Title:

HIMALAYAN EXPEDITIONS

Table of Contents

1. **Title Page**
   - Course Name, Group Members, Submission Date
2. **Step 1: Data Selection & Exploration**
   - Dataset Overview (Himalayan Expedition – Kaggle)
   - peaks.csv Summary
   - exped.csv Summary
   - members.csv Summary
   - Data Challenges
   - Business Questions
3. **Step 2: Data Preprocessing & ETL**
   - Tools & Workflow (Alteryx)
   - Fact & Dimension Table Creation
   - Output Files
   - ETL Challenges & Solutions
4. **Step 3: SSAS Tabular Model & Snowflake**
   - Data Upload & Table Creation
   - Himalayan_Model View
   - SQL Queries
   - Technical Issues & Fixes
5. **Step 4: PySpark Analysis**
   - Query 1: Peak Height vs Success
   - Query 2: Correlation (Height vs Success Rate)
   - Query 3: Success & Team Size Binning
   - Query 4: Season Frequency
   - Query 5: Nation Ranking
6. **Step 5: Visualization in Power BI**
   - Peak Height vs Expeditions
   - Success Trends Over Time
   - Most Dangerous Peaks
   - Country-wise Expedition Patterns
   - Oxygen Use & Safety Impact
7. **Conclusion**

Step 1: Data Selection & Exploration

Himalayan Expedition Dataset

https://www.kaggle.com/datasets/siddharth0935/himalayan-expeditions

This is the Himalayan Expedition Dataset, which holds information about different members and teams from across the world which dares to climb the gigantic mountains of the Himalayas. Their success stories and

DATASET DESCRIPTION

● = Columns Removed ● = Non-calculative columns ● = Primary Key ▭ = Columns Combined

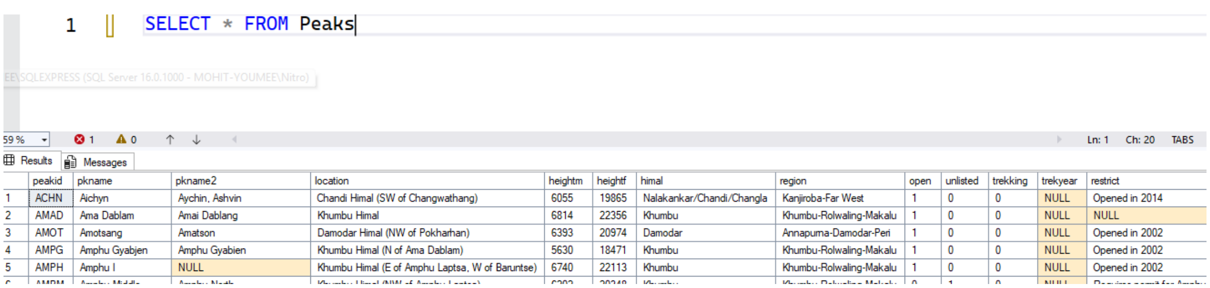
(pre-processing done in step 2 alteryx)

1. peaks.csv - Information About Himalayan Peaks (480 rows)

This file contains data about the peaks in the Himalayas, identified uniquely by peakid. Here's what each column represents and how it might be useful for your project:

- **peakid**: A unique identifier for each peak. This is the primary key linking this file to exped.csv and members.csv, enabling you to track which expeditions and members are associated with a specific peak.
- **pkname**: The primary name of the peak (e.g., Everest, K2). Useful for identifying and referencing peaks in your analysis or visualizations.
- **pkname2**: An alternative or secondary name for the peak. This could help resolve naming inconsistencies or provide cultural context.
- **location**: The geographical location of the peak. Great for mapping peaks or analyzing regional climbing patterns. 1
- **heightm**: Height of the peak in meters. Essential for comparing peak difficulty or studying altitude-related trends.
- **heightf**: Height in feet. Useful if your audience prefers imperial units or for cross-referencing with other datasets.
- **himal**: The specific Himalayan range (e.g., Everest Himal). Allows you to group peaks by range for regional analysis.
- **region**: A broader regional classification. Useful for higher-level geographical studies or regulatory analysis.
- **open**: Indicates if the peak is open for climbing. Key for understanding accessibility and its impact on expedition frequency.

- **unlisted**: Possibly marks peaks not officially listed. It could highlight lesser-known peaks or data gaps.
- **trekking**: Information about trekking availability or routes. Useful for studying trekking versus climbing activities.
- **trekyear**: Year trekking was first allowed or recorded. Helps trace the history of peak accessibility.
- **restrict**: Climbing restrictions (e.g., permits required). Critical for analyzing regulatory impacts on expeditions.
- **phost**: Likely the host country or entity managing the peak. Useful for studying jurisdictional influences. Binray - Nepal only, Nepal & China
- **pstatus**: Status of the peak (e.g., climbed, unclimbed). Great for historical analysis or identifying unexplored peaks.
    - **pyear**: Year of the first climb or significant event. Key for historical timelines or pioneer studies. - Replace null with 0
    - **pseason**: Season of the first climb. Useful for seasonal trend analysis. Replace null with 0
    - **pmonth**: Month of the first climb. Adds granularity to seasonal data. Replace null with 0
    - **pday**: Day of the first climb. Precise historical data for detailed records. Replace null with 0
    - **pexpid**: Expedition ID of the first climb. Links to exped.csv for details on the pioneering expedition.Replace null with 0
    - **pcountry**: Country of the first expedition. Enables nationality-based historical analysis. Replace null with 0
    - **psummiters**: Number of summiters or summit-related data. Useful for measuring peak popularity or difficulty.Replace null with N/A
- **psmtnote**: Notes about the peak or first ascent. Provides qualitative context for anomalies or special cases.



	peakid	pkname	pkname2	location	heightm	heightf	himal	region	open	unlisted	trekking	trekyear	restrict
1	ACHN	Aichyn	Aychin, Ashvin	Chandi Himal (SW of Changwathang)	6055	19865	Nalakankar/Chandi/Changla	Kanjiroba-Far West	1	0	0	NULL	Opened in 2014
2	AMAD	Ama Dablam	Amai Dablang	Khumbu Himal	6814	22356	Khumbu	Khumbu-Rolwaling-Makalu	1	0	0	NULL	NULL
3	AMOT	Amotsang	Amatson	Damodar Himal (NW of Pokharhan)	6393	20974	Damodar	Annapurna-Damodar-Peri	1	0	0	NULL	Opened in 2002
4	AMPG	Amphu Gyabjen	Amphu Gyabien	Khumbu Himal (N of Ama Dablam)	5630	18471	Khumbu	Khumbu-Rolwaling-Makalu	1	0	0	NULL	Opened in 2002
5	AMPH	Amphu I	NULL	Khumbu Himal (E of Amphu Laptsa, W of Baruntse)	6740	22113	Khumbu	Khumbu-Rolwaling-Makalu	1	0	0	NULL	Opened in 2002

This file is foundational for understanding the peaks themselves—their physical traits, climbing history, and accessibility.

