S681 Financial Statistics
Homework 3
April 27 2012
**Cheng-Han Yu**

Textbook: Ruppert, D. (2011) *Statistics and Data Analysis for Financial Engineering*, Springer, New York.

1. Suppose that a stock price follows the Black-Scholes model with $\mu = 0.2$ and $\sigma = 0.5$ with an initial price $S_0 = \$10$, that is $S_t = S_0 \exp\{(\mu - \sigma^2/2)t + \sigma B_t\}$ where $B_t$ is a standard Brownian motion.
(a) Simulate a sample path for each of Brownian motion $(B_{t_1}, \ldots, B_{t_k})$ and price process $(S_{t_1}, \ldots, S_{t_k})$, with $t_j = j/k, j = 1, \ldots, k = 200$. Plot the sample paths.
**Solution:**

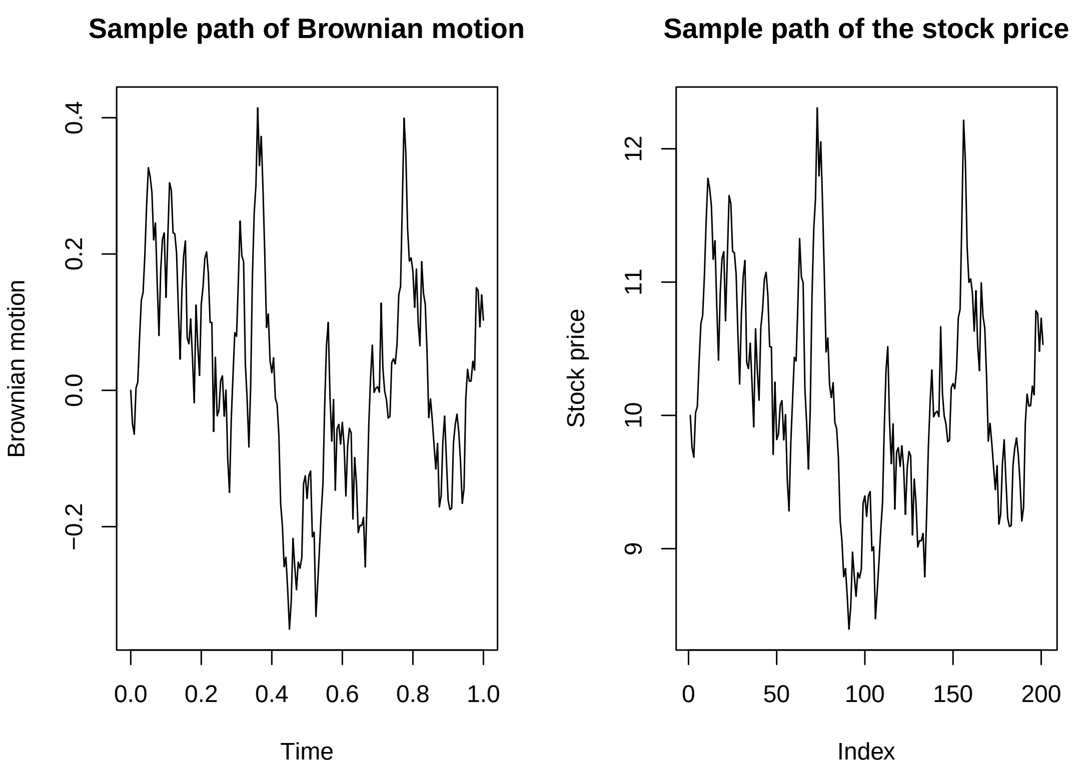


Figure 1: Brownian motion and stock price sample path

The R code is

```
bm<-function(x, t0, T, N)
{
if (T<= t0)
stop("incorrect times")
```

```
# initial value should be less than the terminal value
dt<-(T-t0)/N
# interval b/w 2 time points
t<-seq(t0, T, by = dt)
# create time points seq with interval dt
X<-ts(cumsum(c(x, rnorm(N)*sqrt(dt))), start = t0, deltat = dt,
names="Brownian motion")
# create a time series of Brownian motion
return(invisible(X))
}
BM<-bm(0,0,1,200)
> mu=0.2
> sigma=0.5
> S=rep(0, length(BM))
> S[1]=10
> PR<-rep(0, length(BM))
> dt = 1/200
> for (i in 2:length(BM))
+ {
+ S[i]<-S[1]*exp((mu-(sigma^2/2))*dt+sigma*BM[i])
+ PR[i]<-S[i]/S[i-1]
+ }
> PR<-PR[2:length(BM)]
> par(mfrow=c(1,2))
> plot(BM, type="l", ylab="Brownian motion", main="Sample path of
Brownian motion")
> plot(S, type="l", ylab="Stock price", main="Sample path of the
stock price")
```

