Markowitz-model

Jesse Keränen

1/17/2022

Prologue

This file is little bit more theoretical than previous ones. In this file I examine portfolio optimization in the spirit of Harry Markowitz. File starts by deriving measures for portfolio return and risk. Then optimal set of portfolios is derived. These derivations are not my own work, rather I replicate derivations from my asset management course: By doing this I try to gain deeper understanding of the subject. Thoughout the file I also perform calculations with real world data and compare them to the results from PortfolioOptimization-file. Results of this file also work as a base for calculations in the Cap-model-file.

Modern portfolio theory

To be able to derive cap-model we need first take a look a bit further, more precisely to 50's. Capital asset pricing model is William Sharpe's work, but it is largely based on Harry Markowitz's portfolio theory. Markowitz examined portfolios on a risk return horizon. He used mean historical returns as a proxy for returns and standard deviation of the returns as a proxy for risk.

$$E(r_p) = \frac{1}{n-1} \sum w_i E(r_i)$$

For single stock variance can be calculated with following formula:

$$\sigma^2 = \frac{1}{n-1} \sum_{t=1}^{n} (r_t - \mu)^2$$

```
library(tidyverse)
library(ggplot2)
library(data.table)
library(Quandl)
library(plotly)

Quandl.api_key("bx1qdehfWXg6SNKnicQC")

names <- c("AAPL", "MSFT", "TSLA", "GOOGL", "AMZN")

# For monthly data use collapse = "monthly"
prices <- as.data.table(Quandl(c("WIKI/AAPL.11","WIKI/MSFT.11",
        "WIKI/TSLA.11", "WIKI/GOOGL.11", "WIKI/AMZN.11"), start_date
        = "2010-06-01", end_date = "2015-03-01", collapse = "monthly"))

colnames(prices) <- c("Date", "AAPL", "MSFT", "TSLA", "GOOGL", "AMZN")
```

```r
# Tidying data table
prices <- melt(prices, id.vars = "Date", measure.vars = names, variable.name
               = "Company", value.name = "Adj_Close")

# Calculate simple price changes
prices[, "Returns" := Adj_Close/shift(Adj_Close) -1, by = Company]

prices <- drop_na(prices)

mean_std <- prices[, .(means = mean(Returns), stds = sd(Returns)), by= Company]

mean_std
```

```
##    Company      means       stds
## 1:    AAPL 0.02661200 0.07238047
## 2:    MSFT 0.01557518 0.05838445
## 3:    TSLA 0.05376021 0.18982291
## 4:   GOOGL 0.01910563 0.07189913
## 5:    AMZN 0.02539402 0.07781302
```

When talking about portfolio variance it is not sufficient just take into consideration variance of the individual stocks. There tends to be factors that affect whole markets, which can cause co-movements in stock returns. This co-movement is called covariance. In the code below I have illustrated correlation between Nokia's returns and Omxh25-index movements. Covariance can be calculated with following formula:

$$\sigma_{12} = \frac{1}{n-1} \sum_{t=1}^{n} (r_{1t} - \mu_1)(r_{2t} - \mu_2)$$

```r
cov(prices[Company == "AAPL", Returns], prices[Company == "MSFT", Returns])
```

```
## [1] 0.001380774
```

```r
cor(prices[Company == "AAPL", Returns], prices[Company == "MSFT", Returns])
```

```
## [1] 0.3267413
```

In two asset case we can easily derive portfolio variance formula.

$$E(r_p) = E[xr_1 + (1-x)r_2]$$

$$\sigma_p^2 = \frac{1}{n-1} \sum_{s=1}^{S} (r_s - \mu)^2$$

What is said above is that expected return of the portfolio is expected return of the first asset times weight of the first asset plus expected return of the second asset times one minus weight of the first asset. Second formula states that variance of the portfolio return is the sum of squared differences between realized returns and mean returns. If we plug the first formula into second we get:

$$\sigma_p^2 = \frac{1}{n-1} [(xr_{t1} - x\mu_1) + ((1-x)r_{2t} - (1-x)\mu_2))]^2$$

We know that to get rid of the exponent we can write our formula as:

$$\sigma_p^2 = \frac{1}{n-1}[x^2(r_{1t}-\mu_1)^2 + (1-x)^2(r_{2t}-\mu_2)^2 + 2x(1-x)(r_{1t}-\mu_1)(r_{2t}-\mu_2)]$$

