FGV DIREITO RIO

Gráficos de linha: linheplot e paleta de cores

Programação para Advogados – 2024.2

José Luiz Nunes e Lucas Thevenard

Roteiro da Aula

- Pointplot
    i. Selecionando pontos específicos
    ii. Intervalo de valores
    iii. Customização
- Lembrando: Definindo cores para categorias
- Stripplot e Swarmpot

Vamos carregar os dados

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

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

idh.head()
```

Estamos voltando para o dataset de IDH que contém a série histórica dos países.

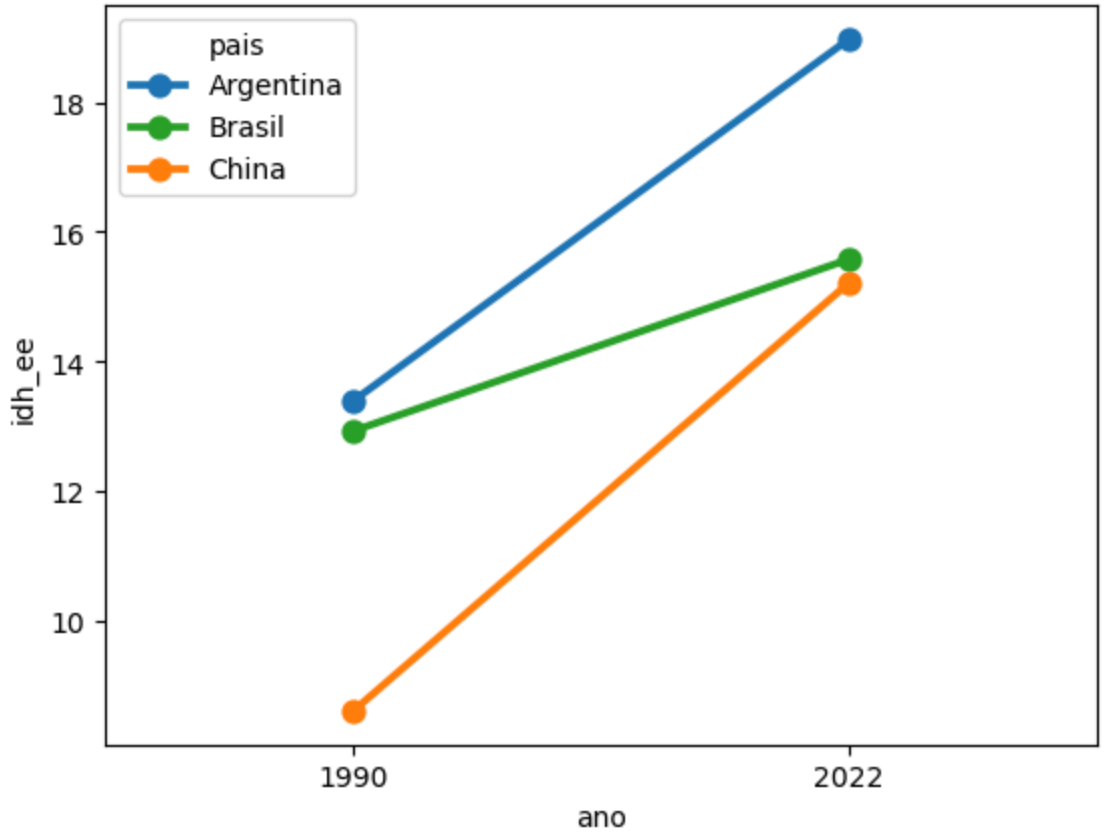
Atenção: Chamamos nosso dataset de `idh`.

Pointplot

- Em aulas anteriores, vimos como criar gráficos de linha com o `lineplot`.
- Vimos também que poderíamos destacar a mudança total com o `pointplot`, mas não fizemos esse gráfico.
- Agora temos todos os recursos para criar um gráfico nesse formato.

Pointplot

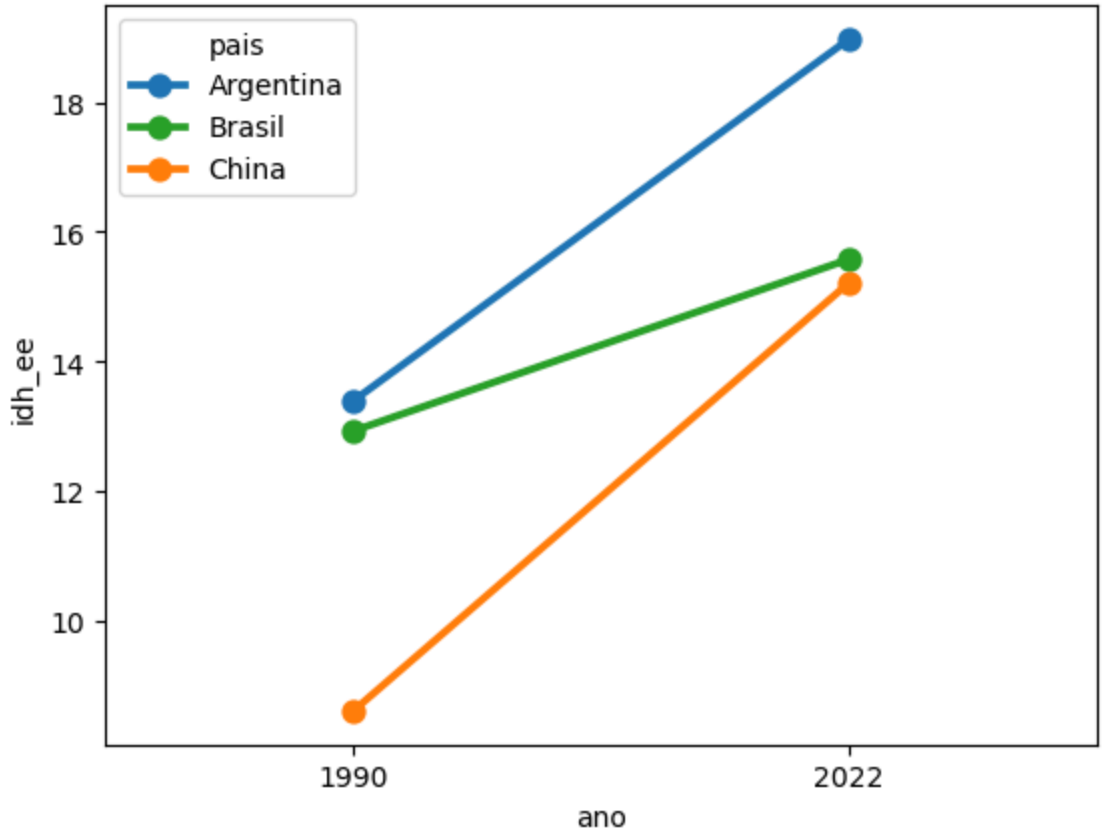
Qual o passo a passo necessário para criar esse gráfico com a função `pointplot`?



Pointplot

