**Examen 2021 : Data mining and Data visualization en R**

**Exercice 1**

1. **Exploration des données**
2. Importer le fichier de données csv ‘*Historical\_Significant\_Volcanic\_Eruption\_Locations*’ dans votre environnement R. Comme son nom l’indique, ce fichier recense les événements d’éruptions volcaniques dans le monde. Prenez le temps de bien comprendre la signification des différentes colonnes. Décrivez à partir de cette observation le jeu de données.
3. Nous nous intéresserons seulement aux éruptions intervenues après l’an 0. Supprimer les lignes concernées par ce filtrage.
4. Déterminer quels sont les 3 pays les plus représentés dans ce jeu de données. Vous pourrez vous servir notamment de la fonction ***sort()***.
5. Utiliser les colonnes des années, des mois et des jours pour ajouter une colonne de date complète (au format YYYY-mm-dd). Vous observerez la présence de valeurs manquantes sur les mois et les jours, celles-ci concernent principalement les éruptions mal répertoriées ou trop anciennes. On pourra remplacer les valeurs manquantes des mois par janvier et les valeurs manquantes de jours par 1.
6. Modifier la classe de cette nouvelle colonne vers la classe ***Date***. Vous pourrez vous servir de la fonction ***as.Date()***.
7. Classer à présent les lignes du jeu de données en fonction de la date des événements, du plus ancien au plus récent. Vous pourrez vous servir de la fonction ***order()***.
8. Tracer la série « temporelle » des dégâts financiers des éruptions. Commenter.
9. La colonne *VOL\_ID* contient les identifiants uniques des volcans présents dans ce jeu de données. Déterminer combien de volcans sont représentés ici.
10. A l’aide d’un graphique, représenter la répartition des éruptions par catégorie du nombre de morts. La colonne DEATHS\_AMOUNT\_ORDER\_TOTAL classifie de 1 à 4 les éruptions en fonction du nombre de morts, 1 étant la catégorie la plus faible. Commenter et expliquer les résultats observés.
11. Déterminer les probabilités empiriques que les éruptions soient suivies d’un tremblement de terre, d’un tsunami et des deux en même temps.
12. **Représentation sous forme de carte Leaflet**
13. Représenter sur une carte leaflet l’emplacement des volcans représentés.
14. Pour chaque volcan, ajouter un pop-up listant le nom du volcan, son altitude, le nombre d’éruptions répertoriées, l’accumulation du nombre de morts et des dégâts financiers.
15. Vous pouvez exporter la carte sous forme de fichier HTML pour l’observer plus facilement. Vous pourrez joindre ce fichier à votre rapport.

**Exercice 2**

1. **Covid-19 : Décès à l’échelle globale**
2. Importer dans votre environnement R le fichier txt suivant : ‘*Deaths\_Global*’. Ce fichier recense le nombre de décès dus au Covid-19 entre le 22 janvier et le 23 mai 2020, sur une échelle quotidienne. Il est extrait des travaux de l’université de Jonhs-Hopkins.
3. A l’aide de ce fichier, établir le classement des pays les plus touchés au 23 mai.
4. Nous nous intéresserons seulement aux pays suivants : France, Italie, Espagne, Allemagne, Royaume-Uni, Chine, Etats-Unis et Brésil. Créez donc une nouvelle variable qui conserve seulement les informations de ces pays. Pour certains pays, plusieurs lignes sont disponibles car il y a un détail régional. Prenez la ligne qui ne correspond à aucune région, celle-ci concerne le pays entier. Pour la Chine, cette ligne n’est pas disponible. Il faudra donc sommer les informations régionales. Vous pourrez vous servir de la fonction ***colSums()***. Nous appellerons cette variable ***data\_deaths***.
5. Ce fichier ne recense que le cumul de décès à partir du 22 janvier. A partir de votre nouvelle variable, créez un nouveau tableau qui recense l’évolution quotidienne du nombre de décès (nombre absolu de décès par jour). Nous appellerons cette variable ***data\_deaths\_evol***.
6. A l’aide de la variable ***data\_deaths***, afficher sur un graphique les cumuls de décès pour chacun des pays. Cela permettra de comparer les évolutions en prenant soin d’afficher une légende et de clarifier les noms des axes. Commenter le graphique.
7. Faire de même pour la variable ***data\_deaths\_evol***. Vous devriez observer, pour la France notamment, des jours où l’évolution est négative. Comment cela est-il possible ?
8. **Covid-19 : Evolution à l’échelle de la France**
9. Importer deux nouveaux fichiers : ‘*Recovered\_Global*’ et ‘*Confirmed\_Global*’. Le premier recense le nombre de guéris tandis que le second recense le nombre de cas confirmés. Ces deux fichiers sont au même format que le fichier de la Partie 1.
10. Cette fois-ci, nous nous intéresserons à la France seulement. Isolez donc les données qui nous intéressent dans une ou deux variables.
11. Calculer le taux de guérison en comparant le nombre de cas au nombre de guérison. Vous pourrez également en observer l’évolution temporelle. Commenter l’évolution de ces chiffres.
12. A présent, cumuler sur un même graphique 3 courbes : nombre de morts en France (***data\_deaths***), nombre de cas confirmés et nombre de cas guéris.
13. Ajouter également des courbes verticales aux dates du 17 mars, date de début du confinement et du 11 mai, date du déconfinement. Commenter l’ensemble du graphique.

**Exercice 3**

1. **Données de température : reconstruction de valeurs manquantes par modèle linéaire**
2. Importer dans votre environnement R les fichiers de l’exercice 3 (8 fichiers de stations + 1 fichier de metadata). Chaque fichier contient des données de température quotidienne pour une station dans le sud-est de la France.
3. Faites une rapide observation des données. Que remarquez-vous à propos de la station Ambert par rapport aux autres stations ?
4. Pour mieux observer cette anomalie, superposer les données de température de Montpellier (par exemple) avec celles d’Ambert. (Attention : l’axe des x doit avoir des données au format ‘*Date*’).
5. Il est ici impossible de reconstruire les données manquantes de la station d’Ambert avec la fonction ***na.approx()***. Pour cela, nous allons devoir nous servir d’une des autres stations environnantes. Avant toute chose, éliminez les stations qui n’ont pas 100% des données sur l’année 2018.
6. Avec les stations restantes, construisez un graphique comparatif pour trouver la plus adaptée à être utilisée pour reconstruire les données de la station d’Ambert. Vous pourrez pour cela vous servir de la fonction ***pairs.panel()*** du package ***psych*** (comme vu lors du TD 3).
7. Déduisez de ce graphe quelle station est la plus adaptée pour reconstruire les données de la station d’Ambert si on se sert d’un modèle linéaire.
8. Confortez votre choix en plaçant rapidement toutes les stations sur une carte, ainsi qu’en observant les altitudes des stations (voir le fichier de *metadata*). Commenter ce choix.
9. Etablissez à présent le modèle linéaire entre la station d’Ambert et la station choisie.
10. Servez-vous finalement du modèle pour reconstruire les données manquantes de la station d’Ambert. Pour cela, utilisez la fonction ***predict()*** qui prendra en *input* le modèle fraichement créé et les données de la station choisie (Lien d’aide : <https://www.dataquest.io/blog/statistical-learning-for-predictive-modeling-r/>).
11. Remplacer pour finir les NA dans la variable de la station Ambert par les données « prédites » et renouveler le graphique de la question 3.