PYTHON BASIQUE

Projet Python : Analyse des données de santé « Cas du Paludisme en Afrique »

Certification : Data Manager

Etudiant : Marcellin SANOU

RAPPORT DETAILLÉ DU PROJET

« Analyse des données de santé « Cas du Paludisme en Afrique »

Mai 2023

Intitulé du projet : Analyse des données de santé « Cas du Paludisme en Afrique »

Sources de l'ensemble des données (Dataset) : https://www.kaggle.com/datasets/lydia70/malaria-in-africa

Contenu du Dataset « Africa Malaria » :

L'ensemble de données "Africa Malaria" comprend des données sur les pays africains de 2007 à 2017, avec les caractéristiques suivantes :

- Code de pays ISO-3 unique : Chaque pays est identifié par un code de pays ISO-3, qui est un code standardisé utilisé pour représenter les pays dans les données internationales.
- Latitude et longitude : Pour chaque pays, l'ensemble de données fournit également les coordonnées de latitude et de longitude, qui donnent la position géographique approximative du pays.
- Cas de paludisme signalés : L'ensemble de données comprend des informations sur les cas de paludisme signalés dans chaque pays et chaque année. Ces données peuvent inclure le nombre total de cas, le nombre de cas selon le sexe, l'âge ou d'autres caractéristiques démographiques, ainsi que la gravité des cas signalés.
- Mesures préventives : L'ensemble de données fournit également des données sur les mesures préventives prises pour lutter contre le paludisme dans chaque pays. Cela peut inclure des informations sur les campagnes de sensibilisation, les programmes de distribution de moustiquaires, les traitements médicaux administrés, etc.

1- Importation des bibliothèques pandas, numpy et matplotlib, pour manipuler les données et effectuer des analyses statistiques

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt
```

2- Chargement du fichier du Dataset dans un dataframe pandas

Lien de téléchargement du Dataset : https://www.kaggle.com/datasets/lydia70/malaria-in-africa

```
df = pd.read_csv("DatasetAfricaMalaria.csv")
```

3- Nettoyage et structuration des données pour l'analyse

3.1. Copie du Dataset

```
dfc = df.copy()
```

3.2. Affichage des 5 premières lignes du Dataset

dfc.head()

3.3. Affichage des 5 dernières lignes du Dataset

dfc.tail()

3.4. Examen des différentes colonnes et types de données dans le Dataset

dfc.info()

Le résultat nous dit que notre DataFrame contient 594 lignes classées de l'index 0 à 593 et de 27 colonnes.

- **Nombre d'index dans notre Dataset**

dfc.index

Le résultat nous donne 594 lignes dans notre DataFrame

- **Conversion du type de données de la colonne « Malaria cases reported » du nombre float en nombre entier**

dfc['Malaria cases reported'] = dfc['Malaria cases reported'].astype('Int64')

- **Renommons certaines colonnes de notre Dataset**

Listons les noms de colonnes du Dataset avant renommage

list(dfc)

- **Renommage des colonnes du Dataset**

dfc.rename(columns = {'Children with fever receiving antimalarial drugs (% of children under age 5 with fever)':'% of children under age 5 with fever receiving antimalarial drugs', 'Intermittent preventive treatment (IPT) of malaria in pregnancy (% of pregnant women)':'% of pregnant women using Intermittent preventive treatment (IPT) of malaria in pregnancy', 'People using safely managed drinking water services (% of population)':'% of population using safely managed drinking water services', 'People using safely managed drinking water services, rural (% of rural population)':'% of rural population using safely managed drinking water services', 'People using safely managed drinking water services, urban (% of urban population)':'% of urban population using safely managed drinking water services', 'People using safely managed sanitation services (% of population)':'% of population using safely managed sanitation services', 'People using safely managed sanitation services, rural (% of rural population)':'% of rural population using safely managed sanitation services', 'People using safely managed sanitation services, urban (% of urban population)':'% of urban population using safely managed sanitation services', 'People using at least basic drinking water services (% of population)':'% of population using at least basic drinking water services', 'People using at least basic drinking water services, rural (% of rural population)':'% of rural

population using at least basic drinking water services', 'People using at least basic drinking water services, urban (% of urban population)':'% of urban population using at least basic drinking water services', 'People using at least basic sanitation services (% of population)':' % of population using at least basic sanitation services', 'People using at least basic sanitation services, rural (% of rural population)':'% of rural population using at least basic sanitation services', 'People using at least basic sanitation services, urban (% of urban population)':'% of urban population using at least basic sanitation services', 'geometry':'Localisation'}, inplace = True)

Affichage des nouvelles colonnes renommées de notre Dataset

list(dfc)

4- Analyse exploratoire des données

4.1. Statistiques descriptives du Dataset

df.describe()

Cette commande affiche en résultat, le nombre de valeurs renseignées (count) pour chacune des 27 colonnes du DataFrame, la moyenne des valeurs renseignées par colonne (mean) de notre DataFrame, les valeurs maximales (max) et minimales (min) renseignées de notre DataFrame, etc.

4.2. Analyses statistiques pour identifier les tendances des maladies, les facteurs de risque

- **Tendances annuelles des cas de paludisme, classées par année du nombre de cas le plus élevé de paludisme au nombre de cas le plus faible**

som_tendances_annuelles = dfc.groupby("Year")["Malaria cases reported"].sum().sort_values(ascending=False)

Affichage du résultat de « som_tendances_annuelles »

som_tendances_annuelles

Le résultat de cette commande ci-dessus nous montre que l'année 2017 a connu le plus grand nombre de cas de paludisme et le nombre de cas de paludisme n'a fait qu'augmenter d'année en année de 2007 à 2017.

Moyenne des tendances

moy_tendances_annuelles = dfc.groupby("Year")["Malaria cases reported"].mean().sort_values(ascending=False)

Affichage du résultat de « moy_tendances_annuelles »

moy_tendances_annuelles

Le résultat de cette commande ci-dessus nous montre que l'année 2017 a connu le plus grand nombre de cas de paludisme et le nombre de cas de paludisme n'a fait qu'augmenter d'année en année de 2007 à 2017.

- **Tendances par pays des cas de paludisme de 2007-2017, classé par pays ayant le nombre de cas de paludisme le plus élevé au nombre de cas le plus faible**

