Test for yaml function

Miguel Alvarez

Introduction

The package `yamlme` is developed to enhance automatic generation of reports and documents from different packages, for instance reporting data sources by the package `vegtable` or producing check-lists by the package `taxlist`. These applications are at the moment tested in an experimental way, for instance in the package `vegtable2` (see function `report_communities()`).

The function `write_yaml()` is at the moment implemented in an own package. For details, a brief description will be retrieved by `?write_yaml`.

```
devtools::install_github("kamapu/yamlme")
library(yamlme)
library(rmarkdown)
```

State of the Art

The idea behind this function is to create Rmd documents with conten by using R-code. For instance a PDF document by setting the different yaml items as arguments in the function.

```
Docu <- write_yaml(
        title="First Function Version",
        author="Miguel Alvarez",
        output="pdf_document")
cat(Docu)
```

```
## ---
##  title: First Function Version
##  author: Miguel Alvarez
##  output: pdf_document
##
## ---
##
##
```

The names of the yaml entries are provided by the user, except for the defined arguments `append`, `body`, and `filename`, which I am assuming are not required in Rmarkdown documents.

To produce a full document, we need to include a body part, an output file and render it with rmarkdown.

```
Docu <- write_yaml(
        title="First Function Version",
        author="Miguel Alvarez",
        output="pdf_document",
        body="This is a first incursion in Rmarkdown.",
```

```
        filename="test_doc.Rmd")
render("test_doc.Rmd")
```

THE IDEA is to insert the entries by a combination of 1) character vectors of length 1 (the previous examples), 2) character vectors longer than 1 and 3) complex structures by lists. At the moment only alternatives 1 and 2 are implemented, while the challenge with more complex entries is to make the formatting in a recursive way and at the same time detect the hierarchical rank (depth) to properly set indentation.

```
Docu <- write_yaml(
        title="First Function Version",
        author="Miguel Alvarez",
        output="pdf_document",
        "header-includes"=c(
                "- \\usepackage[utf8]{inputenc}",
                "- \\usepackage[T1]{fontenc}"),
        append="# Document written with 'write_yaml()'",
        body="This is a first incursion in Rmarkdown.",
        filename="test_doc.Rmd")
```

Note that the name of the entry **header-includes** have to be quoted because of the dash, while back-slashes have to be escaped.

Desiderata

The next step should be to implement lists as arguments for hierarchical entries in the yaml head. For instance

```
output=list(
        pdf_file="default")
```

should write

```
---
output:
  pdf_document: default
---
```

A more complex example can be provided, for instance, by multiple outputs

```
---
output:
  html_document:
    toc: true
    toc_float: true
  word_document:
    fig_caption: false
---
```

In that case, the argument output should be

```
output=list(
        html_document=list(
                toc="true",
                toc_float="true"),
        word_document=list(
                fig_caption="false"))
```

The function is expected to generate automatically the structure of the head entry depending on the structure of the list in the function. The good news is that we can tweak the (current) limitations of `write_yaml()` by using the argument `append`.

```
Docu <- write_yaml(
        title="First Function Version",
        author="Miguel Alvarez",
        append=paste(c(
                        "# Appended code",
                        "output:",
                        "  html_document:",
                        "    toc: true",
                        "    toc_float: true",
                        "  word_document:",
                        "    fig_caption: false"),
                collapse="\n"),
        body=paste(c(
                        "# Breakfast",
                        "",
                        "Just a cup of coffee.",
                        "",
                        "# Midday",
                        "",
                        "- A pizza from the Botan Grill",
                        "- A bottle of coca-cola"),
                collapse="\n"),
        filename="test_doc.Rmd")
render("test_doc.Rmd", output_format="all")
```

Some possible improvements may include:

- Output object could be defined as "S3" class.
- Optionally no object will be produced, just a file.
- Other alternative is to produce a summarized print in the console, when object is not assigned to an object.

Motivation

Why to do this effort? Our intention is to use the function for the "automatic" generation of reports. For instance, we can define an "alias function" including a template of a document.

```
auto_report <- function(title="Automatic Document",
        author="Me Robot",
        output="pdf_document",
        body="Hello world!",
        filename="test_doc.Rmd", ...) {
    write_yaml(
            title=title,
            author=author,
            output=output,
            body=body,
            filename=filename,
            ...)
}
```

```
Docu <- auto_report()
render("test_doc.Rmd")
```

We can then use this function to customize the template by changing the values of the arguments and even adding new ones (I guess, it is not possible to suppress yaml entries).

```
Docu <- auto_report(
        title="Human Report",
        author="Bisrat",
        date="1.1.2013",
        output="html_document",
        body=paste(c(
                        "# Breakfast",
                        "",
                        "Just a cup of coffee.",
                        "",
                        "# Midday",
                        "",
                        "- A pizza from the Botan Grill",
                        "- A bottle of coca-cola"),
                collapse="\n"))
render("test_doc.Rmd")
```

Enjoy!