Minería de datos: PEC1

Autor: Vinicio Alejandro Naranjo Mosquera

Marzo 2024

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 Agenc
##  $ 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 Hour
##  $ 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" "Unknow
##  $ 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 **35766** 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
```

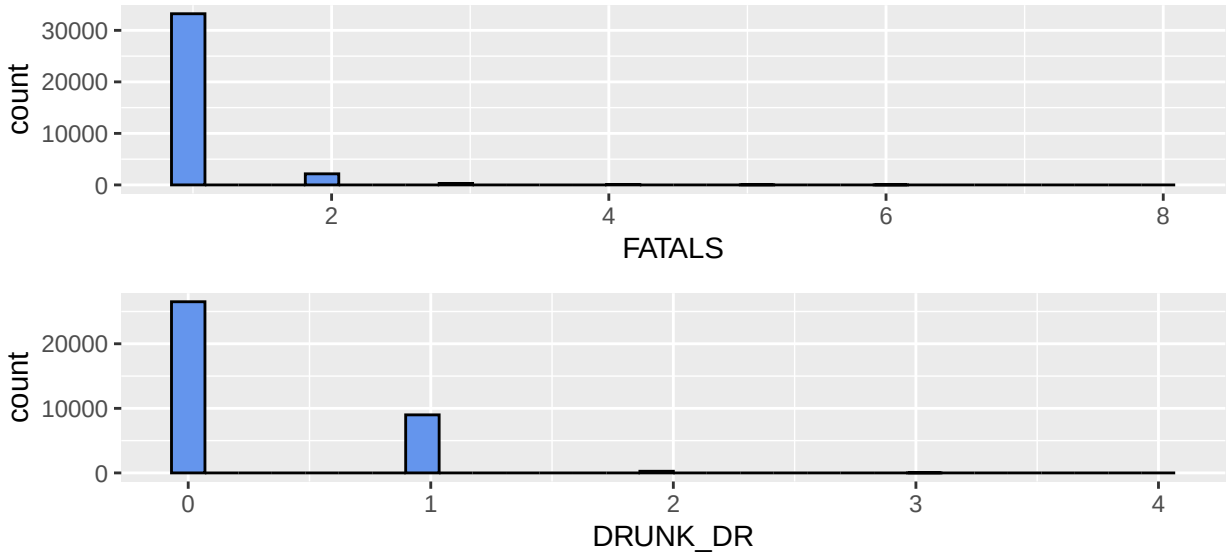
Vemos que no hay valores nulos en los datos. También verificamos si existen campos llenos de espacios en blanco. En este caso sí encontramos el campo TWAY_ID2 con 26997 valores en blanco. Valoramos no hacer ninguna acción de eliminar registros puesto que este campo no lo utilizaremos.

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:

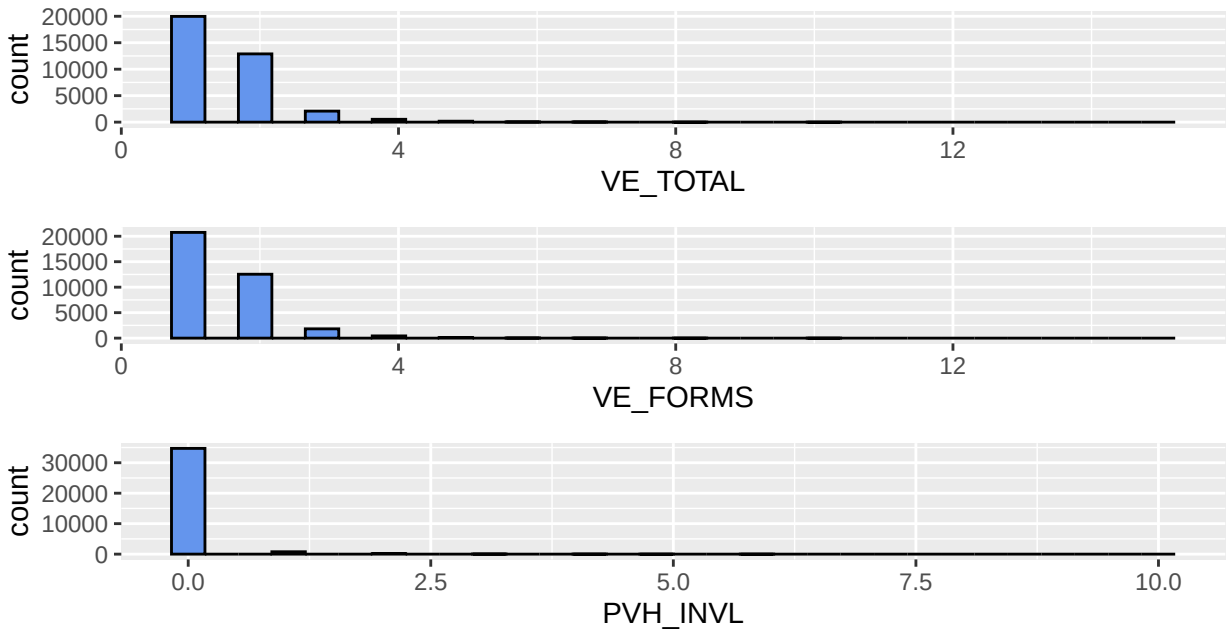
Número de muertes: Todos los accidentes recogidos en estos datos reportan una muerte como mínimo. Siendo el accidente más grave con ocho víctimas y vemos que la distribución se acumula de forma muy evidente en una muerte por accidente.

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.

```r
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
```

```r
#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
```

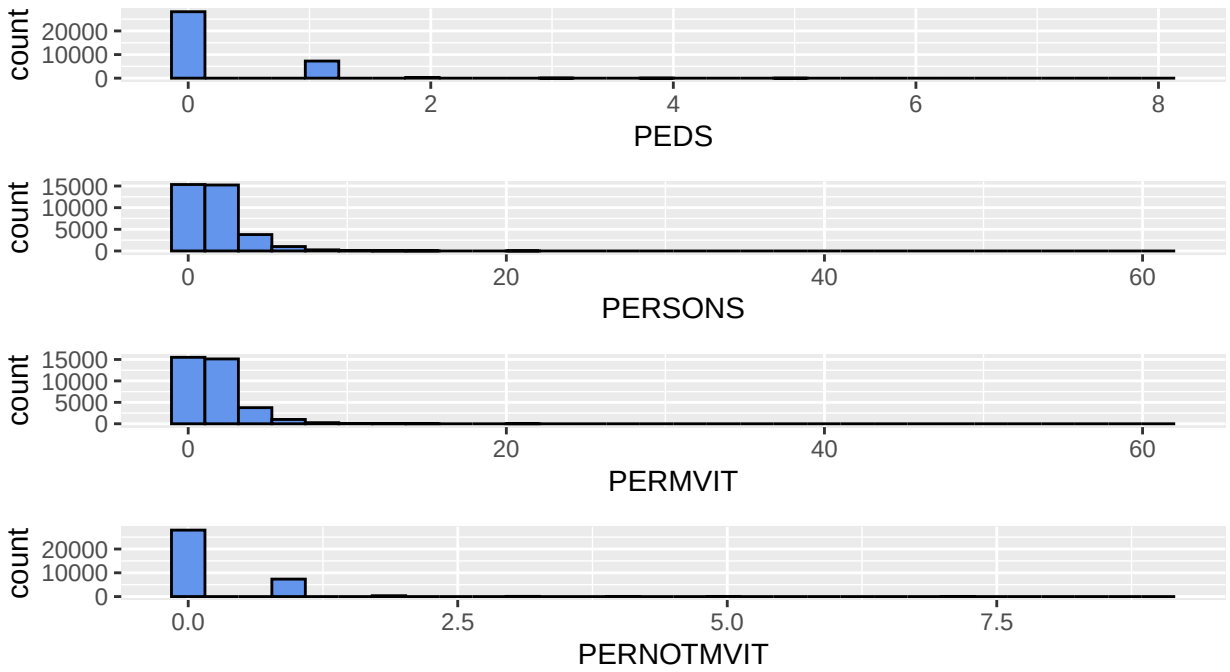
Observaciones en cuanto a los vehículos implicados.

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.

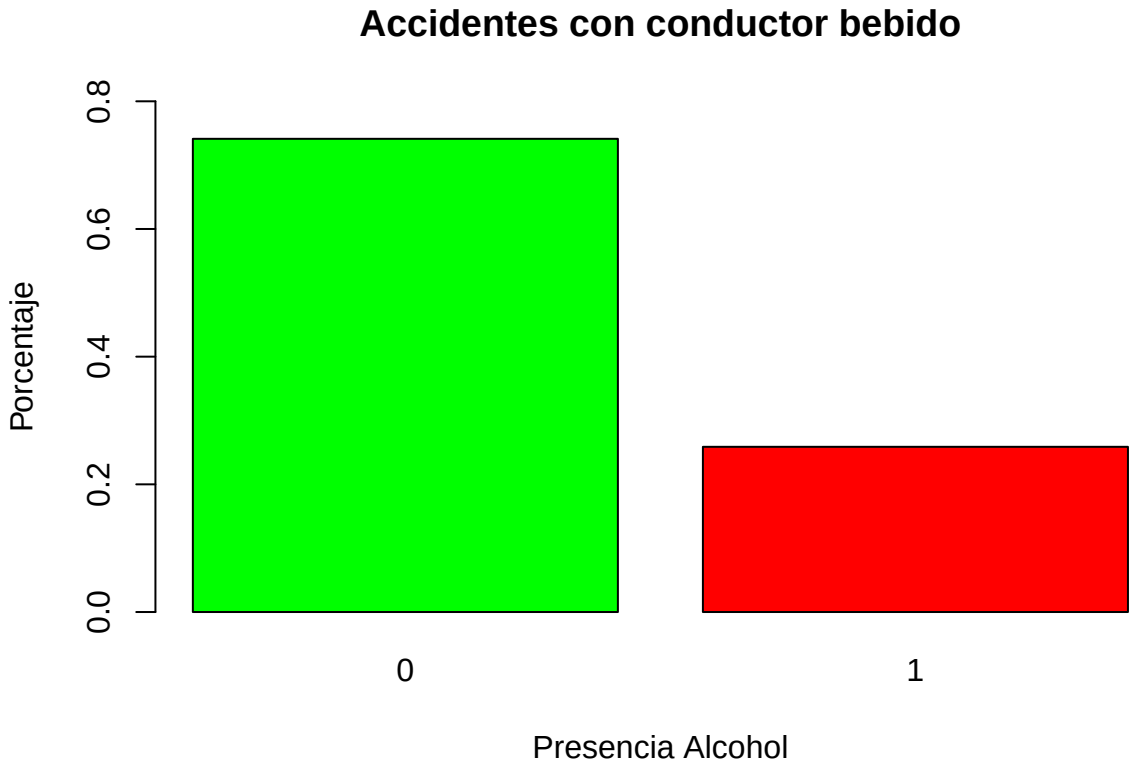
```
accidentData$alcohol <- ifelse(accidentData$DRUNK_DR %in% c(0), 0, 1)
counts <- table(accidentData$alcohol)
barplot(prop.table(counts),col=c("green","red"), main="Accidentes con conductor bebido", legend.texto=c
```

