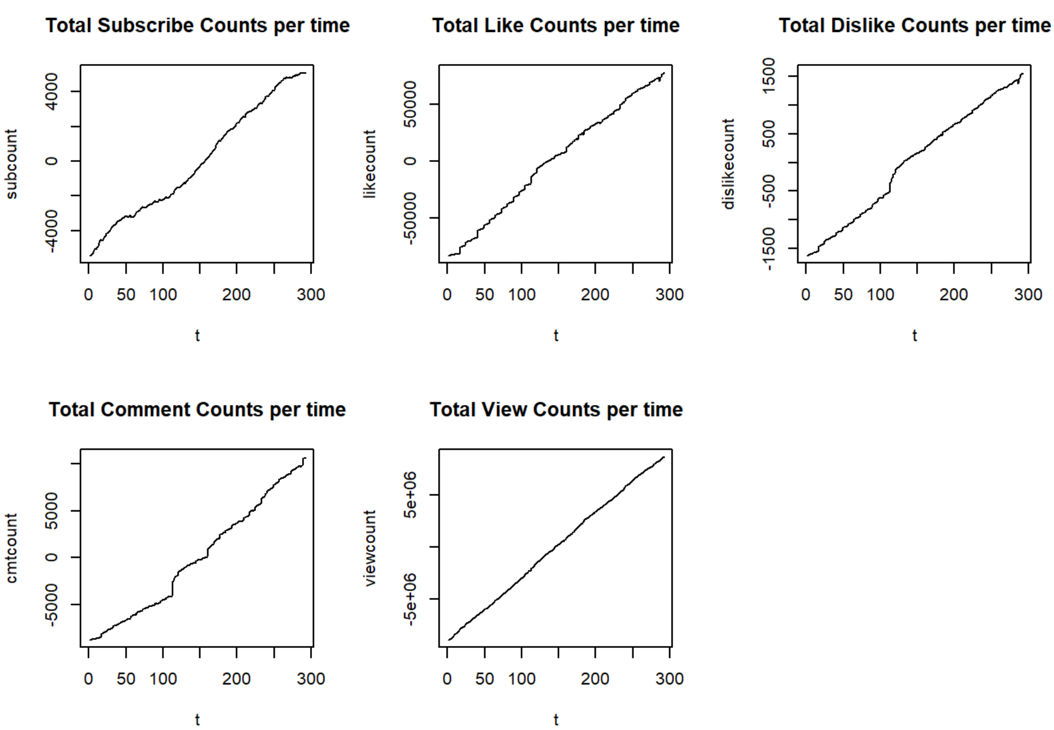
time_series_project.R

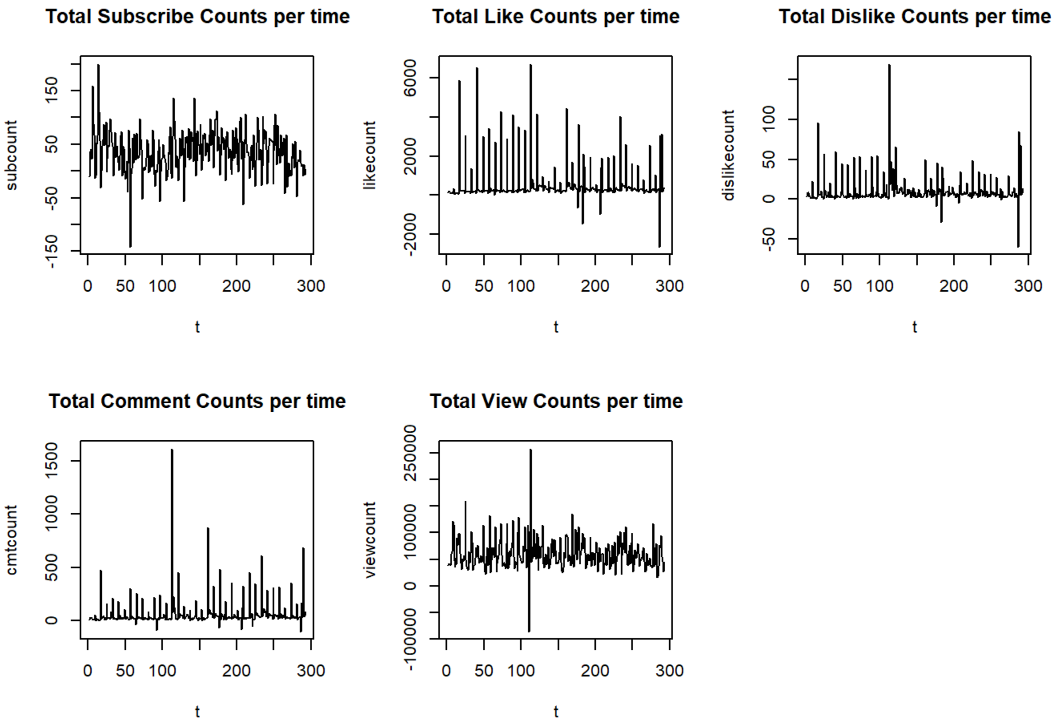
swlee

Fri May 03 09:57:06 2019

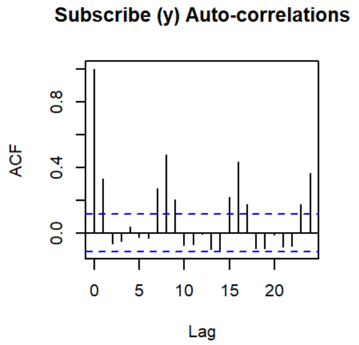
```
## reading data
setwd('C:\\Users\\swlee\\OneDrive\\Desktop\\2019Spr_MSDS\\Time Series\\youtube-5000-channels-videos-daily-co
unt-every-3h')
data<-read.csv('data.csv', header =T)


## centering to zero to meet the assumption (E(X) has to equal to zero)
likecount<-data$likecount - mean(data$likecount)
dislikecount<-data$dislikecount -  mean(data$dislikecount)
cmtcount<-data$cmtcount -  mean(data$cmtcount)
subcount<-data$subcount -  mean(data$subcount)
viewcount<-data$viewcount - mean(data$viewcount)
t<-data$t

## plotting time series data
par(mfrow=c(2,3))
plot(subcount~t, type='l', main='Total Subscribe Counts per time')
plot(likecount~t, type='l', main='Total Like Counts per time')
plot(dislikecount~t, type='l', main='Total Dislike Counts per time')
plot(cmtcount~t, type='l', main='Total Comment Counts per time')
plot(viewcount~t, type='l', main='Total View Counts per time')

likecount<-diff(data$likecount)
dislikecount<-diff(data$dislikecount)
cmtcount<-diff(data$cmtcount)
subcount<-diff(data$subcount)
viewcount<-diff(data$viewcount)
t<-t[-1]

## plotting time series data (diff)
par(mfrow=c(2,3))
```

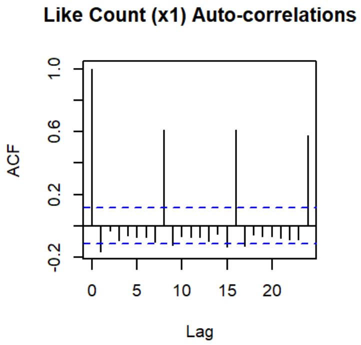


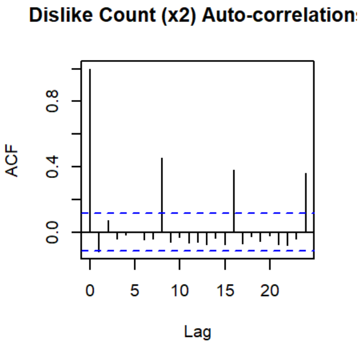
```
plot(subcount~t, type='l', main='Total Subscribe Counts per time')
plot(likecount~t, type='l', main='Total Like Counts per time')
plot(dislikecount~t, type='l', main='Total Dislike Counts per time')
plot(cmtcount~t, type='l', main='Total Comment Counts per time')
plot(viewcount~t, type='l', main='Total View Counts per time')


## plotting autocovariance
par(mfrow=c(2,3))
```

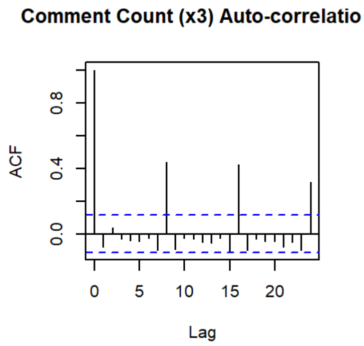


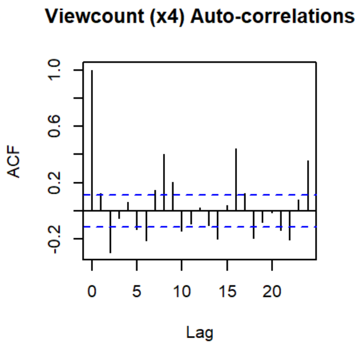

```
acf(subcount, main='Subscribe (y) Auto-correlations')
acf(likecount, main='Like Count (x1) Auto-correlations')
acf(dislikecount, main='Dislike Count (x2) Auto-correlations')
acf(cmtcount, main='Comment Count (x3) Auto-correlations')
acf(viewcount, main='Viewcount (x4) Auto-correlations')


autocov<-matrix(NA, nrow=25, ncol=5)
autocov[,1]<-as.vector(acf(subcount, plot = FALSE)$acf)
autocov[,2]<-as.vector(acf(likecount, plot = FALSE)$acf)
autocov[,3]<-as.vector(acf(dislikecount, plot = FALSE)$acf)
autocov[,4]<-as.vector(acf(cmtcount, plot = FALSE)$acf)
autocov[,5]<-as.vector(acf(viewcount, plot = FALSE)$acf)
write.csv(as.data.frame(autocov), file='autocovmat.csv' ,row.names=F)


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



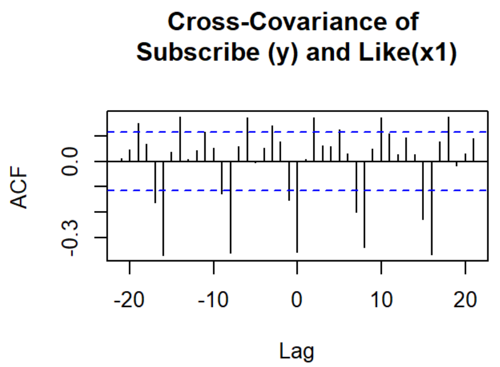


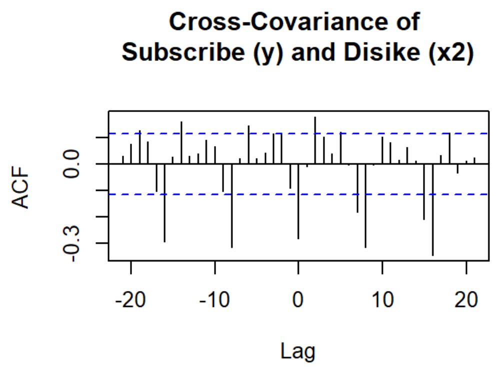


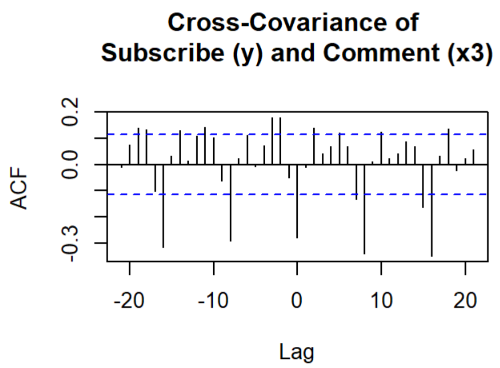


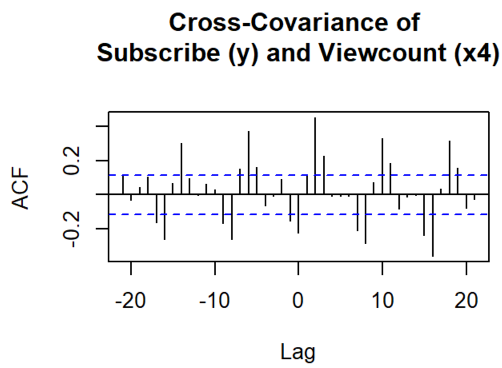



