npc_MMM

Brittany

12/17/2018

load npc data

```
dnpc <- read.csv("../New Policy Count.csv", sep = "\t")
dnpc <- dnpc[,1:3] #get rid of cols
names(dnpc) <- c("t", "date", "y") #name cols
dnpc <- subset(dnpc, !is.na(y)) #get rid of nans (missing data for most recent months)
dnpc$y <-  dnpc$y/1e3 #make smaller
dnpc$date <- as.Date(dnpc$date, '%m/%d/%Y')
yts <- ts(dnpc$y, start=c(2011,1), frequency=12) #turn into time series class
dygraph(yts)
```

fit AAA model w/ ets

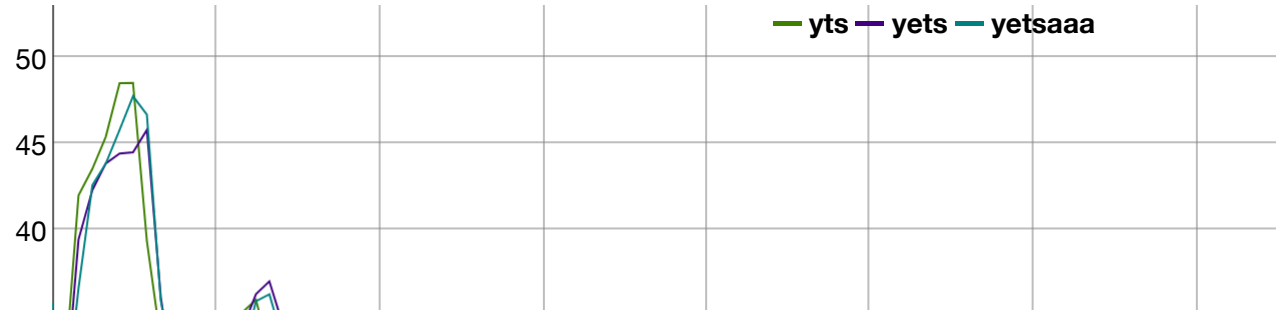
```
etsfit <- ets(yts,model='MMM')
yets <- fitted.values(etsfit)
etsfit
```

```
## ETS(M,M,M)
##
## Call:
##  ets(y = yts, model = "MMM")
##
##   Smoothing parameters:
##     alpha = 0.5656
##     beta  = 0.0543
##     gamma = 2e-04
##
##   Initial states:
##     l = 41.814
##     b = 0.9774
##     s = 0.8885 0.8336 0.8955 0.983 1.1265 1.1399
##            1.1799 1.1651 1.119 1.0578 0.8328 0.7785
##
##   sigma:  0.0718
##
##      AIC     AICc      BIC
## 435.8877 444.2713 478.5723
```

```
yetsaaa <- fitted.values(ets(yts,model='AAA'))

#ets initialization
l0ets <- 41.814
b0ets <- 0.9774
s0ets <- c(0.8885, 0.8336, 0.8955, 0.983, 1.1265, 1.1399, 1.1799, 1.1651, 1.119, 1.0578,
 0.8328, 0.7785)
#ets optimized params
alphaets = 0.5656
betaets  = 0.0543
gammaets = 1e-4
sigets <- 0.0718
```

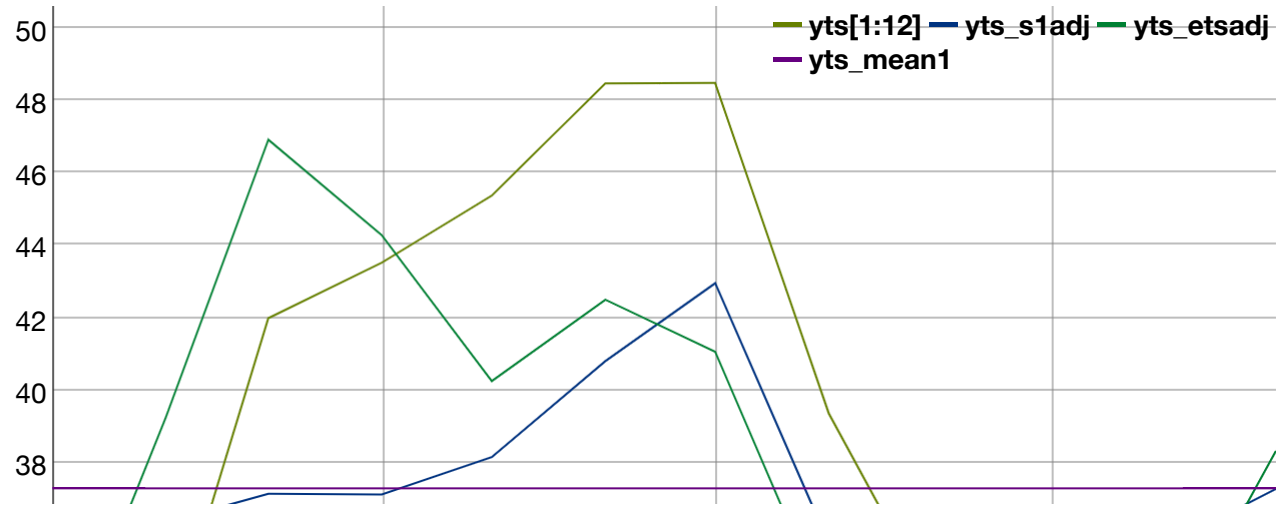
plot npc time series and AAA/AAN model

```
dygraph(cbind(yts,yets,yetsaaa))
```



get initial params

```
m <- 12
N <- nrow(dnpc)
Ns <-  as.integer(N/m+1)

#get yearly avg
savg <- vector()
for (j in 1:Ns) {
  savgsum <- 0.
  scount <- 0
  for (k in (m*(j-1)+1):(m*j)) {
    if (k<=N){
      savgsum <- savgsum + yts[k]
      scount <- scount + 1
    }
  }
  savg[j] <- savgsum/scount
}

#initial seasonality
#get avg ratio from average for each period
s0 <- vector()
for (i in 1:m) {
  value <-  0
  vcount <-  0
  for (j in 1:Ns) {
    if ((m*(j-1)+i)<=N) {
      value <- value + yts[m*(j-1)+i]/savg[j]
      vcount <- vcount + 1
    }
  }
  s0[i] <-  value/vcount
}
s0 <- s0/(sum(s0)/12)

yts_s1adj <-  ts(yts[1:12]/s0, start=c(2011,1), frequency=12)
yts_etsadj <- ts(yts[1:12]/s0ets, start=c(2011,1), frequency=12)
yts_mean1 <- ts(rep(mean(yts[1:12]),times=12), start=c(2011,1), frequency=12)

dygraph(cbind(yts[1:12],yts_s1adj,yts_etsadj,yts_mean1)) #should this be flatter?
```




```
#get l0 and b0 from linear regression of first 10 data points
x <- 1:10
initreg <- lm(yts_s1adj[x]~x)
l0 <- summary(initreg)$coefficients['(Intercept)','Estimate']
b0 <- 1+summary(initreg)$coefficients['x','Estimate']/summary(initreg)$coefficients['(In
tercept)','Estimate']
```

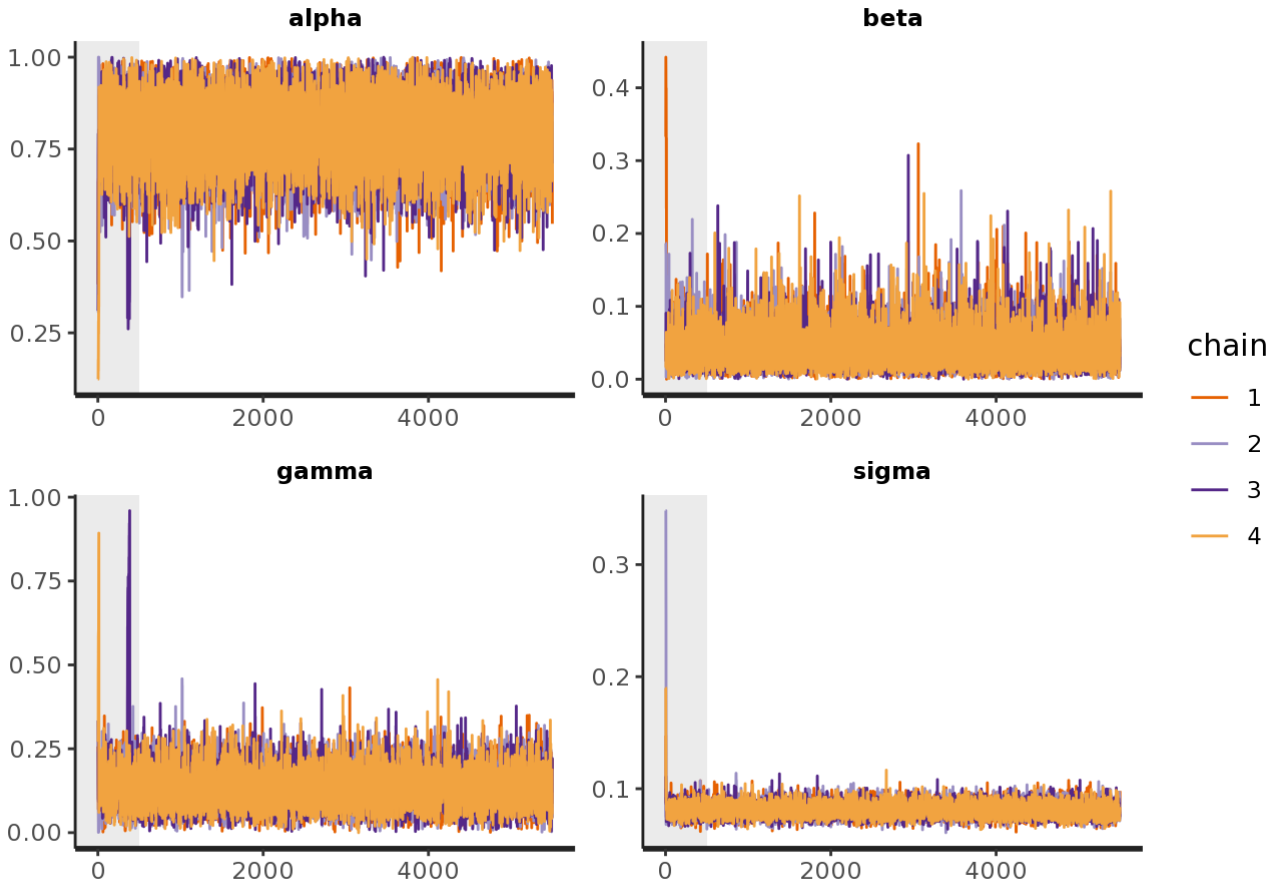
hopefully fit MMM model w/ stan

```
# fit
standata <- list(y = yts, N = N, m = m, Ns = Ns, l0 = l0, b0 = b0, s0 = s0)
# standata_etsinit <- list(y=yts,N=N,m=m,Ns=Ns,l0=l0ets,b0=b0ets,s0=s0ets)
standata_init <- list(y = yts, N = N, m = m, Ns = Ns, init_prior = c(l0, b0,
    s0), prior_width = 0.01)
standata_etsinit <- list(y = yts, N = N, m = m, Ns = Ns, l0 = l0ets, b0 = b0ets,
    s0 = s0ets, init_prior = c(l0, b0, s0), prior_width = 0.01)

rstanfit <- stan(file = "npc_MMM_old.stan", data = standata, warmup = 500, iter = 5500)
 #, control = list(adapt_delta = 0.99))
rstanfit_init <- stan(file = "npc_MMM.stan", data = standata_init, warmup = 500,
    iter = 5000)  #,control=list(adapt_delta=0.99))
# rstanfit_etsinit <-
# stan(file='npc_MMM.stan',data=standata_etsinit,warmup=100,iter=500,control=list(adapt_
delta=0.99))
```

