AI6123 Time Series Analysis
Assignment 2
Zheng, Weixiang
G2103278G

Load Data

In the first section, we imoprt the necessay libraries and initialize the data variable. We use `read.delim` function to load the data from `drug.txt` and print out the beginning year and ending year. Because the first entry is headers, we skip the first entry by specifying `header=TRUE`.

```
library(lubridate)
library(forecast)
library(ggplot2)

drug_data = read.delim("./drug.txt",header=TRUE, sep=",")
begin = ymd(as.Date(drug_data$date[1]))
begin
```

```
## [1] "1991-07-01"
```

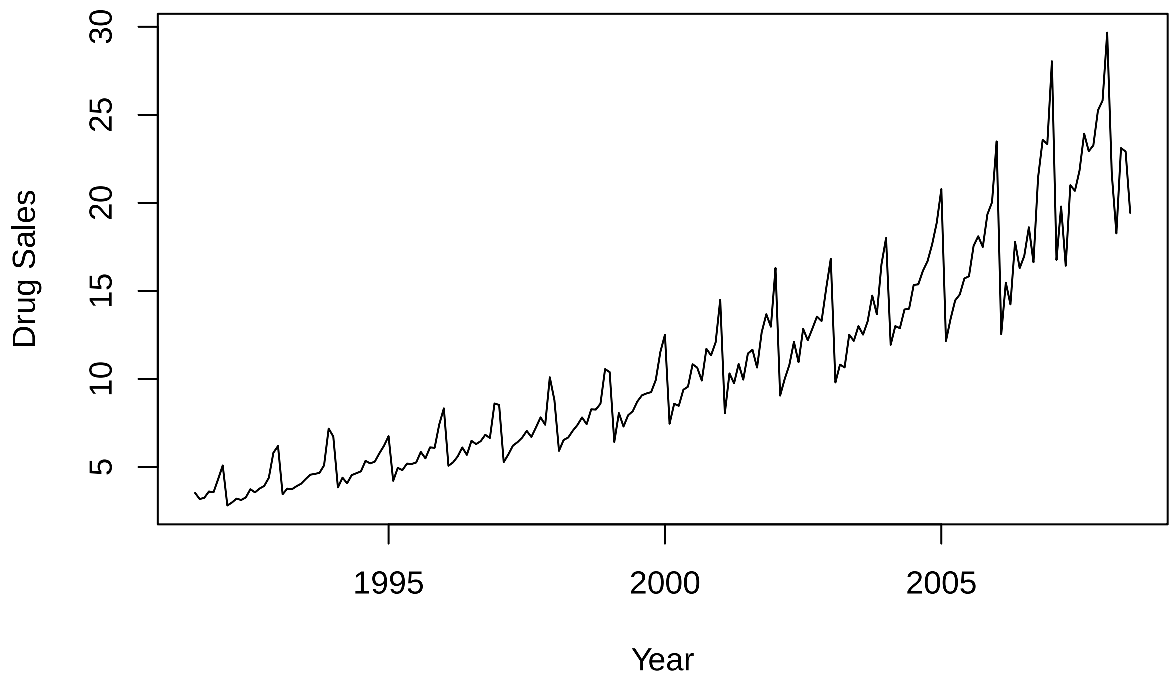
```
end = ymd(as.Date(drug_data$date[nrow(drug_data)]))
end
```

```
## [1] "2008-06-01"
```

Generate time series data

Then in the second section, we generate the time series data based on the data we read from the text file. We further plot the time series in order to see the trend more clearly.

```
time_series = ts(drug_data$value, start=c(year(begin), month(begin)),
                 end=c(year(end) ,month(end)), frequency=12)
plot(time_series,xlab="Year",ylab="Drug Sales")
```

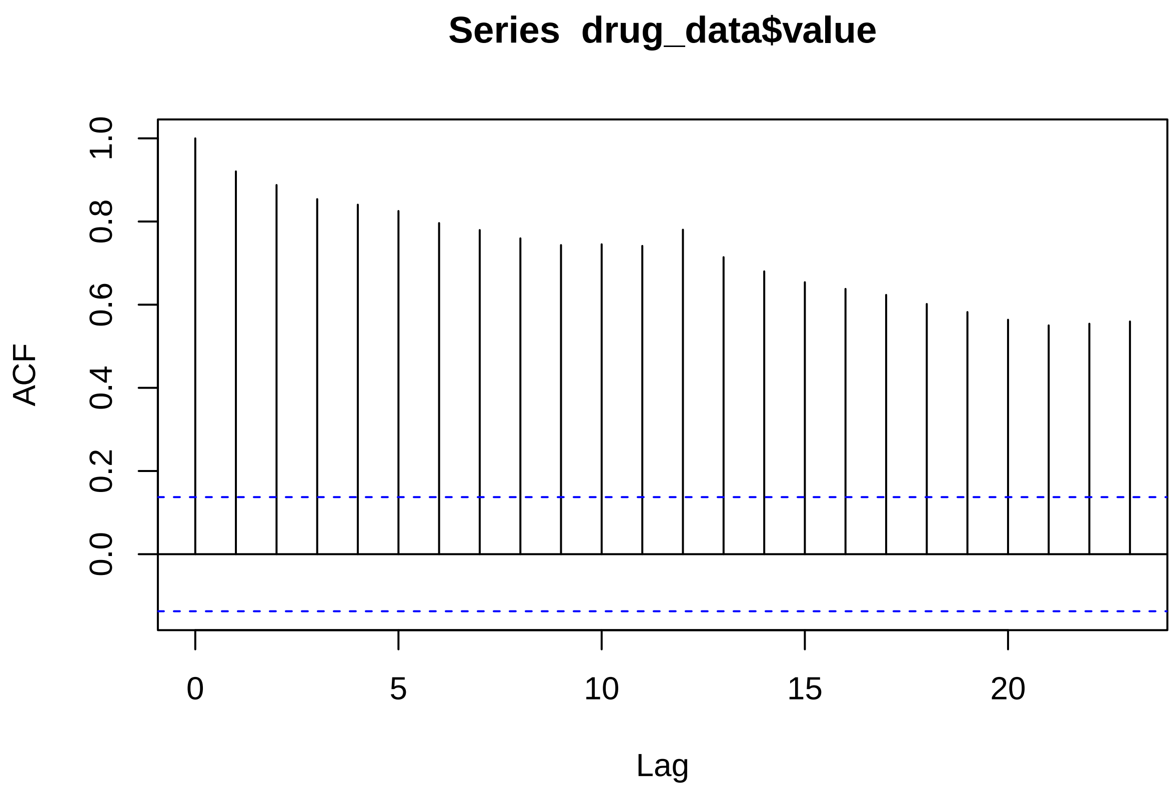


From the plot, we can see very clearly that there exist a obvious upward trend as well as a seasonal pattern.

