R Notebook

```
library(readxl)
FlightData <- read_excel("C:/RBS/Business Forecasting/Group Project/FlightData.xlsx")

library(TTR)
library(fpp2)
```

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

```
## -- Attaching packages ---------------------------------------------- fpp2 2.4 --
```

```
## v ggplot2   3.3.5     v fma       2.4
## v forecast  8.15      v expsmooth 2.3
```

```
##
```

```
library(fpp)
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
## Loading required package: tseries
```

```
##
## Attaching package: 'fpp'
```

```
## The following objects are masked from 'package:fpp2':
##
##     ausair, ausbeer, austa, austourists, debitcards, departures,
##     elecequip, euretail, guinearice, oil, sunspotarea, usmelec
```

```
library(ggplot2)
```

```
str(FlightData)
```

```
## tibble [80 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Year                        : chr [1:80] "2015-01" "2015-02" "2015-03" "2015-04" ...
##  $ Column1                     : POSIXct[1:80], format: "2015-01-31" "2015-02-28" ...
##  $ Quarter                     : chr [1:80] "Q1" "Q1" "Q1" "Q2" ...
##  $ Month                       : chr [1:80] "Jan" "Feb" "Mar" "Apr" ...
##  $ Device_Hrs                  : num [1:80] 960 944 1429 1097 916 ...
##  $ DH_Prev_Year                : chr [1:80] "NA" "NA" "NA" "NA" ...
##  $ DH_YoY_Change               : chr [1:80] "NA" "NA" "NA" "NA" ...
##  $ DH_YoY_Ch_Per               : chr [1:80] "NA" "NA" "NA" "NA" ...
##  $ Total_Inst_Hrs              : num [1:80] 1701 1614 2533 2152 1695 ...
##  $ Total_Inst_Hrs_Prev_Year    : chr [1:80] "NA" "NA" "NA" "NA" ...
##  $ Inst_Hrs_YoY_Change         : chr [1:80] "NA" "NA" "NA" "NA" ...
##  $ Total_Inst_Hrs_YoY_Change_Per2: chr [1:80] "NA" "NA" "NA" "NA" ...
```

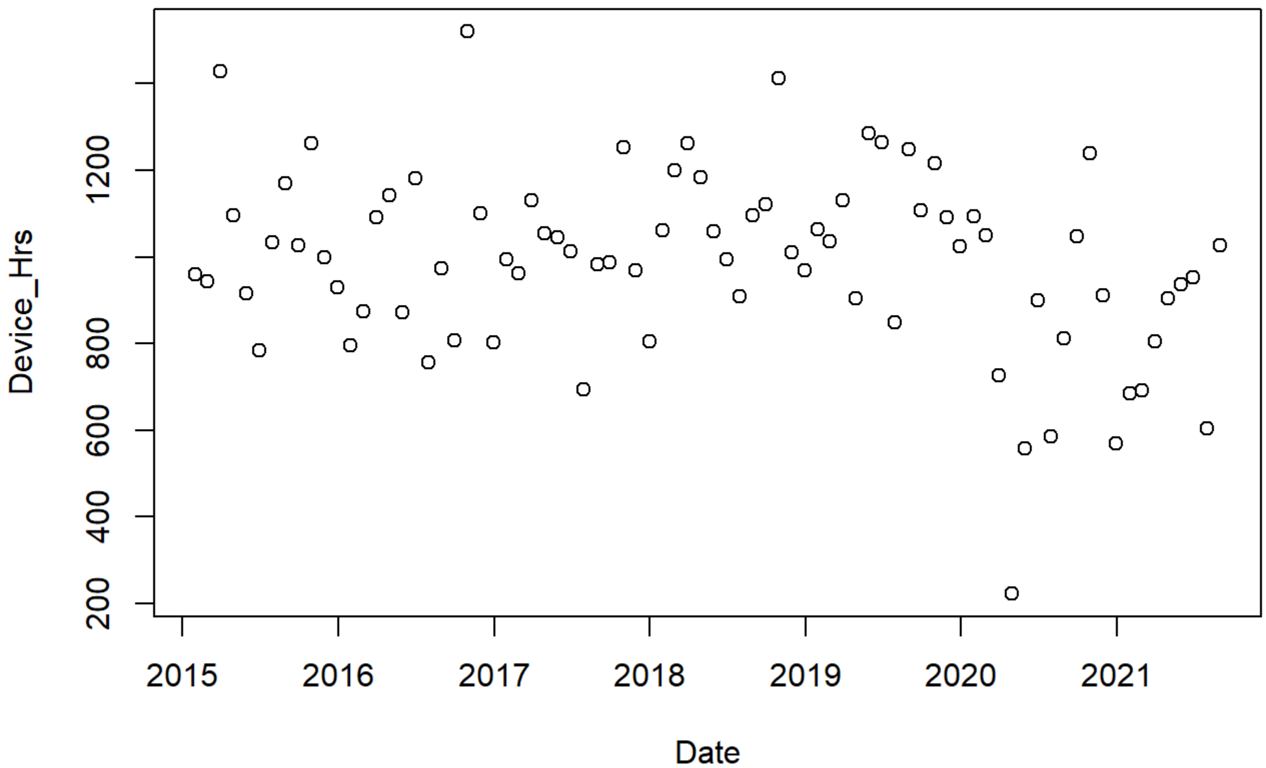
```
names(FlightData)[2] = "Date"
```

View(FlightData)

```
FD4 = subset(FlightData, select = c(Year, Month, Device_Hrs))
```

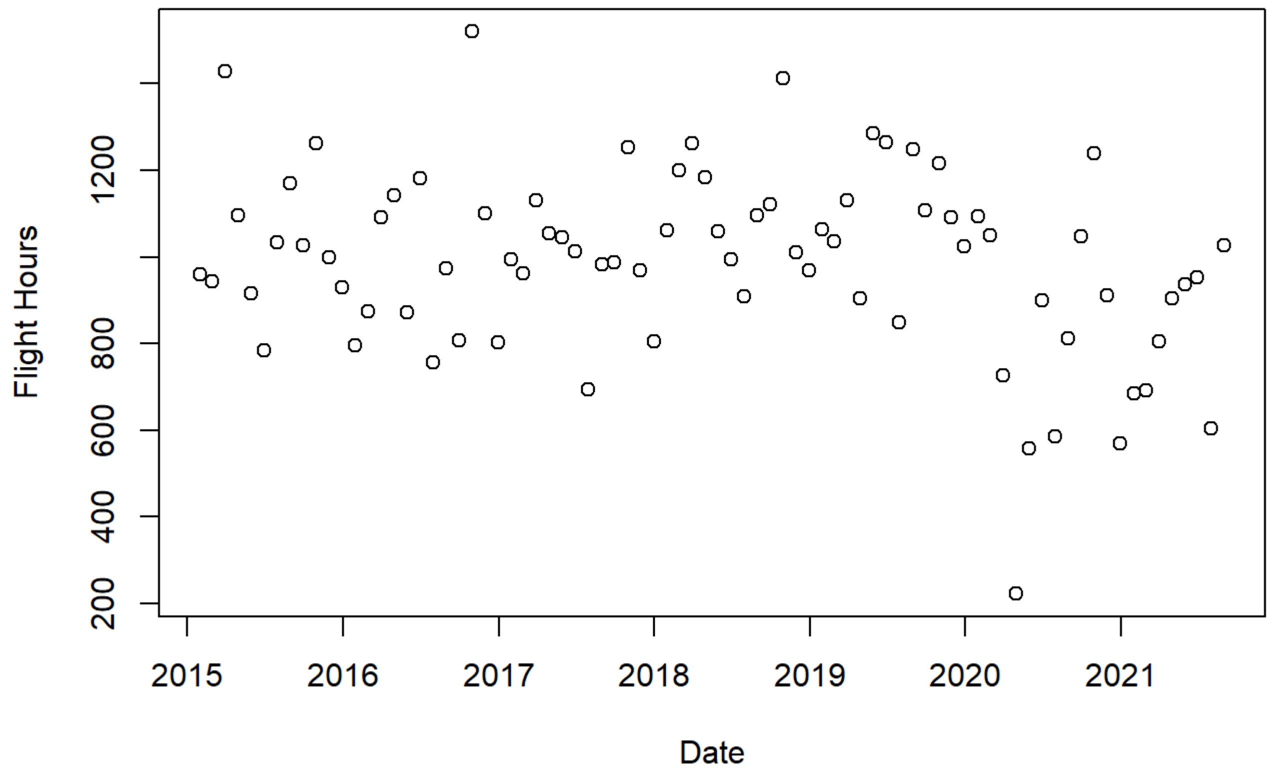
```
FD2 = subset(FlightData, select = c(Date, Device_Hrs))
```

```
plot(FD2)
```



Cleaner Plot

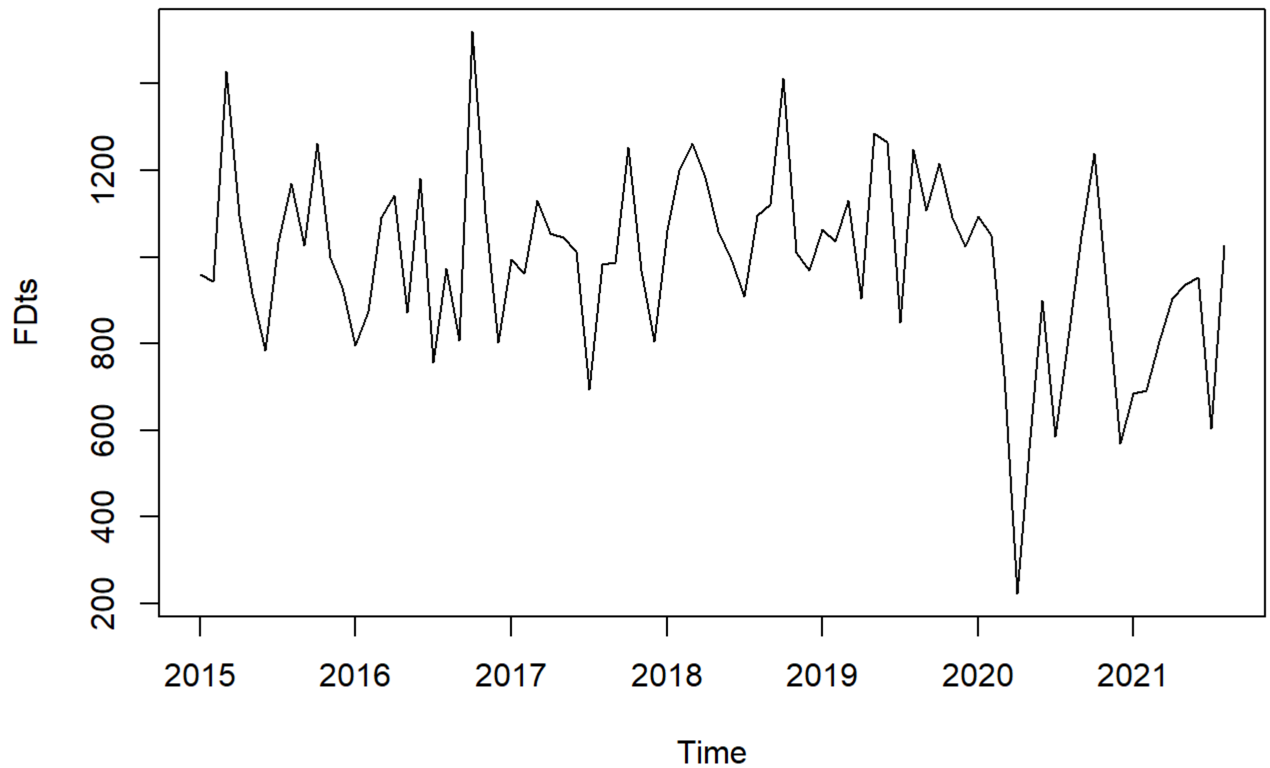
```
plot(FD2, xlab="Date", ylab="Flight Hours")
```




```
library(fpp)
library(fpp2)
library(TTR)
```

