

Chapter 2 Problems

Ryan Heslin

January 10, 2022

Note on transposes: a matrix can be seen either as a collection of linear combination of scalars to apply to other matrices, or as bases for such a transformation. Plotting the column vectors takes the second perspective (as linear combinations of the identity). Plotting a system of equations shows every possible linear combination of each row that equals the corresponding element of b ; intersections between the row subspaces are points of solutions. Transposing converts from one form to another

Diagonal matrices are special because each dimension of the RHS scales only the same dimension of the RHS, so no shearing occurs. This is why the cross product zeroes the diagonals

For projection, if all but one coordinate is fixed for a set of vectors, all those vectors have parallel orthogonal components.

An inverse is the unique matrix for which the two interpretations of matrix multiplication - sums of columns multiplied by scalars, or columns of row-vector dot products - give the same result. This is only possible for square matrices, obviously.

2.1

24.

To prove the determinant formula, just conduct row elimination and record the operations on the inverse. To clear out the second row, we subtract by c/a of the first row and divide by the resulting value of d to get denominators of $d - \frac{bc}{a}$. Cancellation leaves $-c$ and b over $ad - bc$. We clear out the first row and factor out the common $ad - bc$. The 1, 1 element then becomes $ad - bc + bc$, the left $-ab$. Dividing by a gives the final formula $\frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$. So the off-dimension interactions are inverted and the common-dimension swapped.

An inverse is the unique matrix for which the two interpretations of matrix multiplication - sums of columns multiplied by scalars, or columns of row-vector dot products - give the same result. This is only possible for square matrices, obviously.

37.

If x lies on a line between v and w , it is spanned by those vectors and can be arrived at by some linear combination of them. From the second property of linear transformations

$$c_1v + c_2w = xc_1\mathbf{T}(v) + c_2\mathbf{T}(w) = \mathbf{T}(x)$$

so x still lies on the line. A linear transformation of a line maps it onto another line.

40.

A transformation may be expressed

$$\begin{aligned}\mathbf{T}(x) &= x_1\mathbf{T}(e_1) + x_2\mathbf{T}(e_2) + \cdots + x_m\mathbf{T}(e_m) \\ &= \mathbf{T}(x_1e_1) + \mathbf{T}(x_2e_2) + \cdots + \mathbf{T}(x_me_m)\end{aligned}$$

By the addition property:

$$\begin{aligned}&= \mathbf{T}(x_1e_1 + x_2e_2 + \cdots + x_me_m) \\ &= \mathbf{T}(\mathbf{I}x) \\ \mathbf{T}(x) &= \mathbf{T}(x)\end{aligned}$$

Any vector can be expressed as a linear combination of the identity and its elements, so the identity is proved.

40.

All $\mathbf{R} \rightarrow \mathbf{R}$ linear transformations are simply scalar multiplication, increasing or reduce a scalar x .

41.

\mathbf{R}^2 to \mathbf{R} transformations project vectors onto the number line. The graph resembles a vertical line connecting the vector to the x_1 axis.

42.

We see the shadow of the cube.

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.6      v dplyr  1.0.7
## v tidyr   1.1.4      v stringr 1.4.0
## v readr   2.1.0      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(matador)

## Loading required package: magrittr

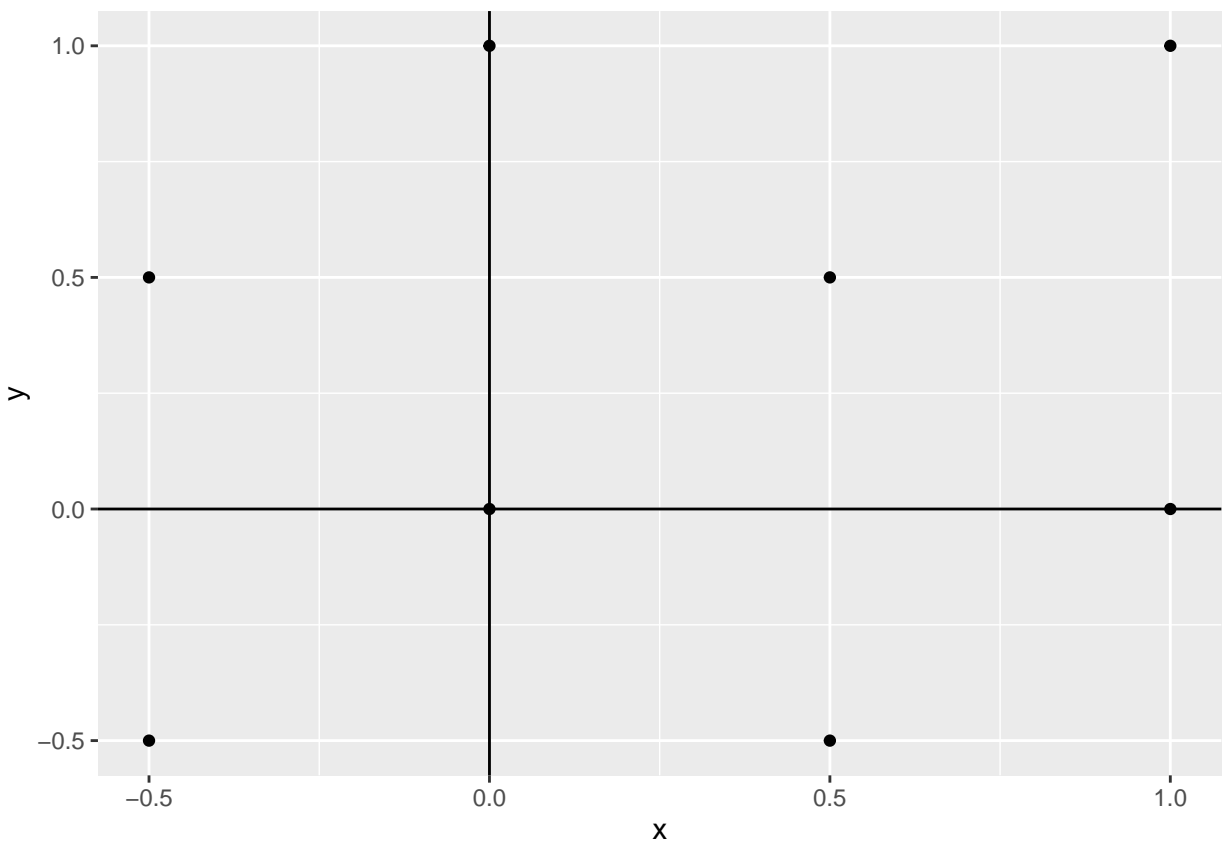
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##   set_names

## The following object is masked from 'package:tidyr':
##
##   extract

cube <- expand.grid(c(0, 1), c(0, 1), c(0, 1)) %>%
  as.matrix() %>%
  t()

proj <- cbind(rep(-.5, 2), diag(nrow = 2)) %*% cube
proj %>%
  t() %>%
  as.data.frame() %>%
  setNames(c("x", "y")) %>%
  ggplot(aes(x, y)) +
  geom_point() +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0)
```



The null space of the transformation is

$$\begin{bmatrix} x_1 \\ .5x_1 \\ .5x_1 \end{bmatrix}$$

43.

The dot product can be represented as $v_1^T v_2$, so it is linear.

For a $\mathbf{R}^3 \rightarrow \mathbf{R}$ transformation, the matrix is a 3×1 projection. If we define this as v , the dot product of v and any x gives the projection.

44.

We need a 3×3 matrix here

The definition can be expanded:

$$\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

What an interesting pattern - the diagonals (the distance in the same dimension the vectors point in the same direction) get zeroed out, and the signs of the off-diagonals, (distance moved in opposite directions) are opposites. If we multiplied by -1 we could reverse the right-hand rule.

45.

The composite transformation is linear.

$$\mathbf{T}(x + h) = \mathbf{T}(x) + \mathbf{T}(h)k\mathbf{T}(x) = \mathbf{T}(kx)\mathbf{S}(\mathbf{T}(x) + \mathbf{S}(\mathbf{T}(h)) = \mathbf{S}$$

50.

Only the trivial solution to the transition matrix exists, so there is no distribution vector.

To solve the transition matrix, first find the matrix \mathbf{A} of $\mathbf{A}x = y$ (the transition matrix itself). For each node, the coefficients of the other nodes are the the inverse of the number of connected nodes, so if x_1 connects only to x_4 its vector is $(0, 0, 0, 1)$.

Then note we are trying to solve for an equilibrium where each $x_i = 0$. Set the augment column to 0, then replace the diagonals with -1 (to represent that each node should lose as much traffic as it gains).

For instance:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1/2 & 0 & 1/2 & 1 \\ 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1/2 & -1 & 1/2 & 1 \\ 1/2 & 0 & -1 & 0 \\ 0 & 0 & 1/2 & -1 \end{bmatrix}$$

Then solve the homogeneous system. With the solution, find the value of t that results in the elements summing to 1. Then solve for each component in terms of that value of t . If any elements are negative, it is not a distribution vector, and no even distribution is possible.

51.

The distribution vector is $(1/13, 4/13, 3/13, 5/13)$.

53.

We obtain the distribution matrix by setting each node's coefficients to .05 and evenly distributing the remaining .8 among connected nodes. Note the diagonals are 1-.05 (traffic staying on each node). The obvious formula is:

$$\mathbf{B} = .2/\mathbf{NE} + .8\mathbf{A}$$

, four-fifths sent to connected nodes and the remainder divided evenly (E is the identity) because we assume 5% of traffic randomly stays in each node.

I don't really feel like finding the distribution vector.

```
library(pracma)

##
## Attaching package: 'pracma'

## The following objects are masked from 'package:matador':
##
##   angle, dot

## The following objects are masked from 'package:magrittr':
##
##   and, mod, or

## The following object is masked from 'package:purrr':
##
##   cross

## The following object is masked from '.my_funs':
##
##   clear

library(matador)
dist <- matrix(c(
  -.95, .45, .05, .05,
  .45, -.95, .05, .85,
  .45, .45, -.95, .05,
  .05, .05, .85, -.95
), nrow = 4, byrow = T)

mat2latex(rref(dist))
```

$$\begin{bmatrix} 1 & 0 & 0 & -0.714285714285714 \\ 0 & 1 & 0 & -1.28571428571429 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

55.

Solve a nasty distribution matrix.

```
library(pracma)
mat <- matrix(c(
  -.95, .45, .45, .05,
  .05, -.95, .05, .85,
  19 / 60, 19 / 60, -.95, 19 / 60,
  .05, .45, .45, -.95
), byrow = TRUE, nrow = 4)

mat2latex(rref(mat))
```

$$\begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

57.

We solve the system to get 14 5-franc and 37 2-franc coins. The matrix to get the value and number of coins from the number of each is the inverse of the transformation matrix. I'll let R handle this:

```
mat2latex(solve(matrix(c(2, 1, 5, 1), nrow = 2)) %*% as.matrix(c(144, 51)))
```

$$\begin{bmatrix} 37 \\ 14 \end{bmatrix}$$

Affine transformations aren't linear because THEY ARE NONVECTORIZED

Diagonal matrices are special because each dimension of the RHS scales only the same dimension of the RHS, so no shearing occurs. This is why the cross product zeroes the diagonals

For projection, if all but one coordinate is fixed for a set of vectors, all those vectors have parallel orthogonal components.

The transpose of the identity reverses the order of vector elements. Is that what transposes do in general?

Each added dimension gives a new degree of freedom; unless there are zeroes somewhere to nullify it, projecting to higher dimensions may be inconsistent, without enough information to produce the right linear combination.

0 is the one constant that can appear in linear transforms because only it always gives the same scalar of 0, just as only the zero vector solves all systems.

58.

We solve the system to find the masses of platinum and silver to be 2400 and 2600, respectively. Clearly Hiero has been cheated.

