PDS: Module I - Linux Scripting

Delivery 1: Linux Lab Exercises

Alejandro Jiménez Rico

3 November 2017

It is worth noting that

- Files used: `jan2017articles.csv` and `examples.bed`.
- The field `Title` is formatted inconsistenly with the file. Whereas the whole file `jan2017articles.csv` uses a single comma (`,`) as field separator, this field contains commas and spaces together in its value (`", "`). It seems that removing this *comma and space* part of every `Title` does not muddle its information, so we decided to substitute the *comma and space* (`", "`) with a single space, using `gsub(", ", " ")`. This procedure can be found in **Q3**, **Q4**, **Q11** and **Q13**.
- There was one row when this substitution was not useful. There was one column separated not by a comma (`,`) but by a comma and space (`", "`). Knowing that this was just happenning in one row, it seemed reasonable to fix it manually. So it must be taken into account that the file is slightly edited.

Q1

Take a look at the last 10 lines of the file. Which command are you goint to use? Modify the command to show just the last line of the file.

```
cd data
head jan2017articles.csv
```

```
## Post date,Content type,Author,Title,Comment count,Path,Tags,Word count
## 31 Jan 2017,Article,Scott Nesbitt,Book review: Ours to Hack and to Own,0,/article/17/1/review-book-o
## 31 Jan 2017,Article,Jason Baker,5 new guides for working with OpenStack,2,/article/17/1/openstack-tu
## 31 Jan 2017,Article,John Mark Walker,Be the open source supply chain,1,/article/17/1/be-open-source-
## 31 Jan 2017,Article,DeLisa Alexander,Developing open leaders,1,/open-organization/17/1/developing-op
## 30 Jan 2017,Article,David Egts,How to get up and running with sweet Orange Pi,12,/article/17/1/how-t
## 30 Jan 2017,Article,Tiberius Hefflin,4 ways to improve your security online right now,3,/article/17/
## 30 Jan 2017,Article,Katie McLaughlin," WOOTConf 2017: Lockpicking, Willie Nelson developers, and mor
## 30 Jan 2017,Article,Jason Baker,"From hobbyist to professional, new analyst papers, and more OpenSta
## 28 Jan 2017,Article,Subhashish Panigrahi,How communities in India support privacy and software freed
```

```
cd data
head example.bed
```

```
## chr1 2025600 2027271 AT1G06620.1 0   +   2025617 2027094 0   3   541,322,429,    0,833,1242,
## chr5 2625558 2628110 AT5G08160.1 0   -   2625902 2627942 0   6   385,143,144,186,125,573,    2167,15
## chr5 2625558 2628110 AT5G08160.2 0   -   2625902 2627942 0   7   258,19,143,144,186,125,573, 2294,21
## chr4 12006985    12009520    AT4G22890.5 0   +   12007156    12009175    0   10  370,107,97,101,57,7
## chr4 12007040    12009206    AT4G22890.2 0   +   12007156    12009175    0   9   315,113,97,101,57,7
## chr4 12006985    12009518    AT4G22890.3 0   +   12007156    12009175    0   10  370,113,97,101,57,7
## chr4 12006985    12009520    AT4G22890.4 0   +   12007156    12009175    0   10  370,104,97,101,57,7
## chr4 12006985    12009520    AT4G22890.1 0   +   12007156    12009175    0   10  370,113,97,101,57,7
## chr2 14578539    14581727    AT2G34630.2 0   +   14578688    14581632    0   11  293,93,81,72,132,87
## chr2 14578629    14581727    AT2G34630.1 0   +   14579725    14581632    0   11  203,96,81,72,132,87
```

Q2

Extract all lines that belong to January 6th from the file and store them in a new file named *"reyes.csv"*. Check that the first line of the new file has the expected values.

```
cd data
awk '{ if ($1 == "06" && $2 == "Jan") { print $0} }' jan2017articles.csv > reyes.csv
```

Q3

Use the original csv to find which entries have 0 at the comment count only for those enteries from january 25th.

```
cd data
awk '{gsub(", ", " "); if ($5 == "0" && $1 == "25 Jan 2017") print $0}' FS="," jan2017articles.csv
```

```
## 25 Jan 2017,Article,Ben Cotton,24 Pull Requests challenge encourages fruitful contributions,0,/articl
## 25 Jan 2017,Article,Rikki Endsley,Announcing the 2016 Open Source Yearbook: Download now,0,/article/
```

