19 ANALYSE DE DONNÉES AVEC PYTHON SERIES ET DATAFRAME

http://www.python-simple.com/python-pandas/panda-intro.php
http://sdz.tdct.org/sdz/comprendre-les-encodages.html
https://riptutorial.com/fr/pandas/example/23811/utiliser--loc
https://www.it-swarm.dev/fr/python/comment-verifier-si-une-valeur-est-nan-dans-un-pandas-dataframe/1052199540/
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.isnull.html

STRUCTURE CLÉ DE PANDAS

Pandas est une librairie python qui permet de manipuler facilement des données à analyser :

- manipuler des tableaux de données avec des étiquettes de variables (colonnes) et d'individus (lignes).
- ces tableaux sont appelés DataFrames.
- on peut facilement lire et écrire ces dataframes à partir ou vers un fichier tabulé.
- on peut facilement tracer des graphes à partir de ces DataFrames grâce à matplotlib.

Pour utiliser pandas : `import pandas`

3 types d'objets dans Pandas :

- Series
- Dataframes : ensemble d'objets Series
- Panels : Ensemble d'objets Dataframe

LES SERIES

a) Correspond à une colonne. Plusieurs Series contenues dans un objet forment un Dataframe.
b) Utilises les valeurs NaN pour gérer les valeurs manquantes
c) Types de données :
    - Float
    - Int
    - Bool
    - Datetime64[ns] : Date et horaire sans la time zone
    - Datetime[ns, tz] : Date et horaire avec la time zone
    - Timedelta[ns] : Différence de date et horaire (seconde, minute, …)
    - Category : pour les variables catégorielles
    - Object : chaîne de caractère.

PROJET GUIDÉ ANALYSE DE DATA DE THANKSGIVING

Ce mini projet portera sur les résultats d'un sondage ayant pour sujet ce que mange les Américains au moment de Thanhsgiving, leurs revenus moyens entre autre choses.

Objectif :
Explorer les données et trouver des tendances ou hypothèses intéressantes.

Les données sont contenues dans le fichier « thanhsgiving.csv ».

1) Le fichier comporte 1059 lignes et 65 colonnes.
2) La 1ère ligne correspond aux questions posées et pourrons servir de noms des colonnes.
3) La première colonne contient un id pour chaque personne interrogé.
4) Pour de nombreuse questions les réponses sont catégorielles (plusieurs choix de réponses possible).

INTRODUCTION AU DATASET

1) Lire le fichier « thanksgiving.csv » avec la librairie pandas et l'assigner à une variable data.
2) Spécifier dans les paramètre de la fonction permettant de lire le fichier « **encoding='latin-1'** » car ce dataset n'est pas encodé normalement.
   Utiliser le noms des colonnes contenu dans la 1ère ligne du fichier.
3) Afficher les premières lignes du dataframe (une méthode en particulier permet de le faire).
4) Afficher le noms des colonnes avec l'attribut columns.

⇒ 1 et 2) **On importe la librairie pandas.**
Data = variable
pd = pandas
read = lire
lire quoi : un csv
ou lire : : l'adresse du csv
encoding : code latin

```
Import pandas as pd
data = pd.read_csv("/home/utilisateur/Documents/COURS/19.2-thanksgiving.csv", encoding = "latin-1")
```

===> pas d'affichage

⇒ **3) Afficher les premières lignes du fichier**
[5 rows x 65 columns]
data = variable

```
print(data.head())
```

```
RespondentID Do you celebrate Thanksgiving?  \
0    4337954960                          Yes
1    4337951949                          Yes
2    4337935621                          Yes
3    4337933040                          Yes
4    4337931983                          Yes

  What is typically the main dish at your Thanksgiving dinner?  \
0                                             Turkey
1                                             Turkey
2                                             Turkey
3                                             Turkey
4                                           Tofurkey
```

⇒ **4)Afficher le noms des colonnes avec l'attribut columns**

```
print(data.columns)
```

```
Index(['RespondentID', 'Do you celebrate Thanksgiving?',
       'What is typically the main dish at your Thanksgiving dinner?',
       'What is typically the main dish at your Thanksgiving dinner? - Other (please specify)',
       'How is the main dish typically cooked?',
       'How is the main dish typically cooked? - Other (please specify)',
       'What kind of stuffing/dressing do you typically have?',
       'What kind of stuffing/dressing do you typically have? - Other (please specify)',
       'What type of cranberry saucedo you typically have?',
       'What type of cranberry saucedo you typically have? - Other (please specify)',
       'Do you typically have gravy?',
```

