STAT-S675

Homework 8

John Koo

Link to assignment

```
dp <- loadNamespace('dplyr')
import::from(magrittr, `%>%`, `%<>%`)
import::from(foreach, foreach, `%do%`)
import::from(readr, read_table2)
import::from(scatterplot3d, scatterplot3d)
library(ggplot2)

source('http://pages.iu.edu/~mtrosset/Courses/675/out.r')
source('http://pages.iu.edu/~mtrosset/Courses/675/stress.r')
source('http://pages.iu.edu/~mtrosset/Courses/675/manifold.r')

theme_set(theme_bw())
```

Exercise 7.4.1

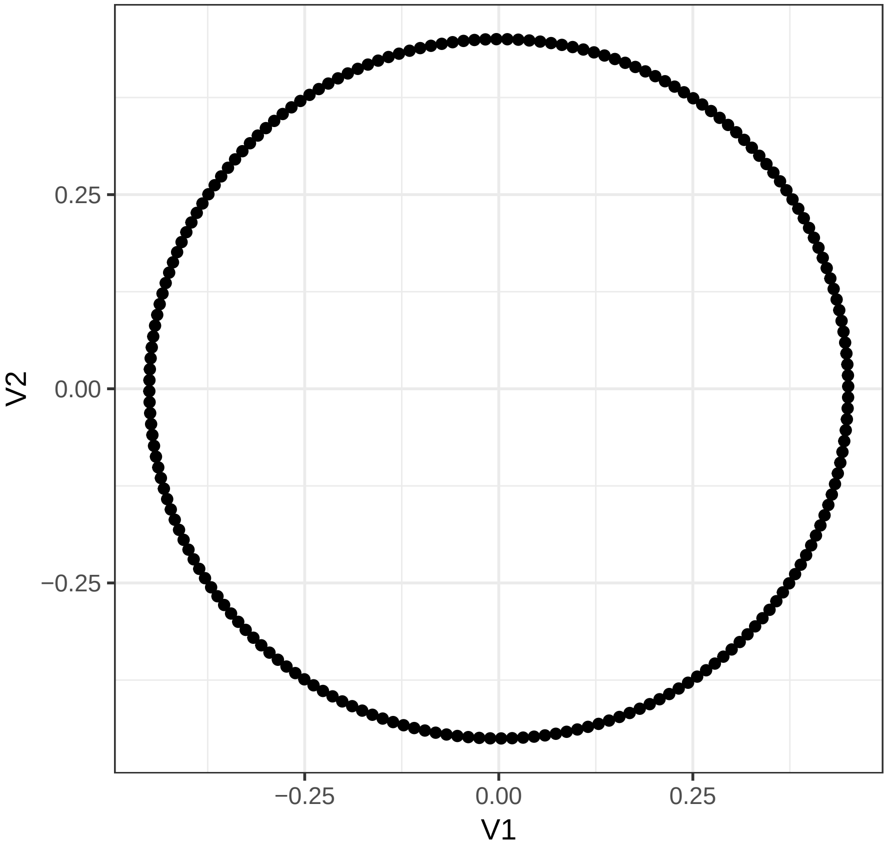
[From the text]

Here we will construct Δ from the circle (since the result is identical in either case).

```
# number of points
n <- 200

# radius of circle
r <- 1 / pi

# construct dissimilarity matrix
theta <- seq(2 * pi / n, 2 * pi, 2 * pi / n)
Delta <- foreach(i = theta, .combine = rbind) %do% {
  foreach(j = theta, .combine = c) %do% {
    theta.ij <- abs(i - j)
    if (theta.ij > pi) theta.ij <- 2 * pi - theta.ij
    arc.length <- theta.ij * r
    return(arc.length)
  }
}
```

```
# spectral decomposition of tau(Delta2)
Delta2 <- Delta ** 2
B <- mds.tau(Delta2)
B.eigen <- eigen(B)

# find first two components
projection.df <- B.eigen$vectors[, 1:2] %>%
  apply(2, function(x) x * sqrt(B.eigen$values[1:2])) %>%
  as.data.frame()
```

```
# plot first two components
ggplot(projection.df) +
  geom_point(aes(x = V1, y = V2)) +
  coord_fixed()
```



