ECON 211C: Problem Set 2

Due: Tuesday, May 9, 2017

Eric M. Aldrich

David Sungho Park

Question 1

(a)

The likelihood function can be calculated as the joint density:

$$\mathcal{L}(k,\lambda|\boldsymbol{y}) = \prod_{i=1}^{n} f_{Y_i}(y_i|k,\lambda) = \prod_{i=1}^{n} \frac{k}{\lambda}\left(\frac{y_i}{\lambda}\right)^{k-1} e^{-\left(\frac{y_i}{\lambda}\right)^k} = \left(\frac{k}{\lambda^k}\right)^n e^{-\sum_{i=1}^n \left(\frac{y_i}{\lambda}\right)^k} \prod_{i=1}^{n} y_i^{k-1}.$$

By log transformation, we obtain the log likelihood:

$$\ell(k,\lambda|\boldsymbol{y}) = n(\log k - k\log\lambda) - \left(\frac{1}{\lambda}\right)^k \sum_{i=1}^{n} y_i^k + (k-1)\sum_{i=1}^{n} \log y_i.$$

(b)

Since $\arg\max \mathcal{L}(k,\lambda|\boldsymbol{y}) = \arg\max \ell(k,\lambda|\boldsymbol{y})$, we take the first order conditions of the log likelihood function.

$$\left.\frac{\partial \ell(k,\lambda|\boldsymbol{y})}{\partial k}\right|_{\hat{k},\hat{\lambda}} = \frac{n}{\hat{k}} - n\log\hat{\lambda} + \log\hat{\lambda}\cdot\hat{\lambda}^{-\hat{k}}\sum_{i=1}^{n} y_i^{\hat{k}} - \left(\frac{1}{\hat{\lambda}}\right)^{\hat{k}} \sum_{i=1}^{n} \log y_i \cdot y_i^{\hat{k}} + \sum_{i=1}^{n} \log y_i = 0 \tag{1}$$

$$\left.\frac{\partial \ell(k,\lambda|\boldsymbol{y})}{\partial \lambda}\right|_{\hat{k},\hat{\lambda}} = -\frac{n\hat{k}}{\hat{\lambda}} + \hat{k}\left(\frac{1}{\hat{\lambda}}\right)^{\hat{k}+1} \sum_{i=1}^{n} y_i^{\hat{k}} = 0 \tag{2}$$

Equation (2) can be simplified as

$$\frac{n}{\hat{\lambda}} = \left(\frac{1}{\hat{\lambda}}\right)^{\hat{k}+1} \sum_{i=1}^{n} y_i^{\hat{k}} \quad \Leftrightarrow \quad n\hat{\lambda}^{\hat{k}} = \sum_{i=1}^{n} y_i^{\hat{k}} \quad \Leftrightarrow \quad \hat{\lambda} = \left(\frac{1}{n}\sum_{i=1}^{n} y_i^{\hat{k}}\right)^{\frac{1}{\hat{k}}}.$$

Inserting this into equation (1) gives us

$$\frac{n}{\hat{k}} \underbrace{- n\log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^{\hat{k}}\right)^{\frac{1}{\hat{k}}} + \log\left(\frac{1}{n}\sum_{i=1}^{n} y_i^{\hat{k}}\right)^{\frac{1}{\hat{k}}} \cdot \left(\frac{1}{n}\sum_{i=1}^{n} y_i^{\hat{k}}\right)^{-1} \sum_{i=1}^{n} y_i^{\hat{k}}}_{=0} - \left(\frac{1}{n}\sum_{i=1}^{n} y_i^{\hat{k}}\right)^{-1} \sum_{i=1}^{n} \log y_i \cdot y_i^{\hat{k}} + \sum_{i=1}^{n} \log y_i = 0$$

$$\Leftrightarrow \quad \frac{n}{\hat{k}} - \left(\frac{1}{n}\sum_{i=1}^{n} y_i^{\hat{k}}\right)^{-1} \sum_{i=1}^{n} \log y_i \cdot y_i^{\hat{k}} + \sum_{i=1}^{n} \log y_i = 0.$$

(c)

To obtain the information matrix, we first derive the Hessian from the first order conditions in part (b).

