# ChangeofBasis

August 11, 2018

In this notebook we will take a look at one of the most basic concepts in linear algebra known as changing basis. A sound conceptual understanding of changing basis will give you a great foundation for more complicated areas of linear algebra.

Imagine we have gone shopping on Monday and on Wednesday. The price of the Monday shop was £5 and the price of the Wednesday shop was £8. If we were to plot this data as a scatter plot we would simply plot the price of Monday's shop on the x-axis and the price of Wednesday's shop on the y-axis:

```
In [1]: %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt
        import matplotlib as mpl
        mpl.rcParams['figure.dpi']= 100

        monday_price = 3
        wednesday_price = 4

        fig = plt.figure(figsize=(3,3))
        plt.plot(monday_price, wednesday_price, 'r*')
        plt.xlim([0, 5])
        plt.xticks(np.arange(6))
        plt.ylim([0, 5])
        plt.yticks(np.arange(6))
        plt.xlabel('Monday Price (£)')
        plt.ylabel('Wednesday Price (£)')
        plt.suptitle('Shopping Prices')
        plt.show()
```


Importantly the datapoint forms a vector in our two dimensional space $\begin{bmatrix}3\\4\end{bmatrix}$. This vector is saying that our datapoint is 5 units along the x-axis and 8 units along the y-axis. Equivalently we could say that our point is:

$$3 * \begin{bmatrix}1\\0\end{bmatrix} + 4 * \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}3\\0\end{bmatrix} + \begin{bmatrix}0\\4\end{bmatrix} = \begin{bmatrix}3\\4\end{bmatrix}$$

In matrix form this would be:

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

which is equivalent to multiplying by the identity matrix and so the vector $\begin{bmatrix}3\\4\end{bmatrix}$ remains as $\begin{bmatrix}3\\4\end{bmatrix}$.

The vectors $\begin{bmatrix}1\\0\end{bmatrix}$ and $\begin{bmatrix}0\\1\end{bmatrix}$ are our basis vectors and are plotted below:

```
In [2]: basis_x = [1,0]
        basis_y = [0,1]

        fig = plt.figure(figsize=(3,3))
        plt.plot(monday_price, wednesday_price, 'r*')
        plt.quiver(0, 0, basis_x[0], basis_x[1], color='b', units='xy',
                   angles='xy', scale_units='xy', scale=1.)
        plt.quiver(0, 0, basis_y[0], basis_y[1], color='g', units='xy',
                   angles='xy', scale_units='xy', scale=1.)

        plt.xlim([-5, 5])
```

```
plt.xticks(np.arange(-5, 6))
plt.ylim([-5, 5])
plt.yticks(np.arange(-5, 6))
plt.xlabel('Monday Price (£)')
plt.ylabel('Wednesday Price (£)')
plt.suptitle('Shopping Prices')
plt.show()
```


Our datapoint expressed in terms of basis vectors would then look like this:

```
In [3]: fig = plt.figure(figsize=(3,3))
        plt.plot(monday_price, wednesday_price, 'r*')

        for x in range(monday_price):
            plt.quiver(x, 0, basis_x[0], basis_x[1], color='b', units='xy',
                       angles='xy', scale_units='xy', scale=1.)

        for y in range(wednesday_price):
            plt.quiver(monday_price, y, basis_y[0], basis_y[1], color='g',
                       units='xy', angles='xy', scale_units='xy', scale=1.)

        plt.xlim([-5, 5])
        plt.xticks(np.arange(-5, 6))
        plt.ylim([-5, 5])
        plt.yticks(np.arange(-5, 6))
        plt.xlabel('Monday Price (£)')
        plt.ylabel('Wednesday Price (£)')
        plt.suptitle('Shopping Prices')
        plt.show()
```


This shows how every datapoint is actually just a linear combination of our basis vectors!

