1.
(a) $Z_t = aX_t + bY_t$

$E(Z_t) = E(aX_t + bY_t) = aE(X_t) + bE(Y_t)$

$\because E(X_t) = E(Y_t) = 0.$

$E(Z_t) = 0.$

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

$= a^2 Cov(X_t, X_{t+h}) + ab\,Cov(X_t, Y_{t+h}) + ba\,Cov(Y_t, X_{t+h}) + b^2 Cov(Y_t, Y_{t+h})$

$= a^2 \gamma_X(h) + b^2 \gamma_Y(h)$ $\because X_t\}$ and $\{Y_t\}$ are independent

Thus, it only depends on h. It is stationary.

(b) $E(V_t) = E\left(\sum_{j=0}^{p} a_j X_{t-j}\right)$

$= \sum_{j=0}^{p} a_j E[X_{t-j}]$

$= 0.$

and autocovariance function.

$\gamma(h) = Cov(V_t, U_{t+h}) = Cov\left(\sum_{j=0}^{p} a_j X_{t-j}, \sum_{k=0}^{p} a_k X_{t+h-k}\right)$

$= \sum_{j=0}^{p} a_j \sum_{k=0}^{p} a_k \underbrace{Cov(X_{t-j}, X_{t+h-k})}_{0 \text{ unless, } t-j = t+h-k}$

$k = h+j.$

$= \frac{1-a_p}{1-a} \cdot \frac{1-a_p}{1-a} \cdot Var(X_t)$

$= \left(\frac{1-a_p}{1-a}\right)^2 \gamma_X(h).$

It is easy to see that autocovariance function only depends on h. Thus, it is stationary.

2. Given $\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (X_{t+h} \cdot X_t)$

$$E(\hat{\gamma}(h)) = E\left(\frac{1}{n} \sum_{t=1}^{n-h} (X_{t+h} \cdot X_t)\right)$$

$$= \frac{1}{n} \sum_{t=1}^{n-h} E(X_{t+h} \cdot X_t)$$

$$= \frac{1}{n} \sum_{t=1}^{n-h} \gamma(t, t+h) = \frac{1}{n} \sum_{t=1}^{n-h} \min\{t, t+h\} \cdot \sigma_w^2 .$$

since $\forall h = 0, 1, \ldots, n-1$:

$$= \frac{1}{n} \sum_{t=1}^{n-h} t$$

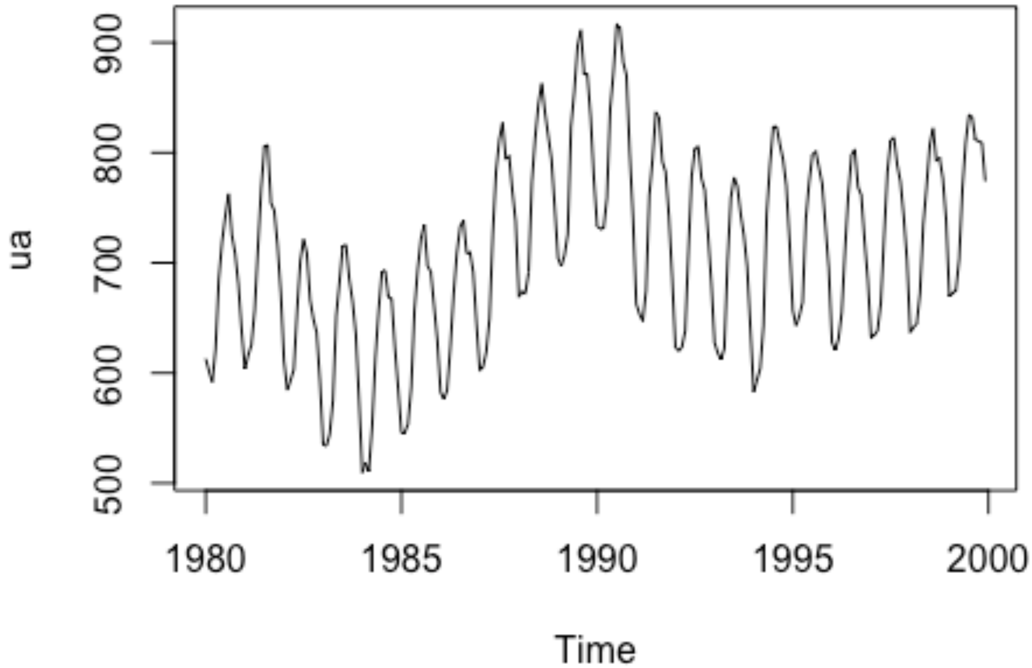
$$= \frac{1}{n} \left[\frac{(n-h)(n-h+1)}{2} \right] = \frac{(n-h)(n-h+1)}{2n}$$

