# Contexte

Les maladies cardiovasculaires (MCV) sont responsables de millions de décès chaque année.

Dans ce projet, vous avez été mandatés par une équipe de recherche pour utiliser vos compétences en Python et les bibliothèques **pandas**, **matplotlib** et **numpy** afin de créer un programme capable de prédire les risques de maladies cardiaques en analysant 11 caractéristiques clés, comme l’âge, le sexe, et la pression artérielle.

Votre mission ? Sauver des vies en détectant précocement les risques cardiovasculaires et en aidant à prévenir les crises cardiaques et les accidents vasculaires cérébraux. **Vous allez travailler avec un jeu de données pour explorer différentes analyses et visualisations.**

**Objectif**

L'objectif de ce projet est de démontrer comment les techniques de programmation et d'analyse de données peuvent être appliquées pour résoudre des problèmes de santé publique. En identifiant précocement les individus à risque, ce modèle pourrait contribuer à une meilleure gestion et prévention des maladies cardiovasculaires, améliorant ainsi la qualité de vie et réduisant le taux de mortalité lié à ces affections.

Ce projet mettra en lumière l'importance de l'interdisciplinarité entre les sciences de la nature et l'informatique, en montrant comment les compétences en programmation peuvent être utilisées pour des applications pratiques et bénéfiques dans le domaine de la santé.

# Parties du projet

1. Partie 1 : Importer et préparer les données, extraire des informations clés pour effectuer des analyses pertinentes.
2. Partie 2 : Visualiser les données sous forme de graphiques, pour en faciliter l'interprétation.
3. Partie 3 : Analyser les résultats et en tirer des conclusions.

# Livrables : Vous devrez remettre un seul fichier Jupyter Notebook (prenom\_nom\_projet.ipynb) contenant tout le code, les analyses et les visualisations et le fichier de données (.csv).

**À la fin de chaque séance**, vous devez remettre votre travail (**fichiers .ipynb et .csv**) dans la boîte de remise prévue à cet effet sur Moodle.

Vous pourrez continuer à travailler sur votre projet entre chaque séance, mais l**a boite de remise sera fermée**.

Ces remises sont des points de contrôle.

# Évaluation : Lors de l’évaluation (semaine 15), vous aurez des questions se rattachant au projet, mais avec un autre jeu de données. Vous serez amenez à répondre à des questions dans Moodle.

# Jeu de données de prédiction de l’insuffisance cardiaque

# Les données de l’expérience sont dans le fichier prediction\_cardiaque.csv. Voici la description des données s’y trouvant. Ce fichier contient les données de patients à travers le monde.

* **Âge du patient:**  en années
* **Sexe du patient :** M : Masculin, F : Féminin
* **Type de douleur thoracique :** TA : Angine Typique, ATA : Angine Atypique, NAP : Douleur Non Angineuse, ASY : Asymptomatique
* **Pression artérielle au repos :** pression artérielle au repos (en mm Hg)
* **Cholestérol :** cholestérol sérique (en mg/dl)
* **Glycémie à jeun :** 1 : si Glycémie à jeun > 120 mg/dl, 0 : sinon
* **Résultats de l’ECG au repos :** Normal : Normal, ST : anomalie de l’onde ST-T (inversions de l’onde T et/ou élévation ou dépression du segment ST de > 0,05 mV), LVH : hypertrophie ventriculaire gauche probable ou certaine selon les critères d’Estes
* **Fréquence cardiaque maximale atteinte:** valeur numérique entre 60 et 202
* **Angine induite par l’exercice :** Y : Oui, N : Non
* **Oldpeak :** ST valeur numérique mesurée en dépression
* **Pente du segment ST au pic de l’exercice :** Up : ascendant, Flat : plat, Down : descendant
* **Maladie cardiaque :** classe de sortie 1 : maladie cardiaque, 0 : Normal

