practica3

2023-05-03

Práctica 3

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

##
## Attaching package: 'mltools'

## The following object is masked from 'package:tidyr':
##
##     replace_na
```

Pregunta 1

```
# Importando el archivo
http_data <- read_table("D:\\MAESTRIA-CIBERSEGURIDAD\\SEMINARIO_DATA_SCIENCE\\LABS\\Lab03_lengR\\epa-ht
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_character(),
##   X4 = col_character(),
```

```
##   X5 = col_character(),
##   X6 = col_double(),
##   X7 = col_character()
## )
```

```
# Renonbrando los encabezados de las columnas
colnames(http_data) <- c("Directions", "Date", "Method", "Resource", "Protocol", "Response_code", "Bytes

############Pregunta 1
# Conversiones
http_data$Method <- as.factor(http_data$Method)
http_data$Protocol <- as.factor(http_data$Protocol)
http_data$Response_code <- as.factor(http_data$Response_code)
http_data$Bytes <- as.numeric(http_data$Bytes)
#http_data$Resource <- as.factor(http_data$Resource)
# Conversión de los valores NA por 0

http_data$Bytes <- ifelse(is.na(http_data$Bytes), 0, http_data$Bytes)
nrow(http_data)
```

```
## [1] 47748
```

```
View(http_data)
```

Pregunta 2

```
#########Pregunta 2
# Creando nueva tabla segun las repeticiones de las direcciones, para obtener direcciones únicas
directions_table <- data.frame(Directions = http_data$Directions, Response_code =http_data$Response_code
concurrences <- as.data.frame(table(directions_table))

# Filtrando los valores existentes y ordenando de forma ascendente por la columna Response_code
# 200, 302, 304, 400, 403, 404, 500, 501
directions_data <- filter(concurrences, Freq > 0)


directions_data <- directions_data %>%
  arrange(Response_code)
View(directions_data)


code200_data <- directions_data %>% filter(Response_code == 200)
nrow(code200_data)
```

```
## [1] 2296
```

```
code302_data <- directions_data %>% filter(Response_code == 302)
nrow(code302_data)
```

```
## [1] 970
```

```r
code304_data <- directions_data %>% filter(Response_code == 304)
nrow(code304_data)
```

```
## [1] 505
```

```r
code400_data <- directions_data %>% filter(Response_code == 400)
nrow(code400_data)
```

```
## [1] 1
```

```r
code403_data <- directions_data %>% filter(Response_code == 403)
nrow(code403_data)
```

```
## [1] 5
```

```r
code404_data <- directions_data %>% filter(Response_code == 404)
nrow(code404_data)
```

```
## [1] 152
```

```r
code500_data <- directions_data %>% filter(Response_code == 500)
nrow(code500_data)
```

```
## [1] 29
```

```r
code501_data <- directions_data %>% filter(Response_code == 501)
nrow(code501_data)
```

```
## [1] 11
```

Pregunta 3

```r
####### Pregunta 3

# contar la frecuencia de la columna http
freq_http <- table(http_data$Method)
method_data <- data.frame(http = names(freq_http), freq_http = as.vector(freq_http))
method_data
```

