Power BI Visualization Guide for Search Analytics

This guide provides recommended visualizations for analyzing Intranet search behavior using the parquet files generated by the processing script.

Table of Contents

Data Sources Setup

Connecting to Parquet Files

1. **Get Data** > **Parquet**
2. Navigate to `output/` folder
3. Load all three files:
   - `searches_daily.parquet` - Daily aggregated metrics (1 row per day)
   - `searches_journeys.parquet` - Session-level data (1 row per search session)
   - `searches_terms.parquet` - Search term analytics (1 row per term per day)

Important: Rate and Average Calculations in DAX

Why DAX Measures Are Required

Rates and averages must be calculated in Power BI using DAX measures, not pre-calculated columns. This is because averaging percentages or ratios gives mathematically incorrect results:

- Day 1: 50 clicks / 100 searches = 50% CTR
- Day 2: 10 clicks / 200 searches = 5% CTR
- Wrong: Average of 50% and 5% = 27.5%
- Correct: (50 + 10) / (100 + 200) = 20%

Building Block Columns

The parquet files include all necessary component columns for correct calculations:

Metric to Calculate	Numerator Column	Denominator Column
Success Click Rate %	`success_clicks`	`search_starts`
Null Rate %	`null_results`	`result_events`
Session Success Rate %	`sessions_with_clicks`	`sessions_with_results`
Session Abandonment Rate %	`sessions_abandoned`	`sessions_with_results`
Avg Searches per Session	`search_starts`	`unique_sessions`
Avg Search Term Length	`sum_search_term_length`	`search_term_count`
Avg Search Term Words	`sum_search_term_words`	`search_term_count`
Avg Seconds to Click	`sum_sec_to_click`	`clicks_with_timing`
Avg Click Position	`sum_click_position`	`click_position_count`
Avg Search Latency	`sum_search_latency_ms`	`latency_event_count`
Avg News Results	`sum_news_result_count`	`result_events_with_results`
ViewMore Click Rate	`clicks_viewmore`	`click_events`

Required DAX Measures

Create these DAX measures to calculate rates and averages correctly across any date range or filter context.

Rate Metrics - DAX Formulas

```
// Success Click Rate (actual result clicks per search)
// Uses success_clicks which only counts SEARCH_RESULT_CLICK (not
trending)
// Can exceed 100% if users click multiple results per search
Success Click Rate % =
```

```
DIVIDE(
    SUM(searches_daily[success_clicks]),
    SUM(searches_daily[search_starts]),
    0
) * 100

// All Clicks Rate (includes navigation: tabs, pagination, filters)
// Use this if you want to count ALL user interactions
All Clicks Rate % =
DIVIDE(
    SUM(searches_daily[click_events]),
    SUM(searches_daily[search_starts]),
    0
) * 100

// Null Result Rate
Null Rate % =
DIVIDE(
    SUM(searches_daily[null_results]),
    SUM(searches_daily[result_events]),
    0
) * 100

// Session Success Rate (sessions with at least one SUCCESS click /
sessions with results)
// Always 0-100% - recommended for KPIs
// Note: sessions_with_clicks is pre-calculated in the parquet using
is_success_click
// (only counts SEARCH_RESULT_CLICK and SEARCH_TRENDING_CLICKED, not
navigation clicks)
Session Success Rate % =
DIVIDE(
    SUM(searches_daily[sessions_with_clicks]),
    SUM(searches_daily[sessions_with_results]),
    0
) * 100

// Session Abandonment Rate (sessions with results but no SUCCESS clicks /
sessions with results)
// Always 0-100% - recommended for KPIs
// Note: sessions_abandoned is pre-calculated using is_success_click
Session Abandonment Rate % =
DIVIDE(
    SUM(searches_daily[sessions_abandoned]),
    SUM(searches_daily[sessions_with_results]),
    0
) * 100
```

Note on Click Types:

- **Success clicks** (`success_clicks`): Only actual result clicks (`SEARCH_RESULT_CLICK` / `SEARCH_RESULT_CLICKED`)

- **ViewMore clicks** (`clicks_viewmore`): "View more" link clicks (`SEARCH_VIEW_MORE_LINK`) - navigation, not content discovery
- **Trending clicks** (`clicks_trending`): Search initiation via suggestion (`SEARCH_TRENDING_CLICKED`) - tracked separately
- **All clicks** (`click_events`): Includes navigation (tabs, pagination, filters, trending, viewmore)
- `Session Success Rate %` is session-based and always 0-100% (did the session have any success clicks?)
- Use **Session Success/Abandonment rates** for executive KPIs

Average Metrics - DAX Formulas

```
// Average Searches per Session
Avg Searches per Session =
DIVIDE(
    SUM(searches_daily[search_starts]),
    SUM(searches_daily[unique_sessions]),
    0
)

// Average Search Term Length (weighted average using SUM columns)
Avg Search Term Length =
DIVIDE(
    SUM(searches_daily[sum_search_term_length]),
    SUM(searches_daily[search_term_count]),
    0
)

// Average Search Term Words (weighted average using SUM columns)
Avg Search Term Words =
DIVIDE(
    SUM(searches_daily[sum_search_term_words]),
    SUM(searches_daily[search_term_count]),
    0
)

// Average Click Position (weighted average using SUM columns)
Avg Click Position =
DIVIDE(
    SUM(searches_daily[sum_click_position]),
    SUM(searches_daily[click_position_count]),
    BLANK()
)
```

Component Columns Reference

Metric to Calculate	Numerator Column	Denominator Column
Success Click Rate %	`success_clicks`	`search_starts`
All Clicks Rate %	`click_events`	`search_starts`

Metric to Calculate	Numerator Column	Denominator Column
Null Rate %	null_results	result_events
Session Success Rate %	sessions_with_clicks	sessions_with_results
Session Abandonment Rate %	sessions_abandoned	sessions_with_results
Avg Searches/Session	search_starts	unique_sessions
Avg Search Term Length	sum_search_term_length	search_term_count
Avg Search Term Words	sum_search_term_words	search_term_count
Avg Click Position	sum_click_position	click_position_count
Avg Search Latency	sum_search_latency_ms	latency_event_count
Avg News Results	sum_news_result_count	result_events_with_results
ViewMore Click Rate	clicks_viewmore	click_events

Click Column Types:

- success_clicks = Result clicks only (SEARCH_RESULT_CLICK / SEARCH_RESULT_CLICKED - user found content)
- clicks_viewmore = ViewMore link clicks (SEARCH_VIEW_MORE_LINK - navigation, not content discovery)
- clicks_trending = Trending suggestion clicks (search initiation, not content discovery)
- click_events = All clicks including navigation (tabs, pagination, filters, trending, viewmore)

User Cohort Columns (Daily File)

The daily file includes columns for analyzing new vs returning users:

Column	Type	Description
new_users	Integer	Users whose first search session was on this date
returning_users	Integer	Users who had searched before this date

DAX Measures for User Cohorts:

```
// New User Percentage (daily)
New User % =
DIVIDE(
    SUM(searches_daily[new_users]),
    SUM(searches_daily[unique_users]),
    0
) * 100

// Returning User Percentage (daily)
Returning User % =
DIVIDE(
```

```
    SUM(searches_daily[returning_users]),
    SUM(searches_daily[unique_users]),
    0
) * 100
```

When to Use Pre-Calculated vs DAX Measures

Scenario	Use Pre-Calculated Column	Use DAX Measure
Single day view (date slicer = 1 day)	OK	OK
Multiple days aggregated	NO	YES
Weekly/Monthly totals	NO	YES
Filtering by other dimensions	NO	YES
Quick glance at daily table	OK	OK

Best Practice

Always use DAX measures for KPI tiles and aggregated views. The pre-calculated columns are useful for:

- Quick reference in detailed daily tables
- Validation that your DAX measures match single-day values
- Export to other tools that will handle aggregation separately

---

Page 1: Executive Dashboard (Daily Metrics)

Row 1: KPI Tiles

Tile	Measure	Description
Total Searches	`SUM(search_starts)`	Total search queries
Unique Users	`SUM(unique_users)`	Distinct users who searched
Session Success Rate	See DAX below	Percentage of sessions with results that had clicks
Null Result Rate	See DAX below	Percentage of searches with zero results
Session Abandonment Rate	See DAX below	Percentage of sessions with results but no clicks

DAX Measures for Correct Aggregation

```
// Session Success Rate (always 0-100%)
Session Success Rate % =
DIVIDE(
```

```
    SUM(searches_daily[sessions_with_clicks]),
    SUM(searches_daily[sessions_with_results]),
    0
) * 100

// Session Abandonment Rate (always 0-100%)
Session Abandonment Rate % =
DIVIDE(
    SUM(searches_daily[sessions_abandoned]),
    SUM(searches_daily[sessions_with_results]),
    0
) * 100

// Null Result Rate
Null Rate % =
DIVIDE(
    SUM(searches_daily[null_results]),
    SUM(searches_daily[result_events]),
    0
) * 100

// Success Click Rate (actual result clicks only)
// Uses success_clicks (SEARCH_RESULT_CLICK only, not trending)
Success Click Rate % =
DIVIDE(
    SUM(searches_daily[success_clicks]),
    SUM(searches_daily[search_starts]),
    0
) * 100

// All Clicks Rate (includes navigation - tabs, pagination, filters)
All Clicks Rate % =
DIVIDE(
    SUM(searches_daily[click_events]),
    SUM(searches_daily[search_starts]),
    0
) * 100

// Average Searches per Session
Avg Searches/Session =
DIVIDE(
    SUM(searches_daily[search_starts]),
    SUM(searches_daily[unique_sessions]),
    0
)

// Average Search Term Length (weighted)
Avg Search Term Length =
DIVIDE(
    SUM(searches_daily[sum_search_term_length]),
    SUM(searches_daily[search_term_count]),
    0
)
```

