Flight Training Simulator

Reading the data

```r
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.0.5
```

```r
Tng_Ctr_Hours <- read_excel("Tng_Ctr_Hour.xlsx")
```

Summary of the data

```r
class(Tng_Ctr_Hours)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```r
summary(Tng_Ctr_Hours)
```

```
##      Year             Quarter             Month             Device_Hrs
##  Length:81          Length:81          Length:81          Min.   : 222.8
##  Class :character   Class :character   Class :character   1st Qu.: 899.0
##  Mode  :character   Mode  :character   Mode  :character   Median :1008.0
##                                                           Mean   : 990.1
##                                                           3rd Qu.:1101.7
##                                                           Max.   :1519.9
##  DH_Prev_Year       DH_YoY_Change      DH_YoY_Ch_Per      Total_Inst_Hrs
##  Length:81          Length:81          Length:81          Min.   : 504.6
##  Class :character   Class :character   Class :character   1st Qu.:1937.3
##  Mode  :character   Mode  :character   Mode  :character   Median :2203.2
##                                                           Mean   :2165.7
##                                                           3rd Qu.:2446.8
##                                                           Max.   :3084.1
##  Total_Inst_Hrs_Prev_Year Inst_Hrs_YoY_Change Total_Inst_Hrs_YoY_Change_Per2
##  Length:81                Length:81           Length:81
##  Class :character         Class :character    Class :character
##  Mode  :character         Mode  :character    Mode  :character
##
##
##
```

Libraries

```r
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.0.5

## -- Attaching packages -------------------------------------------- fpp3 0.4.0 --

## v tibble      3.1.4      v tsibble     1.0.1
## v dplyr       1.0.7      v tsibbledata 0.3.0
## v tidyr       1.1.4      v feasts      0.2.2
## v lubridate   1.7.10     v fable       0.3.1
## v ggplot2     3.3.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'lubridate' was built under R version 4.0.5

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tsibble' was built under R version 4.0.5

## Warning: package 'tsibbledata' was built under R version 4.0.5

## Warning: package 'feasts' was built under R version 4.0.5

## Warning: package 'fabletools' was built under R version 4.0.5

## Warning: package 'fable' was built under R version 4.0.5

## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```r
library(TTR)
```

```
## Warning: package 'TTR' was built under R version 4.0.5
```

```r
library(ggplot2)
library(tsibble)
library(tsibbledata)
library(dplyr)
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(fpp)
```

```
## Warning: package 'fpp' was built under R version 4.0.5

## Loading required package: fma

## Warning: package 'fma' was built under R version 4.0.5

## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 4.0.5

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.0.5

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.0.5

##
## Attaching package: 'zoo'

## The following object is masked from 'package:tsibble':
##
##     index

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: tseries

## Warning: package 'tseries' was built under R version 4.0.5

##
## Attaching package: 'fpp'

## The following object is masked from 'package:fpp3':
##
##     insurance
```

```r
library(fpp2)
```

```
## Warning: package 'fpp2' was built under R version 4.0.5
```

```
##
## Attaching package: 'fpp2'
```

```
## The following objects are masked from 'package:fpp':
##
##     ausair, ausbeer, austa, austourists, debitcards, departures,
##     elecequip, euretail, guinearice, oil, sunspotarea, usmelec
```

```
## The following object is masked from 'package:fpp3':
##
##     insurance
```

```r
library(bsts)
```

```
## Warning: package 'bsts' was built under R version 4.0.5
```

```
## Loading required package: BoomSpikeSlab
```

```
## Warning: package 'BoomSpikeSlab' was built under R version 4.0.5
```

```
## Loading required package: Boom
```

```
## Warning: package 'Boom' was built under R version 4.0.5
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following objects are masked from 'package:fma':
##
##     cement, housing, petrol
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
##
## Attaching package: 'Boom'
```

```
## The following object is masked from 'package:stats':
##
##     rWishart
```

```
##
## Attaching package: 'BoomSpikeSlab'

## The following object is masked from 'package:stats':
##
##     knots

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.0.5

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'bsts'

## The following object is masked from 'package:BoomSpikeSlab':
##
##     SuggestBurn
```

```
library(prophet)
```

```
## Warning: package 'prophet' was built under R version 4.0.5

## Loading required package: Rcpp

## Warning: package 'Rcpp' was built under R version 4.0.5

## Loading required package: rlang

## Warning: package 'rlang' was built under R version 4.0.5
```

```
library(repr)
```

```
## Warning: package 'repr' was built under R version 4.0.5
```

Converting Data Frame to Time Series

```
df_Tng = Tng_Ctr_Hours[,c(4)]
df_Tng
```

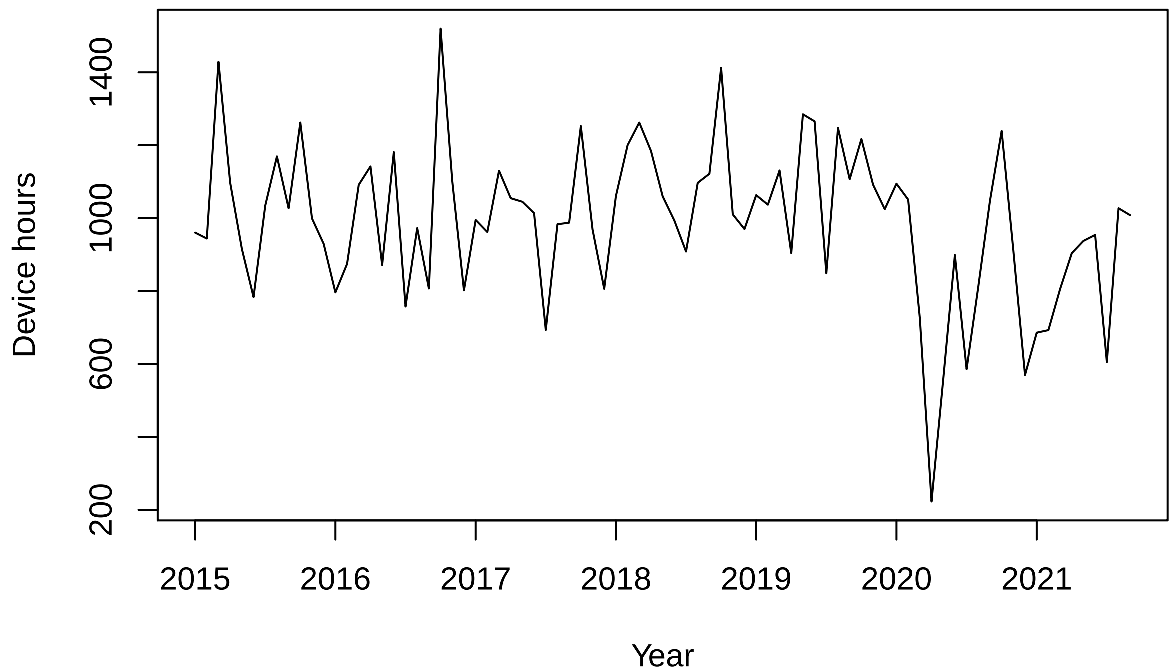
```
## # A tibble: 81 x 1
##    Device_Hrs
##         <dbl>
##  1       960.
##  2       944.
##  3      1429.
##  4      1097
##  5       916.
##  6       783.
##  7      1035.
##  8      1170.
##  9      1027.
## 10      1262.
## # ... with 71 more rows
```

```r
ts_tng = ts(data = df_Tng,frequency = 12,start = c(2015, 1))
ts_tng
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug     Sep
## 2015  960.42  944.08 1429.12 1097.00  915.85  783.45 1034.52 1169.50 1027.08
## 2016  796.42  874.55 1091.55 1141.84  871.36 1181.21  757.59  972.73  807.02
## 2017  995.09  962.00 1130.24 1054.71 1044.95 1013.73  693.33  983.25  987.64
## 2018 1060.57 1200.25 1262.25 1184.45 1059.92  993.55  908.37 1096.93 1121.75
## 2019 1063.13 1036.95 1130.87  903.97 1284.95 1265.56  848.64 1247.40 1106.84
## 2020 1094.62 1050.98  726.19  222.80  556.92  899.00  585.58  811.74 1047.41
## 2021  685.91  692.88  805.42  904.00  937.62  954.00  605.00 1027.23 1008.00
##          Oct     Nov     Dec
## 2015 1262.32  999.25  929.42
## 2016 1519.92 1101.67  801.83
## 2017 1252.69  969.31  806.10
## 2018 1412.47 1010.25  970.12
## 2019 1217.08 1091.84 1024.67
## 2020 1239.26  911.93  569.75
## 2021
```

Plotting the time series

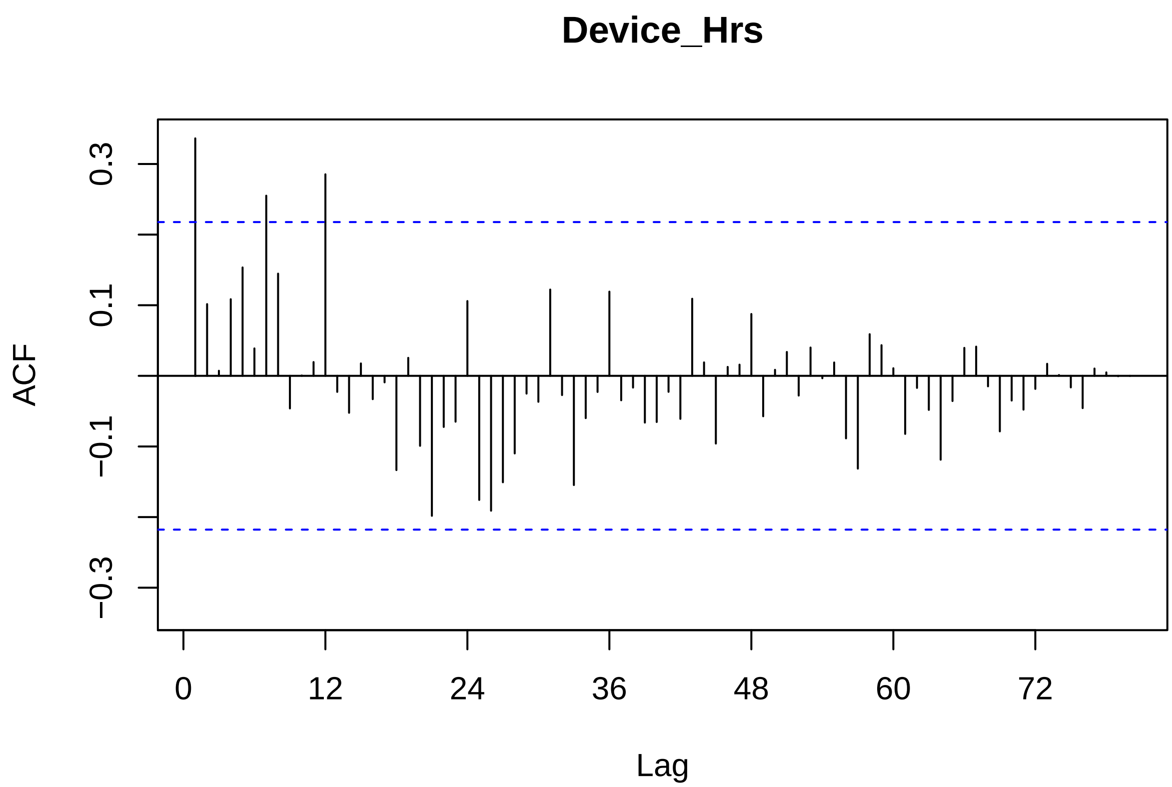
```r
plot(ts_tng,xlab = "Year", ylab = "Device hours")
```



##We can notice in the plot that there is seasonilty and device hours are its peak mostly in the third quarter of every year before 2020.

