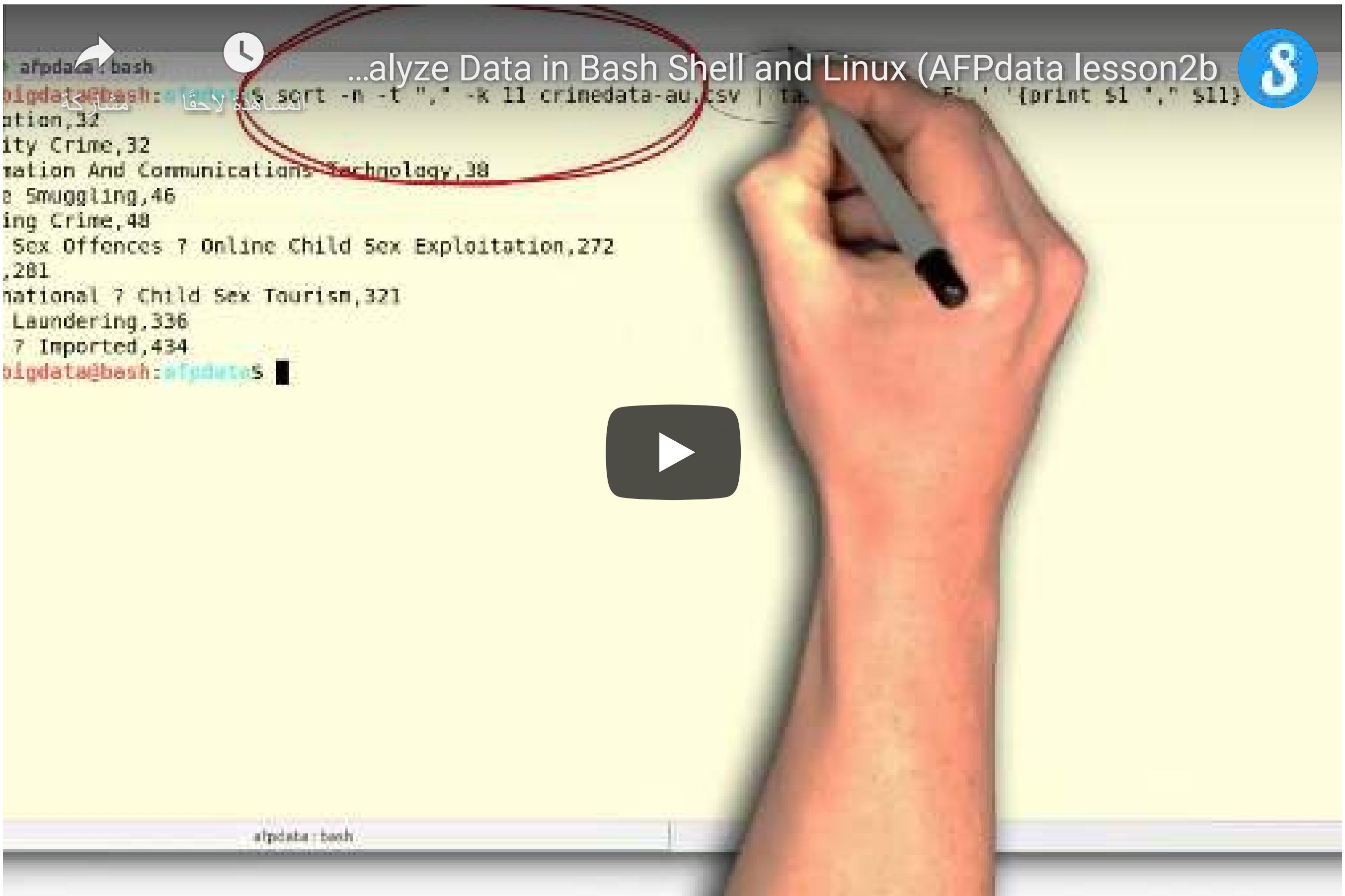
Finding the top most crime in the whole country (sort, tail, awk)

To find the top most crime that happened the highest number of times in Australia, we want to `sort` our `crimedata-au.csv` file by the `Total` column (column `11`).

As mentioned earlier, Bash has a command just for doing that, and as you may have guessed, it's called `sort`. However, running sort on the `.csv` will output all lines of the file on the screen! which is not essentially very useful.

Instead, we would like to sort the file, and then only show the first few lines of the sorted result. To do this, we'll need to use `pipes ( | )` as we did before. As a quick reminder, pipes are convenient because they allow you to move the output of a command into the input of another command, without saving the intermediate result to a file.



```
bigdata@bash:afpdata$ sort -n -t "," -k 11 crimedata-au.csv | ta        F',' '{print $1 "," $11}
ation,32
ity Crime,32
mation And Communications Technology,38
e Smuggling,46
ing Crime,48
 Sex Offences ? Online Child Sex Exploitation,272
,281
national ? Child Sex Tourism,321
 Laundering,336
 ? Imported,434
bigdata@bash:afpdata$ 
```

Video lecture: Sorting the crimes

To sort our comma-separated file by the `Total` column, which is the 11th column (see above), enter the following command:

```
sort --numeric-sort -t "," --key 11 crimedata-au.csv | tail
```

