Carson Slater

6375 Homework 5 Solutions

Preliminary Citations

In creating my `hh()` functions for parts 3(c) and 3(d), there was some subsetting help I recieved from Theophilus (I couldn't figure out a good way to specify the correct lower block). Likewise, I also rely on the `Matrix` package (specifically the `bdiag` function) to create a block diagonal matrix to compile the Q_k's and ensure they have the correct dimensions when recomposing them at the end of the function. Addidtionally, for parts 4(a-d), I found this `wttr` help file to be very useful, and obviously, I can thank the Linux man pages information about the `curl` command.

Question 1

I have read Chapters 10 and 11 of Trefethen and Bau.

Question 2

```
mgs <- function(A) {
  n <- nrow(A)
  m <- ncol(A)
  Q <- matrix(0, n, m)

  # normalize first column
  Q[, 1] <- normalize(A[, 1])

  for (i in 2:(m)) {

    # iterate through remaining columns of A projecting them onto Q[, i-1]
    for (j in i:m) {
      A[, j] <- A[, j] - project_onto(A[, j], Q[, i-1],
                           normalized_a = FALSE)
    }

    # normalized A is set to Q[, i]
    Q[, i] <- if ( is.zero_vec(A[, i]) ) rep(0, nrow(A))
    else normalize(A[, i])
```

```
  }
  discard_zero_cols(Q)
}
```

```
# testing mgs()

(A <- matrix(c( 1, 6, 19, 2,
                1, 2, 7, 3,
                5, 6, 23, 2
                ), nrow = 3,
              byrow = TRUE))
#      [,1] [,2] [,3] [,4]
# [1,]    1    6   19    2
# [2,]    1    2    7    3
# [3,]    5    6   23    2

gs(A)
#           [,1]       [,2]       [,3]
# [1,] 0.1924501  0.9678053 -0.1622214
# [2,] 0.1924501  0.1248781  0.9733285
# [3,] 0.9622504 -0.2185367 -0.1622214

mgs(A)
#           [,1]       [,2]       [,3]
# [1,] 0.1924501  0.9678053 -0.1622214
# [2,] 0.1924501  0.1248781  0.9733285
# [3,] 0.9622504 -0.2185367 -0.1622214

qr(A) |> qr.Q()
#            [,1]       [,2]       [,3]
# [1,] -0.1924501  0.9678053  0.1622214
# [2,] -0.1924501  0.1248781 -0.9733285
# [3,] -0.9622504 -0.2185367  0.1622214
```

Question 3

(a)

Here we show the householder matrix $\mathbf{F}$ for $a_1 = \langle 3, 4 \rangle$ works on a_1 to create the vector $||a_1||_2 e_1$.

```r
a1 <- c(3, 4)

# create the unit vector u
u <- normalize(a1 - norm(a1) * c(1, 0))

# householder matrix F
# used the `base` tcrossprod() function for the outer product
(F <- diag(length(a1)) -
    2 * (tcrossprod(u, u) / dot(u, u)))
#      [,1] [,2]
# [1,]  0.6  0.8
# [2,]  0.8 -0.6

# apply F to a1
F %*% a1
#      [,1]
# [1,]    5
# [2,]    0
```

(b)

```r
make_reflector <- function(A) {
  n <- nrow(A)
  m <- ncol(A)
  I <- diag(n)

  x <- A[, 1]  # Choose the first column as the target vector

  v <- x
  v[1] <- x[1] - norm(x)

  u <- normalize(v)

  F <- I - 2 * tcrossprod(u, u)

  F
}
```

```r
# testing make_reflector
A <- matrix(c( 1, 3, 1,
  5, 6, 1,
  2, 4, 1,
  7, 3, 1,
  3, 3, 1
), nrow = 5, byrow = TRUE)

make_reflector(A) |> round(2)
#      [,1]  [,2]  [,3]  [,4]  [,5]
# [1,] 0.11  0.53  0.21  0.75  0.32
# [2,] 0.53  0.68 -0.13 -0.45 -0.19
# [3,] 0.21 -0.13  0.95 -0.18 -0.08
# [4,] 0.75 -0.45 -0.18  0.38 -0.27
# [5,] 0.32 -0.19 -0.08 -0.27  0.89

make_reflector(A) %*% A |> round(2)
#      [,1]  [,2] [,3]
# [1,] 9.38  7.57 1.92
# [2,] 0.00  3.27 0.45
# [3,] 0.00  2.91 0.78
# [4,] 0.00 -0.82 0.23
# [5,] 0.00  1.36 0.67
```

(c)

```r
hh <- function(A) {
  if (!is.matrix(A)) {
    stop("Input must be a matrix.")
  }

  m <- nrow(A)
  n <- ncol(A)

  if (m < n) {
    stop("Householder triangularization requires a matrix with more rows than
         columns.")
  }
  # making reflector for input matrix A
  A <- make_reflector(A) %*% A
```

```r
  # subsetting the matrix with the lower block
  # computing the householder transformation
  for (i in 2:n) {
    block <- A[i:m, -c(1:i - 1), drop = FALSE]
    A[i:m, -c(1:i - 1)] <- make_reflector(block) %*% block
  }
  A
}
```

```r
(R <- hh(A) |> zapsmall())
#          [,1]     [,2]     [,3]
# [1,] 9.380832 7.568625 1.918806
# [2,] 0.000000 4.660033 0.960781
# [3,] 0.000000 0.000000 0.628555
# [4,] 0.000000 0.000000 0.000000
# [5,] 0.000000 0.000000 0.000000
```

