Paper 165-2023

A Dash of SAS®, a Pinch of R: Cooking up Dashboards Using Two Very Different Programming Languages

Joshua J. Cook, M.S., ACRP-PM, CCRC

Andrews Research & Education Foundation (AREF)

University of West Florida (UWF)

ABSTRACT

Dashboards are often utilized by businesses, schools, and hospitals as a key tool to organize, analyze, and visualize data that is subject to frequent updates and changes. In this way, effective dashboards enable leaders to make data-informed decisions regarding their organizations. However, there are many approaches to "cooking up" effective dashboards that utilize both proprietary and open-source software, including the SAS® and R programming languages. Using base-SAS® code as your primary ingredient, one might build an effective dashboard using a colorful combination of the Output Delivery System (ODS), PROC PRINT, PROC SGPLOT, and PROC SGPIE. In someone else's kitchen, they might include R packages as the main ingredients for their dashboard, such as quarto, readxl, tidyverse, and gt.

As Julia Child once said, "You don't have to cook fancy or complicated masterpieces – just good food from fresh ingredients." With the same notion, this paper offers a direct comparison of two dashboards prepared from two fresh, but very different ingredients: SAS® and R. This paper was written with the intent to duplicate the dishes as much as possible given the flavor (strengths) and aromas (weaknesses) of each ingredient, with emphasis being placed on comparing the setup, steps, and code of each.

INTRODUCTION

Dashboards are defined by Oxford as a "graphical summary of various pieces of important information, typically used to give an overview of a business." In practice, dashboards are being used by more than just businesses, including hospitals, schools, and research organizations. But what exactly makes a dashboard distinct from a graph or other standard visualization tool? The key difference is in the data update capabilities of dashboards, which are typically much more frequent and often real-time compared to a static visualization. Due to this, dashboards have become widely used and highly influential analytic tools that offer real-time insights to inform decision makers.

Just like a family recipe, data scientists have developed many unique approaches to dashboarding that utilize different programming languages and techniques, including Statistical Analysis System (SAS®) and R. SAS® is a proprietary statistical programming language that has become an industry standard in healthcare, research, and business. On the other hand, R is an open-source programming language for statistical reporting that is like the S language. R is not an industry standard but is gaining popularity and has recently been under development for pharmaceutical and regulatory use. A key difference between SAS® and R is the way by which new functions and features are made available to users. SAS® receives updates managed by the SAS® Institute, while base-R is maintained by the R Foundation for Statistical Computing and many functions or capabilities are added through R's community package system. This alone has led to the development of many ways to create dashboards with just these two programming languages.

Both base-SAS® and R have basic functions to make dashboards that can be easily adopted by even novice programmers. A standard dataset will be used in combination with base-SAS® and R code to create two dashboards that are as similar in display and features as possible so that the flavor (strengths) and aromas (weaknesses) can be evaluated for each method. In SAS®, this is accomplished through the Output Delivery System (ODS), with a combination of SAS® statements such as PROC PRINT, PROC SGPLOT, and PROC SGPIE. In R, this is accomplished through the Quarto® package, with a combination of data structures prepared using readxl, tidyverse, and gt packages. Quarto® is an open-source scientific and technical publishing system that is often described as "the next generation of R Markdown." Quarto® is built into the RStudio® Integrated Development Environment (IDE) and offers a simple solution to

dashboarding by being able to execute R code using Knitr which is thus able to create data-rich and visually appealing dashboards in a variety of formats.

EXAMPLE DATA

The dashboard examples in this paper utilize an in-house, but publicly accessible, dataset of publication metrics (ex: publication titles, journals, authors). This dataset was generated by querying the National Center for Biotechnology Information (NCBI) Entrez Programming Utilities (E-utilities), which allows for easy programmatic access to PubMed. Data relevant to the home institution is compiled within an excel sheet. Additionally, an excel sheet with all site author names was manually compiled to be used as a filter for the publication metrics (i.e., all authors that are not part of the home institution are filtered out). The publications dataset consists of 2147 observations of the 14 variables listed in Table 1.

