Discrete Response, Time Series and Panel Data - Exercises

Nils Rechberger

2026-03-05

Series 1

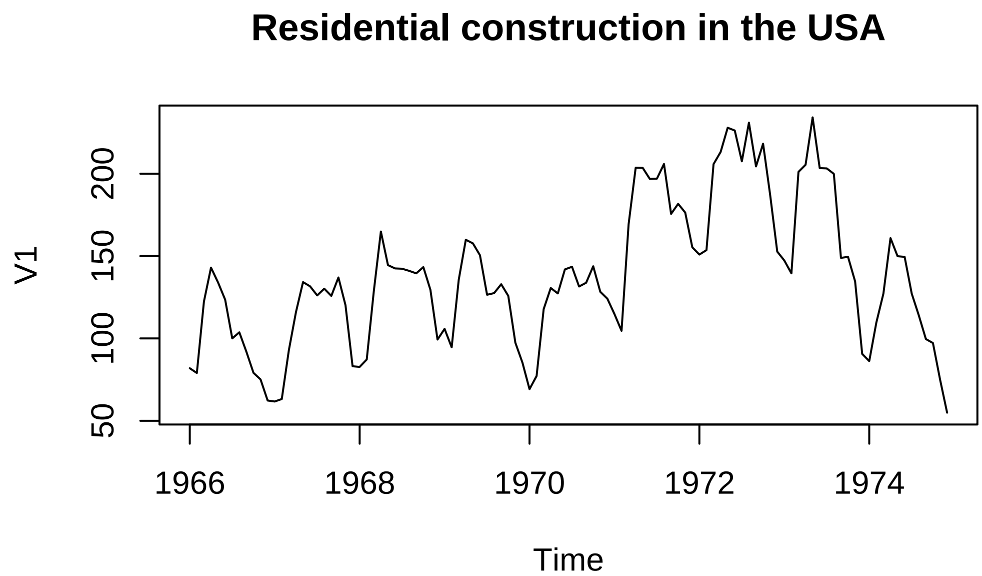
Exercise 1.1

What is the expected period (time period of repetition) and the time step for the following timeseries:

- Sunshine duration per month in Basel from 1990 to 2000: Timestep = 1 month, Period = 12 months.
- Number of newborn babies in the city of Zurich per year from 2000 to 2011: Timestep = 1 year, Periode = None.
- Number of reservations in a restaurant for every night during 4 weeks: Timestep = 1 day, Periode = 7 days.
- Water runoff of a river. The data has been collected every day for 4 years: Time step = 1, periode = 365 days.

Exercise 1.2

```r
hstart <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/hstart.dat")

ts <- ts(hstart,start=c(1966,1),freq=12)

plot(ts, main="Residential construction in the USA")
```



This is not a stationary time series. A stationary time series is one whose statistical properties (mean, variance, etc) are constant over time. This data fails that definition due to two distinct types of non-stationarity.

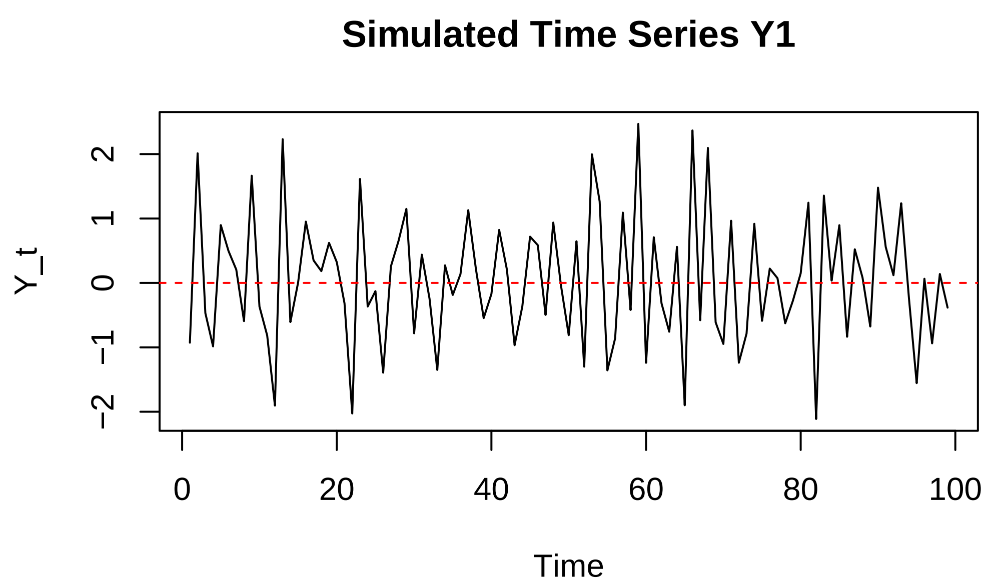
- Trend: The long-term "level" of the data shifts upward and then downward.
- Seasonality: There are recurring annual cycles.
- Changing Variance: The intensity of the fluctuations changes over the years.

Exercises 1.3

A

```
set.seed(1)

Et <- ts(rnorm(101, 0, 1))

Et[1] <- 0
y1 <- 0 #delete later

for (i in 2:length(Et)) {
    y1[i-1] <- Et[i] - 0.5 * Et[i-1]
}
```

```r
y1 = y1[2:length(y1)]
ts.y1 = ts(y1)

plot(ts.y1, main="Simulated Time Series Y1", ylab="Y_t", xlab="Time")
abline(h=0, col="red", lty=2)
```

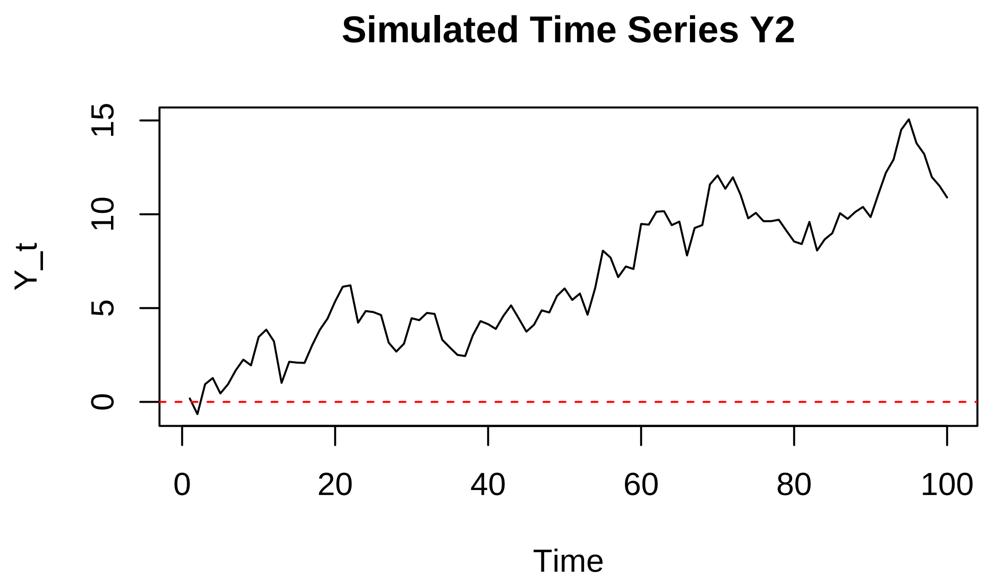


The time series seems to be stationary.

B

```r
set.seed(1)

Et <- ts(rnorm(101, 0, 1))

Et[1] <- 0
y1 <- 0 #delete later

for (i in 2:length(Et)) {
    y1[i] <- y1[i-1] + Et[i]
}
```

```r
y1 = y1[2:length(y1)]
ts.y1 = ts(y1)

plot(ts.y1, main="Simulated Time Series Y2", ylab="Y_t", xlab="Time")
abline(h=0, col="red", lty=2)
```

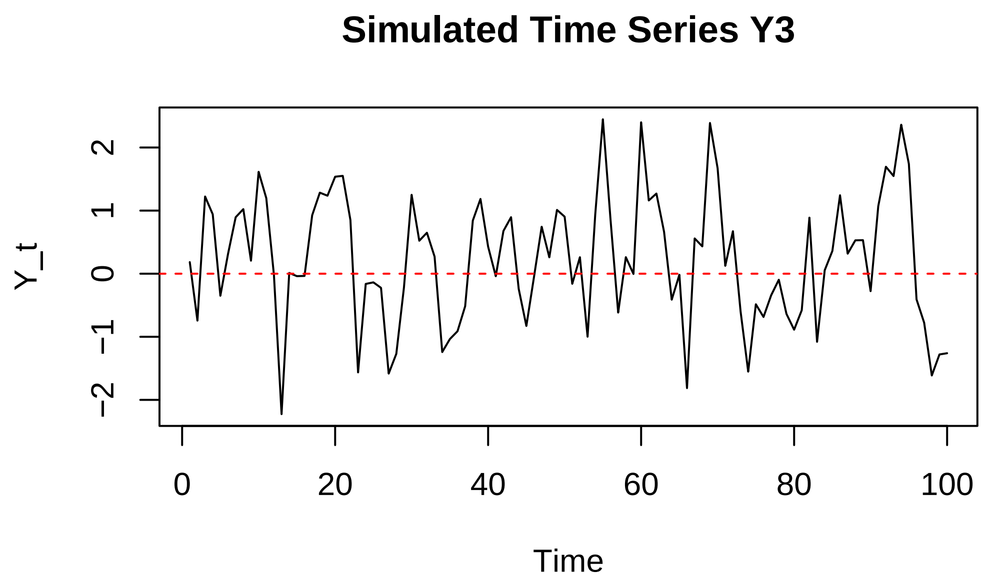


The time series seems not to be stationary.

C

```r
set.seed(1)

Et <- ts(rnorm(101, 0, 1))

Et[1] <- 0
y1 <- 0 #delete later

for (i in 2:length(Et)) {
    y1[i] <- 0.5 * y1[i-1] + Et[i]
}
```

```r
y1 = y1[2:length(y1)]
ts.y1 = ts(y1)

plot(ts.y1, main="Simulated Time Series Y3", ylab="Y_t", xlab="Time")
abline(h=0, col="red", lty=2)
```

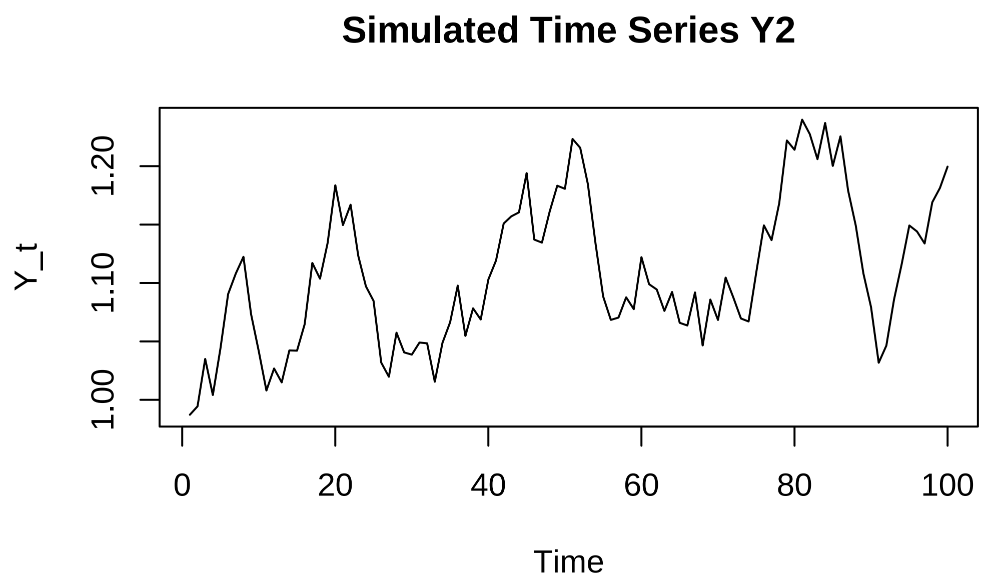


The time series seems to be stationary.

D

```r
set.seed(1)

Et <- ts(runif(101, 0.95, 1.05))

Et[1] <- 0
y1 <- 1 #delete later

for (i in 2:length(Et)) {
    y1[i] <- y1[i-1] * Et[i]
}
```

```
y1 = y1[2:length(y1)]
ts.y1 = ts(y1)

plot(ts.y1, main="Simulated Time Series Y2", ylab="Y_t", xlab="Time")
abline(h=0, col="red", lty=2)
```



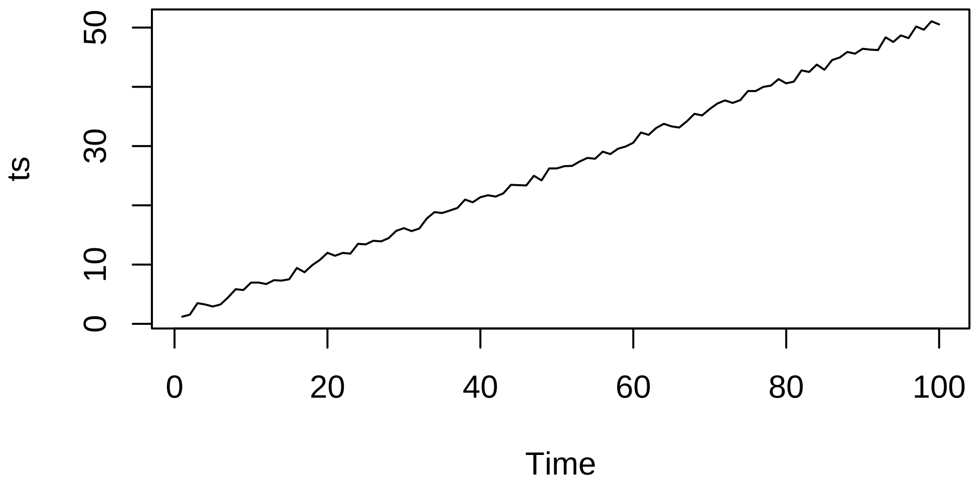
The time series seems not to be stationary.

