Mission Possible: Data Engineering with UNIX Tools

1. Download the file:

curl https://www.spinellis.gr/cgi-bin/oasa-history?id=p2822001 -o oasa.gz

gunzip oasa.gz

The stream's fields are: data acquisition time stamp, line number, bus number, position reporting time stamp, bus position latitude, bus position longitude.

Useful commands:

grep, fgrep, wc, sort, cut, tee, xargs, uniq, awk, sed, join, comm, diff, seq, head, tail, rev, tac, gzip, tr, tsort

For each question provide the answer and the command(s) you used to obtain it. For multi-stage pipelines, add comments on a line before each command or after the command name to explain what your commands do. (On the UNIX shell comments start with the # character.)

All commands given must produce exactly the answer you provide. (It is not allowed to interpret the data manually or with other tools.)

2. What is the data's field separator character?

,

3. What happens when you try to load the data into a spreadsheet?

load data to .csv file: **awk '{print $1,$2,$3,$4,$5,$6}' <oasa> oasa.csv**

Message "File not loaded completely" appears, as the size of the file is large and it opens only the first 1048576 rows

4. How many records are provided with the data?

cat oasa | wc -l

wc -l oasa

#6961147

5. What is the data acquisition time stamp of the last record in the stream you fetched?

tail -n 1 oasa | awk -F "," '{print $1}'

Tail -n 1 command is used to take the last line of the file. With awk -F, is specified that the file is delimited by comma (,) and with print $1 we take the first field of the last line which is the data acquisition time stamp

6. How many different buses appear in the data?

awk -F, '{print $3}' oasa | sort | uniq | wc -l

With awk -F, is specified that the file is delimited by comma (,) and with print $3 we take the third field which indicates the bus's number, then we use sort command to group the different buses so as to omit duplicates with uniq command, finally we count the lines to find the number of different buses.

7. How many different routes are covered by the data?

awk -F, '{print $2}' oasa | sort | uniq | wc -l

With awk -F, is specified that the file is delimited by comma (,) and with print $2 we take the second field which indicates the line(route) number, then we use sort command to group the different routes so as to omit duplicates with uniq command, finally we count the lines to find the number of different routes.

awk -F, 'NR>2 {count[$2]++} END {for (i in count) print i, count[i]}' oasa_sub

8. How many dates are covered by the data?

awk -F, '{print $1}' oasa | awk -FT '{print $1}' | sort | uniq | wc -l

OR

awk -F, '{print $4}' oasa | awk -F " " '{print $1,$2,$3}' | sort | uniq | wc -l

With awk -F, is specified that the file is delimited by comma (,) and with print $4 we take the fourth field position reporting time stamp, we use this as input for the second awk where only the first three fields, which are delimited by tab, are used, as we want to count only date without time stamp, e.g Jan 3 2021, then we use sort command to group the different dates so as to omit duplicates with uniq command, finally we count the lines to find the number of different dates.

9. Which route is associated with the most position reports?

awk -F, '{print $2}' oasa | sort | uniq -c | sort -nr -k1 | head -n 1 | awk '{print $2}'

With awk -F, is specified that the file is delimited by comma (,) and with print $2 we take the second field line number (route), then we use sort command to group the different routes and uniq -c is used to show the number of occurrences of each route which is the same as the number of position reports, subsequently sort -nr -k1 command is used to take in descending order the first field, which is the number of occurrences, and finally we take the first line with head and print the second field, which is the route number.

10. How many position reports appear in the data more than once?

awk -F, '{print $4}' oasa | sort | uniq -d | wc -l

With awk -F, is specified that the file is delimited by comma (,) and with print $4 we take the fourth field position reporting time stamp, then we use sort command to group the different position reports and uniq -d (duplicate) command is used to show the reports that are shown more than once. Finally we count the lines to find the number of different position reports.

11. Which is the most frequent first two-digit sequence in numbers assigned to buses?

awk -F, '{print $3}' oasa |cut -c1,2 | sort | uniq -c | sort -nr | head -n 1

With awk -F, is specified that the file is delimited by comma (,) and with print $3 we take the third field, which indicates bus number, with cut -c1,2 we keep only the first two-digits from the bus number, then we use sort command to group the different bus numbers and uniq -c(count) is used to show the the number of occurrences of each, subsequently sort -nr command is used to take them in descending order and finally we take the first line with head and print the result with awk.

12. How many buses did not travel on February 6th 2020?

grep '^2021-02-06' oasa | awk -F, '{print $3}' | sort | uniq > buses0602.txt

Grep command is used to find all the lines that start with 2021-02-06, then awk is used to print the third field of these lines, subsequently we use sort command to group the different bus numbers and uniq is used so as to omit duplicates, we keep the result in buses0602.txt file.

awk -F, '{print $3}' oasa | sort | uniq > buses.txt

We collect all bus numbers in file buses.txt, by using sort command to group them and uniq command to omit duplicates.

comm -13 buses0602.txt buses.txt | wc –l

Comm command is used to compare two sorted files line by line and find the buses that are unique on file buses.txt, which indicates that they did not travel on February 6th.

