Stat151ALab1

Jiyoon Clover Jeong

8/28/2017

한국어 사용하기

```
x <- 1:9
matrix(x,3,3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
matrix(x,3,3,byrow=TRUE)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
diag(1,5,5)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
```

```
a1 <- c(2, 3, 6, 7, 10)
a2 <- c(1.88, 2.05, 1.70, 1.60, 1.78)
a3 <- c(80, 90, 70, 50, 75)

a = matrix(data=c(a1,a2,a3),5,3)
b1 <- c(1, 4, 5, 8, 9)
b2 <- c(1.22, 1.05, 3.60, 0.40, 2.54)
b3 <- c(20, 40, 30, 80, 100)
b =matrix(data=c(b1,b2,b3), 3,5, byrow = TRUE)



a %*% b
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]
## [1,] 1604.294 3209.974 2416.768 6416.752 8022.775
## [2,] 1805.501 3614.153 2722.380 7224.820 9032.207
## [3,] 1408.074 2825.785 2136.120 5648.680 7058.318
## [4,] 1008.952 2029.680 1540.760 4056.640 5067.064
## [5,] 1512.172 3041.869 2306.408 6080.712 7594.521
```

```
b %*% a
```

```
##         [,1]     [,2]    [,3]
## [1,]  190.00  47.4000  1865.0
## [2,]   55.39  15.7273   654.6
```

```
## [3,] 1900.00 476.6000 18800.0
```

```
t(a) %*% t(b)
```

```
##        [,1]     [,2]    [,3]
## [1,]  190.0  55.3900  1900.0
## [2,]   47.4  15.7273   476.6
## [3,] 1865.0 654.6000 18800.0
```

```
t(b) %*% t(a)
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]
## [1,] 1604.294 1805.501 1408.074 1008.952 1512.172
## [2,] 3209.974 3614.153 2825.785 2029.680 3041.869
## [3,] 2416.768 2722.380 2136.120 1540.760 2306.408
## [4,] 6416.752 7224.820 5648.680 4056.640 6080.712
## [5,] 8022.775 9032.207 7058.318 5067.064 7594.521
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
c = iris[1] + iris[2] *2 + iris[3] * 3 + iris[4] *4
cc = as.matrix(iris[,-5]) %*%  matrix(1:4,4,1)


# test vnorm() with 1:5
v <- 1:5

vnorm <- function(x) {
  a = t(x) %*% x
  return(a^(1/2))

}

size = vnorm(v)

u = v / size


is_square <- function(x) {
  x = as.matrix(x)
  if(nrow(x) == ncol(x))
    return(TRUE)
  else
    return(FALSE)

}

is_square(u)
```

```
## [1] FALSE
```

```
mtrace <- function(x) {
  if(is_square(x) == FALSE) {
    cat("Matrix is not square")
    return(FALSE)
  }
  else{
    return(sum(diag(x)))
  }
}

A = matrix(1:9,3,3)
B = matrix(2:10,3,3)
mtrace(A)
```

```
## [1] 15
```

```
identical(mtrace(A+B),mtrace(A) + mtrace(B))
```

```
## [1] TRUE
```

```
identical(mtrace(5*A),5*mtrace(A))
```

```
## [1] TRUE
```

$tr(A+B) \ = \sum_{i=1}^{p}(a+b)_{ii} = \sum_{i=1}^{p}(a_{ii}+b_{ii}) = \sum_{i=1}^{p}(a_{ii}) + \sum_{i=1}^{p}(b_{ii}) = tr(A) + tr(B)$

$tr(cA) = \sum_{i=1}^{p}(ca)_{ii} = c\sum_{i=1}^{p}(a)_{ii} = c * tr(A)$

since the diagonal entries in square matrix doesn't change when we transpose square matrix, we will prove $tr(AB) == tr(BA)$

since matrix, we will prove $tr(AB) == tr(BA)$$

```
t1 = mtrace(crossprod(A,B))
t2 = mtrace(tcrossprod(A,B))
t3 = mtrace(crossprod(B,A))
t4 = mtrace(tcrossprod(B,A))

sapply(c(t1, t2, t3, t4), identical, t1)
```

```
## [1] TRUE TRUE TRUE TRUE
```

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```r
M <- mtcars[, c(1,3,4,5,6)]
head(M)
```

```
##                   mpg disp  hp drat    wt
## Mazda RX4         21.0  160 110 3.90 2.620
## Mazda RX4 Wag     21.0  160 110 3.90 2.875
## Datsun 710        22.8  108  93 3.85 2.320
## Hornet 4 Drive    21.4  258 110 3.08 3.215
## Hornet Sportabout 18.7  360 175 3.15 3.440
## Valiant           18.1  225 105 2.76 3.460
```

```r
apply(M, 2, mean)
```

```
##        mpg       disp         hp       drat         wt
##  20.090625 230.721875 146.687500   3.596563   3.217250
```

```r
Mc <- scale(M, scale = FALSE)


#Confirm that variables in Mc are mean-centered by calculating the vector of #column-means    <---??
colMeans(Mc)
```

