Seaborn and Matplotlib line plots can *look different* even when you're plotting the same data.

🧠 Why They Look Different

⚠️ 1. Seaborn Auto-Sorts the X-Axis

Seaborn's `lineplot()` **automatically sorts** your `x` values **in ascending order**, unless told not to. So even if your data is in random order or unsorted, Seaborn will try to *connect the dots* in increasing x order.

🧪 Example:

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

x = [10, 2, 30, 5]
y = [1, 4, 3, 2]

sns.lineplot(x=x, y=y)  # This will sort x: [2, 5, 10, 30]
plt.show()
```

❌ Matplotlib doesn't do that

Matplotlib's `plt.plot()` will **preserve your original order** — it just draws lines in the exact sequence you give it.

```
plt.plot(x, y)  # Will follow the order: [10, 2, 30, 5]
plt.show()
```

So Seaborn gives you a *clean trendline*, Matplotlib gives you *raw connection paths*.

SMA

✅ 2. Seaborn Applies Smoothing (by default, for categorical or noisy data)

Especially when working with categorical or low-cardinality data, Seaborn may apply some kind of **estimator function** (like `mean`) unless you disable it. But for lineplots, it typically defaults to just plotting raw points — unless grouped.

Still, to be safe, you can turn off all estimation with:

sns.lineplot(x='age', y='children', data=df, estimator=None)

📊 3. Seaborn is built on top of Matplotlib, but adds style and logic

Seaborn is like Matplotlib's smarter, trendier cousin — it auto-applies themes, gridlines, palettes, and tries to "guess" your intent (e.g., trendlines, grouping). That can mess with expectations if you want pixel-perfect control.

🛠️ What to Do

If you want Seaborn to behave exactly like Matplotlib:

sns.lineplot(x='age', y='children', data=df, sort=False)

Add `sort=False` to preserve the original row order, like Matplotlib does.

💥 TL;DR:

- **Seaborn sorts x by default**, Matplotlib doesn't.
- **Seaborn may group and average** if it thinks the data is categorical.
- **Styling differences** also make it feel different (but that's just cosmetics).

SMA

- **Left (Seaborn):** It **auto-sorted the x-axis (`age`)** before plotting. So the line looks smooth, following increasing age.

- **Right (Matplotlib):** It **preserved the order** of the original data as-is — hence the "zig-zag" path.

If you're plotting something like time-series or any order-sensitive data, **go with Matplotlib or use `sort=False` in Seaborn**.