Projet : Analyser un jeu de données à travers l'univers tidyverse

UE Analyse de données, Master 1 Mathématiques et Applications spécialité Ingenierie Mathématique pour les Sciences du Vivant

Nadia Ouhssaine

11 janvier 2019

Choix du jeu de données

Le jeu de données concerne les accidents de la circulation de l'année 2017 en France. J'ai choisis ce jeu de données car il comporte plusieurs variables repartis sur plusieurs tableau, formant ainsi des données relationnelles. Ces dernières me permettront de travailler sur l'ensemble des fonctionnalités de `tidyverse`. Par ailleurs, chaque année en france, des accidents de la route font des milliers de morts, ce jeu de données va donc nous permettre d'y voir un peu plus clair à ce sujet et peut être définir les causes de ses nombreux accidents.

Je charge mon jeu de données à l'aide de la fonction `read.csv`. Par ailleurs, je vais d'ores et déjà convertir les dataframes obtenu en tibble à l'aide de la fonction `as.tibble` :

```
(caracteristics<-as.tibble(read.csv("caracteristiques-2017.csv")))
```

```
## # A tibble: 60,701 x 16
##    Num_Acc    an  mois  jour  hrmn   lum   agg   int   atm   col   com
##      <dbl> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
##  1 2.02e11    17     1    11  1820     5     2     1     1     1   477
##  2 2.02e11    17     2    13  1630     1     2     3     1     3     5
##  3 2.02e11    17     3     7  1150     1     2     9     1     5    52
##  4 2.02e11    17     4    22  1300     1     2     1     1     6     5
##  5 2.02e11    17     5    20  1230     1     2     1     1     2    11
##  6 2.02e11    17     5    30  1400     1     1     1     1     7     5
##  7 2.02e11    17     6    30  2140     1     2     1     1     3   477
##  8 2.02e11    17     7     7  1740     1     2     1     1     1   477
##  9 2.02e11    17    12    21   830     1     1     1     2     6     5
## 10 2.02e11    17     6    29   620     2     1     6     1     7   250
## # ... with 60,691 more rows, and 5 more variables: adr <fct>, gps <fct>,
## #   lat <int>, long <int>, dep <int>
```

```
(places<-as.tibble(read.csv("lieux-2017.csv")))
```

```
## # A tibble: 60,701 x 18
##    Num_Acc  catr voie     v1 v2     circ   nbv    pr   pr1  vosp  prof
##      <dbl> <int> <fct> <int> <fct> <int> <int> <dbl> <int> <int> <int>
##  1 2.02e11     3 39       NA ""        2     2    NA    NA     2     1
##  2 2.02e11     3 39       NA ""        2     2    NA    NA     0     1
##  3 2.02e11     3 39       NA ""        2     2    NA    NA     0     1
##  4 2.02e11     3 39       NA D         2     2    NA    NA     0     1
##  5 2.02e11     3 39       NA ""        2     2    NA    NA     0     1
##  6 2.02e11     3 41       NA C         2     2    NA    NA     0     1
##  7 2.02e11     3 39       NA ""        2     2    NA    NA     0     1
##  8 2.02e11     4 0        NA ""        2     2    NA    NA     0     1
##  9 2.02e11     3 41       NA C         2     2    NA    NA     0     1
```

```
## 10 2.02e11     2 41       NA ""        3     4    NA    NA     0     1
## # ... with 60,691 more rows, and 7 more variables: plan <int>,
## #   lartpc <int>, larrout <int>, surf <int>, infra <int>, situ <int>,
## #   env1 <int>
```

```
(users<-as.tibble(read.csv("usagers-2017.csv")))
```

```
## # A tibble: 136,021 x 12
##    Num_Acc place  catu  grav  sexe trajet  secu  locp  actp etatp an_nais
##      <dbl> <int> <int> <int> <int>  <int> <int> <int> <int> <int>   <int>
##  1 2.02e11     1     1     3     1      9    13     0     0     0    1968
##  2 2.02e11     2     2     3     2      9    11     0     0     0    1973
##  3 2.02e11     1     1     3     1      1    13     0     0     0    1967
##  4 2.02e11     1     1     1     1      0    11     0     0     0    1953
##  5 2.02e11     1     1     3     1      5    22     0     0     0    1960
##  6 2.02e11     1     1     1     1      1    11     0     0     0    1973
##  7 2.02e11     1     1     3     1      1    11     0     0     0    1938
##  8 2.02e11     1     1     1     1      9    11     0     0     0    1992
##  9 2.02e11     7     2     1     2      1    11     0     0     0    2015
## 10 2.02e11     1     1     3     2      1    11     0     0     0    1960
## # ... with 136,011 more rows, and 1 more variable: num_veh <fct>
```

```
(vehicles<-as.tibble(read.csv("vehicules-2017.csv")))
```

```
## # A tibble: 103,546 x 9
##         Num_Acc  senc  catv occutc   obs  obsm  choc  manv num_veh
##           <dbl> <int> <int>  <int> <int> <int> <int> <int> <fct>
##  1 201700000001     0     7      0     0     2     3     9 B01
##  2 201700000001     0    10      0     0     2     3    13 A01
##  3 201700000002     0     7      0     0     0     1    16 A01
##  4 201700000002     0     1      0     0     0     7     1 B01
##  5 201700000003     0    10      0     0     2     1     1 C01
##  6 201700000003     0     7      0     0     2     3    13 A01
##  7 201700000003     0     7      0     0     2     6     1 B01
##  8 201700000004     0     7      0     6     0     1     1 A01
##  9 201700000005     0    33      0     1     2     1     1 B01
## 10 201700000005     0     7      0     0     2     5    19 A01
## # ... with 103,536 more rows
```

Nettoyer et ranger les données

Pour une analyse efficace d'un jeu de données, il faut au préalable que les données soit bien nettoyées et rangées. Ainsi, la première partie de l'analyse consiste au nettoyage de l'ensemble des fichiers du jeu de données. Ce nettoyage consistera principalement à détecter les variables où la majorité des valeurs sont manquantes afin de ne pas les prendre en compte dans notre analyse de données car elles pourraient la perturber.

Valeurs manquantes :

Dans ce jeu de données, les valeurs manquantes dans les variables catégorielles prennent parfois la valeur "0" ou sont vides. Pour une détection plus simple des valeurs manquantes, nous allons attribuer à tous ces types de valeurs manquantes la valeur `NA`.

Remarque : J'ai préalablement vérifié qu'aucun 0 n'avait autre sens que NA, dans ce jeu de données 0 à toujours le sens de "valeurs manquantes"

```
# Fonction remplaçant toute les valeurs 0 et vide par NA.
valmanq<-function(x){
  x[x[,]==0 | x[,]==""]<-NA
  return(x)
}
# Application de la fonction à tous les tableaux de données
caracteristics<-valmanq(caracteristics)
places<-valmanq(places)
users<-valmanq(users)
vehicles<-valmanq(vehicles)
```

Nous allons maintenant afficher le pourcentage de valeurs manquantes pour chaque variable à l'aide de la fonction `apply`.

```
# Fonction qui applique à chaque colonne le calcul :
# (nb valeurs manquantes / nb valeurs )*100
# ceci correspond à la proportion de valeurs manquantes pour chaque variable
prop_valmanq<-function(x){
  nb_obs<-nrow(x)
  apply(x,2,function(x) ( sum(is.na(x))/nb_obs)*100 )
}
# Application de la fonction pour chaque tableau de données
prop_valmanq(caracteristics)
```

```
##      Num_Acc           an         mois         jour         hrmn
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
##          lum          agg          int          atm          col
##  0.000000000  0.000000000  0.001647419  0.021416451  0.009884516
##          com          adr          gps          lat         long
##  0.000000000  1.354178679  7.591308216 12.736198745 12.736198745
##          dep
##  0.000000000
```

```
prop_valmanq(places)
```

```
##   Num_Acc      catr      voie        v1        v2      circ       nbv
##  0.000000  0.000000 53.303899 99.327853 95.570089  4.947200  6.828553
##        pr       pr1      vosp      prof      plan    lartpc   larrout
## 61.837532 66.201545 93.790877  3.462875 13.986590 81.428642 40.389779
##      surf     infra      situ      env1
```

```
##  3.703399 90.444968 11.355661 34.933527
```

```
prop_valmanq(users)
```

```
##     Num_Acc       place        catu        grav        sexe      trajet
##  0.00000000  8.67660141  0.00000000  0.00000000  0.00000000 21.62460208
##        secu        locp        actp       etatp     an_nais     num_veh
##  6.57986634 92.07401798 91.72113130 91.71083877  0.02720168  0.00000000
```

```
prop_valmanq(vehicles)
```

```
##   Num_Acc      senc      catv    occutc       obs      obsm      choc
##  0.000000 18.078921  0.000000 99.441794 85.661445 17.890599  6.506287
##      manv   num_veh
##  7.370637  0.000000
```

On observe ainsi un bon nombre de variables ayant une majorité de valeurs manquantes. Les variables ayant une trop grande quantité ne peuvent malheureusement pas être pris en compte dans l'analyse car elles pourraient la fausser.

Autres corrections :

Lorsque je souhaite afficher une valeur de `Num_Acc`, il est arrondis comme ceci :

```
# A titre d'exemple
caracteristics$Num_Acc[1]
```

```
## [1] 2.017e+11
```

Ainsi, avec l'arrondis, tous les `Num_Acc` sont égaux et cela peut être problèmatique si nous voulions l'afficher. Cet arrondis est dû au fait que le type de la variable n'est pas correct, comme un bon nombre d'autres variables. Elle est dite numérique car la machine "pense" qu'il s'agit d'une quantité, elle ne sait pas qu'il s'agit en réalité d'un identifiant. Avec la fonction `format` qui enlève l'écriture scientifique, le type est aussi modifié en character.

```
# Fonction qui l'enlève l'écriture scientifique de Num_Acc
corr_format<-function(x){
  x$Num_Acc<-format(x$Num_Acc, scientific=FALSE)
  return(x)
}

# Application de la fonction à tous les tableaux de données
caracteristics<-corr_format(caracteristics)
places<-corr_format(places)
users<-corr_format(users)
vehicles<-corr_format(vehicles)

#A titre d'exemple
caracteristics$Num_Acc[1]
```

```
## [1] "201700000001"
```

La variable `an` dans le tibble `caracteristics` comporte l'année de l'accident sous la forme "yy" (exemple: 2016 -> 16). Nous allons la modifier en ajoutant 2000 afin qu'elle ai la forme complète.

```
caracteristics$an<-2000+caracteristics$an
# A titre d'exemple
caracteristics$an[1]
```

```
## [1] 2017
```

La variable `dep` contient le code de département suivi d'un 0, nous allons supprimer ce 0 afin d'avoir le véritable numéro de département à 2 caractères. Par ailleurs, la variable `com` contient le numéro de commune, code à 3 chiffre donnée par l'Insee. Cependant certains code n'ont pas 3 caractères comme "002" qui est ici "2". Il faut donc rajouter des 0 devant les valeurs comportant moins de 3 caractères. Les variables `dep` et `com` correctement établis nous serviront lorsque l'on voudra les rassembler former les codes Insee.

```
caracteristics$com<-sprintf("%03d",caracteristics$com)
caracteristics$dep<-caracteristics$dep%/%10
```

Nous avons le même problème avec les heures qui devraient normalement être commposé de 4 caractères. Par exemple 8h30 est représenté par 830 alors que cela devrait être 0830. Il sera important, pour la suite, de correctement réécrire cette variable.

```
caracteristics$hrmn<-as.integer(sprintf("%04s",caracteristics$hrmn))
```

Par ailleurs, comme nous l'avons remarqué précédement, certains types ne sont pas correctement détecter par la machine lors du chargement des données. Nous allons donc effectuer une correction des types à l'aide de la fonction `parse_*()`. Celle-ci va également nous permettre à l'aide de `levels` de détecter les valeurs inatendus s'il y en a. Ces modifications sont notamment nécessaire pour le tracer de graphique.

caracteristics :

```
caracteristics$lum<-parse_factor(caracteristics$lum,levels=1:5)
caracteristics$agg<-parse_factor(caracteristics$agg,levels=1:2)
caracteristics$int<-parse_factor(caracteristics$int,levels=1:9)
caracteristics$atm<-parse_factor(caracteristics$atm,levels=1:9)
caracteristics$col<-parse_factor(caracteristics$col,levels=1:7)
```

places :

```
places$catr<-parse_factor(places$catr,levels=c(1:6,9))
places$circ<-parse_factor(places$circ,levels=1:4)
places$vosp<-parse_factor(places$vosp,levels=1:3)
places$prof<-parse_factor(places$prof,levels=1:4)
places$plan<-parse_factor(places$plan,levels=1:4)
places$surf<-parse_factor(places$surf,levels=1:9)
places$infra<-parse_factor(places$infra,levels=1:7)
places$situ<-parse_factor(places$situ,levels=1:5)
```

vehicles:

```
vehicles$catv<-parse_factor(vehicles$catv,levels=c(1:21,30:40,99))
vehicles$senc<-parse_factor(vehicles$senc,levels=1:2)
vehicles$obs<-parse_factor(vehicles$obs,levels=1:16)
vehicles$obsm<-parse_factor(vehicles$obsm,levels=c(1:6,9))
vehicles$choc<-parse_factor(vehicles$choc,levels=1:9)
vehicles$manv<-parse_factor(vehicles$manv,levels=1:24)
```

Nous n'avons eu aucun message d'erreur pour l'instant, cela veut dire qu'aucune valeurs inattendu n'est présente et les variables sont correctement converti au type `factor`.

users:

```
users$place<-parse_factor(users$place,levels=1:9)
users$catu<-parse_factor(users$catu,levels=1:4)
users$grav<-parse_factor(users$grav,levels=1:4)
users$sexe<-parse_factor(users$sexe,levels=1:2)
users$trajet<-parse_factor(users$trajet,levels=c(1:5,9))
users$secu<-parse_factor(users$secu,levels=c(11,12,13,21,22,23,31,32,33,41,42,43,91,92,93))
```

```
## Warning: 30 parsing failures.
## row # A tibble: 5 x 4 col     row   col expected           actual expected   <int> <int> <chr>
## ... ................. ... ....................................... ........ ......................
## See problems(...) for more details.
```

```
users$locp<-parse_factor(users$locp,levels=1:8)
users$actp<-parse_factor(users$actp,levels=c(1:6,9))
users$etatp<-parse_factor(users$etatp,levels=1:3)
```

Un message d'erreur s'affiche, il constate des valeurs inattendus sur la variable `secu`. En regardant de plus près grâce au numéro de ligne donnée, je constate qu'en réalité il n'a detecté comme valeurs inattendu que les valeurs manquantes. Ainsi, aucune valeur inattendu n'est réellement à déplorer.

