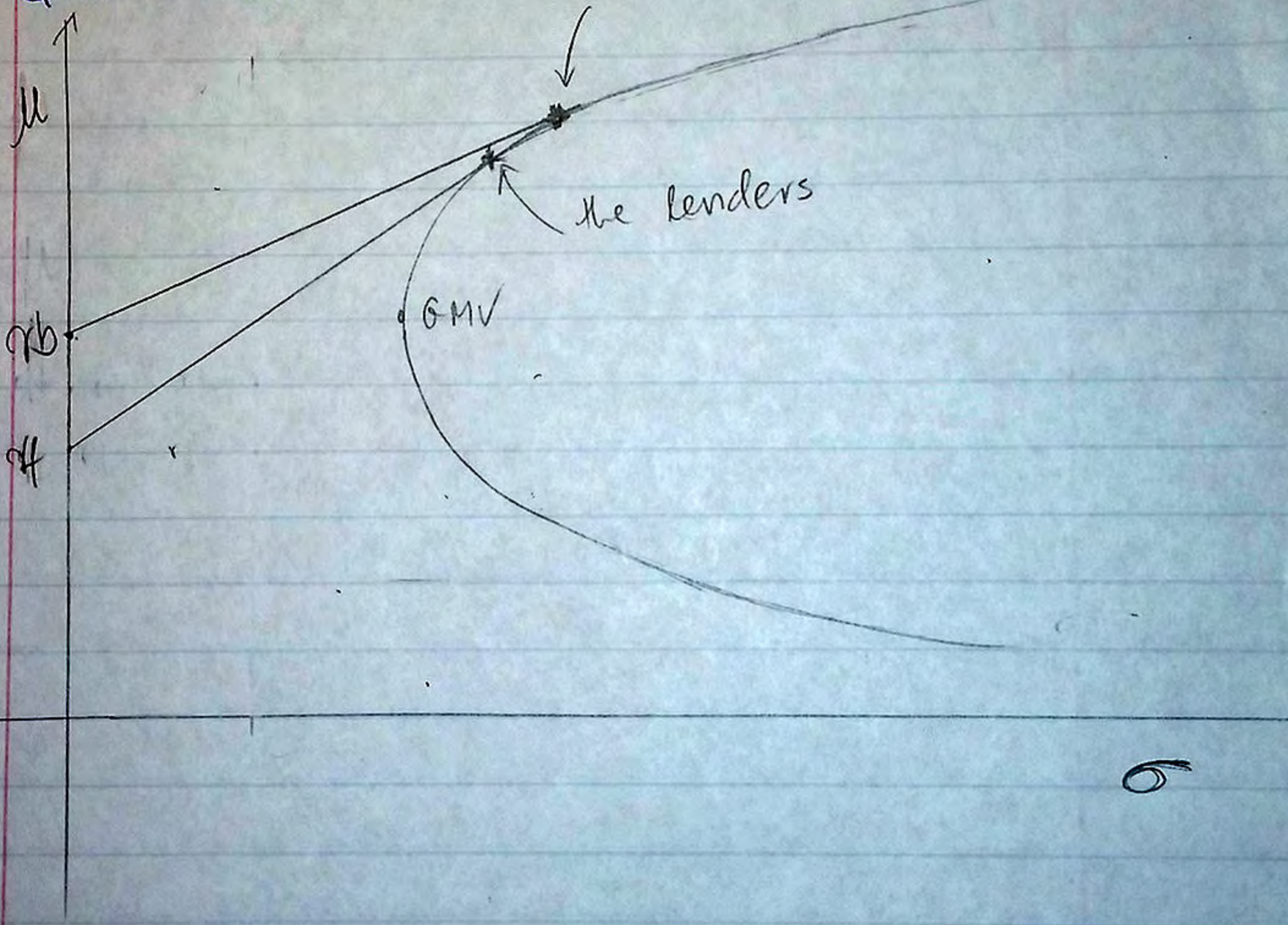
Question 2.



Question 3

#	#Shares	Price	Cap	W's
A	100	1.50	150	1/3
B	150	2.00	300	2/3
		total:	450	

$\rho = \frac{1}{3}$ $N = 2$ $V = [150, 300]$ $w = (\frac{1}{3}, \frac{2}{3})$

a) $\mu_m = w_A(\mu_A) + w_B \mu_B$

$\mu_m = (\frac{1}{3})(\frac{15}{100}) + (\frac{2}{3})(\frac{12}{100}) = \frac{5}{100} + \frac{8}{100} = 0.13$

b) $\mu_m = 0.13$

$\sigma^2_m = w^T \Sigma w$

$\begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix} \begin{bmatrix} 0.15^2 & (\frac{1}{3})(0.15)(0.09) \\ (\frac{1}{3})(0.15)(0.09) & 0.09^2 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \end{bmatrix} =$

$\begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix} \begin{bmatrix} 0.0225 & 0.0045 \\ 0.0045 & 0.0081 \end{bmatrix} \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \end{bmatrix} = 0.0080999838$

