

# Gráficos de linha: linheplot e paleta de cores

**Programação para Advogados – 2024.2**

José Luiz Nunes e Lucas Thevenard

# Roteiro da Aula

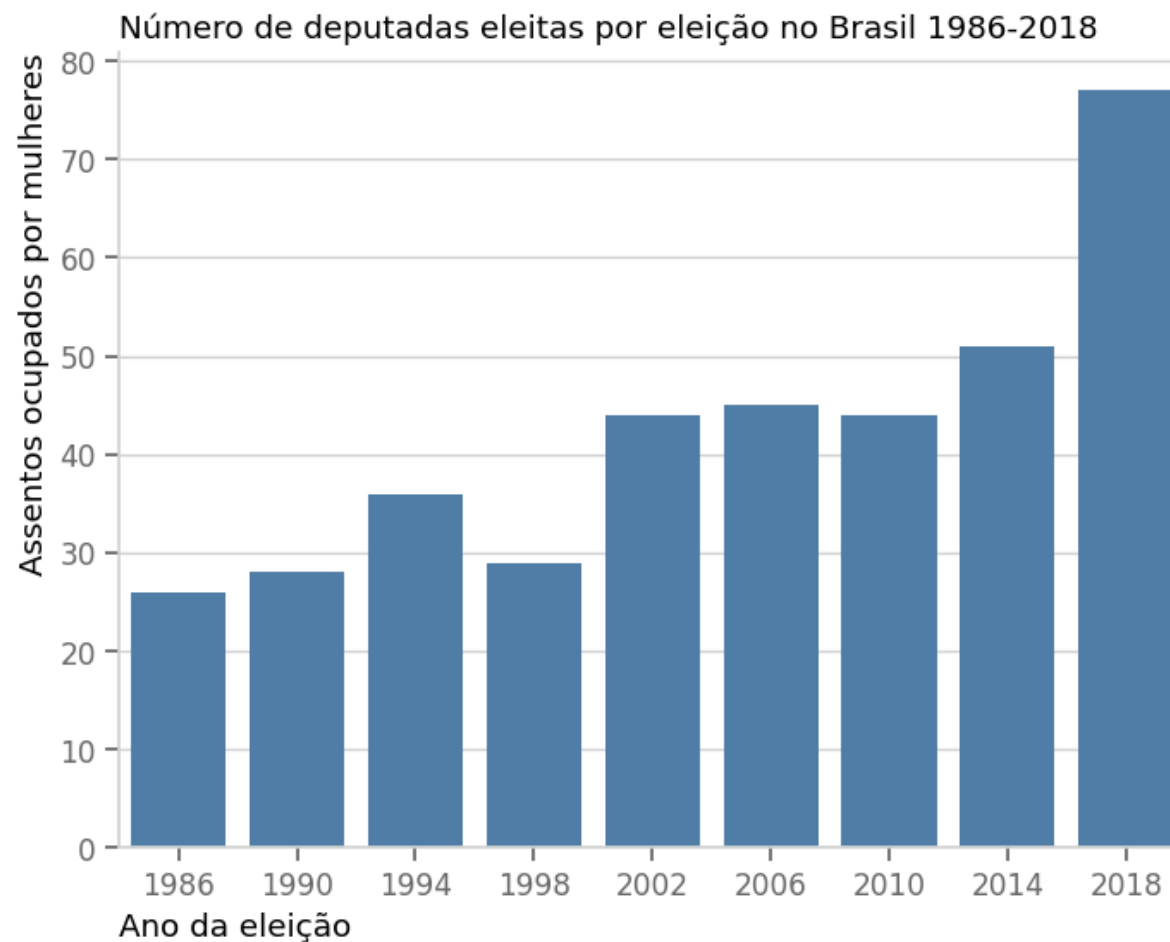
- Gráficos de linha
- Paleta de cores e ordenando categorias
- Atribuindo cor para categoria

# Tendência de uma variável numérica no tempo

- Como podemos visualizar a evolução de uma variável numérica no tempo?
- Queremos enfatizar a tendência/evolução

