

# DATA 605 HW 1

Daniel Moscoe

1/31/2021

Create first initial, D:

```
Dx = c(rep(0, 1000), seq(0, 1, length.out = 500), seq(1, 1.5, length.out = 350),
      rep(1.5, 500), seq(1.5, 1, length.out = 350), seq(1, 0, length.out = 500))

Dy = c(seq(-1, 1, length.out = 1000), rep(1, 500), seq(1, 0.5, length.out = 350),
      seq(0.5, -0.5, length.out = 500), seq(-0.5, -1, length.out = 350), rep(-1, 500))

Dz = rbind(Dx, Dy)
```

Create second initial, M:

```
Mx = c(rep(0, 1000), seq(0, 1, length.out = 700), seq(1, 2, length.out = 700), rep(2,
    1000))

My = c(seq(-1, 1, length.out = 1000), seq(1, 0, length.out = 700), seq(0, 1, length.out = 700),
      seq(1, -1, length.out = 1000))

Mz = rbind(Mx, My)
```

Plotting and transformation functions:

```
plotter <- function(x) {
  plot(x[2, ] ~ x[1, ], xlim = c(-3, 3), ylim = c(-3, 3))
}

shear <- function(x, f) {
  # shears x in the positive x direction with shear element f.
  leftMatrix <- matrix(c(1, 0, f, 1), nrow = 2, ncol = 2)
  leftMatrix %*% x
}

scale <- function(x, f) {
  # scales x by the factor f.
  leftMatrix <- matrix(c(f, 0, 0, f), nrow = 2, ncol = 2)
  leftMatrix %*% x
```

```

}

rotate <- function(x, theta) {
  # rotates x about (0,0) by theta in radians.
  leftMatrix <- matrix(c(cos(theta), sin(theta), -sin(theta), cos(theta)), nrow = 2,
    ncol = 2)
  leftMatrix %*% x
}

project <- function(x, slope) {
  # projects x onto the line containing (0, 0) with slope slope.
  leftMatrix <- matrix(c(1/(1 + slope^2), slope/(1 + slope^2), slope/(1 + slope^2),
    slope^2/(1 + slope^2)), nrow = 2, ncol = 2)
  leftMatrix %*% x
}

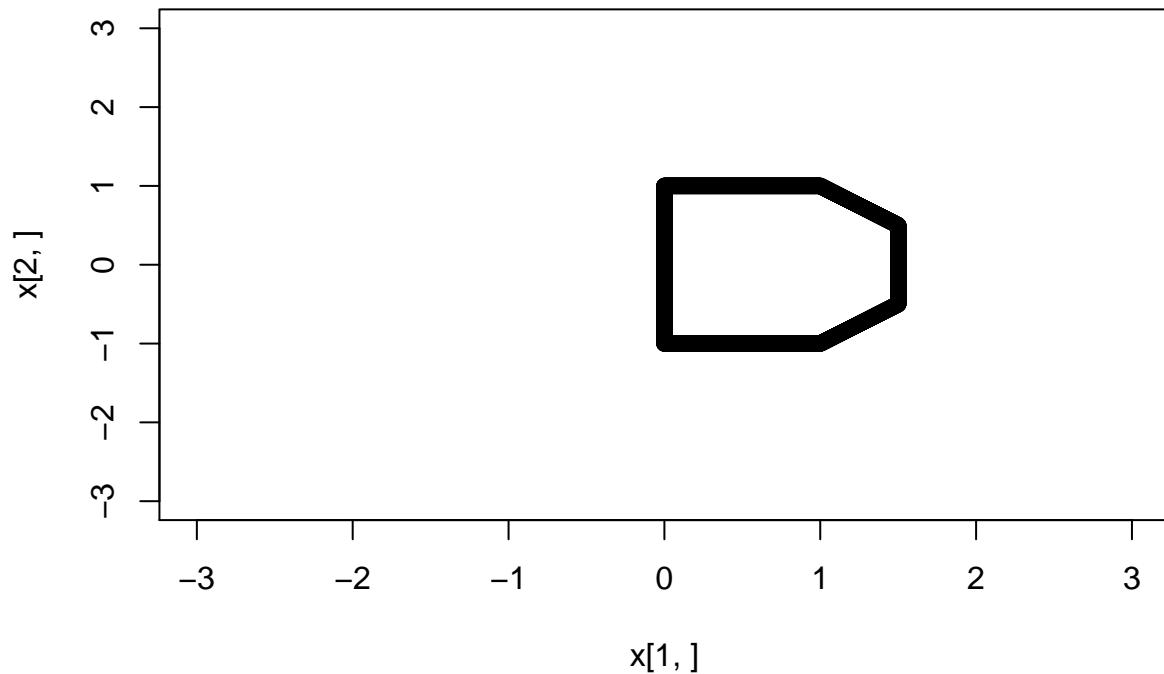
```

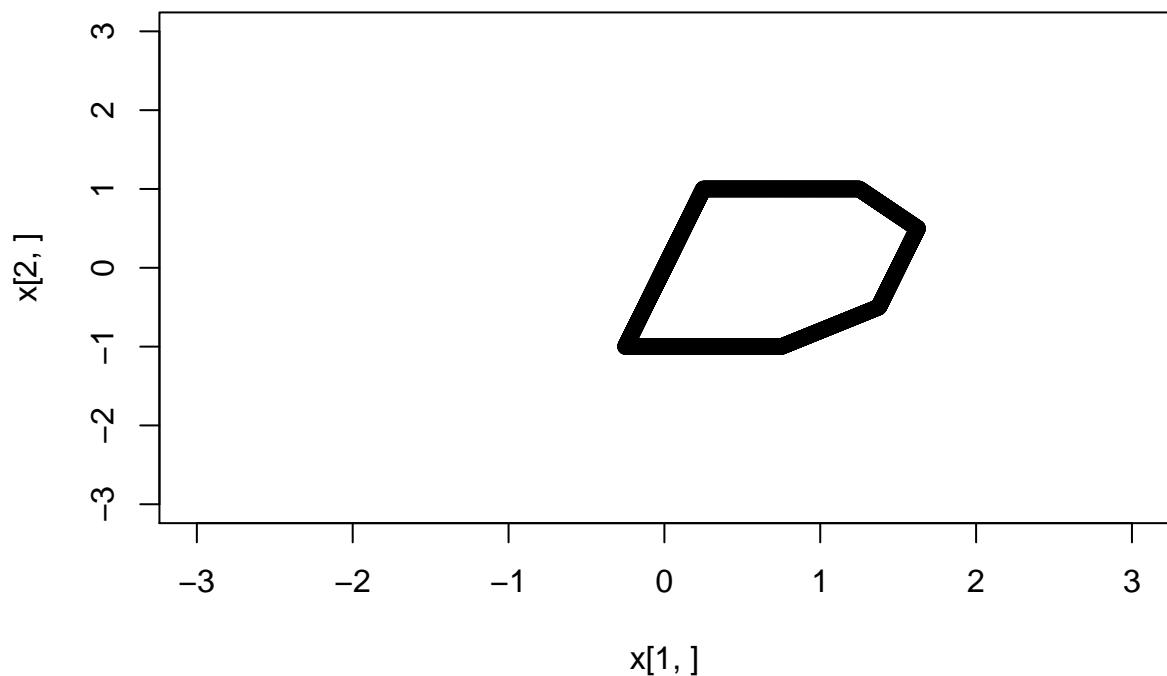
Transformations of D:

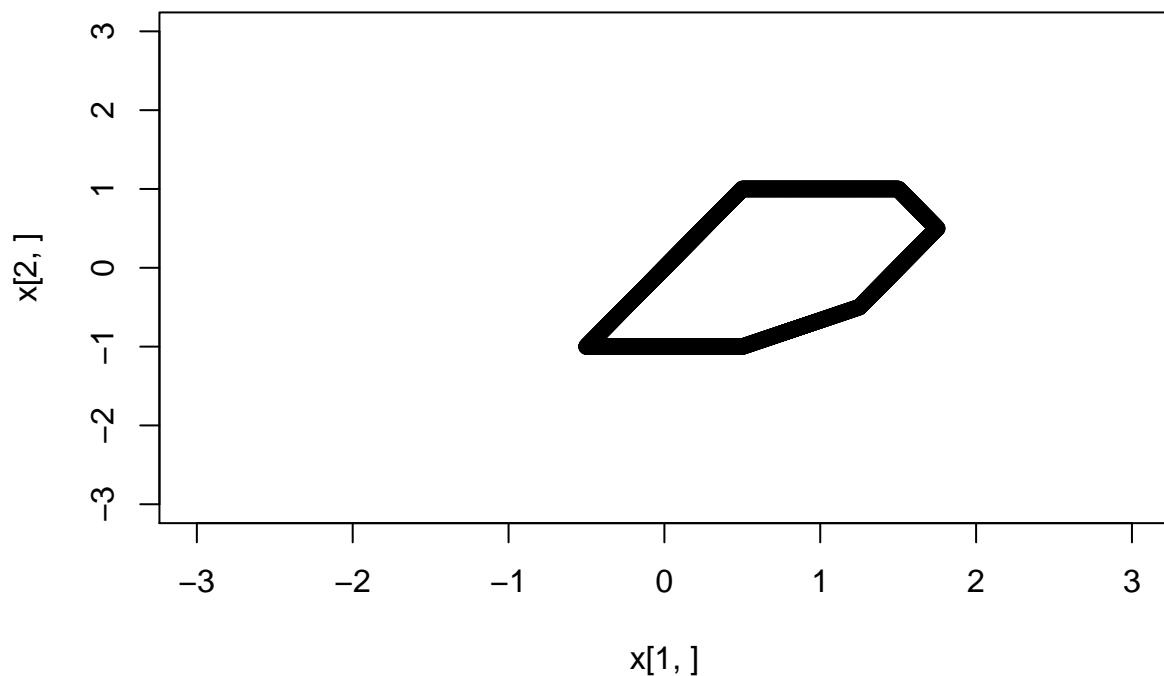
```

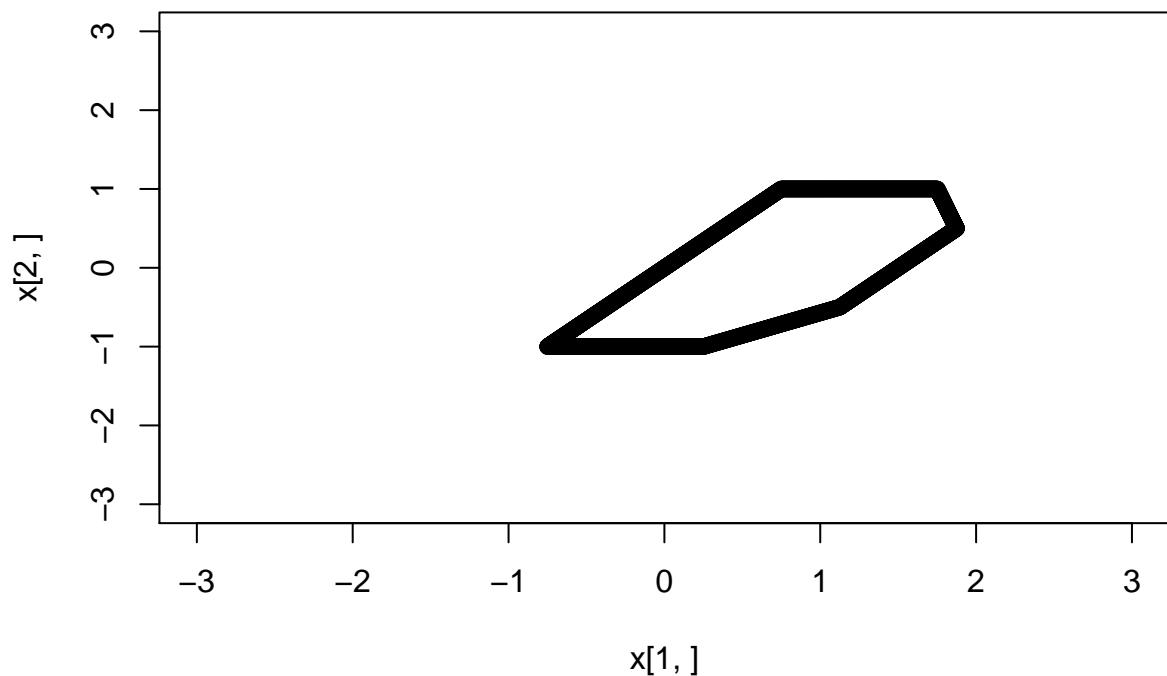
for (i in c(0, 0.25, 0.5, 0.75)) {
  shear(Dz, i) %>%
    plotter()
}