```
solve(matrix(c(1, .05, 1, .1), nrow = 2)) %% as.matrix(c(5000, 370))
```

```
##      [,1]
## [1,] 2600
## [2,] 2400
```

59.

The Fahrenheit-Celsius matrix is invertible. The matrix is:

$$\begin{bmatrix} -5/9 & 32 \\ 0 & 1 \end{bmatrix}$$

and the inverse Celsius-Fahrenheit matrix is (by the formula):

$$9/5 \begin{bmatrix} 1 & -32 \\ 0 & -5/9 \end{bmatrix}$$

which multiplies by 9/5, adds 32, and puts a placeholder 1 in the second element.

##60. Given this matrix, the product is:

```
matrix(c(1, 8, 1 / 8, 1), nrow = 2) %% as.matrix(c(100, 1600))
```

```
##      [,1]
## [1,] 300
## [2,] 2400
```

The shearing here compresses one dimension into the other, since the off-diagonals are inverses of each other and the diagonals 1. *Off-diagonals of the matrix product contribute interactions between dimensions to the product.* Here each element of b is the sum of the amount's value in *both* currencies.

- b. Because both within-dimension and cross-dimension are exactly the same, , the determinant is 0 and the column vectors are just each others' inverses with order swapped. It will only be consistent for multiples of (1, 8) because that is the ratio of the vectors to each other.

62.

For exchange matrices in general:

- a. Diagonals are 1s: each currency in itself.
- b. $a_{ij} = 1/a_{ji}$. The first is the column currency in the row currency, the second the row currency in the column currency.

The table comes out to:

$$\begin{bmatrix} 1 & 4/5 & 4/25 & 5/4 \\ 5/4 & 1 & 4/25 & 1 \\ 25/4 & 25/4 & 1 & 10 \\ 4/5 & 1 & 1/10 & 1 \end{bmatrix}$$

- c. a_{ik} is i in units, of k , a_{kj} k in units of j , and a_{ij} their product: i in units of k . Each elements may be expressed as a chain of linear combinations this way.
- d. Just 1, since each vector is a linear combination of the others. Each row i is the inverse of column i . Therefore, every vectors is some linear combination of the others and the rank is 1. The RREF would give the simplest linear combination of the uneliminated row in terms of each other currency.

##63. This is as simple as creating a coefficient matrix from the free variables. FOr this example

$$\begin{bmatrix} x_1 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ 0 & 5 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_2 \\ x_5 \end{bmatrix}$$

##64. The bottom two rows eliminate, leaving

$$\begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & 1 & 4 \end{bmatrix}$$

The matrix is

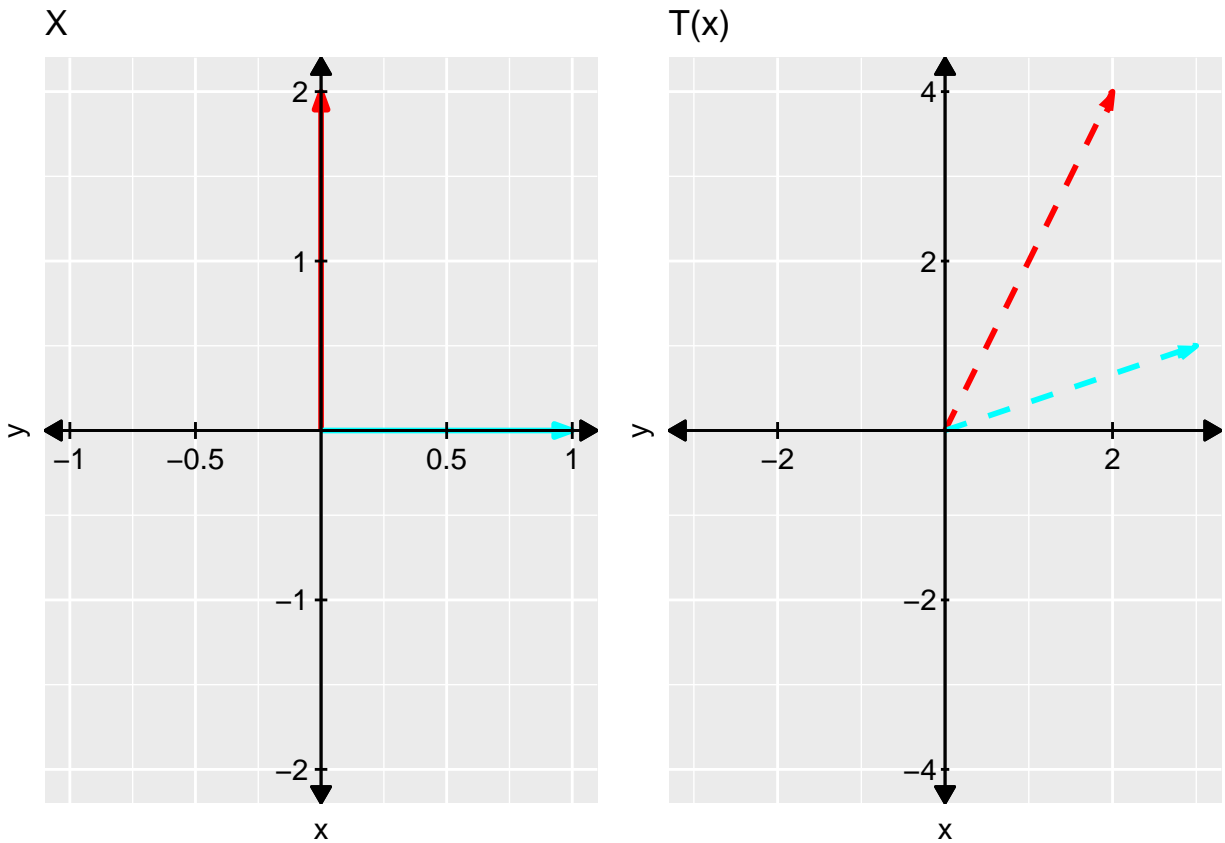
$$\begin{bmatrix} -2 & -3 \\ 0 & -4 \end{bmatrix}$$

Don't forget to flip the signs!

2.2

Using my plotting function, we see both horizontal and vertical shearing.

```
plot_transform(m = matrix(c(0, 2, 1, 0), nrow = 2), trans = matrix(c(3, 1, 1, 2), nrow = 2))
```

2.

A counterclockwise 60-degree rotation is:

```
rotate <- function(theta) {
  matrix(c(cos(theta), sin(theta), -sin(theta), cos(theta)), nrow = 2)
}
```

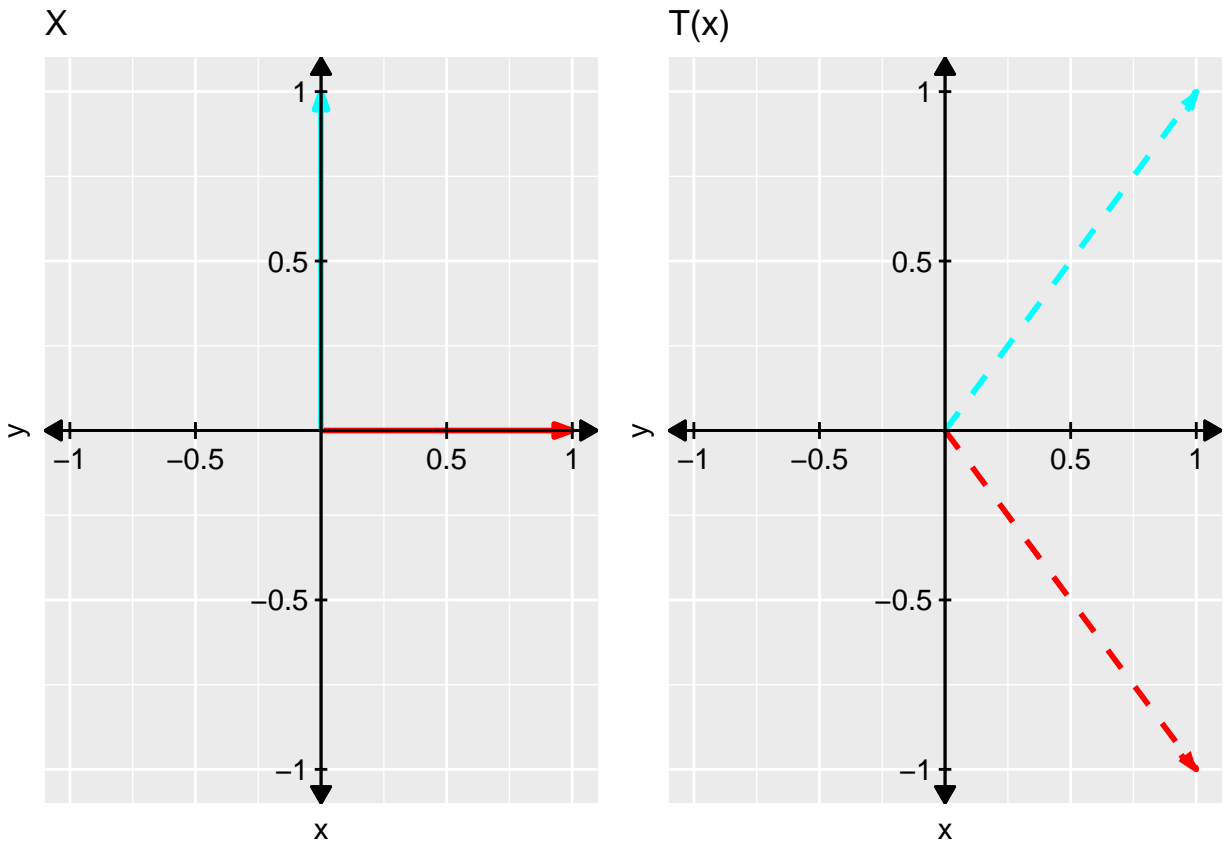
```
rotate(60)
```

```
##           [,1]      [,2]
## [1,] -0.9524130  0.3048106
## [2,] -0.3048106 -0.9524130
```

4. This is a rotation (right, I thin) combine with both horizontal and vertical shearing. So -1 in 1, 2 means rotate right.