Series 2

Exercise 2.1

A

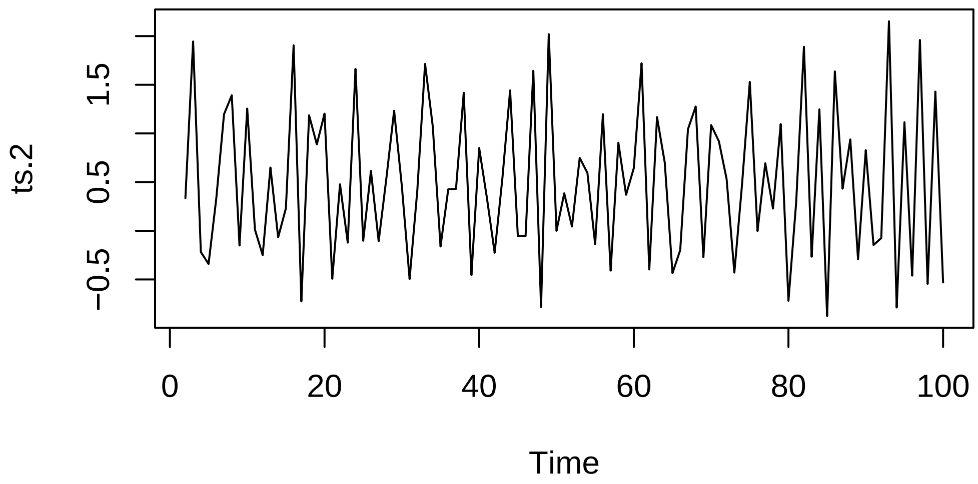
```
# create timeseries object
t <- seq(1, 100, length = 100)
data <- 0.5 * t + 1 + runif(100, -1, 1)

ts <- ts(data)
plot(ts)
```




```
length(ts)
```

```
[1] 100
```

```
# Applying backward differencing
ts.2 <- diff(ts, lag = 1)
plot(ts.2)
```



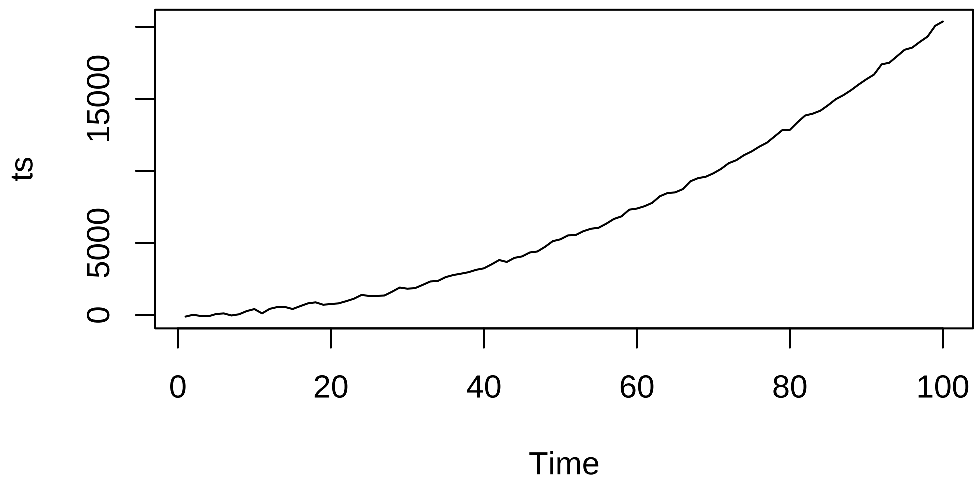

```
length(ts.2)
```

```
[1] 99
```

The new time series are 99 units long instead of 100.

B

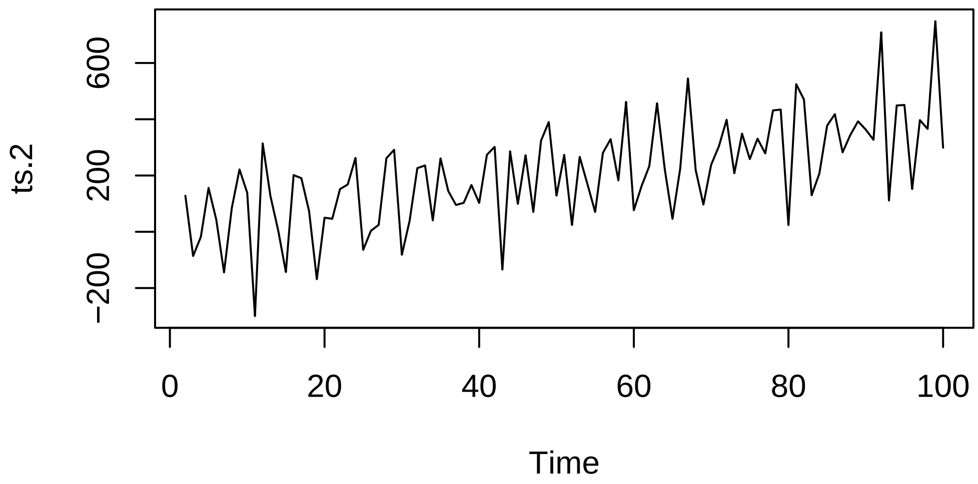
```
# create timeseries object
t <- seq(1, 100, length = 100)
data <- 2 * t **2 + 3 * t - 1 + runif(100, -200, 200)

ts <- ts(data)
plot(ts)
```




```
length(ts)
```

```
[1] 100
```

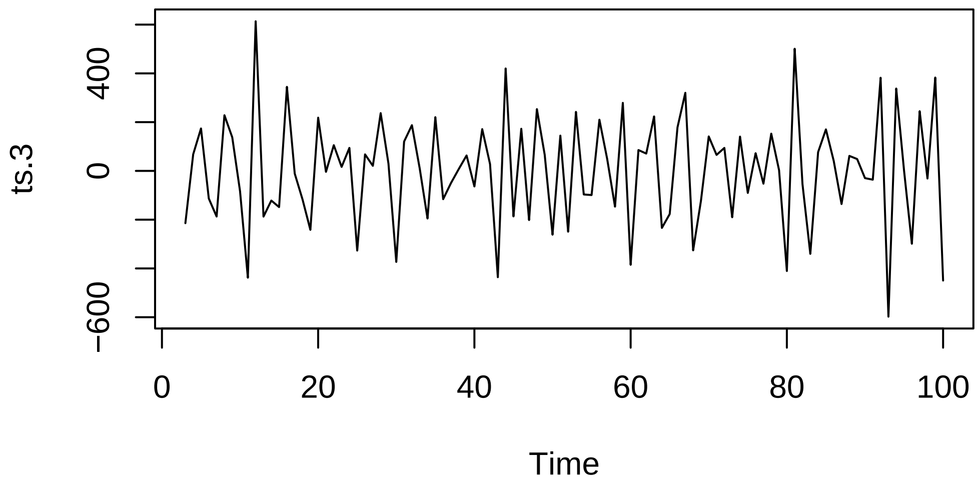
```
# Applying backward differencing
ts.2 <- diff(ts, lag = 1)
plot(ts.2)
```




```r
length(ts.2)
```

```
[1] 99
```

```r
# Again Applying backward differencing
ts.3 <- diff(ts.2, lag = 1)
plot(ts.3)
```




```
length(ts.3)
```

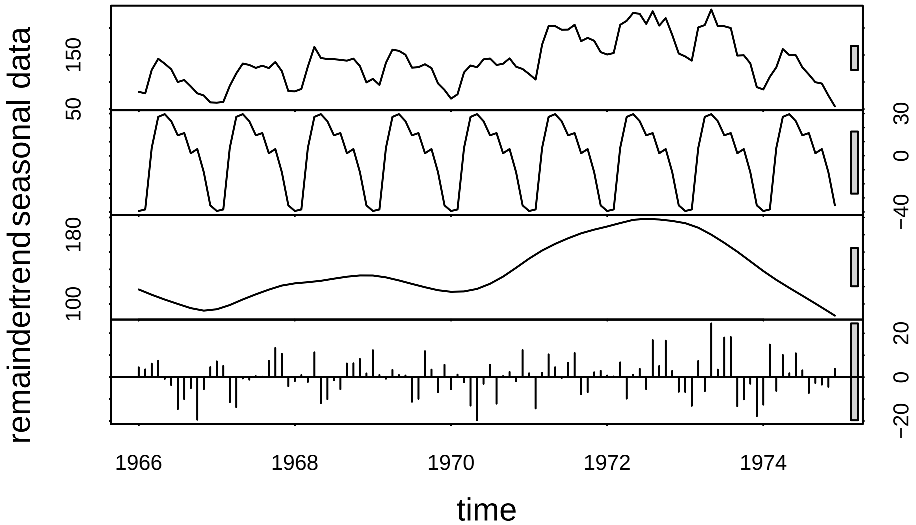
```
[1] 98
```

Since we applied backward differencing twice, the length of the output time series increased to 98.

Exercise 2.2

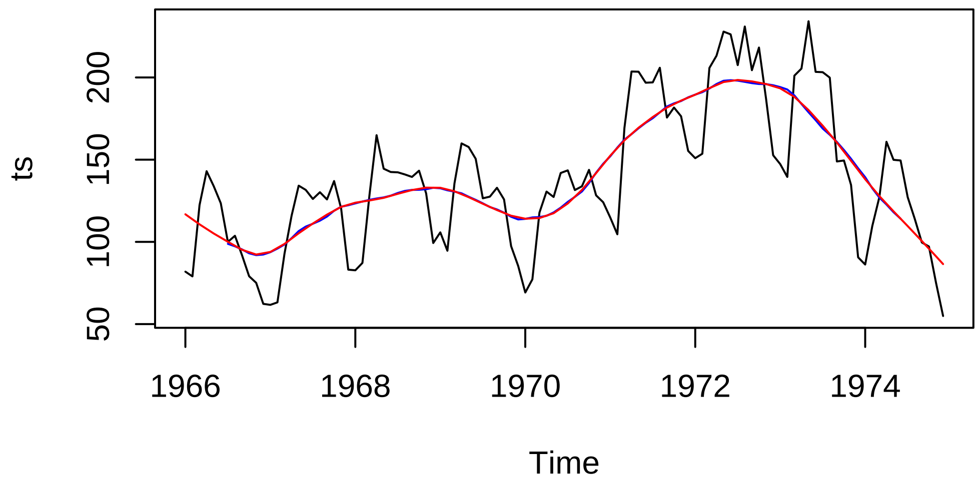
A

```
hstart <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/hstart.dat")

ts <- ts(hstart[, 1], start = c(1966, 1), frequency = 12)
hstart_stl <- stl(ts, s.window = "periodic")

plot(hstart_stl)
```



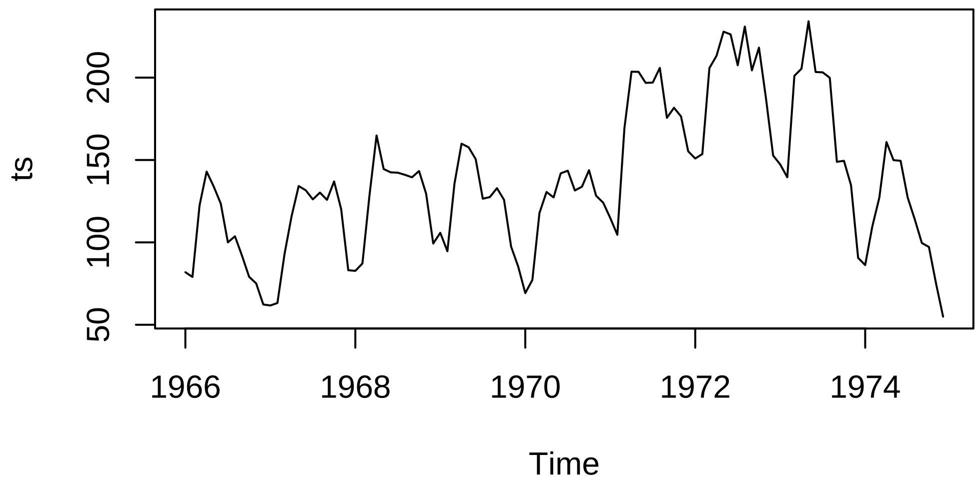
B

```
hstart <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/hstart.dat")

ts <- ts(hstart[, 1], start = c(1966, 1), frequency = 12)

weights <- c(1, rep(2, 11), 1) / 24
filtered <- filter(ts, filter = weights)

plot(ts)
points(filtered, type = "l", col = "blue")
lines(hstart_stl$time.series[, "trend"], col = "red")
```

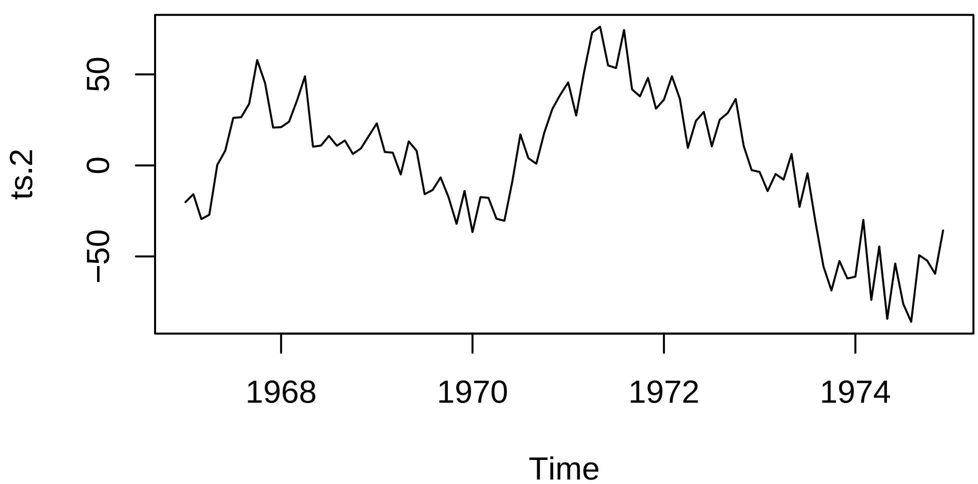


c

```r
hstart <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/hstart.dat")

ts <- ts(hstart[, 1], start = c(1966, 1), frequency = 12)
plot(ts)
```