```

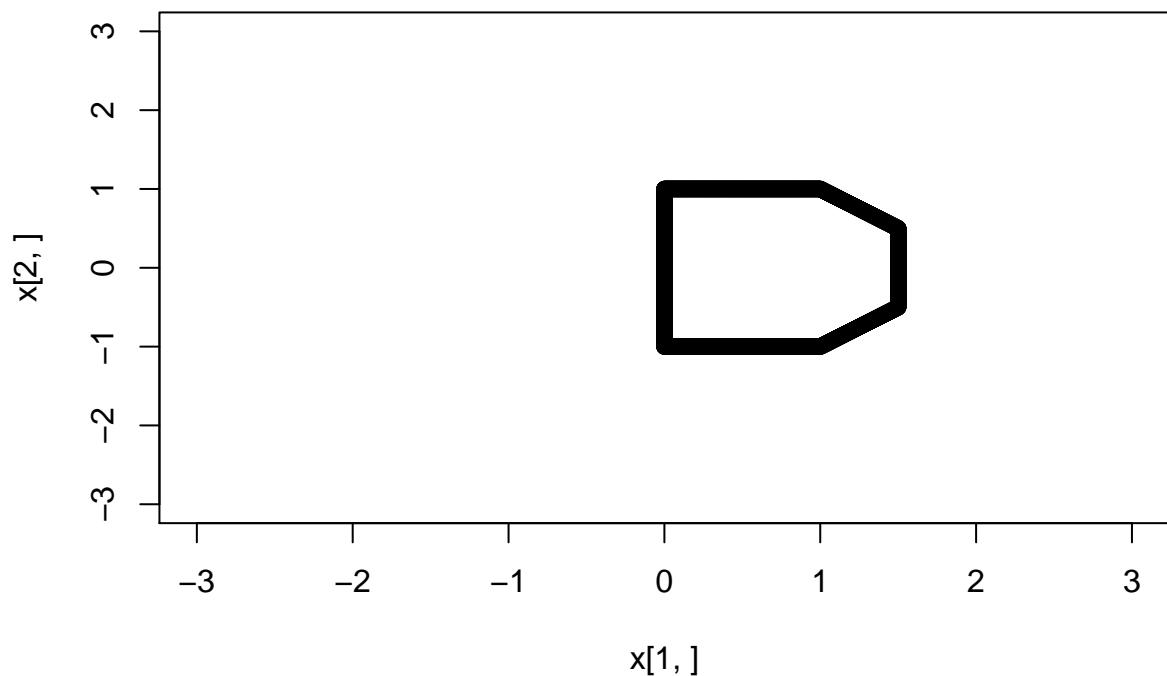


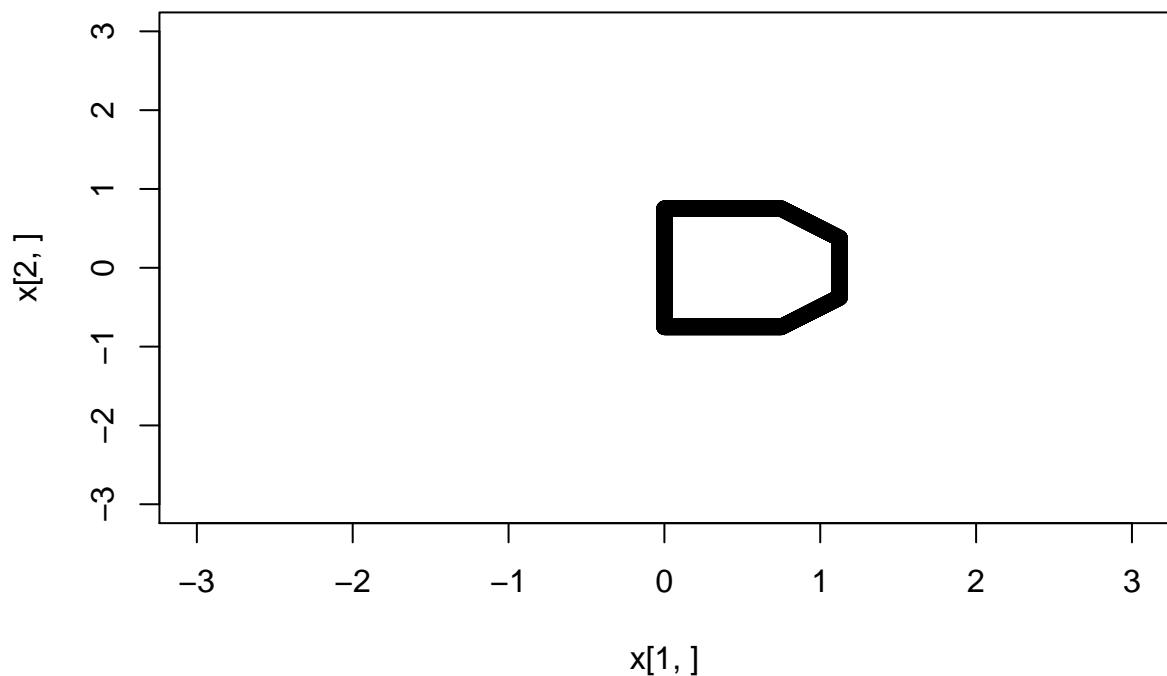


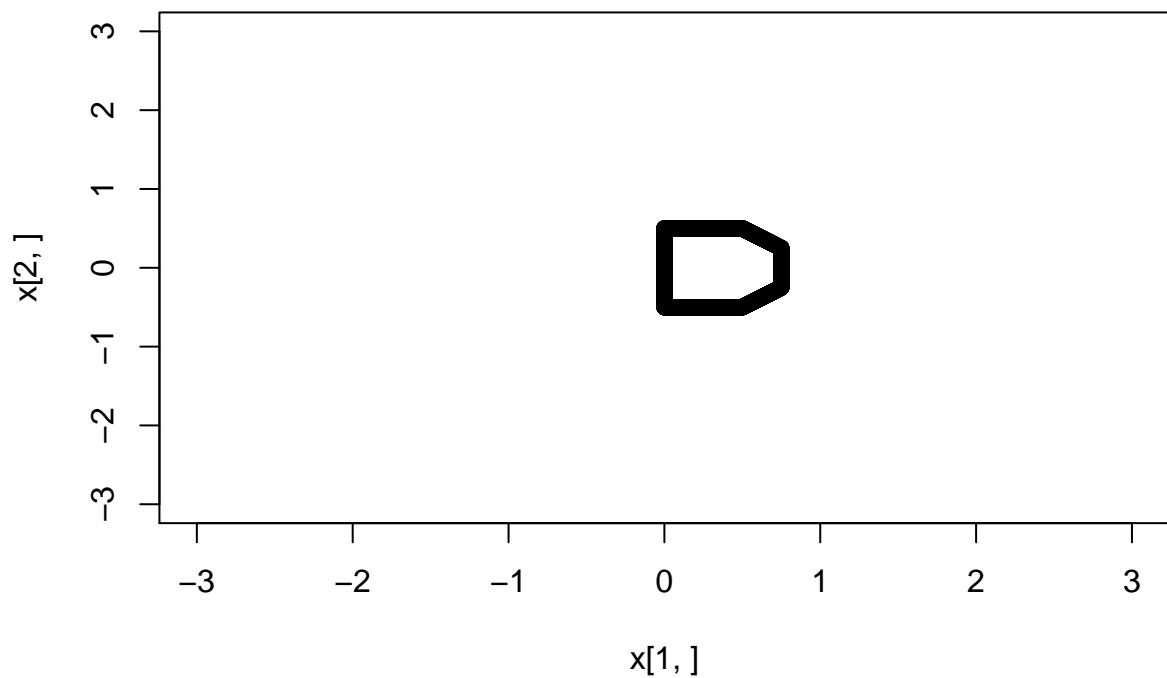


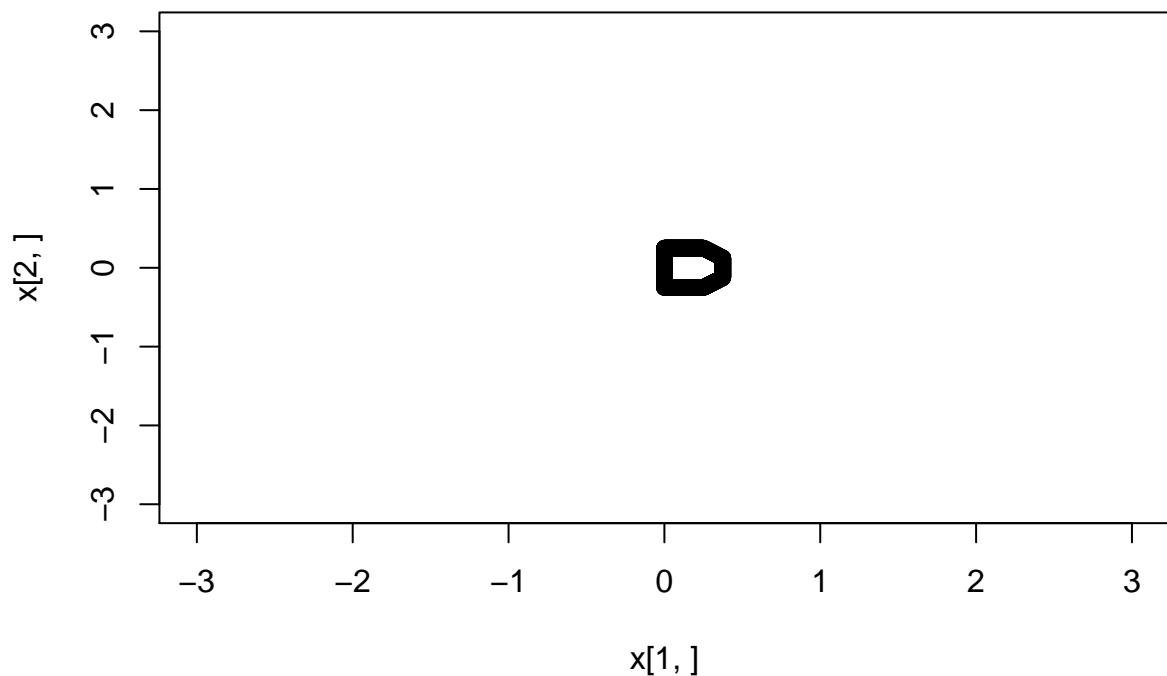


```
for (i in c(1, 0.75, 0.5, 0.25)) {  
  scale(Dz, i) %>%  
    plotter()  
}
```

