**TP python A rendre le lundi 29 Janvier avant 12h00.**

**Partie I: (06 points)**

1. Traduire les algorithmes du devoir en programme python.

**Partie II (14 points)**

**Numpy:**

1. Écrire un programme qui produit un tableau à une dimension avec 100 éléments qui sont les nombres pairs entre 2 et 200
2. Créez un tableau NumPy de taille 10X10 qui représente la table de multiplication
3. À partir du tableau de la table de multiplication de la question 2, affichez que les diagonales (c'est-à-dire uniquement les nombres 1, 4, 9, 16 et ainsi de suite). Utilisez le masque booléen.
4. Construisez un tableau de 10 entiers sélectionnés au hasard (entre 1 à 100) et affichez uniquement les nombres impairs.
5. Construisez un tableau de 10 entiers sélectionnés au hasard (entre 1 à 100) et trouvez le plus petit nombre supérieur à 50
6. Construisez un tableau de 10 entiers choisis au hasard (entre 1 à 100) et trouvez les cinq plus grands nombres.

**Pandas :**

1. **Base « etudiant » :exemple de manipulation**

Étape 1. Importez les bibliothèques nécessaires

Étape 2. Importez l'ensemble de données à partir du fichier « etudiant.csv »

Étape 3. Attribuez-les à une variable appelée df (dataframe).

Étape 4. Pour les besoins de cet exercice, découpez les données de la colone «école» à la colonne «gardien».

Étape 5. Créez une fonction lambda qui mettra les chaînes de caractères en majuscule.

Étape 6. Mettre en majuscule les variables Mjob et Fjob

Étape 7. Affichez les derniers éléments de l'ensemble de données.

Étape 8. Avez-vous remarqué que les données d'origine sont toujours en minuscules? Pourquoi donc? Corrigez-le et mettez en majuscule Mjob et Fjob.

Étape 9. Créez une fonction appelée est\_majeur qui renvoie une valeur booléenne dans une nouvelle colonne appelée legal\_drinker (considérez comme majeurs les plus de 17ans )

**B) Cette fois, vous créerez des données**

Étape 1. Importez les bibliothèques nécessaires

Étape 2. Créez les 3 DataFrames à partir des données brutes suivantes

raw\_data\_1 = {

'subject\_id': ['1', '2', '3', '4', '5'],

'prénom': ['Alex', 'Amy', 'Allen', 'Alice', 'Ayoung'],

'nom de famille': ['Anderson', 'Ackerman', 'Ali', 'Aoni', 'Atiches']}

raw\_data\_2 = {

'subject\_id': ['4', '5', '6', '7', '8'],

'prénom': ['Billy', 'Brian', 'Bran', 'Bryce', 'Betty'],

'last\_name': ['Bonder', 'Black', 'Balwner', 'Brice', 'Btisan']}

raw\_data\_3 = {

'subject\_id': ['1', '2', '3', '4', '5', '7', '8', '9', '10', '11'],

'test\_id': [51, 15, 15, 61, 16, 14, 15, 1, 61, 16]}

Étape 3. Attribuez chacun à une variable appelée data1, data2, data3

Étape 4. Joignez les deux dataframes le long de lignes et attribuez les à all\_data

Étape 5. Joignez les deux dataframes le long des colonnes et attribuez les à all\_data\_col

Étape 6. Affichez les données data3

Étape 7. Fusionnez all\_data et data3 avec la clef subject\_id¶

Étape 8. Fusionnez uniquement les données qui ont le même 'subject\_id' sur data1 et data2

Étape 9. Fusionnez toutes les valeurs de data1 et data2, avec les enregistrements correspondants des deux côtés, le cas échéant.

**C) Filtrage et tri des données**

On utilisera les données de l’euro 2012 (foot.csv)

Étape 1. Importez les bibliothèques nécessaires

Étape 2. Importez l'ensemble de données à partir de foot.csv.

Étape 3. Affectez-le à une variable (dataframe) appelée euro12.

Étape 4. Sélectionnez uniquement la colonne Objectif.

Étape 5. Combien d'équipes ont participé à l'Euro2012?

Étape 6. Quel est le nombre de colonnes dans l'ensemble de données?

Étape 7. Affichez uniquement les colonnes Équipe, Cartons jaunes et Cartons rouges et attribuez-les à un base de données appelée discipline

Étape 8. Triez les équipes par cartons rouges, puis par cartons jaunes

Étape 9. Calculez la moyenne des cartons jaunes attribués par équipe

Étape 10. Filtrer les équipes qui ont marqué plus de 6 buts

Étape 11. Sélectionnez les équipes commençant par G

Étape 12. Sélectionnez les 7 premières colonnes

Étape 13. Sélectionnez toutes les colonnes sauf les 3 dernières.

