# STA457A1

Yuhan Hu

2020/5/20

## Theory

### Q1

#### Part A

ACVF of $Z$

$cov(Z_t, Z_{t+h}) = cov(aX_t + bY_t, aX_{t+h} + bY_{t+h})$

since we have $cov(X + Y, Z) = cov(X, Z) + cov(Y, Z)$

$cov(aX_t + bY_t, aX_{t+h} + bY_{t+h}) = cov(aX_t, aX_{t+h} + bY_{t+h}) + cov(bY_t, aX_{t+h} + bY_{t+h})$

$= cov(aX_t, aX_{t+h}) + cov(aX_t, bY_{t+h}) + cov(bY_t, aX_{t+h}) + cov(bY_t, bY_{t+h})$

where $cov(aX_t, bY_{t+h}) = cov(bY_t, aX_{t+h}) = 0$ due to independence

now we have $cov(aX_t, aX_{t+h}) + cov(bY_t, bY_{t+h}) = a^2 cov(X_t, X_{t+h}) + b^2 cov(Y_t, Y_{t+h})$

and $cov(X_t, X_{t+h}) = \gamma_X(h)$,$cov(Y_t, Y_{t+h}) = \gamma_Y(h)$

Therefore, $\gamma_Z(h) = a^2\gamma_X(h) + b^2\gamma_Y(h)$

#### Part B

$V(t) = \sum_{j=0}^{p} a_j X_{t-j}$

$E(V_t) = \sum a_j E(X_{t-j}) = 0$ since $E(X_{t-j}) = 0$

$\gamma_V(h) = cov(V_t, V_{t+h}) = cov(\sum_{j=0}^{p} a_j X_{t-j}, \sum_{k=0}^{p} a_k X_{t+h-k})$

$= \sum_{j=0}^{p}\sum_{k=0}^{p} a_j a_k cov(X_{t-j}, X_{t+h-k}) = \sum_{j=0}^{p}\sum_{k=0}^{p} a_j a_k \gamma_X(h - k + j)$

### Q2

$E[\hat{\gamma}(h)] = E[\frac{1}{n}\sum_{t=1}^{n-h}(X_{t+h}X_t)] = \frac{1}{n}E[\sum_{t=1}^{n-h}(X_{t+h}X_t)]$ since $E[aX] = aE[X]$

$= \frac{1}{n}E[X_{1+h}X_1 + X_{2+h}X_2 + .... + X_nX_{n-h}]$

$= \frac{1}{n}E[X_{1+h}X_1] + E[X_{2+h}X_2] + ... + X_nX_{n-h}$ since $E[X + Y] = E[X] + E[Y]$

Refer to the $Hint$, we know that ACVF of $X$ is $\gamma_X(s, t) = cov(X_s, X_t) = min(s, t)$

also, $cov(X_s, X_t) = E(X_tX_s) - E(X_t)E(X_s)$

So we want to find $E(X_t)E(X_s)$

given $X_0 = 0$, $X_t = X_{t-1} + W_t$, then we have $X_1 = X_0 + W_1 = W_1$,$X_2 = X_1 + W_2 = W_1 + W_2$,$\ldots$,$X_t = W_1 + ... + W_t$

then $E[X_t] = E[W_t] + E[W_{t-1}] + ... + E[W_1]$, and we have $W_t \sim WN(0,1)$, therefore $E[X_t] = 0$

since $E[X_t] = 0$, $E[X_t]E[X_s] = 0$,$cov(X_s, X_t) = E(X_t X_s) - 0 = E(X_t X_s)$

so $E[X_t X_s] = min(s,t)$

Therefore $E[\hat{\gamma}(h)] = \frac{1}{n}[1 + 2 + ... + (n-h)] = \frac{1}{n} * \frac{(n-h+1)(n-h)}{2} = \frac{(n-h)(n-h+1)}{2n}$

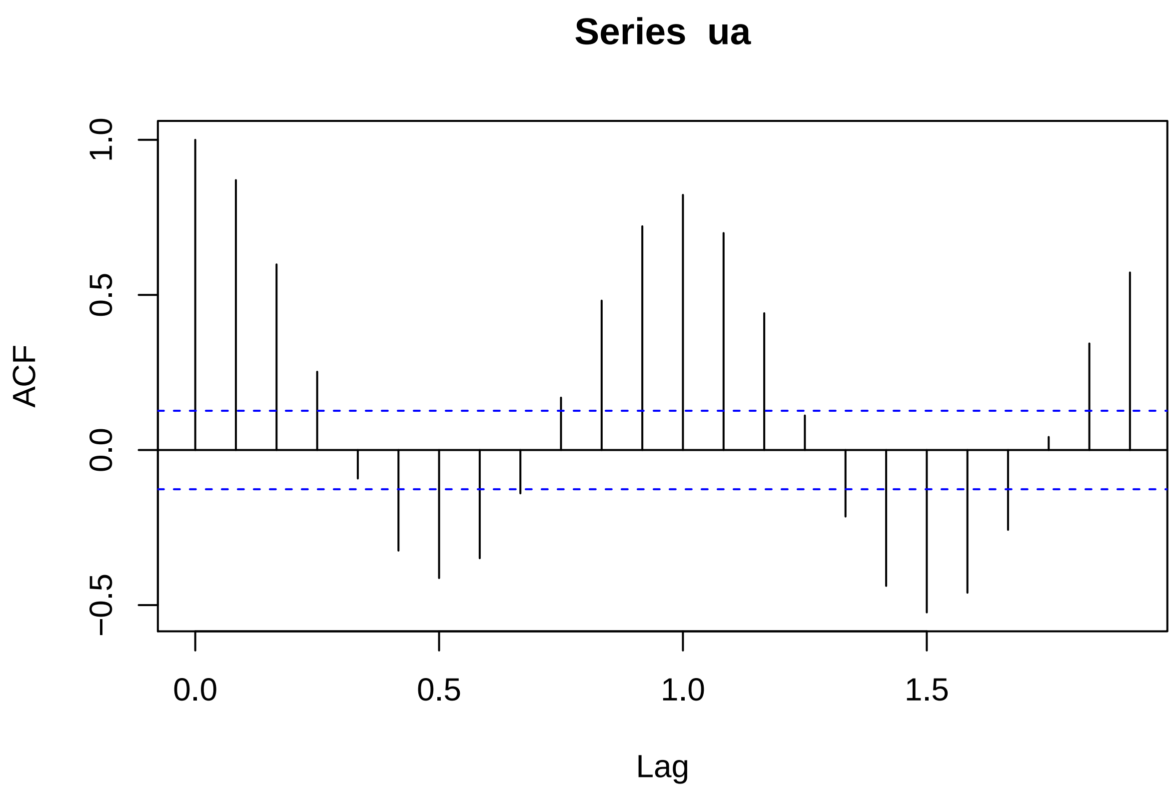
## Practice

```r
#student number:1001311626, last digit 6, use Goods-producing sector data, second last digit is even, s
ua = get_cansim_vector( "v2057813", start_time = "2000-01-01", end_time = "2019-12-01") %>%
pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)
```

