Notebook_chapitre_5_Pandas_avec_commentaires_numerotation

February 1, 2023

1 1. Lire et écrire des fichiers avec Pandas

1.1 1.1. Lecture de fichiers texte (CSV ou TXT)

1.1.1 1.1.1. Lecture basique d'un fichier

[1]:
```
import pandas as pd
donnees=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv")
donnees.head()
#pour lire un fichier txt on utilisera la fonction read_table("monfichier.txt")
#on obtient un dataframe , un tableau à deux dimensions, avec des lignes et des␣
 ↪colonnes
```

[1]:
```
   ID                      Name Sex   Age  Height  Weight            Team  \
0   1                 A Dijiang   M  24.0   180.0    80.0           China
1   2                  A Lamusi   M  23.0   170.0    60.0           China
2   3       Gunnar Nielsen Aaby   M  24.0     NaN     NaN         Denmark
3   4      Edgar Lindenau Aabye   M  34.0     NaN     NaN  Denmark/Sweden
4   5  Christine Jacoba Aaftink   F  21.0   185.0    82.0     Netherlands

   NOC        Games  Year  Season       City          Sport  \
0  CHN  1992 Summer  1992  Summer  Barcelona     Basketball
1  CHN  2012 Summer  2012  Summer     London           Judo
2  DEN  1920 Summer  1920  Summer  Antwerpen       Football
3  DEN  1900 Summer  1900  Summer      Paris     Tug-Of-War
4  NED  1988 Winter  1988  Winter    Calgary  Speed Skating

                            Event Medal
0       Basketball Men's Basketball   NaN
1      Judo Men's Extra-Lightweight   NaN
2           Football Men's Football   NaN
3       Tug-Of-War Men's Tug-Of-War  Gold
4  Speed Skating Women's 500 metres   NaN
```

1.1.2 1.1.2. Gestion de l'en-tête

[2]:
```
import pandas as pd
donnees_francais=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv",skiprows=1,␣
 ↪names=["Id","Nom", "Sexe", "Age", "Taille", "Poids", "Equipe", "CNO",␣
 ↪"Jeux", "Annee", "Saison", "Ville", "Sport", "Epreuve", "Medaille"])
donnees_francais.head()
#skiprows=1 permet d'ignorer la première ligne du tableau c'est à dire␣
 ↪l'en-tête (header)
#si votre tableu ne contient pas d'en-tête vous pouvez en rajouter un en␣
 ↪utilsant l'option names
#si votre tableau ne contient pas de header et que vous n'en voulez pas il␣
 ↪suffit de rajouter header=None
#vous n'avez pas besoin de spécifier header=None car vous passez l'option names␣
 ↪et Pandas comprend qu'il ne doit pas prendre le header
```

[2]:

	Id	Nom	Sexe	Age	Taille	Poids	Equipe	\
0	1	A Dijiang	M	24.0	180.0	80.0	China	
1	2	A Lamusi	M	23.0	170.0	60.0	China	
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	

	CNO	Jeux	Annee	Saison	Ville	Sport	\
0	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	
1	CHN	2012 Summer	2012	Summer	London	Judo	
2	DEN	1920 Summer	1920	Summer	Antwerpen	Football	
3	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	
4	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	

	Epreuve	Medaille
0	Basketball Men's Basketball	NaN
1	Judo Men's Extra-Lightweight	NaN
2	Football Men's Football	NaN
3	Tug-Of-War Men's Tug-Of-War	Gold
4	Speed Skating Women's 500 metres	NaN

1.1.3 1.1.3. Gestion des indexs

[3]:
```
import pandas as pd
donnees_1index=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", index_col=[0])
donnees_1index.head()
#read_csv ou read_table prend la première colonne du tableau comme index␣
 ↪lorsque par exemple le nom de la première colonne est vide
```

```
#on peut spécifier quelle colonne du jeu de données correspond à l'index avec␣
 ↪l'option index_col=0 pour la première colonne etc...
#on a passé ici la première colonne ID comme index...avec Pandas il est␣
 ↪possible d'avoir des index qui ne sont pas uniques
#si un athlète est présent plusieurs fois son ID sera présent plusieurs fois␣
 ↪dans l'index, cela permet de faire des traitements particuliers sur les index
#on peut aussi écrire index_col="ID" ça évite de se prendre les pieds dans le␣
 ↪tapis...
```

[3]:

ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

1.1.4 1.1.4. Filtrer des colonnes lors de la lecture du fichier

[4]:
```
import pandas as pd
donnees_usecols_position=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", usecols=[1,4,5])
donnees_usecols_position.head()
#on peut définir plusieurs colonnes comme étant les index avec index_col=[.,.,.]
```

[4]:

	Name	Height	Weight
0	A Dijiang	180.0	80.0
1	A Lamusi	170.0	60.0
2	Gunnar Nielsen Aaby	NaN	NaN
3	Edgar Lindenau Aabye	NaN	NaN
4	Christine Jacoba Aaftink	185.0	82.0

[5]:
```
import pandas as pd
donnees_usecols =pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv",␣
 ↪usecols=["Name","Height","Weight"])
donnees_usecols.head()
#on peut aussi sélectionner des colonnes par leurs noms, ce qui est plutôt␣
 ↪pratique
```

[5]:

	Name	Height	Weight
0	A Dijiang	180.0	80.0
1	A Lamusi	170.0	60.0
2	Gunnar Nielsen Aaby	NaN	NaN
3	Edgar Lindenau Aabye	NaN	NaN
4	Christine Jacoba Aaftink	185.0	82.0

1.1.5 1.1.5. Gestion des dates lors de la lecture du fichier

[6]:
```
donnees.dtypes
```

[6]:
```
ID          int64
Name       object
Sex        object
Age       float64
Height    float64
Weight    float64
Team       object
NOC        object
Games      object
Year        int64
Season     object
City       object
Sport      object
Event      object
Medal      object
dtype: object
```

[7]:
```
import pandas as pd
donnees_dates=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", parse_dates=[9])
donnees_dates.dtypes
#parse_dates=[9] ou plus simplement parse_dates=["Year"] permet de dire à␣
 ↪Pandas que la colonne Year doit être considérée comme une date
#Pandas va analyser le format de la date et la formater correctement car au␣
 ↪départ (au-dessus) Year est de type int64...
#en-dessous on peut remarquer qu'après avoir parser on a un type datetime64␣
 ↪pour Year
```

[7]:
```
ID                 int64
Name              object
Sex               object
Age              float64
Height           float64
Weight           float64
Team              object
NOC               object
Games             object
Year      datetime64[ns]
Season            object
City              object
Sport             object
Event             object
Medal             object
dtype: object
```

1.2 1.2. Lecture de fichiers Excel

[8]:
```
import pandas as pd
donnees_excel=pd.read_excel('../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.xlsx')
donnees_excel.head()
#il faut penser à installer le paquet openpyxl
```

[8]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

[9]:
```
import pandas as pd
donnees_excel=pd.read_excel('../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.xlsx',␣
 ↪sheet_name='Feuille2')
donnees_excel.head()
#par défaut read_excel lit la première feuille du classeur Excel
#si on souhaite importer une autre feuille il faut rajouter l'option␣
 ↪sheet_name="Nom de la feuille que je veux"
#si vous rencontrez une erreur lors de donnees_excel.head() il faudra installer␣
 ↪le paquet xlrd
```

[9]:

	Exemple	Exemple.1	Exemple.2
0	Exemple	Exemple	Exemple
1	Exemple	Exemple	Exemple
2	Exemple	Exemple	Exemple
3	Exemple	Exemple	Exemple
4	Exemple	Exemple	Exemple

1.3 1.3. Importer des données à partir d'une base de données

[10]:
```
import pandas as pd
import sqlite3
connexion = sqlite3.connect('../datasets/chinook.db')
requete = "SELECT * FROM employees;"
resultats = pd.read_sql(requete, con=connexion)
resultats.head()
#pensez à importer la BDD chinook.db dans datasets
```

[10]:

	EmployeeId	LastName	FirstName	Title	ReportsTo	BirthDate	HireDate	Address	City	State	Country	PostalCode	Phone	Fax
0	1	Adams	Andrew	General Manager	NaN	1962-02-18 00:00:00	2002-08-14 00:00:00	11120 Jasper Ave NW	Edmonton	AB	Canada	T5K 2N1	+1 (780) 428-9482	+1 (780) 428-3457
1	2	Edwards	Nancy	Sales Manager	1.0	1958-12-08 00:00:00	2002-05-01 00:00:00	825 8 Ave SW	Calgary	AB	Canada	T2P 2T3	+1 (403) 262-3443	+1 (403) 262-3322
2	3	Peacock	Jane	Sales Support Agent	2.0	1973-08-29 00:00:00	2002-04-01 00:00:00	1111 6 Ave SW	Calgary	AB	Canada	T2P 5M5	+1 (403) 262-3443	+1 (403) 262-6712
3	4	Park	Margaret	Sales Support Agent	2.0	1947-09-19 00:00:00	2003-05-03 00:00:00	683 10 Street SW	Calgary	AB	Canada	T2P 5G3	+1 (403) 263-4423	+1 (403) 263-4289
4	5	Johnson	Steve	Sales Support Agent	2.0	1965-03-03 00:00:00	2003-10-17 00:00:00	7727B 41 Ave	Calgary					

```
4    AB  Canada    T3B 1Y7   1 (780) 836-9987   1 (780) 836-9543

                      Email
0    andrew@chinookcorp.com
1     nancy@chinookcorp.com
2      jane@chinookcorp.com
3  margaret@chinookcorp.com
4     steve@chinookcorp.com
```

1.4 1.4. Lecture de fichiers au format JSON

[11]:
```
import pandas as pd
pd.read_json("../datasets/exemple.json")
#le format json est un format text mais ses champs sont organisés d'une manière␣
 ↪différente
```

[11]:
```
                                                    employee
address    [{'street': 'a', 'city': 'b', 'state': 'c'}, {…
firstName                                               nara
lastName                                               simha
```

2 2. Structure de données Pandas : les Series (Séries)

2.1 2.1. Créer des séries

2.1.1 2.1.1. A partir de valeurs aléatoires

[12]:
```
import numpy as np
import pandas as pd
mes_nombres = pd.Series(np.random.choice(list(range(0,500)),10))
print(mes_nombres)
#pd.Series(np.random.choice(data,size))
#data=list(range(0,500)) on crée une liste contenant 500 valeurs de 0 à 499
#size=10 on sélectionne 10 valeurs parmi ces 500 valeurs...on fait ceci␣
 ↪aléatoirement en utilisant random.choice()
#on voit apparaître deux colonnes, la première étant l'index comme pour les␣
 ↪ndarrays de NumPy
#la structure de données Series possèdent deux composants principaux:
#1)les valeurs auxquelles on peut accéder avec la méthode values(),
#2)les étiquettes plus couramment appelées les index auxquelles on peut accéder␣
 ↪avec la méthode index()
```

```
0    283
1    427
2     43
3    276
4    107
```

```
5    444
6    272
7    169
8     65
9    275
dtype: int64
```

2.1.2 2.1.2. A partir d'une liste Python

[13]:
```
import pandas as pd
ma_serie = pd.Series([25,45,62,12,11])
print(ma_serie)
```

```
0    25
1    45
2    62
3    12
4    11
dtype: int64
```

[14]:
```
ma_liste = ["hello", 1,2,3]
print(ma_liste)
```

```
['hello', 1, 2, 3]
```

[15]:
```
print(pd.Series(ma_liste))
#le dtype ici est object qui correspond soit au type str soit au type mixte␣
 ↪(mélange de chaînes de carctères, nombres, etc...)
#ceci est normal car les données sont hétérogènes
#même si les séries acceptent des données hétérogènes, ceci est déconseillé car␣
 ↪cela nuit aux performances
```

```
0    hello
1        1
2        2
3        3
dtype: object
```

[16]:
```
series_mixed=pd.Series(ma_liste)
print(type(series_mixed[0])) #type de la valeur à l'index 0 (première ligne...
 ↪hello)
print(type(series_mixed[1])) #type de la valeur à l'index 1 (deuxième ligne...1)
```

```
<class 'str'>
<class 'int'>
```

