Thème: Traitement des données

20

Manipulation de fichiers CSV avec la bibliothèque Pandas

Le module csv utilisé précédemment se contente de lire les données structurées. Il ne fait aucun effort particulier pour analyser les données. Nous nous en sommes aperçus lorsqu'il a fallu convertir par int() toutes les valeurs numériques, qui étaient interprétées comme des chaînes de caractères.

La bibliothèque pandas est par contre spécialement conçue pour l'analyse des données (data analysis) : elle est donc naturellement bien plus performante.

Nous allons utiliser une base de données sur les joueurs NBA.

```
import pandas as pd #import du module pandas, abrégé classiquement par "pd"

df = pd.read_csv('data/players.csv', encoding = 'utf-8')
```

La variable est nommée classiquement df pour dataframe (que l'on peut traduire par table de données)

1. Premiers renseignements sur les fichiers de données

Que contient la variable df?

df

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
0	Juhann Begarin	SG	Boston Celtics	19	6' 5"	6.50	185	NaN	NaN	NaN	NaN	NaN
1	Jaylen Brown	SG	Boston Celtics	24	6' 6"	6.60	223	California	26758928.0	24.7	6.0	3.4
2	Kris Dunn	PG	Boston Celtics	27	6' 3"	6.30	205	Providence	5005350.0	1.3	1.5	0.5
3	Carsen Edwards	PG	Boston Celtics	23	5' 11"	5.11	200	Purdue	1782621.0	4.0	0.8	0.5
4	Tacko Fall	С	Boston Celtics	25	7' 5"	7.50	311	UCF	NaN	2.5	2.7	0.2
					•••							
553	Juwan Morgan	SF	Utah Jazz	24	6' 7"	6.70	232	Indiana	NaN	1.2	1.0	0.3
554	Royce O'Neale	PF	Utah Jazz	28	6' 4"	6.40	226	Baylor	0.0000088	7.0	6.8	2.5
555	Olumiye Oni	SG	Utah Jazz	24	6' 5"	6.50	206	Yale	1782621.0	NaN	NaN	NaN

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
556	Eric Paschall	F	Utah Jazz	24	6' 6"	6.60	255	Villanova	1782621.0	9.5	3.2	1.3
557	Hassan	С	Utah Jazz	32	7' 0"	7.00	265	Marshall	1669178.0	8.1	6.0	0.6
	Whiteside											

558 rows × 12 columns

Les données sont présentées dans l'ordre originel du fichier.

→ Il est possible d'avoir uniquement les premières lignes du fichier avec la commande head() et les dernières du fichier avec la commande tail(). Ces commandes peuvent recevoir en paramètre un nombre entier.

df.head()

	Nam	e Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
(Juhann Begarir	SG	Boston Celtics	19	6' 5"	6.50	185	NaN	NaN	NaN	NaN	NaN
1	L Jaylen Brown	SG	Boston Celtics	24	6' 6"	6.60	223	California	26758928.0	24.7	6.0	3.4
2	Kris Dunn	PG	Boston Celtics	27	6' 3"	6.30	205	Providence	5005350.0	1.3	1.5	0.5
3	Carsen Edward	s PG	Boston Celtics	23	5' 11"	5.11	200	Purdue	1782621.0	4.0	0.8	0.5
2	Tacko Fall	С	Boston Celtics	25	7' 5"	7.50	311	UCF	NaN	2.5	2.7	0.2

df.tail()

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
553	Juwan Morgan	SF	Utah Jazz	24	6' 7"	6.7	232	Indiana	NaN	1.2	1.0	0.3
554	Royce O'Neale	PF	Utah Jazz	28	6' 4"	6.4	226	Baylor	8800000.0	7.0	6.8	2.5
555	Olumiye Oni	SG	Utah Jazz	24	6' 5"	6.5	206	Yale	1782621.0	NaN	NaN	NaN
556	Eric Paschall	F	Utah Jazz	24	6' 6"	6.6	255	Villanova	1782621.0	9.5	3.2	1.3
557	Hassan Whiteside	С	Utah Jazz	32	7' 0"	7.0	265	Marshall	1669178.0	8.1	6.0	0.6

df.head(3)

		Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
() (Juhann Begarin	SG	Boston Celtics	19	6' 5"	6.5	185	NaN	NaN	NaN	NaN	NaN
	L J	Jaylen Brown	SG	Boston Celtics	24	6' 6"	6.6	223	California	26758928.0	24.7	6.0	3.4
4	2 k	Kris Dunn	PG	Boston Celtics	27	6' 3"	6.3	205	Providence	5005350.0	1.3	1.5	0.5

df.tail(5)

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
553	Juwan Morgan	SF	Utah Jazz	24	6' 7"	6.7	232	Indiana	NaN	1.2	1.0	0.3
554	Royce O'Neale	PF	Utah Jazz	28	6' 4"	6.4	226	Baylor	0.0000088	7.0	6.8	2.5
555	Olumiye Oni	SG	Utah Jazz	24	6' 5"	6.5	206	Yale	1782621.0	NaN	NaN	NaN
556	Eric Paschall	F	Utah Jazz	24	6' 6"	6.6	255	Villanova	1782621.0	9.5	3.2	1.3
557	Hassan Whiteside	С	Utah Jazz	32	7' 0"	7.0	265	Marshall	1669178.0	8.1	6.0	0.6

→ Pour avoir des renseignements globaux sur la structure de notre fichier, on peut utiliser la commande df.info()

df.info()

```
1 Position 558 non-null object
2 Team 558 non-null object
3 Age 558 non-null int64
4 Height 558 non-null object
5 Height_i 558 non-null float64
6 Weight 558 non-null int64
7 College 485 non-null object
8 Salary 445 non-null float64
9 Points 476 non-null float64
10 Rebounds 476 non-null float64
11 Assists 476 non-null float64
dtypes: float64(5), int64(2), object(5)
memory usage: 52.4+ KB
```

2. Extraction de colonnes

2.1 Pour accéder aux information d'une colonne

2.1.1 Filter les lignes en appliquant une condition aux colonnes à l'aide de .loc

→ Pour faire afficher une colonne de données :

```
df.loc[:,"Salary"]
0
    26758928.0
1
    5005350.0
2
3
     1782621.0
        NaN
4
    8800000.0
554
     1782621.0
555
      1782621.0
557
      1669178.0
Name: Salary, Length: 558, dtype: float64
```

→ On peut rajouter une contrainte avec, par exemple; uniquement les 10 premiers de la liste

```
df.loc[:10, "Salary"] # les 10 premiers dans la liste

0     NaN
1     26758928.0
2     5005350.0
3     1782621.0
4     NaN
5     1782621.0
6     27000000.0
7     1669178.0
8     NaN
9     3804360.0
10     3631200.0
Name: Salary, dtype: float64

df.loc[] # les 10 premiers dans la liste
```

A vous

Faite apparaitre la colonne correspondant au âge des joueurs, puis faite apparaitre les 15 premiers de la liste.

```
df.loc[:15, "Salary"]

0     NaN
1     26758928.0
```

```
2 5005350.0
    1782621.0
 3
 4
            NaN
    1782621.0
27000000.0
 5
 6
      1669178.0
 7
8 NaN
9 3804360.0
10 3631200.0
11 2283034.0
       2137440.0
 12
 13 11615328.0
 14
      5890000.0
 15 14339285.0
 Name: Salary, dtype: float64
```

→ Pour accéder à une fiche particulière de joueur avec sa position, on peut utiliser la fonction loc():

```
test=df.loc[45]
print(test)
test1=df.loc[45,['Name','Position']]
print(test1)
```

```
Name RJ Barrett
Position SG
Team New York Knicks
Age 21
Height 6'6"
Height_i 6.6
Weight 214
College Duke
Salary 8623920.0
Points 17.6
Rebounds 5.8
Assists 3.0
Name: 45, dtype: object
Name RJ Barrett
Position SG
Name: 45, dtype: object
```

2.1.2 Extraire des informations selon des critères loc()

Cette méthode nécessite de savoir la position du joueur voulu, pour accéder à un joueur avec son nom :

```
df.loc[df['Name']=='Kevin Durant'] # ou df.loc[df.Name=='Kevin Durant']
```