Variable	Meaning
pmid	The unique PubMed identifier assigned to the article.
doi	The unique Digital Object Identifier assigned to the article.
title	The full title of the article.
abstract	The full abstract of the article.
year	The article publication year (electronic).
month	The article publication month (electronic).
day	The article publication day (electronic).
jabbrv	The abbreviated name of the publishing journal.
journal	The full name of the publishing journal.
keywords	Keywords associated with the article.
lastname	The author's last name.
firstname	The author's first name.
address	The address affiliated with the author.
email	The email affiliated with the author.

Table 1. All Variables Included in the Publications Dataset

COOKING IN THE SAS® KITCHEN

SETTING UP YOUR KITCHEN

This demonstration utilized SAS® 3.81 (Enterprise Edition) and was accessed on SAS® Studio via SAS® OnDemand for Academics (ODA). Only procedures from base-SAS® were utilized. The two excel spreadsheets (Publications and Authors) were uploaded to the SAS® Studio server and imported as datasets (PUBS and AUTHORS).

RECIPE OVERVIEW (STEPS)

Building a dashboard using base-SAS® software can be done with any combination of SAS® tables and plots. In this tutorial, the dashboard in Figure 1 is built using tables that were previously wrangled using PROC SQL and displayed using PROC PRINT, as well as plots using PROC SGPLOT and PROC SGPIE. Follow these steps to successfully construct a basic dashboard using these data structures:

1. Setup your kitchen as explained above
2. Specify an ODS HTML5 statement to set the file path, name, and format (html in this example)
3. Specify the dashboard title and any font customizations
4. Specify and ODS LAYOUT statement to set the number of rows and columns that will be used to organize the dashboard layout
5. Specify an ODS REGION statement to control where the following output will be placed with respect to the layout (i.e., designating which rows and/or columns each data structure will go to)
6. Include any PROC PRINT, SGPLOT, SGPIE, etc. procedures that encompass the data structures you would like placed in each ODS REGION
7. Specify an ODS LAYOUT END statement to end the dashboard layout
8. Specify an ODS HTML5 CLOSE statement to render the results to the dashboard file (html in this example)

THE MEAL (EXAMPLE #1 – (1X3) DASHBOARD LAYOUT)

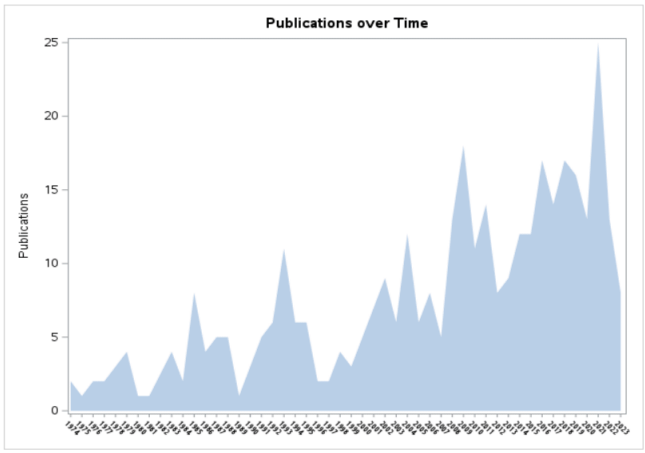
AREF Publications Dashboard

Total Publications

Obs	title_count
1	360

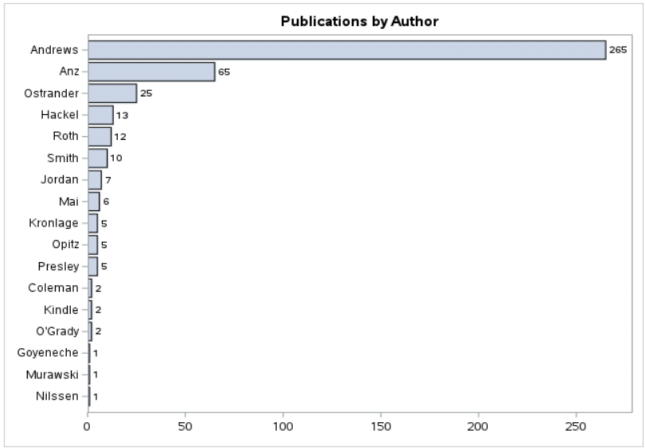
Publications over Time

year	title_count
1974	2
1975	1
1976	2
1977	2
1979	4
1980	1
1981	1
1983	4
1984	2
1985	8
1986	4
1987	5
1988	5
1989	1
1991	5
1992	6
1993	11
1994	6
1995	6
1996	2
1997	2
1998	4
1999	3
2000	5
2001	7
2002	9
2003	6
2004	12
2005	6
2006	8
2007	5
2008	13
2009	18
2010	11
2011	14
2012	8
2013	9
2014	12
2015	12
2016	17
2017	14
2018	17
2019	16
2020	13
2021	25
2022	13
2023	8



