

Package ‘stoRy’

June 11, 2020

Type Package

Title Functions for the Analysis of Star Trek Thematic Data

Version 0.1.5

Date 2020-06-11

Author Paul Sheridan, Mikael Onsjö

Maintainer Paul Sheridan <paul.sheridan.stats@gmail.com>

Description An implementation of 1) the hypergeometric test for over-representation of literary themes in a storyset (a list of stories) relative to a background list of stories, and 2) a recommendation system that takes a user-selected story as input and returns a ranked list of similar stories on the basis of shared themes. The package is currently implemented for the episodes of the Star Trek television franchise series The Original Series (TOS), The Animated Series (TAS), The Next Generation (TNG), and Voyager (VOY).

License GPL-3

Depends R(>= 3.5.0)

Imports R6, data.tree

LazyData True

Encoding UTF-8

NeedsCompilation no

Suggests knitr

VignetteBuilder knitr

RoxygenNote 7.1.0

Repository CRAN

Date/Publication 2020-06-11 05:50:11 UTC

R topics documented:

get_enriched_themes	2
get_similar_stories	3
print_tree	5
story	5

storysets	8
story_ids_by_theme	9
story_keywords	10
story_metadata	10
story_settings	10
story_themes	11
theme	11
theme_dict	12

Index 13

get_enriched_themes	<i>Find enriched themes in a storyset of interest relative to a background storyset.</i>
---------------------	--

Description

get_enriched_themes calculates enrichment scores for themes in a storset relative to a background set of stories according to the hypergeometric test.

Usage

```
get_enriched_themes(
  mystorysets,
  test_storysets = mystorysets$names,
  background_storyset = c("tos", "tas", "tng", "voy"),
  theme_levels = c("central", "peripheral"),
  min_storyset_size = 5,
  min_theme_occurrence = 1
)
```

Arguments

mystorysets	A storysets class object that contains one or many individual storysets.
test_storysets	The storyset names to be analysed. The default mystorysets\$names is to analyse all storysets.
background_storyset	default c("tos", "tas", "tng", "voy"). A string indicating the storyset to use as background. Possible values are c("tos", "tas"), "tng", "voy", and c("tos", "tas", "tng", "voy")
theme_levels	c("central", "peripheral"). A string indicating the theme levels to use in the analysis. Possible values are central and/or peripheral.
min_storyset_size	default 5. The minimum allowable size for a storyset. For small samples the hypergeometric test may be unreliable.
min_theme_occurrence	default 1. The minimum number of times a theme must occur in a storyset. For small samples the hypergeometric test may be unreliable.

Value

Returns a data frame where each row corresponds to a theme. The column Pvalue contains raw enrichment scores, i.e., hypergeometric test P-values.

References

Onsjö, M., and Sheridan, P. (2017): "Theme Enrichment Analysis: A Statistical Test for Identifying Significantly Enriched Themes in a List of Stories with an Application to the Star Trek Television Franchise", ArXiv.

Examples

```
#####
# Load the Star Trek series storysets smt file and storysets object #
#####
file <- system.file("storysets", "series.smt", package = "stoRy")
mystorysets <- storysets$new(file)
print(mystorysets)

#####
# Perform the theme enrichment analysis for each of TOS, TAS, and TNG #
# relative to default TOS/TAS/TNG background #
#####
results <- get_enriched_themes(mystorysets)

#####
# Output top twenty enriched TOS themes #
#####
results$TOS[1:20,]

#####
# Output top ten enriched TAS themes #
#####
results$TAS[1:10,]

#####
# Output top twenty enriched TNG themes #
#####
results$TNG[1:20,]
```

get_similar_stories *Recommend stories similar to a user-selected one.*

Description

get_similar_stories returns a list of stories, sorted by thematic similarity with respect to a user-selected story

print_tree	<i>Theme hierarchy tree view.</i>
------------	-----------------------------------

Description

print_tree Print a theme and its descendents in a tree format.

