Time Series Analysis 2

Trend and Seasonality Estimation Example 2

Time Series Analysis
Zhe Zheng

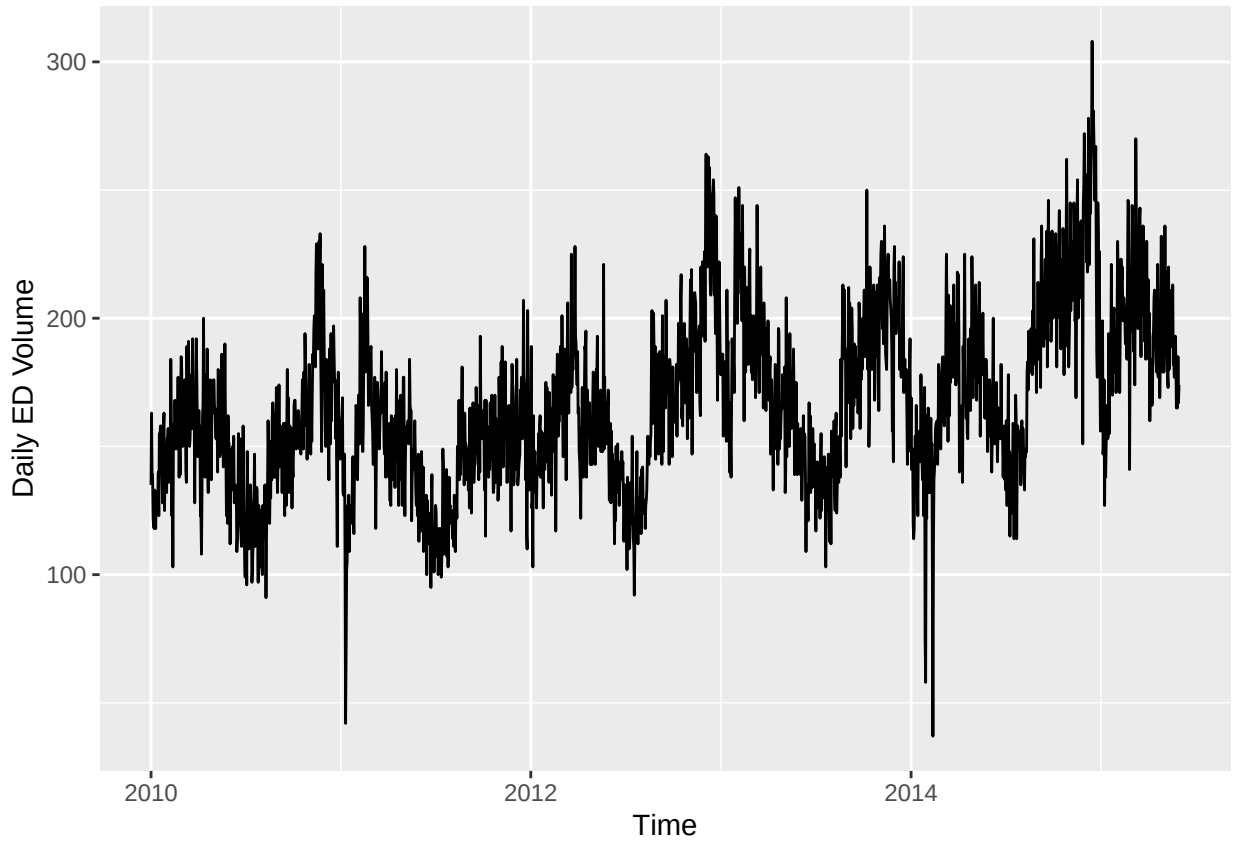
```
## [1] "English_United States.1252"
```

```
rm(list=ls())
library(ggplot2)
library(mgcv)
options(digits=3)
```

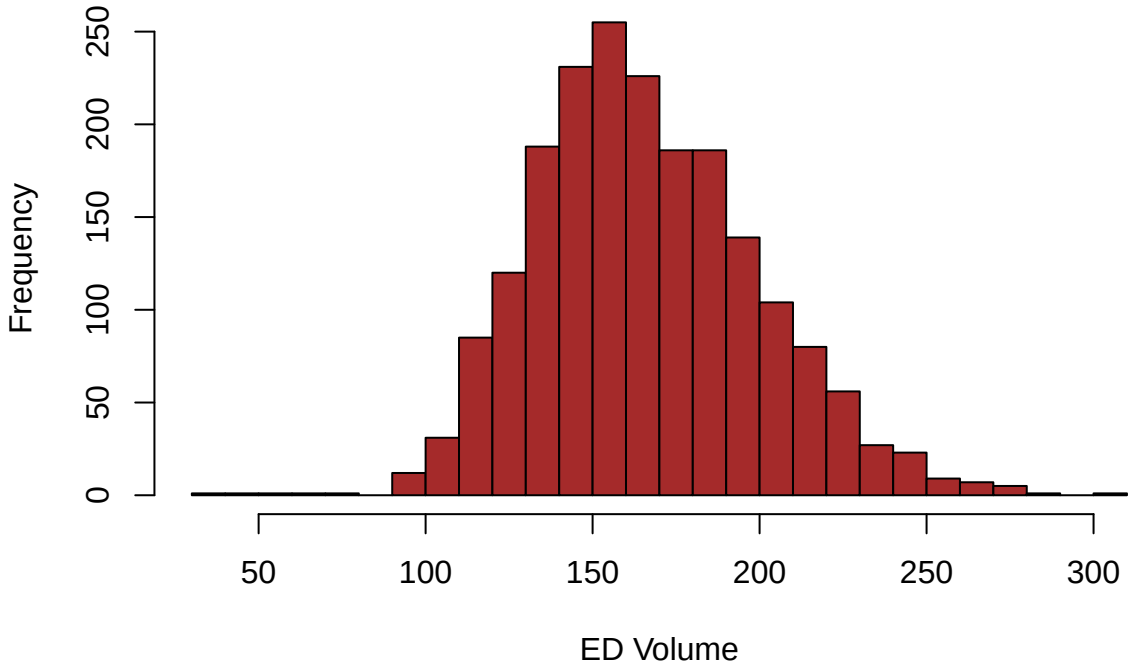
DATA EXPLORATION AND PROCESSING

```
edvoldata = read.csv("EGDailyVolume.csv",header=T)
## Process Dates
year = edvoldata$Year
month = edvoldata$Month
day = edvoldata$Day
datemat = cbind(as.character(day),as.character(month),as.character(year))
paste.dates = function(date){
    day = date[1]; month=date[2]; year = date[3]
    return(paste(day,month,year,sep="/"))
 }
dates = apply(datemat,1,paste.dates) # 1->row, 2-> column
dates = as.Date(dates, format="%d/%m/%Y")
edvoldata = cbind(dates,edvoldata)
attach(edvoldata)
```

