## Complex Financial Instruments

## CASE STUDY

## Bhrigu Bhanot

## Synopsis

Pricing options requires good estimates of expected volatility. Since 2004 traders can directly get an exposure to volatility by trading futures on so called volatility indices. As we will see in Lecture 8, volatility indices reflect the expected variance (under the risk-neutral measure). In this case-study we will investigate how to compute volatility indices, properties of volatility, and how to price derivatives on volatility.

## Objectives

The goal of this case study is to learn how volatility indices are computed and how volatility indices differ to common stocks or stock-indices as an underlying when pricing derivatives.

## Computations:

setwd("C:\\Users\\user\\Desktop\\CourseWork\\Complex Financial Instruments")  
  
#Load the required libraries  
library(reshape2)

## Warning: package 'reshape2' was built under R version 3.4.4

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.4

library(plyr)

## Warning: package 'plyr' was built under R version 3.4.3

source("case study function.R")

sp500\_options\_2010=read.csv("sp500\_options\_2010.csv",header=T)  
sp500\_options\_2010[,"MID"]=(sp500\_options\_2010[,"BID\_PRICE"]+sp500\_options\_2010[,"ASK\_PRICE"])/2  
sp500\_options\_2010[,"STRIKEPRICE"]=sp500\_options\_2010[,'STRIKEPRICE']/1000  
drop=c("UNDERLYING","IMPL\_VOLATILITY","DELTA","GAMMA","VEGA","THETA")  
sp500\_options\_2010=sp500\_options\_2010[,!(names(sp500\_options\_2010) %in% drop)]  
head(sp500\_options\_2010)

## Time BID\_PRICE ASK\_PRICE MATURITYDATE STRIKEPRICE OPTIONTYPE MID  
## 1 4-Jan-10 828 830.80 20100220 300 C 829.400  
## 2 4-Jan-10 0 0.05 20100220 300 P 0.025  
## 3 4-Jan-10 778 780.90 20100220 350 C 779.450  
## 4 4-Jan-10 0 0.10 20100220 350 P 0.050  
## 5 4-Jan-10 728 730.90 20100220 400 C 729.450  
## 6 4-Jan-10 0 0.10 20100220 400 P 0.050

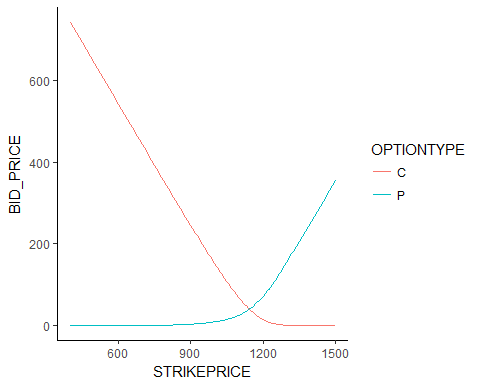
in\_r=0  
in\_date\_start="2010-01-14"  
in\_MATURITYDATE=20100417

## calculating days to expiry

sp500\_options\_2010[,"MATURITYDATE"]=as.numeric(sp500\_options\_2010[,"MATURITYDATE"])  
sp500\_options\_2010[,"date\_start"]=as.Date(sp500\_options\_2010[,"Time"],format="%e-%b-%y")  
sp500\_options\_2010[,"date\_end"]=as.Date(as.character(sp500\_options\_2010[,"MATURITYDATE"]),format="%Y%m%d")  
sp500\_options\_2010[,"days\_to\_expiry"]=sp500\_options\_2010[,"date\_end"]-sp500\_options\_2010[,"date\_start"]  
sp500\_options\_2010[,"days\_to\_expiry"]=as.numeric(sp500\_options\_2010[,"days\_to\_expiry"])

## Example option selection:All options on one date for 1 maturity

data=subset(sp500\_options\_2010,MATURITYDATE==in\_MATURITYDATE & date\_start==as.Date(in\_date\_start))  
ggplot(data,aes(x=STRIKEPRICE,y=BID\_PRICE)) + geom\_line(aes(color=OPTIONTYPE))+theme\_classic()