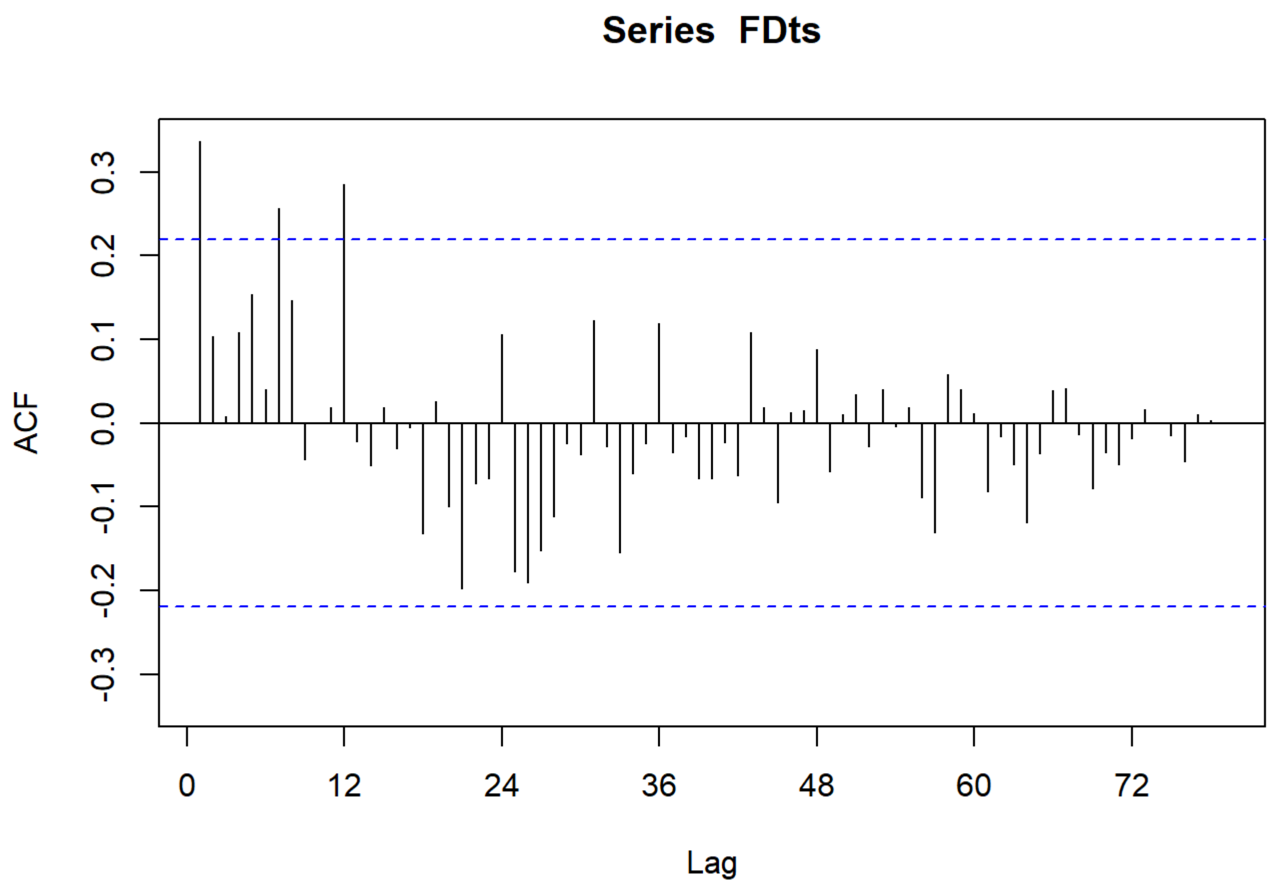
```
FDts = ts(FD2$Device_Hrs,frequency=12, start = c(2015,1))
```

```
plot(FDts)
```



ACF Autocorrelation of the data

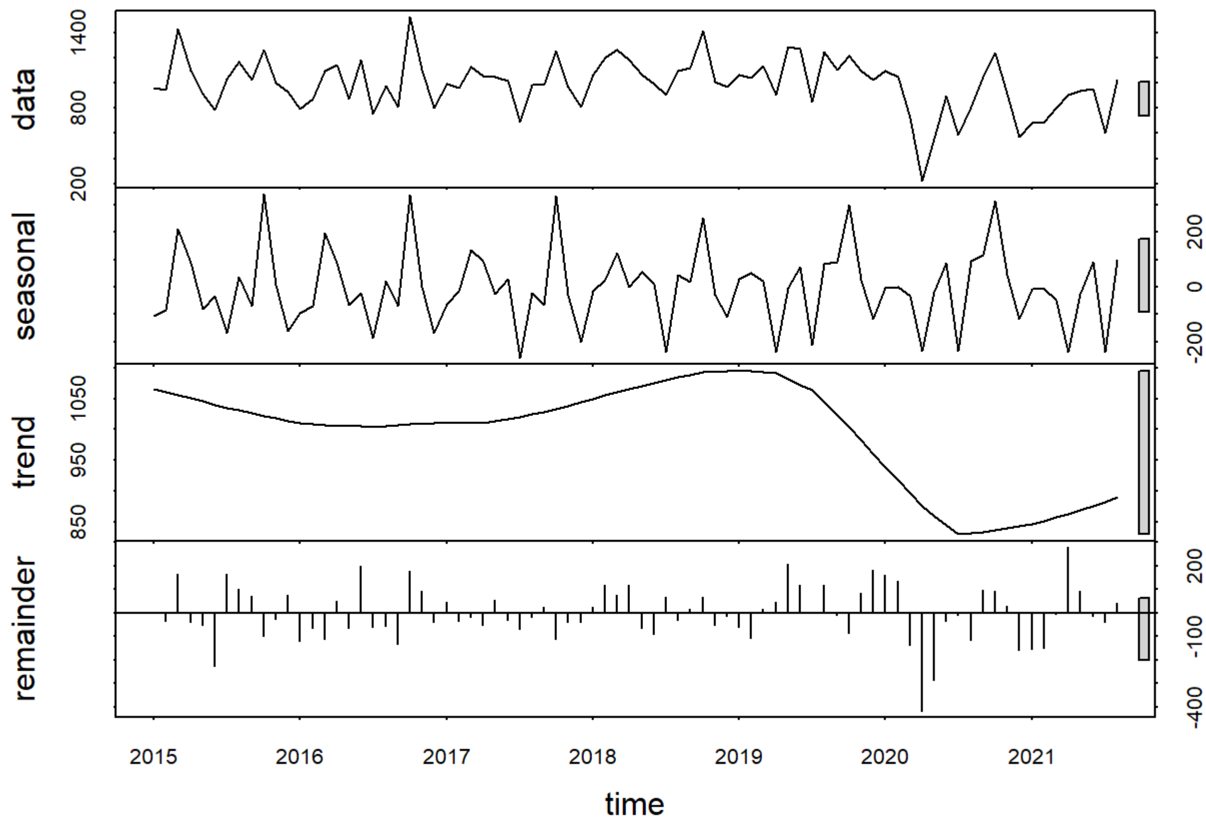
```
Acf(FDts, lag.max = 80)
```




```
fit= stl(FDts, s.window = 5)
```

plot(fit)

```
plot(fit)
```



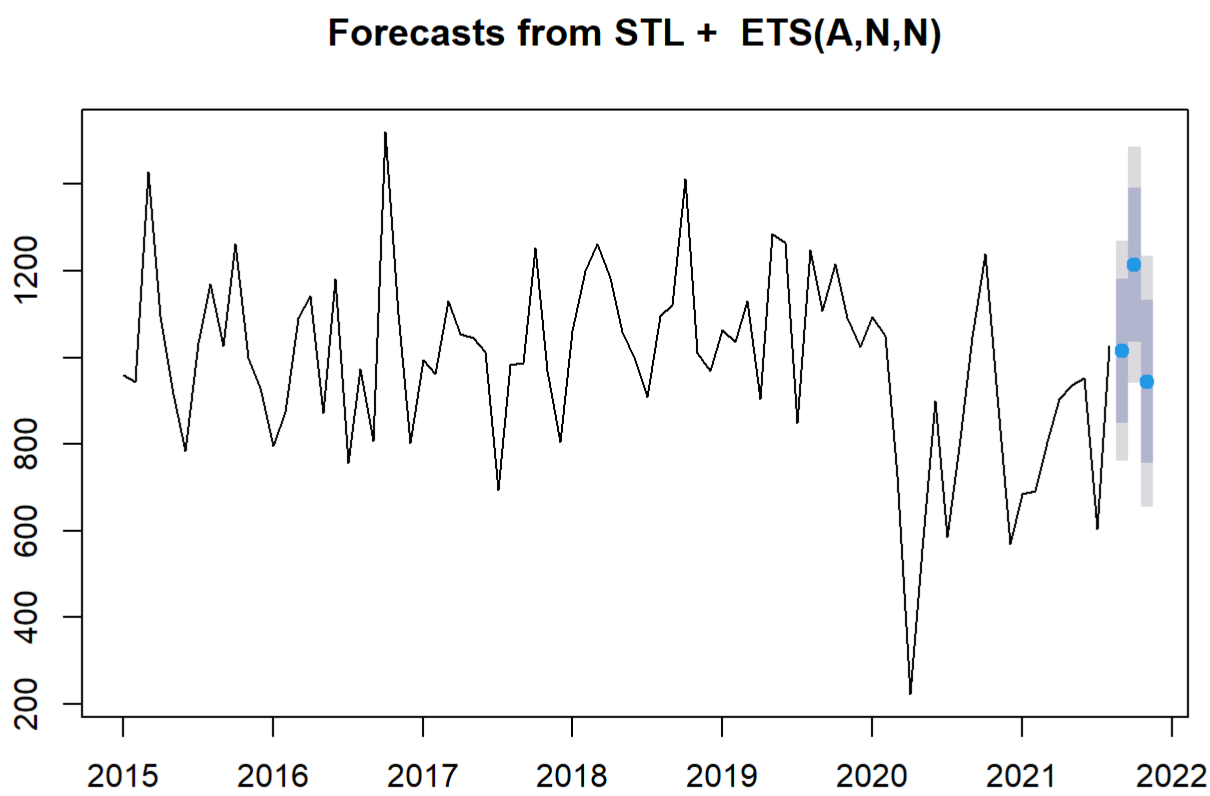

```
class(fit)
```

```
## [1] "stl"
```

