Capstone Project - France Traffic Accident Analysis

Trevor Cummins

30th of May 2019

Abstract

HarvardX PH125.9x Capstone Project: Choose your own project - France Traffic Accident Analysis and Severity Prediction. An exploratory data analysis is performed on the 2017 French Traffic Accident datasets published by the French Government. Generalised Linear Model (GLM) and RPart classification machine learning models are developed based on training sets to predict the severity of an accident and validated against a test set of actual accident outcomes. The objective is to develop a model to simluate a call received by the emergency services and enable first response services to be managed based on the expected severity of the accident.

Contents

1 Executive Summary

1.1 Introduction

This Capstone project is the second of two project submissions required to complete the HarvardX PH125.9x Professional Certificate in Data Science. The course was delivered by Professor Rafael A. Irizarry, Professor of Applied Statistics at Harvard and the Dana-Farber Cancer Institute. The course is hosted and delivered via the edX open online course provider founded in May 2012 by scientists from Harvard and MIT.

Road accident fatalities within the European Union are amongst the lowest levels recorded globally. However annually more than 25 000 people still lose their lives on EU roads, while another 135 000 are seriously injured (European Commission 2018). This means that for every person killed in traffic crashes, five more suffer serious injuries, resulting in long term rehabiliation and healthcare needs. Depsite the downward trend the figures are still extremely high and a significant factor in annual mortality rates internationally. This project reviews the 2017 road accident datasets provided by the French government and build an understanding from the collision data to develop a model that predicts the severity of an accident that has occurred to aid emergency services manage the first responder services needed.

1.2 Project Goals

The goal of the project is to choose your own project to apply machine learning techniques that go beyond standard linear regression. This project compares the accuracy of Generalised Linear Model (GLM) and RPart classification to predict the severity of an accident that has ocurred on a French road. The goal is to provide a machine learning tool that would enable the emergency services to estimate the potential severity of a call received and send the appropriate emergency response units to the location.

The project submission aims to:

- provide a demonstration and application of Data Science modelling techniques and R programming skills learned throughout the course;
- communicate the project methodology followed; and
- present insights gained from the analysis of the dataset.

The project introduces Geographic Information System (GIS) spatial data analysis to provide detailed insights into the data through shapefile rendering in order to add an additional option for data exploration and visual analysis.

1.3 Submission requirements

The following files are submitted for assessment and graded by a scoring rubric that has been defined by course staff:

1. A report in the form of an Rmd (R Markdown) file
2. A report in the form of an Adobe PDF document knit from the Rmd file
3. An R script or Rmd file that generates predicted injury severity for French traffic accidents and calculates the model accuracy

Each submission (reports and script) will be graded by course peers and a course staff member.

The traffic accident severity predictions are compared to the true ratings in a test validation set using model accuracy.

The report documents the analysis and presents the findings, along with supporting statistics and figures including the accuracy generated for the multiple models assessed.

1.4 Dataset

data.gouv.fr is an open platform for French public data intended to encourage the reuse of data beyond the primary use of the data by the administration. The French traffic datasets from 2017 are selected as it provided detailed accident information for a calendar year. It also presents several data science challenges as the figures require significant data cleaning and contains mostly categorical data. The categorical data must be treated and binarised in order to apply the Generalised Linear Model (GLM) and RPart classification models. The data is sourced from the accident reports made by the driver/police unit (police, gendarmerie, etc.) who intervened at the accident site. This data is collected in a sheet entitled "bulletin of accident analysis". All of these forms constitute the national file of personal injury traffic known as "BAAC file" administered by the National Interministerial Observatory of Road Safety "ONISR".

The 2017 French Regions, Department and Communes were sourced from INSEE (Institut National de la Statistique et des Etudes Economiques) and the population, region and department information joined together and the resulting file stored on the github project directory. The National Institute of Statistics and Economic Studies (INSEE) collects, produces, analyzes and disseminates information on the French economy and society.

The maps of France used in the exploratory data analysis section of this project were also sourced from the data.gouv.fr platform. However due to the extremely large file size of the shapefiles the original file was reduced to a significantly smaller size to greatly improve the performance of rendering the graphics. The files are also stored in the github project directory. Shapefiles are used extensively to store spatial information and can be used to plot data on maps (Viswanathanm et al). It is a geospatial vector data format spatially describing data such as points, lines, and polygons, representing for example borders, landmarks, roads, and lakes.

- https://mapshaper.org - Excellent online tool that was used to reduce the shapefile size to 20% enabling map rendering in seconds rather than minutes. All four files (.shp, .dbf, .prj, .shx) are necessary when processing the shapefile and enabling data to be linked to the shapefile elements.

Data File	Notes
communes2017.xlsx	Offical list of French Regions, Departments and Communes published in 2017
departements-100m.shp	Shape format; the feature geometry itself
departements-100m.dbf	Attribute format; columnar attributes for each shape,#dBASE format
departements-100m.prj	Projection description of coordinates
departements-100m.shx	Shape index format; a positional index of the feature geometry to allow seeking forwards and backwards quickly
vehicles.csv	Details of the vehicles involved in the accidents
users.csv	Details of the people involved in the accidents. Reported by Accident and Vehicle
places.csv	Detailed information of the location and infrastructure where the accident occurred
characteristics.csv	Date, location (Department, Commune), weather, atmospheric conditions

1.5 Data Science Pipeline

A data science pipeline is the overall process to prepare, import, tidy, visualise, model, interpret and communicate data (Wickham and Grolemund 2017). As defined by Zacharias Voulgaris, it "is a complex process comprised of a number of inter-dependent steps, each bringing us closer to the end result, be it a set of insights to hand off to our manager or client, or a data product for our end-user." (Voulgaris 2017, ch. 2)

For the purpose of this report, methodologies will refer specifically to machine learning methodologies. I make the distinction here between the definition of **pipeline** and **methodology** due to my background in enterprise applications where methodology in the context of software development lifecycle (SDLC) refers to the waterfall or agile methodologies.

The following pipeline steps where followed for the project and have been documented in separate sections on this report.

Table 2: Data Science Pipeline

Pipeline	Step	Goal
Data Source:		
	Data Extraction	**ETL: Source**
	Data Load	**ETL: Import**
Data Preparation:		
	Data processing and wrangling	**ETL: Clean& Transform**
	Data Exploration	**Initial Discovery**
	Data Visualisation	**Visualise**
Data Modeling:		
	Feature Extraction & Engineering	**Identify Outcome/Features**
	Develop Models	**Capture Pattern**
	Data Learning	**Model Evaluation**
Communication:		
	Summary	**Results**
	Insights	**Intuition**
	Citations and References	**References**

2 Packages

Table 3: Package Installation notes: Additional information for packages required to support this report

Package	Note
bbcplot	British Broadcasting Corporation package for extending ggplot2 themes
broom	Takes the messy output of built-in functions in R and turns them into tidy data frames.
caret	Set of functions to streamline prediction model creation
devtools	Collection of R development tools
GGally	Multivariate plots
ggalt	Support geom_dumbbell() plots
pacman	Contains tools to more conveniently perform tasks associated with add-on packages.
plotly	Graphing library makes interactive, publication-quality graphs online
pROC	Tools for visualizing, smoothing and comparing receiver operating characteristic (ROC curves)
readxl	Excel xlsx integration from Hadley Wickham
rgdal	Provides bindings to the 'Geospatial' Data Abstraction Library ('GDAL')
rpart.plot	Extends plot.rpart() and text.rpart() in the 'rpart' package.
sjmisc	Collection of miscellaneous utility functions integrated into a tidyverse workflow
snakecase	Collection of miscellaneous utility functions, supporting data transformation
stringi	Character String Processing Facilities. Used for formatting lat/long in this project
tictoc	Performance Benchmarking
vcd	Support mosaics
xtable	Added for RMarkdown file functionality
kableExtra	Added for RMarkdown file functionality

```r
if(!require(pacman))install.packages("pacman")
if(!require(devtools))install.packages('devtools')
devtools::install_github('bbc/bbplot')    #Load the BBC plots for use with ggplot2
pacman::p_load('devtools',                                # Development
               'data.table','readxl',                     # Data Importation
               'tidyverse', 'dplyr', 'tidyr', 'stringr',  # Data Manipulation
               'sjmisc', 'snakecase', 'lubridate',        # Data Manipulation
               'stringi',                                 # Data Manipulation
               'ggplot2', 'bbplot', 'ggalt','GGally',     # Visualisation
               'vcd',                                     # Visualisation
               'rgdal', 'plotly',                         # Cartography
               'caret', 'rpart.plot', 'pROC',             # Classification and Regression
               'xtable', 'kableExtra',                    # RMarkdown
               'tictoc')                                  # Performance measuring
```

3 Data Retrieval

The files are downloaded from the data.gouv.fr is an open platform and github project directory created for this project submission. A function module has been written to check if the files have already been downloaded. This avoids the need to download the files from remote servers each time the project is executed.

Key note: Download the files to your working directory. The files does not need to be added to a subfolder for processing.

Learning Point: **read_csv2()** is used instead of read_csv() as the files are semi-colon seperated (common approach in the European Union). read_csv2() is preferred to read.csv2 because a tibble is created but also due to the intepretation of the commune codes. A code of 08 is treated by default as 8 using read.csv2 which is not desired for this project as the code 08 is the formal code used by the French authorities.

There is an issue of codepage when downloading the data. E.g. "dpartementale" should be "départementale". Therefore the codepage locale is set to Latin 1 using **readr::locale** to ensure all West European characters can be interpreted when loading the file. For a global dataset a more comprehensive locale would be necessary.

The commands **tic()** and **toc()** are used to record the processing time for some critical blocks of code. This was useful to benchmark and compare different techniques used in the code. The code is not displayed in the generated report (using **echo=FALSE** in the RMarkdown chunk definition)[1].

```
#Set up the directory and file locations
datagov_url <- "https://www.data.gouv.fr/en/datasets/r/"
vehicles_url <- "109520e1-47ae-4294-a5b6-6d10c7bae9a6"        # Comma delimited
users_url <- "07bfe612-0ad9-48ef-92d3-f5466f8465fe"           # Comma delimited
places_url <- "9b76a7b6-3eef-4864-b2da-1834417e305c"          # Comma delimited
characteristics_url <- "9a7d408b-dd72-4959-ae7d-c854ec505354" # Comma delimited

#2017 Communes were sourced from INSEE (Institut National de la Statistique et des Etudes Economiques)
github_url   <- "https://github.com/iverni/PH125.9x_CYO/blob/master/Data%20files/"
commune_file <- "communes2017.xlsx?raw=true"

#All four files are necessary when processing the shapefile and enabling data to be linked to the shapefile elements.
shapefile_shp <- "departements-100m.shp?raw=true"
shapefile_dbf <- "departements-100m.dbf?raw=true"
shapefile_prj <- "departements-100m.prj"
shapefile_shx <- "departements-100m.shx?raw=true"

#Function module to check if the file has already been downloaded.
download_datefile <- function(url_ref, new_filename) {
  if(!file.exists(new_filename)){
    download.file(url_ref, new_filename)
  }else{
    print(new_filename)
    print("File already exist and do not need to be downloaded again")
  }
}
#Download the French region, department and commune information
download_datefile(paste0(github_url,commune_file),"communes2017.xlsx")
```

```
## [1] "communes2017.xlsx"
## [1] "File already exist and do not need to be downloaded again"
```

```
communes <- read_excel("communes2017.xlsx", sheet = 1)

#Retreive the shapefiles. They will be read during exploratory data analysis
download_datefile(paste0(github_url,shapefile_shp),"departements-100m.shp")
```

```
## [1] "departements-100m.shp"
## [1] "File already exist and do not need to be downloaded again"
```

```
download_datefile(paste0(github_url,shapefile_dbf),"departements-100m.dbf")
```

```
## [1] "departements-100m.dbf"
## [1] "File already exist and do not need to be downloaded again"
```

```
download_datefile(paste0(github_url,shapefile_prj),"departements-100m.prj")
```

```
## [1] "departements-100m.prj"
## [1] "File already exist and do not need to be downloaded again"
```

[1] The LaTeX font size can be adjusted in RMarkdown by setting the size to tiny, scriptsize, footnotesize, small, normalisize, large, huge. See the examples provided in this RMD report

```r
download_datefile(paste0(github_url,shapefile_shx),"departements-100m.shx")
```

```
## [1] "departements-100m.shx"
## [1] "File already exist and do not need to be downloaded again"
```

```r
# Download the vehicles file. Ensure that the accident reference is a character so that it is not treated as a double.
# col_types ensures that the accident number 201700000001 is not imported as 2.017e+11.
download_datefile(paste0(datagov_url,vehicles_url),"vehicles.csv")
```

```
## [1] "vehicles.csv"
## [1] "File already exist and do not need to be downloaded again"
```

```r
vehicles <- read_csv("vehicles.csv", col_types = cols(Num_Acc = "c"))

#Download user data
download_datefile(paste0(datagov_url,users_url),"users.csv")
```

```
## [1] "users.csv"
## [1] "File already exist and do not need to be downloaded again"
```

```r
users <- read_csv("users.csv", col_types = cols(Num_Acc = "c"))

#Download location data
download_datefile(paste0(datagov_url,places_url),"places.csv")
```

```
## [1] "places.csv"
## [1] "File already exist and do not need to be downloaded again"
```

```r
places <- read_csv("places.csv", col_types = cols(Num_Acc = "c"))

#Download accident characteristic information.
download_datefile(paste0(datagov_url,characteristics_url),"characteristics.csv")
```

```
## [1] "characteristics.csv"
## [1] "File already exist and do not need to be downloaded again"
```

```r
characteristics <- read_csv("characteristics.csv", col_types = cols(Num_Acc = "c"), locale = readr::locale(encoding = "latin1"))
#Concatenate Department and Commune columns together to support a join of the French state communes.
#Use substr to only select the first two characters of the department field
characteristics <- characteristics %>% mutate(depcom = paste0(substr(dep, start = 1, stop = 2),com))
toc()
```

4 Data Cleansing

Each file must be checked and cleaned before joining the data together to support exploratory data analysis. Each files is reviewed for missing data, near zero variance features and inconsitencies in the data content or format. The data cleanse for these files was one of the most time consuming activities for this project submission. Data preparation and wrangling is a critical and fundamental phase of a data science project. Without it you cannot work with your own data (Hadley and Grolemund 2017).