## Finding forward price and K0

atm\_strikes=sp500\_options\_2010[,c("date\_start","MATURITYDATE","days\_to\_expiry","OPTIONTYPE","STRIKEPRICE","MID")]  
atm\_strikes=dcast(atm\_strikes,formula=date\_start+MATURITYDATE+days\_to\_expiry+STRIKEPRICE~OPTIONTYPE,value.var = "MID")  
  
atm\_strikes[,"CmP"]=atm\_strikes[,"C"]-atm\_strikes[,"P"]  
forwards=get\_forwards(atm\_strikes,in\_r)  
head(forwards[,-(5:6)])

## date\_start MATURITYDATE days\_to\_expiry STRIKEPRICE CmP Forward  
## 1 2010-01-04 20100220 47 300 829.375 1129.7  
## 2 2010-01-04 20100220 47 350 779.400 1129.7  
## 3 2010-01-04 20100220 47 400 729.400 1129.7  
## 4 2010-01-04 20100220 47 450 679.400 1129.7  
## 5 2010-01-04 20100220 47 475 654.450 1129.7  
## 6 2010-01-04 20100220 47 500 629.450 1129.7

kzero=get\_kzero(forwards)  
  
head(kzero[,-(5:7)])

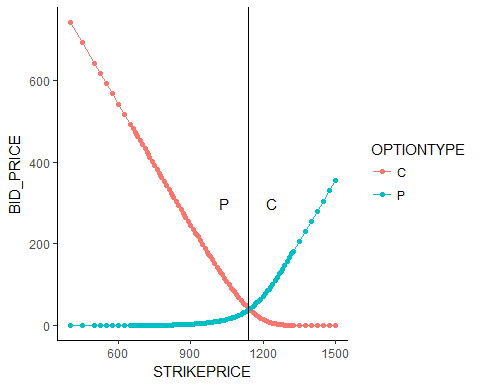
## date\_start MATURITYDATE days\_to\_expiry STRIKEPRICE Forward KmF KZERO  
## 1 2010-01-04 20100220 47 1125 1129.70 4.70 1125  
## 2 2010-01-04 20100320 75 1125 1128.05 3.05 1125  
## 3 2010-01-04 20100331 86 1100 1128.00 28.00 1100  
## 4 2010-01-04 20100417 103 1125 1127.30 2.30 1125  
## 5 2010-01-04 20100619 166 1100 1123.55 23.55 1100  
## 6 2010-01-04 20100630 177 1100 1123.10 23.10 1100

sp500\_options\_2010=merge(sp500\_options\_2010,kzero[,c("date\_start","MATURITYDATE","Forward","KZERO")],by=c("date\_start","MATURITYDATE"))  
  
head(sp500\_options\_2010)

## date\_start MATURITYDATE Time BID\_PRICE ASK\_PRICE STRIKEPRICE  
## 1 2010-01-04 20100220 4-Jan-10 828 830.80 300  
## 2 2010-01-04 20100220 4-Jan-10 0 0.05 300  
## 3 2010-01-04 20100220 4-Jan-10 778 780.90 350  
## 4 2010-01-04 20100220 4-Jan-10 0 0.10 350  
## 5 2010-01-04 20100220 4-Jan-10 728 730.90 400  
## 6 2010-01-04 20100220 4-Jan-10 0 0.10 400  
## OPTIONTYPE MID date\_end days\_to\_expiry Forward KZERO  
## 1 C 829.400 2010-02-20 47 1129.7 1125  
## 2 P 0.025 2010-02-20 47 1129.7 1125  
## 3 C 779.450 2010-02-20 47 1129.7 1125  
## 4 P 0.050 2010-02-20 47 1129.7 1125  
## 5 C 729.450 2010-02-20 47 1129.7 1125  
## 6 P 0.050 2010-02-20 47 1129.7 1125

## example option selection: indicating where to choose put and call options

data=subset(sp500\_options\_2010,MATURITYDATE==in\_MATURITYDATE & date\_start==as.Date(in\_date\_start))  
kzero=data[1,"KZERO"]  
  
ggplot(data,aes(x=STRIKEPRICE,y=BID\_PRICE))+geom\_line(aes(color=OPTIONTYPE))+geom\_point(aes(color=OPTIONTYPE))+geom\_vline(xintercept=kzero)+annotate("text",x=kzero-100,y=300,label="P")+annotate("text",x=kzero+100,y=300,label="C")+theme\_classic()