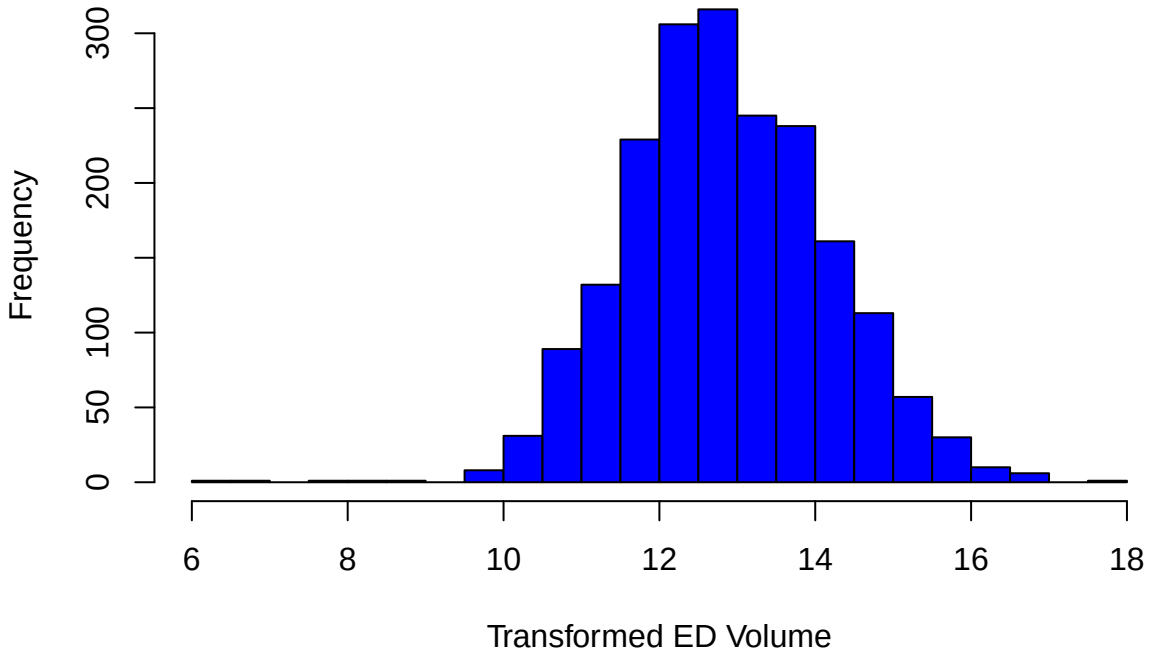
```
#plot(dates, Volume, type='l',ylab='Daily ED Volume',xlab='Time')
ggplot(edvoldata, aes(dates, Volume)) + geom_line() + xlab("Time") + ylab("Daily ED Volume")
```



