Project 5

Yaman Yucel

2023-04-27

Project 5

Before answering questions I have computed betas for periods 01-Jan-2015 to 01-Jan-2020 named as period 1, and 01-Jan-2020 to 31-Mar-2023 named as period 2

```
a_all <- read.csv("stockData.csv", sep=",", header=TRUE)

# Use 5 year data to train
a_period1 <- a_all[1:60,]
r_period1 <- (a_period1[-1,4:ncol(a_period1)]-a_period1[-nrow(a_period1),4:ncol(a_period1)])/a_period1[-
r_m_period1 <- (a_period1[-1,3]-a_period1[-nrow(a_period1),3])/a_period1[-nrow(a_period1),3] # return o

#use test data to compute actual predictions
a_period2 <- a_all[61:nrow(a_all),]
r_period2 <- (a_period1[-1,4:ncol(a_period1)]-a_period1[-nrow(a_period1),4:ncol(a_period1)])/a_period1[-
r_m_period2 <- (a_period1[-1,3]-a_period1[-nrow(a_period1),3])/a_period1[-nrow(a_period1),3] # return o

n_stocks = 30

#Compute period 1 betas
mean_Rm_period1 = mean(r_m_period1)
var_Rm_period1 <- var(r_m_period1)
stdev_Rm_period1 <- var_Rm_period1^.5
mean_Ri_period1 = colMeans(r_period1)

betas_period1 = rep(0,n_stocks)
alphas_period1 = rep(0,n_stocks)
var_es_period1 = rep(0,n_stocks)
var_betas_period1 = rep(0,n_stocks)

for (i in 1:n_stocks){
  fit <- lm(r_period1[,i] ~ r_m_period1)
  betas_period1[i] = fit$coefficients[2]
  alphas_period1[i] = fit$coefficients[1]
  var_es_period1[i] = sum(fit$residuals^2)/ (nrow(r_period1) - 2)
  var_betas_period1[i] = vcov(fit)[2,2]
}
#Compute period 2 betas
mean_Rm_period2 = mean(r_m_period2)
var_Rm_period2 <- var(r_m_period2)
stdev_Rm_period2 <- var_Rm_period2^.5
mean_Ri_period2 = colMeans(r_period2)
```

```r
betas_period2 = rep(0,n_stocks)
alphas_period2 = rep(0,n_stocks)
var_es_period2 = rep(0,n_stocks)
var_betas_period2 = rep(0,n_stocks)

for (i in 1:n_stocks){
  fit <- lm(r_period2[,i] ~ r_m_period2)
  betas_period2[i] = fit$coefficients[2]
  alphas_period2[i] = fit$coefficients[1]
  var_es_period2[i] = sum(fit$residuals^2)/ (nrow(r_period2) - 2)
  var_betas_period2[i] = vcov(fit)[2,2]
}
```

Part a: Assume the single index model holds. Use only the stocks with positive betas in your data. Choose a value of Rf and find the optimal portfolio (point of tangency) using the optimization procedure as discussed in handout #12: http://www.stat.ucla.edu/~nchristo/statistics_c183_c283/statc183c283_tangent.pdf .

```r
#Construct covariance matrix of SIM
covariance_matrix_period1 = matrix(0,n_stocks,n_stocks)
for (i in 1:n_stocks)
{
  for(j in 1:n_stocks){
    if(i == j)
    {
      covariance_matrix_period1[i,j] = betas_period1[i] * betas_period1[i] * var_Rm_period1 + var_es_pe
    }else{
      covariance_matrix_period1[i,j] = betas_period1[i] * betas_period1[j] * var_Rm_period1
    }
  }
}

inv_covmat_single_index = solve(covariance_matrix_period1)
ones = rep(1,n_stocks)

A = as.numeric(t(mean_Ri_period1) %*% inv_covmat_single_index %*% ones)
B = as.numeric(t(mean_Ri_period1) %*% inv_covmat_single_index %*% mean_Ri_period1)
C = as.numeric(t(ones)  %*% inv_covmat_single_index %*% ones)
D = B*C - A^2

E <- seq(-0.2,0.2,.001)

sigmas <- sqrt(seq(1/C,0.03,.0001))
upper_part <- (A + sqrt(D*(C*sigmas^2 - 1)))*(1/C)
lower_part <- (A - sqrt(D*(C*sigmas^2 - 1)))*(1/C)

plot(sigmas, upper_part, lwd=5,type = "l",col = "green",xlim = c(0, 0.2), ylim= c(-0.2,0.2))
lines(sigmas,lower_part, lwd=5,type = "l",col = "green")

R_f = 0.002
R = mean_Ri_period1 - R_f

Z = inv_covmat_single_index %*% R
x_G = Z / sum(Z)
```

```r
print("Composition of tangent")
```

