Task 4

Khaled Hasan

2024-03-15

Setup

import libraries:

```
library(fpp2)
```

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

## -- Attaching packages ---------------------------------------------- fpp2 2.5 --

## v ggplot2   3.4.4      v fma       2.5
## v forecast  8.21.1     v expsmooth 2.3

##
```

```
library(tseries)
library(urca)
library(TSA)
```

```
## Warning: package 'TSA' was built under R version 4.3.3

## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```r
library(signal)
```

```
## Warning: package 'signal' was built under R version 4.3.3
```

```
##
## Attaching package: 'signal'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, poly
```

```r
library(spectral)
```

```
## Warning: package 'spectral' was built under R version 4.3.3
```

```
## Loading required package: rasterImage
```

```
## Loading required package: plotrix
```

```
## Loading required package: lattice
```

```
## Loading required package: RhpcBLASctl
```

```
## Loading required package: pbapply
```

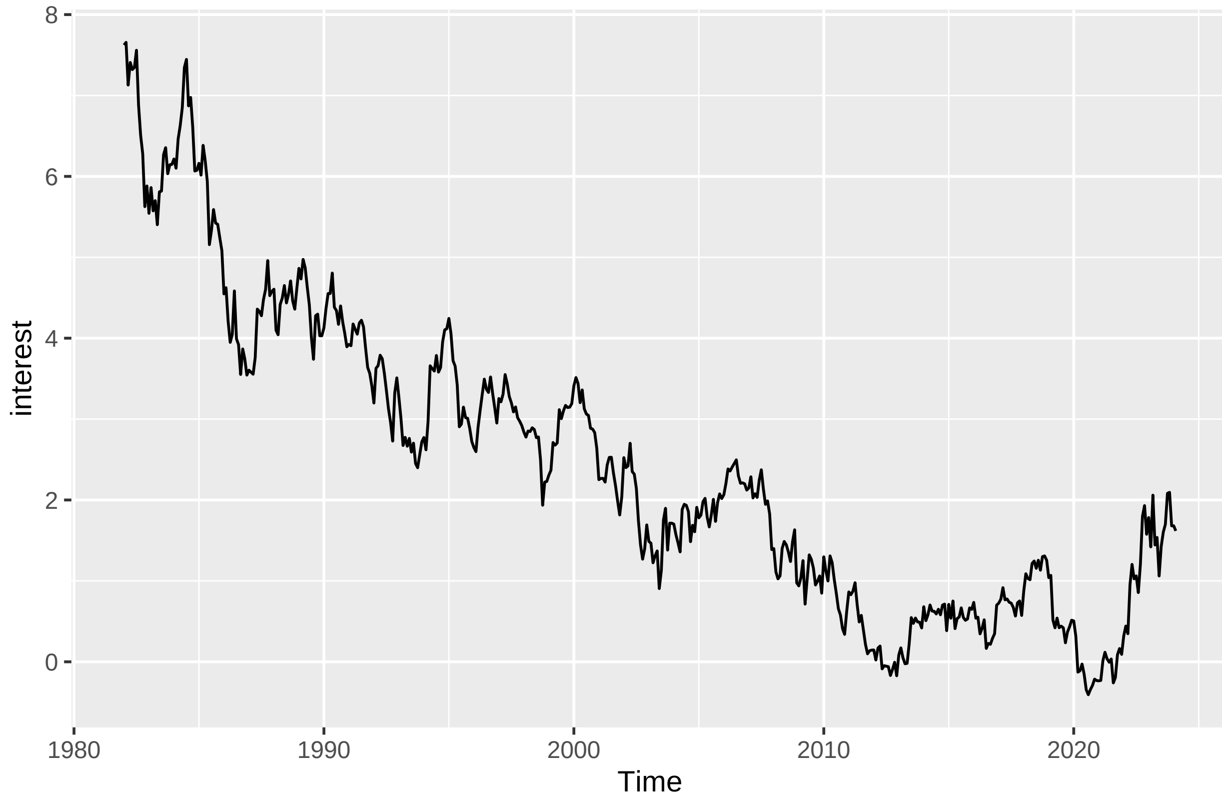
```
## Warning: package 'pbapply' was built under R version 4.3.3
```

```
## Detecting 4 cores
```

Import data

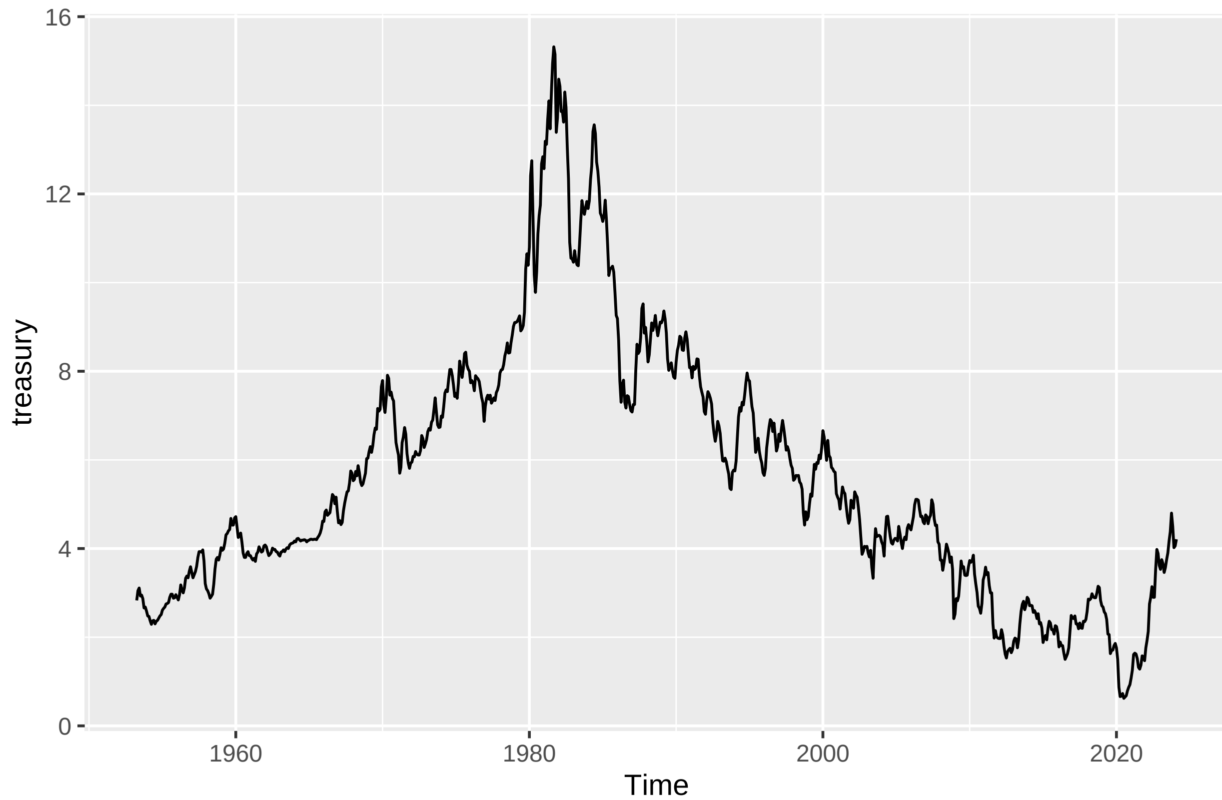
Real interest rate

```r
REAINTRATREARAT10Y <- read.csv("C:\\Users\\ss\\Downloads\\REAINTRATREARAT10Y.csv")
interest <- ts(REAINTRATREARAT10Y[, "REAINTRATREARAT10Y"], frequency = 12, start = c(1982, 1))
autoplot(interest)
```



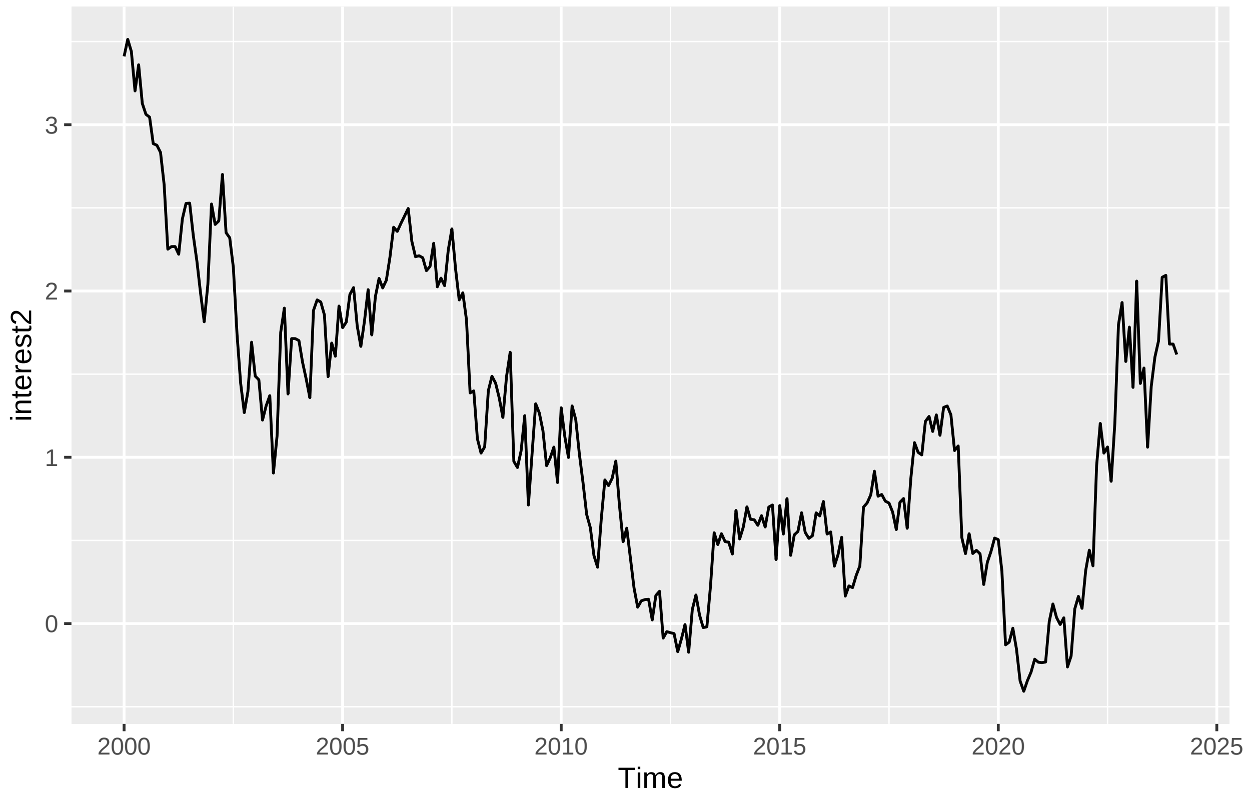
Market Yield on U.S. Treasury

