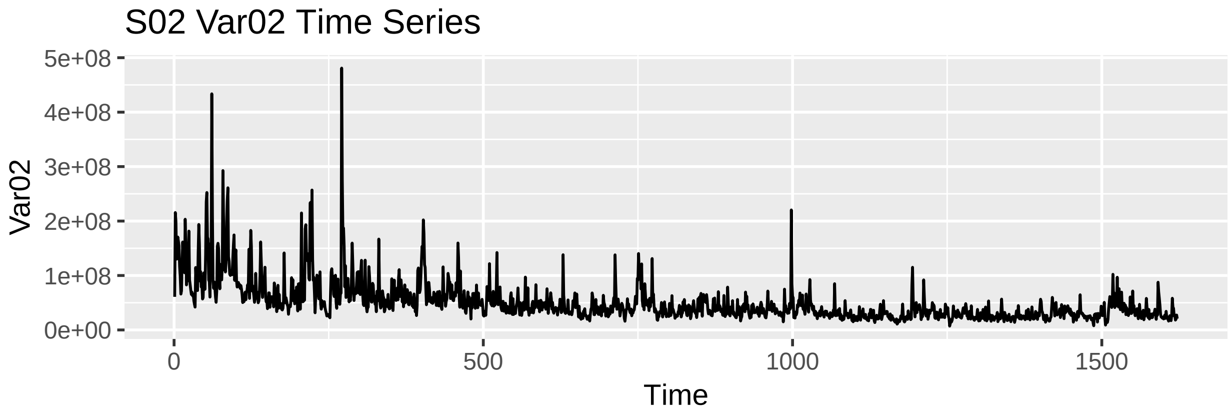
DATA624 Project #1

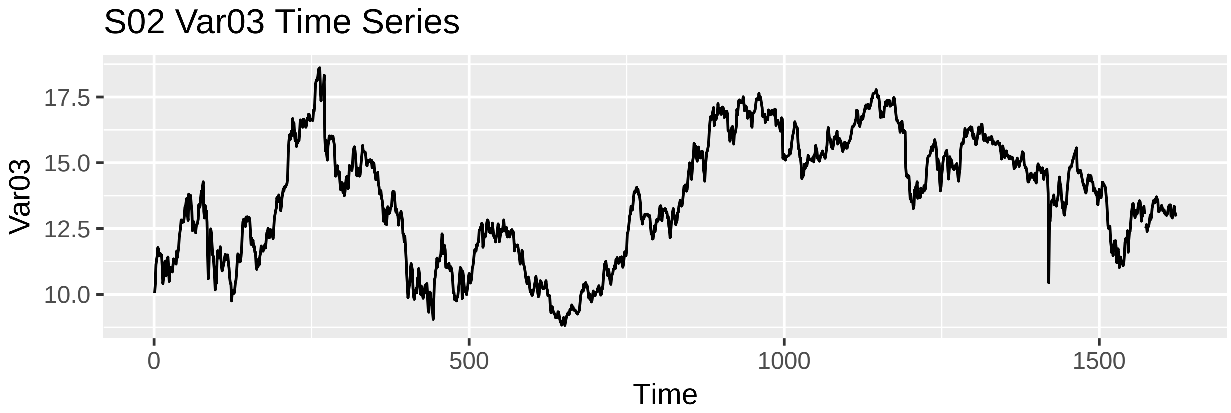
Jonathan Hernandez

- First let's acquire the data, extract the s02 group and the Var02 and Var03 features and convert them to a time series object for analysis.

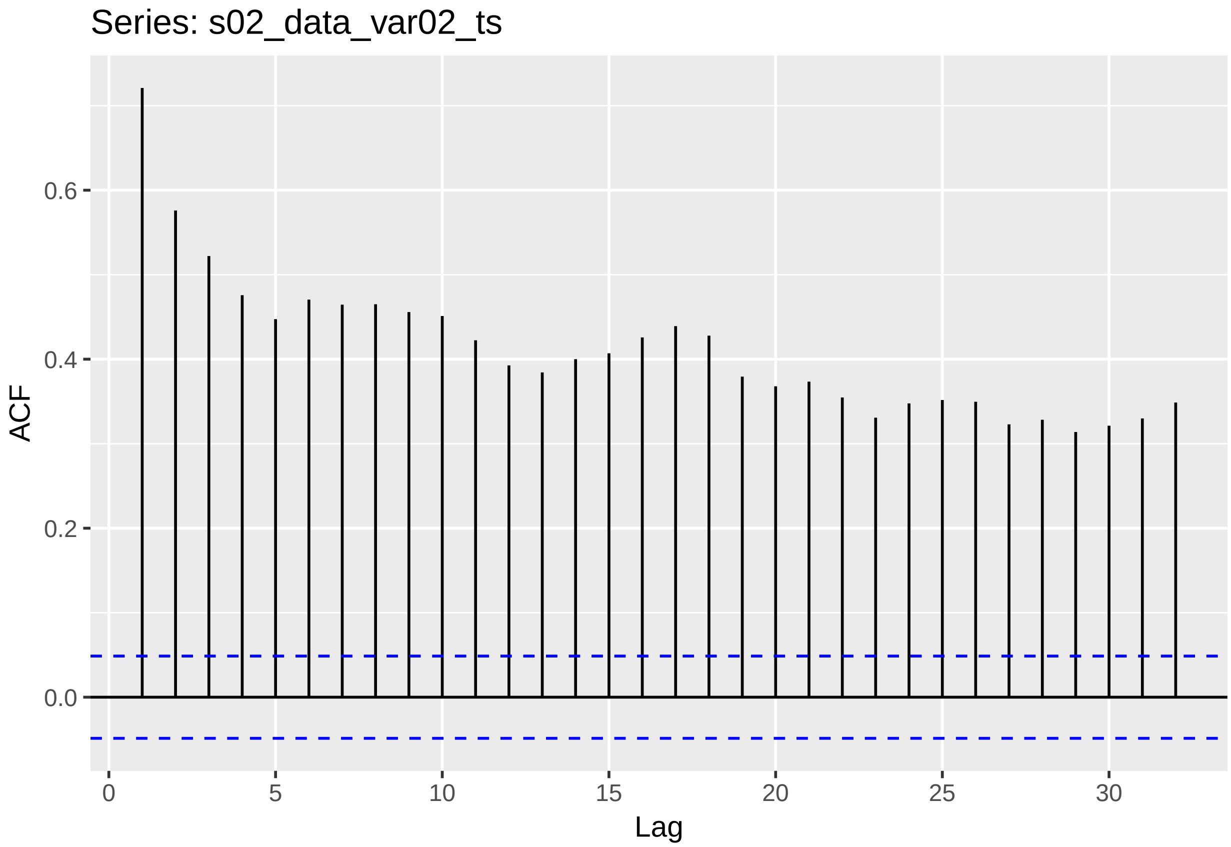
```
##    SeriesInd        group               Var02               Var03
##  Min.   :40669   Length:1622        Min.   :  7128800   Min.   : 8.82
##  1st Qu.:41253   Class :character   1st Qu.: 27880300   1st Qu.:11.82
##  Median :41846   Mode  :character   Median : 39767500   Median :13.76
##  Mean   :41843                      Mean   : 50633098   Mean   :13.68
##  3rd Qu.:42430                      3rd Qu.: 59050900   3rd Qu.:15.52
##  Max.   :43021                      Max.   :480879500   Max.   :38.28
##                                                         NA's   :4
```

