Introduction à Python II: Pandas

Chuan Xu

chuan.xu@univ-cotedazur.fr

Sept. 2024

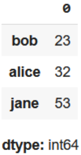
Bibliothèque pandas

- Analyses de données
- opérations d'exploration, de nettoyage et de transformation de données qui sont essentielles pour travailler avec les données en Python.
- Les principales structures de données fournies par pandas sont les **Series** et les **DataFrames**

```
import pandas as pd
```

Series

- Tableau étiqueté unidimensionnel



```
        0
bob    23
alice  32
jane   53

dtype: int64
```

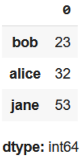
```
ser = pd.Series(data=[23,32,53], index=["bob", "alice", "jane"])
ser = pd.Series(data={"bob":23,"alice":32,"jane":53})

ser.index #type Index de pandas
ser["bob"] #prendre la valeur de "bob"
"bob" in ser # Vérifier si "bob" est dans index
ser[1] #deuxième élément dans data
ser[[0]] # un nouveau série qui contient "bob" et 23.
ser[[2,0]] # un nouveau série qui contient "jane" "bob" et 53, 23
ser*2 # multiplier les data dans la série par 2

ser.value_counts() #La fréquence de chaque valeur
```

Series

- Tableau étiqueté unidimensionnel



	0
bob	23
alice	32
jane	53

dtype: int64

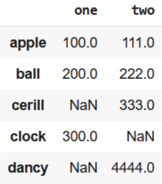
```
ser = pd.Series(data=[23,32,53], index=["bob", "alice", "jane"])
ser = pd.Series(data={"bob":23,"alice":32,"jane":53})

ser.where(ser>20, 40, inplace=True)
#Remplacer les valeurs par 40 quand la condition est False
ser.where(ser>20, inplace=True)
#Remplacer les valeurs par NaN quand la condition est False

ser[ser>20] #Une nouvelle série avec les valeurs satisfaient la condition
```

DataFrame

- Tableau étiqueté deux-dimensionnel



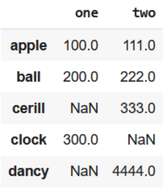
	one	two
apple	100.0	111.0
ball	200.0	222.0
cerill	NaN	333.0
clock	300.0	NaN
dancy	NaN	4444.0

```
d = {'one' : pd.Series([100., 200., 300.], index=['apple', 'ball', 'clock']),
'two' : pd.Series([111, 222, 333, 4444], index=['apple', 'ball', 'cerill', 'dancy'
# d est une dictionnaire où les clés sont les noms pour les colonnes
df = pd.DataFrame(d) # Créer DataFrame à partir du dictionnaire

df_sub = pd.DataFrame(d, index=['dancy', 'ball'])
# Créer un DataFrame qui contient que les lignes de "dancy" et "ball".

df_sub_c = pd.DataFrame(d, index=['dancy', 'ball'], columns=["two"])
# Créer un DataFrame qui contient que les colonnes de "two"
```

DataFrame

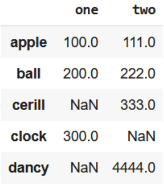


	one	two
apple	100.0	111.0
ball	200.0	222.0
cerill	NaN	333.0
clock	300.0	NaN
dancy	NaN	4444.0

```
# Création de DataFrame par une liste de dictionnaire
l = [{"one":100, "two": 111}, {"one":200, "two":222}, {"two":333},
 {"one":300}, {"two": 4444}]

df = pd.DataFrame(l, index=["apple", "ball", "cerill", "clock", "dancy"])
```

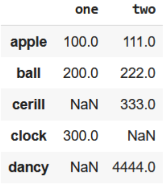
DataFrame : opération basique I



	one	two
apple	100.0	111.0
ball	200.0	222.0
cerill	NaN	333.0
clock	300.0	NaN
dancy	NaN	4444.0

```
df.shape # La taille de dataframe (tuple)
df["one"] # Choisir la colonne et retourne une série
df.iloc[0] # Choisir la première ligne
df.loc["apple"]
df["three"] = df["one"]*df["two"] # Ajouter une colonne "three"
df.insert(0, "zero", df["one"]*2)
# Insérez une colonne avant la première colonne, nommée 'zero',
# avec des valeurs deux fois plus grandes que celles de la colonne 'one'.
df['flag'] = df['one']>250
# Ajouter une colonne de "flag" avec True et False
```

DataFrame : opération basique II



	one	two
apple	100.0	111.0
ball	200.0	222.0
cerill	NaN	333.0
clock	300.0	NaN
dancy	NaN	4444.0

