practica3

Felipe Vásquez - Marco Ramos

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

```
# Cargando el epa-http.csv
http_datos <- read_table("D:\\epa-http.csv", col_names = FALSE)
```

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

```
##   X3 = col_character(),
##   X4 = col_character(),
##   X5 = col_character(),
##   X6 = col_double(),
##   X7 = col_character()
## )
```

```
# Colocando los nombres adecuados en las columnas
colnames(http_datos) <- c("Direcciones", "Fecha", "Metodo", "Recurso", "Protocolo", "Codigo_Respuesta",
###Pregunta 1
# Tipo de dato
http_datos$Metodo <- as.factor(http_datos$Metodo)
http_datos$Protocolo <- as.factor(http_datos$Protocolo)
http_datos$Codigo_Respuesta <- as.factor(http_datos$Codigo_Respuesta)
http_datos$Bytes <- as.numeric(http_datos$Bytes)
# Reemplazando por 0 los valores NA
http_datos$Bytes <- ifelse(is.na(http_datos$Bytes), 0, http_datos$Bytes)
nrow(http_datos)
```

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

```
View(http_datos)
```

Pregunta 2

```
###Pregunta 2
# Creando nueva tabla segun las repeticiones de las direcciones, para obtener direcciones únicas
Tabla_Direcciones <- data.frame(Direcciones = http_datos$Direcciones, Codigo_Respuesta =http_datos$Codi
concurrences <- as.data.frame(table(Tabla_Direcciones))
# Filtrando valores existentes y ordenando de forma ascendente por la columna Codigo_Respuesta
# 200, 302, 304, 400, 403, 404, 500, 501
Datos_Direcciones <- filter(concurrences, Freq > 0)
Datos_Direcciones <- Datos_Direcciones %>%
  arrange(Codigo_Respuesta)
View(Datos_Direcciones)
codigo200_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 200)
nrow(codigo200_data)
```

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

```
codigo302_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 302)
nrow(codigo302_data)
```

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

```
codigo304_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 304)
nrow(codigo304_data)
```

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

```r
codigo400_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 400)
nrow(codigo400_data)
```

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

```r
codigo403_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 403)
nrow(codigo403_data)
```

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

```r
codigo404_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 404)
nrow(codigo404_data)
```

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

```r
codigo500_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 500)
nrow(codigo500_data)
```

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

```r
codigo501_data <- Datos_Direcciones %>% filter(Codigo_Respuesta == 501)
nrow(codigo501_data)
```

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

Pregunta 3

```r
####### Pregunta 3
# Identificar la frecuencia de la columna método
freq_http <- table(http_datos$Metodo)
metodo_data <- data.frame(http = names(freq_http), freq_http = as.vector(freq_http))
metodo_data
```

