

Homework1

2026-01-17

```
suppressPackageStartupMessages({  
  library("TSA")  
  library("forecast")  
  library("urca")  
  library("tseries")  
  library("lmtest")  
  library("astsa")  
  library("seasonal")  
})  
options(scipen = 1000)
```

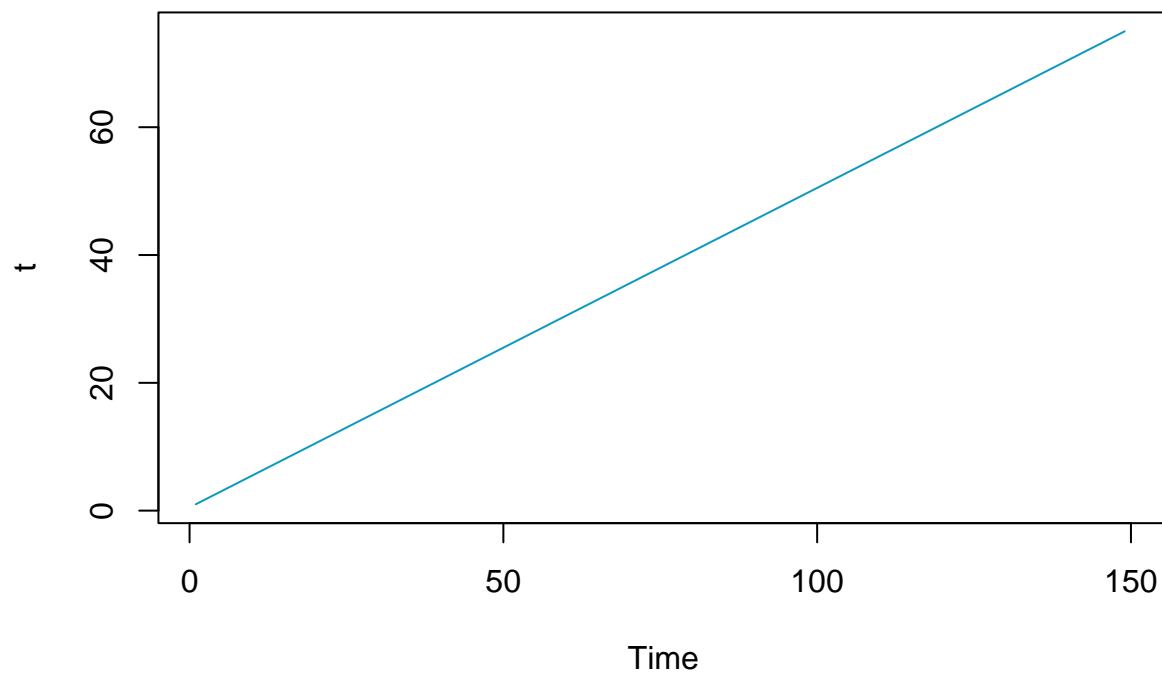
1. Task

Using R code with the cycle, changing and selecting parameters, generate all types of the processes (unit positive, unit negative, stable positive, stable negative, unstable positive, unstable negative) with/without stochastic component based on AR(1) and AR(2)

1) Unit positive ($a_1 = 1$): $Y_t = Y_{t-1} + a_0$

$$Y_t = Y_{t-1} + 0.5, Y_0 = 1$$

```
e <- 150  
t <- rep(NA, e) #creating a vector for time series values  
b <- 2  
t[1]=1  
while (b<e)  
{t[b] <- t[b-1] + 0.5  
  b <- b + 1}  
ts.plot(t, col=5)
```

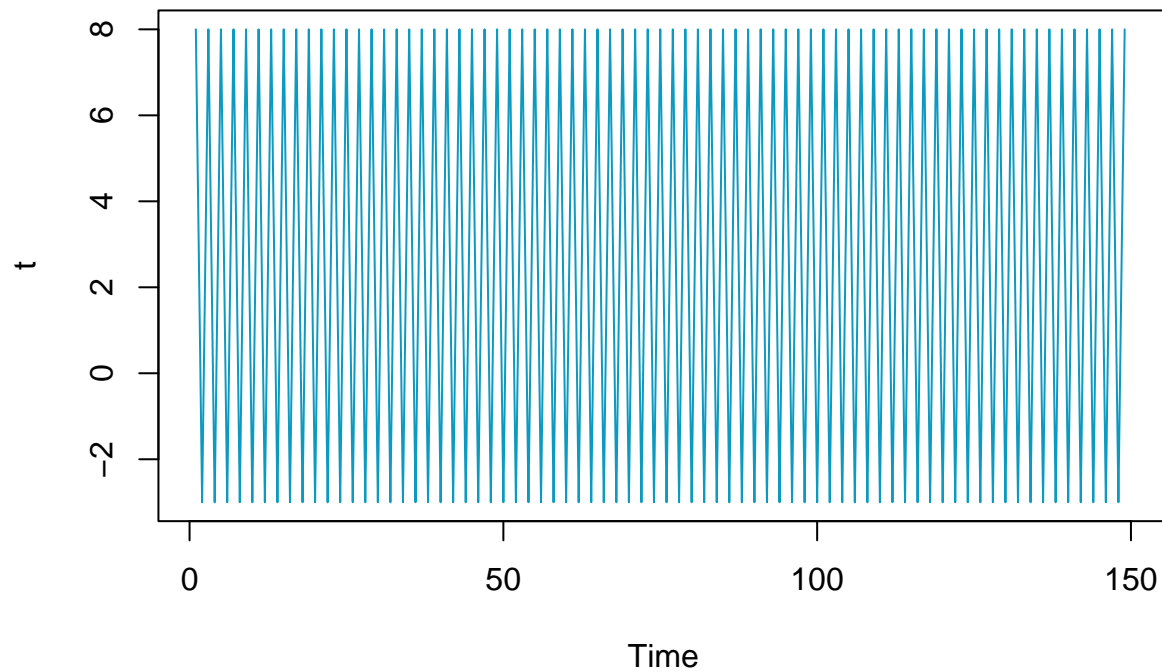


The series follows a linear trend with constant slope.

2) Unit negative ($a_1 = -1$): $Y_t = -Y_{t-1} + a_0$

$$Y_t = -Y_{t-1} + 5, Y_0 = 8$$

```
e <- 150
t <- rep(NA, e)
b <- 2
t[1]=8
while (b<e)
{t[b] <- -t[b-1] + 5
b <- b + 1}
ts.plot(t, col=5)
```

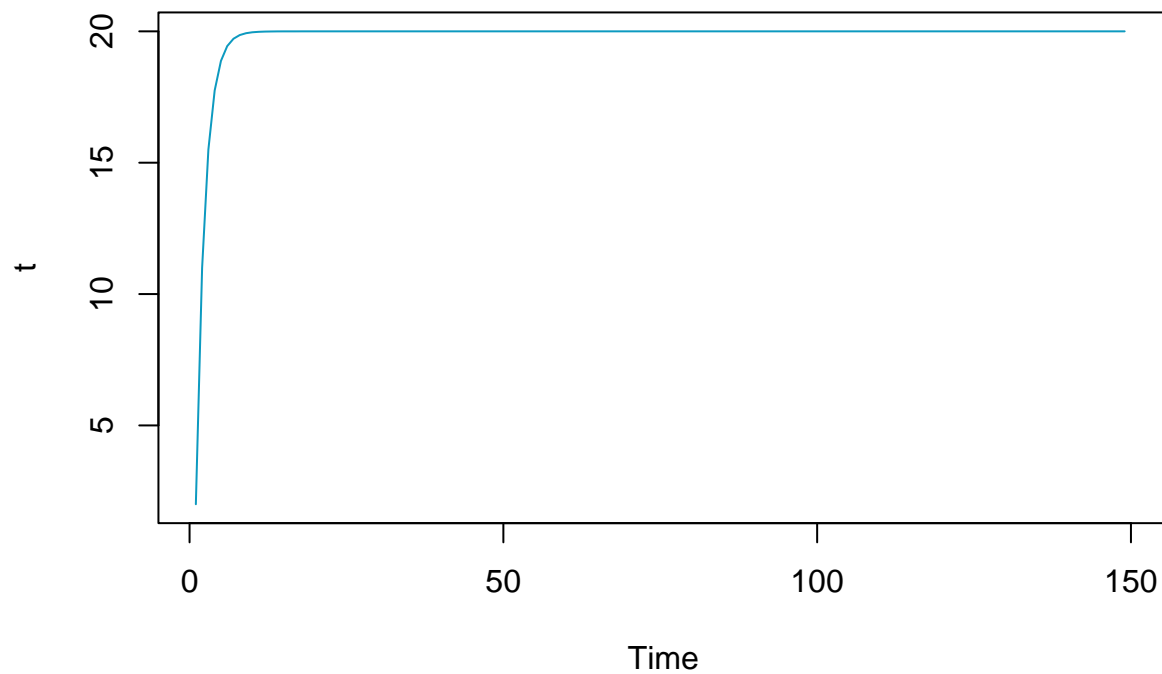


With $a_1 = -1$, the process oscillates between two values. The amplitude remains constant over time.

3) Stable positive ($0 < a_1 < 1$): $Y_t = a_1 \cdot Y_{t-1} + a_0$

$$Y_t = 0.5 \cdot Y_{t-1} + 10, Y_0 = 2$$

```
e <- 150
t <- rep(NA, e)
b <- 2
t[1]=2
while (b<e)
{t[b] <- 0.5*t[b-1] + 10
b <- b + 1}
ts.plot(t, col=5)
```



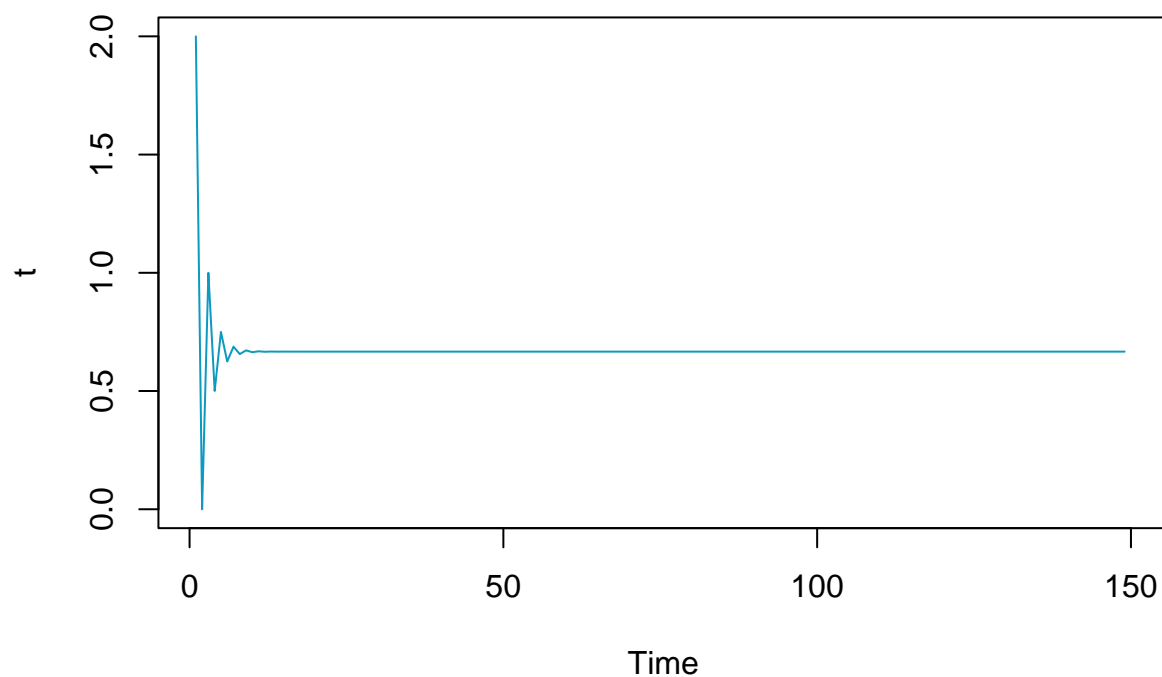
With $0 < a_1 < 1$, the process is stationary and mean-reverting.

