Practial Computer exam 2

Mason Wong

23rd May 2022

R version and setup

The R version used is `4.1.01`.

The following function is used to recreate vectors:

```
recreate_vector = function(X, numeric = TRUE, quiet = FALSE) {
    X = scan(what = character(), text = X, quiet = quiet)
    X = sub("^\\s*\\[\\d+\\]", "", X)
    X = X[X != ""]
    if (numeric)
        X <- as.numeric(X)
    X
}
```

Question 1

We firstly read in the data:

```
# the data in text form:
textual_data = "[1] 12.43 11.51 10.04 11.21 11.56 11.20 11.11 10.92 10.40  9.25  9.85 12.97 13.71 14.04
 [19] 14.31 14.20 13.63 15.64 14.83 12.26 11.51 11.29 12.07 13.80 15.53 14.66 14.54 17.25 16.37 14.43 12
 [37] 10.98 11.70 12.99 14.08 13.75 11.78 12.05 13.55 14.72 13.44 14.56 16.17 14.84 13.32 13.75 15.12 16
 [55] 13.33 14.93 16.20 15.00 13.32 12.45 12.51 12.89 13.66 13.42 15.02 15.48 15.11 13.96 13.93 14.42 14
 [73] 14.09 12.96 13.66 14.14 13.24 14.33 15.92 13.96 12.52 11.43 11.77 14.91 15.44 15.48 15.08 16.23 15
 [91] 12.52 13.19 14.32 14.90 15.13 13.65 12.88 11.65 11.13 12.04 14.72 15.22 14.61 14.35 14.41 15.49 14
[109] 12.83 11.50 11.02 11.24 11.26 13.54 15.57 15.21 14.34 12.21 11.17 13.65"

data = recreate_vector(textual_data)
length(data)
```

```
## [1] 120
```

(i) We fir an $MA(2)$ model to this data and write down the corresponding parameter estimates with their corresponding standard errros:

```
fit_ma2 = arima(data, order = c(0, 0, 2))
fit_ma2
```

```
##
## Call:
## arima(x = data, order = c(0, 0, 2))
##
## Coefficients:
```

```
##          ma1     ma2  intercept
##       1.1280  0.4201    13.5066
## s.e.  0.0833  0.0765     0.2269
##
## sigma^2 estimated as 0.964:  log likelihood = -168.77,  aic = 345.54
```

The parameter estimates with standard errors are:

- $intercept = 13.5066$ with $se = 0.2269$
- $ma1 = 1.128$ with $se = 0.083$
- $ma2 = 0.4201$ with $se = 0.0765$

(ii) We find the next 5 predicted values for this series

```
forecast = predict(fit_ma2, n.ahead = 5, se.fit = TRUE)
pred = forecast$pred
pred
```

```
## Time Series:
## Start = 121
## End = 125
## Frequency = 1
## [1] 14.93285 14.12289 13.50655 13.50655 13.50655
```

Hence the next 5 predicted values for this series are:

- 14.93285, 14.12289, 13.50655, 13.50655, 13.50655

Question 2

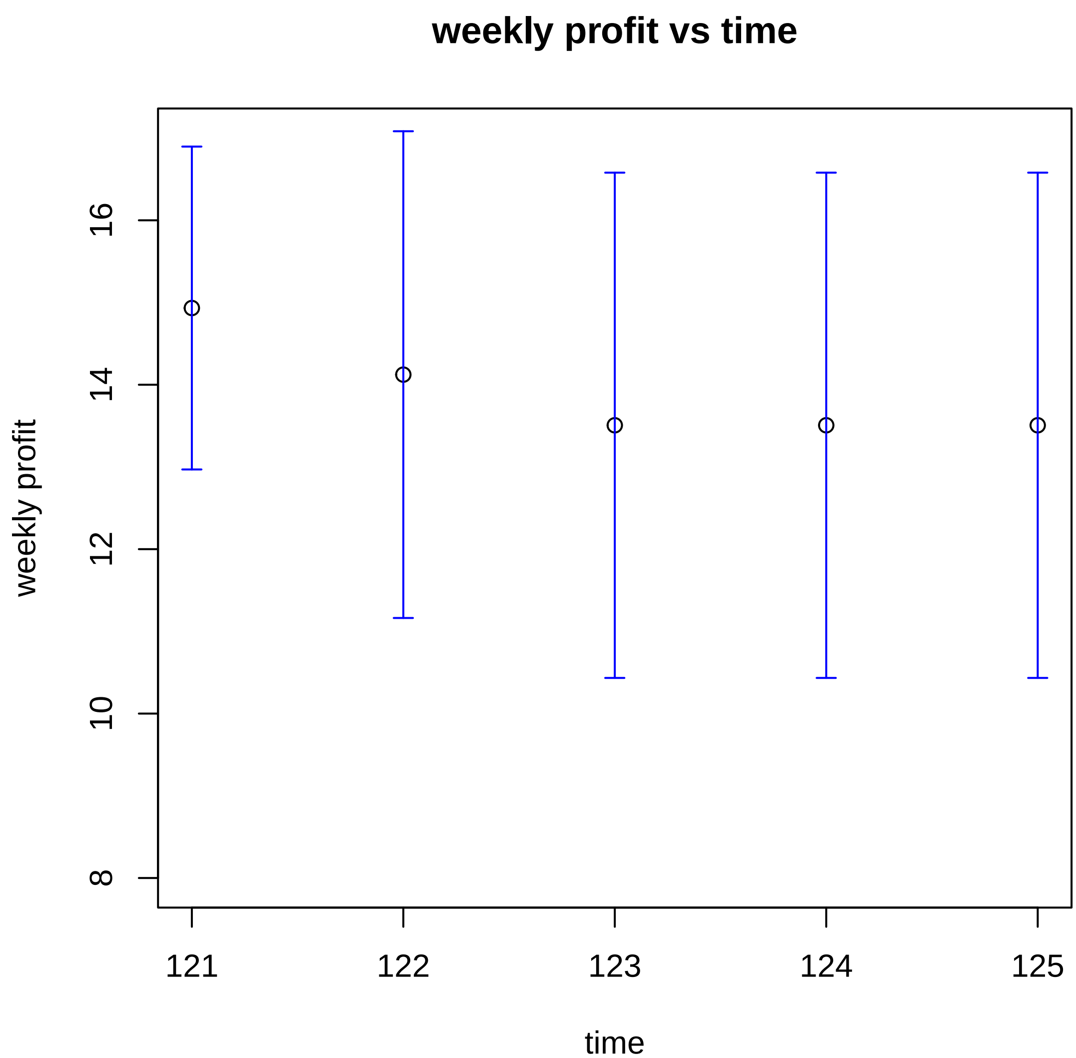
(i) To find 95% prediction intervals for the 5 predicted values in Q1 (ii) we recall that prediction intervals are given by $prediction \pm 2 * standard\ error$. Hence, since the standard error are given by:

```
stderrors = forecast$se
stderrors
```

```
## Time Series:
## Start = 121
## End = 125
## Frequency = 1
## [1] 0.9818537 1.4801163 1.5365191 1.5365191 1.5365191
```

