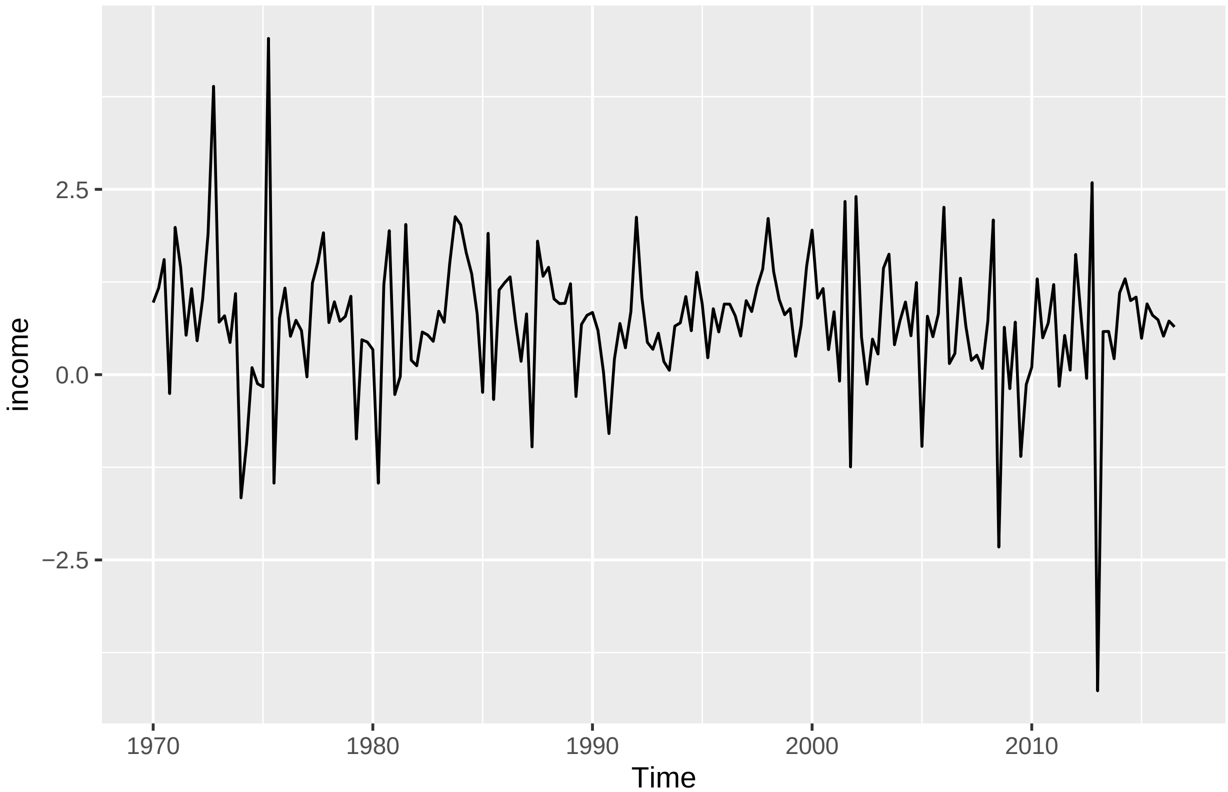
uschange

Julien JACQUES

2/19/2020

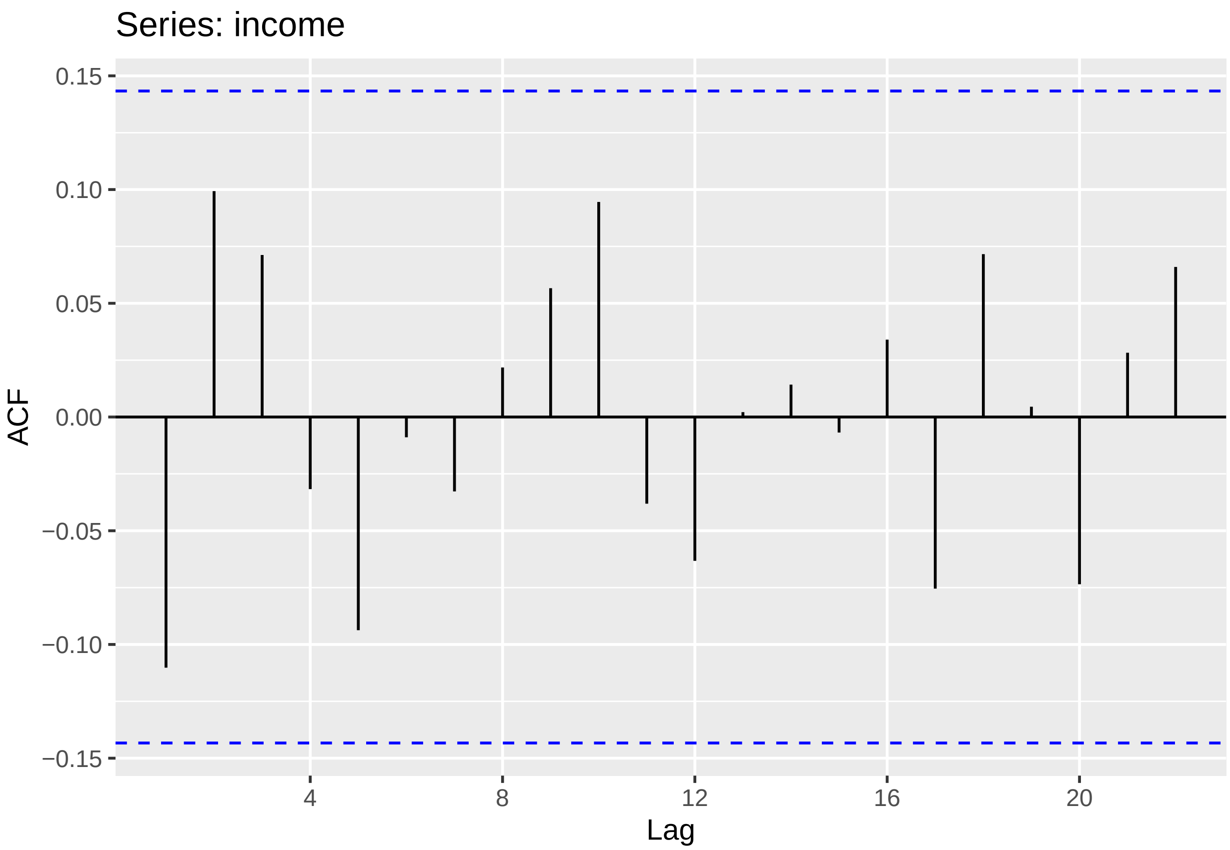
Income

```
library(fpp2)
income=uschange[,"Income"]