4) Stable negative ($-1 < a_1 < 0$):

$$Y_t = a_1 \cdot Y_{t-1} + a_0$$

$$Y_t = -0.5 \cdot Y_{t-1} + 1, Y_0 = 2$$

```
e <- 150
t <- rep(NA, e)
b <- 2
t[1]=2
while (b<e)
{t[b] <- -0.5*t[b-1] + 1
b <- b + 1}
ts.plot(t, col=5)
```

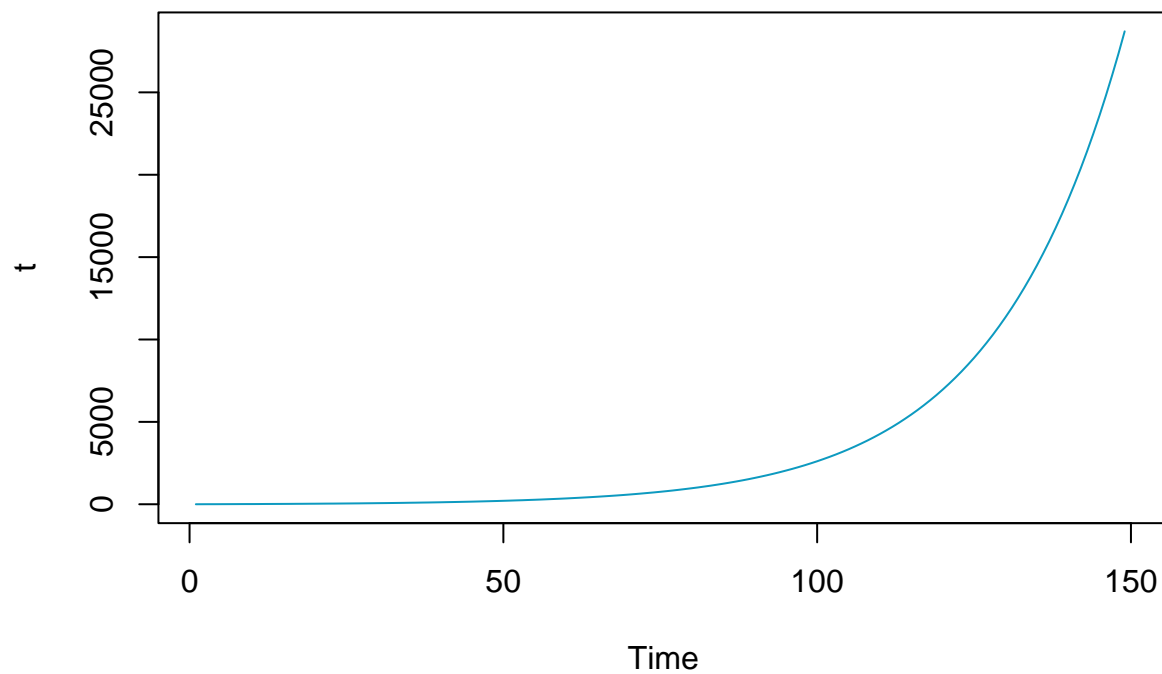


With negative but stationary coefficient ($-1 < a_1 < 0$), the process fluctuates around its mean while converging.

Unstable positive ($a_1 > 1$): $Y_t = a_1 \cdot Y_{t-1} + a_0$

$$Y_t = 1.05 \cdot Y_{t-1} + 1, Y_0 = 1$$

```
e <- 150
t <- rep(NA, e)
b <- 2
t[1]=1
while (b<e)
{t[b] <- 1.05*t[b-1] + 1
b <- b + 1}
ts.plot(t, col=5)
```

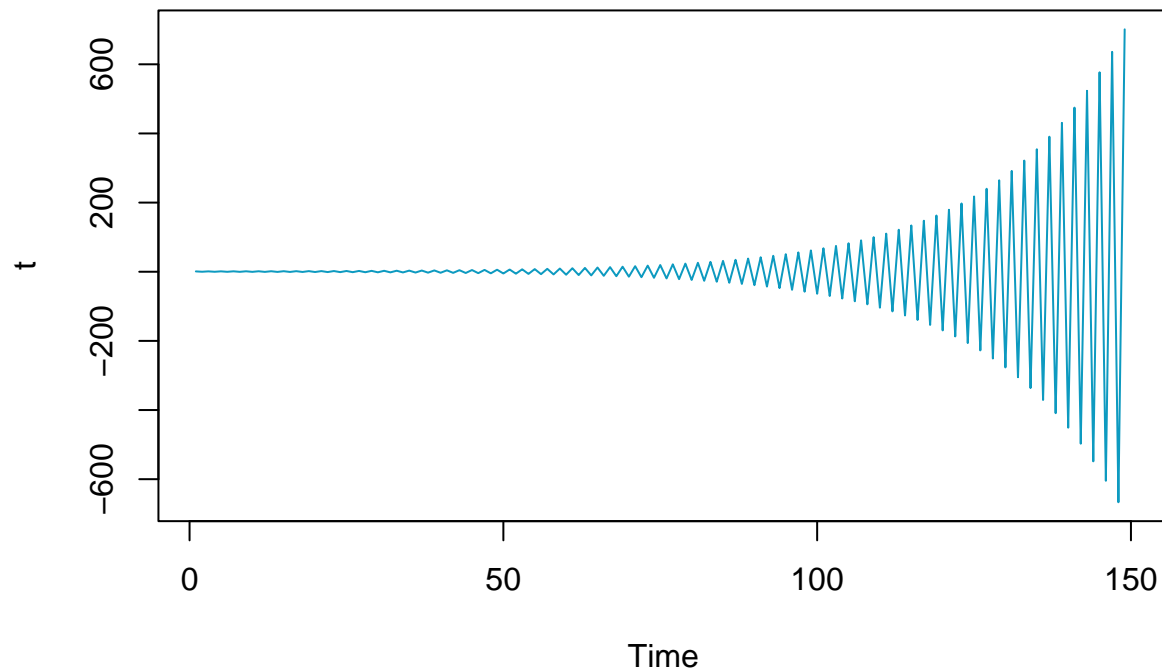


With $a_1 > 1$, the process is explosive and grows exponentially.

Unstable negative ($a_1 < -1$): $Y_t = a_1 \cdot Y_{t-1} + a_0$

$$Y_t = -1.05 \cdot Y_{t-1} + 1, Y_0 = 1$$

```
e <- 150
t <- rep(NA, e)
b <- 2
t[1]=1
while (b<e)
{t[b] <- -1.05*t[b-1] + 1
b <- b + 1}
ts.plot(t, col=5)
```

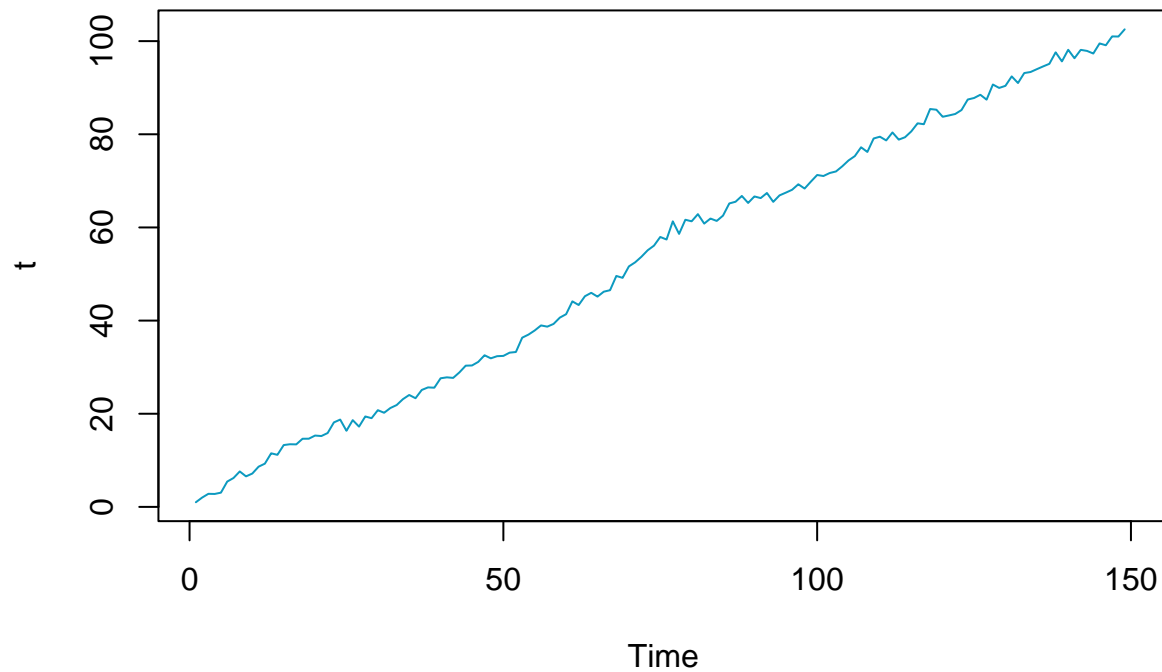


With $a < -1$, the process exhibits explosive oscillation with alternating signs and increasing amplitude. The absolute value grows exponentially.

Expand to AR(2) with stochastic component: $Y_t = a_1 \cdot Y_{t-1} + a_2 \cdot Y_{t-2} + a_0$

Unit positive ($a_1 + a_2 = 1$)

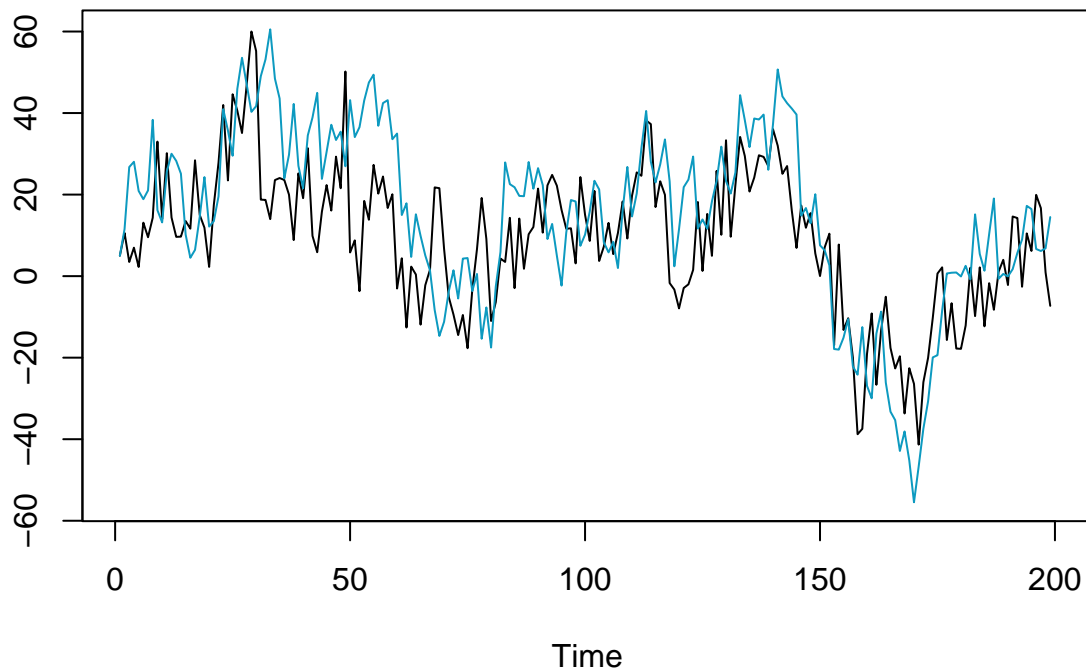
```
b<-3
e<-150
t<-rep(NA, e)
t[1]=1
t[2]=2
while(b<e)
{ t[b] <-0.5*t[b-1]+ 0.5*t[b-2]+1+rnorm(1,mean=0, sd=1)
b<-b+1 }
ts.plot(t, col=5)
```