```
## 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%

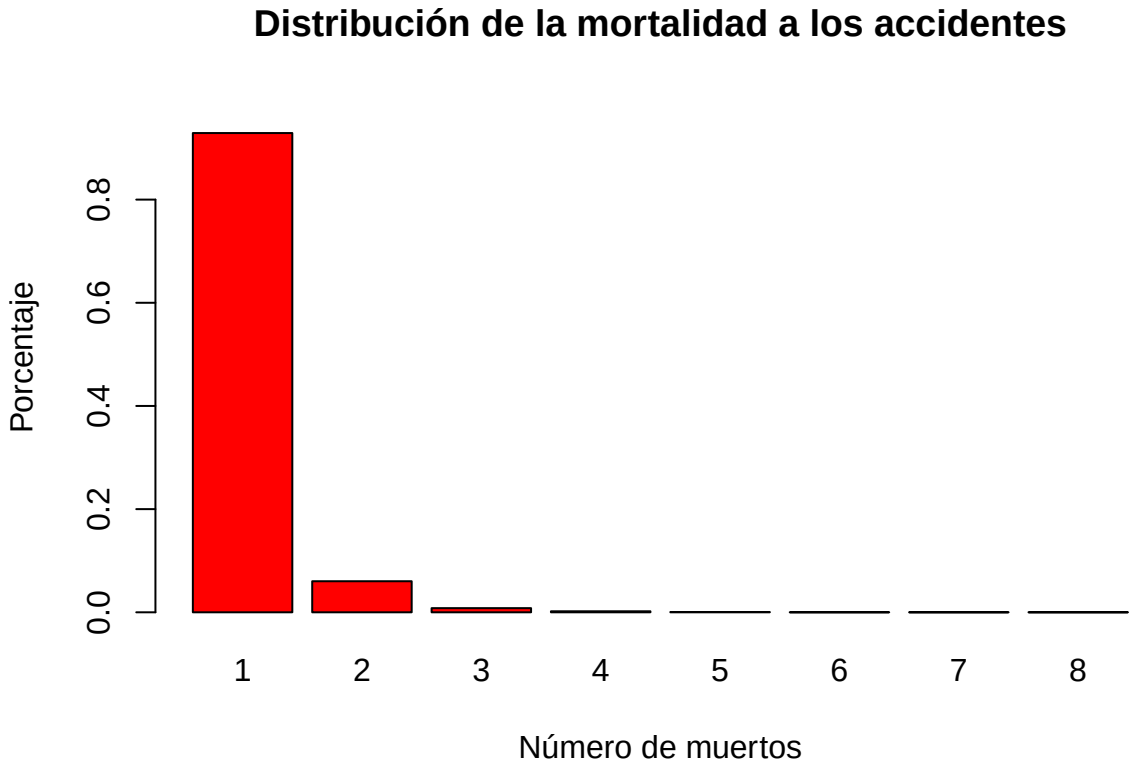
Observamos ahora como se distribuyen las muertes por accidente.

```
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
```

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



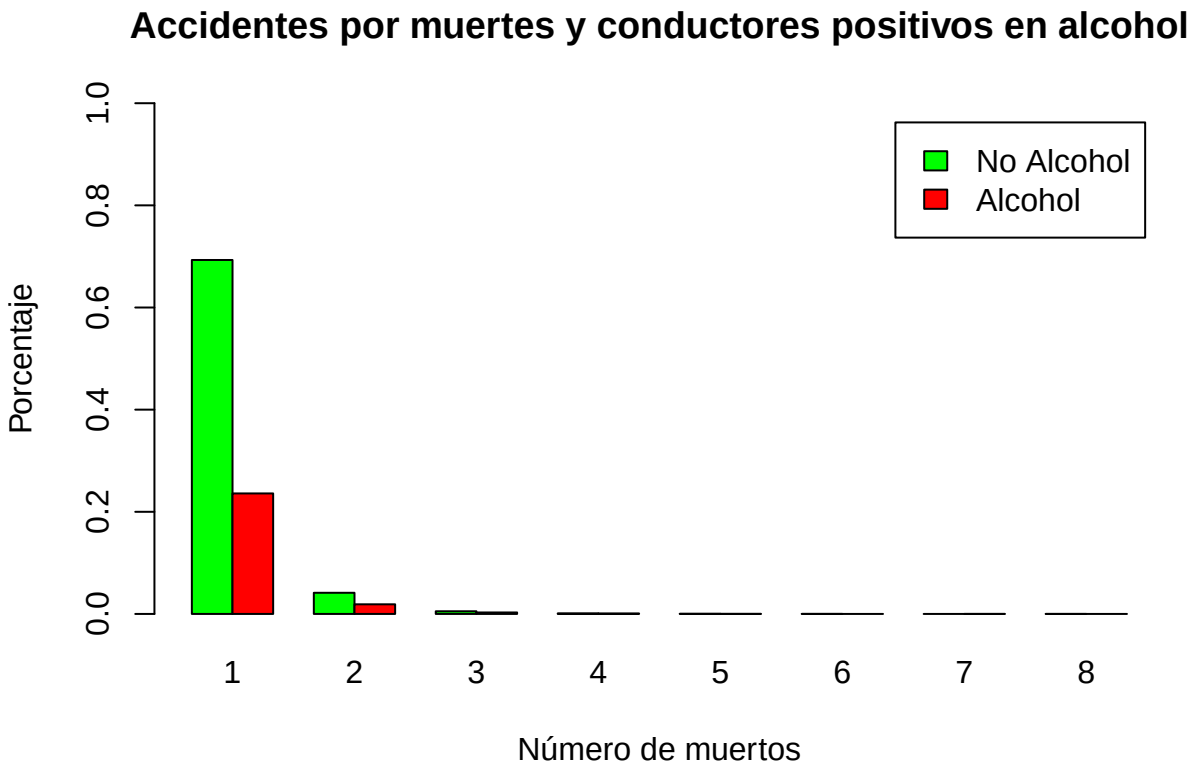
Vemos que la mayoría de los accidentes tienen como mínimo un muerto. Vamos ahora a relacionar mortalidad y alcohol.

```
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
```