Qual o passo a passo necessário para criar esse gráfico?

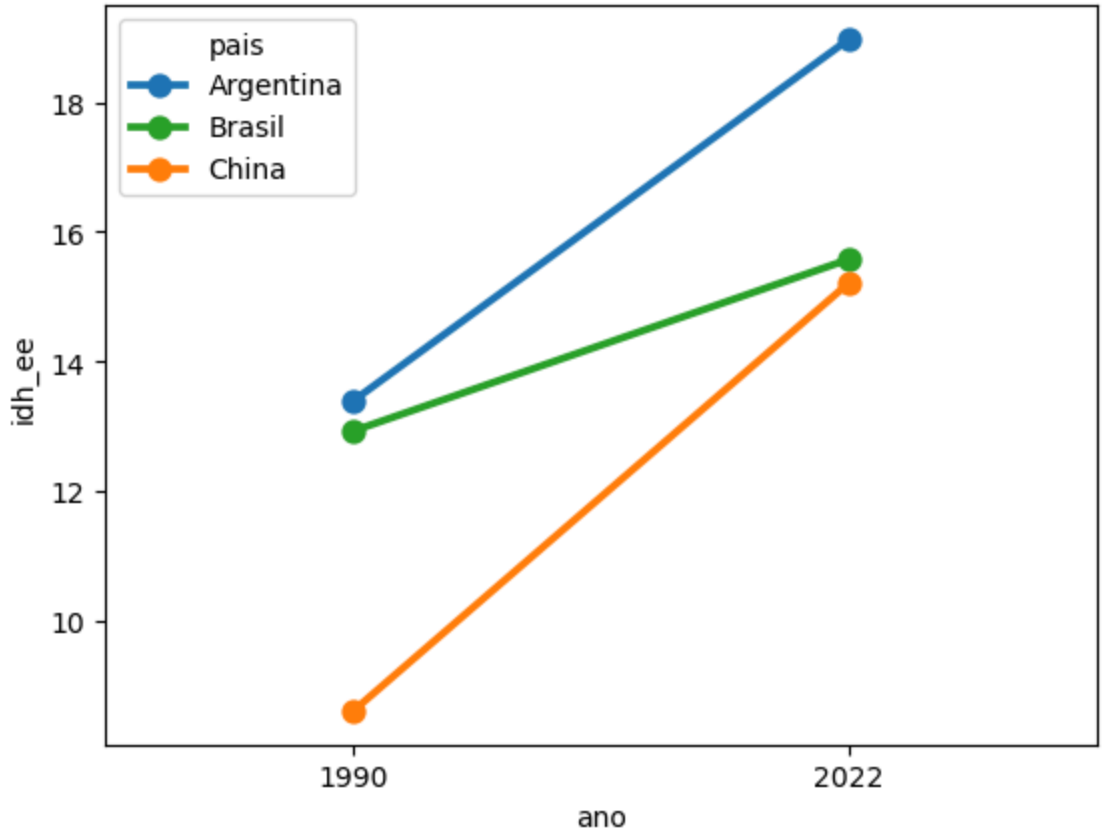
1. Selecionar os **pontos específicos** que queremos destacar.



Pointplot

Qual o passo a passo necessário para criar esse gráfico?

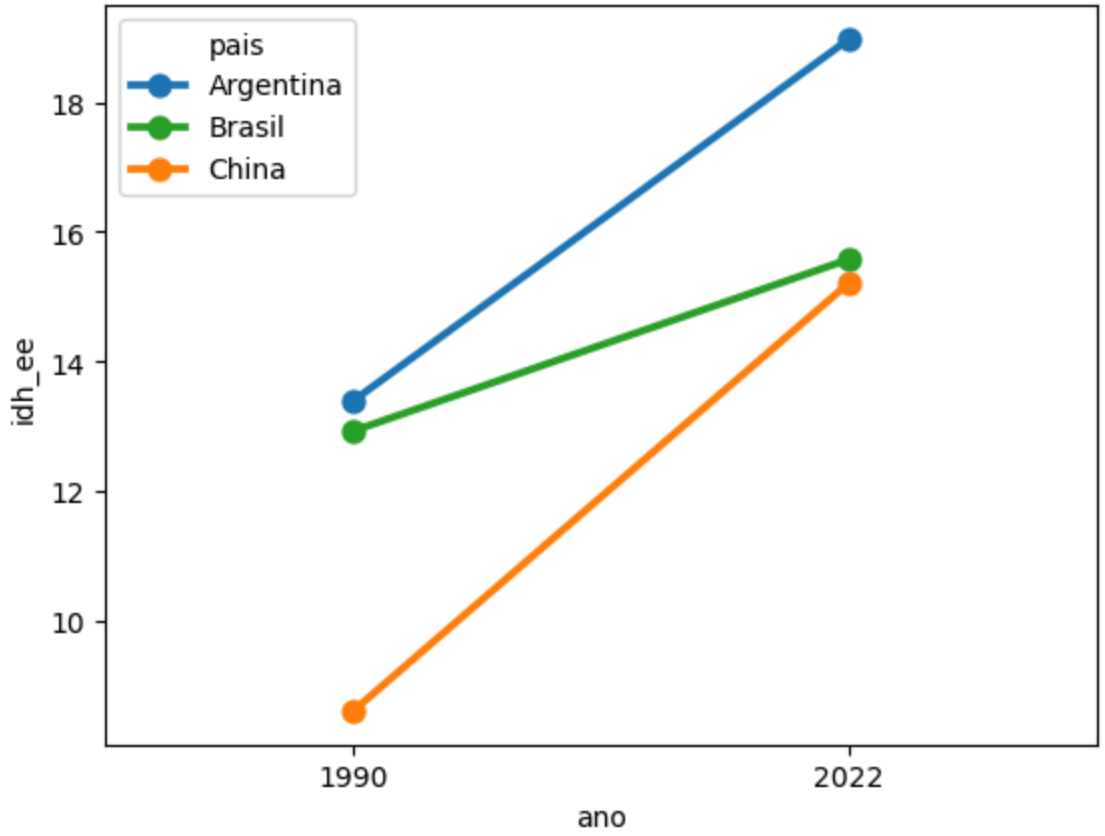
1. Selecionar os **pontos específicos** que queremos destacar.
   a) Selecionar Países
   b) Selecionar Anos



Pointplot

Qual o passo a passo necessário para criar esse gráfico?

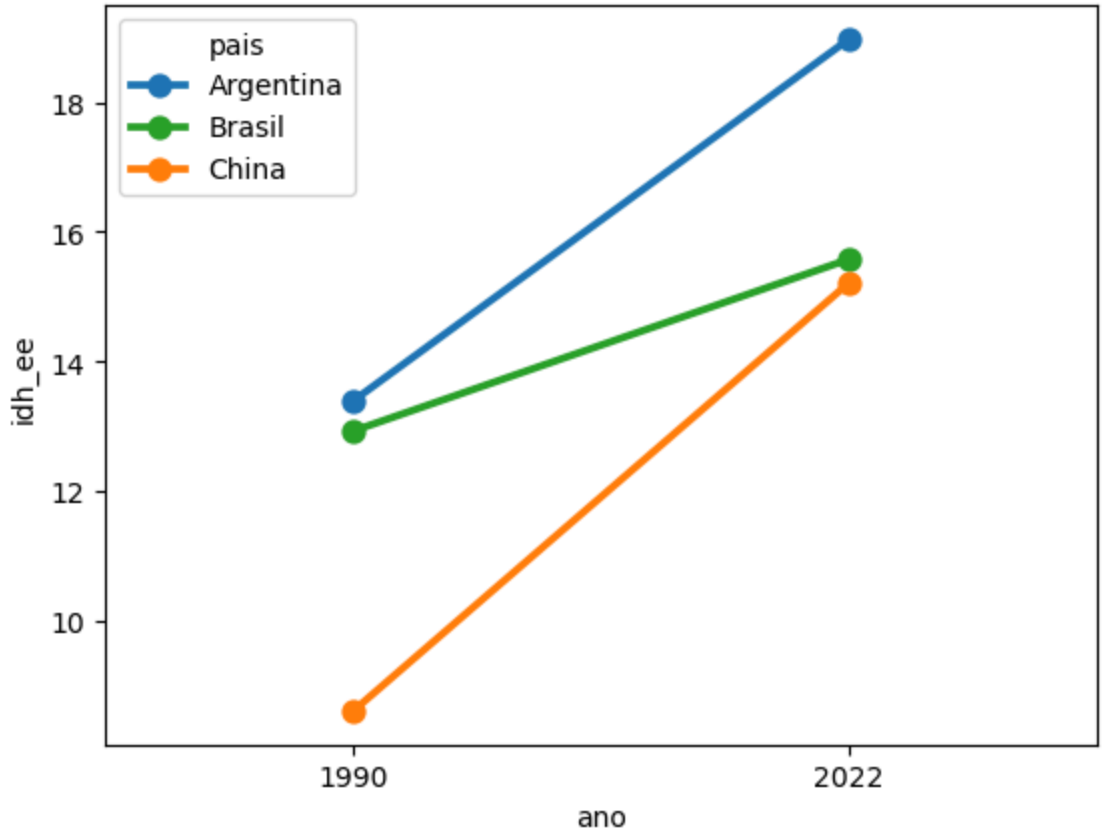
1. Selecionar os **observações** para manter.
   a) Selecionar Países
   b) Selecionar Anos
2. Definir **paleta** de cores



Pointplot

Qual o passo a passo necessário para criar esse gráfico?

1. Selecionar os **observações** para manter.
   a) Selecionar Países
   b) Selecionar Anos
2. Definir **paleta** de cores
3. Criar o **gráfico**



Pointplot

Começando: para fazer a seleção precisamos da variável `ano` e `pais`.

- Como proceder para fazer cada filtro?

Pointplot

Começando: para fazer a seleção precisamos da variável `ano` e `pais`.

- Como proceder para fazer cada filtro?