Bivariate time series

Unit positive (both a_1 and b_1 are greater than 1)

```
b<-2
e<-200
x<-rep(NA, e)
y<-rep(NA, e)
x[1]=5
y[1]=5
while(b<e)
{ x[b] <-0.6*x[b-1]+ 0.2*y[b-1]+1+rnorm(1,mean=0, sd=10)
  y[b] <- 0.8*y[b-1]+0.2*x[b-1]+2+rnorm(1,mean=0, sd=10)
  b<-b+1 }
df=data.frame(x,y)
ts.plot(df, col=c(1,5))
```

2. Task

Find on the Rosstat website monthly data on average wage and inflation in the regions of Russian Federation, select one region, take the longest available series for it, calculate the following series and draw the graphs:

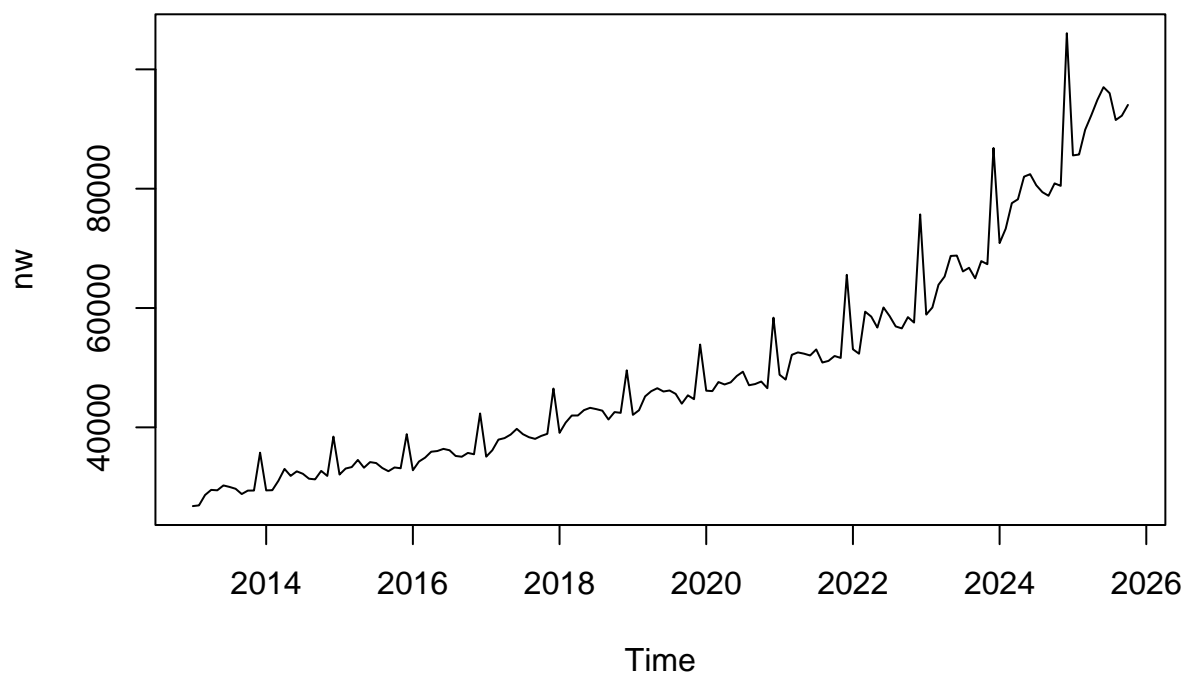
- average nominal wage ,
 - inflation rate,
 - price level (end of the first month= 1),
 - real wages in prices of the beginning of the period,
 - real wages in end-of-period prices.
 - logarithm of real wages,
 - seasonally adjusted logarithm of real wages,
 - growth rate of real wages (in logarithms),
 - seasonally adjusted growth rate of real wages in logarithms.
- You may first seasonally adjust the series and then take first differences or vice versa. Compare the result.

```
library(readxl)
data <- read_excel("D:\\wages.xlsx", skip=1)
nw <- data$'nominal wage'
rw1 <- data$'real wages in 2013 prices'
rw2 <- data$'real wages in 2025 prices'
cpi <- data$cpi
pl<-data$`price level`

nw=ts(nw,start = c(2013, 1), frequency = 12)
rw1=ts(rw1,start = c(2013, 1), frequency = 12)
rw2=ts(rw2,start = c(2013, 1), frequency = 12)
cpi=ts(cpi,start = c(2013, 1), frequency = 12)
pl=ts(pl,start = c(2013, 1), frequency = 12)
```

Average nominal wage in Leningrad region

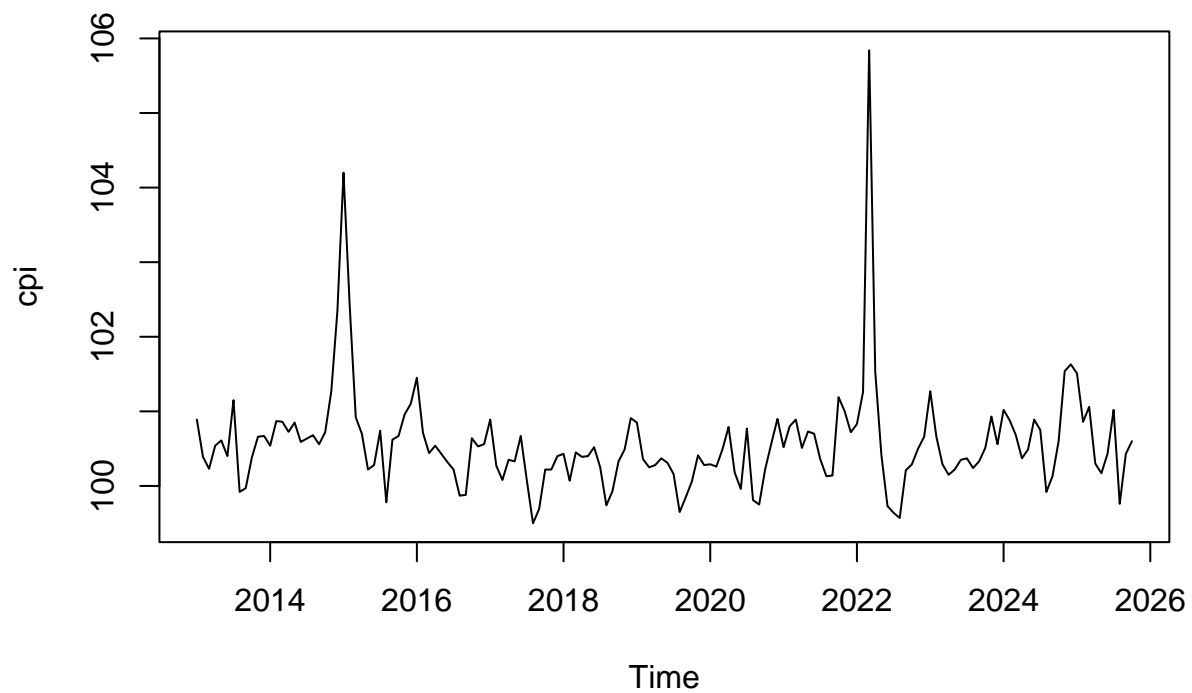
```
ts.plot(nw)
```



Nominal wages show clear upward trend with seasonal peaks in December (year-end bonuses).

Inflation rate in Leningrad region

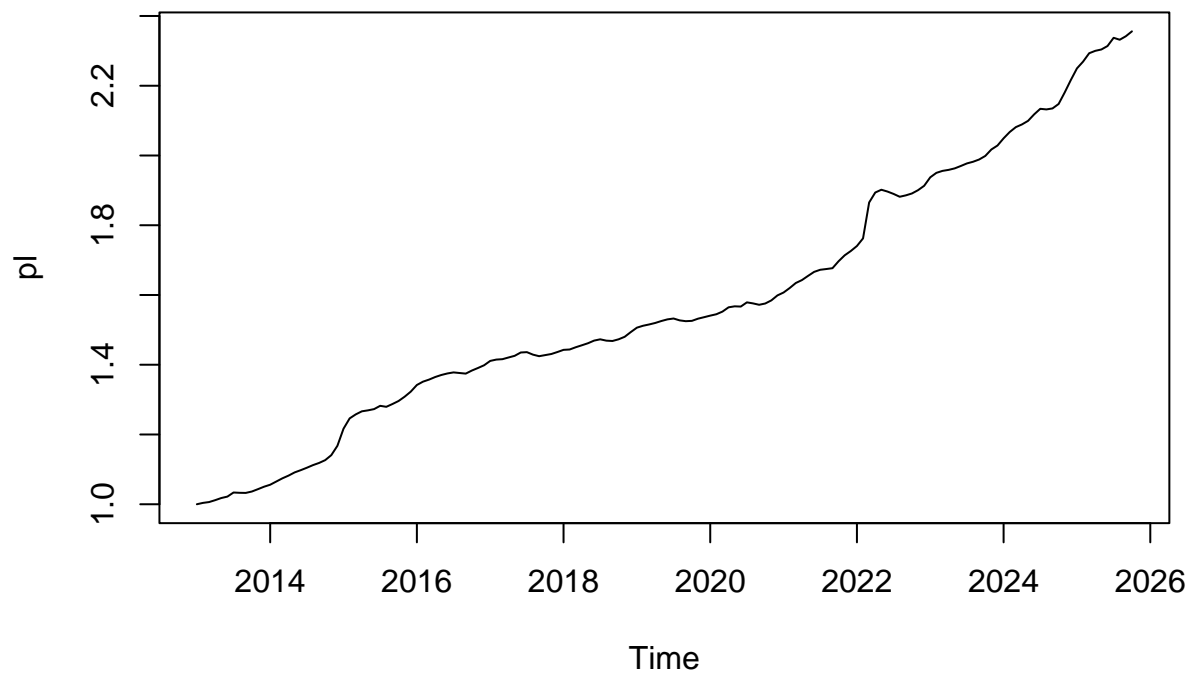
```
ts.plot(cpi)
```



Notable inflation observed in early 2015, 2022, and 2024. Most periods show relatively stable prices.

Price level

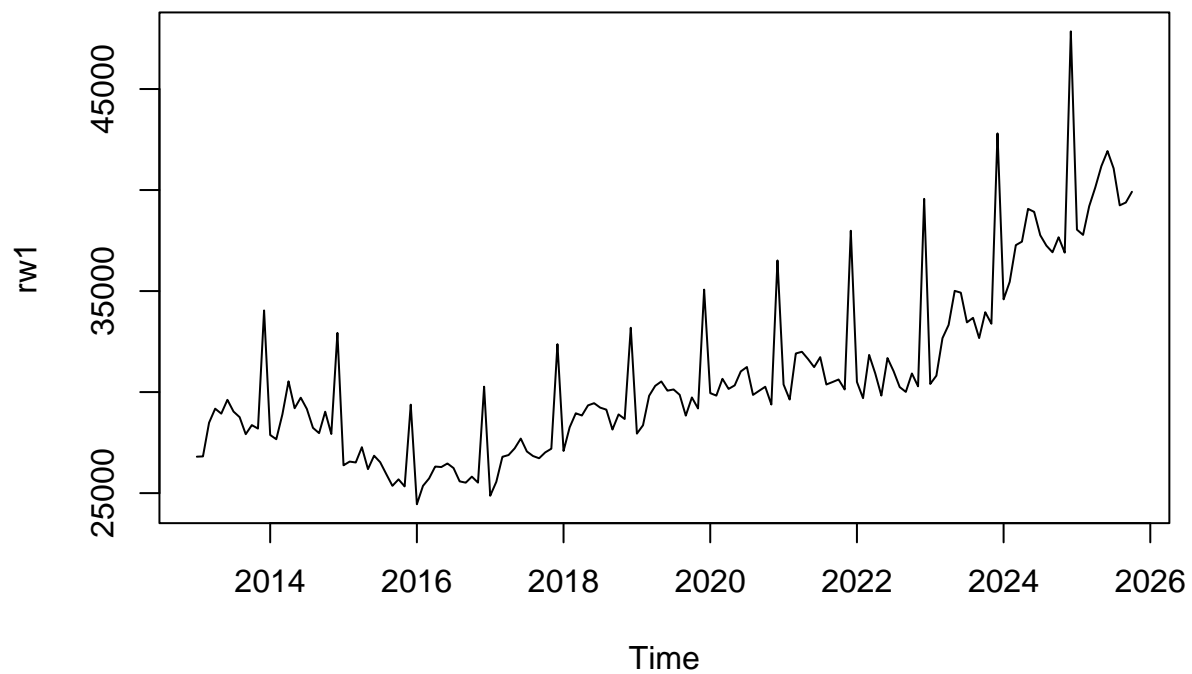
```
ts.plot(pl)
```



Cumulative price level shows steady increase over time, approximately doubling from 2013 to 2025.

