

# Package ‘tidylog’

September 2, 2019

**Type** Package

**Title** Logging for 'dplyr' and 'tidyr' Functions

**Version** 0.2.0

**Description** Provides feedback about 'dplyr' and 'tidyr' operations.

**License** MIT + file LICENSE

**Imports** dplyr, tidyr, glue, clisymbols

**Suggests** testthat, covr, lintr

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/elbersb/tidylog/>

**BugReports** <https://github.com/elbersb/tidylog/issues>

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Benjamin Elbers [aut, cre] (<<https://orcid.org/0000-0001-5392-3448>>)

**Maintainer** Benjamin Elbers <[elbersb@gmail.com](mailto:elbersb@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-09-02 16:10:02 UTC

## R topics documented:

filter . . . . .	2
gather . . . . .	3
group_by . . . . .	3
inner_join . . . . .	4
mutate . . . . .	5
select . . . . .	6
spread . . . . .	6
summarize . . . . .	7
tidylog . . . . .	8
transmute . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

filter	<i>Wrapper around dplyr::filter and related functions that prints information about the operation</i>
--------	---

---

## Description

Wrapper around dplyr::filter and related functions that prints information about the operation

## Usage

```
filter(.data, ...)  
filter_all(.data, ...)  
filter_if(.data, ...)  
filter_at(.data, ...)  
distinct(.data, ...)  
distinct_all(.data, ...)  
distinct_if(.data, ...)  
distinct_at(.data, ...)  
top_n(.data, ...)  
drop_na(.data, ...)
```

## Arguments

.data	a tbl; see <a href="#">filter</a>
...	see <a href="#">filter</a>

## Value

see [filter](#)

## Examples

```
filter(mtcars, mpg > 20)  
#> filter: removed 18 rows (56%), 14 remaining  
filter(mtcars, mpg > 100)  
#> filter: removed all rows (100%)
```

---

gather	<i>Wrapper around tidyr::gather that prints information about the operation</i>
--------	---

---

**Description**

Wrapper around tidyr::gather that prints information about the operation

**Usage**

```
gather(.data, ...)
```

**Arguments**

.data	a tbl; see <a href="#">gather</a>
...	see <a href="#">gather</a>

**Value**

see [gather](#)

**Examples**

```
# create id
mtcars$id <- 1:nrow(mtcars)
gathered <- gather(mtcars, "col", "data", -id)
#> gather: reorganized (mpg, cyl, disp, hp, drat, . . .) into (col, data) [was 32x12, now 352x3]
```

---

group_by	<i>Wrapper around dplyr::group_by and related functions that prints information about the operation</i>
----------	---

---

**Description**

Wrapper around dplyr::group\_by and related functions that prints information about the operation

**Usage**

```
group_by(.data, ...)
group_by_all(.data, ...)
group_by_if(.data, ...)
group_by_at(.data, ...)
```

**Arguments**

.data            a tbl; see [group\\_by](#)  
...                see [group\\_by](#)

**Value**

see [group\\_by](#)

**Examples**

```
group_by(mtcars, am, cyl)
#> group_by: 2 grouping variables (am, cyl)
```

---

inner_join	<i>Wrapper around dplyr::inner_join and related functions that prints information about the operation</i>
------------	---

---

**Description**

Wrapper around dplyr::inner\_join and related functions that prints information about the operation

**Usage**

```
inner_join(x, y, by = NULL, ...)  
full_join(x, y, by = NULL, ...)  
left_join(x, y, by = NULL, ...)  
right_join(x, y, by = NULL, ...)  
anti_join(x, y, by = NULL, ...)  
semi_join(x, y, by = NULL, ...)
```

**Arguments**

x                a tbl; see [inner\\_join](#)  
y                a tbl; see [inner\\_join](#)  
by               a vector; see [inner\\_join](#)  
...               see [inner\\_join](#)

**Value**

see [inner\\_join](#)

**Examples**

```

left_join(dplyr::band_members, dplyr::band_instruments, by = "name")
#> left_join: added one column (plays)
#>           > rows only in x    1
#>           > rows only in y   (1)
#>           > matched rows     2
#>           >                    ===
#>           > rows total       3

```

---

mutate	<i>Wrapper around dplyr::mutate and related functions that prints information about the operation</i>
--------	---

---

**Description**

Wrapper around dplyr::mutate and related functions that prints information about the operation

**Usage**

```

mutate(.data, ...)

mutate_all(.data, ...)

mutate_if(.data, ...)

mutate_at(.data, ...)

add_tally(.data, ...)

add_count(.data, ...)

replace_na(.data, ...)

fill(.data, ...)