```
Treasury <- read.csv("C:/Users/ss/Desktop/Time_series_Analysis/GS10.csv")
treasury <- ts(Treasury[, "GS10"], frequency = 12, start = c(1953, 4))
autoplot(treasury)
```

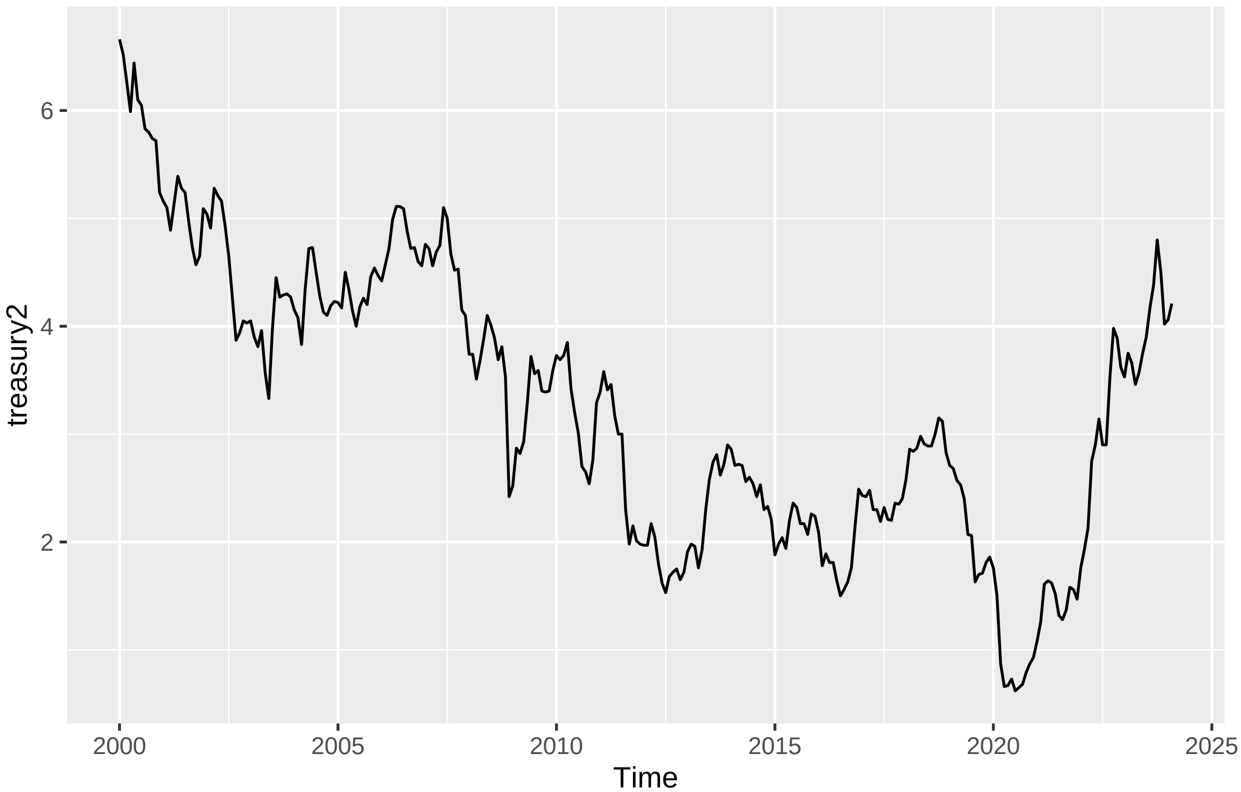


Now we cut windows beyond the year 2000

```
interest2 <- window(interest, frequency = 12, start = c(2000, 1))
autoplot(interest2)
```



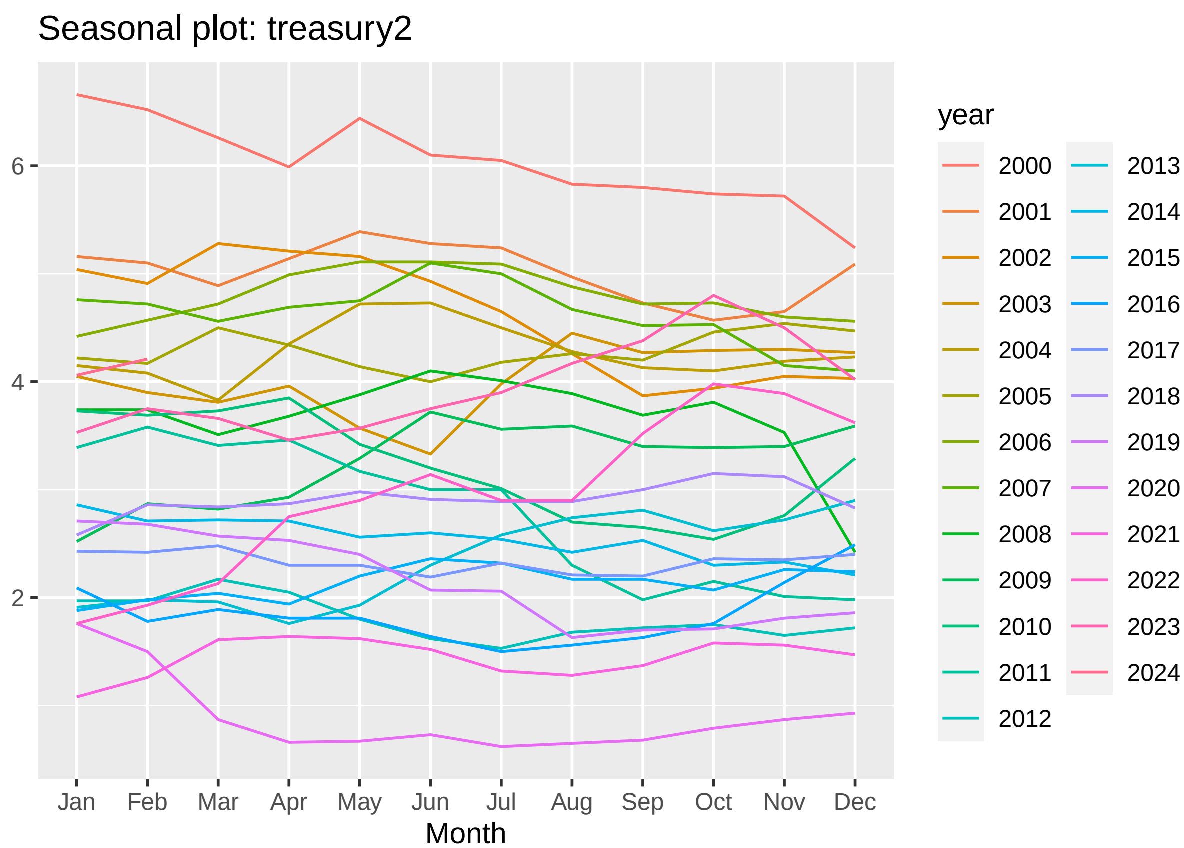