Publications by Author

lastname	COUNT
Andrews	265
Anz	65
Ostrander	25
Hackel	13
Roth	12
Smith	10
Jordan	7
Mai	6
Kronlage	5
Opitz	5
Presley	5
Coleman	2
Kindle	2
O'Grady	2
Goyeneche	1
Murawski	1
Nilssen	1



Publications by Journal

jabbrv	COUNT
Am J Sports Med	76
J Orthop Sports Phys Ther	32
Arthroscopy	29
Orthop J Sports Med	21
Clin Sports Med	17
Sports Health	12
J Shoulder Elbow Surg	11
Int J Sports Med	9
J Am Acad Orthop Surg	8
J Bone Joint Surg Am	8
Med Sci Sports Exerc	8
Orthopedics	8

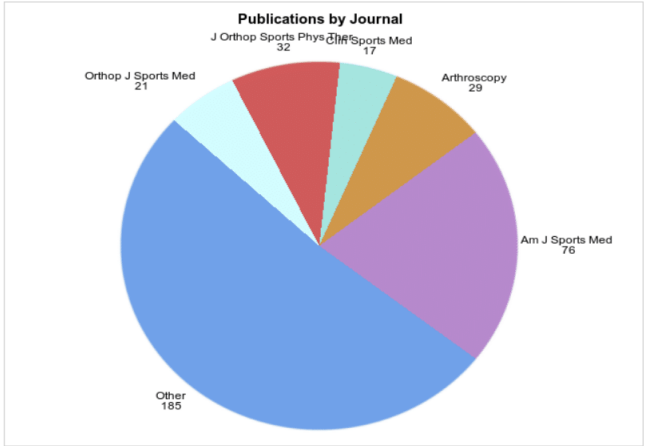


Figure 1. A Simple Dashboard Prepared with Base-SAS® Code as the Primary Ingredient

THE INGREDIENTS (CODE):

```
ods html5 PATH="/home/u60852158/WUSS 2023"
body="Dashboard - HTML (1 x 3) Layout.html"
(url=none);

Title1 font=impact bold h=12 c="#4F6FA1" "AREF Publications Dashboard";

ods layout gridded rows=1 columns=3; /* HTML 1x3 Layout */

ods region ; /* Row 1 Column 1 */

/* Data Step */
data work.pubs;
    set pubs;
run;

data work.pubs_numeric_year;
    set pubs;
    year_numeric = input(year, best.);
run;

data work.unique_titles;
    set work.pubs_numeric_year;
    if first.title and first.lastname;
run;

/* Total Publications  */
proc sql;
    create table work.joined_data as
    select PUBS.title
    from PUBS, authors
    where PUBS.lastname = authors.lastname;
quit;
proc sql;
    create table work.total_unique_titles as
    select count(distinct title) as title_count
    from work.joined_data;
quit;
proc print data=work.total_unique_titles;
      title "Total Publications";
run;


ods region ; /* Row 1 Column 2 */

/* Publications over Time */
proc sql;
    create table work.joined_data as
    select PUBS.year, PUBS.title
    from PUBS, authors
    where PUBS.lastname = authors.lastname;
quit;
proc sql;
    create table work.title_counts_year as
```

```
    select year, count(distinct title) as title_count
    from work.joined_data
    group by year
    order by year;
quit;
data work.title_counts_zero;
    set work.title_counts_year;
    title_count_zero = 0;
run;
proc print data=work.title_counts_year noobs;
    title "Publications over Time";
    var year title_count;
run;

/* Counts by Author */
proc sql;
    create table work.unique_titles as
    select distinct PUBS.lastname, PUBS.title
    from PUBS
    where PUBS.lastname in (select lastname from work.authors);
quit;
proc freq data=work.unique_titles noprint;
    tables lastname / out=work.title_counts (drop=percent);
run;
proc sort data=work.title_counts out=work.sorted_title_counts;
    by descending COUNT;
run;
proc print data=work.sorted_title_counts noobs;
    title "Publications by Author";
    var lastname COUNT;
run;

/* Journal Distribution */
/* Count the occurrences of each unique title in the journal variable */
proc sort data=PUBS out=work.unique_titles nodupkey;
    by title;
run;
proc freq data=work.unique_titles noprint;
    tables jabbrv / out=work.journal_counts;
run;
/* Sorting by Count */
proc sort data=work.journal_counts out=work.sorted_journal_counts;
    by descending COUNT;
run;
proc print data=work.sorted_journal_counts noobs;
    title "Publications by Journal";
    var jabbrv count;
run;

ods region ; /* Row 1 Column 3 */

/* Publications over Time */
proc sgplot data=work.title_counts_zero;
    band x=year lower=title_count_zero upper=title_count;
    xaxis values=(1974 to 2023 by 1) label='Year' valueattrs=(size=6)
display=(nolabel);
    yaxis label='Publications';
```

```
    title 'Publications over Time';
run;

/* Counts by Author */
proc sgplot data=work.sorted_title_counts;
    hbar lastname / response=COUNT datalabel categoryorder=respdesc;
    xaxis display=(nolabel);
    yaxis display=(nolabel);
    title "Publications by Author";
run;

/* Journal Distribution */
proc sgpie data=work.journal_counts;
    pie jabbrv /
    response=count
    datalabelloc=outside;
    title "Publications by Journal";
    run;

ods layout end ; /* End the Layout of Output Results */
ods html5 close ;
```

