**Creating functions in R**

1. Read student\_assignment\_data.csv into a list
2. Create a function called add\_cfa\_col (.data)that adds a column in the list indicating whether the student got the correct answer on their first attempt (“cfa”). Input the data, output a list with the extra column
3. Create a function summarize\_student\_performance(.data, group\_col) with two input parameters data that summarizes student performance group by either student\_id, assignment\_id, or question\_number. Return a list with the summary

Note that in this case group\_col is not a keyword, i.e. the problem is, group\_by already expects a “bare” symbol naming that column; if you tell it to group\_by(group\_col), it will look for a column named “group\_col” in the data, fail to find that column, and throw an error. You need to tell it to *translate the parameter you send in to the name of a column.* To do this, we need two functions from rlang: ensym and !! (pronounced “bang-bang”). First, = we tell our function “think of this parameter as a symbol” using the ensym function, rlang::ensym(group\_col). Then we’ll tell group\_by to process the code we gave it rather than looking for a column named “rlang::ensym(group\_col)” with !!, !!rlang::ensym(group\_col). I think of !! as telling group\_by “It’s not group\_col, but it’s also not *not* group\_col.”

See beginning of the function below

summarize\_student\_performance <- function(.data, group\_col) {

.data<- group\_by(.data, !!rlang::ensym(group\_col))

# Complete the rest

)

return(mySummary)

}

Example:

mysummary<- summarize\_student\_performance(student\_assignment\_data,question\_number)

View(mysummary)

1. Summarize with multiple parameters: Modify the summarize\_student\_performance function by including three dots … to include multiple parameters at once like this

summarize\_student\_performance <- function(.data, …){

.data<- group\_by(.data, !!!rlang::ensyms(…))

# Complete the rest

)

return(mySummary)

}

Note the difference you have three !!! in the new function versus 2 !! in the old one, as well as ensym vs ensyms ( one extra s)

Call the function

mysummary<-summarize\_student\_performance(student\_assignment\_data,assignment\_id, question\_number)

1. Create function to mutate the integer columns correct, view\_hint and requested\_solution to logical (TRUE/FALSE).
2. More with !!rlang. Take the mtcars dataset that group the dataset by any column (dist, speed, etc.) and summarizes by the mean of the column. Notice that column name is a variable use knowledge of point 2 and 3 above.
3. More exercises in function. Write both\_na() function that takes to vectors of the same length and returns the number of positions that have an NA in both vectors.
4. Create a function that finds the match of the letters AC in strings of characters, the output should be a list of the indexes

Inputs: strings like the one below

strings<- c("abcd","ACdc","zcac","dcAC","myActc", "tenAC", "ACten")

mymatch<-c(“AC”)

output: indexes of the matched strings