Problem Set 2, Econ 211C

Swati Sharma

Question 1

Suppose $\{Y_i\}_{i=1}^{n} = pX_1 + (1-p)X_2$ where $p = \frac{1}{3}$, $X_1 \sim N(\mu_1, 1)$, and $X_1 \sim N(\mu_2, 1)$. That is, the random variables $\{Y_i\}$ are drawn from a discrete mixture of Normals.

a. (10 points)

Write the joint density function for $\{Y_i\}$.

Solution:

Evaluating I_t: $E[I_t] = \frac{1}{3}(1) + \frac{2}{3}(0) = \frac{1}{3}$

$$f(Y_i) = \frac{1}{3}f(x_{1i}) + \frac{2}{3}f(x_{2i})$$

$$f(x_1; \mu_1, 1) = \frac{1}{\sqrt{2\pi}}e^{-\frac{(x_1-\mu_1^2)}{2}}$$

$$f(x_2; \mu_2, 1) = \frac{1}{\sqrt{2\pi}}e^{-\frac{(x_2-\mu_2^2)}{2}}$$

$$f(y_i) = \frac{1}{3}(\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_{1i}-\mu_1^2)}{2}}) + \frac{2}{3}(\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_{2i}-\mu_2^2)}{2}})$$

Assuming independence,

$$f(\boldsymbol{y}_T) = \prod_{i=1}^{T} f(Y_i)$$

$$f(\boldsymbol{y}_T) = \prod_{i=1}^{T}[\frac{1}{3}(\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_{1i}-\mu_1^2)}{2}}) + \frac{2}{3}(\frac{1}{\sqrt{2\pi}}e^{-\frac{(x_{2i}-\mu_2^2)}{2}})]$$

$$f(\boldsymbol{y}_T) = (\frac{1}{3\sqrt{2\pi}})^T \prod_{i=1}^{T}[e^{-\frac{(x_{1i}-\mu_1^2)}{2}} + 2e^{-\frac{(x_{2i}-\mu_2^2)}{2}}]$$

b. (10 points)

What is the log likelihood?

Solution:

$$\mathcal{L}(\boldsymbol{y}_T) = f(\boldsymbol{y}_T)$$

$$l(\boldsymbol{y}_T) = log(\mathcal{L}(\boldsymbol{y}_T)) = log((\frac{1}{3\sqrt{2\pi}})^T \prod_{i=1}^{T} [e^{-\frac{(x_{1i}-\mu_1^2)}{2}} + 2e^{-\frac{(x_{2i}-\mu_2^2)}{2}}]) = -Tlog(3) - \frac{T}{2}log(2\pi) + \sum_{i=1}^{T} log(e^{-\frac{(x_{1i}-\mu_1^2)}{2}} + 2e^{-\frac{(x_{2i}-\mu_2^2)}{2}})$$

c. (15 points)

What are the gradient and hessian of the log likelihood?

Solution:

The gradient:

$$\boldsymbol{g}(\boldsymbol{\mu}) = \nabla l(\boldsymbol{\mu}) = [\frac{\partial l}{\partial \mu_1}, \frac{\partial l}{\partial \mu_2}]$$

The hessian:

$$\boldsymbol{H}(\boldsymbol{\mu}) = \nabla^2 l(\boldsymbol{\mu}) = \begin{pmatrix} \frac{\partial^2 l}{\partial \mu_1^2} & \frac{\partial^2 l}{\partial \mu_1 \mu_2} \\ \frac{\partial^2 l}{\partial \mu_1 \mu_2} & \frac{\partial^2 l}{\partial \mu_2^2} \end{pmatrix}$$

