Collège de Bois-de-Boulogne

TP1 - Fouilles de données
Prédiction de survie sur le Titanic

Préparation et exploration préliminaires des données

Manipulation 1 : Mise en place de l'environnement R et identification des paramètres
Manipulation 2 : Charger les données dans l'environnement de travail R
Manipulation 3 : Nettoyage et Transformation des données
Manipulation 4 : Exploration préliminaire des variables

Préparé par : Nesrine Zemirli

Date	Version	Changement
31 Mai 2016	1.0	Version initiale

Mise en contexte et présentation du projet

Format : individuel

Jeux de données : on a à notre disposition une source de données issues du challenge Kaggle (https://www.kaggle.com/c/titanic).

Description :

Le naufrage du RMS Titanic est l'un des évènements les plus connus de l'histoire maritime. Le 15 Avril 1912, au cours de son voyage inaugural, le Titanic a coulé après avoir heurté un iceberg, tuant 1502 de 2224 passagers et membres d'équipage. Cette tragédie sensationnelle a choqué la communauté internationale et a conduit à une meilleure réglementation de sécurité pour les navires.

Une des raisons pour que le naufrage a conduit à la perte de la vie était qu'il n'y avait pas assez de canots de sauvetage pour les passagers et l'équipage. Bien qu'il y ait une certaine part de chance à survivre au naufrage, certains groupes de personnes sont plus susceptibles de survivre que d'autres, tels que les femmes, les enfants et la classe supérieure.

On souhaite connaitre les prédictions de survies des passagers du Titanic.

Pour cela, dans ce TP 1, il vous ait demandé de préparer les données et d'effectuer une exploration préliminaire des données.

En particulier, il vous demande d'appliquer une démarche de fouilles de données :

- Sélection et chargement des données
- Nettoyage des données
- Transformation des données
- Exploration préliminaire des variables

Dans votre exploration vous devez prendre en considération des variables descriptives et des critères qualificatifs décrivant leur situation (classe, genre, âge, etc.).

Fichiers de données

Nom du fichier	Formats
train	.csv (59.76 kb)
test	.csv (27.96 kb)

Manipulation 1 : Mise en place de l'environnement R et identification des paramètres

Objectif

- Identifier les variables pertinentes à exploiter
- Créer le script titanic_tp_1.r
- Configurer les modules d'exploration et visualisation sous R

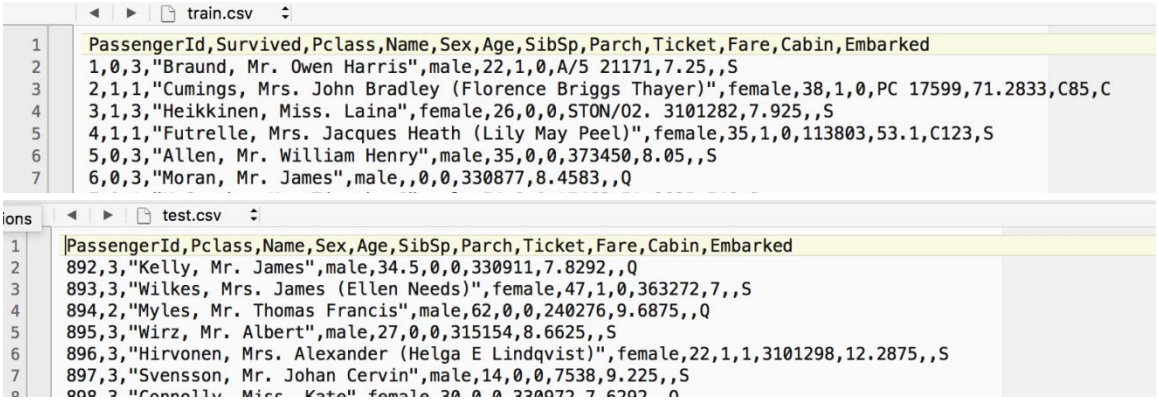
Préliminaire

- R-studio est disponible.

