Data Processing

July 10, 2021

Nettoyage et préparation des données **Auteur** : marshall wilfried

[1]:
```
import csv
import numpy as np
import pandas as pd
from MyModule import Processing
```

0.1 Chargement des données à l'aide du module `pandas`

[2]:
```
d = pd.read_csv('don.csv',encoding='latin-1',delimiter='\"',sep=';')
```

[3]:
```
d.head()
```

[3]:
```
  age;prof;dep.cons;scz.cons;grav.cons;n.enfant;rs;ed;dr
0                              31;autre;0;0;1;2;2;1;1
1                                 49;NA;0;0;2;7;2;2;1
2              50@;prof.intermediaire;0;0;2;2;2;3;2
3                          47;ouvrier;0;0;1;0;2;2;2
4                      23;sans emploi;1;0;2;1;2;2;2
```

Remarque : Les données charger avec pandas n'ont pas un format exploitable. Cela peut etre expliqué par un mauvais formatage des données. dans la suit de ce notebook nous essayerons atravers d'autre approche de fournit un format exploitable de ce jeu de donées en le structurant et en le nettoyant

0.2 Chargement des données à l'aide du module `CSV`

[4]:
```
with open('don.csv') as mon_fichier:
    mon_fichier_reader = csv.reader(mon_fichier, delimiter=';', quotechar='"')
    donnees = [x for x in mon_fichier_reader]
```

[5]:
```
# affichage des 5 premières lignes du jeux de données
donnees[:5]
```

[5]:
```
[['age;prof;dep.cons;scz.cons;grav.cons;n.enfant;rs;ed;dr'],
 ['31;autre;0;0;1;2;2;1;1'],
 ['49;NA;0;0;2;7;2;2;1'],
```

```
['50@;prof.intermediaire;0;0;2;2;2;3;2'],
['47;ouvrier;0;0;1;0;2;2;2']]
```

les données ne sont certes toujours pas dans un format exploitable cependant ils sont dans des objet(`list()`) qu'on sais plus ou moins bien manipuler

0.3 Etape 1

Nous essayerons de separer les elements du jeu de données en colonne et observation contenu de la colonne à l'aide de la methode `split()`

[6]:
```
frame = []
for c in donnees:
    frame.append([j.split(';') for j in c])


df = np.array(frame) # chargement des données spliter dans un object de type␣
 ↪ndarray
df.shape
```

[6]:
```
(800, 1, 9)
```

[ ]:
```

```

[7]:
```
# afffichage des 5 première lignes du jeux de données
df[0:5]
```

[7]:
```
array([[['age', 'prof', 'dep.cons', 'scz.cons', 'grav.cons', 'n.enfant',
         'rs', 'ed', 'dr']],

       [['31', 'autre', '0', '0', '1', '2', '2', '1', '1']],

       [['49', 'NA', '0', '0', '2', '7', '2', '2', '1']],

       [['50@', 'prof.intermediaire', '0', '0', '2', '2', '2', '3', '2']],

       [['47', 'ouvrier', '0', '0', '1', '0', '2', '2', '2']]],
      dtype='<U18')
```

0.4 Etape 2

Organisation des données

Traitement du noms des variable

[8]:
```
df[0] # permet de visualiser le nom des variables
```

[8]:
```
array([['age', 'prof', 'dep.cons', 'scz.cons', 'grav.cons', 'n.enfant',
        'rs', 'ed', 'dr']], dtype='<U18')
```

[9]:
```
name_var = [c for c in df[0][0]]
name_var
```

[9]:
```
['age',
 'prof',
 'dep.cons',
 'scz.cons',
 'grav.cons',
 'n.enfant',
 'rs',
 'ed',
 'dr']
```

construction d'un dictionnaire pour mieux organiser notre jeu de données

[10]:
```
dic = {}
for i in range(0,len(name_var)):
    dic[name_var[i]] = [r[0][i] for r in df[1:df.shape[0]]]
```

[11]:
```
dic.keys()
```

[11]:
```
dict_keys(['age', 'prof', 'dep.cons', 'scz.cons', 'grav.cons', 'n.enfant', 'rs',
'ed', 'dr'])
```