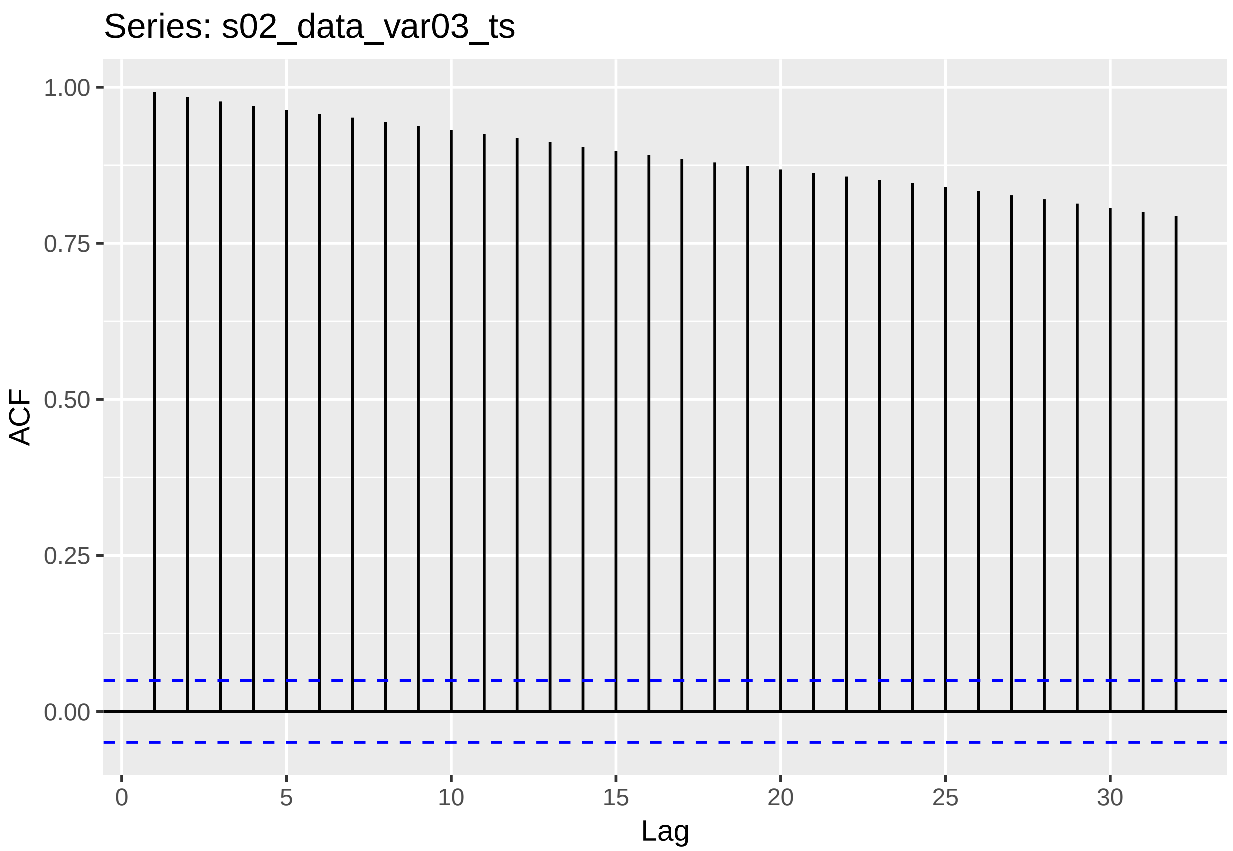
- Convert the variables to time series objects
- We see that there are several missing data in Var03. Let's use R's ImputeTS library and one of its functions na.interpolation and specify to replace the NA's using the spline option which uses polynomial interpretation.
- Let's remove the outlier from Var03 in the s02 group for better analysis
- With the data selected in question, let's look at the time series of Var02 and Var03 using autoplot(). This will show us of the behavior of the data over time or the series Index



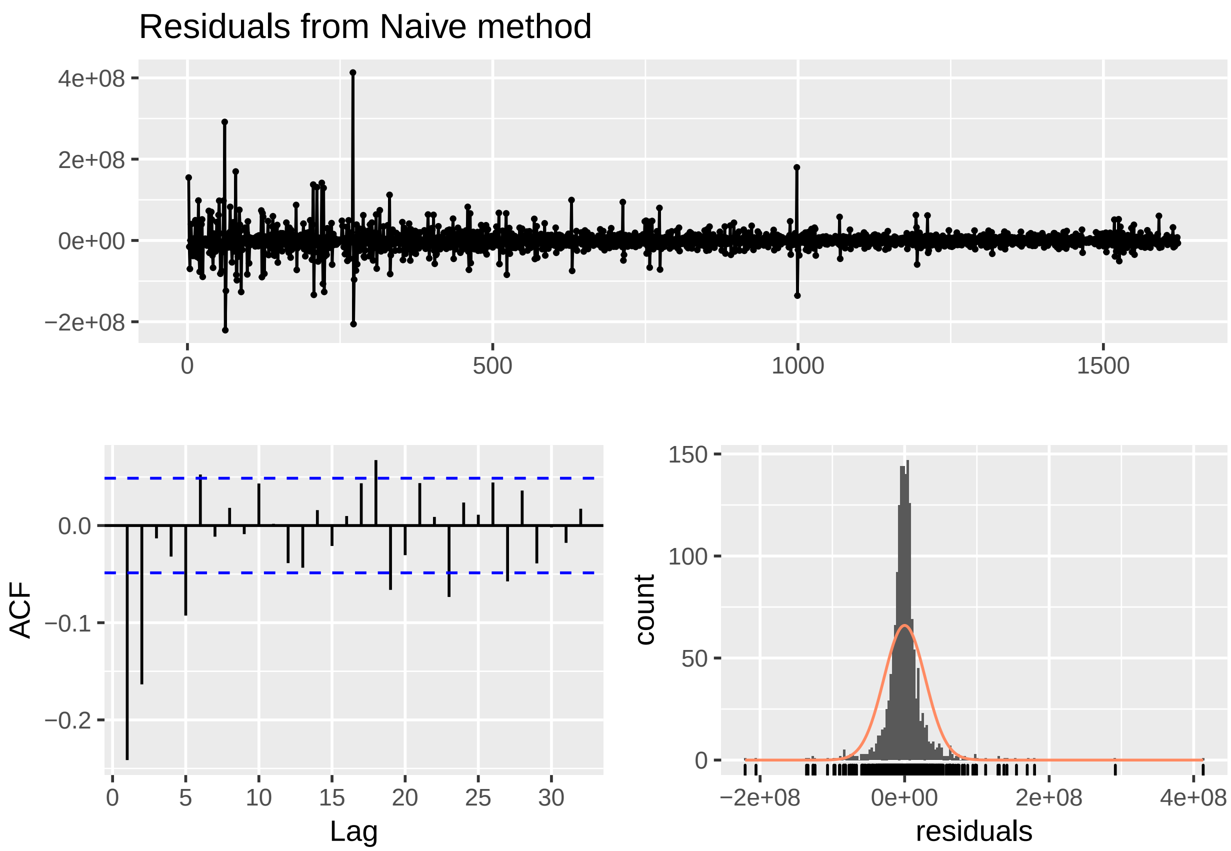


- By using the option “spline” in imputing data in our time series models, we see that it seems to do the best job in replacing NA values. Using splines estimates NA values using a polynomial interpolation. It helps us in the Var02 feature as it follows a downward trend.
- Making ACF plots of each time series:

