Enrollment Projection Using Time Series

Shu Zhang

January 26, 2018

This project is to predict the enrollment of several types of students at Iowa State University in 2018.

Load required libraries

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.4.1
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##     date
```

```r
library(bsts)
```

```
## Warning: package 'bsts' was built under R version 3.4.1
## Loading required package: BoomSpikeSlab
## Warning: package 'BoomSpikeSlab' was built under R version 3.4.1
## Loading required package: Boom
## Warning: package 'Boom' was built under R version 3.4.1
## Loading required package: MASS
##
## Attaching package: 'Boom'
## The following object is masked from 'package:stats':
##
##     rWishart
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 3.4.1
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: xts
## Warning: package 'xts' was built under R version 3.4.2
```

```r
library(plyr)
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:lubridate':
##
##     here
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.1
## Warning: Installed Rcpp (0.12.12) different from Rcpp used to build dplyr (0.12.11).
## Please reinstall dplyr to avoid random crashes or undefined behavior.
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following objects are masked from 'package:xts':
##
##     first, last
## The following object is masked from 'package:MASS':
##
##     select
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.4.3
```

```r
library(prophet)
```

```
## Warning: package 'prophet' was built under R version 3.4.1
## Loading required package: Rcpp
## Warning: package 'Rcpp' was built under R version 3.4.1
```

```r
library(nnfor)
```

```
## Warning: package 'nnfor' was built under R version 3.4.3
## Loading required package: forecast
## Warning: package 'forecast' was built under R version 3.4.3
```

```r
library(magrittr)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.4.1
```

```r
library(RColorBrewer)
library(forecast)
```

group name abbreviations explained

```r
Groups <- c('FRF','FNRF','FFF','FRT','FNRT','FFT','SRF','SNRF','SFF','SRT','SNRT','SFT')
Explaination <- c('fall resident freshman','fall non resident','fall foreign freshman','fall resident t
kable(cbind(Groups,Explaination),caption = "12 groups to be modeled.")
```

Table 1: 12 groups to be modeled.

Groups	Explaination
FRF	fall resident freshman
FNRF	fall non resident
FFF	fall foreign freshman
FRT	fall resident transfer
FNRT	fall non resident transfer
FFT	fall foreign transfer
SRF	summer resident freshman
SNRF	summer non resident
SFF	summer foreign freshman
SRT	summer resident transfer
SNRT	summer non resident transfer
SFT	summer foreign transfer
##read th	e weekly data and print the dataset dimension

```r
week_full <- read.csv('D:/Work/Weeks_Data_010118_new.csv')
dim(week_full)
```

```
## [1] 4075   39
```

print the first 3 rows of the dataframe

```r
head(week_full,3)
```

```
##   Cycle     Date Year FRF_P FNRF_P FFF_P FRF_O FNRF_O FFF_O FRF_A FNRF_A
## 1     9 08/27/03 2004   238    100     9     0      0     0     0      0
## 2    10 08/28/03 2004   246    103     9     0      0     0     0      0
## 3    10 08/29/03 2004   246    103     9     0      0     0     0      0
##   FFF_A FRT_P FNRT_P FFT_P FRT_O FNRT_O FFT_O FRT_A FNRT_A FFT_A SRF_P
## 1     0    71     24     3     0      0     0     0      0     0     1
## 2     0    71     24     3     0      0     0     0      0     0     2
## 3     0    71     24     3     0      0     0     0      0     0     2
##   SNRF_P SFF_P SRF_O SNRF_O SFF_O SRF_A SNRF_A SFF_A SRT_P SNRT_P SFT_P
## 1      1     0     0      0     0     0      0     0     1      2     2
## 2      1     0     0      0     0     0      0     0     1      2     2
## 3      1     0     0      0     0     0      0     0     1      2     2
##   SRT_O SNRT_O SFT_O SRT_A SNRT_A SFT_A
## 1     0      0     0     0      0     0
## 2     0      0     0     0      0     0
```

```
## 3      0       0      0      0       0      0
```

print the last 3 rows of the dataframe

```r
tail(week_full,3)
```

```
##      Cycle     Date Year FRF_P FNRF_P FFF_P FRF_O FNRF_O FFF_O FRF_A
## 4073    27 12/28/17 2018  5079  10349   403  4655   9415   241  2476
## 4074    27 12/29/17 2018  5087  10373   412  4669   9443   256  2483
## 4075    28 01/01/18 2018  5097  10401   423  4674   9459   256  2520
##      FNRF_A FFF_A FRT_P FNRT_P FFT_P FRT_O FNRT_O FFT_O FRT_A FNRT_A FFT_A
## 4073   1528    14   720    353    24   478    200     7   216     61     1
## 4074   1543    14   728    355    26   478    200     7   221     61     1
## 4075   1569    15   739    355    29   478    200     7   227     61     1
##      SRF_P SNRF_P SFF_P SRF_O SNRF_O SFF_O SRF_A SNRF_A SFF_A SRT_P SNRT_P
## 4073    27     58    12    20     43     9     8     14     0    34     25
## 4074    27     58    12    20     43     9     8     14     0    33     26
## 4075    27     59    12    20     43     9     8     15     0    33     26
##      SFT_P SRT_O SNRT_O SFT_O SRT_A SNRT_A SFT_A
## 4073    12    22     20     5    17      3     1
## 4074    12    22     20     5    17      3     2
## 4075    12    22     20     5    17      3     2
```

remove partial month from both ends

```r
week_full <- week_full[-c(1:3,length(week_full$Date)),]
```

convert the column 'Date' to Date type

```r
week_full$Date <-as.Date(week_full$Date,format = "%m/%d/%y")
```

add one row for Aug 31st, 2013, since 2013 have till Aug 30th

```r
which(week_full$Date=='2013-08-30')#2734 2790
```

```
## [1] 2734 2790
```

```r
week_full <- rbind( week_full[1:2734,], week_full[2734,], week_full[2735:dim(week_full)[1],])
week_full[2735,'Date'] <- as.Date('08/31/13',format = "%m/%d/%y")
```

add one row for Aug 31st,2014, since it has only has records till Aug 1st, no sept.

```r
which(week_full$Date=='2014-08-01')#3026 3060
```

```
## [1] 3026 3060
```

```r
week_full <- rbind( week_full[1:3026,], week_full[3026,], week_full[3027:dim(week_full)[1],])
week_full[3027,'Date'] <- as.Date('08/31/14',format = "%m/%d/%y")
```

truncate each cycle by 08-31

```
week_full <- week_full[!((month(week_full$Date)==9)&(week_full$FRF_A>1500)),]
```

find start overlap dates

```
Dates <- week_full$Date
diff(Dates)[diff(Dates)<0]
```

```
## Time differences in days
##  [1]   -12   -62 -4745   -51   -59   -62   -72   -53    -1   -66   -75
## [12]   -70   -77   -77   -73
```

```
which(diff(Dates)<0)
```

```
##  [1]  252  759  892 1032 1273 1579 1879 2157 2327 2421 2716 3008 3321 3606
## [15] 3893
```

correct the mistyped dates in vector Dates

```
Dates[892] <- as.Date("2007-02-16")
Dates[2326]<- as.Date("2012-04-14")
Dates[2327]<- as.Date("2012-04-16")
Dates[2328]<- as.Date("2012-04-17") # data quality, mistyped dates
diff(Dates)[diff(Dates)<0]#-15 -66 -51 -72 -66 -72 -57 -69 -75 -70 -85 -84 -94
```

```
## Time differences in days
##  [1] -12 -62 -51 -59 -62 -72 -53 -66 -75 -70 -77 -77 -73
```

correct the mityped dates in dataset

```
week_full$Date[892] <- as.Date("2007-02-16")
week_full$Date[2326]<- as.Date("2012-04-14")
week_full$Date[2327]<- as.Date("2012-04-16")
week_full$Date[2328]<- as.Date("2012-04-17")
```

find indexes for the overlapping dates between two consequtive cycles

```
former_end_dates <- Dates[which(diff(Dates)<0)]
start_Overlap_dates <- Dates[(which(diff(Dates)<0)+1)]
start_Overlap_dates_index <- (which(diff(Dates)<0)+1)
```

find end overlap dates

```
end_overlap_dates_index <- vector()
for(i in 1:length(start_Overlap_dates_index)){
  j <- start_Overlap_dates_index[i]
```

```
  repeat {
    if (Dates[j]>former_end_dates[i]) {
      end_overlap_dates_index <-c(end_overlap_dates_index,j-1)
      break}
    j=j+1;

  }
}
```

remove overlapping dates

```
index_to_remove <- vector()
for(i in 1:length(start_Overlap_dates_index)){
  index_to_remove <- c(index_to_remove,seq(start_Overlap_dates_index[i],end_overlap_dates_index[i]))
}
dim(week_full)#4033   39
```

```
## [1] 4033   39
```

```
week_rm_overlap <- week_full[-index_to_remove,]
dim(week_rm_overlap)#3510   39
```

```
## [1] 3510   39
```

remove variable Cycle

```
week_rm_overlap <- subset(week_rm_overlap,select = -c(Cycle))
week_rm_overlap$Year <- as.factor(week_rm_overlap$Year)
```

aggregate weekly data to monthly by keeping the last record in each month

```
last <- function(x) { return( x[length(x)] ) }
week_to_month<- week_rm_overlap %>% group_by(month=floor_date(Date, "month")) %>%summarise_all(funs(las
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.1
```

```
week_to_month <- as.data.frame(week_to_month)
monthly_data <- week_to_month[,-1]
#write.csv(monthly_data,"month_data.csv",row.names = F)
```

create time series for each group

```
colnames(monthly_data)
```

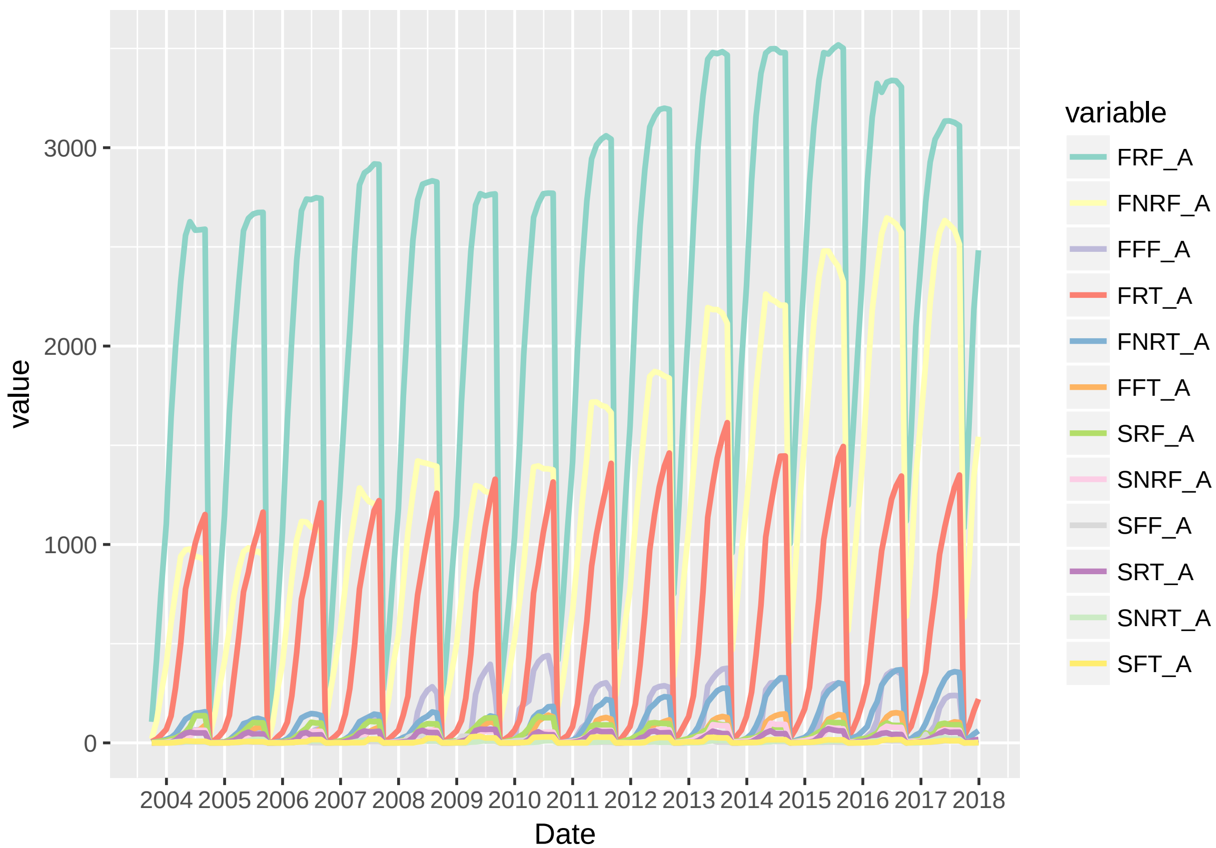
```
##  [1] "Date"   "Year"   "FRF_P"  "FNRF_P" "FFF_P"  "FRF_O"  "FNRF_O"
##  [8] "FFF_O"  "FRF_A"  "FNRF_A" "FFF_A"  "FRT_P"  "FNRT_P" "FFT_P"
## [15] "FRT_O"  "FNRT_O" "FFT_O"  "FRT_A"  "FNRT_A" "FFT_A"  "SRF_P"
## [22] "SNRF_P" "SFF_P"  "SRF_O"  "SNRF_O" "SFF_O"  "SRF_A"  "SNRF_A"
## [29] "SFF_A"  "SRT_P"  "SNRT_P" "SFT_P"  "SRT_O"  "SNRT_O" "SFT_O"
```