Now if we remember that $\frac{1}{n-1}\sum_{t=1}^{n}(r_t-\mu)^2$ is variance of stock and $\frac{1}{n-1}\sum_{t=1}^{n}(r_{1t}-\mu_1)(r_{2t}-\mu 2)$ is the covariance between two stocks, we can further write our formula as:

$$\sigma_p^2 = x^2\sigma_1^2 + (1-x)^2\sigma_2^2 + 2x(1-x)\sigma_{12}$$

Other thing worth mentioning is that correlation is just standardized covariance and can be calculated by dividing covariance by the product of the standard deviations of the two stocks. Above formula could also be written with correlation instead of covariance. Similar formulas can be derived for portfolios with more than two assets, but many times in real world it is more convenient to use matrix notations. More general version of the portfolio variance formula is:

$$\sigma_p^2 = \sum_{i=1}^{N} x_i^2\sigma_i^2 + \sum_{i=1}\sum_{j\neq 1} x_i x_j \sigma_{ij}$$

Matrix notation:

$$\sigma_p^2 = x^t \cdot \underline{\Omega} \cdot x$$

Where $\underline{\Omega}$ is covariance matrix of stocks in portfolio.

```
cov_matrix <- cov(as.data.table(split(prices[, Returns] ,prices$Company)))
cov_matrix
```

```
##               AAPL         MSFT          TSLA          GOOGL          AMZN
## AAPL  0.0052389324 0.0013807740  0.0002069998  0.0017124912 0.0018714985
## MSFT  0.0013807740 0.0034087441  0.0003497711  0.0014111502 0.0004279719
## TSLA  0.0002069998 0.0003497711  0.0360327358 -0.0000873319 0.0005108136
## GOOGL 0.0017124912 0.0014111502 -0.0000873319  0.0051694856 0.0015437988
## AMZN  0.0018714985 0.0004279719  0.0005108136  0.0015437988 0.0060548656
```

Now that we know how to measure portfolio properties we can take a look at another interesting concept of Markovitz's, efficient frontier. Efficient frontier consists of portfolios that with given level of risk produce highest possible return. In the other words tells the optimal portfolios for investor. Well, how can investor find this optimal set of portfolios. Here optimization, which is discussed in more detail in PortfolioOptimization-file, comes handy.

```
# Efficient frontier constructed here, short selling is not possible
n <- 15000

port_return <- vector('numeric', length = n)
port_std <- vector('numeric', length = n)
port_sharpe <- vector('numeric', length = n)

for (i in 1:n){
  rand_weights <- runif(unique(prices$Company))
  rand_weights <- rand_weights/sum(rand_weights)

  port_return[i] <- mean_std$means %*% rand_weights
  port_std[i] <- sqrt(rand_weights %*% cov_matrix %*% matrix(rand_weights))
  port_sharpe[i] <- port_return[i]/port_std[i]
```

```
}
dt <- data.table(port_return, port_std, port_sharpe)

fig <-ggplot(dt, aes(dt$port_std, dt$port_return, color=dt$port_sharpe)) +
  geom_point() + labs(x = "Standard Deviation", y = "Monthly Return",
                      color = "Sharpe Ratio")

# ggplotly(fig)
```

Since efficient frontier is just parabolic function of risk and return, it can also be presented with following formula:

$$\sigma_p^2 = \sigma_{min}^2 + \beta(\mu_p - \mu_{min})^2$$

Where σ_{min}^2 and μ_{min} are the variance and expected return of the minimum variance portfolio. Coefficient β defines the "openness" of our parabola and is defined by characteristics of stocks in our portfolio. Since β is constant, we can solve it by first finding minimum variance portfolio and then finding at least one other portfolio on the efficient frontier and its variance and expected return. We also know that return of the portfolio is weighted sum of returns of underlying stocks. This means that return of the portfolio can't exceed return of the stock with highest expected return. So portfolio with highest amount of return consist of only one stock and then we can use expected return and variance of that stock in our above formula to get β. This function only tells us risks and returns of efficient frontier, but it doesn't tells us which weights produce these risk/return combinations.