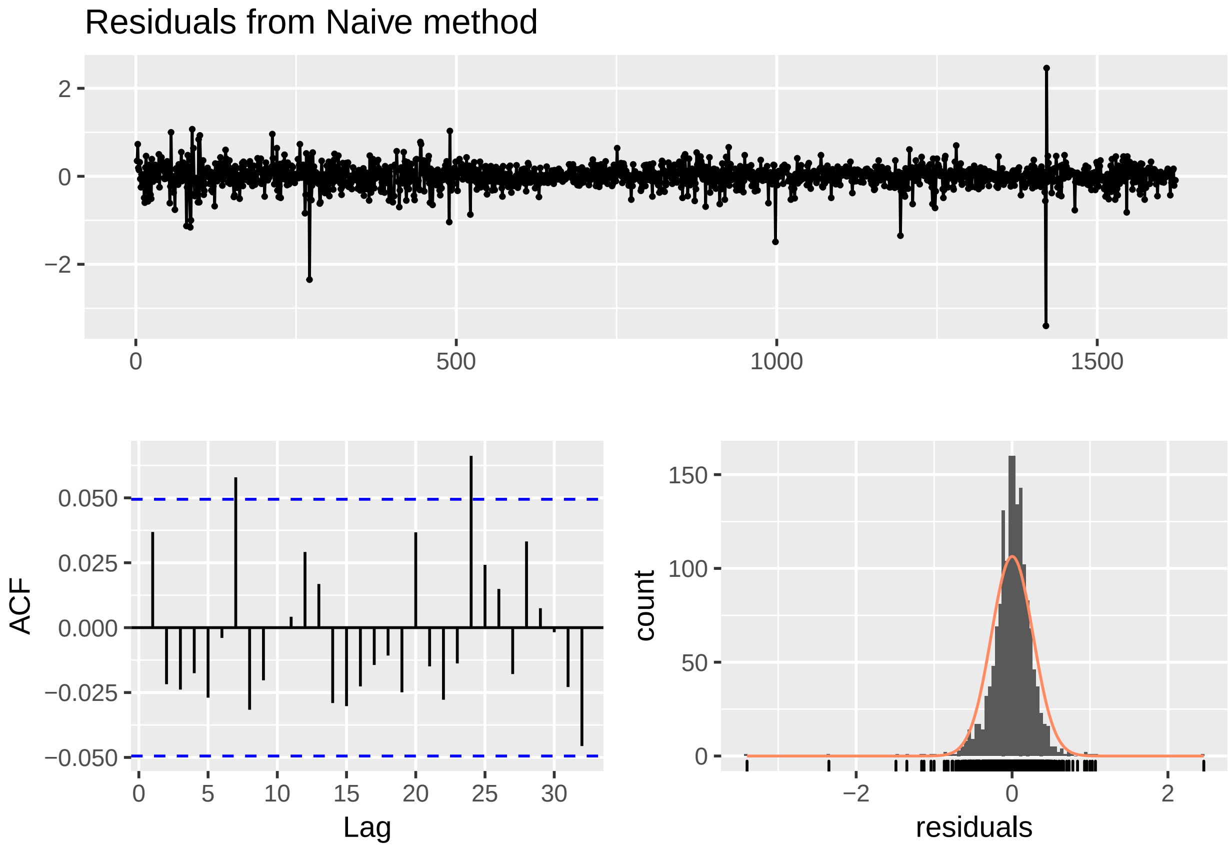




- Now with out data cleaned/imputed, let's do some forecasting using various techniques such as using ARIMA models, Naive forecasting, STL/ETS and Holt methods. I will use training/test sets preferably a 70/30 training/test set for each model. (training data will be from indexes 1 to $\lfloor 1622 * 0.7 \rfloor$, the test set the rest)
- Let's start with fitting a naive model for both time series.