(b) Suppose you provide the 200 simulated stock price data in (a) to a stock analyst without revealing $\mu = 0.2$ and $\sigma = 0.5$. How do you advise him or her to estimate $\mu$ and $\sigma$ based on the 200 data? What are the estimated values?
**Solution:**

$S_{t_i} = S_{t_{i-1}} \exp\{(\mu - \sigma^2/2)(t_i - t_{i-1}) + \sigma(B_{t_i} - B_{t_{i-1}})\} = S_{t_{i-1}} \exp\{(\mu - \sigma^2/2)\Delta + \sigma(B_{t_i} - B_{t_{i-1}})\}$, where $\Delta = t_i - t_{i-1} = 1/k = 1/200$. Hence,

$$\begin{aligned}\log \frac{S_{t_i}}{S_{t_{i-1}}} &= (\mu - \sigma^2/2)(t_i - t_{i-1}) + \sigma(B_{t_i} - B_{t_{i-1}}) \\ &= (\mu - \sigma^2/2)\Delta + \sigma(B_{t_i} - B_{t_{i-1}})\end{aligned}$$

Since $(B_{t_i} - B_{t_{i-1}}) \sim N(0, t_i - t_{i-1}) = N(0, \Delta)$, $\log \frac{S_{t_i}}{S_{t_{i-1}}} \sim N\big((\mu - \sigma^2/2)\Delta, \sigma^2\Delta\big)$.

Let $Y_i = \log \frac{S_{t_i}}{S_{t_{i-1}}}$. We can use the sample data $(Y_1, \ldots, Y_{200})$ to estimate $\mu$

and $\sigma$ based on the sample mean $\bar{Y} \equiv \frac{1}{n}\sum_{i=1}^{n} Y_i$ and sample variance $S_Y^2 \equiv \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2$. Letting $E(Y) = (\mu - \sigma^2/2)\Delta$ and $Var(Y) = \sigma^2\Delta = S_Y^2$, we get the estimators $\hat{\mu} = \frac{\bar{Y}}{\Delta} + \frac{S_Y^2}{2\Delta}$ and $\hat{\sigma} = \frac{S_Y}{\sqrt{\Delta}}$. According to the result of R program, the estimators $\mu$ and $\sigma$ are $\hat{\mu} = 0.1624862$ and $\hat{\sigma} = 0.4703018$..

Theoretically, $E(\hat{mu}) = \mu$ and $E(\hat{\sigma}) = \sigma$, the expectation of sample statistics equals their true parameters (unbiased). In our case, the two estimators are underestimated for the true parameters. This is because we only simulate once for the Brownian motion. We can increases the number of $k$ or do simulation many times instead of once (resampling and use the sample mean) to get more precise estimators for $\mu$ and $\sigma$.

The R code is

```
> Y<-log(PR)
> Ybar<-mean(Y)
> sY<-sd(Y)
> sigmahat<-sY/sqrt(dt)
> muhat<-Ybar/dt+(sY^2)/(2*dt)
> muhat
[1] 0.1624862
> sigmahat
[1] 0.4703018
```

(c) For the 200 simulated stock price data in (a), compute the sample skewness and sample kurtosis for the log returns $\log S_{t_k} - \log S_{t_{k-1}}$.

**Solution:**

The sample skewness and sample kurtosis for the log returns $\log S_{t_k} - \log S_{t_{k-1}}$ are 0.06546059 and 2.398885, respectively. Again, since log returns are normally distributed, the skewness and kurtosis should be 0 and 3, theoretically. Bu in our simulation result, the log returns are slightly right-skewed and have heavier tails than standard normal distribution. These two values vary for each simulation.