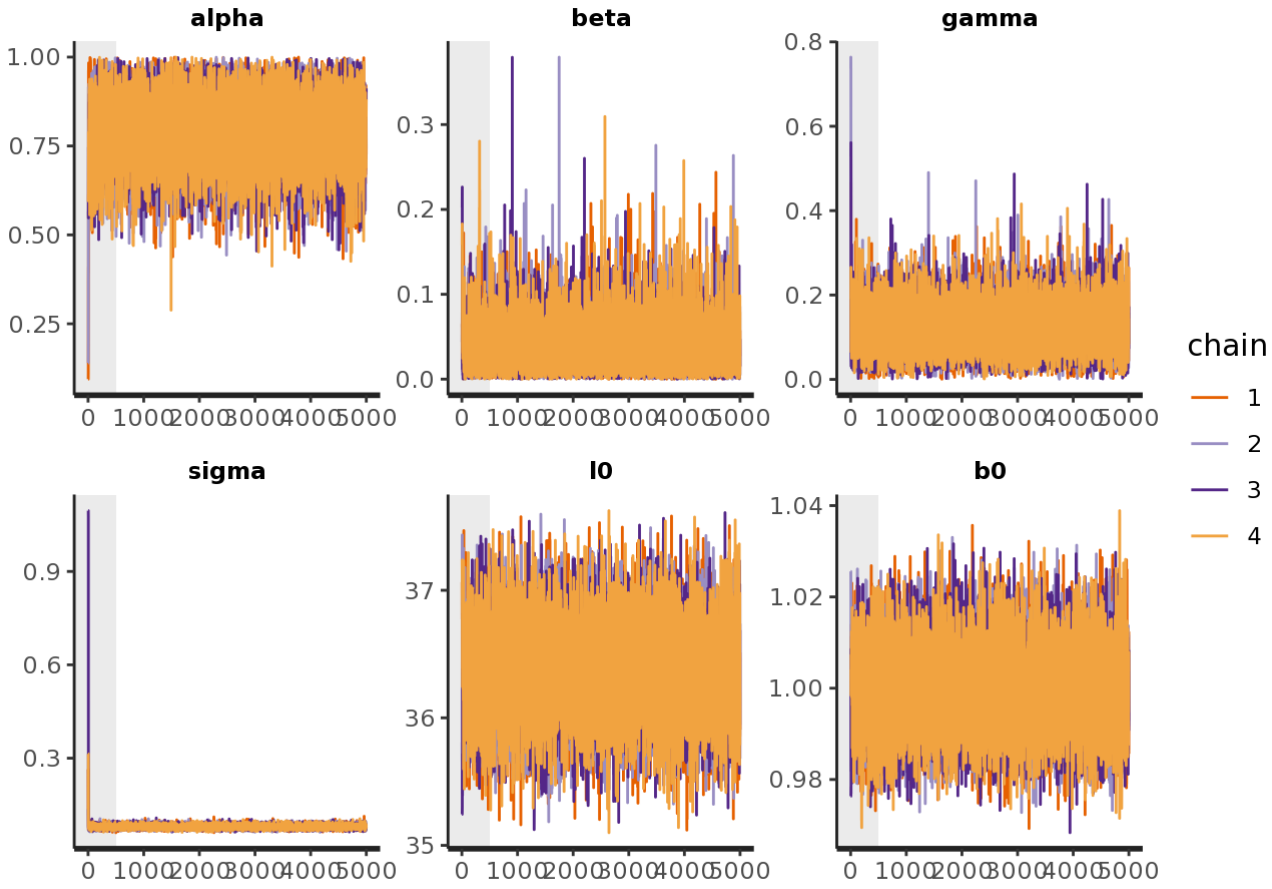
look at fit params/convergence

look at traces

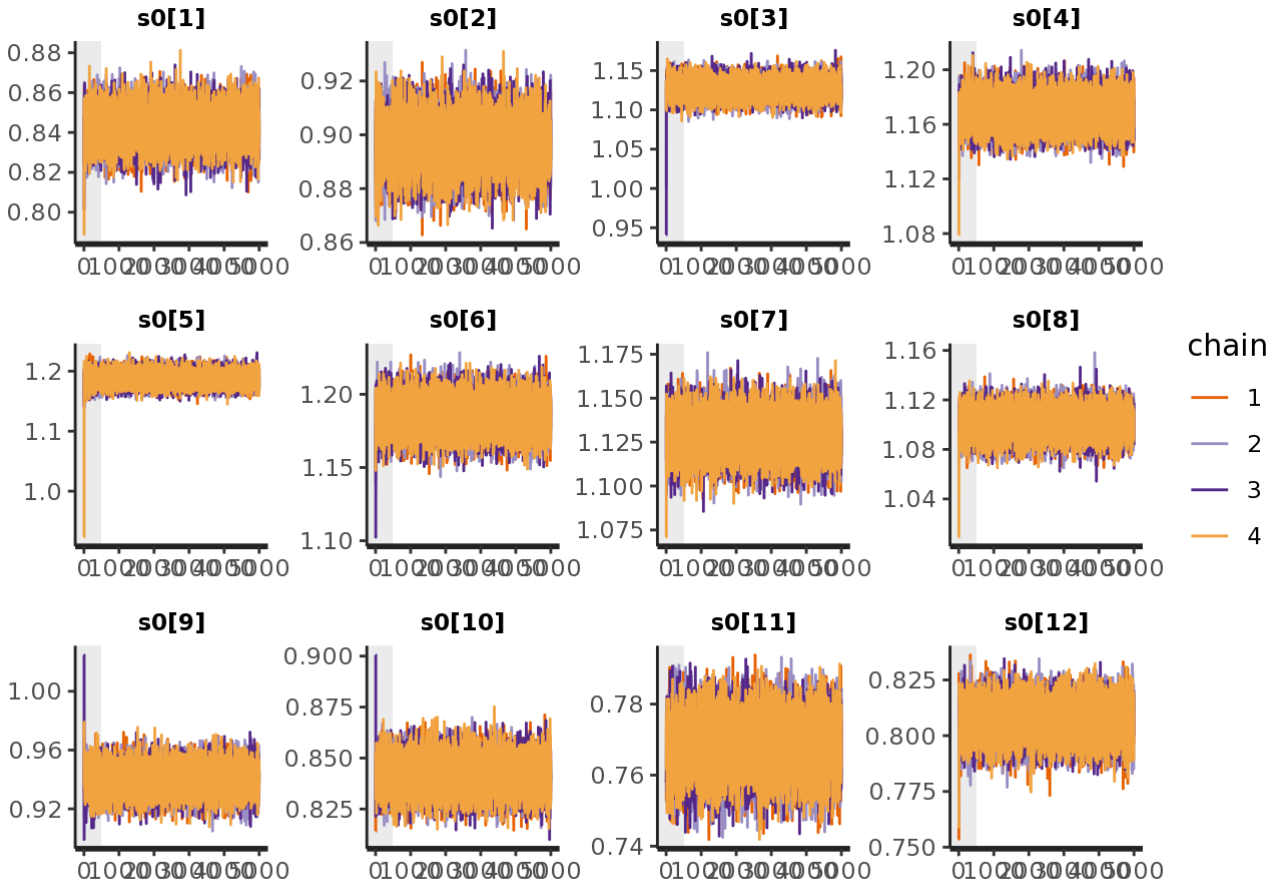
```
plot(rstanfit, plotfun='trace', pars = c('alpha','beta','gamma','sigma'), inc_warmup=TRU
E)
```




```
plot(rstanfit_init, plotfun='trace', pars = c('alpha','beta','gamma','sigma','l0','b0'),
 inc_warmup=TRUE)
```



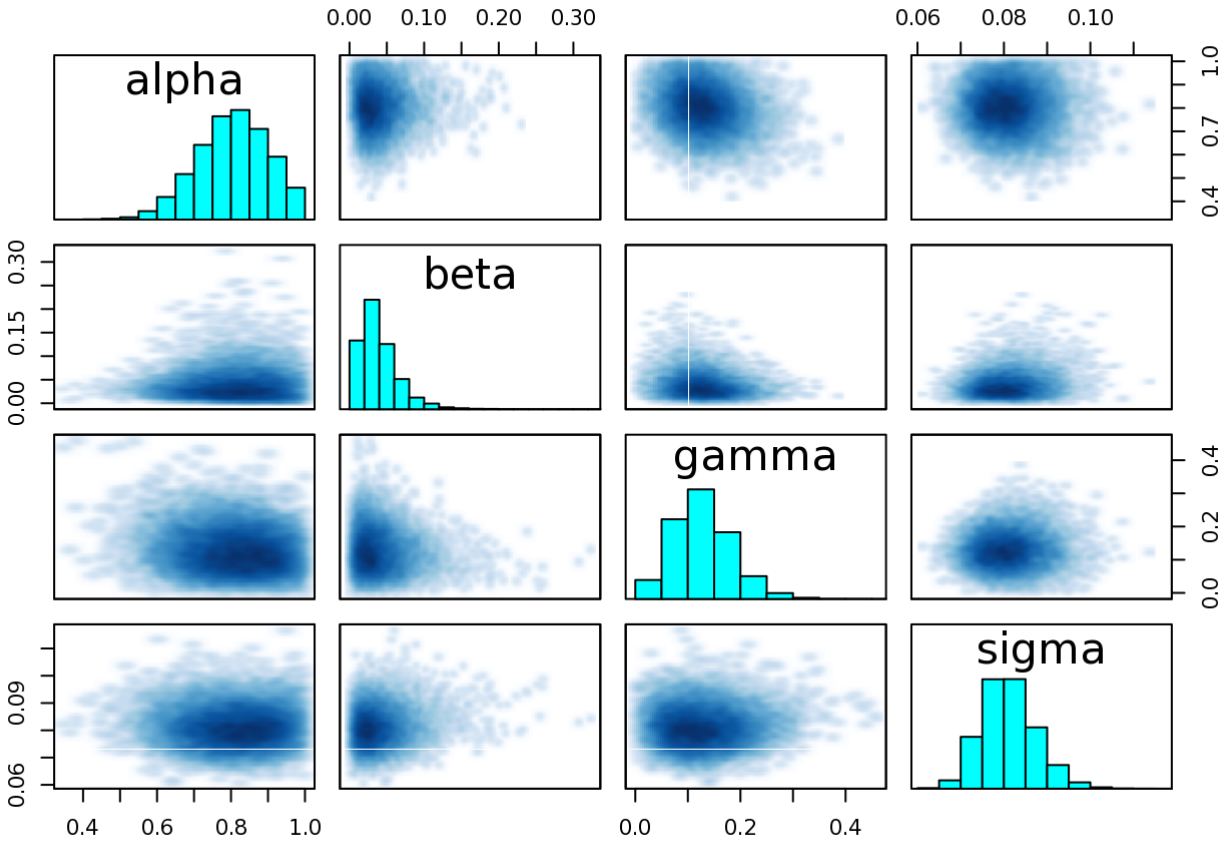

```
plot(rstanfit_init,plotfun='trace',pars=c('s0'),inc_warmup=TRUE)
```



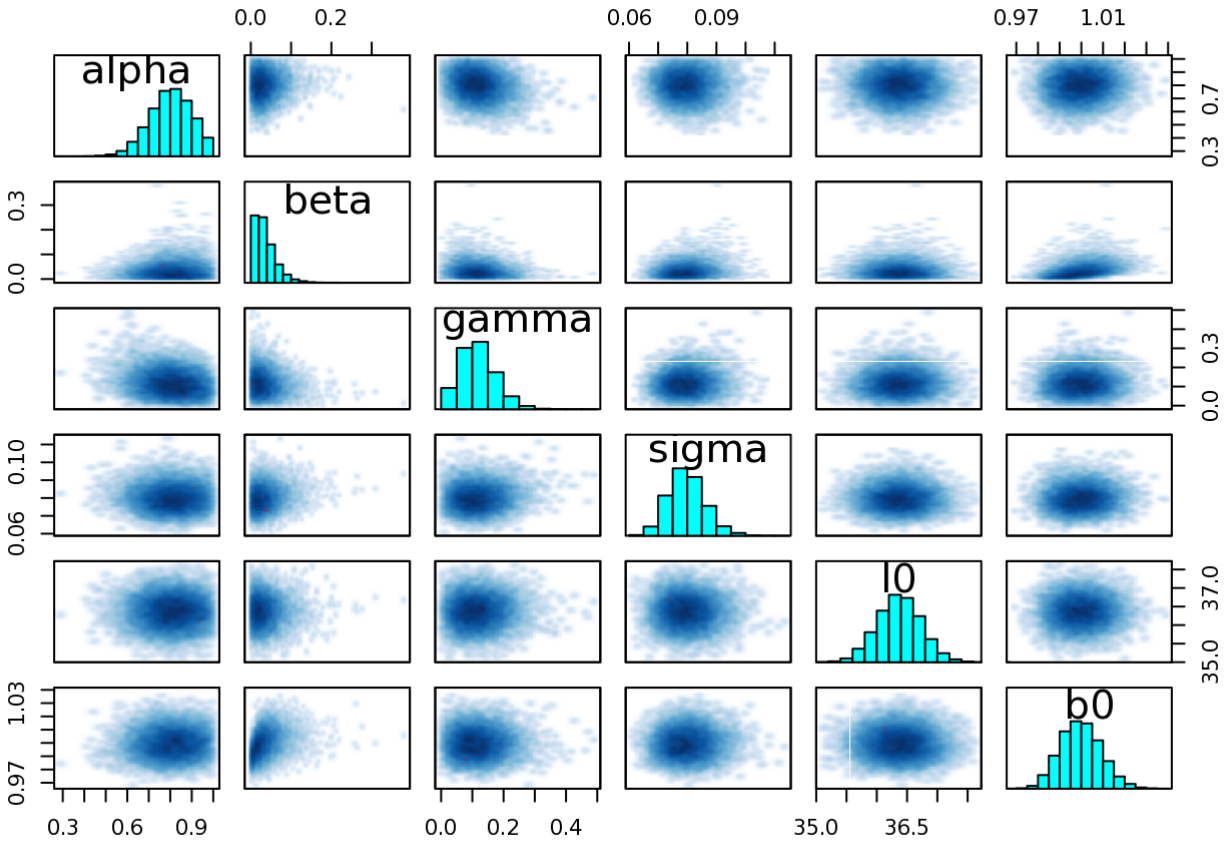

```
#plot(rstanfit_etsinit, plotfun='trace', pars = c('alpha','beta','gamma','sigma','l0','b0'), inc_warmup=TRUE)
```