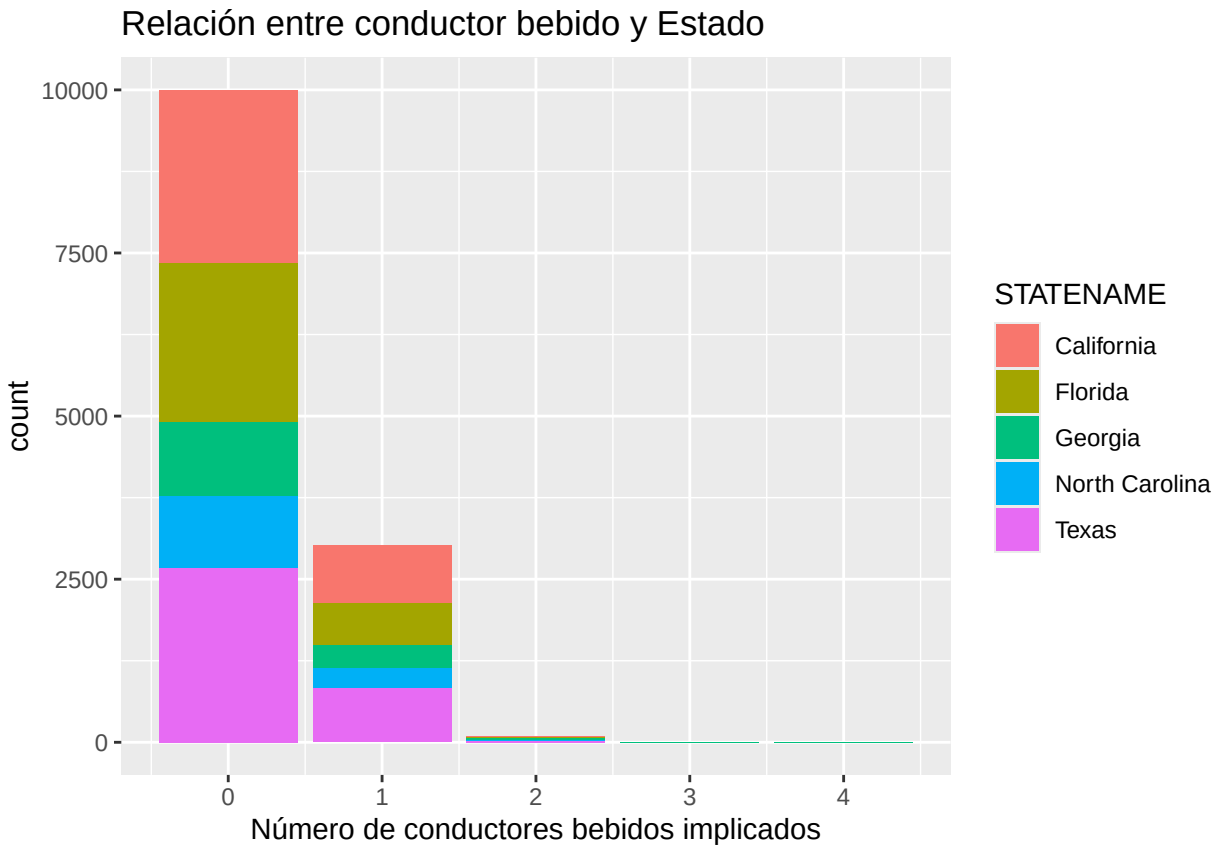
```
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : "fill" is not a graphical parameter
```

```
## 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.

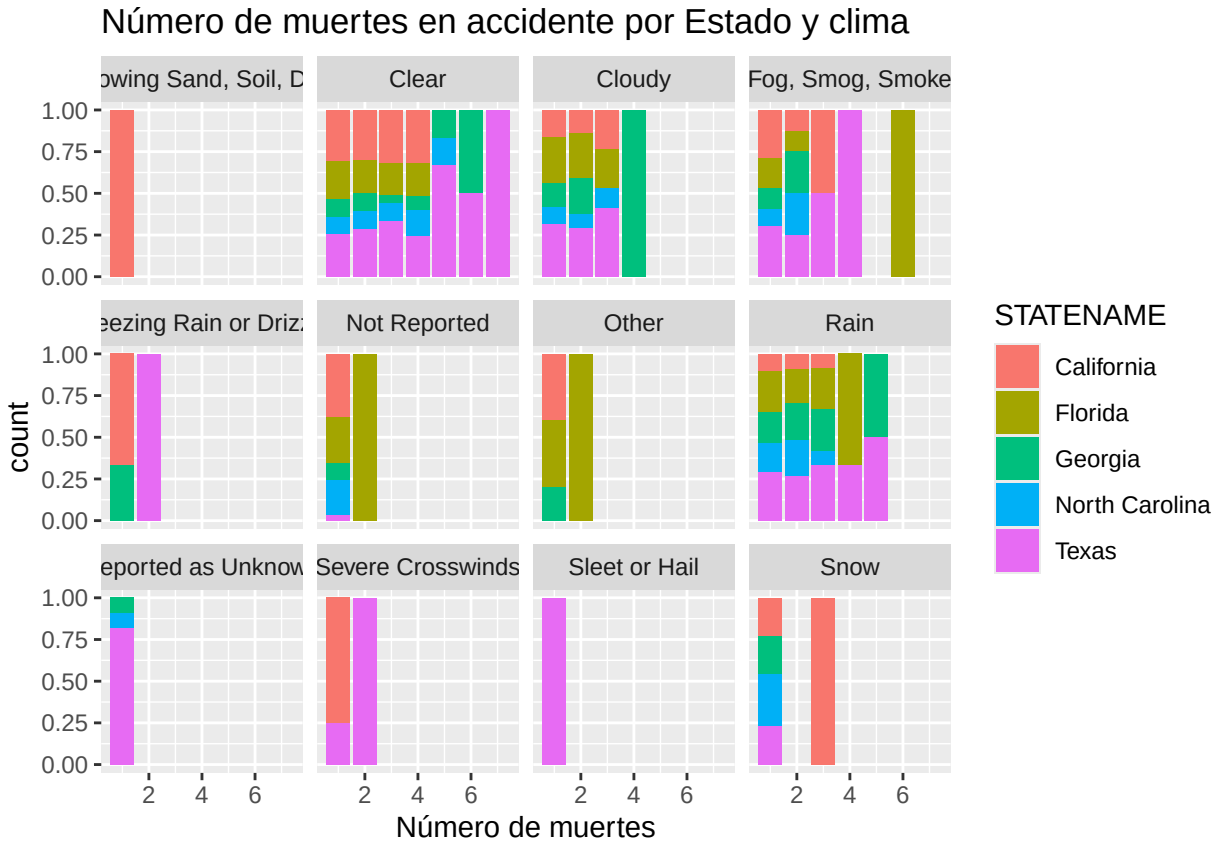
```
accidentDataST5=subset(accidentData, accidentData$STATENAME == "California" | accidentData$STATENAME ==

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
```



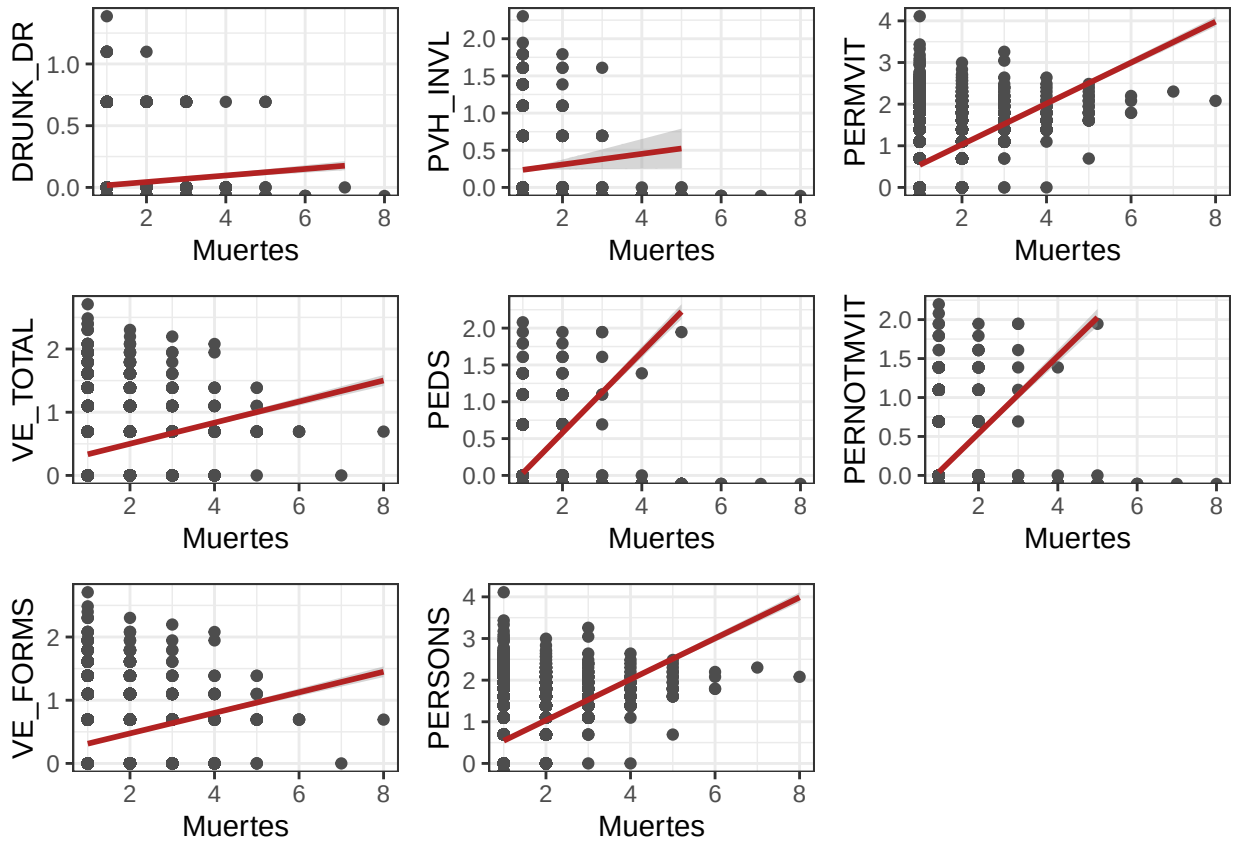
Esta gráfica está bien como mecánica de construcción, pero los resultados los ponemos en entredicho. Está bien como paso inicial, pero hay que profundizar mucho más.

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)
```

