### Vignette edits

* I agree with Beth that one example is probably sufficient for the vignette, because you did a nice job with making it so thorough. It will help the vignette flow better, too, if you move seamlessly through the whole process on one example – from stratify, to recruit, etc.
* Step 1: Stratify, Example One
  + Missing word in first sentence. “researcher who wants to test the \_\_\_\_.”
  + Would it be better to filter on level == “High” instead of g\_10\_offered == “Yes” to get high schools? I noticed this results in less schools, however, so not sure how “High” is defined.
* Step 3: Assess, Example One
  + First sentence “…units from each stratum - 20, 4, 6, and 11 units…” Change to colon (“...each stratum: 20…”) instead of dash – currently looks like a negative sign in front of the number 20

### stratify()

* Selecting stratification variables
  + If users respond with “No” when asked “Is this correct?” for the specified stratifying variables, is there a way to bring them back to the same step to modify their answer, rather than being exited from the function altogether?
  + “Invalid input, please try again” error when selecting variables – may want to remind them of the required input (i.e. numbers separated by spaces, not commas or semicolons)
  + When I input an invalid number for the number of stratifying variables (i.e. a number larger than the # of variables in the dataset), it exits the function with the error “You have to select some stratifying variables.” May want to add an error message similar to the “Invalid input, please try again” that allows them to re-enter valid numbers, and inform them that the column numbers must be between 1 and dim(data[2])
  + Might say ‘Hit <Enter>’ instead of ‘Hit <Return>’
  + Would give some guidance on typical #s of strata to choose. The danger is obviously always that once you give guidance, people will likely just go with that as the default. But, without it, people unfamiliar with stats aren’t going to have any intuition about what a reasonable number might be – or even the appropriate order of magnitude – 5? 500?
    - “Error: The number of strata must be a positive number greater than one.” Can you allow them instead to re-enter a valid number rather than being exited from the function?
    - May want them to specify their sample size before this step, because the max number of possible strata depends on n. In a CRT, you need *at least* 2 sample schools per strata (one treatment and one control), but preferably more in order to estimate the standard error within each strata. May want to ask Beth if she wants to add any specific guidance here – both in the instruction text and relevant error messages.
  + Would be nice if you could print the strata results in the typical mean (sd) format (similar to how it’s presented in the heat map) instead of mean / sd. The current format makes it a bit hard to distinguish the columns from each other, and looks a bit like the / means division.
  + In the heat map and/or in the documentation/vignette, specify that the numbers displayed are mean (sd).
  + Might want to add some additional text to explain that responding “No” to “Would you like to go back and specify a different number of strata?” ends the stratifying process, and they can use the output to proceed to recruit(). Users that are new to R will probably be confused by an abrupt end to the function/guidance and not realize it resulted in an object they can now use.

### recruit()

* This might not be possible/straightforward, but it would be nice if the function could prompt RStudio to open a Finder window for the user to select the folder they want to save the files in. I made a mistake the first time by not adding a / at the end of my file path – users unfamiliar with coding may have trouble here. Relatedly, you may want to include text that tells them by default it will save in their current working directory, so they don’t have to specify a file path unless they want to change where it’s saved.
* Should add an error when they specify a sample size that exceeds the total number of population units
* In the dataframe printed after “# of units to recruit,” renaming the first two columns to something like “stratumID” and “population\_units” may be helpful
* May be helpful to include school name, address, phone number, and website in the .csv files that are created, so users don’t have to merge that info themselves. My impression was that this info was included in files outputted by the Generalizer web version, but I could be wrong.

### assess()

*I think you mentioned this part is still under construction, so I’m sure you just haven’t gotten to many of these things but I thought I’d go ahead and flag what I noticed!*

* Don’t know how hard this would be to make for a general case, but would be helpful if assess\_wrap() returned the tibble from g\_overview (in the vignette) directly (i.e. accompany the gen indexes with an id column based off of the grouping variable).
* Relatedly, would be nice if the output of gen indices also includes a separate column that categorizes the values as “best”, “okay”, “worst”
* trial argument required to be given in quotes – should indicate this requirement in documentation and/or make the input more flexible if possible
* not specifying a join\_var will still run, but it returns gen indices that are zero for everything but TX (the only state that had sample schools).

assess\_wrap(sample, inference\_pop, grouping\_var = "st")[[2]]

Joining, by = "ncessch"

[1] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

[7] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

[13] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

[19] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

[25] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

[31] 0.0000000 0.0000000 0.9776633 0.0000000 0.0000000 0.0000000

[37] 0.0000000 0.0000000

* in the ?assess documentation:
  + Provide info on the output that’s provided by assess()
  + Include info on the generalizability index and how to interpret it (similar to what’s in the vignette)
  + Description for is\_data\_disjoint argument says see details for more information, but no details are included there
* Seems like there are a few things related to the is\_data\_disjoint argument to be worked out. In general, it would be good to have more guidance on the format the data needs to be in before using assess() and assess\_wrap()
  + Is it possible to allow the input of assess() to require *either* sample and population (as in assess\_wrap) *or* trial and data?
  + Seems like assess\_wrap() will only work if sample is a subset of the population and sample covariate data can be obtained by joining with the population dataset. Is it possible to allow the sample input to be a data frame with both IDs and the selection covariate data? Perhaps if the function recognizes it is a dataframe with more than just the ID column, you can create a trial indicator variable in both sample and population and then use inner\_join(sample, population, join\_var).
  + Might want to allow a selection\_covariates argument in assess\_wrap() also, with the default being that it selects all non-ID columns.
* In general, probably need more documentation and error messages related to assess(), but I know that’s not necessarily straightforward b/c of the gen index calculations that are occurring in the background… But one example I encountered when attempting to run the following code to use assess() directly rather than assess\_wrap()

sample\_full <- left\_join(sample, inference\_pop) %>%

mutate(trial = 1)

inference\_pop <- inference\_pop %>%

mutate(trial = 0)

combined <- rbind(sample\_full, inference\_pop)

gen\_results <- assess("trial", selection\_vars,

combined, is\_data\_disjoint = FALSE)

Error in integrate(function(x) sqrt(kg(x, dat1B) \* kg(x, dat2B)), -Inf, :

non-finite function value

In addition: Warning messages:

1: Unknown or uninitialised column: `unitid`.

2: Unknown or uninitialised column: `unitid`.

3: Unknown or uninitialised column: `unitid`.

4: Unknown or uninitialised column: `unitid`.

* After mutating inference\_pop to include a trial variable, I later re-ran assess\_wrap(sample, inference\_pop) and it threw a similar error to above and the algorithm wouldn’t converge (code and error below). At first I thought it was because all observations in the dataset had the same level (and therefore runs into problems in the propensity score calculations), but I created another test variable with all one level, and it performed fine. Once I removed the trial variable it worked fine again – so it’s something with how assess\_wrap() interacts with a variable named “trial”.

assess\_wrap(sample, inference\_pop)

Joining, by = c("ncessch", "trial")

Error in integrate(function(x) sqrt(kg(x, dat1B) \* kg(x, dat2B)), -Inf, :

non-finite function value

In addition: Warning messages:

1: Unknown or uninitialised column: `unitid`.

2: Unknown or uninitialised column: `unitid`.

3: Unknown or uninitialised column: `unitid`.

4: Unknown or uninitialised column: `unitid`.

5: glm.fit: algorithm did not converge