

# HW1

## FUNDAMENTALS OF COMPUTATIONAL MATHEMATICS

Zachary Palmore

1/31/2021

### Homework 1

For week one January 29 - January 31 we were tasked with designing our initials in a plot and performing transformations on them using matrix multiplication and other operations to each of the point vectors. This assignment was open ended about how the final product should look and could have used various methods to reach it, as long as the transformations appeared to run correctly and a loop was used.

#### Directions

(HW=Homework!)

One of the most useful applications for linear algebra in data science is image manipulation. We often need to compress, expand, warp, skew, etc. images. To do so, we left multiply a transformation matrix by each of the point vectors.

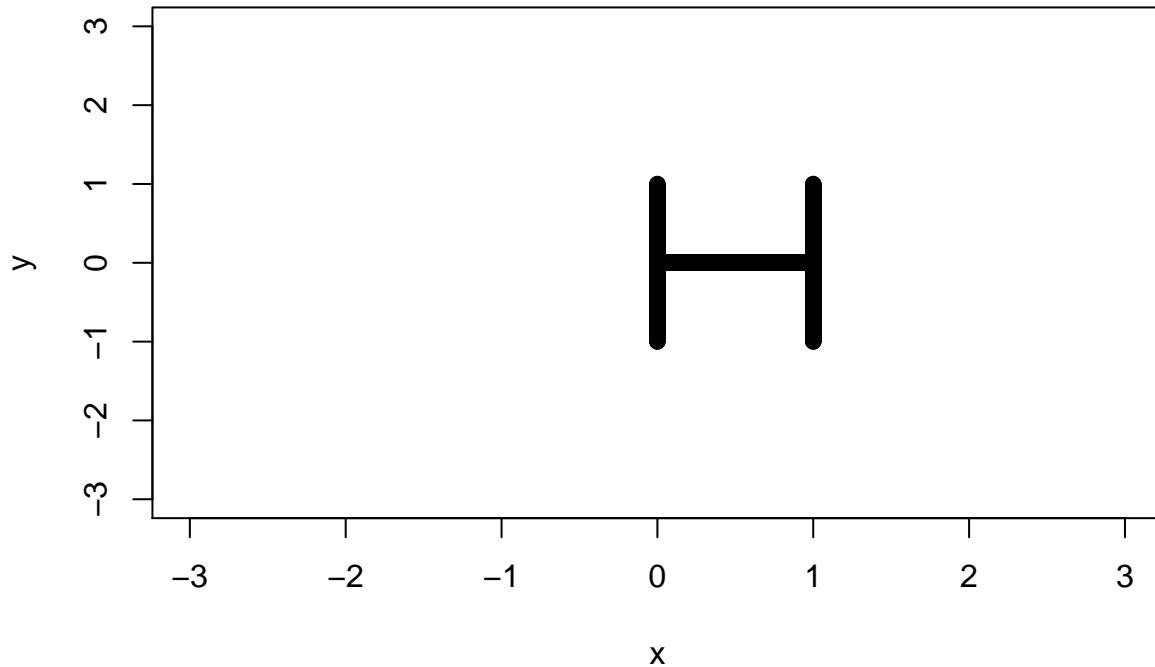
For this assignment, build the first letters for both your first and last name using point plots in R. For example, the following code builds an H.

```
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)

plot(y~x, xlim=c(-3,3), ylim=c(-3,3))
```



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.

Hint: Use `x11()` to open a new plotting window in R.

Upload your document as a .RMD file. I will know if your assignment is correct if the animation runs. correctly

## Solution

In the interest of attempting to create something useful, two methods of manipulation were used to perform the transformations. The first two, shear and scaling, were performed to have more control over the point vectors in the transformation process. The second two, rotation and projection, utilized a function for matrix multiplication that although it still runs in loop, may prove more efficient. Both achieve the desired results. The code to run the full animation is shown.

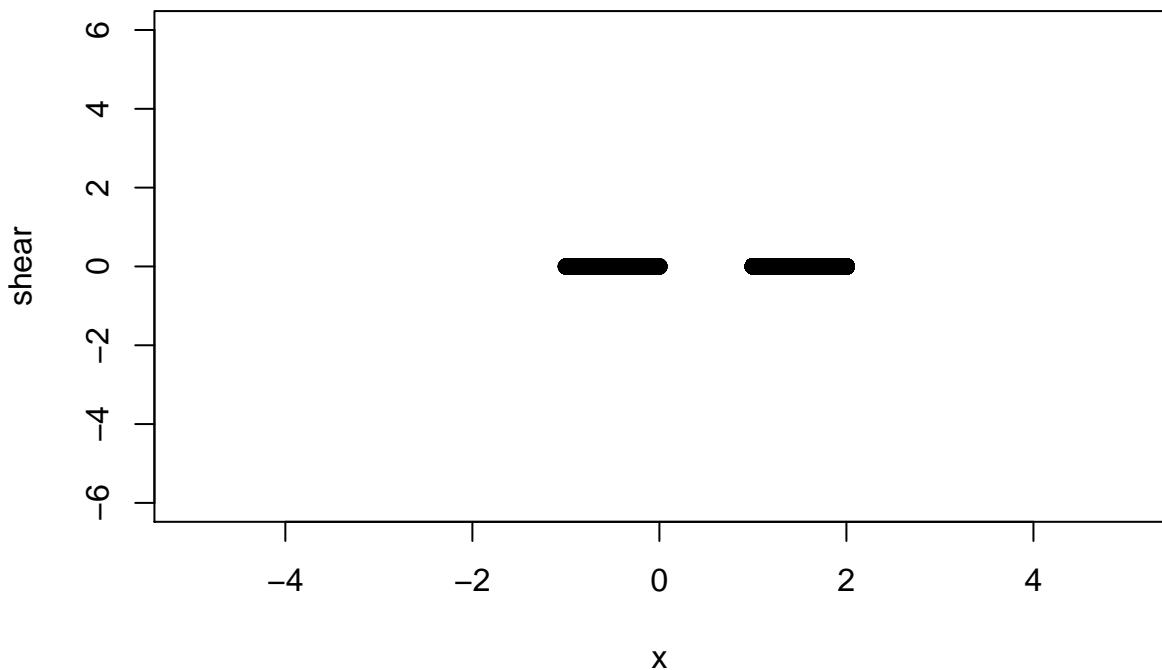
```
x11()
x <- c(seq(-1,0,length.out =500), seq(-1,0,length.out =500), seq(-1,0,length.out =500), # Shows the letter P
       rep(2,500), rep(1,500), seq(1,2,length.out = 500), seq(1,2,length.out = 500),
       rep(1, 500) # Shows the letter P
      )
y <- c(rep(-1,500),seq(-1,1,length.out=500), rep(1,500), # Shows the letter Z
       seq(0,1,length.out = 500),seq(0,1, length.out = 500), rep(1,500),rep(0,500),
       seq(-1,1,length.out = 500) # Shows the letter P
```

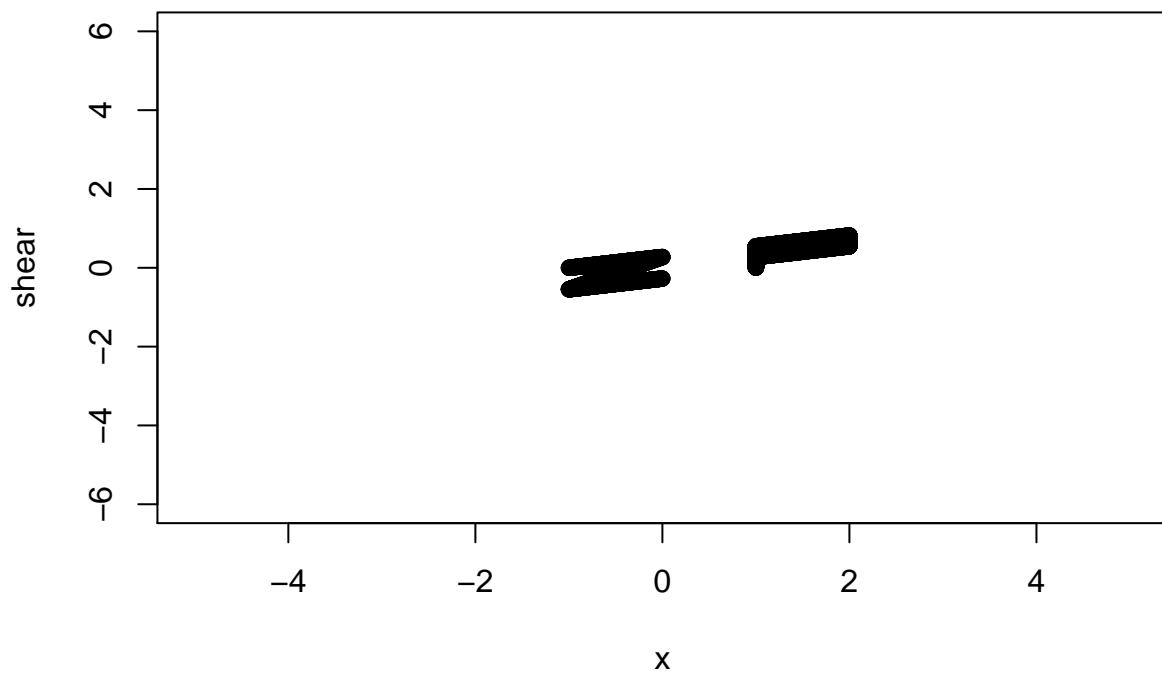
```

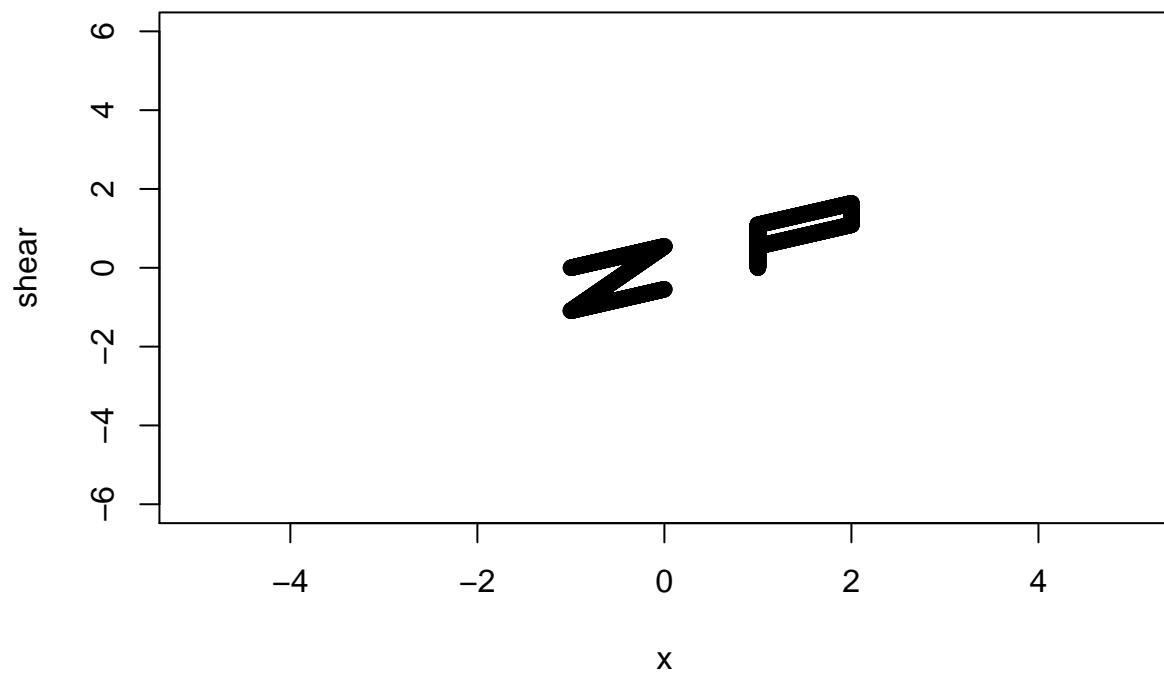
        )
z <- rbind(x,y)
xA <- function(x,y){
  x %*% y
}

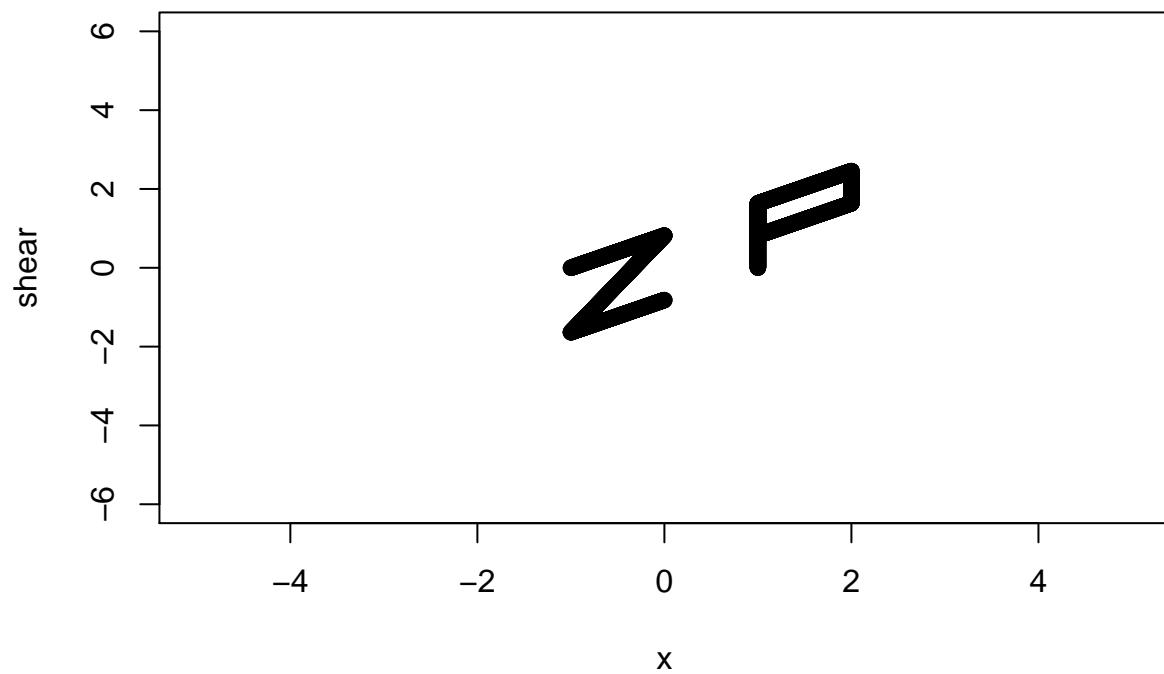
# Shear
for (i in seq(0,3, length.out = 12)) {
  id_m <- 1L * i
  shear <- (x * i) + (y * id_m)
  plot(shear~x,
        xlim=c(-5,5), ylim=c(-6,6))
}

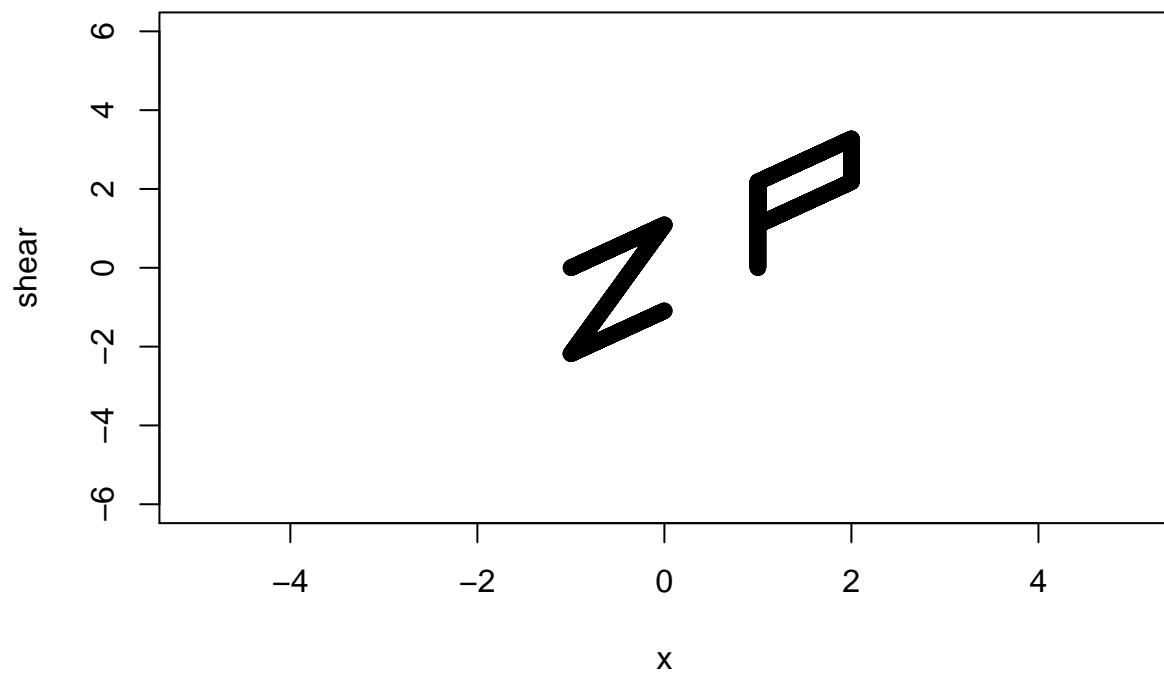
```