We can look at the rows of $B = \tau(\Delta_2)$ to obtain b^2:

```
summary(diag(B))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.1667  0.1667  0.1667  0.1667  0.1667  0.1667 
```

```
b <- sqrt(mean(diag(B)))
print(b ** 2)
```

```
[1] 0.166675
```

Then $b^2 \approx 0.1667$ (and $b \approx 0.4083$).

If b is the radius of the embedding, this is consistent with our intuition. The longest arc length in S is 1, so we expect the longest pairwise Euclidean distance of the embedding, $2b$, or the diameter of of the embedding, to be approximately this length.

To find a, the angle between two consecutive points, we want $A_{i,i+1}$, where $A = \cos^{-1} \frac{B}{b^2}$.

```
A <- acos(B / b ** 2)
print(A[1:6, 1:6])
```

```
           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[1,]        NaN 0.02449490 0.04899347 0.07349940 0.09801637 0.12254810
```

```
[2,] 0.02449490        NaN 0.02449490 0.04899347 0.07349940 0.09801637
[3,] 0.04899347 0.02449490        NaN 0.02449490 0.04899347 0.07349940
[4,] 0.07349940 0.04899347 0.02449490        NaN 0.02449490 0.04899347
[5,] 0.09801637 0.07349940 0.04899347 0.02449490        NaN 0.02449490
[6,] 0.12254810 0.09801637 0.07349940 0.04899347 0.02449490        NaN
```

We can see that $a \approx 0.02449$. In the original circle S, the angle between two consecutive values was $\frac{2\pi}{200} \approx 0.0314$.

Analytical derivation of b^2

Define $B = \tau(\Delta_2) = -\frac{1}{2}P\Delta_2 P$ where $P = I - \frac{ee^\top}{n}$. Then:

$$\begin{aligned} B = \tau(\Delta_2) &= -\frac{1}{2}P\Delta_2 P \\ &= -\frac{1}{2}(I - \frac{ee^\top}{n})\Delta_2(I - \frac{ee^\top}{n}) \\ &= -0.5(\Delta_2 - \frac{ee^\top}{n}\Delta_2)(I - \frac{ee^\top}{n}) \\ &= -0.5(\Delta_2 - \Delta_2\frac{ee^\top}{n} - \frac{ee^\top}{n}\Delta_2 + \frac{ee^\top ee^\top}{n^2}\Delta_2) \end{aligned}$$

We know that $ee^\top = n$ and since Δ_2 and $\frac{ee^\top}{n}$ are symmetric, the two matrices commute. Therefore, this is equal to:

$$\begin{aligned} &= -0.5(\Delta_2 - 2\frac{ee^\top}{n}\Delta_2 + \frac{nee^\top}{n^2}\Delta_2) \\ &= -0.5(\Delta_2 - \frac{ee^\top}{n}\Delta_2) \\ &= -\frac{1}{2}(I - \frac{ee^\top}{n})\Delta_2 \\ &= -\frac{1}{2}P\Delta_2 \end{aligned}$$

We know that the diagonal entries of this matrix are just half the row means of Δ_2 subtracted by the diagonal entries of Δ_2. However, $diag(\Delta_2) = 0$, so ...