The R code is

```
> devY <- Y-Ybar
> skew<-(sum(devY^3)/(length(Y)-1))/(sY^3)
> kurt<-(sum(devY^4)/(length(Y)-1))/(sY^4)
> skew
[1] 0.06546059
> kurt
[1] 2.398885
```

2. Suppose the prices of an asset obey the model under the risk-neutral probability Q,

$$\log \frac{S_t}{S_{t-1}} = r - \frac{1}{2}\sigma_t^2 + \sigma_t z_t \tag{1}$$

$$\sigma_t^2 = 0.00001524 + 0.7162\sigma_{t-1}^2 + 0.1883\sigma_{t-1}^2(z_{t-1} - 0.007452)^2 \tag{2}$$

where $z_t$ are i.i.d. standard normal random variables. The annualized stationary standard deviation is computed as $\bar{\sigma} = 24.13\%$. Consider a European call option with $K =$ \$1, $T = 30$ days, and current price $S_0 = 0.9, 1, 1.1$ and current volatility $\sigma_0 = 24\%$. Assume the risk-free interest rate $r = 0.05/252$.

(a) Simulate the GARCH-M process $n$ times and use antithetic variable method to evaluate the price $e^{-rT}\mathbb{E}_Q(\max\{S_T - K, 0\})$ of the call option for each of the three current prices $S_0$.
The method works as follows: standard normal random variables $(z_1, \ldots, z_T)$ are used to simulate each $S_T$ from equations (1)-(2). We denote $S_T$ by $S_T(z_1, \ldots, z_T)$ to stress the dependence of $S_T$ on $(z_1, \ldots, z_T)$. Each time when one $S_T(z_1, \ldots, z_T)$ is simulated by use of $(z_1, \ldots, z_T)$ from equations (1)-(2), we switch the signs of all $z_T$ and use $(-z_1, \ldots, -z_T)$ to compute $S_T(z_1, \ldots, z_T)$ in the same way as we compute $S_T(z_1, \ldots, z_T)$ by use of $(z_1, \ldots, z_T)$ from equations (1)-(2). Suppose we repeat n times and obtain $S_T^i(z_1^i, \ldots, z_T^i)$, $S_T^i(-z_1^i, \ldots, -z_T^i)$, $i = 1, \ldots, n$. Then the Monte Carlo estimator of by the $\mathbb{E}_Q(\max\{S_T - K, 0)\})$ method of antithetic variable is given by

$$\frac{1}{2n}\sum_{i=1}^{n}(\max\{S_T^i(z_1^i, \ldots, z_T^i) - K, 0\} + \max\{S_T^i(-z_1^i, \ldots, -z_T^i) - K, 0\})$$

**Solution:**

I simulated 10000 times to estimate the price of the call option. In fact, each time we get different simulation results. The results I posted for $S_0 = 0.9, 1$ and 1 are 0.2120569, 0.2819061 and 0.34341 respectively. I did several simulations for each $S_0$. For $S_0 = 0.9$, the price is approximately from 0.20 to 0.22. For $S_0 = 1$, the price is approximately from 0.27 to 0.29. For $S_0 = 1.1$, the price is approximately from 0.33 to 0.35. The result coincidence our expectation that the price of a call option increases as the underlying initial stock price increases. They are positively correlated.

