week-2.R

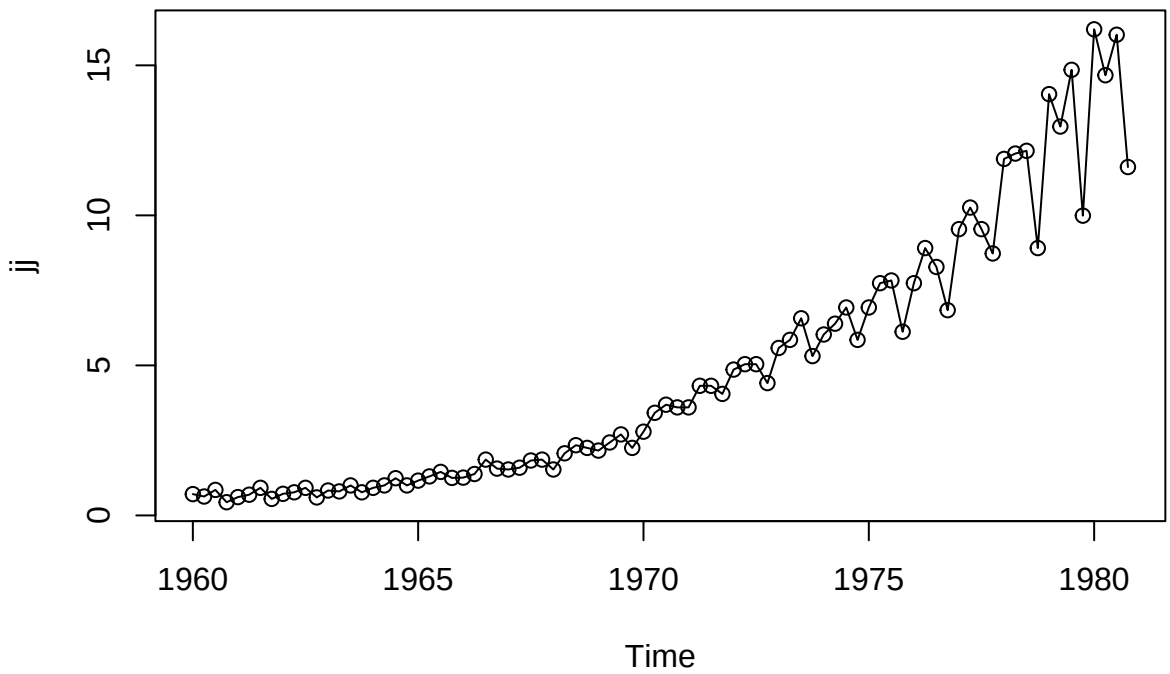
Ahmed

2023-03-01

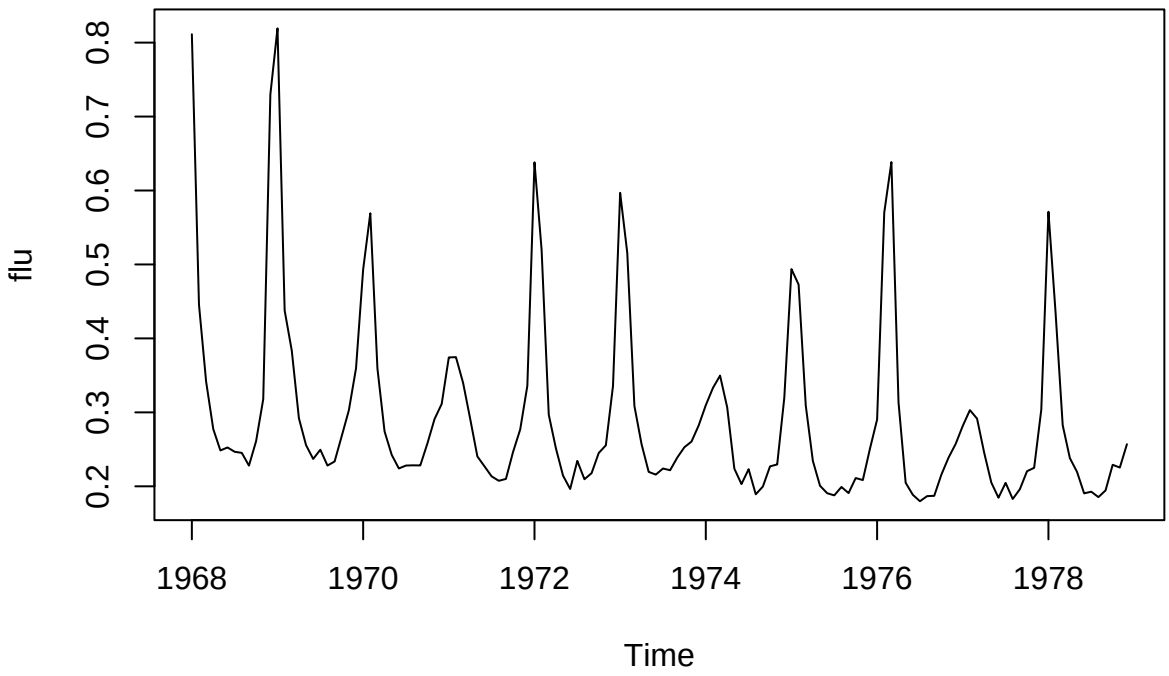
```
library(astsa)
help(astsa)
```

```
## starting httpd help server ... done
```

