Contents

Chapter 1

Vector Independence

Think back to a time when you played with blocks. If you had two blocks, you couldn't make many shapes out of them. With three, you had a few more options. With a dozen, you were able to make many more shapes. In the world of blocks, a span would be all the things you could make with a given set of blocks.

A vector span is similar, but in a mathematical sense. If I give you the coordinates for a vector and ask you to make everything you can from that vector using only the original vector, the result is the span. You can scale the vector, add it to itself — anything that is a linear combination of only that vector. As with the blocks, you'll find what you can make from one vector is limited. The span will be a line. However, when you are given two or more vectors to "play" with, you will be able to create much more. The span will be larger than in the case of having only one vector. The size of the span (sometimes referred to as a subspace) will depend on whether the vectors are linearly independent or dependent. In this chapter, we will examine what independence and dependence mean for vectors. In the next chapter, we will apply what we've learned about independence and dependence to determine the span of a set of vectors.

1.1 Overview: Independence and Dependence

You saw some linearly dependent vectors in the previous chapter. Now, we will expand this concept. A set of linearly independent vectors means that no vector is a combination of any other vector. Let's look at these three:

$$[100]$$

$$[010]$$

$$[001]$$

If you scale each vector as much as possible, the span encompasses the entire 3D real space.

A set of linearly dependent vectors means one or more of the vectors can be written as a combination of one of the vectors.

For example:

$$v_1 = [7 - 22]$$

$$v_2 = [14 - 44]$$

You can see that v_2 is $2 * v_1$. They are linearly dependent. This is a simple example, but when you encounter larger matrices, it won't be as obvious. You will learn computational techniques for figuring out independence.

Vector spans have practical applications in a number of fields. Computer graphics and physics are two of them. For example, in space travel, knowing the vector span is essential to calculating a slingshot maneuver that will give spacecraft a gravity boost from a planet. For this, you'd need to know the gravity vector of the planet relative to the sun and the velocity vectors that characterize the spacecraft. Engineers would use this information to figure out the trajectory angle that would allow the spacecraft to achieve a particular velocity in the desired direction. The span constrains the set of successful solutions.

1.2 Vector Independence

A set of vectors $S = \{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ is linearly independent if the only solution to the equation:

$$a_1\mathbf{v_1} + a_2\mathbf{v_2} + \cdots + a_n\mathbf{v_n} = 0$$

is

$$a_1 = a_2 = ... = a_n = 0$$

This means that no vector in the set can be written as a linear combination of the other vectors.

If there exists a nontrivial solution (i.e., a solution where some $a_i \neq 0$), then the vectors are said to be linearly dependent. This means that at least one vector in the set can be written as a linear combination of the other vectors.

The concept of vector independence is fundamental to the study of vector spaces, bases, and rank. You will learn more about these concepts in future modules.

1.2.1 Dependent Vectors

Let's start by looking at two vectors.

$$\mathbf{v_1} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$\mathbf{v_2} = \begin{bmatrix} -14 \\ -28 \end{bmatrix}$$

These two vectors are dependent, because $\mathbf{v_2} = -7 * \mathbf{v_2}$. This is an obvious example, but let's show it mathematically. If linearly independent, the two vectors must satisfy:

$$a_1\mathbf{v_1} + a_2\mathbf{v_2} = 0$$

which is:

$$2a_1 - 14a_2 = 0$$

$$4a_1 - 28a_2 = 0$$

To solve, multiply the top equation by -2 and add it to the bottom:

$$2a_1 - 14a_2 = 0$$

$$0 + 0 = 0$$

The bottom equation drops out. Now, solve for a_1 in the remaining equation:

$$a_1 = -7a_2$$

As you can see, one vector is a multiple of another.

$$a_1 \neq a_2 \neq 0$$

1.2.2 Independent Vectors

Let's see if these two vectors are independent.

$$\mathbf{v_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\mathbf{v_2} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

To be independent, the two vectors must satisfy:

$$a_1\mathbf{v_1} + a_2\mathbf{v_2} = 0$$

which is:

$$\begin{bmatrix} a_1 + 0 * a_2 = 0 \\ 0 * a_1 + -a_2 = 0 \end{bmatrix}$$

So:

$$a_1 = a_2 = 0$$

These vectors are not only independent, but they are orthogonal (perpendicular) to one another. You'll learn more about orthogonality later.

Here is an example whose solution isn't as obvious. You can solve using Gaussian elmination.

$$\mathbf{v_1} = [2\ 1]$$

$$\mathbf{v_2} = [1\ \ -6]$$

Rewrite as a system of equations:

$$a_1 * 2 + a_2 * 1 = 0$$

$$a_1 * 1 + a_2 * (-6) = 0$$

First, swap the equations so that the the top equation has a coefficient of 1 for a_1:

$$a_1 - 6a_2 = 0$$

$$2a_1 + a_2 = 0$$

Next, multiply row 1 by -2 and add it to row 2:

$$a_1 - 6a_2 = 0$$

$$0 - 11a_2 = 0$$

Multiply row 2 by 1 divided by 11.

$$a_1 - 6a_2 = 0$$

$$0 + a_2 = 0$$

Back substitute a_2 solution into the first equation:

$$a_1 = 0$$

$$a_2 = 0$$

Therefore, $a_1 = a_2 = 0$ and the two vectors are linearly independent.