```r
treasury2 <- window(treasury, frequency = 12, start = c(2000, 1))
autoplot(treasury2)
```



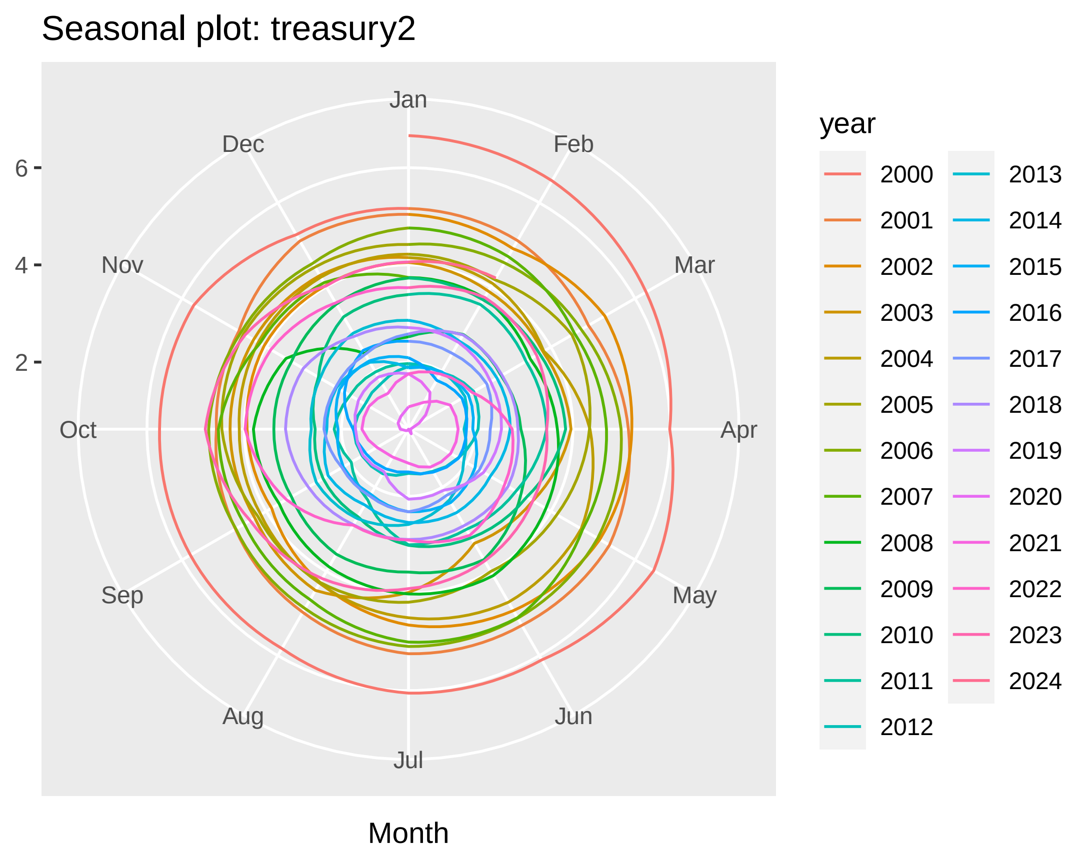
Analysis of the Treasury dataset

Checking the seasonality:

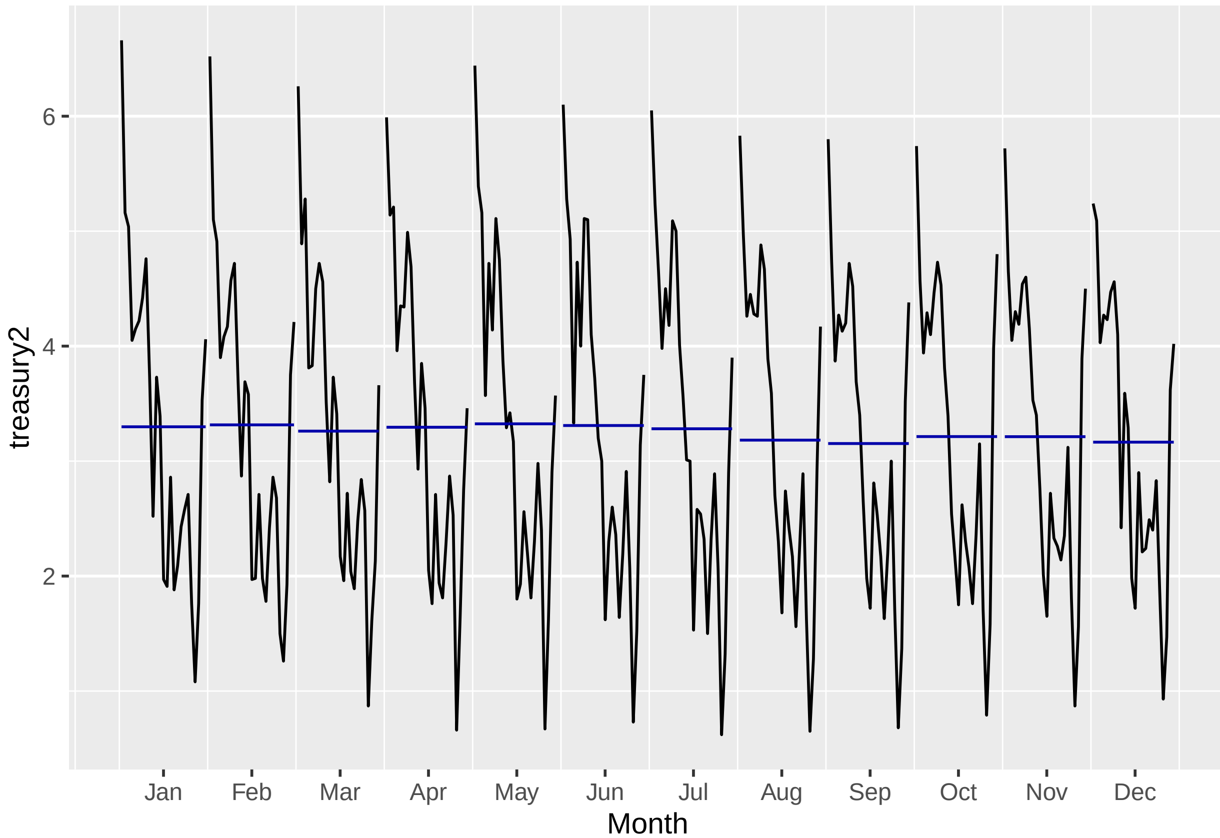
```
ggseasonplot(treasury2)
```



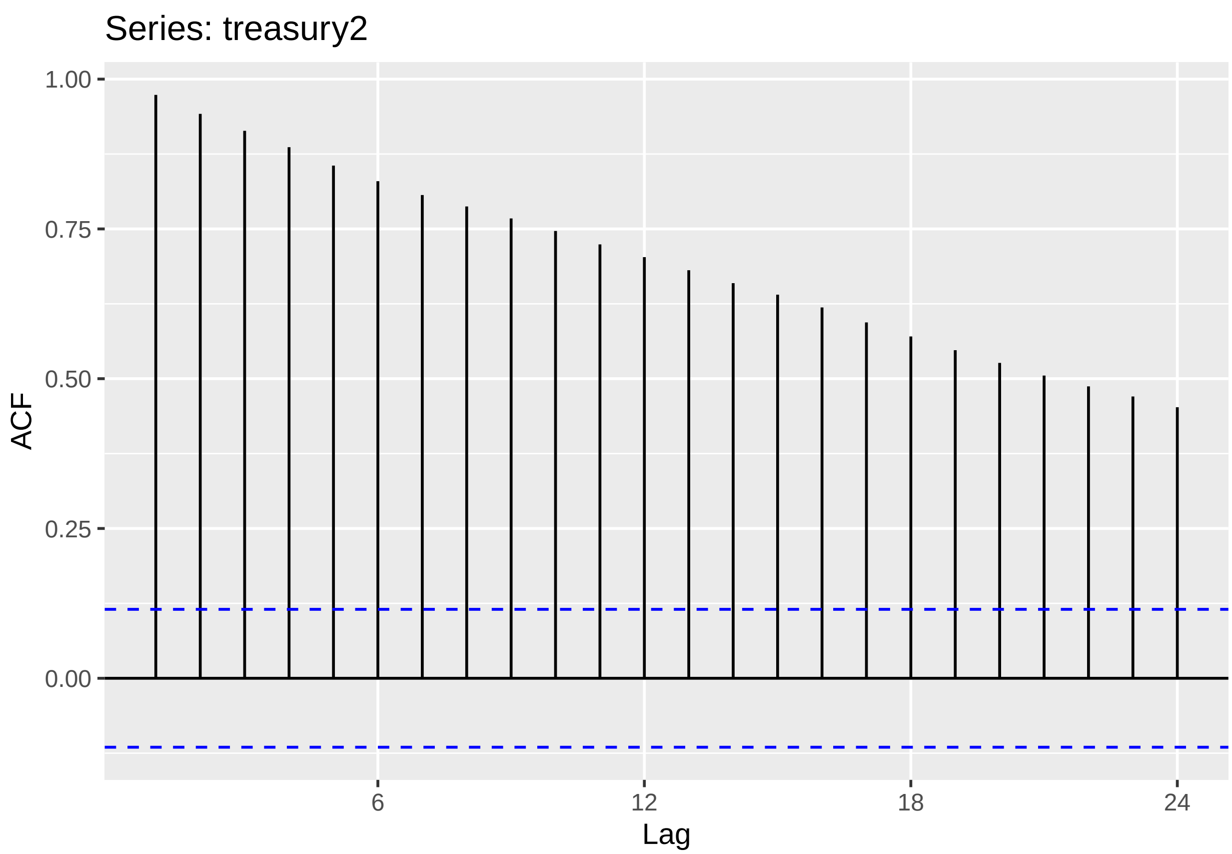