```
## [36] "SRT_A"  "SNRT_A" "SFT_A"
```

```
new_data <- monthly_data%>%select(contains("A"))
new_data <- new_data[,-2]
head(new_data,3)
```

```
##         Date FRF_A FNRF_A FFF_A FRT_A FNRT_A FFT_A SRF_A SNRF_A SFF_A
## 1 2003-09-30   106     20     0     6      3     0     1      0     0
## 2 2003-10-31   424    110     1    19      5     0     6      0     0
## 3 2003-11-26   742    239     1    38      9     0     9      1     1
##   SRT_A SNRT_A SFT_A
## 1     0      0     0
## 2     1      1     0
## 3     1      1     0
```

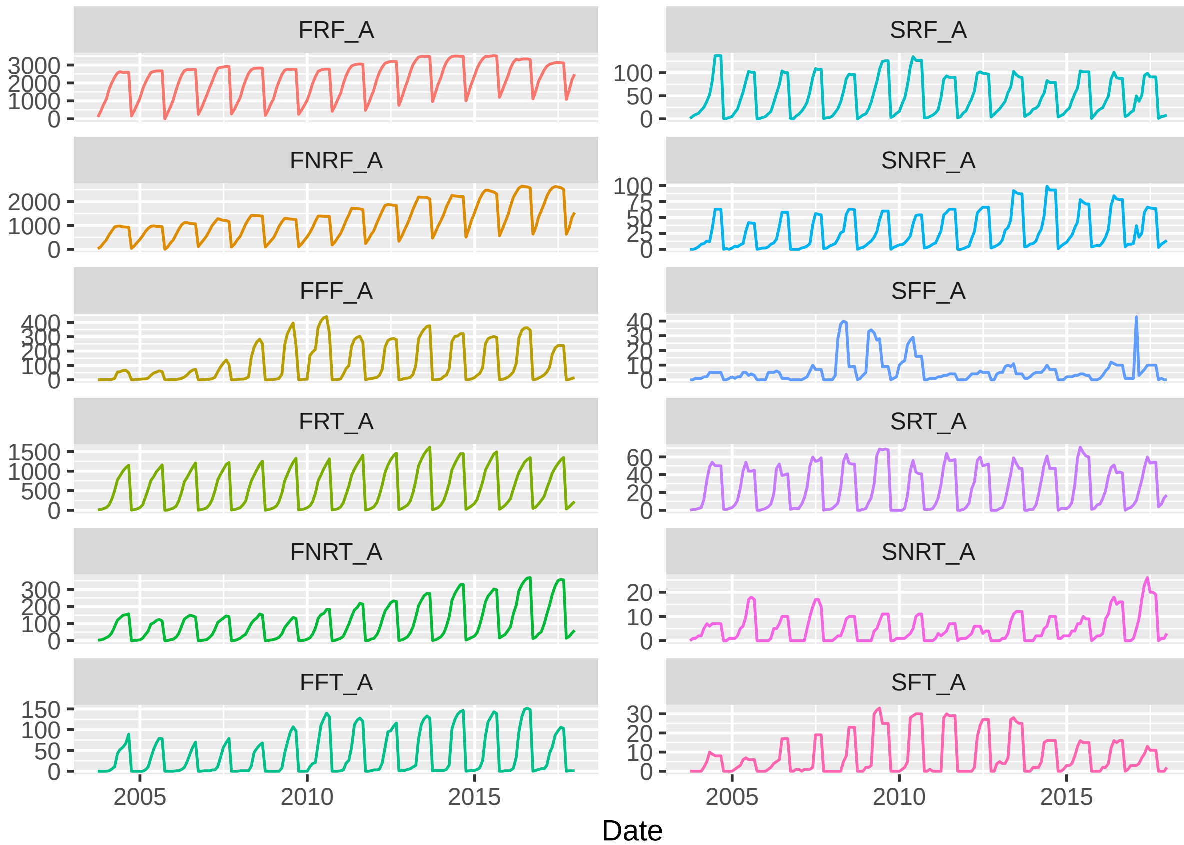
```
# Multiple line plot
meltdf <- melt(new_data,id="Date")
meltdf$Date <- as.Date(meltdf$Date,format='%Y-%m-%d')
ggplot(meltdf,aes(x=Date,y=value,colour=variable,group=variable)) + geom_line(size=1)+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")+ scale_color_brewer(palette="Set3",type="seq
```



graph for each group

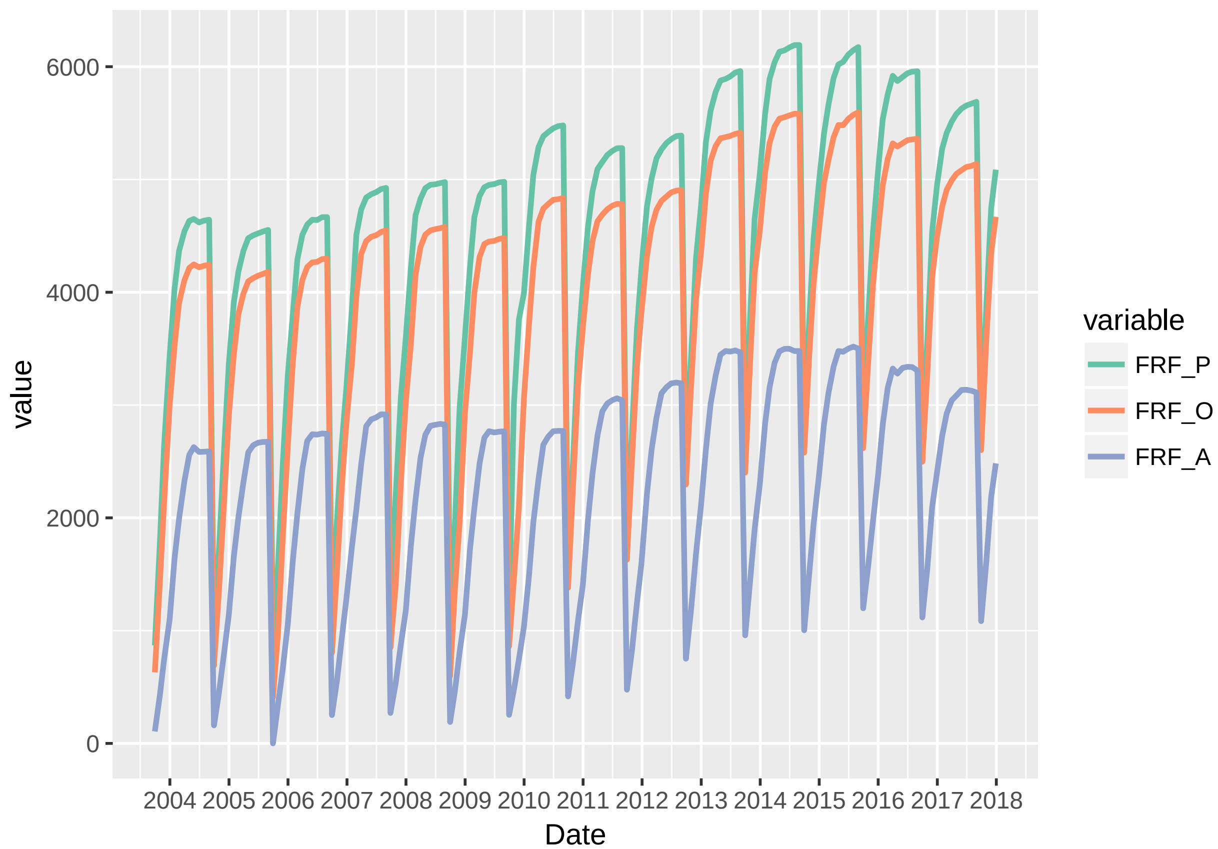
```
ggplot(data=meltdf, aes(x=Date, y=value, col=variable))+
      geom_line( )+
      guides(colour=FALSE)+
```

```
    facet_wrap(~variable, ncol=2,dir="v",scales='free_y')+
    ylab('')
```



##plot FRF

```
Date <- monthly_data$Date

df_FRF <- monthly_data[,c('Date','FRF_P','FRF_O','FRF_A')]

meltdf_FRF <- melt(df_FRF,id="Date")
ggplot(meltdf_FRF,aes(x=Date,y=value,colour=variable,group=variable)) + geom_line(size=1)+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")+ scale_color_brewer(palette="Set2",type="seq
```



#recreate dataset for newly couts each month

```
monthly_data <- read.csv("D:/Work/monthly_data.csv")
new <- monthly_data%>%group_by(Year)%>%select(-Date)%>%lapply(diff)
new <- as.data.frame(new)
new1 <- cbind(Date=monthly_data$Date[2:172],new)

for(row in seq(0, 171, by = 12)[2:15]){
  new1[row,] <- monthly_data[row+1,]
}
dim(new1)
```

```
## [1] 171  38
```

```
head(new1,2)
```

```
##         Date Year FRF_P FNRF_P FFF_P FRF_O FNRF_O FFF_O FRF_A FNRF_A FFF_A
## 1 2003-10-31    0   900   1025    12   791    913     3   318     90     1
## 2 2003-11-26    0   865    902    27   710    778     6   318    129     0
##   FRT_P FNRT_P FFT_P FRT_O FNRT_O FFT_O FRT_A FNRT_A FFT_A SRF_P SNRF_P
## 1   129     48     7    51     11     2    13      2     0     8      5
## 2    89     20    11    60     12     3    19      4     0    17     15
##   SFF_P SRF_O SNRF_O SFF_O SRF_A SNRF_A SFF_A SRT_P SNRT_P SFT_P SRT_O
## 1     1     7      4     1     5      0     0     8      2     2     3
## 2     2    13     11     1     3      1     1    11      2     0     8
##   SNRT_O SFT_O SRT_A SNRT_A SFT_A
## 1      3     2     1      1     0
## 2      0     0     0      0     0
```

```
tail(new1,2)
```

```
##           Date Year FRF_P FNRF_P FFF_P FRF_O FNRF_O FFF_O FRF_A FNRF_A
## 170 2017-11-30    0   877   1942   139   765   1770    89   574    433
## 171 2017-12-29    0   342    923   188   326    765   113   298    200
##     FFF_A FRT_P FNRT_P FFT_P FRT_O FNRT_O FFT_O FRT_A FNRT_A FFT_A SRF_P
## 170     8   193     93     4   161     66     2    75     20     0     4
## 171     4   167     80     9   114     43     3    60     16     0     4
##     SNRF_P SFF_P SRF_O SNRF_O SFF_O SRF_A SNRF_A SFF_A SRT_P SNRT_P SFT_P
## 170     11     0     4      8     1     1      3    -1     9      6     2
## 171     14     0     2      8     0     2      3     0     7      3     5
##     SRT_O SNRT_O SFT_O SRT_A SNRT_A SFT_A
## 170     7     11     3     7      0     0
## 171     4      2     0     3      2     2
```

CV for FRF_A

```
pred_arima <- c()
pred_mlp <- c()
pred_HW <- c()
pred_bsts <- c()
pred_ets <- c()
pred_baggedETS <- c()

i <- 121
while(i < nrow(new1)){
  #ts <- ts(new1[1:i, "FRF_A"], start=c(2003,10), frequency=12)
  a <- scale(new1[1:i, "FRF_A"])
  ts <- ts(as.vector(a), start=c(2003,10), frequency=12)

  p_arima <-forecast(auto.arima(ts), 10)$mean[1:10]
  p_mlp <- forecast(mlp(ts,hd=5), 10)$mean[1:10]
  p_HW <- forecast(HoltWinters(ts, beta=FALSE, gamma=TRUE), 10)$mean[1:10]
  bsts.model <-  bsts(ts, state.specification = AddSeasonal(AddLocalLinearTrend(list(), ts),
                                                          ts, nseasons = 12), niter = 500, ping=0, se
  p_bsts <- (predict.bsts(bsts.model, horizon = 10, burn = SuggestBurn(0.1, bsts.model), quantiles = c(
  p_ets <- forecast(ets(ts), 10)$mean[1:10]
  p_baggedETS <- forecast(baggedETS(ts,bootstrapped_series = bld.mbb.bootstrap(ts, 100)),10)$mean[1:10]

    for(j in 1:10){
      pred_arima <- c(pred_arima,round(sum(new1[,'FRF_A'][(i-1):i])+sum(p_arima[1:j]*attr(a, 'scaled:sca

      pred_mlp <- c(pred_mlp,round(sum(new1[,'FRF_A'][(i-1):i])+sum(p_mlp[1:j]*attr(a, 'scaled:scale')+a
      pred_HW <- c(pred_HW,round(sum(new1[,'FRF_A'][(i-1):i])+sum(p_HW[1:j]*attr(a, 'scaled:scale')+att
      pred_bsts <- c(pred_bsts,round(sum(new1[,'FRF_A'][(i-1):i])+sum(p_bsts[1:j]*attr(a, 'scaled:scale
      pred_ets <- c(pred_ets,round(sum(new1[,'FRF_A'][(i-1):i])+sum(p_ets[1:j]*attr(a, 'scaled:scale')+a
      pred_baggedETS <- c(pred_baggedETS,round(sum(new1[,'FRF_A'][(i-1):i])+sum(p_baggedETS[1:j]*attr(a

