Minería de datos: PEC1

Autor: Pablo Suárez Reyero

octubre 2023

Contents

Ejemplo guiado

No se trata de una pauta para repetir sino diferentes ejemplos para daros ideas e inspiraros.

Descripción del origen del conjunto de datos

Se ha seleccionado un conjunto de datos del National Highway Traffic Safety Administration. El sistema de informes de análisis de mortalidad fue creado en los Estados Unidos por la National Highway Traffic Safety Administration para proporcionar una medida global de la seguridad en las carreteras. (Fuente Wikipedia). Los datos pertenecen al año 2020. Se trata de un conjunto de registros de accidentes que recogen datos significativos que los describen. Todos los accidentes tienen alguna víctima mortal como mínimo. El objetivo analítico que tenemos en mente es entender que hace que un accidente sea grave y que quiere decir que sea grave. https://www.nhtsa.gov/crash-data-systems/fatality-analysis-reporting-system

Análisis exploratorio

Queremos hacer una primera aproximación al conjunto de datos escogido y responder a las preguntas más básicas: ¿Cuánto registros tiene? ¿Cuántas variables? ¿De qué tipología son? ¿Cómo se distribuyen los valores de las variables? ¿Hay problemas con los datos, por ejemplo, campos vacíos? ¿Puedo intuir ya el valor analítico de los datos? ¿Qué primeras conclusiones puedo extraer?

El primer paso para realizar un análisis exploratorio es cargar el fichero de datos.

```
path = 'accident.CSV'
accidentData <- read.csv(path, row.names=NULL)
```

Exploración del conjunto de datos

Verificamos la estructura del juego de datos principal. Vemos el número de columnas que tenemos y ejemplos de los contenidos de las filas.

```
structure = str(accidentData)
```

```
## 'data.frame':    35766 obs. of  81 variables:
##  $ STATE       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ STATENAME   : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ ST_CASE     : int  10001 10002 10003 10004 10005 10006 10007 10008 10009 10010 ...
##  $ VE_TOTAL    : int  1 4 2 1 1 2 1 2 2 2 ...
##  $ VE_FORMS    : int  1 4 2 1 1 2 1 2 2 2 ...
##  $ PVH_INVL    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PEDS        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ PERSONS     : int  4 6 2 5 1 3 1 2 4 3 ...
##  $ PERMVIT     : int  4 6 2 5 1 3 1 2 4 3 ...
##  $ PERNOTMVIT  : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ COUNTY      : int  51 73 117 15 37 103 73 25 45 95 ...
##  $ COUNTYNAME  : chr  "ELMORE (51)" "JEFFERSON (73)" "SHELBY (117)" "CALHOUN (15)" ...
##  $ CITY        : int  0 350 0 0 0 0 330 0 0 1500 ...
##  $ CITYNAME    : chr  "NOT APPLICABLE" "BIRMINGHAM" "NOT APPLICABLE" "NOT APPLICABLE" ...
##  $ DAY         : int  1 2 2 3 4 4 7 8 9 10 ...
##  $ DAYNAME     : int  1 2 2 3 4 4 7 8 9 10 ...
##  $ MONTH       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ MONTHNAME   : chr  "January" "January" "January" "January" ...
##  $ YEAR        : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ DAY_WEEK    : int  4 5 5 6 7 7 3 4 5 6 ...
##  $ DAY_WEEKNAME: chr  "Wednesday" "Thursday" "Thursday" "Friday" ...
##  $ HOUR        : int  2 17 14 15 0 16 19 7 20 10 ...
##  $ HOURNAME    : chr  "2:00am-2:59am" "5:00pm-5:59pm" "2:00pm-2:59pm" "3:00pm-3:59pm" ...
##  $ MINUTE      : int  58 18 55 20 45 55 23 15 0 2 ...
##  $ MINUTENAME  : chr  "58" "18" "55" "20" ...
##  $ NHS         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ NHSNAME     : chr  "This section IS NOT on the NHS" "This section IS NOT on the NHS" "This sectio
##  $ ROUTE       : int  4 6 3 4 4 3 4 4 4 2 ...
##  $ ROUTENAME   : chr  "County Road" "Local Street - Municipality" "State Highway" "County Road" ...
##  $ TWAY_ID     : chr  "cr-4" "martin luther king jr dr" "sr-76" "CR-ALEXANDRIA WELLINGTON RD" ...
##  $ TWAY_ID2    : chr  "" "" "us-280" "" ...
##  $ RUR_URB     : int  1 2 1 1 1 1 2 1 1 1 ...
##  $ RUR_URBNAME : chr  "Rural" "Urban" "Rural" "Rural" ...
##  $ FUNC_SYS    : int  5 4 4 7 5 4 4 5 5 3 ...
##  $ FUNC_SYSNAME: chr  "Major Collector" "Minor Arterial" "Minor Arterial" "Local" ...
##  $ RD_OWNER    : int  2 4 1 2 2 1 4 2 2 1 ...
##  $ RD_OWNERNAME: chr  "County Highway Agency" "City or Municipal Highway Agency" "State Highway Agen
##  $ MILEPT      : int  0 0 49 0 0 390 0 0 0 3019 ...
##  $ MILEPTNAME  : chr  "None" "None" "49" "None" ...
##  $ LATITUDE    : num  32.4 33.5 33.3 33.8 32.8 ...
##  $ LATITUDENAME: chr  "32.43313333" "33.48465833" "33.29994167" "33.79507222" ...
##  $ LONGITUD    : num  -86.1 -86.8 -86.4 -85.9 -86.1 ...
```

```
##  $ LONGITUDNAME: chr  "-86.09485" "-86.83954444" "-86.36964167" "-85.88348611" ...
##  $ SP_JUR      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SP_JURNAME  : chr  "No Special Jurisdiction" "No Special Jurisdiction" "No Special Jurisdiction" "
##  $ HARM_EV     : int  42 12 34 42 42 12 8 12 12 12 ...
##  $ HARM_EVNAME : chr  "Tree (Standing Only)" "Motor Vehicle In-Transport" "Ditch" "Tree (Standing Onl
##  $ MAN_COLL    : int  0 6 0 0 0 2 0 1 1 2 ...
##  $ MAN_COLLNAME: chr  "The First Harmful Event was Not a Collision with a Motor Vehicle in Transport"
##  $ RELJCT1     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RELJCT1NAME : chr  "No" "No" "No" "No" ...
##  $ RELJCT2     : int  1 1 3 1 1 1 3 1 8 1 ...
##  $ RELJCT2NAME : chr  "Non-Junction" "Non-Junction" "Intersection-Related" "Non-Junction" ...
##  $ TYP_INT     : int  1 1 3 1 1 1 2 1 1 1 ...
##  $ TYP_INTNAME : chr  "Not an Intersection" "Not an Intersection" "T-Intersection" "Not an Intersecti
##  $ WRK_ZONE    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WRK_ZONENAME: chr  "None" "None" "None" "None" ...
##  $ REL_ROAD    : int  4 1 4 4 4 1 1 1 1 1 ...
##  $ REL_ROADNAME: chr  "On Roadside" "On Roadway" "On Roadside" "On Roadside" ...
##  $ LGT_COND    : int  2 3 1 1 2 2 3 1 2 1 ...
##  $ LGT_CONDNAME: chr  "Dark - Not Lighted" "Dark - Lighted" "Daylight" "Daylight" ...
##  $ WEATHER     : int  1 2 2 10 2 1 1 1 10 10 ...
##  $ WEATHERNAME : chr  "Clear" "Rain" "Rain" "Cloudy" ...
##  $ SCH_BUS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SCH_BUSNAME : chr  "No" "No" "No" "No" ...
##  $ RAIL        : chr  "0000000" "0000000" "0000000" "0000000" ...
##  $ RAILNAME    : chr  "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
##  $ NOT_HOUR    : int  99 17 14 99 0 17 19 7 20 10 ...
##  $ NOT_HOURNAME: chr  "Unknown" "5:00pm-5:59pm" "2:00pm-2:59pm" "Unknown" ...
##  $ NOT_MIN     : int  99 18 58 99 45 0 23 21 0 3 ...
##  $ NOT_MINNAME : chr  "Unknown" "18" "58" "Unknown" ...
##  $ ARR_HOUR    : int  3 17 15 99 0 17 19 7 20 10 ...
##  $ ARR_HOURNAME: chr  "3:00am-3:59am" "5:00pm-5:59pm" "3:00pm-3:59pm" "Unknown EMS Scene Arrival Hou
##  $ ARR_MIN     : int  10 26 15 99 55 19 29 28 10 7 ...
##  $ ARR_MINNAME : chr  "10" "26" "15" "Unknown EMS Scene Arrival Minutes" ...
##  $ HOSP_HR     : int  99 99 99 99 88 18 88 88 99 10 ...
##  $ HOSP_HRNAME : chr  "Unknown" "Unknown" "Unknown" "Unknown" ...
##  $ HOSP_MN     : int  99 99 99 99 88 51 88 88 99 29 ...
##  $ HOSP_MNNAME : chr  "Unknown EMS Hospital Arrival Time" "Unknown EMS Hospital Arrival Time" "Unknou
##  $ FATALS      : int  3 1 1 1 1 1 1 1 1 1 ...
##  $ DRUNK_DR    : int  1 0 0 0 0 0 0 0 0 0 ...
```

Vemos que tenemos **81** variables y **35.766** registros

Revisamos la descripción de las variables contenidas en el fichero y si los tipos de variables se corresponden con las que hemos cargado. Las organizamos lógicamente para darles sentido y construimos un pequeño diccionario de datos utilizando la documentación auxiliar.

- **ST_CASE** identificador de accidente

HECHOS A ESTUDIAR

- **FATAL** muertes
- **DRUNK_DR** conductores bebidos
- **VE_TOTAL** número de vehículos implicados en total
- **VE_FORMS** número de vehículos en movimiento implicados
- **PVH_INVL** número de vehículos estacionados implicados
- **PEDS** número de peatones implicados

- **PERSONS** número de ocupantes de vehículo implicados
- **PERMVIT** número conductores y ocupantes implicados
- **PERNOTMVIT** número peatones, ciclistas, a caballo... Cualquier cosa menos vehículo motorizado

DIMENSIÓN GEOGRÁFICA

- **STATE** codificación de estado
- **STATENAME** nombre de estado
- **COUNTY** identificador de contado
- **COUNTYNAME** condado
- **CITY** identificador de ciudad
- **CITYNAME** ciudad
- **NHS** 1 ha pasado a autopista del NHS 0 no
- **NHSNAME** TBD
- **ROUTE** identificador de ruta
- **ROUTENAME** ruta
- **TWAY_ID** vía de tránsito (1982)
- **TWAY_ID2** vía de tránsito (2004)
- **RUR_URB** identificador de segmento rural o urbano
- **RUR_URBNAME** segmento rural o urbano
- **FUNC_SYS** clasificación funcional segmento
- **FUNC_SYSNAME** TBD
- **RD_OWNER** identificador propietario del segmento
- **RD_OWNERNAME** propietario del segmento
- **MILEPT** milla int
- **MILEPTNAME** milla chr
- **LATITUDE** latitud int
- **LATITUDENAME** latitud chr
- **LONGITUD** longitud int
- **LONGITUDNAME** longitud chr
- **SP_JUR** código jurisdicción
- **SP_JURNAME** jurisdicción

DIMENSIÓN TEMPORAL

- **DAY** día
- **DAYNAME** día repetido
- **MONTH** mes
- **MONTHNAME** nombre de mes
- **YEAR** año
- **DAY_WEEK** día de la semana
- **DAY_WEEKNAME** nombre de día de la semana
- **HOUR** hora
- **HOURNAME** franja hora
- **MINUTE** minuto int
- **MINUTENAME** minuto chr

DIMENSIÓN CONDICICIONES ACCIDENTE

- **HARM_EV** código primer acontecimiento del accidente que produzca daños o lesiones
- **HARM_EVNAME** primer acontecimiento del accidente que produzca daños o lesiones
- **MAN_COLL** código de posición de los vehículos
- **MAN_COLLNAME** posición de los vehículos
- **RELJCT1** código si hay área de intercambio
- **RELJCT1NAME** si hay área de intercambio
- **RELJCT2** código proximidad cruce
- **RELJCT2NAME** proximidad cruce

- **TYP_INT** código tipo de intersección
- **TYP_INTNAME** tipo de intersección
- **WRK_ZONE** código tipología de obras
- **WRK_ZONENAME** tipología de obras
- **RAIL_ROAD** código ubicación vehículo a la vía
- **RAIL_ROADNAME** ubicación vehículo a la vía
- **LGT_COND** código condición lumínica
- **LGT_CONDNAME** condición lumínica

DIMENSIÓN METEOROLOGIA

- **WEATHER** código tiempo
- **WEATHERNAME** tiempo

OTROS

- **SCH_BUSS** código si vehículo escolar implicado
- **SCH_BUSNAME** vehículo escolar implicado
- **RAIL** código si dentro o cerca paso ferroviario
- **RAILNAME** si dentro o cerca paso ferroviario

DIMENSIÓN SERVICIO EMERGENCIAS

- **NOT_HOUR** hora notificación a emergencias int
- **NOT_HOURNAME** hora notificación a emergencias franja
- **NOT_MIN** minuto notificación a emergencias int
- **NOT_MINNAME** minuto notificación a emergencias chr
- **ARR_HOUR** hora llegada emergencias int
- **ARR_HOURNAME** hora llegada emergencias franja
- **ARR_MIN** minuto llegada emergencias int
- **ARR_MINNAME** minuto llegada emergencias franja
- **HOSP_HR** hora llegada hospital int
- **HOSP_HRNAME** hora llegada hospital franja
- **HOSP_MN** minuto llegada hospital int
- **HOSP_MNNAME** minuto llegada hospital franja

DIMENSIÓN FACTORES RELACIONADOS ACCIDENTE

- **CF1** código factores relacionados con el accidente 1
- **CF1NAME** factores relacionados con el accidente 1
- **CF2** código factores relacionados con el accidente 2
- **CF2NAME** factores relacionados con el accidente 2
- **CF3** código factores relacionados con el accidente 3

Preprocesamiento y gestión de características

Limpieza

El siguiente paso será la limpieza de datos, mirando si hay valores vacíos o nulos.

```
print('NA')
```

```
## [1] "NA"
```

```
colSums(is.na(accidentData))
```

```
##        STATE    STATENAME      ST_CASE     VE_TOTAL     VE_FORMS     PVH_INVL
##            0            0            0            0            0            0
##         PEDS      PERSONS      PERMVIT   PERNOTMVIT       COUNTY   COUNTYNAME
##            0            0            0            0            0            0
```

```
##          CITY     CITYNAME          DAY      DAYNAME        MONTH    MONTHNAME
##             0            0            0            0            0            0
##          YEAR     DAY_WEEK DAY_WEEKNAME         HOUR     HOURNAME       MINUTE
##             0            0            0            0            0            0
##    MINUTENAME          NHS      NHSNAME        ROUTE    ROUTENAME      TWAY_ID
##             0            0            0            0            0            0
##      TWAY_ID2      RUR_URB  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER
##             0            0            0            0            0            0
## RD_OWNERNAME       MILEPT   MILEPTNAME     LATITUDE LATITUDENAME     LONGITUD
##             0            0            0            0            0            0
## LONGITUDNAME       SP_JUR   SP_JURNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##             0            0            0            0            0            0
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##             0            0            0            0            0            0
##   TYP_INTNAME     WRK_ZONE WRK_ZONENAME     REL_ROAD REL_ROADNAME     LGT_COND
##             0            0            0            0            0            0
## LGT_CONDNAME      WEATHER  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL
##             0            0            0            0            0            0
##      RAILNAME     NOT_HOUR NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR
##             0            0            0            0            0            0
## ARR_HOURNAME      ARR_MIN  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN
##             0            0            0            0            0            0
##   HOSP_MNNAME       FATALS     DRUNK_DR
##             0            0            0
```

```
print('Blancos')
```

```
## [1] "Blancos"
```

```
colSums(accidentData=="")
```

```
##         STATE     STATENAME       ST_CASE      VE_TOTAL      VE_FORMS      PVH_INVL
##             0             0             0             0             0             0
##          PEDS       PERSONS       PERMVIT    PERNOTMVIT        COUNTY    COUNTYNAME
##             0             0             0             0             0             0
##          CITY      CITYNAME           DAY       DAYNAME         MONTH     MONTHNAME
##             0             0             0             0             0             0
##          YEAR      DAY_WEEK  DAY_WEEKNAME          HOUR      HOURNAME        MINUTE
##             0             0             0             0             0             0
##    MINUTENAME           NHS       NHSNAME         ROUTE     ROUTENAME       TWAY_ID
##             0             0             0             0             0             0
##      TWAY_ID2       RUR_URB   RUR_URBNAME      FUNC_SYS  FUNC_SYSNAME      RD_OWNER
##         26997             0             0             0             0             0
## RD_OWNERNAME        MILEPT    MILEPTNAME      LATITUDE  LATITUDENAME      LONGITUD
##             0             0             0             0             0             0
## LONGITUDNAME        SP_JUR    SP_JURNAME       HARM_EV   HARM_EVNAME      MAN_COLL
##             0             0             0             0             0             0
## MAN_COLLNAME       RELJCT1   RELJCT1NAME       RELJCT2   RELJCT2NAME       TYP_INT
##             0             0             0             0             0             0
##   TYP_INTNAME      WRK_ZONE  WRK_ZONENAME      REL_ROAD  REL_ROADNAME      LGT_COND
##             0             0             0             0             0             0
## LGT_CONDNAME       WEATHER   WEATHERNAME       SCH_BUS   SCH_BUSNAME          RAIL
##             0             0             0             0             0             0
##      RAILNAME      NOT_HOUR  NOT_HOURNAME       NOT_MIN   NOT_MINNAME      ARR_HOUR
##             0             0             0             0             0             0
```

```
## ARR_HOURNAME       ARR_MIN  ARR_MINNAME       HOSP_HR  HOSP_HRNAME       HOSP_MN
##            0             0            0             0            0             0
##  HOSP_MNNAME       FATALS     DRUNK_DR
##            0             0            0
```

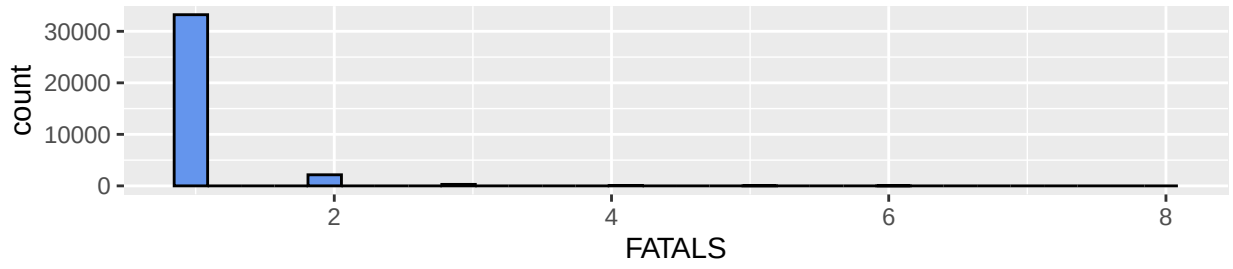
Vemos que no hay valores nulos en los datos. También verificamos si existen campos llenos de espacios en blanco. En este caso sí encontramos el campo TWAY_ID2 con 26.997 valores en blanco. Valoramos no hacer ninguna acción de eliminar registros puesto que este campo no lo utilizaremos.

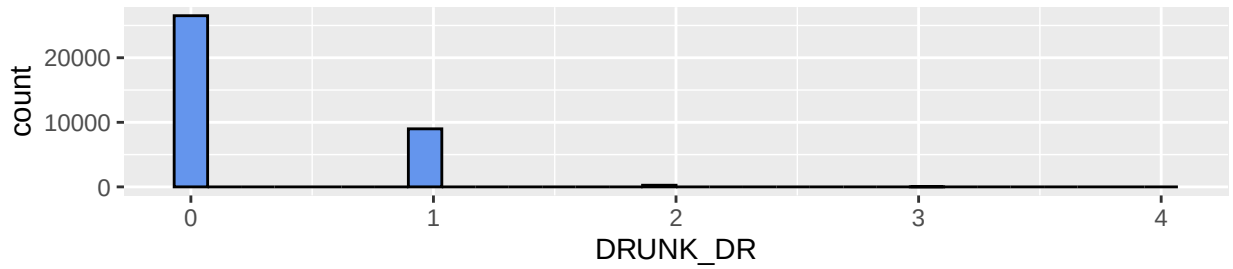
Vamos a crear histogramas y describir los valores para ver los datos en general de estos atributos para hacer una primera aproximación a los datos:

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if(!require('Rmisc')) install.packages('Rmisc'); library('Rmisc')
if(!require('dplyr')) install.packages('dplyr'); library('dplyr')
if(!require('xfun')) install.packages('xfun'); library('xfun')

summary(accidentData[c("FATALS","DRUNK_DR")])
```

```
##      FATALS          DRUNK_DR
##  Min.   :1.000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.0000
##  Median :1.000   Median :0.0000
##  Mean   :1.085   Mean   :0.2664
##  3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :8.000   Max.   :4.0000
```

```
histList<- list()

n = c("FATALS","DRUNK_DR")
accidentDataAux= accidentData %>% select(all_of(n))
for(y in 1:ncol(accidentDataAux)){
col <- names(accidentDataAux)[y]
ggp <- ggplot(accidentDataAux, aes_string(x = col)) +
geom_histogram(bins = 30, fill = "cornflowerblue", color = "black",ggtittle = "Contador de ocurrencias p
histList[[y]] <- ggp # añadimos cada plot a la lista vacía
}
multiplot(plotlist = histList, coles = 1)
```





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

Observaciones:

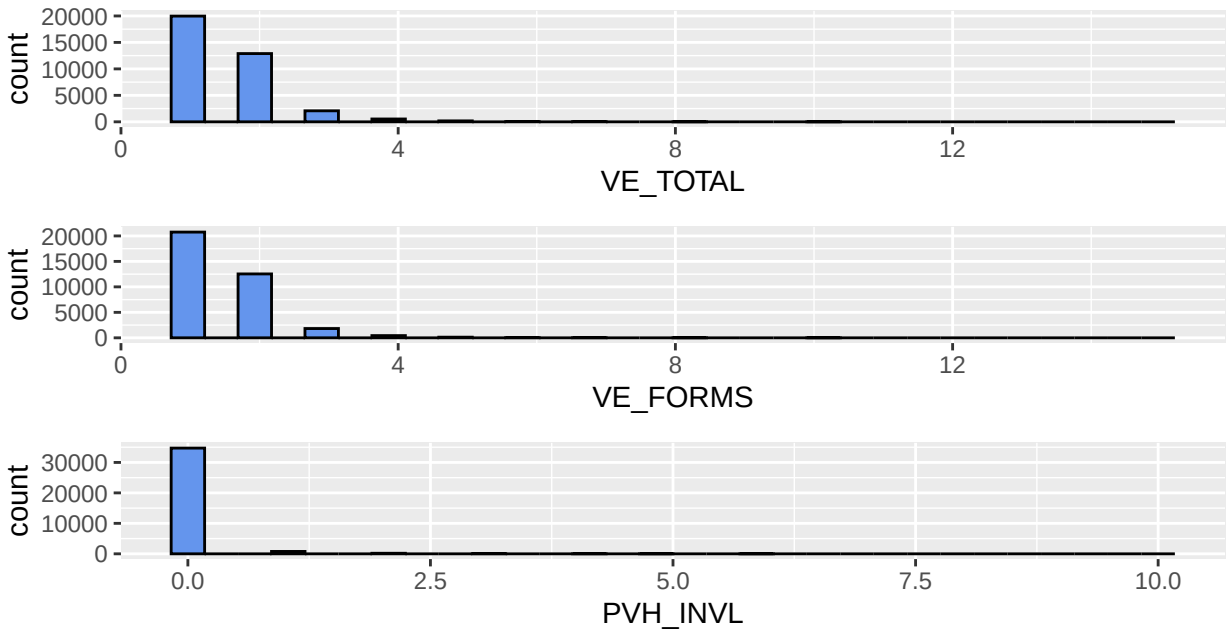
Número de muertes: Todos los accidentes recogidos en estos datos reportan una muerte como mínimo. Siendo el accidente más grave con ocho víctimas y vemos que la distribución se acumula de forma muy evidente en una muerte por accidente.

Conductores bebidos involucrados en el accidente: Analizaremos con más detalle este dato más adelante para derivar un nuevo dato: Accidentes donde el alcohol está presente o no. De entrada, la media es de 0.26% de accidentes donde interviene un conductor bebido. La franja de conductores bebidos por accidente va de un conductor como mínimo a cuatro como máximo.

```
summary(accidentData[c("VE_TOTAL","VE_FORMS","PVH_INVL")])
```

```
##     VE_TOTAL        VE_FORMS         PVH_INVL
##  Min.   : 1.00   Min.   : 1.000   Min.   : 0.00000
##  1st Qu.: 1.00   1st Qu.: 1.000   1st Qu.: 0.00000
##  Median : 1.00   Median : 1.000   Median : 0.00000
##  Mean   : 1.56   Mean   : 1.517   Mean   : 0.04269
##  3rd Qu.: 2.00   3rd Qu.: 2.000   3rd Qu.: 0.00000
##  Max.   :15.00   Max.   :15.000   Max.   :10.00000
```

```
#Crearemos una lista para mostrar los atributos que interesan.
histList<- list()
n = c("VE_TOTAL","VE_FORMS","PVH_INVL")
accidentDataAux= accidentData %>% select(all_of(n))
for(y in 1:ncol(accidentDataAux)){
col <- names(accidentDataAux)[y]
ggp <- ggplot(accidentDataAux, aes_string(x = col)) +
geom_histogram(bins = 30, fill = "cornflowerblue", color = "black")
histList[[y]] <- ggp # añadimos cada plot a la lista vacía
}
multiplot(plotlist = histList, coles = 1)
```




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

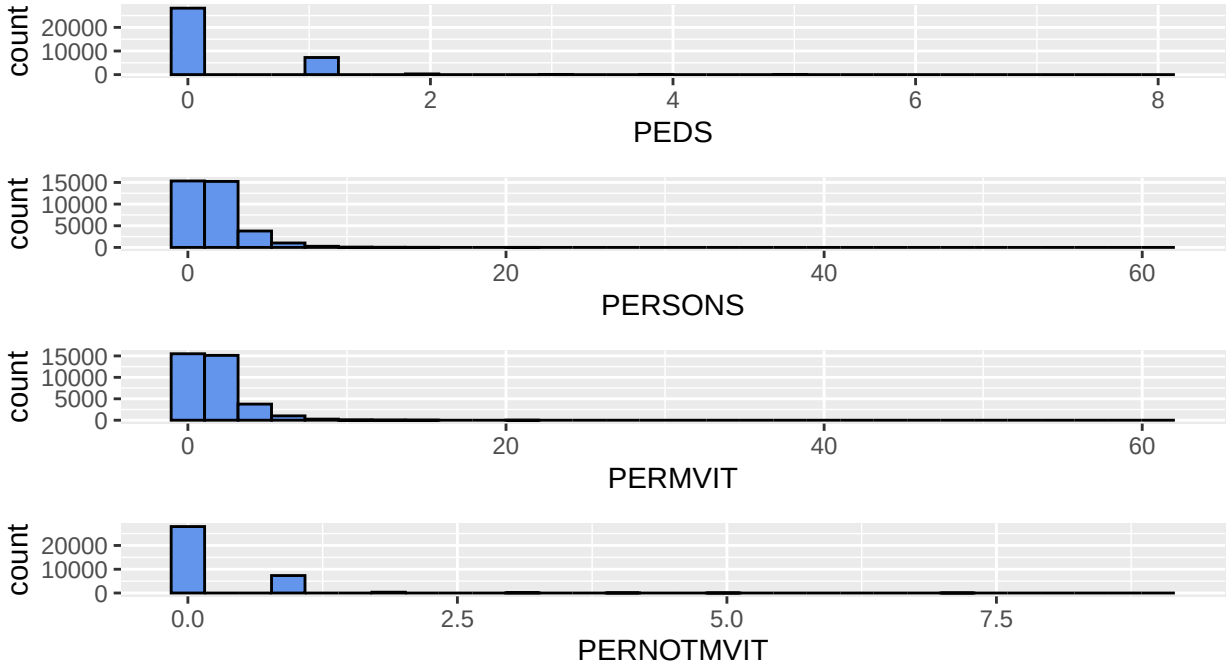
Observaciones en cuanto a los vehículos implicados.

Número de vehículos implicados (VE_TOTAL) en total está en la franja de 1 hasta 59 siendo este el valor máximo y una media de 1,5. Número de vehículos en movimiento implicados (VE_FORMS), el valor más habitual es 1 con un valor máximo también de 59. Prevemos que hay un valor extremo que habrá que tratar para poder sacar más información de esta variable. Número de vehículos estacionados implicados (PVH_INVL): Por lo que respecta a esta variable lo habitual es que no haya vehículos estacionados en los incidentes recogidos en estos datos. Con todo aparecen casos aislados donde incluso había 10 coches estacionados.

```
summary(accidentData[c("PEDS","PERSONS","PERMVIT","PERNOTMVIT")])
```

```
##       PEDS            PERSONS          PERMVIT         PERNOTMVIT
##  Min.   :0.0000   Min.   : 0.000   Min.   : 0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:0.0000
##  Median :0.0000   Median : 2.000   Median : 2.000   Median :0.0000
##  Mean   :0.2285   Mean   : 2.173   Mean   : 2.163   Mean   :0.2387
##  3rd Qu.:0.0000   3rd Qu.: 3.000   3rd Qu.: 3.000   3rd Qu.:0.0000
##  Max.   :8.0000   Max.   :61.000   Max.   :61.000   Max.   :9.0000
```

```
#Crearemos una lista para mostrar los atributos que interesan.
histList<- list()
n = c("PEDS","PERSONS","PERMVIT","PERNOTMVIT")
accidentDataAux= accidentData %>% select(all_of(n))
for(y in 1:ncol(accidentDataAux)){
col <- names(accidentDataAux)[y]
ggp <- ggplot(accidentDataAux, aes_string(x = col)) +
geom_histogram(bins = 30, fill = "cornflowerblue", color = "black")
histList[[y]] <- ggp # añadimos cada plot a la lista vacía
}
multiplot(plotlist = histList, coles = 1)
```




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

Observaciones en cuanto a las personas implicadas en un accidente.

El número de peatones implicados (PEDS) es muy bajo siendo coherente con el tipo de vía que se estudia y dónde no es habitual que haya gente andando. Con todo el valor como media de 0,22 y máximo de 8 obliga a investigar más este dato. (PERSONS) El número de ocupantes de vehículo implicados se sitúa como media en 2,1 (PERMVIT) El número conductores y ocupantes de vehículos en circulación implicados tiene un valor de media de 2,1. Estas dos variables recogen la misma información, pero la cuantifican de diferente manera. El accidente con el mayor número de ocupantes es de 61 personas. Por lo que respecta al número peatones, ciclistas, personas en vehículos aparcados y otros (PERNOTMVIT) vemos que aumenta un poco la media respecto a peatón puesto que entendemos que se incluyen más casos.

Vamos a profundizar un poco en el tema de la relación del alcohol en los conductores y el número de accidentes.

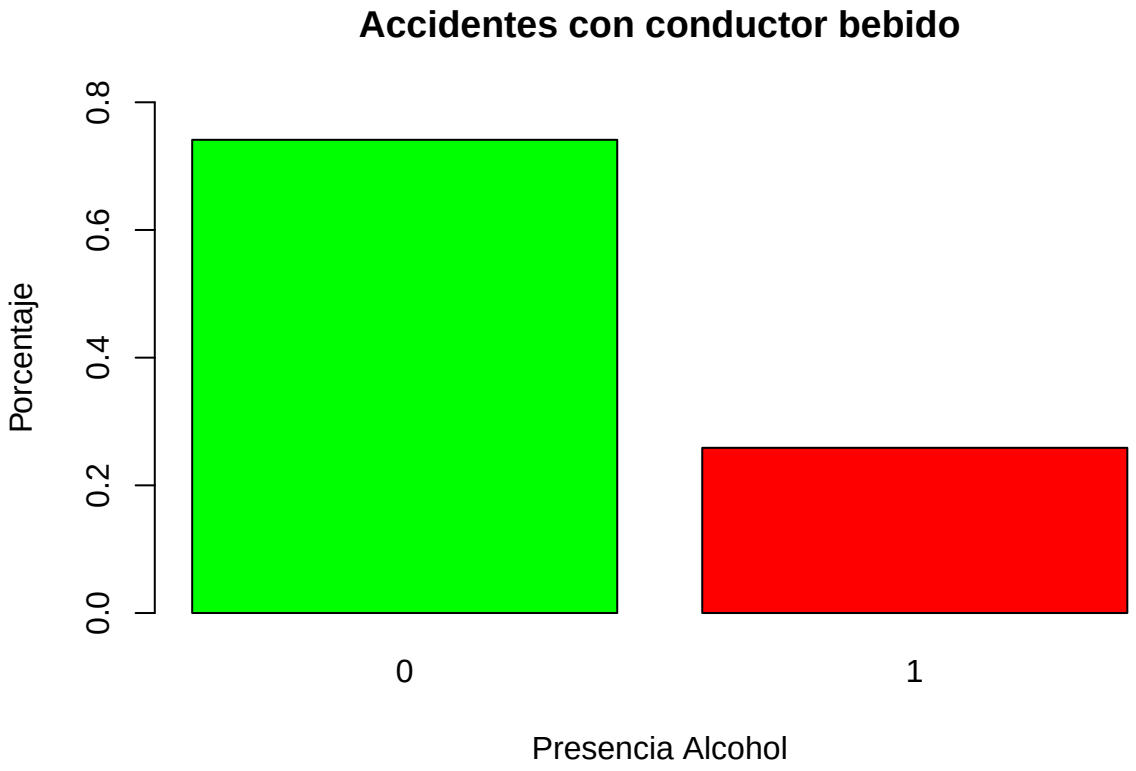
```
accidentData$alcohol <- ifelse(accidentData$DRUNK_DR %in% c(0), 0, 1)
counts <- table(accidentData$alcohol)
barplot(prop.table(counts),col=c("green","red"), main="Accidentes con conductor bebido", legend.texto=c
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "legend.texto" is not a
## graphical parameter
```

```
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : "legend.texto" is not a graphical parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "legend.texto" is not a graphical parameter
```

```
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): "legend.texto"
## is not a graphical parameter
```



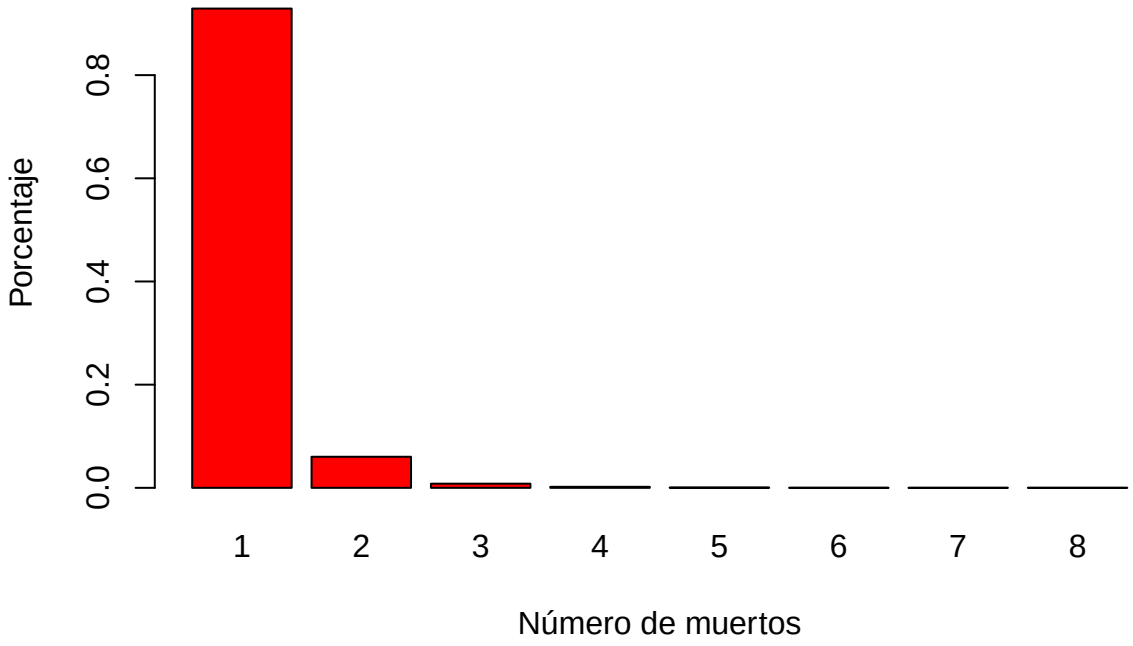
Vemos que porcentualmente, en la gran mayoría de accidentes, alrededor del 75% no hay presencia de alcohol en el conductor. Los conductores que dan positivo están alrededor de un 22%. Hemos buscado contrastar el dato con otros países y estarían en un valor central donde los valores extremos máximo por país superan el 50% y los mínimos están sobre el 10%

Observamos ahora como se distribuyen las muertes por accidente.