```
som_tendances_pays = dfc.groupby("Country Name")["Malaria cases reported"].sum().sort_values(ascending=False)
```

Affichage du résultat de « som_tendances_pays »

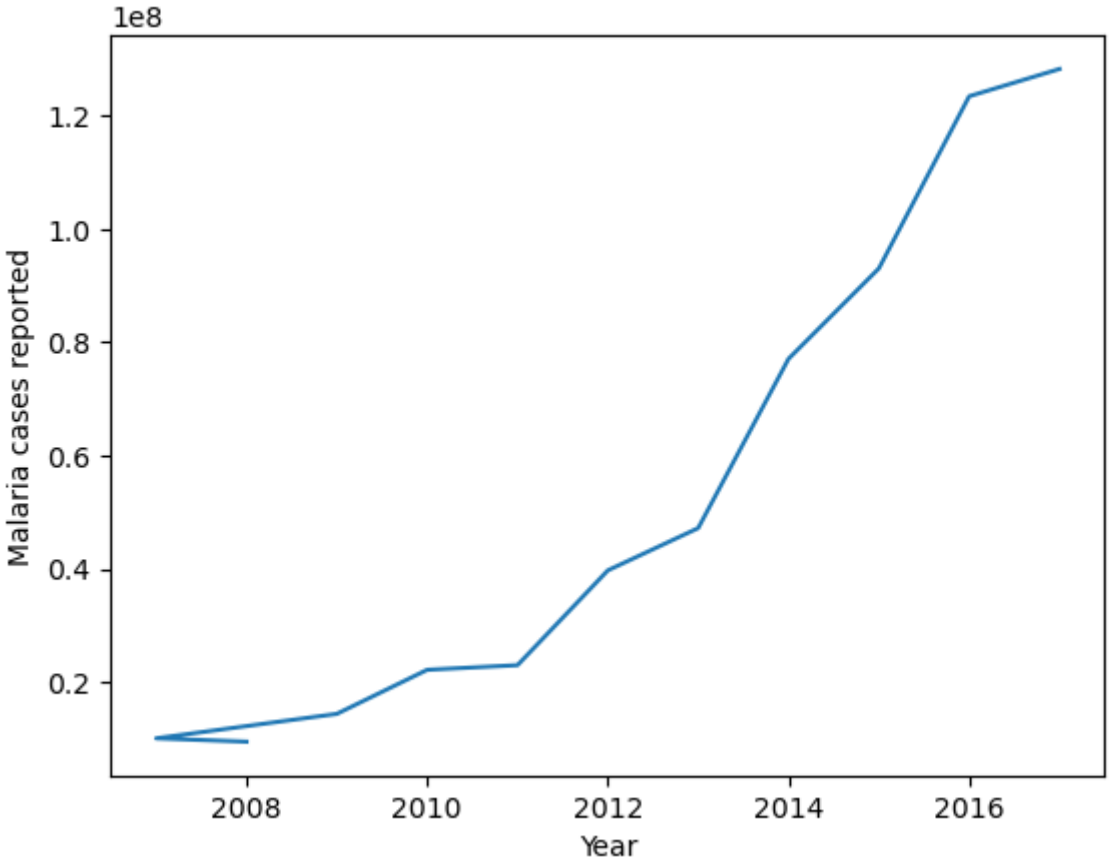
```
som_tendances_pays
```

Le résultat de la commande « som_tendances_pays » nous montre que les 5 pays ayant le plus grand nombre de cas de paludisme reportés de 2007-2017 par ordre décroissant sont : République Démocratique du Congo (RDC), Mozambique, Burkina Faso, Ouganda et Burundi.

- **Visualisation des résultats des tendances annuelles de cas de paludisme en Afrique**

