Linux Lab Exercices

Alexis Oger & Albert Xavier Lopez

Q1-1 Take a look at the last 10 lines of the file. Which command are you going to use?
tac jan2017articles.csv | head #tac reverse the line order
or
tail -10 jan2017articles.csv

Q1-2 Modify the command to show just the last line of the file.
tac jan2017articles.csv | head -1
or
tail -1 jan2017articles.csv

Q2-1 Extract all lines that belong to January 6th from the file and store them in a new file named "reyes.csv".
grep "06 Jan" jan2017articles.csv>reyes.csv
or
grep "^06" jan2017articles.csv>reyes.csv

Q2-2 Check that the first line of the new file has the expected values.
head -2 reyes.csv

Q3 Use the original csv to find which entries have 0 at the comment count only for those entries from january 25th.
grep "25 Jan" jan2017articles.csv | awk -F ',' '$5 == 0'
or
grep "25 Jan" jan2017articles.csv | grep '^.*,.*,.*,.*,0'

Q4 Now count the number of entries of Q3 and compare with the total number of entries.
grep "25 Jan" jan2017articles.csv | cut -d ',' -f5 | grep "0" | wc -l
wc -l jan2017articles.csv # and here we have the total number of rows

Q5-1 Now use example.bed.file. In this file, we are interested in the exon sizes of each entry. They are located in field number 11. Now you have to get the exon sizes of the first 10 entries of the file.
cut -f11 example.bed | head -10

Q5-2 How would you remove the last comma?
cut -f11 example.bed | head -10 | sed 's/,$//'
#replace the comma at the end of the line by nothing

Q6 How would get the smallest size from each of the records? The result should provide a number for each line of the input.

```
cut -f11 example.bed | sed -e 's/,$//' | awk -F ','
'{min=$1;for(i=1;i<=NF;i++)if($i<min)min=$i;print min}' | head
```

we select only exons size, then we do a loop using NF(number of exons on the actual line), and we select the minimum.

Q7 How would you now sort the records so that the first number shown is the smallest exon size? Again, the answer must provide a sorted list of numbers for each line of the output.

```
#!/bin/bash
while read -r line; do
        echo $line | sed -e "s/,/\n/g" | sort -n | tr "\n" "," | sed -e "s/,$//"
        echo
done
```

```
cut -f11 example.bed | sed -e 's/,$//' | sh seven.sh | head
```

For each line we replace ‘,’ by ‘\n’, then we can sort , and finally we replace ‘\n’ by ‘,’

Q8 Now get the 10 largest exons of chr1 stored in example.bed.

```
grep "chr1" example.bed | cut -f11 | sed 's/,*$//g' | awk -F ',' '{print NF}' | sort -nr | head
```

#print the number of exons using NF, sort this number and select only the 10 largest.

Q9 Now modify Q9 script to receive as a parameter the number of exons to search for.

```
#!/bin/bash
N=$1
grep "chr1" example.bed | cut -f11 | sed 's/,*$//g' | awk -v N="$N" -F ',' '{if(NF==N)print $0}' |
sort -nr | head
```

```
sh nine.sh 2
```

#print the lines where NF=input

Q10 Get the first 10 records of jan2017artices.csv with largest number of comments from the original csv file.

```
sed -e 's/, C/,C/' jan2017articles.csv | sed -e 's/, / /g'  | sort -t ',' -k5 -nr | head -10
```

#we replace ‘,_‘ by ‘_‘ (space) because in column number 4 there are some commas which are not field separators. And before we replace ‘,_C’ by ‘,C’ because we have to keep this coma.

Q11 Modify your previous script to receive a number as a parameter N and then show the top N entries with more comments.

```
#!/bin/bash
N=$1
sed -e 's/, C/,C/' jan2017articles.csv | sed -e 's/, / /g'  | sort -t ',' -k5 -nr | head -$N
```

```
sh eleven.sh 5
```

Q12 Now we are going to create a new articles.csv where we get a different output data layout using awk tool.

```
#!/bin/bash
(sed -e 's/, C/,C/' jan2017articles.csv | sed -e 's/, / /g' | awk -F ',' '{print
$1","$4","$5}')>temp1.txt
(rev jan2017articles.csv | cut -d ',' -f1 | rev)>temp2.txt
(paste -d, temp1.txt temp2.txt | awk -F ',' '{print $2","$3","$4","$1}')>articles.csv
sh thirteen.sh
#put in temp1.txt column 1, 4 and 5
#in order to extract column 8 we reverse the line, then we extract column 1 and we reverse
again.(We reverse the line to avoid problems created by some commas)
#Finally we join temp1.txt and temp2.txt and reorder the columns.
```

Q13 Now create a new articles2.csv format where we cut the Title text to 10 characters and we get only the last level of the Path.

```
sed -e 's/, C/,C/' jan2017articles.csv | sed -e 's/, / /g' | awk -F "," '{gsub("/.*/","",$6)}{print
substr($4,1,10)","$6}'
#We use substr to select only 10 first characters of the title.
#and gsub to delete the beginning of the path.
```