tharunte_Homework5

Chapter 6 R commands

```r
library(TSA)
```

```
##
## Attaching package: 'TSA'

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

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

```r
# install.packages("uroot")
library(uroot)
```

Exhibit 6.5 Sample Autocorrelation of an MA(1) Process with = 0

```r
data(ma1.1.s)
win.graph(width=4.875,height=3,pointsize=8)
acf(ma1.1.s,xaxp=c(0,20,10))
```

Exhibit 6.6 Alternative Bounds for the Sample ACF for the MA(1) Process

```r
acf(ma1.1.s,ci.type='ma',xaxp=c(0,20,10))
```

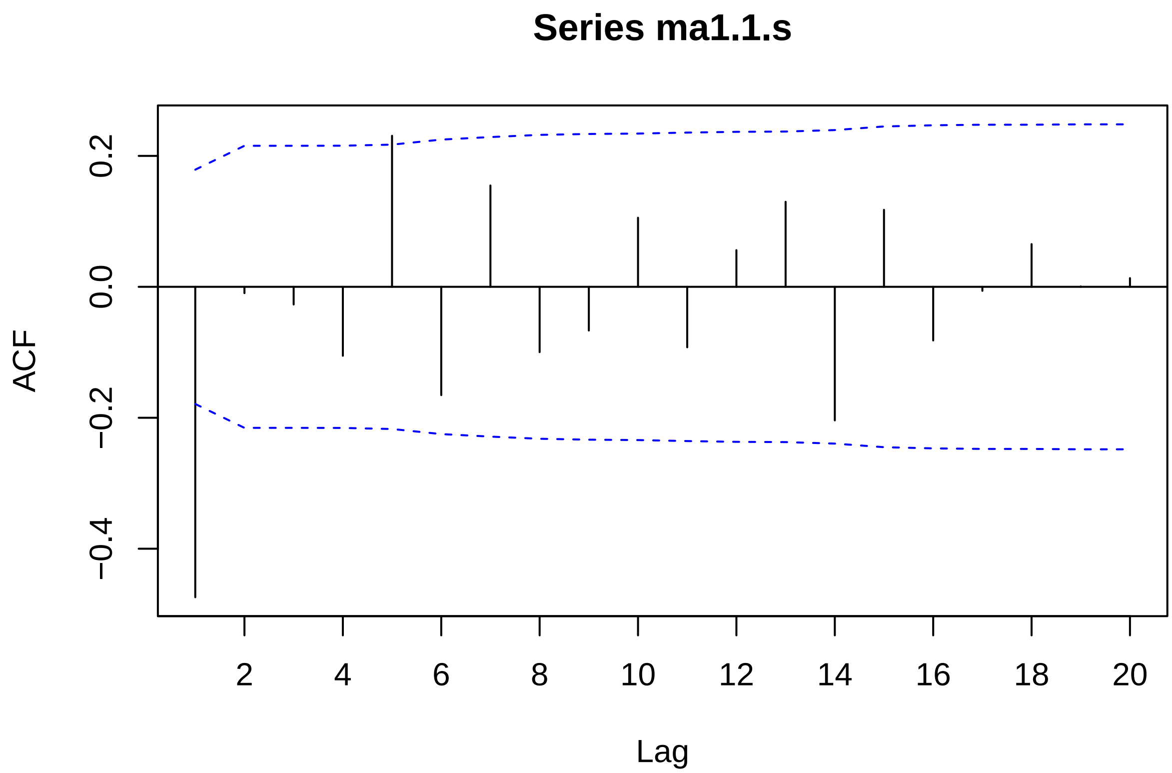


Exhibit 6.7 Sample Autocorrelation for an MA(1) Process with $= -0.9$

```
data(ma1.2.s); acf(ma1.2.s,xaxp=c(0,20,10))
```

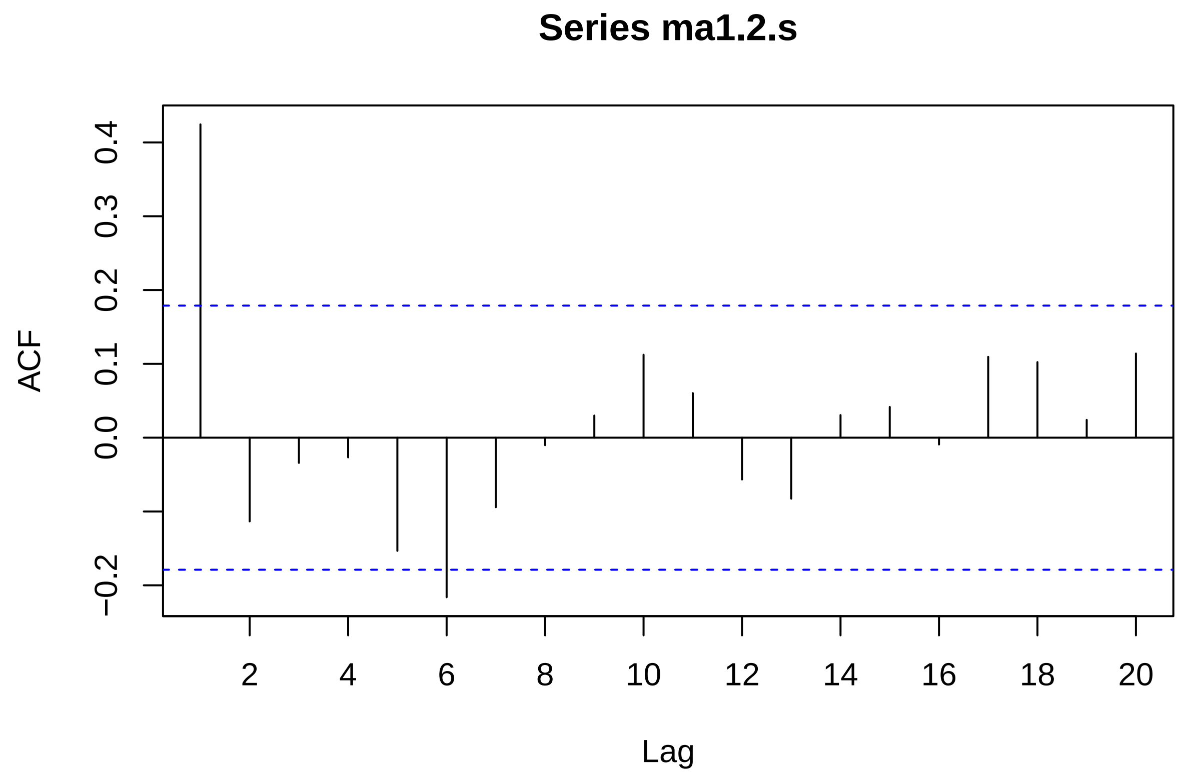


Exhibit 6.8 Sample ACF for an MA(2) Process with $1 = 1$ and $2 = -0.6$

```
data(ma2.s); acf(ma2.s,xaxp=c(0,20,10))
```

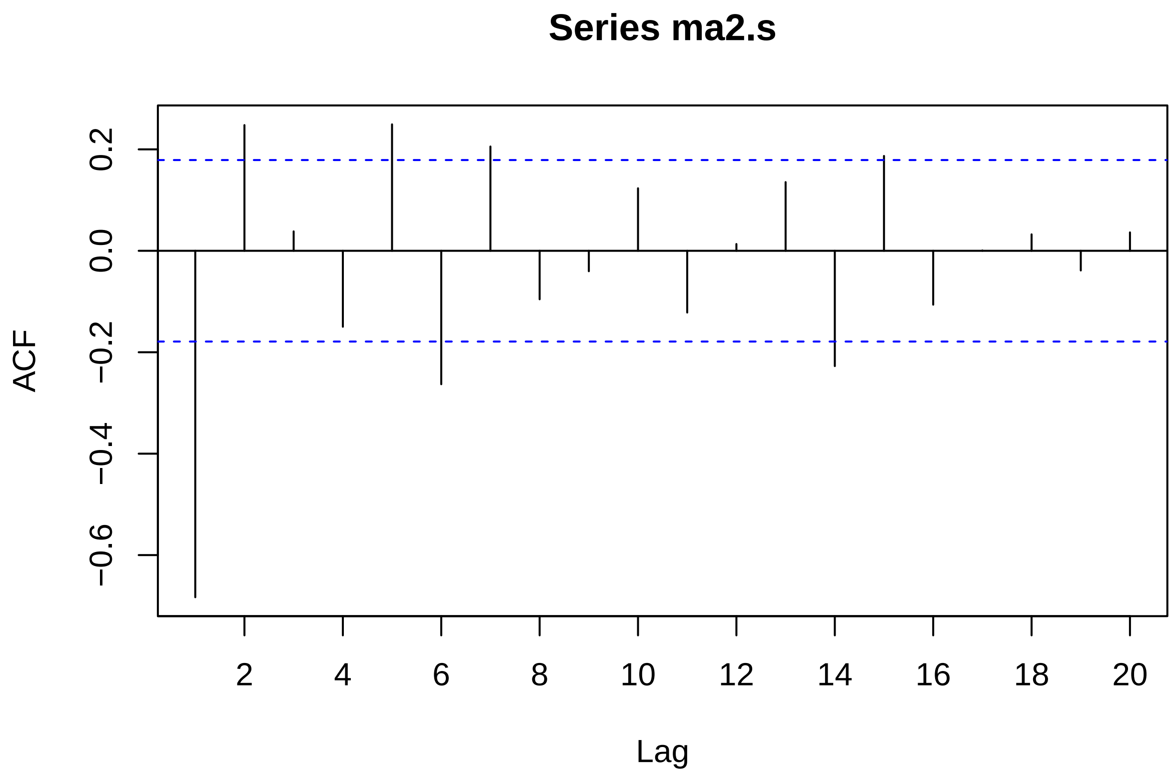


Exhibit 6.9 Alternative Bounds for the Sample ACF for the MA(2) Process