```r
df1 <- accidentData %>%
group_by(accidentData$FATALS) %>%
dplyr::summarise(counts = n())
df1
```

```
## # A tibble: 8 x 2
##   `accidentData$FATALS` counts
##                   <int>  <int>
## 1                     1  33226
## 2                     2   2154
## 3                     3    289
## 4                     4     71
## 5                     5     20
## 6                     6      4
## 7                     7      1
## 8                     8      1
```

```r
counts <- table(accidentData$FATALS)
barplot(prop.table(counts),col=("red"),ylim=c(0,0.99),main="Distribución de la mortalidad a los acciden
```



Vemos que la mayoría de los accidentes tienen como mínimo un muerto. Vamos ahora a relacionar mortalidad y alcohol.

```r
counts <- table(accidentData$alcohol, accidentData$FATALS)
colors <- c("green", "red")
barplot(prop.table(counts), beside = TRUE, col = colors,
ylim = c(0, 1), axes = TRUE,
xlab = "Número de muertos",
ylab = "Porcentaje",
main = "Accidentes por muertes y conductores positivos en alcohol",
legend = c("No Alcohol", "Alcohol"),
```

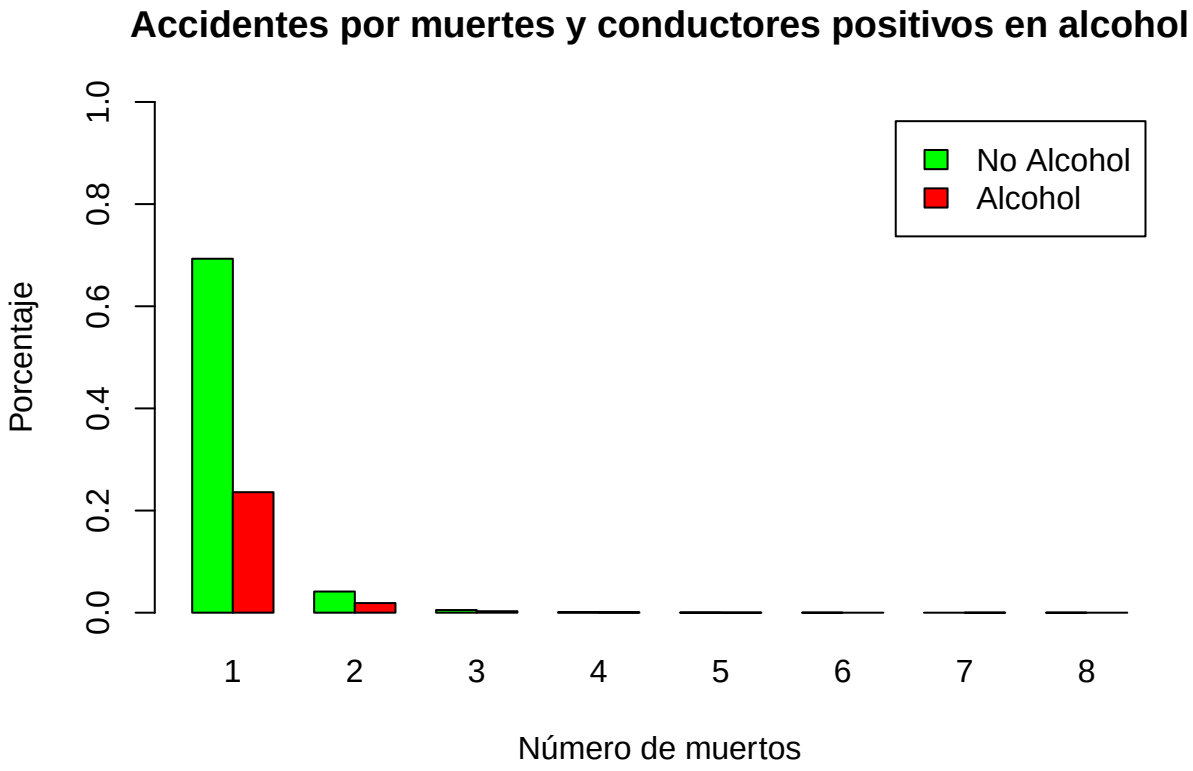
```
fill = colors)
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "fill" is not a graphical
## parameter
```

```
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : "fill" is not a graphical parameter
```

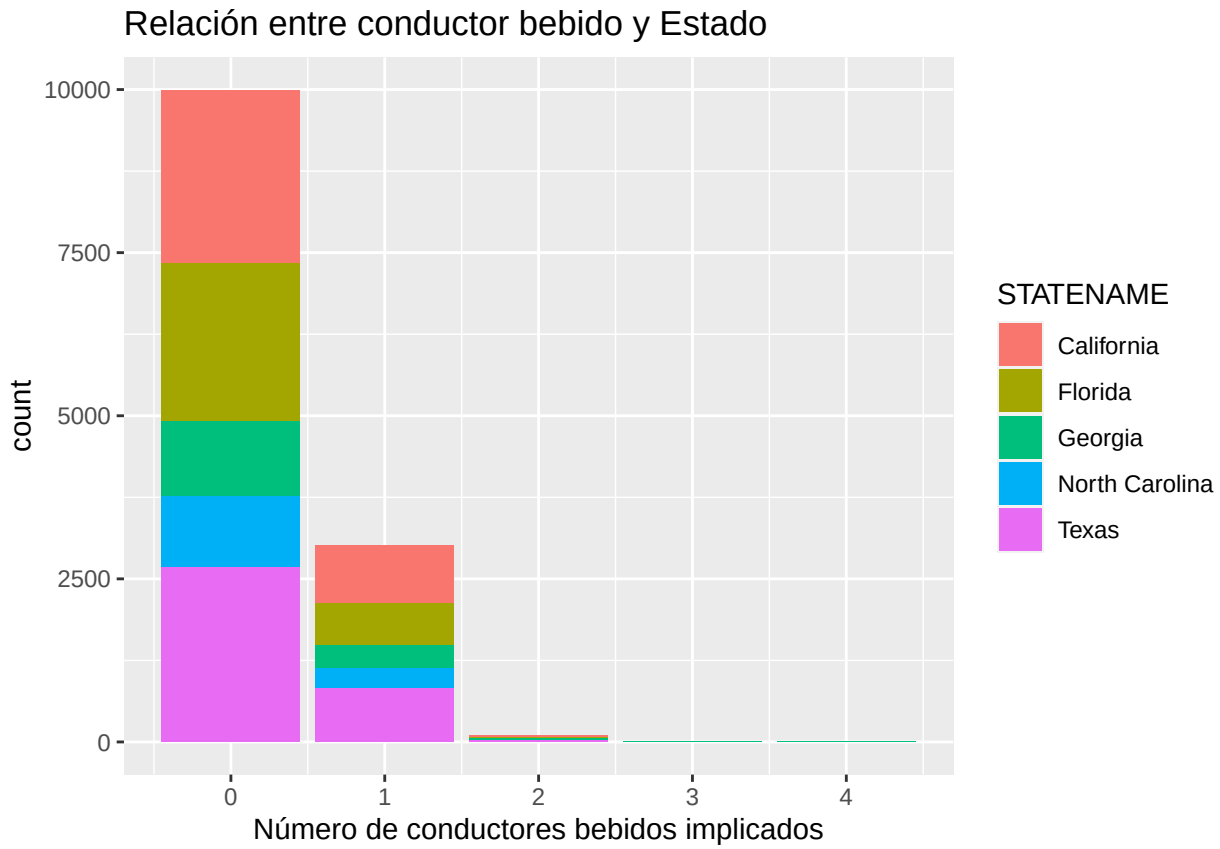
```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "fill"
## is not a graphical parameter
```

```
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): "fill" is not a
## graphical parameter
```



Probaremos ahora si hay relación entre el estado donde ha pasado el accidente y el número de conductores bebidos. Filtramos los cinco estados donde hay más accidentes.

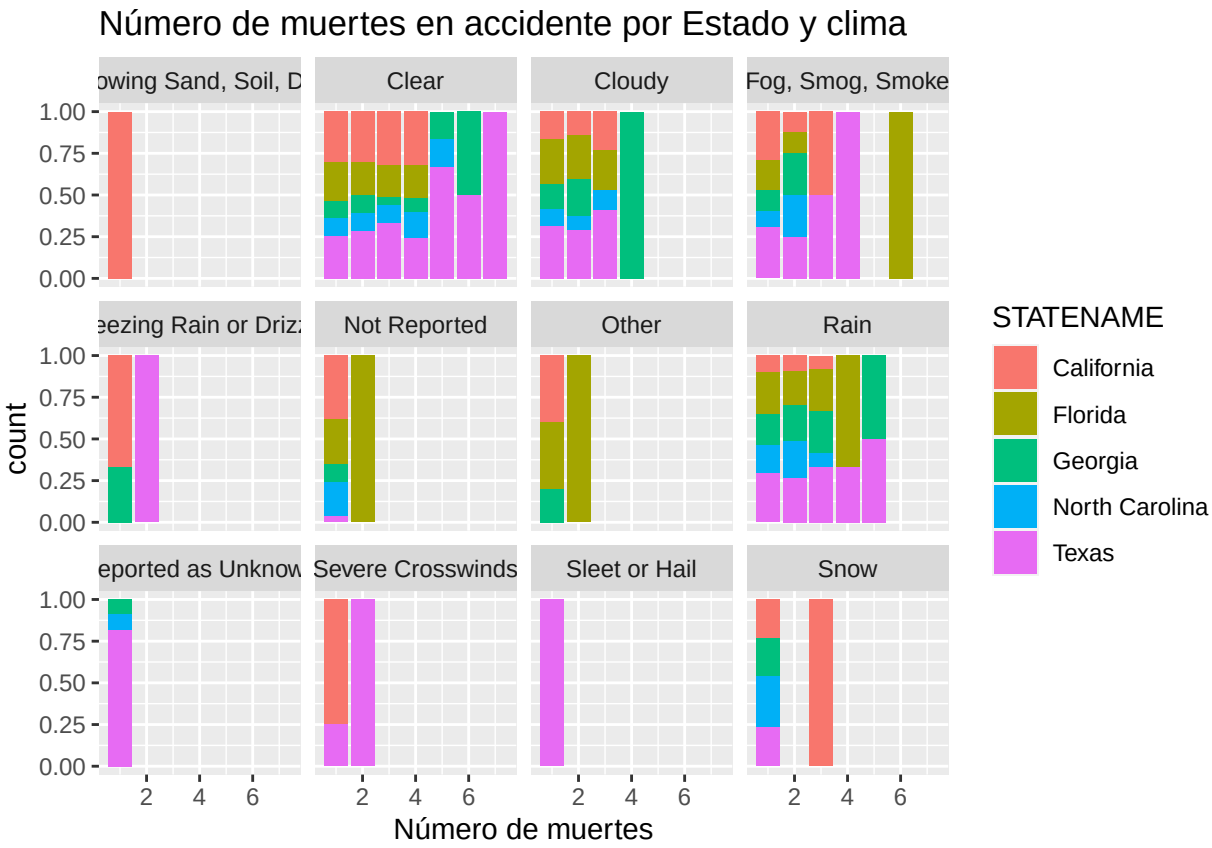
```
accidentDataST5=subset(accidentData, accidentData$STATENAME == "California" | accidentData$STATENAME ==

files=dim(accidentDataST5)[1]
ggplot(data=accidentDataST5[1:files,],aes(x=DRUNK_DR,fill=STATENAME))+geom_bar()+ggtitle("Relación entr
```



Como reflexión este gráfico tiene que pasar por el filtro de percentuar el número de accidentes por estado y la población del estado para no sacar conclusiones apresuradas.

Veamos ahora como en un mismo gráfico de frecuencias podemos trabajar con 3 variables: FATALS, STATENAME y WEATHERNAME.

```
ggplot(data = accidentDataST5[1:files,],aes(x=FATALS,fill=STATENAME))+geom_bar(position="fill")+facet_w
```



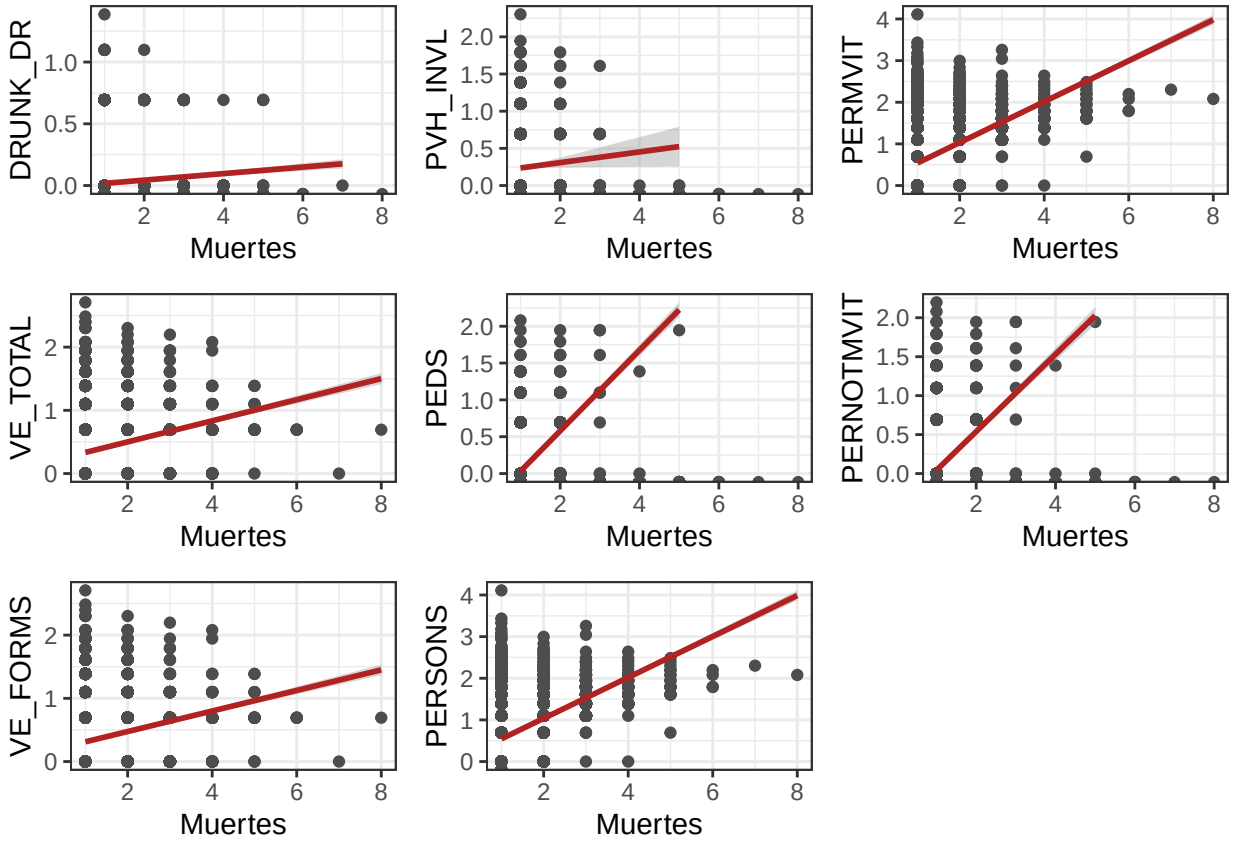
Esta gráfica está bien como mecánica de construcción, pero los resultados los ponemos en entredicho. Está bien como paso inicial, pero hay que profundizar mucho más.

Vamos a buscar las correlaciones en función de las muertes y unas variables elegidas que creemos que pueden ayudar a explicar el aumento de muertes por accidente:

DRUNK_DR conductores bebidos **VE_TOTAL** número de vehículos implicados en total **VE_FORMS** número de vehículos en movimiento implicados **PVH_INVL** número de vehículos estacionados implicados **PEDS** número de peatón implicados **PERSONS** número de ocupante de vehículo implicados **PERMVIT** número conductores y ocupantes implicados **PERNOTMVIT** número peatones, ciclistas... Cualquier cosa menos vehículo motorizado

```
# Utilizamos esta librería para usar la funcio multiplot()
if(!require('Rmisc')) install.packages('Rmisc'); library('Rmisc')

n = c("DRUNK_DR","VE_TOTAL","VE_FORMS","PVH_INVL","PEDS","PERSONS","PERMVIT","PERNOTMVIT")
accidentDataAux= accidentData %>% select(all_of(n))
histList2<- vector('list', ncol(accidentDataAux))
for(i in seq_along(accidentDataAux)){
  message(i)
histList2[[i]]<-local({
  i<-i
  col <-log(accidentDataAux[[i]])
  ggp<- ggplot(data = accidentDataAux, aes(x = accidentData$FATALS, y=col)) +
    geom_point(color = "gray30") + geom_smooth(method = lm,color = "firebrick") +
    theme_bw() + xlab("Muertes") + ylab(names(accidentDataAux)[i])
  })

}
```

```
multiplot(plotlist = histList2, cols = 3)
```

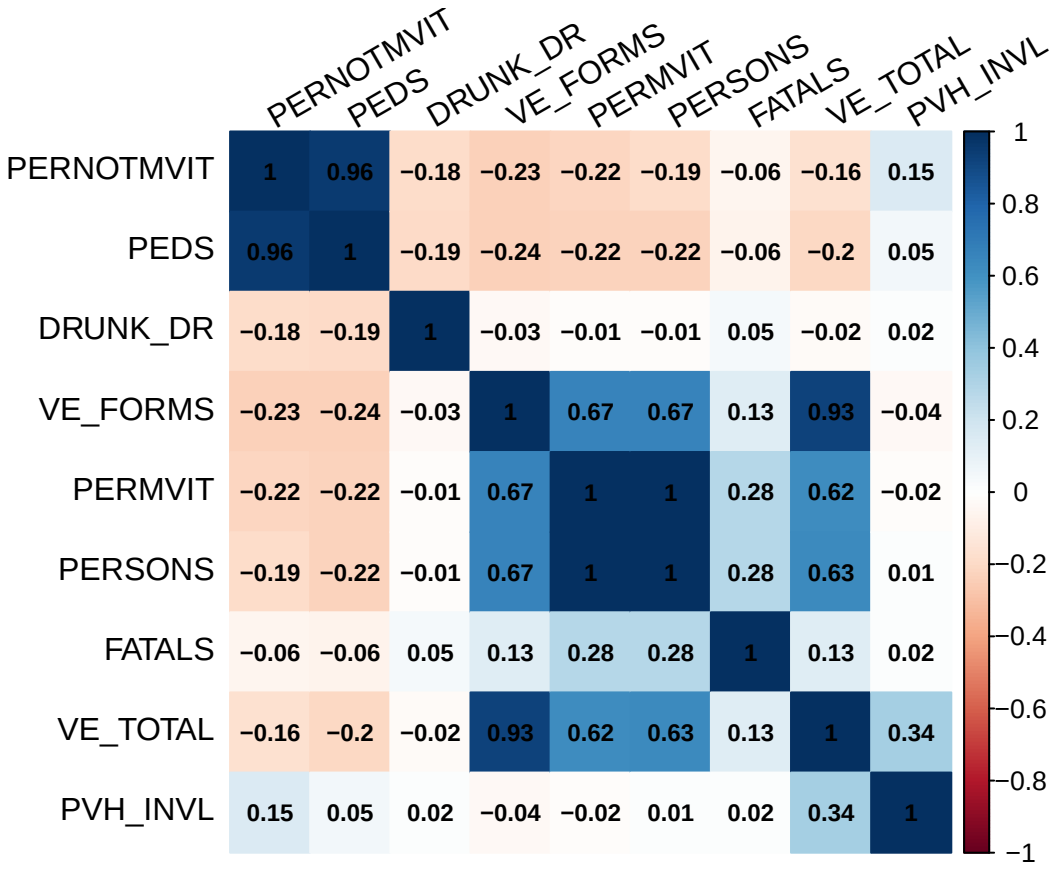


Podemos ver que:

- De forma general cualquier aumento en las variables elegidas implica un aumento de las muertes en el accidente.
- El factor que hace aumentar más el número de víctimas son las variables relacionadas con los peatones y pasajeros de los coches involucrados en el accidente.

Utilizamos las columnas que nos interesa para hacer la matriz y la visualizaremos utilizando la función corrplot.

```
# https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html
if(!require("corrplot")) install.packages("corrplot"); library("corrplot")
n = c("FATALS","DRUNK_DR","VE_TOTAL","VE_FORMS","PVH_INVL","PEDS","PERSONS","PERMVIT","PERNOTMVIT")
factores= accidentData %>% select(all_of(n))
res<-cor(factores)
corrplot(res,method="color",tl.col="black", tl.srt=30, order = "AOE",
number.cex=0.75,sig.level = 0.01, addCoef.col = "black")
```



No vemos que haya una correlación negativa significativa entre dos variables y sí una muy buena correlación ya previsible entre los peatones implicados y personas involucradas en el accidente que no van en coche. (PEDS y PERNOTMVIT) Lo mismo podemos observar en cuanto al número de conductores y ocupantes implicados (PERMVIT) y el número de vehículos implicados en movimiento (VE_FORMS) o el total de vehículos (VE_TOTAL).

Vamos a probar si hay una correlación entre personas implicadas en el accidente y el número de muertes.

```
if (!require('tidyverse')) install.packages('tidyverse'); library('tidyverse')
```

```
## Loading required package: tidyverse
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## v readr     2.1.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::desc()      masks plyr::desc()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
```

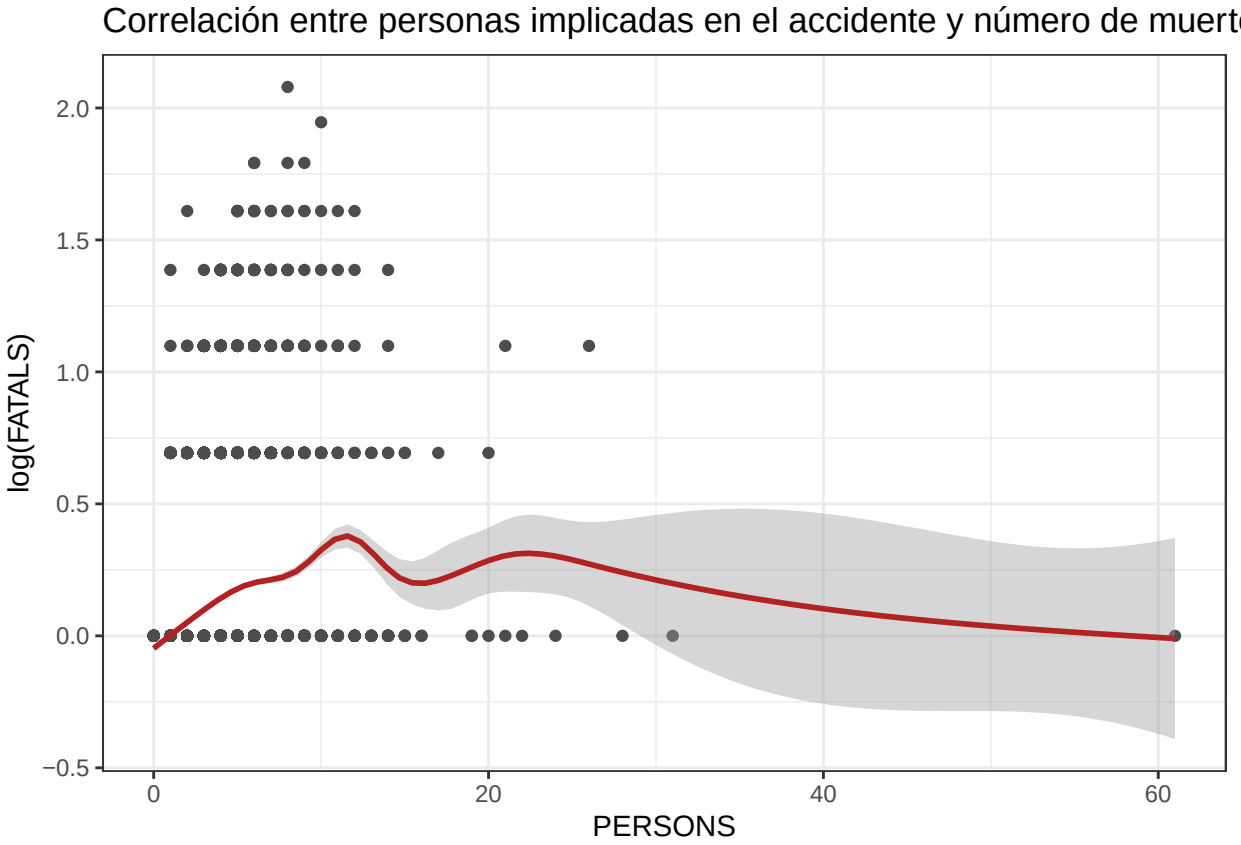
```
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
cor.test(x = accidentData$PERSONS, y = accidentData$FATALS, method = "kendall")
```

```
##
##  Kendall's rank correlation tau
##
## data:  accidentData$PERSONS and accidentData$FATALS
## z = 53.008, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.2558174
```

```
ggplot(data = accidentData, aes(x = PERSONS, y = log(FATALS))) + geom_point(color = "gray30") + geom_sm
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



De la observación de este gráfico podemos concluir que efectivamente el número de muertes aumenta en función de las personas implicadas en un accidente pero que la correlación no es tan elevada ni continúa como se podía prever.

Construcción de conjunto de datos final

Si dos variables están altamente correlacionadas obviamente darán casi exactamente la misma información en un modelo de regresión, por ejemplo. Pero, al incluir las dos variables, en realidad estamos debilitando el modelo. No estamos añadiendo información incremental. En lugar de esto, estamos haciendo un modelo

ruidoso. No es una buena idea.

Cómo hemos visto antes tenemos una correlación muy grande entre PEDS y PERNOTMVIT, por lo tanto, podríamos eliminar la columna de peatones (PEDS) y dejar el total de peatones y otros reflejado a PERNOTMVIT.

```
# accidentData$PEDS<- NULL
str(accidentData)
```

```
## 'data.frame':    35766 obs. of  82 variables:
##  $ STATE       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ STATENAME   : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ ST_CASE     : int  10001 10002 10003 10004 10005 10006 10007 10008 10009 10010 ...
##  $ VE_TOTAL    : int  1 4 2 1 1 2 1 2 2 2 ...
##  $ VE_FORMS    : int  1 4 2 1 1 2 1 2 2 2 ...
##  $ PVH_INVL    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PEDS        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ PERSONS     : int  4 6 2 5 1 3 1 2 4 3 ...
##  $ PERMVIT     : int  4 6 2 5 1 3 1 2 4 3 ...
##  $ PERNOTMVIT  : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ COUNTY      : int  51 73 117 15 37 103 73 25 45 95 ...
##  $ COUNTYNAME  : chr  "ELMORE (51)" "JEFFERSON (73)" "SHELBY (117)" "CALHOUN (15)" ...
##  $ CITY        : int  0 350 0 0 0 0 330 0 0 1500 ...
##  $ CITYNAME    : chr  "NOT APPLICABLE" "BIRMINGHAM" "NOT APPLICABLE" "NOT APPLICABLE" ...
##  $ DAY         : int  1 2 2 3 4 4 7 8 9 10 ...
##  $ DAYNAME     : int  1 2 2 3 4 4 7 8 9 10 ...
##  $ MONTH       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ MONTHNAME   : chr  "January" "January" "January" "January" ...
##  $ YEAR        : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ DAY_WEEK    : int  4 5 5 6 7 7 3 4 5 6 ...
##  $ DAY_WEEKNAME: chr  "Wednesday" "Thursday" "Thursday" "Friday" ...
##  $ HOUR        : int  2 17 14 15 0 16 19 7 20 10 ...
##  $ HOURNAME    : chr  "2:00am-2:59am" "5:00pm-5:59pm" "2:00pm-2:59pm" "3:00pm-3:59pm" ...
##  $ MINUTE      : int  58 18 55 20 45 55 23 15 0 2 ...
##  $ MINUTENAME  : chr  "58" "18" "55" "20" ...
##  $ NHS         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ NHSNAME     : chr  "This section IS NOT on the NHS" "This section IS NOT on the NHS" "This sectio
##  $ ROUTE       : int  4 6 3 4 4 3 4 4 4 2 ...
##  $ ROUTENAME   : chr  "County Road" "Local Street - Municipality" "State Highway" "County Road" ...
##  $ TWAY_ID     : chr  "cr-4" "martin luther king jr dr" "sr-76" "CR-ALEXANDRIA WELLINGTON RD" ...
##  $ TWAY_ID2    : chr  "" "" "us-280" "" ...
##  $ RUR_URB     : int  1 2 1 1 1 1 2 1 1 1 ...
##  $ RUR_URBNAME : chr  "Rural" "Urban" "Rural" "Rural" ...
##  $ FUNC_SYS    : int  5 4 4 7 5 4 4 5 5 3 ...
##  $ FUNC_SYSNAME: chr  "Major Collector" "Minor Arterial" "Minor Arterial" "Local" ...
##  $ RD_OWNER    : int  2 4 1 2 2 1 4 2 2 1 ...
##  $ RD_OWNERNAME: chr  "County Highway Agency" "City or Municipal Highway Agency" "State Highway Agen
##  $ MILEPT      : int  0 0 49 0 0 390 0 0 0 3019 ...
##  $ MILEPTNAME  : chr  "None" "None" "49" "None" ...
##  $ LATITUDE    : num  32.4 33.5 33.3 33.8 32.8 ...
##  $ LATITUDENAME: chr  "32.43313333" "33.48465833" "33.29994167" "33.79507222" ...
##  $ LONGITUD    : num  -86.1 -86.8 -86.4 -85.9 -86.1 ...
##  $ LONGITUDNAME: chr  "-86.09485" "-86.83954444" "-86.36964167" "-85.88348611" ...
##  $ SP_JUR      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SP_JURNAME  : chr  "No Special Jurisdiction" "No Special Jurisdiction" "No Special Jurisdiction"
```

```
##  $ HARM_EV     : int  42 12 34 42 42 12 8 12 12 12 ...
##  $ HARM_EVNAME : chr  "Tree (Standing Only)" "Motor Vehicle In-Transport" "Ditch" "Tree (Standing Onl
##  $ MAN_COLL    : int  0 6 0 0 0 2 0 1 1 2 ...
##  $ MAN_COLLNAME: chr  "The First Harmful Event was Not a Collision with a Motor Vehicle in Transport
##  $ RELJCT1     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RELJCT1NAME : chr  "No" "No" "No" "No" ...
##  $ RELJCT2     : int  1 1 3 1 1 1 3 1 8 1 ...
##  $ RELJCT2NAME : chr  "Non-Junction" "Non-Junction" "Intersection-Related" "Non-Junction" ...
##  $ TYP_INT     : int  1 1 3 1 1 1 2 1 1 1 ...
##  $ TYP_INTNAME : chr  "Not an Intersection" "Not an Intersection" "T-Intersection" "Not an Intersecti
##  $ WRK_ZONE    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WRK_ZONENAME: chr  "None" "None" "None" "None" ...
##  $ REL_ROAD    : int  4 1 4 4 4 1 1 1 1 1 ...
##  $ REL_ROADNAME: chr  "On Roadside" "On Roadway" "On Roadside" "On Roadside" ...
##  $ LGT_COND    : int  2 3 1 1 2 2 3 1 2 1 ...
##  $ LGT_CONDNAME: chr  "Dark - Not Lighted" "Dark - Lighted" "Daylight" "Daylight" ...
##  $ WEATHER     : int  1 2 2 10 2 1 1 1 10 10 ...
##  $ WEATHERNAME : chr  "Clear" "Rain" "Rain" "Cloudy" ...
##  $ SCH_BUS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SCH_BUSNAME : chr  "No" "No" "No" "No" ...
##  $ RAIL        : chr  "0000000" "0000000" "0000000" "0000000" ...
##  $ RAILNAME    : chr  "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
##  $ NOT_HOUR    : int  99 17 14 99 0 17 19 7 20 10 ...
##  $ NOT_HOURNAME: chr  "Unknown" "5:00pm-5:59pm" "2:00pm-2:59pm" "Unknown" ...
##  $ NOT_MIN     : int  99 18 58 99 45 0 23 21 0 3 ...
##  $ NOT_MINNAME : chr  "Unknown" "18" "58" "Unknown" ...
##  $ ARR_HOUR    : int  3 17 15 99 0 17 19 7 20 10 ...
##  $ ARR_HOURNAME: chr  "3:00am-3:59am" "5:00pm-5:59pm" "3:00pm-3:59pm" "Unknown EMS Scene Arrival Hou
##  $ ARR_MIN     : int  10 26 15 99 55 19 29 28 10 7 ...
##  $ ARR_MINNAME : chr  "10" "26" "15" "Unknown EMS Scene Arrival Minutes" ...
##  $ HOSP_HR     : int  99 99 99 99 88 18 88 88 99 10 ...
##  $ HOSP_HRNAME : chr  "Unknown" "Unknown" "Unknown" "Unknown" ...
##  $ HOSP_MN     : int  99 99 99 99 88 51 88 88 99 29 ...
##  $ HOSP_MNNAME : chr  "Unknown EMS Hospital Arrival Time" "Unknown EMS Hospital Arrival Time" "Unknou
##  $ FATALS      : int  3 1 1 1 1 1 1 1 1 1 ...
##  $ DRUNK_DR    : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ alcohol     : num  1 0 0 0 0 0 0 0 0 0 ...
```

Codificación

Seguidamente vayamos a asignar un 1 por accidentes que se producen de madrugada (01h a 06h en invierno) y un 0 para el resto de franja horaria, es decir, vamos a categorizar la variable HOUR y así tendremos una variable numérica que nos permitirá trabajar mejor en el futuro. La denominaremos madrugada. Después la utilizaremos para ver cómo se distribuyen los accidentes en las dos franjas horarias. Debemos tener en cuenta que la hora incluye el código 99 qué quiere decir que la hora no está informada. Miraremos de filtrar los registros con este valor para excluirlos.

