**Paper CT06**

**Using GitHub as Web-based Version-controlled Code Repository**

Spencer Childress, Rho®, Chapel Hill, NC, United States

Shane Rosanbalm, Rho, Chapel Hill, NC, United States

# ABSTRACT

Downloading code from GitHub® manually is straightforward: navigate to the repository website, download the ZIP file, and extract it to the working directory. However this manual process needs to be repeated whenever the repository changes e.g. when the developer applies bug fixes or incorporates new features. SAS® and R provide users the tools to programmatically download and source repositories housed on GitHub. This paper demonstrates how to automate code downloads from GitHub using both SAS and R, increasing efficiency by saving time and keeping code up to date.

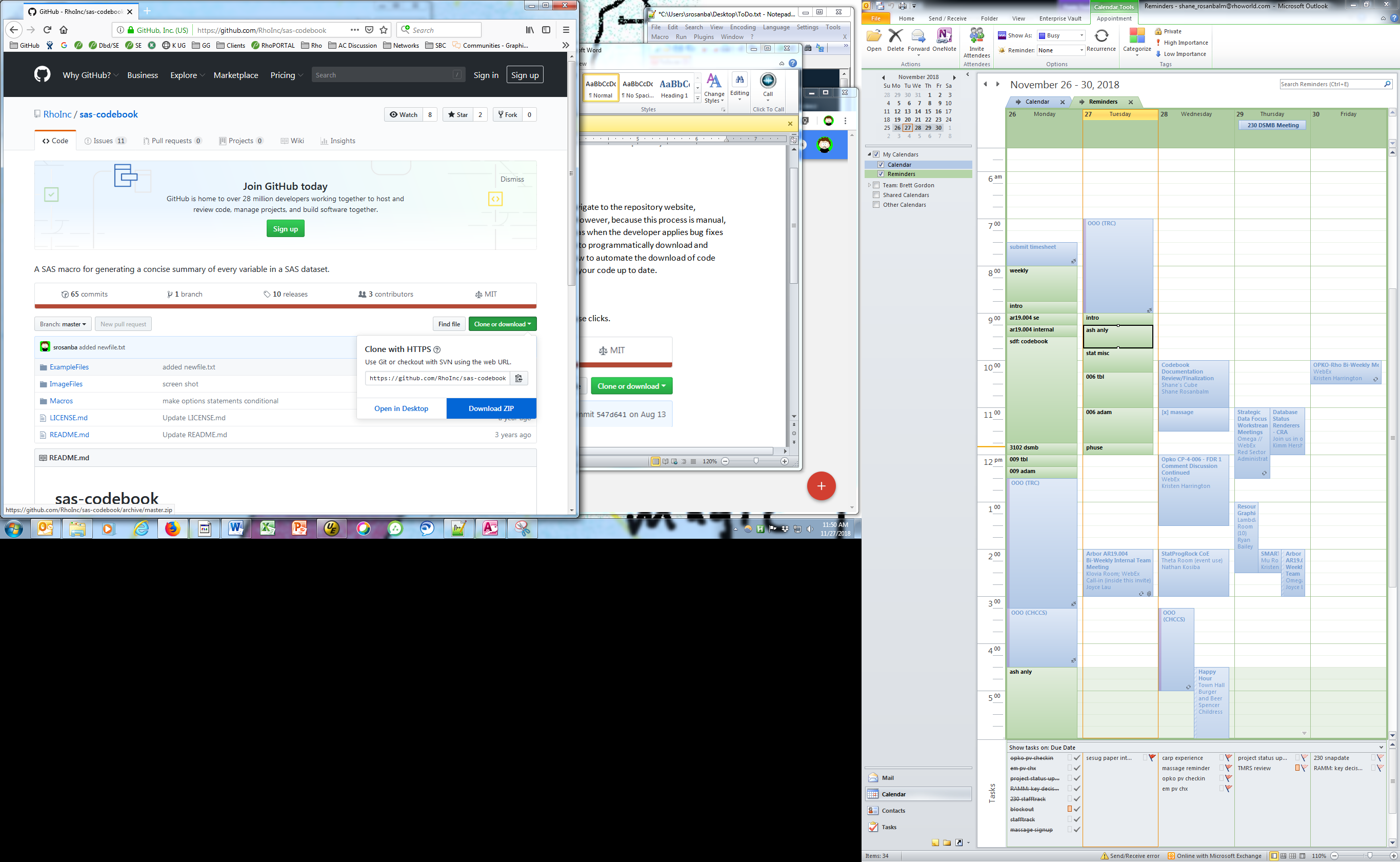
# INTRODUCTION

GitHub allows users to store and access code in the cloud, by leveraging Git, an open-source version control system, to track and control changes to the code. Although structured version control has long been valued in clinical trial documentation, version control systems are rarely used by groups other than software development. In the development and analysis of clinical trial data, however, source code changes frequently as new data come in, investigators amend the protocol, reviewers identify errors, etc. Programmers often update code or need to access an older version of a program, needs version control programs address.

This paper focuses on the automated access of code hosted on GitHub. GitHub provides an application programming interface (API) that exposes the contents of all open-source repositories to users. SAS and R programmers can read and source code to their working session straight from GitHub, without the burden of storing it locally. Additionally, users can access any version of a repository e. g. when the owner of the repository releases a new version or when a version contains bugs and users need an earlier working version.

# MANUAL CODE DOWNLOAD

Code can be manually downloaded from GitHub with a few mouse clicks. Click onthe *Clone or download* button on the repository home page and select the *Download ZIP* option.



Once the ZIP file has been saved locally, extract the contents and point to it from within the software.

In R the pointing looks like:

install.packages(lib="my/download/path/")

And in SAS like:

%include "my/download/path/fancymacro.sas";

Although manual download has the virtue of being straightforward to implement, there are disadvantages including (a) being manual and (b) not being robust to code updates.

# AUTOMATED CODE DOWNLOAD

A more robust solution is to skip the manual download and use statements within the software to access the code directly online. This approach avoids any manual steps and also ensures that users always using the latest version of the code.

In R, the *devtools* package contains a function named *install\_github*. This function allows users to automate the installation of other packages directly from GitHub.

# Install and source the devtools package.  
install.packages("devtools")   
library(devtools)   
  
# Call install\_github to download a repository directly from GitHub.  
install\_github("someuser/endswithR")   
library(endswithR)

SAS lacks built-in functionality to simultaneously download and install code from GitHub, which prompted the creation of the SAS macro %install\_github (available at [RhoInc/sas-install-github](https://github.com/RhoInc/sas-install-github)). This macro behaves like the corresponding R package. After a one-time manual download and install of the %install\_github macro itself, SAS users are henceforth able to use the macro to automagically download and install other SAS code directly from GitHub.

\*--- point to the manually downloaded install\_github macro ---;  
%include "my/utility/macros/install\_github.sas";  
  
\*--- use install\_github to install SAS code directly from GitHub ---;  
%install\_github(repo=rhoinc/violinPlot,file=src/violinPlot.sas);

# CONCLUSION

In order to access R or SAS code directly from GitHub without the hassle of a manual download, use the R package install\_github or the SAS macro %install\_github to help automate the process. GitHub allows its users to be workstation-agnostic: they can access and continue their work anywhere with an internet connection. Additionally, other users can submit issues for improvements and bug fixes and volunteer code to address those issues. While the collaboration tools GitHub offers are a great value-add to programmers, its distributed version control addresses one of the major hassles in clinical trial development: the paper trail.