# Tendência de uma variável numérica no tempo

Opção 1: vamos usar o já conhecido gráfico de barras



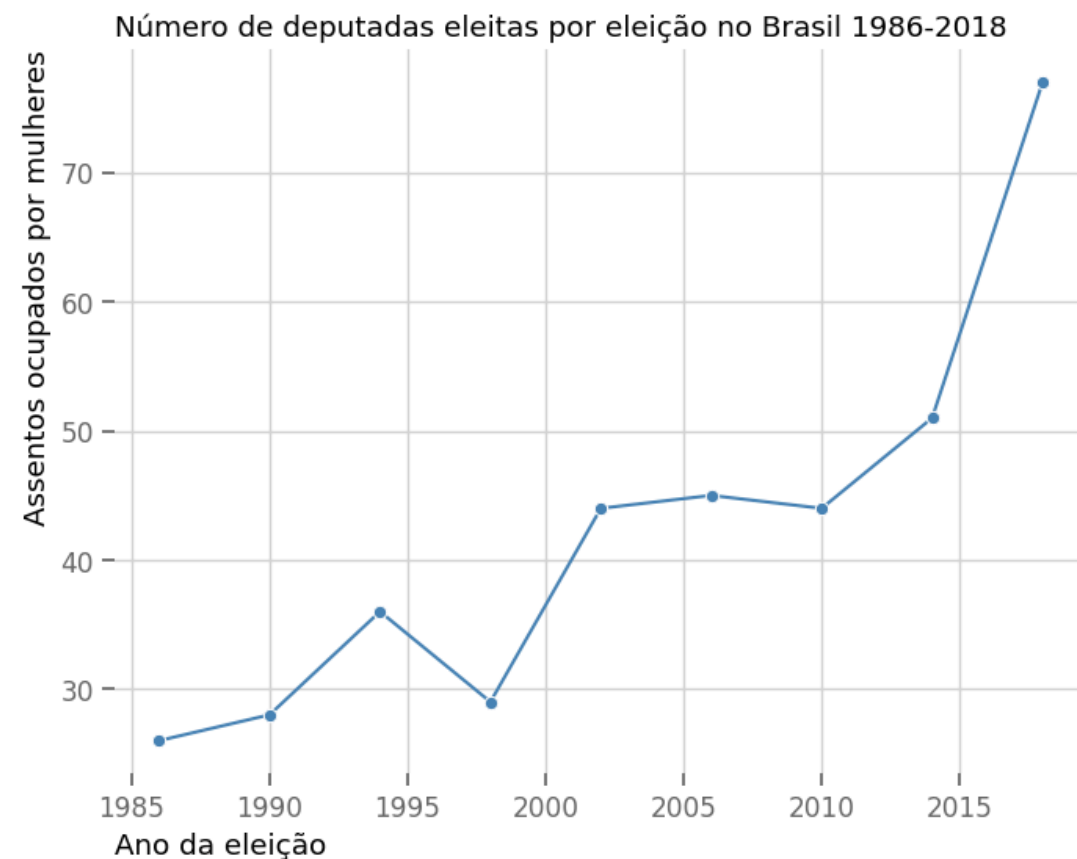
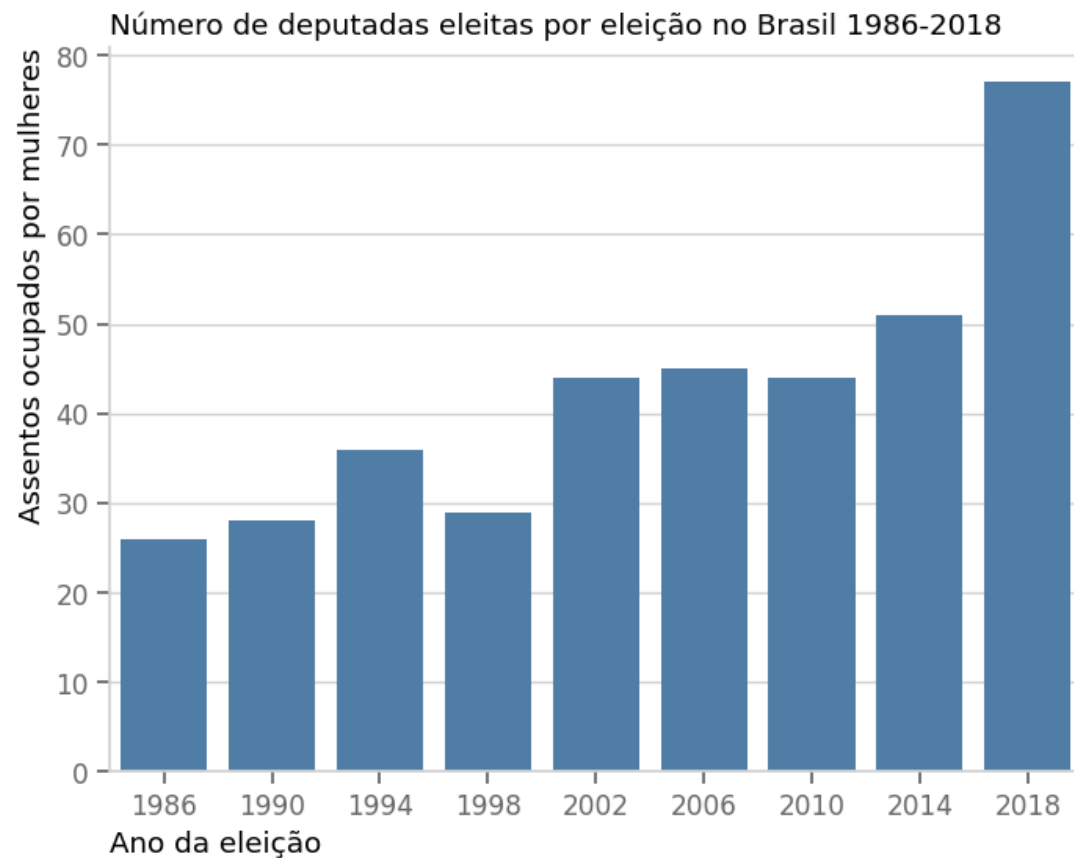
# Tendência de uma variável numérica no tempo