FLAVOR (STRENGTHS) AND AROMAS (WEAKNESSES)

Cooking with SAS® brings several key flavors (strengths) to the dish:

- ODS syntax is easy to understand and simple to use to define exactly where each data structure should appear on the dashboard.
- PROC SQL is included in base-SAS®, which offers a simple and flexible query language that can be used to wrangle the data for any table or visual on a dashboard.
- When printing to HTML, the data structures did not become distorted and were easy to update with new underlying data and code.

However, there are also some unique aromas (weaknesses) to consider:

- In this example, PROC SQL queries often included more code than R (tidyverse) equivalents.
- SAS® Studio via SAS® OnDemand for Academics (ODA) includes a way of saving data through temporary cloud libraries, which can be hard to manage with multiple datasets and when internet disconnections occur.
- Some commonly used visualizations, such as a word cloud, are not available in base-SAS® and thus require more investment in SAS® software to utilize.

COOKING IN THE R KITCHEN

SETTING UP YOUR KITCHEN

This demonstration utilized R v.4.2.3 and was accessed on RStudio® as the primary Integrated Development Environment (IDE). As with any project in RStudio®, a primary folder should be setup that will be used as the working directory. This is where the R project (Rproj) file will be housed, as well as any required subfolders. These examples use the following:

```
C:\WUSS\
```

Create subordinate folders for resources that will be required in dashboards, such as data and images:

```
C:\WUSS\data\
C:\WUSS\images\
```

Within RStudio®, only four libraries are installed and imported -`readxl,` which is used to read in excel spreadsheet files, `quarto`, which is the publishing system used to generate the dashboard, `tidyverse`, which is used for data wrangling and visualization, and `gt,` which is used for the generation of tables:

```
install.packages("readxl")
install.packages("quarto")
install.packages("tidyverse")
install.packages("gt")
library(readxl)
library(quarto)
library(tidyverse)
library(gt)
```

The Quarto® document can then be created with RStudio® and saved to the primary folder with any filename.

RECIPE OVERVIEW (STEPS)

Building a dashboard using R requires the use of dashboard-specific package, such as Quarto®. A full tutorial on Quarto® is outside the scope of this paper, but the following steps can be taken to generate a simple dashboard. Importantly, these dashboards can be reproduced as the underlying data changes by simply re-rendering within RStudio®:

1. Setup your kitchen as explained above
2. In RStudio®, create a new Quarto® Document with the minimum YAML header to ensure proper HTML setup for the dashboard:

```
---
format: html
embed-resources: true
page-layout: custom
editor: visual
colorlinks: TRUE
---
```

