

Tarea 1. Índices de Contaminación

Iker Bali Elizalde A01656437

```
In [ ]: # codigo importante

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('dark_background')

data19=pd.read_csv("indice_2019.csv",encoding="latin1",skiprows=8)
data20=pd.read_csv("indice_2020.csv",encoding="latin1",skiprows=8)
data21=pd.read_csv("indice_2021.csv",encoding="latin1",skiprows=8)
data22=pd.read_csv("indice_2022.csv",encoding="latin1",skiprows=8)

data=pd.concat([data19,data20,data21,data22],ignore_index=True)

Zonas=["Noroeste","Noreste","Centro","Suroeste","Sureste"]
Contaminantes = ["ozono", "dióxido de azufre", "dióxido de nitrógeno", "monóxido de carbono", "PM10", "PM25"]

color = ["yellow", "red", "green", "blue", "white"]
rm = 24 * 90

compuestos = ["$O_3$", "$SO_2$", "$NO_2$", "$CO$", "$PM10$", "$PM25$"]
color = ["yellow","red","green","blue","white"]

nomenclatura_dict = {
    "ozono": "$O_3$",
    "dióxido de azufre": "$SO_2$",
    "dióxido de nitrógeno": "$NO_2$",
    "monóxido de carbono": "$CO$",
    "PM10": "$PM10$",
    "PM25": "$PM25$"
}

# Creando arreglos de fecha

newdates = pd.to_datetime(data["Fecha"] +
                           " " +