3 Period Forecast

```
FDfore3 = forecast(fit, h=3)
```

```
plot(FDfore3)
```



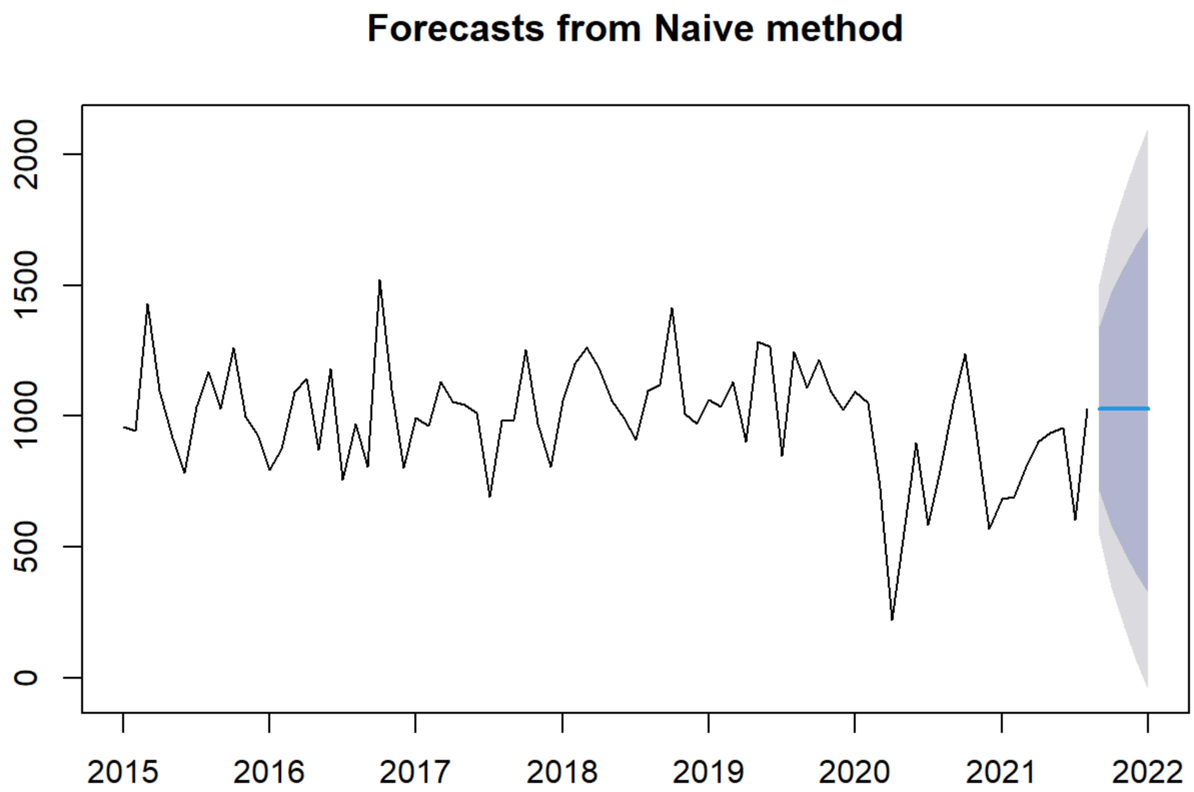
Accuracy of Forecast

```
accuracy(FDfore3)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -5.312275 128.044 97.37976 -3.792075 12.44775 0.5276185 0.2059805
```

Naive Forecast, 5 periods

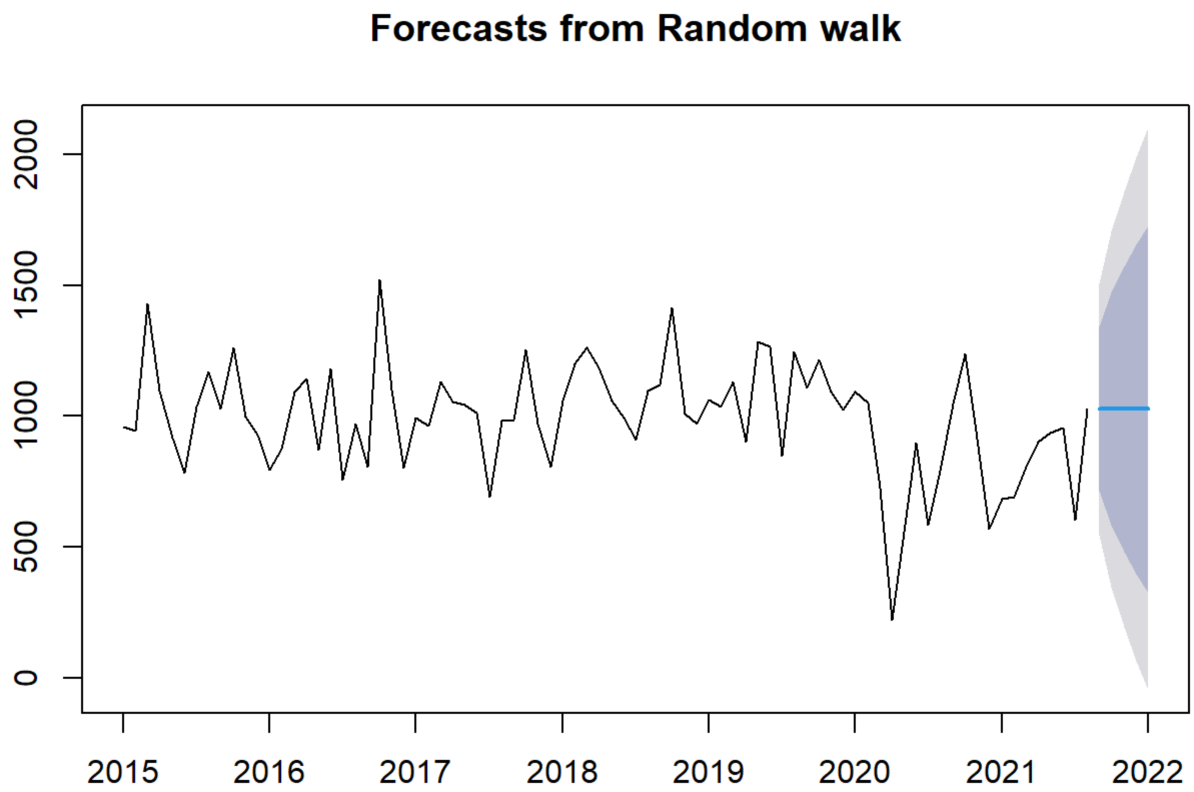
```
naive_forecast <- naive(FDts,5)
plot(naive_forecast)
```



##

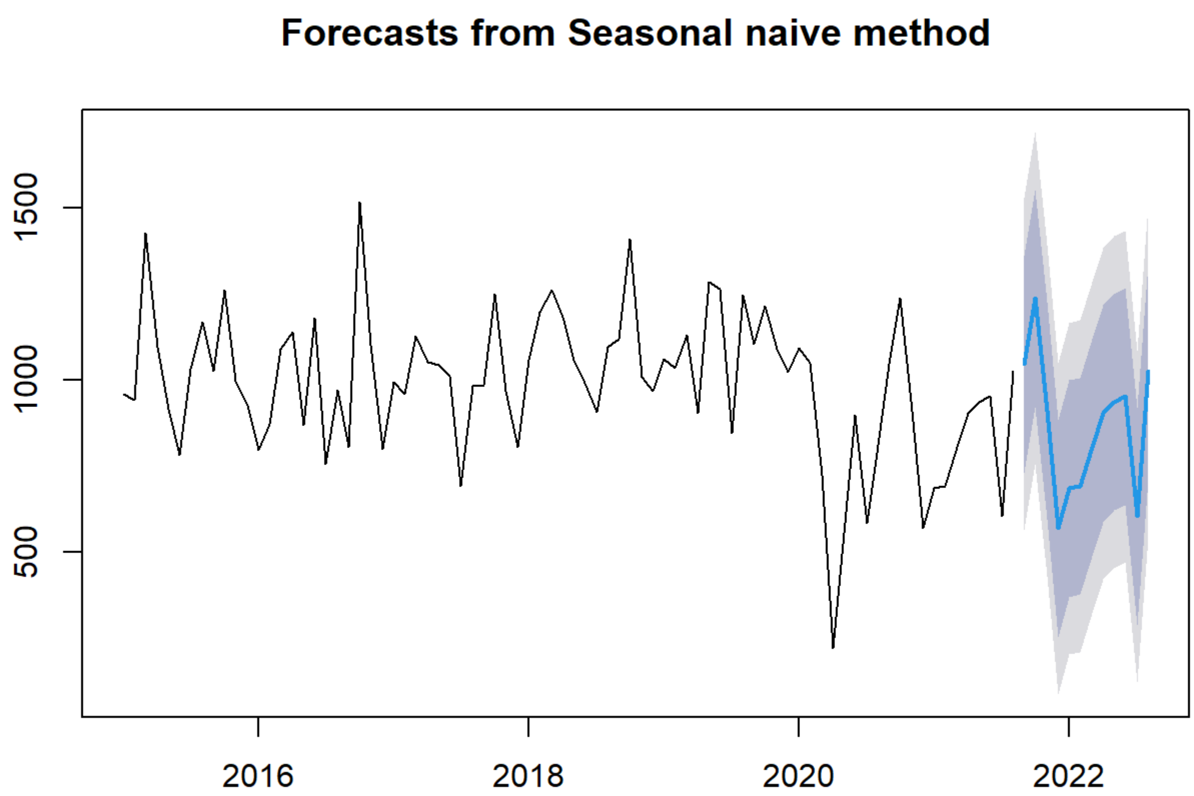
lets try Random Walk

```
rwf_forecast <- rwf(FDts,5)
plot(rwf_forecast)
```