Nous voilà enfin prêt à travailler sur notre jeu de données qui est ordonné et ne comporte plus d'éléments qui pourraient perturber notre analyse.

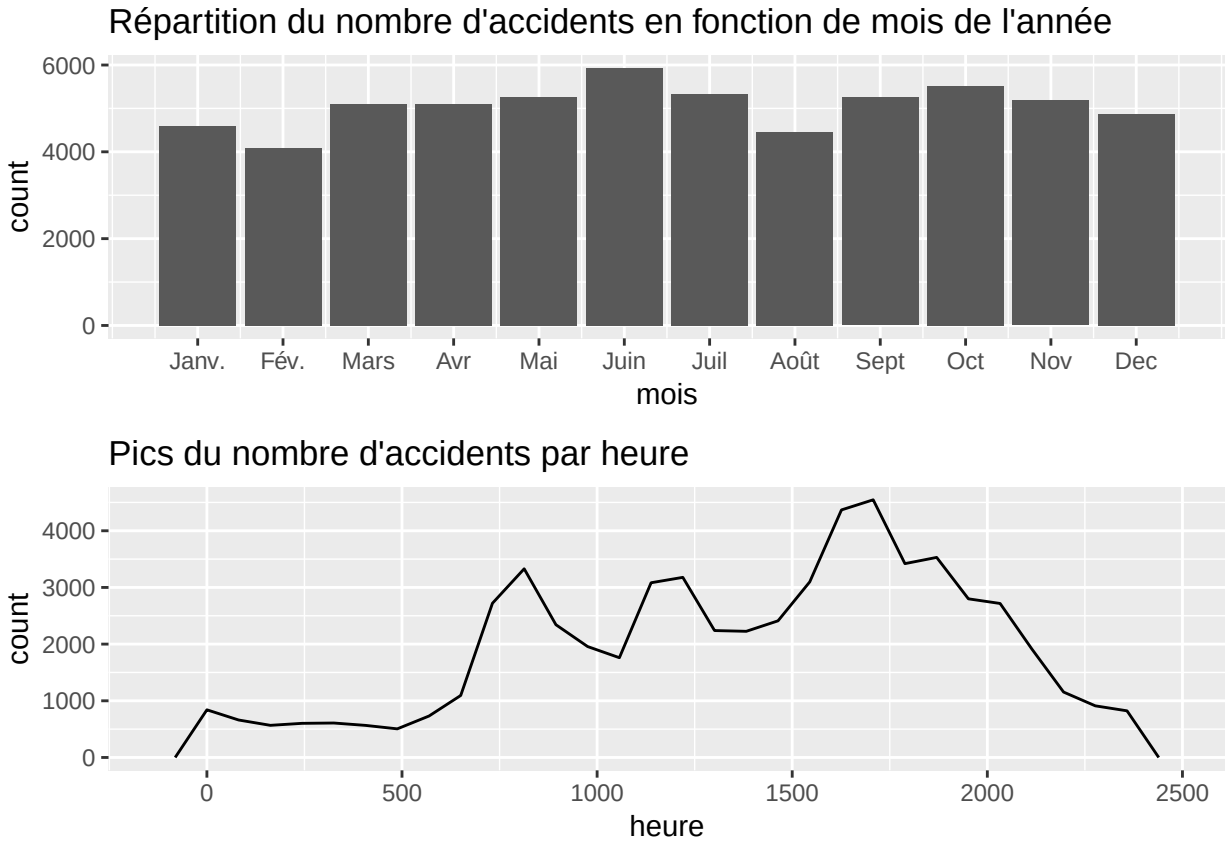
Exploration basée sur le moment des accidents

Voici quelques questions que l'on pourrait se poser :

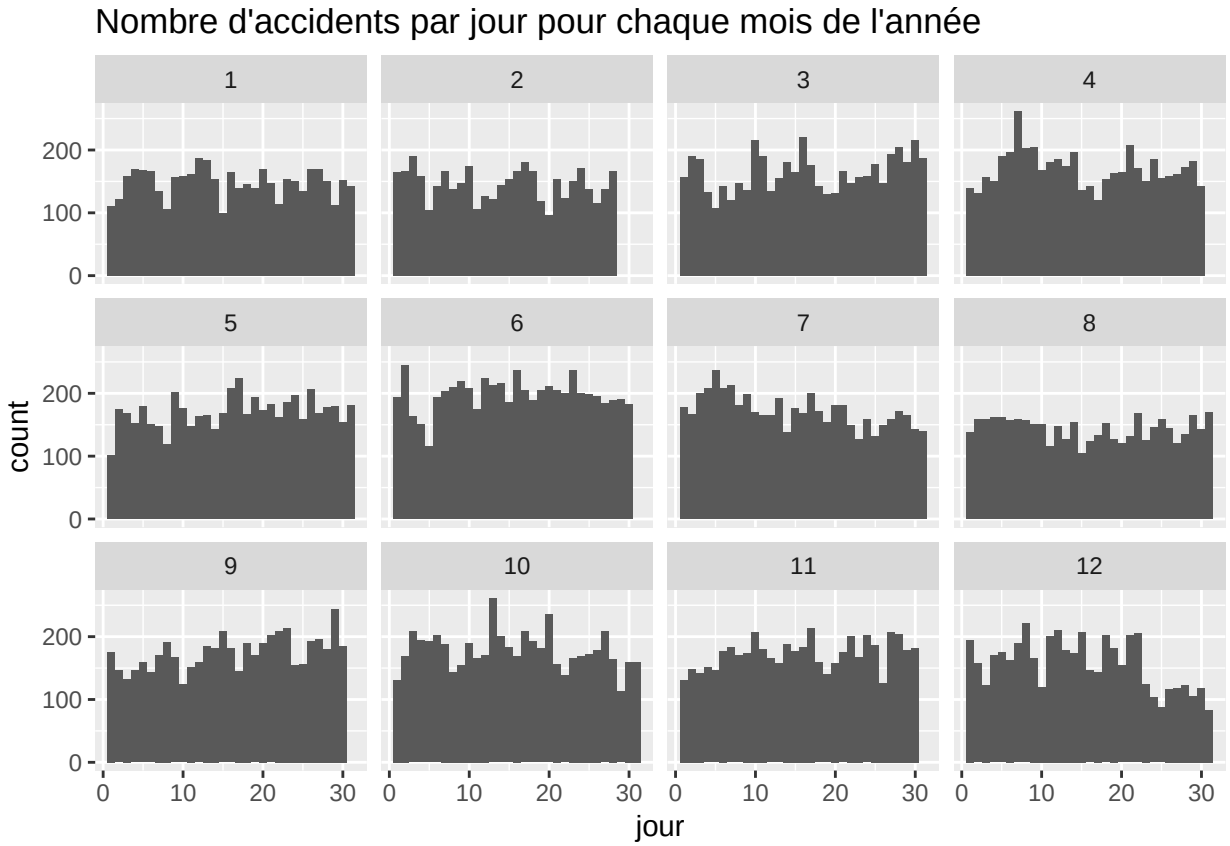
- Quels mois de l'année ont la fréquence la plus élevée d'accidents?
- Quel jour du mois est le plus sûr pour conduire?
- A quelles heures y a t-il le plus d'accident ?

Graphe sur le nombre d'accidents par mois par heure et par jour en fonction des mois :

```
plot1_1<-ggplot(caracteristics)+
geom_bar(aes(x=mois))+
scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10,11,12),
  labels=c("Janv.","Fév.","Mars","Avr","Mai","Juin","Juil","Août","Sept","Oct","Nov","Dec"))+
  labs(title= "Répartition du nombre d'accidents en fonction de mois de l'année")
plot1_2<-ggplot(caracteristics)+
  geom_freqpoly(aes(x=hrmn,y=..count..))+
  labs(title= "Pics du nombre d'accidents par heure",
       x="heure")
#condense les deux graphes afin qu'il ne prennent pas bcp de place
grid.arrange(plot1_1,plot1_2,nrow=2)
```




```
ggplot(caracteristics)+
geom_bar(aes(x=jour))+
facet_wrap(~mois)+
labs(title="Nombre d'accidents par jour pour chaque mois de l'année")
```



Observations :

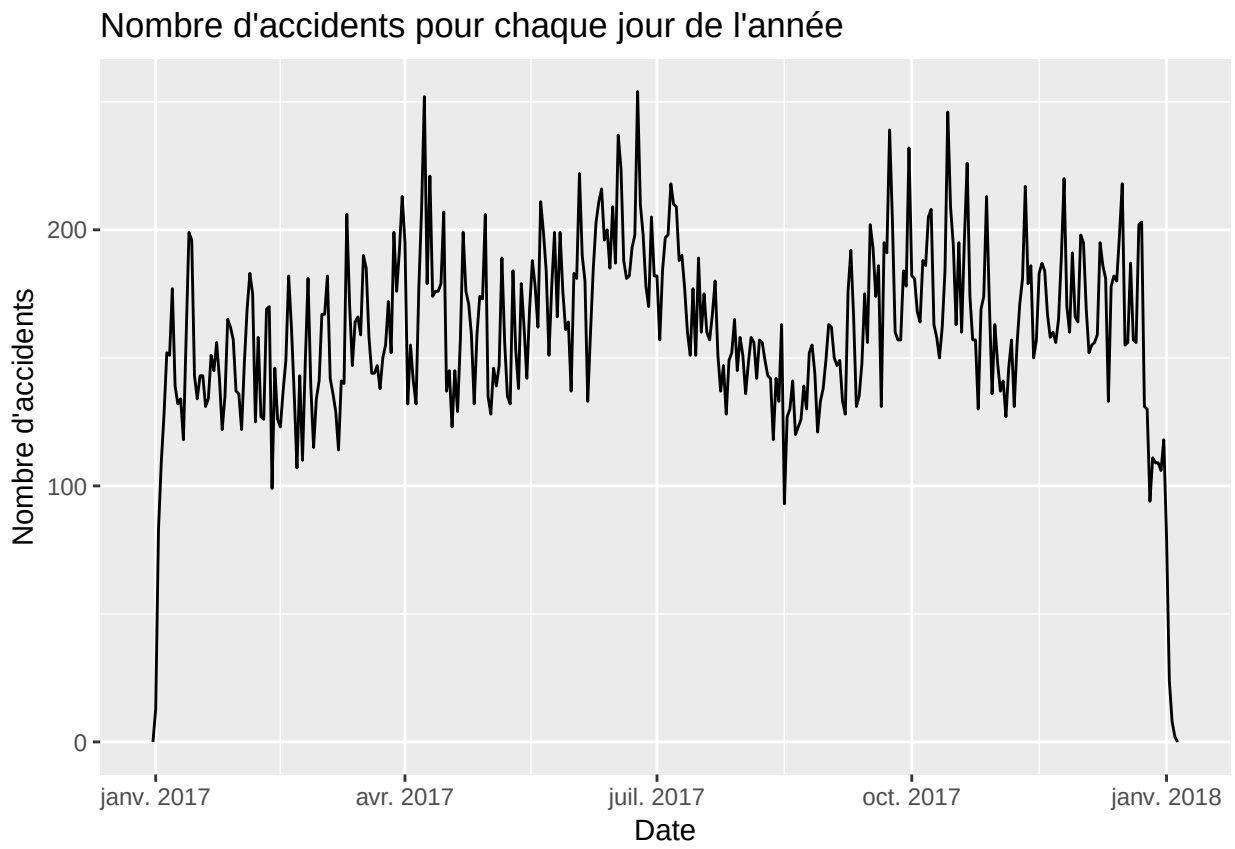
On constate qu'il n'y a pas vraiment de mois qui se démarque. Le mois de juin est le mois ayant eu le plus d'accident en 2017 avec presque 6000 accidents. Par ailleurs, il n'y a pas de mois ayant des jours où les accidents sont plus important. On remarque cependant 3 pics d'accidents à certaines heures, le premier est vers 8 heures, le second vers 12 heures et enfin le dernier vers 17 heures. Ce sont en général des heures d'affluence sur les routes, ceci expliquerai pourquoi il y a plus d'accidents.

Nous allons maintenant créer une variable `dateheure` qui rassemblera les variables `an`, `mois`, `jour`, `heure` afin de former une date suivie de l'heure. Nous allons également créer une variable `date` qui rassemblera `an`, `mois` et `jour`, ainsi qu'une variable `semaine` à l'aide de la fonction `wday`. Ceci nous permettra d'observer le nombre d'accidents en semaine et le week-end. Ces variables formées nous permettront d'avoir une vu d'ensemble sur le nombre d'accidents par jour et la répartion des accidents dans la semaine.