# Consignes et information de départ pour le projet:

1. De Moodle, télécharger sur votre ordinateur, dans le dossier réservé au projet, les deux fichiers :
   * Le fichier de données : ***prediction\_cardiaque.csv***
   * Le fichier de code : **420sn1**\_***projet.ipynb***
2. Dans le fichier de départ (ipynb) que vous allez utiliser pour ce projet, les cellules sont déjà créées et organisées dans un ordre précis. Il est essentiel que vous respectiez cet ordre et le contenu de chaque cellule. Votre tâche consiste à écrire le code dans les cellules désignées, en suivant les instructions fournies.
3. Faites attention à bien répondre aux bons endroits, le code python dans une **cellule de CODE** et les réponses textuelles dans une cellule de **MARCAGE (*MARKDOWN*)**
4. Assurez-vous que tout votre **bloc note Jupyter** (***notebook***) s’exécute correctement en une seule fois avec le bouton **Exécuter Tout** (***Run All***) de VS Code. Les cellules de code suivent un ordre, certaines reprennent le résultat d’une précédente. **Faites attention à ne pas altérer les données entre deux cellules de code.**
5. Pour vous faciliter cette tâche, utilisez **régulièrement** le bouton **Exécuter Tout** (***Run All***) pour vérifier que l’exécution arrive correctement là où vous êtes, **n’attendez pas d’avoir tout codé pour utiliser ce bouton**.

Suivez les directives indiquées dans le bloc note (.ipynb) de départ.

Bon travail et amusez-vous bien !

# Analyse des températures mensuelles de deux villes en 2023

1. Importer et préparer les données
2. Charger les données dans un dataframe.
3. Visualiser les **10 premières** lignes du dataframe.
4. Modifier les noms des colonnes du dataframe (**PAS CELLES DU FICHIER CSV**) par ‘Laval’ et ‘Montréal’.
5. Afficher les **5 dernières** lignes.
6. En visualisant les données, vous vous rendez compte qu’il en manque. Votre chef d’équipe vous informe qu’il s’agit d’une erreur de saisie et qu’en réalité, ces températures correspondent à 0 degrés Celsius. Écrire le code permettant de remplacer ces données par 0.
7. Vérifier le changement en affichant les 5 dernières lignes de données.
8. Extraire des informations clés pour effectuer des analyses pertinentes
9. Définir des **fonctions** pour répondre aux questions d’analyses :
   * **ATTENTION** : vous devez obligatoirement ajouter de ***docstring*** pour chacune des fonctions que vous définissez, et **commenter** le code.
   1. Écrire une **fonction** *qui permet de créer un DataFrame contenant les températures moyennes mensuelles pour chaque ville*.
   2. Écrire une **fonction** *qui renvoie la température minimale enregistrée pour une ville donnée*.
   3. Écrire une **fonction** *qui renvoie la température maximale enregistrée pour une ville donnée*.
   4. Écrire une **fonction** qui prend en paramètre les températures moyennes et le mois de **décembre par défaut**, et renvoie la ville (Montréal ou Laval) *qui a eu la température la plus élevée en décembre*.
   5. Écrire une **fonction** qui prend en paramètre les températures moyennes et le mois de **juillet** **par défaut**, et renvoie la ville (Montréal ou Laval) *qui a eu la température la plus élevée en juillet*.
   6. Écrire une **fonction** qui ne prend pas de paramètre et *renvoie une liste des mois où les températures des deux villes étaient similaires*.   
      **NB** : On considère que deux températures sont similaires si leur différence est **strictement** **inférieure à 1**.
10. Grouper les données par **mois** et calculer la moyenne des températures pour chaque mois pour chacune des deux villes.
    * **ATTENTION** : vous devez **obligatoirement** utiliser la fonction définie en 2a i), des boucles et **UNIQUEMENT** les fonctions vues dans le cours.
    * **ASTUCE** : Utiliser une liste pour stocker les mois de l’année et des boucles imbriquées pour calculer les moyennes.
11. Utiliser *Matplotlib* pour visualiser les données graphiquement
    1. **Graphique 1 :**
    2. **Graphique 2 :**
    3. **Graphique 3 :**
    4. **Graphique 4 :**
12. Analyser les résultats et en tirer des conclusions

Utiliser les fonctions créés dans la partie 2 et les graphiques pour répondre aux questions suivantes, dans les cellules de code prévues à cet effet.

1. **Question 1** : Quelle ville a eu la température moyenne la plus élevée en décembre 2023 ?
2. **Question 2** : Quelle ville a eu la température moyenne la plus basse en juillet 2023 ?
3. **Question 3** : Y a-t-il des mois où les températures des deux villes étaient similaires ?
4. **Question 4** : Y a-t-il des tendances saisonnières observables dans les données ?