```
ccf(likecount, subcount,  main = 'Cross-Covariance of \nSubscribe (y) and Like(x1)')
ccf(dislikecount, subcount, main='Cross-Covariance of \nSubscribe (y) and Disike (x2)')
ccf(cmtcount, subcount,  main = 'Cross-Covariance of \nSubscribe (y) and Comment (x3)')
ccf(viewcount, subcount,  main = 'Cross-Covariance of \nSubscribe (y) and Viewcount (x4)')
```









```r
ccfcov<-matrix(NA, nrow=43, ncol=4)
ccfcov[,1]<-as.vector(ccf(likecount, subcount, plot=F)$acf)
ccfcov[,2]<-as.vector(ccf(dislikecount, subcount, plot=F)$acf)
ccfcov[,3]<-as.vector(ccf(cmtcount, subcount,   plot=F)$acf)
ccfcov[,4]<-as.vector(ccf(viewcount, subcount,   plot=F)$acf)

write.csv(as.data.frame(ccfcov), file='ccfmat.csv' ,row.names=F)


## Spectral Analysis for seasonal term
par(mfrow=c(2,3))
spec.sub<-spec.pgram(subcount,spans=10,taper=0,log="no")
spec.like<-spec.pgram(likecount,spans=10,taper=0,log="no")
spec.dislike<-spec.pgram(dislikecount,spans=10,taper=0,log="no")
spec.cmt<-spec.pgram(cmtcount, spans=10, taper=0, log='no')
spec.view<-spec.pgram(viewcount, spans=10, taper=0, log='no')


## There seems like seasonal (daily) pattern in every 8 time period, Which is obvious because there is patte
rn of sleeping and being awake.
subspec<-cbind(spec.sub$freq, spec.sub$spec)
likespec<-cbind(spec.like$freq, spec.like$spec)
dislikespec<-cbind(spec.dislike$freq, spec.dislike$spec)
cmtspec<-cbind(spec.cmt$freq, spec.cmt$spec)
viewspec<-cbind(spec.view$freq, spec.view$spec)


write.csv(subspec, 'subspec.csv')

library(aTSA)
```

```
## Warning: package 'aTSA' was built under R version 3.5.2
```

```
##
## Attaching package: 'aTSA'
```

```
## The following object is masked from 'package:graphics':
##
##     identify
```

```r
adf.test(subcount, nlag=8)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 -7.85  0.0100
## [2,]   1 -6.87  0.0100
## [3,]   2 -4.88  0.0100
## [4,]   3 -3.82  0.0100
## [5,]   4 -3.68  0.0100
## [6,]   5 -3.07  0.0100
## [7,]   6 -2.15  0.0324
## [8,]   7 -1.53  0.1323
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -12.07  0.0100
## [2,]   1 -11.91  0.0100
## [3,]   2  -9.25  0.0100
## [4,]   3  -7.79  0.0100
## [5,]   4  -7.61  0.0100
## [6,]   5  -6.69  0.0100
## [7,]   6  -4.37  0.0100
## [8,]   7  -2.81  0.0608
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -12.09   0.010
## [2,]   1 -11.94   0.010
## [3,]   2  -9.28   0.010
## [4,]   3  -7.83   0.010
## [5,]   4  -7.63   0.010
## [6,]   5  -6.71   0.010
## [7,]   6  -4.38   0.010
## [8,]   7  -2.84   0.223
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```
adf.test(likecount, nlag = 8)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -15.42   0.010
## [2,]   1  -9.34   0.010
## [3,]   2  -7.18   0.010
## [4,]   3  -5.77   0.010
## [5,]   4  -4.84   0.010
## [6,]   5  -4.15   0.010
## [7,]   6  -3.77   0.010
## [8,]   7  -1.44   0.165
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -20.04    0.01
## [2,]   1 -13.69    0.01
## [3,]   2 -11.85    0.01
## [4,]   3 -10.74    0.01
## [5,]   4 -10.20    0.01
## [6,]   5 -10.05    0.01
## [7,]   6 -10.60    0.01
## [8,]   7  -4.68    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -20.04    0.01
## [2,]   1 -13.70    0.01
## [3,]   2 -11.88    0.01
## [4,]   3 -10.81    0.01
## [5,]   4 -10.30    0.01
## [6,]   5 -10.22    0.01
## [7,]   6 -10.89    0.01
## [8,]   7  -4.89    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```
adf.test(dislikecount, nlag=8)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -13.90   0.010
## [2,]   1  -7.80   0.010
## [3,]   2  -5.96   0.010
## [4,]   3  -4.79   0.010
## [5,]   4  -3.89   0.010
## [6,]   5  -3.31   0.010
## [7,]   6  -2.95   0.010
## [8,]   7  -1.42   0.172
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -19.14    0.01
## [2,]   1 -11.96    0.01
## [3,]   2 -10.00    0.01
## [4,]   3  -8.82    0.01
## [5,]   4  -7.79    0.01
## [6,]   5  -7.31    0.01
## [7,]   6  -7.06    0.01
## [8,]   7  -3.71    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -19.12  0.0100
## [2,]   1 -11.95  0.0100
## [3,]   2 -10.00  0.0100
## [4,]   3  -8.83  0.0100
## [5,]   4  -7.80  0.0100
## [6,]   5  -7.34  0.0100
## [7,]   6  -7.10  0.0100
## [8,]   7  -3.73  0.0227
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

