

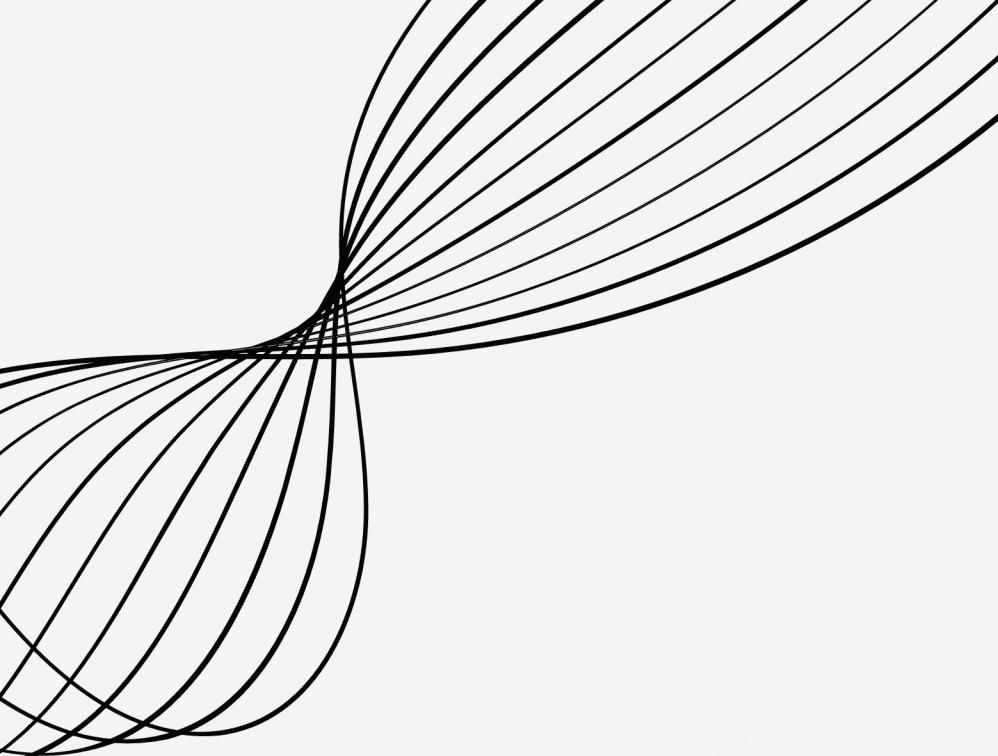


NIKE VS ADIDAS

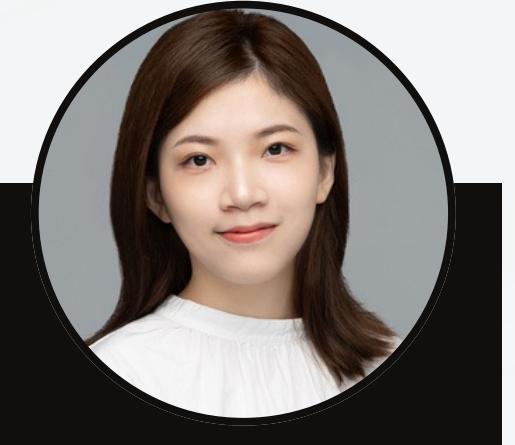
Just Do It vs. Impossible Is Nothing

CONTENT

- 
- 
- 01 ABOUT US
 - 02 DATA DESCRIPTION
 - 03 DATA ANALYSIS
 - 04 STRATEGIC VALUE & RECOMMENDATIONS
 - 05 APPENDIX



PURPLE TEAM 4



Peifen



Shalaka



Kelly



Tony



Rish

WHY THIS DATASET?



Examining data of sports retail to strategize & enhance revenue and profit.



As product analysts, our exploration will analyse pricing, reviews, descriptions, discount trends, ratings, along with revenue and website traffic.



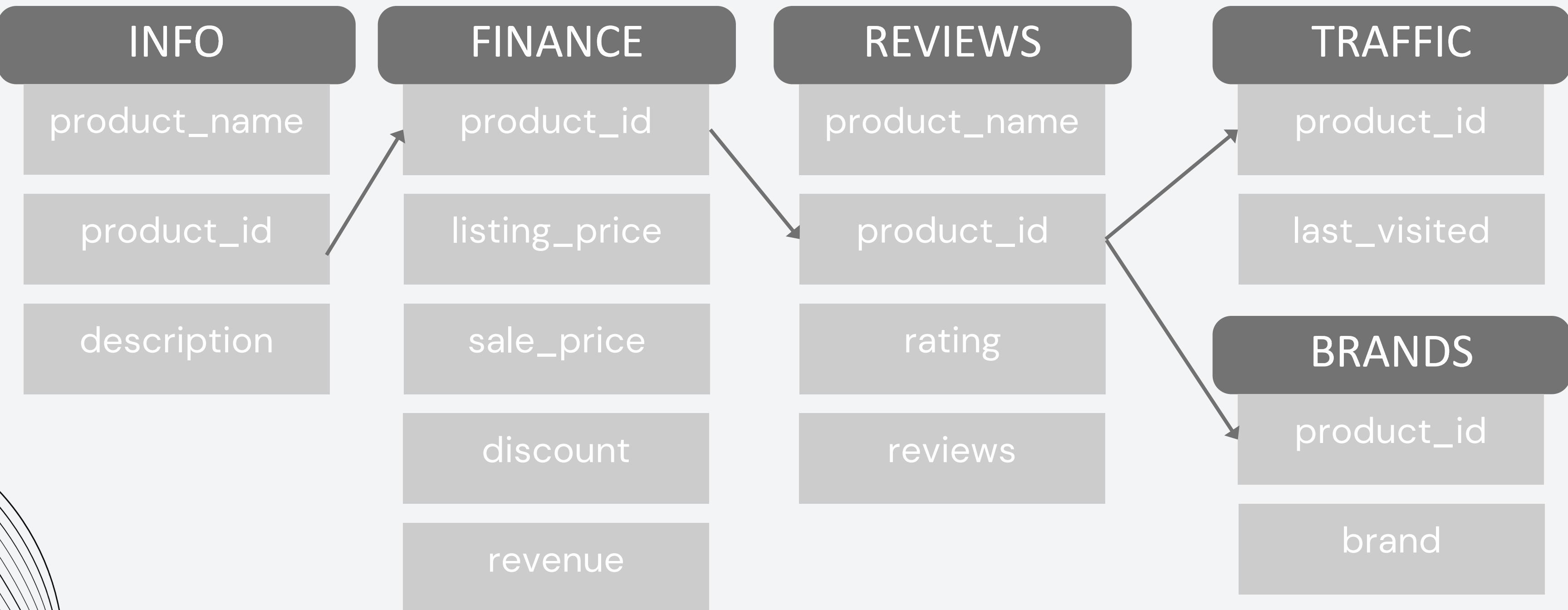
Our findings will help in generating recommendations & strategies for marketing and sales teams.