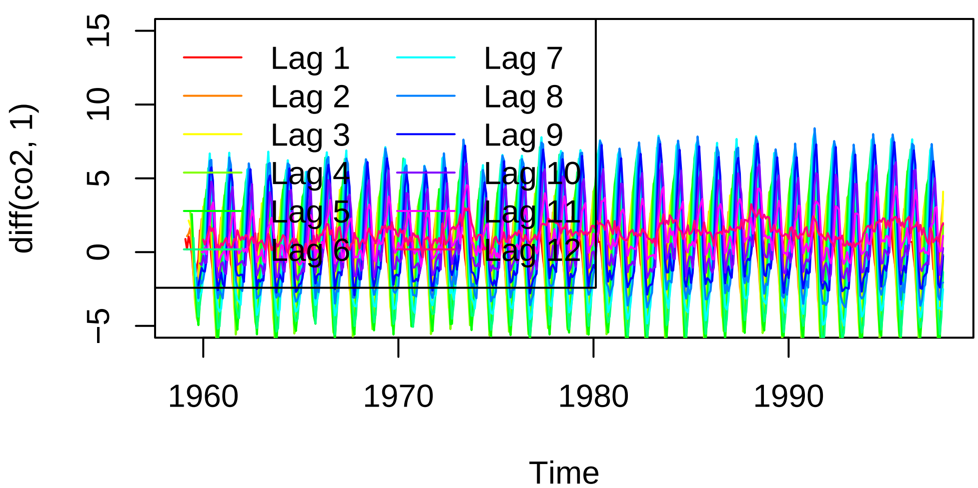



```
ts.2 <- diff(ts, lag=12) # Using lag 12 bc 12 months per year
plot(ts.2)
```



Exercise 2.3

```
data(co2)
plot(diff(co2, 1), type = "n", ylim = c(-5, 15))

for (l in 1:12) {
    ts_diff <- diff(co2, lag = l)
    lines(ts_diff, col = rainbow(12)[l])
}
legend("topleft", legend = paste("Lag", 1:12), col = rainbow(12), lty = 1, ncol = 2)
```



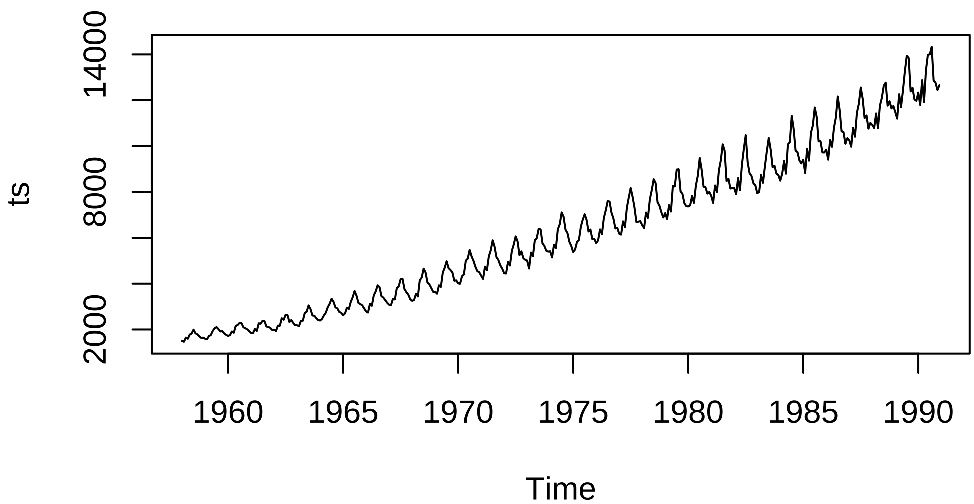
Series 3

Exercise 3.1

- A -> 2
- B -> 3
- C -> 1
- D -> 4

Exercise 3.2

```r
cbe <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/cbe.dat", header = TRUE)
ts <- ts(cbe$elec,  start = c(1958, 1), frequency = 12)
plot(ts)
```

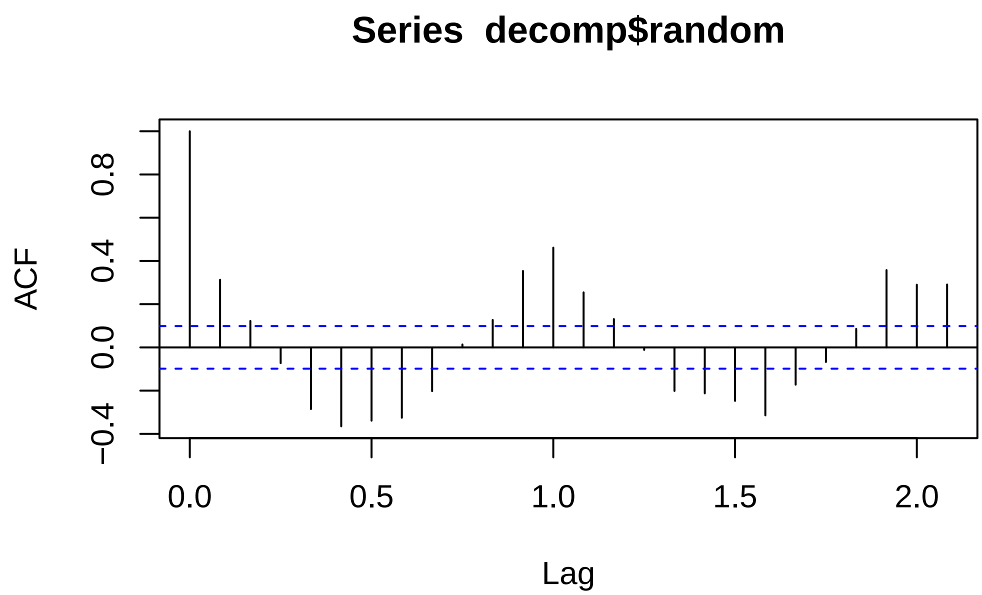


A

At this time, it is not advisable to examine the ACF because the time series is not stationary. We can see trends and seasonality over time. We need to clean that up first before analysing the time series.

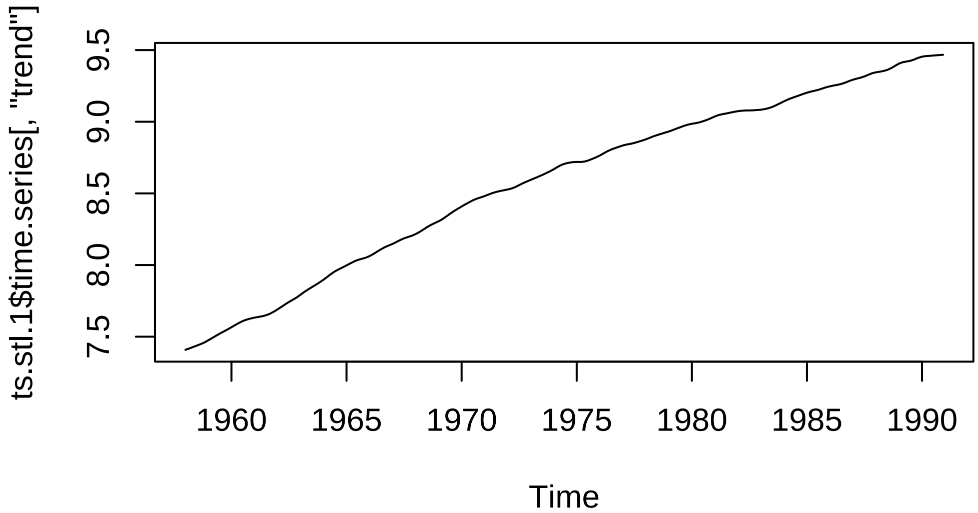
B

```r
decomp <- decompose(ts, type = "multiplicative")
acf(decomp$random, na.action = na.pass)
```

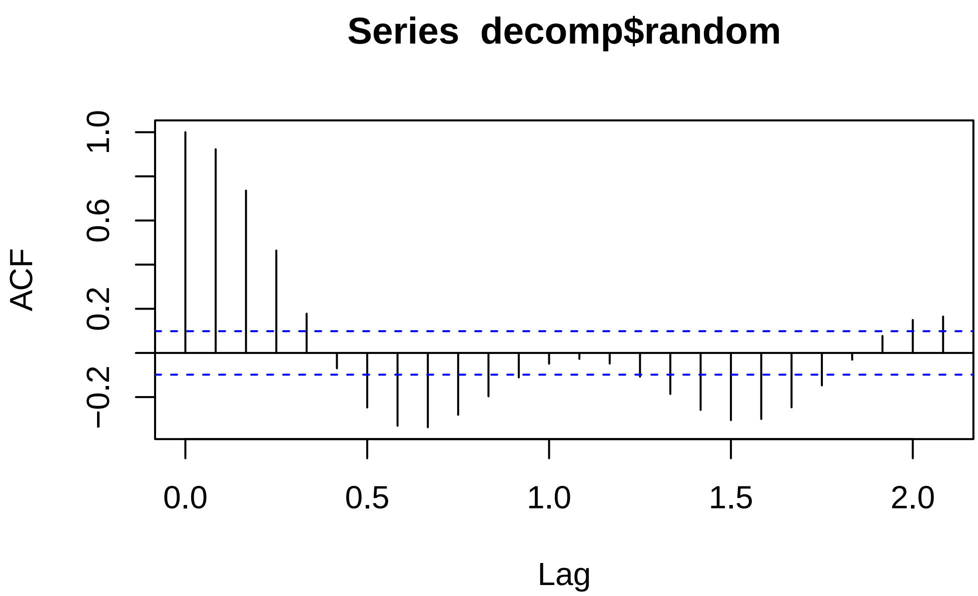


c

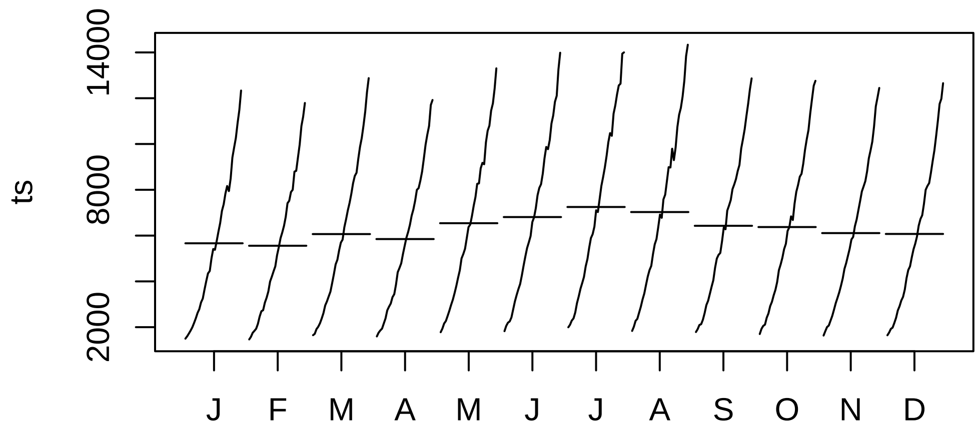
```
ts.stl.1 <- stl(log(ts), s.window = "periodic")
plot(ts.stl.1$time.series[, "trend"])
```



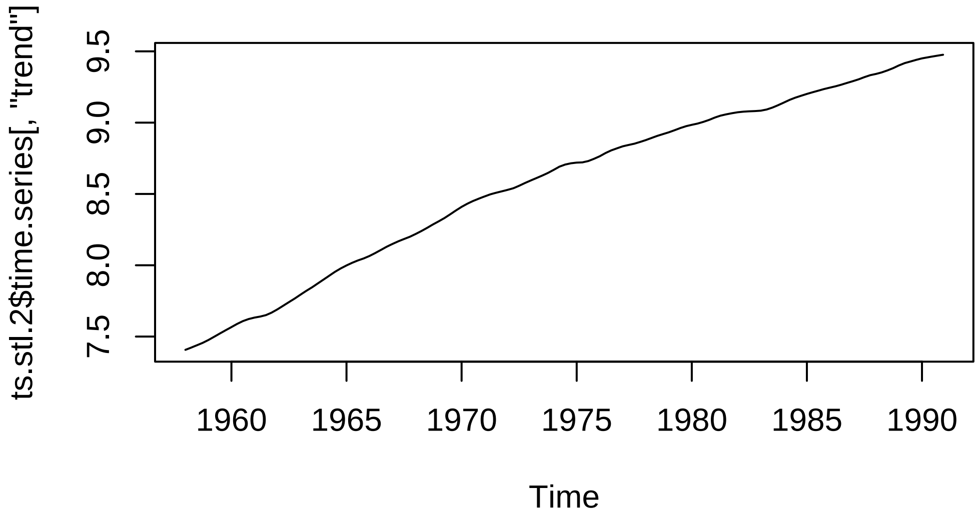

```r
decomp <- decompose(ts.stl.1$time.series[, "trend"], type = "multiplicative")
acf(decomp$random, na.action = na.pass)
```



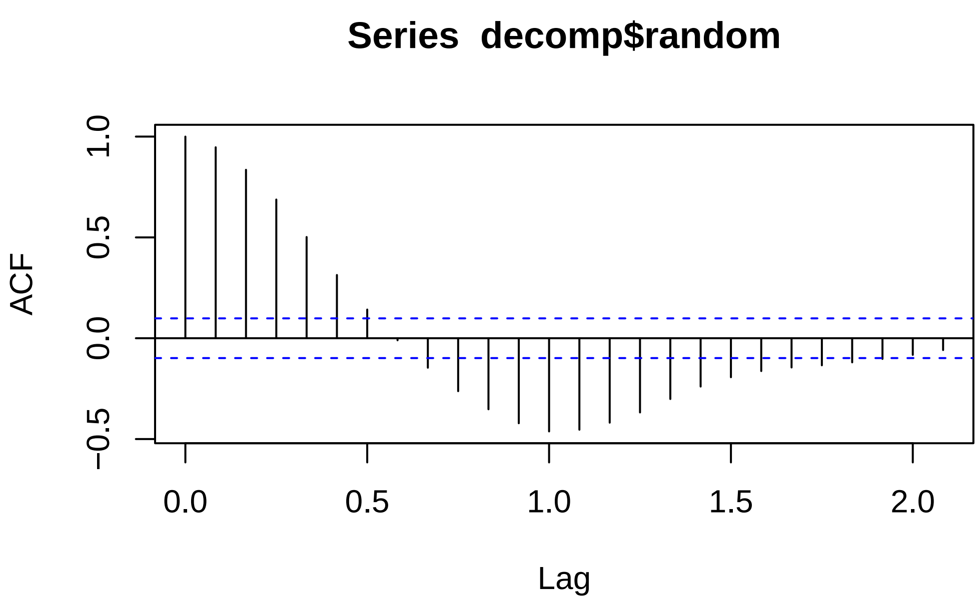
```r
monthplot(ts)
```




```r
ts.stl.2 <- stl(log(ts), s.window = 12)
plot(ts.stl.2$time.series[, "trend"])
```




```r
decomp <- decompose(ts.stl.2$time.series[, "trend"], type = "multiplicative")
acf(decomp$random, na.action = na.pass)
```