```
acf(ma2.s,ci.type='ma',xaxp=c(0,20,10))
```

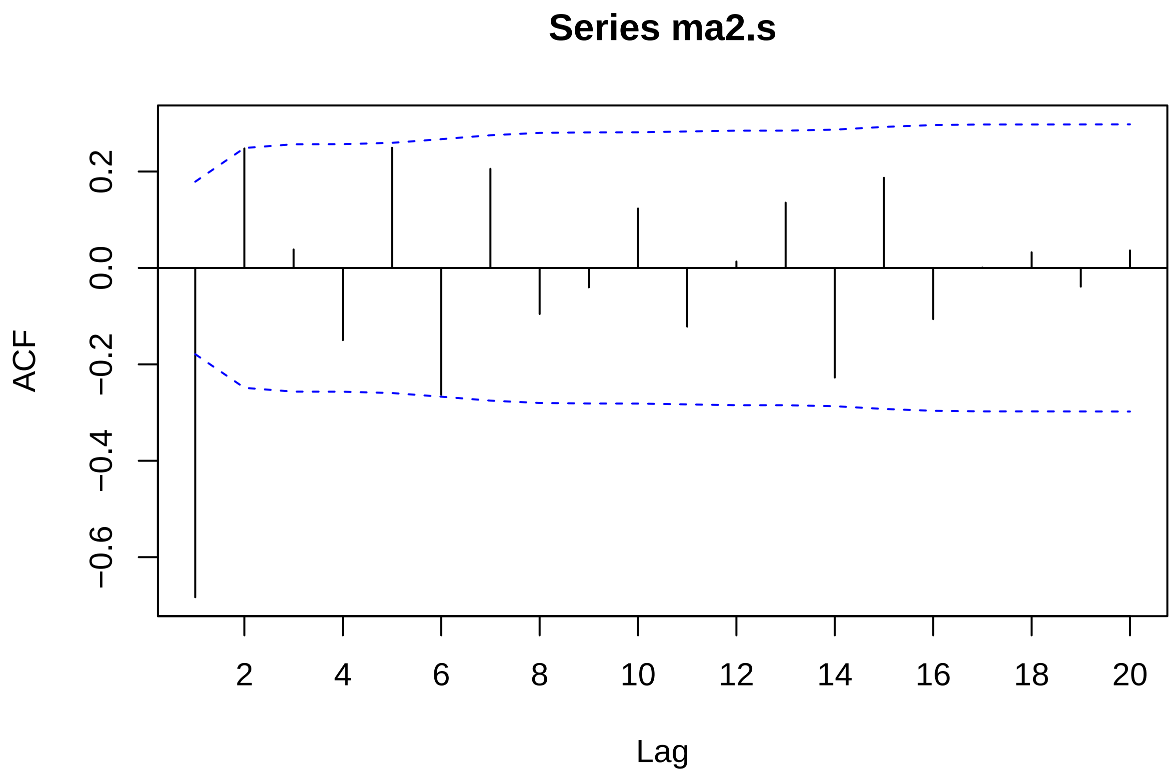


Exhibit 6.11 on page 121.

```
data(ar1.s)
pacf(ar1.s,xaxp=c(0,20,10))
```

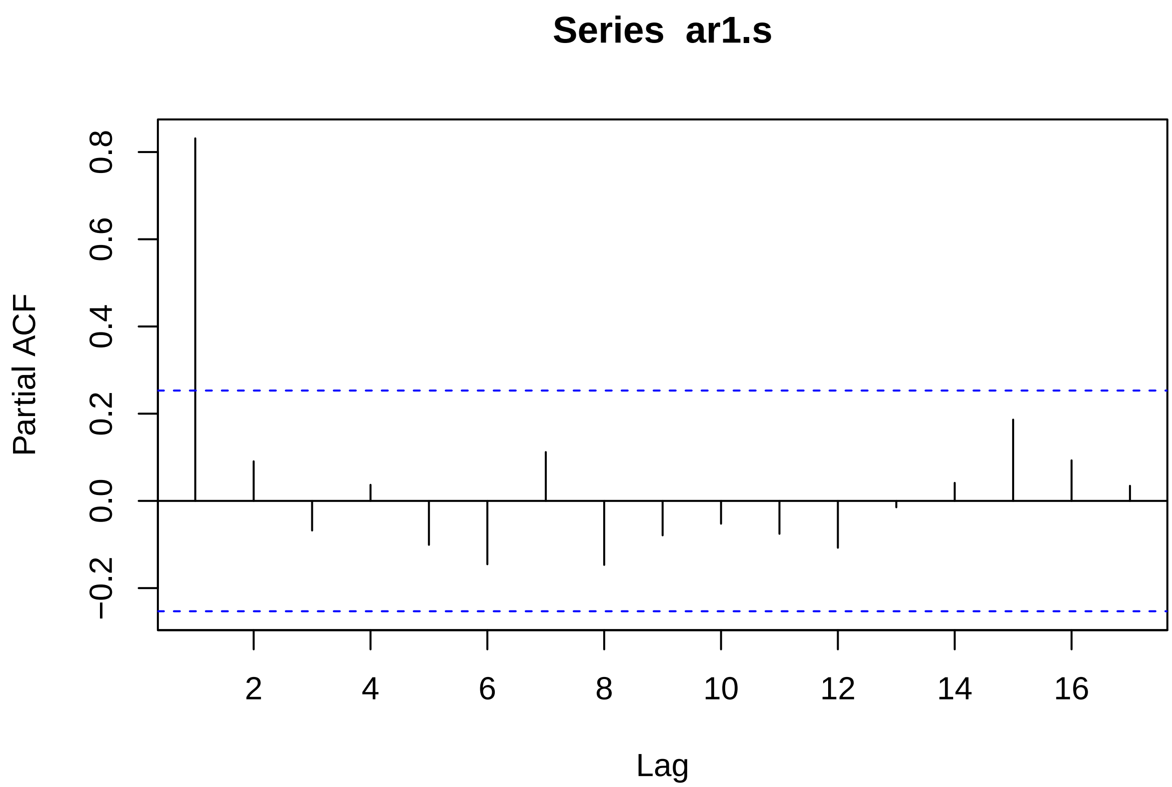


Exhibit 6.17 on page 124.

```
data(arma11.s)
eacf(arma11.s)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x o o o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x o x o o o o o o o o  o  o  o
## 5 x o o o o o o o o o o  o  o  o
## 6 x o o o x o o o o o o  o  o  o
## 7 x o o o x o o o o o o  o  o  o
```

```
eacf(arma11.s,ar.max=10,ma.max=10)
```

```
## AR/MA
##    0 1 2 3 4 5 6 7 8 9 10
## 0  x x x x o o o o o o o
## 1  x o o o o o o o o o o
## 2  x o o o o o o o o o o
## 3  x x o o o o o o o o o
## 4  x o x o o o o o o o o
## 5  x o o o o o o o o o o
```

```
## 6  x o o o x o o o o o o
## 7  x o o o x o o o o o o
## 8  x o o o x o o o o o o
## 9  o o x o x o o o o o o
## 10 x o o o x o o o o o o
```

```r
# install.packages("tseries")
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.4.2
```

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

```r
data("rwalk")
ar(diff(rwalk))
```

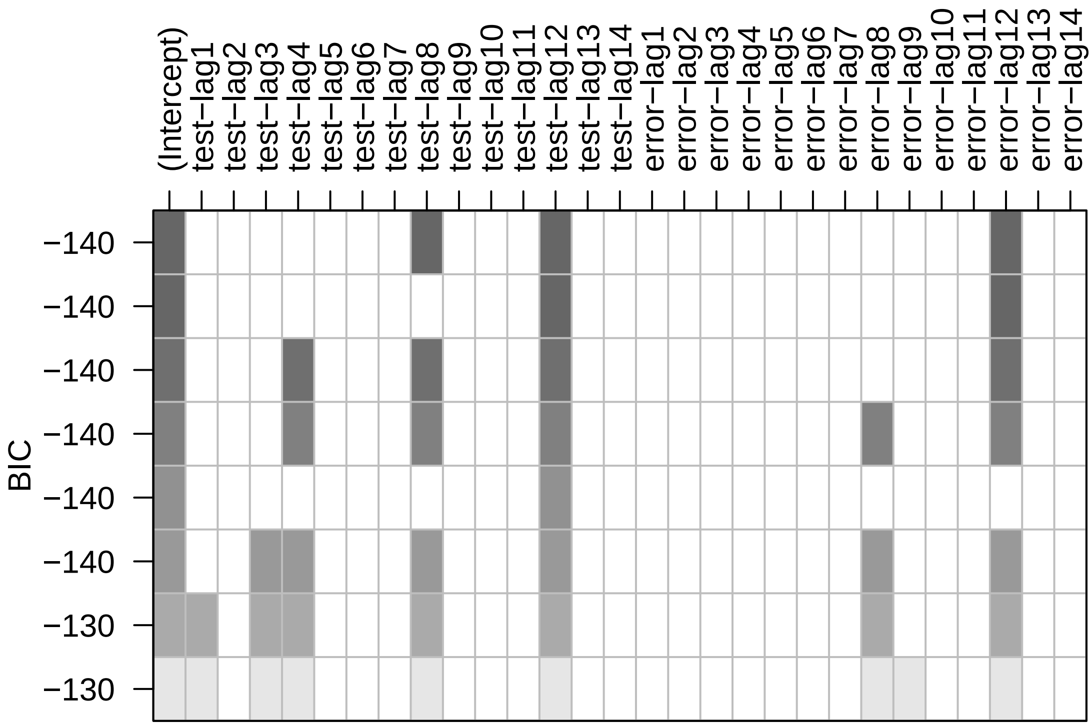
```
##
## Call:
## ar(x = diff(rwalk))
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -0.1622  -0.1234   0.0036  -0.1961  -0.1597  -0.3418   0.0652  -0.3228
##
## Order selected 8  sigma^2 estimated as  0.8394
```

```r
# ADF.test(rwalk,selectlags=list(mode=c(1,2,3,4,5,6,7,8),Pmax=8),itsd=c(1,0,0))
# ADF.test(rwalk,selectlags=list(Pmax=0),itsd=c(1,0,0))
```

Exhibit 6.22 on page 132.

```r
set.seed(92397)
test=arima.sim(model=list(ar=c(rep(0,11),.8),ma=c(rep(0,11),0.7)),n=120)
```

```r
res = armasubsets(y = test, nar = 14, nma = 14, y.name = 'test', ar.method = 'ols')
plot(res)
```



Chapter 7 R Commands

```
estimate.ma1.mom=function(x){r=acf(x,plot=F)$acf[1];
if (abs(r)<0.5) return((-1+sqrt(1-4*r^2))/(2*r))
else return(NA)}

# Example of using the function with a simulated MA(1) series
data(ma1.2.s)  # Load the MA(1) simulated series
estimate.ma1.mom(ma1.2.s)  # Estimate the MA(1) coefficient
```

```
## [1] -0.5554273
```

```
data(ar1.s)
ar(ar1.s,order.max=1,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = ar1.s, order.max = 1, method = "yw", AIC = F)
##
## Coefficients:
##      1
## 0.8314
```

```
##
## Order selected 1  sigma^2 estimated as  1.382
```

Exhibit 7.6, page 165.

```
data(arma11.s)
arima(arma11.s, order=c(1,0,1),method='CSS')
```

```
##
## Call:
## arima(x = arma11.s, order = c(1, 0, 1), method = "CSS")
##
## Coefficients:
##          ar1     ma1  intercept
##       0.5586  0.3669     0.3928
## s.e.  0.1219  0.1564     0.3380
##
## sigma^2 estimated as 1.199:  part log likelihood = -150.98
```

Exhibit 7.10 on page 168.