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
25	Kevin Durant	PF	Brooklyn Nets	32	6' 10"	6.1	240	Texas	42018900.0	26.9	7.1	5.6

→ Pour accéder à des données avec une contrainte (Joueurs touchant plus de 40 millions de dollars par an):

```
df.loc[df.Salary>=40000000]
```

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
25	Kevin Durant	PF	Brooklyn Nets	32	6' 10"	6.1	240	Texas	42018900.0	26.9	7.1	5.6
29	James Harden	SG	Brooklyn Nets	32	6' 5"	6.5		Arizona State	44310840.0	24.6	7.9	10.8
287	Stephen Curry		Golden State Warriors	33	6' 3"	6.3	185	Davidson	45780966.0	32.0	5.5	5.8
331	LeBron James	SF	Los Angeles Lakers	36	6' 9"	6.9	250	NaN	41180544.0	25.0	7.7	7.8
336	Russell Westbrook	PG	Los Angeles Lakers	32	6' 3"	6.3	200	UCLA	44211146.0	22.2	11.5	11.7

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
40	John Wall	PG	Houston	31	6' 3"	6.3	210	Kentucky	44310840.0	20.6	3.2	6.9
40	0		Rockets									

A vous

Donner les joueurs marquant plus de 25 points par match.

df.loc[df.Points>=27]

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
69	Joel Embiid	С	Philadelphia Sixers	27	7' 0"	7.00	280	Kansas	31579390.0	28.5	10.6	2.8
113	Zach LaVine	SG	Chicago Bulls	26	6' 5"	6.50	200	UCLA	19500000.0	27.4	5.0	4.9
171	Giannis Antetokounmpo	PF	Milwaukee Bucks	26	6' 11"	6.11	242	NaN	39344970.0	28.1	11.0	5.9
267	Bradley Beal	SG	Washington Wizards	28	6' 3"	6.30	207	Florida	33724200.0	31.3	4.7	4.4
287	Stephen Curry	PG	Golden State Warriors	33	6' 3"	6.30	185	Davidson	45780966.0	32.0	5.5	5.8
378	Luka Doncic	PG	Dallas Mavericks	22	6' 7"	6.70	230	NaN	10174391.0	27.7	8.0	8.6
449	Zion Williamson	PF	New Orleans Pelicans	21	6' 7"	6.70	284	Duke	10733400.0	27.0	7.2	3.7
529	Damian Lillard	PG	Portland Trail Blazers	31	6' 2"	6.20	195	Weber State	39344900.0	28.8	4.2	7.5

A vous

Donner les joueurs provenant du collège du Kentuchy.

df.loc[df.College=='Kentucky']

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
7	Enes Kanter	С	Boston Celtics	29	6' 10"	6.10	250	Kentucky	1669178.0	11.2	11.0	1.2
51	Kevin Knox II	SF	New York Knicks	22	6' 7"	6.70	215	Kentucky	5845979.0	3.9	1.5	0.5
53	Nerlens Noel	С	New York Knicks	27	6' 11"	6.11	220	Kentucky	8000000.0	5.1	6.4	0.7
57	Immanuel Quickley	SG	New York Knicks	22	6' 3"	6.30	190	Kentucky	2210640.0	11.4	2.1	2.0
58	Julius Randle	PF	New York Knicks	26	6' 8"	6.80	250	Kentucky	21780000.0	24.1	10.2	6.0
75	Tyrese Maxey	PG	Philadelphia Sixers	20	6' 2"	6.20	200	Kentucky	2602920.0	8.0	1.7	2.0
136	Hamidou Diallo	SG	Detroit Pistons	23	6' 5"	6.50	202	Kentucky	5200000.0	11.6	5.2	1.9
146	Trey Lyles	PF	Detroit Pistons	25	6' 9"	6.90	234	Kentucky	2500000.0	5.0	3.7	0.6
158	Isaiah Jackson	PF	Indiana Pacers	19	6' 10"	6.10	205	Kentucky	2451240.0	NaN	NaN	NaN
222	Nick Richards	С	Charlotte Hornets	23	7' 0"	7.00	245	Kentucky	1517981.0	8.0	0.6	0.1
1227	P.J. Washington	PF	Charlotte Hornets	23	6' 7"	6.70	230	Kentucky	4215120.0	12.9	6.5	2.5
228	Bam Adebayo	С	Miami Heat	24	6' 9"	6.90	255	Kentucky	28103500.0	18.7	9.0	5.4
233	Tyler Herro	PG	Miami Heat	21	6' 5"	6.50	195	Kentucky	4004280.0	15.1	5.0	3.4
295	Mychal Mulder	SG	Golden State Warriors	27	6' 3"	6.30	184	Kentucky	1782621.0	5.6	1.0	0.4
304	Eric Bledsoe	SG	Los Angeles Clippers	31		6.10	214	Kentucky	18125000.0	12.2	3.4	3.8

