Optimization HW1

Alyson Brown

1/30/2019

*Note: I used ’ to mean inverse throughout this homework…I couldn’t figure out how to get a superscript of -1.

Q1

Let $B = A'$

So $B^T = (A')^T$

We know that $AB = I$

So then $(AB)^T = I^T$ and $A^T B^T = I$

From this equation we can say that $B^T = (A^T)'$

And so $B^T = (A')^T = (A^T)'$

Q2

```
A = matrix(c(1, 1, 1, 1, .45, -.55, 0, 0, 0, 1, 0, 0, .14, .2, .2, .1), nrow = 4, nco
l = 4, byrow = TRUE)
b =c(250000000,0,62500000,37500000)
A_inv = solve(A,b)
print(A_inv)
```

```
## [1] 76388889 62500000 31944444 79166667
```

```
cat("Based on these parameters, the bank should lend $", A_inv[1], " in First Mortgag
e loans, $", A_inv[2], " in Second Mortgage loans, $", A_inv[3], " in Home Improvemen
t loans, and $", A_inv[4], " in Personal Overdraft loans.")
```

```
## Based on these parameters, the bank should lend $ 76388889  in First Mortgage loan
s, $ 62500000  in Second Mortgage loans, $ 31944444  in Home Improvement loans, and $
79166667  in Personal Overdraft loans.
```

Q3

Let x_1 = number of units manufactured of product variant 1

Let x_2 = number of units manufactured of product variant 2

Let x_3 = number of units manufactured of product variant 3

Let x_4 = number of units manufactured of product variant 4

MAX Z = $1.5x_1 + 2.5x_2 + 3x_3 + 4.5x_4$

ST:

1. $2x_1 + 4x_2 + 3x_3 + 7x_4 <= 100,000$
2. $3x_1 + 2x_2 + 3x_3 + 4x_4 <= 50,000$
3. $2x_1 + 3x_2 + 2x_3 + 5x_4 <= 60,000$
4. $x_1, x_2, x_3, x_4 >= 0$

