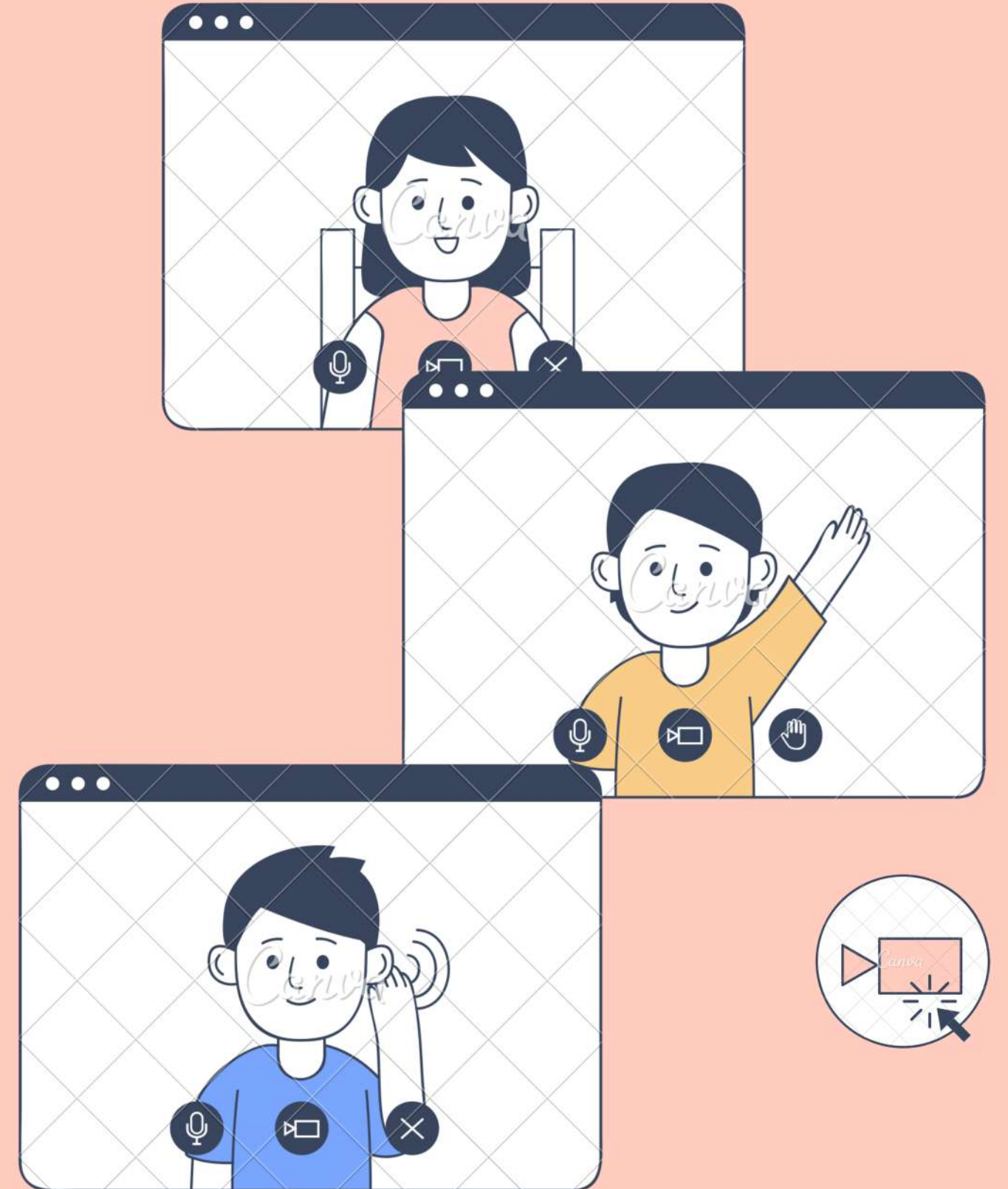
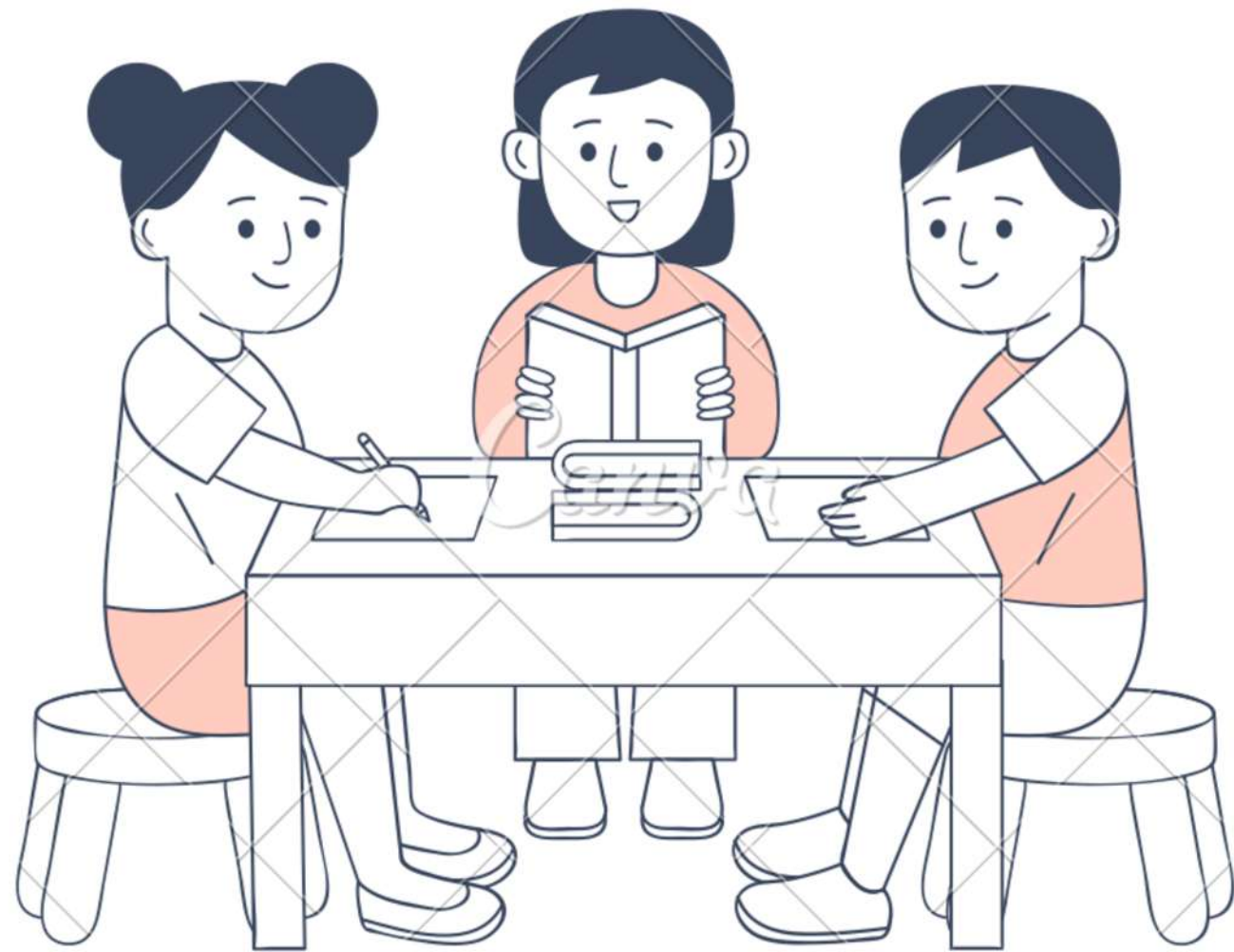


