

# Sorting and Counting - Exercises 1

1. Find top 10 files by size in your home directory including the subdirectories. Sort them by size and print the result including the size and the name of the file (hint: use find with -size and -exec ls -s parameters)
2. Create a dummy file with this command : `seq 15 > 20lines.txt; seq 9 1 20 >> 20lines.txt; echo "20\n20" >> 20lines.txt`; (check the content of file first)
  - a) Sort the lines of file based on alphanumeric characters
  - b) Sort the lines of file based on numeric values and eliminate the duplicates
  - c) Print all duplicated lines of the file
  - d) Print the line which has most repetitions
  - e) Print all lines with the number of repetitions sorted by the number of repetitions from lowest to highest
3. Create another file with this command : `seq 0 2 40 > 20lines2.txt`
  - a) Create 3<sup>rd</sup> file from the first two but without duplicates
  - b) Merge the first two files. Print unique lines together with the number of occurrences inside the merged file and sorted based on line content.
4. Go to `~/Data/opentraveldata`. Get the line with the highest number of engines using sort.

# Sorting and Counting - Exercises 1

1) `find ~ -type f -size +10M -exec ls -sh {} \; | sort -nr | head`

2a) `sort -d 20lines.txt`

2b) `sort -nu 20lines.txt`

2c) `sort -n 20lines.txt | uniq -d`

2d) `sort -n 20lines.txt | uniq -d -c | sort -nr | head -1`

2e) `sort -n 20lines.txt | uniq -c | sort -n`

3a) `sort -nu 20lines.txt 20lines2.txt > 20files_no_dupl.txt`

3b) `sort 20lines2.txt 20lines.txt | uniq -c | sort -k 2n,2`

4) `sort -t "^" -k 7nr,7 optd_aircraft.csv | head -1`

# Processing and filtering - Exercises 2

Go to ~/Data/opentraveldata

1. Change the delimiter of `optd_aircraft.csv` to “,”
2. Check if `optd_por_public.csv` has repeated white spaces
3. How many columns has `optd_por_public.csv`? (hint: use `head` and `tr`)
4. Print column names of `optd_por_public.csv` together with their column number. (hint: use `paste`)
5. Use `optd_airlines.csv` to obtain the airline with the most flights?
6. Use `optd_airlines.csv` to obtain number of airlines in each alliance?

# Processing and filtering - Exercises 2

1) `cat optd_aircraft.csv | tr "^" "," | optd_aircraft_comma.csv`

2) `cat optd_por_public.csv | tr -s "[:blank:]" | wc`  
`wc optd_por_public.csv`  
Compare the size in bytes!

3) `head -n 1 optd_por_public.csv | tr "^" "\n" | wc -l`

4) `paste <(seq 46) <(head -1 optd_por_public.csv | tr "^" "\n")`

5) `cat optd_airlines.csv | cut -d "^" -f 8,14 | sort -t "^" -k 2nr,2 | head -1`

6) `cat optd_airlines.csv | cut -d "^" -f 10 | sort | uniq -c | sort -rn | head`

# Processing and filtering - Exercises 3

Go to ~/Data/opentraveldata

1. Use grep to extract all 7x7 (where x can be any number) airplane models from optd\_aircraft.csv.
2. Use grep to extract all 3xx (where x can be any number) airplane models from optd\_aircraft.csv.
3. Use grep to obtain the number of airlines with prefix “aero” (case insensitive) in their name from optd\_airlines.csv
4. How many optd\_por\_public.csv columns have “name” as part of their name? What are their numerical positions? (hint: use seq and paste)
5. Find all files with txt extension inside home directory (including all sub directories) that have **word** “Science” (case insensitive) inside the content. Print file path and the line containing the (S/s)cience word.

# Processing and filtering - Exercises 3

- 1) `cut -d "^" -f 3 optd_aircraft.csv | grep -E "7[0-9]7"`
- 2) `cut -d "^" -f 3 optd_aircraft.csv | grep -E "3[0-9]{2}"`
- 3) `cat optd_airlines.csv | cut -d "^" -f 8 | grep -i -E "^Aero" | wc -l`
- 4) `paste <(seq 50) <(head -n 1 optd_por_public.csv | tr "^" "\n") | grep name`
- 5) `find ~ -type f -iname "*.txt" -exec grep -iwh "Science" {} \;`

# Processing and filtering - Exercises 4

Use Text\_example.txt

1. Replace every “line” with new line character (“\n”)
2. Delete lines that contain the “line” word.
3. Print ONLY the lines that DON'T contain the “line” word

# Processing and filtering - Exercises 4

1) `sed 's/line/\n/g' Text_example.txt`

2) `sed '/line/d' Text_example.txt`

3) `sed -n '/line/!p' Text_example.txt`



# Working with compressed Files – Exercises 5

1. Go to `~/Data/us_dot/otp`. Show the content of one of the files.
2. Use `head/tail` together with `zcat` command. Any difference in time execution?
3. Compress “`optd_por_public.csv`” with `bzip2` and then extract from the compressed file all the lines starting with `MAD` (hint: use `bzcat` and `grep`)
4. (`On_Time_On_Time_Performance_2015_1.zip`): What are the column numbers of columns having “carrier” in the name ? (don't count!) (hint: we have seen this 😊)
5. (`On_Time_On_Time_Performance_2015_1.zip`) Print to screen, one field per line, the header and first line of the `T100` file, side by side.