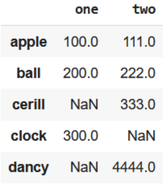
```
df = df.dropna()
# Enlever les lignes qui contienent NaN et retourne df
df = df.dropna(thresh=2)
# Enlever les lignes qui contienent au moins deux NaN

df["one"].fillna(value=df["one"].mean(), inplace=True)
# Remplacer les valeurs NaN par le moyenne

df["one"].fillna(value=df["one"].median(), inplace=True)
# Remplacer les valeurs NaN par la médiane

df["one"].fillna(value=df["one"].mode()[0], inplace=True)
# Remplacer les valeurs NaN par le valeur plus fréquent
```

DataFrame : opération basique III

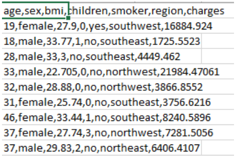


	one	two
apple	100.0	111.0
ball	200.0	222.0
cerill	NaN	333.0
clock	300.0	NaN
dancy	NaN	4444.0

```
del df["two"] #supprime la colonne "two"
df.drop(columns=["two"], inplace=True)
df.drop(["apple", "bail"], inplace=True) #supprime les lignes


df = df[df["two"]<200] # Garder que les ligne dont les valeurs
# pour la colonne "two" est plus petit que 200
```

