Université de Lille

PROGRAMMATION FONCTIONNELLE EN R

Filière : Statistiques

Projet : Nettoyage et manipulation de données sur R

Réalisé par :
M. Mohamed Ben Yassine

Sous la direction de :
Pr. Sławek Staworko

Année universitaire
2019/2020

Table des matières

Chapitre 1

Introduction

Dans ce projet, nous allons explorer le jeu de données Titanic sur le logiciel R, la première partie du projet consiste à explorer et comprendre la structure des données pour pouvoir corriger les problèmes qui peuvent rendre l'étape de l'analyse difficile 'à savoir les valeurs manquantes, les variables avec un type non convenable ...
C'est la partie de nettoyage de données.
Ensuite on va faire des graphes pour avoir une vision sur nos données et les relations entre les variables.

Chapitre 2

Description des variables du jeu de données

Cette base de données contient des informations sur les passagers du navire Titanic, chaque ligne représente un passager.
On a 13 variables dans la base de données Titanic :

- Survival : prend 0 si le passager est mort durant le naufrage, 1 sinon.
- Pclass : cette variable contient la classe de billet des passagers, soit classe 1, 2 ou classe 3.
- sex : Variable qui indique le sexe de chaque passager.
- Âge : Variable qui indique l'age de chaque passager.
- Parch : Variable qui indique le nombre de parents ou d'enfants qui voyagent avec chaque passager.
- Sibsp : Variable qui indique le nombre de frères, sœurs ou‘ des époux qui voyagent avec chaque passager.
- Ticket : Numéro ticket.
- Fare : Prix payé.
- Cabin : Numéro de cabine.
- Embarked : indique la porte d'embarquement parmi les 3 portes.
- boat : indique le numéro du bateau pour ceux qui ont pu être dans un bateau lors du naufrage.
- Name : le nom de chaque passager.
- Body : numéro de chacun trouve, NA si le corps d'un passager n'est pas trouvé.

Chapitre 3

Nettoyage de la base de données

Pour cette partie, nous allons importer la base de données, vérifier sa structure et détecter ce qu'on peut corriger à fin de la preparation de notre base de données à la partie d'analyse.
Pour importer la base de données on écrit la ligne suivante :
- file.choose() : Permettre de choisir un fichier de manière interactive.

```
# Les changement seront faits sur l'obejt "DATA".
# L'objet "TITANIC" va servir pour faire des comparaisons

TITANIC<-read.table(file.choose(),header = TRUE, sep = ",")

DATA<-read.table(file.choose(),header = TRUE, sep = ",")

View(DATA)
```

Avec la fonction **Str** et on vérifie la structure de l'objet DATA et les types de chaque variable. On obtient les résultats suivants :

```
> str(DATA)
'data.frame':   1310 obs. of  14 variables:
 $ pclass   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ survived : int  1 1 0 0 0 1 1 0 1 0 ...
 $ name     : Factor w/ 1308 levels "","Abbing, Mr. Anthony",..: 23 25 26 27 28 32 47 48
 52 56 ...
 $ sex      : Factor w/ 3 levels "","female","male": 2 3 2 3 2 3 2 3 2 3 ...
 $ age      : num  29 0.917 2 30 25 ...
 $ sibsp    : int  0 1 1 1 1 0 1 0 2 0 ...
 $ parch    : int  0 2 2 2 2 0 0 0 0 0 ...
 $ ticket   : Factor w/ 930 levels "","110152","110413",..: 189 51 51 51 51 126 94 17 78
 827 ...
 $ fare     : num  211 152 152 152 152 ...
 $ cabin    : Factor w/ 187 levels "","A10","A11",..: 45 81 81 81 81 151 147 17 63 1 ...
 $ embarked : Factor w/ 4 levels "","C","Q","S": 4 4 4 4 4 4 4 4 4 2 ...
 $ boat     : Factor w/ 28 levels "","1","10","11",..: 13 4 1 1 1 14 3 1 28 1 ...
 $ body     : int  NA NA NA 135 NA NA NA NA NA 22 ...
 $ home.dest: Factor w/ 370 levels "","?Havana, Cuba",..: 310 232 232 232 232 238 163 25
 23 230 ...
```

