Guerrilla\_Automation.R

tiqu2

Wed Apr 08 18:05:47 2020

#uses all of the scripts below to accomplish the following:  
# 1. import\_input\_control : import paths to excel inputs  
# 2. gather\_input\_data : gather data from excel inputs  
# 3. aggregate\_input\_data : aggregate input data into results  
# 4. output\_results : output data and results into a single excel file  
# requires packages readxl, dplyr, and writexl, which can be installed using > install.packages(c("readxl","writexl", "dplyr"))  
main = function(){  
 library(readxl)  
 library(writexl)  
 library(dplyr)  
   
 #Run each step in the process successively.  
 #Alternatively, to accomplish the same thing, simply run the following (which runs back through each step):  
 #output\_results()  
   
 input\_control = import\_input\_control()  
 input\_list = gather\_input\_data(input\_control)  
 aggregate\_list = aggregate\_input\_data(input\_list)  
 output\_results(aggregate\_list)  
   
   
}  
  
#Returns the Input Control  
# filepath - the file path to the Input Control spreadsheet  
import\_input\_control = function(filepath = "Excel Files/Input Control.xlsx"){  
 library(readxl)  
   
 #9001 is chosen arbitrarily, we just need a large number and 9001 is OVER 9000!!!  
 input\_control = read\_excel(filepath, sheet = "Input Control",range="B4:D9001")  
   
 #removes extra rows at the bottom and converts from tibble to dataframe  
 input\_control = data.frame(input\_control[!is.na(input\_control$Filepath),])   
   
 return(input\_control)  
  
}  
  
#Returns a list with two data frames: input\_control - tracks the excel input files, input\_data - contains the granular input data  
# input\_control - dataframe which tracks inputs with columns "Filepath", "Sheet", and "Range"  
gather\_input\_data = function(input\_control = import\_input\_control()){  
 library(readxl)  
 #Using a list to store input data because lists can store multiple data frames  
 #One data frame will show which Excel files imported successfully (+if not: why not?) and the other will have the input data.  
 input\_list = list()  
   
 #initialize data frame to store input data  
 input\_data = data.frame(matrix(nrow = 0, ncol = 5))  
 names(input\_data) = c("Filepath", "Sheet", "Range","Account Number", "Amount")  
   
 #File\_Is\_Good and Error\_Message are used to track which files came in successfully and if not, what went wrong  
 File\_Is\_Good = logical()  
 Error\_Message = character()  
   
 #loop through each row of the input\_control data.frame  
 for(i in 1:nrow(input\_control)){  
   
 #assign row from input\_control to variables with the same column name  
 Filepath = input\_control[i,"Filepath"]  
 Sheet = input\_control[i,"Sheet"]  
 Range = input\_control[i,"Range"]  
   
 #a tryCatch is used to detect errors in file import, Error\_Handle is used to temporarily track error status for each input\_control row  
 Error\_Handle = tryCatch({  
 #pull a single sheet and store temporarily in input\_temp  
 input\_temp = read\_excel(path = Filepath, sheet = Sheet, range = Range)  
   
 #remove NA values from input\_temp which may occur if the range is set to be generously large  
 input\_temp = input\_temp[!is.na(input\_temp$'Account Number'),]  
   
 #add Filepath, Sheet, and Range columns to input\_temp  
 input\_temp = cbind(Filepath, Sheet, Range, input\_temp)  
   
 #append input\_temp to input\_data  
 input\_data = rbind(input\_data, input\_temp)  
   
 #Import successful  
 Error\_Handle = c(TRUE,"")  
   
 },  
 error = function(err){  
 #Import unsuccessful  
 return(c(FALSE, as.character(err)))  
 })  
   
 #Assign new import control columns to track which imports were successful  
 File\_Is\_Good[i] = as.logical(Error\_Handle[1])  
 Error\_Message[i] = Error\_Handle[2]  
   
 }  
   
   
 #Add File\_Is\_Good and Error\_Message columns to the input\_control dataframe  
 input\_control = cbind(input\_control, File\_Is\_Good, Error\_Message)  
   
 #Assign the appropriate data frames to the input\_list  
 input\_list[["input\_control"]] = input\_control  
 input\_list[["input\_data"]] = input\_data  
   
 return(input\_list)  
}  
  
# Summarizes input data and returns a list of relevant data frames  
# input\_list[[1]] - dataframe tracking input from excel files  
# input\_list[[2]] - dataframe with input data from excel input files  
aggregate\_input\_data = function(input\_list = gather\_input\_data()){  
   
 library(dplyr)  
   
 #initialize list to store output data - a list will be helpful for interacting with writexl  
 aggregate\_list = list()  
   
 #first list item: a report on which input files came through properly  
 aggregate\_list[["input\_control"]] = input\_list[["input\_control"]]  
   
 #second list item: the input data itself  
 aggregate\_list[["input\_data"]] = input\_list[["input\_data"]]  
   
 #third list item: a summary of the data; I use the British spelling of 'summarise' because it makes me feel more intelligent (though either spelling works).  
 aggregate\_list[["summary"]] = data.frame(input\_list[[2]] %>% group\_by(`Account Number`) %>% summarise(Amount = sum(Amount)))  
   
 return(aggregate\_list)  
}  
  
#Output the results into a single excel file  
output\_results = function(aggregate\_list = aggregate\_input\_data()){  
 library(writexl)  
   
 #Alternate version:  
 #  
 write\_xlsx(aggregate\_list, "./Excel Files/R Output/Results.xlsx")  
   
 write\_xlsx(aggregate\_list, paste("./Excel Files/R Output/Results ", format(Sys.time(), "%Y%m%d %s"), ".xlsx", sep=""))  
   
   
}