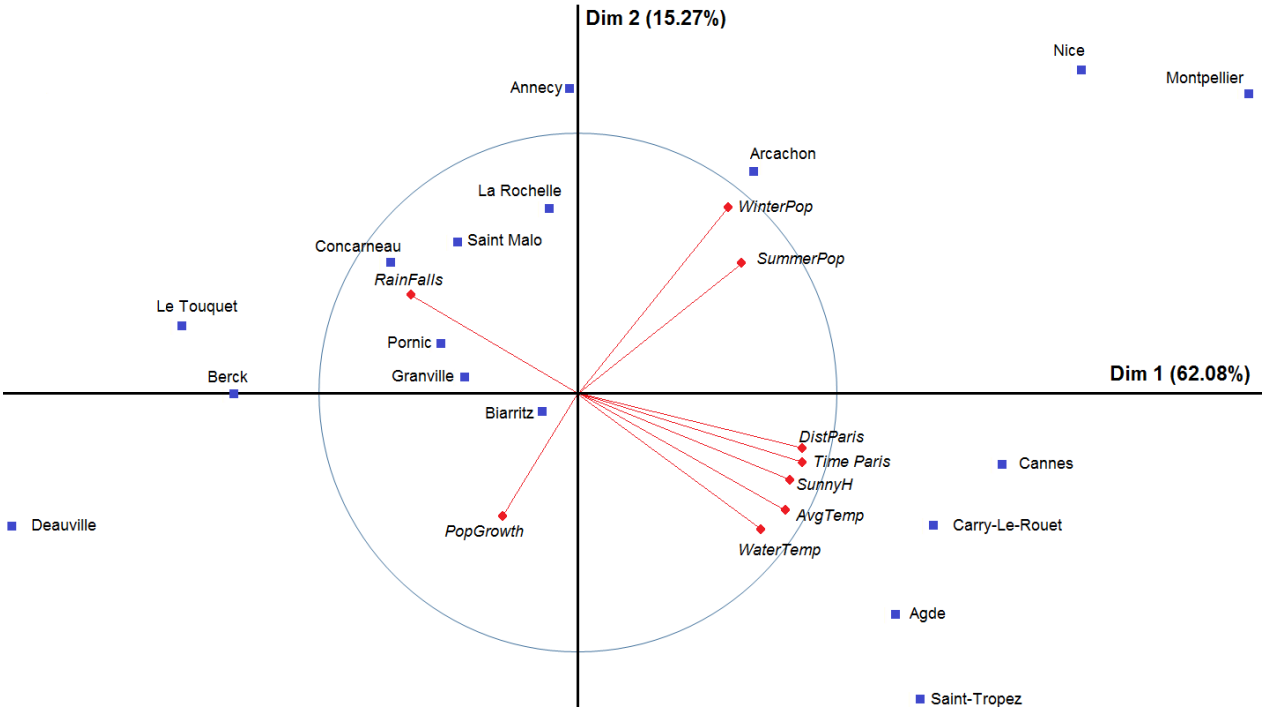
Analyse de données - TP 3

ISEP – 20 Octobre 2020

Instructions : Préparez un rapport incluant le code source et vos résultats, et déposez-le sur Moodle. Pas plus de 2 personnes par groupe. N'oubliez pas de mettre les 2 noms sur le rapport, ou de faire 2 rendus.

A Exercice préliminaire : interprétation d'un cercle de corrélation

Dans cet exercice, nous vous proposons de visualiser une projection des caractéristiques importantes de plusieurs stations balnéaires françaises. Le résultat est présenté ci-après.



— WinterPop : la population hivernale (sans les touristes)
— SummerPop : la population estivale (avec les touristes)
— PopGrowth : l'augmentation de la population (en pourcentage) en comparant décembre et juillet.
— RainFalls : les précipitations moyennes pendant la période estivale (en mm)
— DistParis : la distance depuis Paris (en km)
— TimeParis : le temps de trajet depuis Paris (en h)

— SunnyH : le nombre moyen d'heures d'ensoleillement sur le moins de juillet (en h)
— AvgTemp : la température moyenne en juillet (en degrés Celsius)
— WaterTemp : la température moyenne de l'eau en juillet pour la baignade (en degrés Celsius)