The R code is [1]

```
n<-10000
# simulate n times
K<-matrix(1, 1, n)
```

[1] There may be some bug or problem with the R program in my laptop. Here, I have to write the code for the second row of the matrix. If not and use the for loop "for (i in 2:T+1)", the value of all elements in the second row become zero, which generate wrong results. However, the idea and logic of the code should be correct and reasonable.

```
# strike price matrix
T<-30
# maturity
r<-(0.05/252)
# risk-free interest rate
S0<-1.1
# initial stock price 0.9, 1, and 1.1
Z<-matrix(0, T+1, n)
# standard normal variable matrix
Z1<-matrix(0, T+1, n)
# equals -Z
Sig<-matrix(0, T+1, n)
# variance matrix corresponding Z
Sig1<-matrix(0, T+1, n)
# variance matrix corresponding Z1
Y<-matrix(0, T+1, n)
# log price ratio matrix corresponding Z
Y1<-matrix(0, T+1, n)
# log price ratio matrix corresponding Z1
PR<-matrix(0, T+1, n)
# price ratio matrix corresponding Z
PR1<-matrix(0, T+1, n)
# price ratio matrix corresponding Z1
S<-matrix(0, T+1, n)
# price matrix corresponding Z
S1<-matrix(0, T+1, n)
# price matrix corresponding Z1
for (j in 1:n)
{
Z[,j]<-rnorm(T+1)
Z1[,j]<- -Z[,j]
Sig[1,j]<-(0.24)^2
Sig1[1,j]<-(0.24)^2
Sig[2,j] = 0.00001524+0.7162*Sig[1,j]+0.1883*Sig[1,j]*(Z[1,j]
-0.007452)^2
Sig1[2,j] = 0.00001524+0.7162*Sig1[1,j]+0.1883*Sig1[1,j]*(Z1[1,j]
-0.007452)^2
S[1,j]<-S0
S1[1,j]<-S0
Y[1,j]<- r-(1/2)*Sig[1,j]+sqrt(Sig[1,j])*Z[1,j]
Y1[1,j]<- r-(1/2)*Sig1[1,j]+sqrt(Sig1[1,j])*Z1[1,j]
Y[2,j]<- r-(1/2)*Sig[2,j]+sqrt(Sig[2,j])*Z[2,j]
Y1[2,j]<- r-(1/2)*Sig1[2,j]+sqrt(Sig1[2,j])*Z1[2,j]
PR[1,j]<-exp(Y[1,j])
PR1[1,j]<-exp(Y1[1,j])
```

```
PR[2,j]<-exp(Y[2,j])
PR1[2,j]<-exp(Y1[2,j])
S[2,j]<-S[1,j]*PR[1,j]
S1[2,j]<-S1[1,j]*PR1[1,j]

for (i in 2:T+1)
{
Sig[i,j] = 0.00001524+0.7162*Sig[i-1,j]+0.1883*Sig[i-1,j]*(Z[i-1,j]
-0.007452)^2
Sig1[i,j] = 0.00001524+0.7162*Sig1[i-1,j]+0.1883*Sig1[i-1,j]*(Z1[i-1,j]
-0.007452)^2
Y[i,j]<- r-(1/2)*Sig[i,j]+sqrt(Sig[i,j])*Z[i,j]
Y1[i,j]<- r-(1/2)*Sig1[i,j]+sqrt(Sig1[i,j])*Z1[i,j]
PR[i,j]<-exp(Y[i,j])
PR1[i,j]<-exp(Y1[i,j])
S[i,j]<-S[i-1,j]*PR[i-1,j]
S1[i,j]<-S1[i-1,j]*PR1[i-1,j]
}
}
# simulate the GARCH-M process n times
max<-rep(0,n)
max1<-rep(0,n)
for (j in 1:n)
{
max[j]<-max(S[T+1,j]-K,0)
max1[j]<-max(S1[T+1,j]-K,0)
}
EQ<-(1/(2*n))*sum(max+max1)
# estimate E_Q(max{S_T - K,0}) by antithetic method
C<-exp(-r*T)*EQ
# evaluate the price of the call option
> C
[1] 0.2120569
# one simulated result as S0=0.9
> C
[1] 0.2819061
# one simulated result as S0=1
> C
[1] 0.3425755
# one simulated result as S0=1.1
```

(b) Under the risk-neutral probability $Q$, $\mathbb{E}_Q[S_T] = e^{rT}S_0$. Simulate the GARCH-M process $n$ times and use $S_T$ as a control variable to evaluate the price of the call option for each of the three current prices. That is, denote by $S_T^i$ the $n$ prices at $T$, $i = 1, \ldots, n$, then the Monte Carlo estimator of $\mathbb{E}_Q(\max\{S_T - k, 0\})$ by using control variable $S_T$ is given by

$$\frac{1}{n}\sum_{i=1}^{n}(\max\{S_T^i - K, 0\} - \hat{a}[S_T^i - e^{rT}S_0])$$

where $\hat{a}$ is the ratio of $\text{cov}(\max\{S_T - k, 0\}, S_T)$ and $\text{var}(S_T)$, i.e.,

