Package ‘veccompare’

September 15, 2017

Type Package

Title Perform Set Operations on Vectors, Automatically Generating All n-Wise Comparisons, and Create Markdown Output

Version 0.1.0

Date 2017-09-13

Maintainer Jacob Gerard Levernier <jlevern@upenn.edu>

Description Automates set operations (i.e., comparisons of overlap) between multiple vectors. It also contains a function for automating reporting in 'RMarkdown', by generating markdown output for easy analysis, as well as an 'RMarkdown' template for use with 'RStudio'.

License BSD_3_clause + file LICENSE

URL https://github.com/publicus/r-veccompare

BugReports https://github.com/publicus/r-veccompare/issues

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports corrplot, gtools, grid, pander, purrr, reshape2, qgraph, VennDiagram

Suggests devtools, testthat

RoxygenNote 6.0.1.9000

NeedsCompilation no

Author Jacob Gerard Levernier [aut, cre] (Designed and authored the package source code and documentation. Roles: author, creator, designer, engineer, programmer),
Heather Gaile Wacha [aut] (Provided intellectual overview and consultation during development for use with medieval cartographic datasets. Roles: conceptor, consultant, data contributor)

Repository CRAN

Date/Publication 2017-09-15 10:42:38 UTC

R topics documented:

`veccompare-package` *veccompare: Automatically Generate All n-Wise Set Comparisons on Vectors*

Description

The **veccompare** package contains functions for automating set operations. Given a named list of 5 vectors, for example, **veccompare** can calculate all 2-, 3-, 4-, and 5-way comparisons between those vectors, recording information for each comparison about the set "union" (combined elements), "intersection" (overlap / shared elements), and compliments (which elements are unique to each vector involved in the comparison).

Details

The veccompare package contains functions for automating set operations (i.e., comparisons of overlap) between multiple vectors.

The package also contains a function for automating reporting in RMarkdown, by generating markdown output for easy analysis, as well as an RMarkdown template for use with RStudio.

The primary function from **veccompare** is `compare.vectors`. Complementarily, `compare.vectors.and.return.text.ana` will call `compare.vectors` and generate Markdown-style output from it (for example, for use within an RMarkdown file).

An RMarkdown template illustrating several of **veccompare**'s features can be used from within RStudio by clicking `File -> New File -> R Markdown... -> From Template -> Veccompare Overlap Report`.

veccompare also provides a function, `summarize.two.way.comparisons.percentage.overlap`, that can create correlation-plot-style images and network graphs for all two-way comparisons between vectors. This function is also demonstrated in the `Veccompare Overlap Report` described above.

Author(s)

Maintainer: Jacob Gerard Levernier <jlevern@upenn.edu> (Designed and authored the package source code and documentation. Roles: author, creator, designer, engineer, programmer)

Authors:

- Heather Gaile Wacha <wacha2@wisc.edu> (Provided intellectual overview and consultation during development for use with medieval cartographic datasets. Roles: conceptor, consultant, data contributor)

See Also

Useful links:

- https://github.com/publicus/r-veccompare
- Report bugs at https://github.com/publicus/r-veccompare/issues

---

compare.vectors *Compare all combinations of vectors using set operations*

Description

Compare all combinations of vectors using set operations

Usage

```
compare.vectors(named_list_of_vectors_to_compare,
  degrees_of_comparison_to_include = NULL, draw_venn_diagrams = FALSE,
  vector_colors_for_venn_diagrams = NULL, save_venn_diagram_files = FALSE,
  location_for_venn_diagram_files = "", prefix_for_venn_diagram_files = "",
  saved_venn_diagram_resolution_ppi = 300,
  saved_venn_diagram_dimension_units = "in", saved_venn_diagram_width = 8,
  saved_venn_diagram_height = 6, viewport_npc_width_height_for_images = 1,
  suppress_messages = FALSE)
```

Arguments

`named_list_of_vectors_to_compare`
: A named list of vectors to compare (see, for example, `example.vectors.list`). Duplicate values in a given vector will only be counted once (for example, c("a", "a", "b", "c") will be treated identically to c("a", "b", "c").

`degrees_of_comparison_to_include`
: A number or vector of numbers of which degrees of comparison to print (for example, 'c(2, 5)' would print only 2- and 5-way vector comparisons).

`draw_venn_diagrams`
: A logical (TRUE/FALSE) indicator whether to draw Venn diagrams for all 2- through 5-way comparisons of vectors.