L'objet crée est du type DATA FRAME, on remarque que la variable ticket est de

type facteur alors que c'est plus logique que cette variable soit du type numérique, même remarque pour la variable boat . Pour obtenir plus d'informations sur la base de données, on utilise la fonction Summary qui nous donne les résultats suivants :

```
> summary(DATA)
     pclass          survived                                      name           sex
 Min.   :1.000   Min.   :0.000   Connolly, Miss. Kate                :   2          :  1
 1st Qu.:2.000   1st Qu.:0.000   Kelly, Mr. James                    :   2   female:466
 Median :3.000   Median :0.000                                       :   1   male  :843
 Mean   :2.295   Mean   :0.382   Abbing, Mr. Anthony                 :   1
 3rd Qu.:3.000   3rd Qu.:1.000   Abbott, Master. Eugene Joseph       :   1
 Max.   :3.000   Max.   :1.000   Abbott, Mr. Rossmore Edward         :   1
 NA's   :1       NA's   :1       (Other)                             :1302
      age              sibsp            parch           ticket          fare
 Min.   : 0.1667   Min.   :0.0000   Min.   :0.000   CA. 2343:  11   Min.   :  0.000
 1st Qu.:21.0000   1st Qu.:0.0000   1st Qu.:0.000   1601    :   8   1st Qu.:  7.896
 Median :28.0000   Median :0.0000   Median :0.000   CA 2144 :   8   Median : 14.454
 Mean   :29.8811   Mean   :0.4989   Mean   :0.385   3101295 :   7   Mean   : 33.295
 3rd Qu.:39.0000   3rd Qu.:1.0000   3rd Qu.:0.000   347077  :   7   3rd Qu.: 31.275
 Max.   :80.0000   Max.   :8.0000   Max.   :9.000   347082  :   7   Max.   :512.329
 NA's   :264       NA's   :1        NA's   :1       (Other) :1262   NA's   :2
             cabin      embarked      boat           body
                :1015    :  3           :824   Min.   :  1.0
 C23 C25 C27    :   6   C:270    13     : 39   1st Qu.: 72.0
 B57 B59 B63 B66:   5   Q:123    C      : 38   Median :155.0
 G6             :   5   S:914    15     : 37   Mean   :160.8
 B96 B98        :   4            14     : 33   3rd Qu.:256.0
 C22 C26        :   4            4      : 31   Max.   :328.0
 (Other)        : 271            (Other):308   NA's   :1189
               home.dest
                   :565
 New York, NY      : 64
```

Les résultats de cette fonction nous donnent des statistiques descriptives pour chaque variable, or on remarque que quelques variables ont des valeurs manquantes, voir beaucoup ce qui est le cas pour le variable body qui a 1189 valeurs manquantes. avoir des valeurs manquantes perturbe l'analyse de la base de données.

CONCLUSION : D'apres nos analyses il faut

1. Trouver une solution pour les valeurs manquantes.

2. Affecter le bon type à chaque valeur.

3.1 Manipulation de la base de données

Pour la manipulation des données il faut utiliser les librairies suivantes :

- **Dplyr**
- **Vim**
- **Mice**

3.1.1 Valeurs manquantes

La base de données Titanic admet beaucoup de valeurs manquantes, est-ce qu'il faut les supprimer, ou bien les remplacer ? les remplacer avec quoi et comment ?.
Pour répondre à ces questions, il faut voir le pourcentage des valeurs manquantes dans l'Échantillon, si ce pourcentage dépasse 5% dans ce cas il faut remplacer les valeurs manquantes, sinon il faut les supprimer. Pour cela on va utiliser la fonction **md.pattern**, et ensuite on va utiliser un graphique ce qui va nous montrer plus clairement le pourcentage des valeurs manquantes pour chaque variable.

