RMSC4002 Tutorial 3

Chapter 1-2

September 27, 2017

1 QQ-chisquare Plot

1.1 Using R

```
> u<-cbind(u1,u2,u3)           # combine into matrix u
> n2<-nrow(u)                  # no of row in u
> n1<-n2-180+1                 # 180-th obs before n2
> u180<-u[n1:n2,]              # save the most recent 180 days to u180
> (m<-apply(u180,2,mean))      # compute column mean of u180
           u1            u2            u3
-0.0003229707  0.0002479614 -0.0019383056
> (s<-var(u180))
              u1            u2           u3
u1  1.408494e-04 -7.129762e-07 1.303655e-04
u2 -7.129762e-07  6.318444e-05 1.872616e-05
u3  1.303655e-04  1.872616e-05 3.507328e-04
```

```
> sinv<-solve(s)                              # compute inv(s)
> m<-matrix(m,nr=180,nc=3,byrow=T)        # change m into 180x3 matrix
> d2<-diag((u180-m)%*%sinv%*%t(u180-m))# compute squared gen. dist.
> d2<-sort(d2)                                # sort d2 in ascending order
> i<-((1:180)-0.5)/180                        # create vector of percentile
> q<-qchisq(i,3)                              # compute quantiles of chisq(3)
> qqplot(q,d2)                                # QQ-chisquare plot
> abline(lsfit(q,d2))                         # add the reference line
```

Remark:

- *cbind, rbind*: Take a sequence of vector, matrix or data-frame arguments and combine by colomns or rows, respectively.
- *byrow=T*: It would tell R to fill the elements row wise.

1.2 Using EXCEL

EXCEL has no built-in function for QQ-Chisquare plot. However, we can use the built-in matrix multiplication function mmult() to compute the generalized distance and produce the QQ-Chisquare plot.

- We set up $u1, u2$, and $u3$ in cells A3:C1044.
- Compute the mean vector of $u180$, say m and store it in I12:K12.
- Compute the $u_i - m$ and store them in E865:G1044.

- In cell I3, enter the formula: =VAR.S(E865:E1044); in cell J3 and E4, enter the formula: =COVARIANCE.S(E865:E1044,F865:F1044); in cell K3 and E5, enter the formula: =COVARIANCE.S(F865:F1044,G865:G1044); in cell K4 and J5, enter the formula: =COVARIANCE.S(E865:E1044,G865:G1044); in cell J4, enter the formula: =VAR.S(F865:F1044); in cell K5, enter the formula: =VAR.S(G865:G1044). This will produce the covariance matrix of u180.
- Highlight the cell I8:K10 and enter the formula: =MINVERSE(I3:K5). You need to use **shift-ctrl-enter** to enter. This will produce the inverse of the Covariance matrix S.
- In cell I865, enter the formula: =SUMPRODUCT(MMULT(E865:G865,I8:K10),E865:G865). This formula used to compute the distance $d_1^2 = (u_i - \bar{u})' S^{-1} (u_i - \bar{u})$.
- Create 1 to 180 in J865:J1044 and the corresponding $(i - 0.5)/180$ in K865:K1044.
- In cell L865, enter =CHIINV(1-K865,3) to compute the quantile from the Chi-square(3) distribution. Copy the formula to L866:L1044.
- Sorted the distance in I865:I1044 in ascending order and store them in M865:M1044.
- Finally, plot the cell L865:N865. This will produce the QQ-Chisquare plot.
- Add in the least square trend line as the reference line by right-clicking any points in the plot.