`vector_colors_for_venn_diagrams`
: An optional vector of color names for Venn diagrams (if `draw_venn_diagrams` is `TRUE`). Color names are applied to the named vectors in `named_list_of_vectors_to_compare` in their order in `named_list_of_vectors_to_compare`. If this is blank, a random color will be selected for each vector. Either way, each vector will have a consistent color across the Venn diagrams in which it appears.

`save_venn_diagram_files`
: A logical (TRUE/FALSE) indicator whether to save Venn diagrams as PNG files.

`location_for_venn_diagram_files`
: An optional string giving a directory into which to save Venn diagram PNG files (if `save_venn_diagram_files` is `TRUE`). This location must already exist on the filesystem.

`prefix_for_venn_diagram_files`
: An optional string giving a prefix to prepend to saved Venn diagram PNG files (if `save_venn_diagram_files` is `TRUE`).

`saved_venn_diagram_resolution_ppi`
: An optional number giving a resolution (PPI) for saved Venn diagrams (if `save_venn_diagram_files` is `TRUE`).

`saved_venn_diagram_dimension_units`
: An optional string giving units for specifying `saved_venn_diagram_width` and `saved_venn_diagram_height` (if `save_venn_diagram_files` is `TRUE`). Can be `px` (pixels), `in` (inches, the default), `cm`, or `mm`.

`saved_venn_diagram_width`
: The width (in `saved_venn_diagram_dimension_units` units) for saved Venn diagrams (if `save_venn_diagram_files` is `TRUE`).

`saved_venn_diagram_height`
: The height (in `saved_venn_diagram_dimension_units` units) for saved Venn diagrams (if `save_venn_diagram_files` is `TRUE`).

`viewport_npc_width_height_for_images`
: The scale at which to print an image. If the image is cut off at its edges, for example, this can be set lower than 1.0.

`suppress_messages`
: A logical (TRUE/FALSE) indicator whether to suppress messages. Even if this is `TRUE`, warnings will still be printed.

Value

A list, with one object for each comparison of vectors. The list contains the following elements:

elements_involved The vector names involved in the comparison.

union_of_elements A vector of all (deduplicated) items involved in the comparison, across all of the vectors.

overlap_of_elements A vector of the deduplicated elements that occurred in all of the compared vectors.

elements_unique_to_first_element This element will have a sub-element named for each vector being compared (i.e., for each of the names in `$elements_involved`). The (deduplicated) items that were unique to that vector (i.e., not overlapping with any other vector in the comparison).

venn_diagram If `save_venn_diagram_files` is `TRUE`, and the comparison is of 2 through 5 vectors, a Venn diagram object produced using the **VennDiagram** package. This diagram can be rendered using `render.venn.diagram`.

To compile this list object into a Markdown report, use `compare.vectors.and.return.text.analysis.of.overlap`. For an example of this usage, see the `Veccompare Overlap Report` RMarkdown template for RStudio that is installed as part of the **veccompare** package.

Examples

```
example <- veccompare::compare.vectors(veccompare::example.vectors.list)

# To extract similar elements across list items:
veccompare::extract.compared.vectors(
  example,
  elements_of_output = "elements_involved"
)

# To extract all comparisons that involve "vector_a":
veccompare::extract.compared.vectors(
  example,
  vector_names = "vector_a"
)

# To find all comparisons that were about "vector_a" and "vector_c":
veccompare::extract.compared.vectors(
  example,
  vector_names = c("vector_a", "vector_c"),
  only_match_vector_names = TRUE
)

# To get all elements that did a two-way comparison:
veccompare::extract.compared.vectors(
  example,
  degrees_of_comparison = 2
)
```

---

compare.vectors.and.return.text.analysis.of.overlap

Create a Markdown report from the output of `compare.vectors`

Description

This function is a wrapper for `compare.vectors`. It creates a Markdown report of all degrees of set comparisons between a named list of vectors.

Usage

```
compare.vectors.and.return.text.analysis.of.overlap(named_list_of_vectors_to_compare,
 degrees_of_comparison_to_include = NULL, cat_immediately = FALSE,
 draw_venn_diagrams = FALSE, viewport_npc_width_height_for_images = 1,
 vector_colors_for_venn_diagrams = NULL, save_venn_diagram_files = FALSE,
 location_for_venn_diagram_files = "", prefix_for_venn_diagram_files = "",
 saved_venn_diagram_resolution_ppi = 300,
 saved_venn_diagram_dimension_units = "in", saved_venn_diagram_width = 8,
 saved_venn_diagram_height = 6, base_heading_level_to_use = 1)
```

Arguments

`named_list_of_vectors_to_compare`
: A named list of vectors to compare (see, for example, `example.vectors.list`). Duplicate values in a given vector will only be counted once (for example, c("a", "a", "b", "c") will be treated identically to c("a", "b", "c").

