Mid_term_exam

Mo Pei

10/23/2016

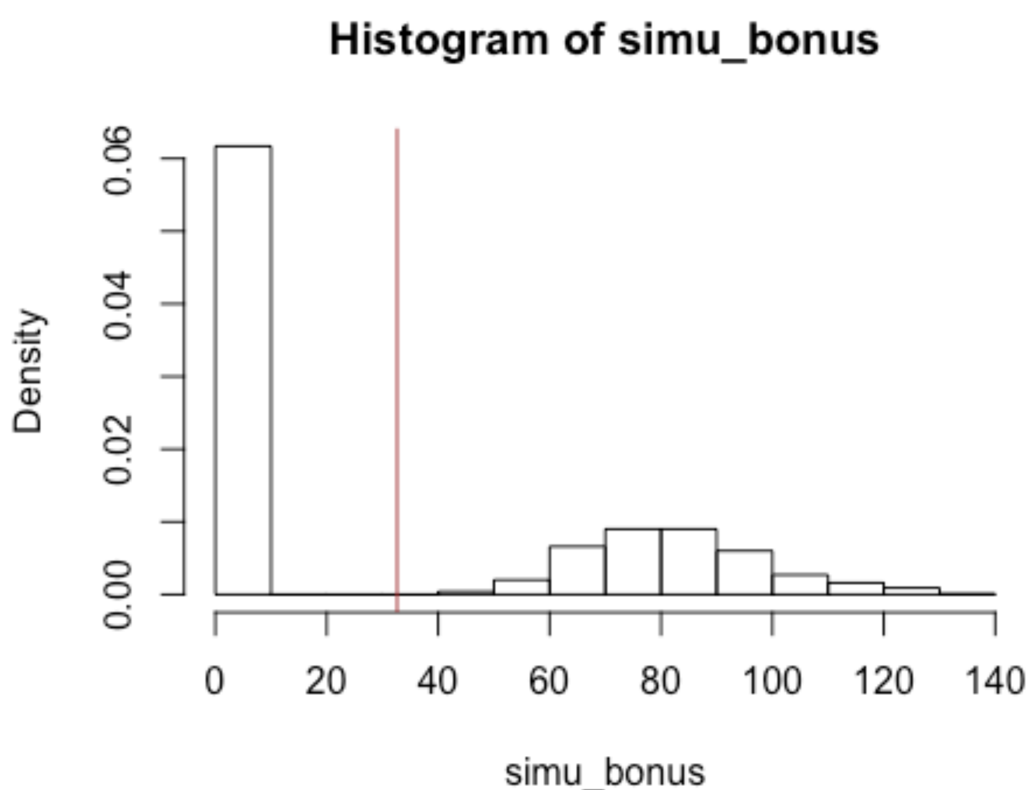
This file includes the R script and its results. The outcome of each R script is highlighted as bolded '**Result**' in order to read this document easier. Thank you!

```
#packages set up
# install.packages('moments')
# install.packages('boot')
# install.packages('logspline')
library(boot)
library(logspline)
library(quantmod)
```

Question 1.

```
n<-1000

simu_bonus<-1:n
for(i in 1:n)
{
  gs4<-sample(c('<=4','>4'),1,prob=c(0.6,0.4))

  if(gs4=='>4')
  {
    n_bi<-sample(c(5,6,7,8),1,prob=c(0.35,0.45,0.15,0.05))
```

```
    bonus<-1:n_bi

    for(j in 1:n_bi)
    {
    bonus[j]<-sample(c(10,15,20,25),1,prob=c(0.40,0.35,0.20,0.05))
    }
    simu_bonus[i]<-sum(bonus)
  }
  else
  {
    simu_bonus[i]<-0
  }

}

hist(simu_bonus,probability = TRUE)
abline(v=mean(simu_bonus),col='Brown')
```




```
mean_simu_bo<-mean(simu_bonus)
sd_simu_bo<-sd(simu_bonus)

cat('The 95% confidence interval of the expected value of bonus is bewteen
```

```
',mean_simu_bo-1.96*(sd_simu_bo/sqrt(n)) ,' and ',
mean_simu_bo+1.96*(sd_simu_bo/sqrt(n)),'\n')
```

