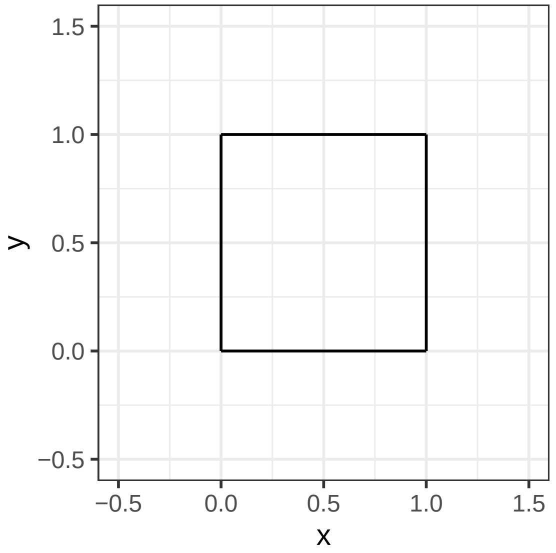
Lab Three – Matrices and Data Frames in R

- Complete the tasks below. Make sure to start your solutions in on a new line that starts with "**Solution**:".
- Make sure to use the Quarto Cheatsheet. This will make completing and writing up the lab *much* easier.

Consider the unit square depicted in Figure 1.

```r
ggplot() +
  geom_segment(aes(x = 0, xend = 0,
                   y=0, yend = 1))+
  geom_segment(aes(x = 0, xend = 1,
                   y=1, yend = 1))+
  geom_segment(aes(x = 1, xend = 1,
                   y=0, yend = 1))+
  geom_segment(aes(x = 0, xend = 1,
                   y=0, yend = 0)) +
  theme_bw() +
  xlim(-0.5, 1.5)+
  ylim(-0.5, 1.5)+
  labs(x = "x",
       y = "y")
```

Figure 1: The unit square.



The unit square can be defined by two basis vectors

$$\mathbf{v}_1 = \begin{bmatrix}1\\0\end{bmatrix} \quad \text{and} \quad \mathbf{v}_2 = \begin{bmatrix}1\\0\end{bmatrix}.$$

From these vectors, we can compute the corner verticies.

$$p_0 = (0,0) = (x_0, y_0) \qquad \text{(The origin)}$$
$$p_1 = v_1 = (x_1, y_1) \qquad \text{(The first basis vector)}$$
$$p_2 = v_1 + v_2 = (x_2, y_2) \qquad \text{(The sum of basis vectors)}$$
$$p_3 = v_1 = (x_3, y_3) \qquad \text{(The second basis vector)}$$

Note that the segments that form the boundary go from p_0 to p_1 to p_2 to p_3.

Consider the matrix M to be the matrix we get from binding the two basis vectors at the column,

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

1 Question 1

Let's transform **M** to demonstrate an interesting linear algebra property.

1.1 Part a

Create the matrix `A` in `R`.

$$\mathbf{A} = \begin{bmatrix} 1 & -7 \\ 5 & 9 \end{bmatrix}$$

Solution:

```
(A = matrix(c(1,5,-7,9),nrow=2,ncol=2))
```

```
     [,1] [,2]
[1,]    1   -7
[2,]    5    9
```

1.2 Part b

Compute the product below in `R` and store it in an object `T`.

$$\mathbf{T} = \mathbf{AM}.$$

Solution:

```
M = matrix(c(1,0,0,1),nrow=2,ncol=2)
T = A %*% M
(T)
```

```
     [,1] [,2]
[1,]    1   -7
[2,]    5    9
```

1.3 Part c

Create `basis.vector.1` (first column of `T`) and `basis.vector.2` (second column of `T`) in `R`.

Solution:

```
(basis.vector.1 = T[,1])
```

```
[1] 1 5
```

```
(basis.vector.2 = T[,2])
```

```
[1] -7  9
```

1.4 Part d

Compute the points p_0, p_1, p_2, and p_3.