Démarche

1. Le besoin exprimé est de passer en revue les sources de données et la structure des variables

    a. Ouvrir les fichiers train.csv et test.csv avec un éditeur de texte comme le montre la figure suivant:



    b. Passer en revue les fichiers

1. Le fichier **train.csv** :

- **a. Nombre de variable** : 12
- **b. Utilisation** : apprentissage du modèle de prédiction

2. Le fichier **test.csv** :

a. **Nombre de variable :** 11

b. **Utilisation :** test du modèle de prédiction (hypothèses)

c. Décrire les variables

Attribut	Description
passenderId	Identifiant du passager du Titanic
survival	Statue de survie (0 = Non ; 1 = Oui)
pclass	La Class du passager (1 = premier; 2 = second; 3 = troisième)
name	Nom
sex	Genre (femme / homme)
âge	Age
sibsp	Nombre de frères et sœurs /conjoints accompagnant le passager
parch	Nombre de parents / enfants accompagnant le passager
ticket	Numéro du ticket
fare	prix du ticket
cabin	Numéro de cabine
embarked	Port d'Embarcation (C = Cherbourg; Q = Queenstown; S = Southampton)

NOTES SUPLEMENTAIRES :

Pclass : est un indicateur du statut socio-économique (SSE) 1er ~ supérieur ; 2e ~ moyenne ; 3 ~ populaire

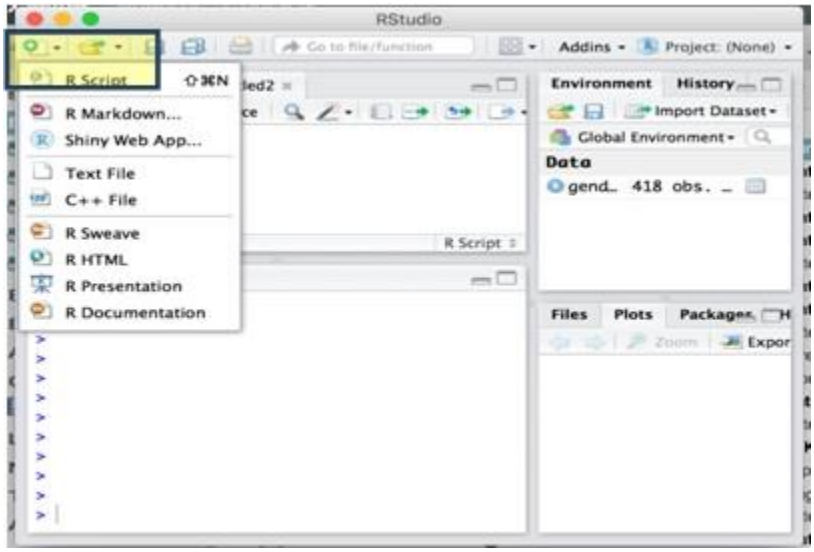
L'âge est en années ; Fractionnel si l'âge inférieur à un (1 an). Si l'âge est estimé, il est sous la forme XX.5

En ce qui concerne les variables sur les relations de la famille entre les passagers (à savoir sibsp et parch) certaines relations ont été ignorées. Voici les définitions utilisées pour sibsp et parch.

Fratrie : Frère, Sœur, demi-frère, ou demi-sœur des passagers à bord du Titanic
Conjoint : Époux ou épouse de passagers à bord du Titanic (Maîtresses et Fiances Ignoré)
Parent : Mère ou père de passagers à bord du Titanic
Enfant : Fils, Fille, beau-fils son ou belle-fille des passagers à bord du Titanic

D'autres parents de la famille exclus de cette étude comprennent cousins, neveux / nièces, oncles / tantes. Certains enfants ont voyagé seulement avec une nounou, donc parch = 0 pour eux. De plus, certains ont voyagé avec des amis très proches ou des voisins dans un village, cependant, les définitions ne prennent pas en charge ces relations.

