Exploración de la base de datos UFO

Amanda Balderas M.

Junio 2015

Objetivo

- Describir el proceso mediante el cual se realizó la obtención y limpieza de los datos de la `base_ufo` .
- Hacer una descripción de la `base_ufo` .
- Presentar un análisis exploratorio de la información que nos permitirá identificar algunas caracteristicas de los datos.

Introducción

Es importante considerar que la información de la `base_ufo` se presenta en una página web,por medio de tablas en formato *html*, por lo que es importante tener un proceso automatizado que nos permita obtener dicha información de manera eficiente, por anterior se utilizó un código que permitió la descarga de la información histórica de manera rápida.

Además, sabemos que para cualquier análisis es importante contar con la información de interés en un formato correcto y con datos limpios, por lo que en este documento se decribe el proceso realizado para obtener la información que se utilizará para el análisis.

En este documento se desarrollan los siguientes puntos:

1. Obtención de los datos
2. Descripción de la base de datos
3. Limpieza y transformación de los datos
4. Análisis exploratorio de los datos
5. Anexo

---

1. Obtención de los datos

La información utilizada para el desarrollo de este proyecto se obtuvo de la página de la organización en los Estados Unidos denominada *The National UFO Reporting Center* NUFORC (http://www.nuforc.org), esta organización se dedica a la recopilación y difusión de avistamientos de OVNIS y/o contactos alienigenas.

La información de los avistamientos proviene de los reportes hechos por las personas a través de diferentes medios: teléfono, fax y sitio web y son almacenados en las tablas que se presentan en la página que son actualizadas continuamente.

La considera los reportes realizados por las personas sobre avistamientos de OVNIS en Estados Unidos, aunque también contiene algunos registros de avistamientos en otros países.

Para poder obtener la información se realizaron los dos procesos siguientes:

Descarga de las tablas mensuales de los eventos

Al realizar la consulta de los datos en la página las tablas se presentan directamente en web en formato "html" por lo que se utilizó las funciones correspondientes para la descarga por medio de *R* y así poder guardar cada una de las tablas en formato de texto (*.txt*).

La opción que se utilizó para obtener la información fue mediante la consulta de las tablas por mes y año, de esta manera se obtienen todos los registros reportados y alemacenados para un mes y año en particular.

National UFO Reporting Center
Monthly Report Index For 04/2015
Click on links for details

NUFORC Home

Date / Time	City	State	Shape	Duration	Summary	Posted
4/30/15 23:20	Martinez	CA	Unknown	10 minutes	Large bright light hovering and making arc like movements.	5/8/15
4/30/15 21:47	Lake Havasu City	AZ	Light	60 seconds	Amber colored orb rotating in the sky.	5/8/15
4/30/15 21:45	Las Vegas	NV	Rectangle	6 minutes	At first I seen a bright green light moving quickly across the sky from left to right. Then it turned red and shot across the sky.	5/8/15
4/30/15 21:40	Montgomery	TX	Circle	still going	Objects not moving. ((NUFORC Note: We suspect a sighting of a celestial body, a star or planet. PD))	5/8/15
4/30/15 21:40	Sycamore	IL	Unknown	10 minutes	We saw seven dull orange near ball-like lights traveling accrost the sky.	5/8/15
4/30/15 19:10	South St. Paul	MN	Teardrop	20 seconds	UFO sighted followed by Aircraft traveling supersonic in the area.	5/8/15
4/30/15 18:34	Tucson	AZ	Unknown	15 minutes	To the west spotted what seemed to be a falling star but as it disappeared into the horizon it excellarated north in a rapid obscure up	5/8/15
4/30/15 03:00	Portland	OR	Light	10 seconds	Star-like white light in SE PDX. ((NUFORC Note: Possible satellite?? PD))	4/30/15

En este proceso se decargaron 865 tablas, correspondientes a los periodos disponibles de junio 1400 a abril de 2015.

Descarga de las descripciones completas de los eventos registrados

Para la descarga de las descripciones completas de cada evento reportado se tuvo que realizar la consulta registro por registro, cada descripción se fue almacenando en una tabla para finalmente obtener la base mensual de descripciones correspondiente. La información mensual de descripciones se guardó en formato de texto (*.txt*).

NUFORC Home Page
Web Report Indexes : by Event Date | by Location | by Shape | by Posting Date
National UFO Reporting Center

Sighting Report

Occurred : 4/30/2015 23:20 (Entered as : 04/30/15 23:20)
Reported: 5/1/2015 2:16:34 AM 02:16
Posted: 5/8/2015
Location: Martinez, CA
Shape: Unknown
Duration:10 minutes

Large bright light hovering and making arc like movements.

At approximately 23:20 hours, myself and two friends noticed a large, bright light. It was the level of the hillside across the road. About 100 yards high. It hovered and moved left and right, forward and back.

The distance of the movements was probably a few feet at a time. These movements happened randomly, and were not rhythmic in any way. When the movements were executed they seemed circular, they did not move in straight line from left to right, but made an arc.

The light was round, bright and white. We could also see a red light that was much smaller and appeared to be behind the white light. At the beginning of the sighting, the object was about a half mile away from us.

After a couple of minutes of watching it, it moved forward toward us to about 1/4 of a mile away. Then it moved back to its original distance. After around 10 minutes it began to move away while keeping the same position in the sky, for the light got smaller and dimmer. It finally moved beyond the crest of the hill and disappeared.

En este proceso se consultaron 97,243 descripciones que se integraron en 865 tablas mensuales, correspondientes a los periodos de junio 1400 a abril de 2015.

Para realizar de manera más rápida la descarga de tablas y descripciones se ejecutó en paralelo el siguiente código:

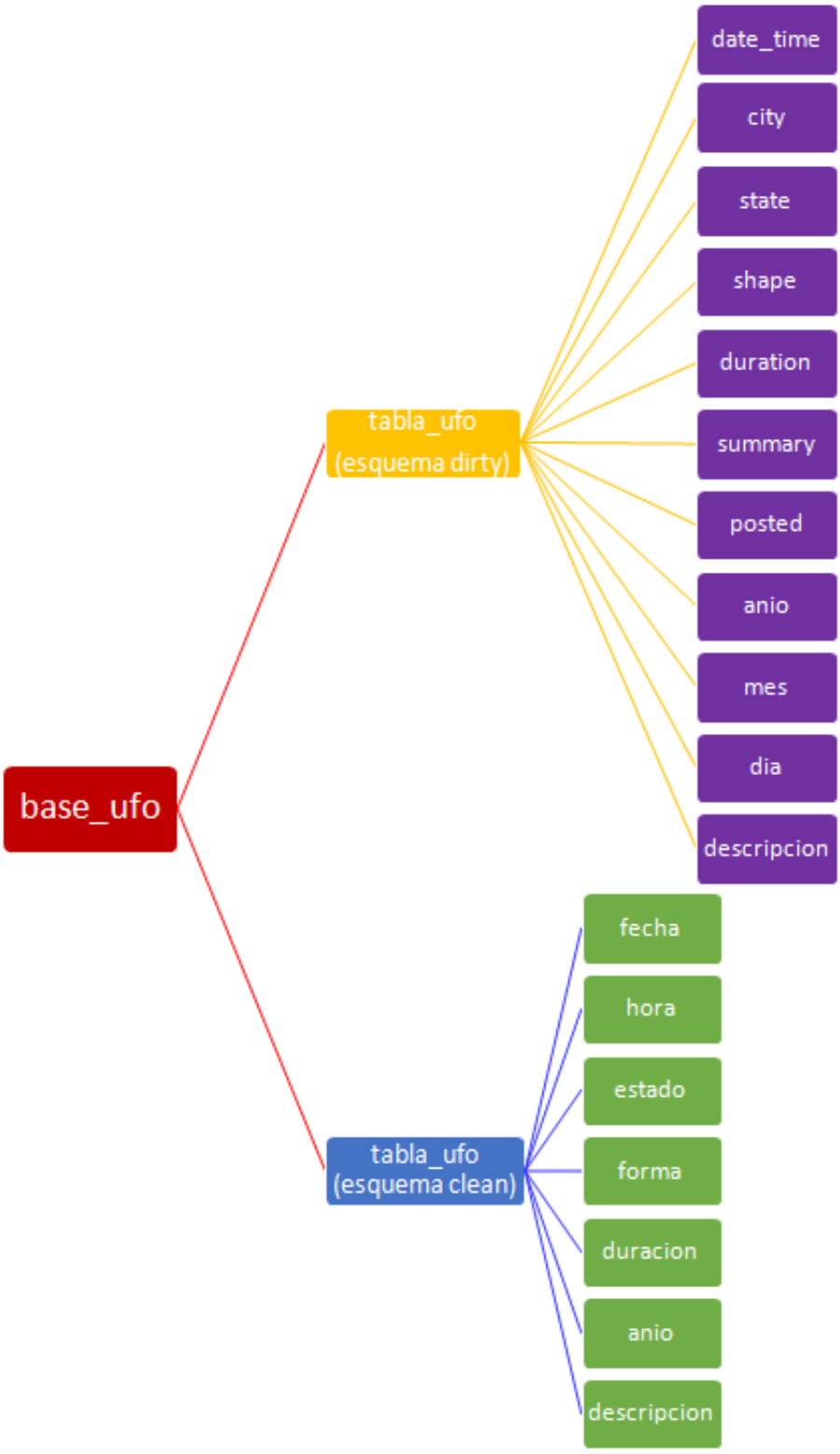
```
# Cargamos librerías
library(rvest)
library(dplyr)

# Definimos url
base_url <- "http://www.nuforc.org/webreports/"

# Obtenemos el índice
ufo_reports_index <- html(paste0(base_url, "ndxevent.html"))

# Obtenemos las URL's de las páginas por día
daily_urls <- paste0(base_url, ufo_reports_index %>%
                       html_nodes(xpath = "//*/td[1]/*/a[contains(@href, 'ndxe')]")  %>%
                       html_attr("href"))
n <- length(daily_urls)

################################################################
################################################################

# Descargamos cada una de las tablas y se guardan en formato "txt"

carpeta1 <- "C:/Users/Amanda29/Documents/archivos_gran_escala/Proyecto_2/UFO/datos_UFO/ta
blas_UFO"

for (i in 1:n){
  table <- daily_urls[i] %>%
    html  %>%
    html_table(fill = TRUE)

  table1 <- data.frame(table)
  anio <- substr(daily_urls[i], 38, 41)
  mes <- substr(daily_urls[i], 42, 43)
  dia <- c()
```

```
  tamanio <- nchar(table1$Date...Time)

  for (j in 1:nrow(table1)){
    if (as.numeric(mes) < 10){
      if (tamanio[j] == 13){
        v_dia <- substr(table1$Date...Time[j],3,4)
      }
      if (tamanio[j] == 12){
        v_dia <- paste0("0", substr(table1$Date...Time[j],3,3))
      }
      if (tamanio[j] == 7){
        v_dia <- substr(table1$Date...Time[j],3,4)
      }
      if (tamanio[j] == 6){
        v_dia <- paste0("0", substr(table1$Date...Time[j],3,3))
      }
    }
    if (as.numeric(mes) > 9){
      if (tamanio[j] == 14){
        v_dia <- substr(table1$Date...Time[j],4,5)
      }
      if (tamanio[j] == 13){
        v_dia <- paste0("0", substr(table1$Date...Time[j],4,4))
      }
      if (tamanio[j] == 8){
        v_dia <- substr(table1$Date...Time[j],4,5)
      }
      if (tamanio[j] == 7){
        v_dia <- paste0("0", substr(table1$Date...Time[j],4,4))
      }
    }
    dia <- c(dia, v_dia)
  }
  table1$anio <- anio
  table1$mes <- mes
  table1$dia <- dia
  nombre <- paste0(mes, "_", anio, ".txt")
  write.table(table1, paste0(carpeta1, "/", nombre), sep = " ")
}

#################################################################
#################################################################

# Descargamos cada una de las descripciones y se guardan por mes en formato "txt"

carpeta2 <- "C:/Users/Amanda29/Documents/archivos_gran_escala/Proyecto_2/UFO/datos_UFO/de
scrip_UFO"

for (i in 1:n){
  reports_url <- paste0(base_url, daily_urls[i] %>%
```

```
                          html %>%
                          html_nodes(xpath = '//*/td[1]/*/a') %>%
                          html_attr('href'))

  anio <- substr(daily_urls[i], 38, 41)
  mes <- substr(daily_urls[i], 42, 43)
  base1 <- data.frame()

  for (j in 1:length(reports_url)){
    try(
      report <- reports_url[j] %>%
        html %>%
        html_nodes(xpath='//*/tr[2]') %>%
        html_text, silent = TRUE)
    if (length(report) == 0){
      base1 <- data.frame(anio = anio, mes = mes, report = "ND", id = j)
    }
    if (length(report) != 0){
      base1 <- data.frame(anio = anio, mes = mes, report = report, id = j)
    }
    if (j == 1){
      base <- base1
    }
    if (j > 1){
      base <- rbind(base, base1)
    }
    report = c()
  }
  nombre <- paste0("descrip_", mes, "_", anio, ".txt")
  write.table(base, paste0(carpeta2, "/", nombre), sep = " ")
}
```