Real wages in prices of the beginning of the 2013

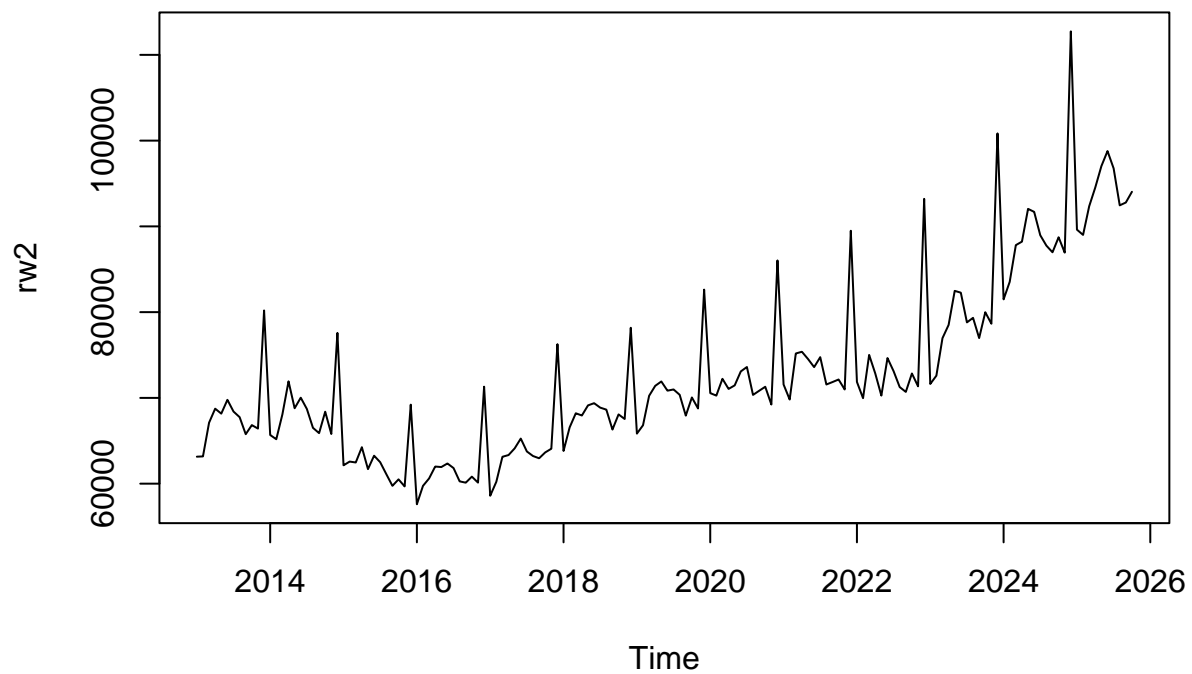
```
ts.plot(rw1)
```



Real wages in constant 2013 prices show moderate growth with fluctuations. The series removes inflation effects, revealing actual purchasing power changes.

Real wages in prices of the end of 2025

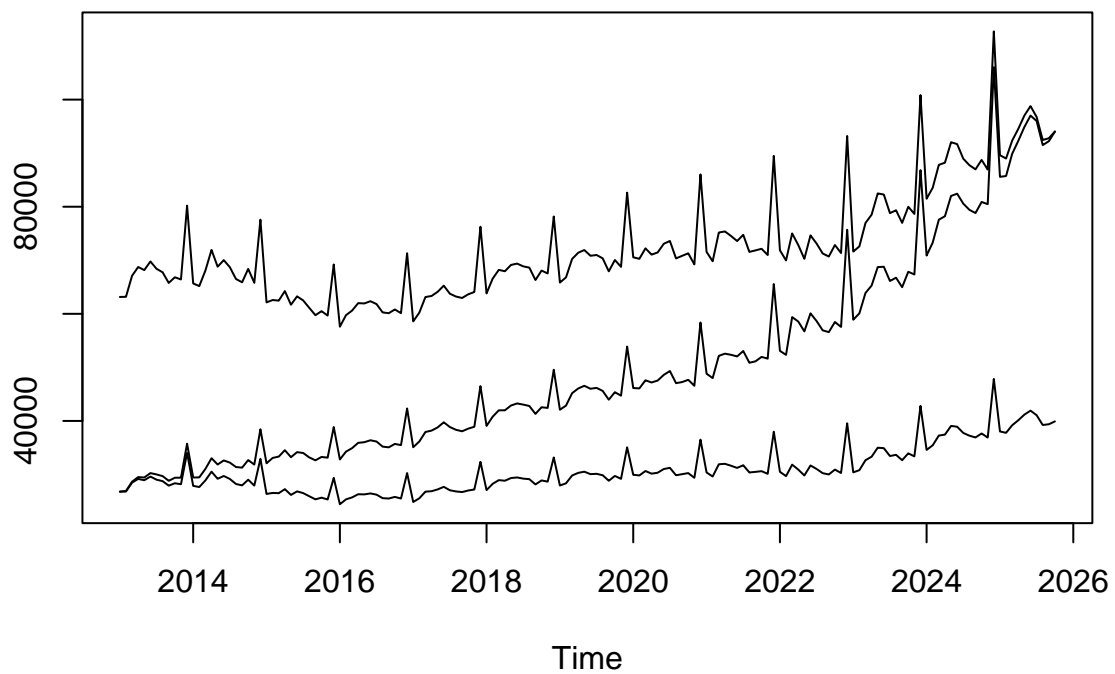
```
ts.plot(rw2)
```



Real wages expressed in end-period prices show similar pattern to 2013-based series but with different scale.

Real wages in both prices of the beginning of the 2013 and of the end of 2025

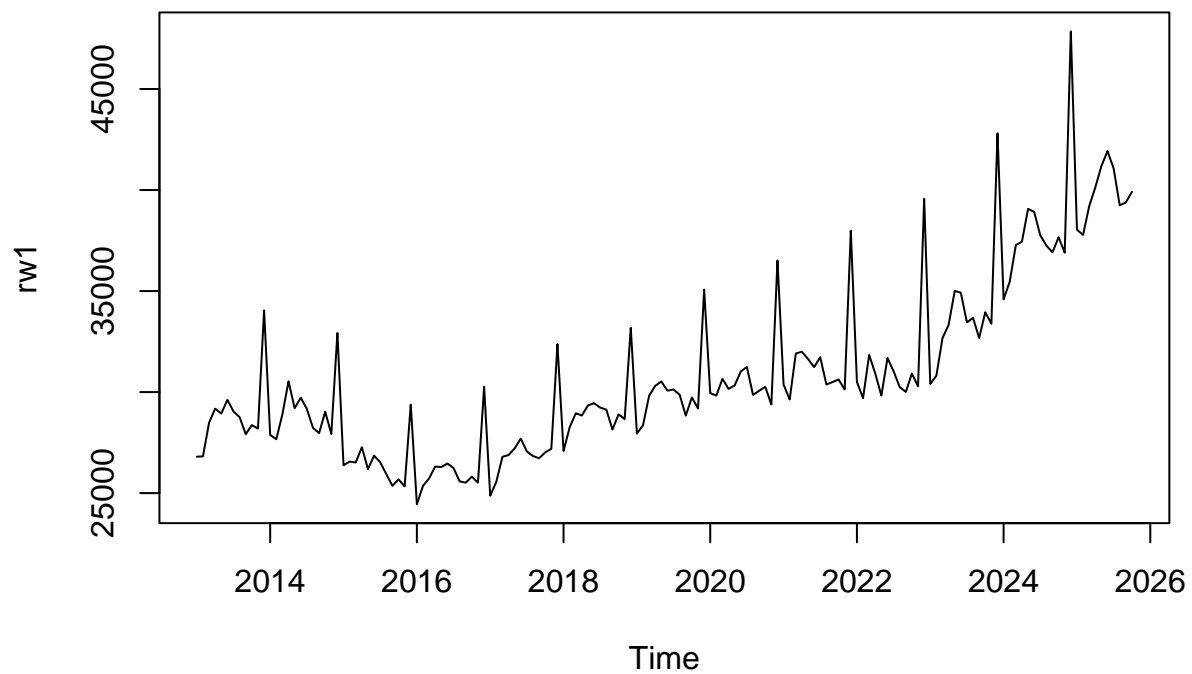
```
df<-data.frame(nw,rw1,rw2)
df=ts(df,start = c(2013, 1), frequency = 12)
ts.plot(df)
```



Logarithm of the average real wage

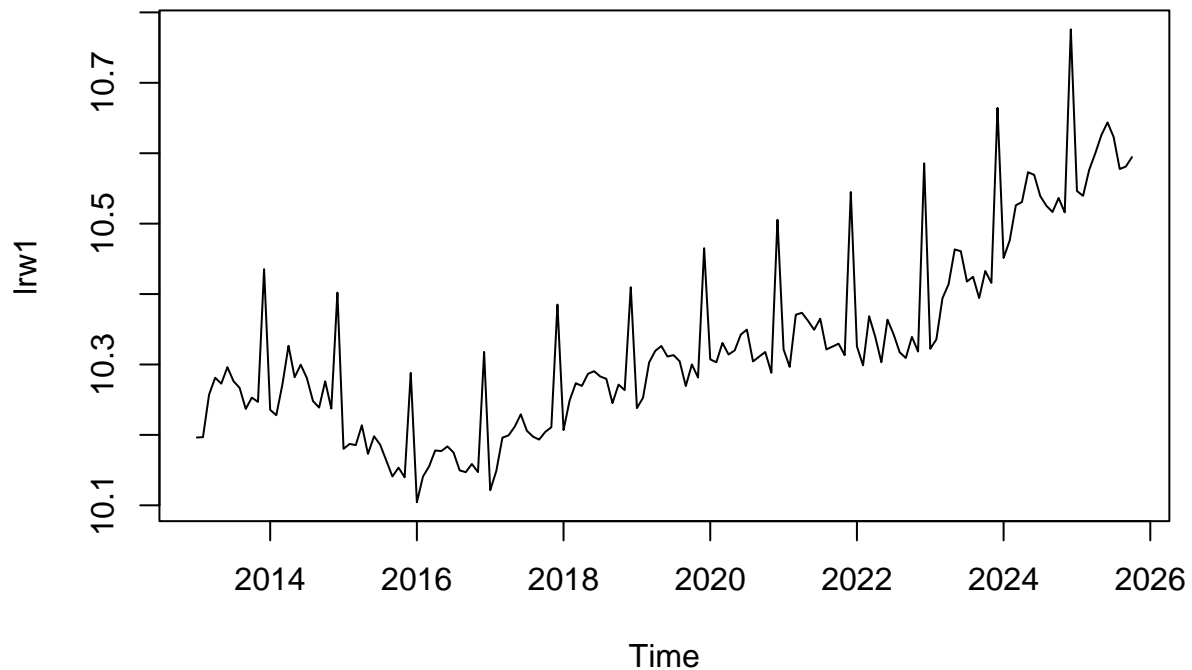
Average real wage

```
ts.plot(rw1)
```



Logarithm of the average real wage

```
lrw1<-log(rw1)  
ts.plot(lrw1)
```

Logarithm transformation linearizes exponential growth and stabilizes variance. The log series shows approximately linear trend with seasonal fluctuations.

