Lab1 - Jin Kweon (3032235207)

Jin Kweon

8/26/2017

```
#matrix -> rows = number of objects/observations & cols = number of features/variables

"+"(2,3) #fancy version of summation
```

```
## [1] 5
```

```
apropos("log") #list out all the functions that include "log"
```

```
##  [1] ".__C__logical"         ".__C__logLik"
##  [3] ".__T__Logic:base"      "as.data.frame.logical"
##  [5] "as.logical"            "as.logical.factor"
##  [7] "dlogis"                "is.logical"
##  [9] "log"                   "log10"
## [11] "log1p"                 "log2"
## [13] "logb"                  "Logic"
## [15] "logical"               "logLik"
## [17] "loglin"                "plogis"
## [19] "qlogis"                "rlogis"
## [21] "SSlogis"
```

```
x <- 1:9 #vector
mat_x <- matrix(x, 3, 3) #by column
mat_x2 <- matrix(x, 3, 3, byrow = T) #by row

diag(5) #Create an 5*5 identity matrix
```

```
##      [,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
```

```
#create three vectors and combine it together to form a matrix.
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 <- cbind(a1, a2, a3)




#create three vectors and combine it together to form a matrix.
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 <- rbind(b1, b2, b3)
```

```r
# matrix multiplication
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
```

```r
B %*% A
```

```
##         a1       a2      a3
## b1  190.00  47.4000  1865.0
## b2   55.39  15.7273   654.6
## b3 1900.00 476.6000 18800.0
```

```r
t(A) %*% t(B)
```

```
##        b1       b2      b3
## a1  190.0  55.3900  1900.0
## a2   47.4  15.7273   476.6
## a3 1865.0 654.6000 18800.0
```

```r
#t(B) %*% crossprod(A, B) #Not computable!!!
```

```r
#Play around with the R-embedded data, iris.
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
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 25           4.8         3.4          1.9         0.2     setosa
## 26           5.0         3.0          1.6         0.2     setosa
## 27           5.0         3.4          1.6         0.4     setosa
## 28           5.2         3.5          1.5         0.2     setosa
## 29           5.2         3.4          1.4         0.2     setosa
## 30           4.7         3.2          1.6         0.2     setosa
## 31           4.8         3.1          1.6         0.2     setosa
## 32           5.4         3.4          1.5         0.4     setosa
## 33           5.2         4.1          1.5         0.1     setosa
## 34           5.5         4.2          1.4         0.2     setosa
## 35           4.9         3.1          1.5         0.2     setosa
## 36           5.0         3.2          1.2         0.2     setosa
## 37           5.5         3.5          1.3         0.2     setosa
## 38           4.9         3.6          1.4         0.1     setosa
## 39           4.4         3.0          1.3         0.2     setosa
## 40           5.1         3.4          1.5         0.2     setosa
## 41           5.0         3.5          1.3         0.3     setosa
## 42           4.5         2.3          1.3         0.3     setosa
## 43           4.4         3.2          1.3         0.2     setosa
## 44           5.0         3.5          1.6         0.6     setosa
## 45           5.1         3.8          1.9         0.4     setosa
## 46           4.8         3.0          1.4         0.3     setosa
## 47           5.1         3.8          1.6         0.2     setosa
## 48           4.6         3.2          1.4         0.2     setosa
## 49           5.3         3.7          1.5         0.2     setosa
## 50           5.0         3.3          1.4         0.2     setosa
## 51           7.0         3.2          4.7         1.4 versicolor
## 52           6.4         3.2          4.5         1.5 versicolor
## 53           6.9         3.1          4.9         1.5 versicolor
## 54           5.5         2.3          4.0         1.3 versicolor
## 55           6.5         2.8          4.6         1.5 versicolor
## 56           5.7         2.8          4.5         1.3 versicolor
## 57           6.3         3.3          4.7         1.6 versicolor
## 58           4.9         2.4          3.3         1.0 versicolor
## 59           6.6         2.9          4.6         1.3 versicolor
## 60           5.2         2.7          3.9         1.4 versicolor
## 61           5.0         2.0          3.5         1.0 versicolor
## 62           5.9         3.0          4.2         1.5 versicolor
## 63           6.0         2.2          4.0         1.0 versicolor
## 64           6.1         2.9          4.7         1.4 versicolor
## 65           5.6         2.9          3.6         1.3 versicolor
## 66           