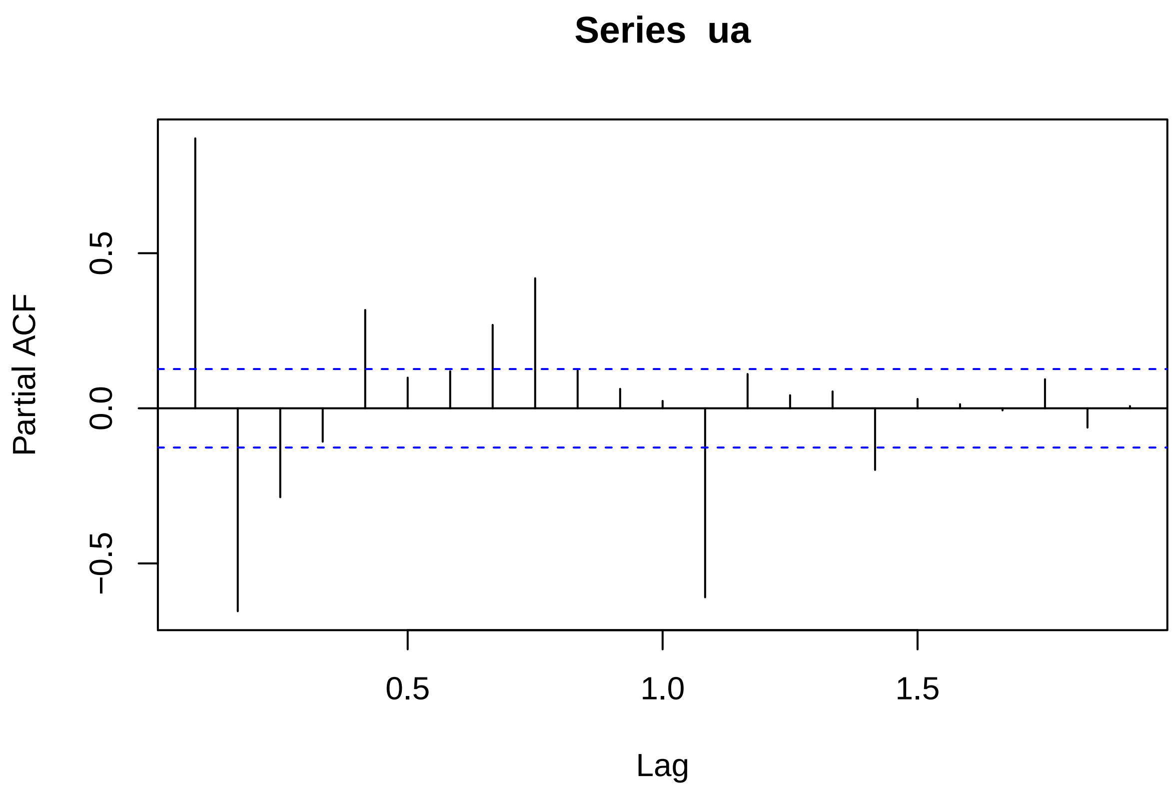
### Q1

#### ACF

```r
acf(ua)
```




```r
pacf(ua)
```




There are periodic changes shown on ACF plot, therefore the data is seasonal.

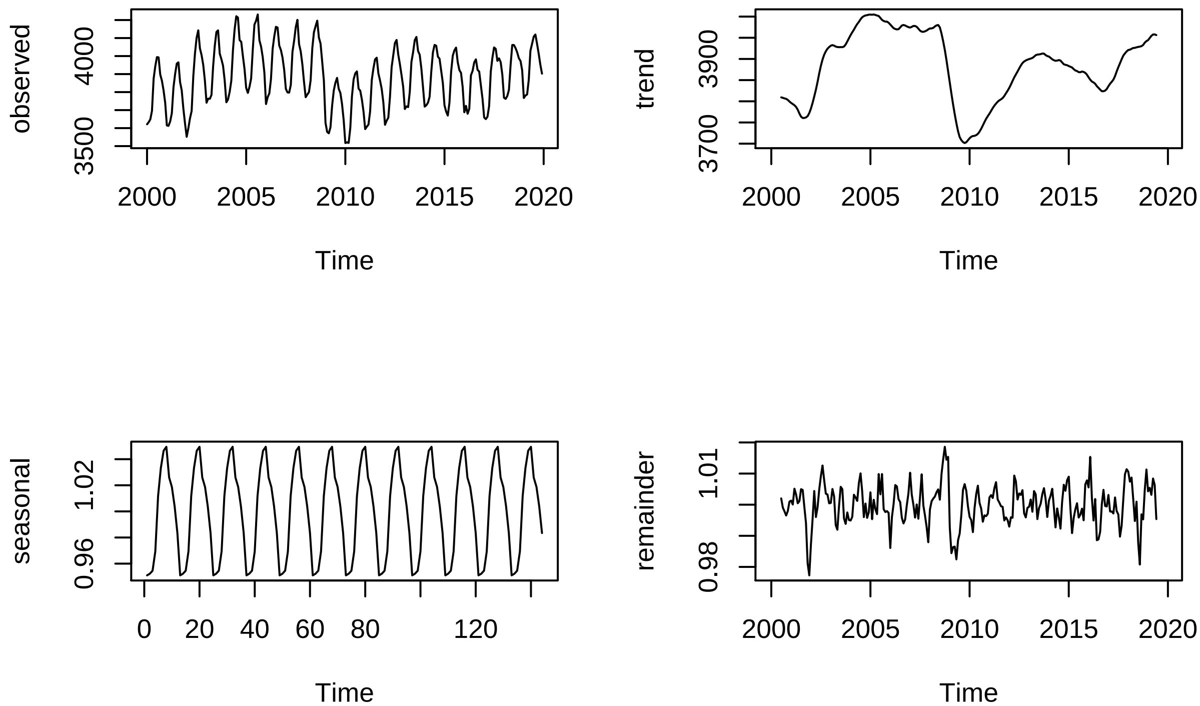
Since both ACF and PACF does not cut off, the data is trended.

Since the data is trended, the data is not stationary.

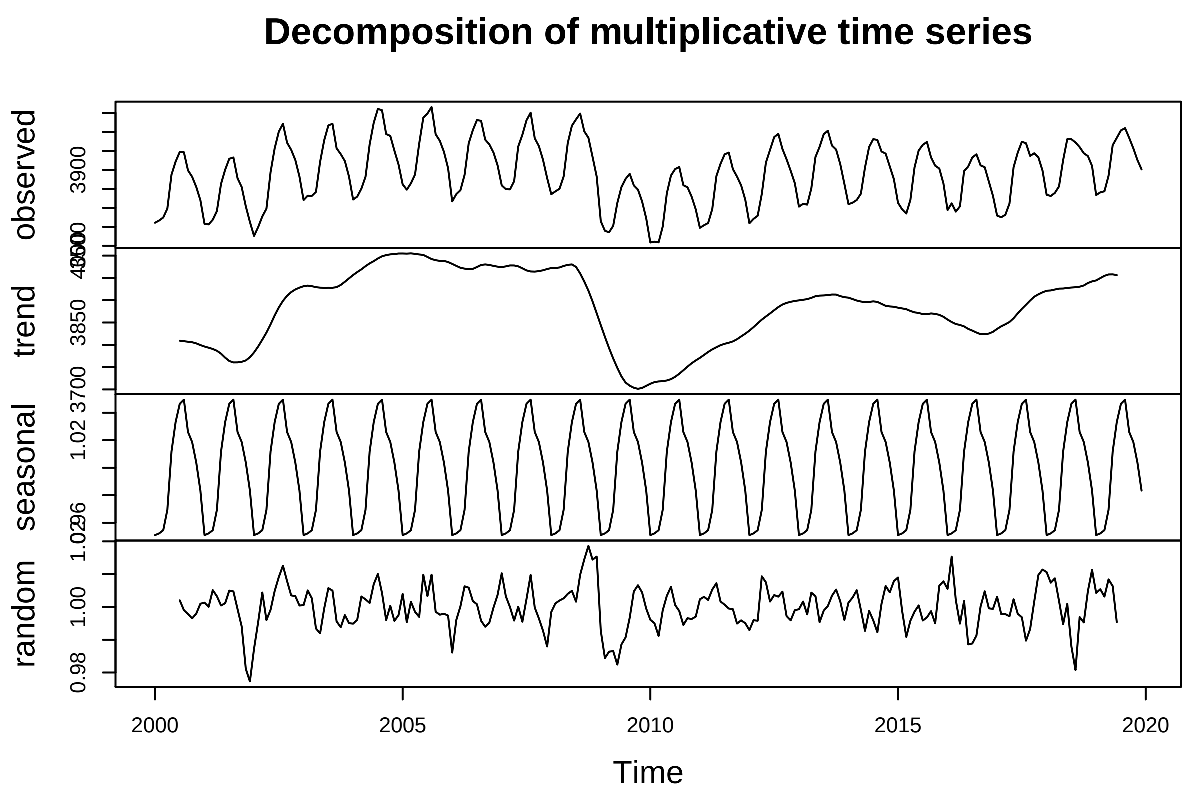
### Q2

```
#trend
trend_ua <- ma(ua,order = 12,centre=T)

#find seasonal*remainder
detrend_ua <- ua/trend_ua

#seasonality
m_ua = t(matrix(data = detrend_ua, nrow = 12))
seasonal_ua = colMeans(m_ua, na.rm = T)

#Remainder/random noise
remainder <- ua/(trend_ua*seasonal_ua)

par(mfrow=c(2,2))

plot(as.ts(ua),ylab='observed')
plot(as.ts(trend_ua),ylab='trend')
```

```r
plot(as.ts(rep(seasonal_ua,12)),ylab='seasonal')
plot(as.ts(remainder),ylab='remainder')
```