OR

diff buses0602.txt buses.txt |egrep '^[<>]' | wc -l

13. On which date were the most buses on the road?

awk -F, '{print $1, $3}' oasa | awk -FT '{print $1, $2}' | awk '{print $1,$3}' | sort | uniq -c | sort -nr -k1 | head -n 1 | awk '{print $2}'

With awk -F, is specified that the file is delimited by comma (,) and with print $1 and $3 we take the fisrt (data acquisition time stamp) and third field (bus number), we use them as input for the second awk where the data timestamp is separated by "T" and we only keep the first (e.g. 2021-01-24) and the third part(e.g. 60495) for our third awk, then we sort them by date and we use uniq -c(count) to show the number of occurrences, subsequently we group them in descending order by first column (number of occurrences) and we print the second field of the first line.

14. Which route has been served by the highest number of different buses?

awk -F , '{print $2,$3}' oasa | sort -u | awk '{print $1}' | uniq -c | sort -nr -k1 | head -n 1

With awk -F, is specified that the file is delimited by comma (,) and with print $2 and $3 we take the second (line number) and third field (bus number), then we use the sort -u command to group them and take only the unique ones, moreover we take only the first field, which indicates the line number and by the use of uniq -c(count) we show the number of occurrences of each line number, finally we sort them in descending order and take first line.

15. On which hour of the day (e.g. 09) are there overall the most buses on the road?

awk -F, '{print $1,$3}' oasa | awk -FT '{print $1,$2}' | awk '{print $3,$2}' | awk -F: '{print $1}' | awk '{print $2,$1}' | sort -u | awk '{print $1}' | uniq -c | sort -nr -k1 | head -n 1

With awk -F, is specified that the file is delimited by comma (,) and with print $1 and $3 we take the first (data acquisition time stamp) and third field (bus number), we use them as input for the second awk where the data timestamp is separated by "T" and we only keep the second (e.g. 88054) and third part (e.g. 10:34:47) reverted in the third awk. Moreover from the time stamp we keep only the first field, which indicates the hours. Subsequently, in fifth awk we have as input the bus number per hour and group them by hour, then in the final awk (= hour)

we count the occurrences by the use of uniq -c (count), we sort them in descending order and show the first line.

16. On which hour of the day (e.g. 23) are there overall the fewest buses on the road?

awk -F, '{print $1,$3}' oasa | awk -FT '{print $1,$2}' | awk '{print $3,$2}' | awk -F: '{print $1}' | awk '{print $2,$1}' | sort -u | awk '{print $1}' | uniq -c | sort -n -k1 | head -n 1

With awk -F, is specified that the file is delimited by comma (,) and with print $1 and $3 we take the first (data acquisition time stamp) and third field (bus number), we use them as input for the second awk where the data timestamp is separated by "T" and we only keep the second (e.g. 88054) and third part (e.g. 10:34:47) reverted in the third awk, Moreover from the time stamp we keep only the first field, which indicates the hours. Subsequently, in fifth awk we have as input the bus number per hour and group them by hour. Then in the final awk (= hour) we count the occurrences by the use of uniq -c (count), we sort them in ascending order and show the first line.

17. For which weekday (e.g. Wednesday) does your data set contain the most records?

Consider using the (GNU) awk strftime and mktime functions to convert between date formats. (5 bonus points — you can get a 10/10 without answering this question.)

awk -F, '{print $1}' oasa |awk -FT '{print $1}' | sort | uniq -c | sort -k1rn | head -n 1 |awk '{print $1, $2}' > d

cat d

date -d $(echo $d|awk -F- '{print $3 "-" $1 "-" $2}') +%A

OR

awk -F, '{print $1}' oasa | awk -FT '{print $1}' | sort | uniq -c | sort -nr | head -n 1 | awk '{print strftime("%A",$2),$1}'

OR

awk -F, '{print $4}' oasa | awk -F" " '{print $1,$2,$3}' | sort | uniq -c | sort -nr | head -n 1 | awk '{print strftime("%A",$2),$1}'

With awk -F, is specified that the file is delimited by comma (,) and with print $4 we take the fourth field (position reporting time stamp). Then we keep only the first three fields delimited by tab (e.g. Jan 5 2021), which indicate the date. After that we use sort command to group the different dates and uniq -c(count) is used to show the number of occurrences of each date.

Subsequently, sort -nr command is used to take them in descending order. Finally, the first line is shown with head command and with strftime function we convert the format of the day from 05 to Thursday and print the result including the number of its occurrences.