```
data('hare')
res=arima(sqrt(hare),order=c(3,0,0))
print(res)
```

```
##
## Call:
## arima(x = sqrt(hare), order = c(3, 0, 0))
##
## Coefficients:
##          ar1      ar2      ar3  intercept
##       1.0519  -0.2292  -0.3931     5.6923
## s.e.  0.1877   0.2942   0.1915     0.3371
##
## sigma^2 estimated as 1.066:  log likelihood = -46.54,  aic = 101.08
```

```
set.seed(12345)
coefm.cond.norm = arima.boot(res, cond.boot = TRUE, is.normal = TRUE, B = 1000, init = sqrt(hare))  # B
signif(apply(coefm.cond.norm, 2, function(x) { quantile(x, c(0.025, 0.975), na.rm = T) }), 3)
```

```
##         ar1    ar2     ar3 intercept noise var
## 2.5%  0.593 -0.667 -0.6740      5.12     0.548
## 97.5% 1.280  0.244 -0.0135      6.38     1.540
```

```
temp=apply(coefm.cond.norm,2,function(x)
{quantile (x,c(.025,.975),na.rm=T)})
signif(temp,3)
```

```
##         ar1    ar2     ar3 intercept noise var
## 2.5%  0.593 -0.667 -0.6740      5.12     0.548
## 97.5% 1.280  0.244 -0.0135      6.38     1.540
```

Chapter 8 R Commands

Exhibit 8.2 on page 177.

```
data(hare)
m1.hare=arima(sqrt(hare),order=c(3,0,0))
m1.hare
```

```
##
## Call:
## arima(x = sqrt(hare), order = c(3, 0, 0))
##
## Coefficients:
##          ar1      ar2      ar3  intercept
##       1.0519  -0.2292  -0.3931     5.6923
## s.e.  0.1877   0.2942   0.1915     0.3371
##
## sigma^2 estimated as 1.066:  log likelihood = -46.54,  aic = 101.08
```

```
m2.hare=arima(sqrt(hare),order=c(3,0,0),fixed=c(NA,0,NA,NA))
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg = xreg,
## : some AR parameters were fixed: setting transform.pars = FALSE
```

```
m2.hare
```

```
##
## Call:
## arima(x = sqrt(hare), order = c(3, 0, 0), fixed = c(NA, 0, NA, NA))
##
## Coefficients:
##          ar1  ar2      ar3  intercept
##       0.9190    0  -0.5313     5.6889
## s.e.  0.0791    0   0.0697     0.3179
##
## sigma^2 estimated as 1.088:  log likelihood = -46.85,  aic = 99.69
```

```
plot(rstandard(m2.hare),ylab='Standardized Residuals',type='b')
abline(h=0)
```

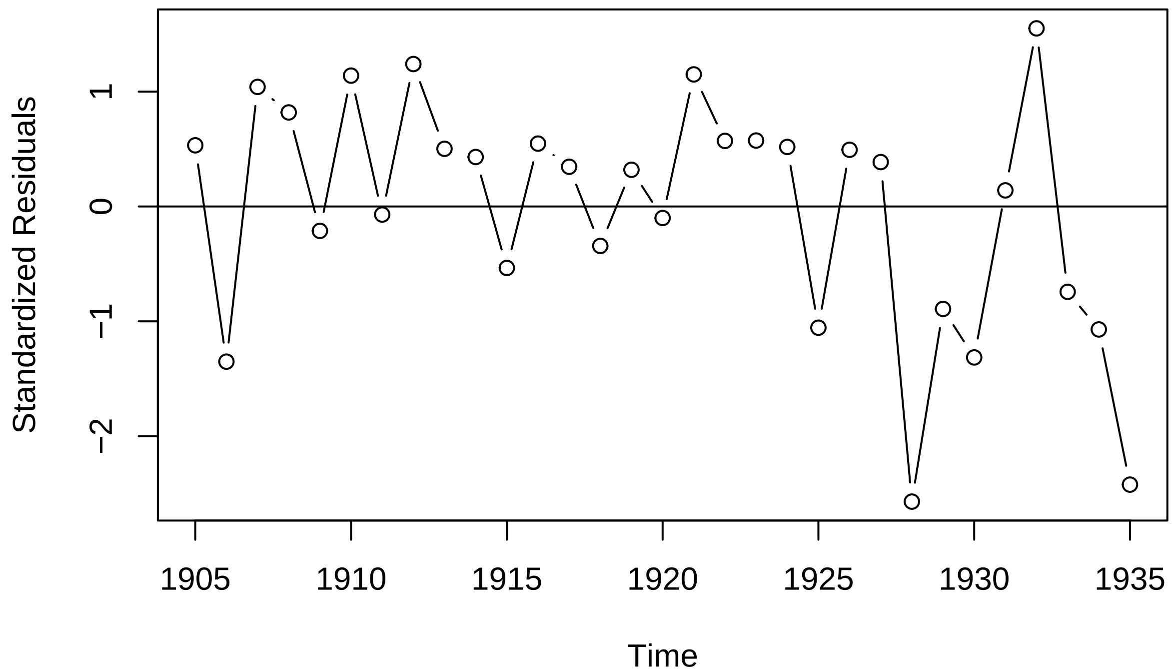
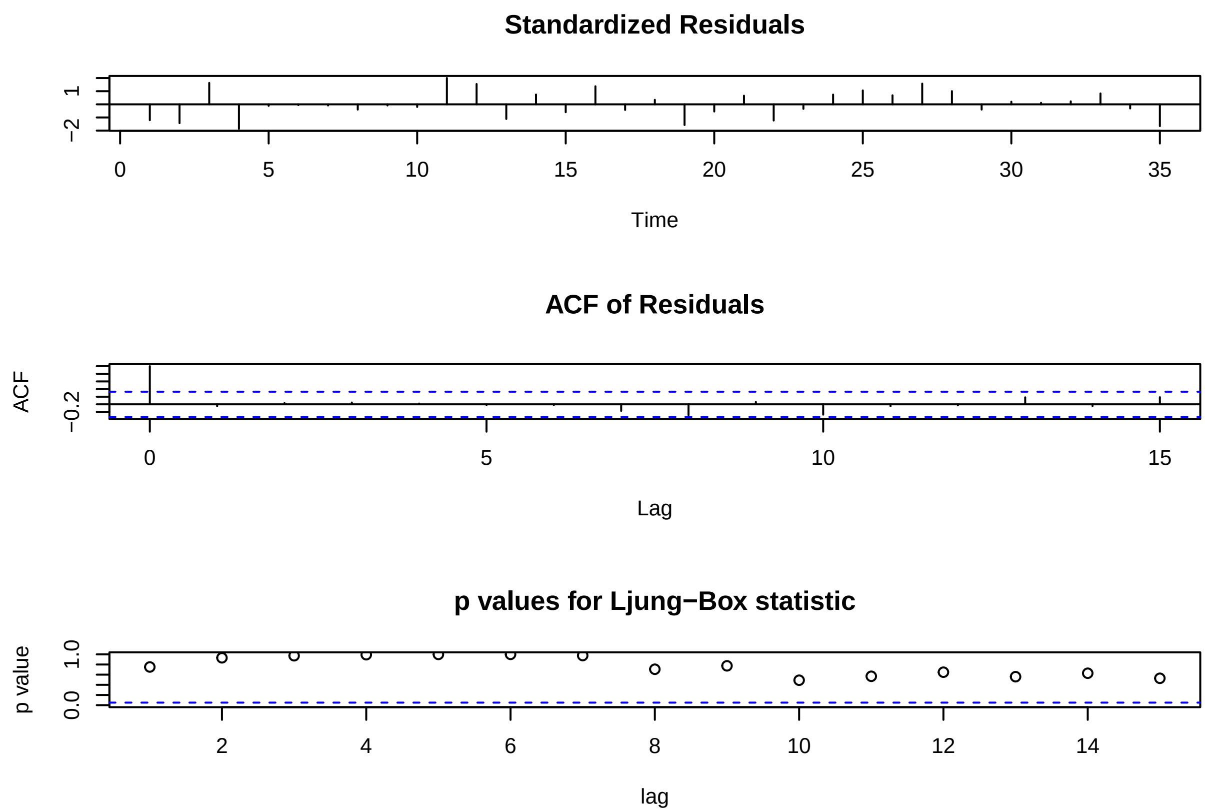


Exhibit 8.12 on page 185 (prefaced by some commands in Exhibit 8.1 on page 176)

```r
data(color)
m1.color=arima(color,order=c(1,0,0))
tsdiag(m1.color,gof=15,omit.initial=F)
```



Question 2

6.20 : Simulate an AR(1) time series with n = 48 and with = 0.7. (a) Calculate the theoretical autocorrelations at lag 1 and lag 5 for this model. (b) Calculate the sample autocorrelations at lag 1 and lag 5 and compare the values with their theoretical values. Use Equations (6.1.5) and (6.1.6) page 111, to quantify the comparisons. (c) Repeat part (b) with a new simulation. Describe how the precision of the estimate varies with different samples selected under identical conditions.

```r
set.seed(1234)
n <- 48
phi <- 0.7

ar1_series <- arima.sim(n = n, list(ar = phi))
```

```r
theoretical_lag_1 <- phi^1
theoretical_lag_5 <- phi^5

cat("Theoretical autocorrelation at lag 1:", theoretical_lag_1, "\n")
```

```
## Theoretical autocorrelation at lag 1: 0.7
```

```r
cat("Theoretical autocorrelation at lag 5:", theoretical_lag_5, "\n")
```

```
## Theoretical autocorrelation at lag 5: 0.16807
```

```r
# b - Calculate sample autocorrelations
acf_values <- acf(ar1_series, plot = FALSE)$acf

lag_1_sample <- acf_values[1]
lag_5_sample <- acf_values[5]

cat("Sample autocorrelation at lag 1:", lag_1_sample, "\n")
```

```
## Sample autocorrelation at lag 1: 0.7683024
```

```r
cat("Sample autocorrelation at lag 5:", lag_5_sample, "\n")
```

```
## Sample autocorrelation at lag 5: 0.2151106
```

```r
# Comparison using Equations (6.1.5) and (6.1.6):
# Variance estimate
n <- length(ar1_series)
var_acf <- 1 / n

cat("Variance of sample autocorrelation estimate:", var_acf, "\n")
```

```
## Variance of sample autocorrelation estimate: 0.02083333
```

```r
cat("Difference at lag 1:", abs(lag_1_sample - theoretical_lag_1), "\n")
```

```
## Difference at lag 1: 0.06830241
```

```r
cat("Difference at lag 5:", abs(lag_5_sample - theoretical_lag_5), "\n")
```

```
## Difference at lag 5: 0.0470406
```

```r
# c- Repeat part (b) with a new simulation
set.seed(42)
ar1_simulation <- arima.sim(n = n, list(ar = phi))

acf_values_simulation <- acf(ar1_simulation, plot = FALSE)$acf

sample_acf_lag1 <- acf_values_simulation[1]
sample_acf_lag5 <- acf_values_simulation[5]

cat("Sample autocorrelation at lag 1 (new simulation):", sample_acf_lag1, "\n")
```

```
## Sample autocorrelation at lag 1 (new simulation): 0.5876167
```

```r
cat("Sample autocorrelation at lag 5 (new simulation):", sample_acf_lag5, "\n")
```