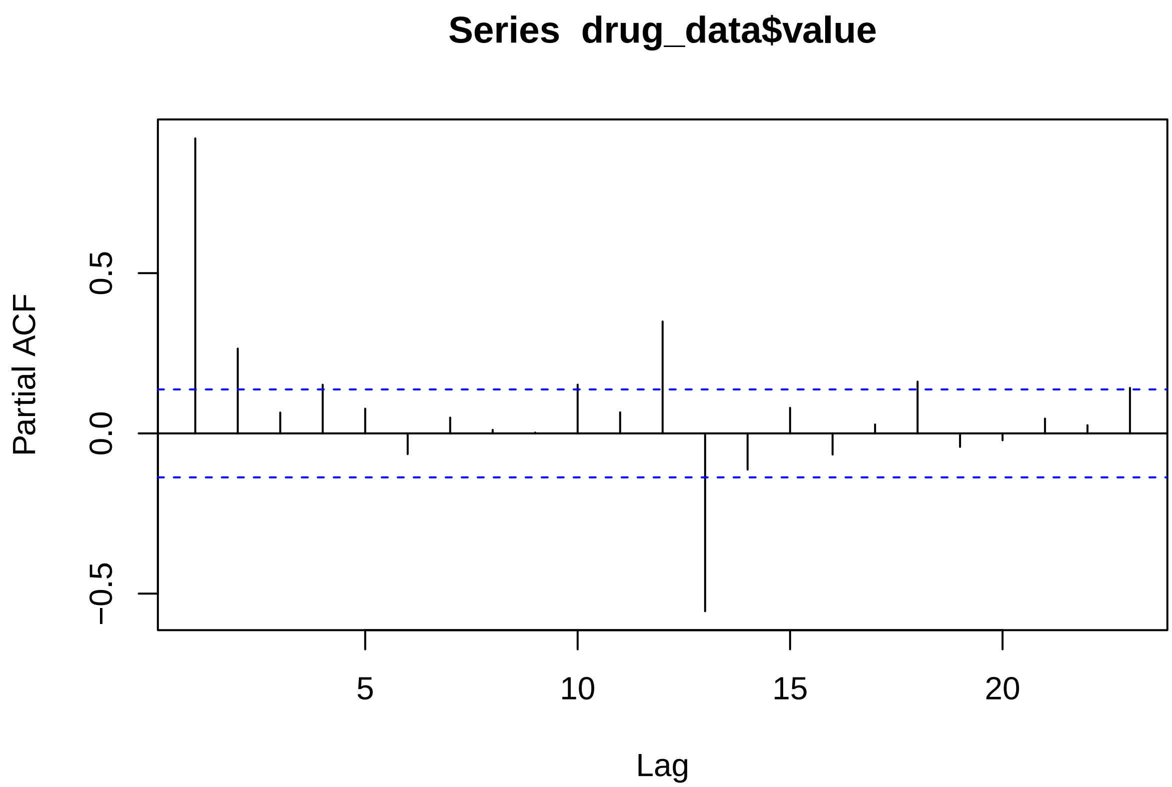
ACF Plot and PACF Plot

In the third section, we plot out the ACF Plot and PACF Plot in order to see whether the time series is stationary or non-stationary.

```
acf(drug_data$value)
```




```
pacf(drug_data$value)
```

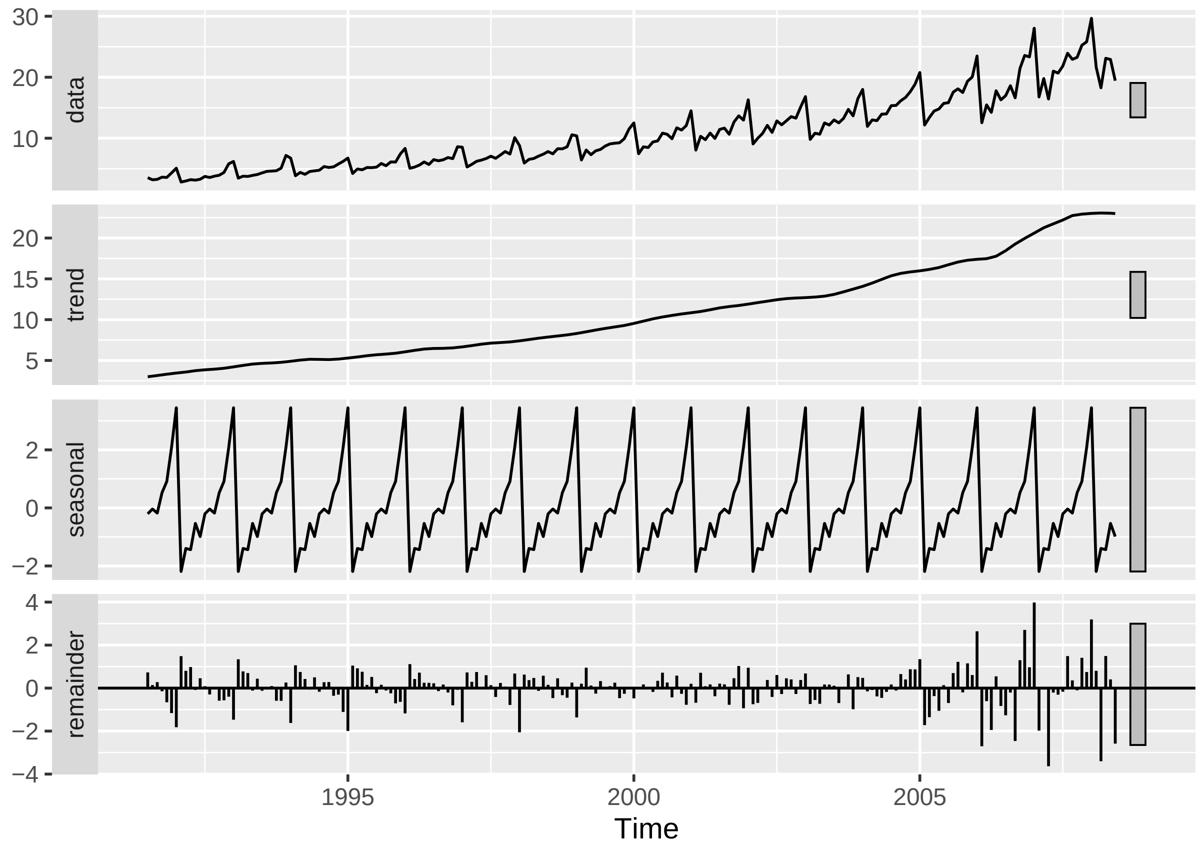


We can see clearly that the ACF dies down slowly, which means the data is non-stationary.