look at parameter contours

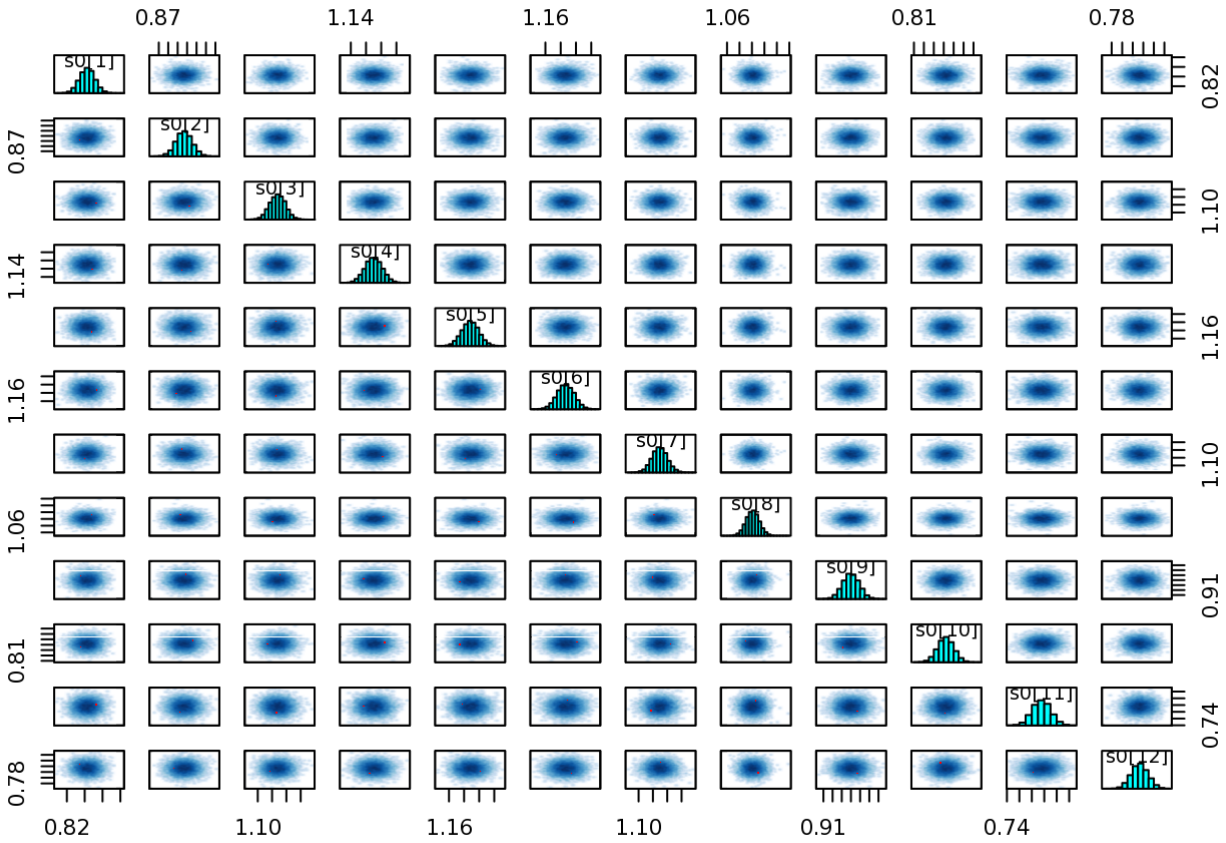
```
pairs(rstanfit, pars=c('alpha','beta','gamma','sigma'))
```




```
#pairs(rstanfit_etsinit, pars=c('alpha','beta','gamma','sigma'))
pairs(rstanfit_init, pars=c('alpha','beta','gamma','sigma','l0','b0'))
```




```
pairs(rstanfit_init,pars=c('s0'))
```




```
stansum <- summary(rstanfit)$summary
stansum_init <- summary(rstanfit_init)$summary
#View(stansum_init)
#stan mean params
alpha <- stansum['alpha','mean']
beta <- stansum['beta','mean']
gamma <- stansum['gamma','mean']
sigma <- stansum['sigma','mean']


#stansum
#sapply(get_sampler_params(stanfit,inc_warmup=0), function(x) mean(x[,'accept_stat__']))
```

compare parameters from stan and ets

```
print(c(alpha,alphaets))
```

```
## [1] 0.8041491 0.5656000
```

```
print(c(beta,betaets))
```

```
## [1] 0.03948572 0.05430000
```

```
print(c(gamma,gammaets))
```

```
## [1] 0.1280678 0.0001000
```

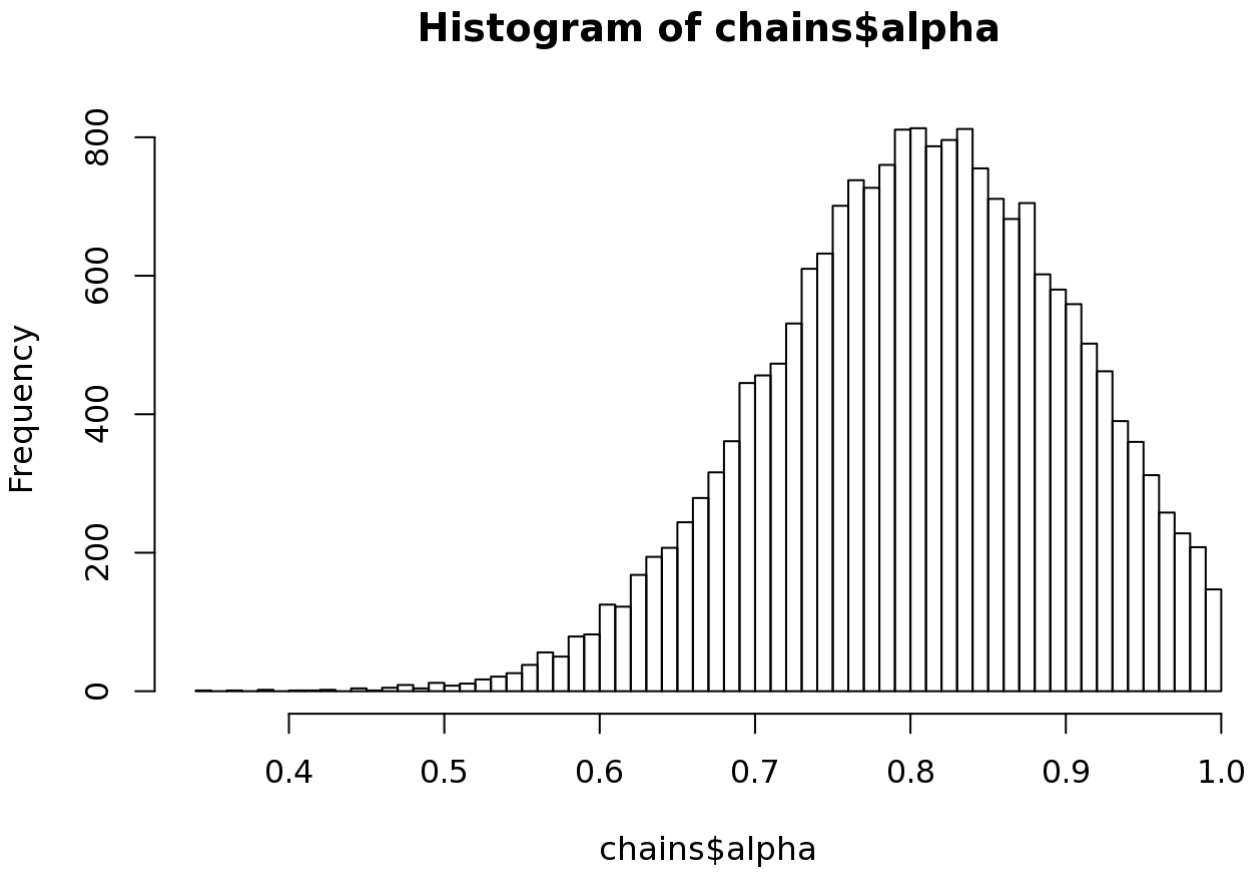
```
print(c(sigma,sigets))
```

```
## [1] 0.08108778 0.07180000
```

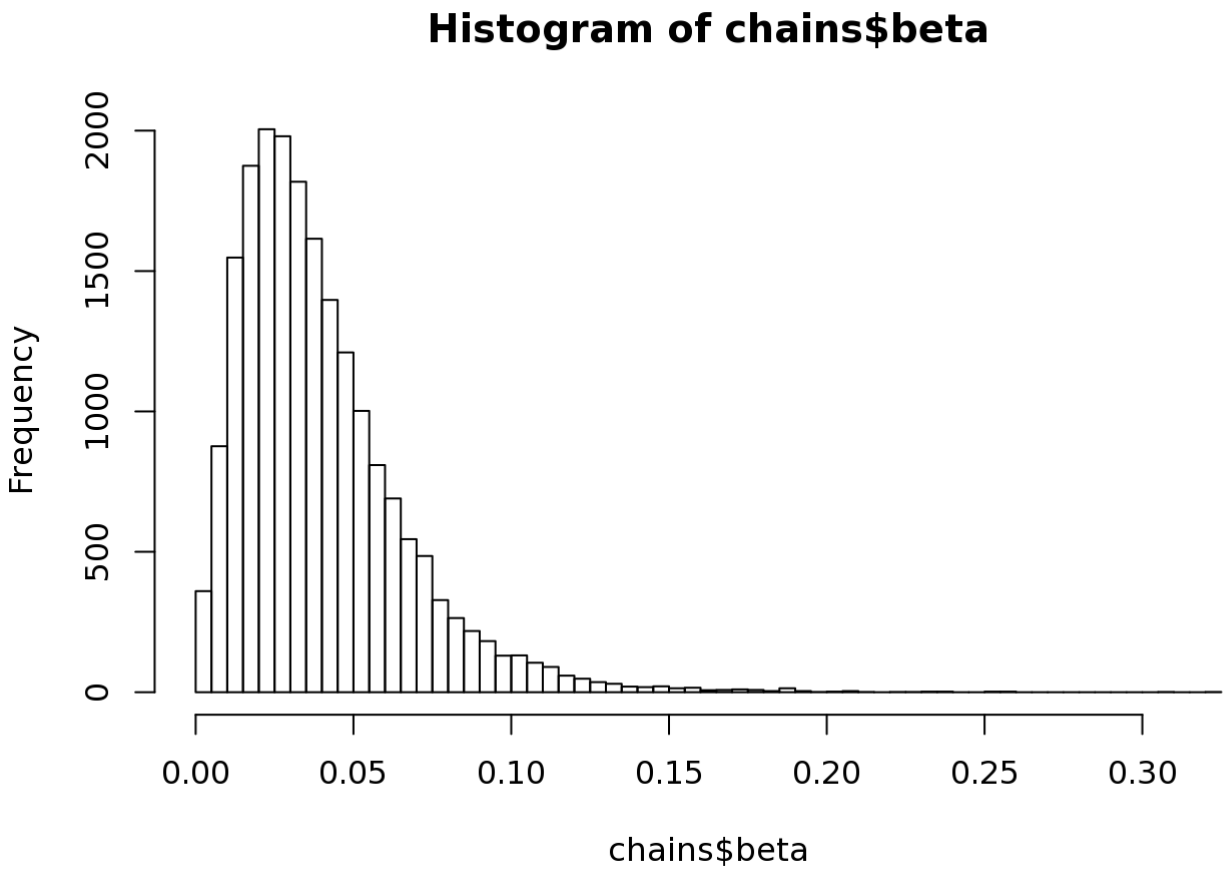
```
#alpha and sigma are similar, not betastar and gamma
```