2. exped.csv - Expedition Details (11417 rows)

This file is the core of expedition data, linked by expid (unique identifier) and peakid (foreign key to peaks.csv). Here's what each column means:

- **expid**: Unique identifier for each expedition. The primary key connects to members.csv.
- **peakid**: The peak targeted by the expedition. Links to peaks.csv for peak-specific details.
- **year**: Year of the expedition. Essential for temporal trend analysis.
- **season**: Season of the expedition (e.g., spring, autumn). Useful for seasonal success or risk studies.
- **host**: Host country or organization. Key for understanding logistical or political influences.
- **route1, route2, route3, route4**: Up to four routes planned or taken. Great for route popularity or success analysis. Replace null with N/A
- **nation**: Nationality of the expedition team. Enables demographic or national comparisons. Map
- **leaders**: Expedition leaders' names. Useful for studying leadership impact. Replace null with N/A
- **sponsor**: Expedition sponsor. It could reveal funding influences on success. Replace null with N/A
- **success1, success2, success3, success4**: Success indicators for different routes or goals. Critical for success rate analysis.
- **ascent1, ascent2, ascent3, ascent4**: Details of ascents (e.g., dates, routes). Adds depth to success data. Replace null with N/A
- **claimed**: Whether success was claimed. Useful for verifying expedition outcomes.
- **disputed**: If the claim was contested. Highlights reliability issues in data.
- **countries**: Countries involved. Useful for multinational expedition studies. Replace null with N/A
- **approach**: Approach route or method to the peak. Key for logistical analysis. Replace null with N/A
- **bcdate**: Base camp establishment date. It marks the start points. Replace null with N/A
- **smtdate**: Summit date. Critical for success, timing analysis. Replace null with N/A
    - New column
- **smttime**: Summit time. Adds precision to the summit records. Replace null with N/A
- **smtdays**: Days to summit from base camp. Measures expedition efficiency. Replace null with N/A
- **totdays**: Total expedition days. Useful for overall effort analysis.
- **termdate**: Termination date. Indicates when the expedition ended.
- **termreason**: Reason for ending (e.g., success, failure, weather). Key for risk or failure studies.
- **termnote**: Notes on termination. Provides context for anomalies. Replace null with "" and Combine both columns
- **highpoint**: Highest point reached. Useful if the summit wasn't achieved.
- **traverse**: Whether a traverse was completed. Highlights unique expedition types.
- **ski**: Skiing involvement. Identifies specialized expeditions.
- **parapente**: Use of paragliders. Another specialized activity indicator.
- **camps**: Camp setup details. Useful for logistical planning studies.
- **rope**: Use of ropes or fixed lines. Indicates technical climbing aspects.
- **totmembers**: Total expedition members. Key for team size analysis.
- **smtmembers**: Members who submitted. Measures individual success within teams.
- **mdeaths**: Member deaths. Critical for risk assessment.
- **tothired**: Total hired personnel (e.g., porters). Useful for support staff analysis.
- **smthired**: Hired personnel who submitted. Highlights their contributions.
- **hdeaths**: Hired personnel deaths. Adds to risk data.
- **nohired**: Possibly indicates no hired staff. Clarifies team composition.

- **o2used**: Use of supplemental oxygen. Key for studying its impact on success or safety.
- **o2none to o2unkwn**: Oxygen use details (none, climbing, descent, sleep, medical, taken but unused, unknown). Granular data for oxygen studies.
- **othersmts**: Other summits achieved. Shows additional expedition achievements.
- **campsites**: Campsite locations. Useful for route and logistics mapping.
- **accidents**: Accident details. Essential for safety analysis.
- **achievment**: Expedition achievements. Qualitative success data. Replace null with N/A
- **agency**: Organizing agency. It could indicate commercial versus independent expeditions. Replace null with N/A
- **comrte**: Commercial route indicator. Useful for commercialization studies.
- **stdrte**: Standard route. Helps identify common paths.
- **primrte**: Primary route. Key for route preference analysis.
- **primmem**: Possibly primary members. Needs clarification, but could highlight key participants.
- **primref**: Primary reference source. Useful for data validation.
- **primid**: Primary ID (possibly expedition-related). Needs clarification.
- **chksum**: Checksum for data integrity. Ensures data accuracy.

```sql
SELECT * FROM Peaks
```

	peakid	pkname	pkname2	location	heightm	heightf	himal	region	open	unlisted	trekking	trekyear	restrict
1	ACHN	Aichyn	Aychin, Ashvin	Chandi Himal (SW of Changwathang)	6055	19865	Nalakankar/Chandi/Changla	Kanjiroba-Far West	1	0	0	NULL	Opened in 2014
2	AMAD	Ama Dablam	Amai Dablang	Khumbu Himal	6814	22356	Khumbu	Khumbu-Rolwaling-Makalu	1	0	0	NULL	NULL
3	AMOT	Amotsang	Amatson	Damodar Himal (NW of Pokharhan)	6393	20974	Damodar	Annapurna-Damodar-Peri	1	0	0	NULL	Opened in 2002
4	AMPG	Amphu Gyabjen	Amphu Gyabien	Khumbu Himal (N of Ama Dablam)	5630	18471	Khumbu	Khumbu-Rolwaling-Makalu	1	0	0	NULL	Opened in 2002
5	AMPH	Amphu I	NULL	Khumbu Himal (E of Amphu Laptsa, W of Baruntse)	6740	22113	Khumbu	Khumbu-Rolwaling-Makalu	1	0	0	NULL	Opened in 2002

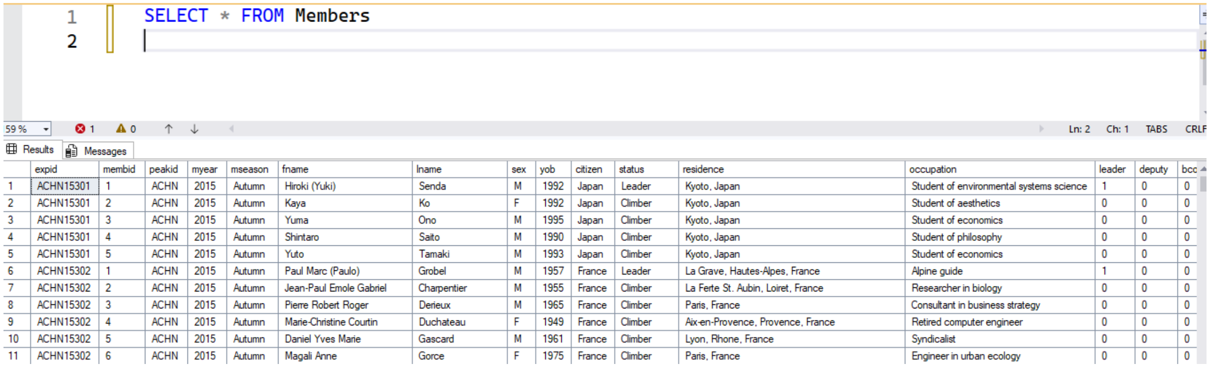
This file is ideal for analyzing expedition logistics, success rates, routes, team dynamics, and risks.

3. members.csv - Individual Expedition Members (88965 rows)

This file tracks individual participants, linked by expid (to exped.csv), peakid (to peaks.csv), and membid (unique member ID). Here's the breakdown:

- **expid**: Expedition ID. Links to exped.csv.
- **membid**: Unique member identifier. Primary key for this file.
- **peakid**: Peak ID. Links to peaks.csv.
- **myear**: Expedition year. Matches the year in the exped.csv.
- **mseason**: Expedition season. Matches the season in exped.csv.
- **fname**: First name. Identifies the member.
- **lname**: Last name. Completes member identification.
- **sex**: Gender. Useful for demographic analysis. Handle 'm', 'M', 'Male'
- **yob**: Year of birth. Allows age calculations. Calculate Age
- **citizen**: Citizenship. Enables nationality studies. Map
- **status**: Role or outcome (e.g., climber, deceased). Clarifies member involvement.