```
## Sample autocorrelation at lag 5 (new simulation): -0.1316713
```

```r
cat("Difference at lag 1 (new simulation):", abs(sample_acf_lag1 - theoretical_lag_1), "\n")
```

```
## Difference at lag 1 (new simulation): 0.1123833
```

```r
cat("Difference at lag 5 (new simulation):", abs(sample_acf_lag5 - theoretical_lag_5), "\n")
```

```
## Difference at lag 5 (new simulation): 0.2997413
```

Repeating the simulation shows that sample autocorrelations vary due to randomness, even under identical conditions. Estimates at shorter lags (e.g., lag 1) are more precise and closer to theoretical values, while those at longer lags (e.g., lag 5) are less accurate and more variable. This variability highlights the impact of sample size and lag on the precision of autocorrelation estimates.

6.21: Simulate an MA(1) time series with n = 60 and with = 0.5. (a) Calculate the theoretical autocorrelation at lag 1 for this model. (b) Calculate the sample autocorrelation at lag 1, and compare the value with its theoretical value. Use Exhibit 6.2 on page 112, to quantify the comparisons. (c) Repeat part (b) with a new simulation. Describe how the precision of the estimate varies with different samples selected under identical conditions

```r
n <- 60
theta <- 0.5

ma1 <- arima.sim(model=list(ma=c(.5)),n=60)
ma1
```

```
## Time Series:
## Start = 1
## End = 60
## Frequency = 1
##  [1]  1.20643016  1.43998162  1.18124245 -0.68267986 -0.61174586  0.57842497
##  [7] -0.64176428 -1.01959049  0.30958209  1.05867699  0.84785696 -0.65389250
## [13] -1.54266905  0.96281656  1.01427494  0.21740095 -0.07667642 -1.25477716
## [19]  0.01483245  0.08885860 -0.29132663  0.84196798  1.28844627  1.80300293
## [25]  0.21988426  0.41226160  1.71628474 -0.41523365 -1.41618703 -1.56213497
## [31] -2.02508334 -0.64962445  0.69319562  1.52756755  1.64523377 -0.48083310
## [37]  1.34687758  0.25746754 -0.22787289 -0.36949898 -0.33347811  0.12701795
## [43]  0.21325748  0.03448793  0.09552645 -0.43139887 -0.74693475 -1.91320765
## [49] -1.21288327 -0.70381712  2.44556587 -0.01117073 -0.54380190 -1.42499696
## [55] -2.21724828 -0.61051548 -0.93428794 -0.50014218 -0.42917019 -0.82780105
```

```r
rho_1_theoretical <- theta / (1 + theta^2)
cat("Theoretical autocorrelation at lag 1:", rho_1_theoretical, "\n")
```

```
## Theoretical autocorrelation at lag 1: 0.4
```

```r
# (b) Calculate the sample autocorrelation at lag 1
acf_values <- acf(ma1, plot = FALSE)$acf
rho_1_sample <- acf_values[1]

# Standard error for sample autocorrelation
se_rho <- 1 / sqrt(n)

cat("Sample autocorrelation at lag 1:", rho_1_sample, "\n")
```

```
## Sample autocorrelation at lag 1: 0.4244832
```

```r
cat("Difference between sample and theoretical values:", abs(rho_1_sample - rho_1_theoretical), "\n")
```

```
## Difference between sample and theoretical values: 0.02448324
```

```r
cat("Standard error for the sample autocorrelation:", se_rho, "\n")
```

```
## Standard error for the sample autocorrelation: 0.1290994
```

```r
# Repeat simulation
set.seed(98)
ma1_series_new <- arima.sim(n = n, list(ma = theta))

# Compute sample autocorrelation for the new simulation
acf_values_new <- acf(ma1_series_new, plot = FALSE)$acf
rho_1_sample_new <- acf_values_new[1]

cat("Sample autocorrelation at lag 1 (new simulation):", rho_1_sample_new, "\n")
```

```
## Sample autocorrelation at lag 1 (new simulation): 0.3120067
```

```r
cat("Difference between new sample and theoretical values:", abs(rho_1_sample_new - rho_1_theoretical),
```

```
## Difference between new sample and theoretical values: 0.08799327
```

```r
cat("Standard error for the sample autocorrelation:", se_rho, "\n")
```

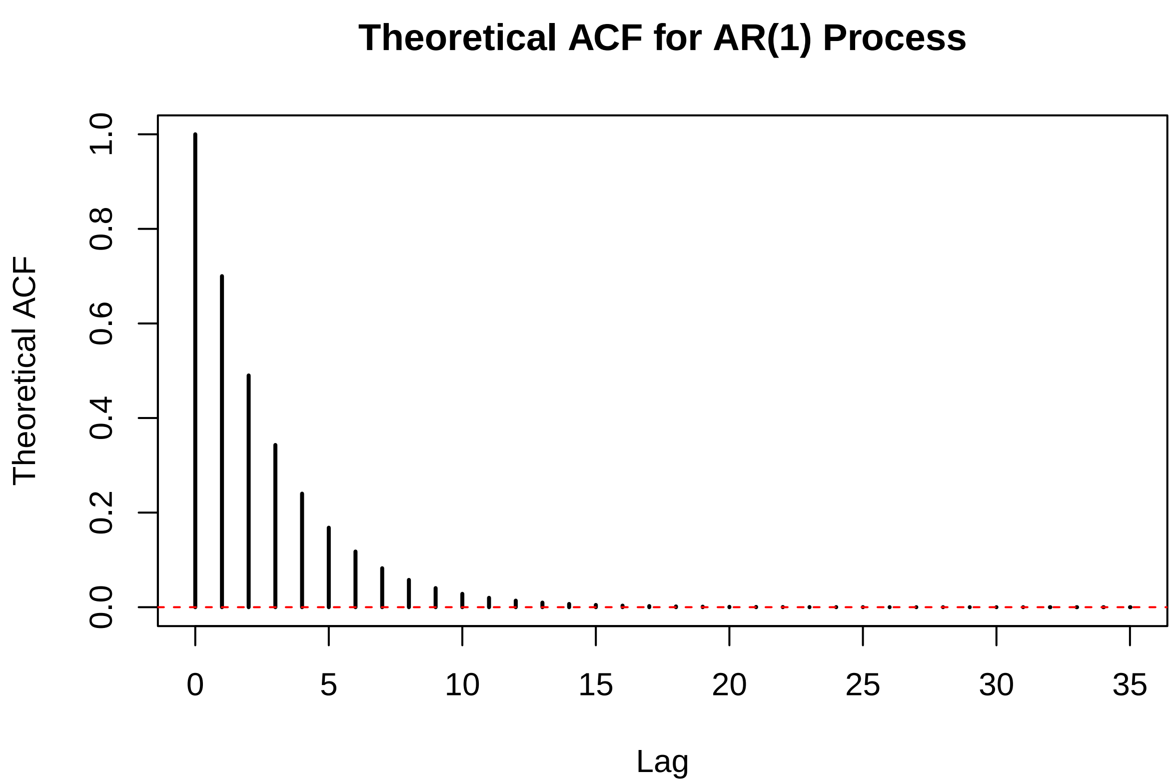
```
## Standard error for the sample autocorrelation: 0.1290994
```

Repeating the simulation shows that sample autocorrelations at lag 1 vary slightly across different samples due to randomness. The sample autocorrelation values are generally close to the theoretical value, with differences falling within the expected standard error range. This highlights that while estimates are reasonably precise, some variation is inherent due to sampling.

6.25 Simulate an AR(1) time series of length n = 36 with   = 0.7. (a) Calculate and plot the theoretical autocorrelation function for this model. Plot sufficient lags until the correlations are negligible. (b) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)? (c) What are the theoretical partial autocorrelations for this model? (d) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)? Use the large-sample standard errors reported in Exhibit 6.1 on page 111, to quantify your answer. (e) Calculate and plot the sample PACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (c)? Use the large-sample standard errors reported on page 115 to quantify your answer

```r
n <- 36
phi <- 0.7
ar1_series <- arima.sim(n = n, list(ar = phi))
```

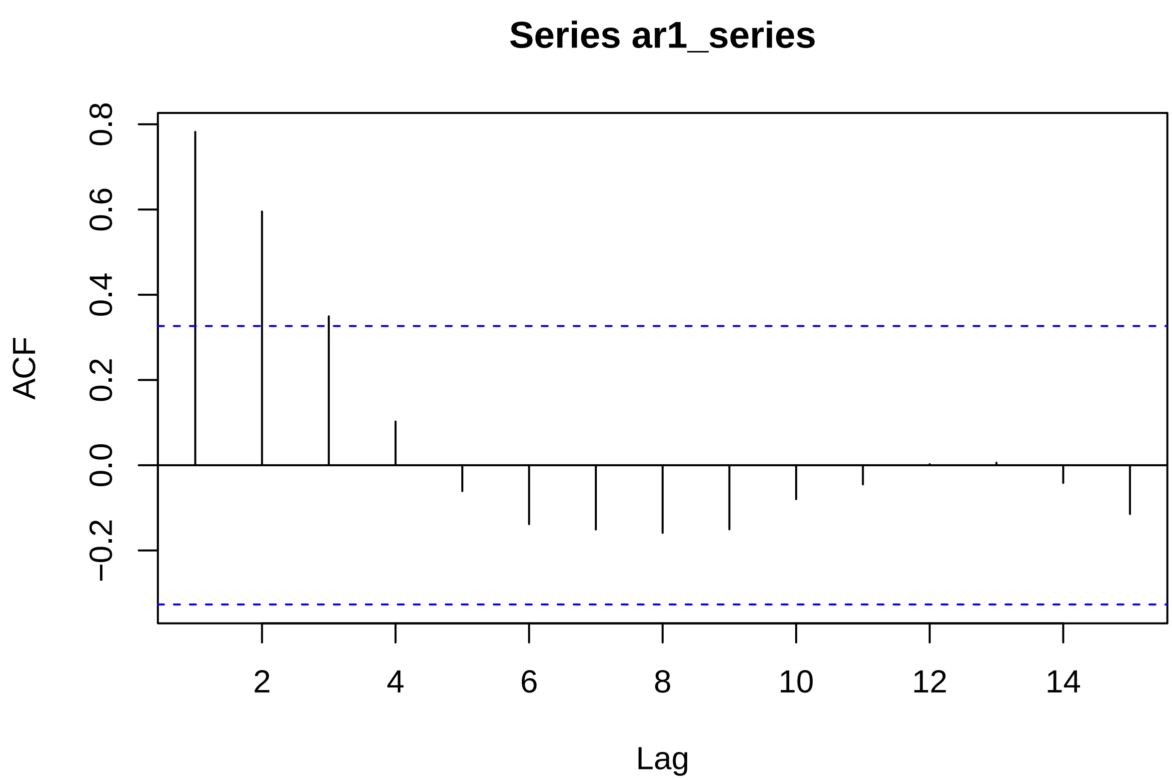
```r
# (a) Theoretical ACF for AR(1) Model