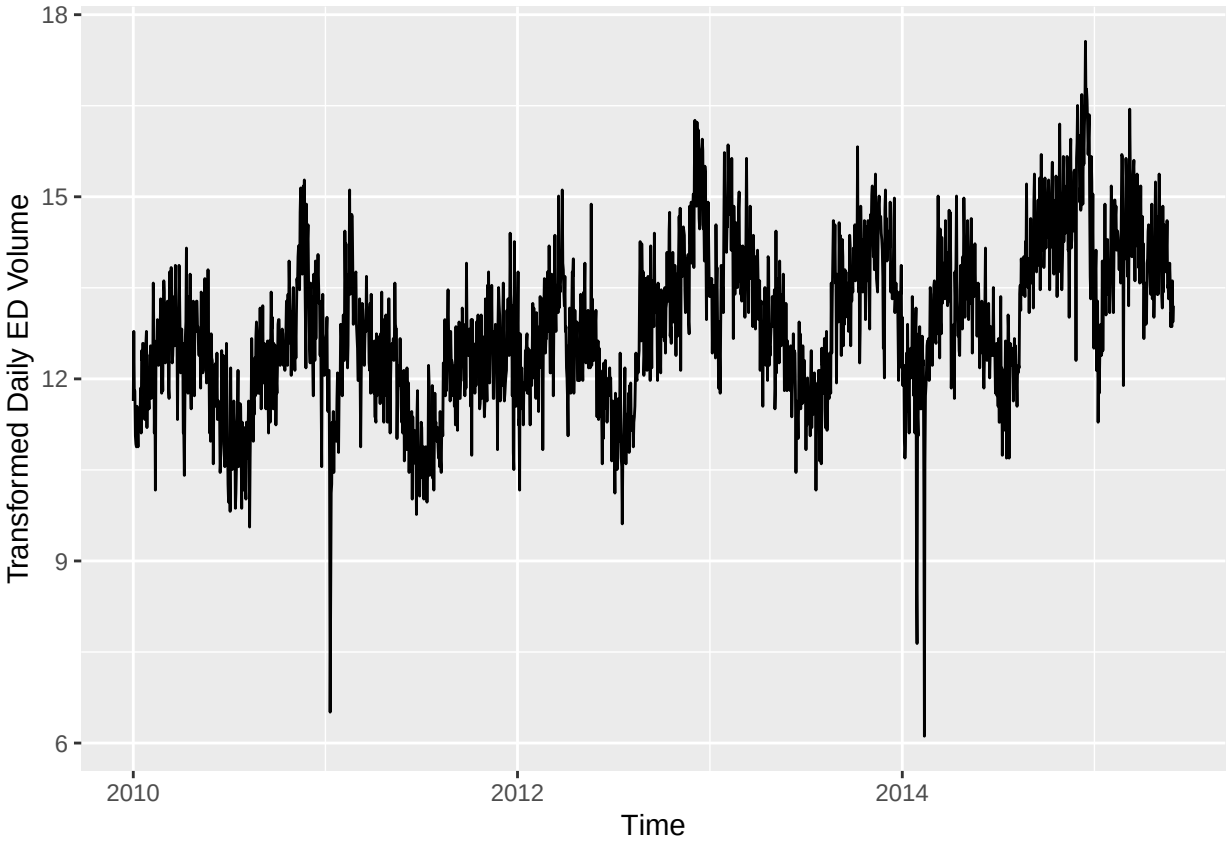

```
## ED Volume is count data: Transform
Volume.tr = sqrt(Volume+3/8)
hist(Volume,nclass=20,xlab="ED Volume", main="",col="brown")
```




```
hist(Volume.tr,nclass=20,xlab= "Transformed ED Volume", main="",col="blue")
```




```
#plot(dates, Volume.tr, type='l',ylab='Transformed Daily ED Volume',xlab='Time')
ggplot(edvoldata, aes(dates, sqrt(Volume+3/8))) + geom_line() + xlab("Time") + ylab("Transformed Daily
```



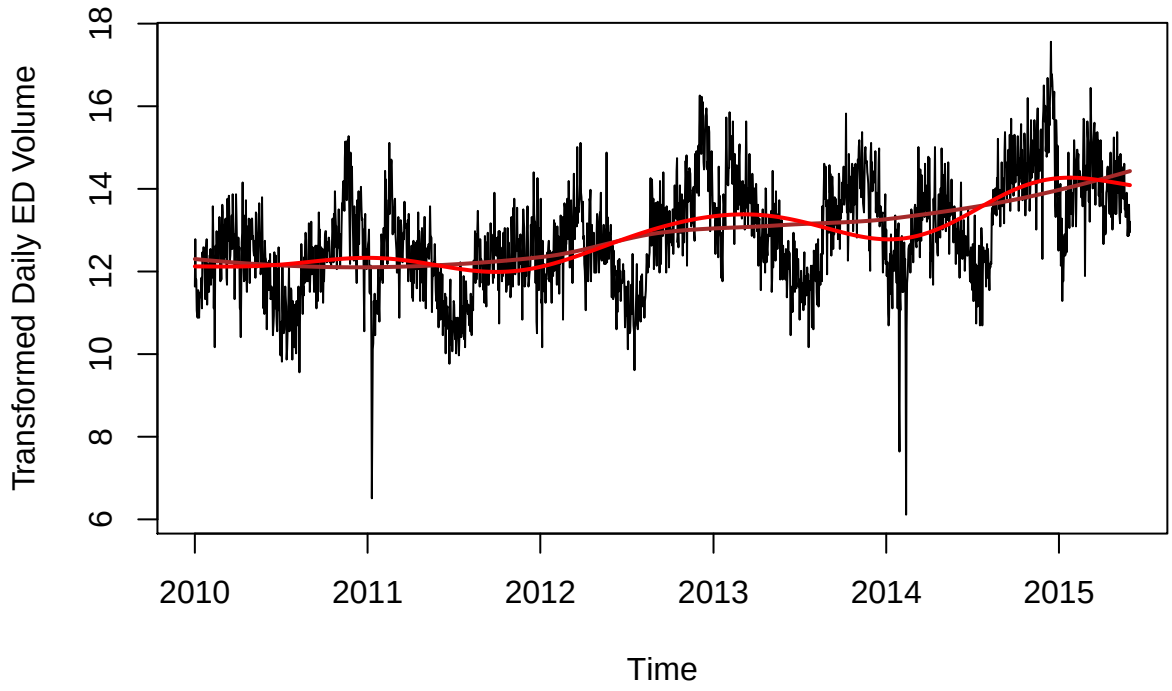
TREND AND SEASONALITY ESTIMATION

```
time.pts = c(1:length(Volume))
time.pts = c(time.pts - min(time.pts))/max(time.pts)
## Trend Estimation: Is there a trend?
## Local Polynomial Trend Estimation
loc.fit = loess(Volume.tr~time.pts)
vol.fit.loc = fitted(loc.fit)
## Splines Trend Estimation
gam.fit = gam(Volume.tr~s(time.pts))
summary(gam.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Volume.tr ~ s(time.pts)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.8577     0.0244     527   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df    F p-value
## s(time.pts) 8.63   8.96 93.1  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.296   Deviance explained = 29.9%
## GCV =  1.184  Scale est. = 1.1782    n = 1977
```

```r
vol.fit.gam = fitted(gam.fit)

## Is there a trend?
plot(dates,sqrt(Volume+3/8), type='l',ylab='Transformed Daily ED Volume',xlab='Time')
#ggplot(edvoldata, aes(dates, sqrt(Volume+3/8))) + geom_line() + xlab("Time") + ylab("Transformed Daily
lines(dates,vol.fit.loc,lwd=2,col="brown") #can not be used with ggplot
lines(dates,vol.fit.gam,lwd=2,col="red")
```



