

# Risk Homework2\_Jiaqi Li

*Jiaqi Li*

*April 15, 2019*

## Problem 1

### question 1

The Citigroup used the C-VaR to model the credit risk of the conduit's assets. The data the Citigroup used are from multiple resources include independent agencies and Citigroup's own information collected from the market. The method that the Citigroup used to compute VaR is to run Monte Carlo simulation and get the distribution of credit risk for the portfolios. Then, Citi applied probability-weighted basis on the portfolios to compute expected losses. Citi keeps monitor the credit characteristics of the conduit's asset and the credit environment to make sure that their C-VaR model is still working and producing useful information for the company to analyze the expected credit risk of the conduit's assets. In addition, Citigroup used 2-year sample to test their exceptions. In 2008, there are about 7-8 exceptions.

### question 2

```
library(quantmod)

## Warning: package 'quantmod' was built under R version 3.5.1
## Loading required package: xts
## Warning: package 'xts' was built under R version 3.5.1
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 3.5.1
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
## Loading required package: TTR
## Version 0.4-0 included new data defaults. See ?getSymbols.
getSymbols(Symbols = "C",from = "2006-01-01",
           to = "2008-12-31", src = "yahoo")

## '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).
## [1] "C"
```

part (a)

```
#-----part a-----#
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.1
library(data.table)

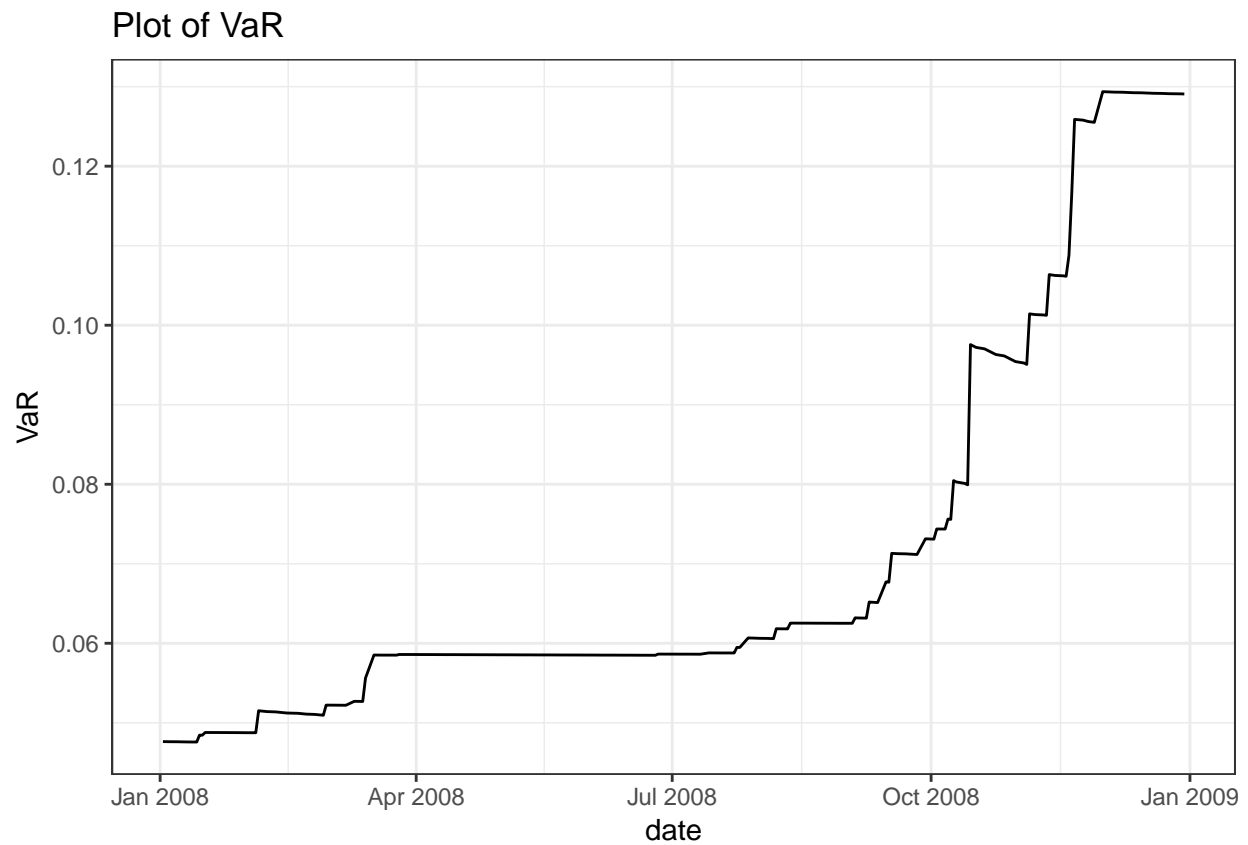
## Warning: package 'data.table' was built under R version 3.5.2
##
## Attaching package: 'data.table'
## The following objects are masked from 'package:xts':
##
##      first, last
C = as.data.table(C)

compute_VaR = function(C){
  n = length(C)
  PRC = as.data.frame(C)[,n]
  C = cbind(C,PRC)
  C[,Year := format(index,"%Y")]
  setkey(C,Year)

  C[, price_shift := shift(PRC)]
  C[, RET := PRC/price_shift-1]
  C = C[-1,]
  C[, Mean := sum(C$RET[1:.I])/.I]
  quant = c()
  for(i in 1:length(C$RET)){
    quant[i] = quantile(C$RET[1:i],0.01)
  }
  C = cbind(C,quant)
  C[, VaR := -(Mean + quant)]
  return(C)
}

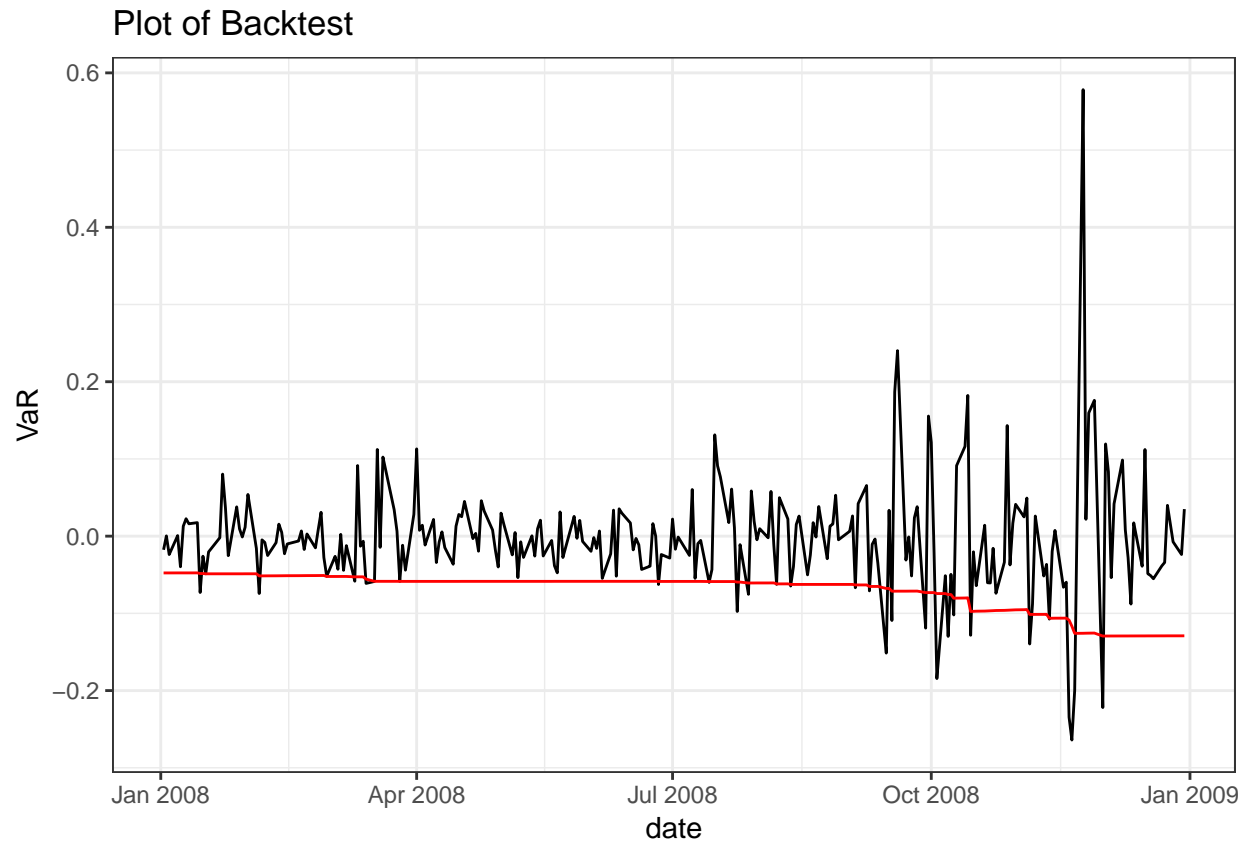
C = compute_VaR(C)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
qplot(C[.("2008")]$index,C[.("2008")]$VaR, geom = "line",
      xlab = "date", ylab = "VaR", main = "Plot of VaR") + theme_bw()
```