Our basis vectors are what allow us to make sense of our vector space and represent our co-ordinate system. However these basis vectors are completely arbitrary! So what happens when we decide to use a different set of bassis vectors to encode our space? For example what if we wanted to use $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}0\\2\end{bmatrix}$ as our new basis vectors? How could we describe the point $\begin{bmatrix}3\\4\end{bmatrix}$ using these new basis vectors? The new basis vectors are plotted below:

```
In [4]: new_basis_x = [1,1]
        new_basis_y = [0,2]

        fig = plt.figure(figsize=(3,3))
        plt.plot(monday_price, wednesday_price, 'r*')
        plt.quiver(0, 0, basis_x[0], basis_x[1], color='b', units='xy',
                   angles='xy', scale_units='xy', scale=1.)
        plt.quiver(0, 0, basis_y[0], basis_y[1], color='g', units='xy',
                   angles='xy', scale_units='xy', scale=1.)
        plt.quiver(0, 0, new_basis_x[0], new_basis_x[1], color='r',
                   units='xy', angles='xy', scale_units='xy', scale=1.)
        plt.quiver(0, 0, new_basis_y[0], new_basis_y[1], color='m',
                   units='xy', angles='xy', scale_units='xy', scale=1.)

        plt.xlim([-5, 5])
        plt.xticks(np.arange(-5, 6))
        plt.ylim([-5, 5])
        plt.yticks(np.arange(-5, 6))
```

```
plt.xlabel('Monday Price (£)')
plt.ylabel('Wednesday Price (£)')
plt.suptitle('Shopping Prices')
plt.show()
```


This is what we mean by a change of basis. Note how the new basis vectors are a linear transformation of the original basis vectors. This linear transformation can be represented as a matrix A:

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

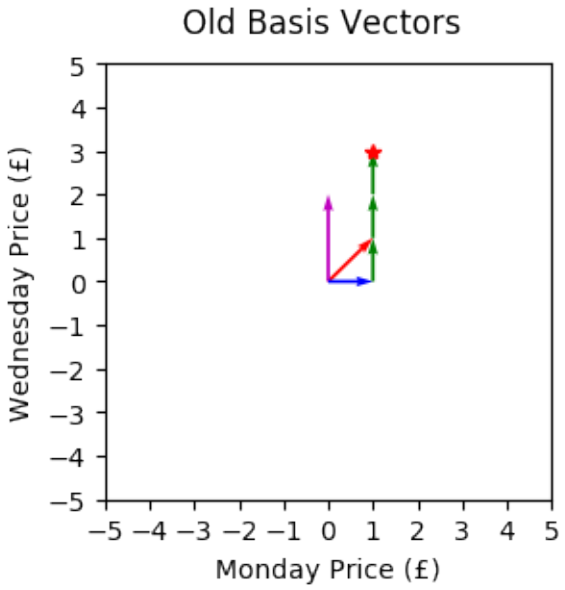
where the columns indicate the positions of the new basis vectors. Imagine we have a new point $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ according to the new basis vectors, what would this point be using our original standard basis vectors ($\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$)? To work this out we can simply apply the transformation matrix A to the point $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$:

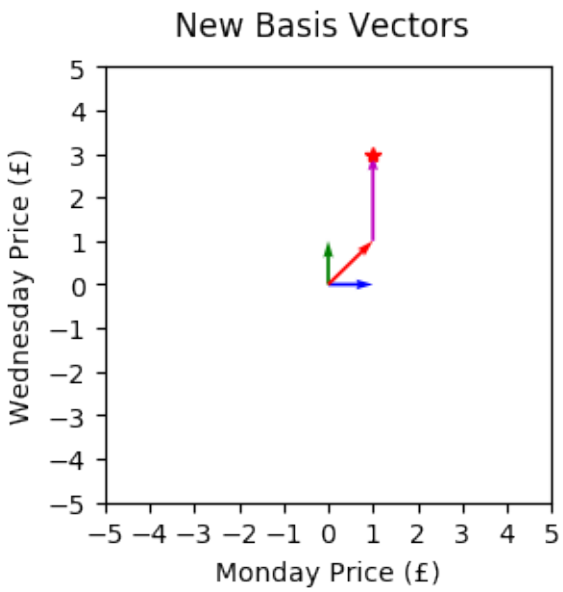
$$\begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