```
##    http freq_http
## 1  "GET     46020
## 2 "HEAD       106
## 3 "POST      1622
```

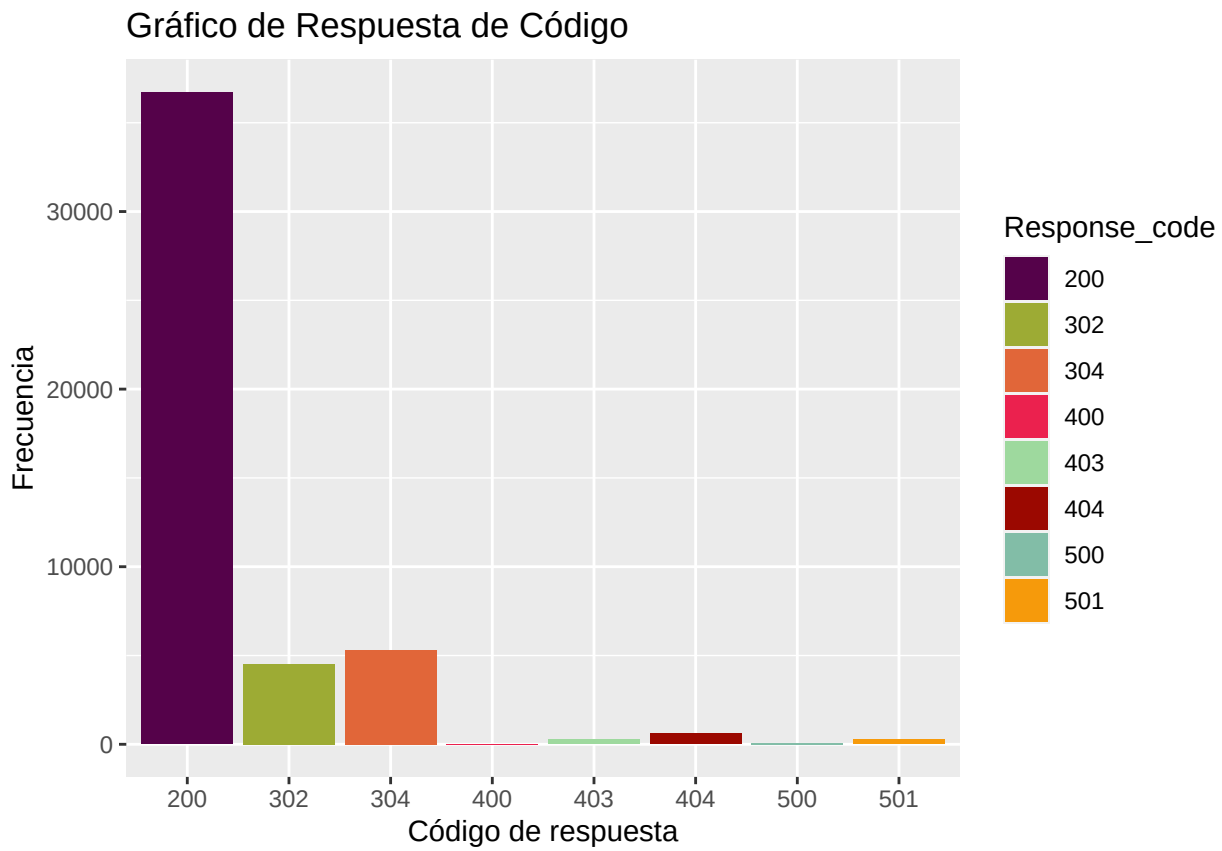
```r
# Hallando la frecuencia de la columna http, filtrando previamente los recursos tipo imagen
different_image_data <- http_data %>%
  filter(!grepl("(?i)\\.(gif|jpg|jpeg|png|bmp)$", Resource))

freq_http2 <- table(different_image_data$Method)
method2_data <- data.frame(http = names(freq_http2), freq_http2 = as.vector(freq_http2))