Assignment1

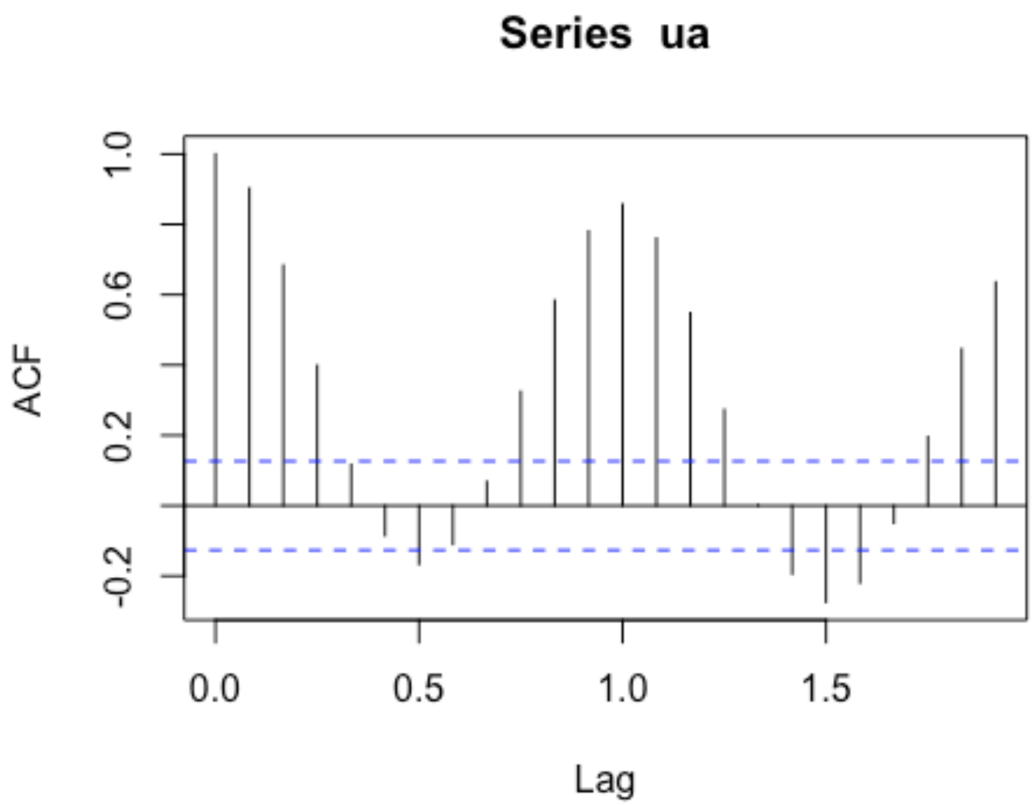
1

Plot the unadjusted series, its ACF & PACF, and comment on the following characteristics: trend, seasonality, stationarity

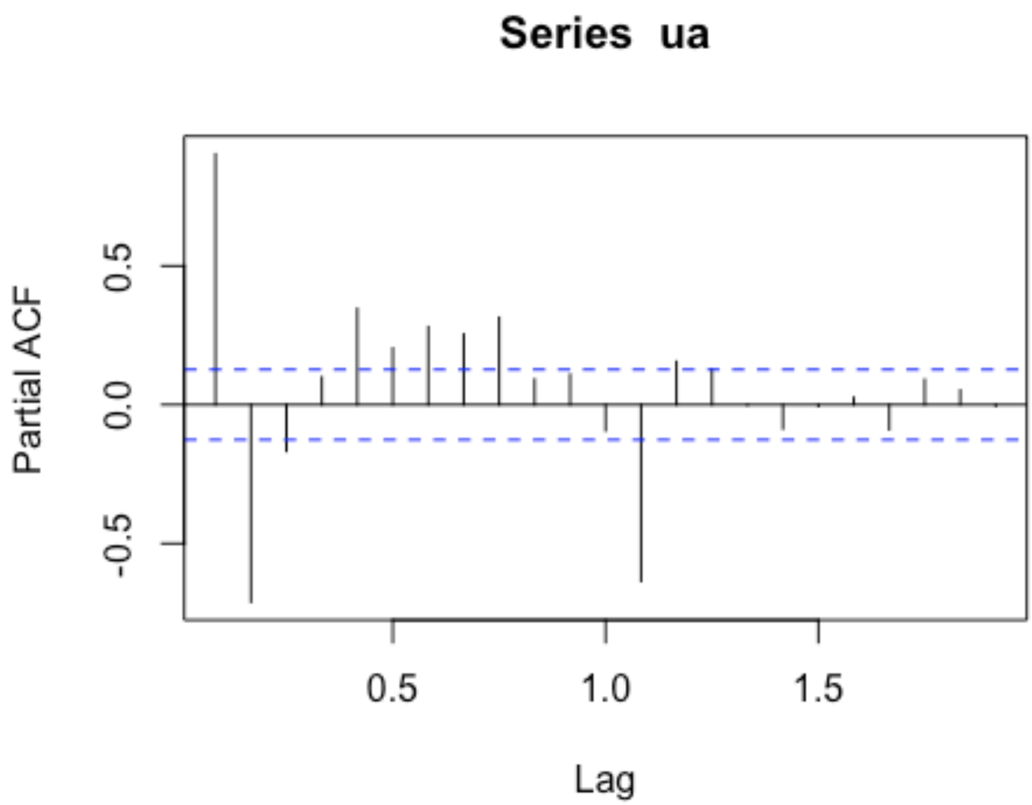
```
library(cansim)
library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────
tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.0     ✓ purrr   0.3.4
## ✓ tibble  3.0.1     ✓ dplyr   0.8.5
## ✓ tidyr   1.0.3     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0

## ── Conflicts ──────────────────────────────────────────────────
tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

ua = get_cansim_vector( "v2057817", start_time = "1980-01-01", end_time =
"1999-12-01") %>% pull(VALUE) %>% ts( start = c(1980,1), frequency = 12)

## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

plot(ua)
```




```
acf(ua)
```