18. What are the bounding box geographic coordinates of the area served by the buses?

awk -F, '{print $5}' oasa | sort -nr | head -n 1

NORTH >> with awk -F, is specified that the file is delimited by comma (,) and with print $5 we take the fifth field, then we sort them in descending order and show the first line.

awk -F, '{print $5}' oasa | sort -n | head -n 1

SOUTH >> as mentioned above we take the fifth field, then we sort them in ascending order and show the first line.

awk -F, '{print $6}' oasa | sort -n | head -n 1

WEST >> with print $6 we take the sixth field, then we sort them in ascending order and show the first line.

awk -F, '{print $6}' oasa | sort -nr | head -n 1

EAST >> similarly we take the sixth field, then we sort them in descending order and show the first line.

19. Which bus has appeared closest to your favorite location?

north:37.998622

east:23.799922

awk -F, '{print $3,$5,$6,sqrt((($5-37.998622)^2+($6-23.799922)^2))}' oasa | sort | uniq | sort -k4 | head -n 1

result: 10212 37.9985670 23.8000190 0.000111508

In order to find the coordinates of the bus closest to my favorite location, Euclidean distance formula was implemented by the use of sqrt function. With awk -F, is specified that the file is delimited by comma (,) and with print $3, $5, $6 we take the bus line, the latitude and longitude. After calculating the distance between my favorite restaurant and the bus's location coordinates, we group them in ascending order based on Euclidean distance formula result, so as to show the smallest- closest distance.

OR

awk -F, '{print $3,$5,$6}' oasa | shuf -n 1

bus number : 10212

With awk -F, is specified that the file is delimited by comma (,) and with print $3, $5, $6 we take the third (bus number), fifth (latitude) and sixth field (longitude).Then shuf command is used to take a random values from the input.

20. How many position reports have been sent by the chosen bus?

awk -F, '{if($3 == "10212") print $4}' oasa | wc -l

With awk -F, is specified that the file is delimited by comma (,) and with print $4, we take the fourth field position reporting time stamp, only if the number of bus is "10212" and then we count them.

21. What was the chosen bus's last position in the obtained data stream?

awk -F, '{if($3 == "10212") print $1","$5","$6}' oasa | sort -t, -nr -k1 | head -n 1 | awk -F, '{print $2, $3}'

With awk -F, is specified that the file is delimited by comma (,) and with print $1, $5, $6 we take the first (data acquisition time stamp), the fifth (latitude) and the sixth (longitude) field only if the number of bus is "10212", then we group them in descending order based on first field, by specifying the separator to "," and finally we show the second and third fields of first line.

22. On which date has the chosen bus given the most position reports?

awk -F, '{if($3 == "10212") print $1}' oasa | awk -FT '{print $1}' | sort | uniq -c | sort -nr -k1 | awk '{print $2}' | head -n 1

With awk -F, is specified that the file is delimited by comma (,) and with print $1 we take the first field (data acquisition time stamp) only if the number of bus is "10212". We use it as input for the second awk where the data timestamp is separated by "T" and we only keep the first part date (e.g. 2021-01-24, then we group the different dates and by the use of uniq –c (count) we show the occurrences of each date, subsequently we sort them in descending order based on the occurrences and we print the second field of the first line.

23. On how many routes has the chosen bus traveled?

awk -F, '{if($3 == "10212") print $2}' oasa | sort | uniq | wc -l

with awk -F, is specified that the file is delimited by comma (,) and with print $2 we take the second field (line number) only if the number of bus is "10212". Then we group the

different routes by the use of sort command and we use uniq command so as to omit duplicates. Finally, we count the lines.

24. How many buses have shared at least one route with the chosen bus?

awk -F, '{if($3 == "10212") print $2}' oasa | sort | uniq > routes_10212.txt

with awk -F, is specified that the file is delimited by comma (,) and with print $2 we take the second field (line number) only if the number of bus is "10212", then we group the different routes by the use of sort command and we use uniq command so as to omit duplicates, finally we store them to a file named routes_10212.txt.

while read route; do awk -F, '{if('$route'== $2 && $3!="10212") print $2,$3}' oasa | sort | uniq >> routes_all.txt;done < routes_10212.txt.txt

For each route of bus 10212 check if another bus does the same route and store it to a file, the file has this format: route bus.

awk '{print $2}' routes_10212.txt | sort | uniq | wc -l

OR

awk -F, '$3=="10212" {print $2,$3}' oasa_p2822008 | sort | uniq > uniq_linenumber

awk -F, '{print $2,$3}' oasa_p2822008 | sort |uniq > line_bus_no

join -1 1 -2 1 -o 1.1,1.2,2.1,2.2 line_bus_no uniq_linenumber | awk '{print $2}' | sort | uniq | wc -l