ShinyConf 2025

Death By Dropdown?

A Developer's Guide To Building Dashboards That Won't Fry Your Client's Brain

Milena Eickhoff & Jeremy Winget, PhD

SKIM

The Problem

Ever felt like this?

- Endless dropdown menus
- Overwhelming complexity
- Users disengaged
- Delayed decision-making
- “Dashboard rot”

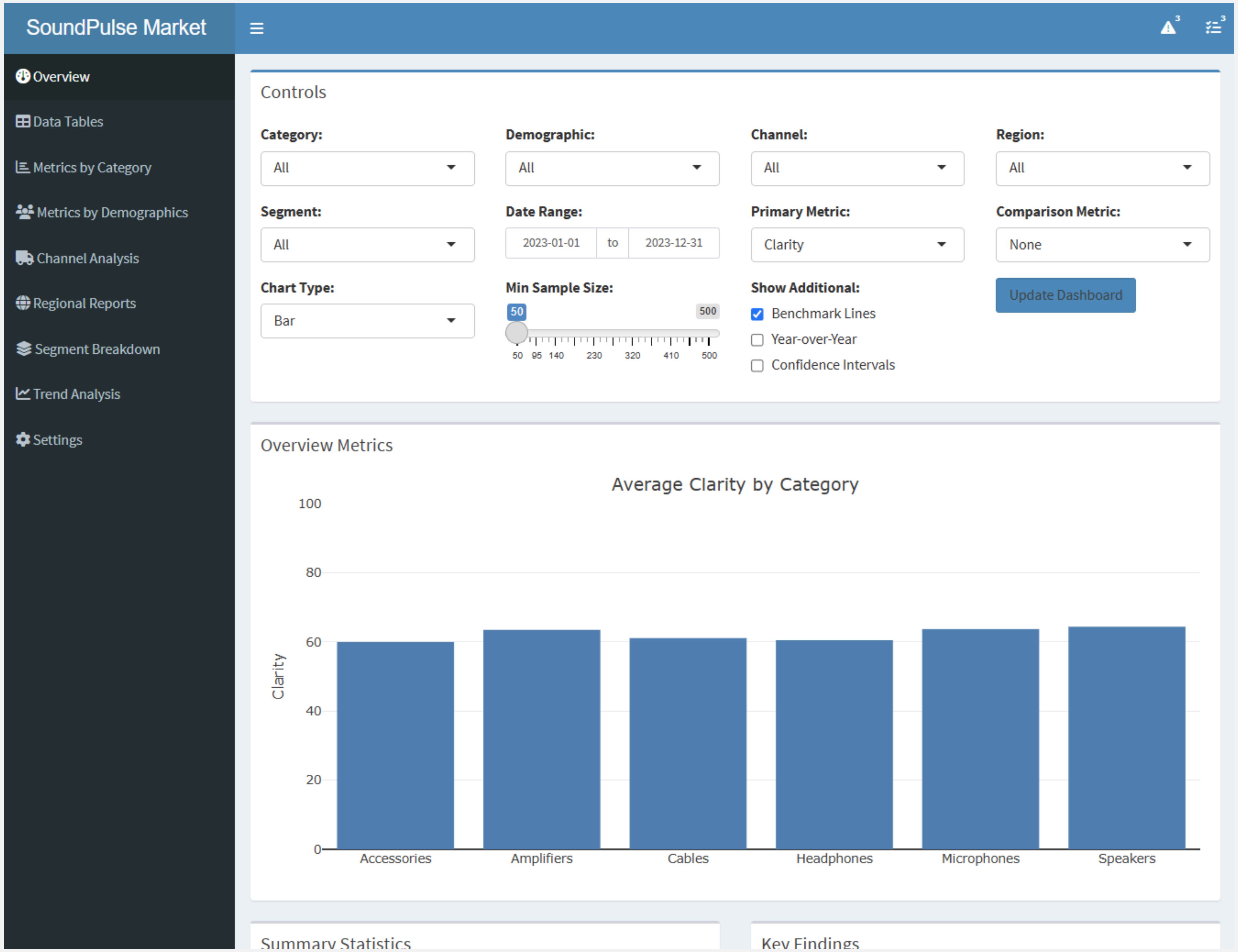
It Doesn’t Have To Be This Way

By shifting the focus from **cramming data** to **crafting stories**

SoundPulse’s market research team was drowning in data, unable to extract meaningful insights from their complex dashboards.