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists
306	DeMarcus Cousins	С	Los Angeles Clippers	31	6' 10"	6.10	270	Kentucky	NaN	8.9	6.4	1.9
316	Patrick Patterson	PF	Los Angeles Clippers	32	6' 8"	6.80	235	Kentucky	NaN	5.2	2.0	0.8
326	Anthony Davis	PF	Los Angeles Lakers	28	6' 10"	6.10	253	Kentucky	35361360.0	21.8	7.9	3.1
333	Malik Monk	SG	Los Angeles Lakers	23	6' 3"	6.30	200	Kentucky	1669178.0	11.7	2.4	2.1
335	Rajon Rondo	PG	Los Angeles Lakers	35	6' 1"	6.10	180	Kentucky	1669178.0	5.4	2.4	4.4
338	Devin Booker	SG	Phoenix Suns	24	6' 5"	6.50	206	Kentucky	31650600.0	25.6	4.2	4.3
357	De'Aaron Fox	PG	Sacremento Kings	23	6' 3"	6.30	185	Kentucky	28103550.0	25.2	3.5	7.2
377	Willie Cauley- Stein	С	Dallas Mavericks	28	7' 0"	7.00	240	Kentucky	4100000.0	5.3	4.5	0.7
408	John Wall	PG	Houston Rockets	31	6' 3"	6.30	210	Kentucky	44310840.0	20.6	3.2	6.9
434	Wenyen Gabriel	PF	New Orleans Pelicans	24	6' 9"	6.90	205	Kentucky	1762796.0	3.4	2.6	0.5
456	Keldon Johnson	SF	San Antonio Spurs	21	6' 5"	6.50	220	Kentucky	2145720.0	12.8	6.0	1.8
482	Jamal Murray	PG	Denver Nuggets	24	6' 3"	6.30	215	Kentucky	29467800.0	21.2	4.0	4.8
499	Karl-Anthony Towns	С	Minnesota Timberwolves	25	6' 11"	6.11	248	Kentucky	31650600.0	24.8	10.6	4.5
500	Jarred Vanderbilt	PF	Minnesota Timberwolves	22	6' 9"	6.90	214	Kentucky	NaN	5.4	5.8	1.2
508	Shai Gilgeous- Alexander	SG	Oklahoma City Thunder	23	6' 6"	6.60	180	Kentucky	5495532.0	23.7	4.7	5.9

2.1.3 Opération classique : Moyenne, maximum et minimum

→ On peut calculer la moyenne des salaires :

```
df["Salary"].mean() # ou df.Salary.mean()

8813695.5752809
```

→ le maximum

```
df["Salary"].max()
45780966.0
```

→ le minimum

```
df["Salary"].min()
925258.0
```

2.1.4 Filtrer les lignes avec des indices en utilisant iloc

ightharpoonup On peut également faire afficher le salaire max et le nom du joueur correspondant :

```
result = df[df["Salary"].max() == df["Salary"]]["Name"].iloc[0]
print(result)
Stephen Curry
```

A vous

Faites de même pour faire apparaître le nom du joueur ayant le salaire minimum.