- **residence**: Place of residence. Adds geographic context.
- **occupation**: Job or profession. Could correlate with experience or skills.
- **leader**: Whether the member was a leader. Key for leadership impact studies.
- **deputy**: Deputy leader status. Another leadership role indicator.
- **bconly**: Base camp only (didn't climb). Identifies support versus climbing roles.
- **nottobc**: Didn't reach base camp. Indicates early dropouts.
- **support**: Support role (e.g., doctor). Highlights non-climbing contributions.
- **disabled**: Disability status. Useful for inclusivity studies.
- **hired**: Hired personnel status. Distinguishes climbers from support staff.
- **sherpa**: Sherpa status. Key for local involvement analysis.
- **tibetan**: Tibetan status. Another local demographic indicator.
- **msuccess**: Summit success. Measures individual achievement.
- **mclaimed**: Summit claim. Verifies success reports.
- **mdisputed**: Disputed claim. Highlights reliability issues.
- **msolo**: Solo ascent. Identifies unique achievements.
- **mtraverse**: Traverse completed. Another specialized feat.
- **mski**: Skiing involved. Indicates specialized skills.
- **mparapente**: Paraglider use. Another niche activity.
- **mspeed**: Speed ascent. Highlights exceptional performance.
- **mhighpt**: Highest point reached. Useful for partial success analysis. Replace null with 0
- **mperhighpt**: Personal high point. Adds individual context.
- **msmtdate1, msmtdate2, msmtdate3**: Summit dates (multiple attempts). Tracks individual summit timing.
- **msmttime1, msmttime2, msmttime3**: Summit times. Adds precision.Replace null with N/A
- **mroute1 to mroute3**: Routes taken. Useful for individual route analysis.
- **mascent1 to mascent3**: Ascent details. Provides depth to success data.
- **mo2used**
- **mo2none to mo2note**: Oxygen use details (used, none, climbing, descent, sleep, medical, notes). Granular data for oxygen impact studies.
- **death**: Death status. Critical for risk analysis.
- **deathdate**: Date of death. Adds temporal context.
- **deathtime**: Time of death. Precise incident data.
- **deathtype**: Cause of death (e.g., avalanche). Key for safety studies.
- **deathhgtm**: Height of death in meters. Correlates risk with altitude.
- **deathclass**: Death classification. Adds detail to incident analysis.
- **msmtbid**: Possibly summit bid ID. Needs clarification but could track attempts.
- **msmtterm**: Summit termination reason. Explains individual failures.
- **hcn**: Unclear, possibly health condition note. Requires metadata for clarity.
- **mchksum**: Checksum for data integrity. Ensures accuracy.



```
SELECT * FROM Members
```

	expid	membid	peakid	myear	mseason	fname	lname	sex	yob	citizen	status	residence	occupation	leader	deputy	bcc
1	ACHN15301	1	ACHN	2015	Autumn	Hiroki (Yuki)	Senda	M	1992	Japan	Leader	Kyoto, Japan	Student of environmental systems science	1	0	0
2	ACHN15301	2	ACHN	2015	Autumn	Kaya	Ko	F	1992	Japan	Climber	Kyoto, Japan	Student of aesthetics	0	0	0
3	ACHN15301	3	ACHN	2015	Autumn	Yuma	Ono	M	1995	Japan	Climber	Kyoto, Japan	Student of economics	0	0	0
4	ACHN15301	4	ACHN	2015	Autumn	Shintaro	Saito	M	1990	Japan	Climber	Kyoto, Japan	Student of philosophy	0	0	0
5	ACHN15301	5	ACHN	2015	Autumn	Yuto	Tamaki	M	1993	Japan	Climber	Kyoto, Japan	Student of economics	0	0	0
6	ACHN15302	1	ACHN	2015	Autumn	Paul Marc (Paulo)	Grobel	M	1957	France	Leader	La Grave, Hautes-Alpes, France	Alpine guide	1	0	0
7	ACHN15302	2	ACHN	2015	Autumn	Jean-Paul Emole Gabriel	Charpentier	M	1955	France	Climber	La Ferte St. Aubin, Loiret, France	Researcher in biology	0	0	0
8	ACHN15302	3	ACHN	2015	Autumn	Pierre Robert Roger	Derieux	M	1965	France	Climber	Paris, France	Consultant in business strategy	0	0	0
9	ACHN15302	4	ACHN	2015	Autumn	Marie-Christine Courtin	Duchateau	F	1949	France	Climber	Aix-en-Provence, Provence, France	Retired computer engineer	0	0	0
10	ACHN15302	5	ACHN	2015	Autumn	Daniel Yves Marie	Gascard	M	1961	France	Climber	Lyon, Rhone, France	Syndicalist	0	0	0
11	ACHN15302	6	ACHN	2015	Autumn	Magali Anne	Gorce	F	1975	France	Climber	Paris, France	Engineer in urban ecology	0	0	0

Challenges and Solutions

- **Data type errors:** Handled irregular data types in the CSV files (e.g., requiring nvarchar(50) to be changed to nvarchar(200) and tinyint to smallint), which caused errors during dataset insertion into SSMS.
- **Too Many Columns:** Identifying only the necessary columns from the extensive dataset for the project and queries required careful selection.
- **Logical Column Selection:** Manually selecting and understanding the context of each column (e.g., peakid, success1, mdeaths) was time-intensive and prone to misinterpretation.

Business Questions for Analysis

1. How does the height of a peak correlate with the number of expeditions and their success rates?

2. What are the trends in expedition success rates over time, and how do they vary by season?

3. Which of the peaks are considered most dangerous for the trek?

4. Which countries have the most expeditions, and how does their success rate compare to others?

5. How does the use of supplemental oxygen affect success rates and safety?

Step 2: Data Preprocessing & ETL

Data Sources

Input files: peaks.csv (peak details), exped.csv (expedition data), members.csv (member info)