4.1 Characteristic File

For obtaining summary information the **str()** command was preferred over **glimpse()** due to processing performance. All efforts are made to reduce the performance requirements when running the RMarkdown report. Runtime memory is also managed using the **rm()** command.

```
##    Num_Acc                an          mois             jour      
##  Length:60701       Min.   :17   Min.   : 1.000   Min.   : 1.00  
##  Class :character   1st Qu.:17   1st Qu.: 4.000   1st Qu.: 8.00  
##  Mode  :character   Median :17   Median : 7.000   Median :16.00  
##                     Mean   :17   Mean   : 6.611   Mean   :15.68  
##                     3rd Qu.:17   3rd Qu.:10.000   3rd Qu.:23.00  
##                     Max.   :17   Max.   :12.000   Max.   :31.00  
##                                                                  
##       hrmn           lum             agg             int       
##  Min.   :   1   Min.   :1.000   Min.   :1.000   Min.   :0.000  
##  1st Qu.: 950   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000  
##  Median :1447   Median :1.000   Median :2.000   Median :1.000  
##  Mean   :1376   Mean   :1.907   Mean   :1.634   Mean   :1.843  
##  3rd Qu.:1810   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:2.000  
##  Max.   :2359   Max.   :5.000   Max.   :2.000   Max.   :9.000  
##                                                                 
##       atm             col            com                adr           
##  Min.   :1.000   Min.   :1.000   Length:60701       Length:60701      
##  1st Qu.:1.000   1st Qu.:3.000   Class :character   Class :character  
##  Median :1.000   Median :4.000   Mode  :character   Mode  :character  
##  Mean   :1.573   Mean   :4.206                                        
##  3rd Qu.:1.000   3rd Qu.:6.000                                        
##  Max.   :9.000   Max.   :7.000                                        
##  NA's   :13      NA's   :6                                            
##     gps                 lat              long          
##  Length:60701       Min.   : 434091   Length:60701      
##  Class :character   1st Qu.:4481995   Class :character  
##  Mode  :character   Median :4766188   Mode  :character  
##                     Mean   :4627474                     
##                     3rd Qu.:4884970                     
##                     Max.   :5107423                     
##                     NA's   :7731                        
##     dep               depcom         
##  Length:60701       Length:60701      
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
##                                       
```

```
##
```

Missing values are checked to identify any field that require cleaning. From the analysis of the NAs only the atm (atmospheric conditions) and col (collison) features will be corrected. The adr (address) field is the postal address filled in for accidents occurring in built up areas. This is not required (but will be checked by performing a NZV).

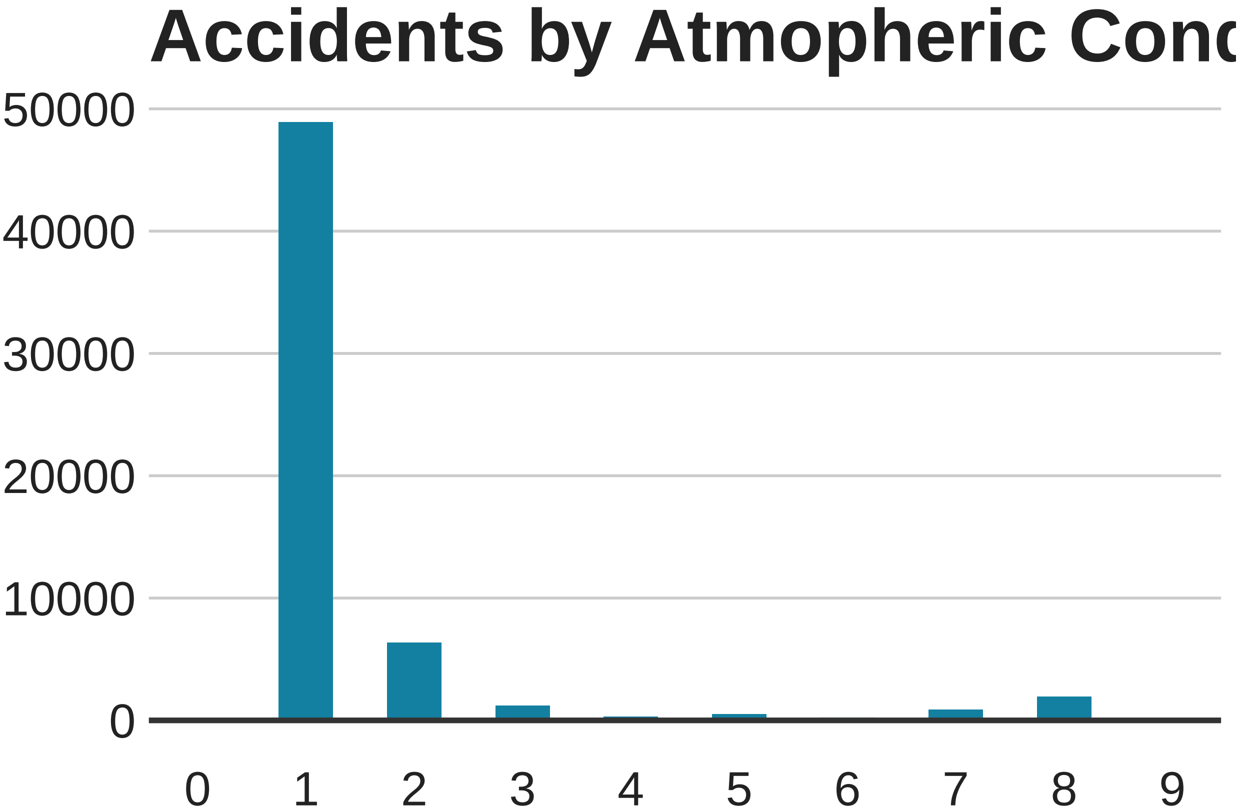
```
#Calculate all the NAs present in each column/feature and then assess if any action is required.
sapply(characteristics, function(x) sum(length(which(is.na(x)))))
```

```
## Num_Acc      an    mois    jour    hrmn     lum     agg     int     atm
##       0       0       0       0       0       0       0       0      13
##     col     com     adr     gps     lat    long     dep  depcom
##       6       0     822    4608    7731    7731       0       0
```

The GPS, lat and long co-ordinate information is not required for this project. GIS data will be summarised at a department level

Replace the atm feature with the median. A histogram analysis of the values shows that the median is “normal” atmospheric conditions.

```
characteristics %>% ggplot(aes(atm)) +
  geom_histogram(fill="#1380A1", binwidth = .5) +
  geom_hline(yintercept = 0, size = 1, colour="#333333") +
  scale_x_continuous(breaks = seq(0, 9, 1),
                     limits=c(0, 9)) +
  xlab("Atmospheric Condition") +
  labs(title="Accidents by Atmopheric Cond.") +
  bbc_style()
```



```
characteristics$atm <- ifelse(is.na(characteristics$atm), median(characteristics$atm, na.rm=TRUE), characteristics$atm)
```

Repeating the check for Na values shows the issue has been corrected

```
sapply(characteristics, function(x) sum(length(which(is.na(x)))))
```

```
## Num_Acc      an    mois    jour    hrmn     lum     agg     int     atm
##       0       0       0       0       0       0       0       0       0
##     col     com     adr     gps     lat    long     dep  depcom
##       6       0     822    4608    7731    7731       0       0
```

The median for the collision type is 6 - Other collision. However the value 6 is chosen as the replacement value because of "other collision" is the best category to apply for the missing data, irrespective of the median.

```
characteristics$col <- ifelse(is.na(characteristics$col), median(characteristics$col, na.rm=TRUE), characteristics$col)
```

Latitude and longitude fields need to be formatted correct. The **stringi** package provides an excellent command that enables characters to be replaced with the open to add another value. The accident time of day is stratified as either day or night to reduce the complexity when applying prediction models. The function **sprintf()** to format the values in the existing file to a required format to supoort the creation of a POSIX date and timestamp.

```
#For example, the first two characters of longitude are extracted e.g.
characteristics$long <- stri_sub_replace(characteristics$long, 3,1, omit_na=FALSE, value = ".")
characteristics$lat <- stri_sub_replace(characteristics$lat, 3,1, omit_na=FALSE, value = ".")
#Convert hrmn to hours and minutes
#Format the timestamp
convert_date_format <- function(year, day, month, hour){
  #Build up the ISO 8601 formatting field
  month_2c <- sprintf("%02d",month)
  day_2c <- sprintf("%02d",day)
  hour_4c <- sprintf("%04d",hour)
  date_temp <- paste0(year, month_2c, day_2c, " ", hour_4c)
  as_datetime(date_temp, "%y%d%m %H%M", tz="CET")
}
#Group by day or night
#Round the time to the nearest hour in order to group by hour later
characteristics <- characteristics %>%
    mutate(day_night = ifelse(lum == 1, "day","night"),
           date_cet = as.POSIXct(convert_date_format(an, mois, jour, hrmn)),
           hrs = as.numeric(hour(round_date(date_cet, unit = "1 hour"))))
```

4.2 Places File

```
#Calculate all the NAs present in each column/feature and then assess if any action is required.
sapply(places, function(x) sum(length(which(is.na(x)))))
```

```
## Num_Acc    catr    voie      v1      v2    circ     nbv      pr     pr1
##       0       0    9789   60293   58007     371     434   34494   34733
##    vosp    prof    plan  lartpc larrout    surf   infra    situ    env1
##     596     449     785    2163    2020     465    3664    3474    3705
```

The following fields will be cleaned:

- circ - Traffic Regime. the missing and 0 value entries in the column denote no recording of the traffic regime. Valid values are: 1 - One way, 2 - Bidirectional, 3 - Separated carriageways 4 With variable assignment channels. The missing values are defaulted to category 0 - no indication of direction
- nbv - total number of traffic lanes.
- vosp - Indicates the existance of a reserved lane.
- prof - Profondeur / Terrain type - 1 Flat, 2 Slope, 3 hill top, 4 Hill bottom

- surf - Surface condition 9 indicates “other”. All 0 values and NAs should be moved to 9
- infra - Infrastructure e.g. underground, bridge
- situ - Situation of the accident

```
places$circ <- ifelse(is.na(places$circ), 0, places$circ)
places$nbv <- as.numeric(ifelse(is.na(places$nbv), 0, places$nbv))
places$vosp <- ifelse(is.na(places$vosp), 0, places$vosp)
places$prof <- ifelse(is.na(places$prof), 0, places$prof)
places$surf <- ifelse(is.na(places$surf), 9, places$surf)
places$surf <- ifelse(places$surf == 0, 9, places$surf)
places$infra <- ifelse(is.na(places$infra), 0, places$infra)
places$situ <- ifelse(is.na(places$situ), 0, places$situ)
```

The following columns are removed due to the large NA occurences but also due to their perceived lack of importance for prediction:

- voie - identifies the number of the road
- V1 - numeric index of the route number
- V2 - Letter alphanumeric index of the road
- pr - Home pr number (used for measurements on French roads)
- pr1 - Number of the distances in metres “bornes”
- plan- Road contour
- lartpc- Central solid land width
- larrout - Witdh of the roadway
- env1 - Proximity to schools. Unfortunately the dataset is not clear and the feature should be removed

```
places <- places %>% select(-voie, -v1, -v2, -pr, -pr1, -plan, -lartpc, -larrout, -env1)
```

4.3 Users File

```
#Calculate all the NAs present in each column/feature and then assess if any action is required.
sapply(users, function(x) sum(length(which(is.na(x)))))
```

```
## Num_Acc   place    catu    grav    sexe  trajet    secu    locp    actp
##       0   11802       0       0       0      11    8950      46      43
##   etatp an_nais num_veh
##      66      37       0
```

To contrast how values can be checked for relevancy run a Near Zero Variance:

- freqRatio: This is the ratio of the percentage frequency for the most common value over the second most common value.
- percentUnique: This is the number of unique values divided by the total number of samples multiplied by 100.

For percentUnique, the lower the percentage, the lower the number of unique values. A high freqRatio indicates that the distributions is heavily skewed. It does not mean we want to remove the data, but it provides an indication of the distribution.

