Groundwater Level Analysis

Grant Corkren

2023-04-28

This project was done in a consulting class where I consulted the Geological Survey of Alabama on the height of water in a well located in Tuscaloosa County. I was tasked with finding if their was a trend and seasonality. Additionally I was asked if it could be forecasted and if rain effected well water height.

This analysis includes time series decomposition to answer the question of trend and seasonality. Their is no trend but their is seasonality. For modeling ARIMA models their needs to be stationarity so a seasonal difference was taken. It also includes an ARMA model with rain as a variable to answer whether or not rain effects well water level. The resulting coefficient of rain in that ARIMA model was 0.02 and was statistically significant.

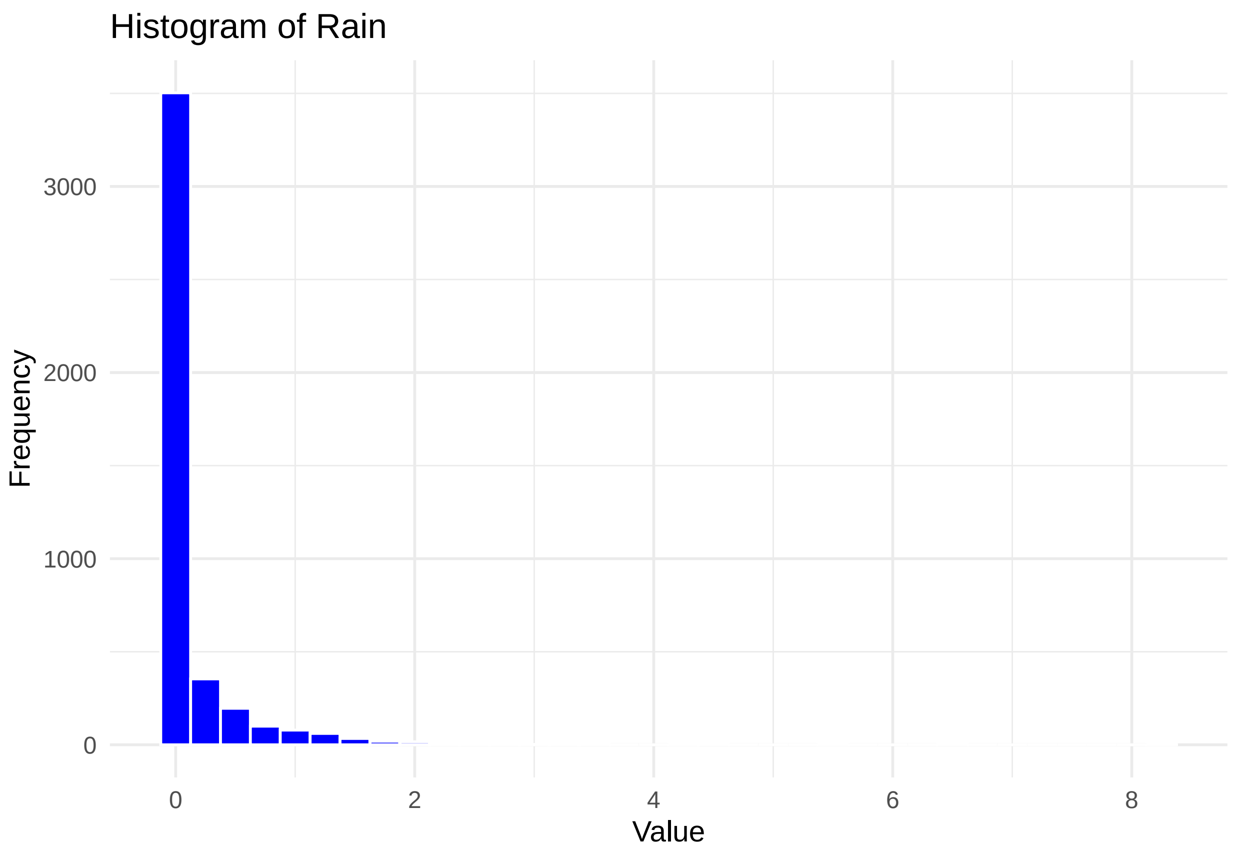
ARIMA models were applied to answer the question of forecasting. Rain is not used as a variable as that would need to be forecasted also or used conditionally.

Ultimately all of the forecasting models suffered from a non constant variance. A second analysis has been done to explore the time series data further and apply GARCH models.

```r
tail(data)
```

```
## # A tibble: 6 x 5
##   DATE                waterlevel ...3  elevation precip
##   <dttm>                   <dbl> <lgl>     <dbl>  <dbl>
## 1 2023-01-15 00:00:00       39.4 NA        -39.4   0
## 2 2023-01-16 00:00:00       39.5 NA        -39.5   0
## 3 2023-01-17 00:00:00       39.6 NA        -39.6   0.78
## 4 2023-01-18 00:00:00       39.6 NA        -39.6   0
## 5 2023-01-19 00:00:00       39.5 NA        -39.5   0.36
## 6 2023-01-20 00:00:00       39.3 NA        -39.3   0
```

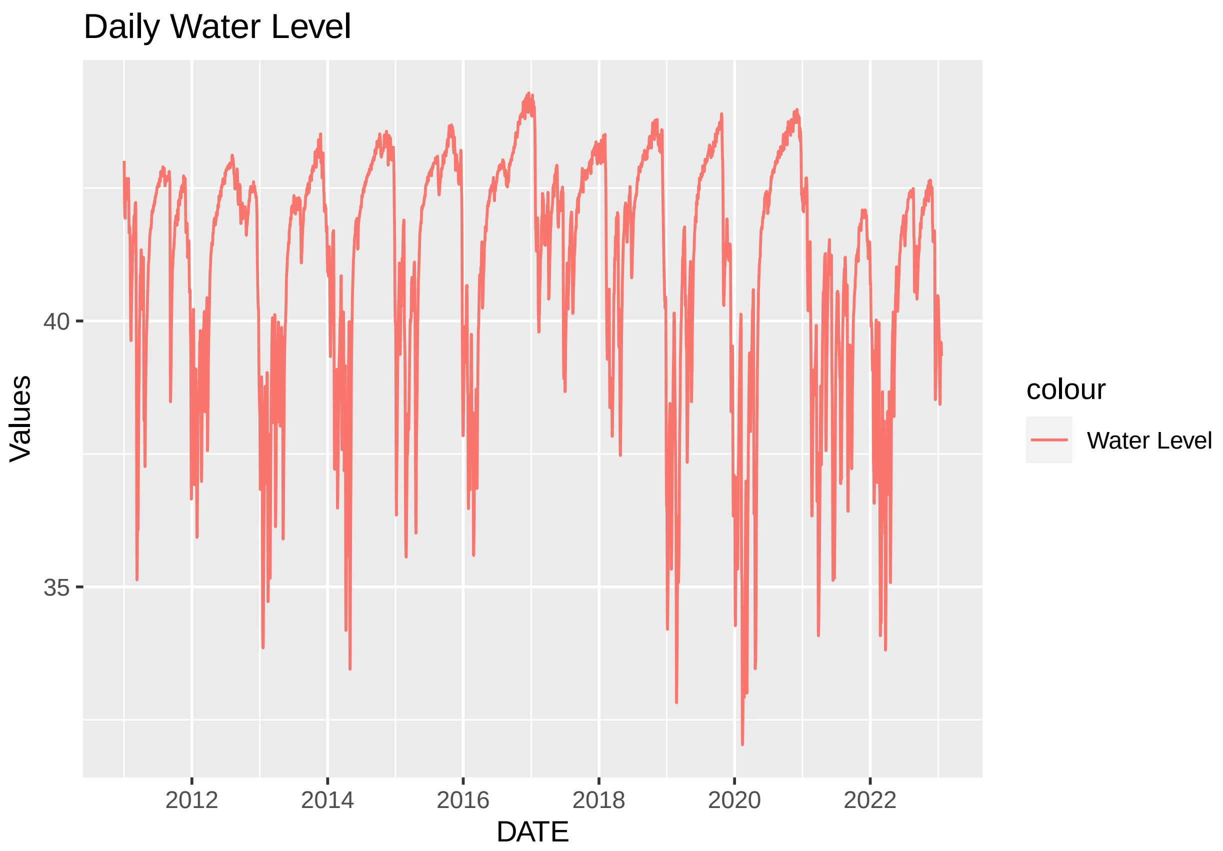
```r
library(ggplot2)
# Analyze rain
ggplot(data, aes(x = precip)) +
  geom_histogram(binwidth = .25, fill = "blue", color = "white") +
  labs(title = "Histogram of Rain", x = "Value", y = "Frequency") +
  theme_minimal()
```