Decompose the seasonal component

In this section, we visualize the different components of the time series in terms of trend, seasonality and remainder.

```
# Seasonal Component
stlresult = stl(time_series, s.window = "periodic")
autoplot(stlresult)
```



We can see from the plot that indeed the original time series consist of an upward trend, a seasonal component and the remainder is very close to white noise.

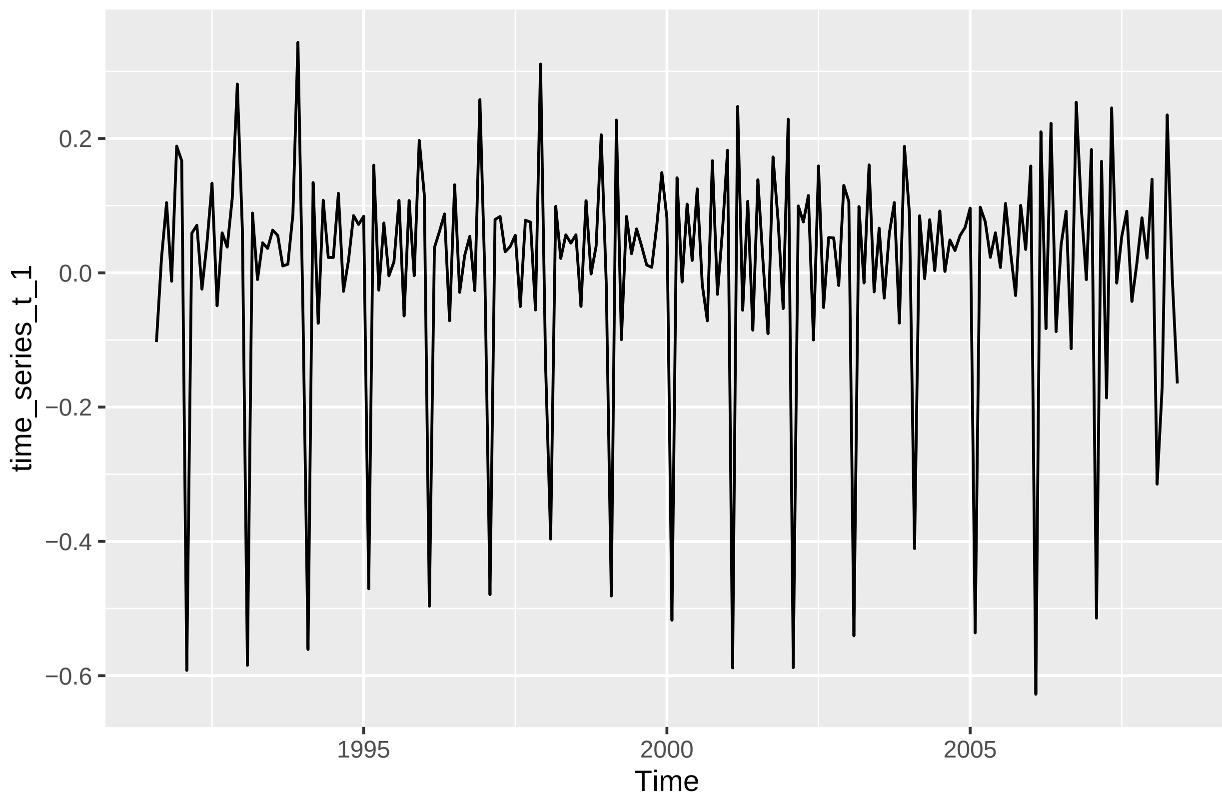
Data Transformation

In this section, we transform the time series into a more easy-to-handle format. The Box-cox transformation is to transform a non-stationary time series into a normal shape, i.e. to stabilize the variance. We pick the lambda to be zero, then it is same as applying a log function on every data point of the time series.

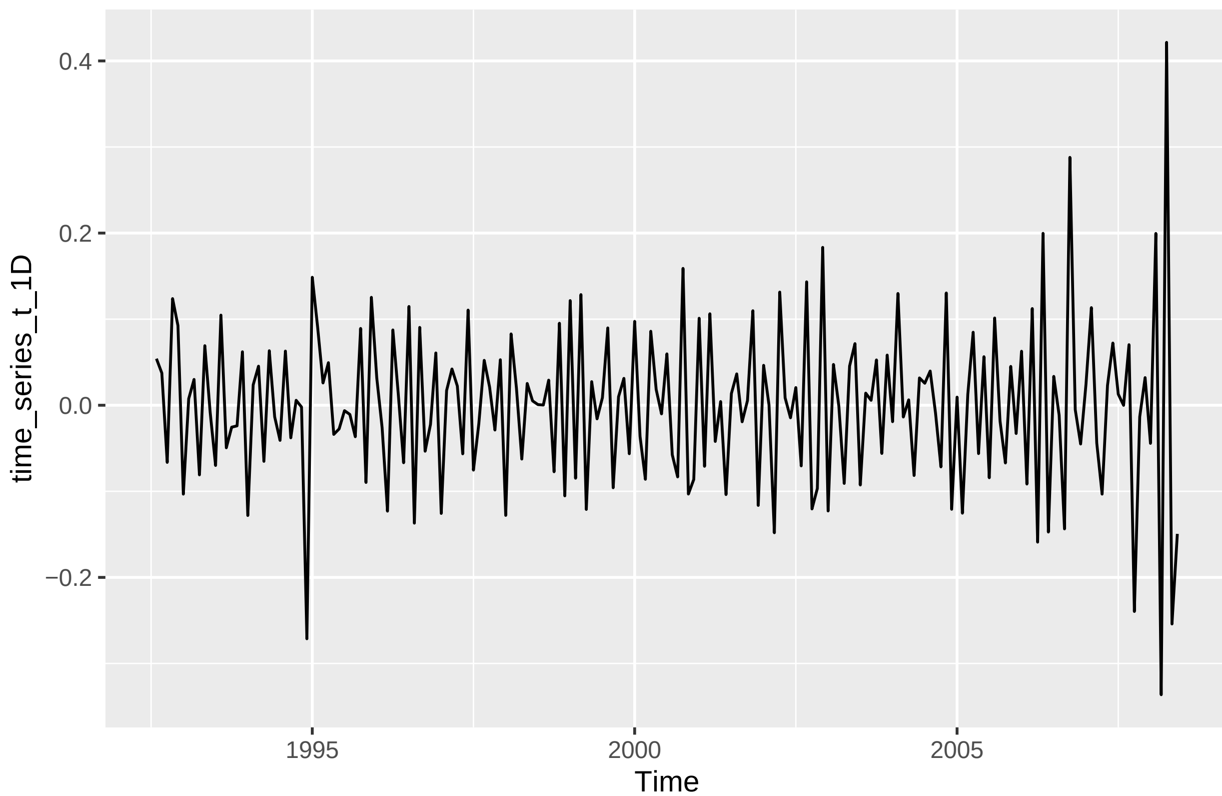
```
# Data Transformation
time_series_t = BoxCox(time_series, 0)
```