```
plot_transform(
  m = matrix(c(1, 0, 0), nrow = 2),
  matrix(c(1, -1, 1, 1), nrow = 2)
)
```

```
## Warning in matrix(c(1, 0, 0), nrow = 2): data
## length [3] is not a sub-multiple or multiple of
## the number of rows [2]
```



5.

$\cos(x) = .8 \sin(x) = .6$, so take the inverse to get in radians: 0.1024164

6,7.

If the line is $\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$ and the line $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, then the orthogonal projection is $\frac{5}{9} \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$. The reflection over this line is the vector minus twice its component orthogonal to the line, which is the vector minus its *parallel* component. This rearranges to twice $x^\perp - x$.

$$refl = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - 2\left(\begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \frac{5}{9} \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}\right)$$

$$refl = 2\left(-\frac{5}{9} \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}\right) - \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

8.

$$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

is a dual shear.

9.

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

is a vertical shear with no scaling.

10. By the formulas, the matrix of the projection onto $\begin{bmatrix} 4 \\ 3 \end{bmatrix}$ is

$$\frac{1}{25} \begin{bmatrix} 16 & 12 \\ 16 & 9 \end{bmatrix}$$

The squares on the diagonals come from dotting x with w twice

11.

The reflection is $2\mathbf{P} - \mathbf{I}$. The unit vector on this line is $\begin{bmatrix} 4/5 \\ 3/5 \end{bmatrix}$, so we compute the projection onto that vector:

$$\begin{bmatrix} .64 & .48 \\ .48 & .36 \end{bmatrix}$$

then double and subtract 1. No need for scaling, these being unit vectors.

$$\begin{bmatrix} .28 & .96 \\ .96 & -.28 \end{bmatrix}$$

13.

A reflection of a unit vector has the matrix:

$$\begin{bmatrix} 2u_1^2 - 1 & 2u_1u_2 \\ 2u_1u_2 & 2u_2^2 - 1 \end{bmatrix}$$

There is no need to scale the orthogonal matrix because this is a unit vector.

13. Generalizing to three dimensions, we find the row denotes the dimension of u , the column the dimension of u by which it must be multiplied u . Thus:

$$\begin{bmatrix} u_1^2 & u_1u_2 & u_1u_3 \\ u_1u_2 & u_2^2 & u_2u_3 \\ u_1u_3 & u_2u_3 & u_3^2 \end{bmatrix}$$

the sum of the diagonal is obviously u 's sum of squares.

14.

The reflection is obvious $2\mathbf{P} - \mathbf{I}$.

16.

The reflection about a line \mathbf{L} defined by the angle θ is just the transformations composed in either order. So we just smush the matrices

$$\begin{bmatrix} (\cos \theta - \sin \theta)(u_1^2 + u_1 u_2) & (\cos \theta - \sin \theta)(u_2^2 + u_1 u_2) \\ (\sin \theta + \cos \theta)(u_1^2 + u_1 u_2) & (\sin \theta + \cos \theta)(u_2^2 + u_1 u_2) \end{bmatrix}$$

A vector for which $\mathbf{A}v = v$ lies on the line of reflection and is a multiple of $[b - 1/b]$

18.

The projection onto the xy plane in \mathbf{R}^3 is given by:

$$\begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & -1/2 \end{bmatrix}$$

We just drop the movement in the z direction, scaling by $1/2$ to ensure unit length.

19.

Projection onto xy plane:

```
tcrossprod(c(0, 1, 0)) + tcrossprod(c(1, 0, 0)) %>%
  matador::mat2latex(sink = FALSE)
```

```
## $$
## \begin{bmatrix}
## 1 & 0 & 0 \\
## 0 & 0 & 0 \\
## 0 & 0 & 0 \end{bmatrix}
## \end{bmatrix}
## $$
```

```
## numeric(0)
```

20.

xz reflection.

```
tcrossprod(c(0, 0, 1)) + tcrossprod(c(1, 0, 0))
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    0    0
## [3,]    0    0    1
```

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

```
A %*% solve(t(A) %*% A) %*% t(A)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    0    0
## [3,]    0    0    1
```

21.

Rotation about z through $\pi/2$. The z coordinate stays constant, so

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

23.

Reflection over xz . 2 ways to solve this:

1. Find an orthogonal basis for the subspace of the reflection. Compute matrices of projections for each vectors, then sum, then convert to a reflection.
2. Obtain the matrix of the projection using the \mathbf{A}^T trick. The cross-products matrix *is* just \mathbf{A} with all directions scaled to unit length.

```
I <- diag(nrow = 3)
```

```
P <- tcrossprod(c(1, 0, 0)) + tcrossprod(c(0, 1 / sqrt(2), 1 / sqrt(2)))
```

```
A <- matrix(c(1, 0, 0, 0, 1 / sqrt(2), 1 / sqrt(2)), nrow = 3)
```

```
P <- A %*% solve(t(A) %*% A) %*% t(A)
```

26.

a.

$$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

b.

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

c.

$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

e.

37.

The trace is:

- a. Projection: $u_1^2 + u_2^2$ squared magnitude
- b. reflection: 0
- c. rotation: $2a$
- d. Shear: 0

38.

Determinants:

- a. Projection: $u_1^2 u_2^2 - u_1 u_2 u_1 u_2 = 0$. Noninvertible, since infinitely many points have the same projection.
- b. Reflection: $a^2 - b^2$; always invertible, since $a^2 = 1 - b^2 \rightarrow a \neq b$
- c. Rotation: $a^2 - b^2$, always invertible for the same reasons.
- d. $0k = 0$, noninvertible.

The inverse of a simple shear matrix is just the matrix with the shear constant's sign reversed, as in

$$\begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & -k & 0 & 1 \end{bmatrix}$$

The “unshear” matrix subtracts the vertical component that the original matrix added to the horizontal component.

42.

The projection of a projection is the same point; the transformation is idempotent. If the point lies on the line of reflection, then:

$$\begin{aligned} &u_1^4 + u_1^2 + u_2^2 \\ &u_1^2(u_1^2 + u_2^2) \\ &u_1^2 \end{aligned}$$

by the unit-length constraint, an identity, and also:

$$\begin{aligned} &u_1^3 u_2 + u_1 u_2^3 \\ &u_1 u_2 (u_1^2 + u_2^2) \\ &u_1 u_2 u_1 u_2 \end{aligned}$$

43.

The inverse of the rotation matrix simply switches the signs of the diagonal. This amounts to rotating clockwise instead of counterclockwise.

44.

The inverse of a rotation combined with a scaling is:

$$\frac{1}{a^2 + b^2} \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

such that the rotation is in the opposite direction and the determinant (the scaling factor) is divided out.

45.

The inverse of a reflection is

$$\frac{1}{-a^2 - b^2} \begin{bmatrix} -a & b \\ b & a \end{bmatrix}$$

Given that $a^2 + b^2 = 1$ (unit length), the inverse is just

$$\begin{bmatrix} a & b \\ b & -a \end{bmatrix}$$

the original matrix.

48.

The next few problems build on the fact that for every linear transformation $\mathbf{R}^2 \rightarrow \mathbf{R}^2$, there exist a pair of unit vectors that remain orthogonal before and after the transformation. For a rotation, these are the standard vectors.

For this transformation, only the standard vectors and their opposite signs qualify.

```
test_orthogonal <-  
  function(t_x, t = seq(0, pi / 2, length.out = 100)) {  
    out <- sapply(t, function(t) {  
      t(t_x %*% c(-sin(t), cos(t))) %*% (t_x %*% c(cos(t), sin(t)))  
    })  
  
    t[which.min(abs(out))]  
  }  
  
test_orthogonal(matrix(c(2, 1, 2, -4), nrow = 2))  
  
## [1] 1.570796
```

```
t(matrix(c(2, 1, 2, -4), nrow = 2) %*% c(0, 1)) %*% (matrix(c(2, 1, 2, -4), nrow = 2) %*% c(-1, 0))

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

50.

```
test_orthogonal(matrix(c(1, 0, 1, 1), nrow = 2))

## [1] 1.015464
```

51.

```
test_orthogonal(matrix(c(2, 1, 1, 2), nrow = 2))

## [1] 0.7933315

test_orthogonal(matrix(c(0, 5, 4, -3)))

## [1] 0
```

Sec. 2.3

17.

```
check_commute(square(1, 0, 0, 2))

##      [,1]      [,2]
## [1,] "a = a"  "2b = b"
## [2,] "c = 2c" "2d = 2d"
```

All diagonal matrices

18.

```
check_commute(square(2, -3, 3, 2))

##      [,1]      [,2]
## [1,] "2a - 3b = 2a + 3c" "3a + 2b = 2b + 3d"
## [2,] "2c - 3d = -3a + 2c" "3c + 2d = -3b + 2d"
```

$b = -c, a = d.$

23.

```
check_commute(square(1, 3, 2, 6))
```

```
##      [,1]      [,2]
## [1,] "a + 3b = a + 2c" "2a + 6b = b + 2d"
## [2,] "c + 3d = 3a + 6c" "2c + 6d = 3b + 6d"
```

This is more complex.

$$b = 2/3c$$

$$d = a + 5/3c$$

19.

All matrices

$$\begin{bmatrix} a & b \\ -c & d \end{bmatrix}$$

commute with

$$\begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix}$$

, plus its inverse.

```
m <- matrix(c(0, 2, -2, 0), nrow = 2)
m %*% matrix(c(-5, 2, -2, -5), nrow = 2)
```

```
##      [,1] [,2]
## [1,]  -4  10
## [2,] -10  -4
```

```
matrix(c(-5, 2, -2, -5), nrow = 2) %*% m
```

```
##      [,1] [,2]
## [1,]  -4  10
## [2,] -10  -4
```

Diagonal matrices commute with diagonal matrices

```
m <- diag(x = 2:4)
```

```
comm <- diag(x = 1, nrow = 3)
```

```
comm[c(1, 3)] <- c(1, 2)
```

```
m %*% comm
```

```
##      [,1] [,2] [,3]
## [1,]    2    0    0
## [2,]    0    3    0
## [3,]    8    0    4
```

```
comm %*% m
```

```
##      [,1] [,2] [,3]
## [1,]    2    0    0
## [2,]    0    3    0
## [3,]    4    0    4
```

27.

Prove matrix distributivity. Easy from the properties of linearity.

$$\begin{aligned} \mathbf{T}(x) &= \mathbf{A} \\ \mathbf{A}(\mathbf{C} + \mathbf{D}) &= \mathbf{AC} + \mathbf{AD} \\ \mathbf{T}(\mathbf{C} + \mathbf{D}) &= \mathbf{T}(\mathbf{C}) + \mathbf{T}(\mathbf{D}) == \mathbf{AC} + \mathbf{AD} \end{aligned}$$

$$\begin{aligned} \mathbf{T}(x) &= \mathbf{A} \\ \mathbf{T}(y) &= \mathbf{B} \\ (\mathbf{T}(x) + \mathbf{T}(y)) &= \mathbf{T}(x + y) \\ \mathbf{T}(x + y)(\mathbf{C}) &= \\ \mathbf{T}(x)\mathbf{C} + \mathbf{T}(y)\mathbf{C} &= \\ \mathbf{AC} + \mathbf{BC} &= \end{aligned}$$

28.

True by the second property of linear transformations.

31.