autoplot(income)
```

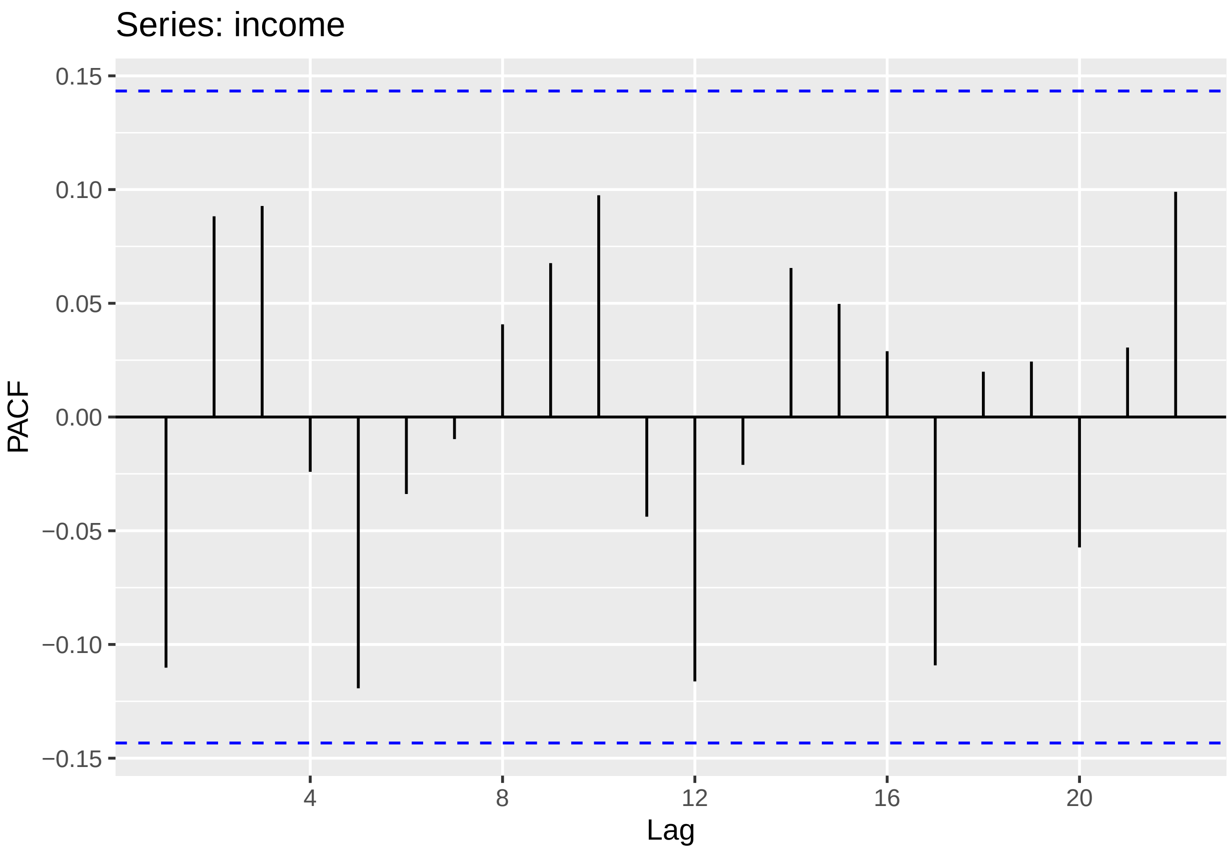


We see no trend and no seasonal pattern... We can check the ACF

```
ggAcf(income)
```




```
ggPacf(income)
```