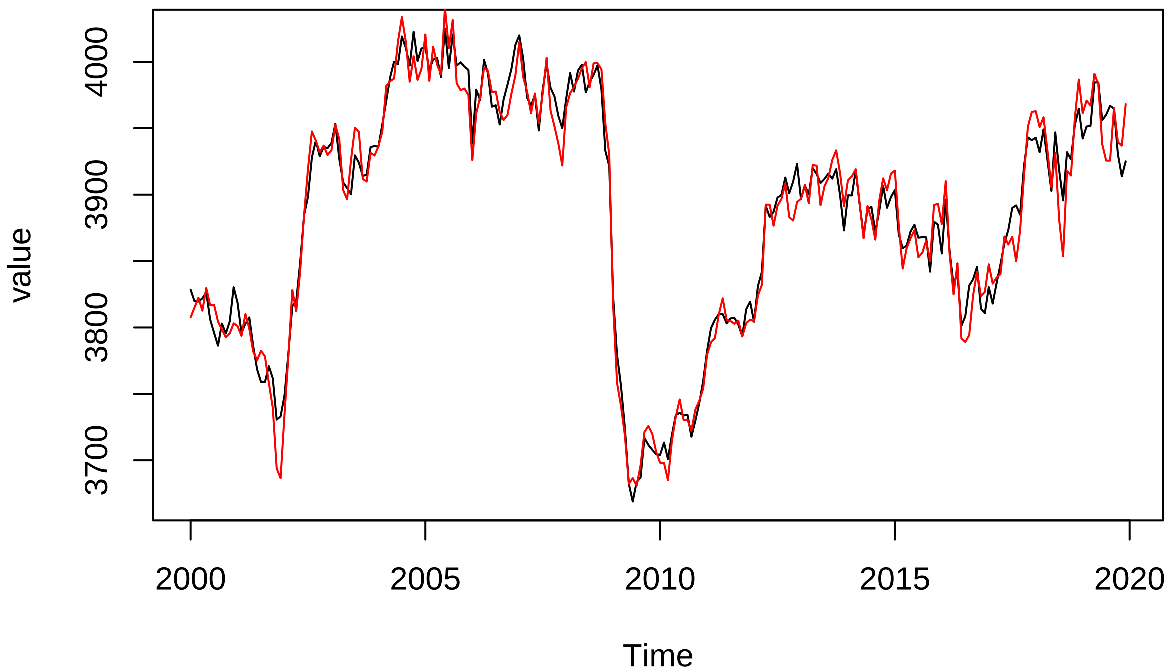
```
built_in_decomposition <- decompose(ua,type='multiplicative')
plot(built_in_decomposition)
```




Reference: https://anomaly.io/seasonal-trend-decomposition-in-r/index.html

## Q3

```
#import seasonally adjusted data
seasonal_adjusted_online = get_cansim_vector( "v2057604", start_time = "2000-01-01", end_time = "2019-1
pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)

#own seasonally adjusted data
seasonal_adjusted_ua <- ua/seasonal_ua
{plot(seasonal_adjusted_online,ylab='value')
lines(seasonal_adjusted_ua,col='red')}
```



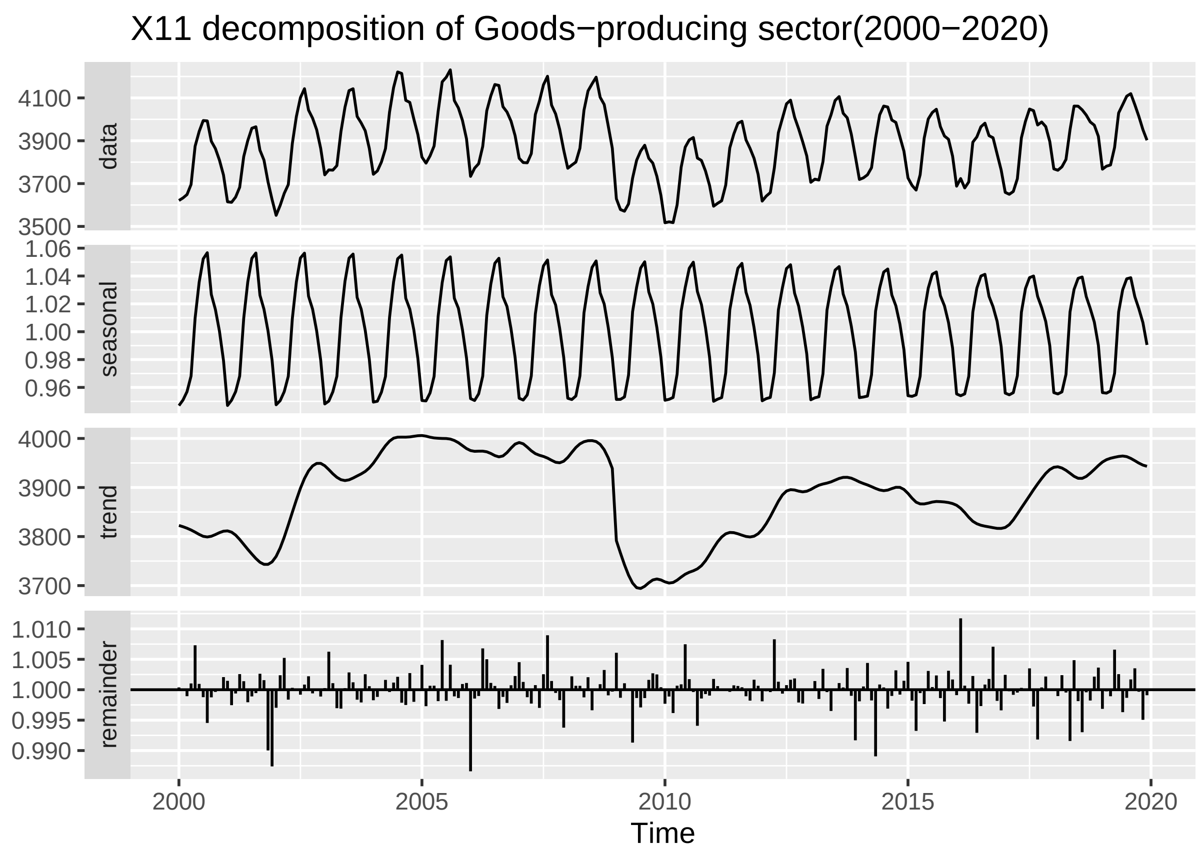

```
mae1 <- mae(seasonal_adjusted_online,seasonal_adjusted_ua)
```

Mean absolute error (MAE) between the two versions (StaCan's and mine) of seasonally adjusted data is 11.2835086

