

HW4_kb42582

Khyathi Balusu

3/10/2020

Question 1

```
max.machines <- function(K)
{
  L = (100000 - 15*K)/12
  num.machines = 0.05 * (L^(2/3)) * K^(1/3)
  return(-num.machines)
}

result = optim(1000, max.machines, method='BFGS')

result$value

## [1] -204.6684
```

Maximum number of machines that can be manufactured = 204

Question 2

```
library(quadprog)
stocks = read.csv('homework4stocks.csv', header=TRUE)
mean.returns = colMeans(stocks[,2:ncol(stocks)])
variance.returns = apply(stocks[,2:ncol(stocks)], 2, var)
sdev.returns = sqrt(variance.returns)
rho = cor(stocks[,2:ncol(stocks)], use='pairwise.complete.obs')
covMat = diag(sdev.returns) %*% rho %*% diag(sdev.returns)

D = 2*covMat
d = rep(0,27)
A = matrix(c(rep(1,27), rep(-1,27), mean.returns), 27)
b = c(1, -1, 0.01)

Q2.result = solve.QP(D,d,A,b)
weights = Q2.result$solution

### Returns
cat('\nMean returns of portfolio = ',sum(weights*mean.returns))

##
## Mean returns of portfolio = 0.01
```

```

### Variance
cat('\nMean variance of portfolio = ',sum(weights*variance.returns))

##
## Mean variance of portfolio = -0.0001122938

### Standard Deviation
cat('\nMean standard deviation of portfolio = ',sum(weights*sdev.returns))

##
## Mean standard deviation of portfolio = 0.03455156

```

Investment fractions:

```

weights

## [1] -0.11973810 0.03867549 -0.07492591 0.16636830 -0.01034152
## [6] 0.02541211 -0.07436094 -0.09669737 -0.03890179 0.01443292
## [11] -0.11294606 -0.05983834 0.09727562 0.06822219 0.16038834
## [16] 0.11682045 -0.07211667 0.07338738 0.02102473 0.01460696
## [21] 0.22763626 -0.07466514 -0.02873011 0.13824440 0.20171402
## [26] 0.01403427 0.38501850

```

Question 3

```

data = read.csv('variable_selection.csv')
data = data[,2:ncol(data)]
y = data[,1]
x1 = data[,2]
x2 = data[,3]
x3 = data[,4]

reg1 = lm(y~x1)
reg2 = lm(y~x2)
reg3 = lm(y~x3)
reg4 = lm(y~x1+x2)
reg5 = lm(y~x2+x3)
reg6 = lm(y~x1+x3)

reg = c(reg1, reg2, reg3, reg4, reg5, reg6)
MSE = rep(0,6)

MSE1 = sum(resid(reg1)^2)
MSE2 = sum(resid(reg2)^2)
MSE3 = sum(resid(reg3)^2)
MSE4 = sum(resid(reg4)^2)
MSE5 = sum(resid(reg5)^2)
MSE6 = sum(resid(reg6)^2)

cat(MSE1, MSE2, MSE3, MSE4, MSE5, MSE6)

```

```
## 7901.299 878.8358 8575.636 26.19087 878.1811 7860.089
```

Lowest MSE: reg4 ($y \sim x_1 + x_2$)

Question 4

Minimize total power

$$i_1^2 + 4i_4^2 + 6i_6^2 + 12i_{12}^2 + 3i_{13}^2$$

The constraints:

Node 1:

$$i_1 + i_4 = 710$$

Node 2 :

$$i_1 - i_6 - i_{12} = 0$$

Node 3 :

$$i_4 + i_6 - i_3 = 0$$

Node 4 :

$$i_{12} + i_3 = 710$$

```
d = rep(0,5)
D = diag(c(1,4,6,12,3),5)

A = matrix(c(c(1,1,0,0,0), c(1,0,-1,-1,0), c(0,1,1,0,-1), c(0,0,0,1,1)), 5)
b = c(710,0,0,710)

Q4.result = solve.QP(D,d,A,b)
Q4.result$solution

## [1] 371.3846 338.6154 163.8462 207.5385 502.4615
```

The current per each resistor:

R1 = 371.38A

R4 = 338.62A

R6 = 163.85A

R12 = 207.54A

R3 = 502.46A

Question 5

```
NFL = read.csv('nflratings.csv', header=FALSE, col.names=c('week',
'home_index', 'away_index', 'home_points', 'away_points'))

get.SSE = function(ratings)
{
  SSE = 0
  home_adv = ratings[33]
  actualSpread.vec = NFL$home_points - NFL$away_points

  for(i in seq(1, nrow(NFL)))
  {
    home_index = NFL[i,2]
    away_index = NFL[i,3]
    predSpread = ratings[home_index] - ratings[away_index] + home_adv
    SSE = SSE + (actualSpread.vec[i] - predSpread)^2
  }
  return(SSE)
}

ratings = rep(0,33)
Q5.res = optim(ratings, get.SSE, method='CG')
norm.ratings = Q5.res$par[1:32]

final.ratings = norm.ratings + 85

home_adv = Q5.res$par[33]

cat('\nFinal Ratings : ',final.ratings)

##
## Final Ratings : 84.52235 89.84145 92.74569 83.08899 88.75996 79.81205
87.54406 76.88701 92.12112 85.63577 70.50407 92.25558 86.98432 90.86235
78.43978 76.8882 86.61526 92.06484 96.12267 95.62867 85.09888 93.14842
75.03286 90.95815 86.64234 67.71995 92.60581 85.24194 74.73183 79.17108
82.18828 80.13629

cat('\n\nHome Advantage',home_adv)

##
##
## Home Advantage 2.172733
```