It seems to be no correlation or auto-correlation... We can check that with a Box.test:

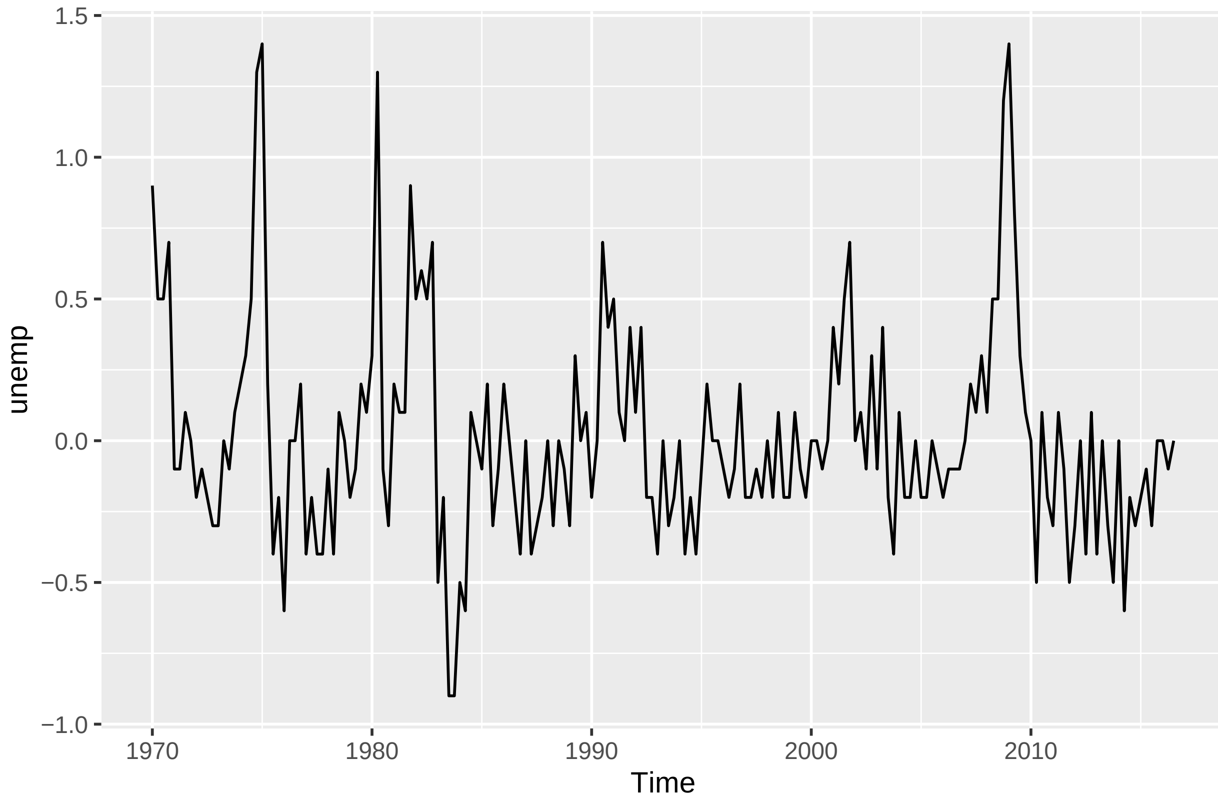
```
Box.test(income,lag=10,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  income
## X-squared = 9.8076, df = 10, p-value = 0.4575
```

Indeed, no reason to reject the fact that it is a white noise : the best forecasting model will then be a simple exponential smoothing !

