Practica_3

Omar Rojas/Edwin Rivera

2023-04-24

PREGUNTA 1 - Apartado 1

Descomprimir el fichero comprimido que contiene los registros del servidor, y a partir de los datos extraídos, cargar en data frame los registros con las peticiones servidas.

## ## ## ## ## ##	IP Length:47748 Class :character Mode :character	Timestamp Length:47748 Class:character Mode:character	Recurso Length:47748 Class :character Mode :character	Cod_retorno Min. :200.0 1st Qu.:200.0 Median :200.0 Mean :227.1 3rd Qu.:200.0 Max. :501.0
##	Tamaño	Metodo_limpio Protocolo_limpio		
##	Min. : 0	GET :46020 HTTF	P/0.2: 1	
##	1st Qu.: 124	HEAD: 106 HTTF	9/1.0:47747	
##	Median: 974	POST: 1622		
## ## ##	Mean : 6531 3rd Qu.: 2788 Max. :4816896			

PREGUNTA 1 Apartado 2

Identificar el número único de usuarios que han interactuado directamente con el servidor de forma segregada según si los usuarios han tenido algún tipo de error en las distintas peticiones ofrecidas por el servidor.

```
## [1] 200 302 304 400 403 404 500 501 ## Levels: 200 302 304 400 403 404 500 501
```

PREGUNTA 1 Apartado 3

Analizar los distintos tipos de peticiones HTTP (GET, POST, PUT, DELETE) gestionadas por el servidor, identificando la frecuencia de cada una de estas. Repetir el análisis, esta vez filtrando previamente aquellas peticiones correspondientes a recursos ofrecidos de tipo imagen.

```
http
frecuencia_http
GET
46020
```

HEAD

106

POST

1622

PREGUNTA 1 Apartado 4

Generar un gráfico que permita visualizar las respuestas del servidor, es decir, la distribución de peticiones según el código de respuesta de esta. Probad distintos tipos de gráficos (por lo menos 2 distintos e incluid estos

Histograma Respuesta de Código 20000 10000 -

304

400

Código de respuesta

403

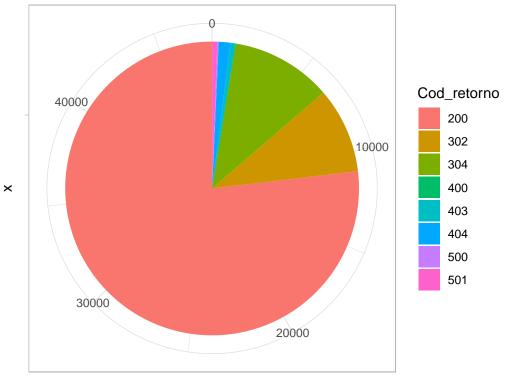
404

500

en el documento RMarkdown).

200

Código de retorno



Freq

PREGUNTA 1 Apartado 5

Utilizando un algoritmo de aprendizaje no supervisado, realizad un análisis de clústering con k-means para los datos del servidor.