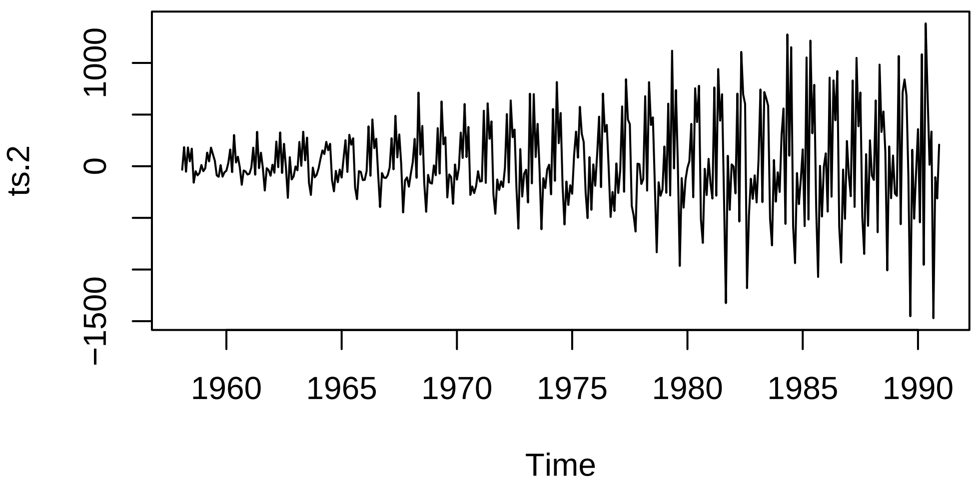


D

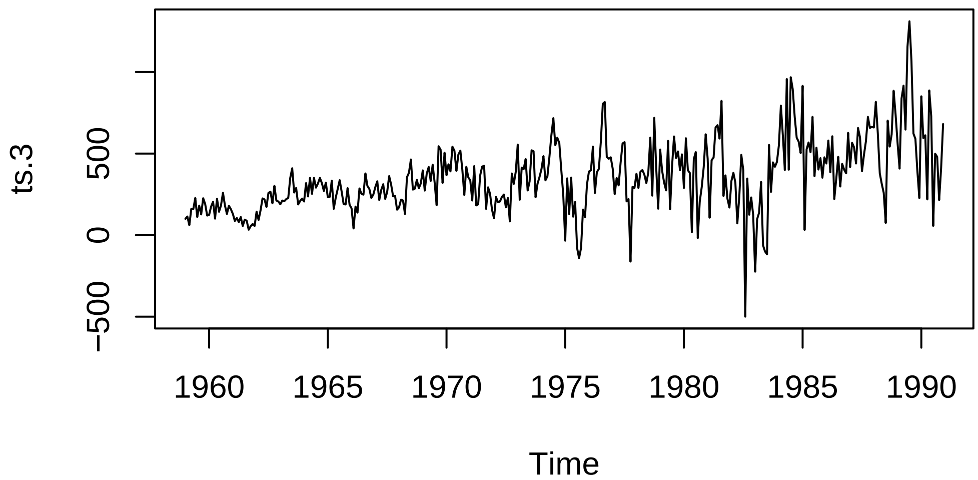
1. With `type = "multiplicative"` we assumes that seasonality is proportional to the trend.
2. We used `stl(log(ts), ...)` because the original time series had an exponential trend.

E

```r
cbe <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/cbe.dat", header = TRUE)
ts <- ts(cbe$elec,  start = c(1958, 1), frequency = 12)

ts.2 <- diff(ts, lag=1)
plot(ts.2)
```




```r
cbe <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/cbe.dat", header = TRUE)
ts <- ts(cbe$elec,  start = c(1958, 1), frequency = 12)

ts.3 <- diff(ts, lag=12)
plot(ts.3)
```



Using differencing alone, we eliminate the trend but do not obtain a fully stationary time series.

Series 4

Exercise 4.1

A

```
arima.sim(list(ar = c(0.8)), n = 1000)
```

```
Time Series:
Start = 1
End = 1000
Frequency = 1
   [1] -2.896758444 -2.387824138 -2.341138840 -2.465136445 -0.990992996
   [6] -0.260385040 -0.298764156 -0.082520833 -0.803328358 -0.844003892
  [11]  0.426973482  0.324830529  0.421653057  2.362083836  1.185972815
  [16]  1.909570636  3.318141562  1.590348086  1.289915015  0.642023383
  [21]  0.022785954 -1.027488889 -1.718202375 -0.105174736  0.509701160
```

```
 [26]  1.183395247  2.504086574  1.637867462  2.126850418  1.640845557
 [31]  0.811298127  1.575101227  1.297018673 -0.028585235 -0.261324542
 [36]  1.286163811  2.201089595  0.303164466  0.337587800  1.117735204
 [41] -0.730176367  0.824422263  0.117777449  0.372886683  0.104336602
 [46]  1.659627463 -0.147845665 -0.262884739 -1.163510937 -0.524266017
 [51]  1.809849388 -0.066617498 -0.115001421 -0.239271926  1.350175528
 [56]  0.098284753  0.575205975  2.157112661  1.464953820  0.466034471
 [61]  0.211649070  0.670641084 -0.477026803  1.233130793  0.992146619
 [66] -2.111181765 -2.796110231 -0.689321252 -1.528287352 -1.324133329
 [71] -1.016656414 -2.410043145 -1.437067144 -0.728050349  1.291463619
 [76]  2.067685219  1.735958486  1.306243027  1.651067852  0.433434136
 [81]  0.452168699  0.714609433  1.122080905 -0.236666245  1.273018543
 [86]  1.720531545  3.883536384  1.216801964  0.383628781 -1.407599272
 [91] -1.547077315 -0.927520476  0.960554205  0.325058560 -0.938550235
 [96] -1.058221103 -0.225522679  0.001484048  1.319588170  0.756761222
[101] -1.042812764  0.117248258 -1.019324347 -0.198493002  0.354699315
[106]  0.653218556  2.246468975  1.591030614 -0.041370649  0.030377577
[111] -0.207675390  0.468920016  2.009780298 -0.200503341 -0.374288624
[116] -0.229064758  0.366455863 -0.403658855  0.067638858  0.435522346
[121]  0.336045107  0.144401099  1.582265472  1.939741054  3.508218107
[126]  2.537533473  0.785475260  0.232677286  0.283538483 -0.011556164
[131] -0.421072887 -1.914076359 -2.328537188 -2.959066533 -2.058944478
[136] -1.302360459  0.497759720  0.068693584  1.003344218  0.323419788
[141] -1.256150965 -0.570384116 -0.975843960 -1.615234201 -2.048834970
[146] -0.549564481  1.132781333  1.913589432  1.257713552 -0.303508700
[151] -0.020233129  1.097246681  1.715191742  1.686670162  1.571554202
[156]  0.413627971  0.774707703  0.675563273  0.608423008  0.284776664
[161] -0.930193923 -1.336898409 -0.303453343 -0.203834558 -0.148420834
[166] -0.305053635  1.156547919  0.943723906  1.004175136  0.952560845
[171] -0.201184505 -0.227414378  1.104990539  1.342117690 -0.378334138
[176] -0.225325033  0.379635246  0.228761627  0.965706999  0.599897036
[181] -0.571376150  0.272350359  0.480545530  0.928094283  1.783535725
[186]  1.624334733 -0.330110501 -0.143048168 -1.751860487 -1.932531495
[191] -0.592345404 -2.194526985 -1.649300969 -1.928109271 -1.843684400
[196] -0.498744995  0.057012673  1.340018206 -0.061187641 -0.918410464
[201] -1.489698663 -1.321394281 -2.058916760 -2.467001687 -2.948153819
[206] -1.754213646 -0.854583329  0.232765994  2.847779162  2.097966264
[211]  2.363387777  5.157124741  4.686300252  3.680022904  1.971575387
[216]  1.030673724 -0.864153349 -2.263695378 -2.215943461 -1.453468351
[221] -1.122346999 -1.287887162 -2.849531960 -1.620444857 -0.836734213
[226]  0.947238966 -1.098399322 -1.165543341  0.817887216  0.770723386
[231]  2.000831867  2.174886406  1.876399937  2.415335938  0.131442433
[236] -0.234726693  0.418483218  1.675916885  2.108020796  1.880142304
```

```
[241]  2.644680533  2.129609231  0.598381475  0.453542541  0.199160695
[246]  0.529388304  0.042686107  0.687101255  2.611022810  0.292173311
[251]  0.817815773 -0.068500505 -0.683965069 -2.363378110 -2.149991586
[256] -1.385360996 -2.535456409 -0.089736894 -0.831319729 -2.943831924
[261] -2.469116306  0.376563621  1.897571700  2.795502871  3.025373155
[266]  2.881763906  1.867341991 -0.013933416 -2.234092864 -2.965946734
[271] -4.155517611 -4.312333228 -2.721377894 -3.061787230 -3.987871201
[276] -4.234672368 -5.105874085 -3.280941728 -4.126540553 -3.446731513
[281] -2.177926598 -0.540815703  1.461258224 -0.591219764  0.451570992
[286] -0.195285375 -0.336812438  1.177990335  0.335260815  0.947571082
[291]  0.664499226  0.041513089  1.443869854  0.930522094  0.531922185
[296]  1.121916222  1.812715483  0.526798070  1.568311719  0.618784332
[301] -0.391415843 -2.646269359 -2.262506292 -1.493041010 -1.901902795
[306] -0.279366678  0.396658853  0.417230150  2.141487607  0.210760107
[311]  0.454248551  1.209105805 -0.028057994 -0.279300519 -0.279296459
[316] -0.668442407 -0.465058823 -0.526718764 -1.252638471 -0.240566424
[321] -0.768960072 -1.241536286 -0.511893703  1.285756122 -0.732621397
[326] -0.388084102  0.086881817  0.098730949  2.639258148  3.368534230
[331]  2.160289698  1.103004330  1.796252151  2.444201255  2.674652828
[336]  1.535010601  1.767062887  1.336819424  2.919375100  1.480592529
[341]  1.217111318 -0.051370427 -1.023345417 -0.814574377 -0.885086679
[346] -1.206957562  0.584146913  0.554814447  1.762552688  0.428818031
[351]  0.097431837 -1.325988367  0.380102454 -0.677278028  0.932422482
[356] -0.245259260 -0.290704753 -3.107705486 -2.733030490 -2.171679906
[361] -3.656431628 -3.212959046 -2.917004683 -4.173192331 -2.439964924
[366] -3.164826950 -2.750825792 -1.636392473 -1.834548354 -0.723264458
[371] -0.449629813  1.128570406  0.240174374 -0.968515502 -0.416038167
[376] -0.527676918 -0.717423543 -0.077298410  0.423074060  0.357243748
[381]  0.920569563  1.490899738  2.026308826  2.586808357  3.363326653
[386]  2.554110302  1.603149553  0.055235728 -0.193464486 -1.081629745
[391] -0.454068440 -0.562119329 -1.007145103 -1.782873205 -1.365563315
[396] -1.689680132 -2.610692803 -3.498601444 -1.697577076 -2.035303687
[401] -2.390416457 -2.204081768 -2.338453943 -2.314705023 -2.164334462
[406] -2.334471996 -2.961512317 -1.654503662 -1.432415193 -2.589730090
[411] -1.265660808 -2.752363725 -2.603211289 -2.370151520 -2.834528616
[416] -1.979955754 -3.089365728 -0.952195569 -0.394347099  1.384384730
[421]  1.751704761 -0.286437220  0.418496218  0.783591202  1.653175119
[426]  2.397518318  2.376324268  2.532646172  1.445650539  2.740712574
[431]  0.428577119 -1.537760275 -2.521927264 -1.107871365 -1.994052774
[436] -1.979366094 -1.500758041 -1.684488907 -3.432332251 -1.578278837
[441] -1.339448840 -0.541137677 -0.428001346 -0.872632506  0.816760649
[446]  1.468868155 -0.331427339 -1.422322043  0.163697303 -0.774682933
[451] -0.604182865 -1.911351489 -0.841101489 -0.078684690 -0.333262996
```