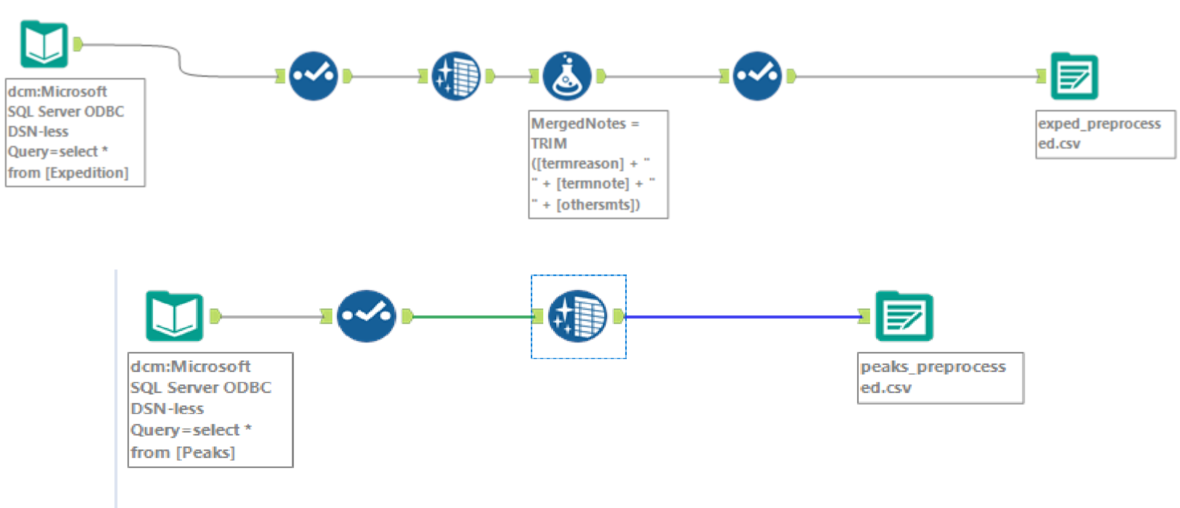
Alteryx Workflow

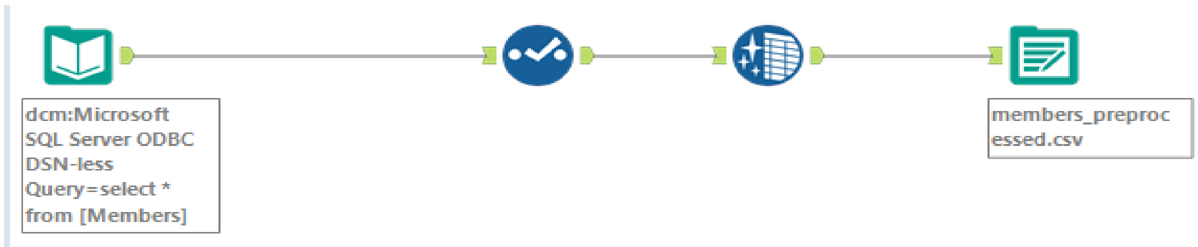
- **Input and Cleaning**: Used Input Data and Select Tools to import CSVs, rename columns (e.g., peakid), and set data types (e.g., heightm as Float).
- **Joining and Integration**: Joined files via Join and Union Tools on peakid, expid, membid, replacing missing values with NULL using Data Cleansing Tool.
- **Fact Table Aggregation**: Summarized data with Summarize Tool for Fact_Expeditions.csv (columns: peakid, year, expid, season, himal, region, nation, Sum_totmembers, Sum_smtmembers, Sum_mdeaths, Avg_Success_Rate, Avg_Death_Rate, Avg_smtdays, Avg_totdays, Avg_heightm, Count, Max_mhighpt). Filtered outliers with the Filter Tool.
- **Dimension Tables**: Created Dim_Peaks.csv (13 columns: peakid, pkname, etc.), Dim_Expeditions.csv (9 columns: expid, year, etc.), and Dim_Members.csv (10 columns: membid, fname, etc.) using Select and Unique Tools, which will be used in step 3 for more queries.
- **Validation and Export**: Previewed data with the Browse Tool, exported CSVs with the Output Data Tool to the VM directory.

Challenges and Solutions

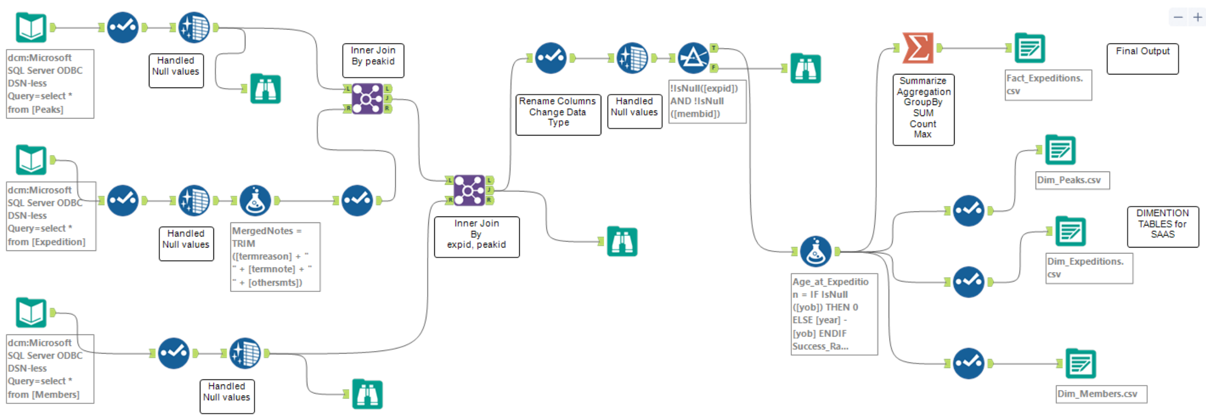
- **Inconsistent Data**: Converted Max_mhighpt ('True'/'False') to VARCHAR for flexibility.
- **Missing Values**: Handled with full joins and NULL replacement.
- **Large Data**: Filtered outliers to manage the 88,911-row dataset.

Pre-Processing pipelines:





Final pipeline:



Outputs:

- The **Fact_Expeditions.csv** file contains aggregated expedition data, including the total number of summiteers, average success rates, and the highest peak reached, providing a comprehensive summary of expedition outcomes per peak and year.
- **Dim_Peaks.csv l**ists unique peaks with attributes like pkname and heightm, serving as a reference table for peak-specific details.
- **Dim_Expeditions.csv** provides unique expedition records with details like season and nation, enabling seasonal and national analysis.
- **Dim_Members.csv** includes distinct member profiles (e.g., fname, sex), supporting demographic studies.

