Time series

Park

12/16/2019

In this project, I'll predict Canada data which is already placed inside tseries package. e stands for employment, prod stands for productivity, rw stands for real wage, and U stands for unemployment. The term "White noise" is a data that is randomly distributed, non-autocorrelated, not irritating.

#Exploring data ##

```
library(vars)
```

```
## Loading required package: MASS
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: urca
```

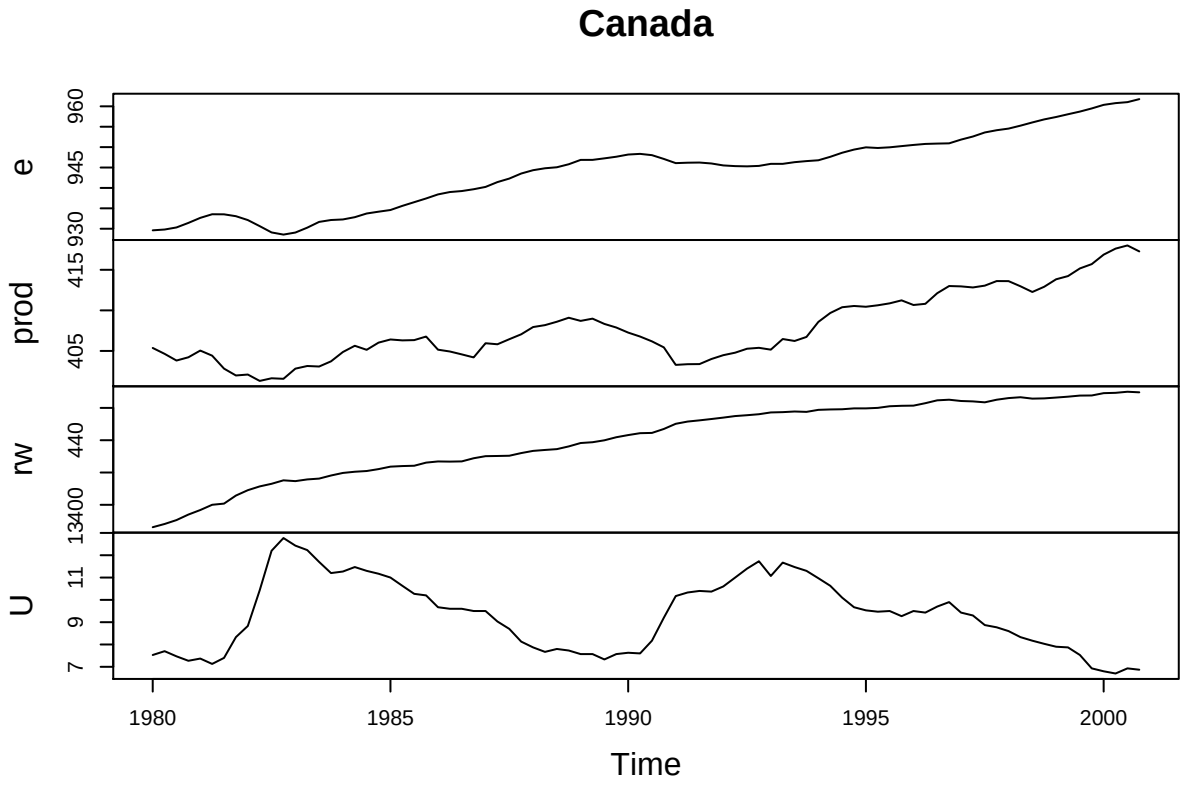
```
## Loading required package: lmtest
```

```
library(tseries)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

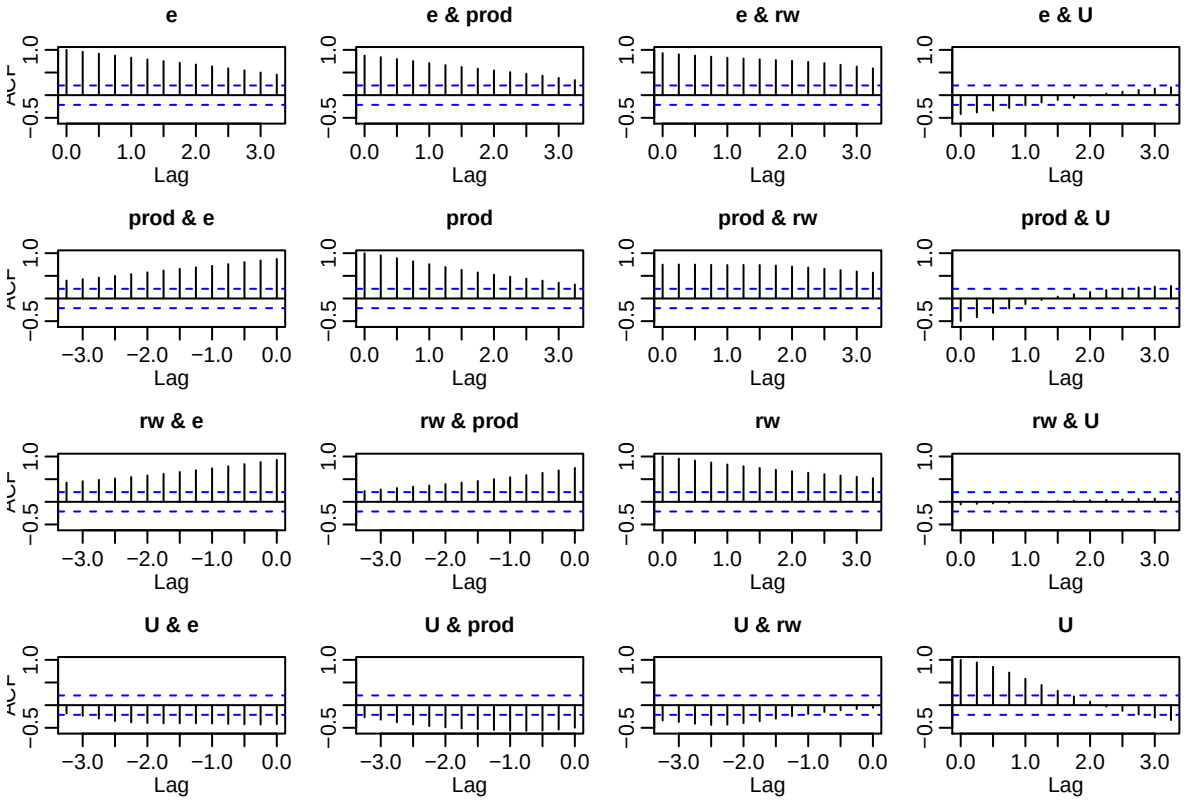
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
data("Canada")
plot.ts(Canada)
```



- In e, prod, and rw variables, they have up right trend. - In U variable, their is a up-down trend that has frequency of 10 years. This is hard to define as a seasonality.

```
acf(Canada)
```



- In the raw data, variables are autocorrelated with each other by looking at CCF. - Variables are also self correlated as well.

```
frequency(Canada); start(Canada); end(Canada)
```

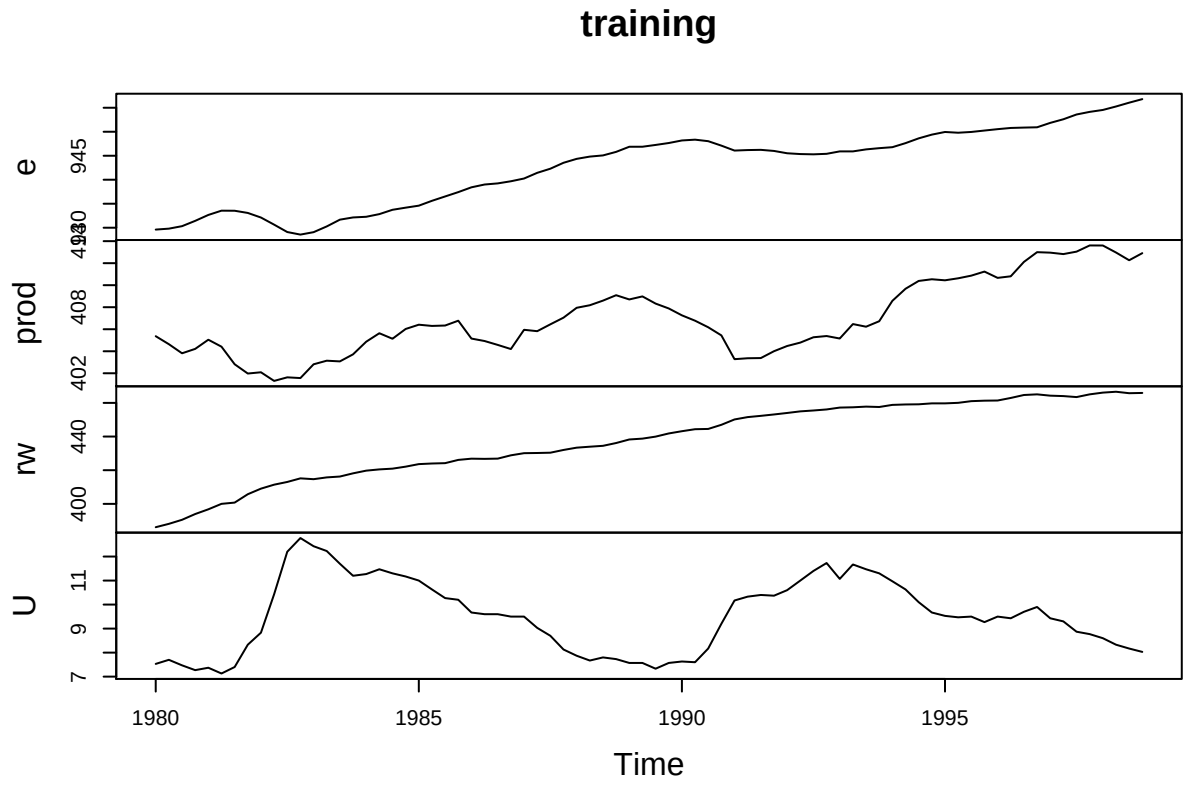
```
## [1] 4
```

```
## [1] 1980    1
```