```
ggseasonplot(treasury2, polar = TRUE)
```




```
ggsubseriesplot(treasury2)
```

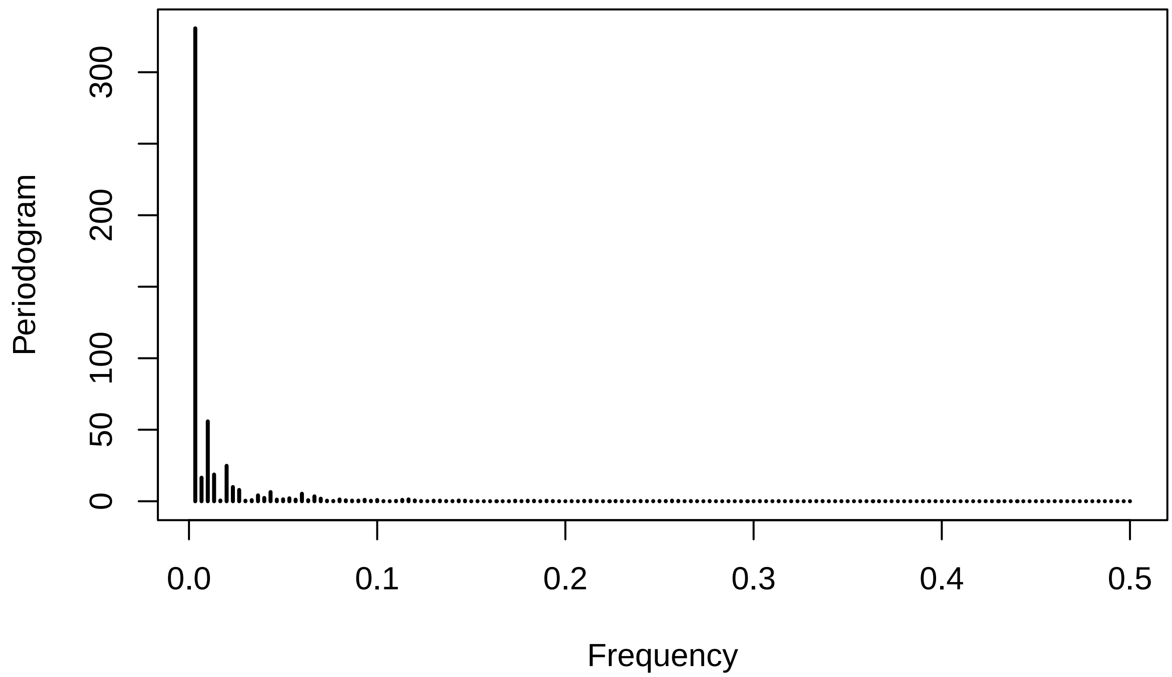



```
ggAcf(treasury2)
```

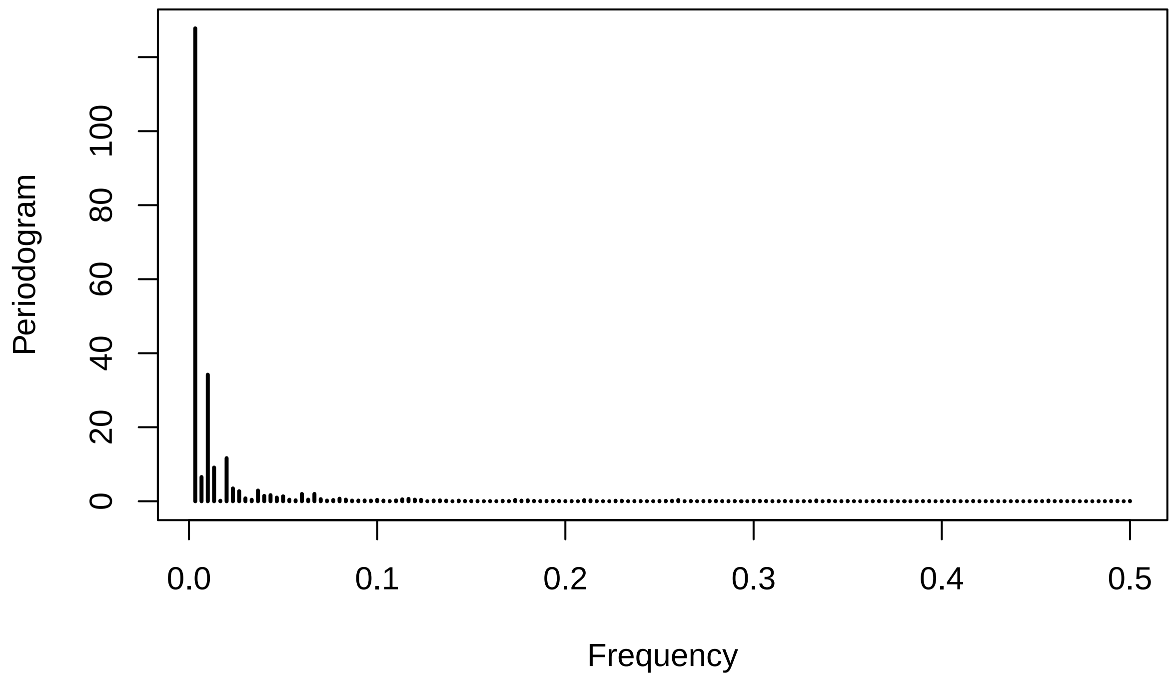


The data does not show clear seasonality.

```r
periodogram(treasury2)
```




```r
periodogram(interest2)
```



Checking Stationarity

The KPSS test (H_0 : the timeseries is stationary)

```
kpss.test(treasury2)
```

```
## Warning in kpss.test(treasury2): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  treasury2
## KPSS Level = 3.1052, Truncation lag parameter = 5, p-value = 0.01
```

$p < 0.05$ therefore, H_0 is rejected, and the time-series is not stationary.

The Dicky-Fuller test (H_0 : there exist a unit root which implies a non-stationary time series):

```
adf.test(treasury2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  treasury2
## Dickey-Fuller = -1.3653, Lag order = 6, p-value = 0.8436
## alternative hypothesis: stationary
```

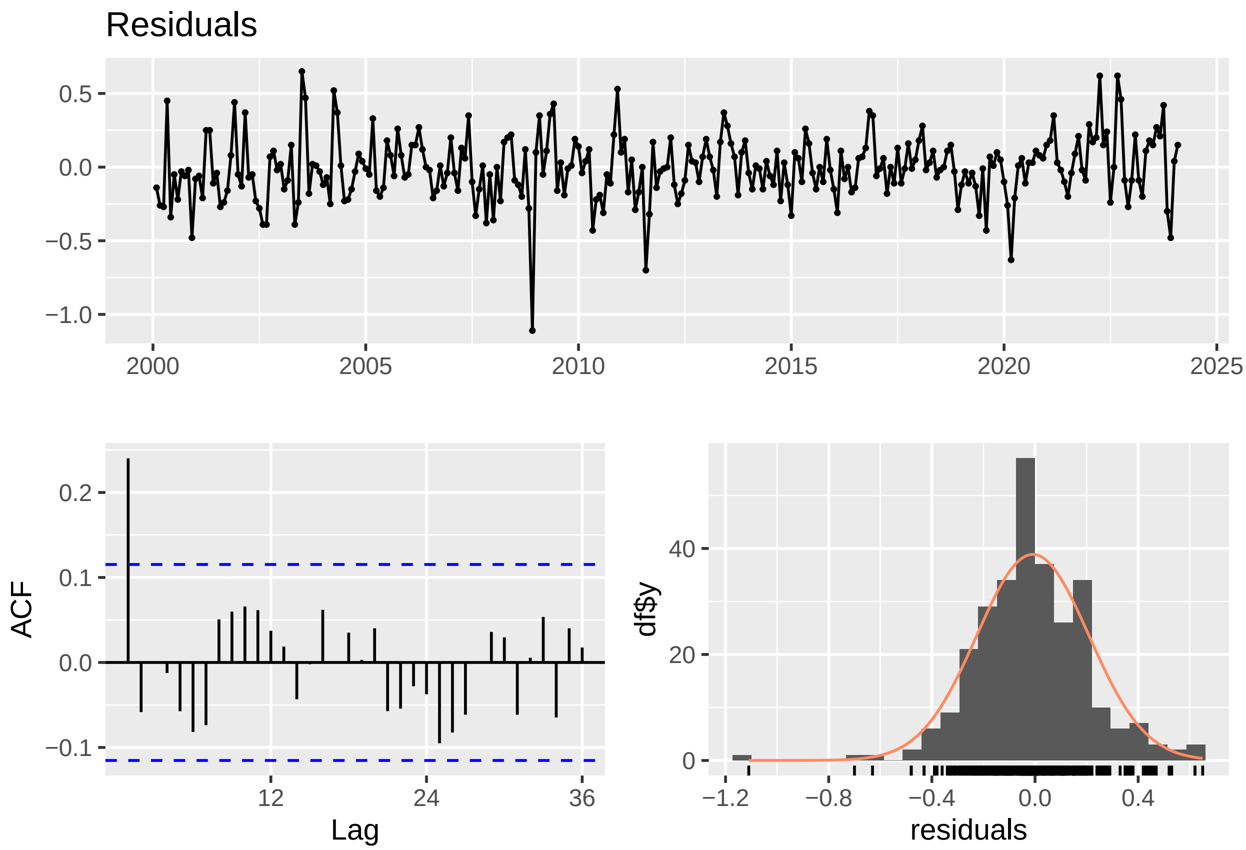
p value is large, and therefore, I will assume a non-stationary data

```r
ndiffs(treasury2)
```

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

The (ndiffs) function in R implies that one differentiation is necessary to reach stationarity.

```r
checkresiduals(diff(treasury2))
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 32.576, df = 24, p-value = 0.1133
##
## Model df: 0.   Total lags used: 24
```

The L-jung box test does not correspond to a white noise.

Arima models:

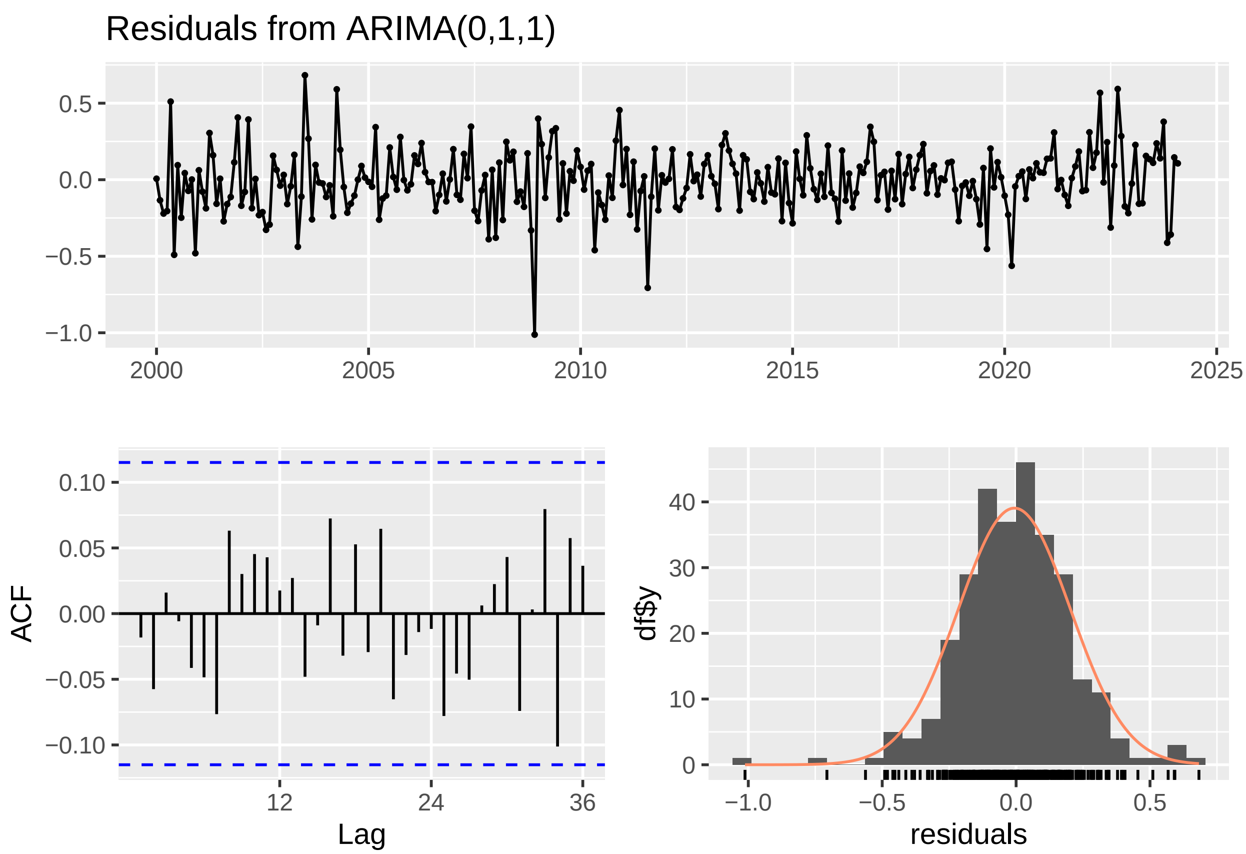
Auto Arima

```r
autofit = auto.arima(treasury2, approximation = FALSE, stepwise = FALSE)
summary(autofit)
```

