Project1

Alice Ding

2024-06-18

Data Importing and Indexing

```
data_start_ind <- 1
data_end_ind <- 1622
forecast_stary_ind <- 1623
forecast_end_ind <- 1722

path <- paste(getwd(), '/Data Set for Class.xls', sep="")
sheet_name <- 'S02'

# Read the specified sheet from the Excel file
s02 <- read_excel(path, sheet = sheet_name)
```

Data Visualization

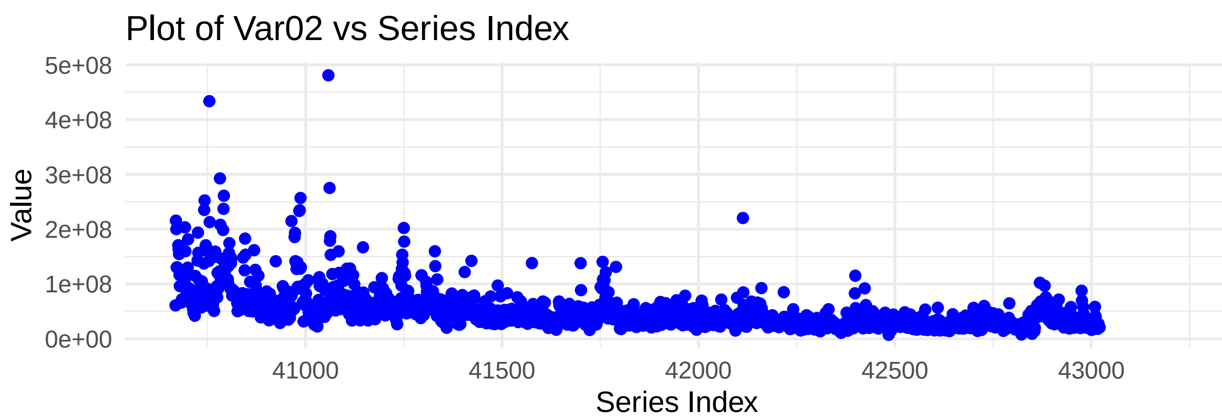
```
var2_plot <- ggplot(s02, aes(x = SeriesInd, y = Var02)) +
  geom_point(color = "blue") +
  labs(title = "Plot of Var02 vs Series Index", x = "Series Index", y = "Value") +
  theme_minimal()
```

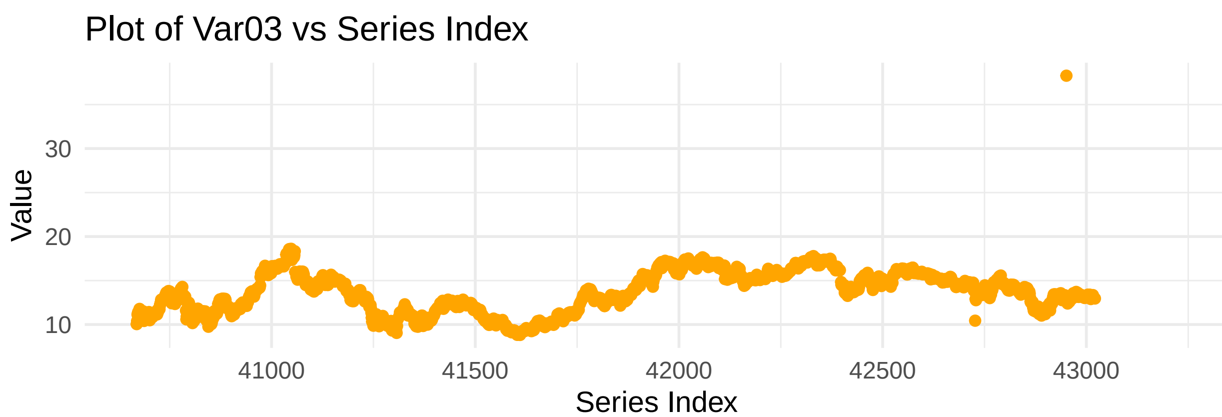
```
var3_plot <- ggplot(s02, aes(x = SeriesInd, y = Var03)) +
  geom_point(color = "orange") +
  labs(title = "Plot of Var03 vs Series Index", x = "Series Index", y = "Value") +
  theme_minimal()
```

```
grid.arrange(var2_plot, var3_plot, nrow = 2)
```

```
## Warning: Removed 140 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 144 rows containing missing values or values outside the scale range
## (`geom_point()`).
```





Data Imputation

I'm using linear imputation, so creating a line of best fit between the last two known points and filling in missing values along that line. This works for Var03, however for Var02, I will impute the median given how it contains more static.

There are also some outliers in Var03 that will be replaced with linear imputation as well.

```
data_range <- which(s02$SeriesInd < 43022)
na_var3 <- which(is.na(s02$Var03[data_range]))
```

```
# Define a function to detect outliers (using z-scores here for simplicity)
is_outlier <- function(x) {
  z_scores <- (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
  return(abs(z_scores) > 2) # You can adjust the threshold as needed
}

# Identify outliers
outliers_var3 <- which(is_outlier(s02$Var03[data_range]))

# Combine NA values and outliers indices
na_and_outliers_var3 <- unique(c(na_var3, outliers_var3))

# Exclude outliers from data used for interpolation
```

```r
valid_data_range <- data_range[!data_range %in% na_and_outliers_var3]

# Perform linear interpolation excluding outliers and NA values
imputed_var3 <- approx(x = s02$SeriesInd[valid_data_range], y = s02$Var03[valid_data_range],
                       xout = s02$SeriesInd[data_range])$y
```

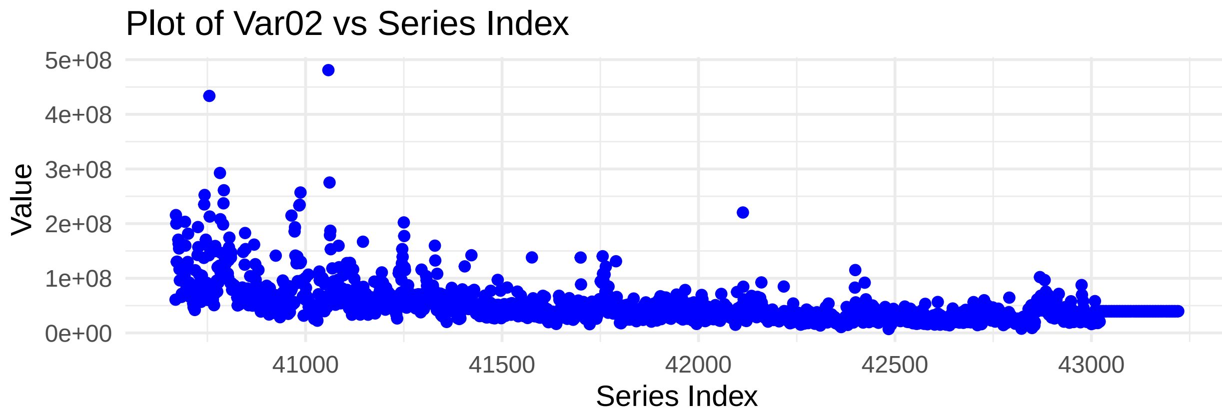
```r
s02 <- s02 |>
  mutate(Var02 = replace_na(Var02, median(Var02, na.rm=TRUE)))

# Impute missing values and outliers with interpolated values
s02$Var03[data_range][na_and_outliers_var3] <- imputed_var3[na_and_outliers_var3]
```