Graphe sur le nombre d'accidents pour chaque jour de l'année :

```
caracteristics%>%
separate(hrmn,into=c("heure","minute"),2,convert=TRUE)%>% #separe heure et minute
mutate(dateheure=make_datetime(an, mois, jour,heure,minute))%>%
#crée la variable composé de la date et l'heure
ggplot()+
geom_freqpoly(aes(x=dateheure,y=..count..),binwidth = 86400 )+ #il y a 86400s dans 1j
labs(title= "Nombre d'accidents pour chaque jour de l'année",
     x="Date",
     y="Nombre d'accidents")
```

```
## Warning: Removed 1075 rows containing non-finite values (stat_bin).
```



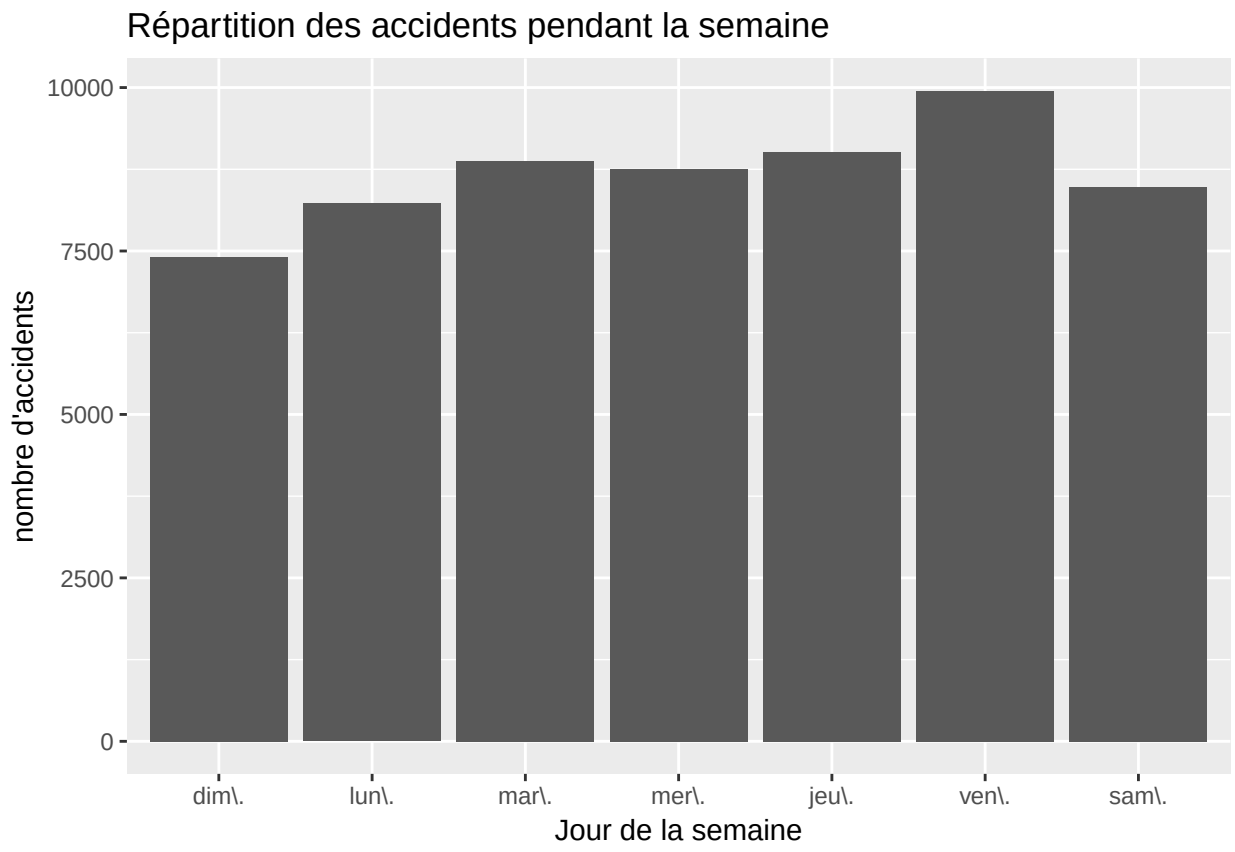
Observations :

Malheureusement, il est difficile de distinguer à quel jour correspond les pics d'accidents. Le graphe précédent était plus lisible. Le seul constat intéressant que l'on puisse faire est le faible nombre d'accidents qui se remarque au mois de décembre pendant les fêtes de fin d'année.

Remarque : Nous reviendrons plus tard sur ce graphique dans la partie ***Modélisation****.*

Graphe sur la répartition des accidents en semaine/week end :

```
caracteristics%>%
  mutate(date=make_date(an, mois, jour))%>%
  mutate(semaine=wday(date,label=TRUE))%>%
  ggplot(aes(semaine))+
  geom_bar()+
  labs(title="Répartition des accidents pendant la semaine",
       x="Jour de la semaine",
       y="nombre d'accidents")
```



Observations :

On constate qu'il y a moins d'accidents le dimanche. Cela peut s'expliquer par le fait que l'on prend moins souvent la route le dimanche car c'est un jour de repos, nous ne travaillons pas et très peu d'activités extérieur demandant un trajet en voiture sont disponible le dimanche.

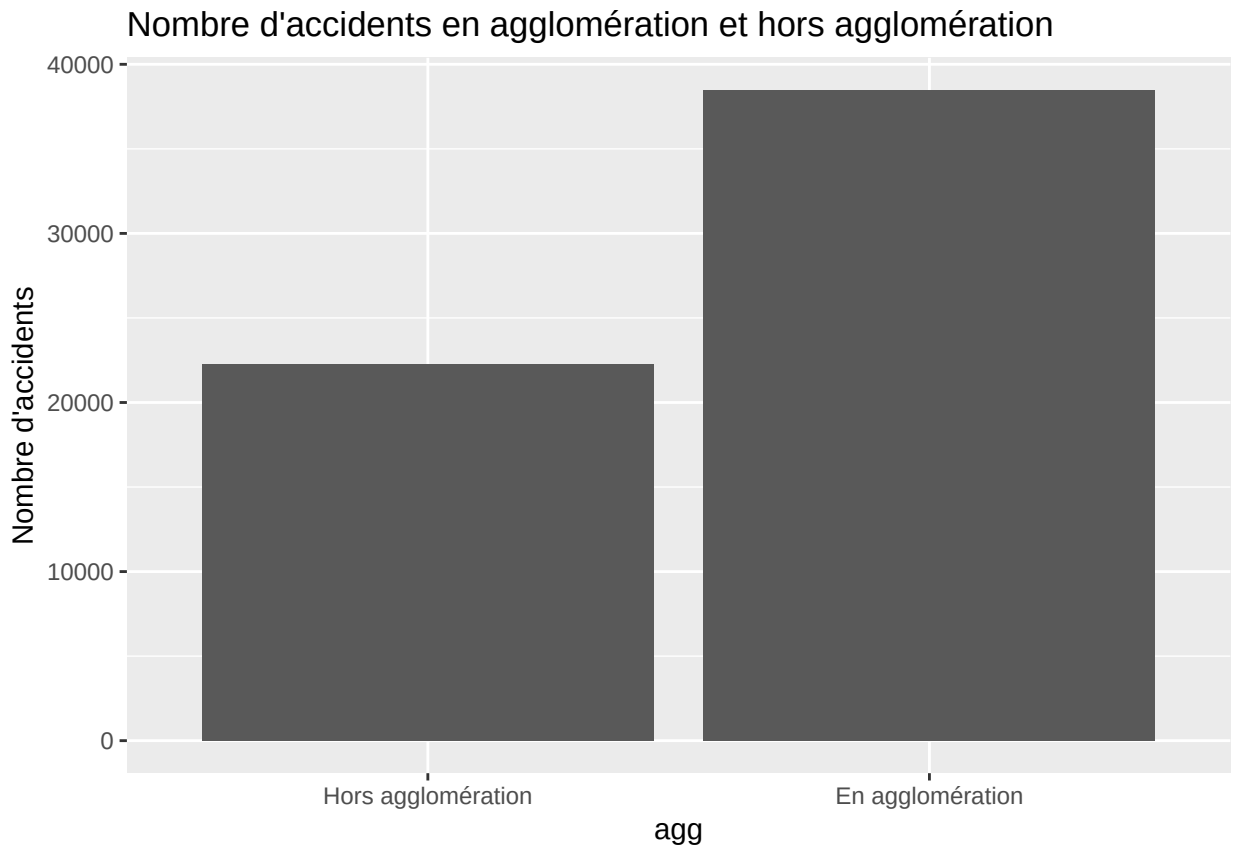
Exploration basée sur les lieux géographique (ville/département) :

Voici quelques questions que l'on pourrait se poser :

- Dans quel zone géographique retrouve t-on le plus grand nombre d'accident ?
- Quelle est le département touché par le plus grand nombre d'accident ?
- Quelle est la commune touché par le plus grand nombre d'accident ?

Graphe sur le nombre d'accident en agglomération et hors agglomération :

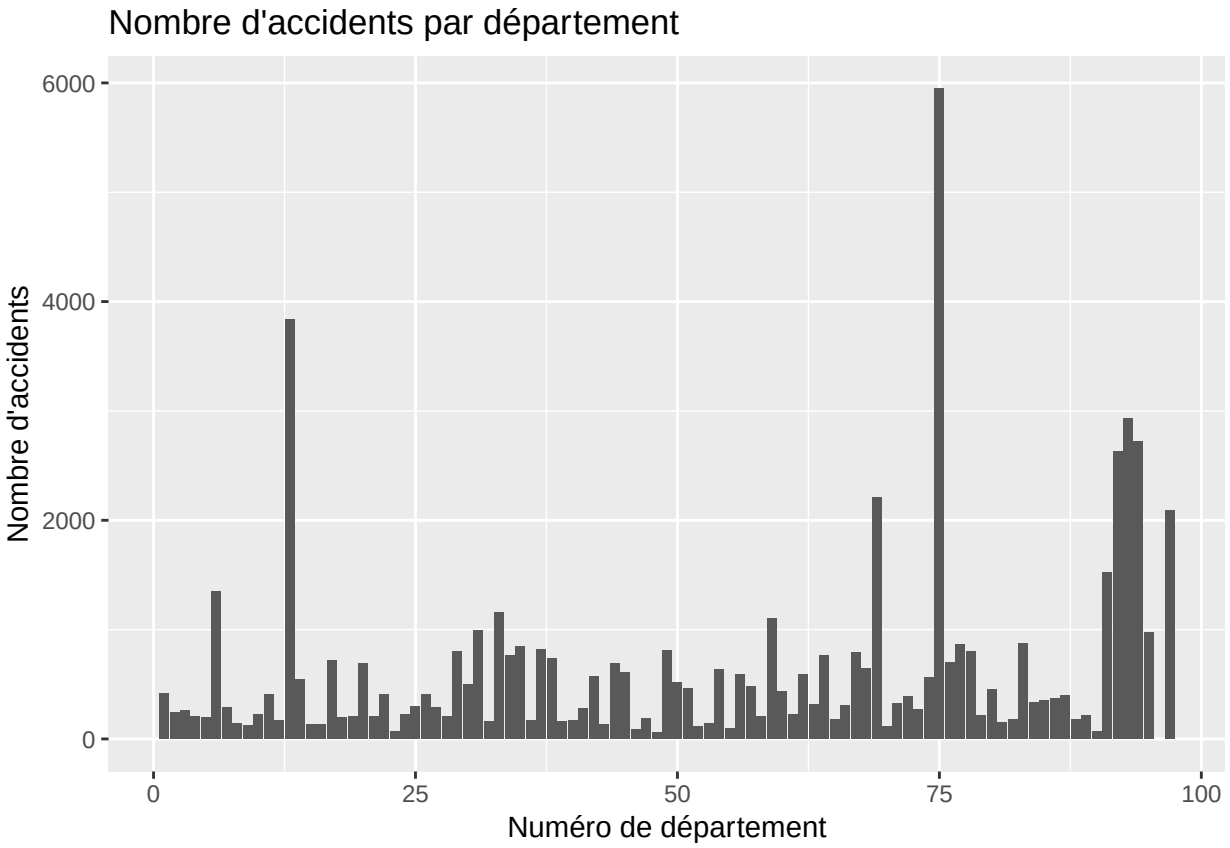
```
ggplot(caracteristics)+
  geom_bar(aes(agg))+
  scale_x_discrete(labels=c("1"="Hors agglomération","2"="En agglomération"))+
  labs(title="Nombre d'accidents en agglomération et hors agglomération",
       y="Nombre d'accidents")
```



Observations : Les accidents ont lieux plus fréquemment en agglomération qu'hors agglomération.