Group Project

Projet intégré





**Hello
classmates!
We are
Group 3!**

created by

RAYEN NASRALLAH



IMPORTATION DES BASES

```
import pandas as pd
df_2018 = pd.read_excel("base finale SDI-2018 (2).xlsx")
df_2018.head(165)
```

	iso	country	Life expectancy 2018	Expect years of schooling 2018	Mean years of schooling 2018	Gross income Level per capita 2018	CO2 emission 2018	Material footprint 2018
0	AFG	Afghanistan	64.5	10.1	3.9	2217.0	0.47	1.20
1	ALB	Albania	78.5	14.7	10.1	13637.0	2.80	11.61
2	DZA	Algeria	76.7	14.5	8.0	11302.0	3.45	3.08
3	AGO	Angola	60.8	11.8	5.2	6361.0	1.03	3.34
4	ATG	Antigua and Barbuda	76.9	12.5	9.3	20139.0	7.66	13.40
...
160	VUT	Vanuatu	70.3	11.4	6.8	3091.0	1.03	7.34
161	VEN	Venezuela	72.1	12.8	10.3	10380.0	5.62	7.90
162	VNM	Viet Nam	75.3	12.7	8.2	7051.0	1.74	12.78
163	YEM	Yemen	66.1	8.7	3.2	1564.0	1.07	1.11
164	ZMB	Zambia	63.5	11.4	7.1	3366.0	0.65	3.53

165 rows x 8 columns

IMPORTATION DES BASES



```
import pandas as pd
df_2018 = pd.read_excel("base finale SDI-2018 (2).xlsx")
df_2018.head(165)
```

	country	life expectancy 2019	expected years of schooling 2019	means years of schooling 2019	GNI per capita 2019	CO2 emission 2019	material foot per cap(tonnes) 2019
0	Afghanistan	64.8	10.2	3.9	2229.0	0.47	1.20
1	Albania	78.6	14.7	10.1	13998.0	2.80	11.61
2	Algeria	76.9	14.6	8.0	11174.0	3.45	3.08
3	Angola	61.1	11.8	5.2	6104.0	1.03	3.34
4	Antigua and Barbuda	77.0	12.8	9.3	20895.0	7.66	13.40
...
160	Vanuatu	70.5	11.7	7.1	3105.0	1.03	7.34
161	Venezuela	72.1	12.8	10.3	7045.0	5.62	7.90
162	Viet Nam	75.4	12.7	8.3	7433.0	1.74	12.78
163	Yemen	66.1	8.8	3.2	1594.0	1.07	1.11
164	Zambia	63.9	11.5	7.2	3326.0	0.65	3.53