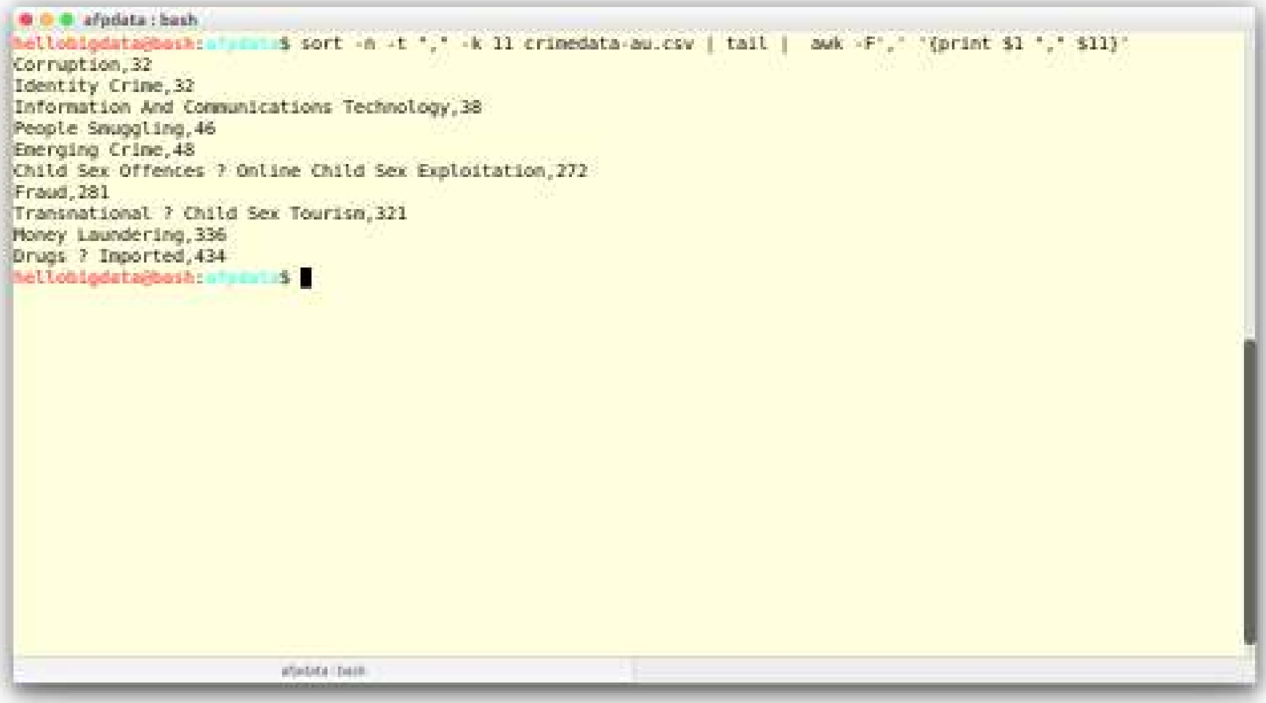
This looks scary but don’t worry, it’s simply a `sort` command, but with a few additional command-line options, so let’s break it down!

The `--numeric-sort` informs Bash to sort numerically (otherwise, it will sort alphabetically!. Next, the `-t ","` specifies that the columns of our file are defined by a comma, and -`-key 11` specifies that we want to sort only on column 11 (note: `--key 10,11` would combine the data in column `10` and `11` for the sorting).

In this case, the column of our interest is the last column so using `--key 11` is equivalent to `--key 11,11`, but had our column of interest been column 10, using `--key 10` instead of `--key 10,10` would sort the data starting at column 10 till the end of the line, and therefore yield incorrect results!

Now, using `awk` (see tutorials for details) you can only print the first column, which has the names of the crimes and the last or 11th column (total number crimes for that crime type):

```
$ sort -n -t "," -k 11 crimedata-au.csv | tail |  awk -F',' '{print $1 "," $11}'
```



```
hellobigdata@bash:afpdata$ sort -n -t "," -k 11 crimedata-au.csv | tail |  awk -F',' '{print $1 "," $11}'
Corruption,32
Identity Crime,32
Information And Communications Technology,38
People Smuggling,46
Emerging Crime,48
Child Sex Offences ? Online Child Sex Exploitation,272
Fraud,281
Transnational ? Child Sex Tourism,321
Money Laundering,336
Drugs ? Imported,434
hellobigdata@bash:afpdata$ 
```

Use the bash 'sort' function to find the total number of crimes, per crime type (Drugs Imported=434, top)

In the `awk` call `-F','` tell that the seperator is a comma. Also note that we have shorthanded the `--numeric-sort` with `-n` and `--key` with `-k`. Now, if you want just the first column's last row, which means the top most crime! The following code will capture or `tail` the last line (`tail -n 1`) and print the first column:

```
sort -n -t "," -k 11 crimedata-au.csv | \
tail -n 1 | \
awk -F',' '{print $1 ", " $11}'
```

The final output tells us that the "Drugs Import" was the top most crime (434) in Australia that year (2013?).