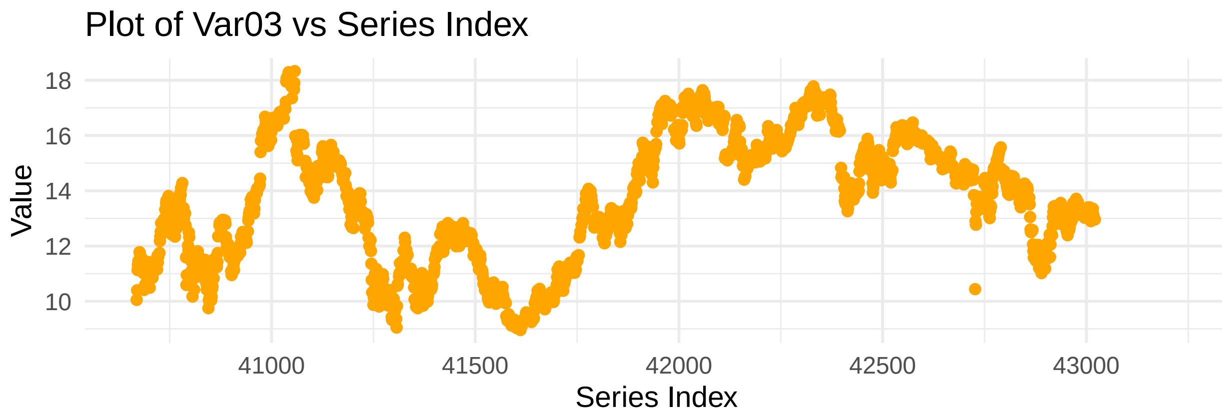
```r
var2_plot <- ggplot(s02, aes(x = SeriesInd, y = Var02)) +
  geom_point(color = "blue") +
  labs(title = "Plot of Var02 vs Series Index", x = "Series Index", y = "Value") +
  theme_minimal()
```

```r
var3_plot <- ggplot(s02, aes(x = SeriesInd, y = Var03)) +
  geom_point(color = "orange") +
  labs(title = "Plot of Var03 vs Series Index", x = "Series Index", y = "Value") +
  theme_minimal()
```

```r
grid.arrange(var2_plot, var3_plot, nrow = 2)
```

```
## Warning: Removed 140 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

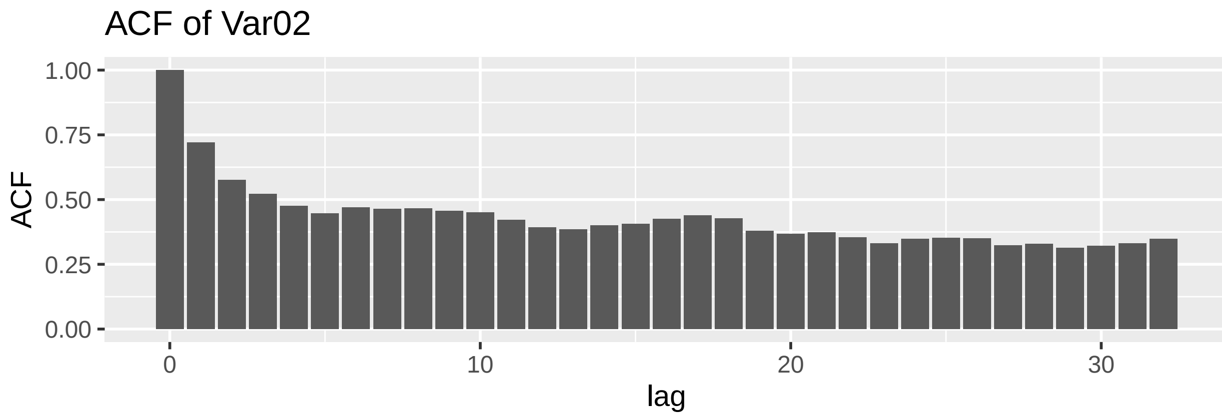


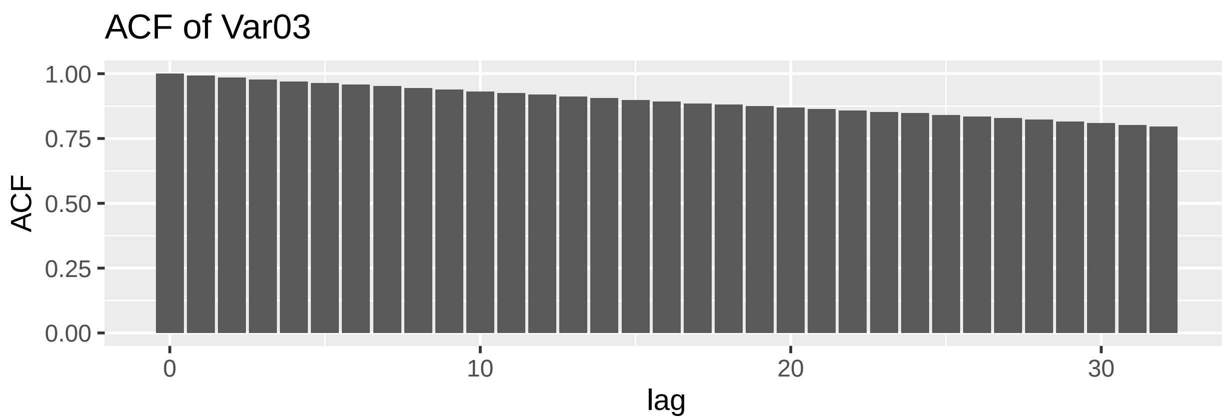


Values to forecast: 43022 - 43221 index numbers: 1623 - 1762

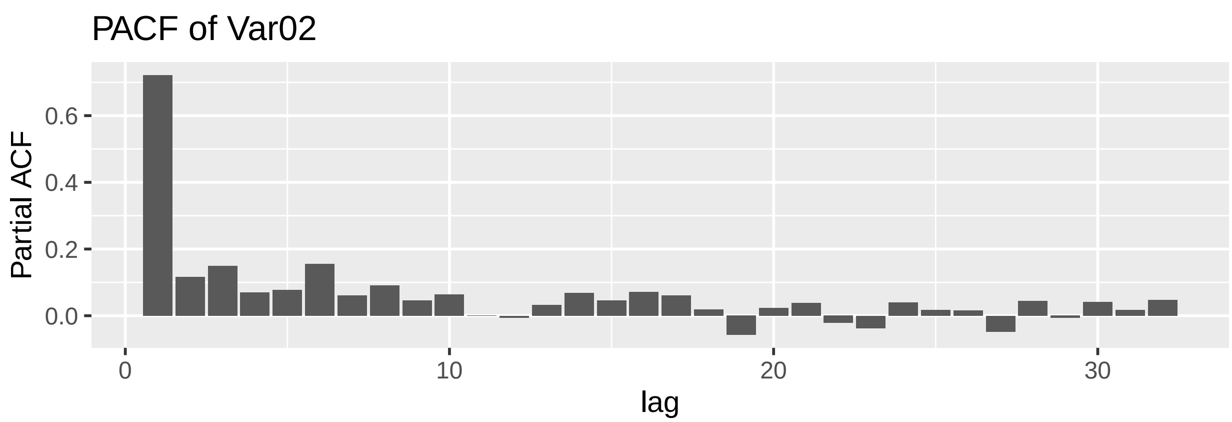
Checking for Stationarity

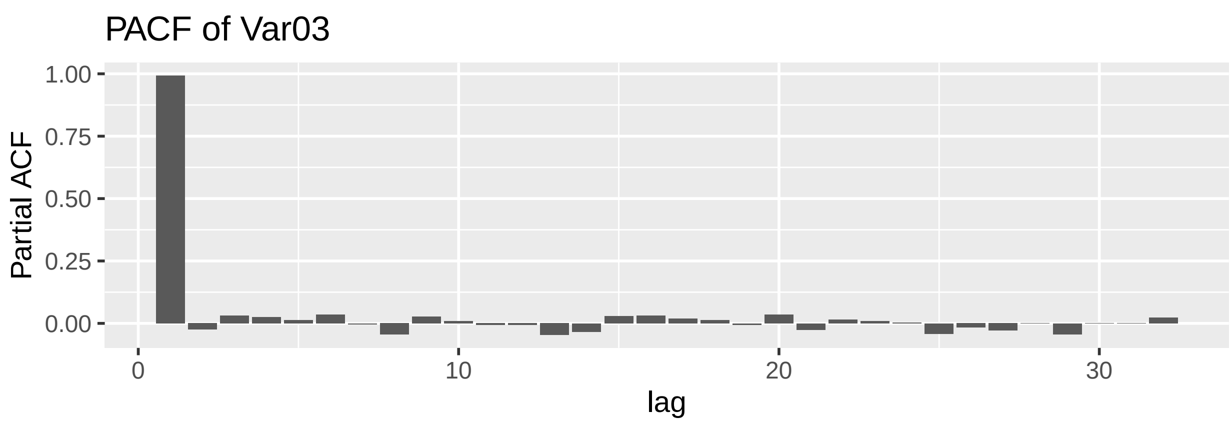
```
acf_var2 <- acf(s02$Var02[data_range], plot = FALSE)
acf_var3 <- acf(s02$Var03[data_range], plot = FALSE)