                           np.mod(data["Hora"]-1,24).astype(str) +
                           ":30:00", format = "%d/%m/%Y-%H:%M:%S")

columnas=data.columns
filas=data.index
```

1. Serie de Tiempo Original

```
In [ ]: plt.figure(figsize=(20, 10))

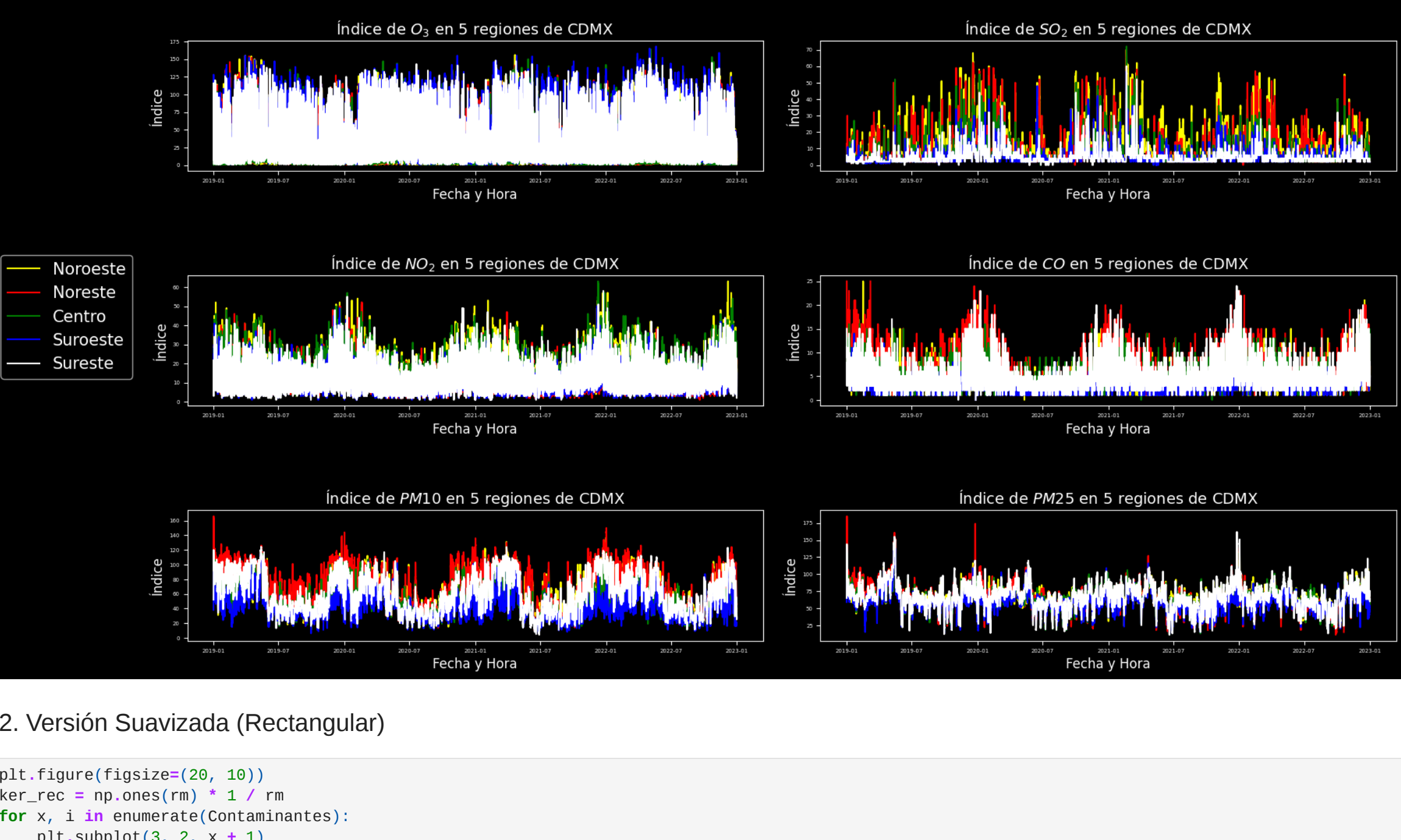
for i in range(1, 7, 1):
    plt.subplot(3, 2, i)
    pos = np.arange(i + 1, 32, 6)
    for j in range(len(Zonas)):
        plt.plot(newdates[pos:], data[columnas[pos[j]]][:~1], label=Zonas[j], color=color[j])
    plt.xlabel("Fecha y Hora", fontsize=12)
    plt.ylabel("Índice", fontsize=12)
    plt.title(f"Índice de {compuestos[i-1]} en 5 regiones de CDMX", fontsize=14)
    plt.tick_params(labelsize=5)

plt.subplots_adjust(wspace=0.1, hspace=0.8)

# Leyenda General

legends = [plt.Line2D([], [], color=color[j], label=Zonas[j]) for j in range(len(pos))]
plt.figlegend(handles=legends, labels=Zonas, loc='center left', fontsize=15)

plt.suptitle("Serie de tiempo original",fontsize=25)
plt.show()
```



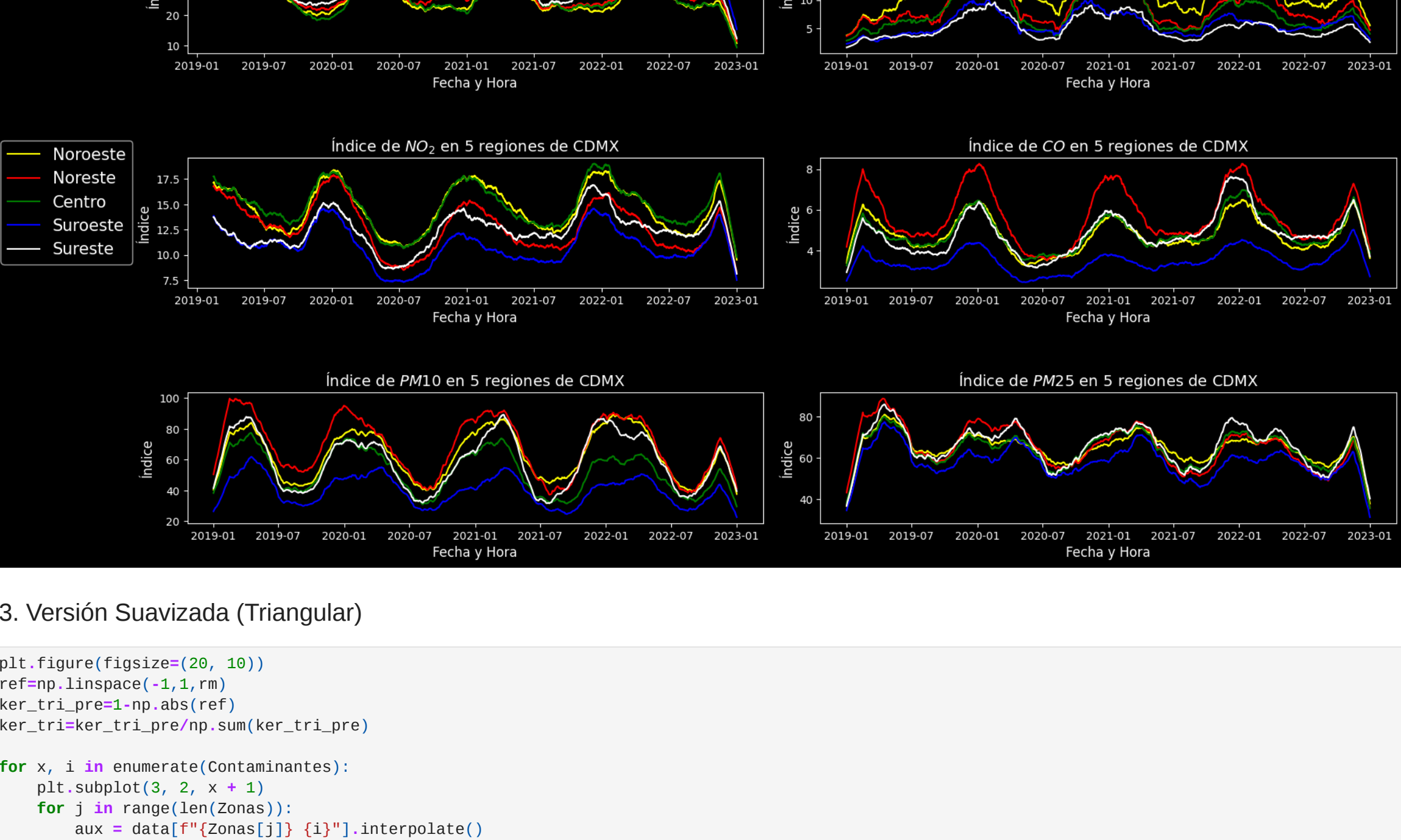
2. Versión Suavizada (Rectangular)

```
In [ ]: plt.figure(figsize=(20, 10))
ker_rec = np.ones(rm) * 1 / rm
for x, i in enumerate(Contaminantes):