1. Créer un nouveau script sous R nommé : titanic_tp_1.r
    a. Ouvrir R-studio
    b. Créer un nouveau script R



2. Configuration de l'environnement de fouille de données R.

    a. Vérifier votre répertoire de travail avec l'instruction :

```
getwd()
```

    b. Si le répertoire n'est pas le bon alors, définir un nouveau répertoire de travail pour votre session d'où vous allez récupérer vos données

```
setwd("~/myCoolProject")
```

    Bien que je ne le recommande pas, vous pouvez également utiliser des fichiers du panneau de R-studio pour accéder à un répertoire, puis le définir comme répertoire de travail à partir du menu : --> Set Working

Directory --> Choose Directory. Ou dans le volet Fichiers, choisissez Plus et Définir comme répertoire de travail.

Sinon il y a une meilleure façon. Une façon qui vous permet de gérer votre travail de R comme un expert.

- Créer une variable contenant le chemin absolu de votre répertoire de données

```
csv.folder <- "~/… /Data/"
```

c. Écrire les instructions dans le script titanic_tp_1.r

d. Importer les modules R pour la fouille de données : Manipulation et visualisation des données

```
#Installation des Packages R
install.packages("dplyr")
install.packages("ggplot2")
```

En sortie sur la console, on devrait avoir un résultat similaire au suivant :

```
essai de l'URL
'https://cran.rstudio.com/bin/macosx/mavericks/contrib
/3.3/dplyr_0.4.3.tgz'
Content type 'application/x-gzip' length 4771657 bytes
(4.6 MB)
==================================================
downloaded 4.6 MB

The downloaded binary packages are in
/var/folders/36/36n173zn4b3cqg29w2_f84jr0000gn/T//Rtmp
NCm6Br/downloaded_packages
>
```

```
essai de l'URL
'https://cran.rstudio.com/bin/macosx/mavericks/contrib
/3.3/ggplot2_2.1.0.tgz'
Content type 'application/x-gzip' length 2009223 bytes
(1.9 MB)
==================================================
downloaded 1.9 MB
```

```
The downloaded binary packages are in
/var/folders/36/36n173zn4b3cqg29w2_f84jr0000gn/T//Rtmp
NCm6Br/downloaded_packages
>
```

e. Inclure les librairies

```
# inclure les  librairies

library(dplyr)  # transformation
library (ggplot2) #  visualisation
```

Manipulation 2 : Chargement des données et visualisation des jeux de données dans R-studio

Objectif

- Importation des fichiers de données dans l'environnement R-studio
- Visualiser les données dans l'environnement de données globales de R-studio
- Afficher les données dans l'interface R-studio

Préliminaire

- R-studio est disponible.

Démarche

1. Chargez le fichier de données d'apprentissage : train.csv

```
train <- read.csv ("train.csv",  header= TRUE)
```

Vous pouvez également utiliser l'instruction suivante :

```
train<- read.csv(file = paste0(csv.folder,"train.csv"))
```

