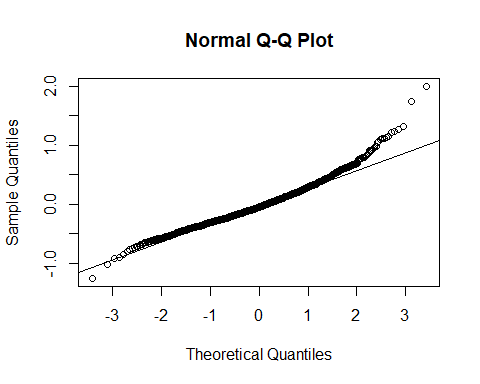
S02\_Var02.fit <- Arima(log(S02\_Var02), c(1,1,4))  
  
arima\_analysis(S02\_Var02.fit)

## Series: log(S02\_Var02)   
## ARIMA(1,1,4)   
##   
## Coefficients:  
## ar1 ma1 ma2 ma3 ma4  
## 0.8218 -1.2796 0.1459 0.1016 0.0422  
## s.e. 0.0718 0.0776 0.0500 0.0414 0.0351  
##   
## sigma^2 estimated as 0.1065: log likelihood=-483.26  
## AIC=978.51 AICc=978.57 BIC=1010.86  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.01418533 0.3257461 0.2512259 -0.1124231 1.42546 0.8812147  
## ACF1  
## Training set -0.0003465617



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,4)  
## Q\* = 5.3952, df = 2.3914, p-value = 0.0942  
##   
## Model df: 5. Total lags used: 7.39141523467536

qqnorm(S02\_Var02.fit$residuals)  
qqline(S02\_Var02.fit$residuals)



Auto arima here wasn’t cutting it, even with the log transform. ARIMA 1, 1, 4 on the log transformed data seems a good balance between maximizing log liklihood, and minimizing AICc. Use this one.

### S02 Var03

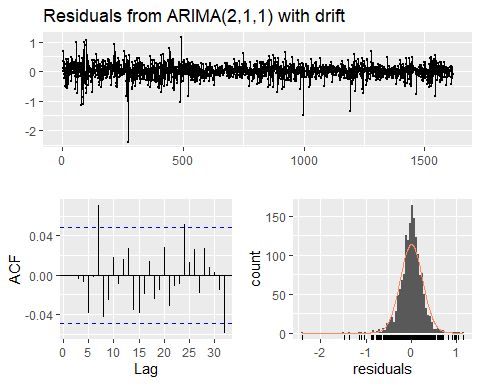
#### Manual

ACF of the differenced series cuts off from a negative value very quickly, indicating AR process. It takes a while for the PACF of the differenced series to drop off, so a higher order AR process. ACF is negative, so maybe an MA term would work well in addition.

We’ll go with ARIMA 2, 1, 1 with drift

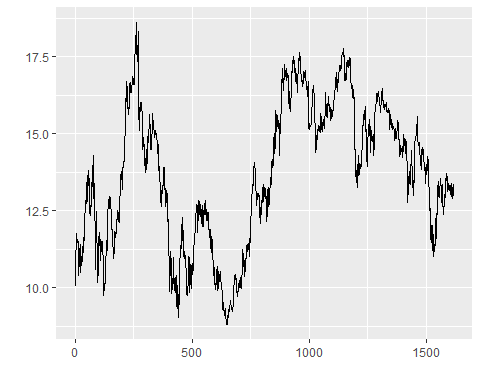
S02\_Var03.fit <- Arima(S02\_Var03, order = c(2, 1, 1), include.drift = TRUE)  
  
arima\_analysis(S02\_Var03.fit)

## Series: S02\_Var03   
## ARIMA(2,1,1) with drift   
##   
## Coefficients:  
## ar1 ar2 ma1 drift  
## 0.0850 -0.0450 0.0378 0.0017  
## s.e. 0.6021 0.0778 0.6028 0.0066  
##   
## sigma^2 estimated as 0.06035: log likelihood=-22.55  
## AIC=55.1 AICc=55.14 BIC=82.05  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 5.422401e-05 0.245292 0.1742698 -0.01404301 1.329092 0.9901759  
## ACF1  
## Training set -2.616005e-05



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1) with drift  
## Q\* = 10.626, df = 3.3889, p-value = 0.01947  
##   
## Model df: 4. Total lags used: 7.38894609761844

autoplot(tsclean(S02\_Var03))

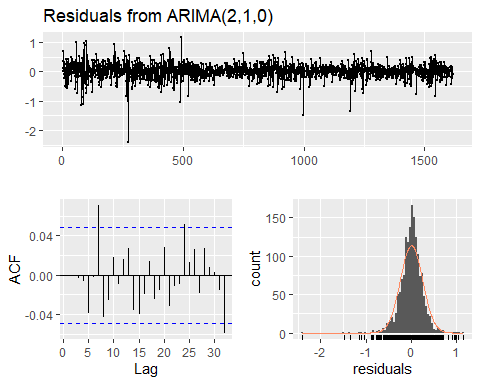


There is significant correlation between residuals. Let’s try automatic.

#### Automatic

S02\_Var03.autofit <- auto.arima(S02\_Var03)  
  
arima\_analysis(S02\_Var03.autofit)

## Series: S02\_Var03   
## ARIMA(2,1,0)   
##   
## Coefficients:  
## ar1 ar2  
## 0.1229 -0.0495  
## s.e. 0.0248 0.0249  
##   
## sigma^2 estimated as 0.06028: log likelihood=-22.58  
## AIC=51.16 AICc=51.18 BIC=67.33  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.001670473 0.2452963 0.1743671 -0.001892675 1.329659 0.990729  
## ACF1  
## Training set -8.655176e-05



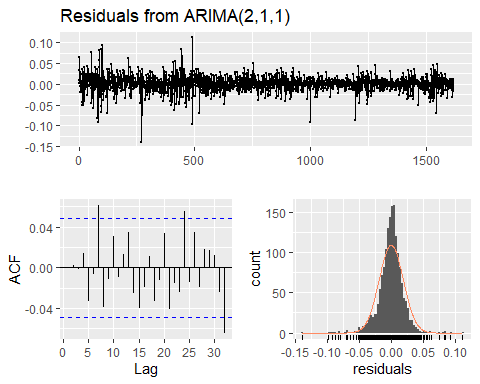
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,0)  
## Q\* = 10.626, df = 5.3889, p-value = 0.07377  
##   
## Model df: 2. Total lags used: 7.38894609761844

Slightly worse AICc and log likelihood, but technically insignificant correlation between residuals. That being said, it’s pretty close. Could try a log transformation.

#### automatic with log transformation

S02\_Var03.autofit <- auto.arima(log(S02\_Var03))  
  
arima\_analysis(S02\_Var03.autofit)

## Series: log(S02\_Var03)   
## ARIMA(2,1,1)   
##   
## Coefficients:  
## ar1 ar2 ma1  
## 0.7031 -0.1122 -0.5807  
## s.e. 0.2834 0.0348 0.2844  
##   
## sigma^2 estimated as 0.0003577: log likelihood=4123.22  
## AIC=-8238.44 AICc=-8238.42 BIC=-8216.89  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0001550128 0.01888909 0.01327902 0.003627398 0.5210391 0.9902776  
## ACF1  
## Training set -0.0001731139



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1)  
## Q\* = 8.2473, df = 4.3889, p-value = 0.1042  
##   
## Model df: 3. Total lags used: 7.38894609761844

Less significant correlation between residuals, but now the other statistics don’t compare with the previous models. Make of this what you will.