If the NZV is TRUE then it should be removed.

```
nzv_users <- nearZeroVar(users, saveMetrics = TRUE)
nzv_users
```

```
##         freqRatio percentUnique zeroVar   nzv
## Num_Acc  1.180000  44.626197425   FALSE FALSE
## place    6.694507   0.006616625   FALSE FALSE
## catu      4.244627   0.002940722   FALSE FALSE
## grav      1.193223   0.002940722   FALSE FALSE
## sexe      2.095819   0.001470361   FALSE FALSE
## trajet    1.932456   0.005146264   FALSE FALSE
## secu      3.263561   0.013233251   FALSE FALSE
## locp     36.499708   0.006616625   FALSE  TRUE
## actp     14.583372   0.005881445   FALSE FALSE
## etatp    14.324449   0.002940722   FALSE FALSE
## an_nais   1.047157   0.075723602   FALSE FALSE
## num_veh   1.832279   0.028672043   FALSE FALSE
```

Now compare NZV of the "places" file. Was there a pattern missed. It recommends taking out vosp and infra. Incidently, a **nzv()** check on characteristics identified year and GPS and NZV. Not surprising given year = 2017. This is a good demonstration of NZV.

```
nzv_places <- nearZeroVar(places, saveMetrics = TRUE)
nzv_places
```

```
##          freqRatio percentUnique zeroVar   nzv
## Num_Acc   1.000000  1.000000e+02   FALSE FALSE
## catr      1.271553  1.153194e-02   FALSE FALSE
## circ      3.349185  8.237097e-03   FALSE FALSE
## nbv       5.045851  2.141645e-02   FALSE FALSE
## vosp     33.138533  6.589677e-03   FALSE  TRUE
## prof      5.026617  8.237097e-03   FALSE FALSE
## surf      4.758593  1.482677e-02   FALSE FALSE
## infra    19.600500  1.317935e-02   FALSE  TRUE
## situ      6.820688  9.884516e-03   FALSE FALSE
```

```
nzv_characteristics <- nearZeroVar(characteristics, saveMetrics = TRUE)
nzv_characteristics
```

```
##            freqRatio percentUnique zeroVar   nzv
## Num_Acc     1.000000  1.000000e+02   FALSE FALSE
## an          0.000000  1.647419e-03    TRUE  TRUE
## mois        1.074403  1.976903e-02   FALSE FALSE
## jour        1.011364  5.107000e-02   FALSE FALSE
## hrmn        1.065949  2.255317e+00   FALSE FALSE
## lum         4.289835  8.237097e-03   FALSE FALSE
## agg         1.729607  3.294839e-03   FALSE FALSE
## int         5.929547  1.647419e-02   FALSE FALSE
## atm         7.716832  1.482677e-02   FALSE FALSE
## col         1.254625  1.153194e-02   FALSE FALSE
## com         3.779037  1.336057e+00   FALSE FALSE
## adr         1.037671  7.472694e+01   FALSE FALSE
## gps        76.411017  8.237097e-03   FALSE  TRUE
## lat         1.568182  7.149965e+01   FALSE FALSE
## long        1.045455  7.734469e+01   FALSE FALSE
## dep         1.548555  1.663894e-01   FALSE FALSE
## depcom      4.087805  1.959441e+01   FALSE FALSE
## day_night   2.064778  3.294839e-03   FALSE FALSE
## date_cet    1.000000  7.163473e+01   FALSE FALSE
## hrs         1.118598  3.953806e-02   FALSE FALSE
```

The field trajet is the reported reason for travelling on the accident form. 9 indicates "other". All 0 values and NAs should be moved to 9.

```
users$trajet <- ifelse(is.na(users$trajet), 9, users$trajet)
users$trajet <- ifelse(users$trajet == 0, 9, users$trajet)
```

There are 37 cases of the age not being recorded. The median value is used.

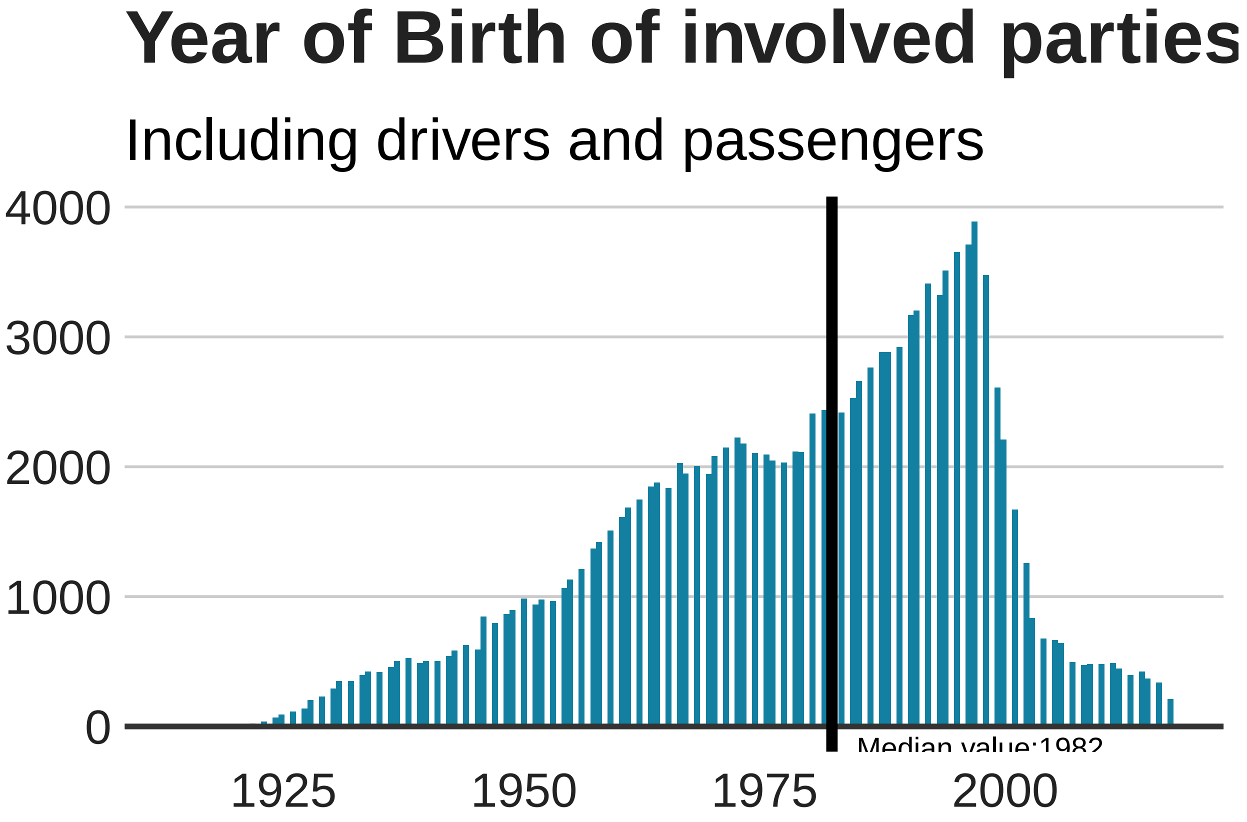
```
median(users$an_nais, na.rm=TRUE)
```

```
## [1] 1982
```

```
users$an_nais <- ifelse(is.na(users$an_nais), median(users$an_nais, na.rm=TRUE), users$an_nais)
```

The severity of accidents is grouped to critical (fatal or serious injury) and normal response (light hospitalisation, no injury). The age profile is split into strata.

```
users <- users %>% mutate(gender = ifelse(sexe == 1, "male","female"),
                          severity = case_when(
                            grav == 1 ~ "normal",   #No injury
                            grav == 2 ~ "critical", #Fatality
                            grav == 3 ~ "critical", #Seriously injured
                            grav == 4 ~ "normal"    #Light injury
                          ),
                          age = 2017-an_nais,
                          age_profile = case_when(
                            age <= 15 ~ "child",
                            age > 15 & age <= 19 ~ "teenager",
                            age > 19 & age < 40 ~ "under40",
                            age >= 40 ~ "over40"))

#histogram of year of birth
users %>% ggplot(aes(an_nais)) +
  geom_histogram(fill="#1380A1", binwidth = .6) +
  geom_hline(yintercept = 0, size = 1, colour="#333333") +
  xlab("Year of Birth") +
  labs(title="Year of Birth of involved parties",
       subtitle="Including drivers and passengers") +
  geom_vline(aes(xintercept = median(an_nais)),col='black', size = 2) +
  geom_text(data = data.frame(x=median(users$an_nais), y=0), mapping = aes(x, y, label=paste0("Median value:",x)), color="black", hjust =-.1, vjus
  bbc_style()
```



The following columns are removed due to the large NA occurences but also due to their perceived lack of importance for prediction:

- place - #56% missing
- secu - dataset is not sufficiently clear to use
- locp - Pedestrian Location - remove due to NZV analysis
- actp - Action of the Pedestrian - remove due to NZV analysis
- etapt - Was the pedestrian alone, in a group - removed as other pedestrian fields are removed

```
users <- users %>% select(-place, -secu, -locp, -actp, -etatp)
```

4.4 Vehicles File

```
#Calculate all the NAs present in each column/feature and then assess if any action is required.
sapply(vehicles, function(x) sum(length(which(is.na(x)))))
```

```
## Num_Acc    senc    catv  occutc     obs    obsm    choc    manv num_veh
##       0      68       0       0      55      42      35      30       0
```

```
#Perform a near zero variance analysis on the vehicles dataset
nzv_vehicles <- nearZeroVar(vehicles, saveMetrics = TRUE)
nzv_vehicles
```

```
##          freqRatio percentUnique zeroVar   nzv
## Num_Acc   1.363636  58.622254843   FALSE FALSE
## senc      1.479350   0.002897263   FALSE FALSE
## catv      7.182992   0.023178104   FALSE FALSE
## occutc  396.030769   0.058911015   FALSE  TRUE
## obs      41.345149   0.016417824   FALSE  TRUE
## obsm      3.889195   0.006760280   FALSE FALSE
## choc      2.456631   0.009657544   FALSE FALSE
## manv      2.965841   0.024143859   FALSE FALSE
## num_veh   1.657468   0.041527437   FALSE FALSE
```

The missing values are defaulted to category 0 - no indication.

- choc - Point of the impact. 0 denotes not specified
- catv - Vehicle category. 99 indicates “other vehicles”. All NAs should be moved to 99
- obsm - Mobile object struck. 9 indicates “other”. All 0 values and NAs should be moved to 9

```
vehicles$choc <- ifelse(is.na(vehicles$choc), 0, vehicles$choc)
vehicles$manv <- as.numeric(ifelse(is.na(vehicles$manv), 00, vehicles$manv)) #Encode the categorical
vehicles$catv <- as.numeric(ifelse(is.na(vehicles$catv), 99, vehicles$catv)) #Encode the categorical
vehicles$obsm <- ifelse(is.na(vehicles$obsm), 9, vehicles$obsm)
vehicles$obsm <- ifelse(vehicles$obsm == 0, 9, vehicles$obsm)
```

The following columns are removed due to the large NA occurences but also due to their perceived lack of importance for prediction:

- senc - Direction of the traffic (flow)
- obs - Struck a fixed Obstacle - remove due to NZV analysis
- occutc - Number of occupants in the public transport - remove due to NZV analysis

```
vehicles <- vehicles %>% select(-obs, -occutc, -senc)
toc()
```

```
## Data Cleansing: 10.982 sec elapsed
```

5 Data Preparation

Now that the official files have been cleansed the data will be joined together to create data tables to support exploratory data analysis.

```
tic("Joining data")
```

The following section contains the data table joins necessary to support subsequent exploratory data anaylsis and modelling. One to one cardinality between the accident characteristics, commune and place files. The data is summarised as follows:

- Summarised data at department and commune level : Number of accidents, count of injuries by severity, mortality rate
- Summarised data at department level only : Number of accidents, count of injuries by severity, mortality rate
- Full accident information : Join of the Characteristics, Place, User and Vehicle datasets to support injury analysis by person and vehicle. This dataset is also used in the full model analysis

The accident and mortality rates are calculated per 100,000 inhabitants. The mortality rate is calculated as follows:

$$Mortalityrate_i = \sum_{j=1,}^{N}(Fatalities_{i,j}) * [100,000/\sum_{j=1,}^{N} Population_{i,j}]$$

Where:

N = Full dataset of French Communes where accidents occurred

i = Department

j = Commune

$Fatalities_{i,j}$ = Total deaths for each department summed from each department commune. The source of the data is from the Vehicles table field **grav** (value equal to 2)

$Population_{i,j}$ = Department popluation recorded for 2016 summed from each department commune. The source of the data is from the *communes2017.xlsx* file.

```
accidents <- characteristics %>%
  left_join(places, by = 'Num_Acc')
#Calculate summary data a department and commune level - number of accidents
depcom_summary <- characteristics %>%
  group_by(depcom) %>%
  summarise(accidents = n()) %>%                     #Number of accidents
  left_join(communes, by ="depcom") %>%
  na.omit() %>%                                      #Skip accidents if no valid commune can be found
  mutate(acc_pp = accidents * 100000 / pop2016)      #Accident rate per 100,000 inhabitants

#Count the injury profile (number of people) by department and commune.
depcom_injuries <- accidents %>%
  left_join(users, by = 'Num_Acc') %>%
  left_join(vehicles, by = c("Num_Acc","num_veh")) %>%
  group_by(depcom, grav) %>%
  summarise(n = n()) %>%                               #Count the different injuries by depart/commune
  spread(grav, n) %>%                                  #Spread the injury field to columns
  rename('light' = '1',                                #Rename the columns
         'fatality' = '2',
         'serious' = '3',
         'uninjured' = '4') %>%
  mutate(light, light = ifelse(is.na(light), 0, light), #Replace NAs introduced by the spread with zero value
         fatality, fatality = ifelse(is.na(fatality), 0, fatality),
         serious, serious = ifelse(is.na(serious), 0, serious),
         uninjured, uninjured = ifelse(is.na(uninjured), 0, uninjured))

#Calculate summary data a department and commune level - number of accidents
depcom_summary <- depcom_summary %>%
  left_join(depcom_injuries, by ="depcom") %>%
  mutate(mortality = fatality * 100000 / pop2016,      #Fatalties per 100,000 inhabitants
         serious_inj_rate = (fatality + serious) / (fatality + serious + light + uninjured))
rm(depcom_injuries)

#Calculate summary data a department level only. This will make EDA much easier later.
dep_summary <- depcom_summary %>%
            group_by(department, depart_name) %>%
            summarise(population = sum(pop2016),
                      accidents  = sum(accidents),
                      light      = sum(light),
                      fatality   = sum(fatality),
                      uninjured  = sum(uninjured),
                      serious    = sum(serious)) %>%
            mutate(mortality = fatality * 100000 / population,
                   serious_inj_rate = (fatality + serious) / (fatality + serious + light + uninjured))

#Join full data together (characteristics, plqce, users and vehicle type to support analysis of accident profiles)
#Format department to a 2 character code so that it may be used for joining with the french department shapefiles
accidents_fulldata <- accidents %>%
  mutate(dep = stri_sub(dep,1,2)) %>%
  left_join(users, by = 'Num_Acc') %>%
  left_join(vehicles, by = c("Num_Acc","num_veh"))
toc()
```

```
## Joining data: 1.321 sec elapsed
```

6 Data Exploration and Visualisation

The following summary information will be used throughout the exploratory data analysis section. The code is not displayed for all outputs in order to reduce the size of the report but can be read in the RMarkdown report.