$$[B]_{jj} = diag(\frac{1}{2}P\Delta_2) = \frac{1}{2n}\sum_k^n \delta_{kj}^2$$

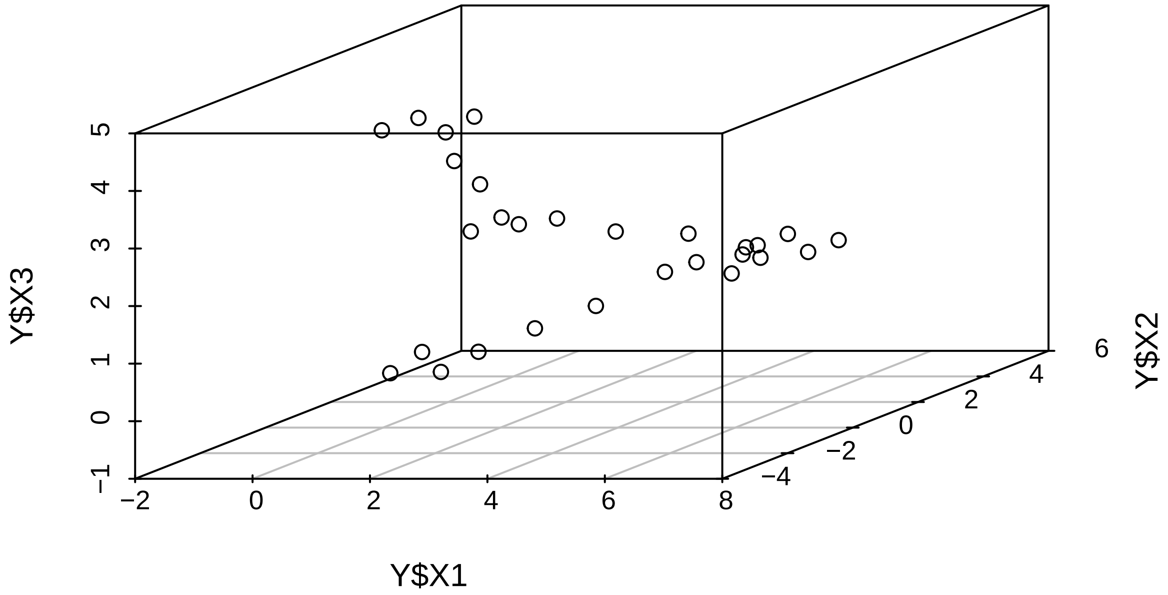
This is equivalent $\forall j \leq n$, so we can just look at the first row of Δ_2. $\delta_{1.}^2 = [0, .01^2, 0.2^2, 0.3^2, ..., .99^2, 1, .99^2, ..., .01^2]$ $= \frac{1}{100^2}[0, 1, 4, 9, ..., 9801, 10000, 9801, ..., 4, 1]$

Then $\sum_k^n \delta_{kj}^2 = \frac{1}{100^2}(\sum_k^{100} k^2 + \sum_k^{99} k^2) = \frac{1}{100^2}(\frac{(100)(101)(201)}{6} + \frac{(99)(100)(199)}{6}) = 66.67$

Therefore, $[B]_{jj} = diag(\frac{1}{2}P\Delta_2) = \frac{1}{2n}\sum_k^n \delta_{kj}^2 = \frac{66.67}{400} = 0.166675$