2. Vérifier le data frame train

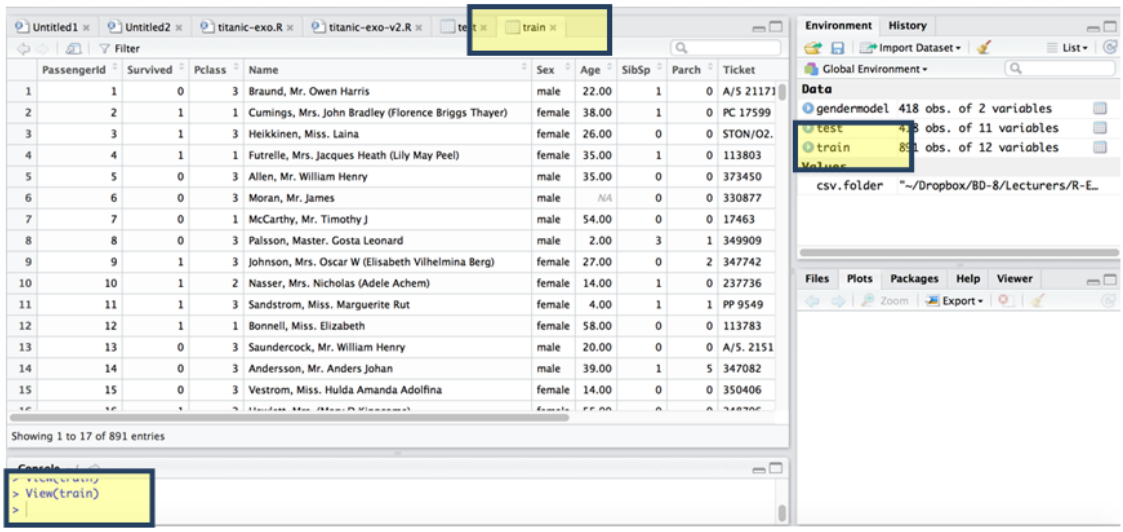
```
str(train)
```

3. En sortie sur la console, on devrait avoir un résultat similaire au suivant :

```
## 'data.frame':    891 obs. of  12 variables:
## $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
## $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
## $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520
629 417 581 ...
## $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
## $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
## $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396
345 133 ...
## $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
```

```
## $ Cabin     : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ..
## $ Embarked  : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

4. Visualiser les données dans l'interface de R-studio



5. Chargez le fichier de données de test : test.csv

```
test <- read.csv ("test.csv",  header= TRUE)
```

Vous pouvez également utiliser l'instruction suivante :

```
test <- read.csv(file = paste0(csv.folder,"test.csv"))
```

6. Vérifier le data frame train

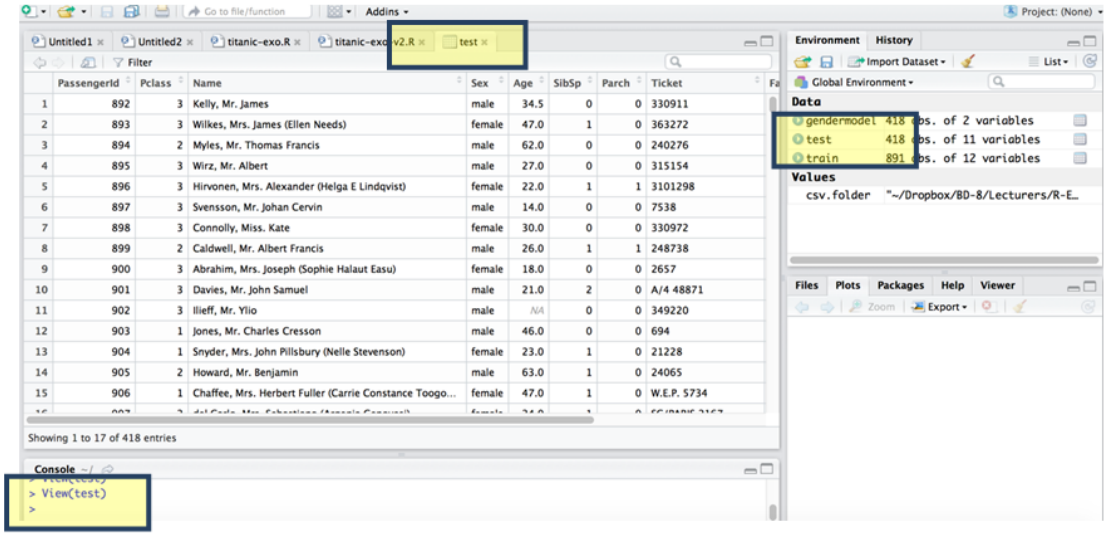
```
str(test)
```

7. En sortie sur la console, on devrait avoir un résultat similaire au suivant :