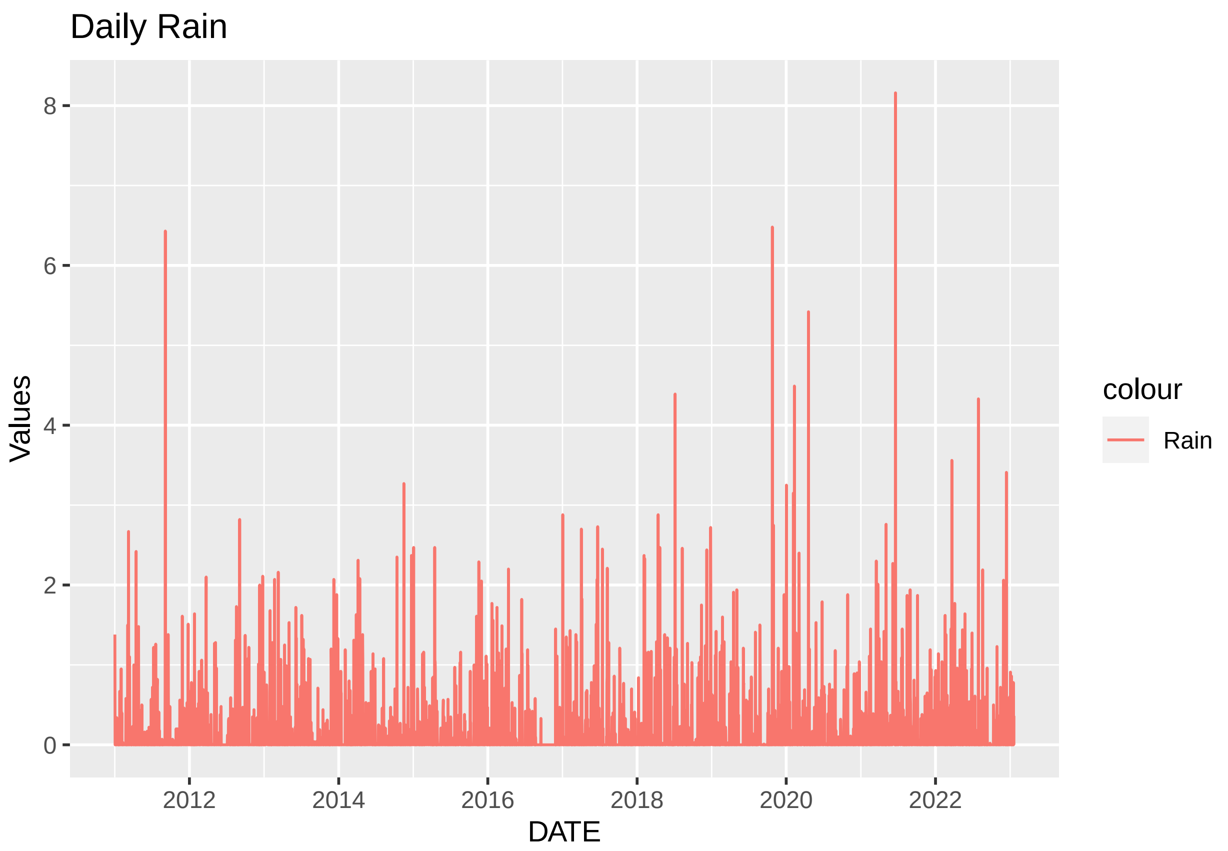
```
summary(data$precip)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   0.0000   0.0000   0.0000   0.1532   0.0500   8.1600
```

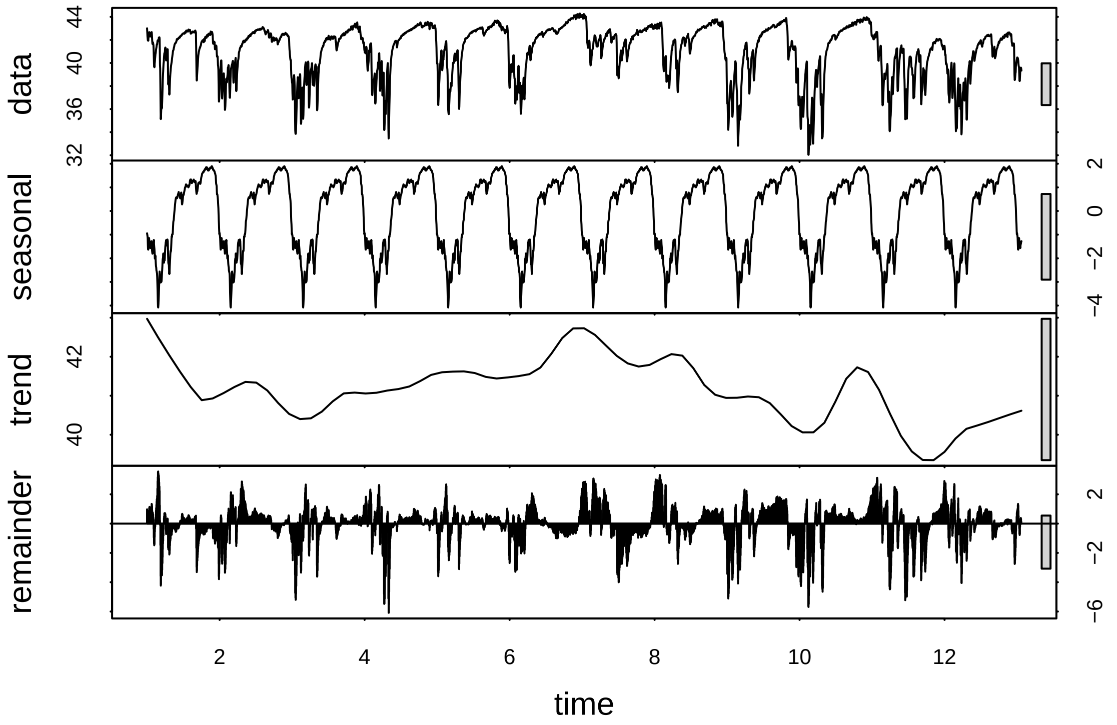
```
library(ggplot2)
ggplot(data, aes(x = DATE)) +
  geom_line(aes(y = waterlevel, color = "Water Level")) +
  labs(y = "Values", title = "Daily Water Level")
```



```
ggplot(data, aes(x = DATE)) +
  geom_line(aes(y = precip, color = "Rain")) +
  labs(y = "Values", title = "Daily Rain")
```