part (b)

```
#-----part b-----#
qplot(C[.("2008")]$index,C[.("2008")]$RET, geom = "line",
      xlab = "date", ylab = "VaR", main = "Plot of Backtest") +
  theme_bw() + geom_line(aes(y=-C[.("2008")]$VaR),color = "red")
```



```
C[, YesNoMaybe := ifelse(RET < -VaR, 1, 0)]
Exceed = sum(C[.("2008")]$YesNoMaybe)
setkey(C, Year)
n = length(C[.("2008")]$index)
Prob_exceed = Exceed/n
Exceed
```

```
## [1] 26
```

Based on the plot and the exceptions, we can say that this model does not work every well.

### part (c)

The Citigroup reported only 7-8 exceptions during 2008 by using their Model. I think the reason is that we used all the historial data instead of the most recent 1 year or 2 years data, which is what Citigroup did. The model may work better if we can add more weight on the recent data.

### question 2

#### part (a)

```
Bank = c("GS", "UBS", "JPM", "BCS", "DB", "BAC", "BNP.PA", "CS", "MS")
for(i in Bank){
  getSymbols(Symbols = i, from = "2006-01-01",
            to = "2008-12-31", src = "yahoo")
}
```

```

}
MS = as.data.table(MS)
CS = as.data.table(CS)
DB = as.data.table(DB)
GS = as.data.table(GS)
GS = as.data.table(GS)
JPM = as.data.table(JPM)
UBS = as.data.table(UBS)
BAC = as.data.table(BAC)
BCS = as.data.table(BCS)
BNP.PA = as.data.table(BNP.PA)

#-----part a-----#
MS = compute_VaR(MS)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
CS = compute_VaR(CS)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
DB = compute_VaR(DB)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
GS = compute_VaR(GS)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
JPM = compute_VaR(JPM)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
UBS = compute_VaR(UBS)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
BAC = compute_VaR(BAC)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
BCS = compute_VaR(BCS)

## Warning in 1:.I: numerical expression has 753 elements: only the first used
BNP.PA = compute_VaR(BNP.PA)

## Warning in 1:.I: numerical expression has 765 elements: only the first used
MS_M = MS[,VaR_m := 2000000*VaR]
CS_M = CS[,VaR_m := 2000000*VaR]
DB_M = DB[,VaR_m := 1000000*VaR]
GS_M = GS[,VaR_m := 1000000*VaR]
JPM_M = JPM[,VaR_m := 1000000*VaR]
UBS_M = UBS[,VaR_m := 2000000*VaR]
BAC_M = BAC[,VaR_m := 2000000*VaR]
BCS_M = BCS[,VaR_m := 1000000*VaR]
BNP.PA_M = BNP.PA[,VaR_m := 1000000*VaR]
C_M = C[,VaR_m := 2000000*VaR]

