

# DATA605\_w1\_Image Manipulation

David Simbandumwe

## Assignment Overview

(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.

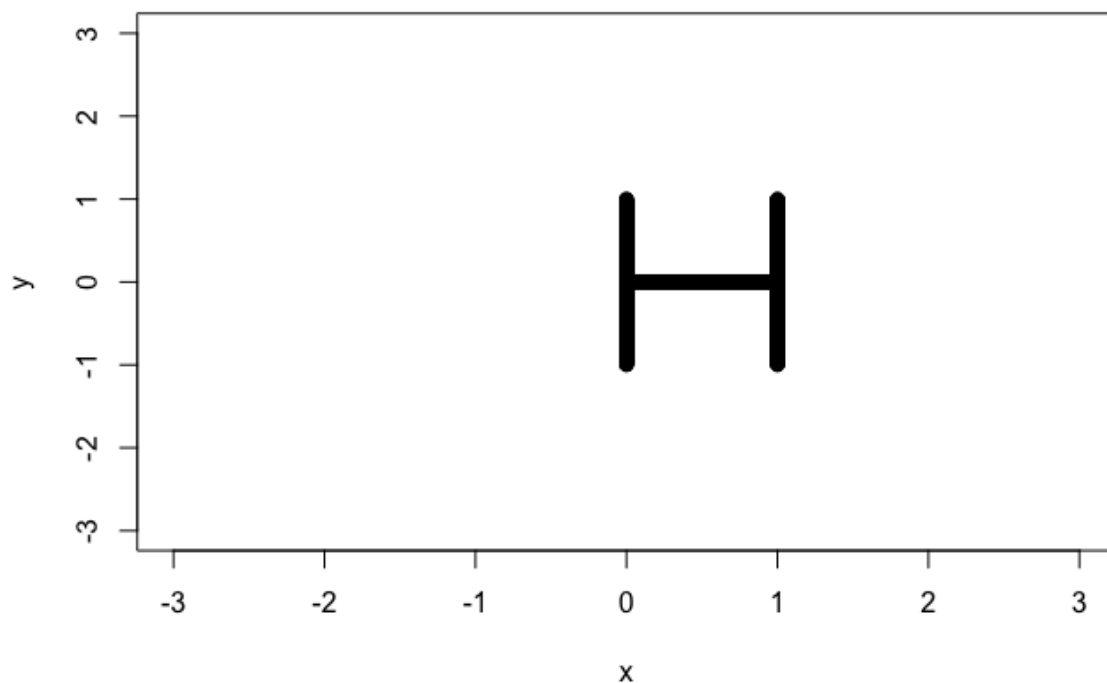
## Questions 1

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))
```



```
dev.control('enable')  
myani=ani.record(reset = TRUE, replay.cur = FALSE)
```

## Answer - Q1

```
x <- c(  
  seq(0,0.5, length.out = 500)  
  ,rep(0, 500)  
  ,seq(0,0.5, length.out = 500)  
  ,rep(0.5, 500)  
  ,seq(0,0.5, length.out = 500)  
  
  ,seq(-0.7,-1, length.out = 500)  
  ,rep(-1, 500)  
  ,seq(-0.7,-1, length.out = 500)  
  ,rep(-0.4, 500)  
  