'Which of these side dishes aretypically served at your Thanksgiving dinner? Please select all that apply. - Brussel sprouts',
'Which of these side dishes aretypically served at your Thanksgiving dinner? Please select all that apply. - Carrots',
'Which of these side dishes aretypically served at your Thanksgiving dinner? Please select all that apply. - Cauliflower',
'Which of these side dishes aretypically served at your Thanksgiving dinner? Please select all that apply. - Corn',
'Which of these side dishes aretypically served at your Thanksgiving dinner? Please select all that apply.

FILTRER LES DONNÉES

1) Utiliser la méthode Series.values_count() pour afficher le décompte du nombre de réponses pour chacune des modalités de la colonnes « Do you celebrate Thanksgiving? »
2) Filtrer et garder toute les lignes du dataframe pour lesquelles la réponse à la question « Do you celebrate Thanksgiving? » est « Yes ».
3) Assigner ce nouveau dataframe à data et afficher le.

⇒ 1) la methode values_count()
https://www.journaldunet.fr/web-tech/developpement/1441075-python-comment-compter-les-valeurs-uniques-par-groupe-avec-pandas/

```
# Utiliser la méthode Series.values_count() pour afficher le décompte du nombre de réponses pour chacune des modalités de la colonnes « Do you celebrate Thanksgiving?

#..............bdd.....................colonne.....................methode
resultat= (data['Do you celebrate Thanksgiving?'].value_counts())

print(resultat)
```

```
Yes    980
No      78
Name: Do you celebrate Thanksgiving?, dtype: int64
```

⇒ **2) Filtrage de toutes les lignes de cette colonne pour la réponse « yes ».**
Le .loc permet d'accéder aux données avec l'adresse entre crochets et le résultat demandé avec ==.

```
print(data.loc[data['Do you celebrate Thanksgiving?']=="Yes",:])
```

```
     RespondentID Do you celebrate Thanksgiving?  \……+ toutes les colonnes du tableau
0      4337954960                          Yes
1      4337951949                          Yes
2      4337935621                          Yes
3      4337933040                          Yes
4      4337931983                          Yes
...           ...                          ...
1053   4335944082                          Yes
1054   4335943173                          Yes
1055   4335943060                          Yes
1056   4335934708                          Yes
1057   4335894916                          Yes
```

⇒ 3) # Assigner ce nouveau dataframe à data et afficher le.
On reprend le même calcul mais on l'assigne avant.
print sert à la vérification

```
data = data[data['Do you celebrate Thanksgiving?']=="Yes"]
print(data)
```

EXPLORATION DES REPAS DE THANKSGIVING

1) Utiliser la méthode Series.values_count() pour afficher combien de fois chaque résultats apparait pour la question « What is typically the main dish at your Thanksgiving dinner? »
2) Afficher la colonne « Do you typically have gravy? » pour les ligne du dataframe data pour lesquelles la colonne « What is typically the main dish at your Thanksgiving dinner? » vaut « Tofurkey » pour la dinde de tofu.

⇒ 1) Quelles sont les occurrences de « What is typically the main dish at your Thanksgiving dinner? » ?

```
print(data['What is typically the main dish at your Thanksgiving dinner?'].value_counts())
```

```
Turkey                   859
Other (please specify)    35
Ham/Pork                  29
Tofurkey                  20
Chicken                   12
Roast beef                11
I don't know               5
Turducken                  3
Name: What is typically the main dish at your Thanksgiving dinner?, dtype: int64
```

