# Some Simple Analysis using the Shell

From A Brief Introduction to the Linux Shell for Data Science - Aris Anagnostopoulos

We will now see how we can use Linux tools to do some simple data-mining tasks. Let us start by downloading a file that we will be using. It can be found here:

http://aris.me/contents/teaching/data-mining-ds-2016/protected/beers.txt

One can download it by using a web browser. Instead we can download it with the command wget. wget allows to download multiple files, follow links and so on, so it can be very useful. Here we will just use it to download our file. Run

```
wget –user username –password pass url
```

where username and pass are the username and password that we said in class (needed because the file is password protected), and url is the address of the file written above. After you wait a bit the file will be downloaded.

After we download it we can get some information. Using head beers.txt we can see that it contains two fields separated by tab character: a name of the beer and a score. To make sure that the fields are separated by the tab character, and not spaces, we can use the option -T of cat, which prints ^I whenever it sees a tab. We don't want to do it to the entire file, so we use a pipe.

```
head beers.txt | cat -T
```

Try also

```
wc -l beers.txt
```

and see that the file has about 3 million beer ratings.

Let us assume that we want to find the 10 beers with the highest number of ratings. How can we do it? We will comnine three commands. sort orders (alphabetically or numerically) the lines of the input. uniq removes duplicate lines and produces counts if asked. cut extracts one or more fields.

Let us start with sort. If we run

```
sort beers.txt
```

we obtain a list of the beers sorted alphabetically. (Press <Ctrl>-C to stop.) We can combine the output with the command uniq. This command takes the input and removes contiguous lines that are the same, leaving only one copy of the line. In addition, if we add the option -c, we obtain also the number of times that the line was repeated. By sorting the beers, we have put all the ratings of the same beer next to each other, thus with uniq we can see how many repetitions we have.

However there is a small issue: Two ratings that have different scores will lead to different lines, so uniq will not be able to distinguish them. Thus we would like to obtain only the beer name and ignore the score.

For this we use the cut command, with which we can extract the first field:

```
cut -f 1 beers.txt
```

takes the input and extracts the first field (assumes that the fields are separated with the tab character, as in our case) and sends it to the output.

We can now combine all these commands:

```
cut -f 1 beers.txt | sort | uniq -c
```

The output is the list of beers, with the number of times that each appears on the left. We want the top-10 beers, so we can sort once again (numerically) and obtain the top-10. The full command is the following:

```
cut -f 1 beers.txt | sort | uniq -c | sort -nr | head
```

We have passed two options to the second `sort` by specifying `-nr`. The option `-n` says that we want to order the lines numerically (so that 10 appears after 9). The option `-r` (we can specify either `-n -r` or `-nr`) says that we want to do reverse sorting, so that we obtain the order from the highest to the smallest. The final output gives us the values that we wanted.