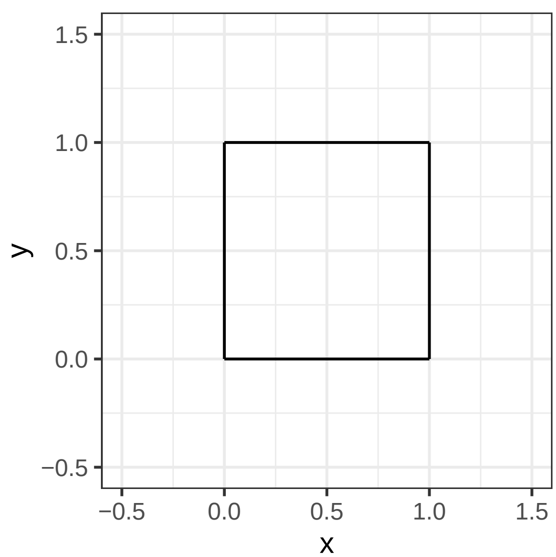
Lab Three – Matrices and Data Frames in R

- 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.

```
p0 <- c(0, 0)
p1 <- c(1, 0)
p2 <- c(1, 1)
p3 <- c(0, 1)
ggplot() +
  geom_segment(aes(x = p0[1], y = p0[2], xend = p1[1], yend = p1[2])) +
  geom_segment(aes(x = p1[1], y = p1[2], xend = p2[1], yend = p2[2])) +
  geom_segment(aes(x = p2[1], y = p2[2], xend = p3[1], yend = p3[2])) +
  geom_segment(aes(x = p3[1], y = p3[2], xend = p0[1], yend = p0[2])) +
  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

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

From these vectors, we can compute the corner vertices.

$$\begin{aligned}
p_0 &= (0, 0) = (x_0, y_0) && \text{(The origin)} \\
p_1 &= v_1 = (x_1, y_1) = (1, 0) && \text{(The first basis vector)} \\
p_2 &= v_1 + v_2 = (x_2, y_2) = (1, 1) && \text{(The sum of basis vectors)} \\
p_3 &= v_2 = (x_3, y_3) = (0, 1) && \text{(The second basis vector)}
\end{aligned}$$

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

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 $\mathbf{M}$ 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(data=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`.

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

Solution

Create object M

```
(M=matrix(data=c(1,0,0,1),
          nrow=2, ncol=2))
```

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

$T = AM$

```
(T=(A %*% M))
```

```
     [,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

$p_0 = (0, 0)$

$p_1 = (1, 5)$

$p_2 = (-6, 14)$

```
basis.vector.1+basis.vector.2
```

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

$p_3 = (-7, 9)$

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

Install `tidyverse` (Wickham et al. 2019)

```
install.packages("tidyverse")
library("tidyverse")
```

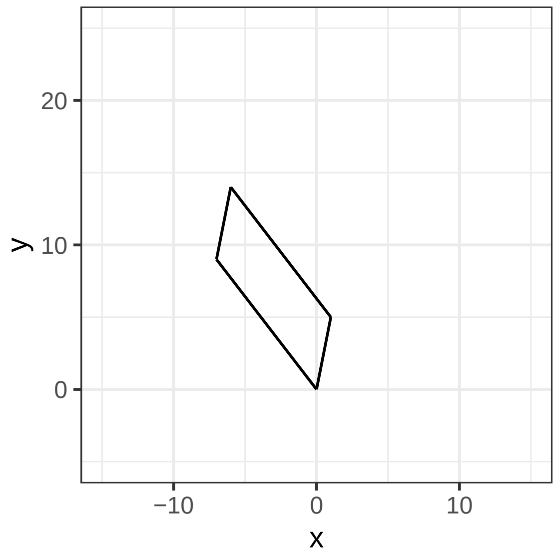
Create vectors

```
p0=c(0,0)
p1=c(1,5)
p2=c(-6,14)
p3=c(-7,9)
```

Draw segments

```
ggplot() +
  geom_segment(aes(x = p0[1], y = p0[2], xend = p1[1], yend = p1[2])) +
  geom_segment(aes(x = p1[1], y = p1[2], xend = p2[1], yend = p2[2])) +
  geom_segment(aes(x = p2[1], y = p2[2], xend = p3[1], yend = p3[2])) +
  geom_segment(aes(x = p3[1], y = p3[2], xend = p0[1], yend = p0[2])) +
  theme_bw() +
  xlim(-15, 15) +
  ylim(-5,25) +
  labs(x = "x", y = "y")
```

Figure 2: Figure 1e.



