Summary of Q&A - 2025-06-16

Goal

To analyze monthly view shares by channel using a pivot table, compute percentage distributions, and visualize them using a customized horizontal stacked bar chart.

Questions and Answers Summary

1. `channel_totals = trend.sum()` — Why use `.sum()` on a DataFrame?

- `trend` is a pivot table where each column is a channel.
- `.sum()` calculates the **total views per channel across all months**.
- Returns a **Series**, not a DataFrame.

---

2. `groupby().size().reset_index(name='count')` — What does it do?

- `.size()` counts the number of rows per group.
- `.reset_index(name='count')` turns the grouped index into columns and names the count column as `"count"`.

---

3. How to calculate channel percentages per row (monthly)?

You do **not** need a `for` loop. Use vectorized operations:

```
python total_watched_times = trend.sum(axis=1) percent_df = trend.divide(total_watched_times, axis=0) * 100
```

If `trend` includes non-numeric columns, use:

```
python trend_numeric = trend.select_dtypes(include='number')
```

---

4. How to visualize the data as a horizontal stacked bar chart?

```
python percent_df.plot( kind='barh', stacked=True, figsize=(10, 6), width=0.8, colormap='tab20' )
```

Set axis, title, grid, and legend using Matplotlib.

---

5. TypeError: `unsupported operand type(s) for +: 'float' and 'str'`

This happens because some columns in `trend` are not numeric.

Fix:

```
python trend_numeric = trend.select_dtypes(include='number')
```

Then do `.sum(axis=1)` safely.

---

6. Color mismatch in the plot — same channel with different colors?

Problem: Matplotlib assigns new colors if column names are inconsistent or duplicated.

Fix:

- Clean column names: `python trend_numeric.columns = trend_numeric.columns.astype(str).str.strip()`
- Manually assign colors:

```
python custom_colors = { '1          ': '#1f77b4', 'AKA': '#ff7f0e', '
              LIJULIKE': '#2ca02c', '                            ': '#d6272
'#9467bd' } color_list = [custom_colors[col] for col in
percent_df.columns]
```

- Use them in the plot:

```
python percent_df.plot( kind='barh', stacked=True,
color=color_list )
```

Final Tip

Make sure `percent_df.index` is string:

```
python percent_df.index = percent_df.index.astype(str)
```

This avoids display issues when the index is a Period object.

References

- Pandas Documentation: groupby, pivot, sum
- Matplotlib Horizontal Bar Chart