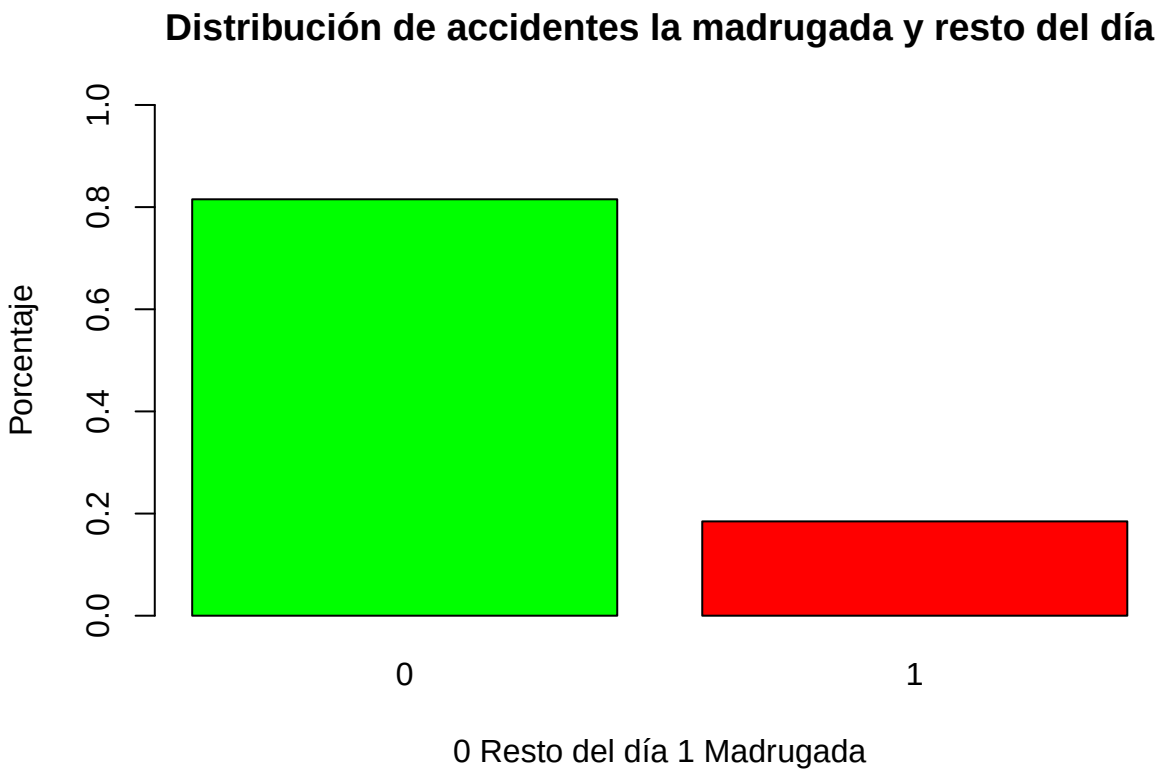
```
accidentDataAux=subset(accidentData, accidentData$HOUR <= 24)

accidentData$madrugada <- NA
accidentData$madrugada[accidentDataAux$HOUR >=1 & accidentDataAux$HOUR <= 6] <- 1
accidentData$madrugada[accidentDataAux$HOUR ==0 | accidentDataAux$HOUR >6 ] <- 0

counts <- table(accidentData$madrugada)
barplot(prop.table(counts),col=c("green","red"),legend.texto=c("Resto del día","Madrugada"),ylim=c(0,1)
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "legend.texto" is not a
## graphical parameter
```

```
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : "legend.texto" is not a graphical parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "legend.texto" is not a graphical parameter
```

```
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): "legend.texto"
## is not a graphical parameter
```



Discretización

Ahora añadiremos un campo nuevo a los datos. Este campo contendrá el valor de la hora del accidente discretizada con un método simple de intervalos de igual amplitud.

```
summary(accidentDataAux[,"HOUR"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    7.00   15.00   13.19   19.00   23.00
```

Discretizamos con intervalos. Los criterios de corte están cogidos de la Web del Parlamento de Cataluña.

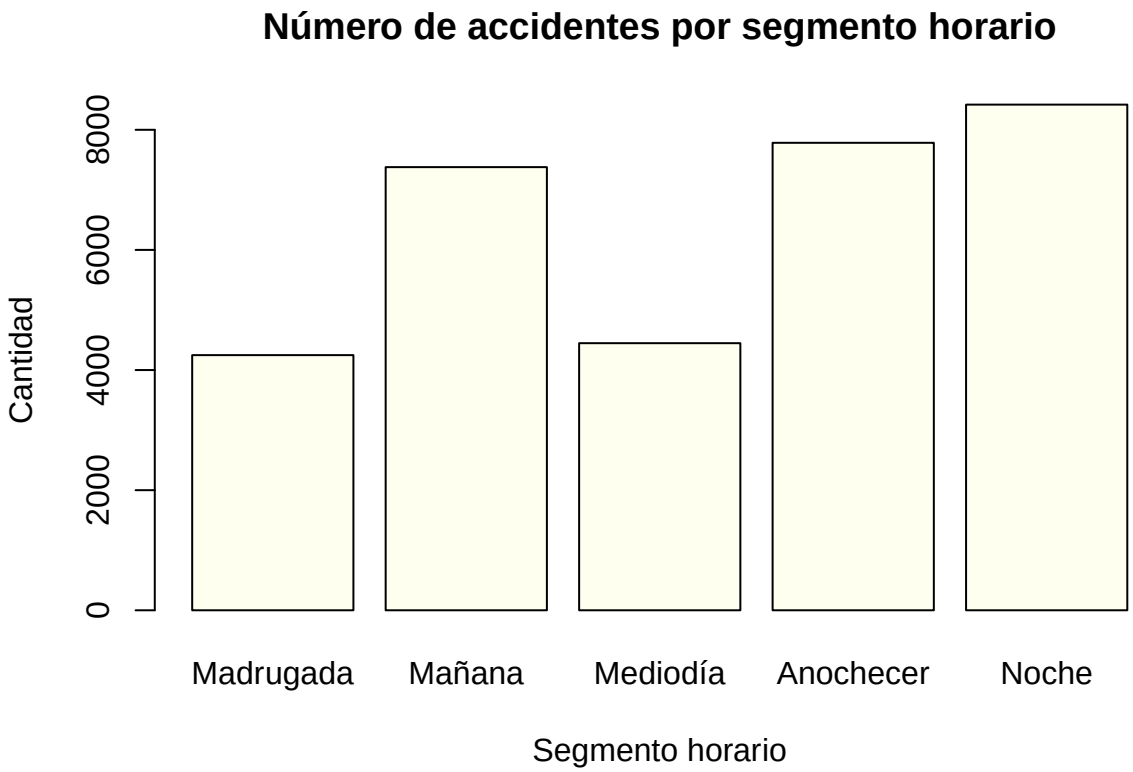
```
accidentDataAux["segmento_horario"] <- cut(accidentDataAux$HOUR, breaks = c(0,4,11,14,18,22), labels =
```

Observamos los datos discretizados y construimos un gráfico para analizar cómo se agrupan los accidentes.

```
head(accidentDataAux$segmento_horario)
```

```
## [1] Madrugada Anochecer Mediodía  Anochecer <NA>      Anochecer
## Levels: Madrugada Mañana Mediodía Anochecer Noche
```

```r
plot(accidentDataAux$segmento_horario,main="Número de accidentes por segmento horario",xlab="Segmento ho
```



Ahora vamos a discretizar la variable que contiene el número de vehículos implicados en un accidente (VE_TOTALS) puesto que era una de las variables en que las distancias entre sus valores eran muy grandes:

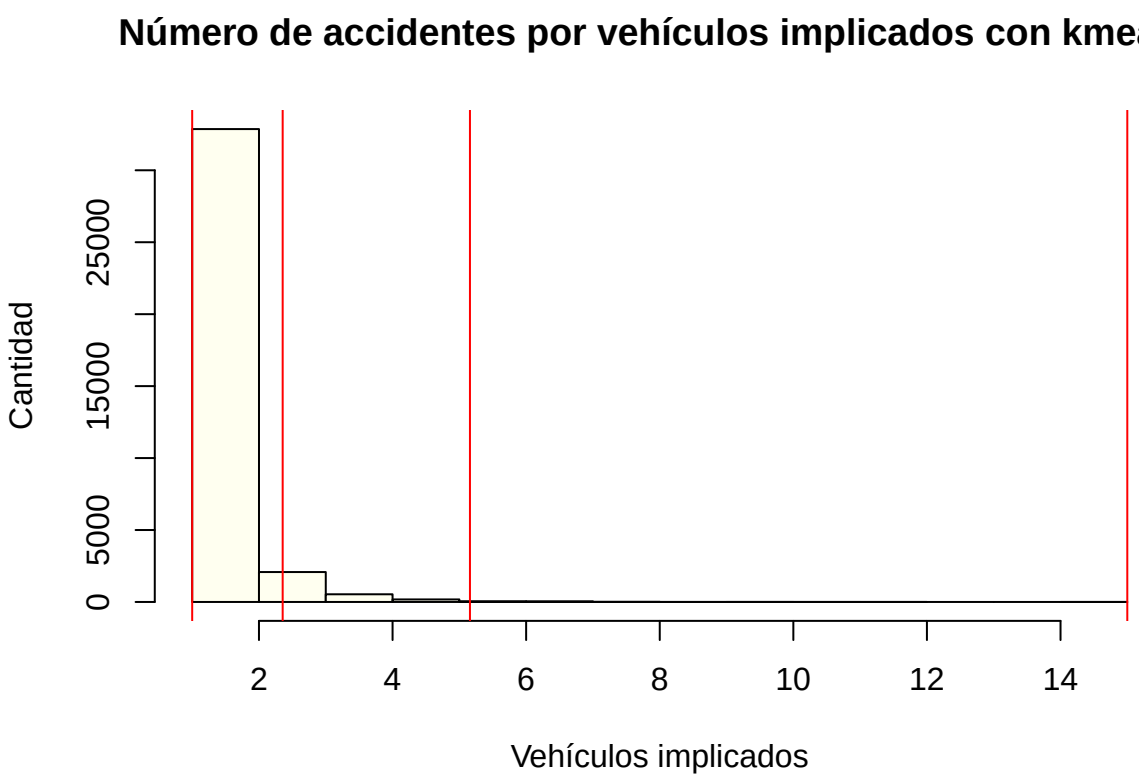
```r
# Utilizaremos la función discretize de arules: This function implements several basic unsupervised meth
# https://cran.r-project.org/web/packages/arules/index.html
if (!require('arules')) install.packages('arules'); library('arules')
```

```
## Loading required package: arules
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Attaching package: 'arules'
## The following object is masked from 'package:dplyr':
##
##     recode
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```r
set.seed(2)
table(discretize(accidentData$VE_TOTAL, "cluster" ))
```

```
##
##    [1,1.57) [1.57,3.38)   [3.38,15]
##       19972       14972         822
```

```
hist(accidentData$VE_TOTAL, main="Número de accidentes por vehículos implicados con kmeans",xlab="Vehícu
abline(v=discretize(accidentData$VE_TOTAL, method="cluster", onlycuts=TRUE),col="red")
```



Podemos observar que sin pasar ningún argumento y permitiendo que el algoritmo elija el conjunto de particiones se muestran tres clústeres que agrupan los vehículos implicados en las franjas mencionadas. Podemos asignar el propio clúster como una variable más al dataset para trabajar después.

```
accidentData$VE_TOTAL_KM<- (discretize(accidentData$VE_TOTAL, "cluster" ))
head(accidentData$VE_TOTAL_KM)
```

```
## [1] [1,1.5)    [2.73,15]  [1.5,2.73) [1,1.5)    [1,1.5)    [1.5,2.73)
## Levels: [1,1.5) [1.5,2.73) [2.73,15]
```

Normalización

Ahora normalizaremos el número de muertes por el máximo añadiendo un nuevo campo a los datos que contendrá el valor.

```
accidentData$FATALS_NM<- (accidentData$FATALS/max(accidentData[,"FATALS"]))
head(accidentData$FATALS_NM)
```

```
## [1] 0.375 0.125 0.125 0.125 0.125 0.125
```

Supongamos que queremos normalizar por la diferencia para ubicar entre 0 y 1 la variable del número de muertes del accidente dado que el algoritmo de minería que utilizaremos así lo requiere. Observamos la distribución de la variable original y las generadas.

```
accidentData$FATALS_ND = (accidentData$FATALS-min(accidentData$FATALS))/(max(accidentData$FATALS)-min(a
```

```r
max(accidentData$FATALS)
```

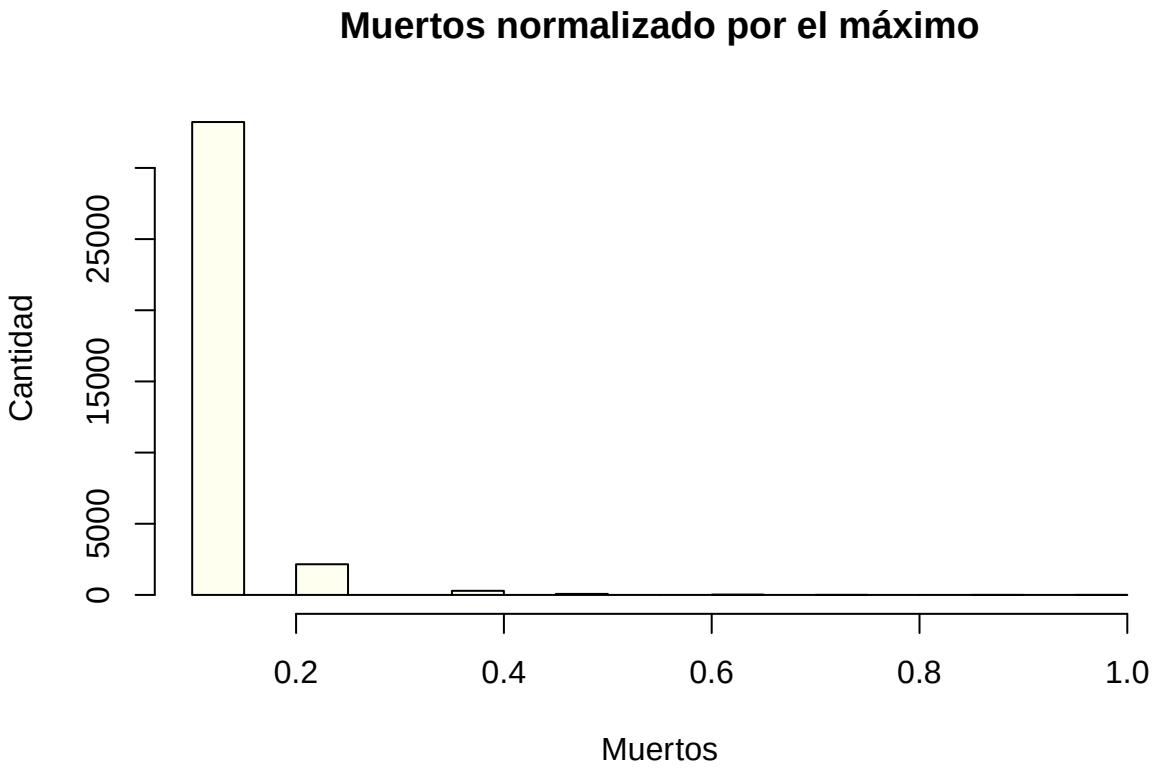
```
## [1] 8
```

```r
min(accidentData$FATALS)
```

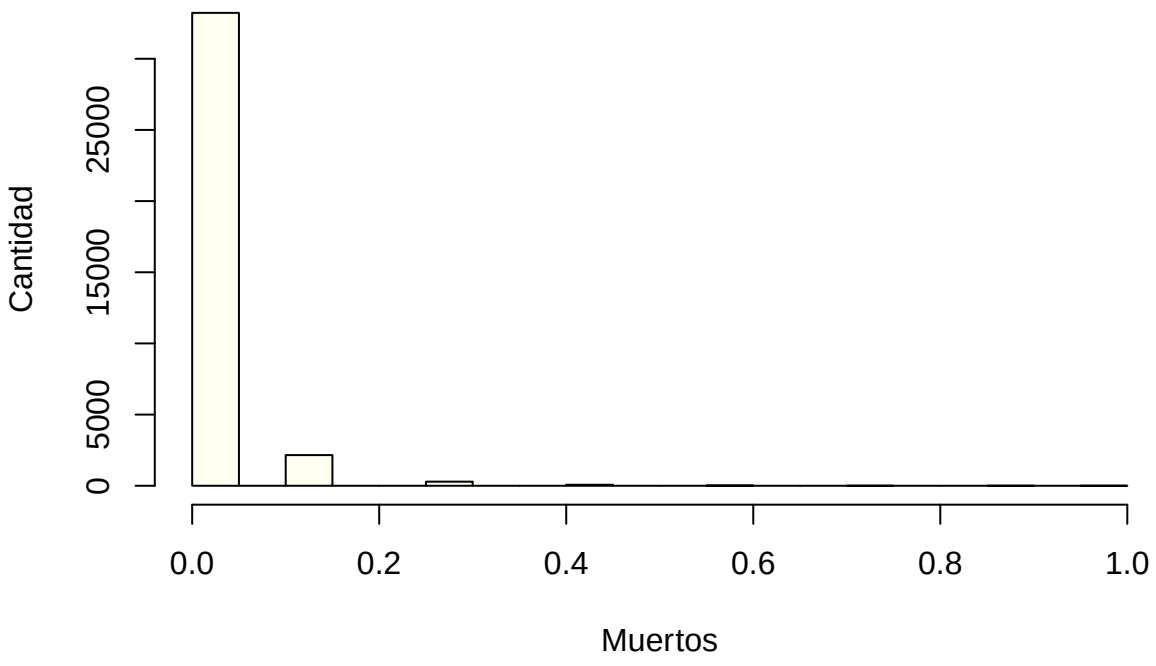
```
## [1] 1
```

```r
hist(accidentData$FATALS,xlab="Muertos", col="ivory",ylab="Cantidad", main="Número de muertes en accide
```


```r
hist(accidentData$FATALS_NM,xlab="Muertos",ylab="Cantidad",col="ivory", main="Muertos normalizado por el
```




```
hist(accidentData$FATALS_ND,xlab="Muertos",ylab="Cantidad", col="ivory", main="Muertos normalizado por l
```



A continuación, vamos a normalizar las otras columnas para asegurarnos que cada variable contribuye por igual en nuestro análisis.

```r
# Definimos la función de normalización
nor <-function(x) { (x -min(x))/(max(x)-min(x))}
# Guardamos un nuevo dataset normalizado

accidentData$type<- NULL
n = c("FATALS","DRUNK_DR","VE_TOTAL","VE_FORMS","PVH_INVL","PEDS","PERSONS","PERMVIT","PERNOTMVIT")
accidentData<- accidentData %>% select(all_of(n))
accidentData_nor <- as.data.frame(lapply(accidentData, nor))

head(accidentData_nor)
```

```
##      FATALS DRUNK_DR   VE_TOTAL   VE_FORMS PVH_INVL PEDS    PERSONS    PERMVIT
## 1 0.2857143     0.25 0.00000000 0.00000000        0    0 0.06557377 0.06557377
## 2 0.0000000     0.00 0.21428571 0.21428571        0    0 0.09836066 0.09836066
## 3 0.0000000     0.00 0.07142857 0.07142857        0    0 0.03278689 0.03278689
## 4 0.0000000     0.00 0.00000000 0.00000000        0    0 0.08196721 0.08196721
## 5 0.0000000     0.00 0.00000000 0.00000000        0    0 0.01639344 0.01639344
## 6 0.0000000     0.00 0.07142857 0.07142857        0    0 0.04918033 0.04918033
##   PERNOTMVIT
## 1          0
## 2          0
## 3          0
## 4          0
## 5          0
## 6          0
```

Proceso de PCA

Tanto el análisis de componentes principales, principal componente analysis (PCA) en inglés, como la descomposición de valores singulares, singular value decomposition (SVD) en inglés, son técnicas que nos permitan trabajar con nuevas características llamadas componentes, que ciertamente son independientes entre sí. En realidad, estas dos técnicas nos permiten representar el juego de datos en un nuevo sistema de coordenadas que denominamos componentes principales. Este sistema está mejor adaptado a la distribución del juego de datos, de forma que recoge mejor su variabilidad.

Aplicamos el análisis de componentes principales al dataset. Empezamos ejecutando la función **prcomp()**.

```r
pca.acc <- prcomp(accidentData_nor)
summary(pca.acc)
```

```
## Importance of components:
##                           PC1     PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     0.1168 0.08755 0.0690 0.0484 0.03269 0.02437 0.01072
## Proportion of Variance 0.4520 0.25389 0.1577 0.0776 0.03539 0.01967 0.00381
## Cumulative Proportion  0.4520 0.70584 0.8635 0.9411 0.97652 0.99619 1.00000
##                              PC8       PC9
## Standard deviation     5.213e-15 2.159e-15
## Proportion of Variance 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.000e+00
```

Como se puede observar la función summary, nos devuelve la proporción de varianza aplicada al conjunto total de cada atributo. Gracias a esto, el atributo 1 explica el 0.452 de variabilidad del total de datos; en cambio, el atributo 7 explica solo el 0.000381.

A continuación, se muestra un histograma para ver el peso de cada atributo sobre el conjunto total de datos:

```r
if (!require('factoextra')) install.packages('factoextra'); library('factoextra')
```

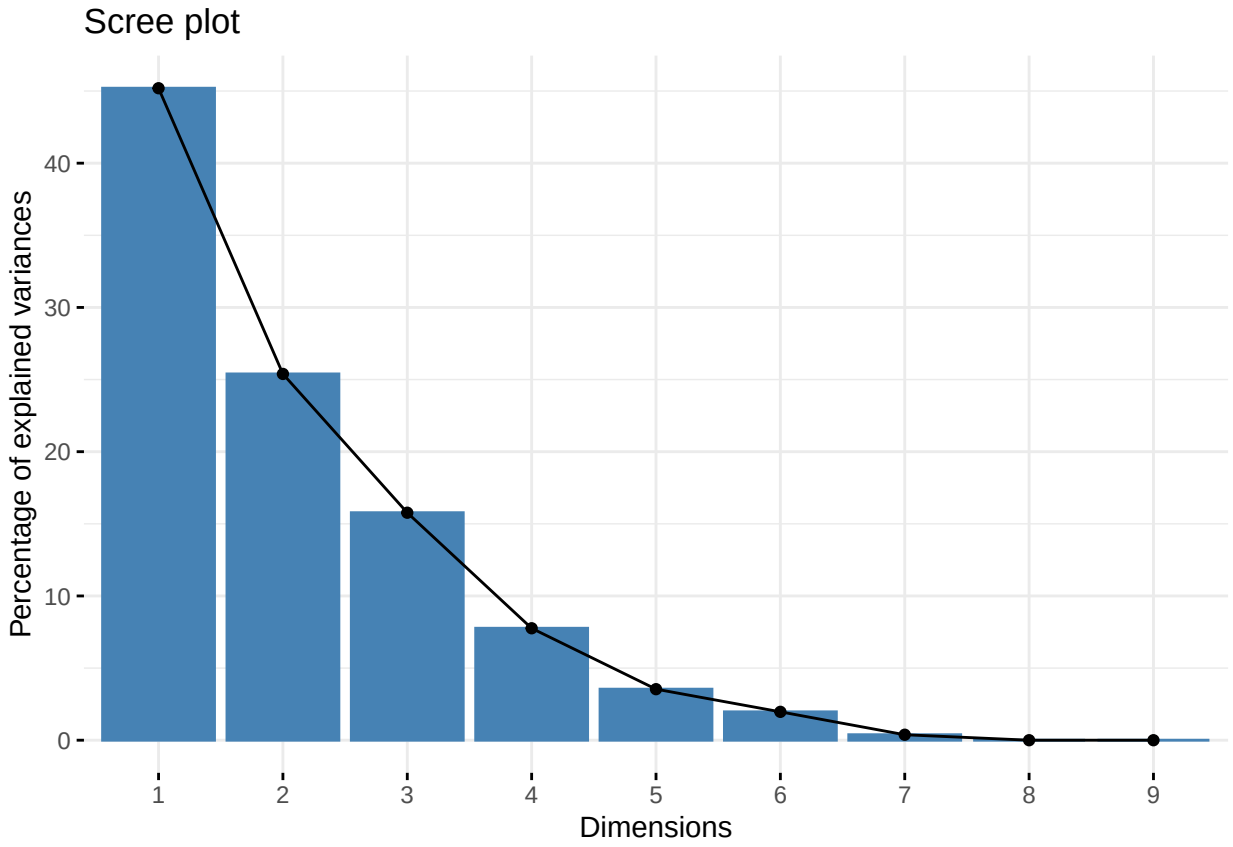
```
## Loading required package: factoextra
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
#Los valores propios corresponden a la cantidad de variación explicada por cada componente principal (P
ev= get_eig(pca.acc)
ev
```

```
##         eigenvalue variance.percent cumulative.variance.percent
## Dim.1 1.364366e-02     4.519469e+01                    45.19469
## Dim.2 7.664656e-03     2.538921e+01                    70.58391
## Dim.3 4.760727e-03     1.576993e+01                    86.35384
## Dim.4 2.342530e-03     7.759642e+00                    94.11348
## Dim.5 1.068327e-03     3.538839e+00                    97.65232
## Dim.6 5.937915e-04     1.966938e+00                    99.61926
## Dim.7 1.149407e-04     3.807416e-01                   100.00000
## Dim.8 2.717197e-29     9.000729e-26                   100.00000
## Dim.9 4.659472e-30     1.543453e-26                   100.00000
```

```r
fviz_eig(pca.acc)
```



En este ejercicio se decidió utilizar el método de Káiser para decidir cuales de las variables obtenidas serán escogidas. Este criterio mantendrá todas aquellas variables cuya varianza sea superior a 1.

```r
# Calculamos la varianza de los componentes principales a partir de la desviación estándar

var_acc <- pca.acc$sdev^2
```

```
var_acc
```

```
## [1] 1.364366e-02 7.664656e-03 4.760727e-03 2.342530e-03 1.068327e-03
## [6] 5.937915e-04 1.149407e-04 2.717197e-29 4.659472e-30
```

Con los resultados obtenidos es muy complicado decidir cuáles son los componentes principales componentes a escoger. *Este hecho podría estar causado por no haber escalado los datos previamente.* Por lo tanto, el siguiente paso es escalar los datos y volver a calcular la varianza para ver qué datos selecciona.

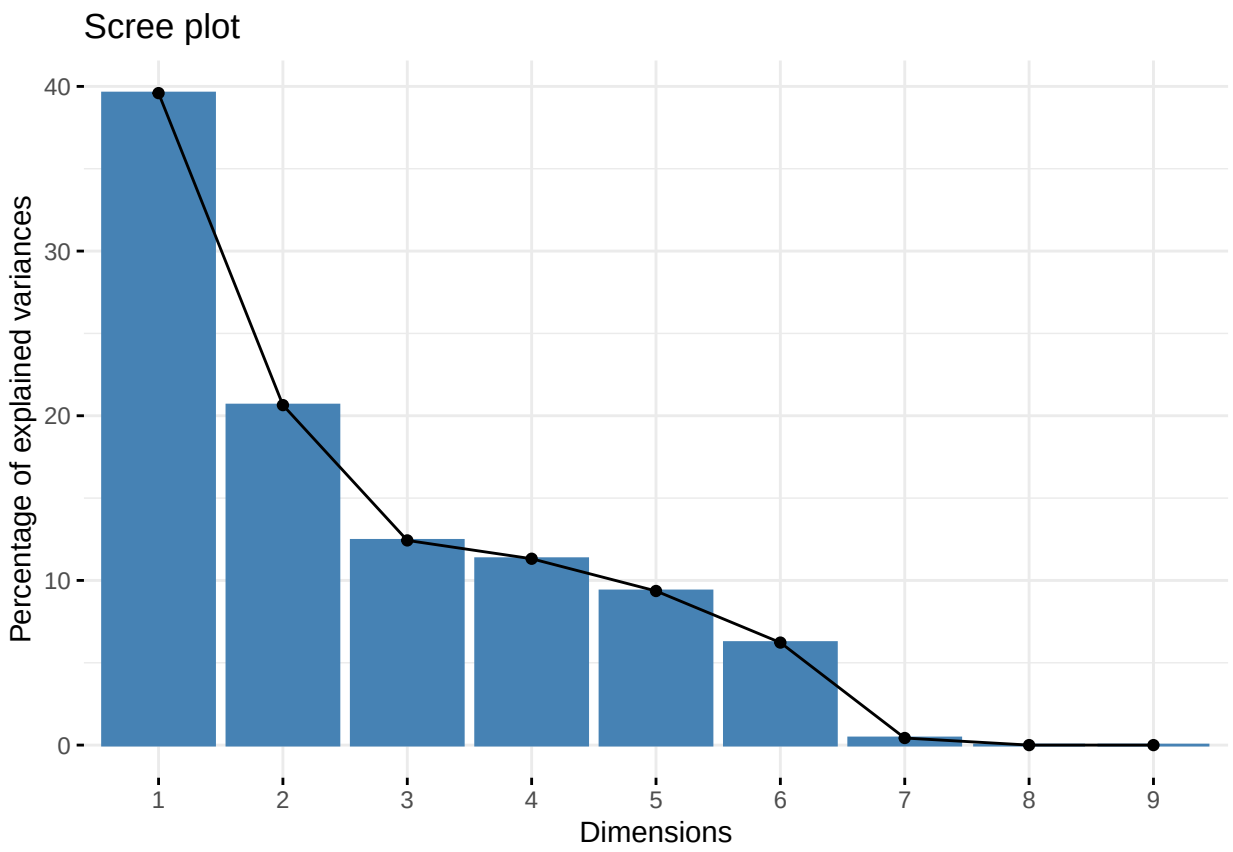
```
# Escalamos los datos
acc_scale <- scale(accidentData_nor)
# Calculamos las componentes principales
pca.acc_scale <- prcomp(acc_scale)
# Mostramos la varianza de dichas variables:
var_acc_scale <- pca.acc_scale$sdev^2
head(var_acc_scale)
```

```
## [1] 3.5632826 1.8581532 1.1186798 1.0185157 0.8423082 0.5603141
```

Después de analizar la varianza y aplicando el criterio de Káiser nos quedaremos con los componentes principales 1,2,3 y 4 que son los superiores a 1. Este criterio tiene el problema de sobreestimar el número de factores, pero a pesar de ello es el que aplicaremos para analizar los resultados.

Mostramos el histograma de porcentaje de varianza explicado con los datos escalados:

```
fviz_eig(pca.acc_scale)
```




```
ev = get_eig(pca.acc_scale)
ev
```

```
##         eigenvalue variance.percent cumulative.variance.percent
```

```
## Dim.1 3.563283e+00     3.959203e+01                 39.59203
## Dim.2 1.858153e+00     2.064615e+01                 60.23817
## Dim.3 1.118680e+00     1.242978e+01                 72.66795
## Dim.4 1.018516e+00     1.131684e+01                 83.98479
## Dim.5 8.423082e-01     9.358980e+00                 93.34377
## Dim.6 5.603141e-01     6.225712e+00                 99.56948
## Dim.7 3.874645e-02     4.305161e-01                100.00000
## Dim.8 4.353003e-26     4.836670e-25                100.00000
## Dim.9 2.164538e-26     2.405042e-25                100.00000
```

Los valores propios se pueden utilizar para determinar el número de componentes principales a retener después de la PCA (Kaiser 1961):

- Un valor propio > 1 indica que los PCs representan más varianza de la que representa una de las variables originales de los datos estandarizados. Esto se utiliza habitualmente como punto de corte para el cual se conservan los PCs. Esto solo es cierto cuando los datos están estandarizados.
- También podemos limitar el número de componentes a este número que representa una determinada fracción de la varianza total. Por ejemplo, si estamos satisfecho con el 80% de la varianza total explicada, usamos el número de componentes para conseguirlo que son los 4 componentes principales vistos antes.