get a better idea of what the params should be

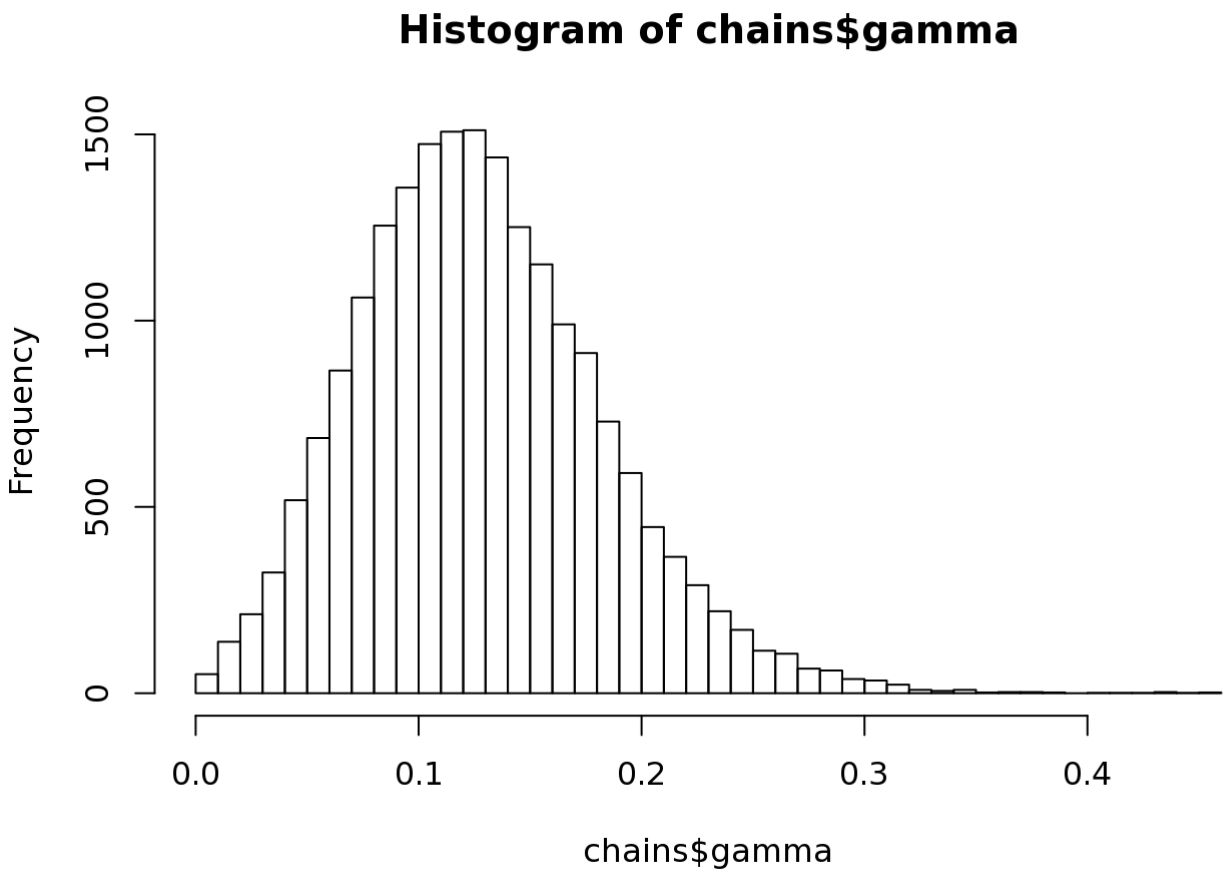
```
chains <- extract(rstanfit)
chains_init <- extract(rstanfit_init)
alphahist <- hist(chains$alpha,breaks=50) #ets gives alpha~1, this is close, a little le
ss than 1
```



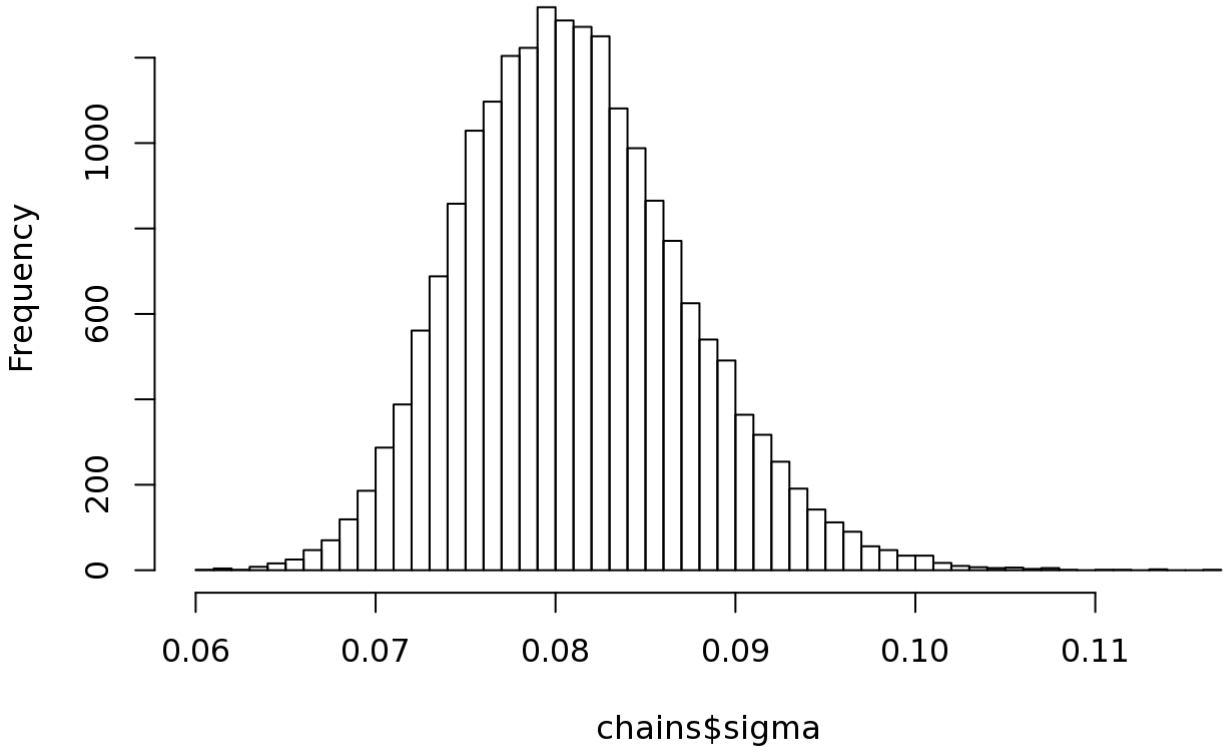
```
betahist <- hist(chains$beta,breaks=75) #ets gives betastar~0.1, this is pretty diff wit
h betastar~0.85
```




```
gammahist <- hist(chains$gamma,breaks=50) #ets gives gamma~1e-4, this is same
```




```
sigmahist <- hist(chains$sigma,breaks=50) #ets gives sigma=0.07, max here is ~3.6, prett
y off
```




```
print(alphahist$breaks[48:49],alphahist$breaks[-4:-3])
```

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

```
print(betahist$breaks[39:40])
```

```
## [1] 0.190 0.195
```

```
print(gammahist$breaks[1:2])
```

```
## [1] 0.00 0.01
```

```
print(sigmahist$breaks[17:18])
```

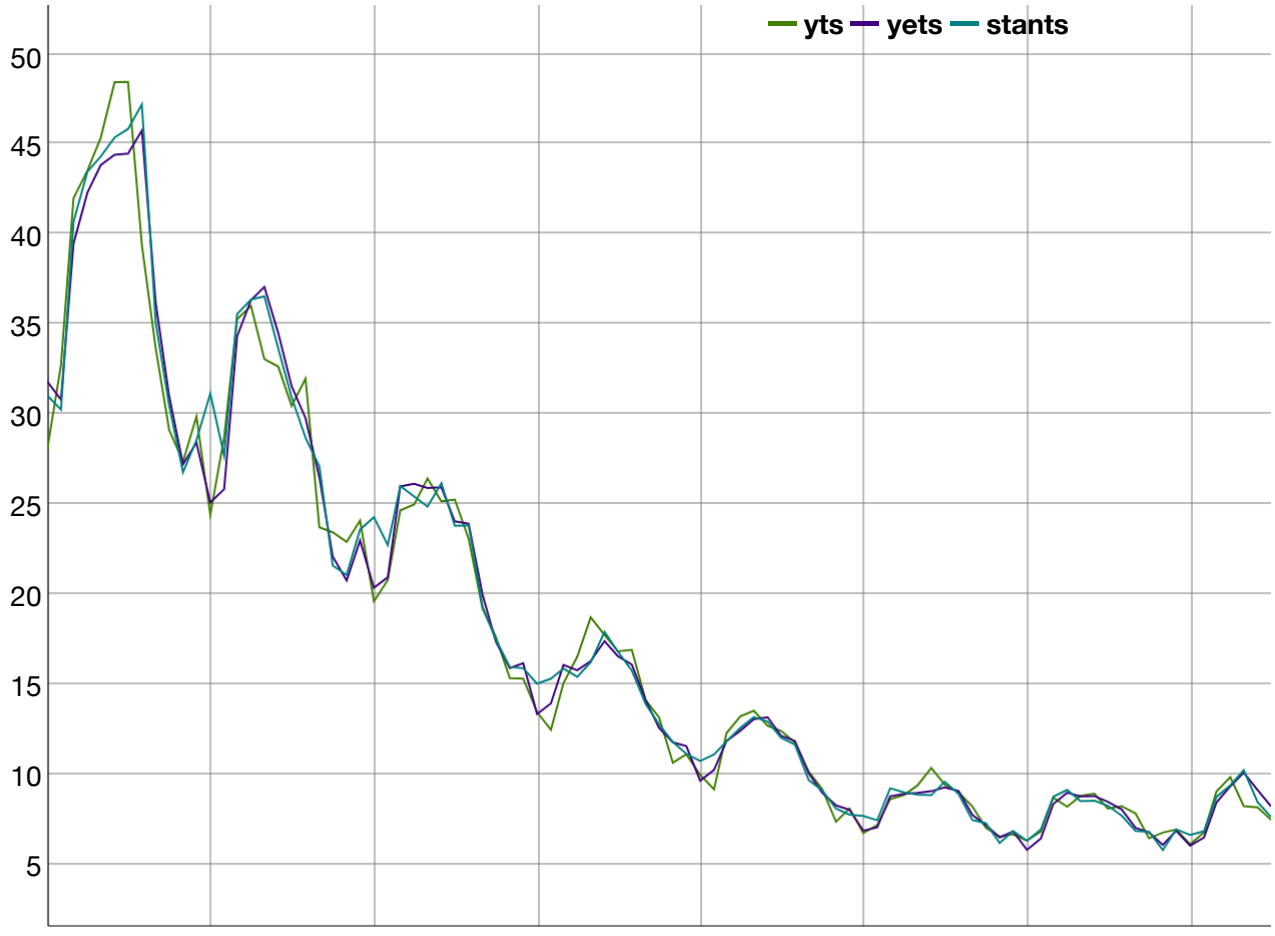
```
## [1] 0.076 0.077
```

plot best fitting model

```
ystan <- vector()

ystan[1] <- l0*b0*s0[1]

for (i in 2:12) {
  lprev <- stansum[sprintf('l[%i]',i-1),'mean']
  bprev <- stansum[sprintf('b[%i]',i-1),'mean']
  sprev <- s0[i]
  ystan[i] <- lprev*bprev*sprev
}

for (i in 13:N) {
  lprev <- stansum[sprintf('l[%i]',i-1),'mean']
  bprev <- stansum[sprintf('b[%i]',i-1),'mean']
  sprev <- stansum[sprintf('s[%i]',i-m),'mean']
  ystan[i] <- lprev*bprev*sprev
}
```

plot npc time series, and ets and stan models

```
stants <- ts(ystan, start=c(2011,1), frequency=12)

dygraph(cbind(yts,yets,stants))
```

