# Post Covid Forecast

**In this dataset, to consider covid as a new normal, we have taken forecasted values of 2020 to smoothen out the outliers due to covid, and have taken the same values of 2021 to consider covid as a new reality.**

### Reading the data

```
library(readxl)
```

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

```
Tng_Ctr_Hours <- read_excel("C:/Users/prach/Desktop/Rutgers/BF/Project/Tng_Hr.xlsx")
```

### Summary of the data

```
class(Tng_Ctr_Hours)
```

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

```
summary(Tng_Ctr_Hours)
```

```
##      Year             Quarter             Month             Device_Hrs
##  Length:81          Length:81          Length:81          Min.   : 605.0
##  Class :character   Class :character   Class :character   1st Qu.: 937.6
##  Mode  :character   Mode  :character   Mode  :character   Median :1027.2
##                                                           Mean   :1031.0
##                                                           3rd Qu.:1121.8
##                                                           Max.   :1519.9
```

### Libraries

```
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
```

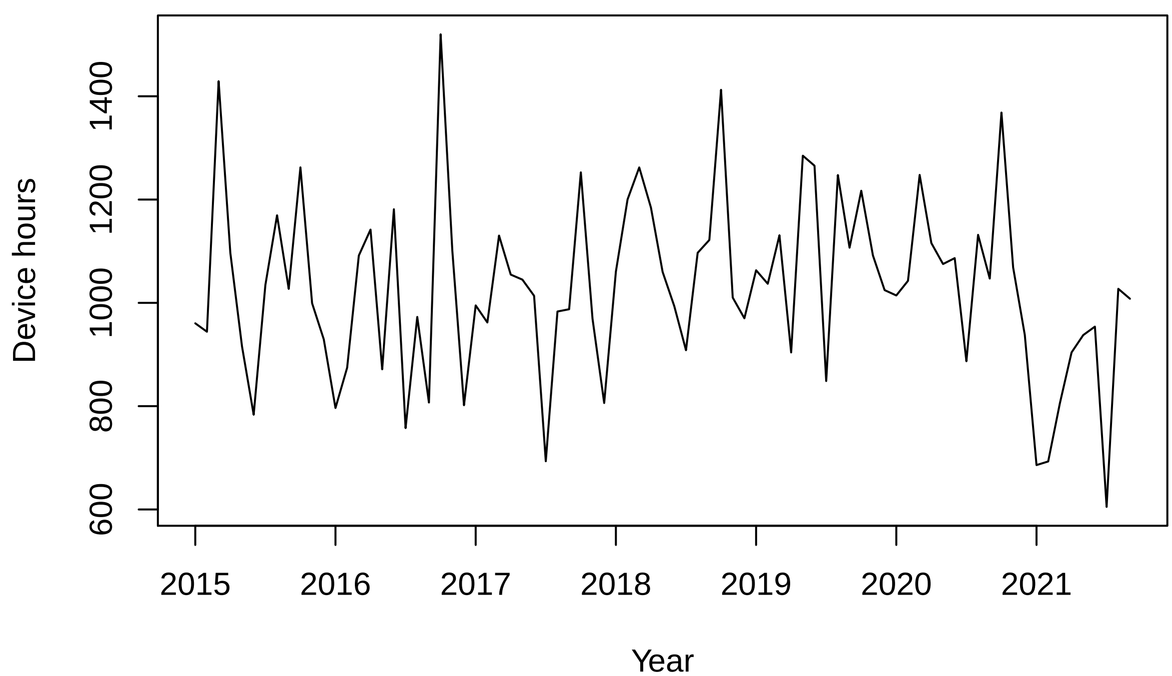
```
## # A tibble: 81 x 1
##    Device_Hrs
##         <dbl>
##  1       960.
##  2       944.
##  3      1429.
##  4      1097
##  5       916.
##  6       783.
##  7      1035.
##  8      1170.
##  9      1027.
## 10      1262.
## # ... with 71 more rows
```

```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 1014.32 1042.63 1247.73 1115.74 1075.18 1086.63  886.98 1131.69 1047.04
## 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 1368.48 1068.67  937.76
## 2021
```

### Plotting the time series

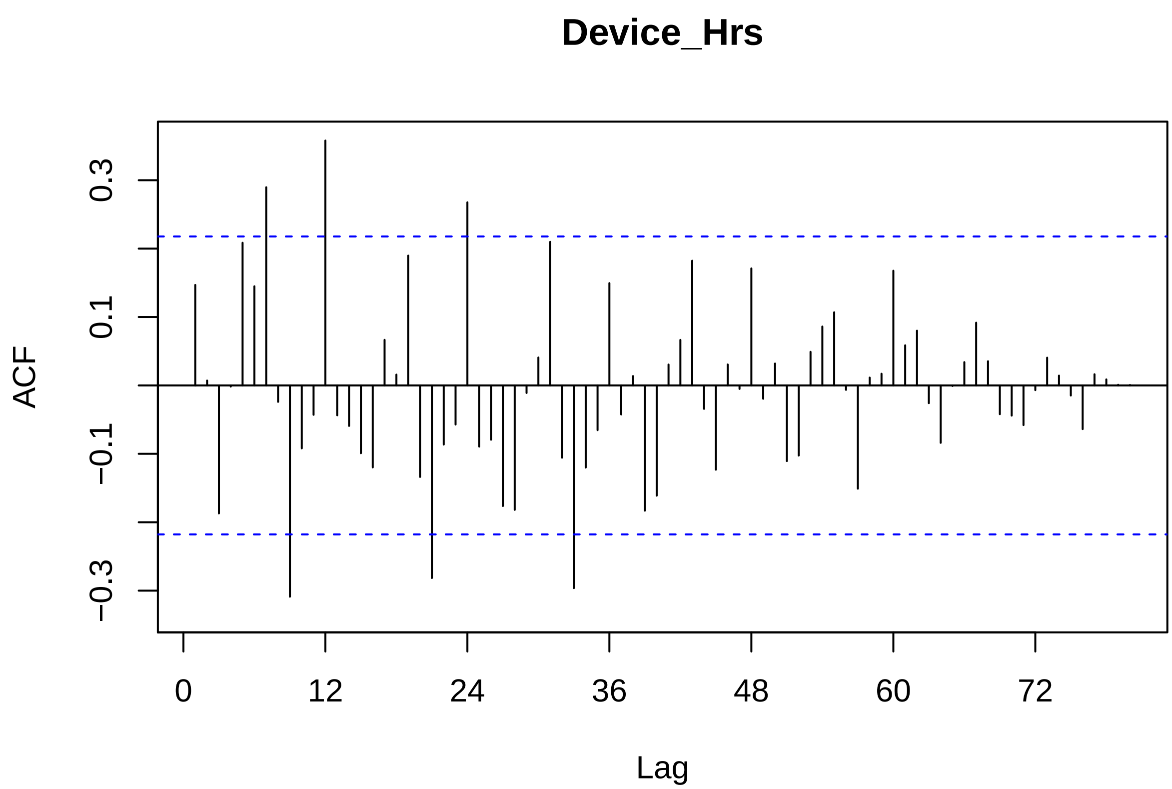
```r
plot(ts_tng,xlab = "Year", ylab = "Device hours")
```



##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)
```

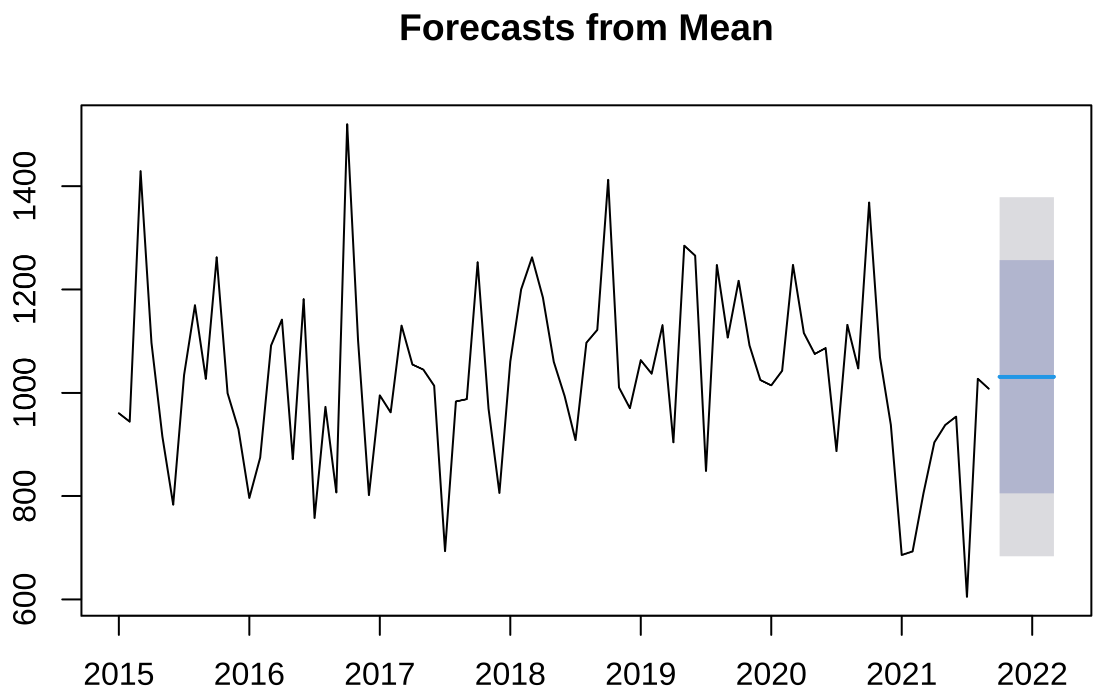



## Forecasting Methods

### Mean Forecast

### Forecasting is for the following 6 period month.

```
mean_forecast = meanf(ts_tng, h=6)
plot(mean_forecast)
```




```r
summary(mean_forecast)
```

```
##
## Forecast method: Mean
##
## Model Information:
## $mu
## [1] 1030.968
##
## $mu.se
## [1] 19.28435
##
## $sd
## [1] 173.5591
##
## $bootstrap
## [1] FALSE
##
## $call
## meanf(y = ts_tng, h = 6)
##
## attr(,"class")
## [1] "meanf"
##
## Error measures:
```

```
##                          ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -3.088305e-14 172.4845 130.8521 -2.998382 13.43832 0.9041035
##                   ACF1
## Training set 0.1469126
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2021       1030.968 805.3109 1256.626 683.4491 1378.487
## Nov 2021       1030.968 805.3109 1256.626 683.4491 1378.487
## Dec 2021       1030.968 805.3109 1256.626 683.4491 1378.487
## Jan 2022       1030.968 805.3109 1256.626 683.4491 1378.487
## Feb 2022       1030.968 805.3109 1256.626 683.4491 1378.487
## Mar 2022       1030.968 805.3109 1256.626 683.4491 1378.487
```

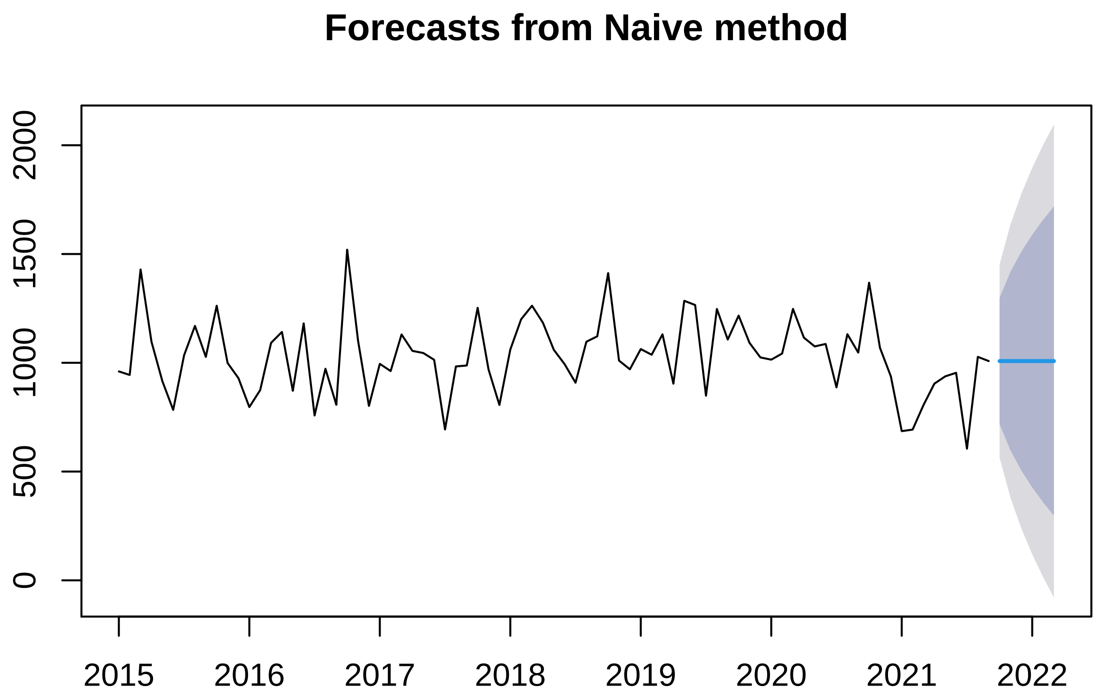
```r
accuracy(mean_forecast)
```

```
##                          ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -3.088305e-14 172.4845 130.8521 -2.998382 13.43832 0.9041035
##                   ACF1
## Training set 0.1469126
```

### Naive Forecast

### Forecasting is for the following 6 period month.

