Matrix Operations and Multivariate Linear Regression

Yunting Chiu

2021-02-27

R Lab 9: Matrix Operations and Multivariate Linear Regression

Define a matrix by entering its elements manually.

```
B <-  matrix(c(1,2,3,4,5,6,7,8,9),3,3)
B
```

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

This is an element-wise operation

```
B^2
```

```
##      [,1] [,2] [,3]
## [1,]    1   16   49
## [2,]    4   25   64
## [3,]    9   36   81
```

This is matrix multiplication, B^2 = B*B

```
B %*% B
```

```
##      [,1] [,2] [,3]
## [1,]   30   66  102
## [2,]   36   81  126
## [3,]   42   96  150
```

Transposed matrix

```
t(B)
```

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

Joining two matrices side by side (as columns)

```
cbind(B, B)
```

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

Joining two matrices below each other (as rows)

```r
rbind(B, B)
```

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

Sub-matrix, a part of matrix B

```r
B[1:3, 1:2]
```

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

Inverting matrices is available in package "matlib"

```r
# install.packages("matlib")
library(matlib)
```

```
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display
```

```
## Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.
```

```r
B
```

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

```r
inv(B)
```

```
## Error in Inverse(X, tol = sqrt(.Machine$double.eps), ...): X is numerically singular
```

This means there is a linear dependence among columns (and among rows) of matrix B. Such matrices are not invertible, and they have a determinant equal det(B)=0

make determinant equal det(B)=0

```r
det(B)
```

```
## [1] 0
```

```r
B[1,1]=100
B
```

```
##      [,1] [,2] [,3]
## [1,]  100    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

We changed the matrix by adding a "ridge", and now the inverse B^-1 exists

```r
inv(B)
```

```
##             [,1]        [,2]       [,3]
## [1,]  0.01010101 -0.02020202 0.01010101
## [2,] -0.02020202 -2.95959596 2.64646465
```

```
## [3,]  0.01010101  1.97979798 -1.65656566
```

Data practice - Multivariate Linear Regression

Define a matrix from the "mtcars" data set and build a regression model that **predicts miles per gallon** based on the number of cylinders, horsepower, axes ratio, weight, and acceleration time.

```r
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
```

Extract X-matrix of predictors

```r
X = data.matrix(mtcars[,c(2,4:7)])
head(X)
```

```
##                   cyl  hp drat    wt  qsec
## Mazda RX4           6 110 3.90 2.620 16.46
## Mazda RX4 Wag       6 110 3.90 2.875 17.02
## Datsun 710          4  93 3.85 2.320 18.61
## Hornet 4 Drive      6 110 3.08 3.215 19.44
## Hornet Sportabout   8 175 3.15 3.440 17.02
## Valiant             6 105 2.76 3.460 20.22
```

Find the vector of responses of Y

```r
Y = data.matrix(mtcars[ ,1])
head(Y)
```

```
##      [,1]
## [1,] 21.0
## [2,] 21.0
## [3,] 22.8
## [4,] 21.4
## [5,] 18.7
## [6,] 18.1
```

We also need a vector of 1s to include the intercept

```r
n = length(Y)
one = matrix(1,n,1) #row 1, col n, values 1
n
```

```
## [1] 32
```

```r
one
```

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

```
##  [6,]    1
##  [7,]    1
##  [8,]    1
##  [9,]    1
## [10,]    1
## [11,]    1
## [12,]    1
## [13,]    1
## [14,]    1
## [15,]    1
## [16,]    1
## [17,]    1
## [18,]    1
## [19,]    1
## [20,]    1
## [21,]    1
## [22,]    1
## [23,]    1
## [24,]    1
## [25,]    1
## [26,]    1
## [27,]    1
## [28,]    1
## [29,]    1
## [30,]    1
## [31,]    1
## [32,]    1
```