  ,seq(-0.70, -0.4, length.out = 500)  
  ,seq(-0.70, -0.4, length.out = 500)  
)
```

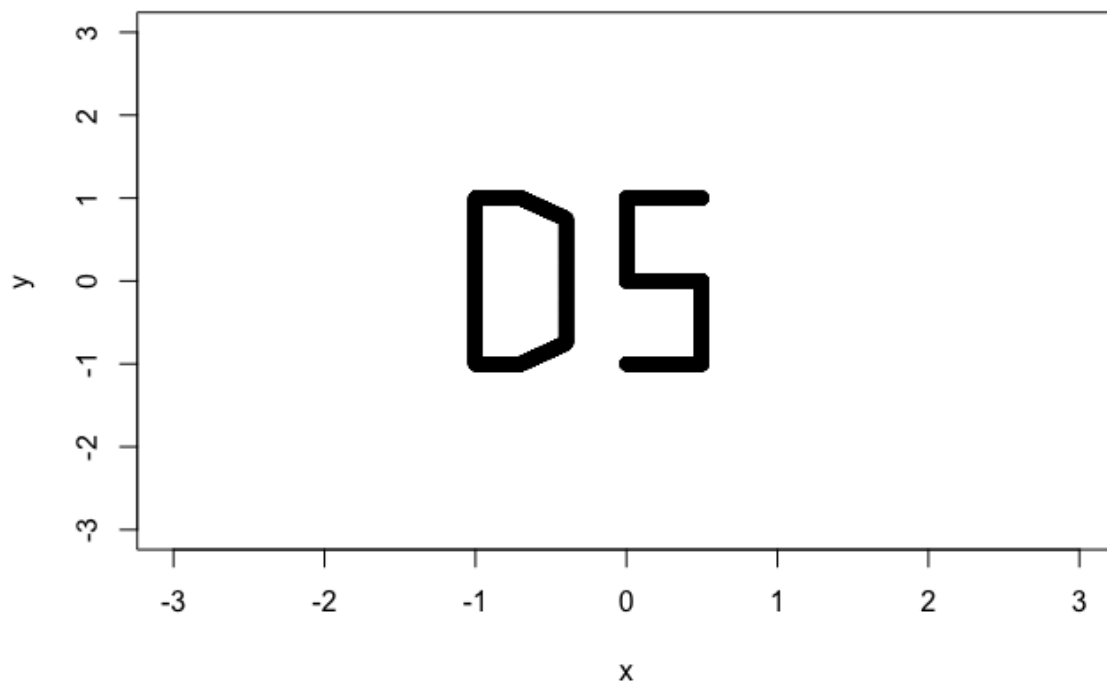
```
y <- c(
  rep(1, 500)
  ,seq(1,0, length.out = 500)
  ,rep(0, 500)
  ,seq(0,-1,length.out = 500)
  ,rep(-1,500)

  ,rep(1, 500)
  ,seq(-1,1, length.out = 500)
  ,rep(-1, 500)
  ,seq(-0.75,0.75, length.out = 500)

  ,seq(1,0.75, length.out = 500)
  ,seq(-1,-0.75, length.out = 500)
)

z <- rbind(x,y)
z <- data.frame(z)

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



## Questions 2

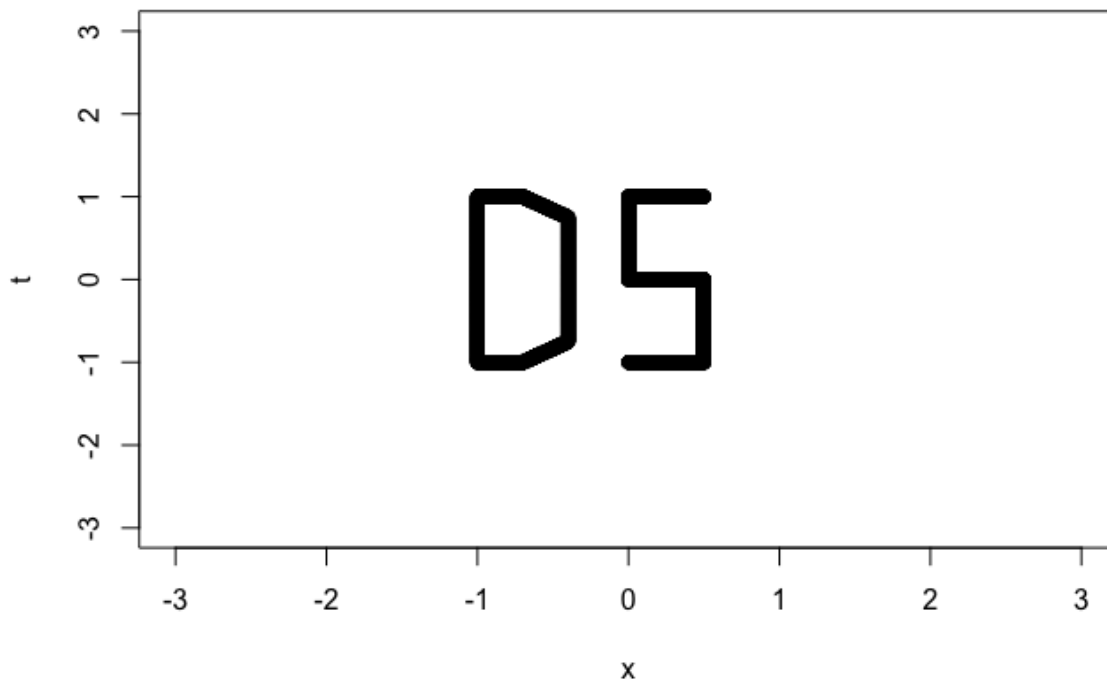
Then, write R code that will left multiply ( $\%>\%$ ) a square matrix ( $x$ ) against each of the vectors of points ( $y$ ). Initially, that square matrix will be the Identity matrix.

