Reproducible Computing and Reporting in a Complex Software Environment

E. J. Harner[a], C. Grant[b], and M. Lilback[b]

[a] *West Virginia University,* [b] *Rc^{2}ai*

The canonical entity for reproducible computing and reporting is a Docker or Singularity computational container, which supports R Markdown. The image, which instantiates the container, should have all the components required to reproduce the report ranging from a stable release of Linux to the application software such as R and RStudio, versioned appropriately. The code underlying the report should be cloned directly from GitHub into the container. If the data is small, it may be available within the container, but it is more likely to be imported from a database, which allows versioning. R packages and other artifacts may be added to the running container as needed. A shell script can be used to push the reproducible image to Docker Hub

The report itself is often generated from an 'Rmarkdown' file, which is not only self documenting but also supportive of several output formats. The report can be rendered directly if it is simple enough, but more generally it can be automated through a 'make' file. If a more powerful computational environment is required, e.g., a Spark cluster, the appropriate drivers should be installed in the base computational image. This has the advantage of having a single compute node with Spark embedded, which might be adequate for reproducibility. These concepts are illustrated by 'rspark', an experimental edge computing environment consisting of RStudio and a Postgres containers capable of connecting to a containerized Spark cluster or more generally a cloud-based cluster. The Rstudio container is based on R running on Ubuntu 18.04 from the Rocker Project. Although RStudio is the main interface for programming, it is also possible to use an alternate editor, such as Vim.

Keywords: Reproducible report, Docker, R markdown, Postgres, Spark