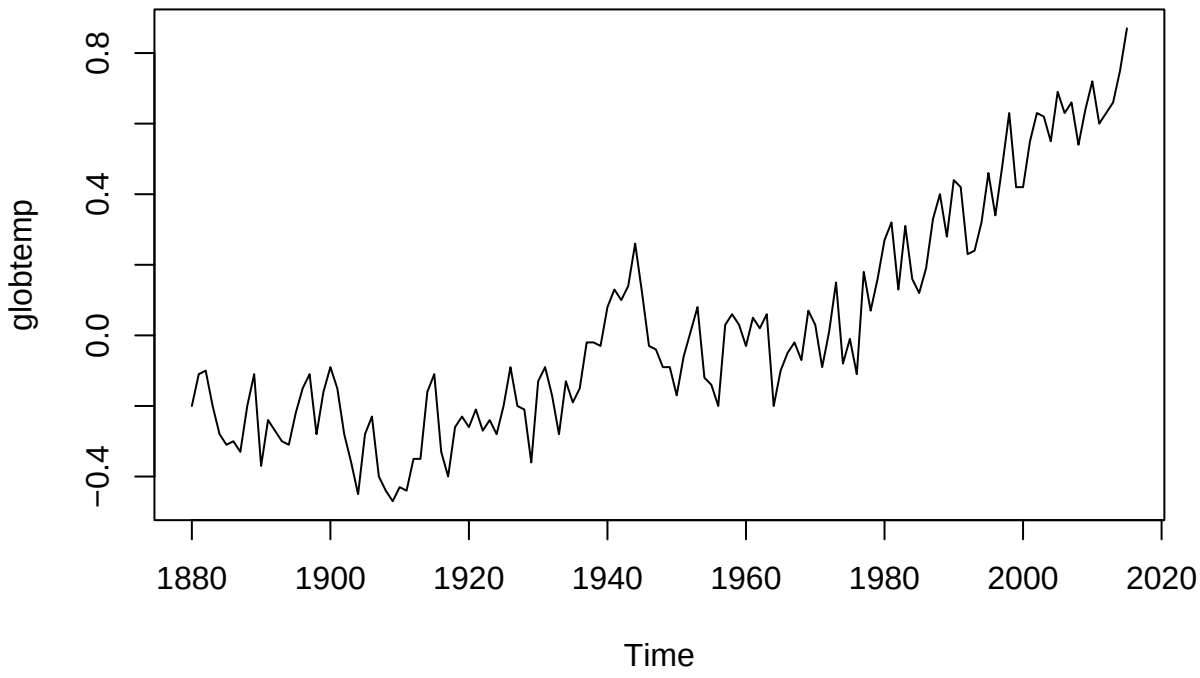
```
help(jj)
plot(jj,type="o")
```



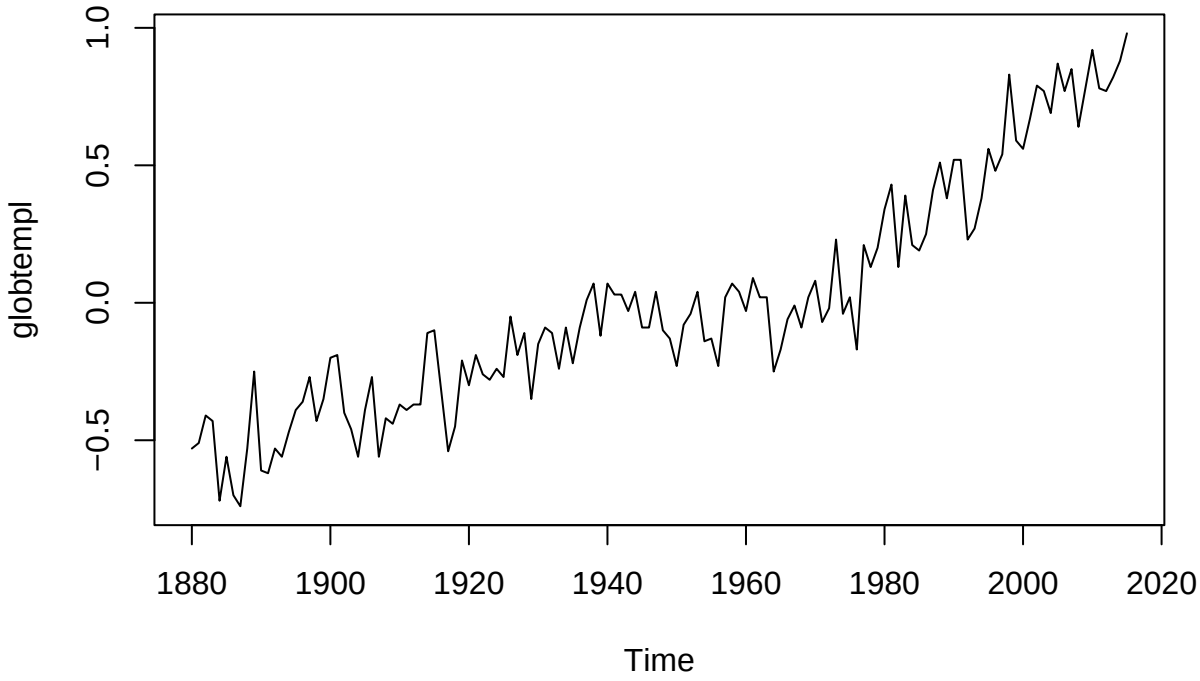

```
help("flu")
plot(flu)
```



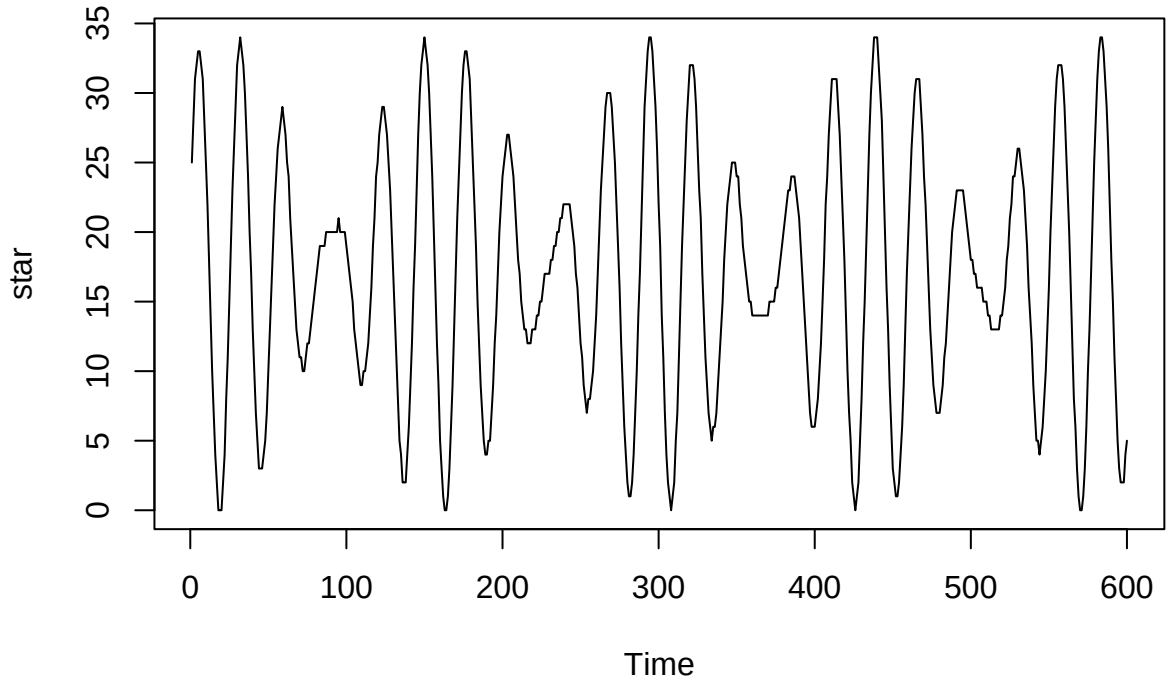

```
plot(globtemp)
```




```
