2- Python – Conversion et Manipulation de dataframe

Pour plus d'informations …
www.icuresearch.eu

Dataframe : Tableau à deux dimensions composé de lignes et de colonnes. Il peut être comparé aux dictionnaires de Python : les clés sont les noms des colonnes et les valeurs sont les séries.

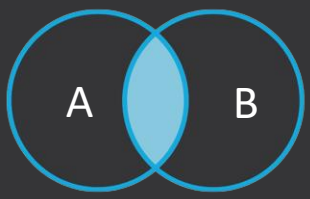
Librairies incontournables (à ajouter au début de son script Python) :
- Pandas (Outil d'analyse, de manipulation de données open source rapide, puissant, flexible et facile à utiliser). => *import pandas as pd*
- Numpy (Outil permettant d'effectuer des calculs scientifiques) => *import numpy as np*

Objectif	Remarque	Code Python
Copier une table *Remarque : En Python l'opérateur d'assignation créé aussi un pointeur vers la variable initiale*	*table_cpy = table → Si on modifie les valeurs de "table", les valeurs de "table_cpy" seront également changées.*	table_cpy = table.copy()
Regarder les premières lignes d'un dataframe	*Observation des 5 premières lignes du dataframe "table".*	table.head(5)
Créer une variable	*Dans le dataframe "table" :* *Total = var1 + var2*	table["total"] = table["var1"] + table["var2"]
Filtrer un dataframe	*Condition du filtre sur le dataframe "table" :* *Total > 10 ET sexe = "Femme"*	table[(table["total"] > 10) & (table["sex"]=="Female")]
SI … ALORS … SINON …	*Dans le dataframe "table" :* *Si total > 10 alors la classe du total est "High".* *Sinon on attrbue la valeur "Low" à la classe.*	table["class_tot"] = np.where(table["class_tot"]>10, "High", "Low")
Trier par valeur	*Tri des valeurs en fonction des variables "total" et "sex" du dataframe "table".* *Remarque : Par default tri ascendant.*	table.sort_values(["total", "sex"])
Sélectionner les données manquantes Ou non manquantes	*Sélection des données égales "NA".* *Sélection des données non égales à "NA".*	table[table["sex"].isna()] table[table["sex"].notna()]
Group by	*Compte des données par sexe dans le dataframe "table".* *Remarque : Peut-être utilisé avec .max, .min, etc.*	table.groupby("sex").count()
Transformation	*Récupération des valeurs moyennes de la variable "total" par sexe.*	moyenne = table.groupby("sex")["total"] table["transf_total"] = table["total"] – moyenne.transform("mean")
Processing by group	*Renvoi la première ligne par groupe "sex" et "smoker".*	table.groupby(["sex", "smoker"]).first()
Sélectionner Supprimer Renommer	*Sélection des variables "sex", "total" et "total_class".* *Suppression de la variable "sex".* *Attribution du nom "total_final" à la variable "total".*	table[["sex", "total", "total_class"]] table.drop("sex", axis] table.rename(columns={"total": "total_final"})

Quentin
Pour plus d'informations ...
www.icuresearch.eu

Le Merge

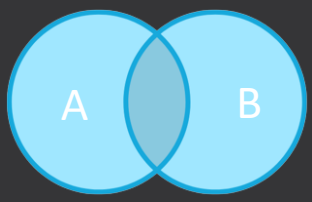
INNER JOIN



Joint les individus communs à A et B.

A.merge(B, on=["key"], how="inner")

FULL OUTER JOIN

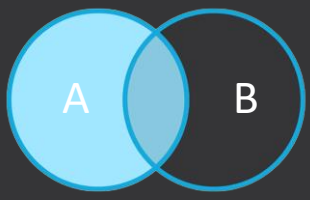


Joint tous les individus contenus dans A et B.

Remarque : La valeur « null » est attribuée quand l'enregistrement d'une table ne correspond pas à un enregistrement de l'autre.

A.merge(B, on=["key"], how="outer")

LEFT JOIN

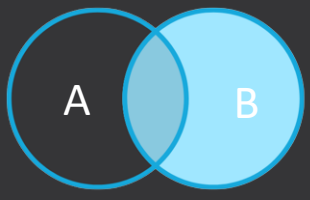


Joint les individus contenus dans A avec ceux ayant une correspondance dans B.

Remarque : La valeur « null » est attribuée pour les enregistrements de A qui n'ont pas de valeurs correspondantes dans B.

A.merge(B, on=["key"], how="left")

RIGHT JOIN



Joint les individus contenus dans B avec ceux ayant une correspondance dans A.

Remarque : La valeur « null » est attribuée pour les enregistrements de B qui n'ont pas de valeurs correspondantes dans A.

A.merge(B, on=["key"], how="right")

Les Formats

1

Créer un dictionnaire contenant en clé la valeur initiale et en valeur le format à appliquer.

dico_format = {0: "Non", 1:"Oui"}

2

Utiliser la fonction .map() pour appliquer les formats.

table["diabete"] = table["diabete"].map (dico_format)

diabete
0
1
0

diabete
Non
Oui
Non