My Document

Table of contents


```
import pyspark
from pyspark import SparkContext
from pyspark.sql import SparkSession
#from pyspark.sql.functions import *
from pyspark.sql import functions as func
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.io as pio
pio.renderers.default = "plotly_mimetype+notebook_connected"

sc = SparkContext.getOrCreate()
spark = SparkSession.builder.appName("Python Spark").getOrCreate()
```

```
df_annual_co2 = spark.read\
    .option("header", "true")\
```

```
    .csv('Data/annual-co2-emissions-per-country.csv', inferSchema='true')

df_annual_death = spark.read\
    .option("header", "true")\
    .csv('Data/annual-number-of-deaths-by-world-region.csv', inferSchema='true')

df_pib = spark.read\
    .option("header", "true")\
    .csv('Data/gdp(pib)-per-capita-maddison-2020.csv', inferSchema='true')

df_population = spark.read\
    .option("header", "true")\
    .csv('Data/population-world.csv', inferSchema='true')

df_nb_wo_clean_cooking_fuel = spark.read\
    .option("header", "true")\
    .csv('Data/number-without-clean-cooking-fuel.csv', inferSchema='true')

df_pop_growth = spark.read\
    .option("header", "true")\
    .csv('Data/population-growth-rate-vs-median-age.csv', inferSchema='true')

df_co2_target = spark.read\
    .option("header", "true")\
    .csv('Data/net-zero-target-set.csv', inferSchema='true')

df_continent = spark.read\
    .option("header", "true")\
    .csv('Data/countryContinent.csv', inferSchema='true')
```

```
df_annual_co2.printSchema()
df_annual_death.printSchema()
df_pib.printSchema()
df_population.printSchema()
df_nb_wo_clean_cooking_fuel.printSchema()
df_pop_growth.printSchema()
df_co2_target.printSchema()
df_continent.printSchema()
```

```
df_annual_co2.show(1)
df_annual_death.show(1)
df_pib.show(1)
df_population.show(1)
df_nb_wo_clean_cooking_fuel.show(1)
df_pop_growth.show(1)
df_co2_target.show(1)
df_continent.show(1)
```

1 Création des dataframes

```
#Join global en inner donc perte de données (seulement jusqu'à 2018)
df_all = df_annual_co2.join(df_annual_death, (df_annual_co2['Code'] == df_annual_death['Co
            & (df_annual_co2['Year'] == df_annual_death['Year']), 'inner')\
         .select(df_annual_co2['Entity'].alias('Country'), df_annual_co2['Code'], df_annua
            df_annual_co2['Annual CO2 emissions'].alias('Annual_CO2_emissions'),\
            df_annual_death['Deaths - sex: all - age: all - variant: estimates'].alias('D

df_all = df_all.join(df_pib, (df_all['Code'] == df_pib['Code']) \
            & (df_all['Year'] == df_pib['Year']), 'inner')\
         .select(df_all['Country'], df_all['Code'], df_all['Year'], df_all['Annual_CO2_emi
            df_all['Deaths'],df_pib['GDP per capita'].alias('PIB_per_capita'))

df_all = df_all.withColumn('PIB_per_capita', func.round(df_all['PIB_per_capita'],2)) # vir

df_all = df_all.join(df_population, (df_all['Code'] == df_population['Code']) \
            & (df_all['Year'] == df_population['Year']), 'inner')\
         .select(df_all['Country'], df_all['Code'], df_all['Year'], df_all['Annual_CO2_emi
            df_all['Deaths'],df_all['PIB_per_capita'], df_population['Population - Sex: a

df_all = df_all.join(df_pop_growth, (df_all['Code'] == df_pop_growth['Code']) \
            & (df_all['Year'] == df_pop_growth['Year']), 'inner')\
         .select(df_all['Country'], df_all['Code'], df_all['Year'], df_all['Annual_CO2_emi
            df_all['Deaths'],df_all['PIB_per_capita'], df_all['Population'],\
            df_pop_growth['Estimates, 1950 - 2020: Annually interpolated demographic indi
            .alias('Rate_of_natural_population_increase'))

df_all = df_all.withColumn('Rate_of_natural_population_increase', func.round(df_all['Rate_
```

```
df_all = df_all.join(df_co2_target, (df_all['Code'] == df_co2_target['Code']) \
            #& (df_all['Year'] == df_co2_target['Year'])
            , 'left')\
         .select(df_all['Country'], df_all['Code'], df_all['Year'], df_all['Annual_CO2_emi
            df_all['Deaths'],df_all['PIB_per_capita'], df_all['Population'], df_all['Rate
            df_co2_target['Year'].alias('Net_zero_target'))

df_all = df_all.join(df_continent, (df_all['Code'] == df_continent['code_3']) #supprime le
            #& (df_all['Year'] == df_co2_target['Year'])
            , 'inner')\
         .select(df_all['Country'], df_all['Code'], df_all['Year'], df_all['Annual_CO2_emi
            df_all['Deaths'],df_all['PIB_per_capita'], df_all['Population'], df_all['Rate
            df_all['Net_zero_target'], df_continent['continent'], df_continent['sub_regio




##############################################
#Join avec données jusqu'à 2020 : emmission CO2, population, continent


df_pop = df_annual_co2.join(df_population, (df_annual_co2['Code'] == df_population['Code']
            & (df_annual_co2['Year'] == df_population['Year']), 'outer')\
         .select(df_annual_co2['Entity'].alias('Country'), df_annual_co2['Code'], df_annua
                df_annual_co2['Annual CO2 emissions'].alias('Annual_CO2_emissions'),\
                df_population['Population - Sex: all - Age: all - Variant: estimates'].al

df_pop = df_pop.join(df_continent, (df_pop['Code'] == df_continent['code_3']) #supprime le
            #& (df_all['Year'] == df_co2_target['Year'])
            , 'outer')\
         .select(df_pop['Country'], df_pop['Code'], df_pop['Year'], df_pop['Annual_CO2_emi
            df_pop['Population'], df_continent['continent'], df_continent['sub_region'] )

df_pop = df_pop.where(df_all.Year>1949)


##############################################

df_world_cO2 = df_annual_co2.filter((df_annual_co2.Entity == 'World') | (df_annual_co2.Ent
                                    | (df_annual_co2.Entity =='United States') | (df_annua
                                    | (df_annual_co2.Entity =='Russia')).select(df_annual_
```

```
                                        df_annual_co2['Code'], df_annual_co2['Year'], df_ann
```