plot(globtempl)
```


globtempl
1.0
0.5
0.0
−0.5
1880
1900
1920
1940
1960
1980
2000
2020
Time

```
plot(star)
```

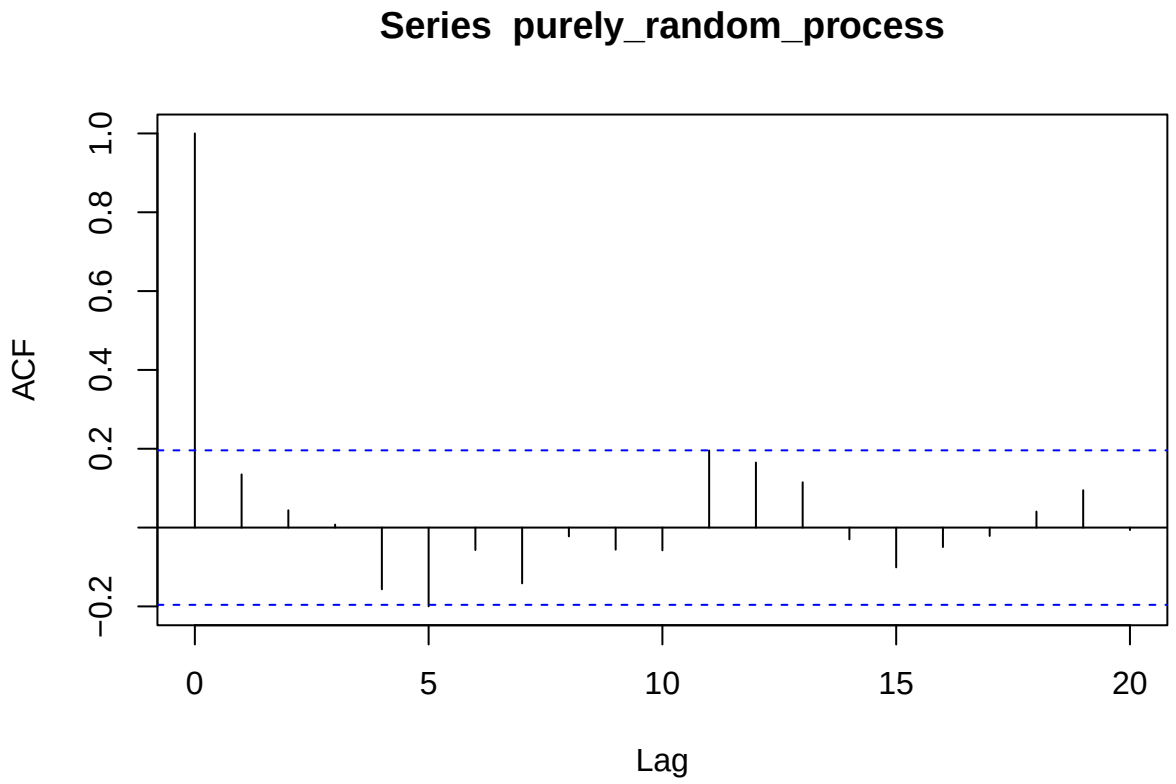



```
purely_random_process <- ts(rnorm(100))
purely_random_process
```