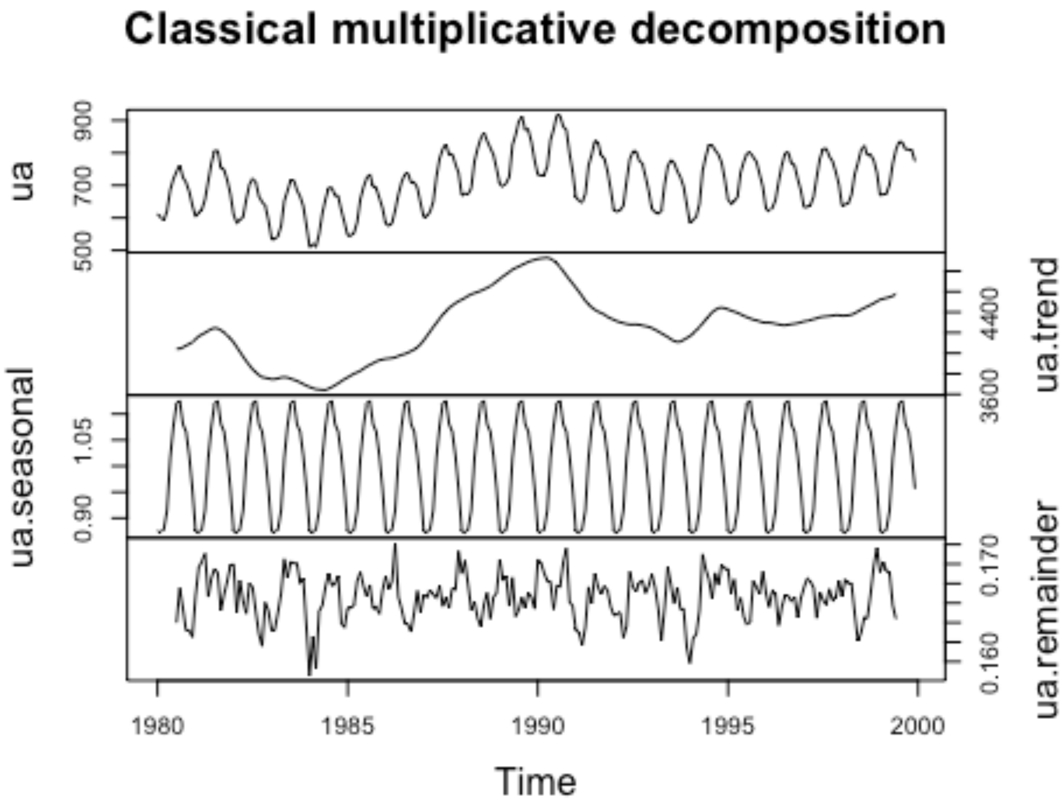

```r
pacf(ua)
```



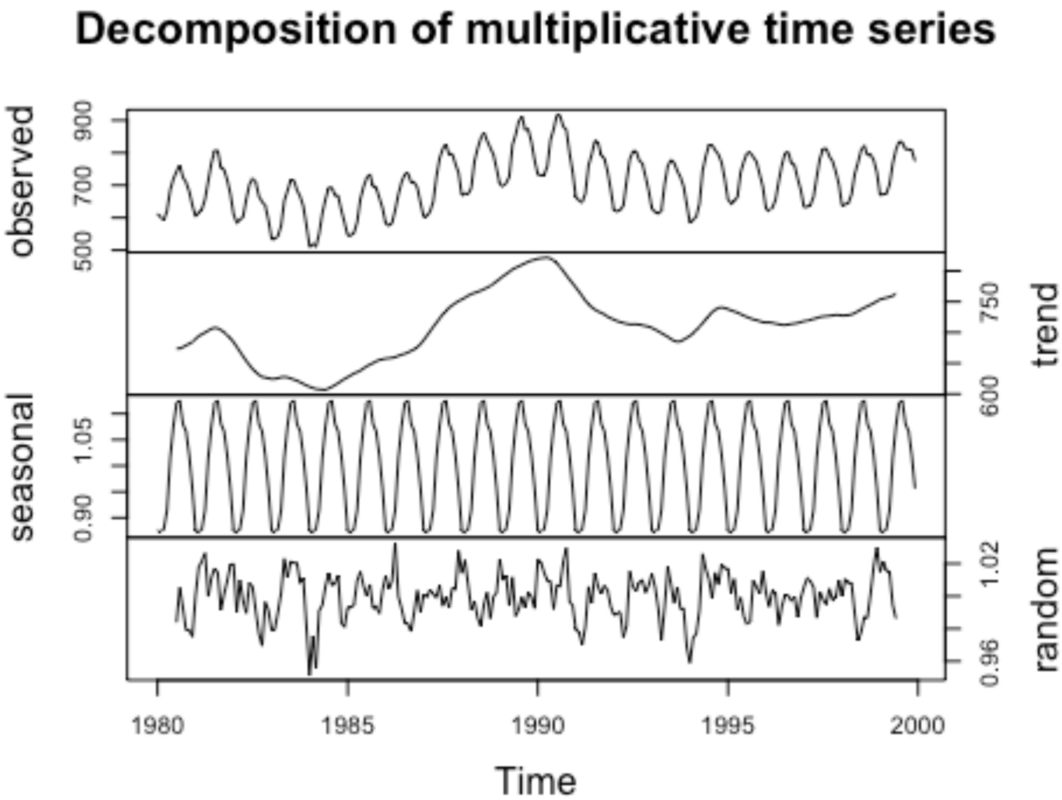
We can see the linear trend on the plot of unadjusted series, and ACF indicates that it has a seasonal pattern with period 1, and we can see both ACF and PACF does not tail off so the series would be non-stationary.

