Animation by Matrix Multiplication

DATA 605, Spring 2021: Assignment 1

Philip Tanofsky

07 February 2021

Prompt

For this assignment, build the first letters for both your first and last name using point plots in R.

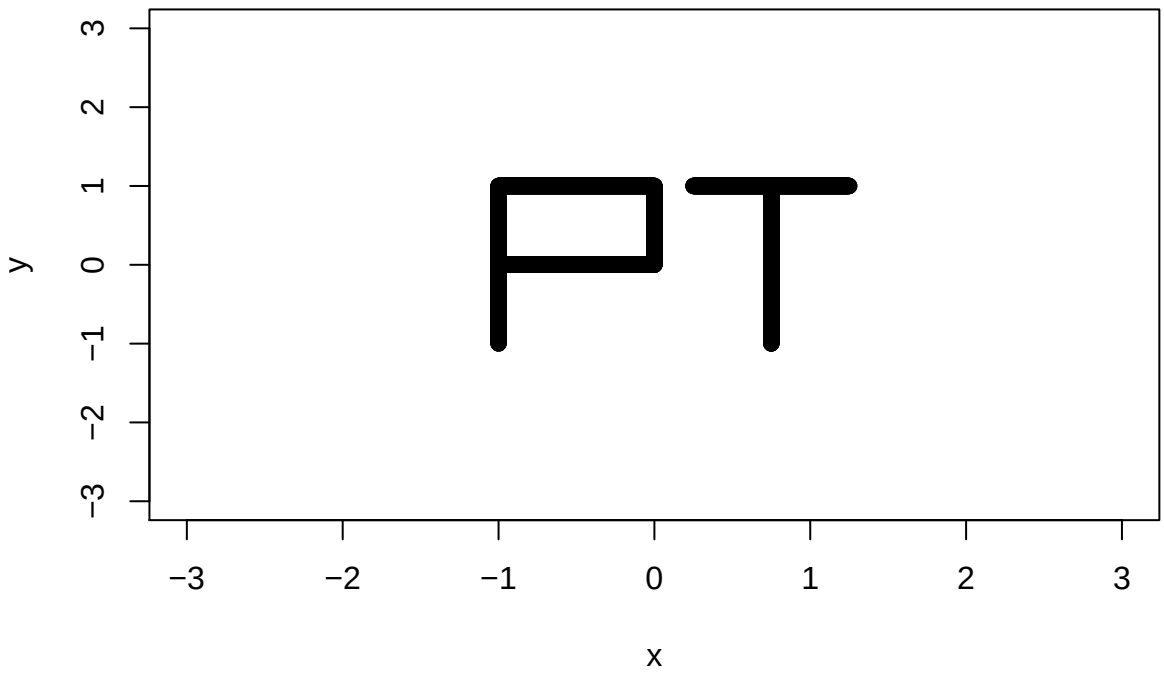
Then, write R code that will left multiply (%>%) a square matrix (x) against each of the vectors of points (y). Initially, that square matrix will be the identity matrix.

Use a loop that changes the transformation matrix incrementally to demonstrate 1) shear, 2) scaling, 3) rotation, and 4) projection in animated fashion.

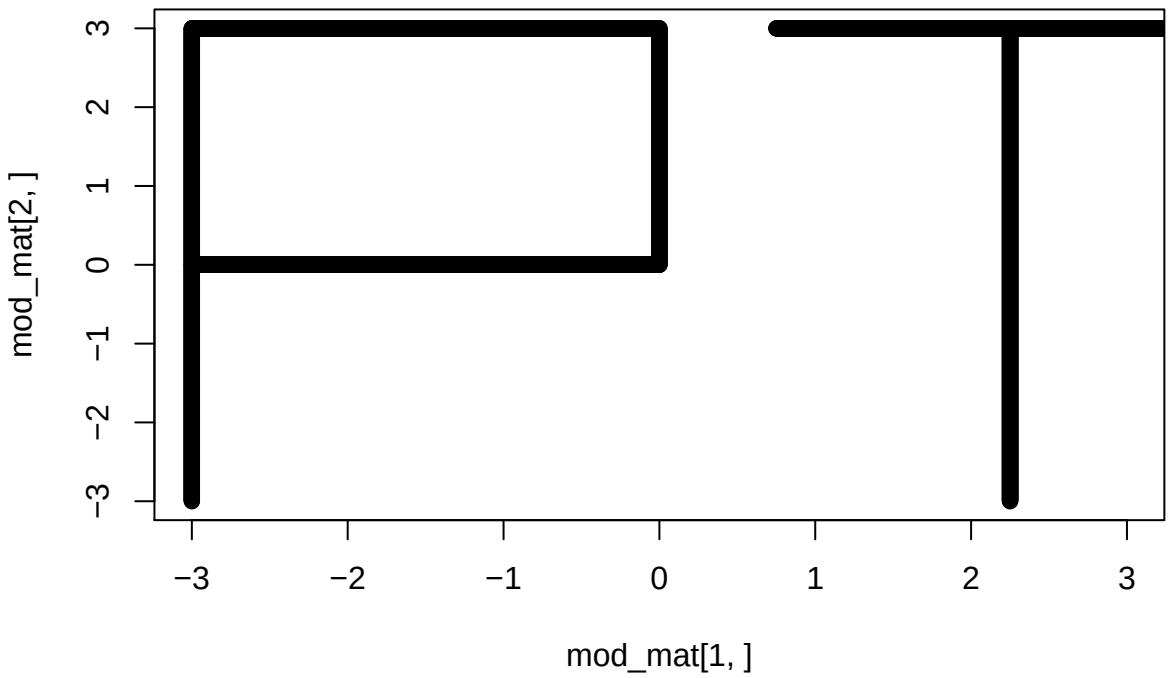
Solution

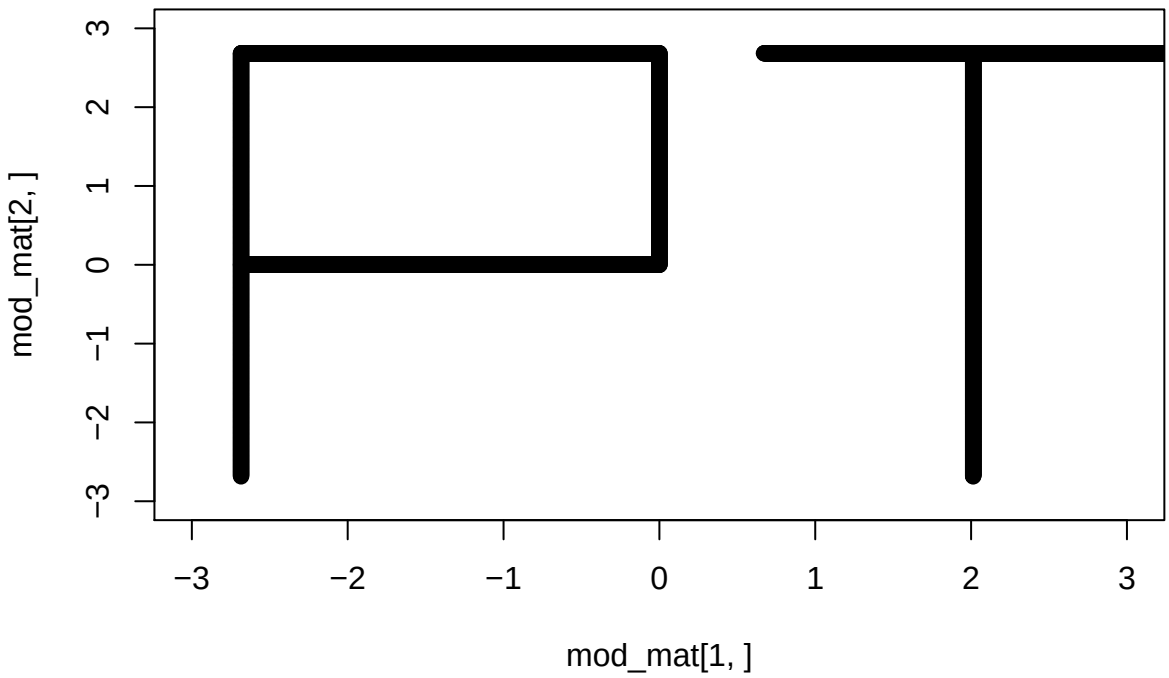
Based on PDF output, code is located on page 1, page 2, page 22, page 42, page 62, page 82.

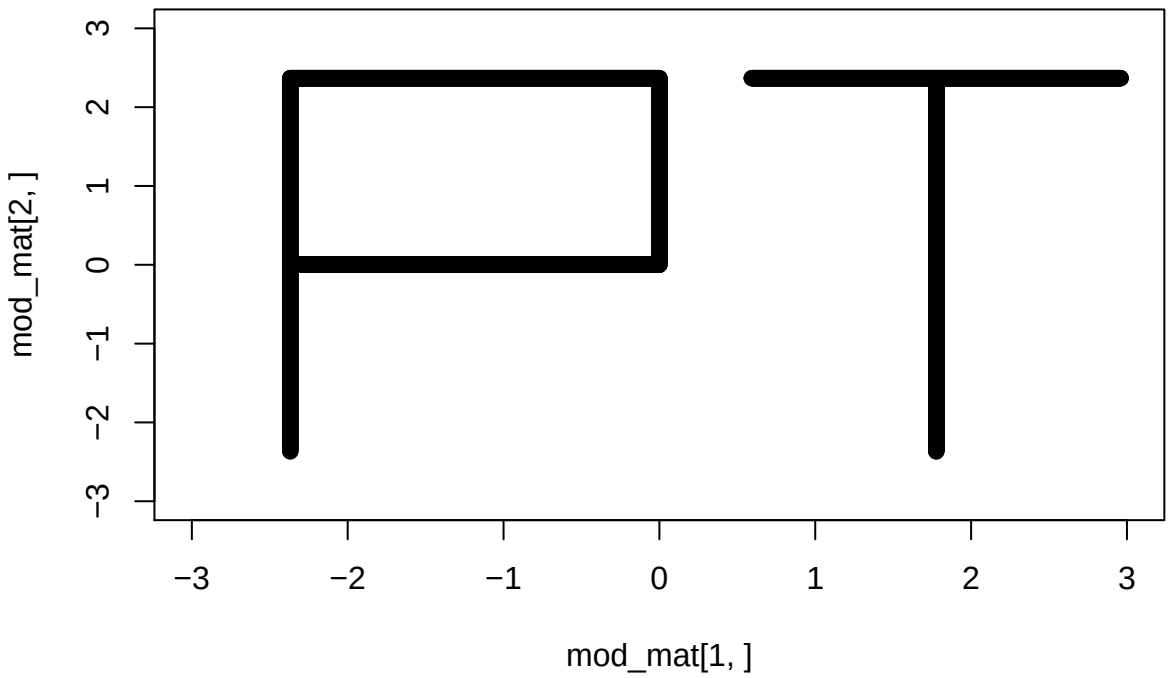
```
library(animation)

# PT: Initials created by plot points
x=c(rep(-1,500), seq(-1,0,length.out=1000), seq(-1,0,length.out=1000),
    rep(0,250), seq(0.25,.75,length.out=500), rep(.75,1000), seq(.75,1.25,length.out=500))

y=c(seq(-1,1,length.out=500), rep(0,1000), rep(1,1000), seq(0,1,length.out=250),
    rep(1,500), seq(-1,1,length.out=1000), rep(1,500))

# H (from prompt)
#x=c(rep(0,500), seq(0,1,length.out=1000), rep(1,500))
#y=c(seq(-1,1,length.out=500), rep(0,1000), seq(-1,1,length.out=500))
z=rbind(x,y)

# Confirm the 'PT' is showing based on prompt code
plot(y~x, xlim=c(-3,3), ylim=c(-3,3))
```

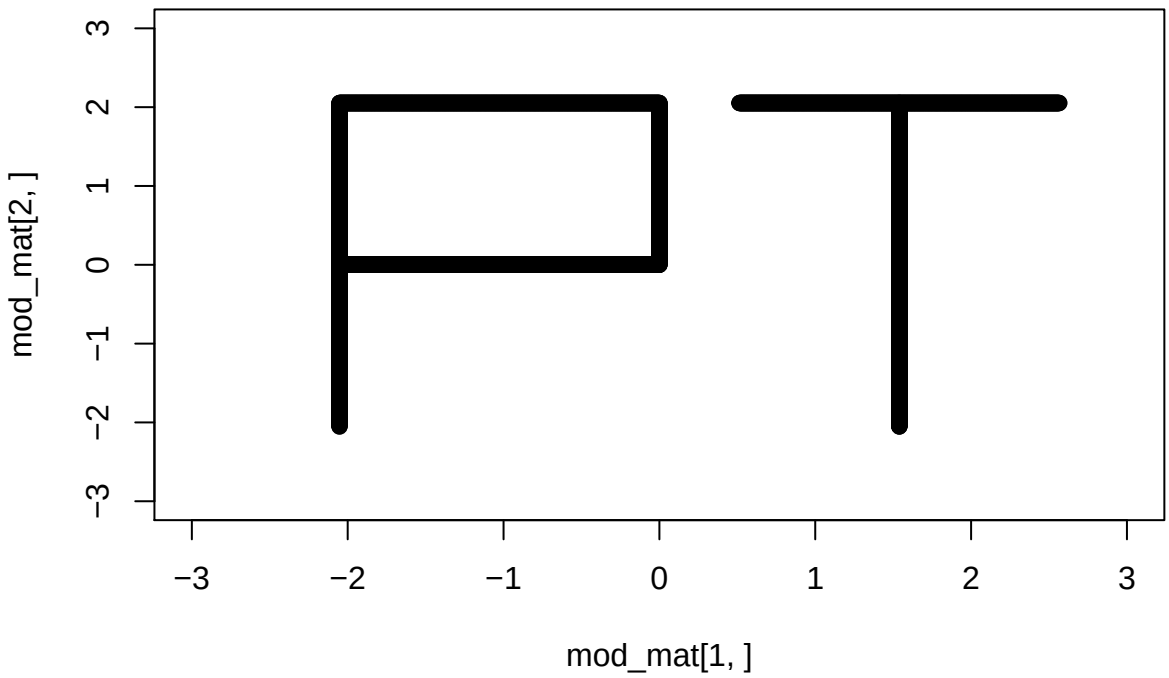


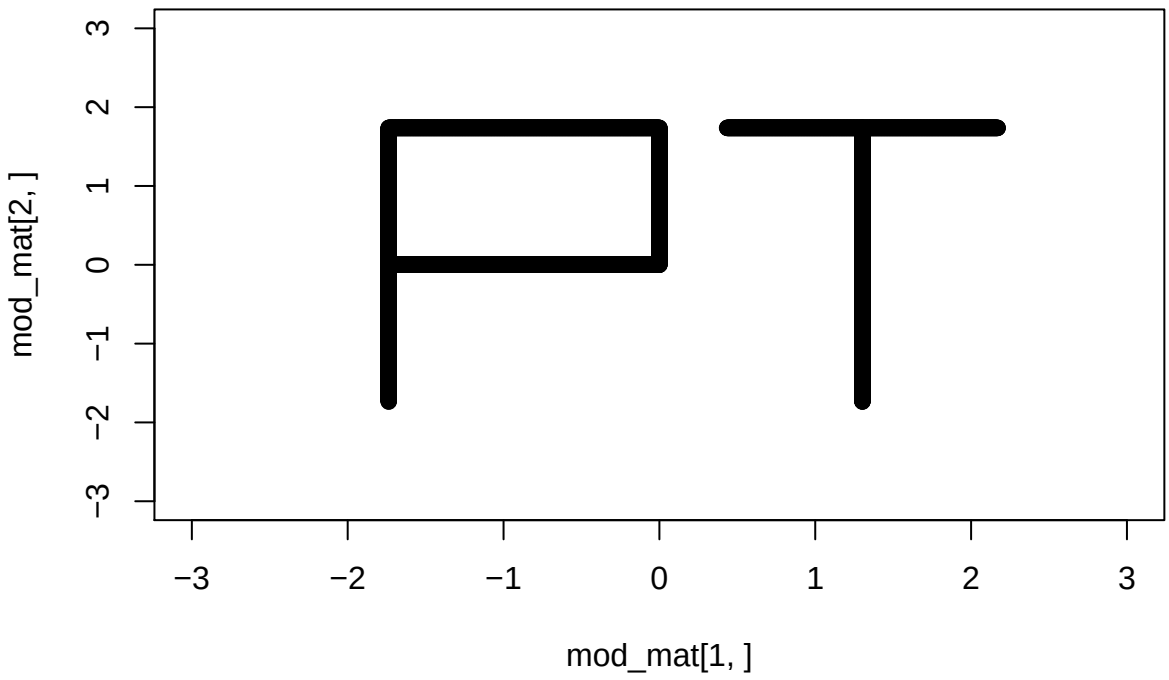

```
dev.control('enable')

anim = ani.record(reset = T, replay.cur = F)

# Create display window for animation
x11()

# Create 2x2 identity matrix
a=diag(2)
#----------------------
# Scale animation
#----------------------
for (i in seq(3,-3, length.out=20)) {
  a[1,1] = i
  a[2,2] = i
  mod_mat = apply(z, 2, function(x) a%*%x)
  plot(mod_mat[2,]~mod_mat[1,], xlim=c(-3,3), ylim=c(-3,3))
  ani.record()
}
```