$$\hat{a} = \frac{\text{cov}(\max\{S_T - k, 0\}, S_T)}{\text{var}(S_T)}$$

**Solution:**

I also simulate 10000 times and use the same realizations of standard normal variables $Z$ to estimate the price of the call option with the control variable method. Again, each simulation has different results. The results I posted are 0.2118683, 0.2856275 and 0.3560922 when $S_0 = 0.9, 1$ and 1. Also, the price of the call option is positively correlated with the current price. As in the antithetic method, I did several simulations and found that the variance of the estimator with control variable method seems larger than that with antithetic method. However, we have to calculate both variances to get the answer.

The R code is

```
var<-var(S[T+1,])
cov<-cov(max,S[T+1,])
ES<-exp(r*T)*S0
a<- -cov/var
# compute the estimated ''a'' that minimizes the variance
EQCon<-(1/n)*sum(max-a*(S[T+1,]-ES))
# estimate E_Q(max{S_T - K,0)}) by control variable method
CCon<-exp(-r*T)*EQCon
# evaluate the price of the call option
> CCon
[1] 0.2118683
> CCon
[1] 0.2856275
> CCon
[1] 0.3560922
```

3. **Chapter 19 Exercise 1** This exercise uses daily BMW returns in the `bmwRet` data set in the `fEcofin` package. Assume that the returns are i.i.d., even though there may be some autocorrelation and volatility clustering is likely.
(a) Compute nonparametric estimates of VaR(0.01, 24 hours) and ES(0.01, 24 hours).
**Solution:**

Suppose the size of the current position $S$ is \$10000. Then the nonparametric estimate of VaR is $\widehat{VaR}^{np}(\alpha) = -S \times \hat{q}(\alpha) = 407.9757$, where $\hat{q}(\alpha)$ is the $\alpha$-quantile of the sample of historic returns. The estimate of ES is

$$\begin{aligned}\widehat{ES}^{np}(\alpha) &= -S \times \frac{\sum_{i=1}^{n} R_i I\left[R_i < \hat{q}(\alpha)\right]}{\sum_{i=1}^{n} I\left[R_i < \hat{q}(\alpha)\right]} \\ &= 564.9151.\end{aligned}$$

The R code is

```
> library(fEcofin)
> data(bmwRet)
> r <- bmwRet[,2]
> VaRnp <- -10000*quantile(r, 0.01)
> I <- as.numeric(r < quantile(r, 0.01))
> ESnp <- -10000*sum(r*I)/sum(I)
> VaRnp
1%
407.9757
> ESnp
[1] 564.9151
```

(b) Compute parametric estimates of VaR(0.01, 24 hours) and ES(0.01, 24 hours) assuming that the returns are **normally distributed.**
**Solution:**

The parametric estimate of VaR ($\alpha$) is $\widehat{VaR}^{par}(\alpha) = -S \times F^{-1}(\alpha|\hat{\theta})$, where $F^{-1}(\alpha|\hat{\theta})$ is an estimate of the $\alpha$-quantile of the return distribution. The estimate of expected shortfall is

$$\widehat{ES}^{par}(\alpha) = -\frac{S}{\alpha} \times \int_{-\infty}^{F^{-1}(\alpha|\hat{\theta})} x f(x|\hat{\theta}) dx.$$

Therefore, when the returns are normally distributed, $\widehat{VaR}^{nor}(\alpha) = -S \times F^{-1}(\alpha|\hat{\theta}) = -S \times [\hat{\mu} + q_{\alpha,z}\hat{\sigma}] = 339.8577$. The expected shortfall is

$$\begin{aligned}\widehat{ES}^{nor}(\alpha) &= S \times \left\{-\hat{\mu} + \hat{\sigma}\left(\frac{\phi\left\{\Phi^{-1}(\alpha)\right\}}{\alpha}\right)\right\} \\ &= 389.8592,\end{aligned}$$

where $\phi$ and $\Phi$ are the standard normal density and CDF.