    }

  i = i +12
}
```

```
pred_prophet <- c()
i <- 121
while(i < nrow(new1)){

  data <-new1[1:i,c('Date','FRF_A')]
  colnames(data) <- c('ds','y')
  fit <- prophet(data,yearly.seasonality=T,weekly.seasonality=TRUE)#,mcmc.samples=2000)#mcmc decrease e
  future <- make_future_dataframe(fit, periods = 10,freq='month')
  forecast <- predict(fit, future)
  p_prophet <- tail(forecast,10)$yhat

  for(j in 1:10){
      pred_prophet <- c(pred_prophet,round(sum(new1[,'FRF_A'][(i-1):i])+sum(p_prophet[1:j])))
  }


  i=i+12
}
```

```
## Initial log joint probability = -6.16183
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -6.45264
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -4.79346
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -5.58938
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -6.89602
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
```

convert preds to real preds

```
actual <- monthly_data%>%select(Year,Date,FRF_A)%>%group_by(Year)%>%filter(month(Date) %in% c(1:8,11,12
actual <- actual[actual$Year %in% c(2013:2017),'FRF_A']
actual <- as.vector(actual$FRF_A)

ets <- accuracy(pred_ets, actual)
arima <- accuracy(pred_arima, actual)
mlp <- accuracy(pred_mlp, actual)
HW <- accuracy(pred_HW, actual)
baggedETS <- accuracy(pred_baggedETS, actual)
bsts <- accuracy(pred_bsts, actual)
prophet <- accuracy(pred_prophet, actual)

accuracy_matrix <- rbind(arima,mlp,HW,prophet,bsts,ets,baggedETS)
rownames(accuracy_matrix) <- c('arima','mlp','HW','prophet','bsts','ets','baggedETS')
```

```
accuracy_matrix
```

```
##                ME        RMSE    MAE           MPE         MAPE
## arima      -92.42 144.4334449  98.94  -3.187340923  3.394663173
## mlp        110.04 216.9574152 175.16   2.872856323  5.789271801
## HW         -82.46 122.0228667  90.06  -2.870258160  3.129168228
## prophet   -462.48 571.6781612 468.48 -14.147294789 14.430829634
## bsts      -125.52 259.6902771 175.52  -3.803395193  5.510860158
## ets        -88.00 126.3851257  96.04  -3.029328773  3.304252180
## baggedETS -470.34 561.8010146 473.46 -14.460768597 14.610503419
```

CV for FNRF_A

```
pred_arima <- c()
pred_mlp <- c()
pred_HW <- c()
pred_bsts <- c()
pred_ets <- c()
pred_baggedETS <- c()

i <- 121
while(i < nrow(new1)){
  #ts <- ts(new1[1:i, "FNRF_A"], start=c(2003,10), frequency=12)
  a <- scale(new1[1:i, "FNRF_A"])
  ts <- ts(as.vector(a), start=c(2003,10), frequency=12)

  p_arima <-forecast(auto.arima(ts), 10)$mean[1:10]
  p_mlp <- forecast(mlp(ts,hd=5), 10)$mean[1:10]
  p_HW <- forecast(HoltWinters(ts, beta=FALSE, gamma=TRUE), 10)$mean[1:10]
  bsts.model <-  bsts(ts, state.specification = AddSeasonal(AddLocalLinearTrend(list(), ts),
                                                            ts, nseasons = 12), niter = 500, ping=0, se
  p_bsts <- (predict.bsts(bsts.model, horizon = 10, burn = SuggestBurn(0.1, bsts.model), quantiles = c(
  p_ets <- forecast(ets(ts), 10)$mean[1:10]
  p_baggedETS <- forecast(baggedETS(ts,bootstrapped_series = bld.mbb.bootstrap(ts, 100)),10)$mean[1:10]

    for(j in 1:10){
      pred_arima <- c(pred_arima,round(sum(new1[,'FNRF_A'][(i-1):i])+sum(p_arima[1:j]*attr(a, 'scaled:s

      pred_mlp <- c(pred_mlp,round(sum(new1[,'FNRF_A'][(i-1):i])+sum(p_mlp[1:j]*attr(a, 'scaled:scale')
      pred_HW <- c(pred_HW,round(sum(new1[,'FNRF_A'][(i-1):i])+sum(p_HW[1:j]*attr(a, 'scaled:scale')+at
      pred_bsts <- c(pred_bsts,round(sum(new1[,'FNRF_A'][(i-1):i])+sum(p_bsts[1:j]*attr(a, 'scaled:scal
      pred_ets <- c(pred_ets,round(sum(new1[,'FNRF_A'][(i-1):i])+sum(p_ets[1:j]*attr(a, 'scaled:scale')
      pred_baggedETS <- c(pred_baggedETS,round(sum(new1[,'FNRF_A'][(i-1):i])+sum(p_baggedETS[1:j]*attr(a

    }

  i = i +12
}

pred_prophet <- c()
i <- 121
```

```
while(i < nrow(new1)){

  data <-new1[1:i,c('Date','FNRF_A')]
  colnames(data) <- c('ds','y')
  fit <- prophet(data,yearly.seasonality=T,weekly.seasonality=TRUE)#,mcmc.samples=2000)#mcmc decrease e
  future <- make_future_dataframe(fit, periods = 10,freq='month')
  forecast <- predict(fit, future)
  p_prophet <- tail(forecast,10)$yhat

  for(j in 1:10){
      pred_prophet <- c(pred_prophet,round(sum(new1[,'FNRF_A'][(i-1):i])+sum(p_prophet[1:j])))
  }


  i=i+12
}
```

```
## Initial log joint probability = -5.7961
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -7.75793
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -5.92601
## Optimization terminated normally:
##   Convergence detected: relative gradient magnitude is below tolerance
## Initial log joint probability = -5.65997
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -6.1386
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
```

opt for FNRF_A

```
actual <- monthly_data%>%select(Year,Date,FNRF_A)%>%group_by(Year)%>%filter(month(Date) %in% c(1:8,11,1
actual <- actual[actual$Year %in% c(2013:2017),'FNRF_A']
actual <- as.vector(actual$FNRF_A)

ets <- accuracy(pred_ets, actual)
arima <- accuracy(pred_arima, actual)
mlp <- accuracy(pred_mlp, actual)
HW <- accuracy(pred_HW, actual)
baggedETS <- accuracy(pred_baggedETS, actual)
bsts <- accuracy(pred_bsts, actual)
prophet <- accuracy(pred_prophet, actual)

accuracy_matrix <- rbind(arima,mlp,HW,prophet,bsts,ets,baggedETS)
rownames(accuracy_matrix) <- c('arima','mlp','HW','prophet','bsts','ets','baggedETS')
accuracy_matrix
```

```
##                ME        RMSE    MAE          MPE         MAPE
## arima     -129.92 163.9064367 142.04 -6.793653929   7.536708026
```

```
## mlp        -61.72 146.2970950 122.04 -3.924944320  6.588013421
## HW         -71.94 166.4780466 151.82 -4.243637597  7.887754205
## prophet   -158.80 258.4589716 207.64 -7.244531265 10.115977540
## bsts      -118.76 182.4556933 146.24 -5.870957630  7.510730738
## ets       -104.18 141.9341397 118.62 -5.646987205  6.384390875
## baggedETS -184.84 278.6091169 225.08 -8.775512125 11.204541918
```

CV for FFF_A

```
pred_arima <- c()
pred_mlp <- c()
pred_HW <- c()
pred_bsts <- c()
pred_ets <- c()
pred_baggedETS <- c()

i <- 121
while(i < nrow(new1)){
  a <- scale(new1[1:i, "FFF_A"])
  ts <- ts(as.vector(a), start=c(2003,10), frequency=12)

  p_arima <-forecast(auto.arima(ts), 10)$mean[1:10]
  p_mlp <- forecast(mlp(ts,hd=5), 10)$mean[1:10]
  p_HW <- forecast(HoltWinters(ts, beta=FALSE, gamma=TRUE), 10)$mean[1:10]
  bsts.model <-  bsts(ts, state.specification = AddSeasonal(AddLocalLinearTrend(list(), ts),
                                                           ts, nseasons = 12), niter = 500, ping=0, se
  p_bsts <- (predict.bsts(bsts.model, horizon = 10, burn = SuggestBurn(0.1, bsts.model), quantiles = c(
  p_ets <- forecast(ets(ts), 10)$mean[1:10]
  p_baggedETS <- forecast(baggedETS(ts,bootstrapped_series = bld.mbb.bootstrap(ts, 100)),10)$mean[1:10]

    for(j in 1:10){
      pred_arima <- c(pred_arima,round(sum(new1[,'FFF_A'][(i-1):i])+sum(p_arima[1:j]*attr(a, 'scaled:sca

      pred_mlp <- c(pred_mlp,round(sum(new1[,'FFF_A'][(i-1):i])+sum(p_mlp[1:j]*attr(a, 'scaled:scale')+a
      pred_HW <- c(pred_HW,round(sum(new1[,'FFF_A'][(i-1):i])+sum(p_HW[1:j]*attr(a, 'scaled:scale')+att
      pred_bsts <- c(pred_bsts,round(sum(new1[,'FFF_A'][(i-1):i])+sum(p_bsts[1:j]*attr(a, 'scaled:scale
      pred_ets <- c(pred_ets,round(sum(new1[,'FFF_A'][(i-1):i])+sum(p_ets[1:j]*attr(a, 'scaled:scale')+a
      pred_baggedETS <- c(pred_baggedETS,round(sum(new1[,'FFF_A'][(i-1):i])+sum(p_baggedETS[1:j]*attr(a

    }

  i = i +12
}

pred_prophet <- c()
i <- 121
while(i < nrow(new1)){

  data <-new1[1:i,c('Date','FFF_A')]
  colnames(data) <- c('ds','y')
  fit <- prophet(data,yearly.seasonality=T,weekly.seasonality=TRUE)#,mcmc.samples=2000)#mcmc decrease e
  future <- make_future_dataframe(fit, periods = 10,freq='month')
```

```
  forecast <- predict(fit, future)
  p_prophet <- tail(forecast,10)$yhat

  for(j in 1:10){
      pred_prophet <- c(pred_prophet,round(sum(new1[,'FFF_A'][(i-1):i])+sum(p_prophet[1:j])))
  }


  i=i+12
}
```

```
## Initial log joint probability = -5.22602
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -5.66414
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -5.99382
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -6.48336
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -6.57691
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
```

```
actual <- monthly_data%>%select(Year,Date,FFF_A)%>%group_by(Year)%>%filter(month(Date) %in% c(1:8,11,12
actual <- actual[actual$Year %in% c(2013:2017),'FFF_A']
actual <- as.vector(actual$FFF_A)

ets <- accuracy(pred_ets, actual)
arima <- accuracy(pred_arima, actual)
mlp <- accuracy(pred_mlp, actual)
HW <- accuracy(pred_HW, actual)
baggedETS <- accuracy(pred_baggedETS, actual)
bsts <- accuracy(pred_bsts, actual)
prophet <- accuracy(pred_prophet, actual)

accuracy_matrix <- rbind(arima,mlp,HW,prophet,bsts,ets,baggedETS)
rownames(accuracy_matrix) <- c('arima','mlp','HW','prophet','bsts','ets','baggedETS')
accuracy_matrix
```

```
##                ME        RMSE   MAE           MPE        MAPE
## arima       -1.56 10.775899034  6.80   -5.538836294  11.018377288
## mlp        -15.06 29.937601774 21.38  -21.329950326  29.634639214
## HW           0.10  1.529705854  1.30   -1.908785218   5.787530334
## prophet    -59.78 77.577702982 61.02 -121.069827656 121.477498093
## bsts       -13.74 22.611943747 19.38  -51.029093896  57.423528968
## ets         -2.32 14.455448800  8.88  -12.851965686  17.598854401
## baggedETS  -12.98 32.248720905 26.90  -63.029416980  68.095782791
```