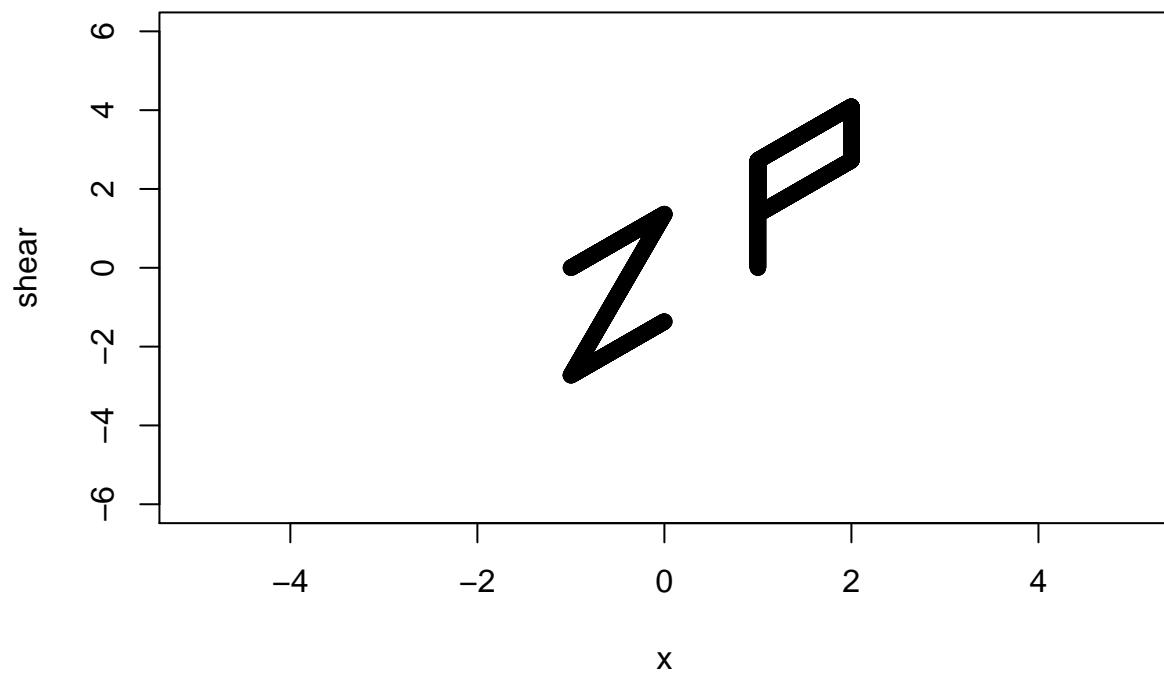


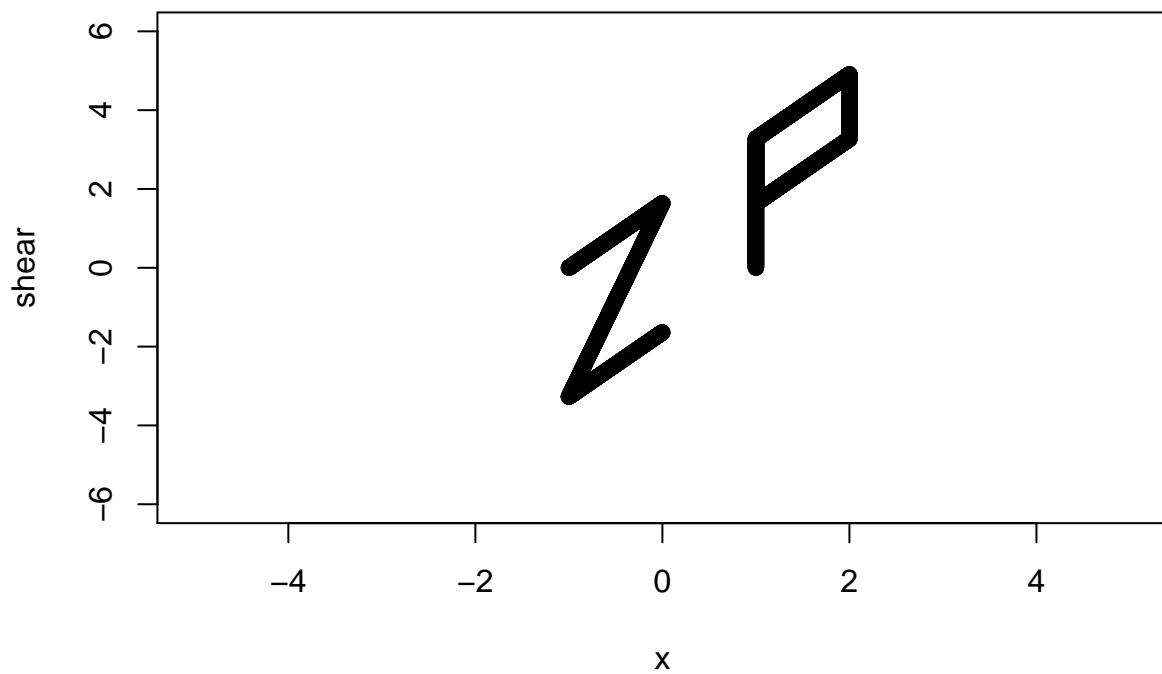


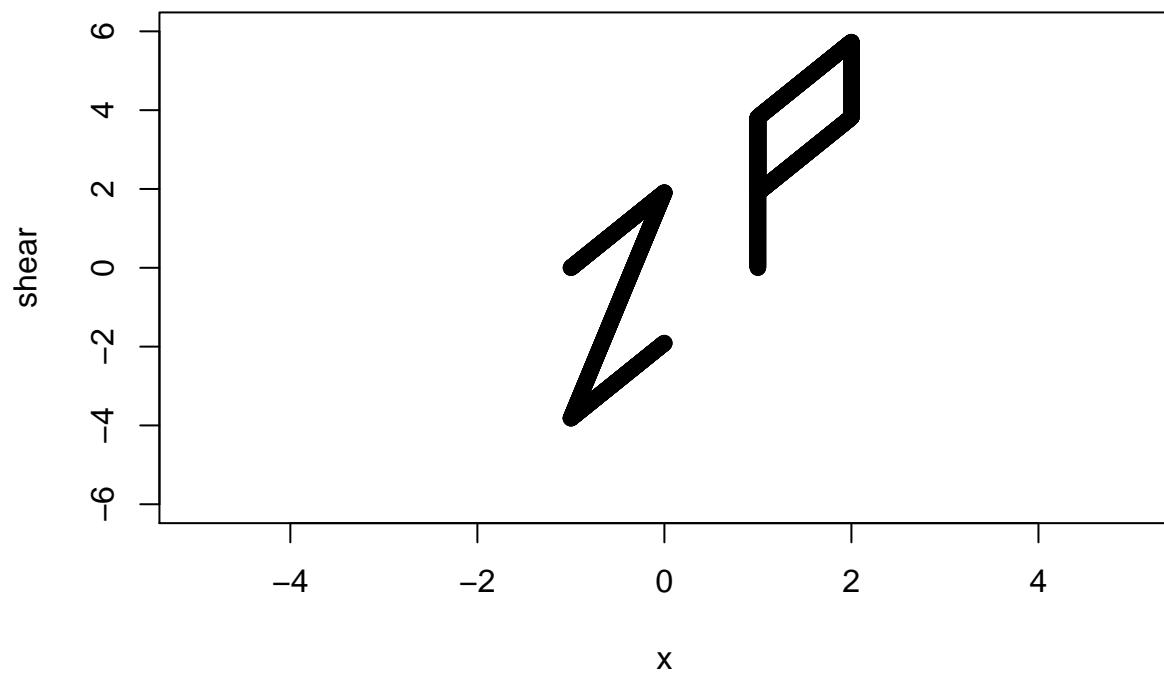


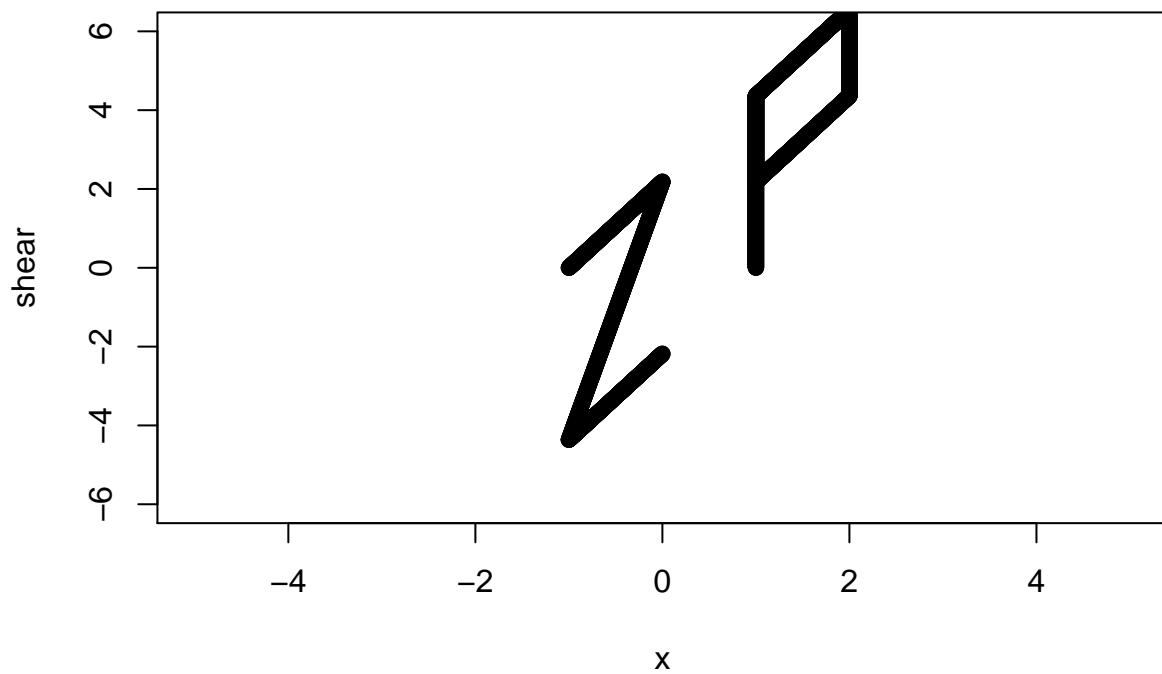


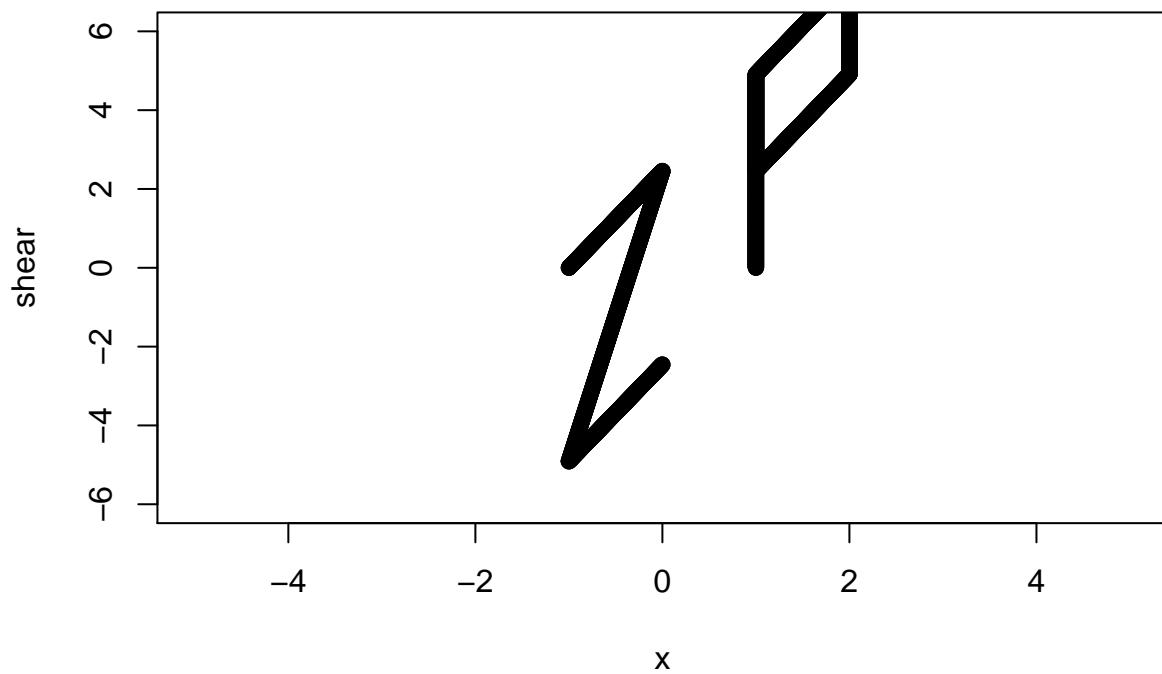