method_data
```

```
##    http freq_http
## 1  "GET     46020
## 2 "HEAD       106
## 3 "POST      1622
```

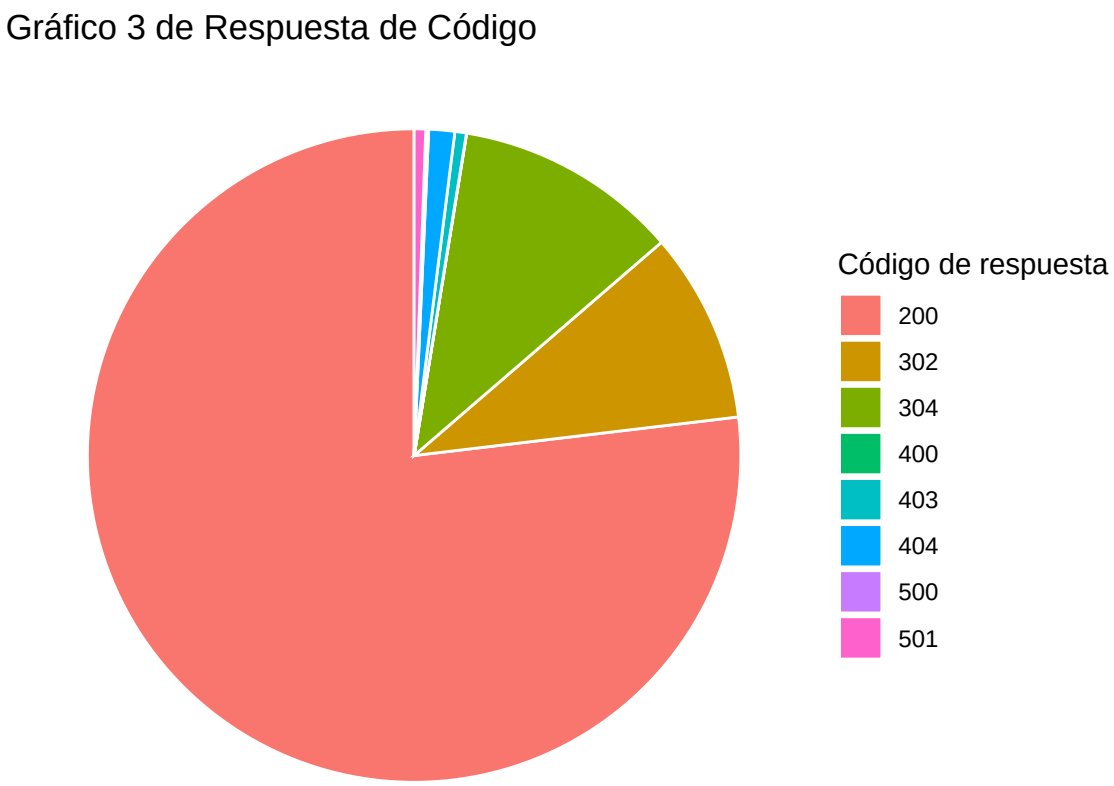
Pregunta 4

Estos tipos de gráficas permiten visualizar la frecuencia de las distintas categorías presentes en una variable, lo que puede ayudar a identificar patrones y tendencias.

```r
### Pregunta 4

tabla_frecuencia <-table(http_data$Response_code)
response_code_table <- data.frame(Response_code = names(tabla_frecuencia),
                          Frecuencia = as.vector(tabla_frecuencia))


ggplot(response_code_table, aes(x = Response_code, y = Frecuencia,  fill = Response_code)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("#55024a", "#9dab34", "#e16639","#eb214e", "#9ed99e","#9b0800","#82bda7"
  labs(title = "Gráfico de Respuesta de Código",
       x = "Código de respuesta",
       y = "Frecuencia")
```




