Writing R packages

Karl Broman

Biostatistics & Medical Informatics, UW–Madison

kbroman.org
github.com/kbroman
@kwbroman
Course web: kbroman.org/AdvData
Why write an R package?

- To distribute R code and documentation
- To keep track of the misc. R functions you write and reuse
- To distribute data and software accompanying a paper.
Why write an R package?

- To distribute R code and documentation
- To keep track of the misc. R functions you write and reuse
- To distribute data and software accompanying a paper.
A simple example: RSkittleBrewer

Skittle-themed color schemes for R graphics with RSkittleBrewer!

Choosing the perfect set of colors for a plot is hard. But people have thought a lot about this problem, and there are solutions! If you’re an R user looking for publication-quality color schemes that are backed by lots of scientific research, check out RColorBrewer, or use the color schemes in ggplot2.

If, on the other hand, you’re looking for a color scheme that reminds you of a bag of Skittles, check out RSkittleBrewer. This is a tiny R package I wrote yesterday to generate vectors of valid R color names for certain Skittle flavors. You can also generate a vector of M&M colors, if (like me) you’re more into chocolate.

The code is on GitHub. Here’s how you use it:

```r
# original
original = RSkittleBrewer("original")

# original
original = RSkittleBrewer("original")

# original
original = RSkittleBrewer("original")

# original
original = RSkittleBrewer("original")
```

And if you want to see how the colors actually look, you can make a plot:

```
plotSkittles()
```

It will look like this:
R package contents

RSkittleBrewer/

DESCRIPTION
NAMESPACE

R/RSkittleBrewer.R
R/plotSkittles.R
R/plotSmarties.R

man/RSkittleBrewer.Rd
man/plotSkittles.Rd
man/plotSmarties.Rd

github.com/alyssafrazee/RSkittleBrewer
Package: RSkittleBrewer
Version: 1.1
Author: Alyssa Frazee
Maintainer: Alyssa Frazee <afrazee@jhsph.edu>
Title: Fun with R Colors
Description: For those times you want to make plots with candy-themed color schemes.
URL: https://github.com/alyssafrazee/RSkittleBrewer
License: MIT + file LICENSE
Imports: graphics, stats, grDevices
Suggests: knitr, rmarkdown
VignetteBuilder: knitr
RoxygenNote: 5.0.1
NAMESPACE file

```r
export(RSkittleBrewer)
export(plotSkittles)
export(plotSmarties)
```
An .Rd file

\name{RSkittleBrewer}
\alias{RSkittleBrewer}
\title{Candy-based color palettes}
\description{Vectors of colors corresponding to different candies.}
\usage{RSkittleBrewer(flavor = c("original", "tropical", "wildberry", "M&M", "smarties"))}
\arguments{
  \item{flavor}{Character string for candy-based color palette.}
}
\value{Vector of character strings representing the chosen set of colors.}
\examples{
  plotSkittles()
  plotSmarties()
}
\keyword{hplot}
\seealso{ \code{\link{plotSkittles}}, \code{\link{plotSmarties}} }
# RSkittleBrewer
#
# Candy-based color palettes
#
# Vectors of colors corresponding to different candies.
#
# @param flavor Character string for candy-based color palette.
#
# @export
# @return Vector of character strings representing the chosen...
#
# @examples
# plotSkittles()
# plotSmarties()
#
# @seealso \code{\link{plotSkittles}},
# \code{\link{plotSmarties}}
# @keywords hplot
RSkittleBrewer <-
...
# build package documentation
doc:
    R -e "devtools::document()"
fun with R Colors
=================

If you want high-quality, scientifically-researched color schemes for your R plots, check out [RColorBrewer](http://cran.r-project.org/web/packages/RColorBrewer). If you want your plots to be colored the same way as packs of Skittles (or M&Ms), then this package (RSkittleBrewer) is the way to go.

``` R
devtools::install_github('RSkittleBrewer', 'alyssafrazee')
```

There are only three functions in this package.

Call `RSkittleBrewer` on a flavor to get a vector of R color names that correspond to that Skittle flavor.
That’s it!
R CMD build RSkittleBrewer
R CMD INSTALL RSkittleBrewer_1.1.tar.gz
R CMD check RSkittleBrewer_1.1.tar.gz

R CMD check --as-cran RSkittleBrewer_1.1.tar.gz

R CMD INSTALL --library=~/Rlibs RSkittleBrewer_1.1.tar.gz
# (~/.Renviron file contains R_LIBS=~/Rlibs)

# On windows:
R CMD INSTALL --build RSkittleBrewer_1.1.tar.gz

# also consider (within R):
library(devtools)
built("/path/to/RSkittleBrewer")
built("/path/to/RSkittleBrewer", binary=TRUE)
Package vignettes

▶ Include *vignettes* to show how to use your package.

▶ It’s simplest to use R Markdown.
  – Create a `vignettes/` subdirectory.
  – Place a `.Rmd` file there.
  – The name of the file becomes the name of the vignette.

▶ Include the following in the `.Rmd` file’s YAML header:

```yaml
output: rmarkdown::html_vignette
vignette: >
  \%\VignetteIndexEntry{Intro to RSkittleBrewer}
  \%\VignetteEngine{knitr::rmarkdown}
  \usepackage[utf8](inputenc)
```

▶ Load the package in an initial chunk

```r
library(RSkittleBrewer)
```
## Package vignettes

- **In the DESCRIPTION file, include:**
  
  Suggests: knitr, rmarkdown
  VignetteBuilder: knitr

- **The following lists the vignettes for a package and then opens a selected vignette.**
  
  ```
  library(RSkittleBrewer)
  vignette(package="RSkittleBrewer")
  vignette("RSkittleBrewer", "RSkittleBrewer")
  ```
Optional stuff

- NEWS file describing changes in each version of the package.
- inst/CITATION file describing how to cite your package.
- inst/doc/ directory any sort of misc. documentation (e.g., pre-compiled computationally heavy vignettes)
- data/ directory containing data
- src/ directory containing C/C++/Fortran code
- demo/ directory with demonstrations (like vignettes, but to be executed in real-time).
- tests/ and/or inst/tests/ containing tests.
devtools

Get to know the `devtools` package.

- `dev_mode()`
- `load_all()`
- `install_github()`, `install_bitbucket`, ...
- `document()`
- `build()`
- `check()`
- `check_doc()`
- `run_examples()`
- `test()` (next week)
usethis

automates things otherwise hard to remember

- create_package()
- use_git(), use_github()
- use_mit_license(), use_gpl3_license()
- use_readme_md()
- use_r("function_name")
- use_roxygen_md()
- use_package("stats", "imports")
- use_vignette()
Summary

- R packages really aren’t that hard.
- R packages are really useful.
  - Distributing software and data
  - Organizing code for a paper
  - Organizing your misc. R functions
- Look at others’ packages, and learn from them.
- Adopt the tools in the devtools package.
- Use usethis package to assist with all of the little nit-picky package development things.