1.6 Part f

Interestingly, the determinant of $\mathbf{A}$ gives us the area of the shape plotted in Part e. Find the determinant of $\mathbf{A}$.

Solution

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

```
[1] 44
```

2 Question 2

Recomplete Question 1 using a new matrix for $\mathbf{A}$. 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(data=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

```r
(T=(A %*% M))
```

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

2.3 Part c

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

Solution

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

```
[1] 1 2
```

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

```
[1] 2 4
```

2.4 Part d

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

Solution

$p_0 = (0, 0)$

$p_1 = (1, 2)$

$p_2 = (3, 6)$

$p_3 = (2, 4)$

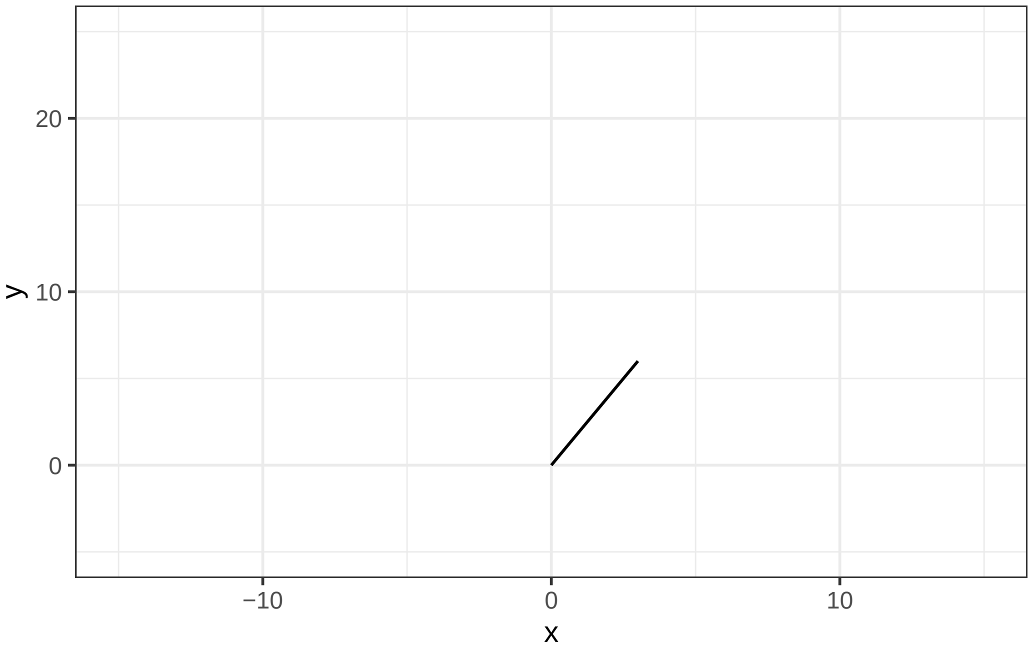
2.5 Part e

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

#| label: fig-unitsquare2e
#| fig-cap: "Figure 2e."
#| fig-width: 3
#| fig-height: 3
#| scale: "80%"
#| fig-align: "center"
#| size: "scriptsize"
ggplot() +
  geom_segment(aes(x = p0[1], y = p0[2], xend = p1[1], yend = p1[2])) +
  geom_segment(aes(x = p1[1], y = p1[2], xend = p2[1], yend = p2[2])) +
  geom_segment(aes(x = p2[1], y = p2[2], xend = p3[1], yend = p3[2])) +
  geom_segment(aes(x = p3[1], y = p3[2], xend = p0[1], yend = p0[2])) +
  theme_bw() +
  xlim(-15, 15) +
  ylim(-5,25) +
  labs(x = "x", y = "y")
```



2.6 Part f

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

Solution

```
(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

The shape is slanted vertically more (the top edge moves up a bit, while the right edge stays the same), but their areas are the same. The new shape is generated by the same matrix A as question 1, and the determinant of matrix $\mathbf{M}_v$ is 1. Thus, the area, calculated by getting the determinant of $\mathbf{T}$, will result in the determinant of $\mathbf{A}$, which is 44.