2. [5 marks] Perform a classical multiplicative decomposition of the unadjusted series (Xua) into trend (T), seasonal (S), and remainder (R) components (i.e. Xua = T × S × R):

```
ua.trend=stats::filter(ua,c(.5,rep(1,11),.5)/2,method="convo",sides=2)
trend = ua/ua.trend
ll <- length(trend)
ff <- frequency(trend)
periods <- ll %/%ff
index <- seq(1,ll,by=ff)-1
mm <- numeric(ff)
for ( i in 1:ff) {
  mm[i] <- mean(trend[index+i],na.rm=TRUE)
}
seasonal <- mm/mean((mm))
ua.seasonal <- ts(rep(seasonal,periods+1)[seq(ll)],start=1980,frequency=ff)
ua.remainder <- ua/(ua.trend*ua.seasonal)
plot(cbind(ua,ua.trend,ua.seasonal,ua.remainder),main="Classical
multiplicative decomposition",yax.flip = TRUE)
```




```r
a<-decompose(ua,type="multiplicative")
plot(a,yax.flip=TRUE)
```

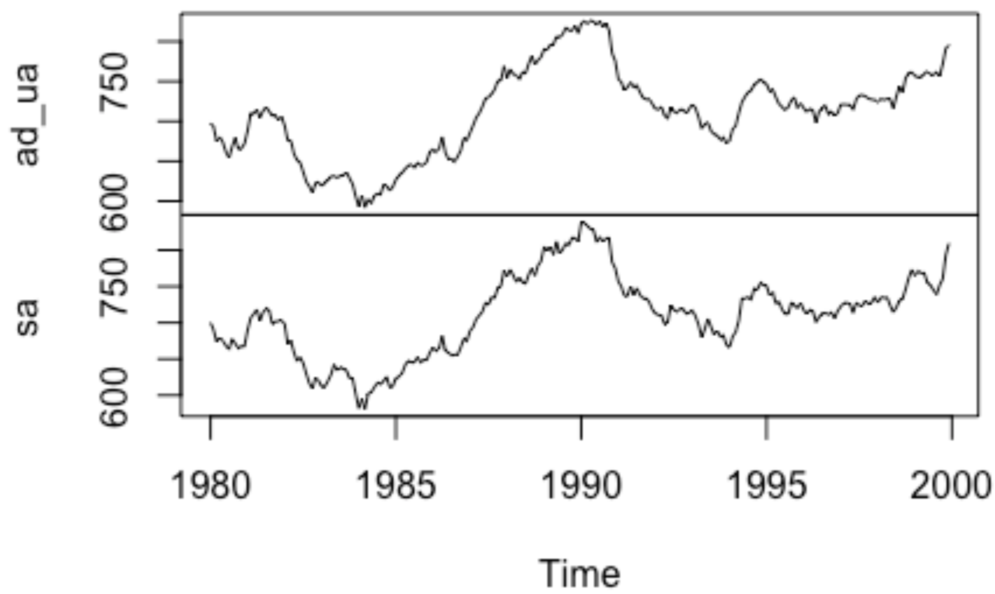


3.

