Charles Zheng EE 378b HW 5

==========

1

Load the data

```
setwd('~/github/misc/ee378b')
cities <- read.table('cities.txt', sep = '\t', skip = 2)
colnames(cities) <- c('city', 'latdeg', 'latmin', 'longdeg', 'longmin')
lats <- cities$latdeg + cities$latmin/60
longs <- cities$longdeg + cities$longmin/60
```

Convert to 3d coordinates

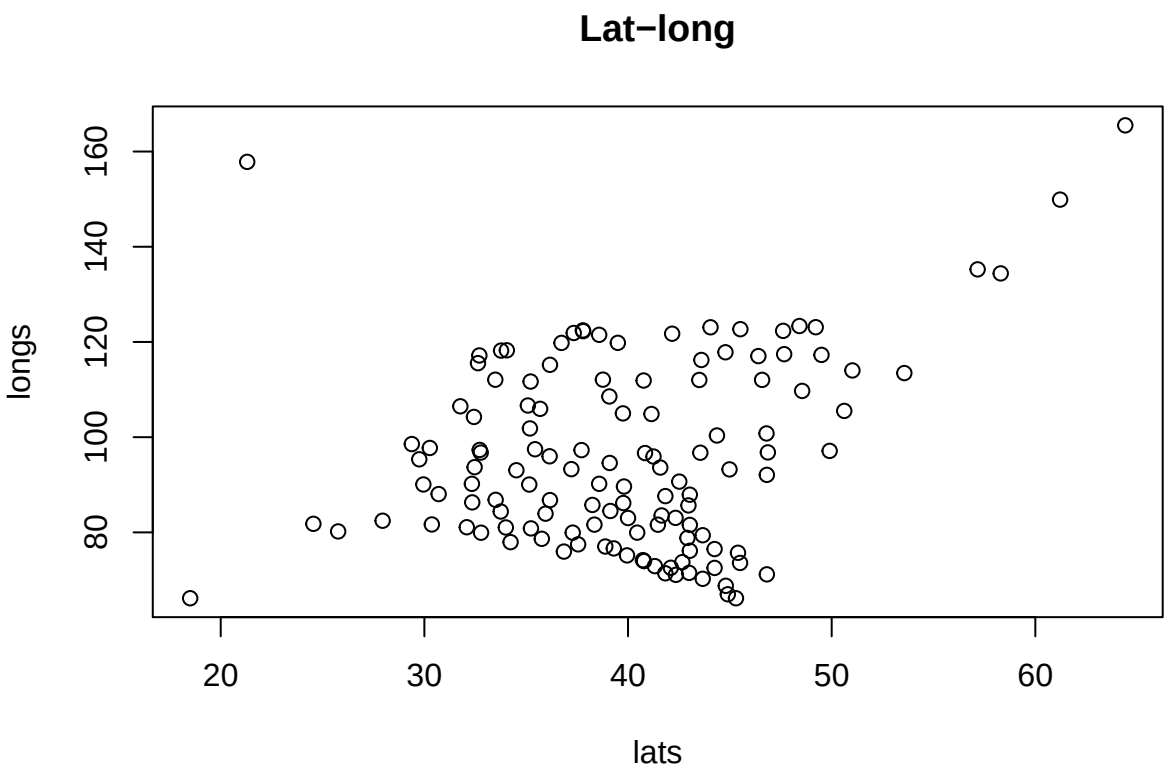
```
z <- sin(pi * lats/180)
y <- cos(pi * lats/180) * sin(pi * longs/180)
x <- cos(pi * lats/180) * cos(pi * longs/180)
pos <- cbind(x, y, z)
```

Compute inner products and convert to arc distance

```
ips <- pos %*% t(pos)
ips[ips > 1] <- 1
dm <- acos(ips)
```

For later reference, plot the original coordinates.

```
plot(lats, longs)
title("Lat-long")
```



2

Convert each row of the distance matrix to 0-1 matrix, with 1 only for nearest neighbors. Then symmetrize the matrix

```
diag(dm) <- Inf
adj <- apply(dm, 1, function(v) {
    as.numeric(v < sort(v)[6])
  })
adj <- adj + t(adj)
adj[adj > 1] <- 1
```