Row i of \mathbf{AB} is the dot product $\mathbf{A}_i \cdot \mathbf{B}_j$.

31.

Should two matrices have a defined product, row i of \mathbf{AB} is the dot product of \mathbf{A}_i with each \mathbf{B}_j .

32.

$\mathbf{AX} = \mathbf{XA}$ for \mathbf{A}^{-1} , \mathbf{I} , the zero matrix, and probably others.

43.

The 90-degree rotation matrix squares to the identity

$$\mathbf{A}^1001 = \mathbf{A}$$

```
mat_pows(matrix(c(0, 1, 1, 0), nrow = 2), 2:4)
```

```
## $'2'
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
##
## $'3'
##      [,1] [,2]
## [1,]    0    1
## [2,]    1    0
##
## $'4'
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

44.

The rotation $\pi/2$ degrees in either direction.

45.

A rotation matrix, for example

```
mat_pows(matrix(c(0, 1, -1, 0), nrow = 2), 1:4)
```

```
## $'1'
##      [,1] [,2]
## [1,]    0   -1
## [2,]    1    0
##
## $'2'
##      [,1] [,2]
## [1,]   -1    0
## [2,]    0   -1
##
## $'3'
##      [,1] [,2]
## [1,]    0    1
## [2,]   -1    0
##
## $'4'
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

43.

The matrix of the rotation $2\pi/3$ radians counterclockwise.

63.

No left inverse exists

```
A <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 3)
```

```
l_inv <- solve(t(A) %*% A) %*% t(A)
A %*% l_inv
```

```
##           [,1]      [,2]      [,3]
## [1,]  0.8333333 0.3333333 -0.1666667
## [2,]  0.3333333 0.3333333  0.3333333
## [3,] -0.1666667 0.3333333  0.8333333
```

64

Nope!

```
A <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, byrow = TRUE)
l_inv <- try(solve(t(A) %*% A) %*% t(A))
```

```
## Error in solve.default(t(A) %*% A) :
## Lapack routine dgesv: system is exactly singular: U[3,3] = 0
```

45.

For a 120-degree rotation, $\mathbf{A}^3 = \mathbf{I}$.

46.

$\mathbf{A}^n = \mathbf{A}$ for all projection matrices. An example:

```
(matrix(c(9, 12, 12, 16), nrow = 2) * 1 / 25) %^% 2
```

```
##           [,1] [,2]
## [1,]  0.36 0.48
## [2,]  0.48 0.64
```

47.

Answered above.

55.

To zero out a partial-rank matrix you can multiply by a matrix whose rows are the ratios of the rows to each other.

```
matrix(c(1, 2, 2, 4), nrow = 2) %*% matrix(c(-2, 1, -2, 1), nrow = 2)
```

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

58.

If we transpose the first, have to transpose the second -columns now instead of rows.

```
t(matrix(c(-2, 1, -2, 1), nrow = 2)) %*% matrix(c(2, 4, 1, 2), nrow = 2)
```

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

59.

How to “invert” a linearly dependent matrix...you can’t, because if no inverse exists $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ has no solution.

Trying to “find” the inverse \mathbf{X} results in an inconsistent systems:

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

$$x_1 + 2x_2 = 0$$

$$x_1 + 2x_2 = 1$$

Because a dimension is collapsed ($ad = bc$), you can’t distinguish between diagonals and off-diagonals to get the identity.

60.

No such matrix exists, as this matrix has no inverse.

62.

There is no left inverse.

```
A <- matrix(c(1, 2, 3, 0, 1, 2), nrow = 3)
```

```
l_inv <- ((solve(t(A) %*% A)) %*% t(A))
```

```
A %*% l_inv
```

```
##      [,1]      [,2]      [,3]
## [1,] 0.8333333 0.3333333 -0.1666667
## [2,] 0.3333333 0.3333333 0.3333333
## [3,] -0.1666667 0.3333333 0.8333333
```

##. 66.

Any lower triangular matrices where the nonzero elements do not share any row-column indices, e.g.:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ c & d & 0 \end{bmatrix}$$

or

67.

The product

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix} \mathbf{A}$$

for any transition matrix is just

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

, because this matrix sums the columns, and \mathbf{A} is only a transition matrix if all columns sum to 1.

68.

$$e = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

$$\mathbf{T}(\mathbf{A}) = \mathbf{T}(\mathbf{B}) = e$$

$$\mathbf{T}(\mathbf{A}(\mathbf{B})) = \mathbf{T}(\mathbf{A})\mathbf{B} = e\mathbf{B} = e$$

74.

If \mathbf{A} is positive and \mathbf{B} is not, their product in either direction must be positive.

75.

No $\mathbf{A}^m = 0$ and every column of \mathbf{A} contains at least one nonzero entry by the definition of a distribution vector. So $\mathbf{A}_i^m \cdot \mathbf{A}_j \neq 0$, because for the nonzero j $\mathbf{A}_{ij}^m \cdot \mathbf{A}_{ji} \neq 0$

77.

Finding an example distribution vector. Page 1 has the highest rank.

```
A <- matrix(c(0.4, 0, 0.6, 0.2, 0.6, 0.2, 0.7, 0.1, 0.2), nrow = 3)
```

```
(A %>% 20) %>% round(digits = 1)
```

```
##      [,1] [,2] [,3]
## [1,]  0.5  0.5  0.5
## [2,]  0.1  0.1  0.1
## [3,]  0.4  0.4  0.4
```

79.

This is cheating, but the identity is $\mathbf{A}x = x$ for all x . Also any distribution matrix where the nodes have identical distributions.

```
m <- matrix(c(.25, .75, .25, .75), nrow = 2)
m %*% matrix(c(1, 3))
```

```
##      [,1]
## [1,]    1
## [2,]    3
```

```
m %*% c(3, 1)
```

```
##      [,1]
## [1,]    1
## [2,]    3
```

80.

A transition matrix has no limit if it involves column permutations, for example

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

,which just shuffles its columns endlessly.

81.

If $\mathbf{A}v = 5v$, then v is an eigenvector of \mathbf{A} , so the eigenvalue gets squared as \mathbf{A} is raised to higher powers.

$$\begin{aligned}\mathbf{A}^2 v &= 25v \\ \mathbf{A}^m v &= 5^m v\end{aligned}$$

82.

Walkthrough of a special case.

```
A <- matrix(c(.4, .6, .3, .7), nrow = 2)
A %*% as.matrix(1):2
```

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

```
A %*% c(1, -1)
```

```
##      [,1]
## [1,]  0.1
## [2,] -0.1
```

Obviously $\mathbf{A} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 0.1 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

b.

```
x <- c(1, 0)
solve(matrix(c(1, 2, 1, -1), nrow = 2), c(1, 0))
```

```
## [1] 0.3333333 0.6666667
```

```
Ax <- A %*% x
```

To express $\mathbf{A}x$ as a linear combination of these vectors, we find an LC of those vectors that gives x and multiply \mathbf{A} by it.

```
A %*% (matrix(c(1, 2, 1, -1), nrow = 2) %*% c(1 / 3, 2 / 3))
```

```
##      [,1]
## [1,] 0.4
## [2,] 0.6
```

```
A %^% 5 %*% matrix(c(1, 2, 1, -1), nrow = 2) %*% c(1 / 3, 2 / 3)
```

```
##      [,1]
## [1,] 0.33334
## [2,] 0.66666
```

```
A %^% 5 %*% x
```

```
##      [,1]
## [1,] 0.33334
## [2,] 0.66666
```

So we have:

$$\mathbf{A}^m x = \mathbf{A}^m \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix}$$

This just finds an alternate basis for x .

83.

$\mathbf{A}x = x$ for a distribution vector with a zero if and only if, for the zero element x_m , every element of row i . This is so because that component of $\mathbf{A}x$ must be zero, so the sum of each \mathbf{A}_{ij} scaled by x 's nonzero components must be 0 as well, and the term corresponding to x 's zero component must also be 0, since otherwise the matrix's row sums after scaling by x would not sum to 1, meaning $\mathbf{A}x$ would not be a distribution vector. If this is so, all entries of row i of \mathbf{A}^m will remain 0 because $0k = 0$ for all k . So \mathbf{A} cannot be regular if $\mathbf{A}x = x$ for an x with a zero component.

84.

If \mathbf{A} has rank n , then $n < m$, so it has full row rank. That means any n of its column vectors span \mathbf{R}^n , so some linear combination of the columns gives the identity. But because the matrix lacks full column rank, any b has infinitely many solutions, so infinitely many right inverses exist.

85.

One, the inverse.

Sec. 2.4

21.

$f(x) = x^2$ is noninvertible because negative and positive roots have the same square.

22.

2^x is noninvertible because it is undefined for $y \leq 0$.

23.

$f(x) = x^3 + x$ is invertible.

24.

$f(x) = x^3 - x$ is noninvertible; $f(1) = f(0)$.

25.

Invertible; x_1^3 and x_2 both span \mathbf{R} and have unique solutions.

26.

Invertible; both functions are invertible

27.

$x_1 + x_2$ is noninvertible, any $x_1 = -x_2$ returns 0

$x_1(x_2)$ is also noninvertible, since same-signed pairs correspond to the same y .

32.

Only the identity and scalars of it have inverse $\mathbf{A}^{-1} = \mathbf{A}$.

If \mathbf{A} is invertible, then $c\mathbf{A}$ is invertible with matrix $c\mathbf{A}^{-1}$

$$\mathbf{T}(\mathbf{A}) = \mathbf{A}^{-1}$$

$$c(\mathbf{T}(\mathbf{A})) = c(\mathbf{T}(\mathbf{A})) = c\mathbf{A}^{-1}$$

30.

If we have

$$\begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix}$$

none of the variables can be 0.

31.

The values have to satisfy

$$\begin{aligned} c_2a &\neq b \\ -c_1a &\neq c \\ -c_1b &\neq c_2c \end{aligned}$$

all must be nonzero.

32.

$$a^2 + c^2 = a$$

33.

Matrices of the form

$$\begin{bmatrix} a & b \\ b & -a \end{bmatrix}$$

are involutory if $a^2 + b^2 = 1$, without other constraints. All reflections have this property

34.

a. Diagonal matrices are invertible IFF all diagonal values are nonzero. If so, the inverse is:

$$\begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1/c \end{bmatrix}$$

the diagonal values inversed.

Obviously, if a diagonal element is 0, *an entire dimension is lost*, leaving no way to recover the input.

- b. This is true for a diagonal matrix of any size.

35.

The upper triangular matrix is invertible so long as the diagonal is all nonzero, with no other constraints. Because each column vector contains $m - j$ zeroes, none can be linearly dependent no matter what values are chosen.

- c. The inverse of an upper triangular matrix is also upper triangular, as follows from the process of row elimination.
d. The same rules apply to lower triangular matrices.

36.

If a diagonal element of a triangular matrix is 0, then the RREF inevitably produces a zero row because 0 is not a multiple of any nonzero scalar. so the system is singular.

37.

Scalars of invertible matrices remain invertible.

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

$$(1/5)\mathbf{A}^{-1}(5\mathbf{A}) = \mathbf{I}$$

The scalar's inverse is the inverse of the scalar, naturally.

