C183 - Project 4

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

stock <- read.csv("stockData.csv", sep=",", header=TRUE)
returns <- (stock[-1,3:ncol(stock)]-stock[-nrow(stock),3:ncol(stock)])/stock[-nrow(stock),3:ncol(stock)]

stock_names <- colnames(returns)
stock_names <- stock_names[2:length(stock_names)]


data <- data.frame()
n <- nrow(returns)


for(i in stock_names){

t <- paste(i, 'X.GSPC', sep = ' ~ ')
model <- lm(t, data = returns)
co <- as.numeric(model$coefficients)
var <- sum(model$residuals^2) / (n-2)
data <- rbind(data, c(co, var))

}


colnames(data) <- c('alpha', 'beta', 'sigma')
sigma <- var(returns$X.GSPC)
rownames(data) <- stock_names

data %>% arrange(desc(beta))
```

```
##               alpha      beta       sigma
## TSLA   0.0274545269 1.8364717 0.026771258
## NVDA   0.0371238778 1.7397344 0.012389815
## C     -0.0064623005 1.5614556 0.004367604
## MU     0.0027510870 1.4150686 0.010227956
## GS    -0.0001222496 1.3738849 0.003146655
## DIS   -0.0057560070 1.2920864 0.003655394
## NFLX   0.0146730658 1.2890893 0.012132061
## AAPL   0.0115784779 1.2743952 0.003665194
## CRM    0.0062143302 1.2741884 0.005108393
## LULU   0.0133726178 1.1580621 0.008922870
## AXP    0.0018524266 1.1446860 0.003117340
## GOOGL  0.0068899061 1.1203555 0.002450166
## TSM    0.0115468177 1.1173894 0.005935593
## META   0.0061839656 1.1129029 0.007428385
## JPM    0.0046053430 1.1004374 0.002675371
## MA     0.0089344267 1.0967111 0.002262860
## NKE    0.0049783369 1.0205110 0.003271393
## MSFT   0.0152866567 0.9895580 0.002016417
## V      0.0071181029 0.9748402 0.001687497
## BABA   0.0004362885 0.9345299 0.012735118
## SBUX   0.0053401422 0.8685494 0.002910960
## TMO    0.0104584288 0.8682832 0.002207838
## BIDU  -0.0029954602 0.8371060 0.015080668
## UNH    0.0124201740 0.6944230 0.002514650
## CVS   -0.0037975366 0.6527675 0.004184400
## MCD    0.0093128349 0.6509412 0.001684795
## GILD  -0.0020627475 0.5657933 0.004508701
## BMY    0.0017961195 0.5226090 0.004265087
## NVO    0.0123936223 0.5223988 0.003599509
## ATVI   0.0143934518 0.4937844 0.006386527
```

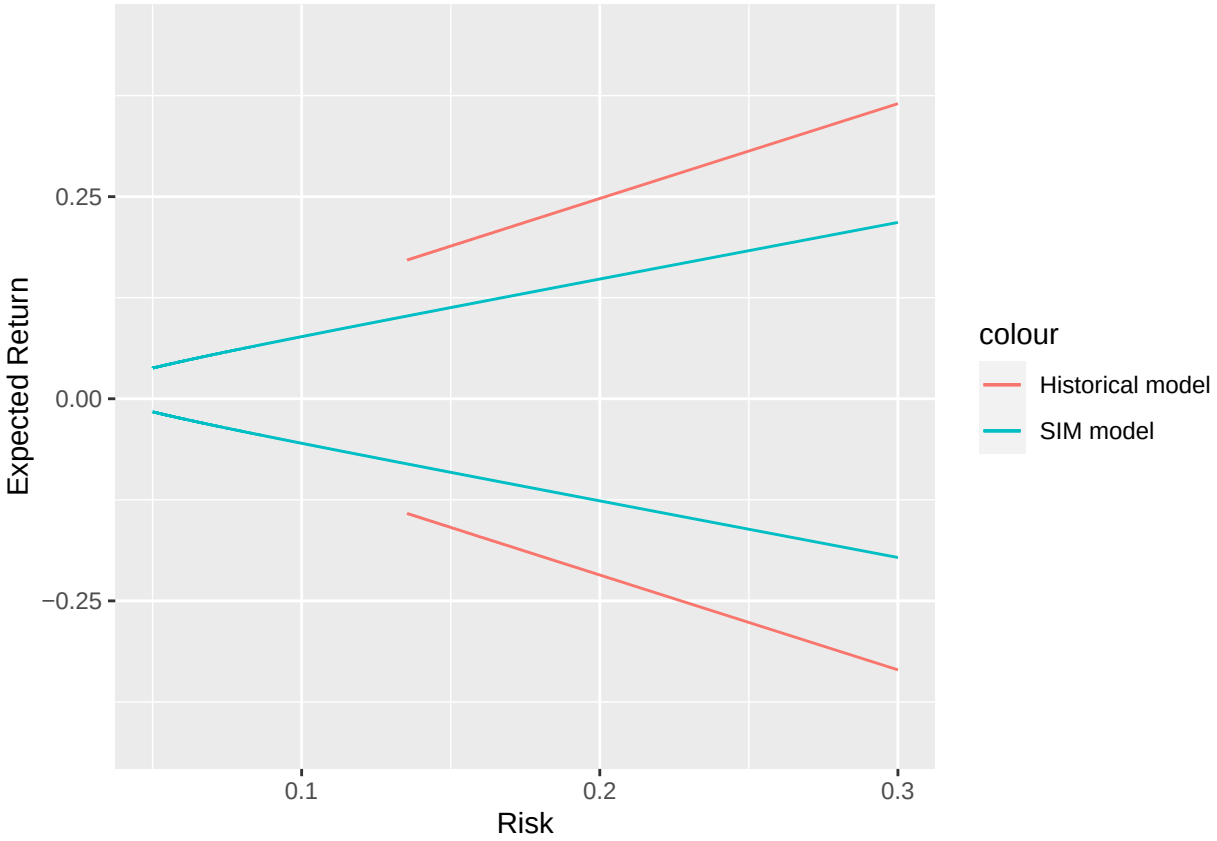
```
m1 <- diag(data$sigma)
b <- as.matrix(data$beta)