The population of France based on the Communes data file is approximately **66,361,658** inhabitants. The population data contained in the communes file is the summary of the 2016 population per commune and includes the French overseas territories.

During the course of 2017 there was **60701** accidents throughout France. The project focuses on mainland France and does not assess the French overseas territories. A total of **3600** tragically lost their lives in these accidents. This continues a steady downward trend from previous year accident mortality rates for France however it still represents an extremely high mortality rate of **5.4248193** per 100,000 inhabitants. This represents an ongoing challenge to government and emergency services and a tremendous amount of financial and political resource continues to be invested in this area to further reduce the rate.

When considering the number of accidents per 100,000 inhabitants the average rate for France returned is **91.4699871**.

The following table shows the distribution of the injury severity for all people involved in the accidents, where:

- 1 : Light injury occurred not requiring hospitalisation

- 2 : Fatality (died at the scene or within 30 days of the accident)
- 3 : Serious injury occurred requiring hospitalisation
- 4 : No injuries

```
table(users$grav)    #Summary of injury severity.
```

```
##
##     1     2     3     4
## 56270  3600 28993 47158
```

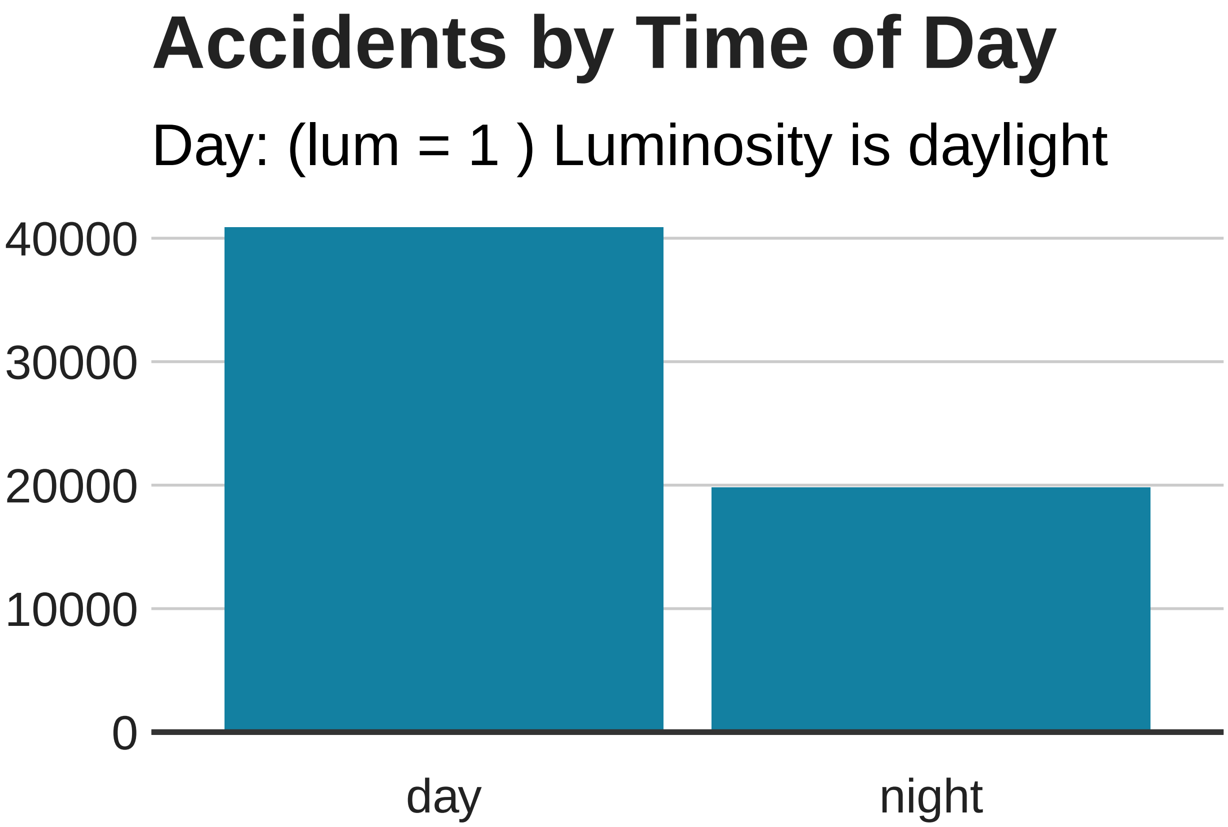
A mixture of styles are used for the graphics to demonstrate different visualisation settings supported by ggplot2. The report requires the installation of the BBC defined R graphics package [2].

6.1 Accident information

Having identified day or night, plot the histogram for all accidents by day or night. Luminosity is daylight (lum =1). The visualisation shows that the majority of accidents occur during daylight.

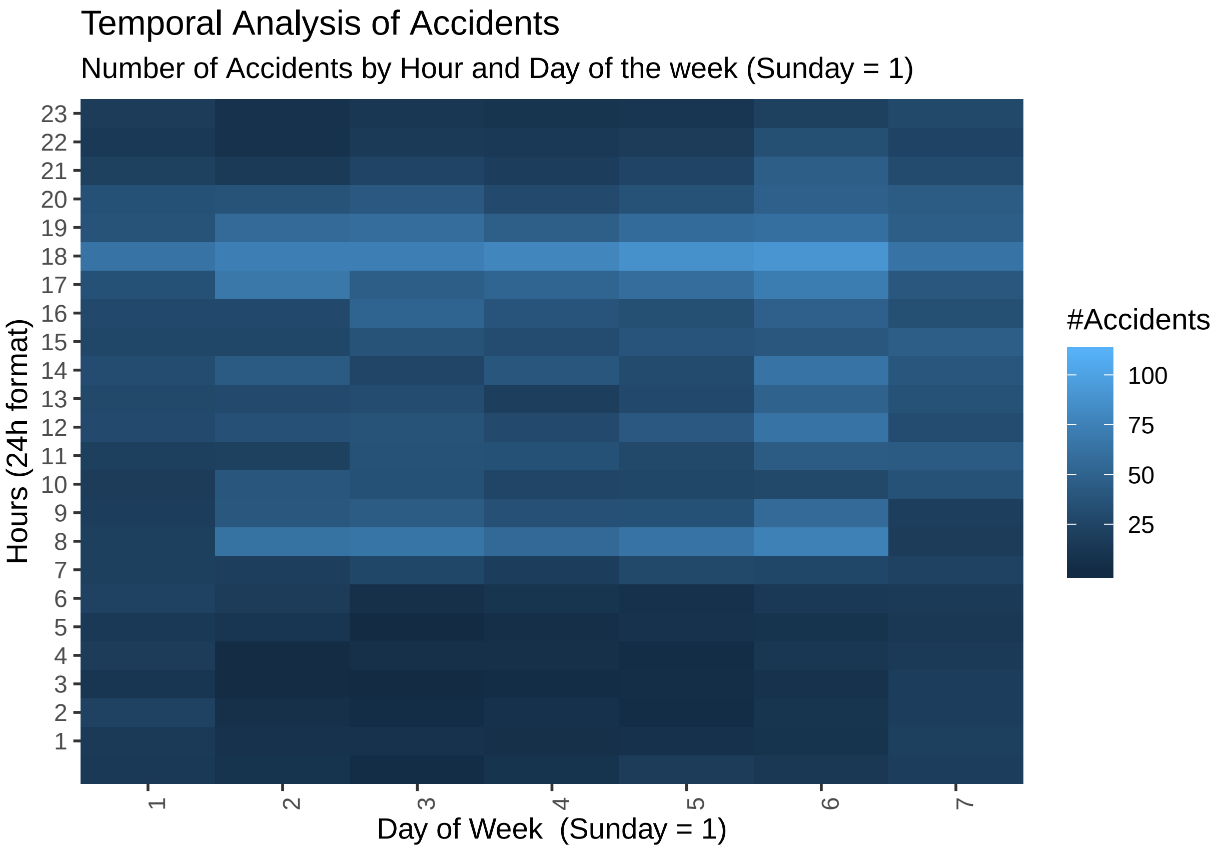
```
characteristics %>% ggplot(aes(day_night)) +
  geom_histogram(fill="#1380A1", stat = "count") +
  geom_hline(yintercept = 0, size = 1, colour="#333333") +
  labs(title="Accidents by Time of Day") +
  labs(subtitle="Day: (lum = 1 ) Luminosity is daylight") +
  bbc_style()
```

[2] The BBC Visual and Data Journalism cookbook for R graphics provides an excellent R package and an R cookbook to make the process of creating publication-ready graphics in the BBC in-house style using R's ggplot2 library.

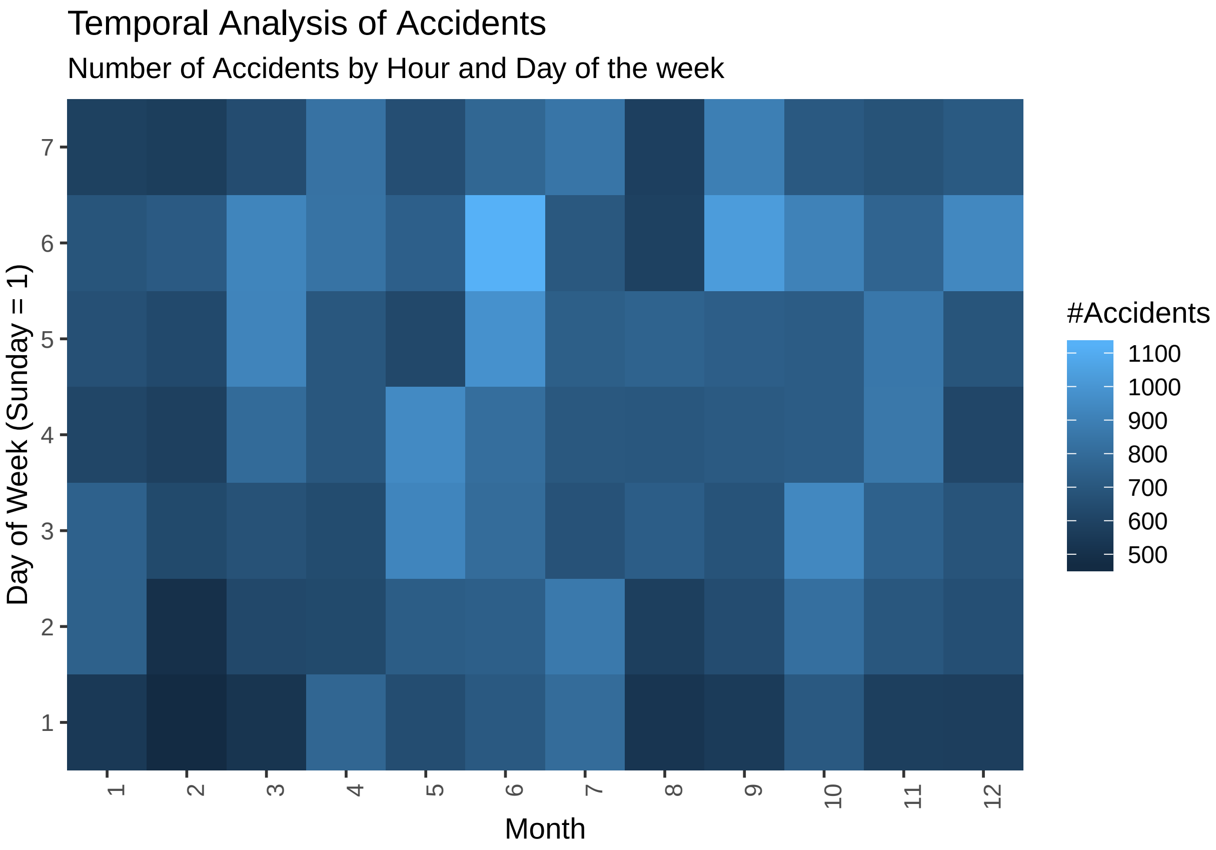


The accident information is assessed for temporal factors - by month, weekday, and time of day. The distribution is displayed first by the Day of the Week and hour of the accidents. The actual week day is determined from the formatting accident time field using the **wday()** function. The aim is to look for the lighter coloured heat zones. This is an indication of the larger frequency. The temporal analysis of the time of day shows two peak periods 08:00-09:00 and 17:00-19:00. Not surprisingly common work communiting hours.

```r
characteristics %>% mutate(day_of_week = wday(date_cet)) %>%
  group_by(mois, day_of_week, hrs) %>%
  summarise(num_accidents = n()) %>%
  ggplot(aes(x=day_of_week, y=hrs, fill=num_accidents)) +   #Plots Weekday and the time of day
  geom_tile() +
  scale_fill_continuous(name="#Accidents") +
  xlab("Day of Week  (Sunday = 1)")+
  ylab("Hours (24h format)")+
  labs(title="Temporal Analysis of Accidents",
       subtitle = "Number of Accidents by Hour and Day of the week (Sunday = 1)") +
  scale_x_continuous(expand=c(0,0),breaks=1:7) +
  scale_y_continuous(expand=c(0,0),breaks=1:24) +
  theme(axis.text.x =element_text(angle = 90, hjust = 1))
```



When assessing by month it is interesting to note that Friday (day=6) features prominently. Of particular note is the number of accidents on Fridays in June. The output of code block in the RMarkdown has been switched off (**echo** = **false**) in order to improve the readability of the report.