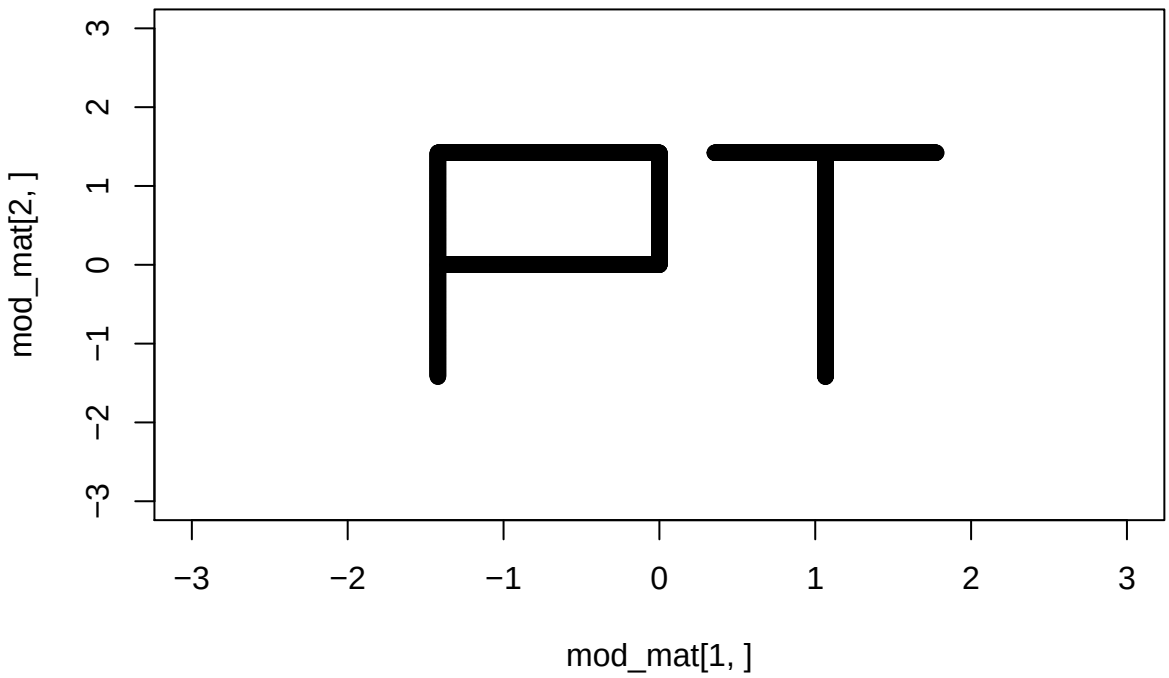

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

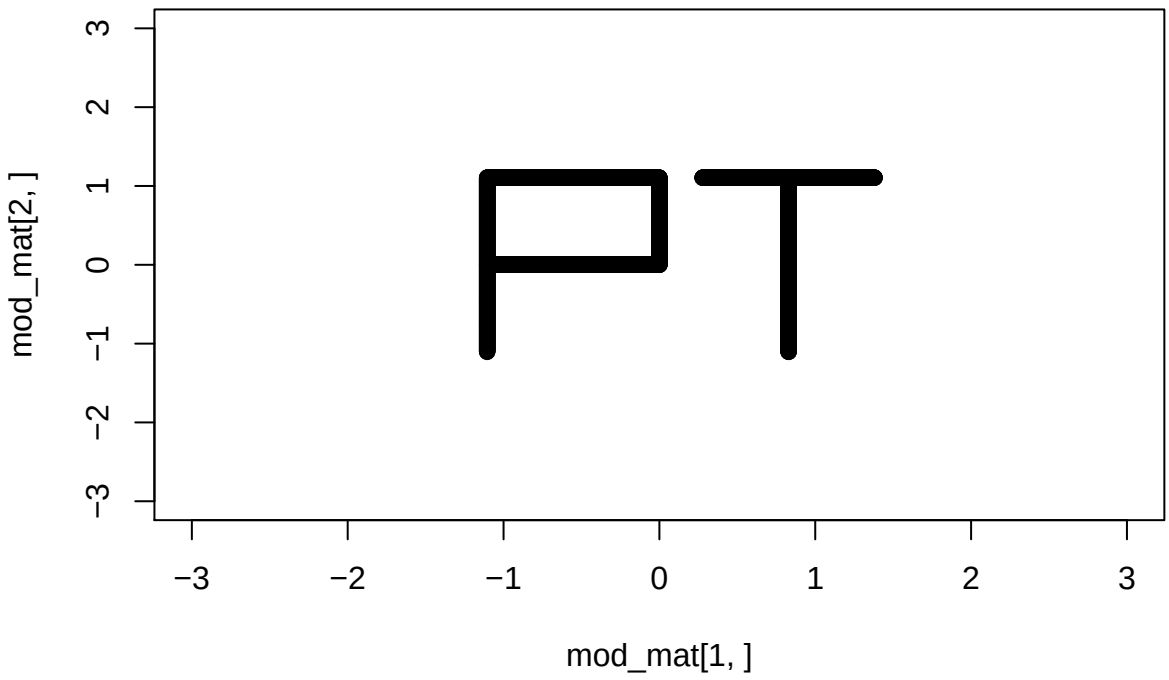

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

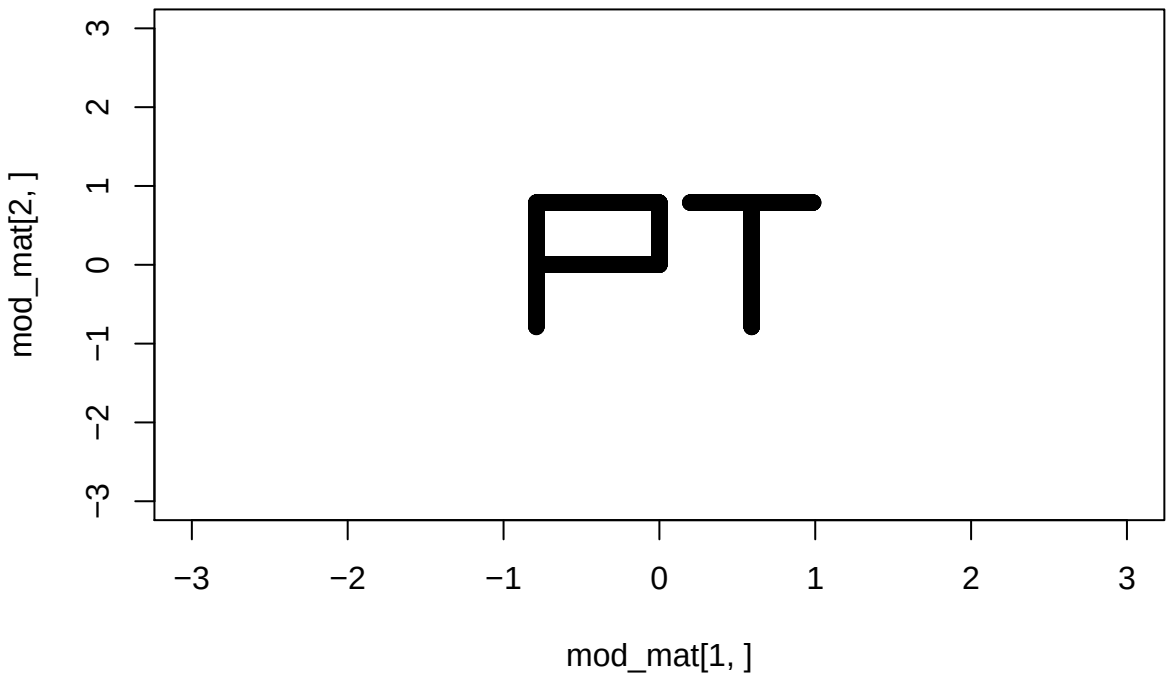

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

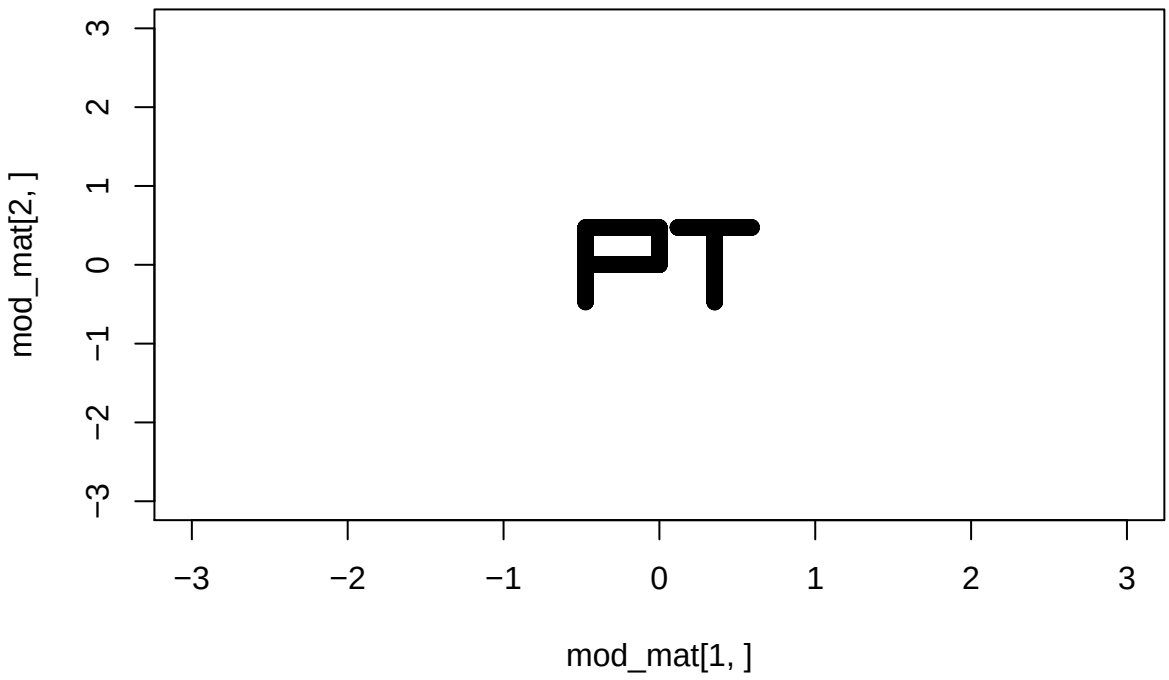

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

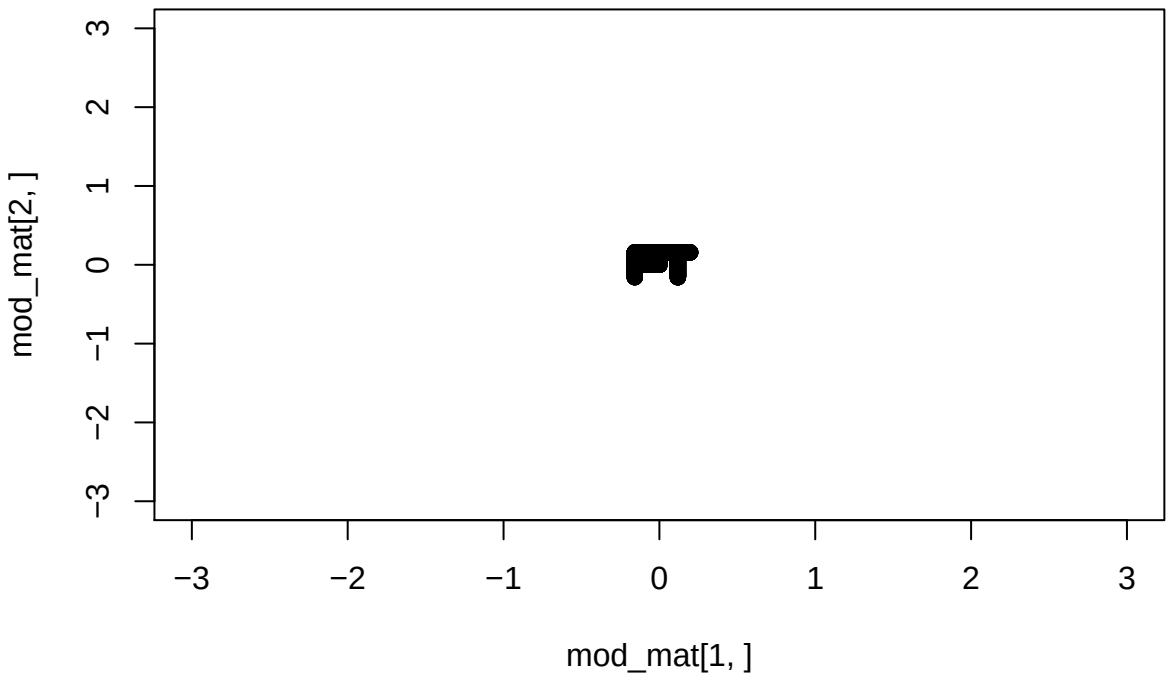

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

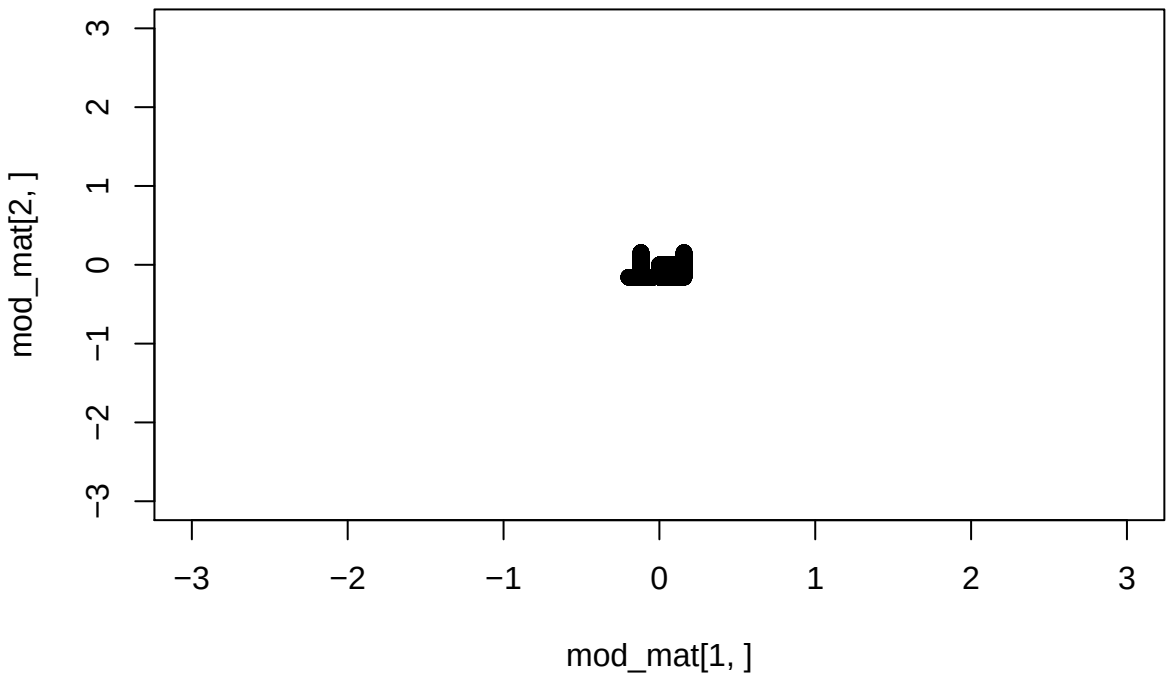

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

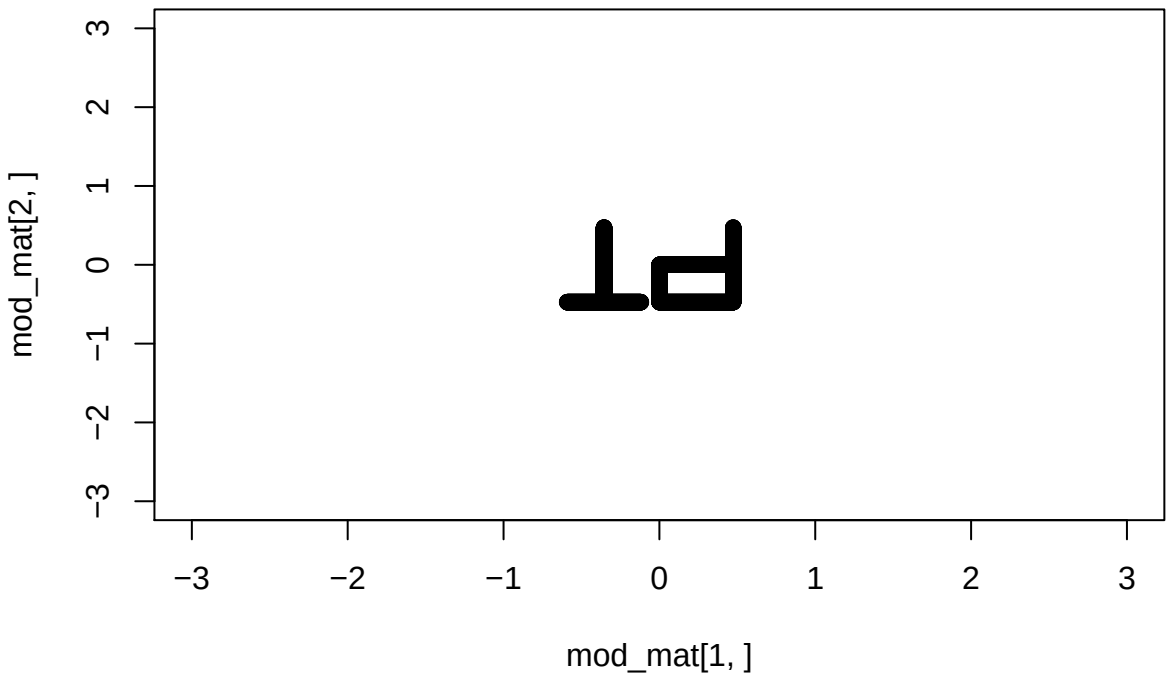

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

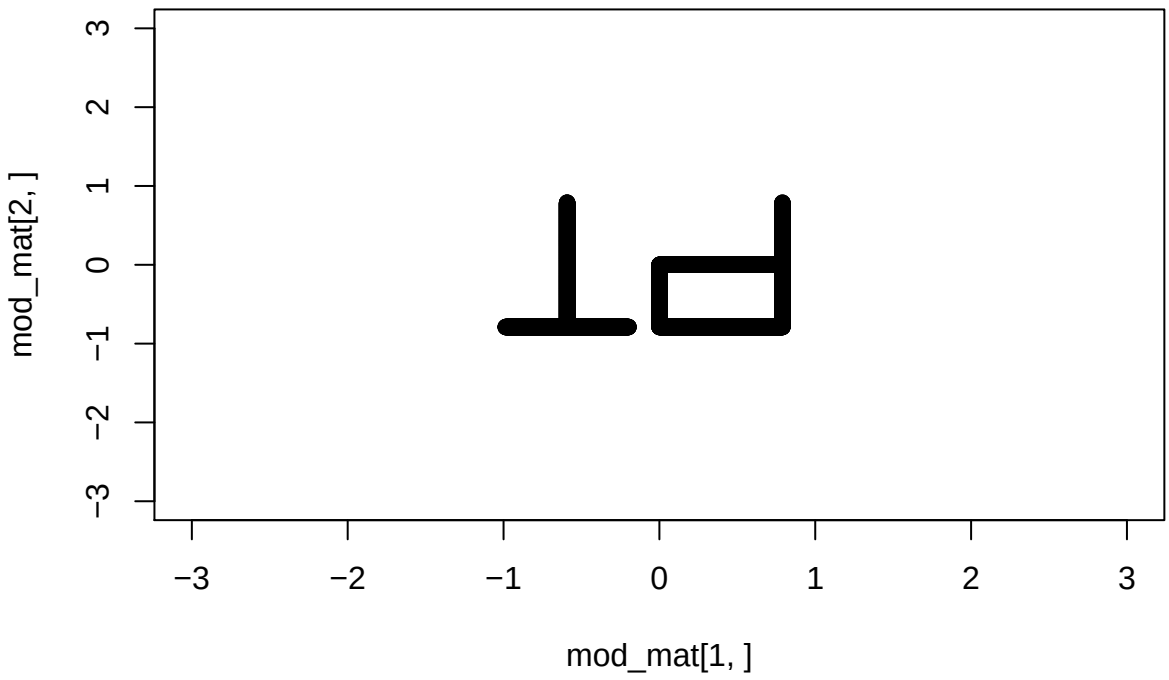

PT
mod_mat[2,]
mod_mat[1,]
−3
−2
−1
0
1
2
3

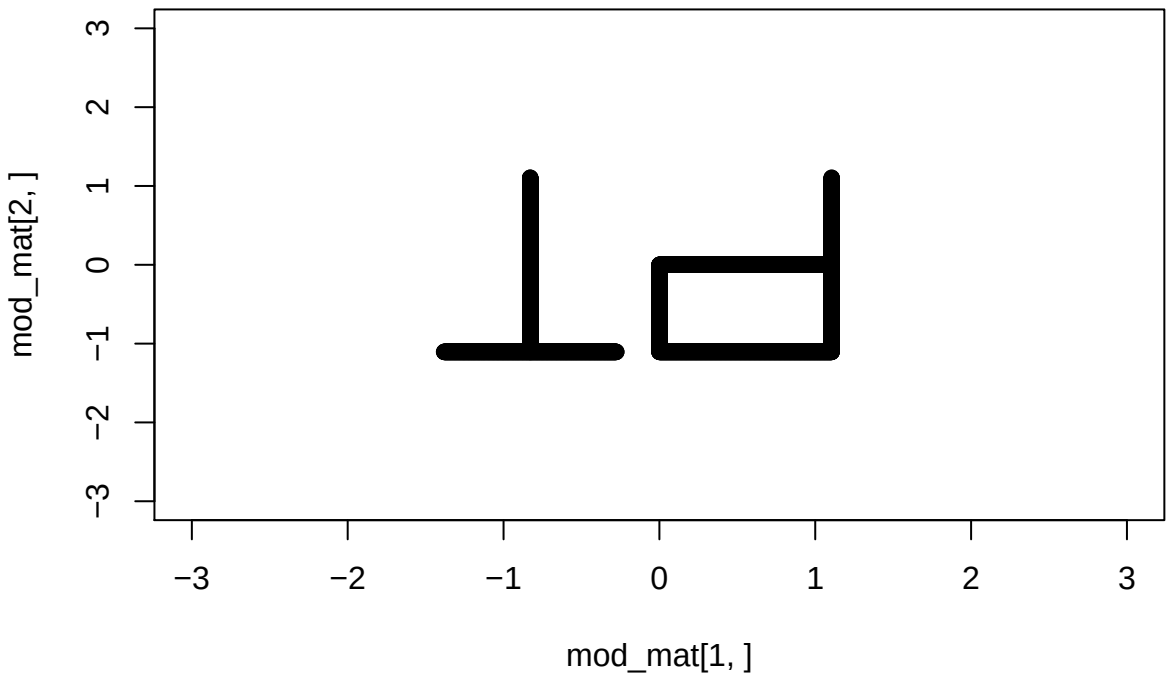

PT
mod_mat[2,]
mod_mat[1,]
−3
−2
−1
0
1
2
3

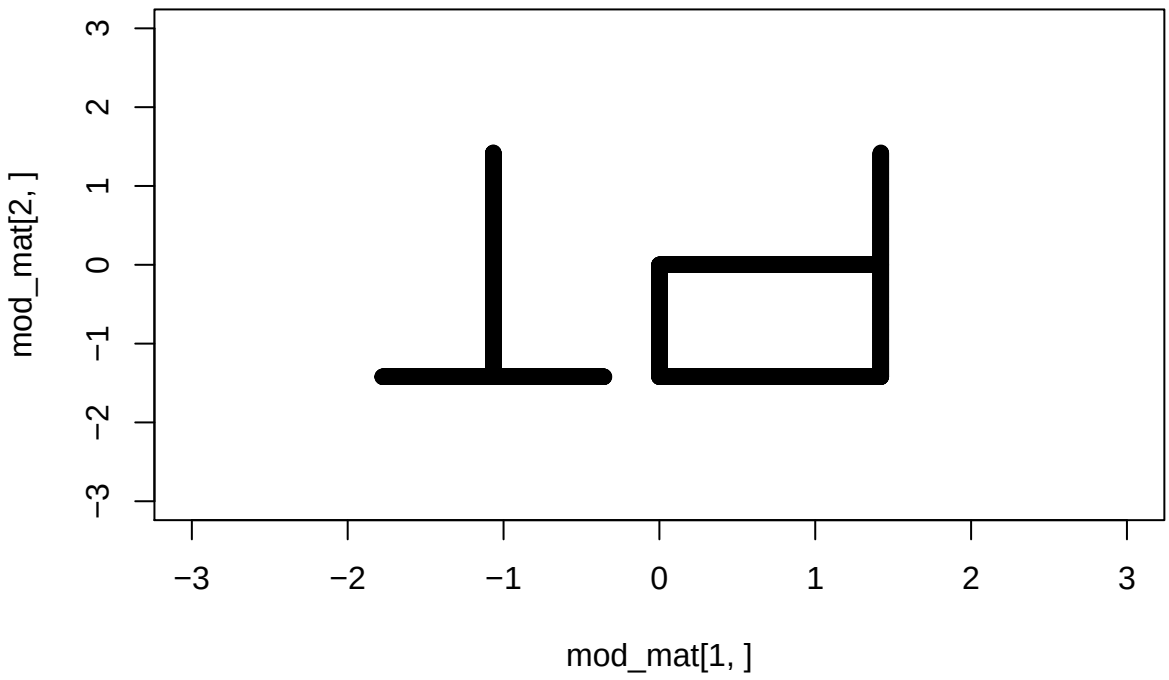

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

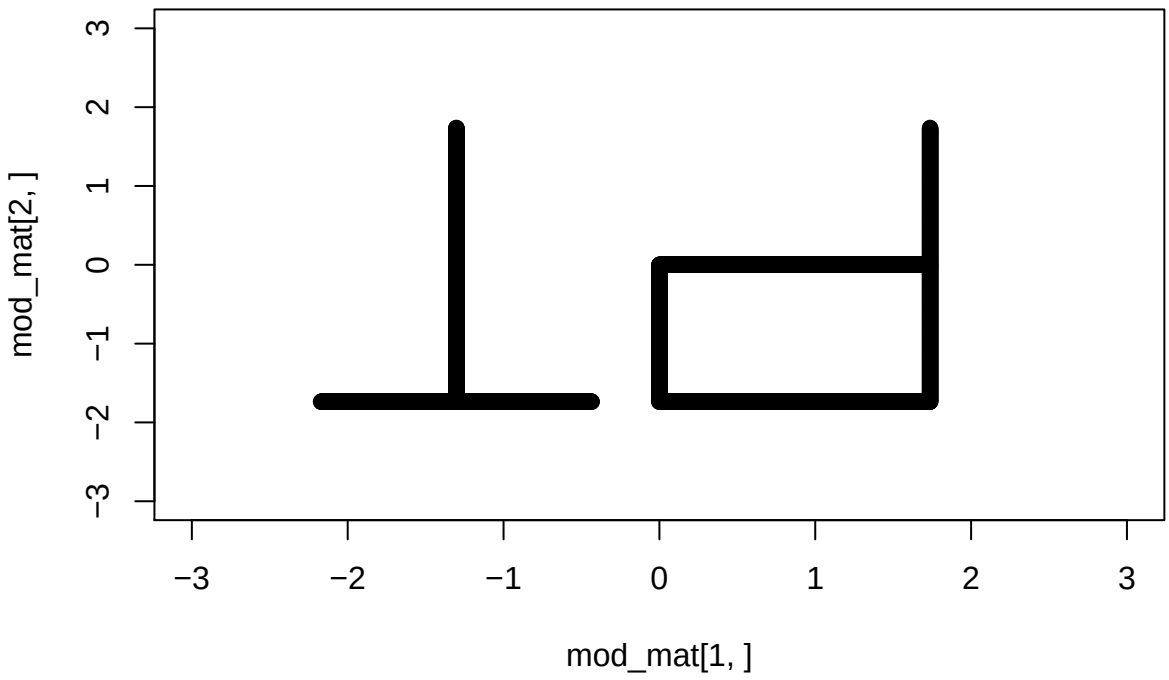

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

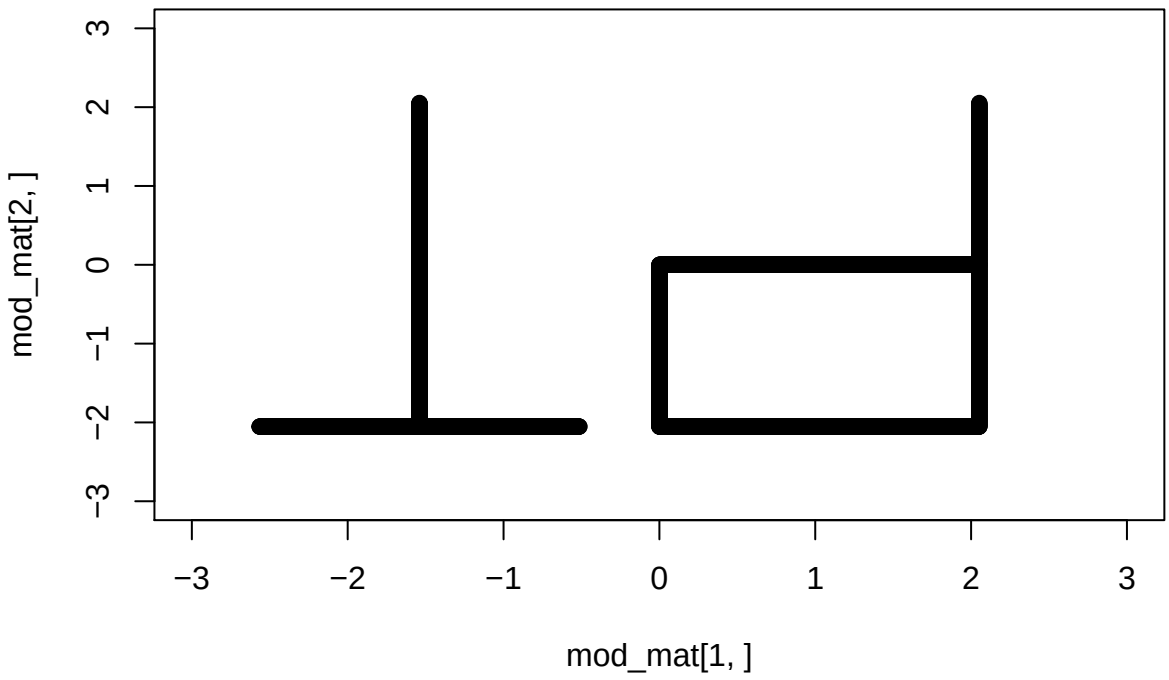

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

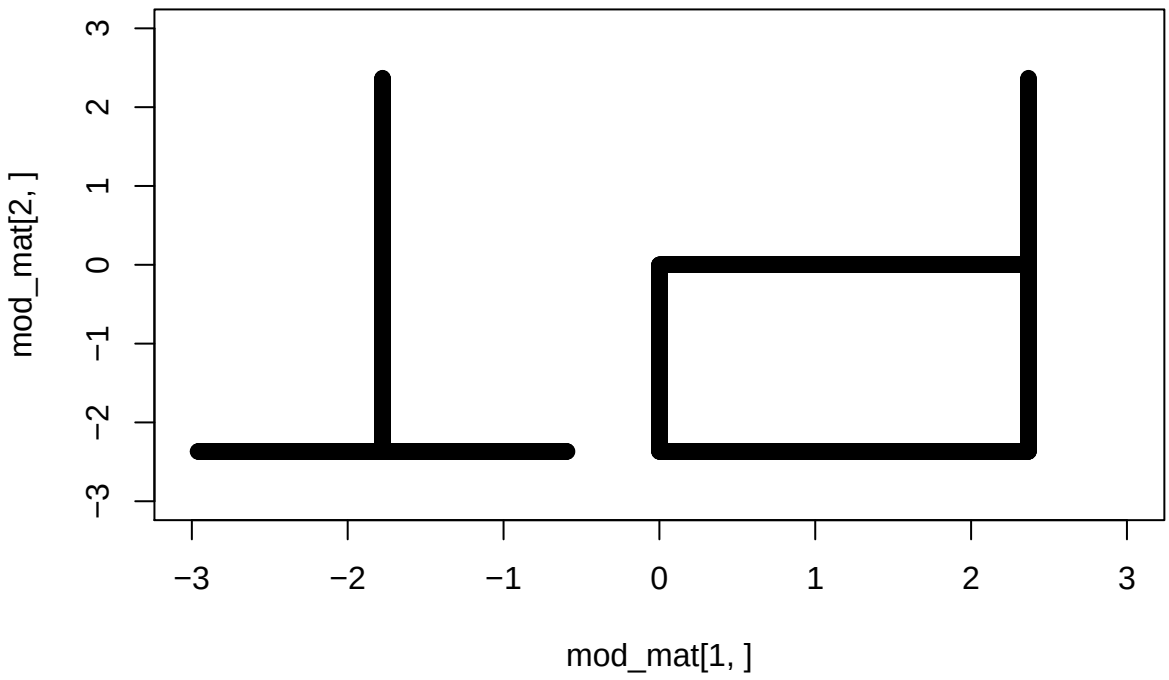

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

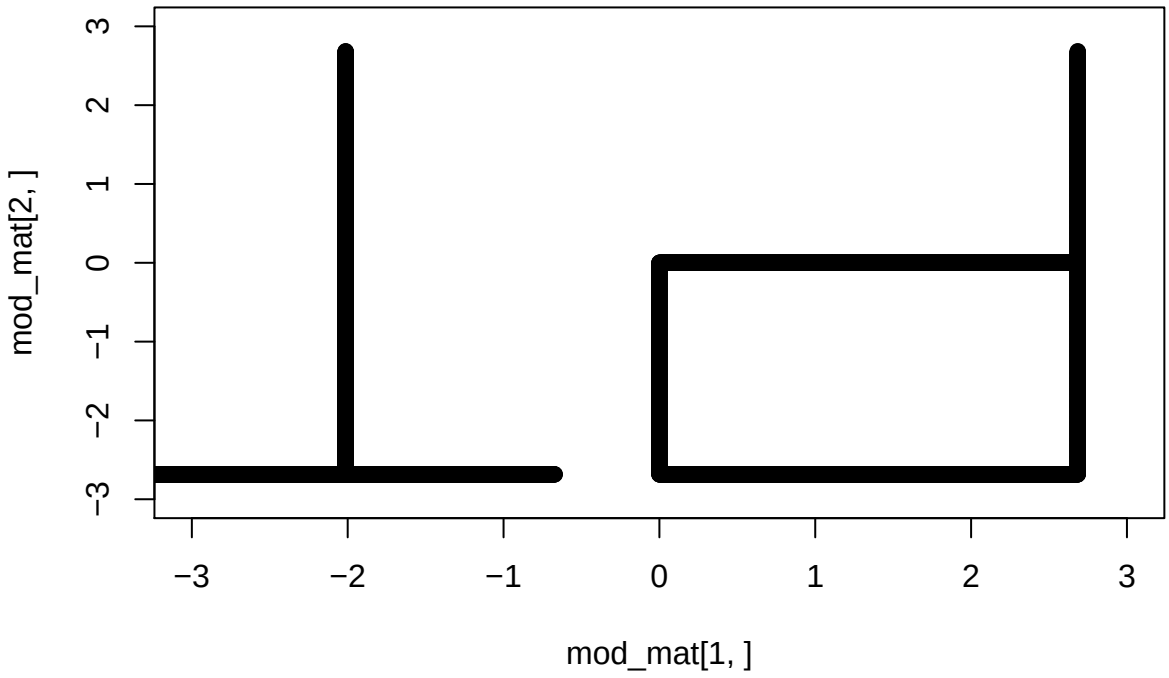

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

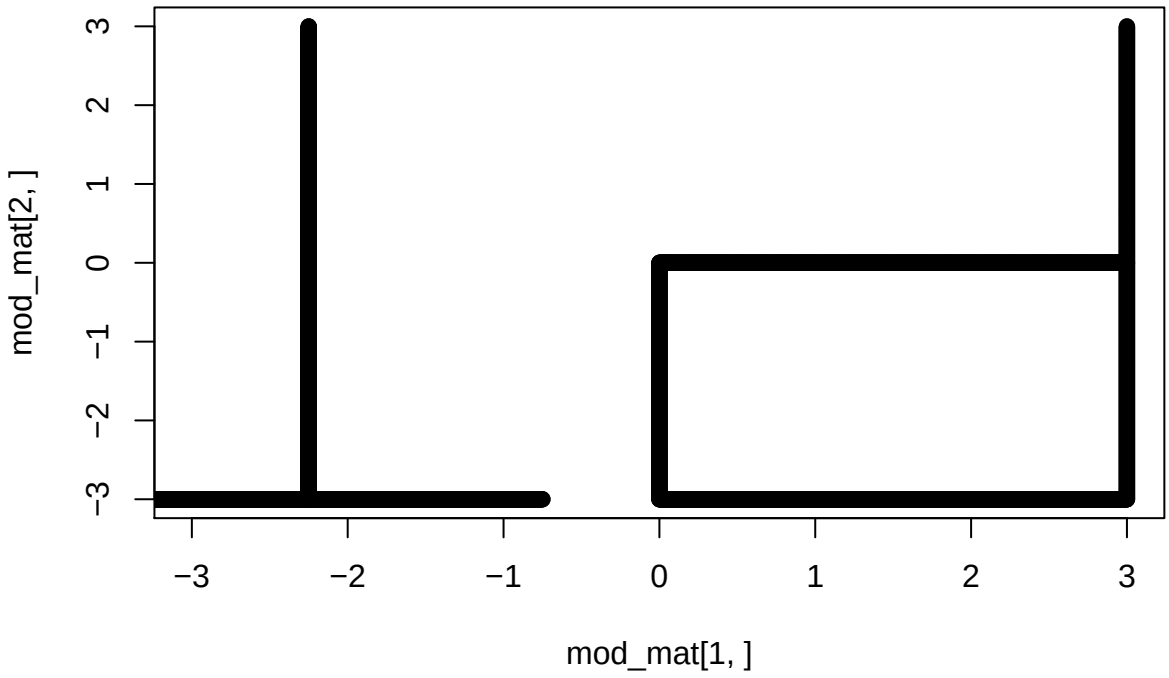

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]


mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

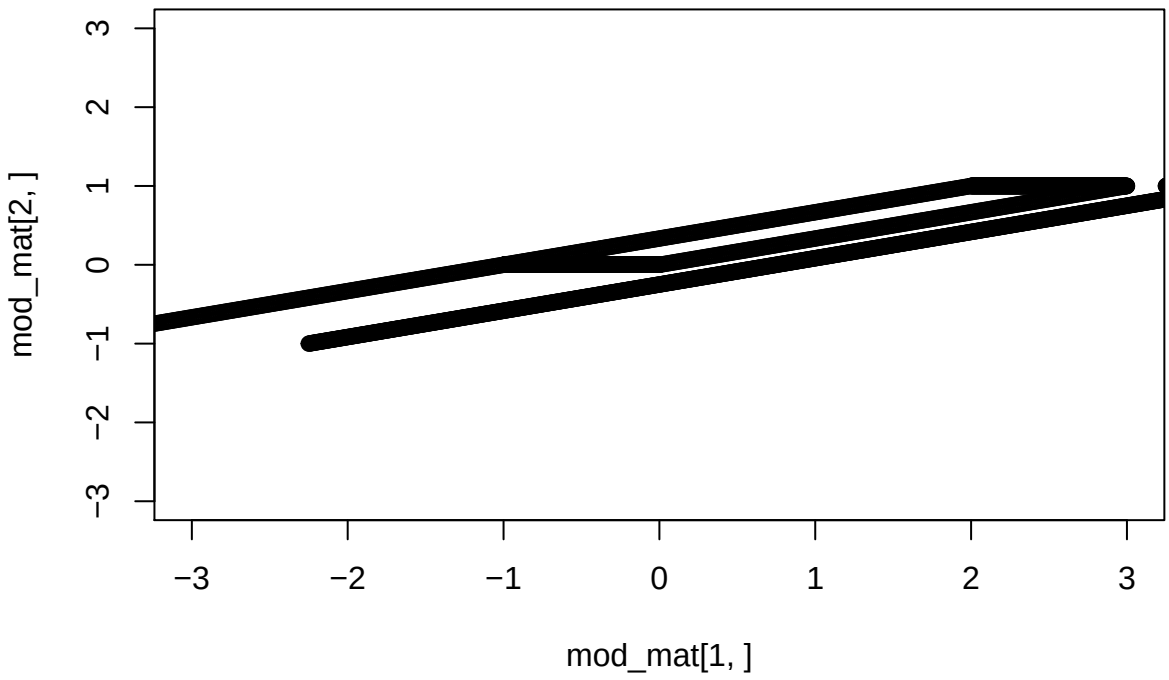

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

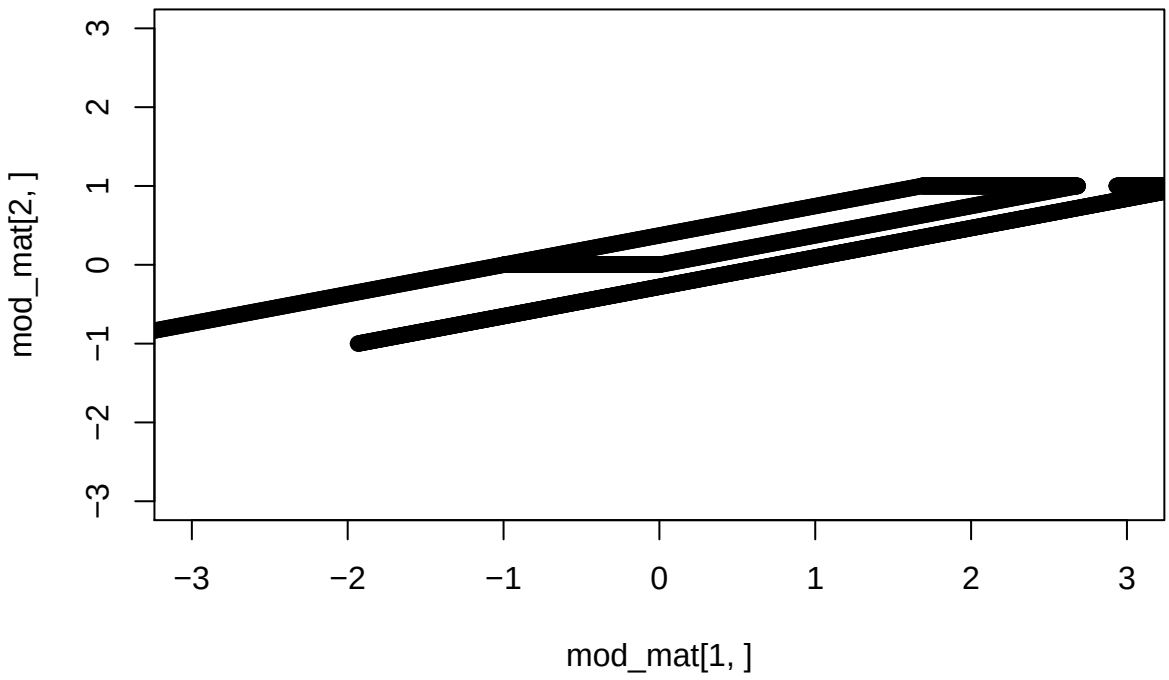

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

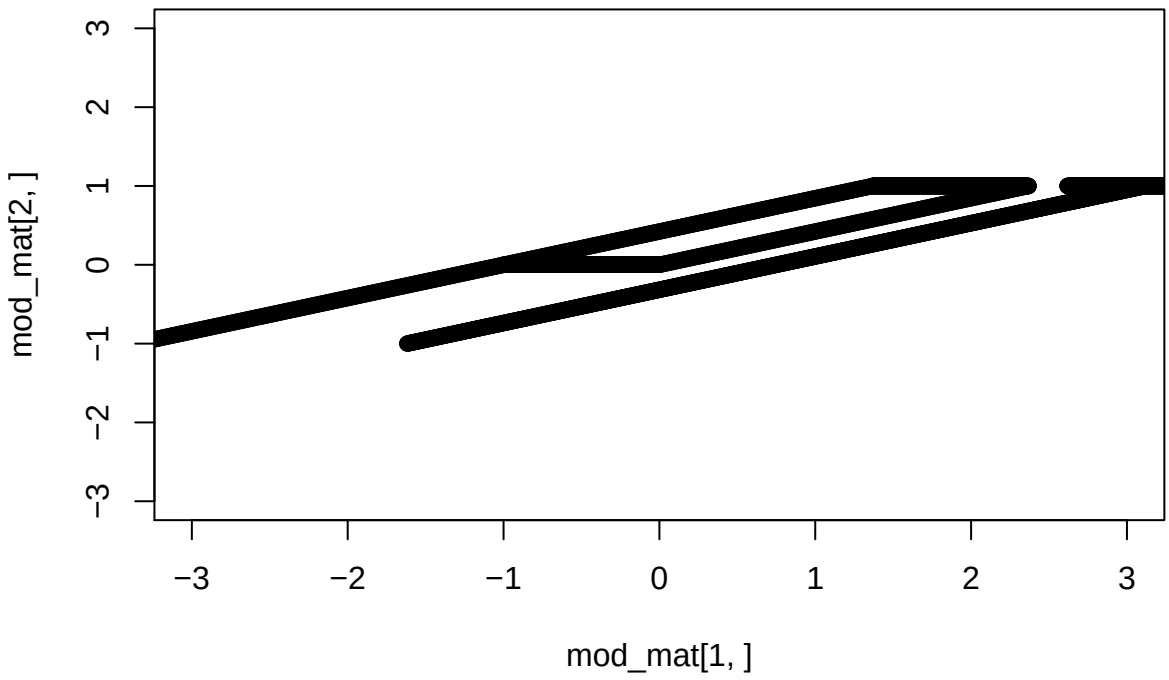

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

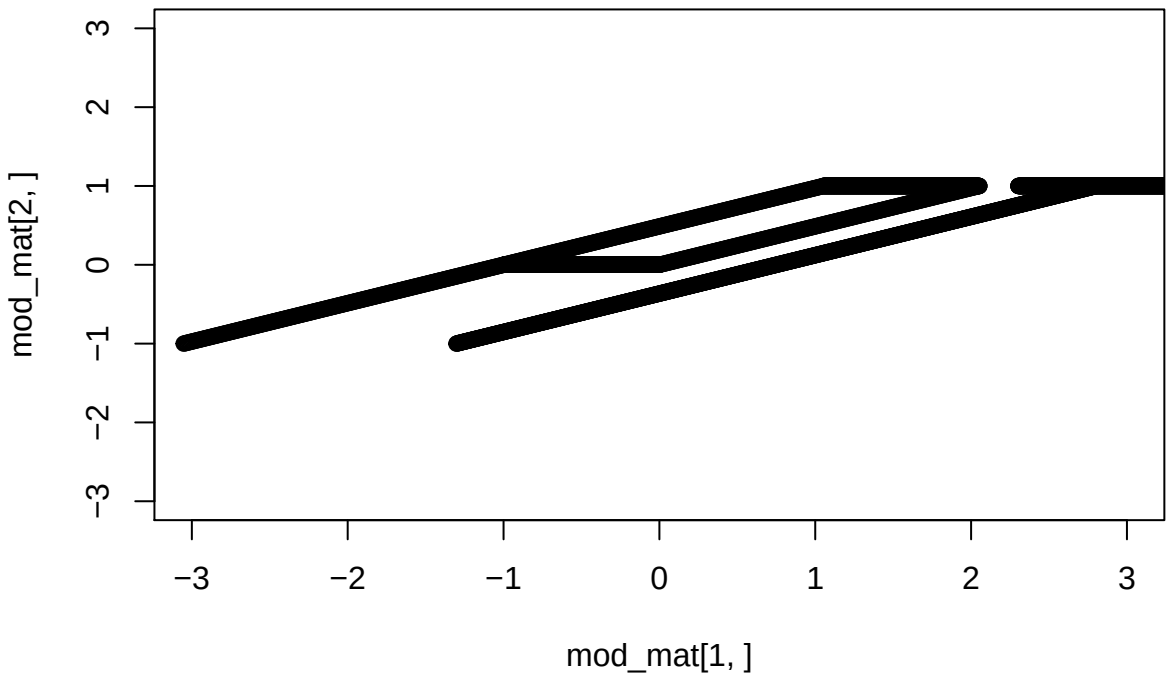


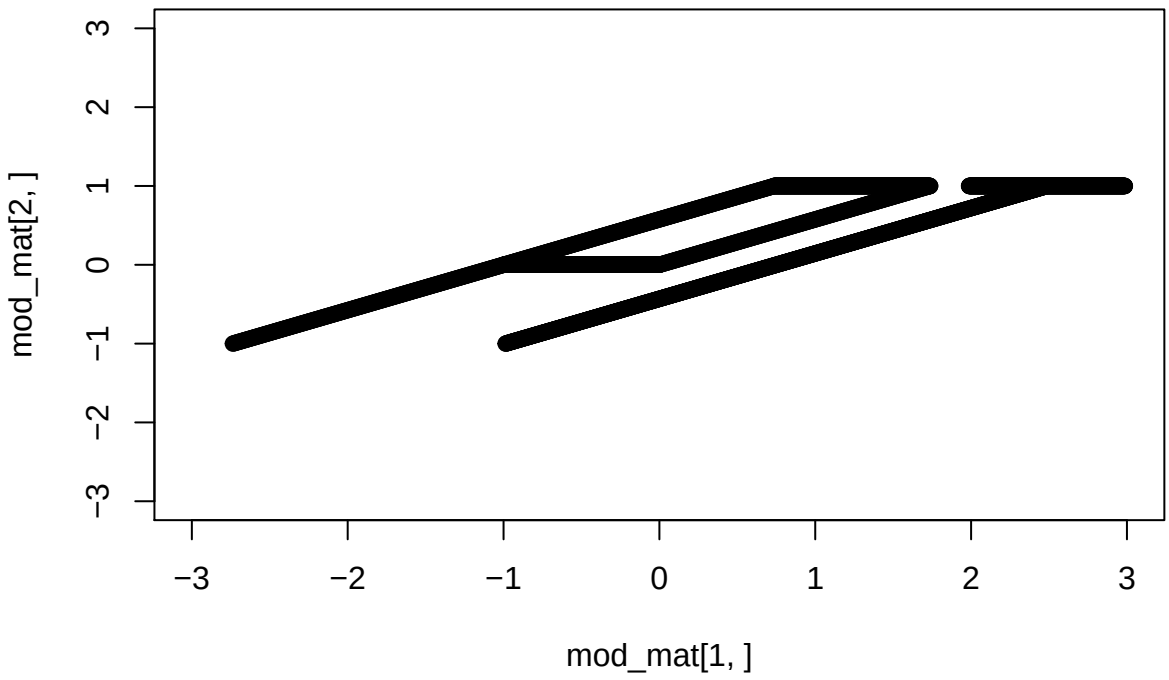

```
# Reset 2x2 identity matrix
a=diag(2)
#----------------------
# Shear
#----------------------
for (i in seq(3,-3, length.out=20)) {
  a[1,2] = i
  mod_mat = apply(z, 2, function(x) a%*%x)
  plot(mod_mat[2,]~mod_mat[1,], xlim=c(-3,3), ylim=c(-3,3))
  ani.record()
}
```