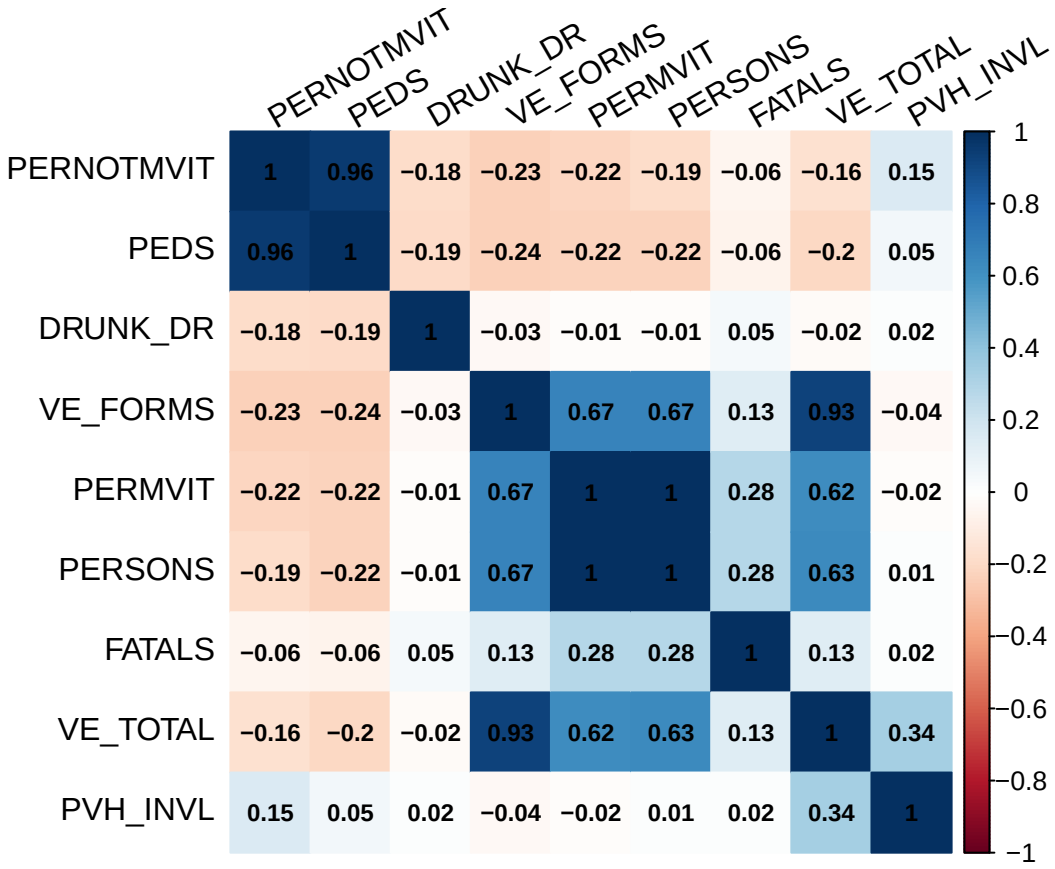


Podemos ver que:

- 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   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## v readr     2.1.5
## -- 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()
```

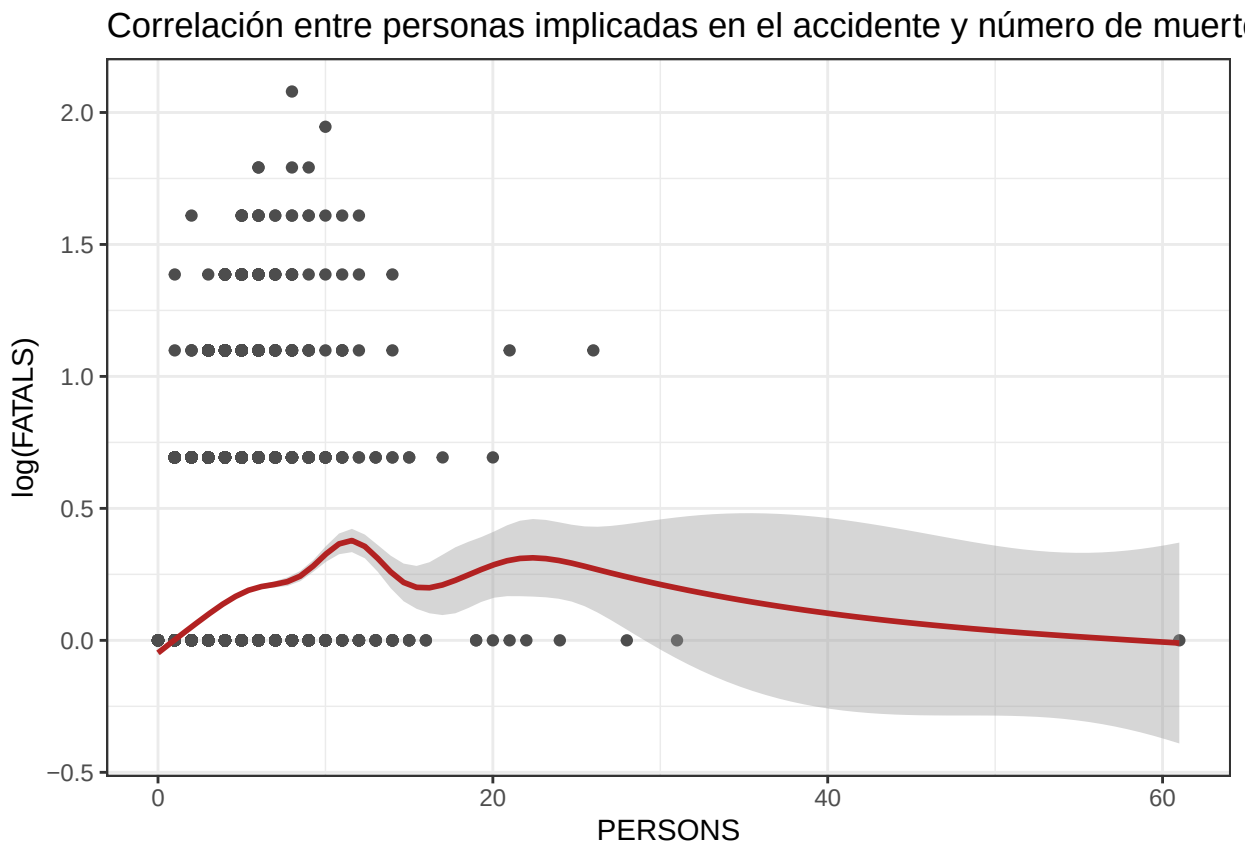
```
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
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" ...
```

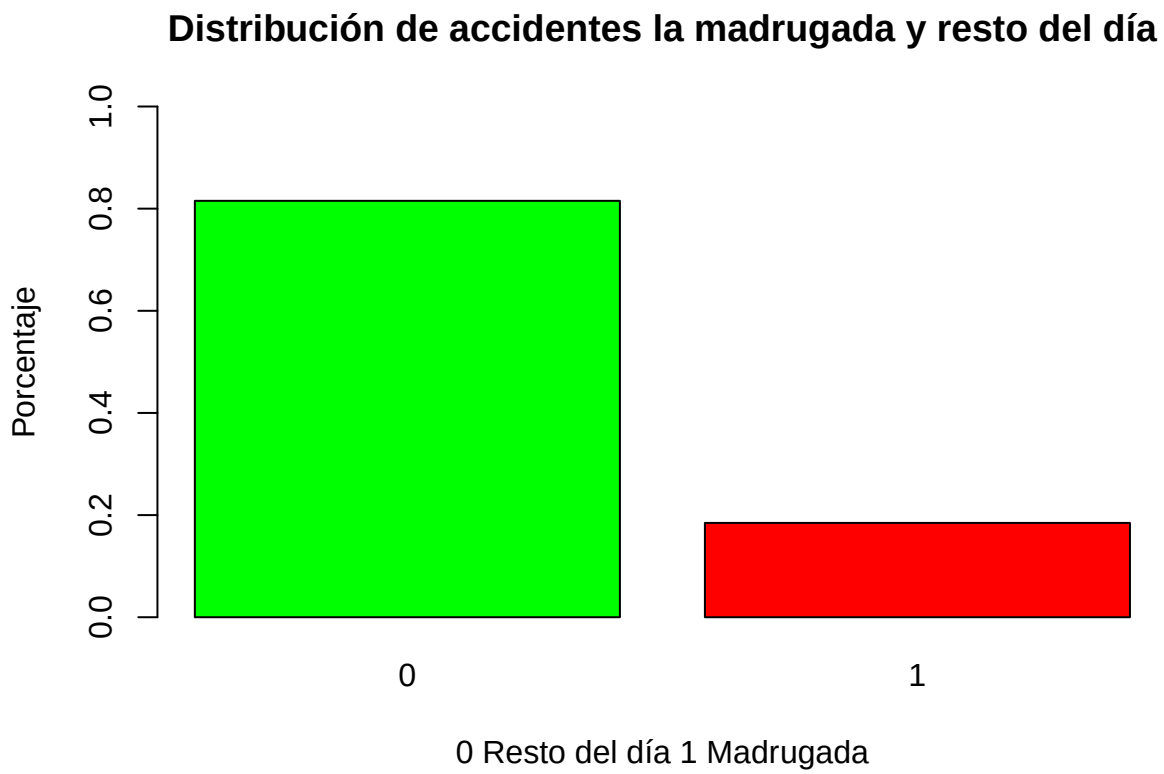
```
##  $ LATITUDE    : num  32.4 33.5 33.3 33.8 32.8 ...
##  $ LATITUDENAME: chr  "32.43313333" "33.48465833" "33.29994167" "33.79507222" ...
##  $ LONGITUD    : num  -86.1 -86.8 -86.4 -85.9 -86.1 ...
##  $ LONGITUDNAME: chr  "-86.09485" "-86.83954444" "-86.36964167" "-85.88348611" ...
##  $ SP_JUR      : 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.

```r
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.