nettoyage des colonnes

```
# Définir une fonction pour nettoyer les colonnes
def nettoyer_colonnes(df):
    dernieres_colonnes = df.columns[-2:]
    for col in dernieres_colonnes:
        valeurs = []
        index_manquants = []
        moyenne = df[col].mean() # Calculer la moyenne de la colonne
        for i, value in enumerate(df[col]):
            if pd.isna(value): # Vérifier si la valeur est manquante (NaN)
                valeurs.append(moyenne) # Ajouter la moyenne à la liste des valeurs
                index_manquants.append(i) # Ajouter l'index à la liste des index manquants
            else:
                valeurs.append(value) # Ajouter la valeur d'origine si elle n'est pas manquante
        for i, index in enumerate(index_manquants):
            df.loc[index, col] = valeurs[i] # Utiliser loc pour modifier par index

    return df
```

```
# Appliquer la fonction aux DataFrames
df_2018_remplace = nettoyer_colonnes(df_2018)
df_2019_remplace = nettoyer_colonnes(df_2019)
```



NETTOYER LES DEUX
DERNIÈRES COLONNES DE
CHAQUE BASE EN REMPLAÇANT
LE VIDE PAR LA MOYENNE
DE LA COLONNE.



renommage des colonnes

```
def renommer_colonnes(df, année):  
    colonnes_renommées = {col: col[:-4]+'_'+str(année) for col in df.columns [-2:]}  
    df.rename (columns=colonnes_renommées, inplace=True)  
    return df  
df_2018 = renommer_colonnes (df_2018, 2018)  
df_2019 = renommer_colonnes (df_2019, 2019)
```

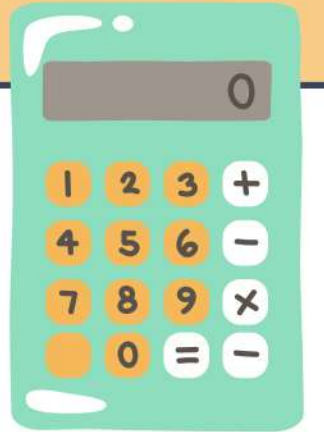


	Expect years of schooling 2018	Mean years of schooling 2018 \
0	10.1	3.9
1	14.7	10.1
2	14.5	8.0
3	11.8	5.2
4	12.5	9.3
..
161	12.8	10.3
162	12.7	8.2
163	8.7	3.2
164	11.4	7.1
165	10.5	8.4