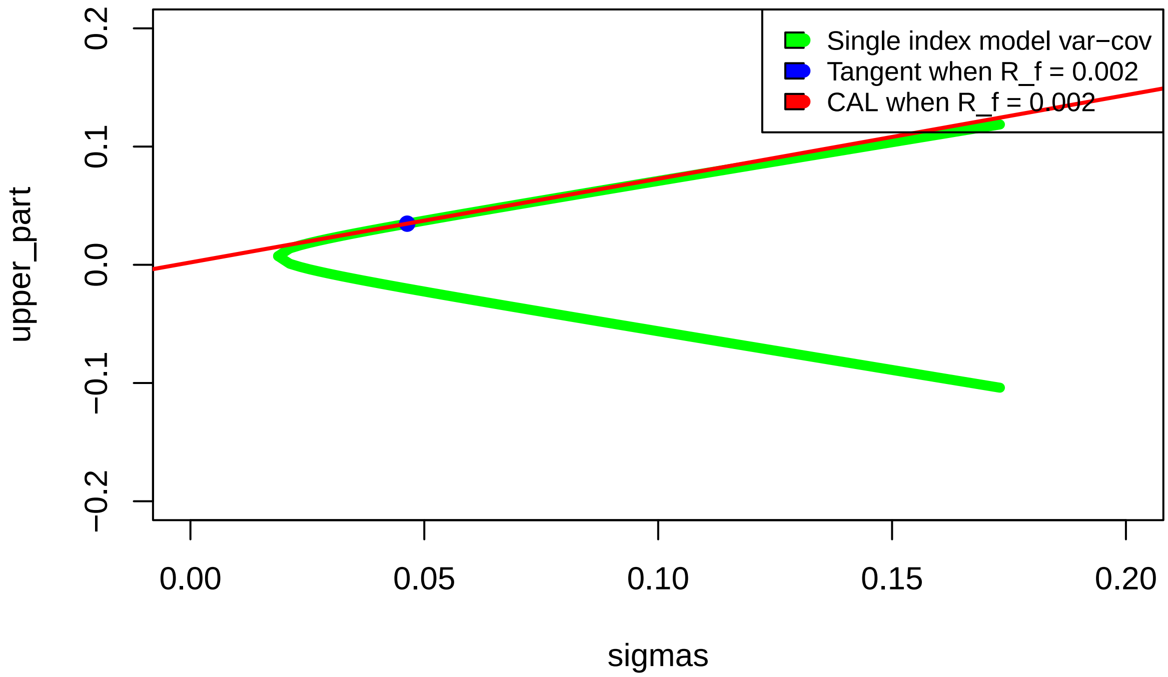
```
## [1] "Composition of tangent"
```

```r
print(x_G)
```

```
##               [,1]
##  [1,]  0.010937796
##  [2,]  0.319321021
##  [3,]  0.131841756
##  [4,]  0.101903830
##  [5,]  0.053551902
##  [6,]  0.113994968
##  [7,]  0.037017266
##  [8,]  0.055757221
##  [9,] -0.120504236
## [10,]  0.240965202
## [11,]  0.049880298
## [12,] -0.107303614
## [13,]  0.141226189
## [14,]  0.041272294
## [15,]  0.012135308
## [16,] -0.123990706
## [17,]  0.352251678
## [18,] -0.447051001
## [19,]  0.053015460
## [20,]  0.161050368
## [21,]  0.052121839
## [22,]  0.013297532
## [23,]  0.077646359
## [24,] -0.102002979
## [25,] -0.029028214
## [26,]  0.001020932
## [27,] -0.010733009
## [28,]  0.020750892
## [29,] -0.077968692
## [30,] -0.022377660
```

```r
varg <- t(x_G) %*% covariance_matrix_period1 %*% x_G
Rg <- t(x_G) %*% mean_Ri_period1
sigmag <- sqrt(varg)
points(sigmag,Rg, pch=19,lwd=1,col="blue")
abline(a = R_f, b = (Rg - R_f)/sigmag , lwd = 2, col = "red")


legend("topright",
       legend=c("Single index model var-cov","Tangent when R_f = 0.002", "CAL when R_f = 0.002"),
       col=c("green","blue","red"),
       pch = 19,
       fill =c("green","blue","red"),
       cex=0.8)
```



Part b: Adjusting the betas: Adjust the betas using Blume's and Vasicek's techniques. For the Blume technique use the two periods: 01- Jan-2015 to 01-Jan-2020 and 01-Jan-2020 to 31-Mar-2023. For the Vasicek technique use only the period 01-Jan-2014 to 01-Jan-2019.