Solution:

```
(p0 = c(0,0))
```

```
[1] 0 0
```

```r
(p1 = basis.vector.1)
```

```
[1] 1 5
```

```r
(p2 = basis.vector.1 + basis.vector.2)
```

```
[1] -6 14
```

```r
(p3 = basis.vector.2)
```

```
[1] -7  9
```

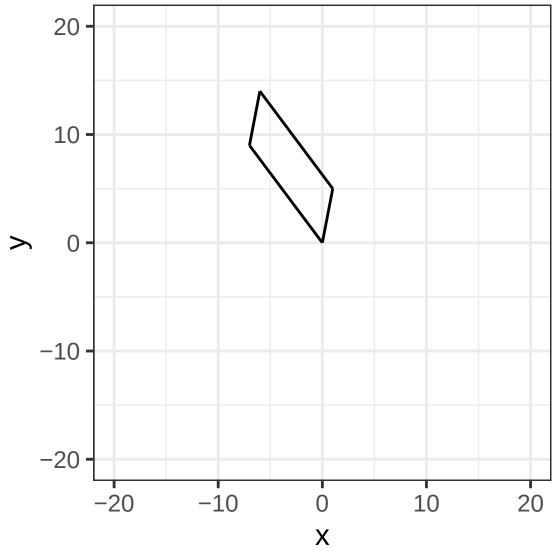
1.5 Part e

Copy and paste the code for the unit square. Edit the code to draw the segments that form the boundary based on the points in Part d.

Solution:

```r
ggplot() +
  geom_segment(aes(x = p0[1], xend = p1[1],
                   y=p0[2], yend = p1[2]))+
  geom_segment(aes(x = p1[1], xend = p2[1],
                   y=p1[2], yend = p2[2]))+
  geom_segment(aes(x = p2[1], xend = p3[1],
                   y=p2[2], yend = p3[2]))+
  geom_segment(aes(x = p3[1], xend = p0[1],
                   y=p3[2], yend = p0[2])) +
  theme_bw() +
  xlim(-20, 20)+
  ylim(-20, 20)+
  labs(x = "x",
       y = "y")
```

Figure 2: My new parallelogram.



1.6 Part f

Interestingly, the determinant of **A** gives us the area of the shape plotted in Part e. Find the determinant of **A**.

Solution:

```r
(det(A))
```

```
[1] 44
```

2 Question 2

Recomplete Question 1 using a new matrix for **A**. This matrix is not as well behaved.

2.1 Part a

Create the matrix `A` in `R`.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

Solution:

```
(A = matrix(c(1,2,2,4),nrow=2,ncol=2))
```

```
     [,1] [,2]
[1,]    1    2
[2,]    2    4
```

2.2 Part b

Compute the product below in `R` and store it in an object `T`.

$$\mathbf{T} = \mathbf{AM}.$$

Solution:

```
M = matrix(c(1,0,0,1),nrow=2,ncol=2)
(T = A %*% M)
```

```
     [,1] [,2]
[1,]    1    2
[2,]    2    4
```

Part c Create `basis.vector.1` (first column of `T`) and `basis.vector.2` (second column of `T`) in `R`.

Solution:

```
(basis.vector.1 = T[,1])
```

```
[1] 1 2
```

```
(basis.vector.2 = 2 * basis.vector.1) #uh oh not full rank
```

```
[1] 2 4
```

Me when my matrix is not full rank ☹ (I'm never gonna get a true inverse)

2.3 Part d

Compute the points p_0, p_1, p_2, and p_3.

Solution:

```
(p0 = c(0,0))
```

```
[1] 0 0
```

```
(p1 = basis.vector.1)
```

```
[1] 1 2
```

```
(p2 = basis.vector.1 + basis.vector.2)
```

```
[1] 3 6
```

```r
(p3 = basis.vector.2)
```

```
[1] 2 4
```

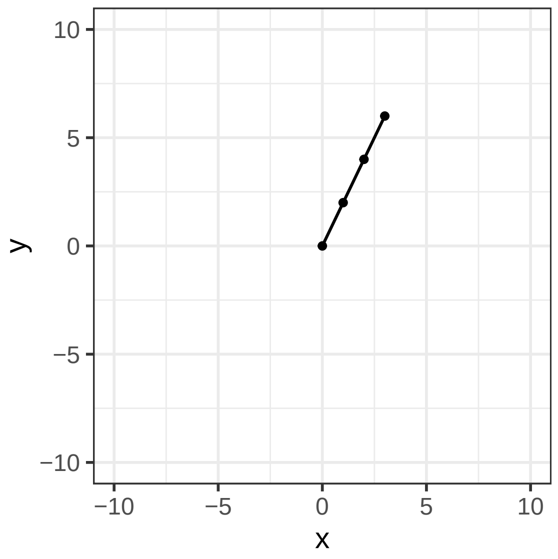
2.4 Part e

Copy and paste the code for plotting the unit square. Edit the code to draw the segments that form the boundary based on the points in Part d.