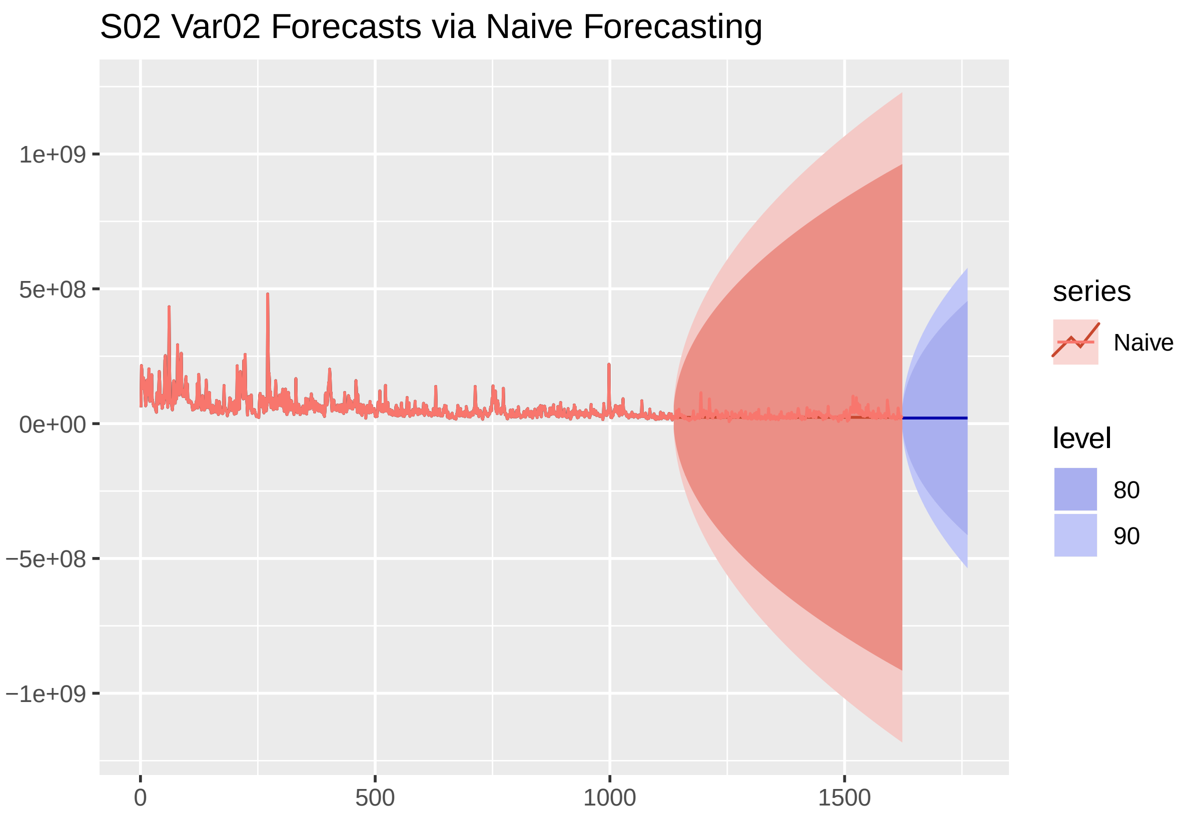


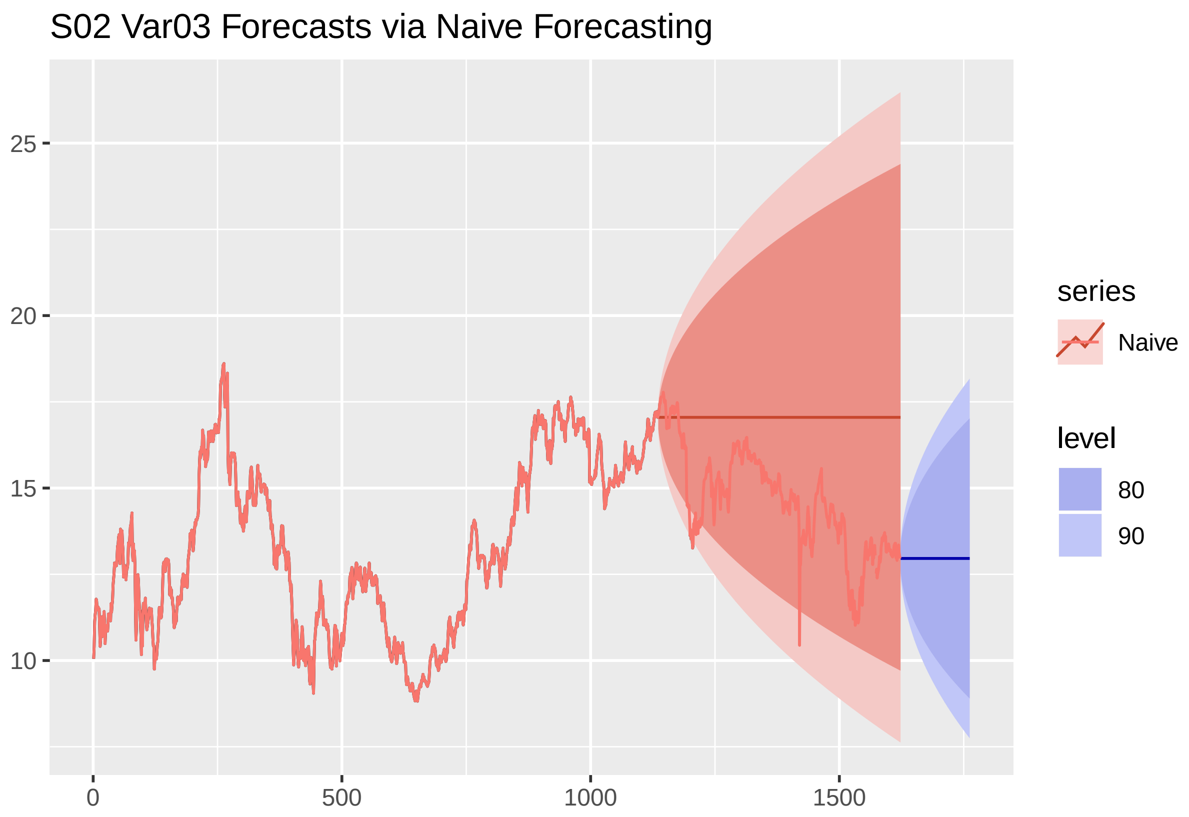
```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 162.36, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 14.24, df = 10, p-value = 0.1623
##
## Model df: 0.   Total lags used: 10
```

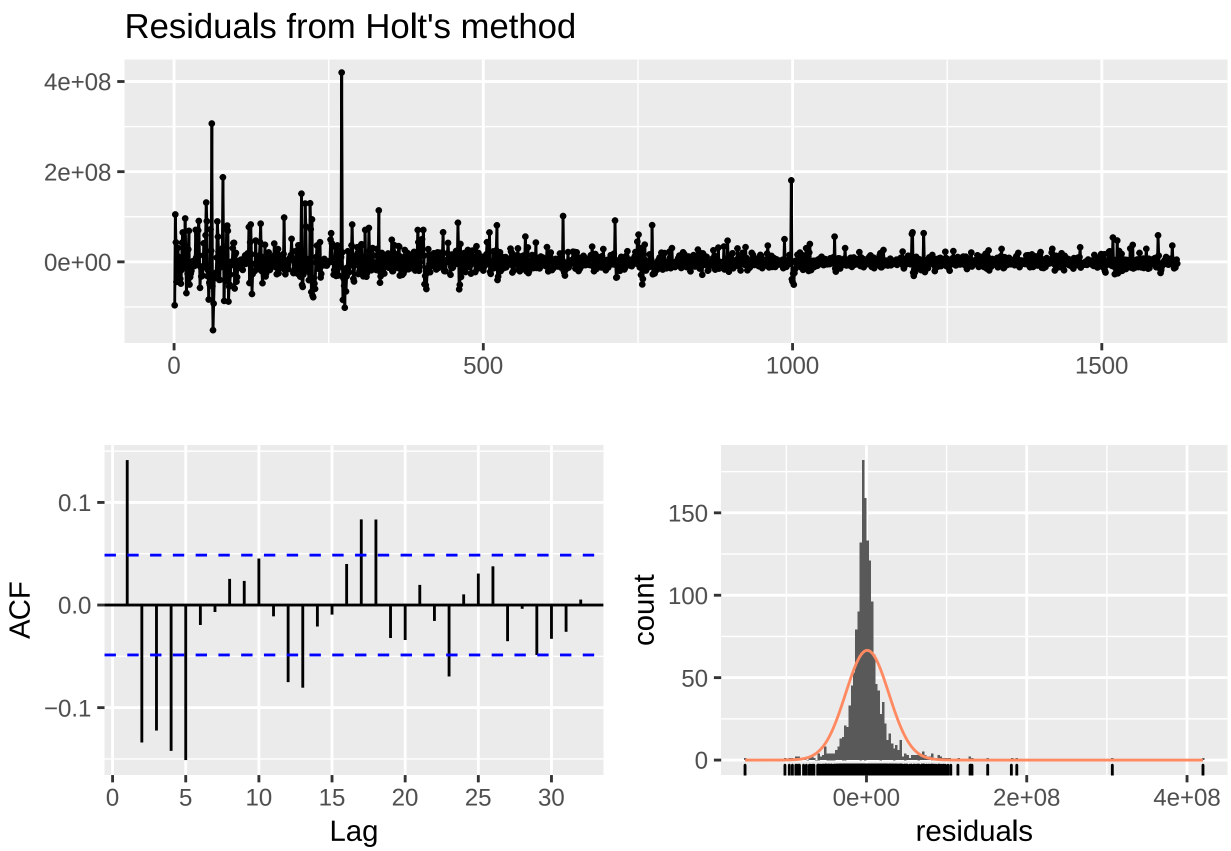
- Plotting the forecasts of both variables using the naive method


S02 Var02 Forecasts via Naive Forecasting
1e+09
5e+08
0e+00
−5e+08
−1e+09
0
500
1000
1500
series
Naive
level
80
90



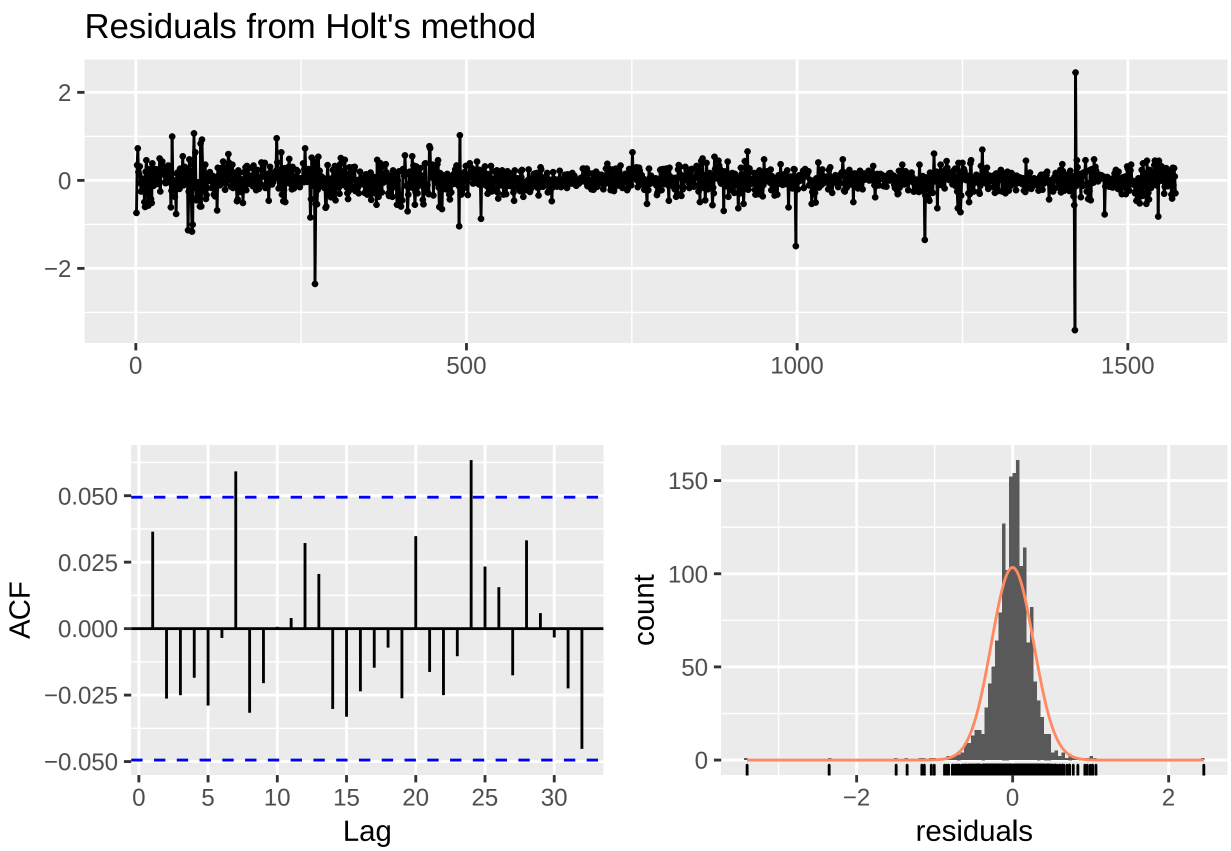
- Next is to use exponential smoothing and methods such as Holt's methond and Holt-Winters Seasonal Method. It seems that Var02 as mentioned earlier is following a downward trend and looks to have so seasonality so Holt's method may be useful. Var03 as what appears to be a seasonal trend.
- Using Holt's method

```
## Warning in ets(x, "AAN", alpha = alpha, beta = beta, phi = phi, damped =
## damped, : Missing values encountered. Using longest contiguous portion of
## time series
```