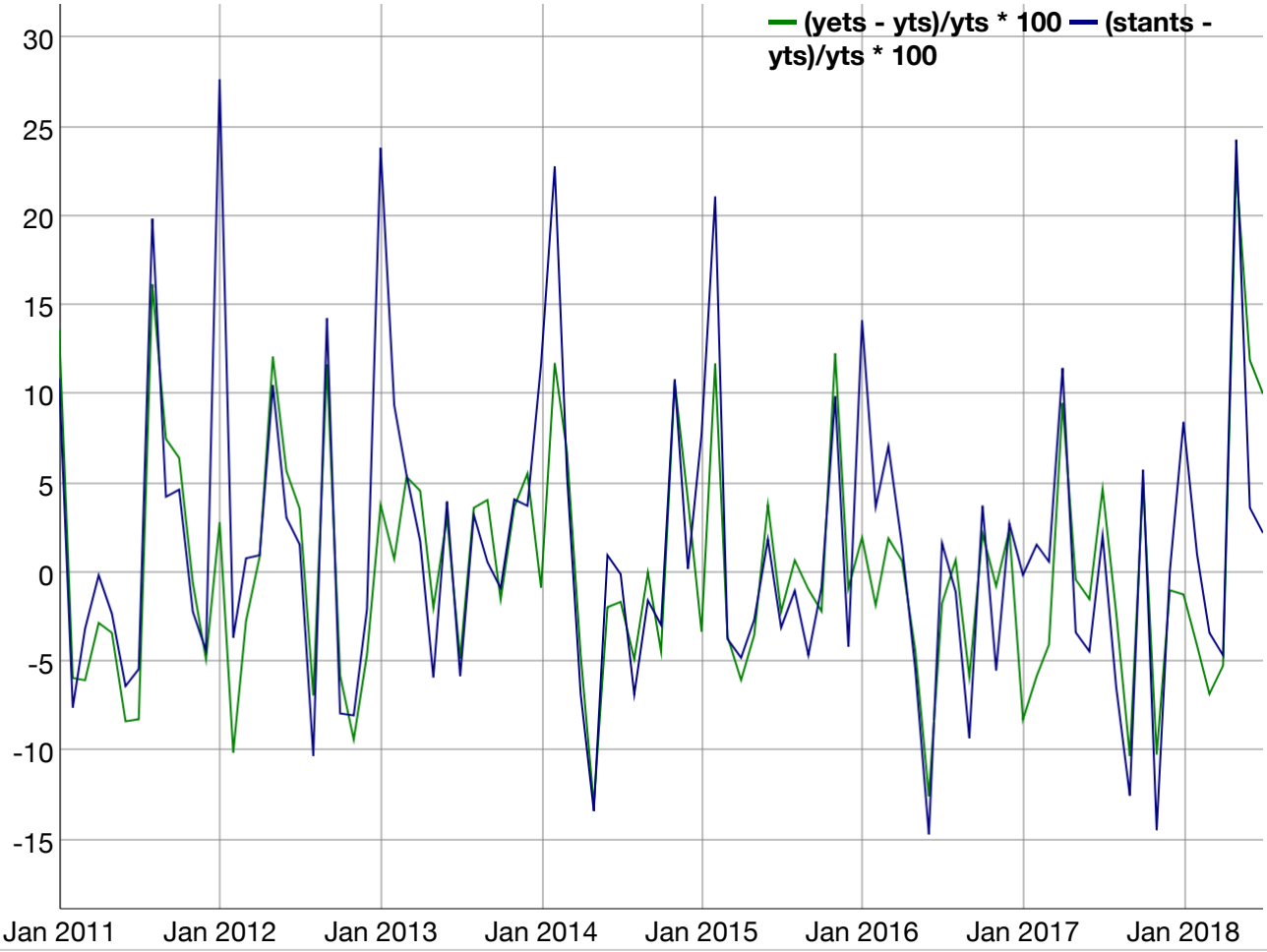


Jan 2011 Jan 2012 Jan 2013 Jan 2014 Jan 2015 Jan 2016 Jan 2017 Jan 2018

```
#looks very similar to yets
```

look at percent difference between the data and ets/my code

```
dygraph(cbind((yets-yts)/yts*100, (stants-yts)/yts*100))
```




```
#my smoothing isn't too different
```

compare my initialization and ets's

```
print(c(l0,l0ets))
```

```
## [1] 36.37717 41.81400
```

```
print(c(b0,b0ets))
```

```
## [1] 1.004042 0.977400
```

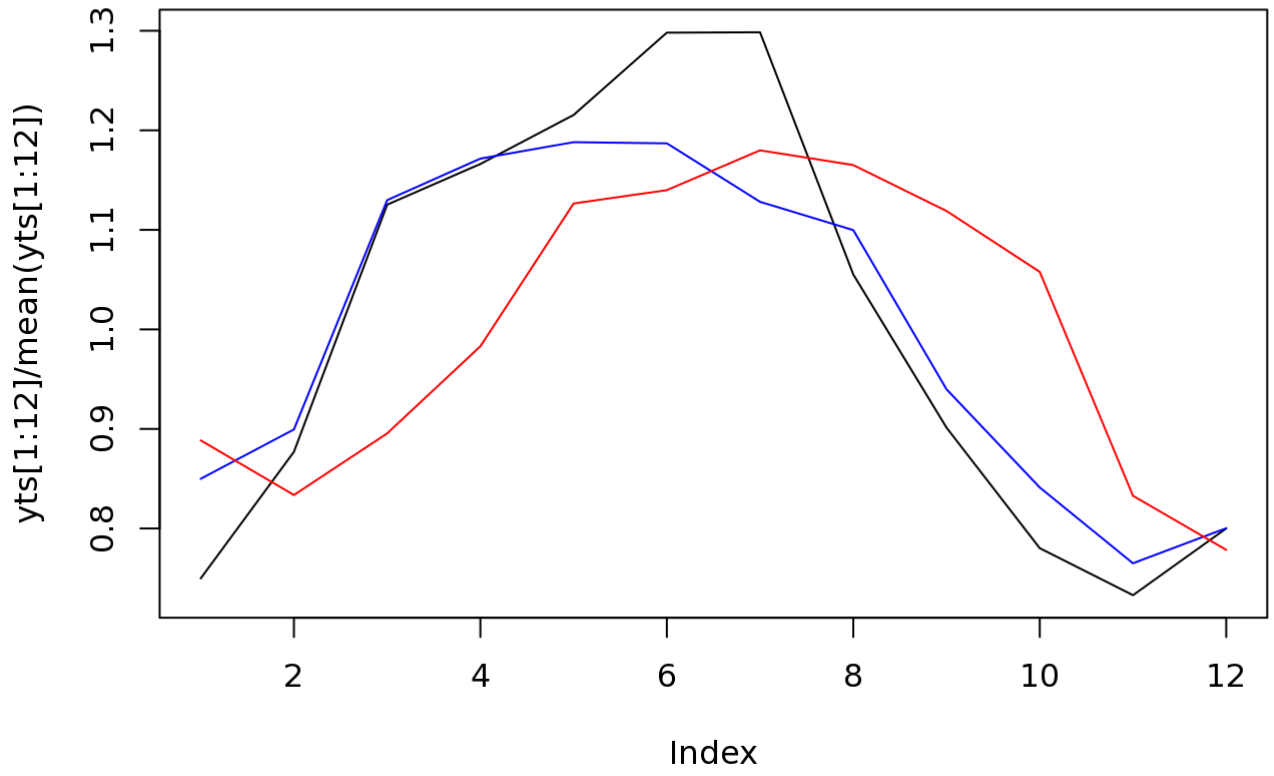
```
print(s0)
```

```
##  [1] 0.8498719 0.8994620 1.1298670 1.1715460 1.1881059 1.1869627 1.1281638
##  [8] 1.0997543 0.9398343 0.8411567 0.7650053 0.8002701
```

```
print(s0ets)
```

```
##  [1] 0.8885 0.8336 0.8955 0.9830 1.1265 1.1399 1.1799 1.1651 1.1190 1.0578
## [11] 0.8328 0.7785
```

```
plot(yts[1:12]/mean(yts[1:12]),type='l')
lines(s0,col='blue')
lines(s0ets,col='red')
```




```
#our initial points are quite different, but their seasonality looks weird, right?
```

make function to smooth data, if components=TRUE, will return components instead of model

```
modelMMM <- function(y,alpha,beta,gamma,l0,b0,s0,components=FALSE) {
  m <- 12
  N <- length(y)
  Ns <-  as.integer(N/m+1)
  betastar <- beta/alpha

  smoothy <- vector()
  l <- vector()
  b <- vector()
  s <- vector()

  smoothy[1] <- l0*b0*s0[1]
  l[1] <-  alpha*(y[1]/s0[1]) + (1-alpha)*(l0*b0)
  b[1] <-  betastar*(l[1]/l0) + (1-betastar)*b0
  s[1] <-  gamma*(y[1]/(l0*b0)) + (1-gamma)*s0[1]

  for (t in 2:12) {
    l[t] <- alpha*(y[t]/s0[t]) + (1-alpha)*(l[t-1]*b[t-1])
    b[t] <- betastar*(l[t]/l[t-1]) + (1-betastar)*b[t-1]
    s[t] <- gamma*(y[t]/(l[t-1]*b[t-1])) + (1-gamma)*s0[t]
    smoothy[t] <- l[t-1]*b[t-1]*s0[t]
  }

  for (t in 13:N) {
    l[t] <- alpha*(y[t]/s[t-m]) + (1-alpha)*(l[t-1]*b[t-1])
    b[t] <- betastar*(l[t]/l[t-1]) + (1-betastar)*b[t-1]
    s[t] <- gamma*(y[t]/(l[t-1]*b[t-1])) + (1-gamma)*s[t-m]
    smoothy[t] <- l[t-1]*b[t-1]*s[t-m]
  }
  if (!components) {return(ts(smoothy,start=c(2011,1),frequency=12))}
  if (components) {return(list(l=ts(l,start=c(2011,1),frequency=12),b=ts(b,start=c(2011,
1),frequency=12),s=ts(s,start=c(2011,1),frequency=12)))}

}

modelMMMv2 <- function(y,alpha,beta,gamma,l0,b0,s0,components=FALSE) {
  m <- 12
  N <- length(y)
  Ns <-  as.integer(N/m+1)
  betastar <- beta/alpha
  gammastar <- gamma/(1-alpha)

  smoothy <- vector()
  l <- vector()
  b <- vector()
  s <- vector()

  smoothy[1] <- l0*b0*s0[1]
  l[1] <-  alpha*(y[1]/s0[1]) + (1-alpha)*(l0*b0)
  b[1] <-  betastar*(l[1]/l0) + (1-betastar)*b0
  s[1] <-  gammastar*(y[1]/l[1]) + (1-gammastar)*s0[1]

  for (t in 2:12) {
```

```
    l[t] <- alpha*(y[t]/s0[t]) + (1-alpha)*(l[t-1]*b[t-1])
    b[t] <- betastar*(l[t]/l[t-1]) + (1-betastar)*b[t-1]
    s[t] <- gammastar*(y[t]/l[t]) + (1-gammastar)*s0[t]
    smoothy[t] <- l[t-1]*b[t-1]*s0[t]
  }

  for (t in 13:N) {
    l[t] <- alpha*(y[t]/s[t-m]) + (1-alpha)*(l[t-1]*b[t-1])
    b[t] <- betastar*(l[t]/l[t-1]) + (1-betastar)*b[t-1]
    s[t] <- gammastar*(y[t]/l[t]) + (1-gammastar)*s[t-m]
    smoothy[t] <- l[t-1]*b[t-1]*s[t-m]
  }
  if (!components) {return(ts(smoothy,start=c(2011,1),frequency=12))}
  if (components) {return(list(l=ts(l,start=c(2011,1),frequency=12),b=ts(b,start=c(2011,
1),frequency=12),s=ts(s,start=c(2011,1),frequency=12)))}

}
```

