

## Exercise



## Multiple ways to access variables

An important lesson you should learn early on is that there are multiple ways to do things in R. For example, to generate the first five integers we note that `1:5` and `seq(1,5)` return the same result.

There are also multiple ways to access variables in a data frame. For example we can use the square brackets `[]` instead of the accessor `$`. Example code is included in the editor.

If you instead try to access a column with just one bracket,

```
murders["population"]
```

R returns a subset of the original data frame containing just this column. This new object will be of class `data.frame` rather than a vector. To access the column itself you need to use either the `$` accessor or the double square brackets `[[`.

Parentheses, in contrast, are mainly used alongside functions to indicate what argument the function should be doing something to. For example, when we did `class(p)` in the last question, we wanted the function `class` to do something related to the argument `p`.

This is an example of how R can be a bit idiosyncratic sometimes. It is very common to find it confusing at first.

### Instructions

100 XP

- Use the square brackets `[]` to extract the state abbreviations and assign them to the object `b`.
- Then use the `identical` function to determine if `a`, as defined in the previous exercises, and `b` are the same.

## script.R

```
1 # We extract the population like this:
2 p <- murders$population
3
4 # This is how we do the same with the square brackets:
5 o <- murders[["population"]]
6
7 # We can confirm these two are the same
8 identical(o, p)
9
10 # Use square brackets to extract `abb` from `murders` and assign it to
11 b<- murders[["abb"]]
12 b
13 # Check if `a` and `b` are identical
14 identical(a,b)
```



## R Console