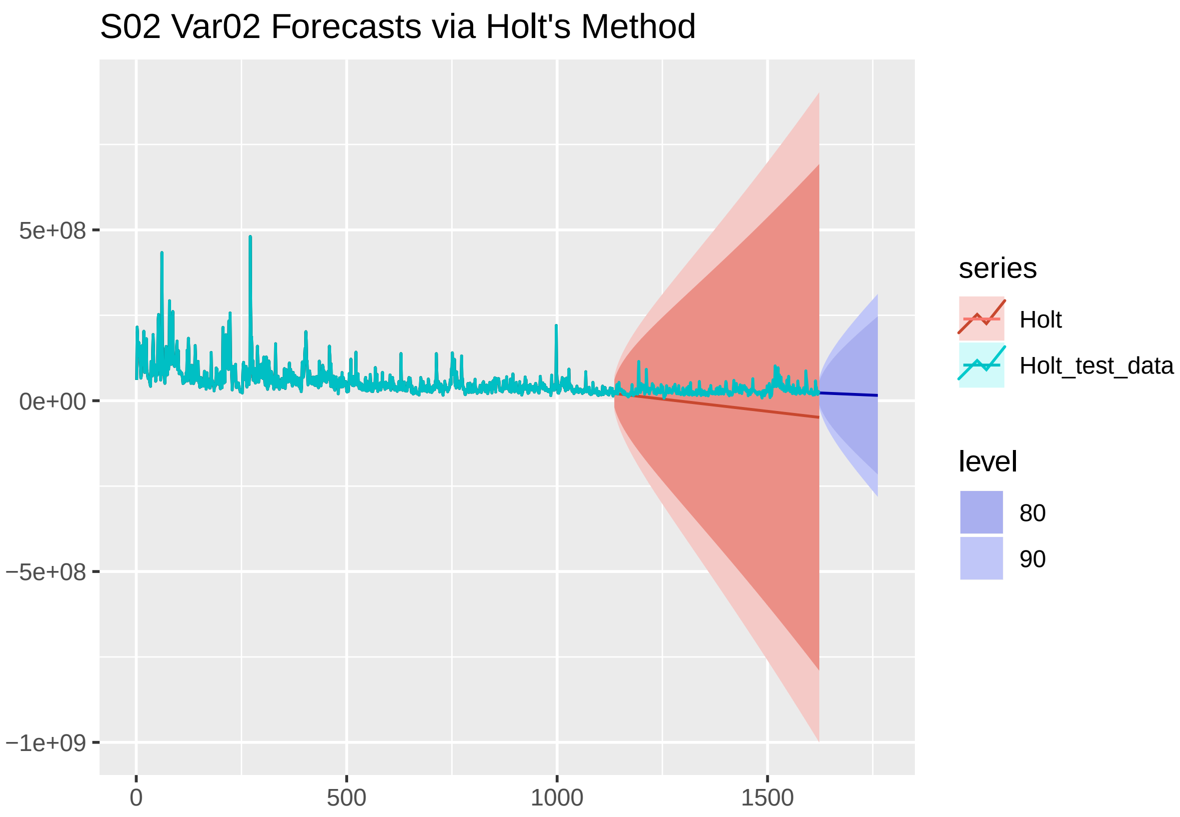
Residuals from Holt's method

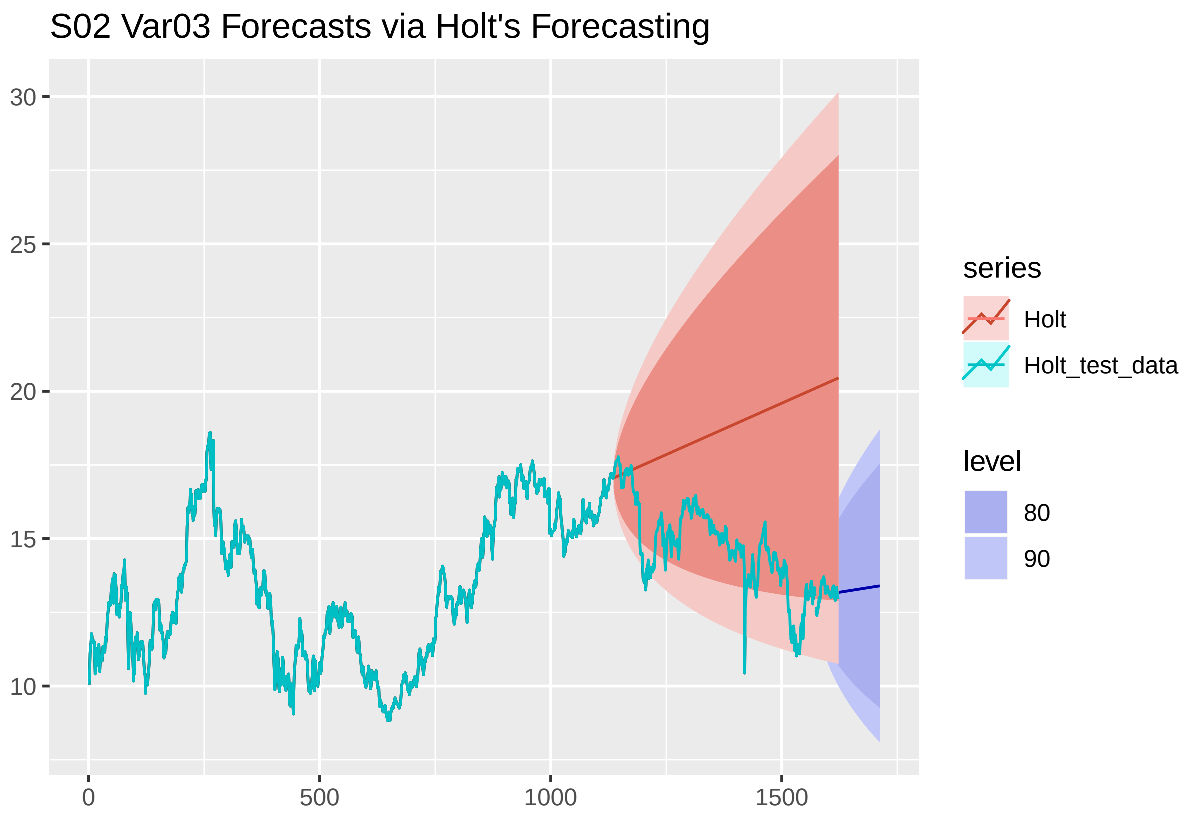
```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 162.04, df = 6, p-value < 2.2e-16
##
## Model df: 4.   Total lags used: 10
```