Continuamos con el análisis de los componentes principales. Después de aplicar el método Káiser se han seleccionado los 4 componentes principales.

```
var <- get_pca_var(pca.acc_scale)
var
```

```
## Principal Component Analysis Results for variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates for the variables"
## 2 "$cor"     "Correlations between variables and dimensions"
## 3 "$cos2"    "Cos2 for the variables"
## 4 "$contrib" "contributions of the variables"
```

Los componentes de get_pca_var() se pueden utilizar en el diagrama de variables de la siguiente manera:

- **var$coord**: coordenadas de variables para crear un diagrama de dispersión.
- **var$cos2**: representa la calidad de representación de las variables al mapa de factores. Se calcula como las coordenadas al cuadrado: var.cos2 = var.coord * var.coord.
- **var$contrib**: contiene las contribuciones (en porcentaje) de las variables a los componentes principales. La contribución de una variable (var) a un determinado componente principal es (en porcentaje): (var.cos2 * 100) / (cos2 total del componente).

```
#Utilizamos los 4 componentes principales encontrados antes
head(var$coord[,1:4],11)
```

```
##                  Dim.1       Dim.2       Dim.3       Dim.4
## FATALS     -0.30725642  0.07606945  0.28817514 -0.73967833
## DRUNK_DR   -0.04921565 -0.38170799 -0.26454706 -0.58497489
## VE_TOTAL   -0.83708697  0.28706095 -0.34390477  0.13727724
## VE_FORMS   -0.86483356  0.18416941 -0.02398033  0.21866655
## PVH_INVL   -0.06062570  0.30358100 -0.85844575 -0.18336810
## PEDS        0.48249027  0.82661079  0.13673775 -0.08353380
## PERSONS    -0.88514448  0.21755791  0.19732665 -0.07582156
## PERMVIT    -0.88736748  0.19749105  0.22352113 -0.06716534
## PERNOTMVIT  0.45872440  0.85355861  0.04773257 -0.10804664
```

Calidad de representación

La calidad de representación de las variables en el mapa de factores se denomina cos2 (coseno cuadrado, coordenadas cuadradas). Podemos acceder al cos2 de la siguiente manera:

```
head(var$cos2[,1:4],11)
```

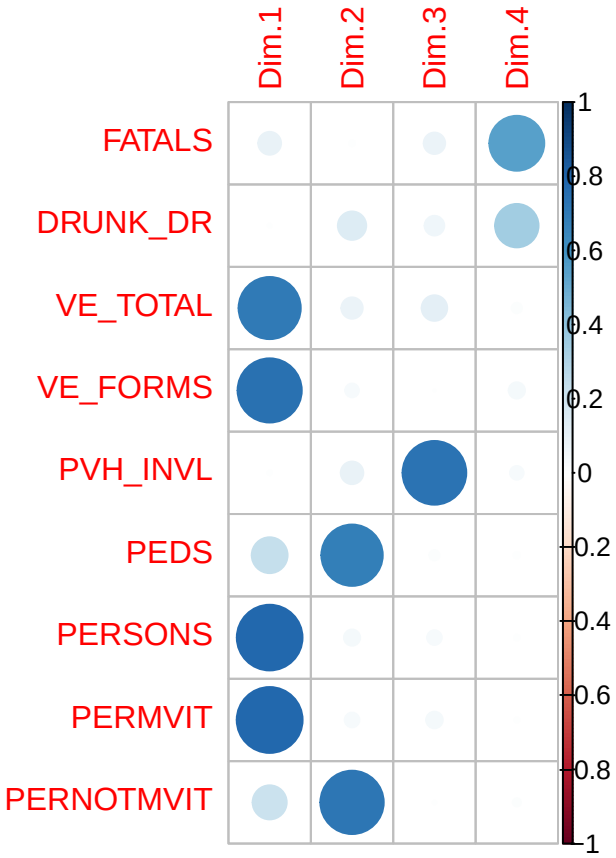
```
##                  Dim.1       Dim.2       Dim.3       Dim.4
## FATALS     0.094406509 0.005786562 0.083044911 0.547124037
## DRUNK_DR   0.002422180 0.145700988 0.069985146 0.342195618
## VE_TOTAL   0.700714591 0.082403990 0.118270488 0.018845040
## VE_FORMS   0.747937080 0.033918370 0.000575056 0.047815060
## PVH_INVL   0.003675475 0.092161422 0.736929104 0.033623861
## PEDS       0.232796860 0.683285394 0.018697211 0.006977896
## PERSONS    0.783480750 0.047331446 0.038937806 0.005748910
## PERMVIT    0.787421038 0.039002717 0.049961697 0.004511183
## PERNOTMVIT 0.210428078 0.728562294 0.002278398 0.011674076
```

```
corrplot(var$cos2[,1:4], is.corre=FALSE)
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt =
## tl.srt, : "is.corre" is not a graphical parameter
```

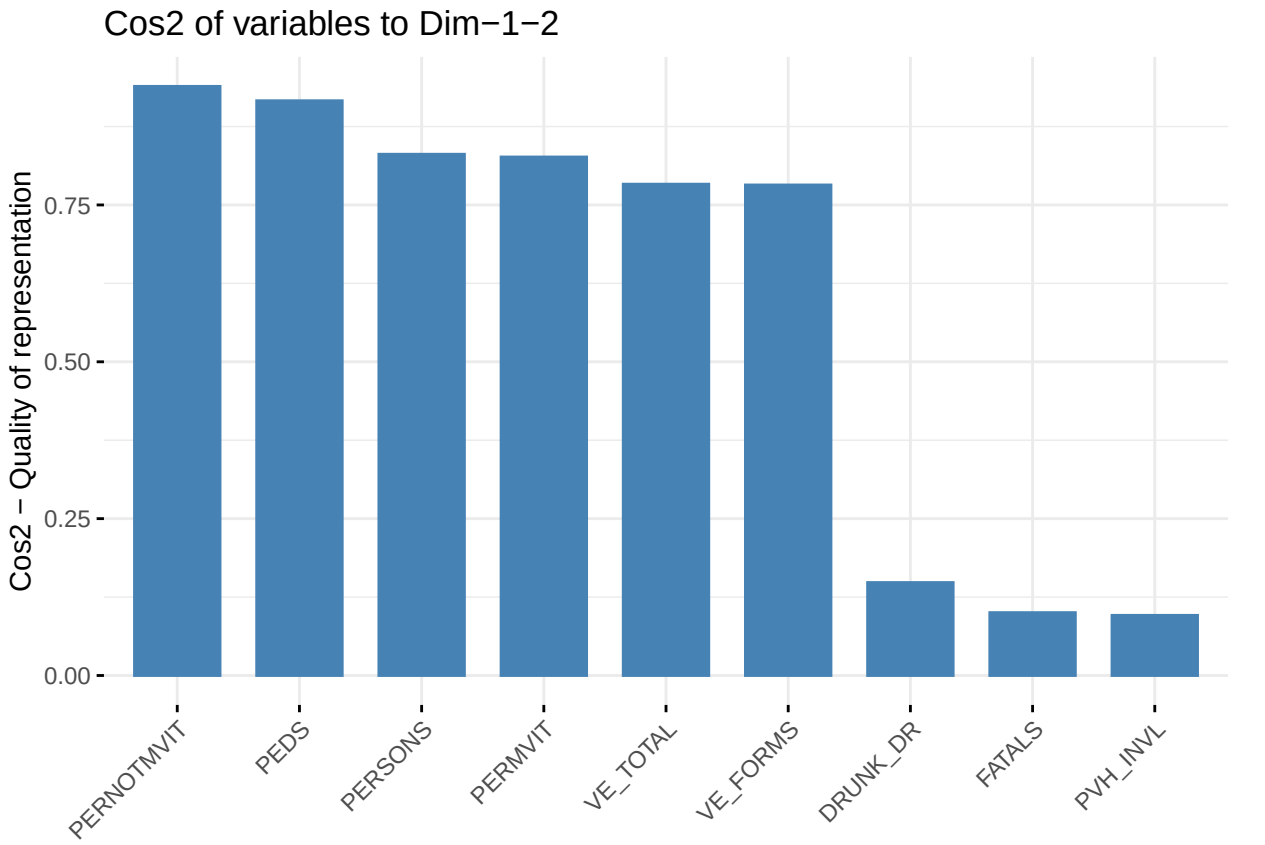
```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col =
## tl.col, : "is.corre" is not a graphical parameter
```

```
## Warning in title(title, ...): "is.corre" is not a graphical parameter
```



También es posible crear un diagrama de barras de variables cos2 mediante la función fviz_cos2():

```
fviz_cos2(pca.acc_scale, choice = "var", axes = 1:2)
```



- Un cos2 elevado indica una buena representación de la variable en el componente principal. En este caso, la variable se coloca cerca de la circunferencia del círculo de correlación.
- Un cos2 bajo indica que la variable no está perfectamente representada por los PC. En este caso, la variable está cerca del centro del círculo.

Para una variable dada, la suma del cos2 de todos los componentes principales es igual a uno.

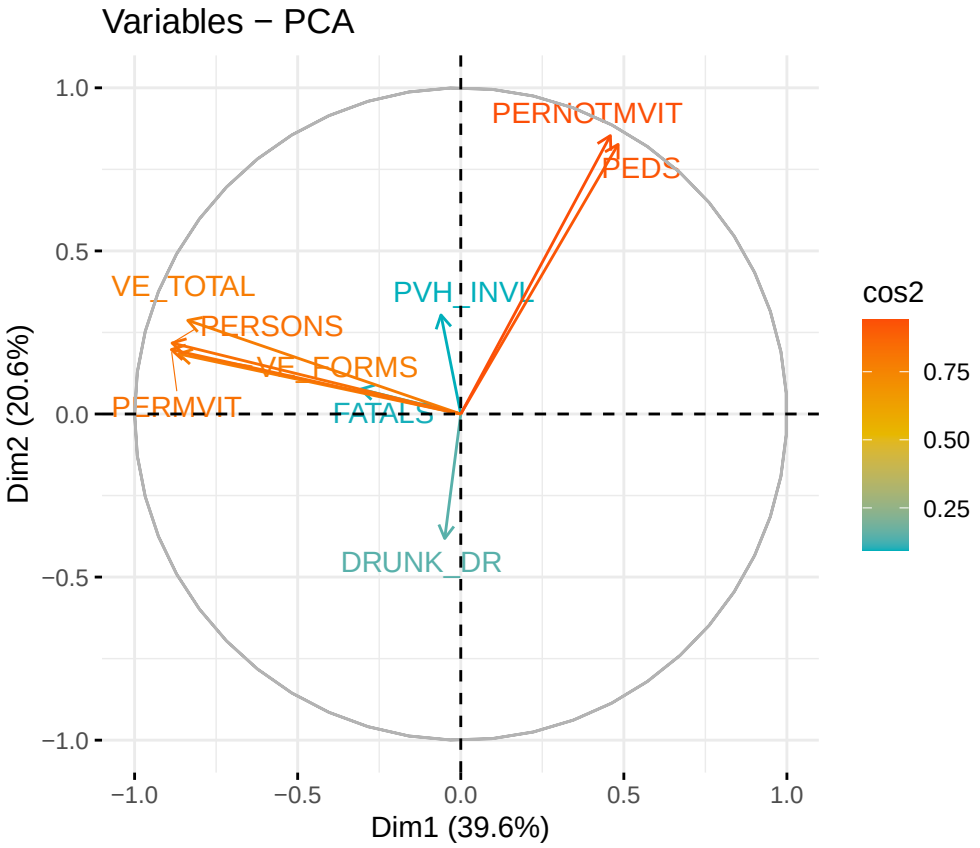
Si una variable está perfectamente representada por solo dos componentes principales (Dim.1 y Dim.2), la suma del cos2 en estos dos PCs es igual a uno. En este caso las variables se colocarán en el círculo de correlaciones.

Para algunas de las variables, pueden ser necesarios más de 2 componentes para representar perfectamente los datos. En este caso las variables se sitúan dentro del círculo de correlaciones.

En resumen:

- Los valores de cos2 se utilizan para estimar la calidad de la representación
- Cuanto más próxima esté una variable al círculo de correlaciones, mejor será su representación en el mapa de factores (y más importante es interpretar estos componentes)
- Las variables que están próximas en el centro de la trama son menos importantes para los primeros componentes.

```
fviz_pca_var(pca.acc_scale,
col.var = "cos2",
gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
repel = TRUE
)
```



Contribución

Las contribuciones de las variables en la contabilización de la variabilidad de un determinado componente principal se expresan en porcentaje.

Las variables que están correlacionadas con PC1 (es decir, Dim.1) y PC2 (es decir, Dim.2) son las más importantes para explicar la variabilidad en el conjunto de datos.

Las variables que no están correlacionadas con ningún PC o con las últimas dimensiones son variables con una contribución baja y se pueden eliminar para simplificar el análisis global.

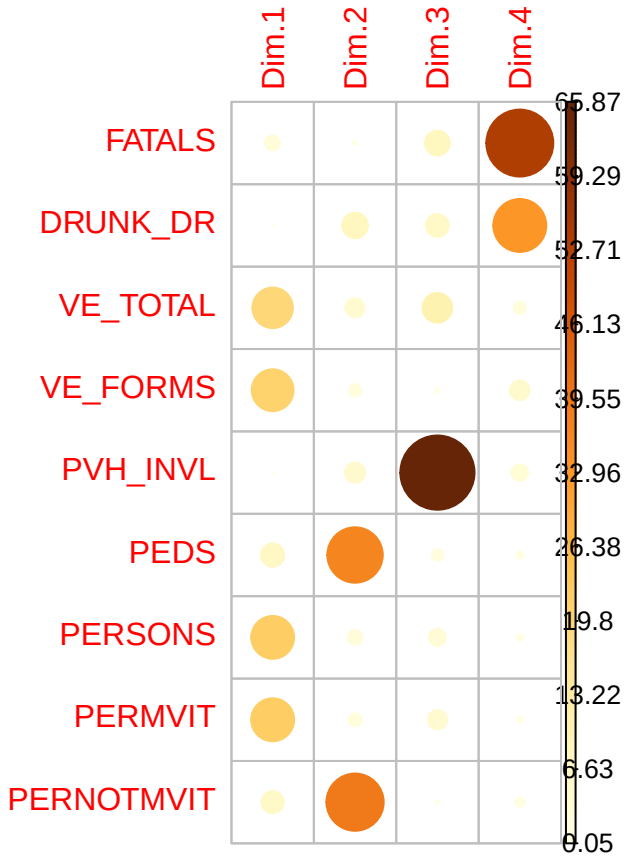
La contribución de las variables se puede extraer de la siguiente manera:

```
head(var$contrib[,1:4],11)
```

```
##                  Dim.1      Dim.2       Dim.3      Dim.4
## FATALS      2.64942527  0.3114147  7.42347448 53.7177824
## DRUNK_DR    0.06797608  7.8411720  6.25604798 33.5974816
## VE_TOTAL   19.66486180  4.4347254 10.57232696  1.8502454
## VE_FORMS   20.99011424  1.8253807  0.05140488  4.6945826
## PVH_INVL    0.10314857  4.9598398 65.87489045  3.3012610
## PEDS        6.53321358 36.7722855  1.67136397  0.6851044
## PERSONS    21.98761218  2.5472306  3.48069265  0.5644400
## PERMVIT    22.09819245  2.0990044  4.46613018  0.4429174
## PERNOTMVIT  5.90545583 39.2089469  0.20366845  1.1461852
```

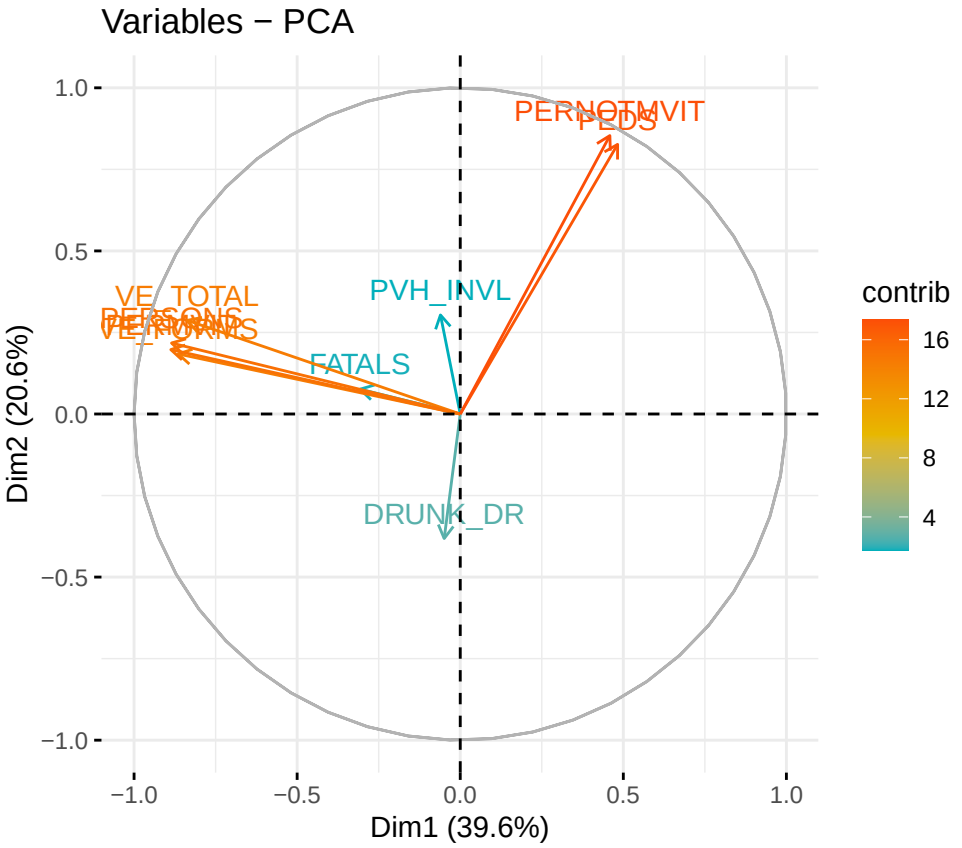
Cuando más grande sea el valor de la contribución, más contribución habrá al componente.

```
corrplot(var$contrib[,1:4], is.cor=FALSE)
```



Las variables más importantes (que más contribuyen) se pueden resaltar a la gráfica de correlación de la siguiente manera:

```
fviz_pca_var(pca.acc_scale, col.var = "contrib",
gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")
)
```



Las variables correlacionadas positivas apuntan al mismo lado de la trama. Las variables correlacionadas negativas apuntan a lados opuestos del gráfico. Por ejemplo, vemos que las personas involucradas en un accidente (PVH_INVL) y conductor bebido (DRUNK_DR) apuntan direcciones opuestas por tanto no están nada correlacionas, además lo hemos visto antes puesto que tienen un coeficiente de correlación de -0.01.

Se observa que las variables que más aportan a las componentes principales son **PEDS y PERNOTMVIT por un lado y VE_TOTAL, VE_FORMS, PERSONS y PERMVIT por otro**. Esto es debido al hecho que están correlacionadas. En concreto por el diagrama de correlación de antes de que PEDS está muy bien correlacionada con PERNOTMVIT. De otra banda VE_TOTAL, VE_FORMS, PERSONS y PERMVIT están también bastante correlacionadas. La correlación de FATALS con este grupo de variables no es elevada, pero apunta en la misma dirección.

Podrían ahora rehacer los componentes excluyendo las variables que no aportan información. Una vez rehechas estas nuevas variables sustituyen a las originales que las forman y se podrían utilizar por ejemplo como un indicador de gravedad de accidente puesto que incluye vehículo en movimiento, parado, peatones, conductores y otros implicados en una sola variable.

Interpretación de los resultados

Los datos estudiados contemplan accidentes de tráfico con víctimas en las redes de autopistas en los EUUU a largo del 2020. Todos los registros tienen un identificador único de accidente y una serie de hechos principales como número de muertos, número de conductores bebidos, vehículos y personas implicadas. Tenemos que añadir otras variables que los caracterizan agrupadas por ubicación geográfica, temporal, condiciones específicas del accidente, meteorológicas, la intervención del servicio de emergencias y otros factores.

Revisados los datos parecen bien informados. Los datos están bastante limpios y bien documentados. No plantean graves problemas de campos con valores nulos o vacíos y tienen bastante potencial para generar nuevos indicadores a partir de los datos.

Podemos afirmar que a lo largo del 2020 en las autopistas de EE. UU. sucedieron 35.766 accidentes en los que perdieron la vida 38.824 personas. Pretendiamos extraer relaciones entre la presencia de alcohol en los conductores y el número de accidentes, pero las conclusiones no fueron claras. Las relaciones más obvias comprobadas son el incremento de muertes en función del incremento del número de vehículos, pasajeros y peatones implicados.

Habría que profundizar mucho más. Sí que podemos perfilar cómo son los accidentes típicos en cuanto al número de vehículos y personas, conductores o peatones implicados.

El más habitual es un muerto por accidente incrementándose este valor en función de las variables relacionadas. Los conductores bebidos aparecen en uno de cada cuatro accidentes mortales aproximadamente. Los vehículos implicados en los accidentes son típicamente uno pudiéndose incrementar a dos en los casos más típicos. El número de peatones implicados es relativamente bajo dado el tipo de vía que estamos estudiando.

En cuanto al consumo de alcohol, con el grado de profundidad estudiado, no se observa un estado donde las proporcionalidades del número de conductor con presencia de alcohol sean superiores a otros estados.

Estudiando el número de muertes en accidente en relación con el estado donde ha sucedido y la condición climática, vemos necesario profundizar con técnicas por ejemplo de agregación para ver cómo se agrupan y poder obtener un perfil.

Se ha estudiado la franja horaria de la madrugada para ver si acumula un mayor número de accidentes, no siendo así. Parece que el número de accidentes mantiene también proporcionalidad respecto las franjas horarias. Se ha estudiado el número de accidentes por segmento horario con una discretización fijada en intervalo arbitrarios. La mayor presencia de accidentes en horario por la mañana y anochecer (ir y volver al trabajo) hace pensar en que queda pendiente estudiar dado el tipo de vía la distribución horaria de los accidentes lunes a viernes respecto a los fines de semana y festivos para ver si hay horas donde se acumulan más accidentes mortales.

Finalmente, con la técnica de los componentes principales hemos generado una nueva variable que combina otras variables con una correlación inicial que se podría considerar como índice de gravedad del accidente.

Ejercicio 1

Propón un proyecto completo de minería de datos. La organización de la respuesta tiene que coincidir en las fases típicas del ciclo de vida de un proyecto de minería de datos. **No hay que realizar las tareas de la fase**. Para cada fase indica cuál es el objetivo de la fase y el producto que se obtendrá. Utiliza ejemplos de qué y cómo podrían ser las tareas. Si hay alguna característica que hace diferente el ciclo de vida de un proyecto de minería respecto a otros proyectos indícalo.

> Escribe aquí la respuesta a la pregunta
>
> Como en cualquier proyecto de Minería de Datos, o de búsqueda de conocimiento en bases de datos, primero hay que tener claro cual es el objetivo principal, en definitiva, cual es el conocimiento que queremos extraer. Como es obvio, al tratarse de un proceso práctico, encontraremos dificultades e incluso evidencias que prueben lo contrario de aquello que queremos demostrar.
>
> Hay que tener en cuenta que la defínción del objetivo(s) es un paso muy importante, ya que tal y como se ha leído en “Proceso de Minería de Datos”; “Los objetivos del proyecto de minería de datos determinan el tipo de modelo de conocimiento que tenemos que extraer”
>
> Para este proyecto de minería de datos, antes de definir el objetivo definitivo, he hecho una pequeña lluvia de ideas que he creído conveniente dejar en la respuesta a este primer ejercicio:
>
> - En que estados de EEUU hay más accidentes donde estén implicados autobuses escolares.

- ¿Cuantas personas de media suele haber implicadas en el coche accidentado en cada estado? Esto nos daría información acerca de la cantidad media de gente que viaja en un coche en cada estado de EEUU, y podría ayudarnos en otros proyectos de mineria de datos.
- La gente que toma drogas, ¿suele ir acompañada cuando tienen un accidente, o van solas? Podríamos hacer una comparativa entre estados y calcular la media de gente que hay en un coche accidentado, esto mismo podríamos hacerlo con el alcohol.
- ¿Hay más accidentes por drogas en zonas rurales o en zonas urbanas? ¿y por alcohol?
- Se podría hacer un estudio acerca de como el tiempo meteorológico afecta en accidentes causados por drogas o alcohol, es decir; ¿cuando llueve o nieva es más probable que la gente beba o se intoxique con drogas? o ¿es el buen tiempo el que propicia este tipo de accidentes?
- Podríamos también demostrar en que hora del día hay más accidentes con autobuses escolares implicados.
- ¿Que tipo de coches suele verse envuelto en un accidente, en que tipo de carreteras? y ¿en que franja temporal suelen darse esos accidentes (noche/dia)? ¿que tiempo hacia? ¿condiciones lumínicas? Luego hacer una comparativa entre estados.
- Buscar una correlación entre el tiempo que suele llegar al hospital la victima y el tipo de vía en el que se ha dado el accidente, y si era un entorno rural o urbano.
- Observando la base de datos me doy cuenta de que en la mayoria de los casos cuando la persona iba bajo los efectos del alcohol, o no se sabía lo que tardaba en llegar al hospital o directamente no iba al hostpital, además de que tal y como se ha descrito antes, siemore hay un muerto en todos los accidentes del archivo csv.
- Buscar una correlación entre las variables de si iba borracho o no y la catidad de gente implicada en el accidente. Hacer una comparativa entre estados, tipo de carretera y entorno (rural o urbano)
- ¿En que tipo de cruces o intersecciones se producen más accidentes? ¿La gente suele ir bebida en esos cruces o intersecciones?
- ¿Suele haber más peatones implicados o menos en un tipo de cruce o otro? Esto nos daría información acerca de cuanto de peligrosos son esos escenarios.
- ¿En que entornos hay más accidentes por intoxicación de alcohol (rurales/urbanos)?

OBJETIVO GENERAL: Tras haberle echado un vistazo a la base de datos "accidentes.csv" he visto que se especificaban datos como el entorno en el que se había dado el accidente (urbano/rural), si hay intersección y el tipo, el condado en el que había sucedido, la ciudad, así como el estado en el que se ha dado el incidente y las condiciones atmosféricas. Por lo tanto, el objetivo que propongo, persiguiría conocer los estados, el entorno y los tipos de interseciones que más accidentes acumulan, para en un futuro predecir posibles lugares geográficos donde más accidentes podrían ocurrir, y consecuentemente, prevenir estos accidentes.

Se trataría de un modelo predictivo, que como sabemos por teoría, este tipo de modelos guardan muchas similitudes con los modelos de clasificación, no obstante, este caso contempla más de dos clases (por ejemplo; hay más de dos estados en EEUU, más de dos ciudades en cada estado, etc. El entorno es la única clase binaria (rural o ubano) por lo tanto la tarea de clasificación puede comprenderse como un proceso de predicción que tiene que asignarle un valor a la etiqueta. Como hemos visto en teoría, varios ejemplos de modelos predictivos serían los arboles de decisión y un ejemplo más contemporáneo serían las series temporales construidas por redes neuronales.

Quitar este siguiente párrafo que a partir de las variables que haya en la base de datos, el modelo remarque aquellas que incrementen la gravedad del accidente, mostrando además el grado de relación entre las distintas variables, para saber además que situación se tiene que dar para que exista tal relación entre un conjunto de variables. Esto sería posible analizando los accidentes y

sacando un común denominador entre ellos, viendo que variables comparten, y aquellas variables que compartan serán las variables que aumenten el factor de riesgo en los accidentes.

Después de haber definido el objetivo del proyecto de minería de datos, se enuncia a continuación las fases del proyecto que se deberian que acometer:

FASES DEL PROYECTO:

- **Defínción del objetivo/tarea del proyecto:**: En esta fase se formula la pregunta cuya respuesta queremos obtener por medio del proyecto de minería de datos. La elección del modelo también forma parte de este apartado.
- **Selección de datos**: Como es de esperar y por lo que se ha inferido de la teoría, se nos proveerá con una base de datos, de la cual tendremos que extraer aquello que nos interese para nuestro proyecto de minería de datos. No obstante, previamente hay que "pulir" y enriquecer todos los datos, afin de asegurar buenos resultados. Como se ha visto en teoría y como dijo Pablo Picasso; "El arte es la eliminación de lo innecesario"
- **Comienzo del proceso de minería**: Esta parte es la que seguramente más tiempo nos lleve junto con la anterior, pues comprende muchos escalones y obstáculos que hay que superar, desde mi punto de vista, creo que es la parte más sucia pero a la vez más enriquecedora, ya que hay que mancharse las manos, ya sea graficando, calculando correlaciones, aproximando resultados, eligiendo el tipo de gráficas, cambiando de escala los datos, etc. No obstante, si ese trabajo es llevado a cabo correctamente, y si se explotan muchas de las infinitas posibilidades, lo más probable es que se acabe extrayendo conocimiento y se descubran relaciones y detonantes nunca vistos, de un conjunto de datos que incialmente podían resultar poco prometedores
- **Evaluación, interpretación e integración**: Primero tendremos que comprobar que el modelo lleva a cabo la tarea que le hemos asignado, luego tendremos que interpretar los resultados que arroje, ya que puede que el modelo haga "algo" pero es posible que los datos obtenidos no sean los esperados, y por ello hay que asegurarse que aquello que devuelve el modelo es lógico. Por último, si todo lo anterior se ha relalizado correctamente, y el modelo se comporta como esperamos, es momento de integrarlo donde corresponda, ya sea en el servidor de una empresa o de una institución académica o gubernamental.

Se han explicado las fases muy brevemente y acorde con aquello que se ha interpretado de la teoría. Tras haber indicado el objetivo del proyecto, ahora se pasa a la segunda fase; Origen de los datos

Origen de los datos

En este caso, no hay que buscar datos, porque ya se tiene la base de datos que se va a explorar en el proyecto; **accident.csv** El primer paso ahora consiste en cargar el fichero de datos

```
# Cargamos el fichero de datos
#nombre_archivo = 'accident.CSV' #Especifico el nombre del fichero que quiero leer o su ruta completa si

#Ahora se lee el fichero en formato csv
#datos <- read.csv(nombre_archivo)
```

Después de cargar los datos, hay que preparar los datos, para ello, primero se procede a comprobar su estructura, afin de:

- Saber como están dispuestos los datos.
- Como en este caso es una tabla, saber cuantas filas y cuantas columnas hay, y conocer hasta cierto rango, los valores de cada una de las filas o columnas.
- El número de variables y sus identificadores

Para ello se realiza esta pequeña exploración;

```
#structure = str(datos)
```

Lo que se obtendría al simular esta linea de código sería una visualización parcial de los datos. En el caso de tener un documento explicativo que explicase en detalle el significado de cada una de las filas o columnas, habría que revisarlo, para saber que información brindan cada una de ellas y saber si se pueden ir descartando filas y/o columnas de datos que no sean interesantes para el objetivo del proyecto Luego de haber llevado a cabo una breve previsualización, hay que estudiar la posibilidad de generar nuevas variables derivadas de algunas ya existentes, pero eso se verá en la siguiente fase; **limpieza de datos**.

En el apartado de la **limpieza de datos** se pretende procesar los datos con el fin de eliminar información redundante, errores tipográficos o incluso añadir datos nuevos. Para ello, primeramente se comprueba la existencia de valores nulos o posiciones vacias en la tabla de datos.

```
#print('NA')
#colSums(is.na(accidentData))
#print('Blancos')
#colSums(accidentData=="")
```

Esto devolverá el número de columnas (variables) que tengan valores NULOS o VACÍOS. En el caso de encontrarse con valores vacios, una práctica común sería calcular la media o la mediana de los valores que ese atributo toma a lo largo del resto de accidentes, y asignárselo a ese campo en concreto, hay que tener en cuenta, que tal y como se estudió en teoría, esto podría afectar al modelo negativamente, introduciendo ruido o confundiéndole, no obstante, es mejor eso que dejar celdas vacías. Si hubiese una columna entera con valores nulos o vacios, entonces se eliminaría ya que no habría ninguna referencia y por lo tanto no aportaría ninguna información al modelo, al revés, impactaría negativamente en él. Se ha podido comprobar en la base de datos de **accident.csv** como había varias celdas en

El siguiente paso consistiría en comprobar que no haya tuplas repetidas, es decir, filas que aparezcan en la tabla de datos más de una vez:

```
# Detectar filas duplicadas
#filas_duplicadas <- datos[duplicated(datos) | duplicated(datos, fromLast = TRUE), ] #fila dupliacada o

#Ahora se visualizan las filas duplicadas
#Por último se eliminan:
#unique_df <- unique(df)
```

La siguiente tarea sería **comprobar** que no hubiese ningún campo (celda dentro de cualquier columna) que contenga **números o caracteres erróneos**, es decir, **comprobar que no haya ningún dato inconsistente**. Si por ejemplo una variable solo puede tomar un conjunto de valores determinado, comprobar que para todos los accidentes, i.e., para la columna correspondiente a esa variable en todas las filas de la tabla de datos, esa variable tome el tipo de valor correcto (string, int o float) dentro del conjunto en cuestión.

```
# Detectar valores erróneos o inconsistentes, para ello se podrían gráficar los datos de cada atributo,
```

La siguiente tarea, podría consistir en la **búsqueda de datos envejedidos** y si se diese el caso, actualizar aquellos valores que indirectamente pueden suponer confusión al modelo. Se puede tomar el ejemplo visto en teoría; sustituir la edad de pacientes o víctimas, por su fecha de nacimiento. En este caso, en el fichero de **accident.csv** no hay ninguna edad especificada, por lo tanto implementar esta tarea no tendría mucho sentido, no obstante, aquellas variables que se hayan descalificado, se deberían de retirar del juego de datos.

La siguiente tarea podría consistir en **comprobar que no hubiese variaciones** en los valores que tomen las variables, es decir, que en el caso del nombre de la ciudad, del condado, el mes del año, el día de la semana y el formato horario, estén todos estos campos escritos de la misma forma,

o que todos empiecen por mayúsculas, o que todos los nombres estén en mayúsculas en el caso de los 4 primeros escenarios, y que en el caso de la hora, esta tenga el mismo formato en todos los accidentes. Por lo tanto, se van a modificar aquellas celdas que representen la misma información, pero que esté escrita/especificada diferentemente. Además, se ha comprobado para el atributo **TWAY_ID** que en algunos accidentes, las letras del código alfanumérico que toma el atributo, están en mayúsculas y otras en minúsculas, y en algunos casos el tipo de valor que toma este atributo puede ser o alfanumérico o puede estar solamente compuesto por letras o palabras, esto supone un problema considerable a la hora de gráficar o calcular datos relativos a este atributo.

```
nombre_archivo = 'accident.CSV'
datos <- read.csv(nombre_archivo)
names(datos) #Aqui leemos el nombre de las columnas
```

```
##  [1] "STATE"         "STATENAME"    "ST_CASE"      "VE_TOTAL"     "VE_FORMS"
##  [6] "PVH_INVL"      "PEDS"         "PERSONS"      "PERMVIT"      "PERNOTMVIT"
## [11] "COUNTY"        "COUNTYNAME"   "CITY"         "CITYNAME"     "DAY"
## [16] "DAYNAME"       "MONTH"        "MONTHNAME"    "YEAR"         "DAY_WEEK"
## [21] "DAY_WEEKNAME" "HOUR"         "HOURNAME"     "MINUTE"       "MINUTENAME"
## [26] "NHS"           "NHSNAME"      "ROUTE"        "ROUTENAME"    "TWAY_ID"
## [31] "TWAY_ID2"      "RUR_URB"      "RUR_URBNAME"  "FUNC_SYS"     "FUNC_SYSNAME"
## [36] "RD_OWNER"      "RD_OWNERNAME" "MILEPT"       "MILEPTNAME"   "LATITUDE"
## [41] "LATITUDENAME" "LONGITUD"     "LONGITUDNAME" "SP_JUR"       "SP_JURNAME"
## [46] "HARM_EV"       "HARM_EVNAME"  "MAN_COLL"     "MAN_COLLNAME" "RELJCT1"
## [51] "RELJCT1NAME"  "RELJCT2"      "RELJCT2NAME"  "TYP_INT"      "TYP_INTNAME"
## [56] "WRK_ZONE"      "WRK_ZONENAME" "REL_ROAD"     "REL_ROADNAME" "LGT_COND"
## [61] "LGT_CONDNAME" "WEATHER"      "WEATHERNAME"  "SCH_BUS"      "SCH_BUSNAME"
## [66] "RAIL"          "RAILNAME"     "NOT_HOUR"     "NOT_HOURNAME" "NOT_MIN"
## [71] "NOT_MINNAME"  "ARR_HOUR"     "ARR_HOURNAME" "ARR_MIN"      "ARR_MINNAME"
## [76] "HOSP_HR"       "HOSP_HRNAME"  "HOSP_MN"      "HOSP_MNNAME"  "FATALS"
## [81] "DRUNK_DR"
```

```
head(datos[c("MONTH","STATENAME","COUNTYNAME","CITYNAME","RUR_URBNAME")])
```

```
##   MONTH STATENAME     COUNTYNAME       CITYNAME RUR_URBNAME
## 1     1   Alabama    ELMORE (51) NOT APPLICABLE       Rural
## 2     1   Alabama JEFFERSON (73)     BIRMINGHAM       Urban
## 3     1   Alabama   SHELBY (117) NOT APPLICABLE       Rural
## 4     1   Alabama   CALHOUN (15) NOT APPLICABLE       Rural
## 5     1   Alabama     COOSA (37) NOT APPLICABLE       Rural
## 6     1   Alabama   MORGAN (103) NOT APPLICABLE       Rural
```

```
head(datos[c("DAY","HOUR")])
```

```
##   DAY HOUR
## 1   1    2
## 2   2   17
## 3   2   14
## 4   3   15
## 5   4    0
## 6   4   16
```

Esto es de especial importancia, ya que a la hora de introducir estos datos en un modelo de aprendizaje automático o en cualquier modelo de detección o clasificación aunque para una persona signifiquen lo mismo, un modelo computacional interpretará que January y JANUARY son cadenas de caractéres diferentes, cuando en realidad reresentan los mismo. En este caso hemos visto como la variable MONTH es de tipo int, al igual que el condado y la ciudad, en cambio el estado es una cadena de caractéres. No obstante existen, variables relativas al nombre del

condado “COUNTYNAME” y al nombre de la ciudad “CITYNAME”, que son las variables que se muestran arriba.