```
'data.frame': 418 obs. of  11 variables:
 $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
 $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
 $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182
370 85 58 5 104 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
 $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
```

```
$ SibSp     : int  0 1 0 0 1 0 0 1 0 2 ...
$ Parch     : int  0 0 0 0 1 0 0 1 0 0 ...
$ Ticket    : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101
270 ...
$ Fare      : num  7.83 7 9.69 8.66 12.29 ...
$ Cabin     : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
$ Embarked  : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

8. Visualiser les données dans l'interface de R-studio



Manipulation 3 : Nettoyage et Transformation des données

Objectif

- Transformer les données en appliquant les fonctions du module R ***dplyr*** pour les grammaires de la manipulation de données, ce qui permet de travailler avec la trame de données comme des objets.

Préliminaire

- R-studio est disponible.

Démarche

1. Modifier la variable **Survived** dans le frame de données **train** du type **int** au type **factor** avant de combiner les frames de données

```
train$Survived<- factor(train$Survived)
```

2. Créer une variable fictive **Survived** dans le frame de données **test** avant de combiner les frames de données

```
test<- mutate(test, Survived = "none")
```

3. Modifier les frames de données **train** et **test** en ajoutant une variable pour le tri nommée **dataset** avant de combiner les frames de données

```
test <- mutate(test, dataset = "testset")
train <- mutate(train, dataset = "trainset")
```

4. Combiner les données des frames train et test dans le frame **titanic.combined** en préparation à l'exploration des données

```
titanic.combined <- rbind(test, train)
```

5. Vérifier la structure du frame de donnée titanic.combined

```
str(titanic.combined)
```

6. En sortie sur la console, on devrait avoir un résultat similaire au suivant :

```
## 'data.frame':    1309 obs. of  13 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : Factor w/ 1307 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182
370 85 58 5 104 ...
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : Factor w/ 929 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101
270..
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : Factor w/ 187 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "C","Q","S","": 2 3 2 3 3 3 2 3 1 3 ...
##  $ Survived   : chr  "none" "none" "none" "none" ...
##  $ dataset    : chr  "testset" "testset" "testset" "testset" ...
```

7. Renommer et créer des frames de données locales pour plus de simplicité

```
data<- tbl_df (titanic.combined)
```

8. Factoriser les variables **Pclass, dataset** et **Survived**

```
data$Pclass <- factor(data$Pclass)
data$dataset <- factor(data$dataset)
data$Survived<- factor(data$Survived)
```

9. Vérifiez l'existence de doublons

```
IDdups <- distinct(data, PassengerId)
dim(IDdups)
```

```
Namedups <- distinct(data, Name)
dim(Namedups)
```

10. En sortie sur la console, on devrait avoir un résultat similaire au suivant :
    a. Sortie 1

```
## [1] 1309   1
```

    b. Sortie 2

```
## [1] 1307   1
```

11. Puisqu'il n'y a que 1307 noms distincts dans l'ensemble de données, il peut y avoir 2 doublons. Cependant, il y a 1309 ID de passagers distincts.

    a. Il faut filtrer les données en recherchant les valeurs doubles comme suit :

```
filter(data, duplicated(Name))
```

    b. En sortie sur la console, on devrait avoir un résultat similaire au suivant :

```
Source: local data frame [2 x 13]
PassengerId Pclass             Name    Sex   Age SibSp Parch Ticket  Fare  Cabin Embarked  datset
(int) (fctr)                   (fctr) (fctr) (dbl) (int) (int) (fctr) (dbl) (fctr)   (fctr)   (fctr)
1       290     3 Connolly, Miss. Kate female    22     0     0 370373  7.75            Q trainset
2       697     3    Kelly, Mr. James   male    44     0     0 363592  8.05            S trainset
Variables not shown: Survived (fctr)
```