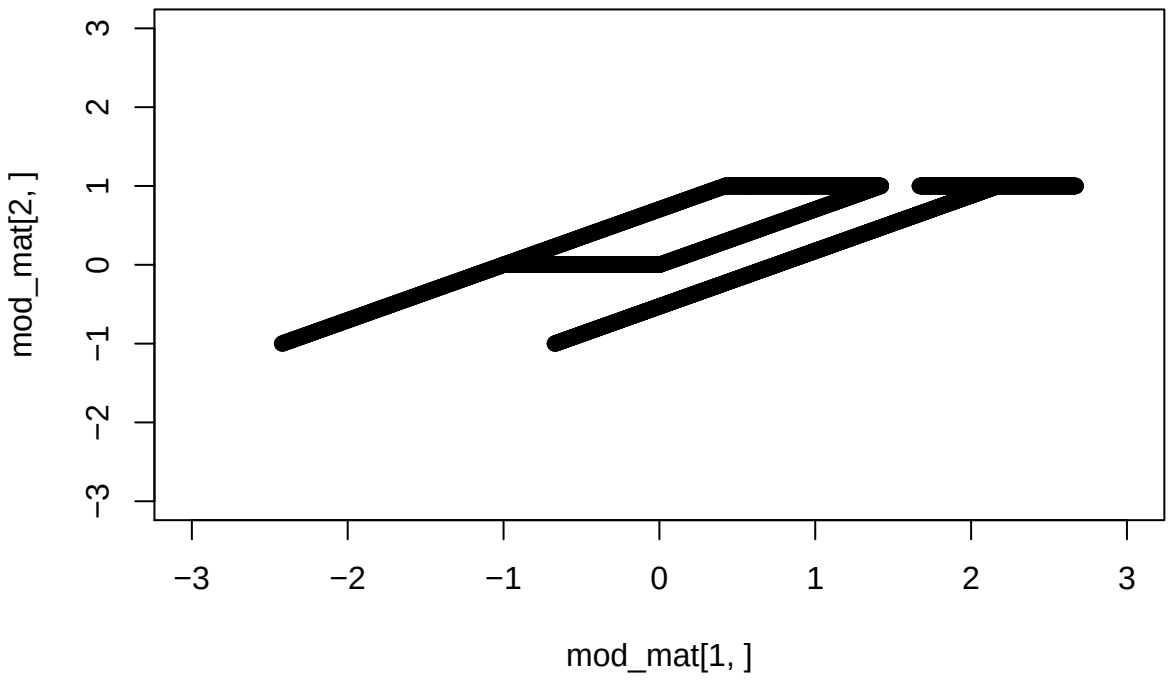

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

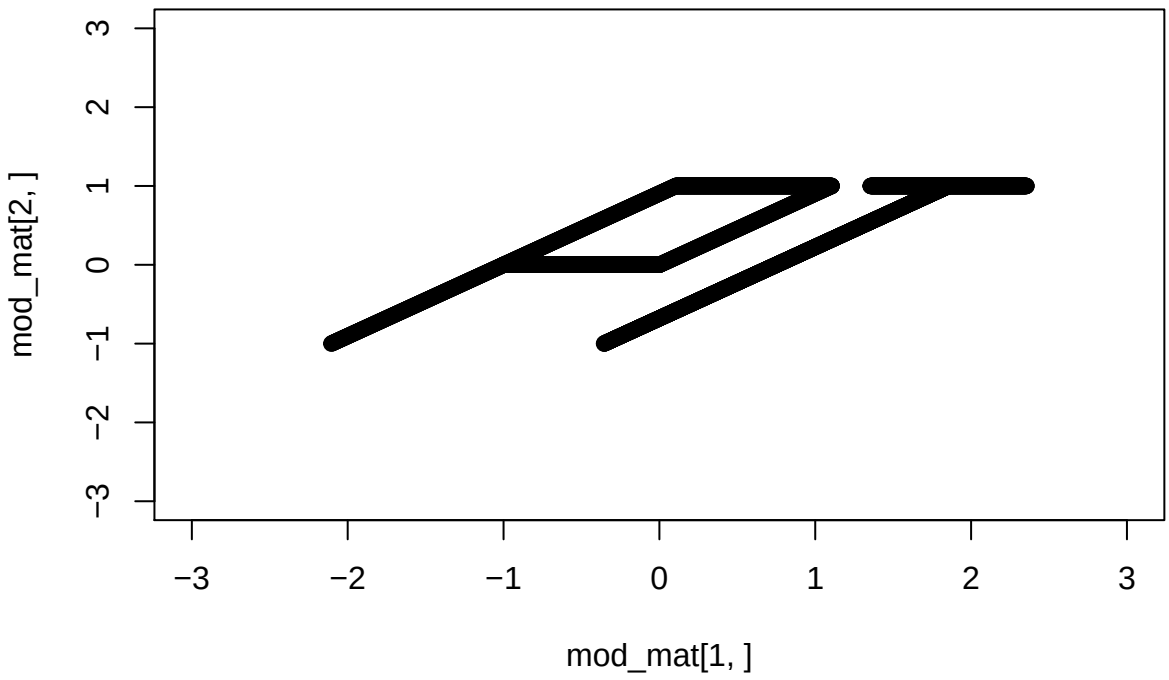

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

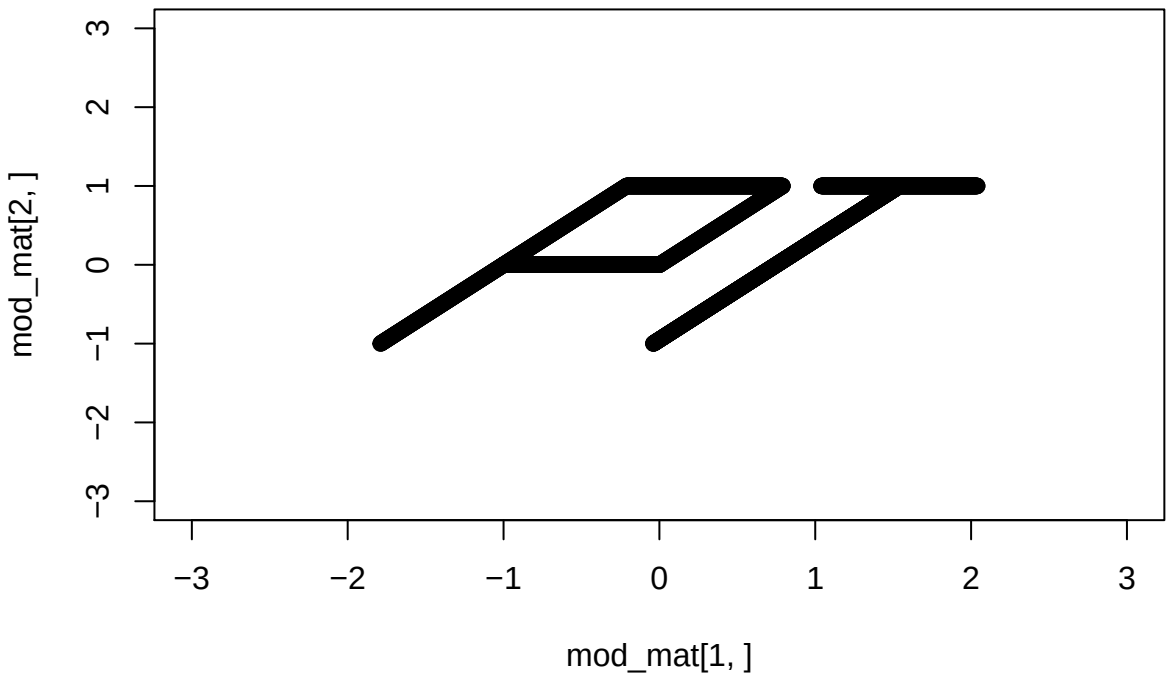

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

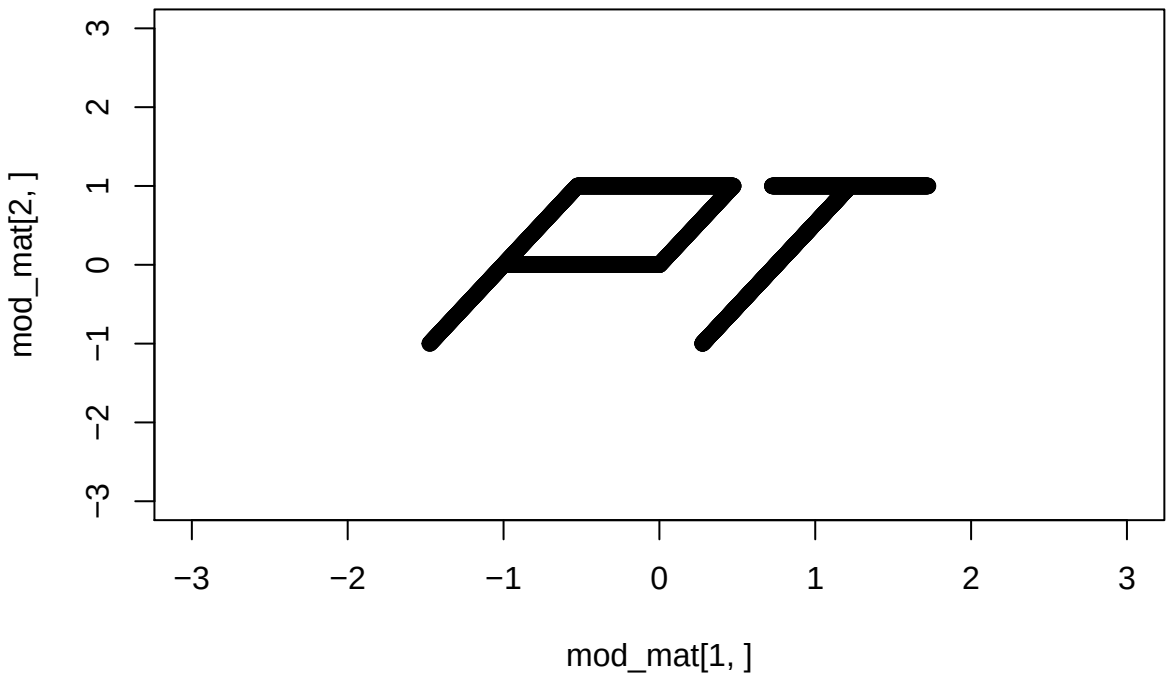

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

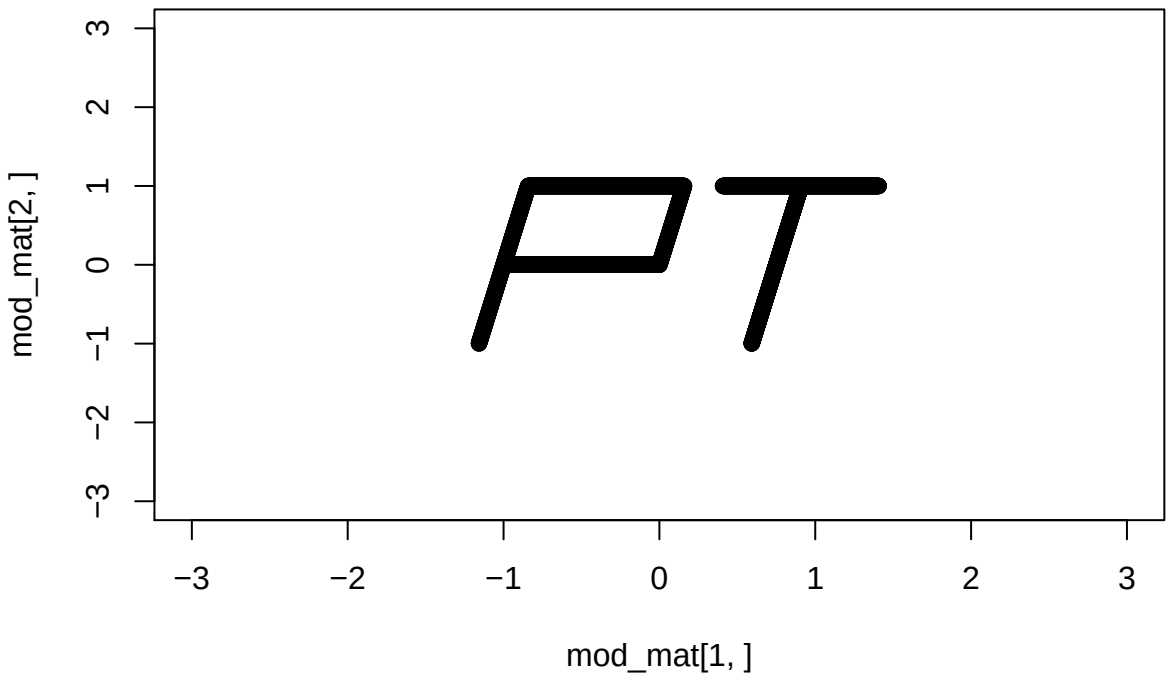

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

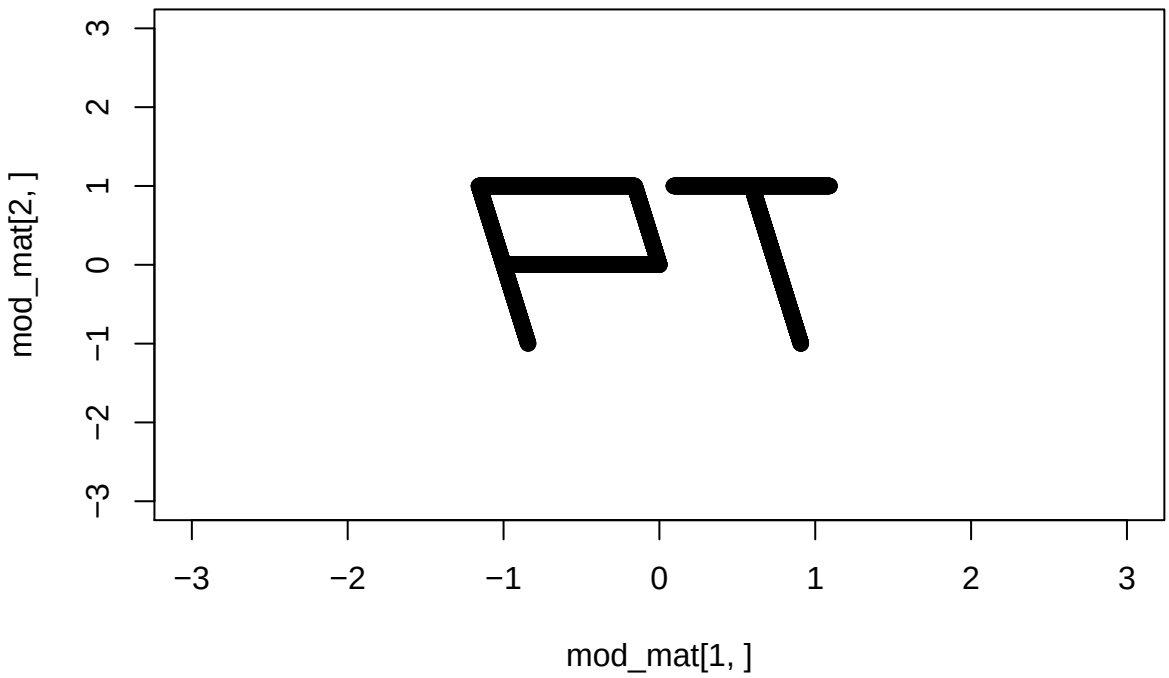

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

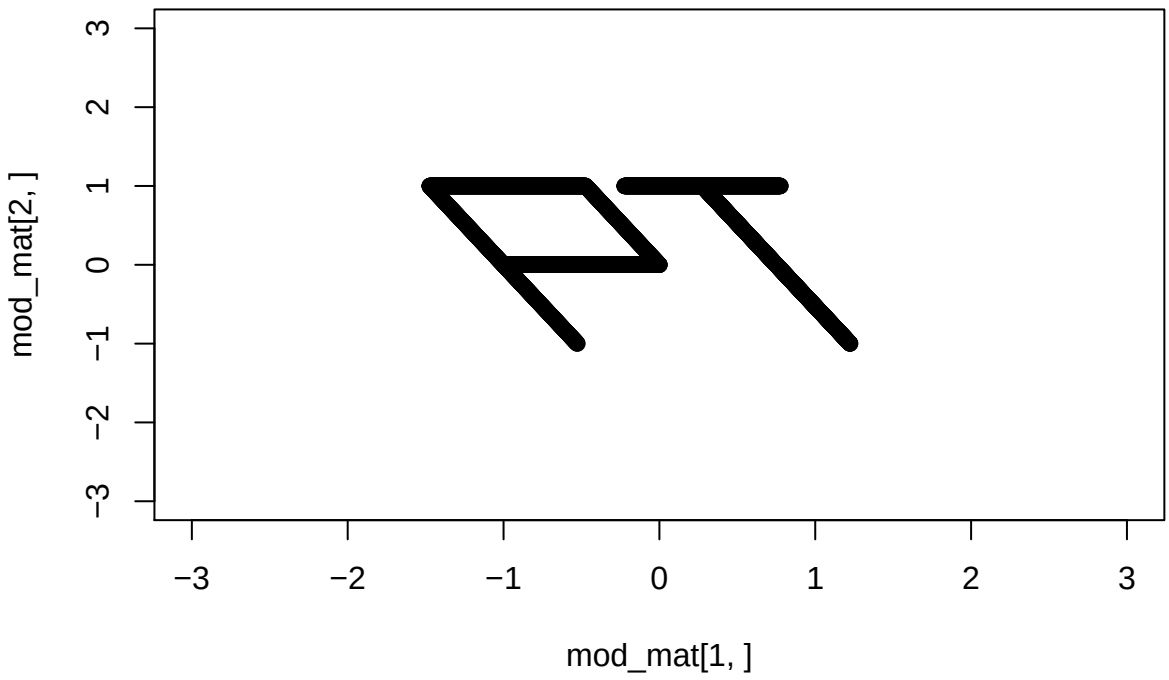

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

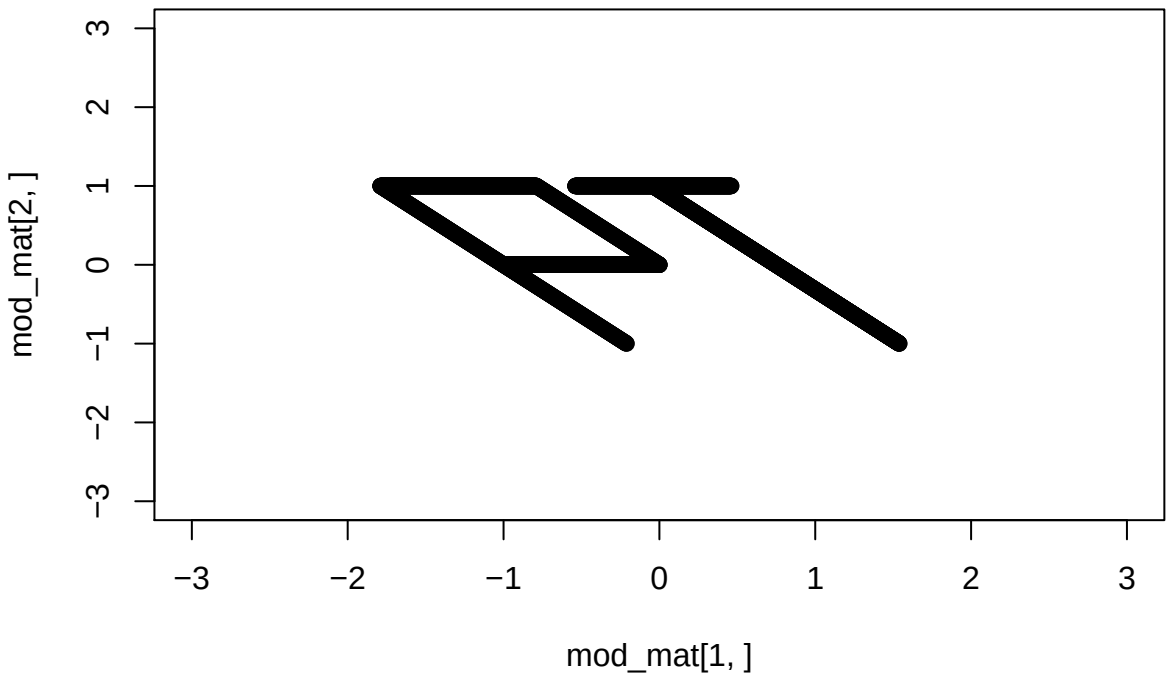

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

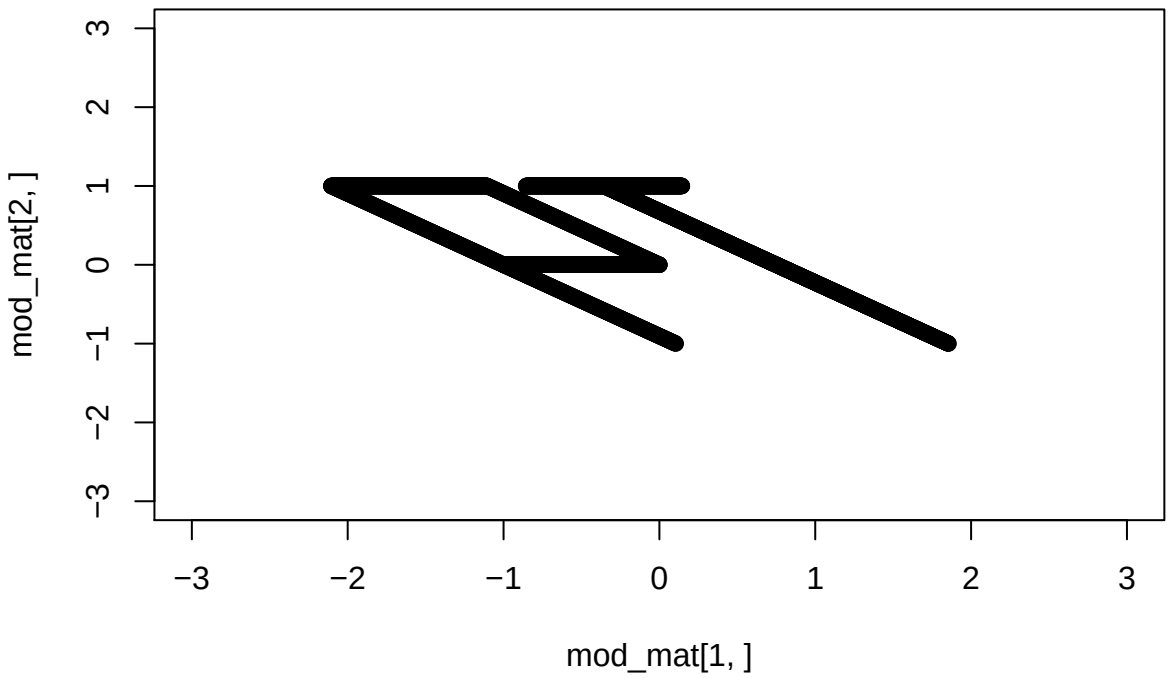

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

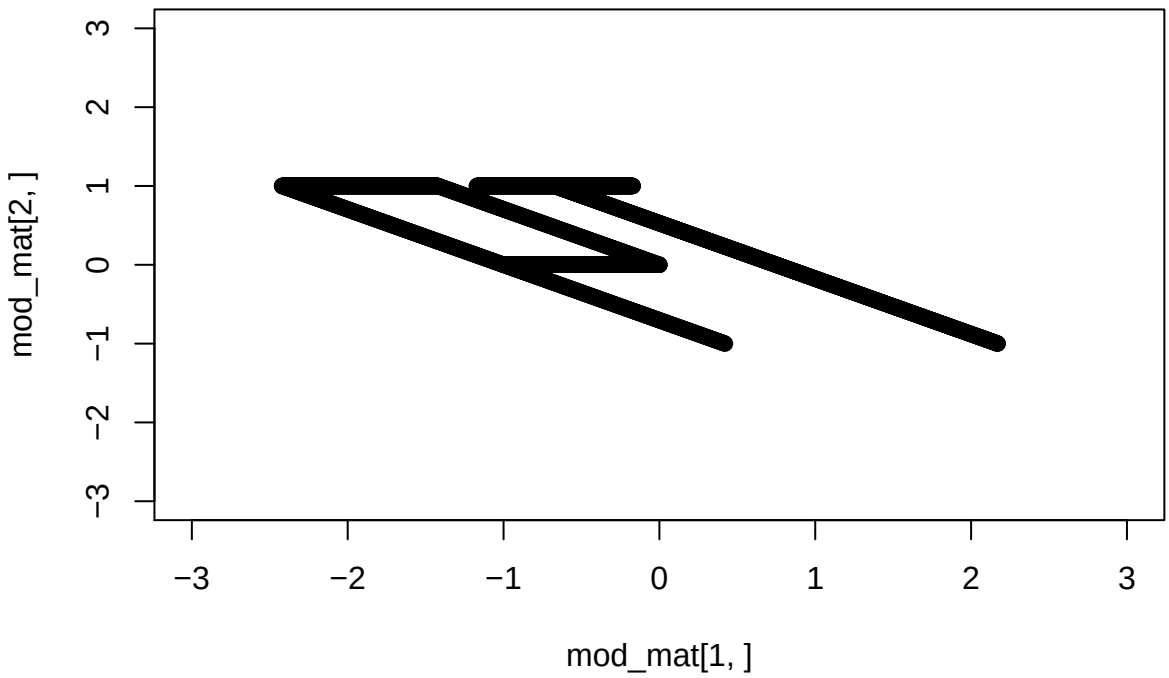

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

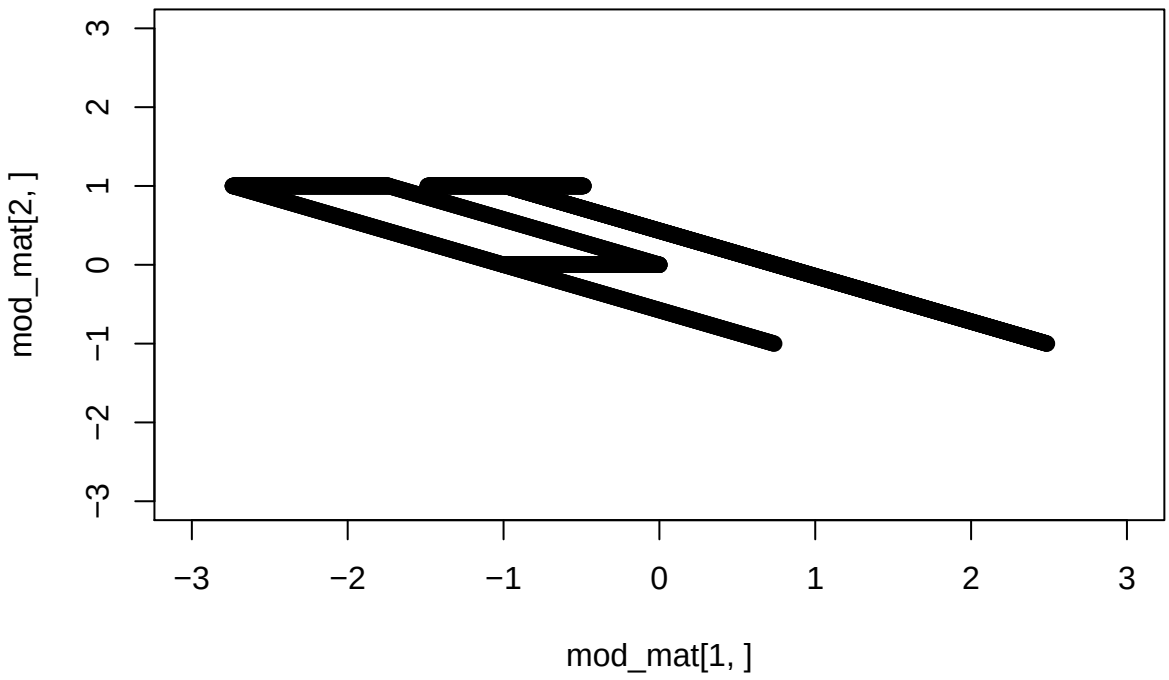

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

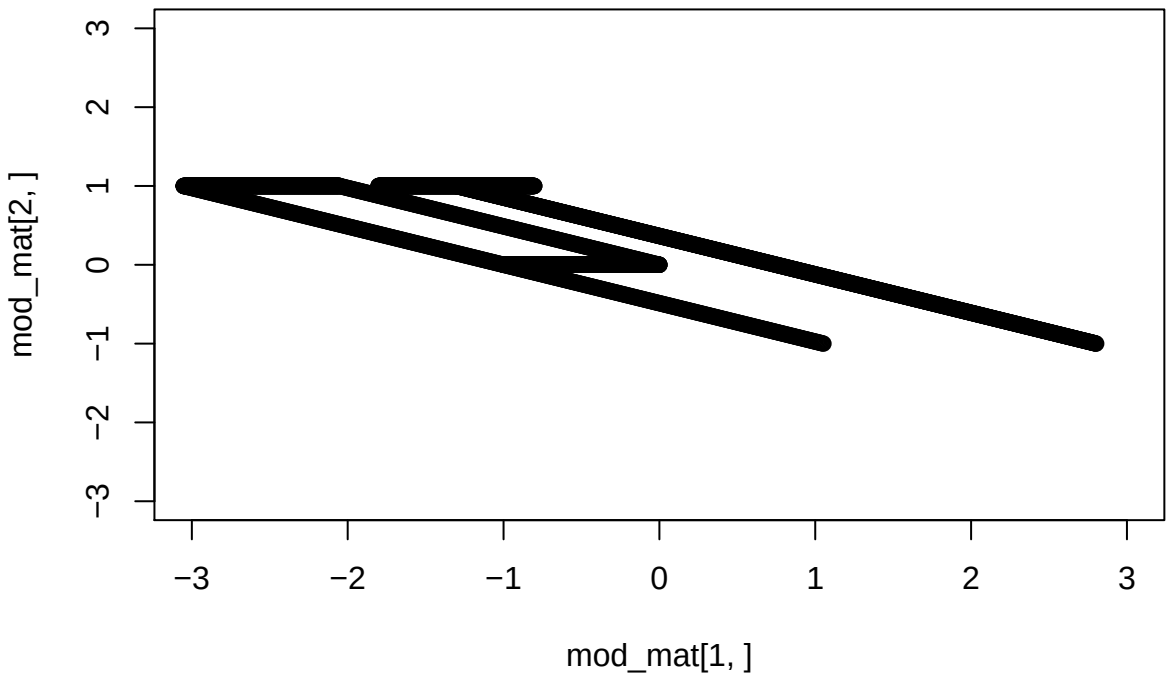

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

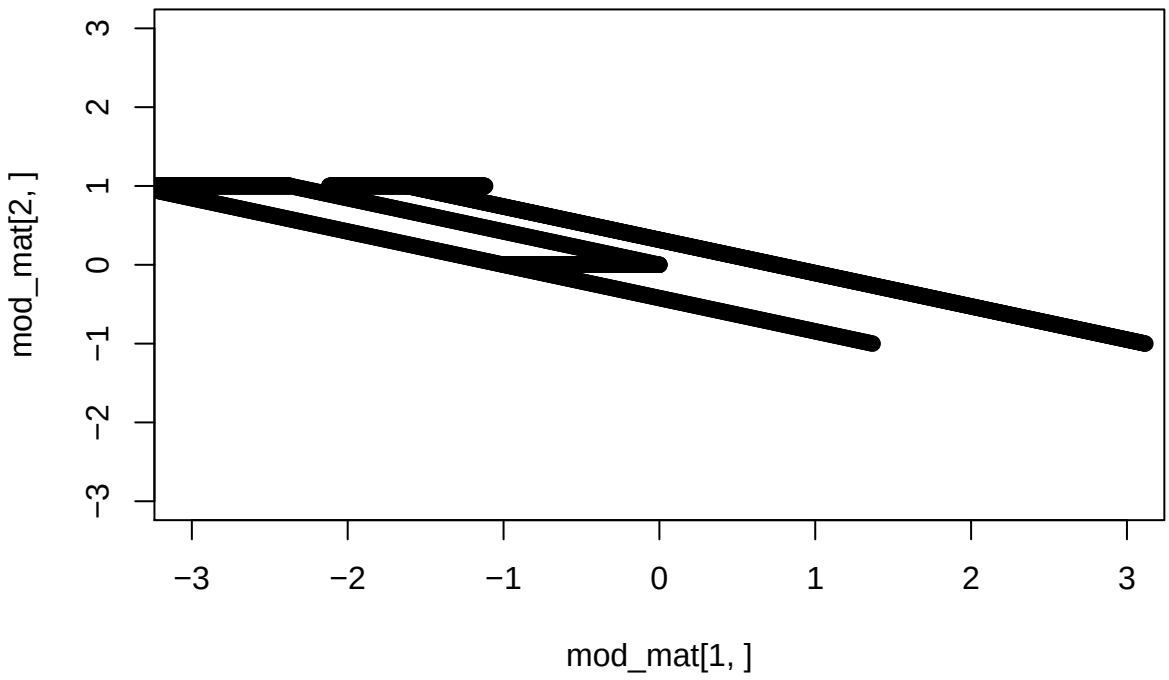

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

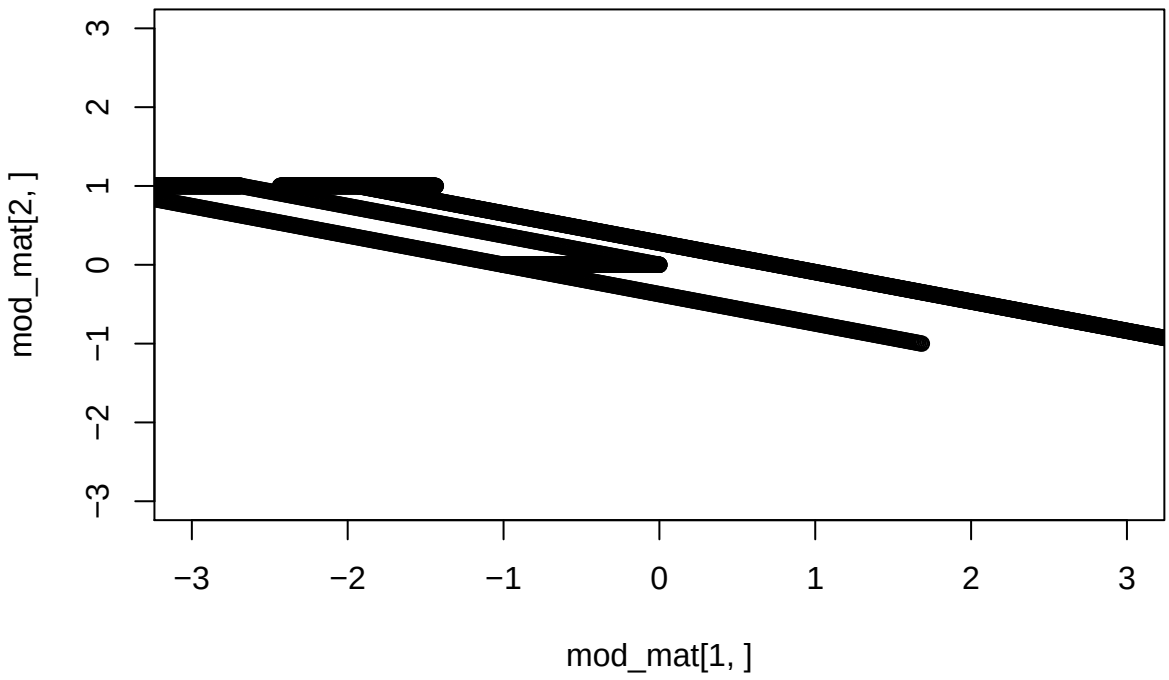

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

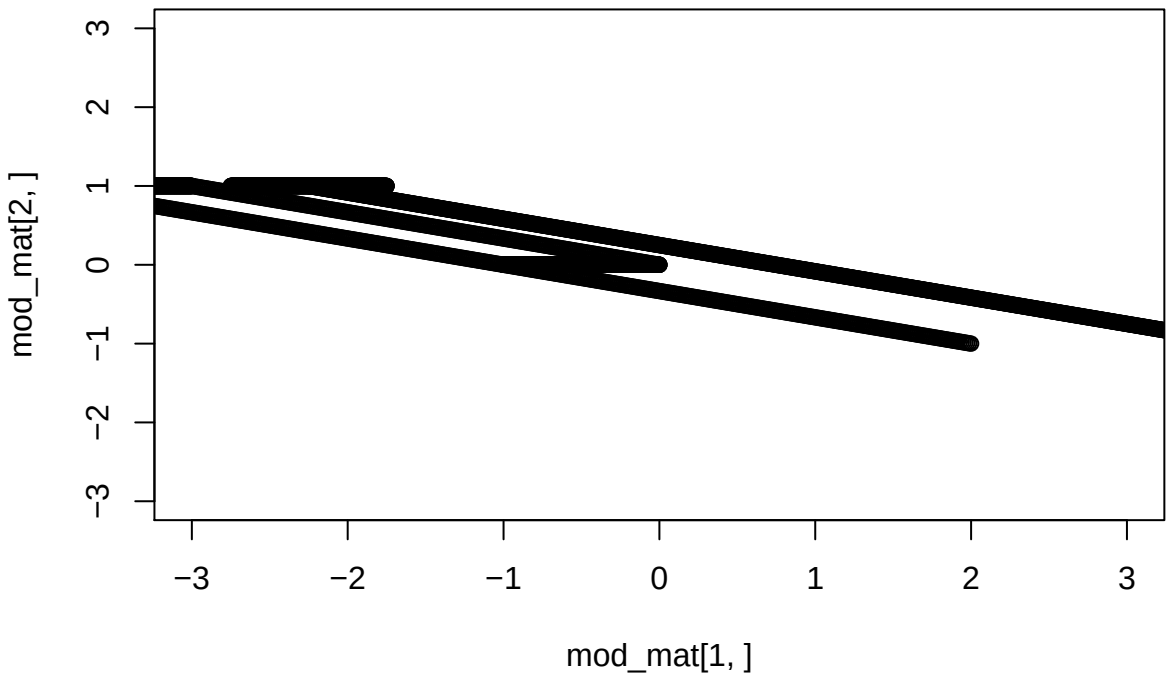

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]


mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

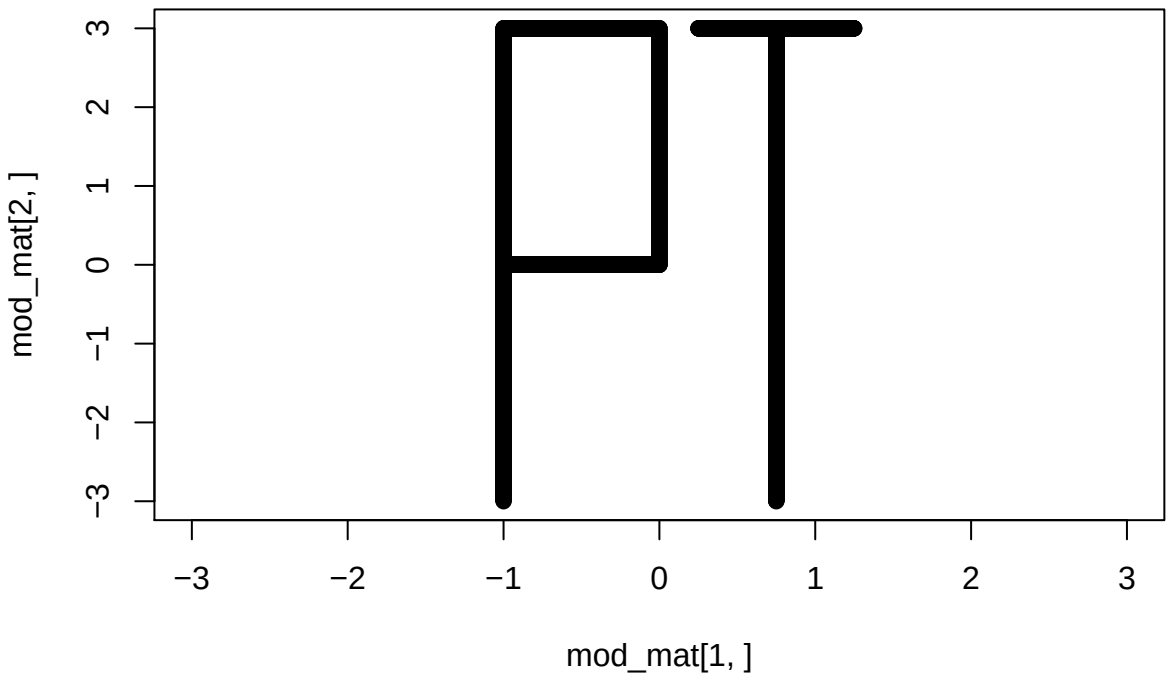

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

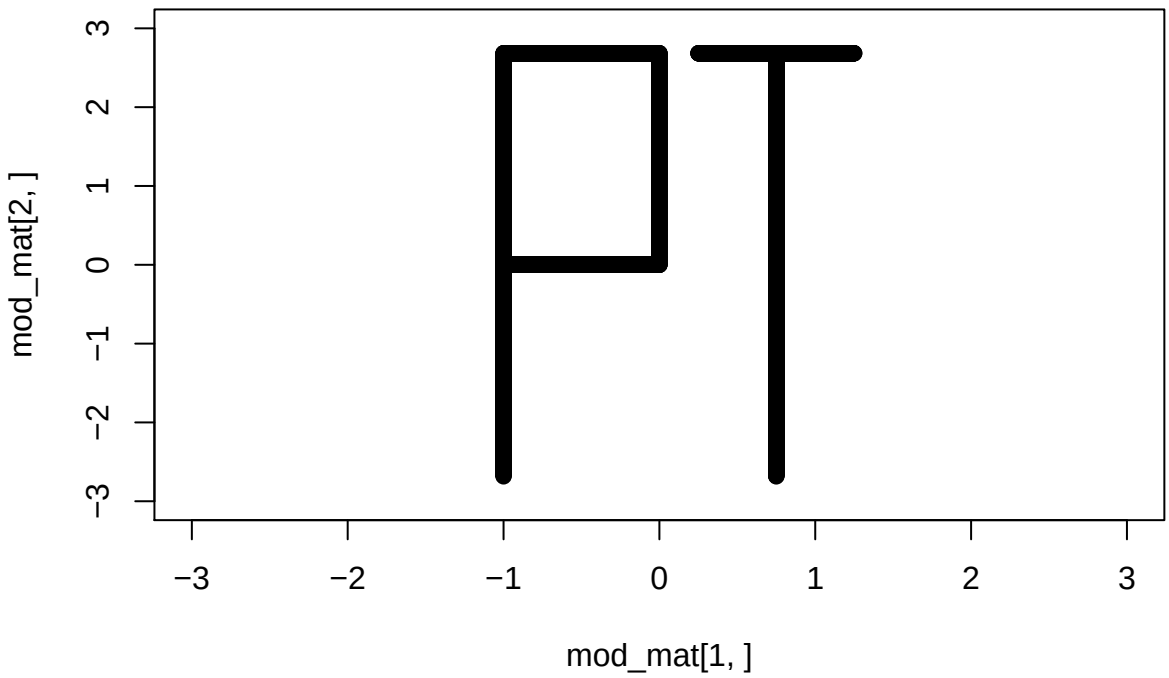

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

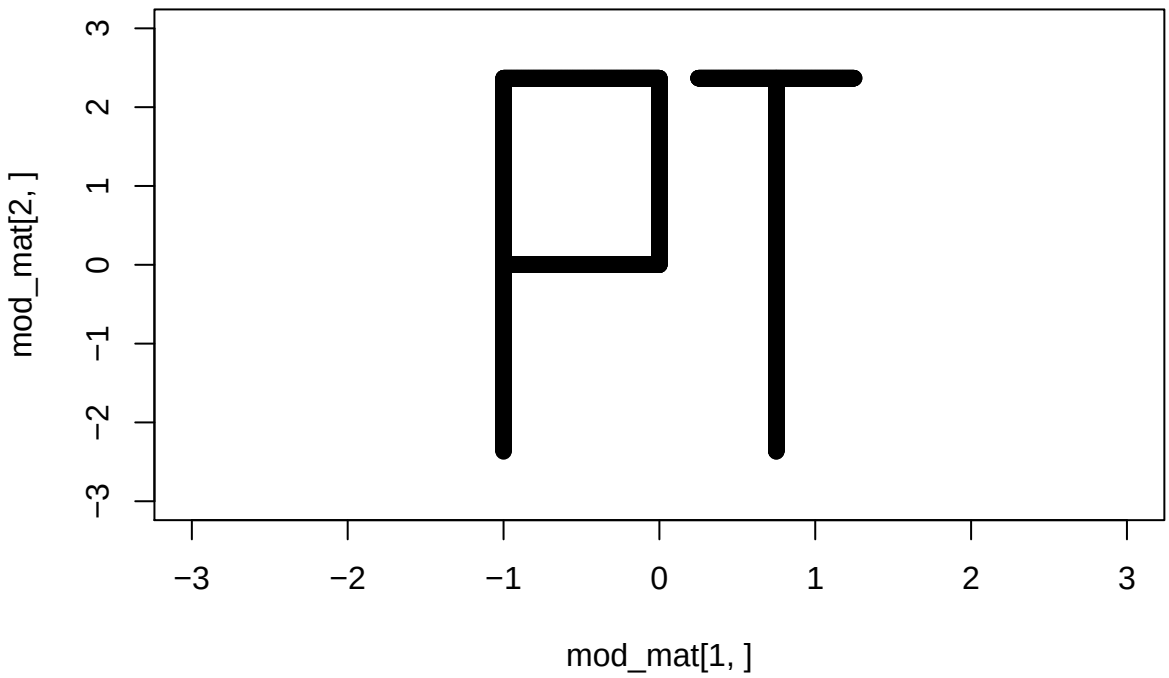

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

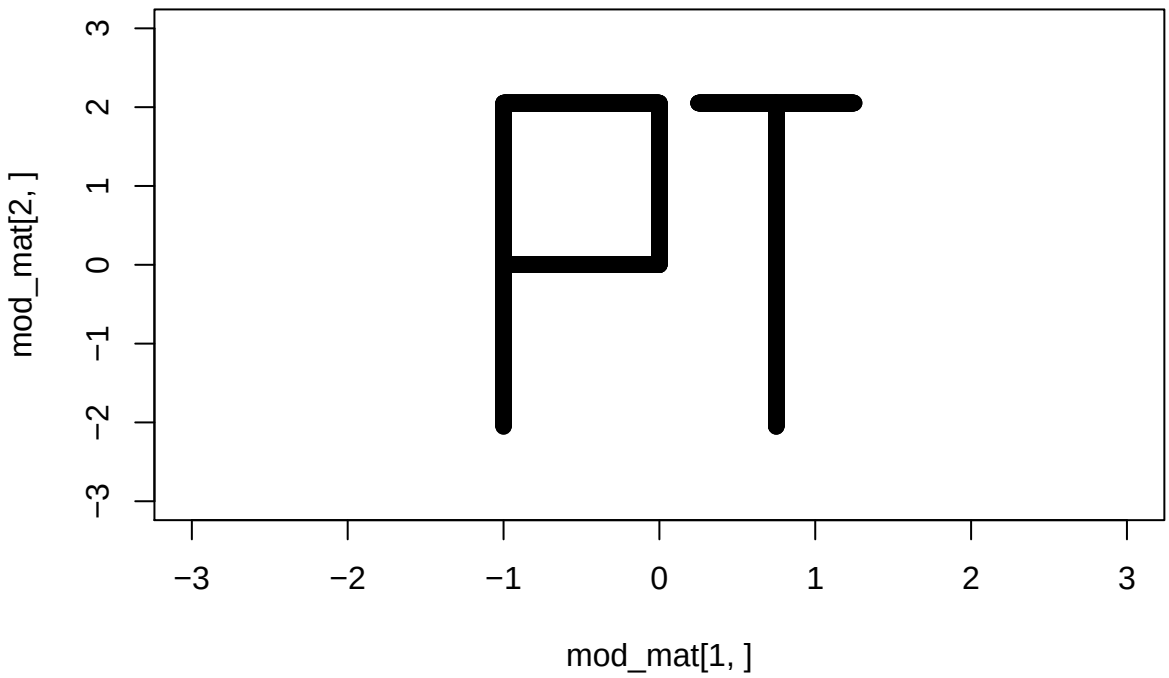


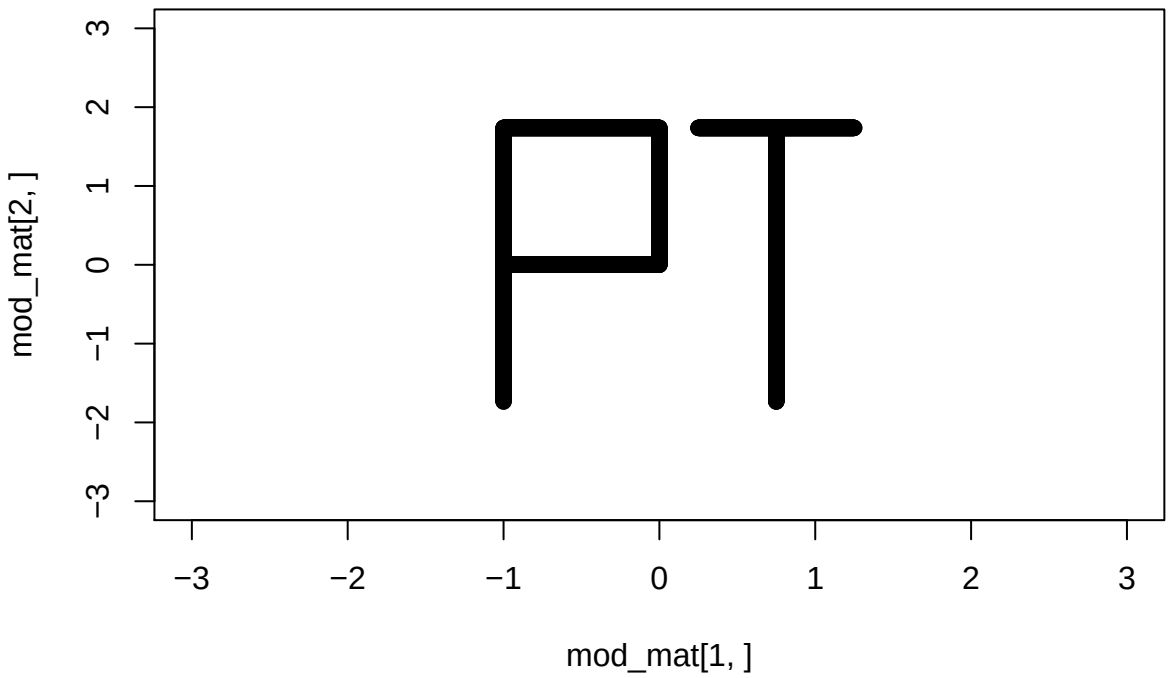

```
# Reset 2x2 identity matrix
a=diag(2)
#----------------------
# Projection on X axis (while X value scales)
#----------------------
for (i in seq(3,-3, length.out=20)) {
#  a[1,1] = i
  a[2,2] = i
  mod_mat = apply(z, 2, function(x) a%*%x)
  plot(mod_mat[2,]~mod_mat[1,], xlim=c(-3,3), ylim=c(-3,3))
  ani.record()
}
```