Opção 2: Podemos usar um gráfico de linhas

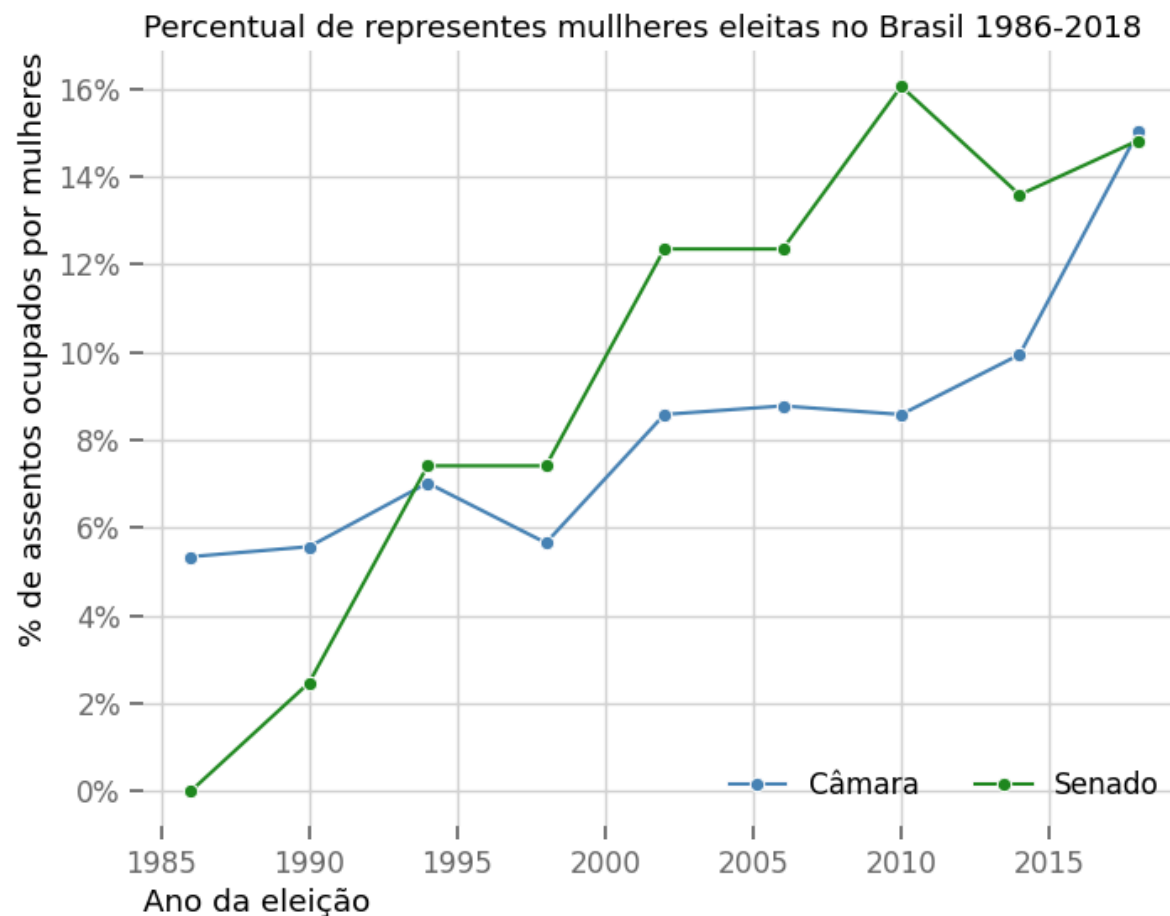


# Tendência de uma variável numérica no tempo

- Qual melhor mostra a evolução?

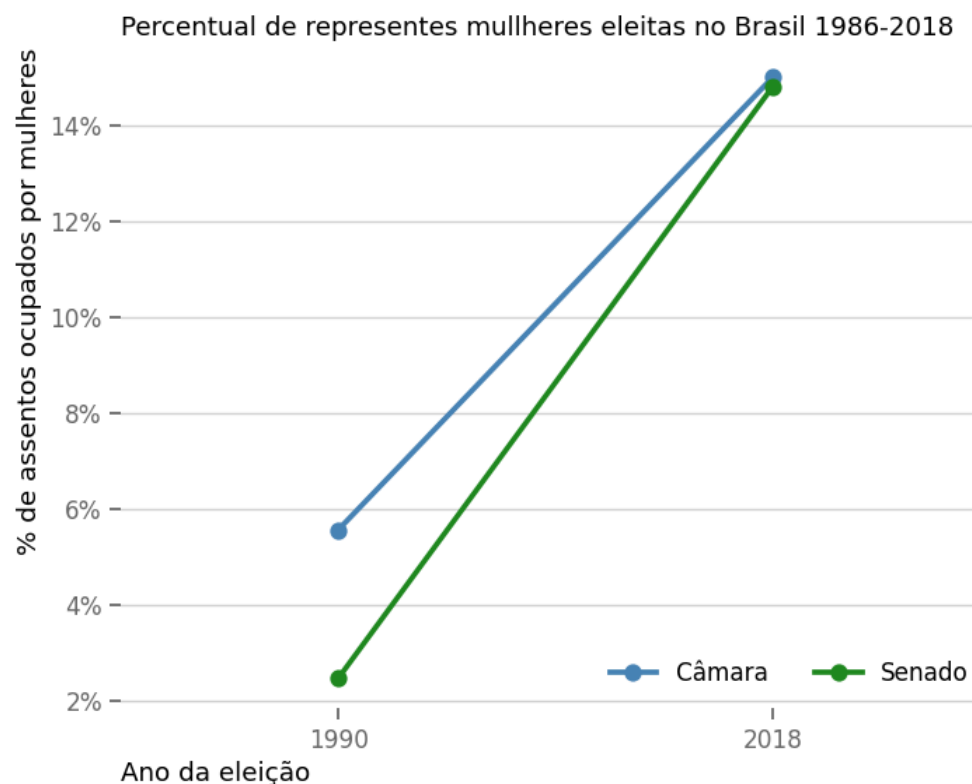


# Podemos também fazer comparações



# Podemos também fazer comparações

- Se queremos enfatizar início e fim poderíamos usar o `pointplot` apenas com o primeiro e último valor





**Voltando aos dados: `sns.lineplot`**

# Gráfico de linha

- Hoje vamos usar a função `lineplot` do Seaborn
- Vamos voltar aos dados do IDH, mas abordar a poluição per capita dos diferentes países
  - Essa variável é representada na coluna `extra_co2`

# Vamos carregar os dados

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("https://bit.ly/idh_tidy")

df.head()
```

**Atenção:** Estamos mudando o nome da variável que contém nossos dados. Hoje vamos chamar o conjunto de dados (dataframe) de `df`.

1. Assim quando formos chamar as funções de plotagem, vamos usar `data=` ??

# Vamos carregar os dados

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("https://bit.ly/idh_tidy")

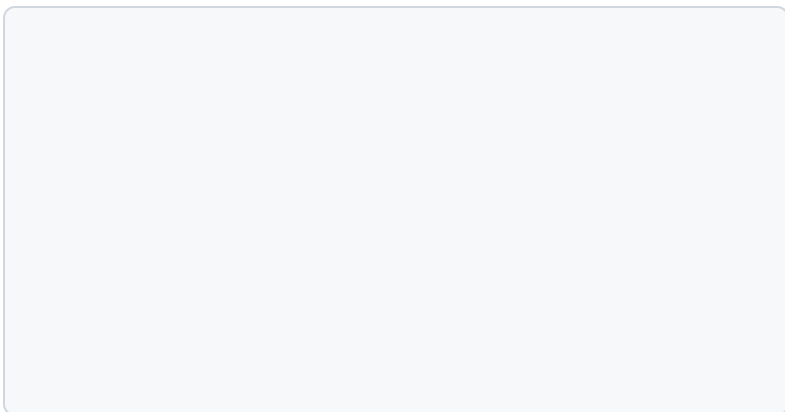
df.head()
```

**Atenção:** Estamos mudando o nome da variável que contém nossos dados. Hoje vamos chamar o conjunto de dados (dataframe) de `df`.

1. Assim quando formos chamar as funções de plotagem, vamos usar `data=df`

# Gráfico de linha

O que devemos escrever em  
nosso código?