2.1.3 2.1.3. A partir d'un fichier texte

[17]:
```
import pandas as pd
ma_serie_de_poids_index_defaut=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", usecols=[5])#,␣
 ↪squeeze=True)
ma_serie_de_poids_index_defaut.squeeze("columns") #ou .squeeze(axis=0) ou .
 ↪squeeze()
#print(ma_serie_de_poids_index_defaut)
#squeeze=True permet de dire à Pandas qu'on souhaite créer un objet de classe␣
 ↪Series et non pas un dataframe
#cette méthode a changé et elle est transformée par l'instruction␣
 ↪ma_serie_de_poids_index_defaut.squeeze("columns")
#usecols=[5] permet de sélectionner uniquement la colonne Weight...on peut␣
 ↪aussi écrire usecols=["Weight"]
#type(ma_serie_de_poids_index_defaut)
```

[17]:
```
0         80.0
1         60.0
2          NaN
3          NaN
4         82.0
          …
271111    89.0
271112    59.0
271113    59.0
271114    96.0
271115    96.0
Name: Weight, Length: 271116, dtype: float64
```

2.2 2.2. Choisir l'index d'une série

[18]:
```
import pandas as pd
ma_serie_de_poids=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", usecols=[1,5],␣
 ↪index_col=[0])
ma_serie_de_poids.squeeze()
#ma_serie_de_poids
#on demande à Pandas de sélectionner uniquement les colonnes 1 et 5 (Name et␣
 ↪Weight) avec l'option usecols
#l'option index_col=[0] permet de définir la première colonne Name comme la␣
 ↪colonnes des index (index_col[0] est une position relative après la␣
 ↪sélection des deux colonnes)
#les index correspondent maintenant au nom des athlètes ce qui est plus␣
 ↪pratique qu'avec des chiffres incrémentés
#c'est dans ce cas précis qu'on voit la puissance des séries avec la␣
 ↪possibilité d'ajouter l'étiquette souhaitée à nos valeurs
```

```
#NaN signifie que des valeurs sont manquantes
#on peut aussi écrire ma_serie_de_poids=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", usecols=[1,5],␣
 ↪index_col=[0]).squeeze()
```

[18]:
```
Name
A Dijiang                   80.0
A Lamusi                    60.0
Gunnar Nielsen Aaby          NaN
Edgar Lindenau Aabye         NaN
Christine Jacoba Aaftink    82.0
                            ...
Andrzej ya                  89.0
Piotr ya                    59.0
Piotr ya                    59.0
Tomasz Ireneusz ya          96.0
Tomasz Ireneusz ya          96.0
Name: Weight, Length: 271116, dtype: float64
```

[19]:
```
import pandas as pd
ma_serie_de_poids=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv",␣
 ↪usecols=["Name","Weight"], index_col=["Name"])
ma_serie_de_poids.squeeze()
#même résultat que la requête ci-dessus mais peut-être plus simple pour se␣
 ↪repérer
```

[19]:
```
Name
A Dijiang                   80.0
A Lamusi                    60.0
Gunnar Nielsen Aaby          NaN
Edgar Lindenau Aabye         NaN
Christine Jacoba Aaftink    82.0
                            ...
Andrzej ya                  89.0
Piotr ya                    59.0
Piotr ya                    59.0
Tomasz Ireneusz ya          96.0
Tomasz Ireneusz ya          96.0
Name: Weight, Length: 271116, dtype: float64
```

2.3 2.3. Accéder aux valeurs d'une série

2.3.1 2.3.1. Indexing via la position des valeurs

[20]:
```
ma_serie_de_poids=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv",␣
 ↪usecols=["Name","Weight"], index_col=["Name"]).squeeze()
ma_serie_de_poids[[0,15,6985,452]]
#nous venons de récupérer les valeurs aux positions 0, 15, 6985 et 452
#le premier couple de crochets correspond à la syntaxe pour sélectionner des␣
 ↪données
#le deuxième couple de crochets correspond à la liste des positions auxquelles␣
 ↪on souhaite récupérer des valeurs
```

[20]:
```
Name
A Dijiang                                 80.0
Per Knut Aaland                           75.0
Gunilla Elisabeth Andersson (-Leinskld)   59.0
Bashir Abdi                               56.0
Name: Weight, dtype: float64
```

2.3.2 2.3.2. Indexing via l'étiquette des valeurs

[21]:
```
ma_serie_de_poids[["Antti Sami Aalto","Andrzej ya"]]
#ma_serie_de_poids.loc[["Antti Sami Aalto","Andrzej ya"]]
```

[21]:
```
Name
Antti Sami Aalto    96.0
Andrzej ya          89.0
Name: Weight, dtype: float64
```

2.3.3 2.3.3. Les indexeurs loc et iloc

[22]:
```
import pandas as pd
ma_serie=pd.Series([10,11,12,13], index=[1,3,2,0])
ma_serie
#on crée une série en lui affectant des index
```

[22]:
```
1    10
3    11
2    12
0    13
dtype: int64
```

[23]:
```
ma_serie[0]
#ne renvoie pas l'index de la première ligne...
```

[23]:
```
13
```

[24]:
```
ma_serie.loc[0]
#l'attribut loc va demander à Pandas d'utiliser les noms d'index
```

[24]:
```
13
```

[25]:
```
ma_serie.iloc[0]
#l'attribut iloc va demander à Pandas d'utiliser la position des valeurs pour␣
 ↪les sélectionner dans la série
```

[25]:
```
10
```

2.3.4 2.3.4. Indexing via une expression booléenne

[26]:
```
ma_serie_de_poids>90
#similaire à la bibliothèque NumPy...
```

[26]:
```
Name
A Dijiang                  False
A Lamusi                   False
Gunnar Nielsen Aaby        False
Edgar Lindenau Aabye       False
Christine Jacoba Aaftink   False
                           …
Andrzej ya                 False
Piotr ya                   False
Piotr ya                   False
Tomasz Ireneusz ya          True
Tomasz Ireneusz ya          True
Name: Weight, Length: 271116, dtype: bool
```

[27]:
```
ma_serie_de_poids[ma_serie_de_poids>90]
#on garde uniquement les athlètes à True
```

[27]:
```
Name
Antti Sami Aalto           96.0
Timo Antero Aaltonen      130.0
Andreea Aanei             125.0
Dagfinn Sverre Aarskog     98.0
Hans Aasns                 93.0
                          …
Henk Jan Zwolle            93.0
Dominik ycki               95.0
Dominik ycki               95.0
Tomasz Ireneusz ya         96.0
Tomasz Ireneusz ya         96.0
```

```
Name: Weight, Length: 16819, dtype: float64
```

[28]:
```
ma_serie_de_poids[(ma_serie_de_poids>90) & (ma_serie_de_poids<100)]
#on peut utiliser plusieurs expressions booléennes
#pour et : &, pour ou : |, pour la négation : ~
```

[28]:
```
Name
Antti Sami Aalto          96.0
Dagfinn Sverre Aarskog    98.0
Hans Aasns                93.0
Hans Aasns                93.0
Hans Aasns                93.0
                          …
Henk Jan Zwolle           93.0
Dominik ycki              95.0
Dominik ycki              95.0
Tomasz Ireneusz ya        96.0
Tomasz Ireneusz ya        96.0
Name: Weight, Length: 9892, dtype: float64
```

2.3.5 2.3.5. Slicing : découpage de valeurs successives

[29]:
```
ma_serie_de_poids_index_defaut.iloc[3:5] #on peut rajouter .squeeze()
#on récupère les valeurs aux positions 3 et 4
#avec Pandas on utilisera iloc pour ne pas avoir faire de confusion entre␣
 ↪l'index et la position
```

[29]:
```
   Weight
3     NaN
4    82.0
```

[30]:
```
ma_serie_de_poids_index_defaut.iloc[3:5].squeeze()
#on récupère les valeurs aux positions 3 et 4
#avec Pandas on utilisera iloc pour ne pas avoir faire de confusion entre␣
 ↪l'index et la position
```

[30]:
```
3     NaN
4    82.0
Name: Weight, dtype: float64
```

[31]:
```
ma_serie_de_poids_index_defaut.iloc[2:11:2]
#on récupère les valeurs aux positions 2, 4, 6, 8 et 10
#serie[start:stop:step]
```

[31]:
```
    Weight
2      NaN
4     82.0
```

```
6     82.0
8     82.0
10    75.0
```

[32]:
```
ma_serie_de_poids_index_defaut.iloc[2:11]
#on récupère les valeurs aux positions 2, 3, ...,9 ,10
```

[32]:
```
     Weight
2       NaN
3       NaN
4     82.0
5     82.0
6     82.0
7     82.0
8     82.0
9     82.0
10    75.0
```

[33]:
```
ma_serie_de_poids_index_defaut.iloc[1000:]
#on récupère les valeurs aux positions 1000 jusqu'à la fin de la série
```

[33]:
```
         Weight
1000       84.0
1001       84.0
1002       84.0
1003       73.0
1004       54.0
…           …
271111     89.0
271112     59.0
271113     59.0
271114     96.0
271115     96.0

[270116 rows x 1 columns]
```

[34]:
```
ma_serie_de_poids_index_defaut.iloc[:20]
#on récupère les valeurs aux positions 0,1,2,...,19
```

[34]:
```
     Weight
0      80.0
1      60.0
2       NaN
3       NaN
4     82.0
5     82.0
6     82.0
```

```
7     82.0
8     82.0
9     82.0
10    75.0
11    75.0
12    75.0
13    75.0
14    75.0
15    75.0
16    75.0
17    75.0
18    72.0
19    72.0
```

[35]:
```
ma_serie_de_poids_index_defaut.iloc[-1:]
#on récupère la valeur de la dernière position de la série
```

[35]:
```
         Weight
271115    96.0
```

2.4 2.4. Les attributs et les méthodes des objets de classe Series

2.4.1 2.4.1. Les attributs des objets de classe Series

[36]:
```
ma_serie_de_poids.size
#retourne le nombre de valeurs de la série c'est-à-dire sa taille
```

[36]:
```
271116
```

2.4.2 2.4.2. Les méthodes des objets de classe Series

[37]:
```
ma_serie_de_poids.describe()
#la méthode describe() permet d'afficher un résumé statistique sur les valeurs␣
 ↪de la série
```

[37]:
```
count    208241.000000
mean         70.702393
std          14.348020
min          25.000000
25%          60.000000
50%          70.000000
75%          79.000000
max         214.000000
Name: Weight, dtype: float64
```

2.5 2.5. Ajouter, supprimer et modifier les valeurs d'une série

2.5.1 2.5.1. Ajouter des valeurs à une série

[38]:
```
ma_serie_de_poids=ma_serie_de_poids.append(pd.Series([90,120], index=["Claire␣
↪Muller", "Julien Villeroy"]))
ma_serie_de_poids.loc[["Claire Muller","Julien Villeroy"]]
```

```
<ipython-input-38-97a835fc0720>:1: FutureWarning: The series.append method is
deprecated and will be removed from pandas in a future version. Use
pandas.concat instead.
  ma_serie_de_poids=ma_serie_de_poids.append(pd.Series([90,120], index=["Claire
Muller", "Julien Villeroy"]))
```

[38]:
```
Claire Muller       90.0
Julien Villeroy    120.0
dtype: float64
```

[39]:
```
ma_serie_de_poids
#on remarquera que le nombre de valeurs est passé de 271116 à 271118
```

[39]:
```
A Dijiang                  80.0
A Lamusi                   60.0
Gunnar Nielsen Aaby         NaN
Edgar Lindenau Aabye        NaN
Christine Jacoba Aaftink   82.0
                           ...
Piotr ya                   59.0
Tomasz Ireneusz ya         96.0
Tomasz Ireneusz ya         96.0
Claire Muller              90.0
Julien Villeroy           120.0
Length: 271118, dtype: float64
```

2.5.2 2.5.2. Supprimer une valeur d'une série

[40]:
```
ma_serie_de_poids.drop(labels=["Claire Muller","Julien Villeroy"], inplace=True)
#ma_serie_de_poids.loc[["Claire Muller","Julien Villeroy"]]
ma_serie_de_poids
#on remarquera que le nombre de valeurs est à nouveau de 271116
#inplace=True signifie qu'on modifie directement la série de départ sur␣
↪laquelle on travaille
#par défaut inplace=False dans ce cas on écrira␣
↪mon_nouveau_dataframe=ma_serie_de_poids.drop(labels=["Claire Muller","Julien␣
↪Villeroy"], inplace=False)
#ou la même chose sans inplace=False
```

[40]:
```
A Dijiang                   80.0
A Lamusi                    60.0
Gunnar Nielsen Aaby          NaN
Edgar Lindenau Aabye         NaN
Christine Jacoba Aaftink    82.0
                            …
Andrzej ya                  89.0
Piotr ya                    59.0
Piotr ya                    59.0
Tomasz Ireneusz ya          96.0
Tomasz Ireneusz ya          96.0
Length: 271116, dtype: float64
```