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

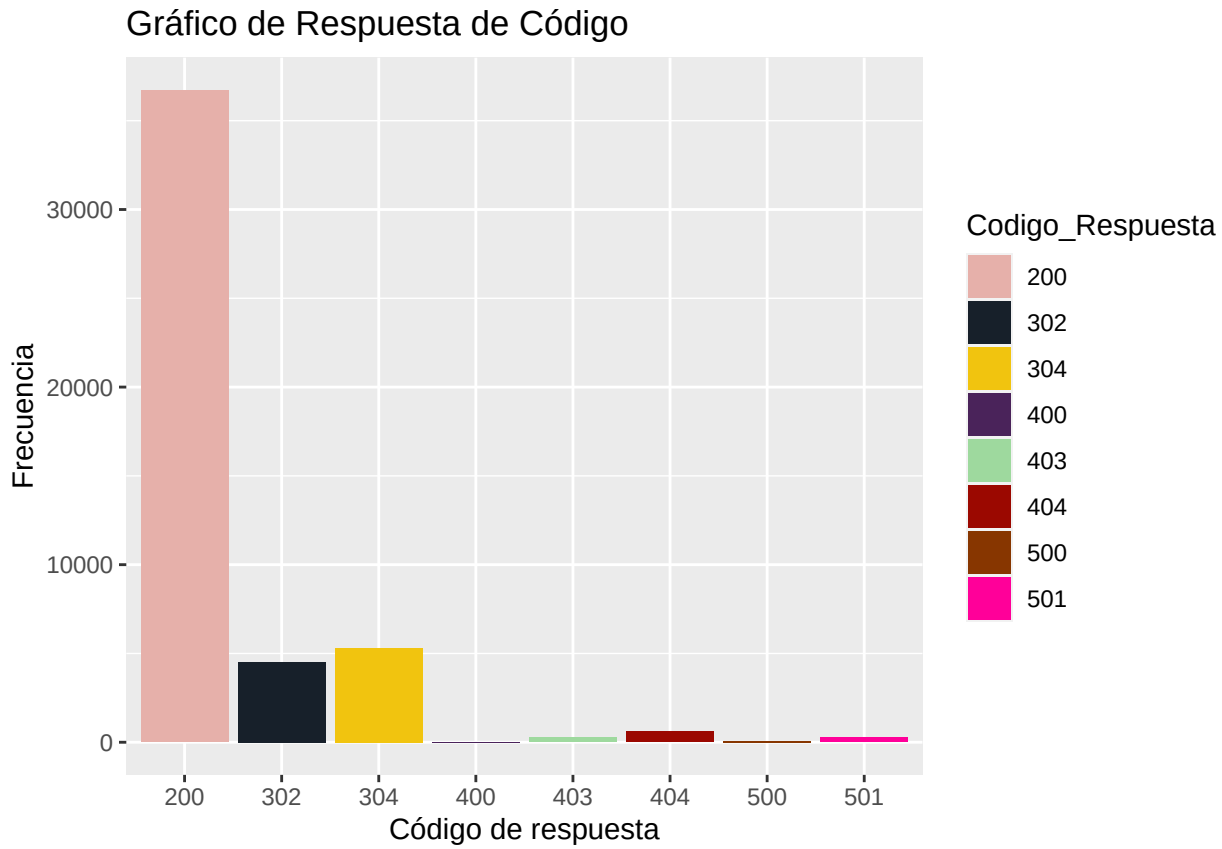
```r
# Encontrando con qué frecuencia aparece la columna "http", luego de filtrar los recursos que son imáge
different_image_data <- http_datos %>%
  filter(!grepl("(?i)\\.(gif|jpg|jpeg|png|bmp)$", Recurso))
freq2_http <- table(different_image_data$Metodo)
metodo2_data <- data.frame(http = names(freq2_http), freq2_http = as.vector(freq2_http))
metodo2_data
```

```
##    http freq2_http
## 1  "GET      23841
## 2 "HEAD         50
## 3 "POST       1416
```