A vous

Faire apparaitre le poste de Kemba Walker

```
result = df[df["Name"] == "Kemba Walker"]["Position"].iloc[0]
print(result)
```

A vous

Faire apparaitre le plus grand joueur

```
result = df[df["Height"].max() == df["Height"]]["Name"].iloc[0]
print(result)
```

A vous

Faire apparaitre le plus petit joueur

```
result = df[df["Height"].min() == df["Height"]]["Name"].iloc[0]
print(result)

df.loc[df['Name']=='Jared Harper']
```

A vous

Faire apparaitre le meilleur rebondeur

A vous

Donner le nom du meilleur marqueur ainsi que sa moyenne de points

```
pointsmax=df["Points"].max()
result = df[df["Points"].max() == df["Points"]]["Name"].iloc[0]
print(result,pointsmax)

result = df[df["Points"].max() == df["Points"]][["Name","Points"]].iloc[0]
print(result)
```

A vous

Donner les joueurs des Brooklyn Nets

3. Création de graphique

```
%matplotlib inline
import matplotlib.pyplot as plt
X = df['Weight']
Y = df['Height_i']
```

```
plt.plot(X,Y,'ro') # r pour red, o pour un cercle. voir https://matplotlib.org/api/markers_api.html
plt.show()
```

L'interprétation numérique permet à pandas d'analyser automatiquement les données, avec notamment la fonction describe().

```
df['Height_i'].describe()

df.boxplot('Height_i')

df[['Name', 'Team']]
```

	Name	Team
0	Juhann Begarin	Boston Celtics
1	Jaylen Brown	Boston Celtics
2	Kris Dunn	Boston Celtics
3	Carsen Edwards	Boston Celtics
4	Tacko Fall	Boston Celtics
		•••
553	Juwan Morgan	Utah Jazz
554	Royce O'Neale	Utah Jazz
555	Olumiye Oni	Utah Jazz
556	Eric Paschall	Utah Jazz
557	Hassan Whiteside	Utah Jazz

558 rows × 2 columns

4. Rajout d'une colonne

Afin de pouvoir trier les joueurs suivant de nouveaux critères, nous allons rajouter un champ pour chaque joueur. Prenons un exemple stupide : fabriquons un nouveau champ 'Taille en cm' qui contiendra la taille des joueurs aprés conversion en cm. Ceci se fera simplement par :

```
df['Taille'] = round(df['Height_i'] * 0.3048,2)
```

df

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists	Taille
0	Juhann Begarin		Boston Celtics	19	6' 5"	6.50	185	NaN	NaN	NaN	NaN	NaN	1.98
1	Jaylen Brown		Boston Celtics	24	6' 6"	6.60	223	California	26758928.0	24.7	6.0	3.4	2.01
2	Kris Dunn		Boston Celtics	27	6' 3"	6.30	205	Providence	5005350.0	1.3	1.5	0.5	1.92
3	Carsen Edwards	_	Boston Celtics	23	5' 11"	5.11	200	Purdue	1782621.0	4.0	0.8	0.5	1.56
4	Tacko Fall		Boston Celtics	25	7' 5"	7.50	311	UCF	NaN	2.5	2.7	0.2	2.29
553	Juwan Morgan	SF	Utah Jazz	24	6' 7"	6.70	232	Indiana	NaN	1.2	1.0	0.3	2.04

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists	Taille
554	Royce	PF	Utah Jazz	28	6' 4"	6.40	226	Baylor	0.0000088	7.0	6.8	2.5	1.95
334	O'Neale												
555	Olumiye Oni	SG	Utah Jazz	24	6' 5"	6.50	206	Yale	1782621.0	NaN	NaN	NaN	1.98
556	Eric Paschall	F	Utah Jazz	24	6' 6"	6.60	255	Villanova	1782621.0	9.5	3.2	1.3	2.01
557	Hassan	С	Utah Jazz	32	7' 0"	7.00	265	Marshall	1669178.0	8.1	6.0	0.6	2.13
557	Whiteside												