Acf

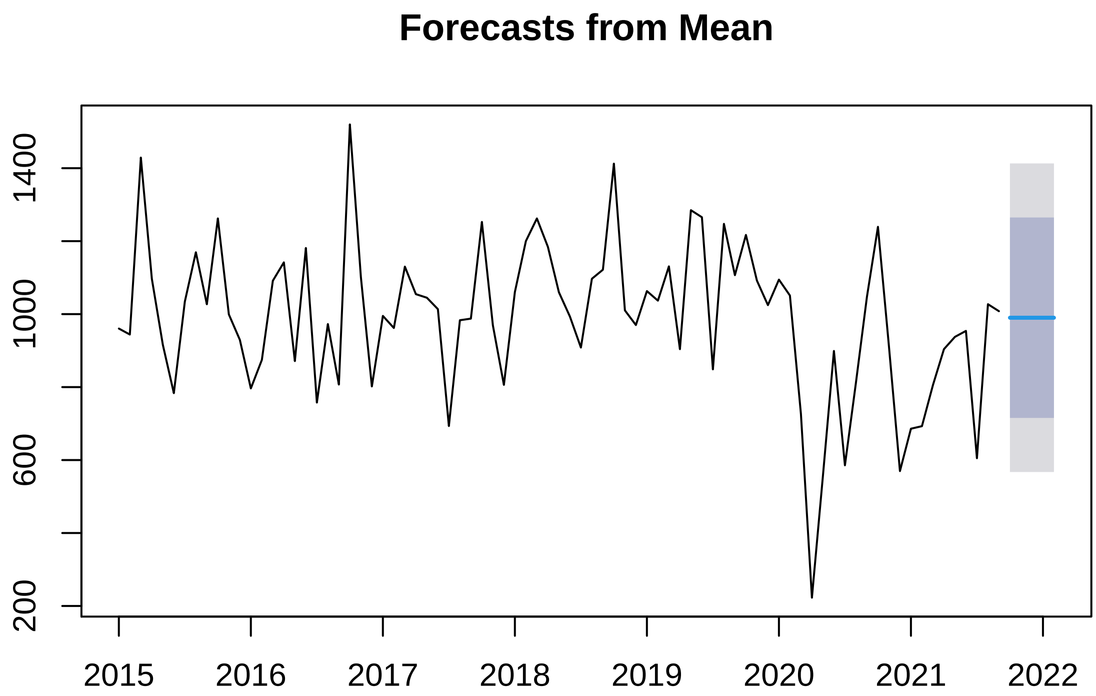
```
Acf(ts_tng,lag = 80)
```



Forecasting Methods

Mean Forecast

```
mean_forecast = meanf(ts_tng, h=5)
plot(mean_forecast)
```



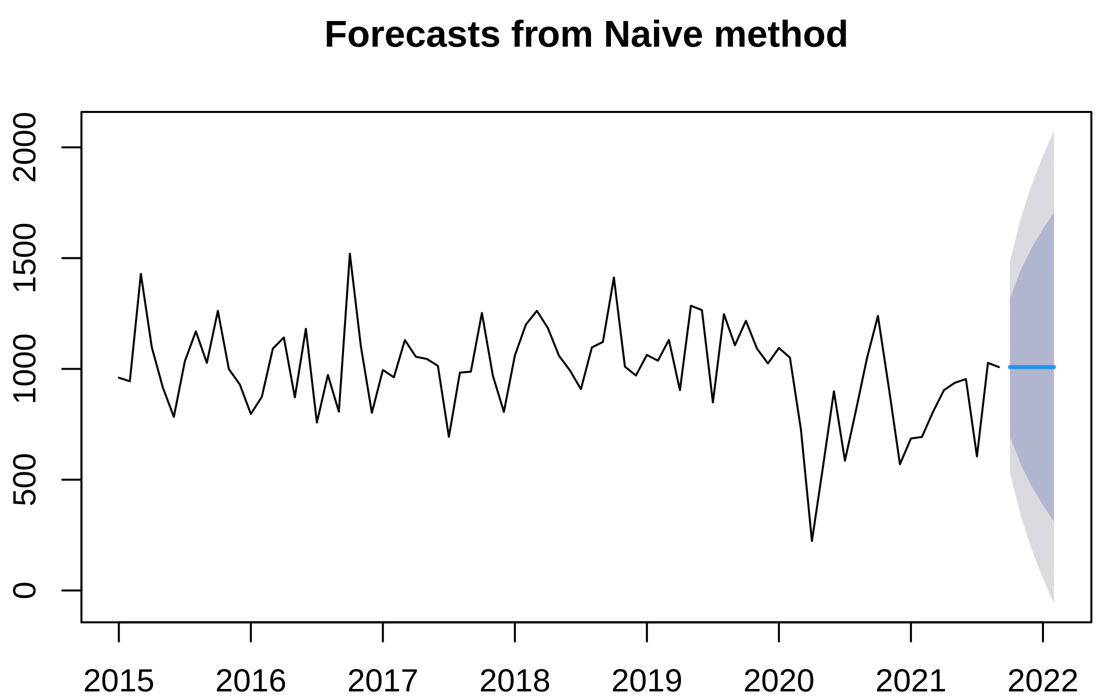

```r
summary(mean_forecast)
```

```
##
## Forecast method: Mean
##
## Model Information:
## $mu
## [1] 990.1452
##
## $mu.se
## [1] 23.4706
##
## $sd
## [1] 211.2354
##
## $bootstrap
## [1] FALSE
##
## $call
## meanf(y = ts_tng, h = 5)
##
## attr(,"class")
## [1] "meanf"
##
## Error measures:
```

```
##                          ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 4.492897e-14 209.9274 156.0292 -7.475987 20.24439 0.8551371
##                   ACF1
## Training set 0.3363161
##
## Forecasts:
##          Point Forecast   Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2021       990.1452 715.502 1264.788 567.1864 1413.104
## Nov 2021       990.1452 715.502 1264.788 567.1864 1413.104
## Dec 2021       990.1452 715.502 1264.788 567.1864 1413.104
## Jan 2022       990.1452 715.502 1264.788 567.1864 1413.104
## Feb 2022       990.1452 715.502 1264.788 567.1864 1413.104
```

Naive Forecast

```r
naive_forecast <- naive(ts_tng,5)
plot(naive_forecast)
```



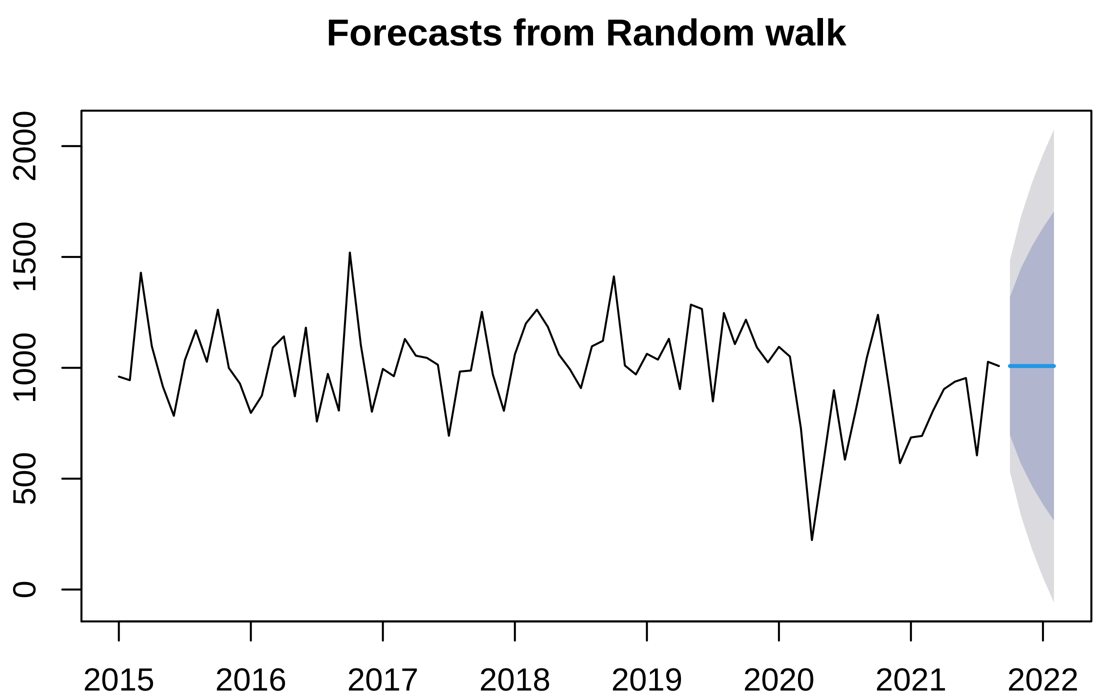

```r
summary(naive_forecast)
```

```
##
## Forecast method: Naive method
##
```

```
## Model Information:
## Call: naive(y = ts_tng, h = 5)
##
## Residual sd: 243.3365
##
## Error measures:
##                   ME     RMSE      MAE       MPE     MAPE     MASE       ACF1
## Training set 0.59475 243.3365 195.7545 -4.593238 22.84353 1.072857 -0.3234359
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Oct 2021           1008 696.1517 1319.848 531.06926 1484.931
## Nov 2021           1008 566.9800 1449.020 333.51808 1682.482
## Dec 2021           1008 467.8630 1548.137 181.93172 1834.068
## Jan 2022           1008 384.3035 1631.697  54.13851 1961.861
## Feb 2022           1008 310.6861 1705.314 -58.44956 2074.450
```