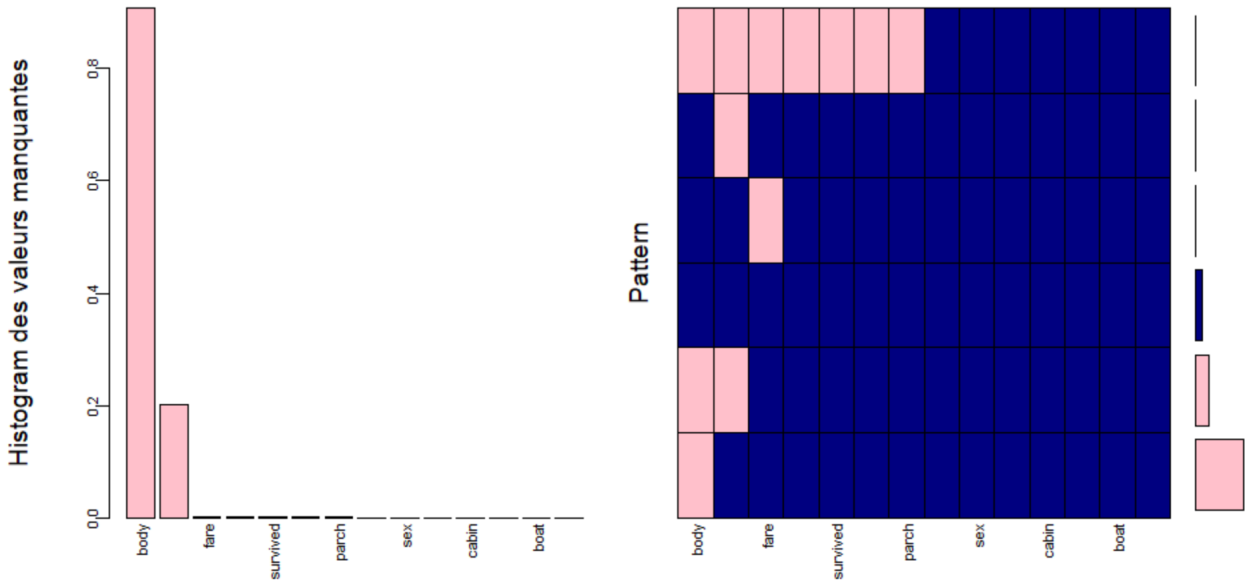
```
> md.pattern(DATA)
     name sex ticket cabin embarked boat home.dest pclass survived sibsp parch fare age body
119     1   1      1     1        1    1         1      1        1     1     1    1   1    1    0
926     1   1      1     1        1    1         1      1        1     1     1    1   1    0    1
1       1   1      1     1        1    1         1      1        1     1     1    1   0    1    1
262     1   1      1     1        1    1         1      1        1     1     1    1   0    0    2
1       1   1      1     1        1    1         1      1        1     1     1    0   1    1    1
1       1   1      1     1        1    1         1      0        0     0     0    0   0    0    7
        0   0      0     0        0    0         0      1        1     1     1    2 264 1189 1459
```

Les résultats se lisent de cette manière
- 119 passagers (119 observations) n'ont aucune valeur manquante.
- 926 observations n'ont pas le numéro de corps (donc elles ont "NA" comme valeur pour la variable body).
- On a pas d'information sur l'age ni sur le corps de 262 passagers.

Pour mieux visualiser les valeurs manquantes pour chaque variable on écrit les lignes de codes suivantes :

```
NAPlot <- aggr(DATA, col=c('navyblue','pink'), numbers=TRUE, sortVars=TRUE,
               labels=names(data), cex.axis=.7, gap=3,
               ylab=c("Histogram des valeurs manquantes","Pattern"))
```

On obtient les résultats suivants :




```
> NAPlot <- aggr(DATA, col=c('navyblue','pink'), numbers=TRUE, sortVars=T
+                labels=names(data), cex.axis=.7, gap=3,
+                ylab=c("Histogram des valeurs manquantes","Pattern"))

 Variables sorted by number of missings:
  Variable        Count
      body 0.9076335878
       age 0.2015267176
      fare 0.0015267176
    pclass 0.0007633588
  survived 0.0007633588
     sibsp 0.0007633588
     parch 0.0007633588
      name 0.0000000000
       sex 0.0000000000
    ticket 0.0000000000
     cabin 0.0000000000
  embarked 0.0000000000
      boat 0.0000000000
 home.dest 0.0000000000
```

On remarque que les variables name, sex, ticket, cabin et embarked n'ont pas de valeurs manquantes La variable age a 20% de valeurs manquantes, la variable body a 90% de valeurs manquantes.
Ce qui dépasse largement 5% donc on va pas supprimer les valeurs manquantes mais on va les remplacer.
L'agorithme KNN pour remplacer les valeurs manquantes Pour avoir des résultats précis on va utiliser l'algorithme de KNN pour remplacer les valeurs manquantes.