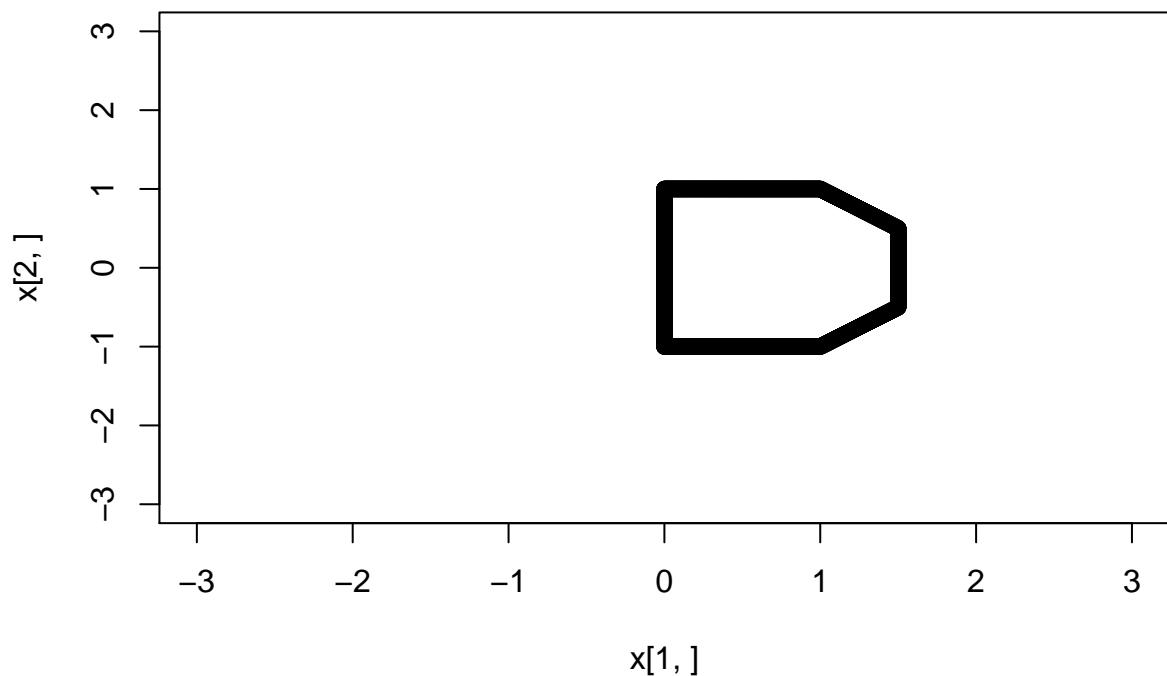


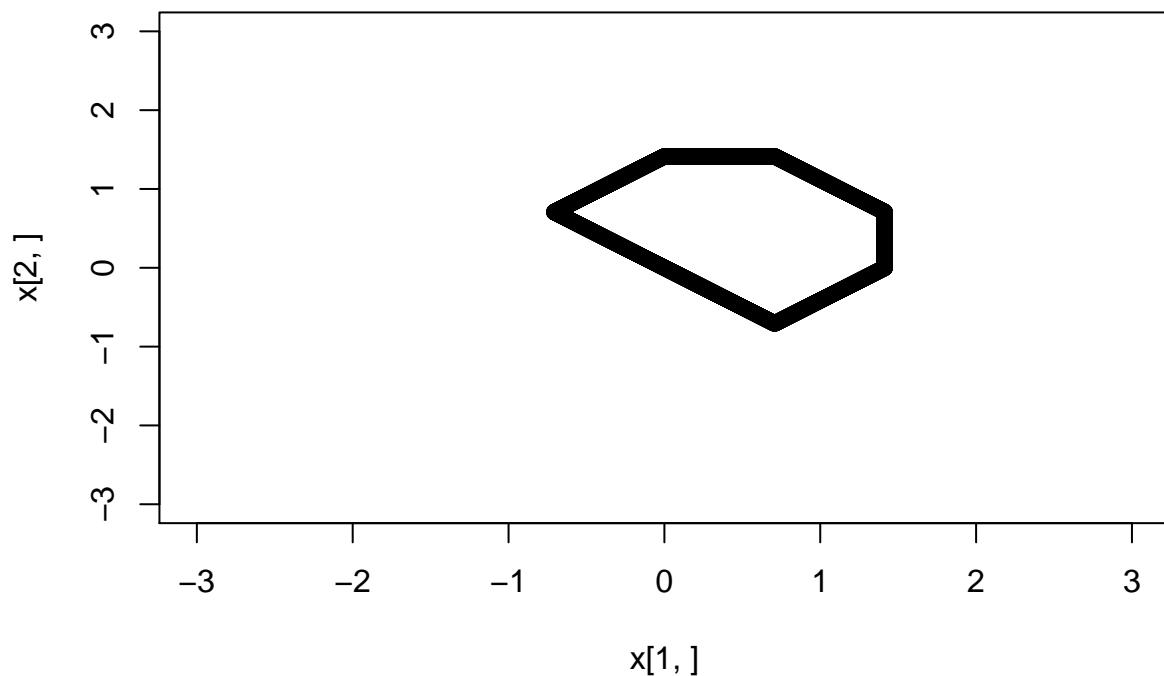


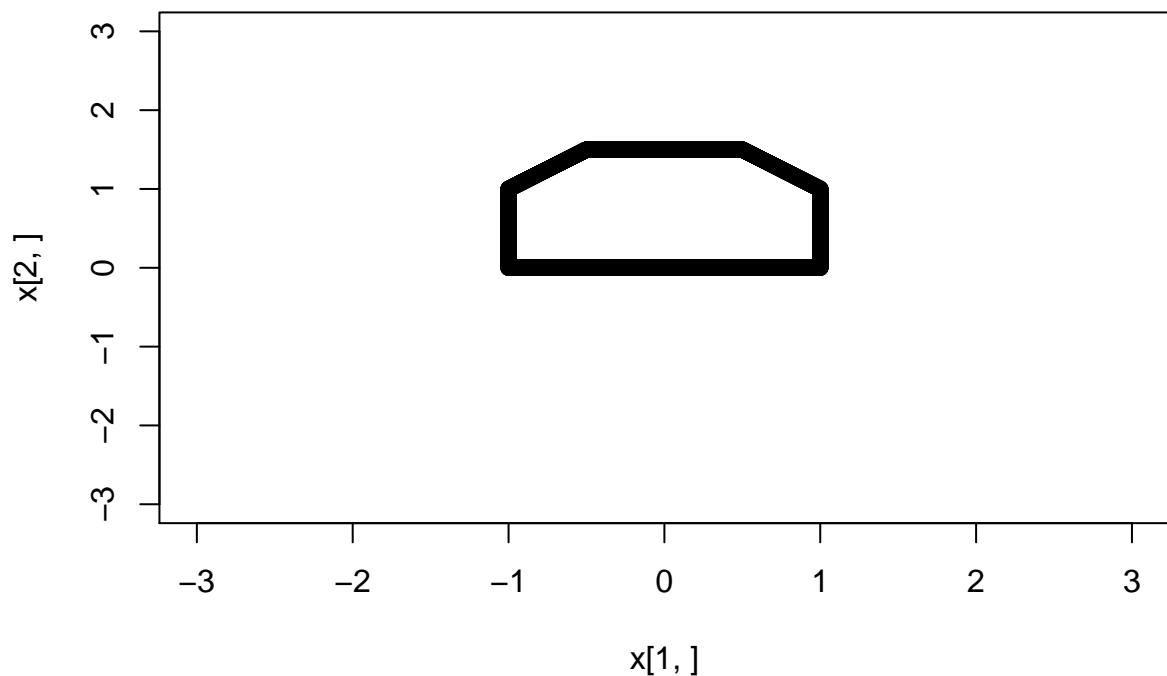


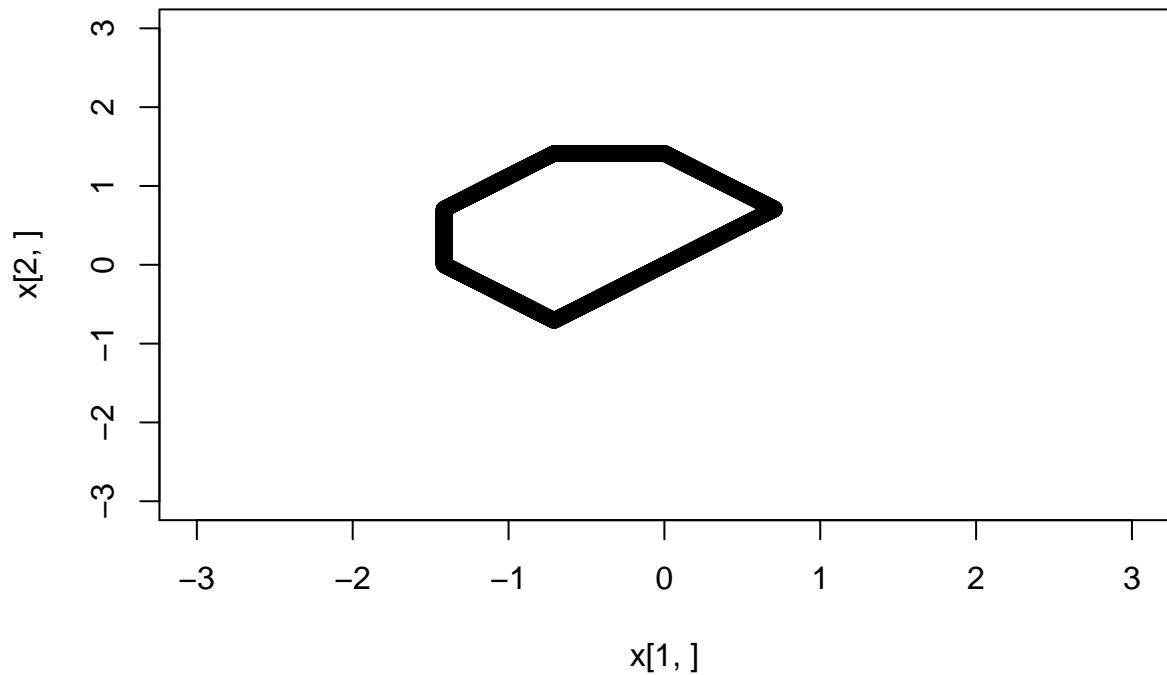


```
for (i in c(0, pi/4, pi/2, 3 * pi/4)) {  
  rotate(Dz, i) %>%  
    plotter()  
}
```