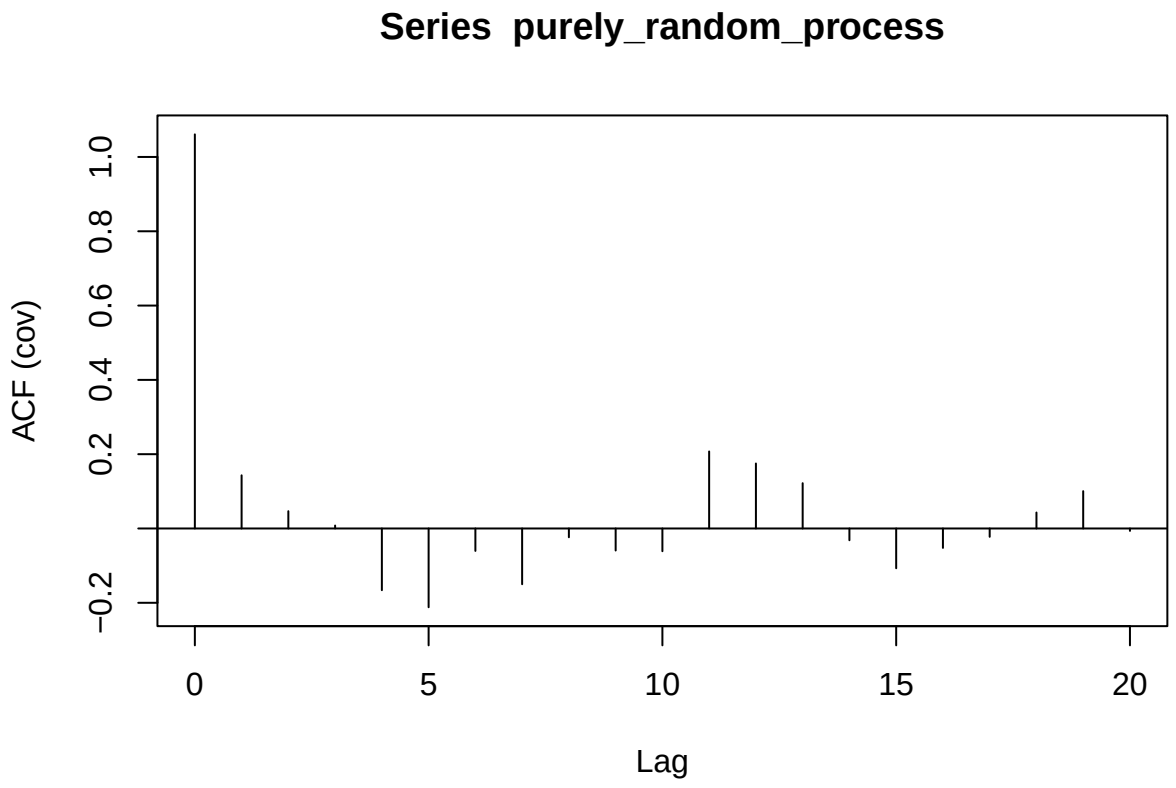
```
## Time Series:
## Start = 1
## End = 100
## Frequency = 1
##   [1] -1.924419161 -0.595502697 -0.426557118 -0.115622360 -0.525853976
##   [6]  1.165891270 -0.238217602  1.460628062 -0.568835896  0.306824952
##  [11] -0.340919882  0.314544586 -0.880168742 -0.494966750  0.734412062
##  [16]  1.544122025  0.741542744 -0.915626363 -0.295663362  0.024634420
##  [21] -0.457777572 -1.310905820  0.625475833 -0.265748318 -2.331913273
##  [26] -1.473398619  1.209261202  0.862764311  0.742896518  1.378374216
##  [31] -0.646304110  1.076536893 -1.806460058 -0.242351402 -0.002067967
##  [36] -0.607770047  0.403275412 -0.091659876 -0.375852208  1.427391641
##  [41]  1.101574674 -0.937466743 -1.843212188 -0.709421820 -1.529249766
##  [46]  0.186410245  0.325138767  1.487293077 -0.014944357  0.704668927
##  [51]  0.477290563 -0.814024485  1.484497955 -2.552548227 -0.759381802
##  [56]  0.827253681 -1.027107097 -0.073329841  1.572764751 -1.462689900
##  [61] -0.772746896 -0.322254649  0.304581325  1.003600123  0.420255473
##  [66] -0.387691841 -1.310306383  0.191719740  0.849953156  1.068327684
##  [71] -1.200598807  0.425307209 -0.846598035 -0.422867498  0.142830449
##  [76] -0.539880475  0.579258787 -0.077438268 -1.264772261  0.392949877
##  [81]  0.465229043 -0.032626028  1.768608356  1.617345283  0.671366049
##  [86]  0.015425257 -1.996660033  0.015101783 -0.187095767 -1.215910926
```

```
##  [91] -0.667413984  0.652620926 -1.126050668  1.862266192  2.174384282
##  [96]  2.357234833  0.764203966  1.673185976  0.289710654 -0.134972327
```

```
acf(purely_random_process)
```

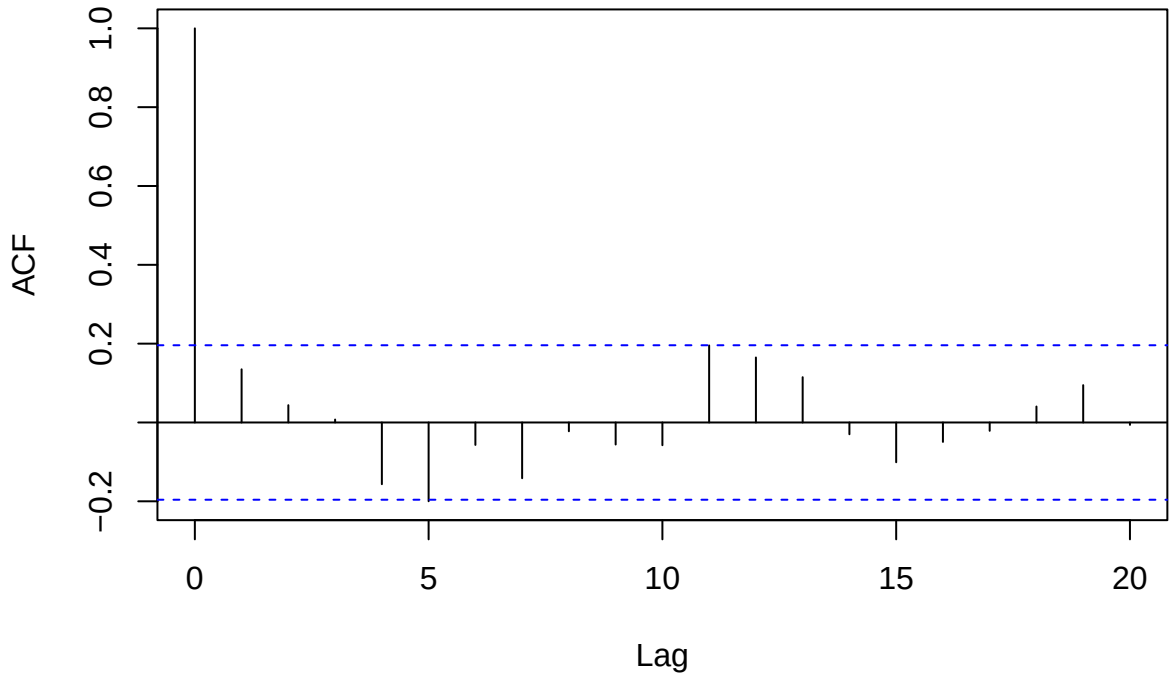



```
(acf(purely_random_process,type = "covariance"))
```



