UNIX

**3)**

error, This data set is too large for the Excel

**4)**

*7967221*

#curl https://www.spinellis.gr/cgi-bin/oasa-history?id=p2822110 -o oasa.gz

#gunzip oasa.gz

wc -l < oasa

**5)**

*2022-03-02T14:55:29*

tail -n 1 oasa

**6)**

*1583*

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

# awk -F for printing the 3 column without problem with seperates

# | combine commands lines

# sort and then using uniq ( "uniq" need a sorted first)

# wc -l count the previous output

**7)**

*466*

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

**8)**

*61*

cut -d "T" -f 1 oasa | sort -u |wc -l

# split the string in "T" , then count the uniq

**9)**

*3629*

awk '!seen[$5,$6]++' oasa | awk -F ',' '{print $2}'| sort | uniq -c | sort | tail -n 1

# first removed the duplicate from 5&6 columns then sorting for the uniq-c ( by the number of occurrences) and then again sorting and output the last line with the max

**10)**

*443347*

awk -F ',’ '{print $5 " " $6}' oasa | sort | uniq -d | wc -l

awk -F ',' '{print $5,$6}' oasa | sort | uniq -d | wc -l

#print the 5,6th column then sort the output , count only the duplicate lines

**11)**

*30 , 246*

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

# take the 3rd column sort uniq and

# split the first 2 digits of the output (3rd column) , sort and take the unit by the number of occurrences , sorting again and print the max

grep -v '2022-02-06' oasa | awk -F, '{print $3}' | sort | uniq |wc -l

**12)**

928

grep 'Feb 6 2022' oasa | awk -F, '{print $3}' | sort | uniq > bus\_6feb.txt

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

comm -3 totalbus.txt bus\_6feb.txt | wc -l

# take the busses wich appears in 6 feb 2022 and save them into bus\_6feb.txt

# take the total uniq busses and save them totalbus.txt

# comm -3 which print lined in file1 which not in file2 ( totalbus.txt , bus\_6feb.txt)

# wc -l count the previous output

**13)**

Feb 3 2022

awk -F, '{print $3","substr($4,1,11)}' oasa | sort -u -d | awk -F, '{print $2}' | sort -d | uniq -c | sort | tail -n 1

# Feb 3 2022

# print 3 , 4th columns take only the date without time

# substr(s, a, b) : it returns b number of chars from string s, starting at position a. The #parameter b, in this case , is 11 (there 2 spaces before the '9' ), starting position is 1 in the #4 string, then sort dictionary the second column of the output and unique by the number #of occurrences , sorting again and print the last column ( max)

14)

222 3566

awk -F, '{print $2,$3}' oasa | sort | uniq | awk '{print$1}' | sort | uniq -c | sort | tail -n 1

# print the 2&3rd columns , sort , take the first column (2nd) sort sort and take the unit by #the number of occurrences , sorting again and print the max

15

11

1582

awk -F ',' '{print $3 "," substr($4,13,2)substr($4,25,2)}' oasa | sort -d | uniq | awk -F , '{print $2}' | sort | uniq -c | sort | tail -n 1

# print 3 , and from the 4th column take only the hour of day and tha am or pm as well

# substr(s, a, b) : it returns b number of chars from string s, starting at position a. The #the #sort dictionary the pair of the output take the uniq and print the second column of the #output sort it and take the uniq by by the number of occurrences , sorting again and print #the last

16

03

701

awk -F ',' '{print $3 "," substr($4,13,2)substr($4,25,2)}' oasa | sort -d | uniq | awk -F , '{print $2}' | sort | uniq -c | sort | head -1

# print 3 , and from the 4th column take only the hour of day and tha am or pm as well

# substr(s, a, b) : it returns b number of chars from string s, starting at position a. The #the #sort dictionary the pair of the output take the uniq and print the second column of the #output sort it and take the uniq by by the number of occurrences , sorting again and print #the first

18

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

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

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

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

19

awk -F, '{print $3,$5,$6, sqrt ( ($5-37.97891248796111)^2 + ($6-23.74463506873661)^2) }' oasa | sort -u -k 4n | head -1

#print bus number and calculate the Euclidean distance then sort according the 4th column #of the output

20

**3841**

awk -F, '{print $3,$5,$6}' oasa | awk '{if($1=="60941") print $2,$3}' |sort | uniq | wc -l

# print the 3,5,6 columns if the first column of the ouptut is 60941 print the lat lon -sort #them and remove duplicate and count

21

awk -F, '{print $3,$4,$5,$6}' oasa | awk '{if($1=="60941") print $5,$6,$7}' | sort -k1 | tail -n 1

# print $3,$4,$5,$6 columns so created an output with 7 column the type of "56084 Jan 7 #2022 02:09:33:000PM 37.9508210 23.7265940"

# so if the first column ( the bus) is 60941 print the 5 6 7 columns of the output and sort them according the time column 5th and take the last

22

awk -F, '{print $3,$4,$5,$6}' oasa | awk '{if($1=="60941") print $2,$3,$4}' | sort | uniq -c | sort | tail -n 1

# print $3,$4,$5,$6 columns so created an output with 7 column the type of "56084 Jan 7 #2022 02:09:33:000PM 37.9508210 23.7265940"

# so if the first column ( the bus) is 60941 print the 2 3 4 columns of the output (the #columns with month date year) sort them , take the uniq by the number of occurrences , #sorting againand take the last

**23**

awk -F, '{print $2,$3}' oasa | awk '{if($2=="60941") print $1}' | sort | uniq | wc -l

# print the 2nd and 3rd column like this type ( 2644 30851) if the second column is 60941 #print the first column the route sorted remove the duplicates and count

24

awk -F, '{print $2,$3}' oasa | awk '{if($2=="60941") print $1}' | sort | uniq > my\_buss\_routes.txt

awk -F, '{print $2,$3}' oasa | awk '{if($2!=60941) print $1,$2}' | sort -u > routes.txt

join -1 1 -2 1 my\_buss\_routes.txt routes.txt > final.txt

awk -F, '{print $1,$2}' final.txt | awk '{print $2}' | sort | uniq | wc -l

#first create a file with the routes which belong in the chosen bus and another file with all the other routes and buses expect the chosen bus

# join the files and sort uniq by 2nd column ( buses) and count