# Gráfico de linha

O lineplot funciona de forma semelhante ao `barplot`

```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    data=df,  
)
```

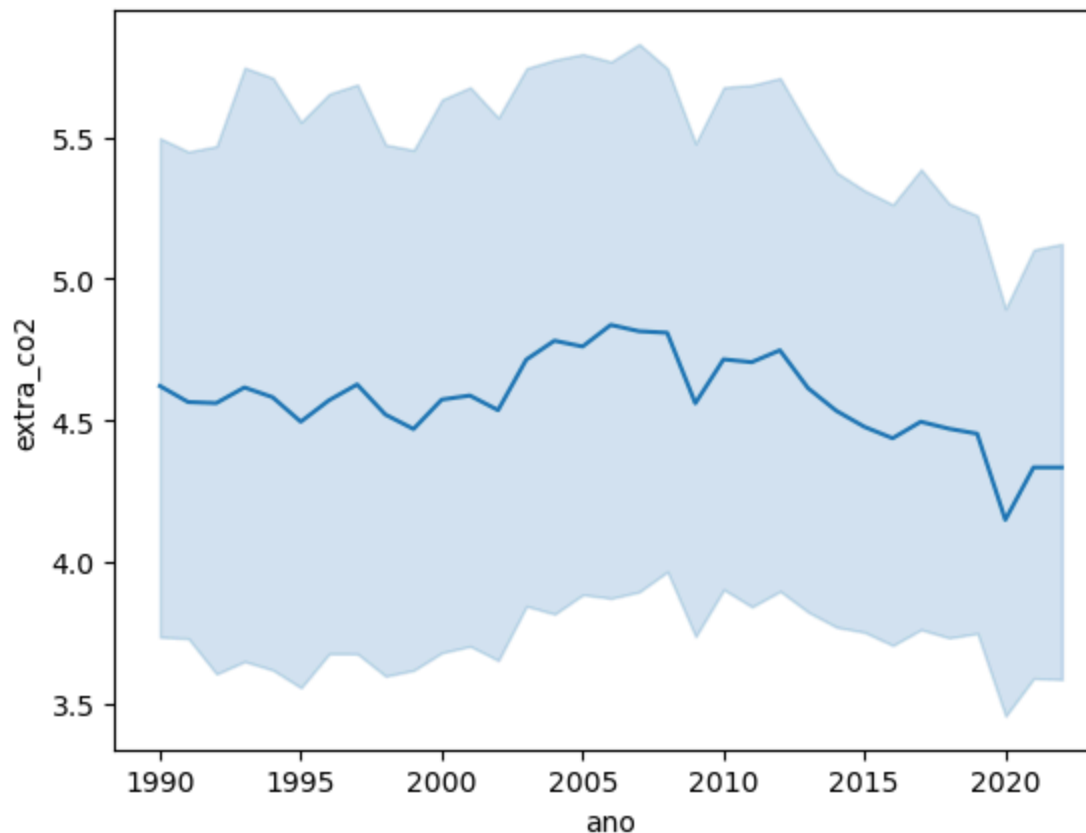
[Documentação lineplot](#)

# Gráfico de linha

O lineplot funciona de forma semelhante ao `barplot`

```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    data=df,  
)
```

Documentação lineplot



# Gráfico de linha

O lineplot funciona de forma semelhante ao `barplot`

Podemos:

- Remover a barra (sombra) de erro com `errorbar=None`
- Alterar o estimador de média para soma com `estimator=sum`

Vamos pular esses passos que já conhecemos e proceder direto para perguntas mais interessantes



# Gráfico de linha

Que pergunta poderíamos abordar do ponto de vista de tendência?

1. Como a poluição per capita evoluiu ao longo dos anos para países de diferentes regiões diferentes?
2. Qual a tendência de poluição per capita para países com diferentes níveis de IDH?

# Gráfico de linha

P1: Como a poluição per capita evoluiu ao longo dos anos para países de diferentes regiões diferentes?

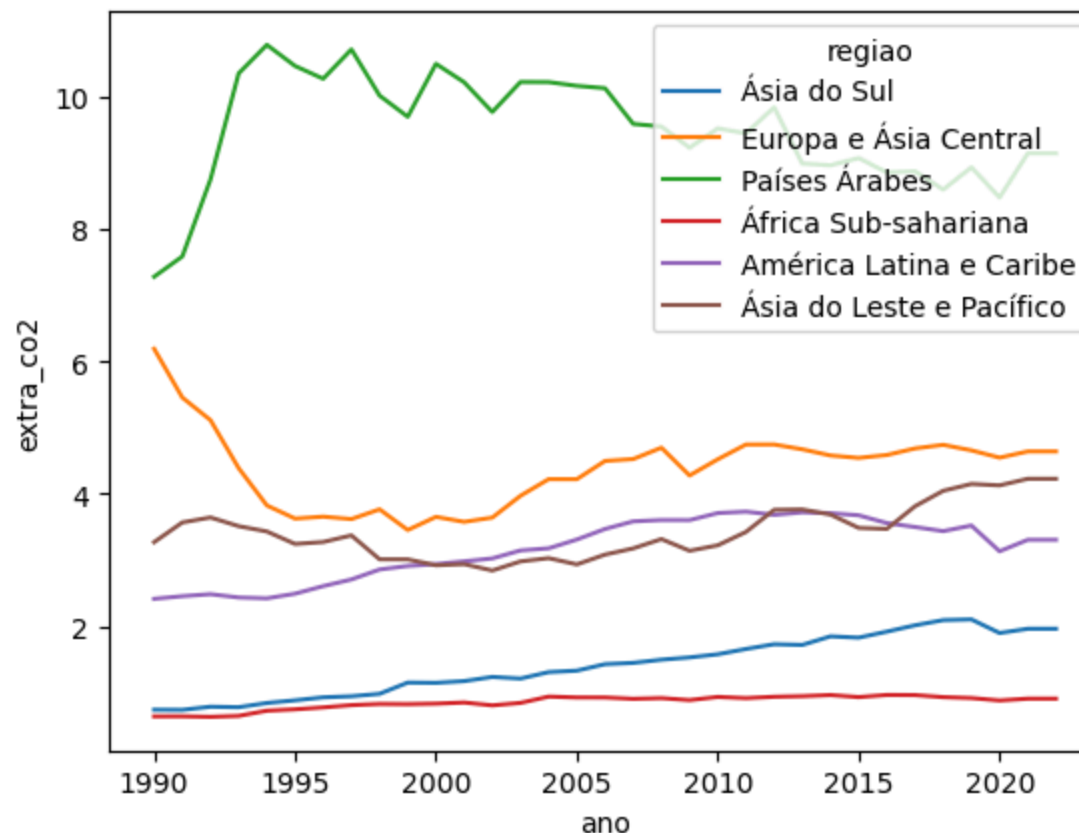
```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    ??  
    ??  
    ??  
    ??  
    data=df,  
)
```