2.5.3 2.5.3. Modifier les valeurs d'une série

[41]:
```
ma_serie_de_poids.loc[["Antti Sami Aalto"]]=56
ma_serie_de_poids.loc[["Antti Sami Aalto"]]
```

[41]:
```
Antti Sami Aalto    56.0
dtype: float64
```

[42]:
```
ma_serie_de_poids.loc[["Antti Sami Aalto","Andrzej ya"]]=[56,70]
ma_serie_de_poids.loc[["Antti Sami Aalto","Andrzej ya"]]
```

[42]:
```
Antti Sami Aalto    56.0
Andrzej ya          70.0
dtype: float64
```

[43]:
```
ma_serie_de_poids.loc[["Christine Jacoba Aaftink"]]=60
ma_serie_de_poids.loc[["Christine Jacoba Aaftink"]]
```

[43]:
```
Christine Jacoba Aaftink    60.0
Christine Jacoba Aaftink    60.0
Christine Jacoba Aaftink    60.0
Christine Jacoba Aaftink    60.0
Christine Jacoba Aaftink    60.0
Christine Jacoba Aaftink    60.0
dtype: float64
```

3 3. Structure de données Pandas : les objets de type DataFrame

3.1 3.1. Introduction

[44]:
```
import pandas as pd
donnees=pd.read_csv("../datasets/
  ↪120-years-of-olympic-history-athletes-and-results.csv", index_col=[1])
donnees.head()
```

```
#ici on définit les noms d'athlètes (deuxième colonne du dataframe de départ)
```

[44]:
```
                          ID Sex   Age  Height  Weight            Team  NOC  \
Name
A Dijiang                  1   M  24.0   180.0    80.0           China  CHN
A Lamusi                   2   M  23.0   170.0    60.0           China  CHN
Gunnar Nielsen Aaby        3   M  24.0     NaN     NaN         Denmark  DEN
Edgar Lindenau Aabye       4   M  34.0     NaN     NaN  Denmark/Sweden  DEN
Christine Jacoba Aaftink   5   F  21.0   185.0    82.0     Netherlands  NED

                                Games  Year  Season       City          Sport  \
Name
A Dijiang                1992 Summer  1992  Summer  Barcelona     Basketball
A Lamusi                 2012 Summer  2012  Summer     London           Judo
Gunnar Nielsen Aaby      1920 Summer  1920  Summer  Antwerpen       Football
Edgar Lindenau Aabye     1900 Summer  1900  Summer      Paris     Tug-Of-War
Christine Jacoba Aaftink 1988 Winter  1988  Winter    Calgary  Speed Skating

                                                 Event Medal
Name
A Dijiang                      Basketball Men's Basketball   NaN
A Lamusi                   Judo Men's Extra-Lightweight   NaN
Gunnar Nielsen Aaby                Football Men's Football   NaN
Edgar Lindenau Aabye           Tug-Of-War Men's Tug-Of-War  Gold
Christine Jacoba Aaftink  Speed Skating Women's 500 metres   NaN
```

3.2 3.2. Indexing : sélectionner des valeurs d'un dataframe

3.2.1 3.2.1. Indexing et slicing avec l'attribut loc

[45]:
```
donnees.loc["Usain St. Leo Bolt"]
#on regarde toutes informations relatives à Usain Bolt
```

[45]:
```
                       ID Sex   Age  Height  Weight     Team  NOC  \
Name
Usain St. Leo Bolt  13029   M  17.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  21.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  21.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  21.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  25.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  25.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  25.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  29.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  29.0   196.0    95.0  Jamaica  JAM
Usain St. Leo Bolt  13029   M  29.0   196.0    95.0  Jamaica  JAM

                          Games  Year  Season       City      Sport  \
```

```
Name
Usain St. Leo Bolt  2004 Summer  2004  Summer          Athina  Athletics
Usain St. Leo Bolt  2008 Summer  2008  Summer         Beijing  Athletics
Usain St. Leo Bolt  2008 Summer  2008  Summer         Beijing  Athletics
Usain St. Leo Bolt  2008 Summer  2008  Summer         Beijing  Athletics
Usain St. Leo Bolt  2012 Summer  2012  Summer          London  Athletics
Usain St. Leo Bolt  2012 Summer  2012  Summer          London  Athletics
Usain St. Leo Bolt  2012 Summer  2012  Summer          London  Athletics
Usain St. Leo Bolt  2016 Summer  2016  Summer  Rio de Janeiro  Athletics
Usain St. Leo Bolt  2016 Summer  2016  Summer  Rio de Janeiro  Athletics
Usain St. Leo Bolt  2016 Summer  2016  Summer  Rio de Janeiro  Athletics

                                                      Event Medal
Name
Usain St. Leo Bolt            Athletics Men's 200 metres   NaN
Usain St. Leo Bolt            Athletics Men's 100 metres  Gold
Usain St. Leo Bolt            Athletics Men's 200 metres  Gold
Usain St. Leo Bolt  Athletics Men's 4 x 100 metres Relay   NaN
Usain St. Leo Bolt            Athletics Men's 100 metres  Gold
Usain St. Leo Bolt            Athletics Men's 200 metres  Gold
Usain St. Leo Bolt  Athletics Men's 4 x 100 metres Relay  Gold
Usain St. Leo Bolt            Athletics Men's 100 metres  Gold
Usain St. Leo Bolt            Athletics Men's 200 metres  Gold
Usain St. Leo Bolt  Athletics Men's 4 x 100 metres Relay  Gold
```

[46]:
```
donnees.loc["Usain St. Leo Bolt",["Games", "Event", "Medal"]]
#on se limite ici à l'affichage des colonnes "Games", "Event", "Medal"␣
 ↪concernant Usain Bolt
```

[46]:
```
                          Games                                 Event Medal
Name
Usain St. Leo Bolt  2004 Summer            Athletics Men's 200 metres   NaN
Usain St. Leo Bolt  2008 Summer            Athletics Men's 100 metres  Gold
Usain St. Leo Bolt  2008 Summer            Athletics Men's 200 metres  Gold
Usain St. Leo Bolt  2008 Summer  Athletics Men's 4 x 100 metres Relay   NaN
Usain St. Leo Bolt  2012 Summer            Athletics Men's 100 metres  Gold
Usain St. Leo Bolt  2012 Summer            Athletics Men's 200 metres  Gold
Usain St. Leo Bolt  2012 Summer  Athletics Men's 4 x 100 metres Relay  Gold
Usain St. Leo Bolt  2016 Summer            Athletics Men's 100 metres  Gold
Usain St. Leo Bolt  2016 Summer            Athletics Men's 200 metres  Gold
Usain St. Leo Bolt  2016 Summer  Athletics Men's 4 x 100 metres Relay  Gold
```

[47]:
```
donnees.loc[["Justin Alexander Gatlin","Usain St. Leo Bolt"],["Games", "Event",␣
 ↪"Medal"]]
#on peut ici comparer les performances d'Usain Bolt avec Justin Alexander␣
 ↪Gatlin sur les trois colonnes "Games", "Event", "Medal"
```

[47]:

Name	Games	Event	Medal
Justin Alexander Gatlin	2004 Summer	Athletics Men's 100 metres	Gold
Justin Alexander Gatlin	2004 Summer	Athletics Men's 200 metres	Bronze
Justin Alexander Gatlin	2004 Summer	Athletics Men's 4 x 100 metres Relay	Silver
Justin Alexander Gatlin	2012 Summer	Athletics Men's 100 metres	Bronze
Justin Alexander Gatlin	2012 Summer	Athletics Men's 4 x 100 metres Relay	NaN
Justin Alexander Gatlin	2016 Summer	Athletics Men's 100 metres	Silver
Justin Alexander Gatlin	2016 Summer	Athletics Men's 200 metres	NaN
Justin Alexander Gatlin	2016 Summer	Athletics Men's 4 x 100 metres Relay	NaN
Usain St. Leo Bolt	2004 Summer	Athletics Men's 200 metres	NaN
Usain St. Leo Bolt	2008 Summer	Athletics Men's 100 metres	Gold
Usain St. Leo Bolt	2008 Summer	Athletics Men's 200 metres	Gold
Usain St. Leo Bolt	2008 Summer	Athletics Men's 4 x 100 metres Relay	NaN
Usain St. Leo Bolt	2012 Summer	Athletics Men's 100 metres	Gold
Usain St. Leo Bolt	2012 Summer	Athletics Men's 200 metres	Gold
Usain St. Leo Bolt	2012 Summer	Athletics Men's 4 x 100 metres Relay	Gold
Usain St. Leo Bolt	2016 Summer	Athletics Men's 100 metres	Gold
Usain St. Leo Bolt	2016 Summer	Athletics Men's 200 metres	Gold
Usain St. Leo Bolt	2016 Summer	Athletics Men's 4 x 100 metres Relay	Gold

[48]:
```
donnees.loc['A Dijiang':'Edgar Lindenau Aabye':1,'Age':'NOC':2]
#ici on demande de sélectionner les athlètes A Dijiang et Edgar Lindenau Aabye␣
 ↪et tous les athlètes entre ces deux-là (pas de 1)
#ensuite on conserve les colonnes entre Age et NOC avec un pas de 2 (il n'y a␣
 ↪que Weight)
```

[48]:
```
                        Age  Weight  NOC
Name
A Dijiang              24.0    80.0  CHN
A Lamusi               23.0    60.0  CHN
Gunnar Nielsen Aaby    24.0     NaN  DEN
Edgar Lindenau Aabye   34.0     NaN  DEN
```

3.2.2 3.2.2. Indexing et slicing avec l'attribut iloc

[49]:
```
donnees.iloc[0:1,0:10]
#ici on sélectionne la première ligne et les dix premières colonnes du␣
 ↪dataframe "donnees"
```

[49]:
```
             ID Sex   Age  Height  Weight   Team  NOC        Games  Year  Season
Name
A Dijiang     1   M  24.0   180.0    80.0  China  CHN  1992 Summer  1992  Summer
```

3.2.3 3.2.3. Indexing avec une expression booléenne

[50]:
```
donnees[(donnees["Sex"]=="F") & (donnees["Medal"]=="Gold")]
#on affiche l'ensemble des femmes qui ont participé aux JO et qui ont obtenu␣
 ↪une médaille d'or
#on remarquera qu'il y a 3747 lignes et 14 colonnes
#on pouurait comparer avec le nombre de médailles d'or obtenus par des hommes␣
 ↪en faisant la négation ~
```