Étape 14. Présentez uniquement les précisions de tir de l’Angleterre, de l’Italie et de la Russie

**D) Le vent : quelques calculs simples.**

**NB:** Les données ont été modifiées pour contenir certaines valeurs manquantes, identifiées par NaN.

L'utilisation de pandas devrait faciliter cet exercice.

Vous devriez être en mesure d'effectuer toutes ces opérations sans utiliser une boucle for ou une autre construction en boucle.

Après importation des données «vent.csv» comme dataframe, on aura le format suivant:

"" "

Yr Mo Dy RPT VAL ROS KIL SHA BIR DUB CLA MUL CLO BEL MAL

61 1 1 15,04 14,96 13,17 9,29 NaN 9,87 13,67 10,25 10,83 12,58 18,50 15,04

61 1 2 14,71 NaN 10,83 6,50 12,62 7,67 11,50 10,04 9,79 9,67 17,54 13,83

61 1 3 18,50 16,88 12,33 10,13 11,17 6,17 11,25 NaN 8,50 7,67 12,75 12,71

"" "

Les trois premières colonnes sont l'année, le mois et le jour. Les 12 colonnes restantes indiquent les vitesses moyennes du vent en nœuds à 12 endroits en Irlande ce jour-là.

Étape 1. Importez les bibliothèques nécessaires

Étape 2. Importez l'ensemble de données à partir de vent.csv

Étape 3. Attribuez-les à une variable appelée data et remplacez les 3 premières colonnes par un index datetime approprié.

Étape 4. Année 2061? Avons-nous vraiment des données de cette année? Créez une fonction pour la corriger et appliquer la fonction .

Étape 5. Définissez les bonnes dates comme index. Faites attention au type de données, il doit être datetime64 [ns].

Étape 6. Calculez le nombre de valeurs manquantes pour chaque variable. Elles doivent être ignorées dans tous les calculs ci-dessous.

Étape 7. Calculez le nombre total de valeurs non manquantes.

Étape 8. Calculez les vitesses moyennes des vents sur tous les emplacements et toutes les heures (moyenne globale).

Étape 9. Créez un DataFrame appelé loc\_stats et calculez les vitesses du vent min, max et moyen et les écarts-types des vitesses du vent à chaque emplacement sur tous les jours (stats par variable).

Étape 10. Créez un DataFrame appelé day\_stats et calculez la vitesse du vent minimale, maximale et moyenne et les écarts-types des vitesses du vent sur tous les emplacements à chaque jour (stats par jour).

Étape 11. Trouvez la vitesse moyenne du vent en janvier pour chaque emplacement.

Traitez janvier 1961 et janvier 1962 comme janvier.

Étape 15. Calculez les vitesses du vent minimale, maximale et moyenne et les écarts types des vitesses du vent à tous les emplacements pour chaque semaine (supposons que la première semaine commence le 2 janvier 1961) pour les 52 premières semaines.

1. **Visualisation**

Achat au détail en ligne¶

Introduction:

Étape 1. Importez les bibliothèques nécessaires

Étape 2. Importez l'ensemble de données à partir de vente\_detail.csv.

Étape 3. Affectez-le à une variable appelée online\_rt

Remarque : si vous recevez une erreur de décodage utf-8, définissez encoding = 'latin1' dans pd.read\_csv ().

Étape 4. Créez un histogramme avec les 10 pays qui ont le plus de « Quantité » commandée, à l'exception du Royaume-Uni

Étape 5. Exclure les entrées de quantité négatives

Étape 6. Créez un nuage de points avec la quantité par unité de prix par CustomerID pour les 3 premiers pays (sauf le Royaume-Uni)

Étape 7. Examinez pourquoi les résultats précédents semblent si peu informatifs.

Cette section peut sembler un peu fastidieuse à parcourir. Mais j'y ai pensé comme une sorte de simulation des problèmes que l'on pourrait rencontrer en traitant des données et d'autres personnes. De plus, il y a un prix à la fin (c'est-à-dire la section 8).

Étape 7.1 Regardez la première ligne de code de l'étape 6. Et essayez de déterminer si cela entraîne un problème quelconque.

Étape 7.1.1 Affichez les premières lignes de ce DataFrame.

Étape 7.1.2 Affichez le type de UnitPrice

Étape 7.1.3 Extrayez les données de online\_rt pour les ID client 12346.0 et 12347.0.