6.7         3.1          4.4         1.4 versicolor
## 67           5.6         3.0          4.5         1.5 versicolor
## 68           5.8         2.7          4.1         1.0 versicolor
## 69           6.2         2.2          4.5         1.5 versicolor
## 70           5.6         2.5          3.9         1.1 versicolor
## 71           5.9         3.2          4.8         1.8 versicolor
## 72           6.1         2.8          4.0         1.3 versicolor
## 73           6.3         2.5          4.9         1.5 versicolor
## 74           6.1         2.8          4.7         1.2 versicolor
## 75           6.4         2.9          4.3         1.3 versicolor
## 76           6.6         3.0          4.4         1.4 versicolor
## 77           6.8         2.8          4.8         1.4 versicolor
## 78           6.7         3.0          5.0         1.7 versicolor
## 79           6.0         2.9          4.5         1.5 versicolor
## 80           5.7         2.6          3.5         1.0 versicolor
## 81           5.5         2.4          3.8         1.1 versicolor
## 82           5.5         2.4          3.7         1.0 versicolor
## 83           5.8         2.7          3.9         1.2 versicolor
## 84           6.0         2.7          5.1         1.6 versicolor
## 85           5.4         3.0          4.5         1.5 versicolor
## 86           6.0         3.4          4.5         1.6 versicolor
## 87           6.7         3.1          4.7         1.5 versicolor
## 88           6.3         2.3          4.4         1.3 versicolor
## 89           5.6         3.0          4.1         1.3 versicolor
## 90           5.5         2.5          4.0         1.3 versicolor
## 91           5.5         2.6          4.4         1.2 versicolor
## 92           6.1         3.0          4.6         1.4 versicolor
## 93           5.8         2.6          4.0         1.2 versicolor
## 94           5.0         2.3          3.3         1.0 versicolor
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2         2.9          4.3         1.3 versicolor
## 99           5.1         2.5          3.0         1.1 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 101          6.3         3.3          6.0         2.5  virginica
## 102          5.8         2.7          5.1         1.9  virginica
## 103          7.1         3.0          5.9         2.1  virginica
## 104          6.3         2.9          5.6         1.8  virginica
## 105          6.5         3.0          5.8         2.2  virginica
## 106          7.6         3.0          6.6         2.1  virginica
## 107          4.9         2.5          4.5         1.7  virginica
## 108          7.3         2.9          6.3         1.8  virginica
## 109          6.7         2.5          5.8         1.8  virginica
## 110          7.2         3.6          6.1         2.5  virginica
## 111          6.5         3.2          5.1         2.0  virginica
## 112          6.4         2.7          5.3         1.9  virginica
## 113          6.8         3.0          5.5         2.1  virginica
## 114          5.7         2.5          5.0         2.0  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 116          6.4         3.2          5.3         2.3  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 130          7.2         3.0          5.8         1.6  virginica
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 
```

```
dim(iris)
```

```
## [1] 150   5
```

```
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
```

```
lincomb <- as.matrix(iris[1:4]) %*% c(1:4)
```

```
lincomb2 <- (1 * diag(150) %*% iris$Sepal.Length) + (2 * diag(150) %*% iris$Sepal.Width) + (3 * diag(150
```

```
# Check out rdist help to get more of these!!!!
# 2-norm == euclidean norm == L2 norm
# 1-norm == manhattan norm
# max norm == chebyshev
```

```
# dot product ~ inner product ==> shows similarity (As angle between x and y gets closer, its bigger an
# X t(X) -> observation view & t(X) X ->  feature/variable view ===> Then, each element x_ij shows the
```

```
#Define vnorm.
v <- 1:5
vnorm <-function(x){
  sqrt(t(x) %*% x)
}
vnorm(v)
```

```
##          [,1]
## [1,] 7.416198
```

```
u <- v / vnorm(v) #unit vector
```

```
#Check whether a matrix is square.
is_square <- function(A){
  if(dim(A)[1] == dim(A)[2]){
    T
  }else{
    F
  }
}
```

```
#Trace is only defined for square matrix.
mtrace <- function(A){
  if(is_square(A) == T){
    sum(diag(A))
  }else{
    NA
  }
}




#Definte p and q to check mtrace is a linear mapping.
p <- matrix(1:9, 3, 3)
q <- matrix(10:18, 3, 3)

#Check it is a linear, by confirming additivity and homogeneity
mtrace(p + q) == mtrace(p) + mtrace(q)
```