CHALLENGES



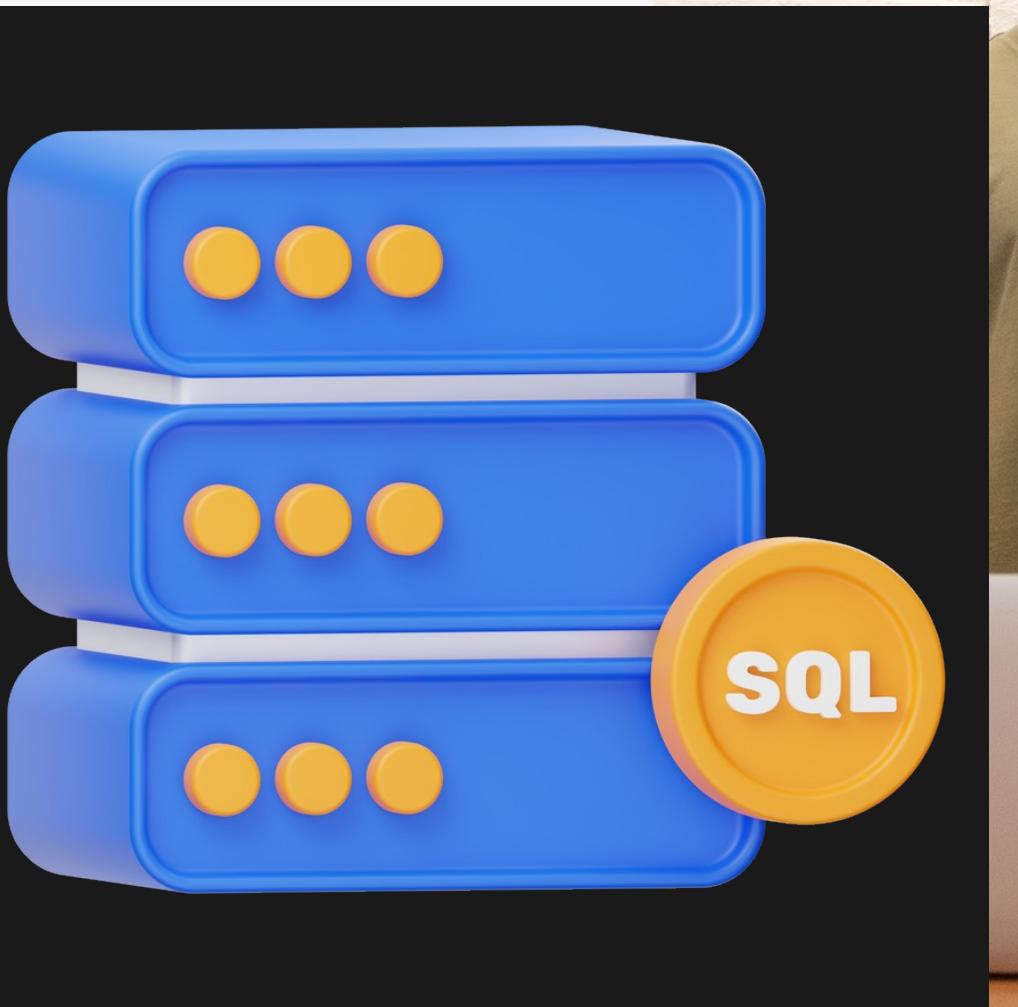
The dataset exhibits minimal missing values, accounting for less than 5% of its overall data, rendering additional processing unnecessary.

DESCRIPTION OF DATASET



The Database provided to us, SPORTS, contains five tables, with `product_id` being the primary key for all of them:

