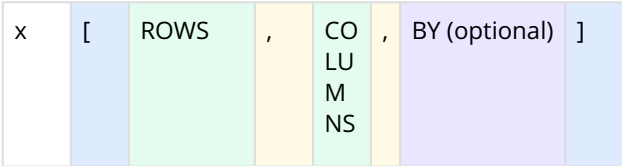


## Data manipulation and filtering with data.table

By using [data.table](#) to structure our cytometry data, we can manipulate and filter the datasets extremely quickly. Subsetting rows or columns of a data.table can be achieved using the following structure:



Filtering approaches:

- Rows can be selected by the row number or values in a specified column
- Columns can be specified as the column numbers or the column names
- BY is used when grouping data together, but this is not demonstrated on this page

Here is a quick guide to some common operations, but a more comprehensive guide can be found on the [data.table home page](#).

Task	Command (with minimum variable inputs)
Access a 'demo' dataset stored within Spectre, and assign it to 'cell.dat'.	<pre>cell.dat &lt;- Spectre::demo.clustering</pre>
Subsetting data using <b>column</b> numbers	<pre>## Select the first column of the data.table sub &lt;- cell.dat[,1] sub  ## Select columns 1, 3, and 5 sub &lt;- cell.dat[,c(1,3,5)] sub  ## Select columns 1 to 10, and 15 sub &lt;- cell.dat[,c(1:10,15)] sub</pre>
Subsetting data using <b>column</b> names	<pre>## View a list of column names as.matrix(names(cell.dat))        [,1] [1,] "FileName" [2,] "NK11" [3,] "CD3" [4,] "CD45" [5,] "Ly6G" [6,] "CD11b" [7,] "B220" [8,] "CD8a" [9,] "Ly6C" [10,] "CD4" [11,] "NK11_asinh" [12,] "CD3_asinh" [13,] "CD45_asinh"</pre>

Task	Command (with minimum variable inputs)						
	<div><pre>[14,] "Ly6G_asinh" [15,] "CD11b_asinh" [16,] "B220_asinh" [17,] "CD8a_asinh" [18,] "Ly6C_asinh" [19,] "CD4_asinh" [20,] "Sample" [21,] "Group" [22,] "Batch" [23,] "FlowSOM_cluster" [24,] "FlowSOM_metacluster" [25,] "Population" [26,] "UMAP_X" [27,] "UMAP_Y"</pre></div> <div><pre>## Select columns names 11 to 19 cols &lt;- names(cell.dat)[c(11:19)] cols</pre></div> <div><pre>[1] "NK11_asinh" "CD3_asinh" "CD45_asinh" "Ly6G_asinh" "CD11b_asinh" "B220_asinh" "CD8a_asinh" "Ly6C_asinh" "CD4_asinh"</pre></div> <p>To select columns based on <b>column name</b>, either <code>'..'</code> needs to go before the vector of column names, or <code>', with = FALSE'</code> needs to be added to the end of the data.table filtering arguments.</p> <div><pre>## OPTION 1 - Select columns using '..' sub &lt;- cell.dat[ ,..cols] sub  ## OPTION 2 - Select columns using 'with = FALSE' sub &lt;- cell.dat[ ,cols, with = FALSE] sub</pre></div>						
Subsetting data using <b>row</b> numbers	<div><pre>## Select the first row of the data.table sub &lt;- cell.dat[1, ] sub  ## Select rows 1, 3, and 5 sub &lt;- cell.dat[c(1,3,5), ] sub  ## Select rows 1 to 10, and 15 sub &lt;- cell.dat[c(1:10,5), ] sub</pre></div>						
Subsetting data using <b>row</b> values in a selected column	<p>Subsetting rows using data.table can be performed with the following structure:</p> <table><tr><td>cell.dat</td><td>[</td><td>CONDITIONS</td><td>,</td><td></td><td>]</td></tr></table> <p>The <b>conditions</b> is essentially a filtering operation to determine which rows have a value in a specific column that is equal to, higher, or lower than a specified value. This can be performed using something like this:</p>	cell.dat	[	CONDITIONS	,		]
cell.dat	[	CONDITIONS	,		]		

Task	Command (with minimum variable inputs)
	<div><pre>## Creates a TRUE/FALSE results for which rows contain "Ly6C_asinh" values above 2 cell.dat[["Ly6C_asinh"]] &gt; 2</pre></div> <p>We can use this conditional TRUE/FALSE results to select which rows to include from our data.table.</p> <div><pre>## Select rows (cells) where 'Ly6C_asinh' is above 2 sub &lt;- cell.dat[cell.dat[["Ly6C_asinh"]] &gt; 2, ] sub  ## Select rows (cells) where 'CD45_asinh' is below 3 sub &lt;- cell.dat[cell.dat[["CD45_asinh"]] &lt; 3, ] sub  ## Select rows (cells) where 'FlowSOM_metacluster' is above 2 sub &lt;- cell.dat[cell.dat[["FlowSOM_metacluster"]] == 5,] sub  ## Select rows (cells) where 'Population' is 'Infil Macrophages' sub &lt;- cell.dat[cell.dat[["Population"]] == 'Infil Macrophages',] sub</pre></div>
Subsetting data using multiple <b>row</b> values in multiple columns	<p>Multiple 'or' arguments an be added by using ' ', for example: <b>cell.dat[A   B   C,]</b>. This is an 'OR' operation, so all cells that are Ly6C_asinh &gt; 2, in addition to all cells that are 'Infil Macrophages' will be included, rather than only cells satisfying both conditions (which would be an 'AND' operation).</p> <div><pre>## Select rows (cells) where # 'Ly6C_asinh' is above 2 OR # 'Population' is 'Infil Macrophages'  sub &lt;- cell.dat[cell.dat[["Ly6C_asinh"]] &gt; 2                   cell.dat[["Population"]] == 'Infil Macrophages'                 ,] sub</pre></div>

## Extracting text before or after a symbol

Let's say we have a character vector, with a symbol dividing it into two:

```
x <- c("everything before|everything after")
```

We can use the following to extract everything **before** '|'.

```
before <- sub("\\|.+", "", x)
before
```

"everything before"

And we can use the following to extract everything **after** '|'.  

---

```
after <- sub(".*\\|", "", x)
after
```

```
"everything after"
```