## Answer - Q2

```
a <- diag(length(y))
t <- array(a %*% y)

z <- rbind(x,t)
z <- data.frame(z)

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



## Questions 3

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.

## Answer - Q3

```
a <- diag(2)
```

```
a
```

```
##      [,1] [,2]
```

```
## [1,]    1    0
```

```
## [2,]    0    1
```

## Shear

```
t1 <- seq(-3,3,length.out=100)
```

```
print(t1)
```

```
## [1] -3.00000000 -2.93939394 -2.87878788 -2.81818182 -2.75757576 -2.69696970
```

```
## [7] -2.63636364 -2.57575758 -2.51515152 -2.45454545 -2.39393939 -2.33333333
```

```
## [13] -2.27272727 -2.21212121 -2.15151515 -2.09090909 -2.03030303 -1.96969697
```

```
## [19] -1.90909091 -1.84848485 -1.78787879 -1.72727273 -1.66666667 -1.60606061
```

```
## [25] -1.54545455 -1.48484848 -1.42424242 -1.36363636 -1.30303030 -1.24242424
```

```
## [31] -1.18181818 -1.12121212 -1.06060606 -1.00000000 -0.93939394 -0.87878788
```

```
## [37] -0.81818182 -0.75757576 -0.69696970 -0.63636364 -0.57575758 -0.51515152
```

```
## [43] -0.45454545 -0.39393939 -0.33333333 -0.27272727 -0.21212121 -0.15151515
```

```
## [49] -0.09090909 -0.03030303 0.03030303 0.09090909 0.15151515 0.21212121
```

```
## [55] 0.27272727 0.33333333 0.39393939 0.45454545 0.51515152 0.57575758
```

```
## [61] 0.63636364 0.69696970 0.75757576 0.81818182 0.87878788 0.93939394
```

```
## [67] 1.00000000 1.06060606 1.12121212 1.18181818 1.24242424 1.30303030
```

```
## [73] 1.36363636 1.42424242 1.48484848 1.54545455 1.60606061 1.66666667
```

```
## [79] 1.72727273 1.78787879 1.84848485 1.90909091 1.96969697 2.03030303
```

```
## [85] 2.09090909 2.15151515 2.21212121 2.27272727 2.33333333 2.39393939
```

```
## [91] 2.45454545 2.51515152 2.57575758 2.63636364 2.69696970 2.75757576
```

