HW6

Huanyu Liu

3/10/2019

1

Sample Mean

```
#Sample Mean
Avg.Mkt = 0.05
Avg.1 = 0.01 + 0.9*Avg.Mkt
Avg.2 = -0.015 + 1.2*Avg.Mkt
Avg.3 = 0.005 + 1.0*Avg.Mkt
Avg = c(Avg.1,Avg.2,Avg.3)
Avg_df = data.frame(Avg.1,Avg.2,Avg.3)
colnames(Avg_df) = c("1st","2nd","3rd")
Avg_df
```

```
##     1st   2nd   3rd
## 1 0.055 0.045 0.055
```

Standard Deviation

```
#Standard Deviation
Variance_matrix = matrix(c(0.1^2,0,0,0,0.15^2,0,0,0,0.05^2),nrow=3)
betas = c(0.9,1.2,1)
Variance.Mkt = 0.15^2
Variance.1 = betas[1]^2 * Variance.Mkt +  Variance_matrix[1,1]
Variance.2 = betas[2]^2 * Variance.Mkt +  Variance_matrix[2,2]
Variance.3 = betas[3]^2 * Variance.Mkt +  Variance_matrix[3,3]
Sd = c(sqrt(Variance.1),sqrt(Variance.2),sqrt(Variance.3))
Sd_df = data.frame(sqrt(Variance.1),sqrt(Variance.2),sqrt(Variance.3))
colnames(Sd_df) = c("1st","2nd","3rd")
Sd_df
```

```
##        1st       2nd       3rd
## 1 0.168003 0.2343075 0.1581139
```

Sharpe Ratio

```
#SharpeRatio
Avg/Sd
```

```
## [1] 0.3273752 0.1920553 0.3478505
```

2

To hedge out the market risk, we go short the market based on the beta of the market provided in the regression equation.
Mean

```r
#Sample Mean
Avg.Hedged.1 = 0.01
Avg.Hedged.2 = -0.015
Avg.Hedged.3 = 0.005
Avg.Hedged = c(Avg.Hedged.1,Avg.Hedged.2,Avg.Hedged.3)
Avg.Hedged_df = data.frame(Avg.Hedged.1,Avg.Hedged.2,Avg.Hedged.3)
colnames(Avg.Hedged_df) = c("1st Hedged","2nd Hedged","3rd Hedged")
Avg.Hedged_df
```

```
##   1st Hedged 2nd Hedged 3rd Hedged
## 1       0.01     -0.015      0.005
```

Standard Deviation

```r
#Standard Deviation
Variance.Hedged.1 = Variance_matrix[1,1]
Variance.Hedged.2 = Variance_matrix[2,2]
Variance.Hedged.3 = Variance_matrix[3,3]
Sd.Hedged = c(sqrt(Variance.Hedged.1),sqrt(Variance.Hedged.2),sqrt(Variance.Hedged.3))
Sd.Hedged_df = data.frame(sqrt(Variance.Hedged.1),sqrt(Variance.Hedged.2),sqrt(Variance.Hedged.3))
colnames(Sd.Hedged_df) = c("1st Hedged","2nd Hedged","3rd Hedged")
Sd.Hedged_df
```

```
##   1st Hedged 2nd Hedged 3rd Hedged
## 1        0.1       0.15       0.05
```

Sharpe Ratio

```r
#SharpeRatio
Avg.Hedged/Sd.Hedged
```

```
## [1]  0.1 -0.1  0.1
```

3

The maximum sharpe ratio squared based on the mean variance efficiency $= (\bar{R}^e)'\Omega^{-1}\bar{R}^e$

Proof

The aim is to minimize portfolio variance ($w'\Omega w$, where w is the weights and Ω is variance-covariance matrix), such that the portfolio returns reach the necessary value of m

The objective function from the lagrangian form is

$min\ \frac{1}{2}w'\Omega w - k(w'\bar{R}^e - m)$

First order differential w.r.t w and set it to 0 to minimize

$\Omega w - k\bar{R}^e = 0$

so $w^{MVE} = k\Omega^{-1}\bar{R}^e$

so, $\bar{R}^e_{MVE} = (w^{MVE})'\bar{R}^e = k(\bar{R}^e)'\Omega^{-1}\bar{R}^e$

$var(R^e{}_{MVE}) = (w^{MVE})'\Omega w^{MVE} = k^2(\bar{R}^e)'\Omega^{-1}\Omega\Omega^{-1}\bar{R}^e$
$= k^2(\bar{R}^e)'\Omega^{-1}\bar{R}^e$

So, the Sharpe Ratio squared for MVE is
$SR^2{}_{MVE} = \frac{(\bar{R}^e{}_{MVE})^2}{var(R^e{}_{MVE})} = (\bar{R}^e)'\Omega^{-1}\bar{R}^e$

Max Sharpe Ratio Value

```
SharpeRatioSq.Max = t(Avg.Hedged)%*%chol2inv(chol(Variance_matrix))%*%Avg.Hedged
sqrt(SharpeRatioSq.Max)
```

```
##           [,1]
## [1,] 0.1732051
```

4

Maximum sharpe ratio squared of stocks and market = Maximum sharpe ratio square of hedged stocks + sharpe ratio square of market