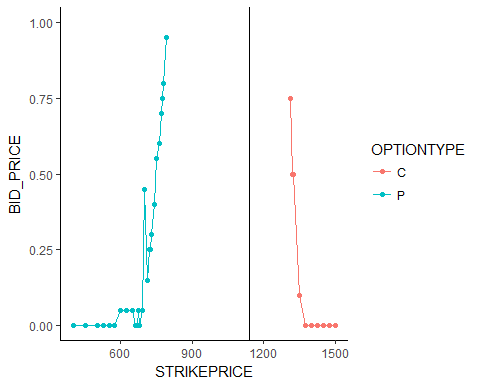
## example option selection: only using OTM options

vix\_options=subset(sp500\_options\_2010,(sp500\_options\_2010[,"OPTIONTYPE"]=="C" & STRIKEPRICE > KZERO)|(sp500\_options\_2010[,"OPTIONTYPE"]=="P" & STRIKEPRICE < KZERO))  
  
data=subset(vix\_options,MATURITYDATE==in\_MATURITYDATE & date\_start==as.Date(in\_date\_start))  
  
kzero=data[1,"KZERO"]  
  
ggplot(data,aes(x=STRIKEPRICE,y=BID\_PRICE))+geom\_line(aes(color=OPTIONTYPE))+geom\_point(aes(color=OPTIONTYPE))+geom\_vline(xintercept=kzero)+annotate("text",x=kzero-100,y=300,label="P")+annotate("text",x=kzero+100,y=300,label="C")+ylim(0,1)+theme\_classic()

## Warning: Removed 62 rows containing missing values (geom\_path).

## Warning: Removed 62 rows containing missing values (geom\_point).

## Warning: Removed 1 rows containing missing values (geom\_text).  
  
## Warning: Removed 1 rows containing missing values (geom\_text).



## Finding two consecutive zero bid prices

doublezerobids=get\_doublezerobids(vix\_options)  
  
vix\_options=merge(vix\_options,doublezerobids,by=c("Time","OPTIONTYPE","days\_to\_expiry"),all.x=T)

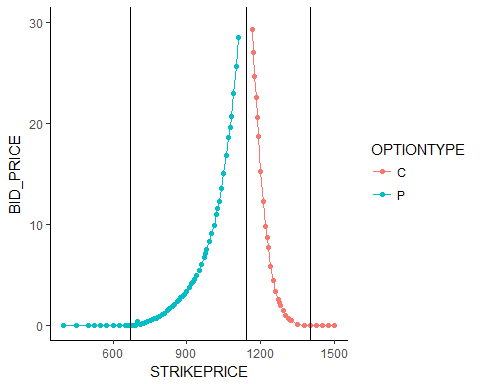
## Example option selection:indicating minimum and maximum strike prices

data=subset(vix\_options,MATURITYDATE==in\_MATURITYDATE & date\_start==as.Date(in\_date\_start))  
  
calls=subset(data,OPTIONTYPE=="C")  
puts=subset(data,OPTIONTYPE=="P")  
  
kzero=data[1,"KZERO"]  
calls=subset(data,OPTIONTYPE=="C")  
kmin=calls[1,"min\_k"]  
puts=subset(data,OPTIONTYPE=="P")  
kmax=puts[1,"max\_k"]  
  
ggplot(data,aes(x=STRIKEPRICE,y=BID\_PRICE))+geom\_line(aes(color=OPTIONTYPE))+geom\_point(aes(color=OPTIONTYPE))+geom\_vline(xintercept=kzero)+geom\_vline(xintercept=kmax)+geom\_vline(xintercept=kmin)+annotate("text",x=kzero-100,y=300,label="P")+annotate("text",x=kzero+100,y=300,label="C")+ylim(0,30)+theme\_classic()

## Warning: Removed 5 rows containing missing values (geom\_path).

## Warning: Removed 5 rows containing missing values (geom\_point).

## Warning: Removed 1 rows containing missing values (geom\_text).  
  
## Warning: Removed 1 rows containing missing values (geom\_text).



vix\_options=subset(vix\_options,(OPTIONTYPE=="P"&STRIKEPRICE>max\_k)|(OPTIONTYPE=="C"&STRIKEPRICE<min\_k))  
vix\_options=subset(vix\_options,BID\_PRICE>0)