```
## Series: treasury2
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.2954
## s.e.  0.0598
##
## sigma^2 = 0.04397:  log likelihood = 41.84
## AIC=-79.68   AICc=-79.64   BIC=-72.35
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE    MAPE     MASE
## Training set -0.006402272 0.2089616 0.1578887 -0.3425268 5.66649 0.231589
##                     ACF1
## Training set -0.0181802
```

Checking residuals of the automated model:

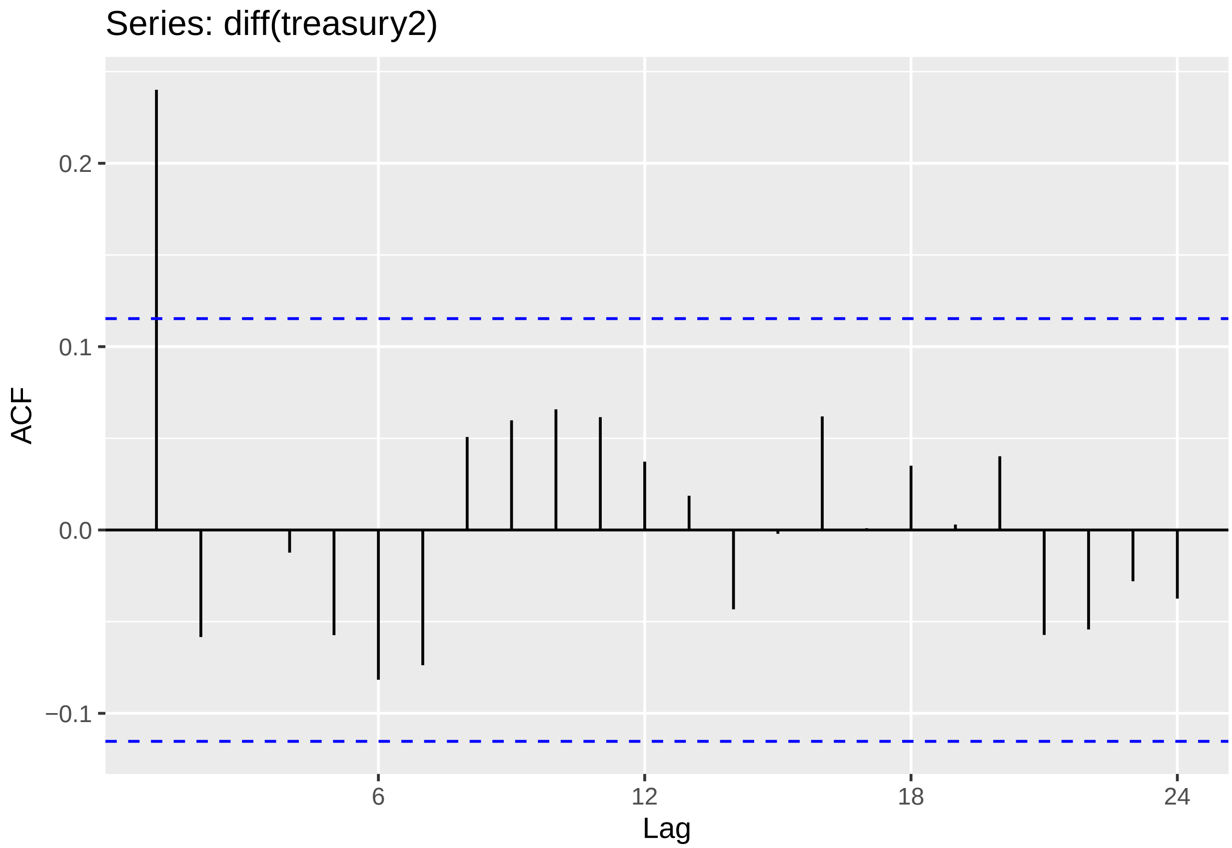
```r
checkresiduals(autofit)
```



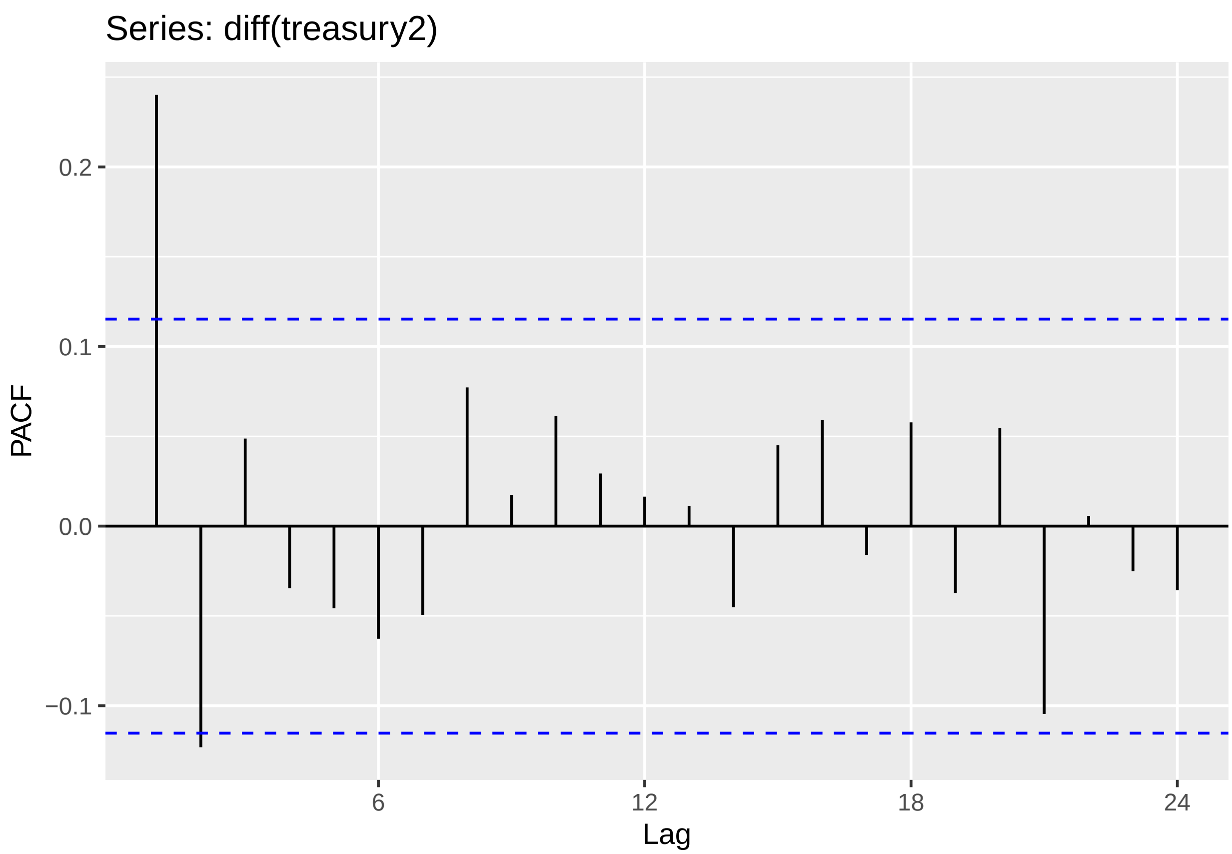

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,1)
## Q* = 13.962, df = 23, p-value = 0.928
##
## Model df: 1.   Total lags used: 24
```

since the p value is large, the null hypothesis (H_0) is not rejected, and the residuals are not correlated.

```r
ggAcf(diff(treasury2))
```