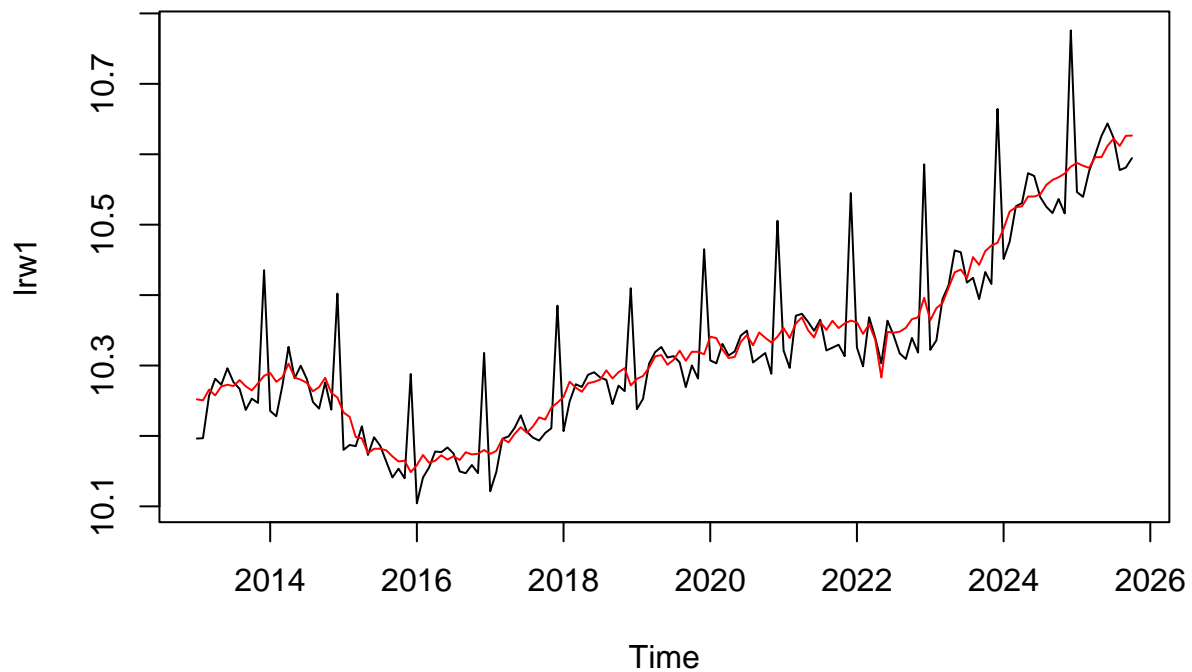
Adjusting for seasonality

```
x=seas(lrw1)
summary(x)
```

```
##
## Call:
## seas(x = lrw1)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## A02022.May    -0.005902   0.001304  -4.527 0.0000059723 ***
## AR-Nonseasonal-01 -0.707734   0.318939  -2.219   0.0265 *
## AR-Nonseasonal-02 -0.275522   0.116175  -2.372   0.0177 *
## MA-Nonseasonal-01 -0.344556   0.331461  -1.040   0.2986
## MA-Seasonal-12    0.427964   0.077118   5.549 0.0000000287 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## SEATS adj. ARIMA: (2 1 1)(0 1 1) Obs.: 154 Transform: log
## AICc: -695.1, BIC: -678.1 QS (no seasonality in final): 0
```

```
## Box-Ljung (no autocorr.): 41.58 * Shapiro (normality): 0.9926
## Messages generated by X-13:
## Warnings:
## - At least one visually significant trading day peak has been
##   found in one or more of the estimated spectra.
```

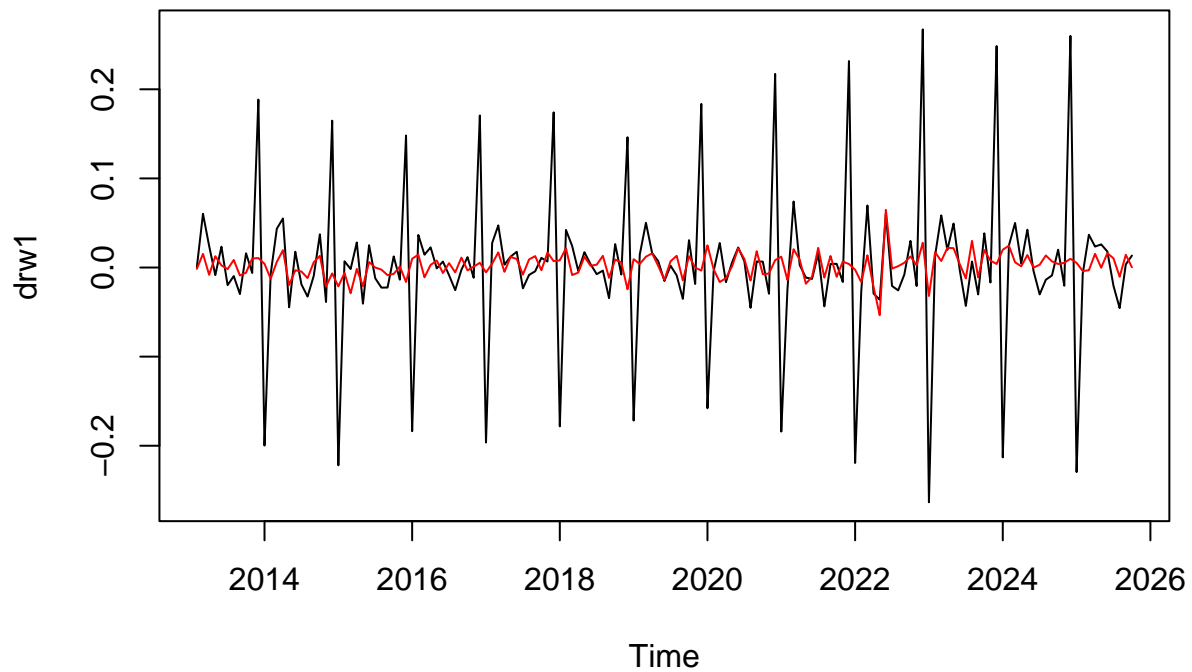
```
lrw1_sa=ts(c(final(x)),start = c(2013, 1), frequency = 12)
ts.plot(lrw1)
lines(lrw1_sa, col="red")
```



Seasonal adjustment removes regular year patterns (December bonuses, summer declines).

Initial and seasonally adjusted growth rate of real wage (in logarithms)

```
drw1<-diff(lrw1)
drw1_sa<-diff(lrw1_sa)
ts.plot(drw1)
lines(drw1_sa, col="red")
```

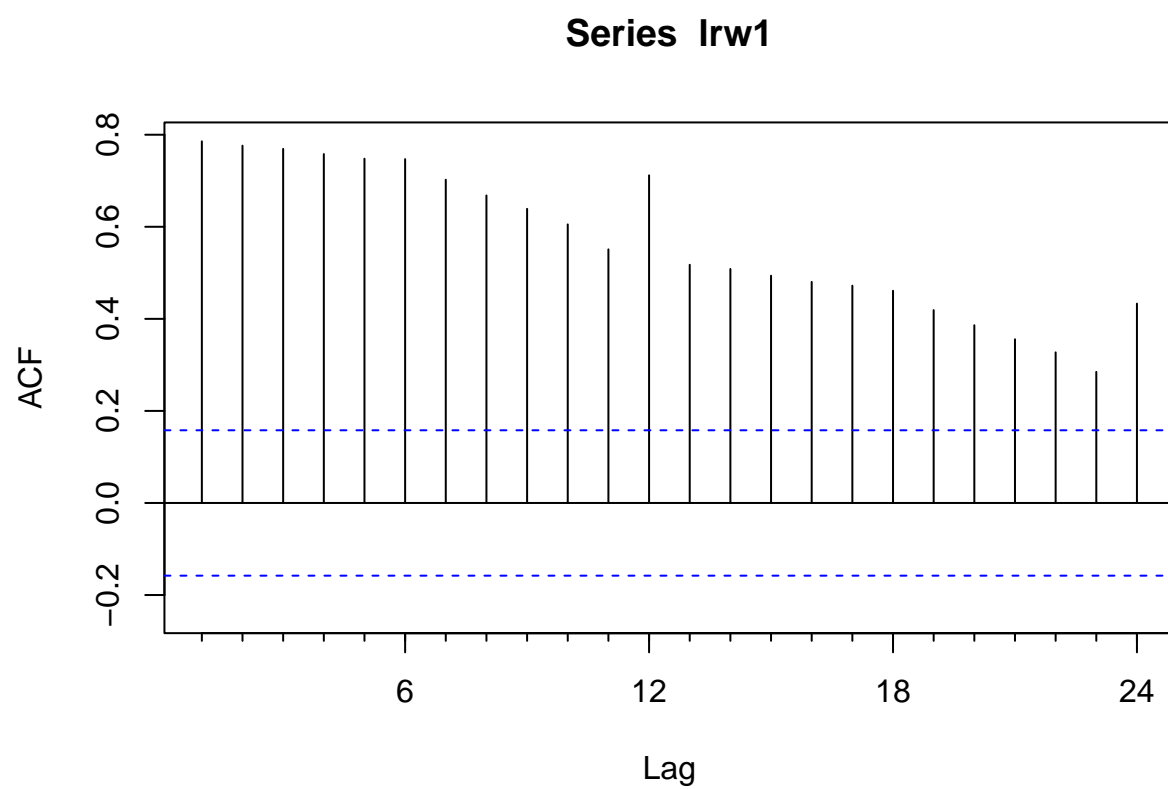


Growth rates show stationarity and fluctuate around zero. Seasonally adjusted series has reduced volatility and clearer cyclical patterns.

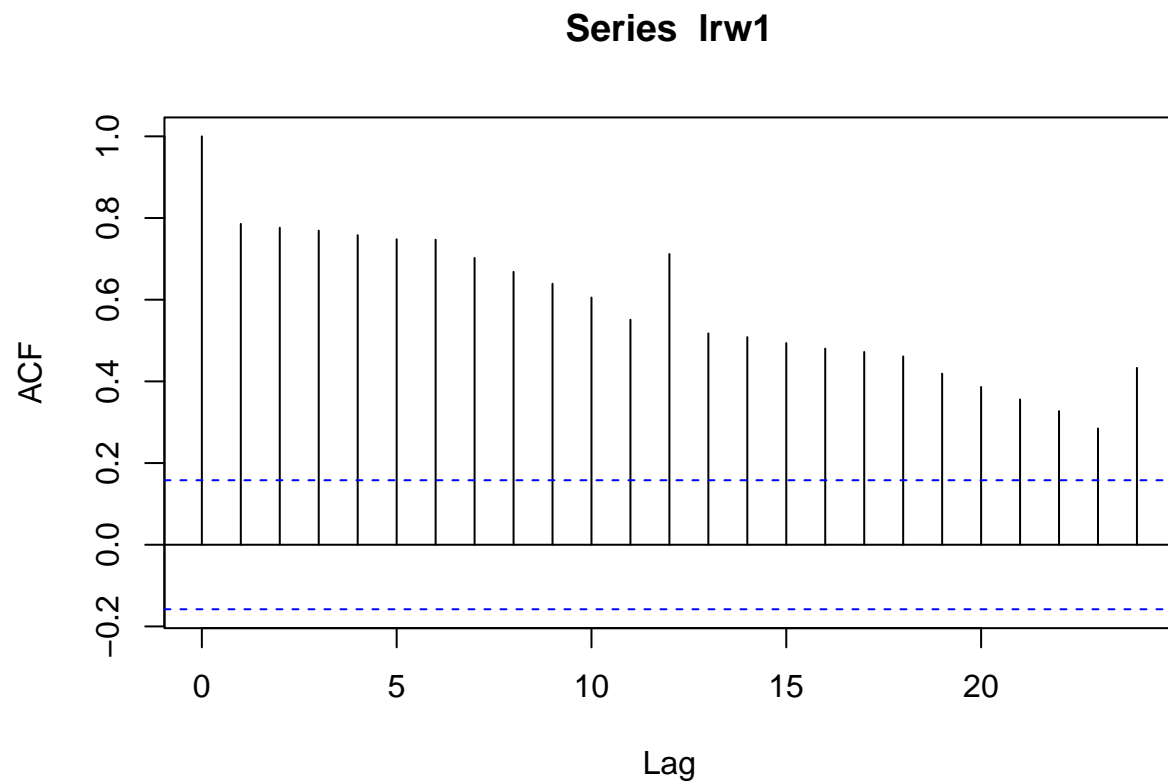
4. Task

Autocorrelation of real wages in logarithms

```
a<-Acf(lrw1)
```



```
plot(a)
```