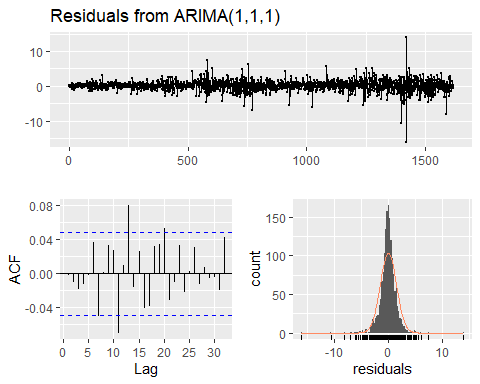
### S03 Var05

#### Manual

Both PACF and ACF of the differenced series cut off after 1 lag. Maybe one MA and AR term with no drift

S03\_Var05.fit <- Arima(S03\_Var05, order = c(1, 1, 1), include.drift = FALSE)  
  
arima\_analysis(S03\_Var05.fit)

## Series: S03\_Var05   
## ARIMA(1,1,1)   
##   
## Coefficients:  
## ar1 ma1  
## -0.3569 0.2007  
## s.e. 0.1534 0.1613  
##   
## sigma^2 estimated as 2.254: log likelihood=-2950.41  
## AIC=5906.82 AICc=5906.84 BIC=5922.99  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.04764078 1.499849 1.004693 0.06315507 1.326978 0.9903145  
## ACF1  
## Training set -0.002421695



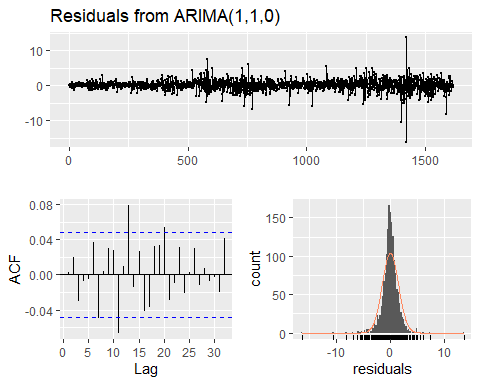
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1)  
## Q\* = 7.3013, df = 5.3889, p-value = 0.2344  
##   
## Model df: 2. Total lags used: 7.38894609761844

Fits well.

#### Automatic

S03\_Var05.autofit <- auto.arima(S03\_Var05)  
  
arima\_analysis(S03\_Var05.autofit)

## Series: S03\_Var05   
## ARIMA(1,1,0)   
##   
## Coefficients:  
## ar1  
## -0.1625  
## s.e. 0.0245  
##   
## sigma^2 estimated as 2.254: log likelihood=-2951.13  
## AIC=5906.26 AICc=5906.26 BIC=5917.03  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.04899978 1.500514 1.006303 0.06502231 1.329274 0.9919011  
## ACF1  
## Training set 0.003052547



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,0)  
## Q\* = 8.5118, df = 6.3889, p-value = 0.2353  
##   
## Model df: 1. Total lags used: 7.38894609761844

Similar ARIMA 1, 1, 1 does slighly better in terms of log liklihood but slightly worse in AIC. Go with ARIMA 1, 1, 1

### S03 Var07

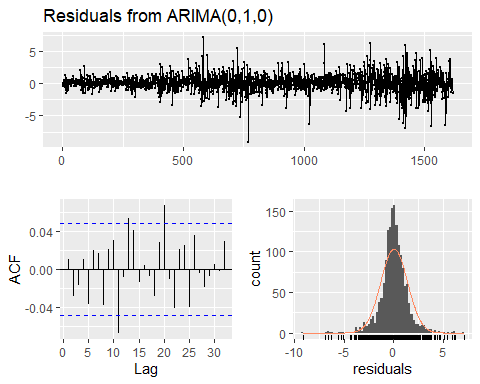
#### Manual

No significant lags in either ACF of PACF of the differenced series. This one has no AR or MA terms

#### Automatic

S03\_Var07.autofit <- auto.arima(S03\_Var07)  
  
arima\_analysis(S03\_Var07.autofit)

## Series: S03\_Var07   
## ARIMA(0,1,0)   
##   
## sigma^2 estimated as 1.815: log likelihood=-2776.46  
## AIC=5554.92 AICc=5554.92 BIC=5560.31  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.04128412 1.346888 0.9318668 0.05733297 1.228299 0.9994022  
## ACF1  
## Training set 0.01027221



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,0)  
## Q\* = 5.5096, df = 7.3889, p-value = 0.6404  
##   
## Model df: 0. Total lags used: 7.38894609761844

Automatic does the same thing. We will this one.

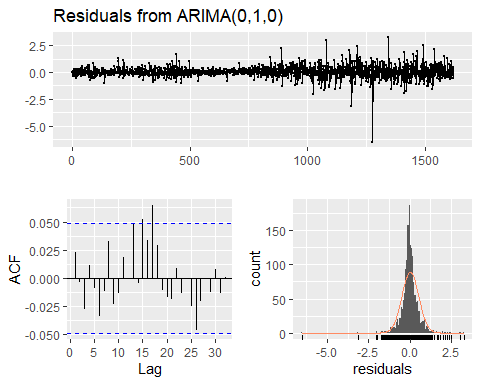
### S04 Var01

#### Manual

No significant lags in either ACF of PACF of the differenced series. This one has no AR or MA terms

S04\_Var01.fit <- Arima(S04\_Var01, order = c(0, 1, 0), include.drift = FALSE)  
  
arima\_analysis(S04\_Var01.fit)

## Series: S04\_Var01   
## ARIMA(0,1,0)   
##   
## sigma^2 estimated as 0.256: log likelihood=-1194.36  
## AIC=2390.73 AICc=2390.73 BIC=2396.12  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01217728 0.5058424 0.3238933 0.03165148 1.224553 0.9994155  
## ACF1  
## Training set 0.02310265

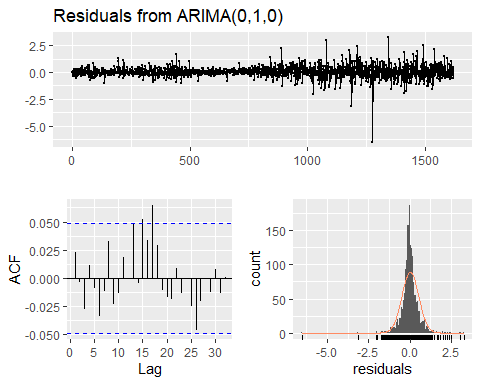


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,0)  
## Q\* = 4.6652, df = 7.3902, p-value = 0.7393  
##   
## Model df: 0. Total lags used: 7.39018142822643

#### Automatic

S04\_Var01.autofit <- auto.arima(S04\_Var01)  
  
arima\_analysis(S04\_Var01.autofit)

## Series: S04\_Var01   
## ARIMA(0,1,0)   
##   
## sigma^2 estimated as 0.256: log likelihood=-1194.36  
## AIC=2390.73 AICc=2390.73 BIC=2396.12  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01217728 0.5058424 0.3238933 0.03165148 1.224553 0.9994155  
## ACF1  
## Training set 0.02310265



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,0)  
## Q\* = 4.6652, df = 7.3902, p-value = 0.7393  
##   
## Model df: 0. Total lags used: 7.39018142822643

Same story: go with either model (they’re both the same)

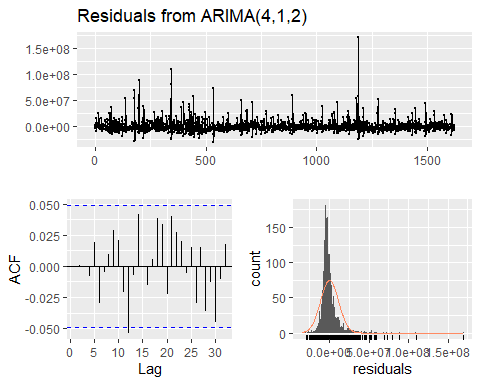
### S04 Var02

#### Manual

A high number of significant lags in the PACF of the differenced series, and a few significant lags in the ACF. Probably a higher number of AR terms, maybe a couple MA terms

