Excel Copy-Paste Test

Hava Blair

July 2, 2020

Contents

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(tidy = TRUE)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff=60))

library(tidyverse)
library(readxl)
library(glue)
```

1 Read in excel files

You can have as many sheets (tabs) as you want in each excel file

```
# fill in directory (folder) where your raw excel files are
xl_files <- dir("./raw_excel/")

# save full file path to each of your excel files
xl_paths <- paste0("./raw_excel/", xl_files)

# function to create dataframe with col for filepath and col
# for sheetnames. Each row in this dataframe represents a
# single 'sheet' (tab) from your raw excel file. You will get
# one dataframe per excel file in your original list.  both
# cols should be character type so bind_rows() below doesn't
# complain about unequal factor levels.
make_df <- function(file_path) {
    df <- data.frame(sheet = excel_sheets(path = file_path))
    df$path = file_path
    df$sheet <- as.character(df$sheet)
    df
}

# apply make_df to every excel file you want to process
# creates list of dfs
sheets_list <- map(xl_paths, make_df)
```

2 Create a dataframe of file paths and sheet names

```
# bind list of dfs together into a single df path is first
# col, sheets second col
sheets_path_df <- bind_rows(sheets_list)
sheets_path_df <- sheets_path_df %>% select(path, sheet)

# unequal factor levels: coercing to character binding
# character and factor vector
```

3 Extract the fluorescence data from your excel files

```
# function to read only fluorescence data from each sheet in
# every excel file

read_sheets <- function(path, sheet) {

    raw_data <- read_excel(path = path, sheet = sheet, range = "B26:N34")
    raw_data %>% rename(Template = ...1) %>% mutate(`1` = as.character(`1`),
        `2` = as.character(`2`), `3` = as.character(`3`), `4` = as.character(`4`),
        `5` = as.character(`5`), `6` = as.character(`6`), `7` = as.character(`7`),
        `8` = as.character(`8`), `9` = as.character(`9`), `10` = as.character(`10`),
        `11` = as.character(`11`), `12` = as.character(`12`))
}

# save the fluor data as a list column
fluor_import <- sheets_path_df %>% mutate(fluor_data = pmap(sheets_path_df,
    read_sheets))
```

4 Create plater templates

```
# read in blank plater template
blnk_template <- read.csv("./templates/template_test.csv", stringsAsFactors = FALSE,
    blank.lines.skip = FALSE)

colnames(blnk_template) <- c("Template", "1", "2", "3", "4",
    "5", "6", "7", "8", "9", "10", "11", "12")

# function to add fluorescence data to plater template

template_fun <- function(fluor_data) {
    # load blank plater template
    blnk_template <- read.csv("./templates/template_test.csv",
        stringsAsFactors = FALSE, blank.lines.skip = FALSE)
    # set col names
    colnames(blnk_template) <- c("Template", "1", "2", "3", "4",
        "5", "6", "7", "8", "9", "10", "11", "12")

    # bind template and plate data
    template <- rbind(blnk_template, fluor_data)

    # return completed template
    template
}

# plater_templates col contains completed template dfs
# export_paths col contains
incl_templates <- fluor_import %>% mutate(plater_templates = map(fluor_data,
    template_fun), export_paths = as.character(glue("./auto-templates/plater-{sheet}.csv")))
```

5 Save completed templates to a new sub-directory

```
# function to save each template as a CSV using the specified
# export path
save_templates_fun <- function(plater_templates, export_paths) {
    write.csv(plater_templates, export_paths, row.names = FALSE)
}

# create a sub-directory in your working directory called
# 'auto-templates'
dir.create("./auto-templates/")
```

```
## Warning in dir.create("./auto-templates/"): '.\auto-templates' already exists
```

```
# write all templates to CSV, save in auto-templates
# sub-directory
save_templates <- incl_templates %>% mutate(saved = walk2(plater_templates,
    export_paths, save_templates_fun))
```