Package ‘tabularaster’

May 22, 2018

Type Package

Title Tidy Tools for ‘Raster’ Data

Version 0.5.0

Description Facilities to work with vector and raster data in efficient
repeatable and systematic work flow. Missing functionality in existing packages
is included here to allow extraction from raster data with 'simple features' and
'Spatial' types and to make extraction consistent and straightforward. Extract cell
numbers from raster data and return the cells, values and weights as a data frame
rather than as lists of matrices or vectors. The functions here allow spatial data
to be used without special handling for the format currently in use.

License GPL-3

LazyData TRUE

Depends R (>= 3.2.5)

Imports dplyr, methods, raster, sp, spatstat, spex (>= 0.4.0), spbabel
(>= 0.4.8), tibble, viridis, gibble, fasterize, magrittr, rlang

RoxygenNote 6.0.1

Suggests covr, testthat, knitr, rmarkdown, sf, spdplyr

VignetteBuilder knitr

URL https://github.com/hypertidy/tabularaster

BugReports https://github.com/hypertidy/tabularaster/issues

NeedsCompilation no

Author Michael D. Sumner [aut, cre]

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

Repository CRAN

Date/Publication 2018-05-21 22:44:03 UTC
as_tibble

Convert a Raster to a data frame.

Description

Generate a data frame version of any raster object. Use the arguments 'cell', 'dim', 'split_date' and 'value' to control the columns that are included in the output.

Usage

```r
## S3 method for class 'BasicRaster'
as_tibble(x, cell = TRUE, dim = n_layers(x) > 1L,
          value = TRUE, split_date = FALSE, xy = FALSE, ...)
```

Arguments

- **x**: a RasterLayer, RasterStack or RasterBrick
- **cell**: logical to include explicit cell number
- **dim**: logical to include slice index
- **value**: logical to return the values as a column or not
- **split_date**: logical to split date into components
- **xy**: logical to include the x and y centre coordinate of each cell
- **...**: unused

Details

If the raster has only one layer, the slice index is not added. Use 'dim = FALSE' to not include the slice index value.
bufext

Value

A data frame (‘tbl_df’/’tibble’ form)

Examples

```r
#library(tabularaster)
#library(tibble)
#as_tibble(raster::raster(volcano))
#as_tibble(raster::setZ(raster::raster(volcano), Sys.Date()), cell = TRUE)
```

---

bufext  Defunct tabularaster function

Description

Ensure a raster extent aligns to whole parts.

Usage

```r
bufext(e1, e2)
```

Arguments

- `e1` input extent
- `e2` grain size

Details

This function is defunct, please use spex::buffer_extent.

Examples

```r
library(spex)
library(raster)
buffer_extent(extent(0.1, 2.2, 0, 3), 2)
```
cellnumbers

Extract cell numbers from a Raster object.

Description

Provide the 'cellnumbers' capability of \texttt{raster::extract} and friends directly, returning a data frame of query-object identifiers 'object_' and the cell number.

Usage

\begin{verbatim}
cellnumbers(x, query, ...)  
\end{verbatim}

\texttt{## Default S3 method:}
cellnumbers(x, query, ...)

\texttt{## S3 method for class 'SpatialLines'}
cellnumbers(x, query, ...)

\texttt{## S3 method for class 'sfc'}
cellnumbers(x, query, ...)

\texttt{## S3 method for class 'sf'}
cellnumbers(x, query, ...)

Arguments

\begin{itemize}
  \item \texttt{x} Raster object
  \item \texttt{query} Spatial object or matrix of coordinates
  \item \texttt{...} unused
\end{itemize}

Details

Raster data is inherently 2-dimensional, with a time or 'level' dimension treated like a layers of these 2D forms. The 'raster' package cell number is counted from 1 at the top-left, across the rows and down. This corresponds the the standard "raster graphics" convention used by 'GDAL' and the 'sp' package, and many other implementations. Note that this is different to the convention used by the \texttt{graphics::image} function.

Currently this function only operates as if the input is a single layer objects, it's not clear if adding an extra level of grouping for layers would be sensible.