Note: For the Blume technique our goal is to adjust the betas in 01-Jan-2020 to 31-Mar-2023 to be better forecasts for the betas in period 01-Apr-2023 to 01-Apr-2027.

For the Vasicek technique our goal is to adjust the betas in 01-Jan-2015 to 01-Jan-2020 to be better forecasts for the betas in period 01-Jan-2020 to 31-Mar-2023.

```
# Blume
# Use both periods to adjust betas
fit = lm(betas_period2 ~ betas_period1)
beta_adj_blume = fit$coef[1] + fit$coef[2] * betas_period2

print("Adjusted betas using blume technique")
```

```
## [1] "Adjusted betas using blume technique"
```

```
print(beta_adj_blume)
```

```
##  [1] 1.2929549 1.1593250 2.0248862 1.0718453 1.2582033 0.9442843 1.0393931
##  [8] 1.0530930 1.0025766 0.4342799 0.4776197 1.0354303 1.6022329 0.6063894
## [15] 1.0001717 2.2070856 0.4415925 0.7638836 0.3725548 0.3437258 0.4144253
## [22] 0.5458005 0.8674676 0.5747603 0.9214674 0.7875896 0.9919519 1.1254699
## [29] 0.8974680 0.9648233
```

```
# Vasicek
# In this part period 2 is the forecasted period and period 1 is the historical period
# So we do not use period 2 to adjust betas
```

```r
vasicek_term_1 = var_betas_period1*mean(betas_period1)/(var(betas_period1)+var_betas_period1)
vasicek_term_2 = var(betas_period1)*betas_period1/(var(betas_period1)+var_betas_period1)
beta_adj_vasicek <- vasicek_term_1 + vasicek_term_2
print("Adjusted betas using vasicek technique")
```

```
## [1] "Adjusted betas using vasicek technique"
```

```r
print(beta_adj_vasicek)
```

```
##  [1] 1.2144277 1.1324378 1.5364148 1.0474245 1.2109141 0.9432582 1.0251021
##  [8] 1.0285781 0.9927587 0.5177847 0.5412688 1.0220532 1.4595359 0.7812784
## [15] 0.9944871 1.8185117 0.4922264 0.7791615 0.4639264 0.4007623 0.4538070
## [22] 0.5801443 0.8757871 0.6405316 0.9280203 0.8019194 0.9757924 1.0104135
## [29] 0.9011495 0.9575712
```

Part c: Compute PRESS only for the Vasicek technique. (You can compute the PRESS only for the Vasicek technique because you have the actual betas in the period 01-Jan-2020 to 31-Mar-2023.)

```r
PRESS_direct = sum((beta_adj_vasicek - betas_period2)^2)/(n_stocks)

fit <- lm(betas_period2 ~ beta_adj_vasicek)
beta_regress = fit$coefficients[2]
r2 = summary(fit)$r.squared

#Bias Component
term1 = (mean(betas_period2) - mean(beta_adj_vasicek))^2
print("BIAS term:")
```

```
## [1] "BIAS term:"
```

```r
print(term1)
```

```
## [1] 0.0005371626
```

```r
Sa2 <- (29/30)*var(betas_period2)
Sp2 <- (29/30)*var(beta_adj_vasicek)

term2 = ((1-as.numeric(beta_regress))^2)*Sp2
print("Inefficiency")
```

```
## [1] "Inefficiency"
```

```r
print(term2)
```

```
## [1] 0.01084516
```

```r
term3 = (1-r2)*Sa2
print("Random error")
```

```
## [1] "Random error"
```

```r
print(term3)
```

```
## [1] 0.005209049
```

```r
PRESS_indirect = term1 + term2 + term3
print("Press with decomposition")
```

```
## [1] "Press with decomposition"
```

```r
print(PRESS_indirect)
```

```
## [1] 0.01659137
```

```r
print("Press direct")
```

```
## [1] "Press direct"
```

```r
print(PRESS_direct)
```

```
## [1] 0.01659137
```