Master Data Science

Para cambiar useradd por mi usuario por defecto.

useradd@LAPTOP-RISUJV9C:~ 1 $ sudo -u $USERNAME rscheze -s /bin/bash

16/11/19 Igor

Todo comienza con el Raw data. El 80% del tiempo se dedica en convertir esto en structured data

Sobre esto después de hará el Data Science y el análisis con los insides

Dentro de Unix, tenemos lo comandos, el terminal donde los escribimos, el Shell que los interpreta y el kernel que es el corazón(el sistema operativo)

Unix es de pago. Crearon in sistema igual pero gratis(GNU). El problema es que no tenían sistema operativo(kernel). En 1991 un señor inventa Linux que es un kernel. Linux kernel. Ahora hay un GNU con kernel Linux y en 2015 crearon un kernel de GNU. Conviven ambos.

**Comandos en Terminal**

**Clone-** es muy útil si no tienes conexión porque te permite trabajar en local y después subir la nueva versión.

**Init**- para crearlo

**Pull-** Antes de empezar a trabajar si el trabajo se comparte hay que hacer un pull para trabajar sobre la última versión.

**Add**- miestras trabajo añado los ficheros que después incluiré en commit.

**Commit-** mejor no hacer commit de varios ficheros ni demasiado grandes. Siempre poner un mensaje para que se pueda identificar lo que está incluido dentro. Esto es como guardar lo que vas haciendo en local.

**Push**- Para subirlo a remoto antes de irnos.

**Echo**- Para escribir. Es un comando y lo demás serán variables. Irá después un espacio y lo que quiero escribir. Si después viene $ es que lo que viene después es una variable

useradd@LAPTOP-RISUJV9C:~ master\* ± **echo $SHELL**  al darle a intro me debe devolver lo siguiente.

/bin/bash

Si pongo **bash** se me abrirá otra Shell

Para ver la **Shell que se está ejecutando** pongo

useradd@LAPTOP-RISUJV9C:~ master\* 6s 127 ± **echo $0**

bash

para **cerrar la Shell** pongo exit

useradd@LAPTOP-RISUJV9C:~ master\* ± **exit**

exit

**Shell es** donde está el programa que se ejecuta cuando pongo un comando en el terminal. Hay varios Shell. Si estás en remoto no sabremos donde está la Shell.

Vamos a ver qué **tipo de máquina** tenemos. Se hace con cat y si ponemos después una barra invertida es porque va un parámetro absoluto.

useradd@LAPTOP-RISUJV9C:~ master\* ± **cat /etc/os-release**

etc es un directorio. Entre cat y la barra invertida hay un espacio

en mi caso me devuelve:

NAME="Ubuntu"

VERSION="18.04.1 LTS (Bionic Beaver)"

ID=ubuntu

ID\_LIKE=debian

PRETTY\_NAME="Ubuntu 18.04.1 LTS"

VERSION\_ID="18.04"

HOME\_URL="https://www.ubuntu.com/"

SUPPORT\_URL="https://help.ubuntu.com/"

BUG\_REPORT\_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY\_POLICY\_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION\_CODENAME=bionic

UBUNTU\_CODENAME=bionic

Para saber la **info de la CPU** donde estoy conectado. Me devuelve la memoria del ordenador al que estoy conectado

useradd@LAPTOP-RISUJV9C:~ master\* ± **cat /proc/cpuinfo**

**La variables son todo mayúsculas.**

useradd@LAPTOP-RISUJV9C:~ master\* ± MASTER=Kchool

useradd@LAPTOP-RISUJV9C:~ master\* ± echo $MASTER $SHELL

Kchool /bin/bash

Si hubiera puesto esto así echo “$MASTER $SHELL” sería un solo parámetro en lugar de dos.

Con comillas simples no se hubiera interpretado nada.

Si pongo echo y después date, me escribirá date. Si pongo solo date, me devolverá la fecha de hoy.