Other problems

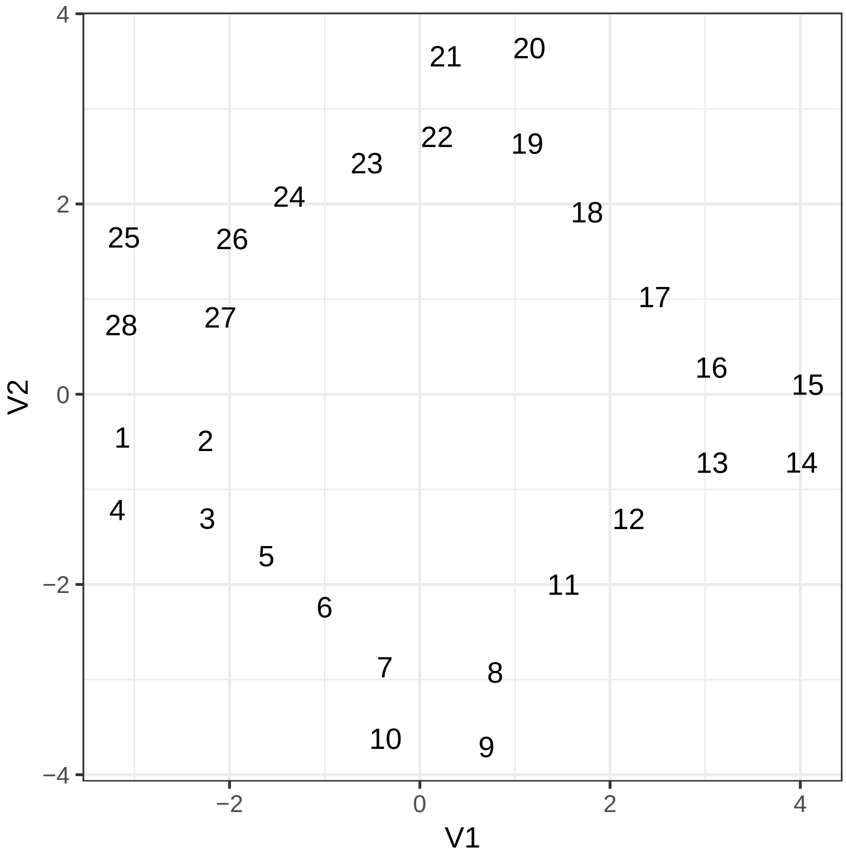
```
file.url <- 'http://pages.iu.edu/~mtrosset/Courses/675/manifold.y'
Y <- read_table2(file.url, col_names = FALSE)

scatterplot3d(Y$X1, Y$X2, Y$X3)
```



Problem 1

CMDS

```
# CMDS on Y
Y.dist <- mds.edm1(Y)
Y.cmds <- cmdscale(Y.dist)

# plot first two components
Y.cmds.df <- Y.cmds %>%
  as.data.frame() %>%
  dp$mutate(id = rownames(.))
ggplot(Y.cmds.df) +
  geom_text(aes(x = V1, y = V2, label = id)) +
  coord_fixed()
```



```
# rank by first component
rank.1 <- rank(Y.cmds.df$V1)
print(rank.1)
```

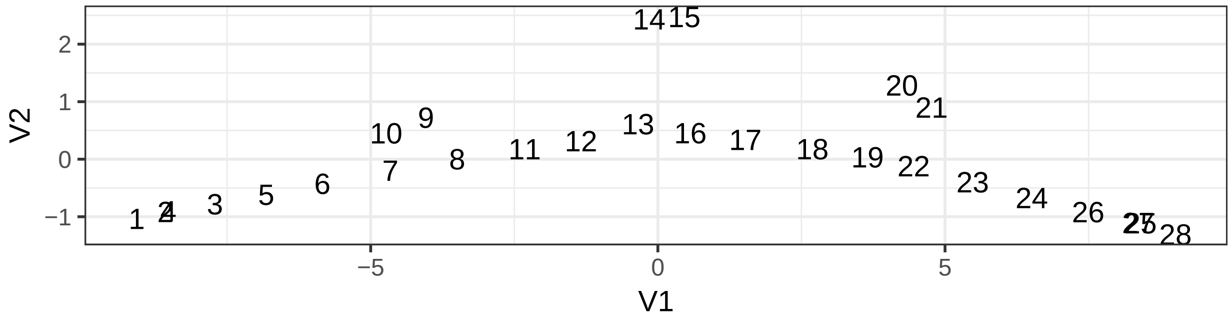
```
 [1]  3  5  6  1  9 11 13 18 17 14 21 23 26 27 28 25 24 22 19 20 16 15 12
[24] 10  4  8  7  2
```