SOUNDPULSE

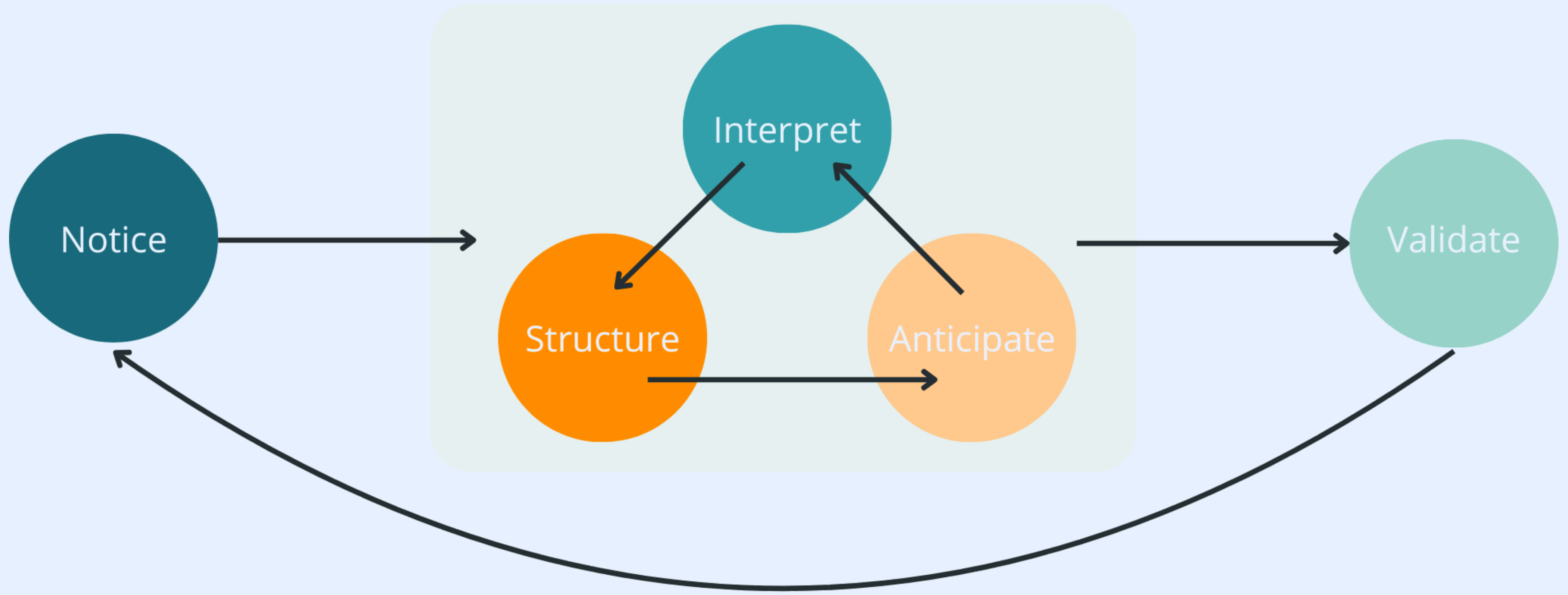
BID In Action: Dashboard Transformation



Controls

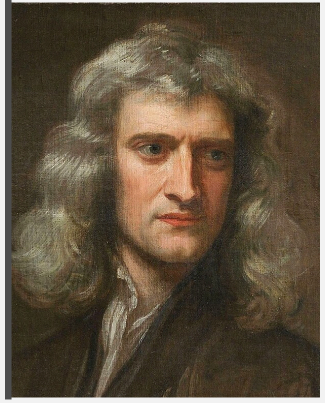
Category: All
Demographic: All
Channel: All
Region: All
Segment: All
Date Range: 2023-01-01 to 2023-12-31
Primary Metric: Clarity
Comparison Metric: None
Chart Type: Bar
Min Sample Size: 50 (50 – 500)
Show Additional:
[✓ Benchmark Lines]
Year-over-Year
Confidence Intervals