Seasonal Naive Forecast

```
snaive_forecast <- snaive(FDts,12)
plot(snaive_forecast)
```

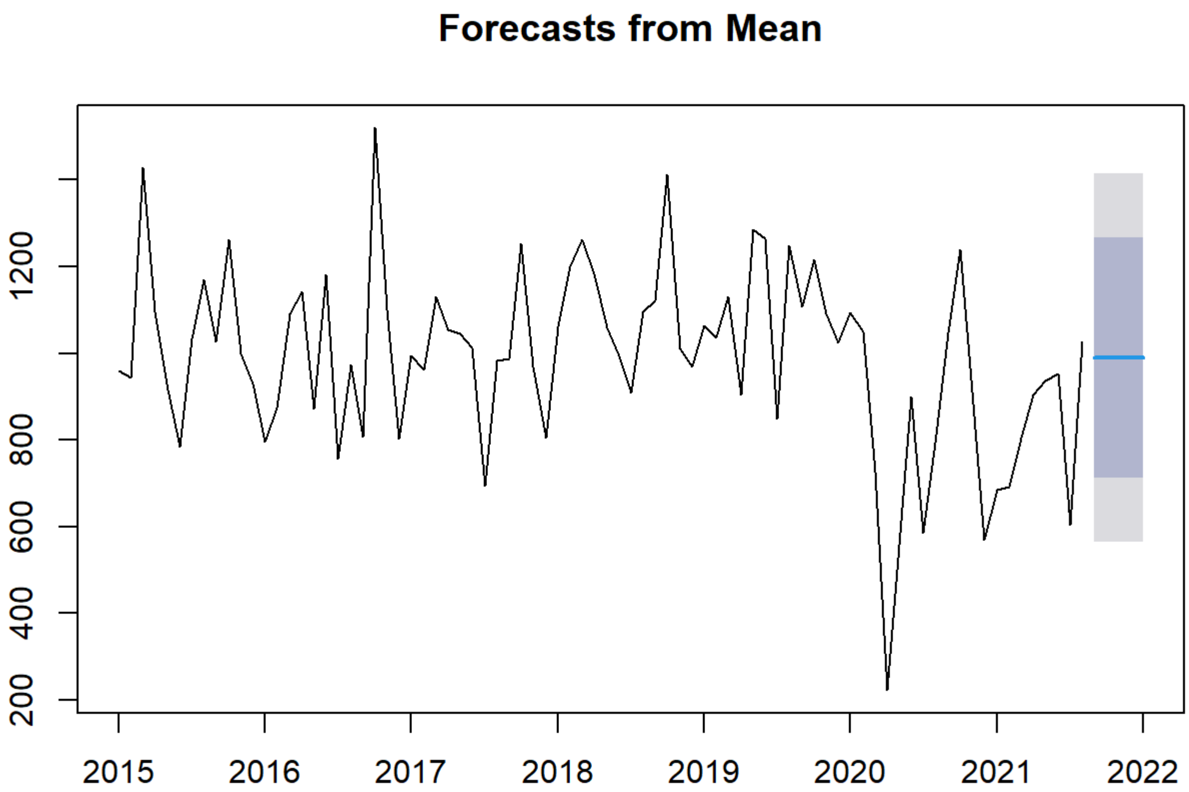



```
accuracy(snaive_forecast)
```

```
##                     ME    RMSE      MAE       MPE     MAPE MASE      ACF1
## Training set -31.93529 245.841 184.5647 -10.01052 24.29849    1 0.4868142
```

#take Mean of all available history

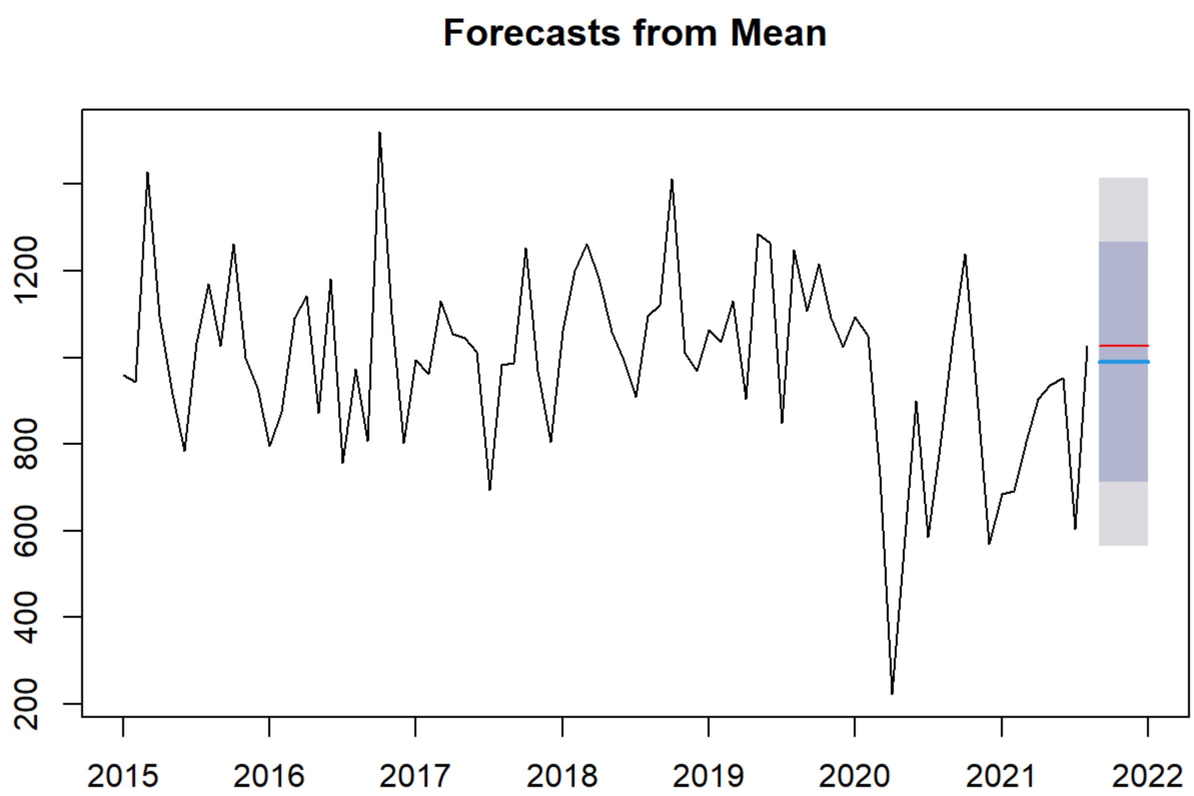
```
mean_forecast <- meanf(FDts,5) # 5 is the forecasting period (5 quarters out)
plot(mean_forecast)
```




```
accuracy(mean_forecast)
```

```
##                         ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -2.559355e-14 211.2258 157.7731 -7.567326 20.47238 0.854839
##                   ACF1
## Training set 0.3361594
```

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



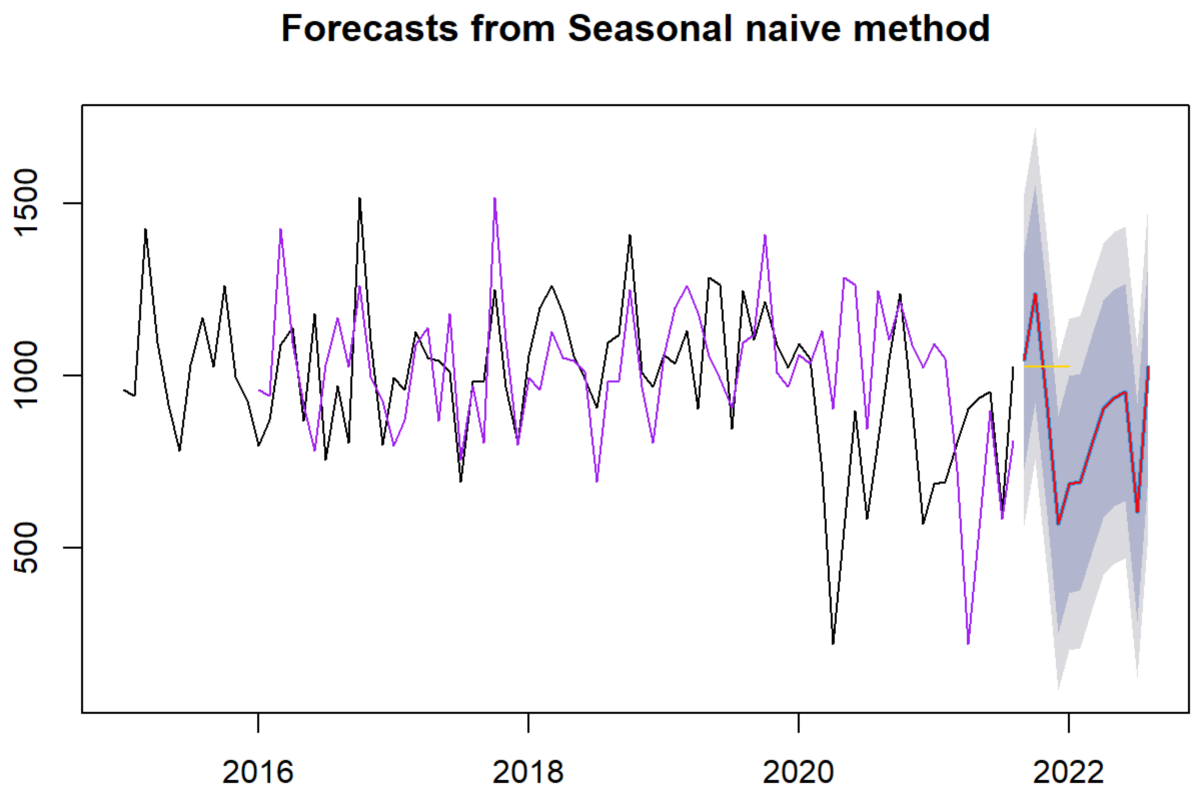

```
attributes(naive_forecast)
```

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

```
attributes(rwf_forecast)
```

