Homework 4 Solution

46-923, Fall 2017

You can submit separate pdf files, one generated from the R Markdown, and the other from the "derivations" required in Question 2. The relevant .Rmd file should also be submitted.

Please do not submit photos of your homework. Scanners are available for your use.

Question 1

We have already seen that the log returns are not well-modelled by a normal distribution, as the normal distribution does not place enough probability in the tails to model the extreme events that can occur.

Here, we will assume that the daily log returns for an equity can be modelled as being i.i.d. with distribution given by σT, where T is a random variable with the t-distribution with ν degrees of freedom. The median of the distribution is assumed to be zero.

a. Write an R function that takes as input a vector of values (e.g., log daily returns from a single equity), along with an assumed value for ν, and then returns the MLE for σ along with the standard error for that estimator and a 95% confidence interval for σ.

```
estimatesigma = function(r, nu) {
    negtdist = function(sigma, r, nu) {
        return((-1) * sum(dt(r/sigma, nu, log = T)) + length(r) *
            log(sigma))
    }
    optout = optim(sd(r), negtdist, r = r, nu = nu, hessian = T)
    se = sqrt(1/optout$hessian)
    list(mle = optout$par, se = se, CI_lower = optout$par - qnorm(0.025) *
        se, CI_upper = optout$par + qnorm(0.025) * se)
}
```

b. Demonstrate the use of this function on some real log return data found using `quantmod()`. Try at least three different equities.

```
AAPL = data.frame(getSymbols("AAPL", from = "2010-01-01", to = Sys.Date() -
    1, auto.assign = FALSE))
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
```

```
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

```r
logrAAPL = diff(log(AAPL$AAPL.Adjusted))
estout = estimatesigma(logrAAPL, 10)
estout
```

```
## $mle
## [1] 0.01335774
##
## $se
##              [,1]
## [1,] 0.0002437405
##
## $CI_lower
##            [,1]
## [1,] 0.01383547
##
## $CI_upper
##            [,1]
## [1,] 0.01288002
```

```r
BABA = data.frame(getSymbols("BABA", from = "2015-07-01", to = Sys.Date() -
    1, auto.assign = FALSE))
logrBABA = diff(log(BABA$BABA.Adjusted))
estout = estimatesigma(logrBABA, 10)
estout
```

```
## $mle
## [1] 0.01644631
##
## $se
##              [,1]
## [1,] 0.0005406601
##
## $CI_lower
##            [,1]
## [1,] 0.01750599
##
## $CI_upper
##            [,1]
## [1,] 0.01538664
```

```r
C = data.frame(getSymbols("C", from = "2015-01-01", to = Sys.Date() -
    1, auto.assign = FALSE))
logrC = diff(log(C$C.Adjusted))
```

```r
estout = estimatesigma(logrC, 10)
estout
```

```
## $mle
## [1] 0.01317947
##
## $se
##              [,1]
## [1,] 0.0004007398
##
## $CI_lower
##            [,1]
## [1,] 0.01396491
##
## $CI_upper
##            [,1]
## [1,] 0.01239404
```

c. Create a second function which maximizes the likelihood over different values of ν, in addition to maximizing over σ. This function does not need to return standard errors or confidence intervals. (We will discuss how to do this soon.) Test this on some different examples.

```r
estimate2 = function(r) {
    # pars[1] for sigma, pars[2] for nu
    negtdist = function(pars, r) {
        return((-1) * sum(dt(r/pars[1], pars[2], log = T)) +
            length(r) * log(pars[1]))
    }
    optout = optim(c(sd(r), 10), negtdist, r = r, hessian = T)
    list(mle = optout$par, hessian = optout$hessian)
}
estout = estimate2(logrAAPL)
estout
```

```
## $mle
## [1] 0.01146971 3.95185603
##
## $hessian
##             [,1]         [,2]
## [1,] 17799739.54 -10447.59005
## [2,]   -10447.59     13.29432
```

Question 2

Suppose that X is binomial(n, p). The MLE for p is, not surprisingly, the sample proportion X/n. (You do not need to prove this.)

a. What is the MLE for $p/(1-p)$? (This is called the *odds*.)

b. Approximate the distribution for the MLE of the odds.

c. Use part (b) to construct a $100(1-\alpha)\%$ confidence interval for the odds.

d. Write a simulation procedure that tests the validity of the confidence interval found in part (c). Is the confidence interval an adequate approximation when $n = 10$ and $p = 0.10$?

```r
p0 = 0.1
theta0 = p0/(1 - p0)
n = 10
reps = 10000
alpha = 0.05
thetahat = numeric(reps)
lower = numeric(reps)
upper = numeric(reps)

coverstruth = logical(reps)

for (i in 1:reps) {
    x = rbinom(1, n, p0)
    thetahat[i] = x/n/(1 - x/n)
    phat = x/n
    se = sqrt(phat/n/(1 - phat)^3)
    lower[i] = thetahat[i] - qnorm(1 - alpha/2) * se
    upper[i] = thetahat[i] + qnorm(1 - alpha/2) * se

    coverstruth[i] = (lower[i] < theta0) & (upper[i] > theta0)
}
coverstruth[is.na(coverstruth)] = FALSE
print(mean(coverstruth, na.rm = T))
```

```
## [1] 0.6471
```

```r
# The confidence inverval is not an adequate approximation
# with n=10 and p=0.1 since we are using asymptotic variance.
```