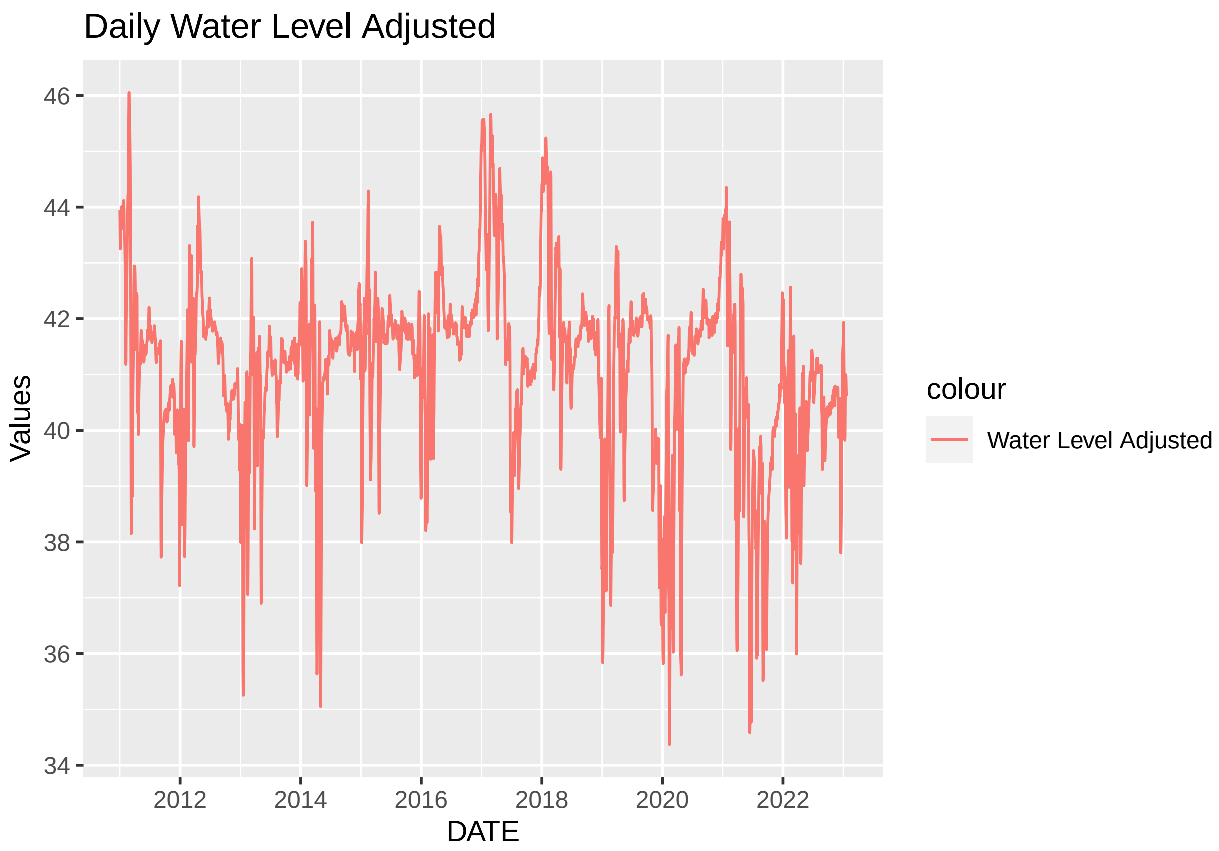
```
decomposition <- stl(ts_water, s.window = "periodic")
plot(decomposition)
```



Decomposition shows no clear trend, but shows a clear sign of seasonality. The first method chosen will choose the seasonality component and subtract it from the the time series data. The seasonality will be added back after the models have been fitted and forcasted.

```
data$water_adj <- data$waterlevel - decomposition$time.series[, "seasonal"]
ts_season <- ts(decomposition$time.series[, "seasonal"])
ts_adj <- ts(data$waterlevel) - ts_season
ts_train_adj <- ts_adj[1:3000]
test_test_adj <- ts_adj[3001:4403]

ggplot(data, aes(x = DATE)) +
  geom_line(aes(y = water_adj, color = "Water Level Adjusted")) +
  labs(y = "Values", title = "Daily Water Level Adjusted")
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



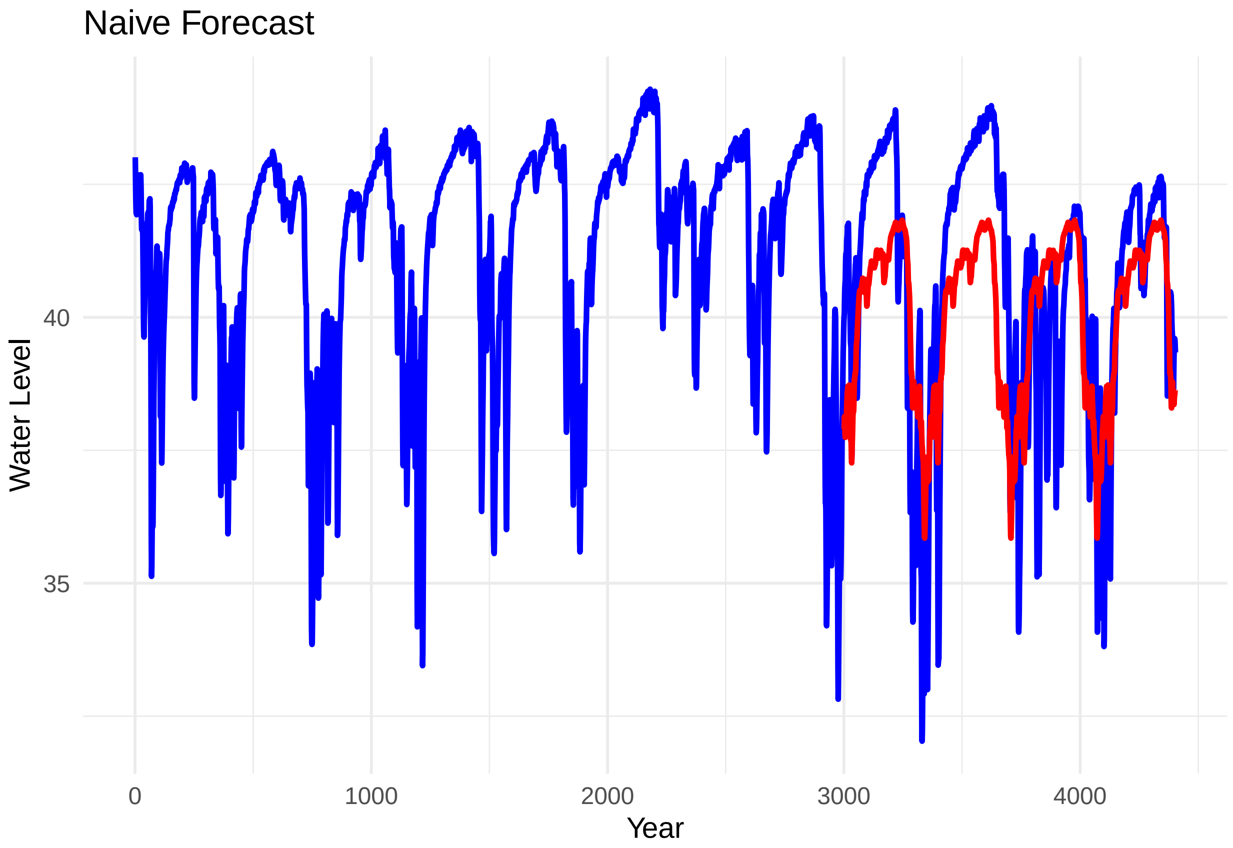
```
# add actual Too
library(forecast)
library(tseries)
```