Q4: Now count the number of entries of Q3 and compare with the total number of entries

```
cd data
awk '{gsub(", ", " "); if ($5 == "0" && $1 == "25 Jan 2017") print $0}' FS="," jan2017articles.csv | wc
cat jan2017articles.csv | tail -n +2 | wc -l
```

```
## 2
## 92
```

Q5

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.

```
cd data
awk '{print $11}' example.bed | head
```

```
## 541,322,429,
## 385,143,144,186,125,573,
## 258,19,143,144,186,125,573,
## 370,107,97,101,57,77,163,98,80,263,
## 315,113,97,101,57,77,163,98,101,
## 370,113,97,101,57,77,163,98,80,257,
## 370,104,97,101,57,77,163,98,80,263,
## 370,113,97,101,57,77,163,98,80,263,
## 293,93,81,72,132,87,72,86,133,189,275,
## 203,96,81,72,132,87,72,86,133,189,275,
```

Q6

How would you remove the last comma?

```
cd data
awk '{print $11}' example.bed | head | sed "s/,$//"
```

```
## 541,322,429
## 385,143,144,186,125,573
## 258,19,143,144,186,125,573
## 370,107,97,101,57,77,163,98,80,263
## 315,113,97,101,57,77,163,98,101
## 370,113,97,101,57,77,163,98,80,257
## 370,104,97,101,57,77,163,98,80,263
## 370,113,97,101,57,77,163,98,80,263
## 293,93,81,72,132,87,72,86,133,189,275
## 203,96,81,72,132,87,72,86,133,189,275
```

Q7

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

```
cd data
awk '{print $11}' example.bed | head | sed "s/,$//" | awk '{m=$1; for (i=1; i<=NF; i++) if ($i<m) m = $
```

```
## 322
## 125
## 19
## 57
## 57
## 57
## 57
## 57
## 72
## 72
```

Q8

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 input.

```
cd data
awk '{print $11}' example.bed | sed "s/,$//" | awk '{m=$1; for (i=1; i<=NF; i++) if ($i<m) m = $i; prin
paste tmpfile example.bed | sort -n 2>/dev/null | head;
rm tmpfile
```

```
## 1    chr3    3628592 3630410 AT3G11530.2 0   -   3628800 3630324 0   5   105,1,52,125,392,   1713,14
## 1    chr4    15669218    15671194    AT4G32470.2 0   -   15669704    15671095    0   5   193,158,48,
## 2    chr1    10274047    10275539    AT1G29355.1 0   +   10274047    10275539    0   3   2,697,225,
## 2    chr2    14807448    14810164    AT2G35130.1 0   -   14807588    14810164    0   8   2,185,233,2
## 2    chr5    1716870 1719541 AT5G05720.1 0   +   1716870 1719541 0   11  2,111,115,33,66,282,66,196,
## 2    chr5    2762028 2763432 AT5G08535.2 0   +   2762721 2763320 0   5   233,76,2,231,244,   0,413,5
## 2    chr5    5003313 5006817 AT5G15410.2 0   -   5003459 5005986 0   9   83,2,670,216,320,112,237,87
## 3    chr1    1086494 1096146 AT1G04160.1 0   +   1086494 1096146 0   38  3,129,144,146,160,59,160,15
## 3    chr1    1262122 1272376 AT1G04600.1 0   +   1262122 1272376 0   42  3,126,144,146,157,59,160,15
## 3    chr2    13560759    13569623    AT2G31900.1 0   -   13560759    13569623    0   40  3,129,144,1
```

Q9

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

```
cd data
awk '{print $11}' example.bed | sed "s/,$//" | awk '{m=$1; for (i=1; i<=NF; i++) if ($i>m) m = $i; prin
paste tmpfile example.bed | sort -nr 2>/dev/null | awk '{if ($2 == "chr1") print $0}' | head
```

```
## 7713 chr1    26488521    26501281    AT1G70320.1 0   -   26488744    26501281    0   15  33,96,207,7
## 5616 chr1    28816640    28822256    AT1G76780.1 0   +   28816640    28822256    0   1   5616,   0,
## 5239 chr1    7560564 7565803 AT1G21580.1 0   -   7560564 7565655 0   1   5239,   0,
## 4755 chr1    7773062 7780586 AT1G22060.1 0   -   7773372 7780586 0   9   78,201,123,165,4755,156,102
## 4154 chr1    731703  737332  AT1G03080.1 0   -   731793  737332  0   3   100,4154,1038,  5529,1224,0
## 4075 chr1    24149542    24154274    AT1G65010.1 0   +   24149542    24154024    0   3   17,196,4075
## 3897 chr1    3333594 3337491 AT1G10170.1 0   -   3333924 3337491 0   1   3897,   0,
## 3882 chr1    20879465    20895393    AT1G55860.1 0   -   20879899    20895393    0   19  100,68,612,
## 3875 chr1    4788558 4794654 AT1G13980.1 0   +   4789586 4794397 0   3   96,864,3875,    0,902,2221,
## 3757 chr1    28075073    28078830    AT1G74720.1 0   +   28075172    28078418    0   1   3757,   0,
```