```
## [97] 2.81818182 2.87878788 2.93939394 3.00000000
```

```
# x11()
```

```
a <- diag(2)
```

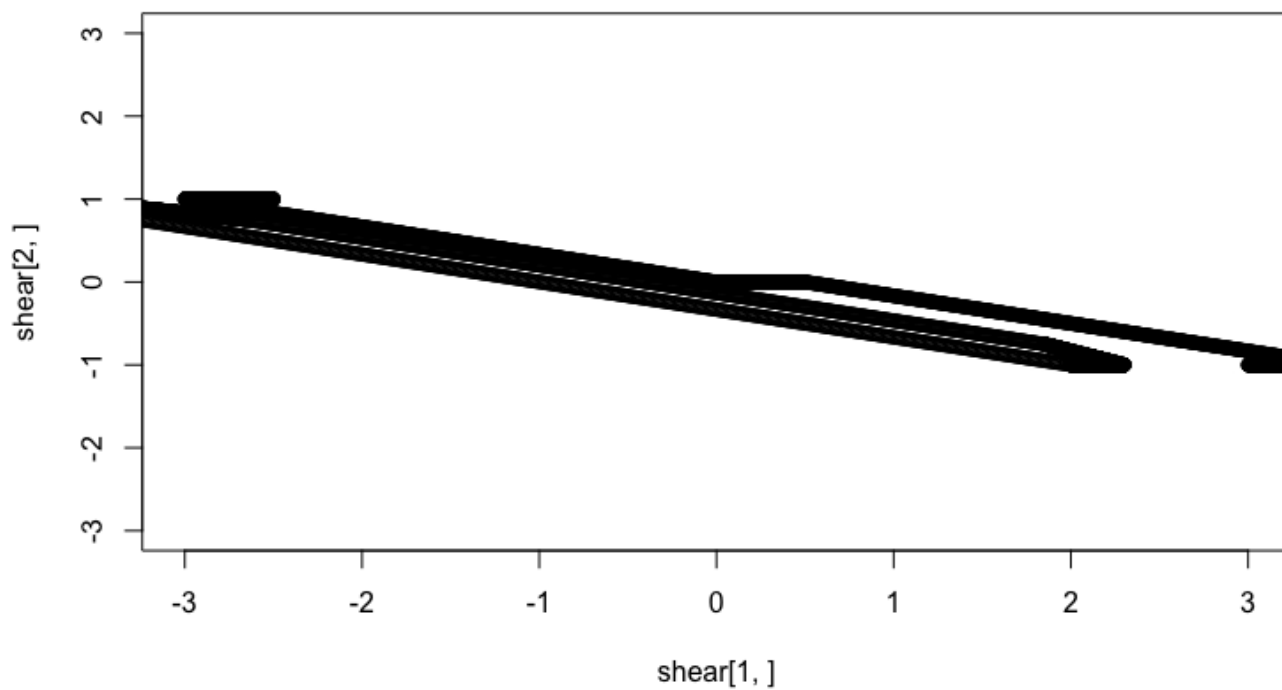
```
for (i in seq(-3,3,length.out=100)){
```