Max Sharpe Ratio = $(\bar{R}^e)'{\Sigma_F}^{-1}\bar{R}^e + (\alpha)'{\Sigma_e}^{-1}\alpha$

Where first term is sharpe ratio of factor portfolio (in this case market) and second term is sharpe ratio of alphas.

```
SharpeRatio.Market = 1/3
```

```
SharpeRatio.Combined = sqrt(SharpeRatioSq.Max + SharpeRatio.Market^2)
```

```
SharpeRatio.Combined
```

```
##           [,1]
## [1,] 0.3756476
```

5

5a

Weights of stocks and market to achieve maximum sharpe ratio and with expected volatility

```
AllReturns = c(Avg,Avg.Mkt)
#Calculate systematic variance and covariance (Betai * Betaj * market variance)
betas5 = c(betas,1)
systematicVar.5 = (betas5%*%t(betas5))*Variance.Mkt
fullCovarianceMatrix.5 = rbind(cbind(Variance_matrix,0),0) + systematicVar.5
SharpeRatio.Max.Combined = t(AllReturns)%*%chol2inv(chol(fullCovarianceMatrix.5))%*%AllReturns
Sd = 0.15
k = Sd/sqrt(SharpeRatio.Max.Combined)
weights_combined = (chol2inv(chol(fullCovarianceMatrix.5))%*%AllReturns) * as.numeric(k)
rownames(weights_combined) = c("Stock1","Stock2","Stock3","Market")
weights_combined
```

```
##               [,1]
## Stock1  0.39931043
## Stock2 -0.26620695
## Stock3  0.79862086
## Market  0.04880461
```

5b

Mean, Standard Deviation, Sharpe Ratio

```r
#Mean
Mean5 = AllReturns%*%weights_combined
#SD
Sd5 = sqrt(t(weights_combined)%*%fullCovarianceMatrix.5%*%weights_combined)
#Sharpe Ratio
SR5 = Mean5/Sd5
output = data.frame(Mean5,Sd5,SR5)
names(output) = c("Mean","SD","Sharpe Ratio")
output
```

```
##         Mean   SD Sharpe Ratio
## 1 0.05634714 0.15    0.3756476
```

6

6a

Mean, Standard Deviation, Sharpe Ratio of factor mimicking portfolio

```r
mimick.Weights = ((betas - mean(betas))/(length(betas)*(mean(betas^2)-mean(betas)^2)))
mimick.Return = mimick.Weights%*%Avg
systematicVar.stocks = (betas%*%t(betas))*Variance.Mkt
fullCovarianceMatrix.stocks = systematicVar.stocks +Variance_matrix
mimick.Sd = sqrt(t(mimick.Weights)%*%fullCovarianceMatrix.stocks%*%mimick.Weights)
mimick.sharpe = mimick.Return/mimick.Sd
output = data.frame(mimick.Return,mimick.Sd,mimick.sharpe)
names(output) = c("Mean","SD","Sharpe Ratio")
output
```

```
##          Mean        SD Sharpe Ratio
## 1 -0.03571429 0.6264168  -0.05701362
```

6b

Correlation between factor mimicking portfolio and market portfolio

```r
cor.mimick.market = mimick.Weights%*%(betas*sqrt(Variance.Mkt)/as.vector(mimick.Sd))
cor.mimick.market
```

```
##           [,1]
## [1,] 0.2394572
```

6c

Variance explained by the PCAs

```r
eigens = eigen(fullCovarianceMatrix.stocks)
output = eigens$values/sum(eigens$values)
names(output) = c("1st PCA","2nd PCA","3rd PCA")
output
```

```
##    1st PCA    2nd PCA    3rd PCA
## 0.80813177 0.13952986 0.05233837
```

6d

Portfolio Weights

```
portfolioweights = eigens$vectors
colnames(portfolioweights) = c("1st PCA","2nd PCA","3rd PCA")
row.names(portfolioweights) = c("1st Stock","2nd Stock","3rd Stock")
portfolioweights
```

```
##              1st PCA    2nd PCA    3rd PCA
## 1st Stock -0.4685584  0.7003778 -0.5384460
## 2nd Stock -0.7451113 -0.6407531 -0.1850531
## 3rd Stock -0.4746180  0.3144940  0.8220896
```

Factor loadings

```
loadings = matrix(nrow=3,ncol=3)
for(i in 1:length(eigens$values)){
  loadings[,i] = portfolioweights[,i]*sqrt(eigens$values[i])
}
colnames(loadings) = c("1st PCA","2nd PCA","3rd PCA")
row.names(loadings) = c("1st Stock","2nd Stock","3rd Stock")
loadings
```

```
##              1st PCA     2nd PCA     3rd PCA
## 1st Stock -0.1385058  0.08602585 -0.04050562
## 2nd Stock -0.2202548 -0.07870229 -0.01392097
## 3rd Stock -0.1402970  0.03862860  0.06184325
```

6e

The PCA Analysis shows that the 3 facts are significant in explaining the variance. The factor mimicking portfolio obtained through Fama-Macbeth doesn't completely resemble the market due to the presence of the intercept. It resembles the second PCA component which is a long-short portfolio (which has no correlation with market). Due to this difference with the market portfolio, the correlation doesn't come out to be 1.

If we do the Fama-Macbeth regression without the intercept, the factor mimicking portfolio will resemble the market portfolio and hence the correlation will be higher than this correlation.