HW4

Korawat Tanwisuth

March 9, 2018

1

maximize $0.05 * (L^{2/3}) * (K^{1/3})$

subject to $12 * L + 15 * K \leq 100000$

```
f <- function(x){
    return(-0.05*(x[1]^(2/3))*(x[2]^(1/3)))
}
gradf <- function(x){
    v1 <- (-0.05)*(2/3)*(x[1]^(-1/3))*(x[2]^(1/3))
    v2 <- (-0.05)*(1/3)*(x[2]^(-2/3))*(x[1]^(2/3))
    return(c(v1,v2))
}

ui <- matrix(c(-12,-15),ncol = 2)
ci <- matrix(c(-100000),ncol = 1)
ans1 <- constrOptim(c(5500,2200),f, grad=gradf, ui, ci)
-ans1$value
```

```
## [1] 204.6647
```

2

Let w_i = the weight invested in stock i $i = 1, 2, \ldots, n$ X_i = return of stock i $i = 1, 2, \ldots, n$ m_i = mean return of stock i $i = 1, 2, \ldots, n$

minimize $\sum_{i=1}^{n} w_i * w_j * Cov(X_i, X_j)$ $\forall i, j$ subject to $\sum_{i=1}^{n} w_i = 1$ $\forall i$ $\sum_{i=1}^{n} w_i * m_i \geq 0.01$ $\forall i$ $w_i \geq 0$ $\forall i$

```
library(quadprog)
stocks <- read.csv("homework4stocks.csv")
dates <- stocks[,1]
stock_ret <- stocks[,2:ncol(stocks)]
mean_ret <- apply(stock_ret,2,mean)
var_ret <- apply(stock_ret,2,var)
cor_ret <- cor(stock_ret)
n <- ncol(stock_ret)
A <- rbind(rep(1,n),mean_ret,diag(n))
Amat <- t(A)
Dmat <- 2*cov(stock_ret)
dvec <- rep(0,n)
bvec <- c(1,0.01,rep(0,n))
ans2 <- solve.QP(Dmat, dvec, Amat, bvec, meq=1)
sol <- ans2$solution
exp_mean <- sum(sol*mean_ret)
exp_var <- sum(sol*var_ret)
exp_sd <- sqrt(exp_var)
```

```r
names(sol) <- colnames(stock_ret)
sol
```

```
##            AA          AAPL           AXP            BA           CAT
##  2.031752e-17  5.169119e-02 -1.287909e-17  0.000000e+00 -2.836481e-17
##          CSCO            DD           DIS            EK           FDX
##  1.443921e-17 -5.000485e-17  1.024284e-16 -5.362116e-18 -4.004812e-17
##            GE            GT           HPQ           IBM            IP
##  6.304565e-17  6.094312e-17  2.485645e-02  1.924800e-18  1.241520e-17
##           JNJ           JPM            KO           MCD           MMM
##  1.527300e-02 -7.033633e-18 -2.894296e-18  1.394130e-01  1.583292e-18
##            MO           MRK          MSFT            PG             T
##  2.701720e-01  3.340617e-18 -4.571458e-17  1.255622e-01  5.833401e-02
##           UTX           WMT
##  5.682633e-18  3.146983e-01
```

```r
print(paste("Expected Mean", exp_mean))
```

```
## [1] "Expected Mean 0.01"
```

```r
print(paste("Expected Variance", exp_var))
```

```
## [1] "Expected Variance 0.00310129698830048"
```

```r
print(paste("Expected Standard Deviation",exp_sd))
```

```
## [1] "Expected Standard Deviation 0.0556892897090678"
```

3

```r
library(knitr)
df<- read.csv("variable_selection.csv")
lm2_1 <- lm(y~x1+x2,df)
lm2_2 <- lm(y~x1+x3,df)
lm2_3 <- lm(y~x2+x3,df)
lm1_1 <- lm(y~x1,df)
lm1_2 <- lm(y~x2,df)
lm1_3 <- lm(y~x3,df)
sse <- function(lm_ob){
    return(sum(lm_ob$residuals^2))
}
sse_vec <- c(sse(lm2_1),sse(lm2_2),sse(lm2_3),sse(lm1_1),sse(lm1_2),sse(lm1_3))
sse_names <- c("x1_x2","x1_x3","x2_x3","x1","x2","x3")
table_sse <- data.frame(sse_vec)
rownames(table_sse) <- sse_names
table_sse
```

```
##          sse_vec
## x1_x2   26.19087
## x1_x3 7860.08876
## x2_x3  878.18105
## x1    7901.29943
## x2     878.83582
## x3    8575.63588
```

This shows that the model with only x_1 and x_2 minimizes residual sum squared.

4

Let x_{ij} = allowable path from node i to node j $\forall i, j$

minimize $I^2 R$

subject to $x_{12} + x_{13} = 710$ $x_{12} - x_{23} - x_{24} = 0$ $x_{13} + x_{23} - x_{34} = 0$ $x_{24} + x_{34} = 710$

```
n <- 5
c1<- c(1,1,rep(0,3))
c2 <- c(1,0,-1,-1,0)
c3 <- c(0,1,1,0,-1)
c4 <- c(rep(0,3),1,1)
A <- rbind(c1,c2,c3,c4)
Amat <- t(A)
bvec <- c(710,0,0,710)
dvec <- rep(0,n)
Dmat <- matrix(0,n,n)
diag(Dmat) <- 2*c(1,4,6,12,3)
ans4<- solve.QP(Dmat, dvec, Amat, bvec, meq=length(bvec))$solution
```

```
names(ans4) <- c("x_12","x_13","x_23","x_24","x_34")
ans4
```

```
##     x_12     x_13     x_23     x_24     x_34
## 371.3846 338.6154 163.8462 207.5385 502.4615
```

5

Let y_i = actual spread of team i

$\hat{y}$ = the predicted spread $\forall i = 1, 2, \ldots, n$

minimize $\sum_{i=1}^{n} (y_i - \hat{y_i})^2$

```
nfl <- read.csv("nflratings.csv",header = FALSE)
colnames(nfl) <- c("Week","HT_index","VT_index","HT_score","VT_score")
avg_rate <- 85
APS <- nfl["HT_score"]-nfl["VT_score"]
HT_index <- as.vector(nfl[["HT_index"]])
VT_index <- as.vector(nfl[["VT_index"]])

pred_spread <- function(HTR,VTR,HTA){
    return(HTR-VTR+HTA)
}
n <- max(HT_index)
pred_err <- function(x){
    x <- (x-mean(x))+avg_rate
    HTR <- x[HT_index]
    VTR <- x[VT_index]
    HTA <- x[length(x)]
    prediction <- pred_spread(HTR,VTR,HTA)
    return(sum((APS-prediction)^2))
}
```

```r
guess <- rep(85,n+1)
res <-optim(guess,pred_err,method = "CG")
best_sol <- res$par
lowest <- res$val

lowest
```

```
## [1] 42925.68
```

```r
best_sol
```

```
##  [1] 87.110699 92.429802 95.334039 85.677339 91.348313 82.400403 90.132408
##  [8] 79.475359 94.709468 88.224120 73.092418 94.843931 89.572673 93.450703
## [15] 81.028134 79.476551 89.203614 94.653189 98.711020 98.217025 87.687233
## [22] 95.736768 77.621216 93.546500 89.230692 70.308303 95.194163 87.830290
## [29] 77.320184 81.759436 84.776631 82.724642  2.172733
```