    c. Supprimer les valeurs doubles

```
filter(data, grepl('Kelly|Connolly', Name, Age ))
```

    d. En sortie sur la console, on devrait avoir un résultat similaire au suivant :

```
## Source: local data frame [7 x 13]
##   PassengerId Pclass                          Name    Sex   Age
##         (int) (fctr)                        (fctr) (fctr) (dbl)
## 1       892     3                   Kelly, Mr. James   male  34.5
## 2       898     3                 Connolly, Miss. Kate female  30.0
## 3       290     3                 Connolly, Miss. Kate female  22.0
## 4       301     3 Kelly, Miss. Anna Katherine "Annie Kate" female    N
## 5       574     3                   Kelly, Miss. Mary female    NA
## 6       697     3                   Kelly, Mr. James   male  44.0
## 7       707     2          Kelly, Mrs. Florence "Fannie" female  45.0
## Variables not shown: SibSp (int), Parch (int), Ticket (fctr), Fare (dbl),
##   Cabin (fctr), Embarked (fctr), Survived (fctr), dataset (fctr)
```

Manipulation 3 : Exploration de données

Objectif

- Explorer les données obtenues à partir de la phase de nettoyage et de transformation
- Obtenir un certain nombre de paramètres décrivant des statistiques sur la collection de données
- Visualiser à l'aide de graphique certaines variables importantes

Préliminaire

- R-studio est disponible.

Démarche

Partie 1- Statistique globale sur les données

1. Afficher des statistiques descriptives des données

```
summary(tbl_df(data))
```

```
 PassengerId   Pclass                                          Name         Sex          Age
Min.   :   1   1:323   Connolly, Miss. Kate                     :   2   female:466   Min.   : 0.17
1st Qu.: 328   2:277   Kelly, Mr. James                         :   2   male  :843   1st Qu.:21.00
Median : 655   3:709   Abbott, Master. Eugene Joseph            :   1                Median :28.00
Mean   : 655           Abelseth, Miss. Karen Marie              :   1                Mean   :29.88
3rd Qu.: 982           Abelseth, Mr. Olaus Jorgensen            :   1                3rd Qu.:39.00
Max.   :1309           Abrahamsson, Mr. Abraham August Johannes:   1                Max.   :80.00
                       (Other)                                  :1301                NA's   :263
    SibSp           Parch           Ticket         Fare                    Cabin      Embarked
Min.   :0.0000   Min.   :0.000   CA. 2343:  11   Min.   :  0.000                :1014   C:270
1st Qu.:0.0000   1st Qu.:0.000   1601    :   8   1st Qu.:  7.896   C23 C25 C27  :   6   Q:123
Median :0.0000   Median :0.000   CA 2144 :   8   Median : 14.454   B57 B59 B63 B66:   5   S:914
Mean   :0.4989   Mean   :0.385   3101295 :   7   Mean   : 33.295   G6           :   5   :  2
3rd Qu.:1.0000   3rd Qu.:0.000   347077  :   7   3rd Qu.: 31.275   C22 C26      :   4
Max.   :8.0000   Max.   :9.000   PC 17608:   7   Max.   :512.329   C78          :   4
                                 (Other) :1261   NA's   :1         (Other)      : 271

    dataset     Survived
testset :418   0   :549
trainset:891   1   :342
               none:418
```

2. En sortie sur la console, on devrait avoir un résultat similaire au suivant :

Un tbl_df de trame de données encapsule une trame de données locale. Le principal avantage d'utiliser un tbl_df sur une trame régulière de données est l'impression :

tbl objets n'impriment que quelques lignes et toutes les colonnes qui tiennent sur un seul écran, décrivant le reste sous forme de texte.