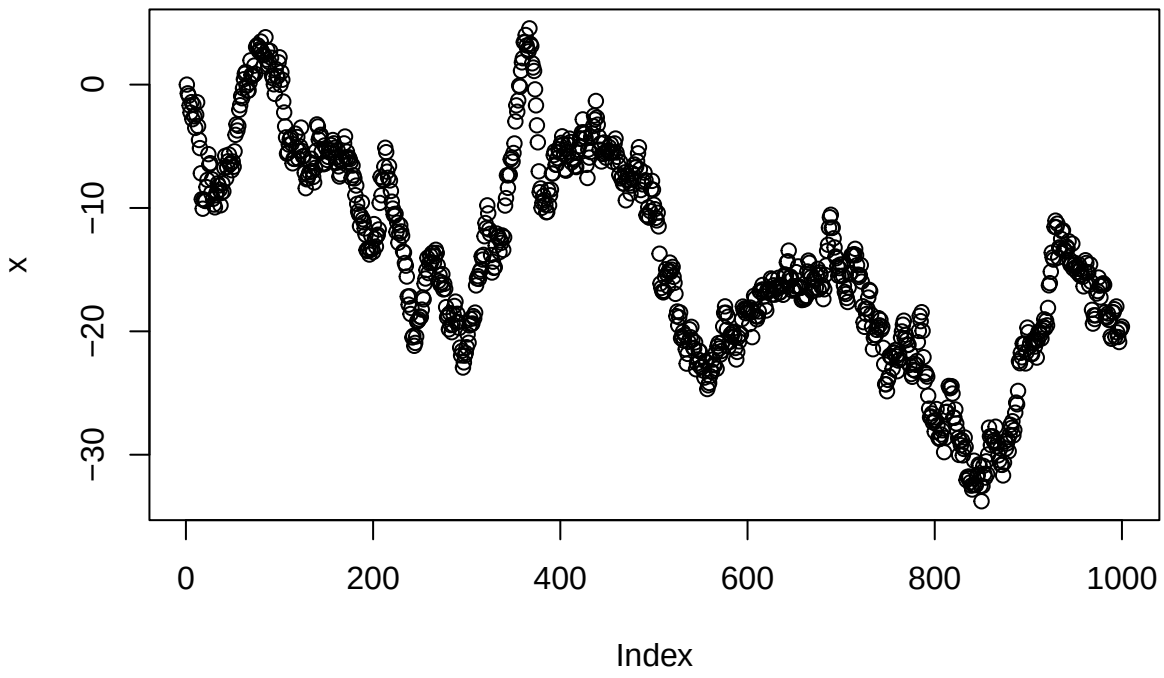

```
##
## Autocovariances of series 'purely_random_process', by lag
##
##        0        1        2        3        4        5        6        7
##  1.06087  0.14317  0.04659  0.00802 -0.16610 -0.21198 -0.06039 -0.15004
##        8        9       10       11       12       13       14       15
## -0.02363 -0.05932 -0.06122  0.20739  0.17502  0.12195 -0.03169 -0.10707
##       16       17       18       19       20
## -0.05231 -0.02243  0.04305  0.10054 -0.00648
```

```
(acf(purely_random_process,main="Correlogram of purely random process"))
```

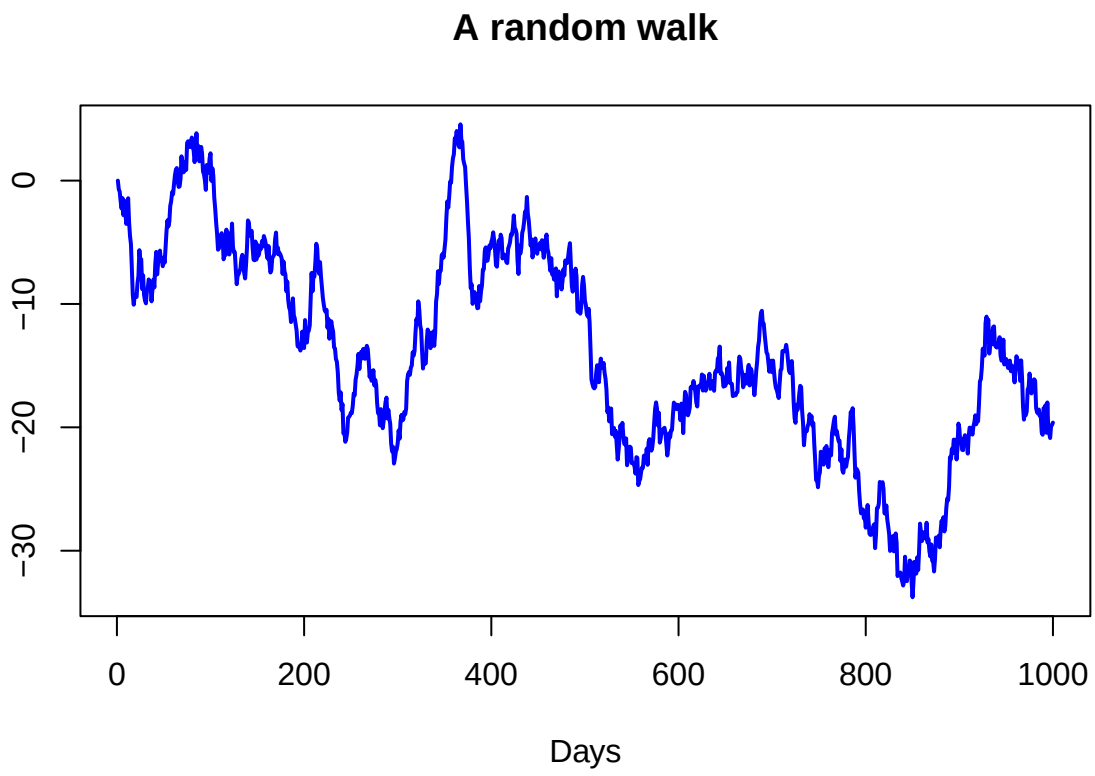



```
##
## Autocorrelations of series 'purely_random_process', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.135  0.044  0.008 -0.157 -0.200 -0.057 -0.141 -0.022 -0.056 -0.058
##     11     12     13     14     15     16     17     18     19     20
##  0.195  0.165  0.115 -0.030 -0.101 -0.049 -0.021  0.041  0.095 -0.006
```