Result:

```
the bonus a salesperson can expect in a day is 35.815 and standard deviation
is

The 95% confidence interval of the expected value of bonus is bewteen
29.98389  and  35.27611
```

Question 2.

a.

er=c(0.24, 0.15) covmat=matrix(c(0.32^2, 0.21*0.32*0.1, 0.21*0.32*0.1, 0.21^2),nrow=2,ncol=2) names(er)=c("Stock_fund","Bond_fund") colnames(covmat)=c("Stock_fund","Bond_fund") rownames(covmat)=c("Stock_fund","Bond_fund")

gmin.port <- globalMin.portfolio(er, covmat)

Result:

The risk and return of the minimum variance portfolio consisting of the stock and bond funds are below

Portfolio expected return: 0.1752833

Portfolio standard deviation: 0.1833003

Portfolio weights:

Stock_fund Bond_fund

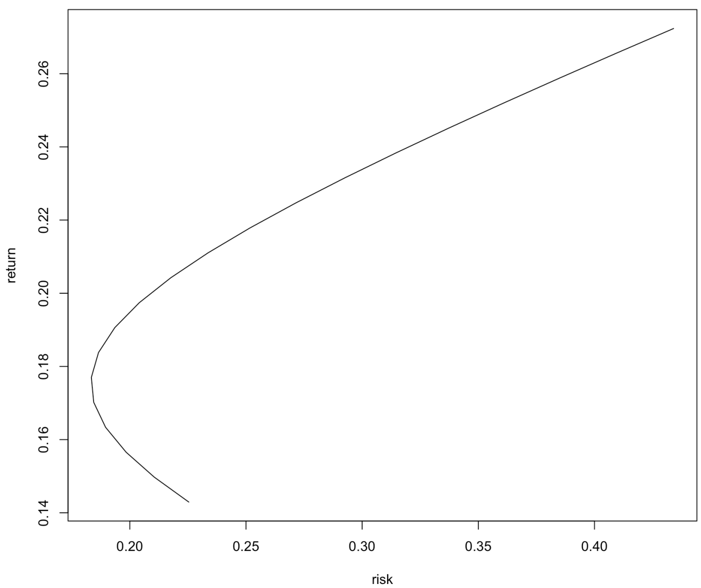
0.2809 0.7191

b.

ef=efficient.frontier(er,covmat) plot(efsd, efer,type="l",xlab="risk",ylab="return") msharp=0

Result:

A plot of the efficient frontier, allowing short sales.



c.

```
rk.free=.05
tan.port <- tangency.portfolio(er, covmat, rk.free)
print(tan.port)
summary(tan.port, risk.free=rk.free)
plot(tan.port)

plot(ef$sd,ef$er,type="l",xlab="risk",ylab="return")
abline(a=rk.free, b=0.7266561)
```

Result:

The weights for the tangent portfolio, as well as the risk and return of the tangent portfolio.