m2 <- sigma * (b %*% t(b))
total <- m1 + m2

colnames(total) <- stock_names
rownames(total) <- stock_names
```

```
stock2 <- read.csv("stockData.csv", sep=",", header=TRUE)[1:60,]
data2 <- (stock2[-1,4:ncol(stock2)]-stock2[-nrow(stock2),4:ncol(stock2)])/stock2[-nrow(stock2),4:ncol(s

r <- as.matrix(colMeans(data2))
sigma_mat <- cov(data2)


i_m <- matrix(rep(1,30), 30, 1)
A <- as.numeric(t(r) %*% solve(sigma_mat) %*% i_m)
B <- as.numeric(t(r) %*% solve(sigma_mat) %*% r)
C <- as.numeric(t(i_m) %*% solve(sigma_mat) %*% i_m)
D <- B*C - A^2
```

```r
sigmas <- exp(seq(-2, -1, 0.0001))
sigmas_S <- exp(seq(-3, -1, 0.0001))
e1 <- A/C + sqrt(D * (C * sigmas^2 - 1)) / C
e2 <- A/C - sqrt(D * (C * sigmas^2 - 1)) / C

# SIM method
A_S <- as.numeric(t(r) %*% solve(total) %*% i_m)
B_S <- as.numeric(t(r) %*% solve(total) %*% r)
C_S <- as.numeric(t(i_m) %*% solve(total) %*% i_m)
D_S <- B_S*C_S - A_S^2

e1_S <- A_S/C_S + sqrt(D_S * (C_S * sigmas_S^2 - 1)) / C_S
e2_S <- A_S/C_S - sqrt(D_S * (C_S * sigmas_S^2 - 1)) / C_S


ER <- ggplot() + geom_line(aes(x = sigmas, y = e1, color = 'Historical model')) +

geom_line(aes(x = sigmas, y = e2, color = 'Historical model')) +
geom_line(aes(x = sigmas_S, y = e1_S, color = 'SIM model')) +
geom_line(aes(x = sigmas_S, y = e2_S, color = 'SIM model')) +
xlab('Risk') +
ylab('Expected Return') +
xlim(0.05, 0.3)
ER
```

```
## Warning: Removed 2040 row(s) containing missing values (geom_path).
## Removed 2040 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 2083 row(s) containing missing values (geom_path).
## Removed 2083 row(s) containing missing values (geom_path).
```


0.25
0.00
−0.25
Expected Return
0.1
0.2
0.3
Risk
colour
Historical model
SIM model