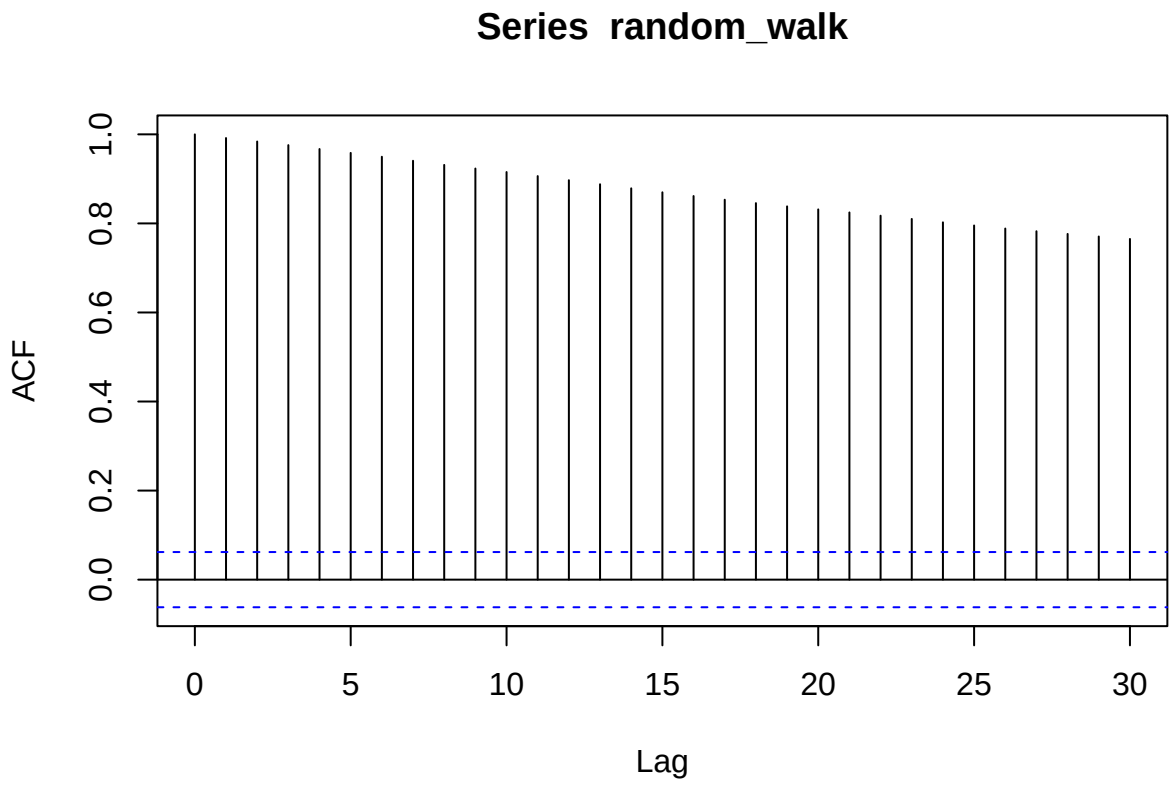
```r
# Random walk
x <- NULL
x[1] <- 0
for (i in 2:1000){
  x[i] <- x[i-1]+rnorm(1)
  }
plot(x)
```



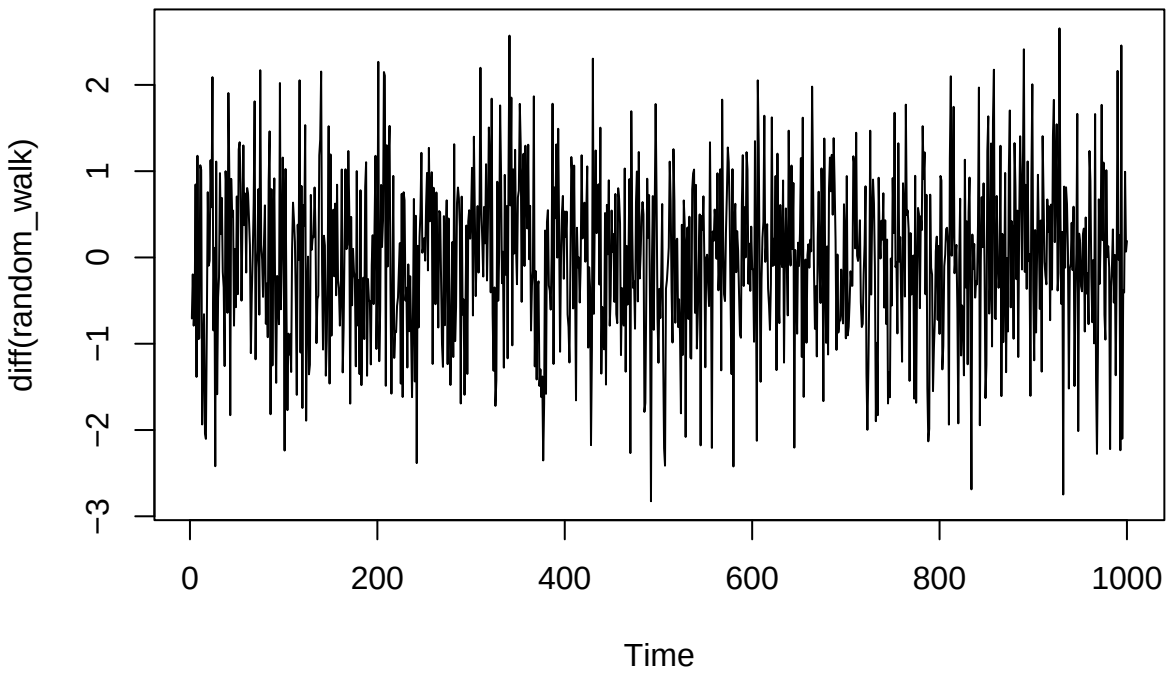

```
random_walk <- ts(x)
plot(random_walk,main="A random walk",ylab=" ",xlab="Days",col="blue",lwd=2)
```