Portfolio expected return: 0.191609

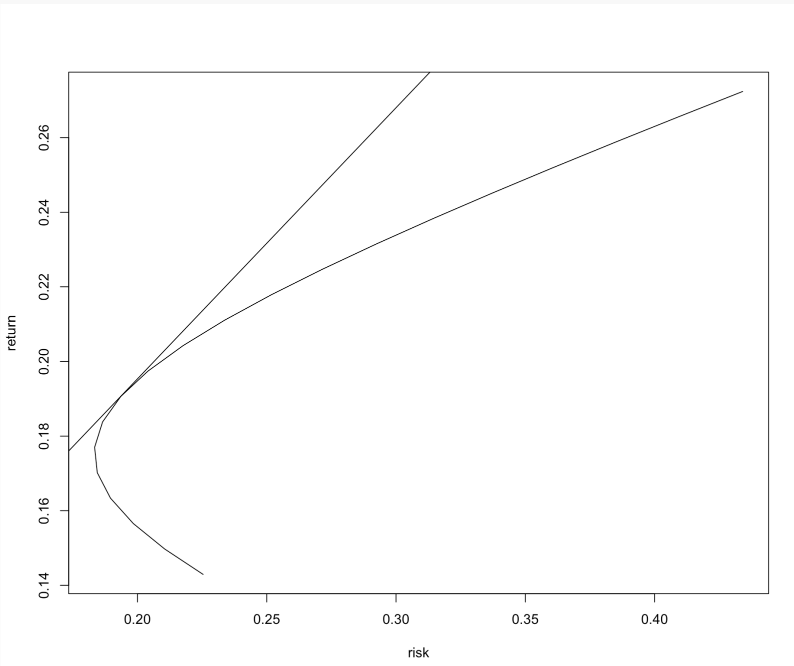
Portfolio standard deviation: 0.1948776

Portfolio Sharpe Ratio: 0.7266561

Portfolio weights:

Stock_fund Bond_fund

0.4623 0.5377



d.

Result:

When allowing short sell, the appropriate portfolio weights and the resulting portfolio standard deviation are below.

	Univariate Statistics		
	Stock Fund	**Bond Fund**	
Average	0.24	0.15	
Standard Devi	0.32	0.21	
Variance	0.1024	0.0441	
Covraiance matrix		**Stock Fund**	**Bond Fund**
	Stock Fund	0.1024	0.00672
	Bond Fund	0.00672	0.0441
w1	1.4444		
w2	-0.4444		
constraint	1.0000		
port mean	0.2800		
port variance	0.2137		
port sd	0.4623		

The reduction in standard deviation could you attain if instead you used the money market fund and the tangent portfolio to construct a portfolio that returned .28

Firstly, we maximized the sharpe ratio, and then formed a new portfolio

that has risk and risky assets. From the results below, we can conclude that the sd of the portfolio is decreased by 0.15.

		sharpe	0.7266561
constraint	1.0000		
port mean	0.1916	w_risk	1.6241976
port variance	0.0380	W_risk_free	-0.624197
port sd	0.1949		
		constraint2	1.000001
Money Marke	0.0500		
		port_mean	0.280001
		port_varianc	0.1001847
sharpe	0.7266561	port_sd	0.3165197

sharpe	0.7266561
w_risk	1.6241976
W_risk_free	-0.624197
constraint2	1.000001
port_mean	0.280001
port_varianc	0.1001847
port_sd	0.3165197

Question 3.

a.

Result:

```
mikeline_new=function(x,y) {
xmed=median(x)
xbar1=median(x[x<=xmed])
ybar1=median(y[x<=xmed])
xbar2=median(x[x>xmed])
ybar2=median(y[x>xmed])
slope = (ybar2-ybar1)/(xbar2-xbar1)
inter = ybar1-slope*xbar1
cat("Intercept = ",inter," Slope = ",slope,"\n")
return(slope)
}
```

b.

```
getSymbols("IBM", from = "2013-09-01")

getSymbols("SPY", from="2013-09-01")

spyret=as.numeric(monthlyReturn(SPY))
ibmret=as.numeric(monthlyReturn(IBM))

fit<-lm(ibmret~spyret)
summary(fit)
```

```
##
## Call:
## lm(formula = ibmret ~ spyret)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.142225 -0.021226  0.000059  0.023327  0.118702
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.009522   0.008012  -1.189   0.2424
## spyret       0.755184   0.254412   2.968   0.0053 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04804 on 36 degrees of freedom
## Multiple R-squared:  0.1966, Adjusted R-squared:  0.1743
## F-statistic: 8.811 on 1 and 36 DF,  p-value: 0.005297

cat('The valueu of Beta by using least squares :',coef(fit)[2] ,'\n')
```

Result:

```
The valueu of Beta by using least squares : 0.7551839

new_slope<-mikeline_new(spyret,ibmret)

## Intercept =  -0.00698411  Slope =  0.4361206

cat('The valueu of Beta by using modified algorithm : ' ,new_slope,'. The
value is off by ',coef(fit)[2] , 'the value of least square method.' ,'\n')
```

Result:

```
## The valueu of Beta by using modified algorithm :  0.4361206 . The value is
off by  0.7551839 the value of least square method.



c.
```