Where

$$\frac{\partial l}{\partial \mu_1} = \sum_{i=1}^{T} (\frac{(x_{1i}-\mu_1)e^{-\frac{1}{2}(x_{1i}-\mu_1)^2}}{e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} + 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2}})$$

$$\frac{\partial l}{\partial \mu_2} = \sum_{i=1}^{T} (\frac{(x_{2i}-\mu_2)e^{-\frac{1}{2}(x_{2i}-\mu_2)^2}}{e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} + 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2}})$$

$$\frac{\partial^2 l}{\partial \mu_1 \mu_2} = -T[\frac{2(x_{1i}-\mu_1)(x_{2i}-\mu_2)e^{\frac{-1}{2}((x_{1i}-\mu_1)^2+(x_{2i}-\mu_2)^2)}}{(e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} + 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2})^2}]$$

$$\frac{\partial^2 l}{\partial \mu_1^2} = \frac{(x_{1i}-\mu_1)^2 e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} - e^{-\frac{1}{2}(x_{1i}-\mu_1)^2}}{e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} + 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2}} - \frac{(x_{1i}-\mu_1)^2 e^{-(x_{1i}-\mu_1)^2}}{(e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} + 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2})^2}$$

$$\frac{\partial^2 l}{\partial \mu_2^2} = T[\frac{(x_{2i}-\mu_2)^2 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2} - 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2}}{e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} + 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2}} - \frac{(x_{2i}-\mu_2)^2 4e^{-(x_{2i}-\mu_2)^2}}{(e^{-\frac{1}{2}(x_{1i}-\mu_1)^2} + 2e^{-\frac{1}{2}(x_{2i}-\mu_2)^2})^2}]$$

d. (15 points)

Using the simulated data on the course website, compute maximum likelihood estimates of μ_1 ane μ_2 using the Newton-Raphson method.

Solution:

NR <- function(init.value,zero=10^(-8)){

nn = 1 y = log(init.value)-4 x0 = init.value

while(abs(y)>zero){ y = log(x0)-4 dydx = 1/x0 x1 = x0 - y/dydx

```
nn = nn+1
if(nn>1000){
  break
}
x0 = x1
```

}

return(x1) }

```r
dta <- read.csv("https://people.ucsc.edu/~ealdrich/Teaching/Econ211C/Assignments/ps2Data.csv")
x <- as.numeric(as.character(unlist(dta)))

n <- length(x)
u1 <- mean(x) # starting values for u1, u2
u2 <- mean(x)
e <- exp(1)

g <- matrix(c((e^(.5*(-(x-u1)^2))*(x-u1))/(e^(.5*(-(x-u1)^2))+(2*e^(.5*(-(x-u2)^2)))), (2*e^(.5*(-(x-u2

H <- matrix(c((e^(.5*(-(x-u1)^2))*(((x-u1)^2)-1))/(e^(-.5*(x-u1)^2)+(2*e^(.5*(-(x-u2)^2)))) - (e^((-(x-
-n*(2*(x-u2)*(x-u1)*e^(((-(x-u2)^2)-(x-u1)^2)/2))/((2*e^((-(x-u2)^2)/2))+(e^((-(x-u1)^2)/2)))^2,
-n*(2*(x-u2)*(x-u1)*e^(((-(x-u2)^2)-(x-u1)^2)/2))/((2*e^((-(x-u2)^2)/2))+(e^((-(x-u1)^2)/2)))^2,
(2*(e^(-.5*(x-u2)^2)*(x-u2)^2)-2*e^(-.5*(x-u2)^2))/(2*e^(-.5*(x-u2)^2)+e^(-.5(x-u1)^2))-(4*e^(-(x-u2)^2

malgebra <- function(u1, u2){
  prod <- crossprod(H,g)
  return(prod)
}

# Newton - Raphson
for(i in 0:100000000) {
  new_u1 <- u1 - malgebra(u1,u2)
  new_u2 <- u2 - malgebra(u1,u2)
  if ((abs(new_u1 - u1) < .000001) & (abs(new_u2 - u2) < .000001)) {
    cat("u1:", new_u1, "u2:", new_u2, "\n")
    cat("old u1:", u1, "\n")
    break
  }
  u1 = new_u1
  u2 = new_u2
}
```

Could not get the hessian matrix to compile - its some arithmetic mistake but I cannot find it. The rest is working but I did not get estimates for μ_1 and μ_2 because of the hessian.