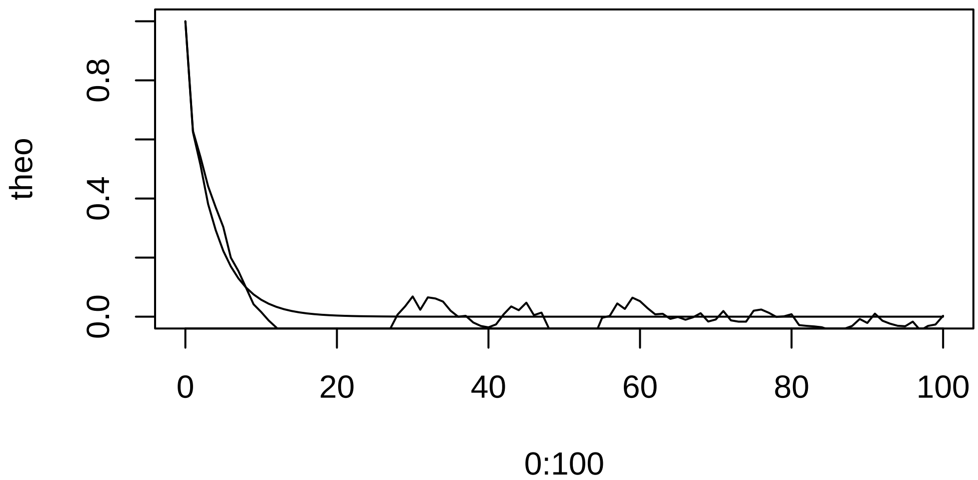
```
[456]  1.287465679  0.519230085  0.123541332  1.200271231  1.418882200
[461]  1.987013799  3.055810958  2.688512061  2.800231583  1.599067110
[466]  1.743713459  1.085770320  2.144636965  1.668098352  1.364579848
[471]  1.088083608 -1.164105757 -1.924104568  0.424574402  1.046188294
[476]  0.876625615  0.542781903  0.271967971  0.656396288  1.305816323
[481]  0.063447436 -0.090827783 -1.328911179 -1.492162619 -0.684953829
[486] -1.994852806 -0.576369416  0.717451443  0.563702389  0.719586782
[491]  1.917698298  0.950552196 -0.176559134 -1.293404363 -2.009906922
[496] -2.319339136 -1.483558832 -2.130361483 -1.978656495 -1.428456702
[501] -2.431418563 -0.526032531  0.902874574 -1.082676114 -1.350210418
[506] -1.453380114 -0.993618189 -1.967878154 -1.950499260 -0.777271769
[511]  0.369353889  0.524767586  0.756665018 -0.041899985 -0.476726670
[516]  0.780370425  0.689347350 -0.130989700  1.529549898  0.638790674
[521]  1.640968540  1.893975875  1.894473767  1.204838138  1.850260512
[526] -0.161656341 -1.117888814 -1.138314481 -0.754594659 -0.501623769
[531] -0.688757187 -0.844200636 -0.205883451 -0.827295273 -2.852208884
[536] -2.277912435 -0.959387119  0.212713664 -0.121553840 -0.158256548
[541]  1.385263163  0.464601465  0.240730693  0.677796008 -0.102945628
[546] -0.546312722  0.130199519 -0.619248219 -0.039514021 -1.229545843
[551] -3.001817225 -1.781730375 -1.558241766 -1.681554152 -1.866997501
[556] -0.501309575 -1.486470068 -0.229343637 -1.222485451 -1.115830328
[561] -1.107372680 -0.312179559 -2.026192962 -1.877346665 -1.640427968
[566] -0.754657400  0.518620369 -0.567250912 -0.126895174  0.346310469
[571]  1.460772496  1.080001080  2.010998744  1.695611569  1.061484221
[576]  1.361520315  1.372529887  1.482567626  2.190858440  1.466720960
[581]  0.936036878  0.545155135  1.508596114  3.157154676  3.490177808
[586]  3.830546835  4.833311092  3.294998726  1.165716091 -0.177793572
[591]  0.112168532  0.113079506 -2.625461689 -0.915080100 -0.951819278
[596] -0.818128183 -0.247938037 -1.507780379 -1.912915590 -0.496440802
[601]  1.575608326  1.948011734  2.296651937  0.129051318  1.155860845
[606]  2.047927785  1.283750222  1.215239031  1.895991252  3.669493845
[611]  1.825855733  2.490192809  3.369308997  3.610258249  3.181830031
[616]  2.393725400  2.079871760  2.532474838  0.947945377 -0.463975666
[621] -1.082625229 -2.290131850 -3.501449892 -1.421923781 -2.057213608
[626] -2.150260930 -2.854940557 -1.771179391 -0.888954730  0.477083540
[631] -0.080369525  1.176773668  1.078440486 -0.436892850  0.801894662
[636]  2.154963208  2.526108442  1.590349812 -0.072421091 -0.020494802
[641] -1.362117631 -2.903698421 -1.618925205 -1.594159828 -1.674219158
[646] -1.982692657 -0.270549071 -1.801407138 -1.885622135 -0.942314630
[651] -0.463526030 -0.986365156 -0.851099944 -0.873490210  0.984361781
[656]  0.890063271  0.057215303  1.232347442 -0.286491718  0.308871206
[661]  1.477947356  0.016650921  0.337530185  1.220891523  2.094146821
[666]  3.199286503  4.477477612  3.782159395  2.178537022  0.985962077
```

```
[671]  1.483874278 -0.385818452  0.191579038 -1.001655429 -1.726525571
[676] -1.039315898 -1.421080681 -1.965302926 -2.235972551 -0.372337188
[681] -1.669372093 -1.226636951 -1.448131813  1.803237913  1.284763150
[686]  2.847235793  3.141531186  0.946781729  1.272881352  1.714524594
[691]  1.397040775 -0.834150681  1.578038192  1.074693378 -0.541496487
[696]  0.815161316  0.405050822  0.569600173 -0.323043419 -2.049320916
[701] -1.806980589 -0.694604534 -0.572707031 -0.430586650 -0.122323173
[706]  0.300773286 -0.020899784 -0.319897451 -0.989489746 -2.173446510
[711] -3.180444581 -3.330665829 -3.407014903 -3.116592169 -3.709028798
[716] -3.857628630 -1.587830273 -0.898425107 -0.457634002 -0.390651059
[721] -1.230210091 -1.576052290 -1.631834888 -1.217543653 -1.008761267
[726]  0.999365255  0.459256139 -0.381558344 -1.944383514 -2.578097750
[731]  0.533293638  0.729833725  1.492614368  1.401941443  1.299567156
[736]  0.873888727  1.256214601  2.449306084  2.860801928  2.066606498
[741]  1.759476514 -0.041639622  1.105240026  2.698016731  0.647374653
[746]  0.499812863 -0.480337808 -1.581747346 -0.196638919  1.009425724
[751]  2.837499075  2.770172001  0.393387822  0.803871318 -0.050792342
[756] -0.239698379  0.050195373  0.110508802  0.882553108  0.050663790
[761] -0.636934065 -0.325728738  0.617125277  0.101469309 -0.587772858
[766] -1.480351530 -1.935111819 -1.831642452 -2.631116943 -1.413668569
[771] -3.187472486 -2.704152492 -3.608216304 -1.670422179 -1.847508880
[776] -0.349468347 -0.699217946 -0.772114091 -1.638352053 -4.393045260
[781] -1.868292845 -2.905848282  0.240459005  0.517960918 -0.151169084
[786]  0.577963409  1.343085824  0.846713689  0.813671700  2.494345593
[791]  0.590084756  0.422984735  1.132542294 -0.507989680 -2.196319107
[796] -0.402657953 -1.069893068 -1.893366932 -1.550865682 -1.795043152
[801] -1.994901273 -3.015794401 -3.320595375 -1.808843206 -0.947611465
[806] -1.395914690  1.298773396  2.519926842  1.276959715  0.923802639
[811] -2.497343621 -1.040811239 -0.273314816 -0.716352572 -0.767062367
[816] -0.527314959  0.595655571 -0.493393566  0.671637549  1.292819194
[821]  0.331003061 -0.022049116  1.823467600  1.302009770 -0.348194819
[826] -1.751659844 -1.470846809 -0.937436018 -0.499528904 -0.664047073
[831] -2.506637614 -2.455186097 -1.037683223 -3.150117689 -1.906160601
[836] -2.998068612 -2.617504075 -3.497419553 -1.977307234 -2.174572064
[841] -1.317688858 -1.854303195 -2.666643700 -3.642872250 -4.070240909
[846] -3.611047828 -1.503387077 -0.633049803 -1.144934925 -0.676143494
[851] -1.541689218 -1.848638008 -1.321180220  1.627116628  1.531160763
[856]  1.325183277 -1.242365895  0.182002147  1.393971357 -0.508068832
[861] -0.409111212 -1.164553254 -0.812789359 -1.151238785  0.104353697
[866]  0.301291171  0.326611592  0.649058302 -1.298409987 -2.213278142
[871] -0.404016000 -0.132624176  0.301043832 -1.164744005  1.240448964
[876]  0.452816395  0.849140888  0.417780710  1.257538157  0.095118561
[881]  1.059922392  1.851007856  0.690045937 -1.105996824 -0.483802707
```

```
[886] -2.735047048 -1.457560048 -0.410792835  0.459932112  0.025882243
[891]  1.844554687  1.327503269  0.090436718 -0.316789593 -1.517087840
[896] -2.491935778 -1.789161667 -1.384761585 -2.016053402 -1.831525953
[901] -1.620516183 -1.602669318  0.487153927 -0.203318854  0.526776165
[906]  0.081340745 -0.996437346 -1.736323081 -1.232145877 -1.557513907
[911] -1.087811637 -0.529558463  0.311717647 -0.775454012  0.742998931
[916] -1.320781578 -2.980459015 -1.248650252  0.242303413 -0.963662158
[921] -0.274307733 -1.551354316  0.351545167  0.343194215  0.128609511
[926] -0.240848170 -1.439849529 -0.308545663 -0.248967666  0.292706717
[931] -0.579868149 -0.894380530 -0.478950025 -2.137385863 -2.662692521
[936] -2.759008164 -1.644427638 -2.268816528 -2.909610631 -1.916171372
[941] -2.299119666 -0.832566692 -2.364952814 -2.275981694 -2.629558485
[946] -2.628536162 -1.387941740 -0.408324103 -1.095359411 -2.753411626
[951] -2.973251889 -1.044634986 -0.856001533 -1.566778499 -1.244694778
[956] -0.172929731  1.726946119  1.922867249  0.547906188  2.715401971
[961]  2.260410944  3.229364535  3.885289838  1.070315796  2.497114604
[966]  1.887765583  0.672253073  0.612473581  0.571297291  0.482729006
[971]  0.238282453  1.693469635  2.386868184  2.246849242  1.216045148
[976]  2.217831012  2.464917959  1.554852638  0.754066097  0.360971493
[981]  0.135698302  0.498953856  1.289344015  1.256563608 -0.674040753
[986] -0.179063076  0.966229274 -0.533231485 -2.314608513 -1.057983636
[991] -1.231025072 -0.469954066 -0.613348246  1.250831934  2.204843060
[996]  2.419849468  2.775205525  3.178832485  3.678865558  2.017329293
```

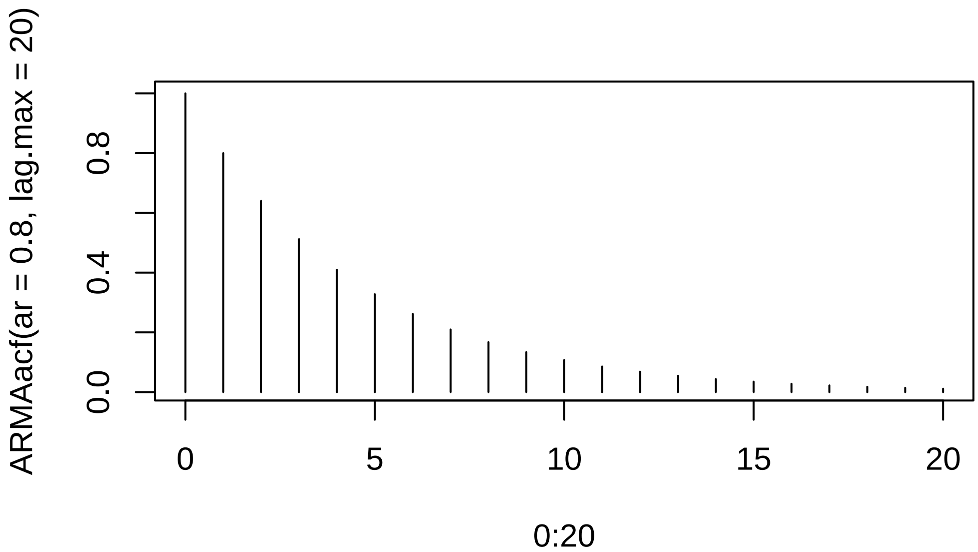
B

```r
theo <- ARMAacf(ar = c(0.5, 0.2), lag.max = 100)
sim_data <- arima.sim(model = list(ar = c(0.5, 0.2)), n = 500)
plugin <- acf(sim_data, lag.max = 100, plot = FALSE)$acf

plot(0:100, theo, type = "l")
lines(0:100, plugin)
```



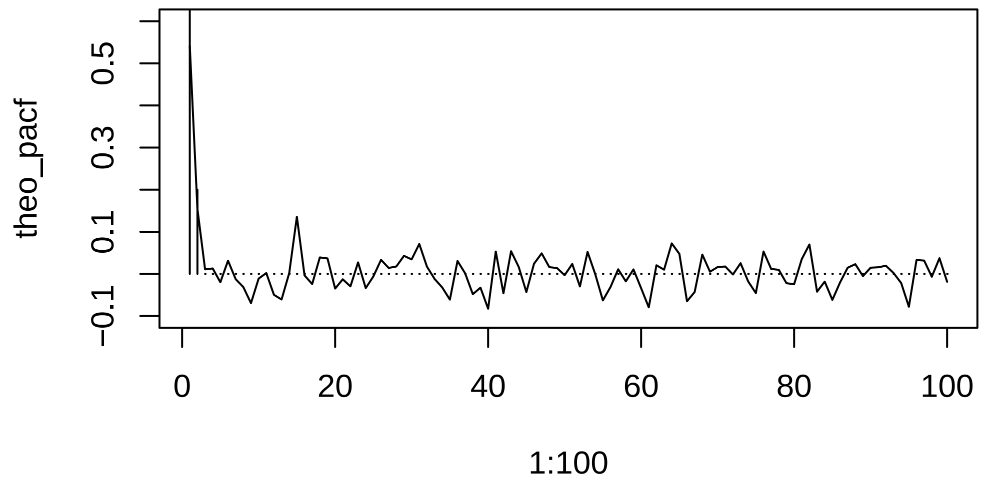
C

```
plot(0:20, ARMAacf(ar = 0.8, lag.max = 20), type = "h")
```



D

```
theo_pacf <- ARMAacf(ar = c(0.5, 0.2), lag.max = 100, pacf = TRUE)
sim_data <- arima.sim(model = list(ar = c(0.5, 0.2)), n = 500)
estim_pacf <- pacf(sim_data, lag.max = 100, plot = FALSE)$acf

plot(1:100, theo_pacf, type = "h", ylim = c(-0.1, 0.6))
lines(1:100, estim_pacf, type = "l")
```



Exercise 4.2

```
data <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/kreatin.dat", header=TRUE)
data
```

```
       n gehalt
1      1     70
2      2     70
3      3     70
4      4     69
5      5     71
6      6     68
7      7     70
8      8     65
9      9     69
10    10     70
11    11     70
12    12     68
13    13     65
14    14     75
15    15     72
```