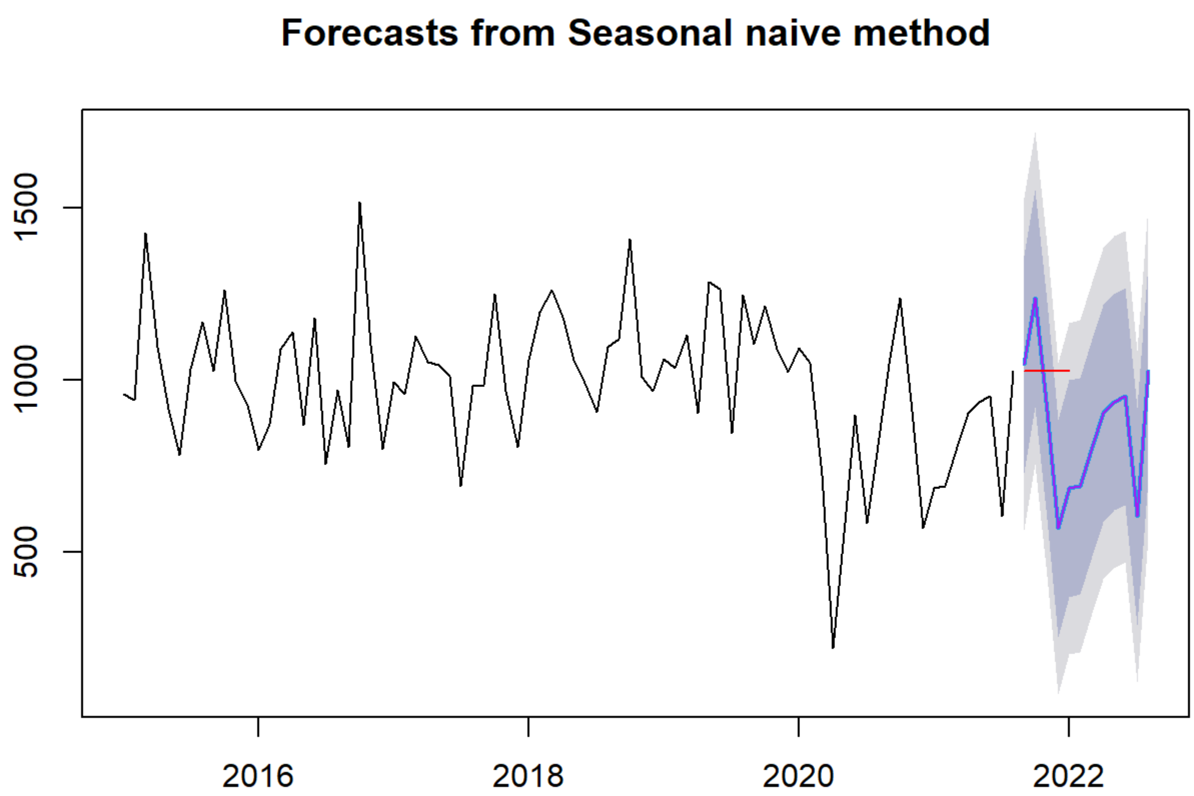
```
## $names
##  [1] "method"    "model"     "lambda"    "x"         "fitted"    "residuals"
##  [7] "series"    "mean"      "level"     "lower"     "upper"
##
## $class
## [1] "forecast"
```

```
plot(snaive_forecast)
lines(rwf_forecast$mean,col="green")
lines(snaive_forecast$mean,col="Red")
lines(snaive_forecast$fitted, col = "Purple")
lines(naive_forecast$mean,col="Gold")
```



ERROR

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

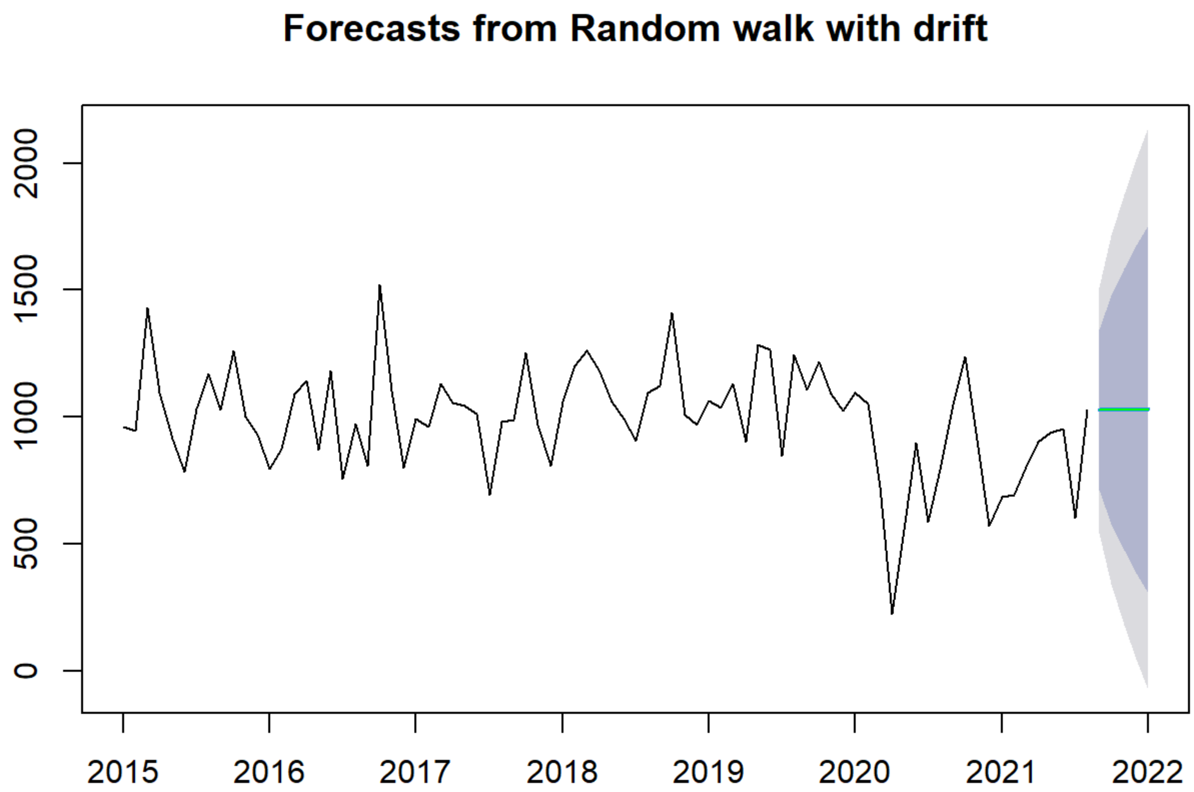


#

ERROR

```
## Show the forecasts overlayed over the actual data
### lines(snaive_forecast$fitted,col="yellow")
```

```
rwf_forecast <- rwf(FDts,5, drift=TRUE)
plot(rwf_forecast)
lines(rwf_forecast$mean,col="green")
```



ETS

Function

```
ets(FDts)
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = FDts)
##
##   Smoothing parameters:
##     alpha = 0.3048
##     gamma = 1e-04
##
##   Initial states:
##     l = 1030.9511
##     s = -126.2111 31.6933 331.7719 20.3817 41.0718 -196.1988
##            30.1869 -39.8469 -82.8874 64.5527 -18.2096 -56.3045
##
##   sigma:  172.6314
##
##      AIC     AICc      BIC
## 1189.358 1196.858 1225.088
```

```
ets_forecast = ets(FDts)
```

```
hw_forecast_level = HoltWinters(FDts, gamma=FALSE)
hw_forecast_level
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = FDts, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.1999853
##  beta : 0.032884
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 828.116898
## b  -4.827966
```

```
hw_forecast_level2 = HoltWinters(FDts)
hw_forecast_level2
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = FDts)
##
## Smoothing parameters:
##  alpha: 0.4796243
##  beta : 0
##  gamma: 0.611468
##
## Coefficients:
##            [,1]
## a    877.336068
## b     -3.477558
## s1   120.492986
## s2   312.540878
## s3    26.838074
## s4  -158.036853
## s5   -49.467372
## s6   -58.103662
## s7   -39.897434
## s8  -127.083638
## s9     6.004482
## s10  101.068646
## s11 -207.145025
## s12  128.740038
```

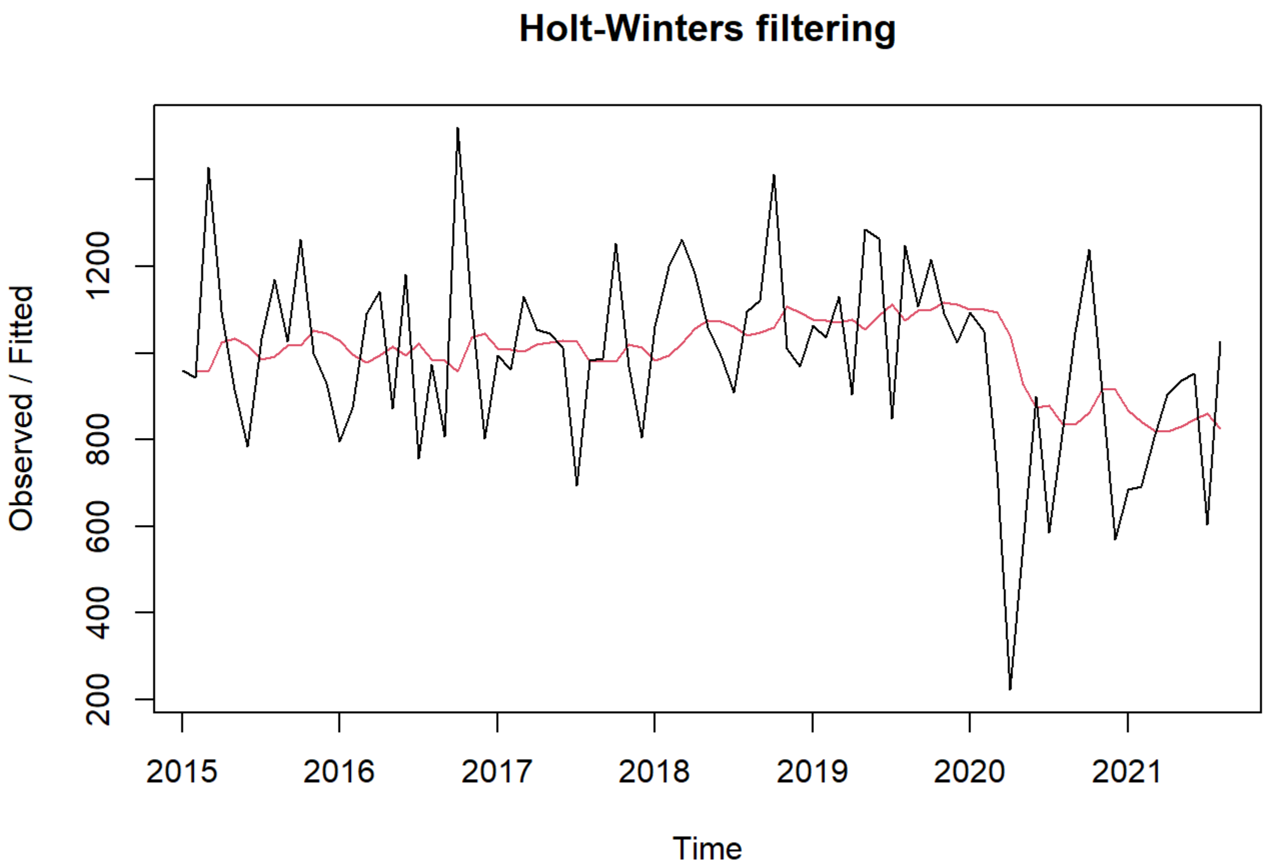
```
HoltWinters(FDts,beta=FALSE,gamma=FALSE)
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = FDts, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.1397632
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 853.3879
```

```
SSE_Simple <- HoltWinters(FDts,beta=FALSE,gamma=FALSE)
attributes(SSE_Simple)
```