`degrees_of_comparison_to_include`
: A number or vector of numbers of which degrees of comparison to print (for example, 'c(2, 5)' would print only 2- and 5-way vector comparisons).

`cat_immediately`
: A logical (TRUE/FALSE) indicator whether to immediately print the output, as in an RMarkdown document.

`draw_venn_diagrams`
: A logical (TRUE/FALSE) indicator whether to draw Venn diagrams for all 2- through 5-way comparisons of vectors.

`viewport_npc_width_height_for_images`
: The scale at which to print an image. If the image is cut off at its edges, for example, this can be set lower than 1.0.

`vector_colors_for_venn_diagrams`
: An optional vector of color names for Venn diagrams (if `draw_venn_diagrams` is `TRUE`). Color names are applied to the named vectors in `named_list_of_vectors_to_compare` in their order in `named_list_of_vectors_to_compare`. If this is blank, a random color will be selected for each vector. Either way, each vector will have a consistent color across the Venn diagrams in which it appears.

`save_venn_diagram_files`
: A logical (TRUE/FALSE) indicator whether to save Venn diagrams as PNG files.

`location_for_venn_diagram_files`
: An optional string giving a directory into which to save Venn diagram PNG files (if `save_venn_diagram_files` is `TRUE`). This location must already exist on the filesystem.

`prefix_for_venn_diagram_files`
: An optional string giving a prefix to prepend to saved Venn diagram PNG files (if `save_venn_diagram_files` is `TRUE`).

`saved_venn_diagram_resolution_ppi`
: An optional number giving a resolution (PPI) for saved Venn diagrams (if `save_venn_diagram_files` is `TRUE`).

`saved_venn_diagram_dimension_units`
: An optional string giving units for specifying `saved_venn_diagram_width` and `saved_venn_diagram_height` (if `save_venn_diagram_files` is `TRUE`). Can be `px` (pixels), `in` (inches, the default), `cm`, or `mm`.

`saved_venn_diagram_width`
: The width (in `saved_venn_diagram_dimension_units` units) for saved Venn diagrams (if `save_venn_diagram_files` is `TRUE`).

`saved_venn_diagram_height`
: The height (in `saved_venn_diagram_dimension_units` units) for saved Venn diagrams (if `save_venn_diagram_files` is `TRUE`).

`base_heading_level_to_use`
: An integer indicating the highest-level heading to print. Defaults to `1` (i.e., start by using first-level headings); `1` is also the minimum value used.

Details

Use of this function is illustrated with the `Veccompare Overlap Report` RMarkdown template for RStudio that is installed as part of the **veccompare** package.

Value

A string of Markdown (and Venn diagrams, if `draw_venn_diagrams` is `TRUE`).

If `cat_immediately` is `TRUE`, nothing is returned by the function; rather, the output Markdown is printed immediately (for example, as part of a Knitted RMarkdown document, or to the console).

If `cat_immediately` is `FALSE`, the output can be saved to an object (as in the example below). This object can then be printed using `cat()`.

NOTE WELL: If `cat_immediately` is `FALSE`, the output *should* be saved to an object. If it is not, R will give an error message when printing to the console, because of unescaped special characters (which work correctly when `cat()` is used).

Examples

```
example <- compare.vectors.and.return.text.analysis.of.overlap(
    veccompare::example.vectors.list,
    cat_immediately = FALSE,
    draw_venn_diagrams = FALSE
)
cat(example)
```

`example.vectors.list` *Example Vectors List*

Description

An example dataset containing several named vectors, which can be compared to one another for overlaps, unique elements, etc.

Usage

```
example.vectors.list
```

Format

A list of named vectors.

---

extract.compared.vectors

Extract elements from the output of `compare.vectors`

Description

Straightforwardly extract particular elements from the output of `compare.vectors`.

