

Lecture Homework #2 (Due Tuesday, 1/21 before lecture by 9:30am)

Notes about this assignment:

- To get full credit for this assignment, you will need to have at least 3 commits spanning 3 different days showing your progress working on this assignment.
- You are encouraged to have a separate text file as a log to explain what you are doing and any problems that you may have encountered throughout solving each problem. (Remember, you can use the `history` command as well as use a nicely formatted markdown file (`.md`) to show the commands you are entering and the results that you get. **Feel free to use markdown (.md) for files instead of .txt where appropriate.**
- You must attempt and try all of these problems for this assignment. If you get stuck, work with others and consult the course discussion forum.
- Everything that we are doing for this assignment is discussed throughout the first chapter of the textbook, so if you are getting stuck, work through the examples in the first chapter of the book.
- If you commit each day (Friday, Saturday, Sunday, Monday) up until Tuesday, you will also get some extra credit for this assignment.
- For the optional, extra credit assignments (1.10.3 and 1.10.4), you can only get extra credit if you successfully complete all of the other parts of the assignment (parts 1-5). Do not attempt these assignments until you have finished the other parts (if time permits).

Part 1:

1. Download this text file using the `wget` command into your `sandbox` directory (within `~/Developers/repos/CSB/unix/sandbox/`): `wget -c http://dev.shawntylerschwartz.com/docs/paragraphs.txt`.
2. Make a new folder in your `class-assignments` directory with the name `homework-two` and move this `paragraphs.txt` file into that location. **Then, do the git add-commit-push workflow again.**
3. Traverse to your `class-assignments/homework-two/` directory and rename the file `easy-question.txt`. **Then, do the git add-commit-push workflow again.**
4. Create a text file titled `hw2-part1-commands.txt` containing the commands you used to accomplish this task. This file should be inside the `homework-two` directory. **Then, do the git add-commit-push workflow again.**

Part 2

1. Use the `head` command to examine the **first 25 lines** of the `Pacifici2013_data.csv` file in `~/Developers/repos/CSB/unix/data/`.
2. Now create a new text file with only the **first 25 lines** of that file with the name `Pacifici-25lines.txt` in your `class-assignments/homework-two/` directory.
3. Now create another text file that contains the number of words in the **first 25 lines** of the file with the name `Pacifici-25wordcount.txt` in your `class-assignments/homework-two/` directory.
4. Create a text file titled `hw2-part2-commands-1.txt` containing the commands you used to accomplish this task. This file should be inside the `homework-two` directory. **Then, do the git add-commit-push workflow again.**

In *section 1.6.1 of the textbook*, the **pipe** `|` is introduced. The pipe command allows you to take the output from one command and subsequently use it in the next command. Given this command, find the number of words in the **first 25 lines** as you did above **using the pipe command to avoid making an intermediate text file**.

5. Create a text file titled `hw2-part2-commands-2.txt` containing the commands you used to accomplish this task. This file should be inside the `homework-two` directory. **Then, do the git add-commit-push workflow again.**

Part 3

Let's use the **pipe** again to count the number of lines in the `~/Developers/repos/CSB/unix/data/` directory (i.e., count the number of lines that are listed when you list the contents of the data directory).

1. Output the number of lines (in one line of code, using pipe) to a text file named `pipe-data-lines.txt`.
2. Using `cat` and `|` in one line of code, concatenate any 2 `.fasta` files from the `~/Developers/repos/CSB/unix/data/miRNA/` directory and count the **total number of characters** using the word count (`wc`) command. Output this to a text file named `fasta-cat.txt`. **Then, do the git add-commit-push workflow again.**

The `cut` command allows one to extract data columns that are formatted in a `.csv` format (`csv` stands for “comma separated values”, although the delimiting (separating) character between values of the file (a.k.a., *the delimiter*) does not necessarily need to be a comma).

3. Examine the `Pacifici2013_data.csv` file using the `head` command and identify the delimiter. (Write this down in a text file named `Pacifici-delim.txt`.) **Then, do the git add-commit-push workflow again.**

4. Now use the `cut` command on this `Pacifici2013_data.csv` file to extract the family of the **first 10 records** (*hint: you will need to use pipe and the `head` command and also specify the delimiter using `-d` with the `cut` command*).

Similar to the `head` command, the `tail` command allows you to look at the end of a file.

5. Look at the **last 5 records** in the `Pacifici2013_data.csv` file using the `tail` command.

The `tail` command can be used to skip the first few lines of a file with the `+` command (it will therefore start the `tail` command from whatever line you give after the plus).

6. Remove the **first line** of the `Pacifici2013_data.csv` file (*as we don't want the header of the file*).
7. Now, create a new text file with the name `Pacifici-ten.txt` that has the **first 10 records** of the `Pacifici2013_data.csv` file (*minus the header*). (Note: you will need to use both `tail` and `head` to accomplish this.)
8. Create a text file titled `hw2-part3-commands.txt` containing the commands you used to accomplish this task. This file should be inside the `homework-two` directory. **Then, do the git add-commit-push workflow again.**

Part 4

The unique (`uniq`) command allows you to see the unique values when duplicates are present.

1. Use the `wget` command to download a text file containing fish species names from a recent study Shawn conducted into your `homework-two` directory: `wget -c http://dev.shawntylerschwartz.com/docs/fish_species.txt`
Then, do the git add-commit-push workflow..
2. Explore the `sort`, `reverse sort`, and `unique` commands with this text file. Copy and paste your results into three separate text files for the results of each of these three commands:
 - `fishes-sort.txt`
 - `fishes-rev-sort.txt`
 - `fishes-unique.txt`
3. Create a text file titled `hw2-part4-commands.txt` containing the commands you used to accomplish this task. This file should be inside the `homework-two` directory. **Then, do the git add-commit-push workflow again.**

Work through exercises 1.6.2 and 1.6.3 from the textbook that will help you with some of the tasks above.

Part 5

We are going to create a data set that has the body size, order, family, genus, and scientific name that we are going to extract from the `Pacifici2013_data.csv` file. Things to keep in mind about the new data set that you are generating:

1. We don't want the header from the original `Pacifici2013_data.csv` file, so you will need to use the `tail` option to avoid extracting the first line.
2. We don't want all of the data, just columns 2-6, so use the `cut` option with the properly specified delimiter and column range values.
3. We don't want semicolons separating the data (as semicolons are a non-standard delimiter). Instead, use the `tr` command to substitute **spaces** as the *new delimiter* instead of **semicolons**.
4. To sort the file, we will want to sort on a numeric column (which is the body mass column, column 6), so we will have to use `-r` to get it in the reverse order, `-n` to sort numerically, and `-k` to specify the column.
5. Create a new file `bodym.csv` that contains all of the specifications from above.
6. Create a text file titled `hw2-part5-commands.txt` containing the commands you used to accomplish this task. This file should be inside the `homework-two` directory. **Then, do the git add-commit-push workflow again.**

EXTRA CREDIT

Complete exercises **1.10.1** and **1.10.2** from the textbook. Try to get as far as you can.

1. Post your solutions in a text file titled `hw2-ec.txt`.