2 Generating Multivariate Normal Random Numbers

```
set.seed(7)                 # set random seed
mu<-apply(u180,2,mean)      # compute daily return rate
sigma<-var(u180)            # compute daily variance rate
C<-chol(sigma)              # Cholesky decomposition of sigma
s<-cbind(t1,t2,t3)          # combine t1,t2,t3 to form s
s0<-s[1043,]                # set s0 to the most recent price
for (i in 1:90) {           # simulate price for future 90 days
   z<-rnorm(3)              # generate normal random vector
   v<-mu+t(C)%*%z           # transform to multivariate normal
   s1<-s0*(1+v)             # new stock price
   s<-rbind(s,t(s1))        # append s1 to s
   s0<-s1                   # update s0
}
```

Remark:

- *set.seed()*: Set the seed of R's random variable generator, which is useful for creating simulations or random objects that can be reproduced.

3 Moving Standard Deviation

```
> d<-read.csv("stock.csv")
> t1<-as.ts(d$HSBC)           # save as time series
> t2<-as.ts(d$CLP)
> t3<-as.ts(d$CK)
> u1<-(lag(t1)-t1)/t1         # compute u
> u2<-(lag(t2)-t2)/t2
> u3<-(lag(t3)-t3)/t3
```

```
msd<-function(t,w) {        # function to compute moving s.d.
   n<-length(t)-w+1
   out<-c()                 # initialize a null vector to store the output
   for (i in 1:n) {
      j<-i+w-1
      s<-sd(window(t,i,j))     # compute the sd of t(i) to t(j)
      out<-append(out,s)       # append the result to out
   }
   out<-as.ts(out)             # convert to time series
}
s1_90<-msd(u1,90)     # compute 90-day moving sd of u1
s1_180<-msd(u1,180)   # compute 180-day moving sd of u1
par(mfrow=c(2,1))     # time series plots
plot(s1_90)
plot(s1_180)
```

4 GARCH Model

```
> library(tseries)              # load library  "tseries"
> res<-garch(u1,order=c(1,1))  # fit GARCH(1,1) model and save it to res
> names(res)                    # see what is in res
 [1] "order"         "coef"          "n.likeli"      "n.used"
 [5] "residuals"     "fitted.values" "series"        "frequency"
 [9] "call"          "asy.se.coef"
> round(res$coef,6)             # display the coefficient using 6 digits
      a0       a1       b1
0.000009 0.029318 0.934555
> -2*res$n.likeli               # compute log-likelihood value
[1] 7590.325
```

Remark:

- *res$n.likeli*: The negative log-likelihood function evaluated at the coefficient estimates(apart from some constants.)

5 Exercise

Exercise 2013-14 final Q2 Let u_t be the relative return of the daily closing stock price X at day t, $t = 1, ..., n$. We compute the sample mean $\bar{u} = (0.00050, 0.00087, 0.00138)'$, and covariance matrix of stock A, B, and C from Jan 4, 2012 to Dec 30, 2012.

$$S = \begin{bmatrix} 0.00022 & 0.00002 & 0.00009 \\ 0.00002 & 0.00005 & 0.00003 \\ 0.00009 & 0.00003 & 0.00020 \end{bmatrix}$$

A L-GARCH(1,1) model $\sigma_n^2 = \omega + \beta\sigma_{n-1}^2 + \alpha u_{n-1}^2 + \theta I_{n-1} u_{n-1}^2$ is fitted to stock A with $\hat{\omega} = 0.0000104$, $\hat{\alpha} = 0.0222$, $\hat{\beta} = 0.9170$ and $\hat{\theta} = 0.0397$

- Given that the next closing price (Dec 31, 2012) of stock A is $S_{n+1} = 78.1$, and $S_n = 79.7$, find the estimated variance rate of stock A.
- If we fit the variance rate using EWMA(λ) model such that the estimated variance rate using this EWMA model is exactly equal to the $\hat{\sigma}_{n+1}^2$ using L-GARCH(1,1) in the first part, what is the value of this parameter λ?
- Assume that u_t^2 are all independent with $Var(u_t^2) = \gamma^2$ for $t = 1, ..., n$ and σ_0^2 is a known constant. Find the variance of $\hat{\sigma}_{n+1}^2$ in the second part. What is the variance of $\hat{\sigma}_{n+1}^2$ when $n \to \infty$?