Usage

```
extract.compared.vectors(output_from_compare.vectors, vector_names = NULL,
  only_match_vector_names = FALSE, degrees_of_comparison = NULL,
  elements_of_output = NULL)
```

Arguments

`output_from_compare.vectors`	The list output of `compare.vectors`.
`vector_names`	An optional vector of names to extract from the named list (`named_list_of_vectors_to_compare`) used with `compare.vectors`.
`only_match_vector_names`	A logical (TRUE/FALSE) indicator whether to match **only** `vector_names`. If `vector_names` is `c("a", "b")`, for example, and `only_match_vector_names` is `TRUE`, this function will output only the comparison between a and b. If `only_match_vector_names` is `FALSE`, however, this function will output the comparison between a and b, as well as between a, b, and c, etc.
`degrees_of_comparison`	An optional number of vector of numbers indicating which degrees of comparison to return (for example, 2 will return only two-way comparisons from `output_from_compare.vectors`.
`elements_of_output`	An optional vector of element names from `output_from_compare.vectors` to return (for example, "elements_involved"). See the **Value** section of `compare.vectors` for a list of the elements to choose from.

Value

A winnowed version of `output_from_compare.vectors`. Depending on arguments, either a list, a vector, or a string.

Examples

```
example <- veccompare::compare.vectors(veccompare::example.vectors.list)

# To extract similar elements across list items:
veccompare::extract.compared.vectors(
  example,
  elements_of_output = "elements_involved"
)

# To extract all comparisons that involve "vector_a":
veccompare::extract.compared.vectors(
  example,
  vector_names = "vector_a"
)

# To find all comparisons that were about "vector_a" and "vector_c":
veccompare::extract.compared.vectors(
  example,
  vector_names = c("vector_a", "vector_c"),
  only_match_vector_names = TRUE
)

# To get all elements that did a two-way comparison:
veccompare::extract.compared.vectors(
  example,
  degrees_of_comparison = 2
)

# A more complex / specific example:
extract.compared.vectors(
  example,
  vector_names = c("vector_a", "vector_c"),
  only_match_vector_names = FALSE,
  degrees_of_comparison = c(2, 3),
  elements_of_output = "elements_involved"
)
```

`generate.random.colors` *Generate Random Colors*

Description

An function to generate a given number of random colors.

Usage

```
generate.random.colors(number_of_colors_to_get)
```

Arguments

`number_of_colors_to_get`
: The number of colors to generate.

Value

A vector of R color names.

Examples

```
generate.random.colors(5)
```

`render.venn.diagram` *Render (Print) a Previously-Computed Venn Diagram*

Description

A wrapper function for printing a `grid`-based image using `grid::grid.draw()`.

Usage

```
render.venn.diagram(venn_diagram_created_with_VennDiagram_package,
  viewport_npc_width_height_for_images = 1)
```

Arguments

`venn_diagram_created_with_VennDiagram_package`
: A grid-based diagram object. For example, a Venn diagram previously generated using `veccompare::compare.vectors()`.

`viewport_npc_width_height_for_images`
: The scale at which to print an image. If the image is cut off at its edges, for example, this can be set lower than 1.0.

Value

The function will not return a value; rather, it will print the image.

Examples

```
# Create comparisons across 5 vectors, specifically creating all 4-way venn diagrams from them:
example <- veccompare::compare.vectors(
  veccompare::example.vectors.list[1:5],
  draw_venn_diagrams = TRUE,
  suppress_messages = TRUE,
  degrees_of_comparison_to_include = 4
)

# Get the first 4-way comparison that includes a diagram:
```

```
diagram <- veccompare::extract.compared.vectors(
  example,
  degrees_of_comparison = 4,
  elements_of_output = "venn_diagram"
)[[1]]$venn_diagram

# Print the diagram:
veccompare::render.venn.diagram(
  diagram,
  viewport_npc_width_height_for_images = .7
     # Scale the image down to 70%,
     # in case it otherwise gets cut off at the margins.
)
```

summarize.two.way.comparisons.percentage.overlap

Summarize Percentage Overlap for Two-Way Comparisons between Vectors

Description

Summarize Percentage Overlap for Two-Way Comparisons between Vectors

Usage

```
summarize.two.way.comparisons.percentage.overlap(named_list_of_vectors_to_compare,
 output_type = "table", melt_table = FALSE, network_graph_minimum = 0,
 margins_for_plot = NULL)
```

Arguments

`named_list_of_vectors_to_compare`
: A named list of vectors to compare (see, for example, `example.vectors.list`). Duplicate values in a given vector will only be counted once (for example, c("a", "a", "b", "c") will be treated identically to c("a", "b", "c").