Si me sale algo raro porque no he cerrado las comillas . Vuelvo a la visión normal con **Control C**

useradd@LAPTOP-RISUJV9C:~ master\* ± echo $MASTER'

>

useradd@LAPTOP-RISUJV9C:~ master\* 130 ± (al devolverlo me indica el error que se produjo. En este caso 130

Para poner el texto en dos líneas se usa **– e y \n**

useradd@LAPTOP-RISUJV9C:~ master\* ± echo -e "test \n second line"

Para tabular se poner -e y después \t

useradd@LAPTOP-RISUJV9C:~ master\* ± **echo -e " prueba \t prueba \t\t prueba"**

prueba prueba prueba

Si ponemos **barra invertida y punto y coma**, lo pinta pero no lo interpreta.

useradd@LAPTOP-RISUJV9C:~ master\* ± echo " This is ok \; echo of echo"

This is ok \; echo of echo

**Vamos a bajar un fichero**

useradd@LAPTOP-RISUJV9C:~ master\* 128 ± **git clone https://github.com/IgorAramb/DS\_CL\_files.git**

Cloning into 'DS\_CL\_files'...

remote: Enumerating objects: 10, done.

remote: Counting objects: 100% (10/10), done.

remote: Compressing objects: 100% (9/9), done.

remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0

Unpacking objects: 100% (10/10), done.

**Importante:** EL repositorio donde haga el clone es donde me guardará el fichero

useradd@LAPTOP-RISUJV9C:~/git master\* ± git clone https://github.com/IgorAramb/DS\_CL\_files.git

**Como nos movemos por los distintos directorios**

Como siempre lo primero es un comando y el resto serían parámetros.

ls para ver todos.

Ls -l me da mucha más información. La segunda ele significa long.

Ls -s lo ordena por tamaño

useradd@LAPTOP-RISUJV9C:~/.config master\* ± **ls -t** (para ordenar por fecha)

Si ponemos **cd -** nos llevará al directorio donde estábamos justo antes.

Si ponemos **cd..** para subir un nivel

Si ponemos **cd ./** significa que, desde donde estoy yo ahora mismo. Es importante si tienes el mismo fichero en varios directorios y quieres que se ejecute exactamente el que tienes en ese directorio.

Si ponemos cd .

useradd@LAPTOP-RISUJV9C:~ master\* ± **cd .** al darle al autocomplite te salen las distintas opciones.

./ ../ .conda/ .config/ .git/ .local/

Cuando el nombre empieza por punto significa que es oculto. Todos los que tienen barra invertida al final son directorios.

**Para concatenar ficheros .** En este caso pongo varias veces el mismo. Me pone uno debajo de otro

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/shell master ± **cat -n** Text\_example.txt Text\_example.txt Text\_example.txt

**Para ver todo el contenido** cat\*

Para ver todo lo que empieza por T entonces cat T\* o que termina con txt entonces cat \*.txt si me da igual como termina después del txt entonces cat \*.txt\* Importante txt no es una extensión. En Linux no existen extensiones como tal.

**REVISAR BIEN. No llegaba al directorio que quería. Shell**

**useradd@LAPTOP-RISUJV9C:~ master\* ± cd DC\_CL\_files**

-bash: cd: DC\_CL\_files: No such file or directory

**useradd@LAPTOP-RISUJV9C:~ master\* 1 ± ls -lrt**

total 1553640

-rw-rw-rw- 1 useradd useradd 530308481 Oct 15 18:20 Anaconda3-2019.10-Linux-x86\_64.sh.2

-rw-rw-rw- 1 useradd useradd 530308481 Oct 15 18:20 Anaconda3-2019.10-Linux-x86\_64.sh.1

-rw-rw-rw- 1 useradd useradd 530308481 Oct 15 18:20 Anaconda3-2019.10-Linux-x86\_64.sh

drwxrwxrwx 1 useradd useradd 4096 Nov 14 22:03 anaconda3

drwxrwxrwx 1 useradd useradd 4096 Nov 16 00:09 master\_kschool\_tst

-rw-rw-rw- 1 useradd useradd 20 Nov 16 00:10 README.md

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:27 DS\_CL\_files

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:44 git

**useradd@LAPTOP-RISUJV9C:~ master\* ± cd git/**

**useradd@LAPTOP-RISUJV9C:~/git master\* ± ls -lrt**

total 0

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:44 DS\_CL\_files

**useradd@LAPTOP-RISUJV9C:~/git master\* ± cd DS\_CL\_files/**

**useradd@LAPTOP-RISUJV9C:~/git/DS\_CL\_files master ± ls -lrt**

total 0

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:44 opentraveldata

-rw-rw-rw- 1 useradd useradd 14 Nov 16 10:44 README.md

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:44 shell

**useradd@LAPTOP-RISUJV9C:~/git/DS\_CL\_files master ± cd shell/**

**useradd@LAPTOP-RISUJV9C:~/git/DS\_CL\_files/shell master ± ls**

Finn.txt Text\_example.txt Text\_example.txt.copy

**useradd@LAPTOP-RISUJV9C:~/git/DS\_CL\_files/shell master ± ls -la**

total 616

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:44 .