Question 2

Assume the following linear model:

$$Y = \beta_1 X_1 + \beta_2 X_2 + \varepsilon, \;\; \varepsilon \sim N(0, \sigma^2). \tag{1}$$

a. (10 points)

Write the expression for the best linear predictor, $\boldsymbol{\beta}^* = (\beta_1, \beta_2)$, with each element of the associated matrices and vectors individually specified.

Solution:

$$\boldsymbol{\beta^*} = E[\boldsymbol{X}_t \boldsymbol{X}_t^{'}]^{-1} E[\boldsymbol{X}_t \boldsymbol{Y}_{t+1}]$$

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = E\left[\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \begin{bmatrix} X_1 & X_2 \end{bmatrix} \right]^{-1} E\left[\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} [Y_{t+1}] \right]$$

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = E\begin{bmatrix} X_1^2 & X_1 X_2 \\ X_1 X_2 & X_2^2 \end{bmatrix}^{-1} E\begin{bmatrix} X_1 Y_{t+1} \\ X_2 Y_{t+1} \end{bmatrix}$$

Alternatively,

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = E\left[\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \begin{bmatrix} X_1 & X_2 \end{bmatrix} \right]^{-1} E\left[\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \begin{bmatrix} X_1 & X_2 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \epsilon \end{bmatrix} \right]$$

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = E\begin{bmatrix} X_1^2 & X_1 X_2 \\ X_1 X_2 & X_2^2 \end{bmatrix}^{-1} E\begin{bmatrix} X_1(\beta_1 X_1 + \beta_2 X_2 + \epsilon) \\ X_2(\beta_1 X_1 + \beta_2 X_2 + \epsilon) \end{bmatrix}$$

b. (10 points)

Suppose you have n obersvations of (Y, X_1, X_2), collected in vectors $\mathbf{y'} = (y_1, \ldots, y_n)$, $\mathbf{x}_1' = (x_{11}, \ldots, x_{1n})$, and$\mathbf{x}_2' = (x_{21}, \ldots, x_{2n})$. Write the expression for the least-squares estimator $\hat{\boldsymbol{\beta}}$, with each element of the associated matrices and vectors individually specified.

Solution:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^{'} \boldsymbol{X})^{-1} \boldsymbol{X}^{'} \boldsymbol{Y}$$

Where $\boldsymbol{X} = \begin{bmatrix} X_{11} & X_{21} \\ \cdot & \cdot \\ \cdot & \cdot \\ X_{1n} & X_{2n} \end{bmatrix}$.

$$\widehat{\boldsymbol{\beta}} = \left(\begin{bmatrix} X_{11}...X_{1n} \\ X_{21}...X_{2n} \end{bmatrix} \begin{bmatrix} X_{11} & X_{21} \\ . & . \\ . & . \\ X_{1n} & X_{2n} \end{bmatrix} \right)^{-1} \begin{bmatrix} X_{11}...X_{1n} \\ X_{21}...X_{2n} \end{bmatrix} \begin{bmatrix} Y_1 \\ . \\ . \\ Y_n \end{bmatrix}$$

$$\widehat{\boldsymbol{\beta}} = \begin{bmatrix} \sum_{i=1}^{n} X_{1i}^2 & \sum_{i=1}^{n} X_{1i}X_{2i} \\ \sum_{i=1}^{n} X_{1i}X_{2i} & \sum_{i=1}^{n} X_{2i}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} X_{1i}Y_i \\ \sum_{i=1}^{n} X_{21}Y_i \end{bmatrix}$$

c. (10 points)

How are the solutions to (a) and (b) related?