3. Afficher une partie une partie des données sur la console

```
head(data)
```

4. En sortie sur la console, on devrait avoir un résultat similaire au suivant :

```
Source: local data frame [6 x 13]

  PassengerId Pclass                                          Name    Sex   Age SibSp Parch  Ticket    Fare
        (int) (fctr)                                         (fctr) (fctr) (dbl) (int) (int)  (fctr)   (dbl)
1         892      3                              Kelly, Mr. James   male  34.5     0     0  330911  7.8292
2         893      3                  Wilkes, Mrs. James (Ellen Needs) female  47.0     1     0  363272  7.0000
3         894      2                        Myles, Mr. Thomas Francis   male  62.0     0     0  240276  9.6875
4         895      3                              Wirz, Mr. Albert   male  27.0     0     0  315154  8.6625
5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female  22.0     1     1 3101298 12.2875
6         897      3                      Svensson, Mr. Johan Cervin   male  14.0     0     0    7538  9.2250
Variables not shown: Cabin (fctr), Embarked (fctr), dataset (fctr), Survived (fctr)
```

- Les valeurs des variables âge et Cabine sont manquants ~ 20% des valeurs
- Fare est manquant 1 valeur
- Embarqué manque 2 valeurs

Partie 2 - Visualisez les caractéristiques de certaines variables potentiellement importantes en fonction de la survie ou pas des passagers

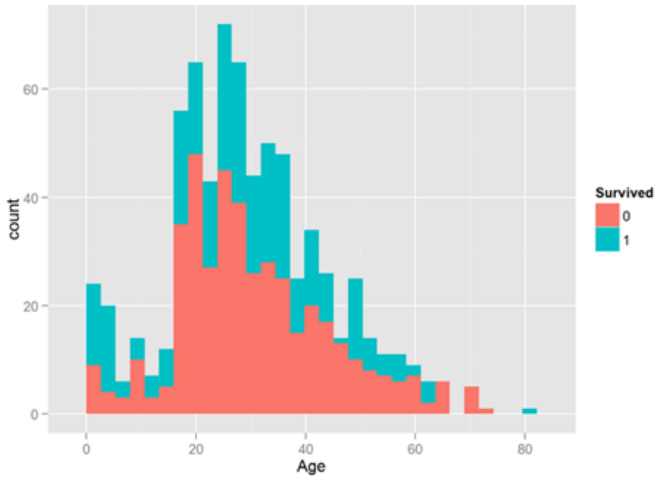
1. La variable Age

a. Afficher l'histogramme de la distribution des chances de survies en fonction de l'âge

```
trainset<-data%>% arrange(dataset)%>%slice(419:1309)
head (trainset)
glimpse(trainset)


hist_Age <- ggplot(trainset, aes(x=Age, fill=Survived))
hist_Age + geom_bar() # affichage par defaut
# pour ajuster l'affichage des histogramme on  peut utiliser
hist_Age + stat_bin(binwidth=2.35) # ou  hist_Age + geom_histogram(binwidth=2.35
```

b. En sortie, on devrait avoir un résultat similaire au suivant :



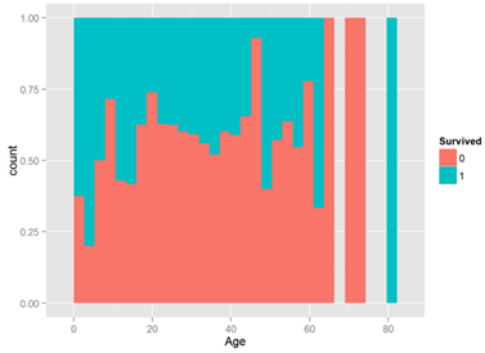
c. Afficher la proportion entre des chances de survies en fonction de l'âge des passagers

```
hist_Age + geom_bar(position= "fill") #proportions

# Ajuster le graphique

hist_Age + stat_bin(binwidth=2.35)
```

d. En sortie, on devrait avoir un résultat similaire au suivant :



e. Comment analyser ces résultats ?