    plt.subplot(3, 2, x + 1)
    for j in range(len(Zonas)):
        aux = data[f"{Zonas[j]} {i}"].interpolate()
        aux_con_rec = np.convolve(aux, ker_rec, "same")
        plt.plot(newdates, aux_con_rec, color=color[j])
    plt.xlabel("Fecha y Hora", fontsize=12)
    plt.ylabel("Índice", fontsize=12)
    plt.title(f"Índice de {nomenclatura_dict[i]} en 5 regiones de CDMX", fontsize=14)
    plt.tick_params(labelsize=10)

plt.subplots_adjust(wspace=0.1, hspace=0.8)

# Leyenda General

legends = [plt.Line2D([], [], color=color[j], label=Zonas[j]) for j in range(len(Zonas))]
plt.figlegend(handles=legends, labels=Zonas, loc='center left', fontsize=15)
plt.suptitle("Serie de Tiempo Suavizada Rectangular",fontsize=25)
plt.show()
```



3. Versión Suavizada (Triangular)

```
In [ ]: plt.figure(figsize=(20, 10))
ref=np.linspace(-1,1,rm)
ker_tri_pre=1-np.abs(ref)
ker_tri=ker_tri_pre/np.sum(ker_tri_pre)

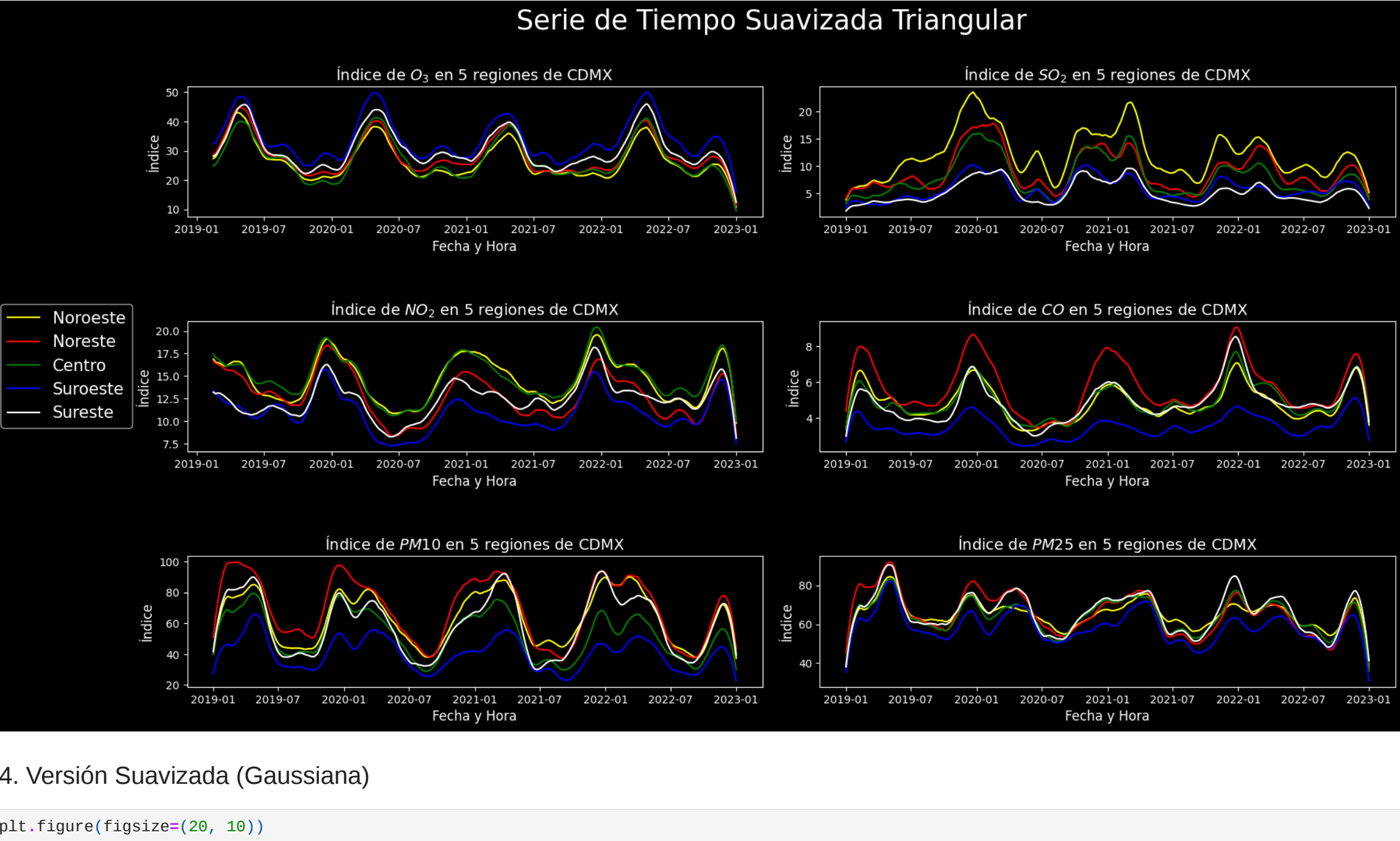
for x, i in enumerate(Contaminantes):
    plt.subplot(3, 2, x + 1)
    for j in range(len(Zonas)):
        aux = data[f"{Zonas[j]} {i}"].interpolate()
        aux_con_rec = np.convolve(aux, ker_tri, "same")
        plt.plot(newdates, aux_con_rec, color=color[j])