- prediction interval: 14.93285 $\pm 2 * 0.9818537$
- prediction interval: 14.12289 $\pm 2 * 1.4801163$
- prediction interval: 13.50655 $\pm 2 * 1.5365191$
- prediction interval: 13.50655 $\pm 2 * 1.5365191$
- prediction interval: 13.50655 $\pm 2 * 1.5365191$

```
plot(121:125, pred, ylim = c(8, 17), xlab = "time", ylab = "weekly profit", main = "weekly profit vs ti
arrows(x0 = 121:125, y0 = pred - 2 * stderrors, x1 = 121:125, y1 = pred + 2 * stderrors,
    code = 3, angle = 90, length = 0.05, col = "blue")
legend(x = 121, y = 0.06, "prediction intervals", cex = 0.8, col = "blue", pch = c(1,
    1))
```



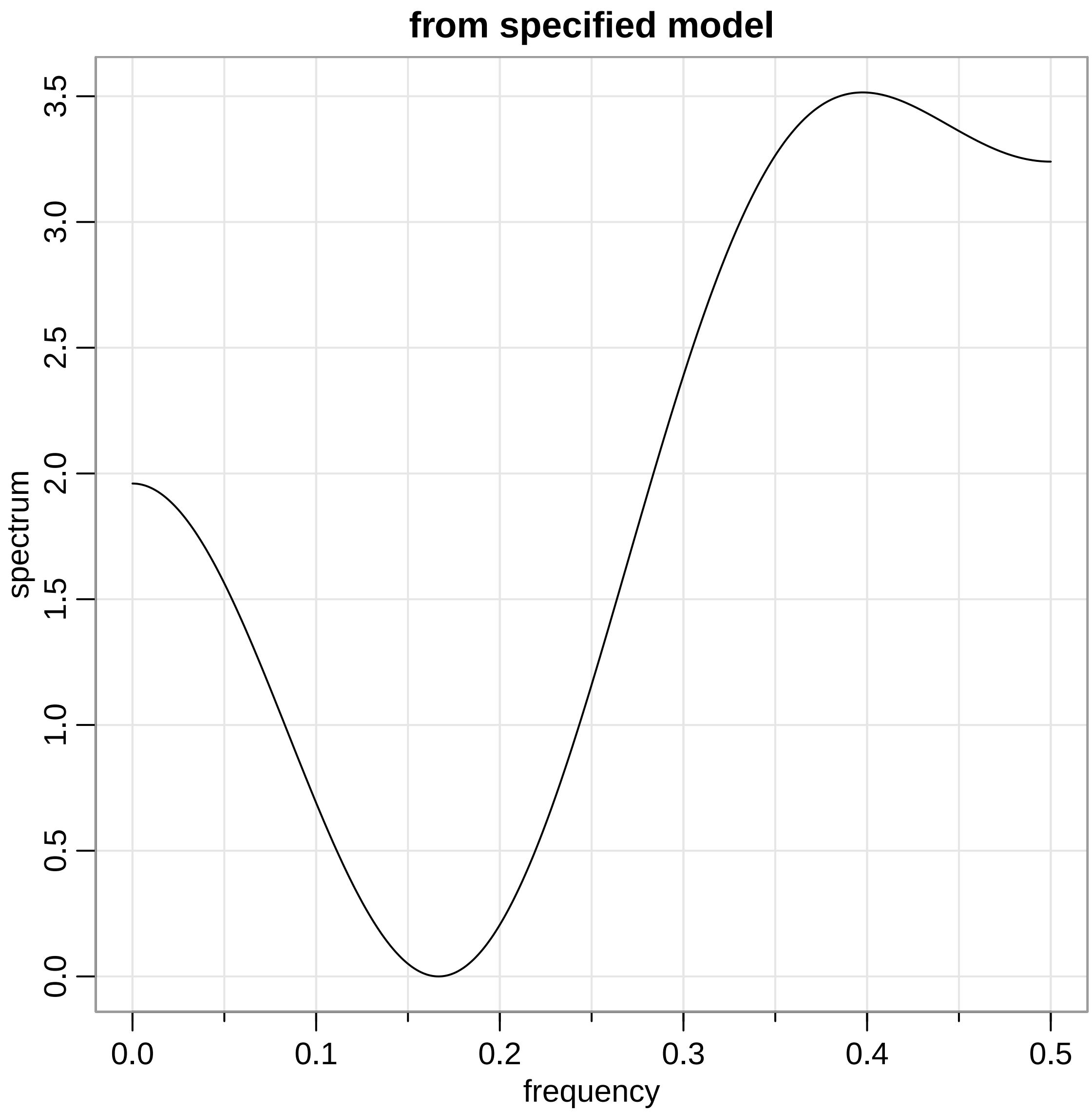
Question 3

(i) We draw the theoretical spectrum of

$$X_t = Z_t - 0.6Z_{t-1} + 0.6Z_{t-2} + 0.4Z_{t-3}$$