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:44 ..

-rw-rw-rw- 1 useradd useradd 627563 Nov 16 10:44 Finn.txt

-rw-rw-rw- 1 useradd useradd 237 Nov 16 10:44 Text\_example.txt

-rw-rw-rw- 1 useradd useradd 237 Nov 16 10:44 Text\_example.txt.copy

**useradd@LAPTOP-RISUJV9C:~/git/DS\_CL\_files/shell master ±**

**Como se interpreta la información anterior. Cada uno es un fichero**.

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:44 ..

-rw-rw-rw- 1 useradd useradd 627563 Nov 16 10:44 Finn.txt

drwxrwxrwx :Directorio- Si hay un guión delante es que no tenemos permisos

1: Numero es el número de enlaces que tenemos en ese directorio. Los subdirectorios que hay por debajo.

useradd useradd :usuario y dentro de que equipo de trabajo está

**Como cambiar los permisos de los directorios**. Drwxrwxrwx

Así le quitaría al user(u) el permiso de Writting(w)

useradd@LAPTOP-RISUJV9C:~/git/DS\_CL\_files/shell master ± **chmod u-w README.md**

chmod significa change mode

Puedo dividirlo en tres. Permisos para mi departamento, para mi grupo y para otros.

useradd@LAPTOP-RISUJV9C:~/git/DS\_CL\_files master ± **chmod 777 README.md**

Primer 7 permiso a departamento

Segundo 7 permiso a mi grupo

Tercer 7 permiso a otros

Es binario 1 o 0 Si doy permiso es 7, 5 para dos grupos y 4 para un grupo solo con permisos.

**Me creo 3 directorios que me ordenará por orden alfabético**

useradd@LAPTOP-RISUJV9C:~/one master\* ± mkdir this and that

useradd@LAPTOP-RISUJV9C:~/one master\* ± ls

and that this

**Me crea un directorio y debajo un subdirectorio y así sucesivamente (two,three…)**

useradd@LAPTOP-RISUJV9C:~/one master\* ± mkdir -p two/three/four

**Para crear un fichero vacio**

useradd@LAPTOP-RISUJV9C:~ master\* ± **touch** example\_unique.txt

**Para copiar el archive en otro directorio**

Copy file *ahora iría el nombre del repositorio donde lo quiero copiar.*

**Para copiar varios ficheros en un directorio**(this el directorio que hemos creado justo antes. Después va barra invertida)

useradd@LAPTOP-RISUJV9C:~/one master\* ± cp file file.copy this/

**Lo que hay en two te lo copia en three(ambos son directorios)**

useradd@LAPTOP-RISUJV9C:~/one master\* 1 ± cp -r two three

**Puedo meterlo del uno ya creado o de uno nuevo**

useradd@LAPTOP-RISUJV9C:~/one master\* ± cp -r two two.copy

**Para ver lo que hay debajo de un directorio ls y el nombre del directorio.**

useradd@LAPTOP-RISUJV9C:~/one master\* ± ls two

three

**Para mover y cambiar los nombres de los ficheros**

useradd@LAPTOP-RISUJV9C:~/one master\* ± mv that ../

**para cambiarlo el nombre**

useradd@LAPTOP-RISUJV9C:~/one master\* ± mv that that\_new

**o ambas cosas a la vez**

useradd@LAPTOP-RISUJV9C:~/one master\* 1 ± mv ./file ./an/new\_file

Para borrar se usar m (remove)