acf_var2_df <- data.frame(lag = acf_var2$lag, acf = acf_var2$acf)
acf_var3_df <- data.frame(lag = acf_var3$lag, acf = acf_var3$acf)

acf1 <- ggplot(acf_var2_df, aes(x = lag, y = acf)) +
  geom_bar(stat = "identity") +
  labs(title = "ACF of Var02", y = 'ACF')

acf2 <- ggplot(acf_var3_df, aes(x = lag, y = acf)) +
  geom_bar(stat = "identity") +
  labs(title = "ACF of Var03", y = 'ACF')

grid.arrange(acf1, acf2, nrow=2)
```






```
pacf_var2 <- pacf(s02$Var02[data_range], plot = FALSE)
pacf_var3 <- pacf(s02$Var03[data_range], plot = FALSE)

pacf_var2_df <- data.frame(lag = pacf_var2$lag, pacf = pacf_var2$acf)
pacf_var3_df <- data.frame(lag = pacf_var3$lag, pacf = pacf_var3$acf)

pacf1 <- ggplot(pacf_var2_df, aes(x = lag, y = pacf)) +
  geom_bar(stat = "identity") +
  labs(title = "PACF of Var02", y = 'Partial ACF')

pacf2 <- ggplot(pacf_var3_df, aes(x = lag, y = pacf)) +
  geom_bar(stat = "identity") +
  labs(title = "PACF of Var03", y = 'Partial ACF')

grid.arrange(pacf1, pacf2, nrow=2)
```





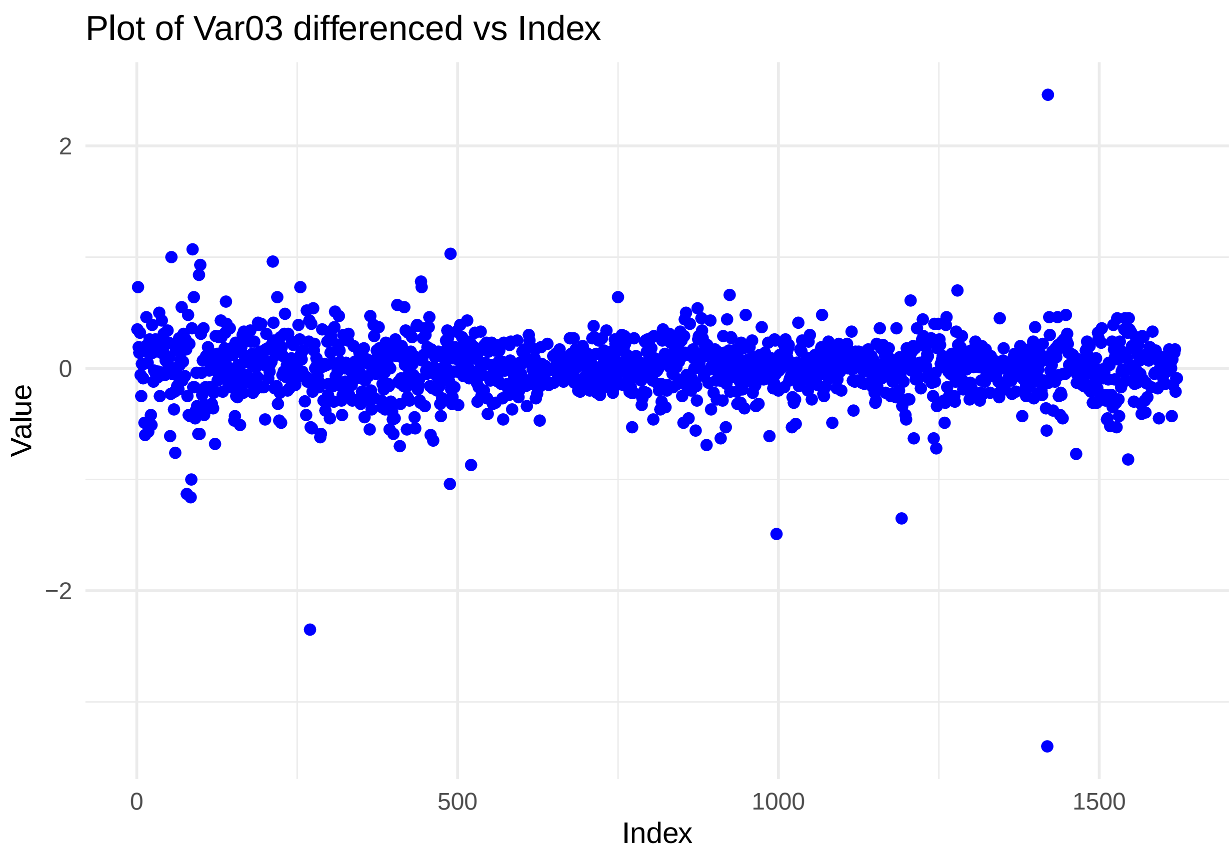
Var03 is non-stationary while Var02 is stationary.

We will preforming differencing to make Var03 stationary.

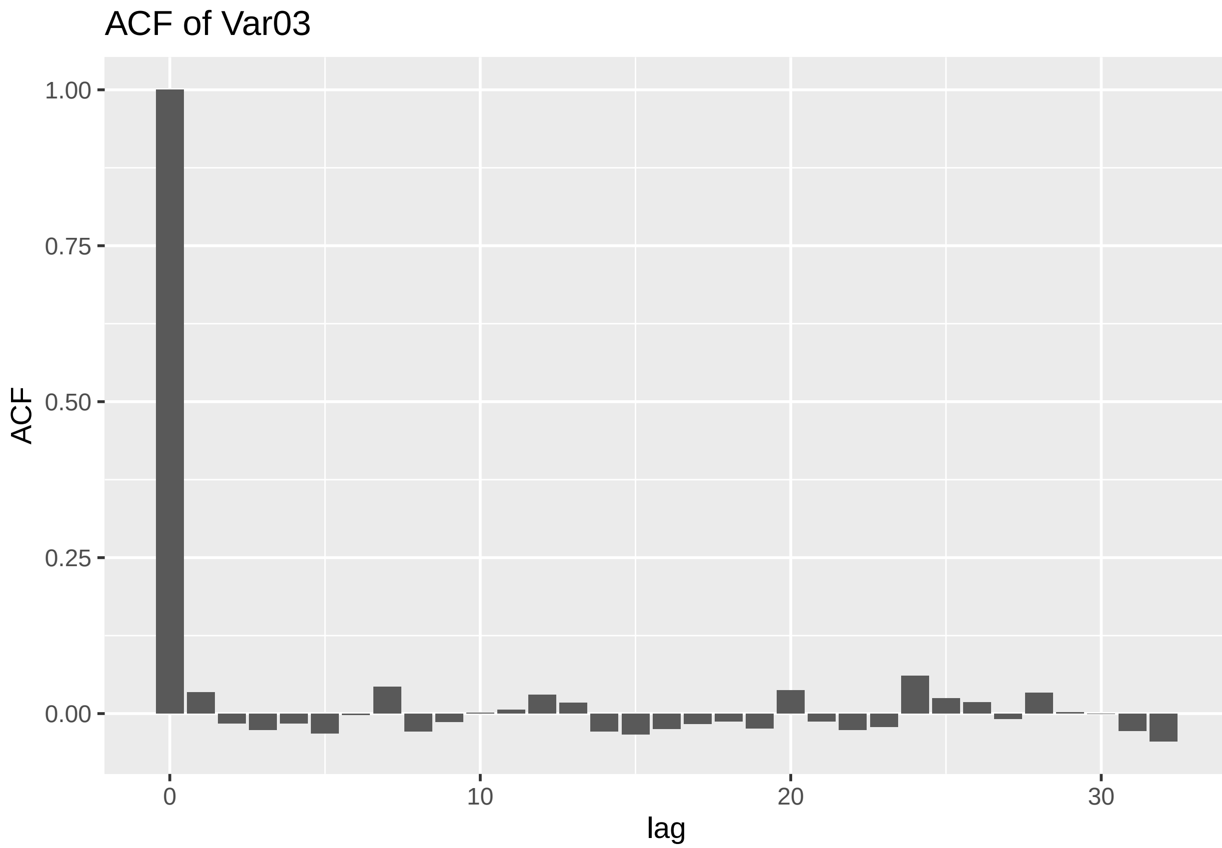
```
var3_diff <- diff(s02$Var03[data_range], differences = 1)