Value

\texttt{tbl_df} data frame
Examples

```r
library(raster)
library(dplyr)

r <- raster(volcano) %>% aggregate(fact = 4)
cellnumbers(r, rasterToContour(r, level = 120))
#library(dplyr)

#cr <- cut(r, pretty(values(r)))
#p <- raster::rasterToPolygons(cr, dissolve = TRUE)
#p <- spex::qm_rasterToPolygons_sp(cr)
#tt <- cellnumbers(cr, p)
#library(dplyr)
#tt %>% mutate(v = extract(r, cell_)) %>%
#group_by(object_) %>%
#summarize(mean(v))
#head(pretty(values(r)), -1)
```

ghrsst  

*Sea surface temperature data.*

Description

A smoothed subset of GHRST.

Format

A raster created GHRST data and raster smoothing.

Details


*sst_regions* is a simple polygon region layer to sit over the SST data.

Examples

```r
library(raster)
plot(ghrsst, col = viridis::viridis(100))
plot(sst_regions, add = TRUE, col = NA)
## cellnumbers(ghrsst, sst_regions)
```
<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>index_extent</strong></td>
<td><strong>Index extent</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Description</strong></td>
<td>Extent in index space.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage</strong></td>
<td>index_extent(x, ex)</td>
</tr>
<tr>
<td></td>
<td><strong>Arguments</strong></td>
<td>x raster layer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ex extent</td>
</tr>
<tr>
<td></td>
<td><strong>Details</strong></td>
<td>Convert a geographic extent into purely index space.</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
<td>extent object</td>
</tr>
<tr>
<td>oisst</td>
<td><strong>Optimally interpolated SST in near-native form.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Description</strong></td>
<td>See data-raw/oisst.R in the source repository. The file was avhrr-only-v2.20170729.nc, its extent -180, 180, -90, 90 with dimensions 1440x720 in the usual raster configuration.</td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong></td>
<td>A data frame of sst values created from OISST data.</td>
</tr>
<tr>
<td>polycano</td>
<td><strong>The raster volcano as polygons.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Description</strong></td>
<td>See data-raw/rastercano.r in the source repository.</td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong></td>
<td>A sp::SpatialPolygonsDataFrame with variables: volcano_elevation.</td>
</tr>
</tbody>
</table>
raster-exports      Re-exports from raster

Description
These functions are imported and then re-exported.

Details
See raster::extentFromCells for details.

rastercano       The raster volcano.

Description
See data-raw/rastercano.r in the source repository.

Format
A raster created from the volcano data.

sharkcano       Sharkcano, the shark and the volcano.

Description
This is just a free image off the internetz. The image was read in and all non-essential items dropped. The dimensions in raster::raster terms is stored in attr(sharkcano, "rasterdim").

Format
A data frame with 117843 rows and 2 variables:
cell_     integer, cell index
byte      integer, byte value of shark image pixels
These are cell values on a grid that is 648x958.

References
This is the small version from here, see script in data-raw/sharkcano.r http://www.freestockphotos.biz/stockphoto/16214 Thanks to @jennybc for pointers on finding free stuff: https://github.com/jennybc/free-photos
Examples

```r
library(raster)
rd <- attr(sharkcano, "rasterdim")
rastershark <- raster(matrix(NA_integer_, rd[1], rd[2]))
rastershark[sharkcano$cell_] <- sharkcano$byte  ## byte, heh
## I present to you, Sharkcano! (Just wait for the 3D version, Quadshark).
#plot(rastercano)
#contour(rastershark, add = TRUE, labels = FALSE)
#plot(rastershark, col = "black")
## another way
#plot(rastercano)
#points(xyFromCell(rastershark, sharkcano$cell_), pch = ".")
```

---

**tabularaster**

**Tabular tools for raster, tidy tools for raster.**

---

Description

Tabular tools for raster, tidy tools for raster.

Details

Tabularaster includes these main functions.

- **as_tibble**
  - convert raster data to data frame form, with control over output and form of dimension/coordinate columns

- **cellnumbers**
  - extract a data frame of query identifiers and cell,pixel index numbers

- **extentFromCells**
  - (now incorporated into raster itself)

- **index_extent**
  - build an extent in row column form, as opposed to coordinate value form
Index

as_tibble, 2, 8
bufext, 3
cellnumbers, 4, 8
extent, 3
extentFromCells, 7, 8
extentFromCells (raster-exports), 7
ghrsst, 5
graphics::image, 4
index_extent, 6, 8
oisst, 6
polycano, 6
raster-exports, 7
raster::extract, 4
rastercano, 7
sharkcano, 7
sst_regions (ghrsst), 5
tabularaster, 8
tabularaster-package (tabularaster), 8
volcano, 7