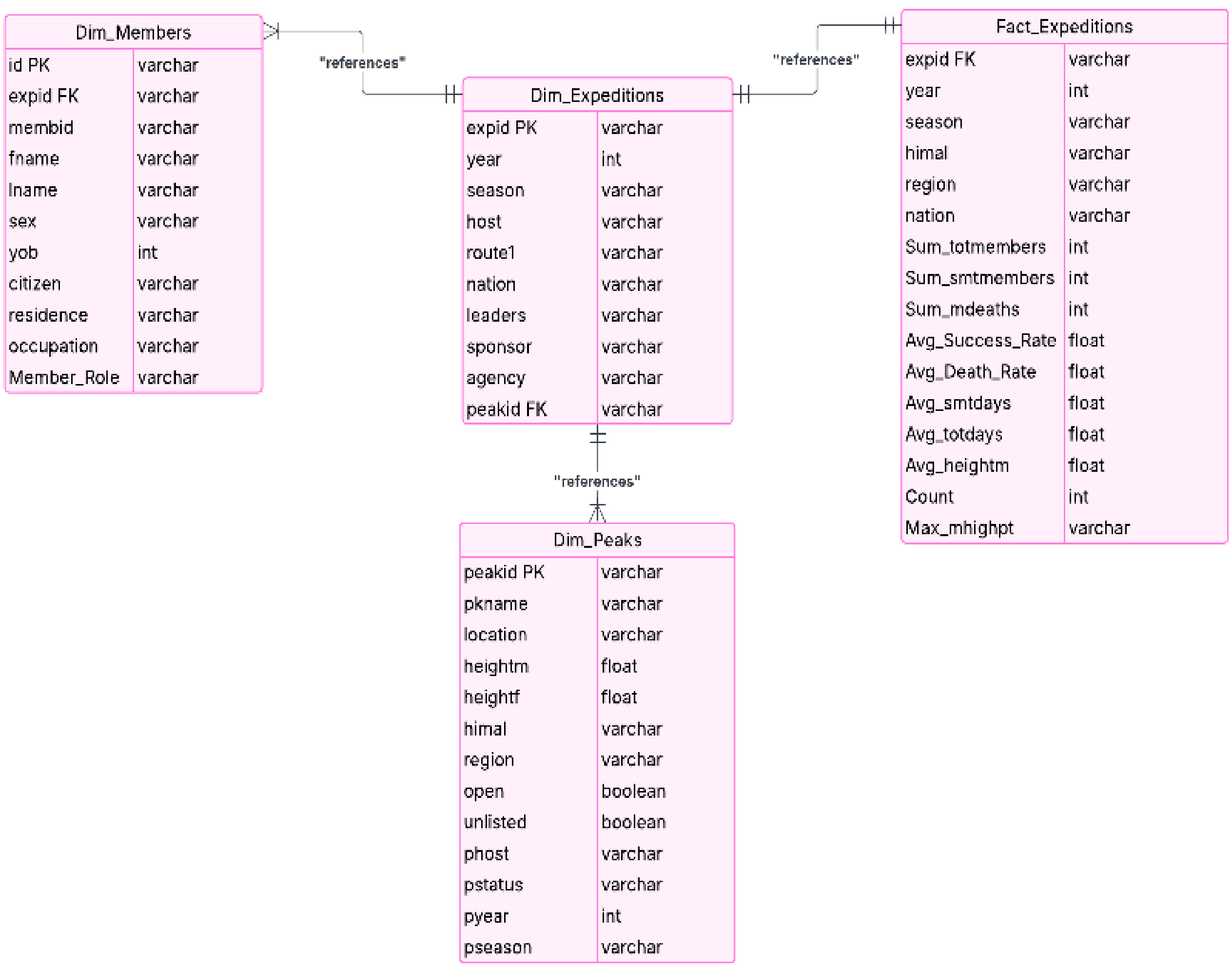
Step 3: SSAS Tabular Model

Data Sources

Input files from Step 2, transferred to /home/id101567404/bigdata_project/, included:

- **Fact_Expeditions.csv** (fact table with 17 columns: peakid, year, etc..).
- **Dim_Peaks.csv** (13 columns: peakid, pkname, etc..).
- **Dim_Expeditions.csv** (9 columns: expid, year, etc..).
- **Dim_Members.csv** (10 columns: membid, fname, etc.).

ERD Diagram for out Database



Snowflake Workflow

- **Setup and Loading**: Connected to Snowflake using snowsql, Mohitpanchasara ID.

```
id101567404@bigdata-04:~$ snowsql -a odivqdz-km75732 -u mohitpanchasara
Password:
* SnowSQL * v1.4.0
Type SQL statements or !help
mohitpanchasara#COMPUTE_WH@(no database).(no schema)>
```

- Created **HIMALAYAN_DB** and **PUBLIC** schema, and a my_stage for file uploads.

```
mohitpanchasara#COMPUTE_WH@(no database).(no schema)>CREATE DATABASE IF NOT EXISTS HIMALAYAN_DB;
                                                      USE DATABASE HIMALAYAN_DB;
                                                      CREATE SCHEMA IF NOT EXISTS PUBLIC;
                                                      USE SCHEMA PUBLIC;
+-------------------------------------------------+
| status                                          |
|-------------------------------------------------|
| HIMALAYAN_DB already exists, statement succeeded. |
+-------------------------------------------------+
1 Row(s) produced. Time Elapsed: 0.130s
+--------------------------------+
| status                         |
|--------------------------------|
| Statement executed successfully. |
+--------------------------------+
1 Row(s) produced. Time Elapsed: 0.099s
+-------------------------------------------+
| status                                    |
|-------------------------------------------|
| PUBLIC already exists, statement succeeded. |
+-------------------------------------------+
1 Row(s) produced. Time Elapsed: 0.101s
+--------------------------------+
| status                         |
|--------------------------------|
| Statement executed successfully. |
+--------------------------------+
1 Row(s) produced. Time Elapsed: 0.103s
mohitpanchasara#COMPUTE_WH@HIMALAYAN_DB.PUBLIC>
```

- Then uploaded CSVs with PUT commands.
- Created tables with matching columns and attempted COPY INTO loads, resolving errors (e.g., PARSE_HEADER vs. SKIP_HEADER conflicts) by recreating the CSV_FORMAT file format with PARSE_HEADER = TRUE.

```
mohitpanchasara#COMPUTE_WH@HIMALAYAN_DB.PUBLIC>SELECT CURRENT_DATABASE(), CURRENT_SCHEMA();
+--------------------+------------------+
| CURRENT_DATABASE() | CURRENT_SCHEMA() |
|--------------------+------------------|
| HIMALAYAN_DB       | PUBLIC           |
+--------------------+------------------+
1 Row(s) produced. Time Elapsed: 0.119s
```

- Schema

```sql
CREATE TABLE `Dim_Peaks` (
  `peakid` varchar(255) PRIMARY KEY,
  `pkname` varchar(255),
  `location` varchar(255),
  `heightm` float,
  `heightf` float,
  `himal` varchar(255),
  `region` varchar(255),
  `open` boolean,
  `unlisted` boolean,
  `phost` varchar(255),
  `pstatus` varchar(255),
  `pyear` int,
  `pseason` varchar(255)
);
```

```sql
CREATE TABLE `Fact_Expeditions` (
  `expid` varchar(255),
  `year` int,
  `season` varchar(255),
  `himal` varchar(255),
  `region` varchar(255),
  `nation` varchar(255),
  `Sum_totmembers` int,
  `Sum_smtmembers` int,
  `Sum_mdeaths` int,
  `Avg_Success_Rate` float,
  `Avg_Death_Rate` float,
  `Avg_smtdays` float,
  `Avg_totdays` float,
  `Avg_heightm` float,
  `Count` int,
  `Max_mhighpt` varchar(255)
);
```

```sql
CREATE TABLE `Dim_Members` (
  `id` varchar(255) PRIMARY KEY,
  `expid` varchar(255),
  `membid` varchar(255),
  `fname` varchar(255),
  `lname` varchar(255),
  `sex` varchar(255)
  `yob` int,
  `citizen` varchar(255),
  `residence` varchar(255),
  `occupation` varchar(255),
  `Member_Role` varchar(255)
);

ALTER TABLE `Dim_Expeditions` ADD FOREIGN KEY (`peakid`) REFERENCES `Dim_Peaks` (`peakid`);

ALTER TABLE `Fact_Expeditions` ADD FOREIGN KEY (`expid`) REFERENCES `Dim_Expeditions` (`expid`);

ALTER TABLE `Dim_Members` ADD FOREIGN KEY (`expid`) REFERENCES `Dim_Expeditions` (`expid`);
```