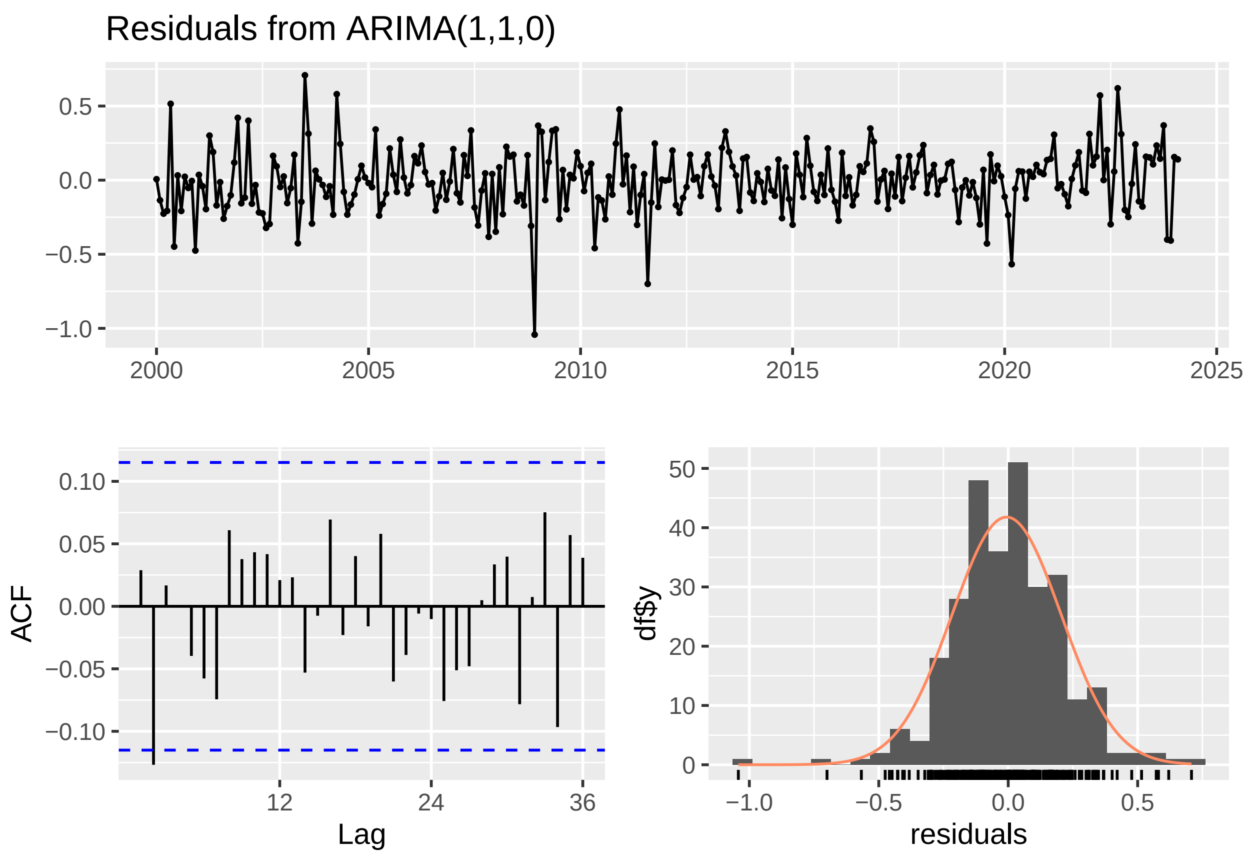

```r
ggPacf(diff(treasury2))
```



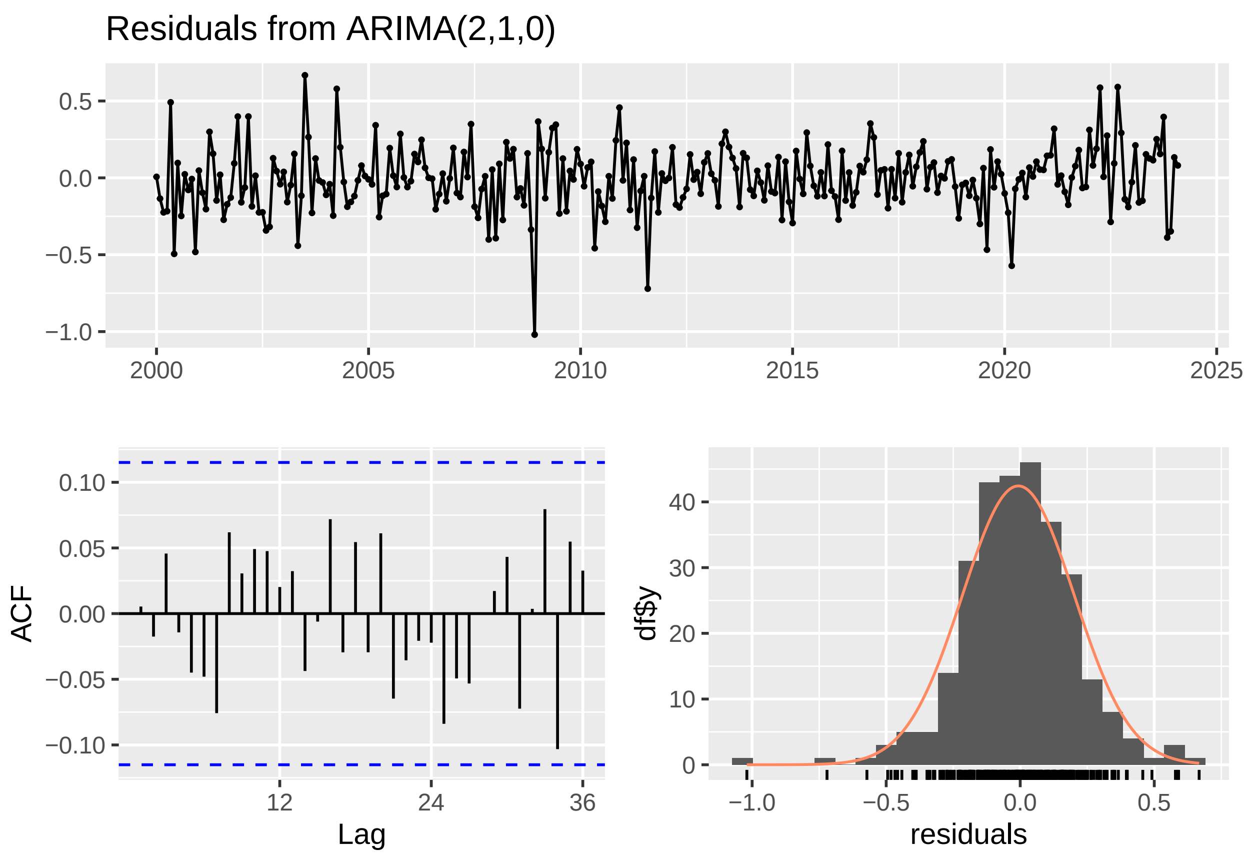
The models suggested by ACF and PACF are $(0, 1, 1)$, $(1, 1, 0)$, $(2, 1, 0)$ and $(1, 1, 1)$. Which agrees with the autoarima results.

```
ar110 = Arima(treasury2, c(1, 1, 0))
ar210 = Arima(treasury2, c(2, 1, 0))
ar111 = Arima(treasury2, c(1, 1, 1))
```

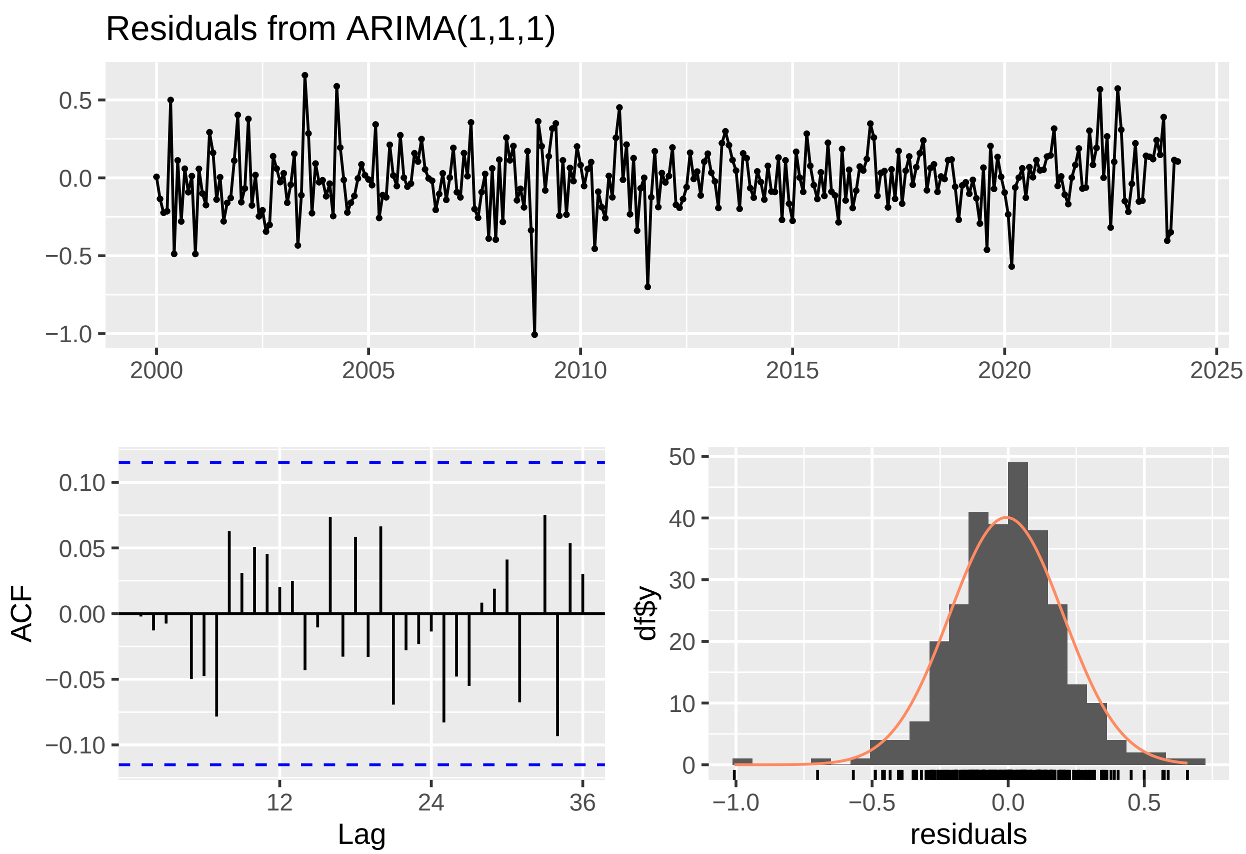
```
{
  checkresiduals(ar110)
  checkresiduals(ar210)
  checkresiduals(ar111)
}
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)
## Q* = 16.94, df = 23, p-value = 0.8122
##
## Model df: 1.   Total lags used: 24
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)
## Q* = 13.919, df = 22, p-value = 0.9043
##
## Model df: 2.   Total lags used: 24
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 13.894, df = 22, p-value = 0.9052
##
## Model df: 2.   Total lags used: 24
```

Regression

using treasury as xreg:

```
autofit_with_regression = auto.arima(interest2, xreg = treasury2, approximation = FALSE, stepwise = FALS
summary(autofit_with_regression)
```

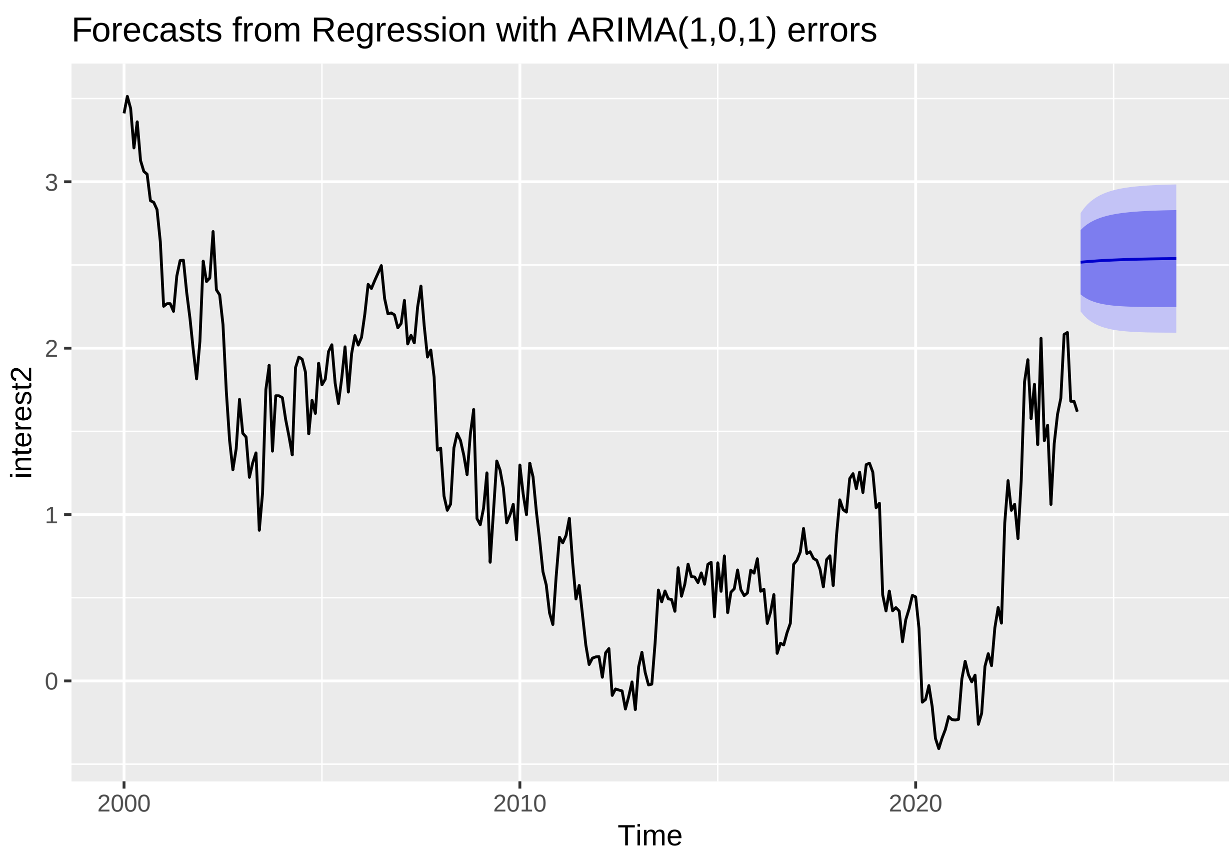
```
## Series: interest2
## Regression with ARIMA(1,0,1) errors
##
## Coefficients:
##          ar1      ma1  intercept    xreg
##       0.9254  -0.4958    -0.7930  0.5990
## s.e.  0.0303   0.0668     0.1122  0.0279
##
```

```
## sigma^2 = 0.02275:  log likelihood = 138.6
## AIC=-267.19   AICc=-266.98   BIC=-248.84
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.002582608 0.1497824 0.1129278 -6.324667 35.80835 0.2457275
##                      ACF1
## Training set -0.003194025
```

