December 24, 2014

### 0.0.1 genNavbar

`genNavbar` generates a navigation bar for a web page. The html file created should be written to the project's `docs/Rmd/include` directory. The common navigation bar html is included prior to the body of the html for each web page in the project's website. `menu` is a vector of names for each dropdown menu. `submenus` is a list of vectors of menu options corresponding to each menu. `files` is a similar list of vectors. Each element is either an html file for a web page to be associated with the submenu link, "header" to indicate the corresponding name in `submenus` is only a group label and not a link to a web page, or "divider" to indicate placement of a bar for separating groups in a dropdown menu.

```
# Functions for Github project websites
genNavbar <- function(htmlfile = "navbar.html", title, menu, submenus, files,
    title.url = "index.html", home.url = "index.html", site.url = "", site.name = "Github",
    include.home = FALSE) {

    fillSubmenu <- function(x, name, file) {
        if (file[x] == "divider")
            return("              <li class=\"divider\"></li>\n")
        if (file[x] == "header")
            return(paste0("              <li class=\"nav-header\">", name[x],
                "</li>\n"))
        paste0("              <li><a href=\"", file[x], "\">", name[x], "</a></li>\n")
    }

    fillMenu <- function(x, menu, submenus, files) {
        paste0("<li class=\"dropdown\">\n            <a href=\"", gsub(" ",
            "-", tolower(menu[x])), "\" class=\"dropdown-toggle\" data-toggle=\"dropdown\">",
            menu[x], " <b class=\"caret\"></b></a>\n            <ul class=\"dropdown-menu\">\n",
            paste(sapply(1:length(submenus[[x]]), fillSubmenu, name = submenus[[x]],
                file = files[[x]]), sep = "", collapse = ""), "            </ul>\n",
            collapse = "")
    }

    if (include.home)
        home <- paste0("<li><a href=\"", home.url, "\">Home</a></li>\n          ") else home <- ""
    x <- paste0("<div class=\"navbar navbar-default navbar-fixed-top\">\n  <div class=\"navbar-inner\">\
        title.url, "\">", title, "</a>\n      <div class=\"nav-collapse collapse\">\n        <ul class=\
        home, paste(sapply(1:length(menu), fillMenu, menu = menu, submenus = submenus,
            files = files), sep = "", collapse = "\n          "), "        </ul>\n        <ul class=\"na
        site.url, "\">\n            <i class=\"fa fa-github fa-lg\"></i>\n            ",
        site.name, "\n          </a>\n        </ul>\n      </div><!--/.nav-collapse -->\n    </div>\n  <
        collpase = "")
    sink(htmlfile)
    cat(x)
    sink()
    x
```

```
}
```

### 0.0.2 genOutyaml

`genOutyaml` generates the `_out.yaml` file for yaml front-matter common to all html files in the project website. The file should be written to the project's `docs/Rmd` directory. `lib` specifies the library directory for any associated files. yaml `includes` for external html common to all project web pages in the site can also be specified with `header`, `before_body`, and `after_body`. These can be specified by file basename only (no path) and the function assumes these files are in the `docs/Rmd/include` directory. At this time all external libraries must be provided by the user, for example in `docs/Rmd/libs`. It is recommended. See the project repo [gh-pages](https://github.com/leonawicz/ProjectManagement/tree/gh-pages "gh-pages") branch for an example.

```
genOutyaml <- function(file, theme = "cosmo", highlight = "zenburn", lib = NULL,
    header = NULL, before_body = NULL, after_body = NULL) {
    output.yaml <- paste0("html_document:\n  self_contained: false\n  theme: ",
        theme, "\n  highlight: ", highlight, "\n  mathjax: null\n  toc_depth: 2\n")
    if (!is.null(lib))
        output.yaml <- paste0(output.yaml, "  lib_dir: ", lib, "\n")
    output.yaml <- paste0(output.yaml, "  includes:\n")
    if (!is.null(header))
        output.yaml <- paste0(output.yaml, "    in_header: ", header, "\n")
    if (!is.null(before_body))
        output.yaml <- paste0(output.yaml, "    before_body: ", before_body,
            "\n")
    if (!is.null(after_body))
        output.yaml <- paste0(output.yaml, "    after_body: ", after_body, "\n")
    sink(file)
    cat(output.yaml)
    sink()
    output.yaml
}
```