Proposed Metrics

Identifying popular businesses as defined by a “Popular business seems to attract more customers compared to other businesses in the same category”. There are several things to consider first we have to define how we tell if a business attracts more customers and what does same category means. There are only a few things that tell how many people are interacting with a business, the most obvious is check-ins so this going to be part of whatever metric we use for this problem. For this problem, we don’t care about review ratings as we just care about the number of people but the number of reviews is useful as it shows engagement. As for category, the straightforward thing is to just use the business category The final thing to consider is where do you put the threshold of what it means to be popular since I am using review count and check-in count should popular business be above 90% percental or 70% percental in those categories? Given the lack of specificity to this, I am just going to say above average.

Metric for Popular Businesses:

A business is considered popular if it has a higher number of check-ins and reviews compared to the average in its category. This suggests both high foot traffic and engagement.

Script Explainer (appendix popular businesses):

This query identifies businesses that exceed the average number of check-ins and reviews within their categories. It calculates average check-ins and reviews for each category and then selects businesses with total check-ins and reviews surpassing these averages. I use a decent amount of Common Table Expressions (CTE) which you will note is a recurring theme in my queries I like just using them.

Identify successful businesses as defined by “Successful businesses have been serving the community for a long time and have loyal customers.” I am keeping this shorter than the last one based on the description I care about how long a business has been operating and above average number of customers and they give it could ratings. For how long there is no direct metric but by proxy, I can use the earliest review date. Above-average customer count can be translated into above-average check-ins like before and it can

Metric for Successful Businesses:

A business is considered successful if it has been operating for a long time at least 3 years (determined by the earliest date of its reviews), has a high rating of 4+, and above average check-ins in its business category.

Scripted Explainer (appendix successful businesses):

This query identifies businesses that are at least three years old, have check-ins above their category's average (using a slightly modified technique from before), and maintain an

average rating above 4. It calculates each business's age from the earliest review date, aggregates total check-ins, and compares them to the category average. Finally, it filters these businesses by their high customer ratings, ensuring they meet all criteria for being considered both popular and high-quality. This ended up needing to be way longer than I wanted and I probably should have done a less complicated metric.

Identifying expensive businesses, we were given no specific definition and besides price range for restaurants, there doesn't appear good data on what is expensive. So if I want to find all businesses that are expensive not just restaurants I am going to look at the text of the reviews to gauge this. I created the following list: high priced, high cost, expensive, that's a bit pricey, costs an arm and a leg, exorbitant, costly, pricey. And decided that any business that had reviews with one of these phrases would be considered expensive.

Metric for Expensive Businesses:

a business received a review with at least one of these phrases: high priced, high cost, expensive, that's a bit pricey, costs an arm and a leg, exorbitant, costly, pricey.

Scripted explainer (appendix expensive businesses):

This is the simplest one I have done I join reviews of a business together and search for every one of these phrases anywhere in the text also making it none case sensitive by making everything lowercase. Then return distinct hits.