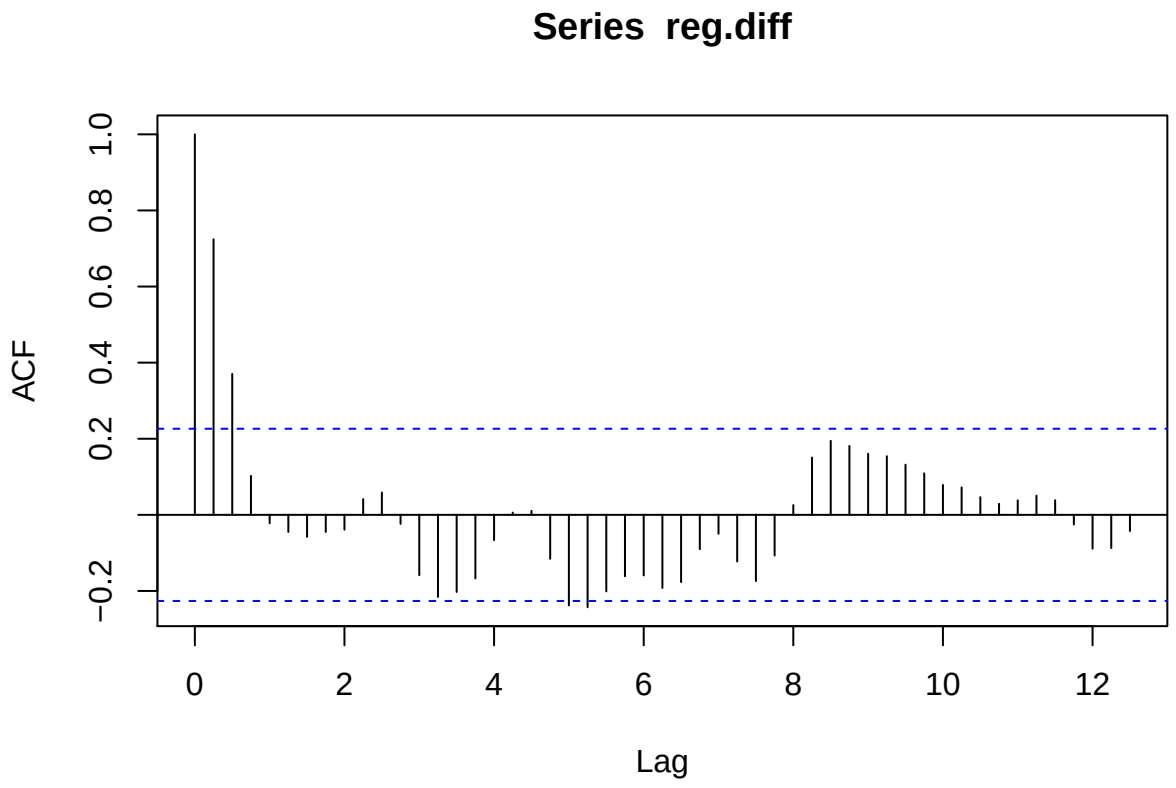
```
## [1] 2000    4
```

```
training = ts(Canada, start = c(1980,1), end = c(1998, 4), freq = 4)    ##Divide data into train, and te
test = ts(Canada, start = c(1999,1), end = c(2000,4), freq = 4)         ##I'm going to set last 8 quarte
canada.e = training[,1]
canada.prod = training[,2]
canada.rw = training[,3]
canada.U = training[,4]
canada.U.test = test[,4]
```

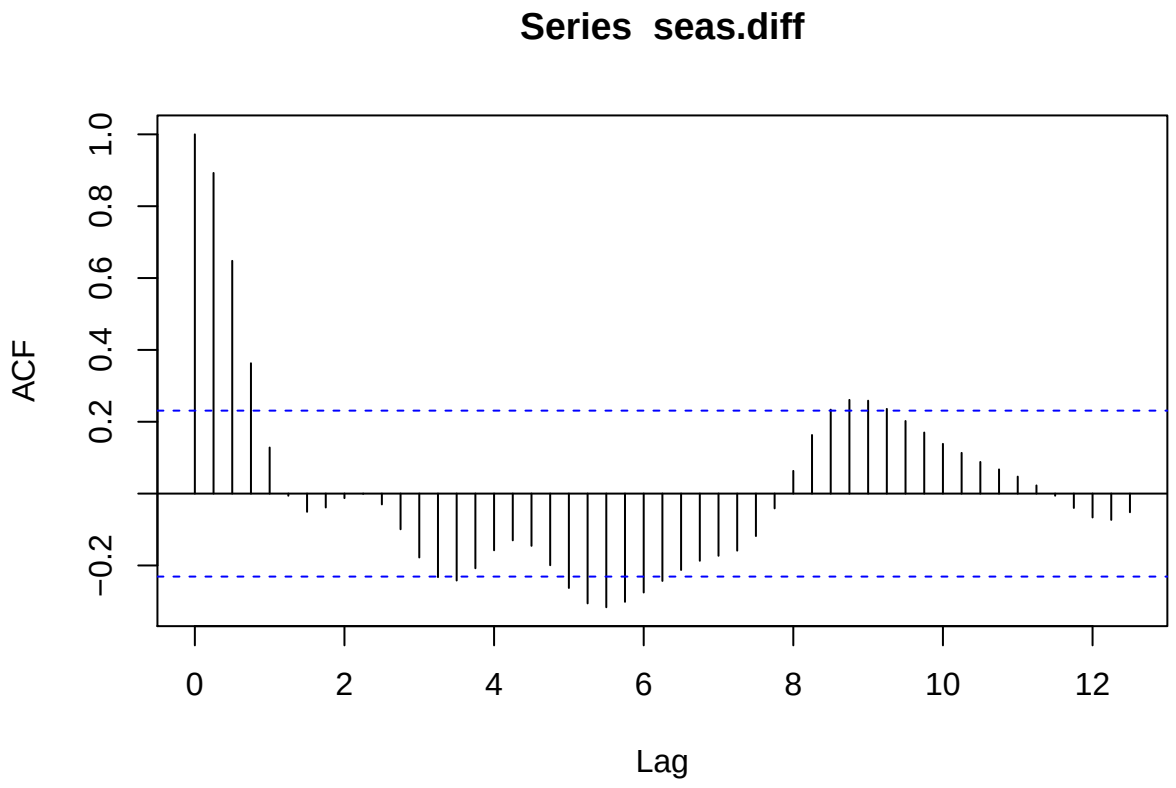
```
#Plot of mts object
plot.ts(training)
```



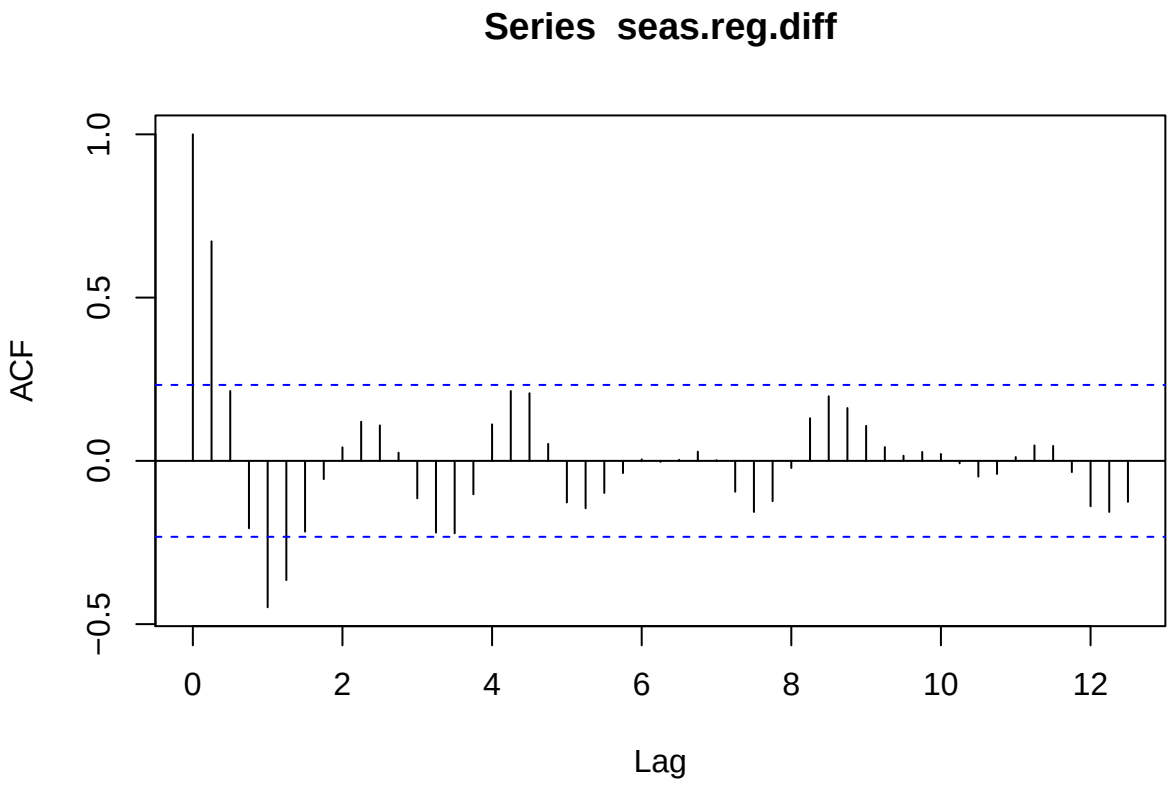

```
#Change Model

#Regular differencing for e

reg.diff=diff(canada.e, lag=1,diff=1)

acf(reg.diff,lag=50)
```



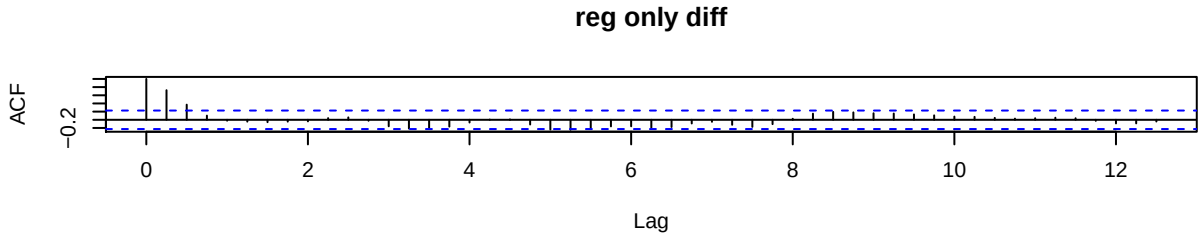

```
#Seasonal differencing for e

seas.diff=diff(canada.e, lag=4,diff=1)

acf(seas.diff,lag=50)
```

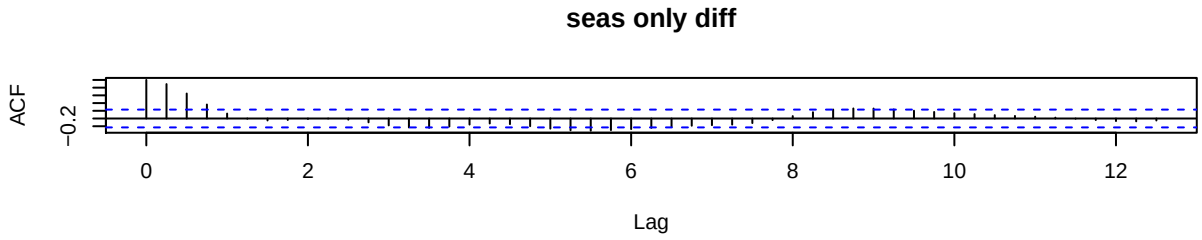


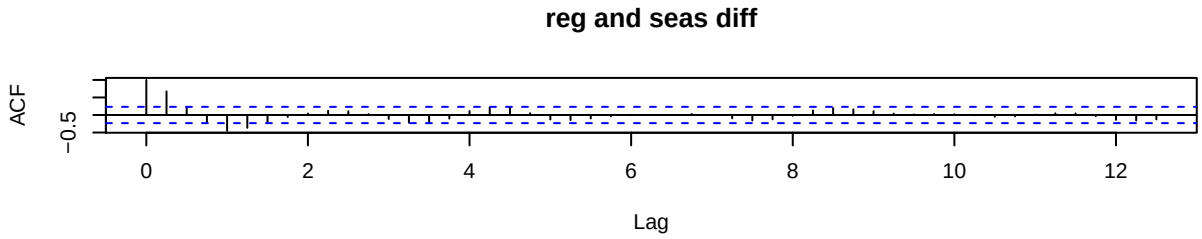

```
#Regular seasonal differencing for e

seas.reg.diff=diff(reg.diff, lag=4,diff=1)

acf(seas.reg.diff,lag=50)
```




```
##See acf plot of each differencing method

par(mfrow=c(3,1))

acf(reg.diff,lag=50, main="reg only diff")

acf(seas.diff,lag=50,main="seas only diff")

acf(seas.reg.diff,lag=50,main="reg and seas diff")
```








```
# when convinced, type

dev.off()
```

```
## null device
##           1
```

```
# conclusion : regular differencing is enough for e
e.diff<-reg.diff

# Do the same for prod
reg.diff=diff(canada.prod, lag=1,diff=1)

acf(reg.diff,lag=50)



# Second check only seasonal differencing

seas.diff=diff(canada.prod, lag=4,diff=1)

acf(seas.diff,lag=50)



# Now do seasonal differencing of regular differencing
```

```r
seas.reg.diff=diff(reg.diff, lag=4,diff=1)

acf(seas.reg.diff,lag=50)



## view the three acf together for easy comparison

par(mfrow=c(3,1))

acf(reg.diff,lag=50, main="reg only diff")

acf(seas.diff,lag=50,main="seas only diff")

acf(seas.reg.diff,lag=50,main="reg and seas diff")

# when convinced, type
dev.off()
```

```
## null device
##           1
```

```r
# conclusion : regular differencing for prod
prod.diff<-reg.diff

# Do the same for rw
reg.diff=diff(canada.rw, lag=1,diff=1)

acf(reg.diff,lag=50)



# Second check only seasonal differencing

seas.diff=diff(canada.rw, lag=4,diff=1)

acf(seas.diff,lag=50)



# Now do seasonal differencing of regular differencing

seas.reg.diff=diff(reg.diff, lag=4,diff=1)

acf(seas.reg.diff,lag=50)



## view the three acf together for easy comparison

par(mfrow=c(3,1))

acf(reg.diff,lag=50, main="reg only diff")
```

```r
acf(seas.diff,lag=50,main="seas only diff")

acf(seas.reg.diff,lag=50,main="reg and seas diff")

# when convinced, type

dev.off()
```

```
## null device
##           1
```