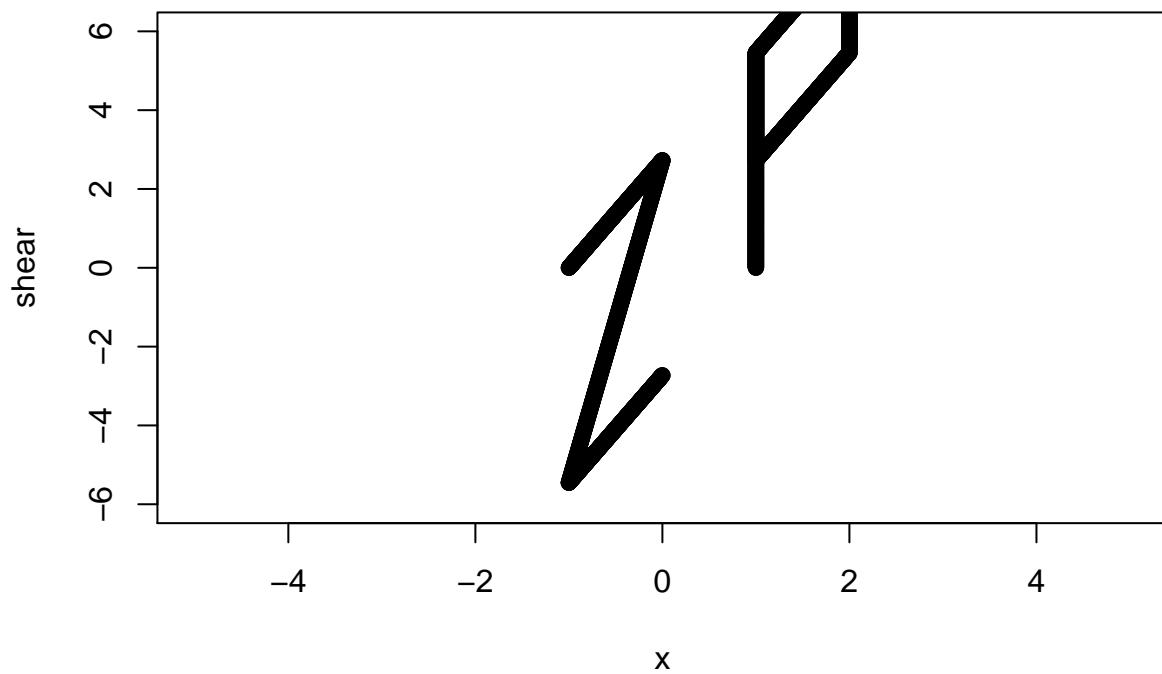


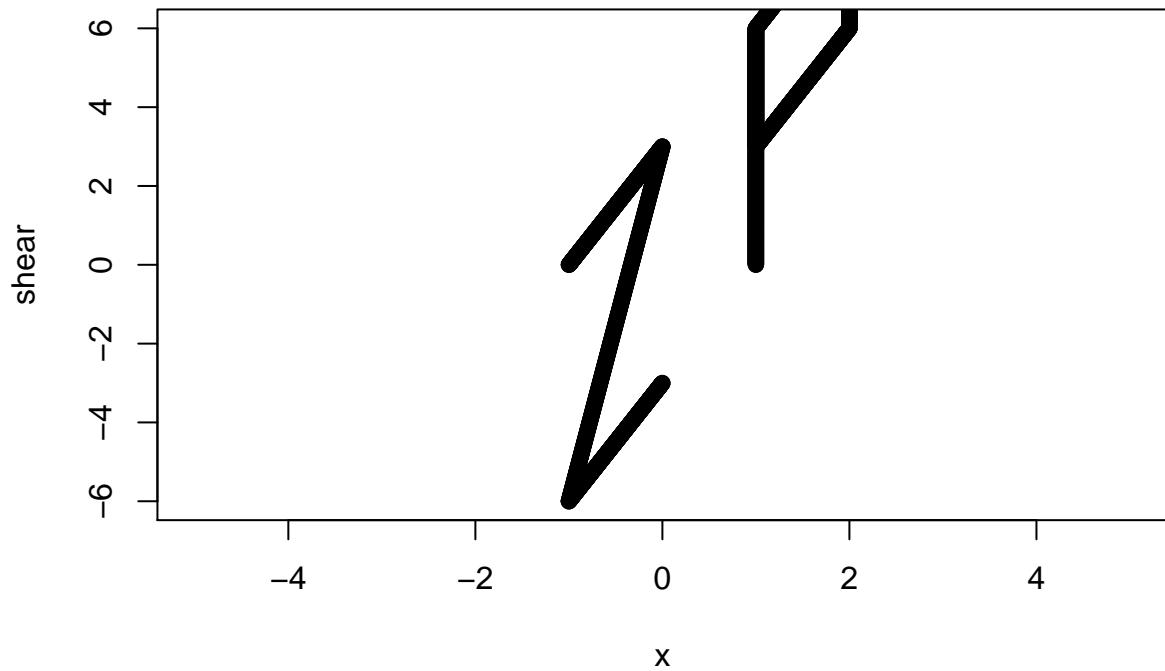




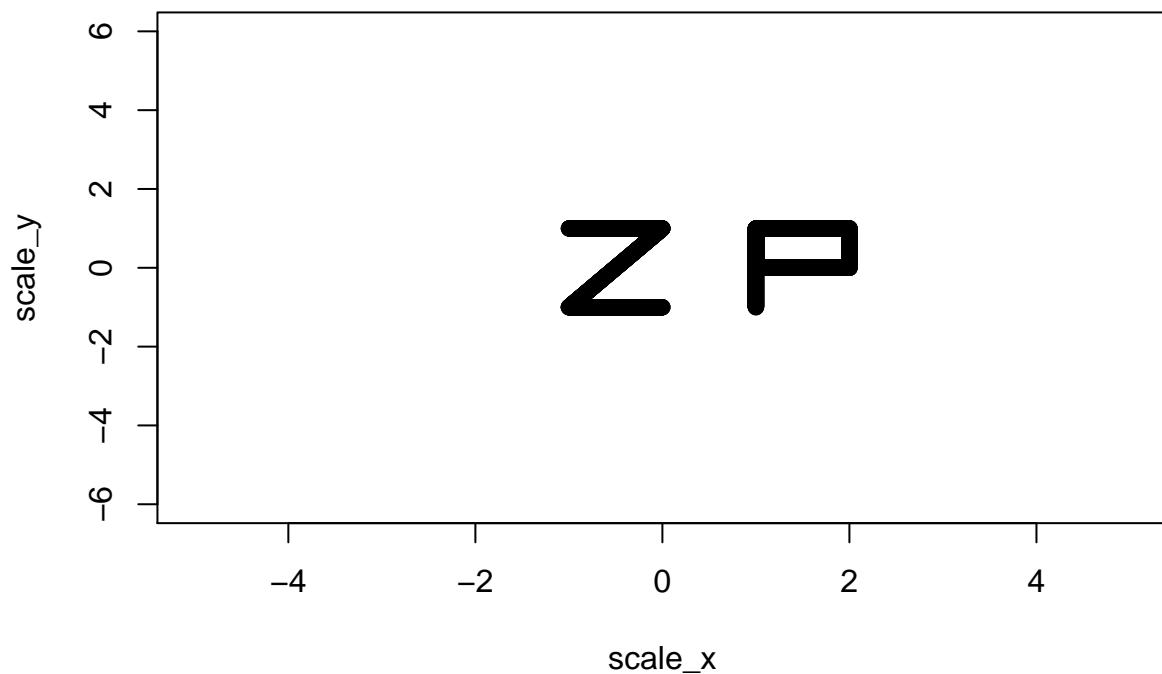


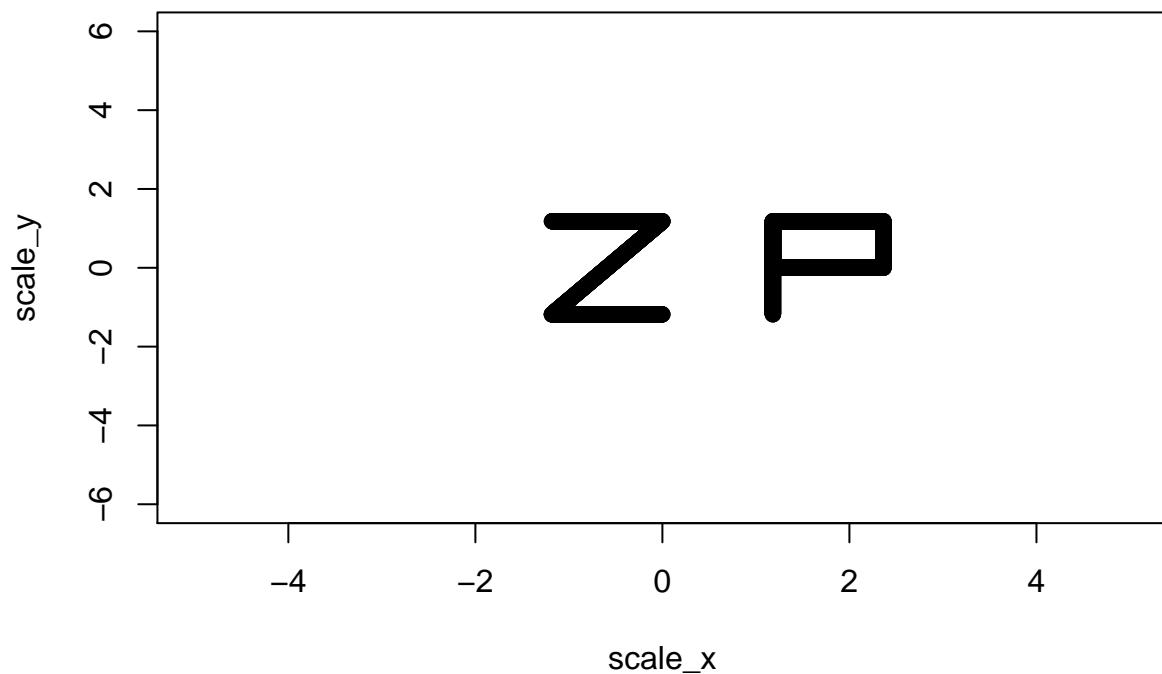


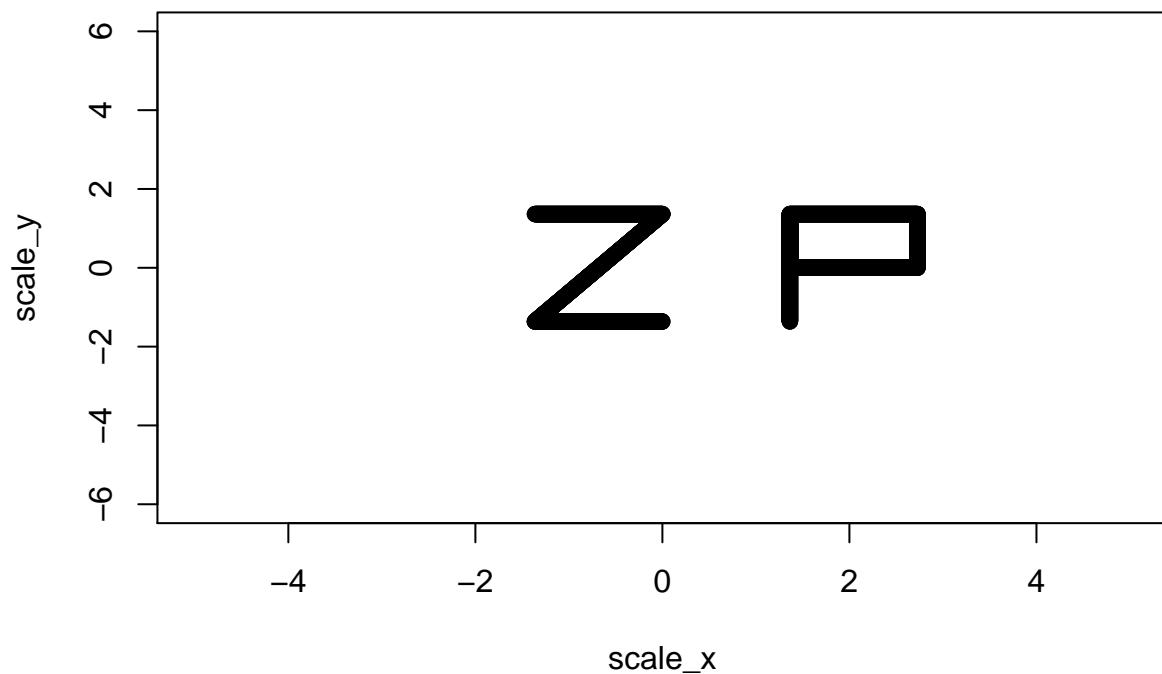


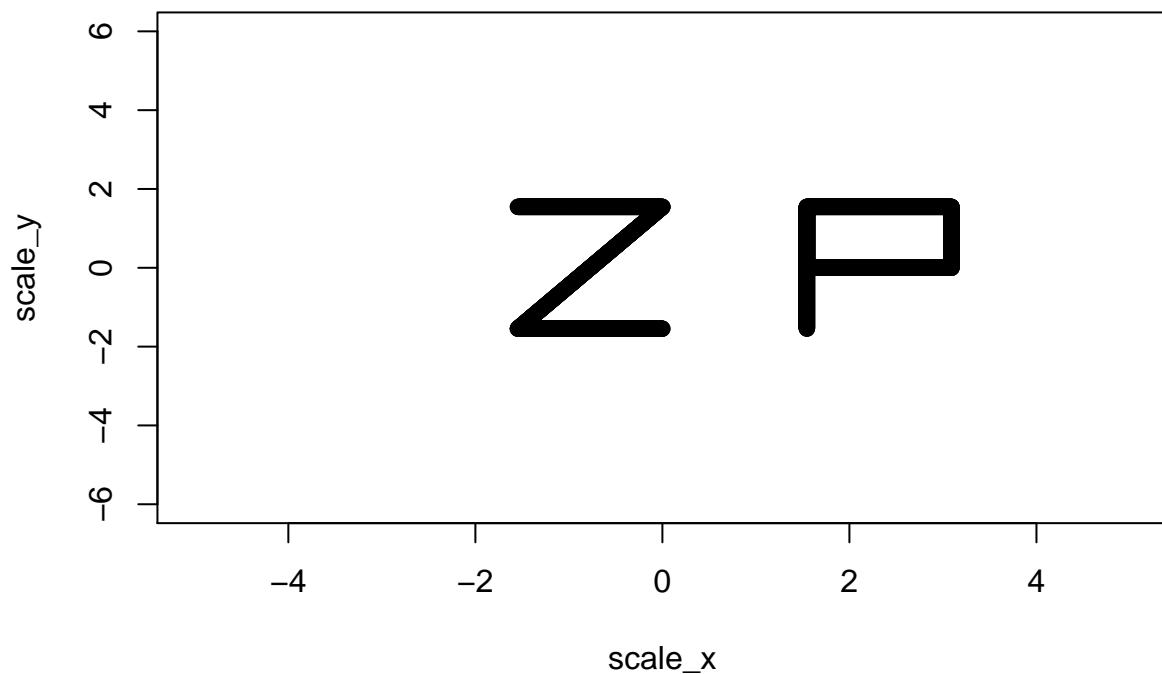