- Importing the values in the created Schema with the following command:

```
COPY INTO Dim_Peaks
FROM @my_stage/Dim_Peaks.csv
FILE_FORMAT = (FORMAT_NAME = CSV_FORMAT)
ON_ERROR = 'CONTINUE';

COPY INTO Dim_Expeditions
FROM @my_stage/Dim_Expeditions.csv
FILE_FORMAT = (FORMAT_NAME = CSV_FORMAT)
ON_ERROR = 'CONTINUE';

COPY INTO Dim_Members
FROM @my_stage/Dim_Members.csv
FILE_FORMAT = (FORMAT_NAME = CSV_FORMAT)
ON_ERROR = 'CONTINUE';
```

- **Data Model**: Defined Fact_Expeditions, Dim_Peaks, Dim_Expeditions, and Dim_Members tables. Added basic foreign key constraints (e.g., peakid to Dim_Peaks). Created a Himalayan_Model view joining all tables for analysis.

```
mohitpanchasara#COMPUTE_WH@HIMALAYAN_DB.PUBLIC>LIST @my_stage;
+----------------------------------+---------+----------------------------------+-------------------------------+
| name                             |    size | md5                              | last_modified                 |
|----------------------------------+---------+----------------------------------+-------------------------------|
| my_stage/Dim_Expeditions.csv.gz  |  591904 | e36e722bfbbbc1441812df1e1fe0d614 | Tue, 24 Jun 2025 07:56:25 GMT |
| my_stage/Dim_Members.csv.gz      | 2405648 | 69fd889928f9ce2734ccc25abd04272f | Tue, 24 Jun 2025 07:57:29 GMT |
| my_stage/Dim_Peaks.csv.gz        |  212752 | d641e06a7bcd4c1870db404605c518a1 | Tue, 24 Jun 2025 07:58:03 GMT |
| my_stage/Fact_Expeditions.csv.gz |  205440 | ba356e44c8e3400e60a7f3314d0b2189 | Mon, 23 Jun 2025 09:37:43 GMT |
+----------------------------------+---------+----------------------------------+-------------------------------+
4 Row(s) produced. Time Elapsed: 0.149s
mohitpanchasara#COMPUTE_WH@HIMALAYAN_DB.PUBLIC>
```

- **Queries**: Ran five simple queries to validate the model

Challenges and Solutions

- **Partial Loading**: COPY INTO failed due to column mismatches (e.g., Max_mhighpt with 'True'/'False'). Adjusted table definitions to VARCHAR and used ON_ERROR = 'CONTINUE' to load partial data.
- **File Format Issues**: Resolved SKIP_HEADER and PARSE_HEADER conflicts by recreating CSV_FORMAT with PARSE_HEADER = TRUE.
- **Time Constraint**: Simplified the process by focusing on basic queries instead of full data fixes.

Queries

Query 1: Average Height by Region

```
SELECT p.region, AVG(f.Avg_heightm) AS Avg_Height

FROM Fact_Expeditions f

JOIN Dim_Peaks p ON f.peakid = p.peakid
```

GROUP BY p.region

LIMIT 10;

```
+-------------------------+----------------+
| REGION                  |     AVG_HEIGHT |
|-------------------------+----------------|
| Annapurna-Damodar-Peri  | 7587.886522492 |
| Khumbu-Rolwaling-Makalu | 8339.586222744 |
| Manaslu-Ganesh          | 8161.359475193 |
| Kanjiroba-Far West      | 6764.339985745 |
| Langtang-Jugal          | 7012.413979545 |
| Dhaulagiri-Mukut        | 8067.967476289 |
| Kangchenjunga-Janak     | 8496.822580239 |
+-------------------------+----------------+
7 Row(s) produced. Time Elapsed: 1.622s
```

Query 2: Deaths by Season

SELECT e.season, SUM(f.Sum_mdeaths) AS Total_Deaths

FROM Fact_Expeditions f

JOIN Dim_Expeditions e ON f.expid = e.expid

GROUP BY e.season

LIMIT 10;

```
mohitpanchasara#COMPUTE_WH@HIMALAYAN_DB.PUBLIC>SELECT e.season, SUM(f.Sum_mdeaths) AS Total_Deaths
                                               FROM Fact_Expeditions f
                                               JOIN Dim_Expeditions e ON f.expid = e.expid
                                               GROUP BY e.season
                                               LIMIT 10;
+--------+--------------+
| SEASON | TOTAL_DEATHS |
|--------+--------------|
| Autumn |        49024 |
| Spring |       130138 |
| Winter |         2333 |
| Summer |          418 |
+--------+--------------+
4 Row(s) produced. Time Elapsed: 0.458s
```

Query 3: Average Success Rate by Region

SELECT p.region, AVG(f.Avg_Success_Rate) AS Avg_Success_Rate

FROM Fact_Expeditions f

JOIN Dim_Peaks p ON f.peakid = p.peakid

GROUP BY p.region

LIMIT 10;

REGION	AVG_SUCCESS_RATE
Annapurna-Damodar-Peri	26.628755477
Khumbu-Rolwaling-Makalu	41.55638325
Manaslu-Ganesh	34.427641356
Kanjiroba-Far West	17.861623053
Langtang-Jugal	20.888085467
Dhaulagiri-Mukut	23.763668814
Kangchenjunga-Janak	34.697336508

Query 4: Total Members by Nation

```
SELECT e.nation, SUM(f.Sum_totmembers) AS Total_Members

FROM Fact_Expeditions f

JOIN Dim_Expeditions e ON f.expid = e.expid

GROUP BY e.nation

LIMIT 10;
```

NATION	TOTAL_MEMBERS
USA	3454851
Austria	291347
Canada	246912
Japan	1390008
USSR	102022
UK	1856293
France	678838
Germany	521701
Spain	371479
Nepal	1368740

10 Row(s) produced. Time Elapsed: 0.349s

Query 5: Average Expedition Days by Year

```
SELECT e.year, AVG(f.Avg_totdays) AS Avg_Expedition_Days

FROM Fact_Expeditions f

JOIN Dim_Expeditions e ON f.expid = e.expid

GROUP BY e.year

LIMIT 10;
```

YEAR	AVG_EXPEDITION_DAYS
1982	17.752587992
1996	25.57980226
2009	24.850820298
1970	23.051660517
1991	22.115812918
2011	22.434315287
2007	28.150807137
2013	22.904859126
2006	29.066997519
1995	14.940397351

10 Row(s) produced. Time Elapsed: 0.487s

Step 4: Data Analysis & Queries

Here, the dataset is shifted to VM again to make the queries in **PySpark**. The .py files of all the scripts are attached offline with the submission. The following are the results obtained by running those pyspark queries using the spark-submit test.py commands.

Query 1: Predict Success Rate Based on Peak Height

This query provides a statistical summary (minimum, maximum, and average) of expedition durations (Avg_totdays) from Fact_Expeditions.csv, grouped by region (region) from Dim_Peaks.csv. It uses PySpark's aggregation functions to calculate these metrics, offering insights into how expedition lengths vary geographically. The result is displayed in a tabular form in the terminal.

Result:

region	min_days	max_days	avg_days
Annapurna-Damodar...	0	104	15.726983850607235
Dhaulagiri-Mukut	0	92	22.358349488274825
Kangchenjunga-Janak	0	133	28.77704362700296
Kanjiroba-Far West	0	52	11.898886639676114
Khumbu-Rolwaling-...	0	280	26.07292342514692
Langtang-Jugal	0	54	12.47329164223829
Manaslu-Ganesh	0	82	18.78186999944408

Query 2: Correlation Analysis of Peak Height and Success Rate

This query performs a correlation analysis to measure the strength and direction of the relationship between peak height (heightm from Dim_Peaks) and expedition success rate (Avg_Success_Rate from Fact_Expeditions). Using PySpark's built-in corr function, it calculates the Pearson correlation coefficient, providing insight into how height impacts success rates. The result is displayed in a tabular form in the terminal.

