

## CDT R Packages

1. **Accounts.** Make sure that R, RStudio and git are installed on your computer (see overleaf).
2. **Packages.** Install the packages `devtools`, `roxygen2`, `testthat`, `knitr`, `rmarkdown`.<sup>1</sup>
3. **Create A Package.** With RStudio, go to `File > New Project...`, and select ‘New Directory’ and then ‘R Package’. Give your package a name.
4. **Version Control.** If prompted when creating the package, you can ask RStudio to create a git repository. Otherwise, you can do this later by going to `Tools > Version Control > Project Setup...`, and then changing the Version Control option from `(none)` to `git`. Allow RStudio to restart. [If this is not an option then you may need to troubleshoot your git installation.]
5. **Code.** Your package’s functions should all be contained in the subfolder `R`. Do not place any other code in this folder, as it will be executed every time you load your package.
6. **Documentation.** Using the Roxygen syntax, write the documentation directly above each of your functions. Use the following guidelines:
  - start with a title and then a brief description of what the function does;
  - every argument should be documented using `@param`;
  - if your function returns an object use the `@return` tag to say what sort of object it is;
  - `@details` can be used to give more information about what the function does;
  - `\eqn{}` and `\deqn{}` can be used to put in mathematics (inline and displayed equations respectively);
7. **Loading and Documenting.** To reload your package after a modification, press `Cmd+Shift+L` or `Ctrl+Shift+L`. To update the documentation first, press `Cmd+Shift+D` or `Ctrl+Shift+D`.
8. **Commits.** To commit your code, go to the `git` window in the top right of your screen and press `Commit`. In the window that opens, select the files you want to ‘stage’ (i.e. whose changes should be recorded), and add a message describing your changes. Click `commit!`
9. **Pushing commits.** To share your commits with collaborators, you need a remote repository. Create a blank repository on Github. Then open a shell in RStudio (`Tools > Shell...`). Type:

```
git remote add origin www.github.com/username/reponame
git remote -v
```

The second command should show you four lines of output.  
If you return to RStudio, you should now be able to ‘push’ your commit to Github.
10. **Add Collaborators.** For the projects you’ll need to be able to add the other member(s) of your team to the same project. Do this on Github by clicking on the ‘Settings’ for the project, ‘Collaborators’, and then adding your partners’ usernames.

---

<sup>1</sup>You can find more useful packages at [www.rstudio.com/products/rpackages/](http://www.rstudio.com/products/rpackages/)

## Installing Git

You can install Git by following the instructions at [git-scm.com](https://git-scm.com) (this is OS dependent).

When this is done, you should be able to open a terminal (or 'command prompt', in Windows) and type `git` to see some documentation.

Set up the configuration file to include your name and email. This makes it easier to blame your commits on you.

```
git config --global user.name "John Doe"  
git config --global user.email johndoe@example.com
```

It should be possible to do *almost* everything else via the graphical interface in RStudio.