DATA ANALYSIS



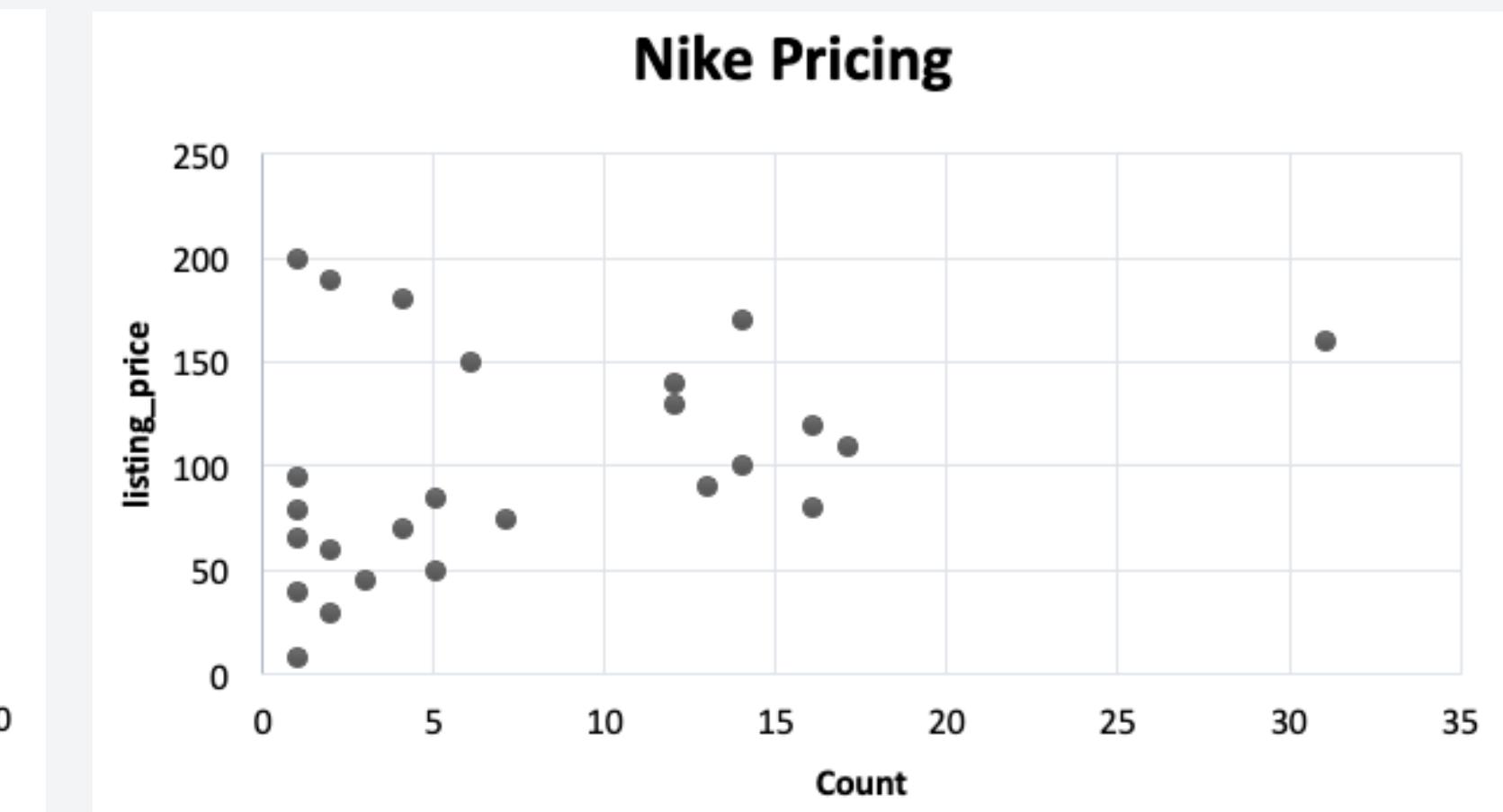
Counting Missing Values

We can see the database contains 3,179 products in total. Of the columns we previewed, only one — `last_visited` — is missing more than five percent of its values.