Exercise 1 Vector Independence

Are these vectors independent?

$$[2 \quad 1 \quad 4]$$

$$[2 \quad -1 \quad 2]$$

$$[0 \quad 1 \quad -2]$$

Show your work.

Working Space

Answer on Page 29

1.3 Checking for Linear Independence Using Python

One way to use Python to check for linear independence is to use the linalg.solve() function to solve the system of equations. You need to create an array that contains the coefficients of the variable and a vector that contains the values on the right-side of each equation. So far, you have either been given equations that equal 0 or you have manipulated each equation to be equal to 0.

Let's first see how to use Python to solve the equations in the previous exercise. If the equations are linearly independent, then $a_1 = a_2 = a_3 = 0$.

Create a file called span_independence.Python and enter this code:

```
import numpy as np

A = np.array([[2, 2, 0],
              [1, -1, 1],
              [4, 2, -2]])
b = np.array([0, 0, 0])
c = np.linalg.solve(A,b)
print(c)
```

You should get this result, which shows the equations are linearly independent.

```
[0., -0.,  0.]
```

However, what happens if the equations are not independent? Let's make the first two equations dependent by making equation 1 two times equation 2. Enter this code into your file:

```
import numpy as np

D = np.array([[2, -2, 2],
              [1, -1, 1],
              [4, 2, -2]])
e = np.array([0, 0, 0])
f = np.linalg.solve(D,f)
print(f)
```

```
You should get many lines indicating an error. Among the spew, you should see:
```

```
raise LinAlgError("Singular matrix")
```

So, while the linalg.solve() function is quite useful for solving a system of independent linear equations, raising an error is not the most elegant way to figure out if the equations are dependent. That is where the concept of a determinant comes in. You will learn about that in the next section, but for now, let's use the linalg.solve() function to find a solution for a set of equations known to be linearly independent.

$$4x_1 + 3x_2 - 5x_3 = 2$$

$$-2x_1 - 4x_2 - 5x_3 = 5$$

$$8x_2 + 8x_3 = -3$$

You will create a matrix that contains all the coefficients and a vector that contains the values on the right-side of the equations.

Enter this code into your file.

```
G = np.array([[4, 3, -5],
              [-2, -4, 5],
              [8, 8, 0]])
h = np.array([2, 5, -3])

j = np.linalg.solve(G, h)
print(j)
```

You should get this answer:

```
[2.20833333, -2.58333333, -0.18333333]
```

CHAPTER 2

Span

2.1 Spans of Vectors

Knowing whether two vectors are linearly dependent or independent allows us to accurately describe the span of those two vectors (this expands to include any number of vectors). In the previous chapter, we saw that linear combinations of two linearly dependent vectors can only make vectors that lie on the same line as the two starting vectors. We saw this in 2D, but it also applies to 3D vectors. Consider the two vectors $\mathbf{u} = [2, 4, 3]$ and $\mathbf{v} = [4, 8, 6]$, shown in figure 2.1.

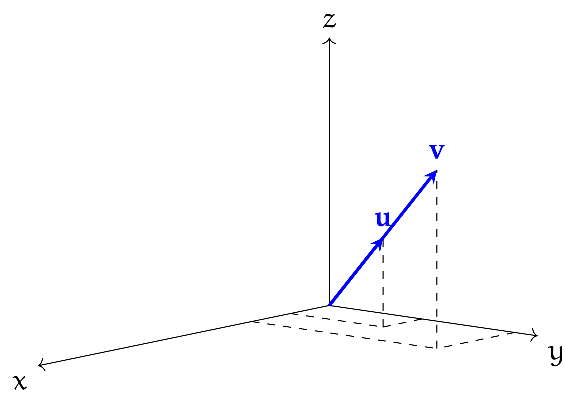


Figure 2.1: 3-dimensional vectors, $\mathbf{u}$ and $\mathbf{v}$

Notice that these two vectors are colinear (that is, they are on the same line), therefore they are linearly dependent and any combination of $\mathbf{u}$ and $\mathbf{v}$ will lie on the same line as $\mathbf{u}$ and $\mathbf{v}$. Therefore, we say the *the span of* $\mathbf{u}$ *and v is a line*. In fact, for any size list of linearly dependent vectors (whether it's one vector or one hundred), the span of that list is a line.

Now that you have a sense of what a span is, it is time for the formal mathematical definition. A vector span is the collection of vectors obtained by scaling and combining the original set of vectors in all possible proportions. Formally, if the set $S = \{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_n}\}$ contains vectors from a vector space V, then the span of S is given by:

$$\text{Span}(S) = \{a_1\mathbf{v_1} + a_2\mathbf{v_2} + \cdots + a_n\mathbf{v_n} : a_1, a_2, ..., a_n \in \mathbb{R}\} \tag{2.1}$$

This means that any vector in the Span(S) can be written as a linear combination of the vectors in S.

2.1.1 Spans of Independent Vectors

What if our list of vectors aren't all linearly dependent on each other? We've seen in 2 dimensions that any two independent vectors can be linearly combined to create any vector in $\mathbb{R}^2$. So, the span is described as a *plane* (in fact, it is the entire xy-plane, which we also call $\mathbb{R}^2$). How does this expand to 3-dimensional vectors?

Let's again consider two 3-dimensional vectors: $\mathbf{u} = [2, 4, 3]$ and $\mathbf{v} = [2, 1, 0]$, as shown in figure 2.2.