Next, we want to see if we remove the trend and seasonality, is the time series going to be white noise shape.

```
# Remove trend
time_series_t_1 = diff(time_series_t, differences = 1)
autoplot(time_series_t_1)
```




```r
# Remove seasonality
time_series_t_1D = diff(time_series_t_1, lag = 12)
autoplot(time_series_t_1D)
```



We can see from the plots that after we apply one time differencing, we remove the upward trend, the plot shows a seasonal fluctuation, then we apply another differencing with lag 12 to remove seasonality, the plot shows a pure random noise fluctuation. We indeed confirm our hypothesis, the time seies is made of a upward trend, a seaonal componenet and a white noise.

Model

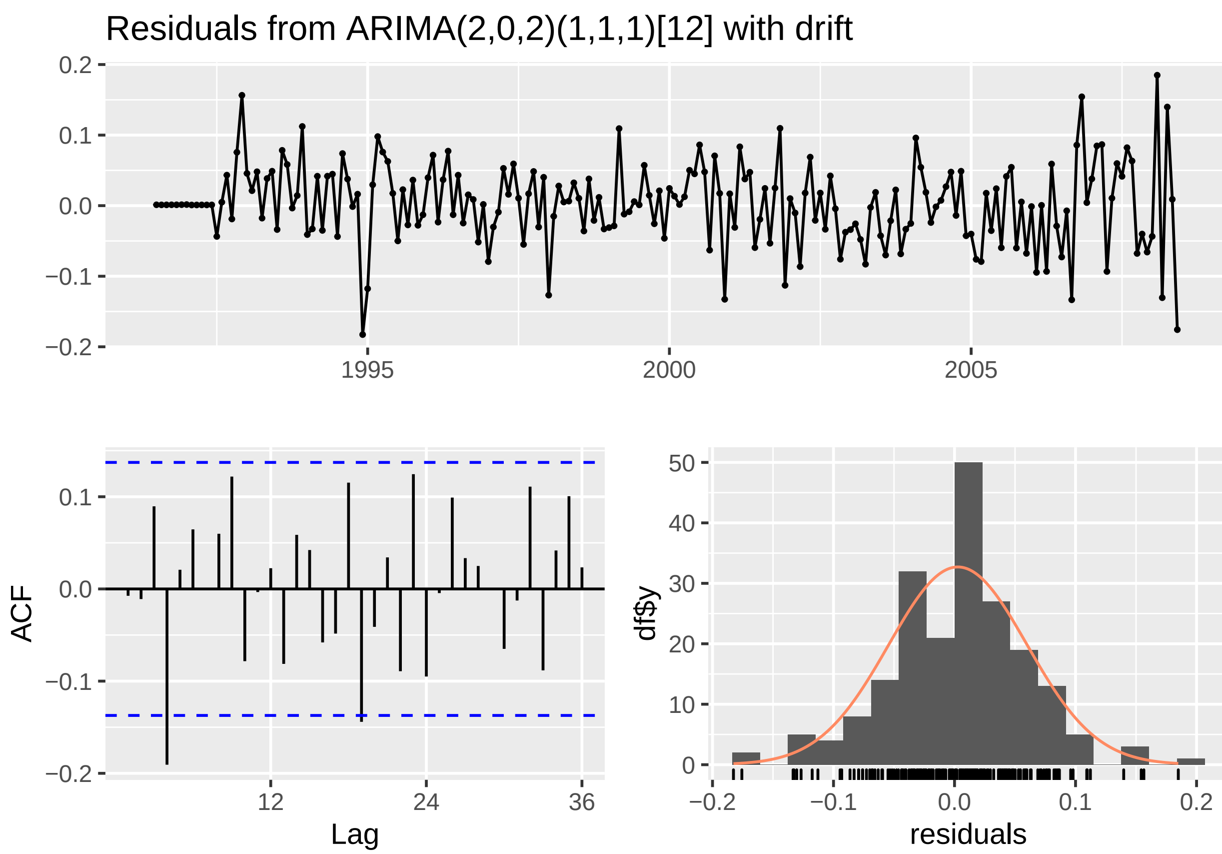
In this section, we fit two moedels to our data, the first one is an seasonal ARIMA model, that is, we want to find a SARIMA(p,d,q,P,D,Q) that could fit the data well. Since there are 6 parameters and there exist many different trails of different combinations of the parameters, I will use the autoarima function to fit. But the manual trail of different parameters will be the same as the process in Assignment 1.

```
# Seasonal ARIMA Model
autoarima = auto.arima(time_series_t)
autoarima
```

```
## Series: time_series_t
## ARIMA(2,0,2)(1,1,1)[12] with drift
##
## Coefficients:
##          ar1     ar2      ma1     ma2    sar1     sma1   drift
##       0.7211  0.1081  -0.6809  0.1198  0.2236  -0.8464  0.0095
## s.e.  0.2364  0.2064   0.2316  0.1752  0.1112   0.0835  0.0003
##
## sigma^2 = 0.003595:  log likelihood = 266.01
## AIC=-516.02   AICc=-515.24   BIC=-489.96
```