3. Create an R code block for setup and data wrangling. This block is used for loading the libraries, reading in the data (using readxl), and wrangling in preparation for the generation of tables and plots (using tidyverse). The output of this block should not be included in the dashboard output.
4. Create an additional code block that will serve as the layout for the dashboard. This is accomplished using the following code cell options (adjusting numbers to fit the needs of the dashboard):

```
#| layout-ncol: 3
#| layout-nrow: 3
```

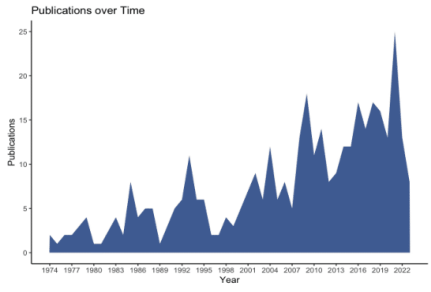
5. Within this code block, tables can be generated using gt and plots using ggplot2 (or any other table/plot package in R). As data structures are generated, they will occupy the spaces within 3x3 layout.
6. Render the document within RStudio® to write the HTML document, which will be saved to the working directory and opened automatically (typically within a web browser).

THE MEAL (EXAMPLE #2 – (3X3) DASHBOARD LAYOUT))

AREF Publications Dashboard

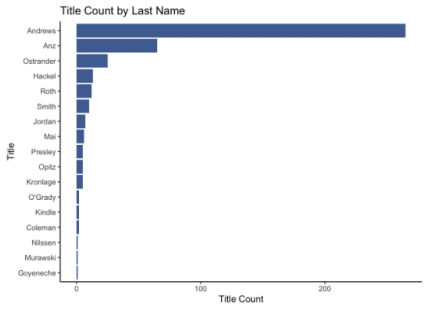
Total
360

year	publications
1974	2
1975	1
1976	2
1977	2
1979	4
1980	1
1981	1
1983	4
1984	2
1985	8
1986	4
1987	5
1988	5
1989	1
1991	5
1992	6
1993	11
1994	6
1995	6
1996	2
1997	2
1998	4
1999	3
2000	5
2001	7
2002	9
2003	6
2004	12
2005	6
2006	8
2007	5
2008	13
2009	18
2010	11
2011	14
2012	8
2013	9
2014	12
2015	12
2016	17
2017	14
2018	17
2019	16
2020	13
2021	25
2022	13



year	publications
2023	8

lastname	Publications
Andrews	265
Anz	65
Ostrander	25
Hackel	13
Roth	12
Smith	10
Jordan	7
Mai	6
Kronlage	5
Opitz	5
Presley	5
Coleman	2
Kindle	2
O'Grady	2
Goyeneche	1
Murawski	1
Nilssen	1



jabbrv	Titles
Other	163
Am J Sports Med	76
J Orthop Sports Phys Ther	32
Arthroscopy	30
Orthop J Sports Med	21
Clin Sports Med	18
Sports Health	12
J Shoulder Elbow Surg	11

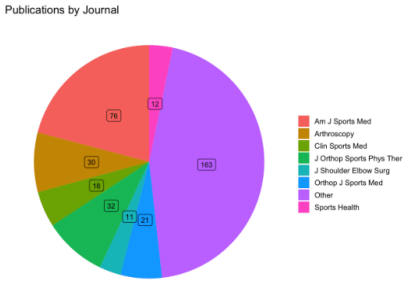


Figure 2. A Simple Dashboard Prepared with R Code as the Primary Ingredient

THE INGREDIENTS (CODE)

```
```{r}
#| label: setup
#| include: false

library(readxl)
library(quarto)
library(tidyverse)
library(gt)

PUBS <- read_xlsx("PublicationsFull.xlsx")
AUTHORS <- read_xlsx("AndrewsInvestigatorList.xlsx")

# Perform an inner join on the lastname variable
joined_data <- inner_join(PUBS, AUTHORS, by = "lastname")

# Unique Titles by Author
unique_title_counts <- joined_data %>%
  group_by(lastname) %>%
  summarise(Publications = n_distinct(title)) %>%
  arrange(desc(Publications))

# Total Unique Titles
total_title_counts <- joined_data %>%
  summarise(Total = n_distinct(title)) %>%
  arrange(desc(Total))

# Annual Publications by Author
yearly_publication_counts <- joined_data %>%
  group_by(year) %>%
  summarise(publications = n_distinct(title))