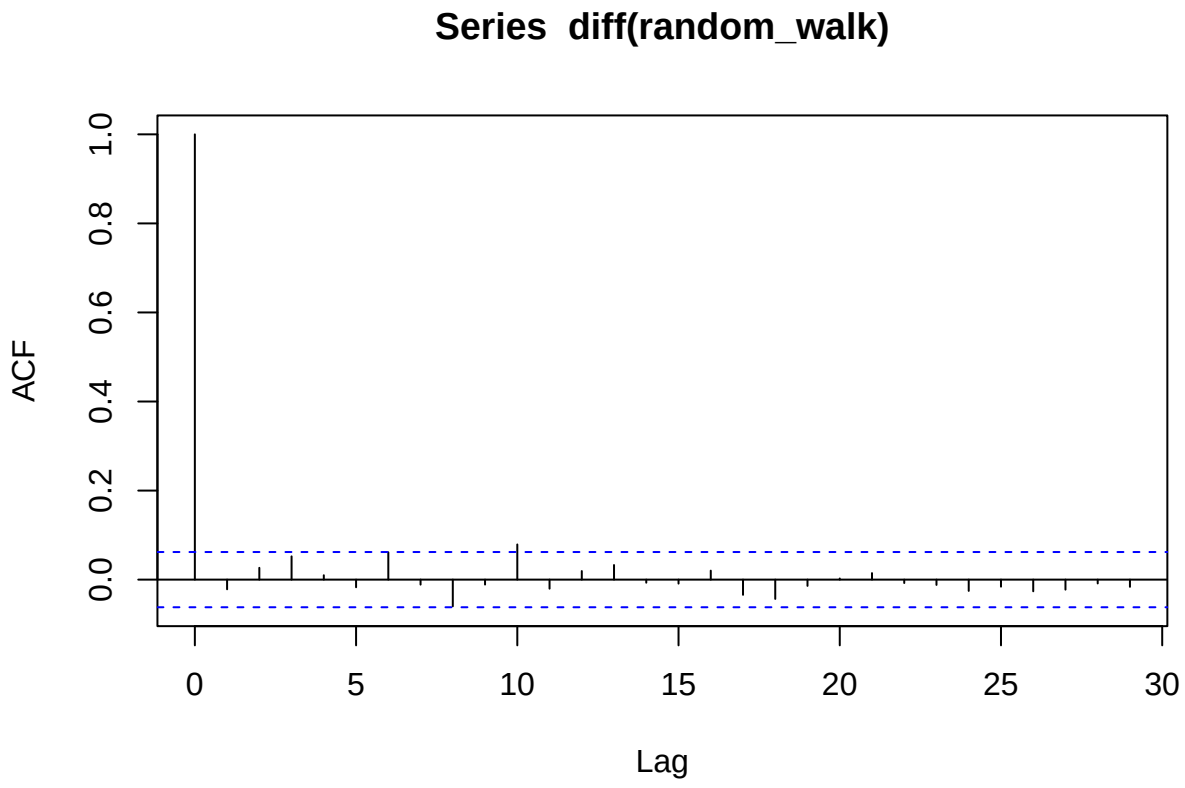

```
acf(random_walk)
```



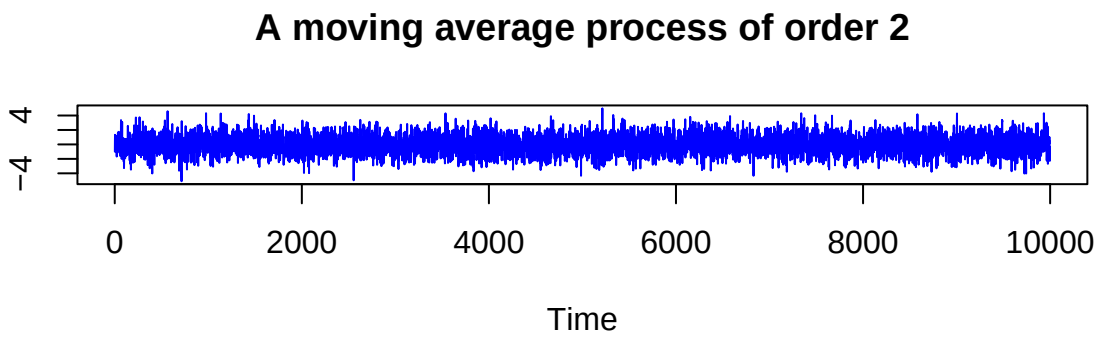

```
plot(diff(random_walk))
```

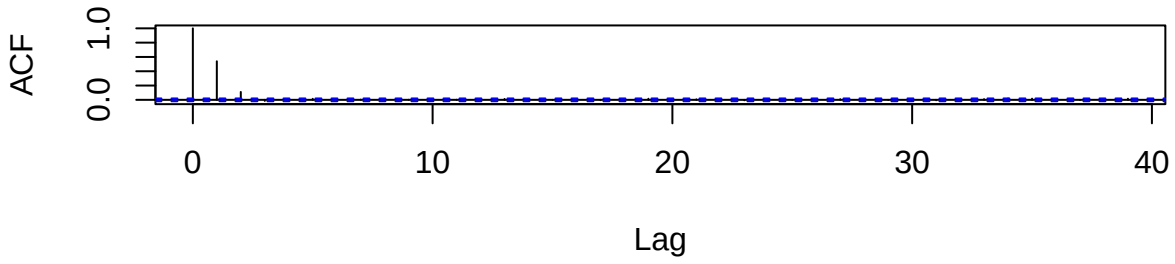



```
acf(diff(random_walk))
```




```r
# Generate noise
noise = rnorm(10000)
movingAvg <- NULL
for(i in 3:10000){
  movingAvg[i] <- noise[i]+0.7*noise[i-1]+0.2*noise[i-2]
}
movingAvgProcess = movingAvg[3:10000]
movingAvgProcess <- ts(movingAvgProcess)
#partition output graphics
par(mfrow=c(2,1))
plot(movingAvgProcess,main="A moving average process of order 2",ylab=" ",col="blue")
acf(movingAvgProcess,main="Corellogram of a moving average process of order 2")
```