38.

Inverse of a shear

$$\begin{bmatrix} 1 & k \\ 0 & -1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -k \\ 0 & -1 \end{bmatrix}$$

39.

Matrices of this form are invertible. The inverse just has the opposite sign of the shear term.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 0 & 1 \end{bmatrix}$$

The opposite sign of the shear value would be scaled by the inverse of the value of the corresponding diagonal.

40.

If \mathbf{A} has two equal columns, then \mathbf{A} does not form a basis because each column has infinitely many linear relations in terms of the others. Therefore, each y in its range, even if spans the target, lacks a unique representation and so fails to be invertible.

41.

Inverses of:

a. Rotation:

$$\begin{bmatrix} \cos \theta & s \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

Counterclockwise subtracts y from x , clockwise x from y .

b. Refection: Itself:

c. Projection: noninvertible, as $u_1^2 u_2^2 = (u_1 u_2)^2$

d. Scaling: invertible, as always diagonal with nonzero values.s

42.

Permutation matrices are invertible. If \mathbf{P} has a 1 in ij , $i \neq j$, then \mathbf{P}^1 has a 1 in ji . In less dumb words, the inverse is the transpose and the transpose the inverse because the columns are orthogonal:

```
solve(matrix(c(0, 0, 1, 1, 0, 0, 0, 1, 0), byrow = TRUE, nrow = 3))
```

```
##      [,1] [,2] [,3]
## [1,]    0    1    0
## [2,]    0    0    1
## [3,]    1    0    0
```

43.

Given \mathbf{A} and \mathbf{B} are invertible:

$$\begin{aligned} y &= \mathbf{A}(\mathbf{B}x) \\ \mathbf{A}^{-1}y &= \mathbf{B}x \\ x &= \mathbf{B}^{-1}\mathbf{A}^{-1}y \end{aligned}$$

44. The inverse of the rotation-scaling

$$\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$$

is

$$\frac{1}{a^2 + b^2} \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

like that for a regular rotation, except we have to reverse the scaling factor.

45.

The reflection inverse is trickier. But we know $-a^2 - b^2 = -1$, so:

$$\begin{bmatrix} a & b \\ b & -a \end{bmatrix} = \frac{1}{-a^2 - b^2} \begin{bmatrix} -a & -b \\ -b & a \end{bmatrix} \begin{bmatrix} a & b \\ b & -a \end{bmatrix}$$

As we'd expect geometrically, the reflection is its own inverse.

28.

Scalar associativity:

$$\begin{aligned}\mathbf{T}(x) &= \mathbf{A} \\ k(\mathbf{AB}) \\ k\mathbf{T}(b) &= \mathbf{T}(k\mathbf{B}) = \mathbf{A}(k\mathbf{B})\end{aligned}$$

29.

Consider a rotation followed by a rotation. Geometrically, this is obviously a rotation. Its matrix is

$$\begin{bmatrix} \cos \alpha \cos \beta - \sin \alpha \sin \beta & -\cos \beta \sin \alpha - \cos \alpha \sin \beta \\ \cos \alpha \sin \beta + \cos \beta \sin \alpha & \cos \beta - \sin \alpha \sin \beta \end{bmatrix}$$

with the typical matrix form of a rotation. Using trig identities, this is

$$\begin{bmatrix} \cos \alpha + \beta & -\sin \alpha + \beta \\ \sin \alpha + \beta & \cos \alpha - \beta \end{bmatrix}$$

So we rotate by the sum of the angles.

44.

Matrices filled by column sequence seem to be uninvertible past the 2×2

```
map(2:12, ~ matrix(seq(1, .x^2), nrow = .x)) %>%
  map(~ try(solve(.x)))

## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[5,5] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[5,5] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[4,4] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[6,6] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[8,8] = 0
## Error in solve.default(.x) :
##   Lapack routine dgesv: system is exactly singular: U[5,5] = 0
```

```

## [[1]]
##      [,1] [,2]
## [1,]   -2  1.5
## [2,]    1 -0.5
##
## [[2]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[3,3] = 0\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[3,3] = 0>
##
## [[3]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[3,3] = 0\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[3,3] = 0>
##
## [[4]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[3,3] = 0\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[3,3] = 0>
##
## [[5]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[5,5] = 0\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[5,5] = 0>
##
## [[6]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[5,5] = 0\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[5,5] = 0>
##
## [[7]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[4,4] = 0\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[4,4] = 0>
##
## [[8]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[3,3] = 0\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[3,3] = 0>
##

```

```
## [[9]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[6,6] = 0\n"
## attr("class")
## [1] "try-error"
## attr("condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[6,6] = 0>
##
## [[10]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[8,8] = 0\n"
## attr("class")
## [1] "try-error"
## attr("condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[8,8] = 0>
##
## [[11]]
## [1] "Error in solve.default(.x) : \n  Lapack routine dgesv: system is exactly singular: U[5,5] = 0\n"
## attr("class")
## [1] "try-error"
## attr("condition")
## <simpleError in solve.default(.x): Lapack routine dgesv: system is exactly singular: U[5,5] = 0>
```

47.

3^x has a unique value for every x but is noninvertible because undefined for $\infty, 0$.

48.

$$\mathbf{B} = \mathbf{A}^{-1}$$

$$\mathbf{BA} = \mathbf{I}_n$$

so \mathbf{B} has n rows. If it has m columns, \mathbf{I} has m rows. But.

$$\mathbf{AB} = \mathbf{I}_m$$

so \mathbf{A} has m rows and n columns so multiplication with \mathbf{B} is defined. \mathbf{B} has m columns, so $m = n$.

49.

$$\mathbf{I}_n - \mathbf{A}x = b$$

$$x = (\mathbf{I}_n - \mathbf{A})^{-1}$$

For the Israeli economy, the tech matrix and its inverse. This ignores the consumer demand vector, which has to be added to $\mathbf{A}x$.

```
A <- matrix(c(
  .293, .0, 0,
  .014, .207, .017,
  .044, .01, .216
), byrow = TRUE, nrow = 3)
I <- diag(nrow = 3)
b <- c(13.3, 17.6, 1.8)
A <- I - A
```

```

A_inv <- solve(A)

mat2latex(A_inv)

## $$
## \begin{bmatrix}
## 1.41442715700141 & 0 & 0 \\
## 0.0266800055164166 & 1.26137895749603 & 0.0273513294355007 \\
## 0.0797214221469724 & 0.0160890173150004 & 1.27585907307953
## \end{bmatrix}
## $$

```

- b. If only one industry requires output, multiplying by the demand vector just selects that industry's column, giving the demands j makes of each industry i to ensure $\mathbf{A}x = x$. just selects the inverse's first column. Subtracting \mathbf{I} removes the demand each industry puts on itself (always 1 per unit), leaving only what must be produced to satisfy other industries.

```

A_inv %*% c(2, 0, 0)

##           [,1]
## [1,] 2.82885431
## [2,] 0.05336001
## [3,] 0.15944284

```

- c. The matrix represents the demands the industries require to produce as much as they consume; it is the amount “left over” after multiplying by some b . Industries always require some of their own input.
- d. Since the required demands are now 1.4144272, 1.288059, 0.0958104, the manufacturing sector must make greater demands to satisfy its own needs in addition to consumer demand.
- e.

The columns represent industry j 's demands of the other industries in order to work together with the other industries to generate enough output for a given input.

In general, subtracting \mathbf{I} from \mathbf{A} removes the values of the input vector from computation - a useful property!

67.

False: $(\mathbf{A} + \mathbf{B})^2 \neq \mathbf{A}^2 + 2\mathbf{AB} + \mathbf{B}^2$.

```

A <- matrix(sample(1:100, 4), nrow = 2)
B <- matrix(sample(1:100, 4), nrow = 2)

```

```

(A + B) %^% 2

```

```

##           [,1] [,2]
## [1,] 32281 13200
## [2,] 32725 18256

```

```

(A %^% 2) + (2 * (A %*% B)) + (B %^% 2)

```

```

##           [,1] [,2]
## [1,] 33939 10590
## [2,] 37434 16598

```


51.

If \mathbf{A} lacks full row rank, at least one row is redundant. That row, i , will zero out in elimination. The system holds only if b_i is the same linear combination of b 's elements as that row is of \mathbf{A} 's rows, i.e. $\mathbf{E}b_i \neq b_i$ of the row-reduced b .

So $\mathbf{E}x = c$ is inconsistent for some c . We have shown elsewhere that $\mathbf{E}^{-1}\mathbf{E} = \mathbf{A}$, so \mathbf{A} must be inconsistent for some vectors as well.

Something not right here. ## 52.

Not hard to find an inconsistent vector.

```
A <- matrix(c(0, 0, 0, 1, 1, 2, 3, 4, 2, 4, 6, 8), nrow = 4)
```

```
try(gen_inverse(A, c(3, 6, 9, 13)))
```

```
## Error in solve.default(t(A) %*% A) :
```

```
## Lapack routine dgesv: system is exactly singular: U[3,3] = 0
```

53.

```
A <- square(3, 3, 1, 5)
```

```
lambda <- 2
```

```
A2 <- A - diag(nrow = 2) * lambda
```

I choose a vector in \mathbf{A} 's kernel, $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$.

Is how that $\mathbf{A}x = \lambda x$ - this linear combination equals the value that must be subtracted from the original matrix to cause linear dependence. So the eigenvalue can be seen as a scalar of the identity that if added to \mathbf{A} would make it linearly independent.

We could also use the determinant and solve

$$(3 - \lambda)(5 - \lambda) = 3$$

```
x <- c(1, -1)
```

```
A %*% x
```

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

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

```
## [2,]   -2
```

```
diag(nrow = 2) %*% x * lambda
```

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

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

```
## [2,]   -2
```

54.

$$\lambda = 6.$$

We are asked to interpret some inverses.

57.

Reflection over the line that forms the angle θ .