```

**Arguments**

.data	a tbl; see <a href="#">mutate</a>
...	see <a href="#">mutate</a>

**Value**

see [mutate](#)

**Examples**

```

mutate(mtcars, new_var = 1)
#> mutate: new variable 'new_var' with one unique value and 0% NA
mutate(mtcars, new_var = NA)

```

---

select	<i>Wrapper around dplyr::select and related functions that prints information about the operation</i>
--------	---

---

**Description**

Wrapper around dplyr::select and related functions that prints information about the operation

**Usage**

```
select(.data, ...)
select_all(.data, ...)
select_if(.data, ...)
select_at(.data, ...)
```

**Arguments**

.data	a tbl; see <a href="#">select</a>
...	see <a href="#">select</a>

**Value**

see [select](#)

**Examples**

```
select(mtcars, mpg, wt)
#> select: dropped 9 variables (cyl, disp, hp, drat, qsec, ...)
select(mtcars, dplyr::matches("a"))
#> select: dropped 7 variables (mpg, cyl, disp, hp, wt, ...)
```

---

spread	<i>Wrapper around tidyr::spread that prints information about the operation</i>
--------	---

---

**Description**

Wrapper around tidyr::spread that prints information about the operation

**Usage**

```
spread(.data, ...)
```

**Arguments**

`.data` a tbl; see [spread](#)  
`...` see [spread](#)

**Value**

see [spread](#)

**Examples**

```
# create id
mtcars$id <- 1:nrow(mtcars)
gathered <- gather(mtcars, "col", "data", -id)
#> gather: reorganized (mpg, cyl, disp, hp, drat, ...) into (col, data) [was 32x12, now 352x3]
spread(gathered, col, data)
#> spread: reorganized (col, data) into (am, carb, cyl, disp, drat, ...) [was 352x3, now 32x12]
```

---

summarize	<i>Wrapper around <code>dplyr::summarize</code> and related functions that prints information about the operation</i>
-----------	---

---

**Description**

Wrapper around `dplyr::summarize` and related functions that prints information about the operation

**Usage**

```
summarize(.data, ...)  
  
summarize_all(.data, ...)  
  
summarize_at(.data, ...)  
  
summarize_if(.data, ...)  
  
summarise(.data, ...)  
  
summarise_all(.data, ...)  
  
summarise_at(.data, ...)  
  
summarise_if(.data, ...)  
  
tally(.data, ...)  
  
count(.data, ...)
```

**Arguments**

.data            a tbl; see [summarize](#)  
...              see [summarize](#)

**Value**

see [summarize](#)

**Examples**

```
summarize_all(mtcars, mean)  
#> summarize_all: now one row and 11 columns, ungrouped
```

---

tidylog

*outputs some information about the data frame/tbl*

---

**Description**

outputs some information about the data frame/tbl

**Usage**

```
tidylog(.data)
```

**Arguments**

.data            a tbl/data frame

**Value**

same as .data

**Examples**

```
tidylog(mtcars)  
#> tidylog: data.frame with 32 rows and 11 columns
```



---

transmute	<i>Wrapper around <code>dplyr::transmute</code> and related functions that prints information about the operation</i>
-----------	---

---

### Description

Wrapper around `dplyr::transmute` and related functions that prints information about the operation

### Usage

```
transmute(.data, ...)
```

```
transmute_all(.data, ...)
```

```
transmute_if(.data, ...)
```

```
transmute_at(.data, ...)
```

### Arguments

`.data` a tbl; see [transmute](#)

`...` see [transmute](#)

### Value

see [transmute](#)

### Examples

```
transmute(mtcars, mpg = mpg * 2)
#> transmute: dropped 10 variables (cyl, disp, hp, drat, wt, ...)
#> transmute: changed 32 values (100%) of 'mpg' (0 new NA)
```

# Index

`add_count` (`mutate`), 5  
`add_tally` (`mutate`), 5  
`anti_join` (`inner_join`), 4  
  
`count` (`summarize`), 7  
  
`distinct` (`filter`), 2  
`distinct_all` (`filter`), 2  
`distinct_at` (`filter`), 2  
`distinct_if` (`filter`), 2  
`drop_na` (`filter`), 2  
  
`fill` (`mutate`), 5  
`filter`, 2, 2  
`filter_all` (`filter`), 2  
`filter_at` (`filter`), 2  
`filter_if` (`filter`), 2  
`full_join` (`inner_join`), 4  
  
`gather`, 3, 3  
`group_by`, 3, 4  
`group_by_all` (`group_by`), 3  
`group_by_at` (`group_by`), 3  
`group_by_if` (`group_by`), 3  
  
`inner_join`, 4, 4  
  
`left_join` (`inner_join`), 4  
  
`mutate`, 5, 5  
`mutate_all` (`mutate`), 5  
`mutate_at` (`mutate`), 5  
`mutate_if` (`mutate`), 5  
  
`replace_na` (`mutate`), 5  
`right_join` (`inner_join`), 4  
  
`select`, 6, 6  
`select_all` (`select`), 6  
`select_at` (`select`), 6  
`select_if` (`select`), 6  
  
`semi_join` (`inner_join`), 4  
`spread`, 6, 7  
`summarise` (`summarize`), 7  
`summarise_all` (`summarize`), 7  
`summarise_at` (`summarize`), 7  
`summarise_if` (`summarize`), 7  
`summarize`, 7, 8  
`summarize_all` (`summarize`), 7  
`summarize_at` (`summarize`), 7  
`summarize_if` (`summarize`), 7  
  
`tally` (`summarize`), 7  
`tidylog`, 8  
`top_n` (`filter`), 2  
`transmute`, 9, 9  
`transmute_all` (`transmute`), 9  
`transmute_at` (`transmute`), 9  
`transmute_if` (`transmute`), 9