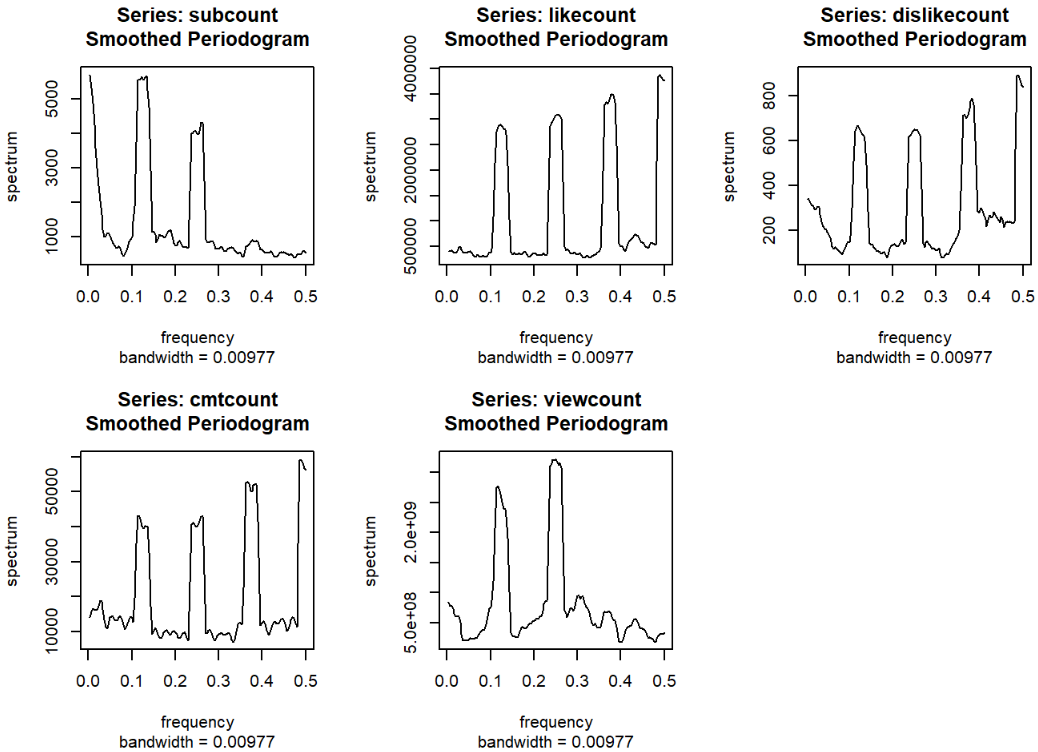
```
adf.test(cmtcount, nlag=8)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -15.04  0.0100
## [2,]   1  -9.11  0.0100
## [3,]   2  -7.07  0.0100
## [4,]   3  -5.88  0.0100
## [5,]   4  -5.03  0.0100
## [6,]   5  -4.31  0.0100
## [7,]   6  -4.05  0.0100
## [8,]   7  -2.06  0.0402
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -18.35    0.01
## [2,]   1 -12.02    0.01
## [3,]   2 -10.01    0.01
## [4,]   3  -8.98    0.01
## [5,]   4  -8.28    0.01
## [6,]   5  -7.64    0.01
## [7,]   6  -7.74    0.01
## [8,]   7  -4.32    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -18.45    0.01
## [2,]   1 -12.12    0.01
## [3,]   2 -10.13    0.01
## [4,]   3  -9.11    0.01
## [5,]   4  -8.43    0.01
## [6,]   5  -7.80    0.01
## [7,]   6  -7.92    0.01
## [8,]   7  -4.43    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

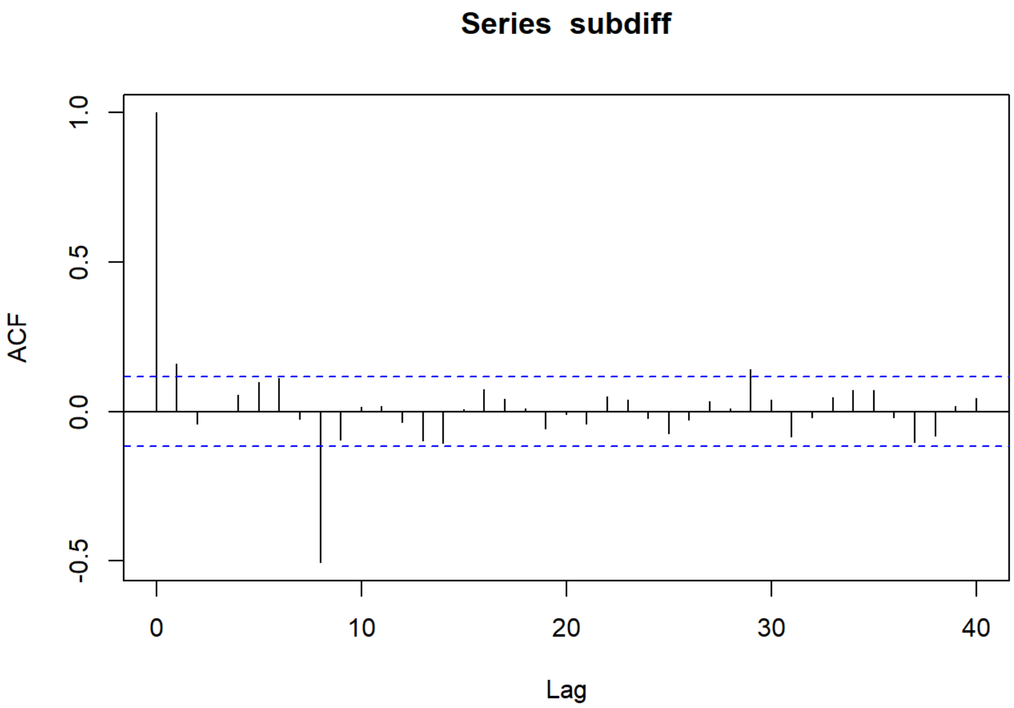
```
adf.test(viewcount, nlag=8)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]   0 -4.999  0.0100
## [2,]   1 -3.925  0.0100
## [3,]   2 -2.340  0.0205
## [4,]   3 -1.760  0.0788
## [5,]   4 -1.588  0.1114
## [6,]   5 -1.421  0.1710
## [7,]   6 -1.052  0.3027
## [8,]   7 -0.802  0.3922
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]   0 -14.96    0.01
## [2,]   1 -15.55    0.01
## [3,]   2 -11.00    0.01
## [4,]   3  -9.52    0.01
## [5,]   4  -9.74    0.01
## [6,]   5 -10.04    0.01
## [7,]   6  -7.37    0.01
## [8,]   7  -4.93    0.01
## Type 3: with drift and trend
##      lag    ADF p.value
## [1,]   0 -14.98    0.01
## [2,]   1 -15.61    0.01
## [3,]   2 -11.09    0.01
## [4,]   3  -9.64    0.01
## [5,]   4  -9.91    0.01
## [6,]   5 -10.27    0.01
## [7,]   6  -7.54    0.01
## [8,]   7  -5.03    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

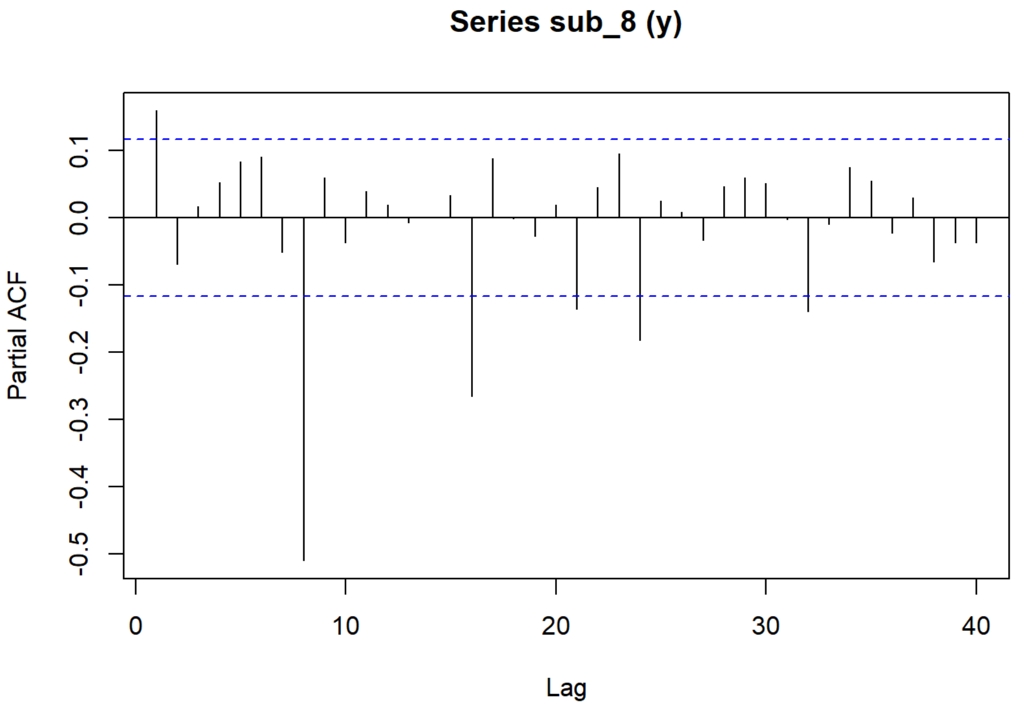
```
## Take 8th difference for 8 lagged term
subdiff<-diff(subcount, 8)
par(mfrow=c(1,1))
```



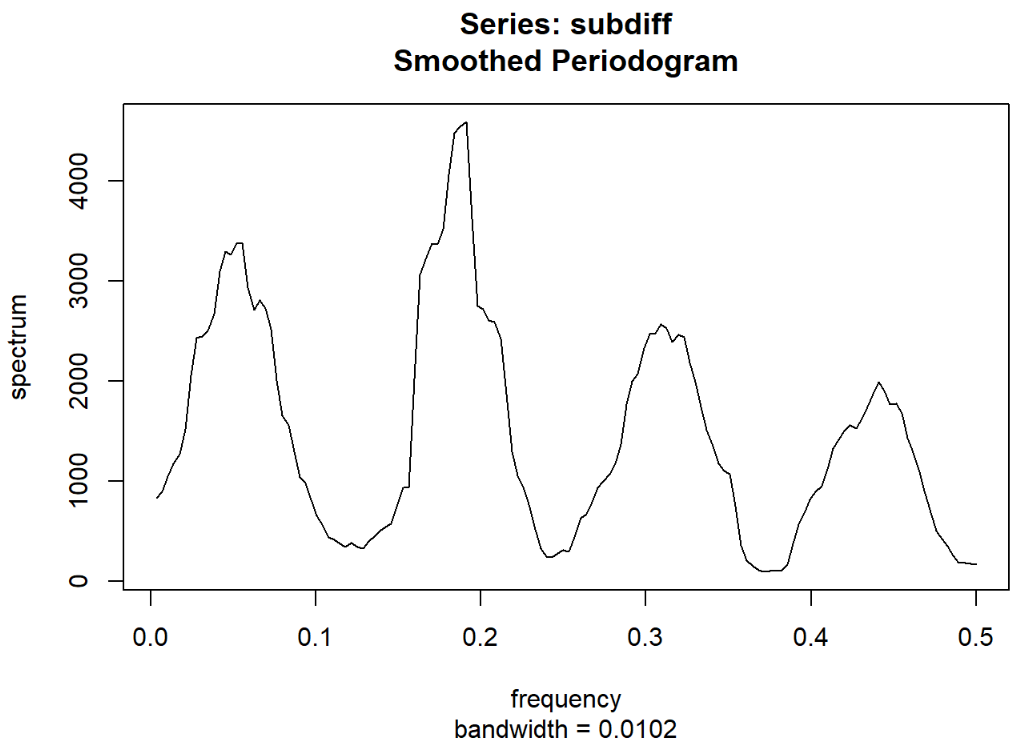

```
acf(subdiff, lag.max=40)
```



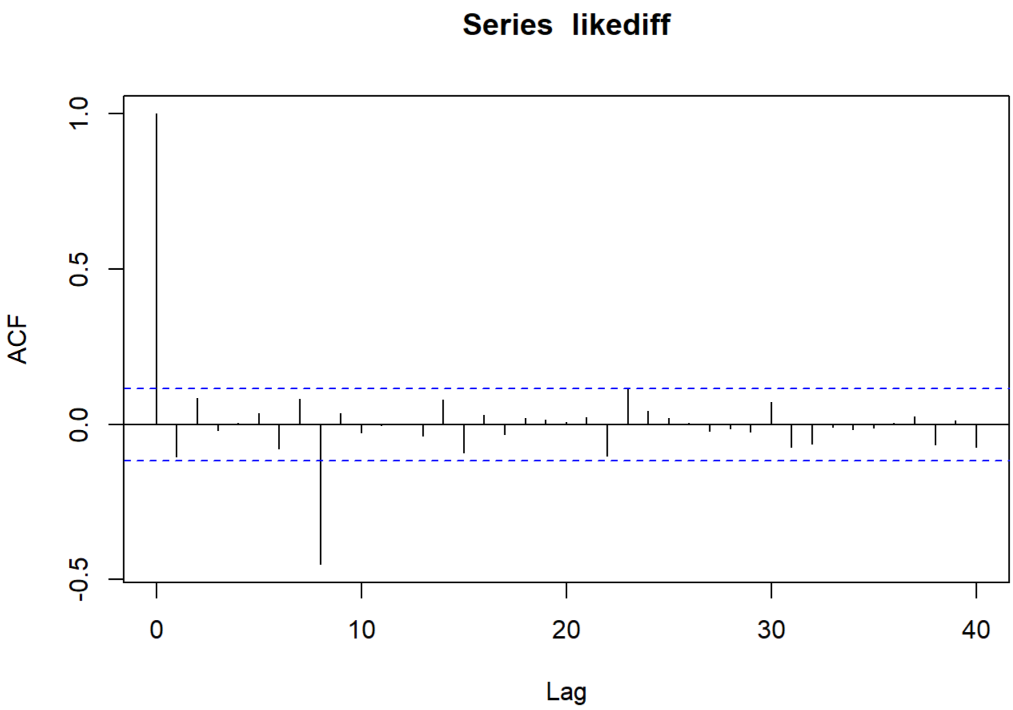

```
pacf(subdiff, lag.max=40, main='Series sub_8 (y)')
```