The R code is

```
> VaRnor <- -10000*qnorm(0.01, mean=mean(r), sd=sd(r))
> ESnor <- 10000*(-mean(r)+sd(r)*dnorm(qnorm(0.01))/0.01)
```

```
> VaRnor
[1] 339.8577
> ESnor
[1] 389.8592
```

(c) Compute parametric estimates of VaR(0.01, 24 hours) and ES(0.01, 24 hours) assuming that the returns are ***t*-distributed**.
**Solution:**

When the returns are $t$-distributed, $\widehat{VaR}^t(\alpha) = -S \times F^{-1}(\alpha|\hat{\theta}) = -S \times [\hat{\mu} + q_{\alpha,t}(\hat{\nu})\hat{\lambda}] = 420.0555$, where $q_{\alpha,t}(\hat{\nu})$ is the $\alpha$-quantile of the $t$ distribution with $\hat{\nu}$ degrees of freedom, and $\lambda$ is the scale parameter. In addition,

$$\begin{aligned}\widehat{ES}^t(\alpha) &= S \times \left\{-\hat{\mu} + \hat{\lambda}\left(\frac{f_{\hat{\nu}}\left\{F_{\hat{\nu}}^{-1}(\alpha)\right\}}{\alpha}\left[\frac{\hat{\nu} + \left\{F_{\hat{\nu}}^{-1}(\alpha)\right\}^2}{\hat{\nu}}\right]\right)\right\} \\ &= 648.8889\end{aligned}$$

The R code is

```
> library(MASS)
> fitt <- fitdistr(r,"t")
> param <- as.numeric(fitt$estimate)
> mean <- param[1]
> df <- param[3]
> sd <- param[2]*sqrt( (df)/(df-2) )
> lambda <- param[2]
> qt <- qt(0.01,df=df)
> VaRt <- -10000*(mean + lambda*qt)
> es1 <- dt(qt,df=df)/(0.01)
> es2 <- (df + qt^2) / (df - 1)
> es3 <- -mean+lambda*es1*es2
> ESt <- 10000*es3
> VaRt
[1] 420.0555
> ESt
[1] 648.8889
```

(d) Compare the estimates in (a), (b), and (c). Which do you feel are most realistic?
**Solution:**

The three different estimates of VaR and ES are summarized in below.

| Method | VaR | ES |
|---|---|---|
| Nonparametric | 407.9757 | 564.9151 |
| Parametric-Normal | 339.8577 | 389.8592 |
| Parametric-$t$ | 420.0555 | 648.8889 |