Result:

Metric	Value
Correlation Coeff...	-0.06742148409554305

Query 3: Clustering the group expeditions based on success rate and team size

This query groups expeditions into bins based on success rate (Avg_Success_Rate) and team size (Sum_totmembers) from Fact_Expeditions.csv.

The ntile(3) function in PySpark splits your data into 3 equal parts (or as close as possible) based on the values in a column. Think of it like dividing a list of numbers into three groups: low, medium, and high.

For Success Rate (Avg_Success_Rate): It looks at all the success rate values, sorts them, and assigns:

- **success_bin = 1** to the lowest third (e.g., 0.0 to 0.33).
- **success_bin = 2** to the middle third (e.g., 0.34 to 0.66).
- **success_bin = 3** to the highest third (e.g., 0.67 to 1.0).

For Team Size (Sum_totmembers): It does the same for team sizes, sorting them and assigning:

- **team_bin = 1** to the smallest third (e.g., 1-10 members).
- **team_bin = 2** to the middle third (e.g., 11-20 members).
- **team_bin = 3** to the largest third (e.g., 21+ members).

Bin Combinations: The output shows how many expeditions fall into each combination of success rate and team size bins (e.g., success_bin = 1 and team_bin = 1 means low success with small teams).

Pattern Insight:

- A high expedition_count in success_bin = 3 and team_bin = 1 might mean that small teams often succeed.
- A high count in success_bin = 1 and team_bin = 3 might suggest that large teams struggle more.

Result:

success_bin	team_bin	expedition_count
1	1	1550
1	2	1327
1	3	924
2	1	830
2	2	1293
2	3	1678
3	1	1421
3	2	1181
3	3	1198

Query 4: Frequency Analysis of Expedition Seasons

This query performs a frequency analysis to count the occurrences of each season (season) from Dim_Expeditions.csv, providing insight into the distribution of expeditions across different seasons (e.g., Spring, Autumn). Using PySpark's groupBy and count, it aggregates the data and displays the results in a tabular form in the terminal.

Result:

season	count
Spring	43881
Autumn	42003
Winter	2261
Summer	766

Query 5: Ranking Analysis of Nations by Total Summiteers.

This query performs a ranking analysis to rank nations by the total number of summiteers (Sum_smtmembers) from Fact_Expeditions.csv and Dim_Expeditions.csv. Using PySpark's Window function with rank, it assigns a rank to each nation based on the sum of summiteers, with ties receiving the same rank. The result is displayed in a tabular form in the terminal, showing the nation and its rank.

Result:

nation	total_summiteers	rank
USA	1714155	1
China	1445255	2
Nepal	815916	3
India	744610	4
UK	691045	5
Russia	504673	6
New Zealand	500178	7
Switzerland	409171	8
Japan	356259	9
France	222395	10
Ukraine	202944	11
Germany	182292	12
Austria	142564	13
Italy	115976	14
W Germany	96323	15
Kyrgyz Republic	93627	16
Netherlands	93011	17
Canada	89611	18
Spain	81937	19
Bahrain	74881	20

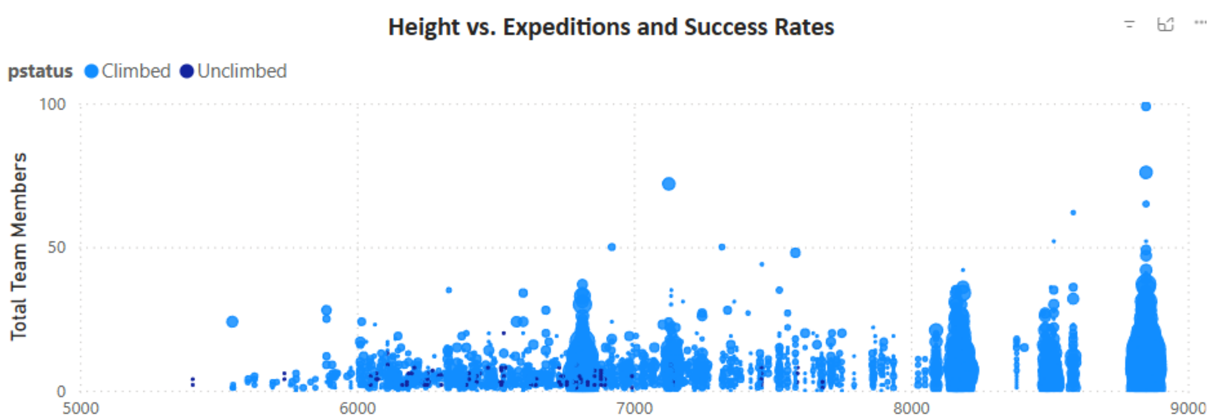
Step 5: Visualization & Reporting in Power BI

Finally, we can address those business queries that we formed at the beginning of this report, and address them with the help of creative visualizations.

Question 1. How does the height of a peak correlate with the number of expeditions and their success rates?

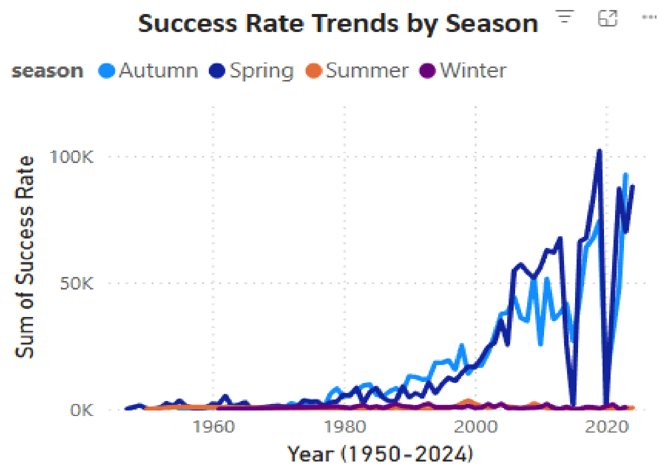
Here we can see the Power BI visualization shows the scatter plot of the graph, which shows the distribution of total team members across different heights of the peaks. For example, the Mt. Everest holds the most data with dynamic Team Size per expedition, and which of those teams have climbed or Unclimbed.

A **bigger circle** represents a **bigger success rate**, as this field is put into the size field. Here is the graph



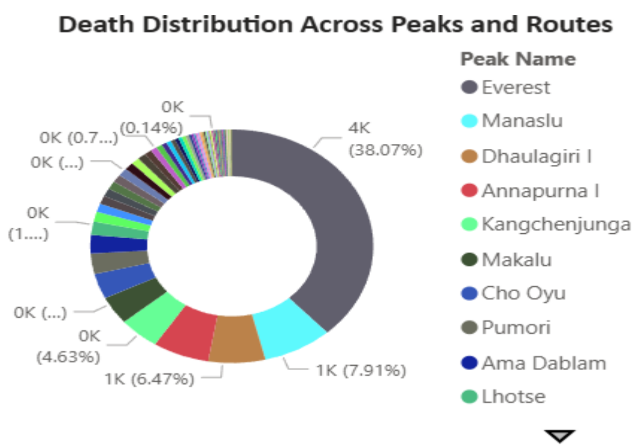
Question 2. What are the trends in expedition success rates over time, and how do they vary by season?