# Gráfico de linha

Evolução da poluição per capita por região:

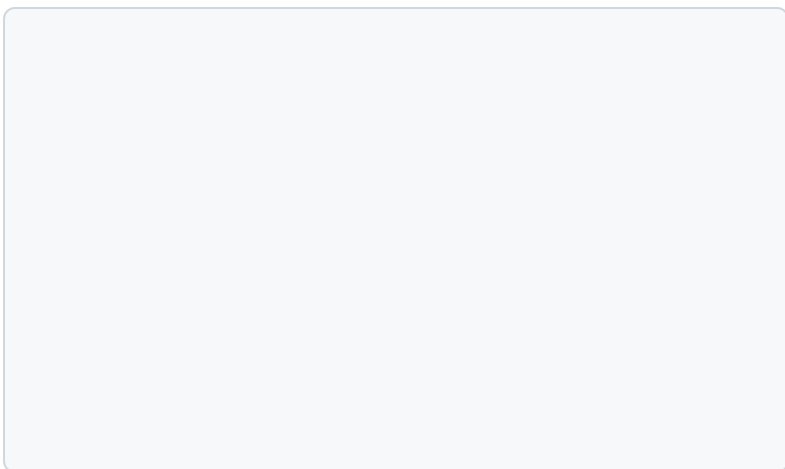
Código:

```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    hue="regiao",  
    errorbar=None,  
    data=df,  
)
```



# Gráfico de linha

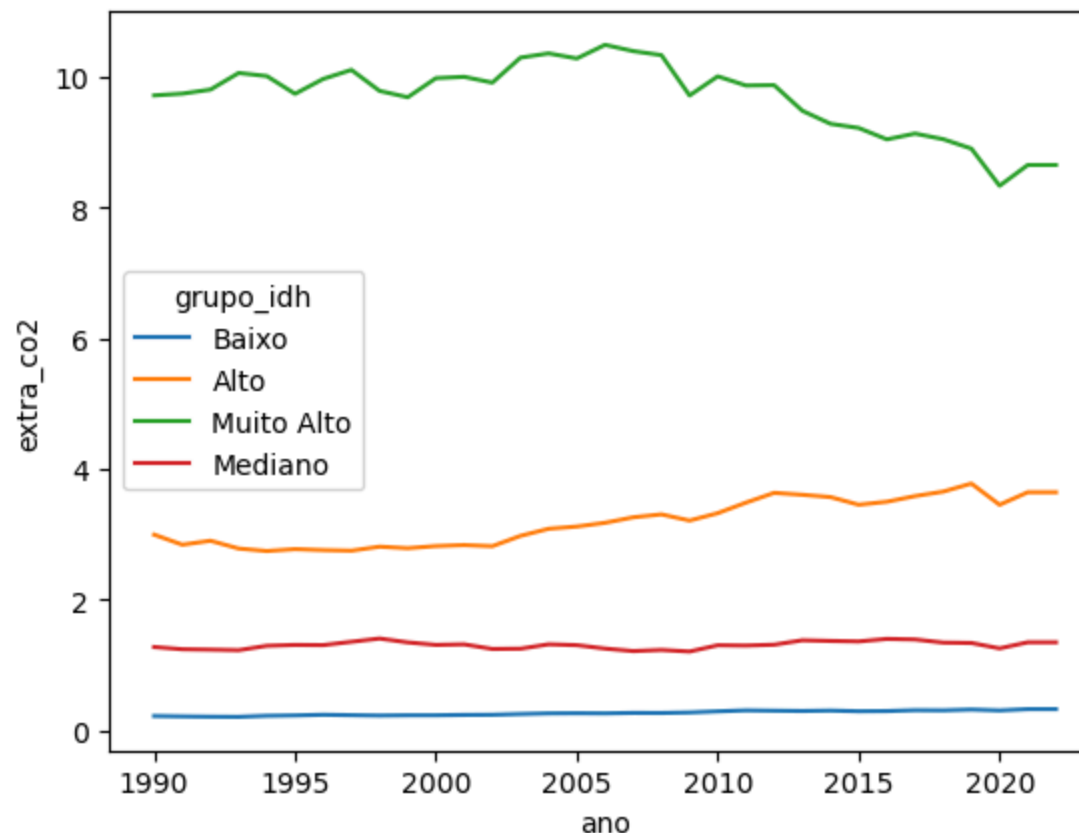
P1: Qual a tendência de poluição per capita para países com diferentes níveis de IDH?



# Gráfico de linha

P1: Qual a tendência de poluição per capita para países com diferentes níveis de IDH?

```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    hue="grupo_idh",  
    errorbar=None,  
    data=df,  
)
```