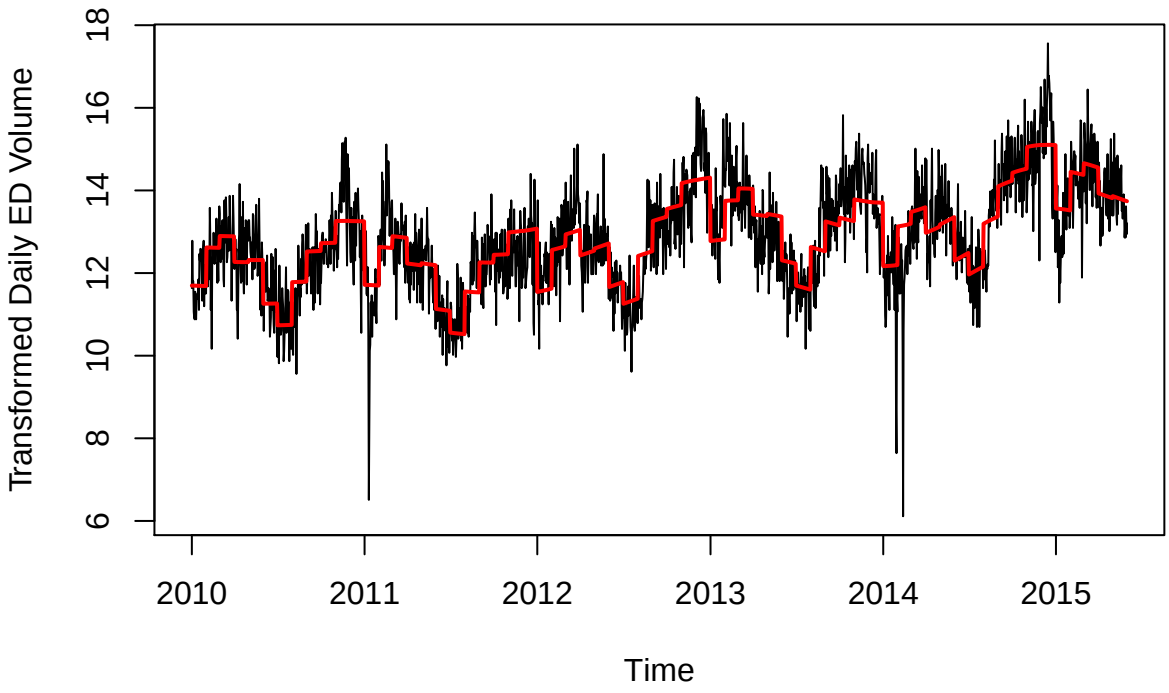
Model Trend + Monthly Seasonality

Using nonparametric trend and linear regression seasonality

```
month = as.factor(format(dates,"%b"))
gam.fit.seastr.1 = gam(Volume.tr~s(time.pts)+month)
summary(gam.fit.seastr.1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Volume.tr ~ s(time.pts) + month
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.8635     0.0620  207.47  < 2e-16 ***
## monthAug     -0.5007     0.0919   -5.45  5.8e-08 ***
## monthDec      0.9364     0.0913   10.26  < 2e-16 ***
## monthFeb      0.3349     0.0884    3.79  0.00016 ***
## monthJan     -0.5991     0.0869   -6.90  7.1e-12 ***
## monthJul     -1.5349     0.0917  -16.74  < 2e-16 ***
## monthJun     -1.0062     0.0922  -10.91  < 2e-16 ***
## monthMar      0.6212     0.0859    7.23  6.8e-13 ***
## monthMay      0.0515     0.0859    0.60  0.54871
## monthNov      0.9413     0.0924   10.19  < 2e-16 ***
## monthOct      0.4149     0.0919    4.52  6.7e-06 ***
## monthSep      0.2266     0.0928    2.44  0.01466 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df   F p-value
## s(time.pts) 8.78   8.99 141  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.598   Deviance explained = 60.2%
## GCV = 0.68006  Scale est. = 0.67292   n = 1977
```

```
vol.fit.gam.seastr.1 = fitted(gam.fit.seastr.1)
plot(dates,sqrt(Volume+3/8), type='l',ylab='Transformed Daily ED Volume',xlab='Time')
# ggplot(edvoldata, aes(dates, sqrt(Volume+3/8))) + geom_line() + xlab("Time") + ylab("Transformed Dail
lines(dates,vol.fit.gam.seastr.1,lwd=2,col="red")
```



