# Introduction to Scientific Computing: A Crash Course

**Presented by Travis J Lawrence and Dana L Carper**
**Quantitative and Systems Biology**
**University of California, Merced**

Worksheet 1.3.2

### Exploring species occurrence data

We are going to be working with a subset of plant species occurrence data from the open access biodiversity database GBIF, specifically, collections from the year 2015. We will be doing an exploratory analysis of the data by looking at sampling efforts at different taxonomic levels and preparing a dataset that could be analyzed in popular species distribution modeling software. The file containing these data is named `Plantae.csv`.

1. The `Plantae.csv` is a `flat file` database so each species occurrence record is on a single line. Which command would you use to determine the number of records contained in the file? How many records are in the file?

2. Using `less` explore the species occurrence data. This dataset has less records than the genome annotation files, but with more fields per record making it difficult to visually inspect the data. However, it appears that there is a header line describing the fields. Does this change your answer to `question 1`?

3. Based on the file extension ( `.csv` ) this is a comma separated value file. Does this appear to be correct?

4. What delimiter is used to separate fields?

5. A common way to get a better grasp on the data is to work with only a few lines and one field at a time. Use `head` to limit the output passed to `cut` using the `|` character to look at one field at a time. This is a similar strategy in `question 9` from the genome annotation worksheet. How many fields are in a record? What information is contained in each field?

6. Based on the output from the command pipeline in `question 5` several of the fields appear to contain no data except the header. Develop a command pipeline using `cut`, `sort`, and `uniq` to determine if these seemingly empty fields are actually empty. Did you find any fields containing no data?

7. To investigate the distribution of sampling effort across phyla create a pipeline that displays the number of samples and the phylum name. You will need to use `cut`, `uniq` (don't forget about the `-c` option), and `sort`. Which phylum had the highest number of samples? Which had the least? Does there seem to be a correlation between species richness and sampling effort?

8. Wanting to fully automate your data generation for `question 7` you want to remove the header line and the line counting the samples with no phylum classification. Extend your pipeline from question `question 7` to accomplish this. You will need to use `grep` (look at the `-v` option). (Hint: remember you can specify a range of characters using `[]`. For example `[0123456789]` would match any number character.)

9. Since you might want to graph these data, likely using a bar graph, it is common to sort the bars from largest to smallest. Adding to your pipeline in `question 8` sort your data in descending order. Don't forget the `-k`, `-n`, and `-r` options of `sort`. Provide the final pipeline. How would you redirect your output to a file?

10. Wanting to get better resolution of the sampling efforts in `Tracheophyta` modify your pipeline from `question 9` to produce sampling effort at the `class` level for `Tracheophyta`. Provide your pipeline. Which `class` has the highest number of samples? What type of plants does this class contain? Is there correlation between known species richness and sampling effort?

11. Using `grep` and the append redirect `>>` create a new file that only contains the two `Classes` of flowering plants. Using `wc` does your new file contain the same number of records indicated by your results in `question 10`?

12. The file generated in `question 11` is missing the header line that describes the fields. Use `head`, `cat`, and the output redirects (`>`, `>>`) to add the heading from the original file to the file produced in `question 11`.

Some taxonomic groups are notoriously difficult to identify to species by non-experts. The next set of questions will guide you through producing a dataset from the file in `question 11` to determine if any `family` seems to have a higher number of records not identified to `species`.

13. Which field contains the taxonomic rank of a record?

14. Using `cut`, `sort`, and `uniq` what taxonomic ranks are represented in our dataset?

15. The first step to producing our dataset is to separate the records identified to `species` level or lower from those that are not. Using the information obtained in `question 14`, `grep`, and the append redirect (`>>`) create two files, 1) containing records identified to `species` level or lower and 2) records not identified to `species` level.

16. When splitting data into multiple files it's a good idea to make sure that you did not lose anything in the process. Use `wc` on the two files you generated in `question 16` and the original file from `question 11`. Did you lose any data when you split the files?

17. Which field contains the `family` classification information? Using `cut`, `sort`, and `uniq` on the file containing records that have not been identified to species to count the occurrences of each `family`. Which `family` had the most records not identified to `species`? Which had the least? This is absolute count data which can be misleading because of total number of records for each `family`. We will correct for this bias in `Python` section of the course.

A common use of species occurrence data is to build a species distribution model. In this last section we will create a dataset that you could use with the program `maxent` to build a species distribution model.

18. We have done most of the data cleaning in the previous questions. Using the file from `question 15` that contains records with `species` identification use `grep` to only select records for `Penstemon newberryi` and save your output to a new file using the redirect ( `>` ). Provide the command you used.

19. The format used by `maxent` contains three fields `species name`, `longitude`, and `latitude` separated by commas. Using `cut` on the file from `question 18` make a new file containing these three fields. Report the command used.

20. Data from `GBIF` has the `latitude` and `longitude` fields reversed from how we need them for `maxent`. Use `cut` to select the `species name` and `longitude` fields and make a file containing these. Do the same for the `latitude` field. Now use `paste` to produce a file with the fields in the correct order.

21. Finally, we need to replace `tabs` with `commas`. Use `tr` to do this. You can't easily type a `tab` character instead use `\t` to represent a `tab` character.