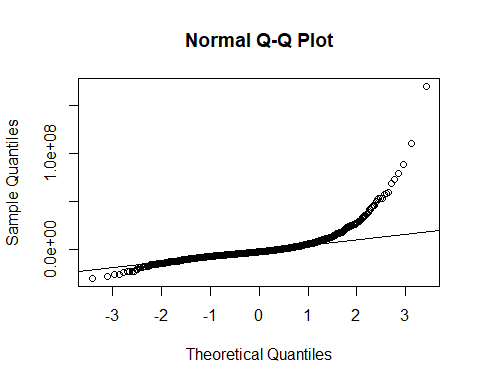
S04\_Var02.fit <- Arima(S04\_Var02, order = c(4, 1, 2), include.drift = FALSE)  
  
arima\_analysis(S04\_Var02.fit)

## Series: S04\_Var02   
## ARIMA(4,1,2)   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2  
## 0.8319 -0.1458 0.0081 0.0687 -1.3362 0.3452  
## s.e. 0.2534 0.1249 0.0323 0.0334 0.2539 0.2481  
##   
## sigma^2 estimated as 1.294e+14: log likelihood=-28634.14  
## AIC=57282.28 AICc=57282.35 BIC=57320.02  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -99214.73 11348712 6681186 -16.47112 33.52886 0.8910764  
## ACF1  
## Training set -0.0005397786



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,1,2)  
## Q\* = 2.209, df = 1.3914, p-value = 0.2094  
##   
## Model df: 6. Total lags used: 7.39141523467536

qqnorm(S04\_Var02.fit$residuals)  
qqline(S04\_Var02.fit$residuals)

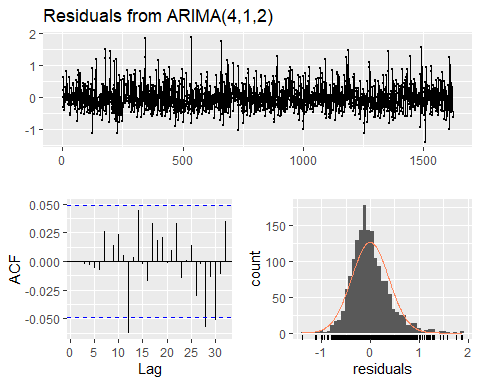


Insignificant correlation betwen residuals, but the residuals are skewed. Let’s try again with a log transform.

#### Manual with log transformation

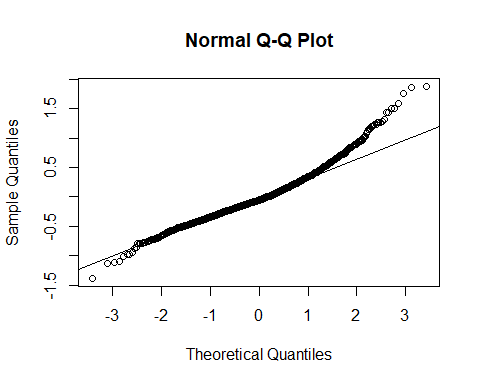
S04\_Var02.fit <- Arima(log(S04\_Var02), order = c(4, 1, 2), include.drift = FALSE)  
  
arima\_analysis(S04\_Var02.fit)

## Series: log(S04\_Var02)   
## ARIMA(4,1,2)   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2  
## 0.3902 0.0574 0.0664 0.0637 -0.8723 -0.0987  
## s.e. 0.4735 0.2334 0.0275 0.0467 0.4746 0.4593  
##   
## sigma^2 estimated as 0.1452: log likelihood=-734.13  
## AIC=1482.26 AICc=1482.33 BIC=1519.99  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.003432372 0.3802857 0.2862319 -0.06976069 1.704067 0.8726637  
## ACF1  
## Training set -0.0003328867



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,1,2)  
## Q\* = 1.3237, df = 1.3914, p-value = 0.3578  
##   
## Model df: 6. Total lags used: 7.39141523467536

qqnorm(S04\_Var02.fit$residuals)  
qqline(S04\_Var02.fit$residuals)

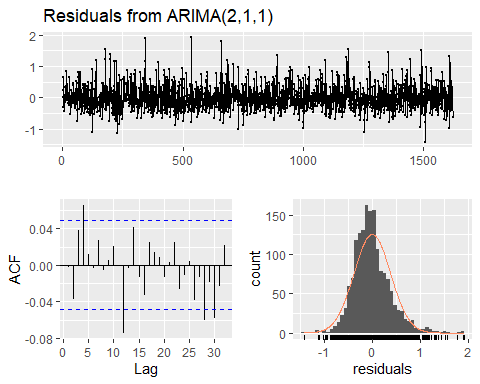


Definitely a better model. We’ll try an automatic fit with log transformation for comparison.

#### Automatic with log transformation

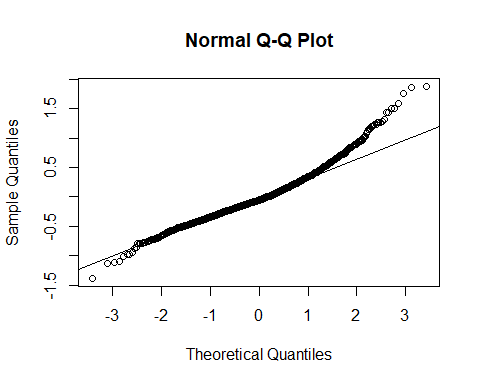
S04\_Var02.autofit <- auto.arima(log(S04\_Var02))  
  
arima\_analysis(S04\_Var02.autofit)

## Series: log(S04\_Var02)   
## ARIMA(2,1,1)   
##   
## Coefficients:  
## ar1 ar2 ma1  
## 0.4769 0.0320 -0.9522  
## s.e. 0.0286 0.0277 0.0140  
##   
## sigma^2 estimated as 0.1463: log likelihood=-741.64  
## AIC=1491.28 AICc=1491.3 BIC=1512.84  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.002523891 0.382075 0.2863235 -0.06405739 1.704565 0.8729428  
## ACF1  
## Training set -0.002713789



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1)  
## Q\* = 13.149, df = 4.3914, p-value = 0.01432  
##   
## Model df: 3. Total lags used: 7.39141523467536

qqnorm(S04\_Var02.fit$residuals)  
qqline(S04\_Var02.fit$residuals)



Automatic here has correlated residuals. The manual model (ARIMA 4, 1, 2 no drift) of the log transformed time series does better on in terms of log likelihood and AICc, along with having no significant correlations among residuals. Use the Manual ARIMA model with the log transformation

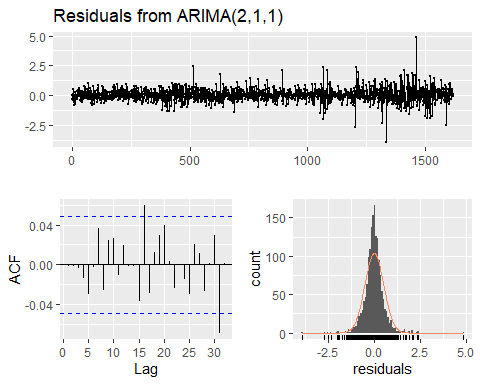
### S05 Var02

#### Manual

Similar ACF and PACF of differenced series. Maybe one of each term, or two AR because PACF starts positive

S05\_Var02.fit <- Arima(S05\_Var02, order = c(2, 1, 1), include.drift = FALSE)  
  
arima\_analysis(S05\_Var02.fit)

## Series: S05\_Var02   
## ARIMA(2,1,1)   
##   
## Coefficients:  
## ar1 ar2 ma1  
## -0.2399 -0.0471 0.3281  
## s.e. 0.2828 0.0391 0.2828  
##   
## sigma^2 estimated as 0.262: log likelihood=-1211.57  
## AIC=2431.15 AICc=2431.17 BIC=2452.7  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0213763 0.5112453 0.3476321 0.04288826 0.9097869 0.9919219  
## ACF1  
## Training set -0.00169794



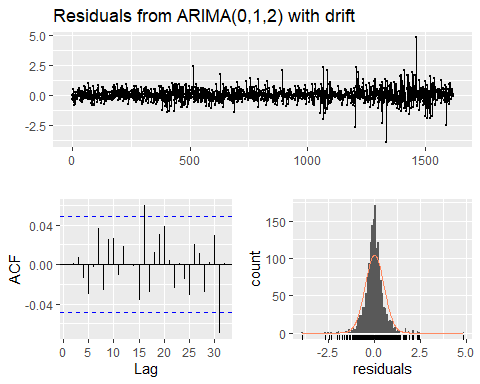
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1)  
## Q\* = 4.0143, df = 4.3902, p-value = 0.461  
##   
## Model df: 3. Total lags used: 7.39018142822643

Works. Looks good.