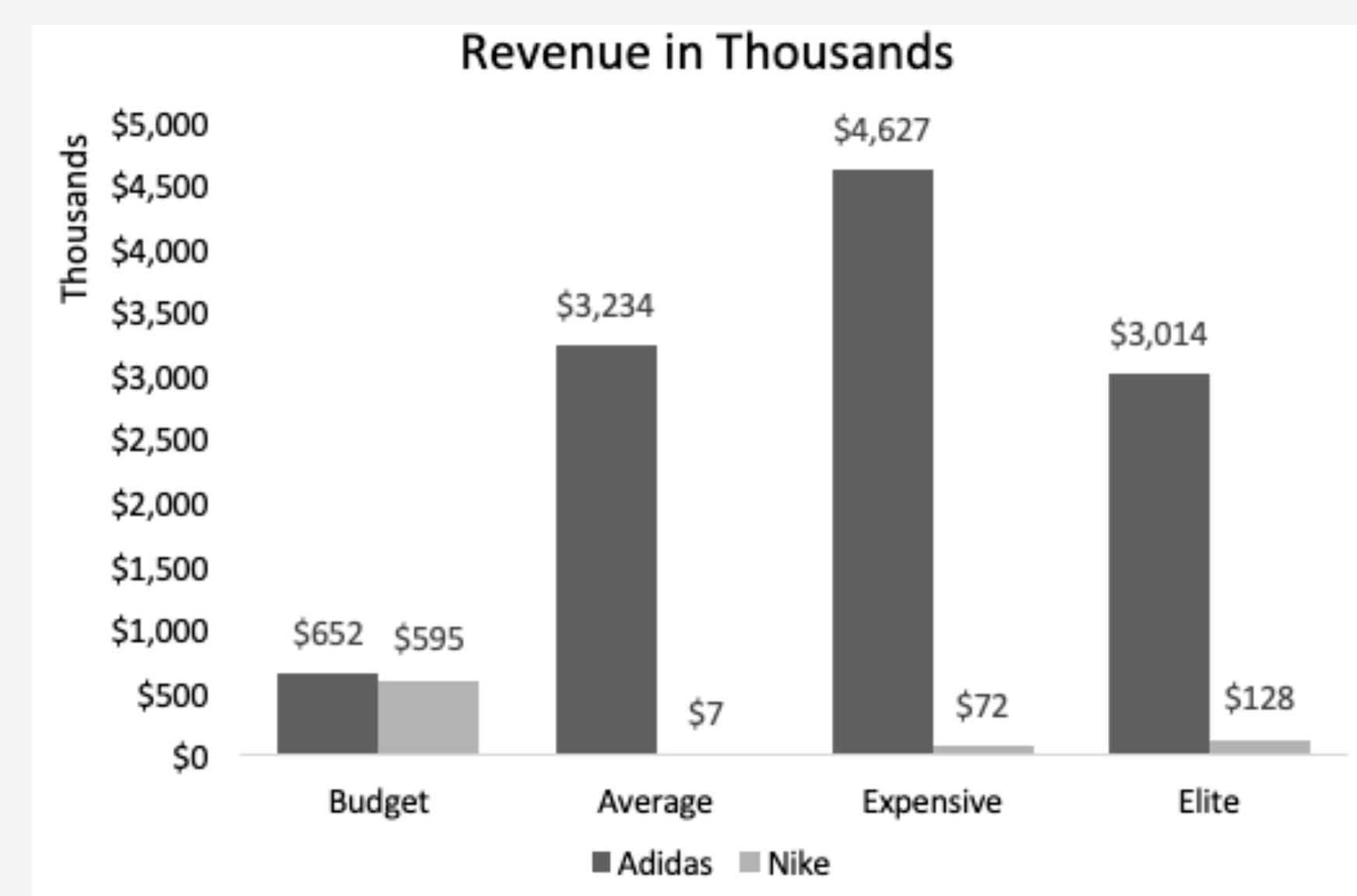
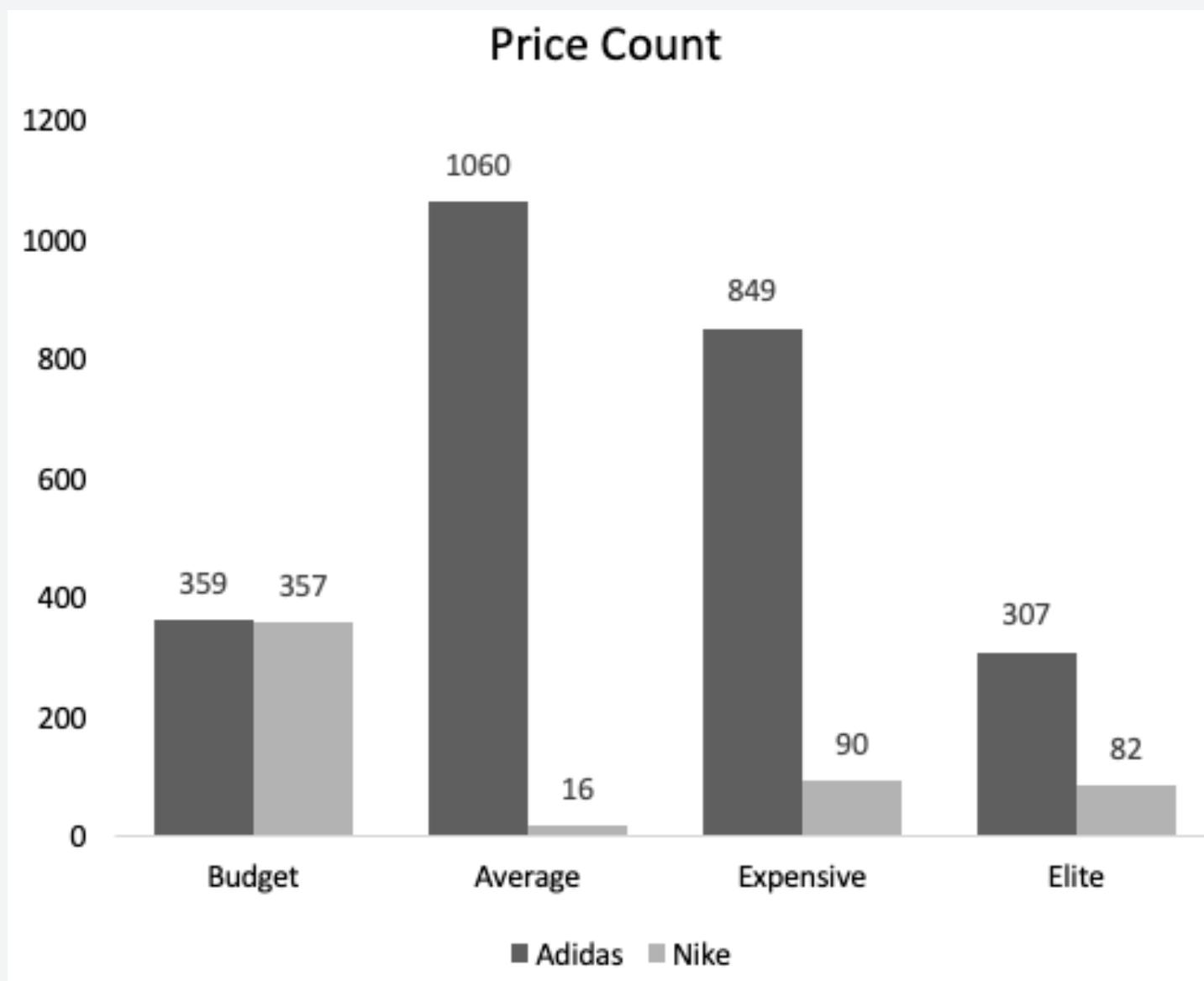
<code>total_rows</code>	<code>count_description</code>	<code>count_listing_price</code>	<code>count_last_visited</code>
3179	3117	3120	2928

Nike vs. Adidas Pricing

- Distribution of the listing_price and the count for each price, grouped by brand
- 77 unique prices



Price Ranges



DISCOUNT BY BRAND

NIKE



WE DON'T DO ANY DISCOUNT



ADIDAS

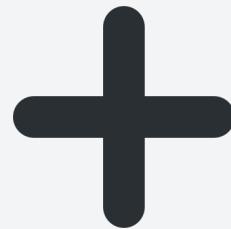


We offer an average discount of
33.45%!