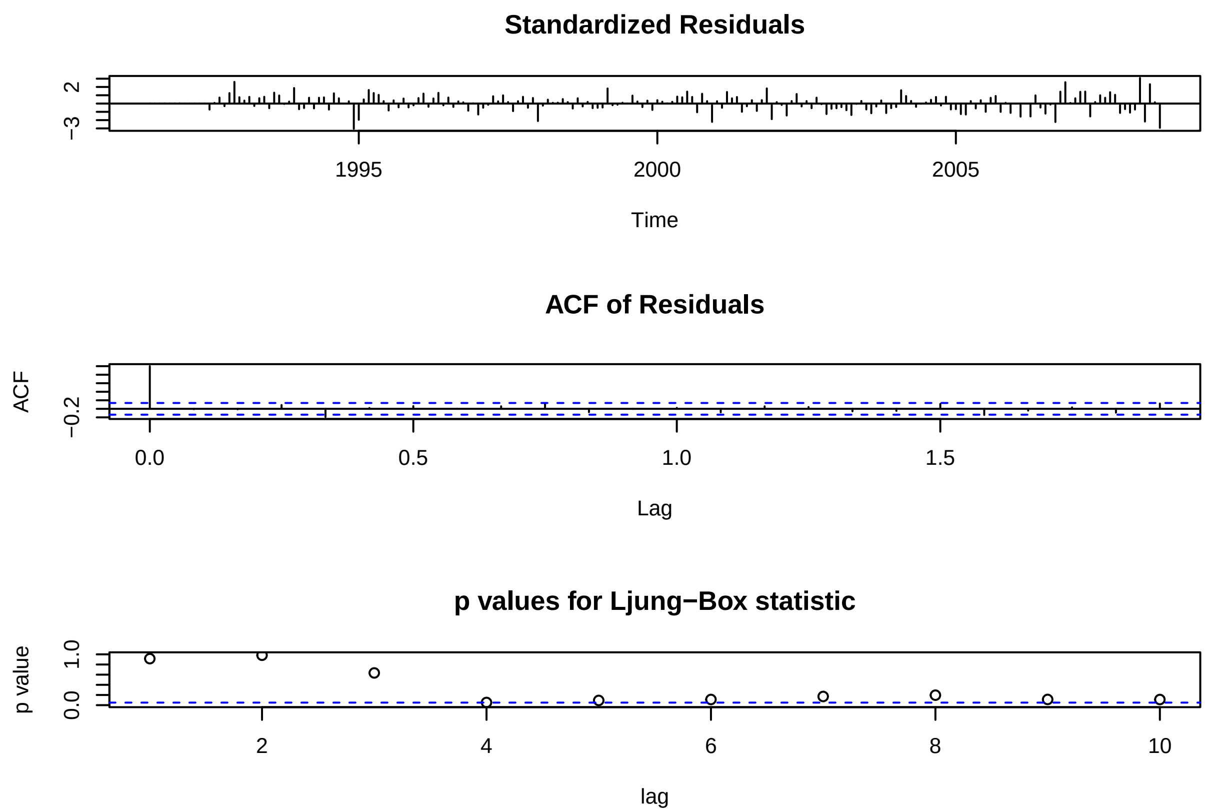
The autoarima function gives us ARIMA(2,0,2)(1,1,1)[12] with drift. Let us run a few diagnoistics to see how good is it.

```
# Diagnostic Checking
checkresiduals(autoarima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,2)(1,1,1)[12] with drift
## Q* = 35.546, df = 17, p-value = 0.005271
##
## Model df: 7.   Total lags used: 24
```

```
tsdiag(autoarima)
```




```
summary(autoarima)
```

```
## Series: time_series_t
## ARIMA(2,0,2)(1,1,1)[12] with drift
##
## Coefficients:
##          ar1     ar2      ma1     ma2    sar1     sma1   drift
##       0.7211  0.1081  -0.6809  0.1198  0.2236  -0.8464  0.0095
## s.e.  0.2364  0.2064   0.2316  0.1752  0.1112   0.0835  0.0003
##
## sigma^2 = 0.003595:  log likelihood = 266.01
## AIC=-516.02   AICc=-515.24   BIC=-489.96
##
## Training set error measures:
##                       ME       RMSE        MAE       MPE     MAPE      MASE
## Training set 0.002771052 0.05709596 0.04329847 0.2056961 1.959566 0.3509022
##                      ACF1
## Training set -0.007416513
```

This particular ARIMA model seems a adequate fit for the following reasons: 1. The p-value is larger than 0.005 2. The residual shows a random shape which means white noise
3. There is a spike in ACF, but it seems not very significant, we will deal with it as a outlier

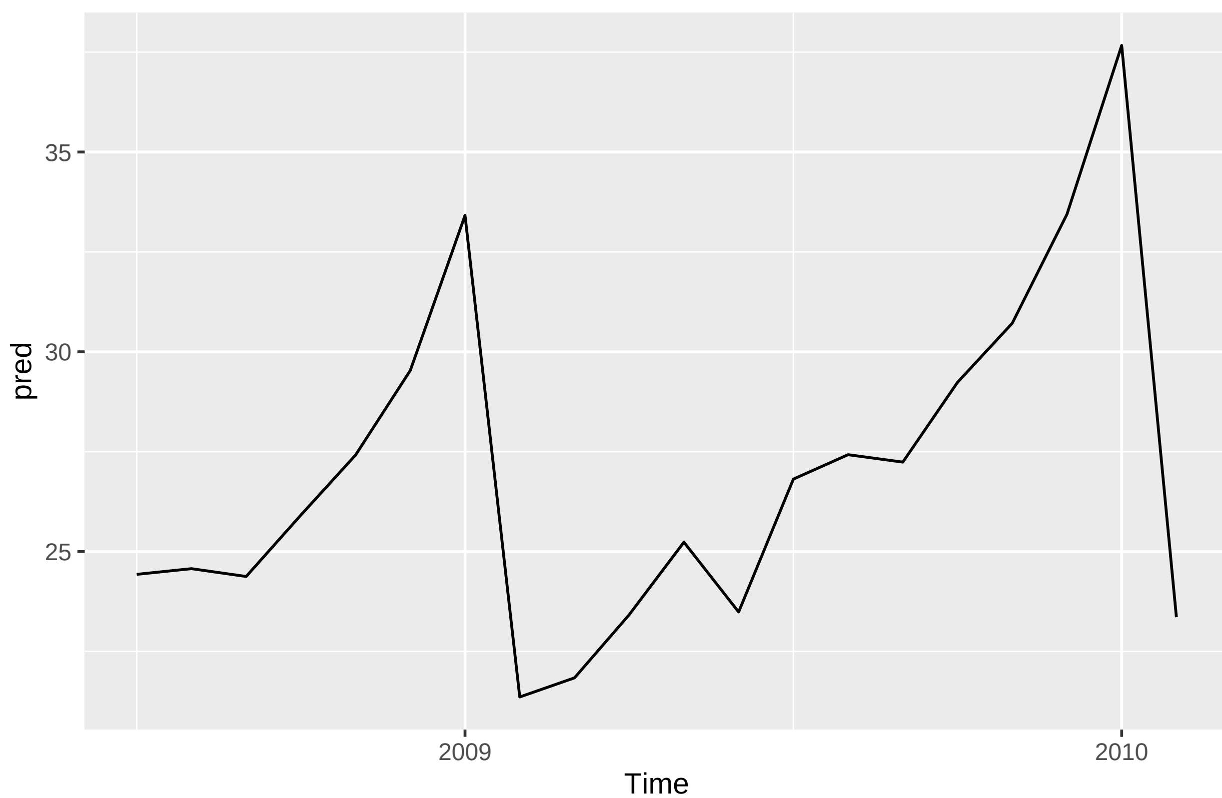
Forecast using ARIMA(2,0,2)(1,1,1)[12]