Behavior Insight Design (BID) Framework



1. **Notice** the Problem
2. **Interpret** the User's Need
3. **Structure** the Dashboard
4. **Anticipate** User Behavior
5. **Validate** & Empower the User

Where BID Fits In



> "If I have seen further, it is by standing on the shoulders of giants."
>
> — Isaac Newton

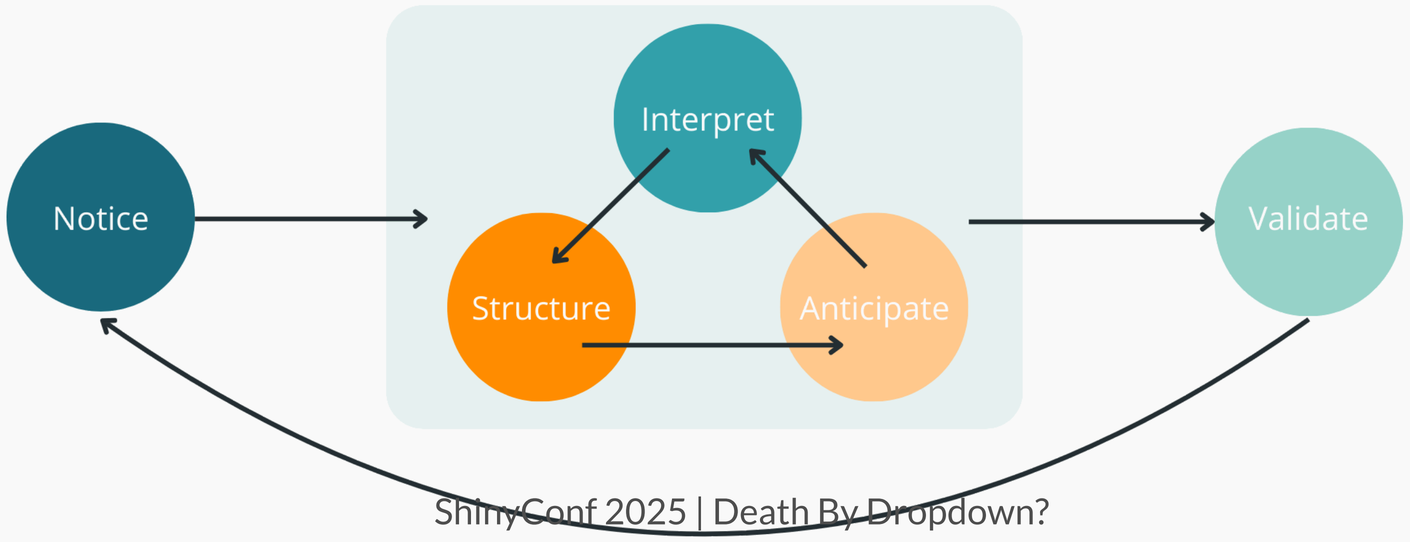
- **UX Design**: Established best practices
- **Data Storytelling**: From data dumps to meaningful narratives
- **Psychological Science**: Evidence-based cognitive principles
- **Visual Communication**: Optimized for information perception