NOMMER CES COLONNES
AVEC LEURS ABREVIATIONS
SUIVI DE 2018 OU 2019
RESPECTIVEMENT



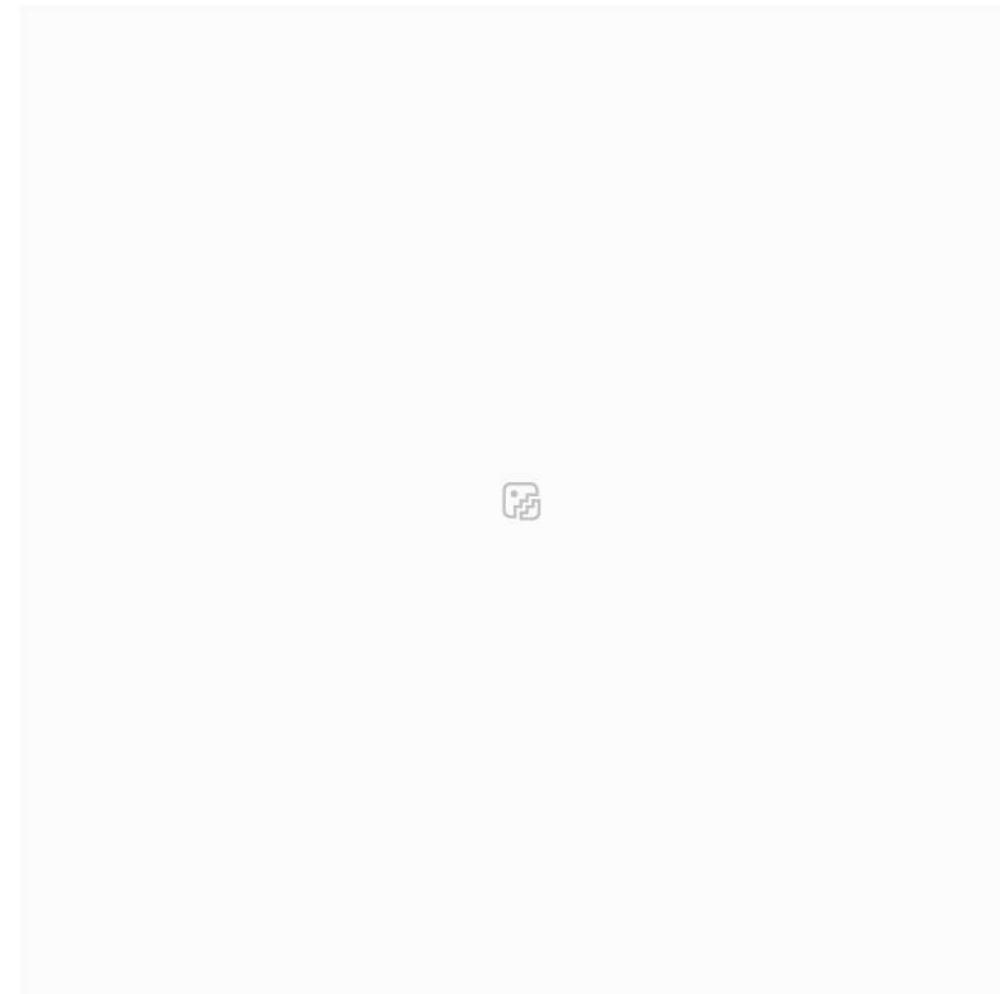
calcul de ID


$$ID_{2018} = (3 ** 0.5) * IEV_{2018} * IR_{2018} * IE_{2018}$$

ON A CALCULÉ ID
A TRAVERS IE,IEV,IR



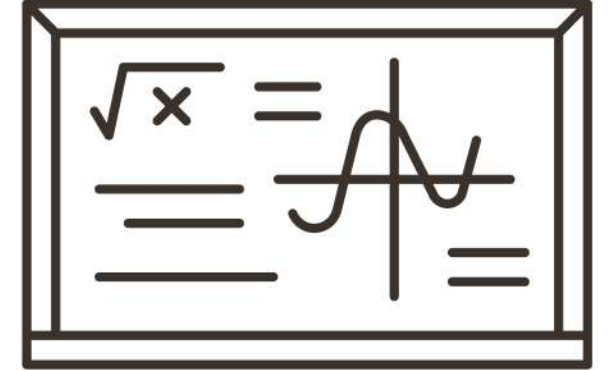
calcul de



ON A CALCULÉ ID
A TRAVERS IE,IEV,IR



calcul de IIE



```
# Fonction pour calculer l'indice de l'impact écologique (IIE)
def calcul_IIE(DM):
    return np.where(DM <= 4, 1 + ((np.exp(DM) - np.exp(1)) / (np.exp(4) - np.exp(1))), DM - 2)
```

ON A CALCULÉ IIE
A TRAVERS DM

calcul de SDI

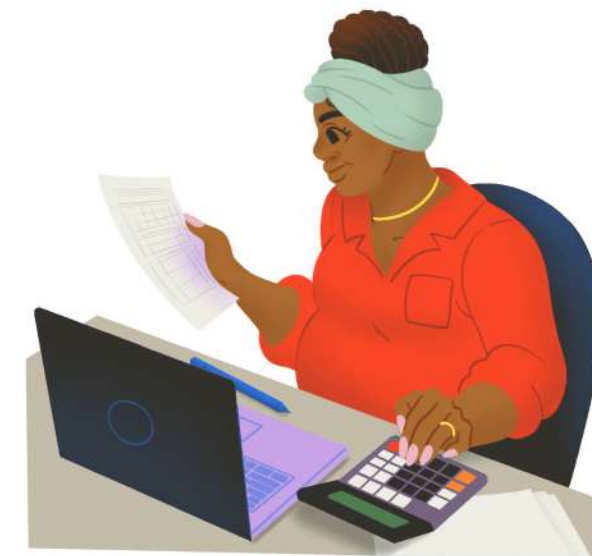
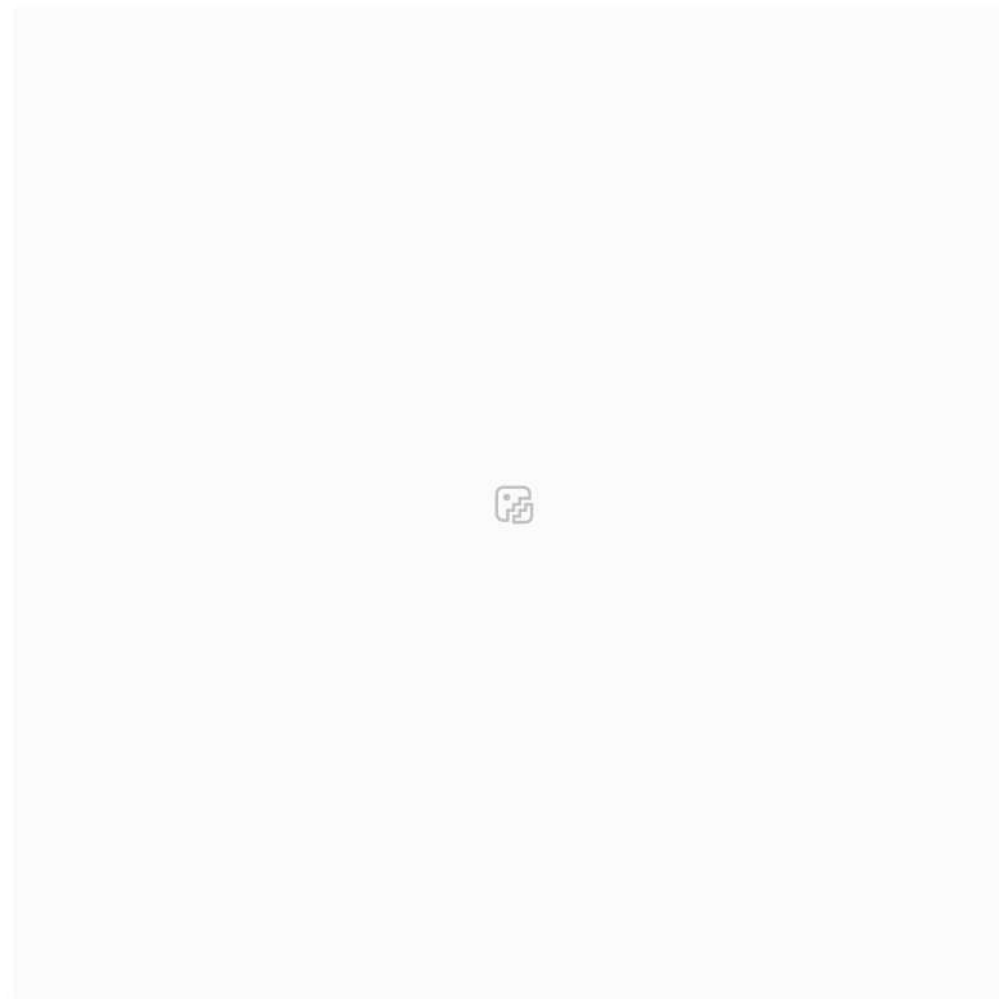
```
# Fonction pour calculer l'Indice de Développement Durable (SDI)
def calcul_SDI(ID, IIE):
    return ID / IIE
```