Solution:

```r
ggplot() +
  geom_segment(aes(x = p0[1], xend = p1[1],
                   y=p0[2], yend = p1[2]))+
  geom_point(aes(x = p0[1], y = p0[2]), size = 1) +
  geom_segment(aes(x = p1[1], xend = p2[1],
                   y=p1[2], yend = p2[2]))+
  geom_point(aes(x = p1[1], y = p1[2]), size = 1) +
  geom_segment(aes(x = p2[1], xend = p3[1],
                   y=p2[2], yend = p3[2]))+
  geom_point(aes(x = p2[1], y = p2[2]), size = 1) +
  geom_segment(aes(x = p3[1], xend = p0[1],
                   y=p3[2], yend = p0[2])) +
  geom_point(aes(x = p3[1], y = p3[2]), size = 1) +
  theme_bw() +
  xlim(-10, 10)+
  ylim(-10, 10)+
  labs(x = "x",
       y = "y")
```

Figure 3: My sad parallelogram.



2.5 Part f

Interestingly, the determinant of **A** gives us the area of the shape plotted in Part e. Find the determinant of **A**.

Solution:

```r
(det(A))
```

```
[1] 0
```

3 Question 3

We can conduct a shear transformation that keeps the area of the shape the same but horizontally or vertically slants the object.

In fact, this is one of the ways artists can make something look like it's leaning or being pushed. Other applications where this is useful include fluid dynamics and crystallography, where the area of a deformed object must remain constant, or in photography where perspective in photos can be adjusted.

A shear transformation is implemented by altering the basis vectors we start with. For vertical slanting,

$$\mathbf{M}_v = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

and for horizontal slanting

$$\mathbf{M}_h = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}.$$

In both cases, $k \in \mathbb{R}$ is called a shear factor and its sign determines the direction of the slant and its magnitude determines the severity of the slant.

3.1 Part a

Use the following to generate a new shape. Compare the shape and its area to your answer in Question 1.

$$\mathbf{M}_v = \begin{bmatrix} 1 & k = 0.5 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} 1 & -7 \\ 5 & 9 \end{bmatrix}$$

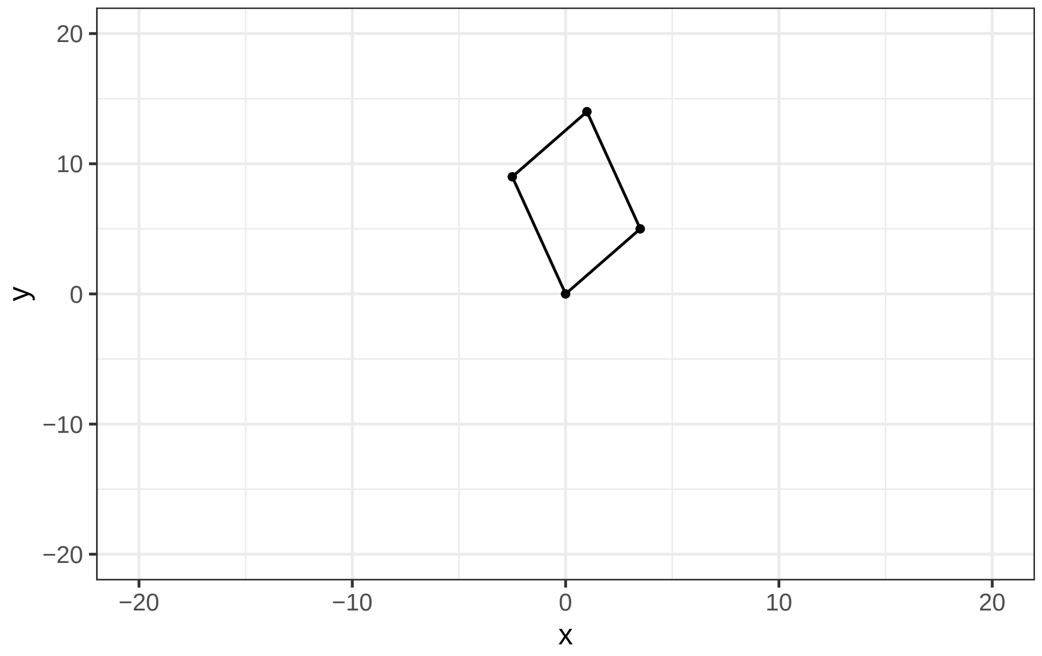
Solution:

```r
#initializing my matrices
Mk = matrix(c(1,0,.5,1),nrow=2,ncol=2)
A = matrix(c(1,5,-7,9),nrow=2,ncol=2)
T = Mk %*% A

#it's basis vector time
basis.vector.1 = T[,1]
basis.vector.2 = T[,2]

#Grabbing my points
p0 = c(0,0)
p1 = basis.vector.1
p2 = basis.vector.1 + basis.vector.2
p3 = basis.vector.2

#Let's plot!
#| label: fig-squarev3
#| fig-cap: "My sad parallelogram."
#| fig-width: 3
#| fig-height: 3
#| scale: "80%"
#| fig-align: "center"
#| size: "scriptsize"
ggplot() +
  geom_segment(aes(x = p0[1], xend = p1[1],
                   y=p0[2], yend = p1[2]))+
  geom_point(aes(x = p0[1], y = p0[2]), size = 1) +
  geom_segment(aes(x = p1[1], xend = p2[1],
                   y=p1[2], yend = p2[2]))+
  geom_point(aes(x = p1[1], y = p1[2]), size = 1) +
  geom_segment(aes(x = p2[1], xend = p3[1],
                   y=p2[2], yend = p3[2]))+
  geom_point(aes(x = p2[1], y = p2[2]), size = 1) +
  geom_segment(aes(x = p3[1], xend = p0[1],
                        y=p3[2], yend = p0[2])) +
  geom_point(aes(x = p3[1], y = p3[2]), size = 1) +
  theme_bw() +
  xlim(-20, 20)+
  ylim(-20, 20)+
  labs(x = "x",
       y = "y")
```