```
ggplot(response_code_table, aes(x = "", y = Frecuencia, fill = Response_code)) +
  geom_bar(stat = "identity", color = "white") +
  coord_polar("y", start = 0) +
  labs(title = "Gráfico 3 de Respuesta de Código",
       fill = "Código de respuesta") +
  theme_void()
```



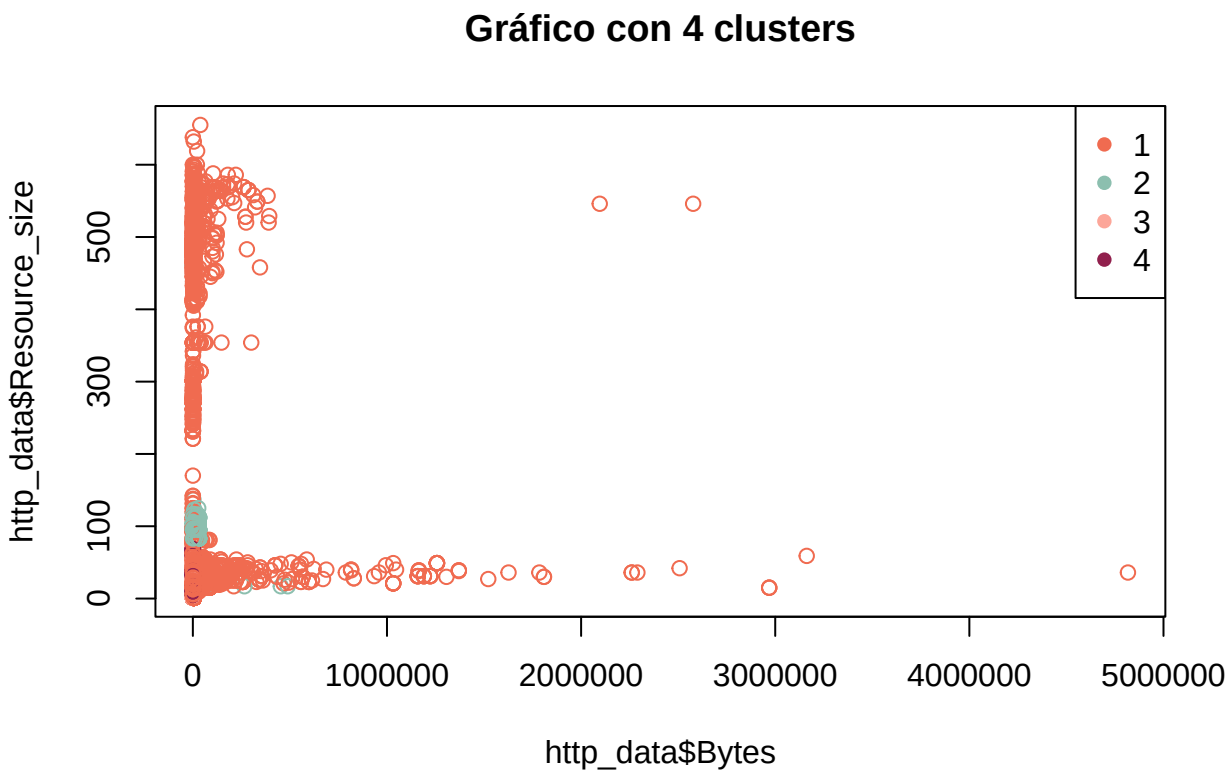
Pregunta 5

```
########Pregunta 5


http_data_filtered <- http_data[, c("Method", "Response_code", "Protocol")]

epa_http_one_hot <- one_hot(as.data.table(http_data_filtered), sparsifyNAs = TRUE)

http_data$Resource_size <- nchar(http_data$Resource)

# Agrupamiento de 4 y 3, esto se puede cambiar por otros valores como 5, 6, 7, etc

results2 <- kmeans(epa_http_one_hot, centers = 4)

results3 <- kmeans(epa_http_one_hot, centers = 3)
```

Pregunta 6

La interpretación de cada gráfica es en base a la cantidad de puntos según el tipo de cluster. Por ejemplo: * En la gráfica 2, se visualiza que existe más presencia de cluster 3 * En el gráfico 3, se aprecia que hay más cantidad de cluster 2

```r
# Gráficas en base a la columna bytes y Resource_size segun el tipo de agrupamiento
# Solo para que los resultados sean reproducibles y no aleatorios
set.seed(123) # si no se coloca la grafica cambia en cada ejecución

## Gráfica con cluster 4
# Solo usar si quieres colores aleatorios
#colores2 <- rainbow(n = length(unique(results2$cluster)))
colores2 <- c("#f06b50", "#8cbfaf", "#fca699", "#91204d")
grap1 <- plot(x = http_data$Bytes, y = http_data$Resource_size, col = colores2[results2$cluster], main=
# solo usar esta opcion si quieren cambiar la escala de notación cientifica a númerica
options(scipen = 999)
# Creando leyenda
legend("topright", legend = levels(factor(results2$cluster)), col = colores2, pch = 16)
```