(d)

```r
hh <- function(A) {
  if (!is.matrix(A)) {
    stop("Input must be a matrix.")
  }

  m <- nrow(A)
  n <- ncol(A)
  F_list <- list()

  if (m < n) {
    stop("Householder triangularization requires a matrix with more rows than
         columns.")
  }
  # making reflector for input matrix A
  F_list[[1]] <- make_reflector(A)
  A <- F_list[[1]] %*% A

  # subsetting the matrix with the lower block
  # computing the householder transformation
  for (i in 2:n) {
```

```r
    block <- A[i:m, -c(1:i - 1), drop = FALSE]

    F_list[[i]] <- make_reflector(block)
    A[i:m, -c(1:i - 1)] <- F_list[[i]] %*% block

    F_list[[i]] <- bdiag(diag(i-1), F_list[[i]])
  }

  Q <- Reduce("%*%", F_list) |> as.matrix()
  list("Q" = Q, "R" = A)
}
```

```r
# testing second hh() QR function
hh(A) |> lapply(zapsmall)
# $Q
#           [,1]       [,2]       [,3]       [,4]       [,5]
# [1,] 0.1066004  0.4706365  0.5461350  0.5921912  0.3437786
# [2,] 0.5330018  0.4218659 -0.6810037  0.0404758  0.2693212
# [3,] 0.2132007  0.5120916  0.1573468 -0.1214275 -0.8079635
# [4,] 0.7462025 -0.5681778  0.1814900  0.2455008 -0.1647622
# [5,] 0.3198011  0.1243651  0.4245867 -0.7567404  0.3596259
#
# $R
#          [,1]     [,2]     [,3]
# [1,] 9.380832 7.568625 1.918806
# [2,] 0.000000 4.660033 0.960781
# [3,] 0.000000 0.000000 0.628555
# [4,] 0.000000 0.000000 0.000000
# [5,] 0.000000 0.000000 0.000000

list(
  "Q" = qr.Q(qr(A), complete = TRUE),
  "R" = qr.R(qr(A), complete = TRUE)
)
# $Q
#            [,1]       [,2]       [,3]         [,4]        [,5]
# [1,] -0.1066004 -0.4706365 -0.5461350 -0.565296749 -0.38641138
# [2,] -0.5330018 -0.4218659  0.6810037 -0.260497806  0.07945488
# [3,] -0.2132007 -0.5120916 -0.1573468  0.781493418 -0.23836465
# [4,] -0.7462025  0.5681778 -0.1814900  0.042973883 -0.29252430
# [5,] -0.3198011 -0.1243651 -0.4245867  0.001327254  0.83784545
```

```
#
# $R
#            [,1]       [,2]       [,3]
# [1,] -9.380832 -7.568625 -1.9188064
# [2,]  0.000000 -4.660033 -0.9607813
# [3,]  0.000000  0.000000 -0.6285548
# [4,]  0.000000  0.000000  0.0000000
# [5,]  0.000000  0.000000  0.0000000

with(hh(A), Q %*% R)
#      [,1] [,2] [,3]
# [1,]    1    3    1
# [2,]    5    6    1
# [3,]    2    4    1
# [4,]    7    3    1
# [5,]    3    3    1

with(hh(A), crossprod(Q) |> zapsmall())
#      [,1] [,2] [,3] [,4] [,5]
# [1,]    1    0    0    0    0
# [2,]    0    1    0    0    0
# [3,]    0    0    1    0    0
# [4,]    0    0    0    1    0
# [5,]    0    0    0    0    1
```

Question 4

(a)

The `curl` command is used to transfer information from a server and can handle many protocols. It fetches information from as many URL's as it is given, and does so in a sequential order.

(b)

This command, `curl -s "http://wttr.in/Waco,TX"`, neatly prints the weather forecast for the current moment, day and subsequent two days.

(c)

The `wttr` help file was helpful in specifying which argument needed to be passed through so that we only displayed today's current weather and forecast.

```
# add a ?1 to the end of the URL to tell feed the `1` parameter into the view argument
curl -s "http://wttr.in/Waco,TX?T1"
```

(d)

```
weather() {
  curl -s "http://wttr.in/${1:-Waco,TX}?T1"
}

weather Houston,TX
# Weather report: Houston,TX
#
#       \   /     Sunny
#         .-.     +51(50) °F
#      (   )     ↓ 4 mph
#         `-'     9 mi
#       /   \     0.0 in
#
#                                Wed 01 Nov
#              Morning                          Noon                      Evening
#
#       \   /     Sunny               \   /     Sunny               \   /     Sunny
#         .-.     +46(41) °F            .-.     +51(57) °F            .-.     64 °F
#      (   )      9-11 mph          (   )     ↓ 4-7 mph          (   )     ↓ 5-7 mph
#         `-'     6 mi                `-'     9 mi                `-'     6 mi
#       /   \     0.0 in | 0%         /   \     0.0 in | 0%         /   \     0.0 in | 0%
#
# Location: Houston, Harris County, Texas, 77002, United States of America [29.7589382,-95
#
# Follow @igor_chubin for wttr.in updates
```

(e)

The `-it` part stands for 'interactive terminal,' and it basically tells the container to open up a terminal when the container is opened. The `--rm` part of the container basically tells the

container to remove itself from memory when it is exited. This ensures that disc space is not occupied by stopped containers that are no longer being used.

(f)

The `tldr docker run` page stipulates we can add the `--volume` argument to open a new container with bind mounted volumes. We can substitute `--volume` for `-v`. Modifying the docker command we use for class to access the user's home directory would look like this:

```
docker run -it --rm -v ~/:/dkahle/student/host_home zsh -c "cd /home/student; su student"
```