2. Descripción de la base de datos.

La `base_ufo` se conforma de las tablas llamadas `tabla_ufo` que se encuentran en el esquema *dirty* y en el esquema *clean*.



La `tabla_ufo` del esquema *dirty*, que contiene la información original, se forma de 10 variables y un total de 97,243 observaciones, cada observación corresponde un reporte de avistamiento.

De las 10 variables que contiene la tabla, 7 son variables originales que se obtienen directamente en la descarga:

1. `Date...Time`
2. `City`
3. `State`
4. `Shape`

5. `Duration`
6. `Summary`
7. `Posted`

Tenemos también 3 variables que fueron generadas en el proceso de descarga para poder identificar correctamente los avistamientos de acuerdo a la fecha reportada del avistamiento:

8. `anio`
9. `mes`
10. `dia`

En esta tabla también se incorpora la variable que corresponde al texto de las descripciones completas de cada uno de los avistamientos reportados:

11. `descripción`

Verificamos la estructura de la `tabla_ufo` en el esquema *dirty*.

Es importante mencionar que para este ejercicio con la `base_UFO` , se lee de *Postgresql* la base completa ya y se carga a *R* como un DataFrame, que esta aún no es muy grande.

```
# Nos conectamos a base_UFO en Postgresql
drv <- dbDriver("PostgreSQL")
cone <- dbConnect(drv, dbname="base_ufo", host="localhost", port=5432, user="postgres", p
assword="bameam29")

# Verificamos que la tabla existe
dbExistsTable(cone, c("dirty", "tabla_ufo"))
```

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

```
# Leeemos la tabla
d_tabla_ufo <- dbReadTable(cone, c("dirty", "tabla_ufo"))

# Checamos dimensión y estructura
dim(d_tabla_ufo)
```

```
## [1] 97243    11
```

```
str(d_tabla_ufo)
```

```
## 'data.frame':    97243 obs. of  11 variables:
##  $ date_time  : chr  "6/30/00 00:00" "4/14/61 08:00" "2/1/15 03:00" "12/11/62 21:00"
...
##  $ city       : chr  "Myers Spring Canyon" "Nurnburg (Germany)" "Truckee/Cisco Grove"
"Lulworth, Dorsetshire (near) (UK/England)" ...
##  $ state      : chr  "TX" NA "CA" NA ...
##  $ shape      : chr  "Circle" "Cylinder" "Cylinder" NA ...
##  $ duration   : chr  NA "30 nins" "45 seconds" ">1 minute" ...
##  $ summary    : chr  "What I have is a picture of stone age art painted at Myers Sprin
g Canyon that looks like a shaman standing next to a ufo." "I would think that Hanz Glase
r had better things to do than spend months on a wood carving that was too ugly to sell.I
t's a map"| __truncated__ "Between truckee ca and cisco grove ca.cylindrical object with
three dim lights hovering over tree line." "Reported in a London paper in 1762: a bright
light in the form of a straight line eight moon diameters long, one diameter wide."| __tr
uncated__ ...
##  $ posted     : chr  "2/1/07" "8/19/12" "2/20/15" "5/15/06" ...
##  $ anio       : chr  "1400" "1561" "1715" "1762" ...
##  $ mes        : chr  "6" "4" "2" "12" ...
##  $ dia        : chr  "30" "14" "1" "11" ...
##  $ descripcion: chr  "what I have is a picture of stone age art painted at myers sprin
g canyon that looks like a shaman standing next to a ufo.((NUFO"| __truncated__ "I would
think that Hanz Glaser had better things to do than spend months on a wood carving that w
as too ugly to sell.It's a map"| __truncated__ "02/17/15 between truckee ca and cisco gro
ve ca.  cylindrical obeject with three dim lights hovering over tree line.I am a truck"|
__truncated__ "Reported in a London paper in 1762:  A bright light in the form of a strai
ght line eight moon diameters long, one diameter wide"| __truncated__ ...
```

Las variables de la `tabla_ufo` en el esquema *dirty* son:

Nombre de la variable	Descripción de la variable	Tipo	Codificación
date_time	Fecha y hora del avistamiento	Varchar	No aplica
city	Ciudad de los Estados Unidos donde se dio el avistamiento	Varchar	No aplica
state	Estado de los Estados Unidos donde se observó el avistamiento	Varchar	Ver anexo
shape	Forma del objeto que fue observado	Varchar	Ver anexo
duration	Duración del avistamiento	Varchar	No aplica
summary	Resumen con la descripción del evento observado	Text	No aplica
posted	Fecha en la que se dio el registro del avistamiento en la base	Varchar	No aplica
anio	Año del avistamiento	Varchar	1400 - 2015
mes	Mes del avistamiento	Varchar	Ver anexo
dia	Día del avistamiento	Varchar	1 - 31
descripcion	Descripción completa del avistamiento	Text	No aplica

La `tabla_ufo` del esquema *clean*, que contiene la información limpia y transformada, se forma de 9 variables y un total de xxx observaciones, está tabla es la que se utilizará para el análisis de los datos.

Nombre de la variable	Descripción de la variable	Tipo	Codificación
fecha	Fecha del avistamiento	Date	No aplica
hora	Hora del avistamiento	Time	No aplica
estado	Estado de los Estados Unidos donde se observó el avistamiento	Varchar	Ver anexo
ciudad	Ciudad de los Estados Unidos donde se dio el avistamiento	Varchar	No aplica
forma	Forma del objeto que fue observado	Varchar	Ver anexo
duracion	Duración del avistamiento en minutos	Integer	No aplica
anio	Año del avistamiento	Varchar	1400 - 2015
descripción	Descripción completa del avistamiento	Text	No aplica

3. Limpieza y transformación de los datos.

Realizamos la limpieza de cada variable de la `tabla_ufo` del esquema *dirty*, este proceso se realiza apoyando nos de las funciones disponibles en *R*.

```
# Copiamos el dataframe d_tabla_ufo para la limpieza
c_tabla_ufo <- d_tabla_ufo
```

1. `date_time`

Para esta variable homogeneizamos el formato de presentación, se debe considerar que dado el formato original de la variable "*m/d/aa*" es posible confundir fechas que son posteriores al año 2000, por lo anterior se utilizarán las variables `dia` , `mes` y `anio` que fueron creadas durante la descarga de la información, para crear la variable `fecha` correcta.

