

STAT-S675

Homework 6

John Koo

Problem 1

Problem 2

Problem 3

Parts 1 and 2

```
import::from(readr, read_table)
import::from(magrittr, `%>%`, `%<>%`)
source('http://pages.iu.edu/~mtrosset/Courses/675/stress.r')
library(ggplot2)
theme_set(ggthemes::theme_base())
set.seed(6756)

# load the data
# note that it's a lower triangular matrix--fill in the rest
data.url <- 'http://pages.iu.edu/~mtrosset/Courses/675/colors.dat'
Delta <- read_table(data.url, col_names = FALSE) %>%
  as.matrix() %>%
  {. + t(.)}

# generate random 2 column matrix
rows <- nrow(Delta)
cols <- 2
stdev <- 40
X.rand <- matrix(rnorm(rows * cols, sd = stdev), nrow = rows, ncol = cols)

# compute raw stress of the random matrix
mds.stress.raw.eq(X.rand, Delta)

[1] 160697

# center X.rand
X.rand %<>% apply(2, function(x) x - mean(x))
apply(X.rand, 2, sum)

[1] -2.309264e-14  6.217249e-15

# and check its stress criterion
mds.stress.raw.eq(X.rand, Delta)

[1] 160697
```

Centering the data matrix did not change the stress criterion. Looking at `mds.stress.raw.eq` more closely, this is expected, since it just compares the distance matrix of `X.rand` to `Delta`, and shifting `X.rand` doesn't

do anything to its distance matrix.

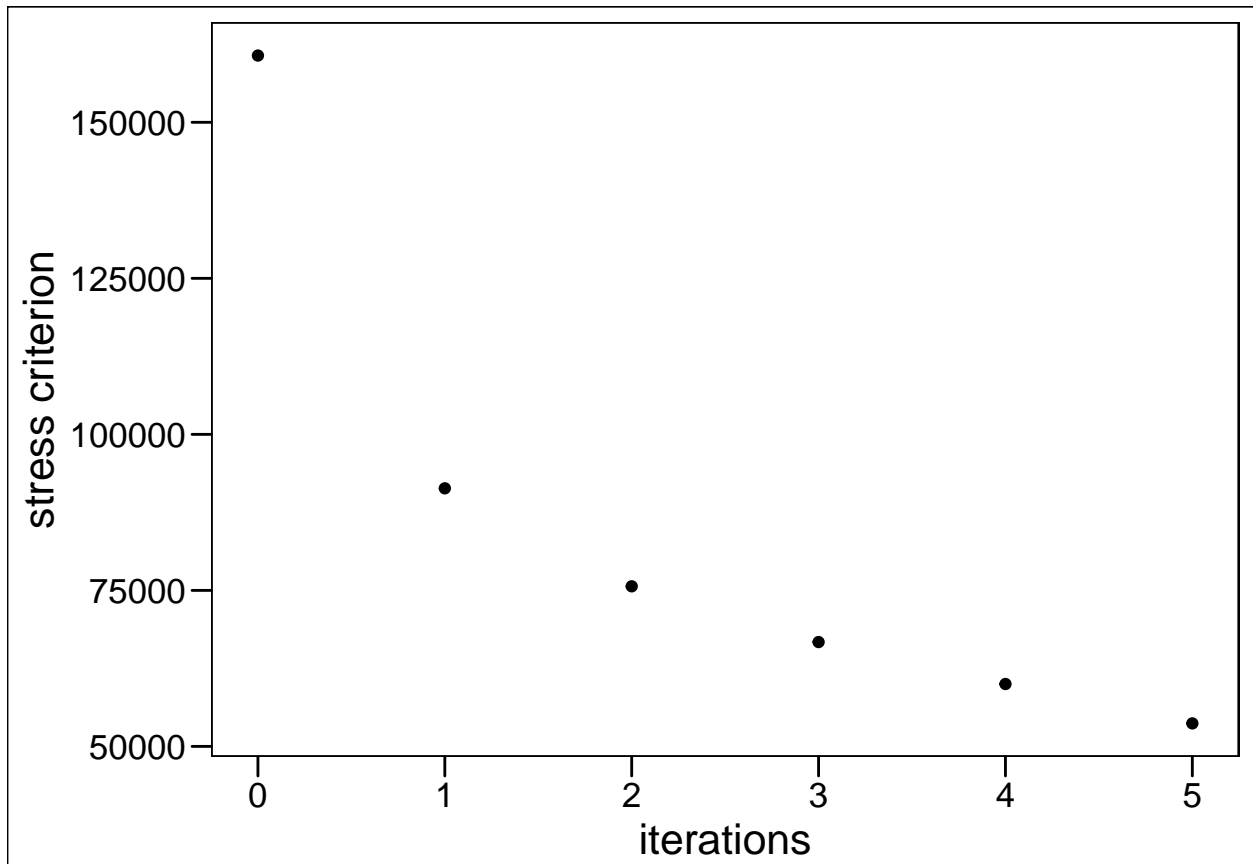
Part 3

```
# iterations
K <- 64
k.vector <- seq(0, K)

# initialize stress vector
stress.vector <- rep(NA, K + 1)
stress.vector[1] <- mds.stress.raw.eq(X.rand, Delta)

# iterate
for (k in k.vector[-1]) {
  X.rand <- mds.guttman.eq(X.rand, Delta)
  stress.vector[k + 1] <- mds.stress.raw.eq(X.rand, Delta)
}

mds.df <- dplyr::data_frame(k = k.vector, random.stress = stress.vector)
ggplot(mds.df[seq(6), ]) +
  geom_point(aes(x = k, y = random.stress)) +
  labs(x = 'iterations', y = 'stress criterion')
```