Row 2: Trend Charts

Chart 1: Search Volume Over Time

- **Type**: Line Chart
- **X-Axis**: `date`
- **Y-Axis**: `search_starts`
- **Secondary Y-Axis** (optional): `unique_users`

Chart 2: Quality Metrics Over Time

- **Type**: Line Chart (multiple series)
- **X-Axis**: `date`
- **Lines**:
  - `Session Success Rate %` (DAX measure)
  - `Null Rate %` (DAX measure)
  - `Session Abandonment Rate %` (DAX measure)

Row 3: Click Distribution

Chart 3: Clicks by Category

- **Type**: Stacked Bar Chart or Donut
- **Values**:
  - `clicks_result` - Actual search result clicks
  - `clicks_viewmore` - View more link clicks
  - `clicks_trending` - Trending item clicks
  - `clicks_tab` - Tab navigation clicks
  - `clicks_pagination` - All pagination clicks (aggregate)
  - `clicks_filter` - Filter clicks
- **Detailed pagination breakdown** (optional):
  - `clicks_pagination_all` - All tab pagination
  - `clicks_pagination_news` - News tab pagination
  - `clicks_pagination_goto` - GoTo tab pagination

Chart 4: Daily Activity Table

- **Type**: Table
- **Columns**: `date`, `search_starts`, `unique_users`, `Session Success Rate %`, `Null Rate %`
- **Conditional Formatting**: Highlight low success rates or high null rates

Row 4: Temporal Patterns

Chart 5: Searches by Day of Week

- **Type**: Bar Chart
- **Setup**:

    1. Drag `day_of_week` to **Axis**
    2. Drag `search_starts` to **Values** → select **Sum**
- **Sort**: By `day_of_week_num` (configure in Model view: select `day_of_week`, set "Sort by column" to `day_of_week_num`)
- **Insight**: Identify busiest days - weekday vs weekend patterns

Chart 6: Search Time Distribution

- **Type**: Stacked Bar or Pie Chart
- **Values** (all from daily file):
    - `searches_night` (0:00-8:00 CET - APAC peak)
    - `searches_morning` (8:00-12:00 CET - EMEA peak)
    - `searches_afternoon` (12:00-18:00 CET - EMEA+Americas overlap)
    - `searches_evening` (18:00-24:00 CET - Americas peak)
- **Insight**: When do users search most? Time periods align with regional business hours.

Page 2: User Journey Analysis (Session Metrics)

Understanding the Journeys File

The `searches_journeys.parquet` file contains **one row per session**. Each row represents a complete user journey from first search to last action.

Key columns:

Column	Type	Description
`session_date`	Date	Date of the session
`journey_outcome`	String	Success, Engaged, Abandoned, No Results, Unknown
`search_count_in_session`	Integer	Number of searches in this session
`result_count`	Integer	Number of SEARCH_RESULT_COUNT events
`click_count`	Integer	Number of ALL clicks (including navigation)
`success_click_count`	Integer	Number of SUCCESS clicks (SEARCH_RESULT_CLICK only)
`null_result_count`	Integer	Number of searches with 0 results
`had_reformulation`	Boolean	Did user search multiple different terms?
`session_complexity`	String	Single Action, Simple, Medium, Complex (based on user actions: searches + clicks)
`search_to_result_bucket`	String	Time bucket for search-to-result
`result_to_click_bucket`	String	Time bucket for result-to-click
`session_duration_bucket`	String	Time bucket for total session duration

Column	Type	Description
sec_search_to_result	Decimal	Seconds from search to result
sec_result_to_click	Decimal	Seconds from result to click
total_duration_sec	Decimal	Total session duration in seconds
search_to_result_sort	Integer	Sort order for search_to_result_bucket
result_to_click_sort	Integer	Sort order for result_to_click_bucket
session_duration_sort	Integer	Sort order for session_duration_bucket
journey_outcome_sort	Integer	Sort order for journey_outcome
session_complexity_sort	Integer	Sort order for session_complexity
had_null_result	Boolean	Did session have any null result?
recovered_from_null	Boolean	Had null result but still clicked (recovered)
user_session_number	Integer	Which session is this for the user (1, 2, 3...)
is_users_first_session	Boolean	Is this the user's first session in dataset?
distinct_click_categories	Integer	Number of different click types in session
had_tab_switch	Boolean	Did user click on multiple tabs/categories?
viewmore_clicks	Integer	SEARCH_VIEW_MORE_LINK events in session
device_type	String	User's device type
department	String	User's department
location	String	User's location
job_title	String	User's job title
query_language	String	Query language
avg_click_position	Float	Average position of result clicks
min_click_position	Integer	Best (lowest) click position
max_news_results	Integer	Max news results shown in session
avg_search_latency_ms	Float	Average search latency (ms)
distinct_tabs_clicked	Integer	Unique tabs clicked
distinct_filters_used	Integer	Unique filters used

Click breakdown columns:

Column	Description
result_clicks	SEARCH_RESULT_CLICK events

Column	Description
trending_clicks	SEARCH_TRENDING_CLICKED events
tab_clicks	SEARCH_TAB_CLICK events
pagination_clicks	All pagination clicks (aggregate)
pagination_all_clicks	All tab pagination
pagination_news_clicks	News tab pagination
pagination_goto_clicks	GoTo tab pagination
filter_clicks	SEARCH_FILTER_CLICK events
viewmore_clicks	SEARCH_VIEW_MORE_LINK events

DAX Measures for Journeys

Create these measures for the journeys visualizations:

```
// Basic session count
Session Count = COUNTROWS(searches_journeys)

// Journey Outcome Counts
Success Sessions =
CALCULATE(
    COUNTROWS(searches_journeys),
    searches_journeys[journey_outcome] = "Success"
)

Engaged Sessions =
CALCULATE(
    COUNTROWS(searches_journeys),
    searches_journeys[journey_outcome] = "Engaged"
)

Abandoned Sessions =
CALCULATE(
    COUNTROWS(searches_journeys),
    searches_journeys[journey_outcome] = "Abandoned"
)

No Results Sessions =
CALCULATE(
    COUNTROWS(searches_journeys),
    searches_journeys[journey_outcome] = "No Results"
)

// Journey Outcome Rates
Success Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
```

```
searches_journeys[journey_outcome] = "Success"),
    COUNTROWS(searches_journeys),
    0
) * 100

Engaged Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[journey_outcome] = "Engaged"),
    COUNTROWS(searches_journeys),
    0
) * 100

Abandonment Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[journey_outcome] = "Abandoned"),
    COUNTROWS(searches_journeys),
    0
) * 100

No Results Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[journey_outcome] = "No Results"),
    COUNTROWS(searches_journeys),
    0
) * 100

// Reformulation Rate
Reformulation Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[had_reformulation] = TRUE()),
    COUNTROWS(searches_journeys),
    0
) * 100

// Average Timing Metrics
Avg Search to Result (sec) =
AVERAGE(searches_journeys[sec_search_to_result])

Avg Result to Click (sec) =
AVERAGE(searches_journeys[sec_result_to_click])

Avg Session Duration (sec) =
AVERAGE(searches_journeys[total_duration_sec])

// Average Searches per Session (from journeys)
Avg Searches per Session (Journeys) =
AVERAGE(searches_journeys[search_count_in_session])

// Null Result Recovery Rate - sessions that had null result but still
succeeded
```

```
Null Recovery Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[recovered_from_null] = TRUE()),
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[had_null_result] = TRUE()),
    0
) * 100

// First-time User Success Rate
First-Time User Success Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
        searches_journeys[is_users_first_session] = TRUE(),
        searches_journeys[journey_outcome] = "Success"),
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[is_users_first_session] = TRUE()),
    0
) * 100

// Returning User Success Rate
Returning User Success Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
        searches_journeys[is_users_first_session] = FALSE(),
        searches_journeys[journey_outcome] = "Success"),
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[is_users_first_session] = FALSE()),
    0
) * 100

// Tab Switch Rate
Tab Switch Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[had_tab_switch] = TRUE()),
    COUNTROWS(searches_journeys),
    0
) * 100

// Average Click Position (result clicks only)
Avg Click Position =
AVERAGE(searches_journeys[avg_click_position])

// Average Search Latency
Avg Search Latency (ms) =
AVERAGE(searches_journeys[avg_search_latency_ms])

// ViewMore Rate (sessions with viewmore clicks)
ViewMore Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[viewmore_clicks] > 0),
    COUNTROWS(searches_journeys),
```

```
    0
) * 100
```

Journey Types Analysis

The Journey Types view provides a human-readable summary of session patterns, showing exactly what happened in each session (e.g., "1 Search → 1 Result → 1 Click").