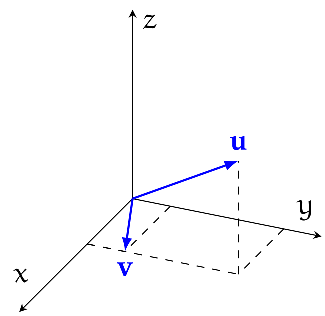


Figure 2.2: Linearly independent 3-dimensional vectors, **u** and **v**

Just like in two dimensions, any two independent vectors in $\mathbb{R}^3$ define a plane (see figure 2.3). This also applies to higher dimensions: the span of any two linearly independent vectors is a plane.

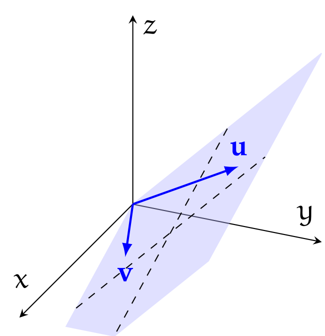


Figure 2.3: Linearly independent 3-dimensional vectors, **u** and **v**, define a plane.

If we have 3 independent vectors, then we can define a *3-dimensional space*. To understand this, first imagine a plane formed by two independent 3-dimensional vectors like in figure 2.3). If a third independent vector is introduced, it must not lie on the plane: if it did, it would be a linear combination of the first two and therefore not independent. This third vector allows us to move off the plane, and therefore all three independent vectors span $\mathbb{R}^3$. In review, 1 vector or set of dependent vectors span a *line*, 2 vectors or sets of dependent vectors span a *plane*, and 3 vectors or sets of dependent vectors span $\mathbb{R}^3$.

Example: Do the vectors $\mathbf{r} = [5, 4, -6]$, $\mathbf{s} = [0, -5, -10]$, and $\mathbf{t} = [0, 2, 4,]$ span a line, plane, or $\mathbb{R}^3$?

Solution: We need to determine the number of *independent vectors*. First, we'll check if **r** and **s** are independent. They are independent if the only solution to the equation below is $a_1 = a_2 = 0$:

$$a_1 [5, 4, -6] + a_2 [0, -5, -10] = [0, 0, 0]$$

Which we can write as a system of equations:

$$5a_1 + 0a_2 = 0$$

$$4a_1 - 5a_2 = 0$$

$$-6a_2 - 10a_2 = 0$$

From the first equation, we see that $5a_1 = 0$ which implies that $a_1 = 0$. Substituting that into the second equation:

$$4(0) - 5a_2 = 0$$

$$-5a_2 = 0$$

$$a_2 = 0$$

Therefore, vectors **r** and **s** are independent. Now let's check **r** and **t**:

$$a_1 [5, 4, -6] + a_2 [0, 2, 4,] = [0, 0, 0]$$

Which we can re-write as a system of equations:

$$5a_1 + 0a_2 = 0$$

$$4a_1 + 2a_2 = 0$$

$$-6a_1 + 4a_2 = 0$$

Again, from the first equation, we see that $a_1 = 0$. Substituting into the second:

$$4(0) + 2a_2 = 0$$

$$2a_2 = 0$$

$$a_2 = 0$$

Therefore, **r** and **t** are also independent. Last, we'll check **s** and **t** for independence:

$$a_1 [0, -5, -10] + a_2 [0, 1, 2,] = [0, 0, 0]$$

The system of equations:

$$0a_1 + 0a_2 = 0$$
$$-5a_1 + a_2 = 0$$
$$-10a_1 + 2a_2 = 0$$

The first equation doesn't tell us anything, since it would be true no matter what a_1 and a_2 are. We can solve the second equation for a_2 and substitute into the third equation:

$$a_2 = 5a_1$$
$$-10a_1 + 2(5a_1) = 0$$
$$-10a_1 + 10a_1 = 0$$

Which is also true for all a_1. In fact, there are many solutions to $a_1[0, -5, -10] + a_2[0, 1, 2,] = [0, 0, 0]$, $a_1 = 1$ and $a_2 = 5$ is an example. Therefore, **s** and **t** are *dependent*. So, we really have 2 independent vectors in the list, and therefore span($\mathbf{r}, \mathbf{s}, \mathbf{t}$) is a plane.

Exercise 2 Determining Span

Geometrically describe (line, plane, or $\mathbb{R}^3$) the span of the list of vectors.

1. $[1, 2, 4]$ and $[-2, -4, -8]$
2. $[2, 0, 0]$ and $[0, 1, 3]$
3. $[3, 0, 0]$ and $[0, 3, 3]$ and $[3, 3, 2]$

Working Space

Answer on Page 30

2.2 Determinants

Checking all these vectors by hand takes a long time. What if you had a list of 5, 10, or even 100 vectors? The determinant of a matrix is a scalar value that indicates whether the columns of a matrix are linearly independent. So, if you put all your vectors together in a matrix and take the determinant of that matrix, the result will tell you if all the vectors are independent or not. For a 2D matrix, the determinant is the area of the parallelogram defined by the column vectors. For a 3D matrix, the determinant is the volume of the parallelepiped (a six-dimensional figure formed by six parallelograms, such as a cube).