16	16	75
17	17	70
18	18	70
19	19	70
20	20	75
21	21	70
22	22	72
23	23	75
24	24	68
25	25	75
26	26	73
27	27	76
28	28	70
29	29	70
30	30	70
31	31	70
32	32	71
33	33	70
34	34	71
35	35	75
36	36	71
37	37	72
38	38	71
39	39	68
40	40	70
41	41	70
42	42	68
43	43	69
44	44	70
45	45	68
46	46	69
47	47	70
48	48	65
49	49	68
50	50	68
51	51	65
52	52	65
53	53	65
54	54	70
55	55	68
56	56	70
57	57	68
58	58	69

59	59	70
60	60	68
61	61	69
62	62	71
63	63	68
64	64	70
65	65	70
66	66	69
67	67	70
68	68	70
69	69	70
70	70	70
71	71	70
72	72	69
73	73	70
74	74	70
75	75	68
76	76	70
77	77	69
78	78	65
79	79	70
80	80	67
81	81	67
82	82	70
83	83	70
84	84	66
85	85	71
86	86	69
87	87	71
88	88	70
89	89	70
90	90	71
91	91	70
92	92	70
93	93	70
94	94	70
95	95	75
96	96	75
97	97	72
98	98	72
99	99	71
100	100	71
101	101	70

102	102	71
103	103	75
104	104	75
105	105	79
106	106	75
107	107	75
108	108	76
109	109	73
110	110	76
111	111	70
112	112	65
113	113	76
114	114	75
115	115	70
116	116	71
117	117	70
118	118	71
119	119	75
120	120	70
121	121	70
122	122	69
123	123	70
124	124	74
125	125	75
126	126	69
127	127	68
128	128	70
129	129	70
130	130	70
131	131	70
132	132	71
133	133	70
134	134	70
135	135	61
136	136	70
137	137	70
138	138	68
139	139	68
140	140	69
141	141	65
142	142	68
143	143	70
144	144	65

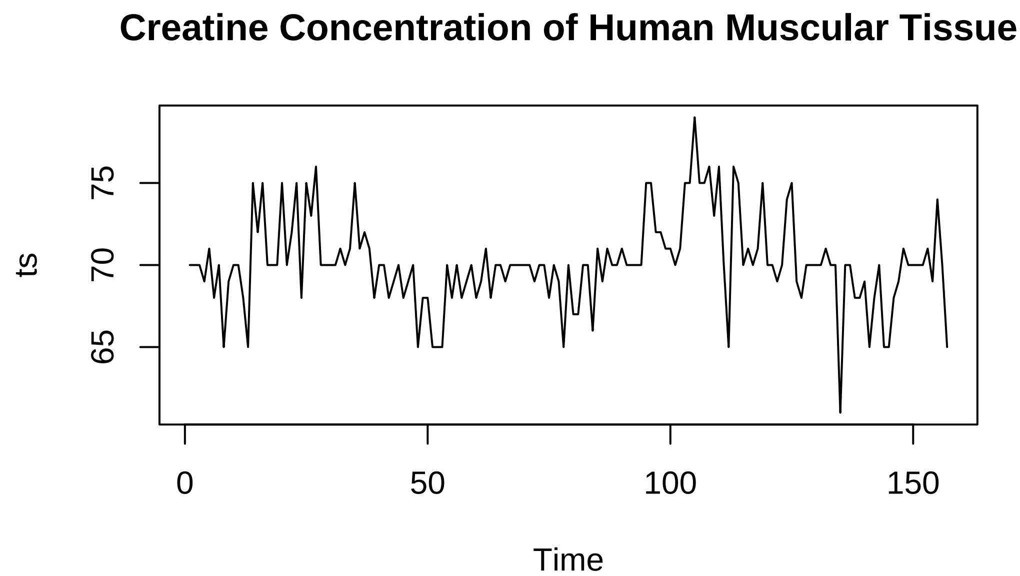
```
145 145      65
146 146      68
147 147      69
148 148      71
149 149      70
150 150      70
151 151      70
152 152      70
153 153      71
154 154      69
155 155      74
156 156      70
157 157      65
```

A

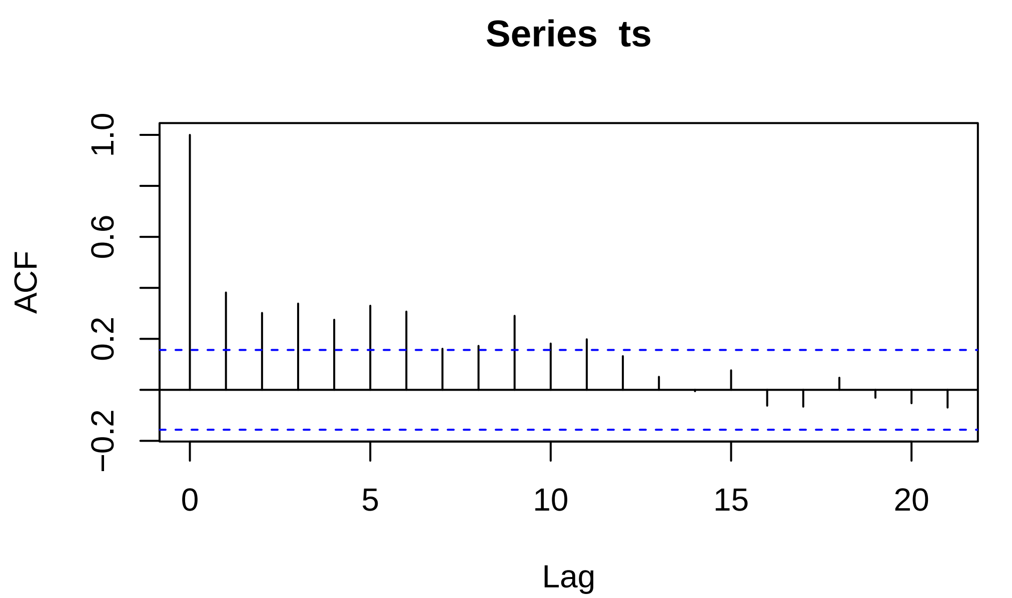
The Model should follow a **Random Walk**.

B

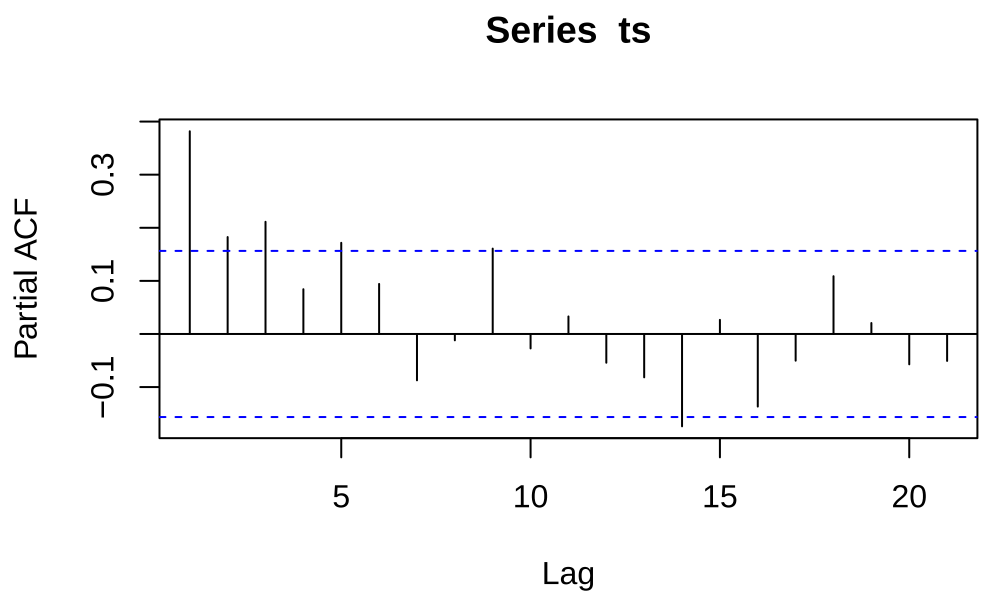
```r
ts <- ts(data$gehalt, start=c(1,1))
plot(ts, main="Creatine Concentration of Human Muscular Tissue")
```


Creatine Concentration of Human Muscular Tissue
ts
65
70
75
0
50
100
150
Time

```
acf(ts)
```


Series ts
ACF
−0.2
0.2
0.6
1.0
0
5
10
15
20
Lag

```r
pacf(ts)
```



Exercise 4.3

```r
data <- read.table("/home/nils/dev/mscids-notes/fs26/rtp/data/ts_S3_A2.dat", header=TRUE)
data
```

```
         ts1         ts2
1   7.427463  1.89344700
2   8.038622  1.39235997
3   7.041123  2.93422413
4   5.773168  2.10064648
5   8.160588  2.73526141
6   5.514591 -0.23575500
7   6.903783  0.60717902
8   7.284780  0.20313980
9   8.174462  0.85700880
10  7.962540 -0.06326550
11  9.337197  0.08407929
12  6.776165  1.34731183
```

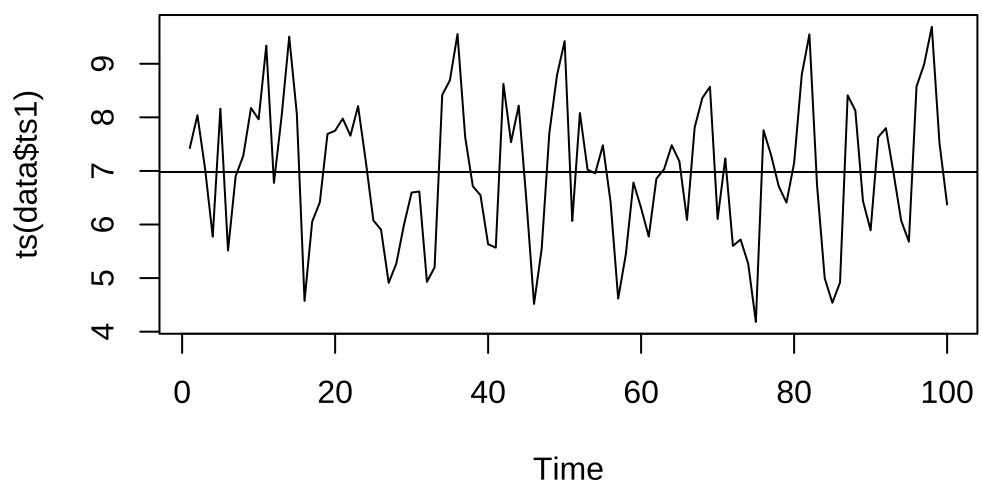
13	8.010281	0.27704869
14	9.504559	0.30647574
15	8.055161	-0.20768157
16	4.574962	0.55853578
17	6.053686	0.32482161
18	6.415292	0.56654555
19	7.689112	-1.33545483
20	7.751134	0.02101520
21	7.976683	-0.26924753
22	7.657075	0.09569397
23	8.209365	0.17903630
24	7.189110	-0.51107990
25	6.076007	0.10100015
26	5.904521	-0.91194955
27	4.911079	-1.87167605
28	5.272051	-0.42921775
29	5.994748	-0.06186044
30	6.596451	0.37270281
31	6.615690	1.25841903
32	4.929451	-0.92061370
33	5.199883	0.12910943
34	8.417599	-3.19097263
35	8.691317	0.65876784
36	9.553059	-2.47282333
37	7.632782	-0.89834419
38	6.714935	-3.78358168
39	6.546750	-2.31858371
40	5.631620	-5.41593970
41	5.568173	-3.61327652
42	8.624064	-4.87393373
43	7.536150	-1.86029757
44	8.218978	-3.13891714
45	6.454686	-2.87620119
46	4.517669	-4.36831467
47	5.544926	-0.99453666
48	7.704792	-2.06242097
49	8.782640	-1.93536262
50	9.421981	-0.27652616
51	6.066220	-0.09226471
52	8.079956	0.82491564
53	7.023590	0.98476985
54	6.953968	1.08480474
55	7.478501	0.74534046

56	6.428462	-0.21617618
57	4.619684	-1.18476322
58	5.444857	1.79089087
59	6.782672	-0.05229164
60	6.306878	2.19121470
61	5.775629	0.96234044
62	6.856888	3.74296443
63	7.031658	3.29587859
64	7.479299	4.49431247
65	7.179595	3.90446597
66	6.086827	2.88027329
67	7.811007	3.25173833
68	8.359928	4.06892465
69	8.570288	3.91413732
70	6.103120	3.13622105
71	7.233487	3.11135092
72	5.600355	2.22822026
73	5.720108	2.08893559
74	5.267714	3.01880137
75	4.181915	4.75152261
76	7.759538	2.00889682
77	7.289992	1.19854723
78	6.708407	-0.42980002
79	6.409256	1.32936588
80	7.149641	-0.31581407
81	8.793360	1.43193906
82	9.547907	0.13391761
83	6.731675	0.87756691
84	4.997799	-0.70971071
85	4.539894	-0.86230888
86	4.917282	-0.41473453
87	8.410019	2.07142213
88	8.129199	0.54368449
89	6.439980	-0.14256344
90	5.892990	-1.42365515
91	7.632958	0.27076179
92	7.798479	0.41497522
93	6.958376	3.54619346
94	6.069862	0.07431166
95	5.681058	1.78325016
96	8.574597	-0.56227142
97	8.991007	0.38310512
98	9.689271	-1.44303228

```
99  7.525061 -1.10243229
100 6.373842 -1.31733214
```

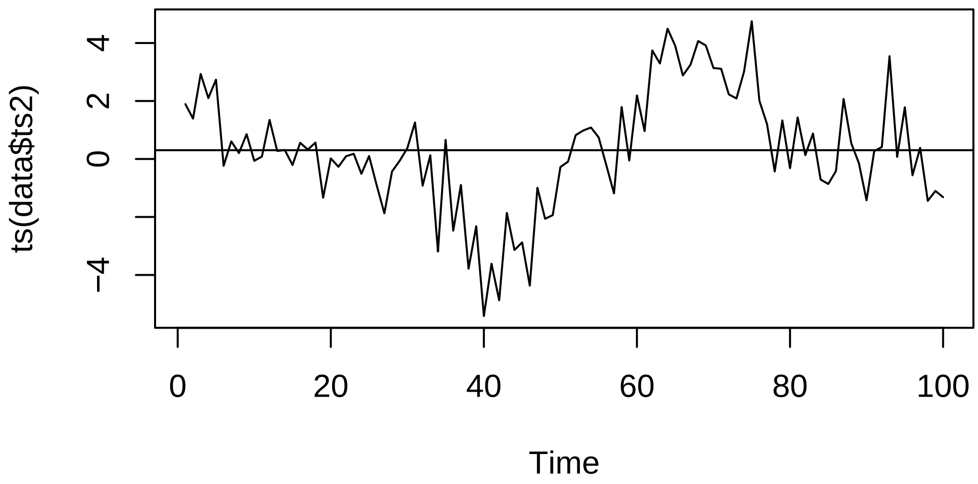
A

```r
plot(ts(data$ts1))
abline(h=mean(data$ts1))
```



$ts1$ seems to be stationary.

