PEC 3: Expresiones regulares

El objetivo de esta PEC consiste en la utilización de expresiones regulares, mediante el comando `grep` para buscar patrones en varios tipos de documentos, y el comando `sed` para su modificación automática. También se usará el comando `awk` para filtrar datos y realizar pequeños informes.

Para la elaboración de esta PEC3 es necesario utilizar el usuario creado en la PEC1, aunque es posible utilizar cualquier otra máquina, siempre y cuando se use la misma versión de Ubuntu solicitada anteriormente.

Enunciado

Para el desarrollo de esta PEC3 vamos a utilizar en exclusiva el fichero adjunto con el enunciado llamado **headlines_words.csv** obtenido a partir de https://github.com/the-pudding/data/tree/master/women-in-headlines. El desarrollo de la PEC deberá hacerse en base al fichero que se adjunta con este enunciado ya que ha sido modificado, por lo que **NO os descarguéis el archivo**. Además, **el nombre del fichero de datos no debe cambiarse.** Todos los archivos deben estar en la misma carpeta.

Este fichero, con datos usados para el ensayo "When women make headlines" de Nicoletti y Sarva (https://pudding.cool/2022/02/women-in-headlines/), contiene el recuento y la frecuencia, entre otros, de las palabras buscadas en titulares, referidos a mujeres, de noticias en India, Gran Bretaña, Estados Unidos y Sudáfrica.

Ejercicio 1 (*1 punto*)

Dominar las expresiones regulares incluye también la capacidad de comprender y modificar expresiones regulares **creadas por otros programadores**. De hecho, es lo más habitual. Este ejercicio está centrado en la comprensión y modificación, si es el caso, de **expresiones regulares ya dadas**. Así, a partir de la siguiente expresión regular:

```
^[^;]*;[^;]*;[^;]*;all.*;[^;]*;[4-9][0-9]{2};[^;]*;[^;]*;[1-3];
```

Explica la utilidad (sin describir técnicamente la estructura de la expresión regular) que se pretende obtener con la expresión regular anterior, en un fichero de texto llamado *Ex3_1.txt*.

Se requiere una explicación concisa y práctica en relación al dataset proporcionado, no una descripción de los cuantificadores y delimitadores. Por ejemplo, encontrar todos los registros cuyo campo X no sea nulo, personas menores de 50 años y que pesen menos de 70 Kg, etc. **No hay que ejecutar comandos ni código**, el entregable que se indica abajo tan sólo debe tener la explicación textual siguiendo las directrices que se han facilitado.

Entregables: `Ex3_1.txt`

Ejercicio 2 (*2 puntos*)

A partir del fichero adjunto al enunciado de esta PEC3, **headlines_words.csv**, vamos a construir incrementalmente **una única expresión regular** con **grep** del tipo **ERE** que devuelva registros de los dos primeros años y con escasa presencia.

a) (0,5 puntos) Cread una única expresión de tipo ERE que devuelva sólo los registros que sean del año 2010 o 2011.

La expresión regular se debe usar dentro de un script en Bash llamado *script3_2_a.sh* que ejecute un comando **`grep`**. El script debe funcionar ejecutándose del siguiente modo:

```
./script3_2_a.sh ./headlines_words.csv
```

A continuación se muestran *algunas* de las líneas de la salida esperada para que veáis lo que se pide:

```
./script3_2_a.sh ./headlines_words.csv

…
11172;65501;2010;all
countries;housewife;48;35448;0.001354096;9;926;926;female stereotypes
11173;65502;2011;all
countries;housewife;51;58436;0.00087275;9;926;926;female stereotypes
11184;65671;2010;all countries;gossip;234;35448;0.006601219;6;936;935;female
stereotypes
11185;65672;2011;all countries;gossip;312;58436;0.005339174;6;936;935;female
stereotypes
11196;65722;2010;all countries;fat;87;35448;0.002454299;3;939;939;female
stereotypes
11197;65723;2011;all
countries;fat;105;58436;0.0017968379999999;3;939;939;female stereotypes
```

```
11208;65739;2010;all countries;sexy;135;35448;0.003808395;4;941;941;female
stereotypes
11209;65740;2011;all countries;sexy;147;58436;0.002515573;4;941;941;female
stereotypes
…
```

b) (0,5 puntos) Ampliad la expresión regular anterior para obtener una única expresión de tipo ERE que devuelva los registros:

- que sean del año 2010 o 2011,

- y que el valor de la columna *country* no sea "all countries"

La expresión regular se debe usar dentro de un script en Bash llamado *script3_2_b.sh* que ejecute un comando **`grep.`** El script debe funcionar ejecutándose del siguiente modo:

```
./script3_2_b.sh ./headlines_words.csv
```

A continuación se muestran *algunas* de las líneas de la salida esperada para que veáis lo que se pide:

```
./script3_2_b.sh ./headlines_words.csv
…
8928;51085;2010;USA;gangrape;0;1912;0.0;8;856;856;crime and violence
8929;51086;2011;USA;gangrape;0;3274;0.0;8;856;856;crime and violence
8952;52173;2010;USA;temple;0;1912;0.0;6;930;930;race, ethnicity and identity
8953;52174;2011;USA;temple;0;3274;0.0;6;930;930;race, ethnicity and identity
8964;52224;2010;USA;witchcraft;0;1912;0.0;10;935;935;female stereotypes
8965;52225;2011;USA;witchcraft;2;3274;0.000610874;10;935;935;female
stereotypes
…
```

c) (1 punto) Cread la expresión regular final, **una única expresión regular** con **grep** del tipo **ERE** que cumpla exactamente todas las condiciones que se detallan a continuación:

- sean del 2010 o 2011,
- el valor de la columna *freq_rank* sea mayor o igual a 910,
- y que el valor de la columna *country* no sea "all countries"

La expresión regular se debe usar dentro de un script en Bash llamado *script3_2.sh* que ejecute un comando **`grep.`** El script debe funcionar ejecutándose del siguiente modo:

```
./script3_2_c.sh ./headlines_words.csv
```

A continuación se muestran *algunas* de las líneas de la salida esperada para que veáis lo que se pide:

```
./script3_2_c.sh ./headlines_words.csv
…
6709;38659;2011;UK;tribal;0;2397;0.0;6;928;928;race, ethnicity and identity
6720;38743;2010;UK;witchcraft;1;1504;0.000664894;10;935;935;female
stereotypes
6721;38744;2011;UK;witchcraft;1;2397;0.000417188;10;935;935;female
stereotypes
8952;52173;2010;USA;temple;0;1912;0.0;6;930;930;race, ethnicity and identity
8953;52174;2011;USA;temple;0;3274;0.0;6;930;930;race, ethnicity and identity
…
```

Para más información sobre expresiones regulares ERE, consultad el apartado *"2.9. Tipos de expresiones regulares POSIX"* de *"Expresiones regulares"* en los materiales docentes del curso.

En la entrega deben adjuntarse los scripts *script3_2_x.sh* y una captura para cada script PEC3_2_x.png con el usuario de la PEC1 en donde se vea la invocación y la salida que genera cada script, usando el ejemplo de ejecución anterior.

Entregables: `script3_2_a.sh`, `script3_2_b.sh`, `script3_2_c.sh` y `PEC3_2_a.png`, `PEC3_2_b.png`, `PEC3_2_c.png`

Ejercicio 3 (*3 puntos*)

Este ejercicio sirve para **ilustrar y mostrar el uso del editor `sed`**. Éste es especialmente útil para hacer sustituciones de texto mediante expresiones regulares, aunque sus funcionalidades no se limitan a eso. En este ejercicio tendréis que poner en práctica el uso de `sed` para diferentes tareas.