Let's plot the parallelogram for this matrix (see figure 2.4):

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

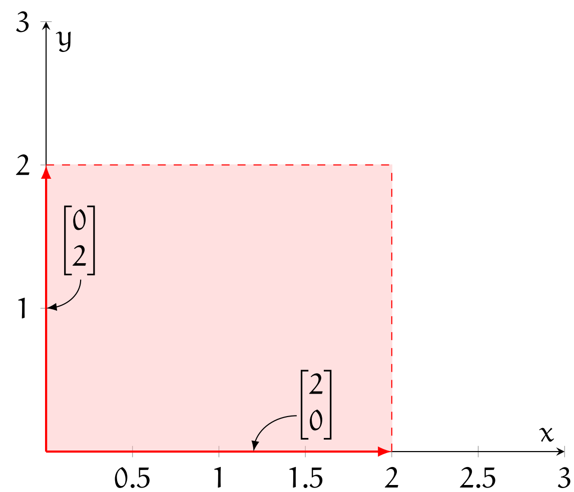


Figure 2.4: A parallelogram constructed from vectors $[2, 0]$ and $[0, 2]$

The formal definition for calculating the determinant of a 2 by 2 matrix is:

$$\det(A) = (a * d) - (b * c)$$

where

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

For the matrix plotted above, the determinant is $(2 * 2) - (0 * 0)$. You can also see that 4.0 is the area, base (2) times height (2).

You can use the determinant to see what happens to a shape when it goes through a linear transformation. Let's scale the 2 by 2 matrix by 4:

$$\begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}$$

Plot it (see figure 2.5):

Find the determinant.

$$(8 * 8) - (0 * 0) = 64$$

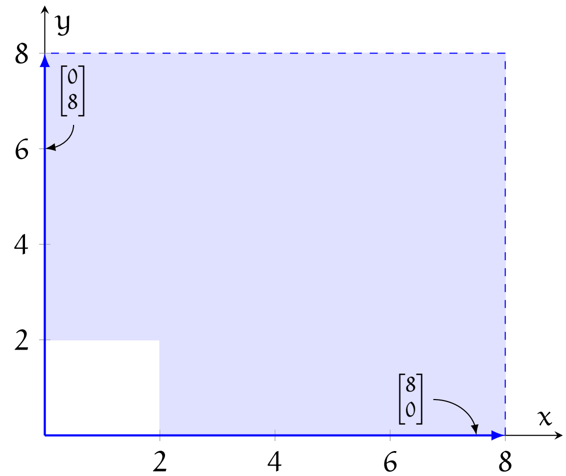


Figure 2.5: Scaling the matrix also scales the parallelogram.

You can see that scaling the matrix scaled the area by the scaling factor squared (see figure 2.6).

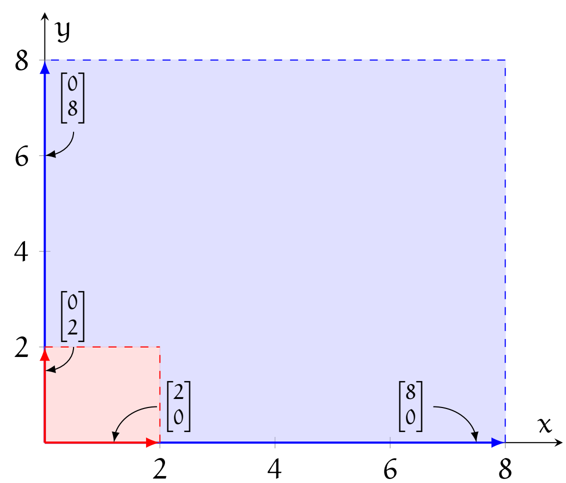


Figure 2.6: Scaling a matrix by a constant c increases the area of the parallelogram by a factor of c^2.

We can show why this is true mathematically. Suppose we have a 2 by 2 matrix A:

$$A = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$$

Then $\det(A) = wz - xy$. We can scale this matrix by a constant, c:

$$cA = c \cdot \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} cw & cx \\ cy & cz \end{bmatrix}$$

And we can take the determinant:

$$\det(cA) = \det\left(\begin{bmatrix} cw & cx \\ cy & cz \end{bmatrix}\right) = cw(cz) - cx(cy) = c^2(wz - xy) = c^2 \cdot \det(A)$$

Therefore, scaling a 2 by 2 matrix by a factor changes the determinant by that factor squared. What about higher dimensions? If each side of a cube were scaled by a factor of c, then the volume of the cube would change by a factor of c^3 (feel free to confirm this yourself). And if a tesseract (a four-dimensional cube) had each side scaled by a factor of c, then the hypervolume (four-dimensional volume) would be scaled by a factor of c^4. Do you notice a pattern?

In fact, scaling an $n \times n$ matrix by a constant factor, c, changes the determinant of that $n \times n$ matrix by a factor of c^n.

What happens if the columns of a matrix are not independent? Let's plot this matrix (see figure 2.7):

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

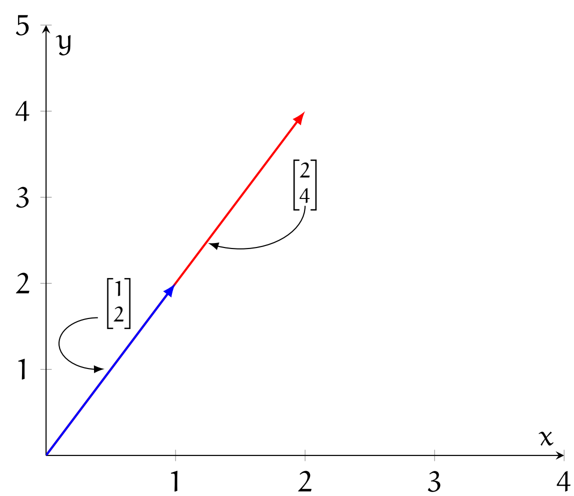


Figure 2.7: The vectors $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} 2 \\ 4 \end{bmatrix}$ are co-linear, so there is no area between them and the determinant of $\begin{bmatrix} 2 & 1 \\ 4 & 2 \end{bmatrix}$ is zero.