6.2 Injury information

6.2.1 List of the Communes with the greatest number of fatalities

```
## # A tibble: 20 x 6
##    commune_name          accidents acc_pp pop2016 mortality serious_inj_rate
##    <chr>                     <int>  <dbl>   <dbl>     <dbl>            <dbl>
##  1 Marseille                  2514   292.  862211     4.18            0.178
##  2 Paris 16e Arrondiss~        615   372.  165446     1.81            0.0352
##  3 Toulouse                    526   111.  475438     2.31            0.0966
##  4 Paris 12e Arrondiss~        523   370.  141494     2.83            0.0576
##  5 Paris 17e Arrondiss~        477   284.  167835     2.38            0.0452
##  6 Paris 18e Arrondiss~        440   226.  195060     1.54            0.0643
##  7 Paris 13e Arrondiss~        426   235.  181552     0               0.0439
##  8 Nice                        406   118.  342637     2.33            0.216
##  9 Cayenne                     387   639.   60580     3.30            0.229
## 10 Paris 19e Arrondiss~        384   206.  186393     0               0.0505
## 11 Rennes                      376   174.  216268     1.85            0.0867
## 12 Paris 20e Arrondiss~        364   186.  195604     0               0.0564
## 13 Paris 8e Arrondisse~        351   963.   36453     2.74            0.0489
## 14 Paris 15e Arrondiss~        349   149.  233484     0.857           0.0489
## 15 Paris 14e Arrondiss~        338   247.  137105     2.19            0.0429
## 16 Tours                       336   246.  136565     2.20            0.0942
## 17 Angers                      331   219.  151229     1.32            0.0850
## 18 Strasbourg                  323   116.  279284     1.07            0.0554
## 19 Saint-Denis                 310   278.  111354     3.59            0.0720
## 20 Paris 11e Arrondiss~        289   197.  147017     1.36            0.0531
```

Marseille has the top number of recorded accidents. This is not surprising given that the entire city has been been grouped at commune level. The number of accidents at **292 per 100,0000 inhabitants** is far greater than the national average of **91.4699871 per 100,0000 inhabitants** and there is also the highest number of fatalities. However it is important to assess the mortality rate wich is **4.18**, less than the national average of **5.4248193** for the 2017. Also 18% of the injuries were serious (that is fatal or seriously hospitalised).

Paris arrondissements feature significantly also and follow the observation of Marseille in that the injury rate and mortality rates are significantly lower than the national average. Therefore perhaps it is more interesting to look at high mortality rates where more than 5 accidents (thereby ignoring outliers).

```
## # A tibble: 20 x 6
##    commune_name          accidents acc_pp pop2016 mortality serious_inj_rate
##    <chr>                     <int>  <dbl>   <dbl>     <dbl>            <dbl>
##  1 Gennevilliers              247  529.    46653      8.57           0.0726
##  2 Saint-Pierre               186  221.    84169      7.13           0.266
##  3 Paris 1er Arrondiss~       176 1083.    16252      6.15           0.0349
##  4 Colombes                   173  203.    85368      7.03           0.0890
##  5 Nîmes                      155  103.   151001      7.28           0.222
##  6 Bastia                     147  328.    44829      8.92           0.274
##  7 Bondy                      144  271.    53193      5.64           0.120
##  8 Le Lamentin                133  331.    40175      9.96           0.0426
##  9 Aix-en-Provence            132   92.3  143006      5.59           0.294
## 10 Brive-la-Gaillarde         119  253.    47004      8.51           0.0650
## 11 Fort-de-France             106  131.    81017      7.41           0.229
## 12 Rueil-Malmaison             98  125.    78195      6.39           0.0905
## 13 Hyères                      94  169.    55772     12.6            0.360
## 14 Cannes                      81  109.    74152      6.74           0.234
## 15 Étampes                     75  307.    24422      8.19           0.0654
## 16 Cagnes-sur-Mer              73  146.    49902      8.02           0.289
## 17 La Ciotat                   71  201.    35366      5.66           0.152
## 18 Fresnes                     71  259.    27416      7.30           0.0655
## 19 Antibes                     70   94.9   73798      8.13           0.371
## 20 Rungis                      70 1248.     5610     17.8            0.102
```

Here we now see an entirely different list. Top of the list is Gennevilliers with 247 accidents and a mortality rate of 8.57. The serious injury rate is 7.26%. In contrast Saint-Pierre in Val-D'Oise had less accidents hozever more fatal (6 deaths/7.13 per 100,000 inhabitants), and a 27% serious injury rate.

6.2.2 List of the Departments with the greatest number of fatalities and discuss

```
## # A tibble: 20 x 6
## # Groups:   department [20]
##    department depart_name   accidents population mortality serious_inj_rate
##    <chr>      <chr>             <int>      <dbl>     <dbl>            <dbl>
##  1 75         PARIS              5948    2190327      1.42           0.0495
##  2 13         BOUCHES-DU-R~      3840    1982821      6.15           0.199
##  3 93         SEINE-SAINT-~      2931    1606660      1.43           0.152
##  4 94         VAL-DE-MARNE       2721    1378151      1.67           0.0871
##  5 92         HAUTS-DE-SEI~      2628    1603268      2.00           0.0991
##  6 97         VAL-D'OISE         1842    1836532      7.84           0.310
##  7 91         ESSONNE            1523    1241650      3.30           0.128
##  8 69         RHONE              1362    1214268      4.36           0.219
##  9 06         ALPES-MARITI~      1351    1051279      5.23           0.286
## 10 33         GIRONDE            1159    1332618      5.48           0.207
## 11 59         NORD               1094    2192167      3.74           0.353
## 12 31         HAUTE-GARONNE       994    1105455      4.79           0.198
## 13 95         VAL-D'OISE          973    1101143      3.54           0.184
## 14 83         VAR                 870    1001212      7.19           0.341
## 15 77         SEINE-ET-MAR~       868    1161405      6.97           0.284
## 16 35         ILLE-ET-VILA~       839     816331      6.61           0.218
## 17 37         INDRE-ET-LOI~       823     488267      6.55           0.203
## 18 78         YVELINES            803    1257362      3.90           0.272
## 19 29         FINISTERE           797     799231      6.01           0.255
## 20 67         BAS-RHIN            788     817306      5.38           0.200
```

Intuition would have suggested Paris as the department with the top number of accidents. However not so obvious is that the mortality rate is significantly less than the national average. Compare the figures to Val-D'Oise were there was 144 fatalaties, with a 31% severe injury rate. This contrasts sharpely with the figures from Paris. It may be more interesting to view by the mortality rate

```
## # A tibble: 20 x 6
## # Groups:   department [20]
##    department depart_name   accidents population mortality serious_inj_rate
##    <chr>      <chr>             <int>      <dbl>     <dbl>            <dbl>
##  1 70         HAUTE-SAONE         118      93154      35.4            0.543
##  2 04         ALPES-DE-HAU~       204     121690      24.7            0.398
##  3 39         JURA                159     131386      23.6            0.501
##  4 55         MEUSE                92      77983      20.5            0.349
##  5 09         ARIEGE              127      94704      18.0            0.409
##  6 58         NIEVRE              207     135456      17.0            0.336
##  7 23         CREUSE               72      53141      16.9            0.524
##  8 32         GERS                164      97122      16.5            0.397
##  9 89         YONNE               210     194606      16.4            0.385
## 10 28         EURE-ET-LOIR        202     262095      16.4            0.444
## 11 41         LOIR-ET-CHER        278     227465      14.9            0.369
## 12 36         INDRE               167     127193      14.9            0.287
## 13 52         HAUTE-MARNE         110      92329      14.1            0.395
## 14 02         AISNE               239     286143      14.0            0.425
## 15 40         LANDES              168     259619      13.9            0.448
## 16 47         LOT-ET-GARON~       185     214869      13.5            0.471
## 17 82         TARN-ET-GARO~       177     203819      13.2            0.450
## 18 15         CANTAL              128      90713      13.2            0.417
## 19 71         SAONE-ET-LOI~       321     359862      12.8            0.383
## 20 48         LOZERE               52      39230      12.7            0.55
```

When viewing by mortality rate the departments that are towards the top of the list are Alpine boarding departments (e.g. Haute-Saone, Alpes-De-Haute-Provence, and JURA, or Ariege in the French Pyrennes)

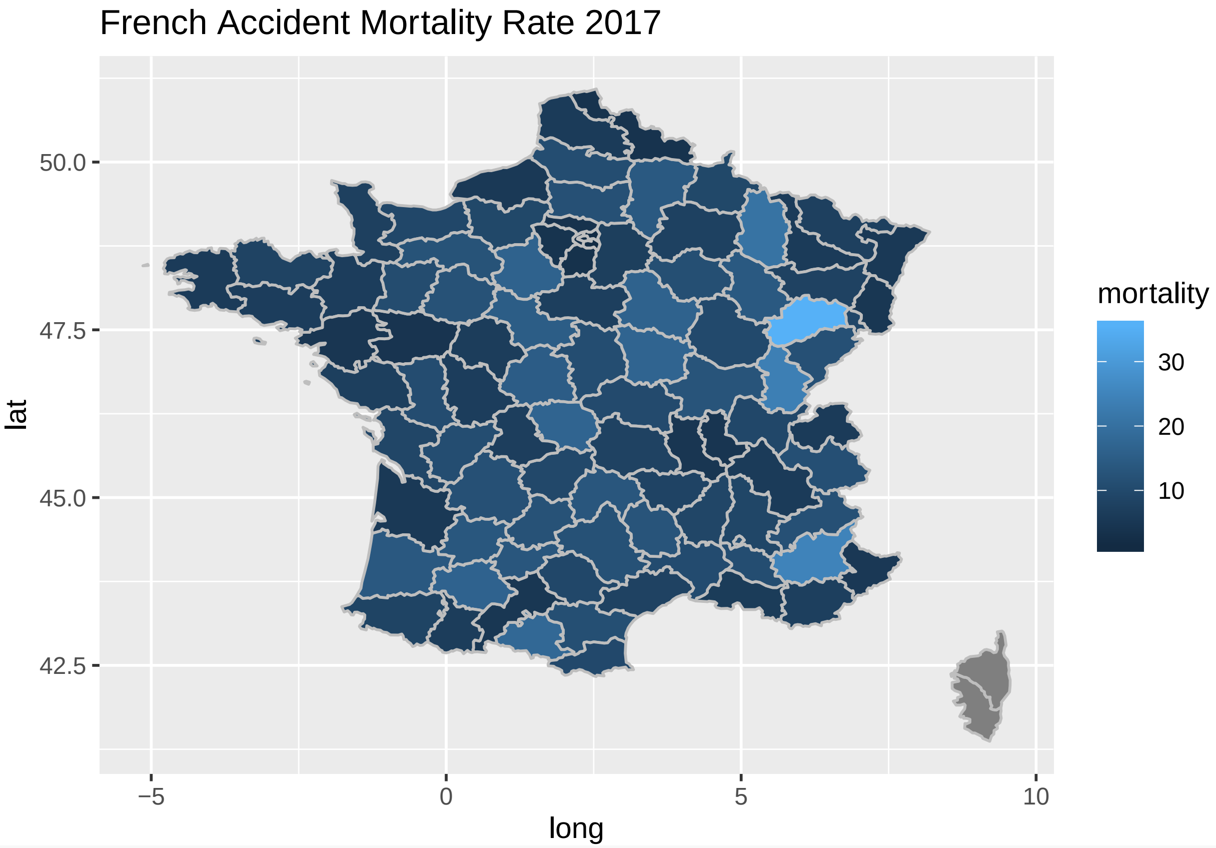
6.3 Cartography

The following section presents the summary information at department level using a shapefile department map of France. The shapefile enables the map to be rendered in R and a dataset linked to the shapefile to enable GIS visualisation. The French overseas territories are removed from the the mapping as this would result in a small scale map given the distribution of the French overseas territories throughout the world.

```
library(webshot)
shapefile_name <- paste0(getwd(), "/departements-100m.shp")
france_shp <- readOGR(shapefile_name, stringsAsFactors = FALSE)  #Read the shapefile that has been downloaded
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/Trevor/DataScience/projects/PH125.9x_CYO/departements-100m.shp", layer: "departements-100m"
## with 101 features
## It has 4 fields
```

