

Package ‘spdplyr’

June 2, 2017

Type Package

Version 0.1.3

Title Data Manipulation Verbs for the Spatial Classes

Description Methods for 'dplyr' verbs for 'sp' 'Spatial' classes. The basic verbs that modify data attributes, remove or re-arrange rows are supported and provide complete 'Spatial' analogues of the input data. The group by and summarize work flow returns a non-topological spatial union. There is limited support for joins, with left and inner to copy attributes from another table.

URL <https://github.com/mdsumner/spdplyr>

BugReports <https://github.com/mdsumner/spdplyr/issues>

Depends R (>= 3.2.3), dplyr, sp

Imports lazyeval, methods, rlang, spbabel, tibble, utils

Suggests testthat, maptools, raster, rmarkdown, knitr, covr

VignetteBuilder knitr

LazyData yes

License GPL-3

RoxygenNote 6.0.1

NeedsCompilation no

Author Michael D. Sumner [aut, cre],
Joscha Legewie [ctb],
Jussi Jousimo [ctb]

Maintainer Michael D. Sumner <mdsumner@gmail.com>

Repository CRAN

Date/Publication 2017-06-02 16:09:42 UTC

R topics documented:

spdplyr-package	2
dplyr-S3	3
dplyr-Spatial	3
mpoint1	7
show,Spatial-method	7
spmap	7

Index	8
--------------	----------

spdplyr-package	<i>Data Manipulation Verbs for Spatial Classes.</i>
-----------------	---

Description

Data Manipulation Verbs for Spatial Classes.

Details

The spdplyr package provides methods for the dplyr verbs.

O. dplyr verbs

<code>filter</code>	filter
<code>slice</code>	slice
<code>arrange</code>	arrange
<code>select</code>	select
<code>rename</code>	rename
<code>distinct</code>	distinct
<code>mutate</code>	mutate
<code>transmute</code>	transmute
<code>group_by</code>	group_by
<code>summarise</code>	summarise (or summarize)

I. dplyr-Spatial

dplyr verbs operating directly on Spatial objects The implementation of `group_by` requires that sp objects are able use the object extending class "data.frame", `summarise` acts on the attributes in the table according to the expressions use and on the geometry by performing a non-topological union.

dplyr-S3

dplyr S3 methods

Description

Worker functions used by dplyr features.

Usage

```
## S3 method for class 'Spatial'  
tbl_vars(x)
```

```
## S3 method for class 'Spatial'  
groups(x)
```

Arguments

x input Spatial object

Details

These will work for the Spatial-DataFrame objects, though not properly for any-Spatial.

Examples

```
if (utils::packageVersion("dplyr") > "0.5.0") {  
  
  spmap %>% mutate_if(is.numeric, as.character)  
  spmap %>% mutate_all(funs(as.character))  
  spmap %>% mutate_at(vars(starts_with("L")), funs(as.integer))  
}
```

dplyr-Spatial

Dplyr verbs for Spatial

Description

Direct application of the dplyr verbs to Spatial objects. There is no need for a conversion from and to Spatial with this approach. Not all verbs are supported, see Details.

Usage

```
## S3 method for class 'Spatial'  
mutate(.data, ...)  
  
## S3 method for class 'Spatial'  
mutate_(.data, ..., .dots)  
  
## S3 method for class 'Spatial'  
summarise_(.data, ...)  
  
## S3 method for class 'Spatial'  
summarise(.data, ...)  
  
## S3 method for class 'Spatial'  
group_by_(.data, ...)  
  
## S3 method for class 'Spatial'  
group_by(.data, ...)  
  
## S3 method for class 'Spatial'  
filter_(.data, ..., .dots)  
  
## S3 method for class 'Spatial'  
filter(.data, ...)  
  
## S3 method for class 'Spatial'  
arrange_(.data, ...)  
  
## S3 method for class 'Spatial'  
arrange(.data, ...)  
  
## S3 method for class 'Spatial'  
slice_(.data, ...)  
  
## S3 method for class 'Spatial'  
slice(.data, ...)  
  
## S3 method for class 'Spatial'  
select_(.data, ...)  
  
## S3 method for class 'Spatial'  
select(.data, ...)  
  
## S3 method for class 'Spatial'  
rename_(.data, ...)  
  
## S3 method for class 'Spatial'  
rename(.data, ...)
```

```
## S3 method for class 'Spatial'
distinct_(.data, ..., .keep_all = FALSE)

## S3 method for class 'Spatial'
distinct(.data, ..., .keep_all = FALSE)

## S3 method for class 'Spatial'
left_join(x, y, by = NULL, copy = FALSE, ...)

## S3 method for class 'Spatial'
inner_join(x, y, by = NULL, copy = FALSE, ...)
```

Arguments

<code>.data</code>	A tbl.
<code>...</code>	Name-value pairs of expressions. See <code>mutate_</code> .
<code>.dots</code>	Used to work around non-standard evaluation.
<code>.keep_all</code>	argument for <code>distinct</code> , we have to set it to TRUE
<code>x</code>	tbls to join
<code>y</code>	tbl to join
<code>by</code>	a character vector of variables to join by. If NULL, the default, <code>*_join()</code> will do a natural join, using all variables with common names across the two tables. A message lists the variables so that you can check they're right (to suppress the message, simply explicitly list the variables that you want to join). To join by different variables on x and y use a named vector. For example, <code>by = c("a" = "b")</code> will match x.a to y.b.
<code>copy</code>	If x and y are not from the same data source, and copy is TRUE, then y will be copied into the same src as x. This allows you to join tables across srcs, but it is a potentially expensive operation so you must opt into it.

