CSC 477- Assignment 5: Interactive Visualization

Monish Shanmugham Suresh

Design Analysis

Overview

This visualization presents a bar chart race of Billboard Hot 100™ songs, focusing on chart longevity rather than weekly peak. Users can switch the ranking metric (Selected Year-to-date, Window total, Lifetime) and the visualization pool (This year only vs Whole file), and control playback (autoplay speed, scrub slider) to examine how long-running hits rise and fall over a calendar year.

Rationale for Design Decisions

Visual encodings

- Position (y) → song identity: Bars are stacked vertically, one lane per song. Position supports quick scanning and stable identity through time.
- Length (x) → ranking value: Length encodes the metric used for ranking (e.g., weeks counted in the selected year). This preserves an intuitive mapping: longer bar ⇒ more weeks.
- Order → descending by selected metric: Each frame sorts songs by the chosen metric so the top performers appear at the top.
- Color → song id (stable per frame): A large palette (≥30 distinct hues) ensures adjacent bars are distinguishable even when an artist has multiple concurrent songs.
- Labels → title by artist with a compact metrics readout (e.g., YTD • Window • Lifetime · #rank) for context without opening details.

Interaction techniques

- Ranking Metric selector: Lets users compare three notions of longevity:
  - Selected Year-to-date (YTD) — weeks counted since Jan 1 of the selected year.
  - Window total — a rolling window carry-in (N weeks before Jan 1) plus YTD.
  - Lifetime — weeks across the entire loaded file (all years).
- Visualization Pool selector:
  - Selected Year — candidates restricted to songs that appear at least once in the selected year.

    - Lifetime — any song from 1958 onward.
- Autoplay + scrub slider: Autoplay shows the year unfolding frame-by-frame; the slider allows precise, non-linear inspection without losing context.
- Dropdowns for Animation Speed / Top N Songs shown / Year / Carry-in: Provide discrete, sensible presets; each change triggers a clean redraw to prevent transition artifacts.

Why these choices

- Bar chart race fits time-varying rankings and conveys cumulative notions cleanly (weeks accrue; bars lengthen).
- Metric + Pool separation clarifies whether we're asking "how much this year?" vs "how much in total?" and whether we're looking only at active titles vs all historical titles in the file.
- Carry-in answers the fairness issue at Jan 1 (some songs enter the year mid-run).
- Accessible tooltips (newline rendering, keyboard focus) improve clarity without crowding the UI.

Alternatives considered

- Stacked bars (carry-in base + YTD overlay): more precise decomposition but visually heavy in motion; rejected for clutter.
- Line/area charts per song: great for histories but scales poorly with many items; loses ranking clarity.
- Sankey/Alluvial of rank bands: compelling, but over-complex for the core question of "who has the most weeks?"

Data Processing & Definitions

- Dataset: CSV located at ./src/assets/hot-100-current.csv (project repository). Columns: chart_week, current_week, title, performer, last_week, peak_pos, wks_on_chart.
- Normalization:
    - Song id = title — normalizeArtist(performer) where features/joins are canonicalized (e.g., feat./ft./with/x, commas/ampersands) to avoid splitting the same title across credit variants.
- Derived values:
    - Selected Year-to-date (YTD): count of weeks for a song from Jan 1 of the selected year through the current frame.
    - Carry-in (N weeks): weeks accrued in the N weeks before Jan 1 of the selected year (used only by Window total).
    - Window total: carry-in + YTD.
    - Lifetime (in file): total weeks across all years present in the loaded CSV.

- Framing: Timeline is weekly; frames are constrained to Week 1 → Week 52 of the first year in view to avoid cross-year spillover.

Animation & Performance

- Clean remounts on control changes (e.g., Bars shown) prevent mid-transition artifacts.
- Width clamping and color fallback avoid invisible bars when the candidate set changes.
- Stable x-axis within the selected year preserves comparability across frames.

Accessibility & Color

- Palette: Start with large categorical palettes (Tableau, Set3, Paired, Set2, Dark2) and extend with evenly spaced hues to reach ≥30 distinct colors; fallback hue for unknowns.
- Text contrast: Labels use dark text over mid-tone fills; value labels sit outside bars to minimize overlap.
- Tooltips: Implemented with white-space: pre-line (newline support), keyboard focus, and role="tooltip".

Development Process

- Team & roles: Single-developer project (design, data processing, D3/React implementation, QA).
- Timeline (people-hours) (replace with your actuals):
    - Ideation & sketching: ~2–3 h
    - Data inspection & parsing: ~2 h
    - Development and Testing: ~8 h
    - Palette tuning, bug fixes: ~2–3 h
    - Write-up & polish: ~1 h
- Biggest time sinks:
    - Ensuring smooth transitions during dynamic Top-N changes (hard remounts + clamping).
    - Reconciling carry-in vs YTD and keeping pool/metric semantics intuitive.
    - Avoiding color repetition while maintaining legibility with many concurrent bars.

References & External Resources

- Data: Billboard Hot 100 CSV (project asset at ./src/assets/hot-100-current.csv).
- APIs/Libs: D3 (scales, transitions, axes), d3-scale-chromatic (categorical schemes), React + Vite.

- Inspiration/Patterns: Bar-chart-race patterns popularized in the D3 community and comparable examples on Observable; general guidance on animated ranking visuals.