Problem 2

Isomap with EDM-1 weights

```
# set k
k <- 4

# dimension of Y
n <- nrow(Y)

# define dissimilarity matrix for Y
Y.knn <- graph.knn(Y.dist, k)
Y.dis <- graph.dis(Y.knn, Y.dist)
Y.shortest <- graph.short(Y.dis)

Y.isomap.df <- Y.shortest %>%
  cmdscale() %>%
  as.data.frame() %>%
  dp$mutate(id = rownames(.))
```

```
ggplot(Y.isomap.df) +
  geom_text(aes(x = V1, y = V2, label = id)) +
  coord_fixed()
```



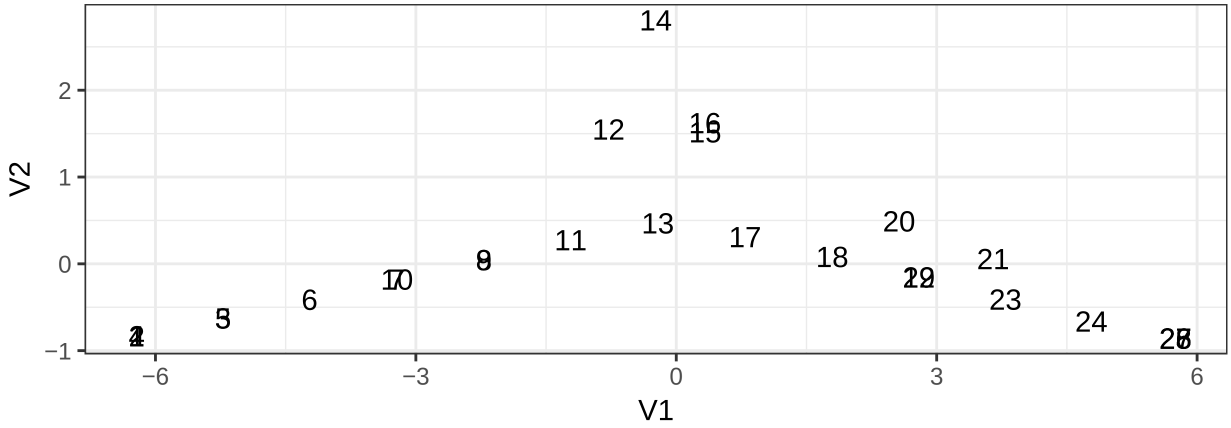
```
rank.2 <- rank(Y.isomap.df$V1)
print(rank.2)
```

```
 [1]  1  2  4  3  5  6  8 10  9  7 11 12 13 14 15 16 17 18 19 20 22 21 23
[24] 24 27 25 26 28
```

Problem 3

Isomap with unit weights

```
Y.dis.unit <- graph.dis(Y.knn, matrix(rep(1, n ** 2), ncol = n, nrow = n))
Y.shortest.unit <- graph.short(Y.dis.unit)

Y.isomap.unit.df <- Y.shortest.unit %>%
  cmdscale() %>%
  as.data.frame() %>%
  dp$mutate(id = rownames(.))

ggplot(Y.isomap.unit.df) +
  geom_text(aes(x = V1, y = V2, label = id)) +
  coord_fixed()
```