```

```

MS = MS[,list(date = index,RET_MS = RET)]
CS = CS[,list(date = index,RET_CS = RET)]
DB = DB[,list(date = index,RET_DB = RET)]
GS = GS[,list(date = index,RET_GS = RET)]
JPM = JPM[,list(date = index,RET_JPM = RET)]
UBS = UBS[,list(date = index,RET_UBS = RET)]
BAC = BAC[,list(date = index,RET_BAC = RET)]
BCS = BCS[,list(date = index,RET_BCS = RET)]
BNP.PA = BNP.PA[,list(date = index,RET_BNP.PA = RET)]
C = C[,list(date = index,RET_C = RET)]

setkey(MS,date); setkey(BNP.PA,date);setkey(CS,date); setkey(DB,date)
setkey(GS,date); setkey(JPM,date); setkey(UBS,date); setkey(BAC,date)
setkey(BCS,date); setkey(C,date)
Portfolio = merge(MS,BNP.PA,all = T)
Portfolio = merge(Portfolio,CS,all = T)
Portfolio = merge(Portfolio,DB,all = T)
Portfolio = merge(Portfolio,GS,all = T)
Portfolio = merge(Portfolio,JPM,all = T)
Portfolio = merge(Portfolio,UBS,all = T)
Portfolio = merge(Portfolio,BAC,all = T)
Portfolio = merge(Portfolio,BCS,all = T)
Portfolio = merge(Portfolio,C,all = T)
Portfolio = na.omit(Portfolio)
temp_p = as.matrix(Portfolio[,-1])

library(xts)
temp = as.xts(Portfolio)
weights = as.matrix(rep(c(1/15,2/15),5))
VaR_p = -(mean(temp_p %*% weights)+quantile(temp_p %*% weights,0.01))*15
print(VaR_p)

##          1%
## 1.480165

```

## part b

```

library(PortfolioAnalytics)

## Warning: package 'PortfolioAnalytics' was built under R version 3.5.1
## Loading required package: foreach
## Warning: package 'foreach' was built under R version 3.5.1
## Loading required package: PerformanceAnalytics
## Warning: package 'PerformanceAnalytics' was built under R version 3.5.1
##
## Attaching package: 'PerformanceAnalytics'
## The following object is masked from 'package:graphics':
##
##      legend

```

```
DVaR = VaR(temp,p = 0.99,sigma = cov(temp),portfolio_method = "marginal",weights = weights)
CVaR = VaR(temp,p = 0.99,sigma = cov(temp),portfolio_method = "component",weights = weights)
DVaR
```

```
## PortfolioVaR      RET_MS RET_BNP.PA      RET_CS      RET_DB
## 1      -0.127278 0.002879775 -0.02710726 -0.001130499 0.001165715
##      RET_GS      RET_JPM      RET_UBS      RET_BAC      RET_BCS
## 1 0.0007543858 0.0001094633 0.0006022385 0.008565362 0.0002335809
##      RET_C
## 1 0.00489096
```

```
CVaR$contribution
```

```
##      RET_MS RET_BNP.PA      RET_CS      RET_DB      RET_GS      RET_JPM
## 0.013360718 0.004589876 0.007866234 0.021129911 0.009524444 0.016194269
##      RET_UBS      RET_BAC      RET_BCS      RET_C
## 0.008901581 0.025038028 0.007893490 0.038208743
```

### part c

DVaR for an asset in the portfolio is the change in VaR caused by an \$1 increment in that asset. CVaR means the change in portfolio VaR caused by an 1% increase in position of that asset. For DVaR, the highest is Morgan Stanley and the smallest is BNP For CVaR, the highest is Citigroup and the smallest is UBS.