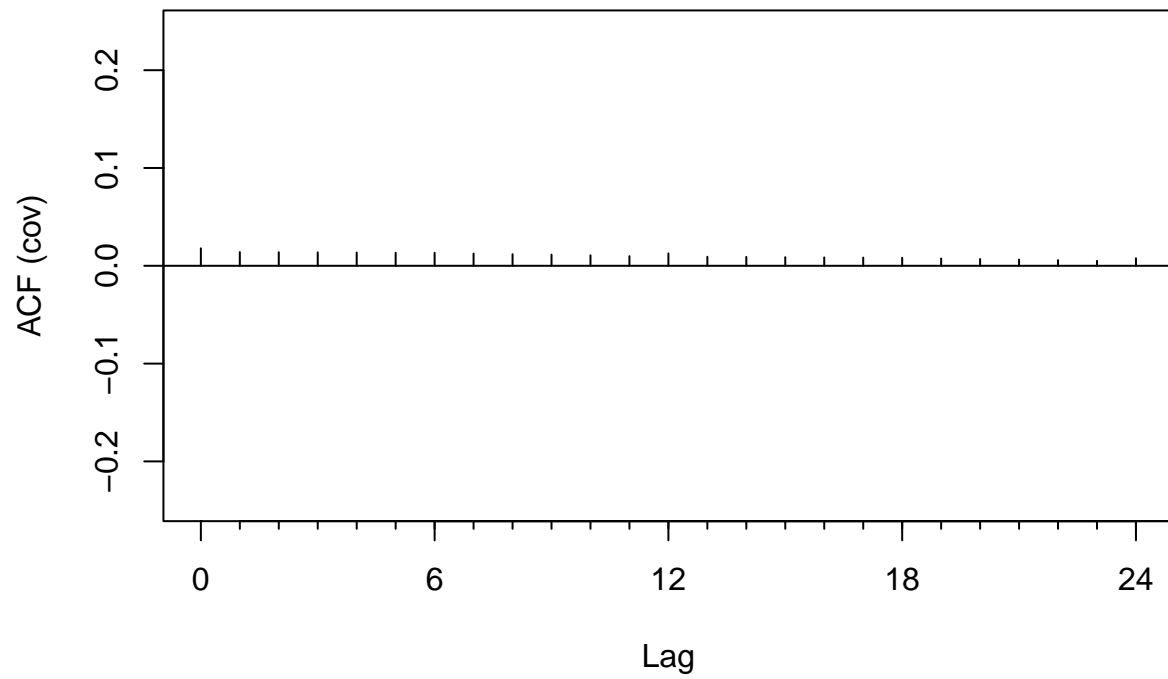


Real wages in logarithms have sample autocorrelation coefficients that are all distinguish from zero.

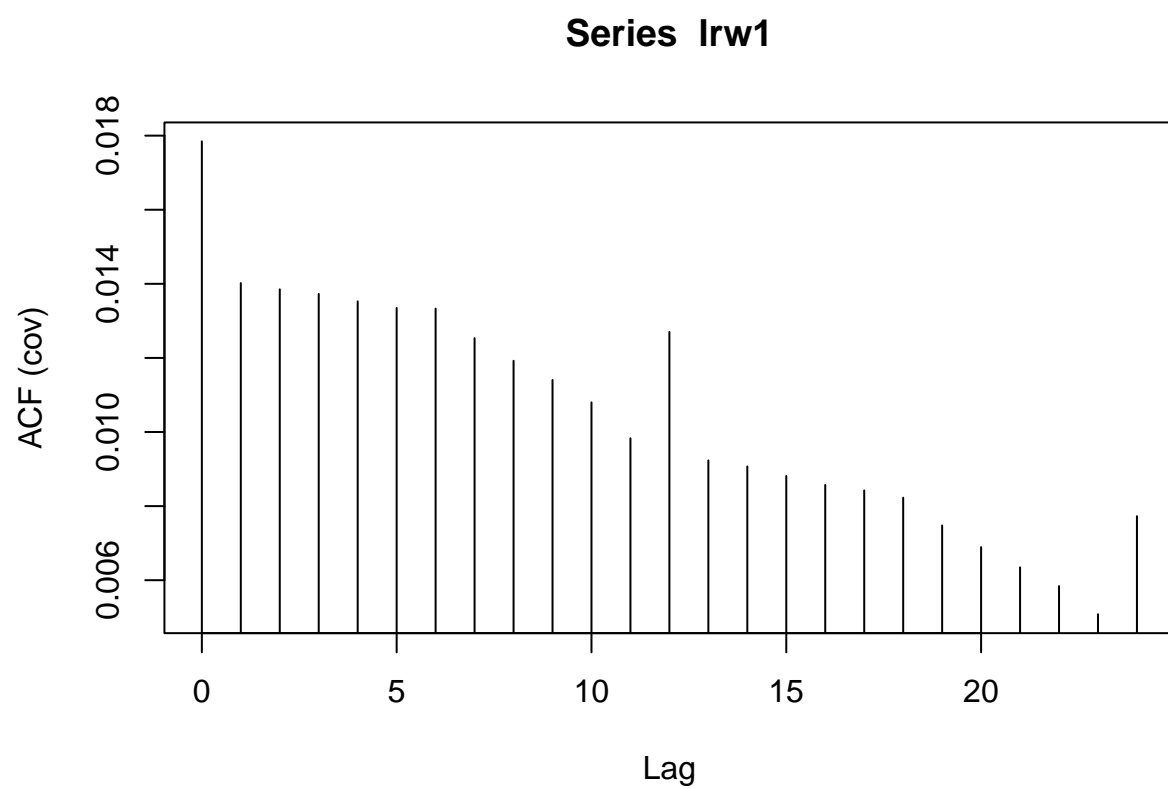
Autocovariance of real wages in logarithms

```
a<-Acf(lrw1, type='covariance')
```