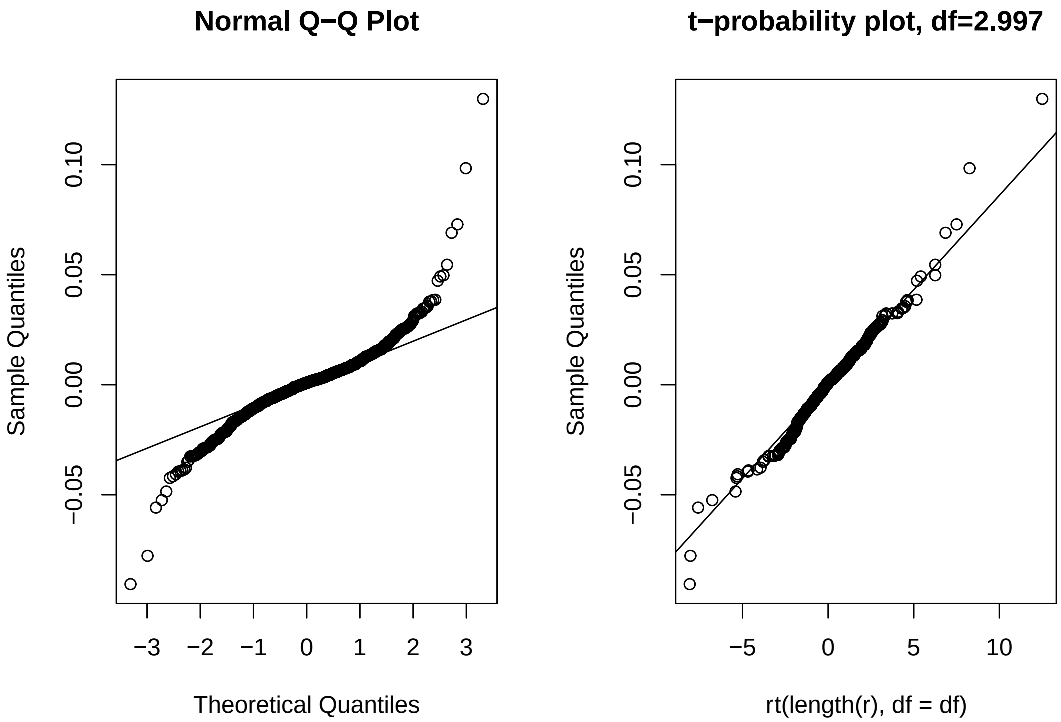


Figure 2: Normal QQ plot and $t$-plot with df = 2.997 of data bmwRet

Checking the Q-Q plots in Figure 2, we found that the returns have much heavier tails than a normal distribution, but are similar to $t$-distribution with degrees of freedom 2.997 except some positive extreme values. So the risk measures based on normality are not realistic, while the measures based on $t$-distribution are more convincing. Nonparametric estimation using sample quantiles works best when the sample size and $\alpha$ are reasonably large. Here, the sample size is 6146, which looks reasonable, but the $\alpha$ is only 0.01, which might be not large enough to use nonparametric estimation unless we increase our sample size further. Hence, if $\alpha$ is 0.05 or 0.1, I prefer the nonparametric method. But if $\alpha$ is set to be 0.01 or even smaller, I would rather choose parametric method based on $t$-distribution.

The R code is

```
> par(mfrow=c(1,2))
> qqnorm(r)
> qqline(r)
> qqplot(rt(length(r),df=df), r, ylab="Sample Quantiles", main="t-probability
plot, df=2.997")
> abline(lm(quantile(r,c(.25,.75))~qt(c(.25,.75),df=2.997)))
```

4. **Chapter 19 Exercise 2** Assume that the loss distribution has a polynomial tail and

an estimate of a is 3.1. If VaR(0.05) = \$252, what is VaR(0.005)?
**Solution:**

By Equation (19.20), $\text{VaR}(\alpha) = \text{VaR}(\alpha_0)\left(\frac{\alpha_0}{\alpha}\right)^{1/a}$. Hence, $\text{VaR}(0.005) = \text{VaR}(0.05)\left(\frac{0.05}{0.005}\right)^{1/3.1} = (252)(10)^{1/3.1} = 529.6405$
The R code is

```
> 252*(0.05/0.005)^(1/3.1)
[1] 529.6405
```

5. **Chapter 19 Exercise 3** Find a source of stock price data on the Internet and obtain daily prices for a stock of your choice over the last 1000 days.
(a) Assuming that the loss distribution is $t$, find the parametric estimate of VaR(0.025, 24 hours).
**Solution:**

I choose the stock price of Coca Cola offered by Yahoo Finance website to analyze this problem.[2] By considering the effect of any dividends distributions and corporate actions on stock price, I use the adjusted closing price to compute the stock return. The sample size is 1080, from 01/03/2008 to 04/16/2012. As in Exercise 1, I assume that the returns are i.i.d., even though there may be some autocorrelation and volatility clustering is likely. Moreover, we also assume the size of position is \$10000.

After running the R code, the VaR(0.025, 24 hours) under $t$-distribution is 282.9089.

The R code is

```
> coke<-read.csv("coke.csv", header=T)
> price<-coke[,7]
> logprice<-log(price)
> return<-diff(logprice)
> fittc <- fitdistr(return,"t")
> paramc <- as.numeric(fittc$estimate)
> meanc <- paramc[1]
> dfc <- paramc[3]
> sdc <- paramc[2]*sqrt( (dfc)/(dfc-2) )
> lambdac <- paramc[2]
> qtc <- qt(0.025,df=dfc)
> VaRtc <- -10000*(meanc + lambdac*qtc)
> VaRtc
[1] 282.9089
```

(b) Find the nonparametric estimate of VaR(0.025, 24 hours).
**Solution:**

[2] The website is http://finance.yahoo.com/. The registered stock name is The Coca-Cola Company (KO) traded in New York Stock Exchange (NYSE).