# By Journal
journal_data <- joined_data %>%
  group_by(jabbrv) %>%
  summarise(Titles = n_distinct(title)) %>%
  arrange(desc(Titles))

journal_data <- journal_data %>%
  mutate(jabbrv = ifelse(Titles < 10, "Other", as.character(jabbrv)))

journal_data <- journal_data %>%
  group_by(jabbrv) %>%
  summarise(Titles = sum(Titles)) %>%
  arrange(desc(Titles))
```

```{r}
#| label: Total
#| echo: false
#| layout-ncol: 3
#| layout-nrow: 2

# Total
```
```

```
gt(total_title_counts)

# Publications over Time
gt(yearly_publication_counts)

yearly_publication_counts <- yearly_publication_counts %>%
  mutate(year = as.numeric(year))

ggplot(yearly_publication_counts, aes(x = year, y = publications)) +
  geom_area(fill = "#4F6FA1") +
  labs(
    title = "Publications over Time",
    x = "Year",
    y = "Publications"
  ) +
  theme_minimal() +
  scale_x_continuous(breaks = seq(min(yearly_publication_counts$year),
                                  max(yearly_publication_counts$year), by =
3)) +
  scale_y_continuous(breaks = seq(0,
max(yearly_publication_counts$publications), by = 5)) +
  theme_classic()

ggplot() + theme_void() # no way to skip columns in Quarto

gt(unique_title_counts)

ggplot(unique_title_counts, aes(x = reorder(lastname, Publications), y =
Publications)) +
  geom_bar(fill="#4F6FA1", stat = "identity") +
  labs(y = "Title Count", x = "Title", title = "Title Count by Last Name") +
  theme_classic() +
  coord_flip()

ggplot() + theme_void() # no way to skip columns in Quarto

gt(journal_data)

ggplot(journal_data, aes(x = factor(1), y = Titles, fill = jabbrv)) +
  geom_col() +
  coord_polar("y", start = 0) +
  geom_label(
    aes(label = Titles, group = jabbrv),
    position = position_stack(vjust = 0.5),
    size = 3,
    show.legend=FALSE
  ) +
  theme_void() +
  theme(legend.position="right") +
  guides(fill = guide_legend(title = NULL, keywidth = 1, keyheight = 1)) +
  labs(title = "Publications by Journal")
```

FLAVOR (STRENGTHS) AND AROMAS (WEAKNESSES)

Cooking with R brings several key flavors (strengths) to the dish:

- Quarto® syntax took very minimal code to specify the layout of the dashboard.

- The tidyverse offers a simple data wrangling solution that required much less code than PROC SQL.
- Almost every visualization that can be thought of is available with the ggplot2 package of the tidyverse, or a package exists elsewhere to bring specific functionality.

However, there are also some unique aromas (weaknesses) to consider:

- Quarto® syntax is easy to understand, but it is harder to specify the exact location of data structures on the dashboard compared to SAS® ODS, which created extra white space.
- When printing to HTML, the data structures did become distorted, and many trials were needed to get the correct resolution to output.
- Some visualizations, such as pie charts, are much more complicated to generate compared to something like SGPIE within SAS®.

CONCLUSION

Dashboards are key analytic tools that are used by businesses, schools, and throughout the healthcare industry. These tools are defined by their ability to utilize frequently regenerated and often real-time data to inform decision makers. There are many programming languages that can be used to create a dashboard, akin to the use of different ingredients used during cooking. This paper explored two popular ingredients (i.e., programming languages) that are used to generate dashboards, SAS® and R. Readers were shown how a combination of Output Delivery System (ODS), PROC PRINT, PROC SGPLOT, and PROC SGPIE could be used to create dashboards using base-SAS® code. This dashboard was recreated in the open-source programming language R using packages such as quarto, readxl, tidyverse, and gt. The flavors (strengths) and aromas (weaknesses) were compared to inform readers of key factors to consider when cooking up a dashboard with these two unique ingredients.

REFERENCES

Cook, Joshua J.; Truett, J. (2023), "Making Publication Metric Tracking Easy: Using a Reproducible, Integrated System of R and Microsoft Power BI® to Ease the Pain of Assessing Publication Metrics," Proceedings of the 2023 PharmaSUG Conference.

Hadden, Louise S. (2021), "Dressing Up your SAS/GRAPH and SG Procedural Output with Templates, Attributes and Annotation," Proceedings of the 2021 SouthEast SAS Users Group (SESUG) Conference.

