GitHub Repository

STA3100 Final Project Report
Spring 2024

Ananya Sista
Elle Strauss

Abstract

InDepth is an attendance sheet parser created specifically for the University of Florida organization Women in Computer Science and Engineering (WiCSE), but has features that allow it to be extended to other organizational use. InDepth interfaces directly with Google Sheets, and takes in the URL of an attendance Google Sheets workbook that an organization uses and parses through the data of a specified individual attendance sheet. That data is then appended to a total attendance sheet, a sheet containing unique members and their attendance counts, and an email list to upload to the organization ListServ. Future extensions to the project include UI enhancements, further documentation, and options that allow the software's functions to be more customizable to the context of the user.

Introduction

Almost every club at the University of Florida collects attendance data during events and general body meetings. This data is used to keep track of member attendance and to generate demographic data at the end of the year. However, organizations often have different formats for their attendance forms and collect different types of data, so a program created for this purpose is often not reusable across campus clubs. For the purpose of this project, we will be using the attendance form structure that UF WiCSE (University of Florida Women in Computer Science and Engineering) implements to create an interactive data dashboard that will achieve the following objectives: parse through individual event attendance forms, add data to a sheet that holds cumulative attendance data, associates events with the type of event they are, tracks unique members and the number of events they attend, catches errors in the field that is used to identify unique members, and generates demographic charts of member information.

Prior to creating InDepth, the WiCSE attendance parsing process was a time-consuming and tedious process. It required the Secretary to set up elaborate excel formulas (as pictured in Figure 1 below), manually copy and paste each individual attendance sheet, and apply various other formulas to graph demographic data.

Figure 1: Previous Excel Formula Used to Get Statistics

To set up the attendance workbook itself took around 2 hours, with an added 15 minutes per event attendance sheet. WiCSE, as other campus organizations, often has over 30 events in a semester with 2-3 happening each week, creating a very time intensive task to do constantly. InDepth provides a mechanism to streamline and automate this process, also creating a parsing system that is less prone to error.

Program Workflow

As InDepth was created anticipating a certain attendance sheet structure, a detailed explanation is required for optimal use of the software. The workflow, including workbook setup and expected attendance sheet structure, is as follows:

1. The Attendance Workbook should be made in Google Sheets, and have the following sheets:
    a. A master attendance sheet: used to store a copy of each event's attendance data on the same sheet
    b. A uniques sheet: used to record unique members and the number of events each attends
    c. An error sheet: used to notify the user of UFIDs that are incorrect (e.g. having alphabetical characters, not being 8 digits long)
    d. The event attendance sheets: the attendance response sheets should also be stored in this workbook, each having a unique name

2. Individual event attendance sheets should have the following fields:
    a. First Name: first name of the attendee
    b. Last Name: last name of the attendee
    c. UFID (Ex: 12345678): the UFID of the attendee
    d. Please enter the email you would like to receive the WiCSE newsletter to!
    e. Leave this question blank if you are already on it or don't wish to be added.: attendees answer this question if they want their email to be added to the WiCSE ListServ
    f. Year (optional field): the attendees year in school (e.g. Freshman, Junior)
    g. Major: the attendees major
3. To use InDepth, the user first responds to the following fields:
    a. Enter Attendance Workbook URL: the URL to the Google Sheets workbook as defined in step 1
    b. Enter Master Worksheet Name: the name of the worksheet as described in step 1a
    c. Enter Event Worksheet Name: the name of the individual event attendance sheet, as described in step 1d
    d. Enter Unique Members Worksheet Name: the name of the worksheet used to store unique member data, as described in step 1b
    e. Enter Error Worksheet Name: the name of the worksheet used to store erroneous UFIDs, as described in step 1c
    f. Select Event Type: the type of event that is being parsed; options include: Social, Career, Workshop, GBM (general body meeting), Technical, and IM (internal mentorship)
4. The user then clicks the "Submit!" button. Upon submission, the program performs the following:
    a. The UFIDs in the event attendance sheet are checked to ensure they are 8 digit numbers. If any are found that don't meet the criteria, they are copied into the specified error worksheet and the parsing process stops. The user can then correct the error and redo the submission.
    b. Once UFIDs are verified, the event attendance is appended to the master sheet. A row is first written with the event type, event worksheet name, and the event attendance count. Then, the data from the attendance sheet is written, followed by a blank row for visual purposes.
    c. The event sheet is then parsed to identify unique members. Going row by row, if the UFID is already present in the Unique Members sheet, their total attendance and the associated event category attendance is incremented. If the UFID is not present, their first name, last name, UFID, year, and major are appended to the sheet, and associated attendance incremented.
5. If the user selects the "Create ListServ Email List" button, all of the unique email addresses, along with the associated first and last name, are written to a new google sheet titled "ListServ". This sheet can then be downloaded as a CSV file and uploaded to the appropriate ListServ.
6. The user can also generate the top members, in terms of their attendance count. The input is restrained to not go below 0, and can take any higher value (if the value is higher than the number of unique members, NA values are output). The program then returns the specified number of members, in decreasing order of total events attended.
7. Finally, the user can generate demographic bar charts that chart the attendance count vs. the total attendance by category, year, and major of the different events. The user has the option to select the metric category, choosing between total (all events), or by event category.

