Práctica

El formato de este ejercicio es más abierto que el resto de las entregas anteriores, ya que consiste en el reto de desarrollar un proyecto propio de interés personal.

La finalidad es que cada estudiante escoja un objetivo de análisis de datos sobre un conjunto abierto y que elabore un pequeño proyecto. La pregunta sobre el porqué, y el qué hacer, se deja a vuestra elección e intereses.

Es imprescindible que en **la memoria** se responda de forma ordenada y numerada a las preguntas del ejercicio 2 y aparezcan todas las capturas de pantalla en las que se vea la ejecución de cada uno de los scripts y su resultado.

Buscad y elegid un **dataset** para su análisis, no olvidéis **incluirlo en el zip**. Dicho dataset debe tener un **mínimo de 300 registros** y un **peso inferior a 1.5 MB** (sin comprimir).

En la práctica se evaluarán solamente los aspectos de carácter técnico. En la práctica debéis **utilizar el usuario creado en la PEC 1** y demostrar la ejecución de los scripts, mediante capturas de pantalla en la memoria.

La entrega la realizareis mediante un fichero **en formato zip** (se adjunta un ejemplo junto al enunciado) con todo lo siguiente:

1. Una **memoria** en formato **PDF**, con una extensión no superior a **10 páginas** (excluyendo portada e índice, de haberlos), que incluya **capturas de pantalla**, donde se aprecie vuestro usuario, y se vea la ejecución de cada script, junto con el resultado íntegro o bien parte de éste. *Si no hay imágenes en la memoria donde se vea vuestro usuario, se considerará la PEC como plagiada.*

2. Cada uno de los scripts pedidos se han de adjuntar en ficheros listos para ser ejecutados, **con el nombre y la extensión indicados en los**

ejercicios, en los que se incluirá el código fuente. Cercioraros bien que al ejecutar cada script la salida es la esperada y que funciona correctamente, ya que será ejecutado en la corrección.

3. El **dataset** que hayáis elegido para el desarrollo de la práctica.

Es responsabilidad del estudiante que todo el código funcione correctamente con una simple ejecución sin tener que editar nada. Para ello es imprescindible usar rutas relativas en vez de absolutas, puesto que el profesorado tendrá su propia estructura de directorios. Evitad rutas dentro del script que impliquen que tengamos que reproducir vuestro ambiente de trabajo para corregir.

Todos los ficheros que se incluyan dentro del fichero zip (scripts, memoria y dataset) tienen que estar ubicados en la misma carpeta.

1. Scripts

Puntuación: 7 puntos

A. Cread un script denominado `a.sh` que descargue el dataset, *siempre que sea posible*, y utilizando las herramientas vistas durante el curso muestre:
- La URL de descarga del dataset.
- Formato del o de los ficheros que lo componen.
- Número de columnas y registros.
- Tipo de datos (entero, decimal, fecha, texto, etc).

El script deberá al menos contemplar una opción / argumento de línea de comandos.

Proponemos la siguiente idea:

- *Si el script se ejecuta sin opciones, muestra la URL de descarga y el número de columnas y registros. En cambio, si se incluye en su*

invocación la opción `-v` deberá además mostrar el formato del o de los ficheros y el tipo de datos de sus columnas.

En este sentido, os proporcionamos una pequeña estructura a modo de ejemplo:

```
#!/bin/bash

while getopts "o:" option; do

  case $option in

    o) L=${OPTARG}

      echo "Cambiamos nombre dataset a --> $L"

      exit;;

...

  esac

done
```

El script debe funcionar usando la invocación:

./a.sh [-v]

(1 punto)

B. Elaborar un script `b.ext` (con Bash, sed o awk) que implemente un sistema de filtrado o de transformación sobre los datos haciendo uso de **expresiones regulares complejas**. Es necesario definir un objetivo en el filtro o en la transformación (¿Por qué estamos realizando este filtro o esta transformación? ¿Qué aporta?)

El script **debe tener un mínimo de 7 líneas** que hagan transformaciones con los datos, es decir, sentencias que modifiquen los datos o generen una estructura nueva a partir de expresiones regulares, `grep`, `sed`, etc. No se computarán como líneas efectivas de código aquellas sentencias que no manipulen los datos. Por ejemplo, no se computarán sentencias básicas para visualizar información (`echo`, `print`, `cat`, etc.), comentarios en el código, asignaciones, construcción de estructuras (`if`, `for`, `while`, `fi`, `do`, `done`, etc.), etcétera. En definitiva, el script debe demostrar claramente el dominio adquirido de las herramientas del curso para el tratamiento de datos.

El script debe funcionar usando la invocación:

./b.ext

Donde 'ext' es la extensión correspondiente en cada caso.

(1 punto)

C. El proyecto debe incorporar al menos la elaboración de **dos** scripts (con los procesos que deseéis). Por ejemplo, hacer más transformaciones sobre los datos, o corregir posibles errores o incongruencias en los datos, en el supuesto de que las haya.

Cada uno de los dos scripts tendrá que contener un proceso "iterativo" (con estructuras tipo `while` o `for`). Uno de los scripts tiene que estar escrito en `Bash` y el otro íntegramente en `awk`.