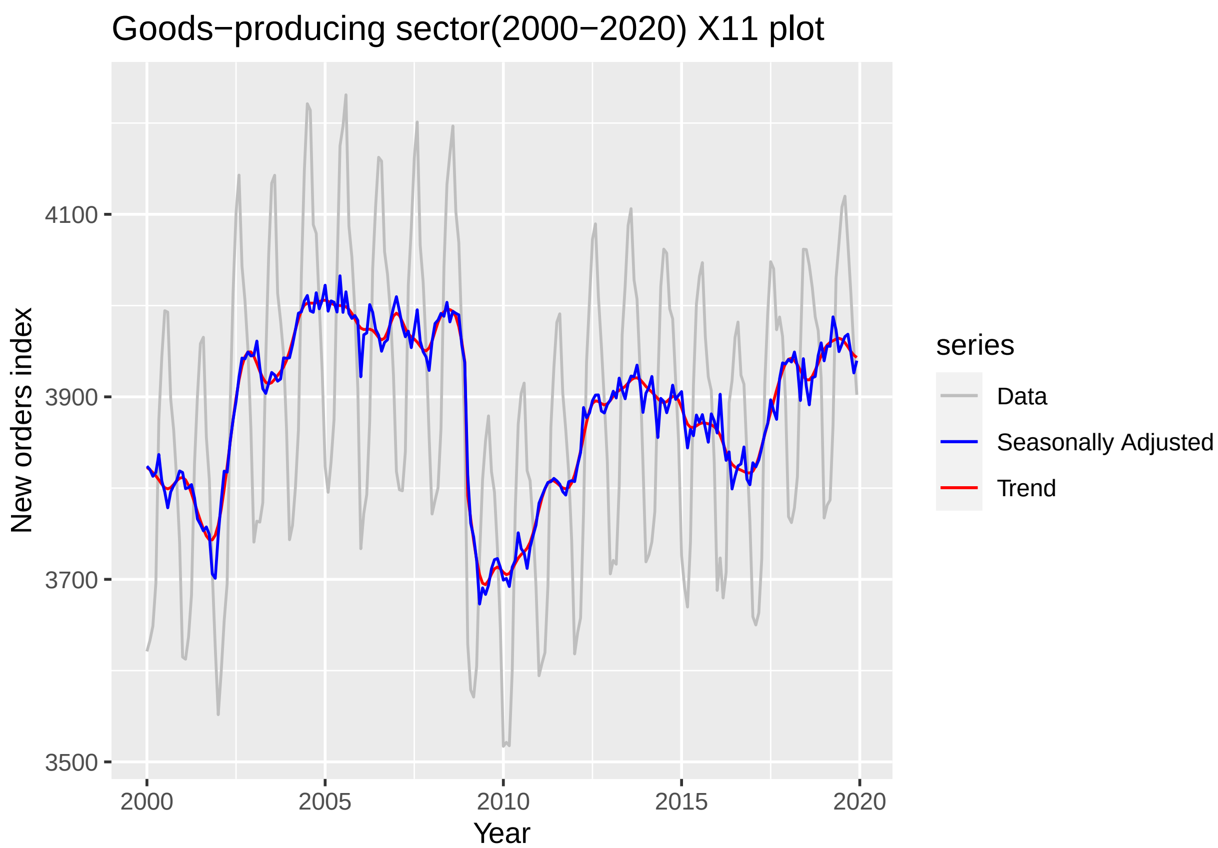
## Q4

### X11

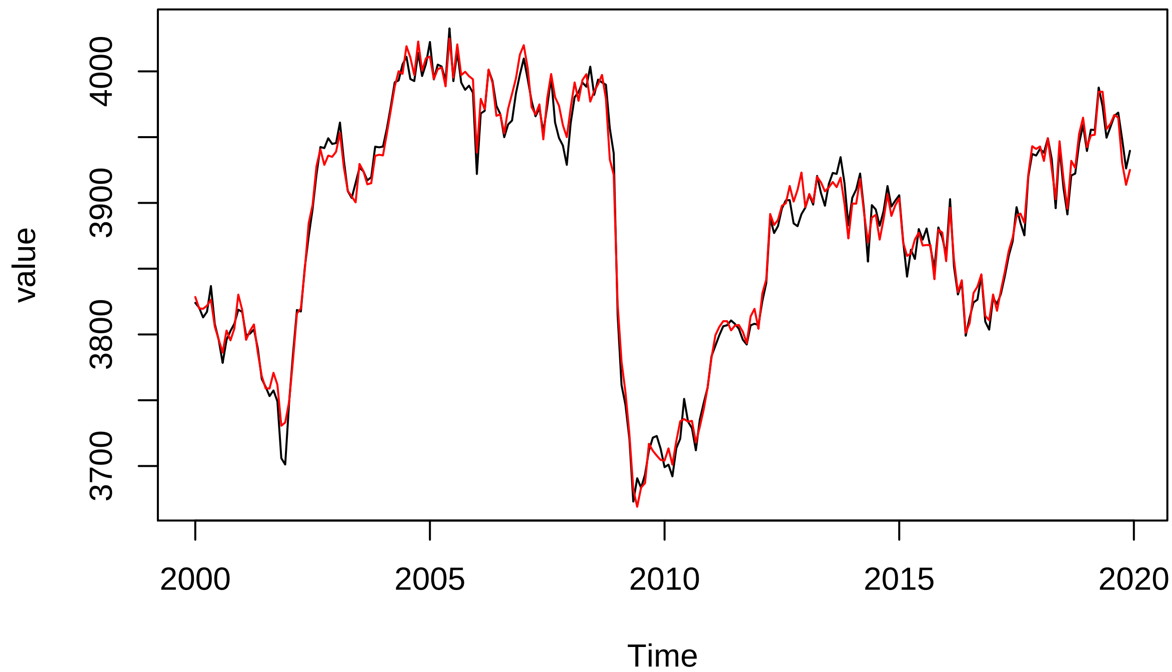
```
#X11
ua %>% seas(x11="") -> X11fit
autoplot(X11fit) +
  ggtitle("X11 decomposition of Goods-producing sector(2000-2020)")
```




```
autoplot(ua, series="Data") +
  autolayer(trendcycle(X11fit), series="Trend") +
  autolayer(seasadj(X11fit), series="Seasonally Adjusted") +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Goods-producing sector(2000-2020) X11 plot") +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```




