ETC5523: Package Assessment Instructions

Stuart Lee

Due date: Oct 23 2020 11.55PM AEST

This assessment requires you to make an R package. This assessment represents 25% of your final grade of ETC5523.

1. Accept the assessment at GitHub Classroom: https://classroom.github.com/a/CxuRZhSh. All your work should be pushed to your package assessment repository. You are encouraged to push to the repository regularly to avoid unwanted technical issues close to the deadline. ***By accepting this assessment, you acknowledge that what you submit to your repository is your own work.*** **(1 mark)**

2. For this assessment you will need to embed your `shiny` application from assessment 2 and make it launch via a function exported by your package. We will download your final submission before the deadline to your package assessment GitHub repository and check your package locally for marking. Marking will take into account the following points:

   - Your repository has a folder called `inst/app/` that contains the relevant R code to run your `shiny` application. ***(1 mark)***
   - There is an exported function called `launch_app()` that will find and load your `shiny` application from `inst/app`. ***(1 mark)***
   - Refactor at least one part of the user interface-side logic of your app into R functions that are exported by your R package. For example, you could modify repetitive calls to create a list of sliders into a single R function that are then called by your app. **(4 marks)**
   - Refactor at least one part of the server-side logic of your app into R functions that are exported by your R package. For example, you could modify complicated `reactive()` calls into separate R functions that are then called by your app. **(4 marks)**
   - Write at least one unit test for the functions you created to simplify the user interface of your app. **(2 marks)**
   - Write at least one unit test for the functions you created to simplify the server logic of your app. **(2 marks)**
   - All functions in the package are documented with `roxygen2` according to guidelines given in lectures. **(3 marks)**
   - All unit tests pass without error. **(2 marks)**
   - The DEESCRIPTION file includes an informative name and title for your package, as well as a valid author field and dependencies. There should also be a valid LICENCE file. **(3 marks)**
   - There should be a vignette that describes how to launch the app, what the app does, and guidelines for its use. **(3 marks)**
   - There is a `README.md` file that has clear instructions on how to install the package from GitHub, and gives a big picture overview of the package. ***(3 marks)***
   - There is a `pkgdown` site that contains the documentation for your package. **(2 marks)**
   - The `pkgdown` site has customised at least one element of the `_pkgdown.yaml` file. **(1 mark)**
   - The `pkgdown` site is available publicly via deployment from Github Actions. **(2 marks)**
   - Documentation and vignettes have correct spelling, grammar and punctuation. ***(1 mark)***
   - The package passes R CMD CHECK without errors or warnings ***(3 marks)***
   - Structure of the package is coherent and logical. ***(2 marks)***

Hints and Notes

- Maximum of 3 bonus marks for significant effort.
- The template we have set up includes Github Actions for automatically checking your package as you make changes. Use this to see where your package needs changing.
- The total marks for the assessment without bonuses is 40.

- Read over chapter 14 of mastering shiny for guidance on turning parts of your app into R functions.
- Read over chapter 16 of R packages for notes on using external files and the help page for `system.file()`.
- It is also acceptable and possible to refactor your app into an R package without using the external `inst/app/`, see chapter 16 of mastering shiny for details.

NO LATE TURN-INS ACCEPTED