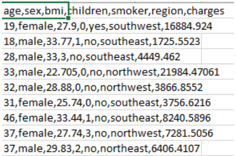
Lire les données



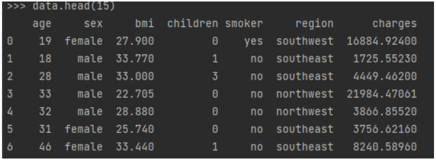
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
```

Lire les données



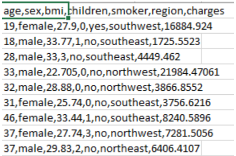
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



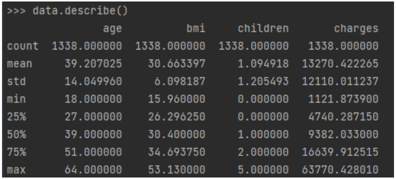
```
>>> data.head(15)
   age     sex     bmi  children smoker     region      charges
0   19  female  27.900         0    yes  southwest  16884.92400
1   18    male  33.770         1     no  southeast   1725.55230
2   28    male  33.000         3     no  southeast   4449.46200
3   33    male  22.705         0     no  northwest  21984.47061
4   32    male  28.880         0     no  northwest   3866.85520
5   31  female  25.740         0     no  southeast   3756.62160
6   46  female  33.440         1     no  southeast   8240.58960
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
```

Lire les données



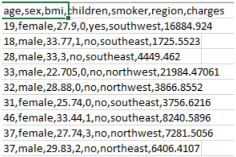
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



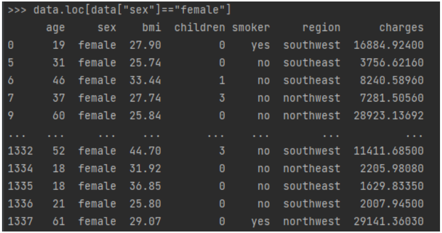
```
>>> data.describe()
               age          bmi     children       charges
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.663397     1.094918  13270.422265
std      14.049960     6.098187     1.205493  12110.011237
min      18.000000    15.960000     0.000000   1121.873900
25%      27.000000    26.296250     0.000000   4740.287150
50%      39.000000    30.400000     1.000000   9382.033000
75%      51.000000    34.693750     2.000000  16639.912515
max      64.000000    53.130000     5.000000  63770.428010
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
data.describe()
data.describe().loc["mean"]["age"]
```

Lire les données



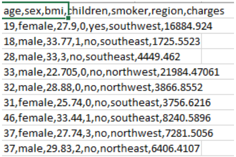
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



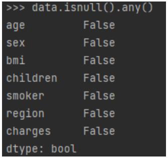
```
>>> data.loc[data["sex"]=="female"]
      age     sex    bmi  children smoker     region      charges
0      19  female  27.90         0    yes  southwest  16884.92400
5      31  female  25.74         0     no  southeast   3756.62160
6      46  female  33.44         1     no  southeast   8240.58960
7      37  female  27.74         3     no  northwest   7281.50560
9      60  female  25.84         0     no  northwest  28923.13692
...   ...     ...    ...       ...    ...        ...          ...
1332   52  female  44.70         3     no  southwest  11411.68500
1334   18  female  31.92         0     no  northeast   2205.98080
1335   18  female  36.85         0     no  southeast   1629.83350
1336   21  female  25.80         0     no  southwest   2007.94500
1337   61  female  29.07         0    yes  northwest  29141.36030
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
data.describe()
data.describe().loc["mean"]["age"]
data.loc[data["sex"]=="female"]
```

Lire les données



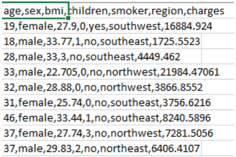
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



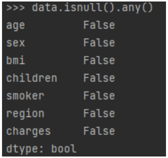
```
>>> data.isnull().any()
age         False
sex         False
bmi         False
children    False
smoker      False
region      False
charges     False
dtype: bool
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
data.describe()
data.describe().loc["mean"]["age"]
data.loc[data["sex"]=="female"]
data.isnull().any()
```

Lire les données



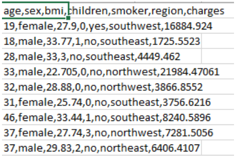
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



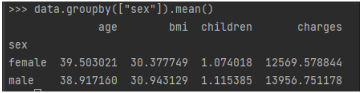
```
>>> data.isnull().any()
age         False
sex         False
bmi         False
children    False
smoker      False
region      False
charges     False
dtype: bool
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
data.describe()
data.describe().loc["mean"]["age"]
data.loc[data["sex"]=="female"]
data.isnull().any()
```

Lire les données



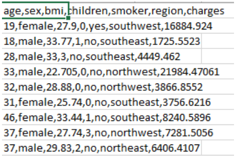
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



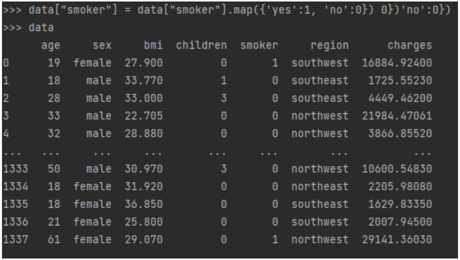
```
>>> data.groupby(["sex"]).mean()
              age        bmi  children       charges
sex
female  39.503021  30.377749  1.074018  12569.578844
male    38.917160  30.943129  1.115385  13956.751178
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
data.describe()
data.describe().loc["mean"]["age"]
data.loc[data["sex"]=="female"]
data.isnull().any()
data.groupby(["sex"]).mean()
```

Prétraitement des données



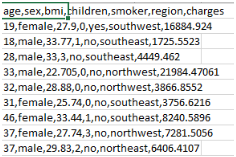
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



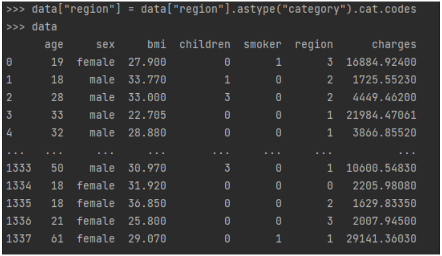
```
>>> data["smoker"] = data["smoker"].map({'yes':1, 'no':0}) 0})'no':0})
>>> data
      age     sex     bmi  children  smoker     region      charges
0      19  female  27.900         0       1  southwest  16884.92400
1      18    male  33.770         1       0  southeast   1725.55230
2      28    male  33.000         3       0  southeast   4449.46200
3      33    male  22.705         0       0  northwest  21984.47061
4      32    male  28.880         0       0  northwest   3866.85520
...   ...     ...     ...       ...     ...        ...          ...
1333   50    male  30.970         3       0  northwest  10600.54830
1334   18  female  31.920         0       0  northeast   2205.98080
1335   18  female  36.850         0       0  southeast   1629.83350
1336   21  female  25.800         0       0  southwest   2007.94500
1337   61  female  29.070         0       1  northwest  29141.36030
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
data.describe()
data.describe().loc["mean"]["age"]
data.loc[data["sex"]=="female"]
data.isnull().any()
data.groupby(["sex"]).mean()
data["smoker"] = data["smoker"].map({'yes':1, 'no':0})
```

Prétraitement des données



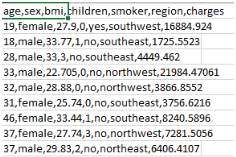
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



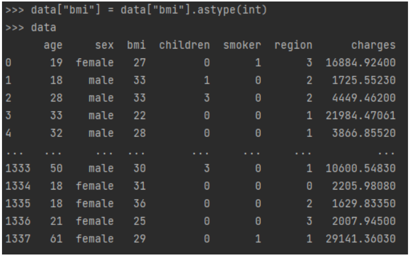
```
>>> data["region"] = data["region"].astype("category").cat.codes
>>> data
      age     sex     bmi  children  smoker  region      charges
0      19  female  27.900         0       1       3  16884.92400
1      18    male  33.770         1       0       2   1725.55230
2      28    male  33.000         3       0       2   4449.46200
3      33    male  22.705         0       0       1  21984.47061
4      32    male  28.880         0       0       1   3866.85520
...   ...     ...     ...       ...     ...     ...          ...
1333   50    male  30.970         3       0       1  10600.54830
1334   18  female  31.920         0       0       0   2205.98080
1335   18  female  36.850         0       0       2   1629.83350
1336   21  female  25.800         0       0       3   2007.94500
1337   61  female  29.070         0       1       1  29141.36030
```