Random Walk Forecast

```
rwf_forecast = rwf(ts_tng,5)
plot(rwf_forecast)
```

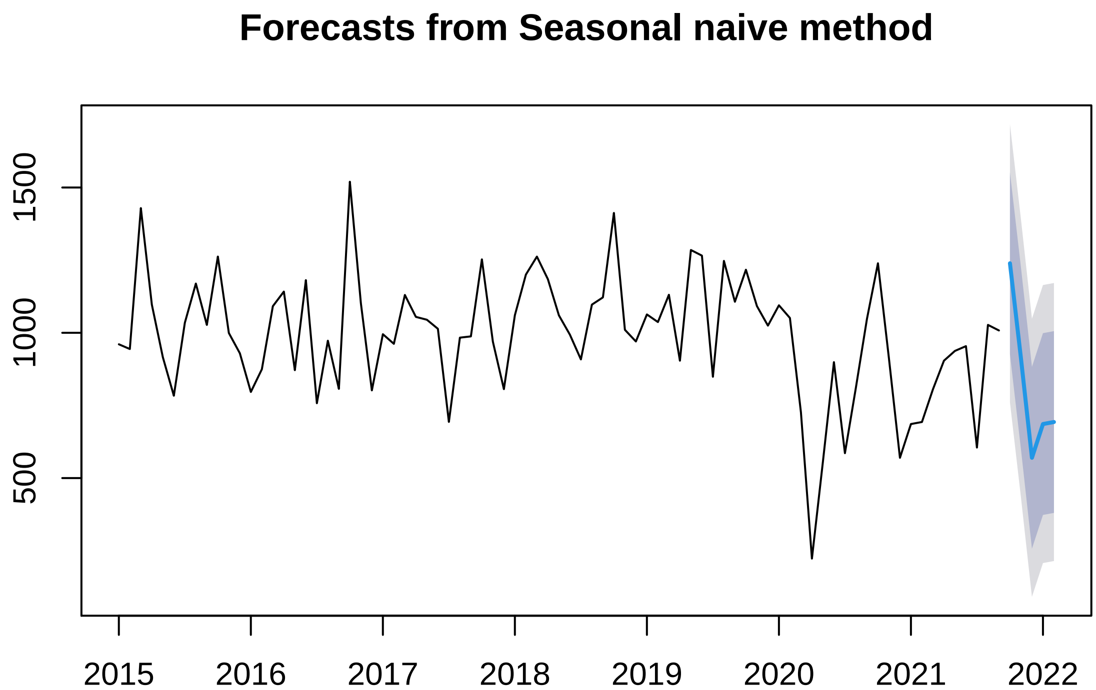


```r
summary(rwf_forecast)
```

```
##
## Forecast method: Random walk
##
## Model Information:
## Call: rwf(y = ts_tng, h = 5)
##
## Residual sd: 243.3365
##
## Error measures:
##                   ME     RMSE      MAE       MPE     MAPE     MASE       ACF1
## Training set 0.59475 243.3365 195.7545 -4.593238 22.84353 1.072857 -0.3234359
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Oct 2021           1008 696.1517 1319.848 531.06926 1484.931
## Nov 2021           1008 566.9800 1449.020 333.51808 1682.482
## Dec 2021           1008 467.8630 1548.137 181.93172 1834.068
## Jan 2022           1008 384.3035 1631.697  54.13851 1961.861
## Feb 2022           1008 310.6861 1705.314 -58.44956 2074.450
```

Seasonal Naive Forecast

```r
snaive_forecast = snaive(ts_tng,5)
plot(snaive_forecast)
```

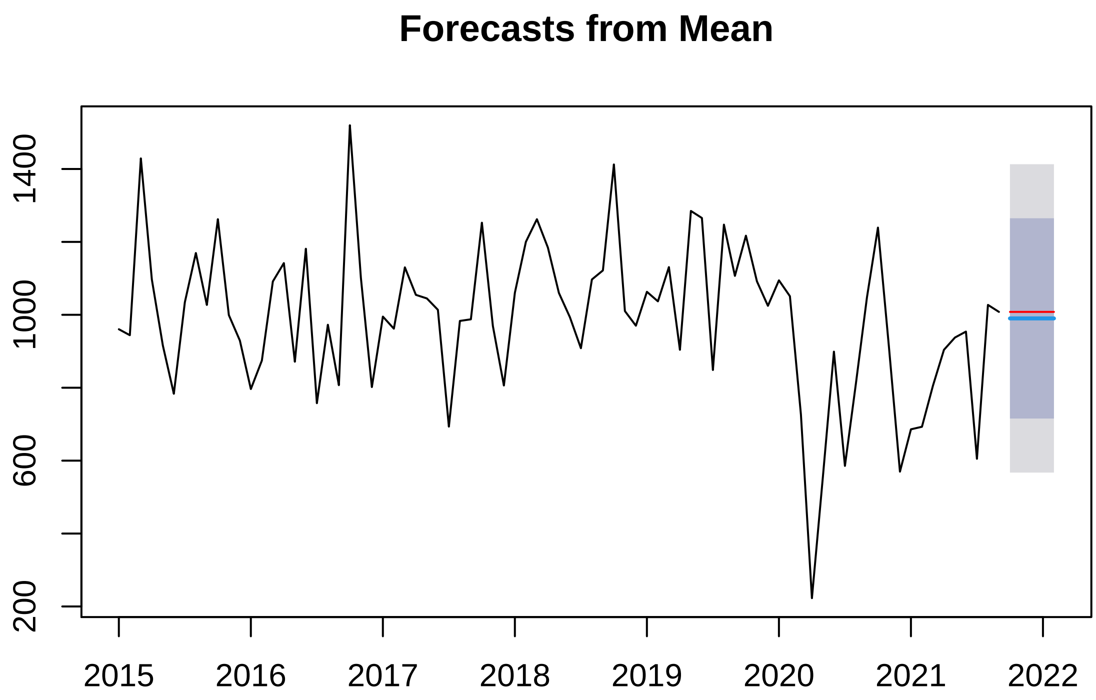



```r
summary(snaive_forecast)
```

```
##
## Forecast method: Seasonal naive method
##
## Model Information:
## Call: snaive(y = ts_tng, h = 5)
##
## Residual sd: 244.0991
##
## Error measures:
##                     ME     RMSE     MAE     MPE   MAPE MASE      ACF1
## Training set -32.04362 244.0991 182.461 -9.9221 24.003    1 0.4863534
##
## Forecasts:
##          Point Forecast    Lo 80     Hi 80     Lo 95    Hi 95
## Oct 2021        1239.26 926.4344 1552.0856 760.83454 1717.685
## Nov 2021         911.93 599.1044 1224.7556 433.50454 1390.355
## Dec 2021         569.75 256.9244  882.5756  91.32454 1048.175
## Jan 2022         685.91 373.0844  998.7356 207.48454 1164.335
## Feb 2022         692.88 380.0544 1005.7056 214.45454 1171.305
```

Plotting mean and naive forecasting together

```r
plot(mean_forecast)
lines(naive_forecast$mean,col="red")
```

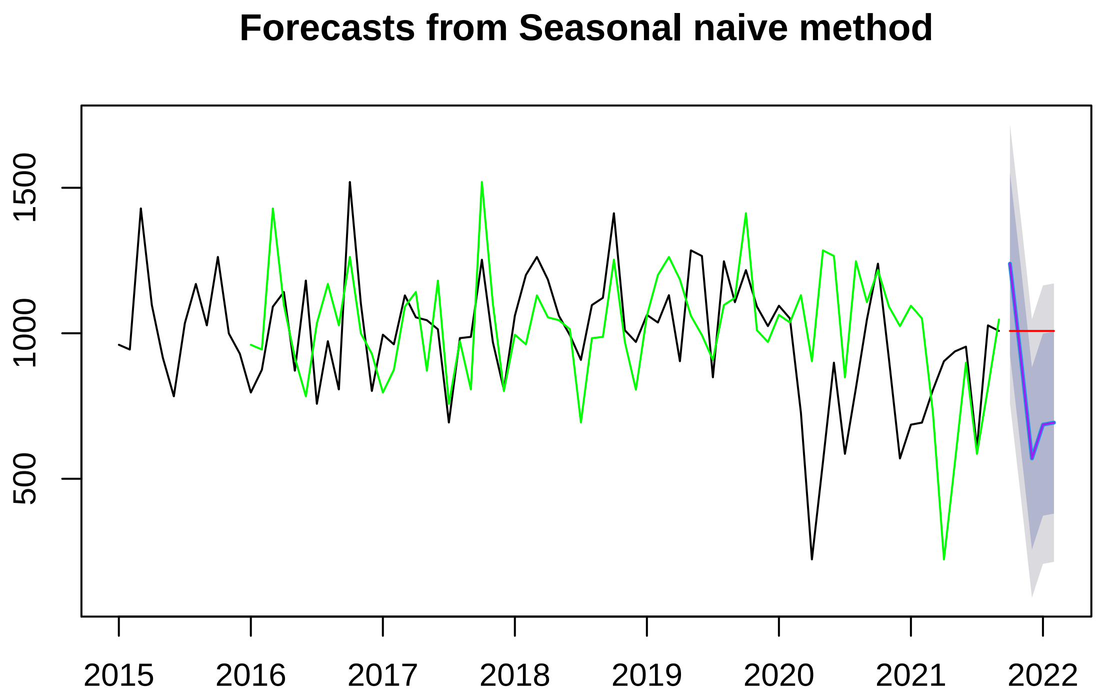



```r
attributes(naive_forecast)
```

```
## $names
##  [1] "method"    "model"     "lambda"    "x"         "fitted"    "residuals"
##  [7] "series"    "mean"      "level"     "lower"     "upper"
##
## $class
## [1] "forecast"
```