test out function

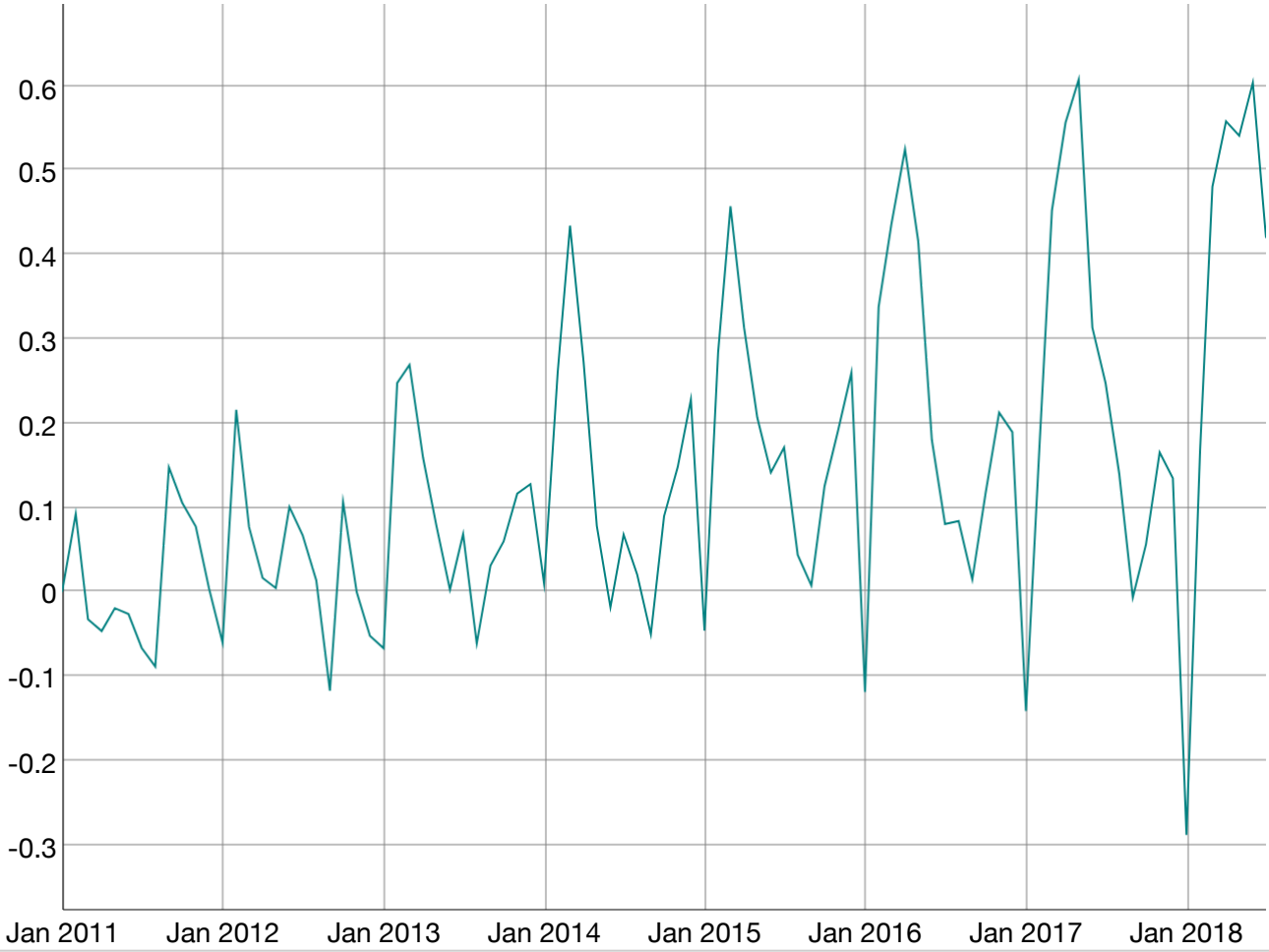
```
stanmmm <- modelMMM(yts,alpha,beta,gamma,l0,b0,s0)
etsmmm <- modelMMM(yts,alphaets,betaets,gammaets,l0ets,b0ets,s0ets)
dygraph(cbind(yts,stants,stanmmm,yets,etsmmm))
```

```
#so i think difference is that what I pull from ets/stan is including best fitting erro
r, while the lines that look like yts are just the smoothed lines?
```

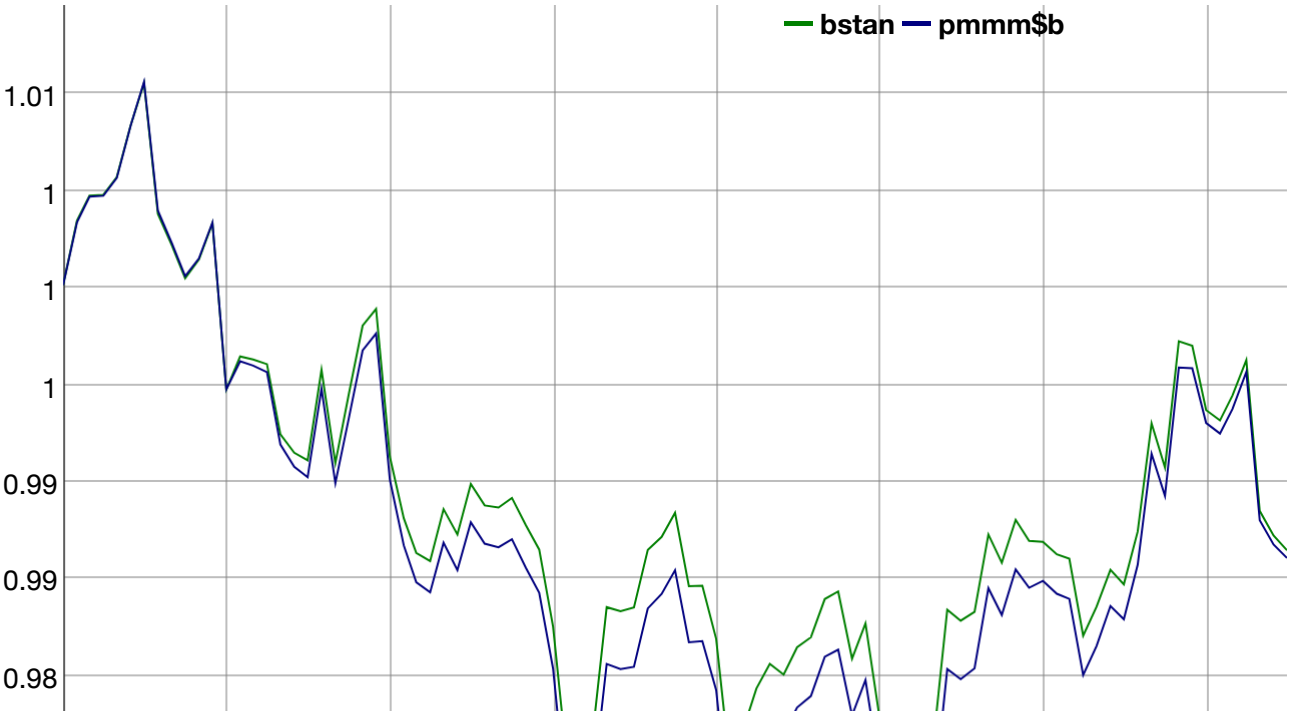
Compare the smoothed data from my function, and from stan's output

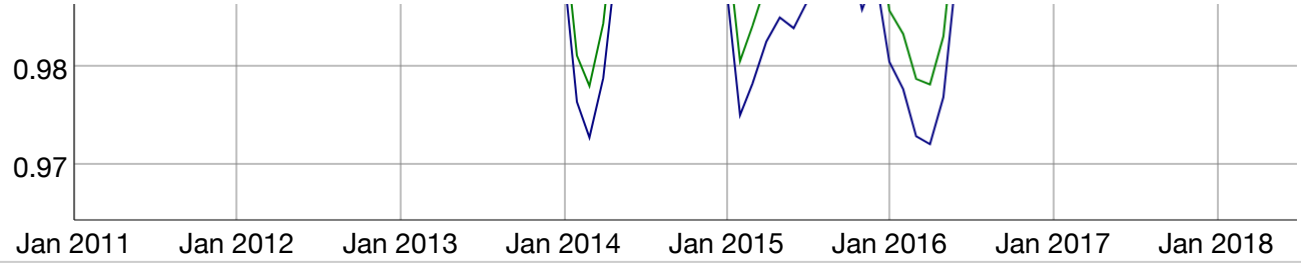
```
pmmm <- modelMMM(yts,alpha,beta,gamma,l0,b0,s0,components=TRUE)

lstan <- vector()
bstan <- vector()
sstan <- vector()

for (t in 1:N) {
  lstan[t] <- stansum[sprintf('l[%i]',t),'mean']
  bstan[t] <- stansum[sprintf('b[%i]',t),'mean']
  sstan[t] <- stansum[sprintf('s[%i]',t),'mean']
}


lstan <- ts(lstan, start=c(2011,1), frequency=12)
bstan <- ts(bstan, start=c(2011,1), frequency=12)
sstan <- ts(sstan, start=c(2011,1), frequency=12)

#they look the same
dygraph(cbind(lstan,pmmm$l))
```


```
dygraph((lstan-pmmm$l)/lstan*100)
```

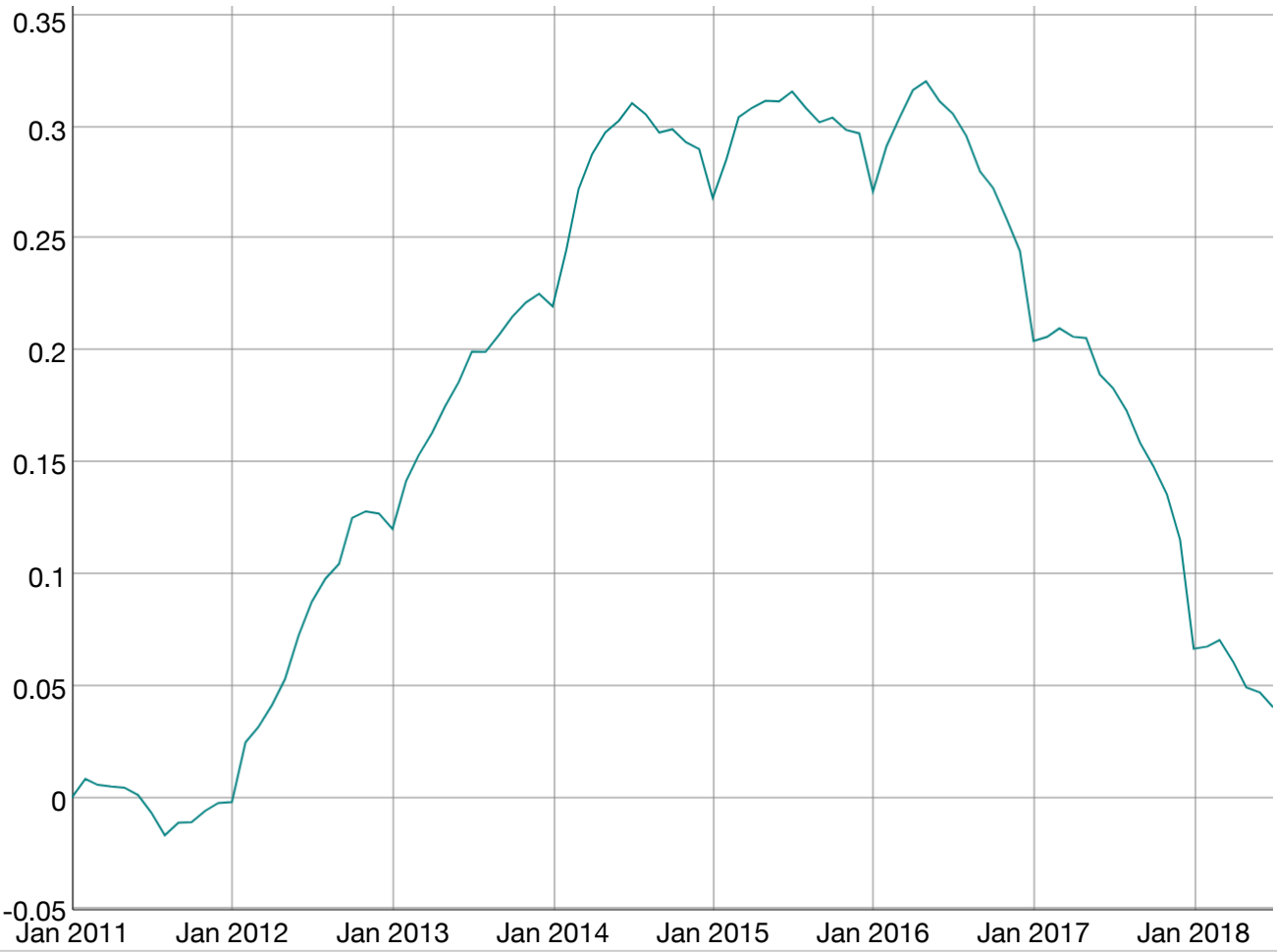



```
#same
dygraph(cbind(bstan,pmmm$b))
```