We could solve weights for minimum variance portfolio by using computational power as is done in PortfolioOptimization-file, but we can also solve it by using Lagrangian method. Our objective function would be formula of portfolio variance:

$$min\ \sigma_p^2 = \overline{x}^t \cdot \underline{\Omega} \cdot \overline{x}$$

We also have constraint that weights need to sum up to one:

$$s.t.\ \overline{x}^t \cdot \overline{1} = 1$$

Then our Lagrangian function would be:

$$L = \overline{x}^t \cdot \underline{\Omega} \cdot \overline{x} - \lambda(\overline{x}^t \cdot \overline{1} - 1)$$

Next we take partial derivative of our Lagrangian function with respect to x and set it to 0:

$$\frac{\partial L}{\partial \overline{x}} = 2\underline{\Omega}\overline{x} - \lambda\overline{1} = 0$$

We are interested in x and we also know that matrix multiplied by its inverse matrix is identity matrix. If we first move $-\lambda\overline{1}$ to other side of the equation and the multiply both sides by $\frac{1}{2}$ and the inverse matrix of our covariance matrix we get:

$$\overline{x} = \frac{\lambda}{2}\underline{\Omega}^{-1}\overline{1}$$

There is still one unknown we need to solve before we get our optimal weights, lambda. In our original problem we have restricted that $\overline{x}^t \cdot \overline{1}$ needs to be one. If we multiply both sides of our equation by vector of ones we get:

$$1 = \frac{\lambda}{2}\overline{1}^t\underline{\Omega}^{-1}\overline{1}$$

Which can also be written as:

$$\frac{\lambda}{2} = \frac{1}{\overline{1}^t\underline{\Omega}^{-1}\overline{1}}$$

And when we plug this back to our previous function we get that:

$$\overline{x} = \frac{1}{\overline{1}^t\underline{\Omega}^{-1}\overline{1}}\underline{\Omega}^{-1}\overline{1}$$

```r
ones <- matrix(rep(1, length(names)))
inv_cov_matrix <- solve(cov_matrix)

x = as.numeric(1/(t(ones) %*% inv_cov_matrix %*% ones)) * (inv_cov_matrix %*% ones)

# These are basically same values that we got in PortfolioOptimization-file
x
```

```
##             [,1]
## AAPL  0.13865260
## MSFT  0.43687927
## TSLA  0.04831945
## GOOGL 0.16201185
## AMZN  0.21413684
```

This function gives us the weights that form minimum variance portfolio. By plugging these weights into our above derived formulas we get the variance and expected returns of minimum variance portfolio. Now we have found minimum variance portfolio, but already looking at the efficient frontier graph we can see that minimum variance portfolio probably doesn't give us best possible ratio of risk and return. What about if we wanted to bear more risk in exchange for more return? We can calculate optimal distribution of capital between stocks in our portfolio, with given required return level by adding another constraint to our optimization formula:

$$s.t.\ \overline{x}^t\ \overline{\mu} = \mu_p$$

Where $\overline{\mu}$ is vector of expected returns and μ_p is required return level of the portfolio. Now our Lagrangian formula gets form of:

$$L = \overline{x}^t\ \underline{\Omega}\ \overline{x} - \lambda(\overline{x}^t\ \overline{1} - 1) - \gamma(\overline{x}^t\ \overline{\mu} - \mu_p)$$

Now we again take the partial derivative with respect to the x:

$$\frac{\partial L}{\partial \overline{x}} = 2\overline{x}\underline{\Omega} - \lambda\overline{1} - \gamma\overline{\mu} = 0$$

Again we want to have only x on the other side of the equation. We first move two latter terms to the right side of the equation, then divide both sides by two and lastly we multiply both sides by the inverse of covariance matrix.

$$\overline{x} = (\frac{\lambda}{2}\overline{1} + \frac{\gamma}{2}\mu)\underline{\Omega}^{-1}$$

In addition to weights, this time we have two unknown variables, λ and γ. Because of our constraint we can temporarily get rid of weights by multiplying our equation by vector of ones and vector of experceted return. We get two equations:

$$\frac{\lambda}{2}\ \overline{1}^t \underline{\Omega}^{-1} \overline{1} + \frac{\gamma}{2}\ \overline{1}^t \underline{\Omega}^{-1} \overline{\mu} = 1$$

and

$$\frac{\lambda}{2}\ \overline{\mu}^t \underline{\Omega}^{-1} \overline{1} + \frac{\gamma}{2}\ \overline{\mu}^t \underline{\Omega}^{-1} \overline{\mu} = \mu_p$$