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

```
c <- runif(1, 1, 500) #randomly pick number.
mtrace(c * p) == c *mtrace(p)
```

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

```
#Verify all traces are equal.
tr1 <- mtrace(crossprod(p, q))
tr2 <- mtrace(tcrossprod(p, q))
tr3 <- mtrace(crossprod(q, p))
tr4 <- mtrace(tcrossprod(q, p))

lapply(list(tr1, tr2, tr3, tr4), identical, tr1)
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
```

```
sapply(list(tr1, tr2, tr3, tr4), identical, tr1)
```

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

```
all(sapply(list(tr1, tr2, tr3, tr4), identical, tr1)) #If you apply "all" into lapply, it shows warning
```

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

```
#know how to get a determinant mathematically -> only defined in square matrix.
```

$Here\ is\ a\ proof\ of\ four\ traces\ being\ equal:$

$$tr(X^TY) \ = \ tr(XY^T) \ = \ tr(Y^TX) \ = \ tr(YX^T)$$

$Let\ the\ element\ of\ matrix\ X\ be\ x_{ij}\ and\ element\ of\ matrix\ Y\ be\ y_{ij}\ where\ X\ and\ Y\ are\ both\ square\ with\ the\ length\ of\ row\ a$

$tr(X^TY) = (X^TY)_{11} \ + \ ... \ + \ (X^TY)_{nn} = x^T_{11} \ y_{11} \ + \ ... + \ x^T_{n1} \ y_{1n} \ + \ ... \ + \ x^T_{n1} \ y_{1n} \ + \ ... \ + \ x^T_{nn} \ y_{nn} \ = x_{11} \ y^T_{11} + \ ... + \ x_{n1} \ y^T_{1n} \ + \ ... \ + \ x_{n1} \ y^T_{1n} \ + \ ... \ + \ x_{nn} \ y^T_{nn} \ = \ tr(XY^T).$

```
#See what mtcars a
class(mtcars)
```

```
## [1] "data.frame"
```

```
typeof(mtcars)
```

```
## [1] "list"
```

```
mode(mtcars)
```

```
## [1] "list"
```

```
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
```

```
#Make a new matrix.
M <- cbind(mtcars$mpg, mtcars$disp, mtcars$hp, mtcars$drat, mtcars$wt)
apply(M, MARGIN = 2, FUN = mean) #Get column means
```

```
## [1]  20.090625 230.721875 146.687500   3.596563   3.217250
```

```
#Mean centered matrix
Mc <- scale(M, center = T, scale = F) #or, do sweep(M, MARGIN = 2, colMeans(M), FUN = "-")

#check Mc
colMeans(Mc) #all closed enough to zero, so it is mean-centered matrix
```

```
## [1]  4.440892e-16 -1.199041e-14  0.000000e+00 -1.526557e-16  3.469447e-17
```

```
#Another version of mean central
sweep(M, MARGIN = 2, colMeans(M), FUN = "-")
```

```
##             [,1]        [,2]     [,3]       [,4]     [,5]
##  [1,]  0.909375  -70.721875 -36.6875  0.3034375 -0.59725
##  [2,]  0.909375  -70.721875 -36.6875  0.3034375 -0.34225
##  [3,]  2.709375 -122.721875 -53.6875  0.2534375 -0.89725
##  [4,]  1.309375   27.278125 -36.6875 -0.5165625 -0.00225
##  [5,] -1.390625  129.278125  28.3125 -0.4465625  0.22275
##  [6,] -1.990625   -5.721875 -41.6875 -0.8365625  0.24275
##  [7,] -5.790625  129.278125  98.3125 -0.3865625  0.35275
##  [8,]  4.309375  -84.021875 -84.6875  0.0934375 -0.02725
##  [9,]  2.709375  -89.921875 -51.6875  0.3234375 -0.06725
## [10,] -0.890625  -63.121875 -23.6875  0.3234375  0.22275
## [11,] -2.290625  -63.121875 -23.6875  0.3234375  0.22275
## [12,] -3.690625   45.078125  33.3125 -0.5265625  0.85275
## [13,] -2.790625   45.078125  33.3125 -0.5265625  0.51275
## [14,] -4.890625   45.078125  33.3125 -0.5265625  0.56275
## [15,] -9.690625  241.278125  58.3125 -0.6665625  2.03275
## [16,] -9.690625  229.278125  