Appendix

```sql
--Popular Businesses
WITH CategoryAverageCheckIns AS (
    SELECT
        bc.CategoryID,
        AVG(c.COUNT) AS AvgCheckIns
    FROM
        CheckIn c
    JOIN BusinessCategory bc ON c.BusinessID = bc.BusinessID
    GROUP BY bc.CategoryID
),
BusinessCheckIns AS (
    SELECT
        c.BusinessID,
        SUM(c.COUNT) AS TotalCheckIns
    FROM CheckIn c
    GROUP BY c.BusinessID
),
CategoryAverageReviews AS (
    SELECT
        bc.CategoryID,
        AVG(sub.ReviewCount) AS AvgReviewCount
    FROM
        BusinessCategory bc
    JOIN (
        SELECT
            r.BusinessID,
            COUNT(*) AS ReviewCount
        FROM Review r
        GROUP BY r.BusinessID
    ) sub ON bc.BusinessID = sub.BusinessID
    GROUP BY bc.CategoryID
```

```
),
BusinessReviews AS (
    SELECT
        r.BusinessID,
        COUNT(*) AS TotalReviews
    FROM Review r
    GROUP BY r.BusinessID
)
SELECT
    DISTINCT b.BusinessID,
    b.BusinessName,
    bci.TotalCheckIns,
    br.TotalReviews
FROM
    Business b
JOIN BusinessCheckIns bci ON b.BusinessID = bci.BusinessID
JOIN BusinessReviews br ON b.BusinessID = br.BusinessID
JOIN BusinessCategory bc ON b.BusinessID = bc.BusinessID
JOIN CategoryAverageCheckIns caci ON bc.CategoryID = caci.CategoryID
JOIN CategoryAverageReviews car ON bc.CategoryID = car.CategoryID
WHERE
    bci.TotalCheckIns > caci.AvgCheckIns AND
    br.TotalReviews > car.AvgReviewCount
ORDER BY
    b.BusinessName;




--Succesful Businesses
WITH BusinessAge AS (
    SELECT
        b.BusinessID,
        MIN(r.ReviewDate) AS FirstReviewDate,
        EXTRACT(YEAR FROM AGE(TIMESTAMP '2023-04-01', MIN(r.ReviewDate)))
AS YearsInOperation
    FROM
        Business b
    JOIN
        Review r ON b.BusinessID = r.BusinessID
```

```
    GROUP BY
        b.BusinessID
    HAVING
        EXTRACT(YEAR FROM AGE(TIMESTAMP '2023-04-01', MIN(r.ReviewDate)))
>= 3
),
CategoryAverageCheckIns AS (
    SELECT
        bc.CategoryID,
        AVG(c.COUNT) AS AvgCheckIns
    FROM
        CheckIn c
    JOIN
        BusinessCategory bc ON c.BusinessID = bc.BusinessID
    GROUP BY
        bc.CategoryID
),
BusinessCheckIns AS (
    SELECT
        c.BusinessID,
        SUM(c.COUNT) AS TotalCheckIns
    FROM
        CheckIn c
    GROUP BY
        c.BusinessID
),
FilteredBusinesses AS (
    SELECT DISTINCT
        b.BusinessID,
        b.BusinessName,
        b.Stars,
        ba.YearsInOperation,
        bci.TotalCheckIns
    FROM
        Business b
    JOIN
        BusinessAge ba ON b.BusinessID = ba.BusinessID
    JOIN
        BusinessCheckIns bci ON b.BusinessID = bci.BusinessID
)
```

```sql
SELECT
    fb.BusinessID,
    fb.BusinessName,
    fb.Stars,
    fb.YearsInOperation,
    fb.TotalCheckIns
FROM
    FilteredBusinesses fb
JOIN
    BusinessCategory bc ON fb.BusinessID = bc.BusinessID
JOIN
    CategoryAverageCheckIns cai ON bc.CategoryID = cai.CategoryID
WHERE
    fb.TotalCheckIns > cai.AvgCheckIns
    AND fb.Stars > 4
GROUP BY
    fb.BusinessID, fb.BusinessName, fb.Stars, fb.YearsInOperation,
fb.TotalCheckIns
HAVING
    COUNT(DISTINCT bc.CategoryID) >= 1;


--Expensive
SELECT DISTINCT
    b.BusinessID,
    b.BusinessName
FROM
    Business b
JOIN
    Review r ON b.BusinessID = r.BusinessID
WHERE
    LOWER(r.ReviewText) LIKE '%high priced%' OR
    LOWER(r.ReviewText) LIKE '%high cost%' OR
    LOWER(r.ReviewText) LIKE '%expensive%' OR
    LOWER(r.ReviewText) LIKE '%that’s a bit pricey%' OR
    LOWER(r.ReviewText) LIKE '%costs an arm and a leg%' OR
    LOWER(r.ReviewText) LIKE '%exorbitant%' OR
    LOWER(r.ReviewText) LIKE '%costly%' OR
    LOWER(r.ReviewText) LIKE '%high end%' OR
    LOWER(r.ReviewText) LIKE '%pricey%'
```

```
ORDER BY
    b.BusinessName;
```