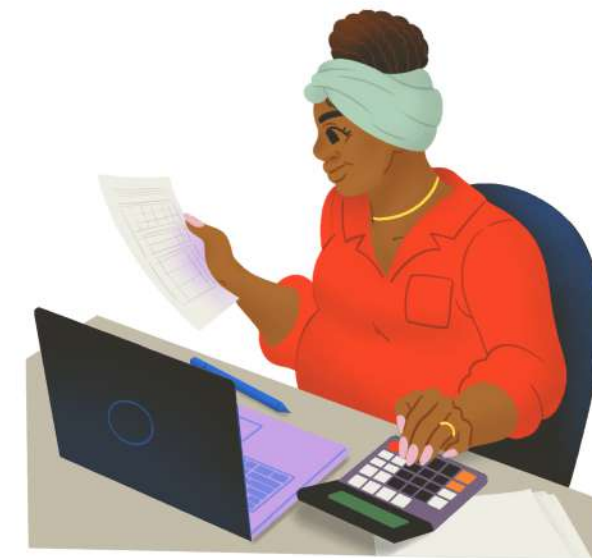
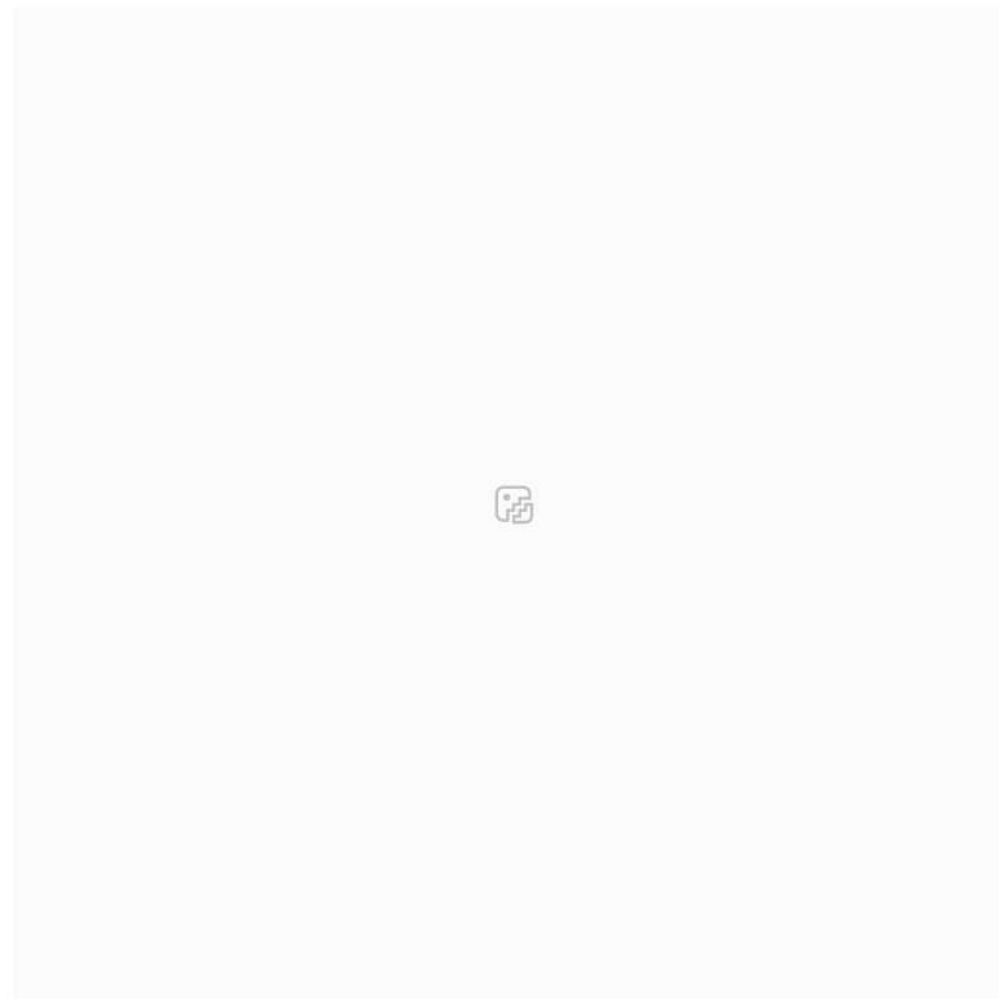
ET ENFIN ON VA
CALCULER SDI
A TRAVERS ID ET IIE,
QU'ON A DEJA CALCULÉ