```r
naive_forecast <- naive(ts_tng,6)
plot(naive_forecast)
```




```r
summary(naive_forecast)
```

```
##
## Forecast method: Naive method
##
## Model Information:
## Call: naive(y = ts_tng, h = 6)
##
## Residual sd: 226.5522
##
## Error measures:
##                    ME     RMSE    MAE       MPE     MAPE     MASE       ACF1
## Training set 0.59475 226.5522 178.13 -2.387636 17.59058 1.230763 -0.4188375
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Oct 2021           1008 717.6616 1298.338 563.96578 1452.034
## Nov 2021           1008 597.3995 1418.600 380.04079 1635.959
## Dec 2021           1008 505.1192 1510.881 238.91018 1777.090
## Jan 2022           1008 427.3233 1588.677 119.93157 1896.068
## Feb 2022           1008 358.7837 1657.216  15.10931 2000.891
## Mar 2022           1008 296.8191 1719.181 -79.65726 2095.657
```

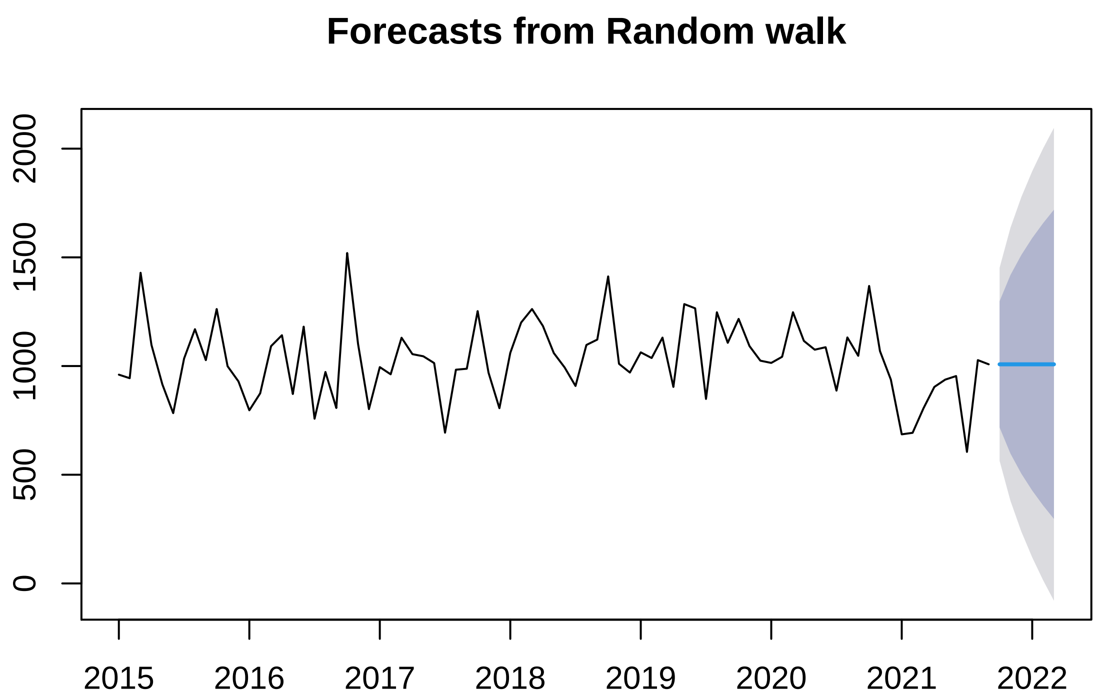
```r
accuracy(naive_forecast)
```

```
##                   ME     RMSE    MAE       MPE     MAPE     MASE       ACF1
## Training set 0.59475 226.5522 178.13 -2.387636 17.59058 1.230763 -0.4188375
```

### Random Walk Forecast

### This provides the same result as Naive Forecast

```r
rwf_forecast = rwf(ts_tng,6)
plot(rwf_forecast)
```




```r
summary(rwf_forecast)
```

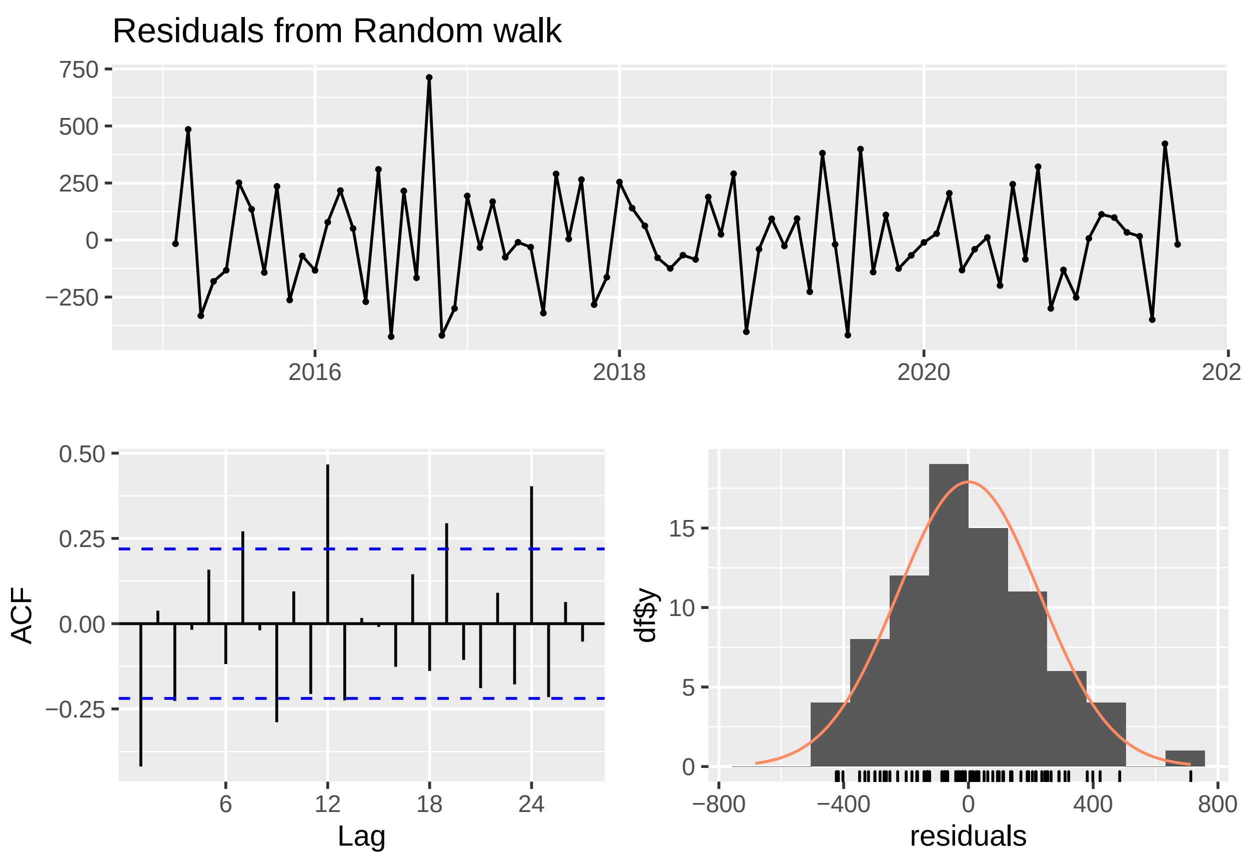
```
##
## Forecast method: Random walk
##
## Model Information:
## Call: rwf(y = ts_tng, h = 6)
##
## Residual sd: 226.5522
##
## Error measures:
##                   ME     RMSE    MAE       MPE     MAPE     MASE       ACF1
## Training set 0.59475 226.5522 178.13 -2.387636 17.59058 1.230763 -0.4188375
```

```
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Oct 2021           1008 717.6616 1298.338 563.96578 1452.034
## Nov 2021           1008 597.3995 1418.600 380.04079 1635.959
## Dec 2021           1008 505.1192 1510.881 238.91018 1777.090
## Jan 2022           1008 427.3233 1588.677 119.93157 1896.068
## Feb 2022           1008 358.7837 1657.216  15.10931 2000.891
## Mar 2022           1008 296.8191 1719.181 -79.65726 2095.657
```

```
accuracy(rwf_forecast)
```

```
##                   ME     RMSE    MAE       MPE     MAPE     MASE       ACF1
## Training set 0.59475 226.5522 178.13 -2.387636 17.59058 1.230763 -0.4188375
```

```
checkresiduals(rwf_forecast)
```

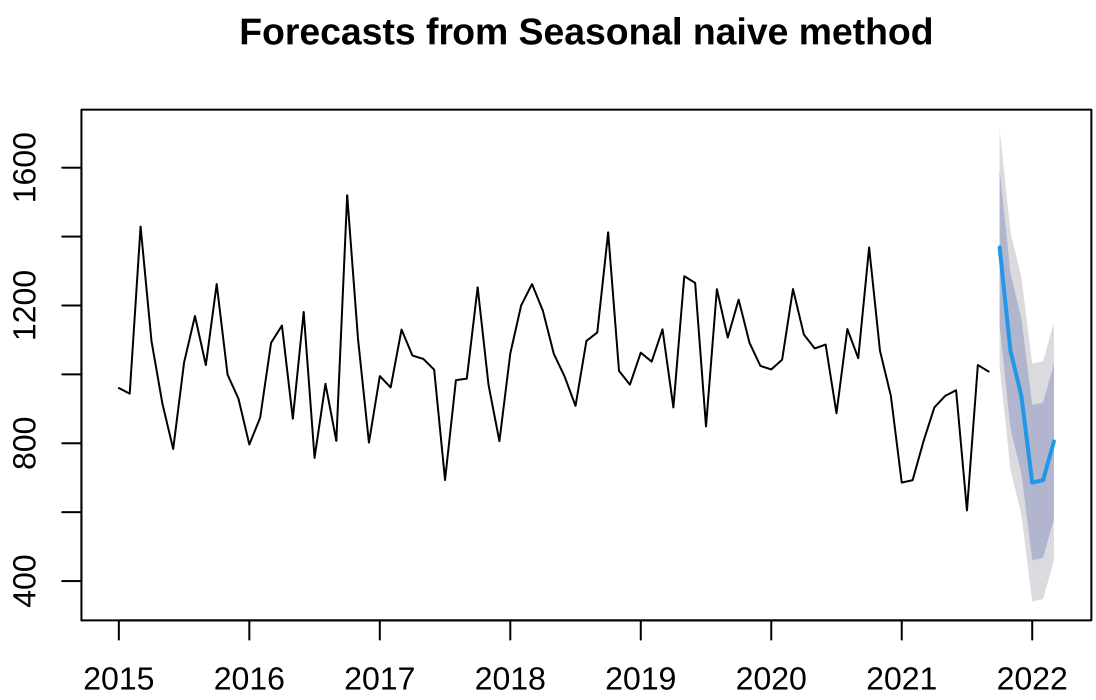



```
##
##  Ljung-Box test
##
## data:  Residuals from Random walk
## Q* = 69.401, df = 16, p-value = 1.27e-08
##
## Model df: 0.   Total lags used: 16
```

### Seasonal Naive Forecast

**It takes into account the previous year's seasonality over the same period as the forecast period:**

```
snaive_forecast = snaive(ts_tng,6)
plot(snaive_forecast)
```



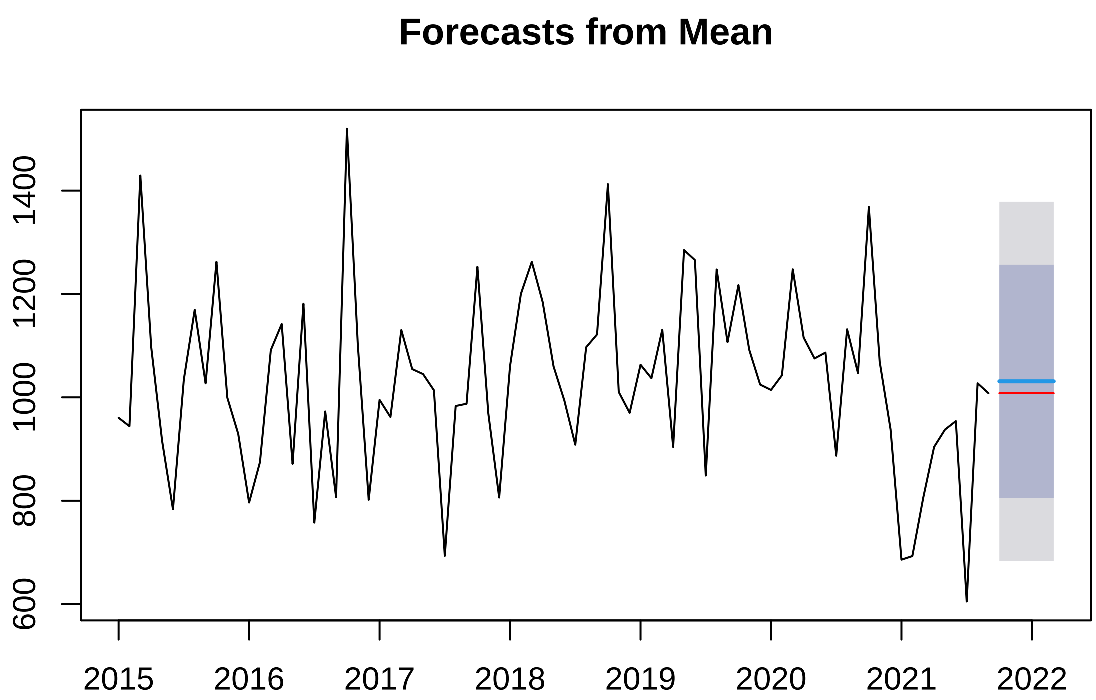