where $Z_t \sim NID(0, 1)$

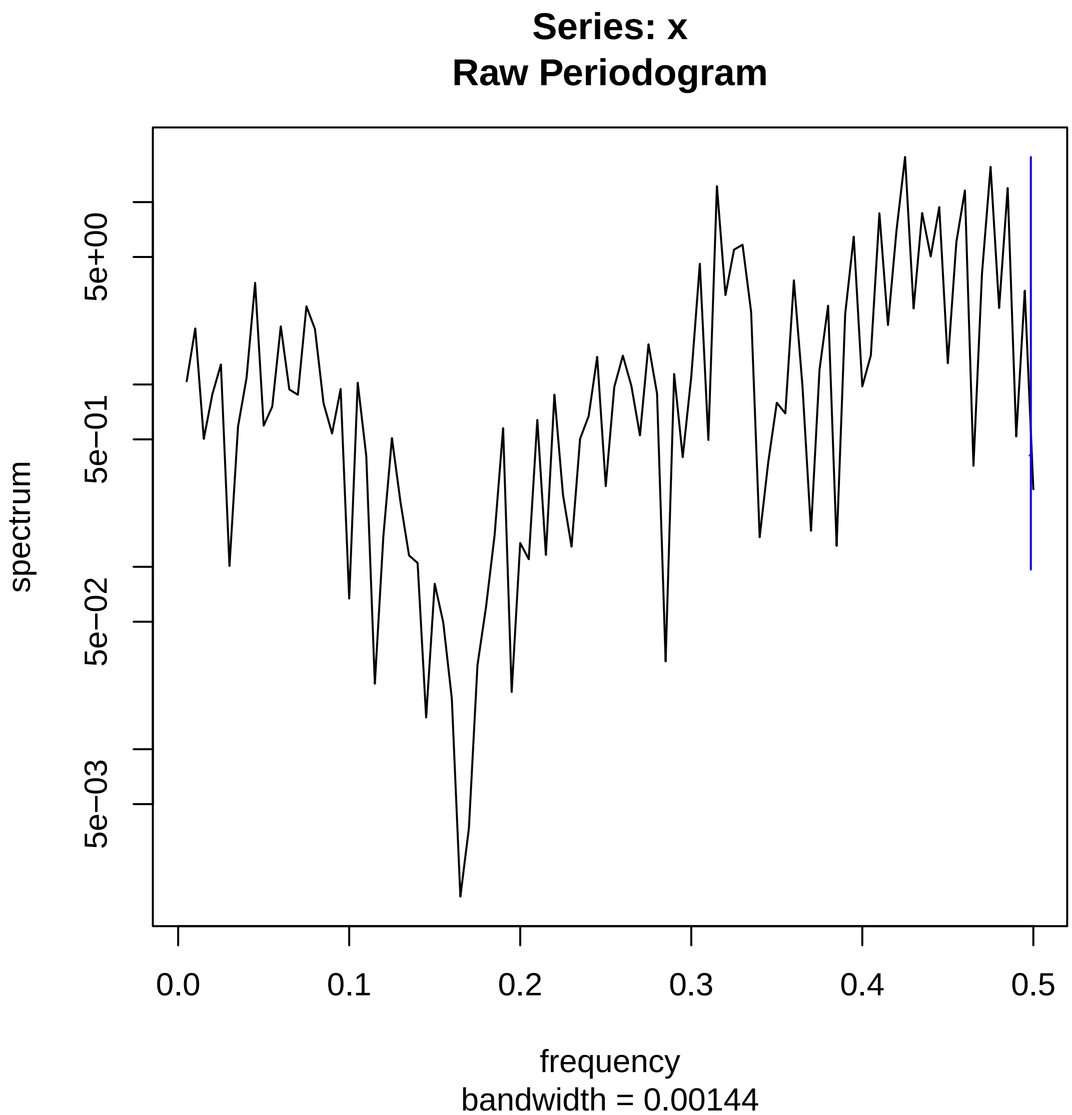
```
library(astsa)
arma.spec(ma = c(-0.6, 0.6, 0.4))
```



(ii) The spectrum is approximately minimised at 0.16. An appropriate interval to look for the minimum would be the open interval $(1.5, 2)$

Question 4

```
set.seed(341)
d4 = arima.sim(list(ma = c(-0.6, 0.6, 0.4)), n = 600)[401:600]
spectrum(d4)
```



The shape of the sample periodogram looks very close to the theoretical periodogram. In face, the frequency for which the minimum occurs in the sample periodogram is also very close to the frequency for which the minimum occurs in the theoretical periodogram.

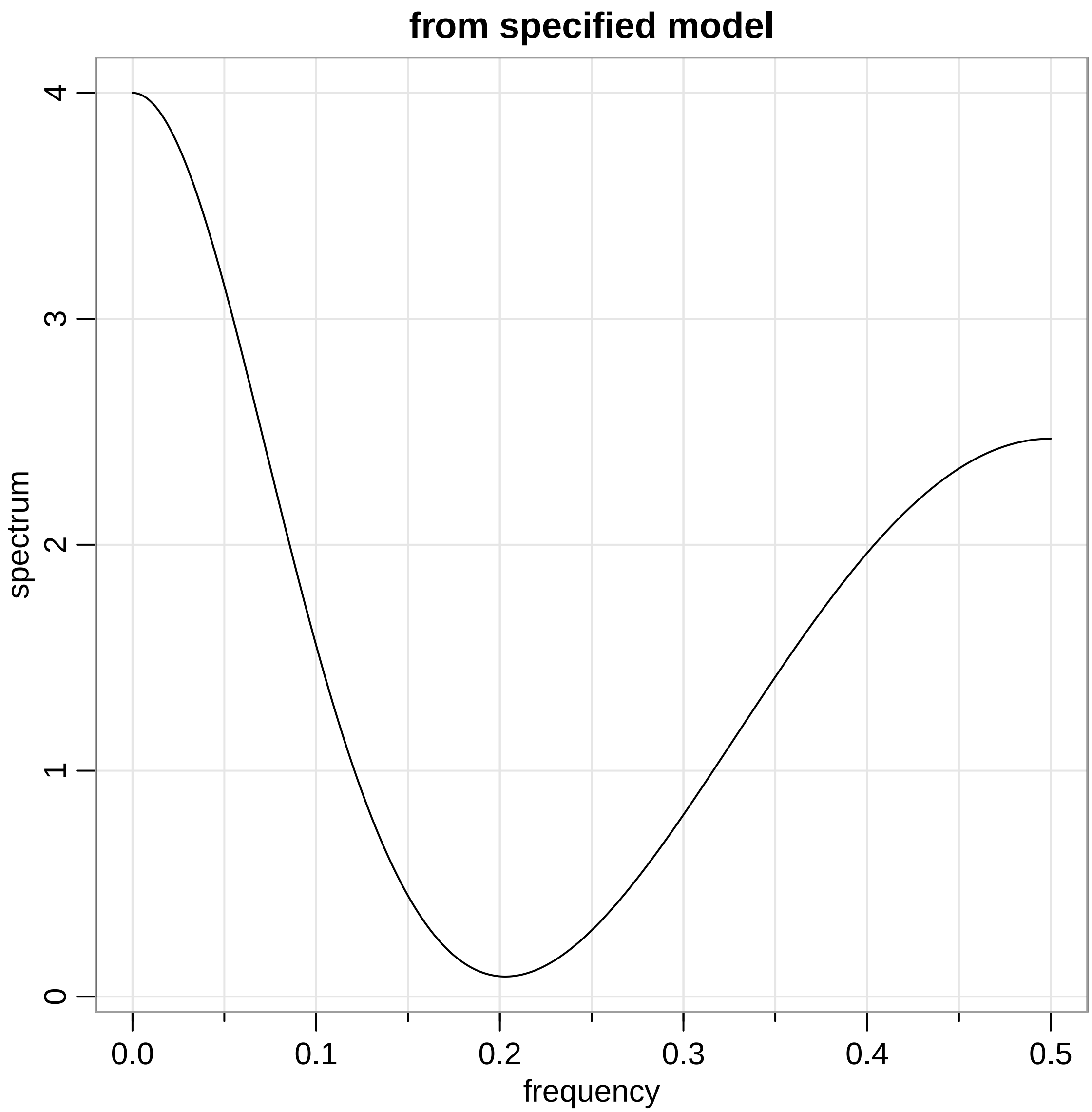
Question 5

We want to see the theoretical spectrum of the $ARMA(1,2)$ process givbn by:

$$X_t = 10 + 0.4X_{t-1} + Z_t - 0.5Z_{t-1} + 0.7Z_{t-2}$$

where $Z_t \sim NID(0,1)$ (i) We draw the theoretical spectrum of the process

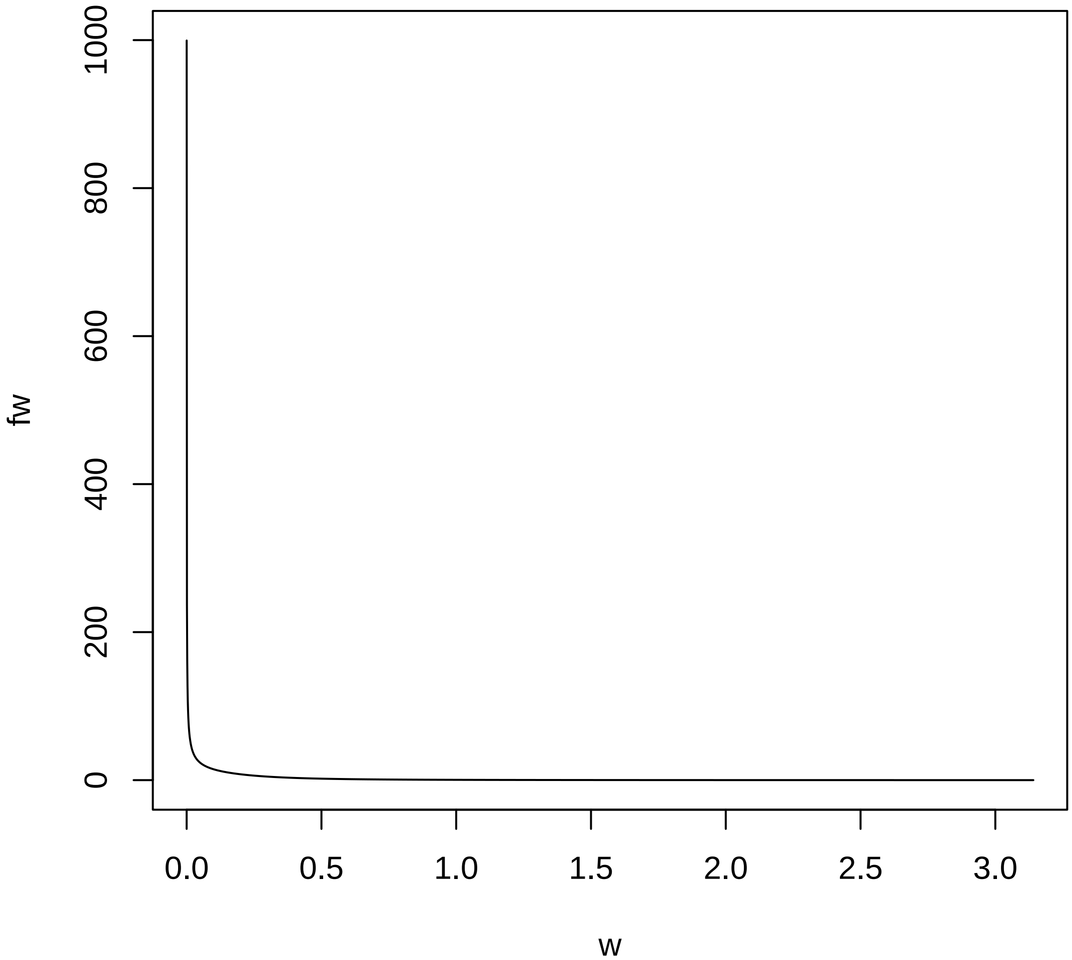
```r
arma.spec(ar = 0.4, ma = c(-0.5, 0.7))
```



- frequency of approximate maximum value(s) = 0 and 0.5
- frequency of approximate minimum value = 0.2
- points of inflexion occur at frequencies of 0.06 and 0.35

Question 6

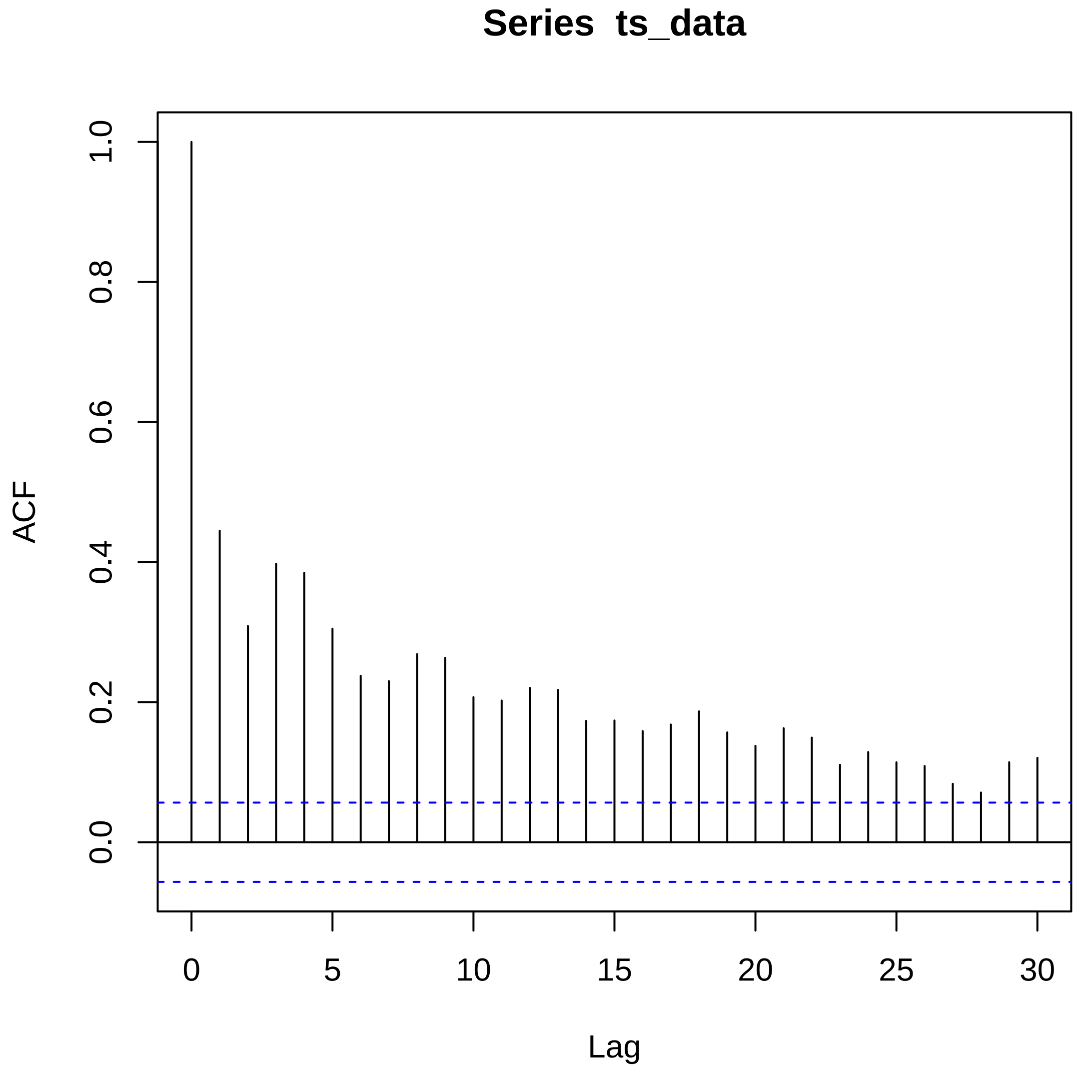
```r
w = seq(1e-04, pi, by = 0.001)
fw = 1/(2 * pi) * (1.25 + cos(w))/(1.49 - 1.4 * cos(w)) * 1/(2 * sin(w/2))^(0.6)
plot(w, fw, type = "l")
```