Q4

```
##Part A
#L = matrix(c(0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0), nrow = 4, ncol = 4, byrow = TRUE)
L = matrix(0, nrow = 20, ncol = 20)

# Loop over my_matrix
for(row in 1:nrow(L)) {
  for(col in 1:ncol(L)) {
    if (col<row) {
      L[row, col] = col/row
    } else {
      L[row, col] = row/col
    }
    #print(L[row, col])
  }
}

print(L)
```

```
##              [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
##  [1,] 1.00000000 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667
##  [2,] 0.50000000 1.0000000 0.6666667 0.5000000 0.4000000 0.3333333
##  [3,] 0.33333333 0.6666667 1.0000000 0.7500000 0.6000000 0.5000000
##  [4,] 0.25000000 0.5000000 0.7500000 1.0000000 0.8000000 0.6666667
##  [5,] 0.20000000 0.4000000 0.6000000 0.8000000 1.0000000 0.8333333
##  [6,] 0.16666667 0.3333333 0.5000000 0.6666667 0.8333333 1.0000000
##  [7,] 0.14285714 0.2857143 0.4285714 0.5714286 0.7142857 0.8571429
##  [8,] 0.12500000 0.2500000 0.3750000 0.5000000 0.6250000 0.7500000
##  [9,] 0.11111111 0.2222222 0.3333333 0.4444444 0.5555556 0.6666667
## [10,] 0.10000000 0.2000000 0.3000000 0.4000000 0.5000000 0.6000000
## [11,] 0.09090909 0.1818182 0.2727273 0.3636364 0.4545455 0.5454545
## [12,] 0.08333333 0.1666667 0.2500000 0.3333333 0.4166667 0.5000000
## [13,] 0.07692308 0.1538462 0.2307692 0.3076923 0.3846154 0.4615385
## [14,] 0.07142857 0.1428571 0.2142857 0.2857143 0.3571429 0.4285714
```

```
## [15,] 0.06666667 0.1333333 0.2000000 0.2666667 0.3333333 0.4000000
## [16,] 0.06250000 0.1250000 0.1875000 0.2500000 0.3125000 0.3750000
## [17,] 0.05882353 0.1176471 0.1764706 0.2352941 0.2941176 0.3529412
## [18,] 0.05555556 0.1111111 0.1666667 0.2222222 0.2777778 0.3333333
## [19,] 0.05263158 0.1052632 0.1578947 0.2105263 0.2631579 0.3157895
## [20,] 0.05000000 0.1000000 0.1500000 0.2000000 0.2500000 0.3000000
##            [,7]      [,8]      [,9]     [,10]      [,11]      [,12]
##  [1,] 0.1428571 0.1250000 0.1111111 0.1000000 0.09090909 0.08333333
##  [2,] 0.2857143 0.2500000 0.2222222 0.2000000 0.18181818 0.16666667
##  [3,] 0.4285714 0.3750000 0.3333333 0.3000000 0.27272727 0.25000000
##  [4,] 0.5714286 0.5000000 0.4444444 0.4000000 0.36363636 0.33333333
##  [5,] 0.7142857 0.6250000 0.5555556 0.5000000 0.45454545 0.41666667
##  [6,] 0.8571429 0.7500000 0.6666667 0.6000000 0.54545455 0.50000000
##  [7,] 1.0000000 0.8750000 0.7777778 0.7000000 0.63636364 0.58333333
##  [8,] 0.8750000 1.0000000 0.8888889 0.8000000 0.72727273 0.66666667
##  [9,] 0.7777778 0.8888889 1.0000000 0.9000000 0.81818182 0.75000000
## [10,] 0.7000000 0.8000000 0.9000000 1.0000000 0.90909091 0.83333333
## [11,] 0.6363636 0.7272727 0.8181818 0.9090909 1.00000000 0.91666667
## [12,] 0.5833333 0.6666667 0.7500000 0.8333333 0.91666667 1.00000000
## [13,] 0.5384615 0.6153846 0.6923077 0.7692308 0.84615385 0.92307692
## [14,] 0.5000000 0.5714286 0.6428571 0.7142857 0.78571429 0.85714286
## [15,] 0.4666667 0.5333333 0.6000000 0.6666667 0.73333333 0.80000000
## [16,] 0.4375000 0.5000000 0.5625000 0.6250000 0.68750000 0.75000000
## [17,] 0.4117647 0.4705882 0.5294118 0.5882353 0.64705882 0.70588235
## [18,] 0.3888889 0.4444444 0.5000000 0.5555556 0.61111111 0.66666667
## [19,] 0.3684211 0.4210526 0.4736842 0.5263158 0.57894737 0.63157895
## [20,] 0.3500000 0.4000000 0.4500000 0.5000000 0.55000000 0.60000000
##            [,13]      [,14]      [,15]     [,16]      [,17]      [,18]
##  [1,] 0.07692308 0.07142857 0.06666667 0.0625000 0.05882353 0.05555556
##  [2,] 0.15384615 0.14285714 0.13333333 0.1250000 0.11764706 0.11111111
##  [3,] 0.23076923 0.21428571 0.20000000 0.1875000 0.17647059 0.16666667
##  [4,] 0.30769231 0.28571429 0.26666667 0.2500000 0.23529412 0.22222222
##  [5,] 0.38461538 0.35714286 0.33333333 0.3125000 0.29411765 0.27777778
##  [6,] 0.46153846 0.42857143 0.40000000 0.3750000 0.35294118 0.33333333
##  [7,] 0.53846154 0.50000000 0.46666667 0.4375000 0.41176471 0.38888889
##  [8,] 0.61538462 0.57142857 0.53333333 0.5000000 0.47058824 0.44444444
##  [9,] 0.69230769 0.64285714 0.60000000 0.5625000 0.52941176 0.50000000
## [10,] 0.76923077 0.71428571 0.66666667 0.6250000 0.58823529 0.55555556
## [11,] 0.84615385 0.78571429 0.73333333 0.6875000 0.64705882 0.61111111
## [12,] 0.92307692 0.85714286 0.80000000 0.7500000 0.70588235 0.66666667
## [13,] 1.00000000 0.92857143 0.86666667 0.8125000 0.76470588 0.72222222
## [14,] 0.92857143 1.00000000 0.93333333 0.8750000 0.82352941 0.77777778
## [15,] 0.86666667 0.93333333 1.00000000 0.9375000 0.88235294 0.83333333
## [16,] 0.81250000 0.87500000 0.93750000 1.0000000 0.94117647 0.88888889
## [17,] 0.76470588 0.82352941 0.88235294 0.9411765 1.00000000 0.94444444
## [18,] 0.72222222 0.77777778 0.83333333 0.8888889 0.94444444 1.00000000
## [19,] 0.68421053 0.73684211 0.78947368 0.8421053 0.89473684 0.94736842
## [20,] 0.65000000 0.70000000 0.75000000 0.8000000 0.85000000 0.90000000
##            [,19] [,20]
##  [1,] 0.05263158  0.05
```

```
##  [2,] 0.10526316  0.10
##  [3,] 0.15789474  0.15
##  [4,] 0.21052632  0.20
##  [5,] 0.26315789  0.25
##  [6,] 0.31578947  0.30
##  [7,] 0.36842105  0.35
##  [8,] 0.42105263  0.40
##  [9,] 0.47368421  0.45
## [10,] 0.52631579  0.50
## [11,] 0.57894737  0.55
## [12,] 0.63157895  0.60
## [13,] 0.68421053  0.65
## [14,] 0.73684211  0.70
## [15,] 0.78947368  0.75
## [16,] 0.84210526  0.80
## [17,] 0.89473684  0.85
## [18,] 0.94736842  0.90
## [19,] 1.00000000  0.95
## [20,] 0.95000000  1.00
```

```
#Part B
tL <- t(L)
#print(tL)

if(all.equal(L, tL)){
  print('Yes it is symmetric!')
}
```

```
## [1] "Yes it is symmetric!"
```

```
#Part C
C = solve(L)
#print(C)
all.equal((C%*%L),diag(1,20,20))
```

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

```
#Part D
d = c(1,2,3,4,5,6,7,8,9,10,10,9,8,7,6,5,4,3,2,1)

#Part E
#Ax = Cd
#matrix multiplication A %*% B
```

- Note I use L here instead of A

Solve for x: $Lx = Cd$

We know that $C = L'$

$x = L'Cd$ and so $x = C * Cd$

```
x = C%*%C%*%d
print(x)
```

```
##                 [,1]
##  [1,] -5.304478e-15
##  [2,]  1.248443e-14
##  [3,]  1.737561e-14
##  [4,] -7.210962e-14
##  [5,]  9.849542e-15
##  [6,] -1.489992e-14
##  [7,]  1.127494e-13
##  [8,] -3.373619e-15
##  [9,] -2.481203e+01
## [10,]  2.006424e+01
## [11,]  3.581375e+01
## [12,] -3.006263e+01
## [13,] -3.736996e-04
## [14,] -2.772044e-04
## [15,] -2.099688e-04
## [16,] -1.619541e-04
## [17,] -1.269228e-04
## [18,] -1.008779e-04
## [19,]  9.505933e+01
## [20,] -1.000629e+02
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.