

# Package ‘dt.infix’

June 29, 2022

**Title** Additional Infix Operators for the R Package Data.table

**Version** 0.0.1

**Description** Additional infix operators and helper functions to use with the R package data.table during exploratory data analysis.

**License** GPL-3 + file LICENSE

**Depends** R (>= 3.5.0)

**Imports** data.table,  
magrittr,  
stats,  
stringi,  
testthat,  
utils

**Encoding** UTF-8

**LazyData** true

**LazyLoad** yes

**RemoteUrl** <https://github.com/andrewrech/dt.infix>

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

## R topics documented:

allduplicated . . . . .	2
coerce_dt . . . . .	2
fbind . . . . .	3
withoutAllNA . . . . .	3
%exclude% . . . . .	4
%excludedef% . . . . .	4
%include% . . . . .	5
%includedef% . . . . .	5
%likef% . . . . .	6
%with% . . . . .	6
%without% . . . . .	7
%withoutrows% . . . . .	7
<b>Index</b>	<b>8</b>

---

<code>allduplicated</code>	<i>Return all duplicated rows of a data.table.</i>
----------------------------	--

---

**Description**

Return all duplicated rows of a data.table.

**Usage**

```
allduplicated(dt, by = NULL)
```

**Arguments**

<code>dt</code>	Data table.
<code>by</code>	Character vector. Names of columns to test for duplicates.

**Value**

A data.table of duplicated rows from dt.

---

<code>coerce_dt</code>	<i>Recursively coerce a list to a flat data table, correctly naming columns.</i>
------------------------	--

---

**Description**

Recursively coerce a list to a flat data table, correctly naming columns.

**Usage**

```
coerce_dt(l, ix = NULL, names = TRUE)
```

**Arguments**

<code>l</code>	List to coerce.
<code>ix</code>	Numeric vector. Vector of nested indices to extract recursively.
<code>names</code>	Logical. Preserve names?

**Value**

Data table with non-coercible objects elided.

---

fbind	<i>Sequentially load and row bind data tables.</i>
-------	--

---

**Description**

fbind sequentially loads data tables from disk and binds the data tables together by row names. The purpose of this method is to support parallelized data table operations, bypassing the 1.5 Gb return vector limit of the lapply family of functions. Both .RDS and plain text tables are supported.

**Usage**

```
fbind(files)
```

**Arguments**

files            Character vector. Files to load and row bind.

**Value**

A data table.

---

withoutAllNA	<i>Convenience infix operator to remove all(NA) or all(NULL) columns and rows from a data.table by reference.</i>
--------------	---

---

**Description**

Remove columns from a data table if all values are NA or NULL; useful when subsetting.

**Usage**

```
withoutAllNA(dt)
```

**Arguments**

dt                A data.table.

**Value**

A data.table subset of dt with columns of all NA or NULL removed.

---

`%exclude%`*Convenience infix operator to return vector elements excluding those matching a regular expression.*

---

**Description**

Convenience infix operator to return vector elements excluding those matching a regular expression.

**Usage**

```
vector %exclude% pattern
```

**Arguments**

<code>vector</code>	Vector.
<code>pattern</code>	Pattern.

**Value**

A vector of elements in `vector` not matching `pattern`.

---

`%excludedef%`*Convenience infix operator to return vector elements excluding a fixed pattern.*

---

**Description**

Convenience infix operator to return vector elements excluding a fixed pattern.

**Usage**

```
vector %excludedef% pattern
```

**Arguments**

<code>vector</code>	Vector.
<code>pattern</code>	Pattern.

**Value**

A vector of elements in `vector` not matching `pattern`.

---

<code>%include%</code>	<i>Convenience infix operator to return vector elements matching a regular expression.</i>
------------------------	--

---

**Description**

Convenience infix operator to return vector elements matching a regular expression.

**Usage**

```
vector %include% pattern
```

**Arguments**

<code>vector</code>	Vector.
<code>pattern</code>	Pattern.

**Value**

A vector of elements in `vector` matching `pattern`.

---

<code>%includef%</code>	<i>Convenience infix operator to return vector elements matching a fixed pattern.</i>
-------------------------	---

---

**Description**

Convenience infix operator to return vector elements matching a fixed pattern.

**Usage**

```
vector %includef% pattern
```

**Arguments**

<code>vector</code>	Vector.
<code>pattern</code>	Pattern.

**Value**

A vector of elements in `vector` matching `pattern`.

---

<code>%likef%</code>	<i>Convenience infix operator to return logical vector of elements matching a fixed pattern.</i>
----------------------	--

---

**Description**

Convenience infix operator to return logical vector of elements matching a fixed pattern.

**Usage**

```
vector %likef% pattern
```

**Arguments**

<code>vector</code>	Vector.
<code>pattern</code>	Pattern.

**Value**

A logical vector the same length as `vector` of elements matching `pattern`.

---

<code>%with%</code>	<i>Convenience infix operator to return a data.table of columns matching a regular expression.</i>
---------------------	--

---

**Description**

Convenience infix operator to return a data.table of columns matching a regular expression.

**Usage**

```
dt %with% cols
```

**Arguments**

<code>dt</code>	A data.table.
<code>cols</code>	Vector of column patterns to include.

**Value**

A data.table subset of `dt` with column names matching `cols`.

---

%without%	<i>Convenience infix operator to remove columns from a data.table by reference.</i>
-----------	---

---

**Description**

Convenience infix operator to remove columns from a data.table by reference.

**Usage**

```
dt %without% cols
```

**Arguments**

dt	A data.table.
cols	Vector of columns patterns to remove.

**Value**

A data.table subset of dt with column names not matching cols.

---

%withoutrows%	<i>Convenience infix operator to remove data.table rows.</i>
---------------	--

---

**Description**

Convenience infix operator to remove data.table rows.

**Usage**

```
dt %withoutrows% rows
```

**Arguments**

dt	A data.table.
rows	Row indices to delete.

**Value**

A data.table subset of dt with rows from index rows.

# Index

`%exclude%`, 4  
`%excludedef%`, 4  
`%include%`, 5  
`%includedef%`, 5  
`%likef%`, 6  
`%with%`, 6  
`%without%`, 7  
`%withoutrows%`, 7  
  
`allduplicated`, 2  
  
`coerce_dt`, 2  
  
`fbind`, 3  
  
`withoutAllNA`, 3