Graphe sur le nombre d'accidents par département :

```
ggplot(caracteristics)+
  geom_bar(aes(dep))+
  labs(title="Nombre d'accidents par département",
       y="Nombre d'accidents",
       x="Numéro de département")
```

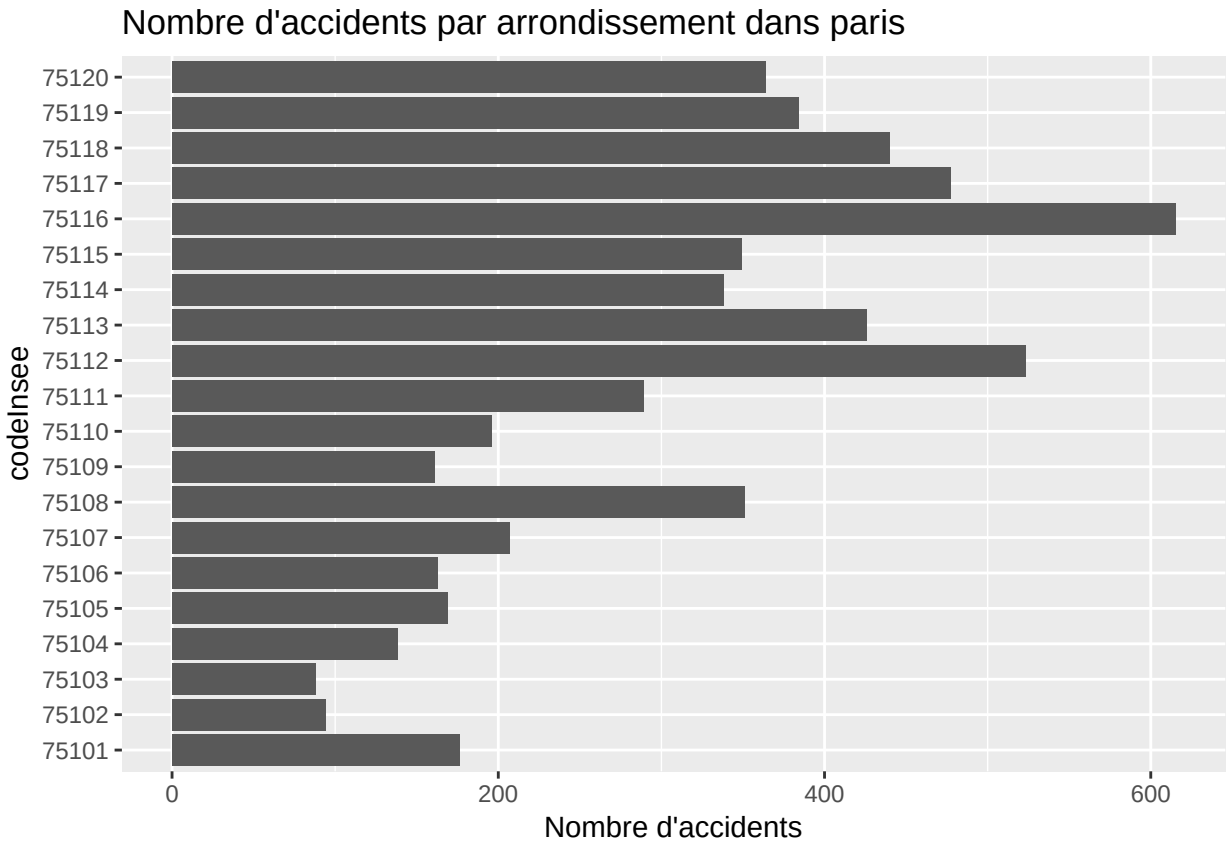


Observations : Les départements 13 et 75 semblent être les départements les plus touchés par les accidents de la route. Ceci peut s'expliquer par le fait que Paris et Marseille sont de grandes villes ayant beaucoup de circulation routière. Paris est la ville la plus touché par les accidents et sa banlieue (92, 93, 94) semble aussi pas mal touché dans l'ensemble.

Graphe sur le nombre d'accidents par arrondissement dans paris :

Nous pouvons déterminer quel arrondissement de Paris est le plus touché grâce au code Insee. Ce code Insee est formé du n° de département suivi du code commune Insee, ces deux variables sont *dep* et *com*. À l'aide de la fonction `unite` nous allons les rassembler. N.B : Nous pourrions aussi le faire pour n'importe quel département.

```
caracteristics %>%
unite(codeInsee,dep,com,sep="")%>%
filter(codeInsee>75000,codeInsee<76000)%>%
  ggplot()+
  geom_bar(aes(codeInsee))+
  coord_flip()+
  labs(title=c("Nombre d'accidents par arrondissement dans paris"),
       y=("Nombre d'accidents"))
```



Observations : C'est le 16ième arrondissement de Paris qui a la fréquence la plus élevé d'accident. Tandis que le 3ième arrondissement a la fréquence la plus faible.

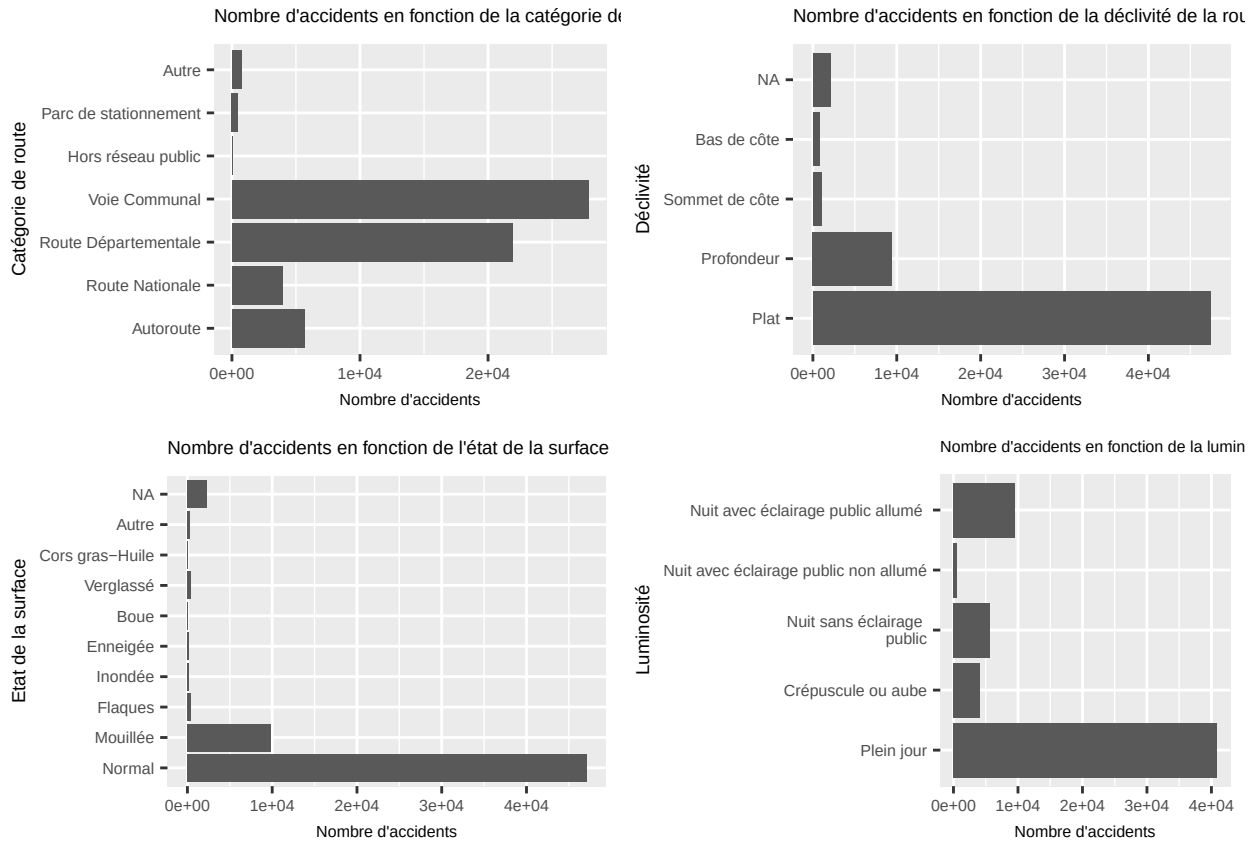
Exploration basée sur les routes où les accidents sont survenues

Voici quelques questions que l'on pourrait se poser :

- Quels types de routes sont à haut risque?
- Quel type de pente routière présente un risque élevé?
- Quels sont les conditions routière (mouillé, glissant,..) présentant les plus grands risques d'accidents ?
- A quelle condition de luminosité constate-t-on le plus grand nombre d'accidents ?

```
plot1_p<-ggplot(places)+
  geom_bar(aes(catr))+
  coord_flip()+
  scale_x_discrete(labels=c("1"="Autoroute","2"="Route Nationale","3"="Route Départementale",
        "4"="Voie Communal","5"="Hors réseau public","6"="Parc de stationnement","9"="Autre"))+
  scale_y_continuous(labels = scientific)+
  theme(axis.text.x = element_text(size=6),axis.text.y = element_text(size=6),
        axis.title.x = element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=7) )+
  labs(title=("Nombre d'accidents en fonction de la catégorie de route"),x=("Catégorie de route"),
       y=("Nombre d'accidents"))

plot2_p<-ggplot(places)+
  geom_bar(aes(prof))+
  coord_flip()+
  scale_x_discrete(labels=c("1"="Plat","2"="Profondeur","3"="Sommet de côte","4"="Bas de côte"))+
  scale_y_continuous(labels = scientific)+
  theme(axis.text.x = element_text(size=6),axis.text.y = element_text(size=6),
        axis.title.x = element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=7))+
  labs(title=("Nombre d'accidents en fonction de la déclivité de la route"),x=("Déclivité"),
       y=("Nombre d'accidents"))

plot3_p<-ggplot(places)+
  geom_bar(aes(surf))+
  coord_flip()+
  scale_x_discrete(labels=c("1"="Normal","2"="Mouillée","3"="Flaques","4"="Inondée",
        "5"="Enneigée","6"="Boue","7"="Verglassé","8"="Cors gras-Huile","9"="Autre"))+
  scale_y_continuous(labels = scientific)+
  theme(axis.text.x = element_text(size=6),axis.text.y  = element_text(size=6),
        axis.title.x = element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=7))+
  labs(title=("Nombre d'accidents en fonction de l'état de la surface"),x=("Etat de la surface"),
       y=("Nombre d'accidents"))

plot4_p<-ggplot(caracteristics)+
  geom_bar(aes(lum))+
  coord_flip()+
  scale_x_discrete(labels=c("1"="Plein jour","2"="Crépuscule ou aube","3"=" Nuit sans éclairage
                            public","4"="Nuit avec éclairage public non allumé",
                            "5"="Nuit avec éclairage public allumé "))+
  scale_y_continuous(labels = scientific)+
  theme(axis.text.x = element_text(size=6),axis.text.y = element_text(size=6),
        axis.title.x = element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=6))+
  labs(title=("Nombre d'accidents en fonction de la luminosité"),x=("Luminosité"),
```

```
       y=("Nombre d'accidents"))

grid.arrange(plot1_p,plot2_p,plot3_p,plot4_p)
```



Observations: On remarque avec ces 4 graphes que la plupart des accidents ont lieu sur les voie communales ou les routes départementales, sur des type de routes plates, à des conditions de route normale et en plein jour.

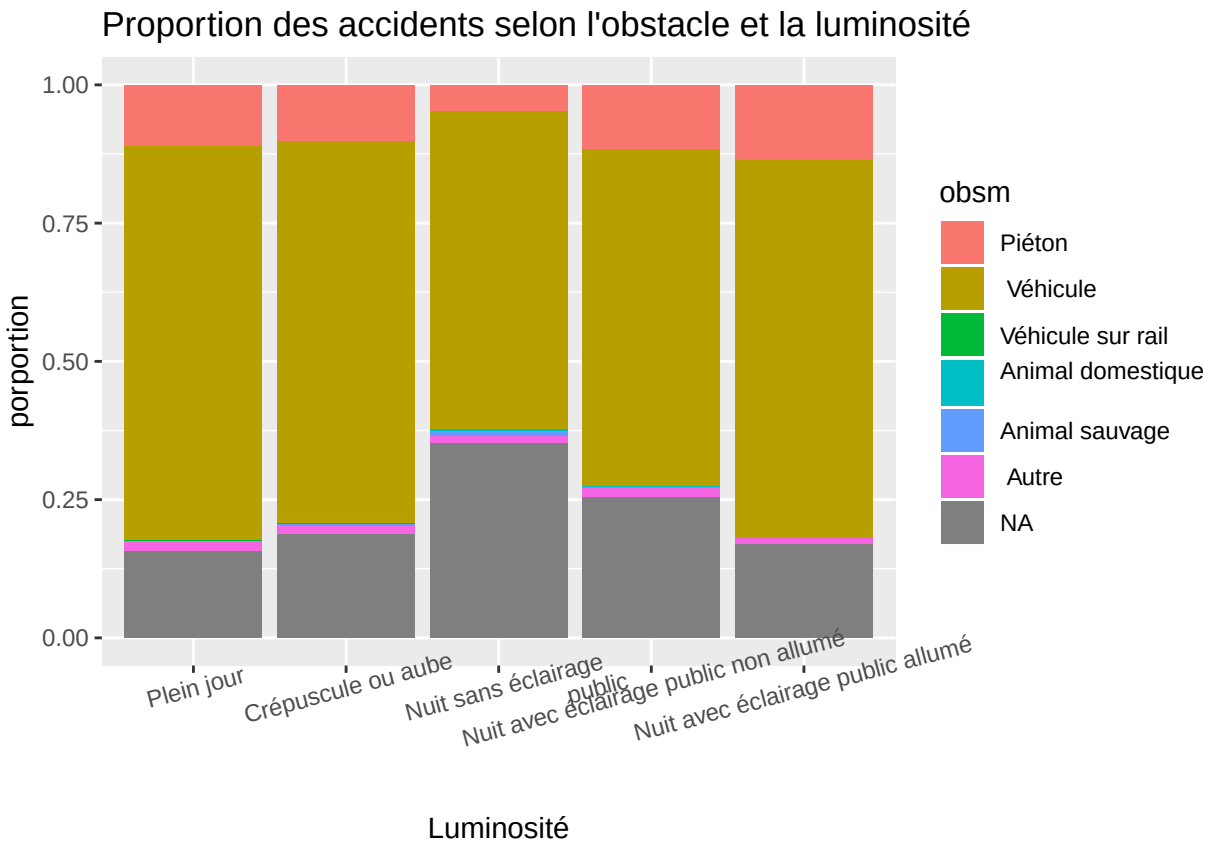
On peut également se poser la question suivant :

- Y a-t-il un lien entre la luminosité et le type d'obstacle heurté ?