```
som_tendances_annuelles.plot(kind="line")

plt.xlabel("Year")

plt.ylabel("Malaria cases reported")

plt.title("Figure 01: Tendances annuelles des cas de paludisme en Afrique")

plt.show()
```

Figure 01: Tendances annuelles des cas de paludisme en Afrique



Lecture de la Figure 01 : la tendance est que le nombre de cas de paludisme en Afrique n'a fait qu'augmenter d'année en année entre 2007 et 2017

- **Tendances des cas de paludisme en fonction de l'utilisation de moustiquaires imprégnées d'insecticide (% de la population de moins de 5 ans)**

som_tendances_treated_bed_nets = dfc.groupby("Use of insecticide-treated bed nets (% of under-5 population)")["Malaria cases reported"].sum()

som_tendances_treated_bed_nets

Résultat de la commande « som_tendances_treated_bed_nets » : Plus le taux d'enfants de moins de -5 ans dormant sous moustiquaires imprégnés augmentent moins il y a des cas de paludisme

- **Visualisation de la tendance des cas de paludisme en Afrique pour enfants de -5 ans dormant sous moustiquaires imprégnés**

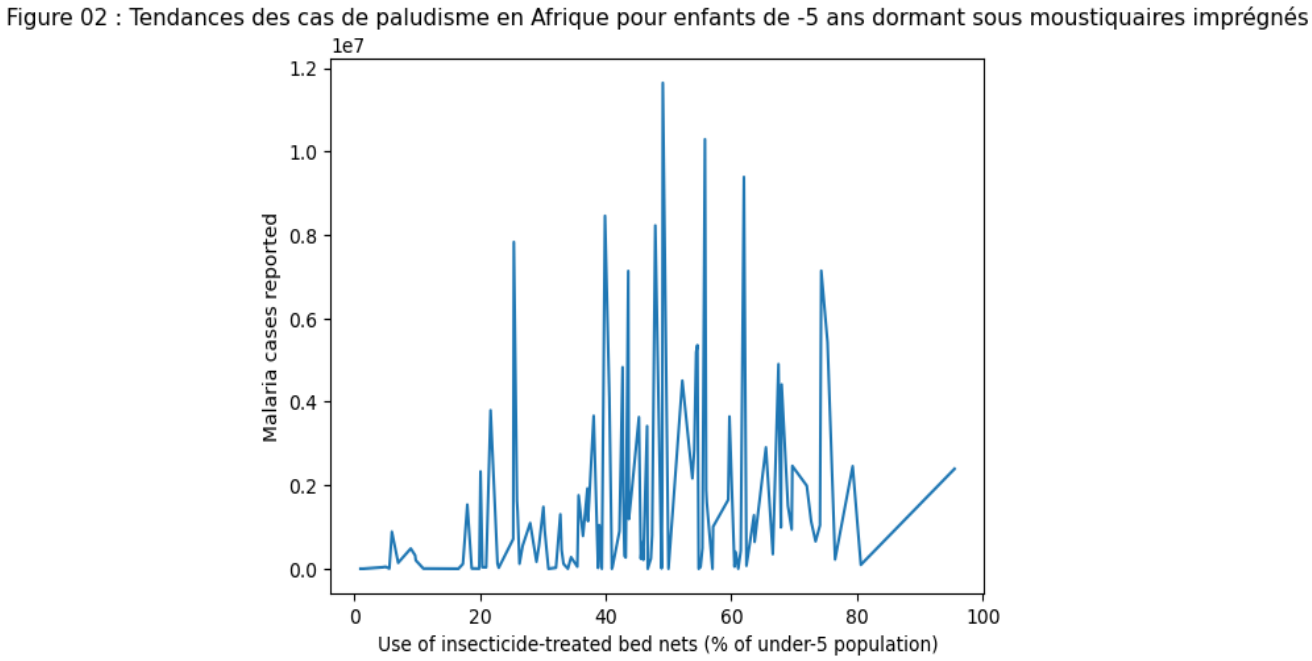
som_tendances_treated_bed_nets.plot(kind="line")

plt.xlabel("Use of insecticide-treated bed nets (% of under-5 population)")

plt.ylabel("Malaria cases reported")

plt.title("Figure 02 : Tendances des cas de paludisme en Afrique pour enfants de -5 ans dormant sous moustiquaires imprégnés")

plt.show()



Lecture de la Figure 02 : la tendance est que plus les enfants de moins de 5 ans dorment sous moustiquaires imprégnés, moins ils ont le paludisme.

4.3. Corrélations entre les variables pour analyser les relations entre les facteurs de risque et les cas de paludisme

Définition : La relation statistique entre deux variables est appelée leur corrélation.

correlation_matrix = dfc.corr()

Affichage de la matrice de corrélation

correlation_matrix

Les valeurs positives de la matrice de corrélation montrent que les 2 variables se déplacent dans la même direction. Cette relation est très faible lorsque la valeur est comprise entre 0 et 0,25 ; faible lorsqu'elle est comprise entre 0,26 et 0,49 ; moyennement forte lorsqu'elle est comprise entre 0,50 et 0,75 et très forte lorsqu'elle est comprise entre 0,76 et 0,99.

Les valeurs négatives de la matrice de corrélation montrent que la valeur d'une des 2 variables augmente tandis que l'autre diminue.

Analyse des corrélations et identification des facteurs de risque potentiels en examinant les valeurs de corrélation