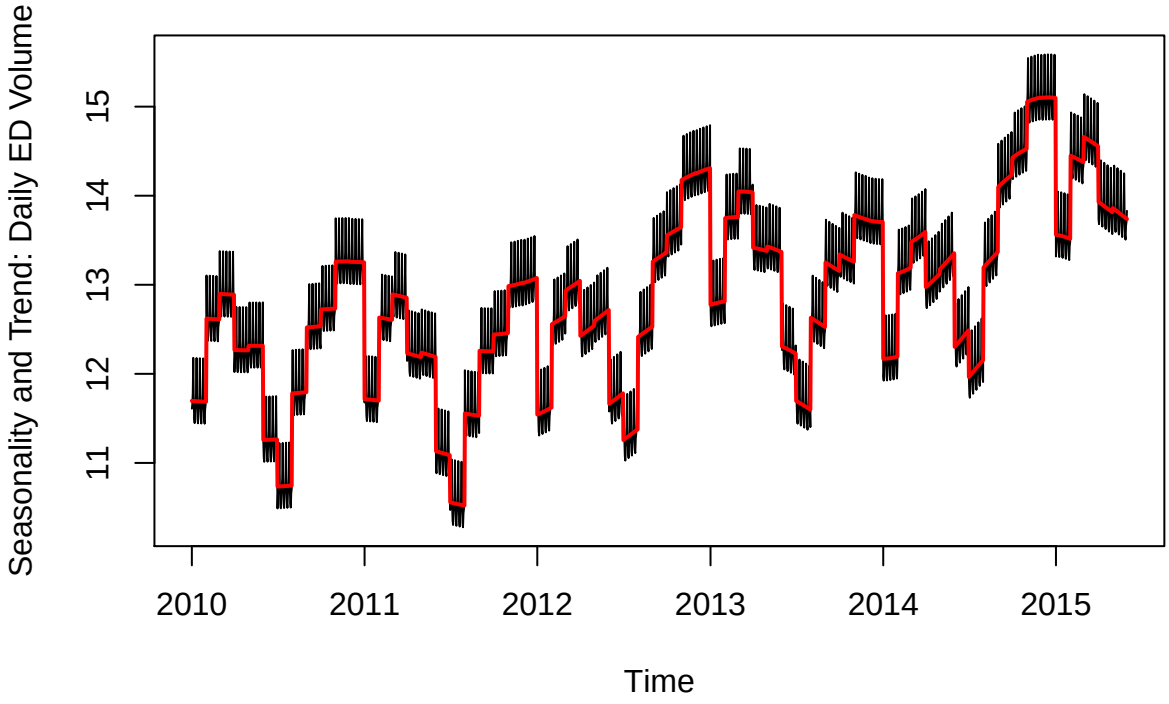
Add day-of-the-week seasonality, see if it adds any accuracy

```
week = as.factor(weekdays(dates))
gam.fit.seastr.2 = gam(Volume.tr~s(time.pts)+month+week)
summary(gam.fit.seastr.2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Volume.tr ~ s(time.pts) + month + week
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12.7777     0.0738  173.06  < 2e-16 ***
## monthAug       -0.5006     0.0885   -5.66  1.8e-08 ***
## monthDec        0.9327     0.0879   10.61  < 2e-16 ***
## monthFeb        0.3359     0.0851    3.95  8.2e-05 ***
## monthJan       -0.5977     0.0836   -7.15  1.3e-12 ***
## monthJul       -1.5353     0.0883  -17.39  < 2e-16 ***
## monthJun       -1.0055     0.0888  -11.32  < 2e-16 ***
## monthMar        0.6182     0.0827    7.47  1.2e-13 ***
```

```
## monthMay        0.0527     0.0827    0.64   0.5240
## monthNov        0.9416     0.0890   10.58  < 2e-16 ***
## monthOct        0.4156     0.0885    4.70  2.8e-06 ***
## monthSep        0.2239     0.0893    2.51   0.0123 *
## weekMonday      0.5717     0.0665    8.60  < 2e-16 ***
## weekSaturday    0.0459     0.0664    0.69   0.4897
## weekSunday      0.1754     0.0664    2.64   0.0083 **
## weekThursday   -0.1603     0.0665   -2.41   0.0160 *
## weekTuesday     0.0810     0.0665    1.22   0.2232
## weekWednesday  -0.1105     0.0665   -1.66   0.0966 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df   F p-value
## s(time.pts) 8.79   8.99 152  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.627   Deviance explained = 63.2%
## GCV = 0.63263  Scale est. = 0.62405   n = 1977
```

```
vol.fit.gam.seastr.2 = fitted(gam.fit.seastr.2)
## Compare the two fits: with & without day-of-the-week seasonality
plot(dates,vol.fit.gam.seastr.2, type='l',ylab="Seasonality and Trend: Daily ED Volume",xlab='Time')
#ggplot(edvoldata, aes(dates, vol.fit.gam.seastr.2)) + geom_line() + xlab("Time") + ylab("Seasonality a
lines(dates,vol.fit.gam.seastr.1,lwd=2,col="red")
```