```
{plot(seasadj(X11fit),ylab='value')
 lines(seasonal_adjusted_online,col='red')}
```



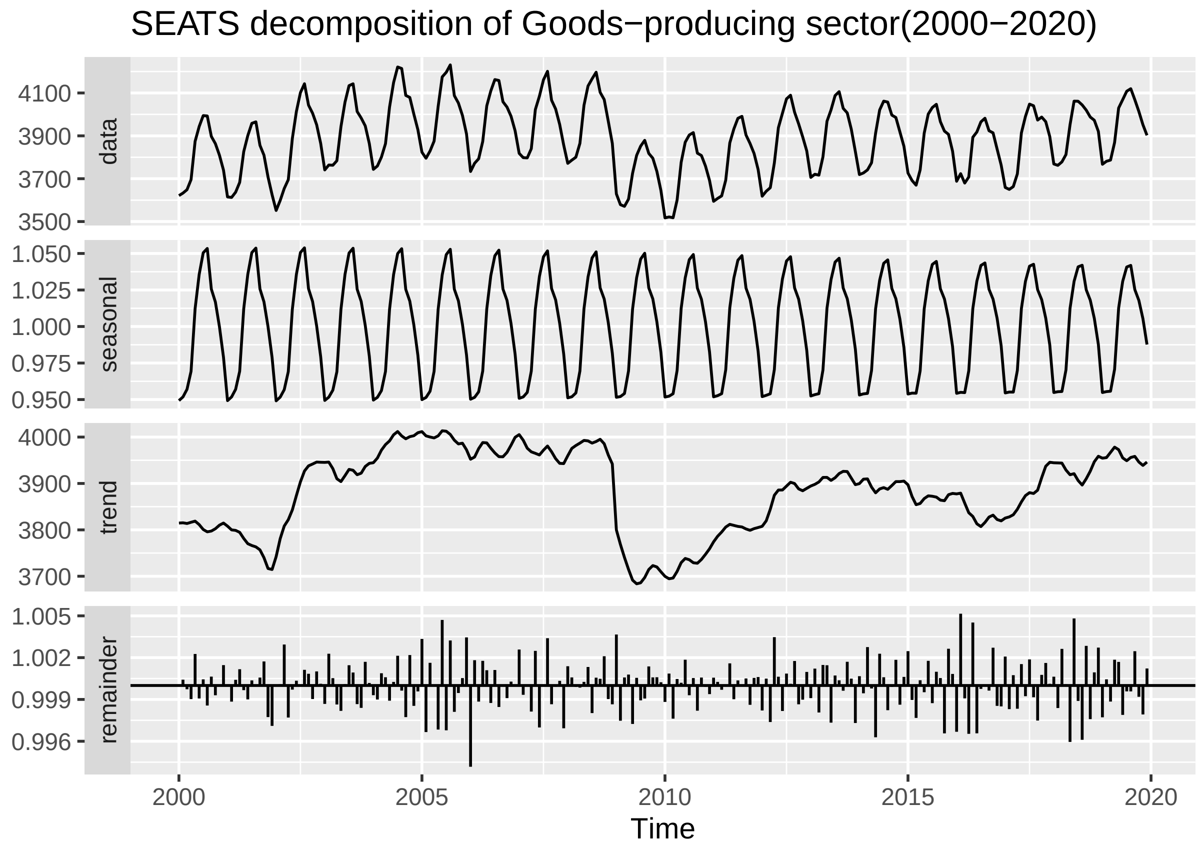

```
mae2 <- mae(seasadj(X11fit),seasonal_adjusted_online)
```

Mean absolute error (MAE) between the two versions (StaCan's and X11) of seasonally adjusted data is 6.9825505

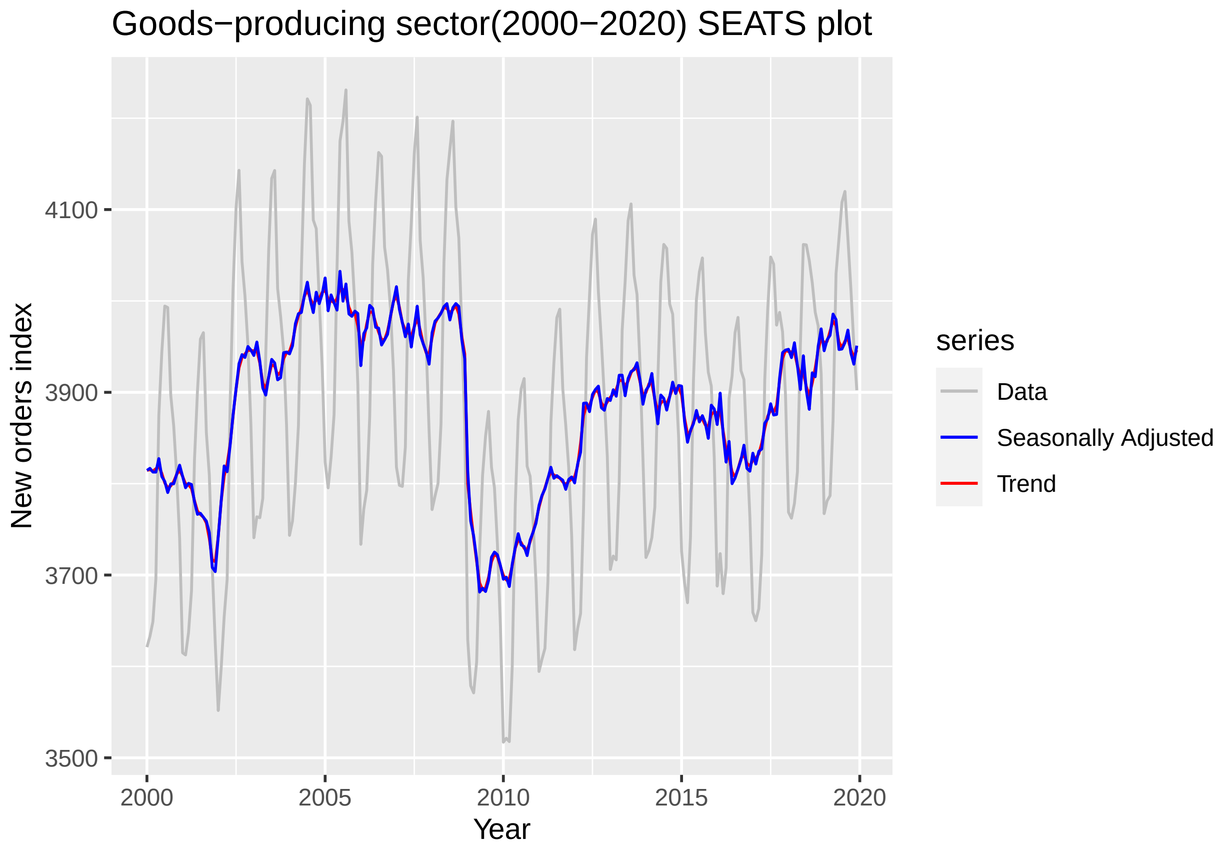
Reference: https://otexts.com/fpp2/x11.html

**SEATS**

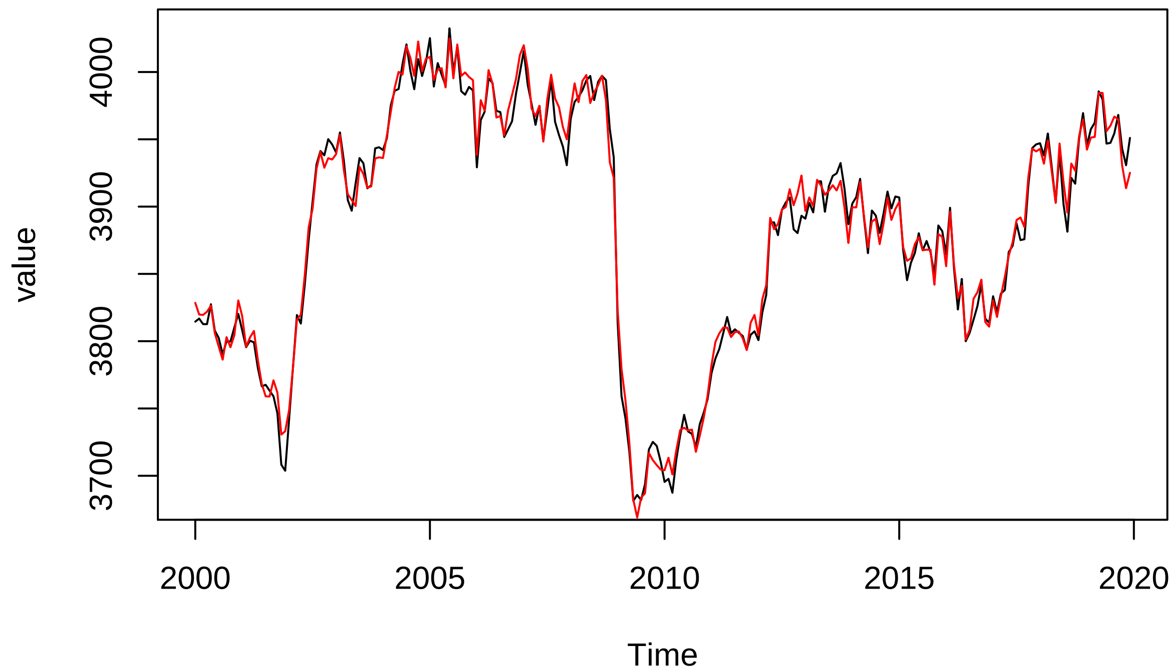
```
#SEATS
ua %>% seas() -> SEATSfit
autoplot(SEATSfit) +
  ggtitle("SEATS decomposition of Goods-producing sector(2000-2020)")
```