#### Automatic

S05\_Var02.autofit <- auto.arima(S05\_Var02)  
  
arima\_analysis(S05\_Var02.autofit)

## Series: S05\_Var02   
## ARIMA(0,1,2) with drift   
##   
## Coefficients:  
## ma1 ma2 drift  
## 0.0868 -0.0723 0.0221  
## s.e. 0.0248 0.0250 0.0129  
##   
## sigma^2 estimated as 0.2616: log likelihood=-1210.19  
## AIC=2428.38 AICc=2428.4 BIC=2449.94  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.928803e-05 0.5108083 0.3475428 -0.01592269 0.9100563 0.9916669  
## ACF1  
## Training set -0.000460003



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,2) with drift  
## Q\* = 4.0784, df = 4.3902, p-value = 0.4521  
##   
## Model df: 3. Total lags used: 7.39018142822643

This automatic model (ARIMA 0, 1, 2 with drift) works better than the manual one. Has Higher log likelihood and lower AICc. Use this one.

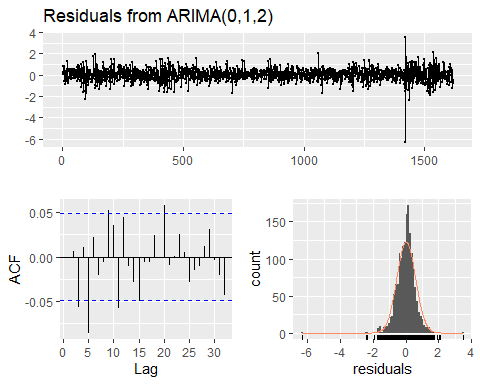
### S06 Var05

#### Manual

The ACF of the differenced series cut off after 1 lag, while the PACF trailed on for a while. Could be just one MA term, maybey 2

S06\_Var05.fit <- Arima(S06\_Var05, order = c(0, 1, 2), include.drift = FALSE)  
  
arima\_analysis(S06\_Var05.fit)

## Series: S06\_Var05   
## ARIMA(0,1,2)   
##   
## Coefficients:  
## ma1 ma2  
## -0.1319 -0.0064  
## s.e. 0.0249 0.0250  
##   
## sigma^2 estimated as 0.3208: log likelihood=-1373.3  
## AIC=2752.61 AICc=2752.62 BIC=2768.77  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01522326 0.5658444 0.414244 0.02734509 1.132739 0.9977172  
## ACF1  
## Training set -0.0004175731



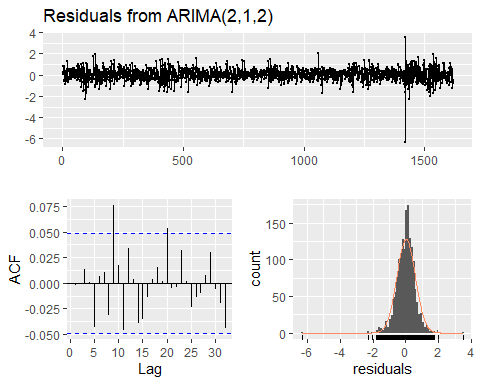
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,2)  
## Q\* = 18.856, df = 5.3883, p-value = 0.002792  
##   
## Model df: 2. Total lags used: 7.38832785957711

Has correlated residuals. No bueno.

#### Automatic

S06\_Var05.autofit <- auto.arima(S06\_Var05)  
  
arima\_analysis(S06\_Var05.autofit)

## Series: S06\_Var05   
## ARIMA(2,1,2)   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2  
## -0.2919 0.5441 0.1653 -0.5875  
## s.e. 0.1655 0.1235 0.1601 0.1082  
##   
## sigma^2 estimated as 0.3184: log likelihood=-1366.24  
## AIC=2742.48 AICc=2742.51 BIC=2769.42  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01715468 0.5633687 0.4139311 0.03150894 1.132408 0.9969635  
## ACF1  
## Training set -0.00202542



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2)  
## Q\* = 3.5921, df = 3.3883, p-value = 0.3691  
##   
## Model df: 4. Total lags used: 7.38832785957711

Better. No correlated residuals. Use this one.

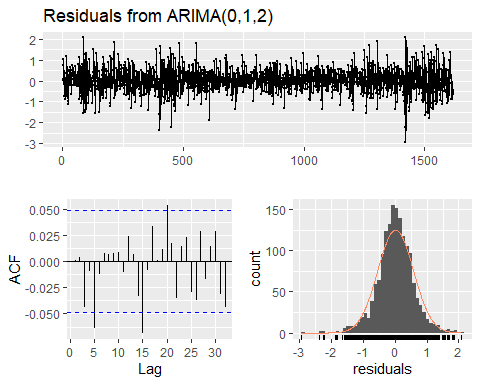
### S06 Var07

#### Manual

The ACF of the differenced series cut off after 1 lag, while the PACF trailed on for a while. Could be just one MA term, maybey 2

S06\_Var07.fit <- Arima(S06\_Var07, order = c(0, 1, 2), include.drift = FALSE)  
  
arima\_analysis(S06\_Var07.fit)

## Series: S06\_Var07   
## ARIMA(0,1,2)   
##   
## Coefficients:  
## ma1 ma2  
## -0.0814 -0.0287  
## s.e. 0.0249 0.0253  
##   
## sigma^2 estimated as 0.314: log likelihood=-1355.95  
## AIC=2717.89 AICc=2717.91 BIC=2734.06  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01441667 0.5598011 0.4172799 0.02501243 1.14156 0.9940068  
## ACF1  
## Training set 0.0007416662



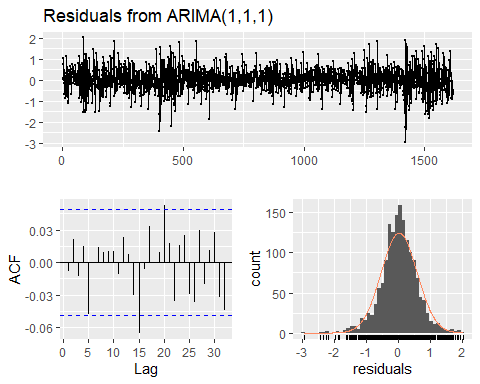
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,2)  
## Q\* = 10.359, df = 5.3883, p-value = 0.08132  
##   
## Model df: 2. Total lags used: 7.38832785957711

Works, but only just. Technically insignificant correlation between residuals, but..

#### Automatic

S06\_Var07.autofit <- auto.arima(S06\_Var07)  
  
arima\_analysis(S06\_Var07.autofit)

## Series: S06\_Var07   
## ARIMA(1,1,1)   
##   
## Coefficients:  
## ar1 ma1  
## 0.6515 -0.7267  
## s.e. 0.1287 0.1168  
##   
## sigma^2 estimated as 0.3131: log likelihood=-1353.71  
## AIC=2713.42 AICc=2713.44 BIC=2729.59  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0165486 0.5590255 0.4171379 0.02939178 1.141395 0.9936686  
## ACF1  
## Training set -0.008169026



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1)  
## Q\* = 5.4487, df = 5.3883, p-value = 0.4119  
##   
## Model df: 2. Total lags used: 7.38832785957711

Much better. Use this one.