Para borrar todo el historial de Git (me borra el fichero .git)

useradd@LAPTOP-RISUJV9C:~ master\* ± rm -rf .git

**Resumen de creación de directorio y archivo con los accesos(Ejecrcicio de la presentación)**

useradd@LAPTOP-RISUJV9C:~ $ cd

useradd@LAPTOP-RISUJV9C:~ $ mkdir first\_dir

useradd@LAPTOP-RISUJV9C:~ $ cd first\_dir

useradd@LAPTOP-RISUJV9C:~/first\_dir $ touch text\_file.txt

useradd@LAPTOP-RISUJV9C:~/first\_dir $ chmod g+x,o+w text\_file.txt

* (*u stands for user.*
* *g stands for group.*
* *o stands for others.*
* *a stands for all.)*

useradd@LAPTOP-RISUJV9C:~/first\_dir $ ls -l

total 0

-rw-rwxrw- 1 useradd useradd 0 Nov 16 12:18 text\_file.txt

useradd@LAPTOP-RISUJV9C:~/first\_dir $ mkdir sub1 sub2 text\_file

useradd@LAPTOP-RISUJV9C:~/first\_dir $ cp text\_file.txt sub1/

useradd@LAPTOP-RISUJV9C:~/first\_dir $ cp text\_file.txt sub1/-file.txt.2

pedir resto ejercicio.

**Revisar less y man para entrar en un fichero.**

**Alias te devuelve los 25 últimos comandos.**

**Para salir de un terminal. Control + d (solo funciona si estoy en una línea en blanco)**

**Control + l para limpiar la pantalla.**

**Con control + u corta toda la línea para pegarla en otro sitio. (control + y)**

**Control + k borra todo desde donde estés.**

**Si pongo history me da todos los comandos utilizados y con las flechas puedo navegar por ellos.**

**Con control + r me busca la última vez que haya utilizado algo en líneas previas**

**Para abrir un archivo, primero me meto en el directorio y después solicito las cabeceras que quiera ver(en este caso 5) Empieza por menos porque es un parámetro y los parámetros empiezan todos por menos.**

useradd@LAPTOP-RISUJV9C:~/**DS\_CL\_files/opentraveldata** master ± **head -5 ./optd\_aircraft.csv**

**otras formas de hacerlo**

**useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master ± head -n +5 ./optd\_aircraft.csv**

**Todo menos las útimas 5 lineas**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master **± head -n -5 ./optd\_aircraft.csv**

**Las últimas 10 lineas del fichero**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master ± **tail ./optd\_aircraft.csv**

**lo mismo que**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master ± **tail -n -10 ./optd\_aircraft.csv**

**Como contamos palabras**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master 1 ± **wc optd\_aircraft.csv**

458 1068 18612 optd\_aircraft.csv

1. Lineas 458
2. Palabras 1068
3. Bytes 18612

Podría haber pedido solo uno. wc – l optd… o wc -w o wc -c

**Contando las 10 primeras líneas nos daría**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master ± **head optd\_aircraft.csv |wc**

10 29 460

**Conteo de las 5 últimas filasde las 10 primeras**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master ± **head optd\_aircraft.csv | tail -5 | wc**

5 21 222

**Para que me de el número de la fila**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master ± **cat -n optd\_aircraft.csv | head | tail -5**

454 YK2^Yakovlev / Jakovlev^Yak 42^YK2^3J^YK42^3^J

455 YK4^Yakovlev / Jakovlev^Yak 40^YK4^3J^YK40^3^J

456 YN2^HAMC / Harbin^Y12^YN2^2T^Y12^2^T

457 YN7^Xian^Yunshuji Y7^YN7^2T^AN24^2^T

458 YS1^NAMC^YS-11^YS1^2T^YS11^2^T

**Para que me lo cuente**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master ± **cat -n optd\_aircraft.csv | head | tail -5 | wc**

5 26 257

**Si quiero que me lo guarde**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* 1 ± **cat -n optd\_aircraft.csv | head | tail -5 > 5lines.csv**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± ll

total 8020

drwxrwxrwx 1 useradd useradd 4096 Nov 16 13:18 ./

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:27 ../

-rw-rw-rw- 1 useradd useradd 0 Nov 16 13:18 5

-rw-rw-rw- 1 useradd useradd 257 Nov 16 13:18 5lines.csv

-rw-rw-rw- 1 useradd useradd 18612 Nov 16 10:27 optd\_aircraft.csv

-rw-rw-rw- 1 useradd useradd 107306 Nov 16 10:27 optd\_airlines.csv

-rw-rw-rw- 1 useradd useradd 8080005 Nov 16 10:27 optd\_por\_public.csv

**Si lo ejecuto de nuevo lo reescribo. Para que lo añada a lo anterior tengo que poner dos >>**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± **cat -n optd\_aircraft.csv | head | tail -5 >> 5lines.csv**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± ll