Observando los resultados de arriba uno se da cuenta de como, cuando se trata de un entorno rural, el nombre de la ciudad no aplica, como es lógico, por lo tanto, podría ser útil obviar una de las variables, porque las dos nos dan la misma información, pero una es más especifica, en este caso, la variable “CITYNAME” indica el nombre de la ciudad, en cambio la otra solo muestra si se trata de un entorno rural o urbano, por lo tanto se podría prescindir de la variable que determina el entorno, porque al saber el nombre de la ciudad, se sabe que ciudad es, y si no fuese una ciudad directamente mostraría un “NA”. No obstante, he decidido no eliminarla por ahora, porque para las primeras gráficas, creo que es importante mantener los entornos, para saber donde se producen más acidentes, y luego ir filtrando por ciudades.

El siguiente paso, podría consistir en analizar si los datos que hemos escogido están sesgados, y estudiar si esto repercute en nuestro objetivo principal del proyecto. En este caso, podríamos remarcar que hay un sesgo, el hecho de que todos los accidentes tengan una víctima mortal, no obstante, en principio, este sesgo no influiría negativamente en nuestro proyecto, ya que nuestro objetivo ya contempla la muerte como algo “normal” o habitual a la hora de llevar a cabo nuestro estudio. En el caso de querer hacer un estudio que no involucre muertes, se deberían de incluir esos datos, pero habría que repetir el proceso de limpieza de datos y habría que estudiar más en detalle como afectaría este sesgo al objetivo principal del proyecto.

Ahora ya se han limpiado los datos, a continuación, se podrían a empezar a graficar parte de los datos ya “limpios” mediante herramientas estadísticas, como los histogramas, que nos permiten visualizar las variaciones de los datos en cada atributo para cada accidente en concreto

```
#unique(datos["RUR_URB"]) # Para saber cuantos valores posibles hay
#unique(datos["CITYNAME"])
#unique(datos["STATENAME"])

#Contamos cuantas veces aparecen cada una de los posibles valores dentro de cada variable, esto nos dara
#table(datos$RUR_URBNAME)
#table(datos$CITYNAME)
#table(datos$STATENAME)

names(which.max(table(datos$RUR_URBNAME)))
```

```
## [1] "Urban"
```

```
names(which.max(table(datos$CITYNAME)))
```

```
## [1] "NOT APPLICABLE"
```

```
names(which.max(table(datos$STATENAME)))
```

```
## [1] "California"
```

```
names(which.max(table(datos$NHSNAME))) # La carretera en la que más accidentes se han procudido, es una
```

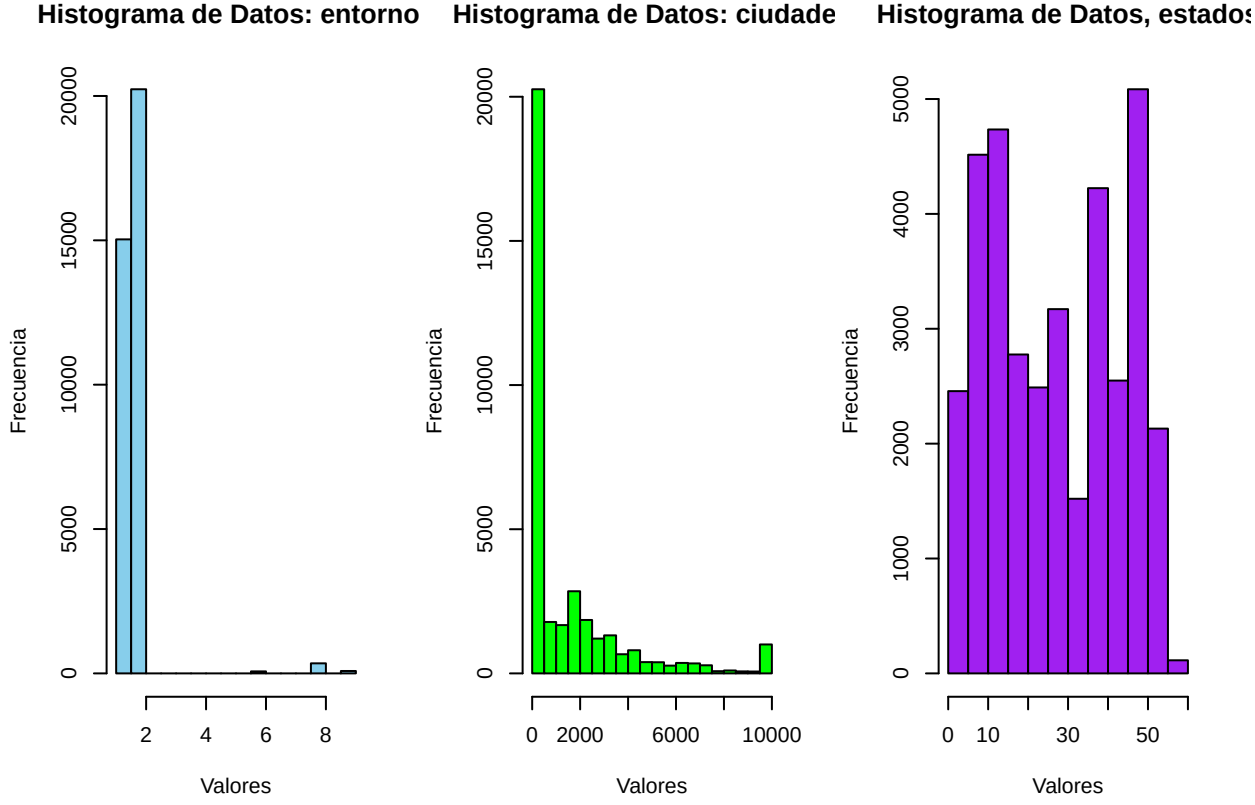
```
## [1] "This section IS NOT on the NHS"
```

```
names(which.max(table(datos$RD_OWNERNAME))) #De quien es la carretera en la que se ha dado el accidente
```

```
## [1] "State Highway Agency"
```

```
par(mfrow = c(1, 3))  # Dividir la ventana de gráficos en dos
#Ahora graficamos los histogramas
hist(datos$RUR_URB,main = "Histograma de Datos: entornos",
     xlab = "Valores",
     ylab = "Frecuencia",
```

```
     col = "skyblue",
     border = "black")
hist(datos$CITY,main = "Histograma de Datos: ciudades",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "green",
     border = "black")
hist(datos$STATE,main = "Histograma de Datos, estados",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "purple",
     border = "black")
```



Como se puede ver por el último output relativo al mayor número de ocurrencias de cada valor dentro de cada variable/coolumna, se infiere como, el mayor número de accidentes se dan en zonas urbanas, que el mayor número de accidentes no se dan en una ciudad en concreto, por lo que podrían y por último, el mayor número de accidentes se dan en el estado de California. Además, se ve como la mayor part de los accidentes se dan en sectores de carreteras que no pertenecen al sistema nacional de autovías pero legalmente pertenece a la agencia estatal de carreteras.

Los histogramas, servirán en el Ejercicio 2 dónde a partir de representciones gráficas se podrán sacar conclusiones e ir "tirando del hilo" hasta poder obtener el conocimiento asociado al objetivo del proyecto de minería de datos.

La siguiente tarea tendría varias fases, y principalmente consistiría en llevar a cabo distintas transformaciones a los datos, con el propósito de darles la forma deseada.

Habiendo estudiado la base de datos algunas de las transformaciones que se podrían llevar a cabo en este fase del proyecto, podrían ser las siguientes:

- **SIMPLIFICACIÓN DECIMAL**: En la base de datos, se especifica la latitud y la longitud. En este caso las dos columnas de valores, tienen muchos decimales, y como se ha visto por teoría y por lo que se infiere por lógica, todos esos decimales requieren de espacio para ser almacenados, por lo tanto cuantos más decimales haya, más espacio necesitará ese valor para almacenarse, por lo tanto esos decimales podrían aproximarse/redondearse hasta cierto punto, ya que todo depende de la exactitud que se necesite, no osbtante, esto no es un problema para el proyeto que se ha propuesto.

  En lo que respecta al objetivo del proyecto que se ha propuesto, la latitud y la longitud no aportan mucha información adicional, ya que la base de datos es exclusiva a los EEUU, y además, ya se conoce el estado, la ciudad, y el condado en el que se ha dado el accidente. No obstante, a no ser que este dato se usase para otro proyecto de minería de datos, en el que hubiese una comparación entre los estados de un país y de otro en términos de accidentes, podría descartarse.

  Aunque, por precaución este dato no quedaría descartado, porque serviría para calcular la media de la latitud y la longitud de cada uno de los accidentes, en el estado en el que más accidentes se producen, esto también podría ayudar a saber con exactitud que punto geográfico concentra más accidentes de media, a partir de ahí se podría trazar un radio en kilómetros que determinase la peligrosidad de la zona. Esto último supondría añadir una variable adicional, algo posible y que se ha estudiado en teoría.

- **TRANSFORMACIÓN DE ESCALA**: Se podrían aplicar transformaciones de escala en el caso de la latitud y de la longitud, ya que poseen valores negativos y positivos que van desde los -90º a +90º y de -180º a +180º correspondientemente, y aunque una comparación entre estos dos atributos no supondría un problema, puede ser que al realizar una gráfica o una comparativa entre una de esas variables y otra del conjunto, pueda dar lugar a confusión.

- **DATOS NUMÉRICOS A CATEGÓRICOS y AGRUPACIÓN DE VALORES**: Se podrían agrupar las latitudes numéricas por latitud; norte y sur y las longitudes también numéricas; oeste y este. pero con el punto 0 en el centro de EEUU, ya que si se tomáse la referencia mundial, llevar a cabo esta transformación sobre estos dos datos no tendría mucho sentido, puesto que EEUU se encuentra en la latitud norte y en longitud oeste.

- **DIVIDIR EL ATRIBUTO ENTORNO**: Por practicidad, parar facilitar la representación gráfica, se podría dividir la variable **ENTORNO** en dos variables: rural y urbano, así se pasaría de una variable categórica a una variable numérica, así podría ahorrarse más espacio, porque aunque se aumente en una columna la base de datos, cada celda solo contendrá un bit de información, un 0 o un 1.

- **CREACIÓN DE UN NUEVO ATRIBUTO**: Creación del atributo gravedad, que permita clasificar los accidentes por MUY GRAVE-GRAVE-NO MUY GRAVE-LEVE. Aunque crear un atributo no es como tal una transformación de los datos, se me he ocurrido en este momento. Pero esto conecta con un tipo de transformación, la **expansión de un atributo**. También es verdad que esto daría para otro proyecto de minería de datos, ya que es igual al objetivo de proyecto mencionado al inicio del ejemplo de arriba del todo de esta prueba. Pero como no hay que desarrollar código, el proyecto principal mencionado al principio de este ejercicio podría verse complementado por la introducción de este objetivo de clasificación de accidentes. a pesar de la alta complejidad que ello conllevaría.

Conectando con lo último mencionado y ahora sí, se podrían crear variables nuevas, que cumplan con el objetivo expuesto antes. No obstante, no se me ocurren otras variables que puedan surgir entre el matrimonio de dos ya existentes, porque en términos prácticos y teniendo en cuenta el objetivo del proyecto que se ha especificado, en principio la base de datos ya tiene las variables suficientes como para obtener conocimiento a lo largo pero sobre todo al final del proceso de la minería de datos.

Luego se podrían enriquecer los datos, con datos de otras bases de datos, pero como lo que

se quiere estudiar son los estados, los tipos de intersecciones y el entorno en el que se dan los accidentes con un fin preventivo, no tiene mucho sentido añadir datos de otros países, no obstante si hubiese otra base de datos que tuviese datos a nivel nacional (de EEUU) entonces si que habría que tener en cuenta este punto.

La siguiente tarea a tener en cuenta sería la **reducción de la dimensionalidad** para ello se podrían estudiar los posibles valores que toman las variables, e intentar reducir el área de operación. Esto es importante, ya que como se ha estudiado en teoría, trabajar con menos datos y obtener los mismos resultados que para un conjunto mayor, es mucho mejor, eficientemente hablando.

```
#Primero se podría calcular la varianza, para ver si hay mucha diversidad en los datos de un atributo en
```

Esto daría como resultado una base de datos más pequeña, ya que se habrían descartado accidentes puntuales en zonas geográficas. En el próximo paso, una vez se han limpiado los atributos por dentro, se podría hacer una criba más considerable, dando un paso más alla, y eliminando atributos que no vayan a aportar información nueva o relevante. Como ya se mencionó en otros apartados, se podría considerar la eliminación de los atributos de longitud y latitud, en el caso de que estos no arrojen información interesante comparándolos con otros atributos. Luego, podrían extraerse o los nombres de los condados, ciudades o estados, o se podrían extraer los identificadores asociados a dichos nombres para esos tres atributos, esto persigue simplificar aún más el juego de datos. Seguramente sea más eficiente trabajar con números que con cadenas de carácteres, por lo tanto se podría implementar un mapeo entre el nombre de la ciudad/condado/estado con su identificador, para que a la hora de interpretar los resultados gráficos, sea posible ver el nombre de la ciudad/estado/condado y no sus identificadores asociados, ya que estos últimos, por sí solos no aportan información. Pero lo que está, es que los dos representan la misma información pero de diferente manera, por lo tanto uno de ellos se tiene que ir fuera, ya sean los identificadores o los nombres. Este proceso recibe el nombre de **selección de atributos** y es una fase crítica en el proceso de transformación de datos.

Esto se podría aplicar también al atributo **ENTORNO**, pues este no aporta información cuando aparece el nombre de una ciudad en el mismo accidente, pues ya se infiere que el accidente se ha dado en un entorno urbano y no rural. En el caso de que el accidente se diese en un entorno rural entonces, en el atributo relativo al nombre de la ciudad, aparecería: **NOT APPLICABLE**. Volviendo al objetivo de nuestro proyecto de minería de datos y conectando con el análisis que se acaba de hacer, se podrían eliminar muchos más atributos como por ejemplo; **DAY_WEEKNAME**, así se quedaría solo, el día de la semana, esto es práctico, porque para una persona es muy fácil asignarle el nombre del día al número del día de la semana, esto supondría un ahorro en memoria. Otros ejemplos serían; **MONTHNAME**,**MINUTENAME**,**NHSNAME**, **LATITUDENAME**,**LONGITUDNAME**, **RELJCT2NAME**, **WRK_ZONENAME**, **RAILNAME**, **SCH_BUSNAME**, todos estos atributos cumplen con una condición, y es que son todos atributos binarios, es decir, solo pueden tomar dos palabras como valor, y las palabras que toman normalmente, son **YES || NO** o repiten aquello que se especifica en su variante númerica pero con palabras, por lo tanto es más eficiente mantener los identificadores numéricos y quitar las variantes de cadenas de caracteres.

En este punto, los datos del conjunto son mejores, porque se ha conseguido eliminar lo innecesario, la tarea que a uno ocuparía ahora sería la del **proceso de construcción del modelo**.

Comienzo del PROCESO de MINERÍA DE DATOS

En este paso se supone que la base de datos ya está "limpia" y se han eliminado los datos redundantes, erróneos, confusos o incluso se han añadido datos, ya sea a nivel de variables/atributos (columnas en la tabla) o a nivel de valores dentro de los propios atributos ya existentes.

En este paso se elegirá un modelo, dentro del conjunto de modelos posibles que cumplan con las características del objetivo del proyecto de minería.

Inicialmente, antes de estudiar la teoría, yo pensaba que el término; modelo, solo hacía referencia a la herramienta que permite extraer/crear conocimiento a partir una base de datos, y aunque en el fondo esto es así, el término modelo también hace referencia a la orientación que se le quiere dar el proyecto, es decir, está relacionado con el objetivo del proyecto enunciado al principio de este ejercicio. Es pertinente mencionar, que puede que me haya adelantado realizando esta tarea de orientación del proyecto de minería y haya especializado demasiado el proceso del tratamiento de la base de datos, siguiendo el objetivo enunciado. A mi parecer esto tiene una vetaja y una desventaja.

La ventaja reside principalmente en el hecho de que se han ido eliminando atributos/variables e información en general, que se creía prescindible respecto al objetivo. No obstante, esto crea una posible desventaja, y es que puede ser que a lo largo del proyecto se vaya alterando el enfoque inicial que se tenía acerca del objetivo principal del proyecto, lo que se traduciría en una necesidad de datos que se fueron descartando a lo largo del tratamiento y el analísis de la base de datos.

Dicho esto, se comienza con la elección de un posible modelo para el proyecto.

Como ya se mencionó al incio de este ejercicio. El objetivo actual e inicial, consiste en intentar predecir posibles lugares geográficos donde más podrían ocurrir accidentes a partir de un proceso de minería de datos centrado principalmente, en atributos geográficos.

Debido al objetivo principal del proyecto de minería de datos, el tipo de modelo que más se ajusta, es un modelo predcitvo, y por teoría, se sabe que los modelos predictivos, presentan muchas similitudes con el proceso de clasificación, pues al fin y al cabo, clasificar algo equivale a predecir el valor de ese algo dentro de un conjunto limitado de valores. Ahora bien, cuando hay más de dos clases, el proceso de clasificación se entiende como una tarea de predicción que tiene indicar el valor de la etiqueta.

En el caso de este proyecto, se podría predecir que lugares podrían albergar accidentes, estudiando aquellos ya documentados en la base de datos. Este estudio se llevaría a cabo en cada estado del país, y para saber que lugares geográficos podrían albergar futuros accidentes en más alta probabilidad, habría que estudiar y relacionar atributos como, condados, ciudades, entornos, tipos de vias, las intersecciones y sus tipos, etc. Este proceso sería complejo y muy elaborado ya que se tendrían en cuenta muchas clases.

MECÁNICA GENERAL DEL PROCESO DE BÚSQUEDA

El conocimiento **a priori** (por ejemplo: se sabe que las zonas más pobladas tienen más probabilidad de albergar accidentes, pues que el modelo tenga un enfoque parecido) que se tiene, el estudio de la base de datos que se ha realizado, y el objetivo del proyecto, apuntan a las **redes neuronales**, como buenas candidatas iniciales para liderar esa extracción del **conocimiento**. No obstante, habría que evitar el problema de los **mínimos locales** mencionado en teoría, para ello se tendría que buscar una métrica que permitiése comparar los resultados del entrenamiento de diferentes redes neuronales con un número distinto de capas y en diferentes arquitecturas, para poder determinar que modelo de red neuronal es mejor. **La métrica que puede determinar**, que modelo es mejor, puede ser el la del valor de pérdidas al final del entrenamiento, ya que aquella arquitectura de red que presente un menor valor de pérdidas al final de su entrenamiento, será la red más adecuada para la extracción del conocimiento en el proyecto de minería de datos

No obstante, hay que tener en cuenta que las redes neuronales, a pesar de presentar un gran potencial a la hora de realizar tareas repetitivas tras su entrenamiento, requieren de una gran cantidad de datos para el entrenamiento, y este además, tiene que ser lo suficientemente largo, como para que la red realice bien la tarea que la ocupa. Por el contrario, si el conjunto de datos de entrenamiento no es suficiente, se podría dar el famoso **overfitting** en castellano sobre-ajuste. Este fenómeno ocurre cuando el modelo no puede generalizar bien y en cambio se ajusta demasiado a los datos del entrenamiento. Por lo tanto, habría que estudiar si todos los datos que hay en la base de datos son suficientes para entrenar la red que se encarge de clasificar las zonas geográficas que más accidentes puedan tener. Habiendo estudiado en más o menos detalle la base de datos,

me he dado cuenta de que hay una gran variedad de accidentes, pues muchos de ellos cuentan con valores muy característicos en sus atributos, esto puede jugar a favor de un buen entrenamiento, ya que no todos los accidentes son iguales, y por ende la red aprendería bien, pero claro, esto la llevará mucho tiempo, y también tiene que haber muchos accidentes dentro de cada tipo de accidente, para que la red tenga un buen ejemplo.

He elegido una red neuronal para este caso, porque se ajusta a las caracteristicas del proyecto de minería, y además hace no mucho realicé my TFG en el campo de la visión por computador, por lo tanto conservo muchas ideas.

Al haber elegido una red neuronal como modelo que acometa el objetivo propuesto hay que tener en cuenta varios aspectos vistos en teoría y muy propios de las redes neuronales, que van a determinar un correcto funcionamiento del mismo:

- **La función de coste y de pérdidas y funciones de activación**: Estas funciones representan cosas diferentes, pero todas son **medidas de calidad** que califican cada una de las arquitecturas neuronales dentro del mundo de la IA. A continuación se explica muy brevemente la misión de cada una de las funciones: la función de pérdidas es una función de optimización que se busca minimizar, para reducir el error de predicción. Cabe mencionar que el trabajo que realizan las redes neuronales, es el de opimización, estas se dedican a buscar minimos en funciones mediante diferentes maneras, como por ejemplo, mediante el algoritmo del gradient descent. Luego, la función de costes se encarga también de medir el error del modelo, pero en un conjunto de datos, a diferencia de la función de pérdidas que solo maneja una instancia de datos. Estas dos funciones determinan cuanto de bien le está yendo al entrenamiento, y por lo tanto constituyen métricas de calidad que han de utilizarse en el proceso de evaluación del modelo, para saber cuanto de bueno es. Por último, la función de activación se aplica en cada neurona y está se encarga de transformar los datos de entrada que le llegan en características abstractas más separables linealmente. Las funciones de activación determinan directamente el entrenamiento de la red y por consecuente condicionan su aprendizaje.
- **La tasa de aprendizaje y los pesos**: La tasa de aprendizaje y los pesos son hiperparámetros de la red, mientras que la primera se encarga de regular el ritmo al que un algoritmo (e.g., el gradient descent) actualiza o aprende los valores de estimación de un parámetro, los pesos determinan la intensidad de interacción entre cada neurona. Estos dos hiperparámetros, también forma parte del conjunto de métricas de calidad
- **Umbrales de detección**: Permiten saber a la red que resultados descartar y cuales no, en principio estos parámetros suelen modificarse habitualmente al realizar el entrenamiento de una red.

VARIEDAD DE MODELOS DE BÚSQUEDA

Seguramente hayan más funciones y parmámetros que permitan evaluar el modelo escogido, pero creo que las más importanes se han mencionado arriba. Soy consciente que en esta última parte me he centrado mucho en explicar el modelo desde la perspectiva de las redes neuronales, y no tanto desde la minería de datos.

Volviendo al conocimiento **a priori** que se ha hecho referencia en esta fase del proyecto, hay que saber como integrarlo. Un método que podría ser intersante sería el de las **relaciones de coocurrencia** ya que permitiría la expresión de dependencia entre dos atributos, en nuestro caso: "siempre que hay un accidente en una ciudad, significa que el entorno del accidente es urbano". Pensando un poco a lo loco, esto podría especificarse en la red neuronal, modificando los valores de los pesos, para que los altos pesos de una neurona y otra reflejen la alta intensidad de interacción entre ellas justificando así la dependencia de estos dos atributos.

Para mejorar el aprendizaje de la red, es pertinente destacar la gran cualidad enriquecedora que tiene el **transferlearning** en el campo de las redes neuronales, ya que esto permite reciclar

conocimiento de otra tarea, con el fin de que la red no empiece desde 0 y tenga conocimientos "innatos" esto en la práctica se traduce en implementar unos pesos ya entrenados.

Conectando con el último párrafo, y según el conocimiento **a priori** que se tenga, este proyecto de minería de datos implementaría un método de **reutilización del conocimiento extraído** por otro sistema de minería de datos, a fin de dirigir mejor la tarea de búsqueda.

Si se tiene en cuenta el **tipo de datos** el modelo escogido basado en **redes neuronales** se corresponde efectivamente con un método de minería de datos **supervisados** ya que la red clasificaría las **observaciones** asignándoles un valor o una etiqueta. Aplicando esto al modelo que se ha propuesto, basado en una red neuronal, la red analizaría cada uno de los accidentes y los clasificaría dependiendo de si forman o no parte de un lugar geográfico con muchos accidentes.

Por último, teniendo en cuenta el método de construcción del modelo, el método de minería de datos seria de naturaleza **batch** ya que la red tomaría la base de datos como el único juego de datos posible, ahorrando así en complejidad. Si se fuese ambicioso y a fin de evitar los problemas de la **deriva de modelos** y **datos envejecidos** se podría implementar una red basada en el método **incremental** modificando sus parámetros según cambian los datos en el tiempo.

EVALUACIÓN e INTERPRETACIÓN del modelo

Esta es la penúltima fase del proyecto, y si todo lo anterior ha ido bien, en esta fase ya se tendría un modelo capaz de representar el conocimiento deseado, respondiendo así a la pregunta formulada al inicio de este ejercicio.

Ahora es el momento de evaluar el modelo, para ello habría que evaluar el funcionamiento de la red que se ha propuesto, por lo tanto la siguiente tarea consistiría en poner a prueba el modelo construido formando tres conjuntos de datos de accidentes, un conjunto para su construcción, otro para validar que el modelo funciona y un último conjunto para evaluar el comportamiento del modelo. Por lo tanto, se extraerían tres conjuntos con accidentes variados en cada unos de ellos.

La tarea a realizar ahora, sería la de aplicar un modelo predictivo de machine learning o de IA (por ejemplo) para formar los tres conjuntos de datos. Como el objetivo es predecir zonas geográficas donde se puedan generar nuevos accidentes a partir de otros atributos, como el tipo de via o intersección, se procedería a separar los accidentes en dos bloques, aquellos que se han dado en zonas con una alta densidad de accidentes (zonas de mayor riesgo) y aquellos que se dan en zonas no tan "desafortunadas". Al primer bloque de datos (accidentes en zonas de mayor riesgo) se le aplicaría el modelo depredictivo mencionado antes, para conocer que accidentes se encuentran en zonas de alta densidad de siniestros y para saber cuales no. Seguidamente, se extraeran las cambinaciones de valores que pronostican la pertenencia a una clase o a otra. Por ejemplo, como se ha visto anteriormente, las zonas de mayor riesgo de accidente, son aquellas en entorno urbano, o interurbano, pues el término que más aparece relativo a las ciudades es: "NO APLICA", dónde el tramo de carretera donde se ha producido el accidente no suele formar parte del sistema nacional de autopistas pero legalmente ese tramo de accidente suele ser propiedad de la agencia estatal de autopistas. Como este conjunto de características es el que más se repite entre los accidentes, esta regla se podría aplicar sobre el conjunto de datos nuevo para llevar a cabo una predicción.

En esta misma tarea, habría que determinar si se etiquetan correctamente los nuevos accidentes siguiendo la regla anterior, para ello se podría establecer una incertidumbre del 6% por lo tanto si los nuevos accidentes se clasifican con una probabilidad del 94%, entonces el modelo funciona, ya que está clasificando accidentes en zonas de riesgo, como accidentes en zonas de riesgo. No obstante en esta tarea se tendría que tener en cuenta también los falsos positivos, que son los accidentes que se clasifican de riesgo, cuando en realidad no son de riesgo y viceversa con los falsos negativos.

Finalmente, en el caso en el que el modelo funcionase correctamente, habría que interpretar los resultados. Esta es una de las tareas más importantes del proceso de minería de datos,

por lo tanto hay que poner mucha atención. En este paso, hay que hacer interpretaciones en profundidad. Aunque no se tienen resultados para poder discutirlos, se puede tomar como ejemplo de interpretación un resultado de un análisis anterior en el que he mencionado que el atributo CITYNAME estaba estrechamente relacionado con el atributo ENTORNO, en ese analisis he llegado a decir que si para algún accidente el nombre de la ciudad es NO APLICA, significa que el entorno dónde ha ocurrido el accidente es rural, pero esto no es veraz, ya que hay accidentes en los que el nombre de la ciuad es "NO APLICA" pero el entorno mencionado es urbano, esto puede parecer una contradicción, pero en el fondo puede que tenga una explicación más profunda. De esto trata este punto del proyecto, de interpretar los resultados, y de entender la relación entre los resultados y el objetivo del proyecto.

Estudiando los términos, se intuye que el conjunto relativo al de la validación del modelo, no debería de ser igual de grande que el conjunto de datos para evaluar el modelo, ya que validar el modelo consistiría solo en comprobar que el modelo funciona, luego evaluarlo, consistirá en comprobar que el modelo funciona a largo plazo y con cualquier tipo de accidentes. No obstante, los tres conjuntos tienen que venir de la misma base de datos.

Ahora bien, para poder determinar si el modelo se ha construido, validado y evaluado correctamente, se precisa de una métrica que permita clasificar cada uno de los tres procesos. Para empezar, lo que se va a hacer va a ser

INTEGRACIÓN DEL MODELO

Esta es la última fase del proyecto, y en ella se llevaría acabo la última tarea del proyecto que consistiría en integrar los resultados del proyecto de minería de datos en el proceso que se encarga de alimentar el modelo que se ah creado con nuevos accidentes, para cumplir con el objetivo del proyecto. Hay que tener en cuenta, que la integración podría suponer un cambio en el código desarrollado, no obstante, gran parte de los sistemas de minería de datos comerciales proporcionan estos servicios.

La última tarea consistiría en conocer cual es el lenguaje de programación en el que se tiene que implementar el modelo, y luego conocer todas las herramientas necesarias para poder realizar esa potencial trauducción de código.

