Financial Risk Forecasting
Seminar 9

Jon Danielsson
London School of Economics

Version 4.0 August 2024

9 Risk in derivatives

9.1 Links from the R notebook

Simulation methods for risk.

9.2 The plan for this session:

1. Simulation methods for risk in desivatives,

9.3 Loading data and libraries

```
library(mvtnorm)
```

We will use the `Prices.RData` and `Returns.RData` files we created in Seminar 2.

```
load('Returns.RData')
load('Prices.RData')
```

9.4 Black-Sholes equation

```
bs = function(K, P, r, sigma, Maturity){
  d1 = (log(P/K) +
    (r + 0.5*sigma^2)*(Maturity))/(sigma*sqrt(Maturity))
  d2 = d1 - sigma*sqrt(Maturity)
  Call = P*pnorm(d1) - K*exp(-r*(Maturity))*pnorm(d2)
  Put = K*exp(-r*(Maturity))*pnorm(-d2) - P*pnorm(-d1)
  return(list(Call=Call,Put=Put))
}
```

9.5 Setup

Set the probability, seed, maturity in years, the annual volatility and the annual risk free rate.

Note these are the constant parameters, the others will change.

```
Asset="JPM"
y=Returns[[Asset]]
seed=888
sigma=sd(y)
mean=0 # could be mean(y)
price=tail(Prices[[Asset]],1)
p=0.01
r=0.09
sigmaAnnual=sqrt(255)*sigma
```

9.6 Calculate VaR for stock

Use simulation to get the VaR of a stock, and compare it to the analytical calculations.

```
set.seed(seed=seed)
ysim=rnorm(S,mean=1+mean, sd=sigma)
Psim=price  * ysim
# or just
Psim=price  * rnorm(S,mean=1+mean, sd=sigma)
Q=Psim-price
Res=-sort(Q)[S* p]
cat(S,Res,-price * qnorm(p,sd=sigma),"\n")
```

9.7 Evaluate number of simulations

```
S=c(100,1000,10000,100000)
S=seq(10000,1e5,by=5000)
SimVaR=vector(length=length(S))
TrueVaR=-price * qnorm(p,sd=sigma)
for(i in 1:length(S)){
  ysim=rnorm(S[i],mean=1+mean, sd=sigma)
  Psim=price  * ysim
  Q=Psim-price
  SimVaR[i]=-sort(Q)[S[i]* p]
  #cat(S[i],SimVaR[i],TrueVaR,SimVaR[i]-TrueVaR,"\n")
}
plot(S,SimVaR)
segments(min(S),TrueVaR,max(S),TrueVaR,col="red")
```

9.8 Calculate VaR for a call option

If instead we have an option, we just have to use the Black-Scholes function to get both today's and the simulated future portfolio values.

```
K=price*0.9
Maturity=2
t =bs(K=K, P=price, r=r, sigma=sigma*sqrt(250), Maturity=Maturity)
S=1e5
Psim=price  * rnorm(S,mean=1+mean, sd=sigma)
Q=t$Call-bs(K=K, P=Psim, r=r, sigma=sigmaAnnual, Maturity=Maturity-1/365)$Call

Res=-sort(Q)[S* p]
Res
```

9.9 Calculate VaR for call option, put option and stock

```
K.call=price*0.9
Maturity.call=2
call =bs(K=K.call, P=price, r=r, sigma=sigmaAnnual, Maturity=Maturity.call)$Call

K.put=price*1.1
Maturity.put=0.7
put =bs(K=K.put, P=price, r=r, sigma=sigmaAnnual, Maturity=Maturity.put)$Put

Q = Psim+
  bs(K=K.call, P=Psim, r=r, sigma=sigma*sqrt(250), Maturity=Maturity.call-1/365)$Call +
  bs(K=K.call, P=Psim, r=r, sigma=sigma*sqrt(250), Maturity=Maturity.put-1/365)$Put -
  (call + put + price )

Res=-sort(Q)[S* p]
Res
```

9.10 Bivariate simulation

If we hold a portfolio of two stocks and options on each, we need to simulate from the bivariate normal distribution. In this case, we call `rmvnorm()`, but it would be straightforward to do it manually with a Choleski decomposition.

```
Assets=c("JPM","AAPL")
y=Returns[,Assets]
price=tail(Prices[,Assets],1)
Sigma=cov(y)
```

9.11 Simulate bivariate normal

It is then easy to use the number of simulations and the covariance matrix,

```
print(Sigma)
set.seed(seed)
r=rmvnorm(10,sigma=Sigma)
Prices1 = as.numeric(price[1])*(1+r[,1])
Prices2 = as.numeric(price[2])*(1+r[,2])
SimPrice=cbind(Prices1,Prices2)
print(r)
```

```
print(SimPrice)
print(cov(r))
print(cor(r))
print(price)
print(colMeans(SimPrice))
```

9.12 VaR

```
set.seed(seed)
r=rmvnorm(S,sigma=Sigma)
Prices1 = as.numeric(price[1])*(1+r[,1])
Prices2 = as.numeric(price[2])*(1+r[,2])
SimPrice=cbind(Prices1,Prices2)

today = sum(price)
tomorrow = rowSums(SimPrice)
Q = tomorrow - today
SimVaR = -sort(Q)[p*S]

w=as.numeric(price/sum(price))
s2= w %*% Sigma %*% w
s2=as.numeric(s2)
value=sum(price)
VaR=-qnorm(p) * value * sqrt(s2 )
cat("VaR = $",VaR,", SimVaR $",SimVaR,"\n",sep="")
```

9.13 Recap

In this seminar we have covered:

- Making VaR forecasts for derivatives

Some new functions used:

- `rmvnorm()`
- `colMeans()`

9.14 Optional exercises

2. Repeat the analysis with expected shortfall instead of VaR
3. Write code that allows you to put arbitrary number of simulations for pricing an option, and try it with 10 billion simulations.
4. Use simulation methods for getting the VaR of a portfolio of three stocks.
5. Use GARCH to get the conditional volatility each day, and use that to get the VaR.
6. Use functions to allow for the efficient specification of multiple options on each stock