var3_diff_df <- data.frame(Index = seq_along(var3_diff), Value = var3_diff)
```

```
var3_plot <- ggplot(var3_diff_df, aes(x = Index, y = Value)) +
  geom_point(color = "blue") +
  labs(title = "Plot of Var03 differenced vs Index", x = "Index", y = "Value") +
  theme_minimal()
```

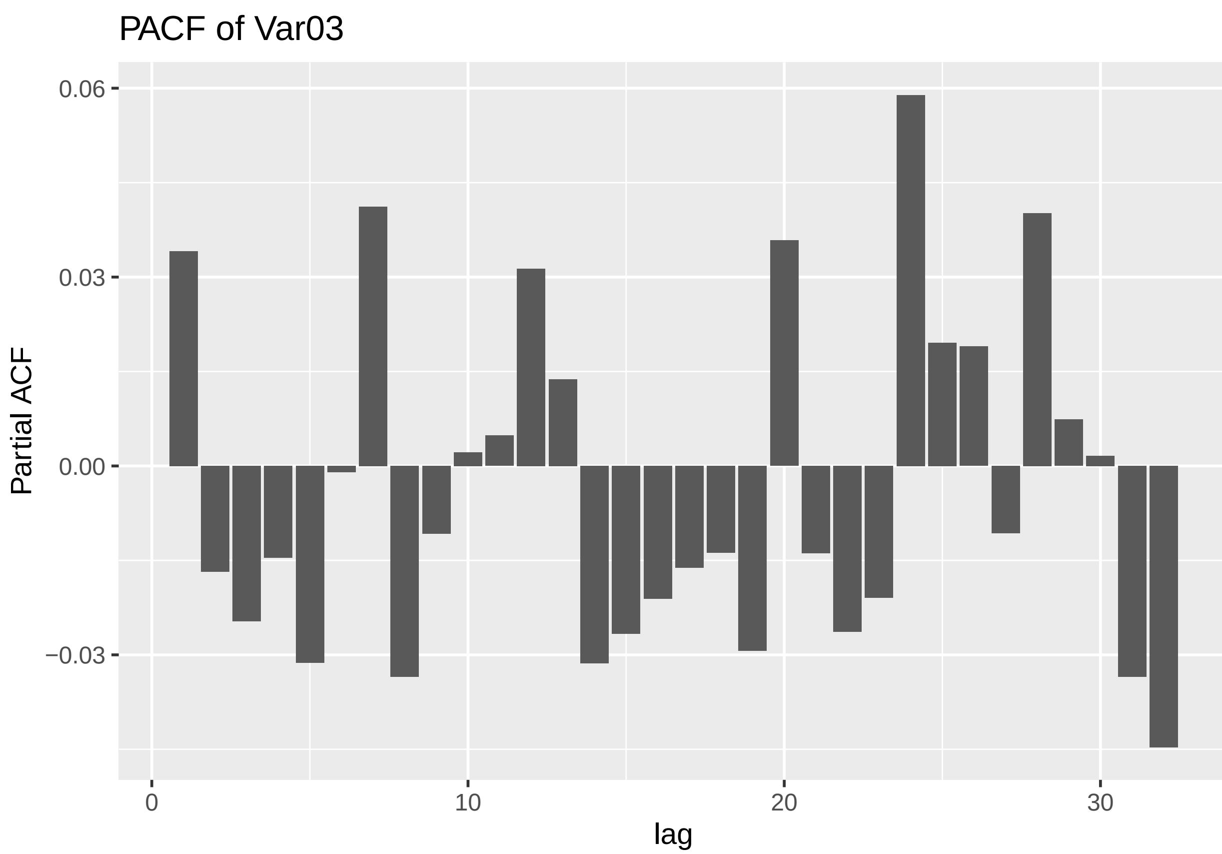
```
grid.arrange(var3_plot, nrow = 1)
```




```r
acf_var3 <- acf(var3_diff, plot = FALSE)

acf_var3_df <- data.frame(lag = acf_var3$lag, acf = acf_var3$acf)

acf2 <- ggplot(acf_var3_df, aes(x = lag, y = acf)) +
  geom_bar(stat = "identity") +
  labs(title = "ACF of Var03", y = 'ACF')

grid.arrange(acf2, nrow=1)
```




```
pacf_var3 <- pacf(var3_diff, plot = FALSE)

pacf_var3_df <- data.frame(lag = pacf_var3$lag, pacf = pacf_var3$acf)

pacf2 <- ggplot(pacf_var3_df, aes(x = lag, y = pacf)) +
  geom_bar(stat = "identity") +
  labs(title = "PACF of Var03", y = 'Partial ACF')

grid.arrange(pacf2, nrow=1)
```

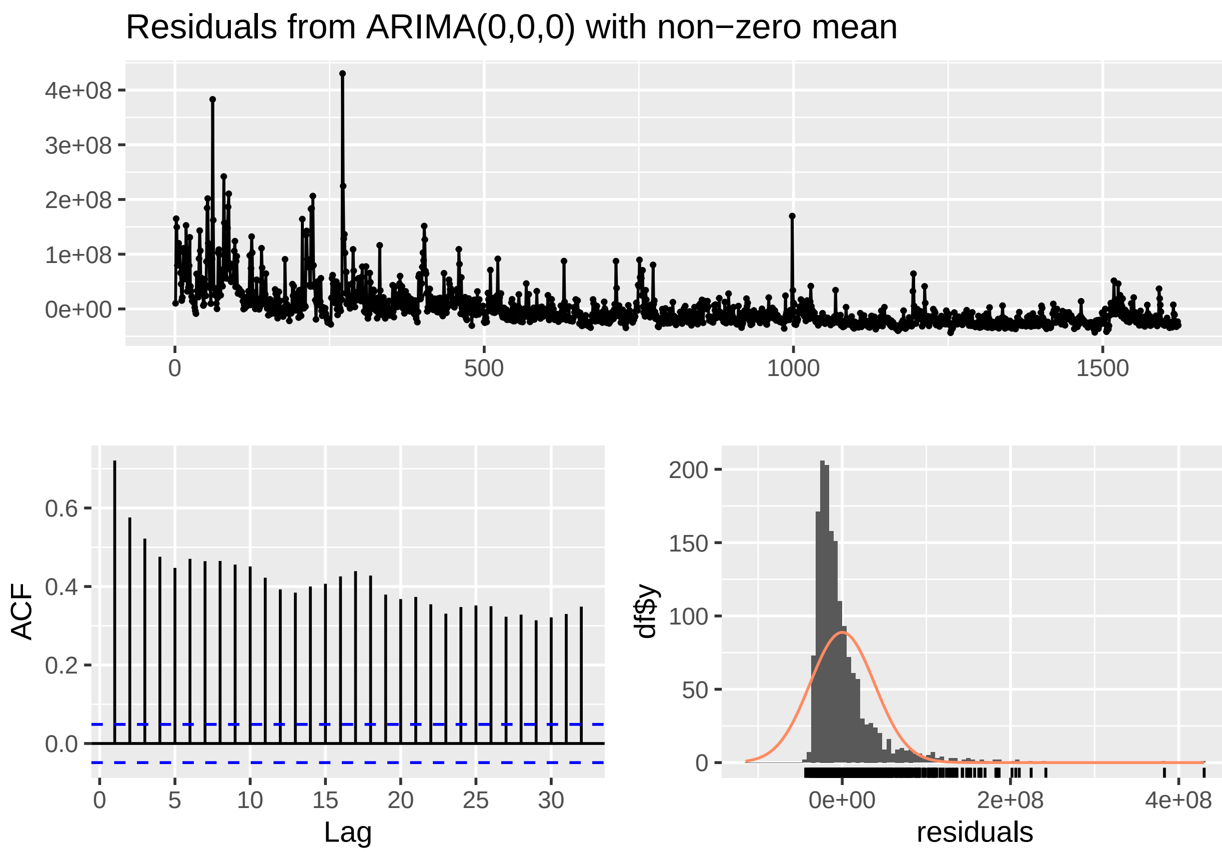


Forecasting