KEY TAKEWAY

REVENUE



REVIEWS

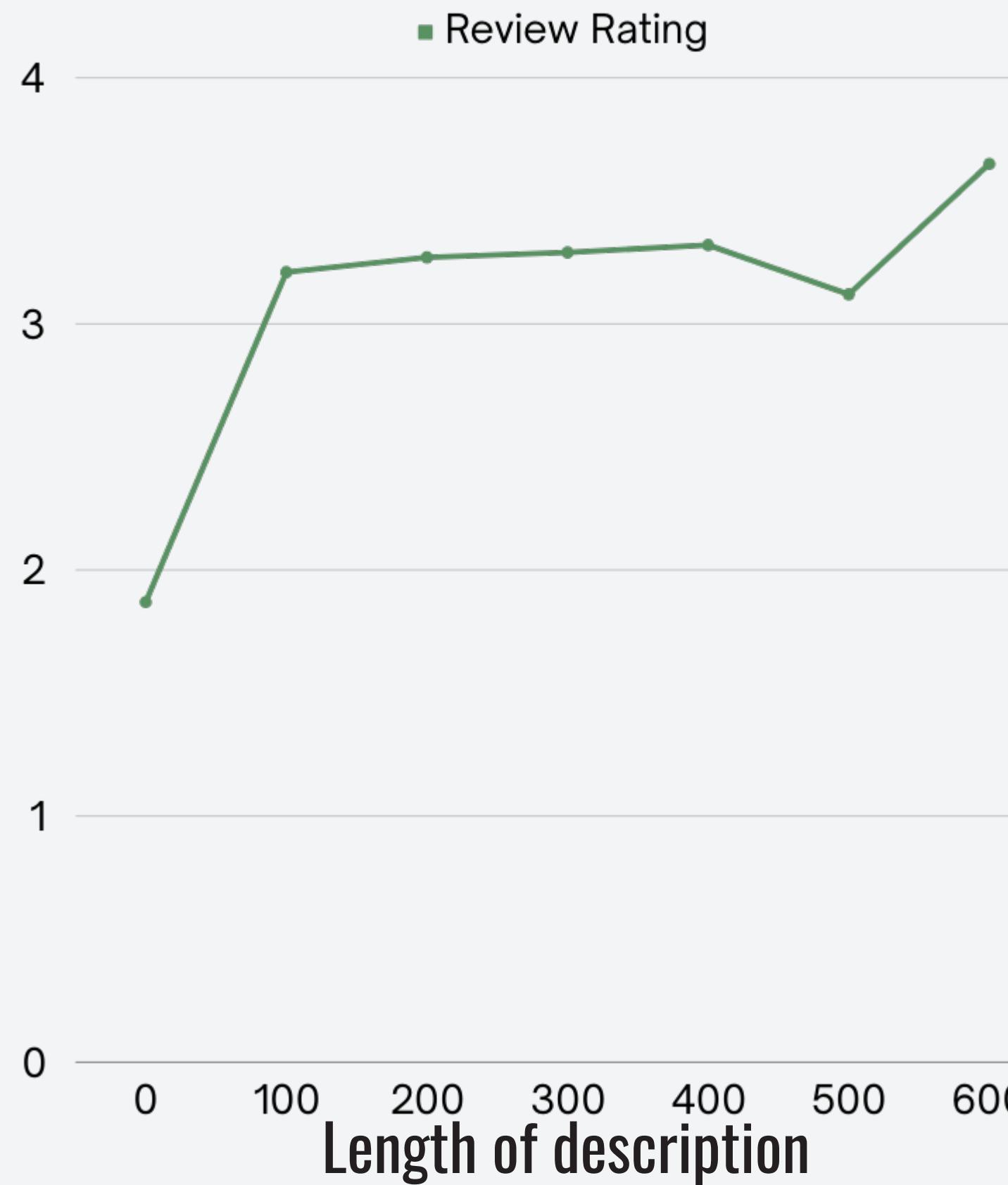
CORRELATION =

0.6519



- Pretty strong correlation
- Potential to increase sales with a larger number of reviews

WORD COUNT MATTERS?