```r
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.

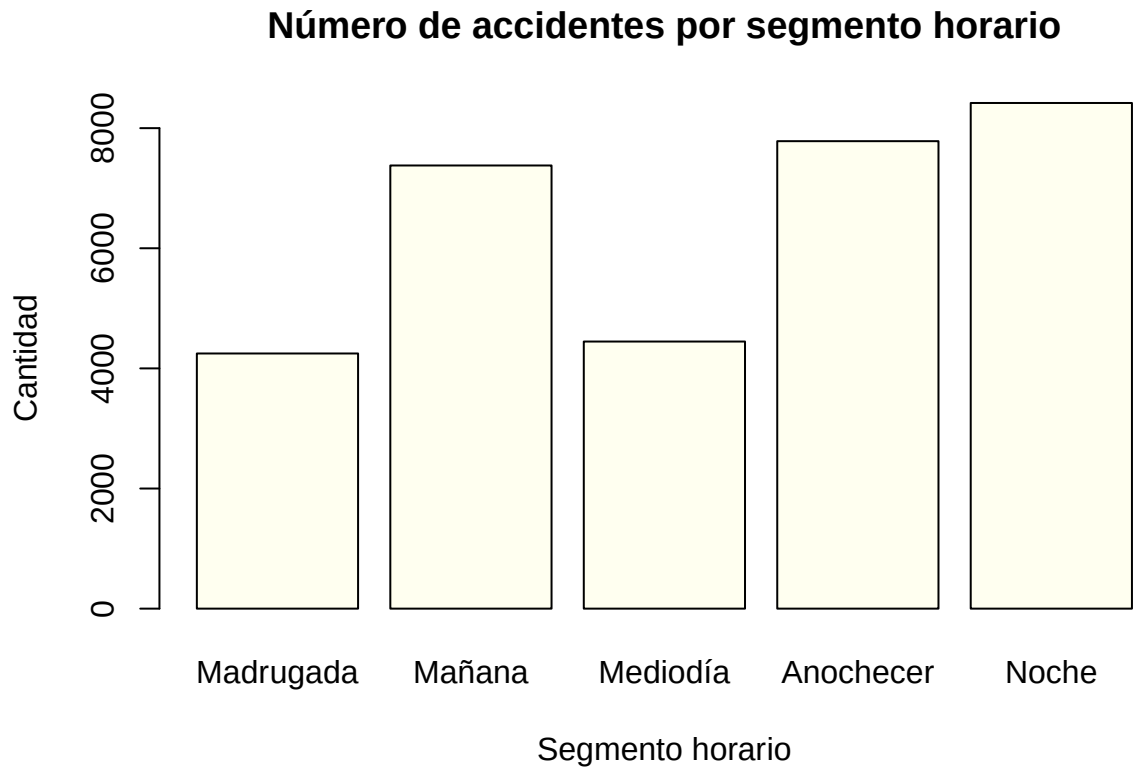
```r
accidentDataAux["segmento_horario"] <- cut(accidentDataAux$HOUR, breaks = c(0,4,11,14,18,22), labels = c
```

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

```r
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'
```

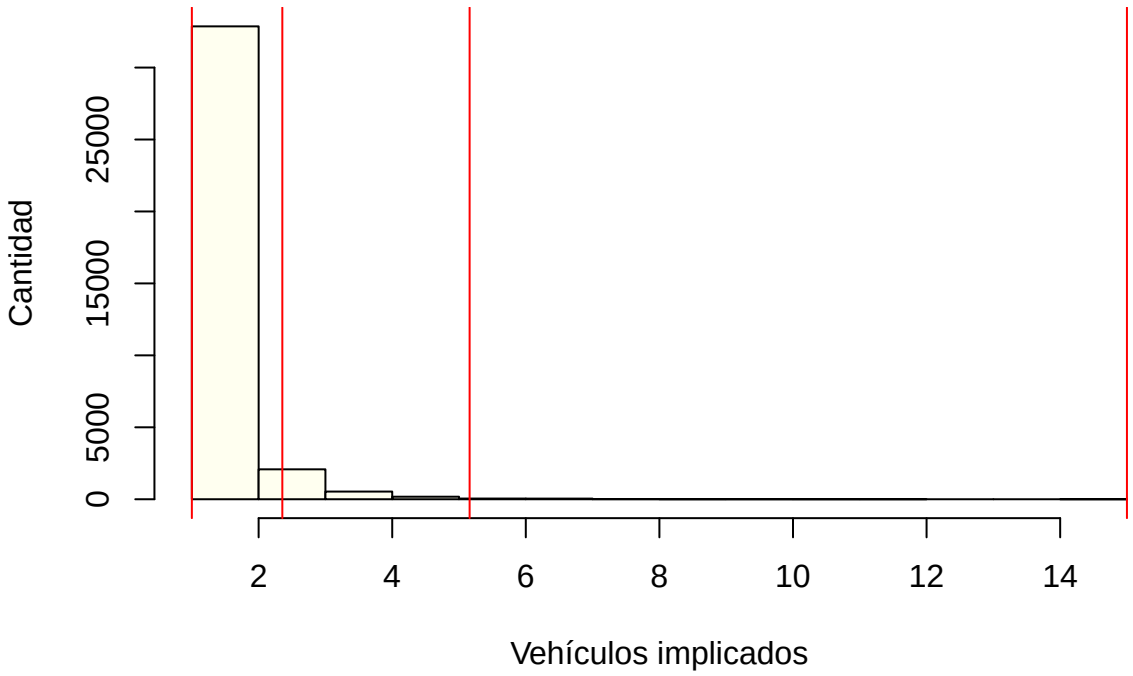
```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## 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
```

```r
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.

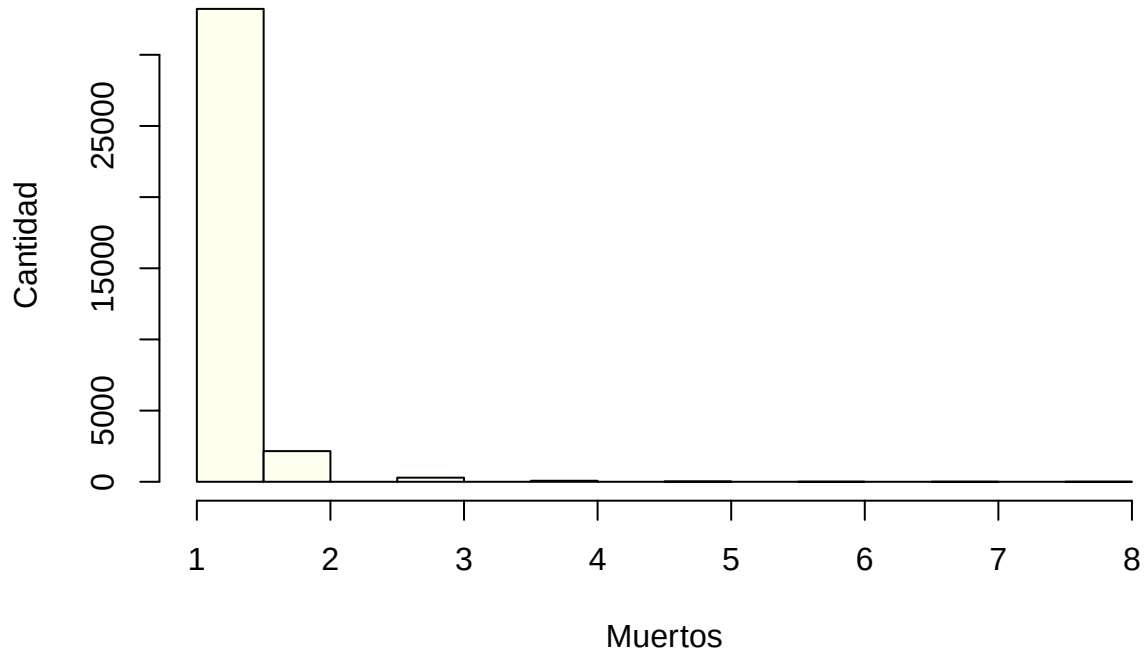
```
accidentData$FATALS_ND = (accidentData$FATALS-min(accidentData$FATALS))/(max(accidentData$FATALS)-min(a