$$\mathcal{H}(k,\lambda|\boldsymbol{y}) = \begin{bmatrix} \frac{\partial^2 \ell(k,\lambda|\boldsymbol{y})}{\partial k^2} & \frac{\partial^2 \ell(k,\lambda|\boldsymbol{y})}{\partial k \partial \lambda} \\ \frac{\partial^2 \ell(k,\lambda|\boldsymbol{y})}{\partial \lambda \partial k} & \frac{\partial^2 \ell(k,\lambda|\boldsymbol{y})}{\partial \lambda^2} \end{bmatrix}$$

where

$$\frac{\partial^2 \ell(k,\lambda|\boldsymbol{y})}{\partial k^2} = -\frac{n}{k^2} - \log\lambda^2 \cdot \lambda^{-k}\sum_{s=1}^{n} y_i^k + 2\log\lambda \cdot \lambda^{-k}\sum_{s=1}^{n}\log y_i \cdot y_i^k - \lambda^{-k}\sum_{s=1}^{n}\log y_i^2 \cdot y_i^k$$

$$\frac{\partial^2 \ell(k,\lambda|\boldsymbol{y})}{\partial k \partial \lambda} = -\frac{n}{k} + \lambda^{-(k+1)}\left[(1-k\log\lambda)\sum_{s=1}^{n} y_i^k + k\sum_{s=1}^{n}\log y_i \cdot y_i^k\right]$$

$$\frac{\partial^2 \ell(k,\lambda|\boldsymbol{y})}{\partial \lambda^2} = nk\lambda^{-2} - k(k+1)\lambda^{-(k+2)}\sum_{s=1}^{n} y_i^k$$

The information matrix is $\mathcal{I}(k,\lambda) = -E[\mathcal{H}(k,\lambda|\boldsymbol{y})]$ and the observed information matrix is $\tilde{\mathcal{I}}(k,\lambda) = -\mathcal{H}(k,\lambda|\boldsymbol{y})$. The inverse of the information matrix contains the approximations of the variances of $\hat{k}$ and $\hat{\lambda}$ in the diagonal.

Question 2

(a)

Table 1 displays the values of the computed least-squares estimates for the three different AR regression equations given.

Table 1: LS estimates from simulation (1 time with 30 obs)

	$AR(1)$	$AR(2)$	$AR(3)$
$E(\hat{\phi}_1)$	0.8388	1.4159	1.1728
$E(\hat{\phi}_2)$		-0.7219	-0.2421
$E(\hat{\phi}_3)$			-0.3456

Table 2 shows the means and standard deviations the least-square estimates from the simulation of 1,000 times with 30 observations. The R code is included in the Appendix.

Table 2: Mean and Standard Deviation of LS estimates from Simulation (a)

	$AR(1)$	$AR(2)$	$AR(3)$
$E(\hat{\phi}_1)$	0.8501	1.217	1.194
$sd(\hat{\phi}_1)$	0.0974	0.1941	0.2039
$E(\hat{\phi}_2)$		-0.4336	-0.3763
$sd(\hat{\phi}_2)$		0.167	0.2737
$E(\hat{\phi}_3)$			-0.0445
$sd(\hat{\phi}_3)$			0.1895

(b)

Table 3 displays the means and standard deviations of the parameter estimates obtained in parts (a) and (b).

Table 3: Mean and Standard Deviation of LS Estimates from Simulations (a) and (b)

	(a) 30 obs x 1,000 times			(b) 1,000 obs x 1,000 times		
	$AR(1)$	$AR(2)$	$AR(3)$	$AR(1)$	$AR(2)$	$AR(3)$
$E(\hat{\phi}_1)$	0.8501	1.217	1.194	0.8629	1.16	1.144
$sd(\hat{\phi}_1)$	0.09735	0.1941	0.2039	0.07167	0.1745	0.1873
$E(\hat{\phi}_2)$		-0.4336	-0.3763		-0.3329	-0.3006
$sd(\hat{\phi}_2)$		0.167	0.2737		0.1566	0.2457
$E(\hat{\phi}_3)$			-0.04445			-0.02154
$sd(\hat{\phi}_3)$			0.1895			0.1573

(c)

Table 4 displays the means and standard deviations of the parameter estimates obtained in parts (a), (b), and (c).

Table 4: Mean and Standard Deviation of LS Estimates from Simulations (a)-(c)

	(a) 30 obs x 1,000 times			(b) 1,000 obs x 1,000 times			(c) 100,000 obs x 1,000 times		
	$AR(1)$	$AR(2)$	$AR(3)$	$AR(1)$	$AR(2)$	$AR(3)$	$AR(1)$	$AR(2)$	$AR(3)$
$E(\hat{\phi}_1)$	0.8501	1.217	1.194	0.8629	1.16	1.144	0.8615	1.16	1.151
$sd(\hat{\phi}_1)$	0.09735	0.1941	0.2039	0.07167	0.1745	0.1873	0.06983	0.1798	0.1938
$E(\hat{\phi}_2)$		-0.4336	-0.3763		-0.3329	-0.3006		-0.3361	-0.3133
$sd(\hat{\phi}_2)$		0.167	0.2737		0.1566	0.2457		0.1644	0.2558
$E(\hat{\phi}_3)$			-0.04445			-0.02154			-0.01785
$sd(\hat{\phi}_3)$			0.1895			0.1573			0.1552