Répondez aux questions suivantes :

1. Que représentent les pourcentages sur les axes des abscisses et des ordonnées ? Commentez.
2. En vous basant uniquement sur la Figure, répondez aux questions suivantes en justifiant vos réponses :
   A. Quelle est la ville où l'eau est la plus chaude en été ?
   B. Quelle est la ville dont la population augmente le plus entre l'hiver et l'été ?
   C. Vrai ou Faux : les attributs "DistParis" et "TimeParis" sont redondants.
   D. Dans les données originales, c'est la ville d'Annecy qui a le plus de millimètres de pluie en juillet. La figure confirme-t-il cette information ? Si oui justifiez. Si non, expliquez comment ça peut être possible.
   E. Vrai ou Faux : ce sont les villes ayant une faible population hivernale qui ont la plus forte augmentation de population en été.
   F. Vrai ou Faux : l'ensoleillement augmente quand on s'éloigne de Paris.
   G. Vrai ou Faux : la population diminue à Annecy pendant la période estivale.
   H. Quelles sont les 2 villes les plus similaires ?

B ACP sur les données Iris

Les Iris de Fisher sont un jeu de données très connu décrivant 150 fleurs de type "Iris" à partir des caractéristiques de leurs pétales et de leurs sépales. Ces données contiennent 3 classes pour 3 espèces de fleurs (50 de chaque) : Iris Virginica, Iris Versicolor et Iris Setosa. L'une des trois classes est linéairement bien séparable, les 2 autres non. L'objectif de cet exercice est d'étudier la visualisation de ces données à l'aide de l'ACP.

1. Récupérez les données iris à avec la bibliothèque panda. Utilisez directement les données Iris de panda.
2. A chaque classe, associez une couleur qui vous servira pour les affichages.
3. Normalisez les données afin qu'elles soient centrées et réduites. Pour ce faire, vous pouvez utiliser le package *StandardScaler* de sklearn, puis la fonction du même nom.

```
#Normalize data
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data = scaler.fit_transform(data)
```

4. Visualisez les données en 2 dimensions avec une ACP. Vous pouvez utilisez des couleurs pour les différentes classes. Commentez.

5. Ecrivez une fonction permettant de tracer un cercle de correlation comme celui vu en cours (vous pouvez vous aider d'internet, mais citez vos sources et commentez votre code). A partir de la figure obtenue, que pouvez-vous dire sur la correlation entre les différents attributs de ces données ?

C Données golub (MDS, LLE et Isomap)

On va maintenant s'intéresser aux données génétiques du dataset "golub". Ces données contiennent les niveaux d'expressions de 7129 gènes prises sur 72 échantillons. Chaque échantillon est associé à une variante de la leucémie, AML(25) et ALL(47). Notre objectif est de visualiser les 72 échantillons sur un plan en 2D.

1. Chargez le jeu de données *Golub_data* en faisant très attention aux paramètres d'ouverture. Que remarquez vous par rapport au descriptif ? Faites les transformations si nécessaire, puis normalisez les données.
2. Ouvrez maintenant les labels dans le fichier *Golub_class2* en faisant aussi attention aux paramètres.
3. Réalisez une ACP sur les données *Golub* et visualisez le résultat. Projetez les labels sur votre ACP. Commentez.
4. Réalisez une MDS sur ce jeu de données en utilisant la fonction **MDS** du package du même nom. Qu'observez-vous par rapport aux résultats obtenus avec l'ACP ?
5. En utilisant le package *LocallyLinearEmbedding* de sklearn, appliquez la fonction LLE sur les données *Golub* en faisant varier le nombre de voisins à 3, 5, 8, 10, 12 et 15. Analysez les résultats et déterminez le(s) meilleur(s) paramètre(s) de voisinage. Remarque : la fonction *subplot* pourra être pratique ici.
6. Analysez maintenant la fonction **Isomap** du package du même nom. Appliquez cette fonction sur les données *Golub* en faisant varier le nombre de voisins à 4, 8, 10, 13, 16 et 20.
7. Concluez sur ces différentes méthodes.

D Données Alon

Chargez les données *Alon*, décrivez-les et ré-appliquez les approches vues dans l'exercice précédent.