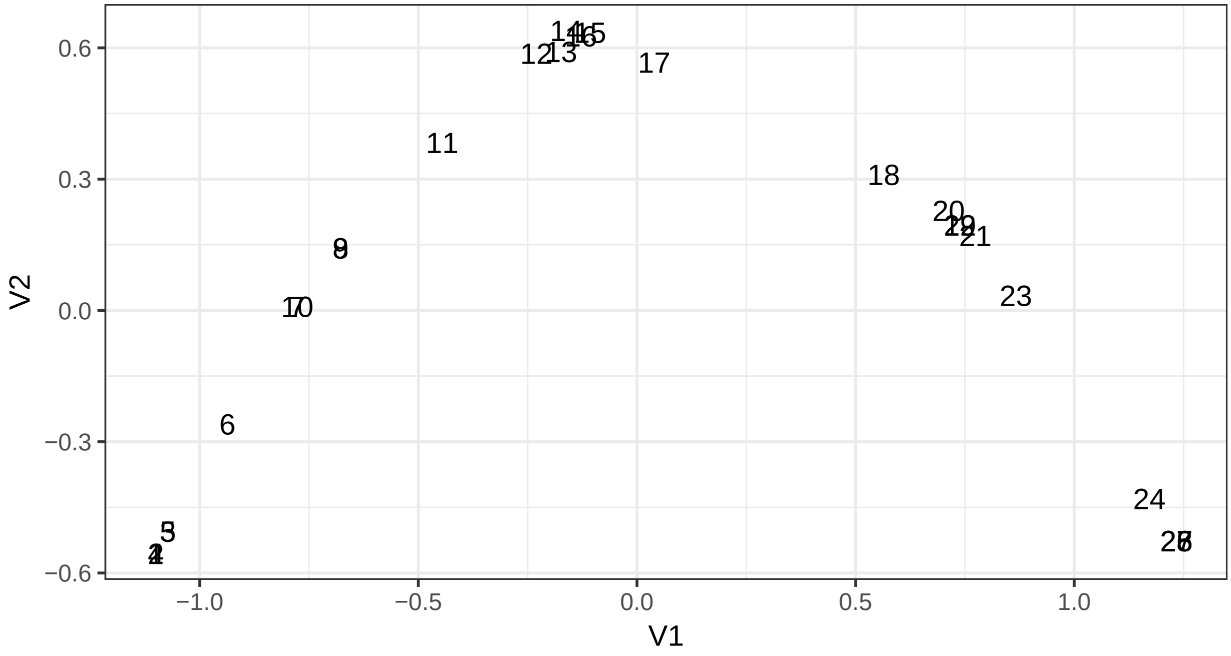
```
rank.3 <- rank(Y.isomap.unit.df$V1, ties.method = 'first')
print(rank.3)
```

```
 [1]  1  2  4  3  5  6  8  9 10  7 11 12 14 13 16 15 17 18 21 19 22 20 23
[24] 24 25 26 27 28
```

Problem 4

Laplacian eigenmap with adjacency matrix

```
Y.adj <- Y %>%
  mds.edm1() %>%
  graph.knn(k) %>%
  graph.adj()

L.adj <- graph.laplacian(Y.adj)
L.adj.eigen <- eigen(L.adj)
L.adj.df <- cbind(
  L.adj.eigen$vectors[, n - 1] / sqrt(L.adj.eigen$values[n - 1]),
  L.adj.eigen$vectors[, n - 2] / sqrt(L.adj.eigen$values[n - 2])
) %>%
  as.data.frame() %>%
    dp$mutate(id = as.numeric(rownames(.)))

ggplot(L.adj.df) +
  geom_text(aes(x = V1, y = V2, label = id)) +
  coord_fixed()
```