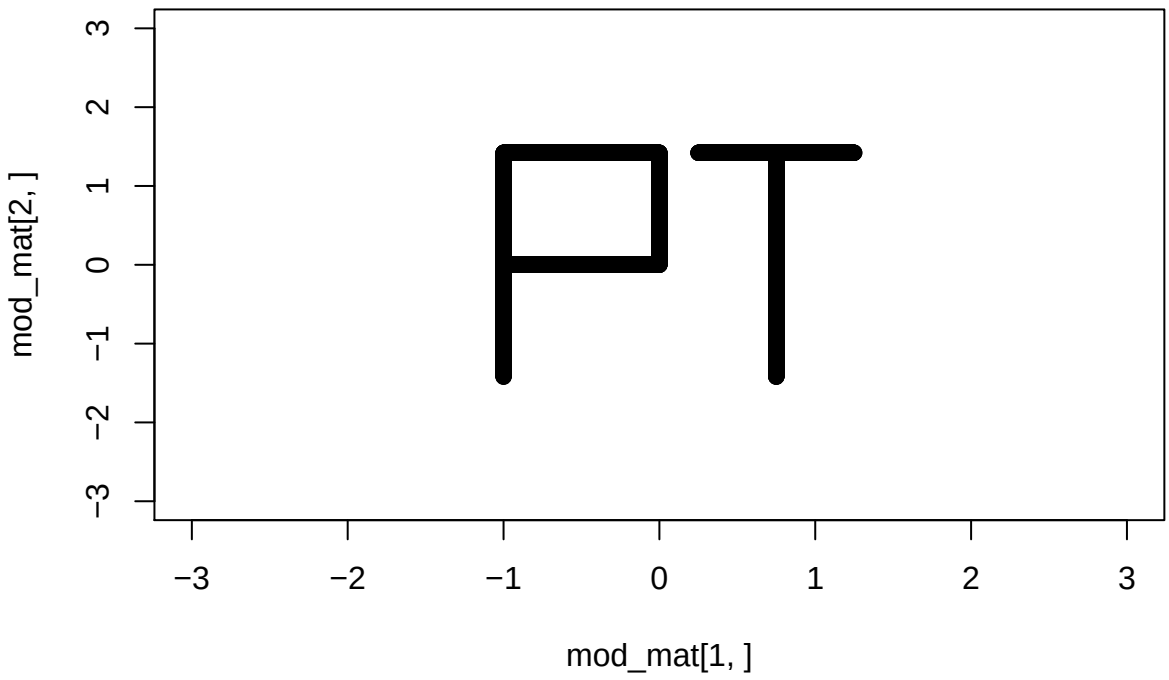

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

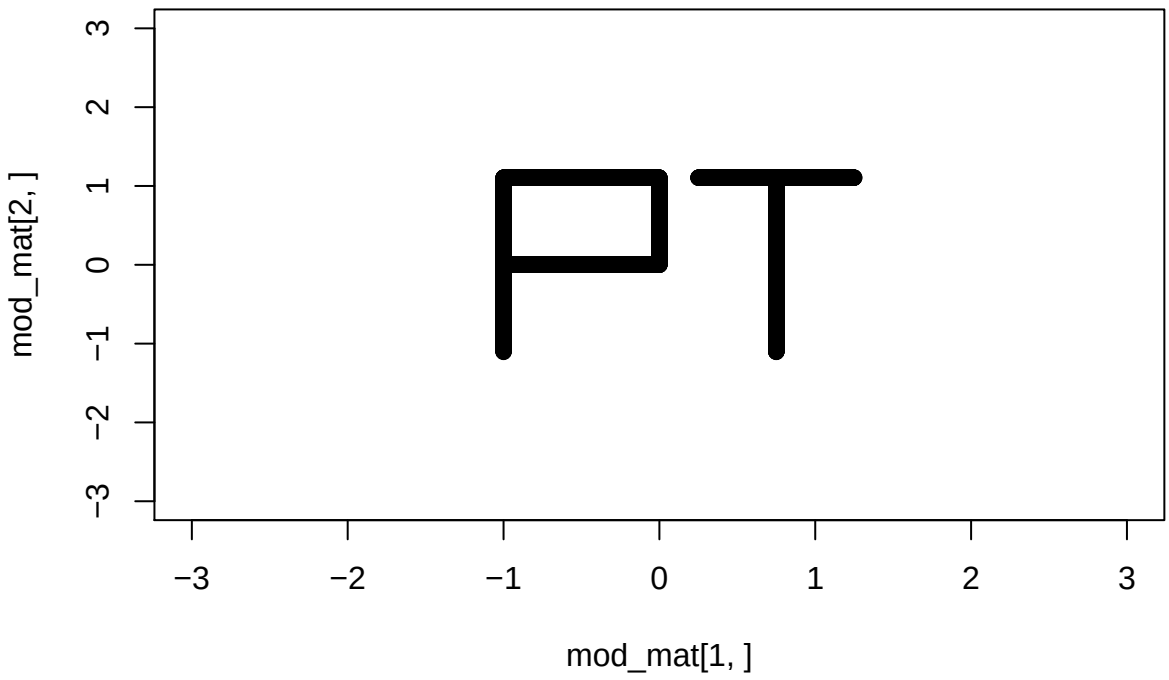

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

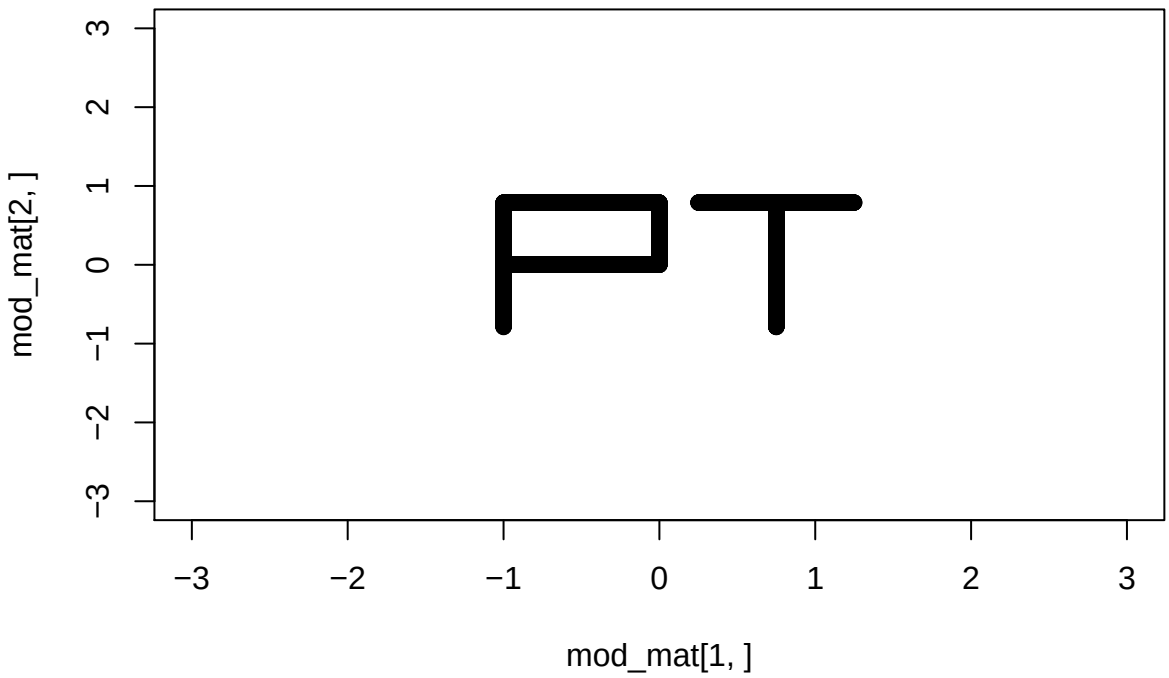

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

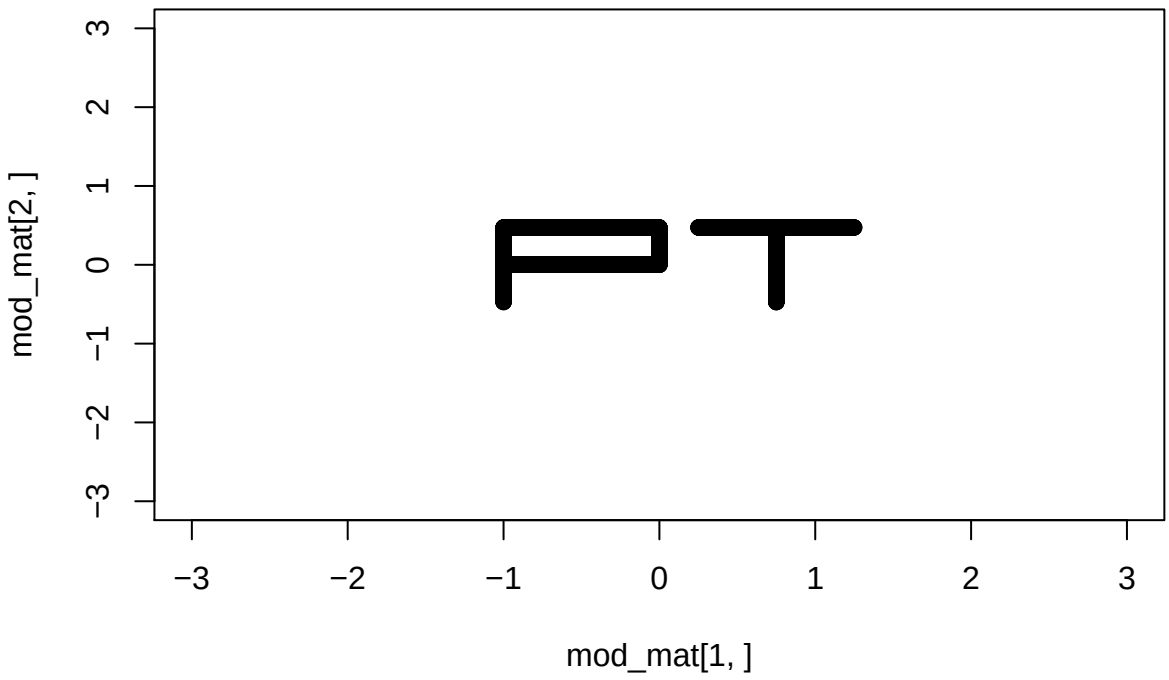

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

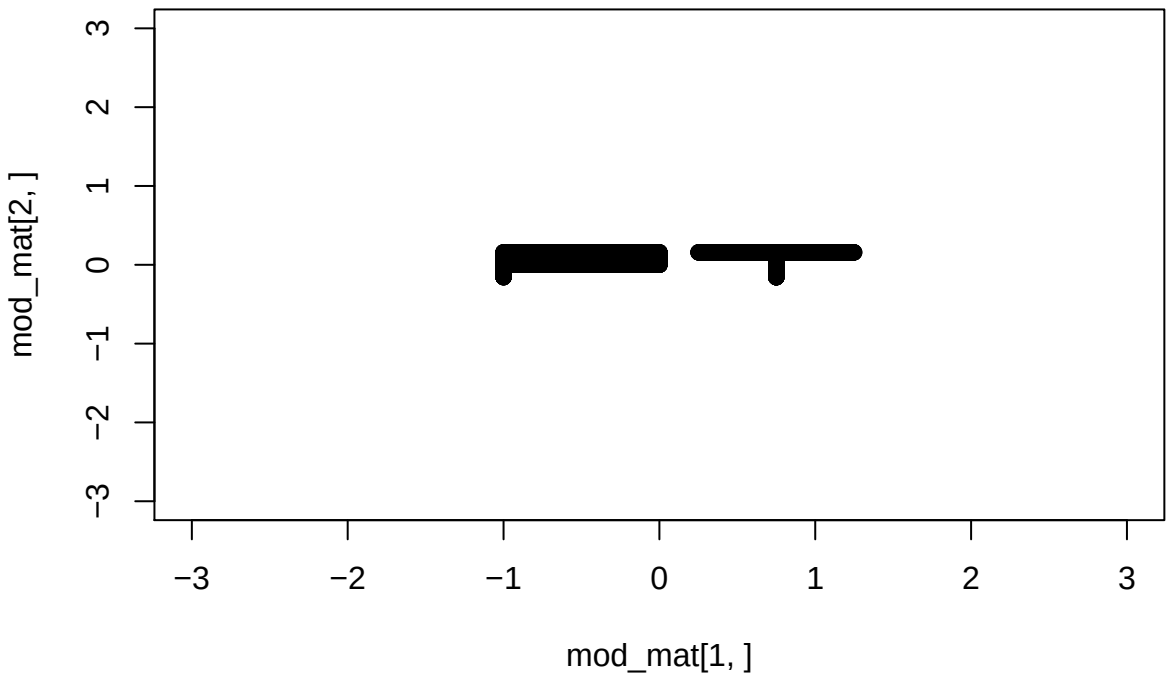

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

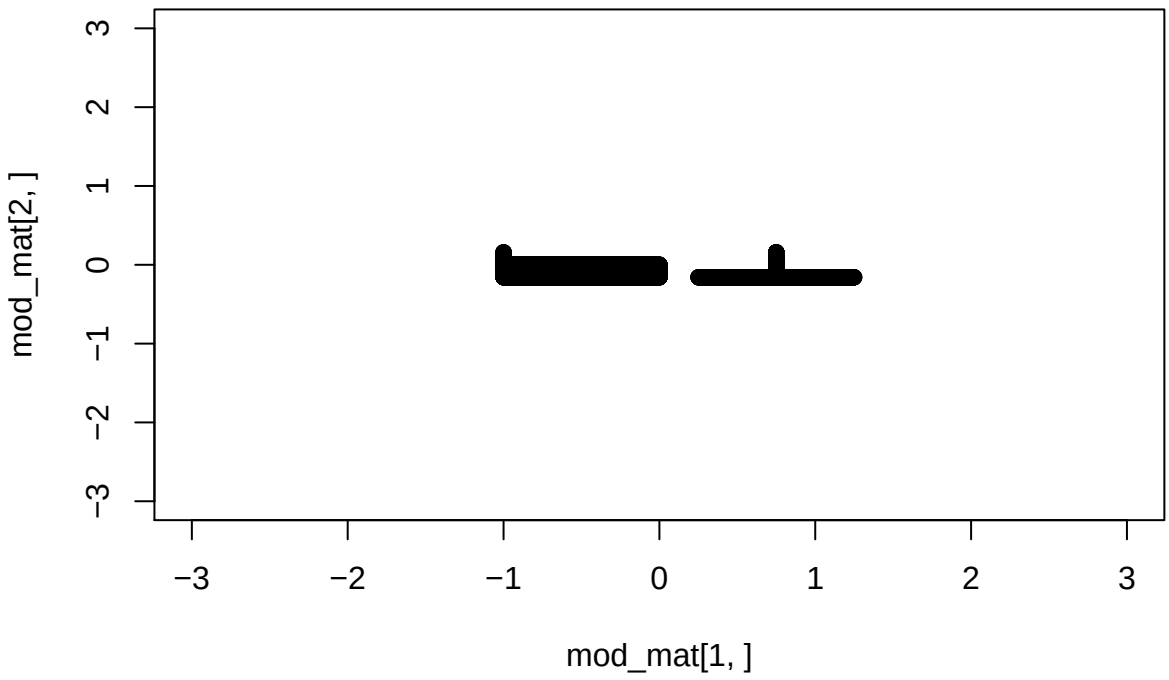

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

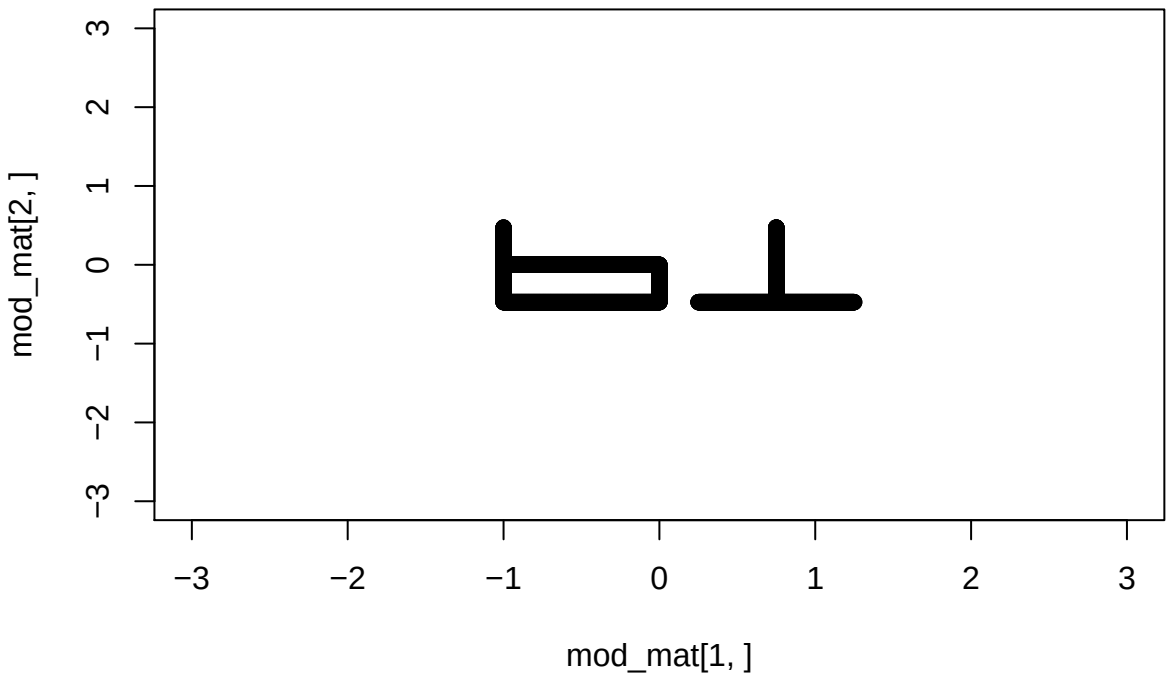

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

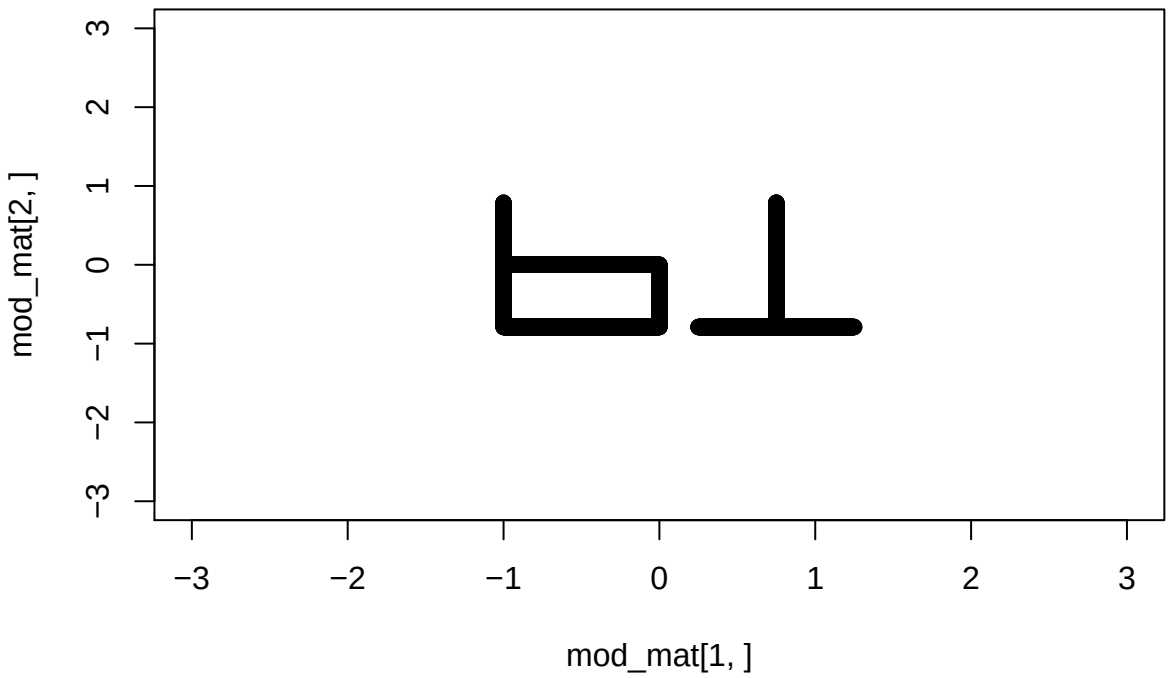

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

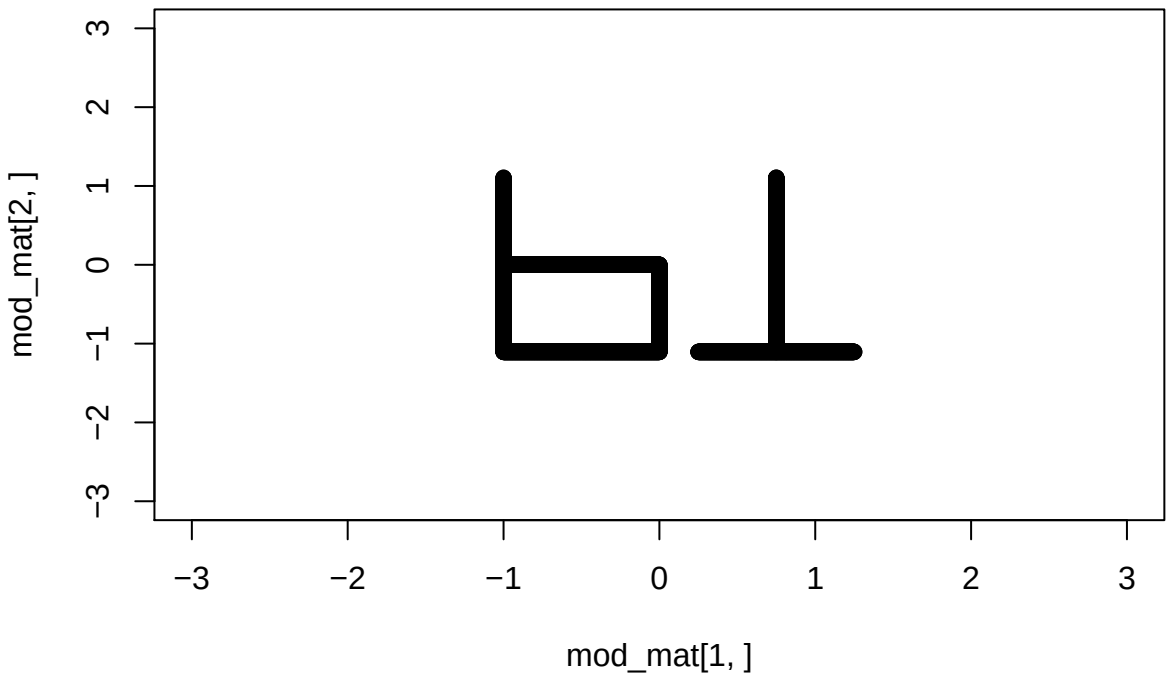

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

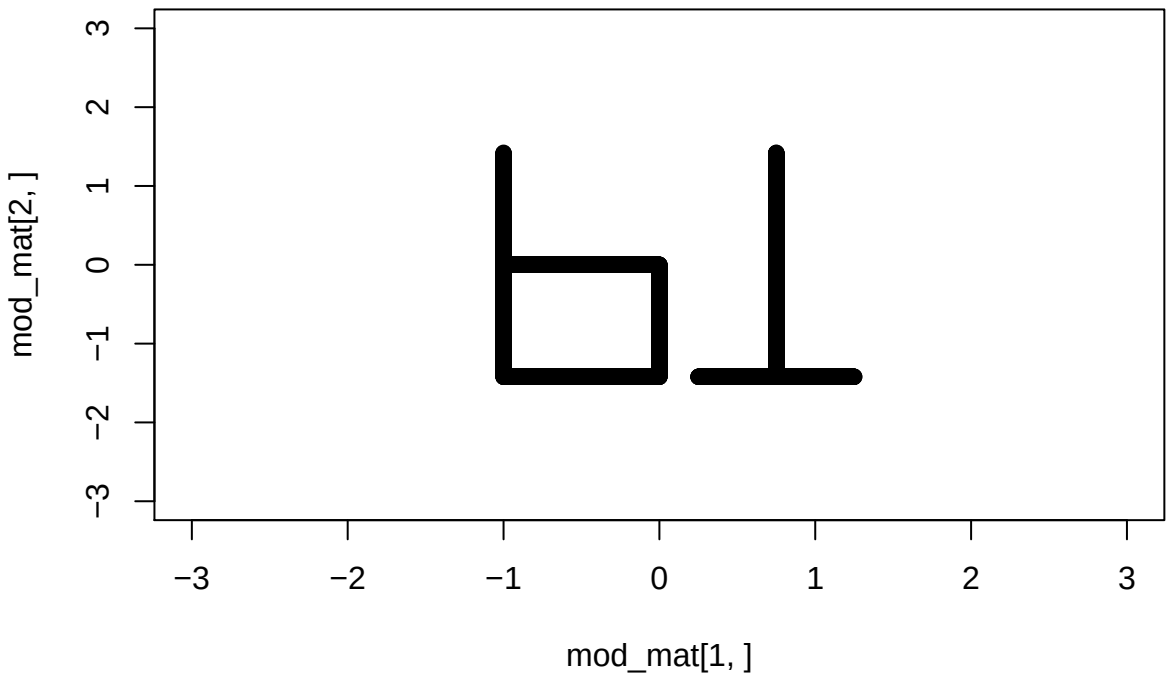

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

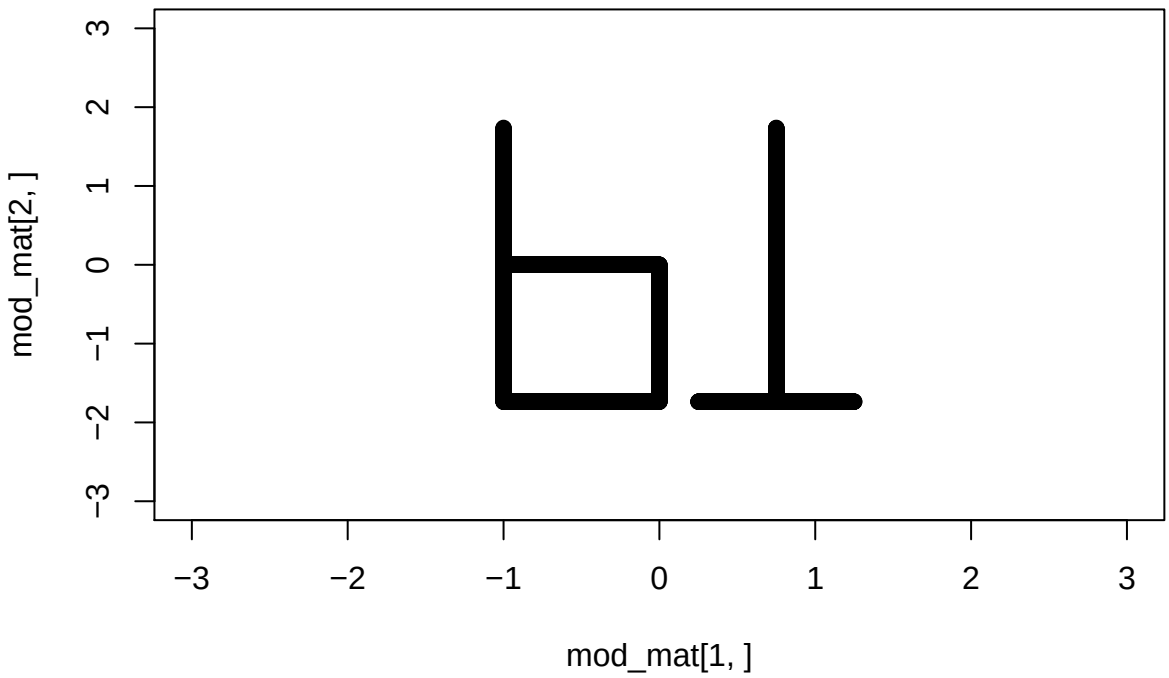

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

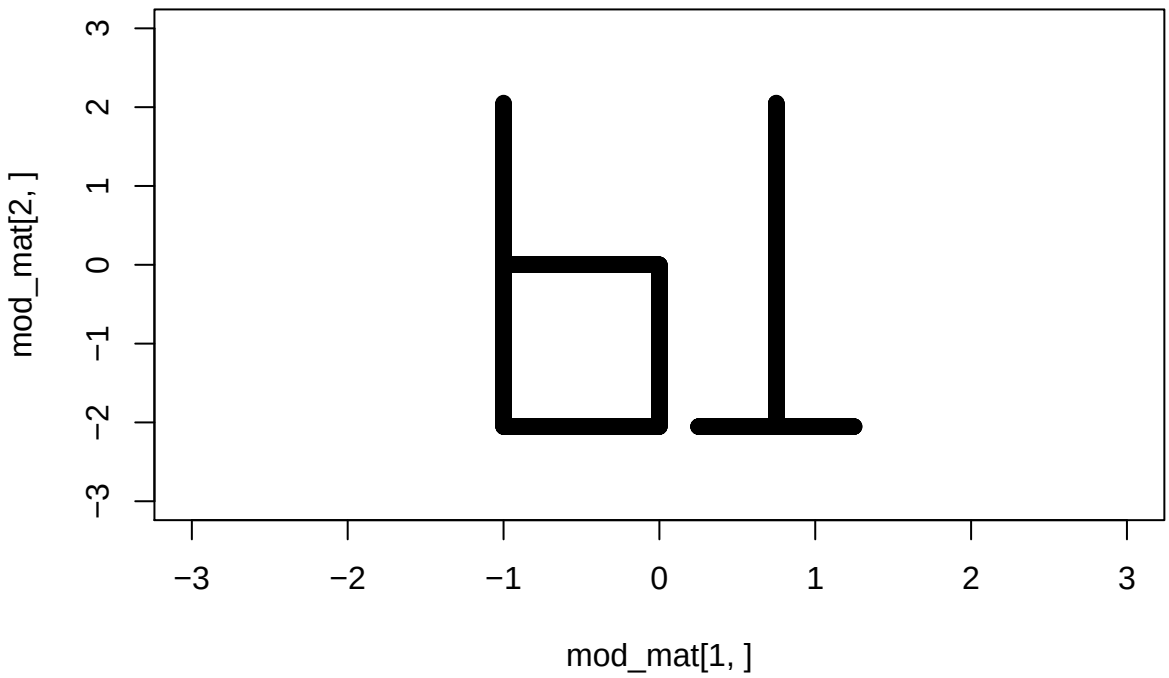

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

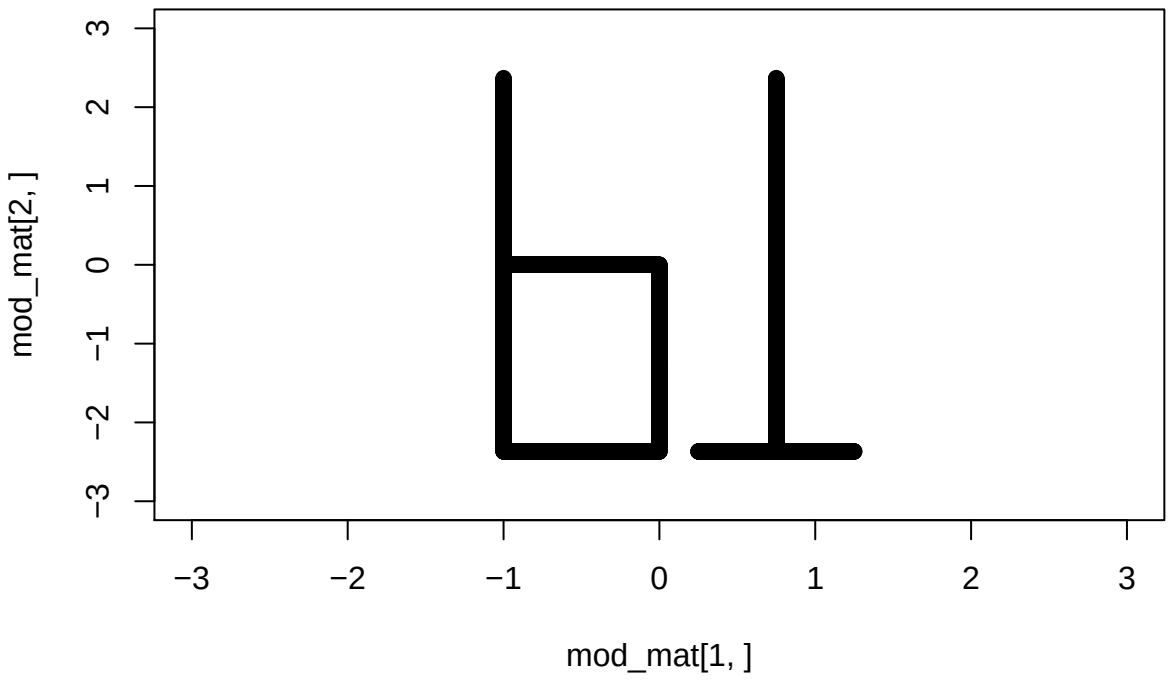

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

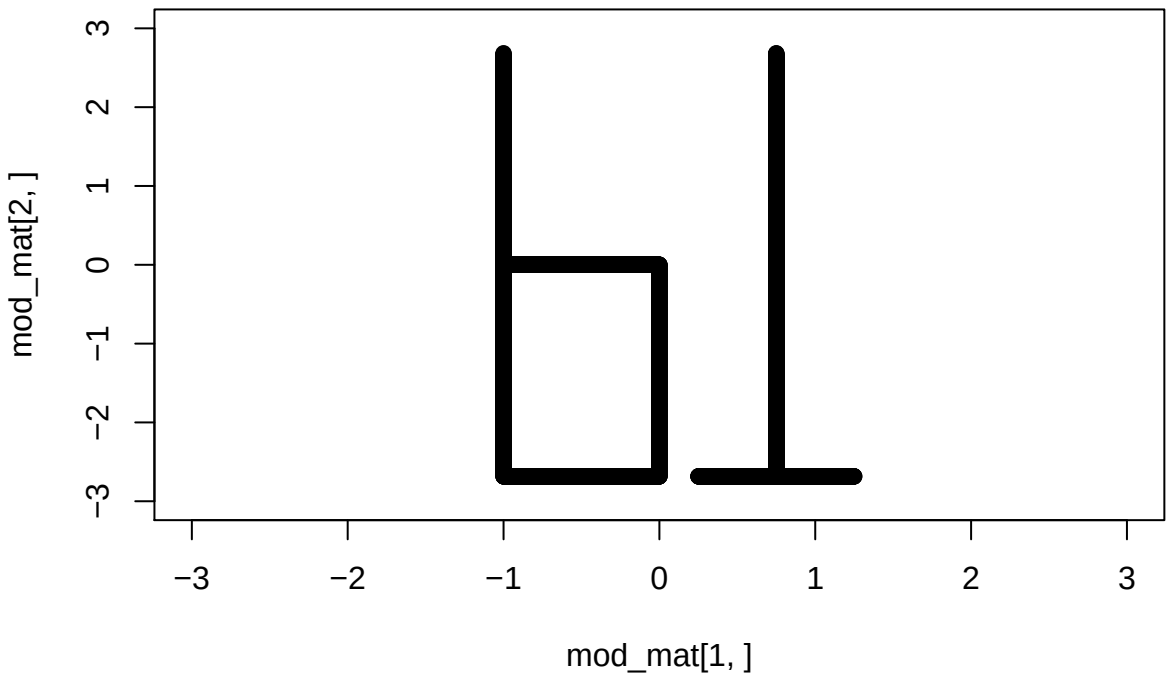

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

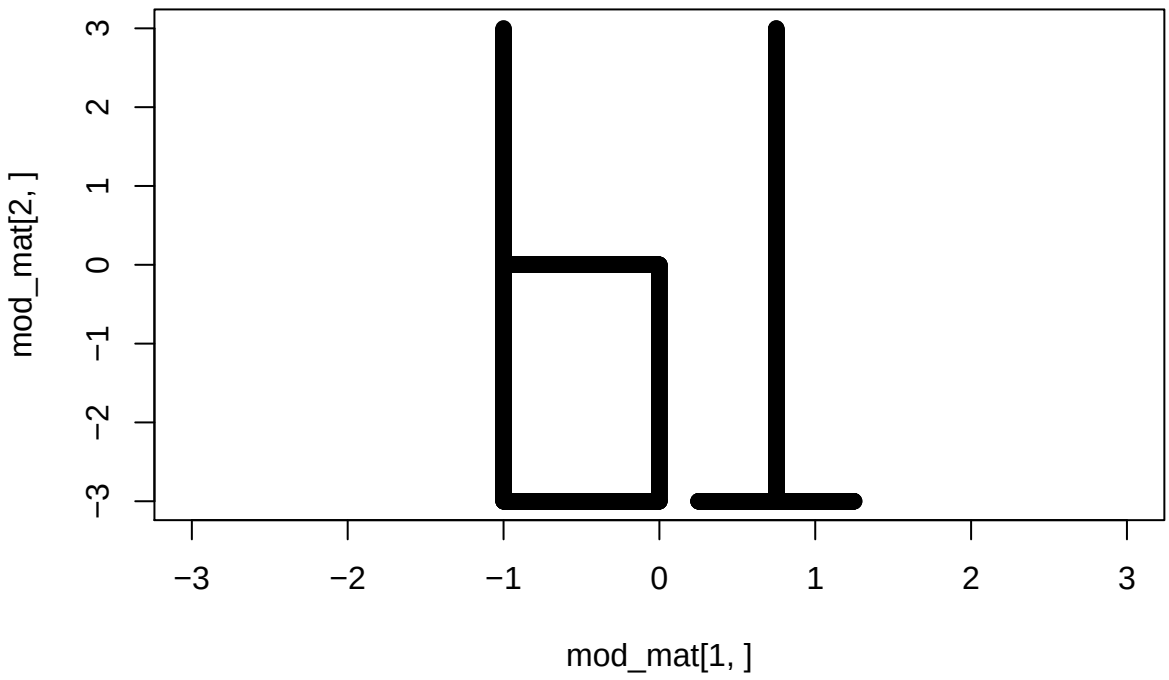

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]


mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

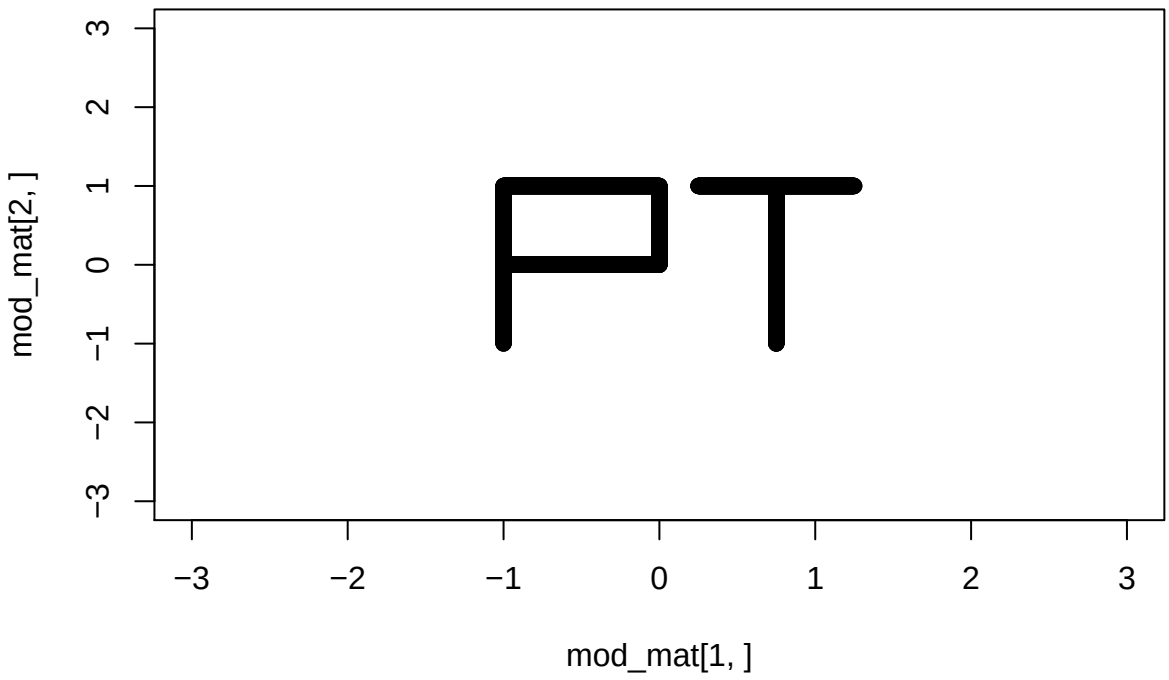

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

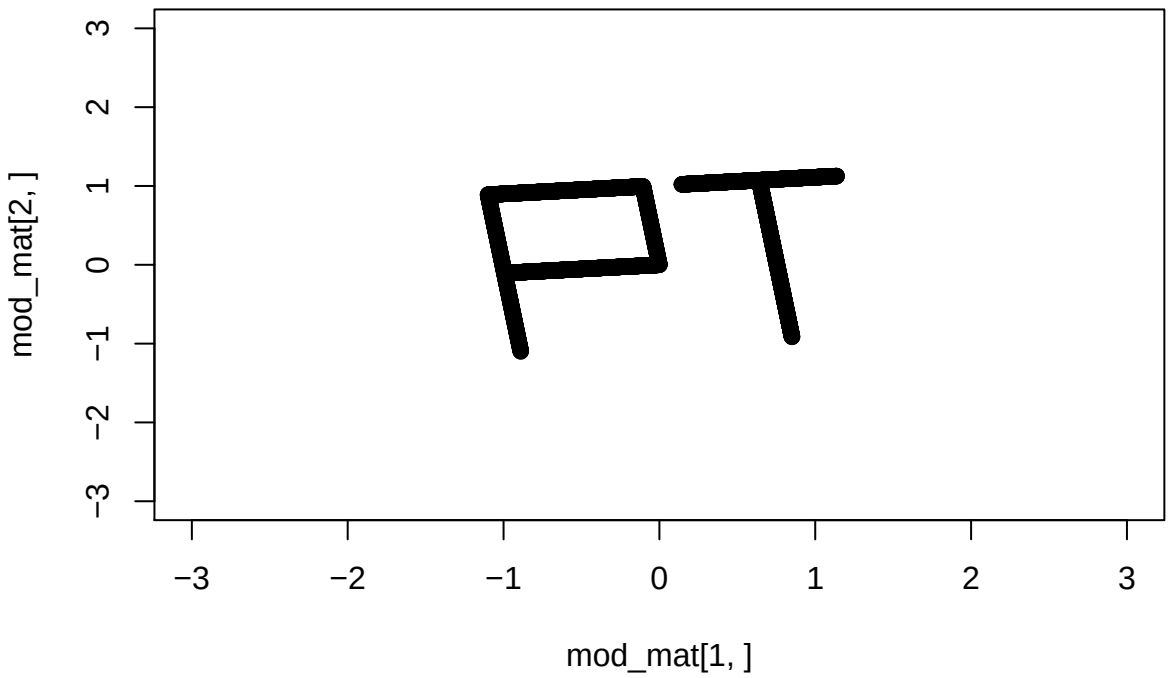

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

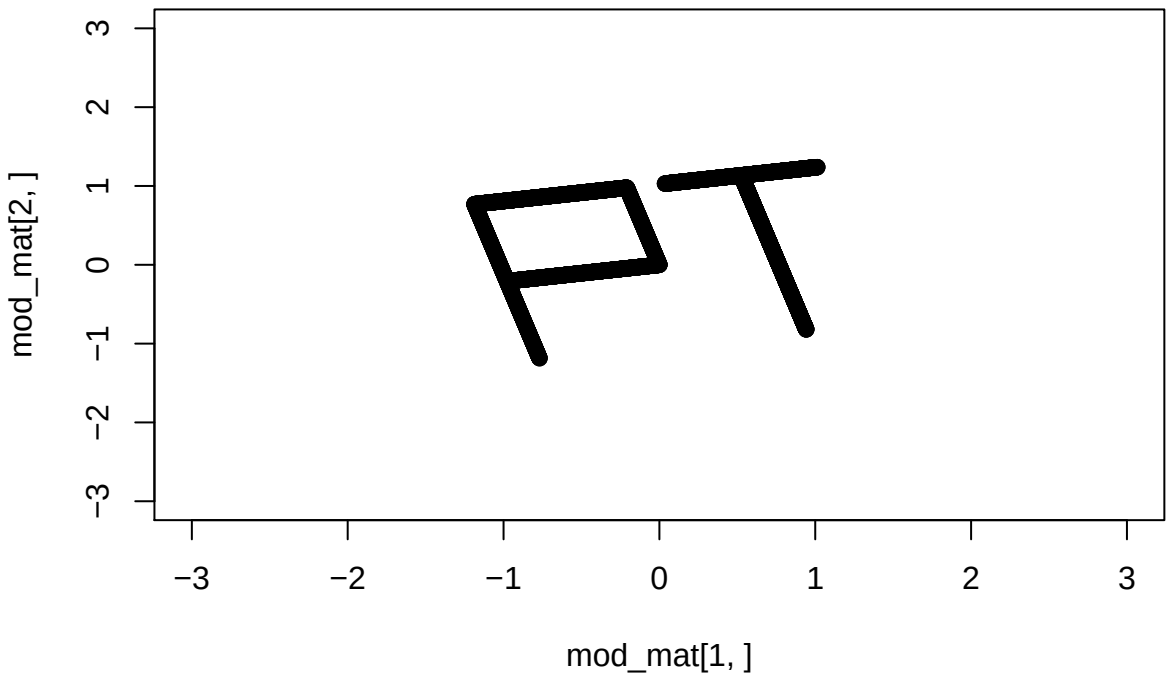

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

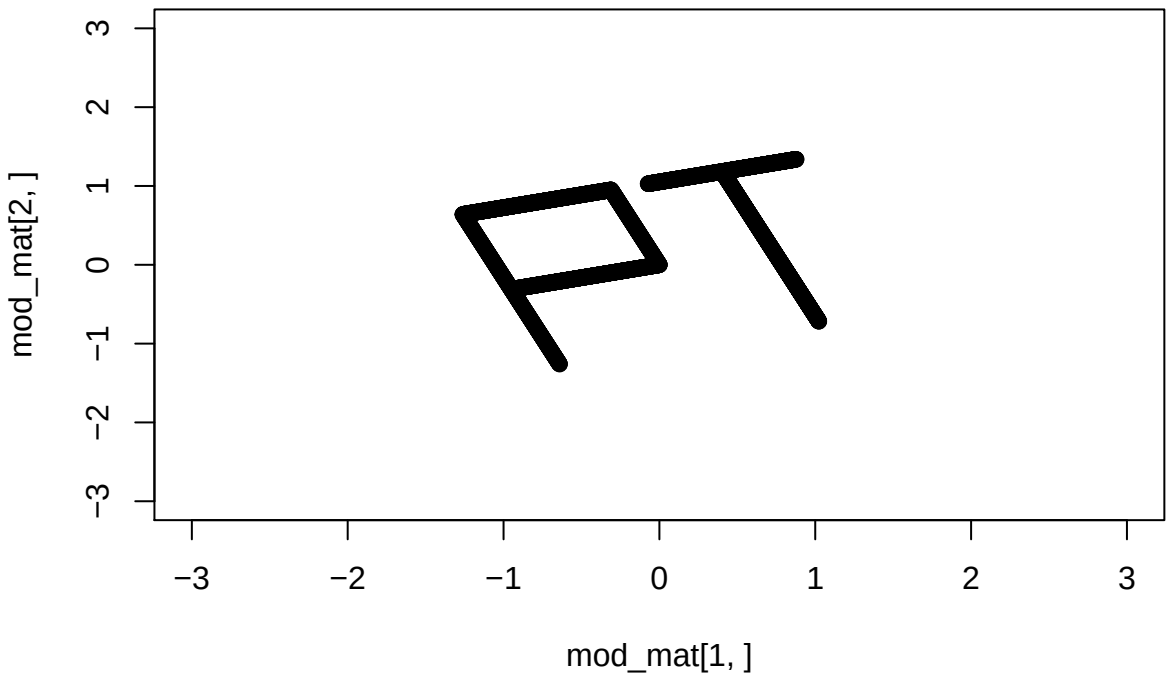


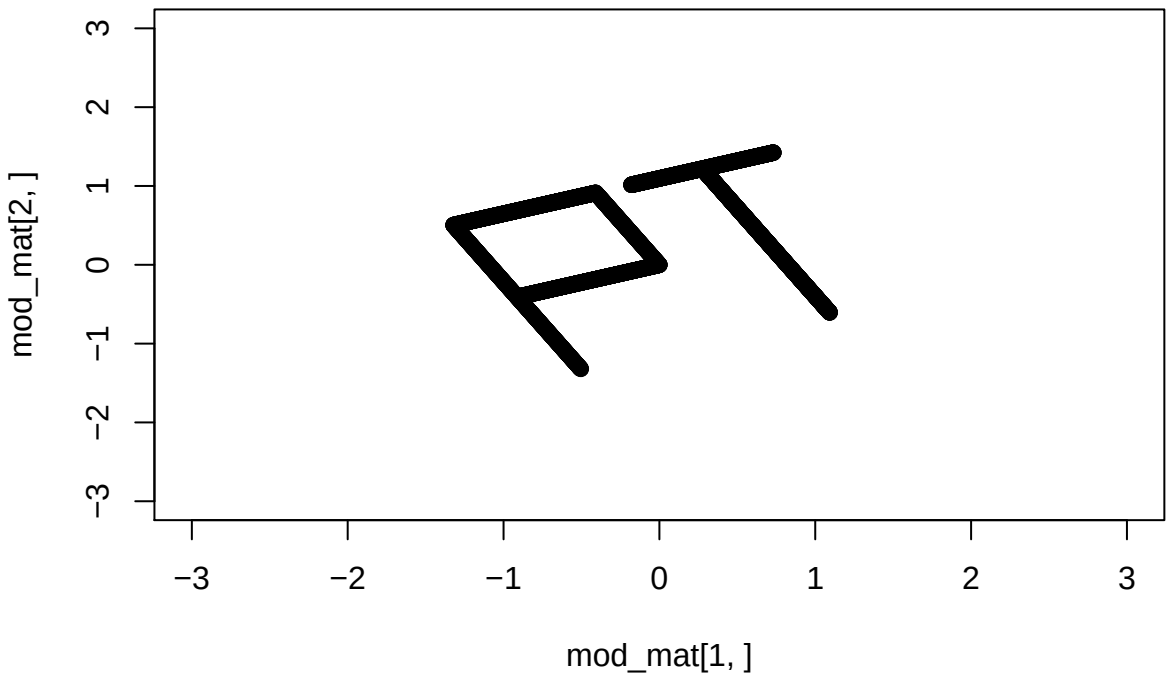

```
# Define rotation function for matrix multiplication
rotate_func=function(x){matrix(c(cos(x), -sin(x), sin(x), cos(x)), byrow=T, nrow=2)}

# Reset 2x2 identity matrix
a=diag(2)
#----------------------
# Rotation animation
#----------------------
for (i in seq(0,2, length.out=20)) {
  a=rotate_func(i)
  mod_mat = apply(z, 2, function(x) a%*%x)
  plot(mod_mat[2,]~mod_mat[1,], xlim=c(-3,3), ylim=c(-3,3))
  ani.record()
}
```