```
summary(snaive_forecast)
```

```
##
## Forecast method: Seasonal naive method
##
## Model Information:
## Call: snaive(y = ts_tng, h = 6)
##
## Residual sd: 176.0772
##
## Error measures:
##                    ME     RMSE      MAE       MPE     MAPE MASE      ACF1
## Training set -22.5658 176.0772 144.7313 -4.118856 14.79478    1 0.2503767
##
## Forecasts:
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2021        1368.48 1142.828 1594.132 1023.375 1713.585
## Nov 2021        1068.67  843.018 1294.322  723.565 1413.775
## Dec 2021         937.76  712.108 1163.412  592.655 1282.865
## Jan 2022         685.91  460.258  911.562  340.805 1031.015
## Feb 2022         692.88  467.228  918.532  347.775 1037.985
## Mar 2022         805.42  579.768 1031.072  460.315 1150.525
```

### Plotting mean and naive forecasting together

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

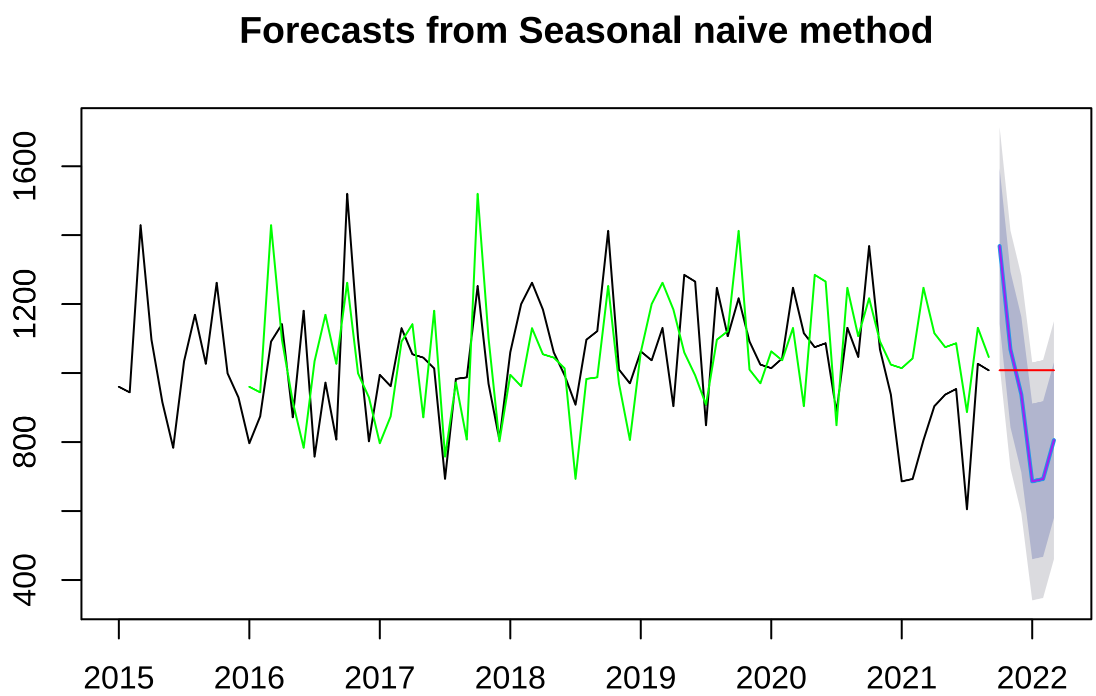



```
attributes(naive_forecast)
```

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

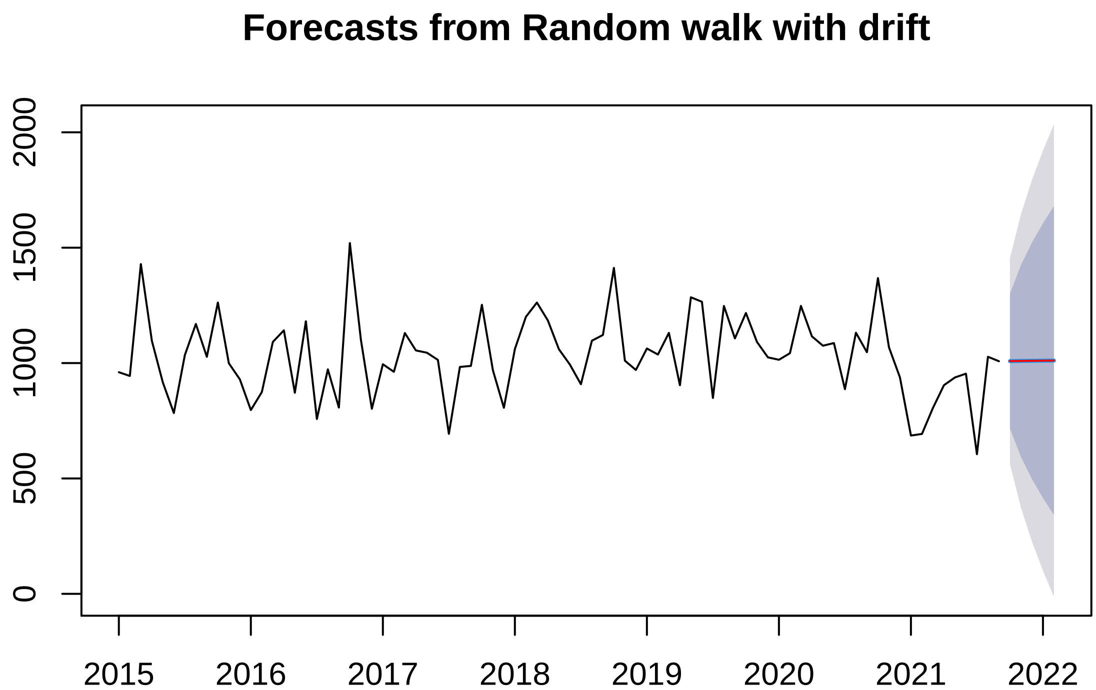
### Plotting other attributes

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



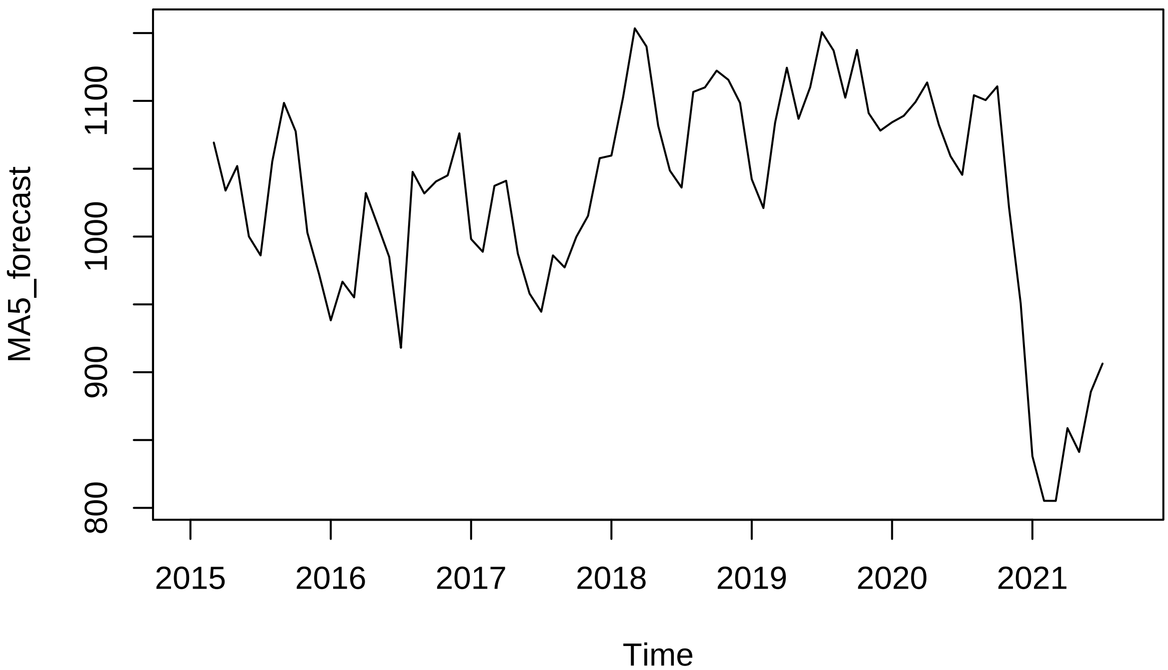

### Drift with RWF

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

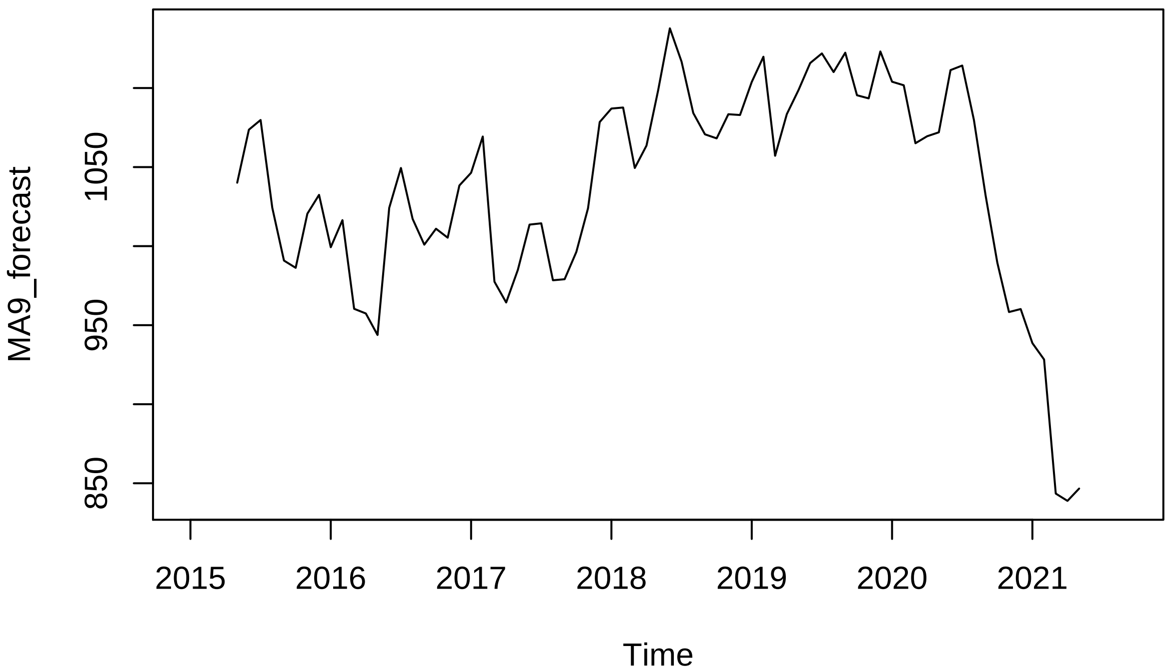



### Moving Average Forecast

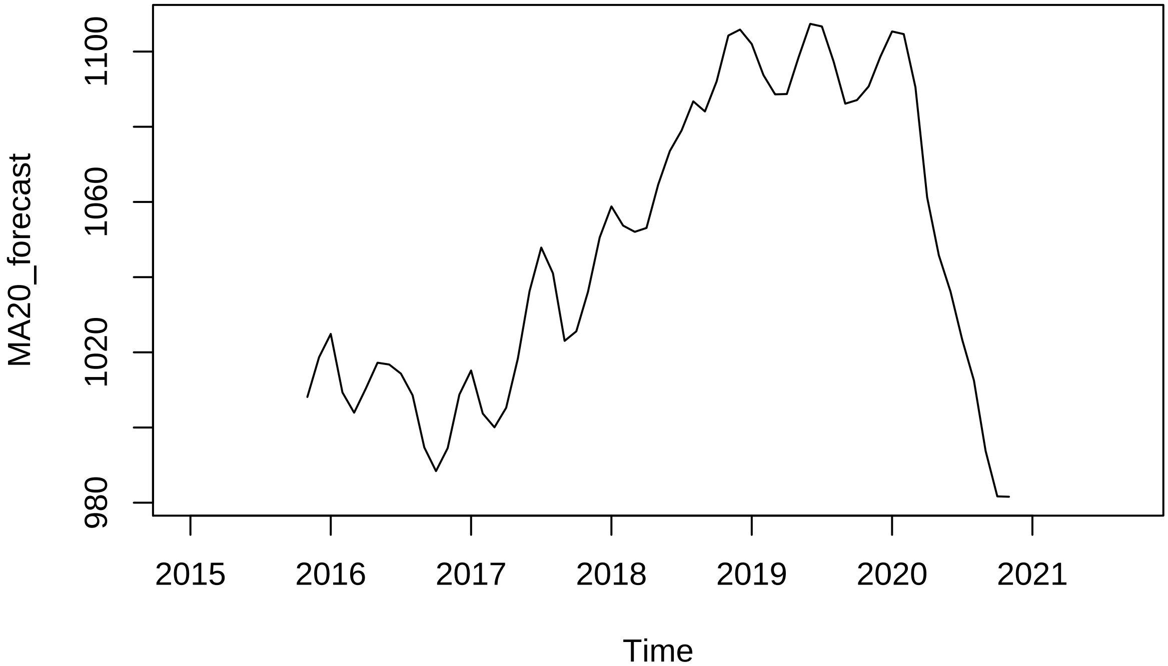
```
MA5_forecast <- ma(ts_tng,order=5)
MA9_forecast <- ma(ts_tng,order=9)
MA20_forecast <- ma(ts_tng,order=20)
plot(MA5_forecast)
```



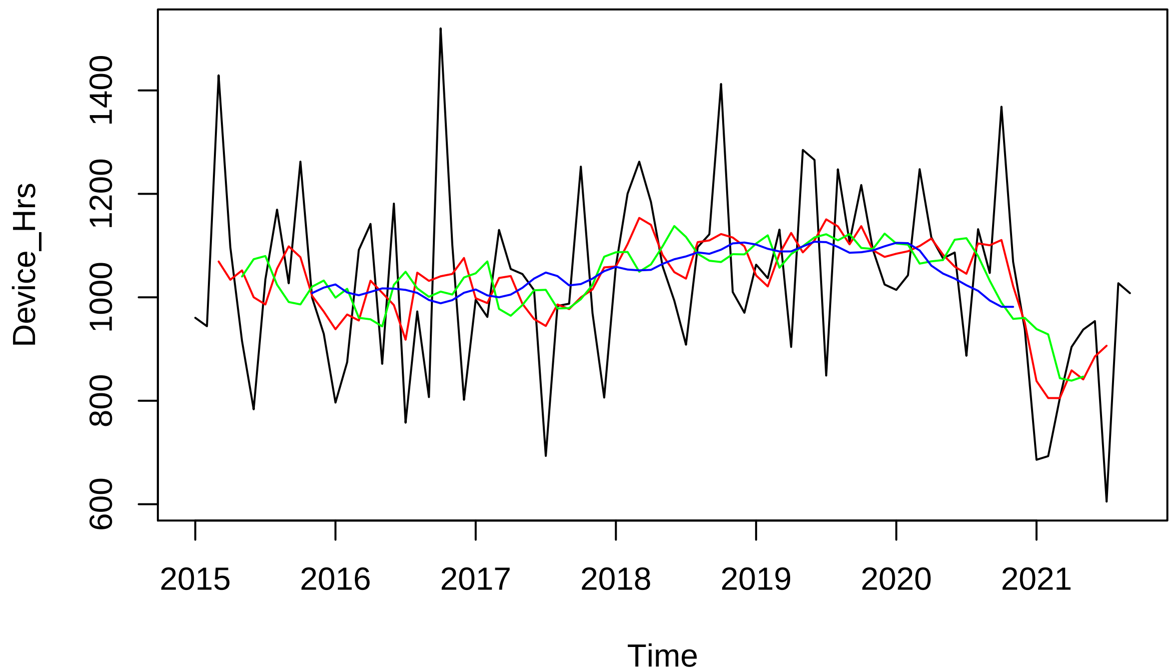