Solution:

The solution to (b) is the sample analogue of the expression from part (a). This means that $\widehat{\beta}$ is a function of a random sample. Furthermore, in part (a), we are forecasting $\boldsymbol{Y}$ in the next time period based on our set of variables $\boldsymbol{X}_t$, whereas in part (b), $\boldsymbol{Y}$ is a function of the variables $\boldsymbol{X}_t$ in its own time period.

Question 3

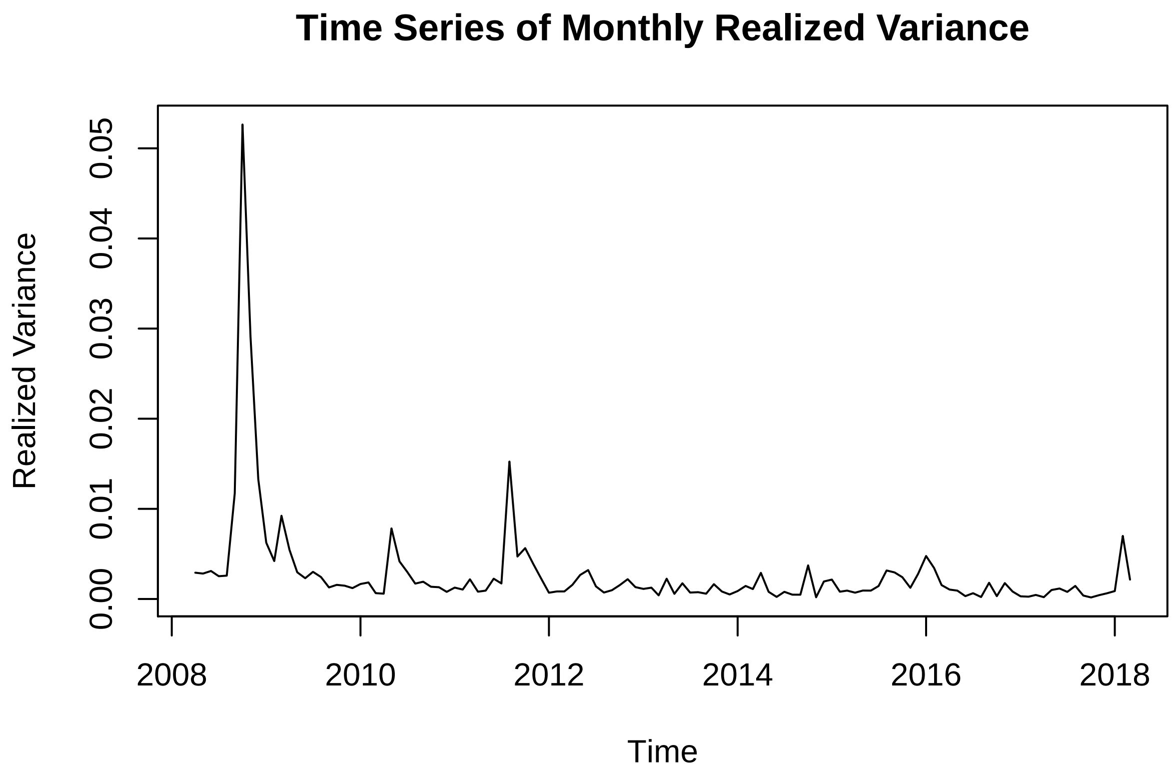
Download data for the CME Group Nasdaq 100 futures front-month contract for the period 1 Apr 2008 to 31 Mar 2018. This data can be obtained from Quandl via the symbol NQ1.

a. (10 points)

Using daily returns, compute monthly realized variance by summing the squared returns within each month: $RV_t = \sum_{i=1}^{n} r_{t+i}^2$, where t denotes the month, i indexes the days within the month, and n represents the number of trading days in the month (typically around 22). Plot the time series of realized variance.