`output_type`
: Either `"table"`, `"matrix_plot"`, or `"network_graph"`. `"table"` will return a matrix showing percentage overlap between each pair of vectors. `"matrix_plot"` will plot this table, coloring it by the amount of overlap. `"network_graph"` will return a network graph image illustrating the overlap percentages between each pair of vectors.

`melt_table`
: A logical (TRUE/FALSE) indicator, when `output_type` is `"table"`, whether to print the output in `melt`ed form (using the **reshape2** package).

`network_graph_minimum`
: `minimum` argument from `qgraph`, for when `output_type` is `"network_graph"`.

`margins_for_plot`
: The margins for image output (if `output_type` is `matrix_plot` or `network_graph`). Specified as a vector of numbers, in the form `c(bottom, left, top, right)`. If `output_type` is `matrix_plot`, defaults to `c(2, 0, 1, 0)`; if `output_type` is `network_graph`, defaults to `c(3, 3, 3, 0.5)`.

Value

Either a matrix (if `output` is `"table"`), or an image (if `output` is `"matrix_plot"` or `"network_graph"`). If an image is printed, nothing is returned by the function; rather, the output is printed immediately.

If `output` is `"table"` and `melt_table` is `FALSE`, the output will be a matrix with `nrow` and `ncol` both equal to the number of vectors in `named_list_of_vectors_to_compare`. This table shows the decimal percentage overlap (e.g., "0.20" = 20%) between each combination of vectors. *This table is intended to be read with row names first, in this form:* "[row title] overlaps with [column title] [cell value] percent."

If `output` is `"table"` and `melt_table` is `TRUE`, the output will be a `melt`ed data.frame with three columns: `Vector_Name`, `Overlaps_With`, and `Decimal_Percentage`.

Examples

```
summarize.two.way.comparisons.percentage.overlap(veccompare::example.vectors.list)
summarize.two.way.comparisons.percentage.overlap(
veccompare::example.vectors.list,
output_type = "table",
melt_table = TRUE
)

summarize.two.way.comparisons.percentage.overlap(
veccompare::example.vectors.list,
output_type = "matrix_plot" # You can also choose "network_graph"
)
```

`vector.print.with.and` *Print a vector with commas and a final "and".*

Description

Print a vector with commas and a final "and".

Usage

```
vector.print.with.and(vector_to_print,
  string_to_return_if_vector_is_empty = "", use_oxford_comma = TRUE)
```

Arguments

`vector_to_print`	A vector of strings (or elements able to be coerced into strings) to print.
`string_to_return_if_vector_is_empty`	If `vector_to_print` is empty, the string that should be returned (for example, "", "(None)", etc.)
`use_oxford_comma`	A logical (TRUE/FALSE) value indicating whether to use an Oxford comma ("One, two, and three" vs. "One, two and three").

Value

A single string that concatenates the input, separating with commas and adding "and" before the final item.

Examples

```
vector.print.with.and(c("One", "Two", "Three", "Four"))
vector.print.with.and(c("One", "Two", "Three", "Four"), use_oxford_comma = FALSE)
vector.print.with.and(c("One", "Two"))
vector.print.with.and(c("One"))
vector.print.with.and(c(), string_to_return_if_vector_is_empty = "(None)") # Outputs "(None)"
vector.print.with.and(c(""), string_to_return_if_vector_is_empty = "(None)") # Outputs ""
```

`which.of.one.set.is.not.in.another`

Which of One Set is not in Another

Description

This function is a wrapper for `setdiff`. It makes it easier to remember which vector is being subtracted from the other, by displaying an explicit message.

Usage

```
which.of.one.set.is.not.in.another(set_1, set_2, suppress_messages = FALSE)
```

Arguments

`set_1`	A vector to be subtracted from.
`set_2`	A vector to subtract from `set_1`.
`suppress_messages`	A logical (TRUE/FALSE) indicator whether to suppress messages.

Value

A vector of the values of `set_1` that are not present in `set_2`. Put differently, a vector resulting from subtracting `set_2` from `set_1`.

Examples

```
veccompare::which.of.one.set.is.not.in.another(
    veccompare::example.vectors.list$vector_a,
    veccompare::example.vectors.list$vector_b
)

veccompare::which.of.one.set.is.not.in.another(
    veccompare::example.vectors.list$vector_b,
    veccompare::example.vectors.list$vector_a
)
```

Index