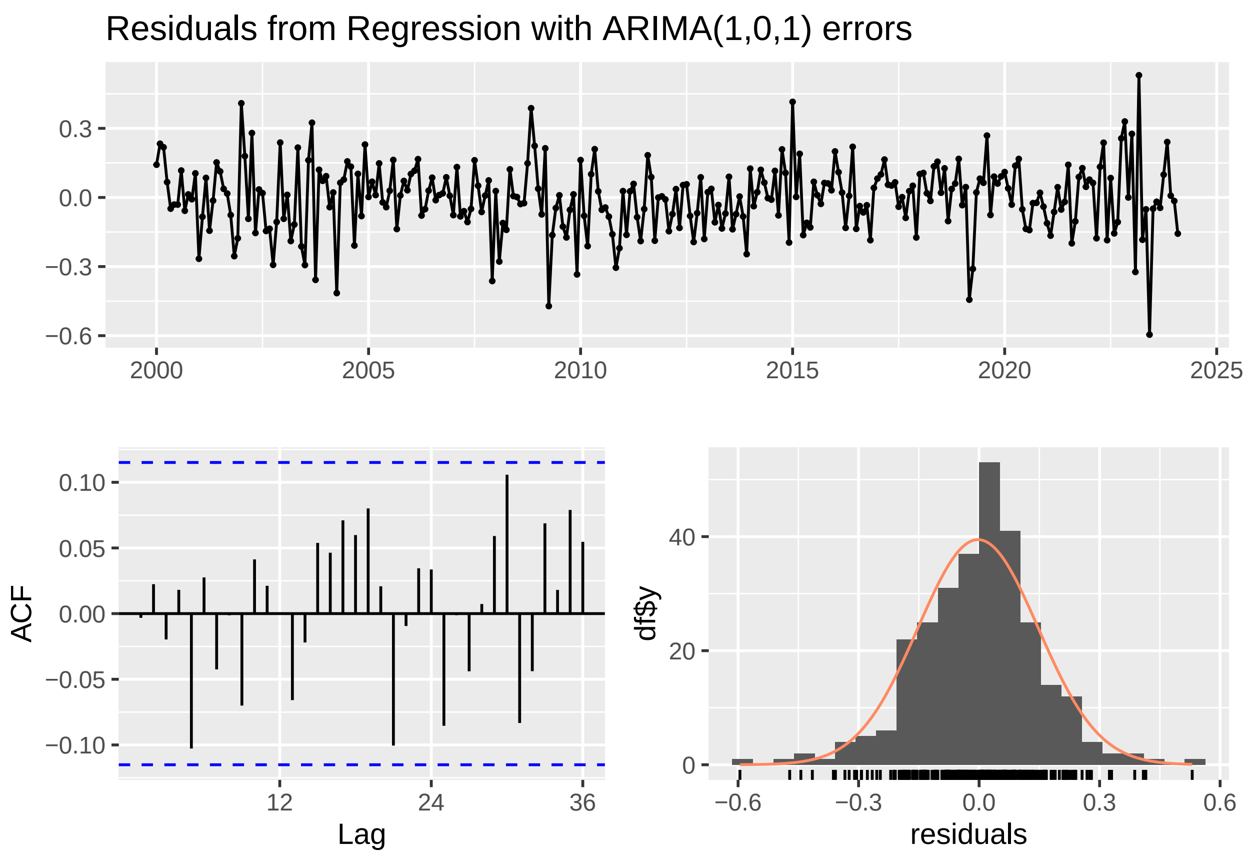
plotting a forcast

```r
treasury_reg = rep(mean(Treasury[, 2]), 30)
```

```r
autoplot(forecast(autofit_with_regression, xreg = treasury_reg))
```




```r
checkresiduals(autofit_with_regression)
```



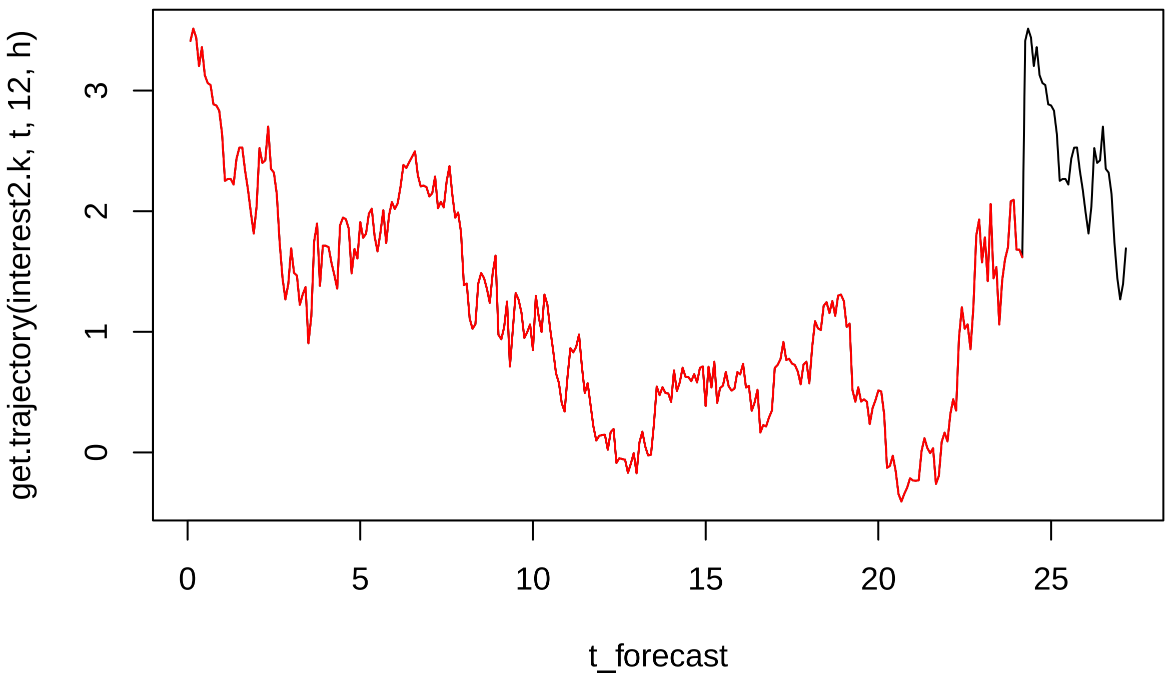

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,1) errors
## Q* = 18.191, df = 22, p-value = 0.6946
##
## Model df: 2.   Total lags used: 24
```

Fourier extrapolation

```
interest2.k = fft(interest2)

freq = 12*seq_along(interest2)/length(interest2)

t = seq_along(interest2)/12
h = 36
t_forecast = c(1:(length(interest2) + h))/12

get.trajectory <- function(X.k,ts,acq.freq, h) {

  N   <- length(ts)
  i   <- complex(real = 0, imaginary = 1)
  x.n <- rep(0,N + h)            # create vector to keep the trajectory
```

```
  ks  <- c(0:(length(X.k)-1))

  for(n in 0:(N-1 + h)) {        # compute each time point x_n based on freqs X.k
    x.n[n+1] <- sum(X.k * exp(i*2*pi*ks*n/N)) / N
  }

  x.n
}

plot(t_forecast, get.trajectory(interest2.k, t, 12, h), type = "l") +
lines(t[1:length(interest2)], interest2, type = "l", col = "red")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): imaginary parts discarded in
## coercion
```

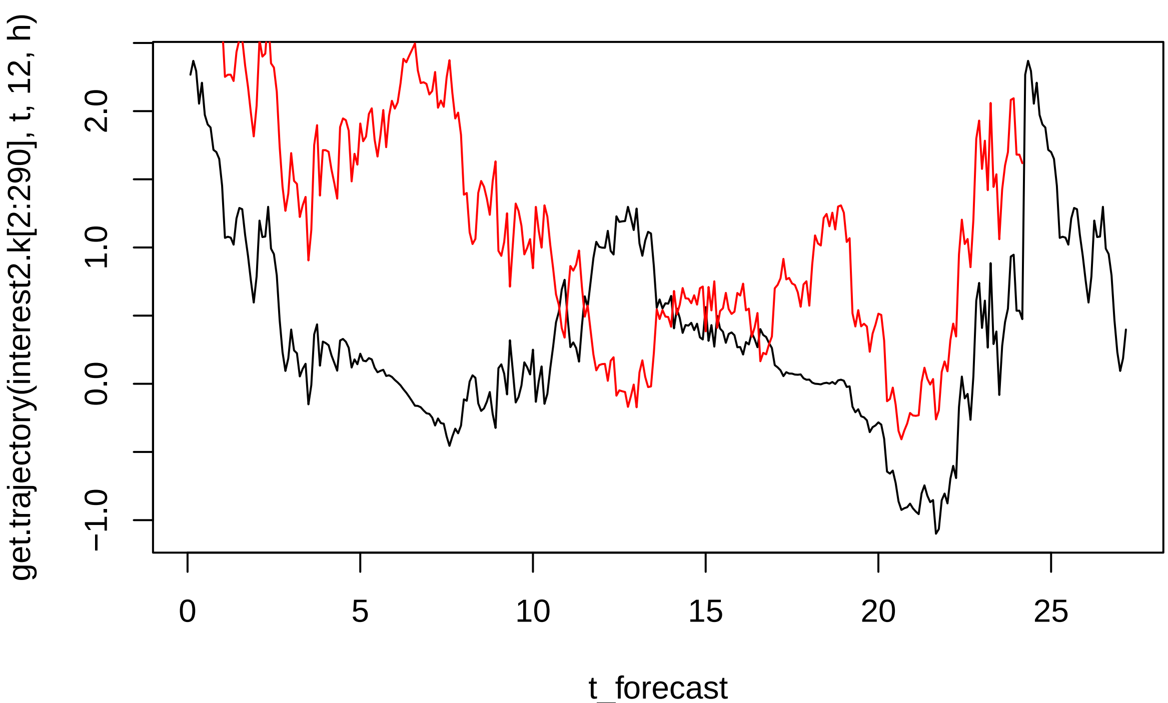