total 8020

drwxrwxrwx 1 useradd useradd 4096 Nov 16 13:18 ./

drwxrwxrwx 1 useradd useradd 4096 Nov 16 10:27 ../

-rw-rw-rw- 1 useradd useradd 0 Nov 16 13:18 5

-rw-rw-rw- 1 useradd useradd 514 Nov 16 13:20 5lines.csv

-rw-rw-rw- 1 useradd useradd 18612 Nov 16 10:27 optd\_aircraft.csv

-rw-rw-rw- 1 useradd useradd 107306 Nov 16 10:27 optd\_airlines.csv

-rw-rw-rw- 1 useradd useradd 8080005 Nov 16 10:27 optd\_por\_public.csv

Si lo hago con un texto que quiero guardar sería igual

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± **echo $SHELL > my\_shell.txt**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± **echo '$SHELL'>> my\_shell.txt**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± **echo 'date' >> my\_shell.txt**

causeradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± **cat my\_shell.txt**

**La forma más chula pedir el conteo de un fichero**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* ± **wc my\_shell.txt**

**para encontrar todos los ficheros o los directorios**

useradd@LAPTOP-RISUJV9C:~/DS\_CL\_files/opentraveldata master\* **± find .**

**La primera vez que lo hago es importante poner find | wc para ver lo grande que es lo que busco.**

**Si busco por nombre**  find . – name

useradd@LAPTOP-RISUJV9C:~ 16s $ **find -name "text\_file"**

**Para que ignore mayúsculas o minúsculas pongo la i delante de name**

useradd@LAPTOP-RISUJV9C:~ 15s $ **find -iname "tExt\_file"**

**Si lo quiero limitar a directorios termino con type d si quiero fichero type f etc**

useradd@LAPTOP-RISUJV9C:~ 15s $ **find -iname "text\_file" -type d**

./first\_dir/text\_file

**Si quiero limitar esa búsqueda que no busque por debajo del nivel x de subdirectorios (en ejemplo 6) se pone maxdepth**

useradd@LAPTOP-RISUJV9C:~ 3s $ **find . -maxdepth 6 -iname "tExt\_file\*" -type f**

./anaconda3/lib/python3.7/distutils/\_\_pycache\_\_/text\_file.cpython-37.pyc

./anaconda3/lib/python3.7/distutils/text\_file.py

./first\_dir/sub1/text\_file.txt

./first\_dir/sub3/text\_file.txt

./first\_dir/text\_file.txt

**Podríamos buscar ficheros que no tengan un determinado permiso**

useradd@LAPTOP-RISUJV9C:~ $ **find . -iname "tExt\_file\*" -type f ! -perm 777**

./anaconda3/lib/python3.7/distutils/\_\_pycache\_\_/text\_file.cpython-37.pyc

./anaconda3/lib/python3.7/distutils/text\_file.py

./anaconda3/lib/python3.7/site-packages/jedi/third\_party/typeshed/stdlib/2and3/distutils/text\_file.pyi

**O que además estén vacíos**

useradd@LAPTOP-RISUJV9C:~ 15s $ **find . -iname "tExt\_file\*" -type f ! -perm 777 -empty**

**o menores de 10 megas de tamaño (si no pongo el menos entonces me busca justo el de tamaño 10)**

useradd@LAPTOP-RISUJV9C:~ 17s $ **find . -iname "tExt\_file\*" -type f ! -perm 777** **-size -10M**

./anaconda3/lib/python3.7/distutils/\_\_pycache\_\_/text\_file.cpython-37.pyc

./anaconda3/lib/python3.7/distutils/text\_file.py

./anaconda3/lib/python3.7/site-packages/jedi/third\_party/typeshed/stdlib/2and3/distutils/text\_file.pyi

**Sobre los filtros que he hecho no puedo ejecutar nada. Para poder hacerlo tengo que añadir exec (excecute)**

useradd@LAPTOP-RISUJV9C:~ 15s $ **find . -iname "tExt\_file\*" -type f -exec**

**Ahora el Sistema pondrá que ha encontrado cada archivo. Ojo no olvidar barra invertida y punto y coma.**

useradd@LAPTOP-RISUJV9C:~ 15s $ **find . -iname "tExt\_file\*" -type f -exec echo "Found it" \;**