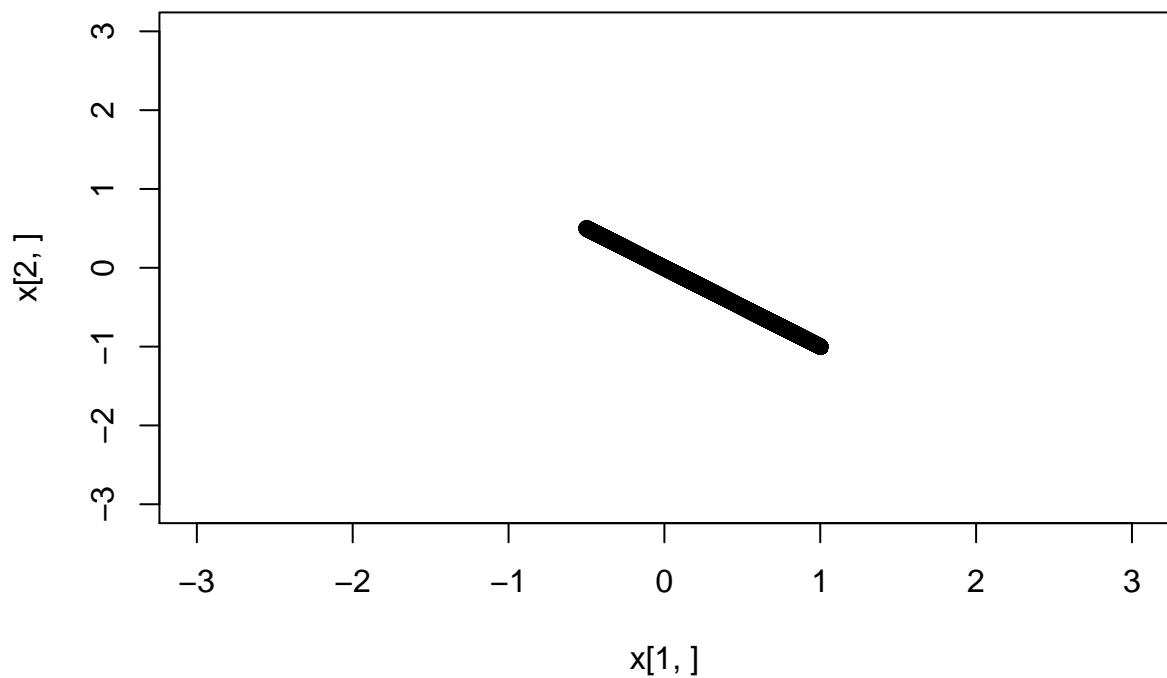


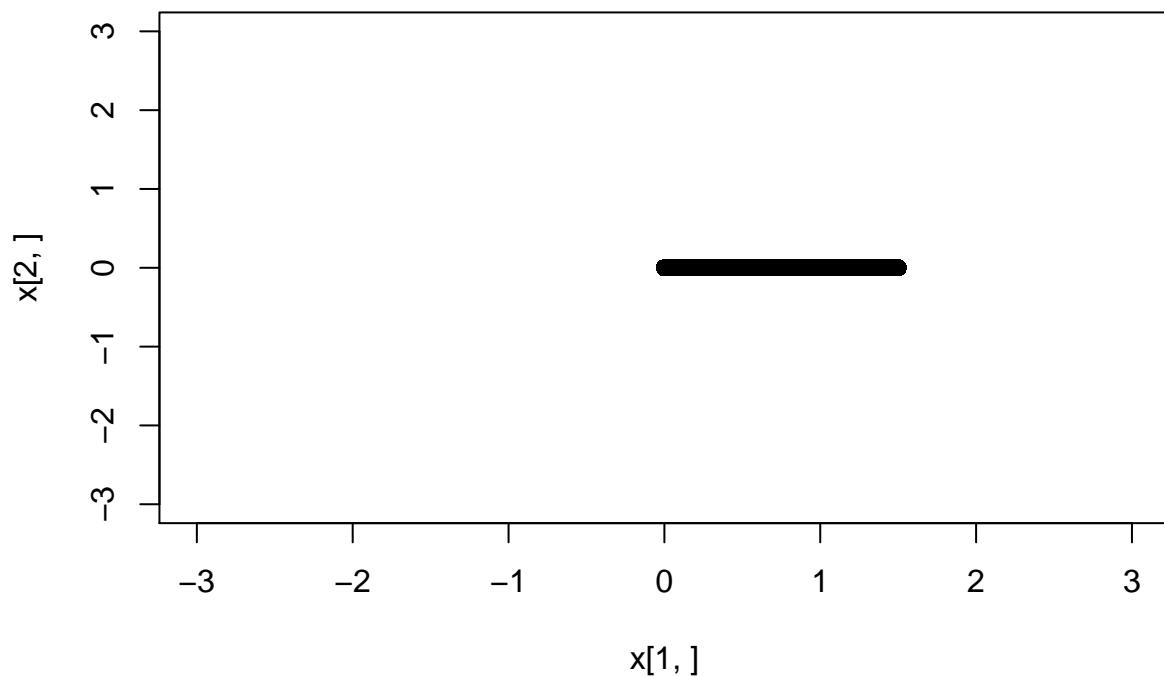


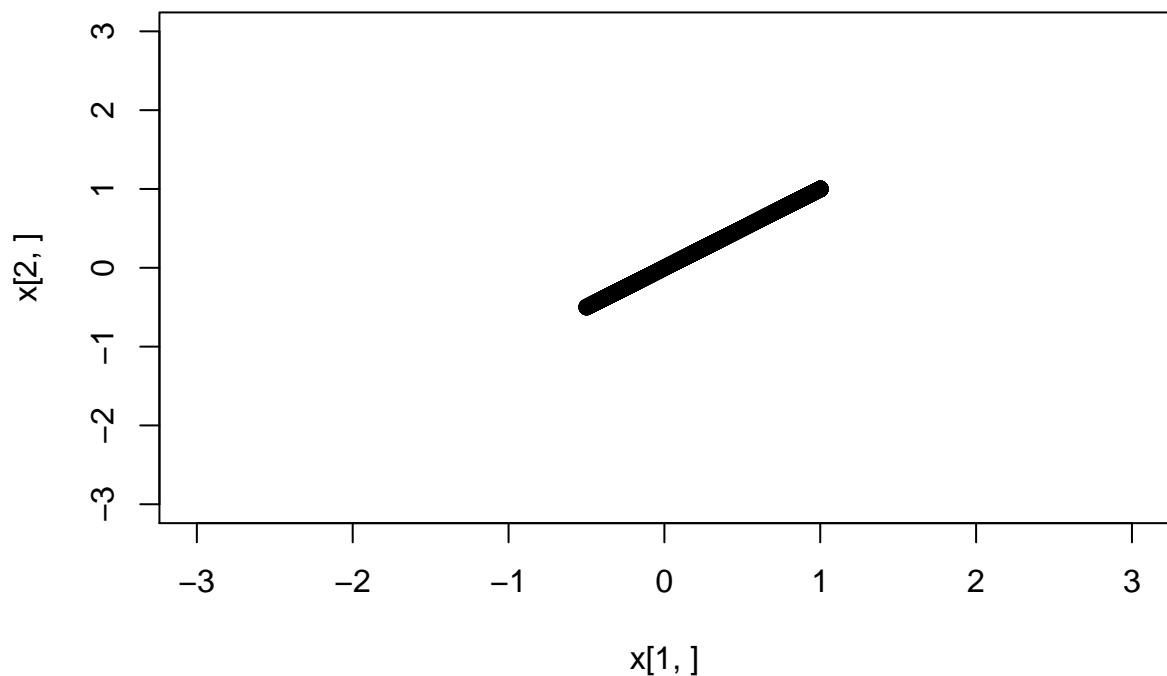


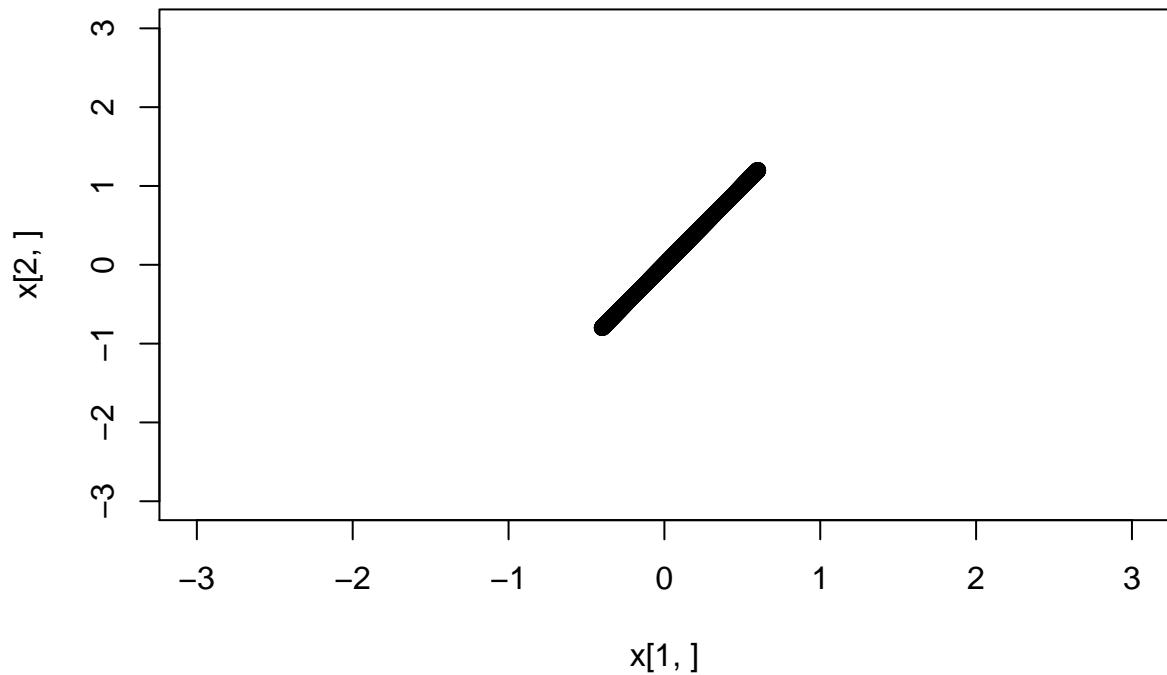


```
for (i in c(-1, 0, 1, 2)) {  
  project(Dz, i) %>%  
    plotter()  
}
```