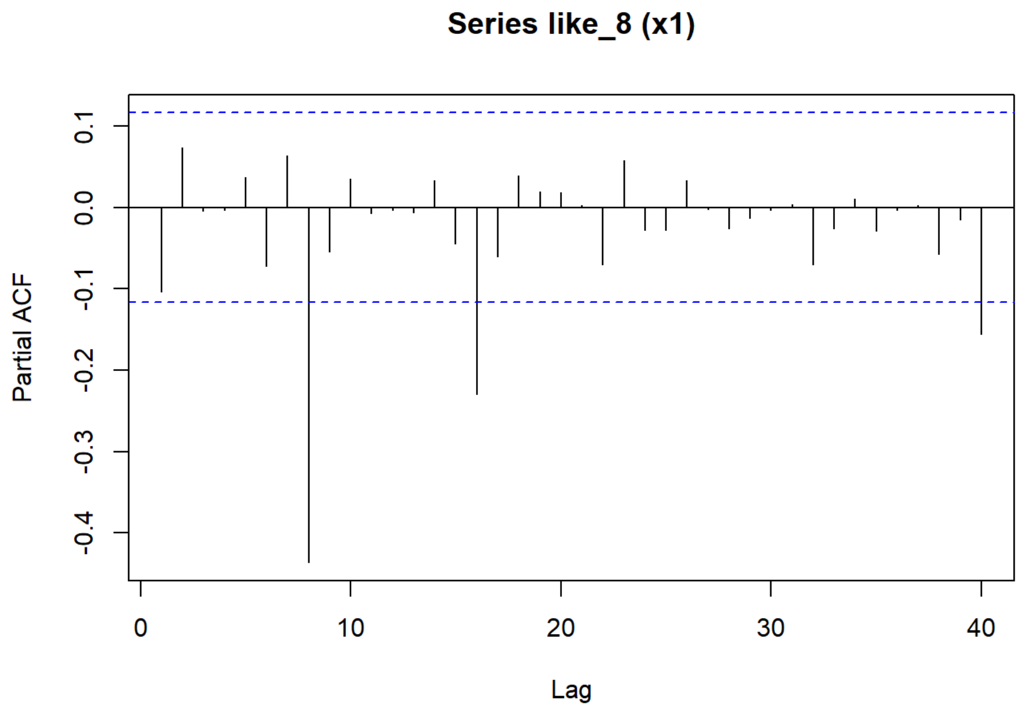

```
spec.pgram(subdiff, spans=10,taper=0,log="no")
```



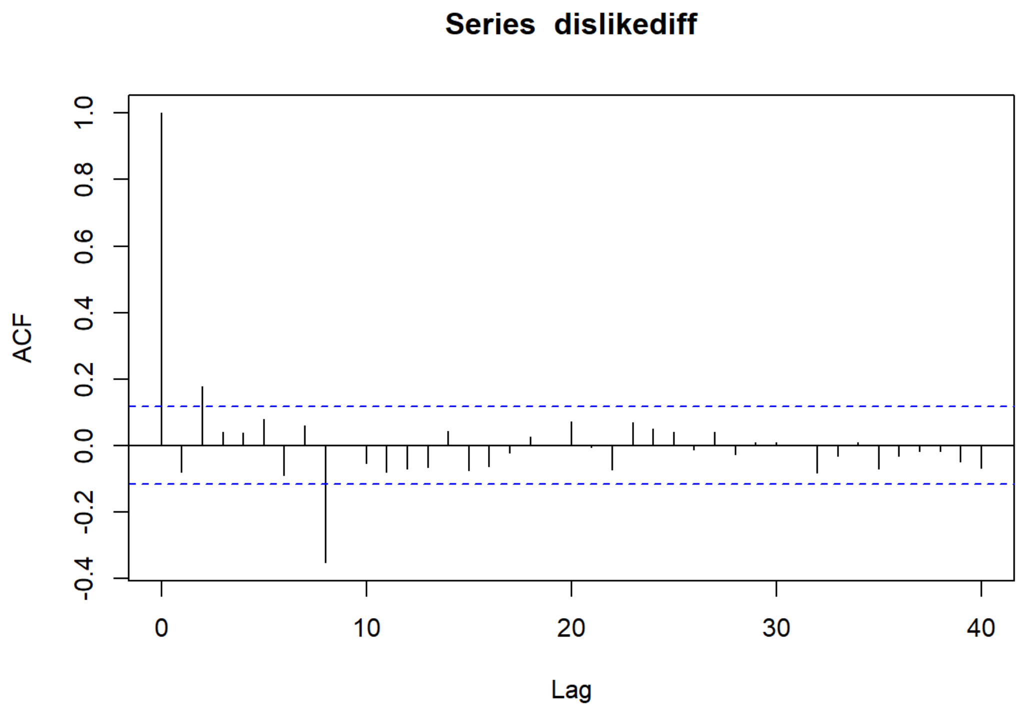

```
## seasonal AR(3)
## non-seasonal AR(2 or 1)
likediff<-diff(likecount, 8)
par(mfrow=c(1,1))
acf(likediff, lag.max=40)
```



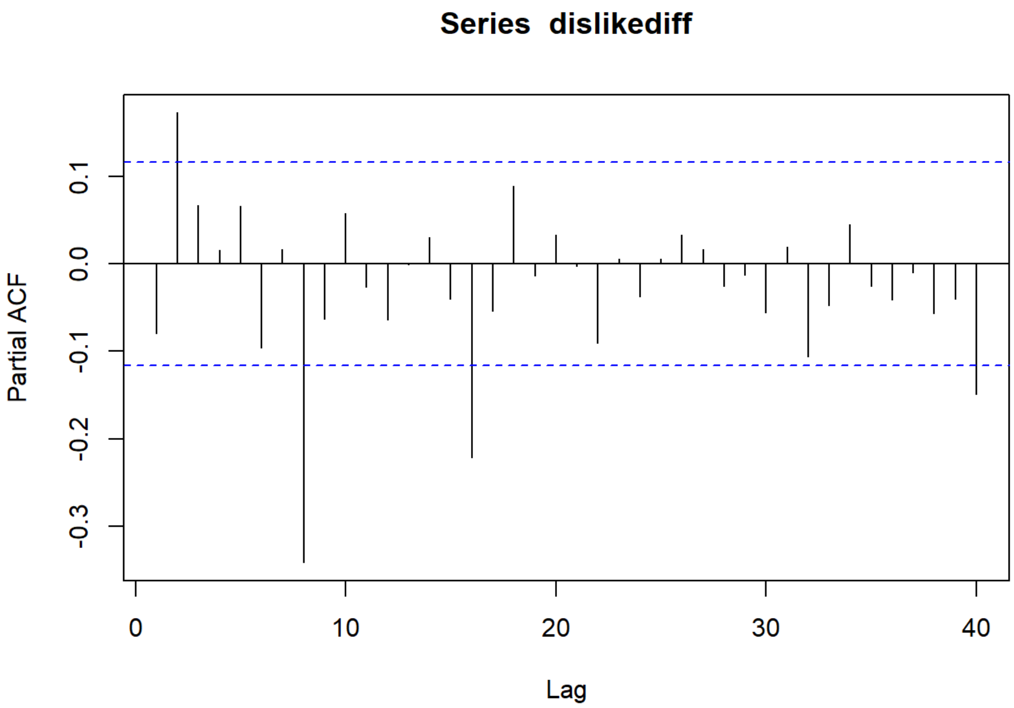

```
pacf(likediff, lag.max=40, main='Series like_8 (x1)')
```



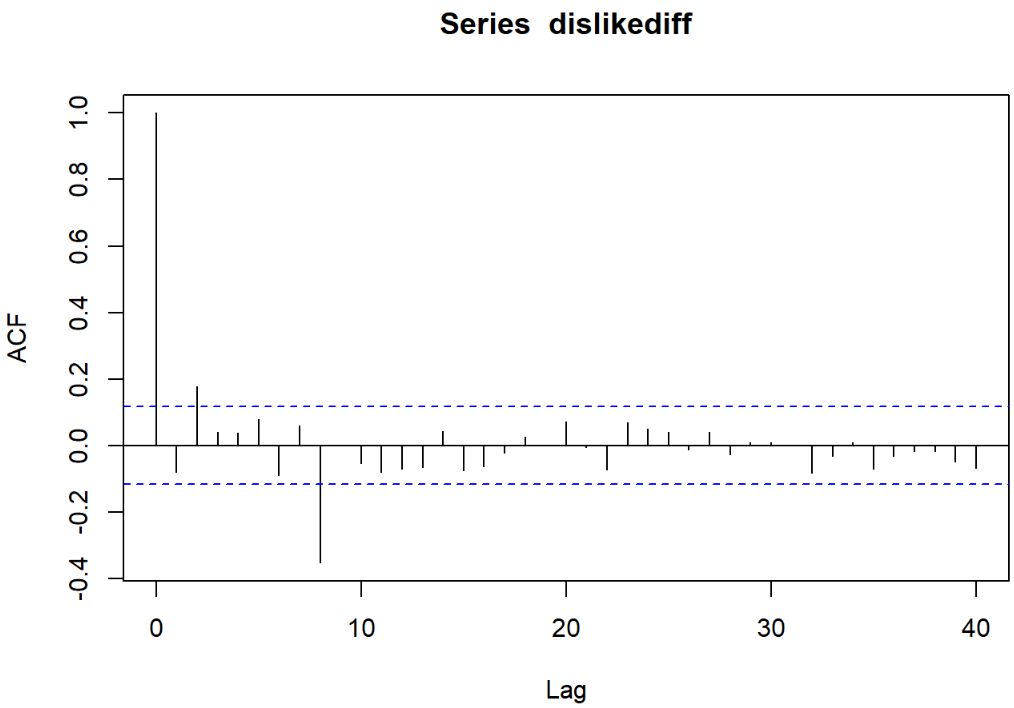

```
## seasonal AR(2)
## non-seasonal AR(0 or 1)

dislikediff<-diff(dislikecount, 8)
par(mfrow=c(1,1))
acf(dislikediff, lag.max=40)
```