calcul de SDI



calcul de SDI



creation d'une nouvelle dataframe



```
# Création de la nouvelle DataFrame
nouvelle_dataframe = pd.DataFrame()

# Ajout des colonnes de code de pays
nouvelle_dataframe['Code Pays'] = df_2018['iso'] # Utilisation de la colonne 'iso' comme code de pays

# Ajout des colonnes d'indice de développement socio-économique pour les années 2018 et 2019 (DI-2018 et DI-2019)
nouvelle_dataframe['ID-2018'] = df_2018_remplace['SDI_2018']
nouvelle_dataframe['ID-2019'] = df_2019_remplace['SDI_2019']

# Ajout des colonnes d'indice de l'impact écologique pour les années 2018 et 2019 (IIE-2018 et IIE-2019)
nouvelle_dataframe['IIE-2018'] = df_2018_remplace['IIE_2018']
nouvelle_dataframe['IIE-2019'] = df_2019_remplace['IIE_2019']

# Ajout des colonnes d'indice du développement durable pour les années 2018 et 2019 (SDI-2018 et SDI-2019)
nouvelle_dataframe['SDI-2018'] = df_2018_remplace['SDI_2018']
nouvelle_dataframe['SDI-2019'] = df_2019_remplace['SDI_2019']

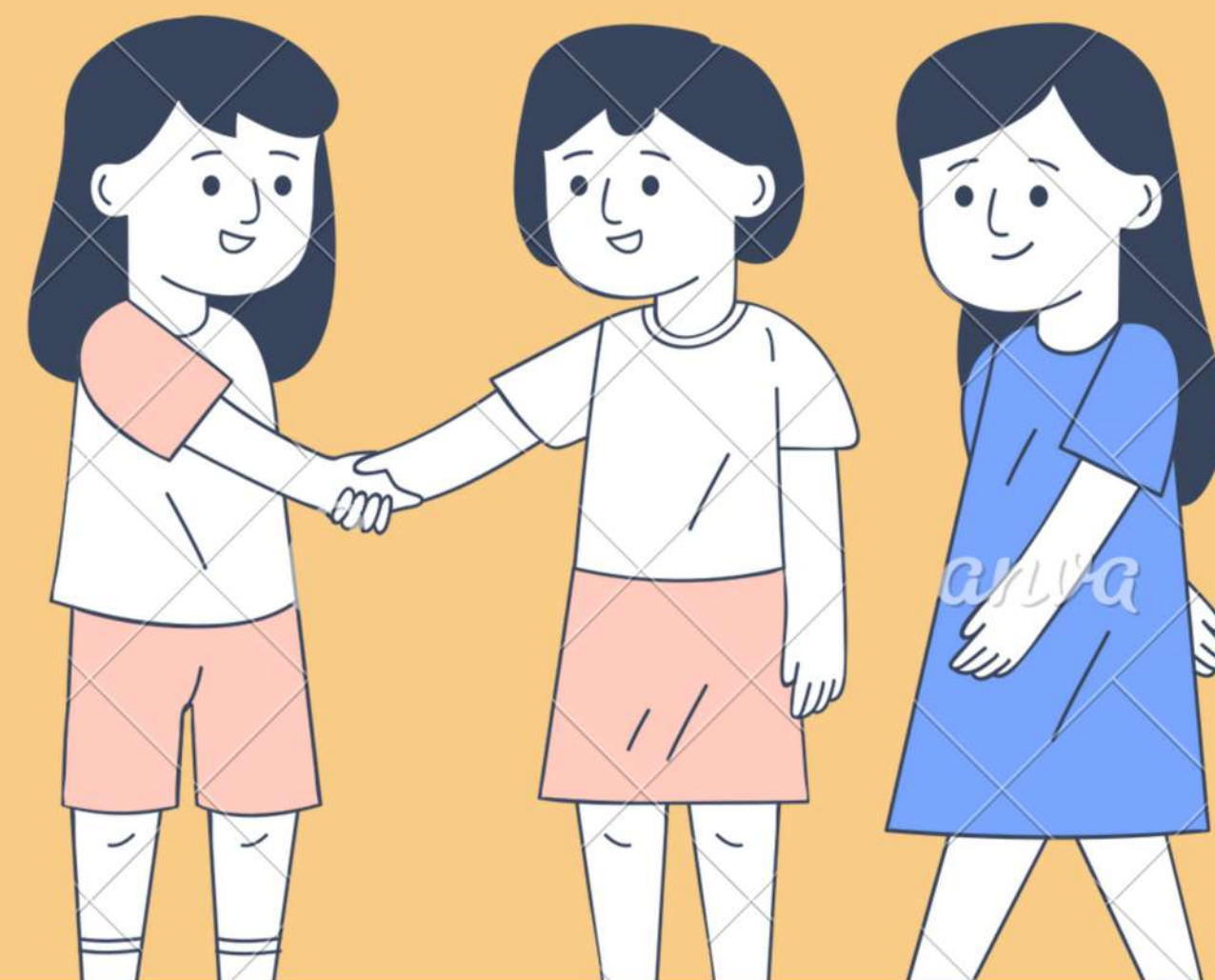
# Affichage de la nouvelle DataFrame
print(nouvelle_dataframe)
```

