Writing Code

Jamie Saxon

Introduction to Programming for Public Policy

September 27, 2017

Our First Script

- Our 'salaries' command was getting hard to read (for humans).
- A **script** is a saved series of instructions for the computer.
- We can write that series in many languages: bash, python, perl, ...

- The command line language is 'bash,'* and can be run via `source`.

```
■ echo "echo hello world" > hw
■ cat hw
echo hello world
■ source hw
hello world
```

- But 'echo' is not a great way to write our long command!

*Bourne Again SHell

Writing Code: Atom, Spyder, and Jupyter

- Computers 'interpret' your code, or run a 'compiled binary.'
  - **Interpreters** are computer programs that follow your instructions 'step-by-step.' So the shell is an interpreter.
  - **Compilers** translate what you write into something the computer understands 'natively' (ones and zeros = binary).
- Files must contain exactly and only the code: nothing extraneous.
  - So, needless to say (?), Microsoft Word won't cut it.

- Use **Atom** or **spyder** (ships with Anaconda) to edit scripts.
  - I will not delve into vi or emacs (command-line editors).
- For graphical work, we'll also use Jupyter notebooks.

Using Atom

1. Open Atom  , and create a file `hw.sh`. Note its location!
2. Write "`echo hello world`" in this file and save it.
3. Navigate to the directory that contains `hw.sh`:
   ```
   cd /Users/jsaxon/...
   ```
4. Finally, run it: `source hw.sh`

chmod

- A file's 'permissions' specify how users can interact with it.
  - Who can run it, read it, change it, etc?
- chmod allows you to change the file's default settings.
  - Add (+), remove (-), or set (=) rights to...
  - read (r), write (w), or execute/run (x) the file...
  - for you (u), people in a 'group' (g), or anyone (o).
- Most often, make a script executable for yourself: `u+x` or just `+x`.

Making a Script Executable

A. Go to the directory containing `hw.sh` give yourself execute permissions.
   - This makes our script into a program. (!)

B. Add "`#!/bin/bash`" to the top of `hw.sh` on a line by itself.
   - This specifies the language to use – in this case, bash.
   - We could also specify "`#!/usr/bin/env python`" which effectively means "whichever python is loaded."

C. Now run it!

```
■ chmod u+x hw.sh # A
■ ./hw.sh         # C -- don't need 'source'!
hello world
```

Two Examples on Income

Putting the Salaries Script Into a File

Full script **ex/salaries.sh** and the data file.

```
#!/bin/bash
#^^ which language?

# SORTING CHICAGO SALARIES <<< A comment
# Comments make your TAs happy and get you points.

echo "Top 10 Salaries in Chicago::"
cat salaries.csv |          # start the party!!
    grep '\$' |             # keep lines with dollar signs
    sed 's/\$//g' |         # remove the dollar signs ...
    sed "s/, //g" |         # and commas in names
    sort -t, -k 7 -n -r |   # sorting is the best.
    head -10 |              # top ten
    column -s, -t           # clean it up
```

Execute (run) `salaries.sh` from the folder that contains `salaries.csv`.

Income in the United States

What are the highest-income and highest-poverty counties in the US?

- Check out the **variables** of the American Community Survey.
- Then use **this call** from the ACS API.
- You then mainly just need to "wrangle the text into order."
  - This will get even easier with Python.
- Bonus: what about the highest and lowest in Illinois, etc...
- Check your answers with **wikipedia**.
- Then compare with a **solution**.

A First Python Script

Create a file `hello_world.py`, using vim Atom, TextEdit, etc.

Write `print("hello world")` in this file and save it.

Navigate to the directory that holds that file:
`cd /Users/jsaxon/Documents/...`

Type: `python hello_world.py`

- To go further, we need the rules and building blocks of Python...