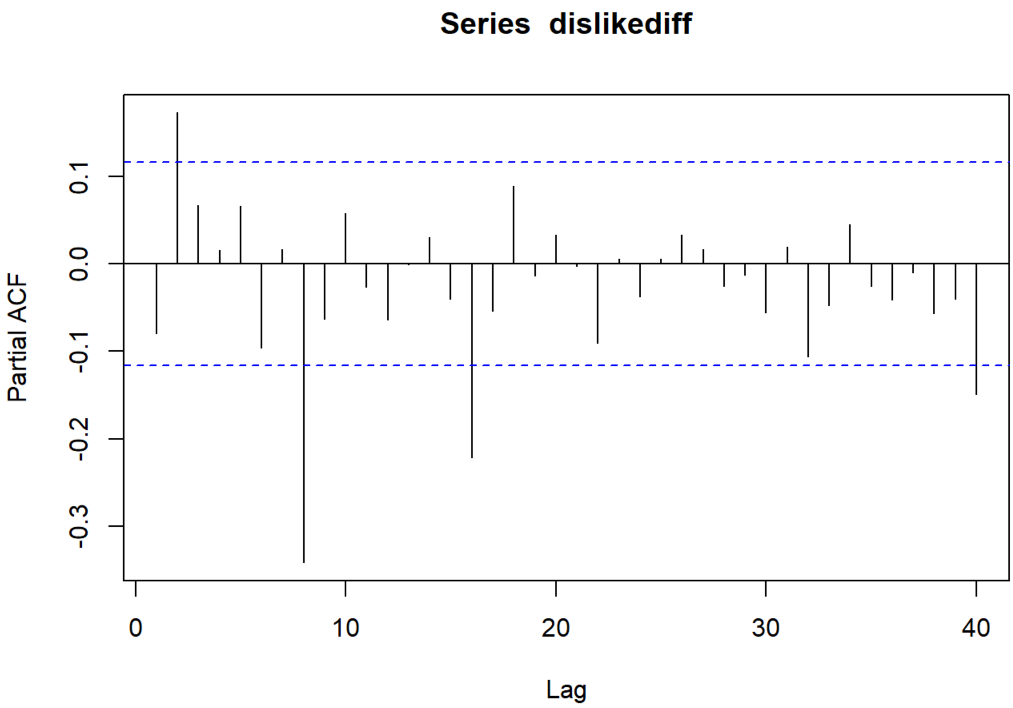

```
pacf(dislikediff, lag.max=40)
```



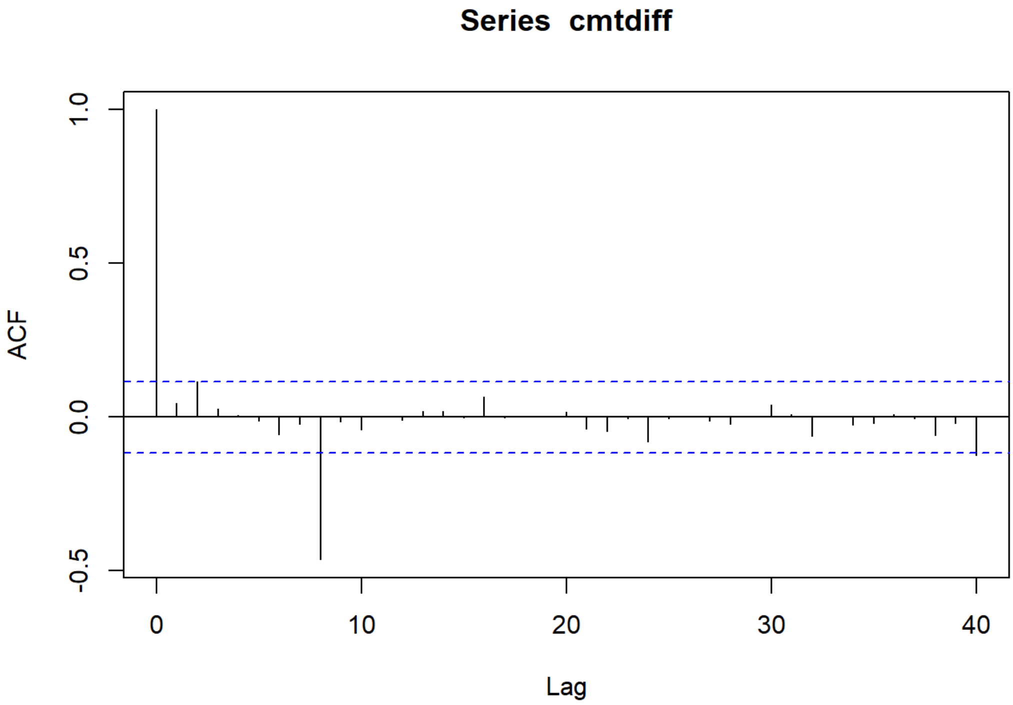

```
## seasonal AR(2)
## non-seasonal AR(0 or 2)

dislikediff<-diff(dislikecount, 8)
par(mfrow=c(1,1))
acf(dislikediff, lag.max=40)
```



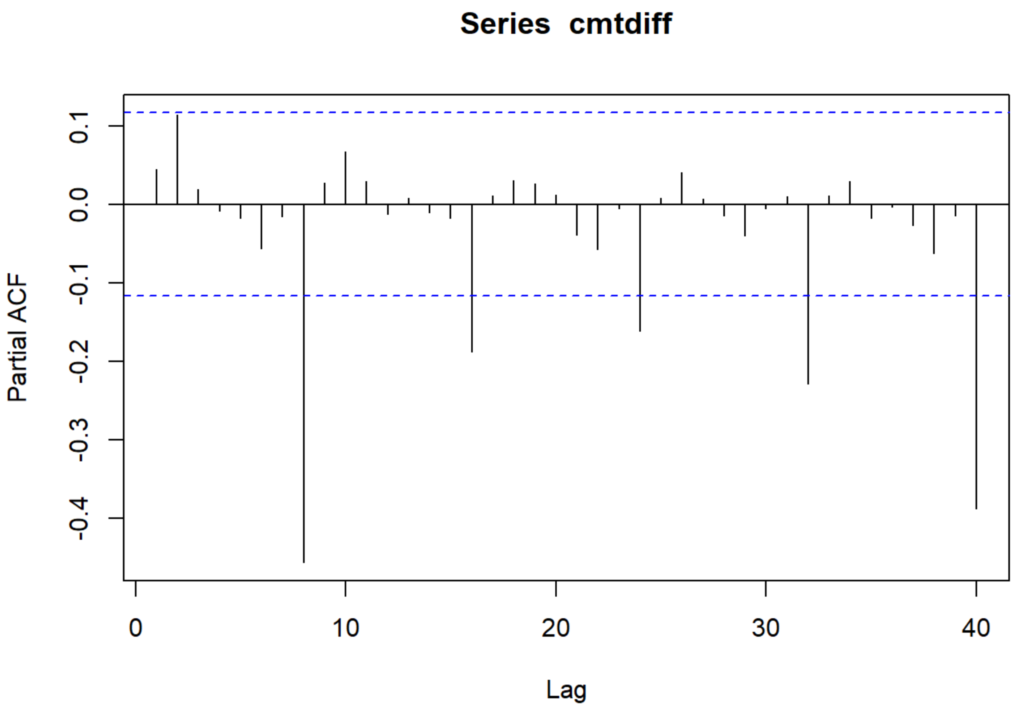

```
pacf(dislikediff, lag.max=40)
```



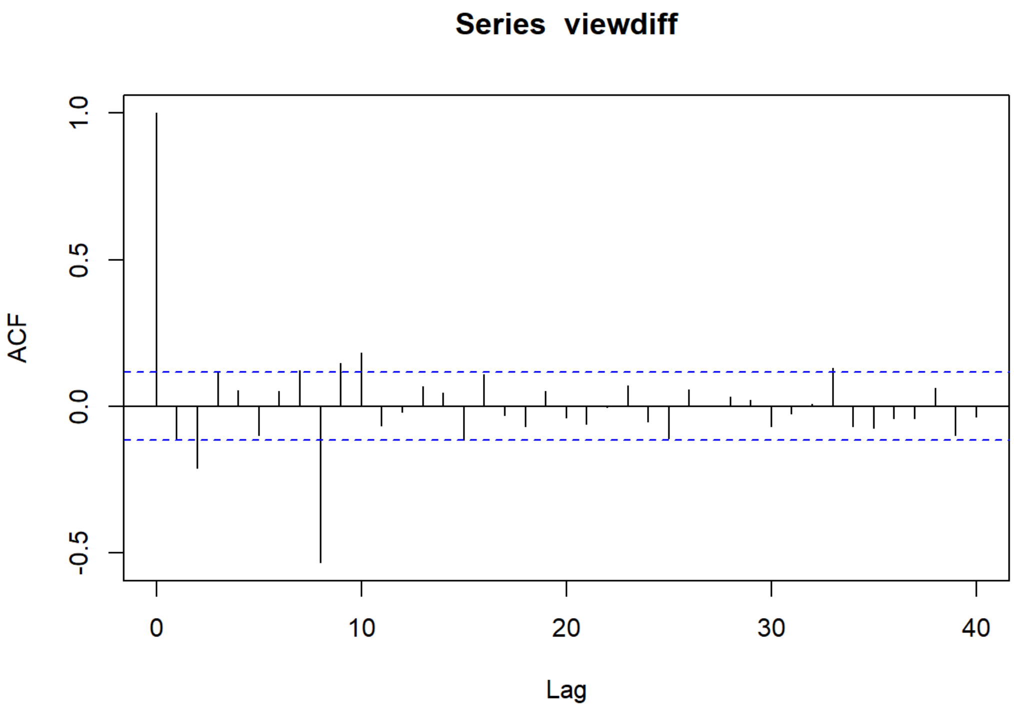

```
cmtdiff<-diff(cmtcount, 8)
par(mfrow=c(1,1))
acf(cmtdiff, lag.max=40)
```