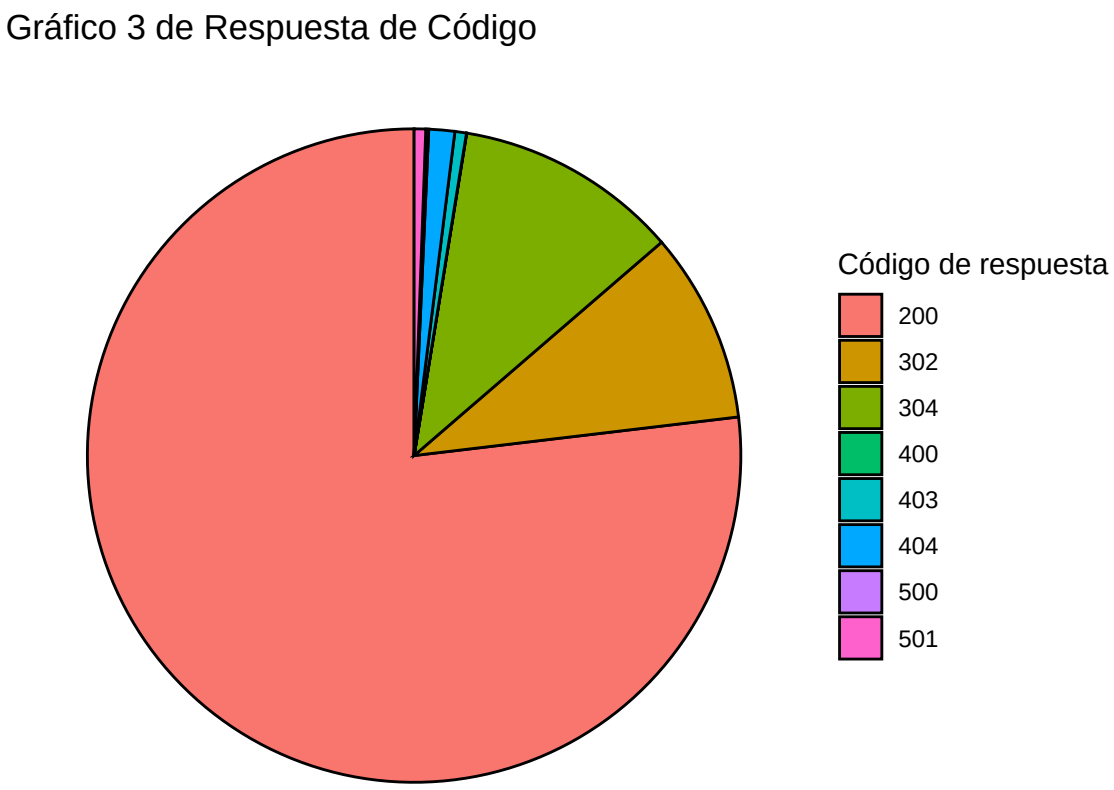
Pregunta 4

Estas clases de gráficos posibilitan la representación visual de la frecuencia de las diferentes categorías que se encuentran en una variable, lo cual puede ser útil para detectar modelos y direcciones

```
### Pregunta 4
Tabla_Frecuencia <-table(http_datos$Codigo_Respuesta)
Tabla_CodigoRespuesta <- data.frame(Codigo_Respuesta = names(Tabla_Frecuencia),
                        Frecuencia = as.vector(Tabla_Frecuencia))
ggplot(Tabla_CodigoRespuesta, aes(x = Codigo_Respuesta, y = Frecuencia,  fill = Codigo_Respuesta)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("#e6b0aa", "#17202a", "#f1c40f","#4a235a", "#9ed99e","#9b0800","#873600"
  labs(title = "Gráfico de Respuesta de Código",
       x = "Código de respuesta",
       y = "Frecuencia")
```




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



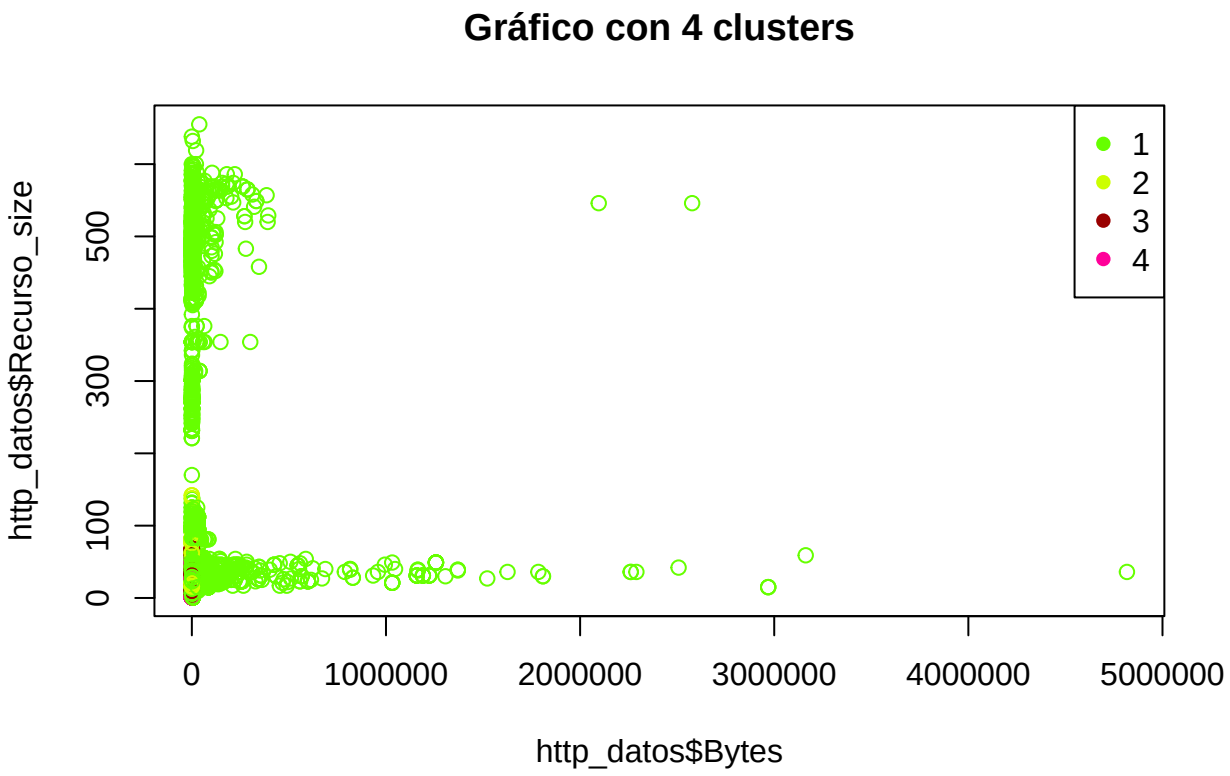
Pregunta 5

```
########Pregunta 5
http_datos_filtrados <- http_datos[, c("Metodo", "Codigo_Respuesta", "Protocolo")]
http_one_hot <- one_hot(as.data.table(http_datos_filtrados), sparsifyNAs = TRUE)
http_datos$Recurso_size <- nchar(http_datos$Recurso)
# Agrupamiento de 4 y 3
resultado2 <- kmeans(http_one_hot, centers = 4)
resultado3 <- kmeans(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