# Plot the theoretical ACF
max_lag <- n-1  # Plot until lag 35
theoretical_acf_values <- phi^(0:max_lag)
plot(0:max_lag, theoretical_acf_values, type = "h", lwd = 2,
     xlab = "Lag", ylab = "Theoretical ACF", main = "Theoretical ACF for AR(1) Process")
abline(h = 0, col = "red", lty = 2)  # Add a horizontal line at 0
```




```r
# (b) Calculate and plot the sample ACF
acf_values <- acf(ar1_series, plot = FALSE)$acf
acf_values
```

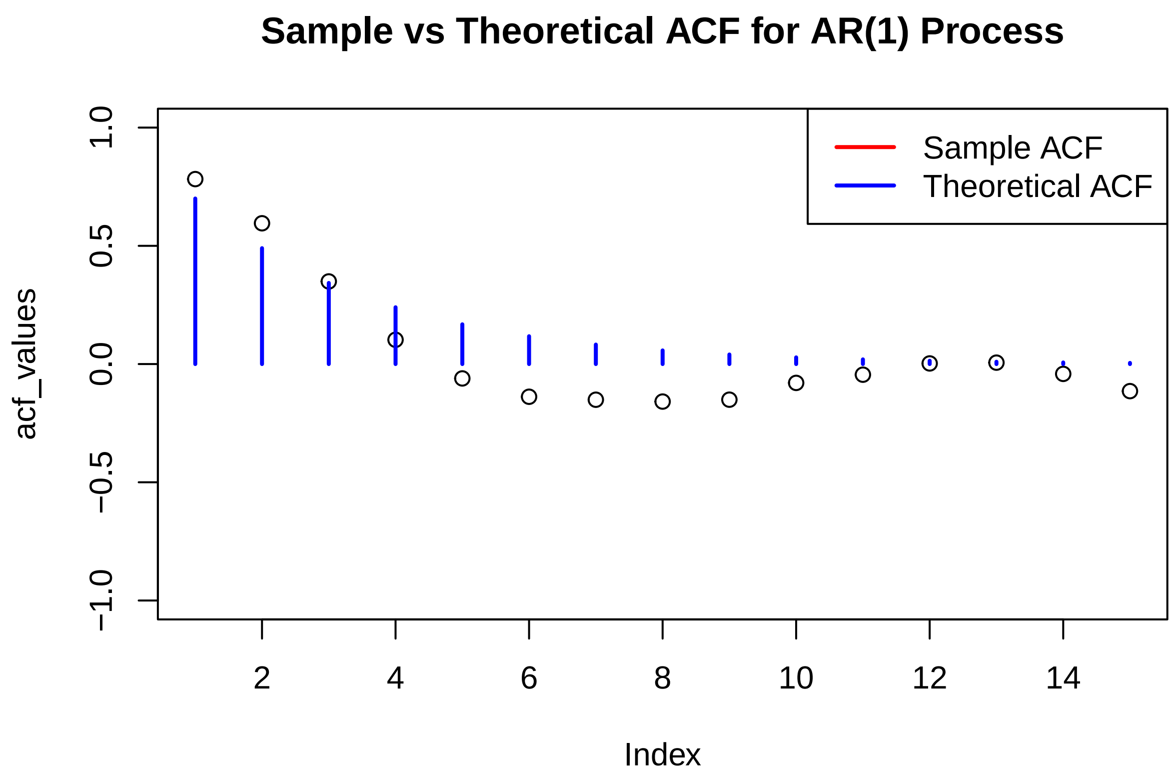
```
## , , 1
##
##                [,1]
##  [1,]  0.782238302
##  [2,]  0.595407672
##  [3,]  0.349416260
##  [4,]  0.102626844
##  [5,] -0.060834924
##  [6,] -0.138285492
##  [7,] -0.150934437
##  [8,] -0.158717950
##  [9,] -0.150672452
## [10,] -0.079759298
## [11,] -0.045000167
## [12,]  0.002552876
## [13,]  0.005972203
## [14,] -0.041578065
## [15,] -0.114489408
```

```r
# Plot the Sample ACF
acf(ar1_series)
```



(c) Theoretical Partial Autocorrelations for AR(1) The PACF for AR(1) is phi(0.7) for lag 1, and 0 for lags greater than 1

```r
# Overlay theoretical ACF on the sample ACF plot for comparison
plot(acf_values, ylim = c(-1, 1), main = "Sample vs Theoretical ACF for AR(1) Process")
points(0:max_lag, theoretical_acf_values, type = "h", lwd = 2, col = "blue")

legend("topright", legend = c("Sample ACF", "Theoretical ACF"),
       col = c("red", "blue"), lwd = 2)
```




```r
acf_se <- 1 / sqrt(n)

for (k in 1:length(acf_values)) {
  diff <- abs(acf_values[k] - theoretical_acf_values[k])  # Comparing lags 1 to max_lag
  cat("Lag", k, ": Sample ACF =", acf_values[k], "Theoretical ACF =", theoretical_acf_values[k],
      "Difference =", diff, "\n")

  # Quantify the difference using the standard error
  cat("Lag", k, ": Difference as standard errors = ", diff / acf_se, "\n")
}
```

```
## Lag 1 : Sample ACF = 0.7822383 Theoretical ACF = 1 Difference = 0.2177617
## Lag 1 : Difference as standard errors =  1.30657
## Lag 2 : Sample ACF = 0.5954077 Theoretical ACF = 0.7 Difference = 0.1045923
## Lag 2 : Difference as standard errors =  0.627554
## Lag 3 : Sample ACF = 0.3494163 Theoretical ACF = 0.49 Difference = 0.1405837
## Lag 3 : Difference as standard errors =  0.8435024
## Lag 4 : Sample ACF = 0.1026268 Theoretical ACF = 0.343 Difference = 0.2403732
```

```
## Lag 4 : Difference as standard errors =  1.442239
## Lag 5 : Sample ACF = -0.06083492 Theoretical ACF = 0.2401 Difference = 0.3009349
## Lag 5 : Difference as standard errors =  1.80561
## Lag 6 : Sample ACF = -0.1382855 Theoretical ACF = 0.16807 Difference = 0.3063555
## Lag 6 : Difference as standard errors =  1.838133
## Lag 7 : Sample ACF = -0.1509344 Theoretical ACF = 0.117649 Difference = 0.2685834
## Lag 7 : Difference as standard errors =  1.611501
## Lag 8 : Sample ACF = -0.1587179 Theoretical ACF = 0.0823543 Difference = 0.2410722
## Lag 8 : Difference as standard errors =  1.446433
## Lag 9 : Sample ACF = -0.1506725 Theoretical ACF = 0.05764801 Difference = 0.2083205
## Lag 9 : Difference as standard errors =  1.249923
## Lag 10 : Sample ACF = -0.0797593 Theoretical ACF = 0.04035361 Difference = 0.1201129
## Lag 10 : Difference as standard errors =  0.7206774
## Lag 11 : Sample ACF = -0.04500017 Theoretical ACF = 0.02824752 Difference = 0.07324769
## Lag 11 : Difference as standard errors =  0.4394861
## Lag 12 : Sample ACF = 0.002552876 Theoretical ACF = 0.01977327 Difference = 0.01722039
## Lag 12 : Difference as standard errors =  0.1033224
## Lag 13 : Sample ACF = 0.005972203 Theoretical ACF = 0.01384129 Difference = 0.007869084
## Lag 13 : Difference as standard errors =  0.0472145
## Lag 14 : Sample ACF = -0.04157806 Theoretical ACF = 0.009688901 Difference = 0.05126697
## Lag 14 : Difference as standard errors =  0.3076018
## Lag 15 : Sample ACF = -0.1144894 Theoretical ACF = 0.006782231 Difference = 0.1212716
## Lag 15 : Difference as standard errors =  0.7276298
```

```r
# (e) Calculate and plot the sample PACF
pacf_values <- pacf(ar1_series, plot = FALSE)$acf
pacf_values
```

```
## , , 1
##
##              [,1]
##  [1,]  0.78223830
##  [2,] -0.04248635
##  [3,] -0.26558418
##  [4,] -0.20627373
##  [5,]  0.02783503
##  [6,]  0.10496813
##  [7,]  0.03774002
##  [8,] -0.13742775
##  [9,] -0.09116965
## [10,]  0.18180446
## [11,]  0.01786888
## [12,] -0.02687919
## [13,] -0.15402629
## [14,] -0.14597969
## [15,] -0.03820138
```

For AR(1): Sample PACF should match theoretical PACF with significant value at lag 1, zero for others

```r
n <- length(ar1_series)
se <- 1 / sqrt(n)

# Theoretical PACF for AR(1)
```

```r
theoretical_pacf <- c(0.7, rep(0, length(pacf_values) - 1))

# Evaluate and compare
for (k in 1:length(pacf_values)) {
  diff <- abs(pacf_values[k] - theoretical_pacf[k])
  cat("Lag", k, ": Sample PACF =", pacf_values[k],
      "Theoretical PACF =", theoretical_pacf[k],
      "Difference =", diff, "\n")

  # Quantify in terms of SE
  cat("Lag", k, ": Difference in terms of SE =", diff / se, "\n\n")
}
```

```
## Lag 1 : Sample PACF = 0.7822383 Theoretical PACF = 0.7 Difference = 0.0822383
## Lag 1 : Difference in terms of SE = 0.4934298
##
## Lag 2 : Sample PACF = -0.04248635 Theoretical PACF = 0 Difference = 0.04248635
## Lag 2 : Difference in terms of SE = 0.2549181
##
## Lag 3 : Sample PACF = -0.2655842 Theoretical PACF = 0 Difference = 0.2655842
## Lag 3 : Difference in terms of SE = 1.593505
##
## Lag 4 : Sample PACF = -0.2062737 Theoretical PACF = 0 Difference = 0.2062737
## Lag 4 : Difference in terms of SE = 1.237642
##
## Lag 5 : Sample PACF = 0.02783503 Theoretical PACF = 0 Difference = 0.02783503
## Lag 5 : Difference in terms of SE = 0.1670102
##
## Lag 6 : Sample PACF = 0.1049681 Theoretical PACF = 0 Difference = 0.1049681
## Lag 6 : Difference in terms of SE = 0.6298088
##
## Lag 7 : Sample PACF = 0.03774002 Theoretical PACF = 0 Difference = 0.03774002
## Lag 7 : Difference in terms of SE = 0.2264401
##
## Lag 8 : Sample PACF = -0.1374277 Theoretical PACF = 0 Difference = 0.1374277
## Lag 8 : Difference in terms of SE = 0.8245665
##
## Lag 9 : Sample PACF = -0.09116965 Theoretical PACF = 0 Difference = 0.09116965
## Lag 9 : Difference in terms of SE = 0.5470179
##
## Lag 10 : Sample PACF = 0.1818045 Theoretical PACF = 0 Difference = 0.1818045
## Lag 10 : Difference in terms of SE = 1.090827
##
## Lag 11 : Sample PACF = 0.01786888 Theoretical PACF = 0 Difference = 0.01786888
## Lag 11 : Difference in terms of SE = 0.1072133
##
## Lag 12 : Sample PACF = -0.02687919 Theoretical PACF = 0 Difference = 0.02687919
## Lag 12 : Difference in terms of SE = 0.1612751
##
## Lag 13 : Sample PACF = -0.1540263 Theoretical PACF = 0 Difference = 0.1540263
## Lag 13 : Difference in terms of SE = 0.9241578
##
## Lag 14 : Sample PACF = -0.1459797 Theoretical PACF = 0 Difference = 0.1459797
```