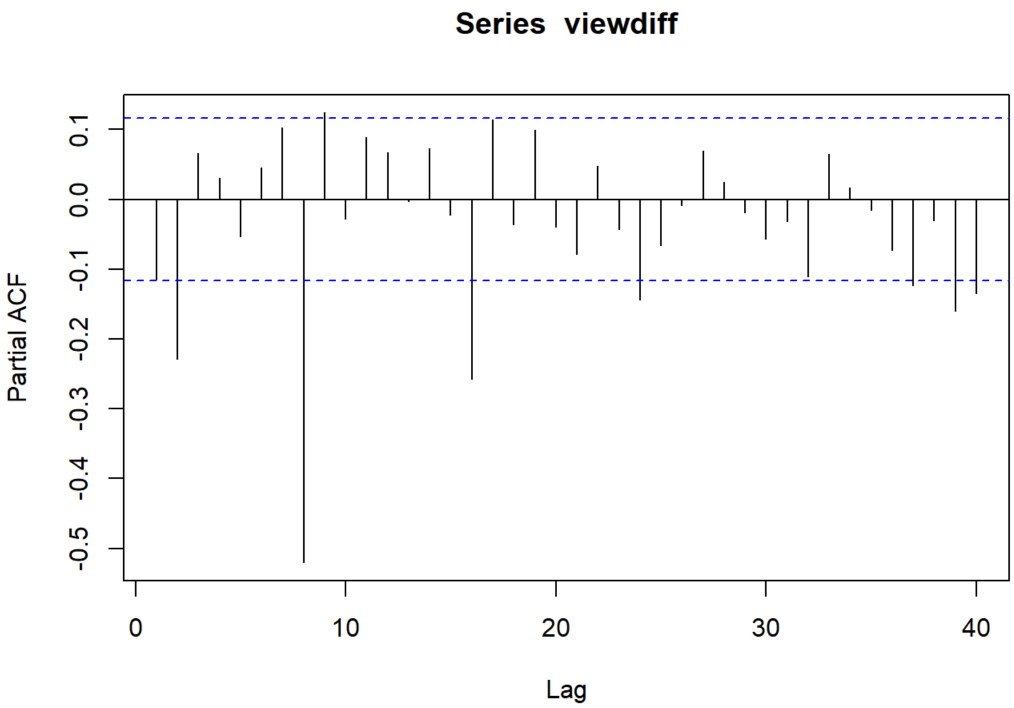

```
pacf(cmtdiff, lag.max=40)
```




```
## seasonal AR(3)
## non-seasonal AR(0 or 1)

viewdiff<-diff(viewcount, 8)
par(mfrow=c(1,1))
acf(viewdiff, lag.max=40)
```




```
pacf(viewdiff, lag.max=40)
```


Series viewdiff
Partial ACF
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
0
10
20
30
40
Lag

```
## seasonal AR(3)
## non-seasonal AR(2 or 1)

## making lag terms for model-fitting
B = c(0,1)
like_1 = filter(likecount,B,sides=1)
dislike_1 = filter(dislikecount, B, sides=1)
sub_1 = filter(subcount,B,sides=1)
cmt_1 = filter(cmtcount,B,sides=1)
view_1 = filter(viewcount,B,sides=1)

B = c(0,0,1)
like_2 = filter(likecount,B,sides=1)
dislike_2 = filter(dislikecount, B, sides=1)
sub_2 = filter(subcount,B,sides=1)
cmt_2 = filter(cmtcount,B,sides=1)
view_2 = filter(viewcount,B,sides=1)

N = length(subcount)
rm(B)

like_1 = like_1[3:N]
like_2 = like_2[3:N]
dislike_1 = dislike_1[3:N]
dislike_2 = dislike_2[3:N]
view_1 = view_1[3:N]
view_2 = view_2[3:N]
cmt_1 = cmt_1[3:N]
cmt_2 = cmt_2[3:N]

subcount<-subcount[3:N]
likecount<-likecount[3:N]
dislikecount<-dislikecount[3:N]
cmtcount<-cmtcount[3:N]
viewcount<-viewcount[3:N]




N = N-2
N1 = N - 50
N2 = seq(N1+1,N)

X1 = cbind(likecount, like_1, like_2, dislikecount, dislike_1, dislike_2, viewcount, view_1, view_2, cmtcoun
t, cmt_1, cmt_2)

subf = subcount[N2]
X1f = X1[N2,]
subt = subcount[1:N1]
X1t = X1[1:N1,]

## fitting arimax model

# simplist
res1 = arima(subt, xreg=X1t, order=c(1,0,1), seasonal = list(order=c(2,1,0), period=8))
res1
```

```
##
## Call:
## arima(x = subt, order = c(1, 0, 1), seasonal = list(order = c(2, 1, 0), period = 8),
##     xreg = X1t)
##
## Coefficients:
##           ar1     ma1     sar1     sar2  likecount   like_1   like_2
##       -0.1871  0.3436  -0.6691  -0.2728    -0.0014  -0.0041  -0.0050
## s.e.   0.3243  0.3077   0.0674   0.0727     0.0049   0.0049   0.0048
##       dislikecount  dislike_1  dislike_2  viewcount  view_1  view_2
##            -0.1212     0.1017     0.2274      2e-04   1e-04   2e-04
## s.e.        0.2900     0.2800     0.2801      1e-04   1e-04   1e-04
##       cmtcount   cmt_1   cmt_2
##         0.0285  0.0334  0.0256
## s.e.    0.0289  0.0281  0.0291
##
## sigma^2 estimated as 914.9:  log likelihood = -1117.28,  aic = 2268.56
```

```
# AR(2)
res2 = arima(subt, xreg=X1t, order=c(2,0,1), seasonal = list(order=c(2,1,0), period=8))
res2
```

```
##
## Call:
## arima(x = subt, order = c(2, 0, 1), seasonal = list(order = c(2, 1, 0), period = 8),
##     xreg = X1t)
##
## Coefficients:
##           ar1     ar2     ma1     sar1    sar2  likecount   like_1
##       -0.8331  0.1125  1.0000  -0.6594  -0.263    -0.0012  -0.0038
## s.e.   0.0671  0.0689  0.0255   0.0691   0.074     0.0050   0.0049
##        like_2  dislikecount  dislike_1  dislike_2  viewcount  view_1
##       -0.0055       -0.1789     0.1455     0.2032      2e-04   1e-04
## s.e.   0.0049        0.2910     0.2811     0.2819      1e-04   1e-04
##       view_2  cmtcount   cmt_1   cmt_2
##        2e-04    0.0331  0.0279  0.0325
## s.e.   1e-04    0.0287  0.0277  0.0288
##
## sigma^2 estimated as 898.2:  log likelihood = -1116.23,  aic = 2268.45
```

```
# SMA(1)
res3 = arima(subt, xreg=X1t, order=c(1,0,1), seasonal = list(order=c(2,1,1), period=8))
res3
```

```
##
## Call:
## arima(x = subt, order = c(1, 0, 1), seasonal = list(order = c(2, 1, 1), period = 8),
##     xreg = X1t)
##
## Coefficients:
##          ar1      ma1     sar1    sar2     sma1  likecount   like_1
##       0.9639  -0.8833  -0.0705  0.0321  -0.8057      2e-04  -0.0029
## s.e.  0.0348   0.0562   0.0969  0.0918   0.0685      5e-03   0.0048
##        like_2  dislikecount  dislike_1  dislike_2  viewcount  view_1
##       -0.0066       -0.0439     0.0884     0.1713      2e-04   0e+00
## s.e.   0.0048        0.2973     0.2914     0.2931      1e-04   1e-04
##       view_2  cmtcount   cmt_1   cmt_2
##        1e-04    0.0107  0.0354  0.0536
## s.e.   1e-04    0.0285  0.0273  0.0289
##
## sigma^2 estimated as 829.6:  log likelihood = -1108.15,  aic = 2252.29
```

```
# SMA(2)
res4 =arima(subt, xreg=X1t, order=c(1,0,1), seasonal = list(order=c(2,1,2), period=8))
res4
```

```
##
## Call:
## arima(x = subt, order = c(1, 0, 1), seasonal = list(order = c(2, 1, 2), period = 8),
##     xreg = X1t)
##
## Coefficients:
##          ar1      ma1     sar1    sar2     sma1     sma2  likecount
##       0.9641  -0.8834  -0.2792  0.0181  -0.5963  -0.1674      2e-04
## s.e.  0.0346   0.0560   0.9250  0.1255   0.9218   0.7282      5e-03
##        like_1   like_2  dislikecount  dislike_1  dislike_2  viewcount
##       -0.0029  -0.0065       -0.0394     0.0865     0.1704      2e-04
## s.e.   0.0048   0.0048        0.2978     0.2914     0.2931      1e-04
##       view_1  view_2  cmtcount   cmt_1   cmt_2
##        0e+00   1e-04    0.0106  0.0354  0.0534
## s.e.   1e-04   1e-04    0.0284  0.0273  0.0289
##
## sigma^2 estimated as 829.5:  log likelihood = -1108.13,  aic = 2254.25
```