Statistics Canada (StatCan) does their own seasonal adjustment using a more sophisticated method (namely, X-12-ARIMA). Download the corresponding seasonally adjusted series for your industry and time period, and plot them on the same plot with your own seasonally adjusted data (Xsa = Xua/S = T × R) from the previous part. The two versions should be close, but not identical. Report the mean absolute error (MAE) between the two versions (StaCan's and yours) of seasonally adjusted data.

```
ad_ua <- get_cansim_vector( "v2057608", start_time = "1980-01-01", end_time =
"1999-12-01") %>% pull(VALUE) %>% ts( start = c(1980,1), frequency = 12)
sa=ua/seasonal
plot(cbind(ad_ua,sa),main="comparison between seasonally adjusted sseries and
own seasonally adjusted data")
```

between seasonally adjusted sseries and own seaso

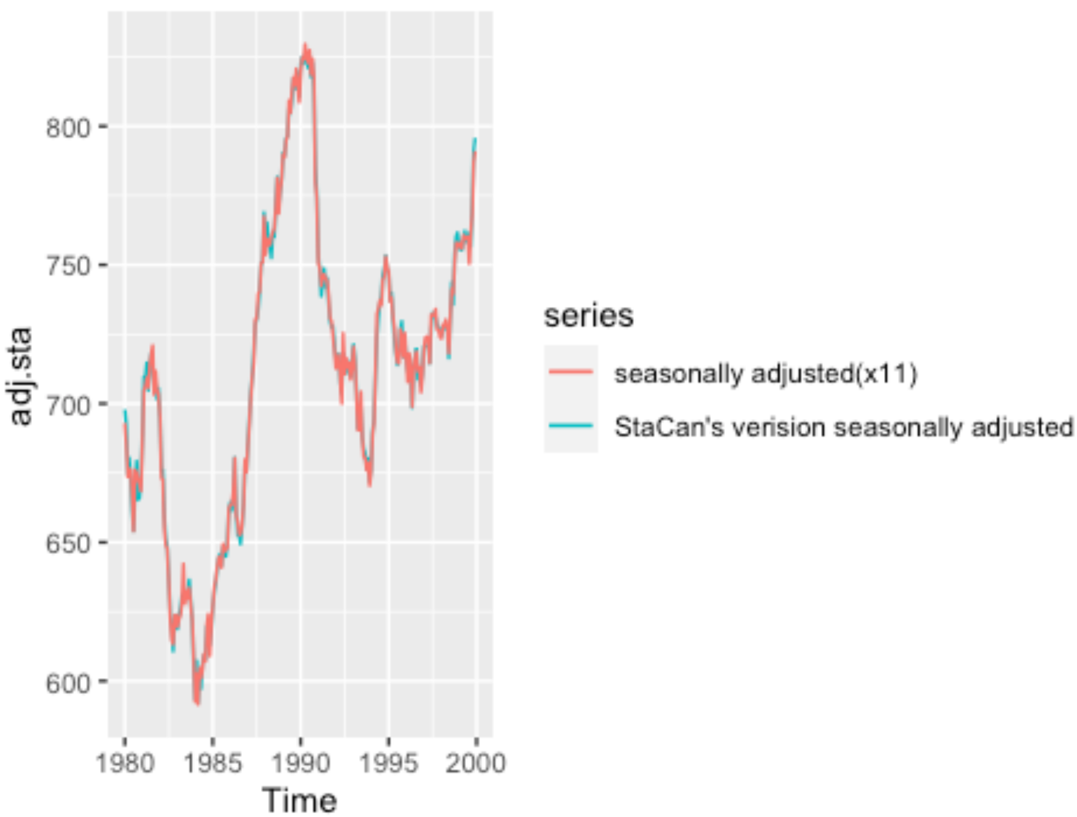



```
mae <- mean(abs(sa-ad_ua))
mae

## [1] 4.750756
```

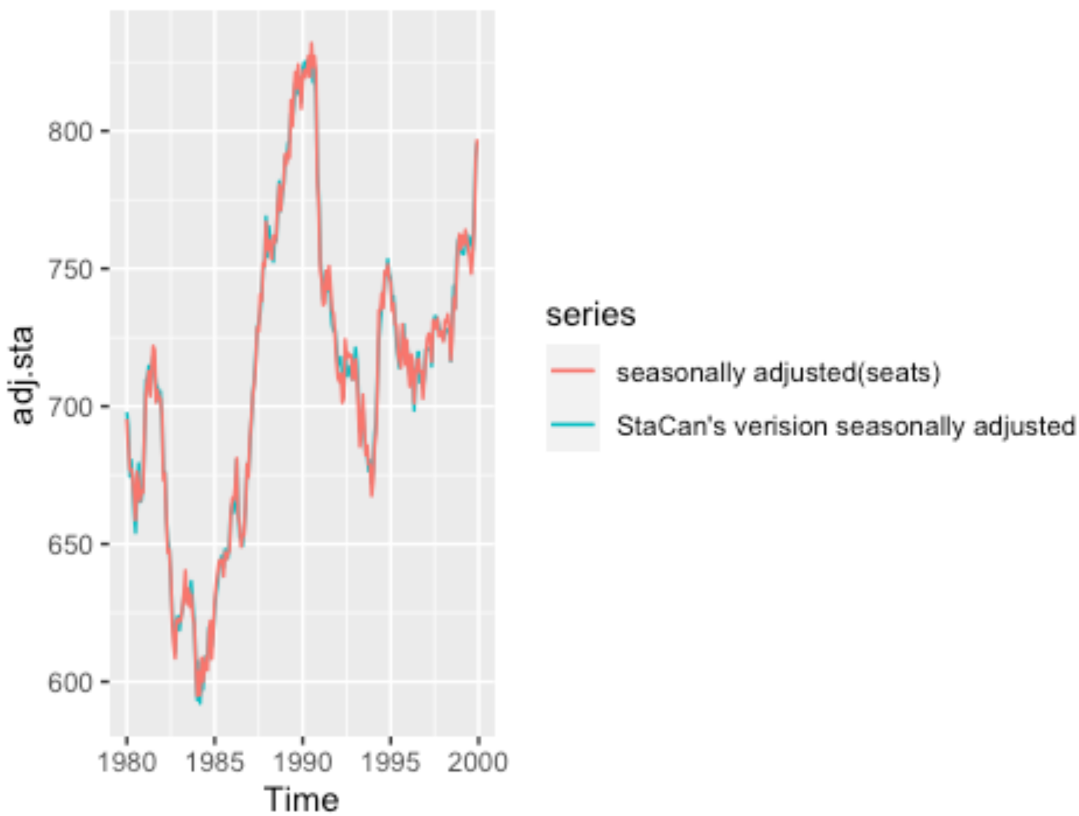
4.The library seasonal contains R functions for performing seasonal adjustments /decompositions using various methods. Use the following three methods described in FPP for performing seasonal adjustments (you don't need to know their details): a.

```
library(seasonal)

##
## Attaching package: 'seasonal'

## The following object is masked from 'package:tibble':
##
##     view

library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(ggplot2)
ua.x11<- seas(ua,x11="")
adj.x11<-seasadj(ua.x11)
sea.x11<-seasonal(ua.x11)
tre.x11<-trendcycle(ua.x11)
rem.x11<-remainder(ua.x11)
adj.sta<-get_cansim_vector( "v2057608", start_time = "1980-01-01", end_time =
"1999-12-01") %>% pull(VALUE) %>% ts( start = c(1980,1), frequency = 12)
autoplot(adj.sta,series = "StaCan's verision seasonally adjusted") +
  autolayer(adj.x11,series="seasonally adjusted(x11) ")
```