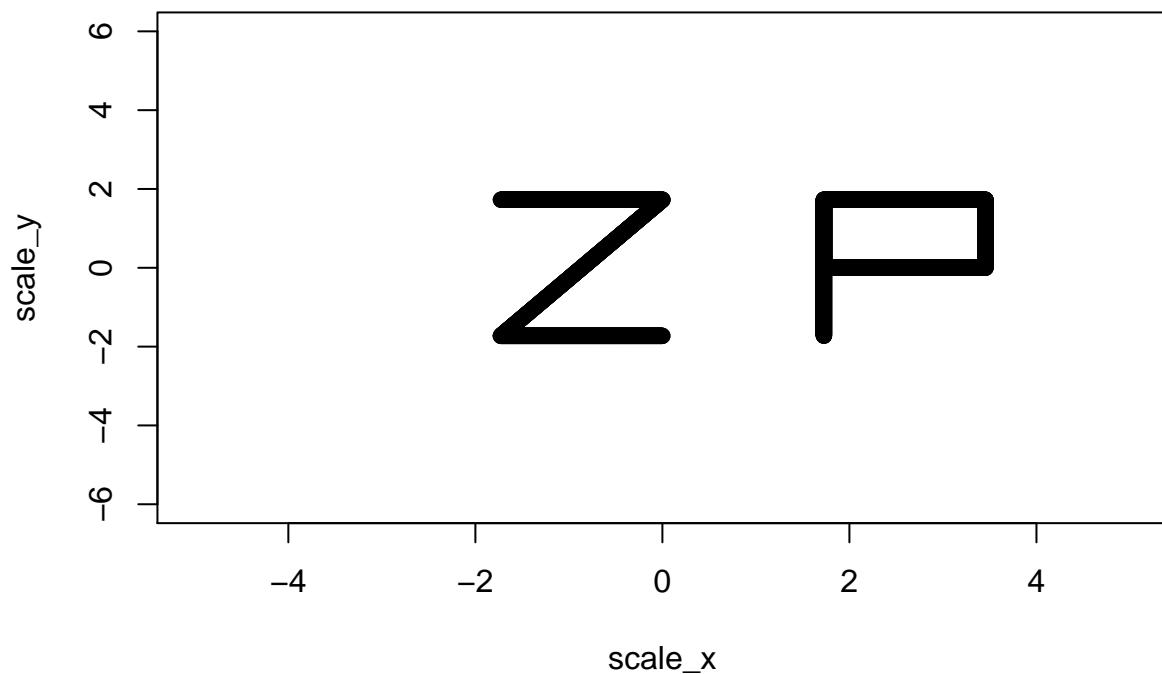
```
# Scale
for (i in seq(1,3, length.out = 12)){
  id_m <- 1L * i
  scale_y <- y * id_m
  scale_x <- x * id_m
  plot(scale_y~scale_x,
       xlim=c(-5,5), ylim=c(-6,6))
}
```