Ejercicio 2

A partir del juego de datos utilizado en el ejemplo de la PEC, realiza las tareas previas a la generación de un modelo de minería de datos explicadas en los módulos "El proceso de minería de datos" y "Preprocesado de los datos y gestión de características". Puedes utilizar de referencia el ejemplo de la PEC, pero procura cambiar el enfoque y analizar los datos en función de las diferentes dimensiones que presentan los datos. Opcionalmente y valorable se pueden añadir al estudio datos de otros años para realizar comparaciones temporales (https://www.nhtsa.gov/file-downloads?p=nhtsa/downloads/FARS/) o añadir otros hechos a estudiar relacionados, por ejemplo, el consumo de drogas en los accidentes (https://static.nhtsa.gov/nhtsa/downloads/FARS/2020/National/FARS2020NationalCSV.zip)

Escribe aquí la respuesta a la pregunta:

Como en el anterior ejercicio ya se ha propuesto un proyecto de minería de datos, en este ejercicio se llevarán a cabo las tareas previas a la generación de un modelo de minería de datos pero con el foco puesto en el objetivo mencionado en el anterior ejercicio.

```
# Redacta aquí el código R para el estudio del juego de datos

# Se carga a continuación el fichero de datos
```

```r
nombre_archivo = 'accident.CSV'
#Ahora se lee el fichero en formato csv
datos <- read.csv(nombre_archivo)

# Se comprueba la estructura del conjunto de datos
structure = str(datos)
```

```
## 'data.frame':    35766 obs. of  81 variables:
##  $ STATE       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ STATENAME   : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ ST_CASE     : int  10001 10002 10003 10004 10005 10006 10007 10008 10009 10010 ...
##  $ VE_TOTAL    : int  1 4 2 1 1 2 1 2 2 2 ...
##  $ VE_FORMS    : int  1 4 2 1 1 2 1 2 2 2 ...
##  $ PVH_INVL    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PEDS        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ PERSONS     : int  4 6 2 5 1 3 1 2 4 3 ...
##  $ PERMVIT     : int  4 6 2 5 1 3 1 2 4 3 ...
##  $ PERNOTMVIT  : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ COUNTY      : int  51 73 117 15 37 103 73 25 45 95 ...
##  $ COUNTYNAME  : chr  "ELMORE (51)" "JEFFERSON (73)" "SHELBY (117)" "CALHOUN (15)" ...
##  $ CITY        : int  0 350 0 0 0 0 330 0 0 1500 ...
##  $ CITYNAME    : chr  "NOT APPLICABLE" "BIRMINGHAM" "NOT APPLICABLE" "NOT APPLICABLE" ...
##  $ DAY         : int  1 2 2 3 4 4 7 8 9 10 ...
##  $ DAYNAME     : int  1 2 2 3 4 4 7 8 9 10 ...
##  $ MONTH       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ MONTHNAME   : chr  "January" "January" "January" "January" ...
##  $ YEAR        : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ DAY_WEEK    : int  4 5 5 6 7 7 3 4 5 6 ...
##  $ DAY_WEEKNAME: chr  "Wednesday" "Thursday" "Thursday" "Friday" ...
##  $ HOUR        : int  2 17 14 15 0 16 19 7 20 10 ...
##  $ HOURNAME    : chr  "2:00am-2:59am" "5:00pm-5:59pm" "2:00pm-2:59pm" "3:00pm-3:59pm" ...
##  $ MINUTE      : int  58 18 55 20 45 55 23 15 0 2 ...
##  $ MINUTENAME  : chr  "58" "18" "55" "20" ...
##  $ NHS         : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ NHSNAME     : chr  "This section IS NOT on the NHS" "This section IS NOT on the NHS" "This sectio
##  $ ROUTE       : int  4 6 3 4 4 3 4 4 4 2 ...
##  $ ROUTENAME   : chr  "County Road" "Local Street - Municipality" "State Highway" "County Road" ...
##  $ TWAY_ID     : chr  "cr-4" "martin luther king jr dr" "sr-76" "CR-ALEXANDRIA WELLINGTON RD" ...
##  $ TWAY_ID2    : chr  "" "" "us-280" "" ...
##  $ RUR_URB     : int  1 2 1 1 1 1 2 1 1 1 ...
##  $ RUR_URBNAME : chr  "Rural" "Urban" "Rural" "Rural" ...
##  $ FUNC_SYS    : int  5 4 4 7 5 4 4 5 5 3 ...
##  $ FUNC_SYSNAME: chr  "Major Collector" "Minor Arterial" "Minor Arterial" "Local" ...
##  $ RD_OWNER    : int  2 4 1 2 2 1 4 2 2 1 ...
##  $ RD_OWNERNAME: chr  "County Highway Agency" "City or Municipal Highway Agency" "State Highway Agen
##  $ MILEPT      : int  0 0 49 0 0 390 0 0 0 3019 ...
##  $ MILEPTNAME  : chr  "None" "None" "49" "None" ...
##  $ LATITUDE    : num  32.4 33.5 33.3 33.8 32.8 ...
##  $ LATITUDENAME: chr  "32.43313333" "33.48465833" "33.29994167" "33.79507222" ...
##  $ LONGITUD    : num  -86.1 -86.8 -86.4 -85.9 -86.1 ...
##  $ LONGITUDNAME: chr  "-86.09485" "-86.83954444" "-86.36964167" "-85.88348611" ...
##  $ SP_JUR      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SP_JURNAME  : chr  "No Special Jurisdiction" "No Special Jurisdiction" "No Special Jurisdiction"
##  $ HARM_EV     : int  42 12 34 42 42 12 8 12 12 12 ...
```

```
##  $ HARM_EVNAME : chr  "Tree (Standing Only)" "Motor Vehicle In-Transport" "Ditch" "Tree (Standing Onl
##  $ MAN_COLL    : int  0 6 0 0 0 2 0 1 1 2 ...
##  $ MAN_COLLNAME: chr  "The First Harmful Event was Not a Collision with a Motor Vehicle in Transport
##  $ RELJCT1     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RELJCT1NAME : chr  "No" "No" "No" "No" ...
##  $ RELJCT2     : int  1 1 3 1 1 1 3 1 8 1 ...
##  $ RELJCT2NAME : chr  "Non-Junction" "Non-Junction" "Intersection-Related" "Non-Junction" ...
##  $ TYP_INT     : int  1 1 3 1 1 1 2 1 1 1 ...
##  $ TYP_INTNAME : chr  "Not an Intersection" "Not an Intersection" "T-Intersection" "Not an Intersecti
##  $ WRK_ZONE    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WRK_ZONENAME: chr  "None" "None" "None" "None" ...
##  $ REL_ROAD    : int  4 1 4 4 4 1 1 1 1 1 ...
##  $ REL_ROADNAME: chr  "On Roadside" "On Roadway" "On Roadside" "On Roadside" ...
##  $ LGT_COND    : int  2 3 1 1 2 2 3 1 2 1 ...
##  $ LGT_CONDNAME: chr  "Dark - Not Lighted" "Dark - Lighted" "Daylight" "Daylight" ...
##  $ WEATHER     : int  1 2 2 10 2 1 1 1 10 10 ...
##  $ WEATHERNAME : chr  "Clear" "Rain" "Rain" "Cloudy" ...
##  $ SCH_BUS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SCH_BUSNAME : chr  "No" "No" "No" "No" ...
##  $ RAIL        : chr  "0000000" "0000000" "0000000" "0000000" ...
##  $ RAILNAME    : chr  "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
##  $ NOT_HOUR    : int  99 17 14 99 0 17 19 7 20 10 ...
##  $ NOT_HOURNAME: chr  "Unknown" "5:00pm-5:59pm" "2:00pm-2:59pm" "Unknown" ...
##  $ NOT_MIN     : int  99 18 58 99 45 0 23 21 0 3 ...
##  $ NOT_MINNAME : chr  "Unknown" "18" "58" "Unknown" ...
##  $ ARR_HOUR    : int  3 17 15 99 0 17 19 7 20 10 ...
##  $ ARR_HOURNAME: chr  "3:00am-3:59am" "5:00pm-5:59pm" "3:00pm-3:59pm" "Unknown EMS Scene Arrival Hou
##  $ ARR_MIN     : int  10 26 15 99 55 19 29 28 10 7 ...
##  $ ARR_MINNAME : chr  "10" "26" "15" "Unknown EMS Scene Arrival Minutes" ...
##  $ HOSP_HR     : int  99 99 99 99 88 18 88 88 99 10 ...
##  $ HOSP_HRNAME : chr  "Unknown" "Unknown" "Unknown" "Unknown" ...
##  $ HOSP_MN     : int  99 99 99 99 88 51 88 88 99 29 ...
##  $ HOSP_MNNAME : chr  "Unknown EMS Hospital Arrival Time" "Unknown EMS Hospital Arrival Time" "Unknou
##  $ FATALS      : int  3 1 1 1 1 1 1 1 1 1 ...
##  $ DRUNK_DR    : int  1 0 0 0 0 0 0 0 0 0 ...
```

Se puede observar como la base de datos tiene 35766 registros y 81 variables. Esto e salgo que ya se conocía del anterior por el pequeño análisis que se realizó, no obstante nunca viene mal comprobar de nuevo las cosas.

Observando el output obtenido antes, puede notarse como la base de datos tiene muchos registros, es decir; muchos accidentes, en este caso hay 35766 accidentes. Cada accidente tiene 81 atributos, que reflejan desde el estado de EEUU en el que se ha producido el accidente, hasta el número de personas implicadas.

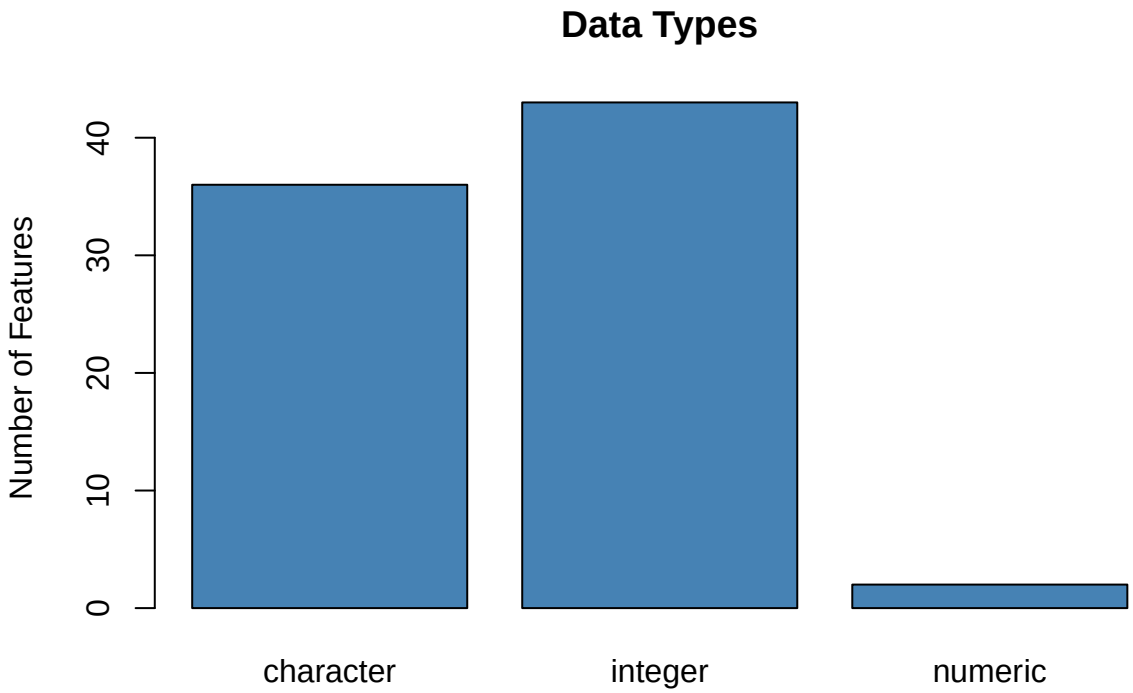
Adicionalmente, se ha podido observar en el output anterior, los tipos de variables del dataset, visualmente se podría decir que predominan los atributos de tipo enteros, no obtante tambiñen hay atributos con valores tipo chr, pero como esto no es un razonamiento objetivo, esto se comprueba a continuación:

```
sapply(datos, class)
```

```
##        STATE    STATENAME      ST_CASE     VE_TOTAL     VE_FORMS     PVH_INVL
##    "integer"  "character"    "integer"    "integer"    "integer"    "integer"
##         PEDS      PERSONS      PERMVIT   PERNOTMVIT       COUNTY   COUNTYNAME
##    "integer"    "integer"    "integer"    "integer"    "integer"  "character"
```

```
##         CITY     CITYNAME          DAY      DAYNAME        MONTH    MONTHNAME
##    "integer"  "character"    "integer"    "integer"    "integer"  "character"
##         YEAR     DAY_WEEK DAY_WEEKNAME         HOUR     HOURNAME       MINUTE
##    "integer"    "integer"  "character"    "integer"  "character"    "integer"
##   MINUTENAME          NHS      NHSNAME        ROUTE    ROUTENAME      TWAY_ID
##  "character"    "integer"  "character"    "integer"  "character"  "character"
##     TWAY_ID2      RUR_URB  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER
##  "character"    "integer"  "character"    "integer"  "character"    "integer"
## RD_OWNERNAME       MILEPT   MILEPTNAME     LATITUDE LATITUDENAME     LONGITUD
##  "character"    "integer"  "character"    "numeric"  "character"    "numeric"
## LONGITUDNAME       SP_JUR   SP_JURNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##  "character"    "integer"  "character"    "integer"  "character"    "integer"
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##  "character"    "integer"  "character"    "integer"  "character"    "integer"
##  TYP_INTNAME     WRK_ZONE WRK_ZONENAME     REL_ROAD REL_ROADNAME     LGT_COND
##  "character"    "integer"  "character"    "integer"  "character"    "integer"
## LGT_CONDNAME      WEATHER  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL
##  "character"    "integer"  "character"    "integer"  "character"  "character"
##     RAILNAME     NOT_HOUR NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR
##  "character"    "integer"  "character"    "integer"  "character"    "integer"
## ARR_HOURNAME      ARR_MIN  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN
##  "character"    "integer"  "character"    "integer"  "character"    "integer"
##  HOSP_MNNAME       FATALS     DRUNK_DR
##  "character"    "integer"    "integer"
```

```r
#A continuación hacemos una pequeña gráfica que muestre visualmente:
tipos_de_datos <- function(datos) {
  res <- lapply(datos, class)
  res_frame <- data.frame(unlist(res))
  barplot(table(res_frame), main="Data Types", col="steelblue", ylab="Number of Features")
}

tipos_de_datos(datos)
```



Como se esperaba, hay más variables enteras que de tipo chr, no obstante, algo que a grandes rasgos no se percibía, era que los atributos LATITUDE y LONGITUDE son variables de tipo NUMERIC, pero esto tiene lógica, porque son valores geográficos decimales y el lenguaje de programación R los llama NUMERIC. Más adelante, se verá si merece la pena o no discretizar estos dos atributos.

Antes de empezar a aplicar las técnicas de preprocesado de datos, habría que familiarizarse con la base de datos. Para ello, se ha estudiado el significado de los atritbutos relativos al archivo *accident.csv* en el archivo **"FARS Analytical Users Manual 1975-2021"**. Esta base de datos cuenta con los accidentes que se producieron en el 202. Además, cabe destacar que ya se empezó con el estudio de la base de datos, en el ejercicio anterior anterior, a fin poder proponer el objetivo principal del proyecto de minería de datos.

Teniendo una idea general del significado de los atributos, ahora se procede a la comprobación de variables vacías y nulas en la base de datos, véase la siguiente celda de código:

```
print('VALORES NULOS')
```

```
## [1] "VALORES NULOS"
```

```
colSums(is.na(datos))
```

```
##        STATE    STATENAME      ST_CASE     VE_TOTAL     VE_FORMS     PVH_INVL
##            0            0            0            0            0            0
##         PEDS      PERSONS      PERMVIT   PERNOTMVIT       COUNTY   COUNTYNAME
##            0            0            0            0            0            0
##         CITY     CITYNAME          DAY      DAYNAME        MONTH    MONTHNAME
##            0            0            0            0            0            0
##         YEAR     DAY_WEEK DAY_WEEKNAME         HOUR     HOURNAME       MINUTE
##            0            0            0            0            0            0
##   MINUTENAME          NHS      NHSNAME        ROUTE    ROUTENAME      TWAY_ID
##            0            0            0            0            0            0
##     TWAY_ID2      RUR_URB  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER
##            0            0            0            0            0            0
```

```
## RD_OWNERNAME       MILEPT   MILEPTNAME     LATITUDE LATITUDENAME     LONGITUD
##            0            0            0            0            0            0
## LONGITUDNAME       SP_JUR   SP_JURNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##            0            0            0            0            0            0
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##            0            0            0            0            0            0
##  TYP_INTNAME     WRK_ZONE WRK_ZONENAME     REL_ROAD REL_ROADNAME     LGT_COND
##            0            0            0            0            0            0
## LGT_CONDNAME      WEATHER  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL
##            0            0            0            0            0            0
##     RAILNAME     NOT_HOUR NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR
##            0            0            0            0            0            0
## ARR_HOURNAME      ARR_MIN  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN
##            0            0            0            0            0            0
##  HOSP_MNNAME       FATALS     DRUNK_DR
##            0            0            0
```

Como se puede comprobar, en lo que respecta a esta base de datos, no habrá que cumplimentar ningun campo, por medio de técnicas ya vistas en teoría como por ejemplo, por medio del cálculo de medias a partir de los valores del mismo atributo, pero para el resto de accidentes. Esto le evitará confusiones futuras al modelo que se desarrolle.

Ahora es el momento de calcular los valores nulos, véase el siguiente código:

```
print('VALORES VACÍOS')
```

```
## [1] "VALORES VACÍOS"
```

```
colSums(datos=='')
```

```
##        STATE    STATENAME      ST_CASE     VE_TOTAL     VE_FORMS     PVH_INVL
##            0            0            0            0            0            0
##         PEDS      PERSONS      PERMVIT   PERNOTMVIT       COUNTY   COUNTYNAME
##            0            0            0            0            0            0
##         CITY     CITYNAME          DAY      DAYNAME        MONTH    MONTHNAME
##            0            0            0            0            0            0
##         YEAR     DAY_WEEK DAY_WEEKNAME         HOUR     HOURNAME       MINUTE
##            0            0            0            0            0            0
##   MINUTENAME          NHS      NHSNAME        ROUTE    ROUTENAME      TWAY_ID
##            0            0            0            0            0            0
##     TWAY_ID2      RUR_URB  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER
##        26997            0            0            0            0            0
## RD_OWNERNAME       MILEPT   MILEPTNAME     LATITUDE LATITUDENAME     LONGITUD
##            0            0            0            0            0            0
## LONGITUDNAME       SP_JUR   SP_JURNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##            0            0            0            0            0            0
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##            0            0            0            0            0            0
##  TYP_INTNAME     WRK_ZONE WRK_ZONENAME     REL_ROAD REL_ROADNAME     LGT_COND
##            0            0            0            0            0            0
## LGT_CONDNAME      WEATHER  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL
##            0            0            0            0            0            0
##     RAILNAME     NOT_HOUR NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR
##            0            0            0            0            0            0
## ARR_HOURNAME      ARR_MIN  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN
##            0            0            0            0            0            0
```

```
##  HOSP_MNNAME       FATALS     DRUNK_DR
##            0            0            0
```

Como se puede ver arriba, el atributo **TWAY_ID2** tiene 26997 celdas vacías, esto significa que para 26997 accidentes, el atrinuto **TWAY_ID2** no aporta ninguna información. Este atributo representa el identificador de la vía de circulación. Este atributo registra la vía de circulación en la que se ha producido el accidente y como el objetivo propuesto para el proyecto, es el de saber que zonas geográficas acumulan más accidentes y que las caracterizan, para en un futuro, poder predecir zonas de riesgo de accidente, hay que recordar que este atributo cumple con una característica principal, y es que se trata de un atributo de carácter geográfico, es decir, este atributo da información acerca de la vía en la que se ha producido el accidente, por lo tanto este atributo podría resultar interesante, pero hay que tener en cuenta que falta el 75.48% de los datos de este atributo, por lo tanto va a resultar muy dificil reconstruir todos los valores que faltan y puede ser mu malo para el modelo. No obstante se va a realizar igualmente el estudio.

Primero se buscan los posibles valores que puede tomar este atributo. Por el output de la celda relativa al tipo de datos, puede verse como se trata de un atributo con valores tipo chr. Ahora hay que conocer los valores que puede tomar este atributo, para ello se recurre al pdf que describe los atributos, en él se vé, que como mínimo, los valores de este presentan 10 caractéres, y según los accidentes son más recientes, este atributo puede llegar a a estar formado por cifras de hasta 30 caracteres, por lo tanto resultaría imposible realizar alguna combinación o cálculo de la media para el resto de accidentes, con el fin de poblar las 26997 celdas de este atributo. Cabe destacar que se trata de un atributo de carácter identificativo, no obstante, se representan a continuación sus valores, para verificar la alta variabilidad de datos que puede haber.

Para ello se inspecciona los aspectos del conjunto de valores del atributo **TWAY_ID2**:

```
print('información del atributo TWAY_ID2')
```

```
## [1] "información del atributo TWAY_ID2"
```

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if(!require('Rmisc')) install.packages('Rmisc'); library('Rmisc')
if(!require('dplyr')) install.packages('dplyr'); library('dplyr')
if(!require('xfun')) install.packages('xfun'); library('xfun')

summary(datos[c("TWAY_ID2")])
```

```
##    TWAY_ID2
##  Length:35766
##  Class :character
##  Mode  :character
```

```
head(datos[c("TWAY_ID2")]) #Visuzalizamos las primeras filas
```

```
##   TWAY_ID2
## 1
## 2
## 3   us-280
## 4
## 5
## 6
```

```
tail(datos[c("TWAY_ID2")]) #Visuzalizamos las primeras filas
```

```
##       TWAY_ID2
## 35761   SR-339
## 35762
```

```
## 35763
## 35764
## 35765
## 35766
```

Como se puede ver en el output de arriba, faltan muchos valores en ese atributo y parece tratarse de valores alfanuméricos, no obstante tras no obtener ninguna información interesante relativa a este atributo, se ha abierto el archivo csv en Excel y se ha estudiado a vista, que valores poblan este atributo, y para sorpresa nuestra, aparecen datos alafanuméricos del estilo ab-xx, donde ab son letras (acrónimos) del tipo de carretera y xx son números, pero también aparecen nombres de calles sin número con varios formatos diferentes. Esta claro que la gran falta de valores de este atributo, y la diversidad de los mismos, hacen imposible la reconstrucción del atributo, por lo tanto, no se va a utilizar para el estudio. Esto signfica que se podría eliminar del data set, y es lo que se hace en la siguiente celda:

```
drop <- c("TWAY_ID2")
datos = datos[,!(names(datos) %in% drop)]
sort(colnames(datos)) #COMPROBAMOS QUE LO HEMOS ELIMINADO
```

```
##  [1] "ARR_HOUR"     "ARR_HOURNAME" "ARR_MIN"      "ARR_MINNAME"  "CITY"
##  [6] "CITYNAME"     "COUNTY"       "COUNTYNAME"   "DAY"          "DAY_WEEK"
## [11] "DAY_WEEKNAME" "DAYNAME"      "DRUNK_DR"     "FATALS"       "FUNC_SYS"
## [16] "FUNC_SYSNAME" "HARM_EV"      "HARM_EVNAME"  "HOSP_HR"      "HOSP_HRNAME"
## [21] "HOSP_MN"      "HOSP_MNNAME"  "HOUR"         "HOURNAME"     "LATITUDE"
## [26] "LATITUDENAME" "LGT_COND"     "LGT_CONDNAME" "LONGITUD"     "LONGITUDNAME"
## [31] "MAN_COLL"     "MAN_COLLNAME" "MILEPT"       "MILEPTNAME"   "MINUTE"
## [36] "MINUTENAME"   "MONTH"        "MONTHNAME"    "NHS"          "NHSNAME"
## [41] "NOT_HOUR"     "NOT_HOURNAME" "NOT_MIN"      "NOT_MINNAME"  "PEDS"
## [46] "PERMVIT"      "PERNOTMVIT"   "PERSONS"      "PVH_INVL"     "RAIL"
## [51] "RAILNAME"     "RD_OWNER"     "RD_OWNERNAME" "REL_ROAD"     "REL_ROADNAME"
## [56] "RELJCT1"      "RELJCT1NAME"  "RELJCT2"      "RELJCT2NAME"  "ROUTE"
## [61] "ROUTENAME"    "RUR_URB"      "RUR_URBNAME"  "SCH_BUS"      "SCH_BUSNAME"
## [66] "SP_JUR"       "SP_JURNAME"   "ST_CASE"      "STATE"        "STATENAME"
## [71] "TWAY_ID"      "TYP_INT"      "TYP_INTNAME"  "VE_FORMS"     "VE_TOTAL"
## [76] "WEATHER"      "WEATHERNAME"  "WRK_ZONE"     "WRK_ZONENAME" "YEAR"
```

Como se puede ver arriba, ya se ha eliminado ese atributo. Ahora bien, para adelantar las labores de limpieza, lo que se podría hacer en esta misma tarea, sería repetir lo que ya se ha hecho pero para las base de datos relativa a los accidentes ocurridos en el 2019 y en el 2021. Pues a lo largo de este ejercicio, se pretenden llevar a cabo comparaciones. Véase la siguiente celda de código:

```
print('ACCIDENTES 2019')
```

```
## [1] "ACCIDENTES 2019"
```

```
nombre_archivo2 = 'accident2019.CSV'

#Ahora se lee el fichero en formato csv
datos2019 <- read.csv(nombre_archivo2)

# Se comprueba la estructura del conjunto de datos para el fichero csv
structure2 = str(datos2019)
```

```
## 'data.frame':    33487 obs. of  91 variables:
##  $ STATE       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ STATENAME   : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ ST_CASE     : int  10001 10002 10003 10004 10005 10006 10007 10008 10009 10010 ...
```

```
##  $ VE_TOTAL    : int  2 2 3 1 1 2 1 1 1 1 ...
##  $ VE_FORMS    : int  2 2 3 1 1 2 1 1 1 1 ...
##  $ PVH_INVL    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PEDS        : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ PERSONS     : int  3 2 4 1 1 2 5 1 1 1 ...
##  $ PERMVIT     : int  3 2 4 1 1 2 5 1 1 1 ...
##  $ PERNOTMVIT  : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ COUNTY      : int  81 55 29 55 3 85 55 69 3 101 ...
##  $ COUNTYNAME  : chr  "LEE (81)" "ETOWAH (55)" "CLEBURNE (29)" "ETOWAH (55)" ...
##  $ CITY        : int  2340 1280 0 2562 0 0 0 790 1493 2130 ...
##  $ CITYNAME    : chr  "OPELIKA" "GADSDEN" "NOT APPLICABLE" "RAINBOW CITY" ...
##  $ DAY         : int  7 23 22 22 18 7 2 9 8 7 ...
##  $ DAYNAME     : int  7 23 22 22 18 7 2 9 8 7 ...
##  $ MONTH       : int  2 1 1 1 1 1 1 2 2 2 ...
##  $ MONTHNAME   : chr  "February" "January" "January" "January" ...
##  $ YEAR        : int  2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 ...
##  $ DAY_WEEK    : int  5 4 3 3 6 2 4 7 6 5 ...
##  $ DAY_WEEKNAME: chr  "Thursday" "Wednesday" "Tuesday" "Tuesday" ...
##  $ HOUR        : int  12 18 19 3 5 12 9 21 7 5 ...
##  $ HOURNAME    : chr  "12:00pm-12:59pm" "6:00pm-6:59pm" "7:00pm-7:59pm" "3:00am-3:59am" ...
##  $ MINUTE      : int  54 3 0 15 50 30 25 2 53 56 ...
##  $ MINUTENAME  : chr  "54" "3" "0" "15" ...
##  $ NHS         : int  1 1 1 1 1 1 1 0 0 0 ...
##  $ NHSNAME     : chr  "This section IS ON the NHS" "This section IS ON the NHS" "This section IS ON
##  $ ROUTE       : int  1 1 1 1 1 1 1 4 6 6 ...
##  $ ROUTENAME   : chr  "Interstate" "Interstate" "Interstate" "Interstate" ...
##  $ TWAY_ID     : chr  "I-85" "I-759" "I-20" "I-59" ...
##  $ TWAY_ID2    : chr  "" "" "" "" ...
##  $ RUR_URB     : int  2 2 1 1 2 1 1 1 2 2 ...
##  $ RUR_URBNAME : chr  "Urban" "Urban" "Rural" "Rural" ...
##  $ FUNC_SYS    : int  1 1 1 1 1 1 1 7 4 5 ...
##  $ FUNC_SYSNAME: chr  "Interstate" "Interstate" "Interstate" "Interstate" ...
##  $ RD_OWNER    : int  1 1 1 1 1 1 1 2 4 4 ...
##  $ RD_OWNERNAME: chr  "State Highway Agency" "State Highway Agency" "State Highway Agency" "State Hig
##  $ MILEPT      : int  641 15 2118 1775 413 1567 1826 0 0 0 ...
##  $ MILEPTNAME  : chr  "641" "15" "2118" "1775" ...
##  $ LATITUDE    : num  32.7 34 33.7 34 30.7 ...
##  $ LATITUDENAME: chr  "32.66622222" "33.99782778" "33.66084167" "33.95647222" ...
##  $ LONGITUD    : num  -85.3 -86.1 -85.4 -86.1 -87.8 ...
##  $ LONGITUDNAME: chr  "-85.33665833" "-86.05399722" "-85.39101111" "-86.14052222" ...
##  $ SP_JUR      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SP_JURNAME  : chr  "No Special Jurisdiction" "No Special Jurisdiction" "No Special Jurisdiction"
##  $ HARM_EV     : int  12 12 12 8 1 12 32 1 3 8 ...
##  $ HARM_EVNAME : chr  "Motor Vehicle In-Transport" "Motor Vehicle In-Transport" "Motor Vehicle In-Tra
##  $ MAN_COLL    : int  1 1 1 0 0 7 0 0 0 0 ...
##  $ MAN_COLLNAME: chr  "Front-to-Rear" "Front-to-Rear" "Front-to-Rear" "The First Harmful Event was No
##  $ RELJCT1     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RELJCT1NAME : chr  "No" "No" "No" "No" ...
##  $ RELJCT2     : int  1 1 1 1 1 1 1 1 3 1 ...
##  $ RELJCT2NAME : chr  "Non-Junction" "Non-Junction" "Non-Junction" "Non-Junction" ...
##  $ TYP_INT     : int  1 1 1 1 1 1 1 1 3 1 ...
##  $ TYP_INTNAME : chr  "Not an Intersection" "Not an Intersection" "Not an Intersection" "Not an Inter
##  $ WRK_ZONE    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WRK_ZONENAME: chr  "None" "None" "None" "None" ...
```

```
##  $ REL_ROAD    : int  1 1 1 1 4 1 3 1 4 1 ...
##  $ REL_ROADNAME: chr  "On Roadway" "On Roadway" "On Roadway" "On Roadway" ...
##  $ LGT_COND    : int  1 2 2 2 2 1 1 2 1 3 ...
##  $ LGT_CONDNAME: chr  "Daylight" "Dark - Not Lighted" "Dark - Not Lighted" "Dark - Not Lighted" ...
##  $ WEATHER1    : int  1 2 10 1 5 1 10 1 1 1 ...
##  $ WEATHER1NAME: chr  "Clear" "Rain" "Cloudy" "Clear" ...
##  $ WEATHER2    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WEATHER2NAME: chr  "No Additional Atmospheric Conditions" "No Additional Atmospheric Conditions" "
##  $ WEATHER     : int  1 2 10 1 5 1 10 1 1 1 ...
##  $ WEATHERNAME : chr  "Clear" "Rain" "Cloudy" "Clear" ...
##  $ SCH_BUS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SCH_BUSNAME : chr  "No" "No" "No" "No" ...
##  $ RAIL        : chr  "0000000" "0000000" "0000000" "0000000" ...
##  $ RAILNAME    : chr  "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
##  $ NOT_HOUR    : int  12 18 19 3 99 12 9 88 7 88 ...
##  $ NOT_HOURNAME: chr  "12:00pm-12:59pm" "6:00pm-6:59pm" "7:00pm-7:59pm" "3:00am-3:59am" ...
##  $ NOT_MIN     : int  59 3 2 4 99 30 21 88 53 88 ...
##  $ NOT_MINNAME : chr  "59" "3" "2" "4" ...
##  $ ARR_HOUR    : int  13 18 19 3 6 12 9 88 7 88 ...
##  $ ARR_HOURNAME: chr  "1:00pm-1:59pm" "6:00pm-6:59pm" "7:00pm-7:59pm" "3:00am-3:59am" ...
##  $ ARR_MIN     : int  9 7 12 11 0 54 24 88 59 88 ...
##  $ ARR_MINNAME : chr  "9" "7" "12" "11" ...
##  $ HOSP_HR     : int  13 99 20 88 88 88 99 88 88 88 ...
##  $ HOSP_HRNAME : chr  "1:00pm-1:59pm" "Unknown" "8:00pm-8:59pm" "Not Applicable (Not Transported)" .
##  $ HOSP_MN     : int  27 99 5 88 88 88 99 88 88 88 ...
##  $ HOSP_MNNAME : chr  "27" "Unknown EMS Hospital Arrival Time" "5" "Not Applicable (Not Transported)"
##  $ CF1         : int  0 0 14 0 0 0 0 0 0 0 ...
##  $ CF1NAME     : chr  "None" "None" "Motor Vehicle struck by falling cargo,or something that came lo
##  $ CF2         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CF2NAME     : chr  "None" "None" "None" "None" ...
##  $ CF3         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CF3NAME     : chr  "None" "None" "None" "None" ...
##  $ FATALS      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DRUNK_DR    : int  1 0 0 0 1 0 0 1 0 0 ...
```

En este caso hay 91 variables/atributos frente a las 81 que había en el archivo del 2020. Ahora se comprueban los valores NULOS y los VACÍOS

```
print('VALORES NULOS 2019')
```

```
## [1] "VALORES NULOS 2019"
```

```
colSums(is.na(datos2019))
```

```
##        STATE    STATENAME      ST_CASE     VE_TOTAL     VE_FORMS     PVH_INVL
##            0            0            0            0            0            0
##         PEDS      PERSONS      PERMVIT   PERNOTMVIT       COUNTY   COUNTYNAME
##            0            0            0            0            0            0
##         CITY     CITYNAME          DAY      DAYNAME        MONTH    MONTHNAME
##            0            0            0            0            0            0
##         YEAR     DAY_WEEK DAY_WEEKNAME         HOUR     HOURNAME       MINUTE
##            0            0            0            0            0            0
##   MINUTENAME          NHS      NHSNAME        ROUTE    ROUTENAME      TWAY_ID
##            0            0            0            0            0            0
##     TWAY_ID2      RUR_URB  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER
##            0            0            0            0            0            0
```

```
## RD_OWNERNAME       MILEPT   MILEPTNAME     LATITUDE LATITUDENAME     LONGITUD
##            0            0            0            0            0            0
## LONGITUDNAME       SP_JUR   SP_JURNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##            0            0            0            0            0            0
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##            0            0            0            0            0            0
##  TYP_INTNAME     WRK_ZONE WRK_ZONENAME     REL_ROAD REL_ROADNAME     LGT_COND
##            0            0            0            0            0            0
## LGT_CONDNAME     WEATHER1 WEATHER1NAME     WEATHER2 WEATHER2NAME      WEATHER
##            0            0            0            0            0            0
##  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL     RAILNAME     NOT_HOUR
##            0            0            0            0            0            0
## NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR ARR_HOURNAME      ARR_MIN
##            0            0            0            0            0            0
##  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN  HOSP_MNNAME          CF1
##            0            0            0            0            0            0
##      CF1NAME          CF2      CF2NAME          CF3      CF3NAME       FATALS
##            0            0            0            0            0            0
##     DRUNK_DR
##            0
```

Al igual que en la base de datos del 2020, no hay ningún valor nulo.