```r
fit_var2 <- auto.arima(s02$Var02[data_range], stationary = TRUE)
summary(fit_var2)
```

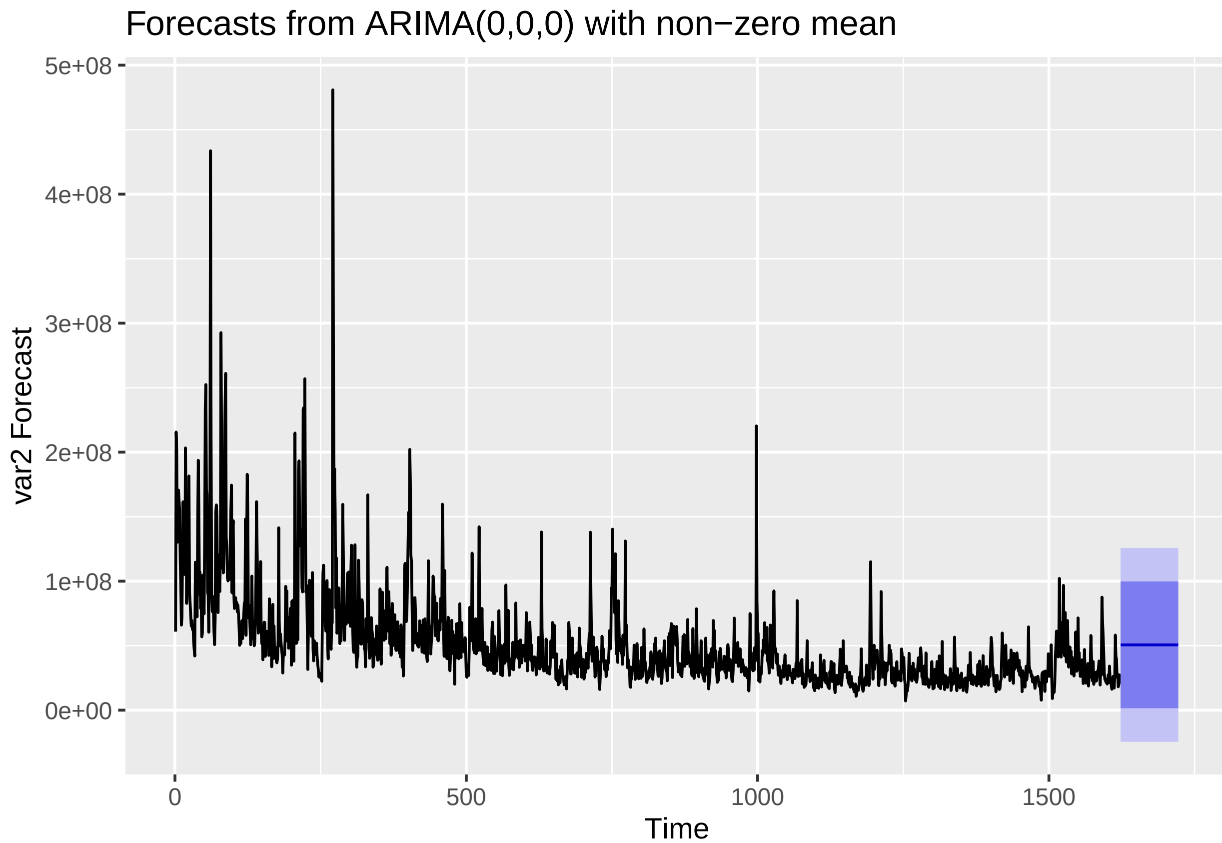
```
## Series: s02$Var02[data_range]
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##           mean
##       50633098
## s.e.   1049544
##
## sigma^2 = 1.47e+15:  log likelihood = -30624.68
## AIC=61253.35   AICc=61253.36   BIC=61264.14
##
## Training set error measures:
##                         ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -3.865758e-07 38334964 24597954 -38.57837 59.99964 1.571629
##                   ACF1
## Training set 0.7209905
```