Question 3

We use the `auto.arima` function. There are three different information criterion for `auto.arima`: AIC, AICc or BIC. `auto.arima` returns $ARIMA(2,1,2)$ if we use AIC or AICc and $ARIMA(0,1,0)$ with drift, or "random walk with constant," if we use BIC. In all cases, $d = 1$, which is the number of differences needed for the time series to become stationary. Recall that $ARIMA$ (Auto-Regressive Integrated Moving Average) contains not only the AR terms and MA terms but also an "integrated" version of a stationary series.
A simple way to think of first differencing our time series of price is to transform it into returns. For the daily returns, `auto.arima` gives us $ARIMA(2,0,2)$ with AIC or AICc and $ARIMA(0,0,0)$ with BIC, both of which yields $d = 0$. Since the series is stationary, we can now say that we have $ARMA$. Table 5 reports the parameter estimates and standard errors in the parentheses.

Table 5: ARMA estimation with different information criteria

Info. Criterion	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\psi}_1$	$\hat{\psi}_2$
AIC	0.6642	-0.5400	-0.7097	0.7505
	(0.024)	(0.018)	(0.015)	(0.011)
AICC	0.6642	-0.5400	-0.7097	0.7505
	(0.024)	(0.018)	(0.015)	(0.011)
BIC	.	.	.	.

