Package 'unheadr'

March 4, 2020

Type Package

Title Handle Data with Messy Header Rows and Broken Values

Version 0.2.1

Depends R (>= 2.10)

Description Verb-like functions to work with messy data, often derived from spreadsheets or parsed PDF tables. Includes functions for unwrapping values broken up across rows, relocating embedded grouping values, and to annotate meaningful formatting in spreadsheet files.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr (>= 0.7.5), rlang (>= 0.2.1), forcats, stringr, tidyr, magrittr, tidyxl, readxl, tibble

RoxygenNote 7.0.2

Suggests knitr, rmarkdown, testthat (>= 2.1.0), covr

VignetteBuilder knitr

URL https://github.com/luisDVA/unheadr, https://unheadr.liomys.mx/

BugReports https://github.com/luisDVA/unheadr/issues

NeedsCompilation no

Author Luis D. Verde Arregoitia [aut, cre] (<https://orcid.org/0000-0001-9520-6543>)

Maintainer Luis D. Verde Arregoitia <luis@liomys.mx>

Repository CRAN

Date/Publication 2020-03-04 01:00:02 UTC

R topics documented:

`annotate_mf` *Annotate meaningful formatting*

Description

Annotate meaningful formatting

Usage

```
annotate_mf(xlfilepath, orig, new)
```

Arguments

`xlfilepath`	Path to spreadsheet file (xls or xlsx).
`orig`	Variable to annotate formatting in.
`new`	Name of new variable with cell formatting pasted as a string.

Details

At this point, only four popular approaches for meaningful formatting (bold, italic, underline, cell highlighting) are hardcoded in the function. The HTML code of the fill color used for cell highlighting is also appended in the output.

Value

A tibble with a new column with meaningful formatting embedded as text.

Examples

```
example_spreadsheet <- system.file("extdata/dog_test.xlsx", package = "unheadr")
annotate_mf(example_spreadsheet, orig = Task, new = Task_annotated)
```

dog_test *dog_test.xlsx spreadsheet*

Description

Open XML Format Spreadsheet with 1 sheet, 2 columns, and 12 rows. Items describe various taks or behaviors that dogs can be evaluated on, assigned into three categories which appear along with their average scores as embedded subheaders with meaningful formatting.

dog_test.xlsx

This data is used in the example for 'annotate_mf()'.

Source

Items are modified from the checklist written by Junior Watson.

References

http://www.dogtrainingbasics.com/checklist-well-behaved-dog/

primates2017 *Comparative data for 54 species of primates*

Description

A dataset with embedded subheaders.

Usage

```
primates2017
```

Format

A data frame with 69 rows and 4 variables:

scientific_name scientific names, with geographic region and taxonomic family embedded as subheaders.

common_name vernacular name, as listed in Estrada et al. (2017)

red_list_status IUCN Red List Status in January 2017

mass_kg mean body mass in kilograms

Source

Estrada, Alejandro, et al. "Impending extinction crisis of the world's primates: Why primates matter." Science Advances 3.1 (2017): e1600946. http://advances.sciencemag.org/content/3/1/e1600946.full

primates2017_broken *Comparative data for 16 species of primates with some broken values*

Description

A dataset with embedded subheaders and some values (T. obscurus, T. leucocephalus and N. bengalensis) in the scientific_names variable broken up across two rows (typically done to fit the content in a table).

Usage

```
primates2017_broken
```

Format

A data frame with 19 rows and 4 variables:

scientific_name scientific names, with embedded subheaders for geographic region and taxonomic family and broken values

common_name vernacular name, as listed in Estrada et al. (2017)

red_list_status IUCN Red List Status in January 2017

mass_kg mean body mass in kilograms

Source

Estrada, Alejandro, et al. "Impending extinction crisis of the world's primates: Why primates matter." Science Advances 3.1 (2017): e1600946. http://advances.sciencemag.org/content/3/1/e1600946.full

primates2017_wrapped *Comparative data for two species of primates*

Description

A dataset in which the elements for some of the values are in separate rows'

Usage

```
primates2017_wrapped
```

Format

A data frame with 9 rows and 6 variables:

scientific_name scientific names, see reference

common_name vernacular name, as listed in Estrada et al. (2017)

habitat habitat types listed in the IUCN Red List assessments

red_list_status IUCN Red List Status in January 2017

mass_kg mean body mass in kilograms

country Countries where the species is present, from IUCN Red List assessments

Source

Estrada, Alejandro, et al. "Impending extinction crisis of the world's primates: Why primates matter." Science Advances 3.1 (2017): e1600946. http://advances.sciencemag.org/content/3/1/e1600946.full

---

unbreak_rows *Merge rows up*

Description

Merge rows up

Usage

```
unbreak_rows(df, regex, ogcol, sep = " ")
```

Arguments

df	A data frame with at least two contiguous rows to be merged.
regex	A regular expression to identify sets of rows to be merged, meant for the leading of the two contiguous rows.
ogcol	Variable with the text strings to match.
sep	Character string to separate the unified values (default is space).