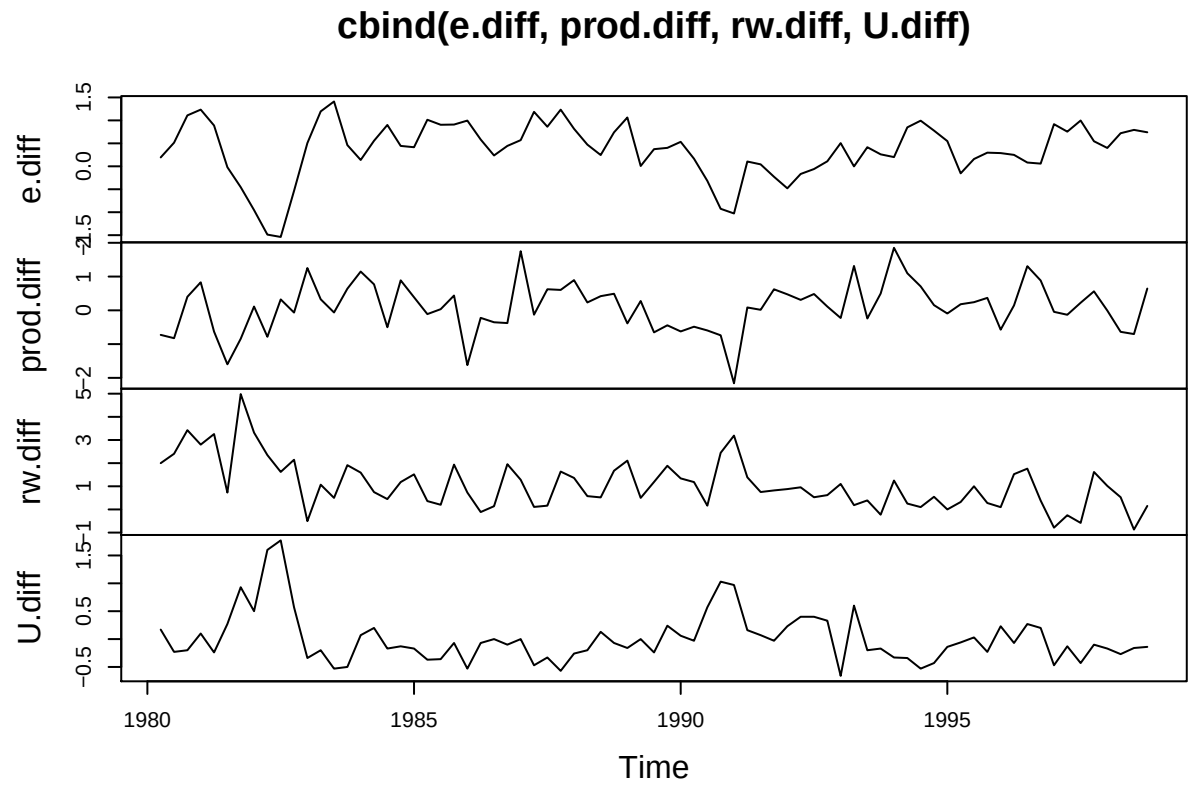
```r
# conclusion : regular differencing for prod
rw.diff<-reg.diff

# Do the same for U
reg.diff=diff(canada.U, lag=1,diff=1)

acf(reg.diff,lag=50)



# Second check only seasonal differencing

seas.diff=diff(canada.U, lag=4,diff=1)

acf(seas.diff,lag=50)



# Now do seasonal differencing of regular differencing

seas.reg.diff=diff(reg.diff, lag=4,diff=1)

acf(seas.reg.diff,lag=50)



## view the three acf together for easy comparison

par(mfrow=c(3,1))

acf(reg.diff,lag=50, main="reg only diff")

acf(seas.diff,lag=50,main="seas only diff")

acf(seas.reg.diff,lag=50,main="reg and seas diff")

# when convinced, type

dev.off()
```

```
## null device
##           1
```

```
# conclusion : regular differencing for prod
U.diff<-reg.diff

#Autocorrelation function for differenced U
acf(U.diff)

#Cross correlation funciton for differenced U
acf(cbind(e.diff,prod.diff,rw.diff,U.diff))
```

- By applying regular differencing, I'm able to define significant cross-correlation between variables.
- Most of variables die down with sin pattern in the non-diagonal slots in the Cross-correlation Function.
- This means that Variables are self-correlated with lag of 2.
- In e.df & prd, significant spike appears is r1 and last for another 1 lag until it decays which means that e.diff is correlated with 2 lag of prod.diff.
- In e.df & rw.d, significant spike appears and last for 5 lags until it decays.
- In U.df & prd, significant spike appears in r1 and last for 1 more lag until it decays.
- In U.df & rw.d, significant spike appears but it fluctuates and last for 2 more lags until it finally decays.

```
plot.ts(cbind(e.diff,prod.diff,rw.diff,U.diff))
```



- By using regular differencing, trends are almost emilinated. - But there are still some seasonality remain on the plot.

#Predict by using VAR Model ##

```
canada.var=VAR(cbind(canada.e,canada.prod,canada.rw,canada.U),p=3, type="trend")
coef(canada.var)
```

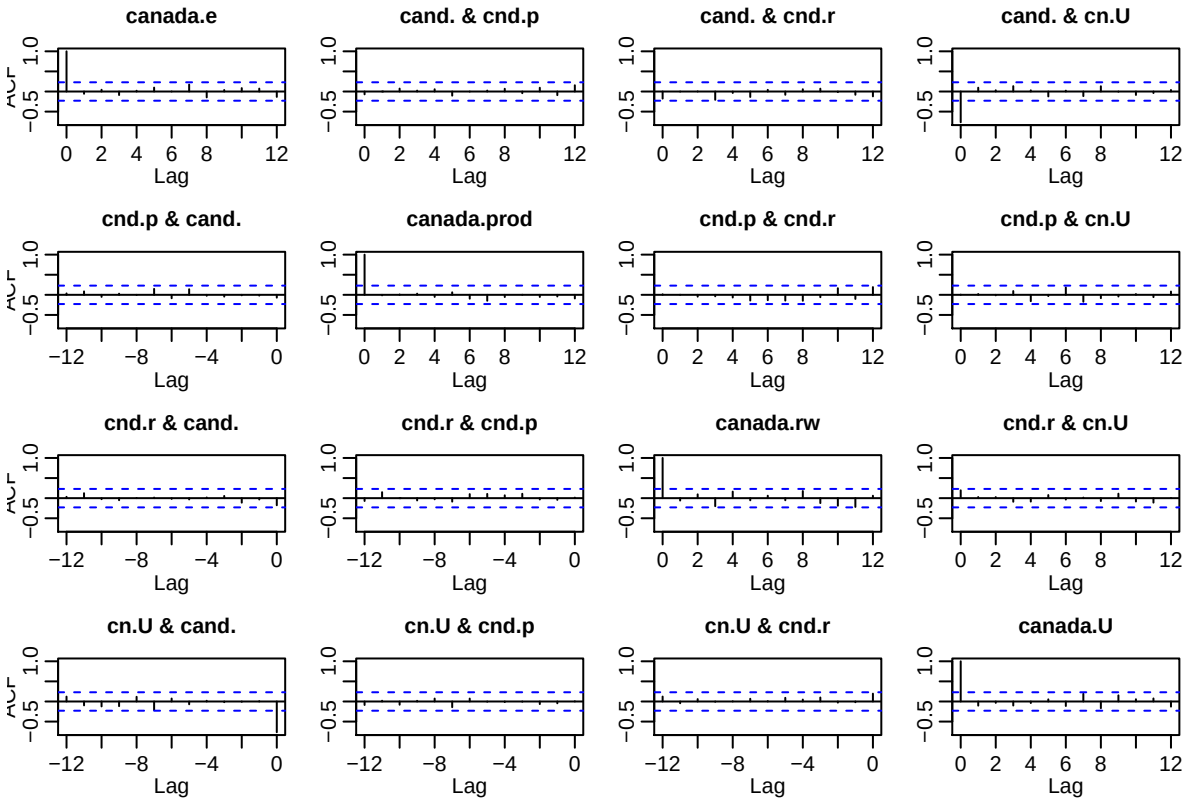
```
## $canada.e
```

```
##                     Estimate Std. Error      t value     Pr(>|t|)
## canada.e.l1     1.7666650404 0.16769309 10.535109130 2.862317e-15
## canada.prod.l1  0.2131802719 0.07172299  2.972272543 4.249645e-03
## canada.rw.l1   -0.0715758841 0.06049456 -1.183178825 2.414052e-01
## canada.U.l1    -0.0126101753 0.21127791 -0.059685252 9.526047e-01
## canada.e.l2    -1.2785557052 0.26288402 -4.863573292 8.688491e-06
## canada.prod.l2 -0.1612374172 0.10483325 -1.538036977 1.292970e-01
## canada.rw.l2   -0.0308422488 0.07677073 -0.401744914 6.892992e-01
## canada.U.l2    -0.1287292898 0.27481356 -0.468424087 6.411787e-01
## canada.e.l3     0.4970907199 0.17984743  2.763957838 7.573087e-03
## canada.prod.l3  0.0006758222 0.07465505  0.009052598 9.928072e-01
## canada.rw.l3    0.0848490378 0.05640664  1.504238470 1.377660e-01
## canada.U.l3     0.1591276132 0.21608182  0.736422968 4.643430e-01
## trend           0.0118315364 0.01811970  0.652965489 5.162728e-01
##
## $canada.prod
##                     Estimate Std. Error     t value     Pr(>|t|)
## canada.e.l1    -0.207369850 0.29355823 -0.70640107 4.826730e-01
## canada.prod.l1  1.064478386 0.12555600  8.47811639 7.524813e-12
## canada.rw.l1    0.002799404 0.10589987  0.02643444 9.789986e-01
## canada.U.l1    -0.773488980 0.36985644 -2.09132219 4.074128e-02
## canada.e.l2    -0.192063569 0.46019647 -0.41735125 6.779111e-01
## canada.prod.l2 -0.170884667 0.18351778 -0.93116137 3.555018e-01
## canada.rw.l2   -0.205960836 0.13439241 -1.53253325 1.306471e-01
## canada.U.l2     0.699690342 0.48107994  1.45441597 1.510426e-01
## canada.e.l3     0.471964106 0.31483524  1.49908286 1.390956e-01
## canada.prod.l3  0.008560784 0.13068878  0.06550512 9.479896e-01
## canada.rw.l3    0.128061174 0.09874368  1.29690498 1.996291e-01
## canada.U.l3     0.334987626 0.37826601  0.88558744 3.793754e-01
## trend           0.058086195 0.03171977  1.83123012 7.203237e-02
##
## $canada.rw
##                    Estimate Std. Error    t value     Pr(>|t|)
## canada.e.l1    -0.43093955 0.33939251 -1.2697380 2.090806e-01
## canada.prod.l1 -0.07596471 0.14515950 -0.5233189 6.026798e-01
## canada.rw.l1    0.83123969 0.12243439  6.7892666 5.717111e-09
## canada.U.l1     0.25982321 0.42760342  0.6076266 5.457281e-01
## canada.e.l2     0.80498969 0.53204855  1.5130004 1.355295e-01
## canada.prod.l2 -0.25070550 0.21217105 -1.1816197 2.420189e-01
## canada.rw.l2   -0.20710901 0.15537557 -1.3329574 1.875853e-01
## canada.U.l2    -0.33480741 0.55619263 -0.6019631 5.494668e-01
## canada.e.l3    -0.23444854 0.36399156 -0.6441043 5.219625e-01
## canada.prod.l3  0.15538307 0.15109367  1.0283890 3.078949e-01
## canada.rw.l3    0.22990388 0.11416088  2.0138587 4.851515e-02
## canada.U.l3     0.08704296 0.43732600  0.1990345 8.429087e-01
## trend           0.08661960 0.03667229  2.3619906 2.143938e-02
##
## $canada.U
##                    Estimate Std. Error    t value     Pr(>|t|)
## canada.e.l1    -0.624333539 0.13730205 -4.5471537 2.693873e-05
## canada.prod.l1 -0.137403866 0.05872462 -2.3397999 2.263941e-02
## canada.rw.l1    0.010625895 0.04953112  0.2145296 8.308615e-01
## canada.U.l1     0.737521307 0.17298799  4.2634250 7.233178e-05
## canada.e.l2     0.567163486 0.21524152  2.6350097 1.069056e-02
```