Creating matrices

```
M_v=matrix(data=c(1,0,0.5,1),
        nrow=2,ncol=2)
A=matrix(data=c(1,5,-7,9),
        nrow=2,ncol=2)
```

Creating basis vectors

```
T=(A %*% M_v)
basis.vector.1=T[,1]
basis.vector.2=T[,2]
p0=c(0,0)
p1=basis.vector.1
p3=basis.vector.2
p2=(p1+p3)
```

Calculating the determinant of $\mathbf{M}_v$ and area

```
(det(M_v))
```

```
[1] 1
```

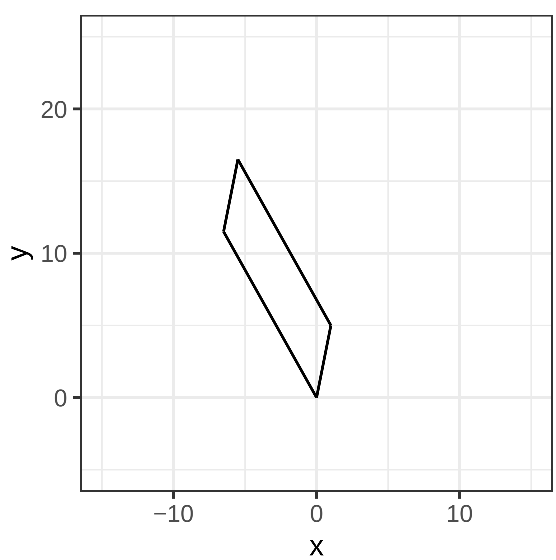
```
(det(T))
```

```
[1] 44
```

Creating figure

```
ggplot() +
  geom_segment(aes(x = p0[1], y = p0[2], xend = p1[1], yend = p1[2])) +
  geom_segment(aes(x = p1[1], y = p1[2], xend = p2[1], yend = p2[2])) +
  geom_segment(aes(x = p2[1], y = p2[2], xend = p3[1], yend = p3[2])) +
  geom_segment(aes(x = p3[1], y = p3[2], xend = p0[1], yend = p0[2])) +
  theme_bw() +
  xlim(-15, 15) +
  ylim(-5,25) +
  labs(x = "x", y = "y")
```

Figure 3: Figure 3a.



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).

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

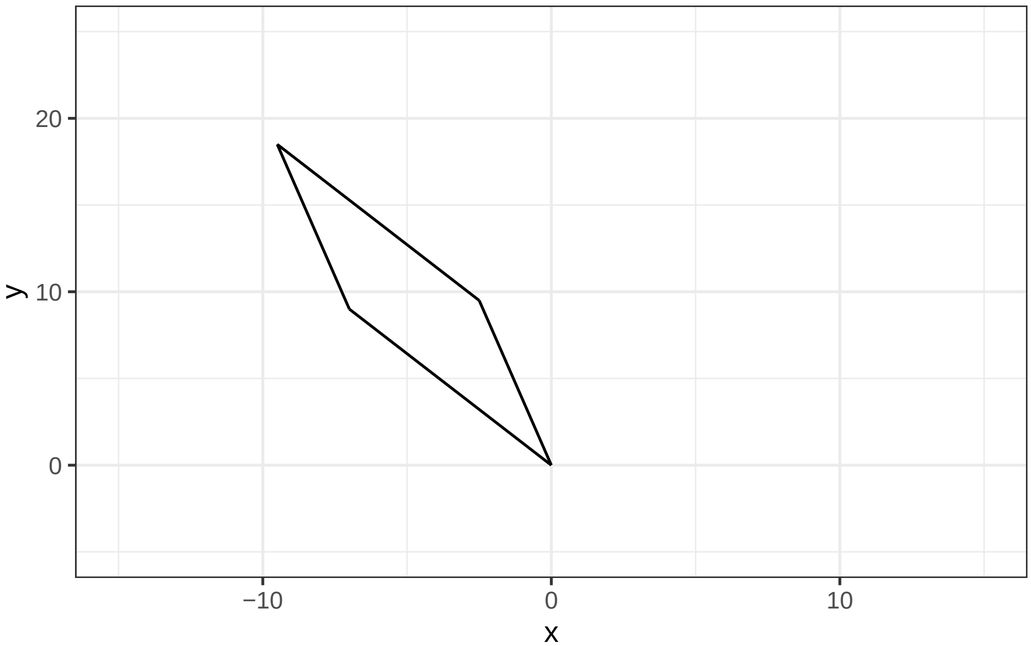
Solution

Compared to the shape in question 1, this shape is slanted horizontally more (the long bottom edge stays the same, but the right edge leans to the right). Compared to part a, the acute angles of the parallelogram is narrower. Similar to explanation in part a, the areas of the shape in 3b equals to those of the previous shapes and is the determinant of $\mathbf{A}$ because the the determinant of matrix $\mathbf{M}_h$ is 1.