One vector overwrites the other. As you can see, the area is 0 because there is no space between the vectors. Therefore, the columns of the matrix are linearly dependent.

Exercise 3 Finding the Determinate

Plot the parallelogram represented by the columns of the matrix. What is the area of this parallelogram?

1. $\begin{bmatrix} 1 & 4 \\ -3 & 1 \end{bmatrix}$
2. $\begin{bmatrix} 5 & -5 \\ 5 & -1 \end{bmatrix}$
3. $\begin{bmatrix} 0 & -5 \\ -2 & 0 \end{bmatrix}$

Working Space

Answer on Page 30

Calculating the determinant for a 2 by 2 matrix is easy. For a larger matrix, finding the determinant is more complex and requires breaking down the matrix into smaller matrices until you reach the 2x2 form. The process is called expansion by minors. For our purposes, we simply want to first check to see if a matrix contains linearly independent rows and columns before using our Python code to solve.

Modify your code so that is uses the $np.linalg.det()$ function. If the determinant is not zero, then you can call the $np.linalg.solve()$ function. Your code should look like this:

```
if (np.linalg.det(D) != 0):
    j = np.linalg.solve(D,e)
    print(j)
else:
     print("Rows and columns are not independent.")
```

2.3 Where to Learn More

Watch this video on *Linear Combinations and Vector Spans from Khan Academy*: http://rb.gy/g1snk

The Wolfram Demonstrations website has a fun, interactive demo where you can enter values for 2D and 3D matrices and see how the area or volume changes. https://demonstrations.wolfram.com/DeterminantsSeenGeometrically/#more

If you are curious about the *Expansion of Minors*, see: https://mathworld.wolfram.com/DeterminantExpansionbyMinors.html

Chapter 3

Matrices and Systems of Linear Equations

In the chapter on linear combinations, we saw that we can linearly combine vectors to create other vectors. Consider 3 vectors:

$$\mathbf{x} = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}$$

We can write a linear combination of these vectors:

$$c\mathbf{x} + d\mathbf{y} + e\mathbf{z}$$

Which we can expand to show the vectors:

$$c\begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} + d\begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} + e\begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -c - d - 2e \\ 2c + 2d + e \\ d \end{bmatrix}$$

We can also represent this combination with a matrix where each column is one of the vectors:

$$\begin{bmatrix} -1 & -1 & -2 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} c \\ d \\ e \end{bmatrix} = \begin{bmatrix} -c - d - 2e \\ 2c + 2d + e \\ d \end{bmatrix}$$

3.1 Trail Mix for Mars

Let's look at an applied problem. Three astronauts, Pat, Kai, and River, are getting ready for a trip to Mars. NASA food service is preparing trail mix for the voyage, tailored to each astronaut's taste. The chef needs to submit a budget based on the cost of the trail mix for each astronaut. The mix is made up of raisins, almonds, and chocolate.

Pat prefers a raisins:almonds:chocolate ratio of 6:10:4, Kai likes 2:3:15, and River wants 14:1:5. The chef can buy a kg of raisins for \$7.50, a kg of almonds for \$14.75, and a kg of chocolate for $22.25. Assuming each astronaut will get 20 kg of trail mix, which astronaut will cost more to feed?

First, set up a matrix to represent the raisins:almonds:chocolate ratios. (Conveniently, these ratios already add to 20.)

$$\text{MixRatios} = \begin{bmatrix} 6 & 10 & 4 \\ 2 & 3 & 15 \\ 14 & 1 & 5 \end{bmatrix}$$

Use a vector to represent the cost of each item:

$$\text{IngredientCost} = \begin{bmatrix} 7.50 \\ 14.75 \\ 22.25 \end{bmatrix}$$

To find the cost of trail mix for each astronaut, we simply find the dot product between the mix ratios and the ingredient costs to get:

Pat = $281.50
Kai = $615.50
River = $231.00

Exercise 4 Vector Matrix Multiplication

Multiply the array A with the vector v. Compute this by hand, and make sure to show your computations.

$$A = \begin{bmatrix} 1 & -2 & 3 & 5 \\ -4 & 2 & 7 & 1 \\ 3 & 3 & -9 & 1 \end{bmatrix}$$

$$v = \begin{bmatrix} 2 \\ 2 \\ 6 \\ -1 \end{bmatrix}$$

Working Space

Answer on Page 32

Exercise 5 Using Vector Matrix Multiplication

A college professor offers three different methods of determining a student's final grade. In method A, the student's grade is 20% based on attendance, 50% homework, 15% midterm, and 15% final. This professor knows many students can learn the material without attending every class, so with method B the student's grade is 50% homework, 20% midterm, and 30% final. Last, the professor knows some students don't do the homework but still show they understand the material by doing well on the tests. With method C, a student's grade is 40% midterm and 60% final. The professor uses whatever method results in the highest grade to determine each student's final grade.