    plt.xlabel("Fecha y Hora", fontsize=12)
    plt.ylabel("Índice", fontsize=12)
    plt.title(f"Índice de {nomenclatura_dict[i]} en 5 regiones de CDMX", fontsize=14)
    plt.tick_params(labelsize=10)

plt.subplots_adjust(wspace=0.1, hspace=0.8)

# Leyenda General

legends = [plt.Line2D([], [], color=color[j], label=Zonas[j]) for j in range(len(Zonas))]

plt.figlegend(handles=legends, labels=Zonas, loc='center left', fontsize=15)
plt.suptitle("Serie de Tiempo Suavizada Triangular",fontsize=25)
plt.show()
```



4. Versión Suavizada (Gaussiana)

```
In [ ]: plt.figure(figsize=(20, 10))
ref=np.linspace(-1,1,rm)
ker_gauss_pre=(1 / ((1/3) * np.sqrt(2 * np.pi))) * np.exp(-(ref)**2 / (2 * (1/3)**2))
ker_gauss=ker_gauss_pre/np.sum(ker_gauss_pre)

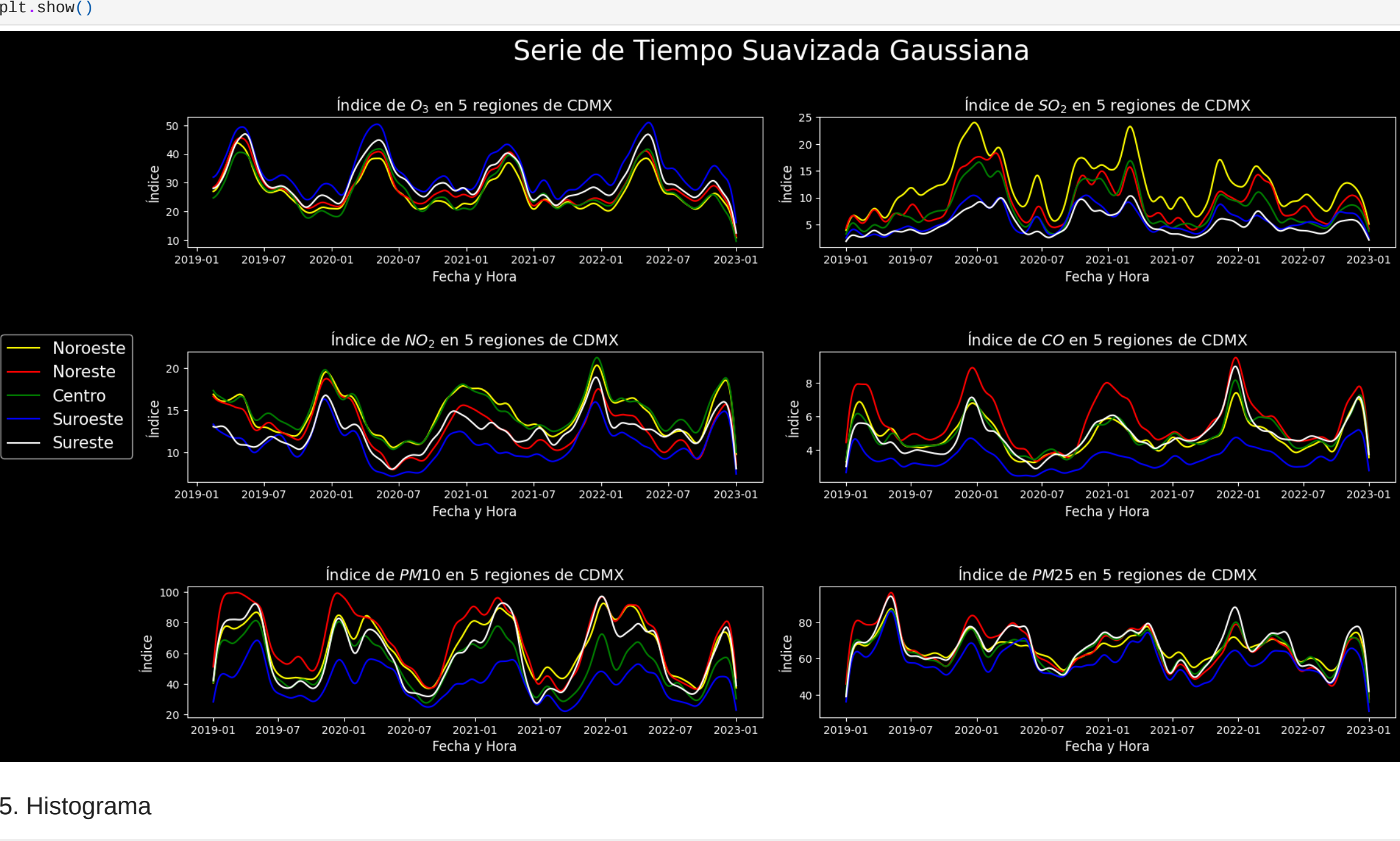
for x, i in enumerate(Contaminantes):
    plt.subplot(3, 2, x + 1)
    for j in range(len(Zonas)):
        aux = data[f"{Zonas[j]} {i}"].interpolate()
        aux_con_rec = np.convolve(aux, ker_gauss, "same")
        plt.plot(newdates, aux_con_rec, color=color[j])
    plt.xlabel("Fecha y Hora", fontsize=12)
    plt.ylabel("Índice", fontsize=12)