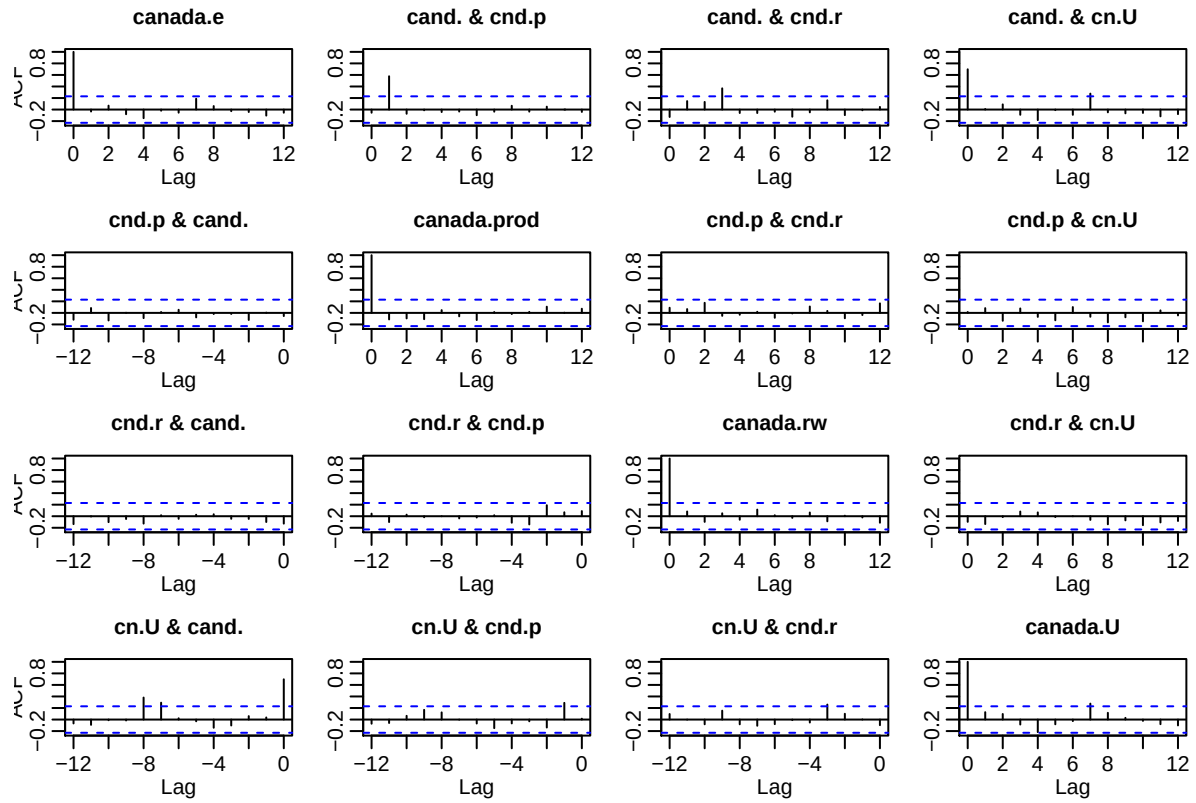
```
## canada.prod.l2  0.134893816 0.08583431  1.5715606 1.213115e-01
## canada.rw.l2    0.071098535 0.06285756  1.1311056 2.625115e-01
## canada.U.l2    -0.037593948 0.22500906 -0.1670775 8.678709e-01
## canada.e.l3     0.062302271 0.14725366  0.4230949 6.737387e-01
## canada.prod.l3 -0.007869266 0.06112531 -0.1287399 8.979940e-01
## canada.rw.l3   -0.081131939 0.04618406 -1.7567088 8.406961e-02
## canada.U.l3     0.243308859 0.17692128  1.3752379 1.741710e-01
## trend          -0.003921209 0.01483586 -0.2643061 7.924498e-01
```

- In regular differenced CCF, some of the variables are correlated with other variables that has lag of 3.
- It is a good idea to use p = 3 for VAR model which makes lag 3 value to be include.
- P-value are significnat when they are less than 0.05.

```
acf(resid(canada.var))
```

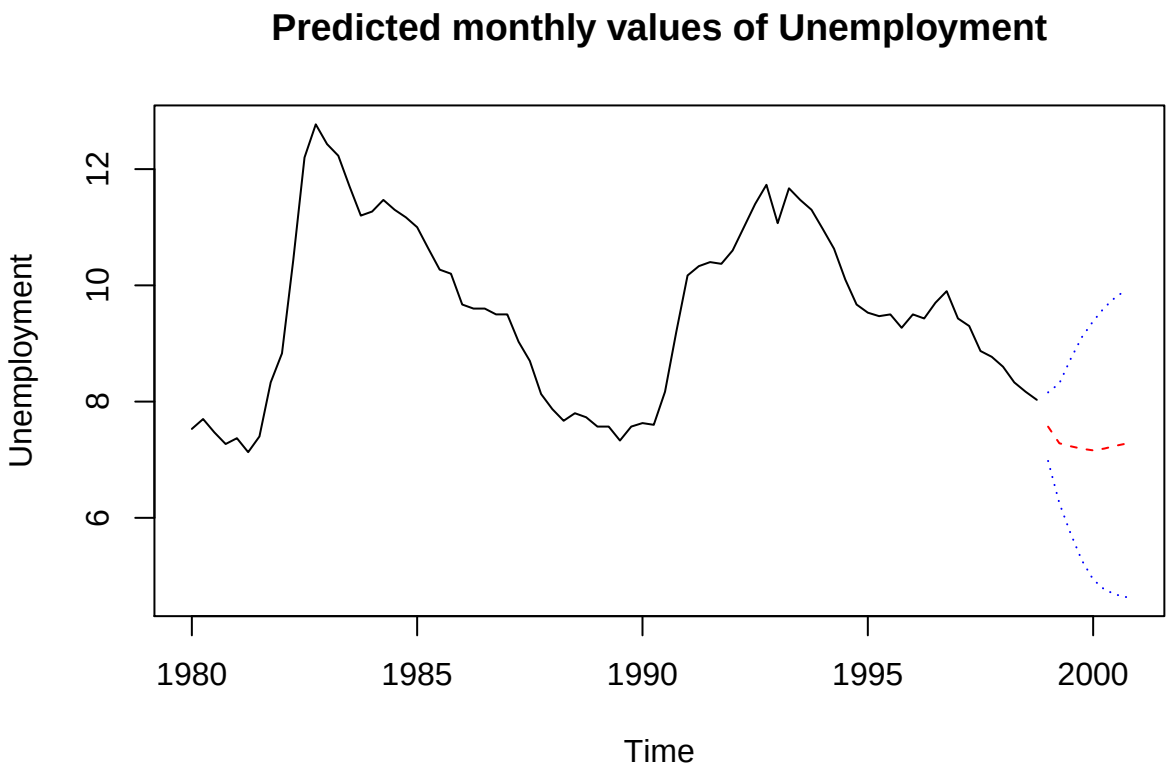



```
acf(resid(canada.var)^2)
```



- All the acf and ccf are white noise. - But there are volatility in acf^2 which may need GARCH model for residuals.

```
canada.predict = predict(canada.var, n.ahead = 8)
predict.val = ts(canada.predict$fcst$canada.U[,1], start = c(1999,1), end = c(2000,4), freq = 4)
ci.low = ts(canada.predict$fcst$canada.U[,2], start = c(1999,1), end = c(2000,4), freq = 4)
ci.high = ts(canada.predict$fcst$canada.U[,3], start = c(1999,1), end = c(2000,4), freq = 4)
ts.plot(cbind(canada.U, predict.val,ci.low, ci.high),
        lty=c(1,2,3,3), col=c("black", "red","blue","blue"),
        main="Predicted monthly values of Unemployment",
        ylab="Unemployment")
```



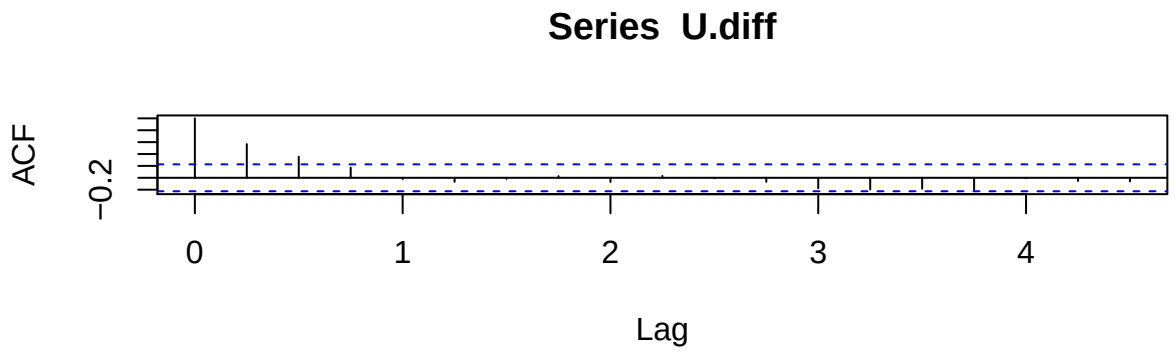

```
#RMSE of Canada Unemployment
rmse= function(x,y){
  a=sum((x-y)^2)
  b=length(x)
  rmse=sqrt(a/b)
  rmse
}
LT.var.rmse = rmse(canada.U.test,predict.val)

#MAD
mad = function(x, y){
  a = abs(sum(x - y))
  mad = a/length(a)
  mad
}
LT.var.mad = mad(canada.U.test,predict.val)

canada.predict = predict(canada.var, n.ahead = 4)
predict.val = ts(canada.predict$fcst$canada.U[,1], start = c(1999,1), end = c(1999,4), freq = 4)
ST.var.rmse = rmse(canada.U.test[1:4],predict.val)
ST.var.mad = mad(canada.U.test[1:4],predict.val)
print(c(LT.var.rmse, LT.var.mad, ST.var.rmse, ST.var.mad))
```

