

## Lists, Vectors, Arrays

\_\_\_\_\_ or \_\_\_\_\_ hold multiple values (usually) of the same type. \_\_\_\_\_ hold multiple values, possibly of different types.

Elements are stored in order, and elements can be referenced by their \_\_\_\_\_. The first element has \_\_\_\_\_ 0 or 1 depending on the language. The \_\_\_\_\_ of a list, vector, or array is the number of elements in it. An empty list has a \_\_\_\_\_ of 0.

You can \_\_\_\_\_ an item to the beginning of a list or vector or \_\_\_\_\_ an item to the end. Sometimes, lists can be \_\_\_\_\_ inside other lists.

In R and Python, you can take a slice of a list (or R vector) using the list indices:

```
my_list[a:b]
```

Example:

```
my_list[3:6]
```

In Python, **a** is the index of the \_\_\_\_\_ value, **b** is the index of the \_\_\_\_\_ value EXCLUSIVE (meaning it's not included).

In R, the first number is the index of the \_\_\_\_\_ value, and the second number is the index of the \_\_\_\_\_ value INCLUSIVE.

## Assigning Values

To change the value of an element in a list, assign a new value to it:

```
my_list = [7,6,5,4]
my_list[2] = 3
my_list
```

`my_list` now contains \_\_\_\_\_.

If instead you assign a new value as:

```
my_list = [7,6,5,4]
my_list = [1,2,3]
my_list
```

`my_list` now contains \_\_\_\_\_.