Dado lo anterior, tenemos que con la variable `date_hime` obtenemos la hora correspondiente para el avistamiento y con las variables con las `dia` , `mes` y `anio` creamos la variable `fecha` y finalmente la variable `date_time` se elimana.

```
# Converimos los vacíos en NA
c_tabla_ufo$date_time[c_tabla_ufo$date_time == ""] <- NA

# Aplicamos el formato de fecha a la variable
c_tabla_ufo$date_time <- as.POSIXct(strptime(c_tabla_ufo$date_time, format = "%m/%d/%y
%H:%M"))

# Con el nuevo formato separamos la hora
c_tabla_ufo$hora <- format(c_tabla_ufo$date_time, "%H:%M")
class(c_tabla_ufo$hora)
```

```
## [1] "character"
```

```
# Creamos la variable fecha considerando las variables: dia, mes y año
c_tabla_ufo$fecha <- as.Date(paste0(c_tabla_ufo$anio, "/", c_tabla_ufo$mes, "/", c_tabl
a_ufo$dia))
class(c_tabla_ufo$fecha)
```

```
## [1] "Date"
```

```
#Eliminamos la variable date_time
c_tabla_ufo$date_time <- NULL
```

2. state

Tenemos que la variable muestra 69 diferentes valores, mientras que el número de estados en Estados Unidos es de 51, entonces considerando la lista de estados de Estados Unidos, conservaremos los registros que corresponden efectivamente a avistamientos de este país, con lo que el número de observaciones queda en 85,120.

```
# Converimos los vacíos en NA
c_tabla_ufo$state[c_tabla_ufo$state == ""] <- NA

# Ponemos todo en mayúsculas
c_tabla_ufo$state <- sapply(c_tabla_ufo$state, function(x) toupper(x))

# Verificamos número de categoirias en la variable
length(unique(c_tabla_ufo$state))
```

```
## [1] 69
```

```
# Cargamos tabla con datos de los estados
estados_usa <- read.table("datos_UFO/estados.csv", header = TRUE, sep = ",")
usa_estados <- which(c_tabla_ufo$state %in% estados_usa$estado)

# Conservamos los registros de Estados Unidos
c_tabla_ufo <- c_tabla_ufo[usa_estados,]
row.names(c_tabla_ufo) <- c(1:nrow(c_tabla_ufo))

# Renombramos la variable y verificamos categorías
colnames(c_tabla_ufo)[2] <- "estado"
c_tabla_ufo$estado <- as.factor(c_tabla_ufo$estado)
class(c_tabla_ufo$estado)
```

```
## [1] "factor"
```

```
length(unique(c_tabla_ufo$estado))
```

```
## [1] 51
```

3. `city`

Podemos ver no existe una codificación y/o formato homogéneo para esta variable, se realiza una limpieza para tratar de tener información lo más homogénea posible.

```
# Converimos los vacíos en NA
c_tabla_ufo$city[c_tabla_ufo$city == ""] <- NA

# Número de registros únicos
length(unique(c_tabla_ufo$city))
```

```
## [1] 17536
```

```
# Covertimos todo a minúsculas
c_tabla_ufo$city <- sapply(c_tabla_ufo$city, function(x) tolower(x))

# Eliminamos los textos que se muestran entre paréntesis
c_tabla_ufo$city  <- sapply(c_tabla_ufo$city, function(x) gsub("\\(.*?\\)", "", x))

# Eliminamos signos de puntuación
c_tabla_ufo$city  <- sapply(c_tabla_ufo$city, function(x) gsub("\\/|[[:punct:]]", " ",
x))

# Eliminamos espacios al inicio o al final del texto
c_tabla_ufo$city  <- sapply(c_tabla_ufo$city, function(x) gsub("(^[[:space:]]+|[[:spac
e:]]+$)", "", x))

# Eliminamos probables espacios dobles y triples
c_tabla_ufo$city  <- sapply(c_tabla_ufo$city, function(x) paste(unlist(strsplit(x, split
= "  ")), collapse = " "))
c_tabla_ufo$city  <- sapply(c_tabla_ufo$city, function(x) paste(unlist(strsplit(x, split
= "  ")), collapse = " "))

# Renombramos la variable y verificamos categorías
colnames(c_tabla_ufo)[1] <- "ciudad"
c_tabla_ufo$ciudad <- as.factor(c_tabla_ufo$ciudad)
class(c_tabla_ufo$ciudad)
```

```
## [1] "factor"
```

```
length(unique(c_tabla_ufo$ciudad))
```

```
## [1] 14698
```

4. shape

Verificamos las categorías para esta variable, encontramos valores con formatos heterogéneos por lo que se realiza una transformación para tratar de tener un formato homogéneo.

```
# Converimos los vacío en NA
c_tabla_ufo$shape[c_tabla_ufo$shape == ""] <- NA

# Número de registros únicos
unique(c_tabla_ufo$shape)
```

```
##  [1] "Circle"    "Cylinder"  "Fireball"  "Light"     "Cross"
##  [6] "Unknown"   NA          "Egg"       "Cigar"     "Sphere"
## [11] "Disk"      "Other"     "Diamond"   "Triangle"  "Oval"
## [16] "Rectangle" "Formation" "Flash"     "Chevron"   "Cone"
## [21] "Changing"  "other"     "Teardrop"  "Delta"     "cylinder"
## [26] "circle"    "rectangle" "light"     "oval"      "triangle"
## [31] "changing"  "fireball"  "Round"     "Dome"      "changed"
## [36] "diamond"   "cigar"     "pyramid"   "sphere"    "flash"
## [41] "Crescent"  "egg"       "unknown"   "delta"     "Flare"
## [46] "Hexagon"
```

```
# Covertimos todo a minúsculas
c_tabla_ufo$shape <- sapply(c_tabla_ufo$shape, function(x) tolower(x))

# Homogeneizamos triangle por triangular
c_tabla_ufo$shape <- sapply(c_tabla_ufo$shape, function(x) gsub("triangle", "triangular",
x))

# Homogeneizamos changing por changed
c_tabla_ufo$shape <- sapply(c_tabla_ufo$shape, function(x) gsub("changing", "changed",
x))

# Renombramos la variable y verificamos categorías
colnames(c_tabla_ufo)[3] <- "forma"
c_tabla_ufo$forma<- as.factor(c_tabla_ufo$forma)
class(c_tabla_ufo$forma)
```

```
## [1] "factor"
```

```
unique(c_tabla_ufo$forma)
```

```
##  [1] circle     cylinder   fireball   light      cross      unknown
##  [7] <NA>       egg        cigar      sphere     disk       other
## [13] diamond    triangular oval       rectangle  formation  flash
## [19] chevron    cone       changed    teardrop   delta      round
## [25] dome       pyramid    crescent   flare      hexagon
## 28 Levels: changed chevron cigar circle cone crescent cross ... unknown
```

5. `duration`

Tenemos que esta variable no tiene un formato homogéneo y se presenta en formato de texto considerando diferentes medidas (segundos, minutos, horas). Haremos la limpieza necesaria para obtener la duración en formato numérico y exclusivamente en segundos.

```
# Convertimos los vacíos en NA
c_tabla_ufo$duration[c_tabla_ufo$duration == ""] <- NA

# Eliminamos caracteres raros.
c_tabla_ufo$duration <- sapply(c_tabla_ufo$duration, function(x) iconv(x, to='ASCII', su
b=""))

# Covertimos todo a minúsculas
c_tabla_ufo$duration <- sapply(c_tabla_ufo$duration, function(x) tolower(x))

# Separamos el texto de la parte númerica
c_tabla_ufo$medida <- sapply(c_tabla_ufo$duration, function(x) gsub("\\d+", "", x))

# Eliminamos caracteres de puntuación
c_tabla_ufo$medida <- sapply(c_tabla_ufo$medida, function(x) gsub("[[:punct:]]", "", x))

# Identificamos textos con las palabras que corresponden a horas, minutos y segundos.
c_tabla_ufo$medida <- sapply(c_tabla_ufo$medida, function(x) gsub("minute|minutes|mins",
"min", x))
c_tabla_ufo$medida <- sapply(c_tabla_ufo$medida, function(x) gsub("seconds|second|secs",
"sec", x))
c_tabla_ufo$medida <- sapply(c_tabla_ufo$medida, function(x) gsub("hours|hour|hrs", "hr",
x))
c_tabla_ufo$medida <- unlist(sapply(c_tabla_ufo$medida, function(x) if(length(grep("hr",
x)) != 0) x <- "hr" else x <- x))
c_tabla_ufo$medida <- unlist(sapply(c_tabla_ufo$medida, function(x) if(length(grep("min",
x)) != 0) x <- "min" else x <- x))
c_tabla_ufo$medida <- unlist(sapply(c_tabla_ufo$medida, function(x) if(length(grep("sec",
x)) != 0) x <- "sec" else x <- x))

# A los casos que no corresponden a las medidas definidas se les asigna NA
c_tabla_ufo$medida[c_tabla_ufo$medida == ""] <- NA
c_tabla_ufo$medida <- sapply(c_tabla_ufo$medida, function(x) if((x != "hr") & (x!="min")
& (x!="sec") & (is.na(x) != TRUE)) x <- NA else x <- x)

# Verificamos que tengamos las categorías de medidas deseadas
unique(c_tabla_ufo$medida)
```

```
## [1] NA    "sec" "min" "hr"
```

```
# Separamos la parte númerica del texto
c_tabla_ufo$dura <- sapply(c_tabla_ufo$duration, function(x) gsub("[a-z]", "", x))

# Eliminamos espacios
c_tabla_ufo$dura <- sapply(c_tabla_ufo$dura, function(x) gsub("[[:space:]]", "", x))

# Seleccionamos los dos primeros caracteres
c_tabla_ufo$dura <- sapply(c_tabla_ufo$dura, function(x) substr(x, 0, 2))

# Eliminamos caracteres de puntuación
c_tabla_ufo$dura <- sapply(c_tabla_ufo$dura, function(x) gsub("[[:punct:]]", "", x))

# Casos sin información se pasan a valores faltantes
c_tabla_ufo$dura[c_tabla_ufo$dura == ""] <- NA

# Modificamos a formato numérico
c_tabla_ufo$dura <- as.numeric(c_tabla_ufo$dura)
class(c_tabla_ufo$dura)
```

```
## [1] "numeric"
```

```
unique(c_tabla_ufo$dura)
```

```
##  [1] NA 45  1 15  5  3  2 20 30 10 51 34  4  8 19 35  6 40 23 57 18  0 90
## [24] 61 60 21  9  7 12 56 78 22 13 24 14 25 68 17 83 50 58 36 31 27 11 72
## [47] 16 81 71 70 28 41 29 32 75 42 80 82 89 52 55 73 65 54 38 53 88 33 91
## [70] 67 46 26 37 43 47 93 39 48 96 85 62 79 59 49 87 44 95
```

```
# Creamos variable duración en segundos
horas <- which(c_tabla_ufo$medida == "hr")
minutos <- which(c_tabla_ufo$medida == "min")
segundos <- which(c_tabla_ufo$medida == "sec")
con_med <- c(horas, minutos, segundos)
nas <- setdiff(c(1:nrow(c_tabla_ufo)), con_med)
c_tabla_ufo$duration[horas] <- 60*60*c_tabla_ufo$dura[horas]
c_tabla_ufo$duration[minutos] <- 60*c_tabla_ufo$dura[minutos]
c_tabla_ufo$duration[segundos] <- c_tabla_ufo$dura[segundos]
c_tabla_ufo$duration[nas] <- NA

# Modificamos a formato numérico el resultado en segundos
c_tabla_ufo$duration <- round(as.numeric(c_tabla_ufo$duration),0)
class(c_tabla_ufo$duration)
```

```
## [1] "numeric"
```

```
# Los casos con cero se pasan a NA
c_tabla_ufo$duration[c_tabla_ufo$duration == 0] <- NA

# Eliminamos la variable medida y dura
c_tabla_ufo$medida <- NULL
c_tabla_ufo$dura <- NULL

# Renombramos la variable y verificamos categorías
colnames(c_tabla_ufo)[4] <- "duracion"
```