```r
autoplot(ua, series="Data") +
  autolayer(trendcycle(SEATSfit), series="Trend") +
  autolayer(seasadj(SEATSfit), series="Seasonally Adjusted") +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Goods-producing sector(2000-2020) SEATS plot") +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```




```
{plot(seasadj(SEATSfit),ylab='value')
 lines(seasonal_adjusted_online,col='red')}
```



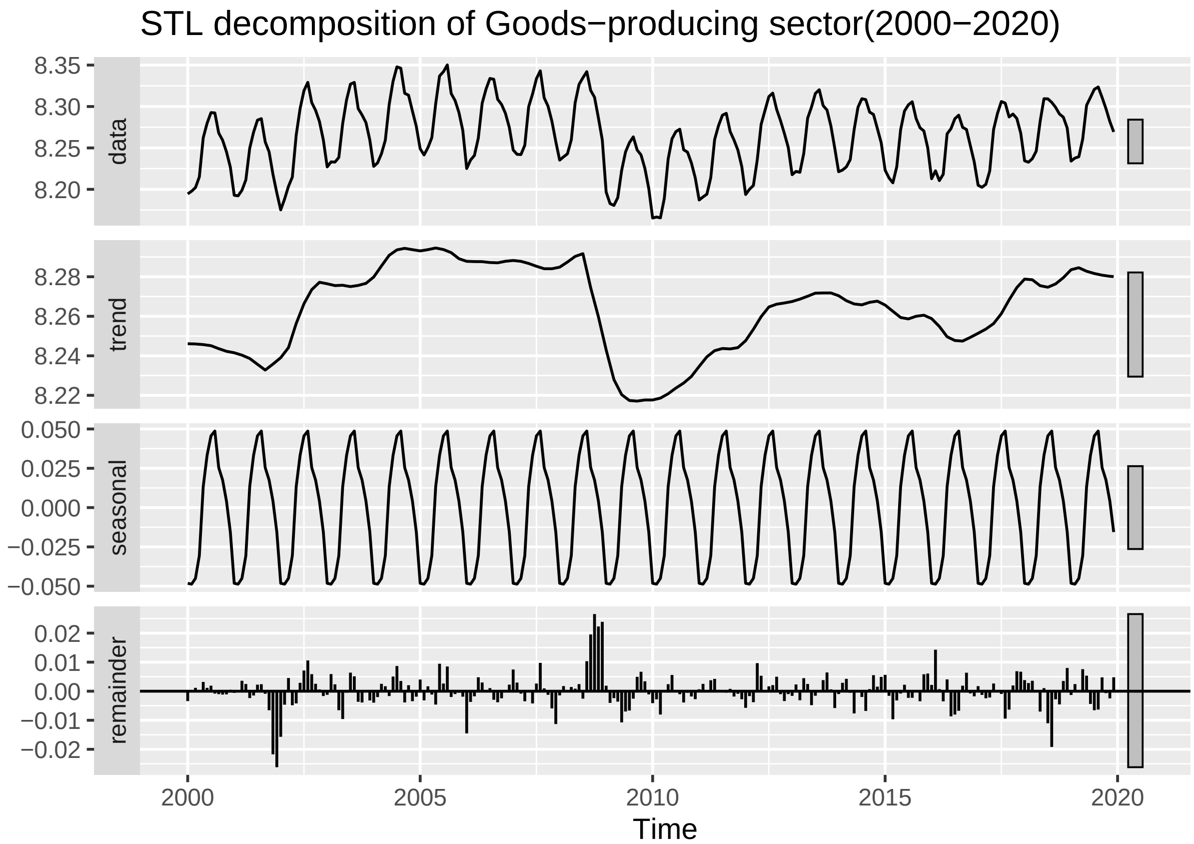

```
mae3 <- mae(seasadj(SEATSfit),seasonal_adjusted_online)
```

Mean absolute error (MAE) between the two versions (StaCan's and X11) of seasonally adjusted data is 7.3839485

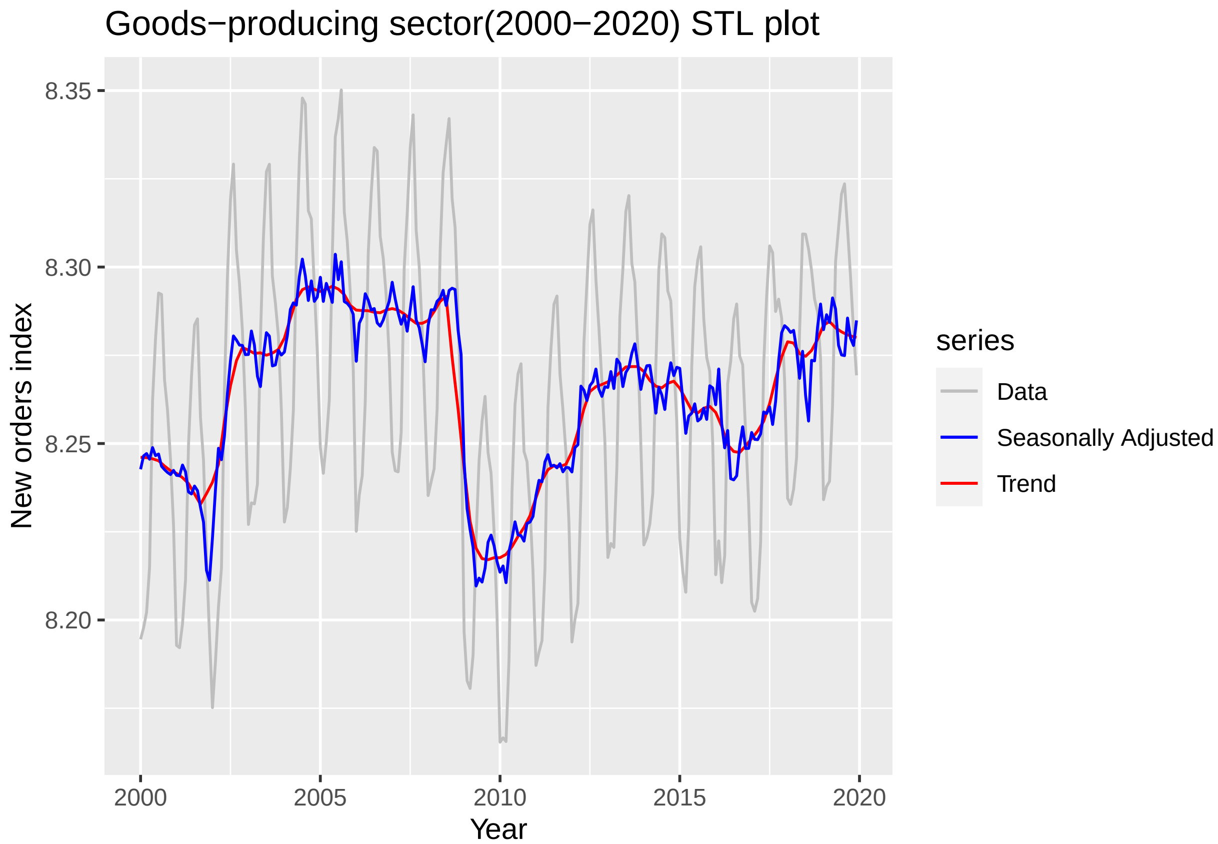
Reference:https://otexts.com/fpp2/seats.html