Result:

```
myfunc<-function(data,i) {

x=data[i,1]
y=data[i,2]

xmed=median(x)
```

```
xbar1=median(x[x<=xmed])
ybar1=median(y[x<=xmed])
xbar2=median(x[x>xmed])
ybar2=median(y[x>xmed])
slope = (ybar2-ybar1)/(xbar2-xbar1)
# inter = ybar1-slope*xbar1
# cat("Intercept = ",inter," Slope = ",slope,"\n")
return(slope)
}

mydata=cbind(spyret,ibmret)
bfit=boot(mydata,myfunc,R=1000)
boot.ci(bfit)

## Warning in boot.ci(bfit): bootstrap variances needed for studentized
## intervals

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bfit)
##
## Intervals :
## Level      Normal              Basic
## 95%   (-0.6369,  1.3011 )   (-0.9008,  1.0360 )
##
## Level     Percentile            BCa
## 95%   (-0.1637,  1.7731 )   (-0.1165,  1.8392 )
## Calculations and Intervals on Original Scale
```

Result:

```
cat('The bootstrap 95% confidence interval is between
',boot.ci(bfit)$percent[4],' and ',boot.ci(bfit)$percent[5])

## The bootstrap 95% confidence interval is between  -0.163711  and  1.773088
```

Question 4.

a.

```
mystocks=read.csv("http://people.fas.harvard.edu/~mparzen/stat107/dow30.csv",
header=FALSE,colClasses="character")
```

Result:

```
getSymbols("SPY",from="2013-09-01",to ="2016-09-01")
```

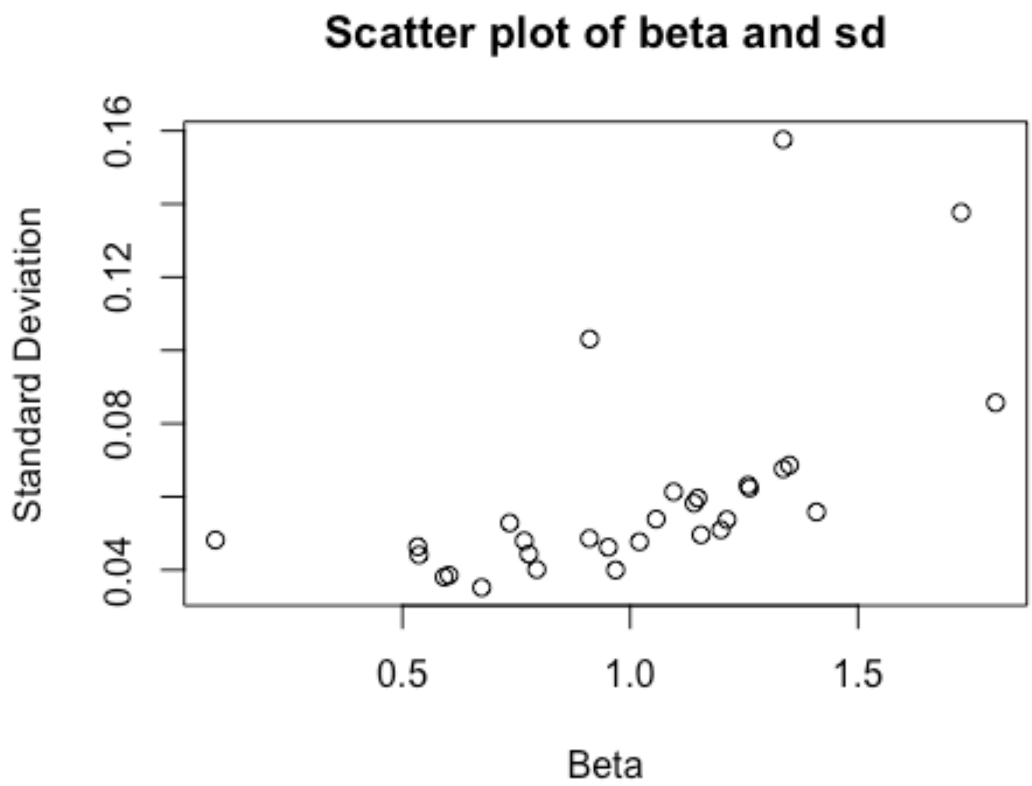
```
## [1] "SPY"

spyret=as.numeric(monthlyReturn(SPY))

all_beta<-1:nrow(mystocks)
all_sd<-1:nrow(mystocks)
for(i in 1:length(all_beta))
{
  stock_obj<-getSymbols(mystocks[i,1], from = "2013-09-01",to ="2016-09-
01",auto.assign=FALSE)
  stock_obj_ret=as.numeric(monthlyReturn(stock_obj))

  fit<-lm(stock_obj_ret~spyret)
  all_beta[i]<-coef(fit)[2]
  all_sd[i]<-sd(stock_obj_ret)
}

cat('The beta of the stocks in mystocks is ',all_beta,'\n')

## The beta of the stocks in mystocks is  1.261771 1.335803 1.258655 1.335971
1.140272 1.148486 1.802246 0.7775952 1.199146 1.212417 1.02034 0.7355329
1.095351 0.6740743 0.794923 1.05718 0.6023883 0.9681937 0.7674213 1.349803
0.9111244 0.9519475 0.5915899 1.155403 0.5337071 0.910551 0.5362807 1.726484
0.08965563 1.408918



b.
```