Suppose Suzy has attended 65% of classes, has an average homework grade of 30%, earned a 80% on the midterm, and earned a 75% on the final. What final grade will her professor post?

Working Space

Answer on Page 32

3.1.1 Vector-Matrix Multiplication in Python

Most real-world problems use very large matrices, where it becomes impractical to do calculations by hand. As long as you understand how matrix-vector multiplication is performed, you will be equipped to use a computing language, like Python, to do the calculations for you.

Create a file called `vectors_matrices.py` and enter this code:

```
# import the python module that supports matrices
import numpy as np
```

```
# create an array
a = np.array([[5, 1, 3, -2],
              [1, -1, 8, 4],
              [6, 2, 1, 3]])

# create a vector
b = np.array([1, 2, 3, -8])

# calculate the dot product
print(a.dot(b))
```

When you run it, you should see:

```
[16, 6, 8]
```

3.2 Where to Learn More

Watch this video from Khan Academy about matrix-vector products: https://rb.gy/frga5

Chapter 4

Matrices

You have already gained experience with matrices earlier in this module, as well as when you have used spreadsheets. In this chapter, you will learn the types of matrices and get an introduction to some of the special matrices used for various calculations.

As you know, a matrix is a rectangular array of numbers arranged in rows and columns. The individual numbers in the matrix are called elements or entries. Matrices can be described by their dimensions. For example, a matrix with 2 rows and 3 columns is a 2 by 3 matrix.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

More generally, a matrix with m rows and n columns is referred to as an $m \times n$ matrix, or simply an m-by-n matrix; m and n are its dimensions.

The general form of a 2×3 matrix A is:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix}$$

4.1 Types of Matrices

Matrices can be described by their shape:

- **Row Matrix:** Has only one row.
- **Column Matrix:** Has only one column.
- **Square Matrix:** Has the same number of rows and columns.
- **Rectangular Matrix:** Has an unequal number of rows and columns.

They can also be described by their unique numerical properties. Special matrices that come in handy for certains types of computations. These are a few of the most common special matrices:

- **Zero Matrix:** Only contains entries that are zero.

- **Identity Matrix:** Sometimes called the unit matrix, it is a square matrix whose diagonal entries are 1 and all other entries are 0.
- **Symmetric Matrix:** A square matrix that equals its transpose. The next section shows how to create the transpose of a matrix,
- **Diagonal Matrix:** Has nonzero elements on the main diagonal, but all other elements are zero
- **Triangular Matrix:** This is a special square matrix that can be upper triangular or lower triangular. If upper, the main diagonal and all entries above it are nonzero while the lower entries are all zero. If lower, the main diagonal and all the entries below it are nonzero, while the upper entries are all zero.

4.1.1 Symmetric Matrices

If you want to find out if a square matrix is symmetric, you need to transpose it. If the transpose is equal to the original matrix, then the matrix is symmetric.

To transpose a matrix, flip it over its diagonal so that the rows and columns are switched, like this:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

After transposing:

$$A^T = \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} \\ a_{1,2} & a_{2,2} & a_{3,2} \\ a_{1,3} & a_{2,3} & a_{3,3} \end{bmatrix}$$

Note that A^T means the transpose of A.

Let's see how this works for the following square matrix, A.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$

Switch the rows and columns to get the transpose:

$$A^T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$

Notice that $A = A^T$, so the matrix is symmetric.

What about matrix B?

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{bmatrix}$$

Switch the rows and columns to get the transpose:

$$B = \begin{bmatrix} 1 & 2 & 7 \\ 2 & 4 & 8 \\ 3 & 5 & 9 \end{bmatrix}$$

Note that $B \neq B^T$, so B is not symmetrical.

Exercise 6 Matrix Transposition

Find the transpose of this matrix. Is it symmetric?

$$A = \begin{bmatrix} 3 & -2 & 4 \\ -2 & 6 & 2 \\ 4 & 2 & 3 \end{bmatrix}$$

Working Space

Answer on Page 33

4.1.2 Creating Matrices in Python

Create a file called `matrices_creation.py` and enter this code:

```
# import the python module that supports matrices
import numpy as np
# Use the function np.array to define a matrix that
# contains specific values that you supply.
A = np.array([[ 5, 1, 3],
              [ 1, -1, 8],
              [ 6, 2, 1]])
# The transpose function returns
A.transpose()
```

When you run it, you should see:

```
array([[ 5, 1, 6],
       [ 1, -1, 2],
       [ 3, 8, 1]])
```

As you can see, $A \neq A^T$, so A is not symmetric. Try another:

```
# create a matrix, B
B = np.array([[ 5, 1, 6],
              [ 1, -1, 2],
              [ 6, 2, 1]])
B.transpose()
```

When you run it, you should see:

```
array([[ 5, 1, 6],
       [ 1, -1, 2],
       [ 6, 2, 1]])
```

B is symmetric. You can actually transpose any matrix using this function, but a matrix cannot be symmetric unless it is square.