```
print('VALORES VACÍOS 2019')
```

```
## [1] "VALORES VACÍOS 2019"
```

```
colSums(datos2019=='')
```

```
##        STATE    STATENAME      ST_CASE     VE_TOTAL     VE_FORMS     PVH_INVL
##            0            0            0            0            0            0
##         PEDS      PERSONS      PERMVIT   PERNOTMVIT       COUNTY   COUNTYNAME
##            0            0            0            0            0            0
##         CITY     CITYNAME          DAY      DAYNAME        MONTH    MONTHNAME
##            0            0            0            0            0            0
##         YEAR     DAY_WEEK DAY_WEEKNAME         HOUR     HOURNAME       MINUTE
##            0            0            0            0            0            0
##   MINUTENAME          NHS      NHSNAME        ROUTE    ROUTENAME      TWAY_ID
##            0            0            0            0            0            0
##     TWAY_ID2      RUR_URB  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER
##        24963            0            0            0            0            0
## RD_OWNERNAME       MILEPT   MILEPTNAME     LATITUDE LATITUDENAME     LONGITUD
##            0            0            0            0            0            0
## LONGITUDNAME       SP_JUR   SP_JURNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##            0            0            0            0            0            0
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##            0            0            0            0            0            0
##  TYP_INTNAME     WRK_ZONE WRK_ZONENAME     REL_ROAD REL_ROADNAME     LGT_COND
##            0            0            0            0            0            0
## LGT_CONDNAME     WEATHER1 WEATHER1NAME     WEATHER2 WEATHER2NAME      WEATHER
##            0            0            0            0            0            0
##  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL     RAILNAME     NOT_HOUR
##            0            0            0            0            0            0
## NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR ARR_HOURNAME      ARR_MIN
##            0            0            0            0            0            0
##  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN  HOSP_MNNAME          CF1
```

```
##           0           0           0           0           0           0
##      CF1NAME         CF2      CF2NAME         CF3      CF3NAME      FATALS
##           0           0           0           0           0           0
##     DRUNK_DR
##           0
```

Al igual que en la base de datos del 2020, solo faltan valores en un solo atributo; el **TWAY_ID2**, esta gran falta de valores en ese atributo, es lo que ya se ha visto en la base de datos de los accidentes curridos en el 2020.

Ahora se realiza la misma tarea, pero para los accidentes que han ocurrido en 2021

```
print('ACCIDENTES 2021')
```

```
## [1] "ACCIDENTES 2021"
```

```
nombre_archivo3 = 'accident2021.CSV'

#Ahora se lee el fichero en formato csv
datos2021 <- read.csv(nombre_archivo3)

# Se comprueba la estructura del conjunto de datos para el fichero csv
structure3 = str(datos2021)
```

```
## 'data.frame':    39508 obs. of  80 variables:
##  $ STATE       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ STATENAME   : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ ST_CASE     : int  10001 10002 10003 10004 10005 10006 10007 10008 10009 10010 ...
##  $ PEDS        : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ PERNOTMVIT  : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ VE_TOTAL    : int  2 1 1 1 2 3 1 2 1 3 ...
##  $ VE_FORMS    : int  2 1 1 1 2 3 1 2 1 3 ...
##  $ PVH_INVL    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PERSONS     : int  3 2 1 1 4 3 1 2 2 3 ...
##  $ PERMVIT     : int  3 2 1 1 4 3 1 2 2 3 ...
##  $ COUNTY      : int  115 73 73 117 73 1 83 125 101 73 ...
##  $ COUNTYNAME  : chr  "ST. CLAIR (115)" "JEFFERSON (73)" "JEFFERSON (73)" "SHELBY (117)" ...
##  $ CITY        : int  0 0 0 0 0 0 0 3050 2130 0 ...
##  $ CITYNAME    : chr  "NOT APPLICABLE" "NOT APPLICABLE" "NOT APPLICABLE" "NOT APPLICABLE" ...
##  $ MONTH       : int  2 2 2 2 1 1 1 1 1 1 ...
##  $ MONTHNAME   : chr  "February" "February" "February" "February" ...
##  $ DAY         : int  12 11 7 3 30 28 27 23 17 20 ...
##  $ DAYNAME     : int  12 11 7 3 30 28 27 23 17 20 ...
##  $ DAY_WEEK    : int  6 5 1 4 7 5 4 7 1 4 ...
##  $ DAY_WEEKNAME: chr  "Friday" "Thursday" "Sunday" "Wednesday" ...
##  $ YEAR        : int  2021 2021 2021 2021 2021 2021 2021 2021 2021 2021 ...
##  $ HOUR        : int  22 18 0 16 22 12 22 2 0 0 ...
##  $ HOURNAME    : chr  "10:00pm-10:59pm" "6:00pm-6:59pm" "0:00am-0:59am" "4:00pm-4:59pm" ...
##  $ MINUTE      : int  10 0 20 20 20 15 25 44 32 1 ...
##  $ MINUTENAME  : chr  "10" "0" "20" "20" ...
##  $ TWAY_ID     : chr  "I-20" "I-459" "I-459" "I-65" ...
##  $ TWAY_ID2    : chr  "" "" "" "" ...
##  $ ROUTE       : int  1 1 1 1 1 1 1 3 1 1 ...
##  $ ROUTENAME   : chr  "Interstate" "Interstate" "Interstate" "Interstate" ...
##  $ RUR_URB     : int  2 2 2 2 1 2 1 2 2 2 ...
##  $ RUR_URBNAME : chr  "Urban" "Urban" "Urban" "Urban" ...
##  $ FUNC_SYS    : int  1 1 1 1 1 1 1 3 1 1 ...
```

```
##  $ FUNC_SYSNAME: chr  "Interstate" "Interstate" "Interstate" "Interstate" ...
##  $ RD_OWNER    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ RD_OWNERNAME: chr  "State Highway Agency" "State Highway Agency" "State Highway Agency" "State Hig
##  $ NHS         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ NHSNAME     : chr  "This section IS ON the NHS" "This section IS ON the NHS" "This section IS ON t
##  $ SP_JUR      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SP_JURNAME  : chr  "No Special Jurisdiction" "No Special Jurisdiction" "No Special Jurisdiction" "
##  $ MILEPT      : int  1569 293 183 2300 2820 0 3595 25 20 242 ...
##  $ MILEPTNAME  : chr  "1569" "293" "183" "2300" ...
##  $ LATITUDE    : num  33.6 33.5 33.4 33.4 33.8 ...
##  $ LATITUDENAME: num  33.6 33.5 33.4 33.4 33.8 ...
##  $ LONGITUD    : num  -86.3 -86.6 -86.8 -86.8 -86.8 ...
##  $ LONGITUDNAME: num  -86.3 -86.6 -86.8 -86.8 -86.8 ...
##  $ HARM_EV     : int  12 25 8 34 12 12 34 12 1 12 ...
##  $ HARM_EVNAME : chr  "Motor Vehicle In-Transport" "Concrete Traffic Barrier" "Pedestrian" "Ditch" .
##  $ MAN_COLL    : int  2 0 0 0 1 1 0 2 0 6 ...
##  $ MAN_COLLNAME: chr  "Front-to-Front" "The First Harmful Event was Not a Collision with a Motor Vehi
##  $ RELJCT1     : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ RELJCT1NAME : chr  "No" "No" "No" "No" ...
##  $ RELJCT2     : int  1 1 1 1 1 1 1 2 19 1 ...
##  $ RELJCT2NAME : chr  "Non-Junction" "Non-Junction" "Non-Junction" "Non-Junction" ...
##  $ TYP_INT     : int  1 1 1 1 1 1 1 2 1 1 ...
##  $ TYP_INTNAME : chr  "Not an Intersection" "Not an Intersection" "Not an Intersection" "Not an Inter
##  $ REL_ROAD    : int  1 3 2 4 1 1 4 1 8 1 ...
##  $ REL_ROADNAME: chr  "On Roadway" "On Median" "On Shoulder" "On Roadside" ...
##  $ WRK_ZONE    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WRK_ZONENAME: chr  "None" "None" "None" "None" ...
##  $ LGT_COND    : int  2 2 2 1 2 1 2 3 3 2 ...
##  $ LGT_CONDNAME: chr  "Dark - Not Lighted" "Dark - Not Lighted" "Dark - Not Lighted" "Daylight" ...
##  $ WEATHER     : int  2 2 2 1 10 1 1 1 1 2 ...
##  $ WEATHERNAME : chr  "Rain" "Rain" "Rain" "Clear" ...
##  $ SCH_BUS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SCH_BUSNAME : chr  "No" "No" "No" "No" ...
##  $ RAIL        : chr  "0000000" "0000000" "0000000" "0000000" ...
##  $ RAILNAME    : chr  "Not Applicable" "Not Applicable" "Not Applicable" "Not Applicable" ...
##  $ NOT_HOUR    : int  22 99 9 16 22 12 22 2 99 0 ...
##  $ NOT_HOURNAME: chr  "10:00pm-10:59pm" "Unknown" "9:00am-9:59am" "4:00pm-4:59pm" ...
##  $ NOT_MIN     : int  13 99 29 20 20 17 30 44 99 1 ...
##  $ NOT_MINNAME : chr  "13" "Unknown" "29" "20" ...
##  $ ARR_HOUR    : int  22 19 9 16 22 12 22 2 1 0 ...
##  $ ARR_HOURNAME: chr  "10:00pm-10:59pm" "7:00pm-7:59pm" "9:00am-9:59am" "4:00pm-4:59pm" ...
##  $ ARR_MIN     : int  25 9 40 28 30 23 54 51 0 15 ...
##  $ ARR_MINNAME : chr  "25" "9" "40" "28" ...
##  $ HOSP_HR     : int  23 88 88 99 88 12 88 88 99 99 ...
##  $ HOSP_HRNAME : chr  "11:00pm-11:59pm" "Not Applicable (Not Transported)" "Not Applicable (Not Trans
##  $ HOSP_MN     : int  2 88 88 99 88 53 88 88 99 99 ...
##  $ HOSP_MNNAME : chr  "2" "Not Applicable (Not Transported)" "Not Applicable (Not Transported)" "Unkn
##  $ FATALS      : int  2 2 1 1 1 1 1 1 1 1 ...
```

En comparación a las bases del 2019 y el 2020, como se puede ver arriba, la base de datos del 2021 tiene 11 atributos menos que el fichero del 2019 y uno menos que el fichero del 2020. Ahora se comprueban los valores NULOS y los VACÍOS

```
print('VALORES NULOS 2021')
```

```
## [1] "VALORES NULOS 2021"
```

```
colSums(is.na(datos2021))
```

```
##        STATE    STATENAME      ST_CASE         PEDS   PERNOTMVIT     VE_TOTAL
##            0            0            0            0            0            0
##     VE_FORMS     PVH_INVL      PERSONS      PERMVIT       COUNTY   COUNTYNAME
##            0            0            0            0            0            0
##         CITY     CITYNAME        MONTH    MONTHNAME          DAY      DAYNAME
##            0            0            0            0            0            0
##     DAY_WEEK DAY_WEEKNAME         YEAR         HOUR     HOURNAME       MINUTE
##            0            0            0            0            0            0
##   MINUTENAME      TWAY_ID     TWAY_ID2        ROUTE    ROUTENAME      RUR_URB
##            0            0            0            0            0            0
##  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER RD_OWNERNAME          NHS
##            0            0            0            0            0            0
##      NHSNAME       SP_JUR   SP_JURNAME       MILEPT   MILEPTNAME     LATITUDE
##            0            0            0            0            0            0
## LATITUDENAME     LONGITUD LONGITUDNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##            0            0            0            0            0            0
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##            0            0            0            0            0            0
##  TYP_INTNAME     REL_ROAD REL_ROADNAME     WRK_ZONE WRK_ZONENAME     LGT_COND
##            0            0            0            0            0            0
## LGT_CONDNAME      WEATHER  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL
##            0            0            0            0            0            0
##     RAILNAME     NOT_HOUR NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR
##            0            0            0            0            0            0
## ARR_HOURNAME      ARR_MIN  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN
##            0            0            0            0            0            0
##  HOSP_MNNAME       FATALS
##            0            0
```

Ningún valor NULO en ninguno de los atributos, al igual que en los archivos del 2019 y del 2020.

```
print('VALORES VACÍOS 2021')
```

```
## [1] "VALORES VACÍOS 2021"
```

```
colSums(datos2021=='')
```

```
##        STATE    STATENAME      ST_CASE         PEDS   PERNOTMVIT     VE_TOTAL
##            0            0            0            0            0            0
##     VE_FORMS     PVH_INVL      PERSONS      PERMVIT       COUNTY   COUNTYNAME
##            0            0            0            0            0            0
##         CITY     CITYNAME        MONTH    MONTHNAME          DAY      DAYNAME
##            0            0            0            0            0            0
##     DAY_WEEK DAY_WEEKNAME         YEAR         HOUR     HOURNAME       MINUTE
##            0            0            0            0            0            0
##   MINUTENAME      TWAY_ID     TWAY_ID2        ROUTE    ROUTENAME      RUR_URB
##            0            0        29859            0            0            0
##  RUR_URBNAME     FUNC_SYS FUNC_SYSNAME     RD_OWNER RD_OWNERNAME          NHS
##            0            0            0            0            0            0
##      NHSNAME       SP_JUR   SP_JURNAME       MILEPT   MILEPTNAME     LATITUDE
```

```
##            0            0            0            0            0            0
## LATITUDENAME     LONGITUD LONGITUDNAME      HARM_EV  HARM_EVNAME     MAN_COLL
##            0            0            0            0            0            0
## MAN_COLLNAME      RELJCT1  RELJCT1NAME      RELJCT2  RELJCT2NAME      TYP_INT
##            0            0            0            0            0            0
##  TYP_INTNAME     REL_ROAD REL_ROADNAME     WRK_ZONE WRK_ZONENAME     LGT_COND
##            0            0            0            0            0            0
## LGT_CONDNAME      WEATHER  WEATHERNAME      SCH_BUS  SCH_BUSNAME         RAIL
##            0            0            0            0            0            0
##     RAILNAME     NOT_HOUR NOT_HOURNAME      NOT_MIN  NOT_MINNAME     ARR_HOUR
##            0            0            0            0            0            0
## ARR_HOURNAME      ARR_MIN  ARR_MINNAME      HOSP_HR  HOSP_HRNAME      HOSP_MN
##            0            0            0            0            0            0
##  HOSP_MNNAME       FATALS
##            0            0
```

Como se puede comprobar, el mismo problema que había con el atributo **TWAY_ID2** en las dos bases anteriores, se repite también en la base de accidentes del 2021

Como el problema del atributo mencionado también está en las bases del 2019 y del 2021, entonces se procede a eliminar ese atributo de esas dos bases de datos:

```
# Primero para la base de datos del 2019:
drop2 <- c("TWAY_ID2")
datos2019 = datos2019[,!(names(datos2019) %in% drop2)]
sort(colnames(datos2019)) #COMPROBAMOS QUE LO HEMOS ELIMINADO
```

```
##  [1] "ARR_HOUR"     "ARR_HOURNAME" "ARR_MIN"      "ARR_MINNAME"  "CF1"
##  [6] "CF1NAME"      "CF2"          "CF2NAME"      "CF3"          "CF3NAME"
## [11] "CITY"         "CITYNAME"     "COUNTY"       "COUNTYNAME"   "DAY"
## [16] "DAY_WEEK"     "DAY_WEEKNAME" "DAYNAME"      "DRUNK_DR"     "FATALS"
## [21] "FUNC_SYS"     "FUNC_SYSNAME" "HARM_EV"      "HARM_EVNAME"  "HOSP_HR"
## [26] "HOSP_HRNAME"  "HOSP_MN"      "HOSP_MNNAME"  "HOUR"         "HOURNAME"
## [31] "LATITUDE"     "LATITUDENAME" "LGT_COND"     "LGT_CONDNAME" "LONGITUD"
## [36] "LONGITUDNAME" "MAN_COLL"     "MAN_COLLNAME" "MILEPT"       "MILEPTNAME"
## [41] "MINUTE"       "MINUTENAME"   "MONTH"        "MONTHNAME"    "NHS"
## [46] "NHSNAME"      "NOT_HOUR"     "NOT_HOURNAME" "NOT_MIN"      "NOT_MINNAME"
## [51] "PEDS"         "PERMVIT"      "PERNOTMVIT"   "PERSONS"      "PVH_INVL"
## [56] "RAIL"         "RAILNAME"     "RD_OWNER"     "RD_OWNERNAME" "REL_ROAD"
## [61] "REL_ROADNAME" "RELJCT1"      "RELJCT1NAME"  "RELJCT2"      "RELJCT2NAME"
## [66] "ROUTE"        "ROUTENAME"    "RUR_URB"      "RUR_URBNAME"  "SCH_BUS"
## [71] "SCH_BUSNAME"  "SP_JUR"       "SP_JURNAME"   "ST_CASE"      "STATE"
## [76] "STATENAME"    "TWAY_ID"      "TYP_INT"      "TYP_INTNAME"  "VE_FORMS"
## [81] "VE_TOTAL"     "WEATHER"      "WEATHER1"     "WEATHER1NAME" "WEATHER2"
## [86] "WEATHER2NAME" "WEATHERNAME"  "WRK_ZONE"     "WRK_ZONENAME" "YEAR"
```

```
# Luego para la base de datos del 2019:
datos2021 = datos2021[,!(names(datos2021) %in% drop2)]
sort(colnames(datos2021)) #COMPROBAMOS QUE LO HEMOS ELIMINADO
```

```
##  [1] "ARR_HOUR"     "ARR_HOURNAME" "ARR_MIN"      "ARR_MINNAME"  "CITY"
##  [6] "CITYNAME"     "COUNTY"       "COUNTYNAME"   "DAY"          "DAY_WEEK"
## [11] "DAY_WEEKNAME" "DAYNAME"      "FATALS"       "FUNC_SYS"     "FUNC_SYSNAME"
## [16] "HARM_EV"      "HARM_EVNAME"  "HOSP_HR"      "HOSP_HRNAME"  "HOSP_MN"
## [21] "HOSP_MNNAME"  "HOUR"         "HOURNAME"     "LATITUDE"     "LATITUDENAME"
## [26] "LGT_COND"     "LGT_CONDNAME" "LONGITUD"     "LONGITUDNAME" "MAN_COLL"
## [31] "MAN_COLLNAME" "MILEPT"       "MILEPTNAME"   "MINUTE"       "MINUTENAME"
```

```
## [36] "MONTH"         "MONTHNAME"    "NHS"          "NHSNAME"      "NOT_HOUR"
## [41] "NOT_HOURNAME" "NOT_MIN"      "NOT_MINNAME"  "PEDS"         "PERMVIT"
## [46] "PERNOTMVIT"   "PERSONS"      "PVH_INVL"     "RAIL"         "RAILNAME"
## [51] "RD_OWNER"     "RD_OWNERNAME" "REL_ROAD"     "REL_ROADNAME" "RELJCT1"
## [56] "RELJCT1NAME"  "RELJCT2"      "RELJCT2NAME"  "ROUTE"        "ROUTENAME"
## [61] "RUR_URB"      "RUR_URBNAME"  "SCH_BUS"      "SCH_BUSNAME"  "SP_JUR"
## [66] "SP_JURNAME"   "ST_CASE"      "STATE"        "STATENAME"    "TWAY_ID"
## [71] "TYP_INT"      "TYP_INTNAME"  "VE_FORMS"     "VE_TOTAL"     "WEATHER"
## [76] "WEATHERNAME"  "WRK_ZONE"     "WRK_ZONENAME" "YEAR"
```

Ya se ha eliminado el atributo deseado, ahora el siguiente paso consiste en empezar a gráficar los atributos de interés, para empezar a entender las posibles relaciones entre ellos. Para ello se podría empezar con unos histogramas de los atributos relacionados con lugares o sitios de accidentes:

- **COUNTY**
- **CITY**
- **NHS**
- **ROUTE**

- **RUR_URB**
- **FUNC_SYS**
- **RD_OWNER**
- **LATITUDE**
- **LONGITUDE**
- **SP_JUR**
- **RELJCT1**
- **RELJCT2**
- **WRK_ZONE**
- **TYP_INT**
- **REL_ROAD**
- **RAIL**
- **LGT_COND**
- **WEATHER**
- **FATALS**
- **DRUNK_DR**
- **SCH_BUS**

Estos cinco últimos atributos no son atributos de carácter geográfico, pero se ha creido pertinente resaltarlos para después, poder relacionarlos con los atributos geográficos. Dicho esto, se comienzan a crear histogramas con atributos cuyos conjuntos de valores son menores:

```
unique(datos["RUR_URB"]) # Para saber cuantos valores posibles hay
```

```
##      RUR_URB
## 1          1
## 2          2
## 628        6
## 994        9
## 7021       8
```

```
unique(datos["WRK_ZONE"])
```

```
##      WRK_ZONE
## 1           0
## 140         2
## 164         1
```

```
## 964         4
## 3838        3
```

```
unique(datos["SCH_BUS"])
```

```
##    SCH_BUS
## 1        0
## 21       1
```

```
unique(datos["NHS"])
```

```
##     NHS
## 1     0
## 10    1
## 994   9
```

```
#Contamos cuantas veces aparecen cada una de los posibles valores dentro de cada variable, esto nos dara
table(datos$RUR_URB)
```

```
##
##     1     2     6     8     9
## 15033 20233    69   348    83
```

```
table(datos$WRK_ZONE)
```

```
##
##     0     1     2     3     4
## 34992   432    47     8   287
```

```
table(datos$SCH_BUS)
```

```
##
##     0     1
## 35715    51
```

```
table(datos$NHS)
```

```
##
##     0     1     9
## 20308 14763   695
```

```
names(which.max(table(datos$RUR_URBNAME)))
```

```
## [1] "Urban"
```

```
names(which.max(table(datos$WRK_ZONE)))
```

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

```
names(which.max(table(datos$SCH_BUS)))
```

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

```
names(which.max(table(datos$REL_ROAD))) # La carretera en la que más accidentes se han procudido, es una
```

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

```
names(which.max(table(datos$RELJCT2))) #De quien es la carretera en la que se ha dado el accidente
```

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

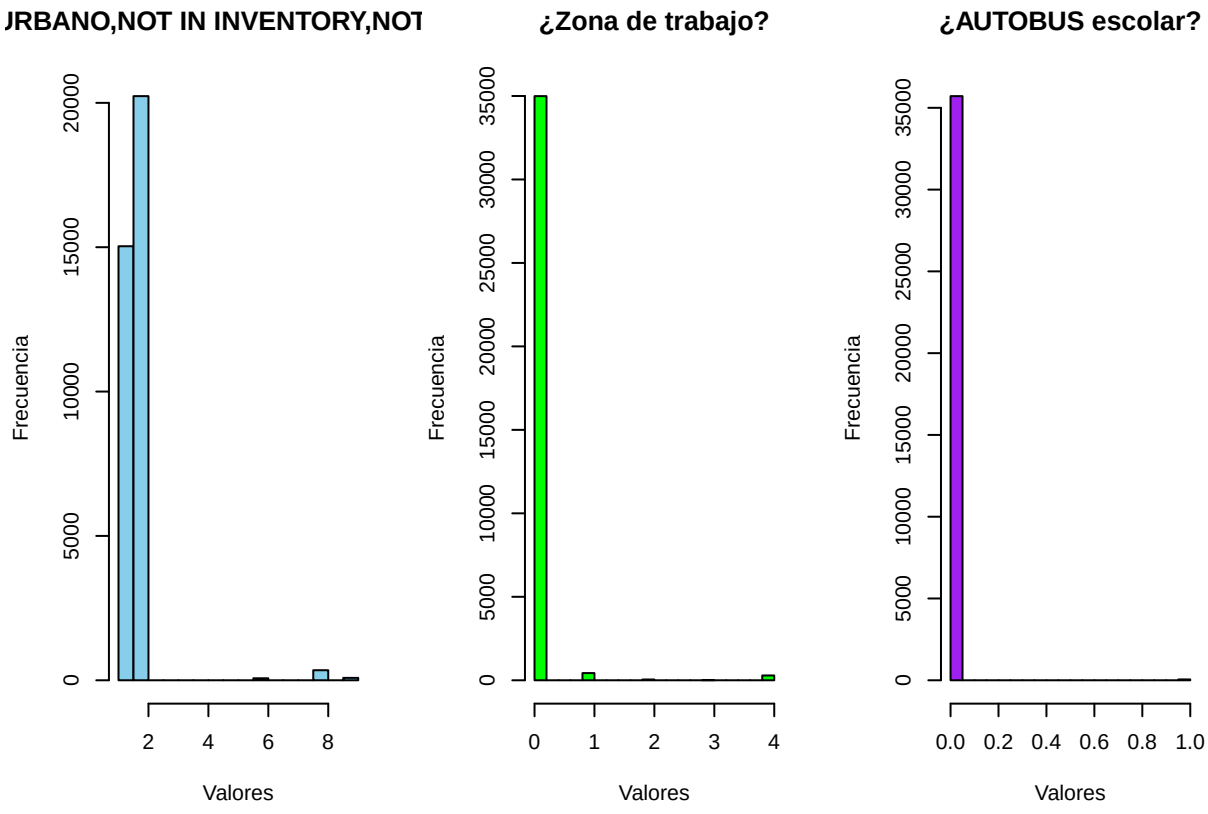
```
names(which.max(table(datos$ROUTE)))
```

```
## [1] "3"
```

```
names(which.max(table(datos$RD_OWNERNAME))) #De quien es la carretera en la que se ha dado el accidente
```

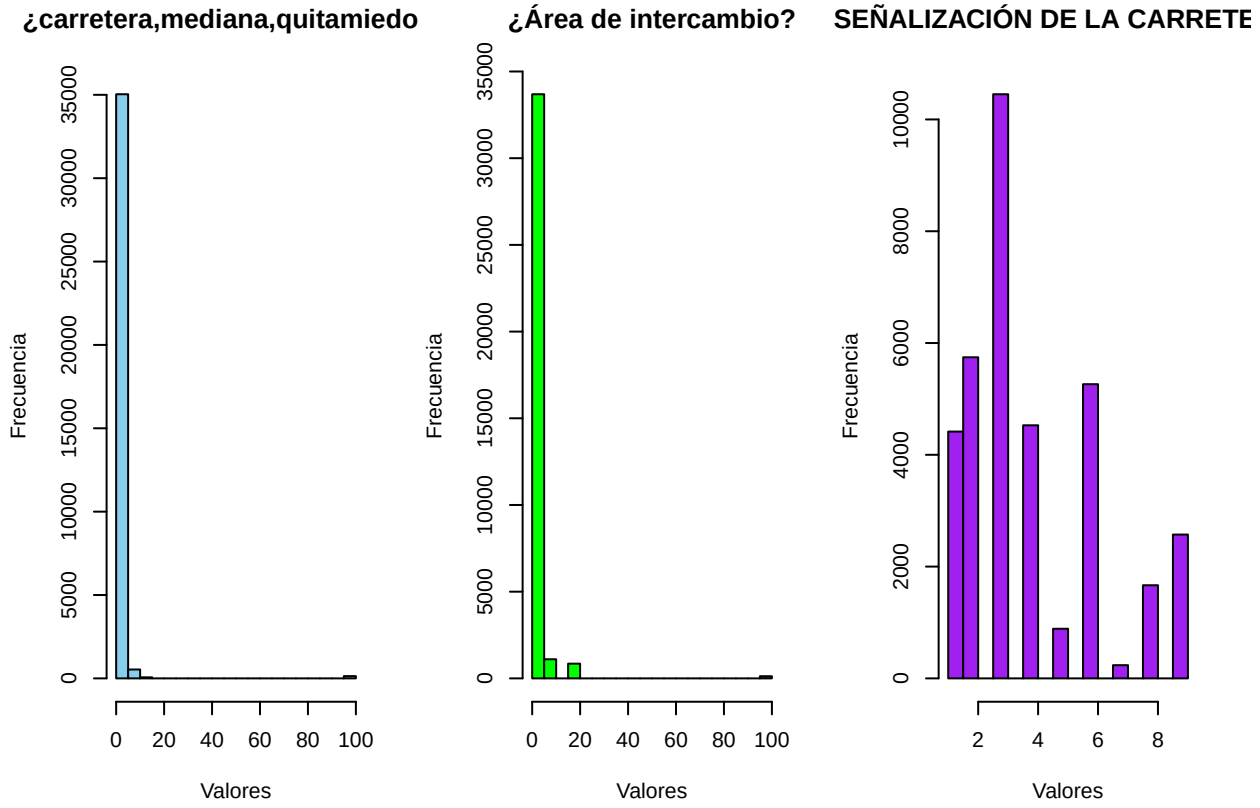
```
## [1] "State Highway Agency"
```

```
par(mfrow = c(1, 3))  # Dividir la ventana de gráficos en tres
#Ahora graficamos los histogramas
hist(datos$RUR_URB,main = "¿RURAL,URBANO,NOT IN INVENTORY,NOT REPORTED?",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "skyblue",
     border = "black")
hist(datos$WRK_ZONE,main = "¿Zona de trabajo?",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "green",
     border = "black")
hist(datos$SCH_BUS,main = "¿AUTOBUS escolar?",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "purple",
     border = "black")
```




```
par(mfrow = c(1, 3))  # Dividir la ventana de gráficos en tres

hist(datos$REL_ROAD,main = "¿carretera,mediana,quitamiedos?",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "skyblue",
```

```
     border = "black")
hist(datos$RELJCT2,main = "¿Área de intercambio?",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "green",
     border = "black")
hist(datos$ROUTE,main = "SEÑALIZACIÓN DE LA CARRETERA ",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "purple",
     border = "black")
```




```
hist(datos$RD_OWNER,main = "¿La carretera de quien es?",
     xlab = "Valores",
     ylab = "Frecuencia",
     col = "yellow",
     border = "black")

cat("Varianza atributo RUR_URB: ", var(datos["RUR_URB"]),"\n")
```

```
## Varianza atributo RUR_URB:  0.8009664
```

```
cat("Varianza atributo WRK_ZONE: ", var(datos["WRK_ZONE"]),"\n")
```

```
## Varianza atributo WRK_ZONE:  0.1454882
```

```
cat("Varianza atributo SCH_BUS: ", var(datos["SCH_BUS"]),"\n")
```

```
## Varianza atributo SCH_BUS:  0.001423942
```

```
cat("Varianza atributo REL_ROAD: ", var(datos["REL_ROAD"]),"\n")
```

```
## Varianza atributo REL_ROAD:  37.94969
```

```
cat("Varianza atributo RELJCT2: ", var(datos["RELJCT2"]),"\n")
```

```
## Varianza atributo RELJCT2:  41.06695
```

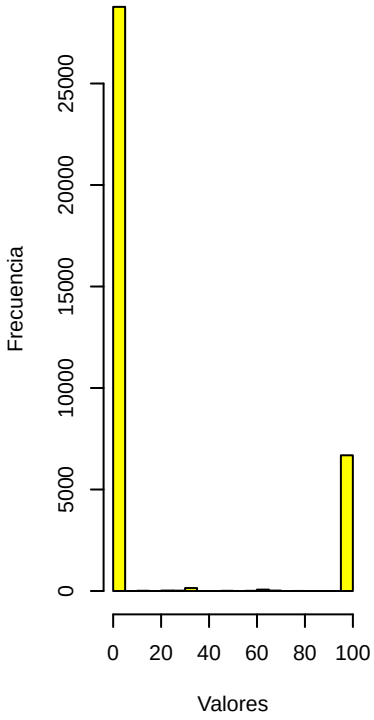
```
cat("Varianza atributo ROUTE: ", var(datos["ROUTE"]),"\n")
```

```
## Varianza atributo ROUTE:  5.252901
```

```
cat("Varianza atributo RD_OWNER: ", var(datos["RD_OWNER"]),"\n")
```

```
## Varianza atributo RD_OWNER:  1416.961
```

¿La carretera de quien es?



Como se puede observar por los histogramas, el mayor número de accidentes se da en los entornos urbanos (atributo RUR_URB de valor 4), seguido por los entornos rurales (atributo con valor 2). Luego, en términos de obras, practicamente la totalidad de los accidentes se dan en zonas donde no hay obras. Es importante destacar, que afortunadamente, practicamente en la totalidad de los accidentes, los autobuses escolares no se suelen ver envueltos en el accidente. Por lo tanto se podría decir que en la mayoría de entornos donde se dan los accidentes, no hay colegios cerca.

Analizando la segunda tanda de histogramas, en el primero de los tres, se muestra como de nuevo, practicamente la totalidad de los accidentes se dan en "roadway" es decir en la carretera (atributo de valor 1), luego una cantidad muy pequeña de accidentes se reparten entre el árcen y la mediana (valores 2 y 3), por último la infima cantidad restante de accidentes parecen no tener un reporte de la parte de la vía en la que se ha producido el choque. En el segundo histograma se observa como la mayoría de los accidentes no se dan en lugares sin intersecciones, no obstante luego hay una cantidad pequeña de accidentes que se dan en intersecciones o cerca de ellas. En el tercer histograma se observa un comportamiento mucho más variable, en el sentido de que el atributo

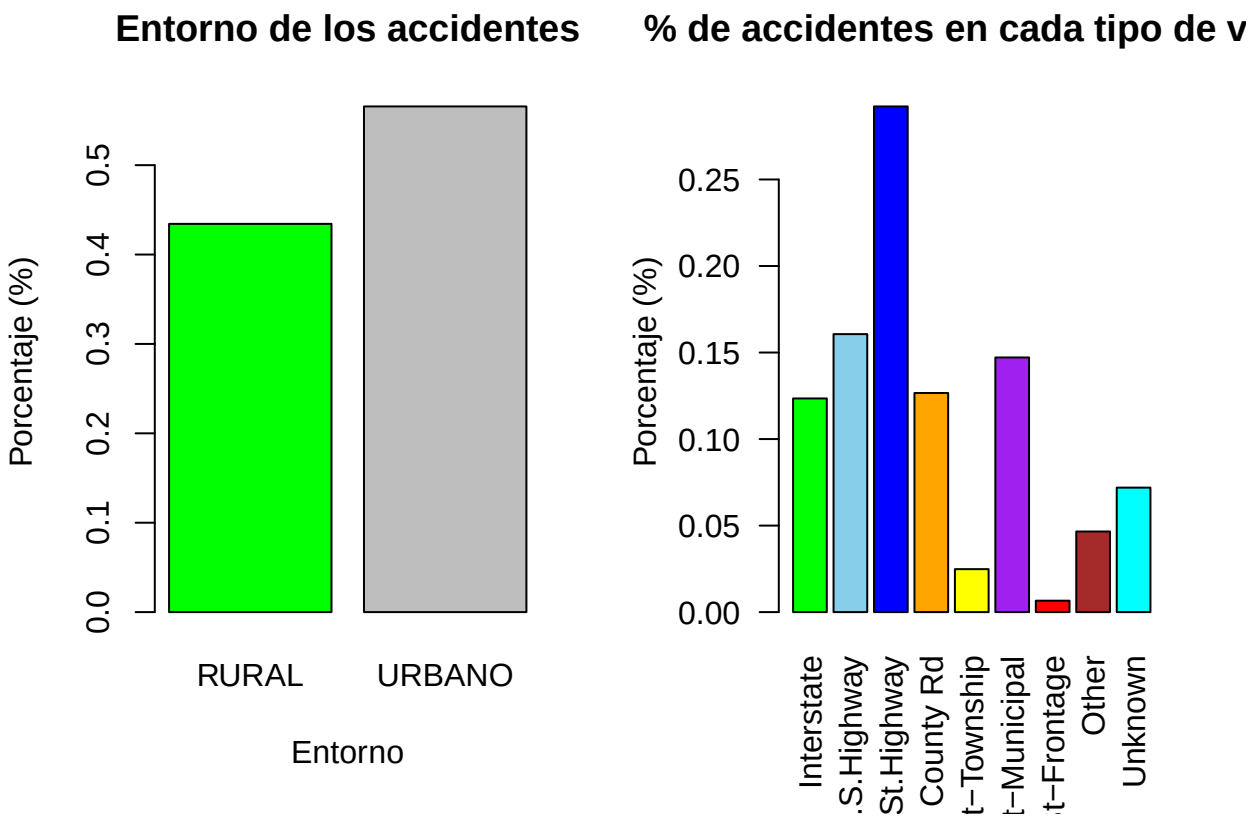
toma de manera distribuida valores dentro del conjunto posible, de hecho como puede verse por su cálculo de la varianza, tiene una varianza de 5 aproximadamente, en comparación a una varianza de 0 propia de los atributos vistos en la primera tanda de histogramas. En este histograma se observa como los accidentes suelen darse en mayor proporción en las autopistas estatales (atributo con valor 3), a continuación se encuentran las carreteras catalogadas como "U.S. Highway", luego en calles municipales, seguidamente se encuentran las carreteras condales, luego las interestatales y en el siguiente escalón se encuentran los accidentes que no tienen información de reporte acerca del tipo de vía.

Por último, el histograma en amarillo, representa el número de accidentes en carreteras según la entidad legal a la que pertenezcan la mismas. En dicho histograma puede verse como el mayor número de accidentes se dan en carreteras propiedad de la Agencia Estatal de Autopistas, seguidamente se encuentran una buena parte de accidentes que no tienen este atributo especificado.

Viendo el resultado de los histogramas, de esta fase puede concluirse, que la mayoría de los accidentes se dan en entornos urbanos, en carreteras propiedad de la Agencia Estatal de Autopistas (en carreteras estatales), el choque suele producirse en la misma carretera y no en el quitamiedos o fuera de la misma, y en los accidentes no suelen estar implicados los buses escolares, además los accidentes no suelen darse ni en zonas de obra o cerca de ellas, ni en interesecciones o próximas a ellas. El resultado tiene sentido, ya que cuantas más coches haya, más probabilidad de accidente, por eso el entorno urbano propicia un mayor número de accidentes, no obstante, teniendo esto en cuenta, es curioso ver como el tipo de vias que más aparece en los accidentes se da en las autopistas estatales, y no en calles, por lo tanto hay que profundizar en este campo. Además, el atributo **ROUTE** es el que presenta un comportamiento diferencial al resto.

Para ello se relaciona el atributo **RUR_URB** con **ROUTE**.

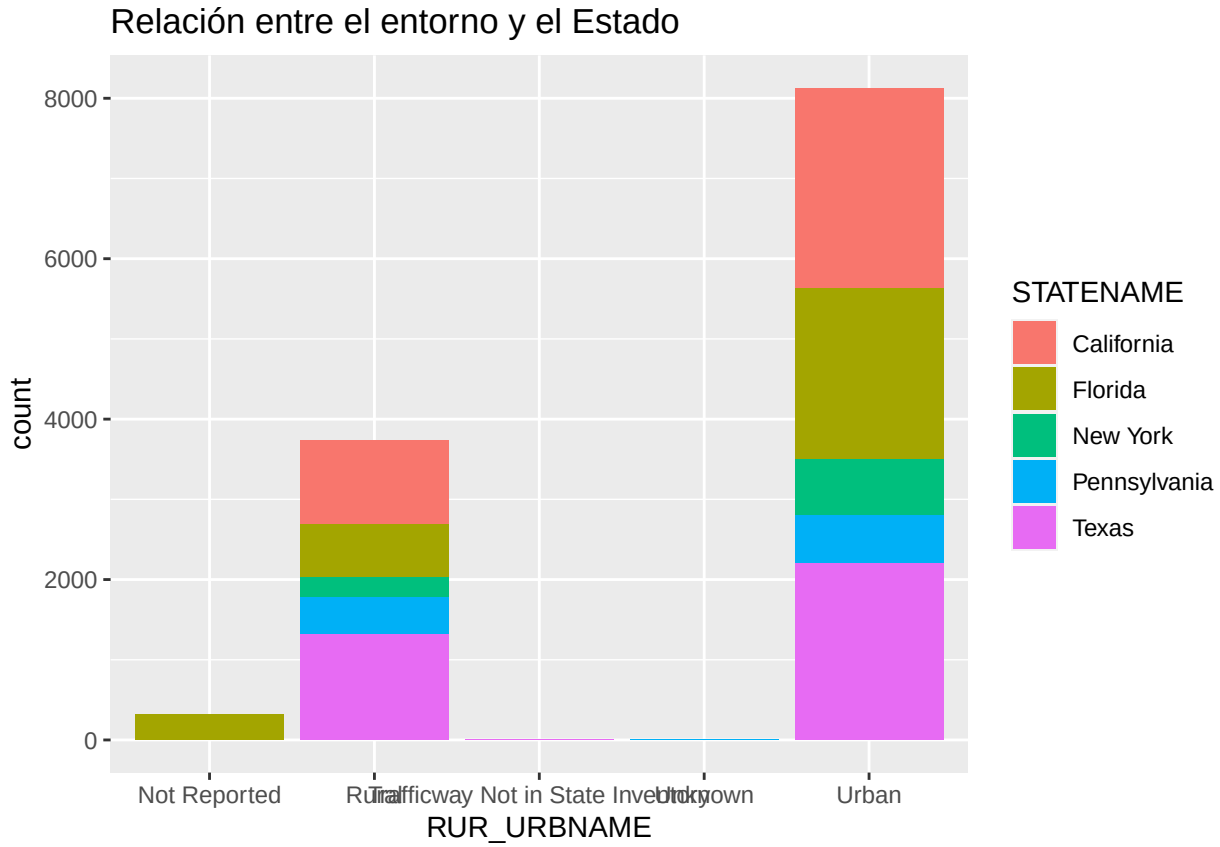
```
par(mfrow = c(1, 2))  # Dividir la ventana de gráficos en tres