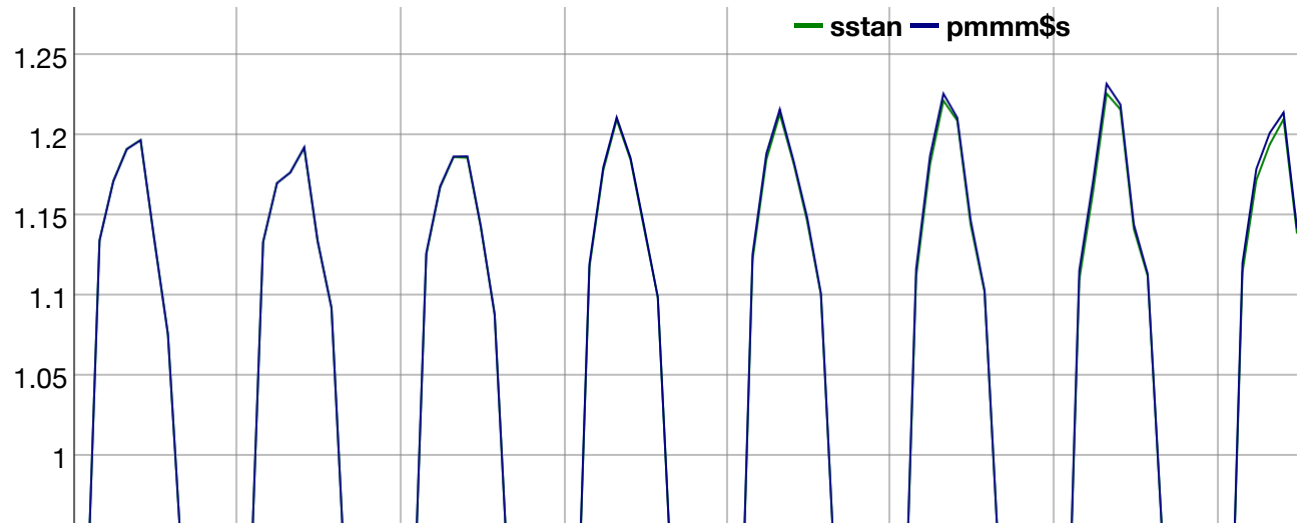


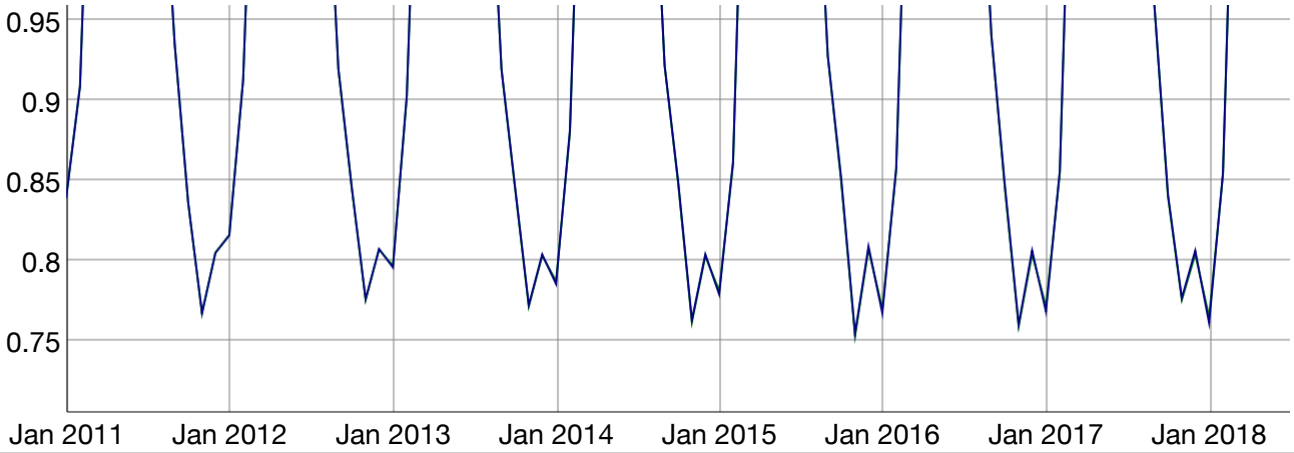



```
dygraph((bstan-pmmm$b)/bstan*100)
```

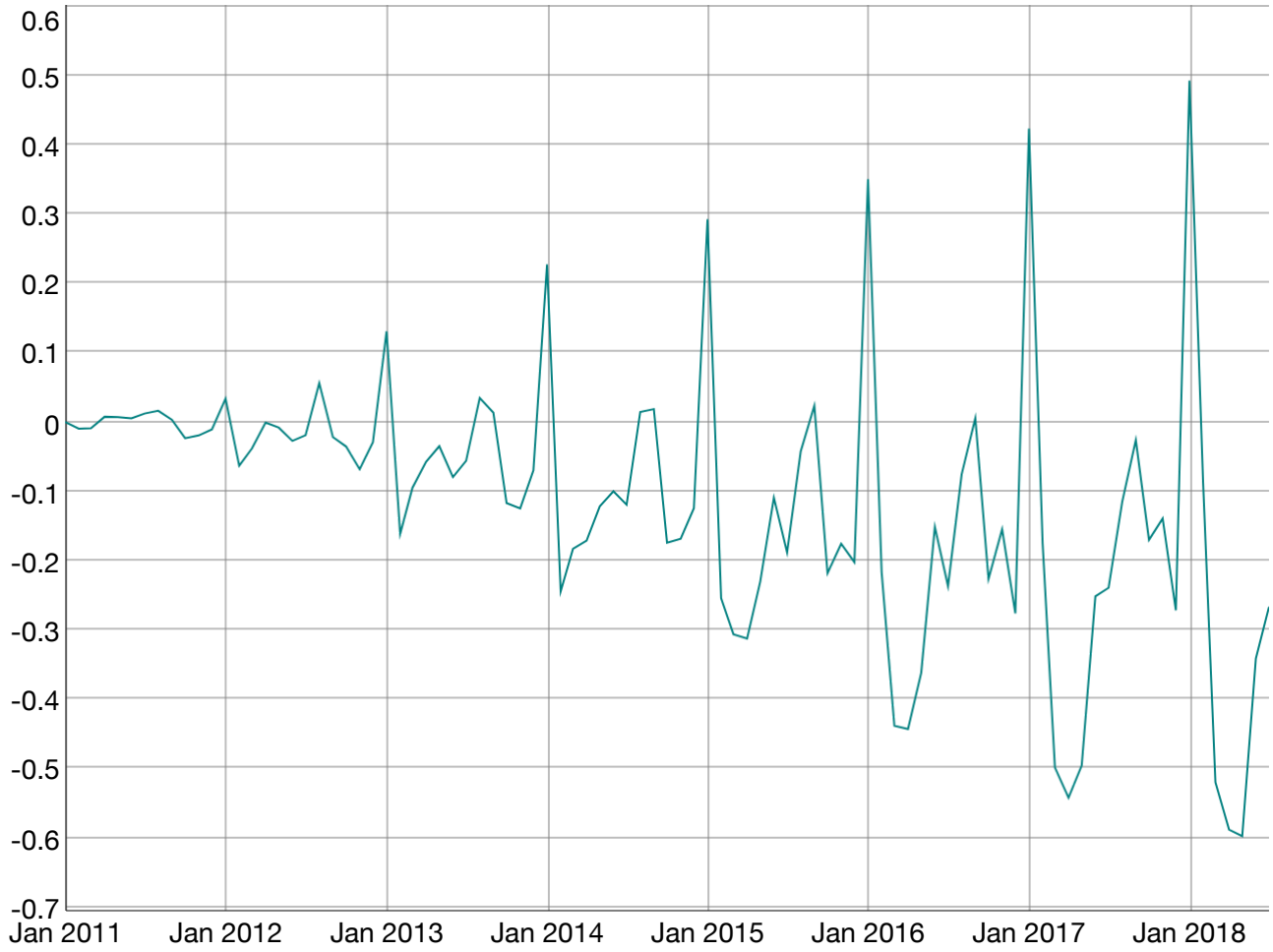



```
#pretty similar
dygraph(cbind(sstan,pmmm$s))
```





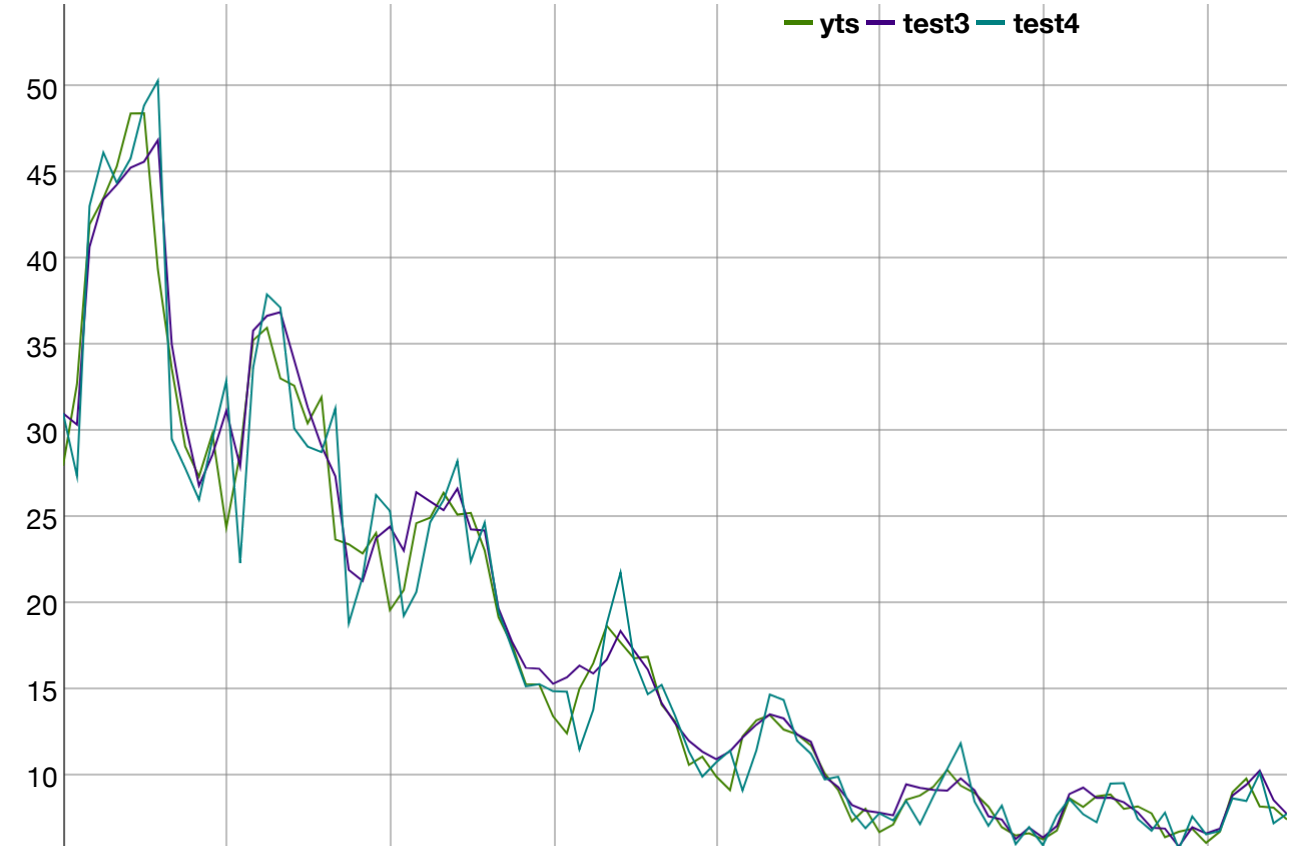

```
dygraph((sstan-pmmm$s)/sstan*100)
```



Check out how varying smoothing parameters changes my coded up model

```
#min/max of alpha
test1 <- modelMMM(yts,0.00001,beta,gamma,l0,b0,s0)
test2 <- modelMMM(yts,1.,beta,gamma,l0,b0,s0)
dygraph(cbind(yts,test1,test2))
```

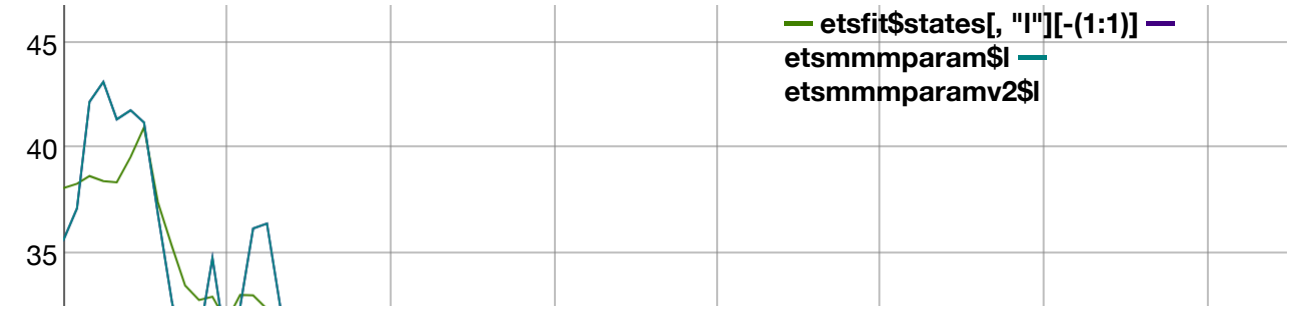

```
#min/max of betastar
test3 <- modelMMM(yts,alpha,0.,gamma,l0,b0,s0)
test4 <- modelMMM(yts,alpha,1.,gamma,l0,b0,s0)
dygraph(cbind(yts,test3,test4))
```