Result:

```
cat('The standard deviation of the stocks in mystocks is ',all_beta,'\n')

## The standard deviation of the stocks in mystocks is  1.261771 1.335803
1.258655 1.335971 1.140272 1.148486 1.802246 0.7775952 1.199146 1.212417
1.02034 0.7355329 1.095351 0.6740743 0.794923 1.05718 0.6023883 0.9681937
0.7674213 1.349803 0.9111244 0.9519475 0.5915899 1.155403 0.5337071 0.910551
0.5362807 1.726484 0.08965563 1.408918



c.
```

Result:

```
plot(all_beta,all_sd,main = 'Scatter plot of beta and sd',xlab = 'Beta',ylab
= 'Standard Deviation')
```




```
d.
```

Result:

```
cor(all_sd,all_beta)

## [1] 0.5762182

cat('Beta and standard deviation are positively correlated. The larger Beta
means more variation when the index changes')

## Beta and standard deviation are positively correlated. The larger Beta
means more variation when the index changes



e.
```

Result:

```
fit_beta_sd<-lm(all_sd~all_beta)
summary(fit_beta_sd)

##
## Call:
```

```
## lm(formula = all_sd ~ all_beta)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.021704 -0.010585 -0.006993 -0.001862  0.083242
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01613    0.01255   1.285 0.209200
## all_beta     0.04359    0.01168   3.731 0.000861 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0231 on 28 degrees of freedom
## Multiple R-squared:  0.332,  Adjusted R-squared:  0.3082
## F-statistic: 13.92 on 1 and 28 DF,  p-value: 0.0008609
```

