Make an R package

yingying

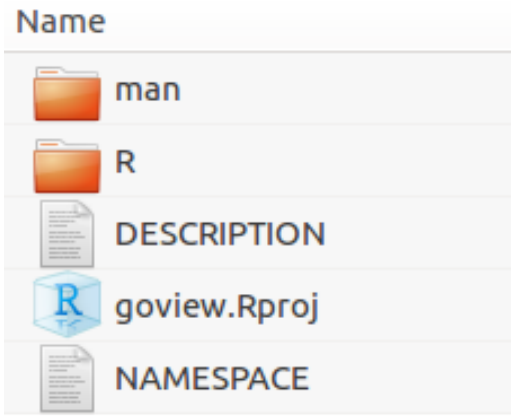
January 10, 2019

Step 0: packages you will need

```
install.packages("devtools")
library("devtools")
devtools::install_github("klutometis/roxygen")
library(roxygen2)
```

Step 1: Create your package project

- Create an R package project which only includes the R package folder and the default files.
- Look in the R package folder



Step 2: Add functions

```
goview <- function(goresults){
  ggplot2::ggplot(data=goresults,
         aes(x=reorder(term.name, -p.value),
             y=overlap.size,fill=p.value)) +
  geom_bar(stat="identity")+
  scale_fill_gradient2(mid='red',
                       high='blue', space='Lab')+
  labs(title="GO enrichment",
       x ="GO terms", y = "number of genes",
       fill = "P value")+
  theme(plot.title = element_text(hjust = 0.5,
                                  size = 12))+
  coord_flip()
}
```

Step 3: Add documentation

- Use package roxygen2 to make documentation.

```
#' goview Function
#' This function allows you to show the results of GO enri
#' @param goresults the go results from gProfileR
#' @keywords GO
#' @author yingying
#' @import ggplot2
#' @export
#' @examples
#' goview()
goview <- function(goresults){
  ...
}
```

Step 4: Process the documentation

```
document()
```

```
> document()
Updating goview documentation
First time using roxygen2. Upgrading automatically...
Writing NAMESPACE
Loading goview
Warning: goview.R:11: Missing name
Writing NAMESPACE
```

Step 5: Install the package

```
setwd("..")
install("goview")
```

```
> install("goview")
✔  checking for file ‘/home/yingying/Desktop/DrStudy/SelfLectures/Rcodes/ShellR4/goview/DESCRIPTION’ ...
─  preparing ‘goview’:
✔  checking DESCRIPTION meta-information ...
─  checking for LF line-endings in source and make files and shell scripts
─  checking for empty or unneeded directories
─  building ‘goview_0.1.0.tar.gz’

Running /usr/lib/R/bin/R CMD INSTALL /tmp/RtmpcxhOy4/goview_0.1.0.tar.gz --install-tests
* installing to library ‘/home/yingying/R/x86_64-pc-linux-gnu-library/3.5’
* installing *source* package ‘goview’ ...
** R
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (goview)
```

Step 6: Make the package a GitHub repo

```
git init
git add .
git commit
git remote add origin git@github.com/yingstat/goview.git
git push -u origin master
```

Step 7: Install the package from github

```r
devtools::install_github("yingstat/goview")
```

```
> install_github("yingstat/goview")
Downloading GitHub repo yingstat/goview@master
✔  checking for file ‘/tmp/Rtmp3RSnXk/remotes1b43203c6d58/yingstat-goview-3179224/DESCRIPTION’ ...
─  preparing ‘goview’:
✔  checking DESCRIPTION meta-information ...
─  checking for LF line-endings in source and make files and shell scripts
─  checking for empty or unneeded directories
─  building ‘goview_0.1.0.tar.gz’

Installing package into ‘/home/yingying/R/x86_64-pc-linux-gnu-library/3.5’
(as ‘lib’ is unspecified)
* installing *source* package ‘goview’ ...
** R
** byte-compile and prepare package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (goview)
```

Step 8: Check the package

```
library(goview)
?goview
```

goview {goview}

goview Function

Description

This function allows you to show the results of GO enrichment analysis.

Usage

```
goview(goresults)
```

Arguments

goresults	the go results from gProfileR

Author(s)

yingying

Examples

```
goview()
```

Step 9: Use the package

```
go <- read.csv(file = "UP_GO.csv", header = TRUE)
goview(go)
```