We can solve λ and γ already from this system of equations, but to save same space let's call $\overline{1}^t \underline{\Omega}^{-1} \overline{1}$ c, $\overline{1}^t \underline{\Omega}^{-1} \overline{\mu}$ a and $\overline{\mu}^t \underline{\Omega}^{-1} \overline{\mu}$ b. If we write above equations with these letters we get:

$$\frac{\lambda}{2} c + \frac{\gamma}{2} a = 1$$

and

$$\frac{\lambda}{2} a + \frac{\gamma}{2} b = \mu_p$$

```
c <- t(ones) %*% inv_cov_matrix %*% ones
a <- t(ones) %*% inv_cov_matrix %*% mean_std[, means]
b <- t(mean_std[, means]) %*% inv_cov_matrix %*% mean_std[, means]
```

Lets solve first γ:

$$\frac{\lambda}{2} = \frac{1 - \frac{\gamma}{2} a}{c}$$

$$\frac{1 - \frac{\gamma}{2} a}{c} a + \frac{\gamma}{2} b = \mu_p$$

$$a - \frac{\gamma}{2} a^2 + \frac{\gamma}{2} bc = \mu_p c$$

$$\frac{\gamma}{2}(a^2 + bc) = \mu_p c - a$$

$$\frac{\gamma}{2} = \frac{\mu_p c - a}{bc - a^2}$$

and then λ:

$$\frac{\lambda}{2} a + \frac{\mu_p c - a}{bc - a^2} b = \mu_p$$

$$\frac{\lambda}{2} a = \frac{\mu_p (bc - a^2) - b(\mu_p c - a)}{bc - a^2}$$

$$\frac{\lambda}{2} a = \frac{\mu_p bc - \mu_p a^2 - b \mu_p c + ab}{bc - a^2}$$

$$\frac{\lambda}{2} = \frac{a(b - \mu_p a)}{a(bc - a^2)}$$

$$\frac{\lambda}{2} = \frac{b - \mu_p a}{bc - a^2}$$

Now if plug these values back to our original function, we can calculate optimal weights with given return level.

$$\overline{x} = \frac{b - \mu_p a}{bc - a^2}\ \underline{\Omega}^{-1}\ \overline{1} + \frac{\mu_p c - a}{bc - a^2}\ \underline{\Omega}^{-1}\ \overline{\mu}$$

```r
# Example with return level of 0.03%
mu_p <- 0.03

x2 <- as.numeric((b-mu_p*a)/(b*c-a^2)) * inv_cov_matrix %*% ones +
  as.numeric((mu_p*c-a)/(b*c-a^2)) * inv_cov_matrix %*% mean_std[, means]
x2
```