⇒ 2) Filtrer et garder toutes les ligne du dataframe pour lesquelles la réponse à la question 'What is typically the main dish at your Thanksgiving dinner?' = "Tofurkey"

```
print(data.loc[data['What is typically the main dish at your Thanksgiving dinner?']=="Tofurkey",:])
```

```
   RespondentID Do you celebrate Thanksgiving?  \
4      4337931983                          Yes
33     4337771439                          Yes
69     4337553422                          Yes
72     4337540484                          Yes
77     4337490067                          Yes
145    4337191550                          Yes
175    4337139327                          Yes
218    4337078951                          Yes
243    4337044348                          Yes
…
  What is typically the main dish at your Thanksgiving dinner?  \
4                                           Tofurkey
33                                          Tofurkey
69                                          Tofurkey
72                                          Tofurkey
77                                          Tofurkey
145                                         Tofurkey
175                                         Tofurkey
218                                         Tofurkey
243                                         Tofurkey
275                                         Tofurkey
```

EXPLORATION DES DESSERTS POUR THANKSGIVING

On cherche ici à savoir combien de personnes ont consommés des tartes à la pomme, la citrouille ou pécan.

1) • Créer un objet Series indiquant avec des booléens les valeurs de la colonnes « Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Apple » qui sont nulles. Assigner le résultat à la variable « apple_isnull ».
2) Créer un objet Series indiquant avec des booléens les valeurs de la colonnes « Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Pumpkin » qui sont nulles. Assigner le résultat à la variable « pumpkin_isnull ».
3) Créer un objet Series indiquant avec des booléens les valeurs de la colonnes « Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Pecan » qui sont nulles. Assigner le résultat à la variable « pecan_isnull ».
4) Combiner les trois objets Series avec l'operateur « & » et assigné le résultat à la variable « pies ».
5) Afficher les valeurs unique et combien de fois elle apparaissent dans la colonnes de pies

⇒ 1) **Quelles sont les valeurs de la colonnes « Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Apple » qui sont nulles ?**
Afin de vérifier si une valeur est NaN, les fonctions `isnull()` ou `notnull()` peuvent être utilisées.

```
apple_isnull = pd.isnull(data["Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply.  - Apple"])
print(apple_isnull)
```

```
0      False
1      False
2      False
3       True
4      False
      ...
1053    True
1054    True
1055   False
1056    True
1057    True

Name: Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Apple, Length: 980, dtype: bool
```

⇒ **2) Quelles sont les valeurs de la colonnes « Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Pumpkin » qui sont nulles ?**

```
pumpkin_isnull = pd.isnull(data["Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply.  - Pumpkin"])

print(pumpkin_isnull)
```

⇒ **3) Quelles sont les valeurs de la colonnes « Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Pecan » qui sont nulles ?**

```
pecan_isnull = pd.isnull(data["Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply.  - Pecan"])

print(pecan_isnull)
```

⇒ **4) Combiner les trois objets Series avec l'operateur « & » et assigné le résultat à la variable « pies ».**

pies = apple_isnull & pumpkin_isnull & pecan_isnull

pies = pd.isnull(data["Which type of pie is typically served at your Thanksgiving dinner? Please select all that apply. - Pecan"])

⇒ 5) Afficher les valeurs unique et combien de fois elle apparaissent dans la colonnes de pies

```
print(pies.value_counts())
False   876
True     104
dtype: int64
```

CONVERTIR L'ÂGE EN VALEUR NUMÉRIQUE

1. Ecrire une fonction qui converti une chaîne de caractère en une valeur entière. Cela permettra de convertir les valeurs de la colonne « Age » en entiers. Cette fonction prendra en paramètre une chaîne de caractères (les valeurs actuelles de la colonne « Age »)

   - Utiliser la fonction is_null() pour vérifier si les valeurs sont nulles. Ajouter une condition if qui retourne None si la valeur est nulle.
   - Séparer les chaine de caractère en fonction de l'espace (' ') et extraire le 1ère élément de la liste. Supprimer le caractère '+' dans le résultat.
   - Convertir le résultat en entier.
   - Retourner le résultat.

2. Utiliser la méthode Series.apply() pour appliquer la fonction à chaque valeur de la colonne 'Age' du dataframe data.
   Assigner le résultat à la nouvelle colonne 'int_age' du dataframe.