6. `summary`

Esta variable corresponde al resumen de la descripción del evento, dado que contamos con las descripciones completas esta variable es redundante, por lo que se elimina.

```
# Eliminando la variable
c_tabla_ufo$summary <- NULL
```

7. `posted`

No se considera que esta variable sea importante dado que nos indica la fecha en la que se dio el registro del evento, lo cual no aporta información interesante para el análisis.

```
# Eliminando la variable
c_tabla_ufo$posted <- NULL
```

8. `anio`

Esta variable se creo durante la descarga de la información para poder contar con el registro correcto de la fecha del avistamiento y es utilizada para generar la variable `fecha` de la `tabla_ufo` en el esquema *clean*. Vamos a conservar esta variable ya que se utilizará como parámetro para la carga, dado que difinimos nuestra base con tablas particionadas por año.

```
# cambiamos el tipo de la variable
c_tabla_ufo$anio <- as.factor(c_tabla_ufo$anio)
class(c_tabla_ufo$anio)
```

```
## [1] "factor"
```

9. `mes`

Esta variable al igual que la variable `anio` , se creo durante la descarga de la información para poder contar con el registro correcto de la fecha del avistamiento y es utilizada para generar la variable `fecha` de la `tabla_ufo` en el esquema *clean*. Por lo anterior ya no se considera en la `tabla_ufo` del esquema *clean*.

```
# Eliminando la variable
c_tabla_ufo$mes <- NULL
```

10. `dia`

Esta variable al igual que la variable `anio` y `mes`, se creo durante la descarga de la información para poder contar con el registro correcto de la fecha del avistamiento y es utilizada para generar la variable `fecha` de la `tabla_ufo` en el esquema *clean*. Por lo anterior ya no se considera en la `tabla_ufo` del esquema *clean*.

```
# Eliminando la variable
c_tabla_ufo$dia <- NULL
```

11. `descripcion`

Realizamos una limpieza general al texto de las descripciones, debemos recordar que durante la descarga, aquellos casos en los que no había disponible una descripción se asignó “ND”, por lo que estos casos se converitiran en valores faltantes.

```
# Convertimos los vacíos en NA
c_tabla_ufo$descripcion[c_tabla_ufo$descripcion == ""] <- NA

# Casos con "ND" se convierten en NA
c_tabla_ufo$descripcion[c_tabla_ufo$descripcion == "ND"] <- NA

# Convertimos todo a minúsculas
c_tabla_ufo$descripcion <- sapply(c_tabla_ufo$descripcion, function(x) tolower(x))

# Eliminamos todos aquellos caracteres raros que no formen parte del abecedario.
c_tabla_ufo$descripcion <- sapply(c_tabla_ufo$descripcion, function(x) iconv(x, to='ASCI
I', sub=""))

# Eliminamos los signos de puntuación.
c_tabla_ufo$descripcion <- sapply(c_tabla_ufo$descripcion, function(x) gsub("[[:punc
t:]]", "", x))

# Eliminamos los caracteres númericos.
c_tabla_ufo$descripcion <- sapply(c_tabla_ufo$descripcion, function(x) gsub("[[:digi
t:]]", "", x))

# Eliminamos algunos caracteres sin sentido
c_tabla_ufo$descripcion <- sapply(c_tabla_ufo$descripcion, function(x) gsub("\t|\r|\n",
"", x))

# Eliminamos los espacios en blanco al inicio y/o final del texto.
c_tabla_ufo$descripcion <- sapply(c_tabla_ufo$descripcion, function(x) gsub("(^[[:spac
e:]]+|[[:space:]]+$)", "", x))

# Eliminamos espacios dobles y/o triples
c_tabla_ufo$descripcion  <- sapply(c_tabla_ufo$descripcion, function(x) paste(unlist(strs
plit(x, split = "  ")), collapse = " "))
c_tabla_ufo$descripcion  <- sapply(c_tabla_ufo$descripcion, function(x) paste(unlist(strs
plit(x, split = "  ")), collapse = " "))

# Asignamos NA a los registros que quedaron sin texto
c_tabla_ufo$descripcion[c_tabla_ufo$descripcion == ""] <- NA
```

Tenemos que despúes de la revisión de cada variable terminamos con 8 variables y 85,120 observaciones.

```
# Reordenando las variables seleccionadas
variables <- c("fecha", "hora", "estado", "ciudad", "forma", "duracion", "anio", "descrip
cion")
c_tabla_ufo <- c_tabla_ufo[, variables]
dim(c_tabla_ufo)
```

```
## [1] 85120     8
```

** Valores faltantes**

Verificaremos el número de valores faltantes para cada variable.

```
faltantes <- lapply(c_tabla_ufo, function(x) sum(is.na(x)))
faltantes
```

```
## $fecha
## [1] 0
##
## $hora
## [1] 1009
##
## $estado
## [1] 0
##
## $ciudad
## [1] 0
##
## $forma
## [1] 2721
##
## $duracion
## [1] 11759
##
## $anio
## [1] 0
##
## $descripcion
## [1] 1
```

Checamos el número de observaciones incompletas (con valores faltantes en alguna de las variables).

```
sum(!complete.cases(c_tabla_ufo))
```

```
## [1] 14013
```

dado que tenemos una importante cantidad de registros con algún valor faltante y para tratar de conservar la mayor cantidad de información haremos imputaciones para los valores faltantes en cada una de las variables que lo requiera.

- `hora`

Vamos a imputar la hora reportada más frecuentemente.

```
horas <- data.frame(table(c_tabla_ufo$hora))
names(horas) <- c("hora", "frecuencia")
horas <- horas[with(horas, order(-frecuencia)),]
horas[1:5,]
```

```
##       hora frecuencia
## 1283 22:00       4963
## 1223 21:00       4828
## 1343 23:00       3671
## 1163 20:00       3438
## 1253 21:30       2534
```

```
hora_na <- is.na(c_tabla_ufo$hora)
c_tabla_ufo$hora[hora_na] <- as.character(horas$hora[1])
sum(is.na(c_tabla_ufo$hora))
```

```
## [1] 0
```

- `forma`

Dado que en las categorías de la variable se tiene la opción *unknown*, vamos a asignar esa categoría a los valores faltantes.

```
forma_na <- is.na(c_tabla_ufo$forma)
c_tabla_ufo$forma[forma_na] <- "unknown"
sum(is.na(c_tabla_ufo$forma))
```

```
## [1] 0
```

- `duracion`

Vamos a imputar la duración promedio de los avistamientos.

```
durac_prom <- mean(c_tabla_ufo$duracion, na.rm = TRUE)
durac_na <- is.na(c_tabla_ufo$duracion)
c_tabla_ufo$duracion[durac_na] <- round(durac_prom,0)
sum(is.na(c_tabla_ufo$duracion))
```

```
## [1] 0
```

Verificamos que ya no hay valores faltantes en ninguna de las variables diferentes a la variable `descripcion` .

```
faltantes <- lapply(c_tabla_ufo, function(x) sum(is.na(x)))
faltantes
```

```
## $fecha
## [1] 0
##
## $hora
## [1] 0
##
## $estado
## [1] 0
##
## $ciudad
## [1] 0
##
## $forma
## [1] 0
##
## $duracion
## [1] 0
##
## $anio
## [1] 0
##
## $descripcion
## [1] 1
```