558 rows × 13 columns

A vous

Faite de même pour convertir le poid en kg.

```
df['Poids'] = round(df['Weight'] * 0.45359 ,0)
```

df

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists	Taille	Poids
0	Juhann Begarin	SG	Boston Celtics	19	6' 5"	6.50	185	NaN	NaN	NaN	NaN	NaN	1.98	84.0
1	Jaylen Brown	SG	Boston Celtics	24	6' 6"	6.60	223	California	26758928.0	24.7	6.0	3.4	2.01	101.0
2	Kris Dunn	PG	Boston Celtics	27	6' 3"	6.30	205	Providence	5005350.0	1.3	1.5	0.5	1.92	93.0
3	Carsen Edwards	PG	Boston Celtics	23	5' 11"	5.11	200	Purdue	1782621.0	4.0	0.8	0.5	1.56	91.0
4	Tacko Fall	С	Boston Celtics	25	7' 5"	7.50	311	UCF	NaN	2.5	2.7	0.2	2.29	141.0
553	Juwan Morgan	SF	Utah Jazz	24	6' 7"	6.70	232	Indiana	NaN	1.2	1.0	0.3	2.04	105.0
554	Royce O'Neale	PF	Utah Jazz	28	6' 4"	6.40	226	Baylor	8800000.0	7.0	6.8	2.5	1.95	103.0
555	Olumiye Oni	SG	Utah Jazz	24	6' 5"	6.50	206	Yale	1782621.0	NaN	NaN	NaN	1.98	93.0
556	Eric Paschall	F	Utah Jazz	24	6' 6"	6.60	255	Villanova	1782621.0	9.5	3.2	1.3	2.01	116.0
557	Hassan Whiteside	С	Utah Jazz	32	7' 0"	7.00	265	Marshall	1669178.0	8.1	6.0	0.6	2.13	120.0

558 rows × 14 columns

A vous

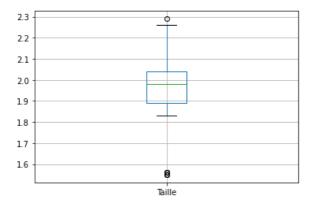
Faire afficher la répartition des tailles (Voir plus haut)

```
df['Taille'].describe()

count 558.000000
mean 1.977419
std 0.101202
min 1.550000
25% 1.890000
50% 1.980000
75% 2.040000
max 2.290000
Name: Taille, dtype: float64
```

df.boxplot('Taille')

<AxesSubplot:>



→ Trier les joueurs par ordre décroissant de poids

```
p=df.sort_values(by=['Poids'], ascending = False)
p
```

A vous

Faire de même pour la taille, les scoreurs et rebondeurs

```
scoreur=df.sort_values(by=['Points'], ascending = False)
scoreur
```

A vous

- 1. Créer une colonne contenant l'IMC de chaque joueur
- $2.\ Cr\'{e}er\ une\ nouvelle\ data frame\ contenant\ tous\ les\ joueurs\ du\ top 14\ class\'{e}s\ par\ ordre\ d'IMC\ croissant.$

A vous

Créer une dataframe contenant les joueurs en provenance de Duke

```
dfduke=df[df['College']=='Duke']
```

dfduke

A vous

Déterminer les joueurs gagnat plus de 15 millions de dollars et marquant moins de 10 points par match (le ET : &)

```
df.loc[(df.Points<10) & (df.Salary>=15000000)]
```

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists	Taille	Poids
12	Ricky	G	Cleveland	30	6' 3"	6.30	190	NaN	17800000.0	8.6	3.3	6.4	1.92	86.0
12	Rubio		Cavaliers											
25	Gary	SG	Orlando	26	6' 4"	6.40	210	Michigan	20482143.0	9.9	2.0	2.0	1.95	95.0
23	Harris		Magic					State						
27	Spencer	PG	Washington	28	6' 5"	6.50	215	Colorado	17142857.0	6.7	4.3	3.0	1.98	98.0
21	Dinwiddie		Wizards											

	Name	Position	Team	Age	Height	Height_i	Weight	College	Salary	Points	Rebounds	Assists	Taille	Poids
	Draymond Green		Golden State Warriors	31	6' 6"	6.60		Michigan State	24026712.0	7.0	7.1	8.9	2.01	104.0
411	Steven Adams		Memphis Grizzlies	28	6' 11"	6.11	265	Pittsburgh	17073171.0	7.6	8.9	1.9	1.86	120.0