```
anos_interesse = [1990, 2022]

idh_anos = idh.query("ano in @anos_interesse")
```

Pointplot

Começando: para fazer a seleção precisamos da variável `ano` e `pais`.

- Como proceder para fazer cada filtro?

```
paises_interesse = ["Brasil", "Argentina", "China"]

df_comp_paises = idh_anos.query("pais in @paises_interesse")
```

Pointplot

Todo nosso processo:

```
idh
# Começamos com os dados carregados na variável `idh`

anos_interesse = [1990, 2022]
idh_anos = idh.query("ano in @anos_interesse")
# criamos uma nova variável com os anos de interesse

paises_interesse = ["Brasil", "Argentina", "China"]
df_comp_paises = idh_anos.query("pais in @paises_interesse")
# criamos uma terceira variável com os países de interesse
# Usamos idh_anos par manter o filtro anterior
```

Pergunta: Se alterarmos a ordem dos filtros, o resultado será o mesmo?

Pointplot

Todo nosso processo:

```
idh
# Começamos com os dados carregados na variável `idh`

anos_interesse = [1990, 2022]
idh_anos = idh.query("ano in @anos_interesse")
# criamos uma nova variável com os anos de interesse

paises_interesse = ["Brasil", "Argentina", "China"]
df_comp_paises = idh_anos.query("pais in @paises_interesse")
# criamos uma terceira variável com os países de interesse
# Usamos idh_anos par manter o filtro anterior
```

Se alterarmos a ordem dos filtros, o resultado será o mesmo!

Pointplot

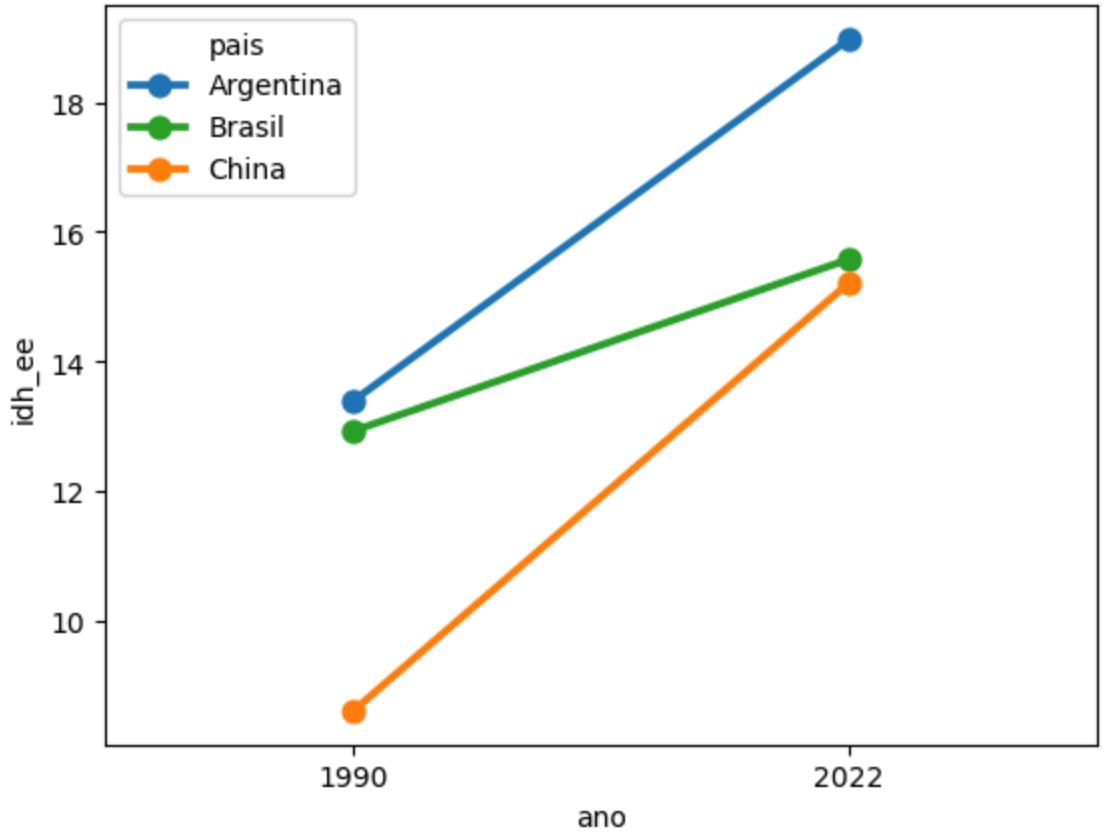
Vamos montar a paleta de cores:

```
cores_pais = {
    "Brasil": "tab:green",
    "China": "tab:orange",
    "Argentina": "tab:blue",
}
```

Pointplot

Agora estamos prontos para criar o gráfico:

```
sns.pointplot(
    x="ano",
    y="idh_ee",
    hue="pais",
    palette=cores_pais,
    data=df_comp_paises,
)
```



Pointplot

O estilo de gráficoq ue criamos pressupõe que temos uma única observação, com o valor ponto para ser visualizado.

O pointplot também pode ser usado para exibir a variação. Na verdade, é uma opção melhor que o gráfico de barras, por exemplo.

Pointplot

Vamos restringir os dados para o ano de 2022 para prosseguir:

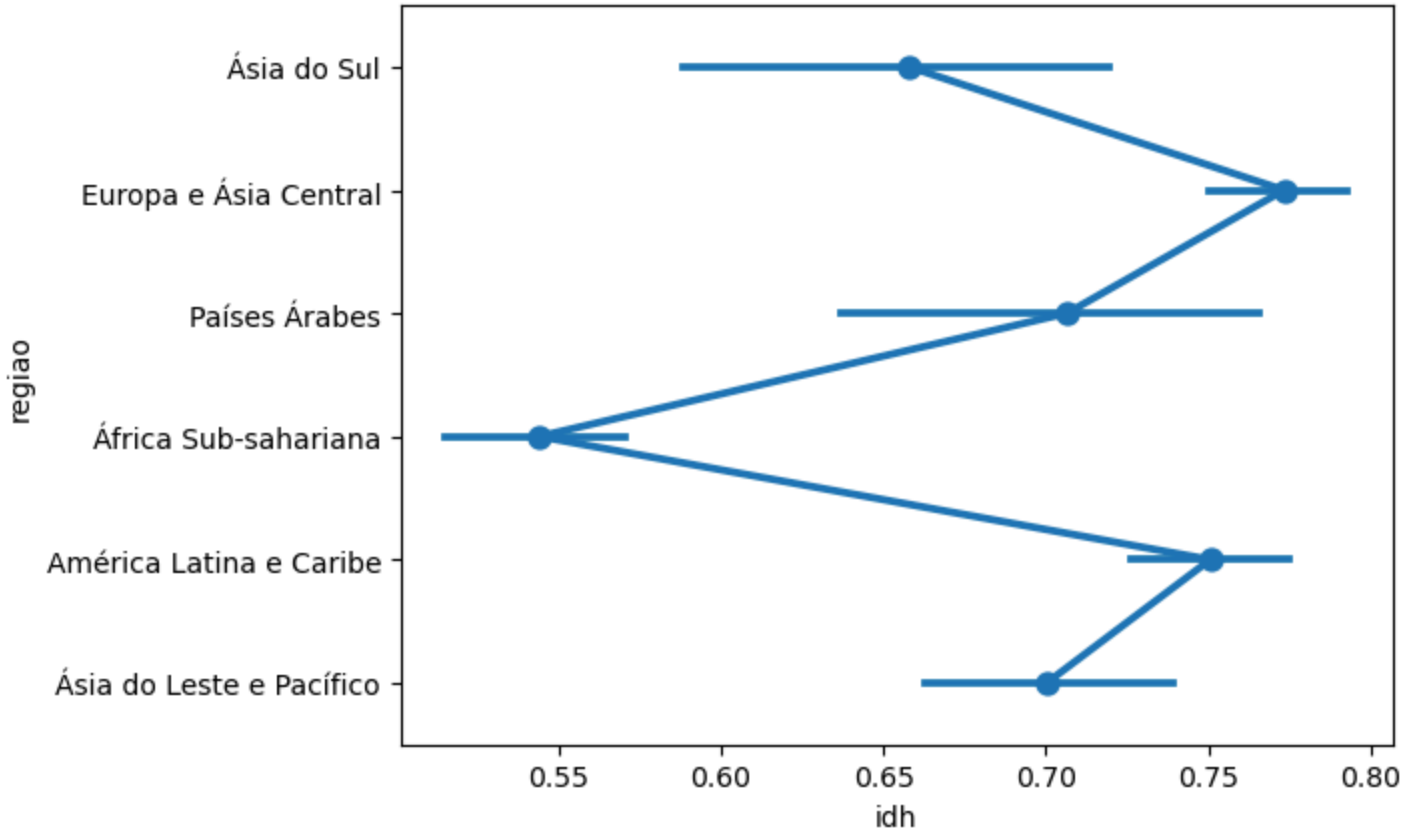
```
idh_2022 = idh.query("ano == 2022")
```

Atenção: O resultado dessa linha será o **mesmo conjunto de dados** que utilizamos em outras aulas, mas carregado diretamente: as informações do IDH de cada país para o ano de 2022.

Pointplot

Passando ao gráfico:

```
sns.pointplot(
    x="idh",
    y="regiao",
    data=idh_2022
)
```

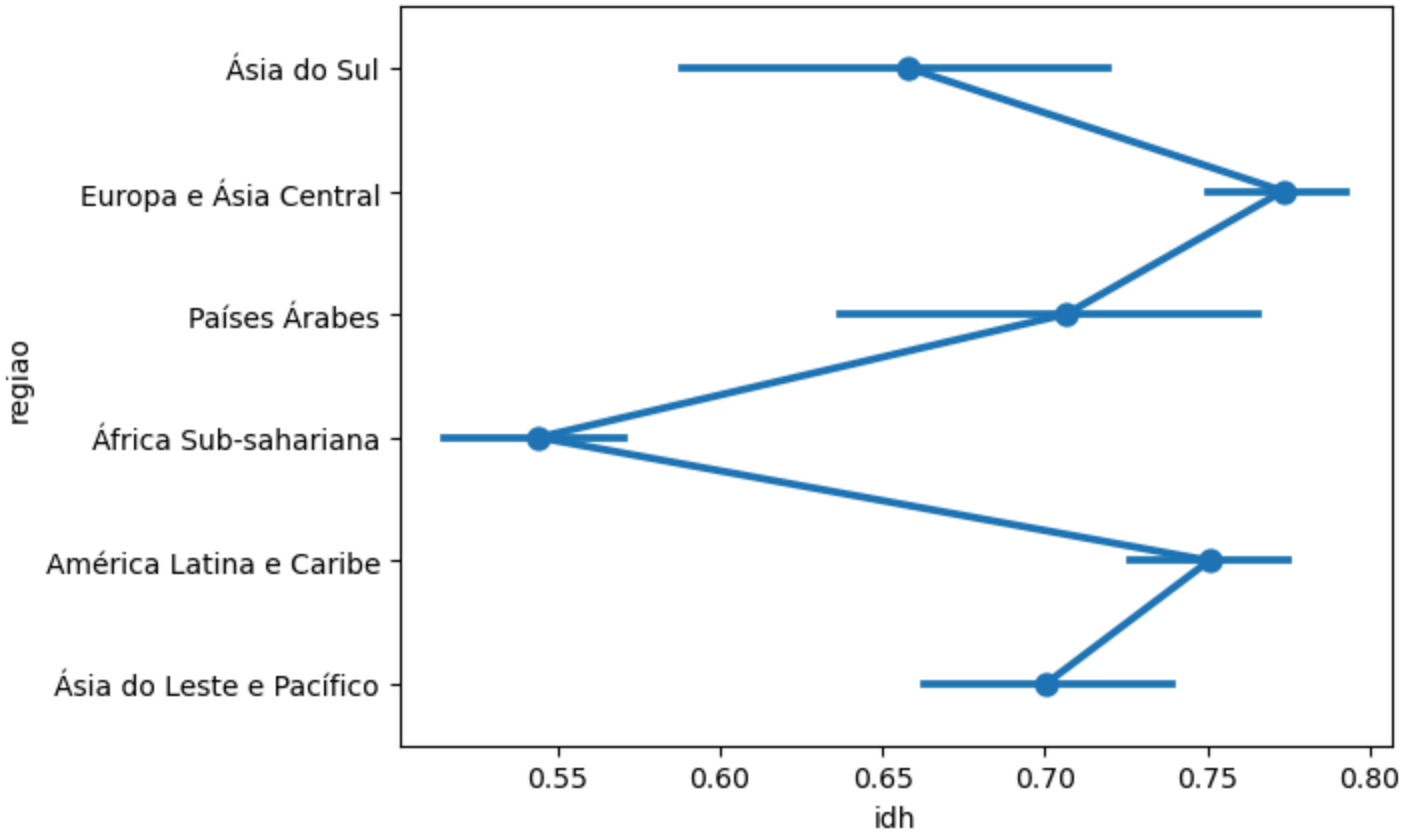


Pointplot

O `pointplot` liga os pontos, semelhante ao `lineplot`.

Ele pode ser usado para destacar variação temporal de variáveis, ou entre categorias não associadas a passagem do tempo.

Também podemos remover a linha, o que em muitos casos é uma ideia melhor.



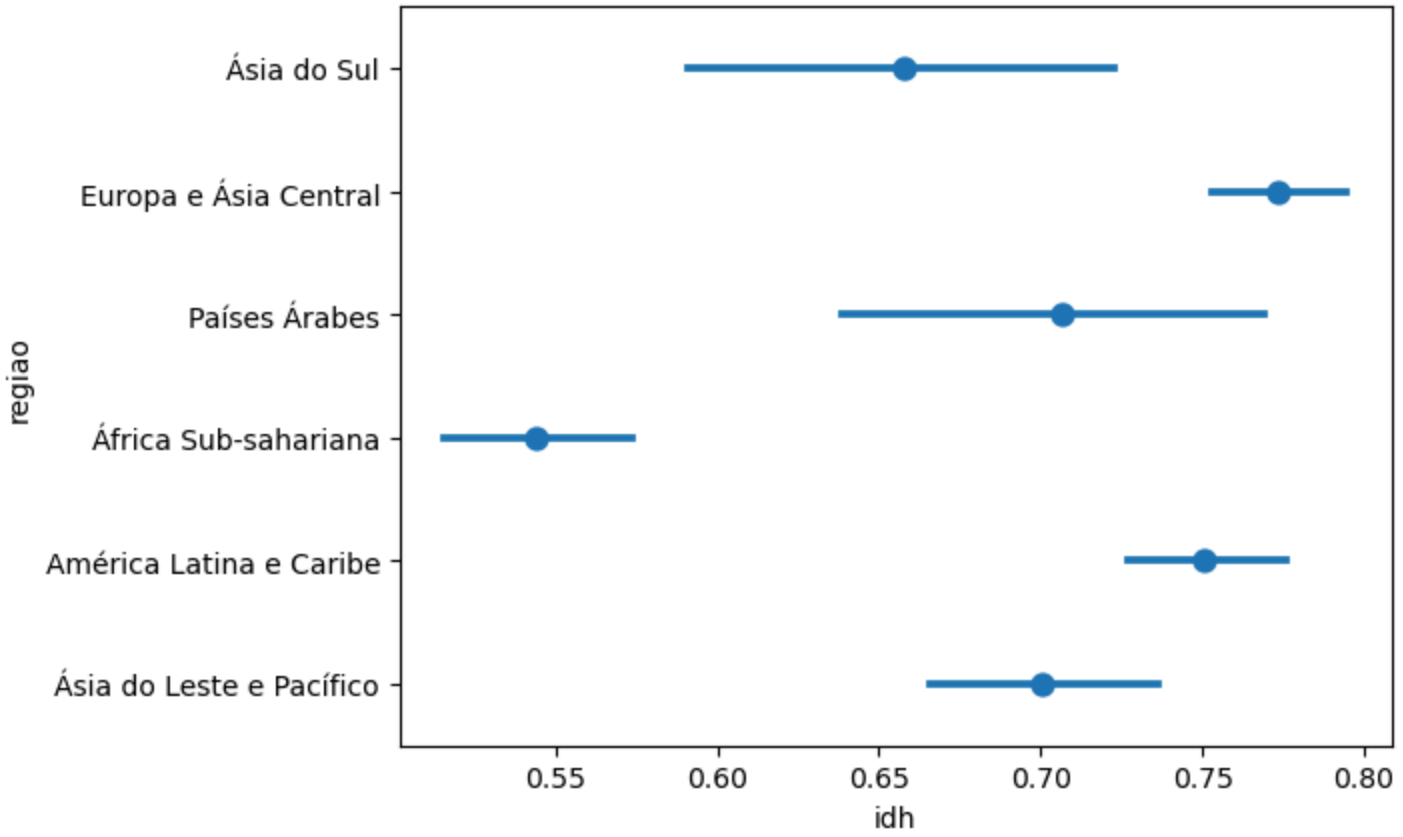
Pointplot

Vamos remover a linha:

```
sns.pointplot(
    x="idh",
    y="regiao",
    linestyle="", # <----
    data=idh_2022
)
```

Ponto é a média.

Barra de erro é o intervalo de confiança, como nas demais funções.



Barras de erro

O seaborn usa um método estatístico chamado *bootstrapping* para calcular o intervalo de confiança de 95% exibido por padrão.

Não precisamos nos preocupar com como isso é feita agora, mas vamos aprender a mudar a barra de erro.

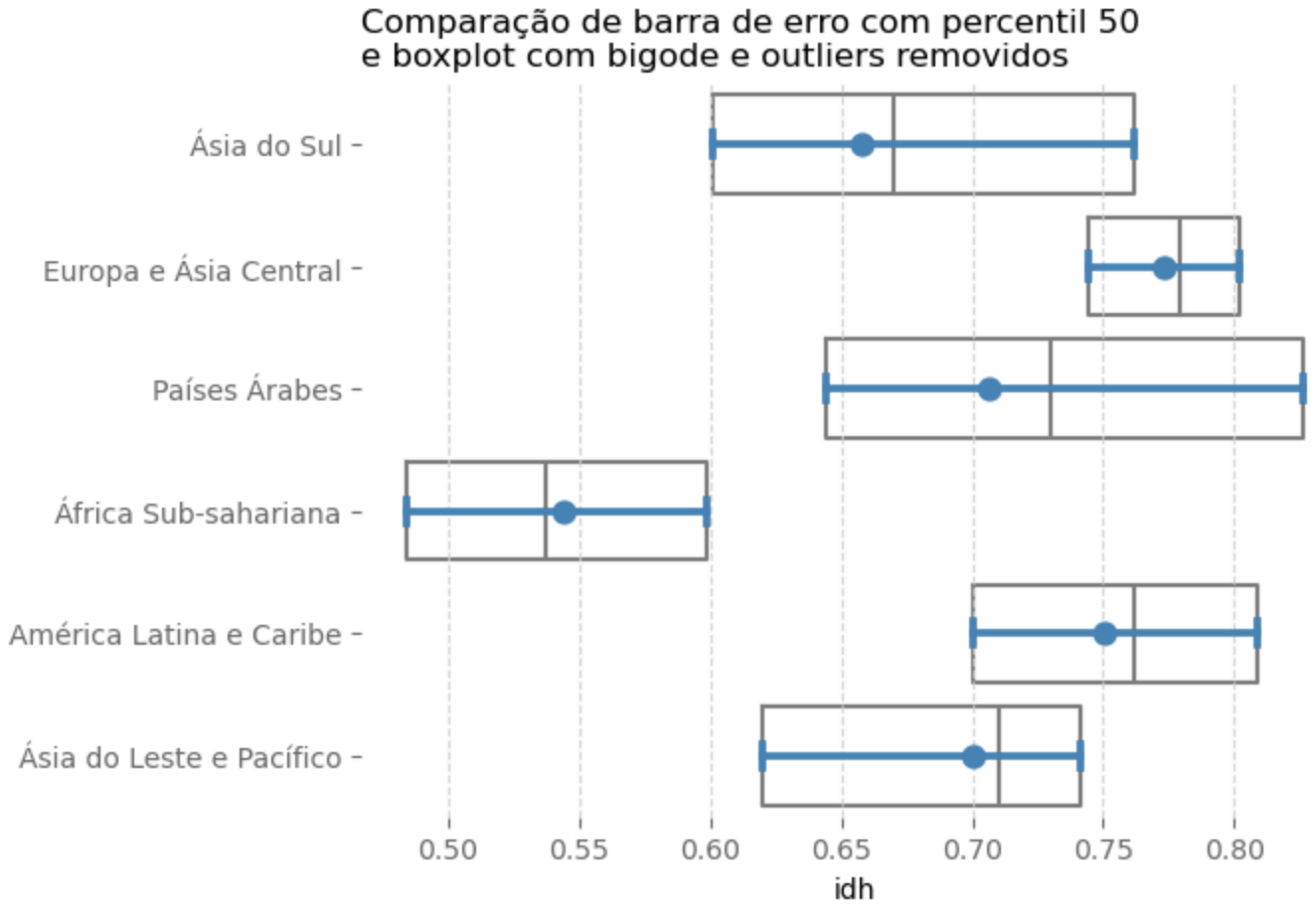
Barras de erro

Para isso vamos usar a ideia de percentil, que generaliza o conceito de quartil para qualquer percentual.