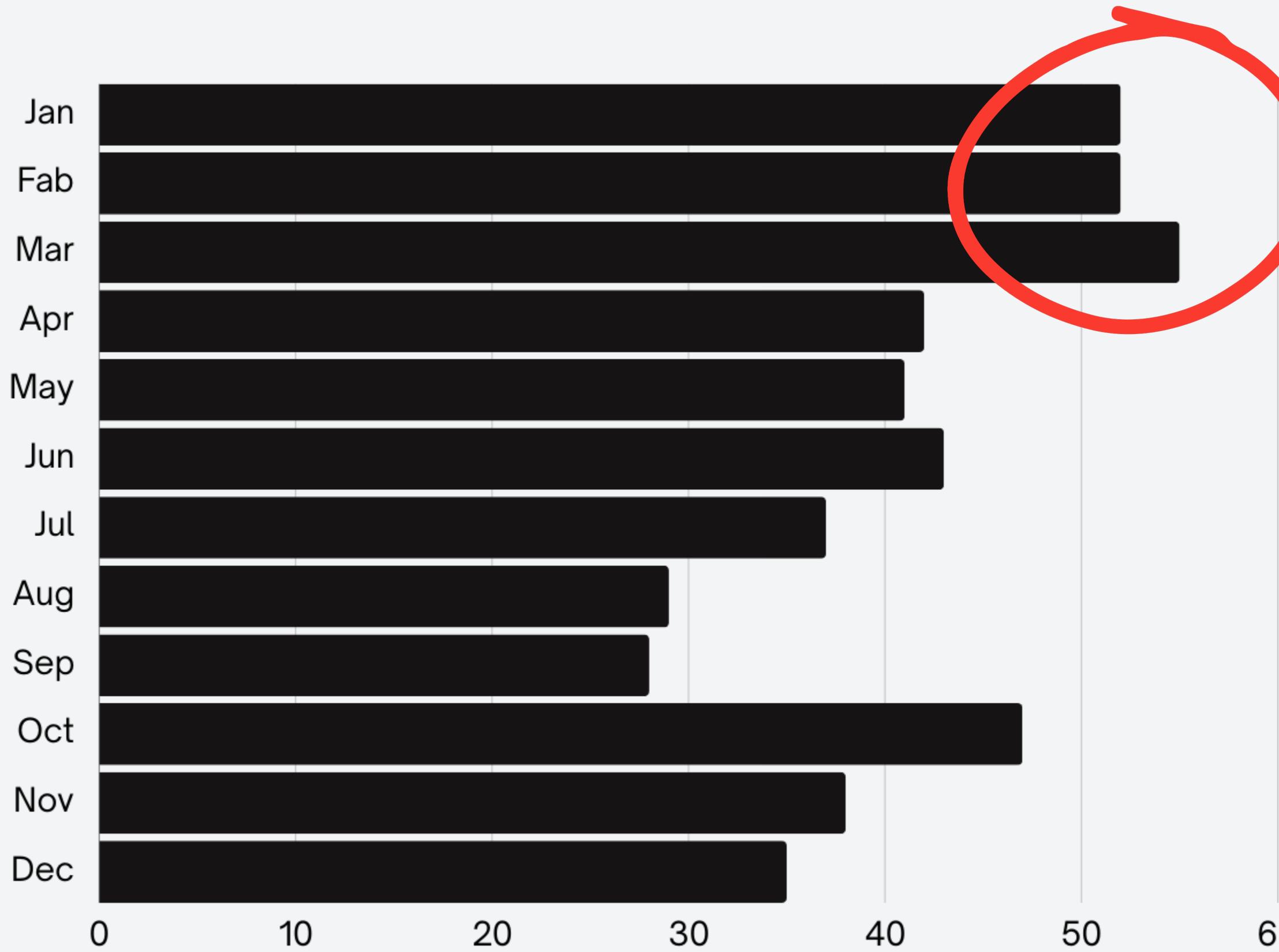
No clear pattern

Between the length of the product's
description and its rating.