Appendix: R Code

```
##################################### Question 2 (a) ######################################
# Simulation_0: 30 obs x 1 time
nObs = 30
nSim = 1000
phi = c(1.3,-0.41)

eps = matrix(rnorm(nObs,0,1), nObs,1)
y = matrix(0,nObs,1)

for(i in 1:length(phi)){
  y[i] = eps[i]
}
for(i in (length(phi)+1):length(y)){
  y[i] = phi %*% y[(i-1):(i-length(phi))] + eps[i]
}

# Estimation
LS1 = signif(summary(lm(y[2:nObs] ~ y[1:(nObs-1)]))$coefficients[2,1],4)
LS2 = signif(summary(lm(y[3:nObs] ~ y[2:(nObs-1)] + y[1:(nObs-2)]))$coefficients[2:3,1],4)
LS3 = signif(summary(lm(y[4:nObs] ~ y[3:(nObs-1)] + y[2:(nObs-2)] + y[1:(nObs-3)]))
$coefficients[2:4,1],4)

# Export to LaTeX
library(xtable)
estResults = matrix("",3,3)
estResults[1,1]   = LS1
estResults[1:2,2] = LS2
estResults[1:3,3] = LS3
xtable(estResults,align = "cccc")


# Simulation_a: 30 obs x 1,000 times
nObs = 30
nSim = 1000
phi = c(1.3,-0.41)
eps_a = matrix(0, nObs,nSim)
y_a = matrix(0,nObs,nSim)

for (i in 1:nObs){
  for (j in 1:nSim){
    eps_a[i,j] = rnorm(1, mean, sigma)
  }
```

```
}

for(j in 1:nSim){
  for(i in 1:length(phi)){
    y_a[i,j] = eps_a[i,j]
  }
  for(i in (length(phi)+1):length(y)){
    y_a[i,j] = phi %*% y_a[(i-1):(i-length(phi)),j] + eps_a[i,j]
  }
}

# Estimation
estResults_a = matrix(0,6,nSim)
for(j in 1:nSim){
  estResults_a[1,j]   = summary(lm(y_a[2:nObs,j] ~ y_a[1:(nObs-1),j]))$coefficients[2,1]
  estResults_a[2:3,j] = summary(lm(y_a[3:nObs,j] ~ y_a[2:(nObs-1),j] + y_a[1:(nObs-2),j]))
   $coefficients[2:3,1]
  estResults_a[4:6,j] = summary(lm(y_a[4:nObs,j] ~ y_a[3:(nObs-1),j] + y_a[2:(nObs-2),j]
   + y_a[1:(nObs-3),j]))$coefficients[2:4,1]
}

# Report Mean and Standard Deviation
mu = signif(apply(estResults_a,1,mean),4)
sigma = signif(apply(estResults_a,1,sd),4)

# Export to LaTeX
TeX_a = matrix("",6,3)
TeX_a[1:2,1] = c(mu[1],sigma[1])
TeX_a[1:4,2] = c(mu[2],sigma[2],mu[3],sigma[3])
TeX_a[1:6,3] = c(mu[4],sigma[4],mu[5],sigma[5],mu[6],sigma[6])
xtable(TeX_a,align="cccc")

#################################### Question 2 (b) ####################################
# Simulation_b: 1,000 obs x 1,000 times
nObs = 1000
nSim = 1000
phi = c(1.3,-0.41)
eps_b = matrix(0, nObs,nSim)
y_b = matrix(0,nObs,nSim)

for (i in 1:nObs){
  for (j in 1:nSim){
    eps_b[i,j] = rnorm(1, mean, sigma)
```

```
  }
}

for(j in 1:nSim){
  for(i in 1:length(phi)){
    y_b[i,j] = eps_b[i,j]
  }
  for(i in (length(phi)+1):length(y)){
    y_b[i,j] = phi %*% y_b[(i-1):(i-length(phi)),j] + eps_b[i,j]
  }
}

# Estimation
estResults_b = matrix(0,6,nSim)
for(j in 1:nSim){
  estResults_b[1,j]   = summary(lm(y_b[2:nObs,j] ~ y_b[1:(nObs-1),j]))$coefficients[2,1]
  estResults_b[2:3,j] = summary(lm(y_b[3:nObs,j] ~ y_b[2:(nObs-1),j] + y_b[1:(nObs-2),j]))
   $coefficients[2:3,1]
  estResults_b[4:6,j] = summary(lm(y_b[4:nObs,j] ~ y_b[3:(nObs-1),j] + y_b[2:(nObs-2),j]
   + y_b[1:(nObs-3),j]))$coefficients[2:4,1]
}

# Report Mean and Standard Deviation
mu = signif(apply(estResults_b,1,mean),4)
sigma = signif(apply(estResults_b,1,sd),4)

# Export to LaTeX
library(xtable)
TeX_b = matrix("",6,3)
TeX_b[1:2,1] = c(mu[1],sigma[1])
TeX_b[1:4,2] = c(mu[2],sigma[2],mu[3],sigma[3])
TeX_b[1:6,3] = c(mu[4],sigma[4],mu[5],sigma[5],mu[6],sigma[6])
xtable(TeX_b,digits = 4)

TeX_ab = cbind(TeX_a,TeX_b)
xtable(TeX_ab,digits = 4)


##################################### Question 2 (c) ####################################
# Simulation_c: 100,000 obs x 1,000 times
nObs = 10000
nSim = 1000
phi = c(1.3,-0.41)
```

```
eps_c = matrix(0, nObs,nSim)
y_c = matrix(0,nObs,nSim)

for (i in 1:nObs){
  for (j in 1:nSim){
    eps_c[i,j] = rnorm(1, mean, sigma)
  }
}

for(j in 1:nSim){
  for(i in 1:length(phi)){
    y_c[i,j] = eps_c[i,j]
  }
  for(i in (length(phi)+1):length(y)){
    y_c[i,j] = phi %*% y_c[(i-1):(i-length(phi)),j] + eps_c[i,j]
  }
}

# Estimation
estResults_c = matrix(0,6,nSim)
for(j in 1:nSim){
  estResults_c[1,j]   = summary(lm(y_c[2:nObs,j] ~ y_c[1:(nObs-1),j]))$coefficients[2,1]
  estResults_c[2:3,j] = summary(lm(y_c[3:nObs,j] ~ y_c[2:(nObs-1),j] + y_c[1:(nObs-2),j]))
   $coefficients[2:3,1]
  estResults_c[4:6,j] = summary(lm(y_c[4:nObs,j] ~ y_c[3:(nObs-1),j] + y_c[2:(nObs-2),j]
   + y_c[1:(nObs-3),j]))$coefficients[2:4,1]
}

# Report Mean and Standard Deviation
mu = signif(apply(estResults_c,1,mean),4)
sigma = signif(apply(estResults_c,1,sd),4)

# Export to LaTeX
library(xtable)
TeX_c = matrix("",6,3)
TeX_c[1:2,1] = c(mu[1],sigma[1])
TeX_c[1:4,2] = c(mu[2],sigma[2],mu[3],sigma[3])
TeX_c[1:6,3] = c(mu[4],sigma[4],mu[5],sigma[5],mu[6],sigma[6])
xtable(TeX_c,digits = 4)

TeX_abc = cbind(TeX_ab,TeX_c)
xtable(TeX_abc,align = "ccccccccc")
```

```
####################################   Question 3   ####################################
install.packages("forecast")
library(forecast)
library(quantmod)
getSymbols("XIV",from="2014-04-25",to="2015-04-24")

# Daily adjusted closing prices
prices = XIV$XIV.Adjusted
auto.arima(prices,ic="aic")
auto.arima(prices,ic="aicc")
auto.arima(prices,ic="bic")

# Daily returns
returns = dailyReturn(prices)
auto.arima(returns,ic="aic")
auto.arima(returns,ic="aicc")
auto.arima(returns,ic="bic")

# Export to LaTeX
library(xtable)
aic  = rbind(auto.arima(returns,ic="aic")$coef,diag(auto.arima(returns,ic="aic")$var.coef))
aicc = rbind(auto.arima(returns,ic="aicc")$coef,diag(auto.arima(returns,ic="aicc")$var.coef))
bic  = matrix("",1,4)
TeX_arma = rbind(aic,aicc,bic)
xtable(TeX_arma, align = "ccccc")
```