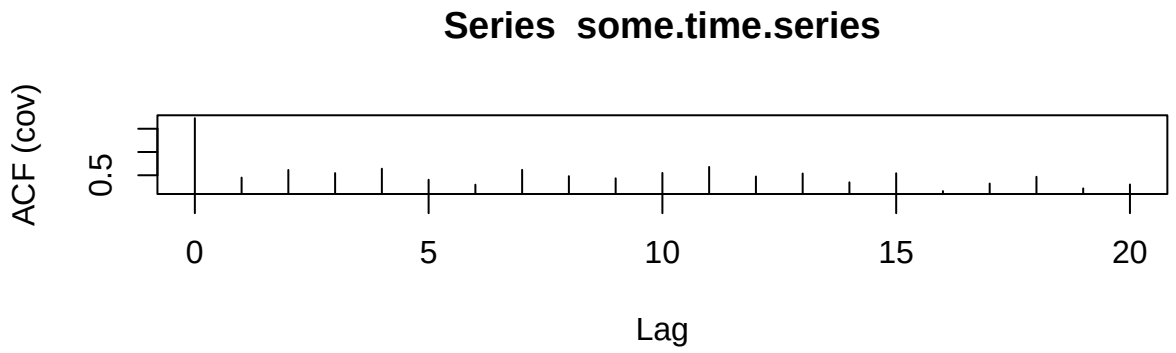


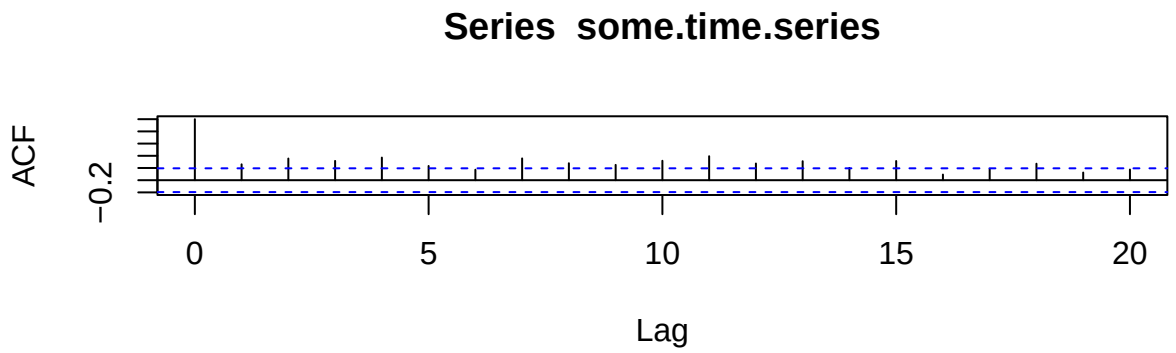

```
# Quiz
# Simulating a non-stationary time series

# Set seed so thet we generate the same dataset
set.seed(2017)
# time variable
t=seq(0,1,1/100)
# generate a time series
some.time.series=2+3*t+ rnorm(length(t))

# obtain acv for this time series below
(acf(some.time.series,type = "covariance"))
```

```
##
## Autocovariances of series 'some.time.series', by lag
##
##     0     1     2     3     4     5     6     7     8     9    10    11    12
## 1.727 0.450 0.614 0.547 0.640 0.403 0.297 0.617 0.480 0.434 0.550 0.679 0.473
##    13    14    15    16    17    18    19    20
## 0.538 0.348 0.543 0.159 0.321 0.467 0.217 0.301
```

```
# Simulating a non-stationary time series

# Set seed so thet we generate the same dataset
set.seed(2017)
```

```r
# time variable
t=seq(0,1,1/100)
# generate a time series
some.time.series=2+3*t+ rnorm(length(t))
# obtain acf of the time series below
(acf(some.time.series,type ="correlation"))
```






```
##
## Autocorrelations of series 'some.time.series', by lag
##
##     0     1     2     3     4     5     6     7     8     9    10    11    12
## 1.000 0.260 0.355 0.317 0.371 0.233 0.172 0.357 0.278 0.251 0.318 0.393 0.274
##    13    14    15    16    17    18    19    20
## 0.312 0.202 0.314 0.092 0.186 0.270 0.126 0.175
```

```r
# Simulating MA(4) process.
# X_t= Z_t+0.2 Z_(t-1)+0.3 Z_(t-2)+ 0.4 Z_(t-3)

set.seed(2^10)
z=NULL
z=rnorm(1000)
data=NULL
for(i in 4:1000){
  data[i-3]=z[i]+0.2*z[i-1]+0.3*z[i-2]+0.4*z[i-3]
}
```

```r
data=ts(data)

# find acf below
(acf(data,type = "correlation"))
```

```
##
## Autocorrelations of series 'data', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.298  0.314  0.302  0.014  0.017  0.031 -0.001  0.028  0.027  0.000
##     11     12     13     14     15     16     17     18     19     20     21
##  0.043 -0.006 -0.048 -0.017 -0.060 -0.052 -0.031  0.038  0.008  0.067  0.070
##     22     23     24     25     26     27     28     29
##  0.046  0.056  0.020  0.005  0.012 -0.002  0.017  0.023
```