ET ENFIN ON VA
CALCULER SDI
A TRAVERS ID ET IIE,
QU'ON A DÉJÀ CALCULÉ



STREAM

3



classement des valeurs

```
# Créer une fonction pour classer les valeurs en trois catégories
```

```
def classer_categorie(valeur):  
    if valeur > seuil_elevé:  
        return 'Élevé'  
    elif valeur > seuil_moyen:  
        return 'Moyen'  
    else:  
        return 'Faible'
```

```
# Définir les seuils pour les catégories
```

```
seuil_elevé = nouvelle_dataframe['SDI-2019'].quantile(0.75)  
seuil_moyen = nouvelle_dataframe['SDI-2019'].quantile(0.5)
```

```
# Appliquer la fonction aux colonnes SDI-2019 et ID-2019 pour créer de nouvelles colonnes de catégorie
```

```
nouvelle_dataframe['Catégorie SDI-2019'] = nouvelle_dataframe['SDI-2019'].apply(classer_categorie)  
nouvelle_dataframe['Catégorie ID-2019'] = nouvelle_dataframe['ID-2019'].apply(classer_categorie)
```

```
# Affichage de la nouvelle DataFrame avec les colonnes de catégorie
```

```
print(nouvelle_dataframe)
```



CLASSER EN TROIS
CATÉGORIES: INDICATEUR
ÉLEVÉ, MOYEN OU FAIBLE.

	Code Pays	ID-2018	ID-2019	IIE-2018	IIE-2019	SDI-2018	SDI-2019
0	AFG	0.585878	0.594853	0.971961	0.971961	0.585878	0.594853
1	ALB	2.029973	2.044251	1.061494	1.061494	2.029973	2.044251
2	DZA	1.803500	1.812993	1.000825	1.000825	1.803500	1.812993
3	AGO	0.869718	0.867413	0.981958	0.981958	0.869718	0.867413
4	ATG	1.425032	1.455679	1.400376	1.400376	1.425032	1.455679
..
161	VEN	1.543963	1.415069	1.101278	1.101278	1.543963	1.415069
162	VNM	1.437067	1.465274	1.031372	1.031372	1.437067	1.465274
163	YEM	0.455704	0.462511	0.974673	0.974673	0.455704	0.462511
164	ZMB	0.870041	0.884720	0.978505	0.978505	0.870041	0.884720
165	ZWE	0.814394	0.820421	0.983732	0.983732	0.814394	0.820421

	Catégorie SDI-2019	Catégorie ID-2019
0	Faible	Faible
1	Élevé	Élevé
2	Élevé	Élevé
3	Faible	Faible
4	Moyen	Moyen
..
161	Moyen	Moyen
162	Moyen	Moyen
163	Faible	Faible
164	Faible	Faible
165	Faible	Faible

[166 rows x 9 columns]


```
# Créer des ensembles pour chaque catégorie de l'indice de développement durable (SDI-2019) et de l'indice de développement
sdi_elevé = set(nouvelle_dataframe[nouvelle_dataframe['Catégorie SDI-2019'] == 'Élevé']['Code Pays'])
sdi_moyen = set(nouvelle_dataframe[nouvelle_dataframe['Catégorie SDI-2019'] == 'Moyen']['Code Pays'])
sdi_faible = set(nouvelle_dataframe[nouvelle_dataframe['Catégorie SDI-2019'] == 'Faible']['Code Pays'])

id_elevé = set(nouvelle_dataframe[nouvelle_dataframe['Catégorie ID-2019'] == 'Élevé']['Code Pays'])
id_moyen = set(nouvelle_dataframe[nouvelle_dataframe['Catégorie ID-2019'] == 'Moyen']['Code Pays'])
id_faible = set(nouvelle_dataframe[nouvelle_dataframe['Catégorie ID-2019'] == 'Faible']['Code Pays'])

# Comparer les ensembles pour chaque catégorie
pays_sdi_id_elevé = sdi_elevé.intersection(id_elevé)
pays_sdi_id_moyen = sdi_moyen.intersection(id_moyen)
pays_sdi_id_faible = sdi_faible.intersection(id_faible)

# Afficher les résultats
print("Pays avec un indice de développement durable (SDI-2019) élevé et un indice de développement (ID-2019) élevé :")
print(pays_sdi_id_elevé)

print("\nPays avec un indice de développement durable (SDI-2019) moyen et un indice de développement (ID-2019) moyen :")
print(pays_sdi_id_moyen)

print("\nPays avec un indice de développement durable (SDI-2019) faible et un indice de développement (ID-2019) faible :")
print(pays_sdi_id_faible)
```


Pays avec un indice de développement durable (SDI-2019) élevé et un indice de développement (ID-2019) élevé :