A partir del dataset adjunto a la PEC3, es decir, del fichero **headlines_words.csv**, cread un script denominado `script3_3.sed` que lleve a cabo las siguientes acciones de forma simultánea **sin modificar el fichero de entrada**:

- Elimine las dos primeras columnas.
- Elimine todas las líneas con recuento 0.
- Mueva el punto decimal de *freq_prop_headlines* dos lugares atrás para obtener el tanto por ciento en lugar del tanto por uno, conservando todos los dígitos y eliminando los ceros inecesarios antes del separador decimal. Algunos registros están en notación científica (todos elevados a -5) y también deben tratarse, mostrándose en notación decimal.

- Debe mantenerse la cabecera para los campos no modificados, eliminar los títulos correspondientes a los campos borrados y cambiar *freq_prop_headlines* por *%_headlines*.

El script debe funcionar usando la siguiente invocación:

```
sed -f script3_3.sed headlines_words.csv
```

A continuación, se muestran algunas de las líneas, **aleatorias**, de la salida esperada para que veáis lo que se pide:

```
…
2019;UK;gay;37;15504;0.2386481;3;824;824;race, ethnicity and identity
2019;UK;hindu;1;15504;0.00645;5;875;875;race, ethnicity and identity
2010;all countries;police;192;35448;0.5416385;6;318;304;crime and violence
2018;India;career;76;19170;0.3964528;6;757;757;empowerment
2015;all countries;entrepreneur;192;212576;0.0903206;12;514;506;empowerment
2016;USA;vagina;12;12290;0.0976404;6;115;115;female stereotypes
2016;UK;marry;73;11237;0.6496396;5;580;580;female stereotypes
2017;USA;dalit;1;12771;0.007829999999999999;5;118;118;race, ethnicity and
identity
2020;USA;career;52;13040;0.398773;6;225;225;empowerment
…
```

En la entrega debe adjuntarse el script *script3_3.sed* y una captura PEC3_3.png con el usuario de la PEC1 en donde se vea la invocación y la salida que genera usando el ejemplo de ejecución anterior.

Entregables: `script3_3.sed y PEC3_3.png`

Ejercicio 4 (*3 puntos*)

Como se ha visto en los materiales docentes, `awk` es una herramienta con muchas posibilidades para filtrar datos, generar informes, etc. Es, de hecho, un lenguaje de programación, por lo que se pueden crear scripts `awk` para automatizar tareas.

En este ejercicio se tendrá que crear un script en `awk` llamado *script3_4.awk* que a partir del dataset adjunto **headlines_words.csv** compare la evolución de la frecuencia de una palabra en dos ámbitos geográficos. El script recibirá como parámetros la palabra y dos ámbitos geográficos.

El *script3_4.awk* creará **dos ficheros, uno para cada ámbito**, con sólo las columnas *year* y *freq_prop_headlines* correspondientes a dicha palabra y lugar, separadas por exactamente un espacio y sin contener ningún otro espacio en el resto de la línea. Los valores que en el original están en **notación científica**

deberán trasladarse a notación decimal estándar. El valor de *freq_prop_headlines* deberá aparecer con **exactamente 8 decimales (**a partir de la función printf de gawk**)**. Los ficheros no debe tener cabecera.

El nombre de cada fichero generado debe estar formado por **la palabra, un guión bajo y el nombre del ámbito geográfico**. La extensión será **ssv**. En el caso de "all countries", en el nombre del fichero deberá aparecer sólo "all".

