meeting report 5.27

linyiguo

2019-5-27

Dear Aaron,

I am writing to give you a brief about the stuff that I have done in last three days.

# I Simulation

I tried to simulate data from a fixed seasonal arima model, the code is mainly referred to [this website](https://robjhyndman.com/hyndsight/simulating-from-a-specified-seasonal-arima-model/). Of course I tried some other methods as well, but this one seems to be more reliable. But the questions is, I think the data simulated is kinda of not right.

library(forecast)

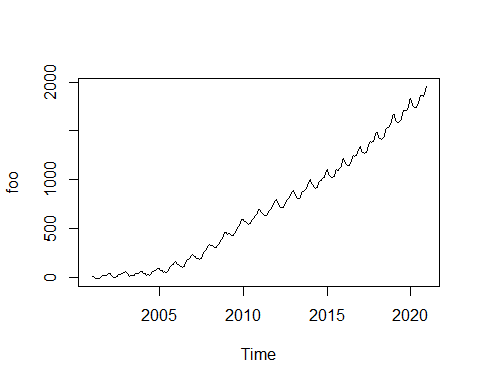
## Registered S3 methods overwritten by 'ggplot2':  
## method from   
## [.quosures rlang  
## c.quosures rlang  
## print.quosures rlang

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

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

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

set.seed(1)  
model <- Arima(ts(rnorm(24000),freq=12), order=c(0,1,1), seasonal=c(0,1,1),fixed=c(theta=0.5, Theta=0.5))  
foo <- simulate(model,nsim = 240)  
plot(foo,type="l")



fit <- Arima(foo, order=c(0,1,1), seasonal=c(0,1,1))  
summary(fit)

## Series: foo   
## ARIMA(0,1,1)(0,1,1)[12]   
##   
## Coefficients:  
## ma1 sma1  
## 0.4743 0.5398  
## s.e. 0.0592 0.0519  
##   
## sigma^2 estimated as 15.79: log likelihood=-636.48  
## AIC=1278.96 AICc=1279.07 BIC=1289.23  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.2033584 3.847513 3.002382 0.661369 4.463736 0.03141502  
## ACF1  
## Training set -0.02291549

# II Reproduction

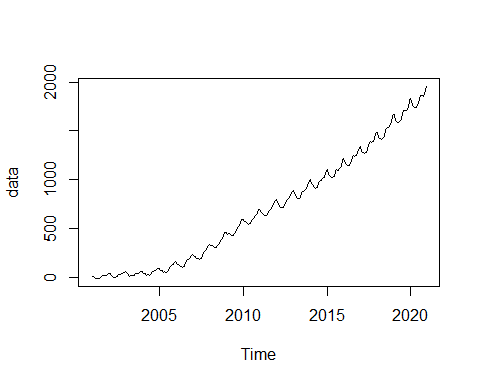
The main reference is the [document](https://cran.r-project.org/web/packages/seasonal/vignettes/seas.pdf) about *X-13ARIMA-SEATS(X-13)*. Something to clearify: By default, a call to **seas** also invokes the following automatic procedures of X-13:

* Transformation selection (log / no log);
* Detection of trading day and Easter effects;
* Outlier detection;
* ARIMA model search.

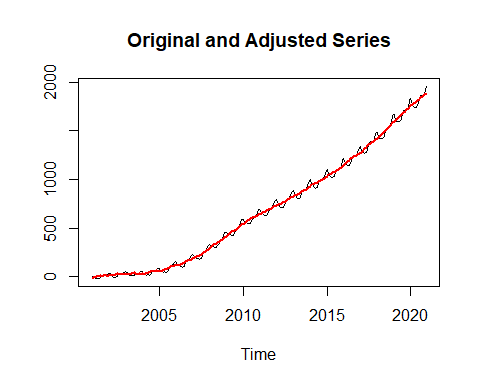
By default, seas calls the *SEATS* adjustment procedure(which decomposes the ARIMA model).To perform the alternative *X-11* adjustment procedure, we need to add **x11 = " "**.