Key Differentiator

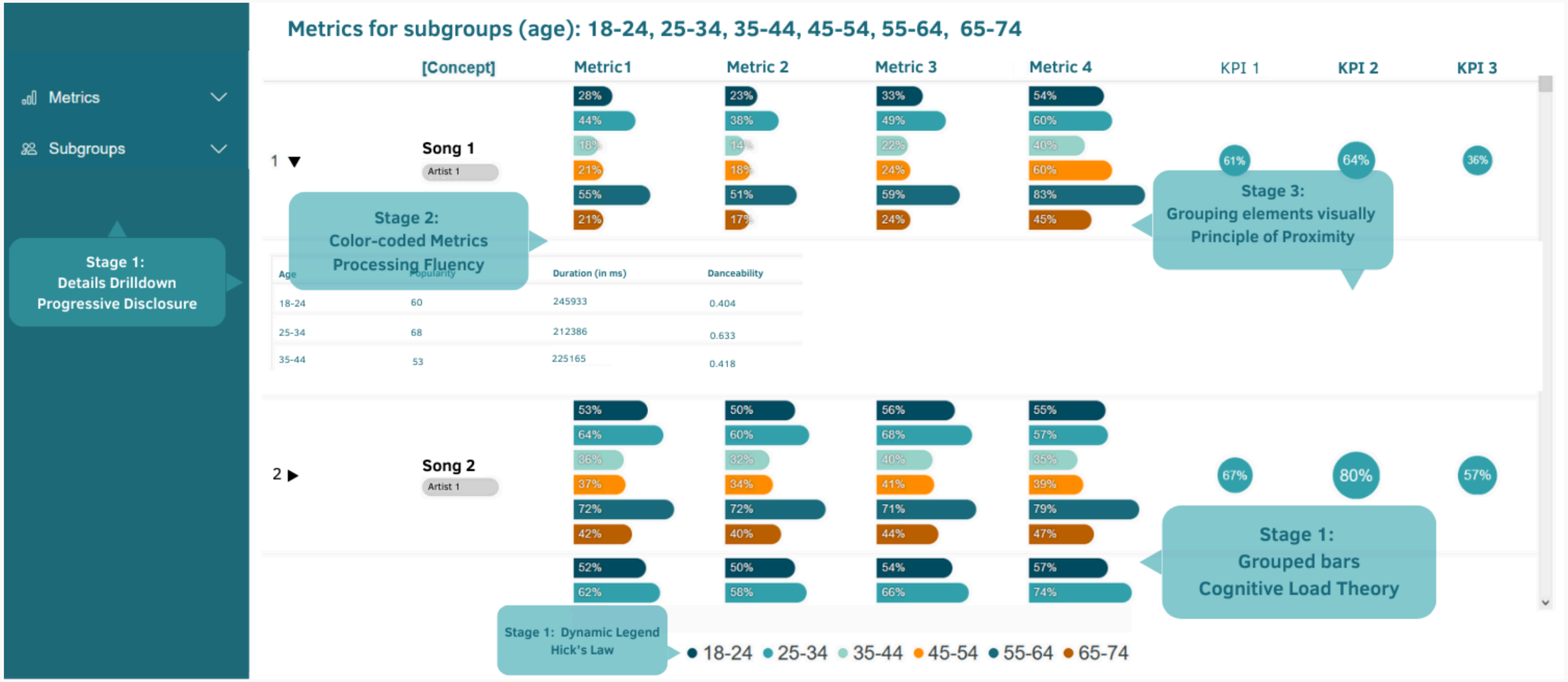
Systematic integration of behavioral science within the natural dashboard development workflow

SoundPulse Case Study: BID Stages 1-3

- **Stage 1: Notice**
  - Complex filtering overwhelmed users (*Cognitive Load*)
  - 72% users reported confusion (*Hick's Law*)
- **Stage 2: Interpret**
  - Intuitive visualizations (*Processing Fluency*)
  - Clearer key insights (*Data Storytelling*)
- **Stage 3: Structure**
  - Group related elements (*Principle of Proximity*)
  - Prioritization of metrics (*Dual-Processing Theory*)



{reactable} Implementation: Visual Result



Metrics for subgroups (age): 18-24, 25-34, 35-44, 45-54, 55-64, 65-74

	[Concept]	Metric1	Metric 2	Metric 3	Metric 4	KPI 1	KPI 2	KPI 3
1 ▼	Song 1 Artist 1	28% 44% 18% 21% 55% 21%	23% 38% 14% 18% 51% 17%	33% 49% 22% 24% 59% 24%	54% 60% 40% 60% 83% 45%	61%	64%	36%
2 ▶	Song 2 Artist 1	53% 64% 36% 37% 72% 42%	50% 60% 32% 34% 72% 40%	56% 68% 40% 41% 71% 44%	55% 57% 35% 39% 79% 47%	67%	80%	57%
		52% 62%	50% 58%	54% 66%	57% 74%			

