

# Actividad Evaluable 1

## Descripción

MÓDULO	Seminario Internacional en Herramientas y Técnicas de Detección de Ciberamenazas
ASIGNATURA	Data Science Aplicado a la Ciberseguridad
Fecha Límite de Entrega	17 de abril de 2023, a las 23:59
Puntos	20% de la Nota Total.
Carácter	Grupo (max 2 personas)

## Enunciado:

En esta actividad se planteará una serie de preguntas relacionadas con los temas vistos en las sesiones 1 y 2. Los estudiantes debe responder a tales preguntas en este mismo documento, de forma clara y concisa. Este documento debe ser exportado a PDF, y entregado a través de la página de la asignatura, antes de la fecha límite de entrega.

Se considerará tanto la corrección de las soluciones como su presentación y el código utilizado para la obtención de los resultados.

Parte de esta actividad implica ejecutar código R. Tal código debe ser entregado en un fichero de código R (extensión `.R`), éste debe poderse ejecutar directamente sobre un terminal nuevo en R o en RStudio. El código es imprescindible para la corrección del ejercicio.

**Las entregas tardías serán marcadas como “tarde”, y pueden NO ser evaluadas. Por favor, entregad a tiempo.**

# 1. Data Science

## Pregunta 1:

De las siguientes preguntas, clasifica cada una como descriptiva, exploratoria, inferencia, predictiva o causal, y razona brevemente (una frase) el porqué:

1. Dado un registro de vehículos que circulan por una autopista, disponemos de su marca y modelo, país de matriculación, y tipo de vehículo (por número de ruedas). Con tal de ajustar precios de los peajes, ¿Cuántos vehículos tenemos por tipo? ¿Cuál es el tipo más frecuente? ¿De qué países tenemos más vehículos?
2. Dado un registro de visualizaciones de un servicio de video-on-demand, donde disponemos de los datos del usuario, de la película seleccionada, fecha de visualización y categoría de la película, queremos saber ¿Hay alguna preferencia en cuanto a género literario según los usuarios y su rango de edad?
3. Dado un registro de peticiones a un sitio web, vemos que las peticiones que provienen de una red de telefonía concreta acostumbran a ser incorrectas y provocarnos errores de servicio. ¿Podemos determinar si en el futuro, los próximos mensajes de esa red seguirán dando problemas? ¿Hemos notado el mismo efecto en otras redes de telefonía?
4. Dado los registros de usuarios de un servicio de compras por internet, los usuarios pueden agruparse por preferencias de productos comprados. Queremos saber si ¿Es posible que, dado un usuario al azar y según su historial, pueda ser directamente asignado a un o diversos grupos?

---

## AQUÍ TU RESPUESTA

1. Se clasifica como pregunta **DESCRIPTIVA**: Esta pregunta busca describir la distribución de los vehículos en la autopista según su tipo, marca y país de matriculación. La respuesta se basará en los datos disponibles en el registro de vehículos.
2. Se clasifica como pregunta **EXPLORATORIA**: Esta pregunta busca explorar la relación entre las preferencias de género literario de los usuarios y su rango de edad. No se plantea ninguna hipótesis específica y el objetivo es obtener una idea general sobre la relación entre estas variables.
3. Se clasifica como pregunta **INFERENCIA**: Esta pregunta busca inferir si los próximos mensajes de una red de telefonía seguirán provocando errores de servicio. Se basa en la observación de que las peticiones de esa red suelen ser incorrectas y en la hipótesis de que esta tendencia puede continuar en el futuro.

4. Se clasifica como pregunta PREDICTIVA: Esta pregunta busca predecir a qué grupo o grupos de preferencias de productos pertenece un usuario en función de su historial de compras. Se basa en la creencia de que la información de compra pasada puede ser utilizada para hacer predicciones precisas sobre futuras compras.

En resumen, la primera pregunta es descriptiva porque busca describir características del conjunto de datos, la segunda es exploratoria porque busca explorar la existencia de una posible relación entre variables, la tercera es inferencial porque busca inferir información basándose en datos previos y la cuarta es predictiva porque busca predecir resultados futuros basados en datos previos.