Part 4

```
X.cmds <- cmdscale(Delta)
mds.stress.raw.eq(X.cmds, Delta)
```

```
[1] 25880.08
```

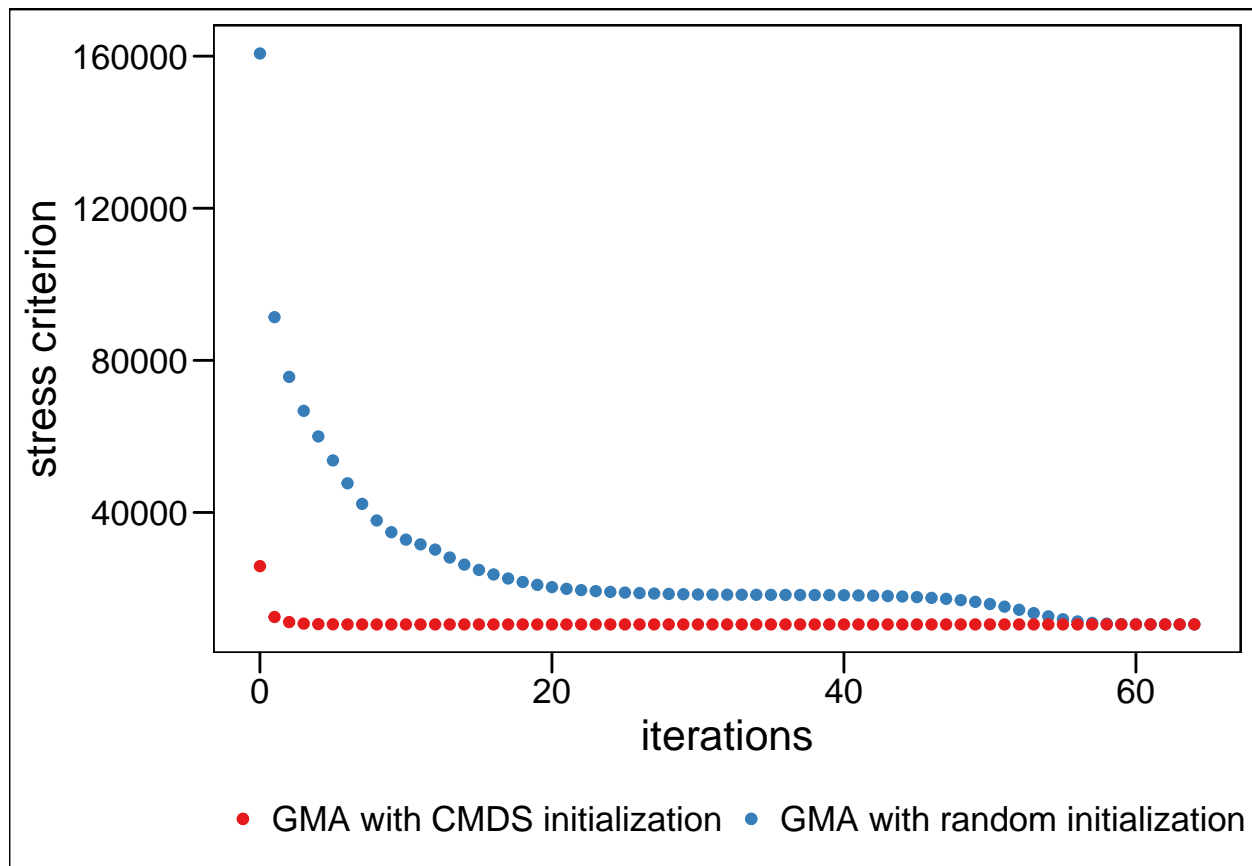
Part 5

```
# initialize
cmds.stress.vector <- rep(NA, K + 1)
cmds.stress.vector <- mds.stress.raw.eq(X.cmds, Delta)

# iterate
for (k in k.vector[-1]) {
  X.cmds <- mds.guttman.eq(X.cmds, Delta)
  cmds.stress.vector[k + 1] <- mds.stress.raw.eq(X.cmds, Delta)
}

mds.df$cmds.stress <- cmds.stress.vector

ggplot(mds.df) +
  geom_point(aes(x = k,
                 y = random.stress,
                 colour = 'GMA with random initialization')) +
  geom_point(aes(x = k,
                 y = cmds.stress,
                 colour = 'GMA with CMDS initialization')) +
  scale_colour_brewer(palette = 'Set1') +
  theme(legend.position = 'bottom') +
  labs(x = 'iterations', y = 'stress criterion', colour = NULL)
```



Here we see that while the CMDS initialization basically converges after 3 or 4 iterations, the randomly initialized matrix has a much higher stress criterion and is far from converging. In fact, it takes ~60 iterations for the random matrix to converge to the same stress criterion as the CMDS initialization, and it seems to have gotten stuck at a higher value for ~30 iterations.