La variable de la luminosité et du type d'obstacle heurté sont dans deux dataframe différent nous allons donc utiliser la fonction `left_join` afin de joindre les deux dataframe en conservant les observations des véhicules.

```
vehicles%>%
  left_join(caracteristics)%>%
  ggplot(aes(lum,fill=obsm))+
  geom_bar(position="fill")+
  scale_fill_discrete(labels=c("1"="Piéton","2"=" Véhicule ","4"="Véhicule sur rail ",
      "5"="Animal domestique
","6"="Animal sauvage ","9"=" Autre "))+
  scale_x_discrete(labels=c("1"="Plein jour","2"="Crépuscule ou aube","3"=" Nuit sans éclairage
                            public","4"="Nuit avec éclairage public non allumé",
                            "5"="Nuit avec éclairage public allumé "))+
  theme(axis.text.x = element_text(angle=15))+
  labs(title=("Proportion des accidents selon l'obstacle et la luminosité"),
       y=("porportion"),x=("Luminosité"))
```

```
## Joining, by = "Num_Acc"
```



Observations : On constate dans un premier temps que les véhicules rentrent la plupart du temps en collision avec d'autres véhicules. Cependant, il n'y a aucun lien à tirer avec la luminosité car toute les proportions par rapport à la luminosité sont globalement pareil.

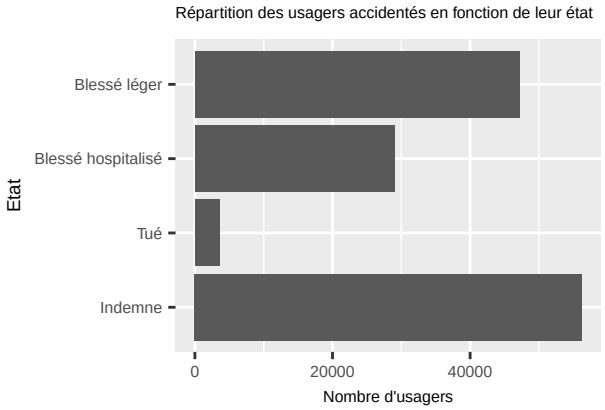
Exploration basée sur les personnes impliquées dans les accidents

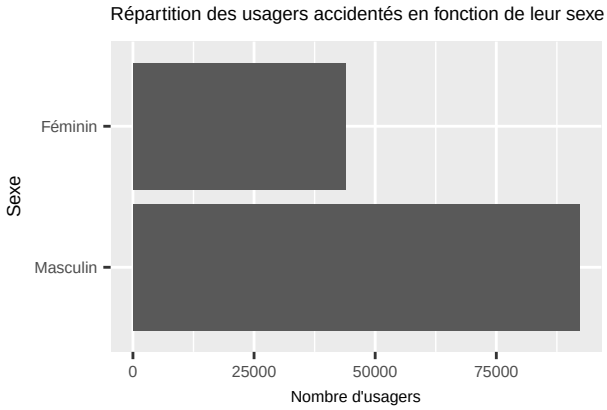
Voici quelques questions que l'on pourrait se poser :

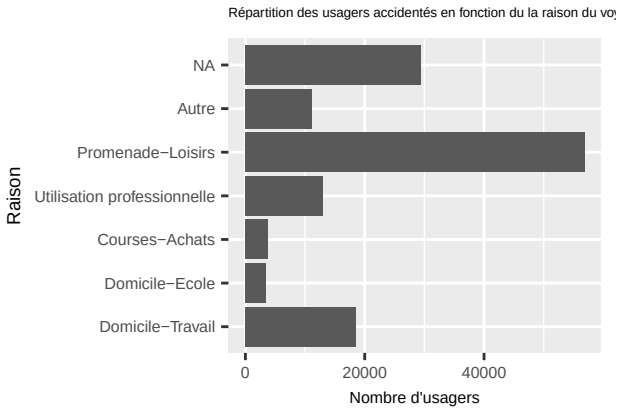
- Quel était l'état des gens après l'accident?
- Quelle était la répartition par âge des personnes impliquées?
- Quelle était la répartition par sexe des personnes impliquées?
- Quelle était la circonstance du voyage?

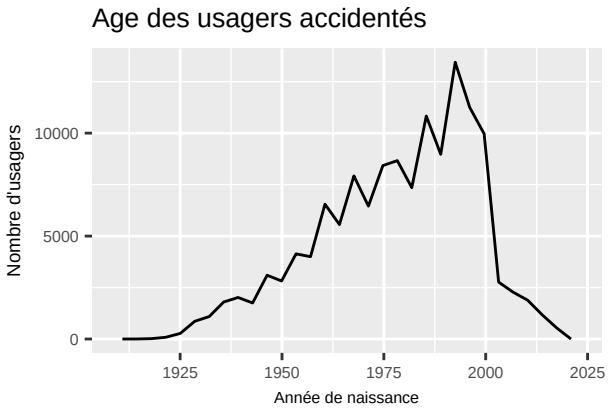
```
plot1_u<-ggplot(users)+
  geom_bar(aes(grav))+
  scale_x_discrete(labels=c("1"="Indemne","2"="Tué","3"="Blessé hospitalisé","4"="Blessé léger"))+
  coord_flip()+
  theme(axis.text.x = element_text(size=6),axis.text.y = element_text(size=6),
        axis.title.x = element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=6) )+
  labs(title=("Répartition des usagers accidentés en fonction de leur état"),x=("Etat"),
       y=("Nombre d'usagers"))


plot2_u<-ggplot(users)+
  geom_bar(aes(sexe))+
  scale_x_discrete(labels=c("1"="Masculin","2"="Féminin"))+
  coord_flip()+
  theme(axis.text.x = element_text(size=6),axis.text.y = element_text(size=6),
        axis.title.x = element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=7) )+
  labs(title=("Répartition des usagers accidentés en fonction de leur sexe"),x=("Sexe"),
       y=("Nombre d'usagers"))

plot3_u<-ggplot(users)+
  geom_bar(aes(trajet))+
  scale_x_discrete(labels=c("1"="Domicile-Travail","2"="Domicile-Ecole","3"="Courses-Achats",
      "4"="Utilisation professionnelle","5"="Promenade-Loisirs","9"="Autre"))+
  coord_flip()+
  theme(axis.text.x = element_text(size=6),axis.text.y = element_text(size=6),
        axis.title.x = element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=5) )+
  labs(title=("Répartition des usagers accidentés en fonction du la raison du voyage"),x=("Raison"),
       y=("Nombre d'usagers"))

plot4_u<-ggplot(users)+
  geom_freqpoly(aes(x=an_nais,y=..count..))+
  theme(axis.text.x = element_text(size=6),axis.text.y = element_text(size=6),
        axis.title.x =element_text(size=6),axis.title.y = element_text(size=7),
        plot.title = element_text(size=10) )+
  labs(title=("Age des usagers accidentés"),x=("Année de naissance"),
       y=("Nombre d'usagers"))


grid.arrange(plot1_u,plot2_u,plot3_u,plot4_u)
```

```
## Warning: Removed 37 rows containing non-finite values (stat_bin).
```









Observations :

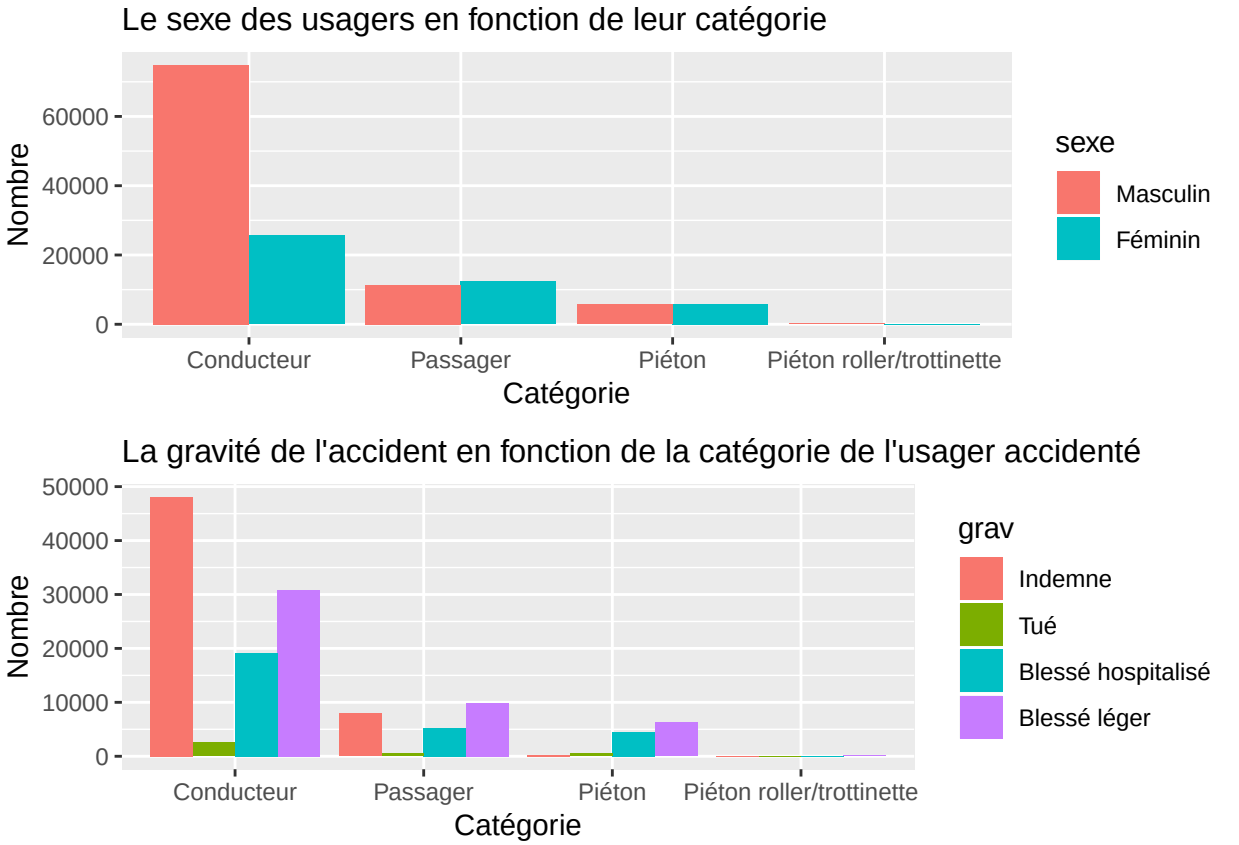
On constate qu'une plus grande fréquence de personnes de sexe masculin sont impliqués dans les accidents routiers. Dans la mojorité des cas, les personnes impliqués ne sont pas tués et s'en sorte plus souvent indemne. Par ailleurs, les accidents ont le plus souvent lieu lors de trajet pour une promenade ou un loisirs, et ceci pourrait s'expliquer par le fait que l'on est plus détendu et moins attentif lorsque l'on est dans ce genre de situation. Et enfin, les accidentés ont en majorité 28/29ans au moment de l'accident.

Ces résultats manquent néanmoins beaucoup de précisions. On peut encore se poser plusieurs questions comme :

- Les hommes sont-ils véritablement ceux qui provoque le plus d'accidents (conducteur) ? Ou bien sont-ils ceux qui sont le plus souvent victime de l'accident (piéton, passagers,..) ?
- Y a t-il un lien entre la gravité de l'accident et la catégorie de l'usager (conducteur, piéton, passager) ?
- Y a t-il un lien entre l'âge des accidentés et la gravité de l'accident ?
- Quel âge ont en moyenne les conducteurs/passager/piéton impliqués dans les accidents ?

```
plot2_1<-ggplot(users,aes(catu,fill=sexe))+
  geom_bar(position="dodge")+
  scale_fill_discrete(labels=c("1"="Masculin","2"="Féminin"))+
  scale_x_discrete(labels=c("1"="Conducteur","2"="Passager",
        "3"="Piéton","4"="Piéton roller/trottinette"))+
  theme(plot.title = element_text(size=12) )+
  labs(title="Le sexe des usagers en fonction de leur catégorie",
       x=("Catégorie"),y=("Nombre"))

plot2_2<-ggplot(users,aes(catu,fill=grav))+
  geom_bar(position="dodge")+
  scale_fill_discrete(labels=c("1"="Indemne","2"="Tué",
```

```
        "3"="Blessé hospitalisé","4"="Blessé léger"))+
  scale_x_discrete(labels=c("1"="Conducteur","2"="Passager",
        "3"="Piéton","4"="Piéton roller/trottinette"))+
  theme(plot.title = element_text(size=12) )+
  labs(title="La gravité de l'accident en fonction de la catégorie de l'usager accidenté",
       x=("Catégorie"),y=("Nombre"))

grid.arrange(plot2_1,plot2_2,nrow=2)
```



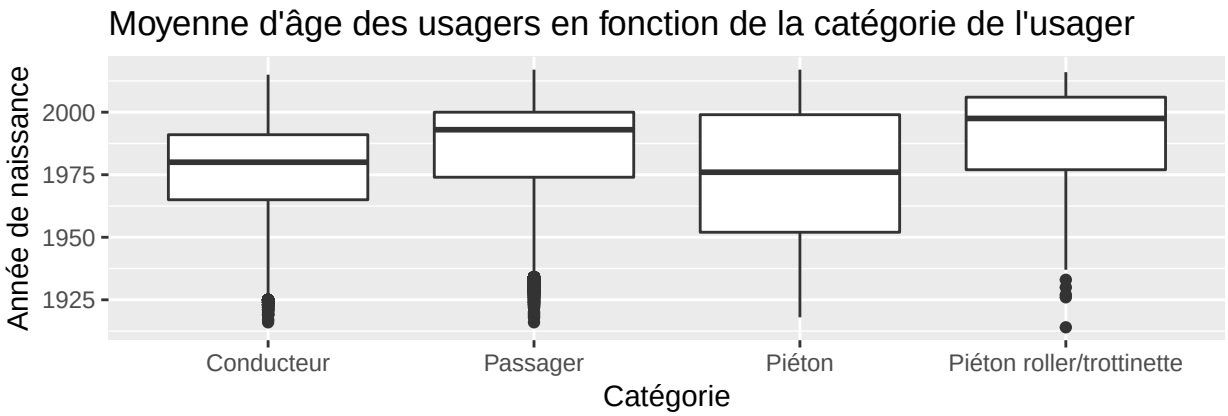
Observations :