```
## Lag 14 : Difference in terms of SE = 0.8758781
##
## Lag 15 : Sample PACF = -0.03820138 Theoretical PACF = 0 Difference = 0.03820138
## Lag 15 : Difference in terms of SE = 0.2292083
```

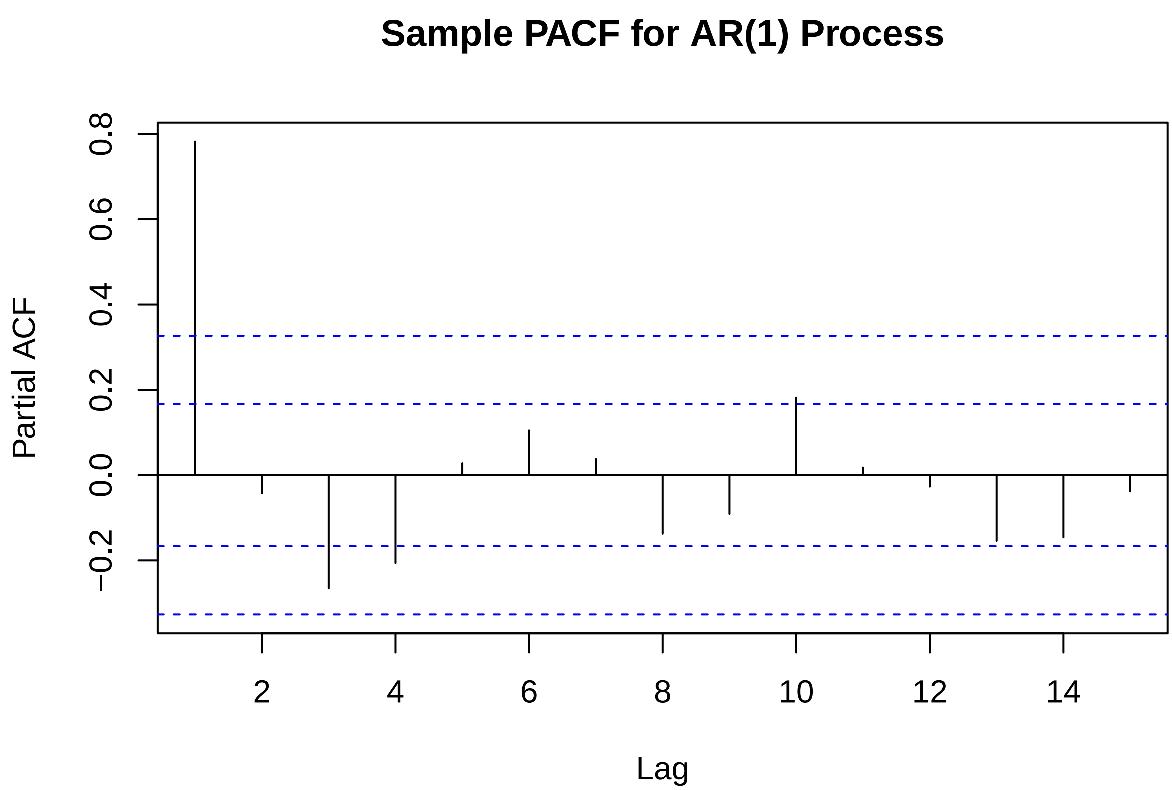
Analysis:

From your provided PACF values:

1. Lag 1: Sample PACF =0.7822, Theoretical PACF = 0.7, Difference = 0.0822. Difference in SE: 0.0822/0.1667 0.49, indicating a close match.
2. Lag k>1: The theoretical PACF is 0, but sample values vary around 0. Differences at higher lags can be quantified similarly in SE units.

Observation: 1. At lag 1, the sample PACF closely matches the theoretical value, as expected for AR(1). 2. At higher lags, the sample PACF fluctuates around 0, with deviations due to random sampling noise. Most deviations are within 2 SE units, consistent with theoretical expectations.

```r
# Plot the Sample PACF
pacf(ar1_series, main = "Sample PACF for AR(1) Process")
abline(h = c(-1, 1) * se, col = "blue", lty = 2)
```

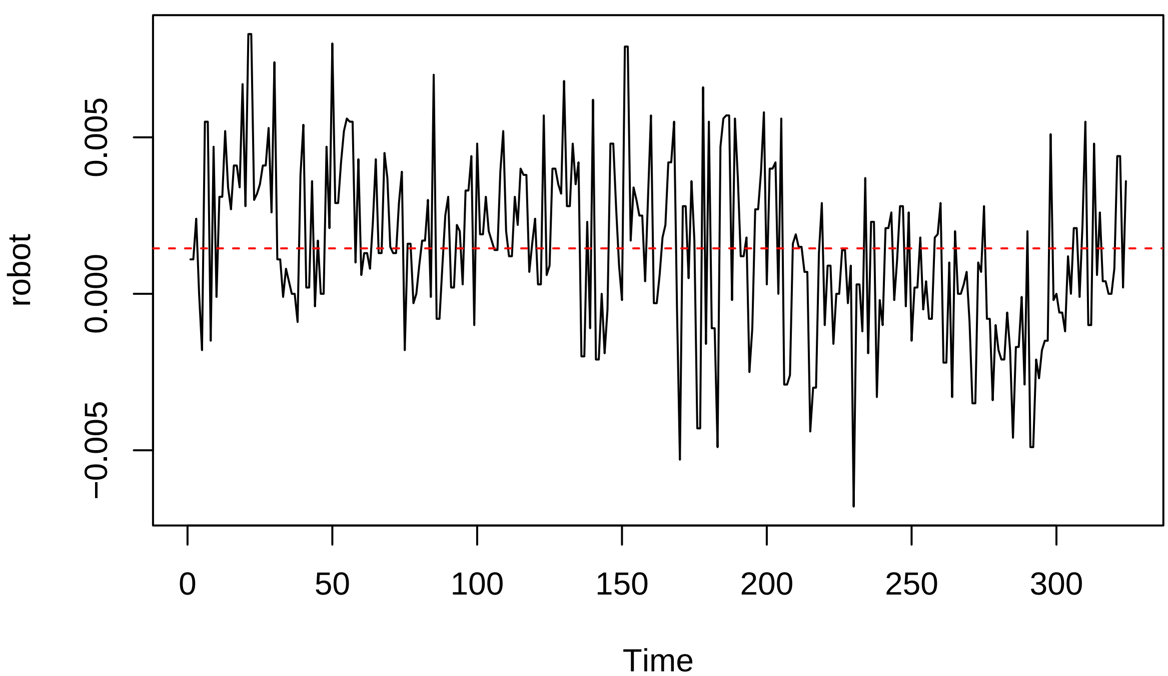


1. The sample PACF for lag 1 is consistent with the theoretical value (=0.7).
2. The sample PACF at higher lags fluctuates near 0, as expected for an AR(1) process, with deviations mostly within the 95% confidence bounds (±2 SE).

6.36 The data file named robot contains a time series obtained from an industrial robot. The robot was put through a sequence of maneuvers, and the distance from a desired ending point was recorded in inches. This was repeated 324 times to form the time series. (a) Display the time series plot of the data. Based on this information, do these data appear to come from a stationary or nonstationary process? (b) Calculate and plot the sample ACF and PACF for these data. Based on this additional information, do these data appear to come from a stationary or nonstationary process? (c) Calculate and interpret the sample EACF.

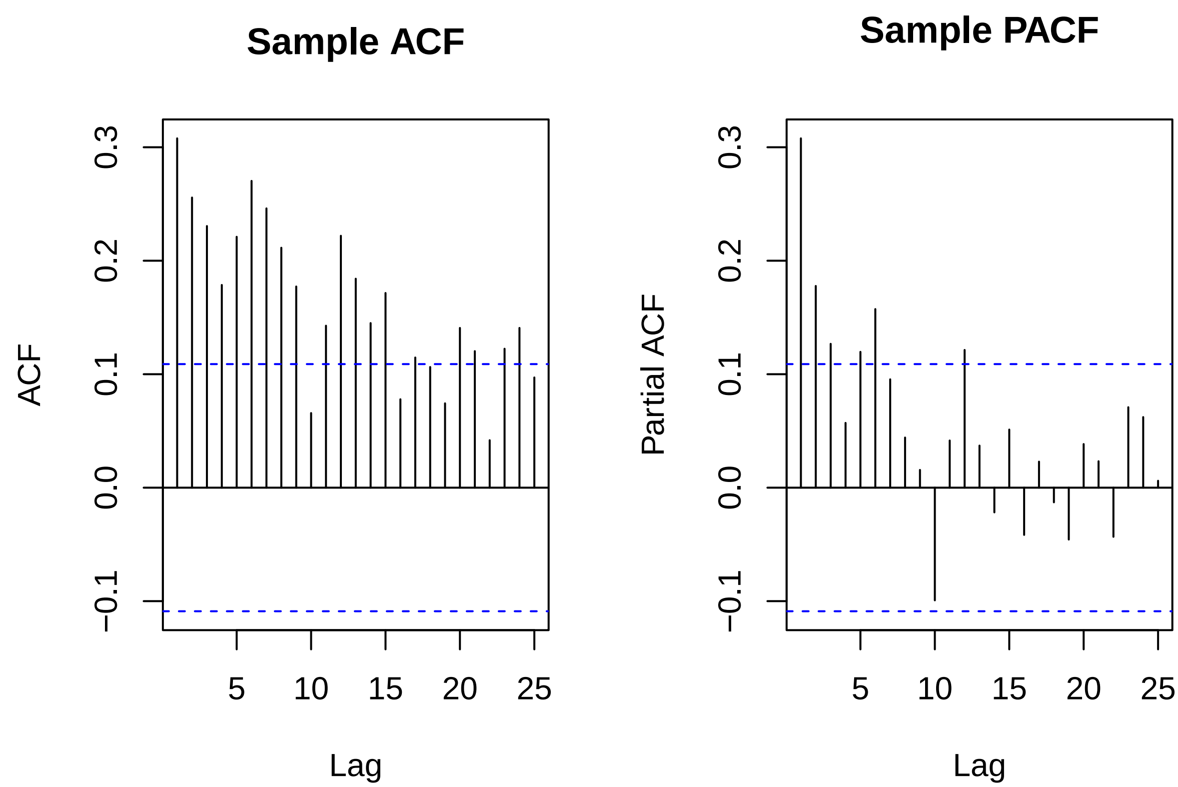
```r
library(TSA)
data(robot)
```

```r
# a) time series plot of the data
plot(robot)
abline(h = mean(robot), col = "red", lty = 2)
```



In general, for a stationary process, ACF should die down quickly while PACF cuts off after a certain lag. If both decay slowly, this implies nonstationarity.