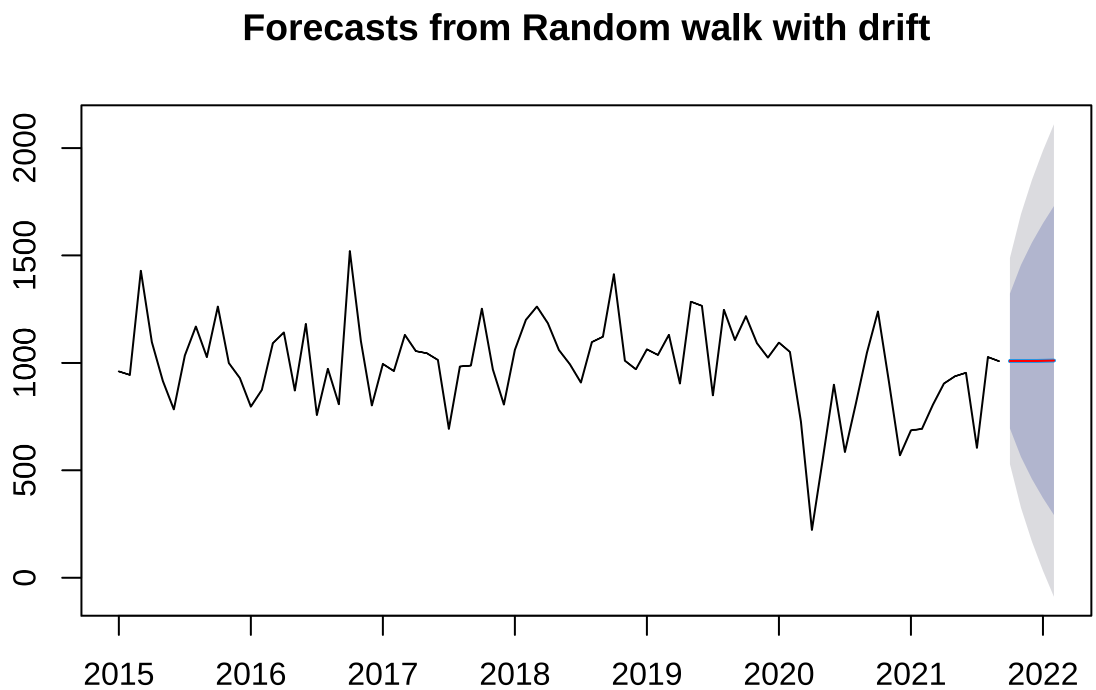
Plotting other attributes

```r
plot(snaive_forecast)
lines(rwf_forecast$mean,col="yellow")
lines(snaive_forecast$mean,col="purple")
lines(snaive_forecast$fitted, col = "green")
lines(naive_forecast$mean,col="red")
```



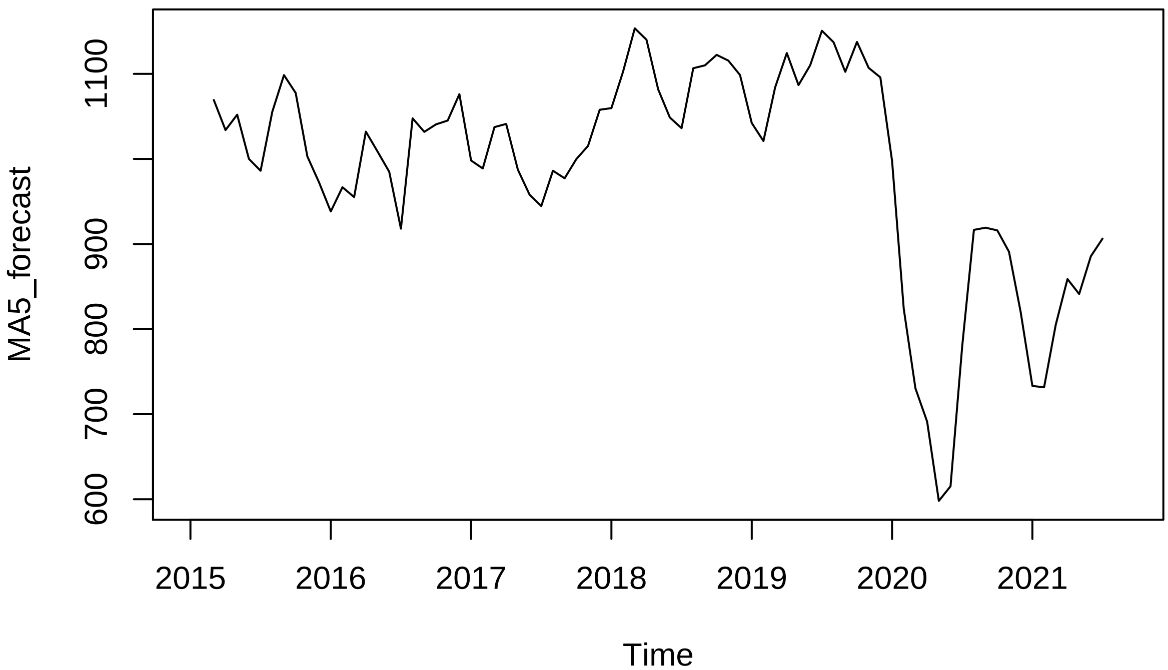
Drift with RWF

```
rwf_drift = rwf(ts_tng,5,drift = TRUE)
plot(rwf_drift)
lines(rwf_drift$mean, col = "red")
```

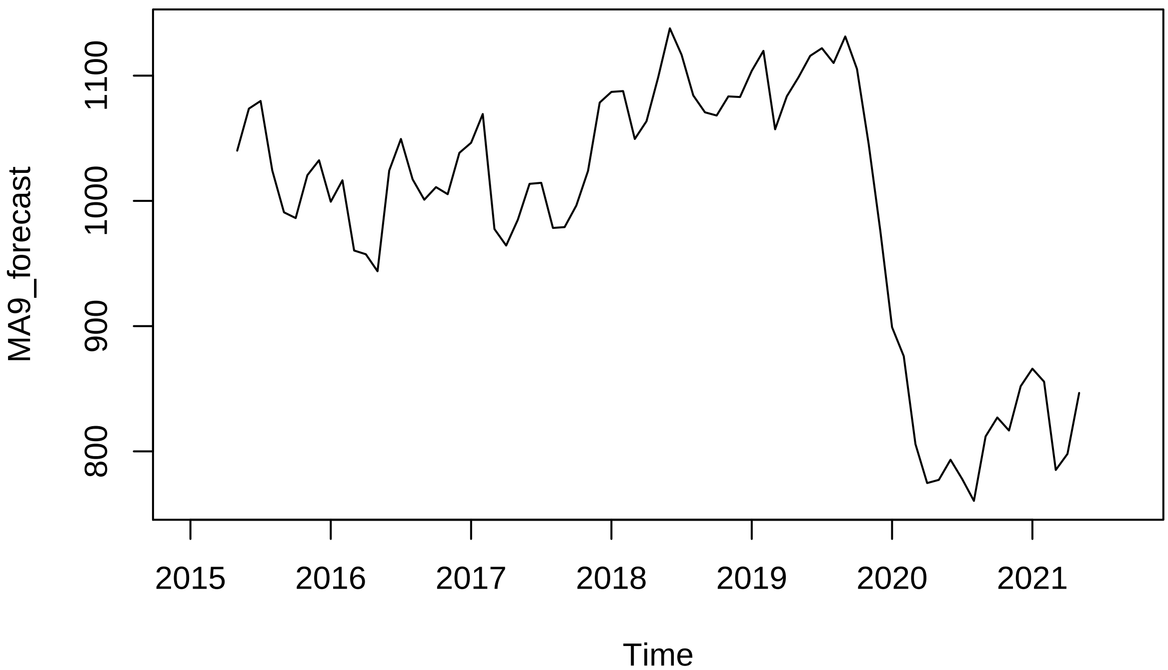


Moving Average Forecast

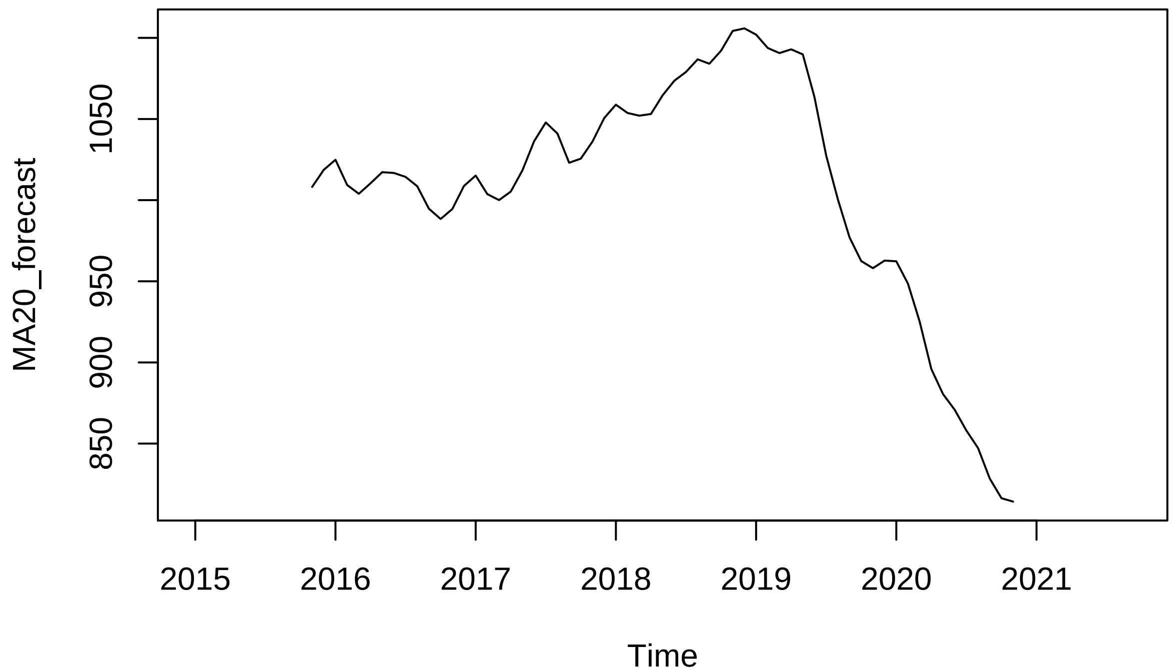
```
MA5_forecast <- ma(ts_tng,order=5)
MA9_forecast <- ma(ts_tng,order=9)
MA20_forecast <- ma(ts_tng,order=20)
plot(MA5_forecast)
```