On utilisera le code suivant, et on remplacera les valeurs manquantes des variables age et fare.

```
DATA<-kNN(DATA,variable=c("age","fare"),k=5)
```

Pour la variable body on va créer une variable have body qui aura 1 comme valeur si on a trouvé le corps du passager et 0 sinon. car si on utilise un algorithme pour remplacer les valeurs manquantes, il se peut que deux passagers aient le même numéro de corps, ce qui n'est pas logique. Pour cela on utilise la ligne de code suivante :

```
#Création de la variable have_body
DATA<-mutate(DATA,have_body=if_else(DATA$body!="NA",1 ,0))
#Remplacer les Na dans la variable have_body par des zéros
DATA$have_body[is.na(DATA$have_body)] <- 0
```

On vérifie s'il reste des valeurs manquantes , on trouve :

```
Variables sorted by number of missings:
  Variable        Count
      body 0.9076335878
    pclass 0.0007633588
  survived 0.0007633588
     sibsp 0.0007633588
     parch 0.0007633588
      name 0.0000000000
       sex 0.0000000000
       age 0.0000000000
    ticket 0.0000000000
      fare 0.0000000000
     cabin 0.0000000000
  embarked 0.0000000000
      boat 0.0000000000
 home.dest 0.0000000000
have_cabin 0.0000000000
 have_body 0.0000000000
 have_boat 0.0000000000
   age_imp 0.0000000000
  fare_imp 0.0000000000
```

On a plus de valeurs manquantes, donc à ce stade on peut appliquer les fonctions de R pour faire des statistiques descriptives, appliquer des tests statistiques, ou faire des graphiques.

3.1.2 Conversion de type de variables

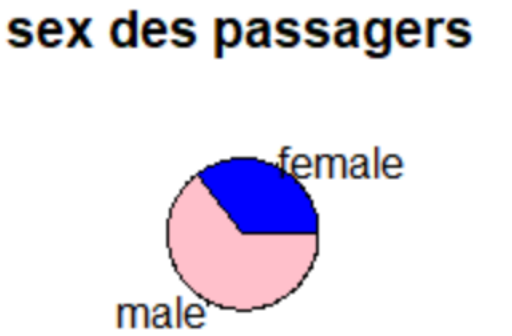
On a vu qu'il y a des variables qui n'ont pas le bon type, ce qui peut causer un dérangement lors de l'étape de l'analyse pour cela, On va régler ce problème avec les lignes de codes suivantes :

```
DATA = transform ( DATA, boat = as.numeric(boat), cabin = as.numeric(cabin))
```

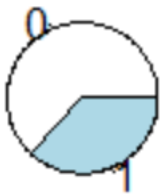
Chapitre 4

Data visualisation

Le nettoyage de données va nous permettre de faire l'analyse de notre base de données. Pour cette partie on aura besoin de la librairie *ggplot2*. On commence par visualiser chaque variable toute seul avec les fonctions usuel de R comme la fonction Pie ou la fonction hist. **On va essayer de répondre à des questions en utilisant ces graphes. Est ce qu'il y avait plus d'hommes parmis les passagers ?**



On remarque que la majorité des passagers était des femmes.
On peut se demander alors des questions par rapports aux femmes seulement, pour cela on peut extraire une sous table où la variable sexe est égale à "female".