Unemployment

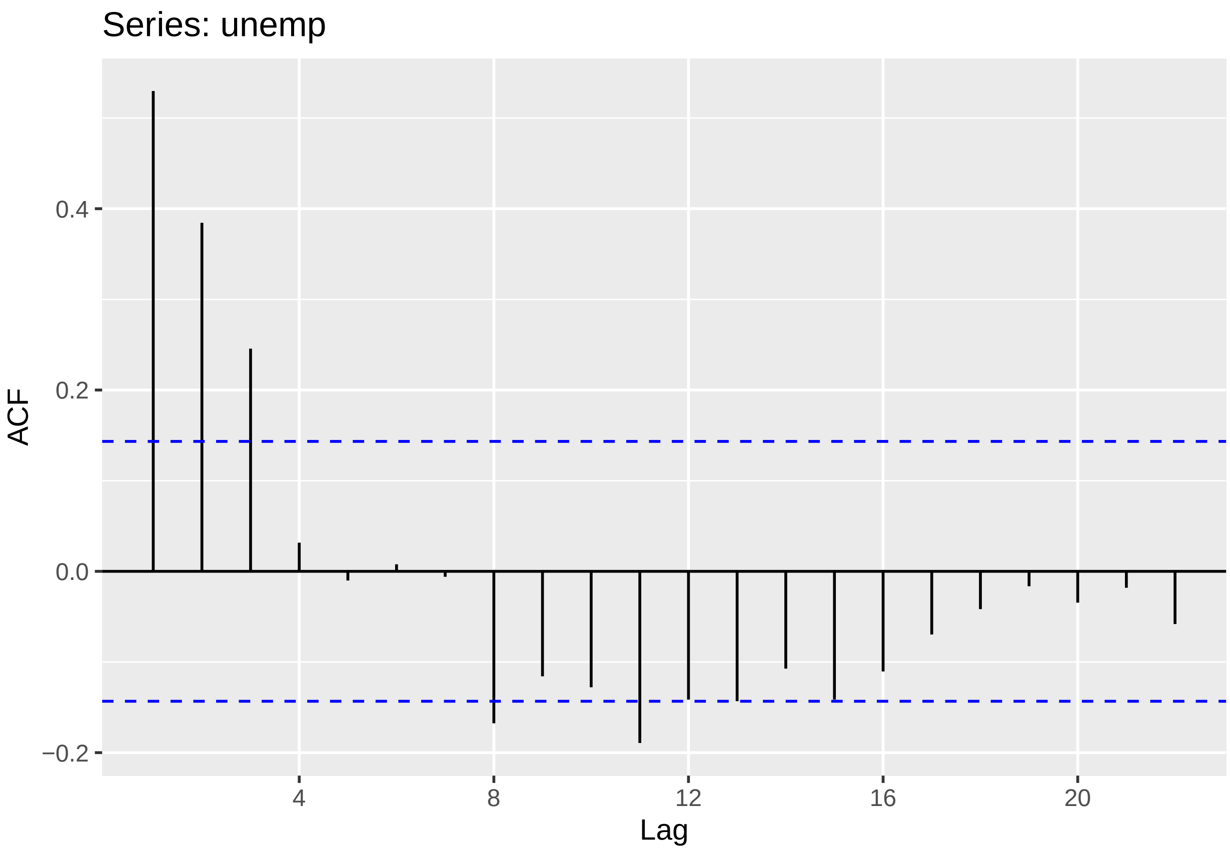
```
library(fpp2)
unemp=uschange[,"Unemployment"]
autoplot(unemp)
```



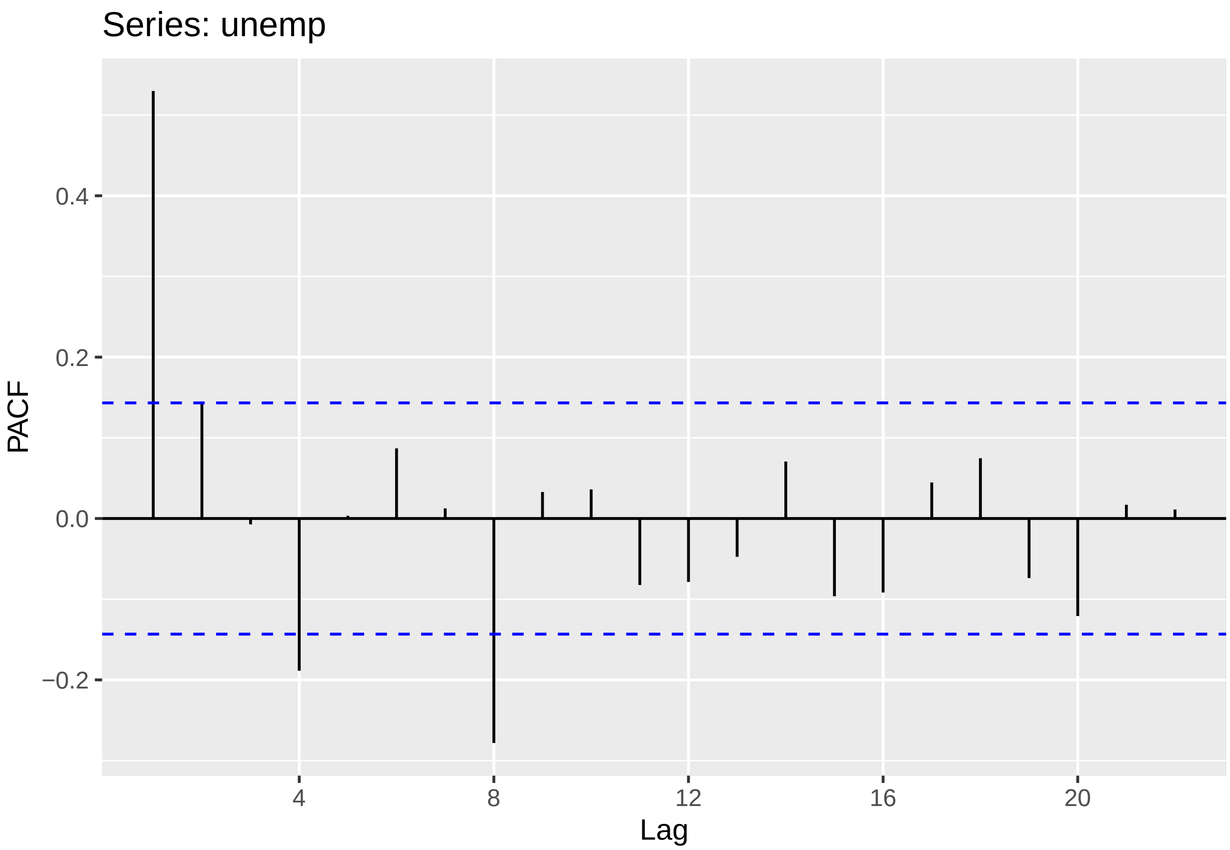

```r
Box.test(unemp,lag=10,type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  unemp
## X-squared = 104.88, df = 10, p-value < 2.2e-16
```

```r
ggAcf(unemp)
```




```
ggPacf(unemp)
```