max(accidentData$FATALS)
```

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

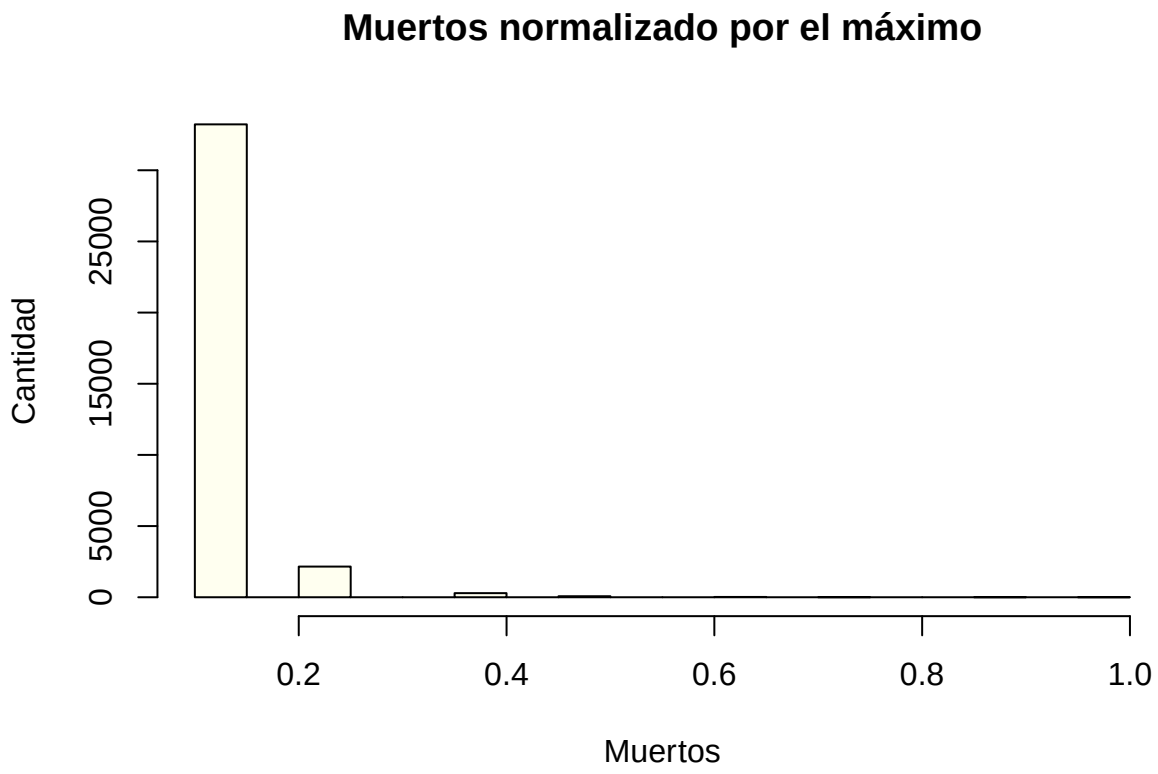
```
min(accidentData$FATALS)
```

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

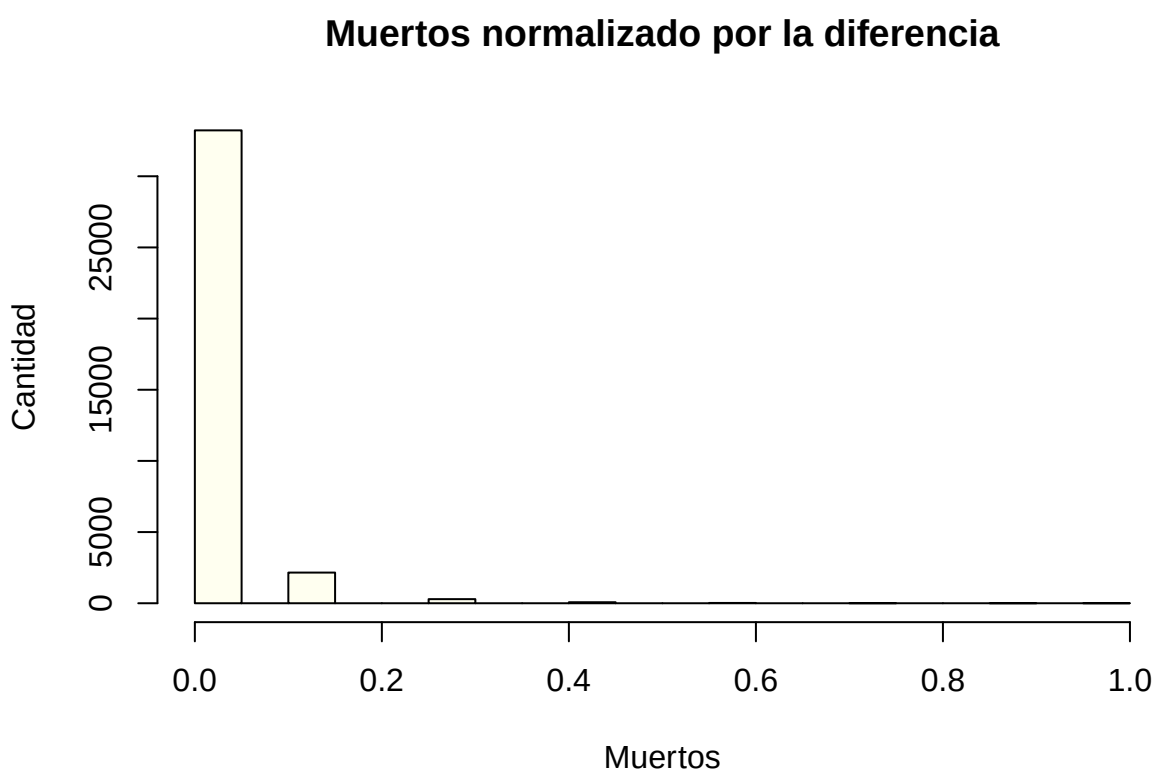
```
hist(accidentData$FATALS,xlab="Muertos", col="ivory",ylab="Cantidad", main="Número de muertes en accide
```




```
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.

```
# 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()**.

```
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     1.888e-16 3.618e-17
## 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:

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

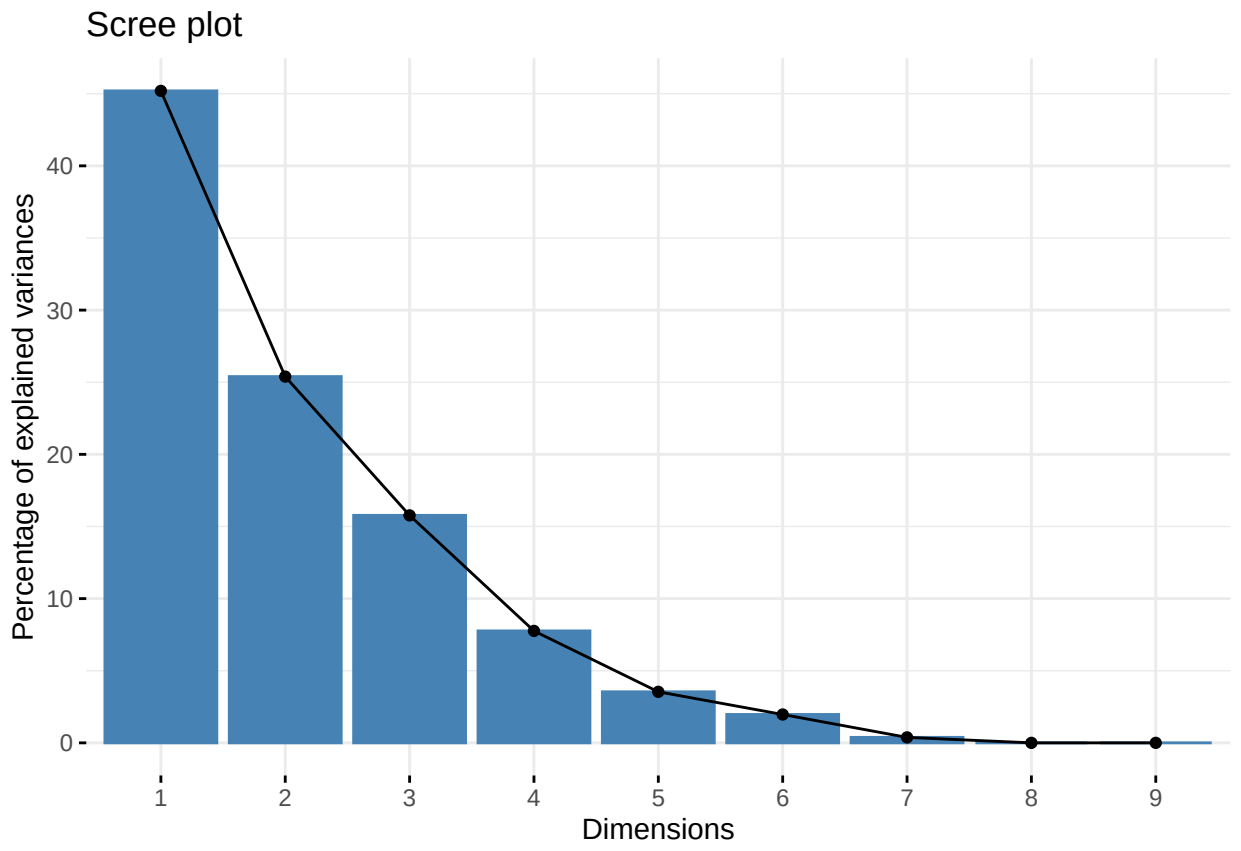
```
## Loading required package: factoextra
```

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

```
#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 3.563508e-32     1.180414e-28                   100.00000
## Dim.9 1.308692e-33     4.335050e-30                   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 3.563508e-32 1.308692e-33
```

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.