```
##
## Attaching package: 'tseries'

## The following object is masked from 'package:itsmr':
##
##     arma

## The following objects are masked from 'package:aTSA':
##
##     adf.test, kpss.test, pp.test
```

```
naive_model <- naive( train_data, h = length(test_data))


naive_forecast = forecast(naive_model, h=length(test_data))
naive_forecast$mean = naive_forecast$mean + ts_season[3001:4403]
accuracy(naive_forecast, test_data)
```

```
##                        ME      RMSE       MAE          MPE      MAPE     MASE
## Training set -0.001027009 0.2902465 0.1573791 -0.005376069 0.4011552 1.000000
## Test set      0.544660643 1.8468730 1.5305503  1.122430448 3.8168082 9.725243
##                   ACF1
## Training set 0.7091038
## Test set            NA
```

```
#plot(train_data, col="blue", xlab="Year", ylab="waterlevel", main="naive forecast",type='l')
#lines(naive_forecast$mean, col="red", lwd=2)
# Convert data to data frames for ggplot
train_df <- data.frame(Time = time(data$DATE), WaterLevel = as.numeric(data$waterlevel))
forecast_df <- data.frame(Time = time(naive_forecast$mean), Forecast = as.numeric(naive_forecast$mean))

ggplot() +
  geom_line(data = train_df, aes(x = Time, y = WaterLevel), color = "blue", size = 1) +
  geom_line(data = forecast_df, aes(x = Time, y = Forecast), color = "red", size = 1) +
  ggtitle("Naive Forecast") +
  xlab("Year") + ylab("Water Level") +
  theme_minimal()
```



ARIMA or ARMA will be used throguh auto.arima function. Is it stationary

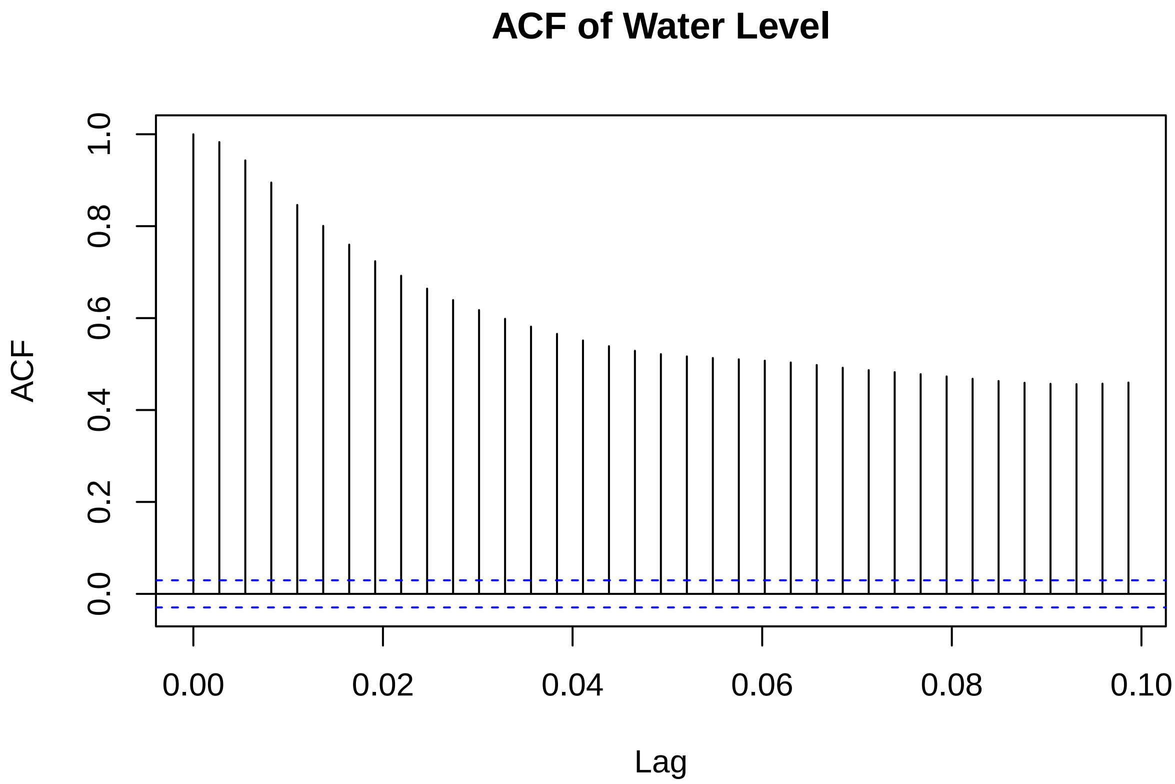
```
library(tseries)
adf_test <- adf.test(data$water_adj)
print(adf_test)
```