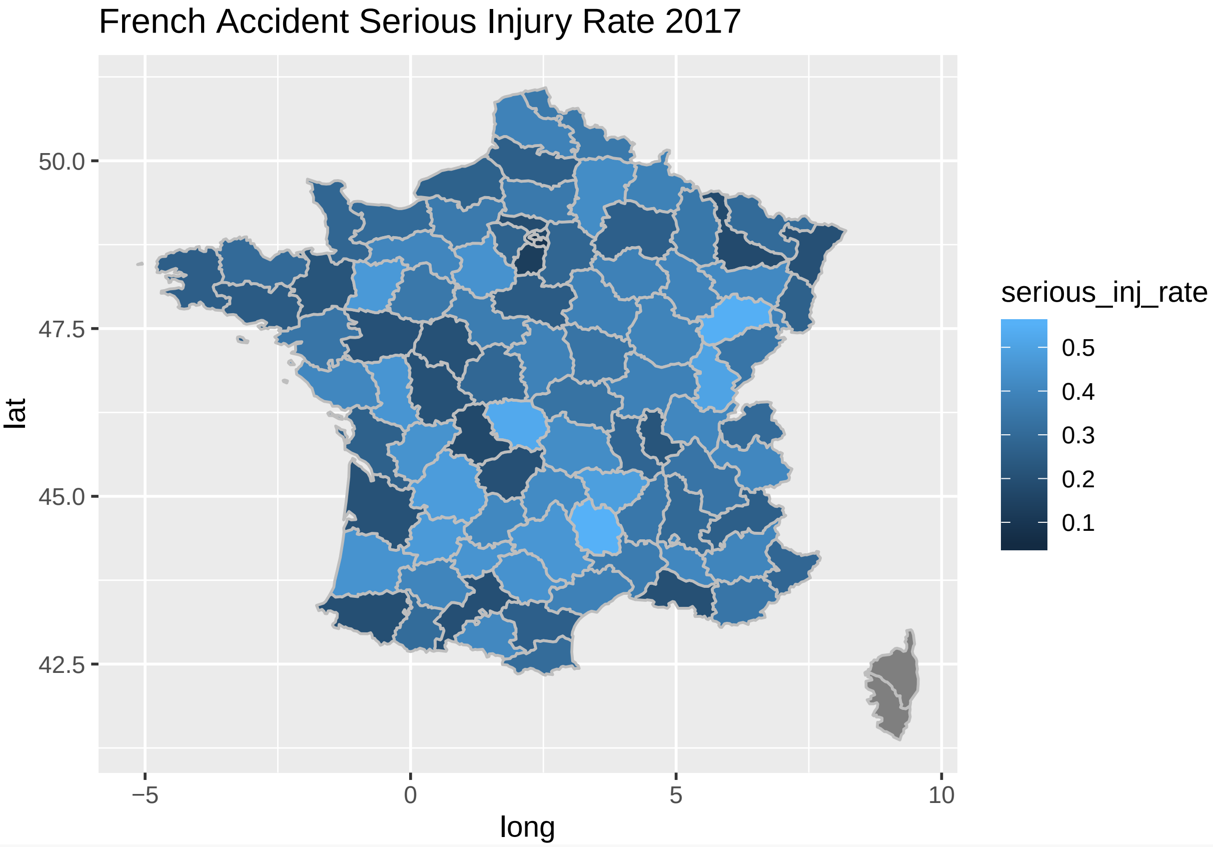
```
france.adm3.shp.df <- broom::tidy(france_shp, region = "code_insee")
france.adm3.shp.df <- france.adm3.shp.df %>% filter(!id %in% c("971","972","973","974","976")) #Exclude French overseas territories
france.adm3.shp.df <- france.adm3.shp.df %>%
  left_join(dep_summary, by = c("id" = "department"))   #Link the data to the shapefile

#Generate the ggplot and then apply the ggplotly function to make the map interactive for Mortality Rate
gg <- ggplot(france.adm3.shp.df, aes(text = depart_name, label = serious_inj_rate, label2 = fatality, label3 = accidents)) +
  geom_polygon(aes(x = long, y = lat, group = group, fill= mortality), colour = "grey") +
  labs(title="French Accident Mortality Rate 2017")
gg #Needed to output for PDF
```



```
ggplotly(gg)  #Enables interactive maps in RMarkdown HTML format
```

The next map shows the serious injury rate by Department.

```
#Generate the ggplot and then apply the ggplotly function to make the map interactive for Serious Injury Rate
gg2 <- ggplot(france.adm3.shp.df, aes(text = depart_name, label = serious_inj_rate, label2 = fatality, label3 = accidents)) +
  geom_polygon(aes(x = long, y = lat, group = group, fill= serious_inj_rate), colour = "grey") +
  labs(title="French Accident Serious Injury Rate 2017")
gg2  #Needed to output for PDF
```



```
ggplotly(gg2) #Enables interactive maps in RMarkdown HTML format
```

6.4 Accident victim profile

A mosaic plot is a square subdivided into rectangular tiles the area of which represents the conditional relative frequency for a cell in the contingency table. Each tile is colored to show the deviation from the expected frequency. You can use the mosaic plot to discover the association between two variables.

- Red tiles indicate significant negative residuals, where the frequency is less than expected
- Blue tiles indicate significant positive residuals, where the frequency is greater than expected (under the null model(independence))

The colours represent the level of the residual for that cell / combination of levels. More specifically, blue means there are more observations in that cell than would be expected under the null model (independence). The intensity of the color represents the magnitude of the residual.

Pearson standardized residuals: The strength of a relation in the mosaic is a measure of how much the observed values deviate from the values in case of independence. It shows the strength and direction of the association for the categorical information.

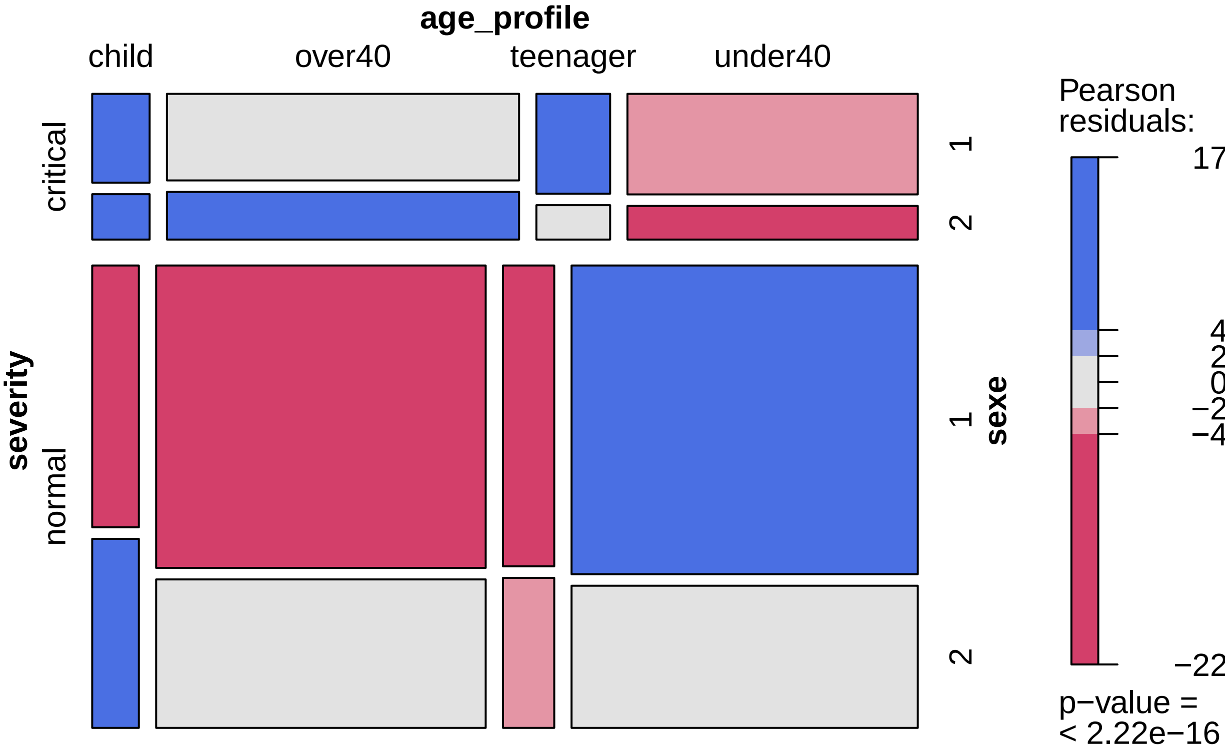
$$r_{ij} = (O_{i,j} - E_{i,j})/\sqrt{E_{i,j}}$$

Where:

$O_{i,j}$ = observed frequency (found in the sample)

$E_{i,j}$ = expected frequency (i = ith row; j = jth column of contingency table)

```
mosaic(~severity + age_profile + sexe, data=accidents_fulldata,
       shade=TRUE, legend=TRUE)
```



So what can be interpreted from the mosaic that has been generated for severity, age profile and gender. Recall the severity of accidents is grouped to critical (fatal or serious injury) and normal response (light hospitalisation, no injury).

This is best interpreted using some specific language. Within the over40 age group there is a significant association between critical severity and women (sexe=2). Also the "Child" age profile shows significant positive residual for both sexes. Male teenagers also have a frequency greater than expected for critical severity.

6.5 Child drivers and Child casualties

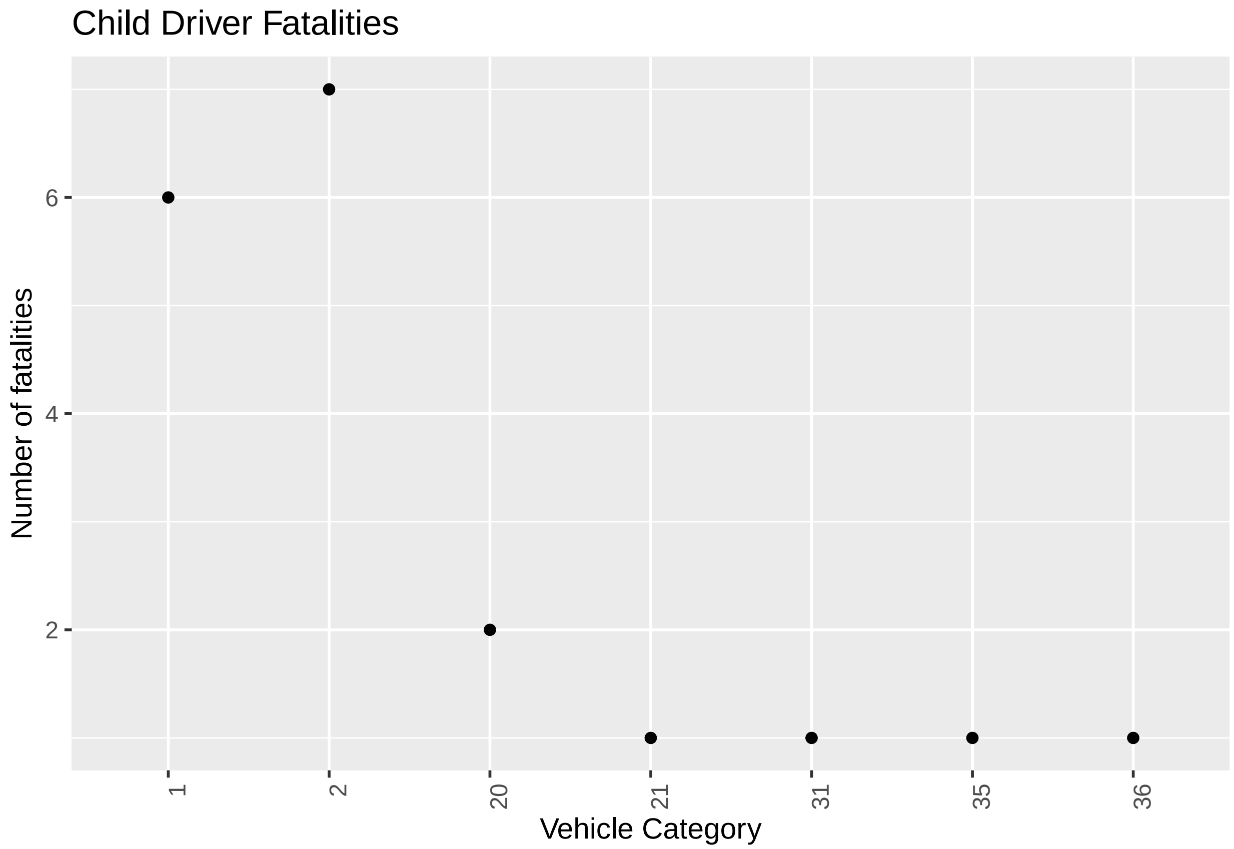
This section will not be included as a feature in the model analysis however has been added in the EDA section for information purposes. An objective of this study is to identify and gain insight through the exploratory analysis of the data.

The first assessment is to identify the main category of vehicle that are driver by children under the age of 15. The objective is to identify the most common and dangerous vehicle so that I can hopefully gain insight and learning that can be applied to my own children to improve their transport safety options.