- Corrélations entre les cas de paludisme et les variables démographiques

- ❖ Avec Incidence du paludisme (pour 1 000 habitants à risque)

corr_cases_population_1 = correlation_matrix["Malaria cases reported"]["Incidence of malaria (per 1,000 population at risk)"]

corr_cases_population_1 : affiche la corrélation entre la variable « Malaria cases reported (Nombre de cas de paludisme) » et « Incidence of malaria (per 1,000 population at risk) (Incidence du paludisme (pour 1000 population à risque)) ». Sa valeur est « 0.28850886831479716 » donc positive et la corrélation est très faible entre ces deux variables qui se déplace dans la même direction.

- ❖ Avec % d'enfants de moins de 5 ans ayant de la fièvre recevant des médicaments antipaludiques

corr_cases_population_2 = correlation_matrix["Malaria cases reported"]["% of children under age 5 with fever receiving antimalarial drugs"]

corr_cases_population_2 : affiche la corrélation entre la variable « Malaria cases reported (Nombre de cas de paludisme) » et « % of children under age 5 with fever receiving antimalarial drugs (d'enfants de moins de 5 ans souffrant de fièvre et recevant des médicaments antipaludiques) ». Sa valeur est « 0.30171947845151337 » donc positive et la corrélation est très faible entre ces deux variables qui se déplace dans la même direction.

- ❖ Forte Corrélation entre "% of rural population using safely managed sanitation services" et "% of rural population using safely managed drinking water services"

corr_cases_population_5 = correlation_matrix["% of rural population using safely managed sanitation services"]["% of rural population using safely managed drinking water services"]

corr_cases_population_5 : affiche la corrélation entre la variable « % of rural population using safely managed sanitation services (Pourcentage de la population rurale utilisant des services d'assainissement gérés en toute sécurité) » et « % of children under age 5 with fever receiving antimalarial drugs (d'enfants de moins de 5 ans souffrant de fièvre et recevant des médicaments antipaludiques) ». Sa valeur est « 0.9780628308867829 » donc positive et la corrélation est très forte entre ces deux variables qui se déplace dans la même direction.

5- Machine Learning

5.1. Modélisation des données pour identifier des modèles et des relations importantes dans les données