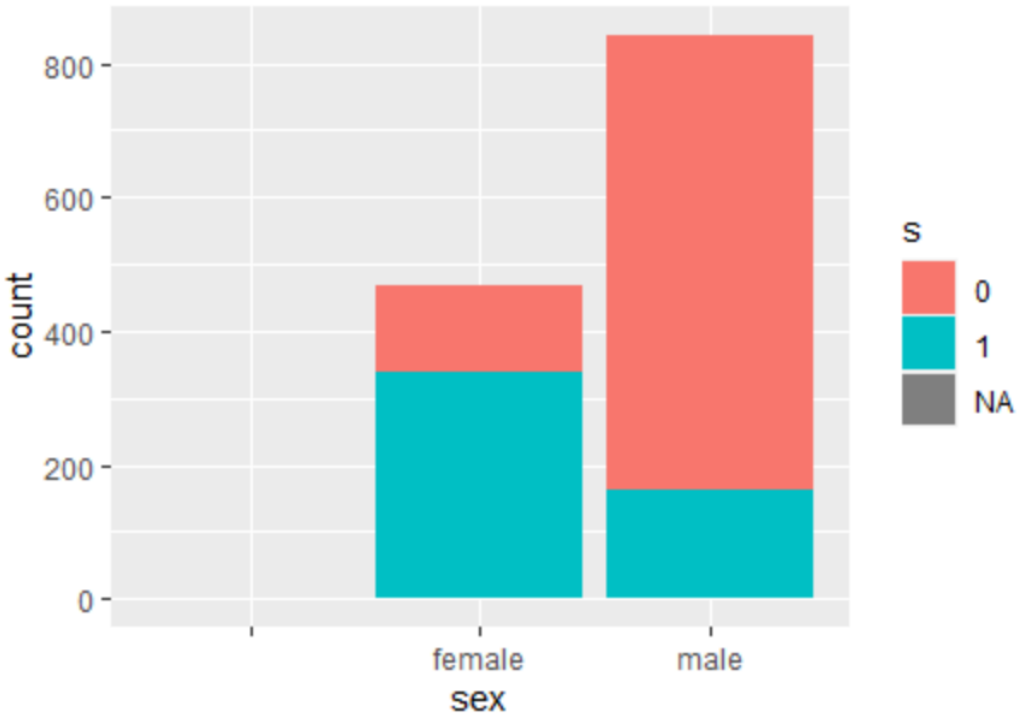


On remarque que la majorité des passagers n'ont pas pu trouver un bateau pour s'en sortir.
Ce qui serait intéressent c'est de faire des analyses sur plusieurs variables en même temps.
On peut se poser la question c'est quoi le sexe qui a pu vivre lors du naufrage ? Donc pour cela on va croiser la variable sexe avec la variable survived.
Pour cela on utilise le code suivant :

```
#VARIABLE SEX EN FONCTION DES VIVANTS

ggplot(data = DATA) +
  geom_bar(mapping = aes(x = sex, fill = s))
ggplot(data = DATA) +
  geom_bar(mapping = aes(x = sex, fill = s), position = "dodge")
```

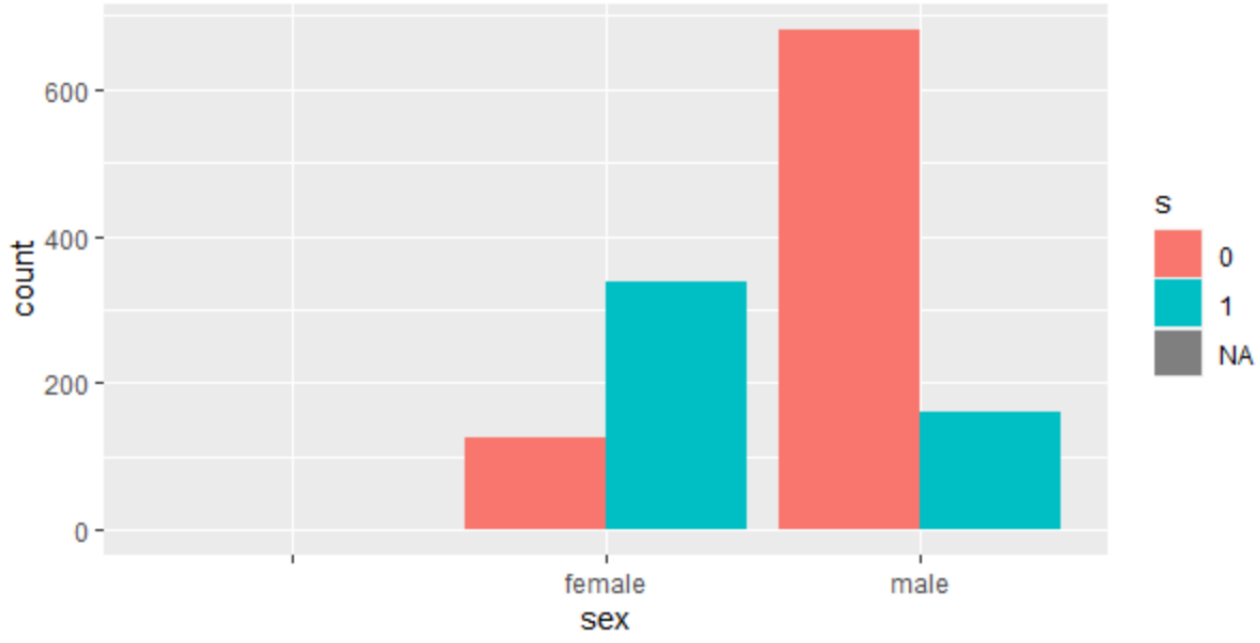
et on obtient le résultat suivant :