Construction d'un DataFrame

[12]:
```
data = pd.DataFrame(dic)
data.shape
```

[12]:
```
(799, 9)
```

[13]:
```
# afffichage des 10 première lignes du jeux de données
data.head()
```

[13]:

	age	prof	dep.cons	scz.cons	grav.cons	n.enfant	rs	ed	dr
0	31	autre	0	0	1	2	2	1	1
1	49	NA	0	0	2	7	2	2	1
2	50@	prof.intermediaire	0	0	2	2	2	3	2
3	47	ouvrier	0	0	1	0	2	2	2
4	23	sans emploi	1	0	2	1	2	2	2

[14]:
```
proc = Processing(data)
proc.stat_missing_value()
```

==========

```
Statistique données manquante                                  colonnes: 9
------------------------------------------------------------------------------
           missing value % of missing value data dtypes  Obs
age                    0                0.0%      object  799
prof                   0                0.0%      object  799
dep.cons               0                0.0%      object  799
scz.cons               0                0.0%      object  799
grav.cons              0                0.0%      object  799
n.enfant               0                0.0%      object  799
rs                     0                0.0%      object  799
ed                     0                0.0%      object  799
dr                     0                0.0%      object  799
==============================================================================
```

Remarque : On à finalement pu abouti à un jeu de données affiché dans un format plus adapter et organiser cepandant in n'est pas exploitable dans la mesure ou il ne sont pas dans le types approprier et avec des impuretés. dans la suite de ce notebook nous travaillerons à la conversion des données dans le format approprier apres les avoir nettoyer

0.5 Etape 3

Traitement et nettoyage de données

traitement de la variables age

[15]:
```
list(data.age[:15])
```

[15]:
```
['31',
 '49',
 '50@',
 '47',
 '23',
 '34',
 '24',
 '52',
 '42',
 '45',
 '31',
 '',
 '21##',
 '40',
 '64']
```

[16]:
```
# identification des intrus
num = [str(i) for i in range(0,10)]
intrus = [j for c in data.age for j in c if j not in num]
```

[17]:
```
# nettoyage de la variable age
age = []
for c in data.age:
    for j in intrus:
        if j in c:
            c = c.replace(j,'')
    age.append(c)

data.age = age
```

[18]:
```
list(data.age[:15])
```

[18]:
```
['31',
 '49',
 '50',
 '47',
 '23',
 '34',
 '24',
 '52',
 '42',
 '45',
 '31',
 '',
 '21',
 '40',
 '64']
```

Remarque: la varaible age à bel et bien été traité les mauvais caracters on tous été supprimer

[19]:
```
data.prof = data.prof.replace('prof.intermediaire','intermediaire')
```

[20]:
```
data.head()
```

[20]:

	age	prof	dep.cons	scz.cons	grav.cons	n.enfant	rs	ed	dr
0	31	autre	0	0	1	2	2	1	1
1	49	NA	0	0	2	7	2	2	1
2	50	intermediaire	0	0	2	2	2	3	2
3	47	ouvrier	0	0	1	0	2	2	2
4	23	sans emploi	1	0	2	1	2	2	2