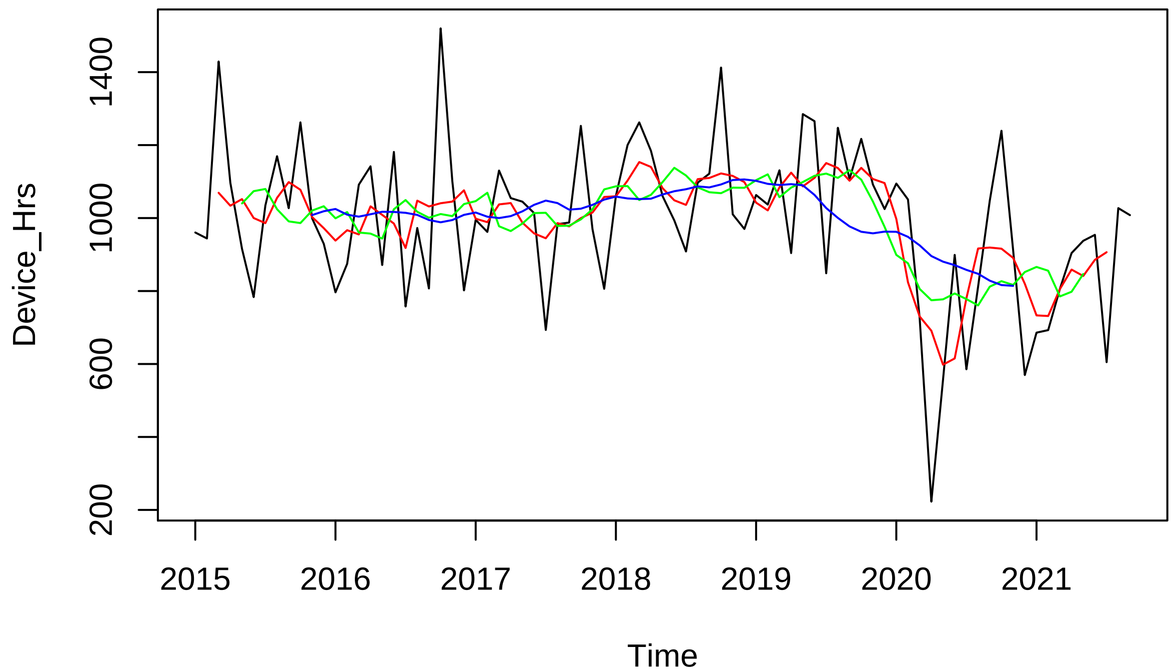

```
plot(MA9_forecast)
```




```r
plot(MA20_forecast)
```




```r
plot(ts_tng)
lines(MA5_forecast, col = "Red")
lines(MA9_forecast, col = "Green")
lines(MA20_forecast, col = "Blue")
```




```
summary(MA5_forecast)
```

```
##        V1
##  Min.   : 598.1
##  1st Qu.: 938.2
##  Median :1021.0
##  Mean   : 989.9
##  3rd Qu.:1081.7
##  Max.   :1153.5
##  NA's   :4
```

As we increase the order, the graph becomes smoother and randomness in the data is decreased.

ETS

```
ets(ts_tng)
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = ts_tng)
```

```
##
##   Smoothing parameters:
##     alpha = 0.3059
##     gamma = 1e-04
##
##   Initial states:
##     l = 1032.5442
##     s = -134.9068 32.2723 332.8318 38.8489 44.9397 -201.0358
##            28.7201 -41.4388 -92.4821 60.5389 -9.4014 -58.8869
##
##   sigma:  171.8918
##
##      AIC     AICc      BIC
## 1204.372 1211.757 1240.289
```

Holt Winters

```
HoltWinters(ts_tng)
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = ts_tng)
##
## Smoothing parameters:
##  alpha: 0.4808712
##  beta : 0
##  gamma: 0.6146842
##
## Coefficients:
##            [,1]
## a    880.345660
## b     -3.477558
## s1   312.455506
## s2    26.881637
## s3  -158.136988
## s4   -49.240618
## s5   -57.936313
## s6   -40.041582
## s7  -127.392228
## s8     6.042409
## s9   101.130866
## s10 -207.296765
## s11  128.799900
## s12  125.004130
```

SSE without trend and without seasonality

```
HoltWinters(ts_tng,beta=FALSE,gamma=FALSE)
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = ts_tng, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.1376349
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##      [,1]
## a 874.898
```

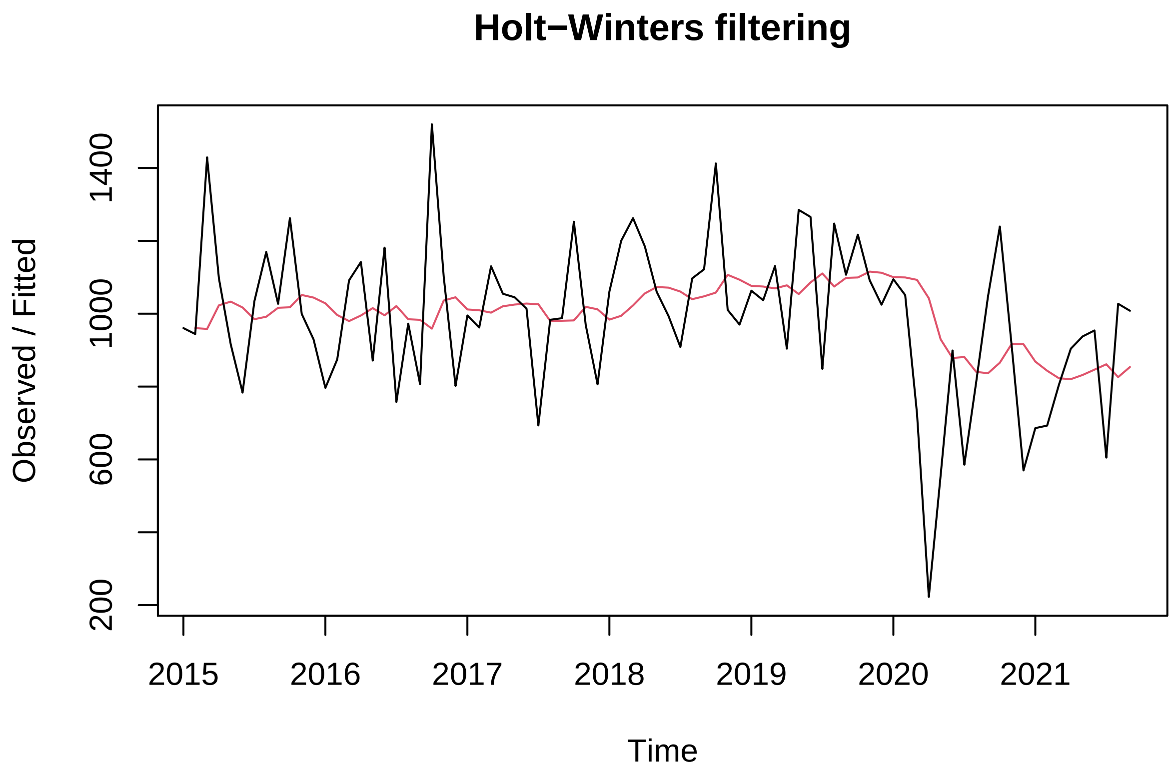
```
hw_forecast_level = HoltWinters(ts_tng,beta=FALSE,gamma=FALSE)
hw_forecast_level
```


```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = ts_tng, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.1376349
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##      [,1]
## a 874.898
```


```
attributes(hw_forecast_level)
```

```
## $names
## [1] "fitted"       "x"            "alpha"        "beta"         "gamma"
## [6] "coefficients" "seasonal"     "SSE"          "call"
##
## $class
## [1] "HoltWinters"
```

```
plot(hw_forecast_level)
```

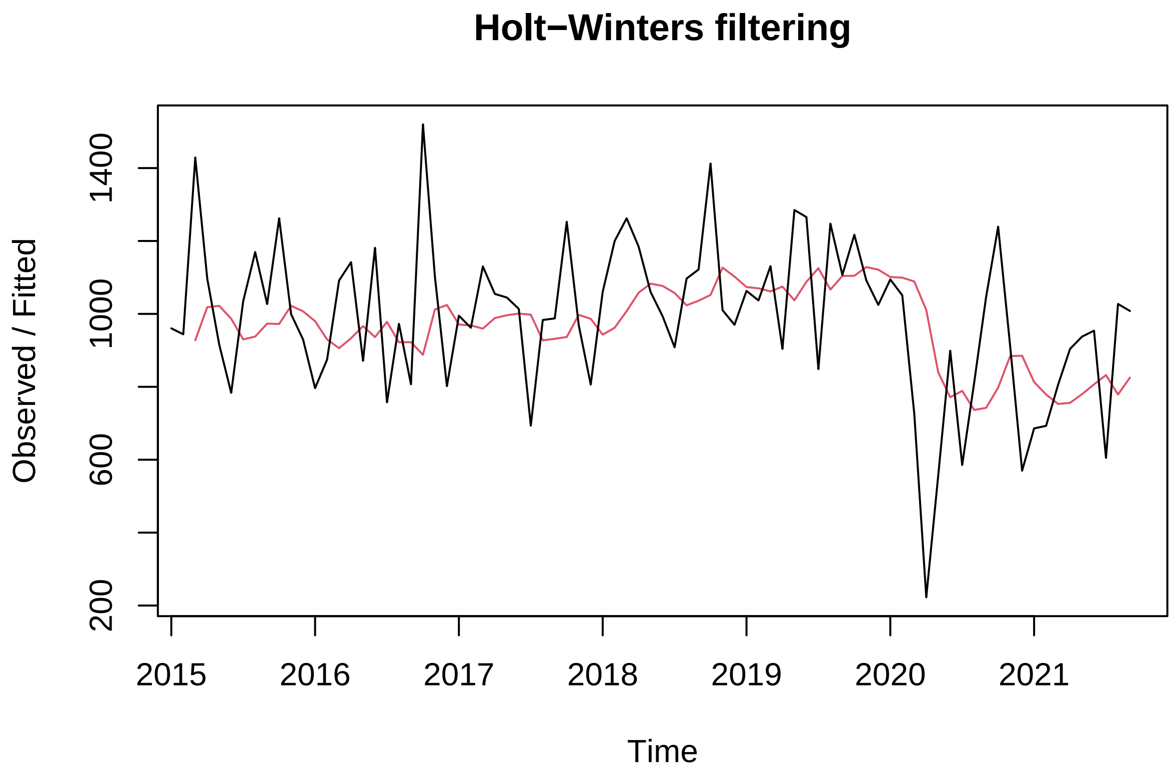



```r
hw_forecast_level$SSE
```

```
## [1] 3401207
```