```r
par(mfrow = c(1, 2))
acf(robot, main = "Sample ACF")
pacf(robot, main = "Sample PACF")
```



Analysis The plots show concerning patterns that suggest non-stationarity in the time series: The ACF plot (left) shows a very slow decay pattern with many significant lags well above the significance bounds (dashed blue lines) The PACF plot (right) also displays a gradual decay rather than a sharp cutoff

```r
#(c) Calculate and Interpret the Sample EACF
eacf(robot)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x o x  x  x  x
## 1 x o o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x x x x o o o o o o o  o  x  o
## 5 x x x o o o o o o o o  o  x  o
## 6 x o o o o x o o o o o  o  o  o
## 7 x o o x o x x o o o o  o  o  o
```

Interpretation: 1. AR and MA model identification: Based on the EACF matrix, it seems that the time series data may have both AR (auto-regressive) and MA (moving average) components, with the possibility of an AR(1) or AR(2) process with significant MA terms up to lag 7 or lag 9. 2. The presence of significant values in the EACF matrix at various lags suggests that an ARMA model (AutoRegressive Moving Average) might be suitable for this time series, with AR(1) or AR(2) and MA components up to around lag 7 or lag 9. 3. You could consider fitting an ARMA(1,7) or ARMA(2,7) model based on these results, but further analysis and model diagnostics (e.g., AIC, BIC, residual analysis) would be required to finalize the best model.

6.37 Calculate and interpret the sample EACF for the logarithms of the Los Angeles rainfall series. The data are in the file named larain. Do the results confirm that the logs are white noise?

```r
data(larain)
larain_log <- log(larain)
eacf(larain_log)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o o o o o o o o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x x x x x o o o o o o  o  o  o
## 6 x x o o x o o o o o o  o  o  o
## 7 x o x o o o o o o o o  o  o  o
```

Interpretation: The EACF results for the logarithms of the Los Angeles rainfall series show several "x" marks, indicating significant autocorrelations at various lags. This suggests that the data are not white noise and likely exhibit some underlying structure, such as autoregressive or moving average processes.

8.4 Simulate an AR(1) model with n = 30 and = 0.5. (a) Fit the correctly specified AR(1) model and look at a time series plot of the residuals. Does the plot support the AR(1) specification? (b) Display a normal quantile-quantile plot of the standardized residuals. Does the plot support the AR(1) specification? (c) Display the sample ACF of the residuals. Does the plot support the AR(1) specification? (d) Calculate the Ljung-Box statistic summing to K = 8. Does this statistic support the AR(1) specification?

```r
set.seed(98)
n <- 30
phi <- 0.5
ar1_series <- arima.sim(n = n, list(ar = phi))
ar1_series
```

```
## Time Series:
## Start = 1
## End = 30
## Frequency = 1
##  [1]  0.639530469  0.717706616 -0.153779276  0.030389940  1.230724051
##  [6]  0.822628055 -1.503951556 -0.133439965  0.840055289 -0.383344705
## [11]  0.329427367  0.391758692 -1.316061513 -2.586903492 -2.611176289
## [16] -1.098189209 -0.756828827 -0.332732826 -0.322327418 -2.079278032
## [21] -2.225099324 -2.178668846  0.003712702 -1.127752547 -0.651538336
## [26] -2.814606735 -0.939342519  0.307306788 -0.283608413 -0.617251959
```

```r
fit_ar1 <- arima(ar1_series, order = c(1, 0, 0))
fit_ar1
```

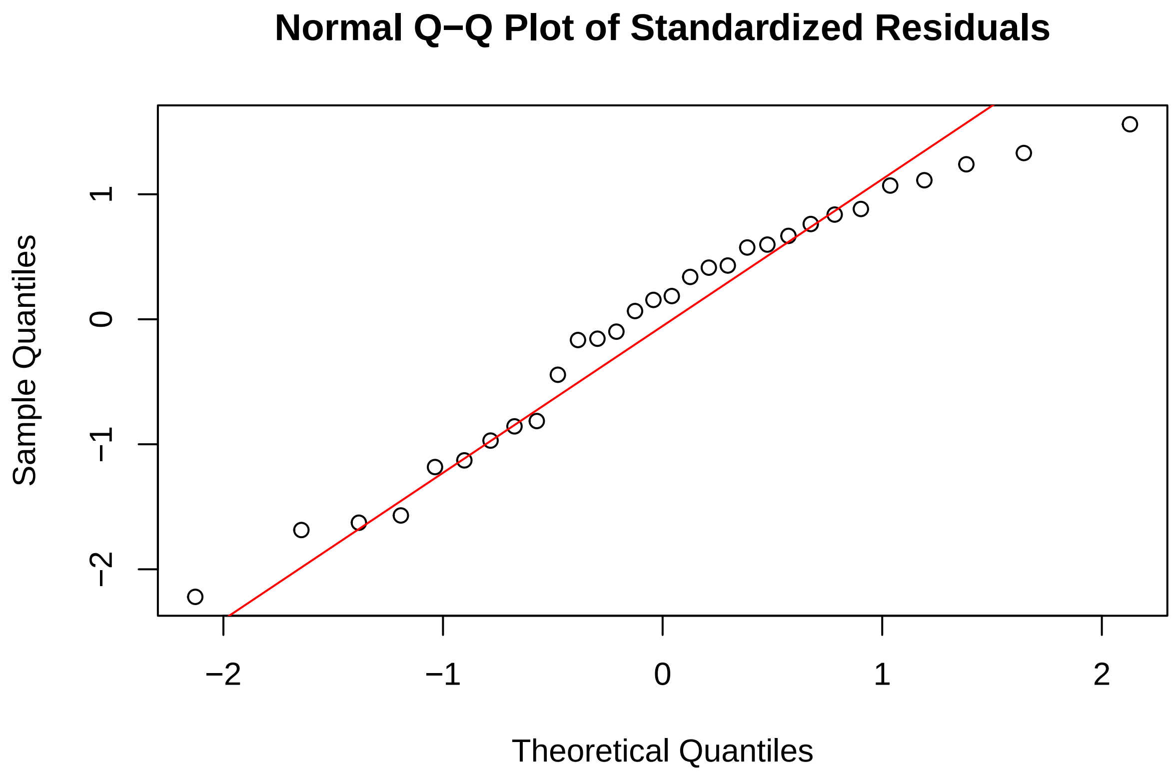
```
##
```

```
## Call:
## arima(x = ar1_series, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.4525    -0.5934
## s.e.  0.1618     0.3216
##
## sigma^2 estimated as 0.9769:  log likelihood = -42.33,  aic = 88.67
```

```r
par(mfrow= c(1, 1))
plot(residuals(fit_ar1), main = "Time Series Plot of Residuals")
abline(h = 0, col = "red", lty = 2)
```

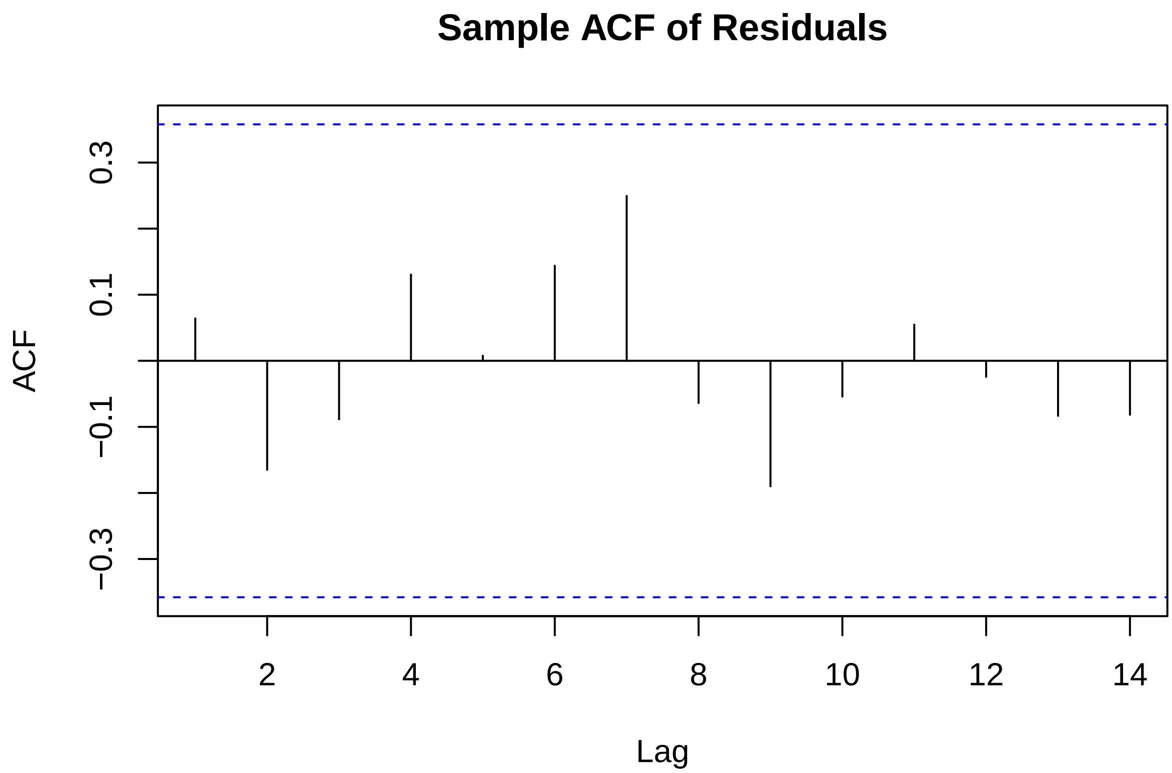
The time series plot shows residuals from an AR(1) model that fluctuate around zero (shown by the red dashed line). The residuals appear to move randomly up and down without any clear pattern, ranging approximately between -2 and 1.5. This random behavior suggests the AR(1) model is a good fit for the data since there's no obvious systematic pattern left in the residuals.

```r
# b) normal quantile-quantile plot of the standardized residuals
qqnorm(rstandard(fit_ar1), main = "Normal Q-Q Plot of Standardized Residuals")
qqline(rstandard(fit_ar1), col = "red")
```



The Q-Q plot shows points following the red diagonal line quite closely, with only slight deviations at the extreme ends.

```r
# c) sample ACF of the residuals
acf(residuals(fit_ar1), main = "Sample ACF of Residuals")
```



Conclusion The ACF plot of residuals strongly supports the AR(1) specification as it demonstrates that: No significant autocorrelation remains in the residuals The model has effectively captured the first-order autoregressive structure The residuals exhibit characteristics of white noise This indicates that an AR(1) model is an appropriate choice for this time series data.