Utilisons la bibliothèque scikit-learn pour effectuer une régression linéaire et importons le modèle de régression linéaire

from sklearn.linear_model import LinearRegression

Le module Impute de Sklearn (scikit-learn) permet de nettoyer notre dataset des valeurs manquantes qui le composes.

SimpleImputer remplace toute valeur manquante par une statistique ou une constante donnée

from sklearn.impute import SimpleImputer

X représente des variables indépendantes : dans notre cas "% d'enfants de moins de 5 ans dormant sous moustiquaires imprégnées" et "% de femmes enceintes utilisant le traitement préventif intermittent (TPI) du paludisme pendant la grossesse"

X = dfc[["Use of insecticide-treated bed nets (% of under-5 population)", "% of pregnant women using Intermittent preventive treatment (IPT) of malaria in pregnancy"]]

y représente la variable cible : dans notre cas "Year (Année)"

y = dfc["Year"]

Utilisons SimpleImputer pour remplacer toute valeur manquante par une statistique ou une constante donnée. Dans notre cas, la moyenne des valeurs non manquantes

imputer = SimpleImputer(strategy='mean', missing_values=np.nan)

###Transformons dans les variables indépendantes contenues dans X, les valeurs manquantes par np.nan

X_imputed = imputer.fit_transform(X)

Utilisons SimpleImputer pour remplacer toute valeur manquante par une statistique ou une constante donnée. Dans notre cas, la moyenne des valeurs non manquantes

imputer_y = SimpleImputer(strategy='mean')

Appelons la méthode `reshape` sur `y.values`, pour remodeler le tableau en un tableau 2D avec une seule colonne, ce qui est le format d'entrée attendu pour la méthode `fit_transform` de la classe `SimpleImputer`.

y_imputed = imputer_y.fit_transform(y.values.reshape(-1, 1))

###Instancions ensuite un objet modèle de régression linéaire en utilisant la classe LinearRegression().

model = LinearRegression()

###À l'aide de la méthode fit(), entraînons le modèle en utilisant les données X et y.

model.fit(X_imputed, y_imputed)

###Utilisons la méthode predict() pour effectuer des prédictions sur les données d'entraînement X et stockez les prédictions résultantes dans la variable predictions.

predictions = model.predict(X_imputed)

5.2. Evaluation du modèle de Regression Linéaire

###Importation de la fonction "mean_squared_error" du module "sklearn.metrics". Cela permettra de calculer l'erreur quadratique moyenne ou Mean Squared Error(MSE).