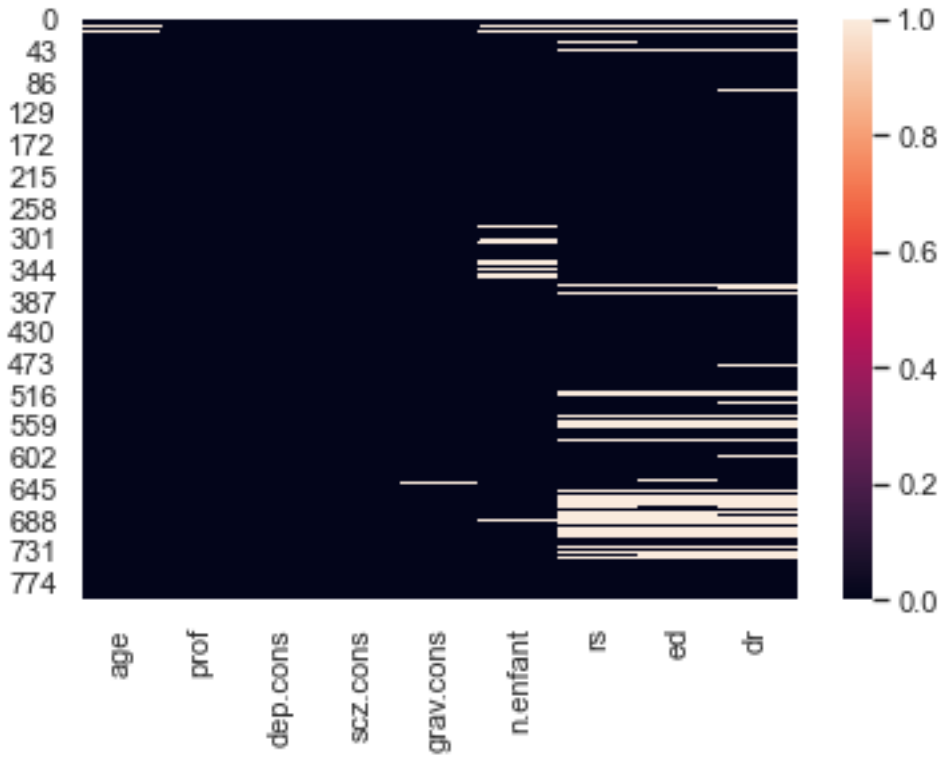
conversion des variables dans le bon type

[21]:
```
for c in data.columns:
    try:
        data[c] = pd.to_numeric(data[c])
    except ValueError as e:
        pass
```

[22]:
```
proc.stat_missing_value()
```

```
================================================================================
Statistique données manquante                                  colonnes: 9
--------------------------------------------------------------------------------
           missing value % of missing value data dtypes  Obs
dr                   111             13.89%     float64  688
ed                   107             13.39%     float64  692
rs                   103             12.89%     float64  696
n.enfant              26              3.25%     float64  773
grav.cons              4               0.5%     float64  795
age                    2              0.25%     float64  797
prof                   0               0.0%      object  799
dep.cons               0               0.0%       int64  799
scz.cons               0               0.0%       int64  799
================================================================================
```

[23]:
```
import seaborn as sns
sns.heatmap(data.isna());
```



remarque: les données on bien été traité et convertir dans le bon type on peut remarquer que l'existance en effet de données manquantes

0.6 Etape 4

Traitemet des données manquante

[24]:
```
data['age'] = data['age'].replace(np.nan,data['age'].mean())
data['age'] = [int(c) for c in data['age']]
```

[25]:
```
for c in data.select_dtypes('float64'):
    data[c] = data[c].replace(np.nan,data[c].value_counts().idxmax())
    data[c] = [int(c) for c in data[c]]
```

[26]:
```
data.head()
```

[26]:

	age	prof	dep.cons	scz.cons	grav.cons	n.enfant	rs	ed	dr
0	31	autre	0	0	1	2	2	1	1
1	49	NA	0	0	2	7	2	2	1
2	50	intermediaire	0	0	2	2	2	3	2
3	47	ouvrier	0	0	1	0	2	2	2
4	23	sans emploi	1	0	2	1	2	2	2

[27]:
```
proc.stat_missing_value()
```

```
================================================================================
Statistique données manquante                              colonnes: 9
--------------------------------------------------------------------------------
           missing value % of missing value data dtypes  Obs
age                    0               0.0%       int64  799
prof                   0               0.0%      object  799
dep.cons               0               0.0%       int64  799
scz.cons               0               0.0%       int64  799
grav.cons              0               0.0%       int64  799
n.enfant               0               0.0%       int64  799
rs                     0               0.0%       int64  799
ed                     0               0.0%       int64  799
dr                     0               0.0%       int64  799
================================================================================
```

[28]:
```
#sauvegade des données
data.to_csv('clean_dataset.csv',index= False)
```