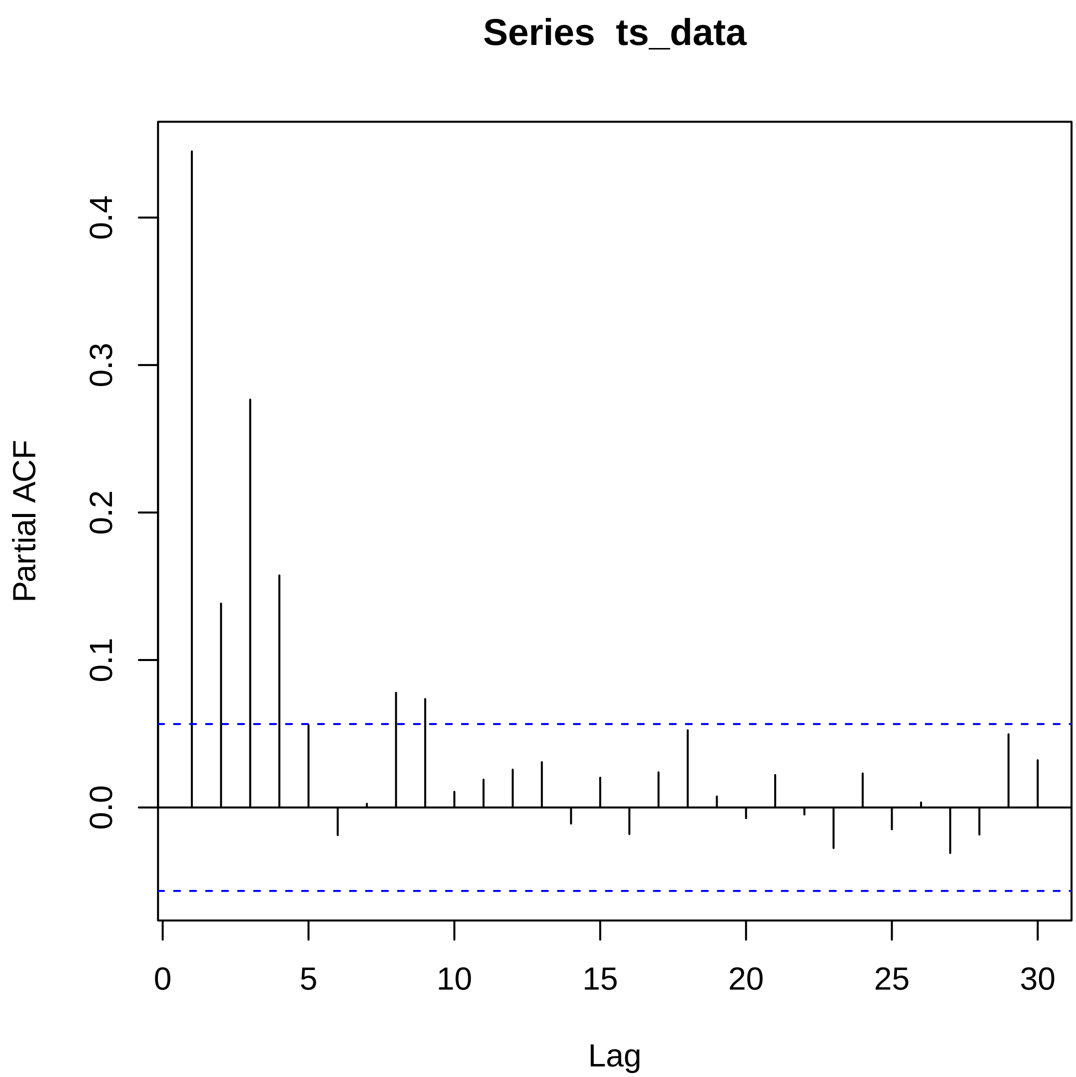
The behaviour near the origin makes it look like it's unbounded and has a large value near the origin. Hence this time series has long memory

Question 7

```
library(fracdiff)
set.seed(1205)
arfima_data = fracdiff.sim(n = 1200, ar = c(0.3, -0.7), ma = c(0.4, -0.6), d = 0.4)
ts_data = arfima_data$series
acf_vals = acf(ts_data)
```




```
pacf_vals = pacf(ts_data)
```