Finalmente, verificamos estructura de la información que se obtuvo después del proceso de limpieza y transformación.

```
# Checamos dimensión y estructura
str(c_tabla_ufo)
```

```
## 'data.frame':    85120 obs. of  8 variables:
##  $ fecha      : Date, format: "1400-06-30" "1715-02-01" ...
##  $ hora       : chr  "00:00" "03:00" "21:00" "20:00" ...
##  $ estado     : Factor w/ 51 levels "AK","AL","AR",..: 44 5 35 19 28 35 11 27 51 48
...
##  $ ciudad     : Factor w/ 14698 levels "","1 25 corridor",..: 8754 13291 1858 712 2185
9020 1989 1672 14628 869 ...
##  $ forma      : Factor w/ 28 levels "changed","chevron",..: 4 8 14 19 14 7 28 28 13 14
...
##  $ duracion   : num  1156 45 60 15 1156 ...
##  $ anio       : Factor w/ 103 levels "1400","1715",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ descripcion: chr  "what i have is a picture of stone age art painted at myers sprin
g canyon that looks like a shaman standing next to a ufonuforc "| __truncated__ "between
truckee ca and cisco grove ca cylindrical obeject with three dim lights hovering over tre
e linei am a truck driver i wa"| __truncated__ "slow moving fireball stench of burning su
lpherexcerpt from brief sketch of the first settlement of the county of schoharie ny b"|
__truncated__ "ufo report communicated by thomas jeffersonfrom httpwwwufodigestcomarticle
reportconcerningunidentifiedflyingobjectcommunicatedt"| __truncated__ ...
```

Cargamos la información limpia a nuestra `tabla_ufo` en el esquema *clean*.

```
dbWriteTable(cone, c("clean", "tabla_ufo"), value = c_tabla_ufo, append = TRUE, row.names
= FALSE)
```

Verificamos en "*psql*" que la información se cargó completa.

```
SELECT COUNT (*) FROM clean.tabla_ufo;
```

Nos desconectamos de *Postgresql*.

```
dbDisconnect(cone)
dbUnloadDriver(drv)
```

---

4. Análisis exploratorio de los datos.

Para el análisis exploratorio utilizamos la información que se cargo a la `tabla_ufo` del esquema *clean*, que ya contiene la información que se obtuvo despúes del proceso de limpieza y transformación.

Sumario Estadístico

Obtenemos un resumen estadístico de los datos.

```
summary(c_tabla_ufo)
```

```
##      fecha                hora              estado
##  Min.   :1400-06-30   Length:85120       CA     :11222
##  1st Qu.:2001-12-16   Class :character   FL     : 5142
##  Median :2007-11-15   Mode  :character   WA     : 5019
##  Mean   :2005-02-17                      TX     : 4373
##  3rd Qu.:2012-06-16                      NY     : 3846
##  Max.   :2015-04-30                      AZ     : 3243
##                                          (Other):52275
##            ciudad             forma          duracion           anio
##  seattle      :  652   light     :17265   Min.   :     1   2014   : 7944
##  new york city:  629   unknown   : 8816   1st Qu.:    60   2012   : 7316
##  phoenix      :  561   circle    : 8265   Median :   300   2013   : 7053
##  las vegas    :  470   triangular: 8235   Mean   :  1156   2011   : 5109
##  portland     :  464   fireball  : 6716   3rd Qu.:  1156   2008   : 4668
##  los angeles  :  424   other     : 5800   Max.   :298800   2009   : 4266
##  (Other)      :81920   (Other)   :30023                    (Other):48764
##  descripcion
##  Length:85120
##  Class :character
##  Mode  :character
##
##
##
##
```

Destacan los siguientes resultados:

- El estado con mayor número de avistamientos es *California*.
- La ciudad con mayor número de avistamientos es *Seattle*.
- La forma más observada ha sido *light*.
- La duración promedio de los avistamientos es de 1146 segundos, que equivale a un poco más de 19 minutos.
- El año con el mayor número de reportes de avistamientos ha sido el 2014.

Algunas frecuencias

Ahora verifiquemos algunos otros casos de frecuencias:

- Se puede observar que la fecha con mayor número de avistamientos registrados ha sido el 4 de julio de 2014.

```
fechas <- data.frame(table(c_tabla_ufo$fecha))
names(fechas) <- c("fecha", "frecuencia")
fechas <- fechas[with(fechas, order(-frecuencia)),]
head(fechas)[1:5,]
```

```
##            fecha frecuencia
## 10569 2014-07-04        266
## 9108  2010-07-04        204
## 9839  2012-07-04        190
## 5241  1999-11-16        187
## 10204 2013-07-04        183
```

- Tenemos que la hora en la que se registran más avistamientos es a las 22:00 hrs., es decir a las 10 de la noche.

```
horas <- data.frame(table(c_tabla_ufo$hora))
names(horas) <- c("hora", "frecuencia")
horas <- horas[with(horas, order(-frecuencia)),]
head(horas)[1:5,]
```

```
##       hora frecuencia
## 1283 22:00       5972
## 1223 21:00       4828
## 1343 23:00       3671
## 1163 20:00       3438
## 1253 21:30       2534
```

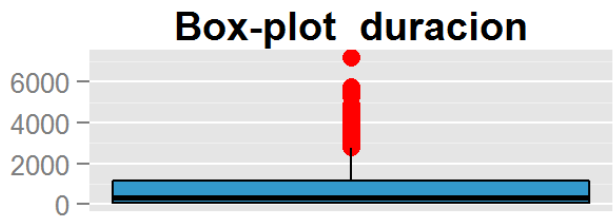
Análisis gráfico

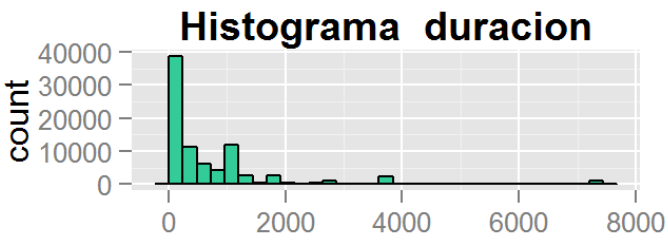
Ahora haremos un análisis gráfico de la variable `duración` , que en este caso es nuestra única variable numérica, vamos a considerar las observaciones con duración menor a 10,000 segundos.

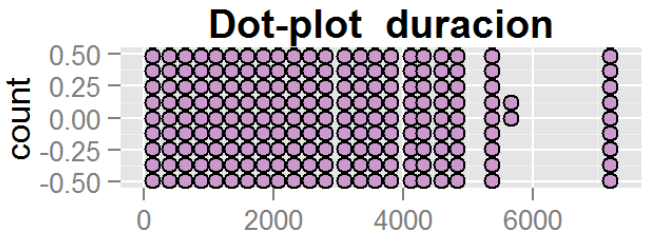
```
# Obtenemos gráficas de la variable duracion
base <- subset(c_tabla_ufo, duracion <= 10000)
x1 <- 6

  base$id <- c(1:nrow(base))
  grafica_0a <- ggplot(base, aes(x = base$id, y = base[,x1]))
  grafica_0b <- ggplot(base, aes(x = base[,x1]))

  # Box-plot
  grafica_1 <- grafica_0a +
  geom_boxplot(fill = '#3399CC', colour = 'black', outlier.colour = 'red', outlier.size =
3) +
  ggtitle(paste('Box-plot ', names(base)[x1])) +
  scale_y_continuous(name = '') +
  scale_x_continuous(name = '', breaks = NULL) +
  theme(plot.title = element_text(lineheight = .8, face = 'bold'))

  # Histograma
  grafica_2 <- grafica_0b +
  geom_histogram(fill = '#33CC99', colour = 'black') +
  ggtitle(paste('Histograma ', names(base)[x1])) +
```

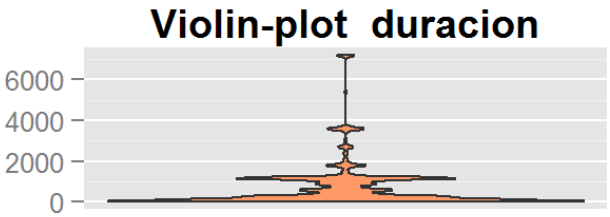
```
scale_x_continuous(name = '') +
theme(plot.title = element_text(lineheight = .8, face = 'bold'))

# Dot-plot
grafica_3 <- grafica_0b +
geom_dotplot(stackdir = 'centerwhole', fill = '#CC99CC') +
ggtitle(paste('Dot-plot ', names(base)[x1])) +
scale_x_continuous(name = '') +
theme(plot.title = element_text(lineheight = .8, face = 'bold'))

# Violin-plot
grafica_4 <- grafica_0a +
geom_violin(fill = '#FF9966') +
ggtitle(paste('Violin-plot ', names(base)[x1])) +
scale_y_continuous(name = '') +
scale_x_continuous(name = '', breaks = NULL) +
theme(plot.title = element_text(lineheight = .8, face = 'bold'))

# Densidad
grafica_5 <- grafica_0b +
  geom_histogram(aes(y = ..density..), fill = '#FFFFCC', colour = 'black') +
  geom_density(color = 'red') +
  ggtitle(paste('Densidad ', names(base)[x1])) +
  scale_x_continuous(name = '') +
  theme(plot.title = element_text(lineheight = .8, face = 'bold'))

# QQ-plot

# Variables que nos serviran para la qqline
yy <- quantile(base[,x1][!is.na(base[,x1])], c(0.25, 0.75))
xx <- qnorm(c(0.25, 0.75))
slope <- diff(yy) / diff(xx)
int <- yy[1L] - slope * xx[1L]

# Generamos la gráfica qqnorm y qqline
grafica_6 <- ggplot(base, aes(sample = base[,x1])) +
  ggtitle(paste('QQ-plot ', names(base)[x1])) +
  stat_qq(shape = 1, size = 4) +
  geom_abline(slope = slope, intercept = int, colour = 'red', size = 1) +
  theme(plot.title = element_text(lineheight = .8, face = 'bold'))

# Presentamos las gráficas en la misma pantalla
grid.newpage()
pushViewport(viewport(layout = grid.layout(3, 2)))
vplayout <- function(x, y) viewport(layout.pos.row = x, layout.pos.col = y)
print(grafica_1, vp = vplayout(1, 1))
print(grafica_2, vp = vplayout(1, 2))
print(grafica_3, vp = vplayout(2, 1))
print(grafica_4, vp = vplayout(2, 2))
print(grafica_5, vp = vplayout(3, 1))
```