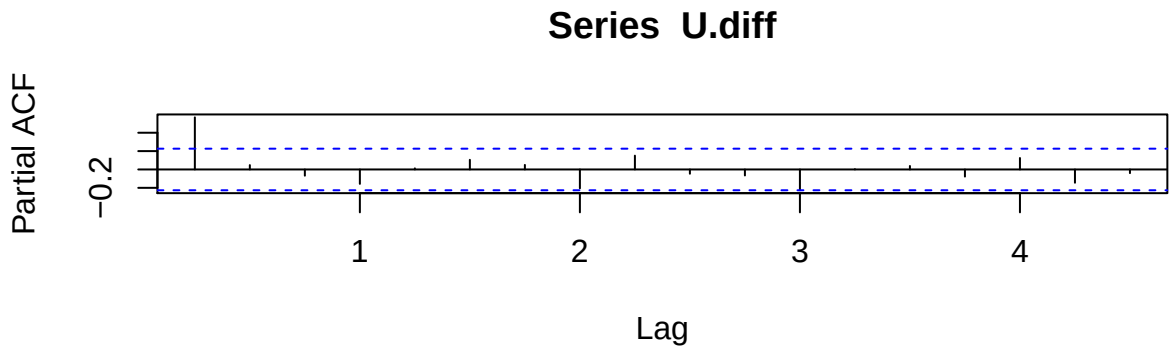
```
## [1] 0.4207964 2.0573132 0.2445786 0.6991225
```

- Predicted values fall into the confidence interval.
- In Short term prediction, RMSE gets lower.

#Predict by using ARIMA Model ##

```
par(mfrow = c(2,1))
acf(U.diff)
pacf(U.diff)
```



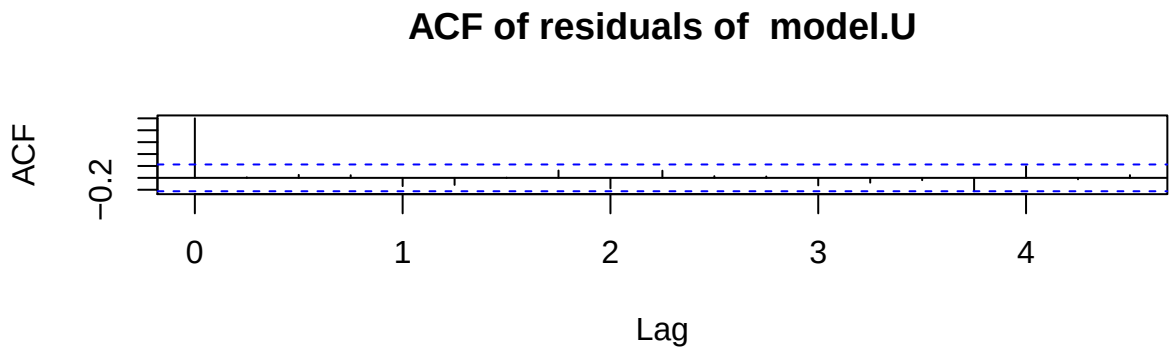


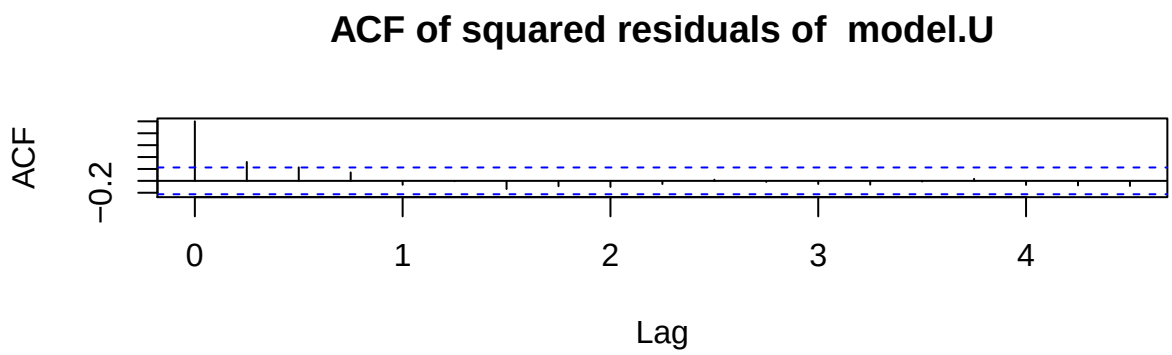

```
dev.off()
```

```
## null device
##           1
```

- In acf, autocorrelation dies dowm.
- I'm going to use AR(2) process for regular differencing.
- Pacf is white noise which is good.
- Not going to use any seasonal differencing.

```
model.U=arima(canada.U,order=c(2,1,0), seas=list(order=c(0,0,0),4))
par(mfrow = c(2,1))
acf(residuals(model.U),main="ACF of residuals of  model.U")
acf(residuals(model.U)^2,main="ACF of squared residuals of  model.U")
```






```
dev.off()
```

```
## null device
##           1
```

```
Box.test(residuals(model.U),lag=12,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(model.U)
## X-squared = 10.047, df = 12, p-value = 0.6118
```

```
model.U$coef
```

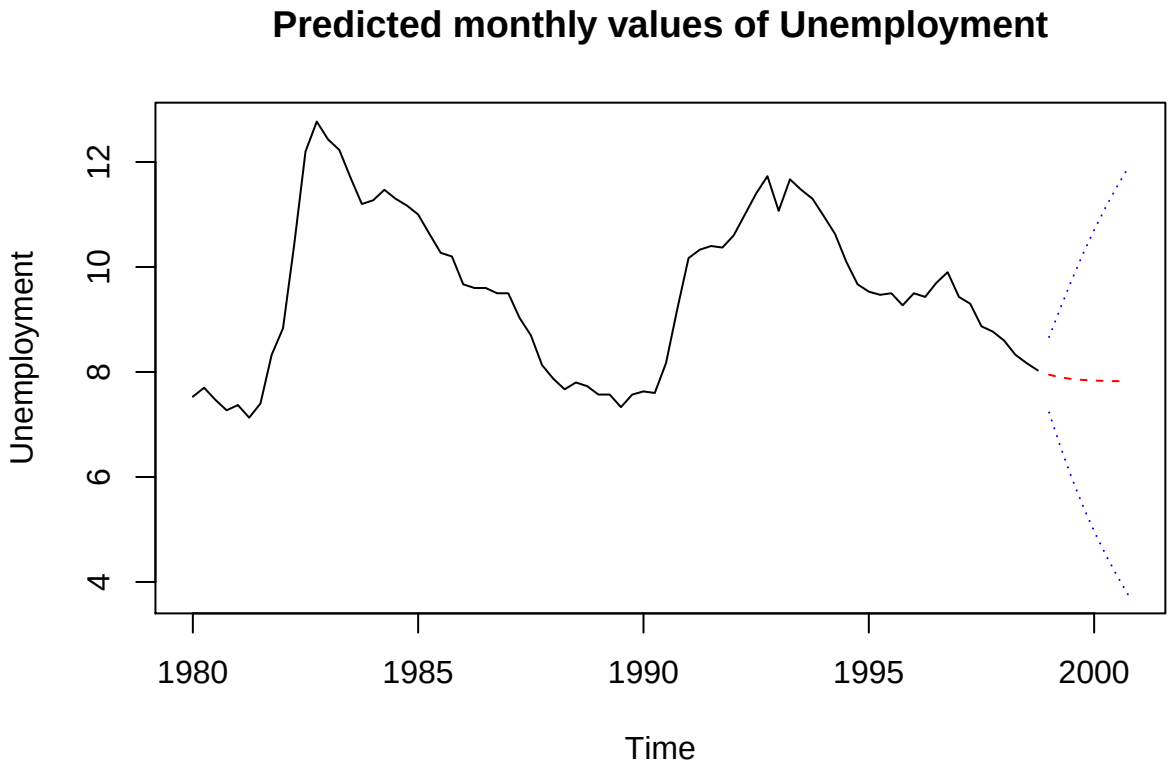
```
##        ar1        ar2
## 0.53583817 0.04517176
```

- In the acf, correlagram of residual of Arima model is white noise.
- There are no volatility in the acf^2.
- P-value of Ljung Box test is 0.61 which rejects the null hypothesis that says that there is an autocorrelation.
- $$Z_t = X_t - X_{t-1}$$
- $$(1 - 0.5358B - 0.0452B^2)Z_t = W_t$$

```
#AIC, and Sigma squared
model.U
```

```
##
## Call:
## arima(x = canada.U, order = c(2, 1, 0), seasonal = list(order = c(0, 0, 0),
##     4))
##
## Coefficients:
##          ar1     ar2
##       0.5358  0.0452
## s.e.  0.1148  0.1147
##
## sigma^2 estimated as 0.1344:  log likelihood = -31.35,  aic = 68.69
```

```r
#Estimate RMSE and MAD for ARIMA Model
LT.forecast = predict(model.U,8)$pred
LT.arima.rmse = rmse(canada.U.test,LT.forecast)
LT.arima.mad = mad(canada.U.test,LT.forecast)

LT.forecast.se = predict(model.U,8)$se
predict.val = ts(LT.forecast, start = c(1999,1), end = c(2000,4), freq = 4)
ci.low = ts(LT.forecast - 1.96*LT.forecast.se, start = c(1999,1), end = c(2000,4), freq = 4)
ci.high = ts(LT.forecast + 1.96*LT.forecast.se, start = c(1999,1), end = c(2000,4), freq = 4)
ts.plot(cbind(canada.U, predict.val,ci.low, ci.high),
        lty=c(1,2,3,3), col=c("black", "red","blue","blue"),
        main="Predicted monthly values of Unemployment",
        ylab="Unemployment")
```



```
ST.forecast = predict(model.U,4)$pred
ST.arima.rmse = rmse(canada.U.test[1:4],ST.forecast)
ST.arima.mad = mad(canada.U.test[1:4],ST.forecast)
```

- RMSE and MAD have lower value when I predict shorter term.
- This indicates predicting longer term will end up having massive error.