Cada uno de los dos script debe tener **un mínimo de 7 líneas** que hagan transformaciones con los datos, es decir, sentencias que modifiquen los datos o generen una estructura nueva (combinar dos campos en uno nuevo, filtrar, agregar, ordenar, etc.). A tal efecto, no se computarán sentencias que no manipulen los datos, como las sentencias básicas para visualizar información o construir estructuras (`echo`, `print`, `cat`, `if`, `for`, `while`, `fi`, `do`, `done`, etc.), comentarios, asignaciones, etcétera. En

definitiva, los scripts deben demostrar el dominio adquirido en las herramientas del curso para el tratamiento de datos.

El script en Bash se ha de adjuntar por medio de un fichero llamado `c_bash.sh`, mientras que el script en awk se almacenará en otro fichero llamado `c_awk.awk`.

(1 punto para el primer script y 1.5 puntos para el segundo.)

Si contiene la estructura iterativa, pero no se llega a las 7 líneas, la mitad de puntuación. Si contiene la estructura iterativa y alcanza o supera las 7 líneas, entonces la puntuación entera).

Los script deben funcionar usando la invocación siguiente:

./c_bash.sh <nombreDelFicheroDeDatos>

gawk -f c_awk.awk <nombreDelFicheroDeDatos>

(2.5 puntos)

D. Además, elaborad un script llamado `d.sh` que genere un documento en formato de texto plano con formato atractivo o HTML5 **con resultados agregados por categorías o por intervalos**.

Por ejemplo, si los datos fueran de acceso a una plataforma en línea, se podría realizar un informe con el número de usuarios por cada provincia, por edad o por otra agrupación de los resultados.

No se considerará válido una simple visualización (total o parcial) del dataset. Debe haber cálculos en forma de agrupaciones o categorías. Se revisará la veracidad de los cálculos y **se valorará la sofisticación de la maquetación**, a saber, los recursos de formato (color, negrita, tablas vs CSS, etc.).

El script debe funcionar usando la invocación:

./d.sh

(1.5 puntos)

E. Es necesario que haya un **script principal** (además de los anteriores), llamado `run.sh` que ejecute paso a paso todo el proyecto, llamando a los scripts anteriores. Debe encargarse de la ejecución de todos los scripts indicados en los apartados A, B, C, y D.

Debéis tener en consideración que **todo debe funcionar sin tener que editar nada, ni llevar a cabo ninguna acción adicional** de ningún tipo. Intentad que el **tiempo máximo de ejecución** no supere los **10 segundos**.

Seremos flexibles en el caso de la necesidad de instalar paquetes que no vengan de serie con la distribución o que no se hayan instalado durante la PEC1, eso sí, vuestro script `run.sh` deberá instalarlos.

El script debe funcionar usando la invocación:

./run.sh

(1 punto)

2. Memoria

Puntuación: 2 puntos

Responded en la memoria de forma ordenada y numerada a cada uno de los aspectos que se indican a continuación:

A. Objetivos del proyecto (qué problema se pretende resolver, y por qué se puede resolver mediante el uso de *scripting*).

B. Es necesario definir el objetivo del script `b` (¿Por qué estamos realizando este filtro o esta transformación? ¿Qué aporta?).

C. Se deben explicar los objetivos de los scripts `c_bash.sh` y `c_awk.awk`.

D. Se deben enumerar aspectos a tener en cuenta y a valorar del proyecto.

E. Explicad las tareas más importantes aprendidas con la elaboración del proyecto y las dificultades que hayáis encontrado.

F. Proponed técnicas para mejorar o extender el proyecto.

3. Valoración global del proyecto

Puntuación: 1 punto

Además de los puntos anteriores, se realizará una valoración global del proyecto: la claridad/presentación, el orden, la adecuación lingüística, los comentarios del código, la nomenclatura de las variables, la sofisticación del trabajo, cumplir los requisitos del enunciado, etc. *(1 punto)*

Resumen de la entrega

- Memoria, como documento en pdf, con capturas de la ejecución de cada script con vuestro usuario.
- Dataset con mínimo de 300 líneas y no superior a 1.5MB (sin comprimir).
- Cada uno de los scripts pedidos listos para ser ejecutados en Ubuntu 20.04.

Todo ello en un fichero en formato zip.

Comentarios

Para la elaboración de este último proyecto deben utilizarse las herramientas utilizadas en las PECs anteriores. Es posible que, para algunas operaciones en concreto, debáis utilizar otras herramientas, posibilidad que está permitida también. En este caso, será preferible utilizar herramientas basadas en terminal. En todo caso, eso sí, debe utilizarse únicamente un entorno basado en GNU/Linux, y de utilizar herramientas no instaladas en la PEC1, el script run.sh deberá instalarlas.

Es **imprescindible** también que todo lo que se haga en este proyecto **se pueda reproducir mediante la ejecución de los scripts**.

No se aceptará en ningún caso la entrega de la práctica después de la fecha máxima de entrega. *Si por alguna razón pensáis que no vais a poder entregar a tiempo, consultadlo con vuestro profesor siempre con anterioridad*.

Fecha máxima de entrega: 02/01/2022 a las 23:59:59