Series lrw1



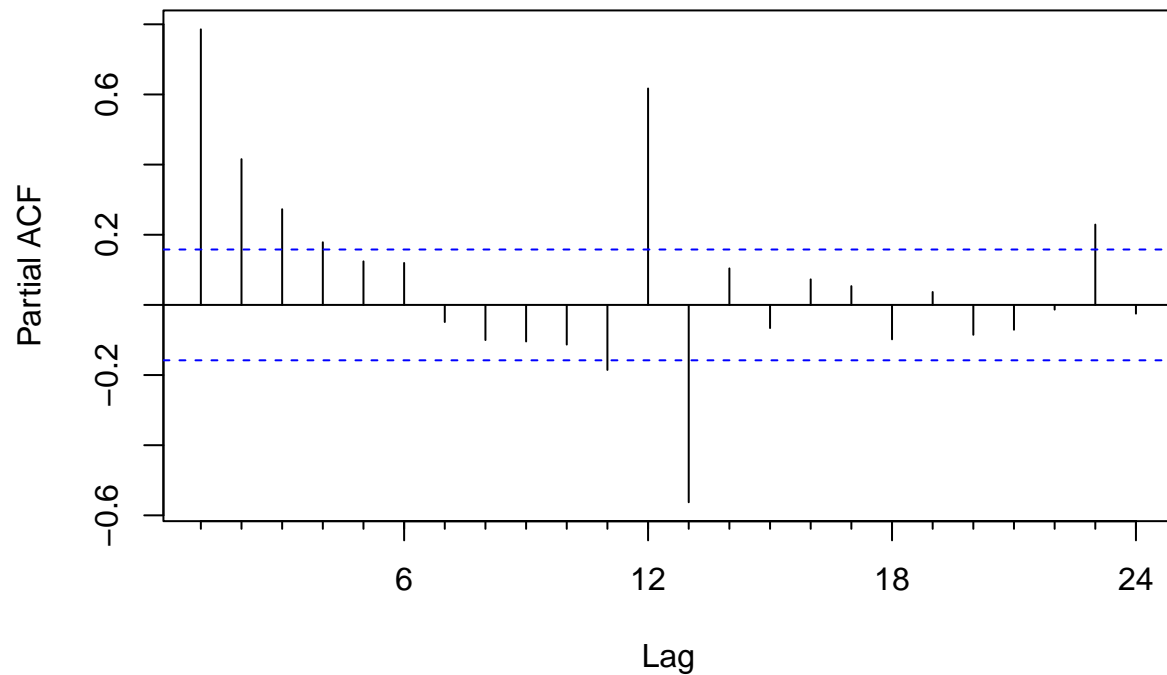
```
plot(a)
```



Partial autocorrelation of real wages in logarithms

```
Pacf(lrw1)
```

Series lrw1



Autocorrelation tests of real wages in logarithms

#Box-Pierce test

```
Box.test(lrw1, lag = 10, type = "Box-Pierce")
```

##

Box-Pierce test

##

data: lrw1

X-squared = 803.53, df = 10, p-value < 0.00000000000000022

#Ljung-Box test

```
Box.test(lrw1, lag = 10, type = "Ljung-Box")
```

##

Box-Ljung test

##

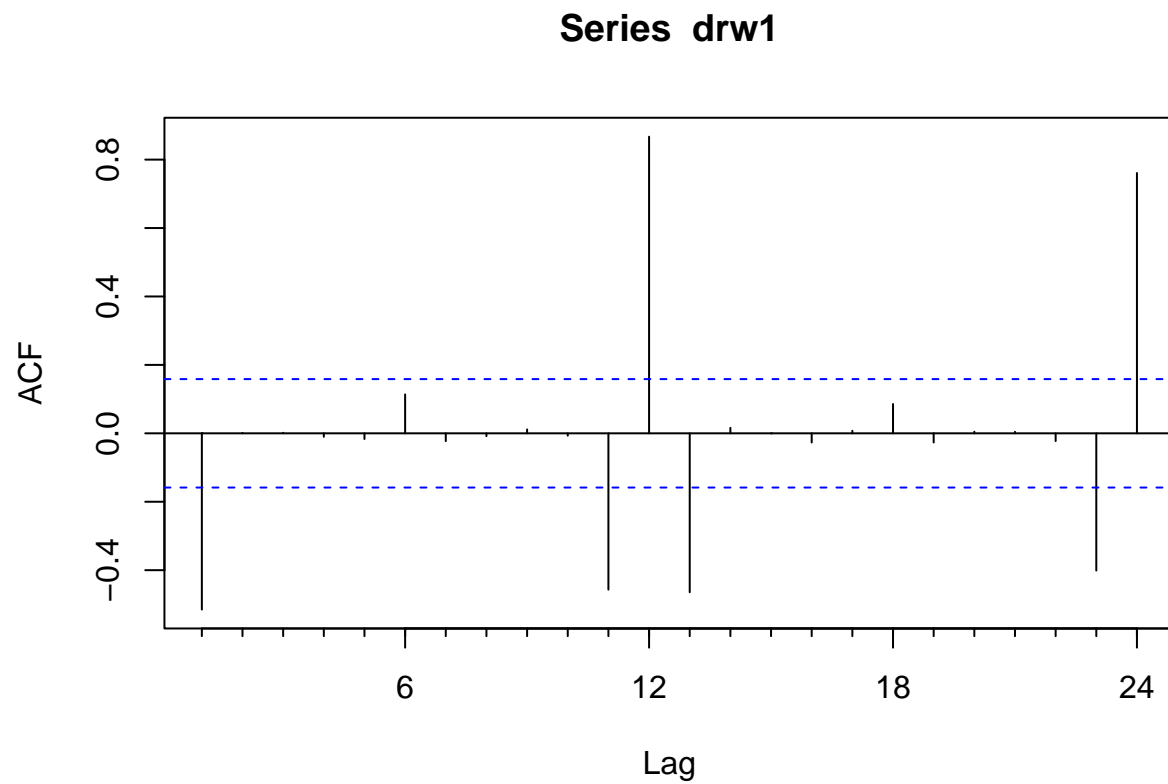
data: lrw1

X-squared = 841.91, df = 10, p-value < 0.00000000000000022

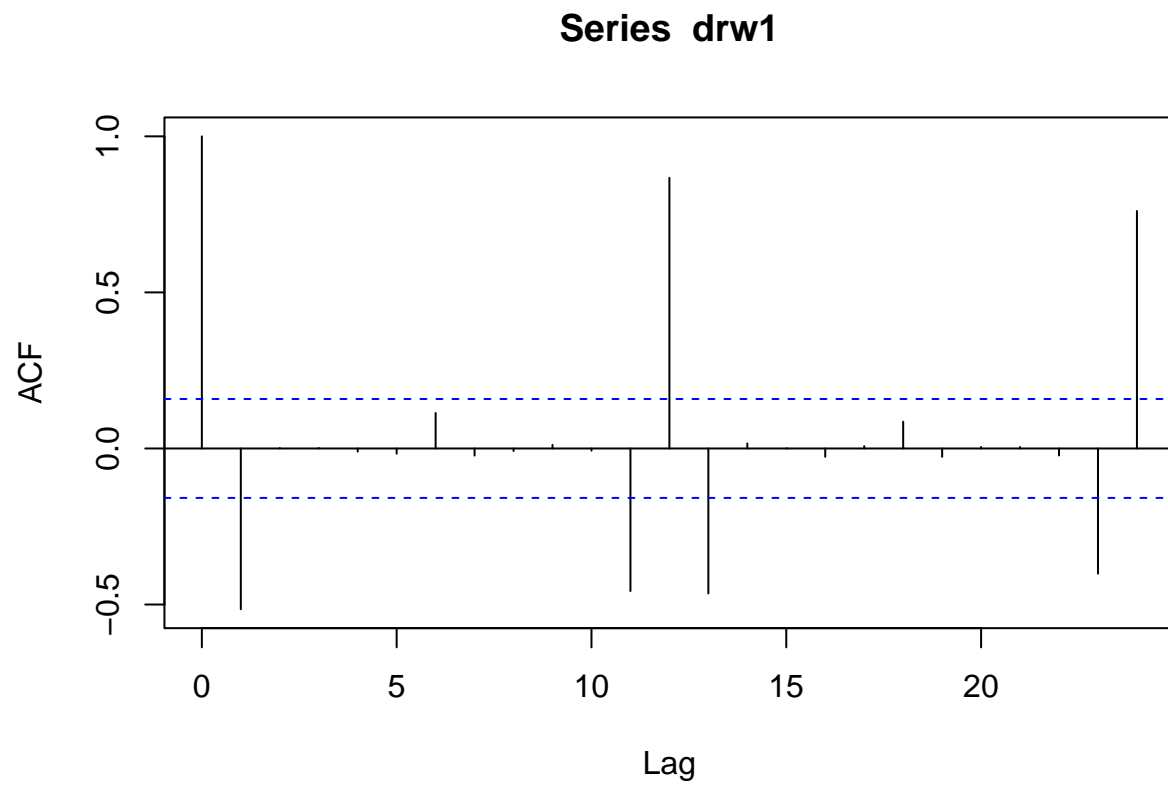
Both tests reject null hypothesis of no autocorrelation ($p < 0.001$), confirming strong serial dependence in log real wages.

Autocorrelation of growth rate of real wages in logarithms

```
a<-Acf(drw1)
```



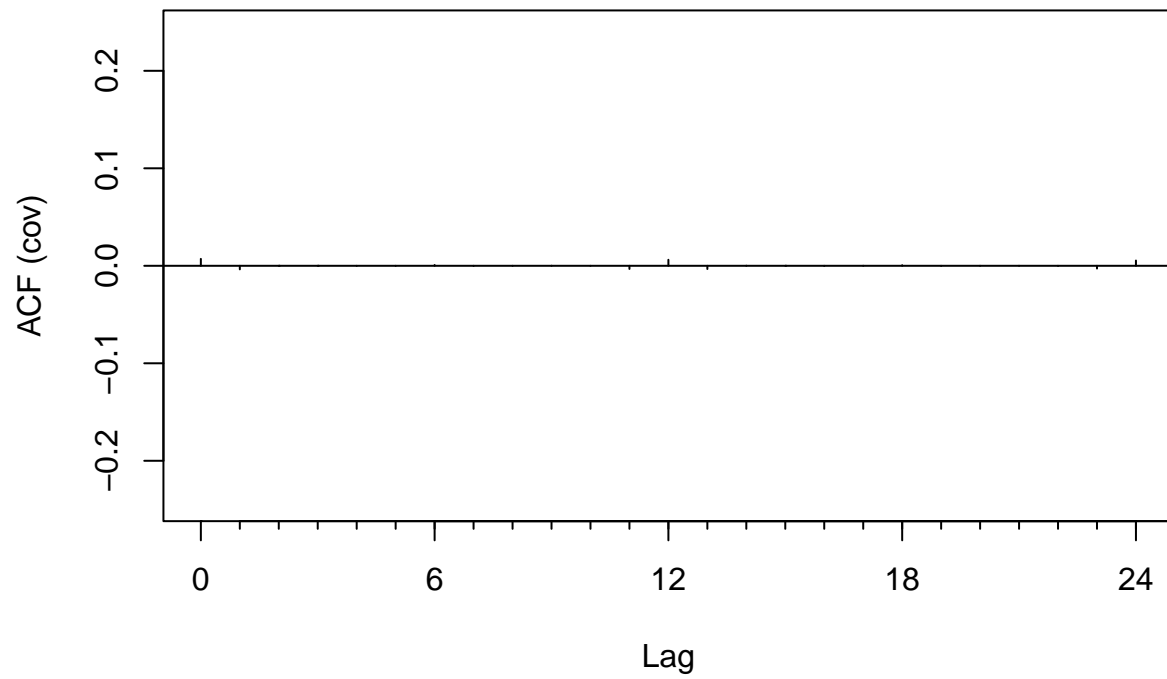
```
plot(a)
```



Autocovariance of growth rate of real wages in logarithms

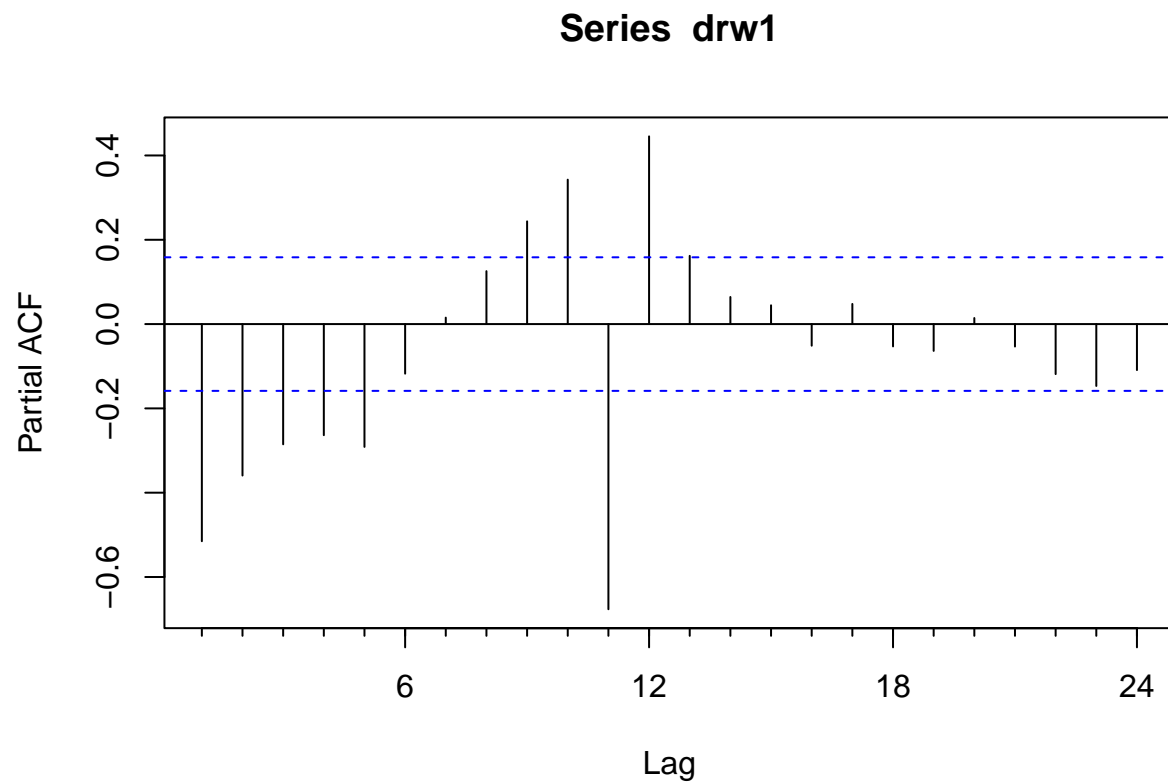
```
Acf(drw1, type='covariance')
```

Series drw1



Partial autocorrelation of growth rate of real wages in logarithms

```
Pacf(drw1)
```



Autocorrelation tests of growth rate of real wages in logarithms

```
#Box-Pierce test
Box.test(drw1, lag = 10, type = "Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: drw1
## X-squared = 42.752, df = 10, p-value = 0.000005507
```

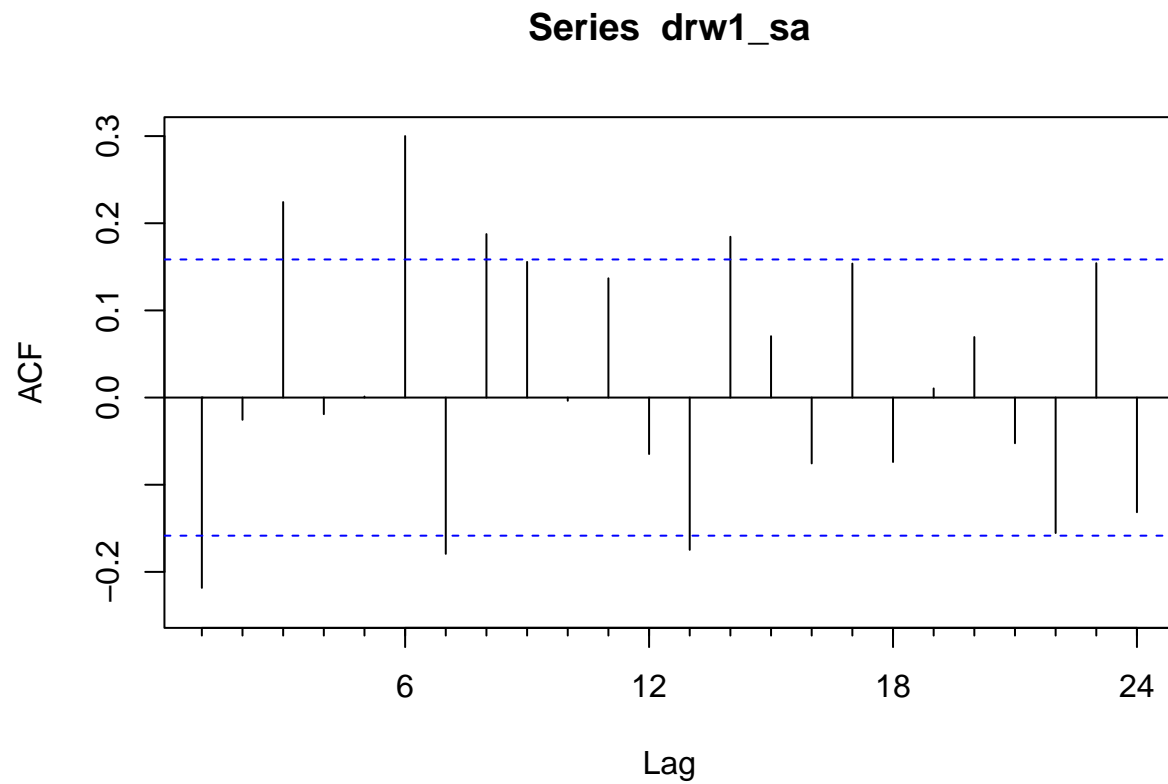
```
#Ljung-Box test
Box.test(drw1, lag = 10, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: drw1
## X-squared = 43.671, df = 10, p-value = 0.00000377
```

P-value is less than 0.05, then H_0 about the presence of autocorrelation can not be denied.