Details

`mutate`, `transmute`, `filter`, `arrange`, `slice`, `select`, `rename`, `distinct` all work with attributes on the "data" slot and leave the geometry unchanged.

`summarise` collapses to a grouped geometries by listing all subgeometries together, it does not perform any topological union or merge, and it takes no account of the calculations done on attributes. This is a brutal collapse of all the data, and is identical to what is seen with `spplot(x, "group")`. The behaviour of geometric collapse like this is touch and go anyway, see the examples for a what `'rgeos::gUnion'` does.

`summarise` for points and multipoints, ... todo single Multipoint for multiple points

Warning

'distinct' uses behaviour identical to 'duplicated', by coercing all the relevant values to text and determining uniqueness from those. 'dplyr::distinct' uses a different internal method that will give different results for some cases of numeric data.

Note

Beware that attributes stored on Spatial objects *are not* linked to the geometry. Attributes are often used to store the area or perimeter length or centroid values but these may be completely unmatched to the underlying geometries.

Examples

```

library(sp)
library(maptools)
data(wrld_simpl)
library(spdplyr)
library(raster)
wrld_simpl %>% mutate(NAME = "allthesame", REGION = row_number())
wrld_simpl %>% transmute(alpha = paste0(FIPS, NAME))
wrld_simpl %>% filter(NAME %in% c("New Zealand", "Australia", "Fiji"))
## Not run:
wrld_simpl %>% arrange(LON)
wrld_simpl %>% slice(c(9, 100))
wrld_simpl %>% dplyr::select(UN, FIPS)
wrld_simpl %>% rename(`TM_WORLD_BORDERS_SIMPL0.2NAME` = NAME)
wrld_simpl %>% distinct(REGION, .keep_all = TRUE) %>%
  arrange(REGION) ## first alphabetically in REGION
wrld_simpl %>% arrange(REGION, desc(NAME)) %>% distinct(REGION, .keep_all = TRUE) ## last

## End(Not run)
## we don't need to use piping
slice(filter(mutate(wrld_simpl, likepiping = FALSE), abs(LON - 5) < 35 & LAT > 50), 4)

## works with Lines
#as(wrld_simpl, "SpatialLinesDataFrame") %>%
  # mutate(perim = rgeos::gLength(wrld_simpl, byid = TRUE))

## Not run:
## summarise/ze can be used after group_by, or without
wrld_simpl %>% filter(REGION == 150) %>% summarize(max(AREA))
wrld_simpl %>% group_by(REGION) %>% summarize(max(AREA)) %>%
plot(col = rainbow(nlevels(factor(wrld_simpl$REGION)), alpha = 0.3))

## End(Not run)
## group_by and summarize

## Not run:
g <- wrld_simpl %>% group_by(REGION) %>%
  summarize(alon = mean(LON), mxlat = max(LAT), mxarea = max(AREA))
g %>% mutate(ar = factor(REGION)) %>% spplot("ar")
w <- wrld_simpl
w$ar <- factor(w$REGION)
spplot(w, "ar")

## End(Not run)
## Not run:

```

```
# compare what rgeos gives
##splot(rgeos::gUnionCascaded(w, id = w$ar)) ## good grief, is this compelling . . .
## this is hardly a clean dissolve
##plot(rgeos::gUnionCascaded(w, id = w$ar), col = rainbow(nlevels(factor(w$ar))), alpha = 0.5))

## End(Not run)
```

mpoint1	<i>MultiPointsDataFrame data set</i>
---------	--------------------------------------

Description

MultiPointsDataFrame data set

show, Spatial-method *sp methods*

Description

Sp methods

Usage

```
## S4 method for signature 'Spatial'
show(object)

## S4 method for signature 'SpatialPoints'
show(object)
```

Arguments

object	Spatial object
--------	----------------

smap	<i>"South-east" map data.</i>
------	-------------------------------

Description

Created in /data-raw/ smap is a subset of wrld_simpl from maptools.

Examples

```
library(dplyr)
smap %>% filter(NAME == "Antarctica")
```

Index

arrange, 2
arrange.Spatial (dplyr-Spatial), 3
arrange_.Spatial (dplyr-Spatial), 3

distinct, 2, 5
distinct.Spatial (dplyr-Spatial), 3
distinct_.Spatial (dplyr-Spatial), 3
dplyr-S3, 3
dplyr-Spatial, 3

filter, 2
filter.Spatial (dplyr-Spatial), 3
filter_.Spatial (dplyr-Spatial), 3

group_by, 2
group_by.Spatial (dplyr-Spatial), 3
group_by_.Spatial (dplyr-Spatial), 3
groups.Spatial (dplyr-S3), 3

inner_join.Spatial (dplyr-Spatial), 3

left_join.Spatial (dplyr-Spatial), 3

mpoint1, 7
mutate, 2
mutate.Spatial (dplyr-Spatial), 3
mutate_, 5
mutate_.Spatial (dplyr-Spatial), 3

rename, 2
rename.Spatial (dplyr-Spatial), 3
rename_.Spatial (dplyr-Spatial), 3

select, 2
select.Spatial (dplyr-Spatial), 3
select_.Spatial (dplyr-Spatial), 3
show, Spatial-method, 7
show, SpatialPoints-method
 (show, Spatial-method), 7
slice, 2
slice.Spatial (dplyr-Spatial), 3

slice_.Spatial (dplyr-Spatial), 3
spdplyr-package, 2
spmap, 7
summarise, 2
summarise.Spatial (dplyr-Spatial), 3
summarise_.Spatial (dplyr-Spatial), 3

tbl_vars.Spatial (dplyr-S3), 3
transmute, 2