```
pred_arima <- c()
pred_mlp <- c()
pred_HW <- c()
```

```
pred_bsts <- c()
pred_ets <- c()
pred_baggedETS <- c()

i <- 121
while(i < nrow(new1)){
  a <- scale(new1[1:i, "FRT_A"])
  ts <- ts(as.vector(a), start=c(2003,10), frequency=12)

  p_arima <-forecast(auto.arima(ts), 10)$mean[1:10]
  p_mlp <- forecast(mlp(ts,hd=5), 10)$mean[1:10]
  p_HW <- forecast(HoltWinters(ts, beta=FALSE, gamma=TRUE), 10)$mean[1:10]
  bsts.model <-  bsts(ts, state.specification = AddSeasonal(AddLocalLinearTrend(list(), ts),
                                                            ts, nseasons = 12), niter = 500, ping=0, see
  p_bsts <- (predict.bsts(bsts.model, horizon = 10, burn = SuggestBurn(0.1, bsts.model), quantiles = c(
  p_ets <- forecast(ets(ts), 10)$mean[1:10]
  p_baggedETS <- forecast(baggedETS(ts,bootstrapped_series = bld.mbb.bootstrap(ts, 100)),10)$mean[1:10]

    for(j in 1:10){
      pred_arima <- c(pred_arima,round(sum(new1[,'FRT_A'][(i-1):i])+sum(p_arima[1:j]*attr(a, 'scaled:sca

      pred_mlp <- c(pred_mlp,round(sum(new1[,'FRT_A'][(i-1):i])+sum(p_mlp[1:j]*attr(a, 'scaled:scale')+a
      pred_HW <- c(pred_HW,round(sum(new1[,'FRT_A'][(i-1):i])+sum(p_HW[1:j]*attr(a, 'scaled:scale')+attr
      pred_bsts <- c(pred_bsts,round(sum(new1[,'FRT_A'][(i-1):i])+sum(p_bsts[1:j]*attr(a, 'scaled:scale
      pred_ets <- c(pred_ets,round(sum(new1[,'FRT_A'][(i-1):i])+sum(p_ets[1:j]*attr(a, 'scaled:scale')+a
      pred_baggedETS <- c(pred_baggedETS,round(sum(new1[,'FRT_A'][(i-1):i])+sum(p_baggedETS[1:j]*attr(a

    }

  i = i +12
}

pred_prophet <- c()
i <- 121
while(i < nrow(new1)){

  data <-new1[1:i,c('Date','FRT_A')]
  colnames(data) <- c('ds','y')
  fit <- prophet(data,yearly.seasonality=T,weekly.seasonality=TRUE)#,mcmc.samples=2000)#mcmc decrease e
  future <- make_future_dataframe(fit, periods = 10,freq='month')
  forecast <- predict(fit, future)
  p_prophet <- tail(forecast,10)$yhat

  for(j in 1:10){
      pred_prophet <- c(pred_prophet,round(sum(new1[,'FRT_A'][(i-1):i])+sum(p_prophet[1:j])))
  }


  i=i+12
}
```

```
## Initial log joint probability = -8.72314
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
```

```
## Initial log joint probability = -8.90163
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -9.45713
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
## Initial log joint probability = -10.4001
## Optimization terminated normally:
##   Convergence detected: relative gradient magnitude is below tolerance
## Initial log joint probability = -9.94197
## Optimization terminated normally:
##   Convergence detected: absolute parameter change was below tolerance
```

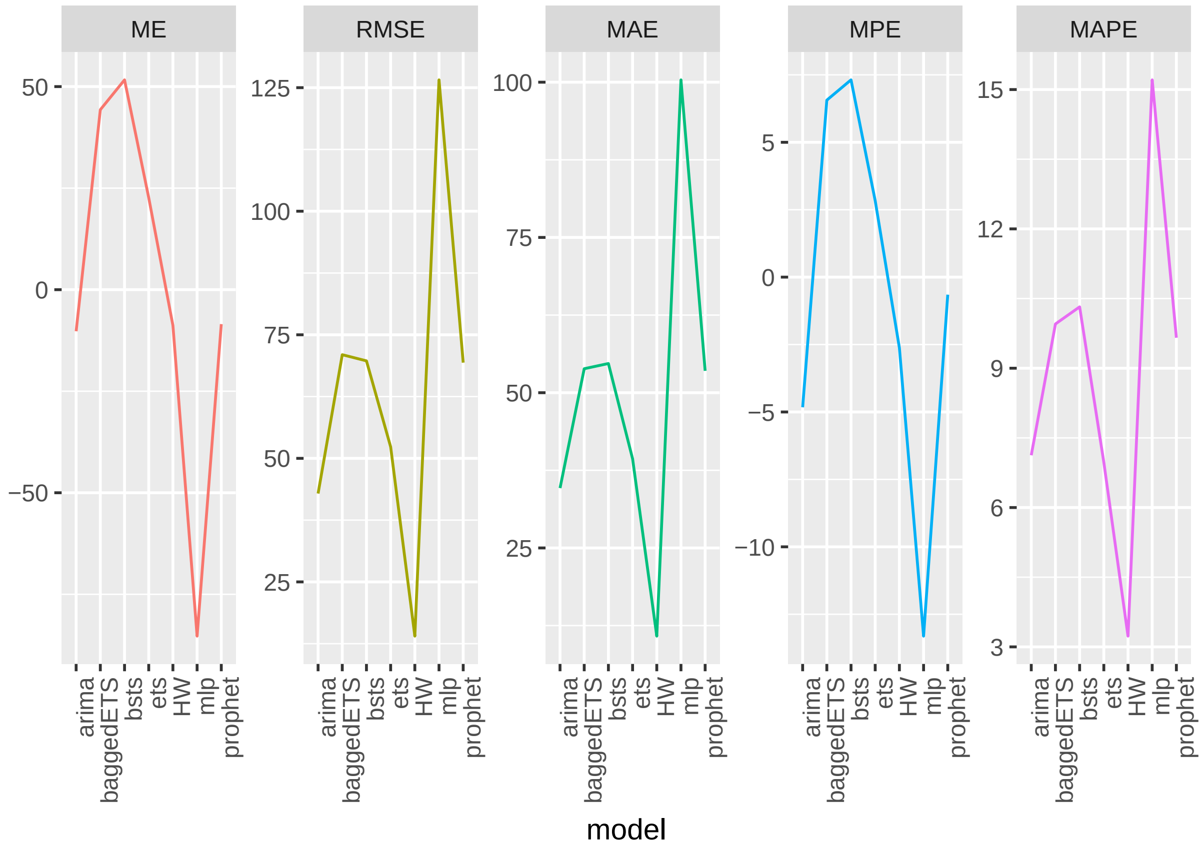
```
actual <- monthly_data%>%select(Year,Date,FRT_A)%>%group_by(Year)%>%filter(month(Date) %in% c(1:8,11,12
actual <- actual[actual$Year %in% c(2013:2017),'FRT_A']
actual <- as.vector(actual$FRT_A)

ets <- accuracy(pred_ets, actual)
arima <- accuracy(pred_arima, actual)
mlp <- accuracy(pred_mlp, actual)
HW <- accuracy(pred_HW, actual)
baggedETS <- accuracy(pred_baggedETS, actual)
bsts <- accuracy(pred_bsts, actual)
prophet <- accuracy(pred_prophet, actual)

accuracy_matrix <- rbind(arima,mlp,HW,prophet,bsts,ets,baggedETS)
rownames(accuracy_matrix) <- c('arima','mlp','HW','prophet','bsts','ets','baggedETS')
accuracy_matrix
```

```
##                ME         RMSE    MAE            MPE         MAPE
## arima      -10.24   42.88216412  34.64  -4.8261387437  7.126176821
## mlp        -85.30 126.58301624 100.38 -13.3111516153 15.209226538
## HW          -8.80   14.02854233  10.80  -2.6208991607  3.231486845
## prophet     -8.48   69.36223756  53.52  -0.6496365456  9.656472643
## bsts        51.66   69.72417084  54.70   7.3152855285 10.321295924
## ets         22.64   52.32475514  39.36   2.8335527648  6.974171429
## baggedETS   44.30   70.97055728  53.86   6.5625777357  9.952583606
```

model comparison

```
library(tibble)
library(reshape2)
accuracy_df <- as.data.frame(accuracy_matrix)
accuracy_df <- rownames_to_column(accuracy_df, "model")
accuracy_df$model <- as.factor(accuracy_df$model)
accuracy_df <- melt(accuracy_df , id.vars = 'model')
```

```
ggplot(data=accuracy_df, aes(x=model, y=value, col=variable,group=variable))+
      geom_line()+
      guides(colour=FALSE)+
      facet_wrap(~variable, ncol=5,dir="h",scales='free_y')+
      theme(axis.text.x = element_text(angle = 90, hjust = 1))+
      ylab('')
```

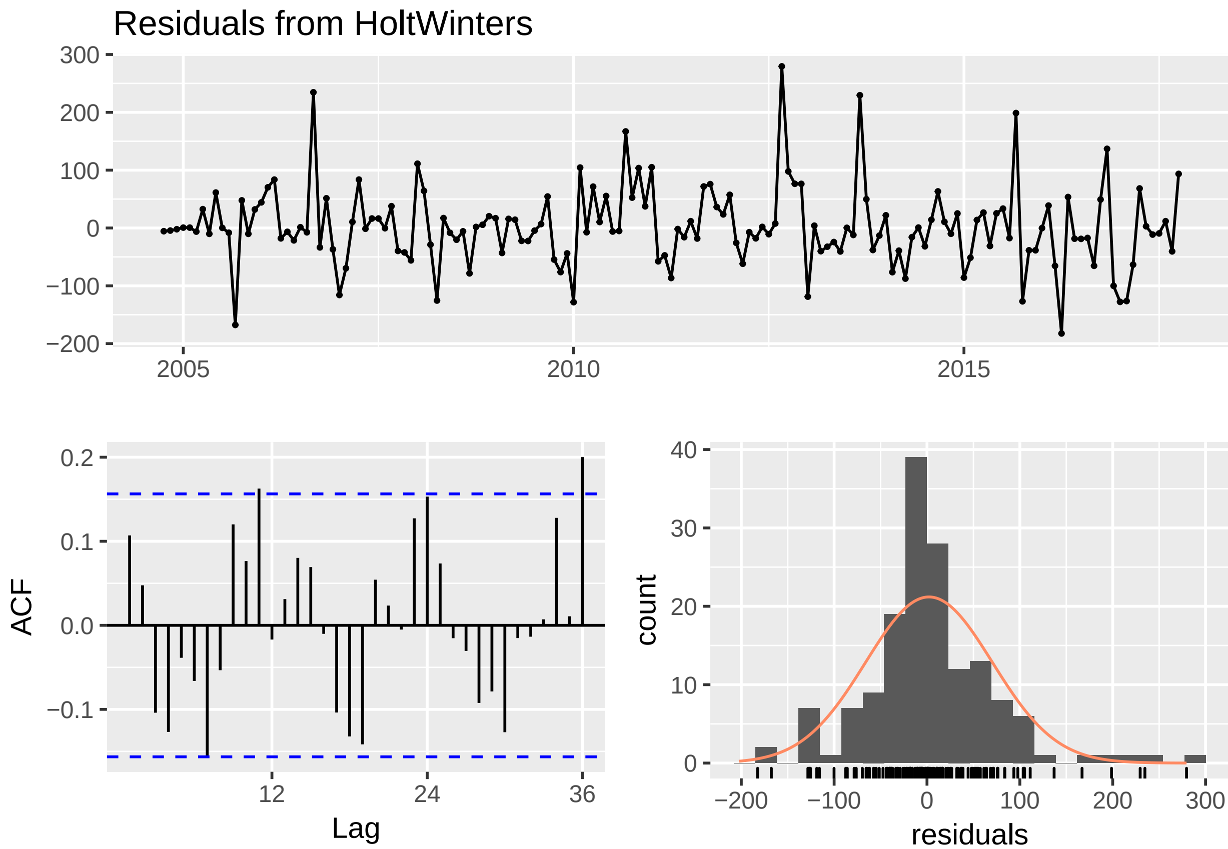


model fitting

```
new2<- new1%>%select(contains("A"))
new2 <- new2[,-2]
head(new2,3)
```

```
##         Date FRF_A FNRF_A FFF_A FRT_A FNRT_A FFT_A SRF_A SNRF_A SFF_A
## 1 2003-10-31   318     90     1    13      2     0     5      0     0
## 2 2003-11-26   318    129     0    19      4     0     3      1     1
## 3 2003-12-31   362    157     0    30      9     0     3      3     0
##   SRT_A SNRT_A SFT_A
## 1     1      1     0
## 2     0      0     0
## 3     1      1     0
```

```
ts <- ts(new2[1:169,2], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
checkresiduals(fit)
```




```
fore <- forecast(fit,10)
pred <- vector()
for(j in 1:10){
      pred<- c(pred,round(sum(new2$FRF_A[168:169])+sum(fore$mean[1:j])))
}
pred
```