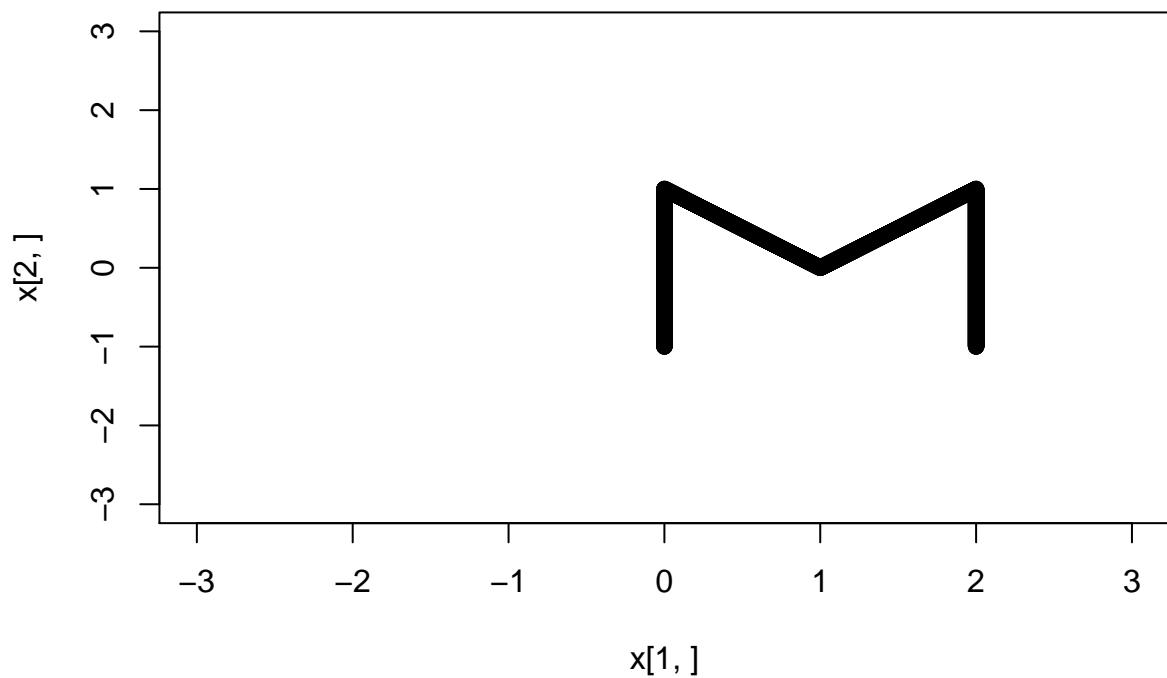


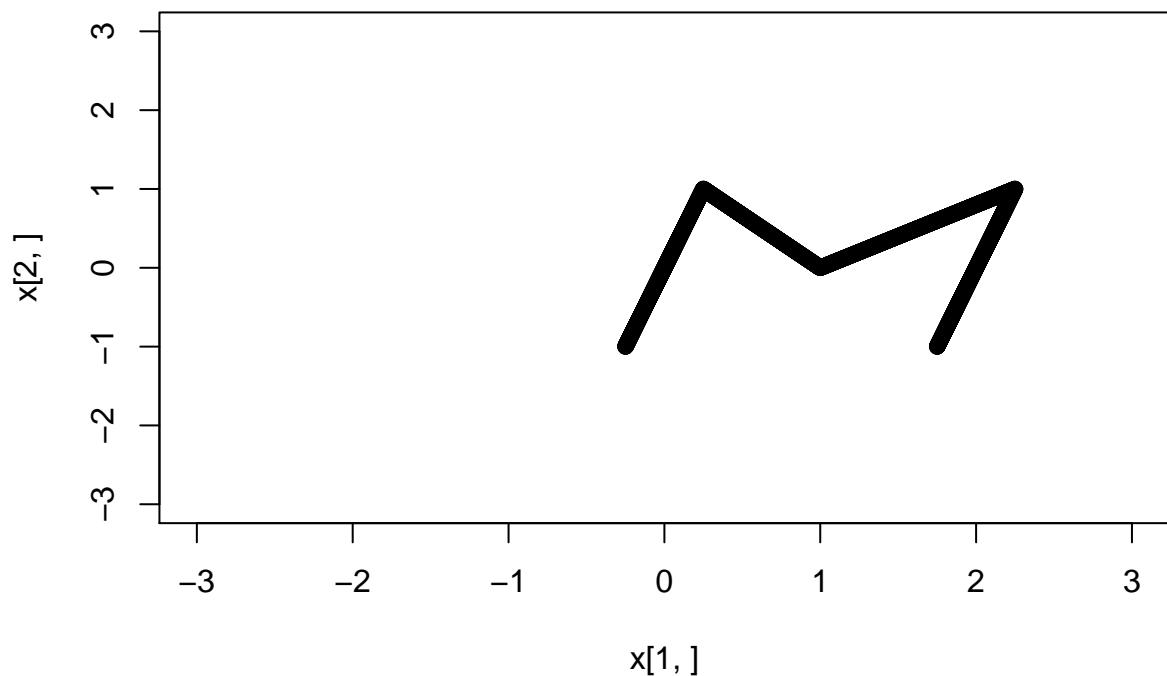


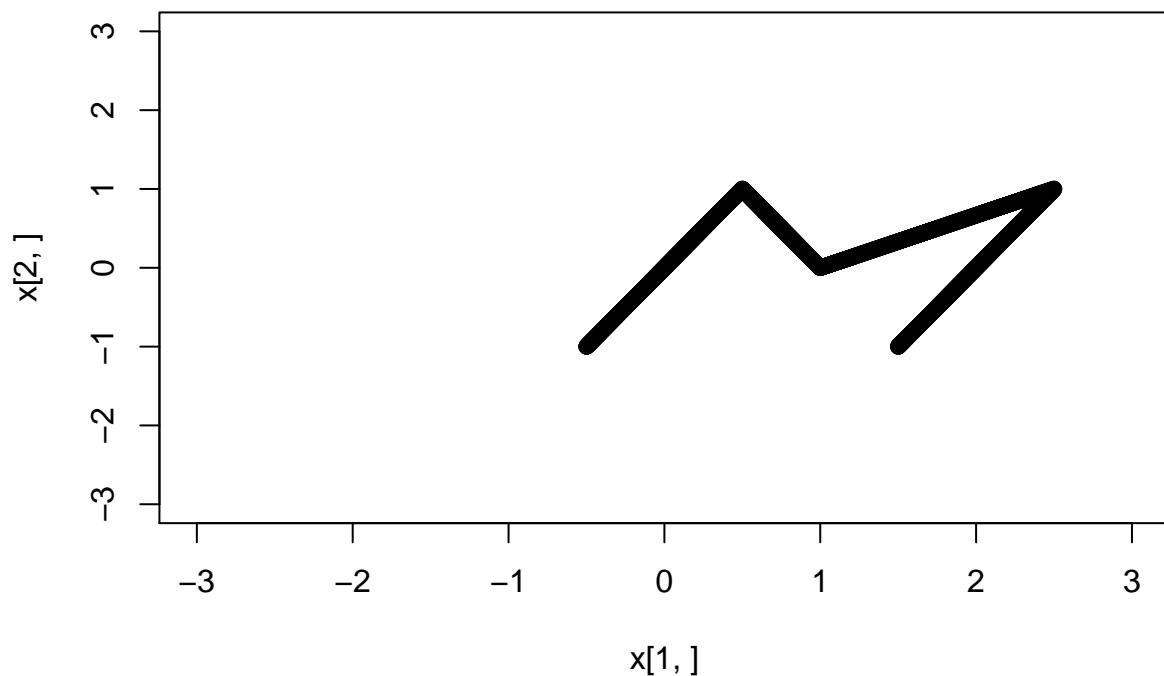


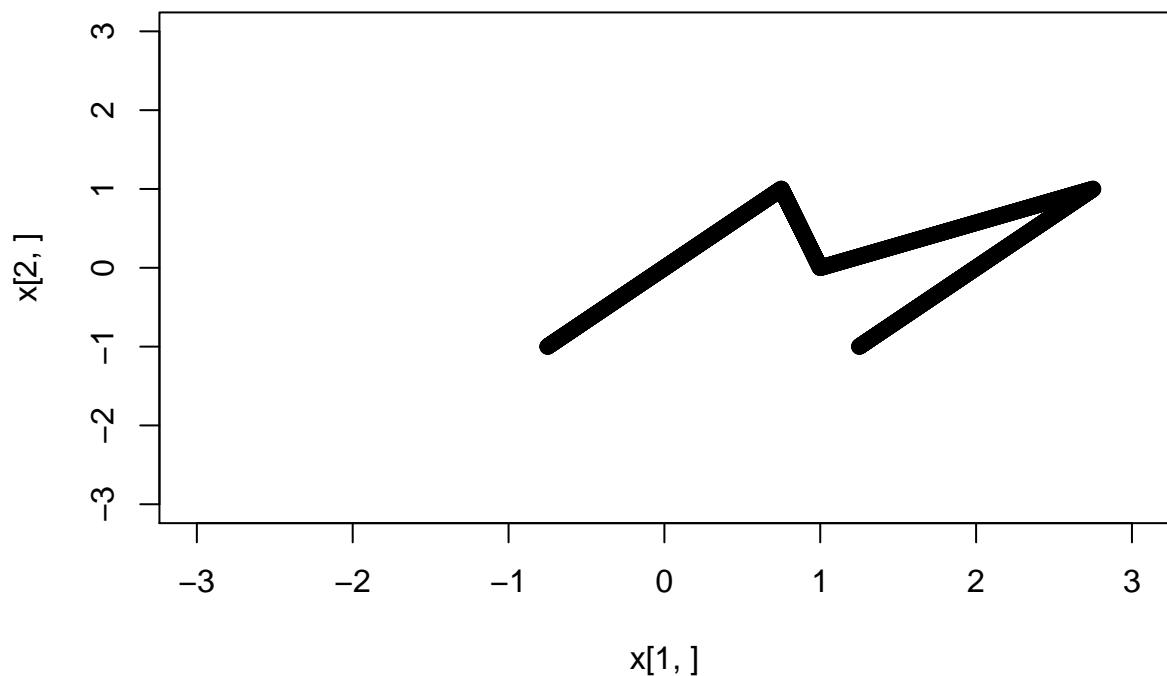
Transformations of M:

```
for (i in c(0, 0.25, 0.5, 0.75)) {  
  shear(Mz, i) %>%  
    plotter()  
}
```

