Class_slide2 (More about matrix in R)

Dr. Niladri Chakraborty

2024-02-23

Some special matrices in R.

- Diaginal matrix

Create a diagonal matrix with diag().

```
# Create a diagonal matrix
diag_matrix <- diag(1:3)
print(diag_matrix)
```

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

- Identity matrix

An identity matrix can be created using diag() with a single numeric argument.

```
# Create an identity matrix of order 3
identity_matrix <- diag(3)
print(identity_matrix)
```

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

Matrix functions

- Determinant

Calculate the determinant with det().

```
# Determinant of m
m <- array(c(3.7,4.2, 2.01, 4.77, 9.8, 3.6, 1.5, 2.7, 8.7), dim = c(3,3))
det_m <- det(m)
print(det_m)
```

```
## [1] 124.222
```

- Inverse

Find the inverse with solve().

```r
# Inverse of m (assuming it's invertible)
inverse_m <- solve(m)
print(inverse_m)
```

```
##             [,1]        [,2]        [,3]
## [1,]  0.60810489 -0.29060072 -0.01465924
## [2,] -0.25046290  0.23486180 -0.02970489
## [3,] -0.03685338 -0.03004541  0.13062100
```

- Generalized inverse

Calculate the generalized inverse of a matrix *when the determinant of the matrix is 0*. You need to MASS package to calculate the generalized inverse with ginv().

```r
library(MASS)
# Inverse of m (assuming it's invertible)
ginv_m <- ginv(m)
print(ginv_m)
```

```
##             [,1]        [,2]        [,3]
## [1,]  0.60810489 -0.29060072 -0.01465924
## [2,] -0.25046290  0.23486180 -0.02970489
## [3,] -0.03685338 -0.03004541  0.13062100
```

- Eigenvalues and Eigenvectors

Compute eigenvalues and eigenvectors with eigen().

```r
# Eigenvalues and eigenvectors
eigen_m <- eigen(m)
print(eigen_m$values)
```

```
## [1] 14.400768  6.464950  1.334282
```

```r
print(eigen_m$vectors)
```

```
##            [,1]       [,2]        [,3]
## [1,] -0.3970154 -0.2366891 -0.89941892
## [2,] -0.7063542 -0.4136537  0.43588517
## [3,] -0.5860397  0.8791296  0.03239948
```

Row names and column names in R

You can provide names to the rows and columns of a matrix using the rownames() and colnames() functions, respectively. These functions allow you to assign and retrieve row and column names of a matrix.

- Assigning row and column names

```r
# Create a matrix
A <- matrix(1:9, nrow = 3, byrow = TRUE)

# Assign row names
```

```r
rownames(A) <- c("Row1", "Row2", "Row3")

# Assign column names
colnames(A) <- c("Col1", "Col2", "Col3")

# Print the matrix with row and column names
print(A)
```

```
##      Col1 Col2 Col3
## Row1    1    2    3
## Row2    4    5    6
## Row3    7    8    9
```

- Retrieving row and column names

```r
# Retrieve row names
row_names <- rownames(A)
print(row_names)
```

```
## [1] "Row1" "Row2" "Row3"
```

```r
# Retrieve column names
col_names <- colnames(A)
print(col_names)
```

```
## [1] "Col1" "Col2" "Col3"
```

- Another example:

Suppose you have a matrix representing data from three different experiments (rows) across three different conditions (columns), and you want to label these appropriately:

```r
# Create a sample matrix
data_matrix <- matrix(c(20, 35, 40, 50, 60, 75, 65, 85, 95), nrow = 3, byrow
= TRUE)

# Assign meaningful row and column names
rownames(data_matrix) <- c("Experiment1", "Experiment2", "Experiment3")
colnames(data_matrix) <- c("Condition1", "Condition2", "Condition3")

# Display the matrix
print(data_matrix)
```

```
##             Condition1 Condition2 Condition3
## Experiment1         20         35         40
## Experiment2         50         60         75
## Experiment3         65         85         95
```

- Notes

Assigning names to rows and columns can significantly improve the readability of your data, making it easier to reference specific elements and understand the matrix's structure.