# *Customizing Syntax Highlighting Themes*

*Yihui Xie & Ramnath Vaidyanathan*

*August 11, 2016*

This manual[1] shows how to customize syntax highlighting of source code using themes. It walks the user through the basics of syntax highlighting in **knitr**, and the use of built-in themes.

The **knitr** package uses the **highr** package to highlight source code in a document. In short, the **highr** package parses the source code, tokenizes it into grammar symbols, and formats their appearance using CSS or LaTeX commands.

## *Usage*

We can use the object `knit_theme` to set / get a theme. See `?knit_theme` for the usage. For example, we set the theme of this document to be `solarized-dark`:

```
thm <- knit_theme$get("solarized-dark")
knit_theme$set(thm)
```

## *Built-in Themes*

The listing on the right shows the CSS file for one of the themes, `edit-eclipse`, which was adapted from Andre Simon's excellent highlighter[2]. The **knitr** package comes pre-loaded with a number of themes based on this highlighter. Here is list of all available code themes[3]:

```
knit_theme$get()

##  [1] "acid"              "aiseered"          "andes"
##  [4] "anotherdark"       "autumn"            "baycomb"
##  [7] "bclear"            "biogoo"            "bipolar"
## [10] "blacknblue"        "bluegreen"         "breeze"
## [13] "bright"            "camo"              "candy"
## [16] "clarity"           "dante"             "darkblue"
## [19] "darkbone"          "darkness"          "darkslategray"
## [22] "darkspectrum"      "default"           "denim"
## [25] "dusk"              "earendel"          "easter"
## [28] "edit-anjuta"       "edit-eclipse"      "edit-emacs"
## [31] "edit-flashdevelop" "edit-gedit"        "edit-jedit"
## [34] "edit-kwrite"       "edit-matlab"       "edit-msvs2008"
```

```
.background {
  color: #ffffff;
}
.num {
  color: #000000;
}
.str {
  color: #0000ff;
}
.com {
  color: #717ab3;
}
.opt {
  color: #000000;
}
.std {
  color: #000000;
}
.kwa {
  color: #7f0055;
  font-weight: bold;
}
.kwb {
  color: #7f0055;
  font-weight: bold;
}
.kwc {
  color: #7f0055;
  font-weight: bold;
}
...
```

[1] source code at `https://github.com/yihui/knitr/blob/master/inst/examples/knitr-themes.Rnw`

[2] `http://www.andre-simon.de/`

[3] For a preview of all themes, see `https://gist.github.com/yihui/3422133`

```
## [37] "edit-nedit"        "edit-vim"          "edit-vim-dark"
## [40] "edit-xcode"        "ekvoli"            "fine_blue"
## [43] "freya"             "fruit"             "golden"
## [46] "greenlcd"          "greyscale0"        "greyscale1"
## [49] "greyscale2"        "kellys"            "leo"
## [52] "lucretia"          "manxome"           "maroloccio"
## [55] "matrix"            "moe"               "molokai"
## [58] "moria"             "navajo-night"      "navy"
## [61] "neon"              "night"             "nightshimmer"
## [64] "nuvola"            "olive"             "orion"
## [67] "oxygenated"        "pablo"             "peaksea"
## [70] "print"             "rand01"            "rdark"
## [73] "relaxedgreen"      "rootwater"         "seashell"
## [76] "solarized-dark"    "solarized-light"   "tabula"
## [79] "tcsoft"            "vampire"           "whitengrey"
## [82] "xoria256"          "zellner"           "zenburn"
## [85] "zmrok"
```

Shown below is how the `solarized-dark` theme looks like when applied to R code:

```
library(XML)
library(plyr)
library(reshape)
# SCRAPE THE DATA FROM WEB

base_url <- "http://www.mlsoccer.com/stats/%s/reg"
years <- 1996:2010
options(width = 40)

#' Function to save data for each year
save_data <- function(y) {
    url <- sprintf(base_url, y)
    tab <- readHTMLTable(url, header = FALSE, stringsAsFactors = FALSE)
    pos <- max(grep("United", tab))
    tab <- tab[[pos]]
    tab$year <- y
    return(tab)
}

team.list <- llply(years, save_data)
mls <- merge_recurse(team.list)
```

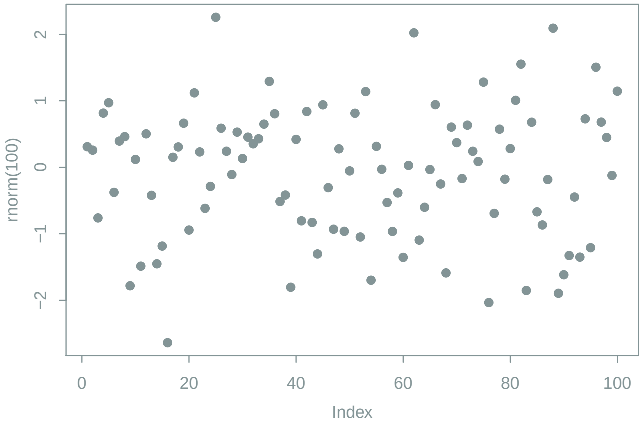
## *Misc*

One thing to consider is the foreground color of plots when we use dark themes; we need to make it lighter, otherwise the graphical elements will be difficult to be read. We can access the foreground color of the theme in the list returned by `knit_theme$get(theme)`, e.g., for this document:

```
## the object thm is from the first chunk
thm$foreground  # foreground color

## [1] "#839496"

thm$background  # background color

## [1] "#002b36"
```

When we make plots, we may use these colors, e.g.

```
## can design a chunk hook to set foreground color automatically
fgcolor <- thm$foreground
par(fg = fgcolor, col.axis = fgcolor, col.lab = fgcolor)
plot(rnorm(100), pch = 19)
```



Of course, we do not need to worry about these colors when we use a white background for the plots.