```r
### Gráfica con Cluster 3

#colores3 <- rainbow(n = length(unique(results3$cluster)))
colores3 <- c("#ad2bad","#00988d", "#dbbf6b")
grap2 <- plot(x = http_data$Bytes, y = http_data$Resource_size, col = colores3[results3$cluster], main=
# solo usar esta opcion si quieren cambiar la escala de notación cientifica a númerica
options(scipen = 999)
# Creando leyenda
legend("topright", legend = levels(factor(results3$cluster)), col = colores3, pch = 16)
```

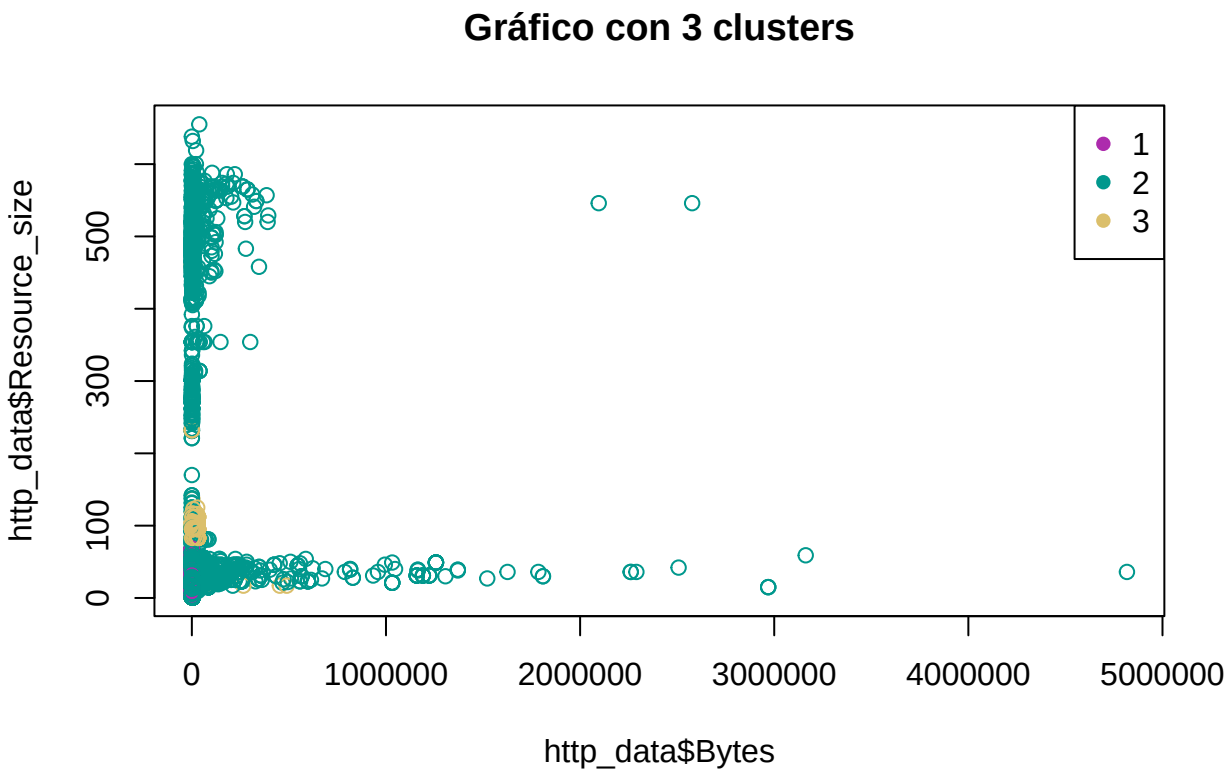


Gráfico con 3 clusters
http_data$Resource_size
0
100
300
500
0
1000000
2000000
3000000
4000000
5000000
http_data$Bytes
1
2
3