Solution:

```
library(Quandl)
library(quantmod)

NQ1 <- Quandl("CHRIS/CME_NQ1", start_date = "2008-04-01", end_date = "2018-03-31", type = "xts", api_key

NQ1$returns_sq <- (dailyReturn(NQ1$Settle))^2

NQ1$month <- lubridate::month(index(NQ1))
NQ1$year <- lubridate::year(index(NQ1))
NQ1$day <- lubridate::day(index(NQ1))

NQ1 <- as.data.frame(NQ1)

NQ1 <- subset(NQ1, day < 23)  ## remove any days past 22 (only supposed to sum until n =  22)

realizedVar <- aggregate(NQ1$returns_sq, by = list(NQ1$month, NQ1$year),FUN=sum)

realizedVar$Date <- as.Date(with(realizedVar, paste(Group.2, Group.1, 1, sep = "-"), format="%Y-%m-%d"))
plot(realizedVar$Date, realizedVar$x, type = "l", main = "Time Series of Monthly Realized Variance", xla
```



b. (10 points)

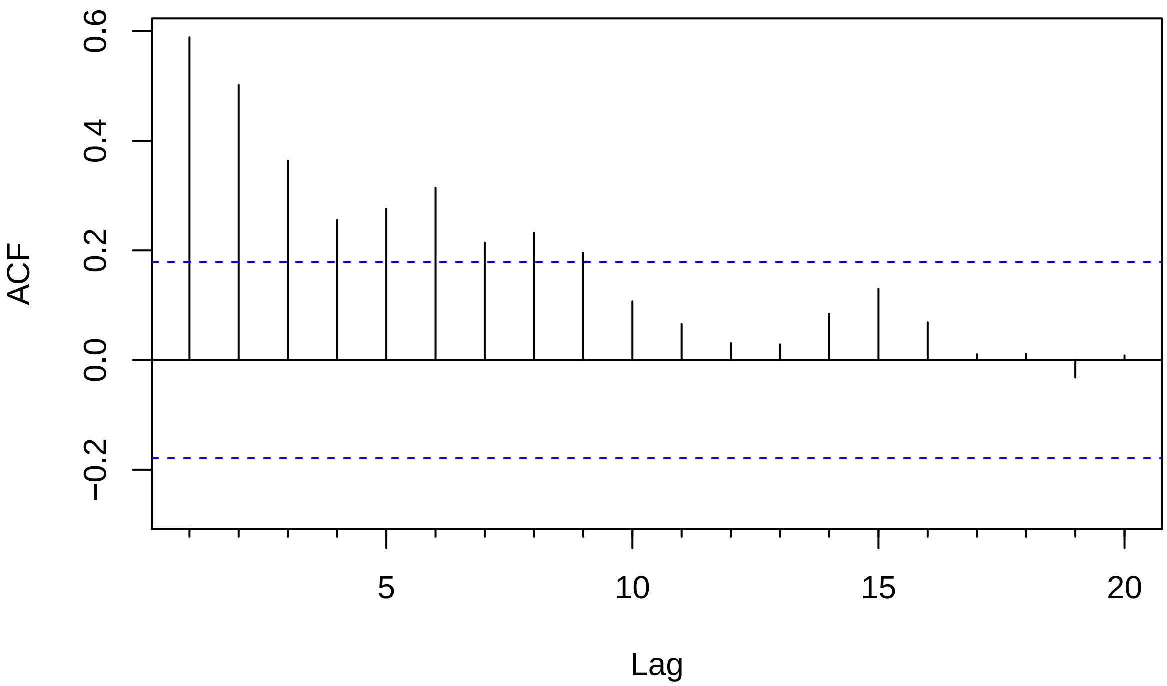
Find the best fitting ARMA model for the log of realized variance. Report the parameter estimates and standard errors and provide some interpretation.