Try this code to see what happens when you transpose a rectangular matrix.

```
# create a matrix, J
J = np.array([[ 5, 1, 3, 0],
              [ 1, -1, 8, 11],
              [ 6, 2, 1,-7]])
J.transpose()
```

Note that transposing a rectangular matrix changes its dimension from 3 by 4 to 4 by 3. You should see a transposed matrix, but it's not symmetric.

```
array([[ 5,  1,  6],
       [ 1, -1,  2],
       [ 3,  8,  1],
       [ 0, 11, -7]])
```

4.1.3 Creating Special Matrices in Python

Use the same file to add this code for creating a zero matrix.

```
# create an 8 by 10 Zero matrix.
 C = np.zeros((8, 10))
 C
```

When you run it, you should see:

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])
```

Add the following code to create an 8 by 8 Identity matrix.

```
# create an 8 by 8 Identity matrix
 D = np.eye(8)
 D
```

When you run it, you should see:

```
array([[1., 0., 0., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 0., 0., 0., 1.]])
```

As you progress in your studies, you will learn the importance of diagonal matrices and of extracting the diagonal of a matrix. Let's see how to extract a diagonal, then create a diagonal matrix.

```
# create a  matrix
W = np.array([[1, 2, 3, 4],
              [5, 6, 7, 8],
              [-8, -7, -6, -5],
              [-4, -3, -2, -1]])
```

Extract the main diagonal using np.diag(<array>,<diagonal to extract>). Passing 0 as the second parameter specifies the main diagonal. A positive value extracts a diagonal from the upper part. A negative value extracts a diagonal from the lower part. Run this code then experiment passing other values to see what you get.

```
print(np.diag(W,0))
```

When you run it, you should see:

```
array([ 1,  6, -6, -1])
```

You can also use np.diag() to create a diagonal matrix from a 1D array. In this case, do not pass a second paramenter.

```
Q = np.array([1, 2, 3])
DiagArray = np.diag(Q))
print(DiagArray)
```

When you run it you should see;

```
[[1 0 0]
 [0 2 0]
 [0 0 3]]
```

Python has functions for extracting upper and lower triangular matrices. Try these:

```
print(np.triu(W))
print(np.tril(W))
```

You should see:

```
[[ 1  2  3  4]
 [ 0  6  7  8]
 [ 0  0 -6 -5]
 [ 0  0  0 -1]]