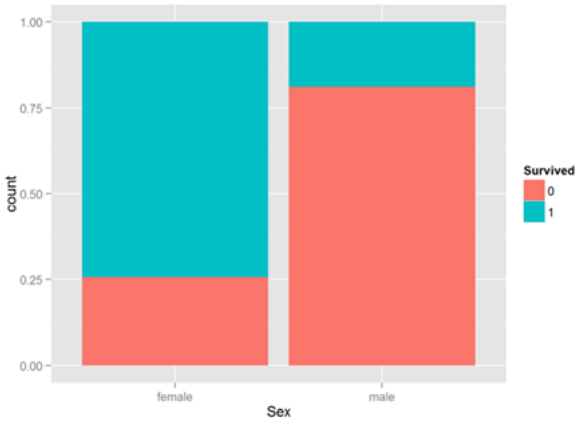
f. Sur la base de ces résultats, et selon votre âge, auriez-vous survécu ou pas au naufrage du Titanic ?

2. La variable sex

a. Afficher la proposition entre des chances de survies en fonction du genre

```
hist_Sex <- ggplot(trainset, aes(x=Sex, fill=Survived))
 hist_Sex + geom_bar(position= "fill") # affichage par defaut
```

b. En sortie, on devrait avoir un résultat similaire au suivant :



c. Comment analyser ces résultats ?

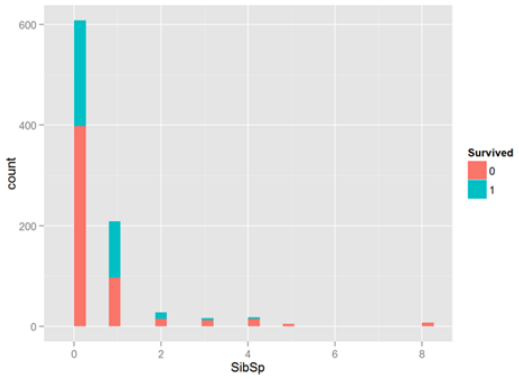
d. Sur la base de ces résultats, et selon votre genre, auriez-vous survécu ou pas au naufrage du Titanic ?

3. La variable SibSp (no frères et sœurs / conjoint)

a. Afficher l'histogramme de la distribution des chances de survies selon les relations familiale

```
hist_SibSp <- ggplot(trainset, aes(x=SibSp, fill=Survived, binwidth = .0005))
 hist_SibSp + geom_bar() # affichage par default
# Affiner l'affichage
hist_SibSp + geom_bar(position= "fill",binwidth=0.5 ) #proportions
```

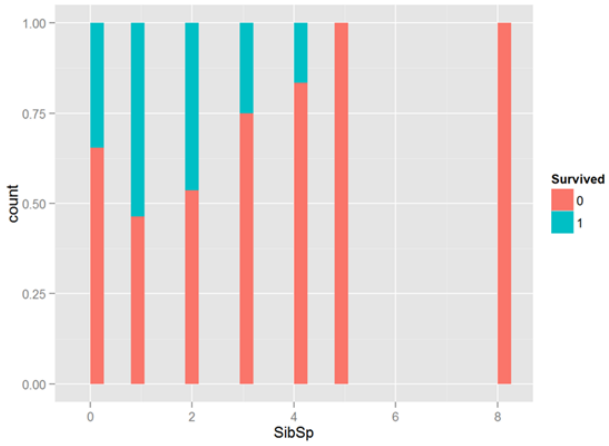
b. En sortie, on devrait avoir un résultat similaire au suivant :



c. Afficher la proposition entre des chances de survies en fonction des relations familiales

```
hist_SibSp + geom_bar(position= "fill", binwidth=0.5 ) #proportions
```

d. En sortie, on devrait avoir un résultat similaire au suivant :



e. Comment analyser ces résultats ?