(i) The acf and pacf values are provided below

```
acf_vals[1:5]
```

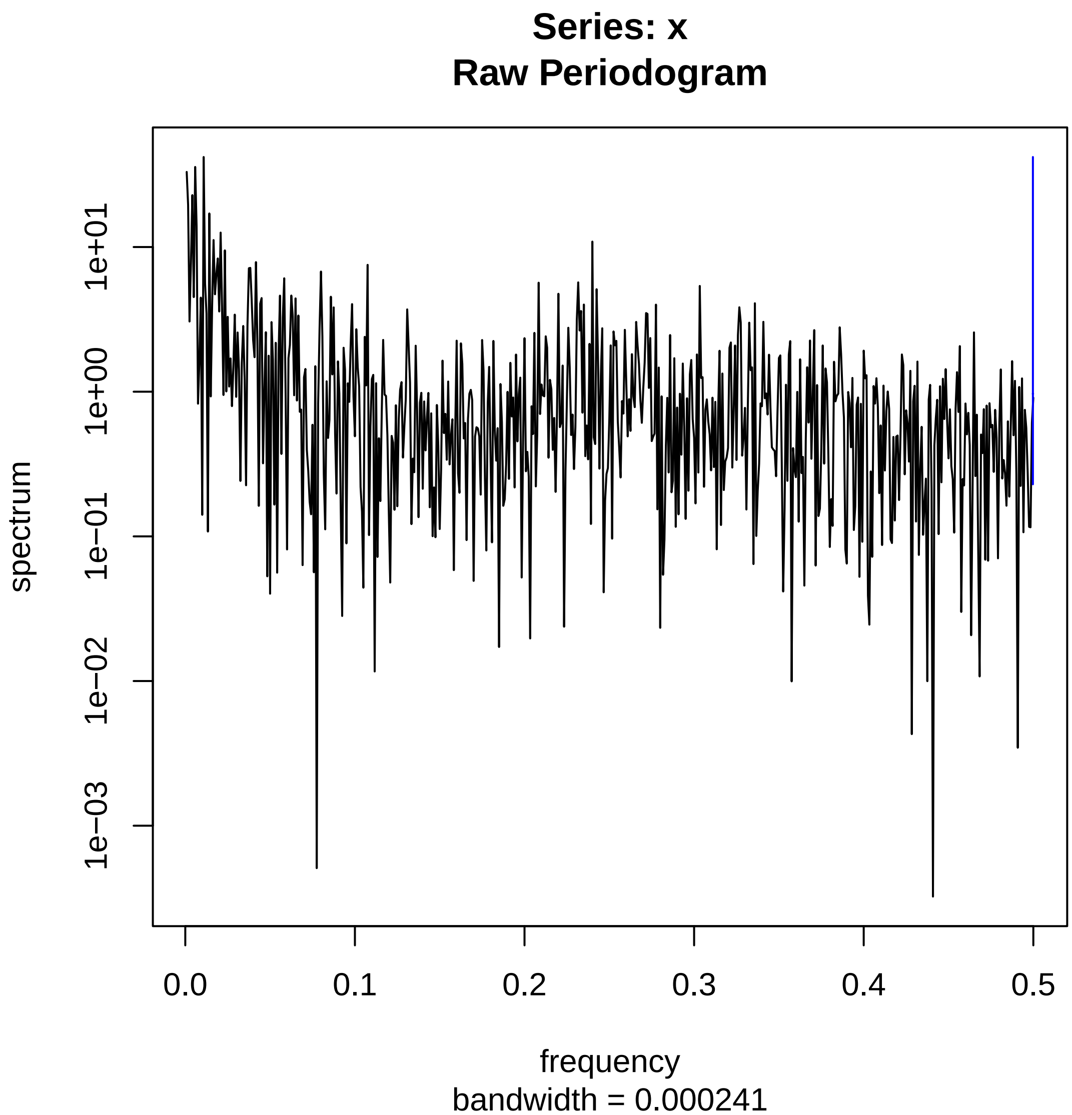
```
##
## Autocorrelations of series 'ts_data', by lag
##
##     1     2     3     4     5
## 0.445 0.309 0.398 0.385 0.305
```

```
pacf_vals[1:5]
```

```
##
## Partial autocorrelations of series 'ts_data', by lag
##
##     1     2     3     4     5
## 0.445 0.138 0.277 0.157 0.056
```

(ii) The first 5 values of the sample periodogram are:

```
spec_values = spectrum(ts_data)
```




```
spec_values$spec[1:5]
```

```
## [1] 33.056906 19.534119  3.057006  7.867935 22.850896
```

Question 8

```
set.seed(8134)
arfima_data2 = fracdiff.sim(n = 1500, ar = 0.4, ma = -0.6, d = 0.4)
d8 = arfima_data2$series[301:1500]
fit = fracdiff(d8, nar = 1, nma = 1)
fit
```

```
##
## Call:
##   fracdiff(x = d8, nar = 1, nma = 1)
##
## Coefficients:
##           d          ar          ma
```

```
##  0.4029887  0.4092010 -0.5737011
## sigma[eps] = 0.9915727
## a list with components:
##  [1] "log.likelihood"  "n"               "msg"             "d"
##  [5] "ar"              "ma"              "covariance.dpq"  "fnormMin"
##  [9] "sigma"           "stderror.dpq"    "correlation.dpq" "h"
## [13] "d.tol"           "M"               "hessian.dpq"     "length.w"
## [17] "residuals"       "fitted"          "call"
```

```
fit$stderror.dpq
```

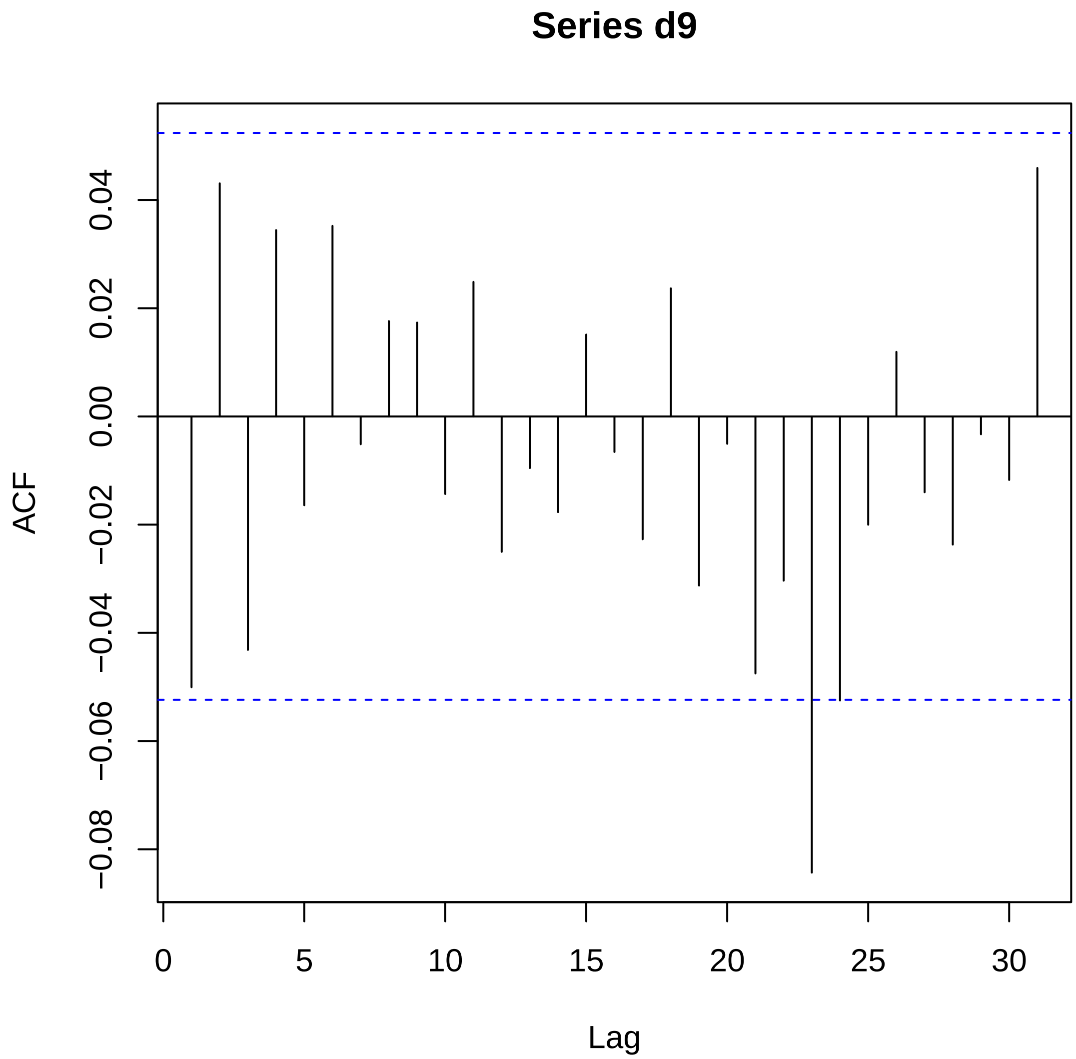
```
## [1] 0.01044004 0.03529640 0.03089261
```