```r
plot(MA9_forecast)
```




```r
plot(MA20_forecast)
```




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




```r
summary(MA5_forecast)
```

```
##        V1
##  Min.   : 805.2
##  1st Qu.: 987.4
##  Median :1045.5
##  Mean   :1032.8
##  3rd Qu.:1098.5
##  Max.   :1153.5
##  NA's   :4
```

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

**ETS**

```r
ets(ts_tng)
```

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

```
##
##   Smoothing parameters:
##     alpha = 0.1873
##     gamma = 5e-04
##
##   Initial states:
##     l = 1046.8166
##     s = -133.0087 2.6368 310.9812 -27.0855 58.3452 -200.0136
##            32.4392 17.5141 23.4495 111.8025 -76.5201 -120.5406
##
##   sigma:  130.761
##
##      AIC     AICc      BIC
## 1160.066 1167.451 1195.983
```

## Holt Winters

```r
HoltWinters(ts_tng)
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = ts_tng)
##
## Smoothing parameters:
##  alpha: 0.2042742
##  beta : 0
##  gamma: 0.4025477
##
## Coefficients:
##            [,1]
## a    888.323744
## b     -3.477558
## s1   292.500947
## s2    18.646735
## s3   -92.804522
## s4  -143.172183
## s5  -102.347393
## s6    55.259057
## s7    32.761000
## s8    46.913686
## s9    75.266159
## s10 -203.941782
## s11  121.059342
## s12   54.788798
```

## SSE without trend and without seasonality

```r
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: 0.09370996
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 957.7362
```

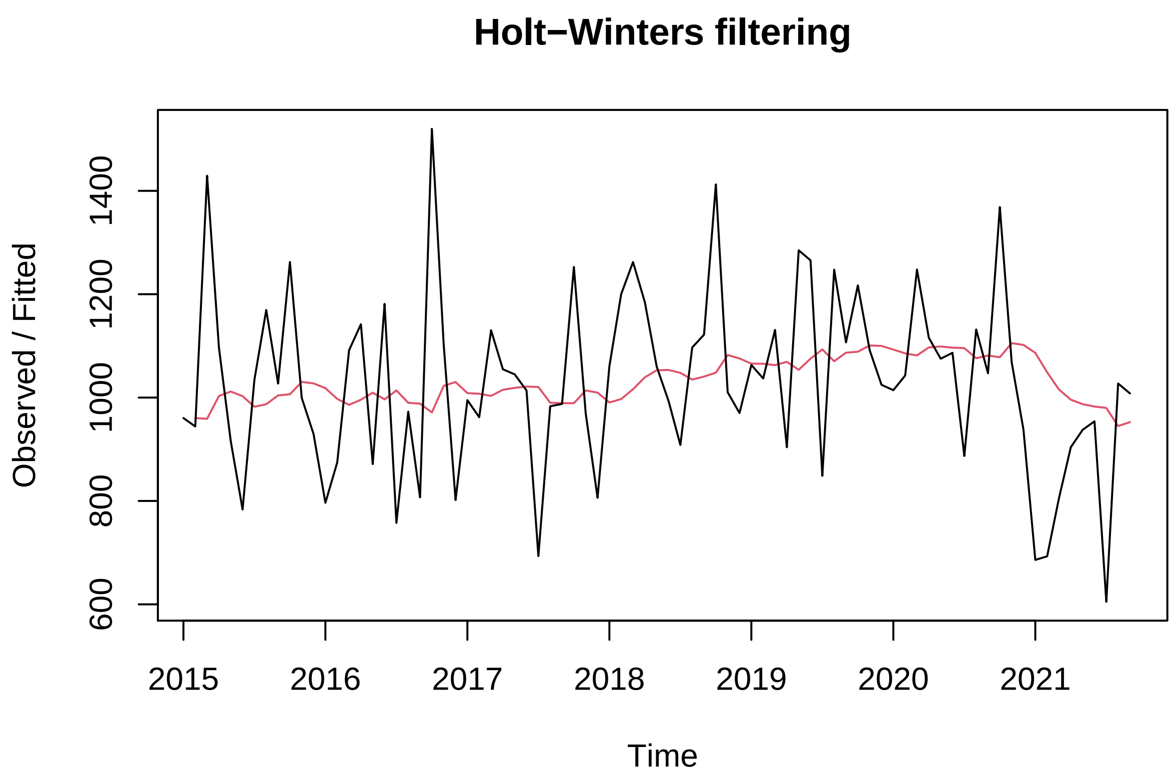
```
hw_forecast_level = HoltWinters(ts_tng,beta=FALSE,gamma=FALSE)
hw_forecast_level
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = ts_tng, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.09370996
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 957.7362
```

```
attributes(hw_forecast_level)
```

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

```
plot(hw_forecast_level)
```

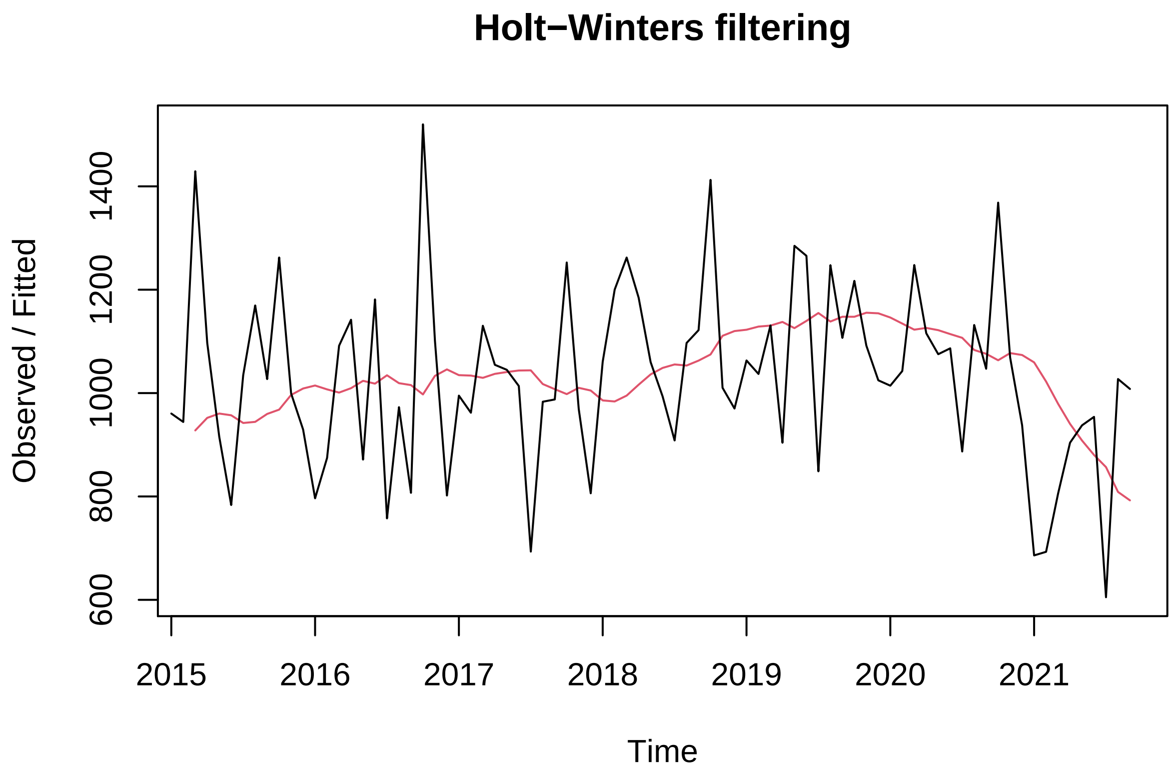



```r
hw_forecast_level$SSE
```

```
## [1] 2486244
```

### 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: 0.05537183
##  beta : 0.4649229
##  gamma: FALSE
##
## Coefficients:
##        [,1]
## a 804.35794
## b -22.89301
```