Step 1: Create a Calculated Column (Model view → `searches_journeys` → New Column)

```
Journey_Type =
VAR searches = [search_count_in_session]
VAR results = [result_count]
VAR clicks = [click_count]
VAR success_clicks = [success_click_count]
VAR nulls = [null_result_count]
RETURN
SWITCH(
    TRUE(),
    // Success: User clicked on actual search result content
    searches > 0 && results > 0 && success_clicks > 0,
        searches & " Search → " & results & " Result → " & success_clicks
& " Click",
    // Engaged: User interacted (tabs/pagination/filters) but didn't click
content
    searches > 0 && results > 0 && clicks > 0 && success_clicks = 0,
        searches & " Search → " & results & " Result → Engaged (" & clicks
& " nav)",
    // No Results: All results were null
    searches > 0 && results > 0 && nulls > 0 && clicks = 0,
        searches & " Search → " & results & " Result (incl. " & nulls & "
null) → No Click",
    // Abandoned: Had results but no interaction at all
    searches > 0 && results > 0 && clicks = 0,
        searches & " Search → " & results & " Result → Abandoned",
    // No Result: Search returned nothing
    searches > 0 && results = 0,
        searches & " Search → No Result",
    "Other"
)
```

Step 2: Create a Percentage Measure (optional)

```
Journey Type % =
DIVIDE(
    COUNTROWS(searches_journeys),
    CALCULATE(COUNTROWS(searches_journeys),
ALL(searches_journeys[Journey_Type])),
```

```
    0
) * 100
```

Step 3: Create the Visualization

- **Type**: Table or Horizontal Bar Chart
- **Setup for Table**:
    1. Drag `Journey_Type` to **Columns**
    2. Add count of `Journey_Type` (Sessions)
    3. Add `Journey Type %` measure
    4. Sort by Sessions descending
- **Setup for Bar Chart**:
    1. Drag `Journey_Type` to **Y-axis**
    2. Count of `Journey_Type` to **X-axis**
    3. Sort by value descending

Example Output:

Journey_Type	Sessions	%
1 Search → 1 Result → 1 Click	1,245	42.1%
1 Search → 1 Result → Abandoned	532	18.0%
2 Search → 2 Result → 1 Click	289	9.8%
1 Search → 1 Result → Engaged (2 nav)	245	8.3%
1 Search → 1 Result (incl. 1 null) → No Click	156	5.3%
1 Search → No Result	98	3.3%

What this tells you:

- **Ideal pattern**: "1 Search → 1 Result → 1 Click" (user found what they needed immediately)
- **Refinement needed**: "2+ Search" patterns indicate users had to refine their query
- **Content gap**: "No Result" patterns suggest missing content
- **Relevance issue**: "Abandoned" patterns suggest results weren't relevant

Setting Up Sort Order for Columns

The bucket and category columns are text and won't sort correctly by default (e.g., "10-30s" would come before "2-5s" alphabetically). The parquet file includes sort order columns to fix this.

To configure "Sort by column" in Power BI:

1. Go to **Model view** (left sidebar)
2. Select the text column you want to sort (e.g., `journey_outcome`)
3. In the **Properties** pane, find **Sort by column**
4. Select the corresponding sort column (e.g., `journey_outcome_sort`)

Column mappings:

Text Column	Sort By Column	Table
journey_outcome	journey_outcome_sort	searches_journeys
session_complexity	session_complexity_sort	searches_journeys
search_to_result_bucket	search_to_result_sort	searches_journeys
result_to_click_bucket	result_to_click_sort	searches_journeys
session_duration_bucket	session_duration_sort	searches_journeys
Query_Length_Bucket	Query_Length_Sort	searches_terms (calculated)

Once configured, the text columns will sort in logical order everywhere they're used.

Row 1: Journey Outcomes

Chart 1: Journey Outcome Distribution

- **Type**: Donut Chart
- **Setup**:
    1. Drag `journey_outcome` to **Legend**
    2. Drag `journey_outcome` to **Values** → select **Count**
- **Colors**:
    - Success = Green (#2E7D32)
    - Abandoned = Orange (#F57C00)
    - No Results = Red (#C62828)
    - Unknown = Gray (#757575)
- **Sort**: Configured via "Sort by column" (see above)

Chart 2: Session Complexity Breakdown

- **Type**: Bar Chart
- **Setup**:
    1. Drag `session_complexity` to **Axis**
    2. Drag `session_complexity` to **Values** → select **Count**
- **Sort**: Configured via "Sort by column" (see above)

Row 2: Timing Analysis

Chart 3: Search-to-Result Time Distribution

- **Type**: Bar Chart (Clustered Bar)
- **Setup**:
    1. Drag `search_to_result_bucket` to **Axis**
    2. Drag `search_to_result_bucket` to **Values** → select **Count**
- **Sort**: Configured via "Sort by column" (see above)
- **Insight**: Shows system performance - ideally most sessions should be < 2s

Chart 4: Result-to-Click Time Distribution

- **Type**: Bar Chart (Clustered Bar)
- **Setup**:
    1. Drag `result_to_click_bucket` to **Axis**
    2. Drag `result_to_click_bucket` to **Values** → select **Count**
- **Sort**: Configured via "Sort by column" (see above)
- **Insight**: Shows user decision time - quick clicks may indicate good relevance

Row 3: Behavior Patterns

Chart 5: Reformulation Rate

- **Type**: Card or KPI
- **Setup**: Use the `Reformulation Rate %` measure
- **Target**: < 30% is good (users find what they need on first try)
- **Insight**: High reformulation suggests search relevance issues

Chart 6: Session Duration Distribution

- **Type**: Bar Chart
- **Setup**:
    1. Drag `session_duration_bucket` to **Axis**
    2. Drag `session_duration_bucket` to **Values** → select **Count**
- **Sort**: Configured via "Sort by column" (see above)
- **Insight**: Very short sessions might be quick successes OR immediate abandonment

Row 4: KPI Cards (Optional)

Create a row of KPI cards for quick insights:

Card	Measure	Target
Total Sessions	`Session Count`	-
Success Rate	`Success Rate %`	> 40%
Avg Searches/Session	`Avg Searches per Session (Journeys)`	< 2
Reformulation Rate	`Reformulation Rate %`	< 30%
Avg Time to Click	`Avg Result to Click (sec)`	< 10s

Page 3: Deep Dive Analysis

Filters Panel

Add slicers for:

- `session_date` (date range)
- `journey_outcome`
- `session_complexity`
- `search_to_result_bucket`

- `device_type` (new)
- `department` (new)
- `location` (new)
- `query_language` (new)

Chart 1: Heatmap - Hour of Day Activity

- **Type**: Matrix
- **Rows**: Day of week (from `session_date`)
- **Columns**: `first_event_hour`
- **Values**: Count of sessions
- **Conditional Formatting**: Heat colors

Chart 2: Click Category by Outcome

- **Type**: Stacked Bar Chart
- **Axis**: `journey_outcome`
- **Legend**: Click type columns (`result_clicks`, `trending_clicks`, `tab_clicks`, `pagination_clicks`, `filter_clicks`)
- **Values**: Sum of click counts

Chart 3: Sessions with Multiple Searches

- **Type**: Histogram or Bar
- **Axis**: `search_count_in_session` (grouped into bins)
- **Values**: Count of sessions
- **Insight**: How often users need multiple searches

Page 4: Search Terms Analysis

Understanding the Terms File

The `searches_terms.parquet` file contains **one row per search term per day**. It enables analysis of which terms users search for and how successful those searches are.

Key columns:

Column	Type	Description
`session_date`	Date	Date
`search_term`	String	The normalized search query
`word_count`	Integer	Number of words in search term
`search_count`	Integer	Times this term was searched
`unique_users`	Integer	Distinct users who searched this
`unique_sessions`	Integer	Sessions containing this term

Column	Type	Description
result_events	Integer	Result count events for this term
null_result_count	Integer	Searches returning 0 results
sum_result_count	Integer	Sum of result counts (for weighted avg in DAX)
click_count	Integer	ALL clicks attributed to this term
success_click_count	Integer	SUCCESS clicks only (SEARCH_RESULT_CLICK)
clicks_with_timing	Integer	Clicks with timing data (for weighting)
sum_sec_to_click	Float	Sum of seconds to click (for weighted avg)
searches_night	Integer	Searches 03:00-09:00 CET (APAC)
searches_morning	Integer	Searches 09:00-16:00 CET (CET)
searches_afternoon	Integer	Searches 16:00-22:00 CET (Americas)
searches_evening	Integer	Searches 22:00-03:00 CET (Dead time)
first_seen_date	Date	First date this term appeared in dataset
is_new_term	Boolean	Is this the first day this term was searched?