```
##           mpg          disp            hp          drat            wt
##  4.440892e-16 -1.199041e-14  0.000000e+00 -1.526557e-16  3.469447e-17
```

```r
sweeped <- sweep(M, 2, colMeans(M))

head(sweeped)
```

```
##                       mpg        disp       hp       drat       wt
## Mazda RX4         0.909375  -70.721875 -36.6875  0.3034375 -0.59725
## Mazda RX4 Wag     0.909375  -70.721875 -36.6875  0.3034375 -0.34225
## Datsun 710        2.709375 -122.721875 -53.6875  0.2534375 -0.89725
## Hornet 4 Drive    1.309375   27.278125 -36.6875 -0.5165625 -0.00225
## Hornet Sportabout -1.390625  129.278125  28.3125 -0.4465625  0.22275
## Valiant           -1.990625   -5.721875 -41.6875 -0.8365625  0.24275
```

```r
head(Mc)
```

```
##                       mpg        disp       hp       drat       wt
## Mazda RX4         0.909375  -70.721875 -36.6875  0.3034375 -0.59725
## Mazda RX4 Wag     0.909375  -70.721875 -36.6875  0.3034375 -0.34225
## Datsun 710        2.709375 -122.721875 -53.6875  0.2534375 -0.89725
## Hornet 4 Drive    1.309375   27.278125 -36.6875 -0.5165625 -0.00225
## Hornet Sportabout -1.390625  129.278125  28.3125 -0.4465625  0.22275
## Valiant           -1.990625   -5.721875 -41.6875 -0.8365625  0.24275
```

```r
#if you want to add 1 to the 1st row, 2 to the 2nd, etc... of the matrix you #defined, you will do:
#sweep (M, 1, c (1: 4), "+")

apply(M,2, max)
```

```
##     mpg    disp      hp    drat      wt
##  33.900 472.000 335.000   4.930   5.424
```

```r
sweep(M,2, apply(M,2,max),"/")
```

```
##                           mpg      disp        hp      drat        wt
```

```
## Mazda RX4           0.6194690 0.3389831 0.3283582 0.7910751 0.4830383
## Mazda RX4 Wag       0.6194690 0.3389831 0.3283582 0.7910751 0.5300516
## Datsun 710          0.6725664 0.2288136 0.2776119 0.7809331 0.4277286
## Hornet 4 Drive      0.6312684 0.5466102 0.3283582 0.6247465 0.5927360
## Hornet Sportabout   0.5516224 0.7627119 0.5223881 0.6389452 0.6342183
## Valiant             0.5339233 0.4766949 0.3134328 0.5598377 0.6379056
## Duster 360          0.4218289 0.7627119 0.7313433 0.6511156 0.6581858
## Merc 240D           0.7197640 0.3108051 0.1850746 0.7484787 0.5881268
## Merc 230            0.6725664 0.2983051 0.2835821 0.7951318 0.5807522
## Merc 280            0.5663717 0.3550847 0.3671642 0.7951318 0.6342183
## Merc 280C           0.5250737 0.3550847 0.3671642 0.7951318 0.6342183
## Merc 450SE          0.4837758 0.5843220 0.5373134 0.6227181 0.7503687
## Merc 450SL          0.5103245 0.5843220 0.5373134 0.6227181 0.6876844
## Merc 450SLC         0.4483776 0.5843220 0.5373134 0.6227181 0.6969027
## Cadillac Fleetwood  0.3067847 1.0000000 0.6119403 0.5943205 0.9679204
## Lincoln Continental 0.3067847 0.9745763 0.6417910 0.6085193 1.0000000
## Chrysler Imperial   0.4336283 0.9322034 0.6865672 0.6551724 0.9854351
## Fiat 128            0.9557522 0.1667373 0.1970149 0.8275862 0.4056047
## Honda Civic         0.8967552 0.1603814 0.1552239 1.0000000 0.2977507
## Toyota Corolla      1.0000000 0.1506356 0.1940299 0.8559838 0.3383112
## Toyota Corona       0.6342183 0.2544492 0.2895522 0.7505071 0.4544617
## Dodge Challenger    0.4572271 0.6737288 0.4477612 0.5598377 0.6489676
## AMC Javelin         0.4483776 0.6440678 0.4477612 0.6389452 0.6332965
## Camaro Z28          0.3923304 0.7415254 0.7313433 0.7565923 0.7079646
## Pontiac Firebird    0.5663717 0.8474576 0.5223881 0.6247465 0.7088864
## Fiat X1-9           0.8053097 0.1673729 0.1970149 0.8275862 0.3567478
## Porsche 914-2       0.7669617 0.2548729 0.2716418 0.8985801 0.3945428
## Lotus Europa        0.8967552 0.2014831 0.3373134 0.7647059 0.2789454
## Ford Pantera L      0.4660767 0.7436441 0.7880597 0.8559838 0.5844395
## Ferrari Dino        0.5811209 0.3072034 0.5223881 0.7342799 0.5106932
## Maserati Bora       0.4424779 0.6377119 1.0000000 0.7180527 0.6581858
## Volvo 142E          0.6312684 0.2563559 0.3253731 0.8336714 0.5125369
```