Solution:

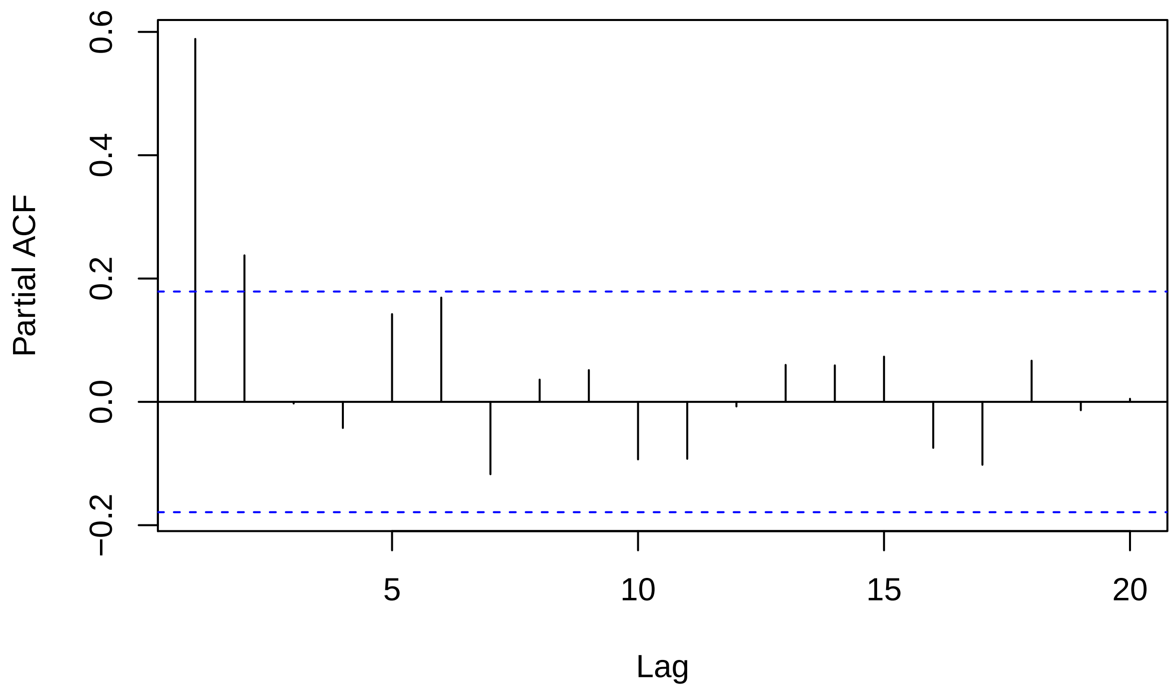
```r
library(forecast)
library(tseries)

adf.test(log(realizedVar$x), alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  log(realizedVar$x)
## Dickey-Fuller = -3.9044, Lag order = 4, p-value = 0.01628
## alternative hypothesis: stationary
```

```r
Acf(log(realizedVar$x), main ='')
```




```r
pacf(log(realizedVar$x), main='')
```