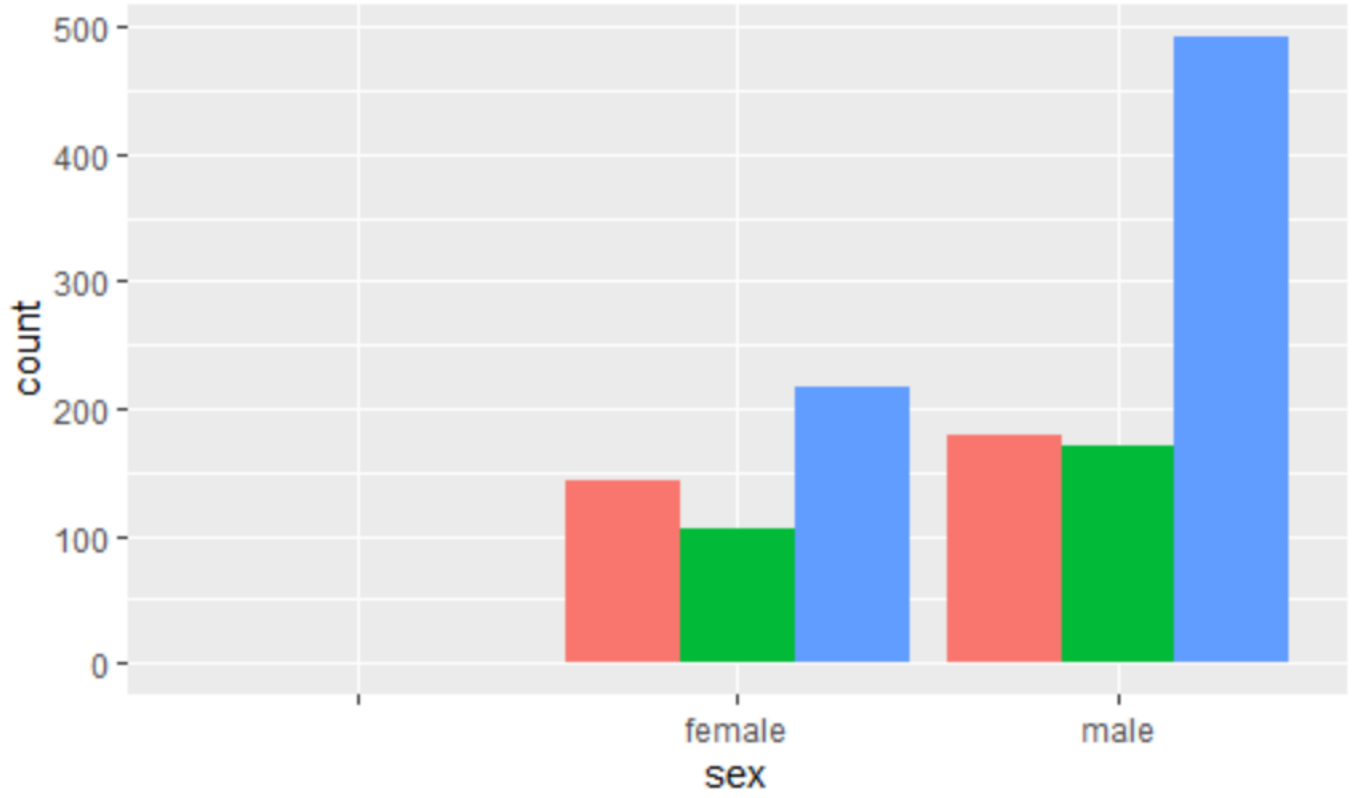
On peut voir que pour les femmes, il a moins de femmes mortes que de femmes qui ont été sauvées. Alors que pour les hommes c'est le contraire, on peut faire une comparaison entre les deux sexes pour cela on écrit le code suivant :

```
ggplot(data = DATA) +
  geom_bar(mapping = aes(x = sex, fill = s), position = "dodge")
```

