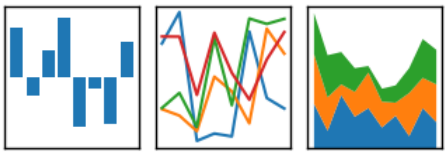
Mémento Pandas

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



Lecture d'un fichier csv	
Importer le module pandas	`import pandas`
Lire le fichier *nom.csv* figurant dans le même dossier que le programme Python et affecter les données à une table de données, nommée T, dont chaque colonne correspond à une variable statistique.	`T = pandas.read_csv('nom.csv', sep = ';')` `sep = ','` si le séparateur du fichier csv est une virgule. `sep = '\t'` si le séparateur du fichier csv est une tabulation. Ajouter l'argument `encoding = 'latin-1'` si le fichier csv comporte des accents. Ajouter l'argument `decimal = ','` si les nombres décimaux du fichier csv sont séparés par une virgule.
Opérations de base sur une table pandas et ses colonnes	
Dimension de T : nombre de lignes (individus) et de colonnes (variables).	`T.shape`
(Nom) et type des variables (ou colonnes).	`T.dtypes`
Afficher les 5 premières lignes de T.	`T.head(5)`
Afficher les 5 dernières lignes de T.	`T.tail(5)`
Afficher des indicateurs statistiques pour chaque colonne de nombres de T.	`T.describe()`
Afficher des indicateurs statistiques pour la colonne de nombres nommée C.	`T['C'].describe()`
Calculer la moyenne de la variable C.	`T['C'].mean()`
Calculer l'écart type de la variable C. Remarque : la fonction std() fournit, par défaut, l'estimation s_{n-1} de l'écart type en considérant C comme un échantillon.	`T['C'].std(ddof = 0)`
Calculer la médiane de la variable C.	`T['C'].median()`
Calculer la somme de la variable C.	`T['C'].sum()`
Compter le nombre de valeurs de la colonne C.	`T['C'].count()`
Déterminer les effectifs de chacune des valeurs prises par la variable C.	`T['C'].value_counts()`
Trier T par ordre croissant selon la variable C.	`T.sort_values(by = 'C ')`
Trier T par ordre décroissant selon la variable C.	`T.sort_values(by = 'C ', ascending = False)`

Extraction d'un sous-tableau sous condition, tableau croisé avec pandas	

Créer une table T1 correspondant aux lignes de T où la variable C est non nulle.	`T1 = T.query('C != 0')`
	ou
	`T1 = T[T['C'] != 0]`

Créer une table T2 correspondant aux lignes de T où la variable A est supérieure à 2 **et** la variable B inférieure à 6.	`T2 = T.query('A >= 2 and B <= 6')`
	ou
	`T2 = T[(T['A'] >= 2) & (T['B'] <= 6)]`

Créer une table T3 correspondant aux lignes de T où la variable A est supérieure à 2 **ou** la variable B inférieure à 6.	`T3 = T.query('A >= 2 or B <= 6')`
	ou
	`T3 = T[(T['A'] >= 2) | (T['B'] <= 6)]`

Créer une table T4 correspondant aux lignes de T où la variable A contient la chaîne de caractères "kiwi".	`T4 = T.query('A == "kiwi"')`
	ou
	`T4 = T[(T['A'] == "kiwi"]`

Créer une table T5 correspondant aux lignes de T où la colonne A contient une variable nommée var.	`T5 = T.query('A == @var')`
	ou
	`T5 = T[(T['A'] == var]`

Créer une table T6 échantillon aléatoire sans remise de 100 lignes de T.	`import random` `T6 = T.sample(n = 100)`
Créer une table des effectifs des colonnes de T en regroupant les données selon les valeurs de la variable C.	`T.groupby('C').count()`
Créer une table des moyennes des colonnes de T en regroupant les données selon les valeurs de la variable C.	`T.groupby('C').mean()`
Créer une table des sommes des colonnes de T en regroupant les données selon les valeurs de la variable C.	`T.groupby('C').sum()`
Créer un tableau croisé en effectifs des variables C1 et C2 de la table T.	`pandas.crosstab(T['C1'], T['C2'], margins = True)`