Usage

```
print_tree(mytheme, pruneMethod = c("simple", "dist"), limit = 100)
```

Arguments

mytheme	A pkgstoRy package theme object.
pruneMethod	default c("simple", "dist"). The "Simple" option is the default and prints up to limit themes as they appear in order in the tree. The "dist" option prints the more upper level themes in the hierarchy.
limit	The maximum number of themes to be printed. The default value is 100.

Value

Given a theme object as input, print_tree prints a tree structure of its descendent themes.

Examples

```
#####
# Display the "the human condition" theme hierarchy in tree format #
#####
theme_name <- "the human condition"
mytheme <- theme$new(theme_name)
print_tree(mytheme, pruneMethod = "dist", limit = 50)
```

story	<i>Story objects</i>
-------	----------------------

Description

The **stoRy** package uses the story object to store story themes together with other metadata.

Fields

Each story object has the following ten fields

story_id A story ID. See examples.

title A string giving the story title.

writer A string giving the story writer.

director A string giving the story director.

air_date A string giving the story original air date in the for YYYY-MM-DD

summary A string giving a summary of the story.

characters A list of story characters. The list has three fields: ObjectCharacters, MajorCharacters, and MinorCharacters. Each list entry should be a ", " separated string of character names.

themes A data frame of story themes with associated metadata.

settings A data frame of story settings with associated metadata.

keywords A data frame of story keywords with associated metadata.

Methods

Public methods:

- `story$new()`
- `story$add_theme()`
- `story$remove_theme()`
- `story$add_setting()`
- `story$remove_setting()`
- `story$add_keyword()`
- `story$remove_keyword()`
- `story$print()`
- `story$clone()`

Method `new()`:

Usage:

```
story$new(
  story_id,
  title,
  writer,
  director,
  air_date,
  summary,
  characters,
  themes,
  settings,
  keywords
)
```

Method `add_theme()`:

Usage:

```
story$add_theme(  
  theme,  
  level,  
  comment,  
  related_characters,  
  character_class,  
  related_aliases,  
  related_things  
)
```

Method remove_theme():

Usage:

```
story$remove_theme(theme)
```

Method add_setting():

Usage:

```
story$add_setting(setting, capacity)
```

Method remove_setting():

Usage:

```
story$remove_setting(setting)
```

Method add_keyword():

Usage:

```
story$add_keyword(keyword, comment, timing, parent_keyword)
```

Method remove_keyword():

Usage:

```
story$remove_keyword(keyword)
```

Method print():

Usage:

```
story$print(...)
```

Method clone(): The objects of this class are cloneable with this method.

Usage:

```
story$clone(deep = FALSE)
```

Arguments:

deep Whether to make a deep clone.

Examples

```
#####
# Create a story object for the Star Trek The Original Series episode #
# tos1x19 "The Arena" and manipulate it in various ways #
#####
story_id <- "tos1x19"
mystory <- story$new(story_id)
print(mystory)

#####
# Add "neo-luddist utopia" as a central theme #
#####
mystory$add_theme(theme = "neo-luddist utopia", level = "central")

#####
# Remove "neo-luddist utopia" as a central theme #
#####
mystory$remove_theme(theme = "neo-luddist utopia")

#####
# Add "mountain" as a setting #
#####
mystory$add_setting(setting = "mountain")

#####
# Add a new keyword #
#####
mystory$add_keyword(keyword = "Captain Kirk is climbing a mountain")
```

storysets

Storysets objects

Description

The **stoRy** package uses the `storysets` object to store lists of story IDs.

Fields

Each `storysets` object has the following five fields

file A string specifying the name of an `.smt` format `storysets` file. See examples.

names A vector of `storyset` names.

comments A vector of comments describing the `storysets`.

no_of_storysets The number of `storysets` in the `.smt` file.

storysets_story_ids A list of vectors containing the `storyset` story IDs.