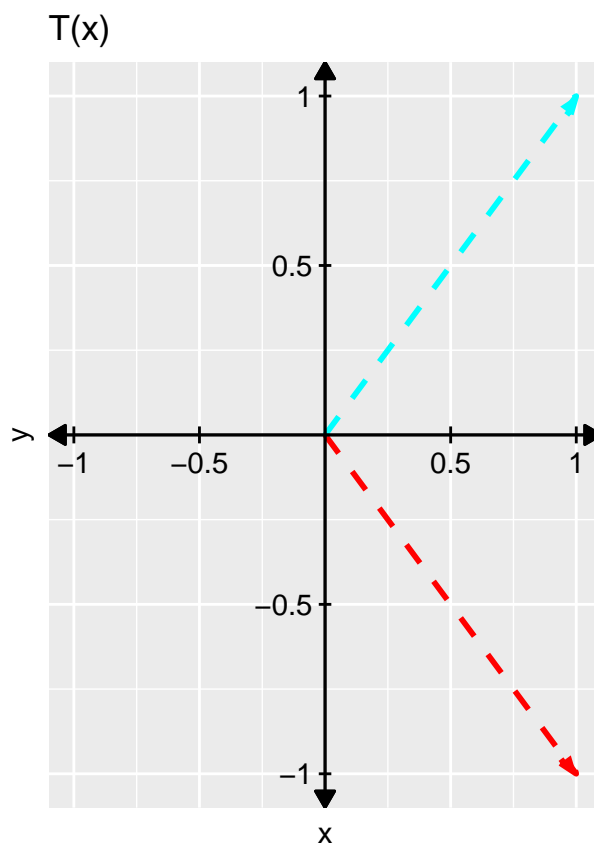
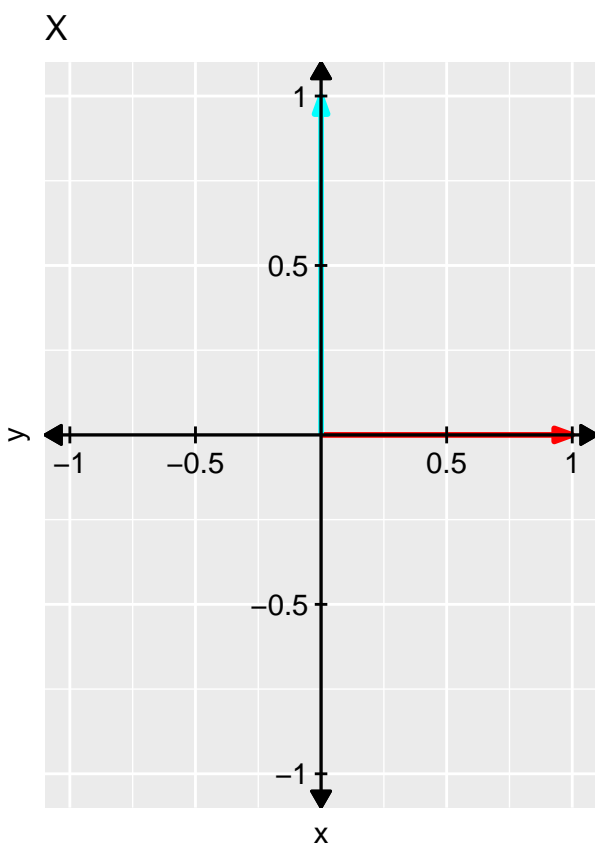
60.

Reflection over the line of -0.9272952 .

61.

Clockwise rotation 45 degrees.

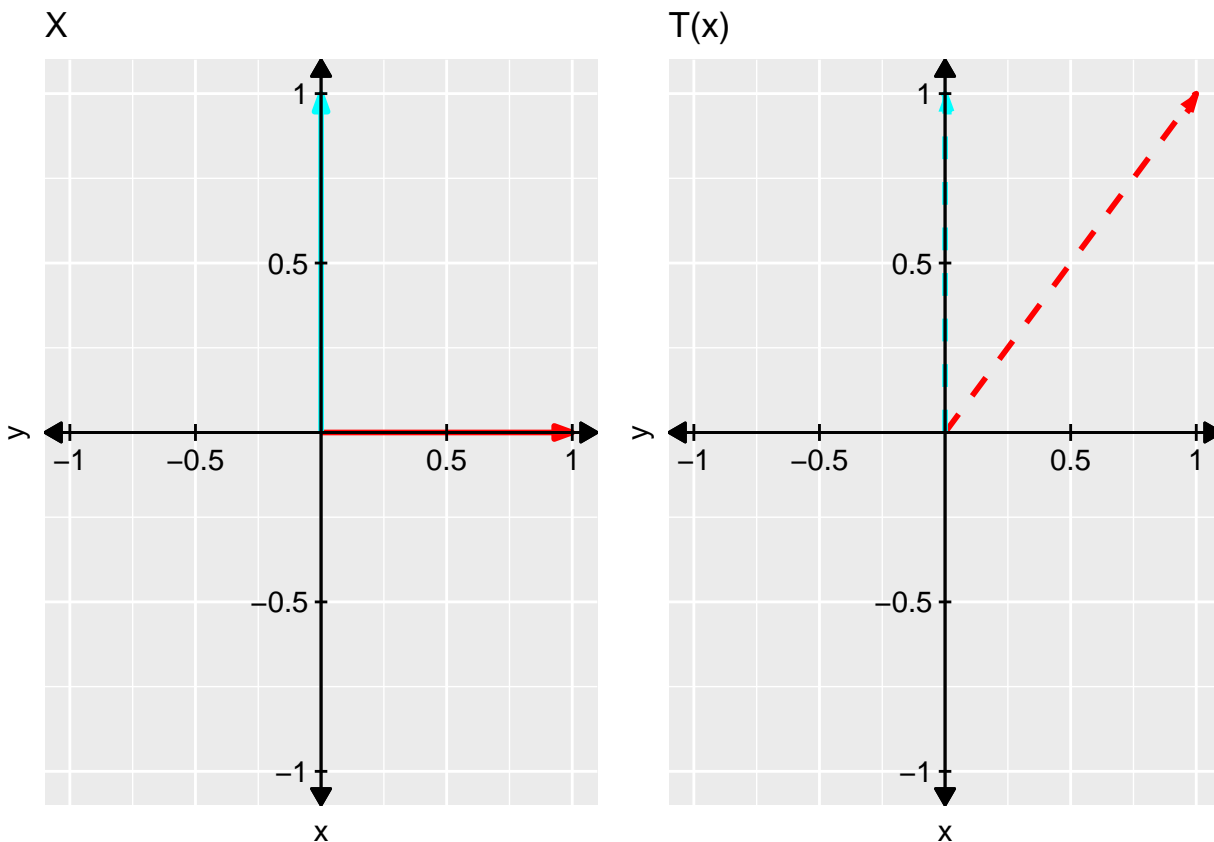
```
plot_transform(trans = square(1, -1, 1, 1))
```



65.

Horizontal shear.

```
plot_transform(trans = square(1, 1, 0, 1))
```



66.

If two matrices are invertible, so is the product.

$$(\mathbf{AB})(\mathbf{BA})^{-1} = \mathbf{I}$$

$$\mathbf{AB} = \mathbf{BA}$$

The equation only holds if both \mathbf{A} and \mathbf{B} are invertible.

68.

$$(\mathbf{A} + \mathbf{B} = \mathbf{C})$$

$$(\mathbf{A} + \mathbf{B})^2$$

$$\mathbf{CC}$$

$$\mathbf{C}(\mathbf{A} + \mathbf{B})$$

$$\mathbf{CA} + \mathbf{CB}$$

$$(\mathbf{A} + \mathbf{B})\mathbf{A} + (\mathbf{A} + \mathbf{B})\mathbf{B}$$

False for the same reasons

69.

False. An invertible sum of matrices does not mean the components are invertible.

70.

$$\begin{aligned}(\mathbf{A}^2)^{-1} &= (\mathbf{A}^{-1})^2 \\ (\mathbf{A}\mathbf{A})^{-1} &= 5\mathbf{A}^{-1}\mathbf{A}^{-1} \\ \mathbf{A}^{-1}\mathbf{A}^{-1} &= \mathbf{A}^{-1}\mathbf{A}^{-1}\end{aligned}$$

72.

$$\begin{aligned}\mathbf{A}\mathbf{B}\mathbf{A}^{-1} &= \mathbf{B} \\ \mathbf{B}\mathbf{A}^{-1} &\neq \mathbf{A}^{-1}\mathbf{B}\end{aligned}$$

73.

True.

$$\begin{aligned}(\mathbf{A}\mathbf{B}\mathbf{A}^{-1})^3 &= \mathbf{A}\mathbf{B}^3\mathbf{A}^{-1} \\ \mathbf{A}\mathbf{B}\mathbf{A}\mathbf{B}\mathbf{A}\mathbf{B}\mathbf{A}^{-1} &= \mathbf{A}\mathbf{B}^3\mathbf{A}^{-1} \\ \mathbf{A}\mathbf{B}^3 &= \mathbf{A}\mathbf{B}^3\end{aligned}$$

74.

I'm satisfied this works.

```
I <- diag(nrow = 2)

A <- matrix(sample(-100:100, 4), nrow = 2)

(I + A) %*% (I + solve(A))

##           [,1]      [,2]
## [1,] -24.00209 -68.01182
## [2,] -80.01391  14.00452

2 * I + A + solve(A)

##           [,1]      [,2]
## [1,] -24.00209 -68.01182
## [2,] -80.01391  14.00452
```

For any invertible \mathbf{S} in \mathbf{R}^m , there exists a unique transformation to \mathbf{R}^n that may be expressed:

$$\mathbf{B} = [\mathbf{T}(v_1) \quad \mathbf{T}(v_2) \quad \dots \quad \mathbf{T}(v_m)]$$

Which is just the definition of any linear transformation using \mathbf{S} as a basis instead of \mathbf{I} .

75.

True if both **A** and **B** are invertible.

78.

We invert the matrix of inputs to get the underlying transformation from **I**, then multiply the matrix of the outputs by this inverse to “peel off” the intermediate transformation and reveal **A**.

```
A_inv <- solve(matrix(c(1, 2, 2, 5), nrow = 2))

res <- matrix(c(7, 5, 3, 1, 2, 3), nrow = 3) %*% A_inv

res %*% solve(A_inv)

##      [,1] [,2]
## [1,]    7    1
## [2,]    5    2
## [3,]    3    3
```

79.

```
A_inv <- solve(matrix(c(3, 1, 1, 2), nrow = 2))

res <- matrix(c(6, 2, 3, 6), nrow = 2) %*% A_inv

res %*% solve(A_inv)

##      [,1] [,2]
## [1,]    6    3
## [2,]    2    6
```

81.

Find matrices of a geometric rotation and reflection.

```
A_inv <- solve(matrix(c(1, -1, -1, 1, 1, 1, -1, -1, 1), nrow = 3))

res <- matrix(c(-1, -1, 1, 1, 1, 1, 1, -1, -1), nrow = 3) %*% A_inv

res %*% solve(A_inv)

##      [,1] [,2] [,3]
## [1,]   -1    1    1
## [2,]   -1    1   -1
## [3,]    1    1   -1
```

```
res <- matrix(c(-1, 1, -1, 1, 1, 1, -1, -1, 1), nrow = 3) %*% A_inv
res %*% solve(A_inv)

##      [,1] [,2] [,3]
## [1,]  -1    1  -1
## [2,]   1    1  -1
## [3,]  -1    1   1
```

82.

This one asks us to try elimination matrices.

a.

$$\mathbf{EA} = \begin{bmatrix} a & b & c \\ -3a + d & -3b + e & -3c + f \\ g & h & k \end{bmatrix}$$

b.

$$\mathbf{EA} = \begin{bmatrix} a & b & c \\ d/4 & e/4 & f/4 \\ g & h & k \end{bmatrix}$$

c. The permutation to swap the bottom two rows is:

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

We have the three elementary matrices: addition, scaling, and permuting.

83.

Yes. The inverse reverses the transformation, so it is the opposite sign of the row subtraction, the inverse of the scalar for scaling, and the transpose of the permutation for permutations.

84.