```
data = pd.read_csv('insurance.csv', sep=',')
data.head(15)
data.describe()
data.describe().loc["mean"]["age"]
data.loc[data["sex"]=="female"]
data.isnull().any()
data.groupby(["sex"]).mean()
data["smoker"] = data["smoker"].map({'yes':1, 'no':0})
data["region"] = data["region"].astype("category").cat.codes
```

Prétraitement des données



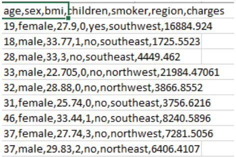
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



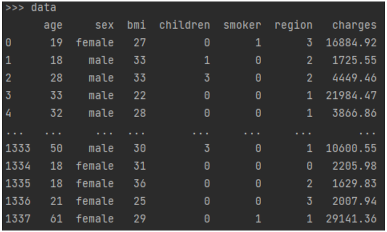
```
>>> data["bmi"] = data["bmi"].astype(int)
>>> data
      age     sex  bmi  children  smoker  region      charges
0      19  female   27         0       1       3  16884.92400
1      18    male   33         1       0       2   1725.55230
2      28    male   33         3       0       2   4449.46200
3      33    male   22         0       0       1  21984.47061
4      32    male   28         0       0       1   3866.85520
...   ...     ...  ...       ...     ...     ...          ...
1333   50    male   30         3       0       1  10600.54830
1334   18  female   31         0       0       0   2205.98080
1335   18  female   36         0       0       2   1629.83350
1336   21  female   25         0       0       3   2007.94500
1337   61  female   29         0       1       1  29141.36030
```

```python
data["smoker"] = data["smoker"].map({'yes':1, 'no':0})
data["region"] = data["region"].astype("category").cat.codes
data["bmi"] = data["bmi"].astype(int)
```

Prétraitement des données



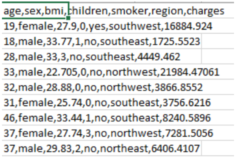
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



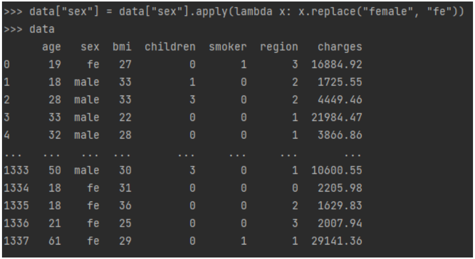
```
>>> data
      age     sex  bmi  children  smoker  region   charges
0      19  female   27         0       1       3  16884.92
1      18    male   33         1       0       2   1725.55
2      28    male   33         3       0       2   4449.46
3      33    male   22         0       0       1  21984.47
4      32    male   28         0       0       1   3866.86
...   ...     ...  ...       ...     ...     ...       ...
1333   50    male   30         3       0       1  10600.55
1334   18  female   31         0       0       0   2205.98
1335   18  female   36         0       0       2   1629.83
1336   21  female   25         0       0       3   2007.94
1337   61  female   29         0       1       1  29141.36
```