The area of the shape (a parallelogram) is the magnitude of the determinant of the matrix

```
(det(T))
```

```
[1] 44
```

As expected, since the sheer transformation M_k has determinant 1 (just like I_2) and because our determinant function is a homomorphism, $\det(AB) = \det A * \det B$. Thus the determinant is the same as in question 1.

3.2 Part b

Use the following to generate a new shape. Compare the shape and its area to your answer in Questions 1 and the vertical shear transform in part (a).

$$\mathbf{M}_h = \begin{bmatrix} 1 & 0 \\ k = 0.5 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{A} = \begin{bmatrix} 1 & -7 \\ 5 & 9 \end{bmatrix}$$

Solution:

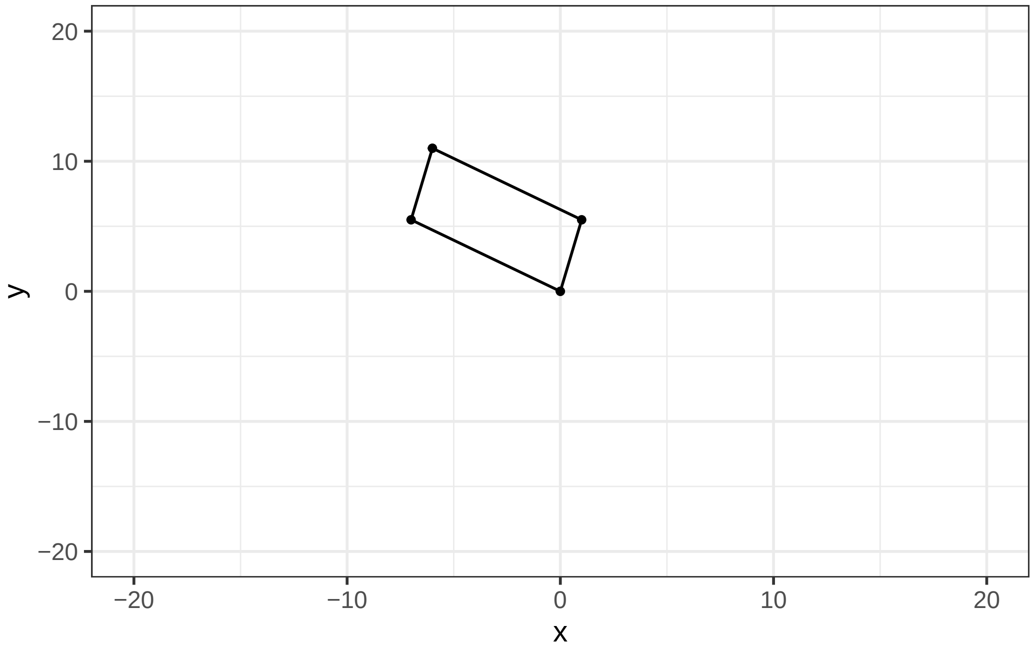
```
#initializing my matrices
Mk = matrix(c(1,.5,0,1),nrow=2,ncol=2)
A = matrix(c(1,5,-7,9),nrow=2,ncol=2)
T = Mk %*% A

#it's basis vector time
basis.vector.1 = T[,1]
basis.vector.2 = T[,2]

#Grabbing my points
```

```r
p0 = c(0,0)
p1 = basis.vector.1
p2 = basis.vector.1 + basis.vector.2
p3 = basis.vector.2

#Let's plot!
#| label: fig-squarev3
#| fig-cap: "My sad parallelogram."
#| fig-width: 3
#| fig-height: 3
#| scale: "80%"
#| fig-align: "center"
#| size: "scriptsize"
ggplot() +
  geom_segment(aes(x = p0[1], xend = p1[1],
                   y=p0[2], yend = p1[2]))+
  geom_point(aes(x = p0[1], y = p0[2]), size = 1) +
  geom_segment(aes(x = p1[1], xend = p2[1],
                   y=p1[2], yend = p2[2]))+
  geom_point(aes(x = p1[1], y = p1[2]), size = 1) +
  geom_segment(aes(x = p2[1], xend = p3[1],
                   y=p2[2], yend = p3[2]))+
  geom_point(aes(x = p2[1], y = p2[2]), size = 1) +
  geom_segment(aes(x = p3[1], xend = p0[1],
                   y=p3[2], yend = p0[2])) +
  geom_point(aes(x = p3[1], y = p3[2]), size = 1) +
  theme_bw() +
  xlim(-20, 20)+
  ylim(-20, 20)+
  labs(x = "x",
       y = "y")
```