1.1 Data cleaning

```
#Suppression de la ligne avec Code = DMA (contient une valeur null)
cond = (df_all.Code == 'DMA')
df_all = df_all.filter(~cond)

#Pareil pour 'World'
cond = (df_all.Code == 'OWID_WRL')
df_all = df_all.filter(~cond)
```

```
#On vérifie que les 'pays' Dominica et World ont bien disparu
df_all.filter(df_all.Code =='DMA').show(truncate=False)
df_all.filter(df_all.Code == 'OWID_WRL').show()#.select(countDistinct(df_all.Year)).show()
```

1.2 Visu en interne : data

```
#nombre total de pays répértoriés
df_all.select(func.countDistinct('Country')).show(truncate=False)
```

```
#rechercher cb de pays sur 163 ont l'info pour une année donnée
annee = 2000
df_all.filter(df_all.Year==annee).select(func.count('Year')).collect()[0][0]

#à partir de l'année
#1950 : on a 117 pays avec toutes les infos
#1960 : on a 141 pays avec toutes les infos
#1970 : on a 144 pays avec toutes les infos
#1980 : on a 159 pays avec toutes les infos
#1990 : on a 163 pays avec toutes les infos
#2000 : tous les pays : 164
```

```
#visualisation de l'évolution du nombre de pays pour lesquels nous avons toutes les inform
fig = plt.figure(figsize=(4,3), dpi = 150)
annees = []
pays_w_infos = []
```

```
for annee in range(1950,2010,10):
  info = df_all.filter(df_all.Year==annee).select(func.count('Year')).collect()[0][0]
  annees.append(annee)
  pays_w_infos.append(info)

new_lst = [list(x) for x in zip(annees, pays_w_infos)]

columns = ["year","pays_w_infos"]
df_intro = spark.createDataFrame(data=new_lst, schema = columns)
#df_intro.printSchema()
#df_intro.show(truncate=False)

df_intro_pd = df_intro.toPandas()
sns.set_style("darkgrid")
sns.lineplot(data=df_intro_pd, x="year", y="pays_w_infos").set(title='Number of countries
                                                               ylabel='Number of countries with
```

```
#Vérification de la suppression de la l'unique ligne DMA en 2014
df_pop_growth.filter((df_pop_growth.Year=='2014')&(df_pop_growth.Code=='DMA')).show(trunca
```

```
#Aperçu France
df_all.filter(df_all.Code == 'FRA').show(5, truncate=False)
```

```
#nombre de pays dont on a la donnée objectif 0 emissions
df_all.filter(df_all.Net_zero_target.isNotNull()).select(func.countDistinct("Country")).sh
```