    plt.title(f"Índice de {nomenclatura_dict[i]} en 5 regiones de CDMX", fontsize=14)
    plt.tick_params(labelsize=10)

plt.subplots_adjust(wspace=0.1, hspace=0.8)

# Leyenda General

legends = [plt.Line2D([], [], color=color[j], label=Zonas[j]) for j in range(len(Zonas))]
plt.figlegend(handles=legends, labels=Zonas, loc='center left', fontsize=15)
plt.suptitle("Serie de Tiempo Suavizada Gaussiana",fontsize=25)

plt.show()
```



5. Histograma

```
In [ ]: plt.figure(figsize=(20, 10))

for x, i in enumerate(Contaminantes):
    plt.subplot(3, 2, x + 1)
    for j in range(len(Zonas)):
        aux = data[f"{Zonas[j]} {i}"]
        plt.hist(aux, color=color[j],histtype='step')
    plt.xlabel("Fecha y Hora", fontsize=12)
    plt.ylabel("Índice", fontsize=12)
    plt.title(f"Índice de {nomenclatura_dict[i]} en 5 regiones de CDMX", fontsize=14)
    plt.tick_params(labelsize=10)

plt.subplots_adjust(wspace=0.1, hspace=0.8)

# Leyenda General

legends = [plt.Line2D([], [], color=color[j], label=Zonas[j]) for j in range(len(Zonas))]
plt.figlegend(handles=legends, labels=Zonas, loc='center left', fontsize=15)
plt.suptitle("Histograma de la Serie de Tiempo",fontsize=25)

plt.show()
```