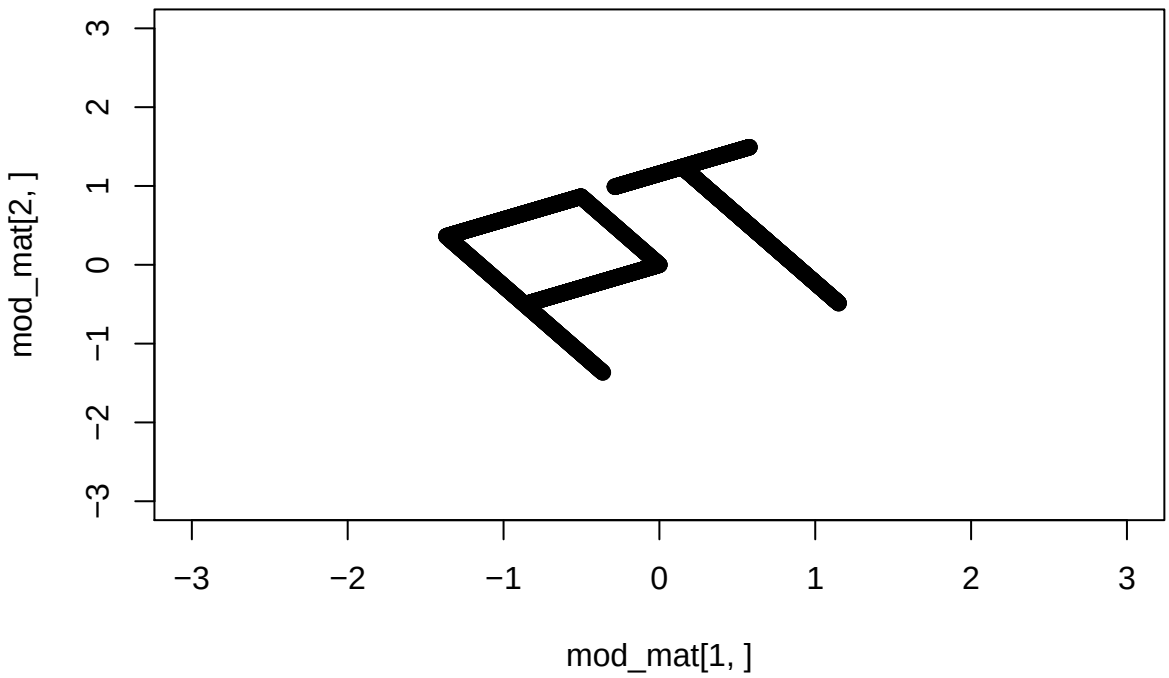

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

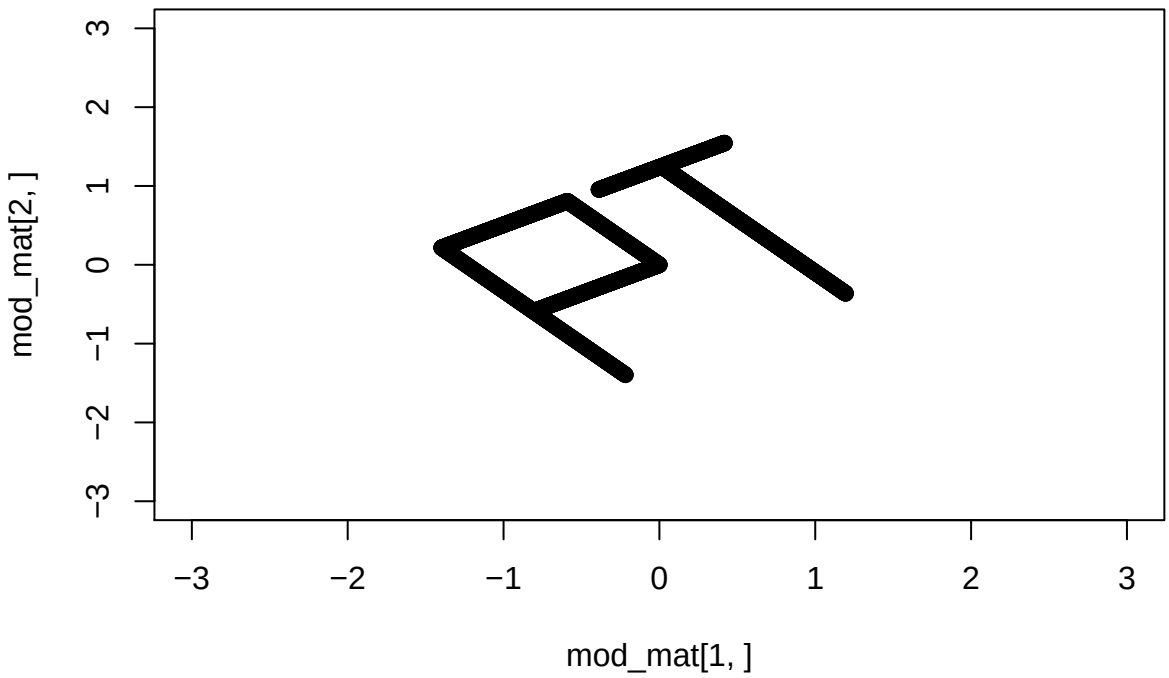

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

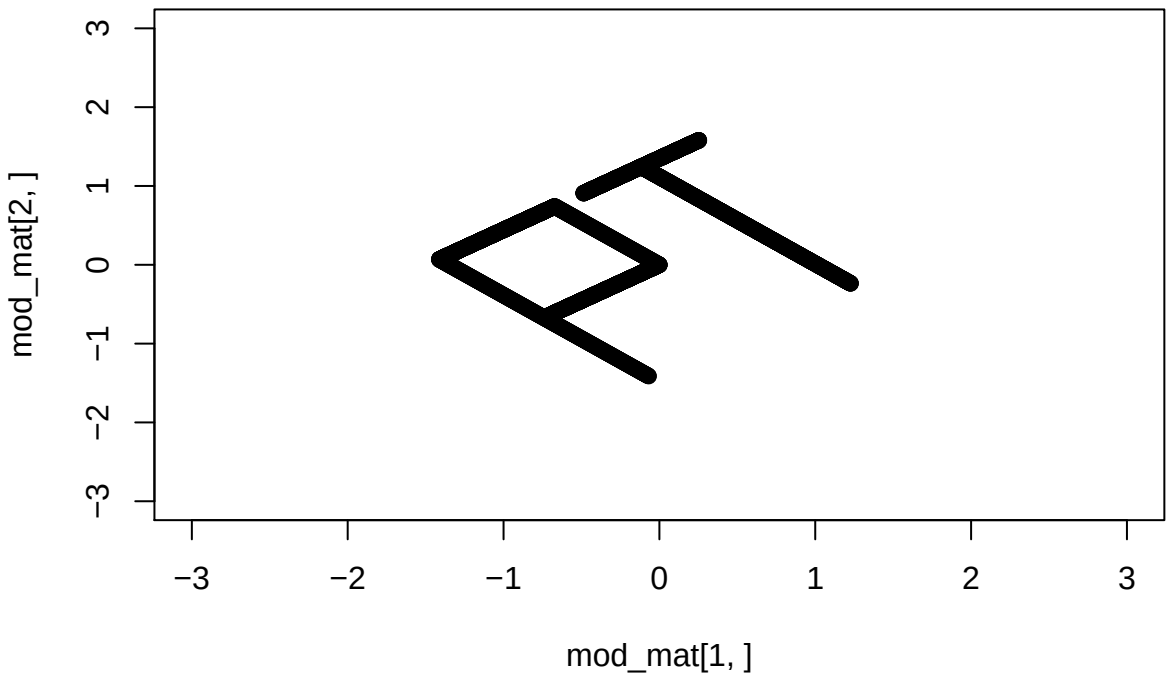

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

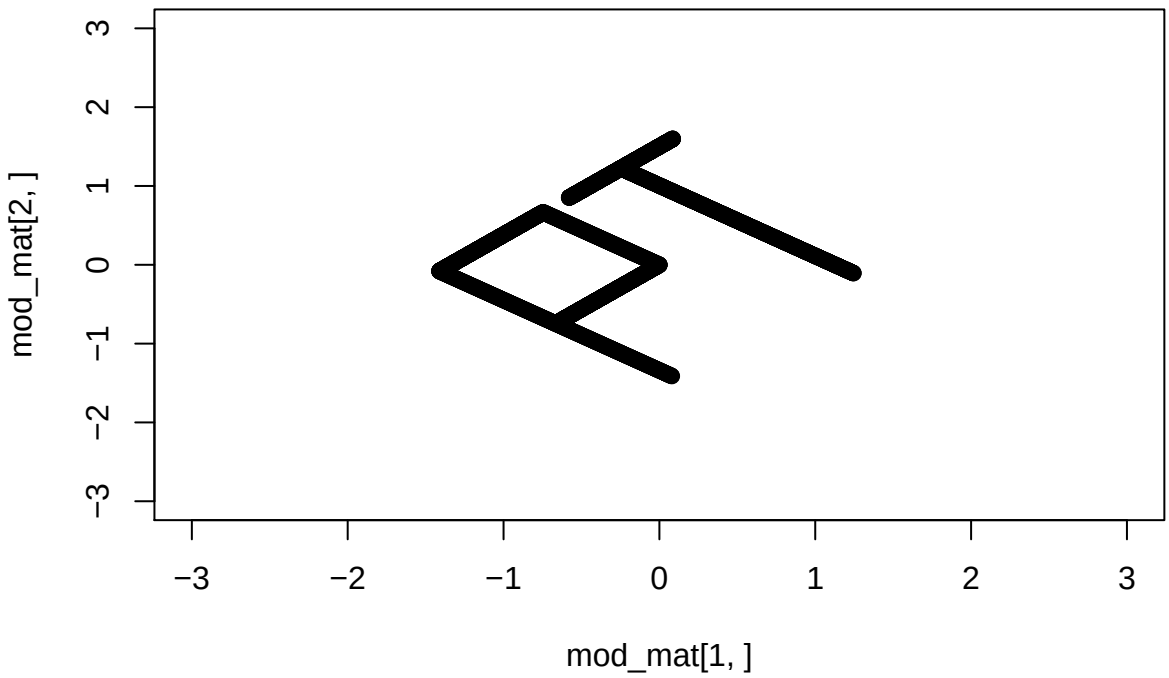

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

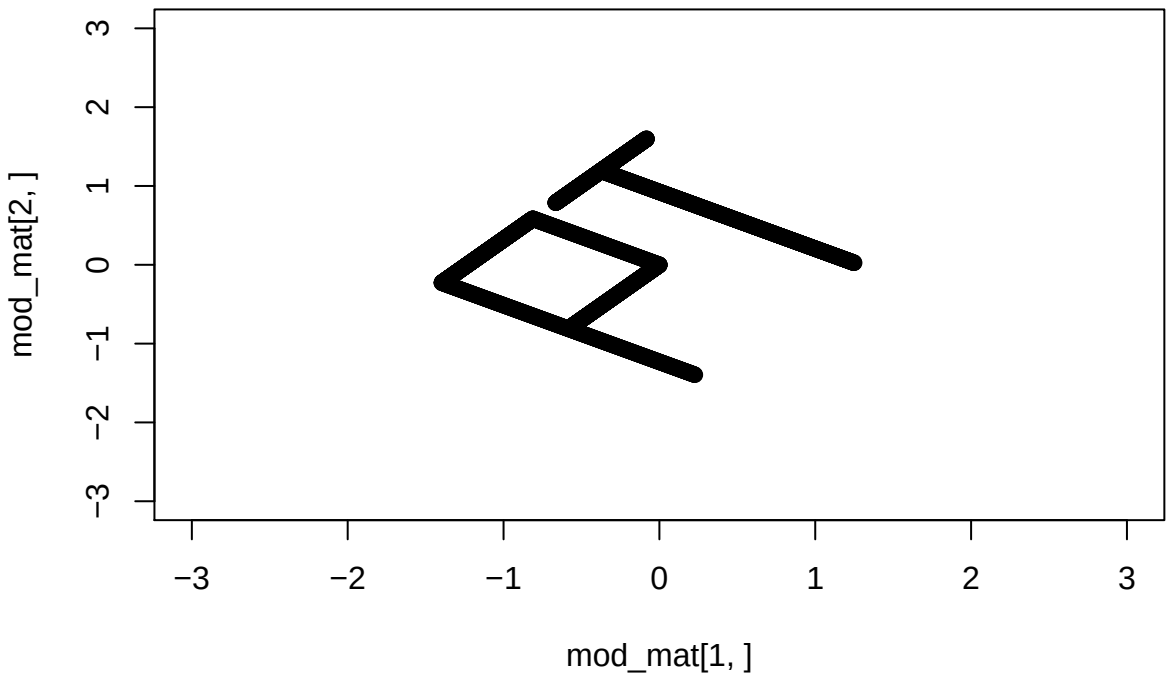

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

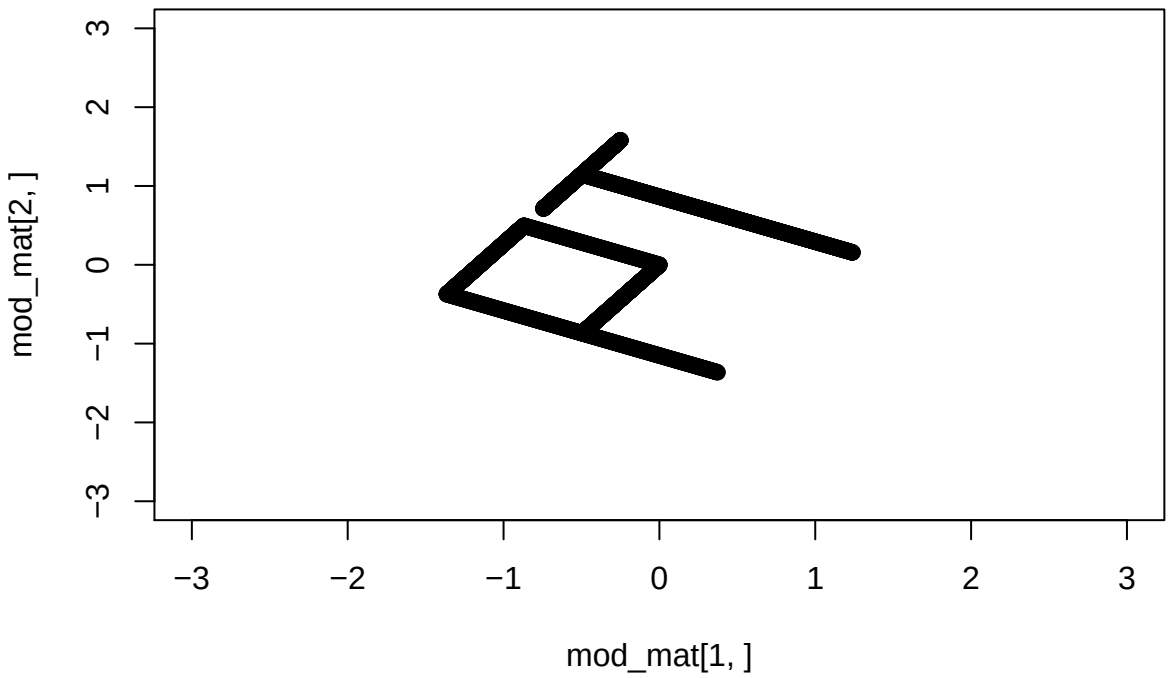

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

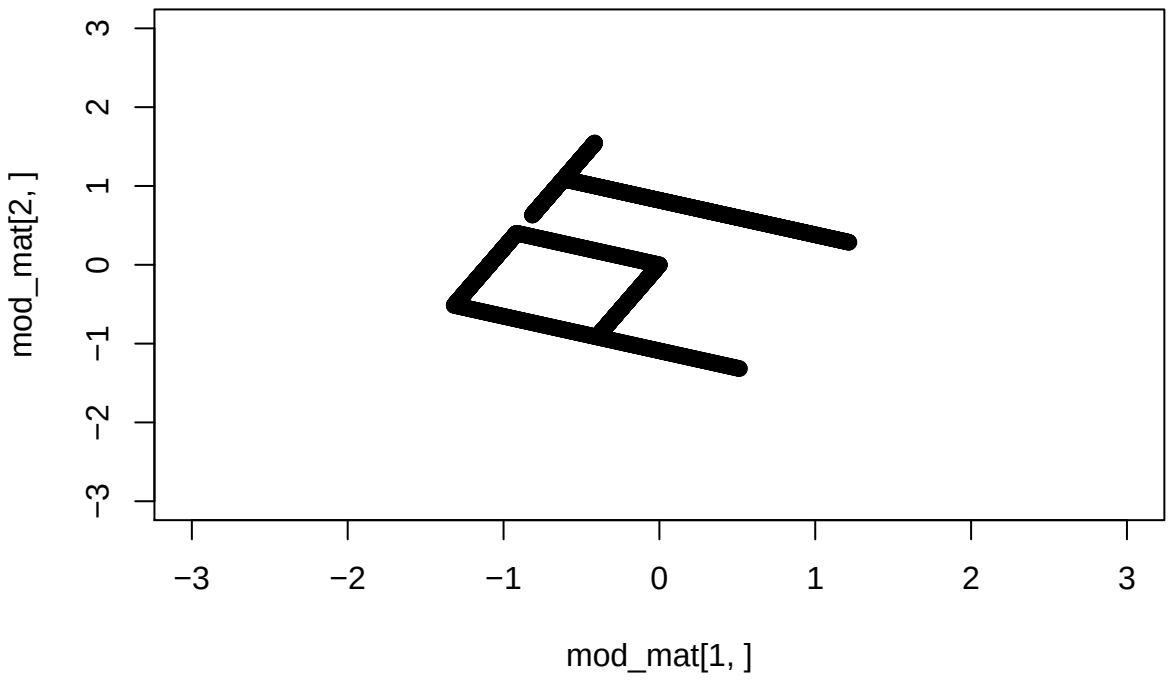

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

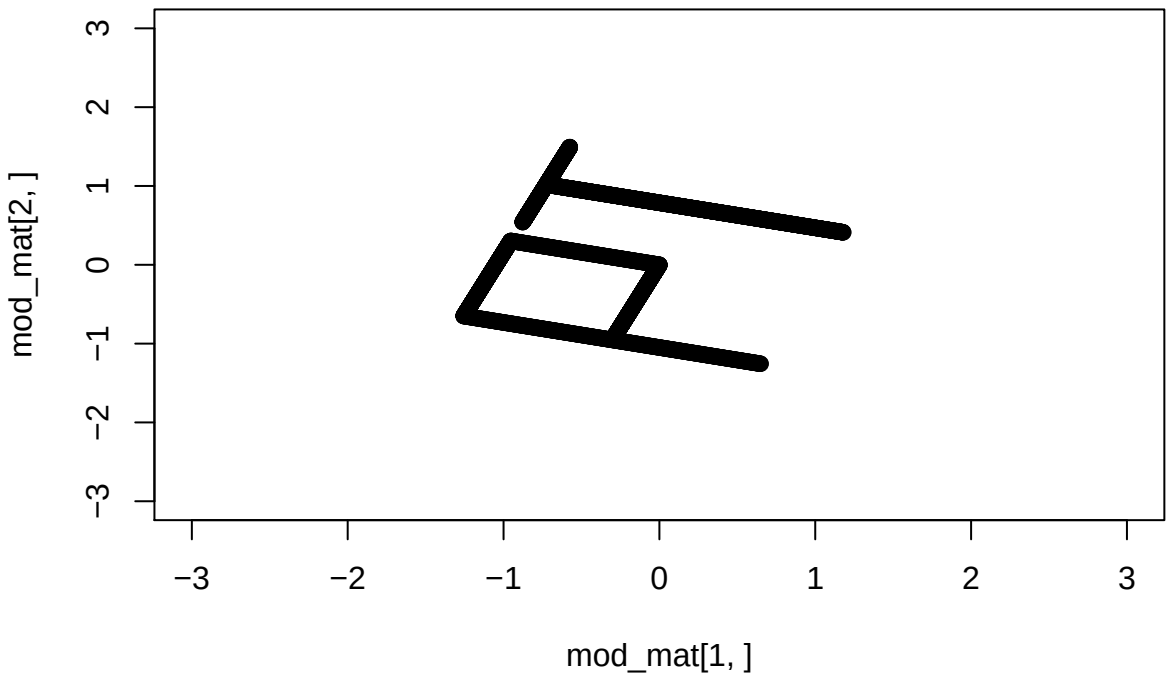

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

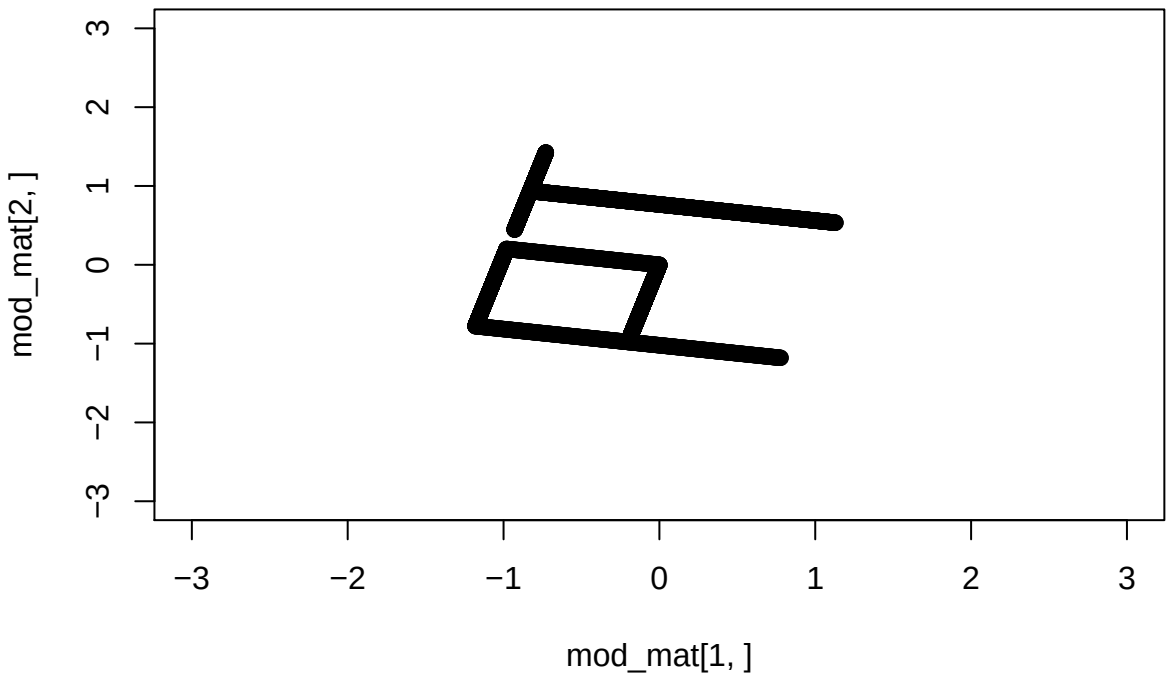

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

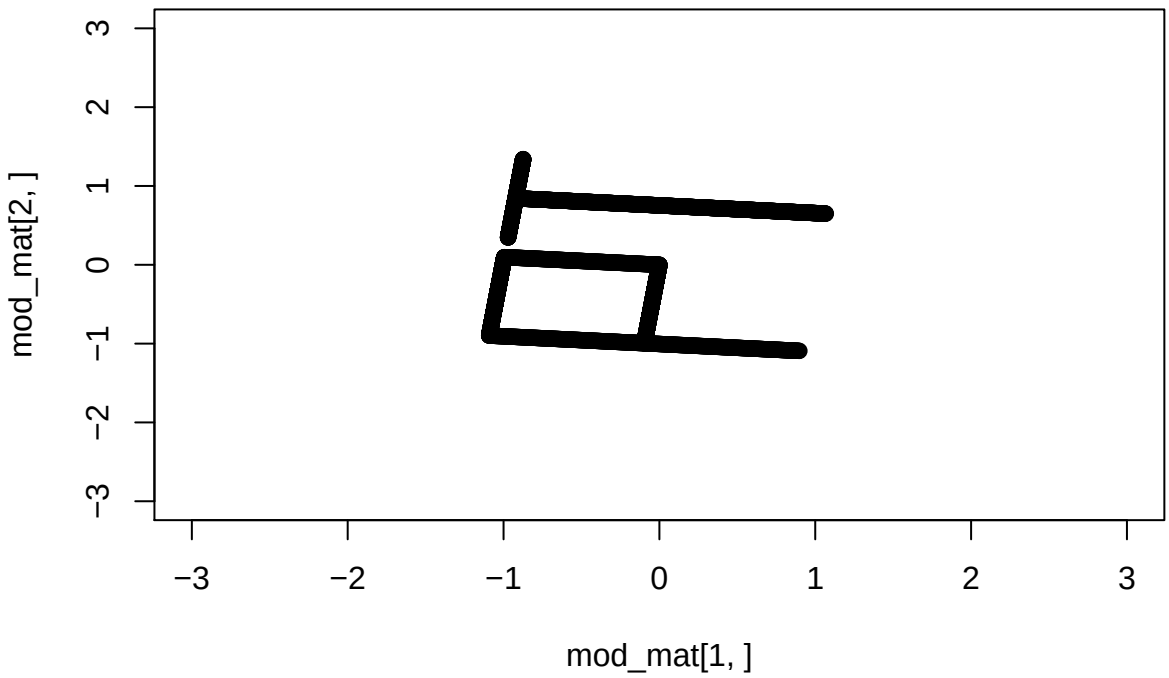

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

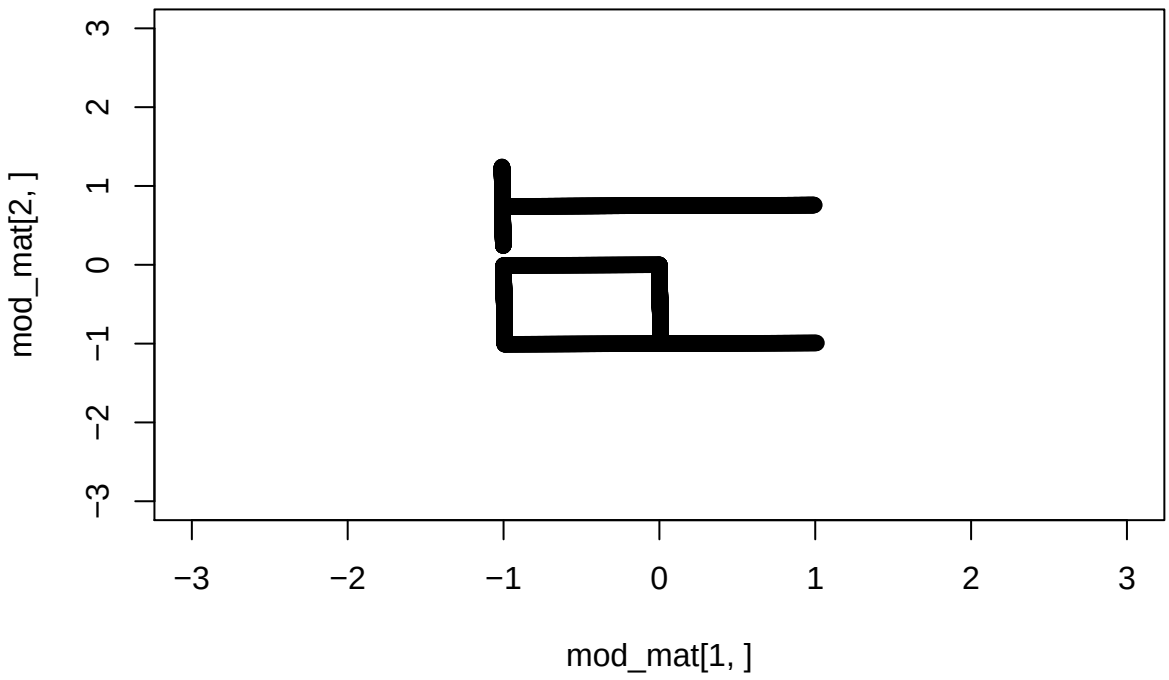

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

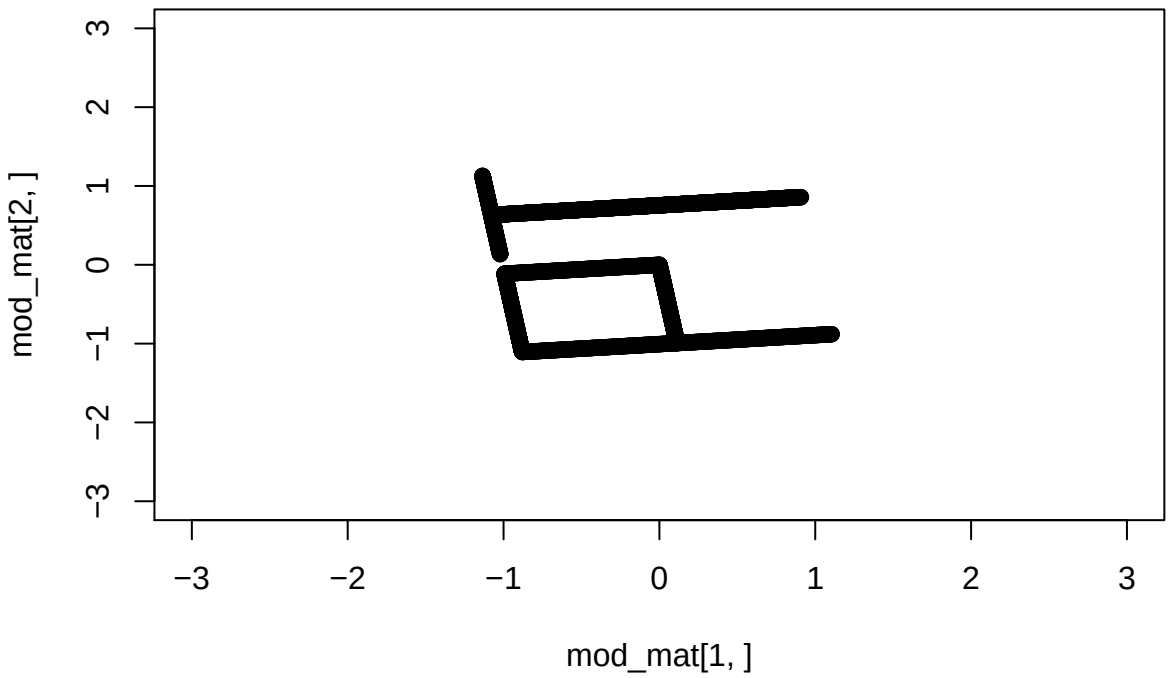

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

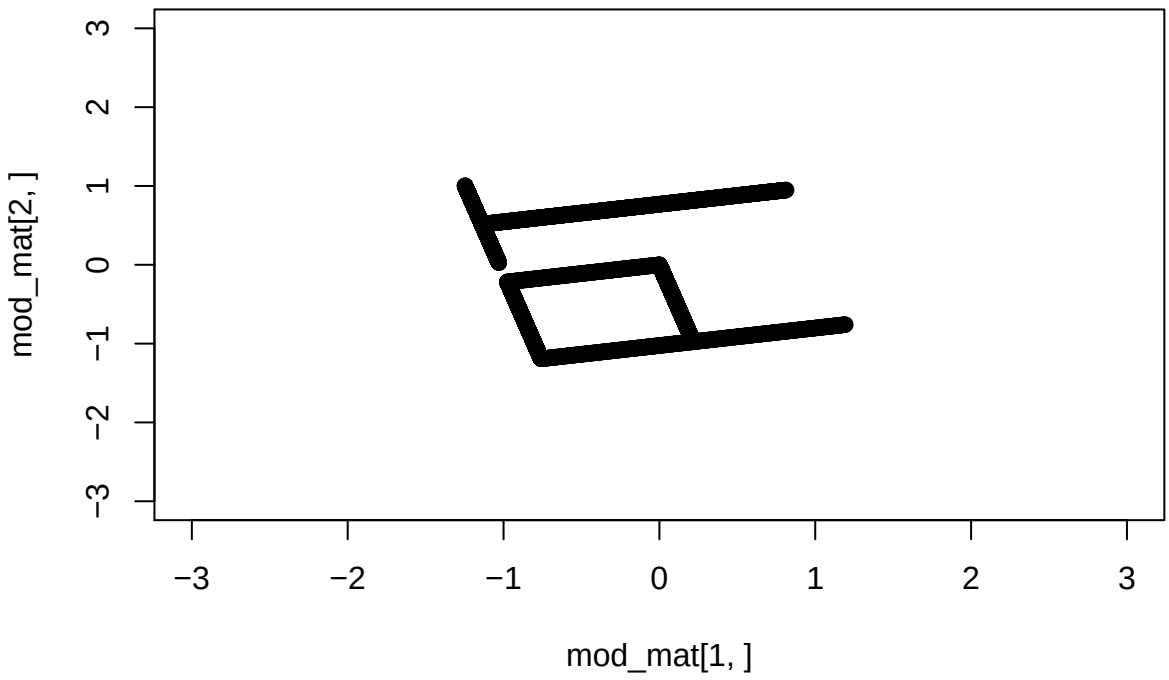

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

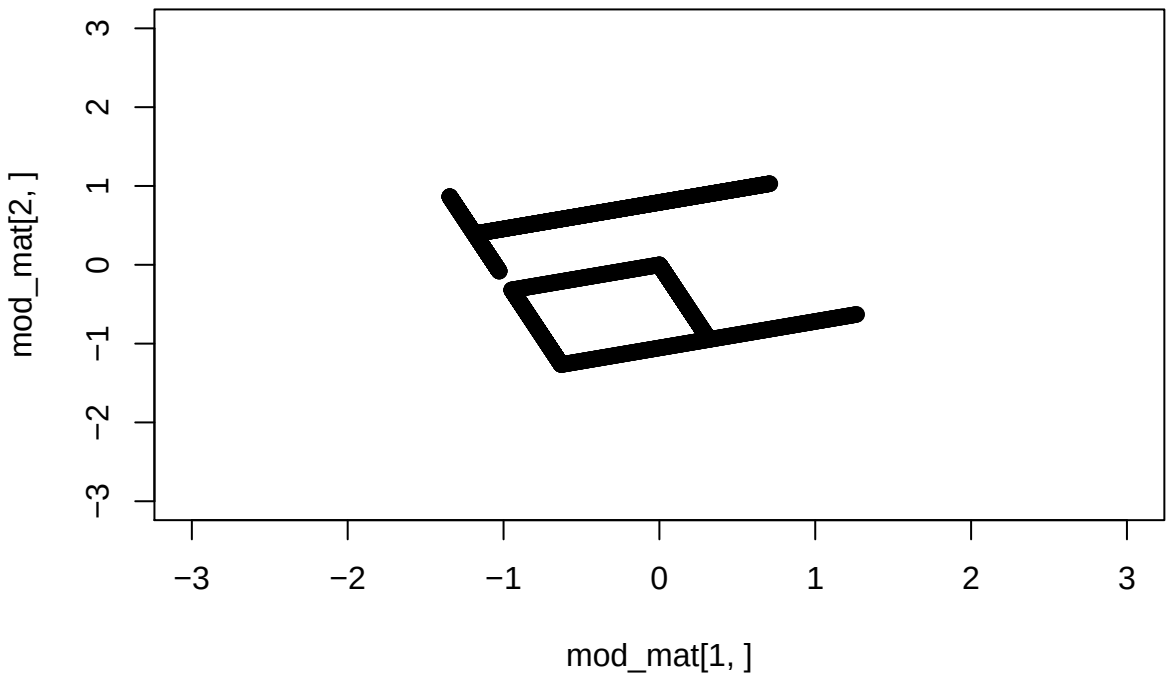

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]

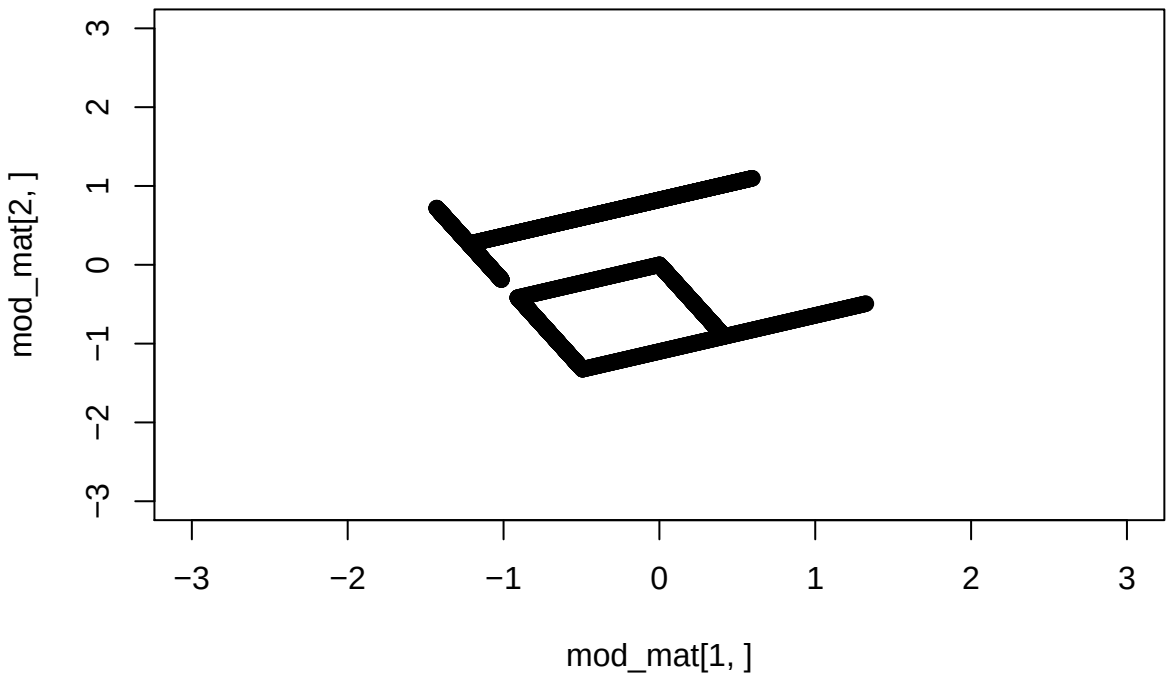

mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]


mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]


mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]


mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]


mod_mat[2,]
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
mod_mat[1,]




```
saveGIF(ani.replay(), img.name = "initials_animation")
```

```
## Output at: animation.gif
```

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

Initial Solution Prior to Meetup

Including this solution I had which I believe technically met the criteria of the Assignment 1 prompt, but was inadequate after seeing the presentation during the first meet-up. Leaving here for posterity.

```
# PT
x=c(rep(-1,500), seq(-1,0,length.out=1000), seq(-1,0,length.out=1000),
    rep(0,250), seq(0.25,.75,length.out=500), rep(.75,1000), seq(.75,1.25,length.out=500))

y=c(seq(-1,1,length.out=500), rep(0,1000), rep(1,1000), seq(0,1,length.out=250),
    rep(1,500), seq(-1,1,length.out=1000), rep(1,500))

z=rbind(x,y)

# Confirm the 'PT' is showing based on prompt code
plot(y~x, xlim=c(-3,3), ylim=c(-3,3))

# Create Identity Matrix
```

```
id_matrix <- diag(2)

# Create matrix for shear
shear_matrix <- matrix(c(1, 0, 1.2, 1), nrow=2, ncol=2)

# Create matrix for scale
scale_matrix <- matrix(c(2, 0, 0, 2), nrow=2, ncol=2)

# Create matrix for rotation (90 degrees counterclockwise)
rot_matrix <- matrix(c(0, 1, -1, 0), nrow=2, ncol=2)

# Create matrix for projection (projection on the y axis)
# Projection of 2D to 1D
proj_matrix <- matrix(c(0, 1, 0, 1), nrow=2, ncol=2)

# Make list of the transformation matrices to allow for looping
transforms <- list(id_matrix, shear_matrix, scale_matrix, rot_matrix, proj_matrix)

# Use x11 for Animation
x11(title="Animation")

# Loop the transformation matrices
# 5 displays total
for (idx in 1:5) {

  # Matrix multiple (apply transformation)
  z <- transforms[[idx]] %*% z

  # Display resulting plot
  plot(z[2,]~z[1,], xlim=c(-10,10), ylim=c(-10,10))

  # Add sleep so the animation will be easier to watch
  Sys.sleep(0.5)
}
```