Value

A tibble or data frame with merged rows. Values of the lagging rows are pasted onto the values in the leading row, whitespace is squished, and the lagging row is dropped.

Examples

```
bball <-
  data.frame(
    stringsAsFactors = FALSE,
    v1 = c(
      "Player", NA, "Sleve McDichael", "Dean Wesrey",
      "Karl Dandleton"
    ),
    v2 = c("Most points", "in a game", "55", "43", "41"),
    v3 = c("Season", "(year ending)", "2001", "2000", "2010")
  )
unbreak_rows(bball, "Most", v2)
```

`unbreak_vals` *Unbreak values using regex to match the broken half of the value*

Description

Unbreak values using regex to match the broken half of the value

Usage

```
unbreak_vals(df, regex, ogcol, newcol, sep = " ", .slice_groups = FALSE)
```

Arguments

`df`	A data frame with one or more values within a variable broken up across two rows.
`regex`	Regular expression for matching the second half of the broken values.
`ogcol`	Variable to unbreak.
`newcol`	Name of the new variable with the unified values.
`sep`	Character string to separate the unified values (default is space).
`.slice_groups`	When '.slice_groups = FALSE' (the default), the extra rows and the variable with broken values will not be dropped.

Details

This function is limited to quite specific cases, but useful when dealing with tables that contain scientific names broken across two rows. For unwrapping values, see `unwrap_cols`.

Value

A tibble with unbroken values. The variable that originally contained the broken values gets dropped, and the new variable with the unified values is placed as the first column.

Examples

```
data(primates2017_broken)
# regex matches strings starting in lowercase (broken species epithets)
unbreak_vals(primates2017_broken, "^[a-z]", scientific_name, sciname_new)
```

`untangle2` *Rectangling embedded subheaders*

Description

Rectangling embedded subheaders

Usage

```
untangle2(df, regex, orig, new)
```

Arguments

`df`	A data frame with embedded subheaders.
`regex`	Regular expression to match the subheaders.
`orig`	Variable containing the extraneous subheaders.
`new`	Name of variable that will contain the group values.

Details

Special thanks to Jenny Bryan for fixing the initial tidyeval code and overall function structure.

Value

A tibble without the matched subheaders and a new variable containing the grouping data.

Examples

```
data(primates2017)
# put taxonomic family in its own variable (matches the suffix "DAE")
untangle2(primates2017, "DAE$", scientific_name, family)
# put geographic regions in their own variable (matching them all by name)
untangle2(primates2017, "Asia|Madagascar|Mainland Africa|Neotropics",
          scientific_name, family)
# with magrittr pipes (re-exported in this package)
primates2017 %>%
  untangle2("DAE$", scientific_name, family) %>%
  untangle2("Asia|Madagascar|Mainland Africa|Neotropics",
            scientific_name, region)
```

unwrap_cols *Unwrap values and clean up NAs used as padding*

Description

Unwrap values and clean up NAs used as padding

Usage

```
unwrap_cols(df, groupingVar, separator)
```

Arguments

df	A data frame with wrapped values and an inconsistent number of NA values used to as within-group padding.
groupingVar	Name of the variable describing the observational units.
separator	Character string defining the separator that will delimit the elements of the unwrapped value.

Details

This is roughly the opposite of 'tidyr::separate_rows()'.

Value

A summarized tibble. Order is preserved in the grouping variable by making it a factor.

Examples

```
data(primates2017_wrapped)
# using commas to separate elements
unwrap_cols(primates2017_wrapped, scientific_name, ", ")

# separating with semicolons
df <- data.frame(
  ounits = c("A", NA, "B", "C", "D", NA),
  vals = c(1, 2, 2, 3, 1, 3)
)
unwrap_cols(df, ounits, ";")
```

%>% *re-export magrittr pipe operator*

Description

re-export magrittr pipe operator

Index