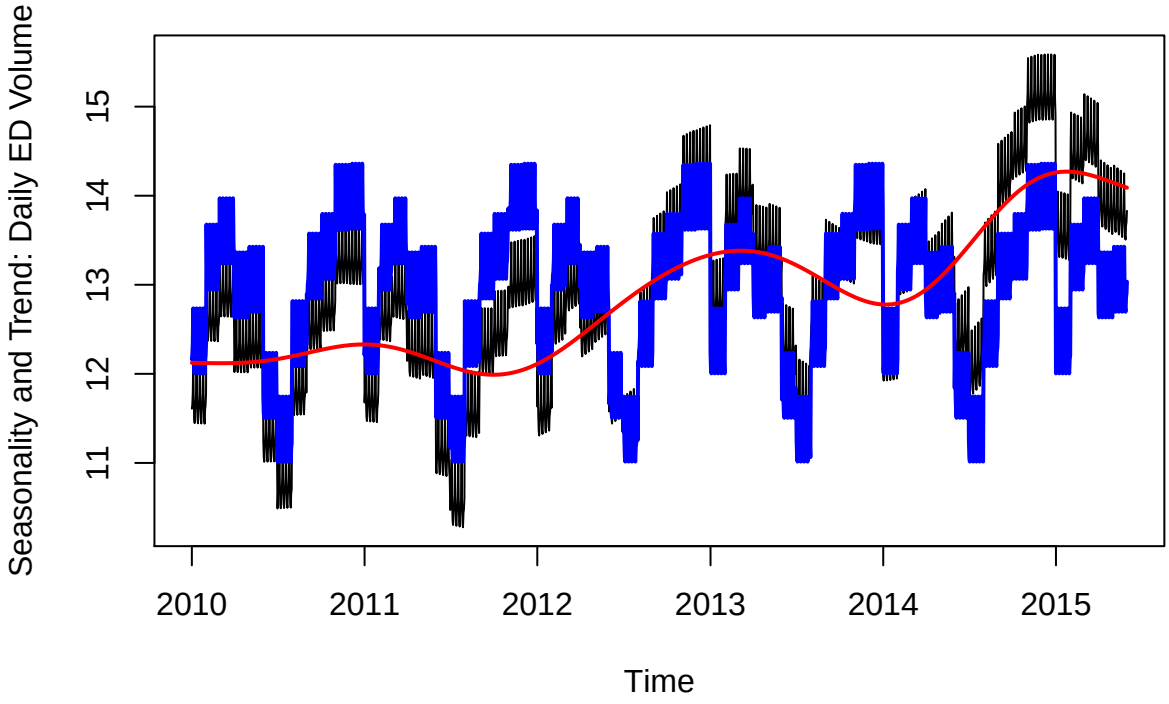


Does the addition of seasonality of day of the week adds predictive power?

```
lm.fit.seastr.1 = lm(Volume.tr~month)
lm.fit.seastr.2 = lm(Volume.tr~month+week)
anova(lm.fit.seastr.1,lm.fit.seastr.2)
```

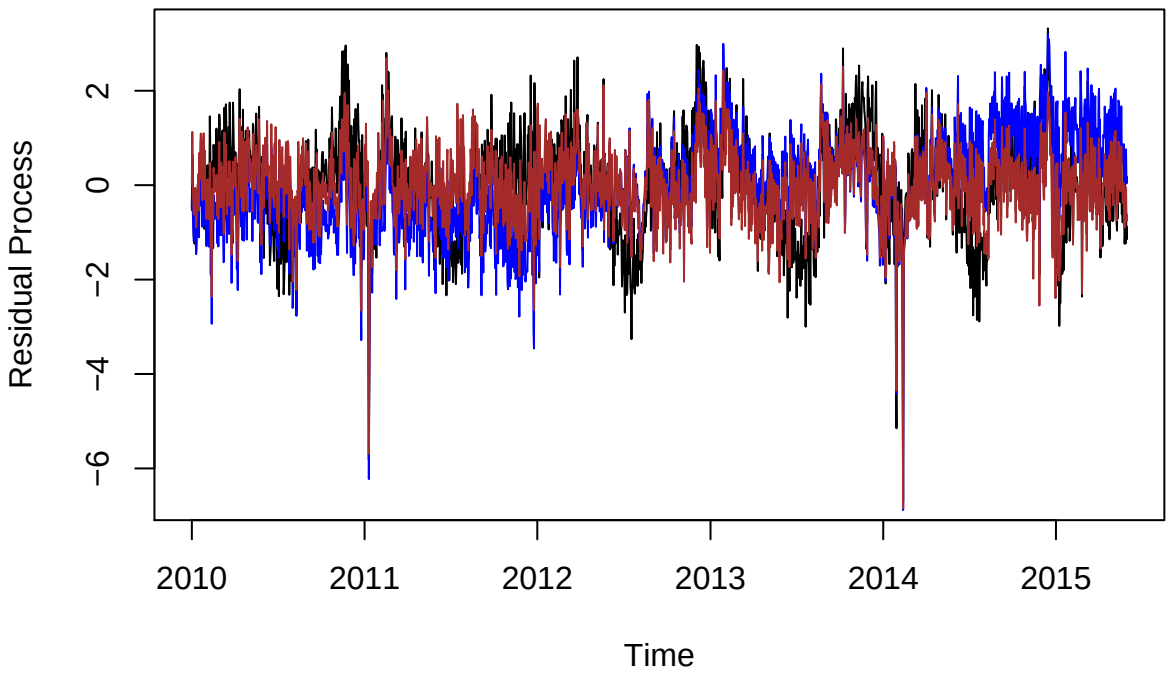
```
## Analysis of Variance Table
##
## Model 1: Volume.tr ~ month
## Model 2: Volume.tr ~ month + week
##   Res.Df  RSS Df Sum of Sq    F Pr(>F)
## 1   1965 2170
## 2   1959 2071  6      98.8 15.6 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vol.fit.lm.seastr.2 = fitted(lm.fit.seastr.2)
## Compare with & without trend
plot(dates,vol.fit.gam.seastr.2, type='l',ylab="Seasonality and Trend: Daily ED Volume",xlab='Time')
#ggplot(edvoldata, aes(dates, vol.fit.gam.seastr.2)) + geom_line() + xlab("Time") + ylab("Seasonality a
lines(dates,vol.fit.lm.seastr.2,lwd=2,col="blue")
lines(dates,vol.fit.gam,lwd=2,col="red")
```