Por ejemplo, la llamada (toda en una sola línea) generaría estos dos ficheros:

```
gawk -f script3_4.awk -v word="tribal" -v country1="UK" -v country2="India" headlines_words.csv
```

Salida tribal_UK.ssv:

```
2010 0.00000000
2011 0.00000000
2012 0.00000000
2013 0.00000000
2014 0.00014333
2015 0.00000000
2016 0.00008900
2017 0.00000000
2018 0.00007820
2019 0.00006450
2020 0.00006400
2021 0.00000000
```

Salida tribal_India.ssv

```
2010 0.00063131
2011 0.00041017
2012 0.00110071
2013 0.00081466
2014 0.00146628
2015 0.00170908
2016 0.00185555
2017 0.00222261
2018 0.00208659
2019 0.00282987
2020 0.00145839
2021 0.00182222
```

- El script3_4.awk mostrará en la salida estándar, para cada país, el nombre del ámbito geográfico, su **media aritmética** y su **desviación típica**, ambos valores con **cinco decimales**. Esta información irá encabezada por el título de cada columna en inglés.

Ejemplos (las invocaciones son todas de una sóla línea):

```
gawk -f script3_4.awk -v word="tribal" -v country1="UK" -v country2="India" headlines_words.csv
```

```
Country,Average,Standard Deviation
UK,0.00004,0.00005
India,0.00153,0.00068
```

gawk -f script3_4.awk -v word="police" -v country1="USA" -v country2="all countries" headlines_words.csv

```
Country,Average,Standard Deviation
USA,0.02174,0.00684
all countries,0.00864,0.00257
```

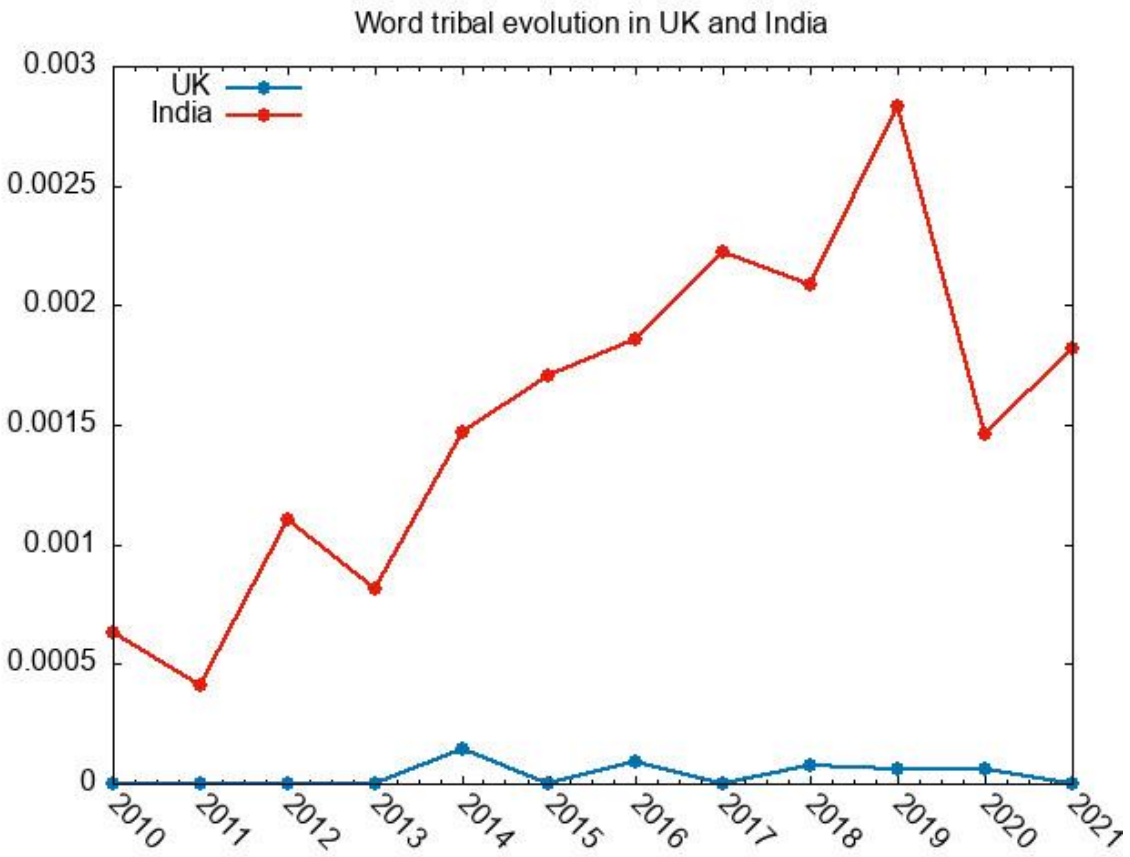
Opcionalmente, se puede comprobar que los ficheros ssv están correctamente generados llamando al script adjunto *graph.sh* (debéis tener **gnuplot** instalado en vuestro sistema) con la invocación siguiente:

./graph.sh <palabra> <country1> <country2>

Un ejemplo de invocación de graph (opcional) sería:

./graph.sh tribal UK India

En caso de ser correctos los datos debería crear el fichero *tribal_UK_India.jpg* con la siguiente representación gráfica:



Nota: Para comprobar el correcto funcionamiento del script durante la corrección se analizará el código fuente y se podrán llevar a cabo juegos de pruebas adicionales, por lo que aunque la salida coincida con la de los ejemplos obligatorios que debéis utilizar, no es garantía para obtener la puntuación máxima del ejercicio, así que probad otras combinaciones de parámetros,

En la entrega debe adjuntarse el script *script3_4.awk* y una captura PEC3_4.png con el usuario de la PEC1 en donde se vea la invocación usando los tres ejemplos y la salida generada.

Entregables: `script3_4.awk y PEC3_4.png`.

Valoración global (*1 punto*)

Se hará una valoración global de la PEC3 atendiendo entre otros a los siguientes criterios: se respetan los aspectos formales de entrega, la forma de invocación, la

claridad del código, el orden, la capacidad de síntesis, los comentarios del código, el nombre significativo de las variables, se gestiona correctamente el parámetro que indica la ruta de dataset de entrada, etc.

Entrega

Un único fichero en formato **zip** que contenga todos los ficheros en la misma carpeta:

- Una única captura de pantalla que contenga la ejecución de cada script para cada ejercicio, a excepción del primero, tal y como se indica al final de la descripción de cada ejercicio, asegurándose que:
  - Se ve la invocación del comando así como la salida (o una parte de ella).
  - Aparezca siempre claramente el usuario de la UOC creado en la PEC1.
  - Se respeta la nomenclatura establecida.
- Fichero de texto plano *Ex3_1.txt* con la respuesta al primer ejercicio.
- Los scripts *script3_2_a.sh, script3_2_b.sh, script3_2_c.sh, script3_3.sed, script3_4.awk* listos para ser ejecutados en Ubuntu. **En caso de manipularlo con otro sistema operativo diferente a Ubuntu**, *Macintosh especialmente*, **prestad mucha atención a la codificación del fichero**, que debe ser UTF-8, y con el final de línea correcto. *Si el script no se ejecuta correctamente en Ubuntu durante su corrección porque está mal la codificación, o hay otro problema técnico, no recibiréis puntuación***.**
- **No hay que enviar el Dataset.**

Para los ejercicios 2, 3 y 4 no se puntuará ningún apartado en donde falte la captura o el código. Ambos son **IMPRESCINDIBLES**.

NOTA IMPORTANTE: E*n una asignatura donde la automatización de tareas es el objetivo, respetar todos los nombres de los scripts es muy importante, vigilad el uso de mayúsculas y minúsculas (GNU/Linux es sensible a esto, a diferencia de otros sistemas operativos como Windows), el uso de guiones bajos, etc.*

NOTA IMPORTANTE: No se aceptará en ningún caso no justificado la entrega de la PEC3 después de la fecha máxima de entrega (14/05/2023*). Si por alguna razón pensáis que no vais a poder entregar a tiempo, consultadlo con vuestro profesor siempre con varios días de anterioridad.*