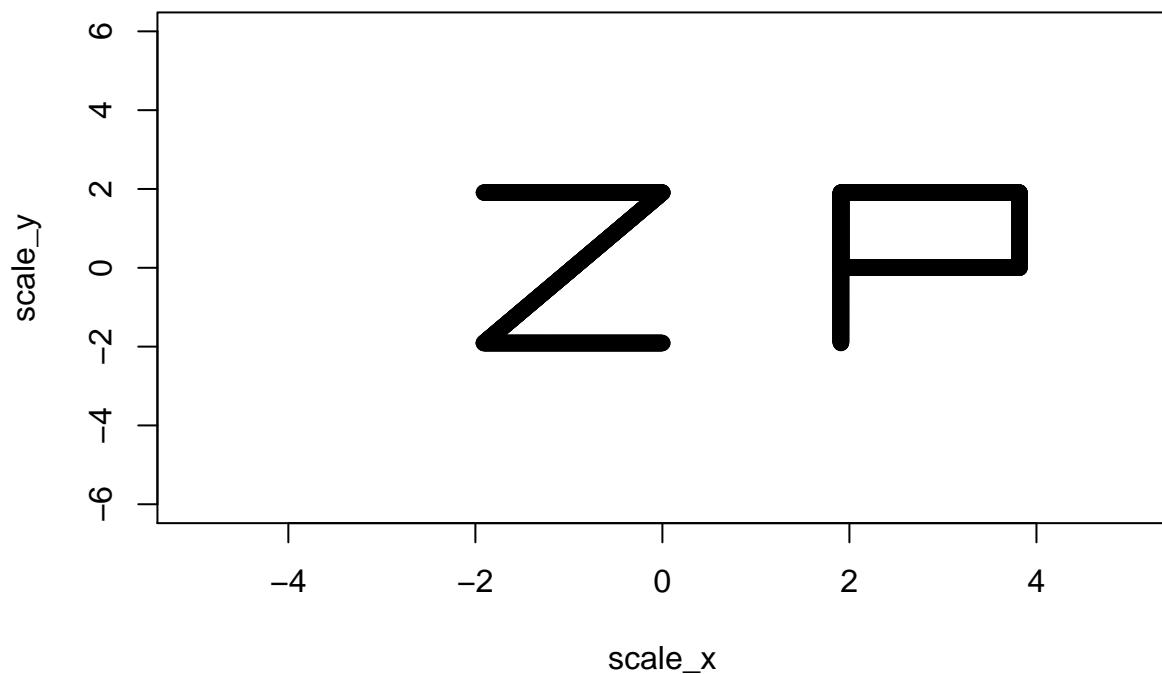


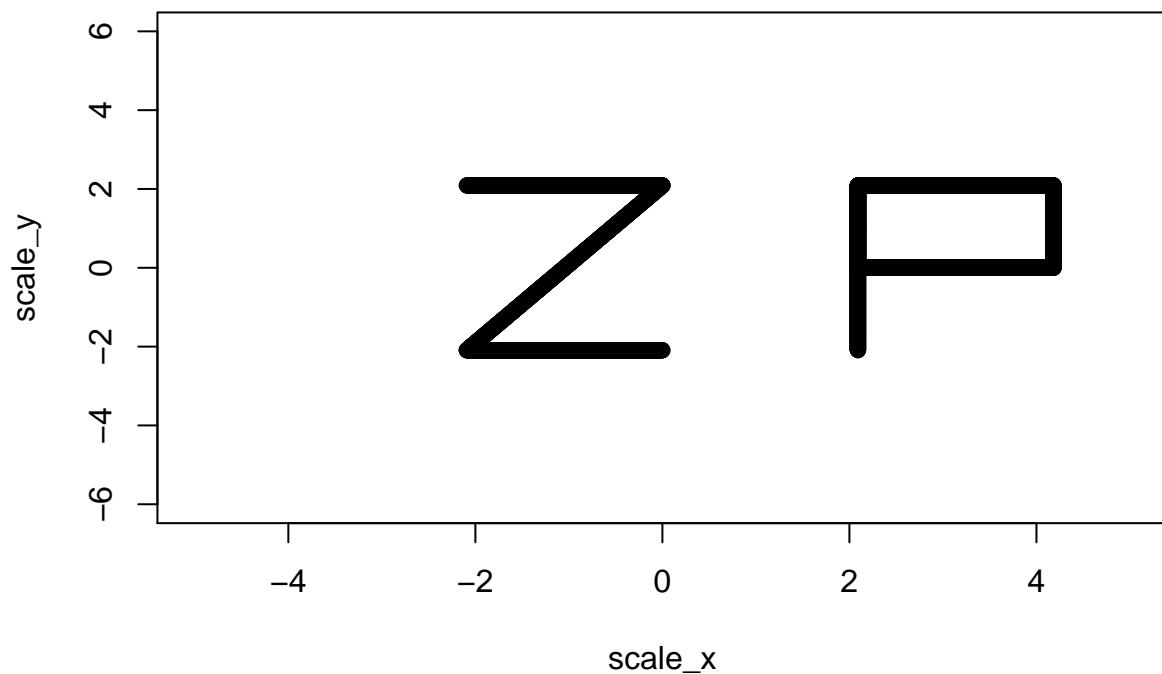


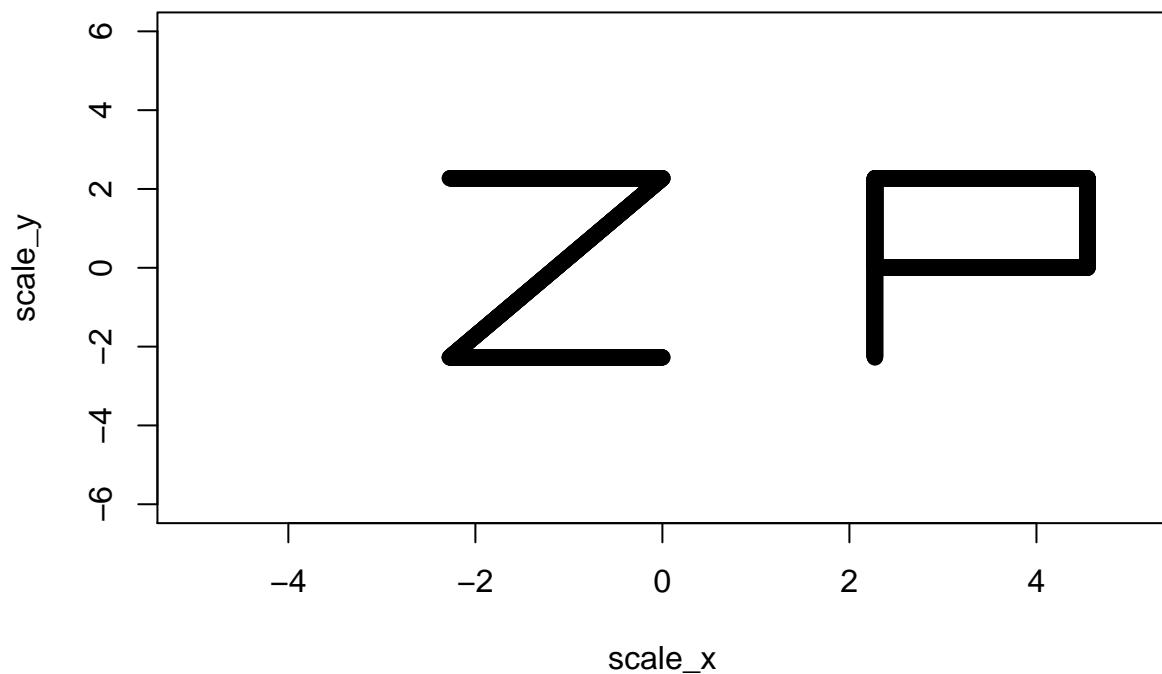


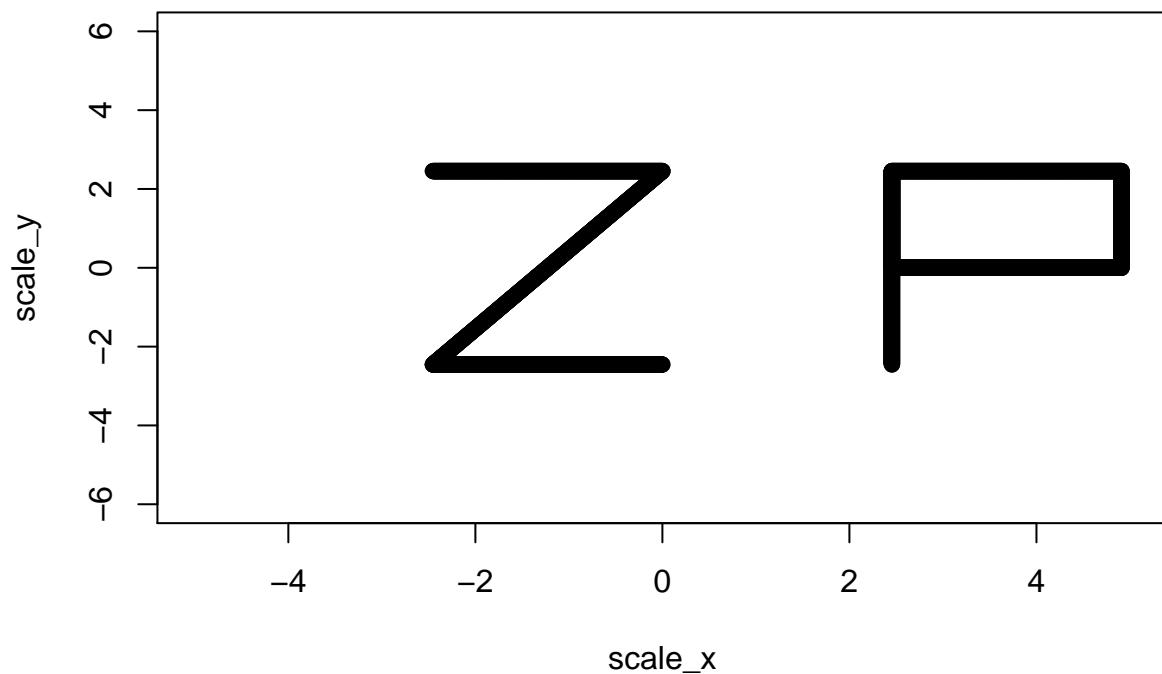


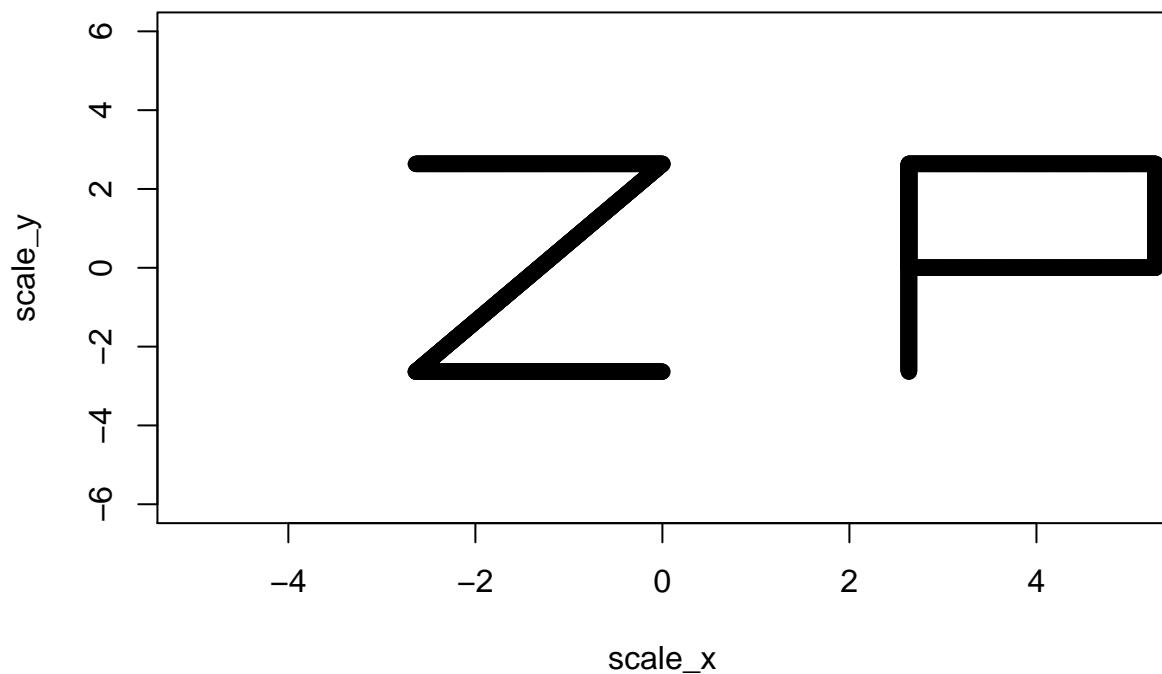


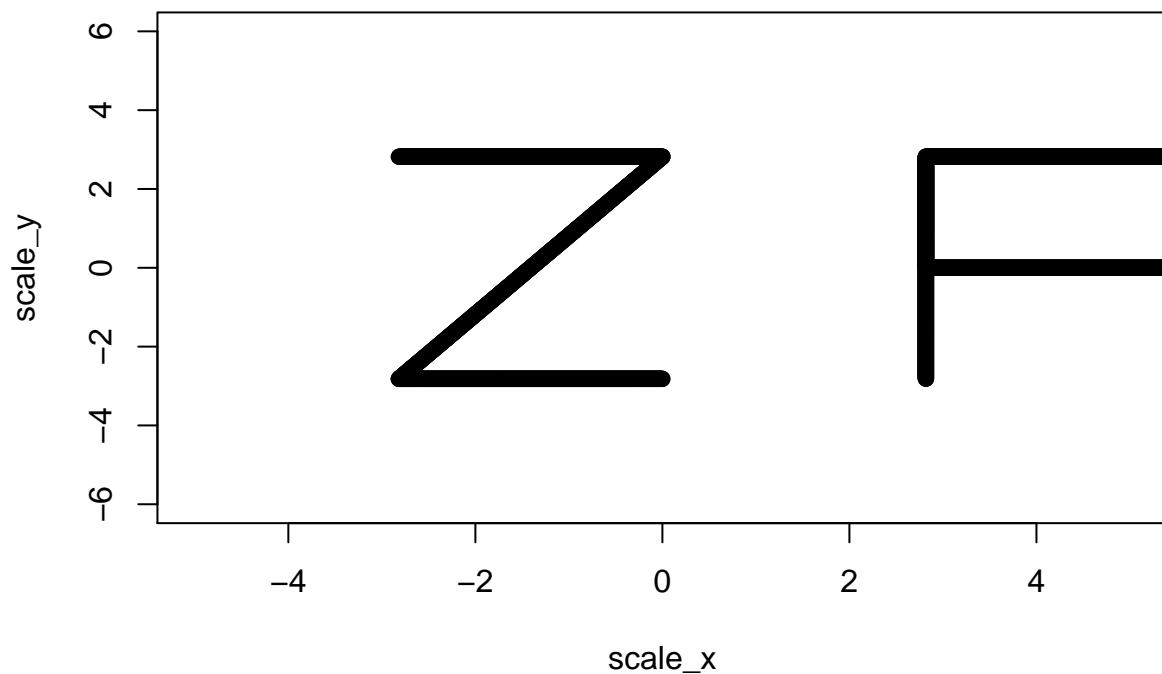