Once again, since the determinant of our matrix M_k is 1, we expect the determinant to be the same as in question 1. Let's check!

```r
det(T)
```

```
[1] 44
```

3.3 Part c

Can we shear vertically and horizontally at the same time? Use the following to generate a new shape. Compare the shape and its area to your answers in parts (a) and (b).

$$\mathbf{M}_{vh} = \begin{bmatrix} 1 & k = 0.5 \\ k = 0.5 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} 1 & -7 \\ 5 & 9 \end{bmatrix}$$

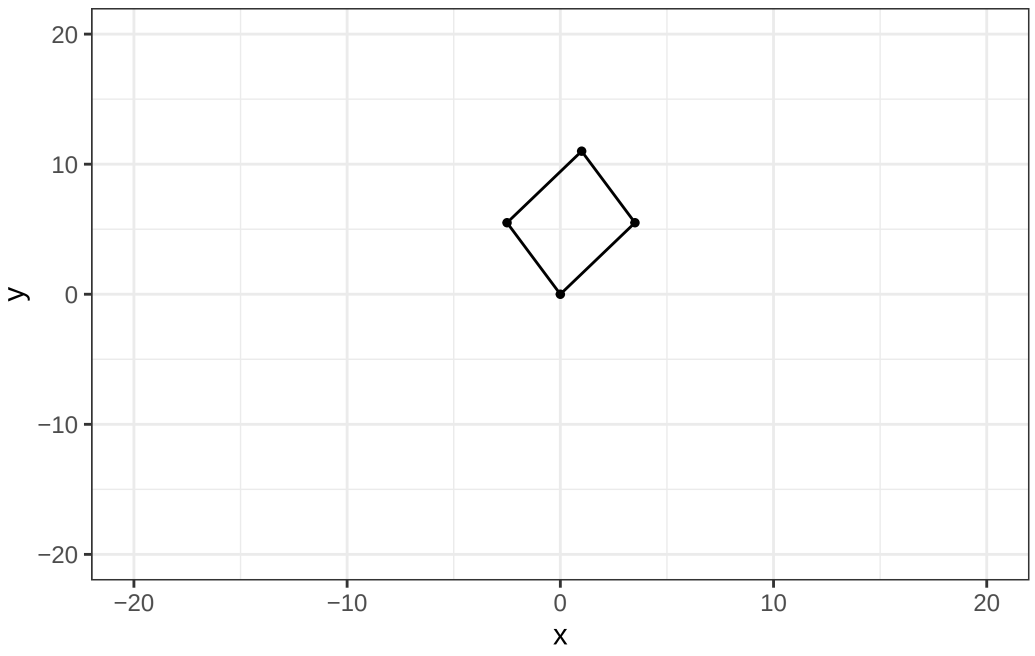
Solution:

```r
#initializing my matrices
Mk = matrix(c(1,.5,.5,1),nrow=2,ncol=2)
A = matrix(c(1,5,-7,9),nrow=2,ncol=2)
T = Mk %*% A

#it's basis vector time
basis.vector.1 = T[,1]
basis.vector.2 = T[,2]

#Grabbing my points
p0 = c(0,0)
p1 = basis.vector.1
p2 = basis.vector.1 + basis.vector.2
p3 = basis.vector.2

#Let's plot!
#| label: fig-squarev3
#| fig-cap: "My sad parallelogram."
#| fig-width: 3
#| fig-height: 3
#| scale: "80%"
#| fig-align: "center"
#| size: "scriptsize"
ggplot() +
  geom_segment(aes(x = p0[1], xend = p1[1],
                   y=p0[2], yend = p1[2]))+
  geom_point(aes(x = p0[1], y = p0[2]), size = 1) +
  geom_segment(aes(x = p1[1], xend = p2[1],
                   y=p1[2], yend = p2[2]))+
  geom_point(aes(x = p1[1], y = p1[2]), size = 1) +
  geom_segment(aes(x = p2[1], xend = p3[1],
                   y=p2[2], yend = p3[2]))+
  geom_point(aes(x = p2[1], y = p2[2]), size = 1) +
  geom_segment(aes(x = p3[1], xend = p0[1],
                   y=p3[2], yend = p0[2])) +
  geom_point(aes(x = p3[1], y = p3[2]), size = 1) +
  theme_bw() +
  xlim(-20, 20)+
  ylim(-20, 20)+
  labs(x = "x",
       y = "y")
```