Creating figure

```
M_h=matrix(data=c(1,0.5,0,1),
        nrow=2,ncol=2)
A=matrix(data=c(1,5,-7,9),
        nrow=2,ncol=2)
T=(A %*% M_h)
basis.vector.1=T[,1]
basis.vector.2=T[,2]
p0=c(0,0)
p1=basis.vector.1
p3=basis.vector.2
p2=(p1+p3)

#| label: fig-unitsquare3b
#| fig-cap: "Figure 3b."
#| fig-width: 3
#| fig-height: 3
#| scale: "80%"
#| fig-align: "center"
#| size: "scriptsize"
ggplot() +
  geom_segment(aes(x = p0[1], y = p0[2], xend = p1[1], yend = p1[2])) +
  geom_segment(aes(x = p1[1], y = p1[2], xend = p2[1], yend = p2[2])) +
  geom_segment(aes(x = p2[1], y = p2[2], xend = p3[1], yend = p3[2])) +
  geom_segment(aes(x = p3[1], y = p3[2], xend = p0[1], yend = p0[2])) +
  theme_bw() +
  xlim(-15, 15) +
  ylim(-5,25) +
  labs(x = "x", y = "y")
```



Calculate area

```
(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

This shape is slanted both vertically and horizontally, and looks every different from question 1 and part (a) and (c). Its acute angles are the narrowest. The area is 33 and is smaller than those of the three previous shapes.

Creating matrices

```
M_vh=matrix(data=c(1,0.5,0.5,1),
        nrow=2,ncol=2)
A=matrix(data=c(1,5,-7,9),
        nrow=2,ncol=2)
```

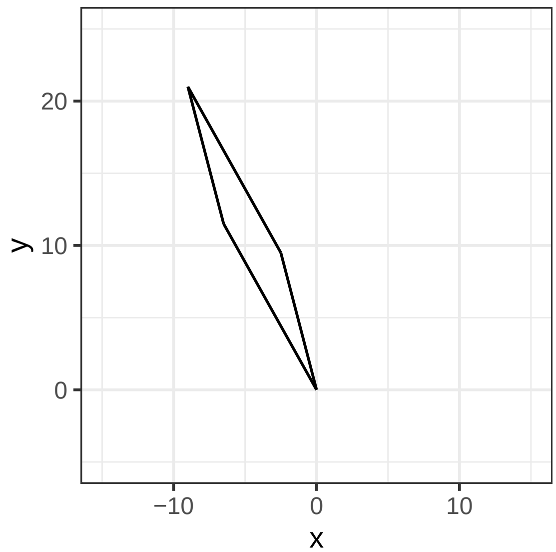
Creating basis vectors:

```
T=(A %*% M_vh)
basis.vector.1=T[,1]
basis.vector.2=T[,2]
p0=c(0,0)
p1=basis.vector.1
p3=basis.vector.2
p2=(p1+p3)
```

Creating figure:

```
ggplot() +
  geom_segment(aes(x = p0[1], y = p0[2], xend = p1[1], yend = p1[2])) +
  geom_segment(aes(x = p1[1], y = p1[2], xend = p2[1], yend = p2[2])) +
  geom_segment(aes(x = p2[1], y = p2[2], xend = p3[1], yend = p3[2])) +
  geom_segment(aes(x = p3[1], y = p3[2], xend = p0[1], yend = p0[2])) +
  theme_bw() +
  xlim(-15, 15) +
  ylim(-5,25) +
  labs(x = "x", y = "y")
```

Figure 4: Figure 3b.



Calculating area of shape

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

```
[1] 33
```

4 Question 4

4.1 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	6
0	20	1	0	10	7
1	0	2	1	10	7
1	20	2	0	12	8
0	2	3	1	12	8
1	2	3	0	14	9
0	4	4	1	14	9
1	4	4	0	16	10
0	6	5	1	16	10
1	6	5	0	18	11
0	8	6	1	18	11

Solution

