Array problem

Mauricio Collado

1/23/2021

```
# Packages
library(tidyverse)
library(Matrix)
library(kableExtra)
library(ggplot2)
#library(plotly)
#library(lpSolve)
#library(lpSolveAPI)
library(quadprog)
```

```
# First, set seed to ensure we have the same result
set.seed(666)



# set the number of species and technologies we want to study
s <- 2 #species
n <- 2 #tecnologies
m <- 1 #quota baskets
```

```
# D (costs)

# set cost for each tech
Cost <- c(1,1)
Cost2 <- 2*Cost

# nxn
D <- matrix(0, nrow=n,ncol=n)
diag(D) <- Cost2
```

```
# d

#prices (sx1)
p <- c(200,200)

# stock in period 0
X_0 <- c(0.5,0.5)

# stock matrix (sxs)
B <- matrix(0, nrow=s,ncol=s)
diag(B) <- X_0

# catchability matrix (sxn)
coefficient <- c(0.05,0.04,
                 0.04,0.05)
Z <- matrix(coefficient, nrow=s,ncol=n)

#d t(sx1)(sxs)(sxn)=(1xn)
t_d <- t(p)%*%B%*%Z

#nx1
d <- t(t_d)
```

```
# quota basket matrix (mxs)
coefficient_qb <- c(1,1)
W <- matrix(coefficient_qb, nrow=m,ncol=s) # CORRECTION it was name D in the document, and D is
 defined as cost. I renamed it as W.

#A(mxs)(sxn)(sxs)=(mxs)
A <- -1*W%*%B%*%Z
t_A <- t(A)
```

```
#quota basket caps (mx1)
b_0 <- c(10)
b <- -1*b_0
```

```
# quadprog

# solve.QP(Dmat, dvec, Amat, bvec, meq=0, factorized=FALSE)
# (1xn).(nxn)(nx1)-(1xn)(nx1) subject to :(mxn)(nx1) = (mx1) ≥ (mx1)
# D: symmetric matrix with the quadratic component (Dmat)
# d: linear term (dvec)
# A: matrix with linear constraints (Amat)
# b: constraints (bvec)

# (1xn).(nxn)(nx1)-(1xn)(nx1) subject to :(mxn)(nx1) = (mx1) ≥ (mx1)

solve.QP(D,d,t_A,b, meq=1)
```

```
## $solution
## [1] 111.1111 111.1111
##
## $value
## [1] 22691.36
##
## $unconstrained.solution
## [1] 4.5 4.5
##
## $iterations
## [1] 2 0
##
## $Lagrangian
## [1] 4738.272
##
## $iact
## [1] 1
```