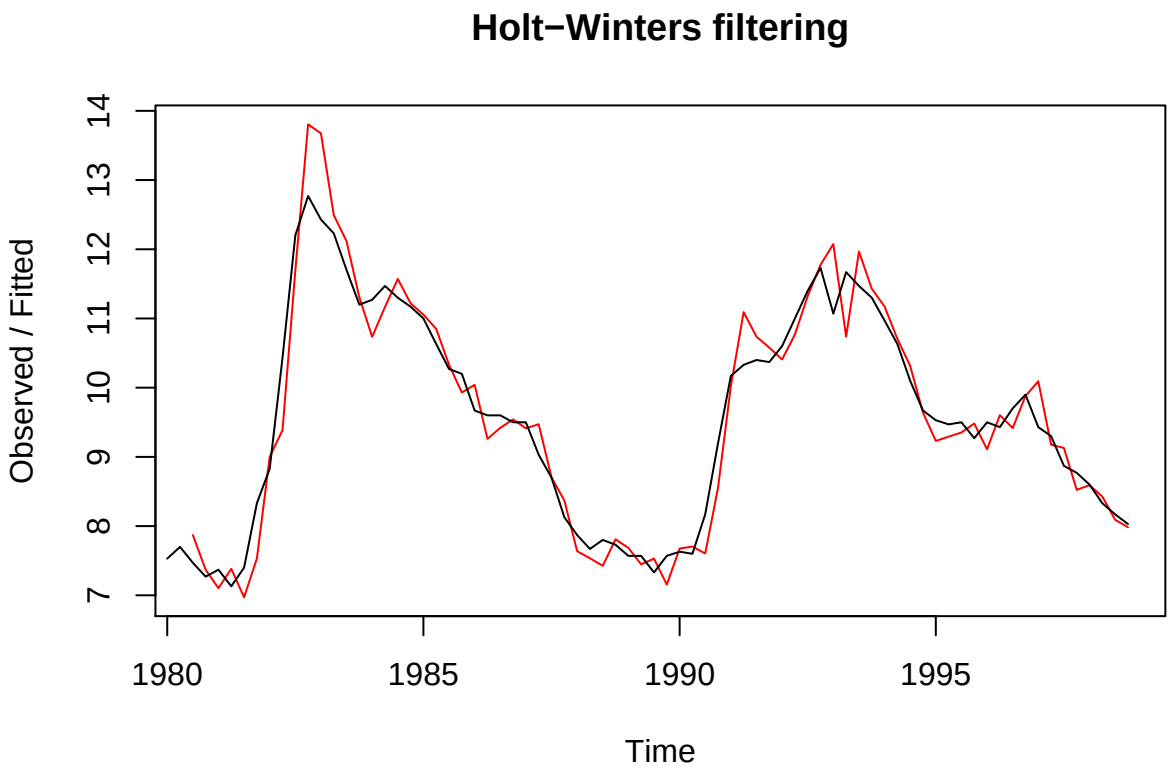
#Predict by using Exponential smoothing ##

```
es = HoltWinters(canada.U, gamma = FALSE)
es
```

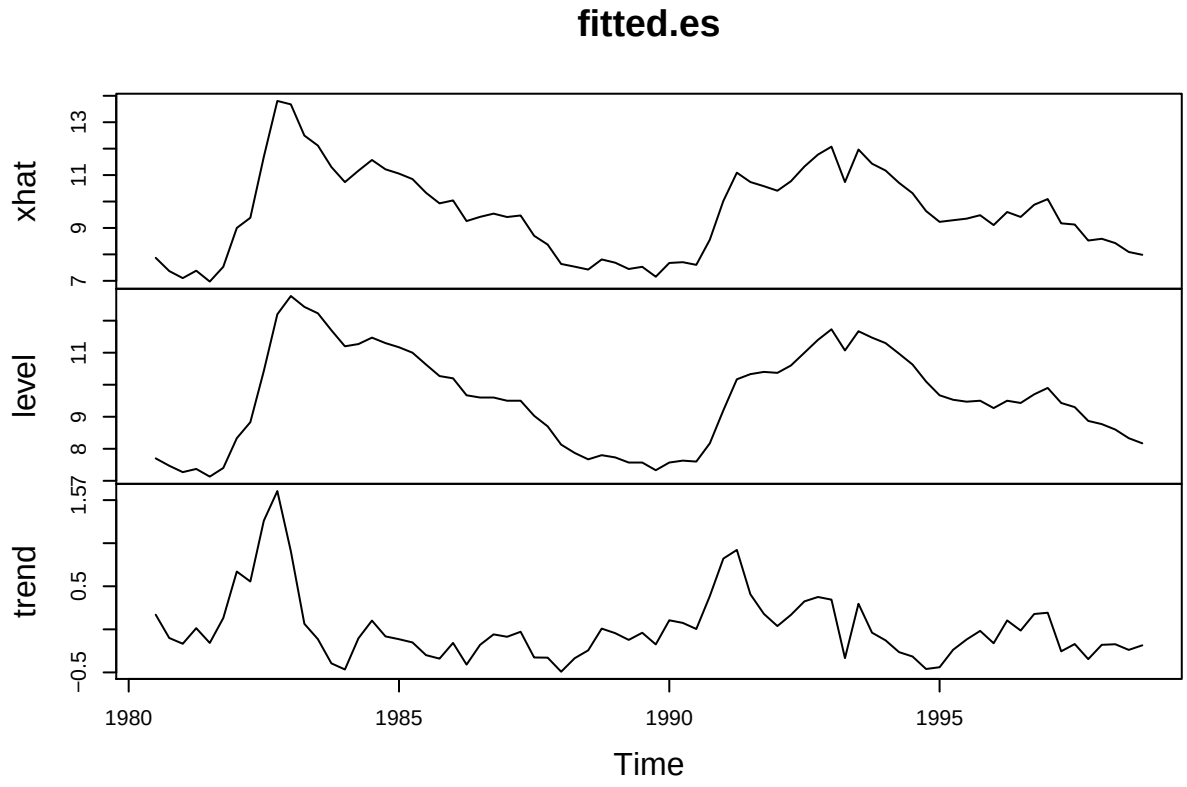
```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = canada.U, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0.6755709
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a  8.0300000
## b -0.1547598
```

- Since we first determine that regular differenced data has a white noise correlagram, there should be a trend in unemployment data.
- Alpha is for level and beta is for trend coefficient.
- By putting gamma = FALSE in the HoltWInters function, we make this funciton to optimize the value of alpha and beta.
- This is a trend correlated non-seasonal Exponential Smoothing.
- $$\alpha = 1, \beta = 0.6756$$
- $$Y_t = r_{0,t} + r_{1,t} * t + W_t$$
- $$\hat{r}_{0,t} = \hat{r}_{0,t-1} + \alpha(y_{t-1} - \hat{r}_{0,t-1}) + \hat{r}_{1,t-1}$$
- $$\hat{r}_{1,t} = 0.6756(\hat{r}_{0,t} - \hat{r}_{0,t-1}) + (1 - 0.6756)\hat{r}_{1,t-1}$$
- $$\hat{y}_t = \hat{r}_{0,t-1} + 1 * (y_{t-1} - \hat{r}_{0,t-1}) * \hat{r}_{1,t-1} + (0.6756 * (\hat{r}_{0,t} - \hat{r}_{0,t-1}) + (1 - 0.6756) * \hat{r}_{1,t-1}) * t + W_t$$

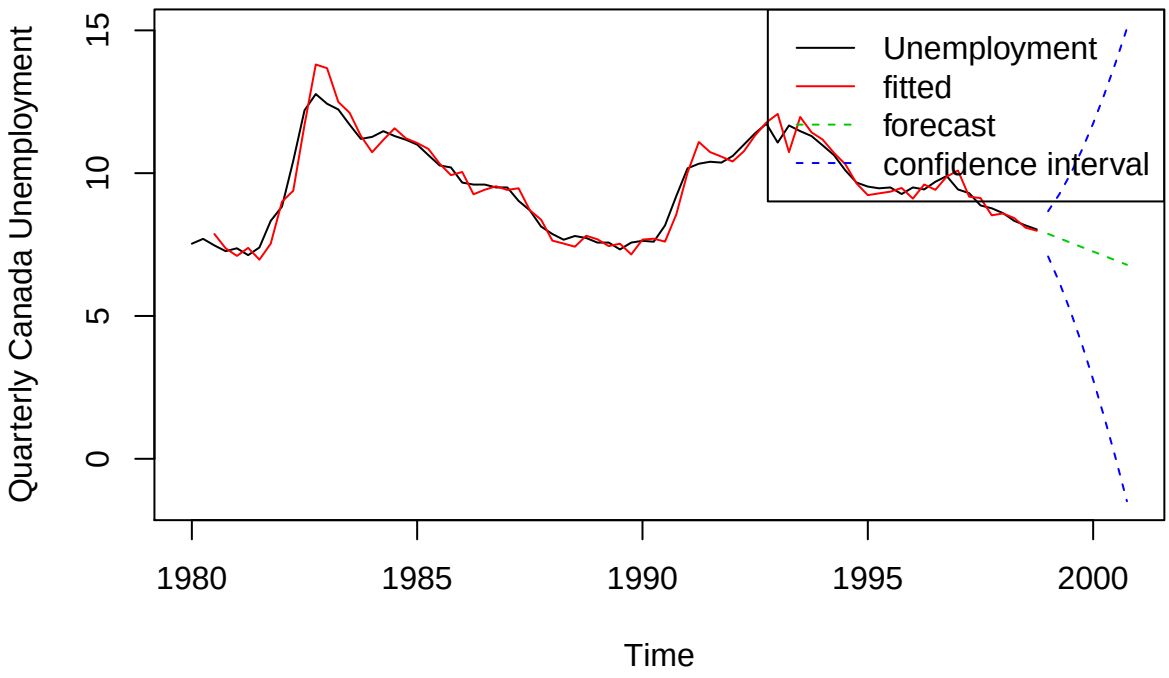
```
plot(es)
```




```
fitted.es = fitted(es)
plot(fitted.es)
```




```
LT.forecast.es= predict(es, n.ahead = 8, prediction.interval = TRUE)
ts.plot(cbind(canada.U,fitted.es[,1]), LT.forecast.es,lty=c(1,1,2,2,2), col=c(1,2,3,4,4), ylab="Quarterl
legend("topright", legend=c("Unemployment","fitted","forecast","confidence interval"), lty=c(1,1,2,2),
```




```
LT.exp.rmse = rmse(canada.U.test,LT.forecast.es[,1])
LT.exp.mad = mad(canada.U.test,LT.forecast.es[,1])

ST.forecast.es = predict(es, n.ahead = 4)
ST.exp.rmse = rmse(canada.U.test[1:4],ST.forecast.es[,1])
ST.exp.mad = mad(canada.U.test[1:4],ST.forecast.es[,1])
```

- By using level and trend in holtwinter funciton, it predicts well on the train data.
- I usually thought of linear trend throughout the time plot.
- Trend seems unusual for me, but there is an obvious trend in the data.