SSE with Trend but no Seasonlaity

```r
hw_forecast_trend = HoltWinters(ts_tng,gamma=FALSE)
plot(hw_forecast_trend)
```




```r
hw_forecast_trend
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = ts_tng, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.2071599
##  beta : 0.03093563
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 863.238298
## b  -3.632841
```

```r
hw_forecast_trend$SSE #Check the residual error magnitude
```

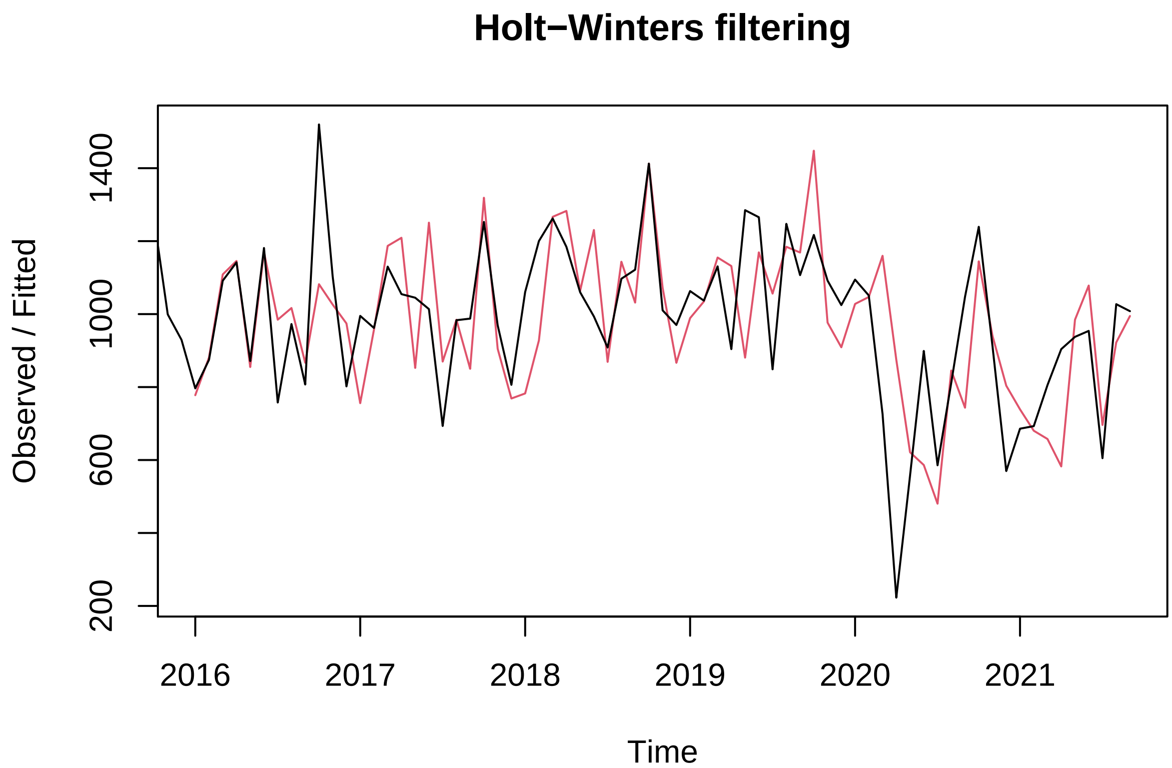
```
## [1] 3586110
```

SSE with trend and seasonality

```r
hw_forecast_season = HoltWinters(ts_tng)
hw_forecast_season
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = ts_tng)
##
## Smoothing parameters:
##  alpha: 0.4808712
##  beta : 0
##  gamma: 0.6146842
##
## Coefficients:
##             [,1]
## a    880.345660
## b     -3.477558
## s1   312.455506
## s2    26.881637
## s3  -158.136988
## s4   -49.240618
## s5   -57.936313
## s6   -40.041582
## s7  -127.392228
## s8     6.042409
## s9   101.130866
## s10 -207.296765
## s11  128.799900
## s12  125.004130
```

```r
plot(hw_forecast_season)
```



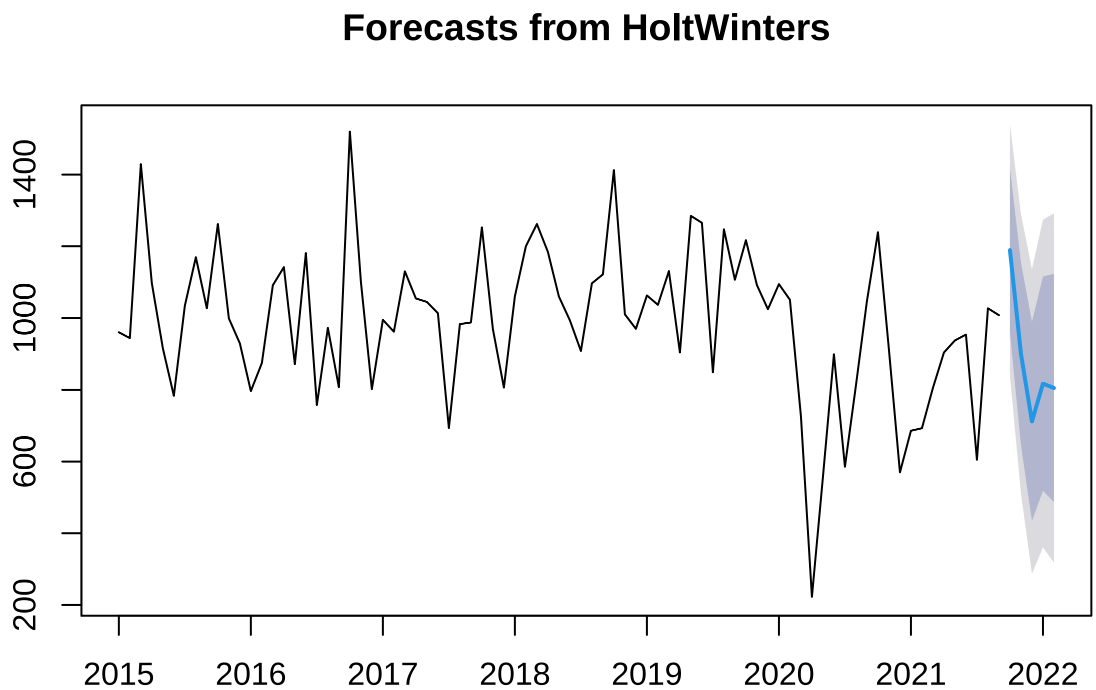

```
hw_forecast_season$SSE
```

```
## [1] 2182216
```

```
hw_forecast_all = forecast(hw_forecast_season,h =5)
hw_forecast_all
```

```
##          Point Forecast    Lo 80     Hi 80    Lo 95    Hi 95
## Oct 2021      1189.3236 959.7727 1418.8745 838.2558 1540.391
## Nov 2021       900.2722 645.5599 1154.9845 510.7234 1289.821
## Dec 2021       711.7760 434.1737  989.3783 287.2198 1136.332
## Jan 2022       817.1948 518.4511 1115.9385 360.3058 1274.084
## Feb 2022       805.0216 486.5369 1123.5063 317.9412 1292.102
```

```
plot(hw_forecast_all)
```




```r
accuracy(hw_forecast_all)
```

```
##                    ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set 2.737314 177.8379 124.1583 -4.279138 16.37587 0.680465 0.1533523
```

SSE of HoltWinters with Trend and Seasonality is smaller than the SSE of Holtwinter without trend, without seasonality and SSE of Holtwinters with Trend and without seasonality.

Ets

```r
ets(ts_tng)
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = ts_tng)
##
##   Smoothing parameters:
##     alpha = 0.3059
##     gamma = 1e-04
##
```

```
##   Initial states:
##     l = 1032.5442
##     s = -134.9068 32.2723 332.8318 38.8489 44.9397 -201.0358
##            28.7201 -41.4388 -92.4821 60.5389 -9.4014 -58.8869
##
##   sigma:  171.8918
##
##      AIC     AICc      BIC
## 1204.372 1211.757 1240.289
```

```
ets_forecast = ets(ts_tng)
attributes(ets)
```

```
## NULL
```