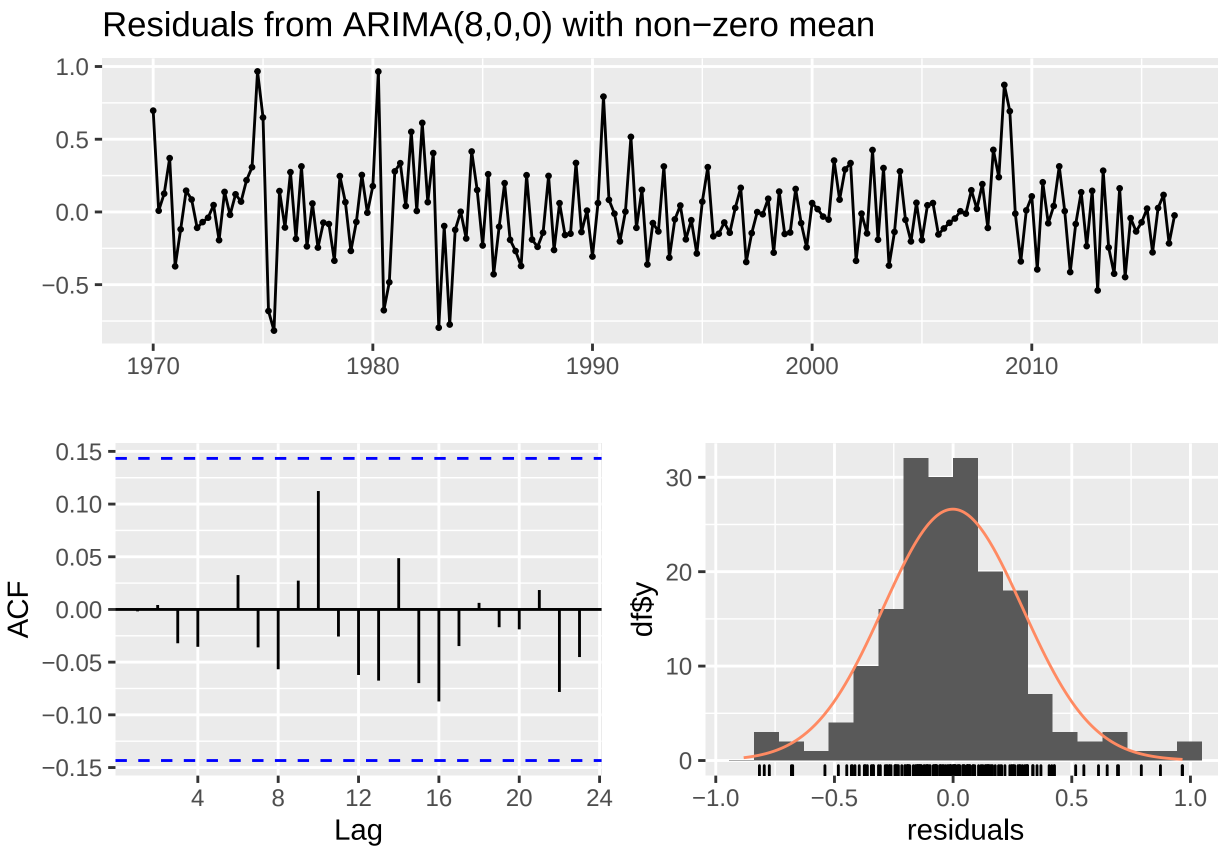
Maybe an AR_8 ?

```
ar8=Arima(unemp,order=c(8,0,0))
print(ar8)
```

```
## Series: unemp
## ARIMA(8,0,0) with non-zero mean
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ar5     ar6     ar7      ar8    mean
##       0.4627  0.2128  0.0654  -0.2486  -0.0290  0.1320  0.1399  -0.2792  0.0097
## s.e.  0.0702  0.0774  0.0779   0.0791   0.0789  0.0789  0.0776   0.0705  0.0396
##
## sigma^2 estimated as 0.09022:  log likelihood=-36.43
## AIC=92.86   AICc=94.11   BIC=125.17
```