ADIDAS REVIEWS BY MONTH

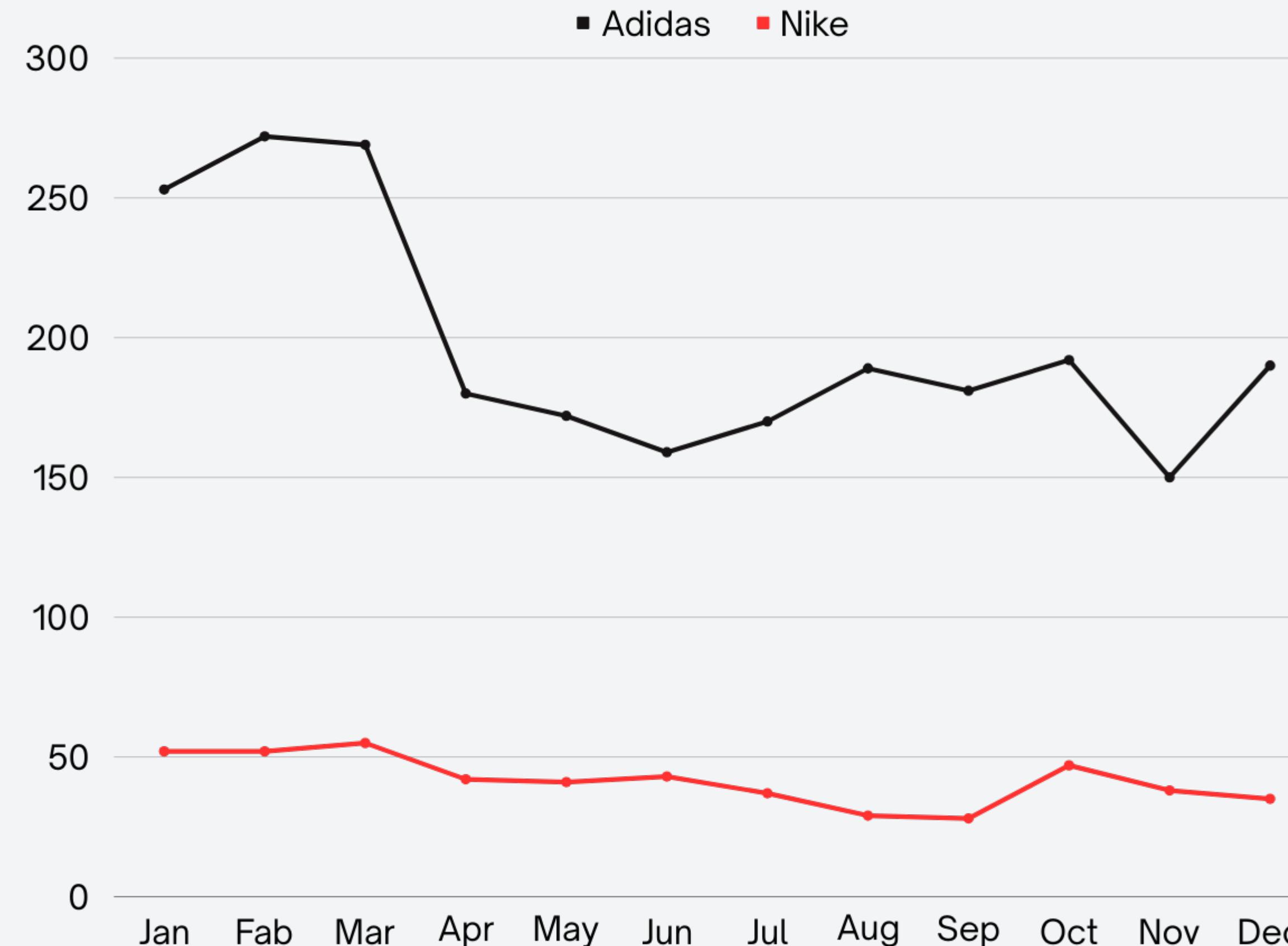


NIKE REVIEWS BY MONTH

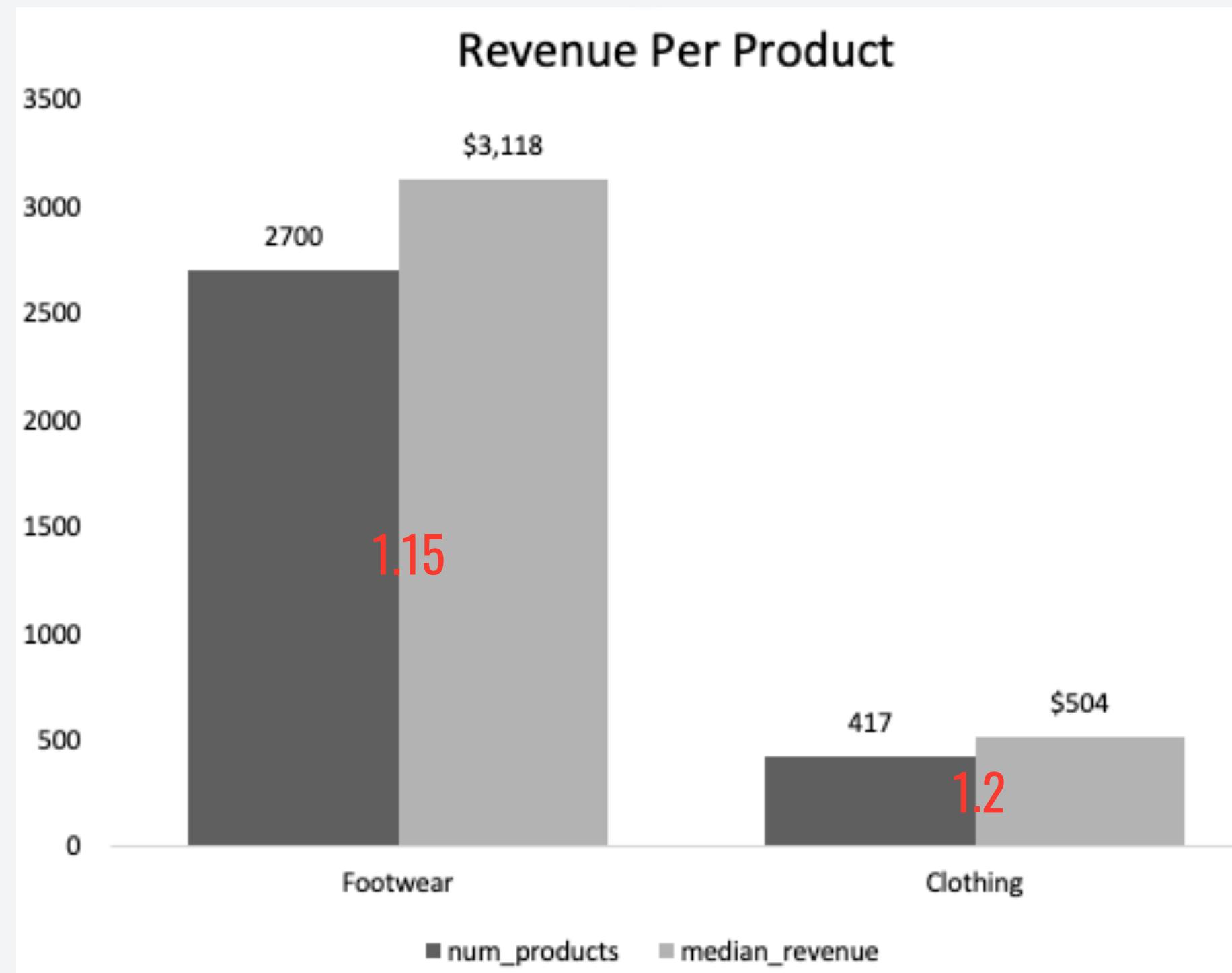


Same principle here, highest reviews in the first quarter of the year

SIDE BY SIDE COMPARISON



Product Performance



STRATEGIC VALUE

Decision makers:
Discounts

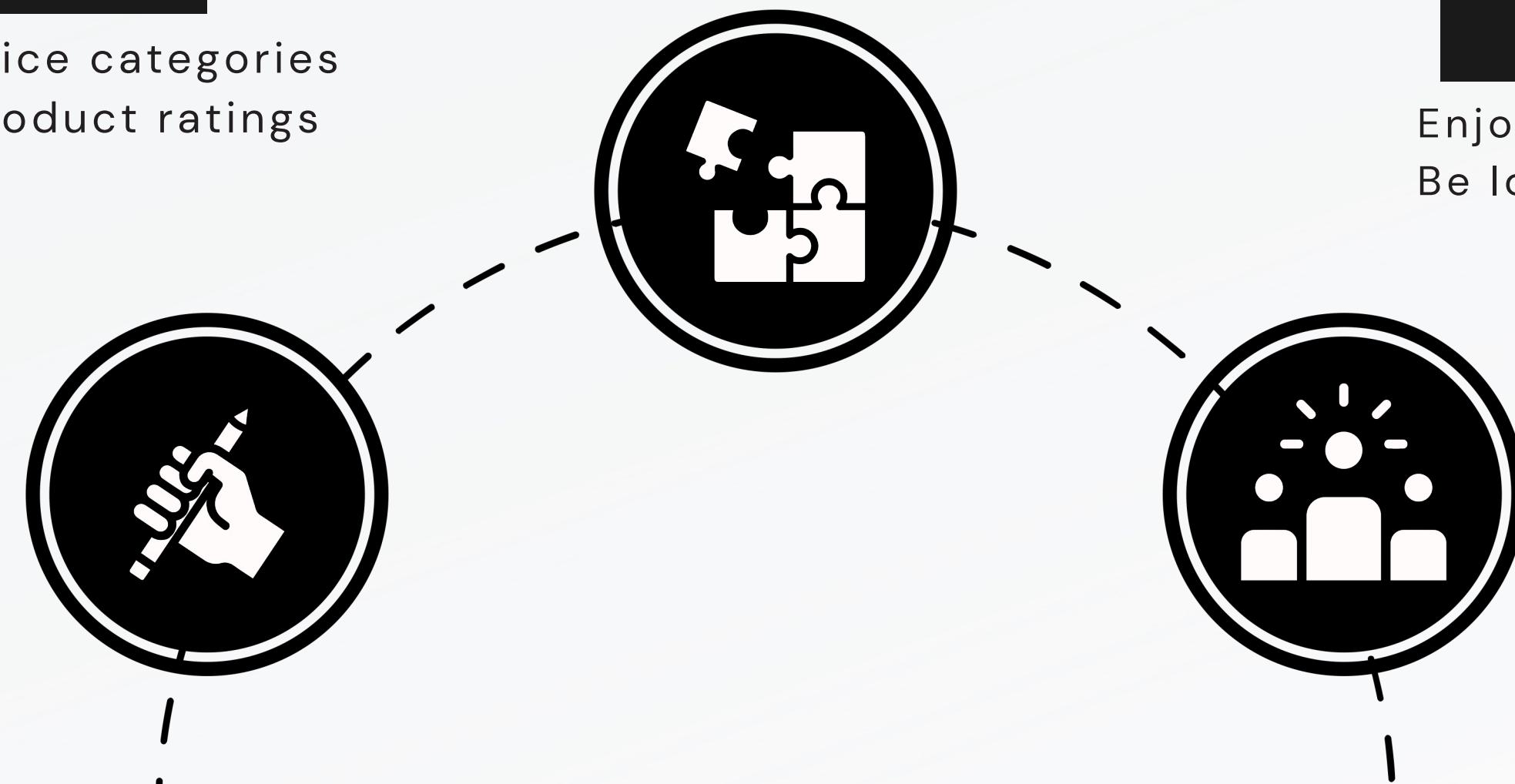
Based on different price categories
Based on different product ratings