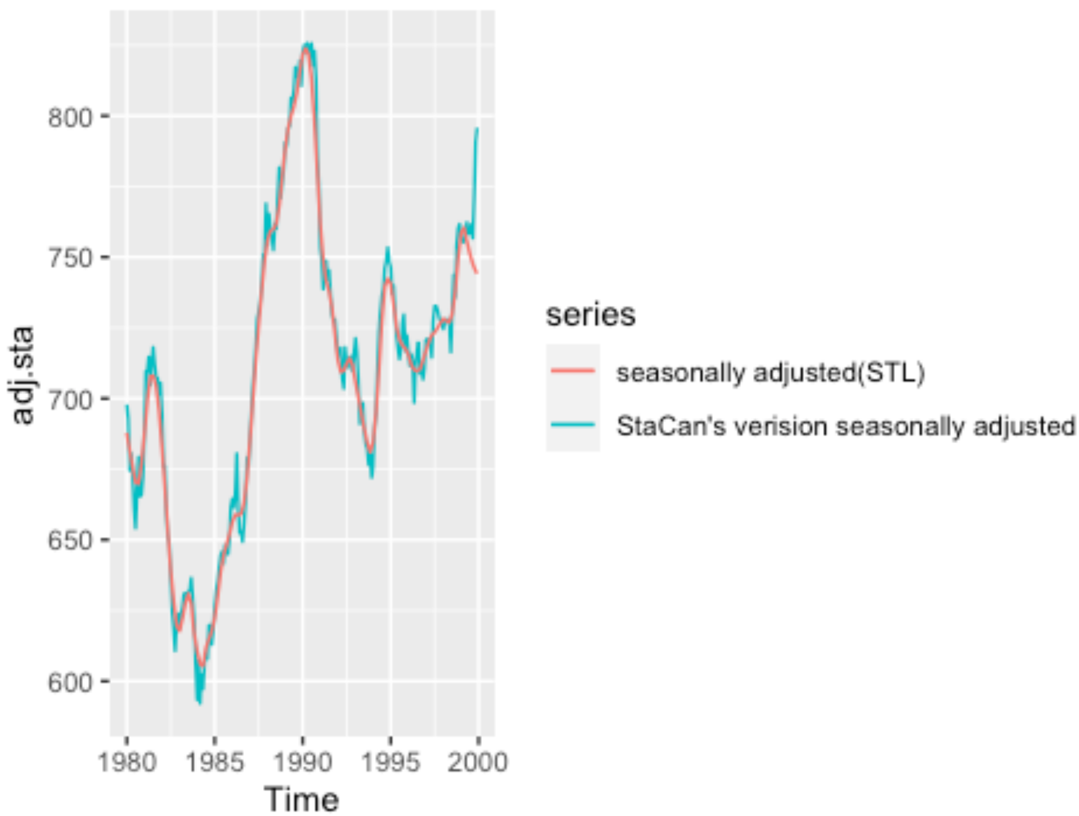

```r
MAE.x11 <-mean(abs(adj.x11-adj.sta))
MAE.x11

## [1] 2.237281

ua.seats <- seas(ua)
adj.seats<-seasadj(ua.seats)
sea.seats<-seasonal(ua.seats)
tre.seats<-trendcycle(ua.seats)
rem.seats<-remainder(ua.seats)
autoplot(adj.sta,series = "StaCan's verision seasonally adjusted") +
  autolayer(adj.seats,series="seasonally adjusted(seats)")
```




```
MAE.seats <- mean(abs(adj.seats-adj.sta))
MAE.seats
```

```
## [1] 3.159652
```

```
logua<-log(ua)
logua.stl<-stl(logua,t.window=13,s.window="periodic",robust=TRUE)
logua.stl$time.series<-exp(logua.stl$time.series)
ua.stl<-logua.stl
adj.stl<-seasadj(ua.stl)
sea.stl<-seasonal(ua.stl)
tre.stl<-trendcycle(ua.stl)
rem.stl<-remainder(ua.stl)
autoplot(adj.sta,series = "StaCan's verision seasonally adjusted") +
 autolayer(adj.stl, series = "seasonally adjusted(STL)")
```