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

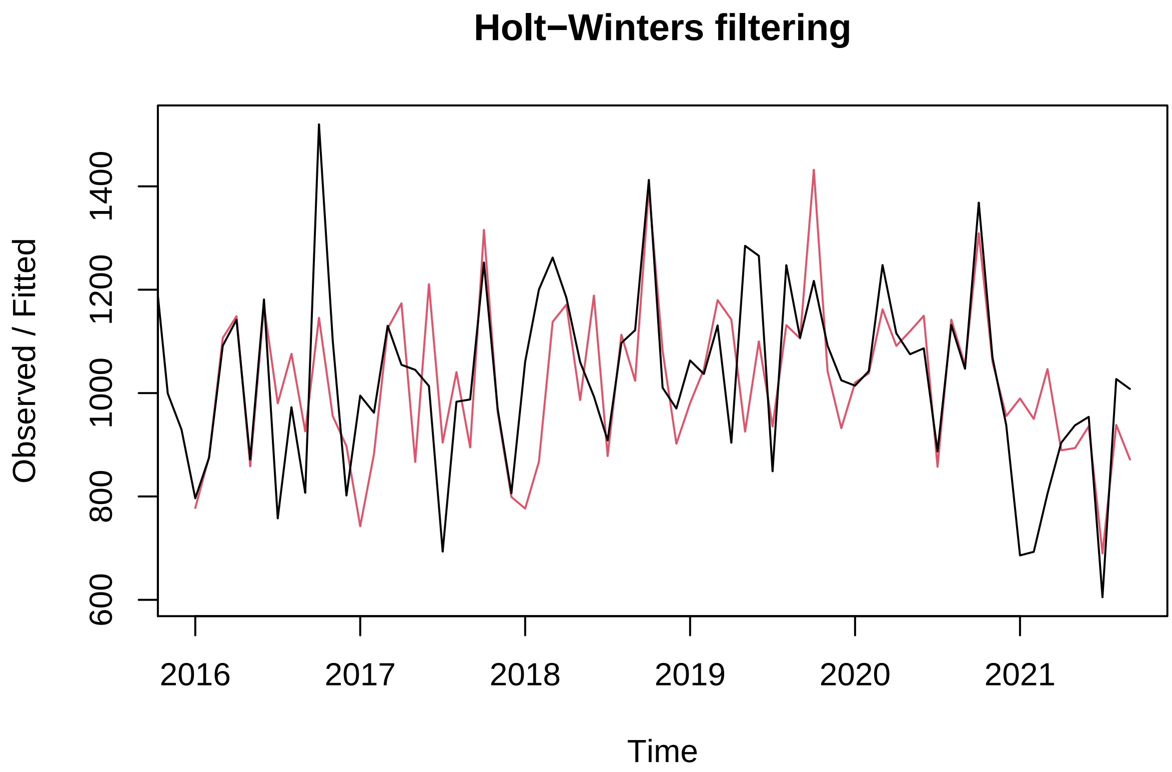
```
## [1] 2531400
```

### 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: 0.2042742
##  beta : 0
##  gamma: 0.4025477
##
## Coefficients:
##             [,1]
## a    888.323744
## b     -3.477558
## s1   292.500947
## s2    18.646735
## s3   -92.804522
## s4  -143.172183
## s5  -102.347393
## s6    55.259057
## s7    32.761000
## s8    46.913686
## s9    75.266159
## s10 -203.941782
## s11  121.059342
## s12   54.788798
```

```r
plot(hw_forecast_season)
```



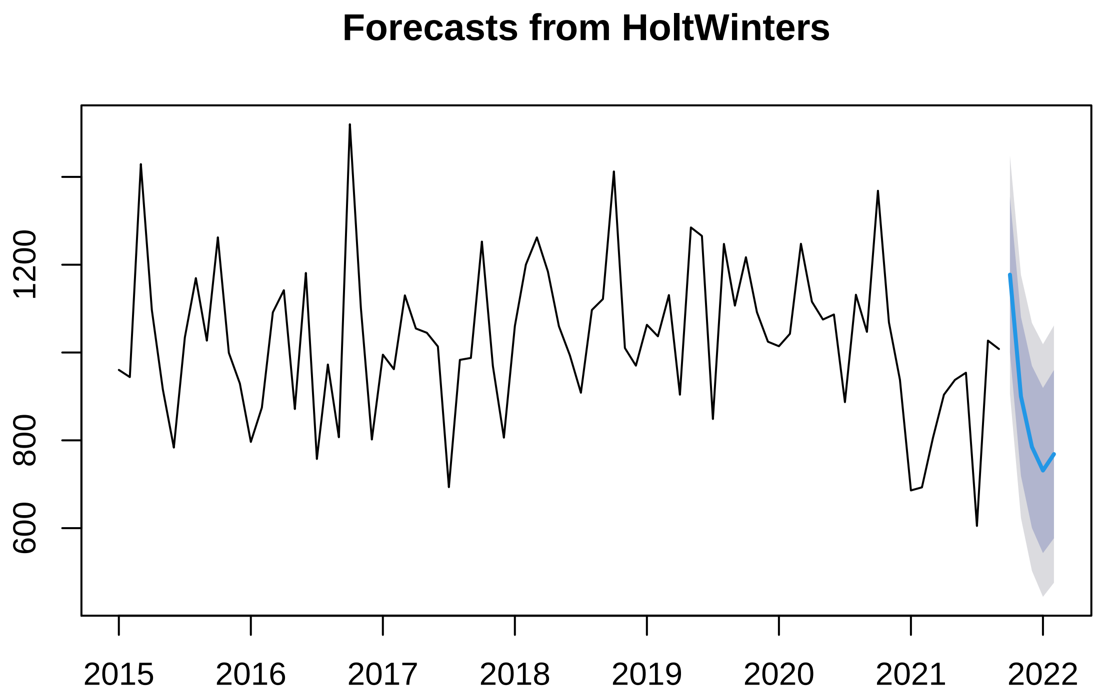

```r
hw_forecast_season$SSE
```

```
## [1] 1305128
```

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

```
##          Point Forecast     Lo 80     Hi 80    Lo 95    Hi 95
## Oct 2021      1177.3471 1000.0330 1354.6613 906.1686 1448.526
## Nov 2021       900.0154  719.0396 1080.9912 623.2368 1176.794
## Dec 2021       785.0865  600.5217  969.6514 502.8190 1067.354
## Jan 2022       731.2413  543.1559  919.3267 443.5895 1018.893
## Feb 2022       768.5886  577.0473  960.1298 475.6515 1061.526
```

```r
plot(hw_forecast_all)
```




```r
accuracy(hw_forecast_all)
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
## Training set 7.009793 137.5315 97.60734 -0.7217056 9.933645 0.6744038 0.1768506
```

**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**

**It is an exponential smoothing state model which can be used on univariate time series.**

```r
ets(ts_tng)
```

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

```
##     alpha = 0.1873
##     gamma = 5e-04
##
##   Initial states:
##     l = 1046.8166
##     s = -133.0087 2.6368 310.9812 -27.0855 58.3452 -200.0136
##            32.4392 17.5141 23.4495 111.8025 -76.5201 -120.5406
##
##   sigma:  130.761
##
##      AIC     AICc      BIC
## 1160.066 1167.451 1195.983
```

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

```
## NULL
```

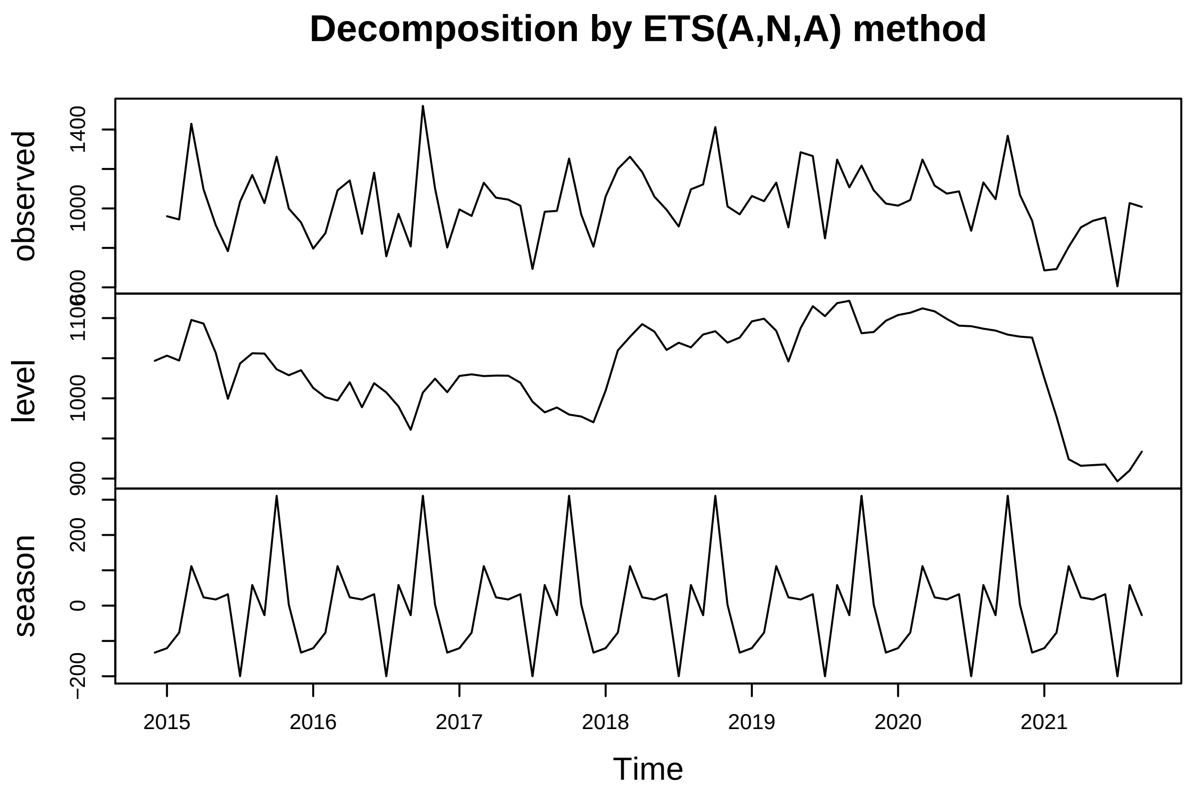
```r
attributes(ets_forecast)
```

```
## $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"
```

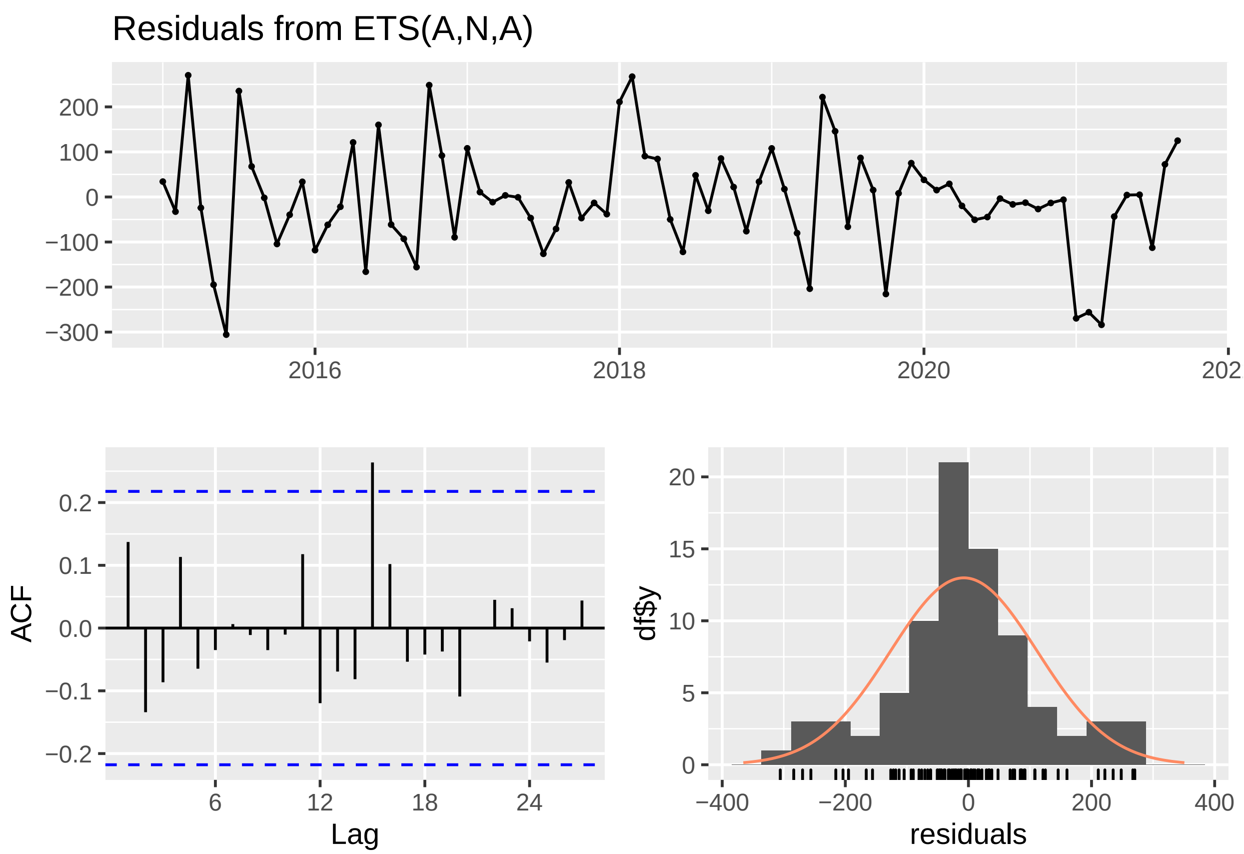
```r
ets_forecast$mse
```