Essayer de selectionner seulement les 50 premiers pays polluants pour la visu car bcp trop ne polluent pas et prennent de la place sur la visu pour rien

```
df = df_all.filter(df_all.Year == 2018).orderBy(func.desc('Annual_CO2_emissions')).select(
df[:5]
```

1.3 Transformation df_all vers pandas pour visu

```
df_all_pd = df_all.toPandas() #df plein
#df_all_2000_pd = df_all_2000.toPandas() #df test avec année min selectionnée au dessus
```

1.4 Visu 1 : World C02 emission

```
df = df_world_c02.toPandas()
fig = px.line(df, x="Year", y='Annual_CO2_emissions', color='Country', \
              title='Carbon dioxide (CO2) emissions from fossil, fuels and industry',\
             labels={'Annual_CO2_emissions':'CO2 emission (billions of t)'})

fig.update_xaxes(range=[1820, 2022])
fig.show()
```

```
Unable to display output for mime type(s): text/html
```

```
Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html
```

```
df_all.columns
```

```
['Country',
 'Code',
 'Year',
 'Annual_CO2_emissions',
 'Deaths',
 'PIB_per_capita',
 'Population',
 'Rate_of_natural_population_increase',
 'Net_zero_target',
 'continent',
 'sub_region']
```

```
df_all_pd[df_all_pd['Country'] == 'France']
```

	Country	Code	Year	Annual_CO2_emissions	Deaths	PIB_per_capita	Population	Rate_of_
3290	France	FRA	1950	202428862	533995	8266.00	41842356	0.66
3291	France	FRA	1951	228915868	566424	8705.00	42196660	0.65
3292	France	FRA	1952	227188014	525413	8869.00	42542396	0.64
3293	France	FRA	1953	214738732	558275	9060.00	42883300	0.63
3294	France	FRA	1954	226568034	519916	9428.00	43225644	0.63
...	...	...	...	...	...	...	...	...

	Country	Code	Year	Annual_CO2_emissions	Deaths	PIB_per_capita	Population	Rate_of_
3354	France	FRA	2014	325885736	546921	36527.00	63588496	0.31
3355	France	FRA	2015	329910376	582205	36827.00	63809768	0.28
3356	France	FRA	2016	333578746	580585	37124.00	63989320	0.25
3357	France	FRA	2017	336995589	594684	37895.00	64144092	0.22
3358	France	FRA	2018	322371888	596919	38515.92	64277812	0.19

```
df = df_pop.toPandas()
fig = px.bar(df, x='Year', y='Population', color='continent', hover_data=['Country'],\
             title = "Evolution de la population mondiale")


fig.show()
```

```
Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html
```

```
df2018 = df_all.filter(df_all.Year =='2018').toPandas()

fig = px.histogram(df2018, x = 'Annual_CO2_emissions', color = 'continent', marginal = 'ru
             hover_data = df2018.columns, \
             title = "Distribution of CO2 emission rates in 2018", labels={'Annual_CO2_emis
fig.show()
```

```
[Stage 1925:> (0 + 0) / 1][Stage 1926:> (0 + 0) / 1][Stage 1927:> (0 + 0) / 1]
```

```
Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html
```

```


#ajouter une colonne Emmission CO2 par habitant en pyspark
df_all = df_all.withColumn("CO2_p_hab", func.lit(df_all['Annual_CO2_emissions']/df_all['Po
df2018 = df_all.filter(df_all.Year =='2018').toPandas()
#df2018.sort_values("CO2_hab", ascending = False)
df1 = df2018.copy()
df1 = df1.loc[df1["Population"] >=5000000]
df1.sort_values("CO2_hab", ascending = False)
len(df1)
```

```
#ajouter une colonne Emmission CO2 par habitant en pyspark
df_all = df_all.withColumn("CO2_p_hab", func.lit(df_all['Annual_CO2_emissions']/df_all['Po
#filtre sur 2018 avec un minimum de population pour ne pas rendre l'échelle illisible
min_pop = 5000000
df2018 = df_all.filter((df_all.Year =='2018')&(df_all.Population >= min_pop)).toPandas()

fig = px.sunburst(df2018, path=['continent', 'Country'], color='CO2_p_hab',
           hover_data =df2018.columns, values='Population',
           title = 'CO2 emissions per capita, ranking by population')

fig.show()
```