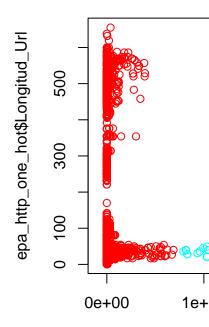
```
## K-means clustering with 2 clusters of sizes 47688, 60
##
## Cluster means:
##
  Cod retorno
           Tamaño Metodo_limpio_GET Metodo_limpio_HEAD
## 1
    227.144
          4676.375
                   0.9637645
                           0.002222781
## 2
    200.000 1480950.217
                   1.0000000
                           0.00000000
##
  Metodo_limpio_POST Protocolo_limpio_HTTP/0.2 Protocolo_limpio_HTTP/1.0
## 1
      0.03401275
                  2.096964e-05
                                  0.999979
## 2
      0.0000000
                   0.000000e+00
                                  1.000000
##
  Longitud_Url
    35.65123
## 1
## 2
    51.93333
##
##
 Clustering vector:
##
   ##
   ##
  ##
##
  ##
```



```
## [47737] 1 1 1 1 1 1 1 1 1 1 1 1 1
## Within cluster sum of squares by cluster:
## [1] 2.135533e+13 3.218883e+13
 (between_SS / total_SS = 70.9 %)
##
## Available components:
##
## [1] "cluster"
       "centers"
           "totss"
                "withinss"
                     "tot.withinss"
## [6] "betweenss"
       "size"
           "iter"
                "ifault"
## K-means clustering with 4 clusters of sizes 1960, 115, 45621, 52
##
## Cluster means:
##
 Cod_retorno
       Tamaño Metodo_limpio_GET Metodo_limpio_HEAD
  200.0000
      46493.648
           0.7163265
                 0.00000000
## 1
                 0.00000000
## 2
  200.0000
     392356.191
           0.9739130
## 3
  228.3739
      2050.448
           0.9743758
                 0.002323491
## 4
                 0.00000000
  200.0000 1578307.769
           1.0000000
 Metodo_limpio_POST Protocolo_limpio_HTTP/0.2 Protocolo_limpio_HTTP/1.0
## 1
    0.28367347
           0.000000e+00
                     1.0000000
## 2
    0.02608696
           0.000000e+00
                     1.0000000
## 3
    0.02330067
           2.191973e-05
                     0.9999781
## 4
    0.00000000
           0.000000e+00
                     1.0000000
##
 Longitud_Url
## 1
  75.79949
## 2
  121.20000
## 3
  33.71055
## 4
  54.57692
##
```

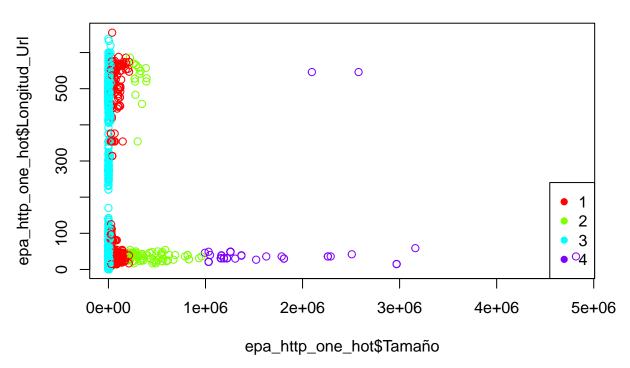
Clustering vector: ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##


```
## [47737] 3 3 3 3 3 3 3 3 3 3 3 3 3
##
## Within cluster sum of squares by cluster:
## [1] 1.979177e+12 3.579821e+12 4.886595e+11 2.846515e+13
## (between_SS / total_SS = 81.3 %)
##
## Available components:
##
## [1] "cluster"
    "totss"
       "tot.withinss"
  "centers"
     "withinss"
## [6] "betweenss"
  "size"
    "iter"
     "ifault"
```



Representad visualmente en gráficos de tipo scatter plot el resultado de vuestros clústering.

Grafico agrupamiento de 4



R Markdown

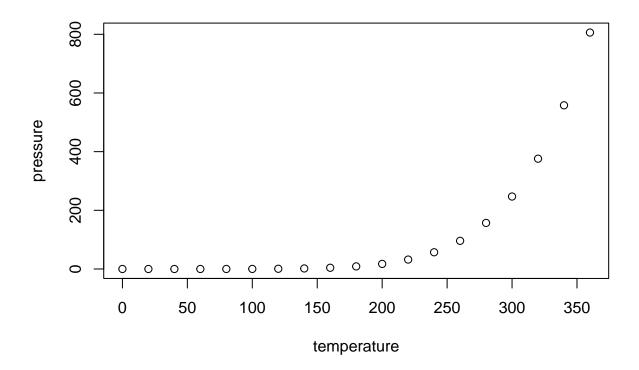
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
##
        speed
                          dist
            : 4.0
                               2.00
##
    Min.
                     Min.
                            :
##
    1st Qu.:12.0
                     1st Qu.: 26.00
##
    Median:15.0
                     Median : 36.00
##
    Mean
            :15.4
                     Mean
                            : 42.98
                     3rd Qu.: 56.00
##
    3rd Qu.:19.0
##
    Max.
            :25.0
                     Max.
                            :120.00
```

Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.