## Pregunta 2:

Considera el siguiente escenario:

Sabemos que un usuario de nuestra red empresarial ha estado usando esta para fines no relacionados con el trabajo, como por ejemplo tener un servicio web no autorizado abierto a la red (otros usuarios tienen servicios web activados y autorizados). No queremos tener que rastrear los puertos de cada PC, y sabemos que la actividad puede haber cesado. Pero podemos acceder a los registros de conexiones TCP de cada máquina de cada trabajador (hacia donde abre conexión un PC concreto). Sabemos que nuestros clientes se conectan desde lugares remotos de forma legítima, como parte de nuestro negocio, y que un trabajador puede haber habilitado temporalmente servicios de prueba. Nuestro objetivo es reducir lo posible la lista de posibles culpables, con tal de explicarles que por favor no expongan nuestros sistemas sin permiso de los operadores o la dirección.

Explica con detalle cómo se podría proceder al análisis y resolución del problema mediante Data Science, indicando de donde se obtendrían los datos, qué tratamiento deberían recibir, qué preguntas hacerse para resolver el problema, qué datos y gráficos se obtendrían, y cómo se comunicarían estos.

---

### AQUÍ TU RESPUESTA

En este escenario, el objetivo es identificar al usuario que ha estado usando la red empresarial para fines no relacionados con el trabajo, con el fin de tomar medidas para evitar que se repita en el futuro. Describiremos cómo se podría abordar este problema mediante Data Science:

1. Obtención de datos: Se obtienen los registros de conexiones TCP de cada máquina de cada trabajador, donde se registran las conexiones entrantes y salientes. Estos registros podrían obtenerse a través de herramientas de monitoreo de red, como Wireshark o TCPdump.
2. Tratamiento de datos: Los registros de conexiones TCP deben ser limpiados y transformados para su análisis. Esto puede incluir la eliminación de registros duplicados o irrelevantes, y la agregación de registros por usuario y por período de tiempo (por ejemplo, por día o por semana).

3. Preguntas clave: Una vez que se tienen los datos limpios y transformados, se pueden hacer las siguientes preguntas clave:
  - ¿Cuántas conexiones TCP han sido registradas por usuario y por período de tiempo?
  - ¿Qué tipo de conexiones se han registrado (por ejemplo, HTTP, FTP, SSH)?
  - ¿Qué máquinas han abierto conexiones hacia el servicio web no autorizado?
  - ¿Qué usuarios han abierto conexiones hacia el servicio web no autorizado?
  - ¿Cuándo se han registrado estas conexiones?
4. Análisis de datos: A partir de los datos tratados y las preguntas clave, se pueden realizar varios análisis, como:
  - Análisis de frecuencia: identificación de los usuarios que han registrado más conexiones y de qué tipo, y qué máquinas han abierto más conexiones hacia el servicio web no autorizado.
  - Análisis de patrones: identificación de patrones de conexión por parte de los usuarios sospechosos y comparación con los patrones de conexión típicos de los usuarios legítimos.
  - Análisis de correlación: identificación de posibles correlaciones entre las conexiones registradas por los usuarios sospechosos y los eventos de actividad no autorizada.
5. Visualización de datos. En cuanto a los gráficos recomendados para un foro de nivel ejecutivo, se podrían obtener los siguientes:
  - Gráfico de barras: para comparar el número de conexiones TCP de cada máquina.
  - Gráfico de líneas: para mostrar la evolución temporal del número de conexiones TCP.
  - Gráfico de torta: para mostrar el porcentaje de conexiones TCP realizadas por cada usuario.
  - Gráfico de dispersión: para visualizar la relación entre el número de conexiones TCP de cada máquina y el uso del servicio web no autorizado.
6. Comunicación de resultados: Los resultados del análisis de datos se deben comunicar de manera clara y concisa a la dirección de la empresa y al usuario sospechoso. Se puede utilizar un informe o presentación para mostrar los hallazgos y explicar el proceso utilizado para llegar a ellos.