```
## [1] 14143.16
```

```r
plot(ets_forecast)
```




```r
checkresiduals(ets_forecast)
```



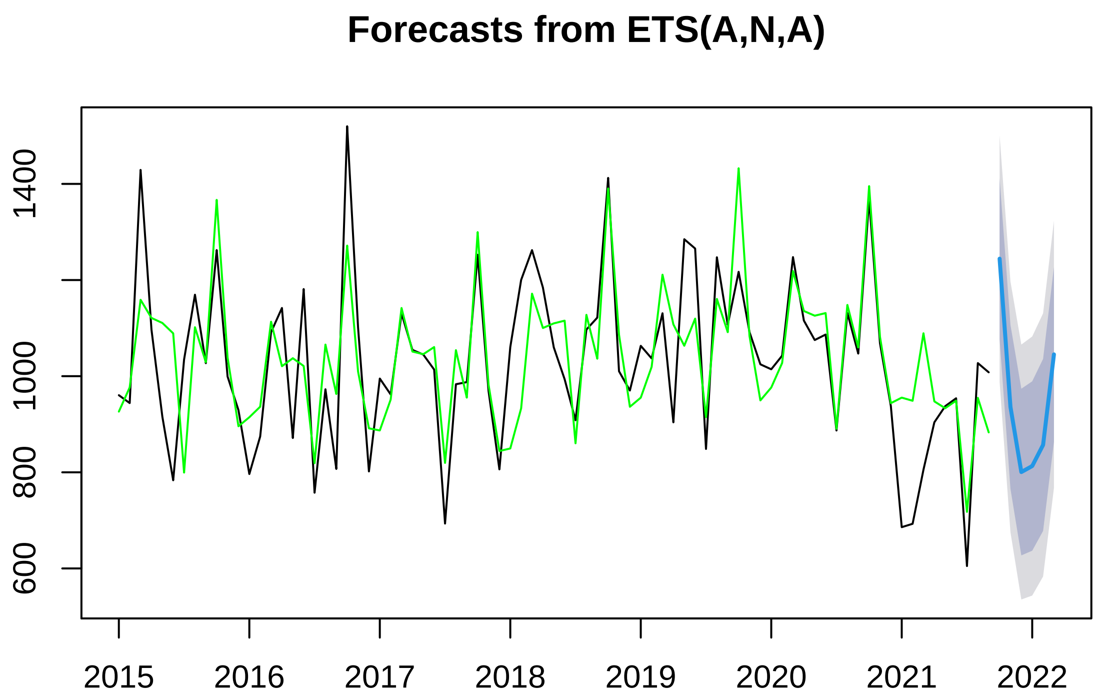

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,A)
## Q* = 17.831, df = 3, p-value = 0.0004765
##
## Model df: 14.   Total lags used: 17
```

### Forecast with Ets

```
forecast_ets = forecast.ets(ets_forecast, h=6)
forecast_ets
```

```
##          Point Forecast     Lo 80     Hi 80    Lo 95    Hi 95
## Oct 2021      1244.4582 1076.8813 1412.0352 988.1714 1500.745
## Nov 2021       936.1555  765.6630 1106.6480 675.4097 1196.901
## Dec 2021       800.5360  627.1770  973.8951 535.4063 1065.666
## Jan 2022       813.0567  636.8778  989.2356 543.6143 1082.499
## Feb 2022       856.9998  678.0454 1035.9541 583.3127 1130.687
## Mar 2022      1045.3390  863.6516 1227.0264 767.4721 1323.206
```

```r
plot(forecast_ets)
lines(forecast_ets$fitted, col="green")
```




```r
accuracy(forecast_ets)
```

```
##                     ME    RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -7.464588 118.925 87.09479 -2.062782 8.933903 0.6017689 0.1371752
```

### Decomposition

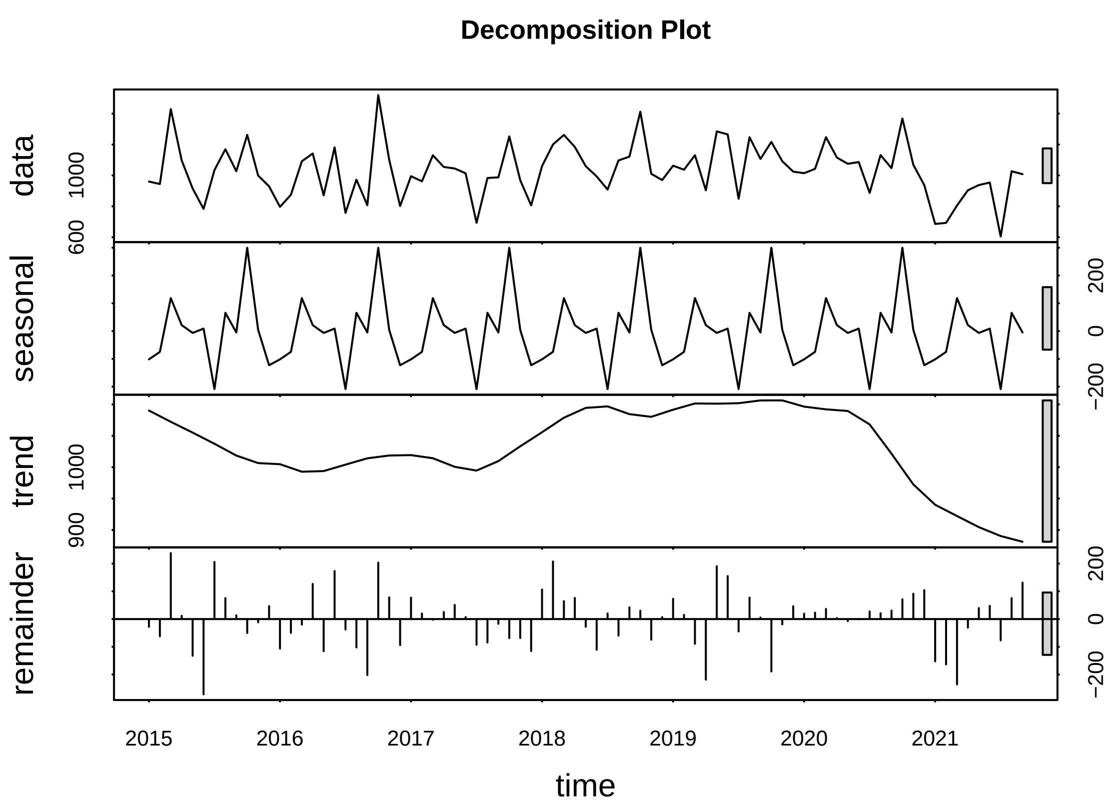
```r
stl_decomp = stl(ts_tng[,1], s.window = "periodic")
stl_decomp
```

```
##  Call:
##  stl(x = ts_tng[, 1], s.window = "periodic")
##
## Components
##             seasonal     trend   remainder
## Jan 2015 -101.616096 1090.1314  -28.095305
## Feb 2015  -74.748406 1081.1729  -62.344524
## Mar 2015  118.742007 1072.2145  238.163536
## Apr 2015   21.505862 1063.6194   11.874736
```

```
## May 2015   -6.645917 1055.0243 -132.528430
## Jun 2015    8.905372 1046.1375 -271.592890
## Jul 2015 -208.686323 1037.2507  205.955634
## Aug 2015   65.864995 1027.8787   75.756294
## Sep 2015   -4.963660 1018.5067   13.536926
## Oct 2015  299.912735 1012.5014  -50.094131
## Nov 2015    4.594318 1006.4961  -11.840377
## Dec 2015 -122.864825 1005.6190   46.665855
## Jan 2016 -101.616096 1004.7419 -106.705786
## Feb 2016  -74.748406  998.7557  -49.457309
## Mar 2016  118.742007  992.7695  -19.961553
## Apr 2016   21.505862  993.2847  127.049485
## May 2016   -6.645917  993.7998 -115.793841
## Jun 2016    8.905372  998.9182  173.386473
## Jul 2016 -208.686323 1004.0366  -37.760230
## Aug 2016   65.864995 1009.0751 -102.210120
## Sep 2016   -4.963660 1014.1137 -202.130036
## Oct 2016  299.912735 1016.3348  203.672482
## Nov 2016    4.594318 1018.5559   78.519812
## Dec 2016 -122.864825 1018.8641  -94.169316
## Jan 2017 -101.616096 1019.1724   77.533684
## Feb 2017  -74.748406 1016.7108   20.037648
## Mar 2017  118.742007 1014.2491   -2.751110
## Apr 2017   21.505862 1007.4084   25.795762
## May 2017   -6.645917 1000.5676   51.028270
## Jun 2017    8.905372  997.5684    7.256236
## Jul 2017 -208.686323  994.5691  -92.552814
## Aug 2017   65.864995 1002.1230  -84.737999
## Sep 2017   -4.963660 1009.6769  -17.073211
## Oct 2017  299.912735 1021.4338  -68.656536
## Nov 2017    4.594318 1033.1907  -68.475049
## Dec 2017 -122.864825 1044.4026 -115.437751
## Jan 2018 -101.616096 1055.6144  106.571676
## Feb 2018  -74.748406 1067.1877  207.810723
## Mar 2018  118.742007 1078.7609   64.747048
## Apr 2018   21.505862 1086.5468   76.397293
## May 2018   -6.645917 1094.3327  -27.766827
## Jun 2018    8.905372 1095.5383 -110.893666
## Jul 2018 -208.686323 1096.7438   20.312479
## Aug 2018   65.864995 1090.5785  -59.513504
## Sep 2018   -4.963660 1084.4132   42.300487
## Oct 2018  299.912735 1082.2782   30.279093
## Nov 2018    4.594318 1080.1432  -74.487490
## Dec 2018 -122.864825 1085.8073    7.177523
## Jan 2019 -101.616096 1091.4714   73.274662
## Feb 2019  -74.748406 1096.4281   15.270266
## Mar 2019  118.742007 1101.3848  -89.256853
## Apr 2019   21.505862 1101.2098 -218.745656
## May 2019   -6.645917 1101.0347  190.561175
## Jun 2019    8.905372 1101.4315  155.223163
## Jul 2019 -208.686323 1101.8282  -44.501866
## Aug 2019   65.864995 1104.0303   77.504656
## Sep 2019   -4.963660 1106.2325    5.571153
## Oct 2019  299.912735 1106.2607 -189.093420
```

```
## Nov 2019    4.594318 1106.2889  -19.043182
## Dec 2019 -122.864825 1101.3362   46.198606
## Jan 2020 -101.616096 1096.3836   19.552520
## Feb 2020  -74.748406 1094.2215   23.156938
## Mar 2020  118.742007 1092.0594   36.928634
## Apr 2020   21.505862 1090.7341    3.500038
## May 2020   -6.645917 1089.4088   -7.582922
## Jun 2020    8.905372 1078.7455   -1.020907
## Jul 2020 -208.686323 1068.0822   27.584093
## Aug 2020   65.864995 1044.7719   21.053066
## Sep 2020   -4.963660 1021.4616   30.542014
## Oct 2020  299.912735  997.0172   71.550064
## Nov 2020    4.594318  972.5728   91.502926
## Dec 2020 -122.864825  956.3659  104.258956
## Jan 2021 -101.616096  940.1590 -152.632887
## Feb 2021  -74.748406  931.2048 -163.576415
## Mar 2021  118.742007  922.2507 -235.572665
## Apr 2021   21.505862  913.4050  -30.910847
## May 2021   -6.645917  904.5593   39.706606
## Jun 2021    8.905372  897.4771   47.617563
## Jul 2021 -208.686323  890.3948  -76.708496
## Aug 2021   65.864995  885.8609   75.504111
## Sep 2021   -4.963660  881.3270  131.636692
```

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



```r
attributes(stl_decomp)
```