Age	Popularity	Duration (in ms)	Danceability
18-24	60	245933	0.404
25-34	68	212386	0.633
35-44	53	225165	0.418

● 18-24 ● 25-34 ● 35-44 ● 45-54 ● 55-64 ● 65-74

{reactable} Implementation: Basics

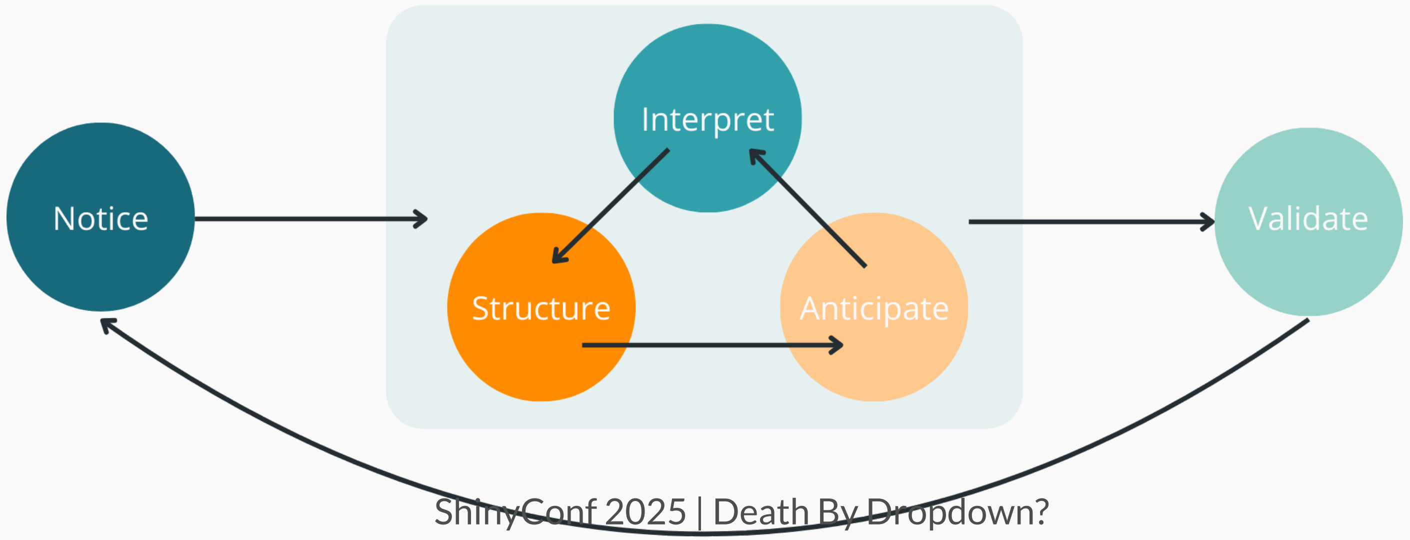
```r
# Packages: reactable, dplyr, bslib
reactable(
  soundpulse_data,
  # Stage 1: Reduce cognitive load with grouped structure
  groupBy = "Song",
  # Stage 2: Enhance processing fluency with color coding
  columns = list(
    `Metric 1` = colDef(
      style = function(value) {
        # Color coding for instant comprehension
        case_when(
          value > 60 ~ list(background = "#1c6d7d", color = "white"),
          value > 40 ~ list(background = "#35a4ae", color = "white"),
          TRUE ~ list(background = "#ffca90", color = "black")
        )
      }
    )
  ),
```