Conclusion

InDepth reduces the time necessary to parse attendance sheets by 85%, with the initial setup of a workbook taking only 10 minutes, and each individual event parsing taking 2 minutes. The software will be put into practical use during the 2024-25 school year, with improvements being made as need arises. Before publishing the software, we have several enhancements in mind to make its use more efficient and intuitive:

- Update the UI to make it more visually appealing

- Include a guide to the software, and a diagram of the expected workbook set up to make InDepth more user friendly
- Create more forms of statistical breakdown, such as tables alongside the bar graphs
- Make our graphs more interactive, such as having the bar graph values appear as the user mouses over them
- Create separate tabs in the application to have functions better separated
- Create comprehensive documentation of both InDepth's use and code, so that other organizations are able to adapt it to their specific needs

Contribution Statement

For this project, Elle and Ananya both collaborated on writing the code and designing the user interface. They used GitHub to keep track of changes and allow for easy access. During the development of the project, Ananya and Elle would meet in-person to work on the project, which allowed for easy communication and helped each other write each part of the code. This enabled both to gain experience with writing in R Shiny and figuring the logic behind the functions.

References

Allaire, J., Xie, Y., Dervieux, C., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2023). *Rmarkdown: Dynamic documents for r.* https://github.com/rstudio/rmarkdown

Bache, S. M., & Wickham, H. (2022). *Magrittr: A forward-pipe operator for r.*

Bryan, J. (2023). *googlesheets4: Access google sheets using the sheets API V4.*

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2024). *Shiny: Web application framework for r.* https://shiny.posit.co/

Grolemund, G., & Wickham, H. (2011). Dates and times made easy with lubridate. *Journal of Statistical Software*, *40*(3), 1–25. https://www.jstatsoft.org/v40/i03/

R Core Team. (2024). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing. https://www.R-project.org/

RStudio Team. (2020). *RStudio: Integrated development environment for r.* RStudio, PBC. http://www.rstudio.com/

Sievert, C., Cheng, J., & Aden-Buie, G. (2024). *Bslib: Custom 'bootstrap' 'sass' themes for 'shiny' and 'rmarkdown'.* https://rstudio.github.io/bslib/

Tang, Y. (2022). *Shinyjqui: 'jQuery UI' interactions and effects for shiny.*

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis.* Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H. (2023a). *Stringr: Simple, consistent wrappers for common string operations.* https://stringr.tidyverse.org

Wickham, H. (2023b). *Tidyverse: Easily install and load the tidyverse.* https://tidyverse.tidyverse.org

Wickham, H., FranC'ois, R., Henry, L., MC<ller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation.* https://dplyr.tidyverse.org

Xie, Y. (2023a). *Knitr: A general-purpose package for dynamic report generation in r.* https://yihui.org/knitr/

Xie, Y. (2023b). *Tinytex: Helper functions to install and maintain TeX live, and compile LaTeX documents.* https://github.com/rstudio/tinytex

Xie, Y. (2024). *Bookdown: Authoring books and technical documents with r markdown.* https://github.com/rstudio/bookdown