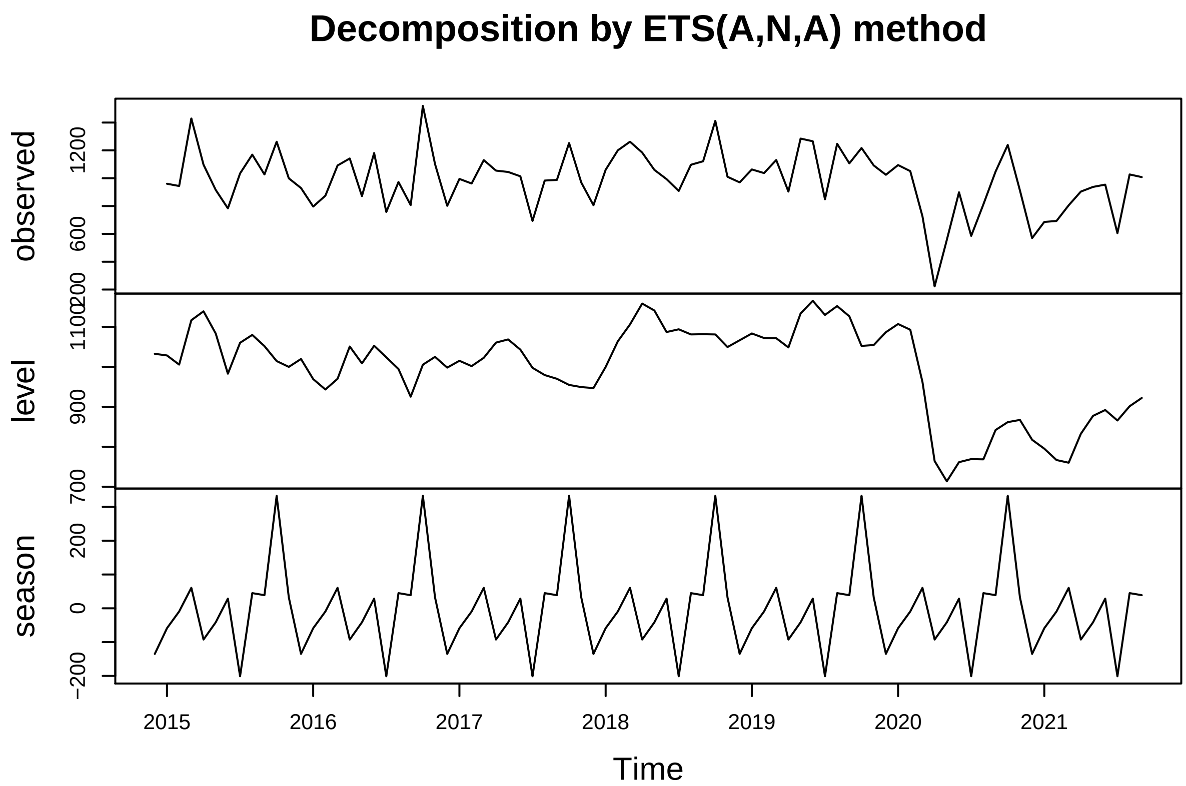
```
attributes(ets_forecast)
```

```
## $names
##  [1] "loglik"     "aic"        "bic"        "aicc"       "mse"
##  [6] "amse"       "fit"        "residuals"  "fitted"     "states"
## [11] "par"        "m"          "method"     "series"     "components"
## [16] "call"       "initstate"  "sigma2"     "x"
##
## $class
## [1] "ets"
```

```
ets_forecast$mse
```

```
## [1] 24439.94
```

```
plot(ets_forecast)
```

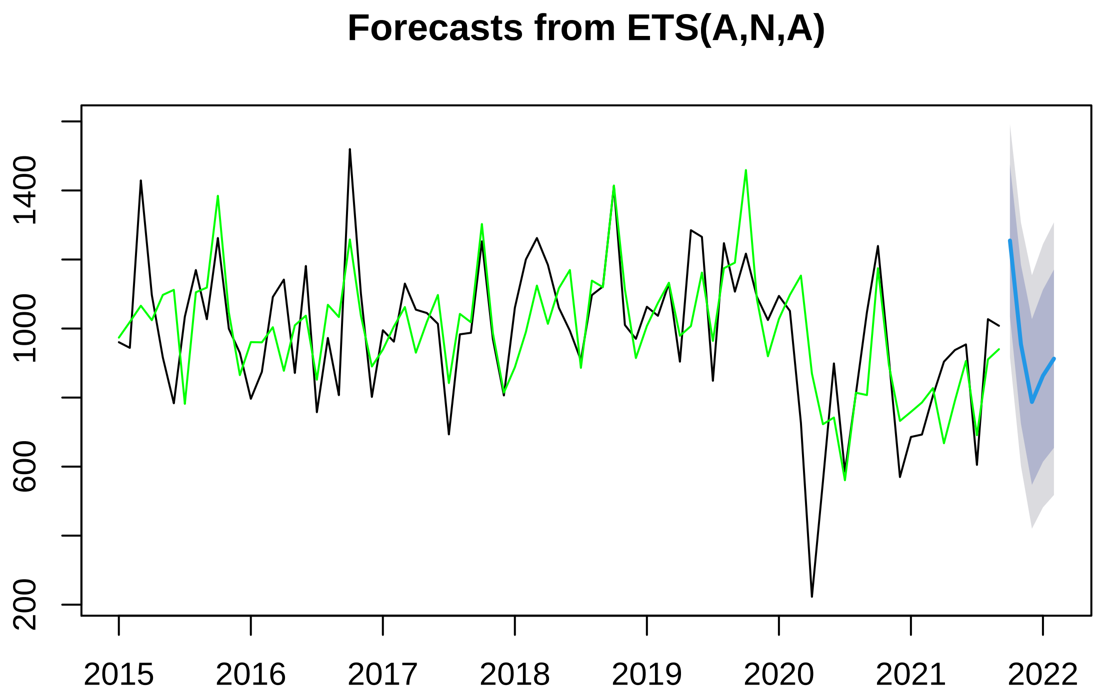


Forecast with Ets

```r
forecast.ets(ets_forecast, h=5)
```

```
##          Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2021      1254.9237 1034.6355 1475.212 918.0219 1591.825
## Nov 2021       954.3654  723.9999 1184.731 602.0517 1306.679
## Dec 2021       787.1911  547.1709 1027.211 420.1118 1154.270
## Jan 2022       863.2255  613.9243 1112.527 481.9522 1244.499
## Feb 2022       912.6818  654.4329 1170.931 517.7242 1307.639
```

```r
forecast_ets = forecast.ets(ets_forecast, h=5)
plot(forecast_ets)
lines(forecast_ets$fitted, col="green")
```




```
accuracy(forecast_ets)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE     ACF1
## Training set -4.457058 156.3328 115.4122 -4.844647 14.80408 0.6325305 0.238284
```

Decomposition

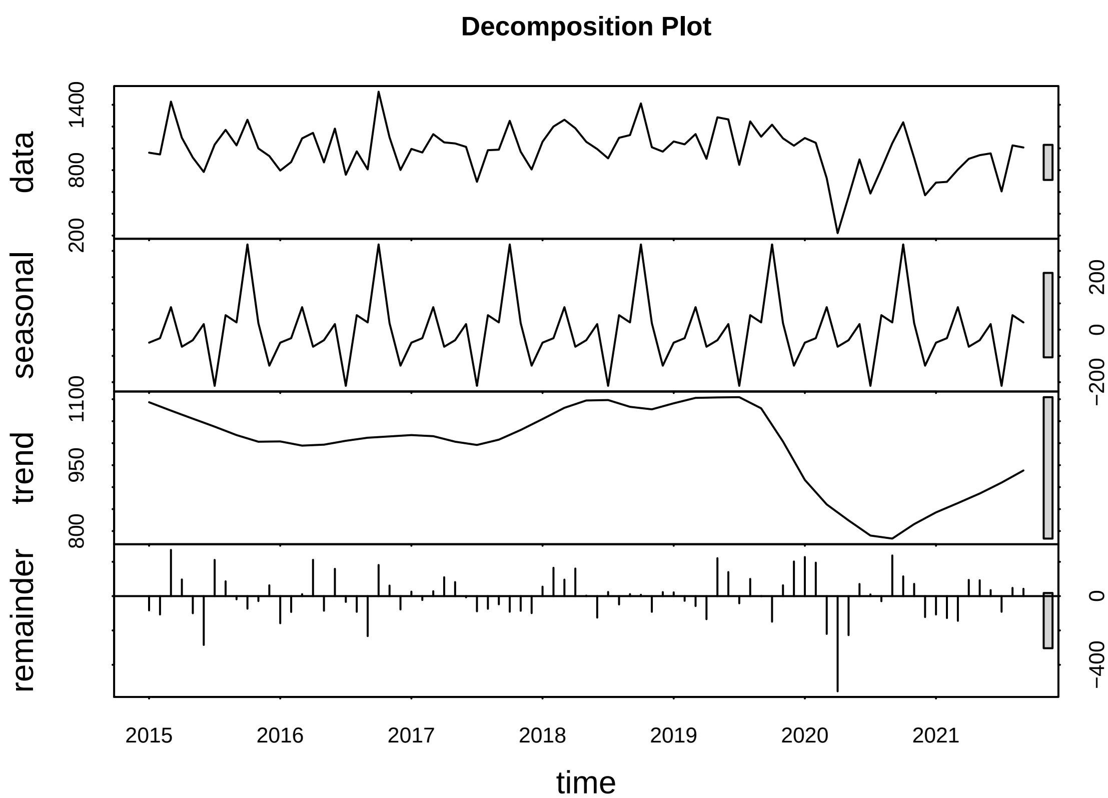
```
stl_decomp = stl(ts_tng[,1], s.window = "periodic")
stl_decomp
```

```
##  Call:
##  stl(x = ts_tng[, 1], s.window = "periodic")
##
## Components
##            seasonal     trend    remainder
## Jan 2015  -49.53300 1093.1208  -83.16779177
## Feb 2015  -32.54166 1083.6081 -106.98644718
## Mar 2015   85.65241 1074.0954  269.37217549
## Apr 2015  -65.11336 1064.9135   97.19989204
## May 2015  -40.21191 1055.7315  -99.66961400
## Jun 2015   21.15910 1046.6548 -284.36385343
## Jul 2015 -214.09860 1037.5780  211.04060499
## Aug 2015   55.16364 1027.9696   86.36679892
```

```
## Sep 2015   27.45590 1018.3611  -18.73703353
## Oct 2015  324.64606 1010.8301  -73.15614722
## Nov 2015   24.72279 1003.2990  -28.77182093
## Dec 2015 -137.30135 1003.6182   63.10318714
## Jan 2016  -49.53300 1003.9373 -157.98428562
## Feb 2016  -32.54166  999.1761  -92.08449074
## Mar 2016   85.65241  994.4150   11.48258223
## Apr 2016  -65.11336  995.4975  211.45586863
## May 2016  -40.21191  996.5800  -85.00806757
## Jun 2016   21.15910 1000.9900  159.06088988
## Jul 2016 -214.09860 1005.4001  -33.71145482
## Aug 2016   55.16364 1008.8778  -91.31141649
## Sep 2016   27.45590 1012.3555 -232.79140456
## Oct 2016  324.64606 1013.8572  181.41676327
## Nov 2016   24.72279 1015.3588   61.58837107
## Dec 2016 -137.30135 1016.8633  -77.73198136
## Jan 2017  -49.53300 1018.3678   26.25518540
## Feb 2017  -32.54166 1017.1312  -22.58953406
## Mar 2017   85.65241 1015.8946   28.69302456
## Apr 2017  -65.11336 1009.6212  110.20214418
## May 2017  -40.21191 1003.3479   81.81404121
## Jun 2017   21.15910  999.6403   -7.06934827
## Jul 2017 -214.09860  995.9326  -88.50403991
## Aug 2017   55.16364 1001.9257  -73.83929623
## Sep 2017   27.45590 1007.9187  -47.73457895
## Oct 2017  324.64606 1018.9562  -90.91225336
## Nov 2017   24.72279 1029.9937  -85.40648779
## Dec 2017 -137.30135 1042.4018  -99.00041465
## Jan 2018  -49.53300 1054.8098   55.29317767
## Feb 2018  -32.54166 1067.6081  165.18354097
## Mar 2018   85.65241 1080.4064   96.19118236
## Apr 2018  -65.11336 1088.7597  160.80367390
## May 2018  -40.21191 1097.1130    3.01894285
## Jun 2018   21.15910 1097.6102 -125.21925187
## Jul 2018 -214.09860 1098.1073   24.36125125
## Aug 2018   55.16364 1090.3812  -48.61480126
## Sep 2018   27.45590 1082.6550   11.63911983
## Oct 2018  324.64606 1079.8006    8.02337706
## Nov 2018   24.72279 1076.9461  -91.41892573
## Dec 2018 -137.30135 1083.8065   23.61486013
## Jan 2019  -49.53300 1090.6668   21.99616517
## Feb 2019  -32.54166 1096.8486  -27.35691609
## Mar 2019   85.65241 1103.0303  -57.81271927
## Apr 2019  -65.11336 1103.5243 -134.44095551
## May 2019  -40.21191 1104.0183  221.14358566
## Jun 2019   21.15910 1104.3475  140.05338656
## Jul 2019 -214.09860 1104.6767  -41.93811471
## Aug 2019   55.16364 1091.9845  100.25185655
## Sep 2019   27.45590 1079.2923    0.09180141
## Oct 2019  324.64606 1041.5025 -149.06860289
## Nov 2019   24.72279 1003.7128   63.40443278
## Dec 2019 -137.30135  960.0656  201.90573284
## Jan 2020  -49.53300  916.4184  227.73455208
## Feb 2020  -32.54166  888.5121  195.00955927
```

```
## Mar 2020   85.65241  860.6057 -220.06815545
## Apr 2020  -65.11336  842.4398 -554.52644481
## May 2020  -40.21191  824.2739 -227.14195677
## Jun 2020   21.15910  807.0541   70.78677612
## Jul 2020 -214.09860  789.8344    9.84420684
## Aug 2020   55.16364  786.3806  -29.80425440
## Sep 2020   27.45590  782.9268  237.02725796
## Oct 2020  324.64606  799.3530  115.26098122
## Nov 2020   24.72279  815.7791   71.42814445
## Dec 2020 -137.30135  829.1369 -122.08559902
## Jan 2021  -49.53300  842.4948 -107.05182331
## Feb 2021  -32.54166  853.0745 -127.65285384
## Mar 2021   85.65241  863.6542 -143.88660628
## Apr 2021  -65.11336  874.5410   94.57237327
## May 2021  -40.21191  885.4278   92.40413024
## Jun 2021   21.15910  897.8348   35.00608204
## Jul 2021 -214.09860  910.2419  -91.14326831
## Aug 2021   55.16364  924.0720   47.99440661
## Sep 2021   27.45590  937.9020   42.64205513
```

```
plot(stl_decomp, main="Decomposition Plot")
```




```
attributes(stl_decomp)
```

```
## $names
```