```
##  [1] 2145 2460 2798 3011 3133 3170 3222 3226 3221 3205
```

forecast Nov-Aug using HolterWinters

```
pred_accept_df <- vector()
for(i in 2:13){
  a <- scale(new2[1:169, i])
  ts <- ts(as.vector(a), start=c(2003,10), frequency=12)
  fit <- HoltWinters(ts,beta=FALSE, gamma=TRUE)
  fore <- forecast(fit,10)
  pred <- vector()
   for(j in 1:10){
      pred<- c(pred,round(sum(new2[,i][168:169])+sum(fore$mean[1:j]*attr(a, 'scaled:scale')+attr(a, 'sca

      }
  pred_accept_df<- cbind(pred_accept_df, pred)
  i=i+1
```

```
}

colnames(pred_accept_df) <- colnames(new2)[2:13]
#pred_accept_df <-as.data.frame(pred_accept_df)
#row.names(pred_accept_df) <- seq(as.Date("2017/11/30"), by = "month", length.out = 10)

pred_accept_df
```

```
##       FRF_A FNRF_A FFF_A FRT_A FNRT_A FFT_A SRF_A SNRF_A SFF_A SRT_A
##  [1,]  2155   1313    11   165     44     3    11      8     1    10
##  [2,]  2460   1562    20   253     56     5    15      9     1    14
##  [3,]  2785   1859    33   360    102     5    47     37    43    20
##  [4,]  2985   2162    52   562    157    13    35     19     3    31
##  [5,]  3100   2389    86   751    211    43    48     25     5    45
##  [6,]  3141   2517   174   953    272    57    91     58     7    61
##  [7,]  3192   2583   220  1091    324    86    96     66    10    74
##  [8,]  3193   2565   236  1195    353    96    88     65    10    68
##  [9,]  3185   2540   237  1290    361   105    88     64    10    71
## [10,]  3169   2472   235  1353    358   102    88     64    10    72
##       SNRT_A SFT_A
##  [1,]      1     2
##  [2,]      2     2
##  [3,]      6     2
##  [4,]     10     3
##  [5,]     18     6
##  [6,]     24     8
##  [7,]     27    12
##  [8,]     21    10
##  [9,]     21    10
## [10,]     20    10
```

```
predicted_accepts <- pred_accept_df
```

forecast Nov-Aug using ets

```
pred_accept_df <- vector()
for(i in 2:13){
  a <- scale(new2[1:169, i])
  ts <- ts(as.vector(a), start=c(2003,10), frequency=12)
  fit <- ets(ts)
  fore <- forecast(fit,10)
  pred <- vector()
   for(j in 1:10){
      pred<- c(pred,round(sum(new2[,i][168:169])+sum(fore$mean[1:j]*attr(a, 'scaled:scale')+attr(a, 'sca


      }
  pred_accept_df<- cbind(pred_accept_df, pred)
  i=i+1

}

colnames(pred_accept_df) <- colnames(new2)[2:13]
```

```r
#pred_accept_df <-as.data.frame(pred_accept_df)
#row.names(pred_accept_df) <- seq(as.Date("2017/11/30"), by = "month", length.out = 10)
pred_accept_df
```

```
##       FRF_A FNRF_A FFF_A FRT_A FNRT_A FFT_A SRF_A SNRF_A SFF_A SRT_A
##  [1,]  2143   1309    10   156     40     3    11      8     2     8
##  [2,]  2456   1574    18   239     55     4    16      9     3     8
##  [3,]  2792   1884    32   342     90     4    38     30     9    12
##  [4,]  3003   2190    51   552    146    11    35     20     9    19
##  [5,]  3123   2413    89   754    201    39    48     28    11    34
##  [6,]  3158   2543   196   967    271    67    86     60    12    56
##  [7,]  3209   2605   243  1104    315    98    97     72    13    66
##  [8,]  3211   2583   259  1216    341   110    88     70     7    57
##  [9,]  3203   2557   260  1308    356   117    88     69     7    57
## [10,]  3186   2489   255  1367    355   114    88     69     7    57
##       SNRT_A SFT_A
##  [1,]      1     1
##  [2,]      1     1
##  [3,]      2     2
##  [4,]      3     2
##  [5,]      6     6
##  [6,]      8    15
##  [7,]     10    17
##  [8,]     12    19
##  [9,]     12    19
## [10,]     12    19
```

forecast Nov-Aug using arima

```r
pred_accept_df <- vector()
for(i in 2:13){
  a <- scale(new2[1:169, i])
  ts <- ts(as.vector(a), start=c(2003,10), frequency=12)
  fit <- auto.arima(ts)
  fore <- forecast(fit,10)
  pred <- vector()
   for(j in 1:10){
      pred<- c(pred,round(sum(new2[,i][168:169])+sum(fore$mean[1:j]*attr(a, 'scaled:scale')+attr(a, 'sca

      }
  pred_accept_df<- cbind(pred_accept_df, pred)
  i=i+1

}

colnames(pred_accept_df) <- colnames(new2)[2:13]
pred_accept_df
```

```
##       FRF_A FNRF_A FFF_A FRT_A FNRT_A FFT_A SRF_A SNRF_A SFF_A SRT_A
##  [1,]  2127   1308    14   176     45     3    11      8     1     8
##  [2,]  2466   1587    25   281     58     5    16     10     1    12
##  [3,]  2799   1932    38   405    101     5    42     31    14    16
##  [4,]  3003   2271    55   585    151    13    35     20     4    26
```

```
## [5,] 3119 2505  84  755   200  41  47  28  5  37
## [6,] 3123 2663 157  921   254  67  89  56  7  50
## [7,] 3171 2750 197 1055   301  98  96  71  8  60
## [8,] 3164 2748 215 1152   330 110  87  70  6  55
## [9,] 3164 2738 221 1234   340 118  87  68  6  56
## [10,] 3147 2695 225 1296  340 115  87  68  6  57
##       SNRT_A SFT_A
##  [1,]      2     3
##  [2,]      3     4
##  [3,]      5     5
##  [4,]      7     7
##  [5,]     10     9
##  [6,]     13    11
##  [7,]     15    14
##  [8,]     13    14
##  [9,]     14    14
## [10,]     14    15
```

Fitted vs actual

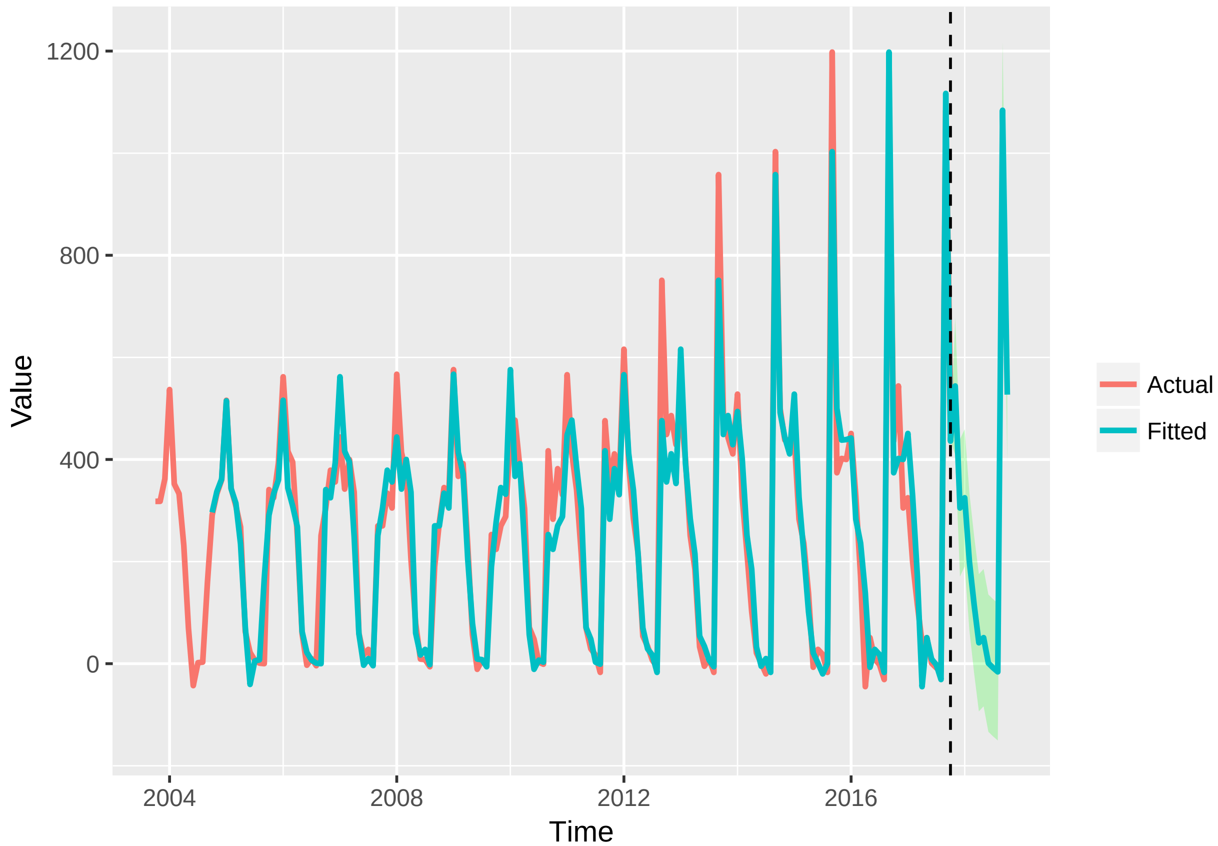
```
HWplot<-function(ts_object,  n.ahead=12,  CI=.95,  error.ribbon='green', line.size=1){

  hw_object<-HoltWinters(ts_object,beta=F, gamma=T)

  forecast<-predict(hw_object,  n.ahead=n.ahead,  prediction.interval=T,  level=CI)


  for_values<-data.frame(time=round(time(forecast),  3),  value_forecast=as.data.frame(forecast)$fit,  d

  fitted_values<-data.frame(time=round(time(hw_object$fitted),  3),  value_fitted=as.data.frame(hw_obje

  actual_values<-data.frame(time=round(time(hw_object$x),  3),  Actual=c(hw_object$x))


  graphset<-merge(actual_values,  fitted_values,  by='time',  all=TRUE)
  graphset<-merge(graphset,  for_values,  all=TRUE,  by='time')
  graphset[is.na(graphset$dev),  ]$dev<-0

  graphset$Fitted<-c(rep(NA,  NROW(graphset)-(NROW(for_values) + NROW(fitted_values))),  fitted_values$v


  graphset.melt<-melt(graphset[, c('time', 'Actual', 'Fitted')], id='time')

  p<-ggplot(graphset.melt,  aes(x=time,  y=value)) + geom_ribbon(data=graphset, aes(x=time, y=Fitted, ym
  return(p)

}
```

FRF

```r
ts <- ts(new2[1:169,2], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