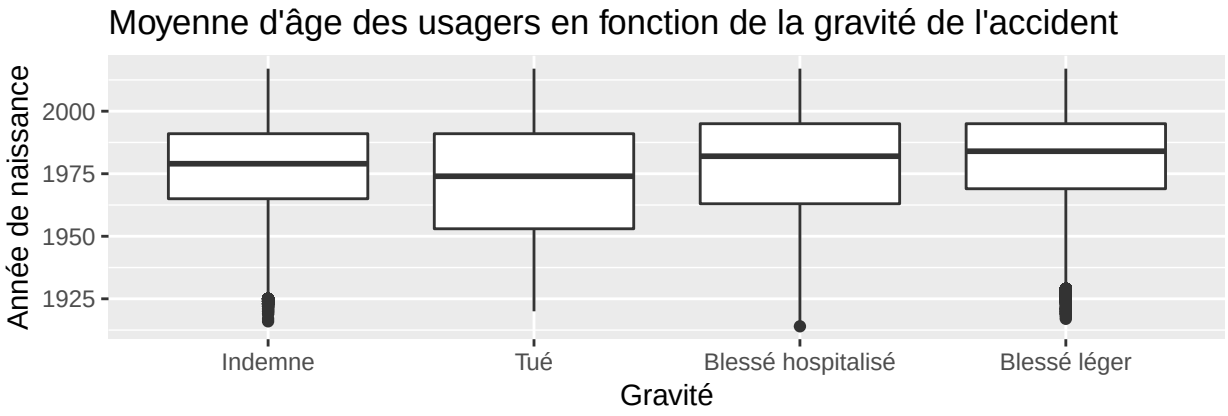
Avec le 1er graphe, on constate bien que les hommes sont bien ceux qui provoque le plus d'accident (les conducteurs sont en majorité des hommes tandis que ca n'est pas le cas pour les passagers et les piétons). Avec le 2ième graphe, on voit que les conducteurs sont plus généralement indemne, les passagers sont plus généralement blessé légèrement et les piétons ne sont pratiquement jamais indemne.

```
plot2_3<-ggplot(users,aes(catu,an_nais))+
  geom_boxplot()+
  scale_x_discrete(labels=c("1"="Conducteur","2"="Passager",
        "3"="Piéton","4"="Piéton roller/trottinette"))+
labs(title=("Moyenne d'âge des usagers en fonction de la catégorie de l'usager"),
     y=("Année de naissance"),x=("Catégorie"))

plot2_4<-ggplot(users,aes(grav,an_nais))+
  geom_boxplot()+
  scale_x_discrete(labels=c("1"="Indemne","2"="Tué",
        "3"="Blessé hospitalisé","4"="Blessé léger"))+
```

```
labs(title=("Moyenne d'âge des usagers en fonction de la gravité de l'accident"),
     y=("Année de naissance"),x=("Gravité"))

grid.arrange(plot2_3,plot2_4)
```

```
## Warning: Removed 37 rows containing non-finite values (stat_boxplot).

## Warning: Removed 37 rows containing non-finite values (stat_boxplot).
```





Observations :

Par ailleurs, aucun lien n'est à constater entre l'âge et l'état des accidentés (la moyenne d'âge est globalement la même pour les 4 catégories). Entre l'âge et la catégorie de l'usagers (conducteur, passager, piéton..) il n'y a pas non plus de grande différence à constater même si les piétons en roller/trottinette accidenté reste relativement plus jeune que le reste (ceci peut s'expliquer par le fait que ceux sont des moyens de transports utilisés davantage par les jeunes).

Exploration basée sur le système de sécurité :

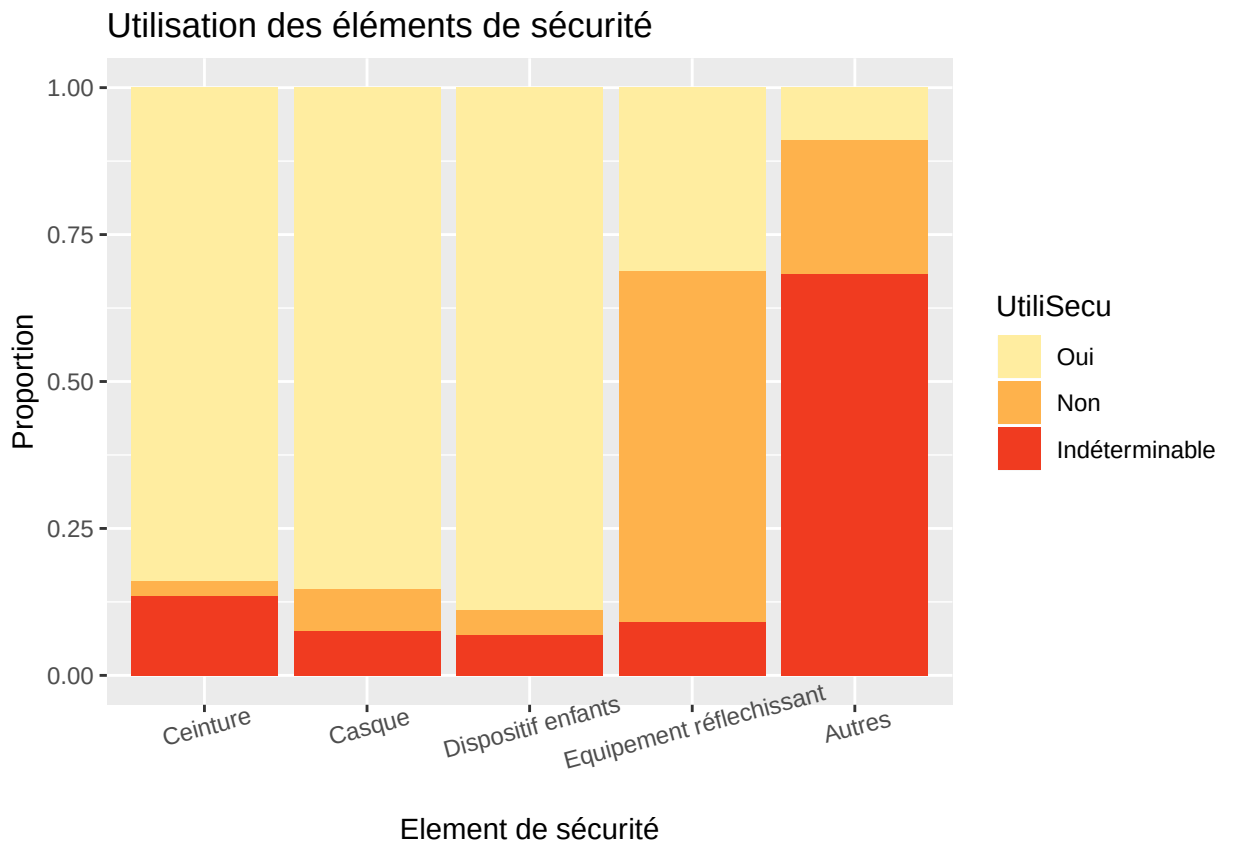
Voici quelques questions que l'on pourrait se poser :

- Les équipement de sécurité sont-ils utilisés ? À quel fréquence ?
- Quel est l'état de l'usagers en fonction de l'utilisation ou non de l'équipement de sécurité ?

La variable `secu` est composé de deux caractères, le premier concerne l'existence d'un équipement de sécurité et le second concerne l'utilisation de l'équipement de sécurité. Ainsi pour pouvoir étudier correctement cette variable, il nous faut la décomposer en deux. Nous ferons cela à l'aide de la fonction `separate`. Nous pourrons ensuite tracer les graphes afin de répondre à nos questions.

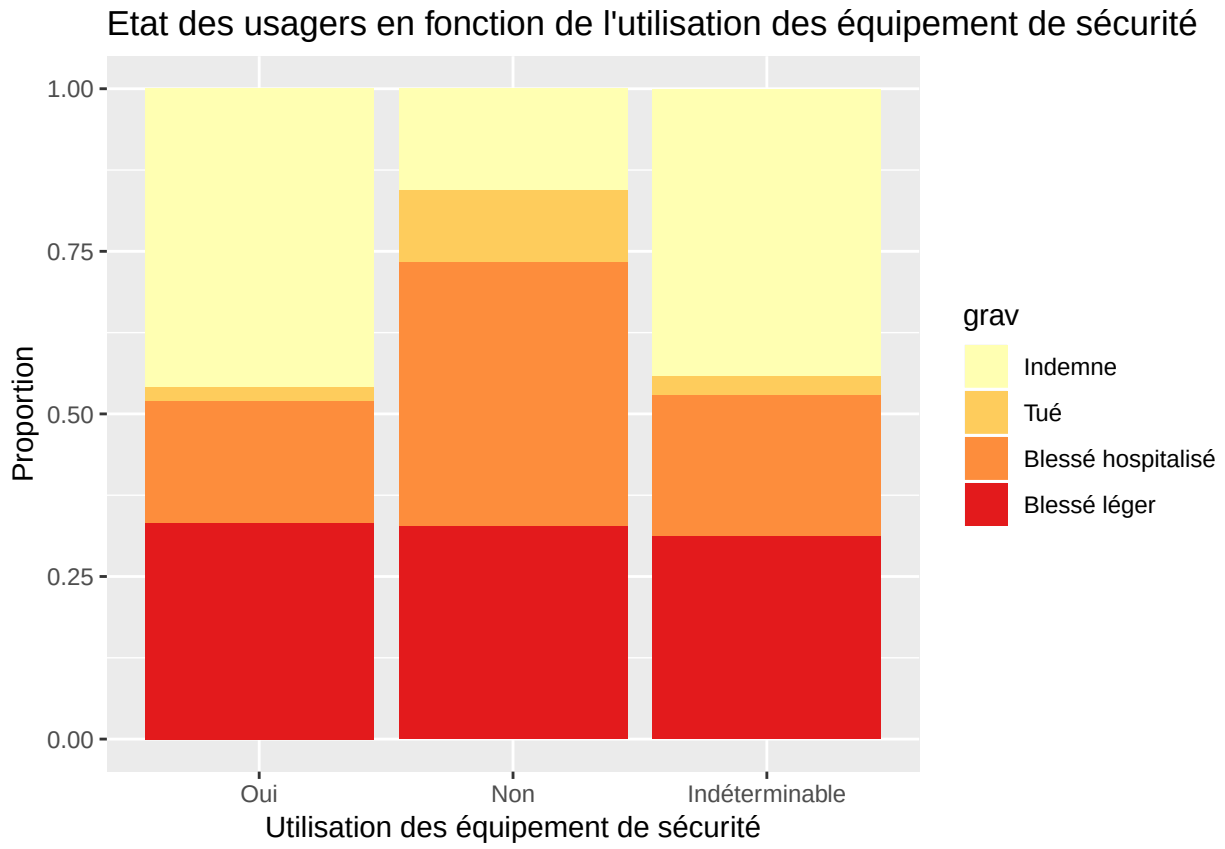
Graphe sur la fréquence d'utilisation des différents équipements de sécurité:

```
users%>%
  separate(secu,into=c("EquiSecu","UtiliSecu"),sep=1)%>%
  subset(!is.na(EquiSecu))%>% #permet de retirer les valeurs manquante de EquiSecu du graphe
  ggplot(aes(EquiSecu,fill=UtiliSecu))+
  geom_bar(position="fill")+
  scale_fill_brewer(palette="YlOrRd",labels=c("1"="Oui","2"="Non",
          "3"="Indéterminable"),breaks=c("1","2","3"))+
  scale_x_discrete(labels=c("1"="Ceinture","2"="Casque",
          "3"="Dispositif enfants","4"="Equipement réflechissant","9"="Autres"))+
  theme(axis.text.x = element_text(angle=15))+
  labs(title=("Utilisation des éléments de sécurité"),y=("Proportion"),x=("Element de sécurité"))
```



Graphe sur l'état des usagers en fonction de l'utilisation des équipements de sécurité :

```
users%>%
  separate(secu,into=c("EquiSecu","UtiliSecu"),sep=1)%>%
  subset(!is.na(UtiliSecu))%>% #permet de retirer les valeurs manquante de UtiliSecu du graphe
  ggplot(aes(UtiliSecu,fill=grav))+
  geom_bar(position="fill")+
  scale_fill_brewer(palette="YlOrRd",labels=c("1"="Indemne","2"="Tué",
          "3"="Blessé hospitalisé","4"="Blessé léger"))+
  scale_x_discrete(labels=c("1"="Oui","2"="Non","3"="Indéterminable"))+
  labs(title=("Etat des usagers en fonction de l'utilisation des équipement de sécurité"),
       x=("Utilisation des équipement de sécurité"),y=("Proportion"))
```



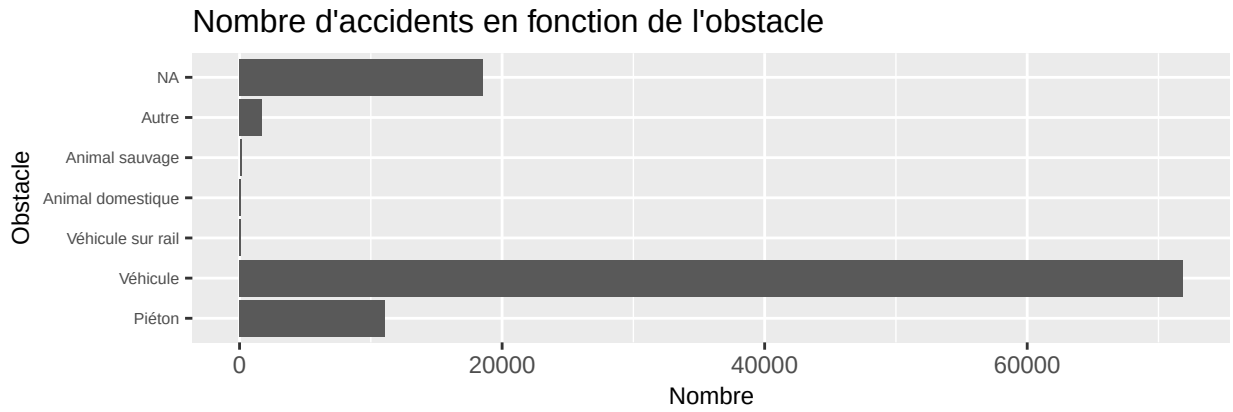
Observations : On constate que seul l'équipement réflechissant est en majorité non utilisé. Par ailleurs, de manière générale, lorsque les équipements de sécurité ne sont pas utilisé il y a davantage de tué et de blessé hospitalisé et beaucoup moins d'indemne que lorsqu'ils sont utilisé.