```
##
```

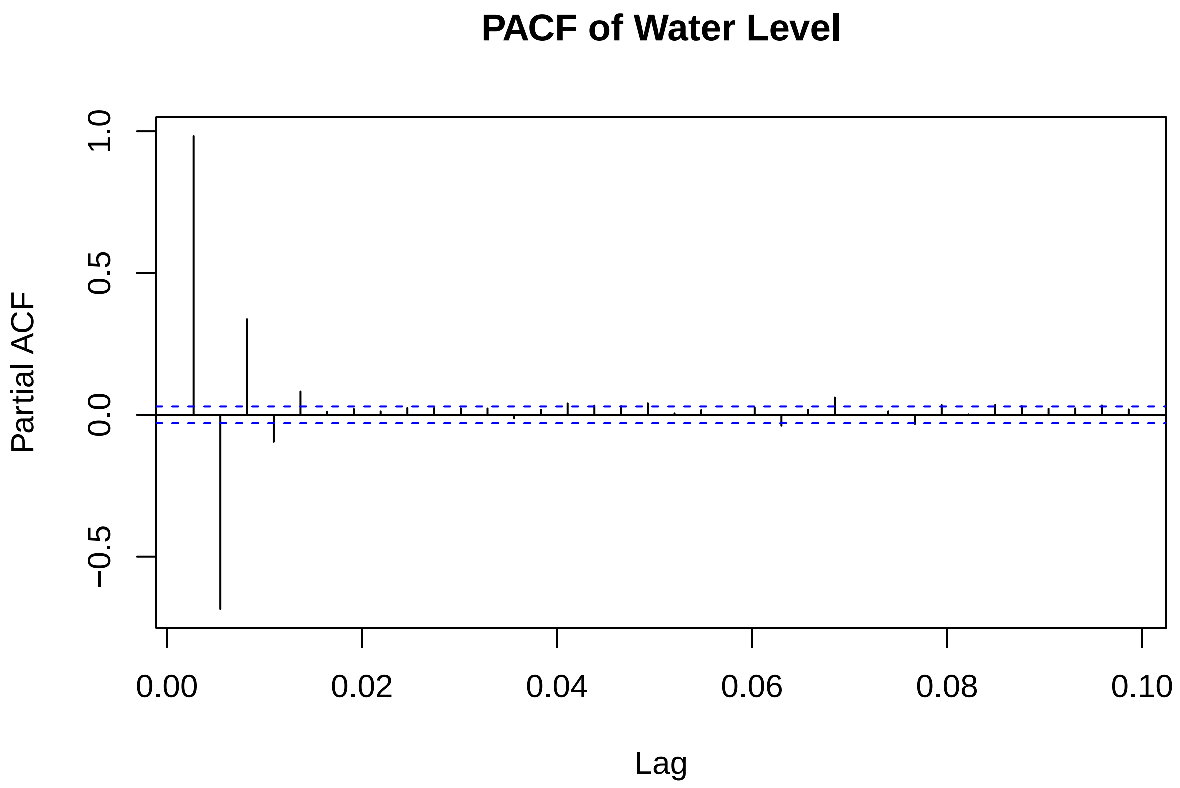
```
##  Augmented Dickey-Fuller Test
##
## data:  data$water_adj
## Dickey-Fuller = -7.0412, Lag order = 16, p-value = 0.01
## alternative hypothesis: stationary
```

Using the Augmented Dickey-Fuller Test shows the seasonal differenced time series is stationary.

```
acf(data$water_adj, main = "ACF of Water Level")
```



```
pacf(data$water_adj, main = "PACF of Water Level")
```



## Rain as exogenous regressor This model is not intended for forecasting although it could be with conditional forecasting. This model is intended to answer whether or not rain has an effect on the waterlevel of the well.

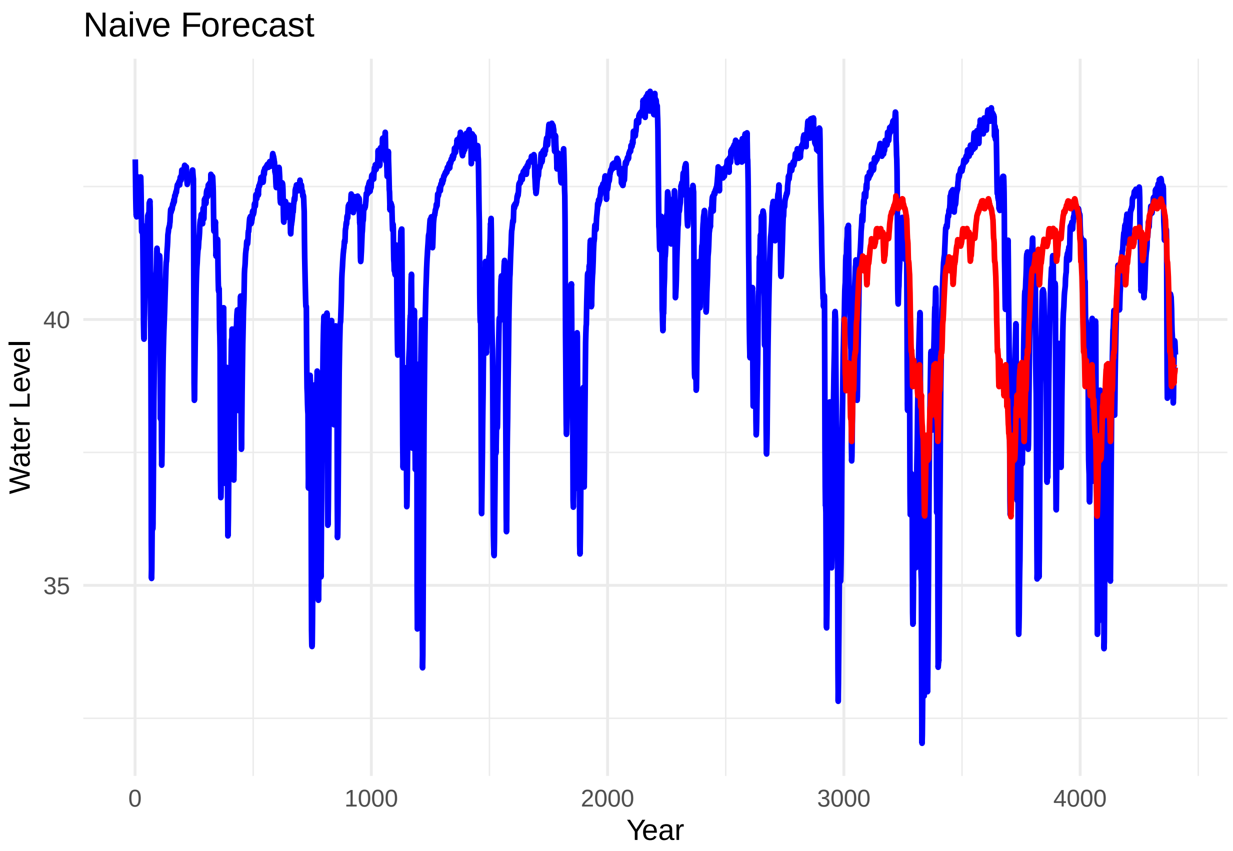
```
library(forecast)
xreg_train = data$precip[1:3000]
xreg_test = data$precip[3001:4403]

# ARIMA model with Rain as an external regressor

arima_with_rain <- auto.arima(ts_train_adj, xreg = xreg_train)
summary(arima_with_rain)
```

```
## Series: ts_train_adj
## Regression with ARIMA(3,1,3) errors
##
## Coefficients:
##          ar1     ar2      ar3     ma1      ma2      ma3    xreg
##       0.5368  0.8025  -0.4894  0.5078  -0.8939  -0.4897  0.0202
## s.e.  0.0446  0.0612   0.0303  0.0446   0.0235   0.0282  0.0030
##
## sigma^2 = 0.02913:  log likelihood = 1049.72
## AIC=-2083.43   AICc=-2083.38   BIC=-2035.38
##
## Training set error measures:
##                          ME     RMSE       MAE          MPE       MAPE       MASE
## Training set -0.001384547 0.170437 0.1013414 -0.004249033 0.2471194 0.6484022
##                      ACF1
## Training set -0.004188554
```