## 2. Introducción a R y Datos Elegantes

El segundo apartado de la práctica consiste en el análisis de un fichero de registro de peticiones HTTP, que debéis descargar (fichero adjunto: [logs-http.zip](#) ), cargar en R, y realizar un análisis

Se recomienda tener cierto nivel de familiaridad y al alcance los cheatsheet de los distintos packages mencionados en las sesiones de teoría para un análisis más fácil:

- readr
- stringr
- tidyr (separate)
- dplyr (mutate, count)

Alternativamente, recordad que podéis consultar la sección de ayuda de RStudio y buscar en la documentación los parámetros así como ejemplos de uso (al final de cada página de documentación) para las funciones (escribiendo `?<nombre-funcion>` o presionando F1 sobre el nombre de la función).

Para las siguientes preguntas se requiere usar R. Indica en este documento para cada pregunta el resultado obtenido, describiendo a grandes rasgos el procedimiento seguido para la obtención de la respuesta, justificando cada decisión tomada a la hora de manipular los datos (descartar, agrupar, transformar, etc).

Asegúrate de entregar también el código en un fichero aparte, para poder ejecutarse directamente en un terminal limpio de R.

## Pregunta 1:

Una vez cargado el Dataset a analizar, comprobando que se cargan las IPs, el Timestamp, la Petición (Tipo, URL y Protocolo), Código de respuesta, y Bytes de reply.

- Primero cargamos el archivo correspondiente: epa\_http

```
library(readr)
epa_http <- read_table("C:/Users/anker/OneDrive/Escritorio/epa-http/epa-http.csv", col_names = FALSE)
View(epa_http)
```
- 1. Importa la librería readr, que proporciona funciones para leer y escribir archivos de datos en formato rectangular, como CSV, TSV y otros formatos similares.
- 2. Luego lee el archivo CSV "epa-http.csv" ubicado en la ruta especificada "C:/Users/anker/OneDrive/Escritorio/epa-http/" utilizando la función read\_table() de readr.
- 3. Posteriormente asigna el resultado de la lectura del archivo a la variable epa\_http.
- 4. Muestra la variable epa\_http en la vista de datos de RStudio utilizando la función View().

Para mayor comodidad cambiaremos el nombre de la DF y de las columnas respectivamente

```
DFPRINCIPAL <- epa_http
colnames(DFPRINCIPAL) <- c("IP", "TIEMPO", "TIPOPET", "URL", "PROTOCOLO", "CODRESPUESTA", "BYTERESPUESTA")
```

### 1. ¿Cuales son las dimensiones del dataset cargado (número de filas y columnas)?

- dim(DFPRINCIPAL)
- 1. La función dim() se utiliza para obtener la dimensión de una DF en R en vector

```
> View(epa_http)
> DFPRINCIPAL <- epa_http
> colnames(DFPRINCIPAL) <- c("IP", "TIEMPO", "TIPOPET", "URL", "PROTOCOLO", "CODRESPUESTA", "BYTERESPUESTA")
> dim(DFPRINCIPAL)
[1] 47748      7
> |
```

En este caso devolvió 47748 filas y 7 columnas

## 2. Valor medio de la columna Bytes

3.

```
DFPRINCIPAL$BYTERESPUESTA <- as.numeric(DFPRINCIPAL$BYTERESPUESTA)
mean(DFPRINCIPAL$BYTERESPUESTA, na.rm = TRUE)
```

1. Convierte la columna BYTERESPUESTA en el dataframe DFPRINCIPAL en un vector numérico utilizando la función `as.numeric()`. Esto es necesario porque a veces las columnas en un dataframe pueden ser almacenadas como factores o caracteres en lugar de números, y se debe convertir a numérico para poder realizar operaciones matemáticas.
2. Calcula la media (promedio) de los valores en la columna BYTERESPUESTA del dataframe DFPRINCIPAL. El argumento `na.rm = TRUE` le dice a la función `mean()` que ignore cualquier valor faltante (NA) en la columna al calcular la media. El resultado de esta línea de código es un número que representa la media de los valores numéricos en la columna BYTERESPUESTA del dataframe.

```
> DFPRINCIPAL$BYTERESPUESTA <- as.numeric(DFPRINCIPAL$BYTERESPUESTA)
Warning message:
NAs introduced by coercion
> mean(DFPRINCIPAL$BYTERESPUESTA, na.rm = TRUE)
[1] 7352.335
>
```