```
checkresiduals(fit_var2)
```

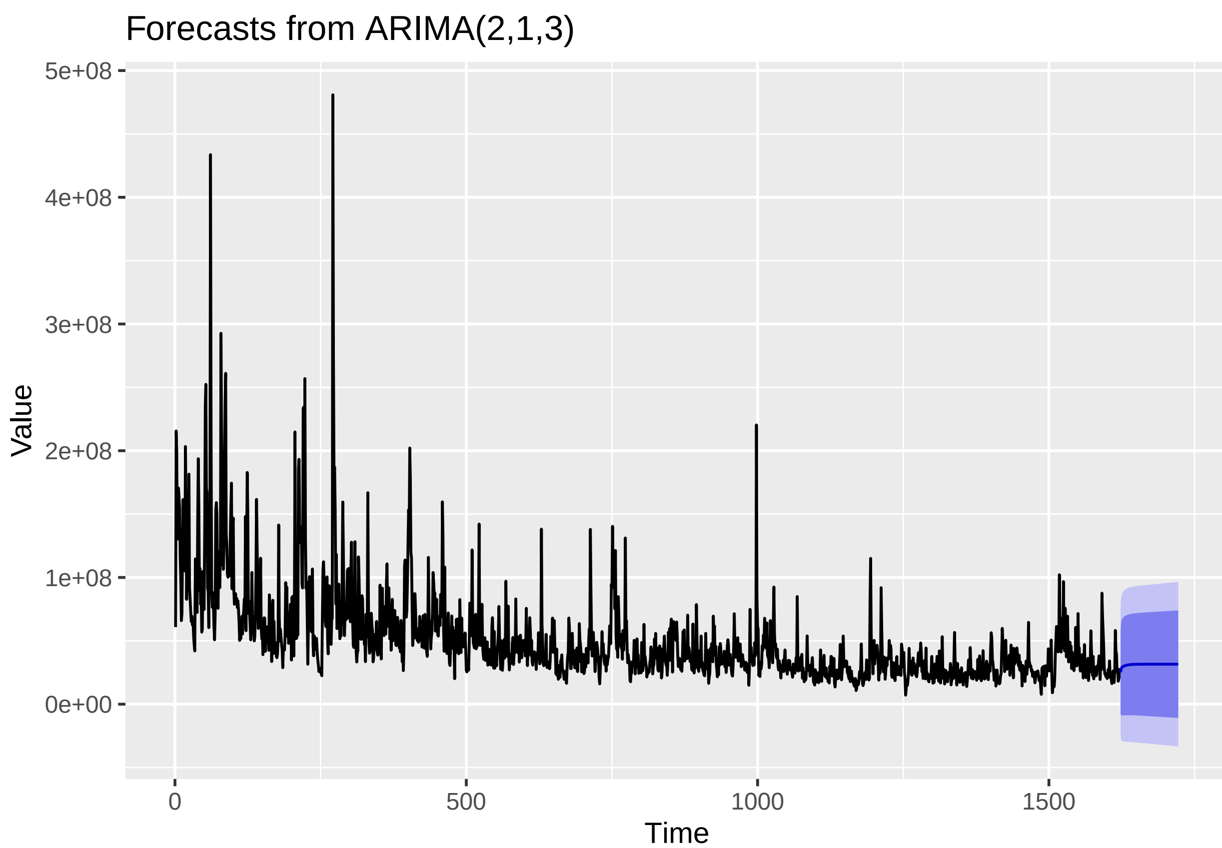



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,0) with non-zero mean
## Q* = 4259.7, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

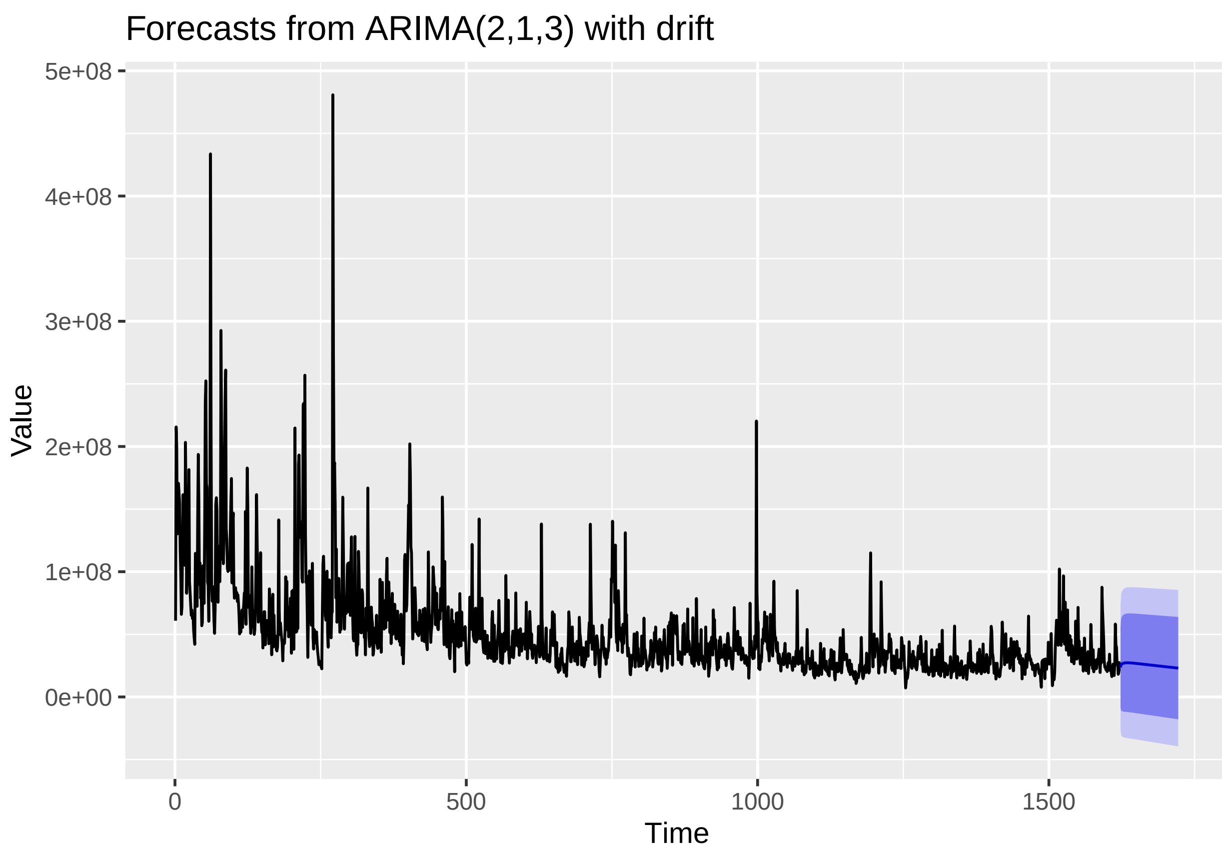
```
fc_var2 <- forecast(fit_var2, h=100)
autoplot(fc_var2) + ylab('var2 Forecast')
```



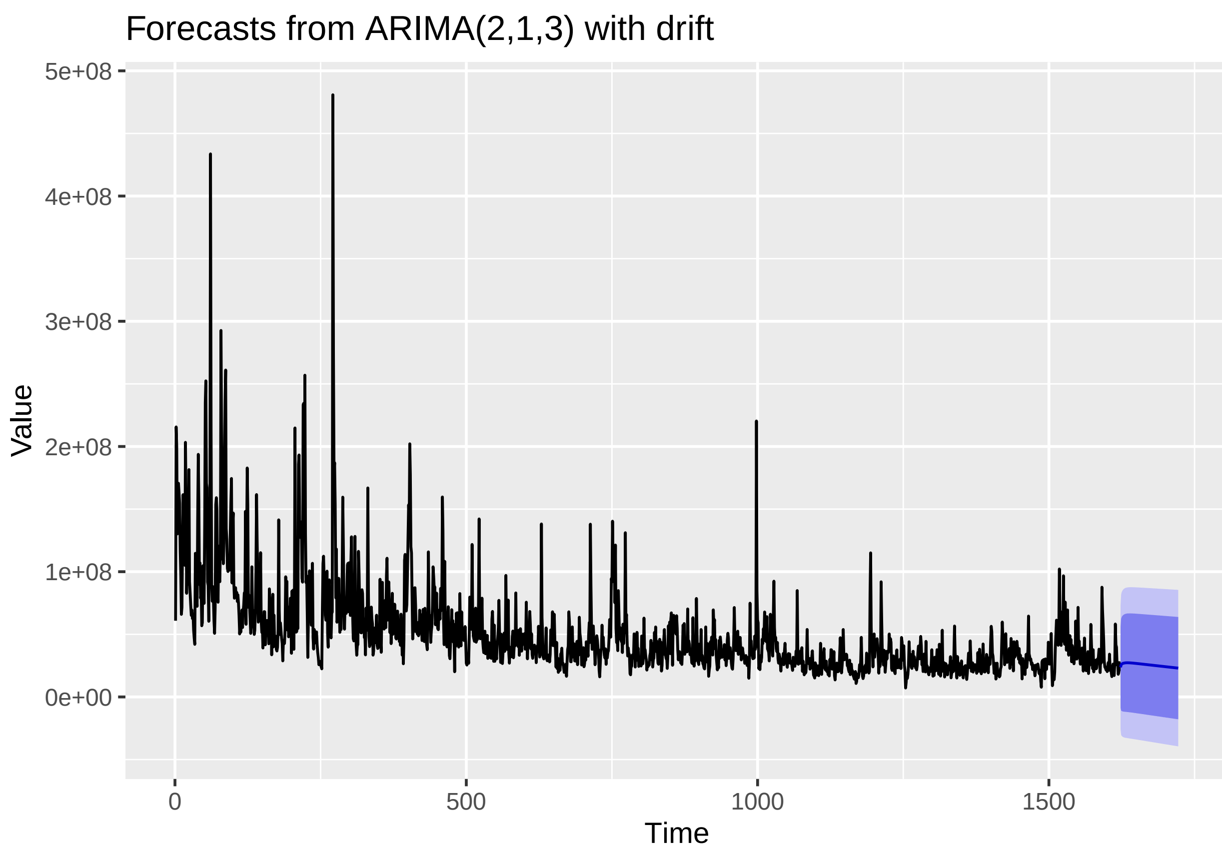

```
fit <- Arima(s02$Var02[data_range], order=c(2,1,3), include.constant=FALSE)
fc <- forecast(fit, h=100)
autoplot(fc) + ylab('Value')
```




```r
fit <- auto.arima(s02$Var02[data_range])
fc <- forecast(fit, h=100)
autoplot(fc) + ylab('Value')
```



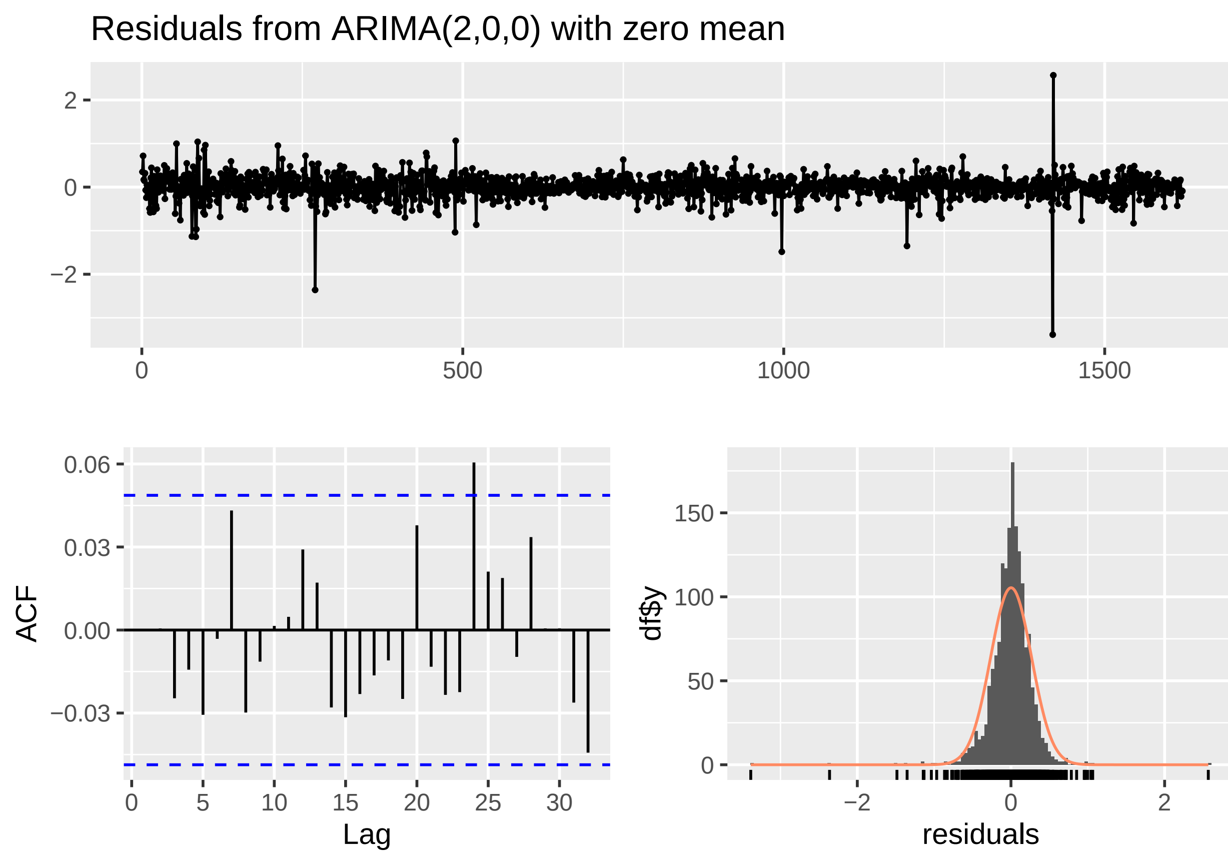