It should be obvious that elementary matrices exist for all matrices. We can compose them all into a single matrix. For example:

```
E <- compose_trans(list(
  matrix(c(1, -3, 0, 1), byrow = TRUE, nrow = 2),
  matrix(c(1, 0, 0, 1 / 2), nrow = 2, byrow = TRUE),
  matrix(c(0, 1, 1, 0), nrow = 2)
))

A <- matrix(c(0, 1, 2, 3), nrow = 2)

E %*% A
```

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

The inverse of the elimination matrix is the original matrix. This works for any invertible matrix because:

$$\begin{aligned}\mathbf{EA} &= \mathbf{I} \\ \mathbf{A} &= \mathbf{E}^{-1}\mathbf{I} \\ \mathbf{A} &= \mathbf{E}^{-1}\end{aligned}$$

so since \mathbf{E} exactly reverses the transformation \mathbf{A} applied to \mathbf{I} , its inverse *is* \mathbf{A} .

```
solve(E)
```

```
##      [,1] [,2]
## [1,]    0    2
## [2,]    1    3
```

88.

Say we have an invertible \mathbf{A} and noninvertible \mathbf{B} . Applying \mathbf{A} 's elimination matrix to \mathbf{AB} reverts it to \mathbf{B}

$$\begin{aligned}\mathbf{EA} &= \mathbf{I} \\ (\mathbf{EA})\mathbf{B} &= \mathbf{IB} \\ \mathbf{E}(\mathbf{AB}) &= \mathbf{B}\end{aligned}$$

If we apply it to \mathbf{I} , we get the square of the elimination applied to \mathbf{A} :

$$\begin{aligned}\mathbf{EA} &= \mathbf{I} \\ \mathbf{A} &= \mathbf{E}^{-1}\mathbf{I} \\ \mathbf{E}^2\mathbf{A} &= \mathbf{EI}\end{aligned}$$

89.

The product of two lower-triangular matrices is also lower triangular because only the elements in the lower triangle interact in multiplication, meaning the upper triangle remains zeroes.

90.

Time for \mathbf{LU} factorization. We factor out the matrix. Note that the lower triangle of the *elimination* matrix is multiples of the rows *after each successive elimination*, while \mathbf{L} is multiples *of the original rows*, since it works in one pass.

```
A <- matrix(c(1, 2, 2, 2, 6, 2, 3, 7, 4), nrow = 3)
A_l <- mat2latex(A, sink = TRUE)

I <- diag(nrow = 3)
E <- replace(I, lower.tri(I), c(-2, -4, 1))
```

```
L <- replace(I, lower.tri(I), c(2, 2, -1))
```

```
U <- E %*% A
```

```
D <- diag(x = diag(U))
```

```
U_scaled <- sweep(U, 1, diag(U), '/')
```

```
L %*% U
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    2    6    7
## [3,]    2    2    4
```

```
L %*% D %*% U_scaled
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    2    6    7
## [3,]    2    2    4
```

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 6 & 7 \\ 2 & 2 & 4 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}$$

92.

The matrix

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

can't be factorized because the diagonal is zeroes; the on-dimension elements would zero out in the multiplication

True or False?

2.

True, inverses commute.

6.

True.

$$\begin{aligned}(\mathbf{A}^2)^{-1} &= (\mathbf{A}^{-1})^2 \\ (\mathbf{A}\mathbf{A})^{-1} &= \mathbf{A}^{-1}\mathbf{A}^{-1} \\ \mathbf{A}^{-1}\mathbf{A}^{-1} &= \mathbf{A}^{-1}\mathbf{A}^{-1}\end{aligned}$$

99.

$$\begin{aligned}\mathbf{A}^2 &= \mathbf{A} \\ \mathbf{A}^2\mathbf{A}^{-1} &= \mathbf{A}\mathbf{A}^{-1} \\ \mathbf{A} &= \mathbf{I}\end{aligned}$$

The identity is the only invertible matrix such that $\mathbf{A}^2 = \mathbf{A}$.

101.

The entries of \mathbf{A} are less than or equal to s , the column sums of \mathbf{B} less than or equal to r . Each entry of \mathbf{AB} is the dot product of a row of \mathbf{A} and a column of \mathbf{B} . Assuming both quantities are maximal, the s of the row factor out, so.

$$\begin{aligned}& s\mathbf{B}_{j1} + s\mathbf{B}_{j2} + \dots + s\mathbf{B}_{jn} \\ & s(\mathbf{B}_{j1} + \mathbf{B}_{j2} + \dots + \mathbf{B}_{jn}) \\ & sr\end{aligned}$$

102.

- a. If all column sums of \mathbf{A} are less than 1 and all entries positive or nonzero, then all entries are also less than 1. So scaling any column by any element reduces the column sum further.

Assume entry ii of \mathbf{A} is r , such that all other entries of column i are 0. Then $\mathbf{A}_{ii}^2 = r^2$, etc. More generally, this dot product is the sum of squares, which is at most r^2 :

$$\begin{aligned}\mathbf{A}_{kk} &= \mathbf{A}_1\mathbf{A}_2 + \dots + \mathbf{A}_n\mathbf{A}_n \\ &= \sum_{i=1}^n \mathbf{A}_i^2 \\ &= r^2\end{aligned}$$

- c. The limit is 0, so the result can be expressed $\mathbf{I} - \mathbf{A}$
- d.

$$\begin{aligned}(\mathbf{I}_n - \mathbf{A})(\mathbf{I}_n + \mathbf{A} + \mathbf{A}^2 + \dots + \mathbf{A}^m) \\ \mathbf{I} - \mathbf{A} + \mathbf{A} - \mathbf{A}^2 + \mathbf{A}^2 - \mathbf{A}^3 + \dots + \mathbf{A}^m - \mathbf{A}^{m+1} \\ \mathbf{I} + \mathbf{A}^m - \mathbf{A}^{m+1}\end{aligned}$$

So in other words,

104.

From the given information, here are the matrix for the eye's color conversion and the sunglasses.

The eye performs the inverse of the M transformation, converting intensities back to colors.

a.

```
M <- matrix(c(
  rep(1 / 3, 3),
  1, -1, 0,
  1, -1 / 2, -1 / 2
), nrow = 3, byrow = TRUE)

A <- diag(x = c(1, 1, 0), nrow = 3)

C <- M %*% A

M

##           [,1]      [,2]      [,3]
## [1,] 0.3333333 0.3333333 0.3333333
## [2,] 1.0000000 -1.0000000 0.0000000
## [3,] 1.0000000 -0.5000000 -0.5000000

A

##           [,1] [,2] [,3]
## [1,]      1    0    0
## [2,]      0    1    0
## [3,]      0    0    0

C

##           [,1]      [,2] [,3]
## [1,] 0.3333333 0.3333333  0
## [2,] 1.0000000 -1.0000000  0
## [3,] 1.0000000 -0.5000000  0

solve(M)

##           [,1] [,2]      [,3]
## [1,]      1    0 0.6666667
## [2,]      1   -1 0.6666667
## [3,]      1    1 -1.3333333
```

a.

True or False?

1.

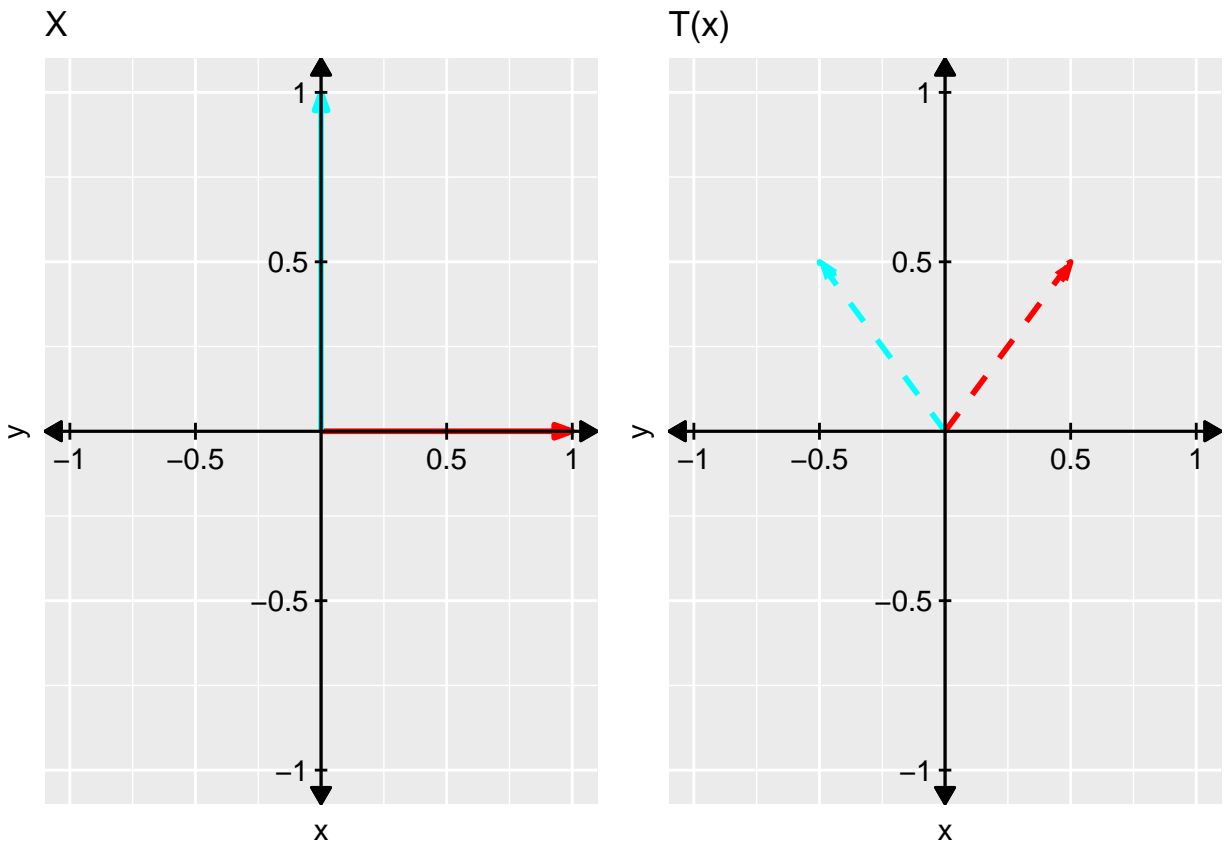
2.

True; invertible matrices commute with their inverses.

4.

True; it's a rotation combined with a scaling.

```
matador::plot_transform(trans = matador::square(.5, .5, -.5, .5))
```



7.

False, few matrices commute

8.

True, $\mathbf{AB} = \mathbf{I}_n$ means inverse.

9.

False; it will be 3×5 .

11.

False; it is non-invertible if k is 5 or 1.

$$k^2 - 6k = -10$$

$$k^2 - 6k + 10 = 0$$

$$(k - 5)(k - 1) = 0$$

12.

$$\begin{bmatrix} k-1 & -2 \\ -4 & k-3 \end{bmatrix}$$

Noninvertible if:

$$k^2 - 4k + 5 = 0$$

13.

True for some ugly polynomial roots.

14.