Étape 7.2 Réinterprétation du problème initial.

Pour réitérer la question que nous traitions :

"Créez un nuage de points avec la quantité par prix unitaire par ID client pour les 3 premiers pays"

La question est ouverte à un ensemble d'interprétations différentes. Nous devons lever l'ambiguïté.

Nous pourrions faire un seul graphique en examinant toutes les données des 3 premiers pays. Ou nous pourrions faire un graphique par pays. Pour garder les choses cohérentes avec le reste de l'exercice, restons-en à cette dernière option. Donc c'est réglé.

Mais "top 3 pays" par rapport à quoi ? Deux réponses s'imposent : le volume total des ventes (c'est-à-dire la quantité totale vendue) ou les ventes totales (c'est-à-dire le chiffre d'affaires). Cet exercice concerne le volume des ventes, alors restons-en là.

Étape 7.2.1 Découvrez les 3 premiers pays en termes de volume de ventes.

Étape 7.2.2

Maintenant que nous avons les 3 premiers pays, nous pouvons nous concentrer sur le reste du problème :

"Quantité par UnitPrice par CustomerID".

Nous devons déballer cela.

La partie "par CustomerID" est simple. Cela signifie que nous allons tracer un point par CustomerID sur notre tracé. En d'autres termes, nous allons regrouper par CustomerID.

"Quantity per UnitPrice" est plus délicat. Voici ce que nous savons :

Un axe représentera une Quantité attribuée à un client donné. C'est facile; nous pouvons simplement tracer la quantité totale pour chaque client. L'autre axe représentera un UnitPrice attribué à un client donné. N'oubliez pas qu'un seul client peut avoir n'importe quel nombre de commandes avec des prix différents, donc résumer les prix n'est pas très utile. De plus, ce que nous entendons par « prix unitaire par client » n'est pas très clair ; ça sonne comme le prix du client ! Une alternative raisonnable consiste à attribuer à chaque client le montant moyen que chacun a payé par article. Réglons donc cette question de cette manière.

Étape 7.3 Modifier, sélectionner et tracer des données

Étape 7.3.1 Ajoutez une colonne à online\_rt appelée Revenue et calculez

les revenus (Quantity \* UnitPrice) de chaque vente.

Nous l'utiliserons plus tard pour déterminer un prix moyen par client.

Étape 7.3.2 Regroupez par CustomerID et Country et découvrez le prix moyen (AvgPrice) que chaque client dépense par unité.

Étape 7.3.3 Tracer

Étape 7.4 Que faire maintenant ?

Nous ne sommes pas beaucoup mieux lotis que ce avec quoi nous avons commencé. Les données sont encore extrêmement dispersées et ne semblent pas assez informatives.

Mais il ne faut pas désespérer ! Il y a deux choses à réaliser :

1) Les données semblent être biaisées vers les axes (par exemple, nous n'avons aucune valeur où Quantité = 50000 et AvgPrice = 5). Cela pourrait donc suggérer une tendance.

2) Nous avons plus de données ! Nous n'avons examiné que les données de 3 pays différents et elles sont représentées sur des graphiques différents.

Donc : nous devrions tracer les données quel que soit le pays et, espérons-le, voir un graphique moins dispersé.

Étape 7.4.1 Tracer les données pour chaque CustomerID sur un seul graphique.

Étape 7.4.2 Zoom avant pour que nous puissions voir cette courbe plus clairement.

8. Tracez un graphique linéaire montrant les revenus (y) par UnitPrice (x).

L'étape 7 nous a-t-elle donné des informations sur les données ? Sûr! Au fur et à mesure que le prix moyen augmente, la quantité commandée diminue. Mais ce n'est guère surprenant. Ce serait étonnant que ce ne soit pas le cas !

Néanmoins, le taux de baisse de la quantité est si drastique que je me demande comment nos revenus évoluent par rapport au prix des articles. Ce ne serait pas surprenant si cela ne changeait pas tant que ça. Mais il serait intéressant de savoir si la plupart de nos revenus proviennent d'articles chers ou bon marché, et à quoi ressemble cette relation.

C'est ce que nous allons faire maintenant.

8.1 Grouper le prix unitaire par intervalles de 1 pour les prix [0,50), et additionner les quantités d’une part et les revenus d’autres part.

8.3 Tracer.

8.4 Rendez-le plus joli.

l'axe des abscisses a besoin de valeurs.

l'axe des y n'est pas si facile à lire ; montrer en termes de millions.

BONUS : Créez votre propre question et répondez-y.