lab3 Q1

Sue Yang, Michelle Kim, Legg Yeung

August 2, 2017

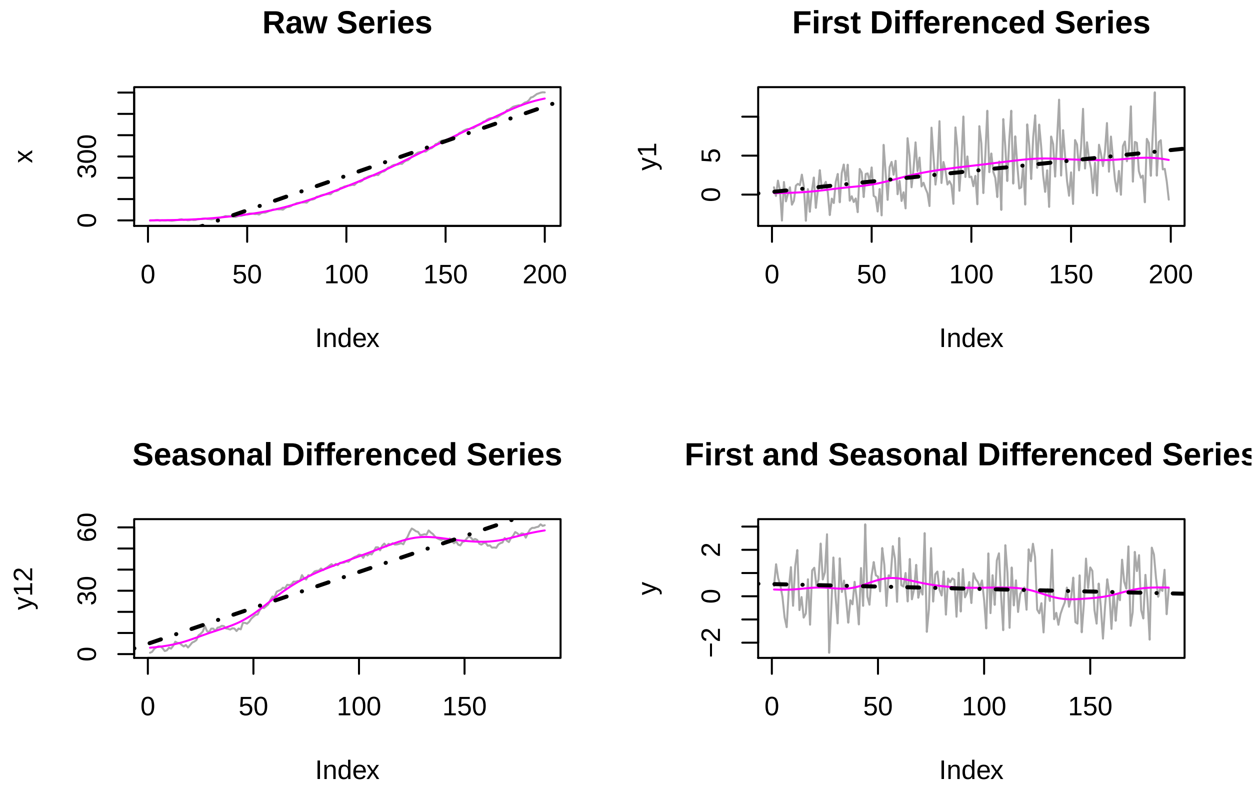
To remove linear trend it is necessary to difference once with lag 1. Because in a purely linear trend, differencing by the last observation singles out the roughly equivalent increments which are stationary at the mean. To remove seasonal patterns, it is necessary to difference once with lag 12 for annual patterns, or lag 4 for quarterly patterns. Because in a purely seasonal series where each observation is same as the one from the last season, differencing by the observation from last season removes seasonal fluctuations and stablize the series at the mean. Real world economic time series, like ours, often exhibit both linear trend in additional to annual seasonal behavior. Therefore differencing by lag 1 and lag 12 will help stablize the raw series at the mean. In the following we simulate a raw series that is a random walk with drift(linear trend) and strong dependence on its seasonal lag 1, then demonstrate how first and seasonal differencing can stablize it at the mean. The model is :