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

```
px.choropleth(df1, locations="Code", color="Annual_CO2_emissions", title = "Annual CO2 emi
             hover_data=df2018.columns)
```

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

```
px.choropleth(df1, locations="Code", color="CO2_hab", title = "CO2 emission per capita (co
             hover_data=df2018.columns)
```

Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html

```
df = df_all.toPandas()

fig = px.scatter(df, x='CO2_p_hab', y='PIB_per_capita', log_x=True, color='continent',size
                size_max=45, animation_frame='Year', animation_group='Country', hover_data
                 hover_name='Country',
                 title="Annual evolution of CO2 emissions by country according to the numb
                )

#range_y=[0,10000000000],range_x=[10000,15000000],
```

```
fig.show()
```

```
Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html
```

```
print(min(df_all_pd['Annual_CO2_emissions']),', ', max(df_all_pd['Annual_CO2_emissions']))
```

```
3664 ,  10289989525
```

1.5 THEODORE

```
temp = df_all.filter(df_all.Year==2000).groupBy('continent').sum('Annual_CO2_emissions','P

lst_all = []
for annee in range(1950,2018,1):
  lst_temp = []
  temp = df_all.filter(df_all.Year==annee).groupBy('continent').sum('Annual_CO2_emissions'
  lst_temp.append(annee)
  lst_temp.append(temp[0][1])
  lst_temp.append(temp[1][1])
  lst_temp.append(temp[2][1])
  lst_temp.append(temp[3][1])
  lst_temp.append(temp[4][1])
  lst_all.append(lst_temp)

columns_continent= ['Year','Europe','Afrique','Amérique','Oceanie','Asie']
df_continent = spark.createDataFrame(data=lst_all, schema = columns_continent)
#pivot des données en pyspark
df_continent_pd = df_continent.withColumnRenamed("Amérique","Amerique")
df_continent_pd = df_continent_pd.selectExpr(\
        'Year', 'stack(5, "Europe", Europe, "Afrique", Afrique, "Amerique", Amerique, "Oce
```

```
df = df_continent_pd.toPandas()
fig = px.line(df, x="Year", y='Annual_CO2_Emissions', color='Continent', title='CO2 emissi
```

```
            labels={'Annual_CO2_Emissions':'CO2 emission (billions of t)'})

fig.show()
```

```
Unable to display output for mime type(s): application/vnd.plotly.v1+json, text/html
```