Next we are going to predict the future values

```
# Forecast
pred = InvBoxCox(forecast(autoarima, h = 20)$mean, 0) # inverse Box Cox
pred
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2008                                                        24.42973 24.57203
## 2009 33.41487 21.36068 21.83909 23.41957 25.23580 23.48795 26.81418 27.42472
## 2010 37.66860 23.35811
##           Sep      Oct      Nov      Dec
## 2008 24.37692 25.91365 27.41698 29.53431
## 2009 27.23939 29.23560 30.71454 33.44293
## 2010
```

```
autoplot(pred)
```

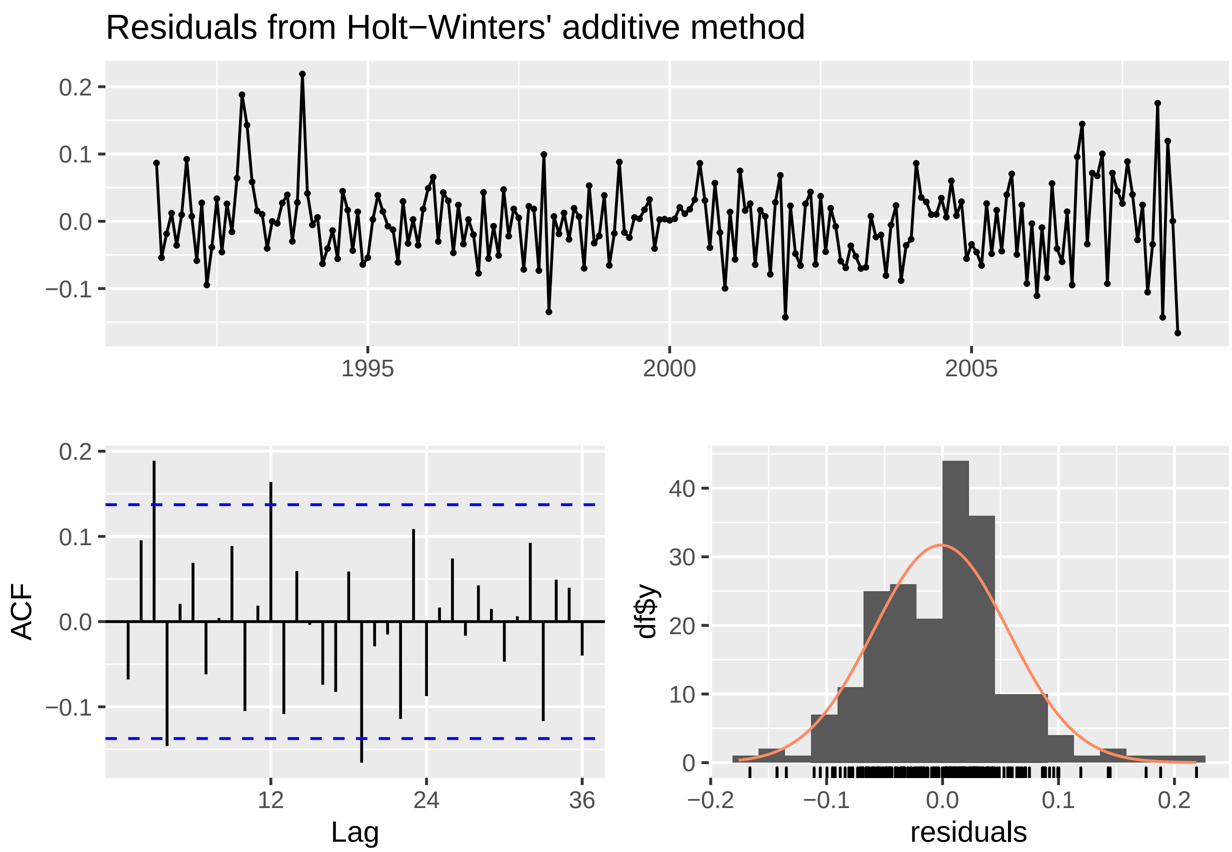


We plot the result of forecast function, and it gives us a similar pattern as before.