Found it

Found it

Found it

Found it

Found it

Found it

Found it

Found it

Found it

**Las llaves son dónde se va a meter lo que se encuentre en el filtro.**

useradd@LAPTOP-RISUJV9C:~ 15s $ **find . -iname "tExt\_file\*" -type f -exec ls -s {} \;**

**esto nos dará error**

**No Podemos juntar la salida de un parámetro con la entrada de otro. Para eso creo una Shell(sh -c*)(-c es porque le voy a enviar parámetros*) mando los parámetros como comando. Es una Shell tipo bash porque es la que tengo en mi ordenador. Como hay 6 archivos, se crearán 6 shell**

useradd@LAPTOP-RISUJV9C:~ 16s $ **find . -iname "tExt\_file\*" -type f -exec sh -c " ls -s {} | wc" \;**

**si ahora incluyo el cat, leería todos los archivos**

useradd@LAPTOP-RISUJV9C:~ 16s $ **find . -iname "tExt\_file\*" -type f -exec cat {} \;**

**Si estamos ejecutando algo y queremos hacer otra cosa podemos bien abrir otra terminal o podemos acabar la línea de comando con & para que se ejecute en segundo plano**

rscheze@LAPTOP-RISUJV9C:~$ **find . -type f -name "\*.txt" -exec ls -l {} \; -exec sleep 5 \; &**

**Si le damos a fg(forward ground) e intro, entonces se ejecutará en primer plano (en mi ordenador esto para la ejecución). Supuestamente con bg iría a segundo plano (background)**

**Con Control Z suspendemos la ejecución. Lo ponemos en pause. Solo funciona en primer plano.**

**Jobs me dirá cuántos procesos tengo**

rscheze@LAPTOP-RISUJV9C:~$ **jobs**

[1]+ Stopped find . -type f -name "\*.txt" -exec ls -l {} \; -exec sleep 5 \;

**Ps -U me dice todos los procesos de mi usuario**

rscheze@LAPTOP-RISUJV9C:~$ **ps -U rscheze**

**Puedo ver con Jobs mis procesos y cargarme uno con kill**

rscheze@LAPTOP-RISUJV9C:~$ **jobs**

[1]+ Stopped find . -type f -name "\*.txt" -exec ls -l {} \; -exec sleep 5 \;

rscheze@LAPTOP-RISUJV9C:~$ **kill $1** *(El 1 es porque delante de +Stopped aparece un 1. Ese numero dependerá del trabajo que quiera cancelar)*

kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]

**CSV**

Primero instalarlo.

Revisa todos commandos de CSV.

**Csvlook (Aunque en los ejemplos viene entre comillas, me da error. Se soluciona poniendo comillas simples)**

csvlook: te abre un archivo y te permite moverte con el cursor por el archivo. Podemos combinarlo con -S (less -S). NO se puede utilizar para ficheros grandes porque esta revisando cada columna y su tamaño para ajustarlo para todas.

**csvlook -d “^” optd\_aircraft.csv | less -S**  (less -S lo pone en formato table y se ve mucho mejor)

csvlook -d “^” optd\_aircraft.csv > file.txt *(grabarlos)*

**csvstat Estadistica sobra cada columna**

csvstat -d “^” optd\_aircraft.csv | less

**csvcut toma las columnas de interés. Ahora además podemos tomar el nombre de la columna.**

csvcut -d “^” -c iata optd\_aircraft.csv

**se puede hacer sobre dos columnas del mismo fichero a la vez. Ojo entre las columnas no puede haber espacios.**

csvcut -d “^” -c iata\_code,manufacturer optd\_aircraft.csv

puedo sacar estadisticas sobre estas columnas

csvcut -d “^” -c iata\_code,manufacturer optd\_aircraft.csv | csvstat “,” *(la coma es por el delimitador, si no pongo nada coge la coma por defecto. Podría tener problemas si hay comas en los títulos )*

c**svgrep búscame una linea donde el iata code es 380**

csvgrep -d “^” iata\_code -m 380 optd\_aircraft.csv

**csvsql te crea una tabla que después podrías guardar como siempre >**

csvsql -d “^” -i mysql optd\_aircraft.csv

-I mysql es el tipo de sql que estamos utilizando . Hay varios tipos distintos.

csvsql -d “^” -i mysql optd\_aircraft.csv > table.sql

**Luego pongo cat para verlo por pantalla**

Cat table.sql