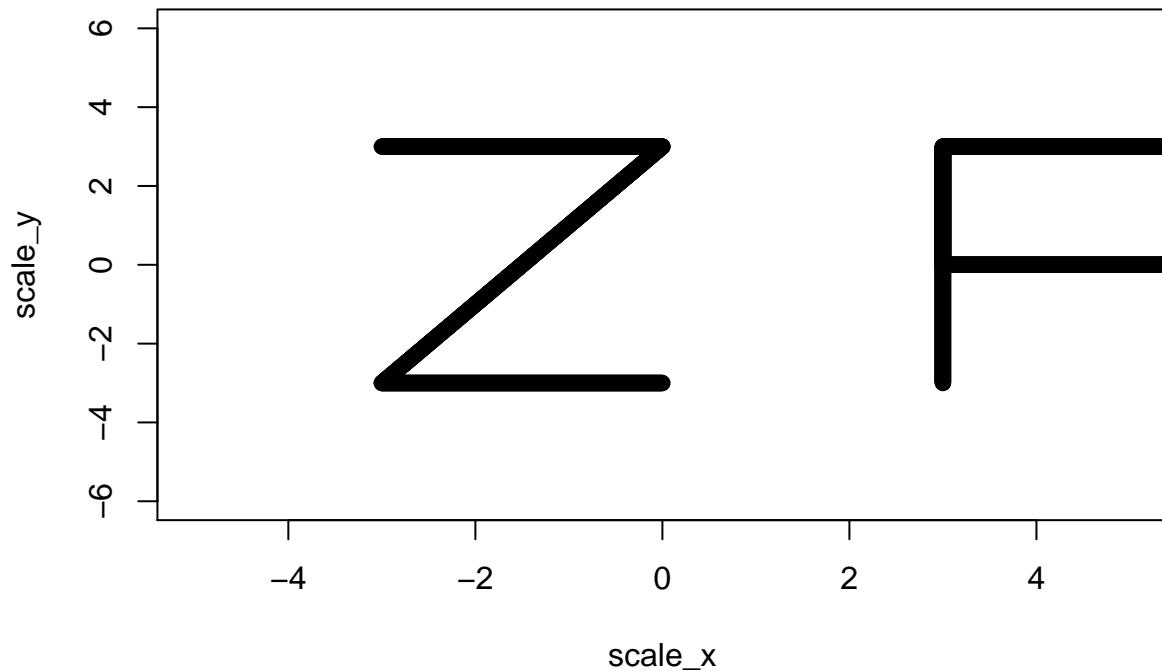




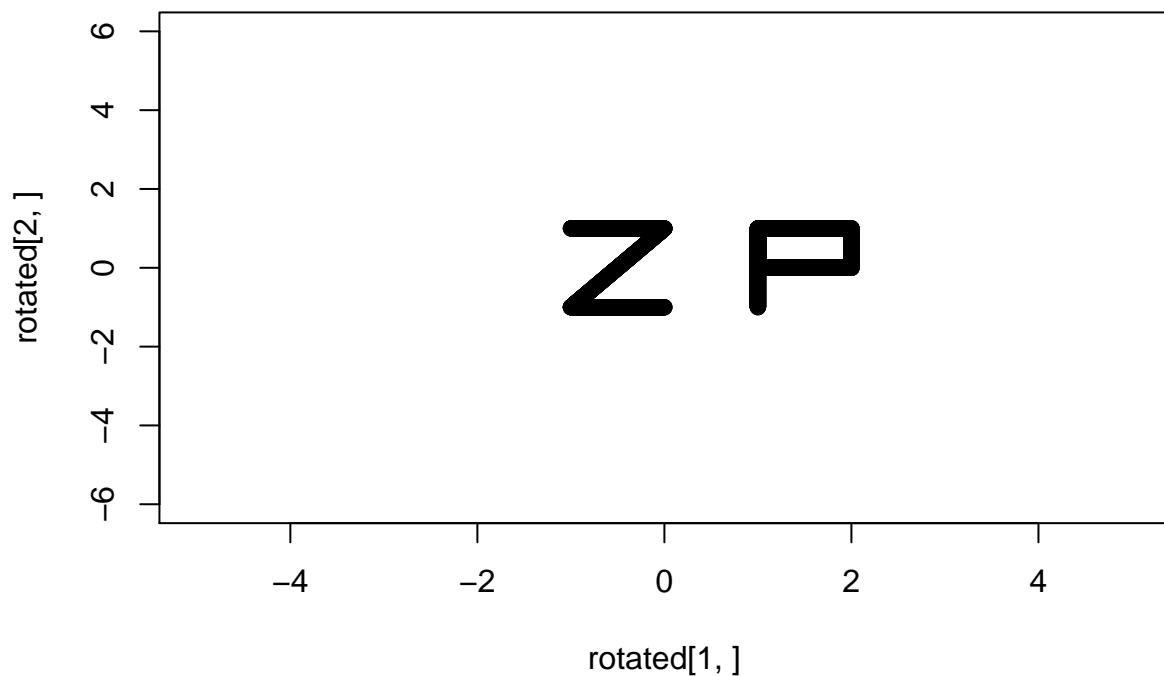


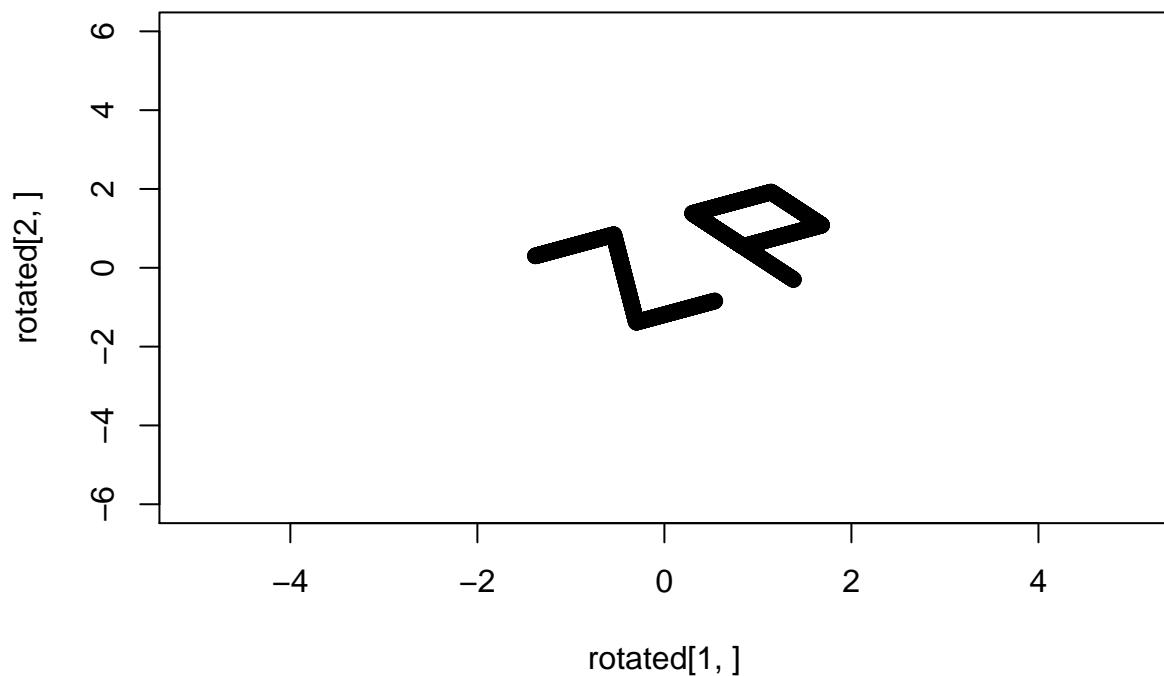


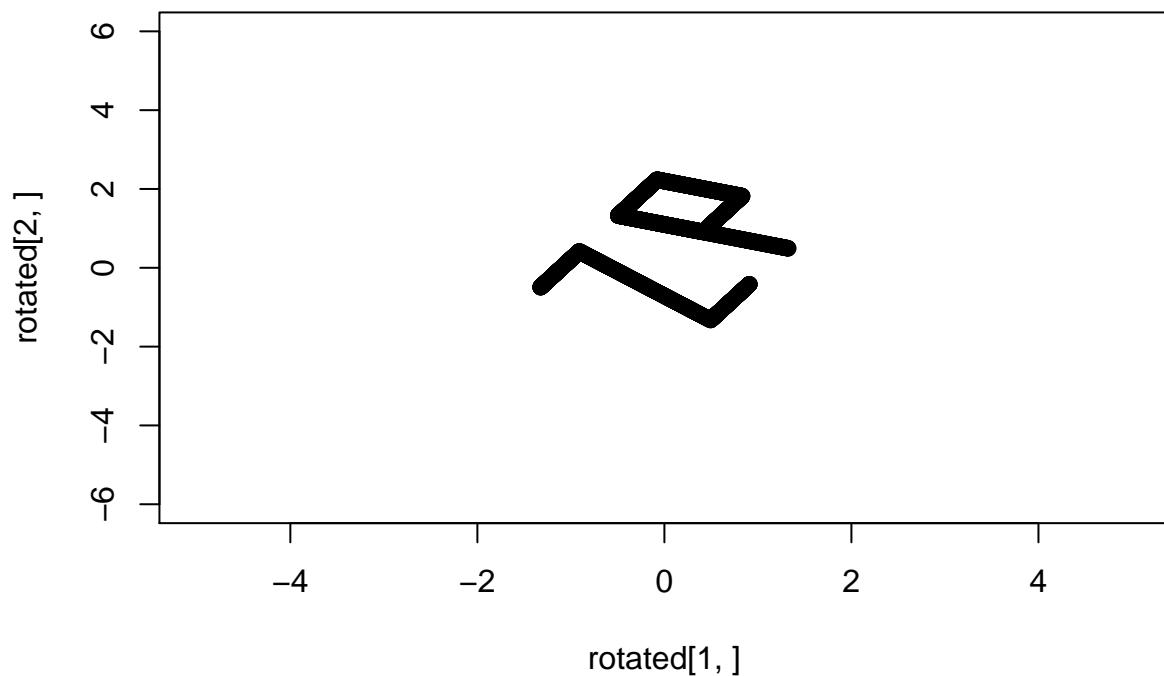


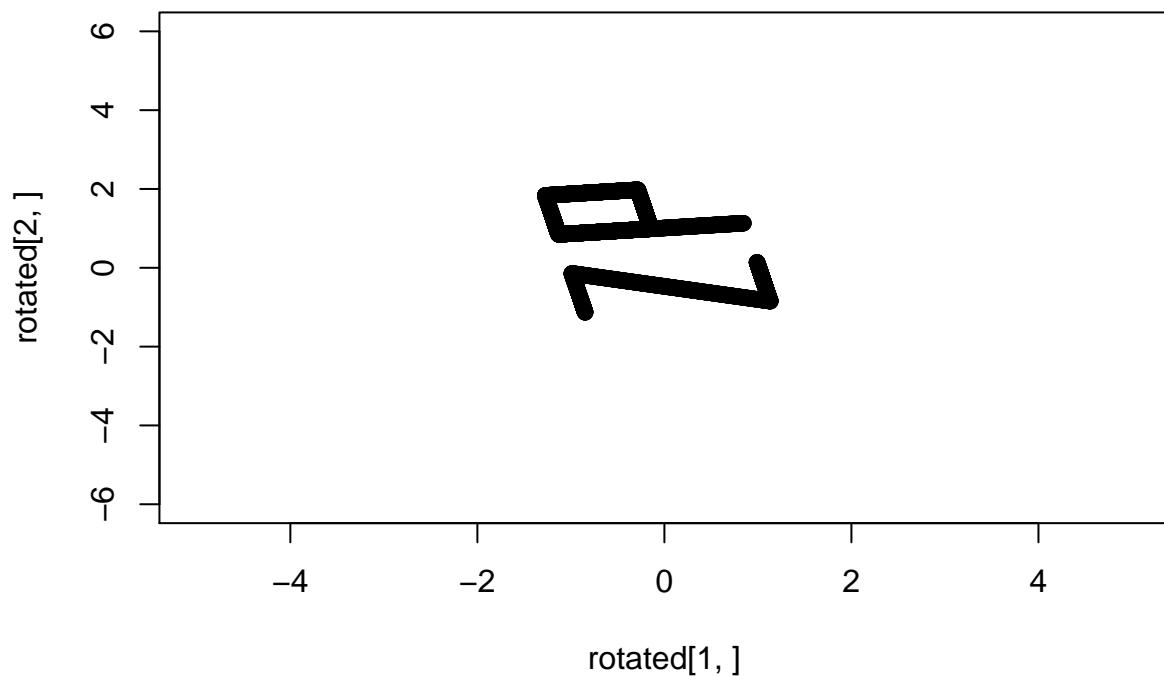


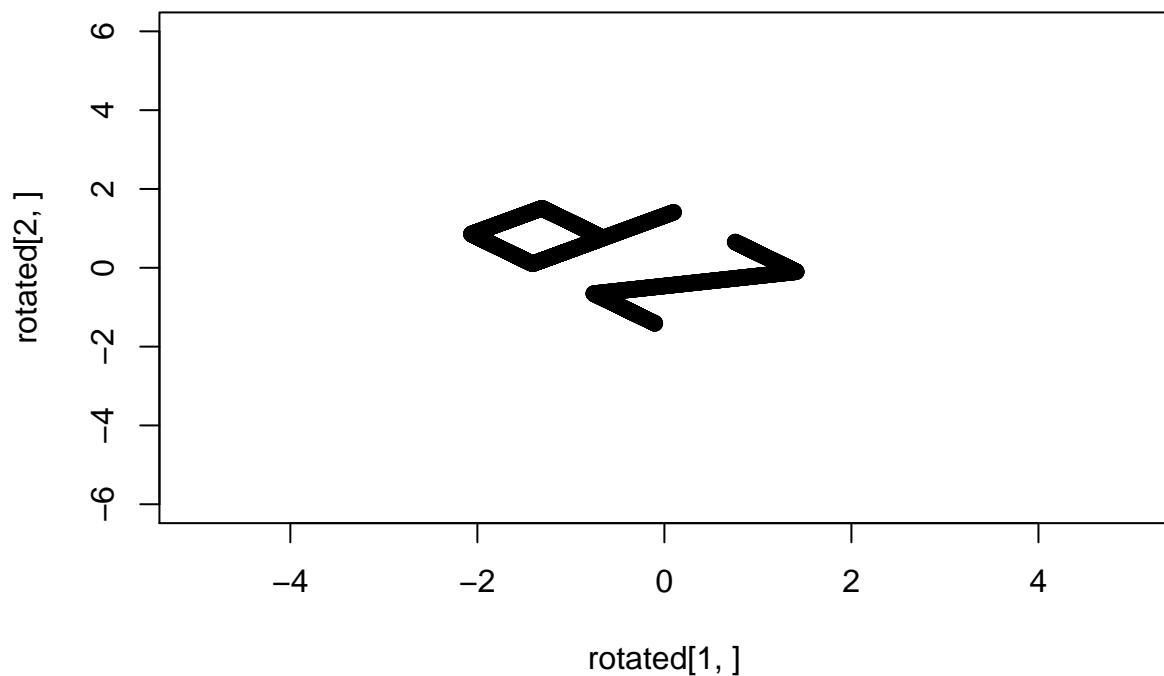
```
# Rotate
for (i in seq(0, 2*pi, length.out = 12)) {
  rotated <- apply(z, 2, function(x)
    xA(x, matrix(c(
      cos(i), -sin(i), sin(i), cos(i)), ncol = 2, nrow = 2)))
  plot(rotated[2,]~rotated[1,],
       xlim=c(-5,5), ylim=c(-6,6))
}
```