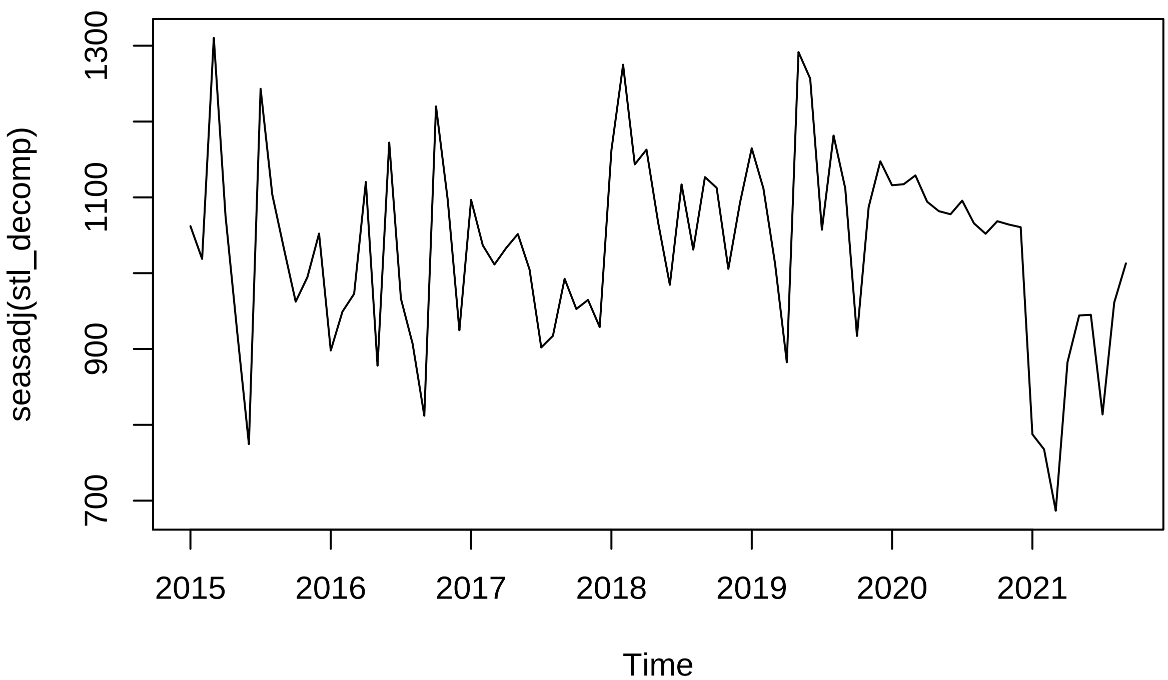
```
## $names
## [1] "time.series" "weights"     "call"        "win"         "deg"
## [6] "jump"        "inner"       "outer"
##
## $class
## [1] "stl"
```

### Seasonal Adjustment

```r
seasadj(stl_decomp)
```

```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 2015 1062.0361 1018.8284 1310.3780 1075.4941  922.4959  774.5446 1243.2063
## 2016  898.0361  949.2984  972.8080 1120.3341  878.0059 1172.3046  966.2763
## 2017 1096.7061 1036.7484 1011.4980 1033.2041 1051.5959 1004.8246  902.0163
## 2018 1162.1861 1274.9984 1143.5080 1162.9441 1066.5659  984.6446 1117.0563
## 2019 1164.7461 1111.6984 1012.1280  882.4641 1291.5959 1256.6546 1057.3263
## 2020 1115.9361 1117.3784 1128.9880 1094.2341 1081.8259 1077.7246 1095.6663
## 2021  787.5261  767.6284  686.6780  882.4941  944.2659  945.0946  813.6863
##            Aug       Sep       Oct       Nov       Dec
## 2015 1103.6350 1032.0437  962.4073  994.6557 1052.2848
## 2016  906.8650  811.9837 1220.0073 1097.0757  924.6948
## 2017  917.3850  992.6037  952.7773  964.7157  928.9648
## 2018 1031.0650 1126.7137 1112.5573 1005.6557 1092.9848
## 2019 1181.5350 1111.8037  917.1673 1087.2457 1147.5348
## 2020 1065.8250 1052.0037 1068.5673 1064.0757 1060.6248
## 2021  961.3650 1012.9637
```

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



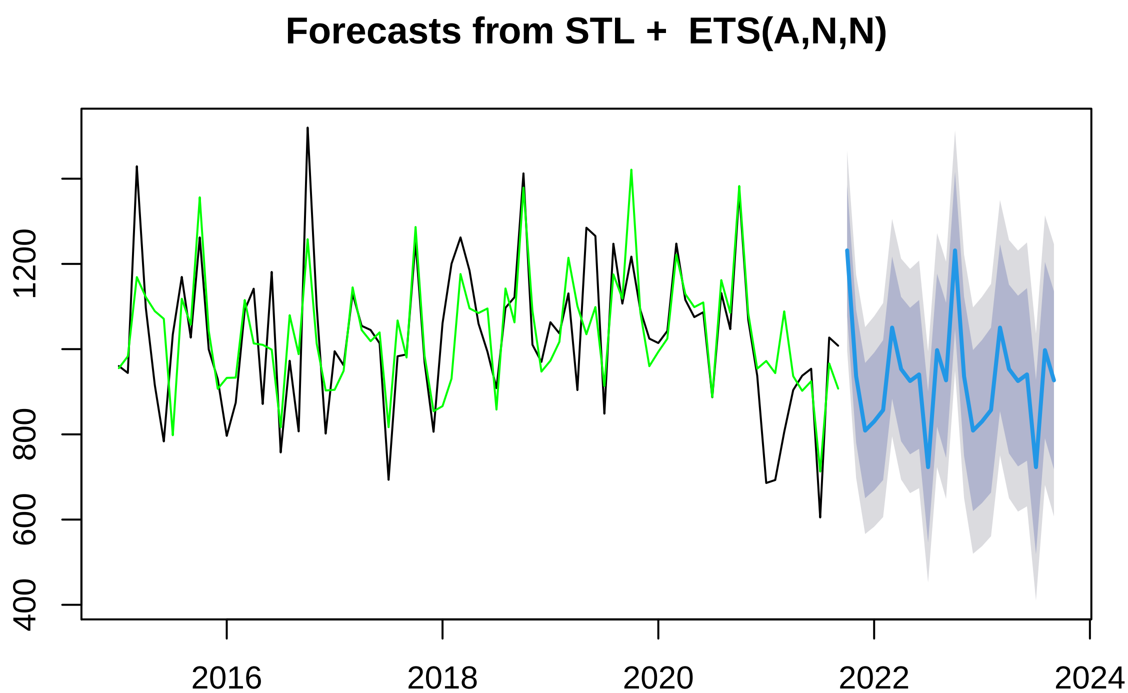

## Default Period Forecast

```
f_stl = forecast(stl_decomp,h = 24)
f_stl
```

```
##          Point Forecast     Lo 80     Hi 80    Lo 95     Hi 95
## Oct 2021      1231.6897 1078.5639 1384.8156 997.5039 1465.8755
## Nov 2021       936.3713  780.4066 1092.3360 697.8439 1174.8987
## Dec 2021       808.9122  650.1594  967.6649 566.1208 1051.7036
## Jan 2022       830.1609  668.6682  991.6536 583.1791 1077.1427
## Feb 2022       857.0286  692.8417 1021.2155 605.9263 1108.1309
## Mar 2022      1050.5190  883.6813 1217.3567 795.3628 1305.6752
## Apr 2022       953.2829  783.8359 1122.7298 694.1361 1212.4296
## May 2022       925.1311  753.1145 1097.1477 662.0544 1188.2078
## Jun 2022       940.6824  766.1339 1115.2309 673.7335 1207.6312
## Jul 2022       723.0907  546.0465  900.1348 452.3250  993.8563
## Aug 2022       997.6420  818.1369 1177.1471 723.1126 1272.1714
## Sep 2022       926.8133  744.8806 1108.7461 648.5712 1205.0555
## Oct 2022      1231.6897 1047.3612 1416.0182 949.7836 1513.5958
## Nov 2022       936.3713  749.6779 1123.0648 650.8483 1221.8943
## Dec 2022       808.9122  619.8833  997.9410 519.8175 1098.0068
## Jan 2023       830.1609  638.8252 1021.4966 537.5382 1122.7836
## Feb 2023       857.0286  663.4135 1050.6437 560.9198 1153.1374
## Mar 2023      1050.5190  854.6510 1246.3870 750.9648 1350.0732
## Apr 2023       953.2829  755.1876 1151.3781 650.3223 1256.2434
## May 2023       925.1311  724.8334 1125.4288 618.8022 1231.4600
```

```
## Jun 2023       940.6824  738.2061 1143.1586 631.0217 1250.3431
## Jul 2023       723.0907  518.4591  927.7223 410.1337 1036.0477
## Aug 2023       997.6420  790.8775 1204.4065 681.4230 1313.8610
## Sep 2023       926.8133  717.9378 1135.6889 607.3657 1246.2609
```

```r
plot(f_stl)
lines(f_stl$fitted,col = "green")
```

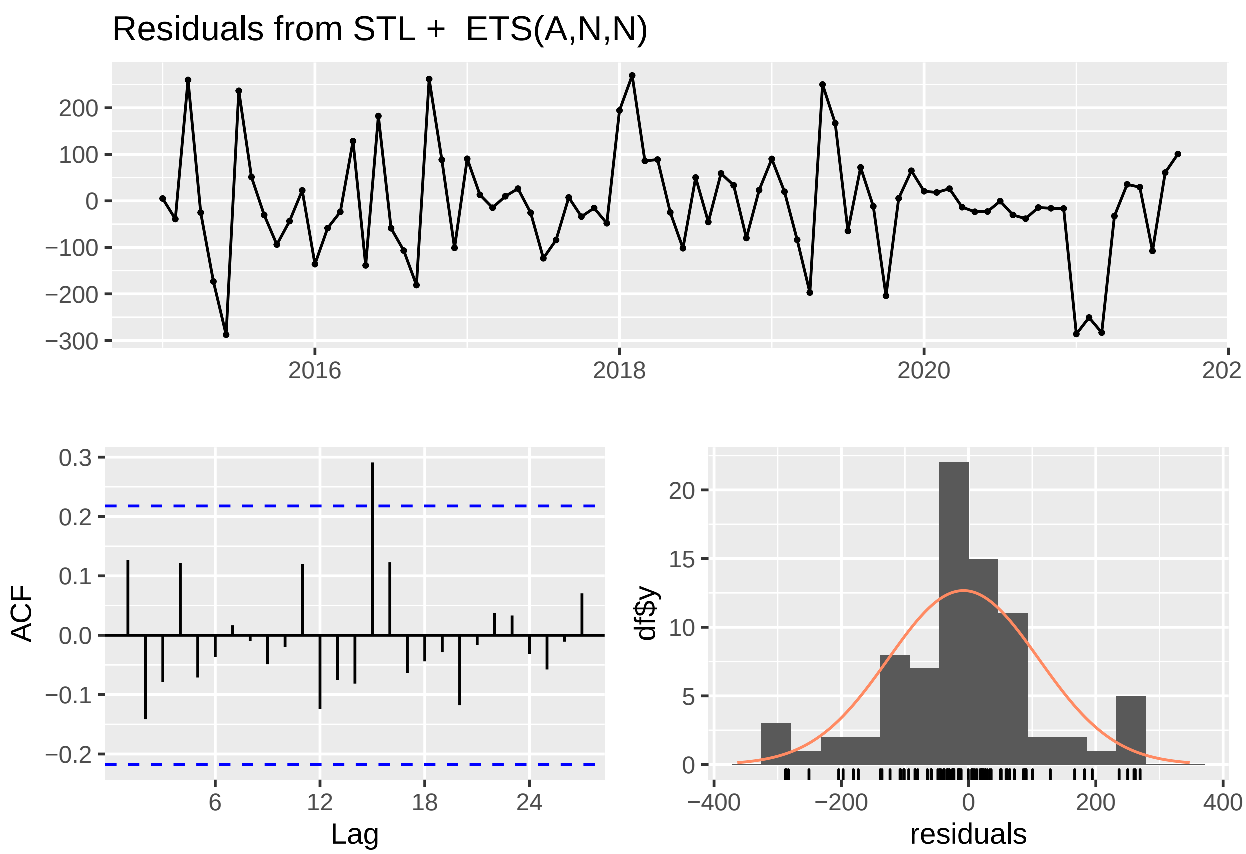



```r
accuracy(f_stl)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -7.989924 118.0004 85.69862 -2.139973 8.800883 0.5921222 0.1270786
```

```r
checkresiduals((f_stl))
```

```
## Warning in checkresiduals((f_stl)): The fitted degrees of freedom is based on
## the model used for the seasonally adjusted data.
```




```
##
##  Ljung-Box test
##
## data:  Residuals from STL +  ETS(A,N,N)
## Q* = 20.057, df = 14, p-value = 0.1284
##
## Model df: 2.   Total lags used: 16
```