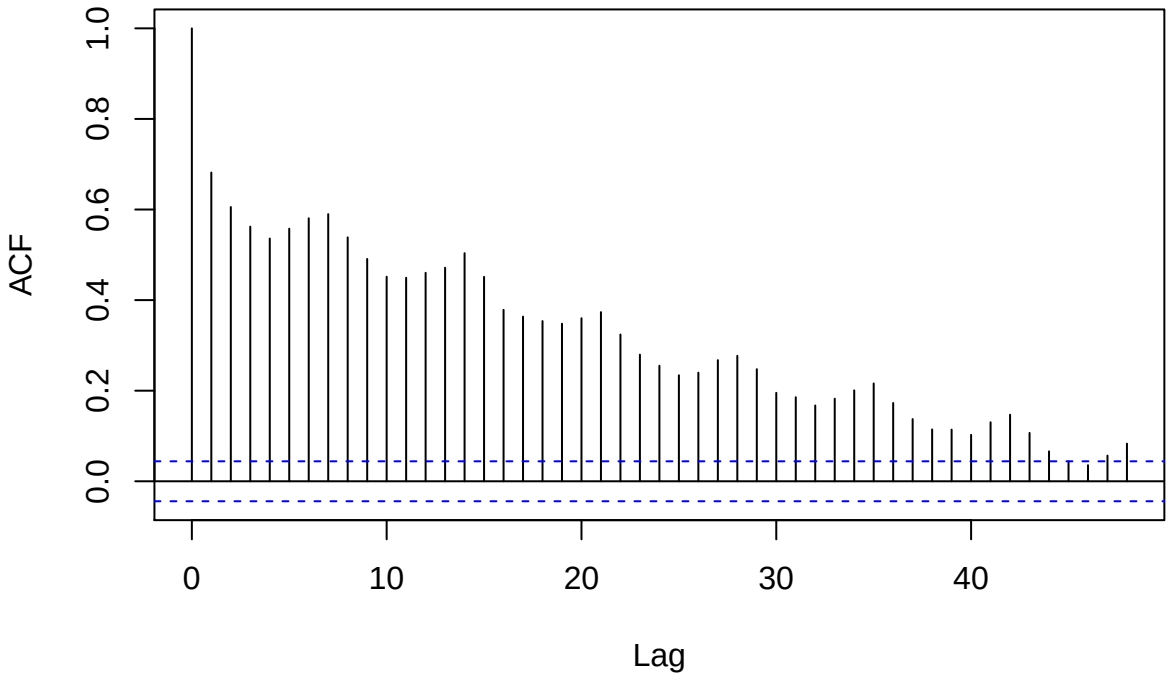


STATIONARITY TEST FOR RESIDUAL PROCESS

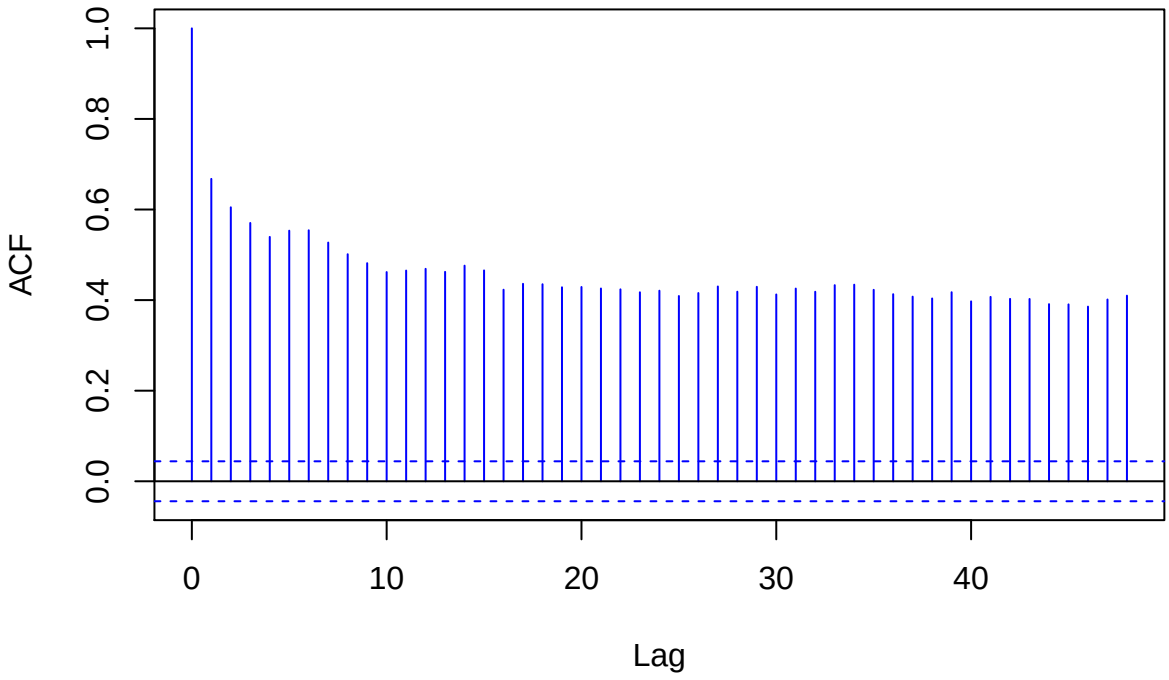
```
## Residual Process: Trend Removal
resid.1 = Volume.tr-vol.fit.gam
## Residual Process: Stationarity Removal
resid.2 = Volume.tr-vol.fit.lm.seastr.2
## Residual Process: Trend & Stationarity Removal
resid.3 = Volume.tr-vol.fit.gam.seastr.2
y.min = min(c(resid.1,resid.2,resid.3))
y.max = max(c(resid.1,resid.2,resid.3))

#ggplot(edvoldata, aes(dates, resid.1),ymin=y.min,ymax=y.max) + geom_line() + xlab("Time") + ylab("Resi
