

# Make GitHub Your Web-based Version-controlled Code Repository

Spencer Childress, Rho<sup>®</sup>, Chapel Hill, NC, United States

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

## ABSTRACT

Downloading code from GitHub<sup>®</sup> manually is straightforward: navigate to the repository website, download the ZIP file, and extract it to your working directory. However, because this process is manual it needs to be repeated whenever the repository changes, such as when the developer applies bug fixes or incorporates new features. SAS<sup>®</sup> and R provide users the tools to programmatically download and source repositories housed on GitHub. In this paper we show how to automate the download of code from GitHub using both SAS and R, saving you time and keeping your code up to date.

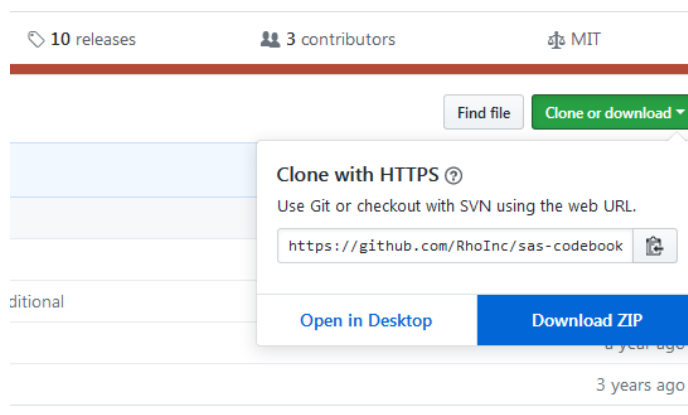
## INTRODUCTION

GitHub allows users to store and access code in the cloud. It leverages Git, an open-source version control system, to track and control changes to the code. Historically version control is a software development tool and has not found widespread use in clinical trials. In the development and analysis of clinical trial data, however, source code changes frequently as new data come in, the 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 addresses.

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 to their working session code straight from GitHub, without the burden of storing it locally. Additionally users can access any version of a repository, for example when the owner of the repository releases a newer version or when a newer version contains bugs and the user needs an older working version.

## MANUAL CODE DOWNLOAD

Code can be manually downloaded from GitHub with a few mouse clicks. Simply locate the *Clone or download* button and select the *Download ZIP* option.



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

In R the pointing looks like:

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

And in SAS like:

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

While manual download has the virtue of being straightforward to implement, it does have the disadvantages of (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 makes sure that you're always using the latest version of the code.

In R the *devtools* package contains a function named *install\_github*. This function allows you 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)
```

In SAS there is no 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 a lot 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 automatically 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

If you wish 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.