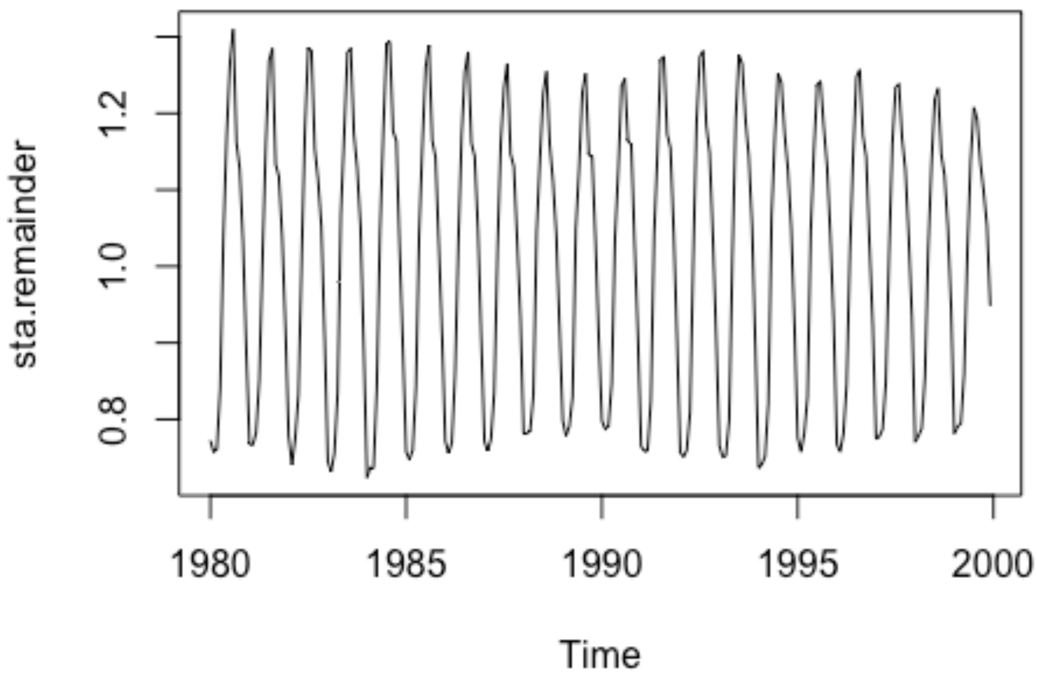

```
MAE.stl <- mean(abs(adj.stl-adj.sta))
MAE.stl

## [1] 5.959617
```

Based on MAE value, we can find that x11 method gives a seasonal adjustment that is the closest to StaCan's.

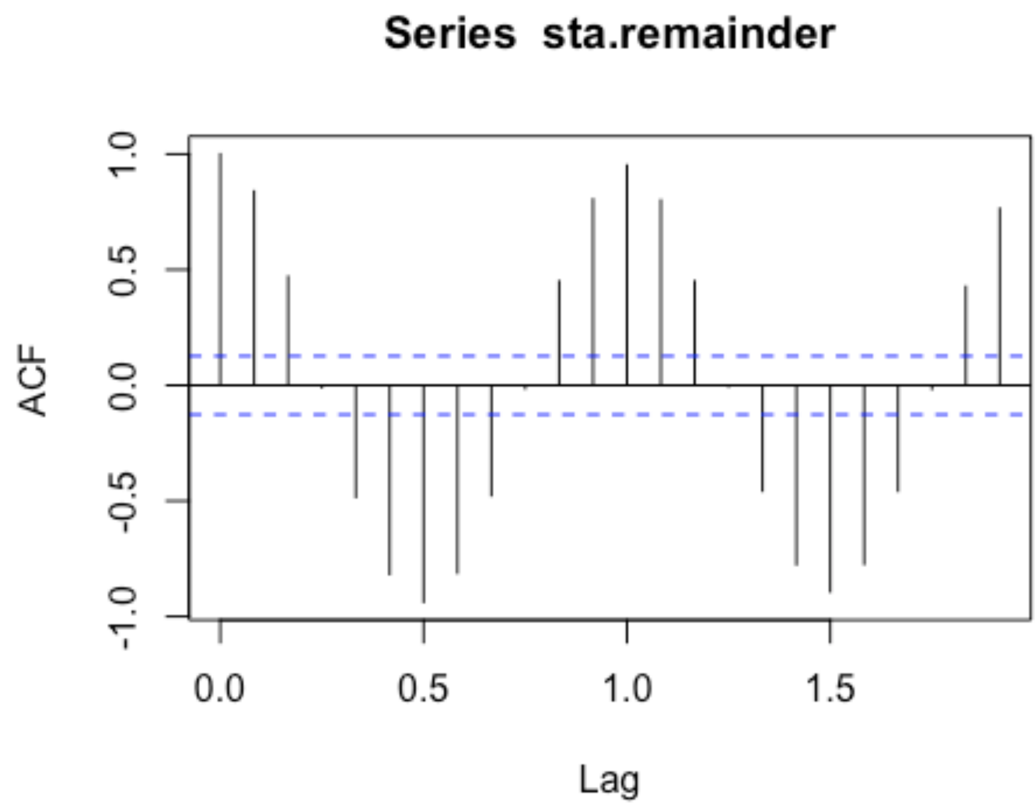
4.