List the Category of vehicles that children drove and where injured. Top of the list are 01 - Bicyclette and 02 - Cyclomoteur < 50cm3

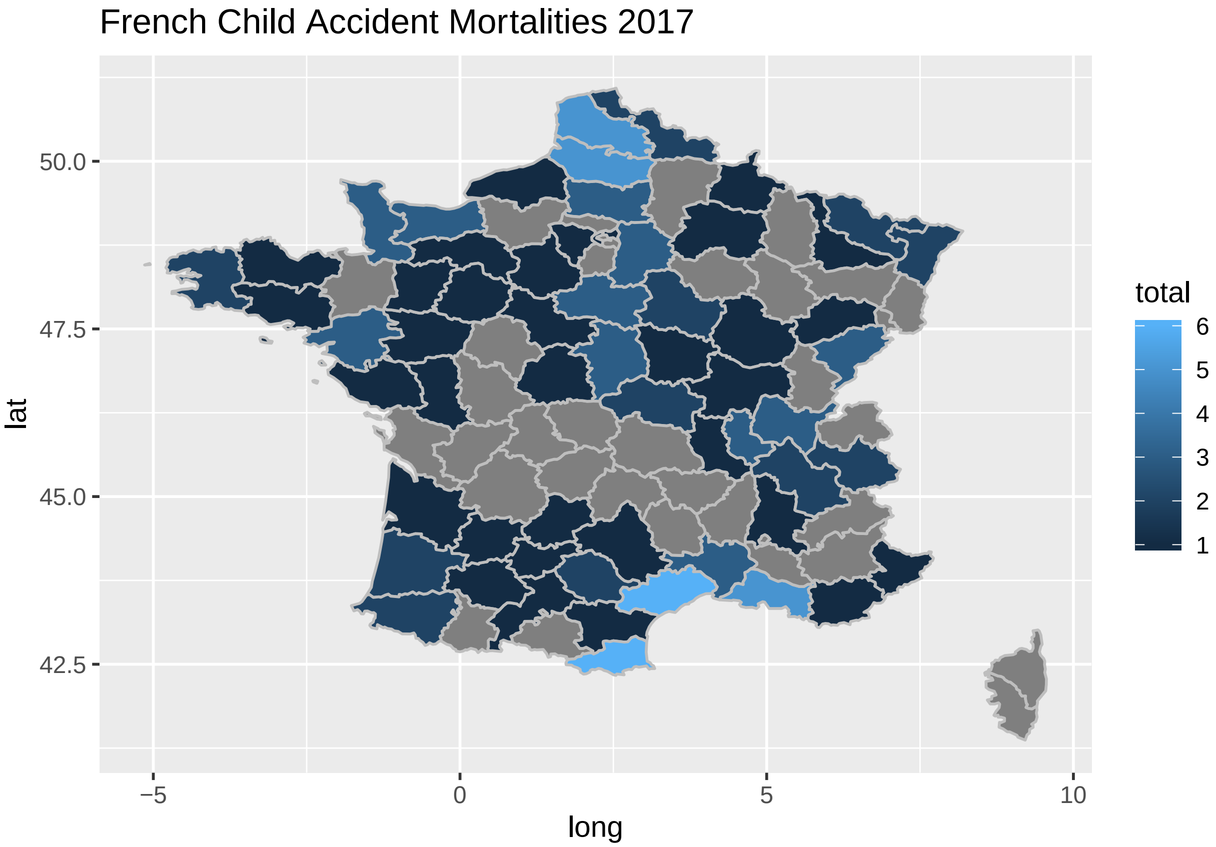
Some other vehciles of interest:

- 20 Special engine
- 21 is an agricultural tractor
- 30 Scooter < 50 cm3
- 31 Motocyclette > 50 cm3 and <= 125cm3
- 35/36 Quad bikes <50cm3 and >50cm3 respectively



The following map of France tragically shows the distribution of child fatalities throughout French departments. Immediately the departments of Pas-De-Calais and the Somme are highlighted where several children have tragically lost their lives. In addition to the obsevation of the vehicle classification and location in this area there is certainly justification for further study of the issue in this area and the possibility of targeted educational training or regionally state intervention.

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/Trevor/DataScience/projects/PH125.9x_CYO/departements-100m.shp", layer: "departement
## with 101 features
## It has 4 fields
```



```
## Exploratory Data Analysis: 7.609 sec elapsed
```

7 Modelling

7.1 Model 1: GLM with stepwise feature determination

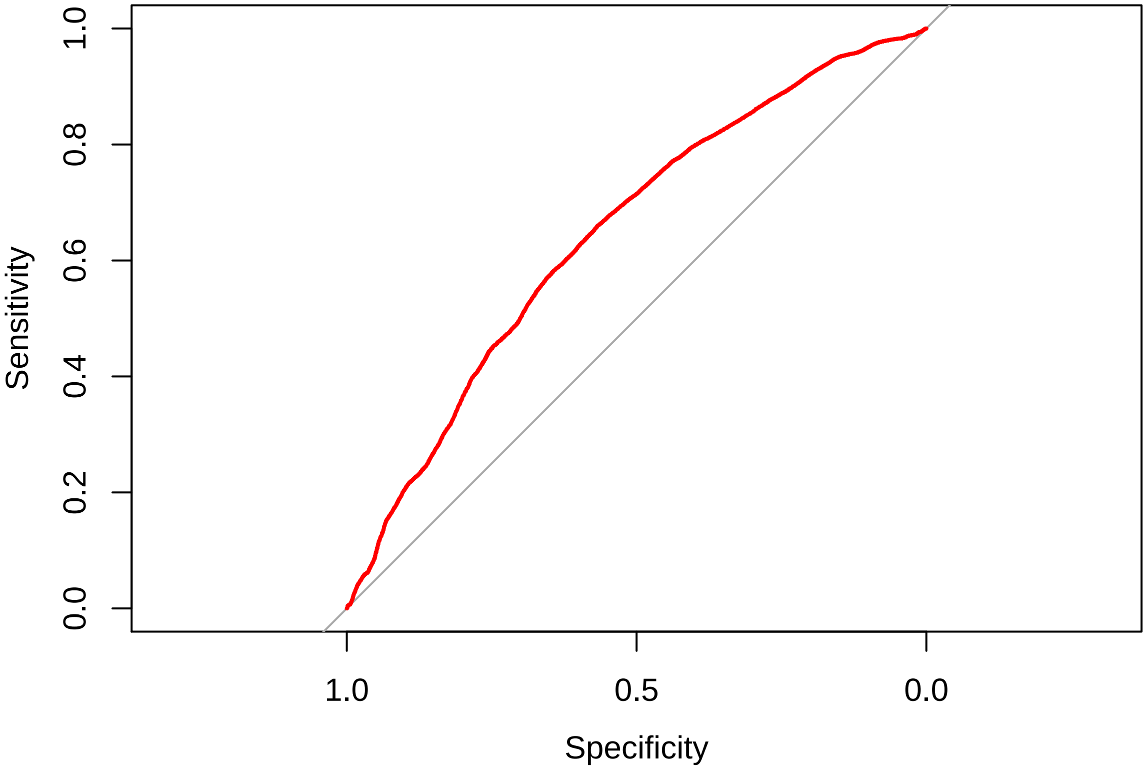
This model uses a stepwise approach to assessing the benefits of features to be added. Predictors are added step by step until no new predictors add any substantial value to the model. It is not guaranteed to find the best possible model. A criticism of the stepwise regression procedure is it can violate some statistical assumptions and result in a model that makes little sense in the real world.

The stepwise model steps from a null model (no feature) and incremently adds the features stepping towards a full model of all features. The stepwise will stop when there is no added value by adding an additional feature.

```
## Start:  AIC=119880.2
## severity ~ 1
##
##                Df Deviance    AIC
## + agg          1   117699 117703
## + catv         1   118759 118763
## + age_profile  3   119330 119338
## + day_night    1   119758 119762
## + atm          1   119776 119780
## + sexe         1   119829 119833
## + mois         1   119863 119867
## + jour         1   119876 119880
## <none>             119878 119880
## + catr         1   119877 119881
##
## Step:  AIC=117702.9
## severity ~ agg
##
##                Df Deviance    AIC
## + catv         1   116373 116379
## + catr         1   116841 116847
## + age_profile  3   117122 117132
## + day_night    1   117595 117601
## + atm          1   117640 117646
## + sexe         1   117651 117657
## + mois         1   117681 117687
## <none>             117699 117703
## + jour         1   117697 117703
##
## Step:  AIC=116379.4
## severity ~ agg + catv
##
##                Df Deviance    AIC
## + catr         1   115546 115554
## + age_profile  3   115762 115774
## + day_night    1   116232 116240
## + atm          1   116303 116311
## + mois         1   116359 116367
## + sexe         1   116370 116378
## <none>             116373 116379
## + jour         1   116372 116380
```

```
##
## Step:  AIC=115554
## severity ~ agg + catv + catr
##
##               Df Deviance    AIC
## + age_profile  3   115001 115015
## + day_night    1   115378 115388
## + atm          1   115484 115494
## + mois         1   115531 115541
## + sexe         1   115543 115553
## <none>             115546 115554
## + jour         1   115545 115555
##
## Step:  AIC=115015.1
## severity ~ agg + catv + catr + age_profile
##
##             Df Deviance    AIC
## + day_night  1   114786 114802
## + atm        1   114943 114959
## + mois       1   114989 115005
## + sexe       1   114994 115010
## <none>           115001 115015
## + jour       1   115000 115016
##
## Step:  AIC=114802.1
## severity ~ agg + catv + catr + age_profile + day_night
##
##        Df Deviance    AIC
## + atm   1   114740 114758
## + mois  1   114781 114799
## + sexe  1   114782 114800
## <none>      114786 114802
## + jour  1   114785 114803
##
## Step:  AIC=114757.6
## severity ~ agg + catv + catr + age_profile + day_night + atm
##
##        Df Deviance    AIC
## + sexe  1   114735 114755
## + mois  1   114736 114756
## <none>      114740 114758
## + jour  1   114738 114758
##
## Step:  AIC=114755.1
## severity ~ agg + catv + catr + age_profile + day_night + atm +
##     sexe
##
##        Df Deviance    AIC
## + mois  1   114731 114753
## <none>      114735 114755
## + jour  1   114734 114756
##
## Step:  AIC=114752.8
## severity ~ agg + catv + catr + age_profile + day_night + atm +
##     sexe + mois
##
##        Df Deviance    AIC
## <none>      114731 114753
## + jour  1   114729 114753
```

Plot the receiver operating characteristic (ROC). The ROC curve plots sensitivity (true positive rate - TPR) versus 1 - specificity or the false positive rate (FPR). We can see that we obtain higher sensitivity with this approach for all values of specificity, which implies it is in fact a better method than guessing.



```
## Area under the curve: 0.6463
```

The GLM model is fit based on the features identified from the stepwise feature approach. The model is fit against the test set. The average severity **0.7603826** is calculated and is compared with the predicted values to determine if the probability of the predicted severity is greater than the average. A confusion matrix is then run to estimate the accuracy of the model prediction against the test set.

```
##
## Call:
## glm(formula = severity ~ agg + catv + age_profile + catr + day_night +
##     sexe + atm, family = "binomial", data = accTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2484   0.4922   0.6245   0.7669   2.2785
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.378194   0.044553   8.489  < 2e-16 ***
## agg                   0.972711   0.017478  55.653  < 2e-16 ***
## catv                 -0.022765   0.000619 -36.775  < 2e-16 ***
## age_profileover40     0.156677   0.029774   5.262 1.42e-07 ***
## age_profileteenager -0.078014   0.037313  -2.091   0.0365 *  
## age_profileunder40   0.439564   0.030178  14.566  < 2e-16 ***
## catr                -0.197378   0.007019 -28.122  < 2e-16 ***
## day_nightnight      -0.221639   0.015614 -14.194  < 2e-16 ***
## sexe                 0.033707   0.015931   2.116   0.0344 *  
## atm                 -0.029400   0.004245  -6.926 4.34e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 119878  on 108816  degrees of freedom
## Residual deviance: 114735  on 108807  degrees of freedom
## AIC: 114755
## 
## Number of Fisher Scoring iterations: 4
```

```
## [1] 0.7603826
```

```
## [1] 0.5799147
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     0     1
##          0  3856  7572
##          1  2645 13131
##                                           
##                Accuracy : 0.6244          
##                  95% CI : (0.6186, 0.6302)
##     No Information Rate : 0.761           
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.1805          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.5931          
##             Specificity : 0.6343          
##          Pos Pred Value : 0.3374          
##          Neg Pred Value : 0.8323          
##              Prevalence : 0.2390          
##          Detection Rate : 0.1417          
##    Detection Prevalence : 0.4201          
##       Balanced Accuracy : 0.6137          
##                                           
##        'Positive' Class : 0               
## 
```

The model enables a prediction of the severity based on the features. Department and Commune have been removed to simplify the illustration. A phone call is received to the emergency service who then ask a series of questions. The parameters can be adjusted in order to generate a new prediction.

```
enter_prediction <- data.frame(day_night = "night",         # Day or night
                               age_profile = "under40",    # Drivers age, if known
                               agg = 1,                   # Built up area or countryside?
                               atm = 1,                  # Weather conditions
                               catr = 1,                 # Highway, national route, department road?
                               catv = 7,                # Type of vehicle
                               sexe = 2)                 # Sex of the driver

result <- predict(fit, newdata = enter_prediction, type = "response")
ifelse((1 - result) > mean(accidents_to_model$severity), "Critical Alert", "Normal Responders")
```

```
##                    1
## "Normal Responders"
```

```
predict(fit, newdata = enter_prediction, type = "terms")
```

```
##          agg      catv age_profile      catr  day_night       sexe
## 1 -0.5906419 0.1084616   0.1882132 0.4385359 -0.1504813 0.02281599
##          atm
## 1 0.01718017
## attr(,"constant")
## [1] 1.216025
```

The prediction for this model is to send **Normal Responders** to the accident.