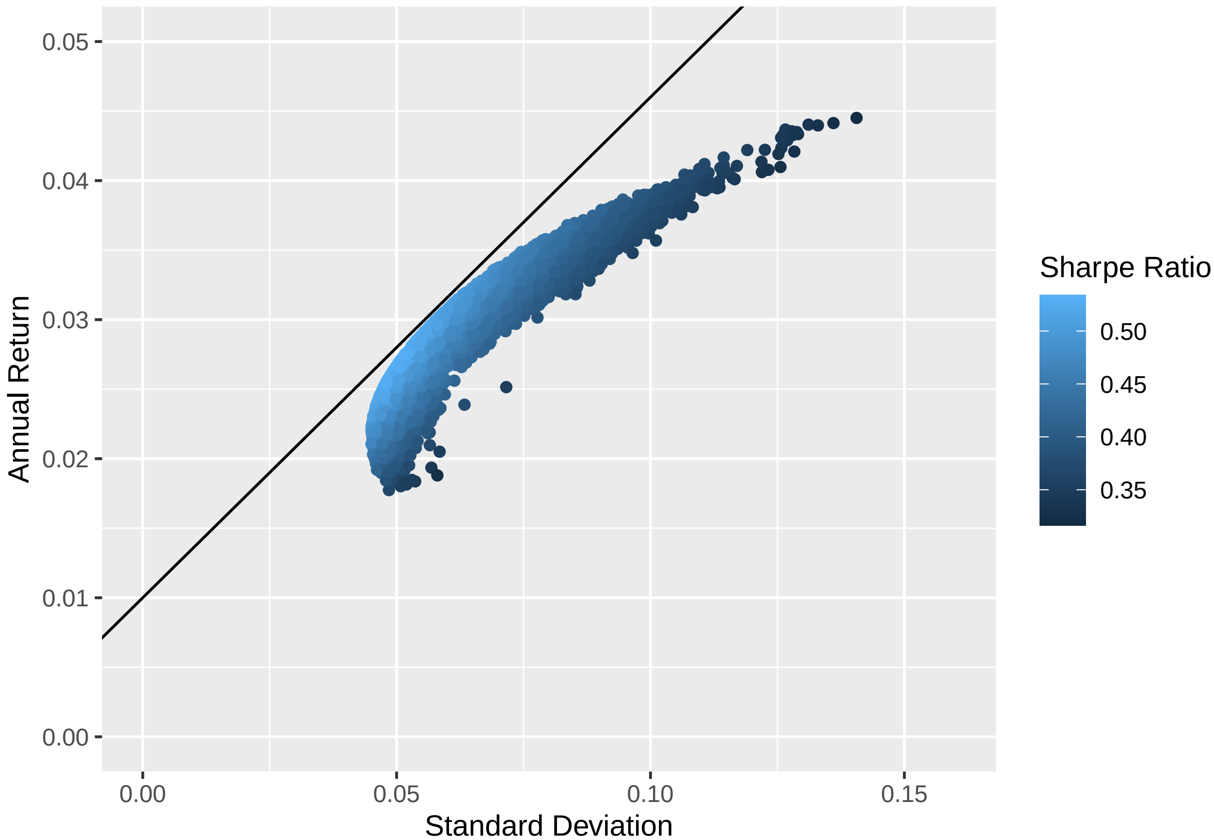
```
##             [,1]
## AAPL  0.38351737
## MSFT  0.06716227
## TSLA  0.18945957
## GOOGL 0.09158565
## AMZN  0.26827514
```

Separation theorem

One thing Markowitz didn't consider was risk free returns. Having risk free returns isn't too unrealistic assumption, since many investors consider government bonds as more or less risk free returns. James Tobin (1958) expanded Markowitz's work with riskless assets. Bringing risk free assets along crucially changes optimal behavior of the investors. Line that tangents our "efficient frontier" in plot below represents now optimal combinations of risk free asset and risky portfolio. In every risk level, except tangent portfolio, this new line offers better returns than our old efficient frontier.

```r
fig2 <-ggplot(dt, aes(dt$port_std, dt$port_return, color=dt$port_sharpe)) +
  geom_point() + labs(x = "Standard Deviation", y = "Annual Return",
                color = "Sharpe Ratio") + geom_abline(slope = 0.36,
                intercept = 0.01) + xlim(c(0,0.16)) + ylim(c(0, 0.05))

fig2
```



Variance of risk free asset is zero. That means also that covariance between any risky asset and risk free asset is zero. If we combine risky portfolio and risk free asset, the variance of this new portfolio is squared weight of the risky portfolio times variance of risky portfolio:

$$\sigma_p^2 = w_{rsk}^2 \sigma_{rsk}^2$$

From this equation we can solve optimal amount of risky portfolio w_{rsk} with given required risk level:

$$w_{rsk} = \frac{\sigma_p}{\sigma_{rsk}}$$

Portfolio return can be calculated as weighted sum of risk free return and expected return of the risky portfolio:

$$\mu_p = x_{rsk}\mu_{rsk} + (1 - x_{rsk})r_f$$

If we plug w_{rsk} to above equation we get:

$$\mu_p = \frac{\sigma_p}{\sigma_{rsk}}\mu_{rsk} + (1 - \frac{\sigma_p}{\sigma_{rsk}})r_f$$
$$\mu_p = r_f + \frac{\sigma_p}{\sigma_{rsk}}(\mu_{rsk} - r_f)$$

This would be convenient if we only had one risky asset. Let's next see how we can also determine the optimal distribution of the capital within the risky portfolio. Now our optimization formula goes as following:

$$min\ \sigma_p^2 = \overline{x}^t\ \underline{\Omega}\ \overline{x}$$
$$s.t.\ \overline{x}^t\ \overline{1} + B = 1$$
$$\mu_p = \overline{x}^t\ \overline{\mu} + Br_f$$