This matrix no longer has determinant one (because the product of the off-diagonal is no longer 0). So this parallelogram should have a different area from the sheer transformations and the original. Let's check!

```r
det(T)
```

```
[1] 33
```

# Question 4

3.4 Part a

Create a data frame `ladder` with the following data. Note your data frame should have three columns; I split it to save vertical space here.

x	y	group	x	y	group
0	0	1	1	8	1
0	20	1	0	10	1
1	0	2	1	10	2
1	20	2	0	12	2
0	2	3	1	12	3
1	2	3	0	14	3
0	4	4	1	14	4
1	4	4	0	16	4
0	6	5	1	16	5
1	6	5	0	18	5
0	8	5	1	18	5

3.5 Part b

Remove `#|eval: false` in the code below. The result should look like a ladder and a very rectangular house if part a is correct.

```r
ggplot() +
  geom_path(data = ladder,
            aes(x = x, y = y, group = group)) +
                geom_segment(aes(x = 10, xend = 10,
                 y=0, yend = 20))+
  geom_segment(aes(x = 10, xend = 20,
                 y=20, yend = 20))+
  geom_segment(aes(x = 20, xend = 20,
                 y=0, yend = 20))+
  geom_segment(aes(x = 10, xend = 20,
                 y=0, yend = 0)) +
  theme_bw() +
  xlim(-2,22) +
  ylim(-2,22)
```

3.6 Part c

Use `as.matrix()` to create a 22×2 matrix $\mathbf{A}$ containing the `x` and `y` observations from the `ladder` data frame. Use matrix multiplication to compute

$$\mathbf{T} = \mathbf{A}\mathbf{M}_h.$$

Recall,

$$\mathbf{M}_h = \begin{bmatrix} 1 & 0 \\ k & 1 \end{bmatrix}$$

and use $k = 0.5$.

3.7 Part d

Create a data frame `leaning.ladder` with the `x` and `y` equal to the first and second column of $\mathbf{T}$, respectively, and the same values of `group` from `ladder`.

3.8 Part e

Remove `eval: false` in the code below. The result should look like a ladder and a very rectangular house if part a is correct. The new leaning ladder should look like it is leaning on the house.

```r
ggplot() +
  geom_path(data = ladder,
            aes(x = x, y = y, group = group, color = "Starting Ladder")) +
  geom_path(data = leaning.ladder,
            aes(x = x, y = y, group = group, color = "Leaning Ladder")) +
  geom_segment(aes(x = 10, xend = 10,
                 y=0, yend = 20))+
  geom_segment(aes(x = 10, xend = 20,
                 y=20, yend = 20))+
  geom_segment(aes(x = 20, xend = 20,
                 y=0, yend = 20))+
  geom_segment(aes(x = 10, xend = 20,
                 y=0, yend = 0)) +
  theme_bw() +
  xlim(-2,22) +
  ylim(-2,22) +
  scale_color_manual("", values=c("darkred", "grey"))
```