The point $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ using our new basis vectors is therefore the point $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$ using our original basis vectors. This is plotted below:

```
In [5]: fig = plt.figure(figsize=(3,3))
        point_x = 1
        point_y = 3
        plt.plot(point_x, point_y, 'r*')
        plt.quiver(0, 0, new_basis_x[0], new_basis_x[1], color='r',
                   units='xy', angles='xy', scale_units='xy', scale=1.)
        plt.quiver(0, 0, new_basis_y[0], new_basis_y[1], color='m',
                   units='xy', angles='xy', scale_units='xy', scale=1.)

        for x in range(point_x):
            plt.quiver(x, 0, basis_x[0], basis_x[1], color='b',
                       units='xy', angles='xy', scale_units='xy',
                       scale=1.)

        for y in range(point_y):
            plt.quiver(point_x, y, basis_y[0], basis_y[1], color='g',
                       units='xy', angles='xy', scale_units='xy',
                       scale=1.)

        plt.xlim([-5, 5])
        plt.xticks(np.arange(-5, 6))
        plt.ylim([-5, 5])
        plt.yticks(np.arange(-5, 6))
        plt.xlabel('Monday Price (£)')
        plt.ylabel('Wednesday Price (£)')
        plt.suptitle('Old Basis Vectors')
        plt.show()

        fig = plt.figure(figsize=(3,3))
        point_x_new = 1
        point_y_new = 1
        plt.plot(point_x, point_y, 'r*')
        plt.quiver(0, 0, basis_x[0], basis_x[1], color='b', units='xy',
                   angles='xy', scale_units='xy', scale=1.)
        plt.quiver(0, 0, basis_y[0], basis_y[1], color='g', units='xy',
                   angles='xy', scale_units='xy', scale=1.)

        for x in range(point_x_new):
            plt.quiver(x * new_basis_x[0], x * new_basis_x[1],
                       new_basis_x[0], new_basis_x[1], color='r',
                       units='xy', angles='xy', scale_units='xy',
                       scale=1.)

        for y in range(point_y_new):
            plt.quiver((point_x_new * new_basis_x[0]) +
                       (y * new_basis_y[0]),
                       (point_x_new * new_basis_x[1]) +
                       (y * new_basis_y[1]),
```

```
                  new_basis_y[0], new_basis_y[1], color='m',
                  units='xy', angles='xy', scale_units='xy',
                  scale=1.)

plt.xlim([-5, 5])
plt.xticks(np.arange(-5, 6))
plt.ylim([-5, 5])
plt.yticks(np.arange(-5, 6))
plt.xlabel('Monday Price (£)')
plt.ylabel('Wednesday Price (£)')
plt.suptitle('New Basis Vectors')
plt.show()
```






So you can see that a matrix can be viewed as a linear transformation or a change of basis, whereby the columns of a matrix represent the locations of the new basis vectors. This is really cool as it basically gives you a set of instructions to jump between different vectors spaces!

Why would this be useful in the real world? Well imagine that we want to go back the other way i.e. we have a point using our original basis vectors and we want to know what this point will be using some other set of basis vectors. In our shopping exmaple, imagine that each shop consisted of a certain number of apples and oranges. Monday's shop consisted of 2 apples and 2 oranges, while Wednesday's shop consisted of 2 apples, 4 oranges. What was the individual price of an apple and an orange? We can solve this using simultaneous equations but it is also useful to think about the problem in terms of a change in basis or linear transformation.

In terms of linear transformations and changes in basis vectors, we want our point in terms of shopping prices to be expressed in terms of prices for individual items. Luckily the numbers of items give us the instructions we need to change our basis in exactly this way! The basis vector for Monday's shop price can be moved to the number of apples on each day $\begin{bmatrix}2\\2\end{bmatrix}$ and the basis vector for Wednesday's shop price can be moved to the number of oranges on each day $\begin{bmatrix}2\\4\end{bmatrix}$. This gives us the following transformation matrix M:

$$M = \begin{bmatrix}2 & 2\\2 & 4\end{bmatrix}$$

To see that this makes sense remember that the two shops can be written as:

$$2a + 2o = 3$$
$$2a + 4o = 4$$

where $a = price\ of\ an\ apple$ and $o = price\ of\ an\ orange$. This can be re-written as a matrix multiplication:

$$\begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix} * \begin{bmatrix} a \\ o \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

which should hopefully re-assure you that M is indeed the matrix transformation that we need and that it tells us the location of our new basis vectors! The problem we then wish to solve is what point $\begin{bmatrix} a \\ o \end{bmatrix}$ in the new vector space encoding the price of items corresponds to the point $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ in the original vector space encoding prices of shopping trips. One way to solve this is to use simultaenous equations:

$$\begin{aligned} 2x + 2y &= 3 \\ 2x + 4y &= 4 \\ 2y &= 1 \\ y &= .5 \\ 2x + 1 &= 3 \\ x &= 1 \end{aligned}$$