```
## integer(0)
```

```
interest2.k = fft(interest2)

freq = 12*seq_along(interest2)/length(interest2)

t = seq_along(interest2)/12
h = 36
t_forecast = c(1:(length(interest2) + h))/12
```

```
get.trajectory <- function(X.k,ts,acq.freq, h) {

  N   <- length(ts)
  i   <- complex(real = 0, imaginary = 1)
  x.n <- rep(0,N + h)              # create vector to keep the trajectory
  ks  <- c(0:(length(X.k)-1))

  for(n in 0:(N-1 + h)) {        # compute each time point x_n based on freqs X.k
    x.n[n+1] <- sum(X.k * exp(i*2*pi*ks*n/N)) / N
  }

  x.n
}

plot(t_forecast, get.trajectory(interest2.k[2:290], t, 12, h), type = "l") +
lines(t[1:length(interest2)], interest2, type = "l", col = "red")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): imaginary parts discarded in
## coercion
```




```
## integer(0)
```

Parameters Stability

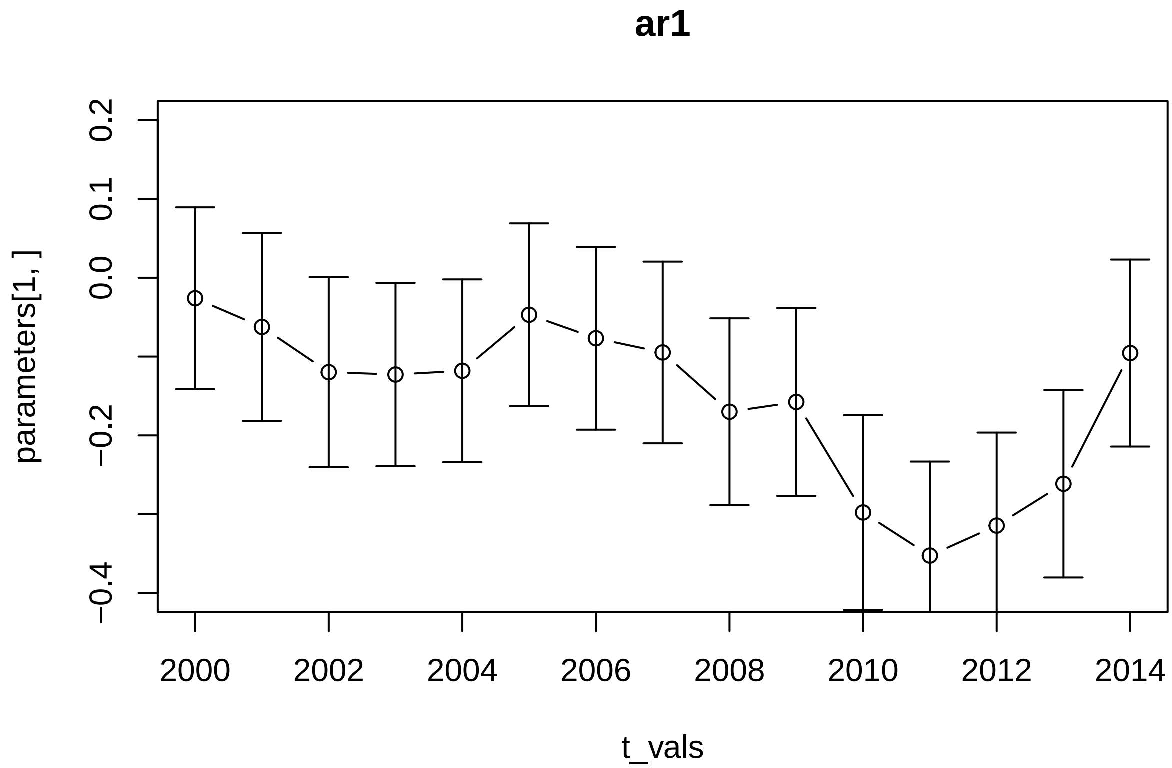
Rolling Window

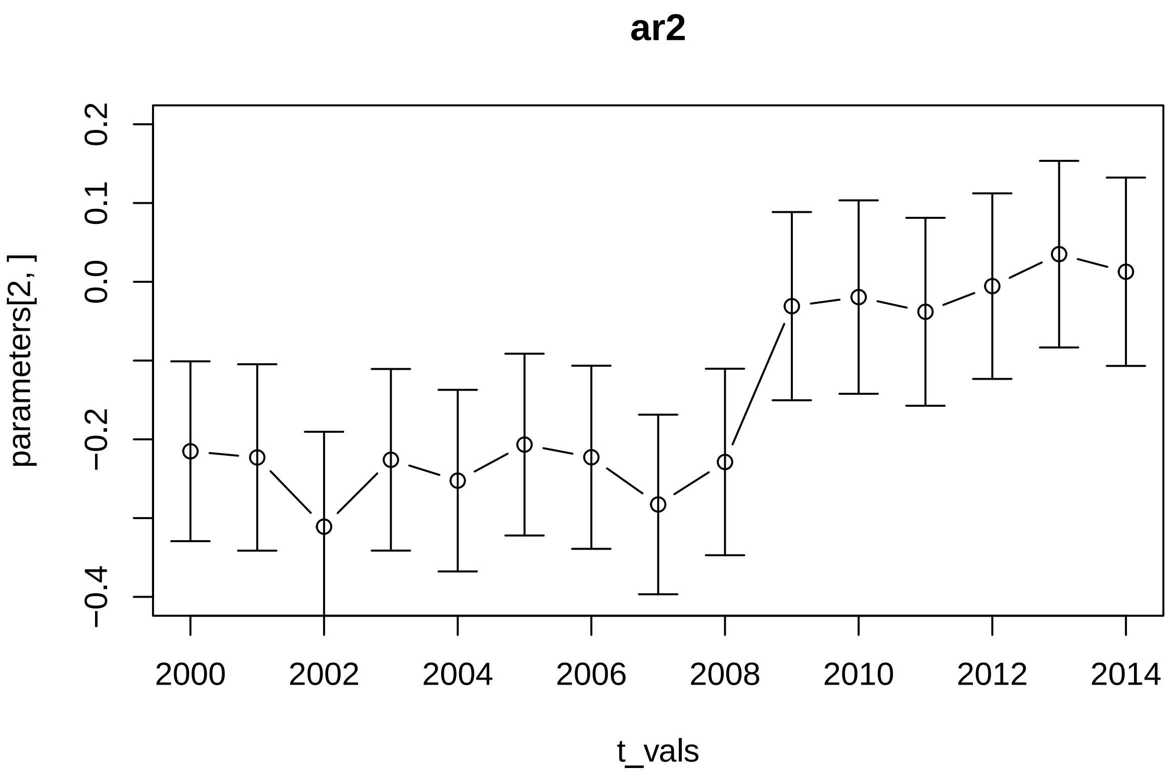
Throughout the following tests, I will be considering the

```
parameters =c()
errors = c()

for (i in 1:15){
    parameters <- cbind(parameters, coefficients(Arima(window(interest, frequency = 12, start=c(2000+i,
    errors <- cbind(errors, sqrt(diag(vcov(Arima(window(interest, frequency = 12, start=c(2000+i, 1), e
)
}

t_vals = c(2000:2014)
```

due to the small sample sizes I preferred to keep the default aicc optimization scheme, rather than the aic.

```
{
  plot(t_vals, parameters[1, ], type='b', main="ar1", ylim = c(-0.4, 0.2))+
  arrows(x0=t_vals, y0=parameters[1, ]-errors[1, ], x1 = t_vals, y1=parameters[1, ]+errors[1, ], code=3
  plot(t_vals, parameters[2, ], type='b', main="ar2", ylim = c(-0.4, 0.2))+
  arrows(x0=t_vals, y0=parameters[2, ]-errors[2, ], x1 = t_vals, y1=parameters[2, ]+errors[2, ], code=3
}
```






```
## integer(0)
```

There is a noticable jumps between the windows that started at 2009 and the windows that started at 2010. this might be due to the aftermath of the 2008 financial crisis.

```
cbind('ar1' = parameters[1,],
      's.e(ar1)' = errors[1, ],
      'ar2' = parameters[2,],
      's.e(ar2)' = errors[2, ]
      )
```

```
##               ar1  s.e(ar1)          ar2  s.e(ar2)
##  [1,] -0.02601037 0.1153361 -0.215162092 0.1141397
##  [2,] -0.06239501 0.1191702 -0.223022810 0.1183246
##  [3,] -0.11978263 0.1206137 -0.310855374 0.1204454
##  [4,] -0.12278747 0.1162711 -0.225992193 0.1152978
##  [5,] -0.11803613 0.1159529 -0.252453295 0.1152734
##  [6,] -0.04691432 0.1159329 -0.206683331 0.1154110
##  [7,] -0.07676653 0.1159690 -0.222783660 0.1162489
##  [8,] -0.09478386 0.1152805 -0.282702201 0.1140171
##  [9,] -0.16999576 0.1185005 -0.228764446 0.1184123
## [10,] -0.15757921 0.1191645 -0.030939217 0.1194848
## [11,] -0.29778598 0.1235024 -0.019411247 0.1227973
## [12,] -0.35248298 0.1192655 -0.038054992 0.1192855
## [13,] -0.31445714 0.1180232 -0.005510709 0.1177982
```

```
## [14,] -0.26137517 0.1188981  0.035087585 0.1184636
## [15,] -0.09552904 0.1186173  0.012763916 0.1195569
```