```
arima_forecast = forecast(arima_with_rain, h=length(test_data), xreg= xreg_test)
arima_forecast$mean = arima_forecast$mean + ts_season[3001:4403]


accuracy(arima_forecast, test_data)
```

```
##                        ME     RMSE       MAE          MPE      MAPE      MASE
## Training set -0.001384547 0.170437 0.1013414 -0.004249033 0.2471194 0.6484022
## Test set      0.091790214 1.761491 1.3925850  0.001758671 3.5198301 8.9100294
##                      ACF1
## Training set -0.004188554
## Test set               NA
```

```
train_df <- data.frame(Time = time(data$DATE), WaterLevel = as.numeric(data$waterlevel))
forecast_df <- data.frame(Time = time(arima_forecast$mean), Forecast = as.numeric(arima_forecast$mean))

ggplot() +
  geom_line(data = train_df, aes(x = Time, y = WaterLevel), color = "blue", size = 1) +
  geom_line(data = forecast_df, aes(x = Time, y = Forecast), color = "red", size = 1) +
  ggtitle("Naive Forecast") +
  xlab("Year") + ylab("Water Level") +
  theme_minimal()
```



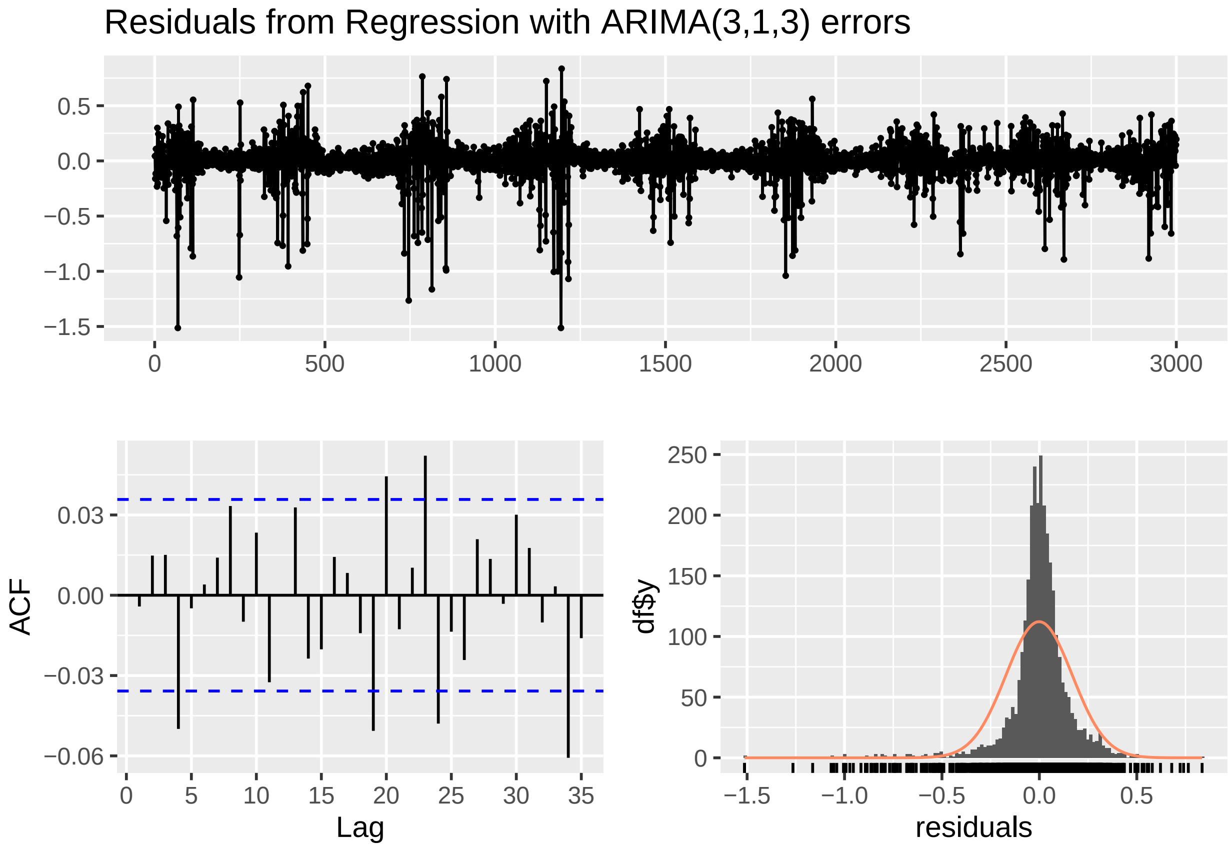
Are baseline model is the naive model with seasonal adjustment which has a mase test 9.7. This arima model has a better mase.

```
# computing p values
model <- arima(ts_train_adj, order = c(3, 1, 3), xreg = xreg_train)

coefs <- model$coef
se <- sqrt(diag(model$var.coef))


t_values <- coefs / se

# Computing p-values (two-tailed test)
p_values <- 2 * (1 - pt(abs(t_values), df = length(ts_train_adj) - length(coefs)))


results <- data.frame(Coefficient = coefs, Std_Error = se, t_value = t_values, p_value = p_values)
print(results)
```

```
##            Coefficient   Std_Error    t_value     p_value
## ar1         0.53682368 0.044579556  12.041925 0.00000e+00
## ar2         0.80251846 0.061178225  13.117714 0.00000e+00
## ar3        -0.48939527 0.030313298 -16.144574 0.00000e+00
## ma1         0.50779220 0.044599431  11.385621 0.00000e+00
## ma2        -0.89391725 0.023470172 -38.087376 0.00000e+00
## ma3        -0.48972426 0.028217916 -17.355083 0.00000e+00
## xreg_train  0.02024588 0.002986977   6.778051 1.46132e-11
```

All p-values are statistically signficant.

```
checkresiduals(arima_with_rain)
```


Residuals from Regression with ARIMA(3,1,3) errors

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(3,1,3) errors
## Q* = 14.892, df = 4, p-value = 0.00493
##
## Model df: 6.   Total lags used: 10
```

For a first model it performs fairly well but suffers from a non-constant variance in residuals. My best guess is that the winter period, which is more volatile, is contributing to that.

Forecasting models

ARIMA

```
#ARMA without rain for forecasting
arima_no_rain <- auto.arima(ts_train_adj)
summary(arima_no_rain)
```

