Homework 1

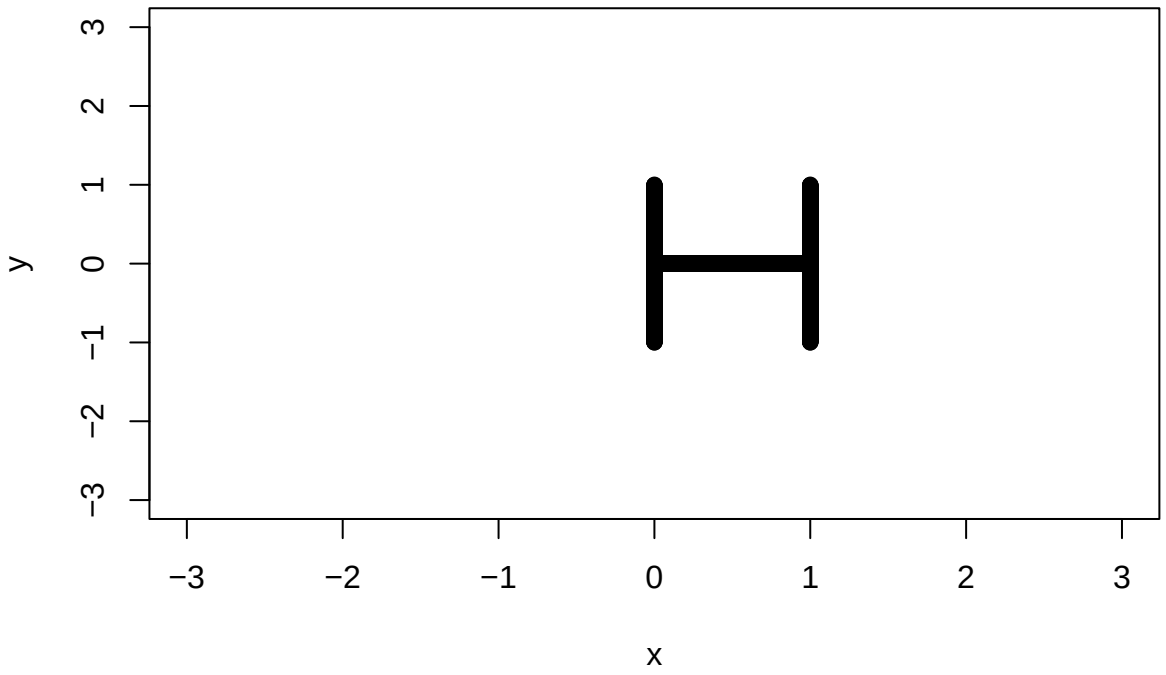
Heleine Fouda

2024-01-28

Getting Started: Setting up the environment.

The Letter 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))
```

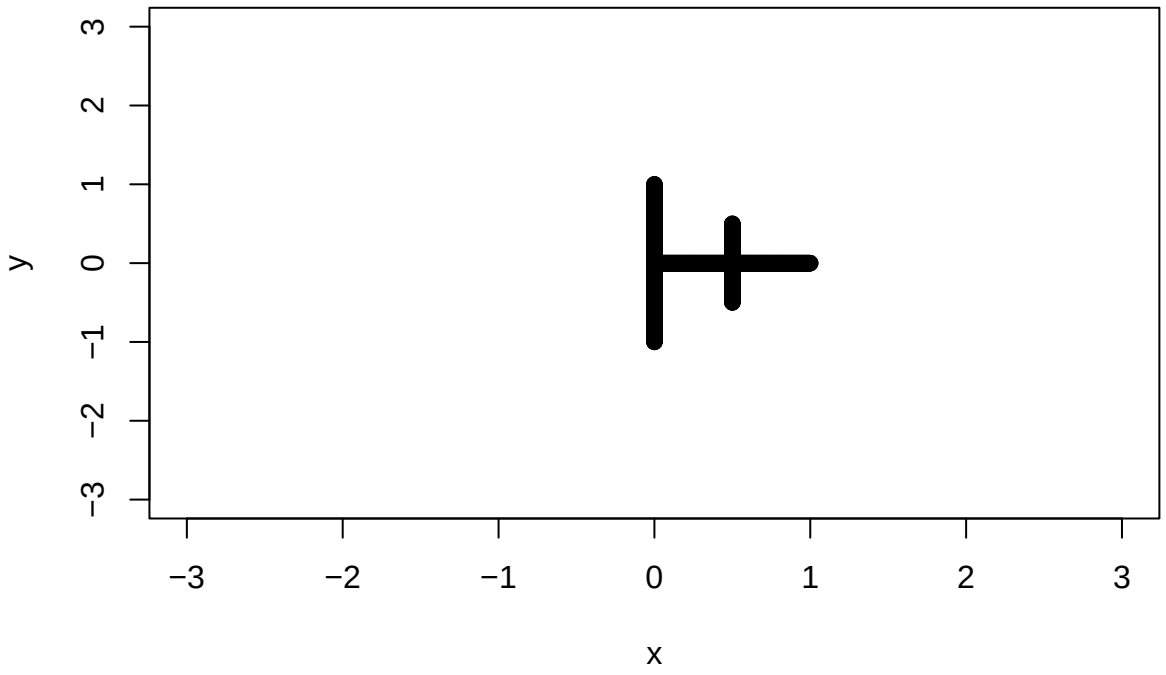