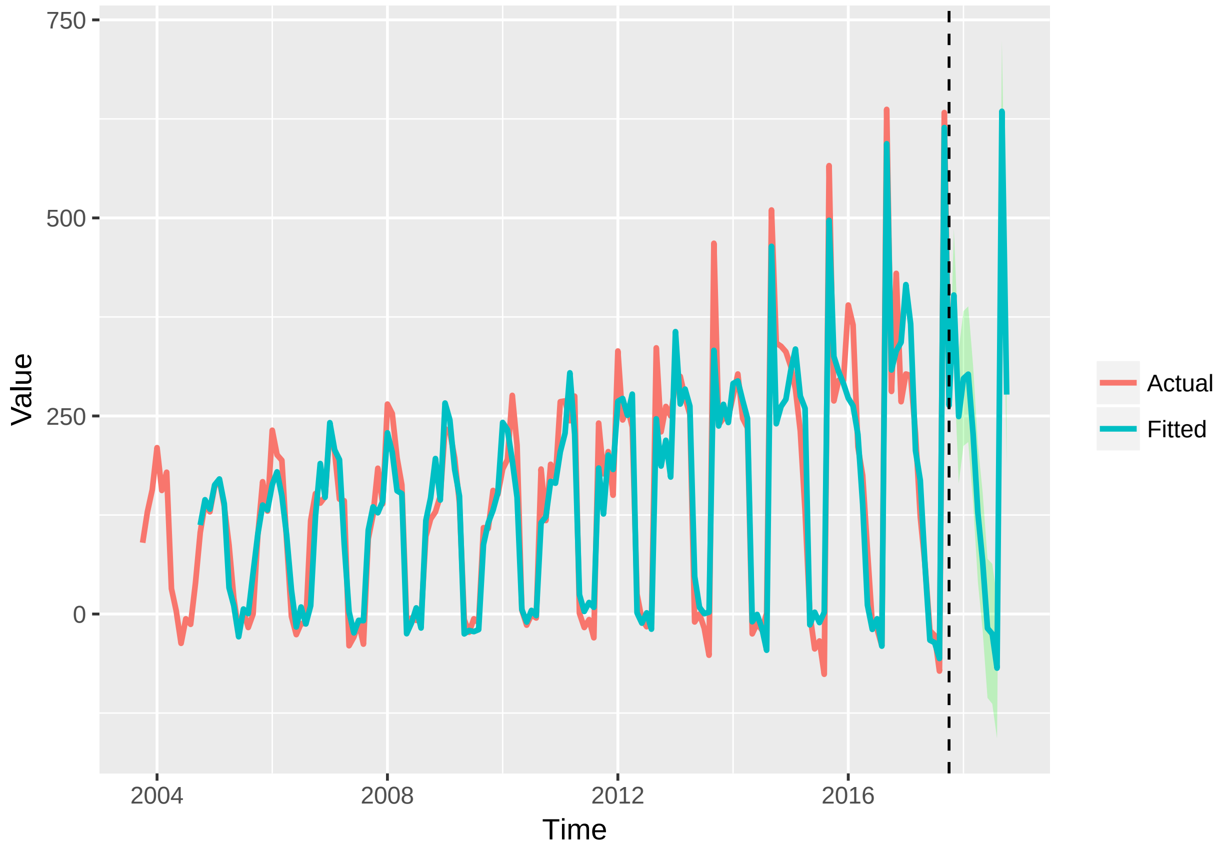


##FNRF

```r
ts <- ts(new2[1:169,3], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

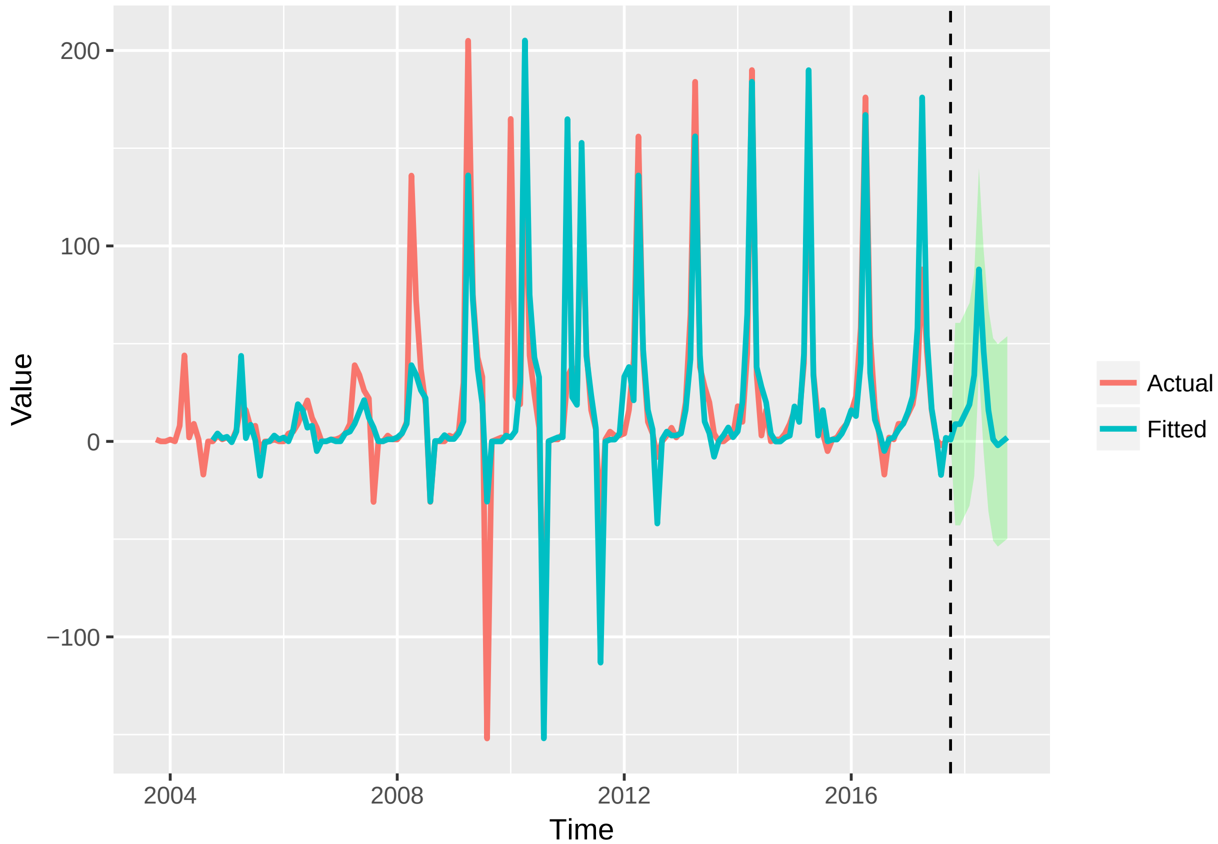


#FFF

```
ts <- ts(new2[1:169,4], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```



##FRT

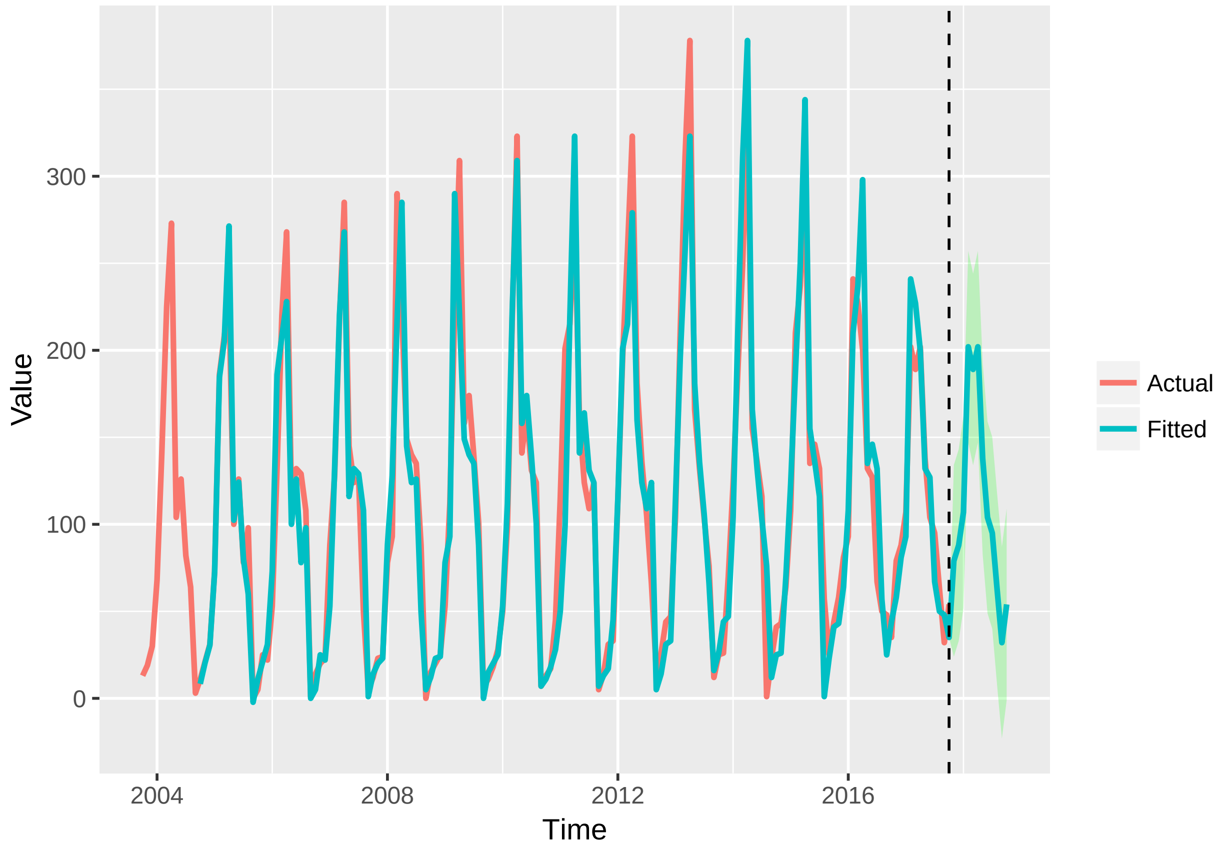
```
ts <- ts(new2[1:169,5], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
```

```
## Warning in HoltWinters(ts): optimization difficulties: ERROR:
## ABNORMAL_TERMINATION_IN_LNSRCH
```

```
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

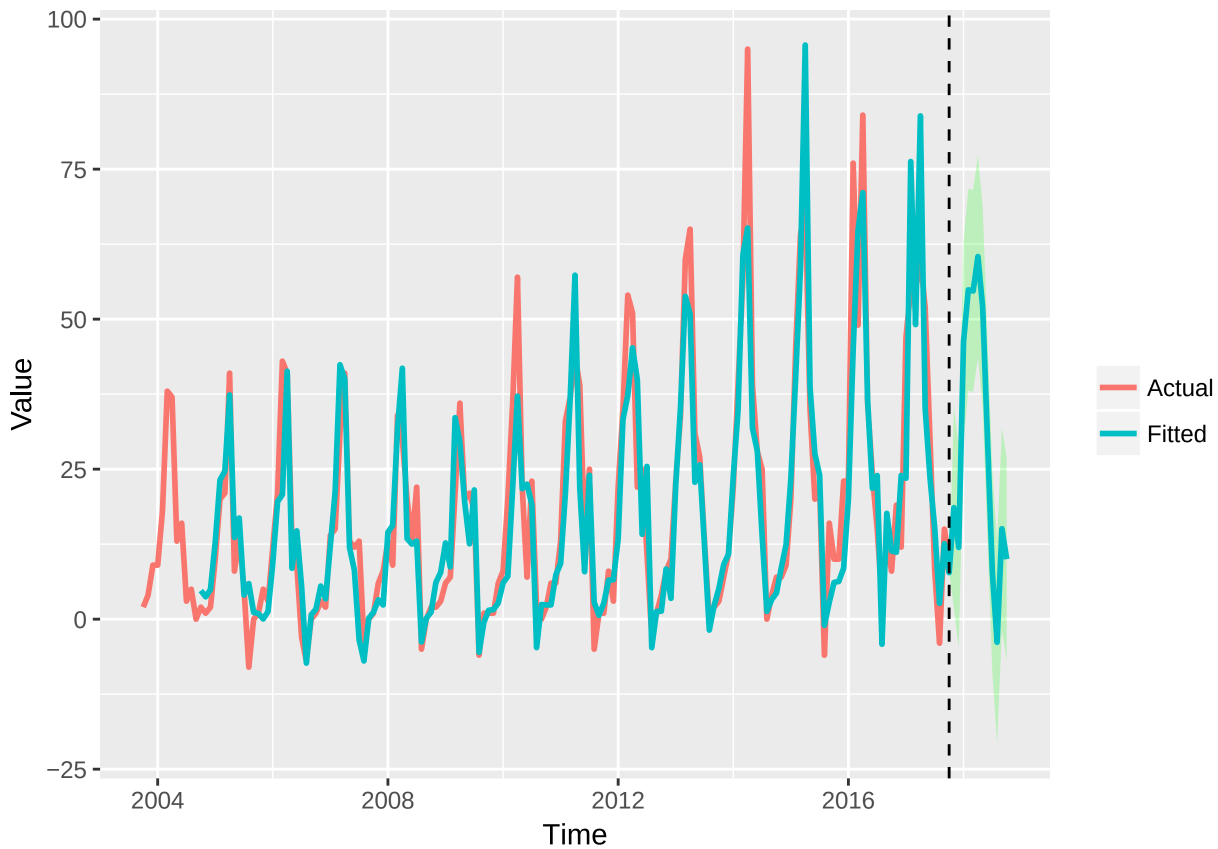


##FNRT