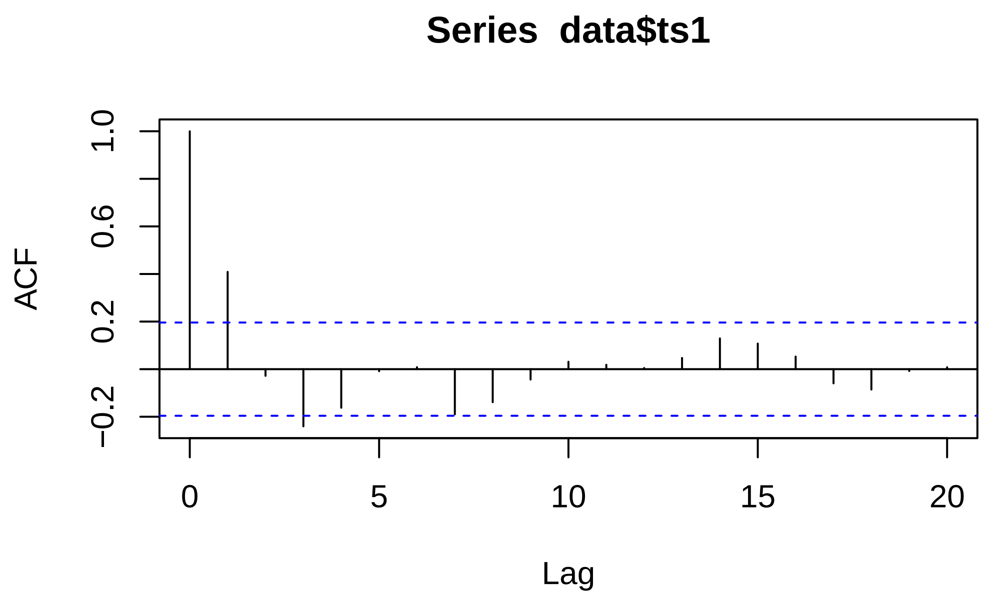
```r
plot(ts(data$ts2))
abline(h=mean(data$ts2))
```



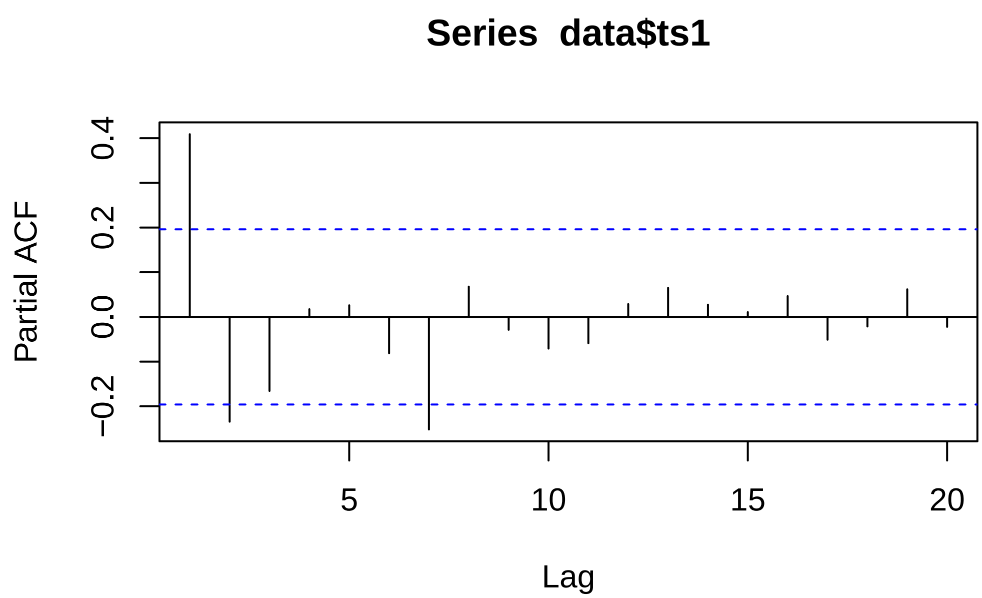
$ts2$ seems not to be stationary.

B

```
acf(data$ts1)
```

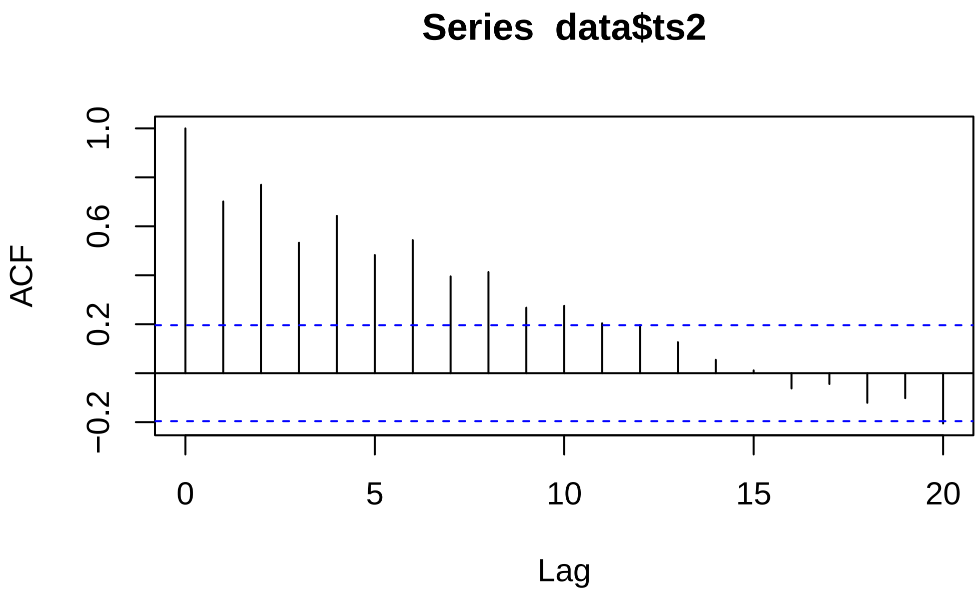



```r
pacf(data$ts1)
```

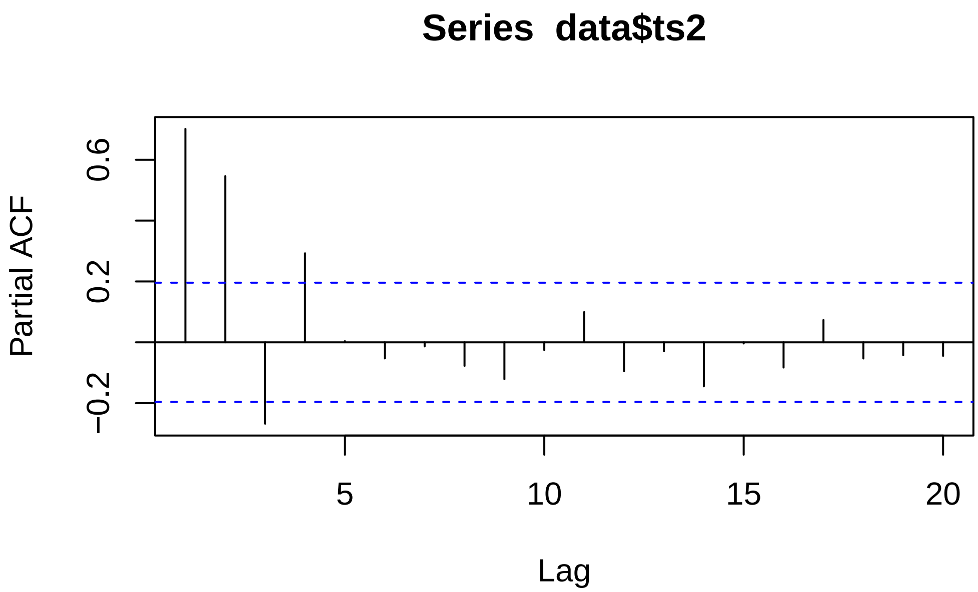


The time serie 1 seems to be a $ACF(1)$-process.

```r
acf(data$ts2)
```




```r
pacf(data$ts2)
```



The time serie 1 seems to be a $ACF(2)$-process.

Series 5

Task 5.1

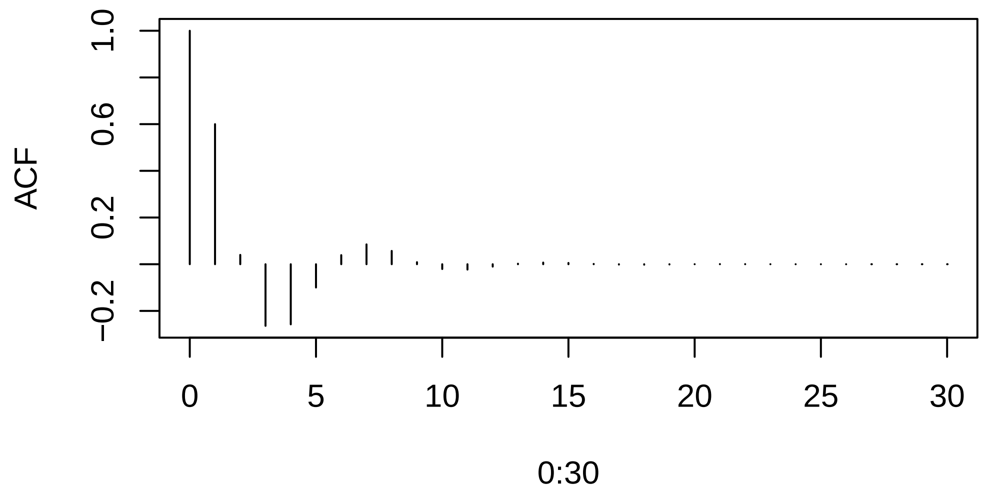
A

I

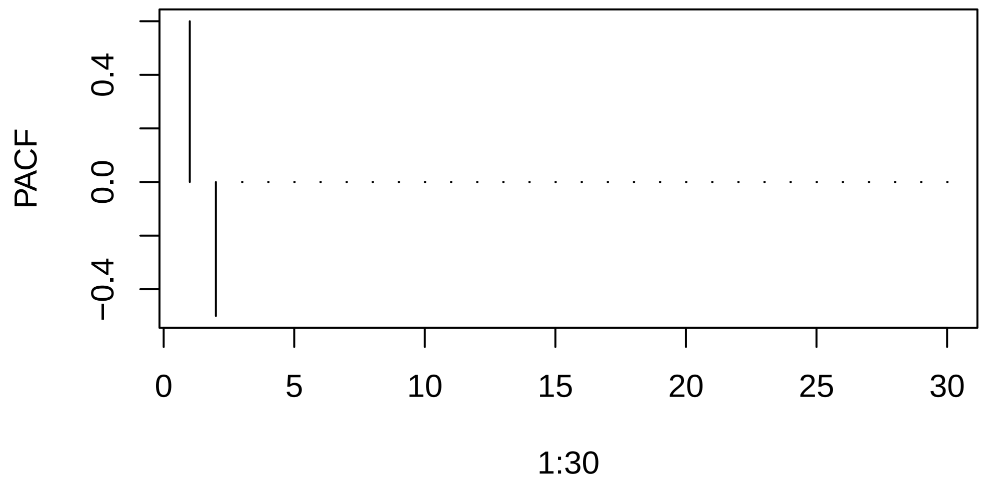
The ACF plot should have a drop at lag 3.

II

```
plot(
    0:30,
    ARMAacf(
        ar = c(0.9, -0.5),
        lag.max = 30
    ),
    type = "h",
    ylab = "ACF"
)
```




```r
plot(
    1:30,
    ARMAacf(
        ar = c(0.9, -0.5),
        lag.max = 30,
        pacf = TRUE
    ),
    type = "h",
    ylab = "PACF"

)
```

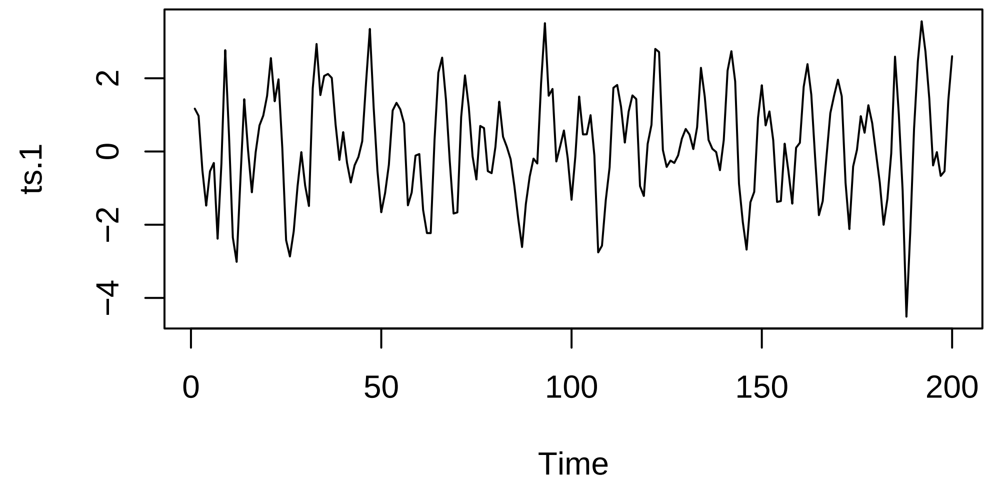


III

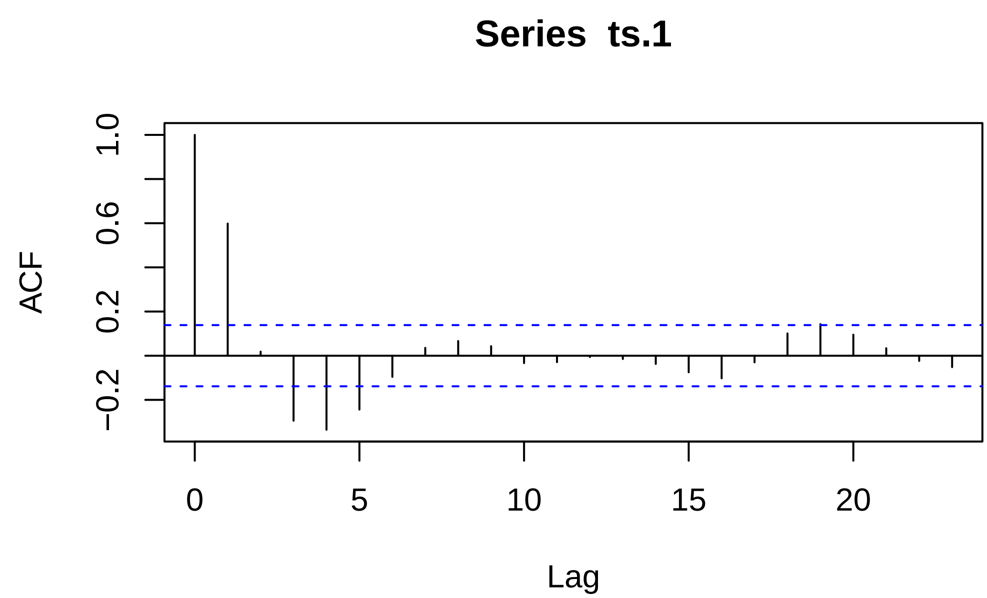
```
r.sim1 <- arima.sim(
    n = 200,
    model = list(ar = c(0.9, -0.5))
)
```