```
data["smoker"] = data["smoker"].map({'yes':1, 'no':0})
data["region"] = data["region"].astype("category").cat.codes
data["bmi"] = data["bmi"].astype(int)
data["charges"] = data["charges"].apply(lambda x: round(x,2))
```

Prétraitement des données



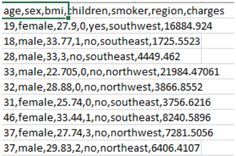
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



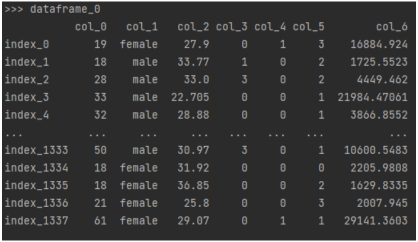
```
>>> data["sex"] = data["sex"].apply(lambda x: x.replace("female", "fe"))
>>> data
      age   sex  bmi  children  smoker  region   charges
0      19    fe   27         0       1       3  16884.92
1      18  male   33         1       0       2   1725.55
2      28  male   33         3       0       2   4449.46
3      33  male   22         0       0       1  21984.47
4      32  male   28         0       0       1   3866.86
...   ...   ...  ...       ...     ...     ...       ...
1333   50  male   30         3       0       1  10600.55
1334   18    fe   31         0       0       0   2205.98
1335   18    fe   36         0       0       2   1629.83
1336   21    fe   25         0       0       3   2007.94
1337   61    fe   29         0       1       1  29141.36
```

```
data["smoker"] = data["smoker"].map({'yes':1, 'no':0})
data["region"] = data["region"].astype("category").cat.codes
data["bmi"] = data["bmi"].astype(int)
data["charges"] = data["charges"].apply(lambda x: round(x,2))
data["sex"] = data["sex"].apply(lambda x: x.replace("female", "fe"))
```

Entre DataFrame et Numpy array



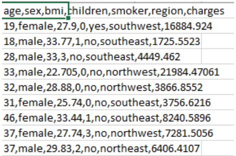
```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



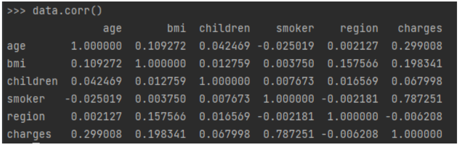
```
>>> dataframe_0
           col_0   col_1   col_2 col_3 col_4 col_5        col_6
index_0       19  female    27.9     0     1     3    16884.924
index_1       18    male   33.77     1     0     2    1725.5523
index_2       28    male    33.0     3     0     2     4449.462
index_3       33    male  22.705     0     0     1  21984.47061
index_4       32    male   28.88     0     0     1    3866.8552
...          ...     ...     ...   ...   ...   ...          ...
index_1333    50    male   30.97     3     0     1   10600.5483
index_1334    18  female   31.92     0     0     0    2205.9808
index_1335    18  female   36.85     0     0     2    1629.8335
index_1336    21  female    25.8     0     0     3     2007.945
index_1337    61  female   29.07     0     1     1   29141.3603
```

```python
numpyarray = data.values #Retourne un tableau numpy
columns = [f'col_{num}' for num in range(numpyarray.shape[1])]
index = [f'index_{num}' for num in range(numpyarray.shape[0])]
dataframe_0 = pd.DataFrame(numpyarray, columns=columns, index=index)
```

Statistics



```
age,sex,bmi,children,smoker,region,charges
19,female,27.9,0,yes,southwest,16884.924
18,male,33.77,1,no,southeast,1725.5523
28,male,33,3,no,southeast,4449.462
33,male,22.705,0,no,northwest,21984.47061
32,male,28.88,0,no,northwest,3866.8552
31,female,25.74,0,no,southeast,3756.6216
46,female,33.44,1,no,southeast,8240.5896
37,female,27.74,3,no,northwest,7281.5056
37,male,29.83,2,no,northeast,6406.4107
```



```
>>> data.corr()
               age       bmi  children    smoker    region   charges
age       1.000000  0.109272  0.042469 -0.025019  0.002127  0.299008
bmi       0.109272  1.000000  0.012759  0.003750  0.157566  0.198341
children  0.042469  0.012759  1.000000  0.007673  0.016569  0.067998
smoker   -0.025019  0.003750  0.007673  1.000000 -0.002181  0.787251
region    0.002127  0.157566  0.016569 -0.002181  1.000000 -0.006208
charges   0.299008  0.198341  0.067998  0.787251 -0.006208  1.000000
```

Pearson correlation coefficient:

$$r_{xy} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}$$

- reflétant une relation linéaire entre deux variables continues
- valeur négative (corrélation négative) signifiant que lorsqu'une des variable augmente, l'autre diminue

---

```
data.corr() # afficher les corrélations entre les deux attributes
```

TP pandas+numpy: à rendre avant le 12 Octobre sur Moodle.

`https://gitlab.inria.fr/chxu/python-pour-ia-2024/-/tree/main/Python_pandas_numpy?ref_type=heads`