Q10

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

Note that `.Rmd` notebook files do not accept arguments as inputs in its scripts. So we just paste the code without computing it.

```
cd data
N=$1
awk '{print $11}' example.bed | sed "s/,$//" | awk '{m=$1; for (i=1; i<=NF; i++) if ($i<m) m = $i; prin
paste tmpfile example.bed | sort -nr 2>/dev/null | awk '{if ($2 == "chr1") print $1}' | head -n$N
```

Q11

Get the first 10 records of jan2017articles.csv with largest number of comments from the original csv file.

```
cd data
awk 'gsub(", ", " ");{print $5}' FS="," jan2017articles.csv> tmpfile;
paste tmpfile jan2017articles.csv | sort -nr 2>/dev/null | head
```

```
## 174  10 Jan 2017,Article,Amanda McPherson,Open medical records community supports new system in Mozar
## 31 Jan 2017,Article,Jason Baker,5 new guides for working with OpenStack,2,/article/17/1/openstack-tu
## 30 Jan 2017,Article,Katie McLaughlin," WOOTConf 2017: Lockpicking Willie Nelson developers and more"
## 30 Jan 2017,Article,Jason Baker,"From hobbyist to professional new analyst papers and more OpenStack
## 30 Jan 2017,Article,David Egts,How to get up and running with sweet Orange Pi,12,/article/17/1/how-to
## 28 Jan 2017,Article,Robin Muilwijk,"New Minecraft launcher comes to Linux Tilt Brush Toolkit and more
## 27 Jan 2017,Article,Jen Wike Huger,"Top 5: Solid state drives in Linux Brotli compression algorithm a
## 27 Jan 2017,Article,Alan Smithee,Data Privacy Day 2017: Solutions for everyday privacy,5,/article/17/
## 26 Jan 2017,Article,Joshua Pearce,Search this database for inactive patents that are now in the publi
## 26 Jan 2017,Article,Jeremy Garcia,How to join a technical community,1,/article/17/1/how-join-technica
```

Q12

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

Note that `.Rmd` notebook files do not accept arguments as inputs in its scripts. So we just paste the code without computing it.

```
cd data
N=$1
awk 'gsub(", ", " ");{print $5}' FS="," jan2017articles.csv> tmpfile;
paste tmpfile jan2017articles.csv | sort -nr 2>/dev/null | head -n $N
```

Q13

Now we are going to create a new articles.csv where we get a different output data layout using awk tool INPUT: Post date,Content type,Author,Title,Comm count,Path,Tags,Word count OUTPUT: Title;Comment count;Word count;Post date.

```
cd data
awk '{gsub(", ", " "); print $4}' FS="," jan2017articles.csv > tmpfile1;
awk '{gsub(", ", " "); print $5}' FS="," jan2017articles.csv > tmpfile2;
awk '{gsub(", ", " "); print $8}' FS="," jan2017articles.csv > tmpfile3;
awk '{gsub(", ", " "); print $1}' FS="," jan2017articles.csv > tmpfile4;
paste -d ";" tmpfile1 tmpfile2 tmpfile3 tmpfile4 > articles.csv
```

Q14

Now create a new article2.csv format where we cut the Title text to 10 characters and we get only the last level of the Path.

```
cd data
awk '{$1 = substr($1, 1, 10); print $0 }' FS=";" OFS=";" articles.csv | head
```

```
## Title;Comment count;Word count;Post date
## Book revie;0;660;31 Jan 2017
## 5 new guid;2;419;31 Jan 2017
## Be the ope;1;1668;31 Jan 2017
## Developing;1;768;31 Jan 2017
## How to get;12;933;30 Jan 2017
## 4 ways to ;3;1242;30 Jan 2017
## " WOOTConf;1;844;30 Jan 2017
## "From hobb;0;327;30 Jan 2017
## How commun;0;453;28 Jan 2017
```