# Working with compressed Files – Exercises 5

1. `zless On_Time_On_Time_Performance_2015_1.zip`
2. `zcat On_Time_On_Time_Performance_2015_1.zip | head`  
`zcat On_Time_On_Time_Performance_2015_1.zip | tail`
3. `bzip2 optd_por_public.csv`  
`bzcat optd_por_public.csv.bz2 | grep -E "^MAD"`  
or  
`bzgrep -E "^MAD" optd_por_public.csv.bz2`
4. `paste <(seq 110) <(zcat ./On_Time_On_Time_Performance_2015_1.zip | head -n 1 | tr "," "\n") | grep -i "carrier"`
5. `paste <(seq 110) <(zcat ./On_Time_On_Time_Performance_2015_1.zip | head -n 1 | tr "," "\n") <(zcat ./On_Time_On_Time_Performance_2015_1.zip | head -n 2 | tail -1 | tr "," "\n")`

# Shell Script – Exercises 6

1. Create a script that will return column names together with their column number from the csv files. The first argument should be file name and the second delimiter.
2. Create a script that accepts a CSV filename as input (\$1 inside your script) and returns the model of the aircraft with the highest number of engines. (use it on ~/Data/opentraveldata/optd\_aircraft.csv)
3. Repeat script 2, but add a second argument to accept number of a column with the number of engines. If several planes have the highest number of engines, then the script will only show one of them. (use it on ~/Data/opentraveldata/optd\_aircraft.csv)
4. Create a script that accepts as input arguments the name of the CSV file, and a number (number of engines) and returns number of aircrafts that have that number of engines. (use it on ~/Data/opentraveldata/optd\_aircraft.csv)

# Shell Script Exercises

1) File: column\_name\_number.sh

```
#!/usr/bin/bash
```

```
FILE_INPUT=$1
```

```
DELIMITER=$2
```

```
#echo "My name is ${0}"
```

```
#echo "Delimiter= ${DELIMITER}"
```

```
#echo "file=${FILE_INPUT}"
```

```
NUM_COLUMNS=$(cat ${FILE_INPUT} | head -1 | tr ${DELIMITER} "\n" | wc -l)
```

```
#echo "Column Number=${NUM_COLUMNS}"
```

```
paste <(seq ${NUM_COLUMNS}) <(head -1 ${FILE_INPUT} | tr ${DELIMITER} "\n")
```

# Shell Script Exercises

2) File: model\_with\_most\_engines.sh

```
#!/usr/bin/bash
```

```
FILE_INPUT=$1
```

```
MODEL=$(sort -t "^" -k 7nr ${FILE_INPUT}|head -1 | cut -d "^" -f 3)
```

```
echo "The model is ${MODEL}"
```

# Shell Script Exercises

3) File: model\_with\_most\_engines2.sh

```
#!/usr/bin/bash
```

```
FILE_INPUT=$1
```

```
COLUMN_INPUT=$2
```

```
MODEL=$(sort -t "^" -k ${COLUMN_INPUT}nr ${FILE_INPUT}|head -1 | cut -d  
"^" -f 3)
```

```
echo "The model is ${MODEL}"
```

# Shell Script Exercises

4) File: num\_of\_engines2.sh

```
#!/usr/bin/bash
```

```
FILE_INPUT=$1
```

```
NUM_ENGINES=$2
```

```
cut -d "^" -f 7 ${FILE_INPUT} | grep "${NUM_ENGINES}" | uniq -c | tr -s  
" " | cut -d " " -f 2
```

# CSVkit – Exercises 7

1. Use `csvstat` to find out how many different manufactures are in the file
2. Extract the column `manufacturer` and using pipes, use `sort`, `uniq` and `wc` find out how many manufacturers are in the file. Why does this number differ to the number reported in `csvstat`?
3. What are the top 5 manufacturers?
4. Using `csvgrep`, get only the records with manufacturer equal to *Airbus* and save them to a file with pipe (`|`) delimiter.



# CSVkit – Exercises 7

1) `csvstat -d "^" -c manufacturer optd_aircraft.csv`

2) `csvcut -d '^' -c manufacturer optd_aircraft.csv | tail -n+2 | sort | uniq | wc -l`

3) `tail -n+2 optd_aircraft.csv | cut -d '^' -f 2 | sort | uniq -c | sort -nr | head -5`

or

`csvcut -d '^' -c manufacturer optd_aircraft.csv | csvsort | tail -n+2 | uniq -c | sort -nr | head -5`

4) `csvgrep -d '^' -c manufacturer -m Airbus optd_aircraft.csv | tr ", " "|" > airbus.csv`

or

`csvgrep -d '^' -c manufacturer -m Airbus optd_aircraft.csv | csvformat -D '|' > airbus.csv`

# SQL - Exercises 3

1. Import optd\_aircraft.csv y optd\_airlines.csv a Postgres ( /Data/opentraveldata/).
2. Que modelo de avion tiene mayor numero de motores? (optd\_aircraft)
3. Que numero de motores es más común en los aviones? (optd\_aircraft)

# SQL - Quick exercises 3

1. **1a)Execute in psql:**

create database optd;

**1b) Execute @shell inside ( ~/Data/opentraveldata/ ) :**

csvsql -d '^' optd\_aircraft.csv -i postgresql > psql\_create\_optd\_aircraft.txt

csvsql -d '^' optd\_airlines.csv -i postgresql > psql\_create\_optd\_airlines.txt

psql -d optd -f psql\_create\_optd\_aircraft.txt

psql -d optd -f psql\_create\_optd\_airlines.txt

**1c) Execute in psql:**

\cd ~/Data/opentraveldata/

\copy optd\_aircraft from './optd\_aircraft.csv' delimiter '^' csv header;

\copy optd\_airlines from './optd\_airlines.csv' delimiter '^' csv header;

2. select manufacturer, model, nb\_engines from optd\_aircraft where nb\_engines is NOT NULL order by nb\_engines DESC limit 1;

3. select nb\_engines, count(\*) from optd\_aircraft where nb\_engines is NOT NULL group by nb\_engines order by count(\*) DESC limit 1;