Harris, Kriss and Richann Watson (2020), SAS Graphics for Clinical Trials by Example, SAS Institute Inc., Cary, NC, USA.

Iannone R, Cheng J, Schloerke B, Hughes E, Lauer A, Seo J (2023). *gt: Easily Create Presentation-Ready Display Tables*. https://gt.rstudio.com, https://github.com/rstudio/gt.

Lafler, Kirk Paul; Joshua M. Horstman and Roger D. Muller (2019), "Building a Better Dashboard Using SAS® Base Software," Proceedings of the 2019 SouthEast SAS Users Group (SESUG) Conference.

Lafler, Kirk Paul (2019), PROC SQL: Beyond the Basics Using SAS, Third Edition, SAS Institute Inc., Cary, NC, USA.

Parker, Chevell (2010), "A SAS® Output Delivery System Menu for All Appetites and Applications," Proceedings of the 2010 Western Users of SAS Software (WUSS) Conference, SAS Institute Inc., Cary, NC USA.

Quarto. Version 1.2. 2022. https://quarto.org/

R: A language and environment for statistical computing. Version 4.2.3. R Foundation for Statistical Computing; 2023. https://www.R-project.org/

RStudio: Integrated Development Environment for R. Version 2023.3.0.386. Posit Software, PBC; 2023. http://www.posit.co/

Sams, Scott (2013), "SAS® BI Dashboard: Interactive, Data-Driven Dashboard Applications Made Easy," Proceedings of the 2013 SAS Global Forum (SGF) Conference, SAS Institute Inc, Cary, NC, USA.

Slaughter, Susan J. and Lora D. Delwiche (2010), "Using PROC SGPLOT for Quick High-Quality Graphs," Proceedings of the 2010 SAS Global Forum (SGF) Conference, SAS Institute Inc, Cary, NC, USA.

Welcome to the tidyverse. Journal of Open Source Software; 2019. https://www.tidyverse.org/

Wickham, H., Çetinkaya-Runde, M., Grolemund, G (2023), R for Data Science, 2nd Edition, O'Reilly Media, Inc.

ACKNOWLEDGMENTS

The author thanks WUSS for their support through the WUSS Student Scholarship Program, and Kirk Paul Lafler (@sasNerd) for his mentorship through the WUSS Mentorship Program. The author also thanks Richann Watson, Louise Hadden, Troy Martin Hughes, Jessi Truett, Ellie Hartsfield, Ashton Hayward, and Adam Anz for their support with this project, and for their general guidance and mentorship.

RECOMMENDED READING

- *PROC SQL: Beyond the Basics Using SAS®*
- *R for Data Science (2e)*
- *SAS® Graphics for Clinical Trials by Example*

CONTACT INFORMATION

Joshua J. Cook, M.S., ACRP-PM, CCRC is a recent graduate of Wake Forest University (WFU) where he earned his Master of Science for Clinical Research Management. He is a current graduate student at the University of West Florida (UWF) studying Data Science and working at the Andrews Research & Education Foundation (AREF) in Florida. Joshua has been working in the field of clinical research for nearly three years, starting in neurology clinical trials and now specializing in orthopedic regenerative medicine as a Research Quality Analyst. He has published his undergraduate honors thesis, entitled "Endurance exercise-mediated metabolic reshuffle attenuates high-caloric diet-induced non-alcoholic fatty liver disease" in the *Annals of Hepatology* and has recently submitted several orthopedic research papers to various journals as part of AREF. He has also presented his research at over ten unique conferences at the local, state, and national levels with topics spanning from the impact of blood sugar on Alzheimer's Disease to publication metric tracking with R and Microsoft Power BI®. Joshua has developed a passion for bench-to-bedside research and aims to synthesize his knowledge of the biomedical sciences, clinical research, and data science to become a physician-scientist capable of integrating clinical care with clinical research in a way that maximizes evidence-based care options for his patients.

Your comments and questions are valued and encouraged. Contact the author at:

Joshua J. Cook, M.S., ACRP-PM, CCRC

Research Quality Analyst / Graduate Student / Mentor

E-mail: jcook0312@outlook.com

LinkedIn: https://www.linkedin.com/in/joshua-j-cook-934075169/

Instagram: jcook0312