## Model selection

Based on reasons described above for each model, the following is the summary table for the model selection.

df

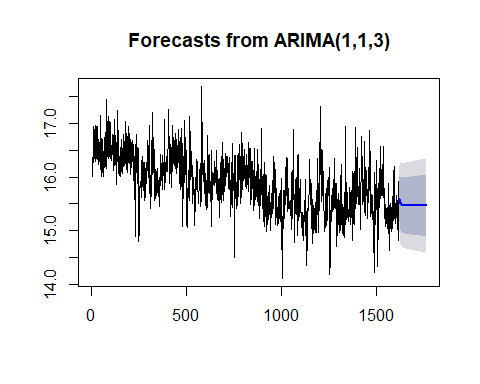
## group model\_selected arima\_type  
## 1 S01 S01\_Var02.autofit Log transform automatic  
## 2 S02 S02\_Var02.fit Manual with log transformation  
## 3 S03 S03\_Var05.fit Manual  
## 4 S04 S04\_Var02.fit Manual with log transformation  
## 5 S05 S05\_Var02.autofit Automatic  
## 6 S06 S06\_Var07.autofit Automatic

## Forecasting

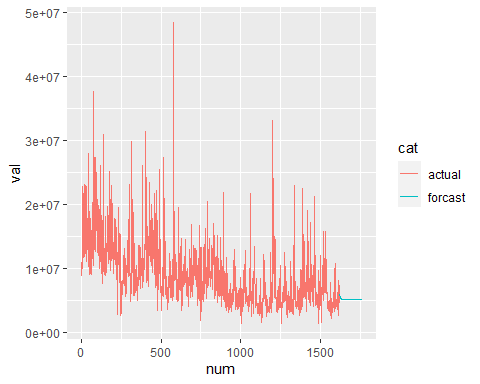
### Group S01

#### Var 02

# Standard Package  
plot(forecast::forecast(S01\_Var02.autofit, h=140))



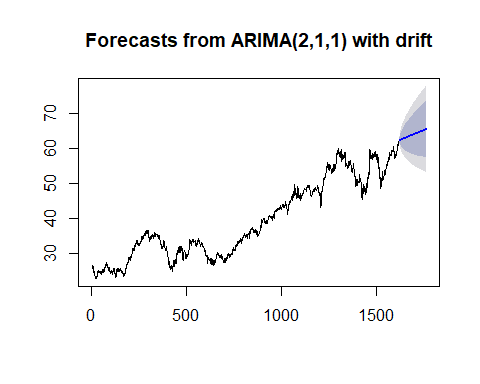
# GG Plot  
S01\_Var02.forcast <- forecast::forecast(S01\_Var02.autofit, h=140)   
  
S01\_Var02.forcast <- data.frame(S01\_Var02.forcast)  
  
S01\_Var02.forcast <- exp(S01\_Var02.forcast$Point.Forecast)  
  
  
S01\_Var02\_df.a <- data.frame(S01\_Var02,"actual")  
names(S01\_Var02\_df.a) <- c("val","cat")  
  
S01\_Var02\_df.b <- data.frame(as.numeric(S01\_Var02.forcast ),"forcast")  
names(S01\_Var02\_df.b) <- c("val","cat")  
  
  
S01\_Var02\_df <- rbind(S01\_Var02\_df.a,S01\_Var02\_df.b)  
  
S01\_Var02\_df$num <- as.numeric(row.names(S01\_Var02\_df))  
  
S01\_Var02\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()



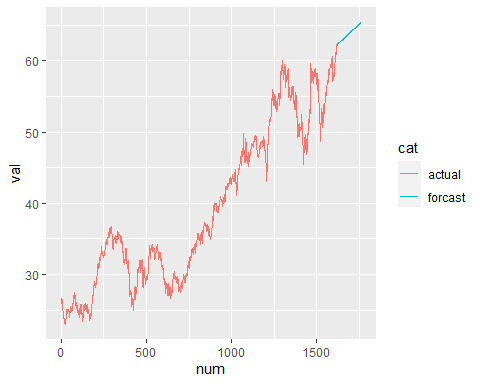
# Value List = S01\_Var02.forcast

#### Var 01

# Standard Package  
plot(forecast::forecast(S01\_Var01.fit, h=140))



# GG Plot  
S01\_Var01.forcast <- forecast::forecast(S01\_Var01.fit, h=140)   
  
S01\_Var01.forcast <- data.frame(S01\_Var01.forcast)  
  
S01\_Var01.forcast <- S01\_Var01.forcast$Point.Forecast  
  
  
S01\_Var01\_df.a <- data.frame(S01\_Var01,"actual")  
names(S01\_Var01\_df.a) <- c("val","cat")  
  
S01\_Var01\_df.b <- data.frame(as.numeric(S01\_Var01.forcast ),"forcast")  
names(S01\_Var01\_df.b) <- c("val","cat")  
  
  
S01\_Var01\_df <- rbind(S01\_Var01\_df.a,S01\_Var01\_df.b)  
  
S01\_Var01\_df$num <- as.numeric(row.names(S01\_Var01\_df))  
  
S01\_Var01\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()

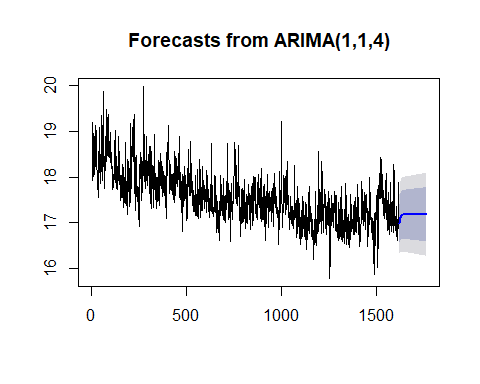


# Value List = S01\_Var01.forcast

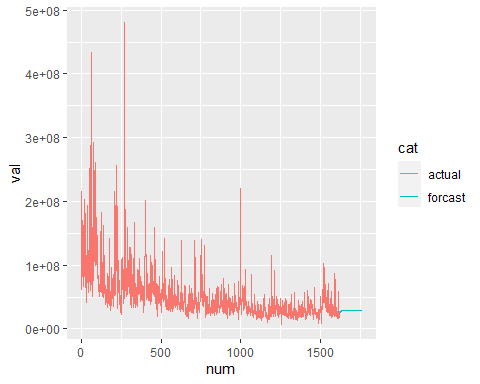
### Group S02

#### Var 02

# Standard Package  
plot(forecast::forecast(S02\_Var02.fit, h=140))



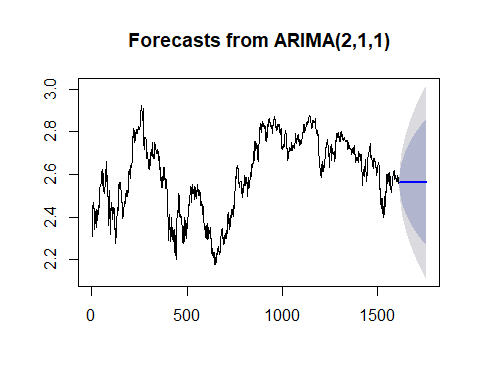
# GG Plot  
S02\_Var02.forcast <- forecast::forecast(S02\_Var02.fit, h=140)   
  
S02\_Var02.forcast <- data.frame(S02\_Var02.forcast)  
  
S02\_Var02.forcast <- exp(S02\_Var02.forcast$Point.Forecast)  
  
  
S02\_Var02\_df.a <- data.frame(S02\_Var02,"actual")  
names(S02\_Var02\_df.a) <- c("val","cat")  
  
S02\_Var02\_df.b <- data.frame(as.numeric(S02\_Var02.forcast ),"forcast")  
names(S02\_Var02\_df.b) <- c("val","cat")  
  