E.g. o percentil 25 é equivalente ao primeiro quartil. O percentil 43 é maior que 43% dos valores (e menor que 57%).

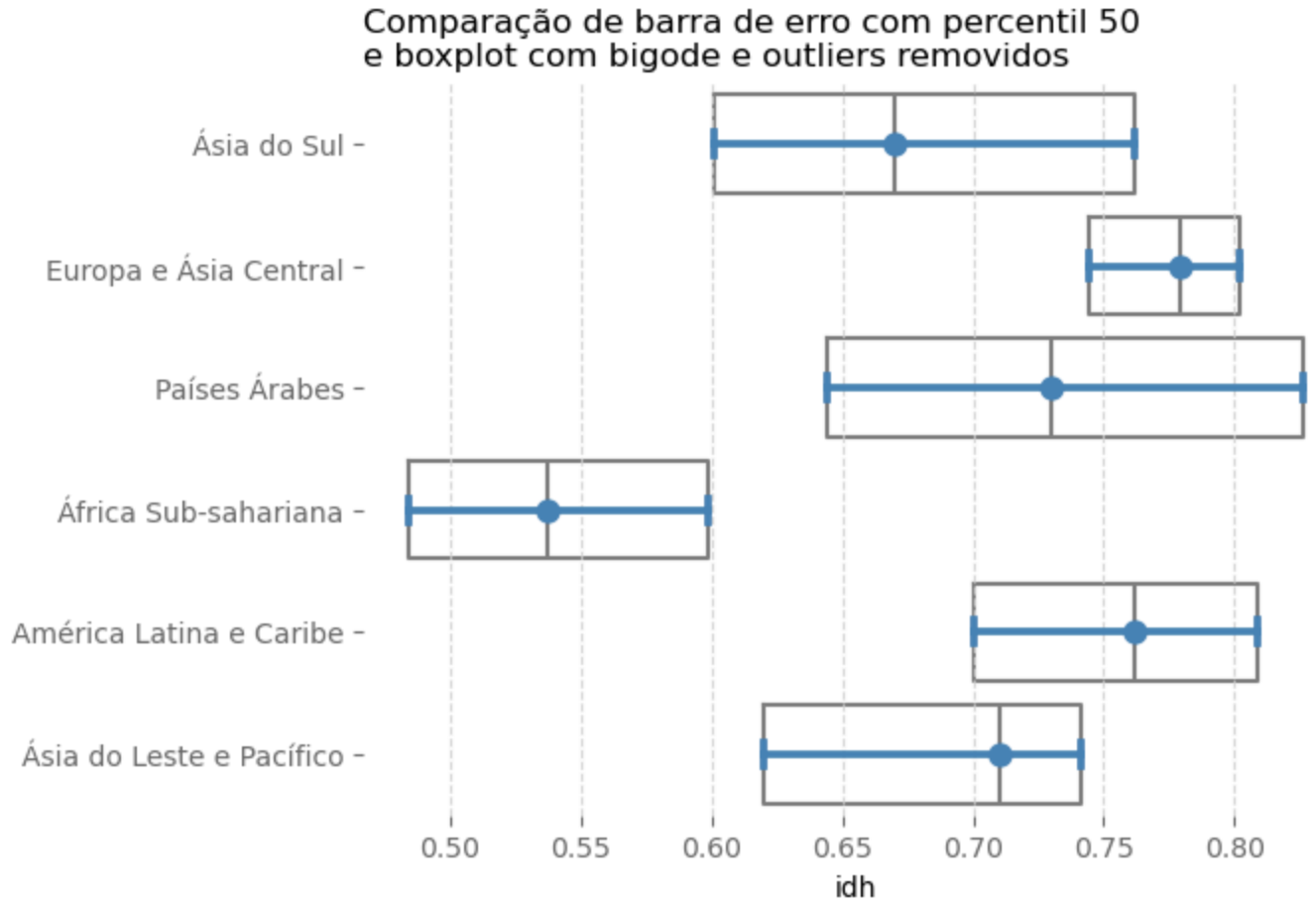
Barras de erro

Barra de erro com percentil 50:



Barras de erro

Podemos mudar o estimador para `estimator="median"`:



Barras de erro

Para fazer a barra de erro desse modo devemos passar o argumento:

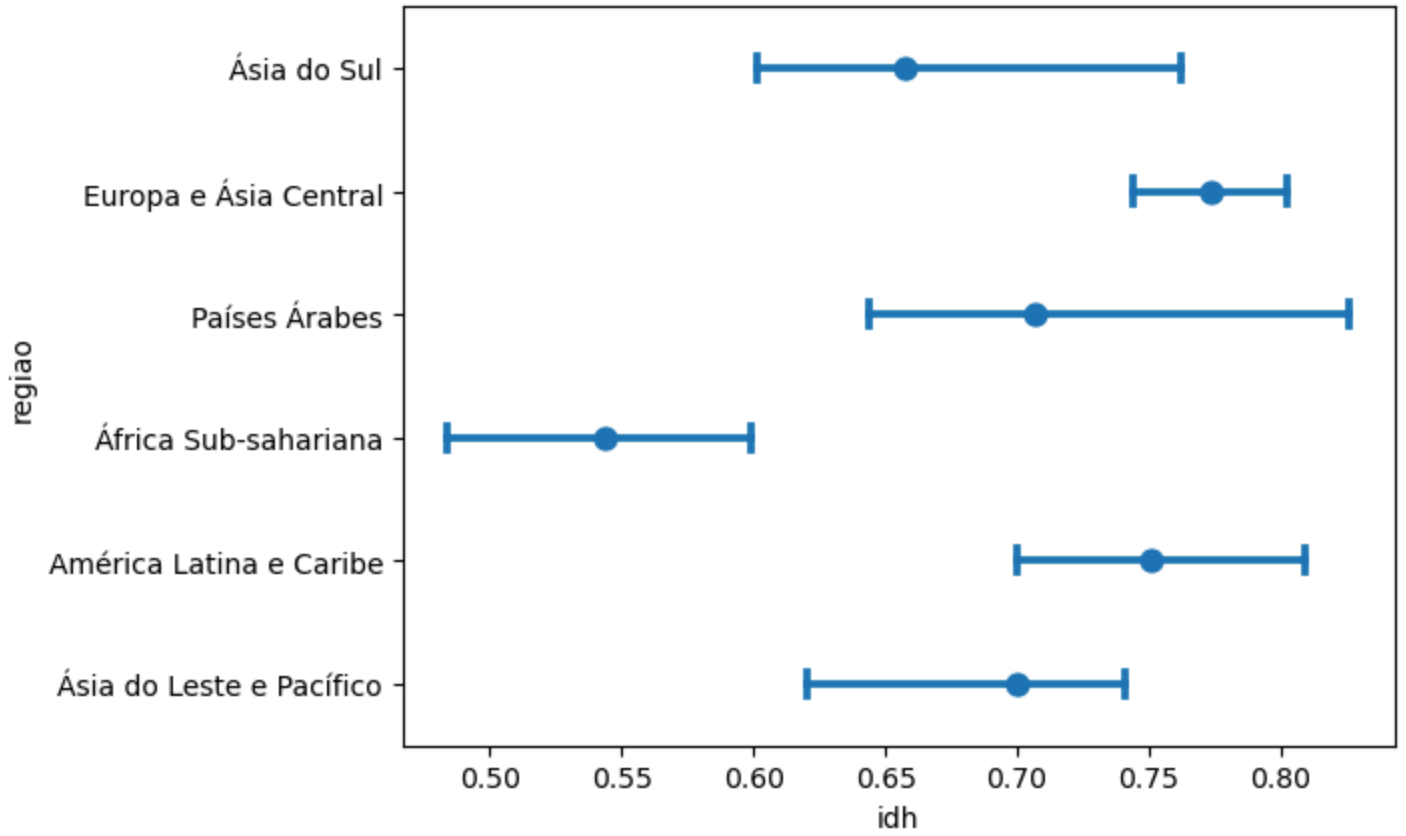
```
errorbar=["pi", 50]
```

Podemos mudar o 50 para o valor desejado entre 0 e 100.

Pointplot

Vamos também exibir o limite da barra de erro usando o argumento `capsize` seguido de um valor numérico:

```
sns.pointplot(
    x="idh",
    y="regiao",
    linestyle="",
    capsize=0.2, # <-----
    errorbar=["pi", 50],
    data=idh_2022
)
```



Distribuições

Vimos muitas formas diferentes de exibir a distribuição de variáveis numéricas: `histplot`, `kdeplot`, `ecdfplot`, `boxplot`, `violinplot`.

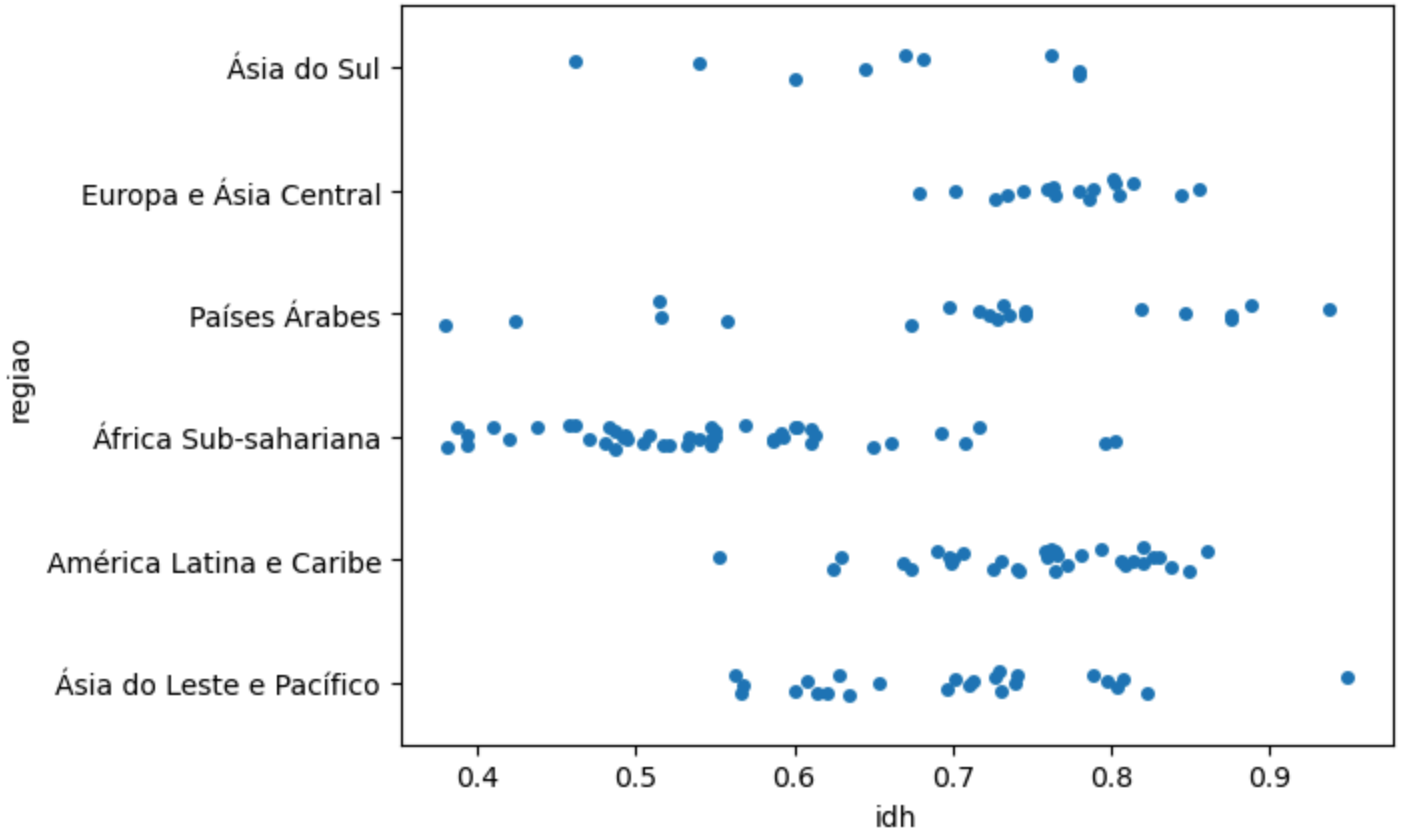
Dessas, apenas quando usamos o `histplot` com contagem podemos passar informação sobre o total de observações. Mas histogramas são difíceis de comparar.

Temos outra alternativa para exibir a distribuição que permitem comparação entre categorias: `swarmplot`, `stripplot`.

Stripplot

Começamos com o `stripplot`. Não temos nenhuma novidade:

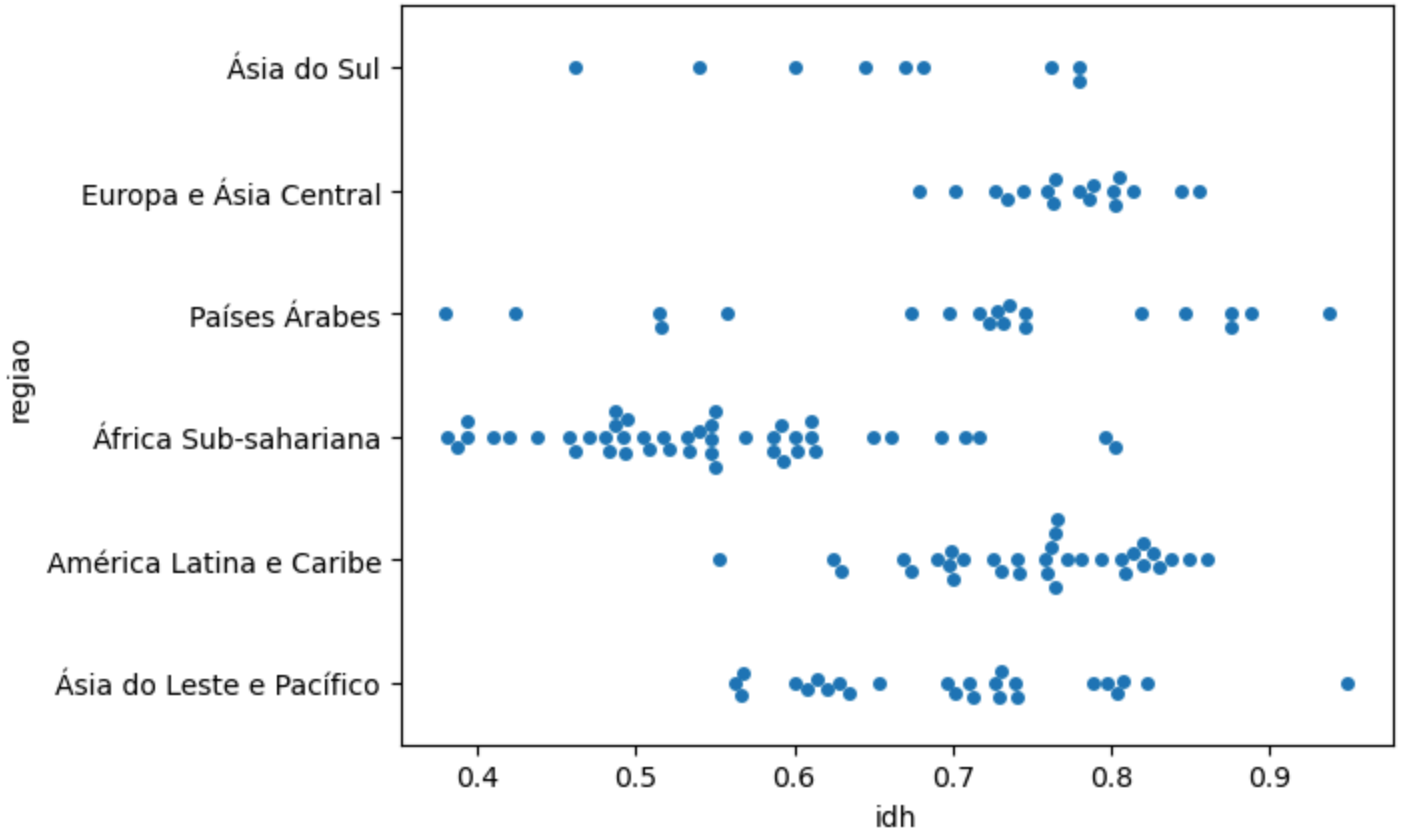
```
sns.stripplot(
    x="idh",
    y="regiao",
    data=idh_2022
)
```



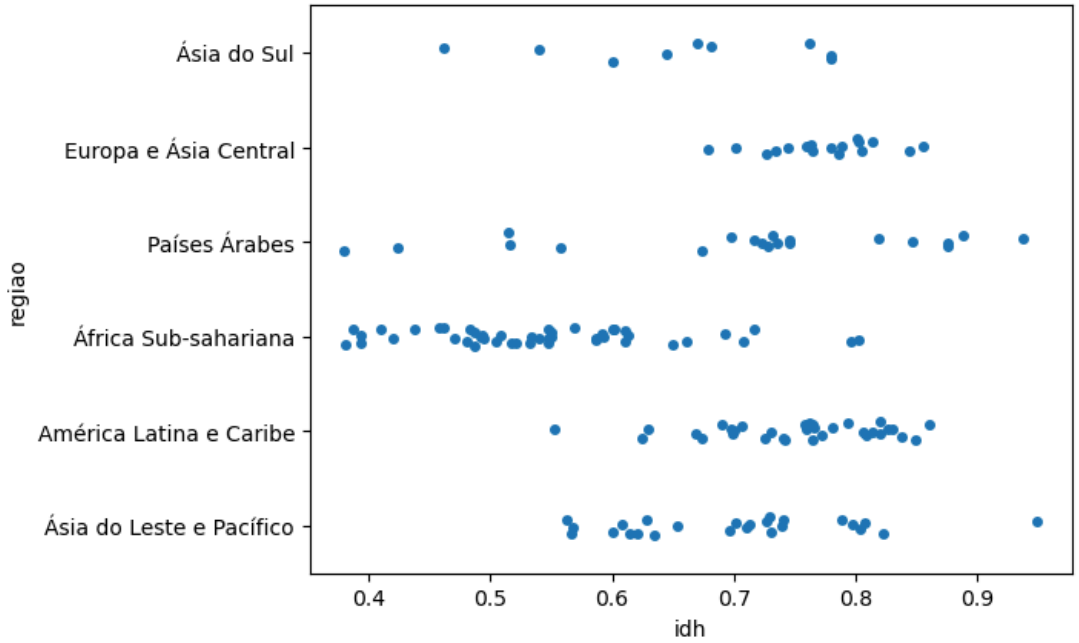
Swarmplot

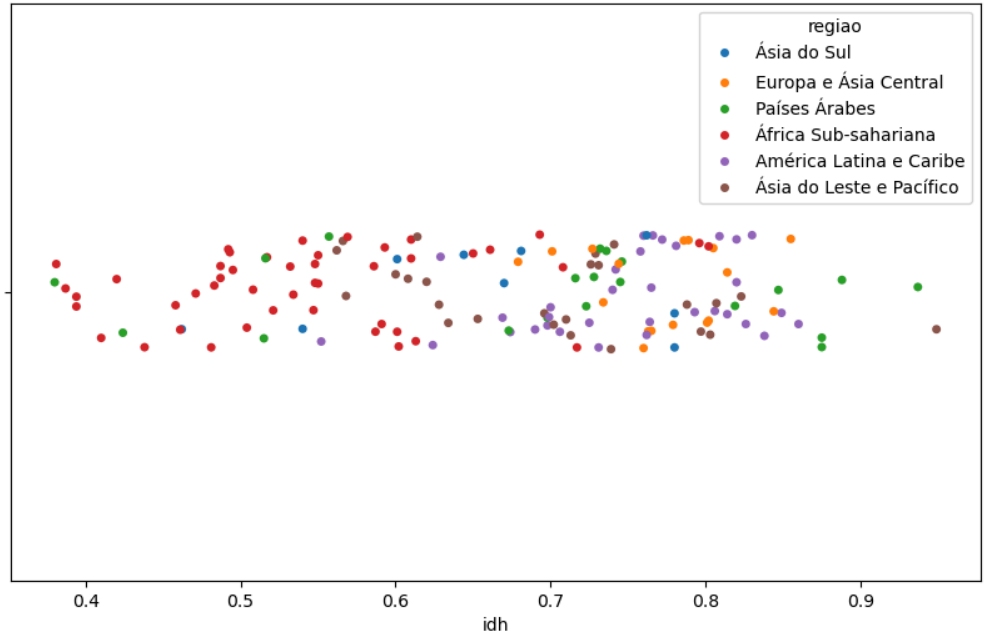
O `swarmplot` é uma variação do `stripplot` que evita a sobreposição de pontos:

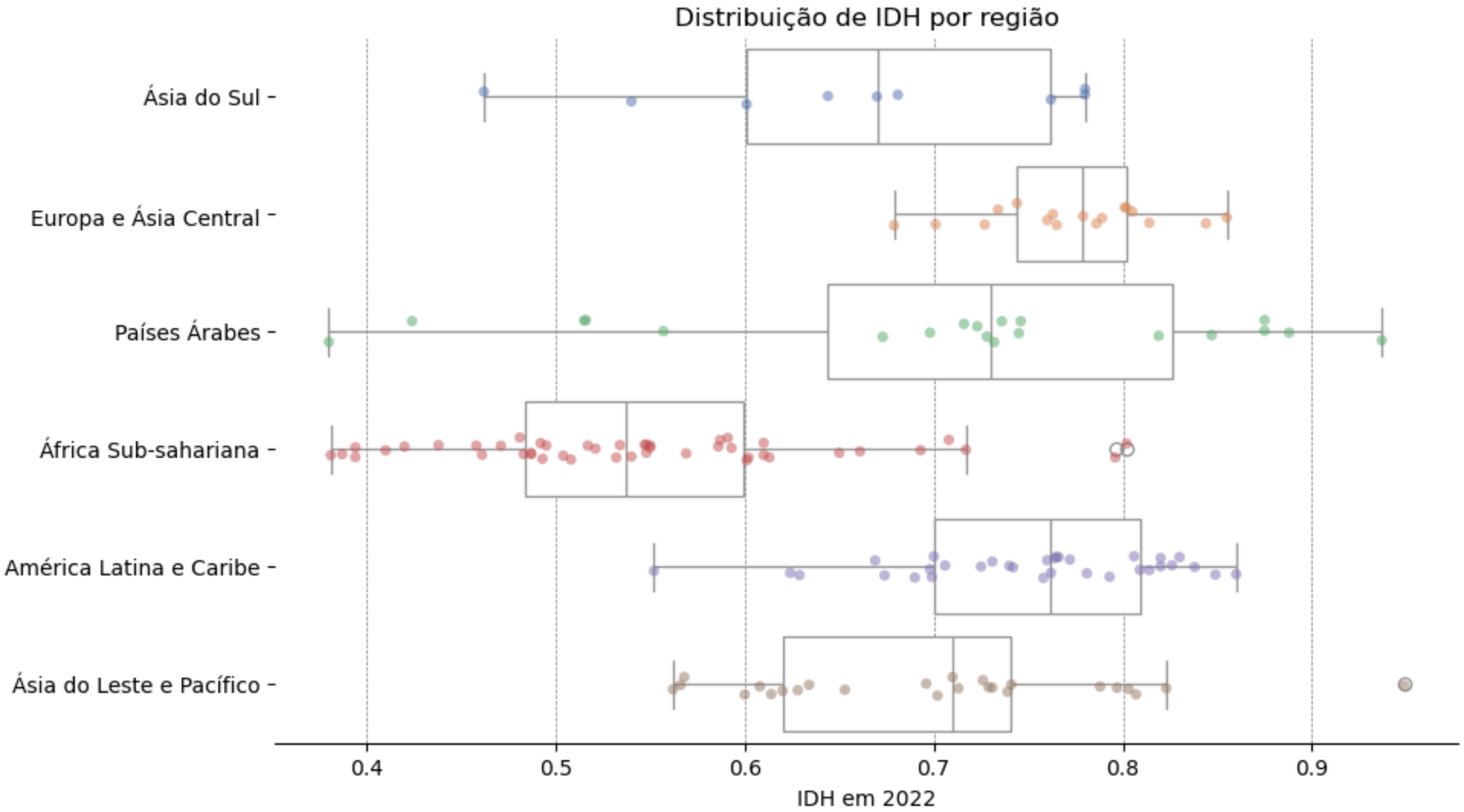
```
sns.swarmplot(
    x="idh",
    y="regiao",
    data=idh_2022
)
```



Hue vs. Dois eixos






Distribuição de IDH por região
Ásia do Sul
Europa e Ásia Central
Países Árabes
África Sub-sahariana
América Latina e Caribe
Ásia do Leste e Pacífico
0.4
0.5
0.6
0.7
0.8
0.9
IDH em 2022

Mãos à obra!