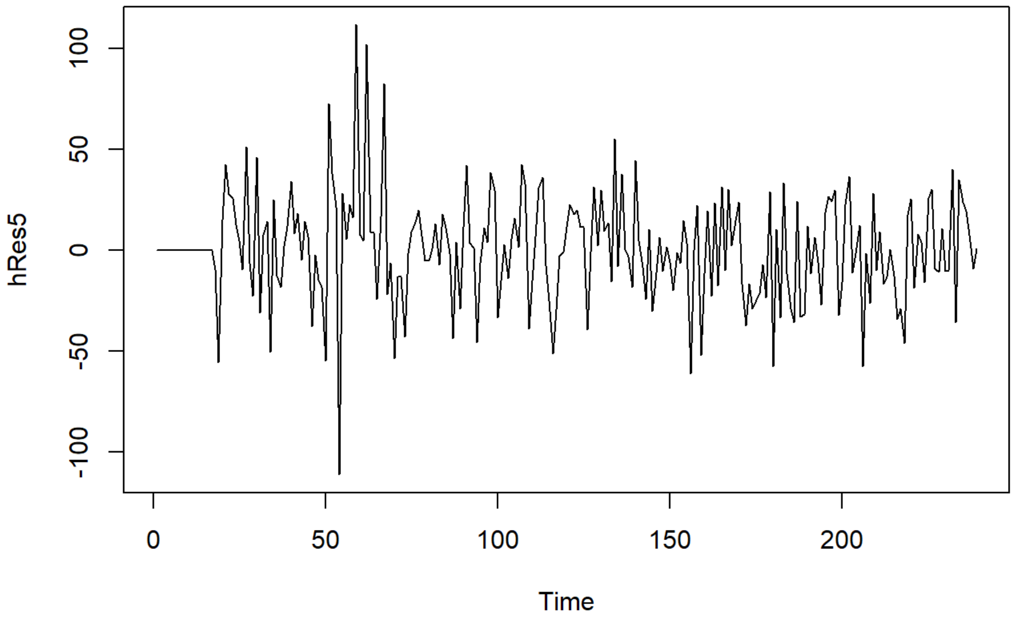
```
# increase everything by 1 except moving SMA
res5 =arima(subt, xreg=X1t, order=c(2,1,2), seasonal = list(order=c(2,2,2), period=8))
res5
```

```
##
## Call:
## arima(x = subt, order = c(2, 1, 2), seasonal = list(order = c(2, 2, 2), period = 8),
##     xreg = X1t)
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1    sar2     sma1    sma2
##       0.5564  -0.1860  -1.3547  0.4432  0.0129  0.0821  -1.9585  0.9942
## s.e.  0.2876   0.0864   0.2764  0.2539  0.0842  0.0835   0.1046  0.1062
##       likecount   like_1   like_2  dislikecount  dislike_1  dislike_2
##         -0.0016  -0.0027  -0.0069        0.1097     0.2174     0.2572
## s.e.     0.0047   0.0046   0.0046        0.2973     0.2899     0.2872
##       viewcount  view_1  view_2  cmtcount   cmt_1   cmt_2
##           2e-04   1e-04   2e-04    0.0162  0.0203  0.0330
## s.e.      1e-04   1e-04   1e-04    0.0281  0.0269  0.0288
##
## sigma^2 estimated as 778.5:  log likelihood = -1091.9,  aic = 2225.8
```

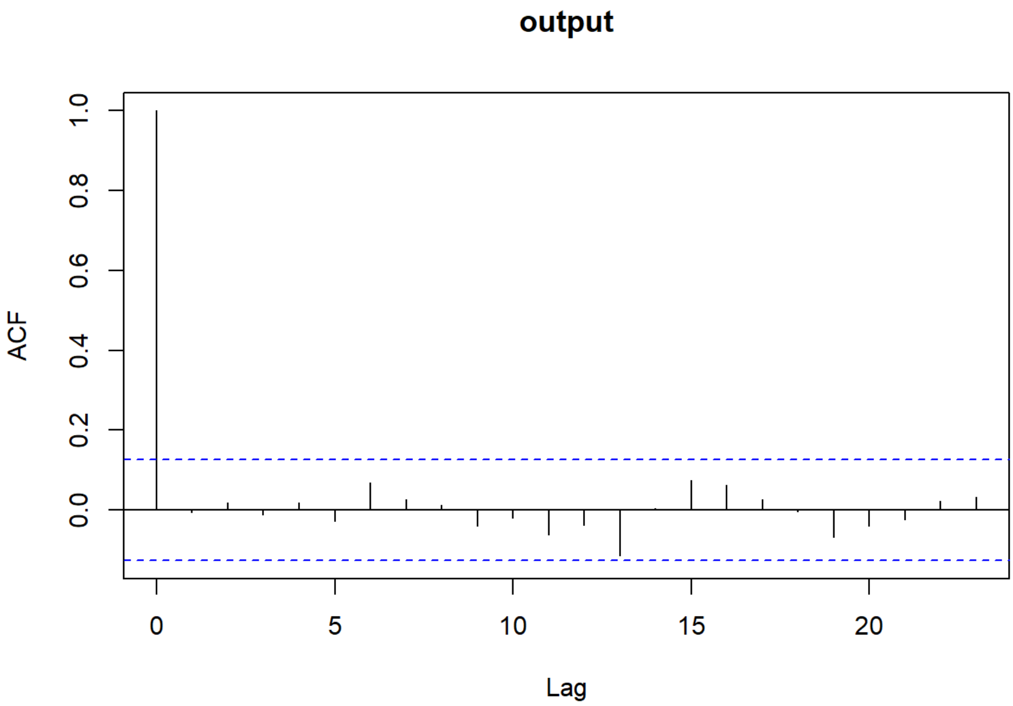
```
AICval<-c(AIC(res1), AIC(res2), AIC(res3), AIC(res4), AIC(res5))
BICval<-c(BIC(res1), BIC(res2), BIC(res3), BIC(res4), BIC(res5))
loglikli<-c(-2*res1$loglik + 9*log(N), -2*res2$loglik + 9*log(N), -2*res3$loglik + 9*log(N), -2*res4$loglik
+ 9*log(N), -2*res5$loglik + 9*log(N))

testresults<-cbind(AICval, BICval, loglikli)
rownames(testresults)<-c("res1", "res2", "res3", "res4", "res5")
testresults
```

```
##        AICval   BICval loglikli
## res1 2268.556 2327.077 2285.554
## res2 2268.454 2330.418 2283.452
## res3 2252.291 2314.255 2267.289
## res4 2254.254 2319.660 2267.252
## res5 2225.800 2297.256 2234.797
```

```
write.csv(testresults, 'testresults.csv')
write.csv(res5$coef, file='res5coef.csv')
## comparing AIC, BIC, loglikelihood values, res5 seems to be the best model among the 4 models

hRes5 = residuals(res5)
plot(hRes5)
```



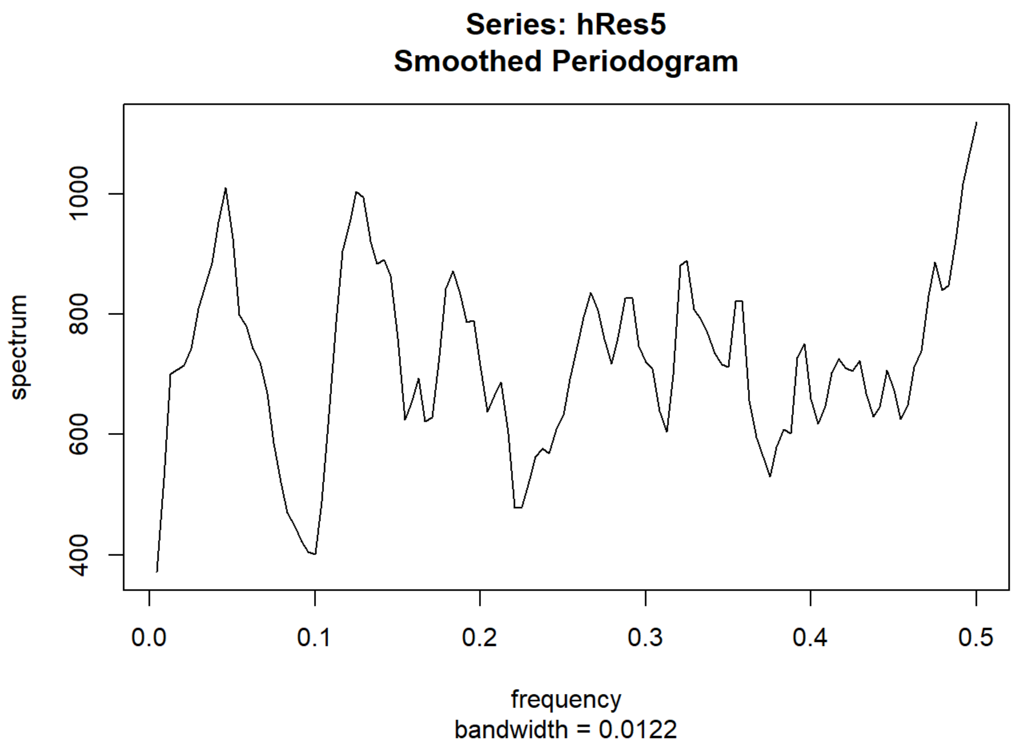

```
acf(hRes5, main="output")
```



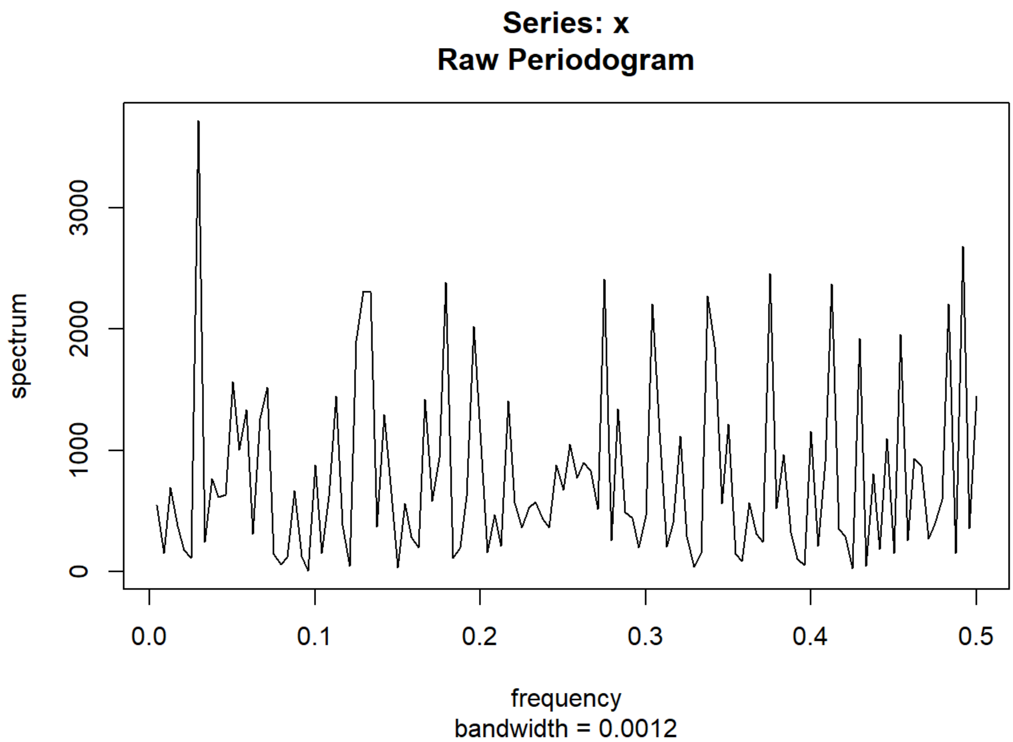

```
par(mfrow=c(1,1))
Box.test(hRes5,lag=20)
```

```
##
##  Box-Pierce test
##
## data:  hRes5
## X-squared = 10.394, df = 20, p-value = 0.9604
```

```
spec.pgram(hRes5, spans= 10, taper=0, log='no')
```




```
spectrum(hRes5,log='no')
```