Click breakdown columns:

Column	Description
clicks_result	Actual result clicks
clicks_viewmore	View more link clicks
clicks_trending	Trending item clicks
clicks_tab	Tab navigation clicks
clicks_pagination	All pagination (aggregate)
clicks_pagination_all	All tab pagination
clicks_pagination_news	News tab pagination
clicks_pagination_goto	GoTo tab pagination
clicks_filter	Filter clicks

New metric columns:

Column	Type	Description
sum_click_position	Integer	Sum of click positions for result clicks (for weighted avg in DAX)
click_position_count	Integer	Result clicks with position data (denominator for avg)

Column	Type	Description
sum_news_result_count	Integer	Sum of news result counts (for weighted avg)
sum_search_latency_ms	Float	Sum of search latency (for weighted avg in DAX)
latency_event_count	Integer	Events with latency data (denominator for avg)

DAX Measures for Search Terms

```
// Total searches for all terms
Total Term Searches = SUM(searches_terms[search_count])

// Term Success CTR (actual result clicks only)
// Uses success_click_count (SEARCH_RESULT_CLICK only, not trending)
Term Success CTR % =
DIVIDE(
    SUM(searches_terms[success_click_count]),
    SUM(searches_terms[search_count]),
    0
) * 100

// Term All Clicks Rate (includes navigation clicks)
Term All Clicks Rate % =
DIVIDE(
    SUM(searches_terms[click_count]),
    SUM(searches_terms[search_count]),
    0
) * 100

// Zero Result Rate for terms
Term Zero Result Rate % =
DIVIDE(
    SUM(searches_terms[null_result_count]),
    SUM(searches_terms[result_events]),
    0
) * 100

// Term Abandonment Rate (had results but no SUCCESS click)
Term Abandonment Rate % =
DIVIDE(
    SUM(searches_terms[result_events]) -
SUM(searches_terms[null_result_count]) -
SUM(searches_terms[success_click_count]),
    SUM(searches_terms[result_events]) -
SUM(searches_terms[null_result_count]),
    0
) * 100

// Unique Terms Count
Unique Terms = DISTINCTCOUNT(searches_terms[search_term])

// Weighted Average Time to Click (seconds)
```

```
// Uses SUM columns for proper aggregation across dates/terms
Avg Time to Click (sec) =
DIVIDE(
    SUM(searches_terms[sum_sec_to_click]),
    SUM(searches_terms[clicks_with_timing]),
    BLANK()
)

// Query Length Bucket - see "Query Length vs Success Analysis" section
below
// for both the bucket column and the sort column definitions

// New Terms Count (terms first seen on selected date range)
New Terms Count =
CALCULATE(
    DISTINCTCOUNT(searches_terms[search_term]),
    searches_terms[is_new_term] = TRUE()
)

// New Term Rate (what % of today's terms are new)
New Term Rate % =
DIVIDE(
    CALCULATE(COUNTROWS(searches_terms), searches_terms[is_new_term] =
TRUE()),
    COUNTROWS(searches_terms),
    0
) * 100

// Weighted Avg Search Latency for terms
Term Avg Search Latency (ms) =
DIVIDE(
    SUM(searches_terms[sum_search_latency_ms]),
    SUM(searches_terms[latency_event_count]),
    BLANK()
)

// Avg News Results per term
Term Avg News Results =
DIVIDE(
    SUM(searches_terms[sum_news_result_count]),
    SUM(searches_terms[result_events]),
    BLANK()
)

// ViewMore Click Rate for terms
Term ViewMore Rate % =
DIVIDE(
    SUM(searches_terms[clicks_viewmore]),
    SUM(searches_terms[click_count]),
    0
) * 100

// Average Click Position for terms (weighted average using SUM columns)
Term Avg Click Position =
```

```
DIVIDE(
    SUM(searches_terms[sum_click_position]),
    SUM(searches_terms[click_position_count]),
    BLANK()
)
```

Term Status Classification (Dynamic DAX Measures)

These measures calculate term status dynamically from aggregated totals, ensuring correct results regardless of date slicer or filter context. **Do not use pre-calculated status columns** from the parquet file as they cannot be correctly aggregated across multiple days.

```
// CTR Percentage - calculated from aggregated totals
Term CTR % =
DIVIDE(
    SUM(searches_terms[success_click_count]),
    SUM(searches_terms[search_count]),
    0
) * 100

// Null Rate Percentage - calculated from aggregated totals
Term Null Rate % =
DIVIDE(
    SUM(searches_terms[null_result_count]),
    SUM(searches_terms[result_events]),
    0
) * 100

// Average Results Shown - weighted average of results per search
// Uses sum/count pattern for correct aggregation across any date range
Avg Results Shown =
DIVIDE(
    SUM(searches_terms[sum_result_count]),
    SUM(searches_terms[result_events]),
    0
)

// Effectiveness Score - CTR minus weighted null penalty
Term Effectiveness Score =
[Term CTR %] - ([Term Null Rate %] * 0.5)

// Term Status - calculated from aggregated rates (safe for any date
range)
Term Status =
VAR NullRate = [Term Null Rate %]
VAR CTR = [Term CTR %]
RETURN
    SWITCH(
        TRUE(),
        NullRate > 50, "High Null Rate",
        CTR > 30, "High CTR",
```

```
        CTR < 10, "Low CTR",
        "Moderate CTR"
    )

// Term Status Sort Order - for proper sorting in visuals
Term Status Sort =
VAR NullRate = [Term Null Rate %]
VAR CTR = [Term CTR %]
RETURN
    SWITCH(
        TRUE(),
        NullRate > 50, 1,  -- High Null Rate first (worst)
        CTR < 10, 2,       -- Low CTR second
        CTR > 30, 4,       -- High CTR last (best)
        3                   -- Moderate CTR
    )
```

Status Classification Thresholds:

Status	Condition	Priority	Interpretation
High Null Rate	Null Rate > 50%	1 (worst)	Content gap - users search but find nothing
Low CTR	CTR < 10%	2	Results exist but don't match user intent
Moderate CTR	CTR 10-30%	3	Average performance, room for improvement
High CTR	CTR > 30%	4 (best)	Term is performing well

Why these measures are aggregation-safe:

- They use `SUM()` which correctly aggregates across any filter context
- Status is recalculated from totals, not read from a pre-stored daily value
- Works correctly whether viewing a single day, week, month, or all time

Term Age & Lifecycle Classification

Calculate the age of a term (elapsed days since first appearance) and classify into lifecycle buckets:

```
// Term Age - elapsed days since term first appeared
Term Age =
DATEDIFF(
    MIN(searches_terms[first_seen_date]),
    MAX(searches_terms[session_date]),
    DAY
) + 1

// Term Lifecycle Bucket - 5-stage classification
Term Lifecycle =
VAR Age = [Term Age]
RETURN
    SWITCH(
```

```
        TRUE(),
        Age <= 3, "New",
        Age <= 7, "Emerging",
        Age <= 14, "Establishing",
        Age <= 30, "Established",
        "Mature"
    )

// Term Lifecycle Sort Order - for proper sorting in visuals
Term Lifecycle Sort =
VAR Age = [Term Age]
RETURN
    SWITCH(
        TRUE(),
        Age <= 3, 1,
        Age <= 7, 2,
        Age <= 14, 3,
        Age <= 30, 4,
        5
    )
```

Lifecycle Classification:

Bucket	Days Since First Seen	Sort	Interpretation
New	1-3 days	1	Just emerged, monitor for trending
Emerging	4-7 days	2	Past initial spike, validate staying power
Establishing	8-14 days	3	Building consistent usage pattern
Established	15-30 days	4	Regular part of user vocabulary
Mature	31+ days	5	Long-standing, stable terms

Usage Notes:

- `Term Age` gives the number of days from when the term first appeared to the latest date in the current filter context
- Use `Term Lifecycle` for badges, icons, or conditional formatting in tables
- Filter by lifecycle stage to focus analysis (e.g., only "New" terms to spot trends)
- The calculation is relative to the filtered date range, making it safe for any slicer selection
- Set "Sort by column" on `Term Lifecycle` to use `Term Lifecycle Sort` for correct ordering

Analysis Use Cases:

- **Flash trends**: High volume in "New" but drops off in "Emerging"
- **Growing terms**: Steady increase from New → Emerging → Establishing
- **Seasonal terms**: Reappear periodically after being "Mature"
- **Core vocabulary**: Consistently high volume in "Mature" bucket

Term Seasonality Analysis

Detect recurring seasonal patterns in search terms (e.g., "performance review" spiking every November/December).

Data Preparation

The `searches_terms` table includes a `month_num` column (1-12) for monthly pattern analysis. Create a month name column for display:

```
// Calculated Column: Month Name (Model view → searches_terms → New
Column)
Month_Name =
FORMAT(DATE(2024, searches_terms[month_num], 1), "MMM")

// Calculated Column: Month Sort (for proper ordering)
Month_Sort = searches_terms[month_num]
```