{reactable} Implementation: Advanced

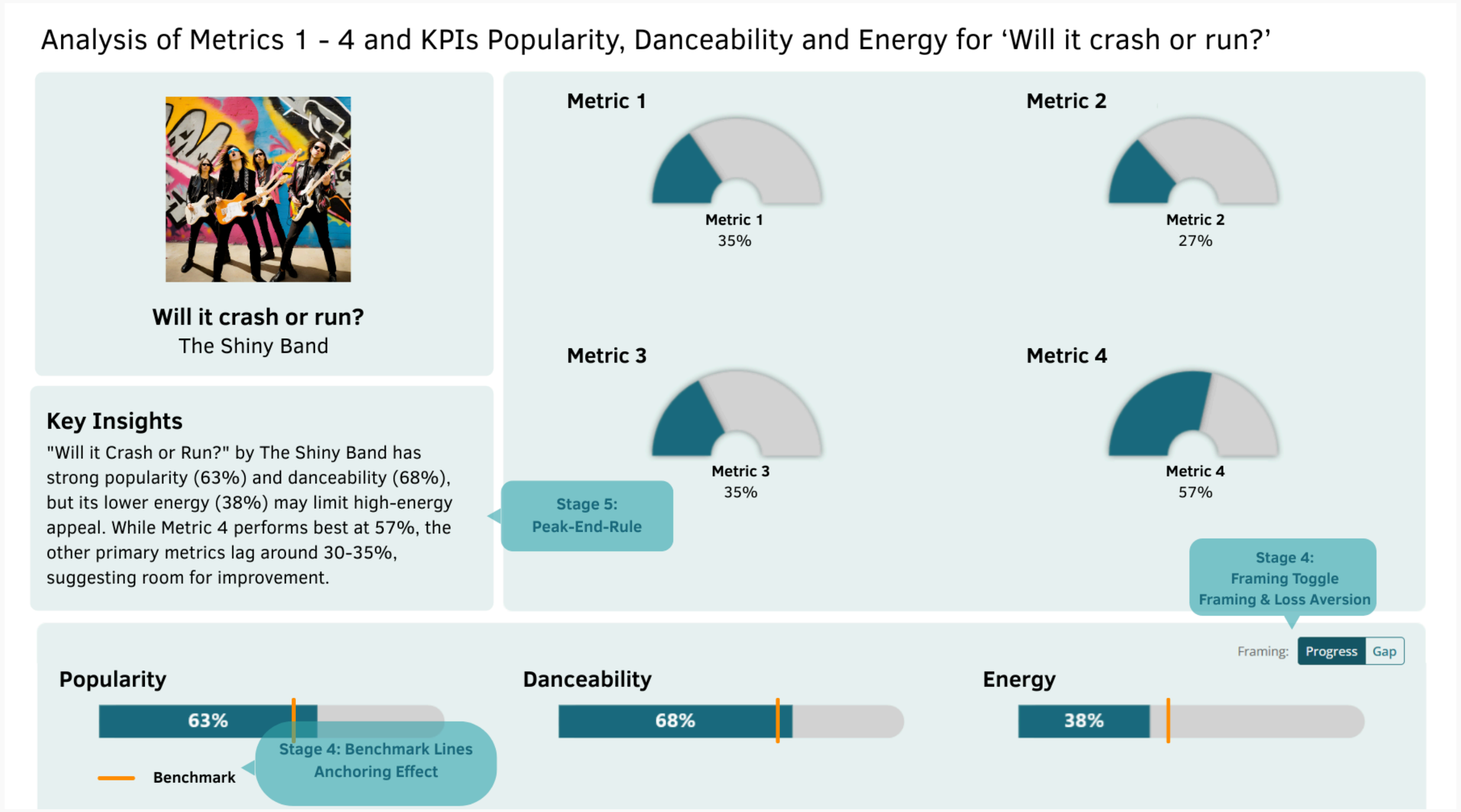
```r
# Packages: reactable, purrr
# Define age group colors for consistency
age_colors <- c(
  "18-24" = "#1c6d7d",
  "25-34" = "#35a4ae",
  "35-44" = "#98d3ca",
  "45-54" = "#ffca90",
  "55-64" = "#1c6d7d",
  "65-74" = "#dd8500"
)

# Create the table with dynamic legend component
div(
  # Main table component
  reactable(
    soundpulse_data,
    # Stage 3: Group related elements (Principle of Proximity)
    columnGroups = list(
```

