Data Analysis with Code Ocean – A How-To Guide for `Political Analysis`

Simon Heuberger

January 7, 2020

Programming languages

- Code Ocean offers the three data analysis software programs that are most often used in political science: R, Stata, and Python

Computing power

- Your material will be run on an AWS EC2 instance with at least 16 cores and 120 GB of memory

Running hours

- Upon creation, your Code Ocean account will include 10 hours of computing time per month

How to set up a Code Ocean capsule and install packages in R, Stata, and Python

- Create a new blank capsule on your dashboard (What is a capsule?; What is the dashboard?)
- Select your base environment (What is a base environment?; Selecting a base environment)
  - For R, Code Ocean offers versions 3.4.4, 3.5.3, and 3.6
  - For Stata, Code Ocean offers Stata 15 with R and Stata 15 with Matlab
  - For Python, Code Ocean offers versions 3.7.3, 3.7.0, 3.6.3, and TensorFlow 2.7
- Install needed packages on the next screen. It is not possible to install packages inside a script – they need to be installed here. Click on `+ Add`, type in the package name and version (the latter is optional), and click on the tick (Adding packages on Code Ocean)
  - For R, use `CRAN` and `GitHub`
  - For Stata, use `ssc` (Using Stata on Code Ocean)
  - For Python, use `conda` and `pip`

How to use two data analysis software programs together

- R and Stata
  - CodeOcean has a pre-provided environment (see the subsection on Stata base environments above)
- R and Python
  - With R as the basis
    - Add `python3-pip` (Python 3) or `python-pip` (for Python 2) with `apt-get` (What is `apt-get`?)

  - With Python as the basis
    * Add `r-base`, `conda-forge`, and any R packages you need (e.g. `r-ggplot2`) with `conda`
  - General rule: If your material is mainly in R, use R as the base. Likewise for Python. The Python/Conda/R combination is best for complicated setups
- LaTeX
  - Add any LaTeXpackages you need with `apt-get`

Folders in the capsule

- The key folders on Code Ocean are `/code`, `/data`, and `/results`
- `/code`
  - Contains all script files
- `/data`
  - Contains all data files
- `/results`
  - Contains all figures and tables
- Important: Every folder except `/results` is reset after each run. Everything that you want to be available for users after processing is completed thus needs to be saved to `/results`. Intermediate products passed between scripts should be saved to `/data`. Example: Say `first.R` runs simulations and saves them as `simulations.csv`, while `second.R` reads in `simulations.csv` and outputs `figure1.pdf`. `simulations.csv`, an intermediate product passed between the scripts, should be saved to and loaded from `/data`. `figure1.pdf` needs to be saved to `/results`, so users can access it after the run. As a general rule, all figures and tables used in the manuscript must be saved to `/results` (Saving files on Code Ocean).

`run`

- `run` is a shell run script that is central to the working of Code Ocean. In order to run any script file in any programming language, it needs to be listed in `run` (What is a run script?)
- `run` is not present when you create a new blank capsule. To create it, select an uploaded script file and select `Set as File to Run` with right-point-and-click. `run` will appear in the Files tab and list the selected script
- It is currently not possible to select several script files and select `Set as File to Run`. If your material includes more than one script file to run (which is very likely), create a master script file that reads in the others (then only this master file needs to be listed in `run`). Alternatively, select `Set as File to Run` for the first file and manually enter the other files into `run`, e.g. (for all three languages):

```
Rscript "first.R"
Rscript "second.R"
python -u "first.py"
python -u "second.py"
stata "first.do"
stata "second.do"
```

- If you want a full log of any run script files (i.e. including all executed commands)

enter script files in the following way into `run`:

```
Rscript -e "source('first.R', echo = T)"
python -m trace --trace "first.py"
```

For Stata, set up a log file within the `.do` file

Managing the capsule

- Creating subfolders
  - Hover the cursor over any folder. A downward arrow will appear on the right. Click it and select `New Folder` to create a new subfolder in this folder
- Uploading files/folders
  - Hover the cursor over any folder. A downward arrow will appear on the right. Click it and select `Upload File(s)` to upload files or folders to this folder. If uploading folders doesn't work, switch to Chrome as your browser (some browsers like Safari don't work with this feature) (Uploading files/folders to Code Ocean)
  - Upload your script files to `/code` and your data files to `/data`
- Readme
  - Briefly describe your material and the related manuscript
  - List the specifications of your Code Ocean environment:
    - AWS instance, number of cores, RAM
    - Number of figures and tables produced by the code
    - Running time
  - The Readme should be uploaded to `/code` as a `.txt` file
- Saving figures and tables
  - Your code needs to replicate and create saved output for all code-created figures and tables in the main text as well as in the supplementary material
  - All figures and tables need to be saved to `/results` (see section `Folders in the capsule` above)
  - If you want to save any figures or tables in subfolders within `/results`, these subfolders have to be set up in `run`. For instance, to set up the subfolder `/figures` in `/results`, enter the following into `run`:
    `mkdir -p /results/figures`
  - Figures need to be saved in `.pdf` format. Tables can be saved in the format you deem most appropriate (e.g. `.csv`, `.txt`, `.tex` etc.)
  - All figures and tables need to be saved according to their respective numbers in the manuscript, e.g. `figure1.pdf`, `table2.tex` etc.
- Reading in files
  - Files that are present when you set up the capsule need to be read in from `/data`
  - Files that are created by a script and needed in subsequent scripts (i.e. intermediate products) also need to be saved to and read in from `/data` (see section `Folders in the capsule` above)
- File paths
  - Always use relative paths, e.g. `../data` and `../results`, when reading in and saving files (File paths on Code Ocean)
  - If you have code in a subfolder, don't forget to add .. (i.e. two dots) as necessary, e.g. `load('../../data/simulations.csv')`. This example code goes up two

parent directories and then looks for `/data/simulations.csv`
  - The working directory is `/code`. If you source any script files from this folder, you thus don't need relative paths, e.g. `source("calculations.R")`
- Running files
  - When you have uploaded your material, click on `Commit Changes` and `Reproducible Run`. This will run all script files listed in `run`
  - You can close your browser once replication has started. Files will continue to be run remotely (Running files on Code Ocean)
  - If there is an error in your code, execution will be halted. Execution will also be halted if your code requires interactive user input
  - You will likely need to hit `Reproducible Run` many times until your code runs without errors
- Looking at resulting plots
  - Click on any saved plot and it will open in a new tab. If you see an empty tab, switch to Chrome as your browser (some browsers like Safari don't work with this feature)