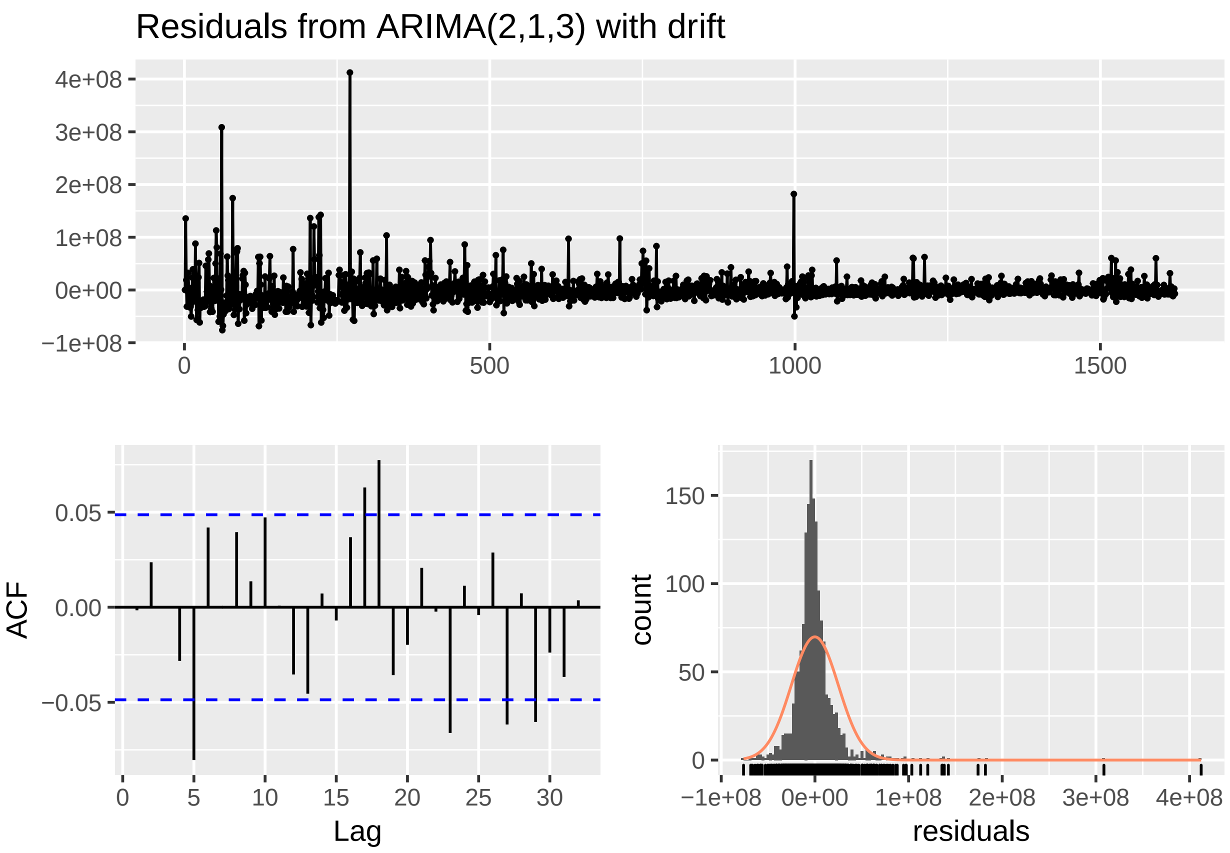
```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 13.838, df = 6, p-value = 0.0315
##
## Model df: 4.   Total lags used: 10
```

- Forecasts using Holt's method:

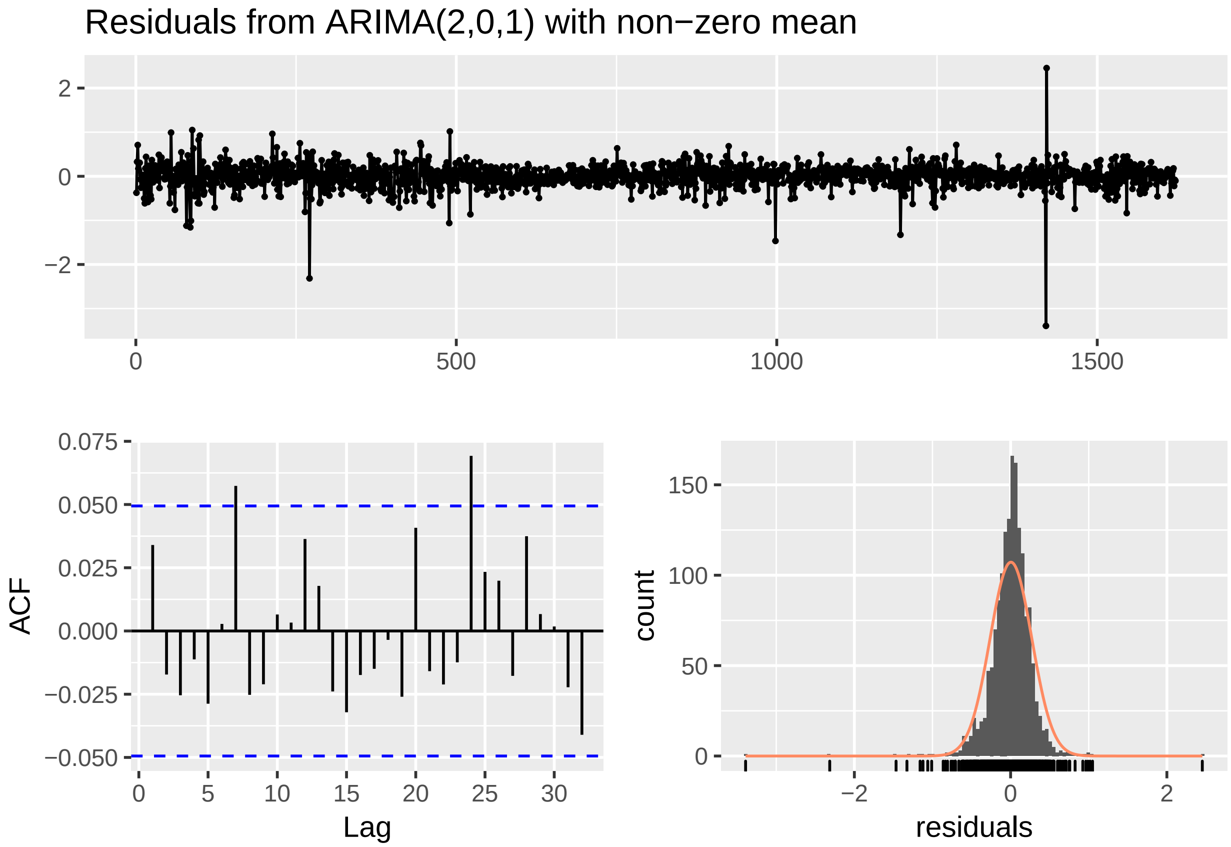

S02 Var02 Forecasts via Holt's Method
5e+08
0e+00
−5e+08
−1e+09
0
500
1000
1500
series
Holt
Holt_test_data
level
80
90



- Using now ARIMA (auto.arima) models:

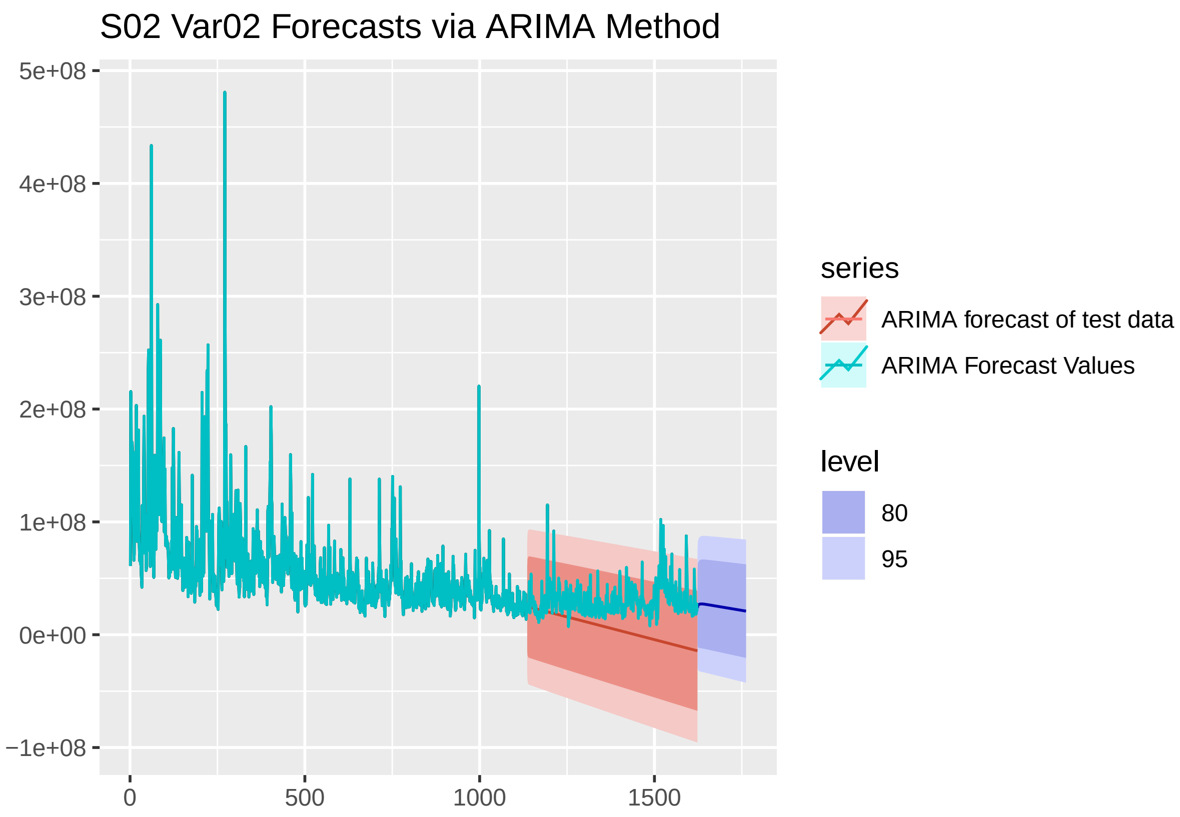


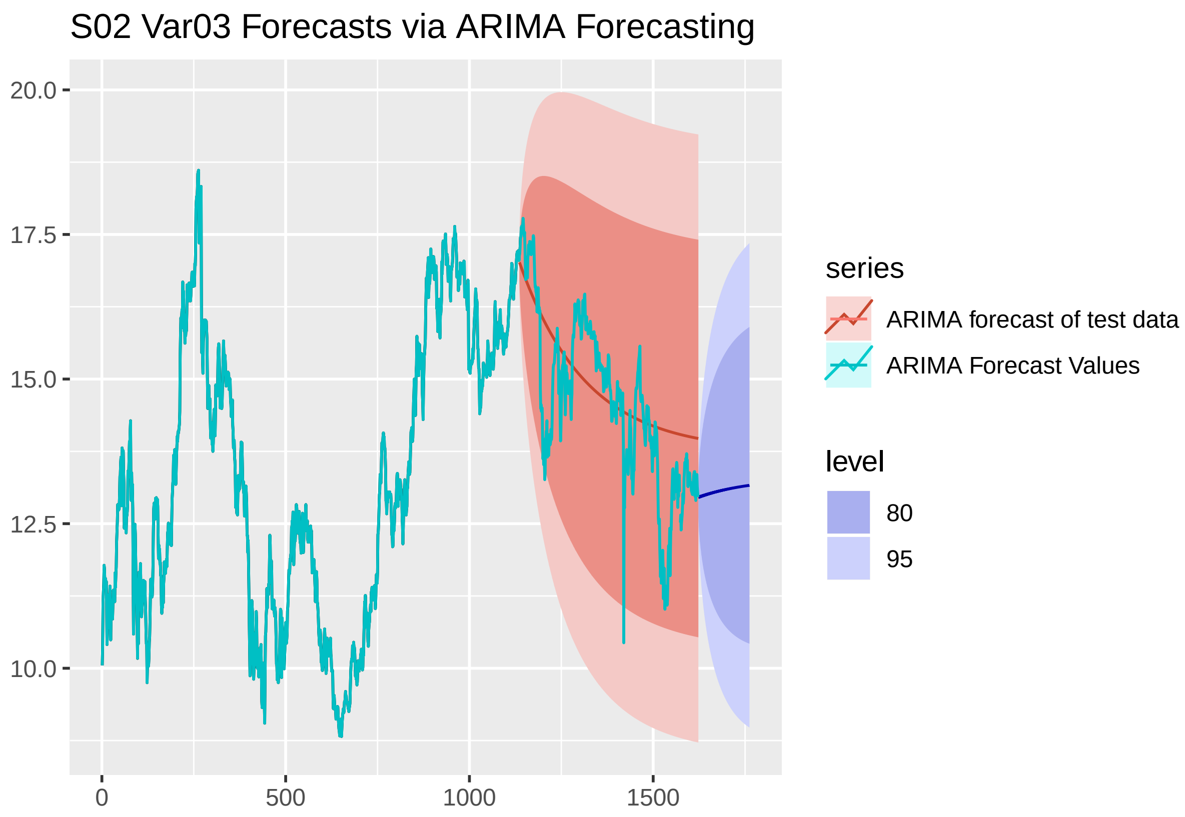
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,3) with drift
## Q* = 22.105, df = 4, p-value = 0.000191
##
## Model df: 6.   Total lags used: 10
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1) with non-zero mean
## Q* = 12.867, df = 6, p-value = 0.0452
##
## Model df: 4.   Total lags used: 10
```

- Forecasts using auto.arima:


S02 Var02 Forecasts via ARIMA Method
5e+08
4e+08
3e+08
2e+08
1e+08
0e+00
−1e+08
0
500
1000
1500
series
ARIMA forecast of test data
ARIMA Forecast Values
level
80
95



- We see that the ARIMA model used for Var02 shows good prediction for the confidence intervals and shows the downward trend. Using the auto.arima for Var03 doesn't show the best results from looking at the p-value of the residuals plot.
- Let's now compute some metrics of our models such as RMSE and MAPE and use the lowest value to predict the next 140 steps/points into the future.
- ARIMA evaluation
- Holt evaluation
- Naive evaluation
- Var02 MAPE evaluation

Method	Value
ARIMA	81.07837
Holt	148.73057
Naive	28.14745

- Var03 MAPE evaluation

Method	Value
ARIMA	6.065755
Holt	30.568227
Naive	18.367871

- Looks like in regards to the MAPE, the ARIMA model works for the Var03 model and that for Var02, the ARIMA model works best when letting auto.arima do all the work.
- Finally, save our predictions 140 steps ahead (h=140)

Appendix

```r
library(readxl)
library(dplyr)
library(ggplot2)
library(ggfortify)
library(GGally)
library(gridExtra)
library(forecast)
library(imputeTS)
library(tidyverse)
library(kableExtra)
s02_data <- read_xls("Set for Class.xls", n_max = 9732)
# For my group, I requested to look at only the Series = 's02'. Per assignment,
# you forecast Var02 and Var03 for S02