Holt Winter Model

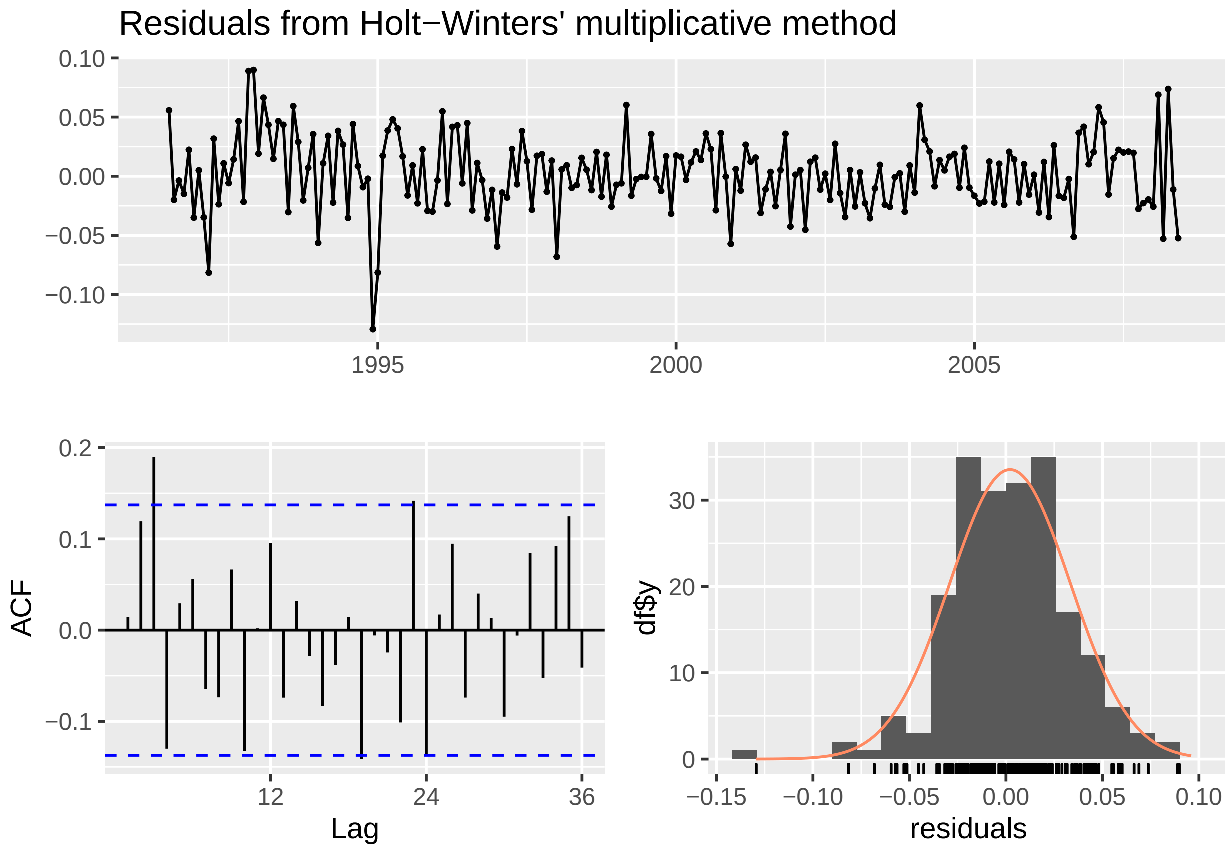
In this section, we are going to experiment with another method, HW Model. We will try out two different methods: Additive and Multiplicative Methods.

```r
# Holt Winter Model
hw_additive = hw(time_series_t, seasonal="additive")
hw_multiplicative = hw(time_series_t, seasonal="multiplicative")
# Residual Checking
checkresiduals(hw_additive)
```



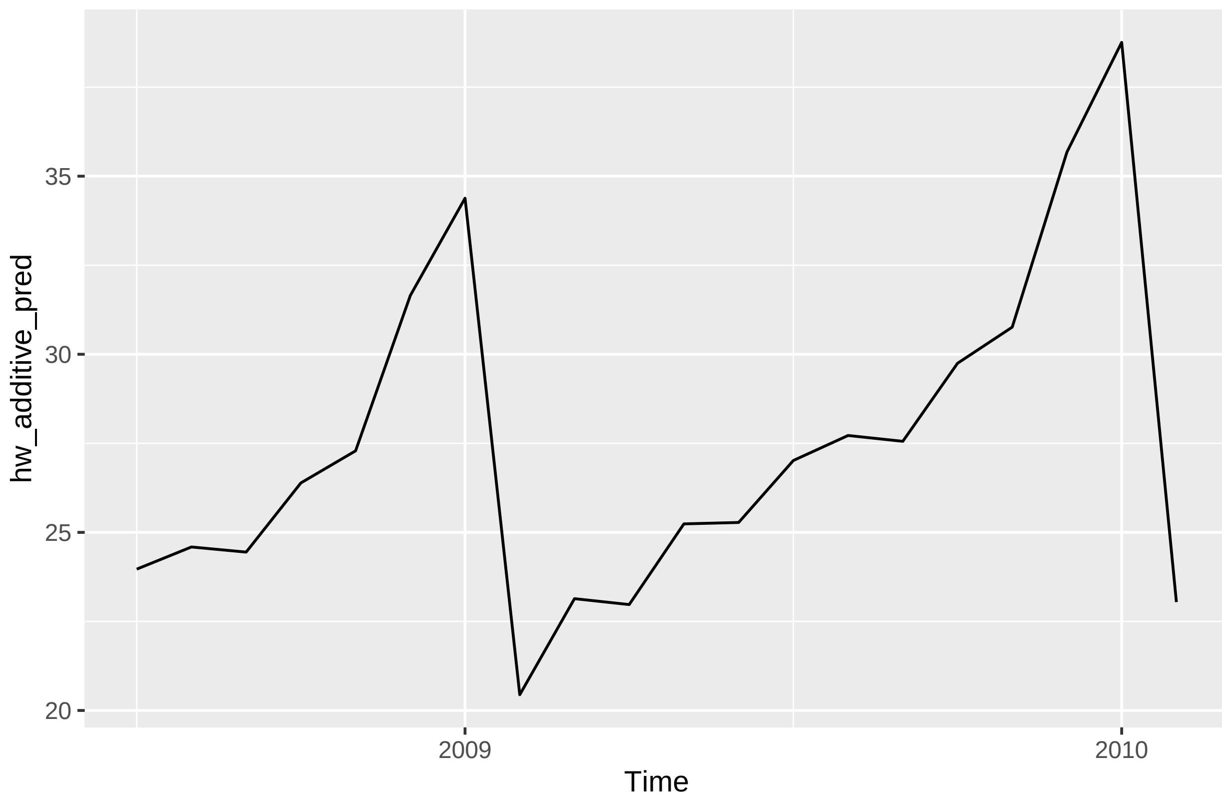

```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters’ additive method
## Q* = 47.643, df = 8, p-value = 1.156e-07
##
## Model df: 16.   Total lags used: 24
```

```r
checkresiduals(hw_multiplicative)
```