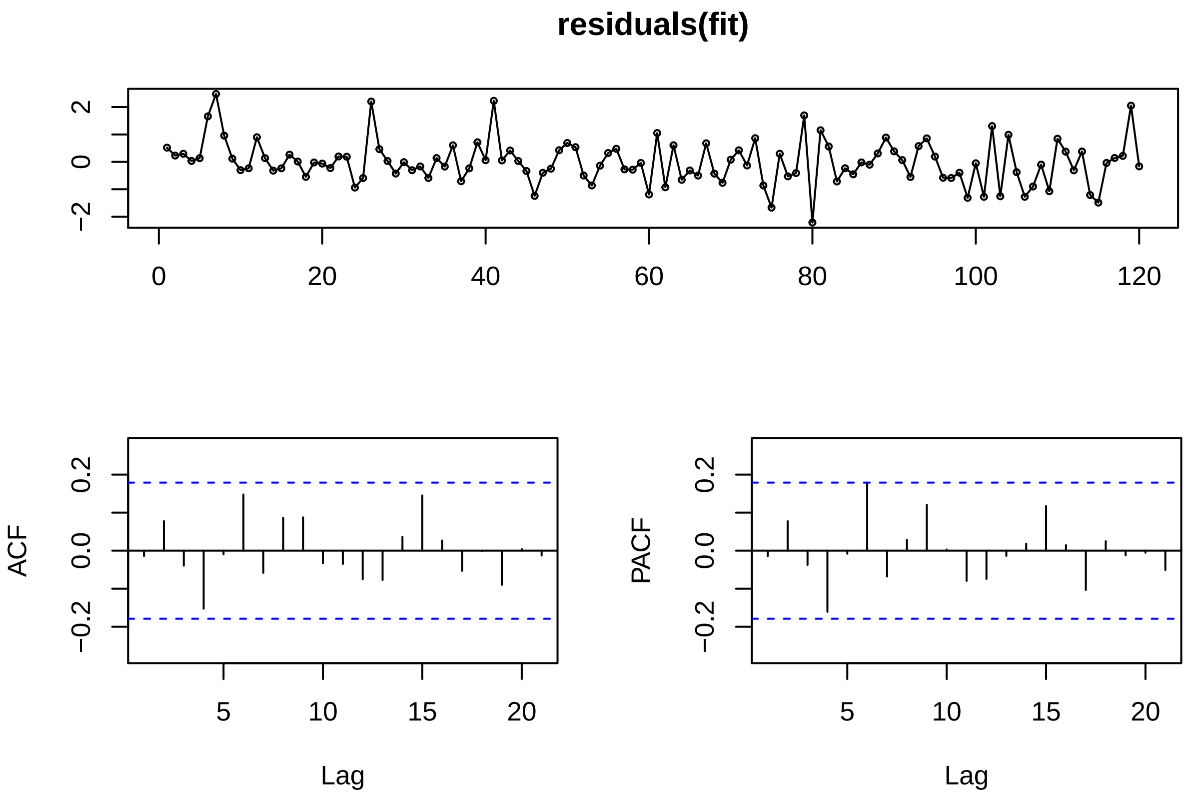



```r
fit <- auto.arima(log(realizedVar$x), max.d = 0)
summary(fit)
```

```
## Series: log(realizedVar$x)
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
```

```
##          ar1      ma1     mean
##       0.8102  -0.3536  -6.4964
## s.e.  0.0824   0.1331   0.2383
##
## sigma^2 estimated as 0.6365:  log likelihood=-141.91
## AIC=291.82   AICc=292.17   BIC=302.97
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.00839372 0.7877619 0.5830832 -1.633056 9.559705 0.8475524
##                     ACF1
## Training set -0.01408069
```

```
tsdisplay(residuals(fit))
```



A formal ADF test rejects the null hypothesis of non-stationarity so I will not difference the series. There are significant autocorrelations at lags below 10 periods, and the partial correllation plot shows a significant spike at lag 1 and 2 so I might want to test models with AR or MA components of order 1 and 2 (and maybe beyond).

The auto.arima function estimates that the best fitting model is an ARMA(1,1). The parameter estimate for the AR(1) is 0.8102 with a standard error of 0.0824, and the estimate for the MA(1) is -0.3536 with a standard error of 0.1331. This means that the next value in the series is taken as a dampened previous value by a factor of 0.81 and is increased by 0.35 of the last error term. Because we have a large ϕ, we can say the process has lots of memory. Also, because θ is less than 0, our first-order lags are negatively autocorrelated. The standard error of the AR(1) coefficient is a lot lower when compared to the MA(1) coefficient (relative to the estimate itself) so the AR(1) coefficient estimate is more reliable.

Additionally, the residual plot has a fairly small error range and is pretty centered around 0. The ACF also looks good - nothing is above or below the Bartlett bands. The model seems to be decent.