68.3125 -0.5965625  2.20675
## [17,] -5.390625  209.278125  83.3125 -0.3665625  2.12775
## [18,] 12.309375 -152.021875 -80.6875  0.4834375 -1.01725
## [19,] 10.309375 -155.021875 -94.6875  1.3334375 -1.60225
## [20,] 13.809375 -159.621875 -81.6875  0.6234375 -1.38225
## [21,]  1.409375 -110.621875 -49.6875  0.1034375 -0.75225
## [22,] -4.590625   87.278125   3.3125 -0.8365625  0.30275
## [23,] -4.890625   73.278125   3.3125 -0.4465625  0.21775
## [24,] -6.790625  119.278125  98.3125  0.1334375  0.62275
## [25,] -0.890625  169.278125  28.3125 -0.5165625  0.62775
## [26,]  7.209375 -151.721875 -80.6875  0.4834375 -1.28225
## [27,]  5.909375 -110.421875 -55.6875  0.8334375 -1.07725
## [28,] 10.309375 -135.621875 -33.6875  0.1734375 -1.70425
## [29,] -4.290625  120.278125 117.3125  0.6234375 -0.04725
## [30,] -0.390625  -85.721875  28.3125  0.0234375 -0.44725
## [31,] -5.090625   70.278125 188.3125 -0.0565625  0.35275
## [32,]  1.309375 -109.721875 -37.6875  0.5134375 -0.43725
```

```
#Find a column maxima
colmax <- apply(M, 2, max)




#Scale by column maxima
sweep(M, 2, colmax, FUN = "/") #or, do scale(M, F, colmax)
```

```
##             [,1]      [,2]      [,3]      [,4]      [,5]
##  [1,] 0.6194690 0.3389831 0.3283582 0.7910751 0.4830383
##  [2,] 0.6194690 0.3389831 0.3283582 0.7910751 0.5300516
##  [3,] 0.6725664 0.2288136 0.2776119 0.7809331 0.4277286
##  [4,] 0.6312684 0.5466102 0.3283582 0.6247465 0.5927360
##  [5,] 0.5516224 0.7627119 0.5223881 0.6389452 0.6342183
##  [6,] 0.5339233 0.4766949 0.3134328 0.5598377 0.6379056
##  [7,] 0.4218289 0.7627119 0.7313433 0.6511156 0.6581858
##  [8,] 0.7197640 0.3108051 0.1850746 0.7484787 0.5881268
##  [9,] 0.6725664 0.2983051 0.2835821 0.7951318 0.5807522
## [10,] 0.5663717 0.3550847 0.3671642 0.7951318 0.6342183
## [11,] 0.5250737 0.3550847 0.3671642 0.7951318 0.6342183
## [12,] 0.4837758 0.5843220 0.5373134 0.6227181 0.7503687
## [13,] 0.5103245 0.5843220 0.5373134 0.6227181 0.6876844
## [14,] 0.4483776 0.5843220 0.5373134 0.6227181 0.6969027
## [15,] 0.3067847 1.0000000 0.6119403 0.5943205 0.9679204
## [16,] 0.3067847 0.9745763 0.6417910 0.6085193 1.0000000
## [17,] 0.4336283 0.9322034 0.6865672 0.6551724 0.9854351
## [18,] 0.9557522 0.1667373 0.1970149 0.8275862 0.4056047
## [19,] 0.8967552 0.1603814 0.1552239 1.0000000 0.2977507
## [20,] 1.0000000 0.1506356 0.1940299 0.8559838 0.3383112
## [21,] 0.6342183 0.2544492 0.2895522 0.7505071 0.4544617
## [22,] 0.4572271 0.6737288 0.4477612 0.5598377 0.6489676
## [23,] 0.4483776 0.6440678 0.4477612 0.6389452 0.6332965
## [24,] 0.3923304 0.7415254 0.7313433 0.7565923 0.7079646
## [25,] 0.5663717 0.8474576 0.5223881 0.6247465 0.7088864
## [26,] 0.8053097 0.1673729 0.1970149 0.8275862 0.3567478
## [27,] 0.7669617 0.2548729 0.2716418 0.8985801 0.3945428
## [28,] 0.8967552 0.2014831 0.3373134 0.7647059 0.2789454
## [29,] 0.4660767 0.7436441 0.7880597 0.8559838 0.5844395
## [30,] 0.5811209 0.3072034 0.5223881 0.7342799 0.5106932
## [31,] 0.4424779 0.6377119 1.0000000 0.7180527 0.6581858
## [32,] 0.6312684 0.2563559 0.3253731 0.8336714 0.5125369
```

```
#scale to have min = 0 and max = 1
scaling <- function(x){
  (x - min(x))/diff(range(x))
}

apply(M, 2, scaling)
```

```