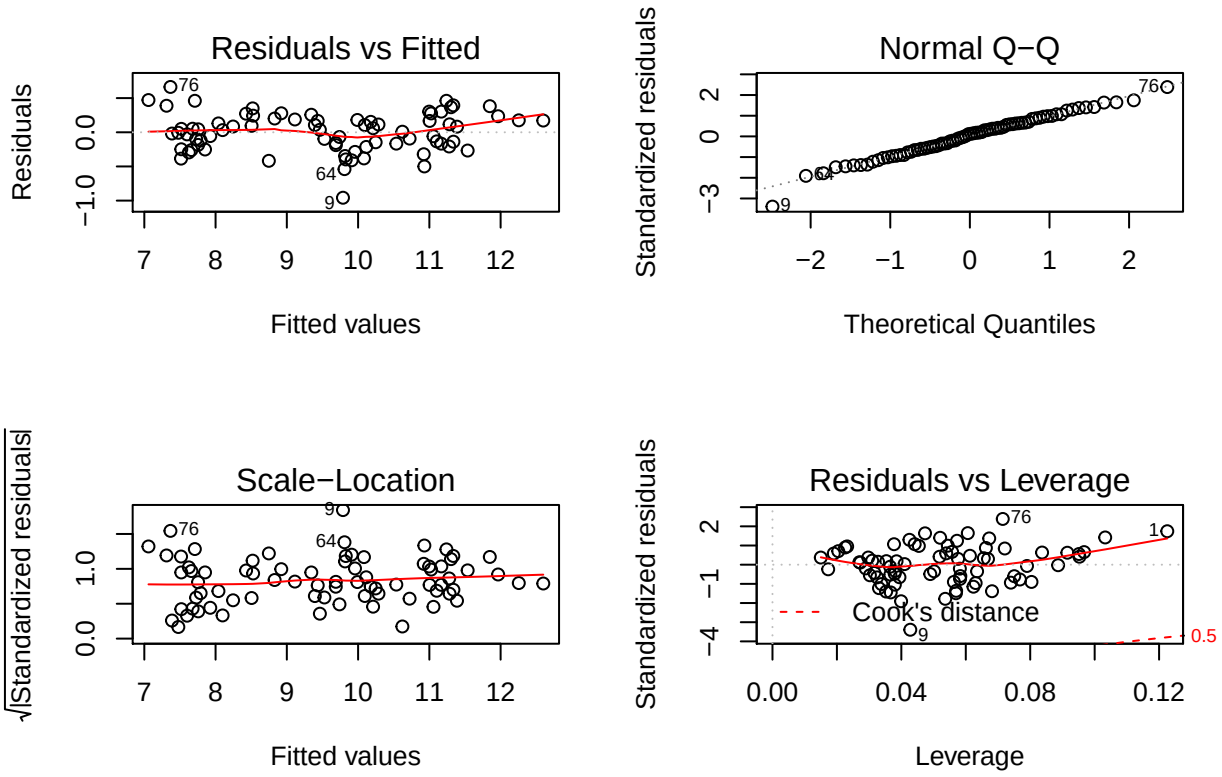
#Predict by using GLS ##

```
canada.lm = lm(canada.U ~ canada.e + canada.prod + canada.rw)
summary(canada.lm)
```

```
##
## Call:
## lm(formula = canada.U ~ canada.e + canada.prod + canada.rw)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95788 -0.18555  0.03382  0.18181  0.66311
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 400.786813   8.944145   44.81   <2e-16 ***
## canada.e     -0.492547   0.013776  -35.75   <2e-16 ***
```

```
## canada.prod  -0.005879   0.017293   -0.34    0.735
## canada.rw     0.172301   0.004061   42.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2886 on 72 degrees of freedom
## Multiple R-squared:  0.9653, Adjusted R-squared:  0.9638
## F-statistic: 666.9 on 3 and 72 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(canada.lm)
```

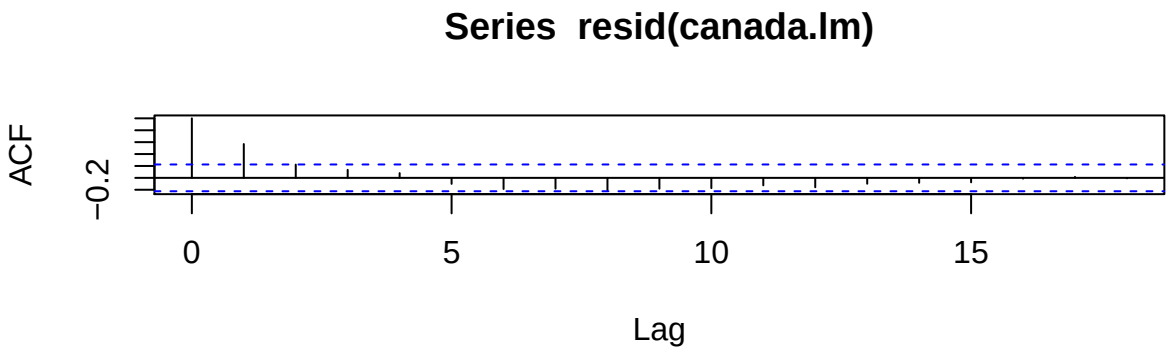


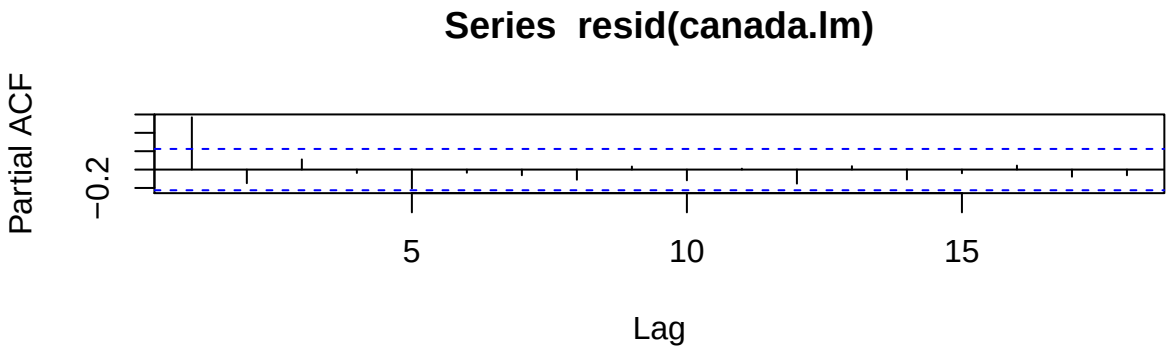

```
dev.off()
```

```
## null device
##           1
```

- Except for productivity, other variables have significant coefficient.
- Residuals are not constant due to the first plot which means that we need additional modeling for residuals.

```
par(mfrow = c(2,1))
acf(resid(canada.lm))
acf(resid(canada.lm), type = "partial")
```



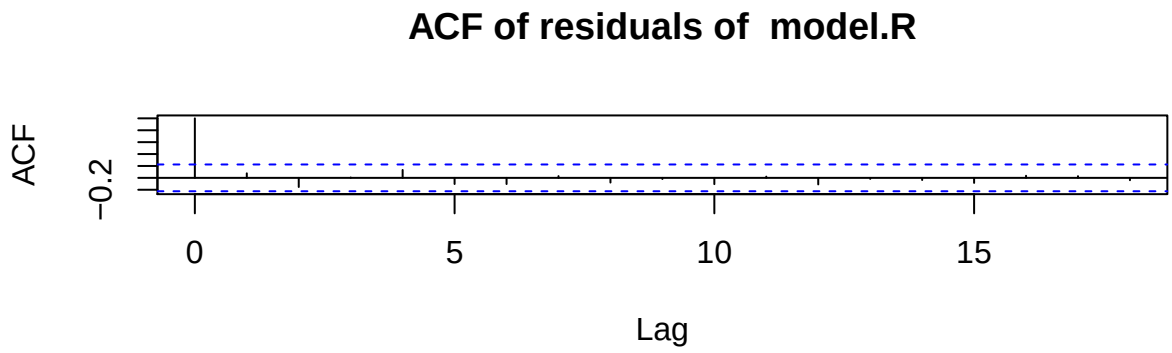


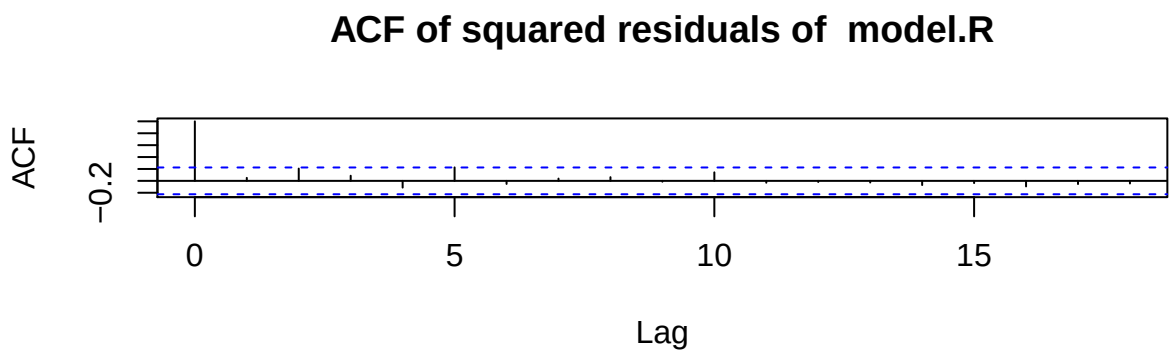

```
dev.off()
```

```
## null device
##           1
```

- In the acf, r1 is significant spike.
- Autocorrelation dies down in acf so I'm going to use AR(1) model for the residual.

```
model.R=arima(resid(canada.lm),order=c(1,0,0), seas=list(order=c(0,0,0),4), include.mean = FALSE)
par(mfrow = c(2,1))
acf(residuals(model.R),main="ACF of residuals of  model.R")
acf(residuals(model.R)^2,main="ACF of squared residuals of  model.R")
```






```
dev.off()
```

```
## null device
##           1
```

```
Box.test(residuals(model.R),lag=12,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(model.R)
## X-squared = 8.4501, df = 12, p-value = 0.749
```

```
model.R  #Find coefficient
```

```
##
## Call:
## arima(x = resid(canada.lm), order = c(1, 0, 0), seasonal = list(order = c(0,
##     0, 0), 4), include.mean = FALSE)
##
## Coefficients:
##          ar1
##       0.6295
## s.e.  0.0953
##
## sigma^2 estimated as 0.05021:  log likelihood = 5.59,  aic = -7.18
```

```r
## Durbin Watson tests
library(lmtest)
dwtest(canada.lm,alternative=c("greater"))
```

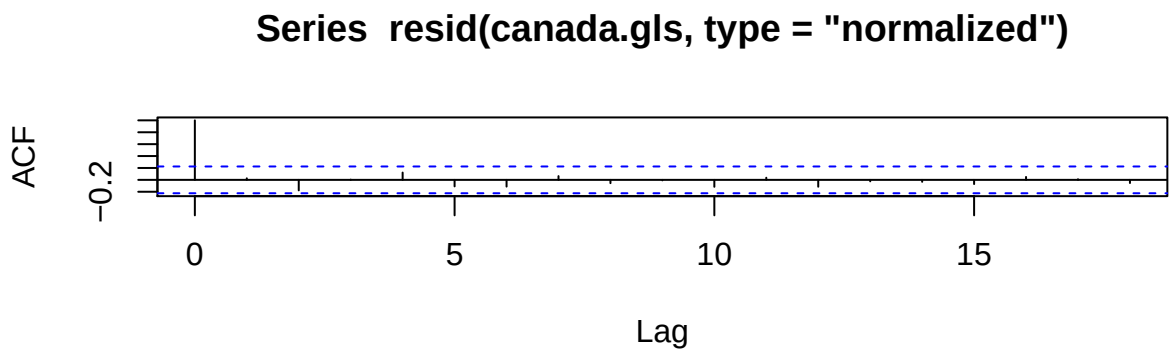
```
##
##  Durbin-Watson test
##
## data:  canada.lm
## DW = 0.75212, p-value = 3.336e-11
## alternative hypothesis: true autocorrelation is greater than 0
```