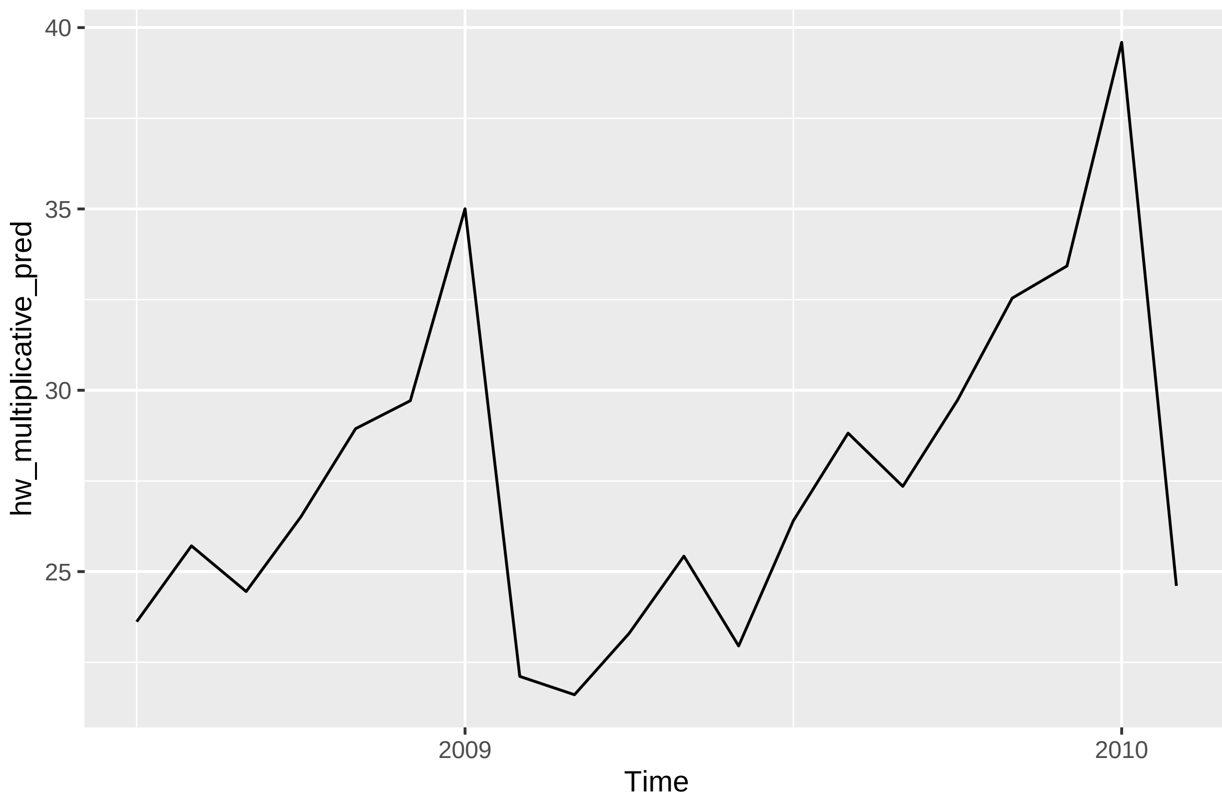



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method
## Q* = 43.464, df = 8, p-value = 7.181e-07
##
## Model df: 16.   Total lags used: 24
```

```
# Forecast
hw_additive_pred <- InvBoxCox(forecast(hw_additive, h=20)$mean, 0)
hw_multiplicative_pred <- InvBoxCox(forecast(hw_multiplicative, h=20)$mean, 0)
autoplot(hw_additive_pred)
```




```
autoplot(hw_multiplicative_pred)
```



We can see from the plots that both models predict very similar pattern as autoarima, the similarity of three models give us confidence that this is actually the future trend. However, the p-values of these two models, are way smaller than that of ARIMA model i.e. 7e-7 vs 1e-7 vs 0.005. Therefore we conclude, the best model of these three is the ARIMA Model.

Conclusion

In this report, we read the data from the text file, we fit it into a time series and plot it out. After plotting out the ACF and PACF, we notice the time series is non-stationary. We run a box-cox transformation with lambda 0 to reduce the variance and turn it into a normal shape. We also notice there is a upward trend, and likely a seasonal component. We used two methods to confirm our hypothesis, the first is use stl:Seasonal Decomposition of Time Series by Loess. We can see clearly from the output that there exist a upward trend, a seasonal fluctuation and a white noise residual. The second mothod is to apply one time differencing to reduce the trend and then apply another differncing with lag 12 to reduce the seasonal component, after doing this, we realize there is only white noise residual left. Then we confirmed our hypothesis that the time series contain three parts. Next we experimented with two models:ARIMA using autoarima and HW Models(additive and multiplicative). We also do forecasting with the three models and all of them give us similar results.

Appendix: Code

```
knitr::opts_chunk$set(echo = TRUE)
library(lubridate)
library(forecast)
```

```r
library(ggplot2)

drug_data = read.delim("./drug.txt",header=TRUE, sep=",")
begin = ymd(as.Date(drug_data$date[1]))
begin
end = ymd(as.Date(drug_data$date[nrow(drug_data)]))
end

time_series = ts(drug_data$value, start=c(year(begin), month(begin)),
                 end=c(year(end) ,month(end)), frequency=12)
plot(time_series,xlab="Year",ylab="Drug Sales")
acf(drug_data$value)
pacf(drug_data$value)
# Seasonal Component
stlresult = stl(time_series, s.window = "periodic")
autoplot(stlresult)
# Data Transformation
time_series_t = BoxCox(time_series, 0)
# Remove trend
time_series_t_1 = diff(time_series_t, differences = 1)
autoplot(time_series_t_1)
# Remove seasonality
time_series_t_1D = diff(time_series_t_1, lag = 12)
autoplot(time_series_t_1D)
# Seasonal ARIMA Model
autoarima = auto.arima(time_series_t)
autoarima
# Diagnostic Checking
checkresiduals(autoarima)
tsdiag(autoarima)
summary(autoarima)
# Forecast
pred = InvBoxCox(forecast(autoarima, h = 20)$mean, 0) # inverse Box Cox
pred
autoplot(pred)
# Holt Winter Model
hw_additive = hw(time_series_t, seasonal="additive")
hw_multiplicative = hw(time_series_t, seasonal="multiplicative")
# Residual Checking
checkresiduals(hw_additive)
checkresiduals(hw_multiplicative)
# Forecast
hw_additive_pred <- InvBoxCox(forecast(hw_additive, h=20)$mean, 0)
hw_multiplicative_pred <- InvBoxCox(forecast(hw_multiplicative, h=20)$mean, 0)
autoplot(hw_additive_pred)
autoplot(hw_multiplicative_pred)
```