Use this checklist as a guide for the items you should be checking in your own scripting (in addition to code bugs/defects) before submitting it for review by your peers.

Add notes for each point as you go through to comment on what was done well/ areas the coder can improve in. Mention line numbers if you notice great examples or inconsistencies.

File Organization/Project Management: *Are the files well-named and accessible?*

1. Are names for the current file under review (e.g., .R, .RMD) and all utilized/referenced files (e.g., .CSV, .DAT) distinctive?

   ☑ ~~Always~~ ☐ Sometimes ☐ Never

   Comments: only 4 items in repo and easy to distinguish

2. Are all referenced files available to the team (e.g. in the repo)?

   ☑ ~~Yes~~ ☐ No ☐ N/A

   Comments: yup, in repo

3. Do names for the current file under review (e.g., .R, .RMD) and all utilized/referenced files (e.g., .CSV, .DAT) accurately describe their content?

   ☑ ~~Always~~ ☐ Sometimes ☐ Never

   Comments: space.csv could be a bit more descriptive but it was named for ease (astronauts has more potential for spelling errors)

Documentation/Comments: *Is the code well annotated?*

1. Are comments present for each **distinct chunk/section of code** and include enough explanation, information, and references for another coder to easily understand the code?

   ☐ Always ☑ ~~Sometimes~~ ☐ Never

   Comments: there are header comments but don’t give much info

2. Are comments present for each **data frame** to describe the variables (or a pointer to the codebook)?

□ Always □ Sometimes ☑ ~~Never~~

Comments: no comments telling us what the variables we're creating are doing, particularly confusing in line 50 when we reference column 22 which did not exist until we used the nearPoints() addDist = T argument to add an extra column

3. Are comments present for each **function** created and specify inputs and outputs?

□ Always □ Sometimes □ Never

Comments: no created functions in this chunk, usage of nearPoints should be commented in line 46 because it would have helped avoid redundancy in the following lines

4. Are comments present for each **plot** created and have captions (or usage of fig.cap = "")?

□ Always □ Sometimes ☑ ~~Never~~

Comments: no caption for the plot on output and no comments specifying how user or other coders would know what's happening

---

Algorithms: *Is each part of the code as simple/non-redundant as possible?*

1. Does the code use at least one function from each package that is loaded?

☑ ~~Yes~~ □ No

Comments: 4 libraries loaded, nearPoints() is from shiny and ggplot2 is used for the scatterplot, other 2 libraries are used in other parts of the code (not currently under review)

2. Are functions created or implemented to avoid repetitive tasks?

□ Always ☑ ~~Sometimes~~ □ Never

Comments: 2 main tasks that utilize user input so no need for creating functions in this chunk. Redundant code in 49, 50, and 52 as the nearPoints() function in use already does this

3. Do code modules/chunks have clearly defined purposes with a limited set of tasks?

☑ ~~Always~~ □ Sometimes □ Never

Comments: code under review has clear purpose and tasks are limited

4. Do code modules/chunks have unique functionality and/or output?

☑ ~~Always~~ □ Sometimes □ Never

Comments: scatterplot and click information are related but separate tasks with separate input and output

---

Structure + Presentation: is the code organized for quick read through + understanding of parts and the whole?

1. Is the code formatted according to the team's agreed upon style guide (e.g., all object names are lowercase with underscores, indentation)?

□ Always ☑ ~~Sometimes~~ □ Never

Comments: style guide would have helped: naming and casing in lines 51 and 52 is not consistent, spacing in line 36, 39, and 52. = vs <- lines 49 and 50, TRUE vs T (and FALSE vs F) on line 48

2. Are object names distinctive?

☑ ~~Always~~ □ Sometimes □ Never

Comments: distinctive but not always descriptive

3. Do object names accurately describe their content?

□ Always ☑ ~~Sometimes~~ □ Never

Comments: subsetting in 49-53 doesn't give any description

4. Has all unused and commented-out code been removed from the script?

□ Yes ☑ ~~No~~

Comments: unused code and irrelevant comments in 37, 39, 44, 45, 47

5. Are line lengths limited so that all text can be read with little left/right scrolling on a laptop?

□ Always ☑ ~~Sometimes~~ □ Never

Comments: mostly no scrolling necessary except for line 44 is a really long comment