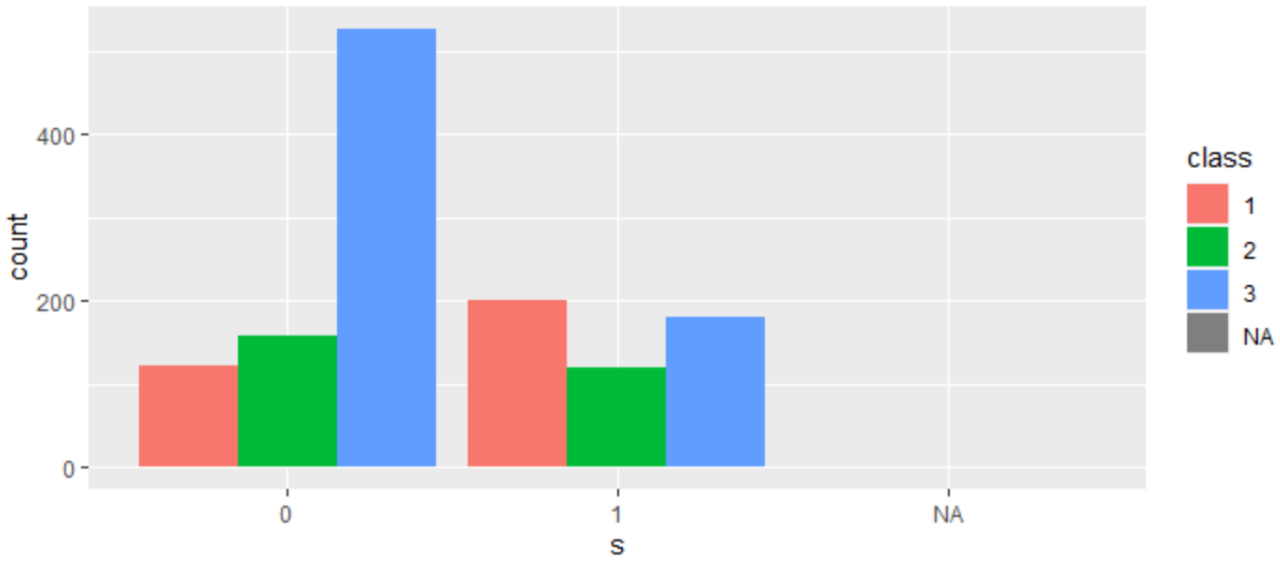
On trouve le résultats suivant :



Ce graphique est plus lisible et nous montre une information de plus, on peut comparer le nombre de morts entre femmes et hommes, on peut voir que les femmes ont été sauvées plus que les hommes. On peut alors voir la distribution des femmes et hommes dans les différents classes du bateau.



On peut dire que la majorité des femmes sont dans classe 3, ce qui est le cas pour les hommes, mais les femmes étaient moins nombreuses à la 3eme class. En croisant la variable class et la variable survived on trouve :



On peut voir que la majorité de ceux qui sont morts sont des passagers de la troixème classe, et que la majorité de ceux qui ont été sauvés sont des passagers de la première classe. On peut donc combiner les résultats qu'on a obtenu pour dire, que les femmes de la première classe avaient plus de chance d'être sauvées.

4.1 Relation entre les variables

	pclass	survived	age	sibsp	parch
pclass	1.00000000	-0.31246936	-0.45105469	0.06083201	0.01832220
survived	-0.31246936	1.00000000	-0.03219644	-0.02782512	0.08265957
age	-0.45105469	-0.03219644	1.00000000	-0.25213420	-0.11990921
sibsp	0.06083201	-0.02782512	-0.25213420	1.00000000	0.37358719
parch	0.01832220	0.08265957	-0.11990921	0.37358719	1.00000000
fare	-0.55874049	0.24447868	0.18504519	0.16038826	0.22166767
boat	-0.29725057	0.75515022	0.01351758	-0.03695735	0.04368396

	fare	boat
pclass	-0.5587405	-0.29725057
survived	0.2444787	0.75515022
age	0.1850452	0.01351758
sibsp	0.1603883	-0.03695735
parch	0.2216677	0.04368396
fare	1.0000000	0.22480011
boat	0.2248001	1.00000000