Generate R Package Example
roxygen2 and pkgdown

Go back to fan's R4Econ.

1 Objective

Document and generate sharable R package.

1. Use roxygen2 from Hadley to document package.
2. Use pkgdown to package file and publish to github pages.

2 File Structure and Naming Convention

Folders:

1. R function functions in the */R* folder
2. RData files in the */data* folder

Naming Conventions:

1. functions files and functions all should be *snake_case_names*
2. function name prefix:
    - *fs_*: for non-project specific script files
    - *ffs_*: for project specific functions script files
    - *fv_*: for non-project specific vignettes files, generally RMD
    - *ffv_*: for project specific functions vignettes files, generally RMD
    - *ff_*: for non-project specific functions files
    - *ffp_*: for project specific functions files
    - Each function file could be prepared to have multiple functions inside, each file have the root which is the function name.

3 Create R Project

3.1 Folder does not exist yet

If the project/folder does not yet exist:

1. *devtools:create* the folder of interest
2. Move your files over to */R* and */data* folders.
    - Write files following conventions above and with r descriptions
3. *pkgdown::build_site()*

```
devtools::create("C:/Users/fan/R4Econ")
```

3.2 Folder already exists

If there is already a folder with a bunch of files including R and not R files, and the folder needs to be converted to become a R package and was previously not a R package.

The idea is to use a folder somewhere to generate a generic template folder. This folder will have files and structure we need for our actual folder. Each time, we will just copy that folder's contents into the folder that we want to turn into a R folder. And do a global search to replace the template folder's folder name with the actual project name. To avoid confusion, generate this folder outside of an existing R package.

If there is already a R folder in your existing project, delete that or rename that and move files back into the */R* folder after completion. Make sure there are no duplicate folder or file names in the old and the new project.

Search replace the word *rprjtemplate* inside your old project folder with the new files, replace that with your project name. Should appear in three different spots.

3.2.1 R project template

The [rprjtemplate][https://github.com/FanWangEcon/Tex4Econ/tree/master/nontex/rprjtemplate] serves this templating role.

```
devtools::create("C:/Users/fan/Tex4Econ/nontex/rprjtemplate")
```

Running the *devtools::create()* command will create the core needed folder structure with:

1. NAMESPACE
2. DESCRIPTION
3. *.gitignore*
4. *.Rbuildignore*
5. *.Rproj*
6. Empty R folder

Now customize this folder for future use with

1. custom *.gitignore*
2. MIT LICENSE
3. etc.

4 Package R Project

4.1 Build Project

```
devtools::build('C:/Users/fan/PrjOptiAlloc')
```

4.2 R project build and generate documentations

Assuming that we have used roxygen2 formats to write functions, now generate *.Rd* automatically with the *document()* function. This should create a */man* folder in which various *.Rd* files are stored. Note that their .Rd are for specific R functions, not for files that contain multiple functions.

```
devtools::document('C:/Users/fan/PrjOptiAlloc')
```

4.3 R project build site with pkgdown

Once we have used *pkgdown::build_site()*, by default, all html and other package presentation files and reference files are saved in the */docs* folder, build_site is for building site, not for generating */man* documenation. If R files change, need to document first, before rebuilding site.

```
pkgdown::build_site("C:/Users/fan/PrjOptiAlloc")
```

4.4 Dealing with Datasets

Datasets to be used with the project should be in the */data* folder. The name of the data file should appear in several spots and be consistent, suppose data is called *abc*

1. Rdata file name: */data/abc.Rdata*
2. Rdata file contains a dataframe inside that has to be called abc as well, so open the Rdata file inside Rstudio or R, what is the file called? is it abc? Look under environment
3. In the */R/ffp_abc.R*, the last line should be "abc", but the file name does not need to be.