  
S02\_Var02\_df <- rbind(S02\_Var02\_df.a,S02\_Var02\_df.b)  
  
S02\_Var02\_df$num <- as.numeric(row.names(S02\_Var02\_df))  
  
S02\_Var02\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()



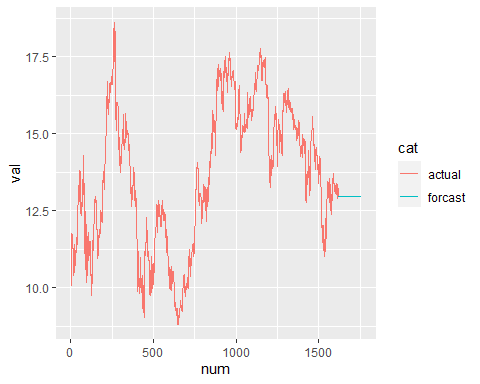
# Value List = S02\_Var02.forcast

#### Var 03

# Standard Package  
plot(forecast::forecast(S02\_Var03.autofit , h=140))



# GG Plot  
S02\_Var03.forcast <- forecast::forecast(S02\_Var03.autofit , h=140)   
  
S02\_Var03.forcast <- data.frame(S02\_Var03.forcast)  
  
S02\_Var03.forcast <- exp(S02\_Var03.forcast$Point.Forecast)  
  
  
S02\_Var03\_df.a <- data.frame(S02\_Var03,"actual")  
names(S02\_Var03\_df.a) <- c("val","cat")  
  
S02\_Var03\_df.b <- data.frame(as.numeric(S02\_Var03.forcast ),"forcast")  
names(S02\_Var03\_df.b) <- c("val","cat")  
  
  
S02\_Var03\_df <- rbind(S02\_Var03\_df.a,S02\_Var03\_df.b)  
  
S02\_Var03\_df$num <- as.numeric(row.names(S02\_Var03\_df))  
  
S02\_Var03\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()

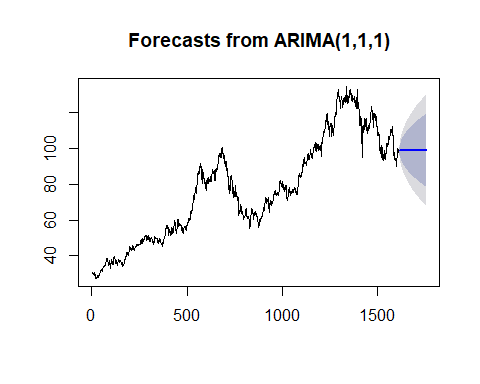


# Value List = S02\_Var03.forcast

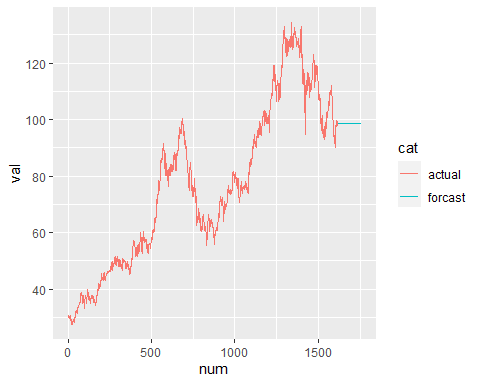
### Group S03

#### Var 05

# Standard Package   
plot(forecast::forecast(S03\_Var05.fit, h=140))



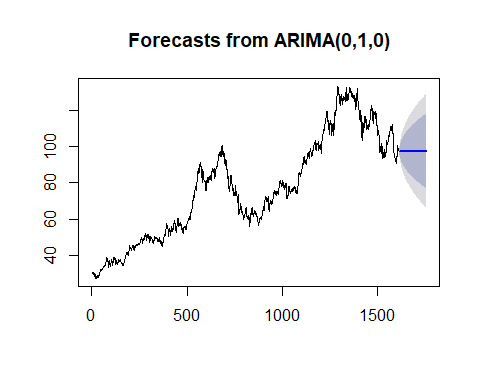
# GG Plot  
S03\_Var05.forcast <- forecast::forecast(S03\_Var05.fit, h=140)   
  
S03\_Var05.forcast <- data.frame(S03\_Var05.forcast)  
  
S03\_Var05.forcast <- S03\_Var05.forcast$Point.Forecast  
  
  
S03\_Var05\_df.a <- data.frame(S03\_Var05,"actual")  
names(S03\_Var05\_df.a) <- c("val","cat")  
  
S03\_Var05\_df.b <- data.frame(as.numeric(S03\_Var05.forcast ),"forcast")  
names(S03\_Var05\_df.b) <- c("val","cat")  
  
  
S03\_Var05\_df <- rbind(S03\_Var05\_df.a,S03\_Var05\_df.b)  
  
S03\_Var05\_df$num <- as.numeric(row.names(S03\_Var05\_df))  
  
S03\_Var05\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()



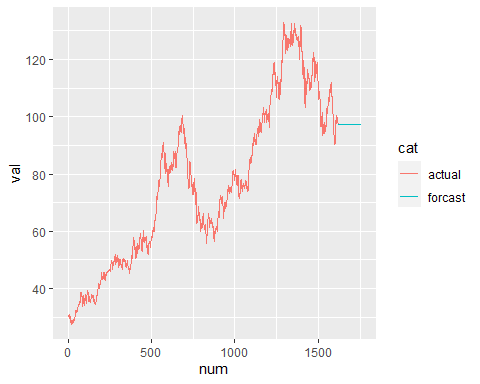
# Value List = S03\_Var05.forcast

#### Var 07

# Standard Package   
plot(forecast::forecast(S03\_Var07.fit, h=140))



# GG Plot  
S03\_Var07.forcast <- forecast::forecast(S03\_Var07.autofit, h=140)   
  
S03\_Var07.forcast <- data.frame(S03\_Var07.forcast)  
  
S03\_Var07.forcast <- S03\_Var07.forcast$Point.Forecast  
  
  
S03\_Var07\_df.a <- data.frame(S03\_Var07,"actual")  
names(S03\_Var07\_df.a) <- c("val","cat")  
  
S03\_Var07\_df.b <- data.frame(as.numeric(S03\_Var07.forcast ),"forcast")  
names(S03\_Var07\_df.b) <- c("val","cat")  
  
  
S03\_Var07\_df <- rbind(S03\_Var07\_df.a,S03\_Var07\_df.b)  
  
S03\_Var07\_df$num <- as.numeric(row.names(S03\_Var07\_df))  
  
S03\_Var07\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()

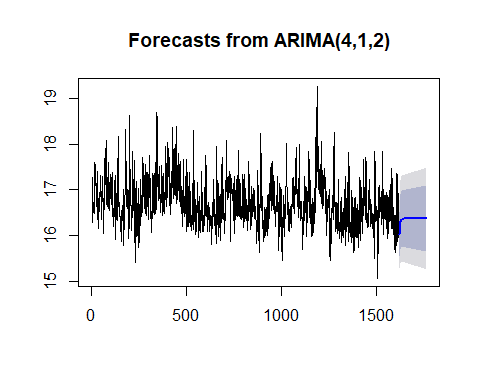


# Value List = S03\_Var07.forcast

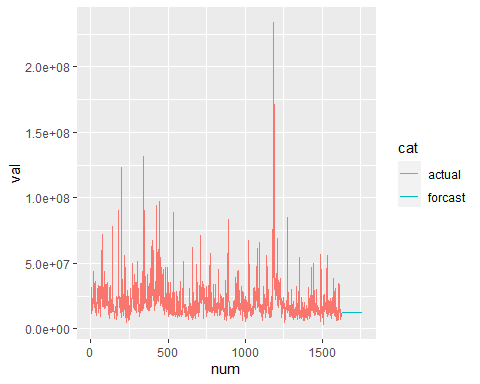
### Group S04

#### Var 02

# Standard Package   
plot(forecast::forecast(S04\_Var02.fit, h=140))



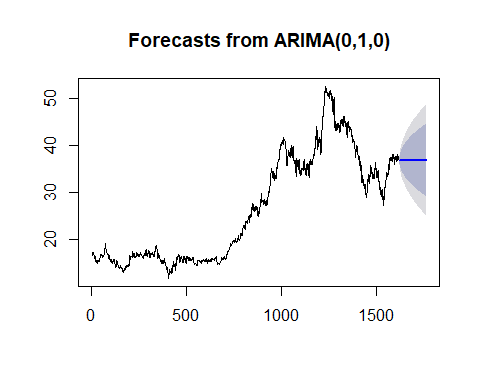
# GG Plot  
S04\_Var02.forcast <- forecast::forecast(S04\_Var02.fit, h=140)   
  