plot(dates,resid.1, type='l',ylab="Residual Process",xlab='Time')
lines(dates,resid.2,col="blue")
lines(dates,resid.3,col="brown")
legend(2012,-3.5,legend=c("Trend","Season","Trend+Season"),lty = 1, col=c("black","blue","brown"))
```



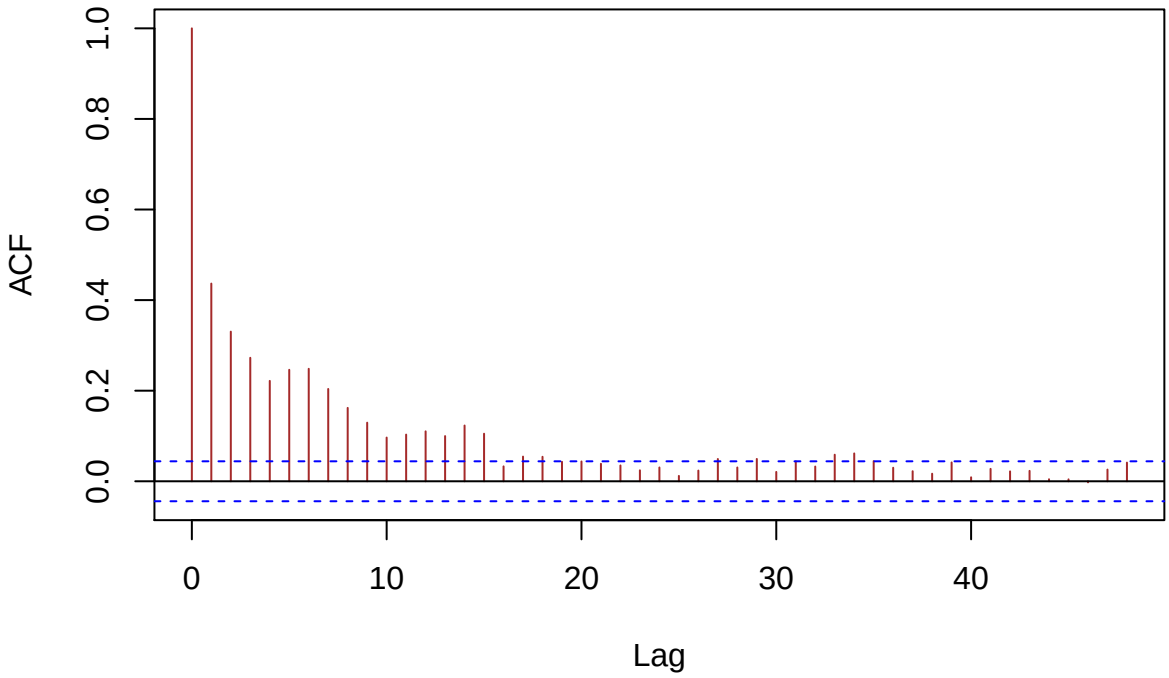

```
acf(resid.1,lag.max=12*4,main="")
```




```
acf(resid.2,lag.max=12*4,main="",col="blue")
```




```
acf(resid.3,lag.max=12*4,main="",col="brown")
```


ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
Lag