After running the R code, the nonparametric estimate of VaR(0.025, 24 hours) is 291.4614.

The R code is

```
> VaRnpc <- -10000*quantile(return, 0.025)
> VaRnpc
2.5%
291.4614
```

(c) Use a $t$-plot to decide if the normality assumption is reasonable.
**Solution:**

Figure 3 shows that the returns of Coca Cola company perform more like $t$-distribution than normal distribution.[3] The returns have heavier tails than the normal distribution, but the t-distribution fits the data pretty well except for some observations in the right tails. Hence, the normality assumption is not reasonable for the stock returns of Coca Cola company.

The R code is

```
> n = length(return)
> grid = (1:n)/(n+1)
> par(mfrow=c(1,2))
> qqplot(return, qnorm(grid), xlab="data coke", ylab="Normal-quantiles",
main="Normal qq plot")
> abline(lm(qnorm(c(.25,.75))~quantile(return,c(.25,.75))))
> qqplot(return, qt(grid,df=dfc), xlab="data coke", ylab="t-quantiles",
main="t-probability plot, df=2.889")
> abline(lm(qt(c(.25,.75),df=2.889)~quantile(return,c(.25,.75))))
```

[3] Note that in Figure 3, the X axis is data (sample quantiles) and Y axis is theoretical quantiles, but X axis represents theoretical quantiles and Y axis stands for sample quantiles in Figure 2. There is no particular reasons to use the presenting way of one or the other. The way of presenting in Figure 2 is the default setting of the R function `qqnorm`, but the way in Figure 2 is same as the way in our textbook.

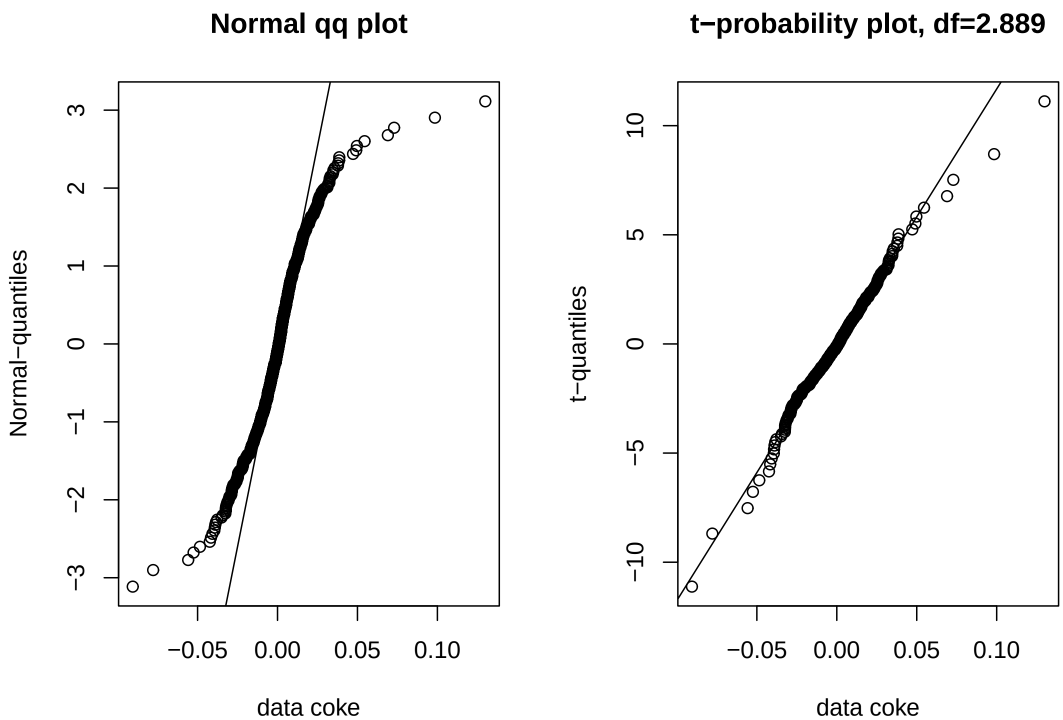


Figure 3: Normal QQ plot and $t$-plot with df = 2.889 of data coke