## Pregunta 2:

De las diferentes IPs de origen accediendo al servidor, ¿cuántas pertenecen a una IP claramente educativa (que contenga ".edu")?

```
- SUBDFPRINCIPAL_EDU <-DFPRINCIPAL[grepl("\\.edu", DFPRINCIPAL$IP), ]  
- nrow(SUBDFPRINCIPAL_EDU)  
-
```

1. En este código, la función `grepl()` busca el patrón ".edu" idénticamente en toda la cadena de caracteres de la columna "IP" de la DF "DFPRINCIPAL\_EDU". La función devuelve un vector lógico de TRUE or FALSE para indicar qué filas de la DF contienen el patrón. Luego, se utiliza este vector lógico para seleccionar solo las filas que contienen el patrón ".edu" y se guardan en una nueva DF llamada "SUBDFPRINCIPAL\_EDU".
2. En este código `nrow` obtenemos la cantidad de filas dentro de la DF SUBDFPRINCIPAL\_EDU

```
> SUBDFPRINCIPAL_EDU <-DFPRINCIPAL[grepl("\\.edu", DFPRINCIPAL$IP), ]  
> View(SUBDFPRINCIPAL_EDU)  
> nrow(SUBDFPRINCIPAL_EDU)  
[1] 6524  
> |
```

## Pregunta 3:

De todas las peticiones recibidas por el servidor cual es la hora en la que hay mayor volumen de peticiones HTTP de tipo "GET"?

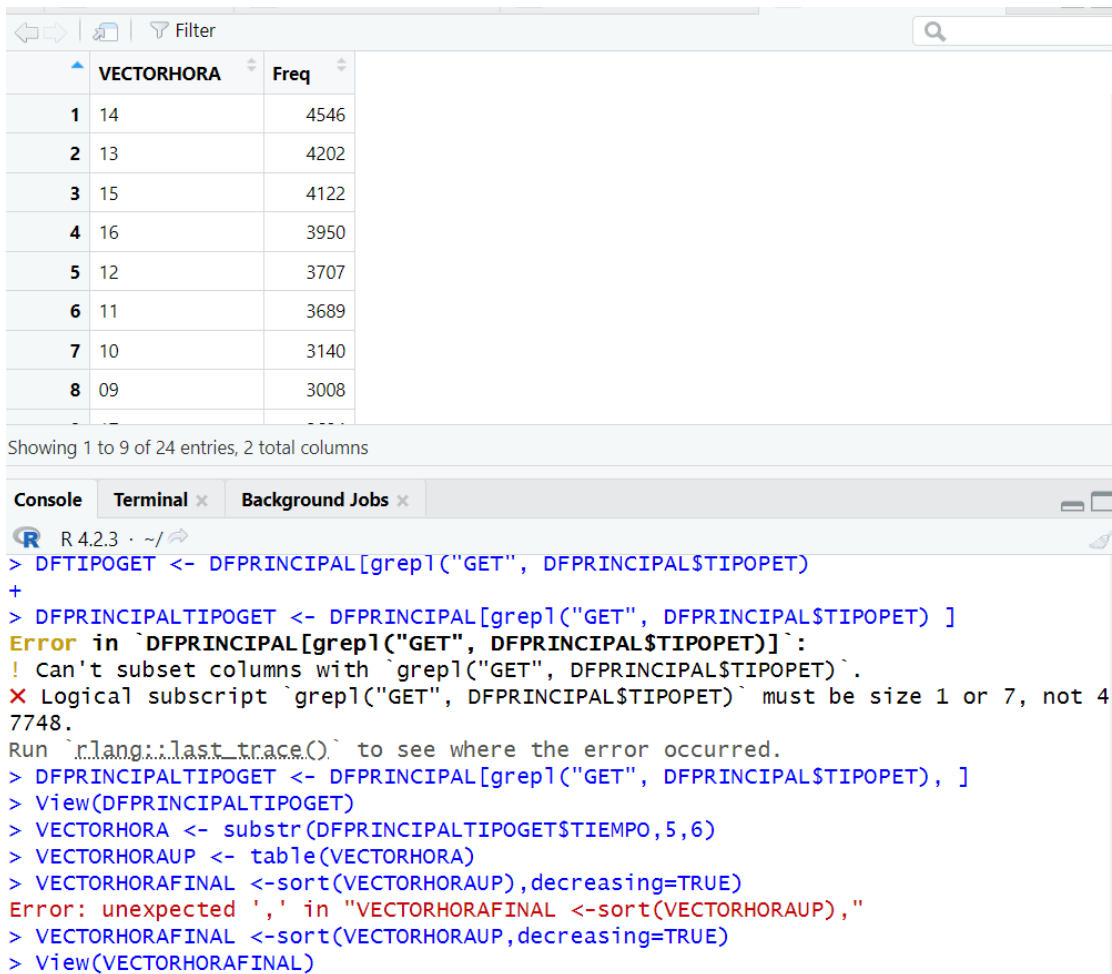
```
- DFPRINCIPALTIPOGET <- DFPRINCIPAL[grepl("GET", DFPRINCIPAL$TIPOPET), ]  
- VECTORHORA <- substr(DFPRINCIPALTIPOGET$TIEMPO,5,6)  
- VECTORHORAUP <- table(VECTORHORA)  
- VECTORHORAFINAL <-sort(VECTORHORAUP,decreasing=TRUE)  
- View(VECTORHORAFINAL)
```