```r
(ladder=data.frame(x=c(0,0,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1),y=c(0,20,0,20,2,3,4,4,6,6,8,8,10,10,12,
```

```
   x  y group
1  0  0     1
2  0 20     1
3  1  0     2
4  1 20     2
5  0  2     3
6  1  3     3
7  0  4     4
8  1  4     4
9  0  6     5
10 1  6     5
11 0  8     6
12 1  8     6
13 0 10     7
14 1 10     7
15 0 12     8
16 1 12     8
17 0 14     9
18 1 14     9
19 0 16    10
20 1 16    10
21 0 18    11
22 1 18    11
```

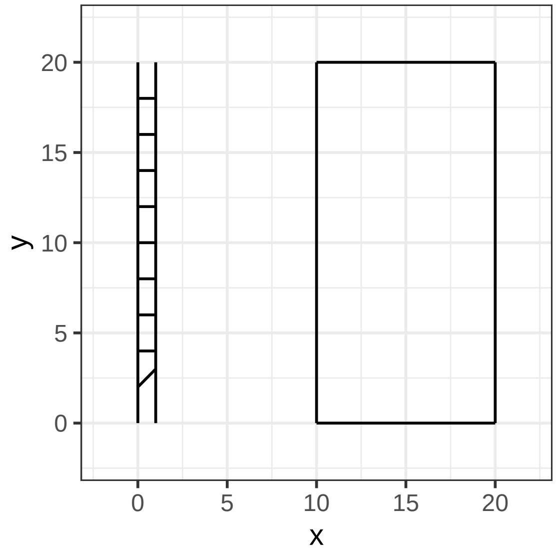
4.2 Part b

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

Solution

```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)
```

Figure 5: A ladder.



4.3 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 $k = 0.5$.

Solution

```
A=as.matrix(ladder[c("x","y")])
M=matrix(data=c(1,0.5,0,1),
        nrow=2,ncol=2)
(T=(A%*%M))
```

```
      [,1] [,2]
 [1,]  0.0    0
 [2,] 10.0   20
 [3,]  1.0    0
 [4,] 11.0   20
 [5,]  1.0    2
 [6,]  2.5    3
 [7,]  2.0    4
 [8,]  3.0    4
 [9,]  3.0    6
[10,]  4.0    6
[11,]  4.0    8
[12,]  5.0    8
[13,]  5.0   10
[14,]  6.0   10
[15,]  6.0   12
[16,]  7.0   12
[17,]  7.0   14
[18,]  8.0   14
```

```
[19,]  8.0   16
[20,]  9.0   16
[21,]  9.0   18
[22,] 10.0   18
```

4.4 Part d

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

Solution

```r
(leaning.ladder=data.frame(x=T[,1],y=T[,2],group=ladder[,3]))
```

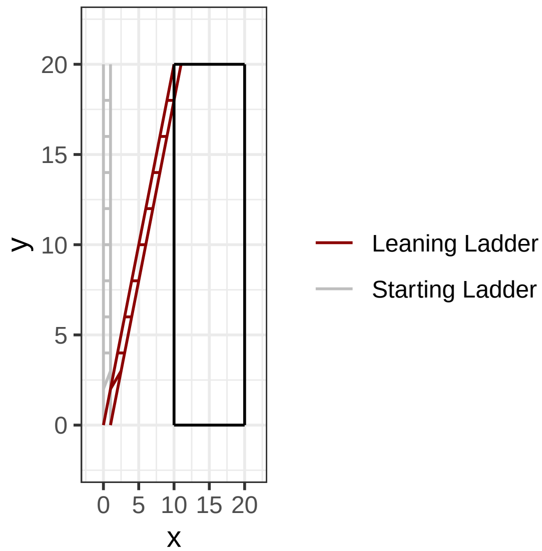
```
      x  y group
1   0.0  0     1
2  10.0 20     1
3   1.0  0     2
4  11.0 20     2
5   1.0  2     3
6   2.5  3     3
7   2.0  4     4
8   3.0  4     4
9   3.0  6     5
10  4.0  6     5
11  4.0  8     6
12  5.0  8     6
13  5.0 10     7
14  6.0 10     7
15  6.0 12     8
16  7.0 12     8
17  7.0 14     9
18  8.0 14     9
19  8.0 16    10
20  9.0 16    10
21  9.0 18    11
22 10.0 18    11
```

4.5 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"))
```

Figure 6: A ladder and a leaning ladder.



References

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.