Core Seasonality Measures

```
// Monthly Searches - for the selected term(s)
Monthly Searches =
SUM(searches_terms[search_count])

// Peak Month Volume - highest monthly average for a term
Peak Month Volume =
VAR TermFilter = SELECTEDVALUE(searches_terms[search_term])
VAR MonthlyAvg =
    ADDCOLUMNS(
        VALUES(searches_terms[month_num]),
        "AvgVol", CALCULATE(AVERAGE(searches_terms[search_count]))
    )
RETURN
    MAXX(MonthlyAvg, [AvgVol])

// Average Monthly Volume - baseline for concentration
Avg Monthly Volume =
AVERAGEX(
    VALUES(searches_terms[month_num]),
    CALCULATE(SUM(searches_terms[search_count]))
)

// Concentration Ratio - how much peak month exceeds average
// Values > 2.0 indicate seasonal patterns
Concentration Ratio =
DIVIDE([Peak Month Volume], [Avg Monthly Volume], 1)

// --- Data Confidence Measures ---
// These ensure seasonality classification is only applied when sufficient
data exists
```

```
// Potential Months - how many months COULD have data (first seen to last
seen)
Potential Months =
VAR FirstSeen = MIN(searches_terms[first_seen_date])
VAR LastSeen = MAX(searches_terms[session_date])
RETURN
    MAX(1, DATEDIFF(FirstSeen, LastSeen, MONTH) + 1)

// Months Active - how many distinct months actually have data
Months Active =
DISTINCTCOUNT(searches_terms[month_num])

// Month Coverage % - what percentage of potential months have data
Month Coverage % =
DIVIDE([Months Active], [Potential Months], 0) * 100

// Has Sufficient Data - requires 6+ months of history for reliable
seasonality
Has Sufficient Data =
[Potential Months] >= 6

// Total Searches for Term (for minimum threshold)
Term Total Searches =
SUM(searches_terms[search_count])

// --- Seasonality Classification with Data Confidence ---

// Seasonality Type - accounts for insufficient data
// Only classifies seasonality when:
// - Term has 6+ months of potential history
// - Term has at least 10 total searches
Seasonality Type =
VAR Ratio = [Concentration Ratio]
VAR HasEnoughHistory = [Potential Months] >= 6
VAR HasEnoughSearches = [Term Total Searches] >= 10
RETURN
    SWITCH(
        TRUE(),
        NOT HasEnoughSearches, "Low Volume",
        NOT HasEnoughHistory, "Insufficient Data",
        Ratio >= 3.0, "Highly Seasonal",
        Ratio >= 2.0, "Moderately Seasonal",
        Ratio >= 1.5, "Slightly Seasonal",
        "Consistent"
    )

// Seasonality Sort (for proper ordering in visuals)
// Sort order: Highly Seasonal first, then Insufficient Data last
Seasonality Sort =
VAR Ratio = [Concentration Ratio]
VAR HasEnoughHistory = [Potential Months] >= 6
VAR HasEnoughSearches = [Term Total Searches] >= 10
RETURN
    SWITCH(
```

```
        TRUE(),
        NOT HasEnoughSearches, 99,      -- Low Volume (hide at bottom)
        NOT HasEnoughHistory, 98,       -- Insufficient Data (near bottom)
        Ratio >= 3.0, 1,                -- Highly Seasonal
        Ratio >= 2.0, 2,                -- Moderately Seasonal
        Ratio >= 1.5, 3,                -- Slightly Seasonal
        4                               -- Consistent
    )
```

Recurrence Detection Measures

```
// Years Active - how many different years the term appeared
Years Active =
COUNTROWS(
    DISTINCT(
        SELECTCOLUMNS(
            searches_terms,
            "Year", YEAR(searches_terms[session_date])
        )
    )
)

// Peak Month - which month has highest volume
Peak Month =
VAR MonthlyTotals =
    ADDCOLUMNS(
        VALUES(searches_terms[month_num]),
        "Vol", CALCULATE(SUM(searches_terms[search_count]))
    )
VAR PeakMonthNum = MAXX(TOPN(1, MonthlyTotals, [Vol], DESC), [month_num])
RETURN
    FORMAT(DATE(2024, PeakMonthNum, 1), "MMMM")

// Recurring Term Flag - appeared in same month across 2+ years
Is Recurring =
VAR PeakMonthNum =
    MAXX(
        TOPN(1,
            ADDCOLUMNS(VALUES(searches_terms[month_num]), "Vol",
CALCULATE(SUM(searches_terms[search_count]))),
            [Vol], DESC),
        [month_num])
VAR YearsInPeakMonth =
    CALCULATE(
        COUNTROWS(
            DISTINCT(
                SELECTCOLUMNS(
                    searches_terms,
                    "Year", YEAR(searches_terms[session_date])
                )
            )
```

```
        ),
        searches_terms[month_num] = PeakMonthNum
    )
RETURN
    IF(YearsInPeakMonth >= 2, "Recurring", "Single Occurrence")

// Activity Density - % of days with activity since first seen
Activity Density % =
VAR DaysActive = DISTINCTCOUNT(searches_terms[session_date])
VAR TotalSpan = DATEDIFF(MIN(searches_terms[first_seen_date]),
MAX(searches_terms[session_date]), DAY) + 1
RETURN
    DIVIDE(DaysActive, TotalSpan, 0) * 100
```

Seasonality Classification

Type	Condition	Sort	Interpretation
Highly Seasonal	concentration ≥ 3.0	1	Very concentrated in 1-2 months (holiday, annual events)
Moderately Seasonal	concentration 2.0 - 3.0	2	Clear seasonal pattern (quarterly reviews, fiscal periods)
Slightly Seasonal	concentration 1.5 - 2.0	3	Some monthly variation, not strongly seasonal
Consistent	concentration < 1.5	4	Evenly distributed throughout year (core vocabulary)
Insufficient Data	potential_months < 6	98	Not enough observation time to determine pattern
Low Volume	total_searches < 10	99	Too few searches for reliable analysis

Note: The `Seasonality Type` measure includes data confidence checks automatically - it will show "Insufficient Data" or "Low Volume" when there isn't enough data to make reliable seasonality claims.

Answering Seasonality Questions

Q1: Which terms spike every November/December? (HR review season)

Visual: Table

Field	Well
search_term	Rows
[Monthly Searches]	Values
[Peak Month]	Values

Field	Well
[Concentration Ratio]	Values
[Is Recurring]	Values

Filters:

- Filter `Peak Month` to "November" or "December"
- Filter `[Concentration Ratio]` >= 2.0
- Sort by `[Monthly Searches]` descending

Q2: Which terms are one-time events vs recurring?

Visual: Stacked Bar Chart

Field	Well
[Is Recurring]	Axis
search_term	Legend (Top N = 10 by searches)
[Monthly Searches]	Values

Alternative: Table with conditional formatting

- Show `[Is Recurring]` with icon formatting (checkmark for recurring)
- Filter `[Years Active]` >= 2 to focus on multi-year terms

Q3: Show me terms with >3x concentration in Q4

Visual: Table

Field	Well
search_term	Rows
[Monthly Searches]	Values
[Peak Month]	Values
[Concentration Ratio]	Values
[Seasonality Type]	Values

Filters:

- Filter `month_num` to 10, 11, 12 (October-December)
- Filter `[Concentration Ratio]` >= 3.0
- Sort by `[Concentration Ratio]` descending

Q4: Monthly volume heatmap for a specific term

Visual: Matrix

Field	Well
Month_Name	Columns
YEAR(session_date)	Rows
[Monthly Searches]	Values

Configuration:

- Add conditional formatting (background color) on `[Monthly Searches]`
- Use Month_Sort for column ordering
- Filter to specific search_term using a slicer

Q5: Seasonal terms by business cycle

Visual: Clustered Bar Chart

Field	Well
[Seasonality Type]	Axis
[Monthly Searches]	Values

Drill-down: Add search_term to see which terms fall into each category

Additional Table: Terms by Quarter Peak

- Create calculated column: `Quarter = "Q" & ROUNDUP(searches_terms[month_num]/3, 0)`
- Group by Quarter to see Q1 (Jan-Mar), Q2 (Apr-Jun), Q3 (Jul-Sep), Q4 (Oct-Dec) patterns

---

Sample Dashboard Layout: Seasonality Analysis

```
|   (Matrix: Months × Years, colored by volume)                              |
|
|
+---------------------------------------------------------------------------+
|                                        |
|
|   Q4 Seasonal Terms                    |   Recurring vs One-Time     |
|   (Table: Terms peaking Oct-Dec)       |   (Stacked Bar)             |
|                                        |
|
+----------------------------------------+-----------------------------
+
```

Time-of-Day Pattern Analysis

The `searches_terms` table includes pre-calculated time distribution columns (`searches_morning`, `searches_afternoon`, `searches_evening`, `searches_night`) that reveal when specific terms are being searched. This can indicate regional usage patterns and help identify terms associated with specific time zones.

Time Period Definitions:

Column	Hours (CET)	Regional Alignment
searches_night	03-09	APAC business hours
searches_morning	09-16	CET business hours
searches_afternoon	16-22	Americas business hours
searches_evening	22-03	Dead time (low activity)

DAX Measures for Time-of-Day Analysis

Create these measures (Report view → Modeling tab → New measure):

Peak Time Period - Identifies which time of day a term is most searched:

```
Peak Time Period =
VAR MorningTotal = SUM(searches_terms[searches_morning])
VAR AfternoonTotal = SUM(searches_terms[searches_afternoon])
VAR EveningTotal = SUM(searches_terms[searches_evening])
VAR NightTotal = SUM(searches_terms[searches_night])
VAR MaxVal = MAX(MAX(MorningTotal, AfternoonTotal), MAX(EveningTotal,
NightTotal))
RETURN
SWITCH(
    TRUE(),
    MaxVal = 0, "No Data",
    NightTotal = MaxVal, "APAC (03-09)",
    MorningTotal = MaxVal, "CET (09-16)",
```

```
    AfternoonTotal = MaxVal, "Americas (16-22)",
    EveningTotal = MaxVal, "Dead Time (22-03)",
    "Unknown"
)
```

Peak Period Sort - For proper sorting in visuals:

```
Peak Period Sort =
VAR MorningTotal = SUM(searches_terms[searches_morning])
VAR AfternoonTotal = SUM(searches_terms[searches_afternoon])
VAR EveningTotal = SUM(searches_terms[searches_evening])
VAR NightTotal = SUM(searches_terms[searches_night])
VAR MaxVal = MAX(MAX(MorningTotal, AfternoonTotal), MAX(EveningTotal,
NightTotal))
RETURN
SWITCH(
    TRUE(),
    MaxVal = 0, 99,
    MorningTotal = MaxVal, 1,
    AfternoonTotal = MaxVal, 2,
    EveningTotal = MaxVal, 3,
    NightTotal = MaxVal, 4,
    99
)
```