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

```
plot(SSE_Simple)
```




```
SSE_Simple$SSE
```

```
## [1] 3377341
```

```
SSE_Simple$fitted
```

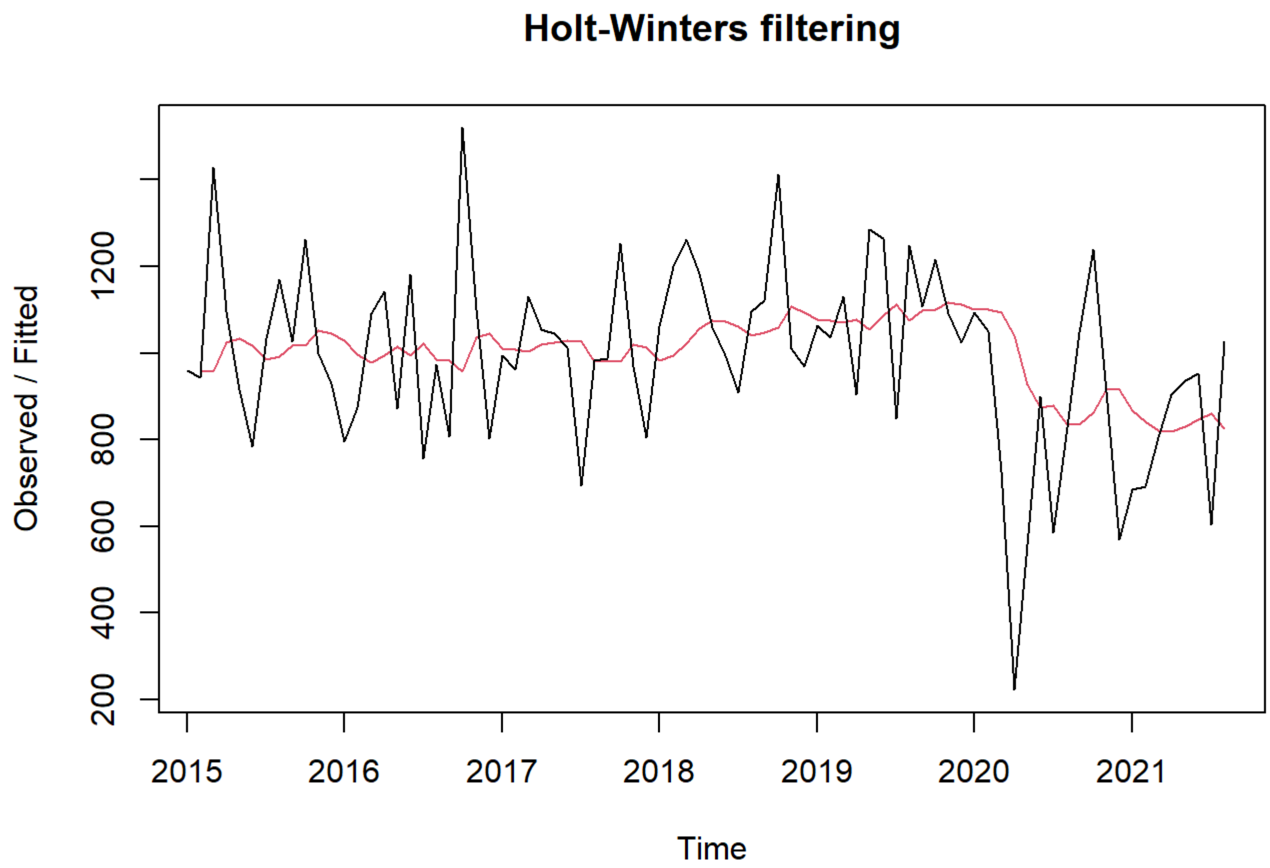
```
##               xhat     level
## Feb 2015  960.4200  960.4200
## Mar 2015  958.1363  958.1363
## Apr 2015 1023.9625 1023.9625
## May 2015 1034.1704 1034.1704
## Jun 2015 1017.6336 1017.6336
## Jul 2015  984.9033  984.9033
## Aug 2015  991.8379  991.8379
## Sep 2015 1016.6685 1016.6685
## Oct 2015 1018.1237 1018.1237
## Nov 2015 1052.2533 1052.2533
## Dec 2015 1044.8454 1044.8454
## Jan 2016 1028.7132 1028.7132
## Feb 2016  996.2472  996.2472
## Mar 2016  979.2384  979.2384
## Apr 2016  994.9354  994.9354
## May 2016 1015.4673 1015.4673
## Jun 2016  995.3264  995.3264
## Jul 2016 1021.3061 1021.3061
## Aug 2016  984.4483  984.4483
## Sep 2016  982.8105  982.8105
## Oct 2016  958.2414  958.2414
## Nov 2016 1036.7434 1036.7434
## Dec 2016 1045.8178 1045.8178
## Jan 2017 1011.7173 1011.7173
## Feb 2017 1009.3934 1009.3934
## Mar 2017 1002.7695 1002.7695
## Apr 2017 1020.5852 1020.5852
## May 2017 1025.3546 1025.3546
## Jun 2017 1028.0933 1028.0933
## Jul 2017 1026.0859 1026.0859
## Aug 2017  979.5788  979.5788
## Sep 2017  980.0919  980.0919
## Oct 2017  981.1469  981.1469
## Nov 2017 1019.0986 1019.0986
## Dec 2017 1012.1400 1012.1400
## Jan 2018  983.3432  983.3432
## Feb 2018  994.1367  994.1367
## Mar 2018 1022.9437 1022.9437
## Apr 2018 1056.3899 1056.3899
## May 2018 1074.2880 1074.2880
## Jun 2018 1072.2799 1072.2799
## Jul 2018 1061.2763 1061.2763
## Aug 2018 1039.9057 1039.9057
## Sep 2018 1047.8756 1047.8756
## Oct 2018 1058.2005 1058.2005
## Nov 2018 1107.7143 1107.7143
## Dec 2018 1094.0924 1094.0924
## Jan 2019 1076.7656 1076.7656
## Feb 2019 1074.8599 1074.8599
## Mar 2019 1069.5615 1069.5615
## Apr 2019 1078.1301 1078.1301
## May 2019 1053.7890 1053.7890
```

```
## Jun 2019 1086.0968 1086.0968
## Jul 2019 1111.1791 1111.1791
## Aug 2019 1074.4858 1074.4858
## Sep 2019 1098.6529 1098.6529
## Oct 2019 1099.7971 1099.7971
## Nov 2019 1116.1889 1116.1889
## Dec 2019 1112.7859 1112.7859
## Jan 2020 1100.4705 1100.4705
## Feb 2020 1099.6528 1099.6528
## Mar 2020 1092.8502 1092.8502
## Apr 2020 1041.6046 1041.6046
## May 2020  927.1658  927.1658
## Jun 2020  875.4191  875.4191
## Jul 2020  878.7148  878.7148
## Aug 2020  837.7454  837.7454
## Sep 2020  834.1108  834.1108
## Oct 2020  863.9222  863.9222
## Nov 2020  916.3806  916.3806
## Dec 2020  915.7585  915.7585
## Jan 2021  867.3993  867.3993
## Feb 2021  842.0338  842.0338
## Mar 2021  821.1876  821.1876
## Apr 2021  818.9838  818.9838
## May 2021  830.8660  830.8660
## Jun 2021  845.7863  845.7863
## Jul 2021  860.9105  860.9105
## Aug 2021  825.1437  825.1437
```

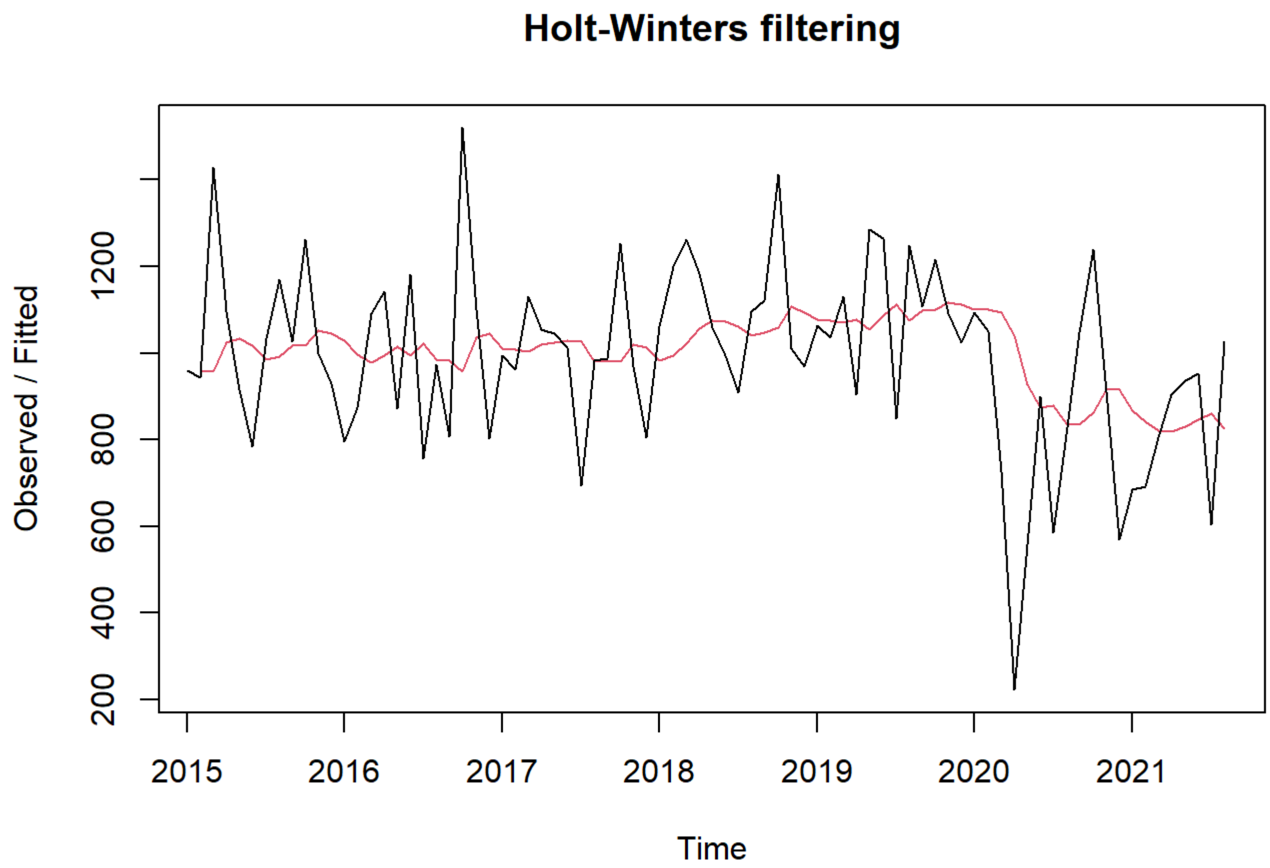
```
SSE_Simple <- HoltWinters(FDts,beta=FALSE,gamma=FALSE)
attributes(SSE_Simple)
```