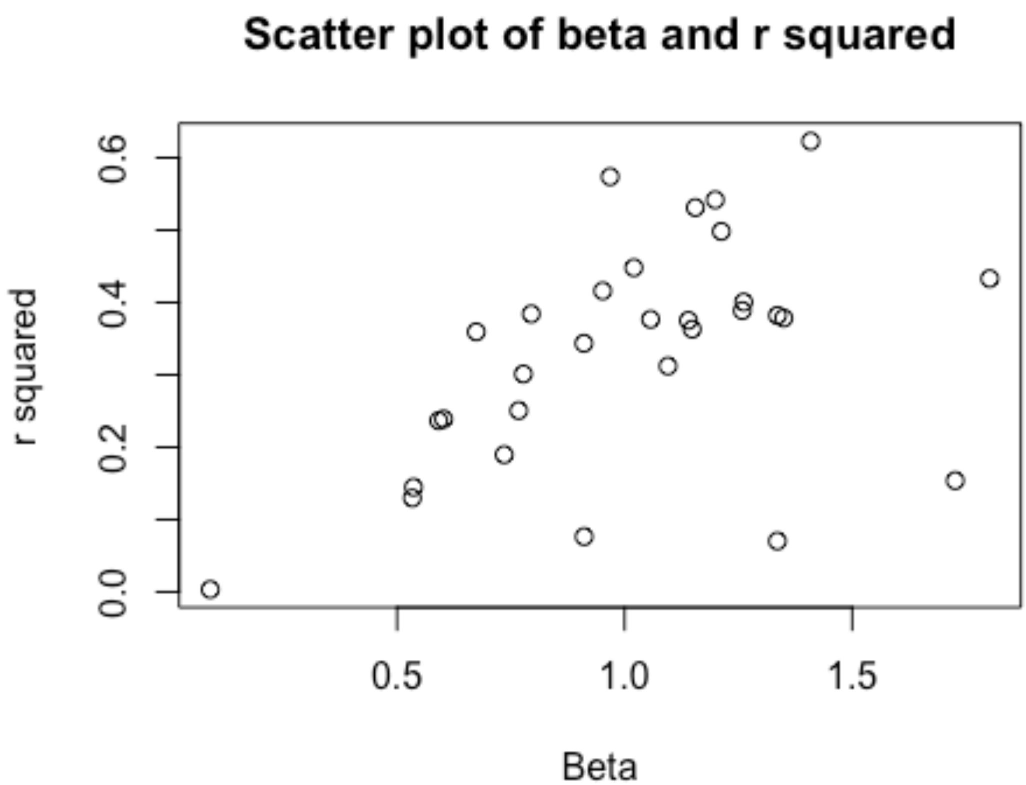
Result:

```
cat('According to the regression output, Intercept p value is 0.209200 which
is not significant, and slope p value is 0.000861 is smaller than 0.05 which
is significant.')

## According to the regression output, Intercept p value is 0.209200 which is
not significant, and slope p value is 0.000861 is smaller than 0.05 which is
significant.
```

Result:

```
cat('According to the regression output, 33.2% of
standard deviation’s variability is explained by Beta
movements.')

## According to the regression output, 33.2% of
## standard deviation’s variability is explained by Beta
## movements.



f.

all_beta<-1:nrow(mystocks)
all_rsquare<-1:nrow(mystocks)
for(i in 1:length(all_beta))
{
  stock_obj<-getSymbols(mystocks[i,1], from = "2013-09-01",to ="2016-09-
01",auto.assign=FALSE)
  stock_obj_ret=as.numeric(monthlyReturn(stock_obj))

  fit<-lm(stock_obj_ret~spyret)
```

```
  all_beta[i]<-coef(fit)[2]
  all_rsquare[i]<-summary(fit)$r.squared
}

plot(all_beta,all_rsquare,main = 'Scatter plot of beta and r squared',xlab =
'Beta',ylab = 'r squared')
```



Result:

```
cat('For single factor linear model, the R squred expresses the amount of
variation explained by x. The rise of Beta would increase the variablity of
x')

## For single factor linear model, the R squred expresses the amount of
variation explained by x. The rise of Beta would increase the variablity of x
```

Question 5.

Since Cov(Ri,Rj) = sigma_m^2 * beta_i * beta_j

Cov(Rp,Rm) = sigma_m^2 * beta_p * beta_m

beta_m=1

Cov(Rp,Rm) = sigma_m^2 * beta_p

beta_p = Cov(Rp,Rm) / sigma_m^2

Since Covariance property below,

a. $\mathrm{cov}(X+Y,Z)=\mathrm{cov}(X,Z)+\mathrm{cov}(Y,Z)$
b. $\mathrm{cov}(cX,Y)=c\mathrm{cov}(X,Y)$

Cov(Rp,Rm)=Cov(xa*Ra+xb*Rb,Rm)

=Cov(xa*(αA +βA*RM +εA)+xb*(αB +βB*RM +εB), Rm)

=Cov(xa*αA+xb* αB, Rm)+Cov(xa*βA*Rm+xb*βB*Rm, Rm)+ Cov(xa*εA+xb*εB,Rm)

= 0 + Cov(xa*βA*Rm+xb*βB*Rm, Rm)+0

= Cov(Rm(xa*βA+xb*βB),Rm)

=(xa*βA+xb*βB)Cov(Rm,Rm)

Since Cov(Rm,Rm)=Var(Rm)

Cov(Rp,Rm) =(xa*βA+xb*βB) Var(Rm)

beta_p =(xa*βA+xb*βB)Var(Rm) / sigma_m^2

beta_p= xa*βA+xb*βB