Where B is the amount of capital invested in risk free asset. Together $\overline{x}$ and B must sum up to 1. Note that B can take positive and negative values, since according to model investor can lend and borrow money at risk free rate. To make our calculations easier we can combine our constraints as one:

$$\overline{x}^t\overline{1}r_f + Br_f = r_f$$
$$\overline{x}^t\overline{1}r_f + Br_f - r_f = \overline{x}^t\overline{\mu} + Br_f - \mu_p$$
$$\overline{x}^t(\overline{\mu} - \overline{1}r_f) = \mu_p - r_f$$
$$\overline{x}^t\overline{\pi} = \mu_p - r_f$$

Where π is the risk premiums of expected returns. Then we can write our Lagrangian function and take partial derivative with respect to $\overline{x}$ from it. After that we multiply both sides of our equation by inverse of covariance matrix to get only $\overline{x}$ to the other side of the equation:

$$L = \overline{x}^t\underline{\Omega}\overline{x} - \lambda(\overline{x}^t\overline{\pi} - \mu_p + r_f)$$
$$\frac{\partial L}{\partial \overline{x}} = 2\overline{x}^t\underline{\Omega} - \lambda\overline{\pi} = 0$$
$$\overline{x} = \frac{\lambda}{2}\underline{\Omega}^{-1}\overline{\pi}$$

If we now plug our $\overline{x}$ equation into our constraint equation we get:

$$\frac{\lambda}{2}\overline{\pi}^t\underline{\Omega}^{-1}\overline{\pi} = \mu_p - r_f$$
$$\frac{\lambda}{2} = \frac{\mu_p - r_f}{\overline{\pi}^t\underline{\Omega}^{-1}\overline{\pi}}$$

And finally we can solve our $\overline{x}$:

$$\overline{x} = \frac{\mu_p - r_f}{\overline{\pi}^t\underline{\Omega}^{-1}\overline{\pi}}\ \underline{\Omega}^{-1}\overline{\pi}$$

Let's next see with our real data what would be our optimal portfolio if we had risk free asset with return of 1%. Let's also define that we want our portfolio's monthly return to be 2.5%.

```
r_f <- 0.01
pi <- mean_std$means - (ones * r_f)
mu_p2 <- 0.025

x3 <- as.numeric((mu_p2-r_f)/(t(pi) %*% inv_cov_matrix %*% pi)) * inv_cov_matrix %*% pi
x3
```

```
##              [,1]
## AAPL  0.29408351
## MSFT  0.01994192
## TSLA  0.14682196
## GOOGL 0.06036916
## AMZN  0.19675474
```

```r
# In this case we invest almost 30% to risk free asset
1-sum(x3)
```

```
## [1] 0.2820287
```

```r
# Let's also calculate standard deviation of this portfolio
sd <- sqrt(((mu_p2-r_f)^2)/(t(pi) %*% inv_cov_matrix %*% pi))

# Seems consistent with the plot we have approximated above
sd
```

```
##             [,1]
## [1,] 0.04323573
```

```r
mu_p2 <- 0.05
x4 <- as.numeric((mu_p2-r_f)/(t(pi) %*% inv_cov_matrix %*% pi)) * inv_cov_matrix %*% pi
x4
```

```
##             [,1]
## AAPL  0.78422270
## MSFT  0.05317845
## TSLA  0.39152522
## GOOGL 0.16098442
## AMZN  0.52467930
```

```r
# Diversification within risky portfolio stays, same regardless of wanted return level
x3/sum(x3)
```

```
##             [,1]
## AAPL  0.40960345
## MSFT  0.02777537
## TSLA  0.20449559
## GOOGL 0.08408297
## AMZN  0.27404263
```

```r
x4/sum(x4)
```

```
##             [,1]
## AAPL  0.40960345
## MSFT  0.02777537
## TSLA  0.20449559
## GOOGL 0.08408297
## AMZN  0.27404263
```

Epilogue

Many times solvers and algorithms provided by many softwares can solve this kind of optimization problems really fast. So usually there isn't really need to calculate these calculations by hand, but on the other hand this kind of manual calculation helps me to understand why returns and risks of a portfolio behave such a way that they do. Understanding these calculations also helps us to understand derivation of the capital asset model.