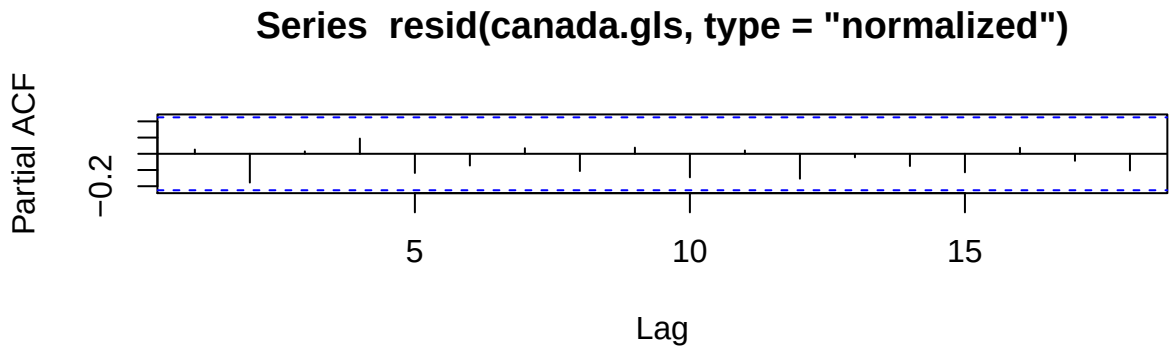
- Both acf and acf^2 show us that the residuals are finally become white noise.
- In Ljung Box test, H0 is rejected so that the residuals are white noise.
- Now we have a coefficient of AR(1), and I'm goint to put it into GLS.
- Durbin Watson tells us that the residuals are AR(1) process.

```r
library(nlme)
canada.gls = gls(canada.U ~ canada.e + canada.prod + canada.rw, correlation=corARMA(c(0.6295), p=1))
summary(canada.gls)
```

```
## Generalized least squares fit by REML
##   Model: canada.U ~ canada.e + canada.prod + canada.rw
##   Data: NULL
##        AIC      BIC    logLik
##   23.68173 37.34173 -5.840866
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi
## 0.7357951
##
## Coefficients:
##                Value Std.Error    t-value p-value
## (Intercept) 385.4141 19.853017  19.413376  0.0000
## canada.e     -0.4764  0.028209 -16.887521  0.0000
## canada.prod   0.0029  0.032793   0.089083  0.9293
## canada.rw     0.1645  0.008825  18.638461  0.0000
##
##  Correlation:
##             (Intr) canad. cnd.pr
## canada.e    -0.846
## canada.prod -0.043 -0.490
## canada.rw    0.837 -0.842  0.143
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.9472777 -0.5920748 -0.1802029  0.5769656  1.6798353
##
## Residual standard error: 0.3395218
## Degrees of freedom: 76 total; 72 residual
```

```r
par(mfrow = c(2,1))
acf(resid(canada.gls, type = "normalized"))
pacf(resid(canada.gls, type = "normalized"))
```



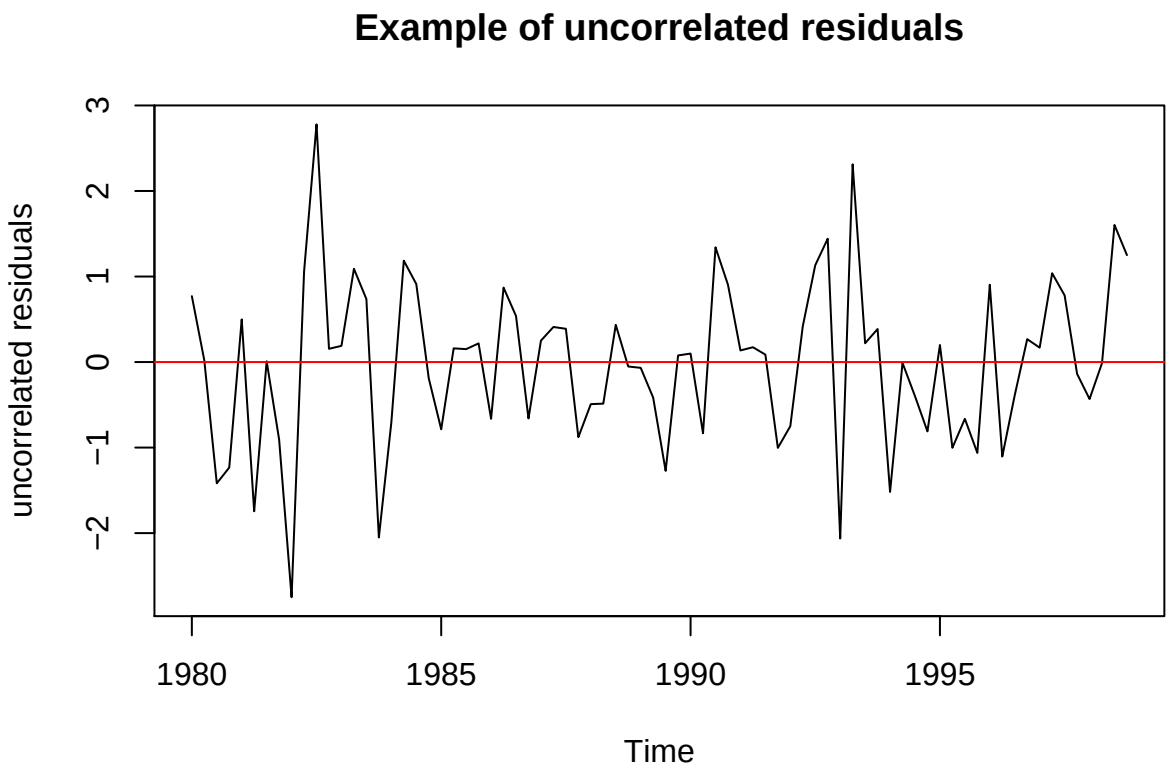



```
dev.off()
```

```
## null device
##           1
```

- By using GLS modeling, residuals have been changed to white noise.
- $$U_t = 385.4141 - 0.4764 * e_t + 0.0029 * prod_t + 0.16458 * rw_t + \hat{W}_t$$
- $$\hat{W}_t = 0.6295 * \hat{W}_{t-1}$$

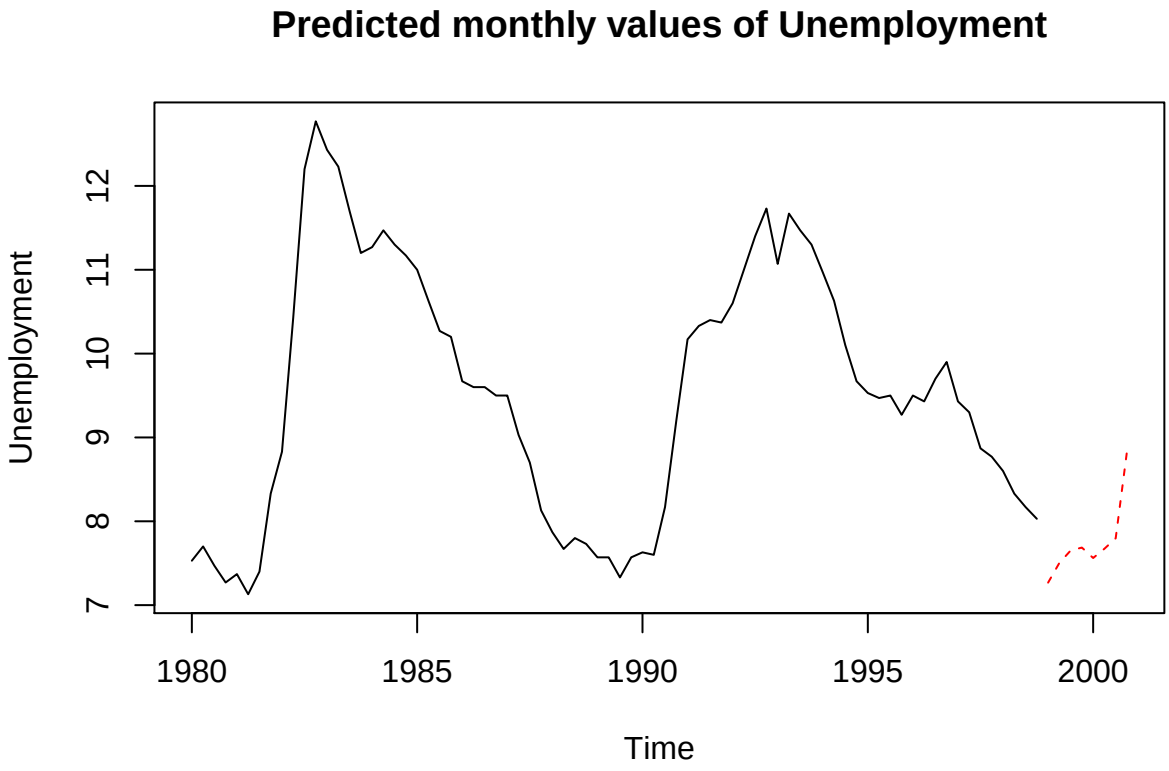
```
#Create dataframe of test dataset with only independent variable
test.independent = data.frame(canada.e = test[,1], canada.prod = test[,2], canada.rw = test[,3])

plot(y=residuals(canada.gls,type="normalized"),
     x=as.vector(time(canada.U)), type="l",
     ylab="uncorrelated residuals",
     xlab="Time",
     main="Example of uncorrelated residuals")
abline(h=0, col = 2)
```




```
?predict
```

```
## Help on topic 'predict' was found in the following packages:
##
##   Package               Library
##   vars                  /Users/josh/Library/R/3.6/library
##   stats                 /Library/Frameworks/R.framework/Versions/3.6/Resources/library
##
##
## Using the first match ...
```

```
LT.forecast = predict(canada.gls, newdata = test.independent)
predict.val = ts(LT.forecast, start = c(1999,1), end = c(2000,4), freq = 4)
ci.low = ts(LT.forecast - 1.96*LT.forecast.se, start = c(1999,1), end = c(2000,4), freq = 4)
ci.high = ts(LT.forecast + 1.96*LT.forecast.se, start = c(1999,1), end = c(2000,4), freq = 4)
ts.plot(cbind(canada.U, predict.val),
        lty=c(1,2), col=c("black", "red"),
        main="Predicted monthly values of Unemployment",
        ylab="Unemployment")
```




```
LT.gls.rmse = rmse(canada.U.test,LT.forecast)
LT.gls.mad = mad(canada.U.test,LT.forecast)


ST.forecast = predict(canada.gls, newdata = test.independent[1:4,])
ST.gls.rmse = rmse(canada.U.test[1:4],ST.forecast)
ST.gls.mad = mad(canada.U.test[1:4],ST.forecast)
```

- In the residual plot, all the residuals seems to be unrelated with previous and post residuals.
- Residual is constant so I'm good to use this model.
- Predicted value seems to follow the pattern of our unemployment dataset.

```
print(c(LT.var.rmse, LT.var.mad, ST.var.rmse, ST.var.mad))
```

```
## [1] 0.4207964 2.0573132 0.2445786 0.6991225
```

```
print(c(LT.arima.rmse, LT.arima.mad, ST.arima.rmse, ST.arima.mad))
```

```
## [1] 0.4808745 2.6735816 0.4197537 1.5896662
```

```
print(c(LT.exp.rmse, LT.exp.mad, ST.exp.rmse, ST.exp.mad))
```

```
## [1] 0.5845503 1.5313540 0.1927709 0.6024017
```

```
print(c(LT.gls.rmse, LT.gls.mad, ST.gls.rmse, ST.gls.mad))
```

```
## [1] 0.3628742 1.7717337 0.2801091 0.1393762
```

- For the long term prediction, GLS modeling has lowest RMSE and Exponential Smoothing has lowest MAD.

- For the short term prediction, Exponential Smoothing has lowest RMSE and GLS has lowest MAD.
- Long term prediction tend to have higher error rate because it's predicting from the predicted value.
- Prediction confidence interval shows that the interval is getting larger and larger as the predicting period goes by.
- It is a good idea to maximize the train data and minimize the test data.
- For the unemployment prediction, GLS seems to be the proper one to use.