Using StatCan's data (unadjusted, and/or seasonally adjusted, and/or trend-cycle), calculate the remainder series (R). Plot R and its sample ACF and PACF, and answer the following questions:

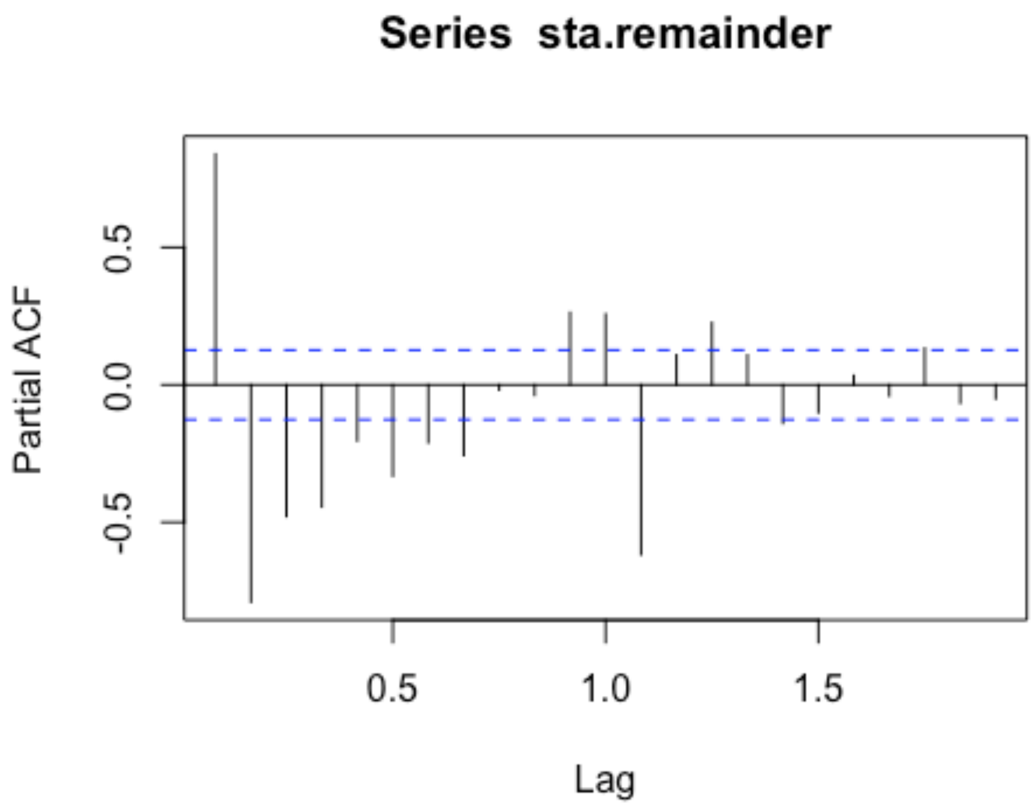
```
sta.trend<-get_cansim_vector( "v123355111", start_time = "1980-01-01",
end_time = "1999-12-01") %>% pull(VALUE) %>% ts( start = c(1980,1), frequency
= 12)
sta.seasonal<-adj.sta/ua
sta.remainder <-ua/(sta.seasonal*sta.trend)
plot(sta.remainder)
```




```
acf(sta.remainder)
```




```
pacf(sta.remainder)
```



a. Based on these plots, can you identify any remaining seasonality in your series?

No, it is detrended well, so it is difficult to capture the seasonality of it.

b. Comment on the stationarity of the series and propose any further pre-processing.

It is hard to say that it acf shows that the stationarity. Since ACF does not drop to zero for larger lags, and there is no change in the pattern. We can differencing it, or work with subset of series.

c. Comment on the (partial) autocorrelations of the series, and propose an appropriate ARMA(p, q) model (i.e. appropriate orders p & q).

PACF plot tails off which gives an intuition that p=0,q>2 since ACF is not cut off yet in the plot.It will be a MA(q) model.