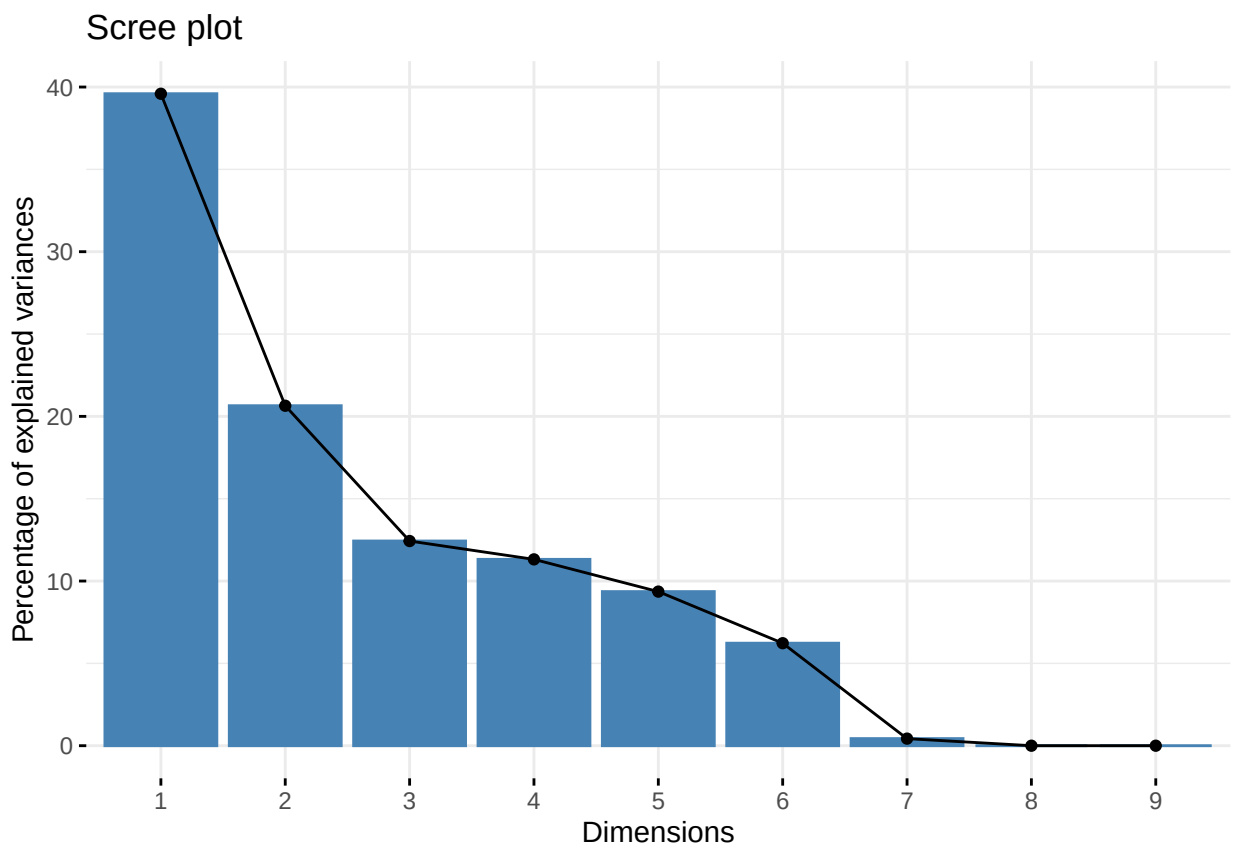
```r
# Escalamos los datos
acc_scale <- scale(accidentData_nor)
# Calculamos las componentes principales
pca.acc_scale <- prcomp(acc_scale)
# Mostramos la varianza de dichas variables:
var_acc_scale <- pca.acc_scale$sdev^2
head(var_acc_scale)
```

```
## [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 2.742090e-29     3.046767e-28                     100.00000
## Dim.9 1.034078e-30     1.148976e-29                     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)
```

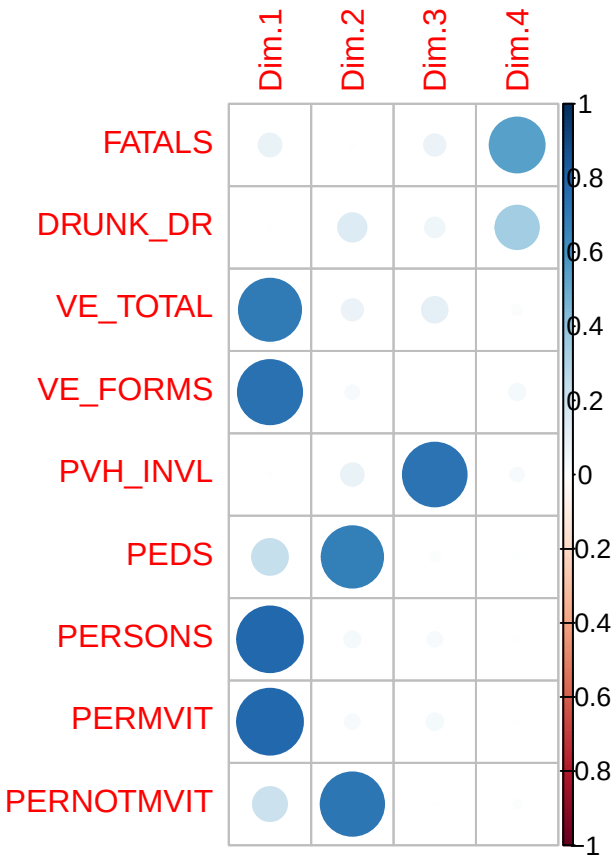
```
##                  Dim.1       Dim.2       Dim.3       Dim.4
## FATALS     0.094406509 0.005786562 0.083044911 0.547124037
## DRUNK_DR   0.002422180 0.145700988 0.069985146 0.342195618
## VE_TOTAL   0.700714591 0.082403990 0.118270488 0.018845040
## VE_FORMS   0.747937080 0.033918370 0.000575056 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
```