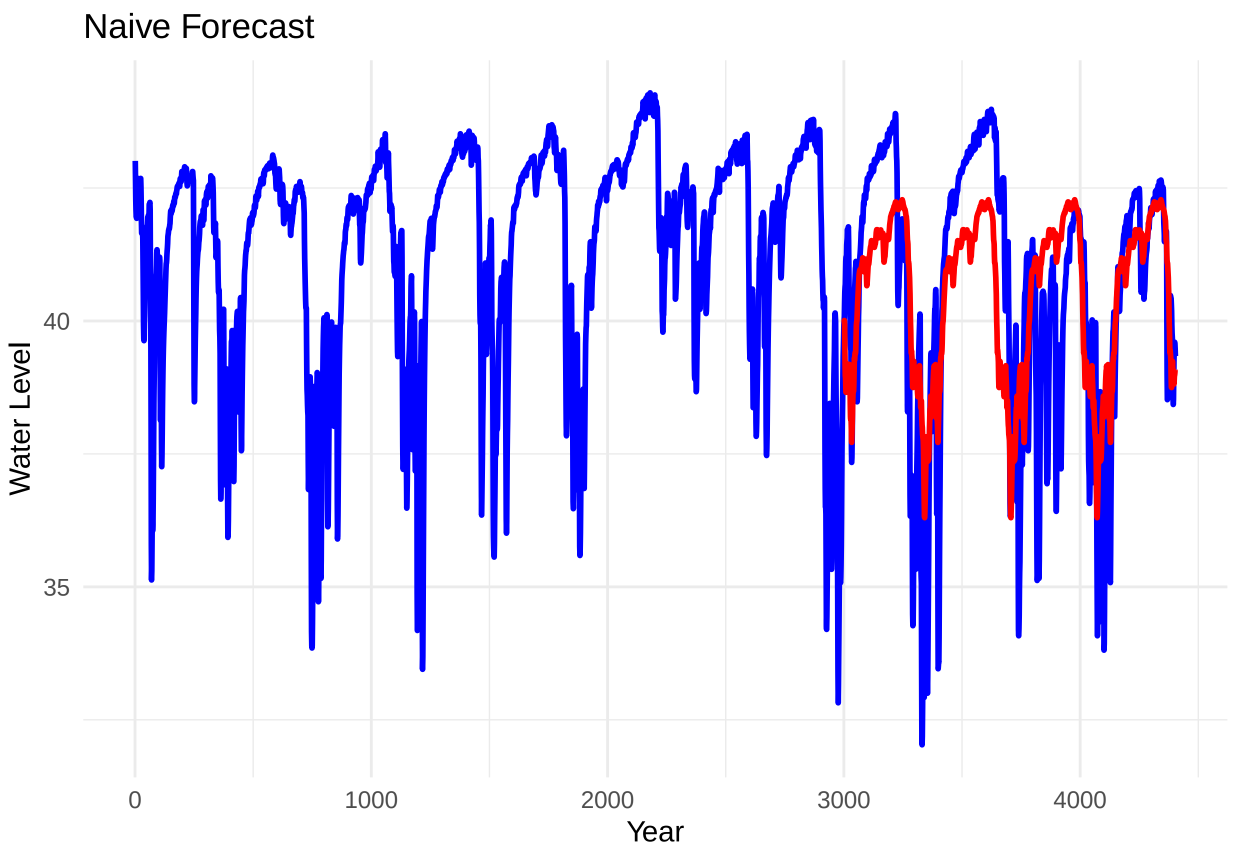
```
## Series: ts_train_adj
## ARIMA(3,1,2)
##
```

```
## Coefficients:
##          ar1      ar2      ar3      ma1      ma2
##       1.4455  -0.5200  -0.0056  -0.4020  -0.5319
## s.e.  0.0407   0.0676   0.0346   0.0363   0.0311
##
## sigma^2 = 0.02962:  log likelihood = 1023.52
## AIC=-2035.03   AICc=-2035   BIC=-1999
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.00139717 0.1719325 0.1005782 -0.00431623 0.2453091 0.6435192
##                      ACF1
## Training set 0.0001403817
```

```r
arima_forecast = forecast(arima_no_rain, h=length(test_data))
arima_forecast$mean = arima_forecast$mean + ts_season[3001:4403]


accuracy(arima_forecast, test_data)
```

```
##                        ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.00139717 0.1719325 0.1005782 -0.00431623 0.2453091 0.6435192
## Test set       0.08490323 1.7607655 1.3911515 -0.01517318 3.5169191 8.9008573
##                      ACF1
## Training set 0.0001403817
## Test set               NA
```

```r
train_df <- data.frame(Time = time(data$DATE), WaterLevel = as.numeric(data$waterlevel))
forecast_df <- data.frame(Time = time(arima_forecast$mean), Forecast = as.numeric(arima_forecast$mean))

ggplot() +
  geom_line(data = train_df, aes(x = Time, y = WaterLevel), color = "blue", size = 1) +
  geom_line(data = forecast_df, aes(x = Time, y = Forecast), color = "red", size = 1) +
  ggtitle("Naive Forecast") +
  xlab("Year") + ylab("Water Level") +
  theme_minimal()
```



```
# computing p values
model <- arima(ts_train_adj, order = c(2, 1, 2))

coefs <- model$coef
se <- sqrt(diag(model$var.coef))


t_values <- coefs / se

# Computing p-values (two-tailed test)
p_values <- 2 * (1 - pt(abs(t_values), df = length(ts_train_adj) - length(coefs)))


results <- data.frame(Coefficient = coefs, Std_Error = se, t_value = t_values, p_value = p_values)
print(results)
```

```
##     Coefficient  Std_Error   t_value p_value
## ar1   1.4514396 0.01918403  75.65875       0
## ar2  -0.5307235 0.01862893 -28.48920       0
## ma1  -0.4070618 0.01946020 -20.91766       0
## ma2  -0.5277155 0.01947715 -27.09408       0
```

The AR3 was not statistically significant and was removed.

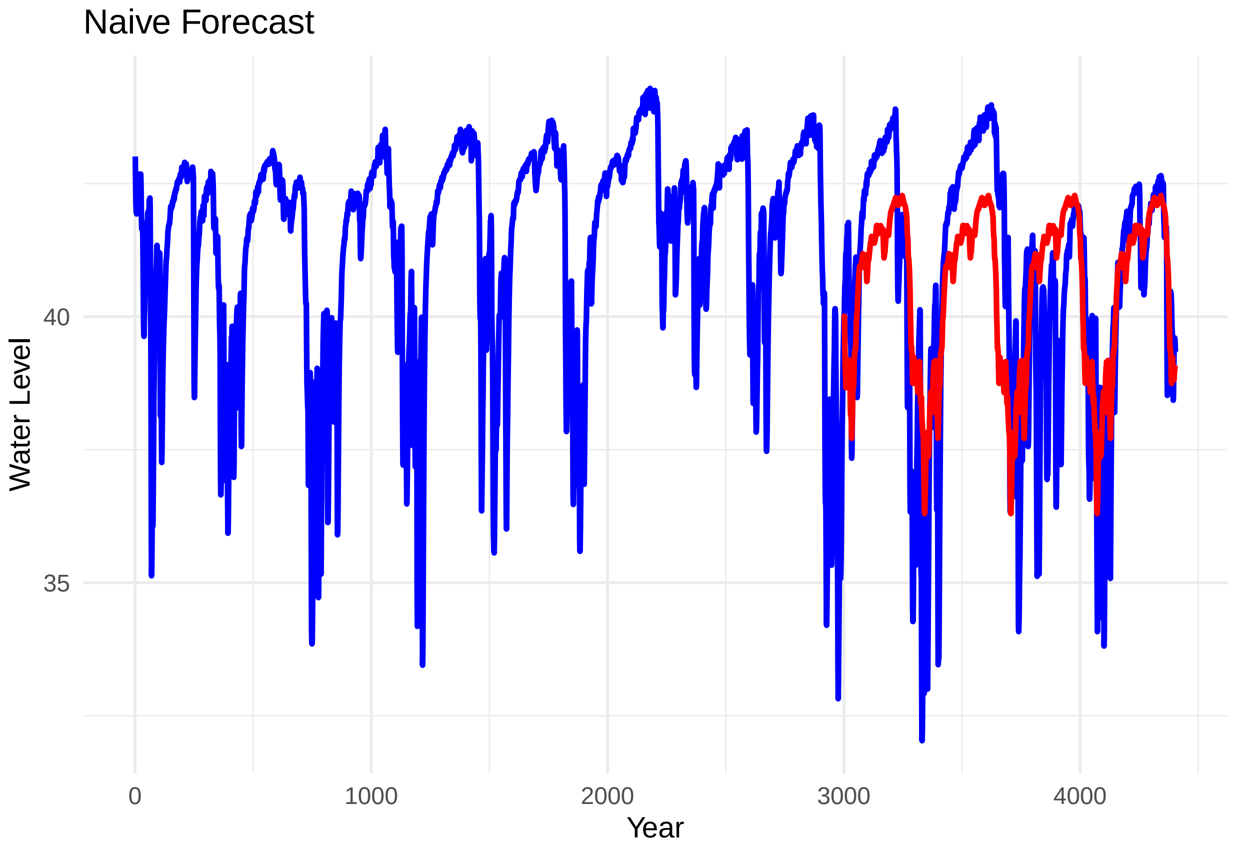
```
summary(model)
```

```
##
## Call:
## arima(x = ts_train_adj, order = c(2, 1, 2))
##
## Coefficients:
##          ar1      ar2      ma1      ma2
##       1.4514  -0.5307  -0.4071  -0.5277
## s.e.  0.0192   0.0186   0.0195   0.0195
##
## sigma^2 estimated as 0.02957:  log likelihood = 1023.5,  aic = -2037.01
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE      MAPE      MASE
## Training set -0.001402732 0.1719332 0.1005766 -0.00433278 0.2453051 0.6435087
##                       ACF1
## Training set -0.0007693952
```