We can check the residual

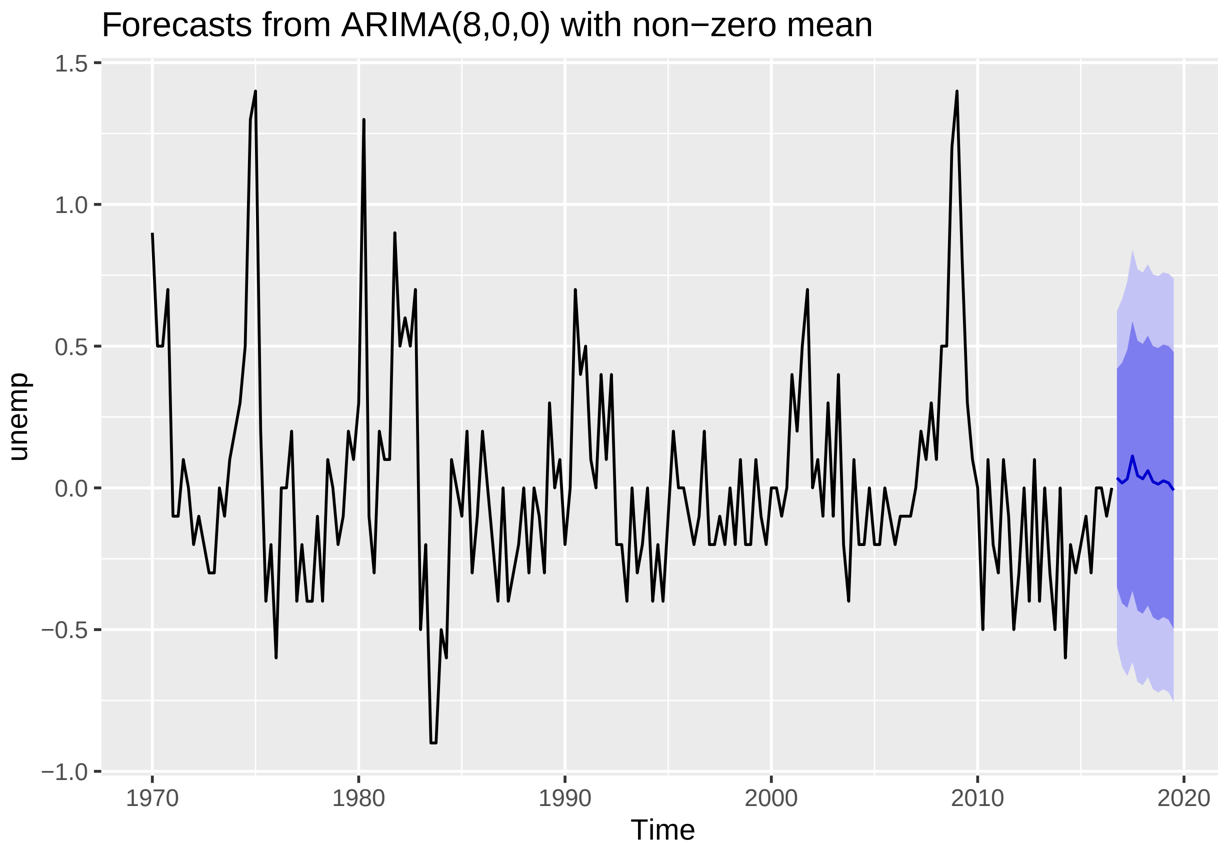
```
checkresiduals(ar8)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(8,0,0) with non-zero mean
## Q* = 5.1242, df = 3, p-value = 0.1629
##
## Model df: 9.   Total lags used: 12
```

The residual are a white noise, we can keep this model for our forecast.

```
autoplot(forecast(ar8,h=12))
```



All 5 series togheter

```
usc_train=window(uschange,start=c(1970,1),end=c(2015,4))
usc_test=window(uschange,start=c(2016,1),end=c(2016,3))
```