```
fit <- Arima(s02$Var02[data_range], order=c(2,1,3), include.drift=TRUE)
fc <- forecast(fit, h=100)
autoplot(fc) + ylab('Value')
```




```r
fit_var3 <- auto.arima(var3_diff, stationary = TRUE)
summary(fit_var3)
```

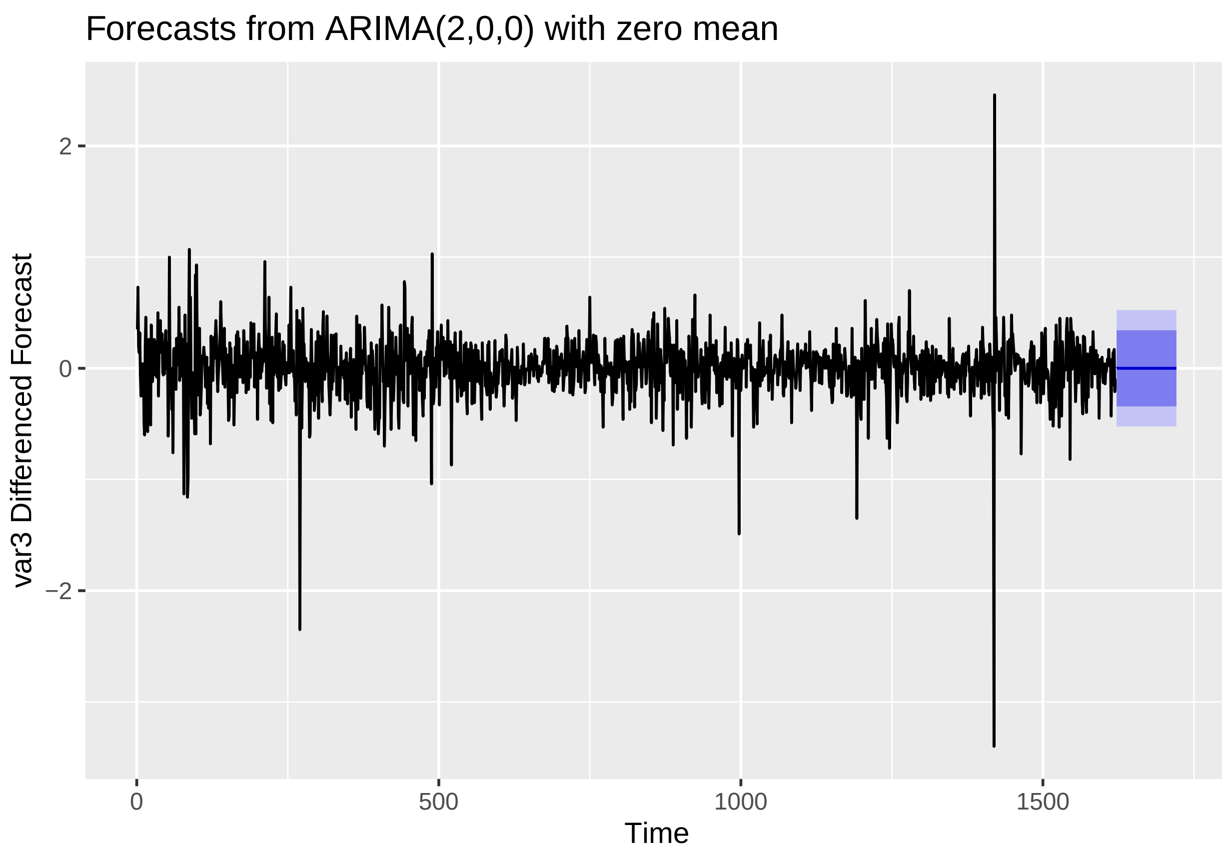
```
## Series: var3_diff
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2
##       0.0347  -0.0168
## s.e.  0.0248   0.0249
##
## sigma^2 = 0.07117:  log likelihood = -157.21
## AIC=320.41   AICc=320.43   BIC=336.58
##
## Training set error measures:
##                       ME      RMSE       MAE MPE MAPE      MASE          ACF1
## Training set 0.001762845 0.2666126 0.1774209 NaN  Inf 0.7512809 -0.0004325905
```

```r
checkresiduals(fit_var3)
```

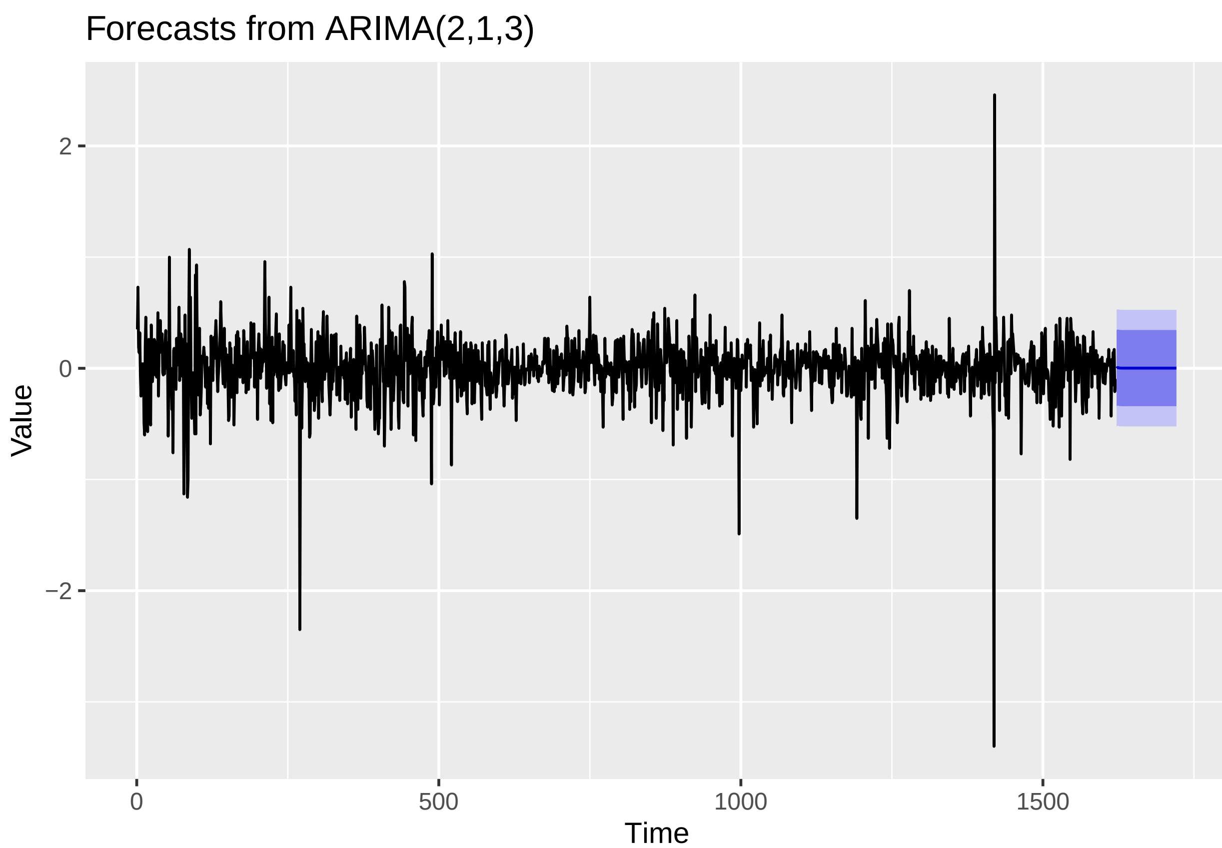



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0) with zero mean
## Q* = 7.5738, df = 8, p-value = 0.4762
##
## Model df: 2.   Total lags used: 10
```