$$(1 - B)(1 - B^{12})X_t = 0.5 + W_t$$

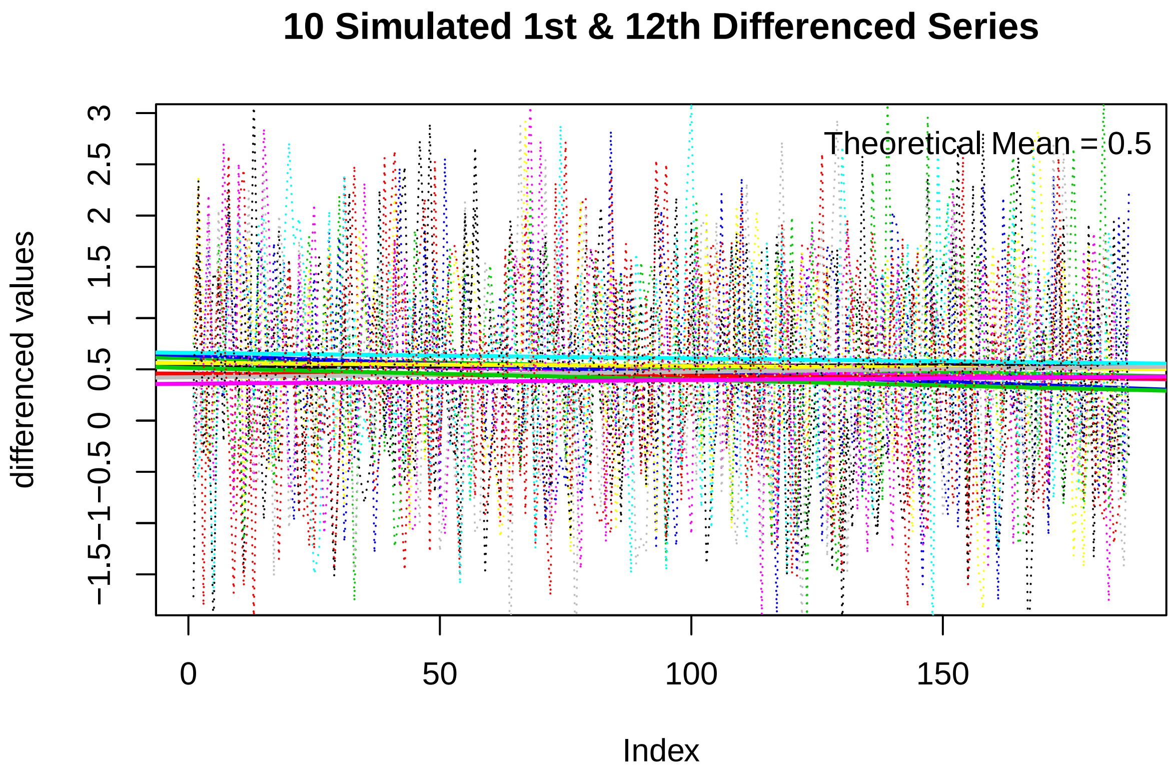
$$X_t = 0.5 + X_{t-12} - X_{t-13} + X_{t-1} + w_t$$

```r
set.seed(30)
x<-w<-rnorm(200)

#x_t = 0.2 + 0.8*x_{t-12}-0.8*x_{t-13}+x_{i-1}+w_t
for(i in 14:200) x[i]<- 0.5 + x[i-12]- x[i-13]+x[i-1]+w[i]


#difference only once
y1<-diff(x)

#difference only seasonally
y12<-diff(x,12)

#difference lag 1 and seasonally
y<-diff(diff(x),12)

# Kernel smoothing
x.k.smooth.widest = ksmooth(time(x),
                             x, kernel = c("normal"),
                             bandwidth = 25)
y1.k.smooth.widest = ksmooth(time(y1),
                              y1, kernel = c("normal"),
                              bandwidth = 25)
y12.k.smooth.widest = ksmooth(time(y12),
                               y12, kernel = c("normal"),
                               bandwidth = 25)
y.k.smooth.widest = ksmooth(time(y),
                             y, kernel = c("normal"),
                             bandwidth = 25)
```

```r
# Make plots
par(mfrow = c(2,2))

plot(x, type="l", col = "darkgray")
title("Raw Series")
lines(x.k.smooth.widest$x, x.k.smooth.widest$y, col = "magenta")
abline(lm(x~time(x)), lty = "dotdash", col = "black", lwd = 2)

plot(y1, type="l", col = "darkgray")
title("First Differenced Series")
lines(y1.k.smooth.widest$x, y1.k.smooth.widest$y, col = "magenta")
abline(lm(y1~time(y1)), lty = "dotdash", col = "black", lwd =2)

plot(y12, type="l", col = "darkgray")
title("Seasonal Differenced Series")
lines(y12.k.smooth.widest$x, y12.k.smooth.widest$y, col = "magenta")
abline(lm(y12~time(y12)), lty = "dotdash", col = "black", lwd = 2)

plot(y, col = "darkgray", type = "l")
title("First and Seasonal Differenced Series")
lines(y.k.smooth.widest$x, y.k.smooth.widest$y, col = "magenta")
abline(lm(y~time(y)), lty = "dotdash", col = "black", lwd =2)
```



```r
y = list()
for (j in 1:10){
  set.seed(j)
  x<-w<-rnorm(200)
  for(i in 14:200) x[i]<- 0.5 + x[i-12]- x[i-13]+x[i-1]+w[i]
  y[[j]]<-diff(diff(x),12)
}

#par(mfrow = c(2,2))

plot(y[[1]], col = rgb(0,0,0, alpha = 0), ylab = "differenced values",
     main = "10 Simulated 1st & 12th Differenced Series")
axis(side = 2, at = seq(-2,3,0.5))
for (j in 1:10){
lines(y[[j]], col = j, type = "l", lwd = 1, lty = "dotted")
abline(lm(y[[j]]~time(y[[1]])),
       col = j*3, lwd = 2)
}
legend("topright",legend = "Theoretical Mean = 0.5", bty = "n")
```


10 Simulated 1st & 12th Differenced Series
Theoretical Mean = 0.5
differenced values
Index