False,

$$\begin{bmatrix} 1 & 1/2 & 0 & 1/2 \end{bmatrix}$$

never becomes regular because the 2,1 dot product is always 0.

16.

True; we can use inverses.

```
B <- matrix(1:4, nrow = 2, byrow = TRUE)
```

```
C <- matrix(5:8, nrow = 2)
```

```
D <- matrix(rep(1, 4), nrow = 2)
```

```
A <- solve(B) %*% D %*% solve(C)
```

```
B %*% A %*% C
```

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

18.

True

21.

False; invertible matrices never, ever have identical rows.

22.

True, $\mathbf{A}^2 = \mathbf{I}$ means invertibility.

23.

False.

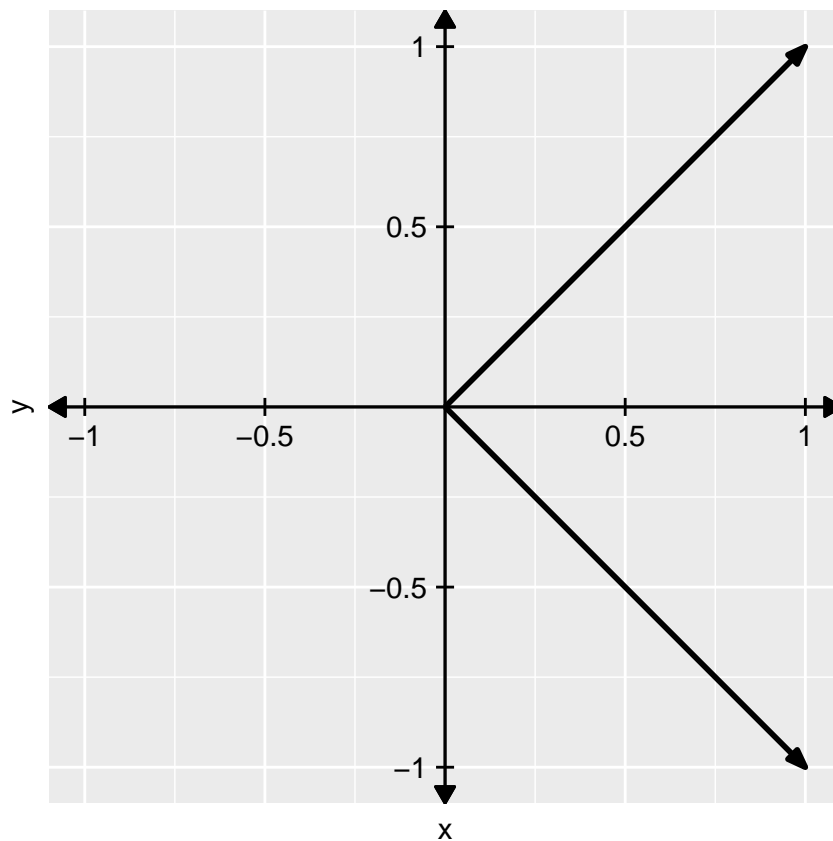
24.

False.

25.

False.

```
plot_mat(square(1, 1, 1, -1), fix_coords = TRUE)
```



True.

$$\det(2\mathbf{A}) = 2ad - 2bc = 2(ad - bc)$$

but

$$\det(\mathbf{A}) = 0$$

$$\det(2\mathbf{A}) = 2(ad - bc) = 2(0) = 0$$

27.

True; it is a diagonal matrix with a diagonal of $ad - bc$.

38.

If \mathbf{A} and \mathbf{B} commute, then:

$$\mathbf{A}^2\mathbf{N} = \mathbf{B}\mathbf{A}^2$$

$$\mathbf{AAB} = \mathbf{BAA} \quad \mathbf{ABA} = \mathbf{Aba}$$

28.

True for an upper triangular matrix with a zero diagonal:

```
square(0, 0, 1, 0) %% 2
```

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

29.

True;

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}^4 = \mathbf{I}$$

because it is a 90-degree rotation. Were the rotation not a multiple of 2π this would not be so.

30.

False. No matrix has inverse

$$\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

because that would require *this* matrix to have an inverse, which it cannot.

31.

False. If a transition matrix is regular, its zeroes eventually become nonzero, so at the very least $\mathbf{A}^2 \neq \mathbf{A}$.

25.

If $\mathbf{A}^{17} = \mathbf{I}_2$, this matrix could be the rotation about $2\pi/17$ radians.

33.

True; if a matrix is invertible, so are all its submatrices.

34.

True

$$\mathbf{A}^2(\mathbf{A}^2)^{-1} = \mathbf{I} \mathbf{A} \mathbf{A}^{-1} \mathbf{A}^{-1} = \mathbf{I} \mathbf{I} = \mathbf{I}$$

35.

False. If $\mathbf{A}^{17} = \mathbf{I}$, \mathbf{A} could be a rotation $2\pi/17$ radians

```
theta <- (2 * pi) / 17
square(cos(theta), sin(theta), -sin(theta), cos(theta)) %^% 17

##           [,1]           [,2]
## [1,]  1.000000e+00 3.885781e-16
## [2,] -3.885781e-16 1.000000e+00
```

36.

False for the same reasons; it could be any involutory matrix.

Matrix commutativity doesn't chain

$\mathbf{AB} = \mathbf{BA}$
 $\mathbf{BC} = \mathbf{CB}$
 $\mathbf{BCA} = \mathbf{CBA} = \mathbf{CAB}$
 $\mathbf{ABC} = \mathbf{ACB} = \mathbf{BAC}$

```
A <- matrix(c(
  1, 2, -1, 0,
  0, 1, 1, 5,
  3, -4, 2, -2,
  1, 0, 3, -1,
  1, -1, 0, 0
), nrow = 5, byrow = TRUE)

b <- matrix(c(2, 3, -2, 0), nrow = 4)

A %*% b

##           [,1]
## [1,]      10
## [2,]       1
## [3,]     -10
## [4,]      -4
## [5,]      -1
```

36.

False. If $\mathbf{A}^2 = \mathbf{I}_2$, it could be a reflection, not just \mathbf{I} or $-\mathbf{I}$.

39.

True.

$\mathbf{AB} = \mathbf{BA}$
 $\mathbf{AAB} = \mathbf{ABA}$
 $\mathbf{AAB} = \mathbf{BAA}$
 $\mathbf{A}^2 = \mathbf{BA}^2$

40.

False; involutory matrices beyond the identity exist.

41.

True; once it becomes positive, a transition matrix never loses that property because a zero dot product cannot occur when all values are positive.

42.

False; an invertible transition matrix would most likely have negative entries for its inverse, making it not a transition matrix.

43.

False; the sum of invertible matrices is not necessarily invertible

45.

True; all reflections are their own inverses:

$$\frac{1}{-a^2 - b^2} \begin{bmatrix} -a & -b \\ -b & a \end{bmatrix} \\ -1 \begin{bmatrix} -a & -b \\ -b & a \end{bmatrix} \begin{bmatrix} 1 & b \\ 1 & -b \end{bmatrix}$$

46.

True, I think...

$$(\mathbf{A}v) \cdot (\mathbf{A}w) = v \cdot w$$

Let $\mathbf{A}v = c$ and $\mathbf{A}w = b$

$$\begin{aligned} c \cdot b &= v \cdot w \\ \mathbf{A}v &= c \\ v &= \mathbf{A}^{-1}c \\ \mathbf{A}\mathbf{A}^{-1}c \cdot \mathbf{A}\mathbf{A}^{-1}b &= v \cdot w \\ v \cdot w &= v \cdot w \end{aligned}$$

47, 48.

These questions are about left and right inverses, which exist for some matrices.

49.

True if \mathbf{A} is the opposite sign of its inverse?

$$\begin{aligned}\mathbf{A}^2 + 3\mathbf{A} + 4\mathbf{I} &= 0 \\ \mathbf{A}\mathbf{A} + 3\mathbf{A} &= -4\mathbf{I} \\ \mathbf{A}(\mathbf{A} + 3\mathbf{A}) &= -4\mathbf{I} \\ \mathbf{A} + 3\mathbf{A} &= \mathbf{A}^{-1}4\mathbf{I} \\ 4\mathbf{A} &= -4\mathbf{A}^{-1} \\ \mathbf{A} &= -\mathbf{A}^{-1}\end{aligned}$$

50.

False. \mathbf{A} could be

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

52.

True. \mathbf{T} is a linear transformation, and so is the cross product (since it has a matrix form). Let \mathbf{A} be that matrix. Therefore:

$$\mathbf{T}(\mathbf{A}) = \mathbf{A}$$

53.

False. An invertible matrix of only one digit must have at least $n - 1$ zeroes to prevent linear dependence, since otherwise a row or column must be repeated. 8 zeroes in a 10×10 is not enough.

```
try(solve(matrix(c(rep(1, 92), rep(0, 8)), nrow = 10, byrow = TRUE)))
```

```
## Error in solve.default(matrix(c(rep(1, 92), rep(0, 8)), nrow = 10, byrow = TRUE)) :  
## Lapack routine dgesv: system is exactly singular: U[2,2] = 0
```

55.

Nope!

```
try(solve(matrix(c(0, 0, 1, 0), nrow = 2)))
```

```
## Error in solve.default(matrix(c(0, 0, 1, 0), nrow = 2)) :  
## Lapack routine dgesv: system is exactly singular: U[1,1] = 0
```

56.

False. If \mathbf{A} lacks full row rank, then \mathbf{A}^2 will as well (since \mathbf{A} does not span \mathbf{R}^n). Therefore inconsistent systems are possible for some b . If \mathbf{A} is invertible:

$$\mathbf{A}^2 x = b \mathbf{A} x = \mathbf{A}^{-1} b$$

False unless b is the zero vector and/or \mathbf{A} the identity.

57.

False. Proof by contradiction.

$$\begin{aligned} \mathbf{A}^{-1} \mathbf{A} &= \mathbf{I} \\ -(\mathbf{A}) \mathbf{A} &= \mathbf{I} \\ -\mathbf{A}^2 &= \mathbf{I} \quad \mathbf{A}^2 = -\mathbf{I} \\ \mathbf{A} &= -\mathbf{A}^{-1} \mathbf{I} \\ \mathbf{A} &= -\mathbf{A} \end{aligned}$$

58.

True. If any such matrix existed, it would satisfy $\mathbf{A}^4 = \mathbf{I}$.

```
library(matador)
A <- matrix(c(1, 0, 0, -1), nrow = 2)
mat_pows(A, 2)
```

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

```
M <- mat2latex(A, sink = TRUE)
```

$$\begin{aligned} \mathbf{A}^2 &= \mathbf{M} \\ \mathbf{A} &= \mathbf{A}^{-1} \mathbf{M} \\ (\mathbf{A}^{-1})^2 &= \mathbf{M} \\ (\mathbf{A}^{-1})^2 &= \mathbf{A}^2 \\ \mathbf{A}^4 &= \mathbf{I} \\ \mathbf{A}^2 \mathbf{A}^2 &= \mathbf{I} \\ \mathbf{A}^4 &= \mathbf{I} \end{aligned}$$