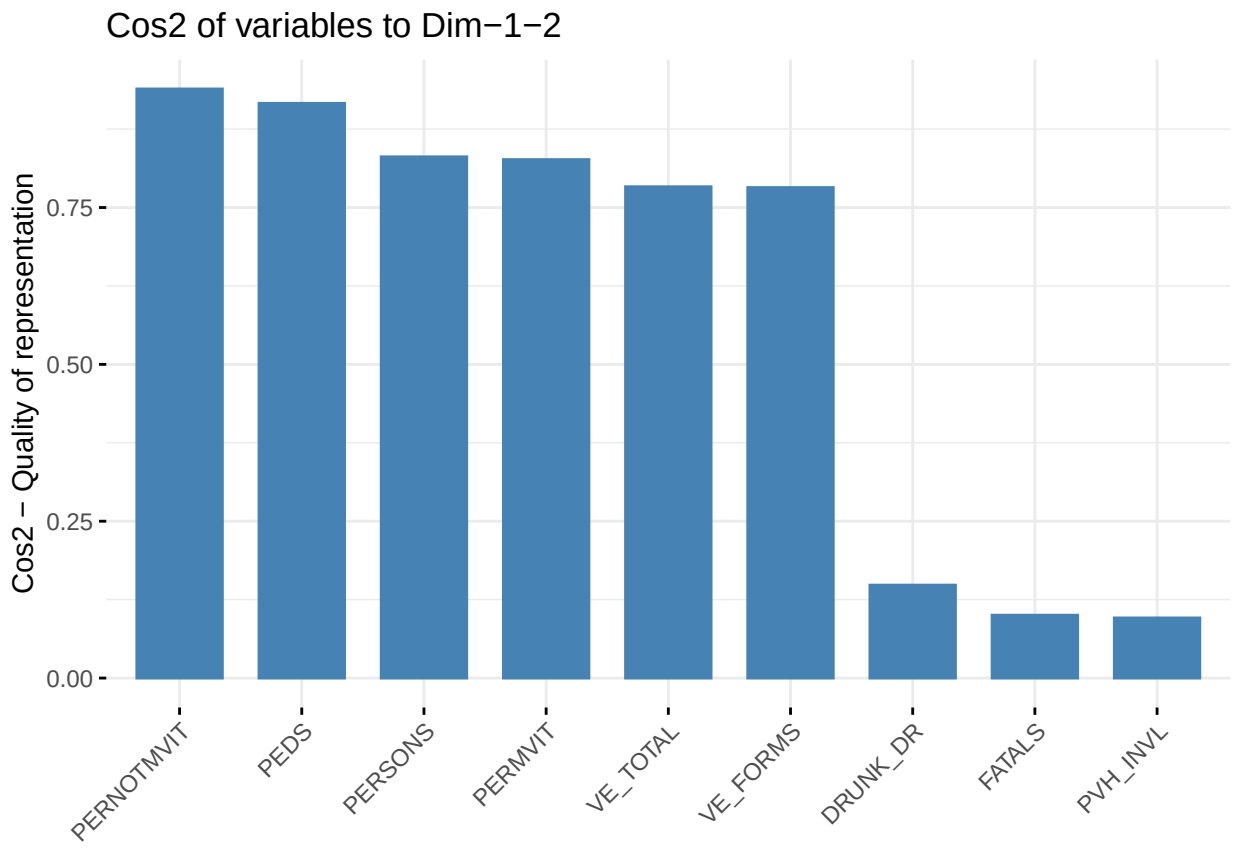
```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col =
## tl.col, : "is.corre" is not a graphical parameter
```

```
## 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.

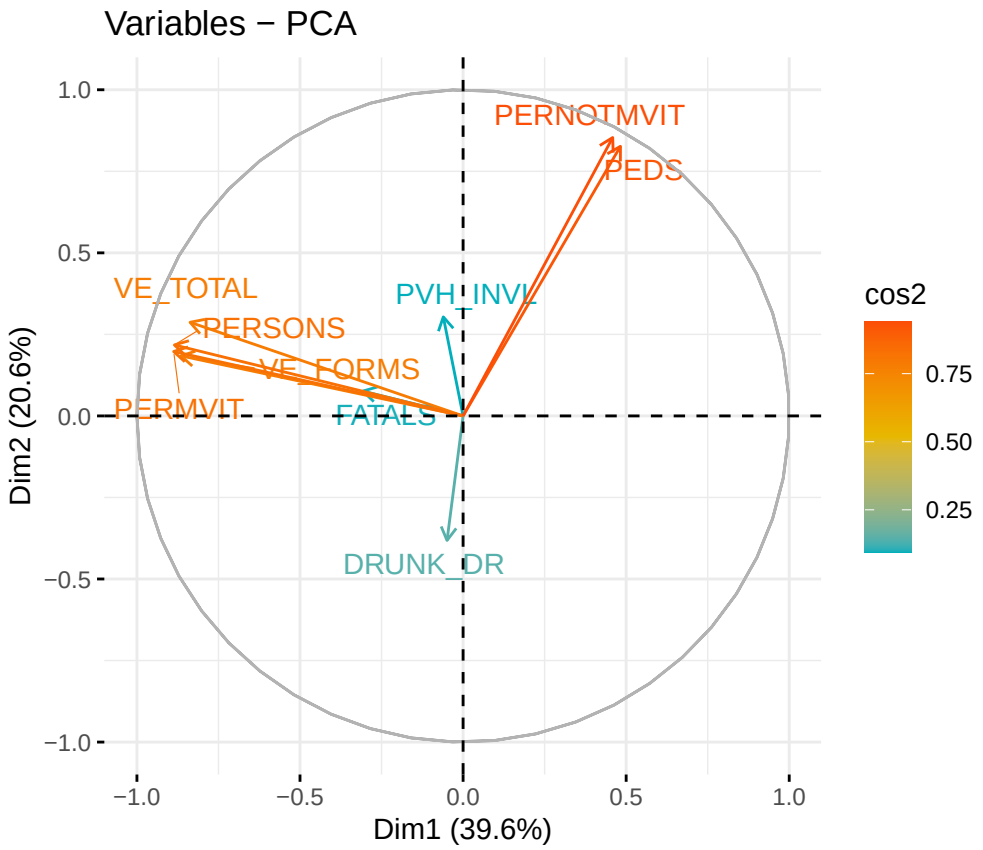
Si una variable está perfectamente representada por solo dos componentes principales (Dim.1 y Dim.2), la suma del cos2 en estos dos PCs es igual a uno. En este caso las variables se colocarán en el círculo de correlaciones.

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:

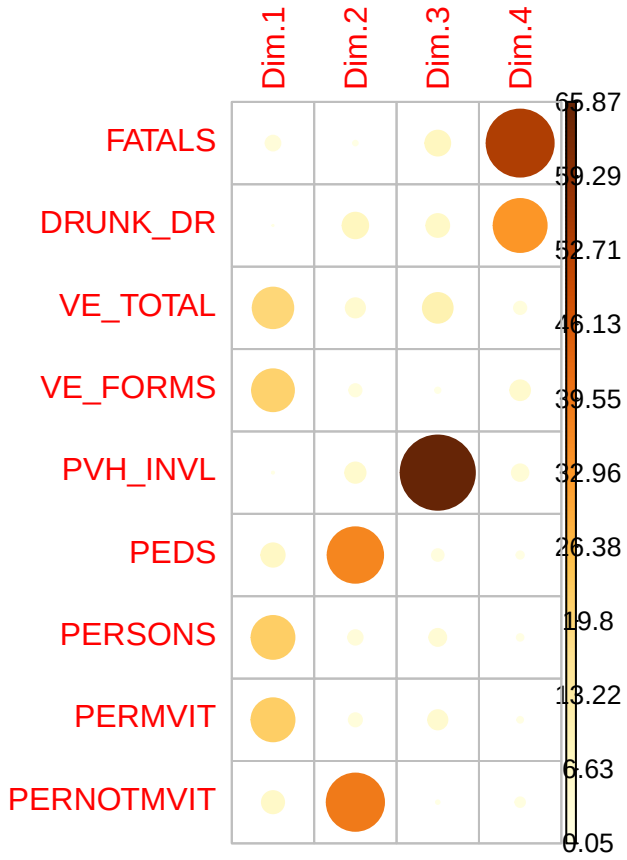
```
head(var$contrib[,1:4],11)
```

```
##                 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
```

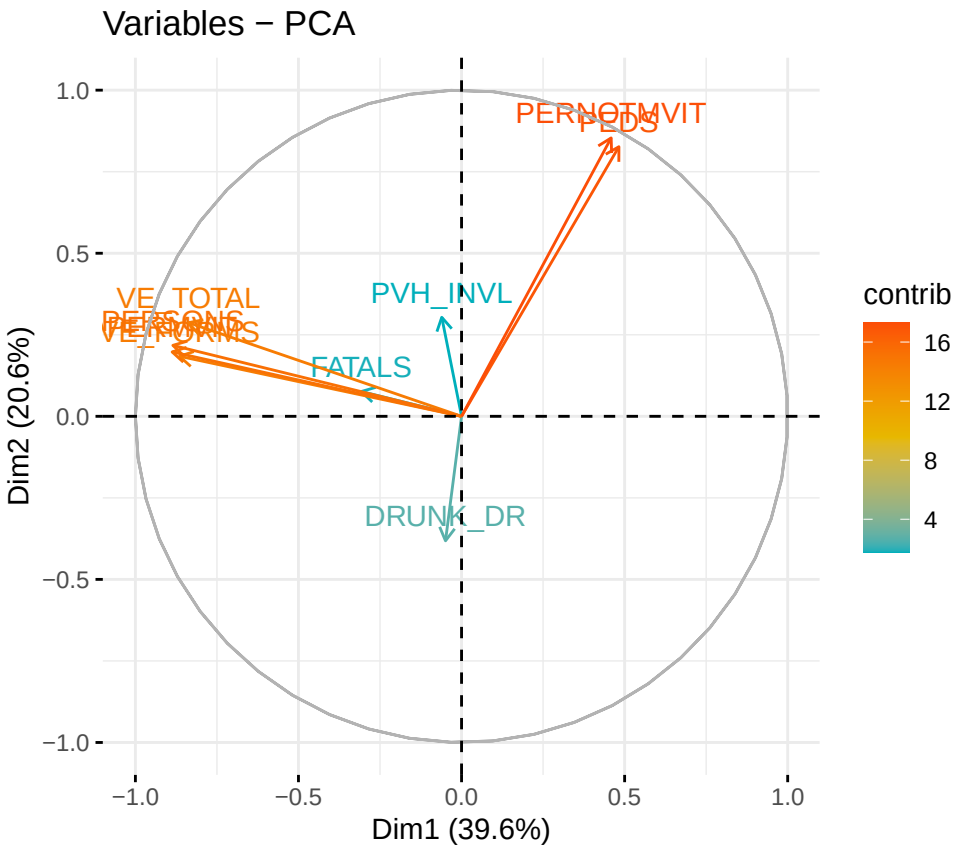
Cuando más grande sea el valor de la contribución, más contribución habrá al componente.

```
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 35766 accidentes en los que perdieron la vida 38824 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**.

Se espera por lo tanto que se responda de forma argumentada a las siguientes preguntas (Metodología *CRISP-DM*):

1. **Comprensión del negocio** - ¿Qué necesita el negocio?
2. **Comprensión de los datos** - ¿Qué datos tenemos/necesitamos? ¿Están limpios?
3. **Preparación de los datos** - ¿Cómo organizamos los datos para el modelado?
4. **Modelado** - ¿Qué técnicas de modelado debemos aplicar?
5. **Evaluación** - ¿Qué modelo cumple mejor con los objetivos del negocio?
6. **Implementación** - ¿Cómo acceden los interesados a los resultados?

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.

Extensión mínima: 400 palabras

Escribe aquí la respuesta a la pregunta

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. Así, no se puede utilizar la combinación de variables utilizada en el ejemplo: "FATALS","DRUNK_DR","VE_TOTAL","VE_FORMS","PVH_INVL","PEDS","PERSONS
Se debe analizar cualquier otra combinación que puede incluir (o no) algunas de estas variables con otras nuevas.

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)

```
# Cargamos el juego de datos

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

Escribe aquí la respuesta a la pregunta