

For the following practice problems, define matrices **A**, **B**, **C**, and **D** as follows:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 4 & 9 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} -2 & -6 \\ 5 & 15 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 7 & 1 & 4 \\ -1 & 0 & 4 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

I. Practicing with Matrices - By Hand

Execute the matrix operations as indicated. NOTE: If you believe the matrix operation cannot be executed, state so and suggest an explanation why.

1. $\mathbf{A} + \mathbf{B}$
2. $k \times \mathbf{C}$ (Leave k as an unknown here.)
3. $\mathbf{A} - \mathbf{B}$ (This should be the same as $\mathbf{A} + (-1) \times \mathbf{B}$.)
4. $\mathbf{B} + \mathbf{D}$
5. $\mathbf{A} \times \mathbf{B}$
6. $\mathbf{B} \times \mathbf{A}$
7. $\mathbf{D} \times \mathbf{A}$
8. $\mathbf{A} \times \mathbf{D}$
9. \mathbf{C}^T
10. $(3 \times \mathbf{D})^T$
11. $3 \times (\mathbf{D}^T)$

II. Practicing with Matrices - With Python

The `numpy` library has the ability to work well with N -dimensional arrays. Since a matrix, per our notes, is a rectangular array with (usually) real-valued entries, the `numpy` library will be incredibly helpful for dealing with linear algebra.

There are two places within the `numpy` library that may be of particular importance when working on linear algebra-related things: [numpy.linalg](#) and [numpy.matlib](#). Check out the documentation for these and the functions available within each routine.

Try to recreate your answers from Part I, problems 1-10 here. We've specified how to put **A** into `numpy` below as well as some functions that may be helpful for you. Remember to also check out the documentation, linked above.

```
import numpy as np ## import numpy library
a = np.array([[2,3],[4,9]]) ## create matrix A; remember that np.array takes one input
encased in [brackets] and then each row is in a separate set of [brackets] inside the
larger set.
## you'll need to create matrix B for this to work
a + b ## matrix addition
2 * a ## scalar multiplication
np.matmul(a,b) ## matrix multiplication
np.linalg.inv(a) ## matrix inverse
np.round(np.linalg.inv(a),3) ## sometimes floating point arithmetic gives us some issues
so it's a good idea to round to a few decimal places; the number of digits to which you
round your answers will be dependent on your use case.
np.linalg.det(a) ## matrix determinant
np.eye(2) ## identity matrix of dimension 2x2 (eye = I)
```

In addition to recreating your answers from Part I, try to find the answers to the following practice problems.

1. Generate the 4×4 identity matrix.
2. Calculate the determinant of \mathbf{A} .
3. Calculate the inverse of \mathbf{A} (written \mathbf{A}^{-1}).
4. Show that $\mathbf{A} \times \mathbf{A}^{-1}$ and $\mathbf{A}^{-1} \times \mathbf{A}$ both yield the identity matrix.
5. Calculate the determinant of \mathbf{B} .
6. Calculate \mathbf{B}^{-1} .
7. Based on the value reached in 2. and the value reached in 5., what might we theorize about the relationship between determinants and inverses?

III. Theoretical Practice

1. Using matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} above, plus $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_2$, identify which, if any, are:
 - i. Triangular
 - ii. Diagonal
 - iii. Vectors
 - iv. Square
 - v. Symmetric
2. Consider $\mathbf{X}_{k \times l}$ and $\mathbf{Y}_{m \times n}$.
 - i. What must be true about k , l , m , and n for \mathbf{XY} to exist? ($\mathbf{XY} = \mathbf{X} \times \mathbf{Y}$)
 - ii. What must be true k , l , m , and n for \mathbf{YX} to exist?
 - iii. What must be true k , l , m , and n for both \mathbf{XY} and \mathbf{YX} to exist?
3. Compare the first column and the second column of \mathbf{B} . What do you notice?
 - i. On Monday 12/5, we're going to discuss how this directly affects Part II, problems 2., 3., 5., and 6.