```
## Model 1: GLM with stepwise feature determination: 13.74 sec elapsed
```

7.2 Model 2: RPart classification

Build an Accident Severity model predicting accident severity outcome versus the predictors/features using the RPart machine learning method. Use the same features that were applied in model 1.

The following function creates entries in a tibble that contains all the results of each model assessed. It will be printed in the results section of this report.

```
#Calculate the average rating for each movie and calculate the bias (difference) for each
add_results <- function(n_accuracy, n_sens, n_spec, n_F1, n_ppv, n_npv, model_method){
  bind_rows(model_results,
            tibble(method=model_method,
                   model_accuracy = n_accuracy,
                   model_sens = n_sens,
                   model_spec = n_spec,
                   model_F1 = n_F1,
                   model_ppv = n_ppv,
                   model_npv = n_npv))
}
```

```
set.seed(1976)
accidents_to_model <- accidents_fulldata %>%
  select(severity, agg, catv, age_profile, catr, day_night, sexe, atm)

#Create the training and test sets
indexes <- createDataPartition(accidents_to_model$severity, times = 1, p = 0.8, list = FALSE)
accTrain <- accidents_to_model[indexes, ]
accTest <- accidents_to_model[-indexes, ]
#Fit the model using rpart()
modelFit <- rpart(as.factor(severity) ~ .,data=accTrain, method = "class", control = rpart.control(cp = 0))
rpart_prediction <- predict(modelFit, accTest, type = "class")
# Use a confusion matrix to tabulate each combination of prediction and actual value.
confmatrix2 <- confusionMatrix(rpart_prediction, as.factor(accTest$severity))
confmatrix2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction critical normal
##   critical     1234    737
##   normal       5284  19948
##
##                Accuracy : 0.7787
##                  95% CI : (0.7737, 0.7836)
##     No Information Rate : 0.7604
##     P-Value [Acc > NIR] : 5.593e-13
##
##                   Kappa : 0.2019
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.18932
##             Specificity : 0.96437
##          Pos Pred Value : 0.62608
##          Neg Pred Value : 0.79058
##              Prevalence : 0.23961
##          Detection Rate : 0.04536
##    Detection Prevalence : 0.07246
##       Balanced Accuracy : 0.57685
##
##        'Positive' Class : critical
##
```

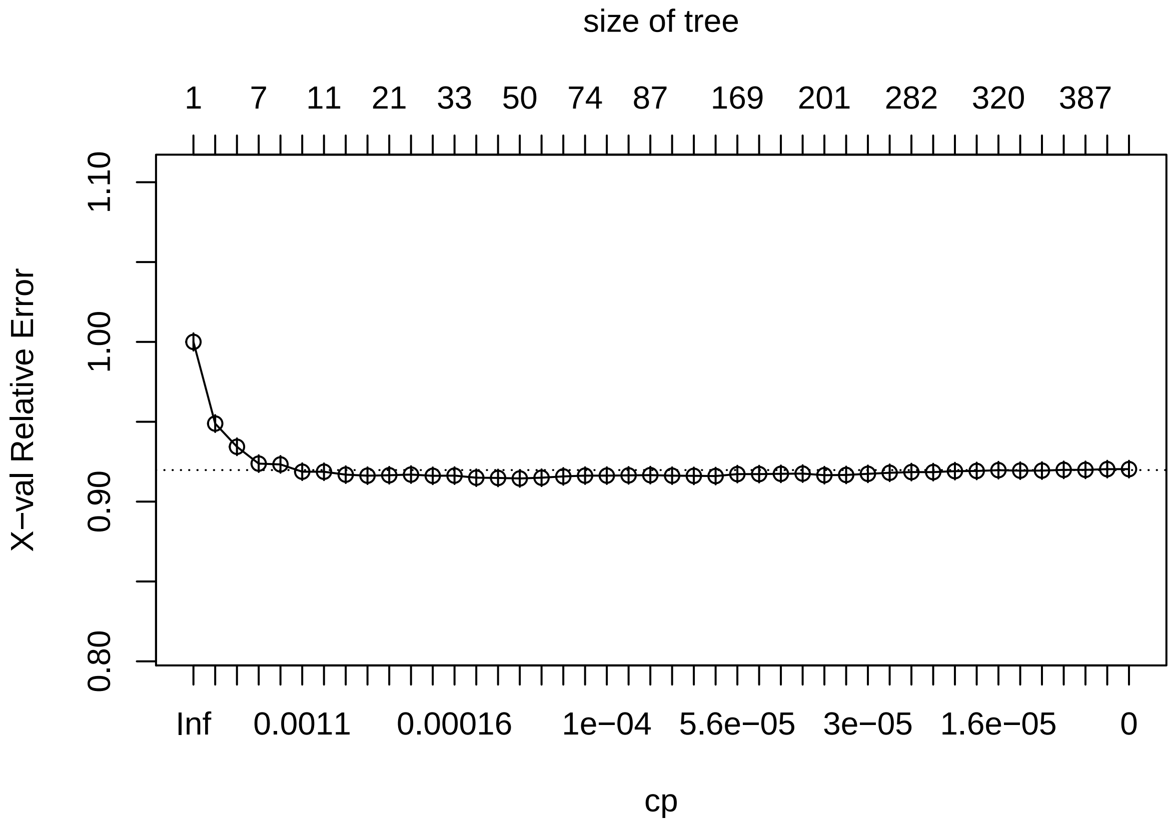
```
model_results <- add_results(confmatrix2$overall['Accuracy'],
                             confmatrix2$byClass['Sensitivity'],
                             confmatrix2$byClass['Specificity'],
                             confmatrix2$byClass['F1'],
                             confmatrix2$byClass['Pos Pred Value'],
                             confmatrix2$byClass['Neg Pred Value'],
                             "Rpart")
```

Model a phone call received to the emergency service who then ask a series of questions. The parameters can be adjusted in order to generate a new prediction.

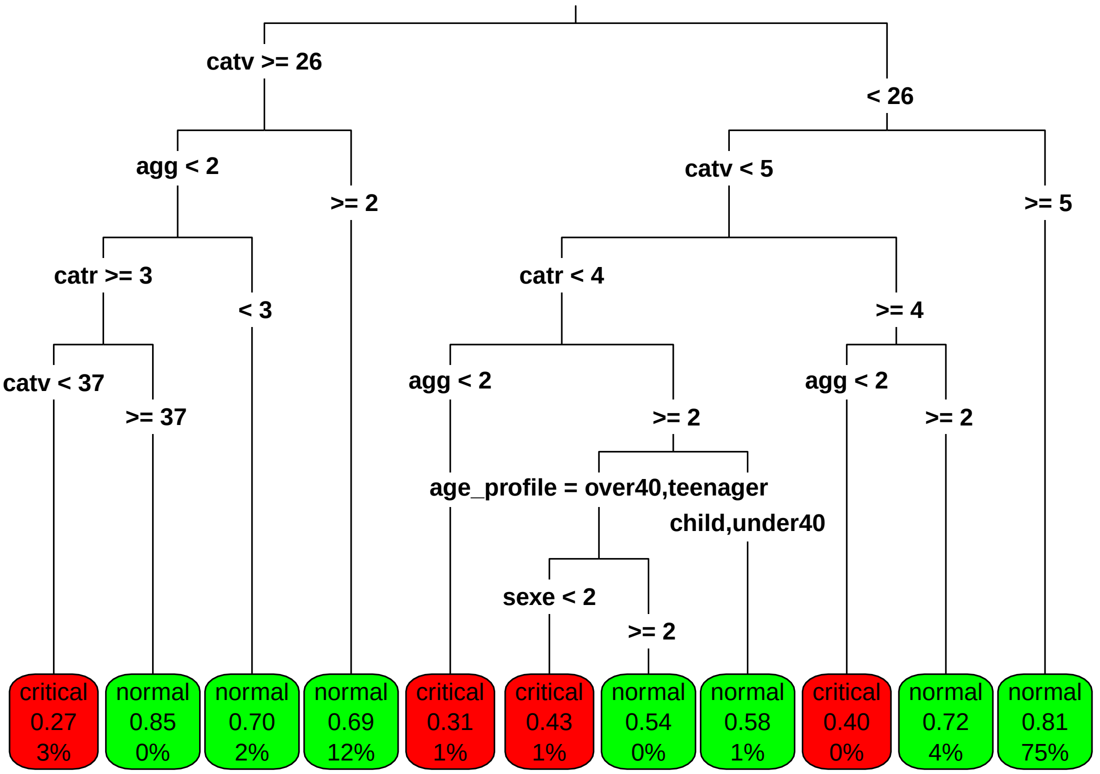
```
enter_rpart_pred <- data.frame(day_night = "night",        # Day or night
                               age_profile = "under40",   # Drivers age, if known
                               agg = 1,                   # Built up area or countryside?
                               atm = 1,                   # Weather conditions
                               catr = 1,                  # Highway, national route, department road?
                               catv = 7,                 # Type of vehicle
                               sexe = 2)                  # Sex of the driver

result <- predict(modelFit, enter_rpart_pred, type = "class")
result
```

```
##      1
## normal
## Levels: critical normal
```

```
plotcp(modelFit)
```



```
rpart.plot(prune(modelFit, cp = 0.00045), type = 3, box.palette = c("red", "green"), fallen.leaves = TRUE)
```



The prediction for this model is to send **normal** responders to the accident.

```
## Model 2: RPart Classification: 2.85 sec elapsed
```

8 Results Section

The results of both models listed as follows:

```
model_results %>%
  kable() %>%
  kable_styling("striped", full_width = F)
```

method	model_accuracy	model_sens	model_spec	model_F1	model_ppv	model_npv
GLM	0.6244302	0.5931395	0.6342559	0.4301411	0.3374169	0.8323403
Rpart	0.7786641	0.1893219	0.9643703	0.2907292	0.6260781	0.7905834

A principal component analysis has also been included in the R code submitted. However as the results of the PCA were not satisfactory they have not been includded in the final report. Two models are compared for this project submission.

9 Conclusion

This section contains the summary of key main findings from the Choose Your Own Capstone project. The course was an excellent introduction to Data Science and R and this project in particular was an interesting topic to study.

The accuracy of the RPart model is signficantly higher than the GLM method and therefore appears to be the better model. However it is important to consider the other results of the confusion matrices of the models before making a decision. The sensitivity (true positive rate) differs considerably. The GLM has a much higher sensitivity than the RPart model infering that the prediction of accidents being severe/critical is higher. There are less accidents predicted as normal severity when in fact the actual accident was severe.

Another comparison that helps compare the models is the F1 Score. This is the harmonic average of the model precision (positive prediction value) and recall (sensitivity).

$$F_1 Score = (Precision * Recall)/(Precision * Recall)$$

Where:

$Precision$ = true_positives / (true_positives + false_positives)

$Recall$ = true_positives / (true_positives + false_negatives)

The F1 score of the GLM model is significantly better performing than the RPart.

In conclusion, the GLM model is selected as the more appropriate model to be used based on a comparison of the models. However the results of both models are not satisfactory. The F1 scores do not provide confidence in the models, in addition to the overal accuracy. The results of the models show that further improvement of the models is necessary.

For a future study more machine learning methods should be selected to provide a comparison of a greater number of algorithms. Also further dummifcation of the categorical data may improve the performance of the general logistic regression and classification algorithms.

Thank you for reading and assessing this submission for the HarvardX PH125.9x Professional Certificate in Data Science capstone project. The course was an excellent introduction to Data Science and R Programming and I would like to greatly thank Professor Rafael A. Irizarry, the course staff team, edx platform teams and the other course students who have provided great solutions and advice in the course discussion forums.

10 References

Please note that the references had to be added manually as it was not an option to include a .bib file. It is also possible to add a reference inline in the YAML header though this was not preferred.

European Commission. April 2018. *Road Safety in the European Union – Trends, statistics and main challenges.* Publications Office of the European Union, ISBN 978-92-79-80281-2. https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/vademecum_2018.pdf.

Chinnamgari, Sunil. 2019. *R Machine Learning Projects Implement Supervised, Unsupervised, and Reinforcement Learning Techniques Using R 3. 5.* Birmingham: Packt Publishing Ltd.

Hodnett, Mark. 2018. *R Deep Learning Essentials : A Step-by-Step Guide to Building Deep Learning Models Using Tensorflow, Keras, and Mxnet.* Birmingham, UK: Packt Publishing.

Irizzarry, Rafael A. Prof. 2019. *Introduction to Data Science.* https://rafalab.github.io/dsbook/large-datasets.html#recommendation-systems.

Voulgaris, Zacharias. 2017. *Data Science Mindset, Methodologies, and Misconceptions.* City: Technics Pubns Llc.

Wickham, Hadley, and Garrett Grolemund. 2017. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.* 1st ed. O'Reilly Media, Inc.

Viswanathanm, Shanth, Viswanathan, Viswa, Yu-Wei,Chiu and Gohil, Atmajitsinh.*R: Recipes for Analysis, Visualization and Machine Learning.* Published by Packt Publishing, 2016