[50]:
```
                                          ID Sex   Age  Height  Weight  \
Name
Ragnhild Margrethe Aamodt                 21   F  27.0   163.0     NaN
Margaret Ives Abbott (-Dunne)            150   F  23.0     NaN     NaN
Nicola Virginia Adams                    832   F  29.0   164.0    51.0
Nicola Virginia Adams                    832   F  33.0   164.0    51.0
Valerie Kasanita Adams-Vili (-Price)     846   F  23.0   193.0   120.0
…                                        …  ..  …      …       …
Olha Valentynivna Zubareva            135331   F  22.0   182.0    90.0
Nataliya Vladimirovna Zuyeva          135488   F  19.0   176.0    62.0
Ellina Aleksandrovna Zvereva (Kisheyeva-) 135501 F  39.0   183.0   100.0
Julia Zwehl                           135520   F  28.0   167.0    60.0
Galina Ivanovna Zybina (-Fyodorova)   135553   F  21.0   168.0    80.0

                                                Team  NOC        Games  \
Name
Ragnhild Margrethe Aamodt                     Norway  NOR  2008 Summer
Margaret Ives Abbott (-Dunne)          United States  USA  1900 Summer
Nicola Virginia Adams                  Great Britain  GBR  2012 Summer
Nicola Virginia Adams                  Great Britain  GBR  2016 Summer
```

```
Valerie Kasanita Adams-Vili (-Price)         New Zealand  NZL  2008 Summer
…                                                      …  …          …
Olha Valentynivna Zubareva                  Soviet Union  URS  1980 Summer
Nataliya Vladimirovna Zuyeva                      Russia  RUS  2008 Summer
Ellina Aleksandrovna Zvereva (Kisheyeva-)        Belarus  BLR  2000 Summer
Julia Zwehl                                      Germany  GER  2004 Summer
Galina Ivanovna Zybina (-Fyodorova)         Soviet Union  URS  1952 Summer

                                          Year  Season            City  \
Name
Ragnhild Margrethe Aamodt                 2008  Summer         Beijing
Margaret Ives Abbott (-Dunne)             1900  Summer           Paris
Nicola Virginia Adams                     2012  Summer          London
Nicola Virginia Adams                     2016  Summer  Rio de Janeiro
Valerie Kasanita Adams-Vili (-Price)      2008  Summer         Beijing
…                                            …       …             …
Olha Valentynivna Zubareva                1980  Summer          Moskva
Nataliya Vladimirovna Zuyeva              2008  Summer         Beijing
Ellina Aleksandrovna Zvereva (Kisheyeva-) 2000  Summer          Sydney
Julia Zwehl                               2004  Summer          Athina
Galina Ivanovna Zybina (-Fyodorova)       1952  Summer        Helsinki

                                                      Sport  \
Name
Ragnhild Margrethe Aamodt                          Handball
Margaret Ives Abbott (-Dunne)                          Golf
Nicola Virginia Adams                                Boxing
Nicola Virginia Adams                                Boxing
Valerie Kasanita Adams-Vili (-Price)               Athletics
…                                                        …
Olha Valentynivna Zubareva                         Handball
Nataliya Vladimirovna Zuyeva               Rhythmic Gymnastics
Ellina Aleksandrovna Zvereva (Kisheyeva-)          Athletics
Julia Zwehl                                          Hockey
Galina Ivanovna Zybina (-Fyodorova)                Athletics

                                                                  Event  \
Name
Ragnhild Margrethe Aamodt                          Handball Women's Handball
Margaret Ives Abbott (-Dunne)                        Golf Women's Individual
Nicola Virginia Adams                              Boxing Women's Flyweight
Nicola Virginia Adams                              Boxing Women's Flyweight
Valerie Kasanita Adams-Vili (-Price)             Athletics Women's Shot Put
…                                                                        …
Olha Valentynivna Zubareva                         Handball Women's Handball
Nataliya Vladimirovna Zuyeva               Rhythmic Gymnastics Women's Group
Ellina Aleksandrovna Zvereva (Kisheyeva-)     Athletics Women's Discus Throw
```

```
Julia Zwehl                                          Hockey Women's Hockey
Galina Ivanovna Zybina (-Fyodorova)                Athletics Women's Shot Put

                                                Medal
Name
Ragnhild Margrethe Aamodt                        Gold
Margaret Ives Abbott (-Dunne)                    Gold
Nicola Virginia Adams                            Gold
Nicola Virginia Adams                            Gold
Valerie Kasanita Adams-Vili (-Price)             Gold
…                                                …
Olha Valentynivna Zubareva                       Gold
Nataliya Vladimirovna Zuyeva                     Gold
Ellina Aleksandrovna Zvereva (Kisheyeva-)        Gold
Julia Zwehl                                      Gold
Galina Ivanovna Zybina (-Fyodorova)              Gold

[3747 rows x 14 columns]
```

[51]:
```
donnees[~(donnees["Sex"]=="F") & (donnees["Medal"]=="Gold")]
#négation de la requête précédente et équivalente à␣
 ↪donnees[(donnees["Sex"]=="M") & (donnees["Medal"]=="Gold")]
```

[51]:
```
                                    ID Sex   Age  Height  Weight  \
Name
Edgar Lindenau Aabye                 4   M  34.0     NaN     NaN
Paavo Johannes Aaltonen             17   M  28.0   175.0    64.0
Paavo Johannes Aaltonen             17   M  28.0   175.0    64.0
Paavo Johannes Aaltonen             17   M  28.0   175.0    64.0
Kjetil Andr Aamodt                  20   M  20.0   176.0    85.0
…                                  …  ..  …      …      …
Albert Hermann Zrner            135474   M  18.0     NaN     NaN
Jules Alexis "Louis" Zutter     135481   M  30.0     NaN     NaN
Zurab Zviadauri                 135503   M  23.0   182.0    90.0
Ronald Ferdinand "Ron" Zwerver  135523   M  29.0   200.0    93.0
Henk Jan Zwolle                 135545   M  31.0   197.0    93.0

                                          Team  NOC        Games  Year  \
Name
Edgar Lindenau Aabye            Denmark/Sweden  DEN  1900 Summer  1900
Paavo Johannes Aaltonen                Finland  FIN  1948 Summer  1948
Paavo Johannes Aaltonen                Finland  FIN  1948 Summer  1948
Paavo Johannes Aaltonen                Finland  FIN  1948 Summer  1948
Kjetil Andr Aamodt                      Norway  NOR  1992 Winter  1992
…                                          …  …          …   …
Albert Hermann Zrner                   Germany  GER  1908 Summer  1908
Jules Alexis "Louis" Zutter        Switzerland  SUI  1896 Summer  1896
```

```
Zurab Zviadauri                        Georgia  GEO  2004 Summer  2004
Ronald Ferdinand "Ron" Zwerver     Netherlands  NED  1996 Summer  1996
Henk Jan Zwolle                    Netherlands  NED  1996 Summer  1996

                                Season         City          Sport  \
Name
Edgar Lindenau Aabye            Summer        Paris     Tug-Of-War
Paavo Johannes Aaltonen         Summer       London     Gymnastics
Paavo Johannes Aaltonen         Summer       London     Gymnastics
Paavo Johannes Aaltonen         Summer       London     Gymnastics
Kjetil Andr Aamodt              Winter  Albertville  Alpine Skiing
…                                …           …             …
Albert Hermann Zrner            Summer       London         Diving
Jules Alexis "Louis" Zutter     Summer       Athina     Gymnastics
Zurab Zviadauri                 Summer       Athina           Judo
Ronald Ferdinand "Ron" Zwerver  Summer      Atlanta     Volleyball
Henk Jan Zwolle                 Summer      Atlanta         Rowing

                                                          Event Medal
Name
Edgar Lindenau Aabye                   Tug-Of-War Men's Tug-Of-War  Gold
Paavo Johannes Aaltonen         Gymnastics Men's Team All-Around  Gold
Paavo Johannes Aaltonen             Gymnastics Men's Horse Vault  Gold
Paavo Johannes Aaltonen         Gymnastics Men's Pommelled Horse  Gold
Kjetil Andr Aamodt                   Alpine Skiing Men's Super G  Gold
…                                                          …    …
Albert Hermann Zrner                    Diving Men's Springboard  Gold
Jules Alexis "Louis" Zutter     Gymnastics Men's Pommelled Horse  Gold
Zurab Zviadauri                           Judo Men's Middleweight  Gold
Ronald Ferdinand "Ron" Zwerver       Volleyball Men's Volleyball  Gold
Henk Jan Zwolle                        Rowing Men's Coxed Eights  Gold

[9625 rows x 14 columns]
```

3.3 3.3. Ajout, suppression et modification sur un dataframe

3.3.1 3.3.1. Ajouter une ou plusieurs colonnes à un dataframe

[52]:
```
donnees_test=donnees.copy()
donnees_test["Test"]=0
donnees_test.head()
#cette méthode est la plus simple au niveau syntaxique permet d'ajouter une␣
 ↪colonne à partir d'une liste ou d'une série
#cette colonne s'ajoutant à la fin du dataframe
#dataframe['nouvelle colonne']=[mavaleur1,mavaleur2,...]
#ou dataframe['nouvelle colonne']=ma_serie
```

```
#attention il faut que le nombre de valeurs dans votre liste ou votre série␣
↪soit le même que votre nombre de lignes du dataframe
```

[52]:

Name	ID	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal	Test
A Dijiang	1	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN	0
A Lamusi	2	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN	0
Gunnar Nielsen Aaby	3	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN	0
Edgar Lindenau Aabye	4	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold	0
Christine Jacoba Aaftink	5	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN	0

[53]:
```
donnees_test.insert(2, "Poids_athlètes", ma_serie_de_poids)
donnees_test.head()
#cette deuxième méthode permet d'ajouter la colonne "Poids_athlètes" là où on␣
↪le souhaite en l'occurence ce sera la troisième colonne de notre dataframe
```

[53]:

Name	ID	Sex	Poids_athlètes	Age	Height	Weight	Team	NOC	Games	Year	Season
A Dijiang	1	M	80.0	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer
A Lamusi	2	M	60.0	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer
Gunnar Nielsen Aaby	3	M	NaN	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer
Edgar Lindenau Aabye	4	M	NaN	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer
Christine Jacoba Aaftink	5	F	60.0	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter

```
                                City          Sport  \
Name
A Dijiang                   Barcelona     Basketball
A Lamusi                       London           Judo
Gunnar Nielsen Aaby         Antwerpen       Football
Edgar Lindenau Aabye            Paris     Tug-Of-War
Christine Jacoba Aaftink      Calgary  Speed Skating

                                                    Event Medal  Test
Name
A Dijiang                        Basketball Men's Basketball   NaN     0
A Lamusi                      Judo Men's Extra-Lightweight   NaN     0
Gunnar Nielsen Aaby                Football Men's Football   NaN     0
Edgar Lindenau Aabye            Tug-Of-War Men's Tug-Of-War  Gold     0
Christine Jacoba Aaftink  Speed Skating Women's 500 metres   NaN     0
```

3.3.2 3.3.2. Ajouter une ligne à un dataframe

[54]:
```
ajout_une_ligne = donnees.append(pd.Series(['135572', 'F', 29, 175, 58,␣
 ↪'China', 'CHN', '2012 Summer', 2012, 'Summer', 'London', 'Judo', "Judo␣
 ↪Women's Extra-Lightweight",'NaN' ], index=donnees.columns␣
 ↪),ignore_index=True)
ajout_une_ligne.tail() #permet de visualiser les 5 dernières lignes de notre␣
 ↪dataframe
#index=donnees.columns on fait correspondre les index de valeurs de notre série␣
 ↪créée à la volée aux noms des colonnes du dataframe
#ignore_index=True permet de garder les valeurs d'index incrémentées de notre␣
 ↪dataframe de départ
```

```
<ipython-input-54-3bc1c7413d08>:1: FutureWarning: The frame.append method is
deprecated and will be removed from pandas in a future version. Use
pandas.concat instead.
  ajout_une_ligne = donnees.append(pd.Series(['135572', 'F', 29, 175, 58,
'China', 'CHN', '2012 Summer', 2012, 'Summer', 'London', 'Judo', "Judo Women's
Extra-Lightweight",'NaN' ], index=donnees.columns ),ignore_index=True)
```

[54]:
```
            ID Sex   Age  Height  Weight    Team  NOC        Games  Year  \
271112  135570   M  27.0   176.0    59.0  Poland  POL  2014 Winter  2014
271113  135570   M  27.0   176.0    59.0  Poland  POL  2014 Winter  2014
271114  135571   M  30.0   185.0    96.0  Poland  POL  1998 Winter  1998
271115  135571   M  34.0   185.0    96.0  Poland  POL  2002 Winter  2002
271116  135572   F  29.0   175.0    58.0   China  CHN  2012 Summer  2012

        Season            City        Sport  \
271112  Winter           Sochi  Ski Jumping
271113  Winter           Sochi  Ski Jumping
```

```
271114  Winter          Nagano    Bobsleigh
271115  Winter  Salt Lake City    Bobsleigh
271116  Summer          London         Judo

                                          Event Medal
271112  Ski Jumping Men's Large Hill, Individual   NaN
271113       Ski Jumping Men's Large Hill, Team   NaN
271114                    Bobsleigh Men's Four   NaN
271115                    Bobsleigh Men's Four   NaN
271116           Judo Women's Extra-Lightweight   NaN
```

[55]:
```
ajout_une_ligne = donnees.append(pd.Series(['135572', 'F', 29, 175, 58,␣
 ↪'China', 'CHN', '2012 Summer', 2012, 'Summer', 'London', 'Judo', "Judo␣
 ↪Women's Extra-Lightweight",'NaN' ], index=donnees.columns, name= "Claire␣
 ↪Muller" ))
ajout_une_ligne.tail()
#par défaut "ignore_index=False" donc ici en rajoutant "name='Claire Muller'"␣
 ↪on dit à Pandas de considérer
#la colonne Name qui contient Claire Muller comme étiquette de la nouvelle␣
 ↪ligne ajoutée
#index.columns renvoie la liste des noms de colonnes du dataframe
```

```
<ipython-input-55-58dc48a17d4e>:1: FutureWarning: The frame.append method is
deprecated and will be removed from pandas in a future version. Use
pandas.concat instead.
  ajout_une_ligne = donnees.append(pd.Series(['135572', 'F', 29, 175, 58,
'China', 'CHN', '2012 Summer', 2012, 'Summer', 'London', 'Judo', "Judo Women's
Extra-Lightweight",'NaN' ], index=donnees.columns, name= "Claire Muller" ))
```