##            [,1]       [,2]       [,3]       [,4]       [,5]
##  [1,] 0.4510638 0.22175106 0.20494700 0.52534562 0.28304781
##  [2,] 0.4510638 0.22175106 0.20494700 0.52534562 0.34824853
##  [3,] 0.5276596 0.09204290 0.14487633 0.50230415 0.20634109
##  [4,] 0.4680851 0.46620105 0.20494700 0.14746544 0.43518282
##  [5,] 0.3531915 0.72062859 0.43462898 0.17972350 0.49271286
##  [6,] 0.3276596 0.38388626 0.18727915 0.00000000 0.49782664
##  [7,] 0.1659574 0.72062859 0.68197880 0.20737327 0.52595244
##  [8,] 0.5957447 0.18857570 0.03533569 0.42857143 0.42879059
##  [9,] 0.5276596 0.17385882 0.15194346 0.53456221 0.41856303
## [10,] 0.3744681 0.24070841 0.25088339 0.53456221 0.49271286
## [11,] 0.3148936 0.24070841 0.25088339 0.53456221 0.49271286
## [12,] 0.2553191 0.51060115 0.45229682 0.14285714 0.65379698
## [13,] 0.2936170 0.51060115 0.45229682 0.14285714 0.56686269
## [14,] 0.2042553 0.51060115 0.45229682 0.14285714 0.57964715
## [15,] 0.0000000 1.00000000 0.54063604 0.07834101 0.95551010
## [16,] 0.0000000 0.97006735 0.57597173 0.11059908 1.00000000
## [17,] 0.1829787 0.92017960 0.62897527 0.21658986 0.97980056
## [18,] 0.9361702 0.01895735 0.04946996 0.60829493 0.17565840
## [19,] 0.8510638 0.01147418 0.00000000 1.00000000 0.02608029
## [20,] 1.0000000 0.00000000 0.04593640 0.67281106 0.08233188
## [21,] 0.4723404 0.12222499 0.15901060 0.43317972 0.24341601
## [22,] 0.2170213 0.61586431 0.34628975 0.00000000 0.51316799
## [23,] 0.2042553 0.58094288 0.34628975 0.17972350 0.49143442
## [24,] 0.1234043 0.69568471 0.68197880 0.44700461 0.59498849
## [25,] 0.3744681 0.82040409 0.43462898 0.14746544 0.59626694
## [26,] 0.7191489 0.01970566 0.04946996 0.60829493 0.10790079
## [27,] 0.6638298 0.12272387 0.13780919 0.76958525 0.16031705
## [28,] 0.8510638 0.05986530 0.21554770 0.46543779 0.00000000
## [29,] 0.2297872 0.69817910 0.74911661 0.67281106 0.42367681
## [30,] 0.3957447 0.18433525 0.43462898 0.39631336 0.32140118
## [31,] 0.1957447 0.57345972 1.00000000 0.35944700 0.52595244
## [32,] 0.4680851 0.12446994 0.20141343 0.62211982 0.32395807
```

```
#There is a way to get covariance in slide 5. Divided by n-1 as it is a sample.
#First
cov_m <- (1/(nrow(M) - 1)) * t(Mc) %*% Mc

cov_m
```

```
##             [,1]        [,2]       [,3]        [,4]        [,5]
## [1,]   36.324103  -633.09721 -320.73206   2.1950635  -5.1166847
## [2,] -633.097208 15360.79983 6721.15867 -47.0640192 107.6842040
## [3,] -320.732056  6721.15867 4700.86694 -16.4511089  44.1926613
## [4,]    2.195064   -47.06402  -16.45111   0.2858814  -0.3727207
## [5,]   -5.116685   107.68420   44.19266  -0.3727207   0.9573790
```

```
#Second
cov(M)
```

```
##             [,1]        [,2]       [,3]        [,4]        [,5]
## [1,]   36.324103  -633.09721 -320.73206   2.1950635  -5.1166847
## [2,] -633.097208 15360.79983 6721.15867 -47.0640192 107.6842040
## [3,] -320.732056  6721.15867 4700.86694 -16.4511089  44.1926613
## [4,]    2.195064   -47.06402  -16.45111   0.2858814  -0.3727207
## [5,]   -5.116685   107.68420   44.19266  -0.3727207   0.9573790
```

```
#Although identical function gives "FALSE" but there is only small difference in a decimal point....




#There is a way to get correlation in slide 5.