SoundPulse Case Study: BID Stages 4-5

- Stage 4: Anticipate
  - Executives comparing metrics without context (*Anchoring Effect*)
  - Different perspectives needed for different teams (*Framing & Loss Aversion*)
- Stage 5: Validate
  - Teams needed actionable summaries (*Peak-End Rule*)



{echarts4r} Implementation: Visual Result



Analysis of Metrics 1 - 4 and KPIs Popularity, Danceability and Energy for 'Will it crash or run?'

Will it crash or run?
The Shiny Band

Key Insights

"Will it Crash or Run?" by The Shiny Band has strong popularity (63%) and danceability (68%), but its lower energy (38%) may limit high-energy appeal. While Metric 4 performs best at 57%, the other primary metrics lag around 30-35%, suggesting room for improvement.

{echarts4r} Implementation: Gauge Chart

```r
# Packages: echarts4r, bslib
# Create intuitive gauge chart (Stage 2: Processing Fluency)
create_gauge_chart <- function(value, title) {
  e_charts() |>
    e_title(title) |>
    e_gauge(
      value,
      startAngle = 180,
      endAngle = 0,
      detail = list(formatter = "{value}%"),
      axisLine = list(
        lineStyle = list(
          color = list(
            c(value/100, "#1c6d7d"),  # Filled portion
            c(1, "#e9e9e9")           # Empty portion
          ),
          width = 30
        )
```