[55]:
```
                      ID Sex   Age  Height  Weight    Team  NOC  \
Name
Piotr ya            135570   M  27.0   176.0    59.0  Poland  POL
Piotr ya            135570   M  27.0   176.0    59.0  Poland  POL
Tomasz Ireneusz ya  135571   M  30.0   185.0    96.0  Poland  POL
Tomasz Ireneusz ya  135571   M  34.0   185.0    96.0  Poland  POL
Claire Muller       135572   F  29.0   175.0    58.0   China  CHN

                         Games  Year  Season            City        Sport  \
Name
Piotr ya            2014 Winter  2014  Winter           Sochi  Ski Jumping
Piotr ya            2014 Winter  2014  Winter           Sochi  Ski Jumping
Tomasz Ireneusz ya  1998 Winter  1998  Winter          Nagano    Bobsleigh
Tomasz Ireneusz ya  2002 Winter  2002  Winter  Salt Lake City    Bobsleigh
Claire Muller       2012 Summer  2012  Summer          London         Judo

                                                  Event Medal
Name
```

```
Piotr ya            Ski Jumping Men's Large Hill, Individual   NaN
Piotr ya                  Ski Jumping Men's Large Hill, Team   NaN
Tomasz Ireneusz ya                     Bobsleigh Men's Four   NaN
Tomasz Ireneusz ya                     Bobsleigh Men's Four   NaN
Claire Muller                Judo Women's Extra-Lightweight   NaN
```

3.3.3 3.3.3. Supprimer des lignes ou colonnes d'un dataframe

[56]:
```
donnees_test.drop(["Test","Poids_athlètes"], axis=1, inplace=True)
donnees_test.head()
#la méthode drop permet de supprimer des lignes ou des colonnes
#en rajoutant les noms de colonnes et axis=1 on dit à Pandas d'enlever ces deux␣
 ↪colonnes
#pour supprimer des lignes on peut passer par leur étiquette index
#dataframe_lignes_en_moins=dataframe.drop(["maligne1","maligne2",...],axis=0)
#dataframe_lignes_en_moins=dataframe.drop(dataframe.index[2,3],axis=0) on␣
 ↪enlève les lignes aux positions 2 et 3 du dataframe
#inplace=True signifie qu'on modifie directement le dataframe de départ sur␣
 ↪lequel on travaille
#par défaut inplace=False dans ce cas on écrira␣
 ↪mon_nouveau_dataframe=ma_serie_de_poids.drop(labels=["Claire Muller","Julien␣
 ↪Villeroy"], inplace=False)
#ou la même chose sans inplace=False
```

[56]:
```
                        ID Sex   Age  Height  Weight            Team  NOC  \
Name
A Dijiang                1   M  24.0   180.0    80.0           China  CHN
A Lamusi                 2   M  23.0   170.0    60.0           China  CHN
Gunnar Nielsen Aaby      3   M  24.0     NaN     NaN         Denmark  DEN
Edgar Lindenau Aabye     4   M  34.0     NaN     NaN  Denmark/Sweden  DEN
Christine Jacoba Aaftink 5   F  21.0   185.0    82.0     Netherlands  NED

                                Games  Year  Season       City          Sport  \
Name
A Dijiang                1992 Summer  1992  Summer  Barcelona     Basketball
A Lamusi                 2012 Summer  2012  Summer     London           Judo
Gunnar Nielsen Aaby      1920 Summer  1920  Summer  Antwerpen       Football
Edgar Lindenau Aabye     1900 Summer  1900  Summer      Paris     Tug-Of-War
Christine Jacoba Aaftink 1988 Winter  1988  Winter    Calgary  Speed Skating

                                                  Event Medal
Name
A Dijiang                      Basketball Men's Basketball   NaN
A Lamusi                   Judo Men's Extra-Lightweight   NaN
Gunnar Nielsen Aaby             Football Men's Football   NaN
Edgar Lindenau Aabye         Tug-Of-War Men's Tug-Of-War  Gold
```

```
Christine Jacoba Aaftink  Speed Skating Women's 500 metres   NaN
```

[57]:
```
ajout_une_ligne.drop("Claire Muller", axis=0, inplace=True)
ajout_une_ligne.tail()
#on supprime la dernière ligne Claire Muller qu'on a ajouté précédemment
```

[57]:
```
                        ID Sex   Age  Height  Weight      Team  NOC  \
Name
Andrzej ya          135569   M  29.0   179.0    89.0  Poland-1  POL
Piotr ya            135570   M  27.0   176.0    59.0    Poland  POL
Piotr ya            135570   M  27.0   176.0    59.0    Poland  POL
Tomasz Ireneusz ya  135571   M  30.0   185.0    96.0    Poland  POL
Tomasz Ireneusz ya  135571   M  34.0   185.0    96.0    Poland  POL

                           Games  Year  Season            City        Sport  \
Name
Andrzej ya          1976 Winter  1976  Winter       Innsbruck         Luge
Piotr ya            2014 Winter  2014  Winter           Sochi  Ski Jumping
Piotr ya            2014 Winter  2014  Winter           Sochi  Ski Jumping
Tomasz Ireneusz ya  1998 Winter  1998  Winter          Nagano    Bobsleigh
Tomasz Ireneusz ya  2002 Winter  2002  Winter  Salt Lake City    Bobsleigh

                                                  Event Medal
Name
Andrzej ya                        Luge Mixed (Men)'s Doubles   NaN
Piotr ya            Ski Jumping Men's Large Hill, Individual   NaN
Piotr ya                  Ski Jumping Men's Large Hill, Team   NaN
Tomasz Ireneusz ya                       Bobsleigh Men's Four   NaN
Tomasz Ireneusz ya                       Bobsleigh Men's Four   NaN
```

3.3.4 3.3.4. Modifier des valeurs dans un dataframe

[58]:
```
donnees_modif=donnees.copy()
donnees_modif.loc["A Dijiang", "Age"] = 25
donnees_modif.head()
#dataframe.loc["étiquette de ligne","nom de colonne"]=valeur
#dataframe.iloc[position ligne,position colonne]=valeur
```

[58]:
```
                        ID Sex   Age  Height  Weight            Team  NOC  \
Name
A Dijiang                1   M  25.0   180.0    80.0           China  CHN
A Lamusi                 2   M  23.0   170.0    60.0           China  CHN
Gunnar Nielsen Aaby      3   M  24.0     NaN     NaN         Denmark  DEN
Edgar Lindenau Aabye     4   M  34.0     NaN     NaN  Denmark/Sweden  DEN
Christine Jacoba Aaftink 5   F  21.0   185.0    82.0     Netherlands  NED

                           Games  Year  Season       City          Sport  \
```

```
Name
A Dijiang                 1992 Summer  1992  Summer  Barcelona     Basketball
A Lamusi                  2012 Summer  2012  Summer     London           Judo
Gunnar Nielsen Aaby       1920 Summer  1920  Summer  Antwerpen       Football
Edgar Lindenau Aabye      1900 Summer  1900  Summer      Paris     Tug-Of-War
Christine Jacoba Aaftink  1988 Winter  1988  Winter    Calgary  Speed Skating

                                                    Event Medal
Name
A Dijiang                       Basketball Men's Basketball   NaN
A Lamusi                      Judo Men's Extra-Lightweight   NaN
Gunnar Nielsen Aaby                Football Men's Football   NaN
Edgar Lindenau Aabye             Tug-Of-War Men's Tug-Of-War  Gold
Christine Jacoba Aaftink  Speed Skating Women's 500 metres   NaN
```

[59]:
```
donnees_modif=donnees.copy()
donnees_modif.loc[(donnees_modif["Height"] >= 170) & (donnees_modif["Height"]␣
 ↪<= 180), "Height"] = 175
donnees_modif.head()
#on demande de sélectionner les lignes dont la valeur de colonne Height est␣
 ↪comprise entre 170 et 180 et
#de remplacer cette valeur par 175
```

[59]:
```
                          ID Sex   Age  Height  Weight            Team  NOC  \
Name
A Dijiang                  1   M  24.0   175.0    80.0           China  CHN
A Lamusi                   2   M  23.0   175.0    60.0           China  CHN
Gunnar Nielsen Aaby        3   M  24.0     NaN     NaN         Denmark  DEN
Edgar Lindenau Aabye       4   M  34.0     NaN     NaN  Denmark/Sweden  DEN
Christine Jacoba Aaftink   5   F  21.0   185.0    82.0     Netherlands  NED

                                Games  Year  Season       City          Sport  \
Name
A Dijiang                 1992 Summer  1992  Summer  Barcelona     Basketball
A Lamusi                  2012 Summer  2012  Summer     London           Judo
Gunnar Nielsen Aaby       1920 Summer  1920  Summer  Antwerpen       Football
Edgar Lindenau Aabye      1900 Summer  1900  Summer      Paris     Tug-Of-War
Christine Jacoba Aaftink  1988 Winter  1988  Winter    Calgary  Speed Skating

                                                    Event Medal
Name
A Dijiang                       Basketball Men's Basketball   NaN
A Lamusi                      Judo Men's Extra-Lightweight   NaN
Gunnar Nielsen Aaby                Football Men's Football   NaN
Edgar Lindenau Aabye             Tug-Of-War Men's Tug-Of-War  Gold
Christine Jacoba Aaftink  Speed Skating Women's 500 metres   NaN
```

[60]:
```
donnees_modif=donnees.copy()
donnees_modif.replace(to_replace="China", value="Chine", inplace=True)
donnees_modif.head()
#permet de remplacer le mot China par Chine pour l'ensemble du dataframe
```

[60]:

Name	ID	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
A Dijiang	1	M	24.0	180.0	80.0	Chine	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
A Lamusi	2	M	23.0	170.0	60.0	Chine	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
Gunnar Nielsen Aaby	3	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
Edgar Lindenau Aabye	4	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
Christine Jacoba Aaftink	5	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

4 3.4. Nettoyage et préparation des données avec Pandas

4.1 3.4.1. Gestion des données manquantes

[61]:
```
import pandas as pd
donnees=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", index_col=[1])
donnees.head()
```

[61]:

Name	ID	Sex	Age	Height	Weight	Team	NOC
A Dijiang	1	M	24.0	180.0	80.0	China	CHN
A Lamusi	2	M	23.0	170.0	60.0	China	CHN
Gunnar Nielsen Aaby	3	M	24.0	NaN	NaN	Denmark	DEN
Edgar Lindenau Aabye	4	M	34.0	NaN	NaN	Denmark/Sweden	DEN
Christine Jacoba Aaftink	5	F	21.0	185.0	82.0	Netherlands	NED

```
                                Games  Year  Season       City          Sport  \
Name
A Dijiang                 1992 Summer  1992  Summer  Barcelona     Basketball
A Lamusi                  2012 Summer  2012  Summer     London           Judo
Gunnar Nielsen Aaby       1920 Summer  1920  Summer  Antwerpen       Football
Edgar Lindenau Aabye      1900 Summer  1900  Summer      Paris     Tug-Of-War
Christine Jacoba Aaftink  1988 Winter  1988  Winter    Calgary  Speed Skating

                                                    Event Medal
Name
A Dijiang                        Basketball Men's Basketball   NaN
A Lamusi                      Judo Men's Extra-Lightweight   NaN
Gunnar Nielsen Aaby                Football Men's Football   NaN
Edgar Lindenau Aabye            Tug-Of-War Men's Tug-Of-War  Gold
Christine Jacoba Aaftink  Speed Skating Women's 500 metres   NaN
```

[62]:
```
donnees.isna().sum()
#la méthode isna() va remplacer l'ensemble des valeurs du dataframe par True␣
 ↪(si la valeur est NaN) et False sinon
#la méthode sum() va compter le nombre de valeurs à True par colonnes
```

[62]:
```
ID             0
Sex            0
Age         9474
Height     60171
Weight     62875
Team           0
NOC            0
Games          0
Year           0
Season         0
City           0
Sport          0
Event          0
Medal     231333
dtype: int64
```