In this visualization, we can see the line graph, classified through the season column, where each color represents a different season and their successful expedition across different timelines. The timeline runs from 1950 to 2024, with a trend seen that climbers have increased a lot in autumn and spring, especially after the 2000s. This shows the popularity of climbers preferring Spring and Autumn for expeditions.



Question 3. Which of the peaks are considered most dangerous for the trek?

Now, what can be the most dangerous trek? Logically, the one that has the most deaths. This is represented with the help of a donut chart, which shows the Death distribution across different peaks. This reveals Mt. Everest has the most number of deaths, maybe due to its popularity, followed by Manaslu, Dhaulagiri, and Annapurna peaks. Making Everest the most dangerous trek of all time.



Question 4. Which countries have the most expeditions, or from which continent do most people come to the Himalayas?

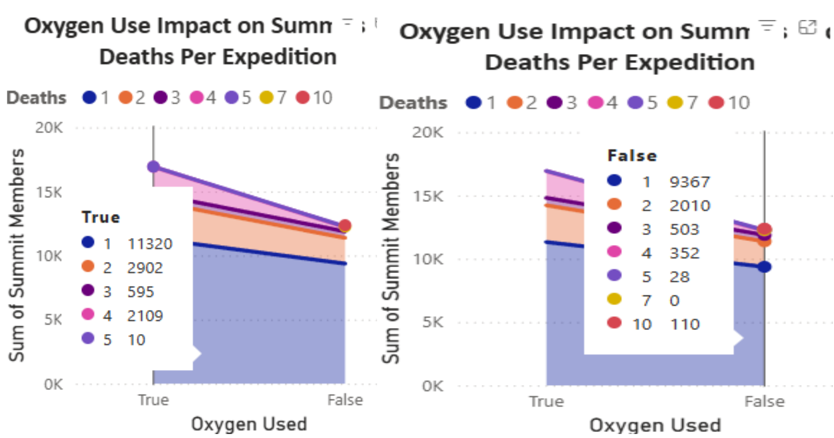
In this representation, the total number of expeditions is mapped with the Nations, which shows the attraction of the Himalayas from all over the world. From the graph, it seems most of the people come from Europe, with a great popularity to conquer greater peaks in the Himalayas and a passion towards mountain climbing.

Question 5. How does the use of supplemental oxygen affect success rates and safety?

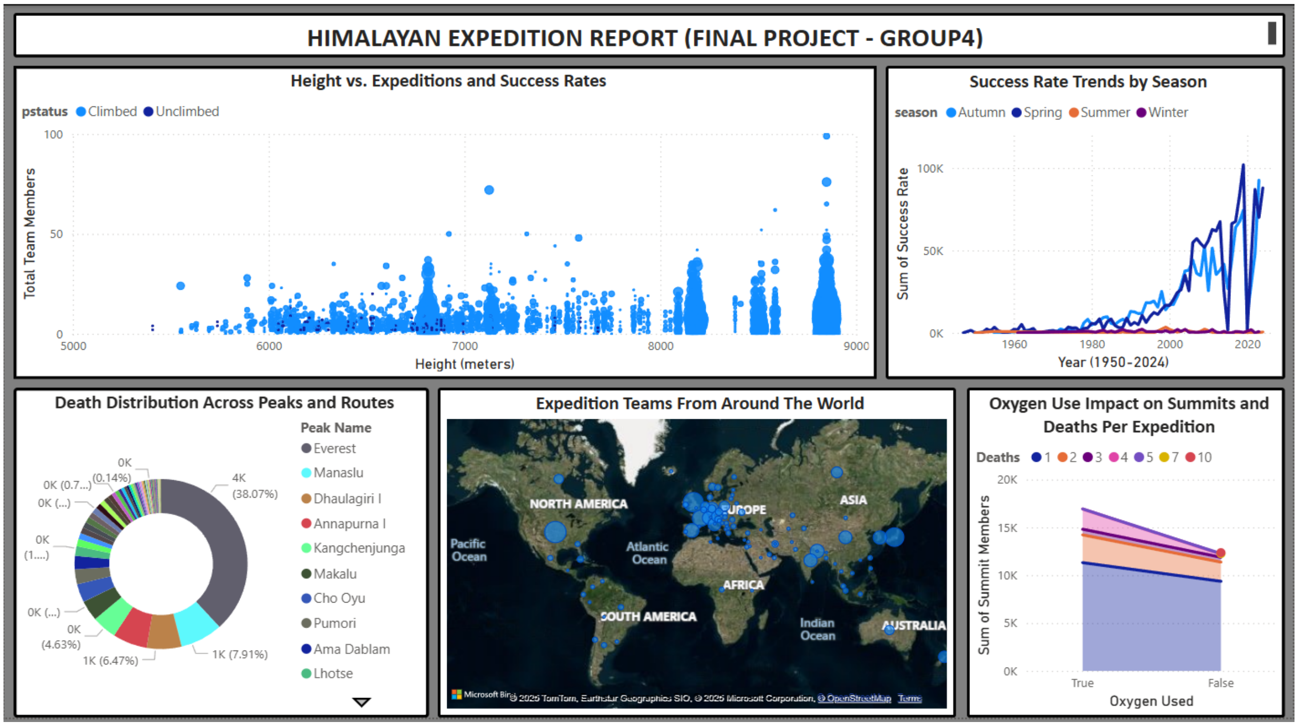
In this graph, we have done a stacked area chart, which represents a very interesting pattern. The x-axis shows whether the oxygen is used or not. Because oxygen is preferred by most people, and they certainly need it while conquering very high peaks. So this graph tells us there are more people who have used supplemental oxygen during an expedition.

Then comes the Deaths, which represents the deaths per expedition; the lower the number lower the fewer deaths in a single team. A higher number (10 over here) shows a bigger loss to a team. The overall scenario where the team member loses someone is 5, which is the highest most occurring death toll in all of the expeditions, resulting in low success rates and safety.

See the graph below and the difference between oxygen used and not used, along with the deaths at each time. Bigger teams can have bigger losses in challenging conditions.



Reporting in Power BI



7. Conclusion

This project successfully demonstrated the complete lifecycle of a big data analytics pipeline from data exploration to advanced visualizations—using real-world Himalayan expedition data.

By leveraging tools such as Alteryx for ETL, Snowflake and SSAS for data modeling, PySpark for large-scale analysis, and Power BI for interactive dashboards, we were able to extract meaningful insights from a complex dataset of over 88,000 individual records.

Key takeaways from the project include:

- Clear correlations between peak height, expedition success, and fatality rates
- The impact of seasonal trends and oxygen usage on safety and outcomes
- Identification of high-risk peaks and top contributing nations in Himalayan expeditions

Despite challenges such as inconsistent data types, missing values, and integration errors, the team overcame them through effective preprocessing and validation strategies.

This project not only answered important business questions but also showcased the power of modern big data tools to generate actionable insights. It reflects the practical application of academic concepts and tools to solve real-world data problems.