

Données « rythme cardiaque / Apple Watch »

L'objectif est d'afficher sur un graphique l'évolution temporelle d'un rythme cardiaque tel que capté par une Apple Watch 5 puis de déterminer les moments de forte variation.

Les données se trouvent dans un fichier XML : exportExtract.xml.gz (77497 lignes)

- 1) Récupérer le fichier et étudier son contenu : en-tête XML, structure, format...
- 2) Avec Pandas, ouvrir le fichier en ne conservant dans un DataFrame que les instants de captation (heures/min/secondes) et les valeurs du rythme cardiaque
- 3) Afficher sur un graphique l'évolution temporelle du rythme cardiaque
- 4) Rajouter une colonne dans le DataFrame où l'on indique l'écart du rythme cardiaque entre deux mesures.
- 5) Calculer un delta moyen toutes les 30 mesures puis indiquer les 5 minutes où les changements de rythme cardiaque ont été les plus forts.

Données « évolution des hospitalisations Covid-19 »

L'objectif est d'afficher l'évolution de l'incidence du Covid-19 dans chaque région de France puis de partitionner (approche de type *clustering* K-moyennes) les régions de telle sorte que dans chaque classe apparaissent des régions où l'évolution de l'incidence du covid-19 est comparable.

Les données se trouvent dans un fichier CSV : covid-hospit-incid-reg-2022-05-30-19h00.csv (15258 lignes)

- 1) Récupérer le fichier et étudier son contenu notamment afin de connaître son format (paramètres pour lire le CSV)
 - 2) Avec Pandas, ouvrir le fichier et transférer son contenu dans un DataFrame
 - 3) Afficher sur un graphique l'évolution temporelle du taux d'incidence en réanimation : une courbe par région (aide : utiliser la fonction pivot de Pandas)
 - 4) La réalisation de la question 3) a dû passer par la création d'un DataFrame où les valeurs de l'index correspondent aux noms de régions et les colonnes aux jours des relevés. Les valeurs des lignes seront vues comme des vecteurs associés aux régions. Utiliser ces vecteur pour appliquer la méthode `sklearn.cluster.KMeans` (<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>). Créer un modèle (KMeans) puis obtenir les groupes avec la méthode `predict` appliquée au modèle.
 - 5) Afficher un graphique où chaque point correspond à une région en utilisant une couleur différente par classe. NB: les points étant dans un espace de dimension > 3 , une réduction de la dimension est nécessaire.
 - 6) Faire varier le nombre d'itérations (condition d'arrêt des kMeans) et le nombre de classes (clusters) et estimer l'évolution de la cohérence des groupes grâce à l'estimation d'un indice de Rand
- ```
from sklearn import metrics
metrics.adjusted_rand_score(_le résultat de la prédiction_, _labels_de_référence_ou_précédents)
```