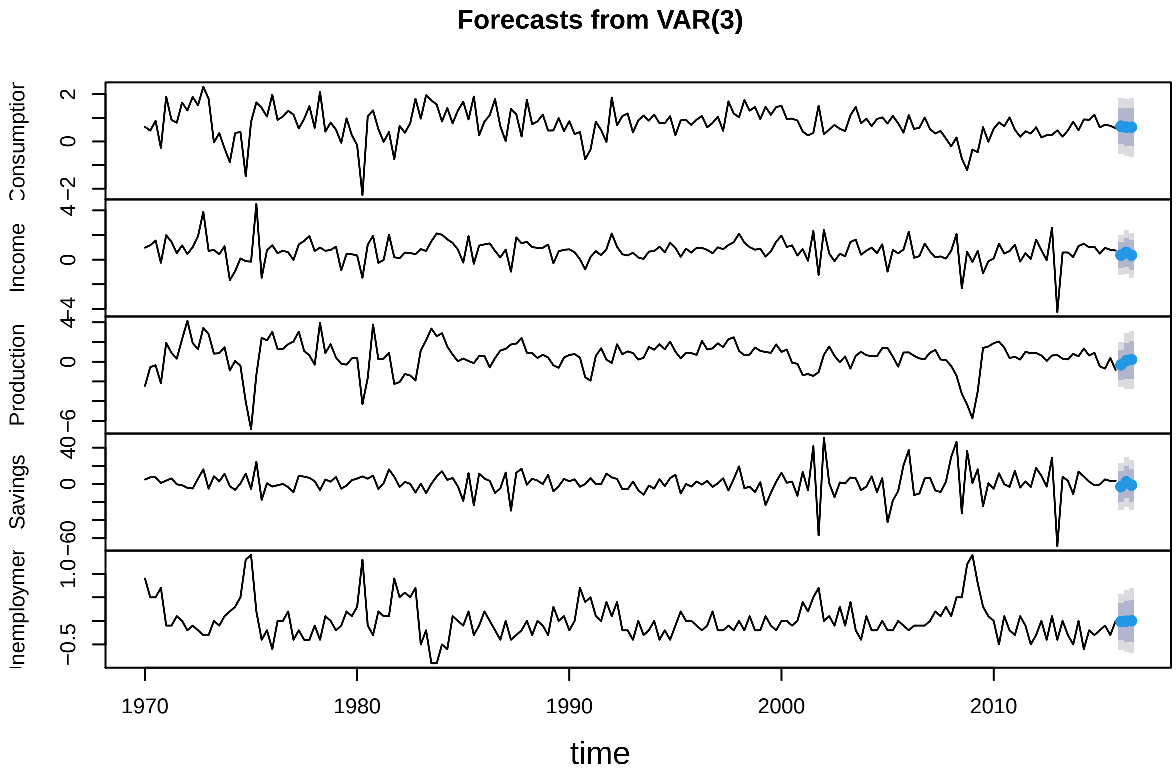
We use Vectorial Auto-Regressive models. We choose the order with VARselect function :

```
library(vars)
VARselect(usc_train, lag.max=10, type="const")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      3      1      1      3
##
## $criteria
##                 1          2          3           4          5          6
## AIC(n) -0.7807689 -0.8480625 -0.8551580 -0.79199008 -0.7919826 -0.6439896
## HQ(n)  -0.5598194 -0.4429884 -0.2659594 -0.01866691  0.1654651  0.4975827
## SC(n)  -0.2361041  0.1504895  0.5972812  1.11433639  1.5682311  2.1701114
## FPE(n)  0.4581164  0.4286055  0.4263251  0.45562807  0.4580806  0.5353834
##                 7          8          9         10
## AIC(n) -0.7627672 -0.7118667 -0.6274209 -0.4903386
## HQ(n)   0.5629297  0.7979547  1.0665251  1.3877320
## SC(n)   2.5052211  3.0100088  3.5483419  4.1393114
## FPE(n)  0.4807233  0.5133774  0.5694339  0.6691194
```

AIC leads to select an order equal to 3:

```r
var <- VAR(usc_train, p=3,type = "const")
prev=forecast(var,h=3)
plot(prev)
```




```r
print(sqrt(mean((prev$forecast$Consumption$mean-usc_test[,"Consumption"])^2)))
```

```
## [1] 0.2998763
```

```r
print(sqrt(mean((prev$forecast$Income$mean-usc_test[,"Income"])^2)))
```

```
## [1] 0.19955
```

```r
print(sqrt(mean((prev$forecast$Production$mean-usc_test[,"Production"])^2)))
```

```
## [1] 0.2310848
```

```r
print(sqrt(mean((prev$forecast$Savings$mean-usc_test[,"Savings"])^2)))
```

```
## [1] 4.222851
```

```r
print(sqrt(mean((prev$forecast$Unemployment$mean-usc_test[,"Unemployment"])^2)))
```

```
## [1] 0.05563519
```