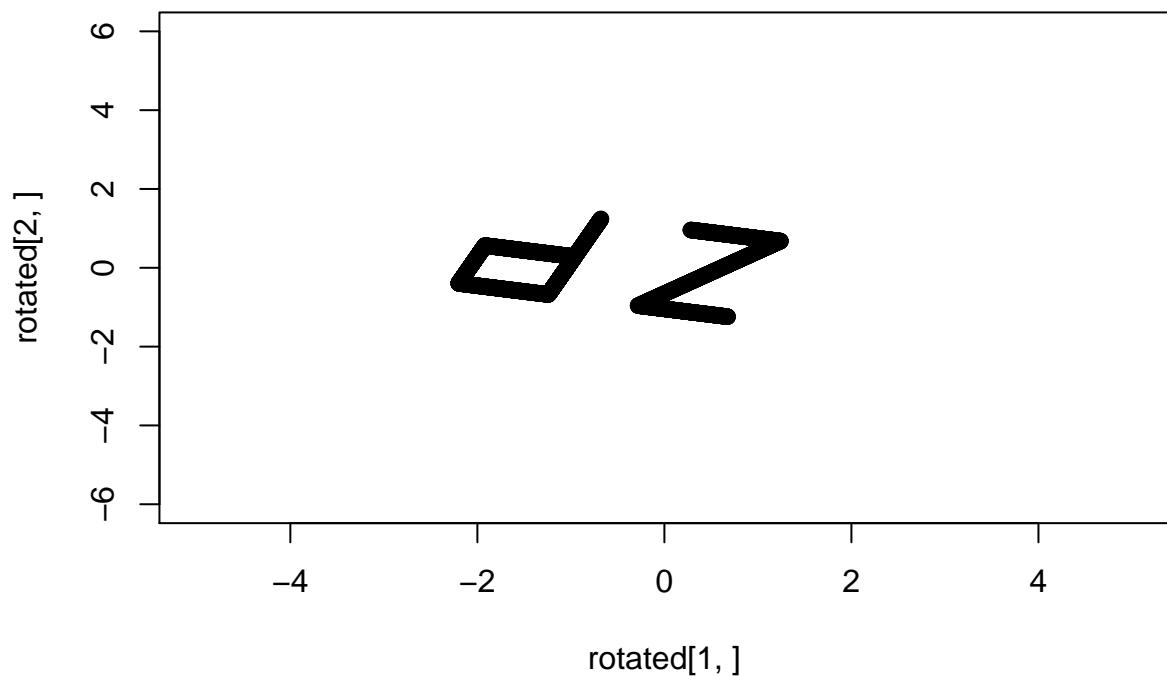


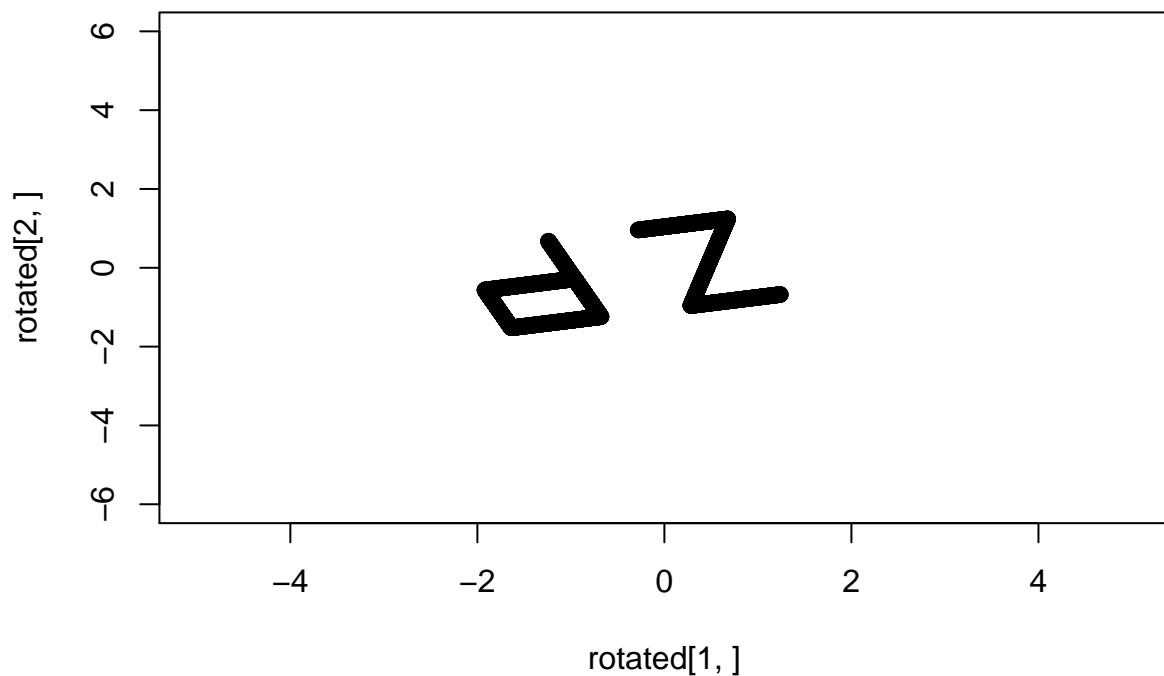


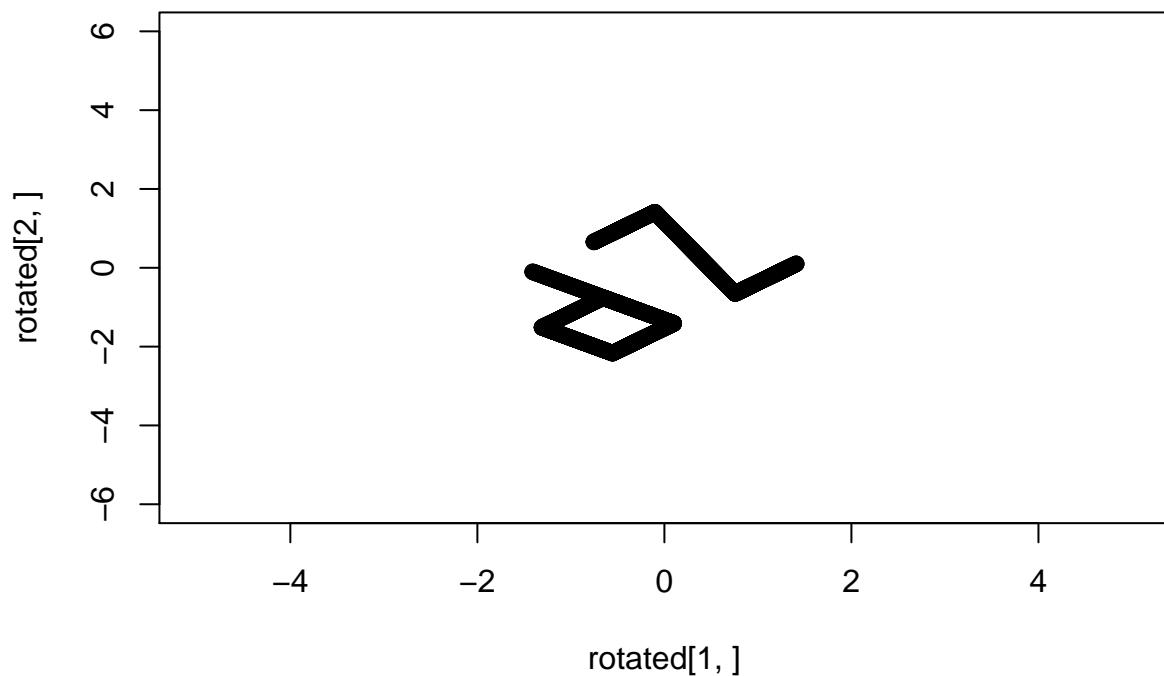


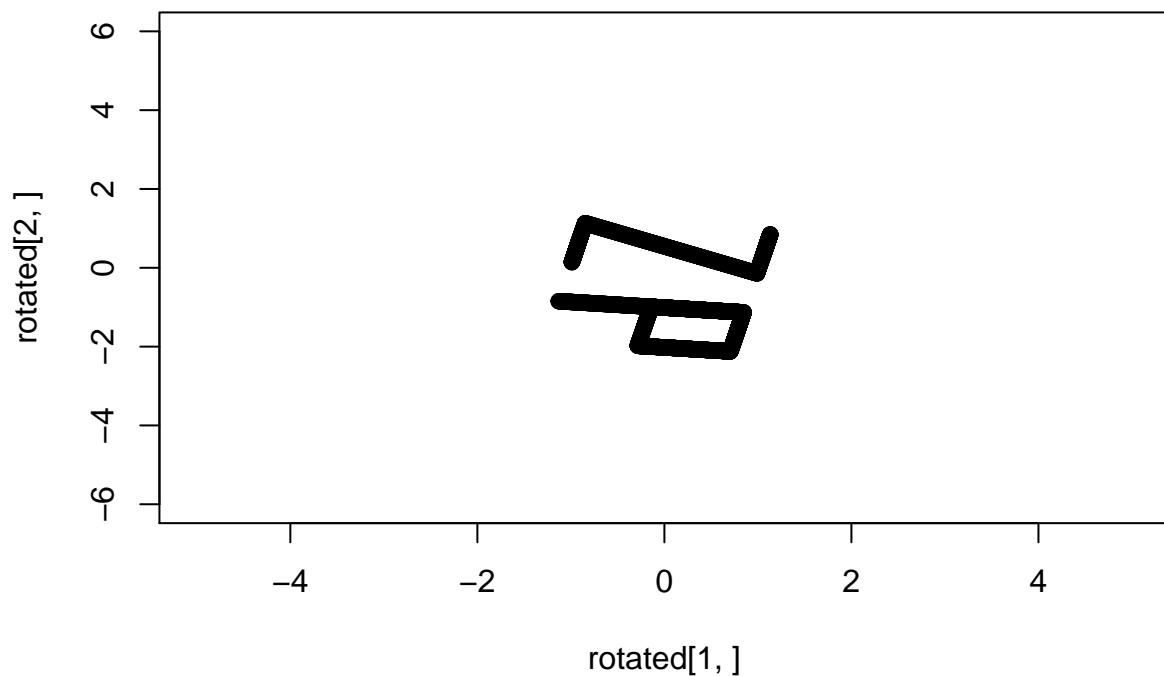


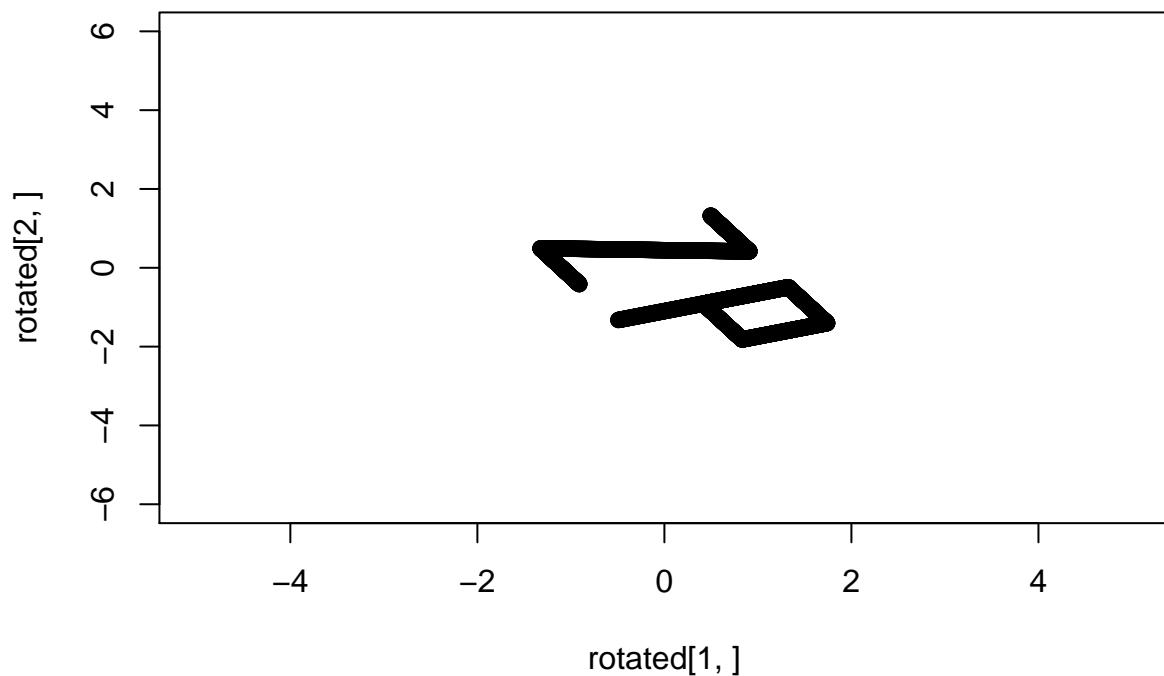


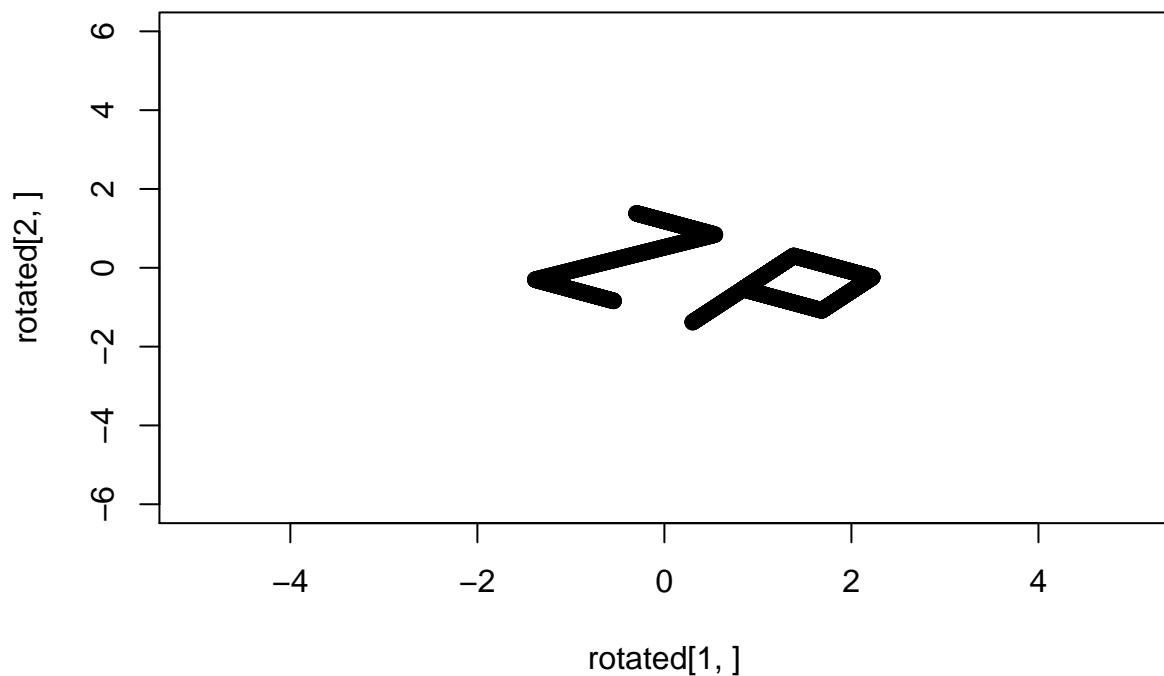


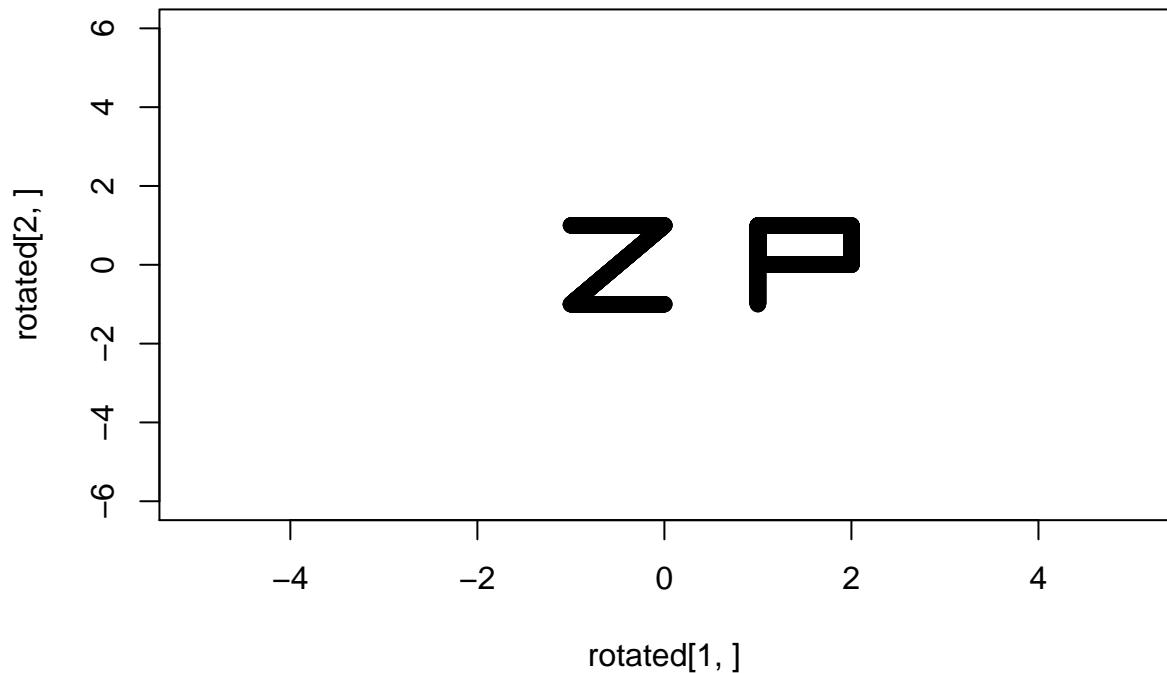












```
# Project
for (i in seq(0, 2*pi, length.out = 12)) {
  proj <- rbind(z, rep(0, ncol(z)))
  projected <- apply(proj, 2, function(x)
    xA(x, matrix(c(1, 0, 0, 0,
                  cos(i), -sin(i), 0, sin(i), cos(i)), ncol = 3, nrow = 3)))
  plot(projected[2,]~projected[1,],
       xlim=c(-5,5), ylim=c(-6,6))
}
```