[[ 1  0  0  0]
 [ 5  6  0  0]
 [-8 -7 -6  0]
 [-4 -3 -2 -1]]
```

Appendix A

Answers to Exercises

Answer to Exercise 1 (on page 7)

Rewrite as a system of equations:

$$\begin{aligned} 2*a_1 + 2*a_2 + 0*a_3 &= 0 \\ 1*a_1 - 1*a_2 + 1*a_3 &= 0 \\ 4*a_1 + 2*a_2 - 2*a_3 &= 0 \end{aligned}$$

Simplify

$$\begin{aligned} 2a_1 + 2*a_2 &= 0 \\ a_1 - a_2 + a_3 &= 0 \\ 4a_1 + 2a_2 - 2a_3 &= 0 \end{aligned}$$

Swap row 2 and 1:

$$\begin{aligned} a_1 - a_2 + a_3 &= 0 \\ 2a_1 + 2*a_2 &= 0 \\ 4a_1 + 2a_2 - 2a_3 &= 0 \end{aligned}$$

Multiply row 1 by -2 and add to row 2:

$$\begin{aligned} a_1 - a_2 + a_3 &= 0 \\ 0 + 3*a_2 - 2a_3 &= 0 \\ 4a_1 + 2a_2 - 2a_3 &= 0 \end{aligned}$$

Multiply row 1 by -4 and add to row 3:

$$\begin{aligned} a_1 - a_2 + a_3 &= 0 \\ 0 + 3*a_2 - 2a_3 &= 0 \\ 0 + 6a_2 - 6a_3 &= 0 \end{aligned}$$

Multiply row 2 by -4 and add to row 3:

$$\begin{aligned} a_1 - a_2 + a_3 &= 0 \\ 0 + 3*a_2 - 2a_3 &= 0 \\ 0 + 0 - 2a_3 &= 0 \end{aligned}$$

Multiply row 3 by -1 and add to row 2:

$$\begin{aligned} a_1 - a_2 + a_3 &= 0 \\ 0 + 3*a_2 + 0 &= 0 \\ 0 + 0 - 2a_3 &= 0 \end{aligned}$$

Divide row 3 by -2 and row 2 by $\frac{1}{3}$:

$$\begin{aligned} a_1 - a_2 + a_3 &= 0 \\ 0 + a_2 + 0 &= 0 \\ 0 + 0 + a_3 &= 0 \end{aligned}$$

Backsubstitute a_2 and a_3 into row 1:

$$\begin{aligned} a_1 + 0 + 0 &= 0 \\ 0 + a_2 + 0 &= 0 \\ 0 + 0 + a_3 &= 0 \end{aligned}$$

Therefore

$$a_1 = a_2 = a_3 = 0$$

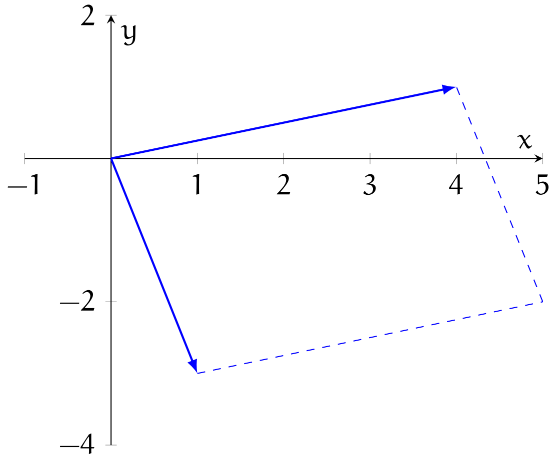
.

Answer to Exercise 2 (on page 12)

1. Since the second vector is a scalar multiple of the first, the span of $S = \{[1, 2, 4], [-2, -4, -8]\}$ is a *line*.
2. Since the second vector is not a scalar multiple of the first, the span of $S = \{[2, 0, 0,], [0, 1, 3]\}$ is a *plane*.
3. None of the three vectors are scalar multiples or linear combinations of the other two. Therefore, the span of $S = \{[3, 0, 0], [0, 3, 3], [3, 3, 2]\}$ is $\mathbb{R}^3$.

Answer to Exercise 3 (on page 16)

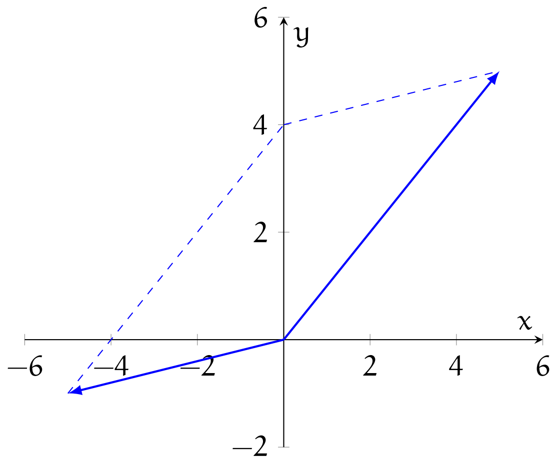
1. Our two vectors from the columns of the matrix are $[1, -3]$ and $[4, 1]$. Plotting:



The area of this parallelogram is the same as the determinant of the matrix:

$$\det\left(\begin{bmatrix} 1 & 4 \\ -3 & 1 \end{bmatrix}\right) = 1 \cdot 1 - (4 \cdot -3) = 1 + 12 = 13$$

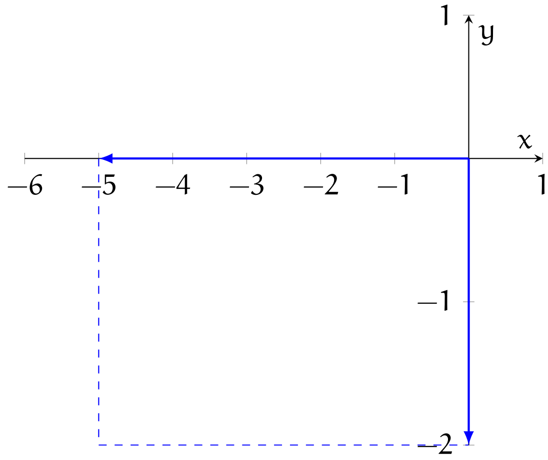
2. Our two vectors from the columns of the matrix are $[5, 5]$ and $[-5, -1]$. Plotting:



The area of this parallelogram is the same as the determinant of the matrix:

$$\det\left(\begin{bmatrix} 5 & -5 \\ 5 & -1 \end{bmatrix}\right) = 5 \cdot -1 - (-5 \cdot 5) = -5 + 25 = 20$$

3. Our two vectors from the columns of the matrix are $[0, -2]$ and $[-5, 0]$. Plotting:



This is a rectangle, and we can see the area is $5 \cdot 2 = 10$. However, the determinant is:

$$\det\left(\begin{bmatrix} 0 & -5 \\ -2 & 0 \end{bmatrix}\right) = 0 \cdot 0 - (-5 \cdot -2) = 0 - 10 = -10$$

We will discuss this unusual response in a future chapter.

Answer to Exercise 4 (on page 20)

$$Av = \begin{pmatrix} 11 & 37 & -43 \end{pmatrix}$$

Answer to Exercise 5 (on page 21)

The different methods can be represented in a matrix:

$$\begin{bmatrix} 0.20 & 0.50 & 0.15 & 0.15 \\ 0 & 0.50 & 0.20 & 0.30 \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix}$$

And Suzy's individual grades can be represented by a vector:

$$\begin{bmatrix} 65 \\ 30 \\ 80 \\ 75 \end{bmatrix}$$

To see the results of the three different methods, we can multiply the matrix and the vector:

$$\begin{bmatrix} 0.20 & 0.50 & 0.15 & 0.15 \\ 0 & 0.50 & 0.20 & 0.30 \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 65 \\ 30 \\ 80 \\ 75 \end{bmatrix} = \begin{bmatrix} 0.2(65) + 0.5(30) + 0.15(80) + 0.15(75) \\ 0(65) + 0.5(30) + 0.2(80) + 0.3(75) \\ 0(65) + 0(30) + 0.4(80) + 0.6(75) \end{bmatrix}$$

Which yields:

$$\begin{bmatrix} 51.25 \\ 53.5 \\ 77 \end{bmatrix}$$

Since method C yields the highest grade, the professor will post a final grade of 77.

Answer to Exercise 6 (on page 25)

$$A = A^t = \begin{bmatrix} 3 & -2 & 4 \\ -2 & 6 & 2 \\ 4 & 2 & 3 \end{bmatrix}$$