3. Appeler la méthode Series.describe() sur la colonne « int_age » du dataframe data et afficher le résultat.

1) Ecrire une fonction qui converti une chaîne de caractère en une valeur entière pour convertir « Age » en entier.

```
df is_null(val_str):
   if pd.isnull(val_str):
      return None
   liste = str.split(' ')
   liste = l[0]
   liste = l.replace("+", "")
   liste = int(liste)
   return liste
```

Création de la fonction df is_null(str)
pandas.isnull
si c'est nul alors le résultat est « None »
spliter la chaine de caractère
Extraire le 1er élément
Remplacer les + par « pas d'espace)
Transformer liste en integer

2) Appliquer la fonction à chaque valeur de la colonne 'Age' du dataframe data.
Assigner le résultat à la nouvelle colonne 'int_age' du dataframe.

```
data["int_age"] = data["Age"].apply(is_null)
```

Création de la colonne int_age
Application de la fct is_null

```
print(data["int_age"])
```

```
0      18.0
1      18.0
2      18.0
3      30.0
4      30.0
      ...
1053    30.0
1054    60.0
1055    60.0
1056     NaN
1057     NaN
Name: int_age, Length: 1058, dtype: float64
```

CONVERTIR LES REVENUS EN VALEURS NUMÉRIQUES

1) Ecrire une fonction pour convertir les revenus en valeur unique de format entier.
   a) Utiliser la fonction isnull() pour vérifier si la valeur est nulle. Si c'est le cas, retourner « None ».
   b) Séparer la chaine de caractère en prenant l'espace comme délimiteur et extraire le premier élément de la liste résultante.
   c) Si le résultat vaut « Prefer » retourner « None ».
   d) Supprimer les caractères « $ » et « , ».
   e) Utiliser int() pour convertir le résultat en entier.
   f) Retourner le résultat.
2) Utiliser la méthode Series.apply() pour appliquer la fonction précédente à chaque valeur de la colonne « How much total combined money did all members of your HOUSEHOLD earn last year? » du dataframe data.
   a) Assigner le résultat à la nouvelle colonne « int_income » du dataframe data.
3) Appeler la méthode Series.describe() à la colonne int_income du dataframe data et afficher le résultat

1) Fonction pour transformer un string revenus en int revenus sans $ et sans ,

```
def is_null(str):                          Création de la fonction
    if pd.isnull(str):                     Si le résultat est « null »
        return None                        Ne rien retourner
    liste = str.split(' ')                 Splitter la liste
    liste = liste[0]                       Récupérer le 1er caractère de cette liste
    if liste == "Prefer":
        return None
    liste = liste.replace("$", "")         Remplacer les $ par rien
    liste = liste.replace(",", "")         Remplacer les , par rien
    liste = int(liste)                     Transformer la liste en integer
    return liste                           Retourner la liste (du mot)
```

2) Appliquer la fonction is_null à « How much total combined money did all members of your HOUSEHOLD earn last year ».

```
data["int_income"] = data["How much total combined money did all members of your HOUSEHOLD earn last year?"].apply(is_null)

print(data["int_income"])
```

```
0       75000.0
1       50000.0
2           0.0
3      200000.0
4      100000.0
        ...
1053   100000.0
1054    50000.0
1055   100000.0
1056        NaN
1057        NaN
Name: int_income, Length: 1058, dtype: float64
```

3) Appeler la méthode Series.describe() à la colonne int_income du dataframe data et afficher le résultat

```
print(data["int_income"].describe())
```

```
count      889.000000
mean     74077.615298
std      59360.742902
min          0.000000
25%      25000.000000
50%      50000.000000
75%     100000.000000
max     200000.000000
Name: int_income, dtype: float64
```

LIEN ENTRE DISTANCE ET REVENUS

1. Regarder de quel manière les personnages gagnant moins de 150 000 dollars voyagent.
   - Filtrer data en sélectionnant seulement les valeur de « int_income » infèrieures à 150 000
   - Sélectionner la colonne « How far will you travel for Thanksgiving? » en prenant en compte le filtre.
   - Utiliser la méthode value_counts() pour compter combien e fois chaque vaaleur apparait dans la colonne.
   - Afficher le résultats.
2. Faire de même avec les personnages gagnant plus de 150 000 dollars.