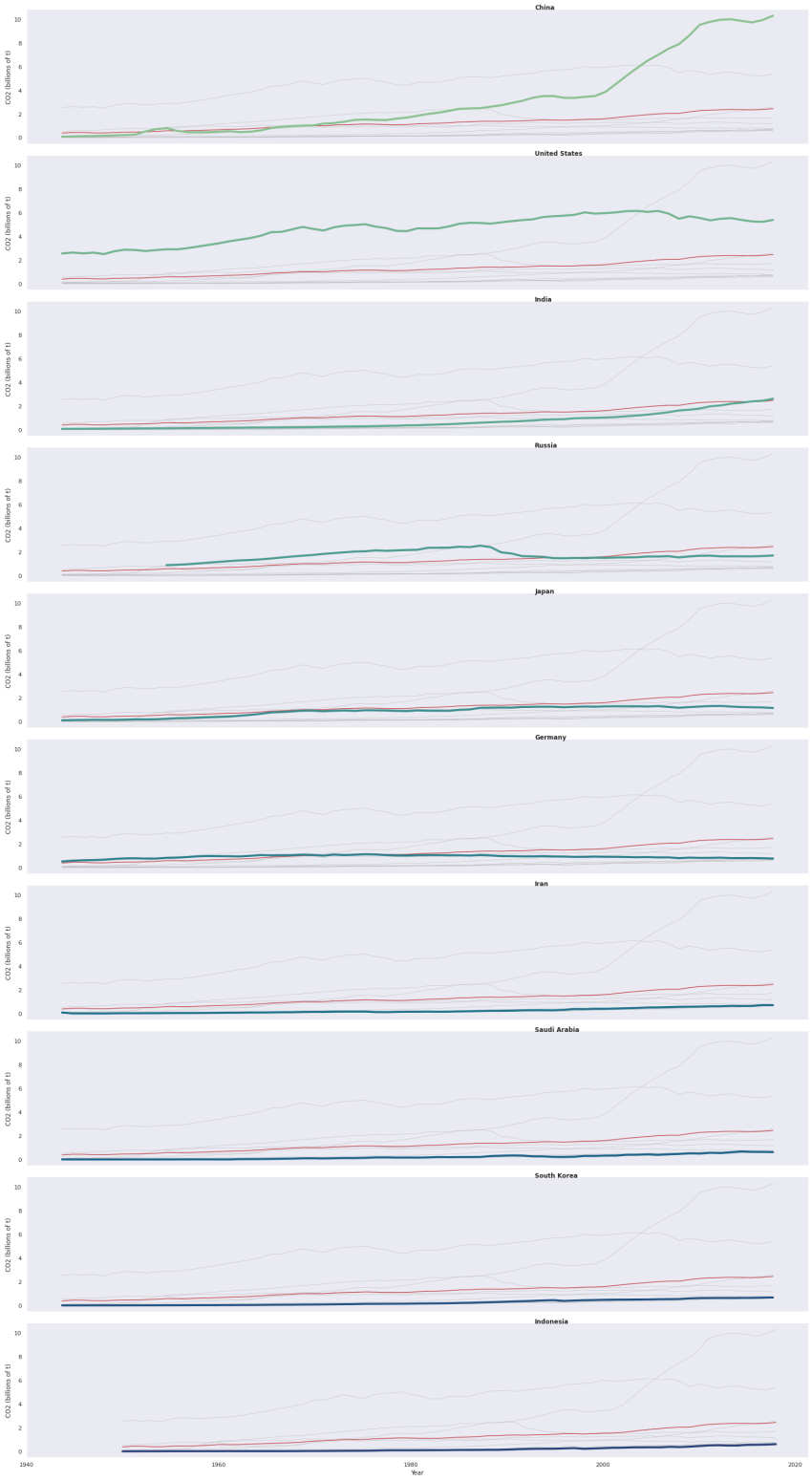
1.6 Visu 2 = Emissions C02 pour les N pays les plus polluants

1.7 Test visu sur nombre de pays limité avant d'appliquer sur all (à supprimer une fois all fonctionnel)

```
n = input("Please enter number n of most polluting countries...:\n") #nbr de pays les plus
n = int(n)

#En pyspark, liste des n pays les plus émetteurs de C02 en 2018
c_list = df_all.filter(df_all.Year == 2018).orderBy(func.desc('Annual_CO2_emissions')).sel
c_list = c_list[:n]

#En pyspark avant conversion pandas, sélectionner seulement les pays compris dans n
df_list_pd = df_all[df_all.Country.isin(c_list)].orderBy(func.desc('Annual_CO2_emissions')


#####Partie plot

df = df_list_pd

sns.set_theme(style="dark")

# Plot each year's time series in its own facet
g = sns.relplot(
    data=df,
    x="Year", y="Annual_CO2_emissions", col="Country", hue="Country",
    kind="line", palette="crest", linewidth=4, zorder=5,
    col_wrap=1, height=4, aspect=5.5, legend=False
)

# Iterate over each subplot to customize further
for country, ax in g.axes_dict.items():
```

```
    # Add the title as an annotation within the plot
    ax.text(.65, 1, country, transform=ax.transAxes, fontweight="bold")

    # Plot l'émission des n pays en fond
    sns.lineplot(
        data=df, x="Year", y="Annual_CO2_emissions", units="Country",
        estimator=None, color="0.7", linewidth=0.5, ax=ax, legend='full'
    )

    #Plot la moyenne entre les n pays sélectionnés
    sns.lineplot(
        data=df, x="Year", y="Annual_CO2_emissions",
        estimator=np.mean, color="r", linewidth=1.5, ax=ax, ci=None
    )


# Reduce the frequency of the x axis ticks
ax.set_xticks(ax.get_xticks()[::2])

# Tweak the supporting aspects of the plot
g.set_titles("")
#g.set_title(f"Emission en CO2 depuis 1950 des {n} pays les plus polluants en 2018 \n \n \
g.set_axis_labels("Year", "CO2 (billions of t)")
g.tight_layout()
```

```
Please enter number n of most polluting countries...:
10
```


China
United States
India
Russia
Japan
Germany
Iran
Saudi Arabia
South Korea
Indonesia
CO2 (billions of t)
0
2
4
6
8
10
1940
1960
1980
2000
2020
Year