S04\_Var02.forcast <- data.frame(S04\_Var02.forcast)  
  
S04\_Var02.forcast <- exp(S04\_Var02.forcast$Point.Forecast)  
  
  
S04\_Var02\_df.a <- data.frame(S04\_Var02,"actual")  
names(S04\_Var02\_df.a) <- c("val","cat")  
  
S04\_Var02\_df.b <- data.frame(as.numeric(S04\_Var02.forcast ),"forcast")  
names(S04\_Var02\_df.b) <- c("val","cat")  
  
  
S04\_Var02\_df <- rbind(S04\_Var02\_df.a,S04\_Var02\_df.b)  
  
S04\_Var02\_df$num <- as.numeric(row.names(S04\_Var02\_df))  
  
S04\_Var02\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()



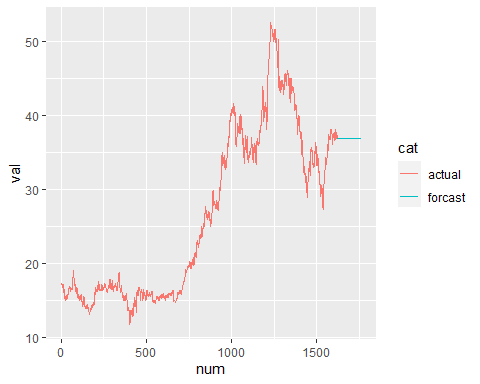
# Value List = S04\_Var02.forcast

#### Var 01

# Standard Package   
plot(forecast::forecast(S04\_Var01.fit, h=140))



# GG Plot  
S04\_Var01.forcast <- forecast::forecast(S04\_Var01.fit, h=140)   
  
S04\_Var01.forcast <- data.frame(S04\_Var01.forcast)  
  
S04\_Var01.forcast <- S04\_Var01.forcast$Point.Forecast  
  
  
S04\_Var01\_df.a <- data.frame(S04\_Var01,"actual")  
names(S04\_Var01\_df.a) <- c("val","cat")  
  
S04\_Var01\_df.b <- data.frame(as.numeric(S04\_Var01.forcast ),"forcast")  
names(S04\_Var01\_df.b) <- c("val","cat")  
  
  
S04\_Var01\_df <- rbind(S04\_Var01\_df.a,S04\_Var01\_df.b)  
  
S04\_Var01\_df$num <- as.numeric(row.names(S04\_Var01\_df))  
  
S04\_Var01\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()

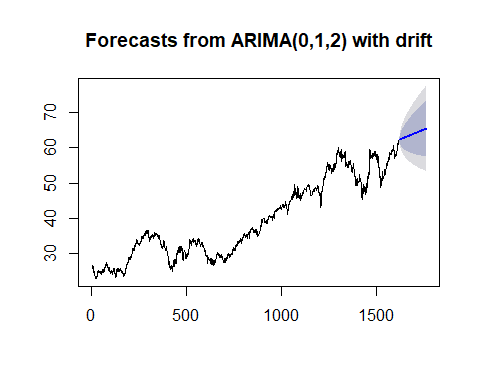


# Value List = S04\_Var01.forcast

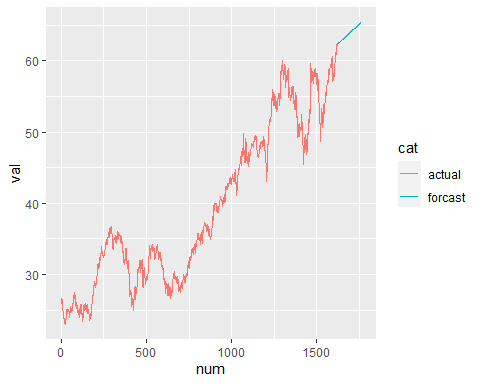
### Group S05

#### Var 02

# Standard Package   
plot(forecast::forecast(S05\_Var02.autofit, h=140))



# GG Plot  
S05\_Var02.forcast <- forecast::forecast(S05\_Var02.autofit, h=140)   
  
S05\_Var02.forcast <- data.frame(S05\_Var02.forcast)  
  
S05\_Var02.forcast <- S05\_Var02.forcast$Point.Forecast  
  
  
S05\_Var02\_df.a <- data.frame(S05\_Var02,"actual")  
names(S05\_Var02\_df.a) <- c("val","cat")  
  
S05\_Var02\_df.b <- data.frame(as.numeric(S05\_Var02.forcast ),"forcast")  
names(S05\_Var02\_df.b) <- c("val","cat")  
  
  
S05\_Var02\_df <- rbind(S05\_Var02\_df.a,S05\_Var02\_df.b)  
  
S05\_Var02\_df$num <- as.numeric(row.names(S05\_Var02\_df))  
  
S05\_Var02\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()

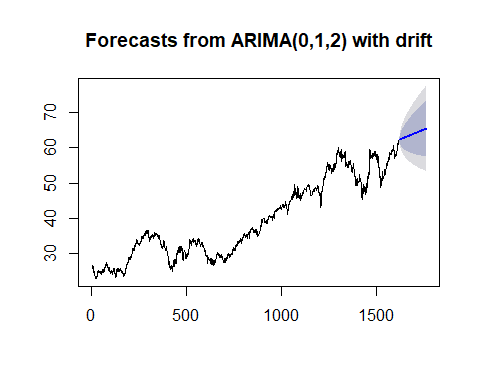


# Value List = S05\_Var02.forcast

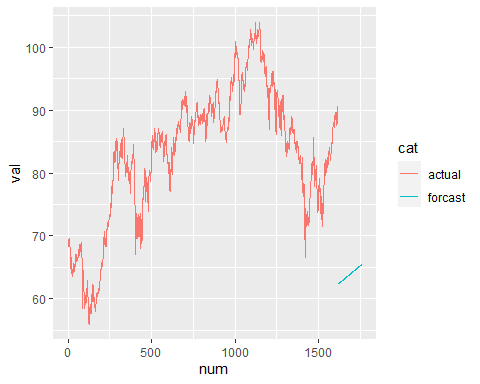
#### Var 03

Missing Model - used Var02 as a place holder

# Standard Package   
plot(forecast::forecast(S05\_Var02.autofit, h=140))



# GG Plot  
S05\_Var03.forcast <- forecast::forecast(S05\_Var02.autofit, h=140)   
  
S05\_Var03.forcast <- data.frame(S05\_Var03.forcast)  
  
S05\_Var03.forcast <- S05\_Var03.forcast$Point.Forecast  
  
  
S05\_Var03\_df.a <- data.frame(S05\_Var03,"actual")  
names(S05\_Var03\_df.a) <- c("val","cat")  
  
S05\_Var03\_df.b <- data.frame(as.numeric(S05\_Var03.forcast ),"forcast")  
names(S05\_Var03\_df.b) <- c("val","cat")  
  
  
S05\_Var03\_df <- rbind(S05\_Var03\_df.a,S05\_Var03\_df.b)  
  
S05\_Var03\_df$num <- as.numeric(row.names(S05\_Var03\_df))  
  