$\sigma_m = \sqrt{\sigma^2_m} = 0.090$

c) $\beta_A = \frac{\sigma_{AM}}{\sigma^2_M}$

$\sigma_{AM} = Cov(w_A A + w_B B, A) =$

$w_A Var(A) + w_B Cov(B, A) =$

$\frac{1}{3}(0.15)^2 + (\frac{2}{3})(\frac{1}{3})(0.15)(0.09) =$

$0.0075 + 0.003 = 0.0105$

$\beta_A =$

$$\beta_A = \frac{0.0105}{0.0081} = 1.296$$

d) $\mu_A = rf + \beta_A (\mu_m - rf)$

$$0.15 = rf + 1.296 (0.13 - rf)$$
$$0.15 = rf + 0.1685 - 1.296\, rf$$
$$0.296\, rf = 0.0185$$
$$rf = \frac{0.0185}{0.296}$$

$$rf = 0.063$$

The risk free rate is 0.063

The expected rate of return of the market portfolio is 0.13

The std. deviation of the market portfolio is 0.09

β of stock A is 1.296

Question 4, part a

```
getSig <- function(X){
 m <- nrow(X)
 k <- ncol(X)
 h <- rep(1,m)
 # column average (add up and devide by m)
 mu <- (h%*%X)/m
 MatMu <- matrix(rep(mu, m), nrow = m, byrow =T)
 SS <- X - MatMu
 sigma2 <- t(SS)%*%SS
 sigma <- (1/(m-1))*sigma2

 return (sigma)
}

## test getSig()
i <- 4
j <- 10
mat <- matrix(round(runif(j*i)*10, 0), nrow =j)
mat

result1 <- getSig(mat)
result1

##verify the results
var(mat)
```

OUTPUT:

```
> ## Question 4, part a
>
> getSig <- function(X){
+   m <- nrow(X)
+   k <- ncol(X)
+   h <- rep(1,m)
+   # column average (add up and devide by m)
+   mu <- (h%*%X)/m
+   MatMu <- matrix(rep(mu, m), nrow = m, byrow =T)
+   SS <- X - MatMu
+   sigma2 <- t(SS)%*%SS
+   sigma <- (1/(m-1))*sigma2
+
+   return (sigma)
+ }
>
> ## test getSig()
> i <- 4
> j <- 10
> mat <- matrix(round(runif(j*i)*10, 0), nrow =j)
> mat
     [,1] [,2] [,3] [,4]
 [1,]    9    3    7    9
 [2,]    8    9   10    8
 [3,]    5   10    9    7
 [4,]    4    8    5    7
 [5,]    8    3    3    3
 [6,]    0    9    3    9
 [7,]    6    2    6    1
 [8,]    1    2    6    5
 [9,]    5    5    1    7
[10,]    4    7    8    3
>
> result1 <- getSig(mat)
> result1
         [,1]      [,2]     [,3]      [,4]
[1,]  8.6666667 -2.000000 2.333333 -0.5555556
[2,] -2.0000000  9.955556 2.844444  4.5333333
[3,]  2.3333333  2.844444 8.177778  0.2000000
[4,] -0.5555556  4.533333 0.200000  7.6555556
>
> ##verify the results
> var(mat)
         [,1]      [,2]     [,3]      [,4]
[1,]  8.6666667 -2.000000 2.333333 -0.5555556
[2,] -2.0000000  9.955556 2.844444  4.5333333
[3,]  2.3333333  2.844444 8.177778  0.2000000
[4,] -0.5555556  4.533333 0.200000  7.6555556
```

```
>
```

Question 4. Part b.

```
getCoef <- function(x, y){
A <- matrix(c(rep(1,length(x)), x), nrow =length(x), byrow = F)
A
return(solve(t(A)%*%A, t(A)%*%y))
}
## test getCoef()
vecY<-c(1,2,3,4,5,6,7)
vecX<-c(2,3,6,8,5,4,3)
solution  <- getCoef(vecX, vecY)
solution
##verify
lsfit(vecX,vecY, intercept = T)
```

OUTPUT

```
> ## Question 4. Part b.
>
> getCoef <- function(x, y){
+ A <- matrix(c(rep(1,length(x)), x), nrow =length(x), byrow = F)
+ A
+ return(solve(t(A)%*%A, t(A)%*%y))
+ }
>
> ## test getCoef()
> vecY<-c(1,2,3,4,5,6,7)
> vecX<-c(2,3,6,8,5,4,3)
> solution  <- getCoef(vecX, vecY)
> solution
        [,1]
[1,] 3.3111111
[2,] 0.1555556
> ##verify
> lsfit(vecX,vecY, intercept = T)
$coefficients
Intercept         X
3.3111111 0.1555556

$residuals
[1] -2.6222222 -1.7777778 -1.2444444 -0.5555556  0.9111111  2.0666667  3.2222222
```

```
$intercept
[1] TRUE

$qr
$qt
[1] -10.5830052   0.7888106  -0.9312197  -0.6052515   1.4057962   2.7428121   4.0798280

$qr
     Intercept           X
[1,] -2.6457513 -11.71689866
[2,]  0.3779645   5.07092553
[3,]  0.3779645  -0.44125397
[4,]  0.3779645  -0.83565929
[5,]  0.3779645  -0.24405131
[6,]  0.3779645  -0.04684865
[7,]  0.3779645   0.15035401

$qraux
[1] 1.377964 1.150354

$rank
[1] 2

$pivot
[1] 1 2

$tol
[1] 1e-07

attr(,"class")
[1] "qr"
```

```
>
```