[63]:
```
donnees[donnees["Height"].notna()]
#la méthode dropna() permet de supprimer les lignes contenant un NaN dans au␣
 ↪moins une colonne ou toutes les colonnes
#dataframe.dropna(how="any") permet de spécifier si une seule valeur d'une␣
 ↪ligne est à NaN alors on supprime cette ligne
#dataframe.dropna(how="all") permet de spécifier si toutes les valeurs d'une␣
 ↪ligne sont à NaN alors on supprime cette ligne
#la méthode notna() supprime une ligne lorsqu'une colonne spécifique contient␣
 ↪NaN
```

```
#donnees["Height"].notna() va créer un objet de type Séries contenant True si␣
 ↪la valeur n'est pas manquante et False sinon
#on rajoute donnees[] ce qui permet de ne sélectionner que les lignes dont les␣
 ↪valeurs sont différentes de NaN dans la colonne "Height"
```

[63]:

```
                              ID Sex   Age  Height  Weight         Team  NOC  \
Name
A Dijiang                      1   M  24.0   180.0    80.0        China  CHN
A Lamusi                       2   M  23.0   170.0    60.0        China  CHN
Christine Jacoba Aaftink       5   F  21.0   185.0    82.0  Netherlands  NED
Christine Jacoba Aaftink       5   F  21.0   185.0    82.0  Netherlands  NED
Christine Jacoba Aaftink       5   F  25.0   185.0    82.0  Netherlands  NED
…                            …  ..   …      …        …           …  …
Andrzej ya                135569   M  29.0   179.0    89.0     Poland-1  POL
Piotr ya                  135570   M  27.0   176.0    59.0       Poland  POL
Piotr ya                  135570   M  27.0   176.0    59.0       Poland  POL
Tomasz Ireneusz ya        135571   M  30.0   185.0    96.0       Poland  POL
Tomasz Ireneusz ya        135571   M  34.0   185.0    96.0       Poland  POL

                                Games  Year  Season            City  \
Name
A Dijiang                 1992 Summer  1992  Summer       Barcelona
A Lamusi                  2012 Summer  2012  Summer          London
Christine Jacoba Aaftink  1988 Winter  1988  Winter         Calgary
Christine Jacoba Aaftink  1988 Winter  1988  Winter         Calgary
Christine Jacoba Aaftink  1992 Winter  1992  Winter     Albertville
…                               …     …      …               …
Andrzej ya                1976 Winter  1976  Winter       Innsbruck
Piotr ya                  2014 Winter  2014  Winter           Sochi
Piotr ya                  2014 Winter  2014  Winter           Sochi
Tomasz Ireneusz ya        1998 Winter  1998  Winter          Nagano
Tomasz Ireneusz ya        2002 Winter  2002  Winter  Salt Lake City

                                  Sport  \
Name
A Dijiang                    Basketball
A Lamusi                           Judo
Christine Jacoba Aaftink  Speed Skating
Christine Jacoba Aaftink  Speed Skating
Christine Jacoba Aaftink  Speed Skating
…                                    …
Andrzej ya                         Luge
Piotr ya                   Ski Jumping
Piotr ya                   Ski Jumping
Tomasz Ireneusz ya           Bobsleigh
Tomasz Ireneusz ya           Bobsleigh
```

```
                                                                Event Medal
Name
A Dijiang                                 Basketball Men's Basketball   NaN
A Lamusi                                Judo Men's Extra-Lightweight   NaN
Christine Jacoba Aaftink          Speed Skating Women's 500 metres   NaN
Christine Jacoba Aaftink       Speed Skating Women's 1,000 metres   NaN
Christine Jacoba Aaftink          Speed Skating Women's 500 metres   NaN
…                                                               …   …
Andrzej ya                              Luge Mixed (Men)'s Doubles   NaN
Piotr ya                  Ski Jumping Men's Large Hill, Individual   NaN
Piotr ya                        Ski Jumping Men's Large Hill, Team   NaN
Tomasz Ireneusz ya                            Bobsleigh Men's Four   NaN
Tomasz Ireneusz ya                            Bobsleigh Men's Four   NaN

[210945 rows x 14 columns]
```

[64]:
```
donnees[['Age','Height', 'Weight']] = donnees[['Age','Height','Weight']].
 ↪fillna(donnees[['Age','Height','Weight']].mean())
donnees.head()
#la méthode fillna() remplace ici sur les 3 colonnes qui nous intéresse les␣
 ↪valeurs manquantes par la moyenne de la variable
#si on avait voulu remplacer l'ensemble des NaN par une certaine valeur alors␣
 ↪on écrit
#dataframe.fillna(value=0,inplace=True)
```

[64]:
```
                          ID Sex   Age      Height     Weight            Team  \
Name
A Dijiang                  1   M  24.0  180.00000  80.000000           China
A Lamusi                   2   M  23.0  170.00000  60.000000           China
Gunnar Nielsen Aaby        3   M  24.0  175.33897  70.702393         Denmark
Edgar Lindenau Aabye       4   M  34.0  175.33897  70.702393  Denmark/Sweden
Christine Jacoba Aaftink   5   F  21.0  185.00000  82.000000     Netherlands

                          NOC        Games  Year  Season       City  \
Name
A Dijiang                 CHN  1992 Summer  1992  Summer  Barcelona
A Lamusi                  CHN  2012 Summer  2012  Summer     London
Gunnar Nielsen Aaby       DEN  1920 Summer  1920  Summer  Antwerpen
Edgar Lindenau Aabye      DEN  1900 Summer  1900  Summer      Paris
Christine Jacoba Aaftink  NED  1988 Winter  1988  Winter    Calgary

                                    Sport                              Event  \
Name
A Dijiang                      Basketball       Basketball Men's Basketball
A Lamusi                           Judo      Judo Men's Extra-Lightweight
Gunnar Nielsen Aaby            Football           Football Men's Football
Edgar Lindenau Aabye         Tug-Of-War       Tug-Of-War Men's Tug-Of-War
```

```
Christine Jacoba Aaftink  Speed Skating  Speed Skating Women's 500 metres

                         Medal
Name
A Dijiang                  NaN
A Lamusi                   NaN
Gunnar Nielsen Aaby        NaN
Edgar Lindenau Aabye      Gold
Christine Jacoba Aaftink   NaN
```

4.1.1 3.4.2. Gestion des données dupliquées

[65]:
```
donnees[donnees.duplicated()]
#permet de détecter les lignes dupliquées dans le dataframe donnees
#forme générale dataframe[dataframe.duplicated(subset=[macolonne1,macolonne2,...
 ↪],keep='first')]
```

[65]:
```
                                        ID Sex   Age     Height     Weight  \
Name
Dsir Antoine Acket                     704   M  27.0  175.33897  70.702393
William Truman Aldrich                2449   M  48.0  175.33897  70.702393
William Truman Aldrich                2449   M  48.0  175.33897  70.702393
Hermann Reinhard Alker                2777   M  43.0  175.33897  70.702393
Hermann Reinhard Alker                2777   M  43.0  175.33897  70.702393
…                                      …  ..   …        …         …
Anna Katrina Zinkeisen (-Heseltine)  135072   F  46.0  175.33897  70.702393
Anna Katrina Zinkeisen (-Heseltine)  135072   F  46.0  175.33897  70.702393
Anna Katrina Zinkeisen (-Heseltine)  135072   F  46.0  175.33897  70.702393
Doris Clare Zinkeisen (-Johnstone)   135073   F  49.0  175.33897  70.702393
Henri Achille Zo                     135173   M  58.0  175.33897  70.702393

                                              Team  NOC        Games  Year  \
Name
Dsir Antoine Acket                         Belgium  BEL  1932 Summer  1932
William Truman Aldrich               United States  USA  1928 Summer  1928
William Truman Aldrich               United States  USA  1928 Summer  1928
Hermann Reinhard Alker                     Germany  GER  1928 Summer  1928
Hermann Reinhard Alker                     Germany  GER  1928 Summer  1928
…                                            …  …          …    …
Anna Katrina Zinkeisen (-Heseltine)  Great Britain  GBR  1948 Summer  1948
Anna Katrina Zinkeisen (-Heseltine)  Great Britain  GBR  1948 Summer  1948
Anna Katrina Zinkeisen (-Heseltine)  Great Britain  GBR  1948 Summer  1948
Doris Clare Zinkeisen (-Johnstone)   Great Britain  GBR  1948 Summer  1948
Henri Achille Zo                            France  FRA  1932 Summer  1932

                                     Season         City              Sport  \
Name
```

```
Dsir Antoine Acket                   Summer  Los Angeles  Art Competitions
William Truman Aldrich               Summer    Amsterdam  Art Competitions
William Truman Aldrich               Summer    Amsterdam  Art Competitions
Hermann Reinhard Alker               Summer    Amsterdam  Art Competitions
Hermann Reinhard Alker               Summer    Amsterdam  Art Competitions
…                                     …          …              …
Anna Katrina Zinkeisen (-Heseltine)  Summer       London  Art Competitions
Anna Katrina Zinkeisen (-Heseltine)  Summer       London  Art Competitions
Anna Katrina Zinkeisen (-Heseltine)  Summer       London  Art Competitions
Doris Clare Zinkeisen (-Johnstone)   Summer       London  Art Competitions
Henri Achille Zo                     Summer  Los Angeles  Art Competitions

 Event  \
Name
Dsir Antoine Acket                      Art Competitions Mixed Painting, Unknown
Event
William Truman Aldrich               Art Competitions Mixed Painting, Drawings
And …
William Truman Aldrich               Art Competitions Mixed Painting, Drawings
And …
Hermann Reinhard Alker               Art Competitions Mixed Architecture,
Designs F…
Hermann Reinhard Alker               Art Competitions Mixed Architecture,
Architect…
…
…
Anna Katrina Zinkeisen (-Heseltine)         Art Competitions Mixed Painting,
Paintings
Anna Katrina Zinkeisen (-Heseltine)         Art Competitions Mixed Painting,
Paintings
Anna Katrina Zinkeisen (-Heseltine)     Art Competitions Mixed Painting, Unknown
Event
Doris Clare Zinkeisen (-Johnstone)      Art Competitions Mixed Painting, Unknown
Event
Henri Achille Zo                        Art Competitions Mixed Painting, Unknown
Event

                                    Medal
Name
Dsir Antoine Acket                    NaN
William Truman Aldrich                NaN
William Truman Aldrich                NaN
Hermann Reinhard Alker                NaN
Hermann Reinhard Alker                NaN
…                                     …
Anna Katrina Zinkeisen (-Heseltine)   NaN
Anna Katrina Zinkeisen (-Heseltine)   NaN
```

```
Anna Katrina Zinkeisen (-Heseltine)   NaN
Doris Clare Zinkeisen (-Johnstone)    NaN
Henri Achille Zo                      NaN

[1385 rows x 14 columns]
```

[66]:
```
donnees[donnees.duplicated()].shape
```

[66]:
```
(1385, 14)
```

[67]:
```
donnees.drop_duplicates(inplace=True)
#permet de nettoyer le dataframe, on supprime toutes les lignes dupliquées sauf␣
 ↪la première occurence
donnees.shape
#on remarquera que 269731+1385=271116...
```

[67]:
```
(269731, 14)
```

4.2 3.5. Exploration préliminaire d'un dataframe

4.2.1 3.5.1. Les principaux attributs

[68]:
```
donnees.columns
#donne la liste des noms du dataframe
```

[68]:
```
Index(['ID', 'Sex', 'Age', 'Height', 'Weight', 'Team', 'NOC', 'Games', 'Year',
       'Season', 'City', 'Sport', 'Event', 'Medal'],
      dtype='object')
```

[69]:
```
type(donnees.values)
#surprise ici donnees.values génère un ndarray NumPy
```

[69]:
```
numpy.ndarray
```

[70]:
```
donnees.shape
#dimension de donnees
```

[70]:
```
(269731, 14)
```

[71]:
```
donnees.dtypes
```

[71]:
```
ID          int64
Sex        object
Age       float64
Height    float64
Weight    float64
Team       object
```

```
NOC        object
Games      object
Year        int64
Season     object
City       object
Sport      object
Event      object
Medal      object
dtype: object
```

4.2.2 3.5.2. Définition des termes variable, variable quantitative et variable qualitative et découverte de la méthode describe()

[72]:
```
donnees=pd.read_csv("../datasets/
 ↪120-years-of-olympic-history-athletes-and-results.csv", index_col=[1])
donnees.drop_duplicates(inplace=True)
donnees.head()
```

[72]:

Name	ID	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
A Dijiang	1	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
A Lamusi	2	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
Gunnar Nielsen Aaby	3	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
Edgar Lindenau Aabye	4	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
Christine Jacoba Aaftink	5	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