But when I tried to use these code on the simulated data, the curves I got from them are too smooth and looks same. I am thinking: maybe the data I simulated before is not appropriate. In addition, something wrong with the SEATS, cause the model from it(*SARIMA(0,1,1)(0,1,0)[12]*) is different from that of x-11, which is close to our true model *SARIMA(0,1,1)(0,1,1)[12]*

library(seasonal)  
library(forecast)  
set.seed(1)  
model <- Arima(ts(rnorm(24000),freq=12), order=c(0,1,1), seasonal=c(0,1,1),fixed=c(theta=0.5, Theta=0.5))  
data <- simulate(model,nsim=240)  
plot(data)



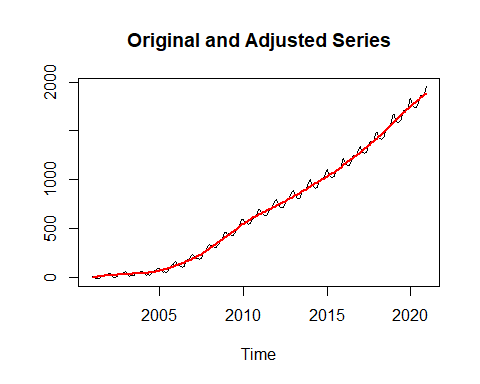
m\_x11 <- seas(data, x11 = "", regression.aictest = NULL)  
plot(m\_x11)



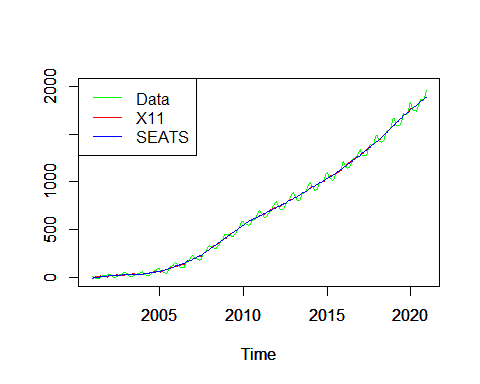
m\_seats <- seas(data, regression.aictest = NULL)

## Model used in SEATS is different: (0 1 1)(0 1 0)

plot(m\_seats)



plot(data,col="green",ylim=c(-10,2000),ylab="")  
par(new=T)  
plot(final(m\_x11),col="red",ylim=c(-10,2000),ylab="")  
par(new=T)  
plot(final(m\_seats),col="blue",ylim=c(-10,2000),ylab="")  
legend("topleft",c("Data","X11","SEATS"),col=c("green","red","blue"),lty=c(1,1,1))



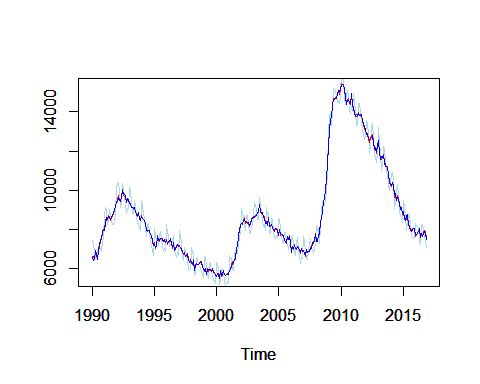
I tried the same code on the data set **unemp**, whose results seem to be good, at least not very smooth and not totally same:

library(seasonal)  
library(seasonalview)

##   
## Attaching package: 'seasonalview'

## The following object is masked from 'package:seasonal':  
##   
## view

eg\_seats <- seas(unemp)  
eg\_x11 <- seas(unemp, x11 = "")  
plot(unemp,col="lightblue",ylim=c(5500,15300),ylab="")  
par(new=T)  
plot(final(eg\_seats),col="red",ylim=c(5500,15300),ylab="")  
par(new=T)  
plot(final(eg\_x11),col="blue",ylim=c(5500,15300),ylab="")



And I am still working on the state space model.

**Update 5.30** 以下是Aaron的回复，包含了几条不错的建议：

* The discuss about ‘spickness’ is not enough, to verify *SEAS* can decompose our seasonal/trend component, we can try: i) remove the S/T from our data, which can be achieved only when data is simulated, cause we know the specific model for each component; ii) detrend/deseason our data by *seas*, and compare two series to see whether *seas* works well(yes if both seem similar)
* the seasonal component we used before is kinda of easy or regular, maybe we can try some more complicated case? like kinds of holidays.
* In reality, the noise is always not gaussian, so maybe we can create some our own noise. To be specific, use the residuals of one data set, like **unemp**. The residual is just the true value minus the prediction of some reasonable model. And we can build a ‘noise’ library, this may be helpful in future.