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

```
ndiffs(ts_tng)
```

```
## [1] 0
```

```
tng_arima = auto.arima(ts_tng,trace=TRUE,stepwise = FALSE)
```

```
##
##  ARIMA(0,0,0)                with zero mean     : 1358.152
##  ARIMA(0,0,0)                with non-zero mean : 1068.372
##  ARIMA(0,0,0)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(0,0,1)[12] with non-zero mean : 1060.312
```

```
##  ARIMA(0,0,0)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(0,0,2)[12] with non-zero mean : 1055.327
##  ARIMA(0,0,0)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,0)[12] with non-zero mean : 1054.996
##  ARIMA(0,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(1,0,2)[12] with non-zero mean : 1057.553
##  ARIMA(0,0,0)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 1054.334
##  ARIMA(0,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)             with zero mean     : 1280.149
##  ARIMA(0,0,1)             with non-zero mean : 1068.855
##  ARIMA(0,0,1)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : 1059.914
##  ARIMA(0,0,1)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(0,0,2)[12] with non-zero mean : 1053.235
##  ARIMA(0,0,1)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,0)[12] with non-zero mean : 1053.263
##  ARIMA(0,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,0)[12] with non-zero mean : 1050.763
##  ARIMA(0,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,1)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)             with zero mean     : 1218.392
##  ARIMA(0,0,2)             with non-zero mean : 1070.658
##  ARIMA(0,0,2)(0,0,1)[12] with zero mean     : 1179.066
##  ARIMA(0,0,2)(0,0,1)[12] with non-zero mean : 1061.338
##  ARIMA(0,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(0,0,2)[12] with non-zero mean : 1053.62
##  ARIMA(0,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,0)[12] with non-zero mean : 1054.113
##  ARIMA(0,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,0)[12] with non-zero mean : 1051.029
##  ARIMA(0,0,2)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,2)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)             with zero mean     : 1197.963
##  ARIMA(0,0,3)             with non-zero mean : 1067.564
##  ARIMA(0,0,3)(0,0,1)[12] with zero mean     : 1171.957
##  ARIMA(0,0,3)(0,0,1)[12] with non-zero mean : 1060.795
##  ARIMA(0,0,3)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(0,0,2)[12] with non-zero mean : 1053.415
```

```
##  ARIMA(0,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,0)[12] with non-zero mean : 1055.043
##  ARIMA(0,0,3)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,3)(2,0,0)[12] with non-zero mean : 1052.405
##  ARIMA(0,0,4)            with zero mean     : 1175.072
##  ARIMA(0,0,4)            with non-zero mean : 1068.493
##  ARIMA(0,0,4)(0,0,1)[12] with zero mean     : 1151.146
##  ARIMA(0,0,4)(0,0,1)[12] with non-zero mean : 1062.416
##  ARIMA(0,0,4)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(0,0,4)(1,0,0)[12] with non-zero mean : 1056.86
##  ARIMA(0,0,5)            with zero mean     : Inf
##  ARIMA(0,0,5)            with non-zero mean : 1067.695
##  ARIMA(1,0,0)            with zero mean     : 1114.543
##  ARIMA(1,0,0)            with non-zero mean : 1068.78
##  ARIMA(1,0,0)(0,0,1)[12] with zero mean     : 1102.209
##  ARIMA(1,0,0)(0,0,1)[12] with non-zero mean : 1059.639
##  ARIMA(1,0,0)(0,0,2)[12] with zero mean     : 1089.956
##  ARIMA(1,0,0)(0,0,2)[12] with non-zero mean : 1052.478
##  ARIMA(1,0,0)(1,0,0)[12] with zero mean     : 1089.573
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 1052.587
##  ARIMA(1,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 1049.418
##  ARIMA(1,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)            with zero mean     : Inf
##  ARIMA(1,0,1)            with non-zero mean : 1070.99
##  ARIMA(1,0,1)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(0,0,1)[12] with non-zero mean : 1061.908
##  ARIMA(1,0,1)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(0,0,2)[12] with non-zero mean : 1054.782
##  ARIMA(1,0,1)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,0)[12] with non-zero mean : 1054.796
##  ARIMA(1,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 1051.563
##  ARIMA(1,0,1)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,1)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)            with zero mean     : Inf
##  ARIMA(1,0,2)            with non-zero mean : 1071.446
##  ARIMA(1,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,1)[12] with non-zero mean : 1062.6
##  ARIMA(1,0,2)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(0,0,2)[12] with non-zero mean : 1054.58
```

```
##  ARIMA(1,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,0)[12] with non-zero mean : 1056.916
##  ARIMA(1,0,2)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,2)(2,0,0)[12] with non-zero mean : 1052.723
##  ARIMA(1,0,3)            with zero mean     : Inf
##  ARIMA(1,0,3)            with non-zero mean : 1069.33
##  ARIMA(1,0,3)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(0,0,1)[12] with non-zero mean : 1062.861
##  ARIMA(1,0,3)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,3)(1,0,0)[12] with non-zero mean : 1057.189
##  ARIMA(1,0,4)            with zero mean     : Inf
##  ARIMA(1,0,4)            with non-zero mean : 1067.336
##  ARIMA(2,0,0)            with zero mean     : 1102.013
##  ARIMA(2,0,0)            with non-zero mean : 1070.978
##  ARIMA(2,0,0)(0,0,1)[12] with zero mean     : 1090.513
##  ARIMA(2,0,0)(0,0,1)[12] with non-zero mean : 1061.901
##  ARIMA(2,0,0)(0,0,2)[12] with zero mean     : 1080.453
##  ARIMA(2,0,0)(0,0,2)[12] with non-zero mean : 1054.751
##  ARIMA(2,0,0)(1,0,0)[12] with zero mean     : 1079.69
##  ARIMA(2,0,0)(1,0,0)[12] with non-zero mean : 1054.748
##  ARIMA(2,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(1,0,2)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 1072.3
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)            with zero mean     : Inf
##  ARIMA(2,0,1)            with non-zero mean : 1072.995
##  ARIMA(2,0,1)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(0,0,1)[12] with non-zero mean : 1063.782
##  ARIMA(2,0,1)(0,0,2)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(0,0,2)[12] with non-zero mean : 1056.45
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,0)[12] with non-zero mean : 1056.42
##  ARIMA(2,0,1)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 1053.263
##  ARIMA(2,0,2)            with zero mean     : Inf
##  ARIMA(2,0,2)            with non-zero mean : 1066.816
##  ARIMA(2,0,2)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(0,0,1)[12] with non-zero mean : 1059.287
##  ARIMA(2,0,2)(1,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,2)(1,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,3) with zero mean     : Inf
##  ARIMA(2,0,3)            with non-zero mean : 1068.695
##  ARIMA(3,0,0)            with zero mean     : Inf
##  ARIMA(3,0,0)            with non-zero mean : 1069.827
##  ARIMA(3,0,0)(0,0,1)[12] with zero mean     : 1090.119
##  ARIMA(3,0,0)(0,0,1)[12] with non-zero mean : 1062.208
```

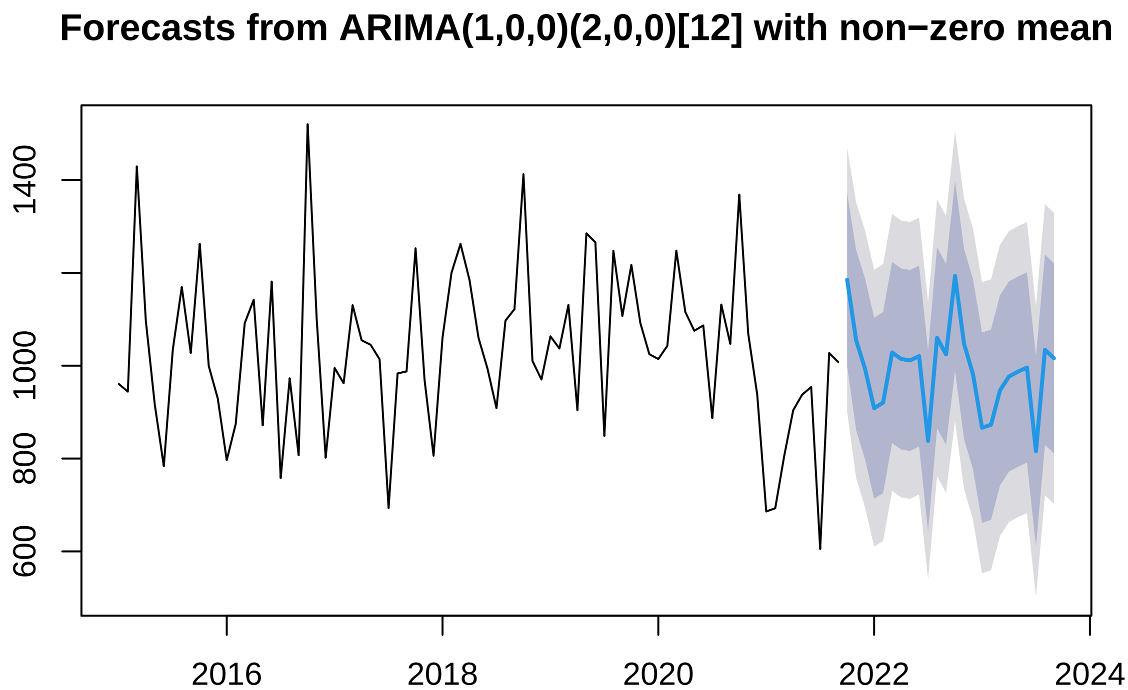
```
##  ARIMA(3,0,0)(0,0,2)[12] with zero mean     : 1080.572
##  ARIMA(3,0,0)(0,0,2)[12] with non-zero mean : 1055.635
##  ARIMA(3,0,0)(1,0,0)[12] with zero mean     : 1079.016
##  ARIMA(3,0,0)(1,0,0)[12] with non-zero mean : 1056.174
##  ARIMA(3,0,0)(1,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,0)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(3,0,0)(2,0,0)[12] with zero mean     : 1070.881
##  ARIMA(3,0,0)(2,0,0)[12] with non-zero mean : 1053.823
##  ARIMA(3,0,1)            with zero mean     : Inf
##  ARIMA(3,0,1)            with non-zero mean : 1072.066
##  ARIMA(3,0,1)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(3,0,1)(0,0,1)[12] with non-zero mean : 1064.446
##  ARIMA(3,0,1)(1,0,0)[12] with zero mean     : 1084.11
##  ARIMA(3,0,1)(1,0,0)[12] with non-zero mean : 1058.126
##  ARIMA(3,0,2)            with zero mean     : Inf
##  ARIMA(3,0,2)            with non-zero mean : 1069.003
##  ARIMA(4,0,0)            with zero mean     : Inf
##  ARIMA(4,0,0)            with non-zero mean : 1071.845
##  ARIMA(4,0,0)(0,0,1)[12] with zero mean     : Inf
##  ARIMA(4,0,0)(0,0,1)[12] with non-zero mean : 1064.157
##  ARIMA(4,0,0)(1,0,0)[12] with zero mean     : 1072.435
##  ARIMA(4,0,0)(1,0,0)[12] with non-zero mean : 1057.371
##  ARIMA(4,0,1)            with zero mean     : Inf
##  ARIMA(4,0,1)            with non-zero mean : 1070.895
##  ARIMA(5,0,0)            with zero mean     : Inf
##  ARIMA(5,0,0)            with non-zero mean : 1069.578
##
##
##
##  Best model: ARIMA(1,0,0)(2,0,0)[12] with non-zero mean
```

```r
forecast_arima = forecast(tng_arima)
summary(forecast_arima)
```

```
##
## Forecast method: ARIMA(1,0,0)(2,0,0)[12] with non-zero mean
##
## Model Information:
## Series: ts_tng
## ARIMA(1,0,0)(2,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1    sar1    sar2       mean
##       0.3015  0.3236  0.3478  1016.0933
## s.e.  0.1091  0.1164  0.1388    49.4108
##
## sigma^2 estimated as 21035:  log likelihood=-519.31
## AIC=1048.62   AICc=1049.42   BIC=1060.59
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -2.224397 141.4076 106.8018 -2.294505 10.96822 0.7379314
##                     ACF1
## Training set -0.01030866
```

```
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2021      1185.0722 999.2036 1370.941 900.8106 1469.334
## Nov 2021      1054.9412 860.8083 1249.074 758.0406 1351.842
## Dec 2021       992.3639 797.4971 1187.231 694.3409 1290.387
## Jan 2022       908.2076 713.2743 1103.141 610.0828 1206.332
## Feb 2022       920.5975 725.6581 1115.537 622.4634 1218.732
## Mar 2022      1028.4481 833.5081 1223.388 730.3131 1326.583
## Apr 2022      1014.4662 819.5262 1209.406 716.3312 1312.601
## May 2022      1011.2462 816.3061 1206.186 713.1111 1309.381
## Jun 2022      1020.5324 825.5924 1215.472 722.3974 1318.667
## Jul 2022       838.1384 643.1984 1033.078 540.0034 1136.273
## Aug 2022      1059.9068 864.9668 1254.847 761.7718 1358.042
## Sep 2022      1024.2386 829.2986 1219.179 726.1035 1322.374
## Oct 2022      1193.3554 989.3455 1397.365 881.3492 1505.362
## Nov 2022      1046.9542 842.1397 1251.769 733.7175 1360.191
## Dec 2022       981.1660 776.2786 1186.053 667.8177 1294.514
## Jan 2023       866.3262 661.4322 1071.220 552.9678 1179.685
## Feb 2023       872.7605 667.8658 1077.655 559.4011 1186.120
## Mar 2023       946.8107 741.9159 1151.705 633.4512 1260.170
## Apr 2023       976.5759 771.6812 1181.471 663.2165 1289.935
## May 2023       987.2283 782.3335 1192.123 673.8688 1300.588
## Jun 2023       995.9313 791.0365 1200.826 682.5718 1309.291
## Jul 2023       815.5056 610.6109 1020.400 502.1461 1128.865
## Aug 2023      1034.1466 829.2519 1239.041 720.7871 1347.506
## Sep 2023      1015.9142 811.0194 1220.809 702.5547 1329.274
```

```r
plot(forecast_arima)
```




**Here, we can see that the MAPE of stl_decompostion is the lowest i.e. 8%, so we consider the stl_decomp forecast as out best forecasting model.**