```
ts <- ts(new2[1:169,6], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

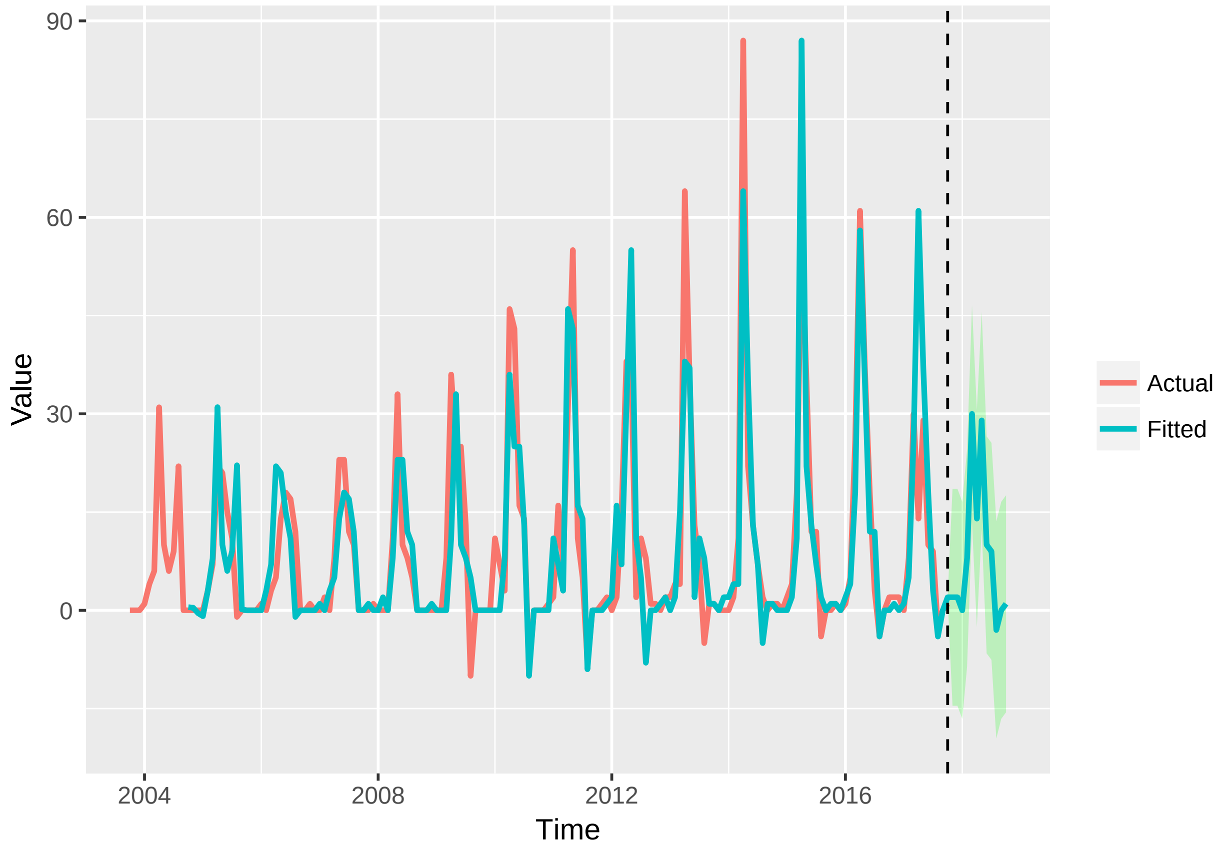


##FFT

```
ts <- ts(new2[1:169,7], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

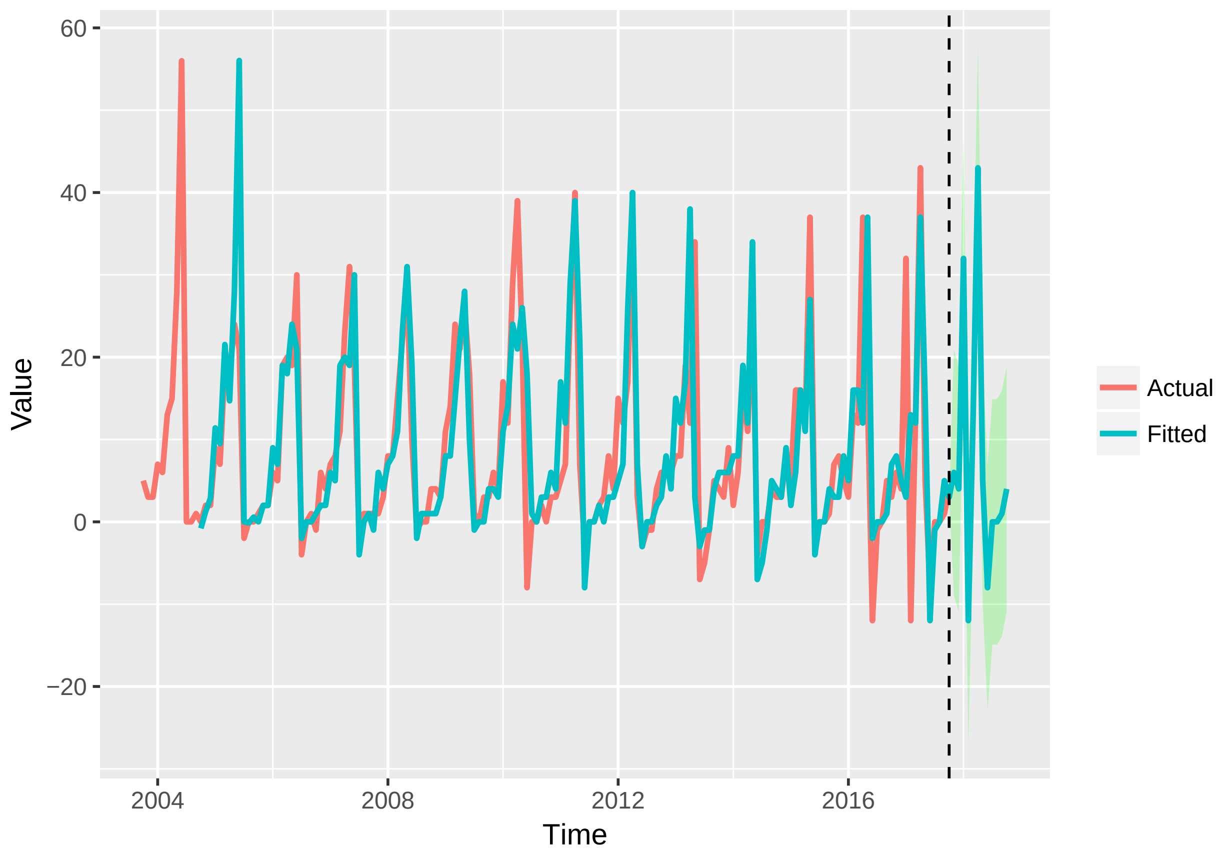


##SRF

```
ts <- ts(new2[1:169,8], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

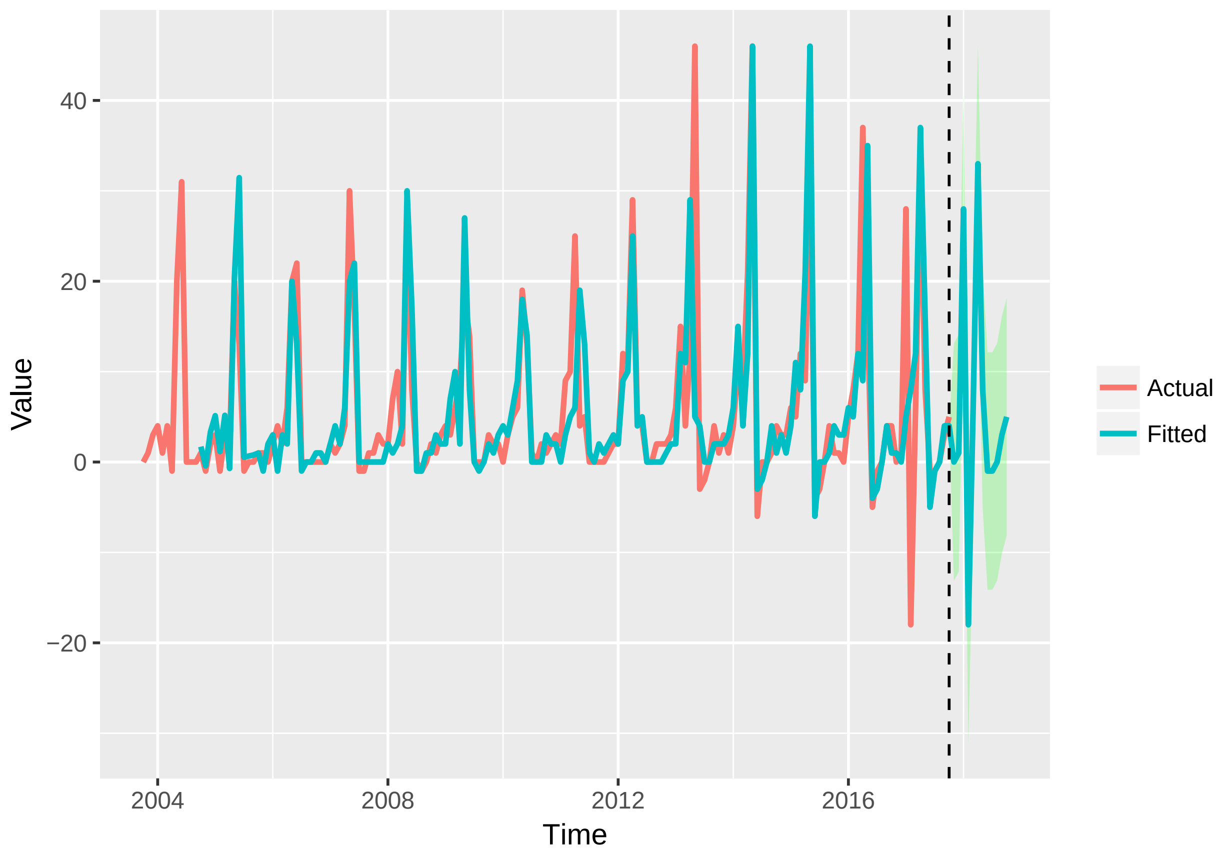


##SNRF

```
ts <- ts(new2[1:169,9], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

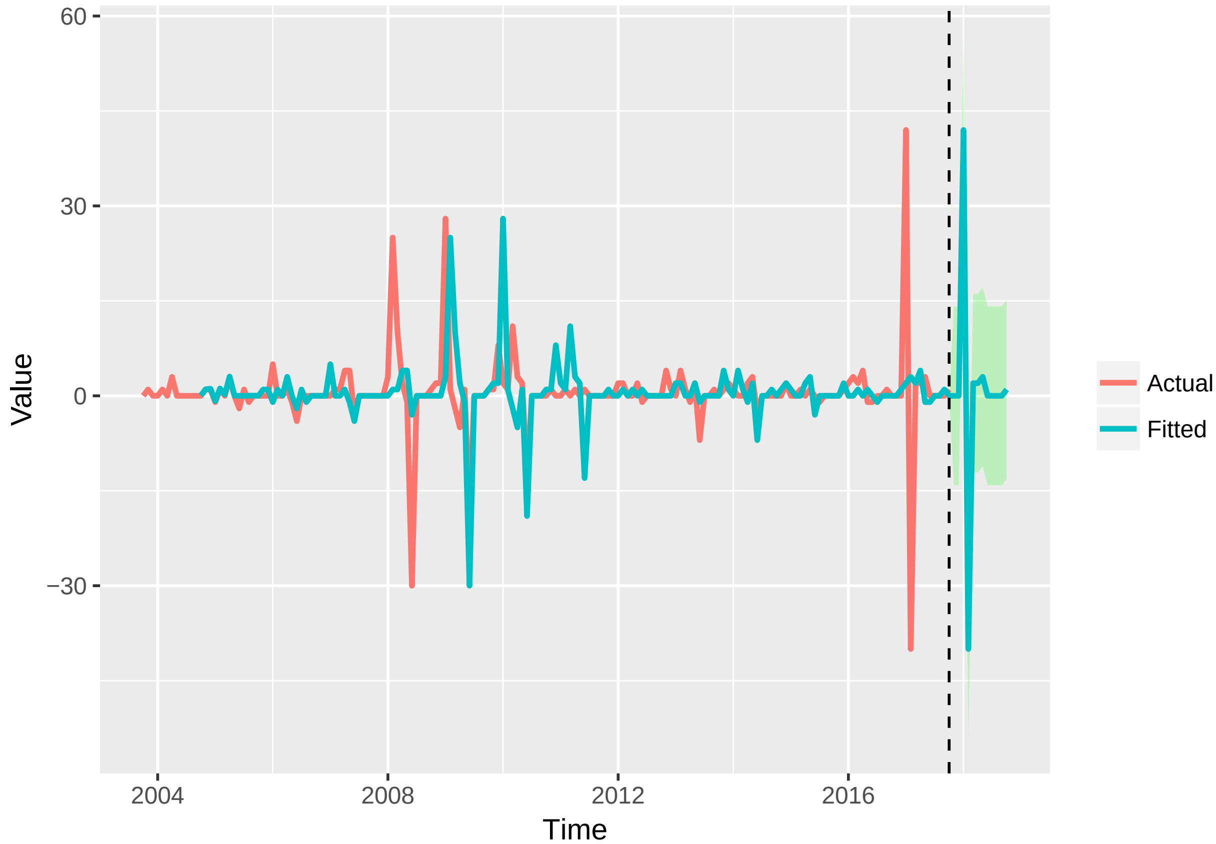


#SFF

```
ts <- ts(new2[1:169,10], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