1a) # liste des personnes qui ont des revenus < 150 000

```
individu = data[data["int_income"] < 150000]
print(individu)
```

```
  RespondentID Do you celebrate Thanksgiving?  \
0     4337954960                         Yes
1     4337951949                         Yes
2     4337935621                         Yes
4     4337931983                         Yes
5     4337929779                         Yes
...          ...                         ...
1051  4335944854                         Yes
1052  4335944115                          No
1053  4335944082                         Yes
1054  4335943173                         Yes
1055  4335943060                         Yes

     What is typically the main dish at your Thanksgiving dinner?  \
0                                    Turkey
1                                    Turkey
2                                    Turkey
```

1b) Sélectionner la colonne « How far will you travel for Thanksgiving? » en prenant en compte le filtre.

```
print(individu["How far will you travel for Thanksgiving?"].value_counts())
```

```
Thanksgiving is happening at my home--I won't travel at all                    281
Thanksgiving is local--it will take place in the town I live in                203
Thanksgiving is out of town but not too far--it's a drive of a few hours or less
150
Thanksgiving is out of town and far away--I have to drive several hours or fly     55
Name: How far will you travel for Thanksgiving?, dtype: int64
```

2) La même chose mais avec ceux qui gagnent plus de 150 000

2a) # liste des personnes qui ont des revenus < 150 000

```
individu2 = data[data["int_income"] > 150000]
print(individu2)
```

```
  RespondentID Do you celebrate Thanksgiving?  \
3      4337933040                         Yes
15     4337857295                         Yes
16     4337856362                         Yes
25     4337790002                         Yes
39     4337732348                         Yes
...           ...                         ...
982    4335981057                         Yes
983    4335979596                         Yes
993    4335973959                         Yes
1015   4335960288                         Yes
1026   4335957096                         Yes

     What is typically the main dish at your Thanksgiving dinner?  \
3                                    Turkey
15                                   Turkey
16                                Turducken
25                                   Turkey
39                                Ham/Pork
```

2a)

```
print(individu2["How far will you travel for Thanksgiving?"].value_counts())
```

```
Thanksgiving is happening at my home--I won't travel at all                  49
Thanksgiving is local--it will take place in the town I live in              25
Thanksgiving is out of town but not too far--it's a drive of a few hours or less    16
Thanksgiving is out of town and far away--I have to drive several hours or fly      12
Name: How far will you travel for Thanksgiving?, dtype: int64
```

LIEN ENTRE PASSER THANKGIVING ENTRE AMIS AVEC L'ÂGE ET LE REVENUS

1. Générer un pivot de table montrant la moyenne d'âge des sondés pour chaque catégorie des questions « Have you ever tried to meet up with hometown friends on Thanksgiving night? » et « Have you ever attended a "Friendsgiving?" ».
   - Appeler la méthode pivot_table() sur le data frame data.
   - Passer au paramètre « index » la valeur « Have you ever tried to meet up with hometown friends on Thanksgiving night? ».
   - Passer au paramètre « columns » la valeur « Have you ever attended a "Friendsgiving?" ».
   - Passer au paramètre « values » la valeur « int_age »

     Afficher les résultats.
2. • Faire de même avec les revenus avec ces deux questions.

1) Générer un pivot de table montrant la moyenne d'âge des sondés pour chaque catégorie des questions « Have you ever tried to meet up with hometown friends on Thanksgiving night? » et « Have you ever attended a "Friendsgiving?" ».

pivot_table pour créez un tableau croisé dynamique de type feuille de calcul en tant que DataFrame

```
print(data.pivot_table(index = ['Have you ever tried to meet up with hometown friends on Thanksgiving night?'], columns = ['Have you ever attended a "Friendsgiving?"'], values=['int_age'], aggfunc=pd.Series.mean))
```

	int_age	
Have you ever attended a "Friendsgiving?"	No	Yes
Have you ever tried to meet up with hometown fr...		
No	42.283702	37.010526
Yes	41.475410	33.976744

2) • Faire de même avec les revenus avec ces deux questions.

```
print(data.pivot_table(index = ['Have you ever tried to meet up with hometown friends on Thanksgiving night?'], columns = ['Have you ever attended a "Friendsgiving?"'], values=['int_income'], aggfunc=pd.Series.mean))
```

	int_income	
Have you ever attended a "Friendsgiv	No	Yes
Have you ever tried to meet up with hometown fr...		
No	78914.549654	72894.736842
Yes	78750.000000	66019.736842