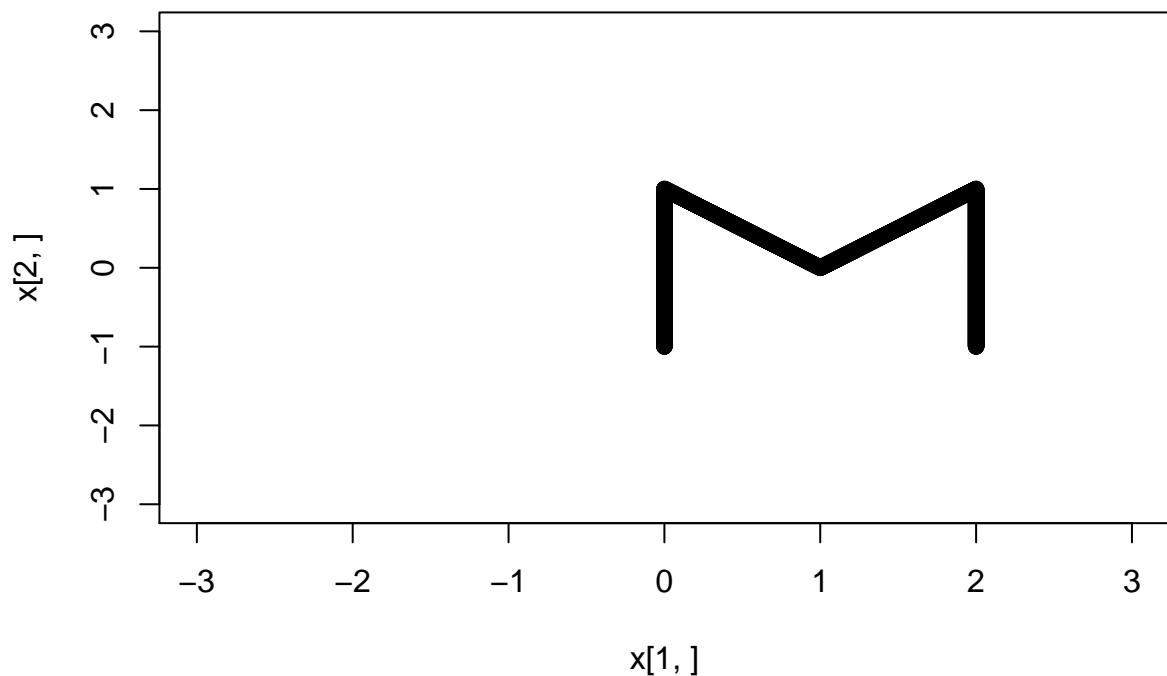


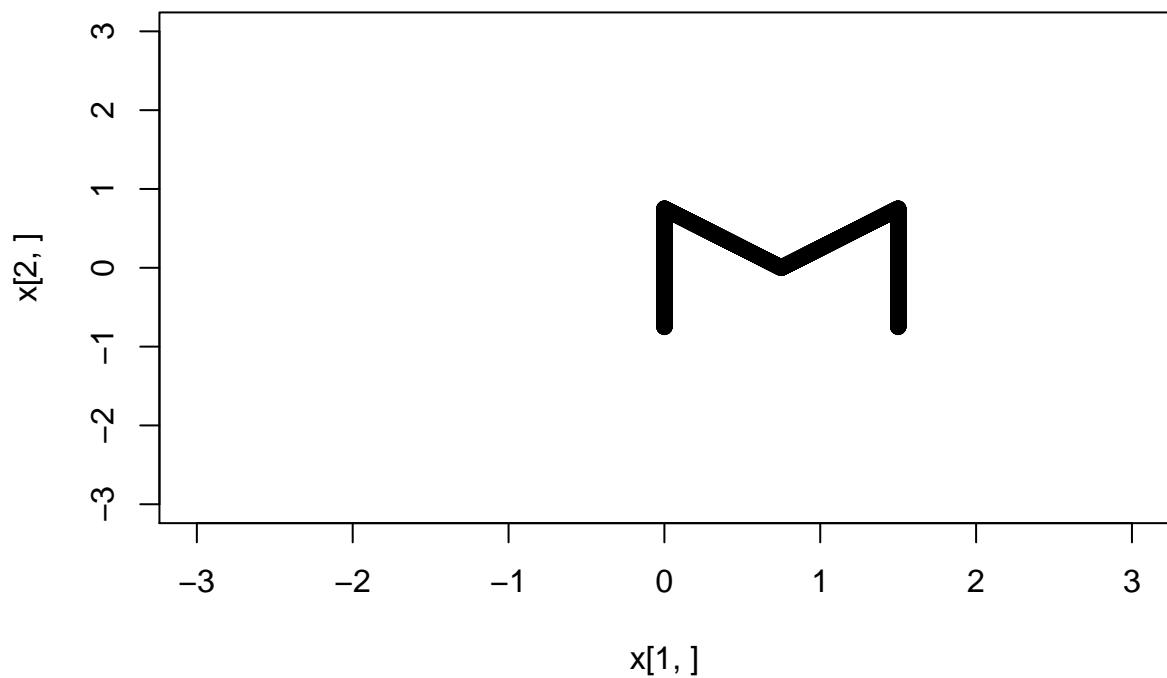


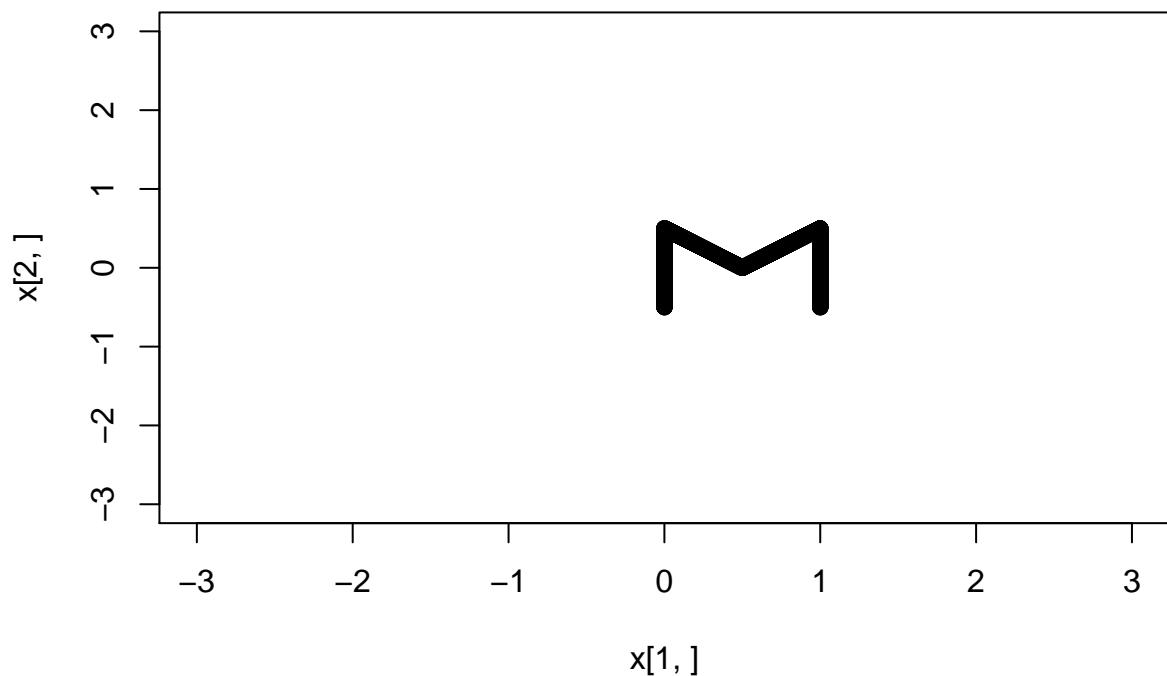


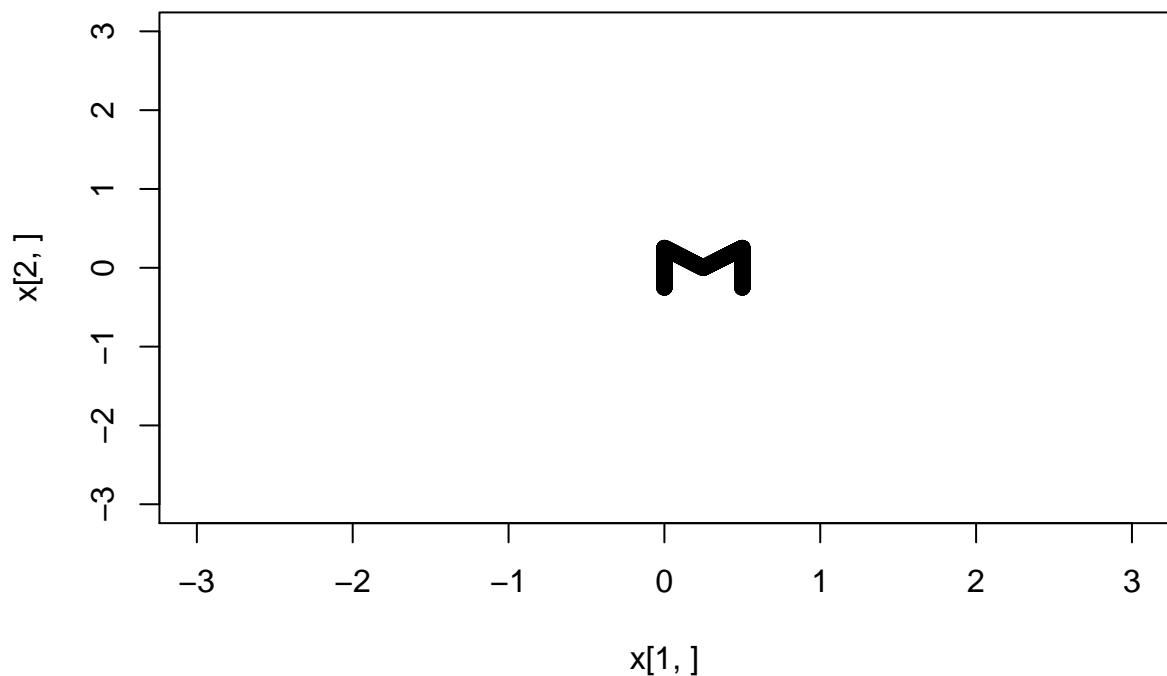


```
for (i in c(1, 0.75, 0.5, 0.25)) {  
  scale(Mz, i) %>%  
    plotter()  
}
```

