Creating an R Data Package

Math 241

Week 3

Step 1 (I did this for you): Create a version control R package.

- Create package with `usethis::create_package("~/filepath/math241S20PkgGrpX")` (insert your group number).
- Add `git` with `usethis::use_git()`.
  - Don't commit.
  - But do restart.
  - Then commit the DESCRIPTION and NAMESPACE.
- Add to GitHub with `usethis::use_github(organisation = "Reed-Statistics", private = TRUE)` (You can't do this step without changing settings so skip for now.)

Step 2: Create a version control R project called, `math241S20PkgGrpX` (insert your group number), that is linked to the GitHub repo, `math241S20PkgGrpX` (insert your group number).

- Make sure you have a **Git** pane and a **Build** pane in the upper-right window of your RStudio session.

Step 3: List the following files in the `.gitignore`:

- `.gitignore`
- `.Rbuildignore`
- `math241S20PkgGrpX` (insert your group number)

Step 4: Add the raw data and transform it into the clean form I want to share in the package.

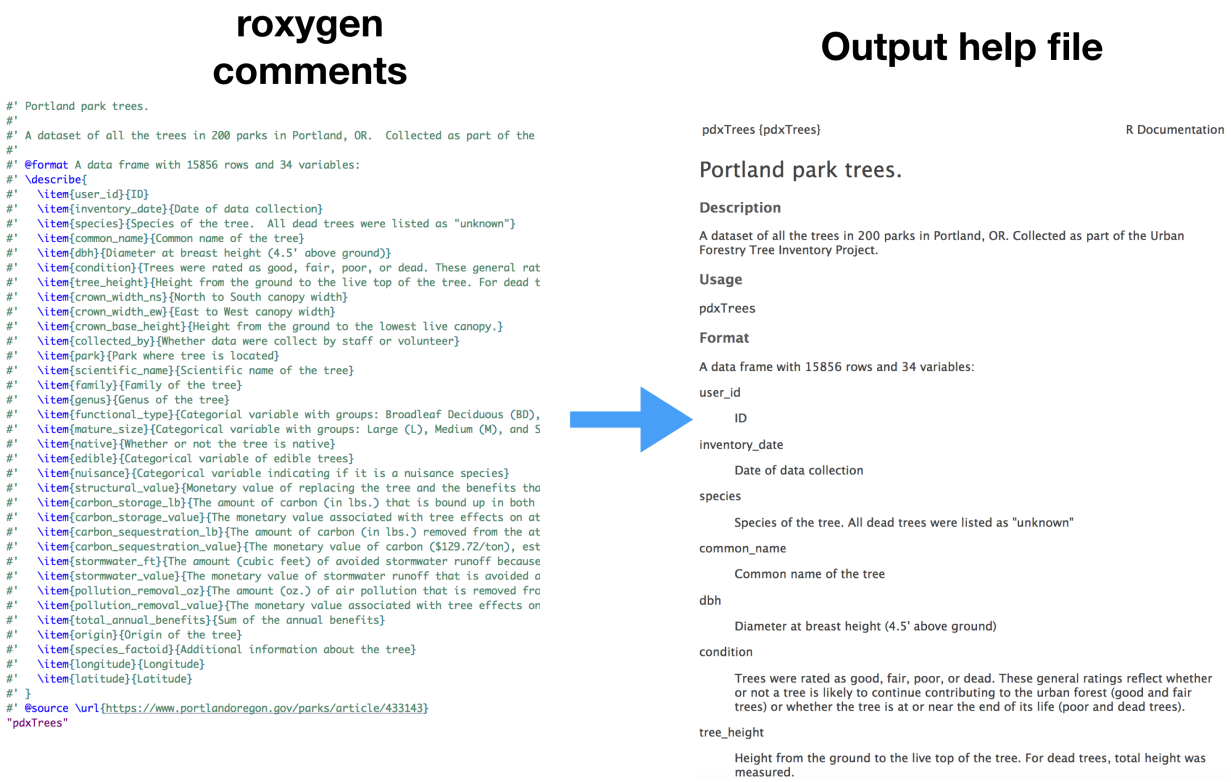
- Run the function `usethis::use_data_raw()` to create a `data-raw` folder in the R project and the file `DATASET.R`.
- Put the raw data files in the `data-raw` folder.
- Use the file `DATASET.R` to load and wrangle the raw data.
- At the bottom of the wrangling file, include the following code to create a clean `.Rda` file:

```
# Swap pdxTrees for the name of your data
usethis::use_data(pdxTrees, overwrite = TRUE)
```

- Run the code in `DATASET.R` to create a new folder called `data` that contains the cleaned data file.

Step 5: Create the documentation/help file for the data.

- Run `usethis::use_r("insert_data_name")` to create a script file.
- Add `roxygen` comments that document the dataset.
  - Here's an example.
  - See the Object Documentation Chapter of R packages.
- Run `devtools::document()`, which creates the `man` folder and an `Rd` help file.
- Run `?insert_data_name` in the console to view the help file.



- Notice that you need to include the dataset name as a string in the bottom of the R script. If you forget this, the help file for that dataset won't be created!

Step 6: Edit the DESCRIPTION file and consider creating a Readme file. The code `usethis::use_readme_md(open = interactive())` generates a skeleton Readme file.

Step 7: Delete any extraneous files or list them in .Rbuildignore.

Step 8: Run checks with `devtools::check(document = FALSE)` to make sure the package compiles without errors or warnings. Run `devtools::load_all()` to simulates the process of building, installing, and attaching the package if you want to check its contain.

- Fix any errors or warnings. (Note: it will still compile under warnings.)

Step 9: Try installing the package with `devtools::install()`.

Step 10/Always: Push the changes to the GitHub repository.