3

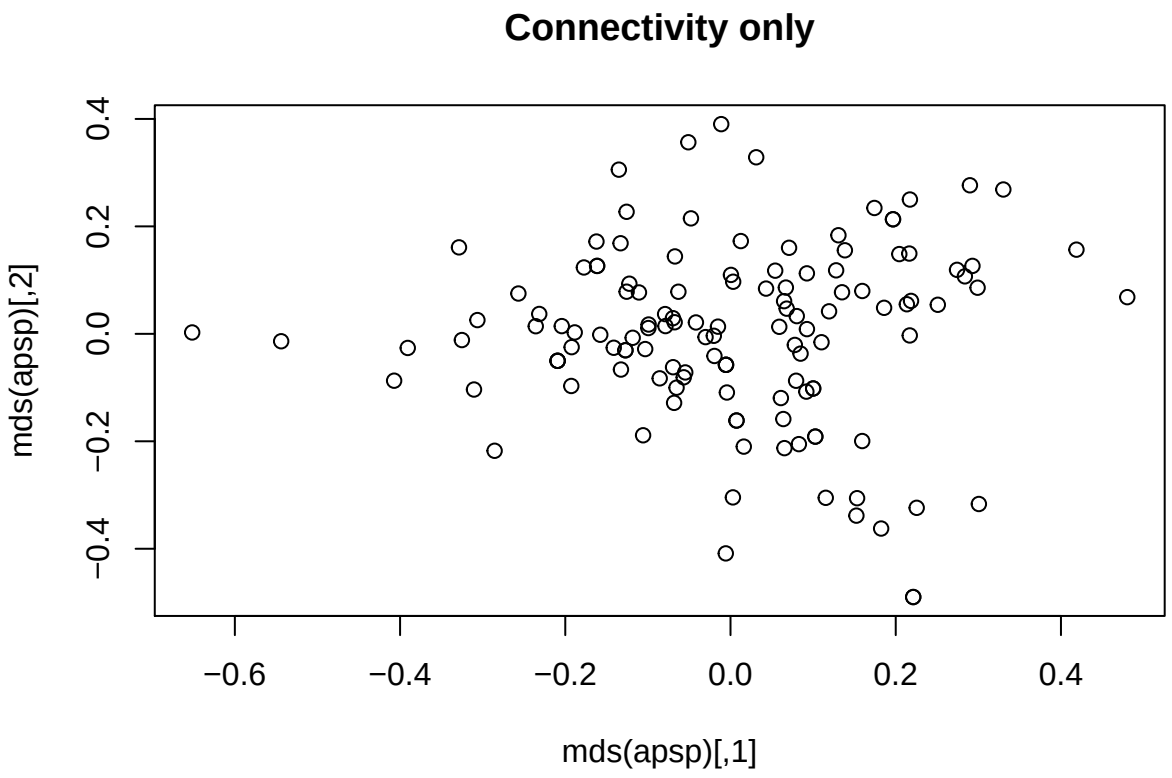
Compute shortest paths

```
apsp <- adj
apsp[apsp == 0] <- Inf
diag(apsp) <- 0

minplusmult <- function(a, b) {
  n <- dim(a)[1]
  p <- dim(a)[2]
  q <- dim(b)[2]
  c <- matrix(0, n, q)
  for (i in 1:n) {
```

```
    for (j in 1:q) {
      c[i, j] <- min(a[i, ] + b[, j])
    }
  }
  c
}

while(sum(apsp == Inf) > 0) apsp <- minplusmult(apsp, apsp)
```

Apply MDS-map

```
a <- apsp
mds <- function(a, d = 2) {
  n <- dim(a)[1]
  u <- t(t(rep(1, n)))
  tu <- t(u)
  b <- a - (1/n) * u %*% (tu %*% a) - (1/n) * (a %*% u) %*% tu + mean(a)
  res <- eigen(b)
  coords <- res$vectors[, 1:d] %*% diag(sqrt(res$values[1:d]))
  coords
}
plot(mds(apsp), main = "Connectivity only")
```



4

Compute the distance matrix

```r
apsp <- adj * dm
apsp[apsp == 0] <- Inf
diag(apsp) <- 0
while(sum(apsp == Inf) > 0) apsp <- minplusmult(apsp, apsp)
plot(mds(apsp), main = "With edge weights")
```