```r
# d) Ljung-Box statistic
Box.test(residuals(fit_ar1), lag = 8, type = "Ljung-Box")
```

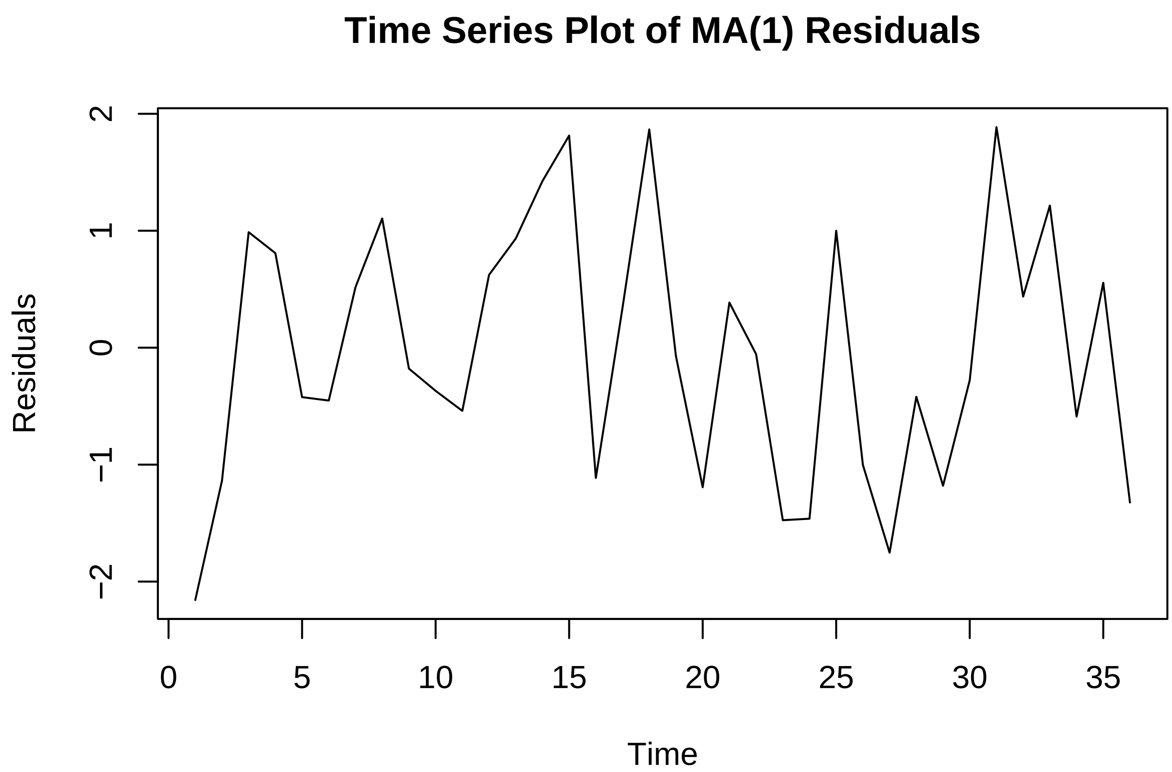
```
##
##  Box-Ljung test
##
## data:  residuals(fit_ar1)
## X-squared = 5.5871, df = 8, p-value = 0.6934
```

p-value = 0.6934 p-value is greater than 0.05, it suggests that there is no significant autocorrelation, and the AR(1) specification is supported.

8.5 Simulate an MA(1) model with n = 36 and = −0.5. (a) Fit the correctly specified MA(1) model and look at a time series plot of the residuals. Does the plot support the MA(1) specification? (b) Display a normal quantile-quantile plot of the standardized residuals. Does the plot support the MA(1) specification? (c) Display the sample ACF of the residuals. Does the plot support the MA(1) specification? (d) Calculate the Ljung-Box statistic summing to K = 6. Does this statistic support the MA(1) specification?

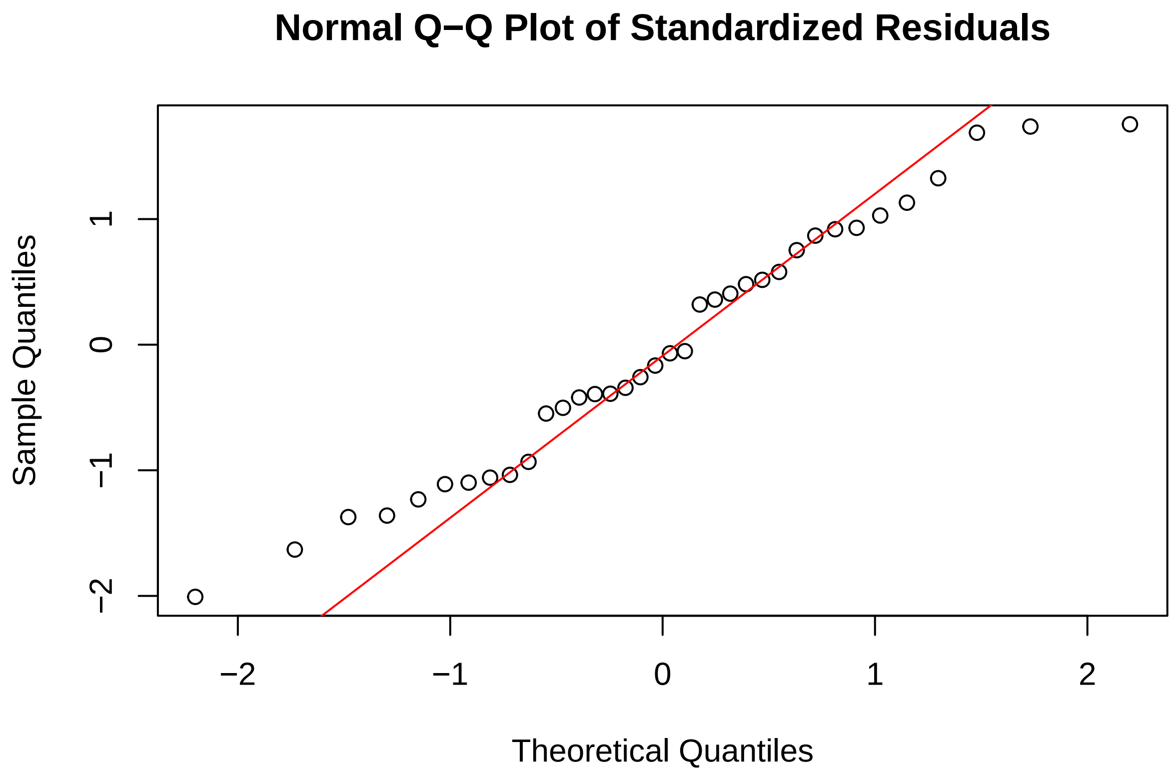
```r
n <- 36
theta <- -0.5
ma1_series <- arima.sim(n = n, list(ma = theta))
fit_ma1 <- arima(ma1_series, order = c(0, 0, 1))
```

```r
residuals_ma1 <- residuals(fit_ma1)
plot(residuals_ma1, type = "l", main = "Time Series Plot of MA(1) Residuals", ylab = "Residuals", xlab =
```



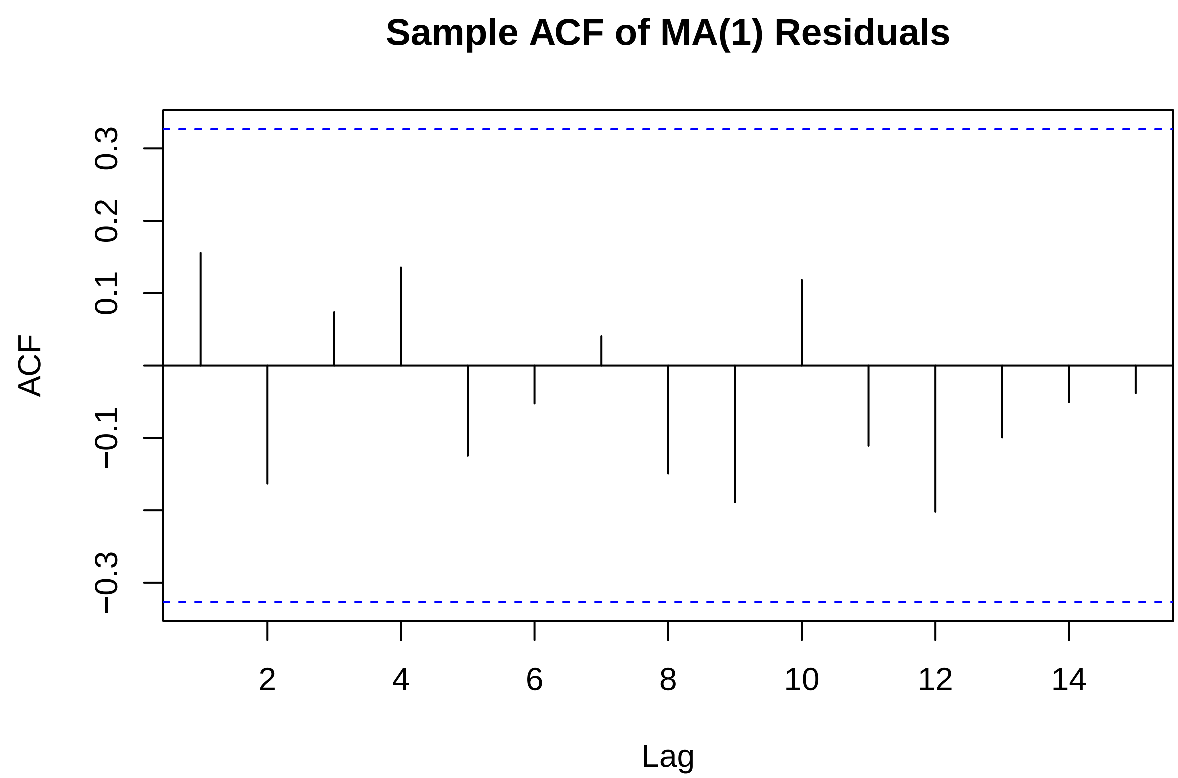
The residuals appear to fluctuate randomly around zero There is no obvious pattern or systematic behavior The variation appears relatively constant throughout the series No clear trend or seasonality is visible

```r
qqnorm(rstandard(fit_ma1), main = "Normal Q-Q Plot of Standardized Residuals")
qqline(rstandard(fit_ma1), col = "red")
```



The residuals lie on or near the red line, it suggests that the residuals follow a normal distribution, which supports the MA(1) specification.

```
acf(residuals_ma1, main = "Sample ACF of MA(1) Residuals")
```



Concept: the sample ACF should show a significant spike at lag 1, and all subsequent lags should show no significant autocorrelation (close to zero). This behavior would support the MA(1) model specification.

But our plot is randomly scattered and no significat spike at lag 1 hence it will not support MA(1) model

```r
# (d) Ljung-Box Statistic Summing to K = 6
Box.test(residuals_ma1, lag = 6, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals_ma1
## X-squared = 3.8396, df = 6, p-value = 0.6984
```

p-value = 0.7874

The p-value is large (greater than 0.05), it suggests no significant autocorrelation, which supports the MA(1) model.