Approach and Challenges: Sunburst Chart Visualization

Approach

1. Requirements Analysis

- Carefully reviewed the requirements to replicate Plotly's Sunburst chart features using Chart.js.
- Identified key features: hierarchical data display, interactive navigation, segment labeling, percentage display, and color management.

2. Technology Selection

- Chose Chart.js for its flexibility, performance, and customization capabilities.
- Decided on a modular React component structure for maintainability and reusability.

3. Component Design

- Designed a main `SunburstChart` component to handle rendering and user interaction.
- Developed a custom hook for navigation and breadcrumb management.
- Created utility functions for data transformation, color assignment, and percentage calculation.

4. User Experience

- Ensured breadcrumbs are always visible for better navigation.
- Implemented smooth transitions and responsive design.
- Focused on clear segment labeling and accurate percentage display.

5. Iterative Development

- Incorporated user feedback to refine features, such as persistent breadcrumbs and improved segment information.
- Made several adjustments to interaction and display logic based on testing and user requests.

Challenges

1. Replicating Plotly Features in Chart.js

- **Challenge:** Plotly offers built-in sunburst support, while Chart.js does not.
- **Solution:** Built custom logic for hierarchical data rendering, navigation, and segment calculation.

2. Breadcrumb Navigation

- **Challenge:** Keeping breadcrumbs always visible and accurately reflecting the current path.
- **Solution:** Developed a custom navigation state and rendering logic to ensure breadcrumbs update correctly on every interaction.

3. Segment Labeling and Percentages

- **Challenge:** Displaying segment names and percentages clearly, especially for small or deeply nested segments.
- **Solution:** Implemented dynamic label positioning, text wrapping, and percentage calculations relative to parent segments.

4. Color Assignment

- **Challenge:** Ensuring distinct and consistent colors for different hierarchy levels.
- **Solution:** Created a color management utility to assign and persist colors based on hierarchy depth and path.

5. User Feedback and Iteration

- **Challenge:** Adapting to evolving requirements and user feedback, such as always-visible breadcrumbs and improved segment info.
- **Solution:** Maintained a flexible codebase, allowing for quick adjustments and feature enhancements.

Conclusion

The project required a creative approach to replicate advanced sunburst chart features in Chart.js. Overcoming the lack of native support, managing navigation, and ensuring a user-friendly experience were key challenges addressed through custom logic and iterative development.