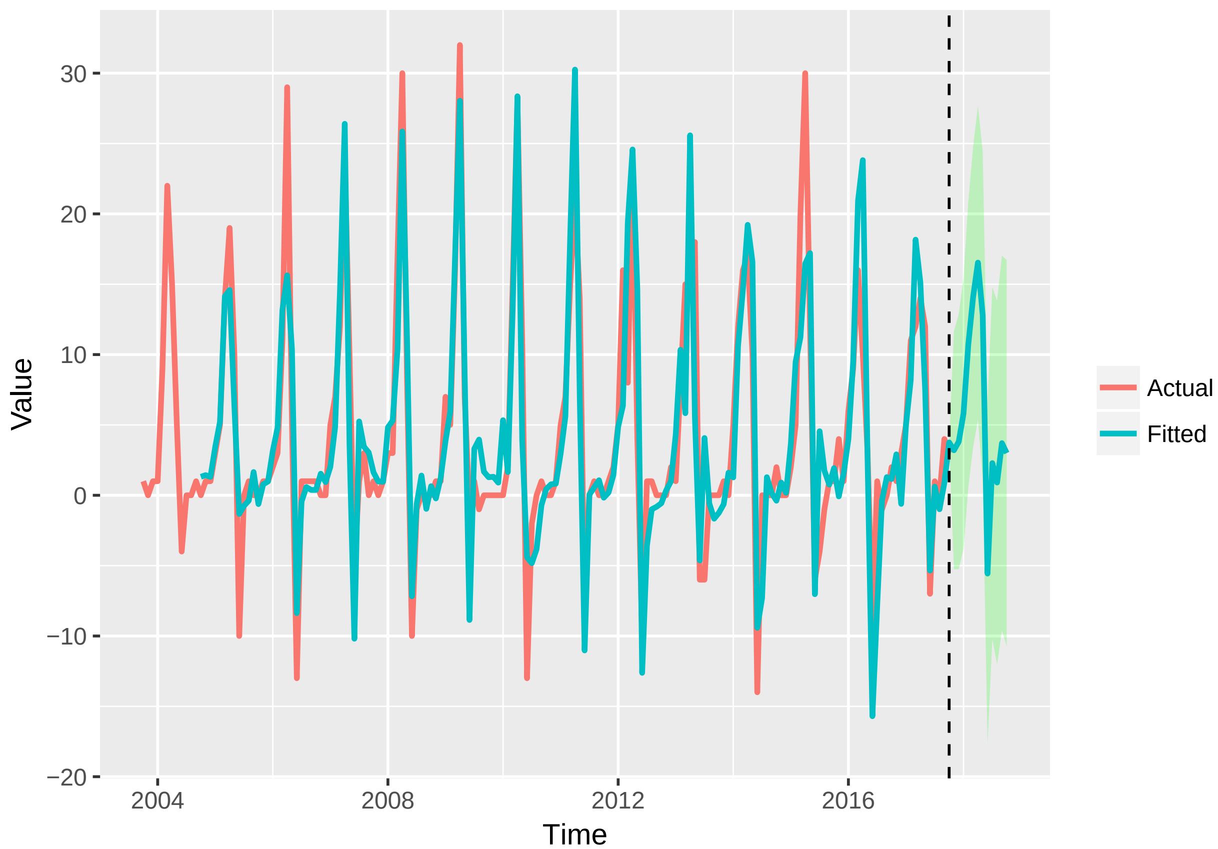


##SRT

```
ts <- ts(new2[1:169,11], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

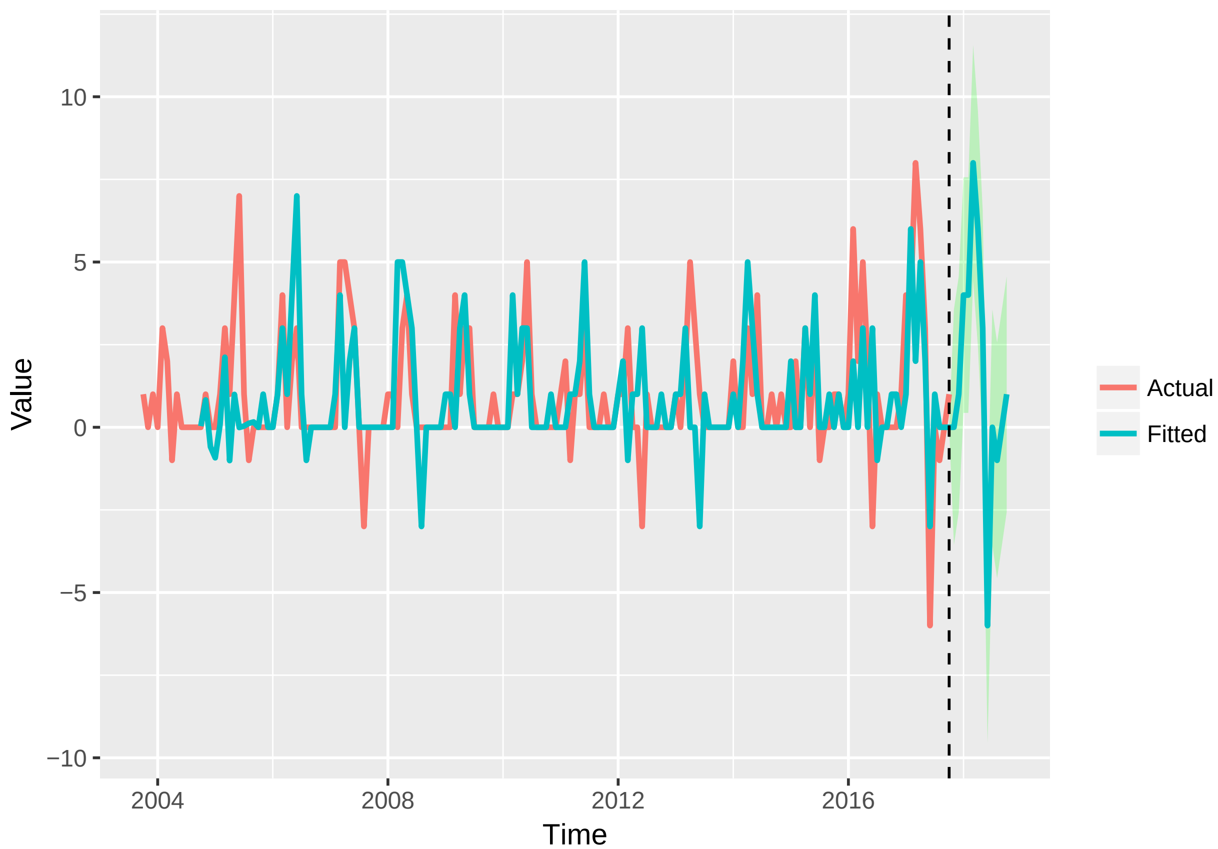


##SNRT

```
ts <- ts(new2[1:169,12], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```

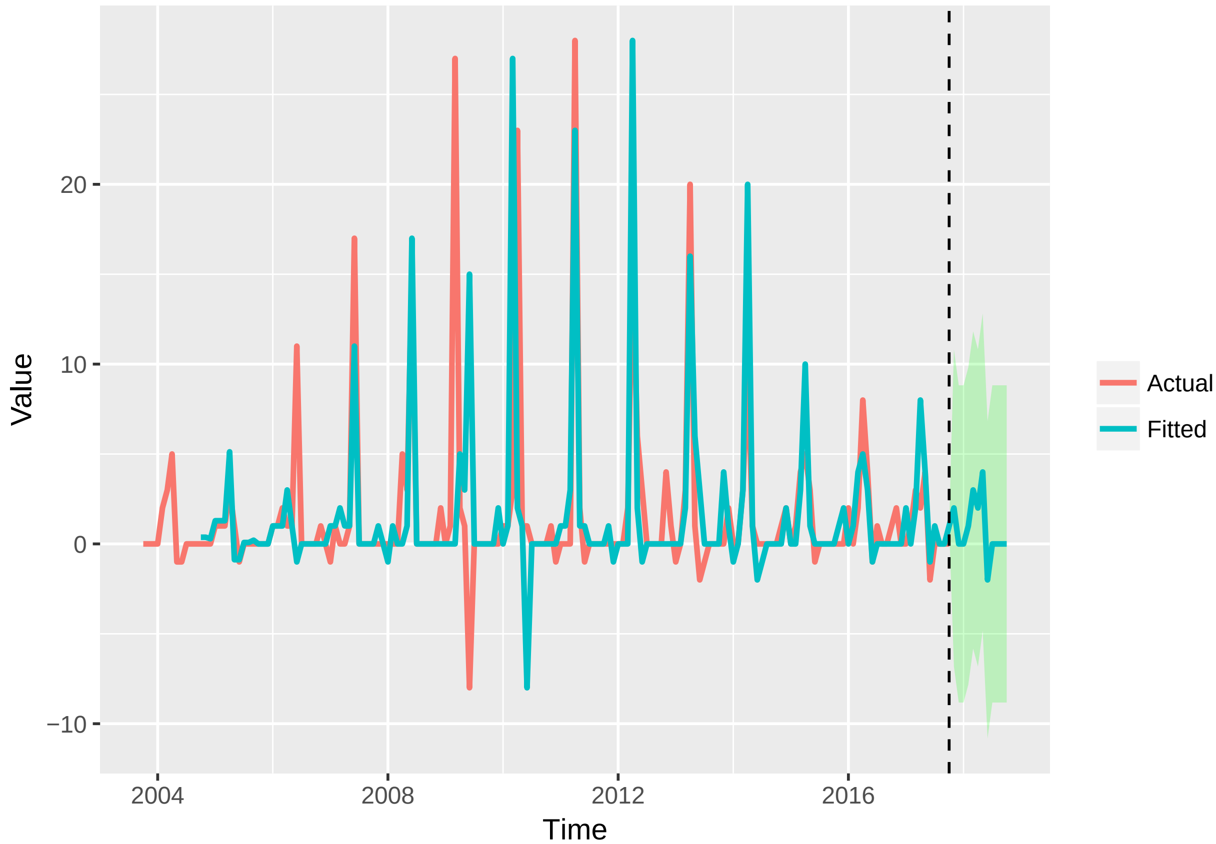


##SFT

```
ts <- ts(new2[1:169,13], start=c(2003,10), frequency=12)
fit <- HoltWinters(ts)
HWplot(ts, n.ahead = 12)
```

```
## Warning: Ignoring unknown aesthetics: y
```

```
## Warning: Removed 24 rows containing missing values (geom_path).
```



regression enroll on accept

```
final_accepts <- new_data[month(new_data$Date)==8,]
final_accepts_FS <- add(final_accepts[,2:7]+final_accepts[,8:13])
final_enrolls_FS <- read.csv("D:/Work/actual_enrol_FS.csv")

beta0 <- vector()
beta1 <- vector()
r_squared <-vector()

for(i in 1:6){
  fit_lm <- lm(final_enrolls_FS[,i]~final_accepts_FS[,i])
  beta0 <- c(beta0,summary(fit_lm)$coefficients[1])
  beta1 <- c(beta1,summary(fit_lm)$coefficients[2])
  r_squared <- c(r_squared,summary(fit_lm)$r.squared)
}

regression_matrix <- cbind(beta0=beta0,beta1=beta1,r_squared=r_squared)
row.names(regression_matrix) <- colnames(final_enrolls_FS)

regression_matrix
```

```
##            beta0       beta1    r_squared
## RF  73.939368567 0.9711277948 0.9959973124
## NRF  2.269236815 0.9840619230 0.9995922023
```

```
## FF  -2.373942470 0.9009744562 0.9832725919
## RT  28.583994459 0.9626212247 0.9821580650
## NRT  8.755461846 0.9060818792 0.9935604213
## FT   5.108378917 0.9137058433 0.9798333400
```

combine fall and summer for each type of students

```
predicted_accepts_FS <- add(predicted_accepts[,1:6]+predicted_accepts[,7:12])
predicted_accepts_FS
```

```
##       FRF_A FNRF_A FFF_A FRT_A FNRT_A FFT_A
##  [1,]  2166   1321    12   175     45     5
##  [2,]  2475   1571    21   267     58     7
##  [3,]  2832   1896    76   380    108     7
##  [4,]  3020   2181    55   593    167    16
##  [5,]  3148   2414    91   796    229    49
##  [6,]  3232   2575   181  1014    296    65
##  [7,]  3288   2649   230  1165    351    98
##  [8,]  3281   2630   246  1263    374   106
##  [9,]  3273   2604   247  1361    382   115
## [10,]  3257   2536   245  1425    378   112
```

predict Enrollment in August 2018 using forecasted values of Accept

```
predicted_enrollments_FS <- vector()
for(i in 1:6){
  a<- regression_matrix[i,1]+regression_matrix[i,2]*predicted_accepts_FS[10,i]
  predicted_enrollments_FS <- c(predicted_enrollments_FS,a)
}
predicted_enrollments_FS <- t(predicted_enrollments_FS)
colnames(predicted_enrollments_FS) <- colnames(final_enrolls_FS)
predicted_enrolls <- round(predicted_enrollments_FS,0)
predicted_enrolls
```

```
##        RF  NRF  FF   RT NRT  FT
## [1,] 3237 2498 218 1400 351 107
```