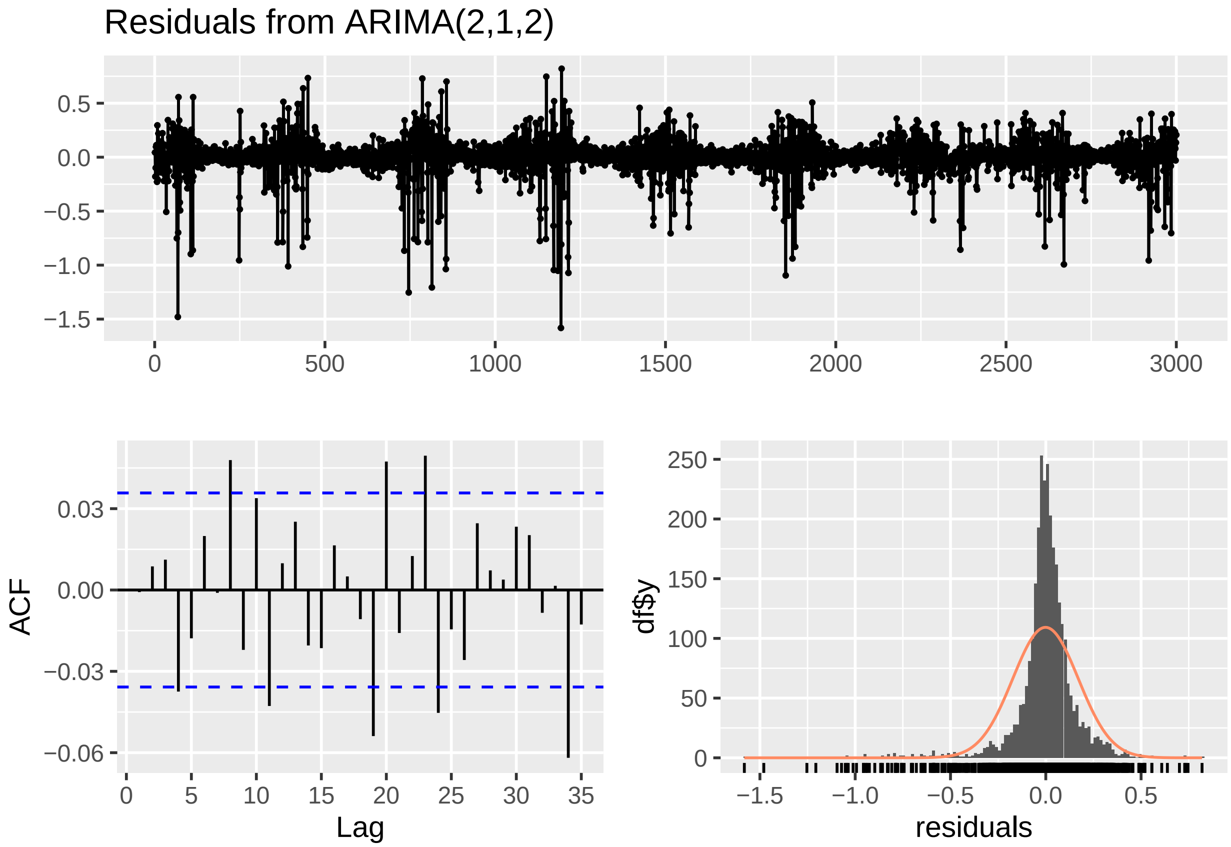
```
arima_forecast = forecast(model, h=length(test_data))
arima_forecast$mean = arima_forecast$mean + ts_season[3001:4403]


accuracy(arima_forecast, test_data)
```

```
##                        ME      RMSE       MAE          MPE      MAPE      MASE
## Training set -0.001402732 0.1719332 0.1005766 -0.004332780 0.2453051 0.6435087
## Test set      0.087444319 1.7607914 1.3915512 -0.008881844 3.5176518 8.9034147
##                       ACF1
## Training set -0.0007693952
## Test set                NA
```

```
train_df <- data.frame(Time = time(data$DATE), WaterLevel = as.numeric(data$waterlevel))
forecast_df <- data.frame(Time = time(arima_forecast$mean), Forecast = as.numeric(arima_forecast$mean))

ggplot() +
  geom_line(data = train_df, aes(x = Time, y = WaterLevel), color = "blue", size = 1) +
  geom_line(data = forecast_df, aes(x = Time, y = Forecast), color = "red", size = 1) +
  ggtitle("Naive Forecast") +
  xlab("Year") + ylab("Water Level") +
  theme_minimal()
```



```
checkresiduals(model)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2)
## Q* = 18.804, df = 6, p-value = 0.004508
##
## Model df: 4.   Total lags used: 10
```

Given the seasonal naive model has a testing MASE of 9.7, it can be concluded that seasonally adjust ARIMA model with the order (Auto Regressive order of 2, Integration order of 1, Moving Average order of 2) is a better model. This model does suffer from a non constant variance of residuals. Examining the residuals of the model shows that the residuals are

Conclusion

The analysis shows that their is seasonality but no trend. It also shows that rain does effect well water height, increasing the well water height by 0.02 inches for every inch of rain. The ARIMA model created is suitable for forecasting but suffers non constant variance of residuals, periods of higher inaccuracy. ARCH and GARCH may need to be applied to address this.