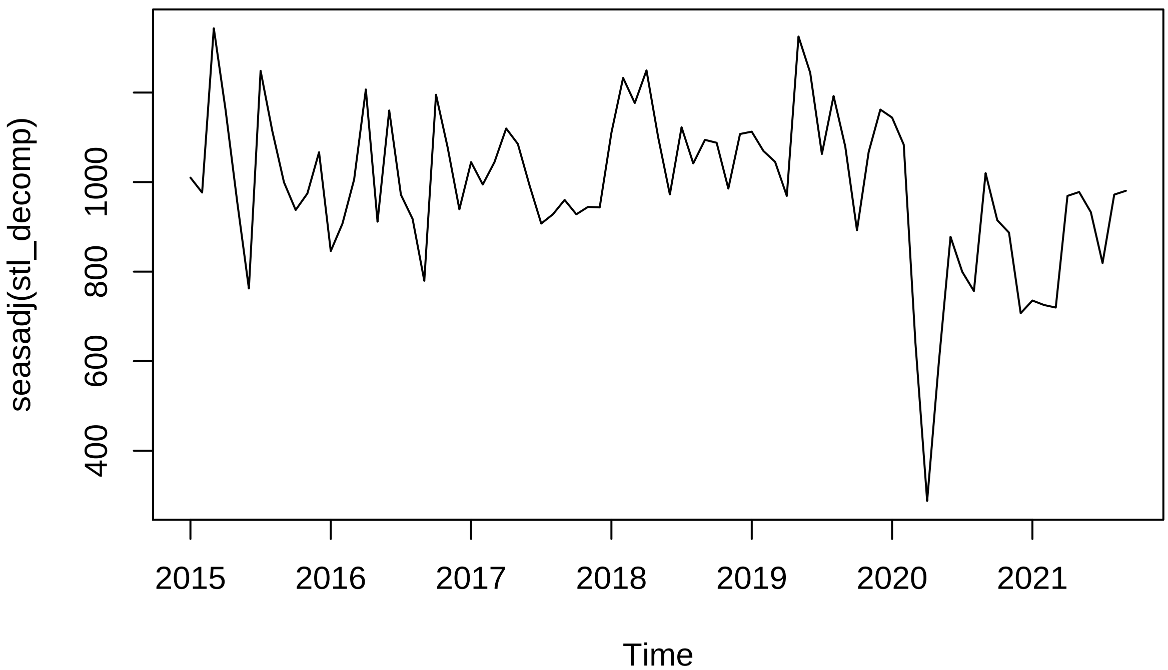
```
## [1] "time.series" "weights"     "call"        "win"         "deg"
## [6] "jump"        "inner"       "outer"
##
## $class
## [1] "stl"
```

Seasonal Adjustment

```r
seasadj(stl_decomp)
```

```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 2015 1009.9530  976.6217 1343.4676 1162.1134  956.0619  762.2909 1248.6186
## 2016  845.9530  907.0917 1005.8976 1206.9534  911.5719 1160.0509  971.6886
## 2017 1044.6230  994.5417 1044.5876 1119.8234 1085.1619  992.5709  907.4286
## 2018 1110.1030 1232.7917 1176.5976 1249.5634 1100.1319  972.3909 1122.4686
## 2019 1112.6630 1069.4917 1045.2176  969.0834 1325.1619 1244.4009 1062.7386
## 2020 1144.1530 1083.5217  640.5376  287.9134  597.1319  877.8409  799.6786
## 2021  735.4430  725.4217  719.7676  969.1134  977.8319  932.8409  819.0986
##            Aug       Sep       Oct       Nov       Dec
## 2015 1114.3364  999.6241  937.6739  974.5272 1066.7213
## 2016  917.5664  779.5641 1195.2739 1076.9472  939.1313
## 2017  928.0864  960.1841  928.0439  944.5872  943.4013
## 2018 1041.7664 1094.2941 1087.8239  985.5272 1107.4213
## 2019 1192.2364 1079.3841  892.4339 1067.1172 1161.9713
## 2020  756.5764 1019.9541  914.6139  887.2072  707.0513
## 2021  972.0664  980.5441
```

```r
plot(seasadj(stl_decomp))
```

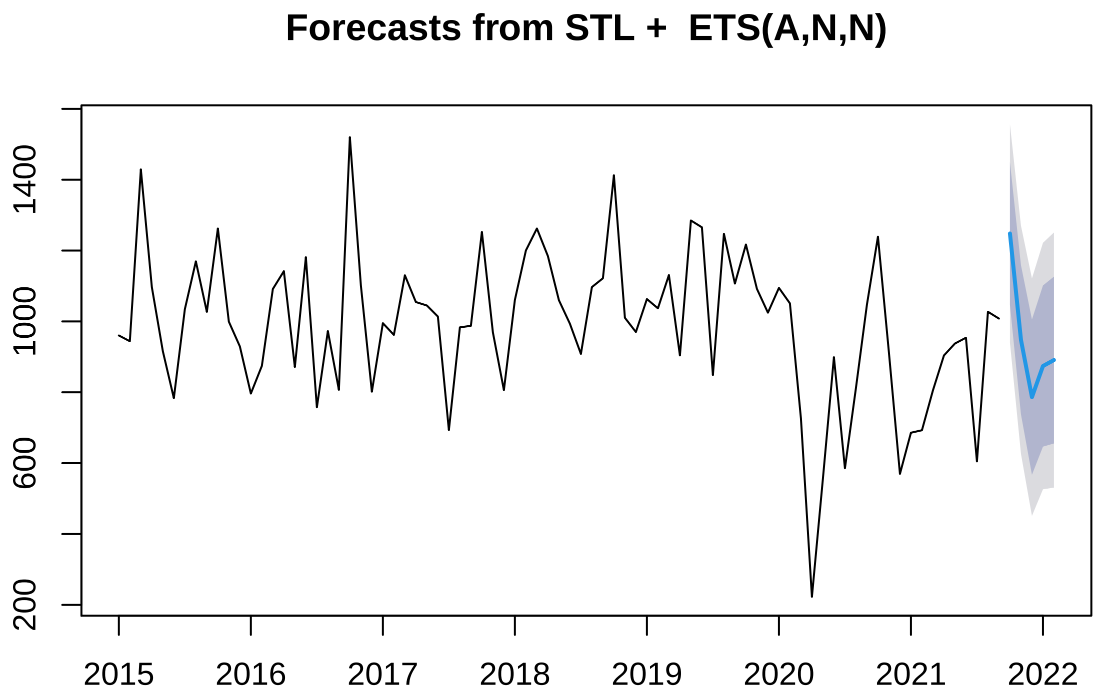


Default Period Forecast

```r
f_stl = forecast(stl_decomp,h = 5)
f_stl
```

```
##          Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2021      1248.2305 1046.6833 1449.778 939.9906 1556.470
## Nov 2021       948.3072  737.7759 1158.839 626.3274 1270.287
## Dec 2021       786.2831  567.1357 1005.430 451.1261 1121.440
## Jan 2022       874.0514  646.6142 1101.489 526.2161 1221.887
## Feb 2022       891.0428  655.6073 1126.478 530.9753 1251.110
```

```r
plot(f_stl)
```




```r
accuracy(f_stl)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -5.584029 155.3144 112.6539 -5.014121 14.67709 0.6174134 0.2430407
```

Accuracy is improved for stl decomp as MAPE is slightly lower compared to other forecasts.