[73]:
```
donnees.describe()
#on remarquera que seules les données numériques (variables quantitatives) ont␣
 ↪des statistiques
```

[73]:
```
                 ID            Age         Height         Weight  \
count  269731.000000  260416.000000  210917.000000  208204.000000
mean    68264.949591      25.454776     175.338953      70.701778
std     39026.253843       6.163869      10.518507      14.349027
min         1.000000      10.000000     127.000000      25.000000
25%     34655.500000      21.000000     168.000000      60.000000
50%     68233.000000      24.000000     175.000000      70.000000
75%    102111.000000      28.000000     183.000000      79.000000
max    135571.000000      97.000000     226.000000     214.000000

                Year
count  269731.000000
mean     1978.623073
std        29.752055
min      1896.000000
25%      1960.000000
50%      1988.000000
75%      2002.000000
max      2016.000000
```

[74]:
```
donnees.describe(include="all")
#en rajoutant include='all' toutes les colonnes ont des stats (y compris les␣
 ↪variables qualitatives)
#unique donne le nombre de valeurs uniques dans la colonne
#top affiche la valeur la plus présente dans la colonne
#freq donne le nombre d'occurences de la valeur la plus fréquente dans la␣
 ↪colonne
```

[74]:
```
                  ID     Sex            Age         Height         Weight  \
count   269731.000000  269731  260416.000000  210917.000000  208204.000000
unique            NaN       2            NaN            NaN            NaN
top               NaN       M            NaN            NaN            NaN
freq              NaN  195353            NaN            NaN            NaN
mean     68264.949591     NaN      25.454776     175.338953      70.701778
std      39026.253843     NaN       6.163869      10.518507      14.349027
min          1.000000     NaN      10.000000     127.000000      25.000000
25%      34655.500000     NaN      21.000000     168.000000      60.000000
50%      68233.000000     NaN      24.000000     175.000000      70.000000
75%     102111.000000     NaN      28.000000     183.000000      79.000000
max     135571.000000     NaN      97.000000     226.000000     214.000000

                 Team     NOC        Games           Year  Season    City  \
count          269731  269731       269731  269731.000000  269731  269731
unique           1184     230           51            NaN       2      42
top     United States     USA  2000 Summer            NaN  Summer  London
freq            17598   18604        13821            NaN  221167   22297
mean              NaN     NaN          NaN    1978.623073     NaN     NaN
```

```
std              NaN     NaN          NaN      29.752055     NaN     NaN
min              NaN     NaN          NaN    1896.000000     NaN     NaN
25%              NaN     NaN          NaN    1960.000000     NaN     NaN
50%              NaN     NaN          NaN    1988.000000     NaN     NaN
75%              NaN     NaN          NaN    2002.000000     NaN     NaN
max              NaN     NaN          NaN    2016.000000     NaN     NaN
```

```
             Sport                    Event  Medal
count       269731                   269731  39772
unique          66                      765      3
top     Athletics  Football Men's Football   Gold
freq         38624                    5733  13369
mean           NaN                     NaN    NaN
std            NaN                     NaN    NaN
min            NaN                     NaN    NaN
25%            NaN                     NaN    NaN
50%            NaN                     NaN    NaN
75%            NaN                     NaN    NaN
max            NaN                     NaN    NaN
```

4.2.3 3.5.3. Les méthodes de tri d'un dataframe

[75]:
```
#on peut trier un dataframe soit par ses valeurs (sort_values()), soit par ses␣
 ↪index (sort_values())
donnees.sort_index(axis=1).head()
#on trie les données sur l'axe 1 (nom des colonnes)
#puis on affiche les 5 premières lignes avec la méthode head()
#donnees.sort_index(axis=0) permet de trier les données sur l'axe 0 (étiquettes␣
 ↪de lignes)
#on remarquera que les colonnes sont bien triées par ordre alphabétique
```

[75]:
```
                          Age       City                          Event  \
Name
A Dijiang                24.0  Barcelona       Basketball Men's Basketball
A Lamusi                 23.0     London      Judo Men's Extra-Lightweight
Gunnar Nielsen Aaby      24.0  Antwerpen           Football Men's Football
Edgar Lindenau Aabye     34.0      Paris       Tug-Of-War Men's Tug-Of-War
Christine Jacoba Aaftink 21.0    Calgary  Speed Skating Women's 500 metres

                            Games  Height  ID Medal  NOC  Season Sex  \
Name
A Dijiang                1992 Summer   180.0   1   NaN  CHN  Summer   M
A Lamusi                 2012 Summer   170.0   2   NaN  CHN  Summer   M
Gunnar Nielsen Aaby      1920 Summer     NaN   3   NaN  DEN  Summer   M
Edgar Lindenau Aabye     1900 Summer     NaN   4  Gold  DEN  Summer   M
Christine Jacoba Aaftink 1988 Winter   185.0   5   NaN  NED  Winter   F
```

```
                                Sport            Team  Weight  Year
Name
A Dijiang                  Basketball           China    80.0  1992
A Lamusi                         Judo           China    60.0  2012
Gunnar Nielsen Aaby           Football         Denmark     NaN  1920
Edgar Lindenau Aabye       Tug-Of-War  Denmark/Sweden     NaN  1900
Christine Jacoba Aaftink  Speed Skating     Netherlands    82.0  1988
```

[76]:
```
donnees.sort_values("Weight", axis = 0, na_position="last").head()
#on effectue un trie sur les valeurs de la colonne "Weight"
#na_position="last" permet de mettre les valeurs manquantes (NaN) soient mises␣
 ↪à la fin du tableau trié
```

[76]:
```
                   ID Sex   Age  Height  Weight         Team  NOC  \
Name
Choi Myong-Hui  21049   F  14.0   135.0    25.0  North Korea  PRK
Choi Myong-Hui  21049   F  14.0   135.0    25.0  North Korea  PRK
Choi Myong-Hui  21049   F  14.0   135.0    25.0  North Korea  PRK
Choi Myong-Hui  21049   F  14.0   135.0    25.0  North Korea  PRK
Choi Myong-Hui  21049   F  14.0   135.0    25.0  North Korea  PRK

                       Games  Year  Season    City       Sport  \
Name
Choi Myong-Hui  1980 Summer  1980  Summer  Moskva  Gymnastics
Choi Myong-Hui  1980 Summer  1980  Summer  Moskva  Gymnastics
Choi Myong-Hui  1980 Summer  1980  Summer  Moskva  Gymnastics
Choi Myong-Hui  1980 Summer  1980  Summer  Moskva  Gymnastics
Choi Myong-Hui  1980 Summer  1980  Summer  Moskva  Gymnastics

                                                  Event Medal
Name
Choi Myong-Hui            Gymnastics Women's Uneven Bars   NaN
Choi Myong-Hui           Gymnastics Women's Balance Beam   NaN
Choi Myong-Hui         Gymnastics Women's Floor Exercise   NaN
Choi Myong-Hui        Gymnastics Women's Team All-Around   NaN
Choi Myong-Hui  Gymnastics Women's Individual All-Around   NaN
```

[77]:
```
donnees.sort_values("Weight", axis = 0, na_position="last", ascending=False).
 ↪head()
#en rajoutant ascending=False on trie les poids par ordre décroissant␣
 ↪"ascending" est à True par défaut
```

[77]:
```
                                ID Sex   Age  Height  Weight  \
Name
Ricardo Blas, Jr.            12177   M  21.0   183.0   214.0
Ricardo Blas, Jr.            12177   M  25.0   183.0   214.0
Aytami Ruano Vega           103159   M  27.0   200.0   198.0
```

```
Marek Galiski                    38075   M  29.0   200.0   190.0
Christopher J. "Chris" Taylor   118869   M  22.0   196.0   182.0

                                          Team  NOC        Games  Year  Season  \
Name
Ricardo Blas, Jr.                         Guam  GUM  2008 Summer  2008  Summer
Ricardo Blas, Jr.                         Guam  GUM  2012 Summer  2012  Summer
Aytami Ruano Vega                        Spain  ESP  2004 Summer  2004  Summer
Marek Galiski                           Poland  POL  1980 Summer  1980  Summer
Christopher J. "Chris" Taylor  United States  USA  1972 Summer  1972  Summer

                                    City      Sport  \
Name
Ricardo Blas, Jr.                Beijing       Judo
Ricardo Blas, Jr.                 London       Judo
Aytami Ruano Vega                 Athina       Judo
Marek Galiski                     Moskva  Wrestling
Christopher J. "Chris" Taylor     Munich  Wrestling

                                                                        Event  \
Name
Ricardo Blas, Jr.                                          Judo Men's Heavyweight
Ricardo Blas, Jr.                                          Judo Men's Heavyweight
Aytami Ruano Vega                                          Judo Men's Heavyweight
Marek Galiski                   Wrestling Men's Super-Heavyweight, Greco-Roman
Christopher J. "Chris" Taylor   Wrestling Men's Super-Heavyweight, Greco-Roman

                               Medal
Name
Ricardo Blas, Jr.                NaN
Ricardo Blas, Jr.                NaN
Aytami Ruano Vega                NaN
Marek Galiski                    NaN
Christopher J. "Chris" Taylor    NaN
```

5 4. Structure de données Pandas : les panels

Les panels peuvent être décrits comme des tableaux à trois dimensions...on ne les abordera pas dans ce TP.

6 5. Manipulation avancée des données avec Pandas

6.1 5.1. Les opérations groupby

6.1.1 5.1.1. groupby sur une colonne

[78]:
```
donnees.groupby("Sport").mean()
#il est possible de grouper plusieurs colonnes ou lignes puis d'y appliquer des␣
 ↪fonctions dessus
#groupby("nom_de_colonne").mean() retourne un dataframe avec, en index de␣
 ↪ligne, chaque modalité de la colonne nom_de_colonne
#ainsi que la moyenne de chaque colonne numérique pour chaque modalité de␣
 ↪nom_de_colonne
```

[78]:

Sport	ID	Age	Height	Weight	Year
Aeronautics	107506.000000	26.000000	NaN	NaN	1936.000000
Alpine Skiing	66487.294371	23.205462	173.489052	72.068110	1985.203534
Alpinism	79639.320000	38.812500	NaN	NaN	1925.600000
Archery	71015.393316	27.935226	173.203085	70.011135	1987.533847
Art Competitions	67506.074680	44.719596	174.343750	77.894737	1933.756960
…	…	…	…	…	…
Tug-Of-War	70270.747059	29.309524	182.480000	95.615385	1909.552941
Volleyball	70635.202996	25.183800	186.994822	78.900214	1992.202115
Water Polo	68645.384295	25.659627	184.834648	84.566446	1975.624545
Weightlifting	69105.432563	25.502010	167.824801	78.726663	1980.595377
Wrestling	67972.349036	25.798289	172.358586	75.495570	1972.972044

[66 rows x 5 columns]

[79]:
```
donnees.groupby("Sport").mean().loc[:,['Age','Height','Weight']]
#on peut faire du slicing après le calcul de la moyenne pour ne faire afficher␣
 ↪que certaines colonnes
#on peut utiliser l'indexeur loc après la focntion mean() car le "." permet de␣
 ↪récupérer l'objet créé par la méthode
#ou fonction, ce qui permet d'enchaîner de manière très efficace les actions␣
 ↪sur une seule ligne
```

[79]:

Sport	Age	Height	Weight
Aeronautics	26.000000	NaN	NaN
Alpine Skiing	23.205462	173.489052	72.068110
Alpinism	38.812500	NaN	NaN
Archery	27.935226	173.203085	70.011135
Art Competitions	44.719596	174.343750	77.894737
…	…	…	…
Tug-Of-War	29.309524	182.480000	95.615385
Volleyball	25.183800	186.994822	78.900214

```
Water Polo          25.659627  184.834648  84.566446
Weightlifting       25.502010  167.824801  78.726663
Wrestling           25.798289  172.358586  75.495570

[66 rows x 3 columns]
```

6.1.2 5.1.2. groupby sur plusieurs colonnes

[80]:
```
donnees.groupby(["Sport", "Sex"]).mean().loc[:,['Age','Height','Weight']]
#on peut effectuer un regroupement sur plusieurs colonnes
#on remarque rapidement que certains sports comme le Water Polo ne possèdent␣
 ↪pas d'équipe féminine
```