Time Concentration % - How concentrated the searches are in the peak period:

```
Time Concentration % =
VAR MorningTotal = SUM(searches_terms[searches_morning])
VAR AfternoonTotal = SUM(searches_terms[searches_afternoon])
VAR EveningTotal = SUM(searches_terms[searches_evening])
VAR NightTotal = SUM(searches_terms[searches_night])
VAR Total = MorningTotal + AfternoonTotal + EveningTotal + NightTotal
VAR MaxVal = MAX(MAX(MorningTotal, AfternoonTotal), MAX(EveningTotal,
NightTotal))
RETURN
IF(Total > 0, DIVIDE(MaxVal, Total) * 100, 0)
```

Interpretation: 40%+ = strongly time-concentrated, 25-40% = normal distribution, <25% = evenly distributed

Primary Region - Infers likely user region based on search timing:

```
Primary Region =
VAR MorningTotal = SUM(searches_terms[searches_morning])
VAR AfternoonTotal = SUM(searches_terms[searches_afternoon])
VAR EveningTotal = SUM(searches_terms[searches_evening])
```

```
VAR NightTotal = SUM(searches_terms[searches_night])
VAR Total = MorningTotal + AfternoonTotal + EveningTotal + NightTotal
VAR MorningPct = DIVIDE(MorningTotal, Total)
VAR AfternoonPct = DIVIDE(AfternoonTotal, Total)
VAR EveningPct = DIVIDE(EveningTotal, Total)
VAR NightPct = DIVIDE(NightTotal, Total)
RETURN
SWITCH(
    TRUE(),
    Total = 0, "No Data",
    MorningPct > 0.4, "APAC",
    AfternoonPct > 0.4, "EMEA",
    EveningPct > 0.4 || NightPct > 0.4, "Americas",
    AfternoonPct + EveningPct > 0.6, "EMEA/Americas",
    MorningPct + AfternoonPct > 0.6, "APAC/EMEA",
    "Global"
)
```

Morning Share %, **Afternoon Share %**, **Evening Share %**, **Night Share %** - Individual period percentages:

```
Morning Share % =
VAR MorningTotal = SUM(searches_terms[searches_morning])
VAR Total = SUM(searches_terms[searches_morning]) +
SUM(searches_terms[searches_afternoon]) +
            SUM(searches_terms[searches_evening]) +
SUM(searches_terms[searches_night])
RETURN IF(Total > 0, DIVIDE(MorningTotal, Total) * 100, 0)
```

Create similar measures for Afternoon, Evening, and Night by changing the numerator variable

Questions You Can Answer

Q: Which terms are "CET terms" (European business hours)?

1. Create Table visual
2. Add: `search_term`, `Total Searches`, `Morning Share %`, `Peak Time Period`
3. Filter: `Peak Time Period = "CET (09–16)"`
4. Sort by `Morning Share %` descending

Q: Which terms are Americas-focused?

1. Create Table visual
2. Add: `search_term`, `Total Searches`, `Primary Region`, `Afternoon Share %`
3. Filter: `Primary Region = "Americas"`
4. Sort by `Afternoon Share %` descending

Q: What is the time distribution for a specific term?

1. Select term using slicer
2. Create Donut/Pie chart with:
    - Values: `Morning Share %`, `Afternoon Share %`, `Evening Share %`, `Night Share %`
3. Or use a 100% Stacked Bar with the four share measures

Q: Which terms have unusual time patterns?

1. Create Table visual
2. Add: `search_term`, `Total Searches`, `Time Concentration %`, `Peak Time Period`
3. Filter: `Time Concentration % > 50` (highly concentrated) or `Time Concentration % < 25` (evenly spread)
4. High concentration may indicate regional-specific terms; even distribution indicates global usage

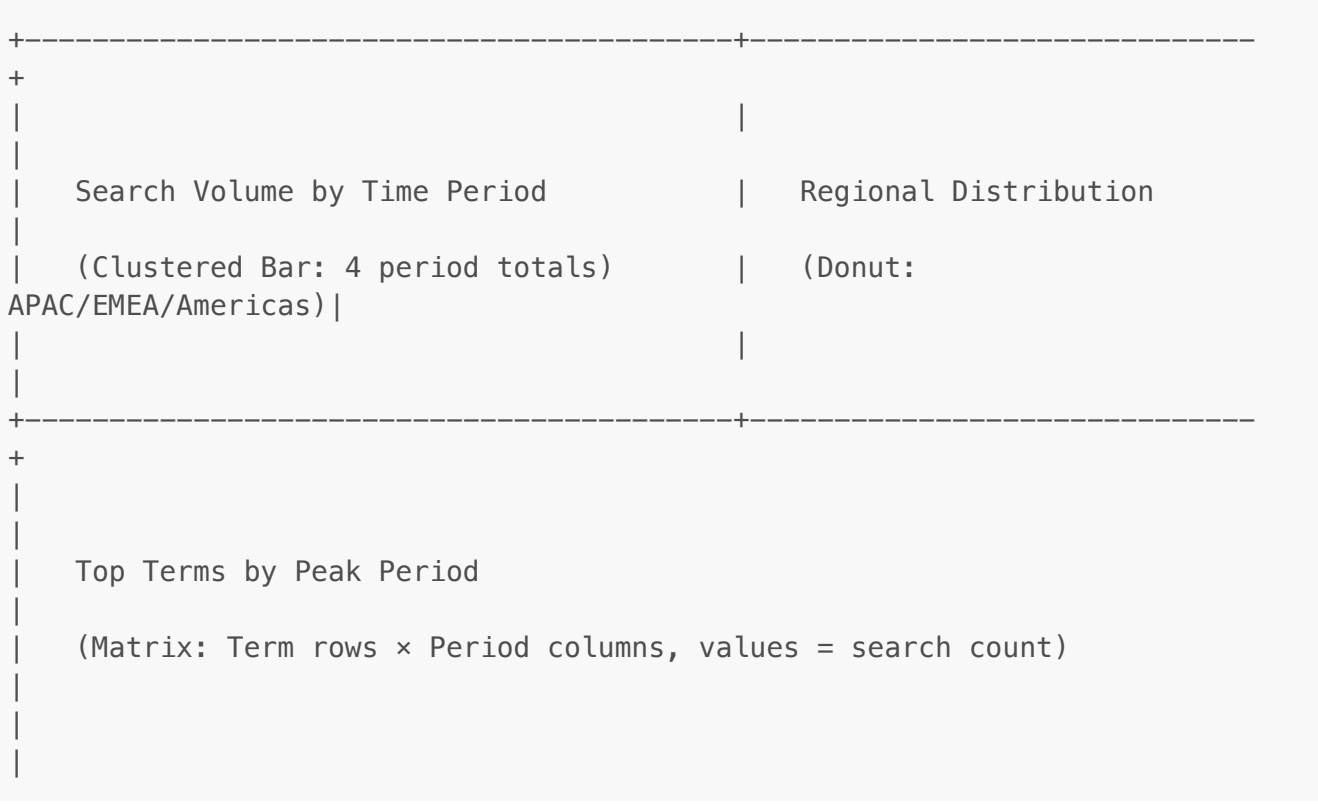
Q: How does search volume vary by time of day across all terms?

1. Create Clustered Bar Chart
2. Use separate bars for each time period total
3. Create measure for each:

```
Total Morning Searches = SUM(searches_terms[searches_morning])
Total Afternoon Searches = SUM(searches_terms[searches_afternoon])
Total Evening Searches = SUM(searches_terms[searches_evening])
Total Night Searches = SUM(searches_terms[searches_night])
```

4. Add all four measures to Values

Sample Dashboard Layout: Time-of-Day Analysis



```
+-------------------------------------------------------------------------
+
|                                          |
|
|   Time-Concentrated Terms                |   Global Terms
|
|   (Table: Concentration > 50%)           |   (Table: Concentration <
30%)|
|                                          |
|
+------------------------------------------+-----------------------------
+
```

Query Length vs Success Analysis

Use the `word_count` column to understand if longer queries perform better.

Step 1: Create the Bucket Calculated Column (Model view → `searches_terms` → New Column):

```
Query_Length_Bucket =
SWITCH(
    TRUE(),
    searches_terms[word_count] = 1, "1 word",
    searches_terms[word_count] = 2, "2 words",
    searches_terms[word_count] = 3, "3 words",
    searches_terms[word_count] = 4, "4 words",
    searches_terms[word_count] >= 5, "5+ words",
    "Unknown"
)
```

Step 2: Create the Sort Column (Model view → `searches_terms` → New Column):

```
Query_Length_Sort =
SWITCH(
    TRUE(),
    searches_terms[word_count] = 1, 1,
    searches_terms[word_count] = 2, 2,
    searches_terms[word_count] = 3, 3,
    searches_terms[word_count] = 4, 4,
    searches_terms[word_count] >= 5, 5,
    99
)
```

Step 3: Configure Sort Order

1. Go to **Model view** (left sidebar)
2. Select `Query_Length_Bucket` column
3. In **Column tools** ribbon → click **Sort by column**

4. Select `Query_Length_Sort`

Now your bar charts will sort correctly: 1 word → 2 words → 3 words → 4 words → 5+ words

Visualization: CTR by Query Length

- **Type**: Bar Chart
- **Setup**:
    1. Drag `Query_Length_Bucket` to **Axis**
    2. Drag `Term Success CTR %` measure to **Values**
- **Sort**: Already configured via "Sort by column"
- **Insight**: Do longer, more specific queries have higher CTR?

