Margin-Bleed from Freight Ratios (Profitability)

Language: PostgreSQL

```postgresql
/*
   BUSINESS QUESTION:
   "In which states and product categories is shipping cost eating the
   largest share of the item price — and how much total freight are we
   subsidizing on low-price items?"

   WHY THIS MATTERS:
   A R$30 item with R$20 freight (67% ratio) likely has zero margin after
   logistics. This query identifies the state × category combinations
   where freight ratios are highest, letting the pricing team decide
   where to raise minimum order thresholds, negotiate carrier rates, or
   introduce free-shipping-over-X promotions.
*/

WITH delivered_base AS (
    SELECT
        c.customer_unique_id,
        o.order_id,
        o.order_purchase_timestamp
    FROM olist_customers_dataset        AS c
    INNER JOIN olist_orders_dataset      AS o
        ON o.customer_id = c.customer_id
    WHERE o.order_status = 'delivered'
),

/* ■■ Geographic: most recent state per customer ■■ */
dim_geo AS (
    SELECT customer_unique_id, customer_state
    FROM (
        SELECT
            c.customer_unique_id,
            c.customer_state,
            ROW_NUMBER() OVER (PARTITION BY c.customer_unique_id
                               ORDER BY o.order_purchase_timestamp DESC) AS rn
        FROM olist_customers_dataset    AS c
        INNER JOIN olist_orders_dataset  AS o ON o.customer_id = c.customer_id
        WHERE o.order_status = 'delivered'
    ) ranked WHERE rn = 1
),

/* ■■ Line-item level freight analysis (aggregated by state × category) ■■
```

```sql
   This is intentionally NOT at the customer level — we want the
   state × category grain to see where the margin bleed is happening.
   The fan-out here is controlled because this is the ONLY join path
   (delivered_base → items → products → translation) and we aggregate
   directly to the final grain.                                          */
freight_by_state_category AS (
    SELECT
        geo.customer_state,
        COALESCE(t.product_category_name_english, 'unknown')
                                                    AS product_category,
        COUNT(*)                                    AS total_line_items,
        ROUND(AVG(oi.price)::NUMERIC, 2)            AS avg_item_price,
        ROUND(AVG(oi.freight_value)::NUMERIC, 2)    AS avg_freight,
        ROUND(AVG(
            oi.freight_value / NULLIF(oi.price, 0)
        )::NUMERIC, 4)                              AS avg_freight_ratio,
        ROUND(SUM(oi.freight_value)::NUMERIC, 2)    AS total_freight_cost,
        ROUND(SUM(oi.price)::NUMERIC, 2)            AS total_item_revenue,

        /* Count items where freight > 30% of item price — the "red zone"  */
        SUM(CASE WHEN oi.freight_value / NULLIF(oi.price, 0) > 0.30
                 THEN 1 ELSE 0 END)                 AS items_in_red_zone,
        ROUND(100.0 *
            SUM(CASE WHEN oi.freight_value / NULLIF(oi.price, 0) > 0.30
                     THEN 1 ELSE 0 END)::NUMERIC
            / COUNT(*)
        , 1)                                        AS pct_items_in_red_zone

    FROM delivered_base                             AS db
    INNER JOIN dim_geo                              AS geo
        ON geo.customer_unique_id = db.customer_unique_id
    INNER JOIN olist_order_items_dataset             AS oi
        ON oi.order_id = db.order_id
    INNER JOIN olist_products_dataset                AS pr
        ON pr.product_id = oi.product_id
    LEFT JOIN product_category_name_translation      AS t
        ON t.product_category_name = pr.product_category_name
    GROUP BY geo.customer_state,
             COALESCE(t.product_category_name_english, 'unknown')
)

/* ■■ FINAL: Top margin-bleed hotspots ■■ */
SELECT
    customer_state                                  AS state,
    product_category,
    total_line_items,
    avg_item_price,
```

```
    avg_freight,
    avg_freight_ratio,
    total_freight_cost,
    total_item_revenue,
    items_in_red_zone,
    pct_items_in_red_zone
FROM freight_by_state_category
WHERE total_line_items >= 30          /* ← Statistical minimum for meaningful ratios */
ORDER BY avg_freight_ratio DESC
LIMIT 50;
```