1. Crea una nueva DF llamada DFPRINCIPALTIPOGET, que contiene únicamente las filas de la DF DFPRINCIPAL desde la columna TIPOPET que contiene la cadena de caracteres "GET".
2. Crea un vector llamado VECTORHORA, que contiene las horas de las solicitudes HTTP realizadas mediante GET. Este vector se obtiene extrayendo el 5to y 6to carácter de la columna TIEMPO de la DF DFPRINCIPALTIPOGET.
3. Crea una tabla de frecuencias de las horas de las solicitudes seleccionadas (GET), agrupadas por día. Esto se hace mediante la función `table()`, que



4. cuenta el número de ocurrencias de cada valor único en el vector VECTORHORA. El resultado se almacena en la DF VECTORHORAUP.
5. Ordena la DF de frecuencias VECTORHORAUP de forma descendente, es decir, de mayor a menor, utilizando la función sort(). El resultado se almacena en la DF VECTORHORAFINAL.
6. Finalmente, se visualiza la DF VECTORHORAFINAL mediante la función View(). Que nos muestra la hora de mayores peticiones recibidas

**La mayor cantidad de peticiones recibidas se realizaron a las 14 horas.**



The screenshot shows the R Studio interface. The top pane displays a data frame with two columns: 'VECTORHORA' and 'Freq'. The data is as follows:

	VECTORHORA	Freq
1	14	4546
2	13	4202
3	15	4122
4	16	3950
5	12	3707
6	11	3689
7	10	3140
8	09	3008

Below the data frame, it says 'Showing 1 to 9 of 24 entries, 2 total columns'.

The bottom pane shows the R console with the following code and output:

```

R 4.2.3 ~ /
> DFTIPOGET <- DFPRINCIPAL[grepl("GET", DFPRINCIPAL$TIPOPET)
+
> DFPRINCIPALTIPOGET <- DFPRINCIPAL[grepl("GET", DFPRINCIPAL$TIPOPET) ]
Error in `DFPRINCIPAL[grepl("GET", DFPRINCIPAL$TIPOPET)]`:
! Can't subset columns with `grepl("GET", DFPRINCIPAL$TIPOPET)`.
✖ Logical subscript `grepl("GET", DFPRINCIPAL$TIPOPET)` must be size 1 or 7, not 47748.
Run `rlang::last_trace()` to see where the error occurred.
> DFPRINCIPALTIPOGET <- DFPRINCIPAL[grepl("GET", DFPRINCIPAL$TIPOPET), ]
> View(DFPRINCIPALTIPOGET)
> VECTORHORA <- substr(DFPRINCIPALTIPOGET$TIEMPO,5,6)
> VECTORHORAUP <- table(VECTORHORA)
> VECTORHORAFINAL <-sort(VECTORHORAUP,decreasing=TRUE)
Error: unexpected ',' in "VECTORHORAFINAL <-sort(VECTORHORAUP),"
> VECTORHORAFINAL <-sort(VECTORHORAUP,decreasing=TRUE)
> View(VECTORHORAFINAL)
  