Decision makers:
Reviews

Consumer engagements

Consumers:
User experience

Enjoy more customized services
Be loyal to the company



RECOMMENDATIONS

Revenue Optimization Strategies



Product Mix Enhancement:
Strategically allocate more inventory space to high-performing Adidas products to boost revenue.



Discount Strategy Refinement:
Consider adjusting discounts across brands to potentially increase overall revenue.

Sales Enhancement (Customer Engagement)



Review-Driven Sales Boost:
Encourage and increase the number of product reviews and leverage the strong positive correlation between reviews and revenue to boost sales.



Seasonal Review Volume Experimentation:
Run experiments to increase review volume in the latter nine months.

Product Portfolio and Revenue Insights



Footwear Dominance:
Recognize and leverage the substantial revenue potential of footwear products.



Clothing Product Performance:
Evaluate and refine strategies to enhance the revenue contribution of clothing products.

THANK YOU

Q & A



APPENDIX

1.

```
SELECT COUNT(*) AS total_rows,  
       COUNT(i.description) AS count_description,  
       COUNT(f.listing_price) AS count_listing_price,  
       COUNT(t.last_visited) AS count_last_visited  
  FROM info AS i  
 INNER JOIN finance AS f ON i.product_id =  
          f.product_id  
 INNER JOIN traffic AS t on t.product_id = f.product_id;
```

3.

```
SELECT b.brand, COUNT(f.*), SUM(f.revenue) as  
total_revenue,  
CASE WHEN f.listing_price < 42 THEN 'Budget'  
      WHEN f.listing_price >= 42 AND f.listing_price < 74  
      THEN 'Average'  
      WHEN f.listing_price >= 74 AND f.listing_price < 129  
      THEN 'Expensive'  
      ELSE 'Elite' END AS price_category  
  FROM finance AS f  
 INNER JOIN brands AS b  
    ON f.product_id = b.product_id  
 WHERE b.brand IS NOT NULL  
 GROUP BY b.brand, price_category  
 ORDER BY total_revenue DESC;
```

2.

```
SELECT b.brand, f.listing_price::integer, COUNT(f.*)  
  FROM finance AS f  
 INNER JOIN brands AS b  
    ON f.product_id = b.product_id  
 WHERE f.listing_price > 0  
 GROUP BY b.brand, f.listing_price  
 ORDER BY listing_price DESC;
```

4.

```
SELECT b.brand, AVG(f.discount) * 100 AS  
average_discount  
  FROM brands AS b  
 INNER JOIN finance AS f  
    ON b.product_id = f.product_id  
 GROUP BY b.brand  
 HAVING b.brand IS NOT NULL  
 ORDER BY average_discount;
```

5.

```
SELECT corr(r.reviews, f.revenue) AS  
review_revenue_corr  
  FROM reviews AS r  
 INNER JOIN finance AS f  
    ON r.product_id = f.product_id;
```

6.

```
SELECT TRUNC(LENGTH(i.description), -2) AS
description_length,
    ROUND(AVG(r.rating::numeric), 2) AS
average_rating
FROM info AS i
INNER JOIN reviews AS r
    ON i.product_id = r.product_id
WHERE i.description IS NOT NULL
GROUP BY description_length
ORDER BY description_length;
```

8.

```
WITH footwear AS
(
    SELECT i.description, f.revenue
    FROM info AS i
    INNER JOIN finance AS f
        ON i.product_id = f.product_id
    WHERE i.description ILIKE '%shoe%'
        OR i.description ILIKE '%trainer%'
        OR i.description ILIKE '%foot%'
        AND i.description IS NOT NULL
)
```

```
SELECT COUNT(*) AS num_footwear_products,
    percentile_disc(0.5) WITHIN GROUP (ORDER BY revenue) AS
median_footwear_revenue
FROM footwear;
```

7.

```
SELECT b.brand, DATE_PART('month', t.last_visited) AS month, COUNT(r.*) AS
num_reviews
FROM brands AS b
INNER JOIN traffic AS t
    ON b.product_id = t.product_id
INNER JOIN reviews AS r
    ON t.product_id = r.product_id
GROUP BY b.brand, month
HAVING b.brand IS NOT NULL
    AND DATE_PART('month', t.last_visited) IS NOT NULL
ORDER BY b.brand, month;
```

9.

```
WITH footwear AS
(
    SELECT i.description, f.revenue
    FROM info AS i
    INNER JOIN finance AS f
        ON i.product_id = f.product_id
    WHERE i.description ILIKE '%shoe%'
        OR i.description ILIKE '%trainer%'
        OR i.description ILIKE '%foot%'
        AND i.description IS NOT NULL
)
SELECT COUNT(i.*) AS num_clothing_products,
    percentile_disc(0.5) WITHIN GROUP (ORDER BY f.revenue) AS
median_clothing_revenue
FROM info AS i
INNER JOIN finance AS f on i.product_id = f.product_id
WHERE i.description NOT IN (SELECT description FROM
footwear);
```