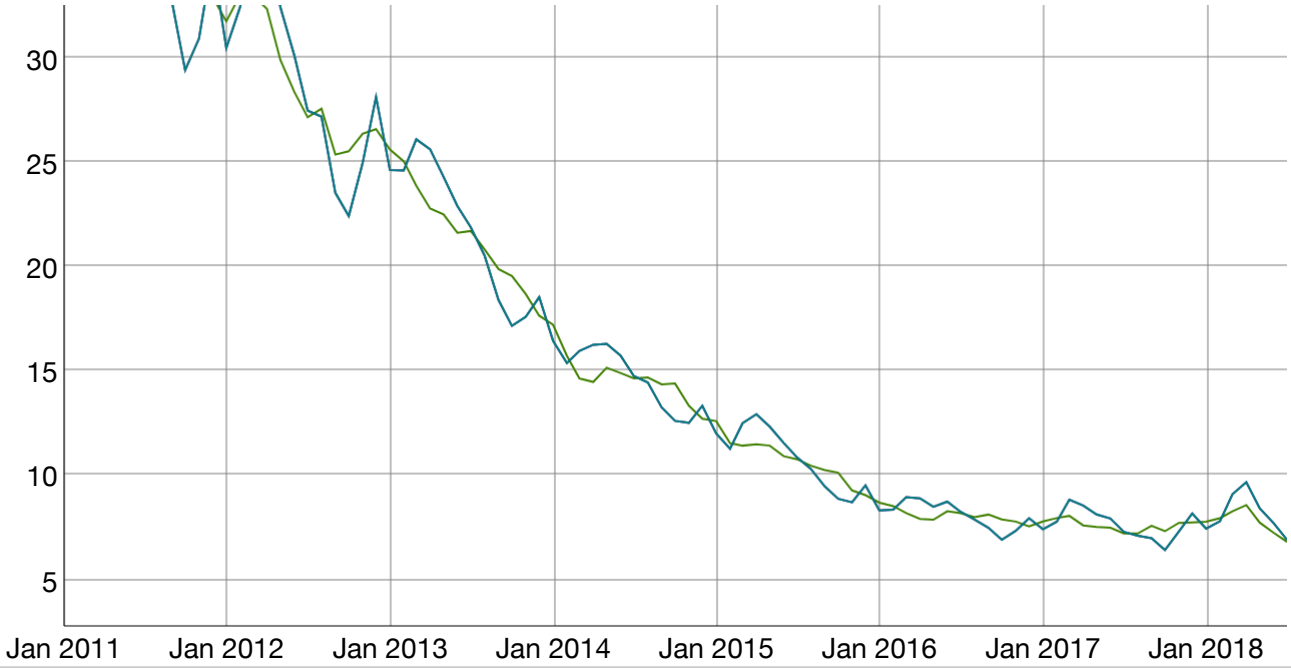



```
#min/max of gamma
test5 <- modelMMM(yts,alpha,beta,0.,l0,b0,s0)
test6 <- modelMMM(yts,alpha,beta,1.,l0,b0,s0)
dygraph(cbind(yts,test5,test6)) %>% dyAxis('y',valueRange=c(0,60))
```

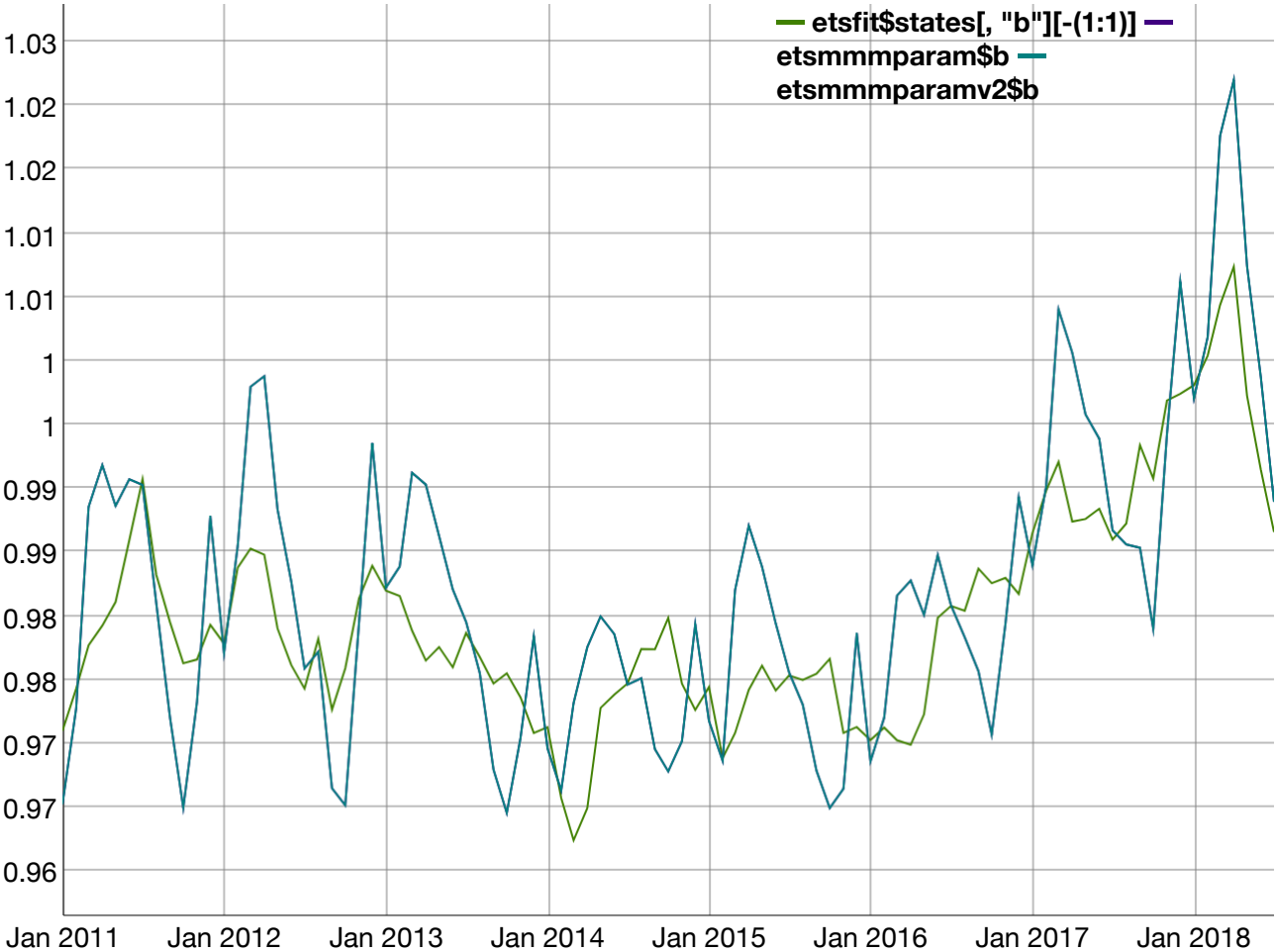
test my calculated states compared to ets

```
etsmmmparam <- modelMMM(yts,alphaets,betaets,gammaets,l0ets,b0ets,s0ets,components=TRUE)
etsmmmparamv2 <- modelMMMv2(yts,alphaets,betaets,gammaets,l0ets,b0ets,s0ets,components=T
RUE)

dygraph(cbind(etsfit$states[,'l'][-(1:1)],etsmmmparam$l,etsmmmparamv2$l))
```

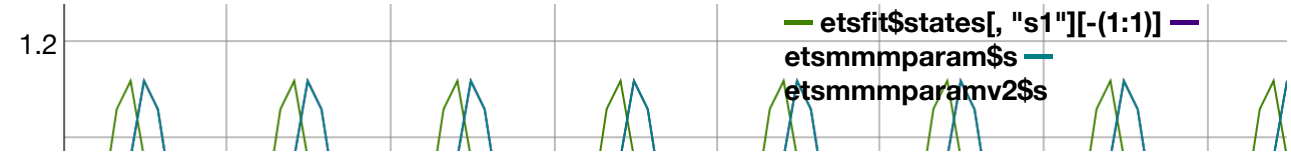


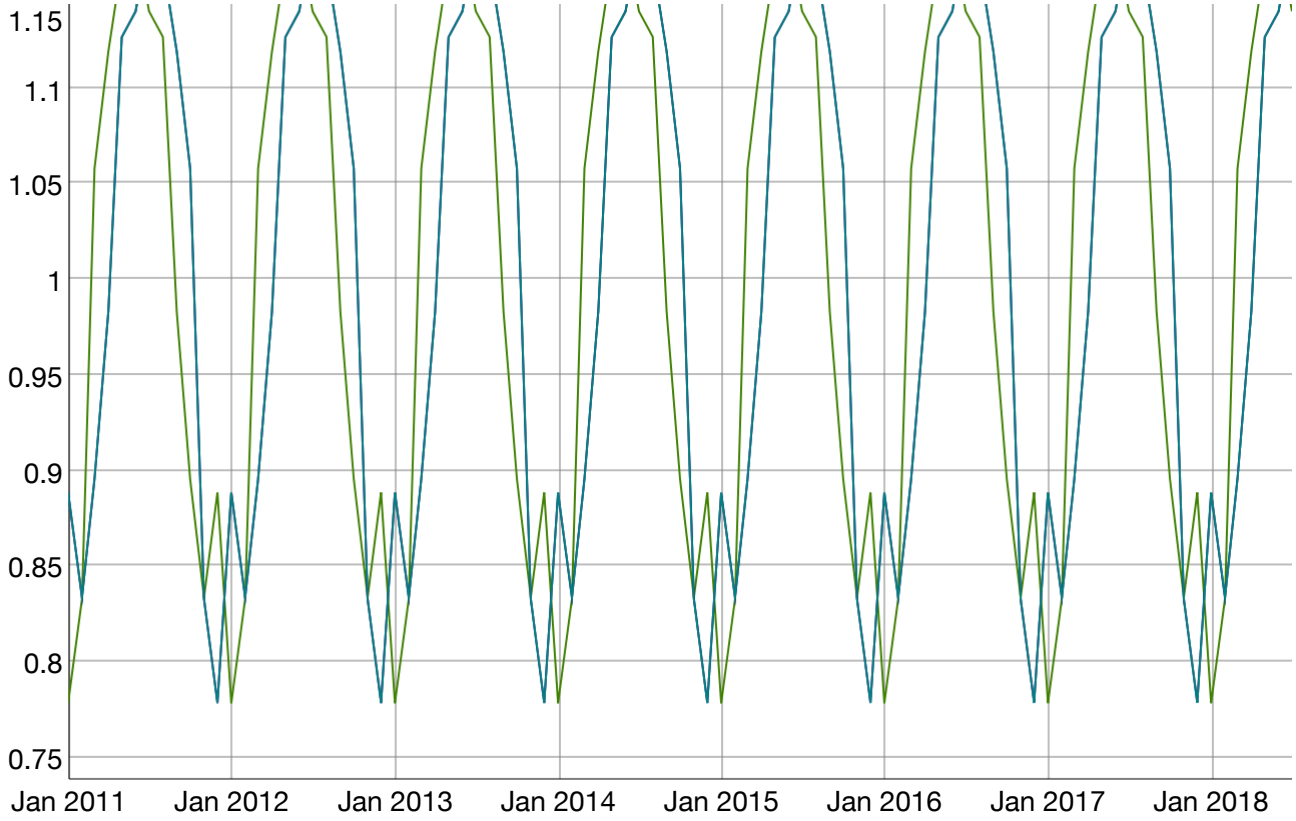



```
dygraph(cbind(etsfit$states[,'b'][-(1:1)],etsmmmparam$b,etsmmmparamv2$b))
```




```
dygraph(cbind(etsfit$states[,'s1'][-(1:1)],etsmmmparam$s,etsmmmparamv2$s))
```




1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018