```

## Pregunta 4:

De las peticiones hechas por instituciones educativas (.edu), ¿Cuántos bytes en total se han transmitido, en peticiones de descarga de ficheros de texto ".txt"?

```

- SUBDFPRINCIPAL_EDU <-DFPRINCIPAL[grepl("\\.edu", DFPRINCIPAL$IP), ]
- SUBDFPRINCIPAL_TXT <- SUBDFPRINCIPAL_EDU[grepl("\\.txt", SUBDFPRINCIPAL_EDU$URL), ]
- sum(SUBDFPRINCIPAL_TXT$BYTERESPUESTA, na.rm = TRUE)

```

1. Crea una subDF llamada "SUBDFPRINCIPAL\_EDU" que contiene todas las filas de "SUBDFPRINCIPAL" donde la columna "IP" contiene el texto ".edu".
2. Crea otra subDF llamada "SUBDFPRINCIPAL\_TXT" que contiene todas las filas de "SUBDFPRINCIPAL\_EDU" donde la columna "URL" contiene el texto ".txt".
3. Calcula la suma de todos los valores de la columna "BYTERESPUESTA" de la subDF "SUBDFPRINCIPAL\_TXT" y devuelve el resultado.

```

> SUBDFPRINCIPAL_EDU <-DFPRINCIPAL[grepl("\\.edu", DFPRINCIPAL$IP), ]
> SUBDFPRINCIPAL_TXT <- SUBDFPRINCIPAL_EDU[grepl("\\.txt", SUBDFPRINCIPAL_EDU$URL), ]
> sum(SUBDFPRINCIPAL_TXT$BYTERESPUESTA, na.rm = TRUE)
[1] 2705408
~ |

```

La suma de todos los todos los valores de la columna "BYTERESPUESTA" es: 2 705, 408

## Pregunta 5:

Si separamos la petición en 3 partes (Tipo, URL, Protocolo), usando `str_split` y el separador " " (espacio), ¿cuántas peticiones buscan directamente la URL = "/"?

Cuando se cargó el archivo "epa-http.csv" se utilizó el delimitador "whitespace", con lo cual se tuvo como resultado el siguiente código:

```

library(readr)
epa_http <- read_table("C:/Data Science/epa-http/epa-http.csv")
View(epa_http)

```

Como resultado se obtuvo la información debidamente separada en columnas.

```

DFPRINCIPALURL <- DFPRINCIPAL[grepl("^/$", DFPRINCIPAL$URL), ]
nrow(DFPRINCIPALURL)

```

1. El código filtra el data frame "DFPRINCIPAL" y selecciona las filas donde la columna "URL" contiene exactamente el carácter "/" (la raíz del sitio web).

```
> DFPRINCIPALURL <- DFPRINCIPAL[grep("^/$", DFPRINCIPAL$URL), ]  
> nrow(DFPRINCIPALURL)  
[1] 2382
```

## Pregunta 6:

Aprovechando que hemos separado la petición en 3 partes (Tipo, URL, Protocolo)  
¿Cuántas peticiones NO tienen como protocolo "HTTP/0.2"?

```
DFPRINCIPALHTTP <- DFPRINCIPAL[!grep("HTTP/0.2", DFPRINCIPAL$PROTOCOLO),]  
nrow(DFPRINCIPALHTTP)
```

```
> DFPRINCIPALHTTP <- DFPRINCIPAL[!grep("HTTP/0.2", DFPRINCIPAL$PROTOCOLO),]  
> nrow(DFPRINCIPALHTTP)  
[1] 47747  
>
```

La cantidad de peticiones que NO tienen como protocolo "HTTP/0.2" es: 47,747