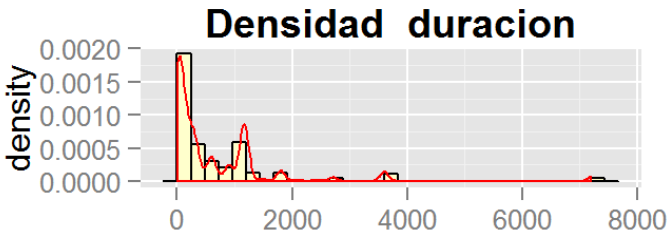
```
  print(grafica_6, vp = vplayout(3, 2))
```

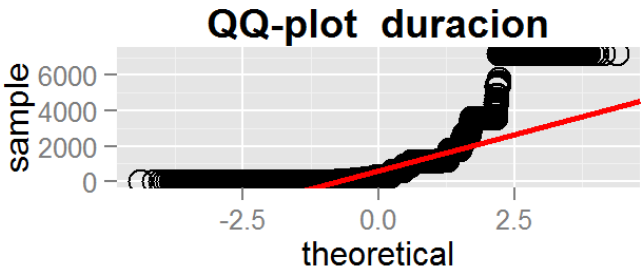






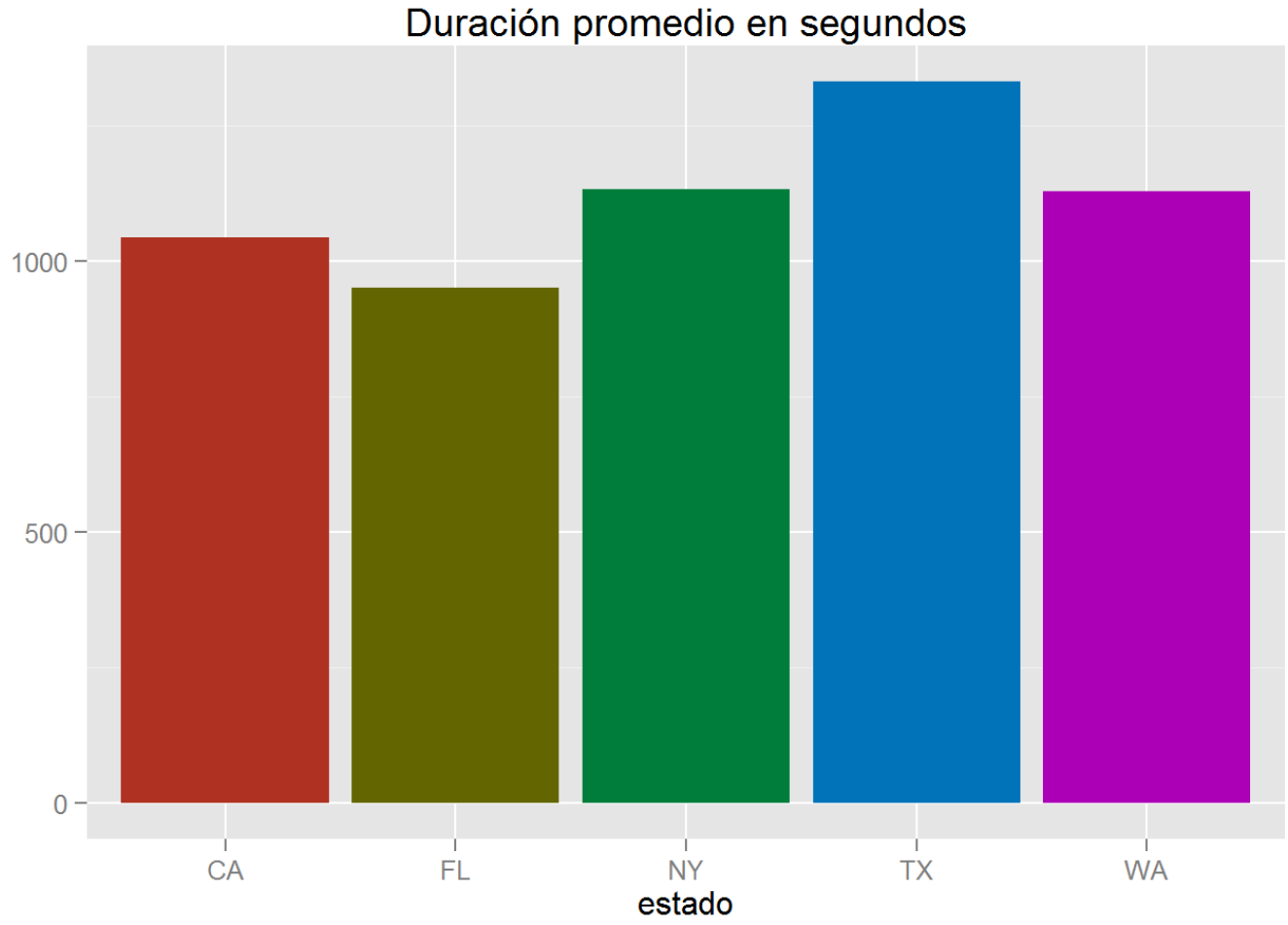






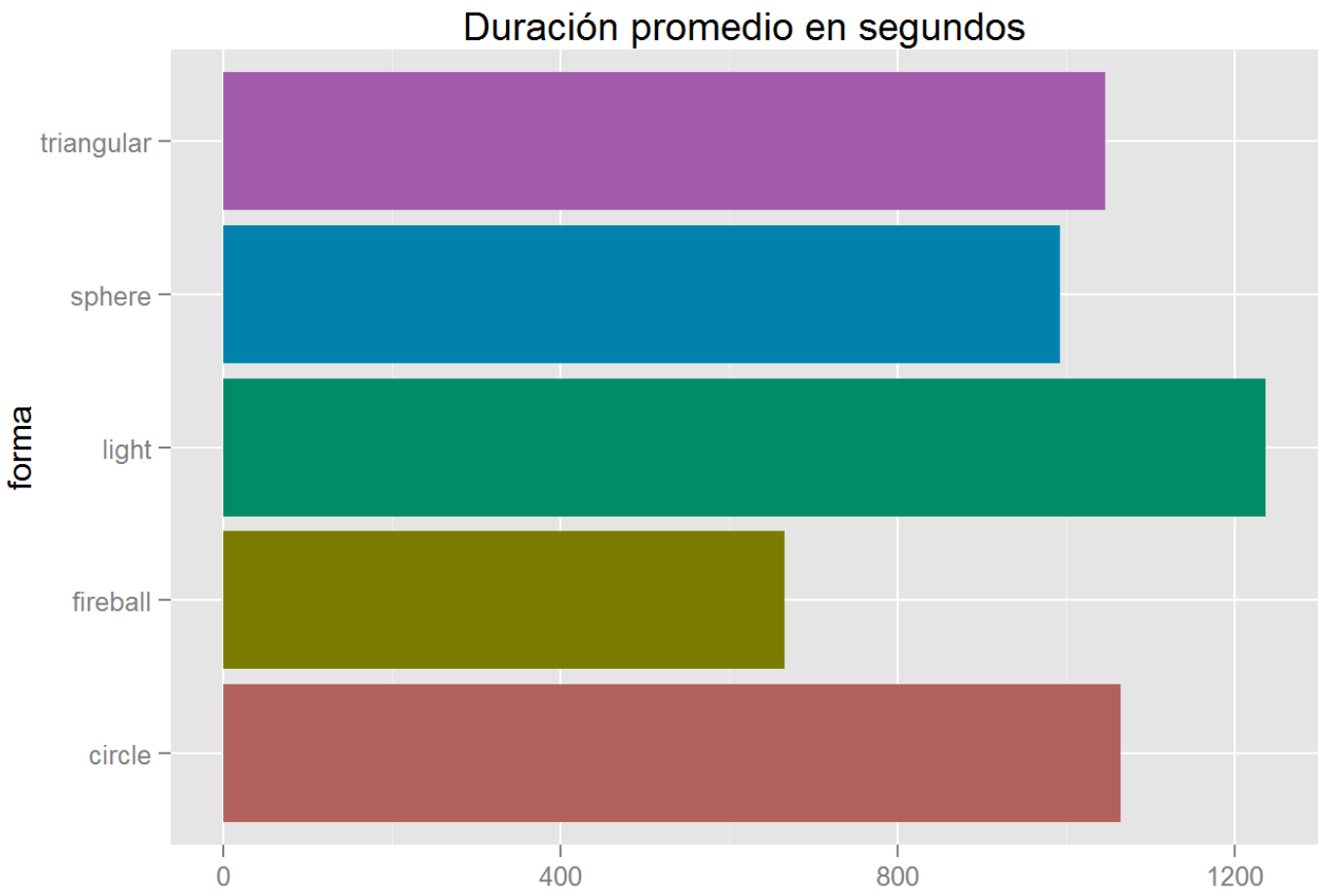
- Veamos la duración promedio de los avistamientos para los 5 estados con mayor número de avistamientos en la história.

```
# Obtenemos el promedio de duración por forma
dura_estado <- c_tabla_ufo %>%
    group_by(estado) %>%
    summarise(duracion = round(mean(duracion), 0), frecuencia = n())
dura_estado <- data.frame(dura_estado)
dura_estado <- dura_estado[with(dura_estado, order(-frecuencia)),]

# Graficamos
ggplot(data = dura_estado[1:5,], aes(x = estado, y = duracion, fill = estado)) +
  geom_bar(stat = 'identity') +
  theme(legend.position="none") +
  scale_y_continuous(name = "") +
  scale_fill_hue(l = 40) +
  ggtitle("Duración promedio en segundos")
```