from sklearn.metrics import mean_squared_error

###Calcul de l'erreur quadratique moyenne (MSE) entre les valeurs réelles 'y' et les valeurs prédites 'predictions' en utilisant la fonction 'mean_squared_error'.

mse = mean_squared_error(y, predictions)

###Calcul de la racine carrée de l'erreur quadratique moyenne (MSE) en utilisant la fonction `np.sqrt` de numpy.

rmse = np.sqrt(mse)

###Affichage de la valeur du RMSE en utilisant la fonction 'print'. Habituellement, un score RMSE inférieur à 180 est considéré comme un bon score pour un algorithme qui fonctionne modérément ou bien. Dans le cas où la valeur RMSE dépasse 180, nous devons effectuer une sélection de caractéristiques et un réglage des paramètres hyper sur les paramètres du modèle.

print("RMSE:", rmse)

Résultat RMSE est de **3.0520908649792458** qui est inférieur à 180, donc notre algorithme de Machine Learning fonctionne bien.

5.3. Communication des résultats à travers leur visualisation sous forme de graphique

plt.scatter(X['Use of insecticide-treated bed nets (% of under-5 population)'], y)

plt.plot(X['Use of insecticide-treated bed nets (% of under-5 population)'], predictions, color='red')

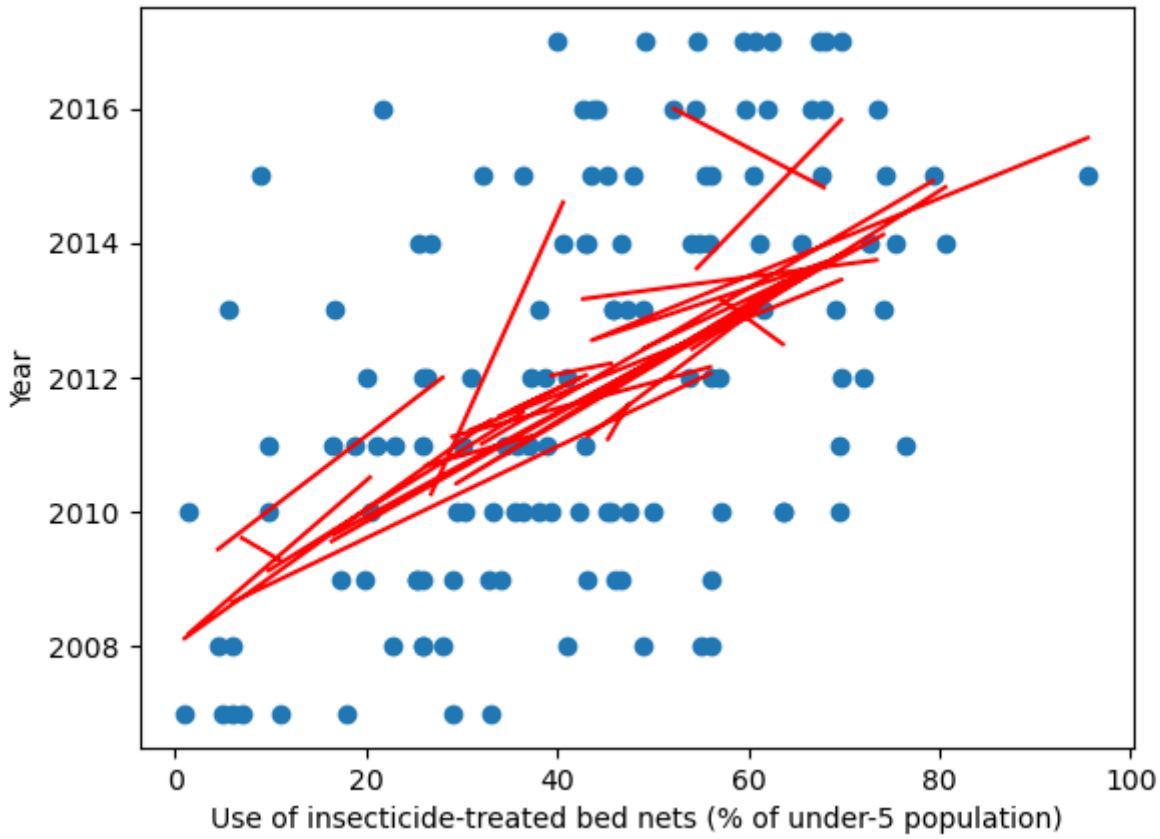
plt.xlabel('Use of insecticide-treated bed nets (% of under-5 population)')

plt.ylabel('Year')

plt.title("Figure 03 : Visualisation des résultats de notre modèle de Regression Linéaire")

plt.show()

Figure 03 : Visualisation des résultats de notre modèle de Regression Linéaire



Lecture de la Figure 03 : le taux d'enfants de moins de 5 ans dormant sous moustiquaires imprégnés, continuera à augmenter d'année en année afin de réduire le nombre de cas de paludisme.

PRINCIPALES RECOMMANDATIONS :

Pour réduire le taux de mortalité dû au Paludisme dans les différents pays africains, nous proposons les recommandations suivantes :

- Augmenter les quantités de moustiquaires imprégnées distribués chaque année dans les différents pays concernés et sensibiliser les populations à dormir sous moustiquaires imprégnées (surtout enfants de moins de 5 ans et femmes enceintes) ;
- Rendre disponibles au sein des pays concernés les médicaments contre le Paludisme au profit des populations (surtout enfants de moins de 5 ans et femmes enceintes) ;
- Améliorer le cadre de vie des populations urbaines et rurales en mettant à leur disposition des services sanitaires bien gérés et de l'eau potable.