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

```
plot(SSE_Simple)
```




```
plot(SSE_Simple)
```




```
SSE_Simple$SSE
```

```
## [1] 3377341
```

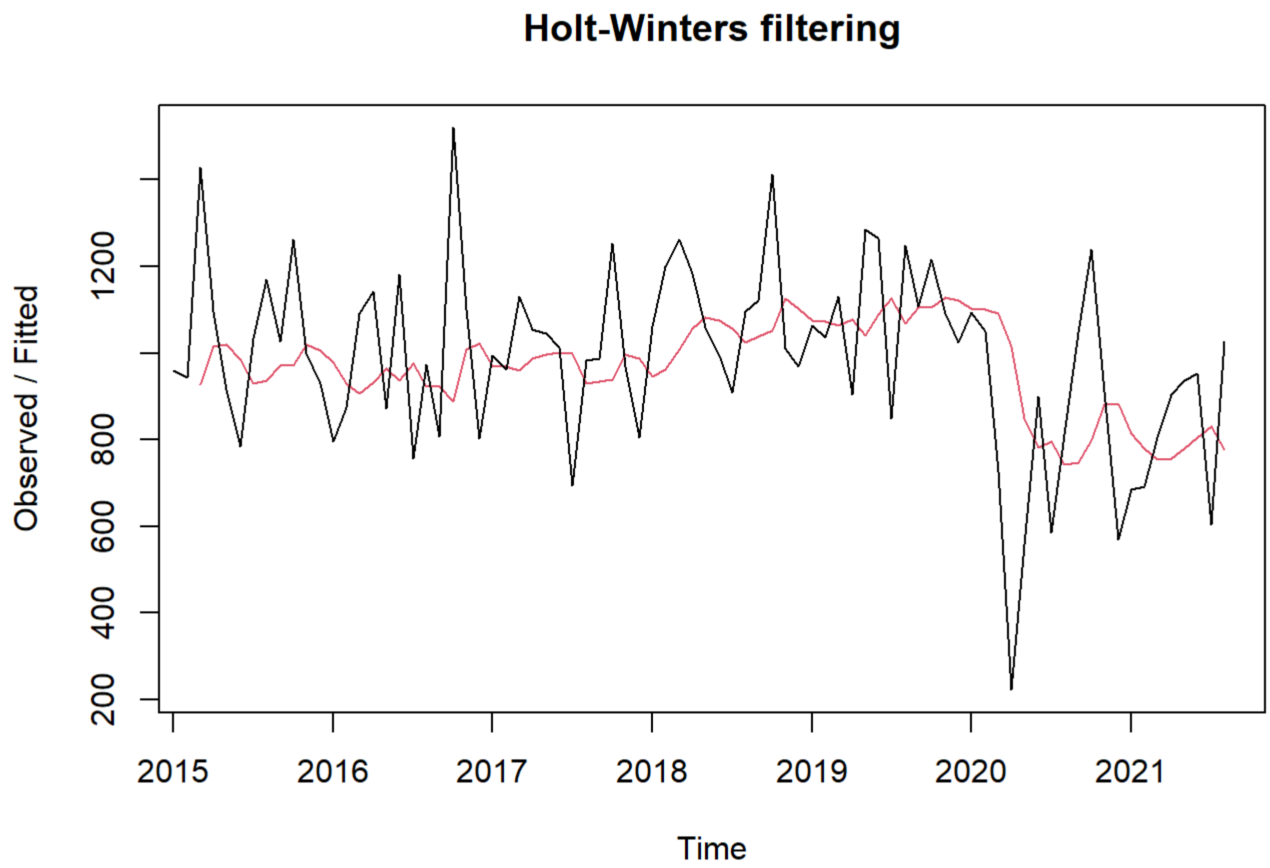
```
SSE_Simple$fitted
```

```
##                xhat     level
## Feb 2015  960.4200  960.4200
## Mar 2015  958.1363  958.1363
## Apr 2015 1023.9625 1023.9625
## May 2015 1034.1704 1034.1704
## Jun 2015 1017.6336 1017.6336
## Jul 2015  984.9033  984.9033
## Aug 2015  991.8379  991.8379
## Sep 2015 1016.6685 1016.6685
## Oct 2015 1018.1237 1018.1237
## Nov 2015 1052.2533 1052.2533
## Dec 2015 1044.8454 1044.8454
## Jan 2016 1028.7132 1028.7132
## Feb 2016  996.2472  996.2472
## Mar 2016  979.2384  979.2384
## Apr 2016  994.9354  994.9354
## May 2016 1015.4673 1015.4673
## Jun 2016  995.3264  995.3264
## Jul 2016 1021.3061 1021.3061
## Aug 2016  984.4483  984.4483
## Sep 2016  982.8105  982.8105
## Oct 2016  958.2414  958.2414
## Nov 2016 1036.7434 1036.7434
## Dec 2016 1045.8178 1045.8178
## Jan 2017 1011.7173 1011.7173
## Feb 2017 1009.3934 1009.3934
## Mar 2017 1002.7695 1002.7695
## Apr 2017 1020.5852 1020.5852
## May 2017 1025.3546 1025.3546
## Jun 2017 1028.0933 1028.0933
## Jul 2017 1026.0859 1026.0859
## Aug 2017  979.5788  979.5788
## Sep 2017  980.0919  980.0919
## Oct 2017  981.1469  981.1469
## Nov 2017 1019.0986 1019.0986
## Dec 2017 1012.1400 1012.1400
## Jan 2018  983.3432  983.3432
## Feb 2018  994.1367  994.1367
## Mar 2018 1022.9437 1022.9437
## Apr 2018 1056.3899 1056.3899
## May 2018 1074.2880 1074.2880
## Jun 2018 1072.2799 1072.2799
## Jul 2018 1061.2763 1061.2763
## Aug 2018 1039.9057 1039.9057
## Sep 2018 1047.8756 1047.8756
## Oct 2018 1058.2005 1058.2005
## Nov 2018 1107.7143 1107.7143
## Dec 2018 1094.0924 1094.0924
## Jan 2019 1076.7656 1076.7656
## Feb 2019 1074.8599 1074.8599
## Mar 2019 1069.5615 1069.5615
## Apr 2019 1078.1301 1078.1301
## May 2019 1053.7890 1053.7890
```

```
## Jun 2019 1086.0968 1086.0968
## Jul 2019 1111.1791 1111.1791
## Aug 2019 1074.4858 1074.4858
## Sep 2019 1098.6529 1098.6529
## Oct 2019 1099.7971 1099.7971
## Nov 2019 1116.1889 1116.1889
## Dec 2019 1112.7859 1112.7859
## Jan 2020 1100.4705 1100.4705
## Feb 2020 1099.6528 1099.6528
## Mar 2020 1092.8502 1092.8502
## Apr 2020 1041.6046 1041.6046
## May 2020  927.1658  927.1658
## Jun 2020  875.4191  875.4191
## Jul 2020  878.7148  878.7148
## Aug 2020  837.7454  837.7454
## Sep 2020  834.1108  834.1108
## Oct 2020  863.9222  863.9222
## Nov 2020  916.3806  916.3806
## Dec 2020  915.7585  915.7585
## Jan 2021  867.3993  867.3993
## Feb 2021  842.0338  842.0338
## Mar 2021  821.1876  821.1876
## Apr 2021  818.9838  818.9838
## May 2021  830.8660  830.8660
## Jun 2021  845.7863  845.7863
## Jul 2021  860.9105  860.9105
## Aug 2021  825.1437  825.1437
```

SSE With Trend

```
SSE_Trend <- HoltWinters(FDts,gamma=FALSE)
plot(SSE_Trend)
```




```
SSE_Trend
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = FDts, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.1999853
##  beta : 0.032884
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 828.116898
## b  -4.827966
```

```
attributes(SSE_Trend)
```