{'GEO', 'BRB', 'BRA', 'MLT', 'HUN', 'FRA', 'FJI', 'BIH', 'LKA', 'AZE', 'THA', 'PER', 'JAM', 'MDA', 'CRI', 'BGR', 'MKD', 'HRV', 'PRT', 'DZA', 'OMN', 'KRL', 'DOM', 'ROU', 'RUS', 'ARM', 'CHL', 'CUB', 'MEX', 'PRY', 'PHL', 'TUN', 'COL', 'ECU', 'TUR', 'ARG', 'UKR', 'PAN', 'MUS', 'ITA', 'JOR', 'ALB'}

Pays avec un indice de développement durable (SDI-2019) moyen et un indice de développement (ID-2019) moyen :

{'URY', 'SRB', 'UZB', 'IRN', 'SLV', 'MNE', 'GRC', 'MDV', 'POL', 'SUR', 'IDN', 'DNK', 'GAB', 'IRL', 'SVN', 'WSM', 'LBN', 'NIC', 'GBR', 'IRQ', 'LBY', 'CZE', 'VEN', 'ATG', 'GTM', 'LVA', 'ESP', 'TJK', 'ZAF', 'DEU', 'NZL', 'MNG', 'BLZ', 'VNM', 'BOL', 'CPV', 'KGZ', 'MAR', 'MYS', 'EGY', 'ISR'}

Pays avec un indice de développement durable (SDI-2019) faible et un indice de développement (ID-2019) faible :

{'CYP', 'BGD', 'MDG', 'BHS', 'BWA', 'HTI', 'HND', 'LTU', 'GIN', 'SEN', 'CAF', 'MWI', 'NPL', 'TTO', 'NAM', 'COG', 'KWT', 'VUT', 'BRN', 'SAU', 'MMR', 'PNG', 'GHA', 'CAN', 'SWZ', 'KHM', 'MLI', 'CHN', 'PAK', 'STP', 'CHE', 'CIV', 'MRT', 'QAT', 'USA', 'AUT', 'UGA', 'CMR', 'LAO', 'SWE', 'FIN', 'IND', 'AFG', 'TCD', 'ETH', 'NGA', 'NER', 'SLE', 'KAZ', 'GMB', 'TGO', 'AGO', 'BFA', 'SYC', 'DJI', 'ERI', 'SYR', 'BHR', 'BTN', 'COD', 'LSO', 'EST', 'BEN', 'NOR', 'KOR', 'BDI', 'TKM', 'ZMB', 'LBR', 'MOZ', 'ISL', 'AUS', 'ARE', 'SVK', 'YEM', 'NLD', 'JPN', 'TZA', 'KEN', 'BEL', 'SGP', 'RWA', 'ZWE'}

DETERMINER L'OPPORTUNITÉ D'AMELIORATION



DETERMINER L'
OPPORTUNITÉ D'
AMELIORATION

```
# Créer une fonction pour déterminer l'opportunité d'amélioration socio-économique et durable
def determiner_opportunite(row):
    if row['Catégorie SDI-2019'] == 'Faible' and row['Catégorie ID-2019'] == 'Faible':
        return 'Pays à potentiel d'amélioration durable'
    elif row['Catégorie SDI-2019'] == 'Faible' and row['Catégorie ID-2019'] != 'Faible':
        return 'Pays à potentiel d'amélioration socio-économique'
    elif row['Catégorie SDI-2019'] != 'Faible' and row['Catégorie ID-2019'] == 'Faible':
        return 'Pays à potentiel d'amélioration en développement durable'
    else:
        return 'Pays à la pointe du développement durable'

# Appliquer la fonction pour créer la nouvelle colonne
nouvelle_dataframe['Opportunité d'amélioration socio-économique et durable'] = nouvelle_dataframe.apply(determiner_opportunite, axis=1)

# Afficher la DataFrame avec la nouvelle colonne
print(nouvelle_dataframe)
```

Opportunité d'amélioration socio-économique et durable

0	Pays à potentiel d'amélioration durable
1	Pays à la pointe du développement durable
2	Pays à la pointe du développement durable
3	Pays à potentiel d'amélioration durable
4	Pays à la pointe du développement durable
..	...
161	Pays à la pointe du développement durable
162	Pays à la pointe du développement durable
163	Pays à potentiel d'amélioration durable
164	Pays à potentiel d'amélioration durable
165	Pays à potentiel d'amélioration durable

[166 rows x 10 columns]

SDI EVOLUTION RATE

```
# Créer la colonne "SDI evolution rate"
nouvelle_dataframe['SDI evolution rate'] = ((nouvelle_dataframe['SDI-2019'] - nouvelle_dataframe['SDI-2018']) / nouvelle_dataframe['SDI-2018'])

# Afficher la DataFrame avec la nouvelle colonne
print(nouvelle_dataframe)
```

	Opportunité d'amélioration socio-économique et durable	SDI evolution rate
0	Pays à potentiel d'amélioration durable	1.531873
1	Pays à la pointe du développement durable	0.703401
2	Pays à la pointe du développement durable	0.526354
3	Pays à potentiel d'amélioration durable	-0.265111
4	Pays à la pointe du développement durable	2.150609
..
161	Pays à la pointe du développement durable	-8.348208
162	Pays à la pointe du développement durable	1.962771
163	Pays à potentiel d'amélioration durable	1.493907
164	Pays à potentiel d'amélioration durable	1.687219
165	Pays à potentiel d'amélioration durable	0.740049

[166 rows x 11 columns]

Thank You

Do you have any
questions of us?