```
zerotoone = apply(M, 2, FUN = function(X) (X - min(X))/diff(range(X)))
head(zerotoone)
```

```
##                       mpg      disp        hp      drat        wt
## Mazda RX4         0.4510638 0.2217511 0.2049470 0.5253456 0.2830478
## Mazda RX4 Wag     0.4510638 0.2217511 0.2049470 0.5253456 0.3482485
## Datsun 710        0.5276596 0.0920429 0.1448763 0.5023041 0.2063411
## Hornet 4 Drive    0.4680851 0.4662010 0.2049470 0.1474654 0.4351828
## Hornet Sportabout 0.3531915 0.7206286 0.4346290 0.1797235 0.4927129
## Valiant           0.3276596 0.3838863 0.1872792 0.0000000 0.4978266
```

```
D <- as.matrix(sweep(M, 2, colMeans(M)))
# substract column mean from each cell
n = ncol(D)
covmatrix <- (n-1)^(-1) * (t(D) %*% D)
head(covmatrix)
```

```
##              mpg        disp         hp        drat         wt
## mpg    281.51180  -4906.5034 -2485.6734   17.011742 -39.654306
## disp -4906.50336 119046.1987 52088.9797 -364.746148 834.552581
## hp   -2485.67344  52088.9797 36431.7188 -127.496094 342.493125
## drat    17.01174   -364.7461  -127.4961    2.215580  -2.888586
## wt     -39.65431    834.5526   342.4931   -2.888586   7.419687
```

```
head(cov(M))
```

```
##              mpg        disp         hp        drat          wt
## mpg    36.324103  -633.09721 -320.73206   2.1950635  -5.1166847
## disp -633.097208 15360.79983 6721.15867 -47.0640192 107.6842040
## hp   -320.732056  6721.15867 4700.86694 -16.4511089  44.1926613
## drat    2.195064   -47.06402  -16.45111   0.2858814  -0.3727207
## wt     -5.116685   107.68420   44.19266  -0.3727207   0.9573790
```

```
# http://www.itl.nist.gov/div898/handbook/pmc/section5/pmc541.htm

STD <- diag(diag(covmatrix)^(-1/2))


#constructs the correlation matrix
cormatrix <- STD %*% covmatrix %*% STD
head(cormatrix)
```

```
##            [,1]       [,2]       [,3]       [,4]       [,5]
## [1,]  1.0000000 -0.8475514 -0.7761684  0.6811719 -0.8676594
## [2,] -0.8475514  1.0000000  0.7909486 -0.7102139  0.8879799
## [3,] -0.7761684  0.7909486  1.0000000 -0.4487591  0.6587479
## [4,]  0.6811719 -0.7102139 -0.4487591  1.0000000 -0.7124406
## [5,] -0.8676594  0.8879799  0.6587479 -0.7124406  1.0000000
```

```
head(cor(M))
```

```
##             mpg       disp         hp       drat         wt
## mpg   1.0000000 -0.8475514 -0.7761684  0.6811719 -0.8676594
## disp -0.8475514  1.0000000  0.7909486 -0.7102139  0.8879799
## hp   -0.7761684  0.7909486  1.0000000 -0.4487591  0.6587479
## drat  0.6811719 -0.7102139 -0.4487591  1.0000000 -0.7124406
## wt   -0.8676594  0.8879799  0.6587479 -0.7124406  1.0000000
```

```
cyl <- factor(mtcars$cyl)
# all categories


# a = model.matrix(~0 + cyl, M)
# a <- c(1,2,3,1,4,7,9,1)
# sapply(c("1","3"), function(x) as.integer(x == a))
# sapply(c(1,2,3,4,5,6), function(x) {as.integer(x == a)})




dummify <- function(x, all) {
  n = length(x)    # number of rows
  a = matrix(0, n, nlevels(x))
  colnames(a) <- levels(x)
  for(i in 1:n){
    a[i, as.character(x[i])] = 1
  }
  if(all==T)
    return(a)
  else
```

```
    return(a[,-nlevels(x)])

}


CYL1 <- dummify(cyl, all = TRUE)
head(CYL1)
```

```
##      4 6 8
## [1,] 0 1 0
## [2,] 0 1 0
## [3,] 1 0 0
## [4,] 0 1 0
## [5,] 0 0 1
## [6,] 0 1 0
```

```
# minus one category
CYL2 <- dummify(cyl, all = FALSE)
head(CYL2)
```

```
##      4 6
## [1,] 0 1
## [2,] 0 1
## [3,] 1 0
## [4,] 0 1
## [5,] 0 0
## [6,] 0 1
```

```
gear <- factor(mtcars$gear)

crosstable <- function(first, second) {
  a = dummify(first, T)
  b = dummify(second, T)
  return (t(a) %*% b)
}

xtb <- crosstable(cyl, gear)
head(xtb)
```

```
##    3 4 5
## 4  1 8 2
## 6  2 4 1
## 8 12 0 2
```