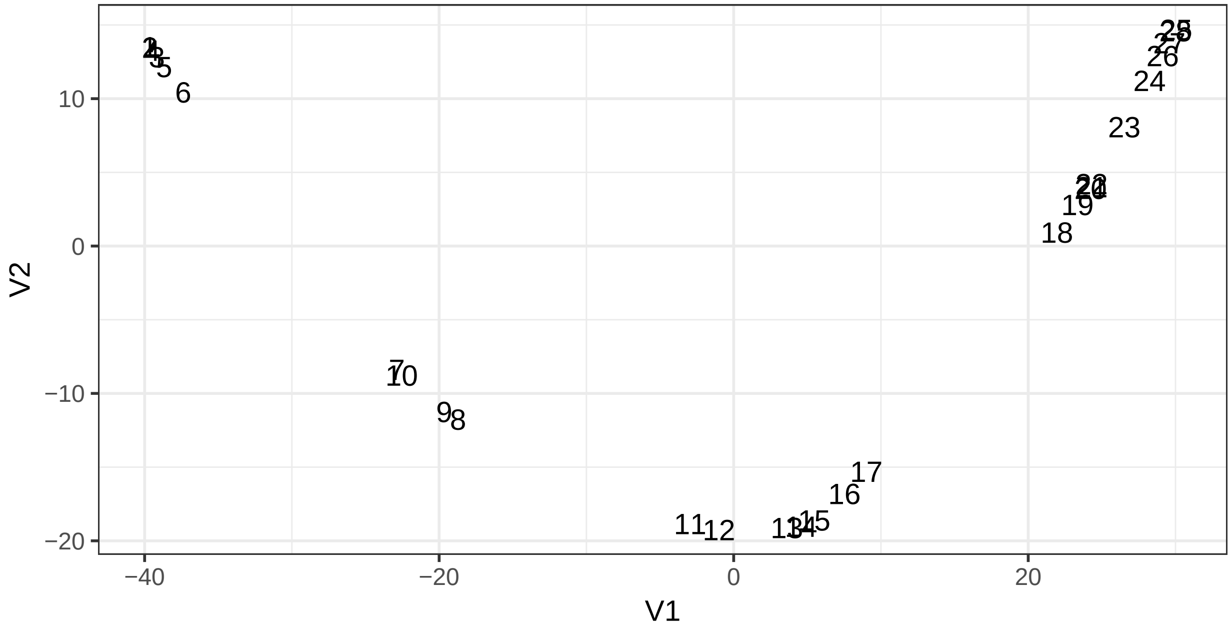
```
rank.4 <- rank(L.adj.df$V1)
print(rank.4)
```

```
 [1]  2  1  4  3  5  6  8 10  9  7 11 12 13 14 16 15 17 18 21 19 22 20 23
[24] 24 27 27 27 25
```

Problem 5

Laplacian eigenmap with heat kernel

```r
s <- 5

Y.heat <- Y %>%
  mds.edm1() %>%
  graph.heat(s)

L.heat <- graph.laplacian(Y.heat)

L.heat.eigen <- eigen(L.heat)

L.heat.df <- cbind(
  L.heat.eigen$vectors[, n - 1] / sqrt(L.heat.eigen$values[n - 1]),
  L.heat.eigen$vectors[, n - 2] / sqrt(L.heat.eigen$values[n - 2])
) %>%
  as.data.frame() %>%
    dp$mutate(id = as.numeric(rownames(.)))

ggplot(L.heat.df) +
  geom_text(aes(x = V1, y = V2, label = id)) +
  coord_fixed()
```



```r
rank.5 <- rank(L.heat.df$V1)
print(rank.5)
```

```
 [1]  1  2  4  3  5  6  7 10  9  8 11 12 13 14 15 16 17 18 19 20 21 22 23
[24] 24 28 25 26 27
```

Problem 6

```
cor(cbind(seq(n), rank.1, rank.2, rank.3, rank.4, rank.5),
    method = 'spearman')
```

```
                     rank.1     rank.2     rank.3    rank.4     rank.5
       1.0000000 0.07115490 0.99343186 0.99233716 0.9879520 0.99397920
rank.1 0.0711549 1.00000000 0.07443897 0.08045977 0.0859803 0.07553366
rank.2 0.9934319 0.07443897 1.00000000 0.99507389 0.9934284 0.99835796
rank.3 0.9923372 0.08045977 0.99507389 1.00000000 0.9945237 0.99178982
rank.4 0.9879520 0.08598030 0.99342841 0.99452370 1.0000000 0.99288076
rank.5 0.9939792 0.07553366 0.99835796 0.99178982 0.9928808 1.00000000
```

We can see that the CMDS method is the odd one out here. The first component of the CMDS method fails to recover any structure.

The Isomap and various Laplacian eigenmap methods perform better and all have similar results, with the heat map performing the best, at least based on just the ranking of the first component.

The fact that CMDS failed here matches our intuition. CMDS requires some sort of correlation and cannot detect pairwise similarities very well. Since our original data `Y` did not have this correlated structure, it failed to find a good embedding.