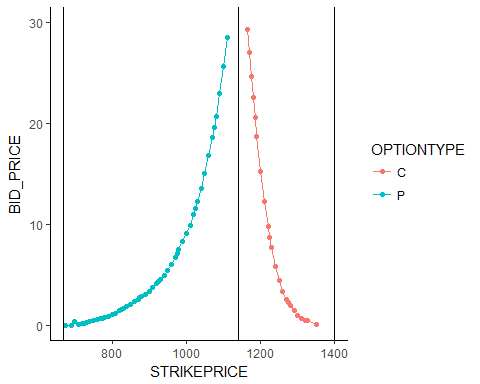
## Example option selection:final optoins used for calculating variance swap on given date and maurity

data=subset(vix\_options,MATURITYDATE==in\_MATURITYDATE & date\_start==as.Date(in\_date\_start))  
kzero=data[1,"KZERO"]  
calls=subset(data,OPTIONTYPE=="C")  
kmin=calls[1,"min\_k"]  
puts=subset(data,OPTIONTYPE=="P")  
kmax=puts[1,"max\_k"]  
  
ggplot(data,aes(x=STRIKEPRICE,y=BID\_PRICE))+geom\_line(aes(color=OPTIONTYPE))+geom\_point(aes(color=OPTIONTYPE))+geom\_vline(xintercept=kzero)+geom\_vline(xintercept=kmax)+geom\_vline(xintercept=kmin)+annotate("text",x=kzero-100,y=300,label="P")+annotate("text",x=kzero+100,y=300,label="C")+ylim(0,30)+theme\_classic()

## Warning: Removed 5 rows containing missing values (geom\_path).

## Warning: Removed 5 rows containing missing values (geom\_point).

## Warning: Removed 1 rows containing missing values (geom\_text).  
  
## Warning: Removed 1 rows containing missing values (geom\_text).



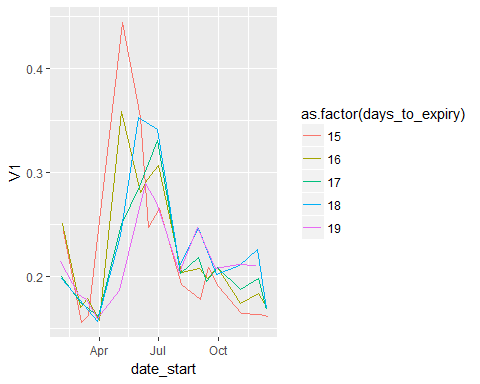
vix\_atm\_options=subset(sp500\_options\_2010,STRIKEPRICE==KZERO)  
vix\_options=vix\_options[,!(names(vix\_options)%in%c("min\_k","max\_k"))]  
vix\_options=rbind(vix\_atm\_options,vix\_options)  
  
vix\_options=vix\_options[order(vix\_options[,"date\_start"],vix\_options[,"MATURITYDATE"],vix\_options[,"STRIKEPRICE"]),]

vix\_options=get\_previousstrike(vix\_options)  
  
variance\_swap=ddply(vix\_options,.(date\_start,MATURITYDATE,days\_to\_expiry),function(x){  
 T=x[1,"days\_to\_expiry"]/365  
 F=x[1,"Forward"]  
 K=x[1,"KZERO"]  
 x[,"Delta"]=(x[,"STRIKEPRICE"]-x[,"LAST\_STRIKEPRICE"])/(x[,"STRIKEPRICE"]^2)  
 sum\_otm=sum(x[,"Delta"]\*x["MID"],na.rm=T)  
 price=(2/T)\*exp(in\_r\*T)\*sum\_otm-(1/T)\*((F/K-1)^2)  
 return(price)  
})

vix\_prices=variance\_swap  
vix\_prices[,"V1"]=sqrt(vix\_prices[,"V1"])

## Warning in sqrt(vix\_prices[, "V1"]): NaNs produced

ggplot(subset(vix\_prices,days\_to\_expiry<20),aes(x=date\_start,y=V1))+geom\_line(aes(color=as.factor(days\_to\_expiry)))



## code to find the 30 days VIX value

ttable=vix\_prices  
my\_vix\_30=ddply(ttable,.(date\_start),function(x){  
  
test\_start\_date <- x[1,"date\_start"] #extract the date from vix data using ddply  
  
  
x[,"N1\_N2"]=abs(x[,"days\_to\_expiry"]-30)   
x <- x[order(x[,"N1\_N2"]),] #arranging the table with the closest terms to 30 days expiries   
  
 near\_term<-x[1,"days\_to\_expiry"]  
Nt1<-abs(near\_term)  
T1<-x[1,"days\_to\_expiry"]/365  
sigma1<-x[1,"V1"]  
  
next\_term<-x[2,"days\_to\_expiry"]  
Nt2<-abs(next\_term)  
T2<-x[2,"days\_to\_expiry"]/365  
sigma2<-x[2,"V1"]  
  