- d = 0.4029887 with std error 0.01044004
- ar = 0.4092010 with std errror 0.03529640
- ma = -0.5737011 with std error 0.03089261

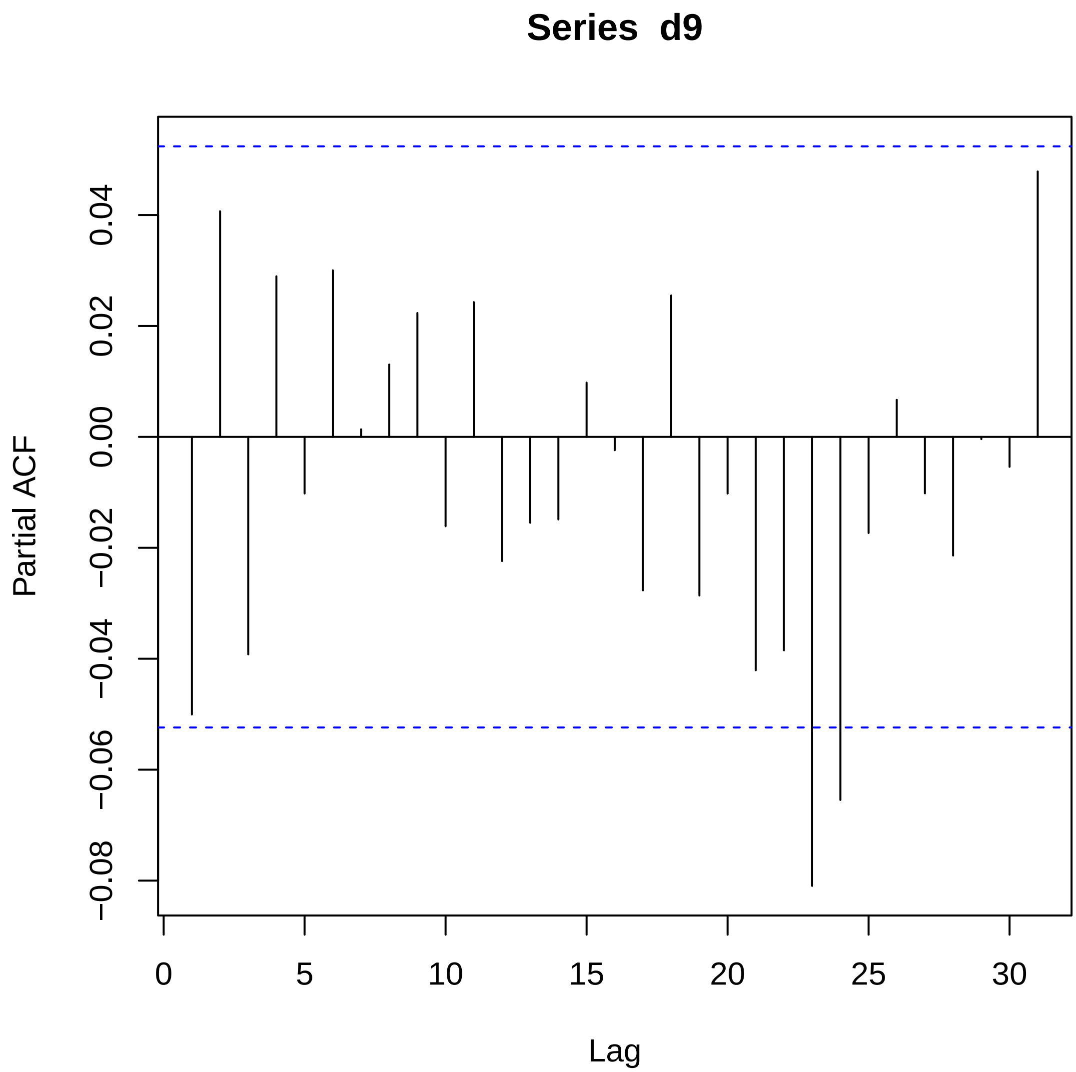
Question 9

(i) We write down the first acf and pacf values of d9

```
set.seed(1456)
library(TSA)
simulate_vals = garch.sim(alpha = c(0.58, 0.3), beta = 0.4, n = 1800)
d9 = simulate_vals[401:1800]
acf_vals = acf(d9)
```




```
pacf_vals = pacf(d9)
```




```r
acf_vals[1:3]
```