- Ahora tenemos la duración promedio por tipo de forma del objeto, consideramos los 5 tipos de forma más reportados, sin tomar en cuenta las categorías *unknown* y *other*.

```
# Obtenemos el promedio de duración por forma
dura_forma <- c_tabla_ufo %>%
    filter(forma != c("unknown", "other")) %>%
    group_by(forma) %>%
    summarise(duracion = round(mean(duracion), 0), frecuencia = n())
dura_forma <- data.frame(dura_forma)
dura_forma <- dura_forma[with(dura_forma, order(-frecuencia)),]

# Graficamos
ggplot(data = dura_forma[1:5,], aes(x = forma, y = duracion, fill = forma)) +
  geom_bar(stat = 'identity') +
  theme(legend.position="none") +
  scale_y_continuous(name = "") +
  coord_flip() +
  scale_fill_hue(c = 60, l = 50) +
  ggtitle("Duración promedio en segundos")
```



Análisis de la variable `descripcion`

Vamos a contar el número de palabras de cada descripción para obtener el promedio de palabras utilizadas, para lo anterior utilizamos los textos que ya limpiamos.

```
# Obtenemos el número promedio de palabras
no_palabras <- sapply(c_tabla_ufo$descripcion, function(x) length(strsplit(as.characte
r(x), " ")[[1]]))
```

```
## [1] "El mínimo de palabras es:"
```

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

```
## [1] "El máximo de palabras es:"
```

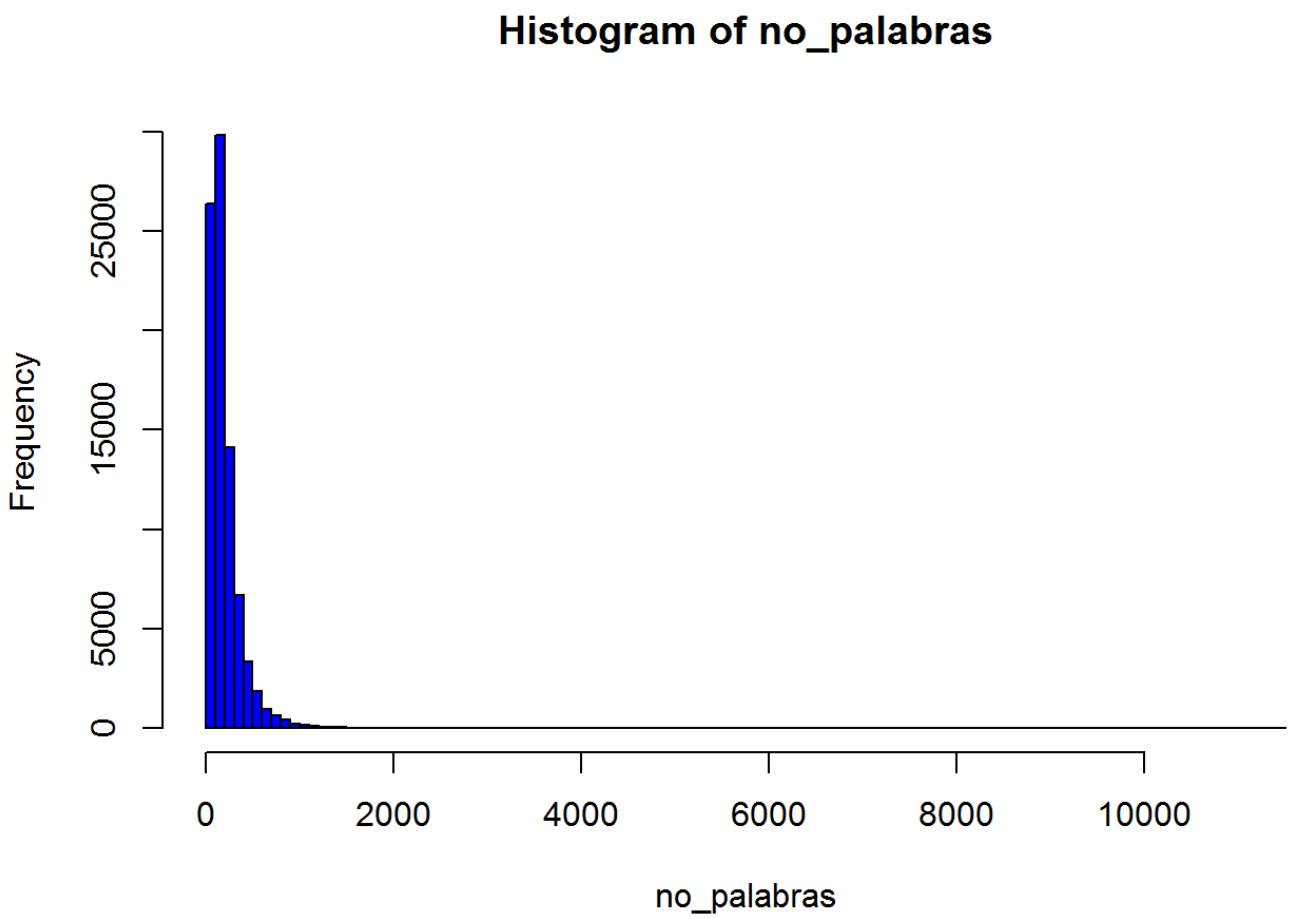
```
## [1] 11414
```

```
## [1] "El promedio de palabras es:"
```

```
## [1] 197
```

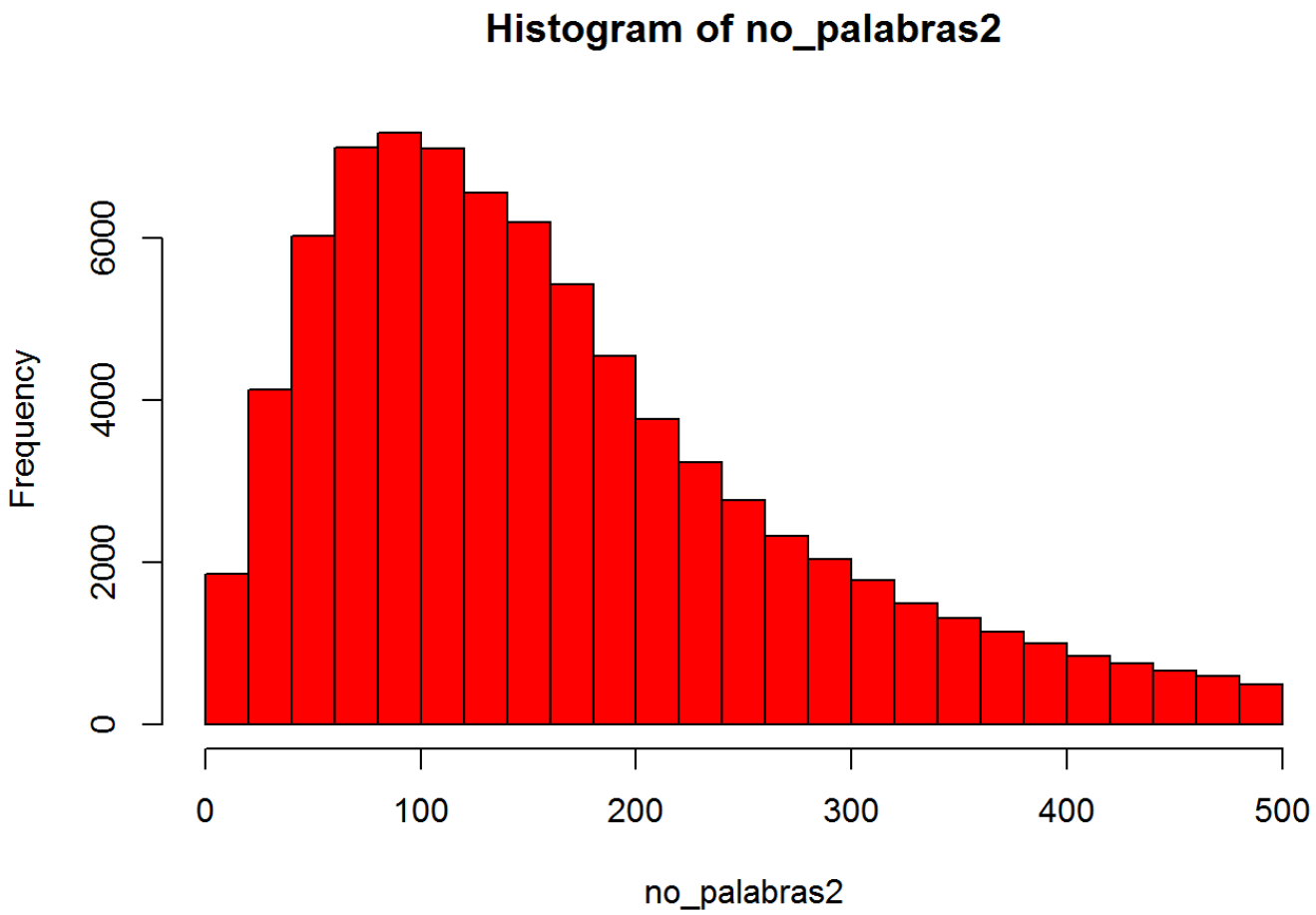
Graficando la frecuencia de palabras utilizadas tenemos:

```
# Graficamos frecuencias
inter <- seq(0, max(no_palabras) + 100, by = 100)
hist(no_palabras, breaks = inter, col = 'blue')
```



Verificando, intervalos más pequeñs tenemos que la mayor parte de los textos incluyen entre 60 y 120 palabras.

```
# Graficamos frecuencias sin considerar los textos con más de 100 palabras
no_palabras2 <- no_palabras[no_palabras <= 500]
inter2 <- seq(0, 500, by = 20)
hist(no_palabras2, breaks = inter2, col = 'red')
```



Para realizar el análisis de la variable `descripción` que corresponde al texto de las decripciones completas de los avistamientos, utilizaremos el paquete "*tm*" que nos permite utilizar algunas funciones para análisis de textos.