Visualization: Null Rate by Query Length

- **Type**: Bar Chart
- **Setup**:
    1. Drag `Query_Length_Bucket` to **Axis**
    2. Drag `Term Zero Result Rate %` measure to **Values**
- **Sort**: Already configured via "Sort by column"
- **Insight**: Do very specific (4+ word) queries fail more often?

Expected Pattern:

Query Length	Typical CTR	Typical Null Rate	Interpretation
1 word	Low (15-20%)	Low (5-8%)	Vague queries, many results but poor match
2 words	Medium (20-30%)	Low (4-6%)	Starting to get specific
3 words	High (25-40%)	Low (3-5%)	Optimal specificity
4 words	High (25-35%)	Medium (5-8%)	Specific, good match potential
5+ words	Medium (20-30%)	Higher (10-15%)	Very specific, may not find exact match

Advanced: Search Effectiveness Score

Create a combined metric that shows overall query performance:

```
// Uses success_click_count for actual content discovery measurement
Search Effectiveness Score =
VAR ctr = DIVIDE(SUM(searches_terms[success_click_count]),
SUM(searches_terms[search_count]), 0)
VAR nullRate = DIVIDE(SUM(searches_terms[null_result_count]),
SUM(searches_terms[result_events]), 0)
RETURN
(ctr * 100) - (nullRate * 50)
```

Interpretation: Higher score = better performance (high CTR, low null rate). Negative scores indicate problematic query lengths.

Term Outcome Classification

Create a calculated column to classify search terms by their outcome (Model view → `searches_terms` → New Column):

```
// Uses success_click_count for actual content discovery measurement
Term_Outcome =
VAR nullRate = DIVIDE([null_result_count], [result_events], 0)
VAR ctr = DIVIDE([success_click_count], [search_count], 0)
RETURN
SWITCH(
    TRUE(),
    nullRate = 1, "Zero Results",
    nullRate > 0.5, "Mostly No Results",
    ctr = 0, "No Clicks",
    ctr < 0.2, "Low CTR",
    "Success"
)
```

Recommended Filters for Query Analysis Page

Essential Filters:

Filter	Type	Purpose
`session_date`	Date Range Slicer	Filter to specific time period
Min Search Count	Numeric Slicer	Filter out low-volume terms (recommend: > 5)

Drill-Down Filters:

Filter	Type	Purpose
`Query_Length_Bucket`	Dropdown/Chips	Analyze specific query lengths
`Term_Outcome`	Dropdown/Chips	Focus on problem areas (Zero Results, No Clicks)
`is_new_term`	Toggle	Compare new vs established terms

Min Search Count Filter Setup:

1. Create a measure:

```
Show Term =
IF(SUM(searches_terms[search_count]) >= 5, 1, 0)
```

2. Filter visuals where `Show Term = 1` to hide low-volume terms

Recommended Page Layout:

```
+-------------------------------------------------+
| Filters:                                        |
| [Date Range    ] [Min Searches: 5] [Term Outcome] |
| [Query Length ▼]                                |
+-------------------------+-----------------------+
|                         |                       |
|  CTR by Query Length    |  Null Rate by Length  |
|  (Bar Chart)            |  (Bar Chart)          |
|                         |                       |
+-------------------------+-----------------------+
|                                                 |
|  Search Effectiveness Score by Query Length     |
|  (Bar Chart - combined metric)                  |
|                                                 |
+-------------------------------------------------+
```

Row 1: Top Search Terms

Chart 1: Top 20 Search Terms by Volume

- **Type**: Bar Chart (Horizontal)
- **Setup**:
  1. Drag `search_term` to **Y-axis**
  2. Drag `search_count` to **X-axis** → select **Sum**
  3. Add a **Top N filter** on `search_term` = Top 20 by Sum of `search_count`
- **Sort**: By X-axis values descending

Chart 2: Search Term Word Cloud (Optional)

- **Type**: Word Cloud (requires custom visual from AppSource)
- **Category**: `search_term`
- **Values**: `search_count`
- **Note**: Install "Word Cloud" from Power BI AppSource

Row 2: Problem Terms (Actionable Insights)

Chart 3: Terms with High Zero Result Rate

- **Type**: Table
- **Columns**: `search_term`, Sum of `search_count`, `Term Zero Result Rate %`
- **Filter**: `Term Zero Result Rate %` > 50 AND Sum of `search_count` > 5
- **Insight**: Content gaps - these terms need content created

Chart 4: High Volume + Low CTR Terms

- **Type**: Table
- **Columns**: `search_term`, Sum of `search_count`, `Term Success CTR %`
- **Filter**: `Term Success CTR %` < 20 AND Sum of `search_count` > 10 AND `Term Zero Result Rate %` < 50
- **Insight**: Relevance problems - results exist but don't match user intent

Row 3: Success Stories

Chart 5: High Performing Terms

- **Type**: Table
- **Columns**: `search_term`, Sum of `search_count`, `Term Success CTR %`
- **Filter**: `Term Success CTR %` > 50 AND Sum of `search_count` > 10
- **Insight**: Learn from these - what makes them successful?

Chart 6: Search Term Trend (Selected Term)

- **Type**: Line Chart
- **Setup**:
  1. Drag `session_date` to **X-axis**
  2. Drag `search_count` to **Y-axis** → select **Sum**
  3. Add a slicer for `search_term` to filter to specific terms
- **Insight**: Track popularity of specific terms over time

Row 4: KPI Cards

Card	Measure	Description
Unique Terms	`Unique Terms`	Total distinct search terms
Avg CTR	`Term Success CTR %`	Overall success click-through rate
Zero Result Rate	`Term Zero Result Rate %`	% of searches with no results
Top Term	First value of `search_term` sorted by `search_count`	Most searched term

Key Questions This Page Answers

1. **What are users searching for?** - Top search terms by volume
2. **What content is missing?** - Terms with high zero result rate
3. **Where is relevance poor?** - High volume terms with low CTR
4. **What's working well?** - High CTR terms to learn from
5. **Are search patterns changing?** - Term trends over time
6. **Do longer queries perform better?** - Query length vs CTR/null rate analysis

---

Page 5: Advanced Analytics

This page focuses on deeper behavioral insights: user cohorts, recovery patterns, and emerging trends.

Row 1: User Cohort Analysis

Chart 1: New vs Returning Users Trend

- **Type**: Stacked Area Chart
- **Setup**:
  1. Drag `date` to **X-axis**
  2. Add `new_users` and `returning_users` to **Values**
- **Insight**: Track user acquisition and retention patterns

Chart 2: First-Time vs Returning User Success Comparison

- **Type**: Clustered Bar Chart
- **Setup**:
  1. Create a bar for `First-Time User Success Rate %` measure
  2. Create a bar for `Returning User Success Rate %` measure
- **Alternative**: Use a KPI card comparison
- **Insight**: Do experienced users perform better? If so, there may be a learning curve for the search interface.

Row 2: Null Result Recovery Analysis

Chart 3: Null Recovery Rate KPI

- **Type**: KPI Card
- **Setup**: Use the `Null Recovery Rate %` measure
- **Target**: > 50% is good (users recovered despite initial failure)
- **Insight**: Measures resilience - can users succeed even when initial search fails?

Chart 4: Recovery Funnel

- **Type**: Funnel Chart
- **Setup**:
  1. Stage 1: Sessions with null results (`had_null_result = TRUE`)
  2. Stage 2: Recovered sessions (`recovered_from_null = TRUE`)
- **DAX for filtering**:

```
Sessions with Null =
CALCULATE(COUNTROWS(searches_journeys), searches_journeys[had_null_result]
= TRUE())

Sessions Recovered =
CALCULATE(COUNTROWS(searches_journeys),
searches_journeys[recovered_from_null] = TRUE())
```

- **Insight**: Visual representation of how many null-result sessions eventually succeed

Row 3: Session Flow Patterns

Chart 5: Tab Switching Analysis

- **Type**: Clustered Bar Chart
- **Setup**:
    1. Drag `journey_outcome` to **Axis**
    2. Create two measures: sessions with `had_tab_switch = TRUE` vs `FALSE`
- **Insight**: Do users who switch tabs have different success rates?