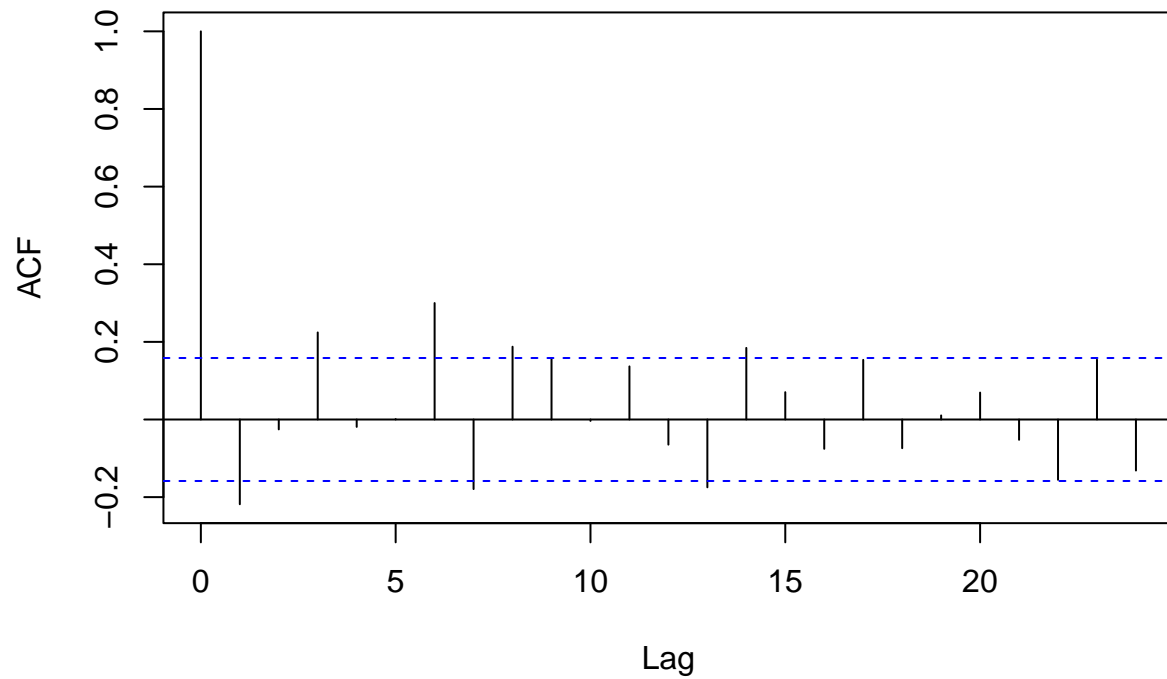
Autocorrelation of the seasonally adjusted growth rate of the real wages in logarithms

```
a<-Acf(drw1_sa)
```



```
plot(a)
```

Series drw1_sa

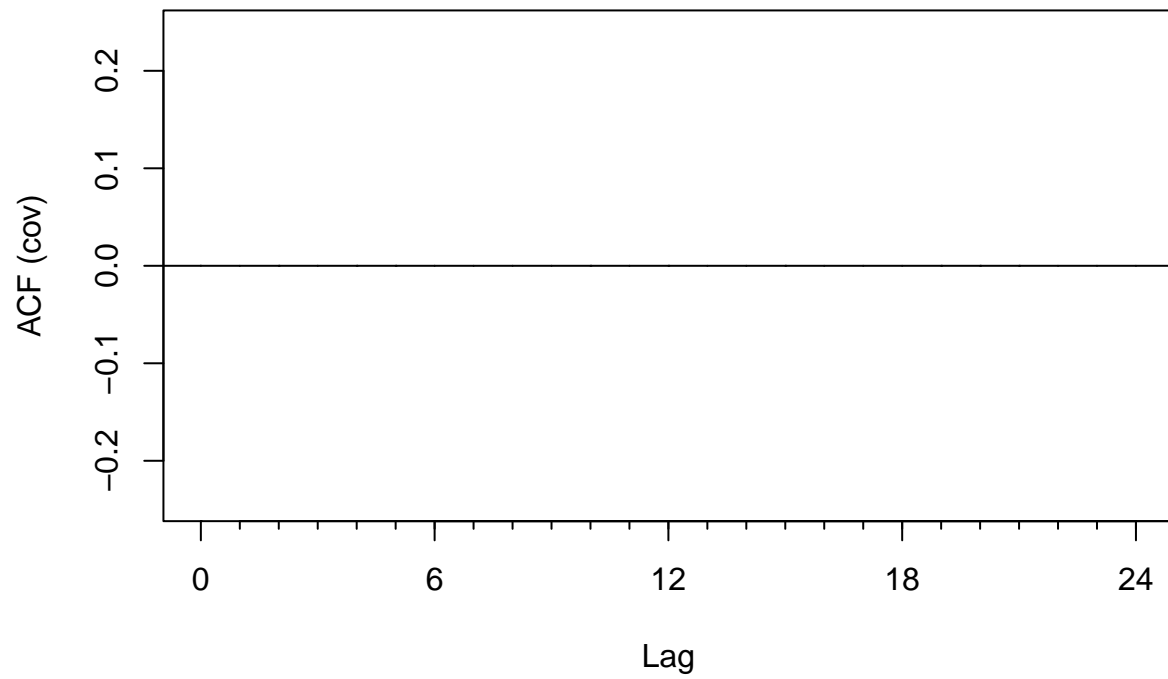


Seasonally adjusted growth rates of the real wages have sample autocorrelation coefficients that **mainly are indistinguishable from zero** => probably, there is no autocorrelation and the current real wage does not depend on the previous values

Autocovariance of the real wages in logarithms

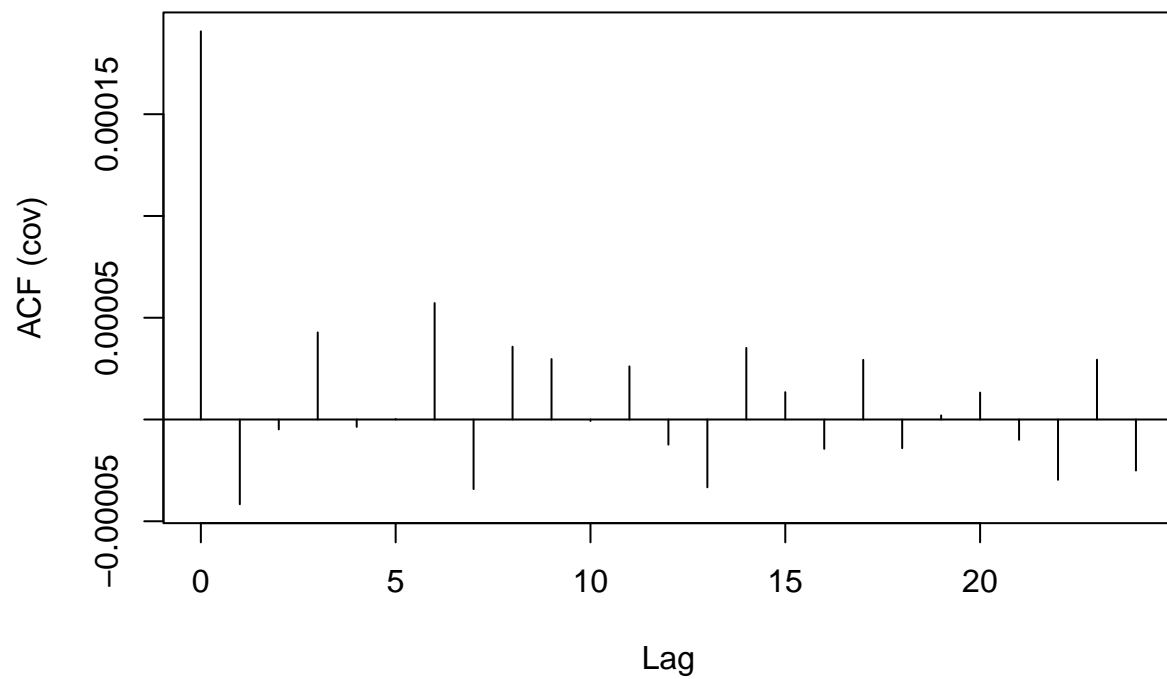
```
a<-Acf(drw1_sa, type='covariance')
```

Series drw1_sa



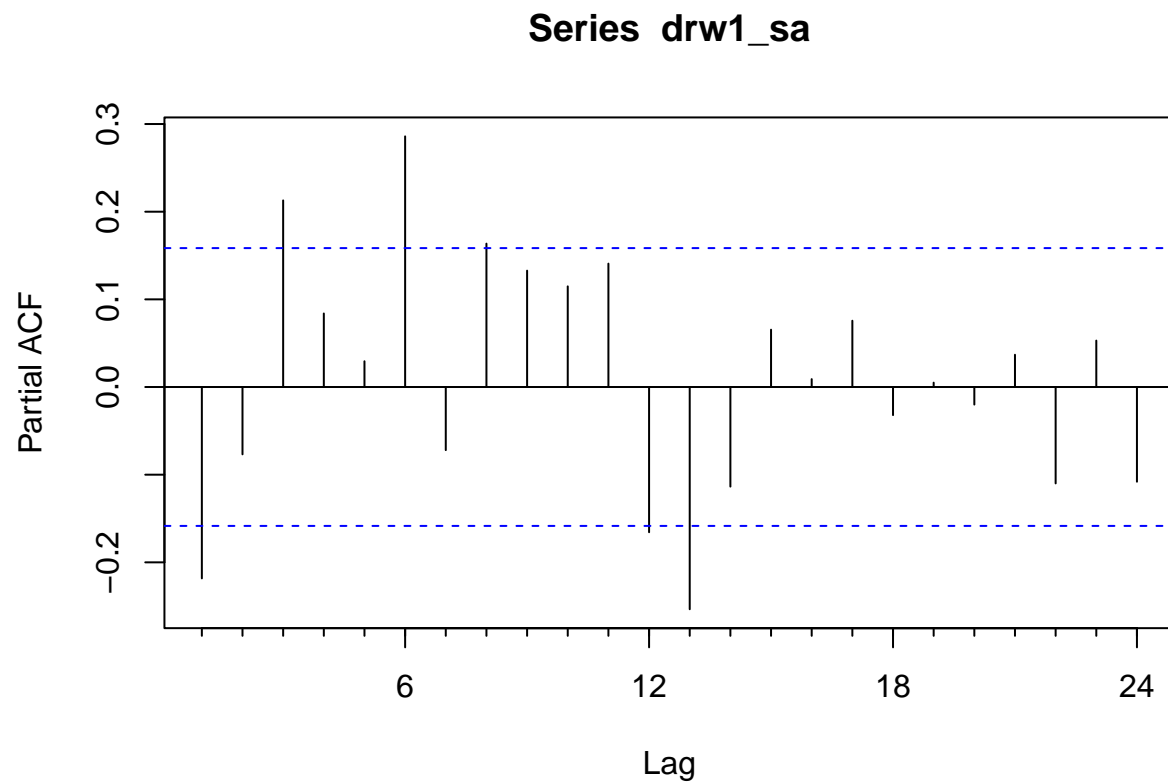
```
plot(a)
```

Series drw1_sa



Seasonally adjusted growth rates of the real wages have sample autocovariance that are **rather small**. This supports the results with autocorrelation.

```
Pacf(drw1_sa)
```

Autocorrelation tests of seasonally adjusted growth rate of real wages in logarithms

```
#Box-Pierce test  
Box.test(drw1_sa, lag = 10, type = "Box-Pierce")
```

```
##  
## Box-Pierce test  
##  
## data: drw1_sa  
## X-squared = 42.924, df = 10, p-value = 0.000005131
```

```
#Ljung-Box test  
Box.test(drw1_sa, lag = 10, type = "Ljung-Box")
```

```
##  
## Box-Ljung test  
##  
## data: drw1_sa  
## X-squared = 45.037, df = 10, p-value = 0.000002142
```