  
N30<-30  
  
N365<-365  
 prod1<-T1\*(sigma1^2)\*((Nt2-N30)/(Nt2-Nt1))  
   
 prod2<-T2\*(sigma2^2)\*((N30-Nt1)/(Nt2-Nt1))  
 result<<-prod1+prod2  
  
myyvix<-100\*sqrt((result\*365)/N30)  
return(myyvix)   
})

## code to find the 90 days VIX value

ttable=vix\_prices  
ttable=na.omit(ttable)  
  
my\_vix\_90=ddply(ttable,.(date\_start),function(x){  
  
  
x[,"N1\_N2"]=abs(x[,"days\_to\_expiry"]-90)   
x <- x[order(x[,"N1\_N2"]),]   
  
  
 near\_term<-x[1,"days\_to\_expiry"]  
Nt1<-abs(near\_term)  
T1<-x[1,"days\_to\_expiry"]/365  
sigma1<-x[1,"V1"]  
  
next\_term<-x[2,"days\_to\_expiry"]  
Nt2<-abs(next\_term)  
T2<-x[2,"days\_to\_expiry"]/365  
sigma2<-x[2,"V1"]  
  
  
N90<-90  
  
N365<-365  
 x<-sigma1^2  
 prod1<-T1\*(sigma1^2)\*((Nt2-N90)/(Nt2-Nt1))  
   
 prod2<-T2\*(sigma2^2)\*((N90-Nt1)/(Nt2-Nt1))  
 result<<-prod1+prod2  
  
myyvix<-100\*sqrt((result\*365)/N90)  
return(myyvix)   
})

Appendix

# get\_forwards

get\_forwards=function(options,interest\_rate) { forwards=ddply(options,.(date\_start,MATURITYDATE),function(x){

x[which.min(abs(x[,"CmP"])),]

}) forwards[,“Forward”]=forwards[,“STRIKEPRICE”]+exp(interest\_rate*forwards[,“days\_to\_expiry”]/365)*forwards[,“CmP”] forwards=forwards[,c(“date\_start”,“MATURITYDATE”,“Forward”)] options=merge(options,forwards[,c(“date\_start”,“MATURITYDATE”,“Forward”)],by=c(“date\_start”,“MATURITYDATE”)) return(options) }

# get\_kzero

get\_kzero=function(options) { atm\_strikes=subset(options,Forward>=STRIKEPRICE) atm\_strikes[,“KmF”]=round((atm\_strikes[,“Forward”]-atm\_strikes[,“STRIKEPRICE”]),2) kzero=ddply(atm\_strikes,.(date\_start,MATURITYDATE),function(x){x[which.min(x[,“KmF”]),] }) kzero[,“KZERO”]=kzero[,“STRIKEPRICE”] return(kzero) }

# get\_doublezerobids

get\_doublezerobids=function(options){ options=.add\_lagged\_column(options,column\_name=“BID\_PRICE”,orderby=“STRIKEPRICE”) zero\_bids=options[,c(“Time”,“OPTIONTYPE”,“days\_to\_expiry”,“STRIKEPRICE”,“BID\_PRICE”,“LAST\_BID\_PRICE”)] zero\_bids=subset(zero\_bids,BID\_PRICE==0 & LAST\_BID\_PRICE==0) zero\_bids=ddply(zero\_bids,.(Time,OPTIONTYPE,days\_to\_expiry),summarize,min\_k=min(STRIKEPRICE),max\_k=max(STRIKEPRICE)) return(zero\_bids) }

# get\_previousstrike

get\_previousstrike=function(options){ options=.add\_lagged\_column(options,column\_name=“STRIKEPRICE”,orderby=“STRIKEPRICE”) return(options) }

# .add\_lagged\_column

.add\_lagged\_column=function(data,column\_name,orderby){ data=ddply(data,.(Time,days\_to\_expiry),function(x){ x=x[order(x[,orderby]),] n=length(row.names(x)) x[,paste(“LAST\_”,column\_name,sep = “”)]=c(NA,x[1:(n-1),column\_name]) return(x) }) return(data) }