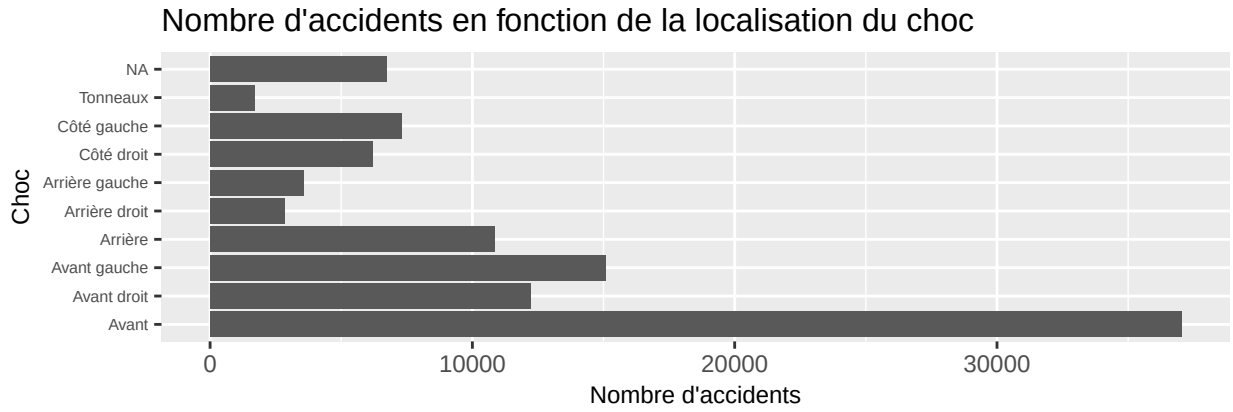
Exploration basée sur la collisions des vehicules :

Voici quelques questions que l'on pourrait se poser :

- Quels est l'élément de collision d'un véhicule le plus fréquent dans les accidents ?
- À quel endroit du véhicule ont lieu le plus fréquemment les chocs ?

```
plot1_v<-ggplot(vehicles)+
  geom_bar(aes(obsm))+
  scale_x_discrete(labels=c("1"="Piéton","2"="Véhicule","4"="Véhicule sur rail",
      "5"="Animal domestique","6"="Animal sauvage","9"="Autre"))+
  coord_flip()+
  theme(axis.text.y = element_text(size=6),plot.title = element_text(size=12),
        axis.title.y = element_text(size=9),axis.title.x = element_text(size=9) )+
  labs(title="Nombre d'accidents en fonction de l'obstacle", x=("Obstacle"),y=("Nombre"))


plot2_v<-ggplot(vehicles)+
  geom_bar(aes(choc))+
  scale_x_discrete(labels=c("1"="Avant","2"="Avant droit","3"="Avant gauche",
      "4"="Arrière","5"="Arrière droit","6"="Arrière gauche","7"="Côté droit",
      "8"="Côté gauche","9"="Tonneaux"))+
  coord_flip()+
  theme(axis.text.y = element_text(size=6),plot.title = element_text(size=12),
        axis.title.y = element_text(size=9),axis.title.x = element_text(size=9) )+
  labs(title="Nombre d'accidents en fonction de la localisation du choc", x=("Choc"),
       y=("Nombre d'accidents"))

grid.arrange(plot1_v,plot2_v)
```





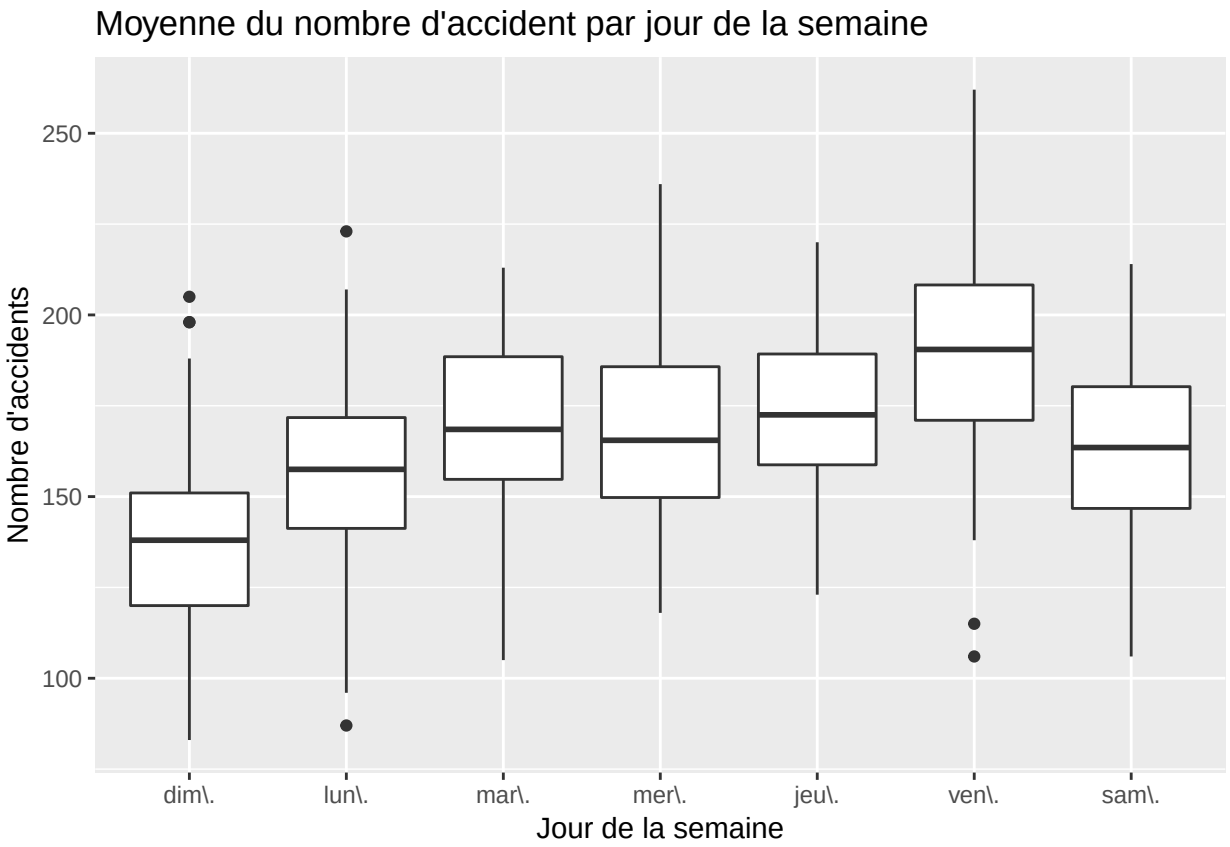
Observations :

On constate sans surprise que les obstacles les plus fréquent sont les véhicules et les piétons.Et aussi sans surprise, les chocs ont le plus souvent lieu à l'avant de la voiture (avant fauche, avant droit, avant).

Modélisation

On se demande toujours ce qui peut bien augmenter le nombre d'accidents quotidiens. Rappelons nous du graphe sur le nombre d'accidents pour chaque jour de l'année, il ne nous avait malheureusement donné aucunes informations concernant les jours les plus affectés par les accidents. Nous allons donc reprendre cette question afin de faire un modèle qui nous en dira davantage. Nous allons dans un premier temps compter le nombre d'accident par jour que nous allons stocker dans une nouvelle dataframe que nous appelerons `model` et nous visualiserons la répartition des accidents en fonction des jours de la semaine.

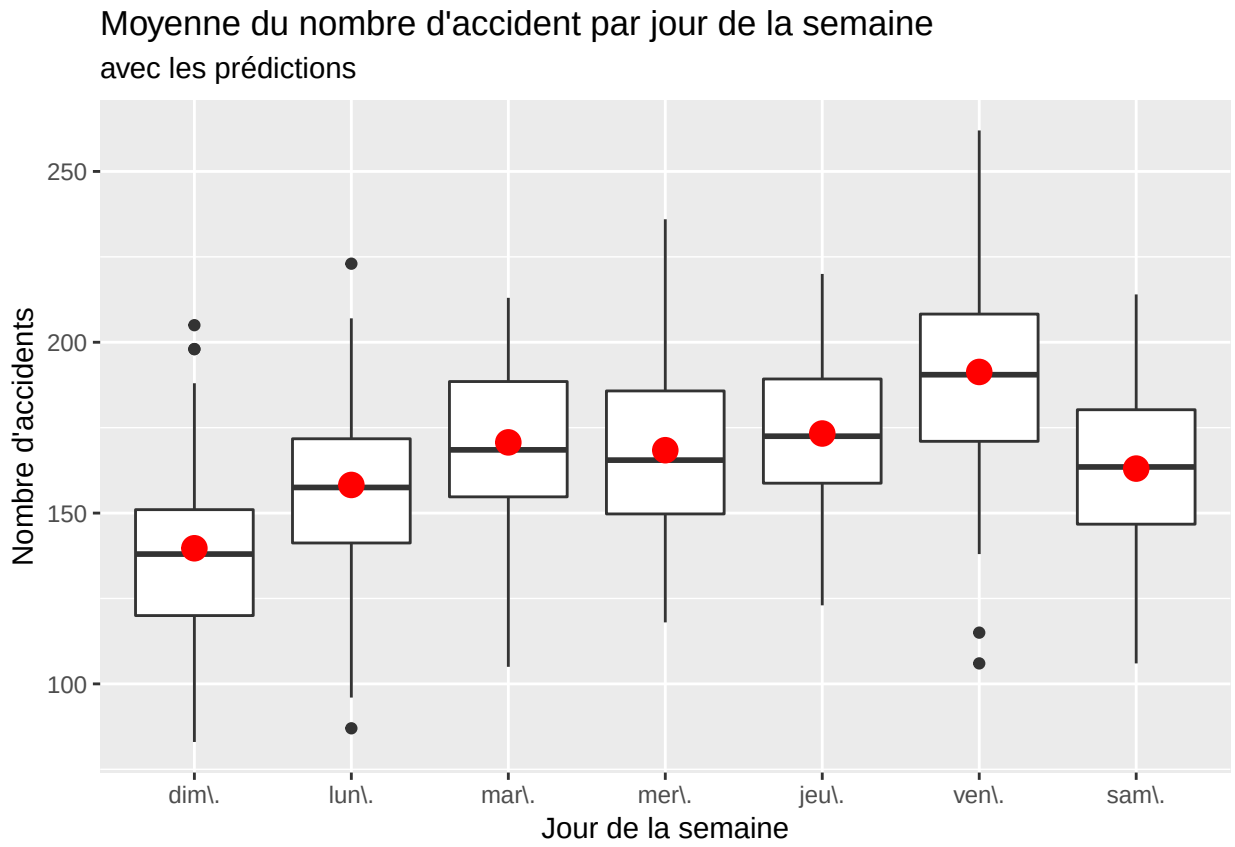
```
model <- caracteristics %>%
  mutate(date = make_date(an, mois, jour)) %>%
  group_by(date) %>%
  summarise(n = n())

model <- model %>%
  mutate(semaine = wday(date, label = TRUE))

ggplot(model, aes(semaine, n)) +
  geom_boxplot()+
  labs(title=("Moyenne du nombre d'accident par jour de la semaine"),x=("Jour de la semaine"),
       y=("Nombre d'accidents"))
```