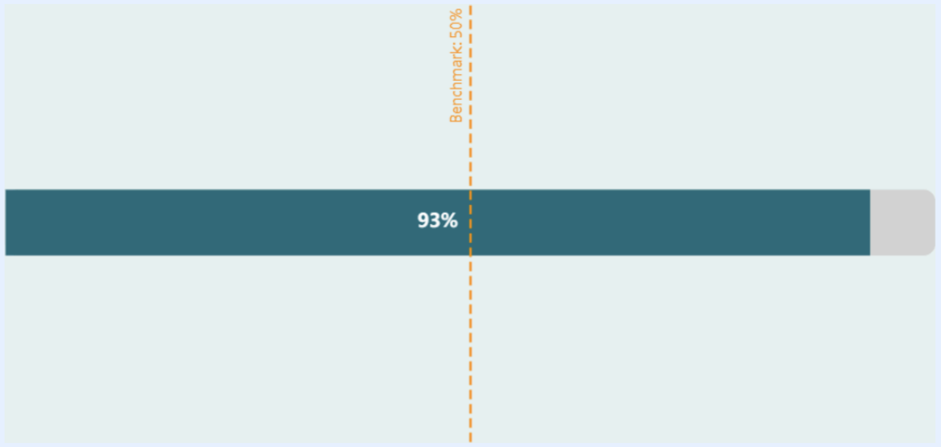
{echarts4r} Implementation: Context

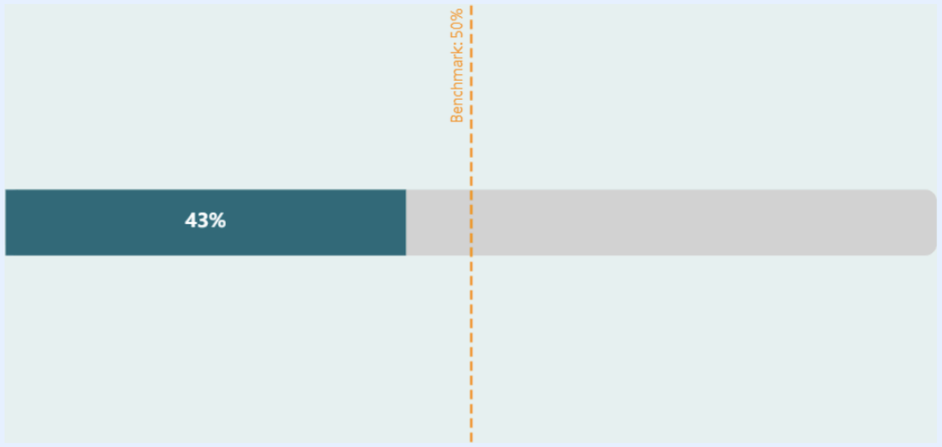
```r
# Packages: echarts4r, bslib
# Create benchmark bar (Stage 4: Anchoring Effect)
create_benchmark_bar <- function(value, title, benchmark = 50) {
  e_charts() |>
    e_title(title) |>
    e_bar(
      value,
      legend = list(show = FALSE),
      showBackground = TRUE,
    ) |>
    e_flip_coords() |>
    e_labels(position = "inside") |>
    e_y_axis(show = FALSE) |>
    e_x_axis(show = FALSE, min = 0, max = 100) |>
    e_hide_grid_lines() |>
    e_mark_line(
      data = list(yAxis = benchmark),
      lineStyle = list(color = "#dd8500", type = "dashed", width = 2),
```

{echarts4r} Implementation: Framing

```r
# Packages: shiny, echarts4r, dplyr, purrr, bslib
# UI with framing toggle (Stage 4: Framing & Loss Aversion)
ui <- fluidPage(
  titlePanel("SoundPulse Dashboard"),

  # Stage 4: Framing toggle
  radioButtons(
    "framing", "Framing:",
    choices = c("Progress" = "progress", "Gap" = "gap"),
    selected = "progress",
    inline = TRUE
  ),

  # Layout for charts using programmatically generated outputs
  layout_column_wrap(
    width = 1 / 3,
    card(echarts4rOutput("chart_popularity")),
    card(echarts4rOutput("chart_danceability")),
```

How Framing Affects Decision Making





- **Progress (gain) Framing:**
  - Motivates by highlighting achievement
  - Creates positive momentum
  - Supports incremental improvement

- **Gap (loss) Framing:**
  - Creates urgency to address shortfall
  - Highlights areas needing improvement
  - May trigger risk-avoidance behaviors

From Framework to Workflow: {bidux}

- **Current Features (Phase 1):**
  - 📚 Concept browser
    - `bid_concept("processing fluency")`
  - 🧠 BID stage functions
    - `bid_notice()` to `bid_validate()`

- **Coming Soon (Phases 2-4):**
  - 🤖 LLM Integration
  - 🛠 UI Component Library
  - 📊 Testing and Validation Tools

What {bidux} Looks Like Today

Notice | Concepts | Chained Stages | Components

```r
library(bidux)

# BID Notice stage
notice_stage <- bid_notice(
  problem = "Users overwhelmed by filtering options and dropdown menus",
  # Theory parameter is optional - will auto-suggest appropriate theory if empty
  evidence = "Feedback shows 65% of users abandon the dashboard after first use"
)

notice_stage$theory
```

```
[1] "Hick's Law"
```

```r
notice_stage$suggestions
```

```
[1] "Reduce dropdown options or use hierarchical menus for better organization."
```

More at: github.com/jrwinget/bidux

Real-World Impact: SoundPulse Results

> "The BID framework transformed how we approach our dashboards. What once took weeks of refinement now has clear direction from day one."
>
> — Maya Chen, SoundPulse Research Director

- Dramatically faster decision-making
- Substantial increase in insights per session
- Stakeholder satisfaction transformed from frustration to enthusiasm
- Significantly streamlined implementation of new metrics

Key Takeaways

1. **BID** helps reduce friction and improve decision-making
2. `{reactable}` + `{echarts4r}` help bring BID to life
3. `{bidux}` supports you in applying BID at every stage (try it today!)
4. Build dashboards that guide — not fry — your users

Thank you!

Milena Eickhoff

Jeremy Winget, PhD

Let's connect on LinkedIn!

Slides available at our GitHub repo

SKIMgroup.com

SKIM