Class Prep 7 | 3.1.2 - 3.2.1

Riley Primeau

2022-10-10

Section 3.1.2 Elementary Row Operations

```
scalerow = function(m, row, k){
  m[row,] = m[row,] * k
  return(m)
}

swaprows = function(m, row1, row2){
  row.tmp = m[row1,]
  m[row1,] = m[row2,]
  m[row2,] = row.tmp

  return(m)
}

A = matrix(1:15, 5)
scalerow(A, 2, 10)
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   11
## [2,]   20   70  120
## [3,]    3    8   13
## [4,]    4    9   14
## [5,]    5   10   15
```

```
replacerow = function(m, row1, row2, k){
  m[row2,] = m[row2,] + m[row1,] * k
  return(m)
}

swaprows(A, 1, 4)
```

```
##      [,1] [,2] [,3]
## [1,]    4    9   14
## [2,]    2    7   12
## [3,]    3    8   13
## [4,]    1    6   11
## [5,]    5   10   15
```

```r
replacerow(A, 1, 3, -3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   11
## [2,]    2    7   12
## [3,]    0  -10  -20
## [4,]    4    9   14
## [5,]    5   10   15
```

Section 3.2.1 Row Echelon Form

```
refmatrix = function(m){
  count.rows = nrow(m)
  count.cols = ncol(m)
  piv = 1

  for(row.curr in 1:count.rows){
    if(piv <= count.cols){
      i = row.curr
      while(m[i, piv] == 0 && i < count.rows){
        i = i + 1
        if(i > count.rows){
          i = row.curr
          piv = piv + 1
          if(piv > count.cols)
            return(m)
        }
      }
      if (i != row.curr)
        m = swaprows(m, i, rows.curr)
      for(j in row.curr:count.rows)
        if(j != row.curr){
          k = m[j, piv] / m[row.curr, piv]
          m = replacerow(m, row.curr, j, -k)
        }
      piv = piv + 1
    }
  }
  return(m)
}
```

```
(A = matrix(c(5, 5, 5, 8, 2, 2, 6, 5, 4), 3))
```

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

```
refmatrix(A)
```

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

```
(A = matrix(c(2, 4, 2, 4, 9, 4, 3, 6, 7, 7, 3, 9), 3))
```

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

```
refmatrix(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    4    3    7
## [2,]    0    1    0  -11
## [3,]    0    0    4    2
```

```
(A = matrix(c(2, 8, 5, 5, 1, 2, 3, 8, 4), 3))
```

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

```
refmatrix(A)
```

```
##      [,1]          [,2]      [,3]
## [1,]    2  5.000000e+00  3.000000
## [2,]    0 -1.900000e+01 -4.000000
## [3,]    0  1.776357e-15 -1.289474
```

```
rrefmatrix = function(m){
  count.rows = nrow(m)
  count.cols = ncol(m)
  piv = 1

  for(row.curr in 1:count.rows){
    if(piv <= count.cols){
      i = row.curr
      while(m[i, piv] == 0 && i < count.rows){
        i = i + 1
        if(i > count.rows){
          i = row.curr
          piv = piv + 1
          if(piv > count.cols)
            return(m)
        }
      }
      if (i != row.curr)
        m = swaprows(m, i, rows.curr)
      piv.val = m[row.curr, piv]
      m = scalerow(m, row.curr, 1 / piv.val)
      for(j in 1:count.rows)
        if(j != row.curr){
          k = m[j, piv] / m[row.curr, piv]
          m = replacerow(m, row.curr, j, -k)
        }
      piv = piv + 1
    }
  }
  return(m)
}

A = matrix(c(5, 5, 5, 8, 2, 2, 6, 5, 4), 3)
rrefmatrix(A)
```

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

```
A = matrix(c(2, 4, 2, 4, 9, 4, 3, 6, 7, 7, 3, 9), 3)
rrefmatrix(A)
```

```
##      [,1] [,2] [,3]   [,4]
## [1,]    1    0    0  24.75
## [2,]    0    1    0 -11.00
## [3,]    0    0    1   0.50
```

```
A = matrix(c(2, 8, 5, 5, 1, 2, 3, 8, 4), 3)
rrefmatrix(A)
```

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

```
(A = matrix(c(2, 3, 1, 1, 2, -5, -1, -2, 4), 3))
```

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

```
(b = c(1, 1, 3))
```

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

```
rrefmatrix(cbind(A, b))
```

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

```
solvematrix = function(A, b){
  m = cbind(A, b)
  m = rrefmatrix(m)
  x = m[, ncol(m)]

  return(x)
}

solvematrix(A, b)
```

```
## [1] 1 2 3
```

```
solve(A, b)
```

```
## [1] 1 2 3
```