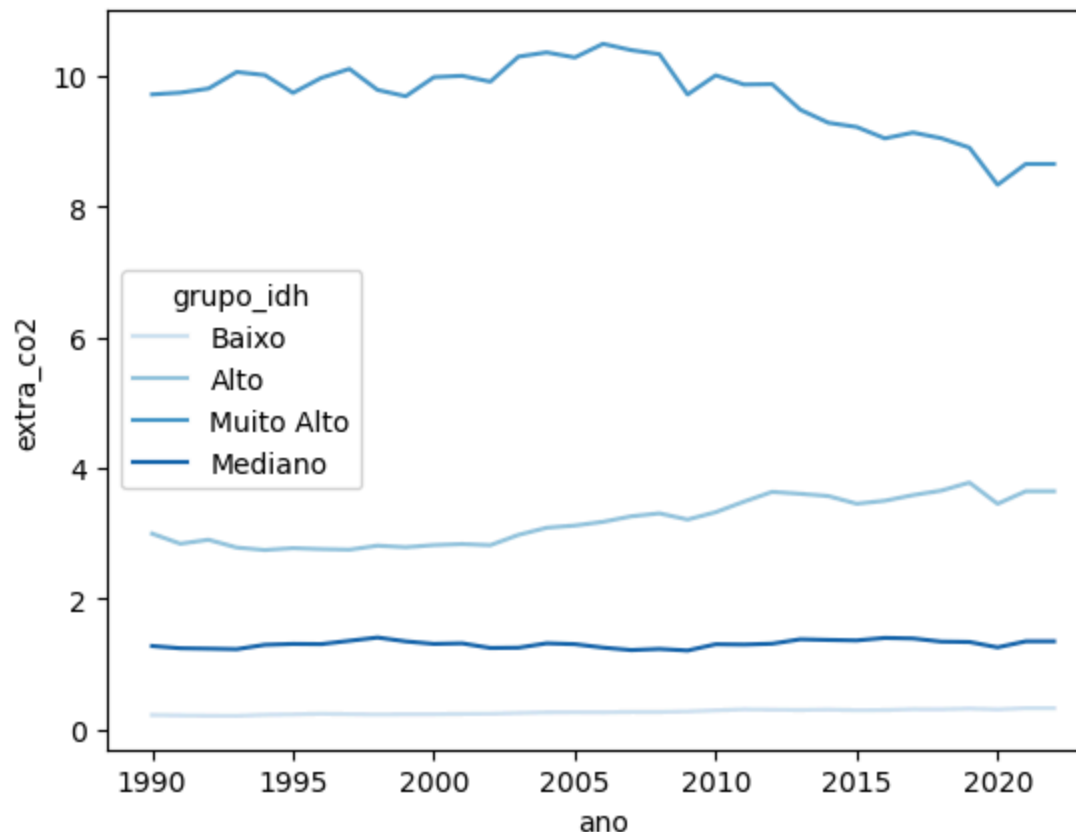


## **Agora vamos ajustar o gráfico**

# Ajustes de gráfico

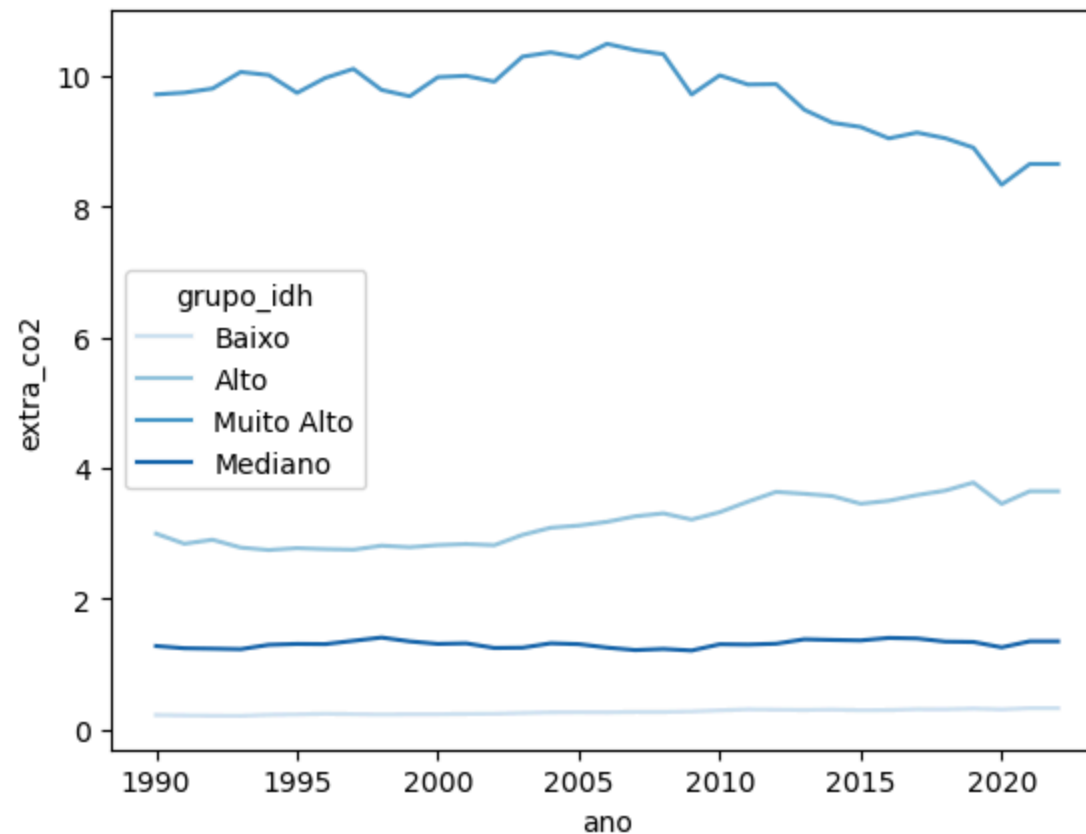
1. Vamos mudar a **paleta** de cores com o argumento `palette`

```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    hue="grupo_idh",  
    errorbar=None,  
    palette="Blues",  
    data=df,  
    ax=ax,  
)
```



# Ajustes de gráfico

Mas a ordem da categoria está estranha, não?





# Ajustes de gráfico

2. Vamos ajustar a ordem das categorias com o argumento `hue_order`

Para isso precisamos criar uma lista com o nome das categorias na ordem desejada.

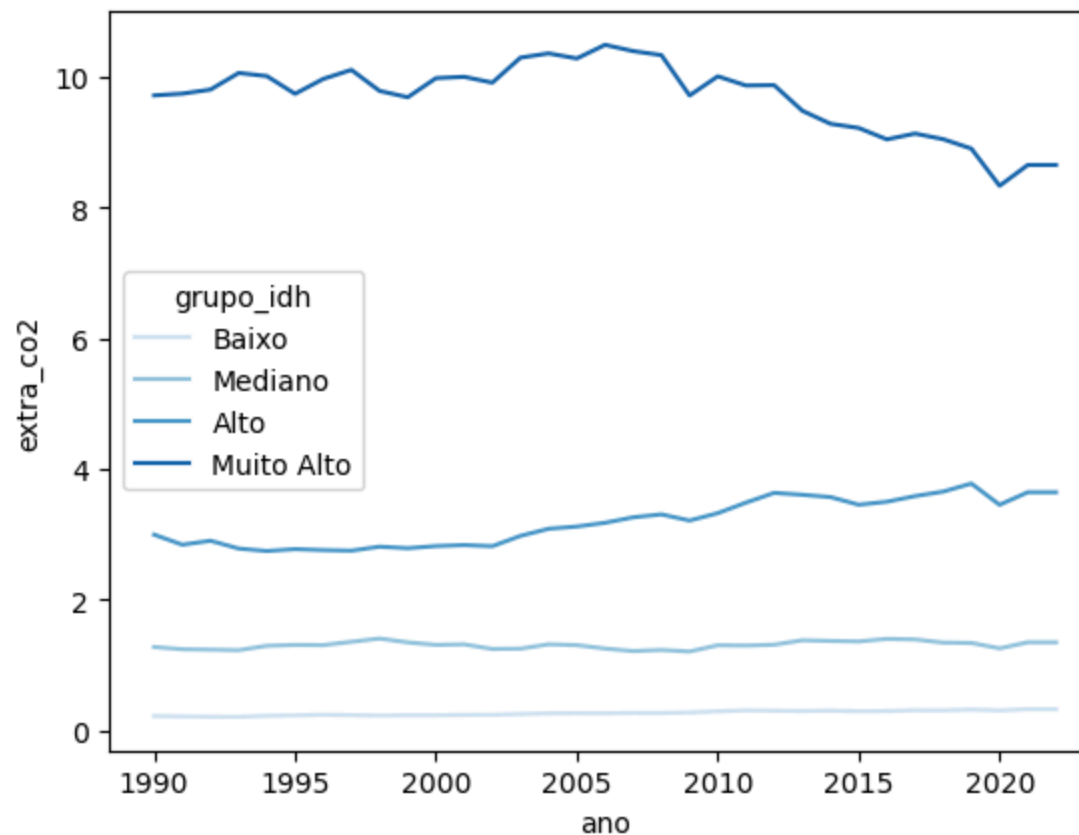
Para criar uma lista, usamos colchetes `[]` e separamos os elementos com vírgula.