S05\_Var03\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()

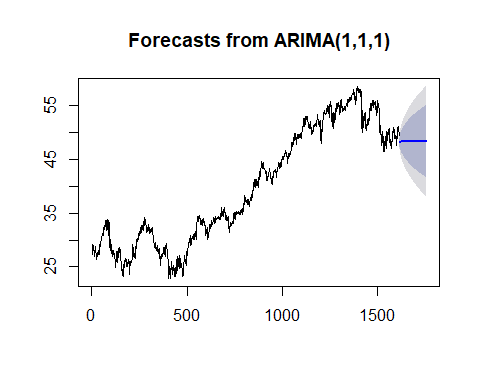


# Value List = S05\_Var03.forcast

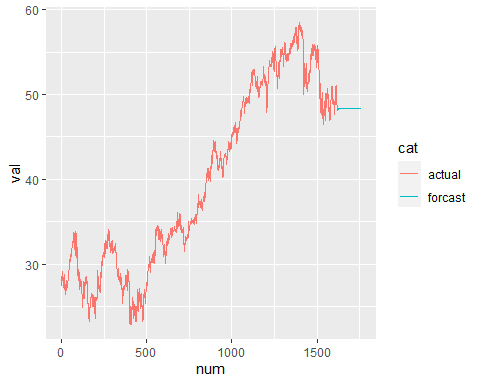
### Group S06

#### Var 07

# Standard Package   
plot(forecast::forecast(S06\_Var07.autofit, h=140))



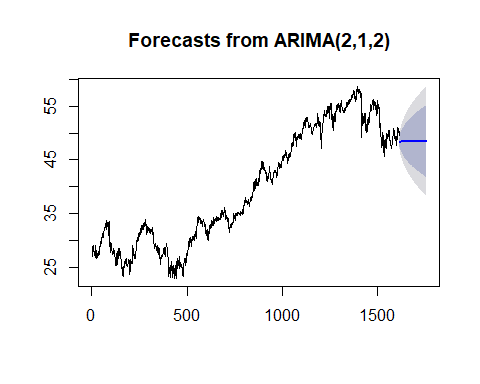
# GG Plot  
S06\_Var07.forcast <- forecast::forecast(S06\_Var07.autofit, h=140)   
  
S06\_Var07.forcast <- data.frame(S06\_Var07.forcast)  
  
S06\_Var07.forcast <- S06\_Var07.forcast$Point.Forecast  
  
  
S06\_Var07\_df.a <- data.frame(S06\_Var07,"actual")  
names(S06\_Var07\_df.a) <- c("val","cat")  
  
S06\_Var07\_df.b <- data.frame(as.numeric(S06\_Var07.forcast ),"forcast")  
names(S06\_Var07\_df.b) <- c("val","cat")  
  
  
S06\_Var07\_df <- rbind(S06\_Var07\_df.a,S06\_Var07\_df.b)  
  
S06\_Var07\_df$num <- as.numeric(row.names(S06\_Var07\_df))  
  
S06\_Var07\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()



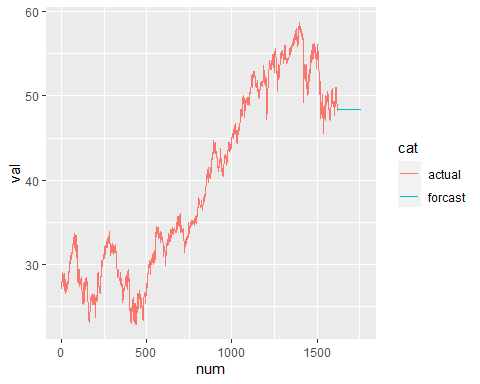
# Value List = S06\_Var07.forcast

#### Var 05

# Standard Package   
plot(forecast::forecast(S06\_Var05.autofit, h=140))



# GG Plot  
S06\_Var05.forcast <- forecast::forecast(S06\_Var05.autofit, h=140)   
  
S06\_Var05.forcast <- data.frame(S06\_Var05.forcast)  
  
S06\_Var05.forcast <- S06\_Var05.forcast$Point.Forecast  
  
  
S06\_Var05\_df.a <- data.frame(S06\_Var05,"actual")  
names(S06\_Var05\_df.a) <- c("val","cat")  
  
S06\_Var05\_df.b <- data.frame(as.numeric(S06\_Var05.forcast ),"forcast")  
names(S06\_Var05\_df.b) <- c("val","cat")  
  
  
S06\_Var05\_df <- rbind(S06\_Var05\_df.a,S06\_Var05\_df.b)  
  
S06\_Var05\_df$num <- as.numeric(row.names(S06\_Var05\_df))  
  
S06\_Var05\_df %>% ggplot(aes(x=num,y=val,col=cat)) + geom\_line()



# Value List = S06\_Var05.forcast

## Appendex

library(dplyr)  
library(forecast)  
library(readxl)  
library(skimr)  
library(tidyr)  
library(dplyr)  
library(seasonal)  
library(ggfortify)  
library(urca)  
library(forecast)  
library(aTSA)  
library(kableExtra)  
  
  
url<-"https://github.com/jnataky/Predictive\_Analytics/raw/main/Project1/Data\_Set\_for\_Class.xls"  
temp.file <- paste(tempfile(),".xls",sep = "")  
download.file(url, temp.file, mode = "wb")  
  
dataset <- read\_excel(temp.file, sheet = 1)  
  
str(dataset)

## tibble[,7] [10,572 x 7] (S3: tbl\_df/tbl/data.frame)  
## $ SeriesInd: num [1:10572] 40669 40669 40669 40669 40669 ...  
## $ group : chr [1:10572] "S03" "S02" "S01" "S06" ...  
## $ Var01 : num [1:10572] 30.6 10.3 26.6 27.5 69.3 ...  
## $ Var02 : num [1:10572] 1.23e+08 6.09e+07 1.04e+07 3.93e+07 2.78e+07 ...  
## $ Var03 : num [1:10572] 30.3 10.1 25.9 26.8 68.2 ...  
## $ Var05 : num [1:10572] 30.5 10.2 26.2 27 68.7 ...  
## $ Var07 : num [1:10572] 30.6 10.3 26 27.3 69.2 ...

df <- dataset %>%  
 mutate(SeriesInd = as.Date(SeriesInd, origin="1900-01-02"))  
head(df)

## # A tibble: 6 x 7  
## SeriesInd group Var01 Var02 Var03 Var05 Var07  
## <date> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2011-05-09 S03 30.6 123432400 30.3 30.5 30.6  
## 2 2011-05-09 S02 10.3 60855800 10.0 10.2 10.3  
## 3 2011-05-09 S01 26.6 10369300 25.9 26.2 26.0  
## 4 2011-05-09 S06 27.5 39335700 26.8 27.0 27.3  
## 5 2011-05-09 S05 69.3 27809100 68.2 68.7 69.2  
## 6 2011-05-09 S04 17.2 16587400 16.9 16.9 17.1

df %>%  
 group\_by(group) %>%  
 summarise(var1\_NAsum = sum(is.na(Var01)),  
 var2\_NAsum = sum(is.na(Var02)),  
 var3\_NAsum = sum(is.na(Var03)),  
 var5\_NAsum = sum(is.na(Var05)),  
 var7\_NAsum = sum(is.na(Var07)))

## # A tibble: 6 x 6  
## group var1\_NAsum var2\_NAsum var3\_NAsum var5\_NAsum var7\_NAsum  
## <chr> <int> <int> <int> <int> <int>  
## 1 S01 142 140 144 144 144  
## 2 S02 142 140 144 144 144  
## 3 S03 142 140 144 144 144  
## 4 S04 142 140 144 144 144  
## 5 S05 143 141 145 145 145  
## 6 S06 143 141 145 145 145

df <- df %>% gather("Var", "Value", 3:7) %>% drop\_na()