### STL

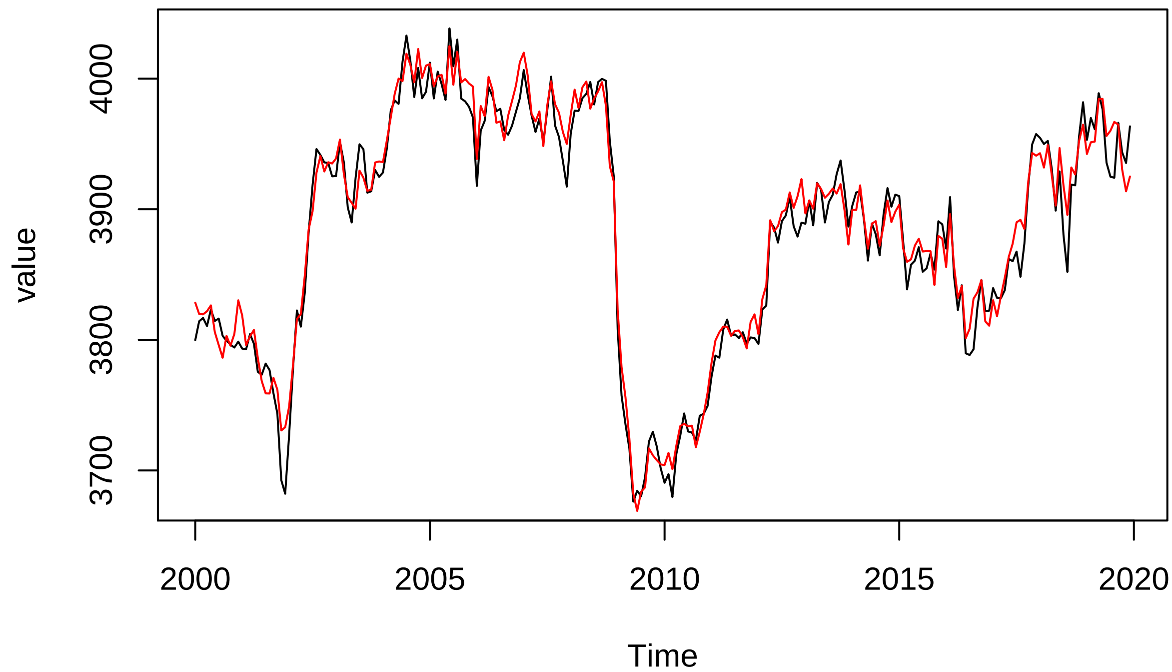
```
#STL
log_ua <- log(ua)
log_ua %>% stl(t.window=12, s.window="periodic", robust=TRUE) -> STLfit
autoplot(STLfit) +
  ggtitle("STL decomposition of Goods-producing sector(2000-2020)")
```




```r
autoplot(log_ua, series="Data") +
  autolayer(trendcycle(STLfit), series="Trend") +
  autolayer(seasadj(STLfit), series="Seasonally Adjusted") +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Goods-producing sector(2000-2020) STL plot") +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```




```
{plot(exp(seasadj(STLfit)),ylab = 'value')
 lines(seasonal_adjusted_online,col='red')}
```




```
mae4 <- mae(exp(seasadj(STLfit)),seasonal_adjusted_online)
```

Mean absolute error (MAE) between the two versions (StaCan's and X11) of seasonally adjusted data is 11.7756494

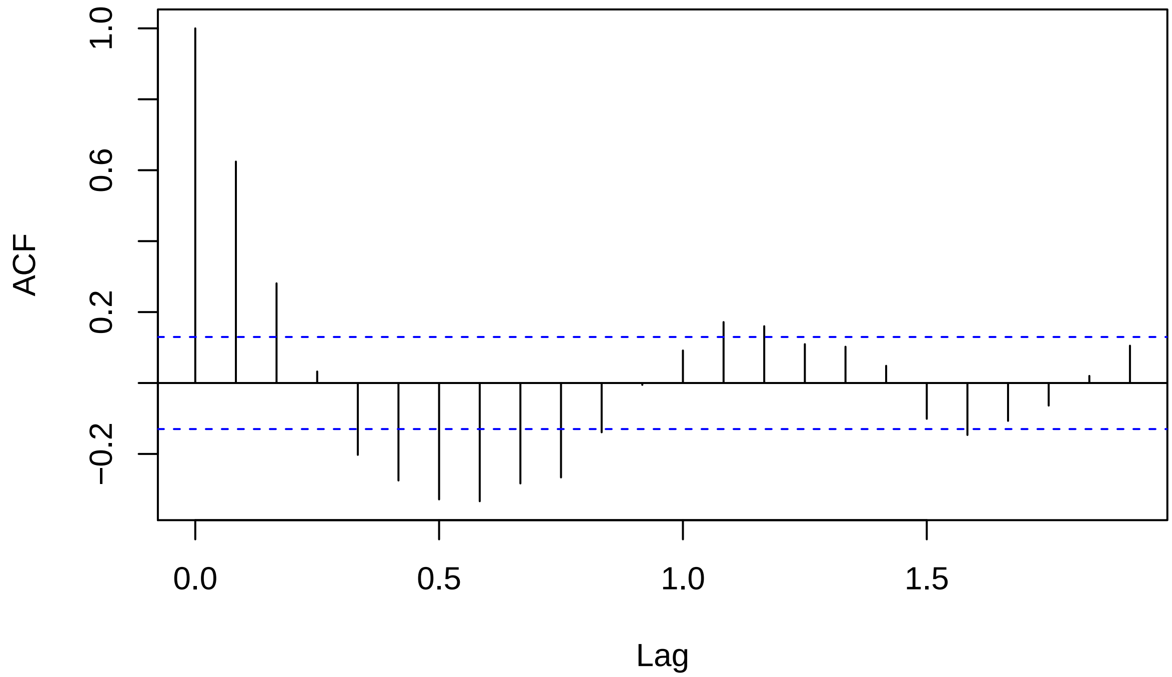
Reference:https://otexts.com/fpp2/stl.html

Based on MAE, X11 method gives a seasonal adjustment that is closest to StatCan's.

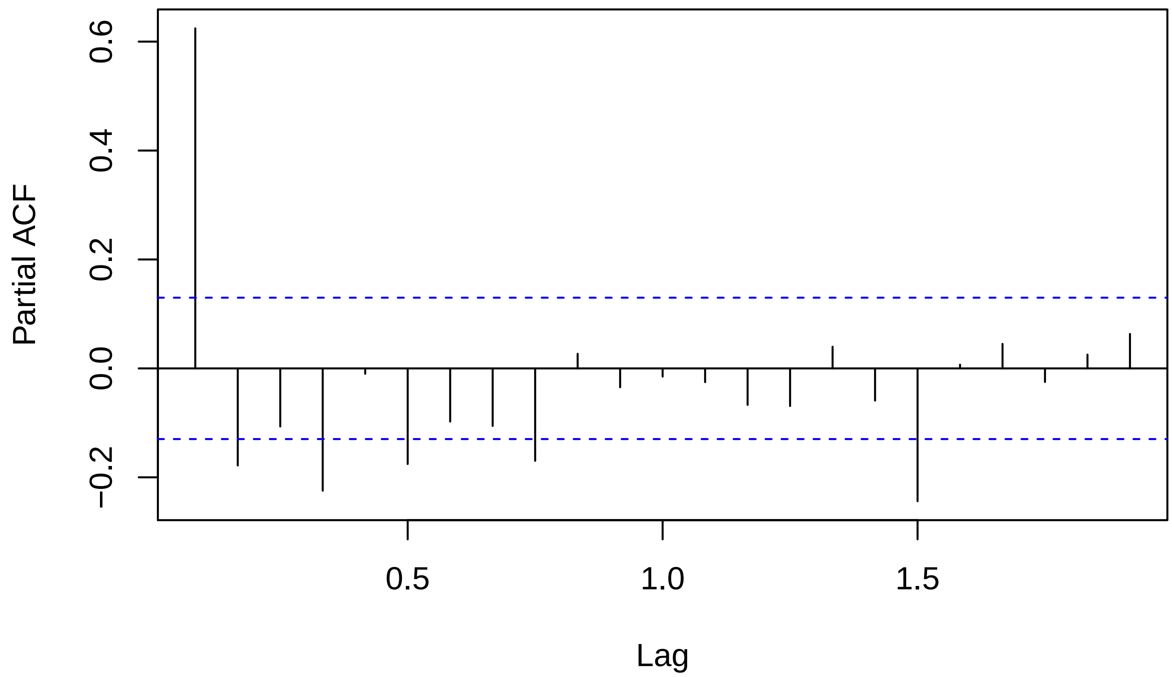
## Q5

### ua

