**Accuracy coding rules**

LAYOUT

* In spreadsheet generated by ‘dataWrangling’ R script, make two columns next to Keys called ‘AccuracyScore’ and ‘No.Errors’

IN GENERAL

* Mark correct key presses as ‘1’ and incorrect as ‘0’ in the AccuracyScore column
* Total the number of errors made for each word and mark the total in the AccuracyTotal column. Repeat the total for each row associated with that word/trial
* Mark ‘return’ key presses as ‘na’ in the AccuracyScore column but put the accuracy total score in the AccuracyTotal column
* Focus on key press order, not letter position, as this will inflate errors
* Mark the first time a key is pressed (in case a key is pressed twice after correcting an error) and the second press as incorrect
* The key press following ‘backspace’ must always be marked as 0

TRICKY ONES

* Where a mistake is made after the last letter key but before pressing return:
* Code last letter press as 0 and all following key presses – mark as 1 error
* If a couple of keys are pressed before the error is realised and backspace is pressed, mark however many keys are pressed before backspace as correct/incorrect. If a key/keys are pressed again after backspace, ignore the second press
* If correct keys are pressed but in the wrong order, mark both as incorrect regardless of whether these were corrected (e.g. CORETX instead of CORTEX) and score as 1 error
* If someone tries to press a letter but gets it wrong twice (e.g. B-BACKSPACE-V-BACKSPACE-C), mark this as 2 errors
* Where a letter is missed and the following letters are in the correct order, mark the letters in the correct order as correct. If the error is corrected and the letters are pressed again, ignore the second presses. Score as 1 error
  + An exception to this is when the first letter is skipped. If the first key pressed is not correct, code as ‘0’ and mark as 1 error.
* If someone incorrectly presses backspace when the key before was actually correct, mark the first key press as correct and the same key pressed again after backspace as a 0. This would be 1 error
* If too many keys are deleted so they have to press a key again that they already pressed, mark the extra delete/s as an error
* If ‘BACKSPACE’ is pressed an incorrect number of times to correct their error, count an error for each correct letter they delete
* If an extra key is pressed, mark the extra incorrect key as ‘0’ and the correct one as ‘1’. Mark as 1 error.
* If a letter is missed, but then corrected by pressing backspace, mark this as an error but IF the additional letter does NOT immediately follow ‘backspace’, AND it is the first time they’ve pressed this letter, code the letter as ‘1’ . All keys following ‘backspace’ should be coded as ‘0’ (because presses following an error correction are likely to be slower (e.g., Q-W-R-BACKSPACE-BACKSPACE-U-A-R… for ‘quarry’ – the ‘A’ would be coded as ‘1’)
* If an incorrect key is pressed, which is a letter used later in the word, e.g. ‘S-U-R-E-F-E-R’ instead of ‘surfer’, mark both ‘E’ presses as ‘0’ and mark as 1 error. This is because having pressed the ‘E’ twice when there is only one in the word, ‘surfer’ this could be considered practice and there may be an advantageous RT for the second press.
* If a key is pressed twice but it is clearly because the letter has not displayed on the screen (e.g., someone has typed fast enough that the letter did not show as having been pressed on their screen, and so they would naturally think they missed it and press it again), mark the extra press as ‘0’ but don’t include as an error
* If an extra key is pressed after ‘return’, but it is associated with the same trial (i.e., not the start of the next trial), code as ‘0’ and mark as one error, because this means the participant likely pressed this extra key while pressing the ‘return’ key
* If a letter is missed but this shows up in the ‘typedWord’ column without an error correction, mark as 0 errors – this means what appeared on screen for the participant was correct and a key press has not been recorded
  + If this happens, remove the IKI for the letter following an ‘invisible’ letter and change to ‘NA’. E.g., a participant has typed ‘portal’ but we only have the following key presses: ‘P-O-R-T-L’. In this case the IKI for the ‘L’ would be changed to NA