# Extract only seriesid, group, var02 and var03
s02_data <- s02_data %>% filter(group == "S02") %>%
  select("SeriesInd", "group", "Var02", "Var03")

summary(s02_data)
# Type Conversions. Change var02/03 to be time series
s02_data_var02_ts <- ts(s02_data$Var02, start = 1, end = 1622,
                        frequency=1)
s02_data_var03_ts <- ts(s02_data$Var03, start = 1, end = 1622,
                        frequency=1)
s02_data_var03_ts <- na.interpolation(s02_data_var03_ts, option = "spline")
# remove the large outlier the value == 38.28 per the summary
idx_outlier_var03 <- which.max(s02_data_var03_ts)
s02_data_var03_ts[idx_outlier_var03] <- NA
# time series plot of var02
var02_plot <- autoplot(s02_data_var02_ts) +
  ggtitle("S02 Var02 Time Series") +
  ylab("Var02")

# time series plot of var03
var03_plot <- autoplot(s02_data_var03_ts) +
  ggtitle("S02 Var03 Time Series") +
  ylab("Var03")

# 2x1 plot arrangement
grid.arrange(var02_plot, var03_plot)
ggAcf(s02_data_var02_ts)
ggAcf(s02_data_var03_ts)
# naive forecasts and predict 140 steps ahead with 80% confidence invterval

var02_window_training <- window(s02_data_var02_ts, start=1, end=floor(1622*0.7))
```

```
var02_window_test <- window(s02_data_var02_ts, start=floor(1622*0.7))

var03_window_training <- window(s02_data_var03_ts, start=1, end=floor(1622*0.7))
var03_window_test <- window(s02_data_var03_ts, start=floor(1622*0.7))

# train a naive forecast using training data
s02_var02_naive_test_train <- naive(var02_window_training,
                                    h = length(var02_window_test), level = c(80, 90))
s02_var03_naive_test_train <- naive(var03_window_training,
                                    h = length(var03_window_test), level = c(80, 90))

# forecasts using naive method using the test windows/values
s02_var02_naive_test_fit <- naive(s02_data_var02_ts, h = 140, level = c(80, 90))
s02_var03_naive_test_fit <- naive(s02_data_var03_ts, h = 140, level = c(80, 90))

# forecast values using forecast()

checkresiduals(s02_var02_naive_test_fit)
checkresiduals(s02_var03_naive_test_fit)
# var02 plot
autoplot(s02_var02_naive_test_fit) +
  autolayer(s02_var02_naive_test_train, series="Naive") +
  autolayer(s02_data_var02_ts, series="Naive") +
  ggtitle("S02 Var02 Forecasts via Naive Forecasting") +
  xlab("") + ylab("")

# var02 plot
autoplot(s02_var03_naive_test_fit) +
  autolayer(s02_var03_naive_test_train, series="Naive") +
  autolayer(s02_data_var03_ts, series="Naive") +
  ggtitle("S02 Var03 Forecasts via Naive Forecasting") +
  xlab("") + ylab("")
# make holt predictions using the training data
s02_var02_holt_test_train <- holt(var02_window_training,
                                  h = length(var02_window_test), level=c(80,90))
s02_var03_holt_test_train <- holt(var03_window_training,
                                  h = length(var03_window_test), level=c(80,90))

# forecasts using naive method using the test windows/values
s02_var02_holt_test_fit <- holt(s02_data_var02_ts, h = 140, level=c(80,90))
s02_var03_holt_test_fit <- holt(s02_data_var03_ts, h = 140, level=c(80,90))

checkresiduals(s02_var02_holt_test_fit)
checkresiduals(s02_var03_holt_test_fit)
# var02 plot
autoplot(s02_var02_holt_test_fit) +
  autolayer(s02_var02_holt_test_train, series="Holt") +
  autolayer(s02_data_var02_ts, series="Holt_test_data") +
  ggtitle("S02 Var02 Forecasts via Holt's Method") +
  xlab("") + ylab("")


# var02 plot
```

```
autoplot(s02_var03_holt_test_fit) +
  autolayer(s02_var03_holt_test_train, series="Holt") +
  autolayer(s02_data_var03_ts, series="Holt_test_data") +
  ggtitle("S02 Var03 Forecasts via Holt's Forecasting") +
  xlab("") + ylab("")
# train an arima model using the training data
# var02 not seasonal but more of a trend
s02_var02_arima_train <- auto.arima(var02_window_training, seasonal = FALSE)
s02_var03_arima_train <- Arima(var03_window_training, order = c(2,0,1))

# make forecasts of the training data using arima models
s02_var02_arima_fit <- forecast(s02_var02_arima_train, h=length(var02_window_test))
s02_var03_arima_fit <- forecast(s02_var03_arima_train, h=length(var03_window_test))

# forecast on the test data for arima
s02_var02_arima_test <- auto.arima(s02_data_var02_ts, seasonal = FALSE) %>%
  forecast(h=140)
s02_var03_arima_test <- Arima(s02_data_var03_ts, order = c(2,0,1), seasonal = FALSE) %>%
  forecast(h=140)

# stl decomposition

checkresiduals(s02_var02_arima_test)
checkresiduals(s02_var03_arima_test)

# var02 plot
autoplot(s02_var02_arima_test) +
  autolayer(s02_var02_arima_fit, series="ARIMA forecast of test data") +
  autolayer(s02_data_var02_ts, series="ARIMA Forecast Values") +
  ggtitle("S02 Var02 Forecasts via ARIMA Method") +
  xlab("") + ylab("")


# var02 plot
autoplot(s02_var03_arima_test) +
  autolayer(s02_var03_arima_fit, series="ARIMA forecast of test data") +
  autolayer(s02_data_var03_ts, series="ARIMA Forecast Values") +
  ggtitle("S02 Var03 Forecasts via ARIMA Forecasting") +
  xlab("") + ylab("")
# ARIMA
arima_accuracy_var02 <-
  data.frame(accuracy(s02_var02_arima_fit, var02_window_test))[2, "MAPE"]
arima_accuracy_var03 <-
  data.frame(accuracy(s02_var03_arima_fit, var03_window_test))[2, "MAPE"]
holt_accuracy_var02 <- data.frame(accuracy(forecast(s02_var02_holt_test_train,
                  h=length(var02_window_test)), var02_window_test))[2, "MAPE"]

holt_accuracy_var03 <- data.frame(accuracy(forecast(s02_var03_holt_test_train,
                  h=length(var03_window_test)), var03_window_test))[2, "MAPE"]

naive_accuracy_s02 <- data.frame(accuracy(forecast(s02_var02_naive_test_train,
                  h=length(var02_window_test)), var02_window_test))[2, "MAPE"]
```

```r
naive_accuracy_s03 <- data.frame(accuracy(forecast(s02_var03_naive_test_train,
                  h=length(var03_window_test)), var03_window_test))[2, "MAPE"]
var02_mape <- data.frame(Method=c("ARIMA", "Holt", "Naive"),
                         Value=c(arima_accuracy_var02,
                                 holt_accuracy_var02,
                                 naive_accuracy_s02))
kable(var02_mape) %>% kable_styling(fixed_thead = T)
var03_mape <- data.frame(Method=c("ARIMA", "Holt", "Naive"),
                         Value=c(arima_accuracy_var03,
                                 holt_accuracy_var03,
                                 naive_accuracy_s03))
kable(var03_mape) %>% kable_styling(fixed_thead = T)
predictions_var02 <- s02_var02_arima_test$mean
write.csv(round(predictions_var02), "s02_var02_forecasts.csv")
predictions_var03 <- s02_var03_arima_test$mean
write.csv(round(predictions_var03, digits = 3), "s02_var03_forecasts.csv")
```