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

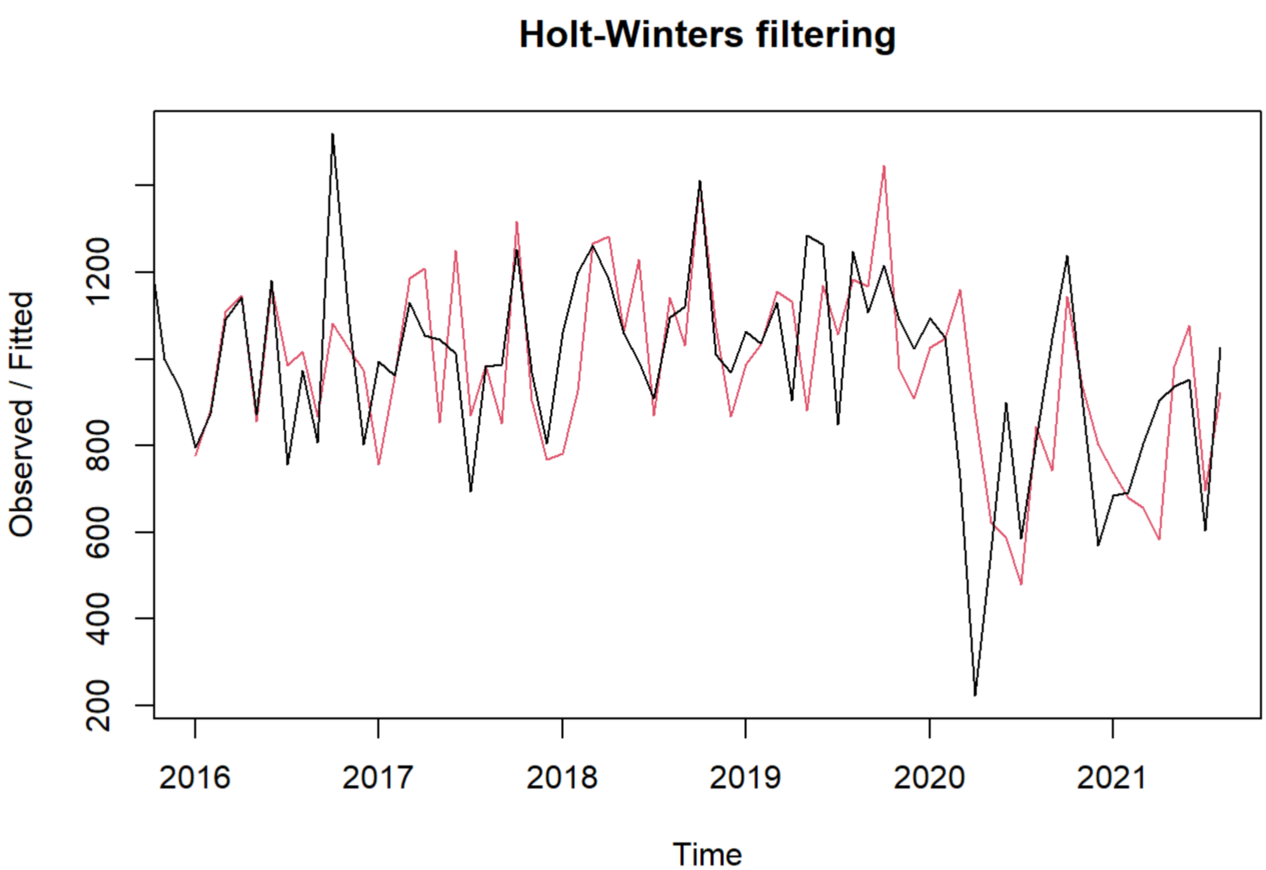
```
SSE_Trend$SSE # check the residual error magnitude
```

```
## [1] 3552399
```

```
# now Holts Winters
SSE_Winters <- HoltWinters(FDts)
SSE_Winters$SSE
```

```
## [1] 2182035
```

```
plot(SSE_Winters)
```




```
# lets play with ETS
ets(FDts)
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = FDts)
##
##   Smoothing parameters:
##     alpha = 0.3048
##     gamma = 1e-04
##
##   Initial states:
##     l = 1030.9511
##     s = -126.2111 31.6933 331.7719 20.3817 41.0718 -196.1988
##            30.1869 -39.8469 -82.8874 64.5527 -18.2096 -56.3045
##
##   sigma:  172.6314
##
##      AIC     AICc      BIC
## 1189.358 1196.858 1225.088
```

```
ets_forecast <- ets(FDts)
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"
```

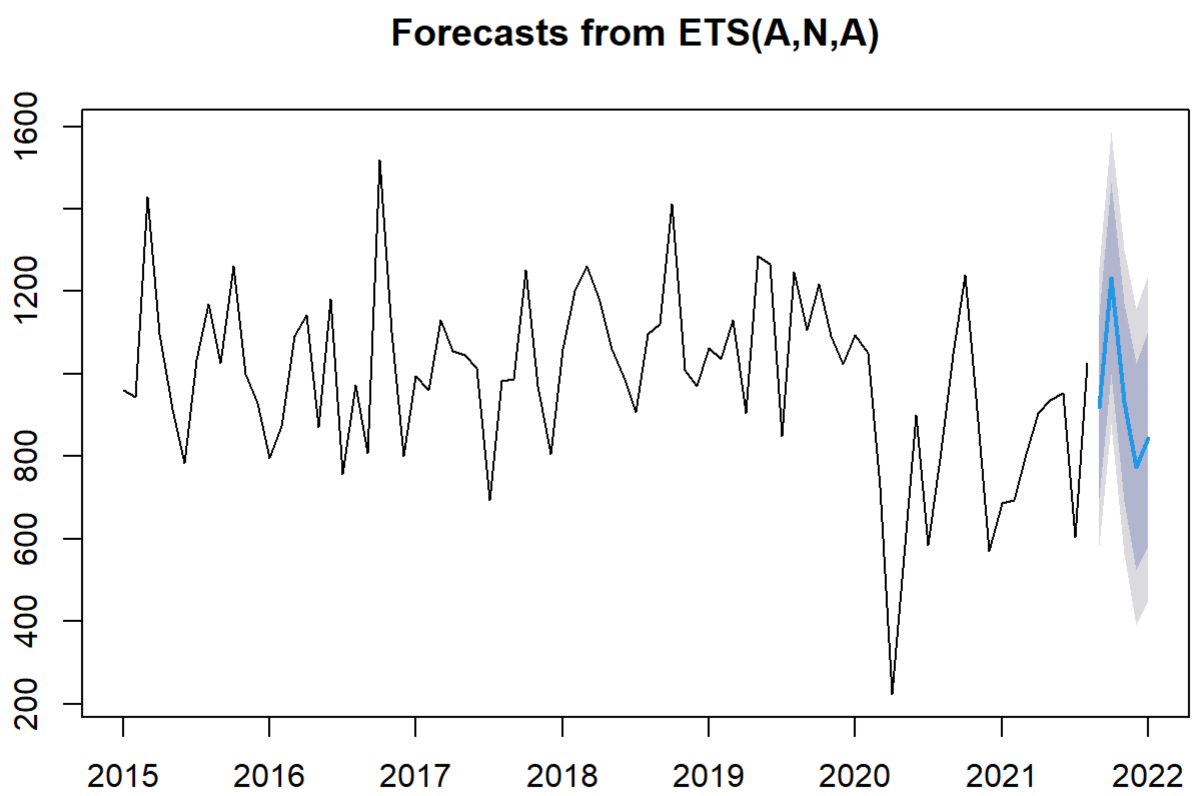
```
ets_forecast$mse
```

```
## [1] 24586.32
```

```
# how do we forecast now!!
forecast.ets(ets_forecast, h=5)
```

```
##          Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
## Sep 2021       920.6504  699.4144 1141.886 582.2991 1259.002
## Oct 2021      1232.0357 1000.7494 1463.322 878.3138 1585.758
## Nov 2021       931.9591  691.0414 1172.877 563.5073 1300.411
## Dec 2021       774.0541  523.8755 1024.233 391.4390 1156.669
## Jan 2022       843.9844  584.8757 1103.093 447.7119 1240.257
```

```
forecast_ets <- forecast.ets(ets_forecast, h=5)
plot(forecast_ets)
```



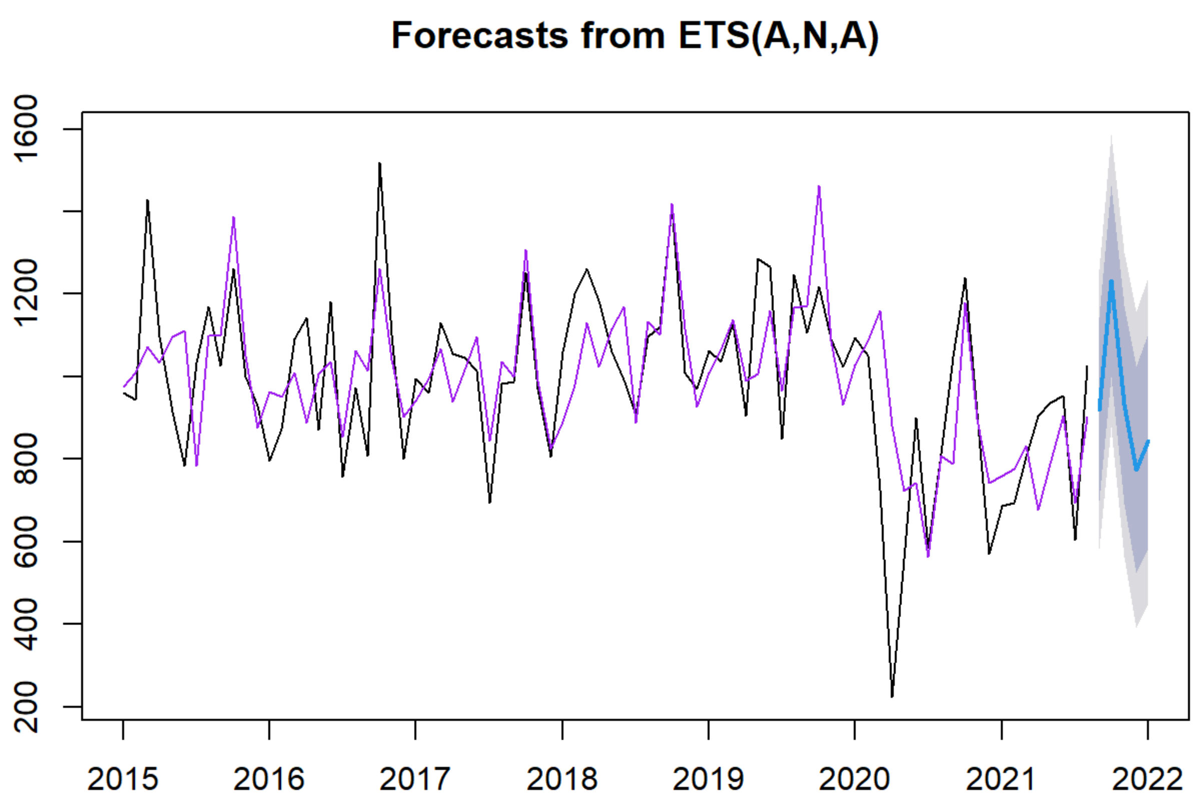
ERROR

```
### SAts = seasadj(FDts)
```

```
# forecast with ets
forecast.ets(ets_forecast, h=5)
```

```
##          Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
## Sep 2021       920.6504  699.4144 1141.886 582.2991 1259.002
## Oct 2021      1232.0357 1000.7494 1463.322 878.3138 1585.758
## Nov 2021       931.9591  691.0414 1172.877 563.5073 1300.411
## Dec 2021       774.0541  523.8755 1024.233 391.4390 1156.669
## Jan 2022       843.9844  584.8757 1103.093 447.7119 1240.257
```

```
forecast_ets <- forecast.ets(ets_forecast, h=5)
plot(forecast_ets)
lines(forecast_ets$fitted, col = "Purple")
```




```
accuracy(forecast_ets)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE     ACF1
## Training set -5.358499 156.8003 114.5659 -5.046979 14.81766 0.6207359 0.235456
```