```r
X = cbind(one,X)
X
```

```
##                       cyl  hp drat    wt  qsec
## Mazda RX4           1   6 110 3.90 2.620 16.46
## Mazda RX4 Wag       1   6 110 3.90 2.875 17.02
## Datsun 710          1   4  93 3.85 2.320 18.61
## Hornet 4 Drive      1   6 110 3.08 3.215 19.44
## Hornet Sportabout   1   8 175 3.15 3.440 17.02
## Valiant             1   6 105 2.76 3.460 20.22
## Duster 360          1   8 245 3.21 3.570 15.84
## Merc 240D           1   4  62 3.69 3.190 20.00
## Merc 230            1   4  95 3.92 3.150 22.90
## Merc 280            1   6 123 3.92 3.440 18.30
## Merc 280C           1   6 123 3.92 3.440 18.90
## Merc 450SE          1   8 180 3.07 4.070 17.40
## Merc 450SL          1   8 180 3.07 3.730 17.60
## Merc 450SLC         1   8 180 3.07 3.780 18.00
## Cadillac Fleetwood  1   8 205 2.93 5.250 17.98
## Lincoln Continental 1   8 215 3.00 5.424 17.82
## Chrysler Imperial   1   8 230 3.23 5.345 17.42
## Fiat 128            1   4  66 4.08 2.200 19.47
## Honda Civic         1   4  52 4.93 1.615 18.52
## Toyota Corolla      1   4  65 4.22 1.835 19.90
## Toyota Corona       1   4  97 3.70 2.465 20.01
## Dodge Challenger    1   8 150 2.76 3.520 16.87
## AMC Javelin         1   8 150 3.15 3.435 17.30
```

```
## Camaro Z28          1   8 245 3.73 3.840 15.41
## Pontiac Firebird    1   8 175 3.08 3.845 17.05
## Fiat X1-9           1   4  66 4.08 1.935 18.90
## Porsche 914-2       1   4  91 4.43 2.140 16.70
## Lotus Europa        1   4 113 3.77 1.513 16.90
## Ford Pantera L      1   8 264 4.22 3.170 14.50
## Ferrari Dino        1   6 175 3.62 2.770 15.50
## Maserati Bora       1   8 335 3.54 3.570 14.60
## Volvo 142E          1   4 109 4.11 2.780 18.60
```

This is matrix X'X

```
t(X) %*% X
```

```
##                     cyl        hp       drat         wt       qsec
##         32.000   198.000   4694.00   115.0900   102.9520   571.160
## cyl    198.000  1324.000  32204.00   691.4000   679.4040  3475.560
## hp    4694.000 32204.000 834278.00 16372.2800 16471.7440 81092.160
## drat   115.090   691.400  16372.28   422.7907   358.7190  2056.914
## wt     102.952   679.404  16471.74   358.7190   360.9011  1828.095
## qsec   571.160  3475.560  81092.16  2056.9140  1828.0946 10293.480
```

Slope $\beta 1$ = (X'X)-1X'Y

```
slope = inv(t(X) %*% X) %*% t(X) %*% Y
slope
```

```
##             [,1]
## [1,] 25.94553598
## [2,] -0.48955421
## [3,] -0.01505188
## [4,]  1.13092435
## [5,] -3.38272539
## [6,]  0.34985343
```

Another method to get thesame slope by using built-in function `lm`

```
attach(mtcars)
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

```
recheckReg <- lm(mpg ~ cyl + hp + drat + wt + qsec)
recheckReg
```

```
##
## Call:
## lm(formula = mpg ~ cyl + hp + drat + wt + qsec)
##
## Coefficients:
## (Intercept)          cyl           hp         drat           wt         qsec
##    25.94521     -0.48967     -0.01539      1.13077     -3.38279      0.35011
```

Our estimated regression equation is `mpg = 25.95 - 0.49 cyl - 0.015 hp + 1.13 drat - 3.38 wt + 0.35 qsec + e`