[80]:
```
                          Age      Height     Weight
Sport         Sex
Aeronautics   M     26.000000         NaN        NaN
Alpine Skiing F     22.334609  167.221001  62.640307
              M     23.758266  177.891374  78.626035
Alpinism      F     43.000000         NaN        NaN
              M     38.533333         NaN        NaN
…                         …         …          …
Water Polo    M     25.736542  186.801739  87.706172
Weightlifting F     24.028078  160.467391  67.724622
              M     25.710832  169.153061  80.251796
Wrestling     F     25.305921  163.865132  60.554455
              M     25.821827  172.870686  76.400640

[116 rows x 3 columns]
```

6.1.3 5.1.3. Appliquer plusieurs fonctions avec la méthode groupby et la méthode aggregate

[81]:
```
donnees.groupby(['Sport', 'Sex']).agg(['median','mean']).loc[:
 ↪,['Age','Height','Weight']]
#la méthode aggregate (dont l'alias est .agg()) permet d'appliquer plusieurs␣
 ↪fonctions de calculs et pas uniquement une seule
#qu'on souhaite appliquer au dataframe groupé par modalité  de la (ou des)␣
 ↪colonne(s)
#ici on calcule la moyenne et la médiane pour chaque colonne numérique puis on␣
 ↪récupère que les 3 colonnes qui nous intéressent
```

```
<ipython-input-81-5e21a3577808>:1: FutureWarning: ['Team', 'NOC', 'Games',
'Season', 'City', 'Event', 'Medal'] did not aggregate successfully. If any error
is raised this will raise in a future version of pandas. Drop these columns/ops
to avoid this warning.
  donnees.groupby(['Sport',
'Sex']).agg(['median','mean']).loc[:,['Age','Height','Weight']]
```

[81]:

		Age		Height		Weight	
		median	mean	median	mean	median	mean
Sport	Sex						
Aeronautics	M	26.0	26.000000	NaN	NaN	NaN	NaN
Alpine Skiing	F	22.0	22.334609	168.0	167.221001	62.0	62.640307
	M	23.0	23.758266	178.0	177.891374	78.0	78.626035
Alpinism	F	43.0	43.000000	NaN	NaN	NaN	NaN
	M	37.0	38.533333	NaN	NaN	NaN	NaN
…		…	…	…	…	…	
Water Polo	M	25.0	25.736542	187.0	186.801739	87.0	87.706172
Weightlifting	F	23.0	24.028078	160.0	160.467391	63.0	67.724622
	M	25.0	25.710832	170.0	169.153061	75.0	80.251796
Wrestling	F	25.0	25.305921	164.0	163.865132	61.0	60.554455
	M	25.0	25.821827	172.0	172.870686	74.0	76.400640

[116 rows x 6 columns]

[82]:
```
donnees.groupby(['Sport', 'Sex']).agg({'Age' : ['median','mean'], 'Height' :␣
↪['median', 'sum'], 'Weight' : ['mean','median']})
#on peut aussi ne pas appliquer les mêmes fonctions à toutes les colonnes
#ici on applique les fonctons median et mean à la variable Age, et les␣
↪fonctions median et sum à la vaiable Height
```

[82]:

		Age		Height		Weight	
		median	mean	median	sum	mean	median
Sport	Sex						
Aeronautics	M	26.0	26.000000	NaN	0.0	NaN	NaN
Alpine Skiing	F	22.0	22.334609	168.0	441129.0	62.640307	62.0
	M	23.0	23.758266	178.0	668160.0	78.626035	78.0
Alpinism	F	43.0	43.000000	NaN	0.0	NaN	NaN
	M	37.0	38.533333	NaN	0.0	NaN	NaN
…		…	…	…	…	…	
Water Polo	M	25.0	25.736542	187.0	429644.0	87.706172	87.0
Weightlifting	F	23.0	24.028078	160.0	73815.0	67.724622	63.0
	M	25.0	25.710832	170.0	431002.0	80.251796	75.0
Wrestling	F	25.0	25.305921	164.0	49815.0	60.554455	61.0
	M	25.0	25.821827	172.0	871614.0	76.400640	74.0

[116 rows x 6 columns]

6.2 5.2. Appliquer une fonction à un dataframe avec la méthode apply

[83]:
```
import numpy as np
print(donnees.loc[:,["Age", "Height", "Weight"]].apply(np.mean, axis=0))
#la méthode apply() permet d'appliquer une fonction sur un axe du dataframe␣
↪(axis=0 : colonne, axis=1 : ligne)
#par défaut la méthode apply appliquera la fonction par colonne
```

```
#ici on calcule la moyenne pour les colonnes Age, Height et Weight
```

```
Age        25.454776
Height    175.338953
Weight     70.701778
dtype: float64
```

[84]:
```
print(donnees[["Age", "Height", "Weight"]].apply(lambda x : x * 10))
#ce qui est intéressant avec la méthode apply() est de pouvoir créer sa propre␣
 ↪fonction
#on va créer une fonction qu'on appelle une fonction lambda, qu'on utilisera␣
 ↪qu'une fois
#ici on a créé une fonction lambda qui va multiplier l'ensemble des valeurs␣
 ↪qu'elle reçoit en entrée , x, par 10
```

```
                           Age  Height  Weight
Name
A Dijiang                240.0  1800.0   800.0
A Lamusi                 230.0  1700.0   600.0
Gunnar Nielsen Aaby      240.0     NaN     NaN
Edgar Lindenau Aabye     340.0     NaN     NaN
Christine Jacoba Aaftink 210.0  1850.0   820.0
…                        …      …       …
Andrzej ya               290.0  1790.0   890.0
Piotr ya                 270.0  1760.0   590.0
Piotr ya                 270.0  1760.0   590.0
Tomasz Ireneusz ya       300.0  1850.0   960.0
Tomasz Ireneusz ya       340.0  1850.0   960.0

[269731 rows x 3 columns]
```

[85]:
```
donnees['Height'] = donnees['Height'].apply(lambda x : x / 100)
donnees.head()
#dans ce cas pratique, on veut que les tailles affichées des athlètes qui sont␣
 ↪en centimètres soient affichées en mètres
```

[85]:
```
                          ID Sex   Age  Height  Weight            Team  NOC  \
Name
A Dijiang                  1   M  24.0    1.80    80.0           China  CHN
A Lamusi                   2   M  23.0    1.70    60.0           China  CHN
Gunnar Nielsen Aaby        3   M  24.0     NaN     NaN         Denmark  DEN
Edgar Lindenau Aabye       4   M  34.0     NaN     NaN  Denmark/Sweden  DEN
Christine Jacoba Aaftink   5   F  21.0    1.85    82.0     Netherlands  NED

                              Games  Year  Season       City          Sport  \
Name
A Dijiang                1992 Summer  1992  Summer  Barcelona     Basketball
```

```
A Lamusi                  2012 Summer  2012  Summer     London           Judo
Gunnar Nielsen Aaby       1920 Summer  1920  Summer   Antwerpen       Football
Edgar Lindenau Aabye      1900 Summer  1900  Summer      Paris     Tug-Of-War
Christine Jacoba Aaftink  1988 Winter  1988  Winter    Calgary  Speed Skating
```

```
                                                  Event Medal
Name
A Dijiang                      Basketball Men's Basketball   NaN
A Lamusi                     Judo Men's Extra-Lightweight   NaN
Gunnar Nielsen Aaby               Football Men's Football   NaN
Edgar Lindenau Aabye           Tug-Of-War Men's Tug-Of-War  Gold
Christine Jacoba Aaftink  Speed Skating Women's 500 metres   NaN
```

6.3 5.3. Remodeler / ré-organiser des dataframes

6.3.1 5.3.1. Pivotage : la méthode pivot_table (les tableaux croisés dynamiques)

[86]:
```
import pandas as pd
pd.set_option('display.max_rows', 70)
print(donnees.pivot_table(index='Sport', columns='Medal', values="Sex",␣
 ↪aggfunc='max'))
#pd.set_option('display.max_rows', 10)
#on souhaite que pour chaque sport de la variable Sport et pour chaque type de␣
 ↪médaille (or, argent, bronze) de la variable Medal
#savoir si le genre prédominant des athlètes les ayant obtenus
#on va créer un tabeau croisé dynamique avec la colonne Sport qui sera note␣
 ↪option index, la colonne Medal sera notre option columns
#la variable Sex sera notre option values
#la commande pd.set_option('display.max_rows', 70) permet de dire à Pandas␣
 ↪d'afficher les 70 premières lignes du dataframe
#sachant qu'il y a 66 sports cela permet d'afficher le tableau entier en sortie␣
 ↪de méthode pivot_table()
#l'option aggfunc() permet de dire quelle fonction d'agrégation on souhaite␣
 ↪utiliser
```

```
Medal                Bronze Gold Silver
Sport
Aeronautics             NaN    M    NaN
Alpine Skiing             M    M      M
Alpinism                NaN    M    NaN
Archery                   M    M      M
Art Competitions          M    M      M
Athletics                 M    M      M
Badminton                 M    M      M
Baseball                  M    M      M
Basketball                M    M      M
Basque Pelota           NaN    M    NaN
```

Beach Volleyball	M	M	M
Biathlon	M	M	M
Bobsleigh	M	M	M
Boxing	M	M	M
Canoeing	M	M	M
Cricket	NaN	M	M
Croquet	M	M	M
Cross Country Skiing	M	M	M
Curling	M	M	M
Cycling	M	M	M
Diving	M	M	M
Equestrianism	M	M	M
Fencing	M	M	M
Figure Skating	M	M	M
Football	M	M	M
Freestyle Skiing	M	M	M
Golf	M	M	M
Gymnastics	M	M	M
Handball	M	M	M
Hockey	M	M	M
Ice Hockey	M	M	M
Jeu De Paume	M	M	M
Judo	M	M	M
Lacrosse	M	M	M
Luge	M	M	M
Military Ski Patrol	M	M	M
Modern Pentathlon	M	M	M
Motorboating	NaN	M	NaN
Nordic Combined	M	M	M
Polo	M	M	M
Racquets	M	M	M
Rhythmic Gymnastics	F	F	F
Roque	M	M	M
Rowing	M	M	M
Rugby	M	M	M
Rugby Sevens	M	M	M
Sailing	M	M	M
Shooting	M	M	M
Short Track Speed Skating	M	M	M
Skeleton	M	M	M
Ski Jumping	M	M	M
Snowboarding	M	M	M
Softball	F	F	F
Speed Skating	M	M	M
Swimming	M	M	M
Synchronized Swimming	F	F	F
Table Tennis	M	M	M
Taekwondo	M	M	M

```
Tennis                          M    M      M
Trampolining                    M    M      M
Triathlon                       M    M      M
Tug-Of-War                      M    M      M
Volleyball                      M    M      M
Water Polo                      M    M      M
Weightlifting                   M    M      M
Wrestling                       M    M      M
```

6.3.2 5.3.2. Les méthodes stack (empiler) et unstack (désempiler)

[87]:
```
multi_index=donnees.groupby(["Sport", "Sex"]).mean()[['Age','Height','Weight']]
print("Sans stack")
print(multi_index)
print("Après stack")
print(multi_index.stack())
#la méthode stack() permet de transformer l'index de colonne le plus intérieur␣
 ↪au dataframe en index de ligne
#ici il y a un seul index de colonne qui est (Age,Height,Weight)
#la méthode unstack() fait l'inverse de la méthode stack()
#en effet, elle permet de transformer l'index de ligne le plus intérieur au␣
 ↪dataframe en index de colonne
#ce qui permettra de réorganiser le tableau
```

```
Sans stack
                        Age    Height     Weight
Sport         Sex
Aeronautics   M    26.000000       NaN        NaN
Alpine Skiing F    22.334609  1.672210  62.640307
              M    23.758266  1.778914  78.626035
Alpinism      F    43.000000       NaN        NaN
              M    38.533333       NaN        NaN
…                        …       …         …
Water Polo    M    25.736542  1.868017  87.706172
Weightlifting F    24.028078  1.604674  67.724622
              M    25.710832  1.691531  80.251796
Wrestling     F    25.305921  1.638651  60.554455
              M    25.821827  1.728707  76.400640

[116 rows x 3 columns]
Après stack
Sport          Sex
Aeronautics    M    Age       26.000000
Alpine Skiing  F    Age       22.334609
                    Height     1.672210
                    Weight    62.640307
               M    Age       23.758266
```

```
                                 …
Wrestling      F    Height     1.638651
                    Weight    60.554455
               M    Age       25.821827
                    Height     1.728707
                    Weight    76.400640
Length: 324, dtype: float64
```