```
# Function to perform matrix-vector multiplication
matrix_multiply <- function(matrix, vector) {
  result <- matrix %*% vector
  return(result)
}

# Function to create an animation
animate_transformations <- function() {
  # Set up the plotting window
  x11()

  # Define the square matrix (initially the Identity matrix)
```

```
  identity_matrix <- matrix(c(1, 0, 0, 0, 1, 0, 0, 0, 1), nrow = 3)

  # Define the vectors of points (e.g., a square)
  points <- matrix(c(0, 1, 1, 1, 1, 0, 0, 0, 1), nrow = 3)

  # Loop through transformations
  for (angle in seq(0, 360, by = 5)) {
    # Shear transformation matrix
    shear_matrix <- matrix(c(1, tan(angle * pi / 180), 0, 0, 1, 0, 0, 0, 1), nrow = 3)

    # Scaling transformation matrix
    scale_matrix <- matrix(c(cos(angle * pi / 180), 0, 0, 0, sin(angle * pi / 180), 0, 0, 0, 1), nrow =

    # Rotation transformation matrix
    rotation_matrix <- matrix(c(cos(angle * pi / 180), -sin(angle * pi / 180), 0, sin(angle * pi / 180)

    # Projection transformation matrix
    projection_matrix <- matrix(c(1, 0, 0, 0, 1, 0, 0, 0, 0), nrow = 3)

    # Apply the transformation
    transformed_points_shear <- matrix_multiply(shear_matrix, points)
    transformed_points_scale <- matrix_multiply(scale_matrix, points)
    transformed_points_rotation <- matrix_multiply(rotation_matrix, points)
    transformed_points_projection <- matrix_multiply(projection_matrix, points)

    # Plot the original and transformed points
    plot(points[1, ], points[2, ], xlim = c(-2, 2), ylim = c(-2, 2), col = "blue", pch = 16, main = "Tra
    points(transformed_points_shear[1, ], transformed_points_shear[2, ], col = "red", pch = 16)
    points(transformed_points_scale[1, ], transformed_points_scale[2, ], col = "green", pch = 16)
    points(transformed_points_rotation[1, ], transformed_points_rotation[2, ], col = "purple", pch = 16
    points(transformed_points_projection[1, ], transformed_points_projection[2, ], col = "orange", pch =

    # Pause for a short time
    Sys.sleep(0.1)
  }
}

# Run the animation
animate_transformations()
```

The Letter F

```
# Generate coordinates for the letter F
x <- c(rep(0, 500), seq(0, 1, length.out = 1000), rep(0, 500), rep(0.5, 500))
y <- c(seq(-1, 1, length.out = 500), rep(0, 1000), seq(1, -1, length.out = 500), seq(0.5, -0.5, length.

# Combine into a matrix
z <- rbind(x, y)

# Plot the letter F
plot(y ~ x, xlim = c(-3, 3), ylim = c(-3, 3))
```




```r
# Function to perform matrix-vector multiplication
matrix_multiply <- function(matrix, vector) {
  result <- matrix %*% vector
  return(result)
}

# Function to create an animation
animate_transformations <- function() {
  # Set up the plotting window
  x11()

  # Generate coordinates for the letter F
  x <- c(rep(0, 500), seq(0, 1, length.out = 1000), rep(0, 500), rep(0.5, 500))
  y <- c(seq(-1, 1, length.out = 500), rep(0, 1000), seq(1, -1, length.out = 500), seq(0.5, -0.5, lengt

  # Combine into a matrix
  z <- cbind(x, y)

  # Identity matrix
  identity_matrix <- matrix(c(1, 0, 0, 0, 1, 0, 0, 0, 1), nrow = 3)

  # Loop through transformations
  for (angle in seq(0, 360, by = 5)) {
    # Rotation transformation matrix
    rotation_matrix <- matrix(c(cos(angle * pi / 180), -sin(angle * pi / 180), sin(angle * pi / 180), c

    # Apply the transformation
    transformed_points_rotation <- t(matrix_multiply(rotation_matrix, t(z)))

    # Plot the original and transformed points
    plot(z[, 1], z[, 2], xlim = c(-3, 3), ylim = c(-3, 3), pch = 16, col = "blue", asp = 1, main = "Tra
    points(transformed_points_rotation[, 1], transformed_points_rotation[, 2], col = "purple", pch = 16
```

```
    # Pause for a short time
    Sys.sleep(0.1)
  }
}

# Run the animation
animate_transformations()
```