Como estamos lidando com texto, precisamos colocar cada categoria entre aspas `""`.

```
ordem_grupos = ["Baixo", "Mediano", "Alto", "Muito Alto"]
```

# Ajustes de gráfico

```
ordem_grupos = [  
    "Baixo",  
    "Mediano",  
    "Alto",  
    "Muito Alto"  
]  
  
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    hue="grupo_idh",  
    hue_order=ordem_grupos,  
    errorbar=None,  
    palette="Blues",  
    data=df,  
    ax=ax,  
)
```



# Ajustes de gráfico

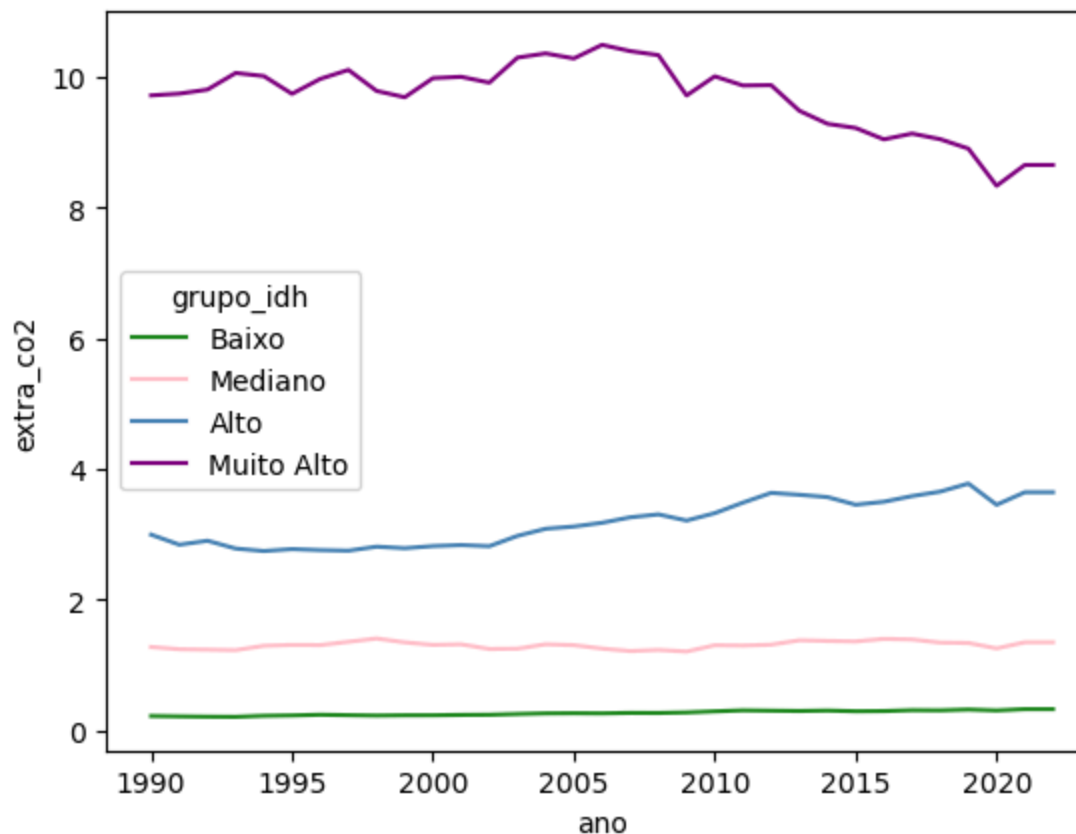
2. Agora vamos definir as cores individuais que queremos usar

Novamente, vamos criar uma lista. A correspondência das cores com categorias vai seguir a ordem que as categorias aparecem no gráfico.

```
lista_cores = ["forestgreen", "pink", "steelblue", "purple"]
```

# Ajustes de gráfico

```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    hue="grupo_idh",  
    hue_order=ordem_grupos,  
    errorbar=None,  
    palette=lista_cores,  
    data=df,  
    ax=ax,  
)
```