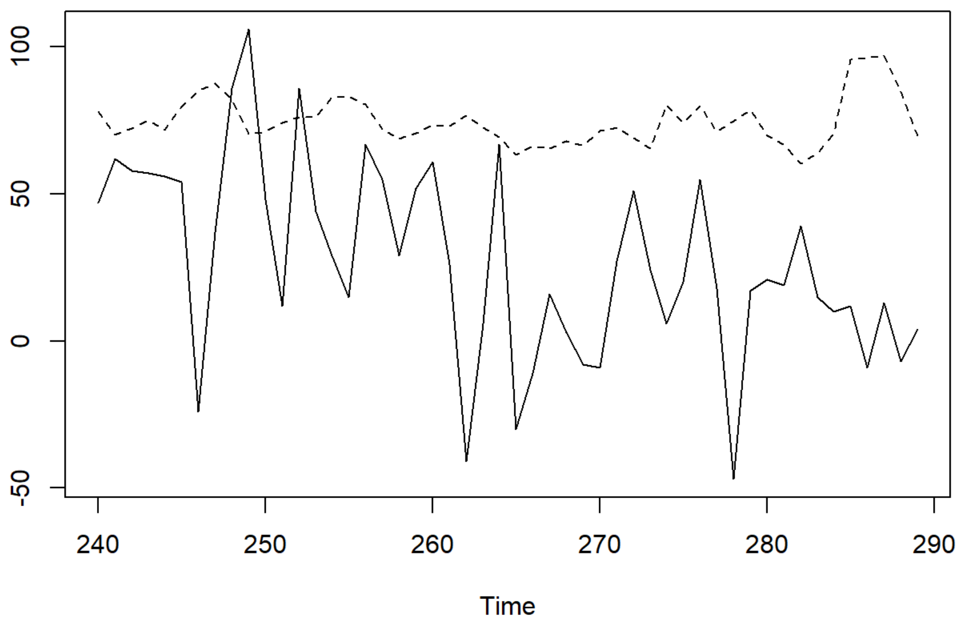

```
coef(res5)
```

```
##           ar1           ar2           ma1           ma2          sar1 
##  5.563731e-01 -1.860152e-01 -1.354747e+00  4.431564e-01  1.293568e-02 
##          sar2          sma1          sma2     likecount        like_1 
##  8.207612e-02 -1.958472e+00  9.942375e-01 -1.630284e-03 -2.729631e-03 
##        like_2  dislikecount     dislike_1     dislike_2     viewcount 
## -6.926844e-03  1.096708e-01  2.174098e-01  2.571911e-01  1.937541e-04 
##        view_1        view_2      cmtcount         cmt_1         cmt_2 
##  6.718189e-05  1.639484e-04  1.620639e-02  2.030994e-02  3.296984e-02
```

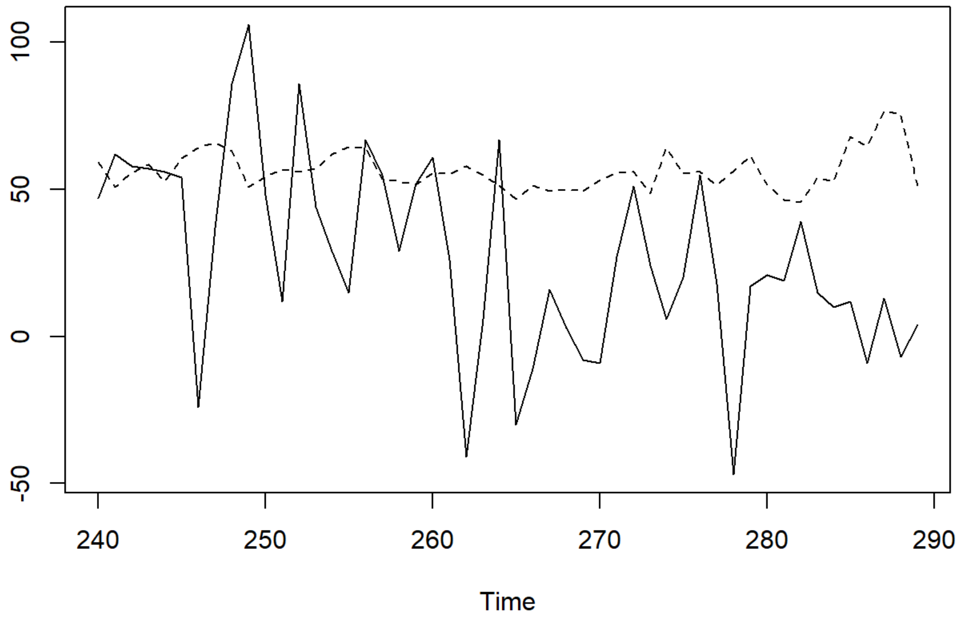
```
res5
```

```
##
## Call:
## arima(x = subt, order = c(2, 1, 2), seasonal = list(order = c(2, 2, 2), period = 8),
##     xreg = X1t)
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1    sar2     sma1    sma2
##       0.5564  -0.1860  -1.3547  0.4432  0.0129  0.0821  -1.9585  0.9942
## s.e.  0.2876   0.0864   0.2764  0.2539  0.0842  0.0835   0.1046  0.1062
##       likecount   like_1   like_2  dislikecount  dislike_1  dislike_2
##         -0.0016  -0.0027  -0.0069        0.1097     0.2174     0.2572
## s.e.     0.0047   0.0046   0.0046        0.2973     0.2899     0.2872
##       viewcount  view_1  view_2  cmtcount   cmt_1   cmt_2
##           2e-04   1e-04   2e-04    0.0162  0.0203  0.0330
## s.e.      1e-04   1e-04   1e-04    0.0281  0.0269  0.0288
##
## sigma^2 estimated as 778.5:  log likelihood = -1091.9,  aic = 2225.8
```

```
yh=predict(res5,newxreg=X1f)
yhp = yh$pred
ts.plot(subf, yhp, lty=1:2)
```




```
yh=predict(res4,newxreg=X1f)
yhp = yh$pred
ts.plot(subf, yhp, lty=1:2)
```




```
########################################################################################################
#                                                ....Etc....
#
########################################################################################################
# plot(data)
# plot(subcount~likecount)
# plot(subcount~dislikecount)
# plot(likecount~dislikecount)
# plot(viewcount~likecount)
# plot(cmtcount~likecount)
#
#
# # Plotting for entire lags for each variable
# par(mfrow=c(2,3))
# acf(subcount, main='Subscribe (y) Auto-correlations', lag.max=length(subcount))
# acf(likecount, main='Like Count (x) Auto-correlations', lag.max=length(likecount))
# acf(dislikecount, main='Dislike Count (x) Auto-correlations', lag.max=length(dislikecount))
# acf(viewcount, main='Viewcount (x) Auto-correlations', lag.max=length(viewcount))
# acf(cmtcount, main='Comment Count (x) Auto-correlations', lag.max=length(cmtcount))
#
# # Fitting sine and cosine on spectral term ==> didn't work well
# sub.spectralfit=lm(subcount~cos(2*pi*t/8)+ sin(2*pi*t))
# summary(sub.spectralfit)
#
# par(mfrow=c(1,2))
# acf(subcount, main='Subscribe (y) Auto-correlations')
# acf(sub.spectralfit$residuals, main='Subscribe (y) Auto-correlations (After)')
# acf(subcount, main='Subscribe (y) Auto-correlations', lag.max=length(subcount))
# acf(sub.spectralfit$residuals, main='Subscribe (y) Auto-correlations (After)', lag.max=length(subcount))
#
# like.spectralfit=lm(likecount~cos(2*pi*t/8)+sin(2*pi*t))
# summary(like.spectralfit)
#
# acf(likecount, main='Like Count (x1) Auto-correlations')
# acf(like.spectralfit$residuals, main='Like (x1) Auto-correlations (After)')
# acf(likecount, main='Like Count (x) Auto-correlations', lag.max=length(likecount))
# acf(like.spectralfit$residuals, main='Like (y) Auto-correlations (After)', lag.max=length(likecount))
#
# dislike.spectralfit=lm(dislikecount~cos(2*pi*t/8))
# summary(dislike.spectralfit)
#
# acf(dislikecount, main='Disike Count (x) Auto-correlations')
# acf(dislike.spectralfit$residuals, main='Dislike (y) Auto-correlations (After)')
# acf(dislikecount, main='Dislike Count (x) Auto-correlations', lag.max=length(likecount))
# acf(dislike.spectralfit$residuals, main='Dislike (y) Auto-correlations (After)', lag.max=length(likecount)
```

```
)
#
# viewcount.spectralfit=lm(viewcount~cos(2*pi*t/8))
# summary(viewcount.spectralfit)
#
# acf(viewcount, main='View Count (x) Auto-correlations')
# acf(viewcount.spectralfit$residuals, main='View Count (y) Auto-correlations (After)')
# acf(viewcount, main='View Count(x) Auto-correlations', lag.max=length(likecount))
# acf(viewcount.spectralfit$residuals, main='View Count(y) Auto-correlations (After)', lag.max=length(likeco
unt))
#
# cmt.spectralfit=lm(cmtcount~cos(2*pi*t/8))
# summary(cmt.spectralfit)
#
# acf(cmtcount, main='Comment Count (x) Auto-correlations')
# acf(cmt.spectralfit$residuals, main='Comment (y) Auto-correlations (After)')
# acf(cmtcount, main='Comment Count (x) Auto-correlations', lag.max=length(likecount))
# acf(cmt.spectralfit$residuals, main='Comment (y) Auto-correlations (After)', lag.max=length(likecount))
#
#
# acf(subcount, plot=F)[8]
# acf(sub.spectralfit$residuals, plot=F)[8]
#
# acf(likecount, plot=F)[8]
# acf(like.spectralfit$residuals, plot=F)[8]
#
# acf(dislikecount, plot=F)[8]
# acf(dislike.spectralfit$residuals, plot=F)[8]
#
# acf(cmtcount, plot=F)[8]
# acf(cmt.spectralfit$residuals, plot=F)[8]
#
# acf(viewcount, plot=F)[8]
# acf(viewcount.spectralfit$residuals, plot=F)[8]
#
# subcount<-sub.spectralfit$residuals
# likecount<-like.spectralfit$residuals
# dislikecount<-dislike.spectralfit$residuals
# viewcount<-viewcount.spectralfit$residuals
# cmtcount<-cmt.spectralfit$residuals
```