IV

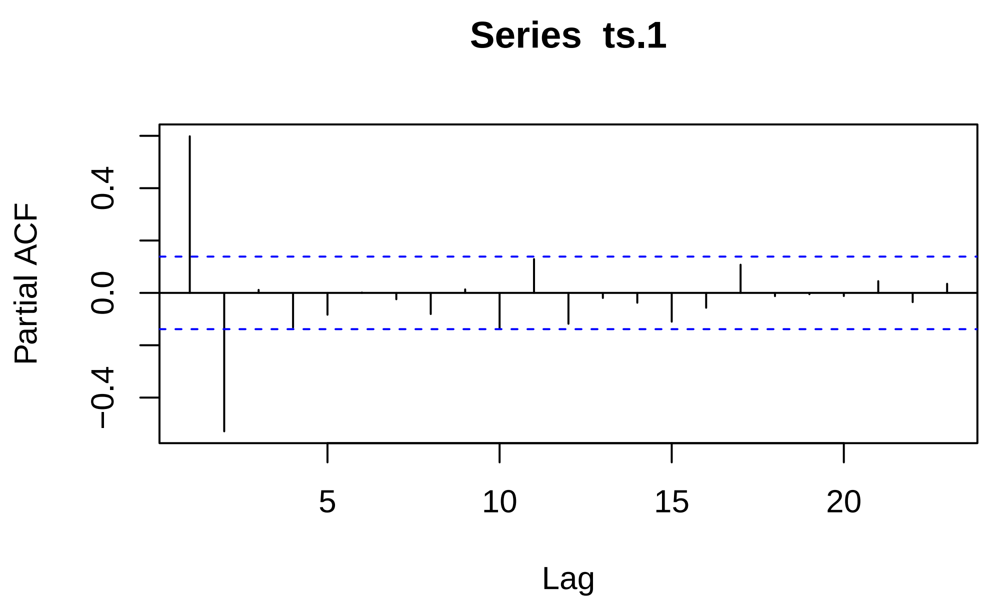
```
ts.1 <- ts(r.sim1)
plot(ts.1)
```


ts.1
2
0
−2
−4
0
50
100
150
200
Time

```
acf(ts.1)
```


Series ts.1
ACF
1.0
0.6
0.2
−0.2
0
5
10
15
20
Lag

```r
pacf(ts.1)
```



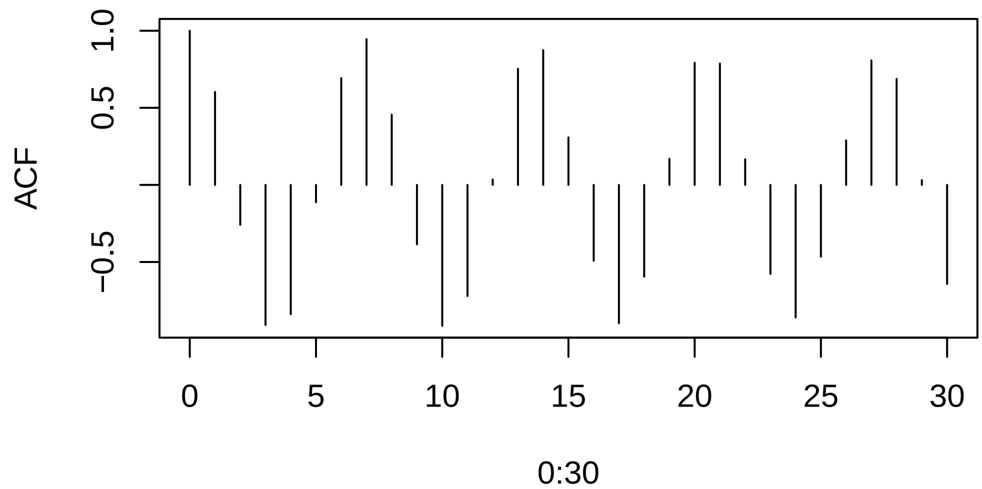
B

I

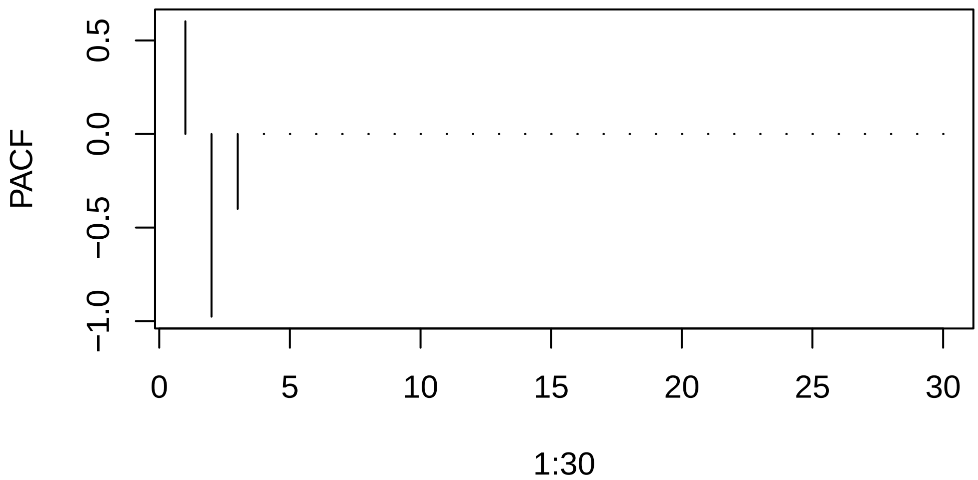
The PACF plot should have a drop at lag 4.

II

```r
plot(
    0:30,
    ARMAacf(
        ar = c(0.8, -0.5, -0.4),
        lag.max = 30
    ),
    type = "h",
    ylab = "ACF"
)
```




```
plot(
    1:30,
    ARMAacf(
        ar = c(0.8, -0.5, -0.4),
        lag.max = 30,
        pacf = TRUE
    ),
    type = "h",
    ylab = "PACF"

)
```

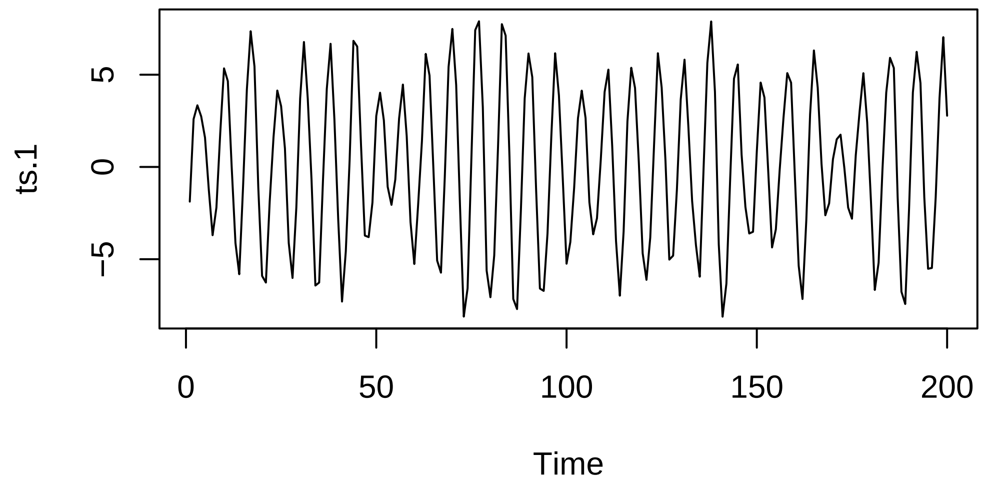


III

```
r.sim1 <- arima.sim(
    n = 200,
    model = list(ar = c(0.8, -0.5, -0.4))
)
```

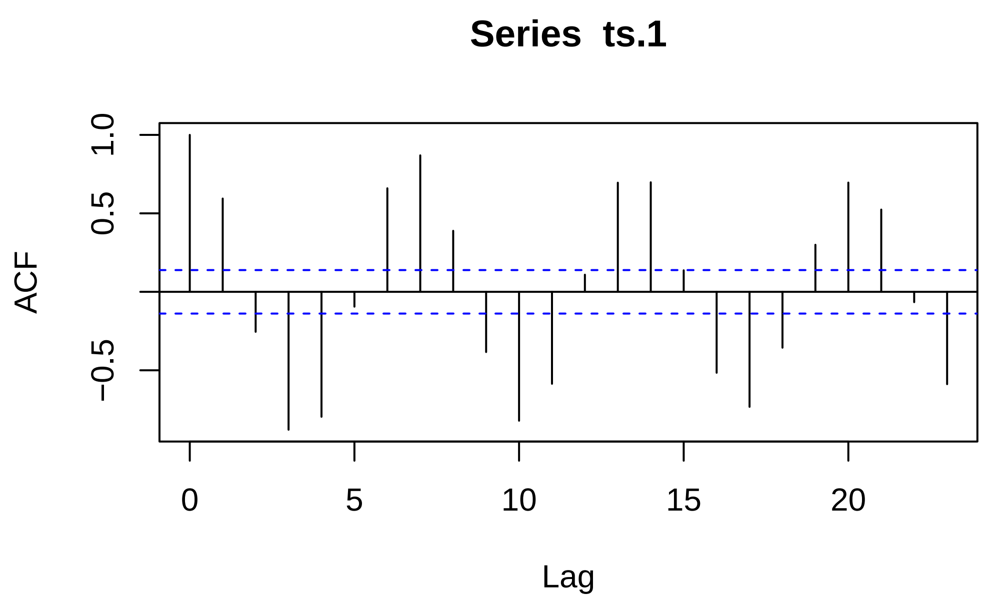
IV

```
ts.1 <- ts(r.sim1)
plot(ts.1)
```



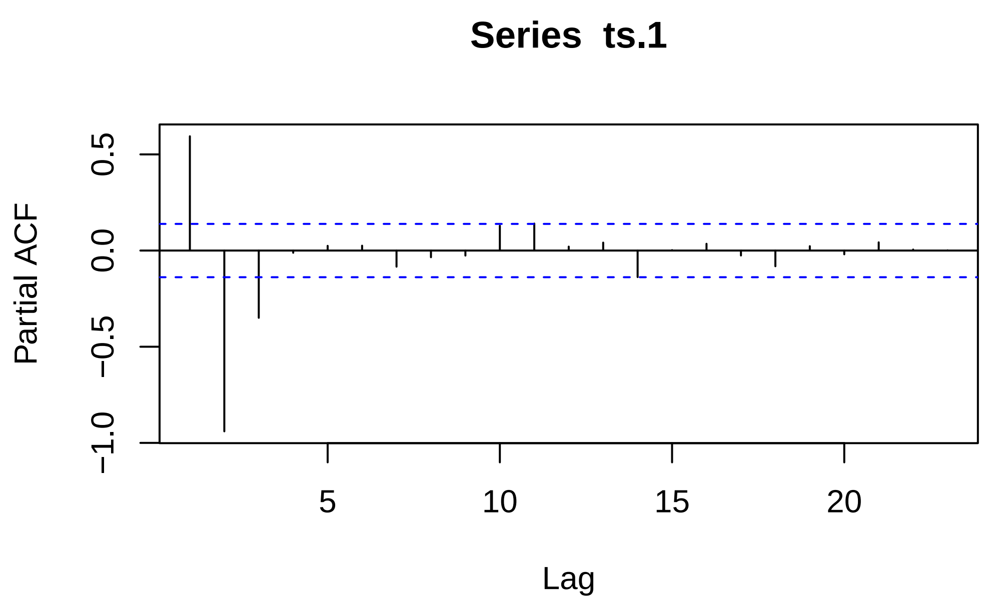
ts.1
5
0
−5
0
50
100
150
200
Time


```
acf(ts.1)
```



Series ts.1
ACF
1.0
0.5
−0.5
0
5
10
15
20
Lag

```r
pacf(ts.1)
```



Exercise 5.2

A

I

```r
roots_i <- polyroot(c(1, -0.5, -2))
abs(roots_i)
```

```
[1] 0.5930703 0.8430703
```

The process is not stationary.

II

```r
roots_ii <- polyroot(c(1, -1))
abs(roots_ii)
```

```
[1] 1
```

The process is not stationary.

B

$$0.5 + x = 1$$

$$x = 1 - 0.5$$

$$x = 0.5$$

The process is stationary for $\alpha_2 = 0.5$.

C

The condition for stationarity is $\mid \alpha \mid < 1$. Therefore, if $\mid \alpha \mid \geq 1$, the root $z = \dfrac{1}{\alpha}$ lies either on the unit circle ($\mid z \mid = 1$) or inside it ($\mid z \mid \leq 1$). This mathematically confirms that the model is non-stationary in these cases.