Para verificar algunas características de los textos realizamos los pasos siguientes:

- Primero preparamos los textos en el formato correspondiente para poder realizar el análisis.

```
# Damos el formato para análizar los textos
descripciones <- na.omit(c_tabla_ufo$descripcion)
textos <- Corpus(VectorSource(descripciones), readerControl = list(language ='english'))
```

- Completamos la limpieza de los textos.

```
# Se eliminan _Stopwords_ definidos para palabras en inglés
textos <- tm_map(textos, removeWords, stopwords("english"))

# Aplicamos _Stemming_ a los textos
textos <- tm_map(textos, stemDocument)

# Se eliminan los posibles espacios en blanco sobrantes
textos <- tm_map(textos, stripWhitespace)

# Se eliminan los espacios en blanco al inicio y/o final del texto
limpieza <- content_transformer(function(x, caracter) gsub(caracter, "", x))
textos <- tm_map(textos, limpieza, "(^[[:space:]]+|[[:space:]]+$)")
```

Verificamos que despues de esta limpieza complementaria, no hayan quedado registros sin texto.

```
no_palabras1 <- sapply(textos, function(x) length(strsplit(as.character(x), " ")[[1]]))
min(no_palabras1)
```

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

- Creamos Matriz Términos Documentos

Para analizar algunas características de los textos, de acuerdo a las palabras utilizadas en las descripciones creamos la matriz de términos documentos ponderada por frecuencia de términos.

El resultado de la matriz términos documentos, nos muestra un valor alto en el parámetro *Sparsity*, lo cual nos indica que hay diversos términos que no aparecen en diversos documentos.

```
tdm_textos <- TermDocumentMatrix(textos, control = list(wordLengths = c(1, Inf)))
tdm_textos
```

```
## <<TermDocumentMatrix (terms: 158650, documents: 85119)>>
## Non-/sparse entries: 6093097/13498036253
## Sparsity           : 100%
## Maximal term length: 271
## Weighting          : term frequency (tf)
```

Dado el resultado del parámetro *Sparsity* y las dimensiones de nuestra matriz, aplicaremos la función *removeSparseTerms* para eliminar los términos escasos.

```
tdm_textos1 <- removeSparseTerms(tdm_textos, sparse = 0.99)
tdm_textos1
```

```
## <<TermDocumentMatrix (terms: 1013, documents: 85119)>>
## Non-/sparse entries: 4833582/81391965
## Sparsity           : 94%
## Maximal term length: 12
## Weighting          : term frequency (tf)
```

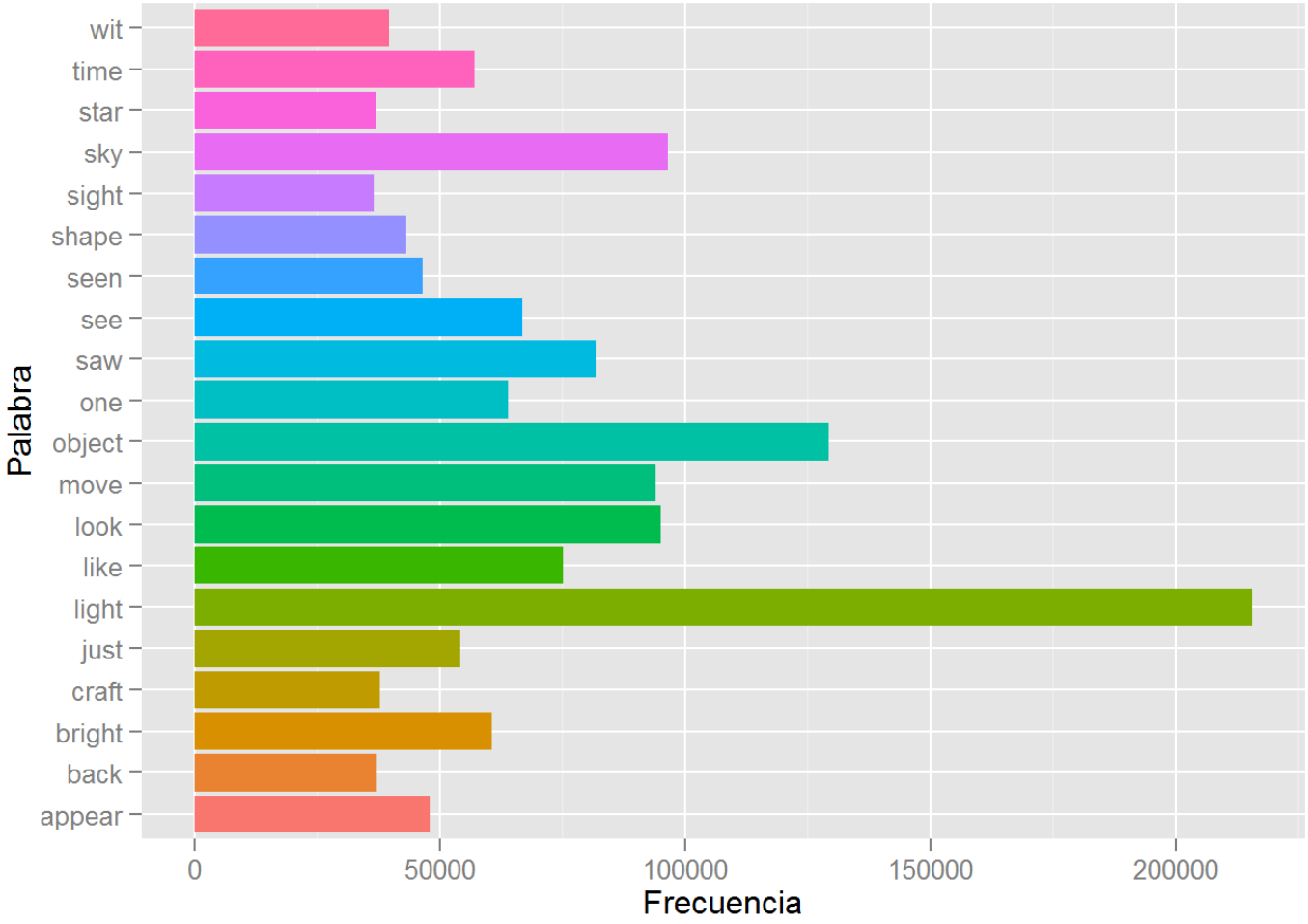
Con lo anterior, vemos que la cantidad de términos reduce significativamente, lo cual nos muestra que había muchos términos que no aportarían información importante al análisis.

Obtenemos la frecuencia histórica de cada término de nuestra matriz y generamos histograma de frecuencia de las palabras más utilizadas en las descripciones. Verificamos que la palabra *light* es la palabra más frecuente, seguida por *object* y *sky*.

```
# Obtenemos frecuencias
freq_palabras <- sort(rowSums(as.matrix(tdm_textos1)), decreasing = TRUE)
freq_palabras_df <- data.frame(Palabra = names(freq_palabras), Frecuencia = freq_palabra
s)
head(freq_palabras_df)
```

```
##        Palabra Frecuencia
## light    light     215670
## object  object     129337
## sky        sky      96575
## look      look      95035
## move      move      93994
## saw        saw      81841
```

```
# Obtenemos gráfica de las 20 palabras más frecuentes
ggplot(data = freq_palabras_df[1:20,], aes(x = Palabra, y = Frecuencia, fill = Palabra))
+
  geom_bar(stat = "identity") +
  coord_flip() +
  theme(legend.position="none")
```



6. Anexo

- Tabla de categorías de la variable `estado`

Nombre de la variable	Códigos	Descripción del código	Códigos	Descripción del código
State / estado	AL	Alabama	MT	Montana
	AK	Alaska	NE	Nebraska
	AZ	Arizona	NV	Nevada
	AR	Arkansas	NH	New Hampshire
	CA	California	NJ	New Jersey
	CO	Colorado	NM	New Mexico
	CT	Connecticut	NY	New York
	DE	Delaware	NC	North Carolina
	DC	District of Columbia	ND	North Dakota
	FL	Florida	OH	Ohio
	GA	Georgia	OK	Oklahoma
	HI	Hawaii	OR	Oregon
	ID	Idaho	PA	Pennsylvania
	IL	Illinois	RI	Rhode Island
	IN	Indiana	SC	South Carolina
	IA	Iowa	SD	South Dakota
	KS	Kansas	TN	Tennessee
	KY	Kentucky	TX	Texas
	LA	Louisiana	UT	Utah
	ME	Maine	VT	Vermont
	MD	Maryland	VA	Virginia
	MA	Massachusetts	WA	Washington
	MI	Michigan	WV	West Virginia
	MN	Minnesota	WI	Wisconsin
	MS	Mississippi	WY	Wyoming
	MO	Missouri		

- Tabla de categorías de la variable `forma`

Nombre de la variable	Categorías	
Shape / forma	changed	flare
	chevron	flash
	cigar	formation
	circle	hexagon
	cone	light
	crescent	other
	cross	oval
	cylinder	pyramid
	delta	rectangle
	diamond	round
	disk	sphere
	dome	teardrop
	egg	triangular
	fireball	unknown

- Tabla de categorías de la variable `mes`

Nombre de la variable	Códigos	Descripción del código
mes	01	Enero
	02	Febrero
	03	Marzo
	04	Abril
	05	Mayo
	06	Junio
	07	Julio
	08	Agosto
	09	Septiembre
	10	Octubre
	11	Noviembre
	12	Diciembre