Methods**Public methods:**

- `storysets$new()`
- `storysets$print()`
- `storysets$clone()`

Method new():*Usage:*`storysets$new(file)`**Method print():***Usage:*`storysets$print(...)`**Method clone():** The objects of this class are cloneable with this method.*Usage:*`storysets$clone(deep = FALSE)`*Arguments:*`deep` Whether to make a deep clone.**Examples**

```
#####
# List all storysets files available in the \pkg{stoRy} package      #
#####
list.files(system.file("storysets", package = "stoRy"))

#####
# Load the Star Trek aliens storysets smt file and storysets object #
#####
file <- system.file("storysets", "aliens.smt", package = "stoRy")
mystorysets <- storysets$new(file)
print(mystorysets)
```

story_ids_by_theme	<i>Story theme summary</i>
--------------------	----------------------------

Description

A table in which each theme is summarised according to the stories in which it is featured.

Format

Data frame

Source

Theme Ontology. See <http://www.themeontology.org> for more details.

story_keywords	<i>Story keywords</i>
----------------	-----------------------

Description

Miscellaneous keywords for the episodes of Star Trek The Original Series (tos), The Animated Series (tas), The Next Generation (tng), and Voyager (voy).

Format

Data frame

Source

Theme Ontology. See <http://www.themeontology.org> for more details.

story_metadata	<i>Story metadata</i>
----------------	-----------------------

Description

Metadata for the episodes of Star Trek The Original Series (tos), The Animated Series (tas), The Next Generation (tng), and Voyager (voy).

Format

Data frame

Source

Theme Ontology. See <http://www.themeontology.org> for more details.

story_settings	<i>Story settings</i>
----------------	-----------------------

Description

Settings for the episodes of Star Trek The Original Series (tos), The Animated Series (tas), The Next Generation (tng), and Voyager (voy).

Format

Data frame

Source

Theme Ontology. See <http://www.themeontology.org> for more details.

story_themes	<i>Themed stories</i>
--------------	-----------------------

Description

Themes for the episodes of Star Trek The Original Series (tos), The Animated Series (tas), The Next Generation (tng), and Voyager (voy).

Format

Data frame

Source

Theme Ontology. See <http://www.themeontology.org> for more details.

theme	<i>Theme objects</i>
-------	----------------------

Description

The **stoRy** package uses the theme object to store information about themes in the package theme dataset.

Methods**Public methods:**

- `theme$new()`
- `theme$print()`
- `theme$clone()`

Method new():

Usage:

`theme$new(theme)`

Arguments:

theme A string representing a theme in the dataset.

Method print():

Usage:

`theme$print(...)`

Method clone(): The objects of this class are cloneable with this method.

Usage:

`theme$clone(deep = FALSE)`

Arguments:

deep Whether to make a deep clone.

Examples

```
#####
# Create a theme object for the theme "utopia" and print the result #
#####
theme_name <- "utopia"
mytheme <- theme$new(theme_name)
mytheme$print()

#####
# Display the "utopia" theme and its descendent themes in tree format #
#####
print_tree(mytheme)

#####
# Display the "society" theme hierarchy in tree format #
#####
theme_name <- "society"
mytheme <- theme$new(theme_name)
print_tree(mytheme, pruneMethod = "dist", limit = 50)
```

 theme_dict

Theme dictionary

Description

A hierarchical structured vocabulary of defined themes in four domains: the human condition, society, alternate reality, and the pursuit of knowledge.

Format

Data frame

Source

Theme Ontology. See <http://www.themeontology.org> for more details.

Index

`get_enriched_themes`, [2](#)
`get_similar_stories`, [3](#)

`print_tree`, [5](#)

`story`, [5](#)
`story_ids_by_theme`, [9](#)
`story_keywords`, [10](#)
`story_metadata`, [10](#)
`story_settings`, [10](#)
`story_themes`, [11](#)
`storysets`, [8](#)

`theme`, [11](#)
`theme_dict`, [12](#)