```
R <- remainder(decompose(ua))
acf(na.omit(R),main='')
```




```
pacf(na.omit(R),main='')
```




```r
auto.arima(R)
```

```
## Series: R
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2
##       0.7399  -0.1851
## s.e.  0.0653   0.0654
##
## sigma^2 estimated as 372.8:  log likelihood=-997.79
## AIC=2001.58   AICc=2001.69   BIC=2011.87
```

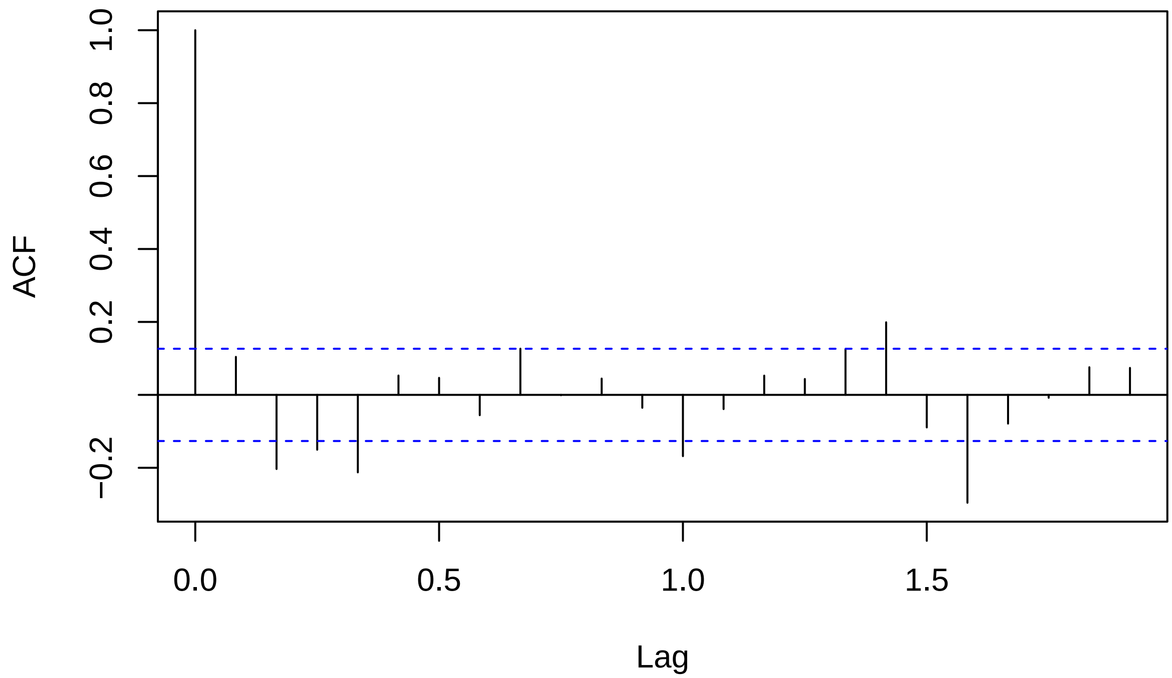
From ACF plot, I can identify some periodic pattern. So there is seasonality remaining, though the scale is small.Refer to the auto.arima table, there is no seasonality.

The remainder is stationary. For futher pre-processing, we prefer to do classical decomposition first, either multiplicative or additive. Decompose the time series into trend, seasonality and remainder, then build ARMA model on the staionnary remainder.

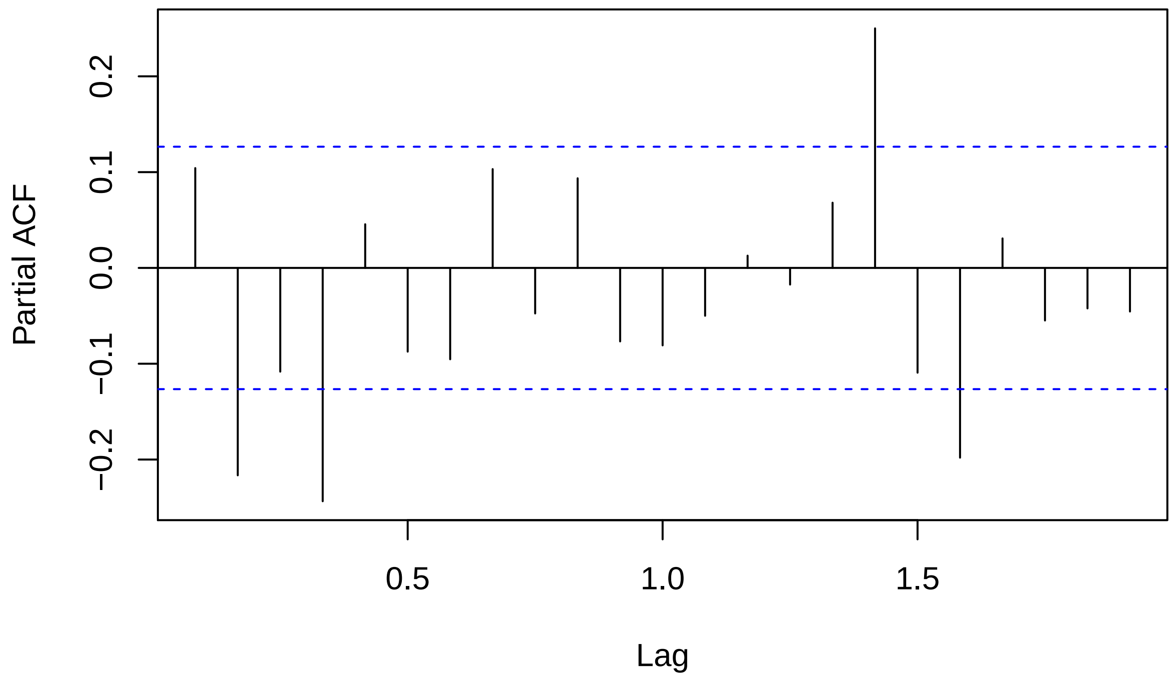
PACF tail off.Refer to auto.arima table, we chose ARIMA(2,0,0) model.

## X11fit

```r
R11 <- remainder(X11fit)
acf(na.omit(R11),main = '')
```




```r
pacf(na.omit(R11), main = '')
```




```r
auto.arima(R11)
```

```
## Series: R11
## ARIMA(1,0,2)(0,0,2)[12] with non-zero mean
##
## Coefficients:
##          ar1      ma1      ma2     sma1     sma2    mean
##       0.5293  -0.4679  -0.2821  -0.1623  -0.1113  0.9999
## s.e.  0.1409   0.1430   0.0705   0.0663   0.0606  0.0001
##
## sigma^2 estimated as 9.597e-06:  log likelihood=1048.49
## AIC=-2082.97   AICc=-2082.49   BIC=-2058.61
```

From ACF plot, tthere is no periodic pattern. So there is no seasonlity remaining.However, refer to auto.arima table, there is seasonlity remaining.

Though seasonality exist, the scale is small, so the remainder is staitonary.For futher pre-processing, we prefer to do classical decomposition first, either multiplicative or additive. Decompose the time series into trend, seasonality and remainder, then build ARMA model on the staionnary remainder.

PACF plot tail off. Refer to auto.arima table, we chose ARIMA(1,0,2), where p=1 for AR, q=2 for MA.