A1 Getting Started

These guides use LiveScripts which can be edited, interacted with and run "live". There are also exports of standard MATLAB .m files and PDF versions that include the figure outputs. The LiveScripts are in the folder `source`, with data in the folders below `data`. The scripts assume you run them from the level above `source` and `data`. You can tell where you are by typing pwd at the MATLAB command line. When a LiveScript is open, you can step through using the "**Run and Advance**" button.

The code is fairly self-explanatory, but here are a few points:

- You can get help on MATLAB functions by typing `doc`.
- Variables that hold numbers have a size that is described as: number_of_rows x number_of_columns. So, a single scalar is size 1x1.
- A list, or vector, of N numbers, can be a row vector (1 row, N columns so size 1xN), or a column vector (size `Nx1`). Usually it is better to work with column vectors.
- For arrays, the index numbering starts at 1 (not 0 as in some other languages).
- For arrays, the first row is the **top** row.
- Functions that input x and y arrays accept them in the order x,y. For example `plot(x,y)`

```
a = 6  % sets variable 'a' to be the scalar number 6
```

```
a =
     6
```

```
rowvec= [12 24 7 99]
```

```
rowvec = 1×4
    12    24     7    99
```

```
colvec = [55 ; 3 ; 67]
```

```
colvec = 3×1
    55
     3
    67
```

```
whos a rowvec colvec
```

```
  Name        Size            Bytes  Class     Attributes

  a           1x1                 8  double
  colvec      3x1                24  double
  rowvec      1x4                32  double
```

We can see that a is listed with size 1x1 and type (Class) `double`, using 8 bytes (64 bits). Note the sizes of `rowvec` and `colvec`.

The default class for numeric variables is `double` and this generally works well.

Some functions that return values can be used with a single number to set the size of the output. For example, `zeros(4),` gives a 4x4 matrix of zeros. It is preferable to be clear about the size, for example use `zeros([4 4])` to get a 4x4 matrix.

in summary, think of variables as always being 2D with a size: rows x columns. Arrays with more dimensions e.g. 3D, 4D, have additional dimenisons but the first two are always rows then columns.

David Atkinson

```
disp(datetime)
```

```
    06-Jan-2025 22:12:25
```