R Code to Complement Math Review Packet

Properties of Logarithms

Practice Problems.

1. Solve the following:

a. $log_e(e^x) = x$

```
# If x = 3
log(exp(3))
```

```
## [1] 3
```

b. $log_{10}(100) = 2$

```
log10(100)
```

```
## [1] 2
```

c. $log_{10}(\frac{1}{10}) = -1$

```
log10(1/10)
```

```
## [1] -1
```

d. $log_{10}(0) =$ No solution

```
log10(0)
```

```
## [1] -Inf
```

3c. $log_{10}(5) + log_{10}(2) = 1$

```
log10(5) + log10(2)
```

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

Matrix Algebra

Type ?matrix in the **Condole** or matrix in **Help** to look at the inputs of the matrix function.

If we want to store the matrix we need to call it "A", for example and store the matrix.

For example $\mathbf{Z} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ $\mathbf{Y} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$

```
#Sorting by row
Z <- matrix(data = c(1,2,3,4,5,6,7,8,9),nrow = 3, ncol = 3, byrow = TRUE)
print(Z)
```

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

```r
#Sorting by column
Y <- matrix(data = (1:9), nrow = 3, ncol = 3, byrow = FALSE)
print(Y)
```

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

Definitions

The transpose of an $m \times n$ matrix, A is the $n \times m$ matrix (denoted A^T) such that every element a_{ij} in matrix A is moved to row j and column i. For example, if:

$$A = \begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 7 \end{bmatrix}$$

then,

$$A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 5 & 7 \end{bmatrix}$$

```r
# Matrix A
A <- matrix(data = c(1,2,5,3,4,7), nrow = 2, ncol = 3, byrow = TRUE)

print(A)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    5
## [2,]    3    4    7
```

```r
# A transpose
print(t(A))
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
## [3,]    5    7
```

Identity Matrix

```r
# Identity Matrix: use diag() to create an identity matrix
# I = 4x4 identity matrix
I <- diag(4)
print(I)
```

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

```r
class(I)
```

```
## [1] "matrix"
```

Diagonal

```r
X <- matrix(data = c(1,0,0,0,4,0,0,0,5), nrow = 3, byrow = TRUE)
print(X)
```

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

```r
# To obtain the obtain the values of the diagonal of an n x n matrix
diag(X)
```

```
## [1] 1 4 5
```

Upper and Lower Triangular Matrices

```r
# Create an empty matrix
V <- W <- matrix(data = 0, nrow = 3, ncol = 3, byrow = TRUE)
print(W)
```

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

```r
# Use upper.tri() or lower.tri() to change the elements of the upper
# triangular matrix or lower triangular matrix, respectively.

# Upper Triangular
W[upper.tri(W, diag = TRUE)] <- c(1:6)
print(W)
```

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

```r
# Lower Triangular
V[lower.tri(V, diag = TRUE)] <- c(1:6)
print(V)
```

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

Inverse of a matrix

```r
U <- matrix(data = c(1:4), nrow = 2, byrow = TRUE)

# Use solve() to obtain the inversie of a matrix
inU <- solve(U)

# Confirm you get an identity matrix
round(U %*% inU, 2)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

```r
round(inU %*% U, 2)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

Determinant of a matrix

```r
# Use det() to obtain the determinant of a matrix
det(U)
```

```
## [1] -2
```

Operations with matrices

Adding matices

```r
#
```

Practice Problems

1. Solve the following:

a. $\begin{bmatrix} 2 & 4 & 2 \\ 1 & 3 & 0 \\ 1 & 6 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 5 & 0 \\ -2 & -3 & 0 \\ 1 & 9 & 5 \end{bmatrix} = \begin{bmatrix} 3 & 9 & 2 \\ -1 & 0 & 0 \\ 2 & 15 & 7 \end{bmatrix}$

```r
M1 <- matrix(data = c(2,4,2,1,3,0,1,6,2),
             nrow = 3, ncol = 3, byrow = TRUE)

M2 <- matrix(data = c(1,5,0,-2,-3,0,1,9,5),
             nrow = 3, ncol = 3, byrow = TRUE)

M1 + M2
```

```
##      [,1] [,2] [,3]
## [1,]    3    9    2
## [2,]   -1    0    0
## [3,]    2   15    7
```