We could compare to univariate SARIMA models…

```r
fit <- auto.arima(usc_train[,"Consumption"])
prev=forecast(fit,h=3)
print(sqrt(mean((prev$mean-usc_test[,"Consumption"])^2)))
```

```
## [1] 0.283259
```

```r
fit <- auto.arima(usc_train[,"Income"])
prev=forecast(fit,h=3)
```

```
print(sqrt(mean((prev$mean-usc_test[,"Income"])^2)))
```

```
## [1] 0.1232476
```

```
fit <- auto.arima(usc_train[,"Production"])
prev=forecast(fit,h=3)
print(sqrt(mean((prev$mean-usc_test[,"Production"])^2)))
```

```
## [1] 0.1566074
```

```
fit <- auto.arima(usc_train[,"Savings"])
prev=forecast(fit,h=3)
print(sqrt(mean((prev$mean-usc_test[,"Savings"])^2)))
```

```
## [1] 2.766953
```

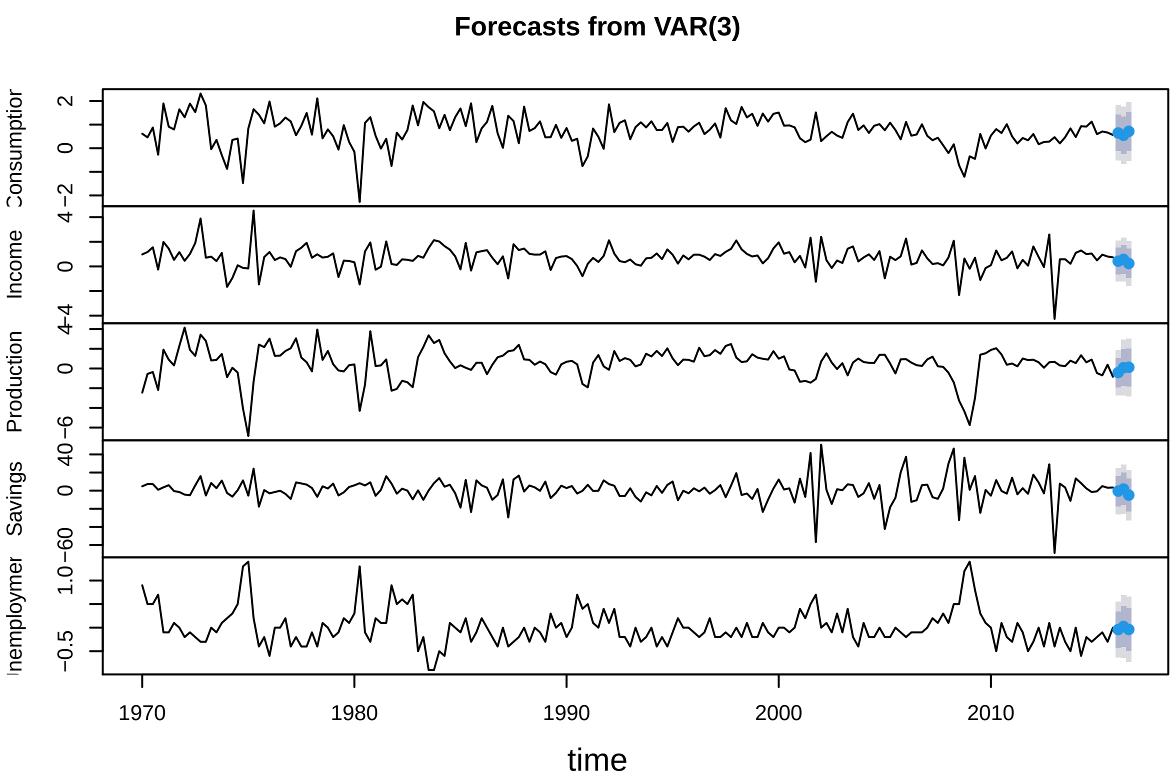
```
fit <- auto.arima(usc_train[,"Unemployment"])
prev=forecast(fit,h=3)
print(sqrt(mean((prev$mean-usc_test[,"Unemployment"])^2)))
```

```
## [1] 0.1198583
```

The results are better with univariate time series ! Probably because we have more flexible models for univariate models (MA part...)

Warning it is possible to add a seasonal pattern in VAR function

```
var <- VAR(usc_train, p=3,type = "const",season=4)
prev=forecast(var,h=3)
plot(prev)
```




```
print(sqrt(mean((prev$forecast$Consumption$mean-usc_test[,"Consumption"])^2)))
```

```
## [1] 0.3204709
```

```r
print(sqrt(mean((prev$forecast$Income$mean-usc_test[,"Income"])^2)))
```

```
## [1] 0.2573874
```

```r
print(sqrt(mean((prev$forecast$Production$mean-usc_test[,"Production"])^2)))
```

```
## [1] 0.2612618
```

```r
print(sqrt(mean((prev$forecast$Savings$mean-usc_test[,"Savings"])^2)))
```

```
## [1] 3.964729
```

```r
print(sqrt(mean((prev$forecast$Unemployment$mean-usc_test[,"Unemployment"])^2)))
```

```
## [1] 0.07874742
```

But the results are not significantly better…