```
# Graficando la columna de bytes y el tamaño del recurso según el tipo de agrupamiento.
set.seed(123)
## Gráfica con cluster 4
colores2 <- c("#66FF00", "#CCFF00", "#990000", "#FF0099")
grafico1 <- plot(x = http_datos$Bytes, y = http_datos$Recurso_size, col = colores2[resultado2$cluster],
options(scipen = 999)
# Creando leyenda
legend("topright", legend = levels(factor(resultado2$cluster)), col = colores2, pch = 16)
```




```
### Gráfica con Cluster 3
#colores3 <- rainbow(n = length(unique(resultado3$cluster)))
colores3 <- c("#873600","#FF0099", "#330000")
grafico2 <- plot(x = http_datos$Bytes, y = http_datos$Recurso_size, col = colores3[resultado3$cluster],
options(scipen = 999)
# Creando leyenda
legend("topright", legend = levels(factor(resultado3$cluster)), col = colores3, pch = 16)
```

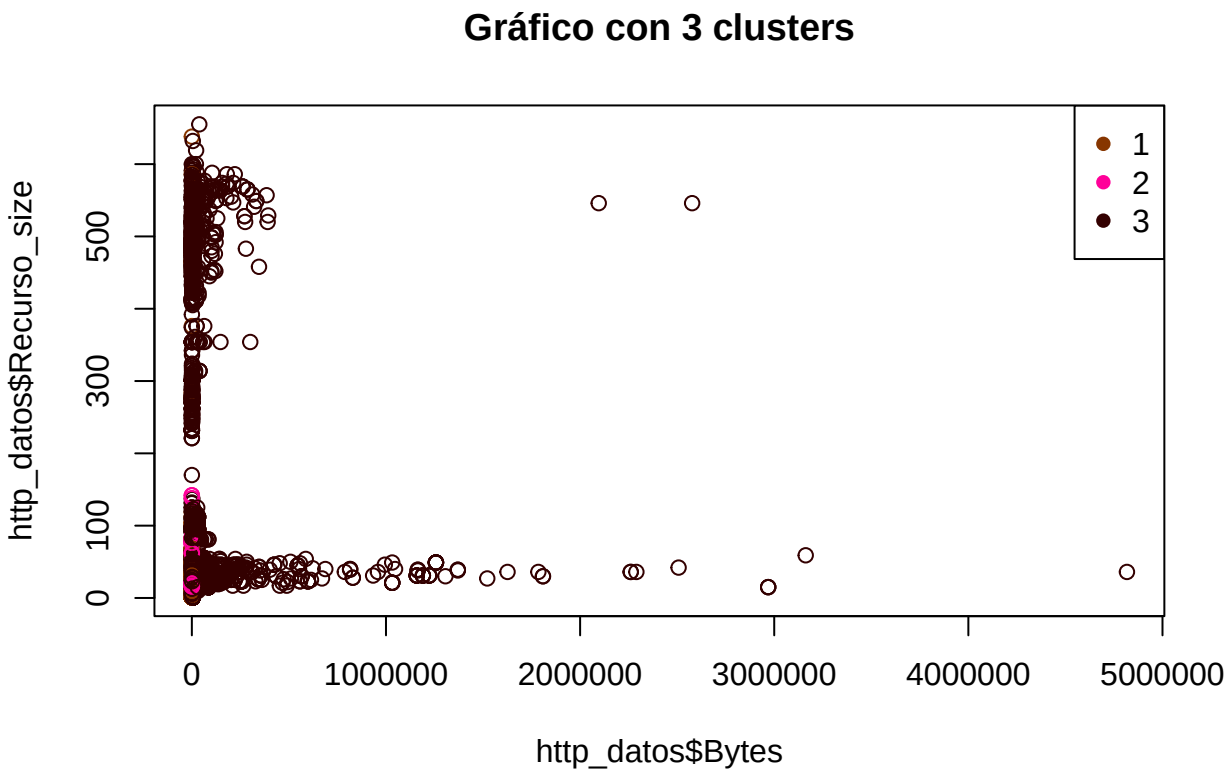


Gráfico con 3 clusters
http_datos$Recurso_size
0
100
300
500
0
1000000
2000000
3000000
4000000
5000000
http_datos$Bytes
1
2
3