# Ajustes de gráfico

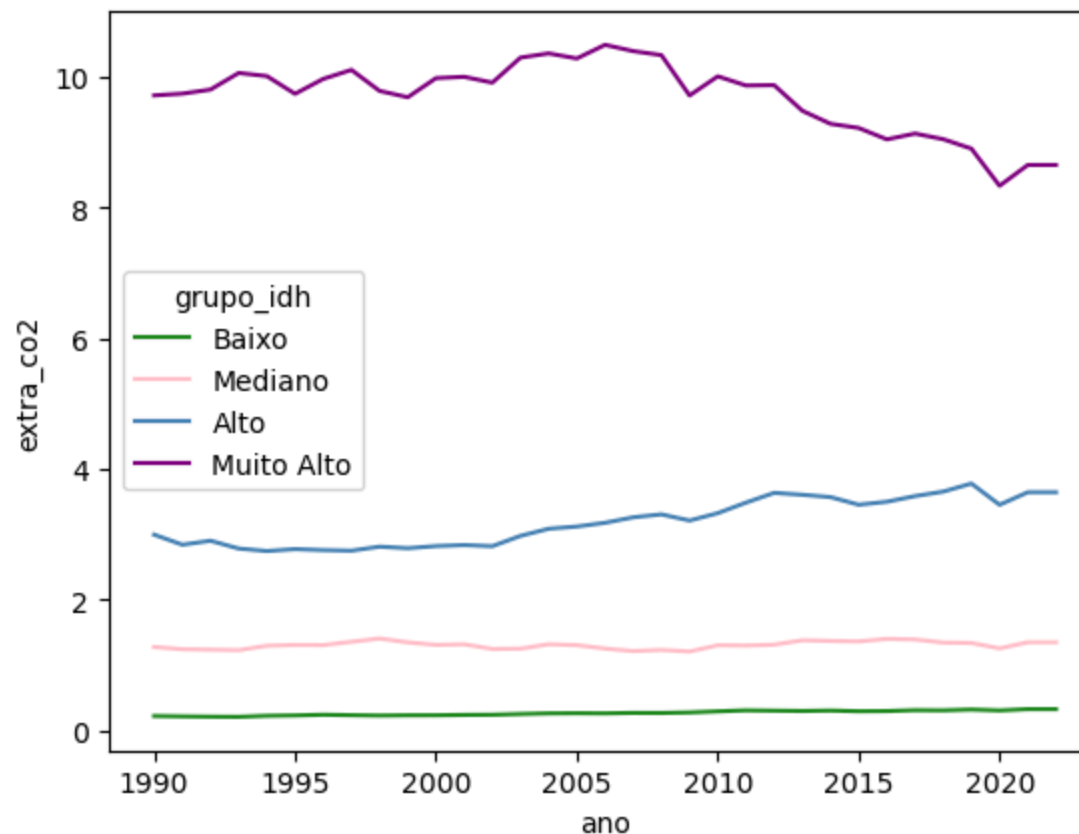
Podemos estipular as cores diretamente para cada categoria também, mas precisamos criar um dicionário.

O dicionário associa uma chave (categoria) a um valor (cor). Usamos chaves `{}` para criar, ligando cada chave ao valor com `:` e separando os pares com vírgula.

```
cores_grupos = {  
    "Baixo": "lightblue",  
    "Mediano": "pink",  
    "Alto": "forestgreen",  
    "Muito Alto": "steelblue",  
}
```

# Ajustes de gráfico

```
sns.lineplot(  
    x="ano",  
    y="extra_co2",  
    hue="grupo_idh",  
    # Podemos usar o hue_order ou não  
    # hue_order=ordem_grupos,  
    errorbar=None,  
    palette=cores_grupos,  
    data=df,  
    ax=ax,  
)
```



## Nosso gráfico final

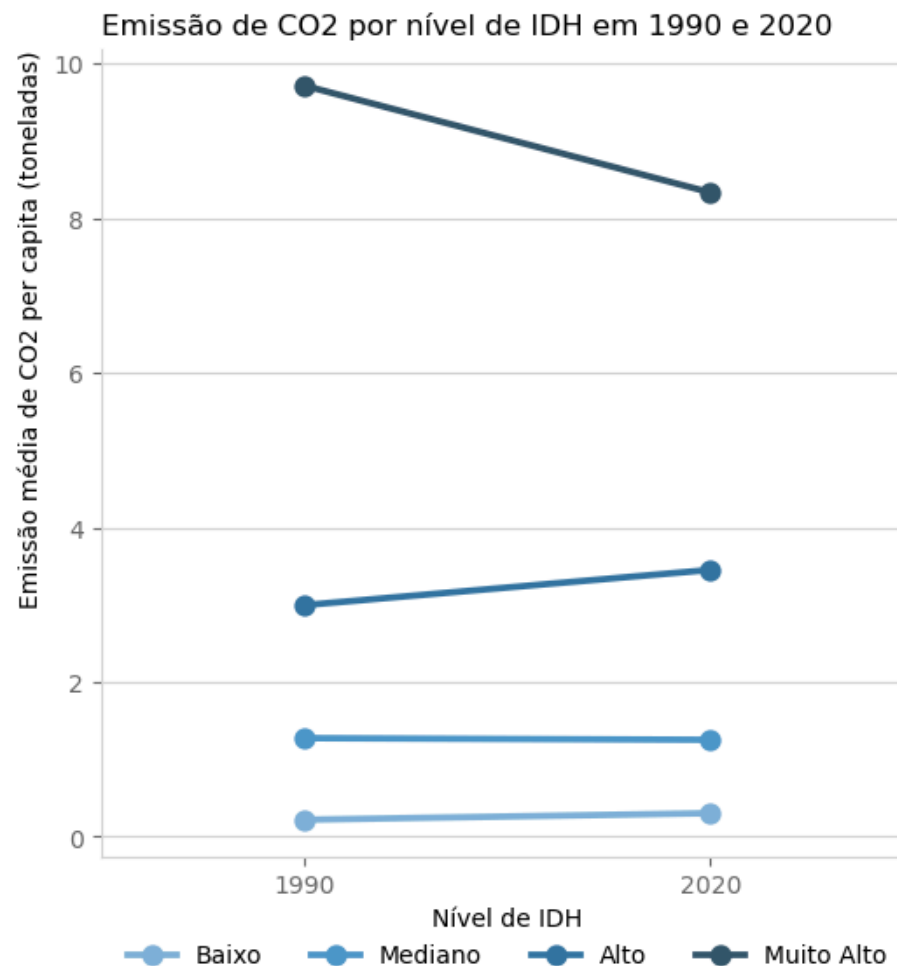
```
fig, ax = plt.subplots(figsize=(10, 6))

# Escolha o gráfico que você preferiu da aula
sns.lineplot(
    x="ano",
    y="extra_co2",
    hue="grupo_idh",
    palette=cores_grupos,
    hue_order=ordem_grupos,
    errorbar=None,
    data=df,
    ax=ax,
)

ax.set_title("Emissão anual por nível de IDH em 2022")
ax.set_ylabel("Emissão de CO2 per capita (toneladas) média")
ax.set_xlabel("")
```

# Um gráfico alternativo para enfatizar mudança total

- Deixar apenas primeiro e último ano
- Usar `pointplot`





**Mãos a obra!**