#First -> as Mc are mean-centered. Try to use the way on pg 31 of slide 3, but it is not working...
#cov_m / (t(Mc) %*% Mc)^2
#(t(Mc) %*% Mc) / sqrt((t(Mc) %*% Mc)) %*% sqrt((Mc %*% t(Mc)))




#Second
scaling2 <- sweep(Mc, 2, apply(Mc, 2, FUN = sd), FUN = "/")
cor_m2 <- (1/(nrow(M) - 1)) * t(scaling2) %*% scaling2

cor_m2
```

```
##            [,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
```

```
#Third
scaling <- scale(M, T, T)
cor_m <- (1/(nrow(M) - 1)) * t(scaling) %*% scaling

cor_m
```

```
##            [,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
```

```
#Fourth
cor(M)
```

```
##            [,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
```

```
#Dummify function
dummify <-function(x, all = T){
  if(all == T){
    new_matrix<- matrix(1, length(x), length(levels(x))) #create an arbitrary matrix that is made of 1
    colnames(new_matrix) <- c(levels(x))
    rownames(new_matrix) <- x
    #assign 0 binary if row and column numbers are not the same
    for (i in 1:length(x)){
      for (j in 1:length(levels(x))){
        if(rownames(new_matrix)[i] != colnames(new_matrix)[j]){
          new_matrix[i,j] <- 0
        }
      }
    }

    return(new_matrix)
  }else{
    new_matrix<- matrix(1, length(x), length(levels(x))) #create an arbitrary matrix that is made of 1
    colnames(new_matrix) <- c(levels(x))
    rownames(new_matrix) <- x
    #assign 0 binary if row and column numbers are not the same
    for (i in 1:length(x)){
      for (j in 1:length(levels(x))){
        if(rownames(new_matrix)[i] != colnames(new_matrix)[j]){
          new_matrix[i,j] <- 0
        }
      }
    }
    # IF all=F, then we only extract k-1 indicators, by taking out one column randomly.
    random_number <- sample(1:3,1)
    return(new_matrix[ ,-(random_number)])
  }
}

#other way
dummify2 <-function(x, all = T){
  x <- as.factor(x)
  #this function compares the set (4, 6, 8) with each element of x.
  sapply(levels(x), function(data_element){as.integer(x == data_element)})
}

cyl <- factor(mtcars$cyl)
CYL1 <- dummify(cyl, all = T)
CYL2 <- dummify(cyl, all = F)

#Crosstable function
```

```
crosstable <-function(x, y){
  dumc <- dummify(x, all = T)
  dumy <- dummify(y, all = T)
  return(t(dumc) %*% dumy)
}

gear <- factor(mtcars$gear)
xtb <- crosstable(cyl, gear)


CYL1
```

```
##   4 6 8
## 6 0 1 0
## 6 0 1 0
## 4 1 0 0
## 6 0 1 0
## 8 0 0 1
## 6 0 1 0
## 8 0 0 1
## 4 1 0 0
## 4 1 0 0
## 6 0 1 0
## 6 0 1 0
## 8 0 0 1
## 8 0 0 1
## 8 0 0 1
## 8 0 0 1
## 8 0 0 1
## 8 0 0 1
## 4 1 0 0
## 4 1 0 0
## 4 1 0 0
## 4 1 0 0
## 8 0 0 1
## 8 0 0 1
## 8 0 0 1
## 8 0 0 1
## 4 1 0 0
## 4 1 0 0
## 4 1 0 0
## 8 0 0 1
## 6 0 1 0
## 8 0 0 1
## 4 1 0 0
```

```
dummify2(cyl, all = T)
```

```
##       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
##  [7,] 0 0 1
##  [8,] 1 0 0
##  [9,] 1 0 0
## [10,] 0 1 0
## [11,] 0 1 0
## [12,] 0 0 1
## [13,] 0 0 1
## [14,] 0 0 1
## [15,] 0 0 1
## [16,] 0 0 1
## [17,] 0 0 1
## [18,] 1 0 0
## [19,] 1 0 0
## [20,] 1 0 0
## [21,] 1 0 0
## [22,] 0 0 1
## [23,] 0 0 1
## [24,] 0 0 1
## [25,] 0 0 1
## [26,] 1 0 0
## [27,] 1 0 0
## [28,] 1 0 0
## [29,] 0 0 1
## [30,] 0 1 0
## [31,] 0 0 1
## [32,] 1 0 0
```

```
xtb
```

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