A vous

Créer une dictionnaire avec les divers colléges comme clé et le nombre de joueurs comme valeur.

```
dico={}
for k in df.College:
    if k not in dico:
        if k !="nan":
             dico[k]=1
    else:
        dico[k]+=1
print(dico)
```

{nan: 73, 'California': 1, 'Providence': 1, 'Purdue': 2, 'UCF': 1, 'Maryland': 6, 'Florida': 7, 'Kentucky': 30, 'Vanderbilt': 5, 'Indiana': 8, 'Duke': 27, 'Oregon': 8, 'Tennessee': 6, 'Oklahoma State': 2, 'Texas A&M': 4, 'Texas': 14, "Saint Joseph's": 4, 'Miami (FL)': 2, 'West Virginia': 2, 'Georgia': 3, 'Pepperdine': 1, 'Florida State': 11, 'Oklahoma': 3, 'Arizona State': 3, 'Virginia': 8, 'Eastern Arizona Coll. (J.C.)': 1, 'Missouri State': 1, 'Wake Forest': 7, 'Fresno State': 2, "Saint Mary's": 2, 'Louisiana Tech': 1, 'Mississippi State': 3, 'North Carolina': 13, 'LSU': 6, 'Creighton': 3, 'Colorado': 3, 'USC': 10, 'Houston': 4, 'Auburn': 5, "St. John's": 3, 'Memphis': 4, 'Dayton': 2, 'Connecticut': 5, 'Western Kentucky': 1, 'Cincinnati': 1, 'Kansas': 12, 'Michigan State': 8, 'Arkansas': 5, 'SMU': 3, 'Iowa State': 5, 'Gonzaga': 9, 'DePaul': 2, 'Charleston': 2, 'Washington': 8, 'Nebraska': 2, 'UNLV': 3, 'New Mexico JC': 1, 'Wisconsin': 2, 'San Diego State': 3, 'Baylor': 5, 'Louisville': 6, 'Arizona': 11, 'New Mexico State': 1, 'Wichita State': 2, 'George Washington': 1, 'Villanova': 9, 'UCLA': 13, 'Illinois': 2, 'Radford': 1, 'Alabama': 5, 'Penn State': 2, 'Truman State': 1, 'Kansas State': 3, 'Belmont': 1, 'Iowa': 2, 'Syracuse': 5, 'Michigan': 11, 'North Carolina State': 2, 'USC Upstate': 1, 'Butler': 3, 'Xavier': 2, 'BYU': 1, 'Notre Dame': 1, 'IUPUI': 1, 'Stanford': 7, 'Seton Hall': 1, 'Florida Gulf Coast': 1, 'Utah': 3, 'Delaware': 1, 'Nevada': 3, 'Pittsburgh': 2, 'Marquette': 5, 'TCU': 3, 'UC Santa Barbara': 3, 'Georgetown': 3, 'Rhode Island': 1, 'Long Beach State': 1, 'South Carolina': 3, 'Liberty': 1, 'David Lipscomb University': 1, 'Davidson': 1, 'Drexel': 1, 'Salt Lake CC UT': 1, 'Washington State': 2, 'Minnesota': 2, 'Boston College': 1, 'Ohio': 1, 'John A. Logan College': 1, 'Old Dominion': 1, 'North Carolina-Wilmington': 1, 'Boise State': 1, 'Murray State': 2, 'Louisiana-Lafayette': 1, 'Ole Miss': 1, 'Bowling Green': 1, 'Wyoming': 2, 'Utah State': 2, 'Texas Tech': 2, 'Vermont': 1, 'Cal Poly': 1, 'Ohio State': 4, 'Loyola (MD)': 1, 'Missouri': 3, 'Virginia Tech': 1, 'Oregon State': 1, 'Cleveland State': 1, 'Western Texas Coll. (J.C.)': 1, 'Georgia Tech': 3, 'Tulsa': 1, 'Virginia Commonwealth': 1, 'William & Mary': 1, 'Moravian': 1, 'Bucknell': 1, 'Montana State': 1, 'Tennessee State': 1, 'Weber State': 1, 'Lehigh': 1, 'New Mexico': 1, 'Yale': 1, 'Marshall': 1}

A vous