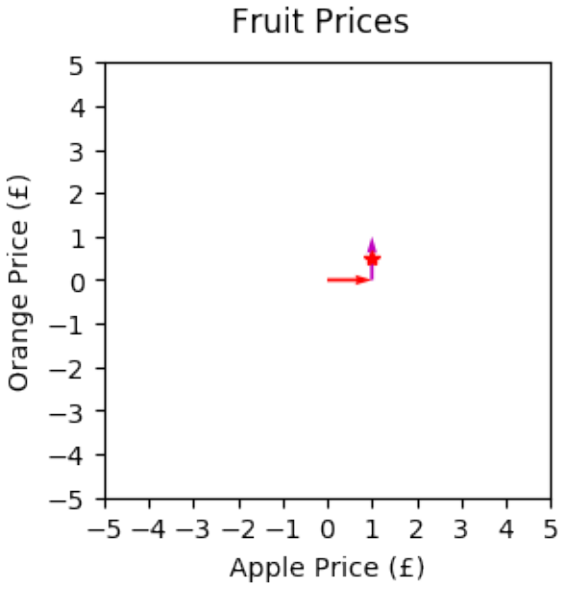
Therefore:

$$\begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix} * \begin{bmatrix} 1 \\ .5 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

The vector $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ in our vector space encoding the prices of our shops is equivalent to the vector $\begin{bmatrix} 1 \\ .5 \end{bmatrix}$ in the vector space encoding the prices of individual apples and oranges! One apple cost £1 and one orange cost £0.5! To re-emphasise the point here are the two vectors in their respective vector spaces:

```
In [6]: fig = plt.figure(figsize=(3,3))
        plt.plot(monday_price, wednesday_price, 'r*')

        for x in range(monday_price):
            plt.quiver(x, 0, basis_x[0], basis_x[1], color='b',
                       units='xy', angles='xy', scale_units='xy',
                       scale=1.)

        for y in range(wednesday_price):
```

```
    plt.quiver(monday_price, y, basis_y[0], basis_y[1],
               color='g', units='xy', angles='xy',
               scale_units='xy', scale=1.)

plt.xlim([-5, 5])
plt.xticks(np.arange(-5, 6))
plt.ylim([-5, 5])
plt.yticks(np.arange(-5, 6))
plt.xlabel('Monday Price (£)')
plt.ylabel('Wednesday Price (£)')
plt.suptitle('Shopping Prices')
plt.show()

apple_price = 1
orange_price = .5

fig = plt.figure(figsize=(3,3))
plt.plot(apple_price, orange_price, 'r*')
plt.quiver(0, 0, basis_x[0], basis_x[1], color='r',
           units='xy', angles='xy', scale_units='xy',
           scale=1.)
plt.quiver(apple_price, 0, basis_y[0], basis_y[1],
           color='m', units='xy', angles='xy',
           scale_units='xy', scale=1.)

plt.xlim([-5, 5])
plt.xticks(np.arange(-5, 6))
plt.ylim([-5, 5])
plt.yticks(np.arange(-5, 6))
plt.xlabel('Apple Price (£)')
plt.ylabel('Orange Price (£)')
plt.suptitle('Fruit Prices')
plt.show()
```


and here is what the linear transformation M looks like in our original vector space:

```
In [7]: new_basis_x = [2,2]
        new_basis_y = [2,4]

        fig = plt.figure(figsize=(3,3))
        plt.plot(monday_price, wednesday_price, 'r*')
```

```
for x in range(monday_price):
    plt.quiver(x, 0, basis_x[0], basis_x[1], color='b',
               units='xy', angles='xy', scale_units='xy',
               scale=1.)

for y in range(wednesday_price):
    plt.quiver(monday_price, y, basis_y[0], basis_y[1],
               color='g', units='xy', angles='xy',
               scale_units='xy', scale=1.)

plt.quiver(0, 0, new_basis_x[0], new_basis_x[1], color='r',
           units='xy', angles='xy', scale_units='xy',
           scale=1.)
plt.quiver(new_basis_x[0] * apple_price,
           new_basis_x[1] * apple_price,
           new_basis_y[0], new_basis_y[1],
           color='m', units='xy', angles='xy',
           scale_units='xy', scale=1.)

plt.xlim([-5, 5])
plt.xticks(np.arange(-5, 6))
plt.ylim([-5, 7])
plt.yticks(np.arange(-5, 6))
plt.xlabel('Monday Price (£)')
plt.ylabel('Wednesday Price (£)')
plt.suptitle('Shopping Prices')
plt.show()
```

Here I hope you can really see how the linear transformation matrix M changes the span of our basis vectors and causes the co-ordinates of our datapoint, which is expressed in terms of the basis vectors, to also change. In fact all points in space are transformed by the matrix M, not just the basis vectors or our single datapoint. Imagine a cloud of infinite datapoints and how all these points would move given the matrix transformation M.

The idea behind changes in basis are really central to linear algebra so if you can get your head around this concept you will be in a great position to delve further into the field! Please leave comments below if you have any questions or suggestions on how to improve this tutorial!