Introduction to Julia for Statistics

Paul Stey, PhD

Brown Center for Biomedical Informatics

April 4, 2017

1 Outline

1. Setting up Environment
    (a) Docker Container
        i. Install Docker (if you haven't yet)
        ii. Ensure Docker daemon is running
        iii. Open terminal, run the following:
            A. `docker pull pstey/bcbi_edu:0.6`
            B. `cd ~`
            C. `mkdir statsjl`
            D. `cd statsjl`
            E. `mkdir gate`
            F. `docker run -it -v gate:/home/bcbi/gate -p 8888:8888 pstey/bcbi_edu:0.6`
            G. `sudo chown bcbi gate`
            H. `./run_jupyter.sh`
        iv. Open a web browser, and go to `localhost:8888`
    (b) AWS (if you cannot get Docker working)
        i. Go to URL shown in class
    (c) Jupyter Notebooks
2. Fundamental Objects for Statistical Computing
    (a) Vectors and N-Dimensional Arrays
        i. Initializing and populating `Arrays`

ii. Indexing, slicing, and subsetting

(b) `DataFrame` Objects

i. `NA` Type

ii. Reading and writing to file

3. Basic Descriptive and Inferential Statistics

   (a) Descriptives statistics and simple plotting

   (b) Binomial Test

   (c) Student's T-test

   (d) χ^2-test

4. Intro to Linear and Generalized Linear Models

   (a) GLM.jl Package

   i. Linear regression

   ii. Binomial logistics regression

   iii. Poisson regression

   (b) RCall.jl Package

   i. Using `$` for interactive mode

   ii. Passing objects between R and Julia

5. Intro to Statistical (Machine) Learning

   (a) Inference vs. Prediction and Classification

   (b) Overfitting

   (c) Penalized Regression

   i. Ridge regression

   ii. Lasso

   iii. Elastic Net

   (d) Ensemble Methods

   i. Bagging

   ii. Random Forests

   iii. Boosting

   (e) ScikitLearn.jl Package

6. Additional Topics (not discussed)

   (a) Bayesian Methods using Julia

   i. Mamba.jl Package

   ii. Stan.jl Package

    (b) Deep Neural Networks
        i. MXNet.jl Package

7. Resources
    (a) Books
    (b) Julia Docs (`docs.julialang.org`)
    (c) R Blog aggregator (`r-bloggers.com`)

2 Appendix

2.1 Re-starting Docker Container

To restart a Docker container previously created using `Docker run ...`, we first need to get the container ID. You can do this by running the following from your terminal.

```
docker ps --all
```

This will print a table of all Docker containers on your machine. Some of these will be temporary containers, and will look unfamiliar. The container ID you want will be at (or near) the top of the printed output. It will be the one whose `IMAGE ID` is `pstey/bcbi_edu:0.6`.

The container ID will be a string of 12 alphanumeric characters. Once you have the container ID, run the following to re-enter that container:

```
docker start -i CONTAINER-ID
```

where `CONTAINER-ID` is the alphanumeric string you observed in the output above.

This re-starts and puts you inside your earlier container, meaning that your prior settings will still hold. That is, if the paths were correctly specified at the first running of the container, your `gate/` directory on your local machine will be shared with the `/home/bcbi/gate/` directory in the Docker container, and you will still be forwarding to port 8888.

So once you've started a Jupyter server, you will once again be able to open Jupyter in your local machine's browser by going to `localhost:8888`. And as you will recall, you can start a Jupyter server by running `./run_jupyter.sh` from the `/home/bcbi/` directory in your docker container.

If you want to start a *new* container using the Docker image used in class, you would use the same commands as listed in Step 1.a.iii.F above:

```
docker run -it -v gate:/home/bcbi/gate -p 8888:8888 pstey/bcbi_edu:0.6
```

And of course, that above command assumes you created a `gate/` directory in your home folder. If you put it somewhere else, you would modify that path as necessary.

2.2 Copying Files from Docker Container

In order to copy a file to your local machine from a Docker container, you can use the command `docker cp`. In particular, you would use this command in combination with the `CONTAINER-ID` (which we can get using `docker ps --all` as discussed above) as well as the path of the desired file *inside the container*.

For example, if we wanted to copy the "plot_utils.jl" file we used for plotting, we could use the following:

```
docker cp CONTAINER-ID:/home/bcbi/utils/plot_utils.jl .
```

where CONTAINER-ID is the appropriate container ID we obtained previously using the `docker ps --all` command.

Note the period at the end of the command above. This indicates the 'plot_utils.jl' file will be copied to your current directory.