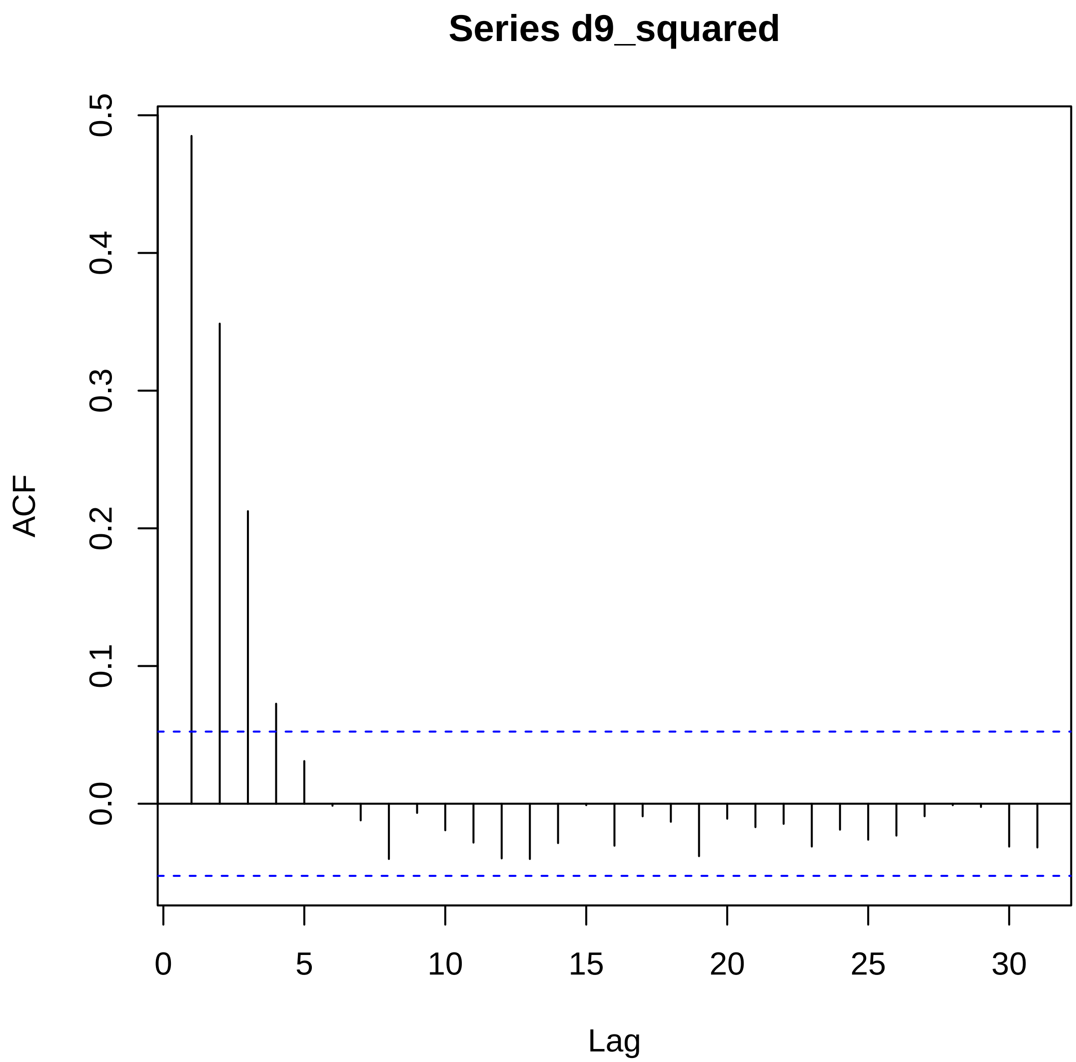
```
##
## Autocorrelations of series 'd9', by lag
##
##      1      2      3
## -0.050  0.043 -0.043
```

```r
pacf_vals[1:3]
```

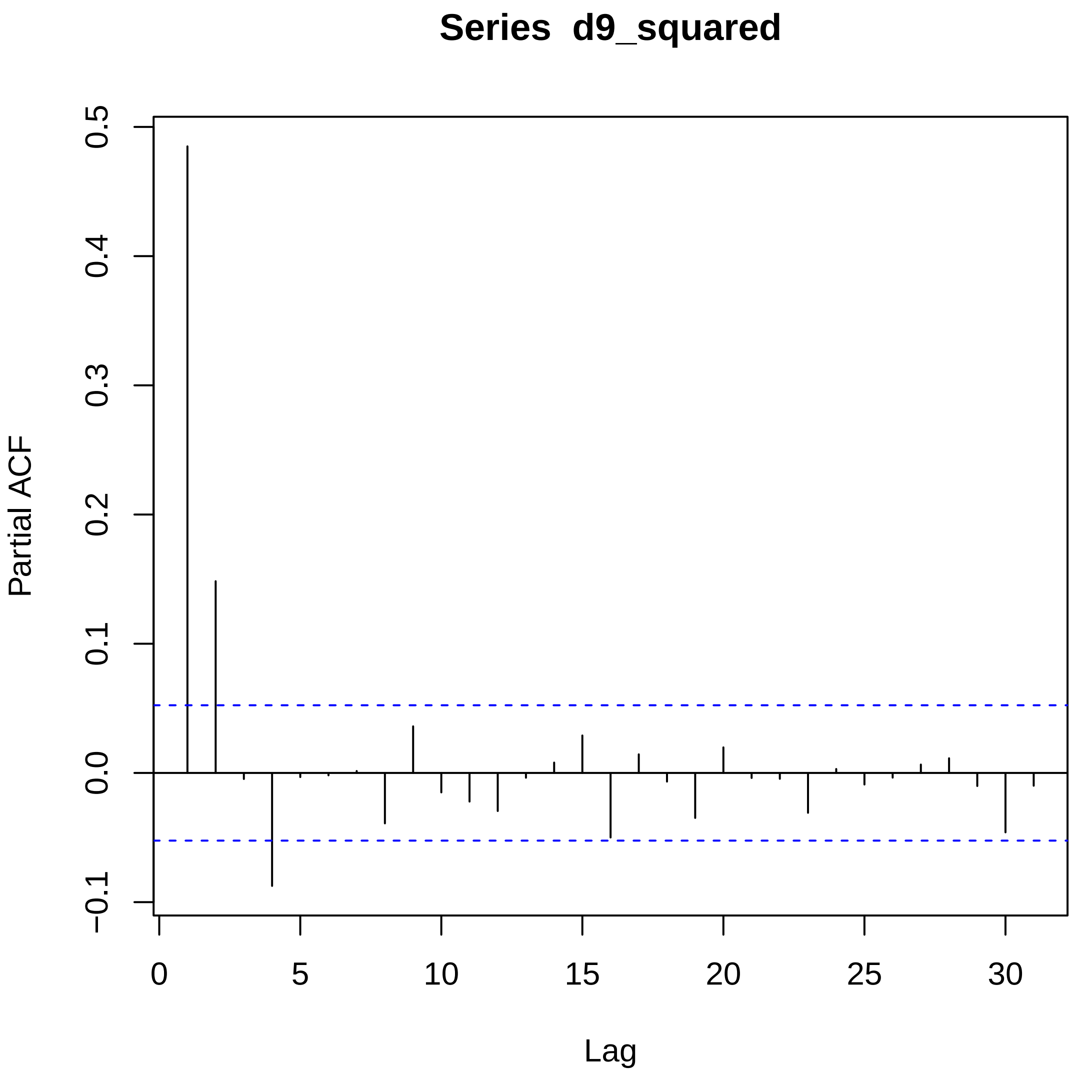
```
##
## Partial autocorrelations of series 'd9', by lag
##
##      1      2      3
## -0.050  0.041 -0.039
```

(ii)

```
d9_squared = d9^2
acf(d9_squared)
```




```
pacf(d9_squared)
```



The data seems more correlated for the squared values compared to the normal values.

Question 10

Not enough time