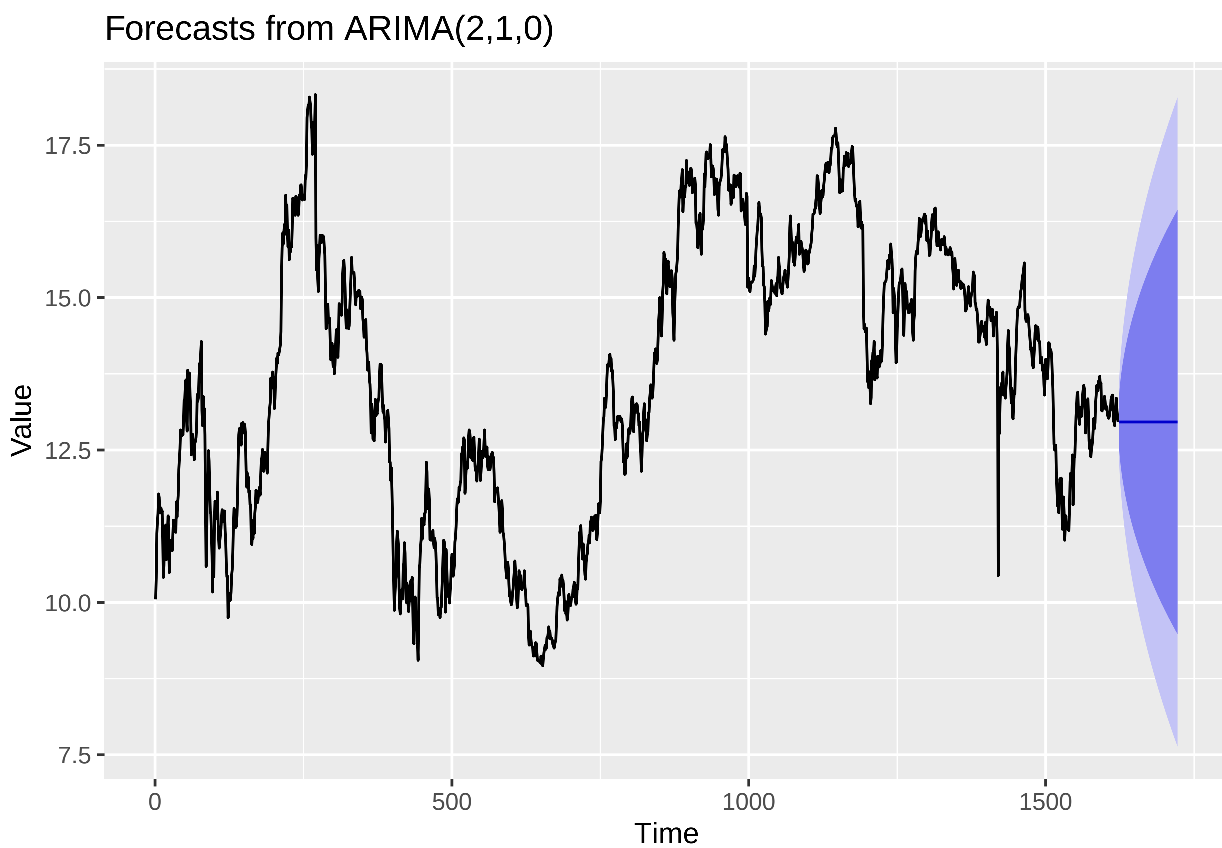
```
fc_var3 <- forecast(fit_var3, h=100)
autoplot(fc_var3) + ylab('var3 Differenced Forecast')
```



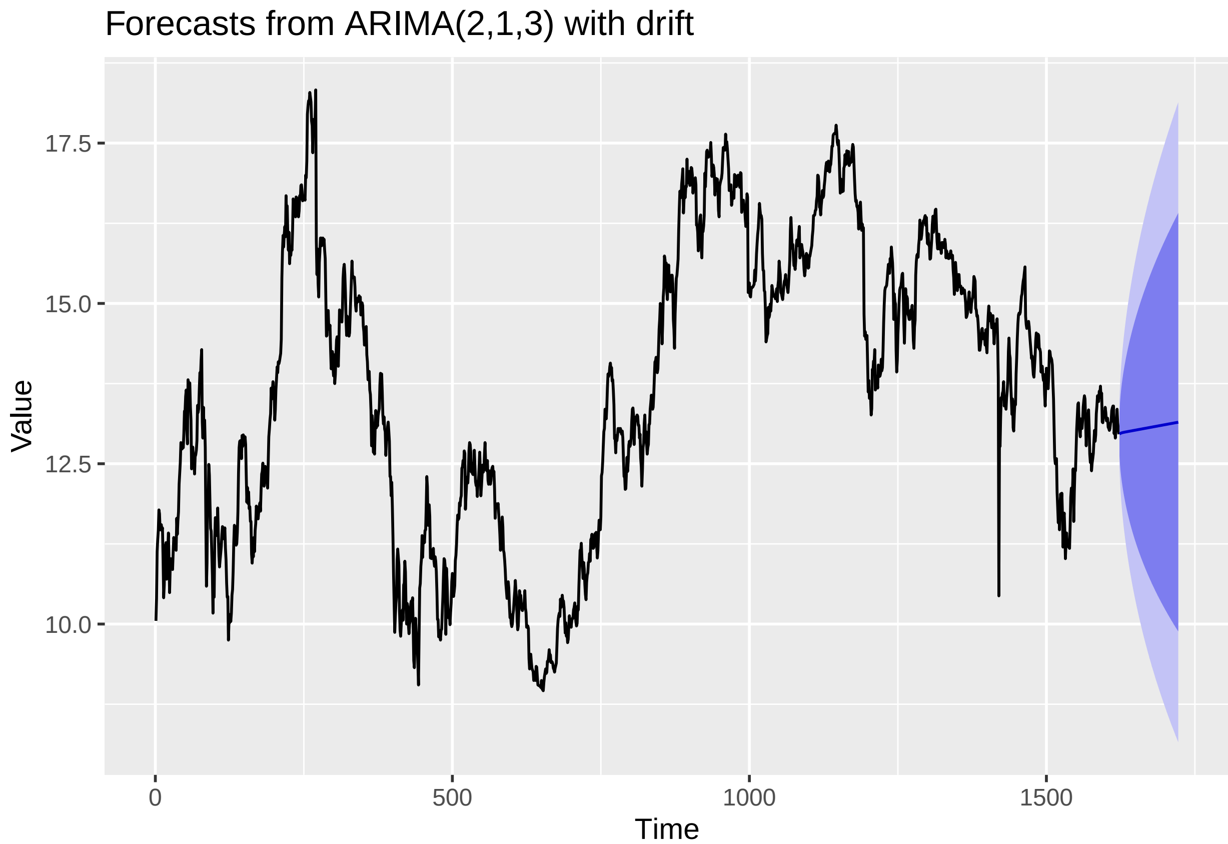

```r
fit <- Arima(var3_diff, order=c(2,1,3), include.constant=FALSE)
fc <- forecast(fit, h=100)
autoplot(fc) + ylab('Value')
```




```r
fit <- auto.arima(s02$Var03[data_range])
fc <- forecast(fit, h=100)
autoplot(fc) + ylab('Value')
```




```r
fit <- Arima(s02$Var03[data_range], order=c(2,1,3), include.drift=TRUE)
fc <- forecast(fit, h=100)
autoplot(fc) + ylab('Value')
```


Forecasts from ARIMA(2,1,3) with drift
17.5
15.0
12.5
10.0
Value
0
500
1000
1500
Time