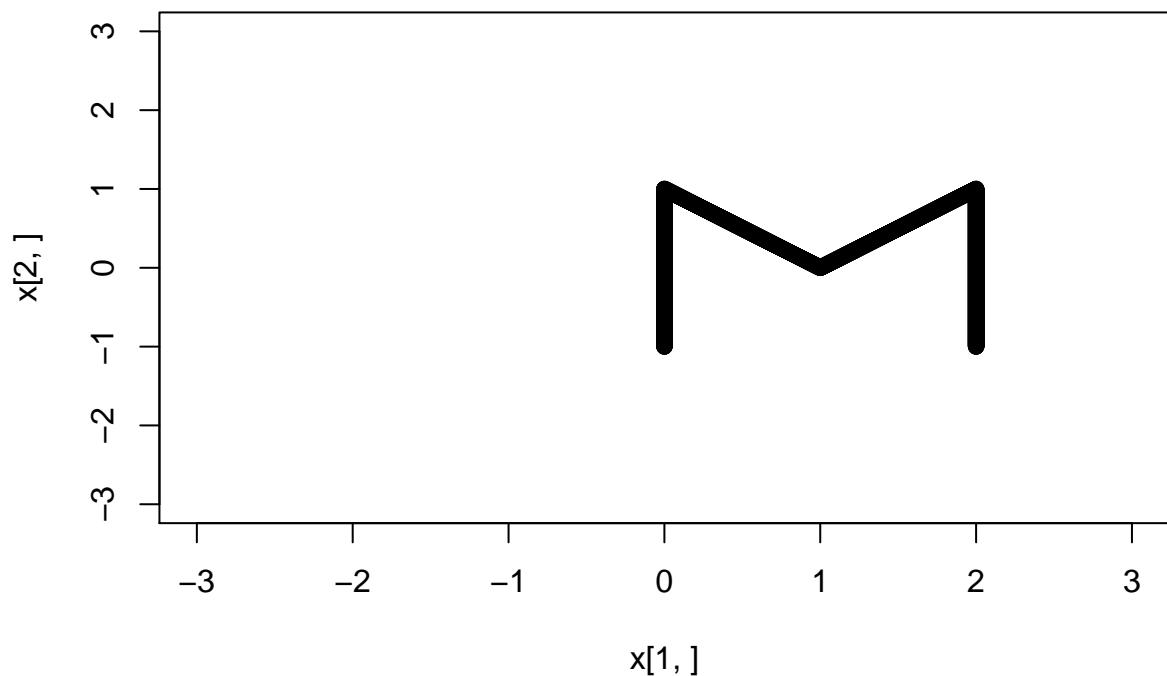


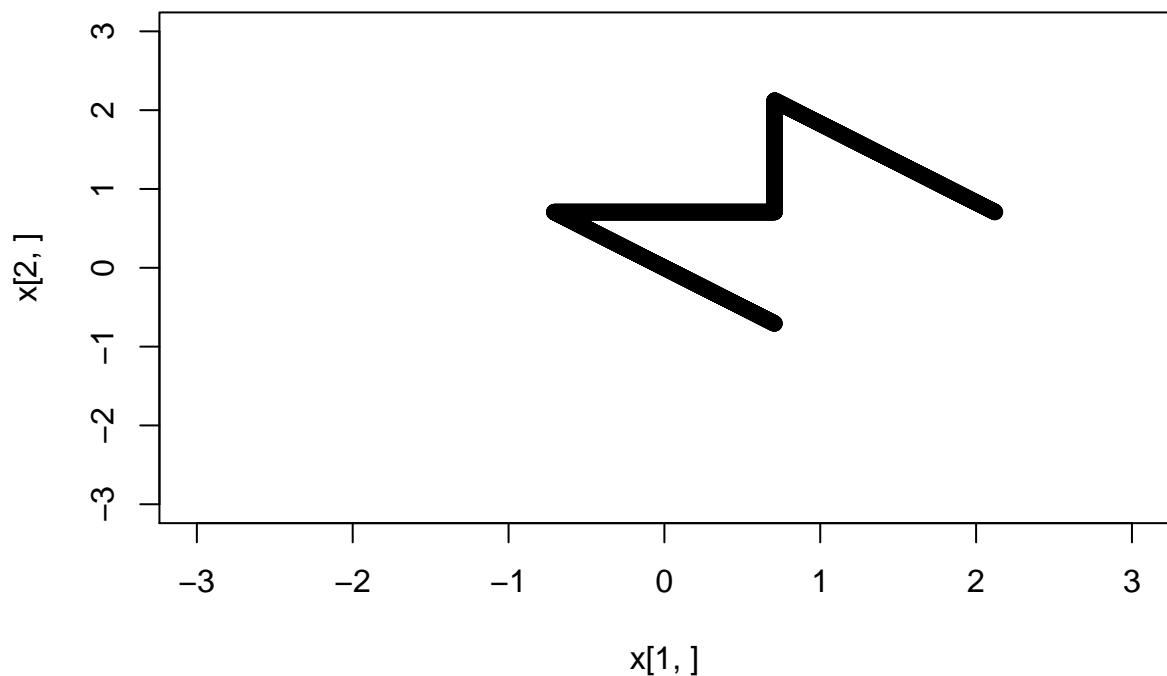


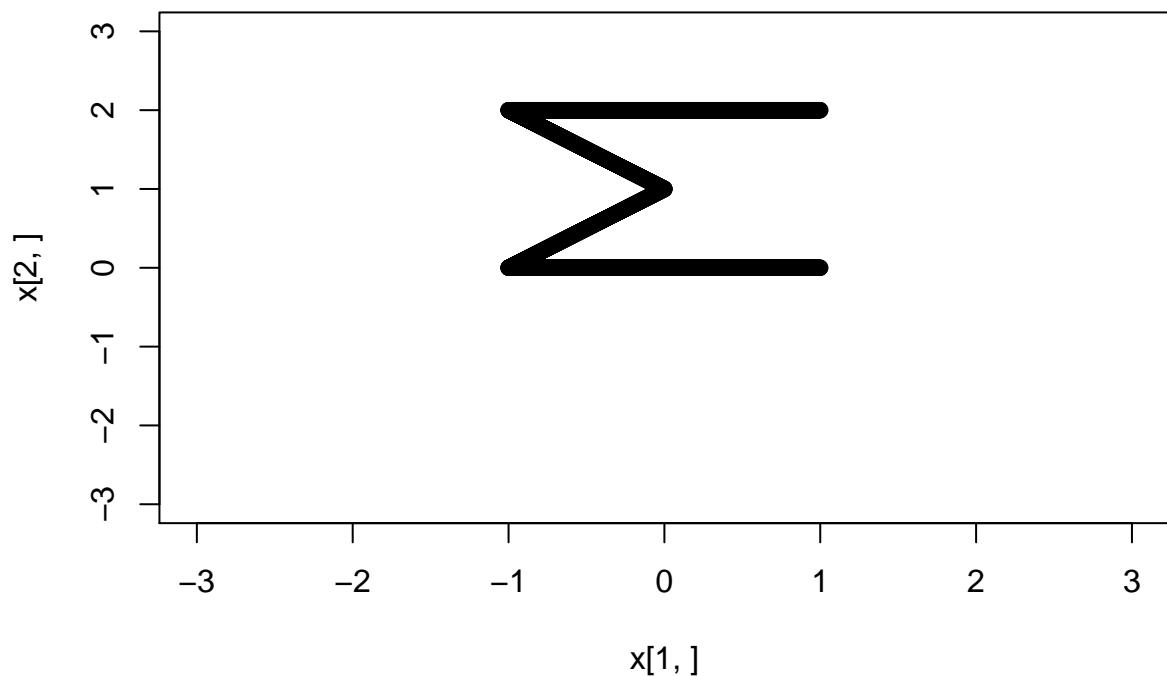


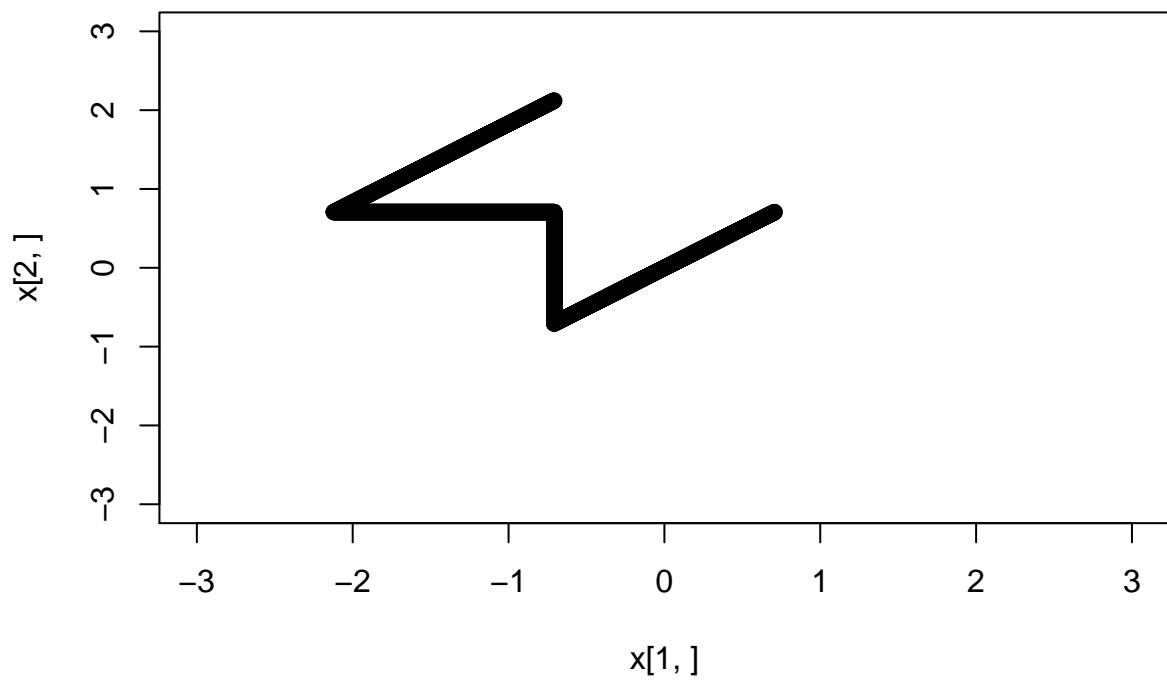


```
for (i in c(0, pi/4, pi/2, 3 * pi/4)) {  
  rotate(Mz, i) %>%  
    plotter()  
}
```