b. $\begin{bmatrix} 2 & 1 \\ -2 & 2 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 5 & 2 & -1 \\ 3 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 13 & 8 & 0 \\ -4 & 4 & 6 \\ 26 & 16 & 0 \end{bmatrix}$

```r
M3 <- matrix(data = c(2,-2,4,1,2,2), nrow = 3, ncol = 2, byrow = FALSE)

M4 <- matrix(data = c(5,3,2,4,-1,2), nrow = 2, ncol = 3, byrow = FALSE)

M3 %*% M4
```

```
##      [,1] [,2] [,3]
## [1,]   13    8    0
## [2,]   -4    4    6
## [3,]   26   16    0
```

c. Let $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 4 & 0 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 4 & 4 \\ 1 & 2 \\ 7 & 0 \end{bmatrix}$ $\mathbf{A}^T\mathbf{B} = \begin{bmatrix} 35 & 10 \\ 13 & 18 \end{bmatrix}$

```
A <- matrix(data = c(1,2,3,5,4,0), nrow = 3, ncol = 2, byrow = TRUE)

B <- matrix(data = c(4,1,7,4,2,0),nrow = 3, ncol=2, byrow = FALSE)

# Finding A transpose
tA <- t(A)
print(tA)
```

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

```
tA %*% B
```

```
##      [,1] [,2]
## [1,]   35   10
## [2,]   13   18
```

d. Using the same matrices as in part c, $\mathbf{B}^T\mathbf{A} = \begin{bmatrix} 35 & 13 \\ 10 & 18 \end{bmatrix}$

```
tB <- t(B)

tB %*% A
```

```
##      [,1] [,2]
## [1,]   35   13
## [2,]   10   18
```

Show that $\mathbf{A}$ and $\mathbf{B}$ are inverses: $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0.5 & 1 \\ 1 & 0 & -1 \\ 0 & -0.5 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

```
A <- matrix(data = c(1,2,1,2,2,0,1,1,1), nrow = 3, ncol = 3, byrow = TRUE)

#Use the solve() to find the inverse of a matrix
solve(A)
```

```
##      [,1] [,2] [,3]
## [1,]   -1  0.5    1
## [2,]    1  0.0   -1
## [3,]    0 -0.5    1
```

```
B <- matrix(data = c(-1, 0.5, 1, 1, 0, -1, 0, -0.5, 1), nrow = 3, ncol = 3, byrow = TRUE)

#Use the solve() to find the inverse of a matrix
solve(B)
```

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

```
A %*% B
```

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

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

3. Suppose that A is a 4x3 matrix and B is a $3x8$ matrix.

```
A <- matrix (data= (1:12), nrow = 4, ncol = 3, byrow = FALSE)
B <- matrix (data = (1:24), nrow = 3, ncol = 8, byrow = TRUE)
```

a. **AB** exists and is 4x8 matrix.

```
A %*% B
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]  199  214  229  244  259  274  289  304
## [2,]  226  244  262  280  298  316  334  352
## [3,]  253  274  295  316  337  358  379  400
## [4,]  280  304  328  352  376  400  424  448
```

```
dim(A%*%B)
```

```
## [1] 4 8
```

b. **BA** does not exist.

```
# The following produce errors
# B %*% A
# dim(B %*% A)
```

4. The determinant of $det \begin{bmatrix} 1 & -2 \\ 4 & 3 \end{bmatrix} = 11$

```
M5 <- matrix(data = c(1, -2, 4, 3), nrow = 2, ncol = 2, byrow = TRUE)

det(M5)
```

```
## [1] 11
```