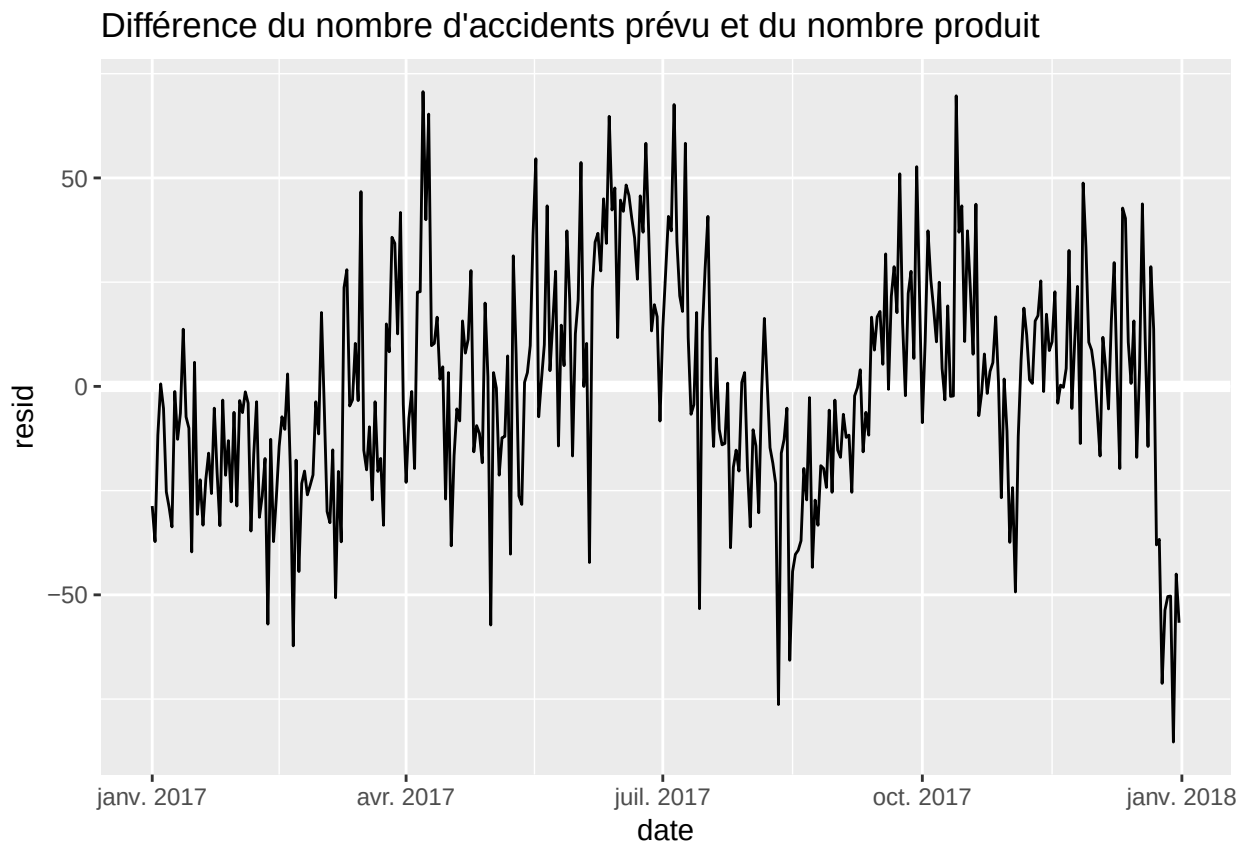
On constate qu'il y a moins d'accident le dimanche et plus d'accident le vendredi. Le plus faible nombre d'accidents le dimanche peut s'expliquer par le fait que l'on prend moins souvent la route le dimanche car c'est un jour de repos, nous ne travaillons pas et très peu d'activités extérieur demandant un trajet en voiture sont disponible le dimanche.

Nous allons donc d'abord ajuster le modèle et afficher ses prédictions superposées sur les données d'origine à l'aide de la fonction `data_grid` et de la fonction `add_predictions`, ensuite nous calculerons et visualiseront les résidus à l'aide de la fonction `add_residuals`.

```
mod <- lm(n ~ semaine, data = model)

grid <- model %>%
  data_grid(semaine) %>%
  add_predictions(mod, "n")

ggplot(model, aes(semaine, n)) +
  geom_boxplot() +
  geom_point(data = grid, colour = "red", size = 4)+
  labs(subtitle=("avec les prédictions"),title=("Moyenne du nombre d'accident par jour de la semaine"),
       x=("Jour de la semaine"),y=("Nombre d'accidents"))
```



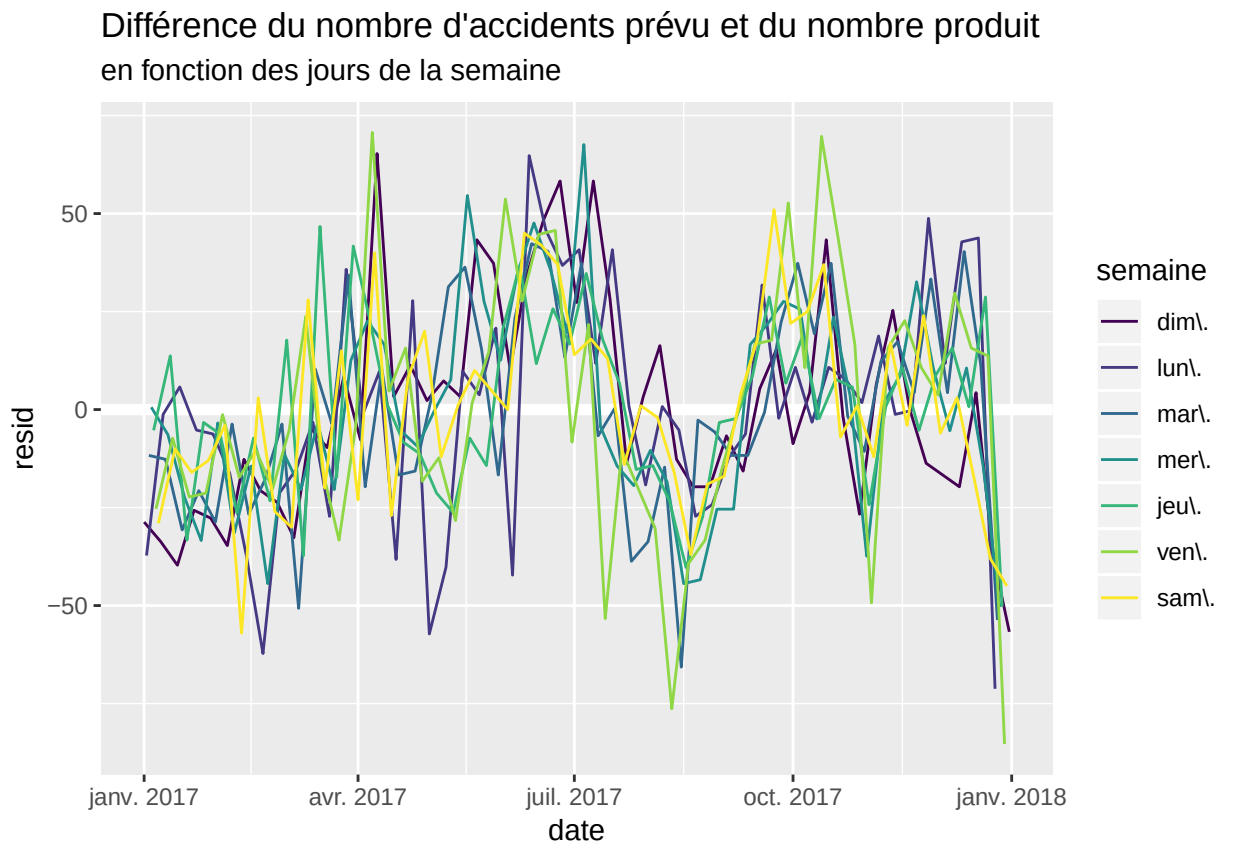

```
model <- model %>%
  add_residuals(mod)
model %>%
  ggplot(aes(date, resid)) +
  geom_ref_line(h = 0) +
  geom_line()+
  labs(title="Différence du nombre d'accidents prévu et du nombre produit")
```



Ce graphe nous permet d'observer l'écart par rapport au nombre d'accidents prévu, compte tenu du jour de la semaine. Malheureusement ce graphe ne nous permet pas de voir d'autres motifs plus subtils (que ceux du modèle d'origine) qui auraient pu nous interesser.

Nous allons tout de même essayer d'approfondir sur certains points. Nous observons quelques petites chutes du nombre d'accidents, nous allons alors dessiner une parcelle avec une ligne pour chaque jour de la semaine afin de savoir à quel jour de la semaine correspondent ces chutes.

```
ggplot(model, aes(date, resid, colour = semaine)) +
  geom_ref_line(h = 0) +
  geom_line()+
  labs(title="Différence du nombre d'accidents prévu et du nombre produit",
       subtitle="en fonction des jours de la semaine")
```



On constate que ces “chutes” ne correspondent pas toutes au même jour (samedi, dimanche, lundi, vendredi,..).

Nous allons alors essayer de trouver une explications en les filtrant et en regardant à quelle date elles correspondent.

```
model %>%
  filter(resid < -50)
```

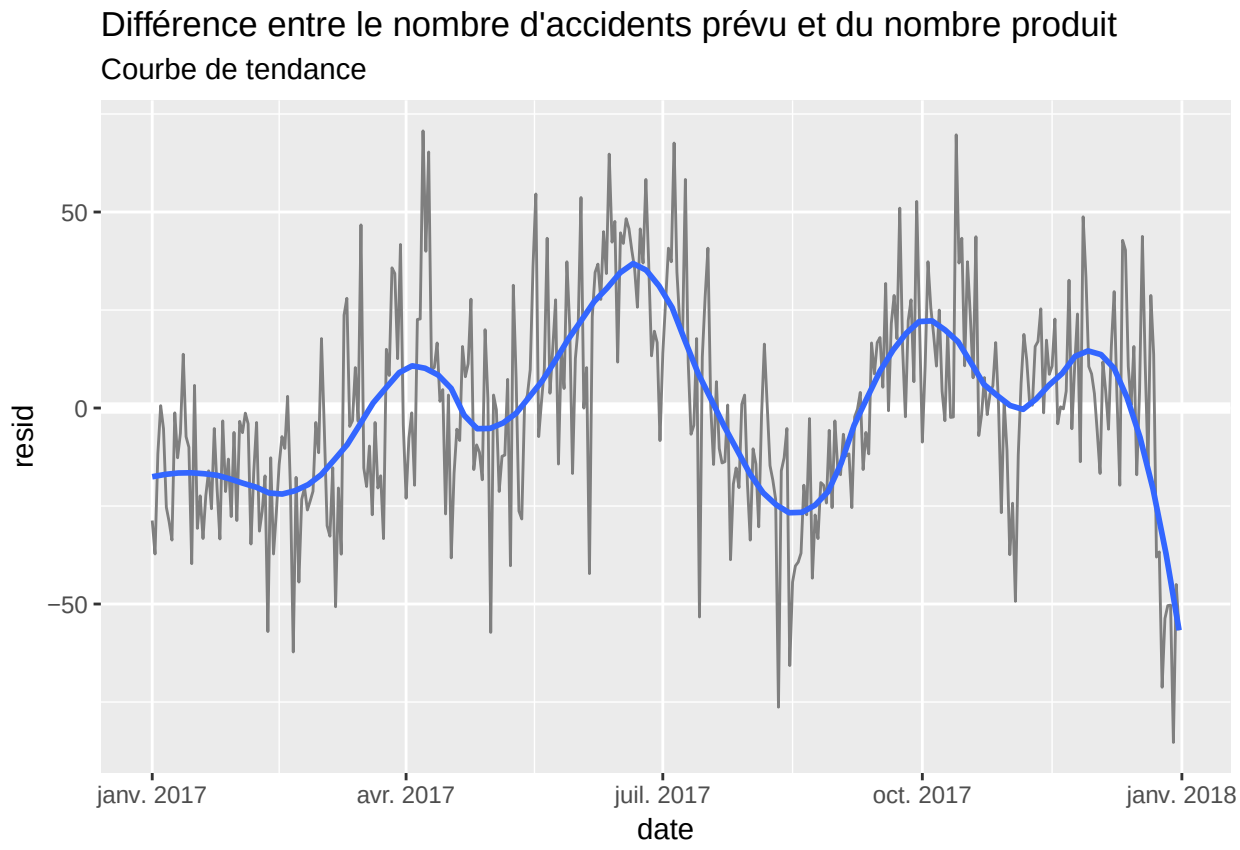
```
## # A tibble: 13 x 4
##    date           n semaine  resid
##    <date>     <int> <ord>    <dbl>
##  1 2017-02-11   106 "sam\\." -57.0
##  2 2017-02-20    96 "lun\\." -62.2
##  3 2017-03-07   120 "mar\\." -50.7
##  4 2017-05-01   101 "lun\\." -57.2
##  5 2017-07-14   138 "ven\\." -53.3
##  6 2017-08-11   115 "ven\\." -76.3
##  7 2017-08-15   105 "mar\\." -65.7
##  8 2017-12-25    87 "lun\\." -71.2
##  9 2017-12-26   117 "mar\\." -53.7
## 10 2017-12-27   118 "mer\\." -50.4
## 11 2017-12-28   123 "jeu\\." -50.3
## 12 2017-12-29   106 "ven\\." -85.3
## 13 2017-12-31    83 "dim\\." -56.7
```

On constate que la moitié correspondent au fêtes de fin d’années. Le 1er mai et le 14 juillet et le 15 août sont des jours fériés. Ainsi sur les 13 jours ayant le moins d’accidents 9 jours correspondent à des jours férié et des

périodes de fêtes. Pendant ces jours, nous utilisons en effet moins les routes.

```
model %>%
  ggplot(aes(date, resid)) +
  geom_ref_line(h = 0) +
  geom_line(colour = "grey50") +
  geom_smooth(se = FALSE, span = 0.20)+
  labs(title="Différence entre le nombre d'accidents prévu et du nombre produit",
       subtitle="Courbe de tendance")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Malgré que l'on n'ai pas tiré beaucoup d'informations de ce modèle, on peut tout de même constater qu'il n'y a pas de tendance linaire des accidents, il y a au contraire des périodes de hausse et de baisse qui se suivent et forme une coubre presque sinusoïdale. Il y a donc des périodes de hausses des accidents mais il est difficile de voir à quoi elles correspondent vraiment et si elles sont particulière à cette année ou non (ils nous auraient fallu des données qui s'étendent sur plusieurs années). Une hypothèse sur la baisse des accidents pendant les périodes de vacances scolaire peut être interrogé car il y a en effet une baisse pendant le mois de février, avril, juillet/août et novembre.

Conclusion de cette analyse de données

Cette vaste analyse de données, nous a permis de voir quelques facteurs potentiels qui seraient susceptible d'augmenter le nombre d'accidents. Néanmoins, il est très difficile de donner les facteurs les plus prédominants, d'autres méthodes ou analyses statistiques sont à déployer afin de répondre à la question.