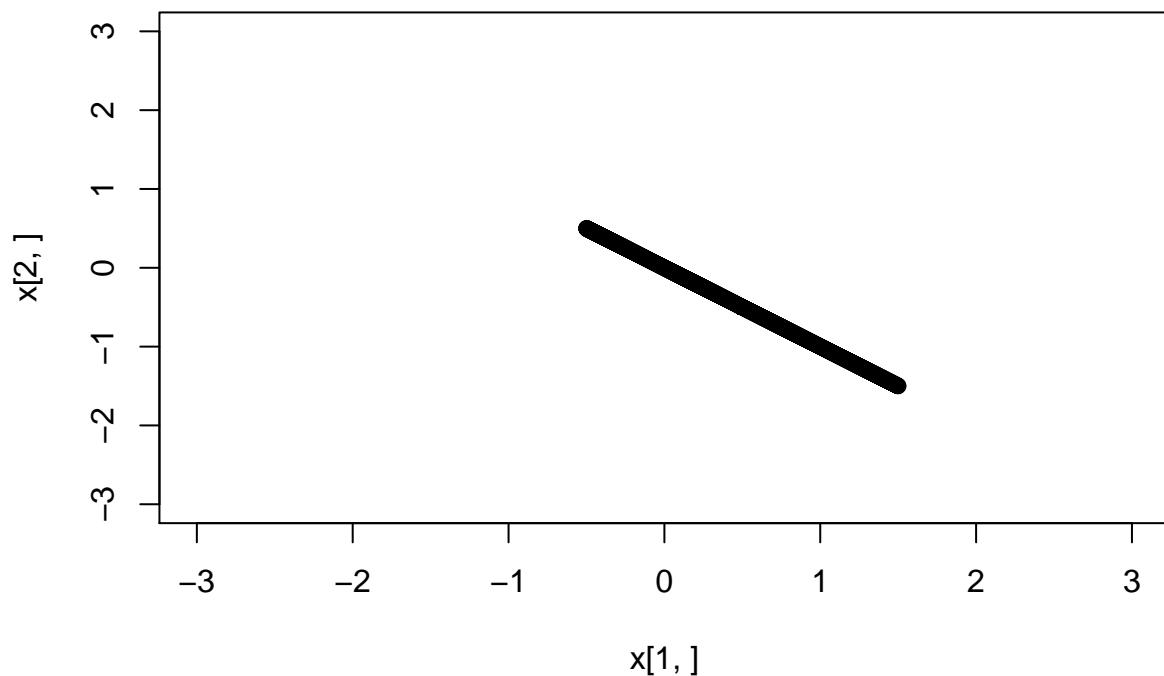


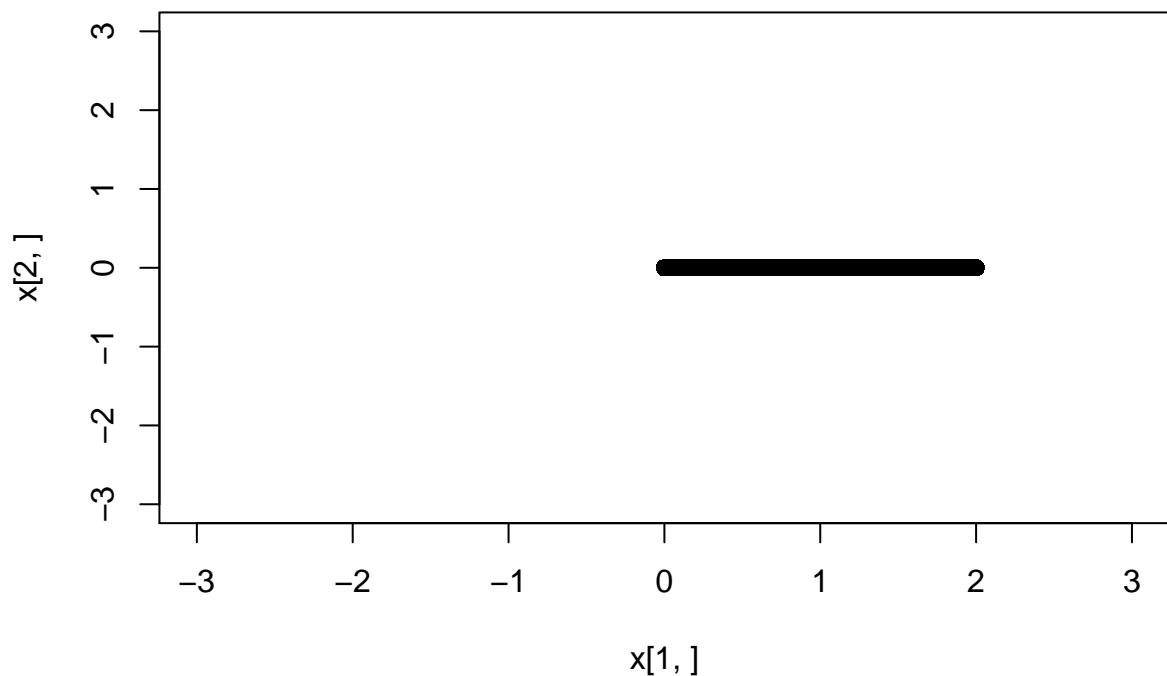


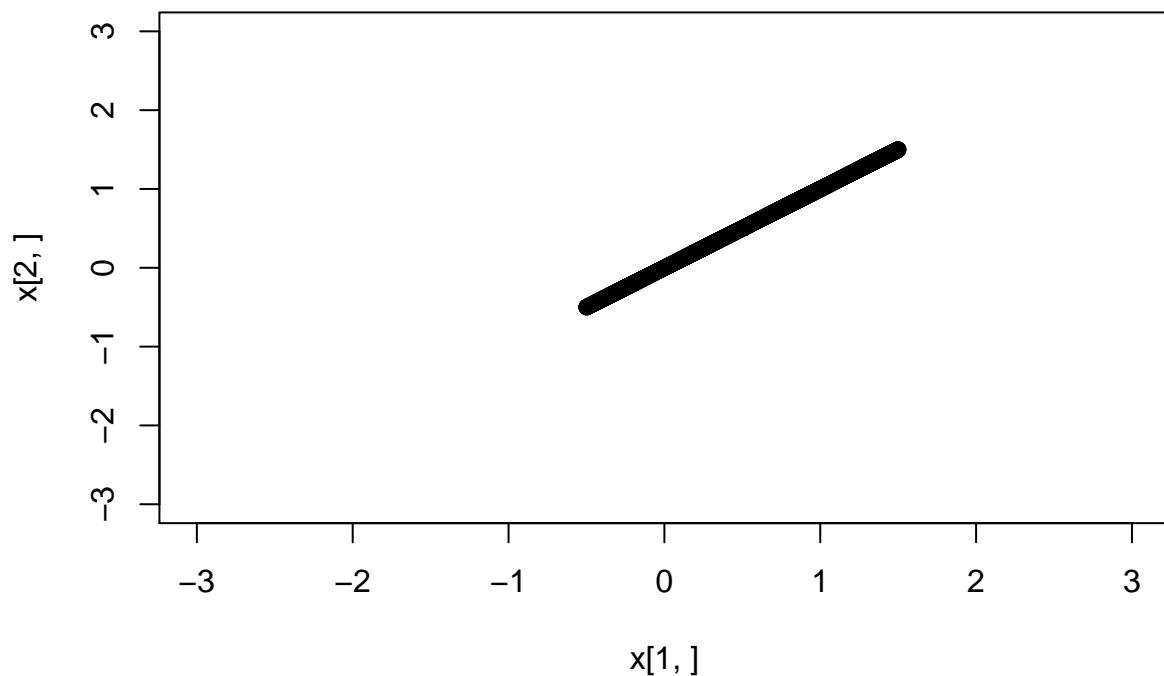


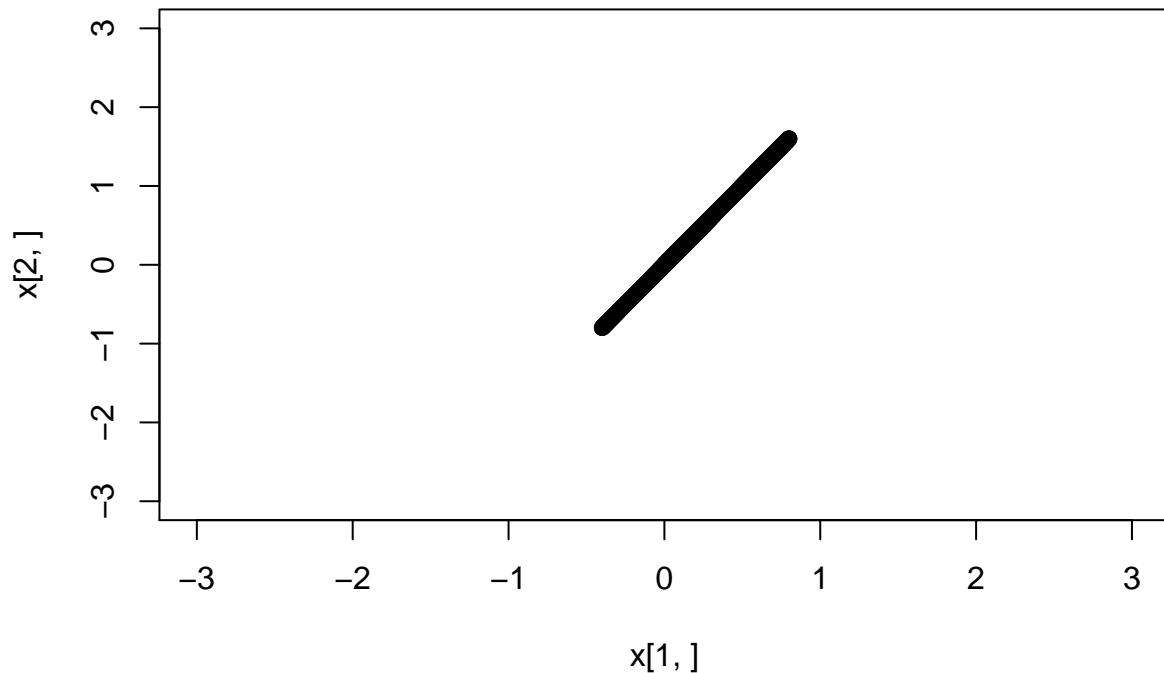


```
for (i in c(-1, 0, 1, 2)) {  
  project(Mz, i) %>%  
    plotter()  
}
```









**Notes:**

This is my first Rmd document, and it's also my first time working with matrices in R. While I'm sure there is much that could be improved about this work, I'm proud of it as a starting point.

The assignment indicated that each series of transformations should begin with the identity matrix, but I don't think this is possible for orthogonal projections. Is it?

The assignment also indicated the plots should appear as animations, and that these animations could be produced using the function `x11()`. I was not able to figure this out. Is it possible to include animations in the knitted output of an Rmd file?