datos$carreteras <- ifelse(datos$RUR_URB %in% c(2), 'URBANO', 'RURAL')
counts <- table(datos$carreteras)
barplot(prop.table(counts),col=c("green","grey"), main="Entorno de los accidentes",xlab ="Entorno", ylab

counts <- table(datos$ROUTE)
barplot(prop.table(counts),col=c("green","skyblue","blue","orange","yellow","purple","red","brown","cyan
```



Como se puede observar, aproximadamente el 55% de los accidentes se dan en un entorno urbano, y un 45% en un entorno rural. Por razones de simplcidad, se han obviado los otros valores del conjunto posible de valores que puede tomar el atributo entorno, porque su repercusión en el porcentaje total es infima. A la derecha se observa el porcentaje de accidentes en cad tipo de vía, y como ya se vió en anteriores gráficos, el mayor número de acccidentes (aproximadamente el 30% del total de los accidentes) se dan en autopistas estatales, seguidamente, el 17% se dan autopistas categroziadas como "US Highway" y en el tercer puesto con un 15% de accidentes, se encuentran las calles locales de municipios.

No obstante, este es el incio del proceso, aún hay que relacionar los atributos entre sí, para poder sacar conclusiones más vinculantes. Ahora se va a relacionar el entorno de los accidentes con los estados en los que se han producido los accidentes, para ello se han seleccionado los estados con mayor población en EEUU.

```
datosESTADOS=subset(datos, datos$STATENAME == "California" | datos$STATENAME == "Texas" | datos$STATENAM

files=dim(datosESTADOS)[1]
ggplot(data=datosESTADOS[1:files,],aes(x=RUR_URBNAME,fill=STATENAME))+geom_bar()+ggtitle("Relación entre
```

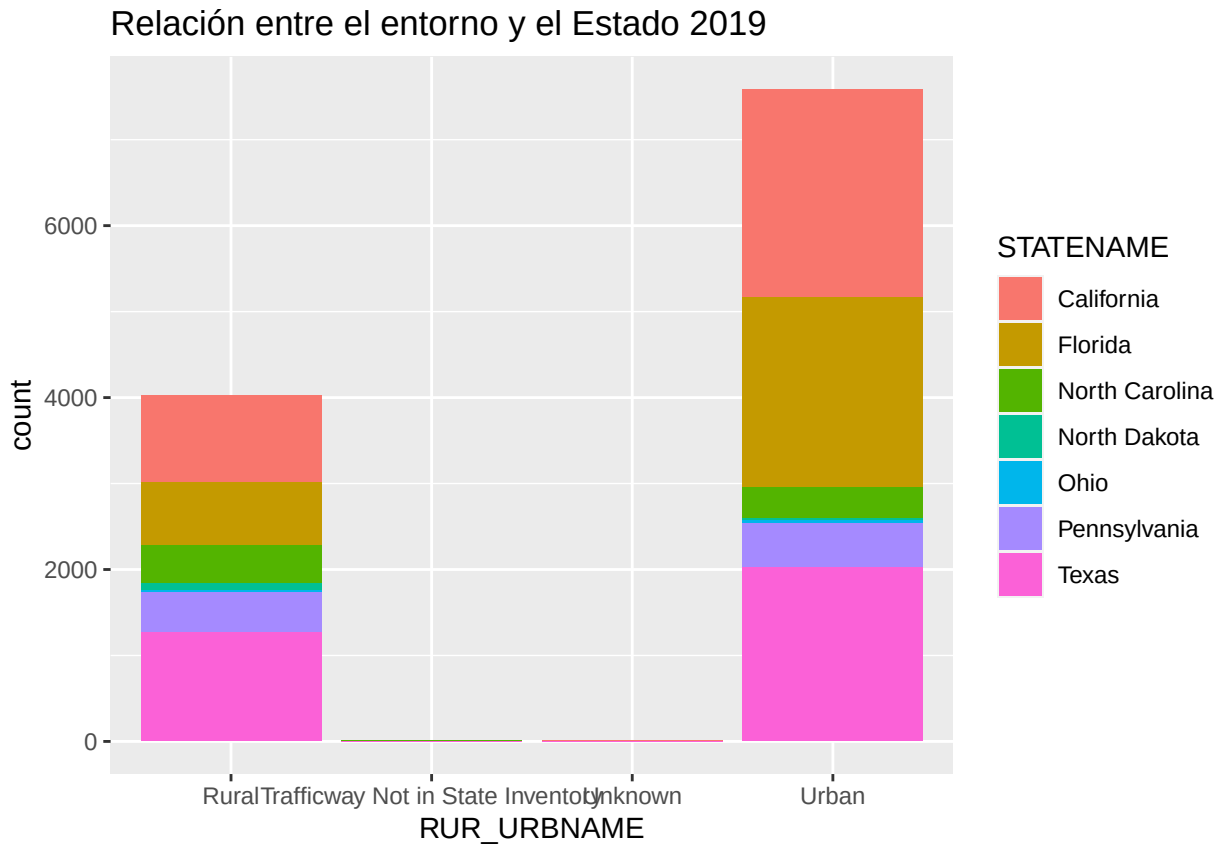


Cabe destacar que, hay una gran diferencia entre lso dos entornos, mientras que en el anterior gráfico estaban más cercanos el uno del otro, esto es porque esta vez solo se han seleccionado un grupo de estados en concreto, que además restultan ser los m´sa poblados del país. Por eso se nota esa diferencia.

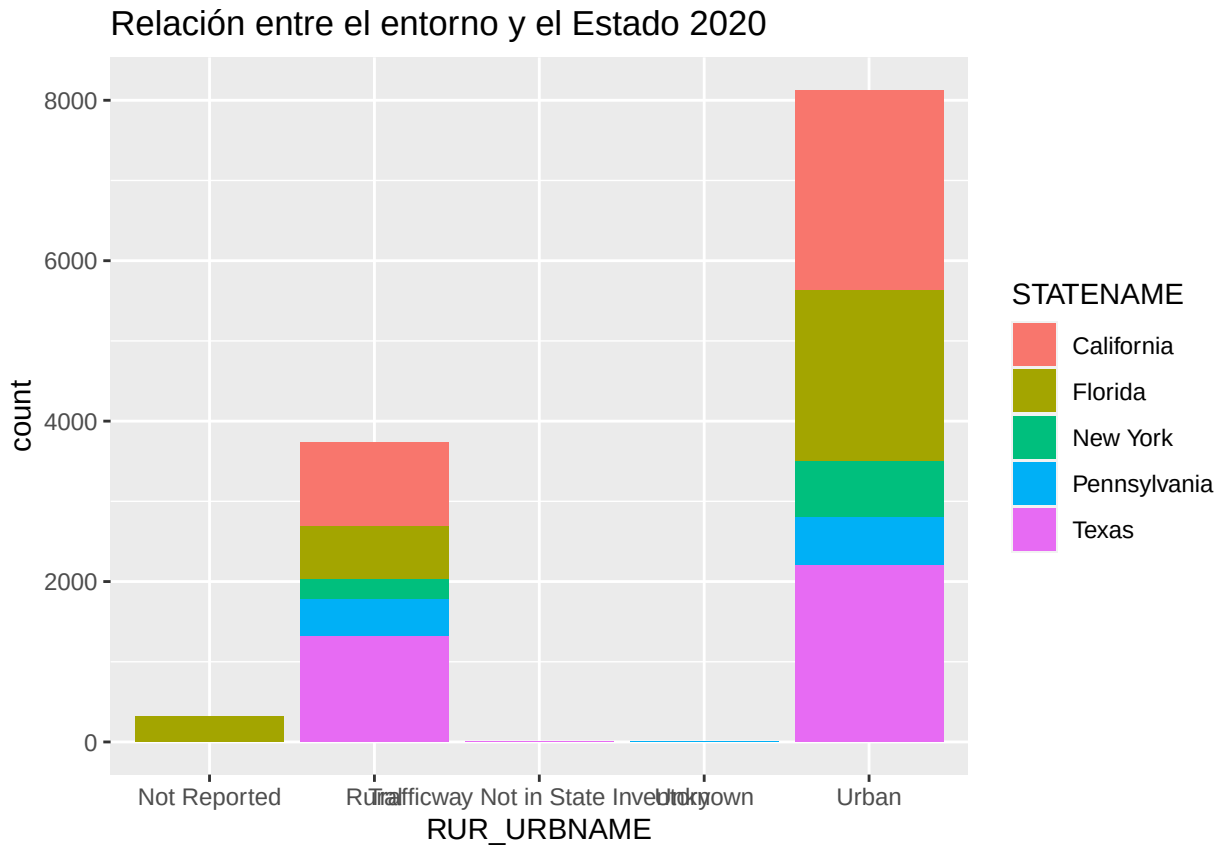
En los dos entornos principales puede apreciarse el mismo comportamiento jerárquico, el estado que más accidentes tiene es California, luego la Florida, luego Nueva York, seguidamente se encuentra Pennsylvania, y por último Texas. Era de esperar que California fuese el estado con más accidentes, más aun en zonas urbanas, no obstante, el estado de FLorida tiene un mayor número de accidentes que el estado de Texas teniendo en cuenta sus respectivas poblaciones. en 2022 la población de la Florida se estimó en 22,244,823 personas, frente a los 30,029,572 habitantes de Texas, esto implica una diferencia de población de 8 millones de personas, que equivale aproximadamente a la población de la Comunidad de Madrid. El hecho de que Texas tenga "tan pocos accidentes" teniendo en cuenta su población y comparándola con el estado de la FLorida y sus accidentes, precisa de una aclaración, por ello, se va a hacer una comparativa con las otras dos bases de datos, respectivas a los años 2019 y 2021, para ver si este comportamiento se mantiene en el tiempo. Véase la siguiente celda de código:

```
par(mfrow = c(1, 3))  # Dividir la ventana de gráficos en tres

# ggplot(                                                    # begin ggplot!
#     mapping = aes(x = datos2019$RUR_URBNAME, fill = datos2019$STATENAME))+
#   geom_bar() +
#   theme_classic() +
#   labs(
#     x = "RUR_URB",
#     y = "counts"
#   )
```

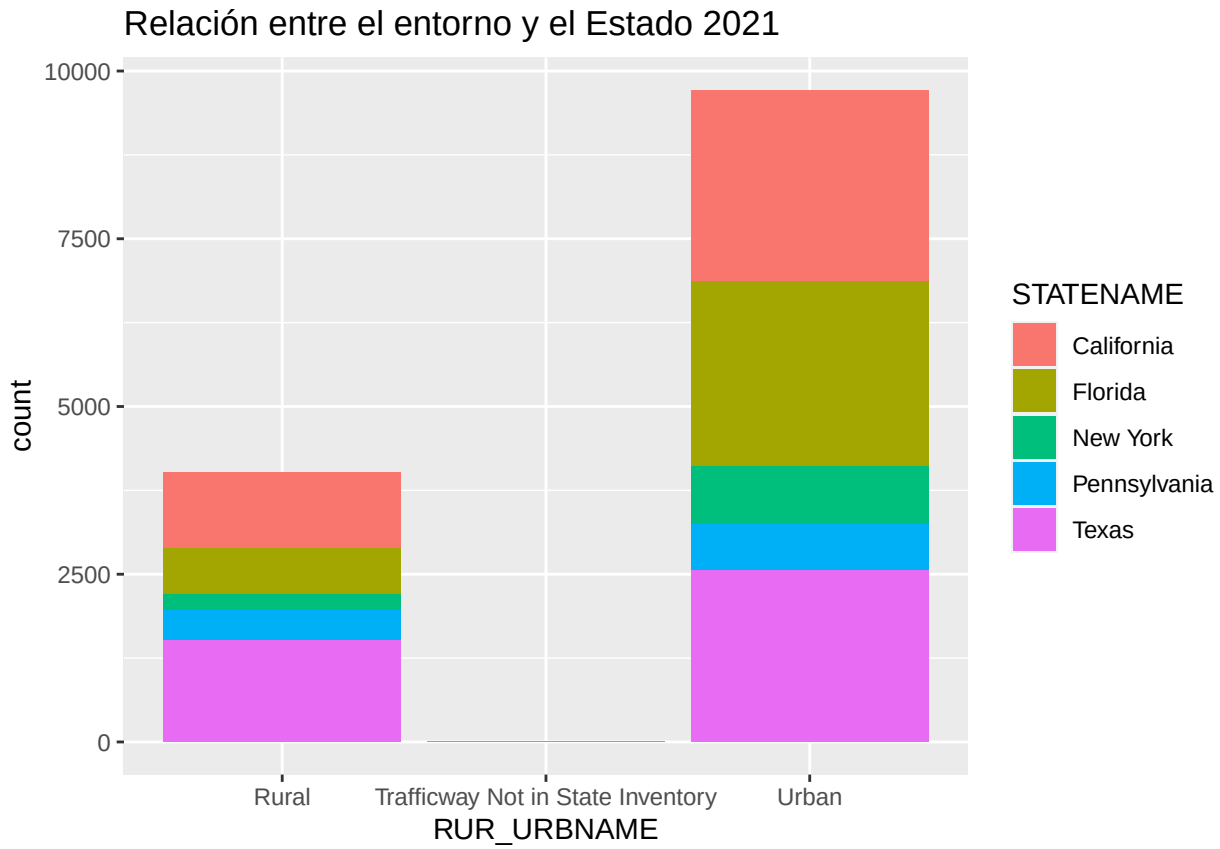
```r
# BASE DE DATOS DEL 2019
datos2019ESTADOS=subset(datos2019, datos2019$STATENAME == "California" | datos2019$STATENAME == "Texas"

files=dim(datos2019ESTADOS)[1]
ggplot(data=datos2019ESTADOS[1:files,],aes(x=RUR_URBNAME,fill=STATENAME))+geom_bar()+ggtitle("Relación
```




```r
# BASE DE DATOS DEL 2020
datosESTADOS=subset(datos, datos$STATENAME == "California" | datos$STATENAME == "Texas" | datos$STATENA

files2=dim(datosESTADOS)[1]
ggplot(data=datosESTADOS[1:files2,],aes(x=RUR_URBNAME,fill=STATENAME))+geom_bar()+ggtitle("Relación ent
```



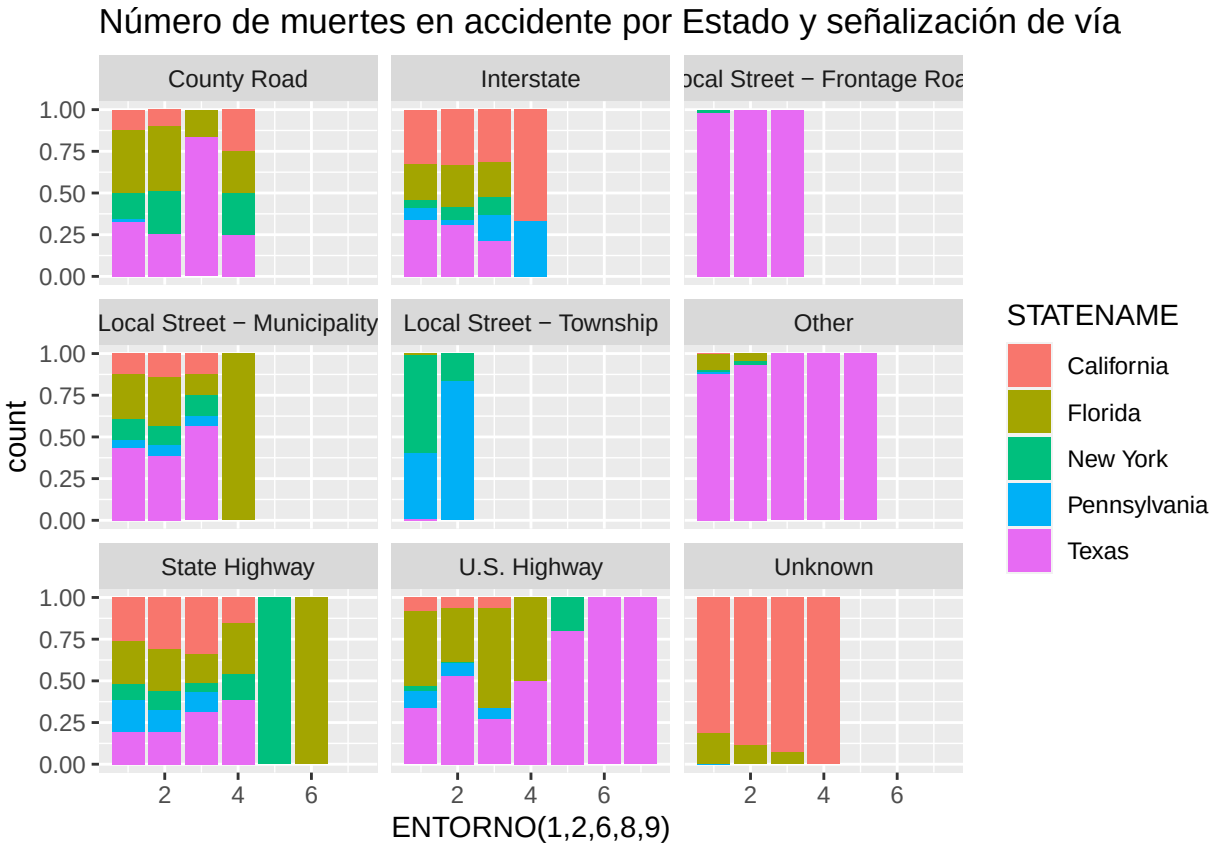

```
# BASE DE DATOS DEL 2021
datos2021ESTADOS=subset(datos2021, datos2021$STATENAME == "California" | datos2021$STATENAME == "Texas"

files3=dim(datos2021ESTADOS)[1]
ggplot(data=datos2021ESTADOS[1:files3,],aes(x=RUR_URBNAME,fill=STATENAME))+geom_bar()+ggtitle("Relación
```



Como se puede comprobar por los resultados de arriba, las bases de datos del 2019 y del 2020 parecen ser idénticas, por ello es pertinente mencionar que se ha comprobado varias veces que se estuviese leyendo el archivo correcto en cada caso. Dicho esto, se observa un aumento en el número de accidentes para el año 2021 en los dos entornos, aunque el comportamiento jerárquico se mantiene constante, siendo California y la Florida los dos estados con más accidentes acumulados.

Para intentar relacionar las variables aún más para sacar más conclusiones de los datos, se va a gráficar las variables de muertes y estado, para el tipo de señalización de la vía (atributo **ROUTE**) Volveremos a centrarnos en la base de datos del 2020 al no haber encontrado ningúna informacion relevante en las otras dos bases de datos.

```
ggplot(data = datosESTADOS[1:files2,],aes(x=FATALS,fill=STATENAME))+geom_bar(position="fill")+facet_wra
```



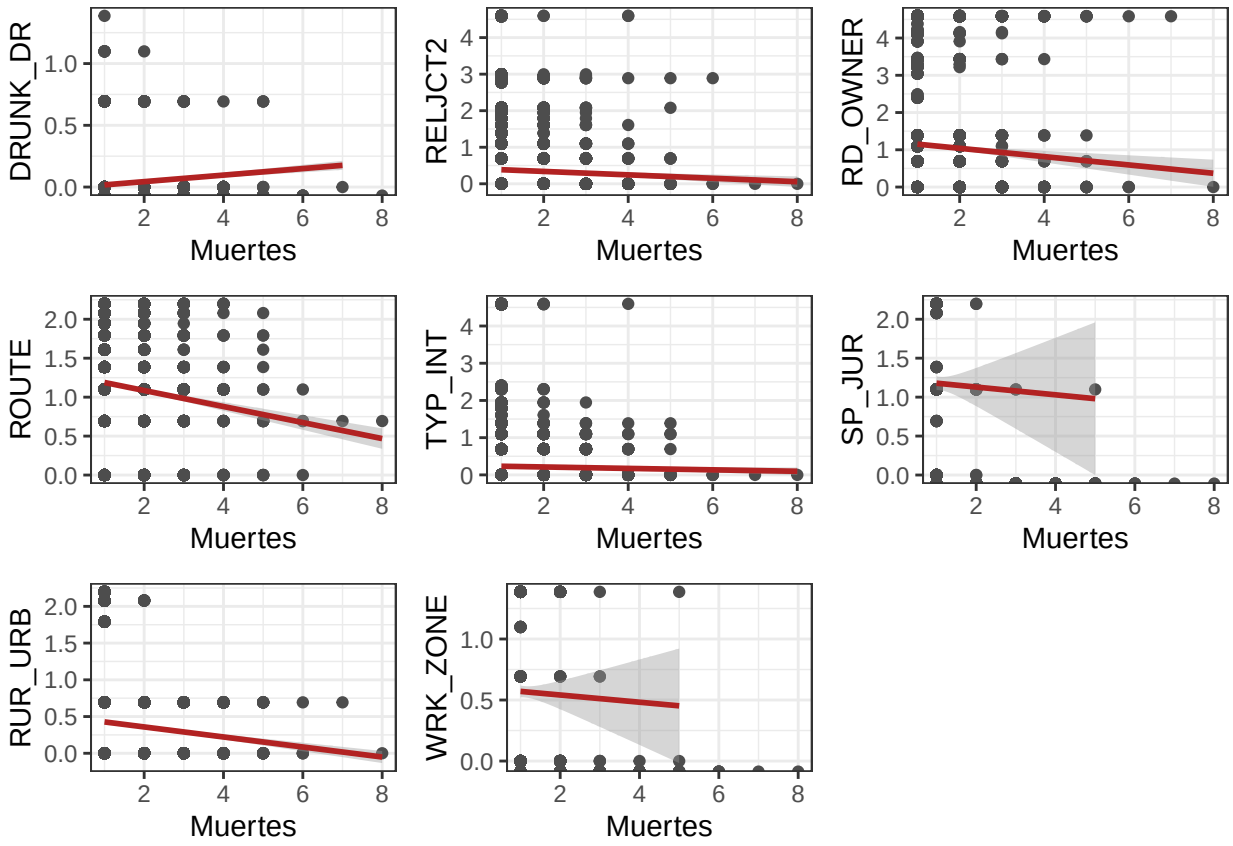
Como se puede observar, California es el estado que más porcentaje de muertes acumula en vías cuya señalización se desconoce. Además, se observa como para las vías "Local Street-Frontage Road" el porcentaje de (1,2 y 3) muertes se dispara en el estado de Texas, lo mismo pasa en carreteras de California cuya señalización se desconoce, en este caso, el porcentaje de 1,2,3 y 4 muertes es muy alto. Cabe mencionar que el estado de California, tiene un alto porcentaje de cualquier número de muertes en gran parte del conjunto de vías. Por último, poniendo el foco en las vías señalizadas como "U.S. Highway" el estado de Texas según aumenta el número de muertes, también lo hace el porcentaje de que el accidente sea posible, esto también se observa para el estado de Pennsylvania. Por lo contrario, en el caso del estado de la FLorida, según aumenta el número de muertes, el porcentaje de accidentes se reduce.

Si se estudian las vías señalizadas como "State Highway" se observa algo similar, excepto para el estado de la Florida, pues ahora según aumenta el porcentaje de accidente, el número de muertos aumenta.

A parte de las conclusiones anteriores no se ha podido generalizar ningun comportamiento, por ello, ahora se va a buscar la correlación en función de las muertes y un conjunto de variables de carácter geográfico.

```
# Utilizamos esta librería para usar la funcio multiplot()
if(!require('Rmisc')) install.packages('Rmisc'); library('Rmisc')

n = c("DRUNK_DR","ROUTE","RUR_URB","RELJCT2","TYP_INT","WRK_ZONE","RD_OWNER","SP_JUR")
accidentDataAux= datos %>% select(all_of(n))
histList2<- vector('list', ncol(accidentDataAux))
for(i in seq_along(accidentDataAux)){
  message(i)
histList2[[i]]<-local({
```

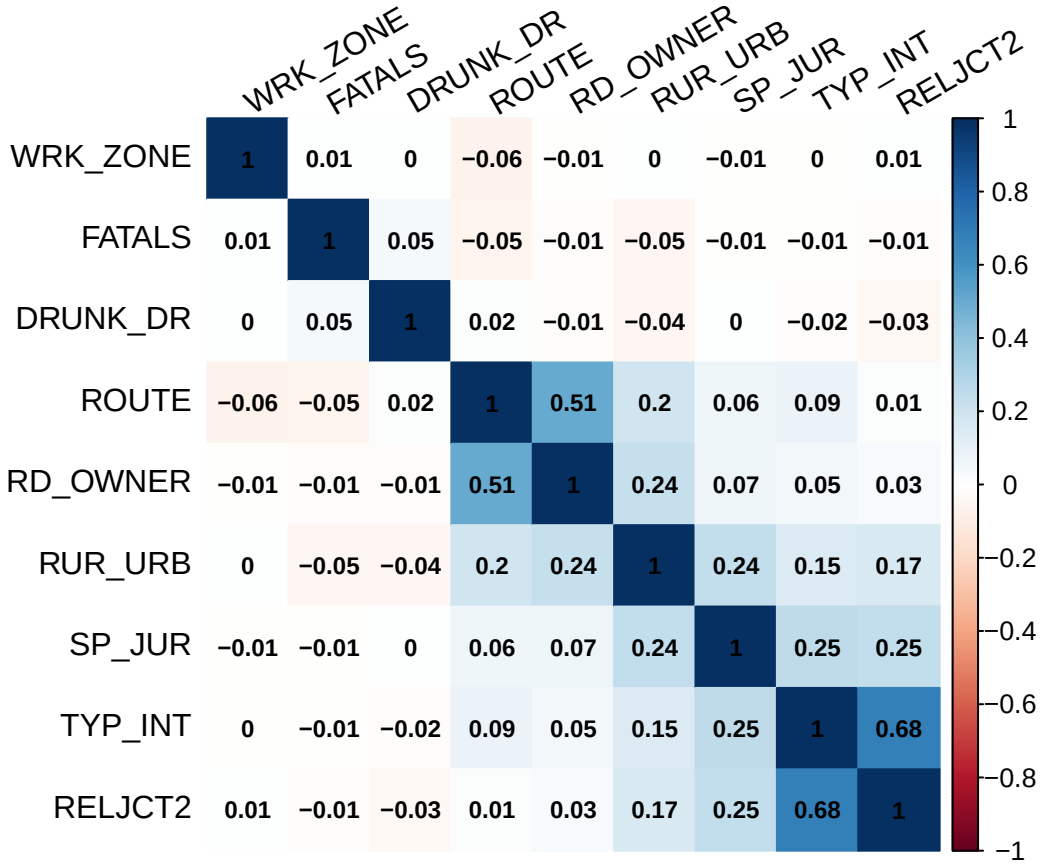
```
  i<-i
  col <-log(accidentDataAux[[i]])
  ggp<- ggplot(data = accidentDataAux, aes(x = accidentData$FATALS, y=col)) +
    geom_point(color = "gray30") + geom_smooth(method = lm,color = "firebrick") +
    theme_bw() + xlab("Muertes") + ylab(names(accidentDataAux)[i])
  })

}
multiplot(plotlist = histList2, cols = 3)
```



Como se puede observar arriba, la mayoría de tendencias son negativas, excepto la de los conductores bebidos, pues el número de muertos aumenta según lo hace el número de conductores alcohólicos. En el resto de gráficas se observa un comportamiento decreciente, que será más difícil de estudiar por la cantidad de valores que pueden tomar esos atributos, no obstante, se hace un pequeño estudio ahora. Y es que para muchos de estos atributos, el valor 0 (eje y) en cada atributo, suele determinar que aquello lo que el atributo represente, el valor 0 hace decrecer el número de víctimas, por lo tanto estamos provando lo contrario. Esto no es que sea malo, simplemente, lo que nos transmite esta gráfica es que en las zona donde hay obras o cerca de ellas, en los entornos rurales, en las zonas sin intersecciones, etc, el número de muertes decrece.

A continuación se construye la matriz corrplot, para ver en valor numérico las correlaciones:

```
if(!require("corrplot")) install.packages("corrplot"); library("corrplot")
n = c("FATALS","DRUNK_DR","ROUTE","RUR_URB","RELJCT2","TYP_INT","WRK_ZONE","RD_OWNER","SP_JUR")
factores= datos %>% select(all_of(n))
res<-cor(factores)
corrplot(res,method="color",tl.col="black", tl.srt=30, order = "AOE",
number.cex=0.75,sig.level = 0.01, addCoef.col = "black")
```



Como se puede observar hay una correlación considerable entre el atributo **TYP_INT** y **RELJCT2**, esto tiene sentido porque son dos atributos que representan el tipo de intersección y si cerca o en el lugar del accidente había una intersección. Luego, se observa una correlación del 0.51 entre **RD_OWNER** y **ROUTE**, esto era de esperar también, puesto que el atributo **RD_OWNER** representa la entidad legal de quien es propiedad la carretera, y el atributo **ROUTE** identifica el tipo de señalización de la carretera del accidente.

Aunque la correlación de **TYP_INT** y **RELJCT2** no es el del 90%, se es consciente de que no aportan demasiada información por seaparado, por lo tanto se va a eliminar uno de ellos. El criterio de eliminación de un atributo es el mismo que se ha visto en teoría, y es que, hay que eliminar el atributo que menos relación guarde con el objetivo principal del proyecto. Como el objetivo del proyecto era el de la identificación de las zonas de carreteras, estados, entornos, tipos de vías, etc dónde más accidentes se producen, es decir la identificación de atribiutos geográficos que más accidentes acumulan, para en un futuro poder prevenir accidentes con ayuda de un modelo de IA/ML. Como el atributo **TYP_INT** solo se centra en el tipo de intersección, mientras que el atributo **RELJCT2** identifica la zona del accidente respecto a la presencia o proximidad de elementos típicos de cruces, salidas, avenidas, etc, se va a proceder a eliminar el atributo **TYP_INT**.

‘{r} drop <- c("TYP_INT","TYP_INTNAME") datos = datos[,!(names(datos) %in% drop)] sort(colnames(datos)) #COMPROBAMOS QUE LO HEMOS ELIMINADO > >Anteriormente, se vió como los entornos de los accidentes predominantemente, eran dos; RURAL y URBANO, no obstante, el atributo entorno, podría tomar hasta 5 valores. Por lo tanto lo que se va a hacer ahora, va a ser CODIFICAR con un cada vez que un accidente se de en un entorno urbano y en 0 cuando se de en un entorno rural.

INTERPRETACIÓN DE LOS RESULTADOS

Por los útlimos datos que se han estudiado, se podria decir que el estado de California, es el estado que más muertes acumula en cualquier tipo de vía. Estudiando los datos del estado de Texas, a pesar de población, es el estado que menos accidentes suele registrar (dentro del conjunto de estados que se ha estudiado)

Finalmente, a modo de reflexión, creo que a lo largo del desarollo de este proyecto, he sufrido de lo llamado como parálisis por análisis, pues he querido buscar un buen objetivo, y a la hora de analizar los resultados, al haber tantos datos y atributos, he colapsado, olvidando así, ideas interesantes para relacionar atributos y perdiendo de vista el objetivo.