Chart 6: Click Categories Distribution

- **Type**: Histogram
- **Setup**:
    1. Drag `distinct_click_categories` to **Axis**
    2. Count of sessions to **Values**
- **Insight**: Most users should have 1-2 categories; many with 3+ may indicate confusion

Row 4: Term Trend Detection

Chart 7: New Terms Over Time

- **Type**: Line Chart
- **Setup**:
    1. Drag `session_date` to **X-axis**
    2. Use `New Terms Count` measure as **Values**
- **Insight**: Spike in new terms may indicate emerging topics or organizational changes

Chart 8: Recently Emerged Terms Table

- **Type**: Table
- **Columns**: `search_term`, `first_seen_date`, Sum of `search_count`, `Term Success CTR %`
- **Filter**: `first_seen_date` >= DATE (recent, e.g., last 7 days)
- **Sort**: By Sum of `search_count` descending
- **Insight**: Identify trending new topics that may need content attention

Key Insights This Page Provides

Question	Metric	Action
Are new users struggling?	Compare first-time vs returning success	Improve onboarding, add search tips
Can users recover from failed searches?	Null Recovery Rate %	If low, improve suggestions/synonyms
Are users confused about tab navigation?	Tab Switch Rate %	If high, reconsider tab organization

Question	Metric	Action
What new topics are emerging?	New Terms list	Create content for trending terms

Key Questions This Dashboard Answers

Daily Trends

1. **How many searches happen per day?** - Search volume trend
2. **Are users finding what they need?** - Click-through rate trend
3. **Is content missing?** - Null result rate trend
4. **What content types are most clicked?** - Click category breakdown

User Behavior

1. **What percentage of searches succeed?** - Journey outcome distribution
2. **How complex are search sessions?** - Session complexity breakdown
3. **Do users refine their searches?** - Reformulation rate
4. **How long do users take to decide?** - Result-to-click timing

System Performance

1. **How fast do results appear?** - Search-to-result timing
2. **Are there performance issues?** - Sessions in slow timing buckets

Temporal Patterns

1. **Which days are busiest?** - `day_of_week` in daily file
2. **When do users search?** - `searches_morning`, `searches_afternoon`, `searches_evening`, `searches_night`
3. **When is a specific term searched?** - Hour distribution per term in terms file

User Cohorts & Behavior

1. **Are new users struggling?** - Compare `First-Time User Success Rate %` vs `Returning User Success Rate %`
2. **How is user acquisition trending?** - `new_users` vs `returning_users` over time
3. **Can users recover from failed searches?** - `Null Recovery Rate %`
4. **Do users switch tabs to find content?** - `Tab Switch Rate %`, `distinct_click_categories`

Term Trends

1. **What new topics are emerging?** - Filter by `is_new_term = TRUE`, sort by volume
2. **When did a term first appear?** - `first_seen_date` in terms file
3. **Is search vocabulary expanding?** - `New Terms Count` measure over time

---

Recommended Alerts

Set up Power BI alerts for:

Alert	Threshold	Meaning
Null Rate Spike	> 15%	Content gap or search issues
Session Success Drop	< 30%	Results not matching user intent
Session Abandonment Spike	> 70%	Results showing but not useful
Slow Results	> 10% in "> 5s" bucket	System performance issue

Color Scheme Recommendations

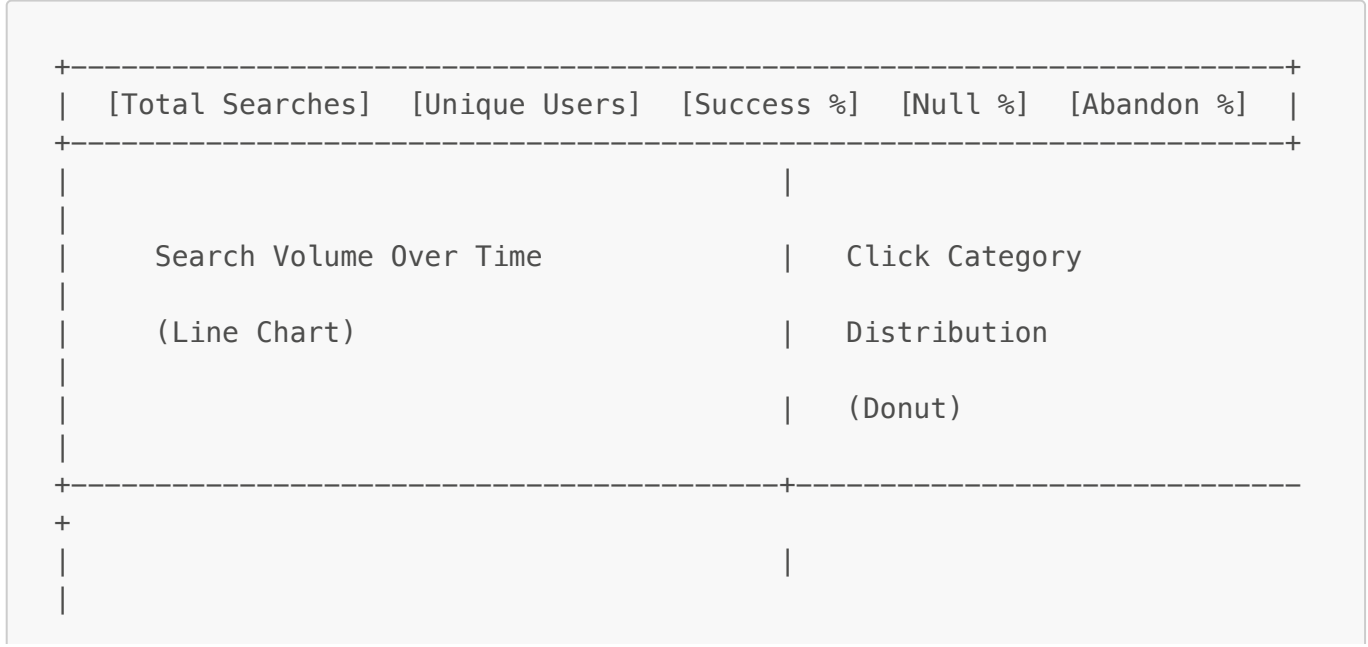
Metric Type	Good	Warning	Bad
Session Success Rate %	> 40% (Green)	25-40% (Yellow)	< 25% (Red)
Null Rate %	< 5% (Green)	5-15% (Yellow)	> 15% (Red)
Session Abandonment %	< 50% (Green)	50-70% (Yellow)	> 70% (Red)
Success Click Rate %	> 30% (Green)	15-30% (Yellow)	< 15% (Red)

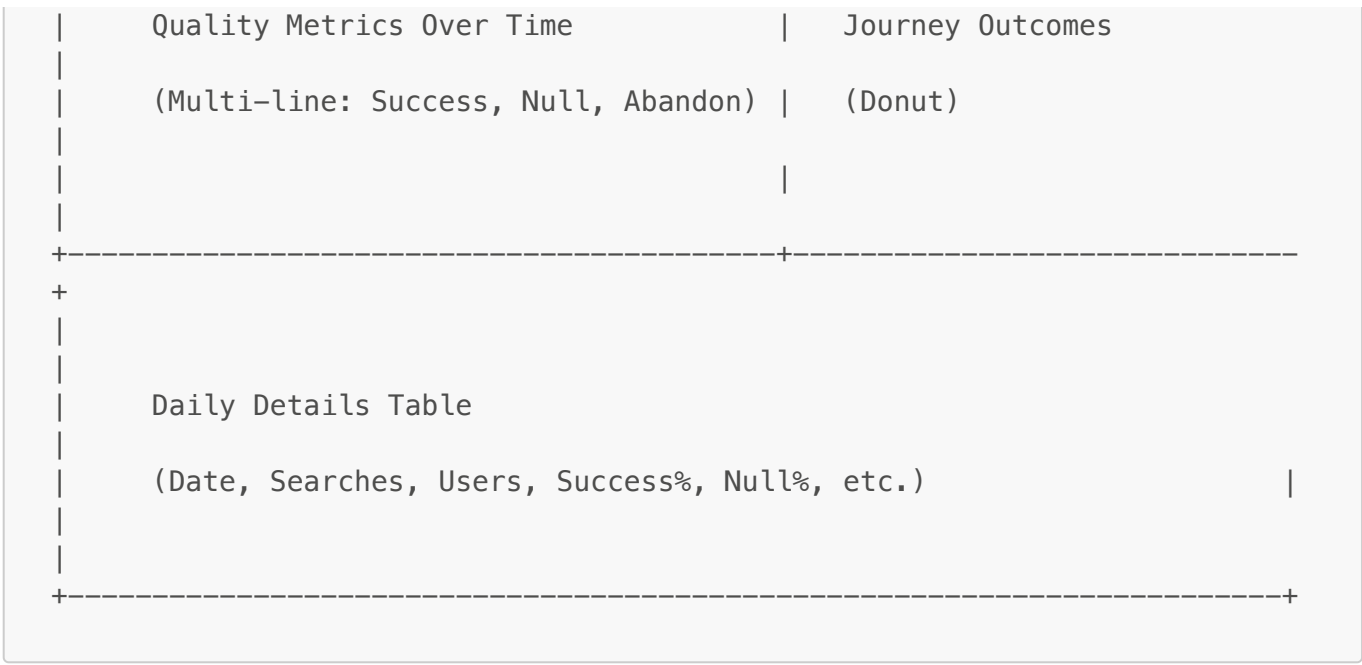
Tips for Effective Analysis

1. **Compare weekdays vs weekends** - Search patterns differ
2. **Filter by time of day** - Peak hours may show different behavior
3. **Track reformulation rate** - High rates suggest search UI/relevance issues
4. **Monitor null results** - These are opportunities to improve content
5. **Watch for "quick clicks" (< 2s)** - May indicate good results OR users just clicking first result
6. **Look for "extended sessions" (> 5 min)** - Users may be struggling to find content

---

Sample Report Layout




Quality Metrics Over Time
(Multi-line: Success, Null, Abandon)
Journey Outcomes
(Donut)
Daily Details Table
(Date, Searches, Users, Success%, Null%, etc.)