```
  a[1,2]=i
```

```
  shear <- apply(z,2,function(x) a%*%x)
```

```
  plot(shear[2,] ~ shear[1,], xlim=c(-3,3), ylim=c(-3,3))
```

```
}
```

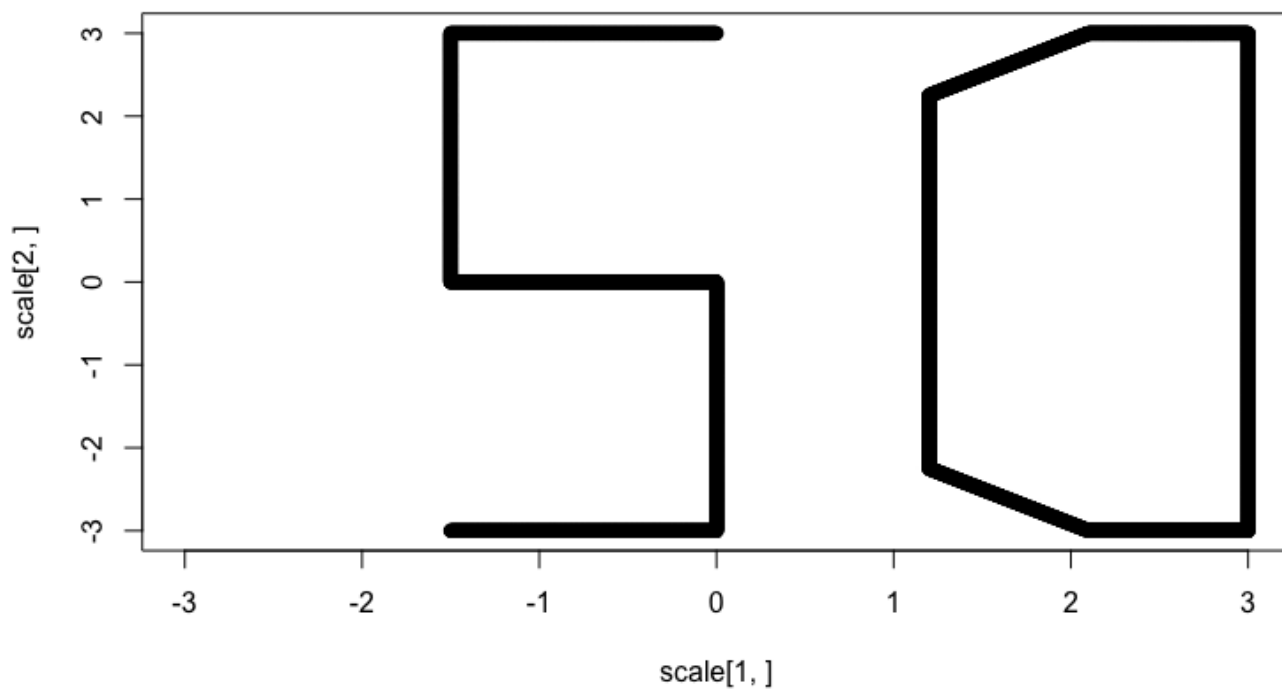


## Scaling

```
# x11()

a <- diag(2)

for (i in seq(-3,3,length.out=100)){
  a[1,1]=i
  a[2,2]=i
  scale <- apply(z,2,function(x) a%%x)
  plot(scale[2,]~scale[1,], xlim=c(-3,3), ylim=c(-3,3))
}
```

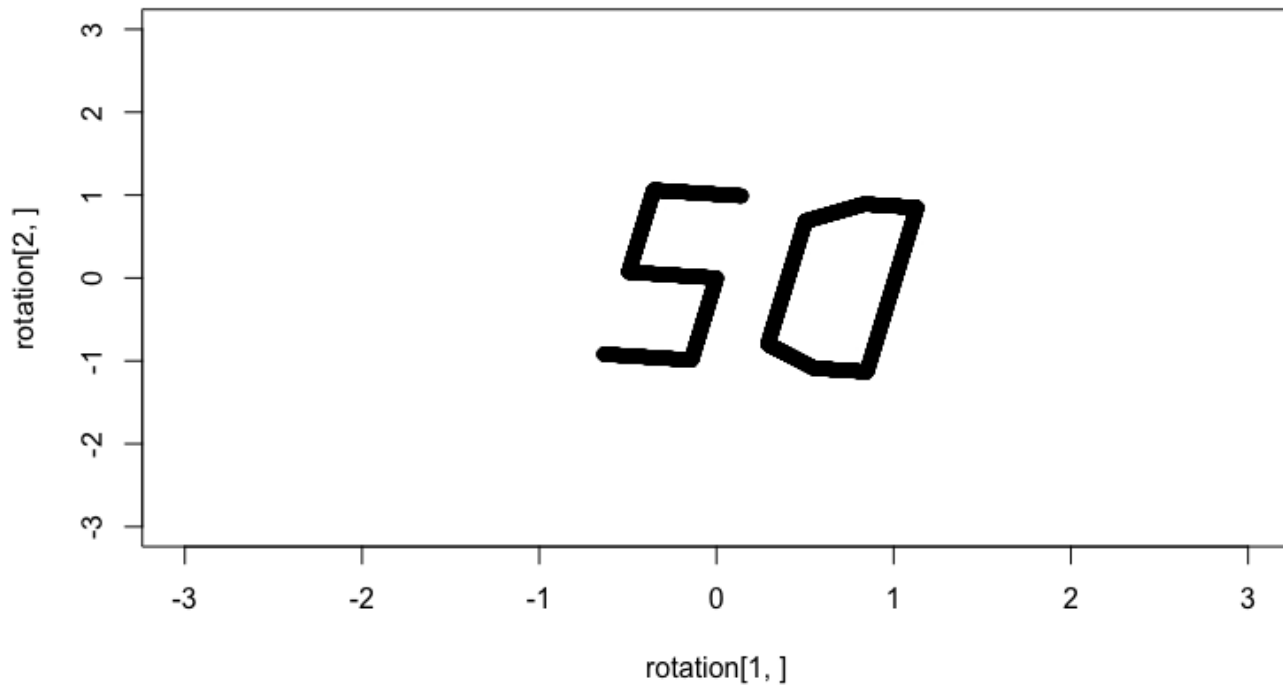


## Rotation

```
# x11()

a <- diag(2)

for (i in seq(-3,3,length.out=100)){
  a <- matrix(c(cos(i), -sin(i), sin(i), cos(i)), nrow = 2, ncol = 2 )
  rotation <- apply(z,2,function(x) a%*%x)
  plot(rotation[2,]~rotation[1,], xlim=c(-3,3), ylim=c(-3,3))
}
```



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