Quantifying-the-Impact-of-Layoffs-A-Comprehensive-SQL-Analysis

Exploratory Data Analysis (EDA) on Global Layoffs Data

Project Overview

This documentation provides an outline of an Exploratory Data Analysis (EDA) conducted on the world_layoffs dataset. The main objective of this analysis is to explore, analyze, and extract insights from the data on global layoffs. We identify key trends, patterns, outliers, and points of interest, such as companies with significant layoffs, industries, and countries most affected by layoffs.

Dataset: world_layoffs

The dataset contains information about companies worldwide that underwent layoffs. The key columns of interest include:

- company: Name of the company.
- total_laid_off: Total number of employees laid off.
- percentage_laid_off: Percentage of the company laid off.
- funds_raised_millions: Total funds raised by the company.
- location: Location of the layoffs (e.g., city).
- country: Country where layoffs occurred.
- date: Date of the layoffs.
- industry: Industry classification of the company.
- stage: Stage of the company (e.g., startup, growth, mature).

SQL Code Walkthrough and Insights

1. Data Overview

To begin the EDA, we first load the data to understand the structure and content:

sql

Copy code

```
SELECT *
FROM world_layoffs.layoffs_staging2;
```

- This query retrieves all records from the dataset for an initial overview of the columns, data types, and any missing values.

2. Maximum Layoffs

We identify the maximum number of layoffs that occurred in a single event:

sql

```
SELECT MAX(total_laid_off)
FROM world_layoffs.layoffs_staging2;
```

- **Insight**: This provides the maximum number of employees laid off in one instance.

3. Layoff Percentages

We explore the highest and lowest layoff percentages to determine how severely companies were impacted:

sql

```
SELECT MAX(percentage_laid_off), MIN(percentage_laid_off)
FROM world_layoffs.layoffs_staging2
WHERE percentage_laid_off IS NOT NULL;
```

- **Insight**: A percentage_laid_off of 1 means 100% of the workforce was laid off, indicating companies that completely shut down.

4. Companies with 100% Layoffs

We look for companies that laid off 100% of their workforce, which might indicate they went out of business:

sql

Copy code

```
SELECT *
FROM world_layoffs.layoffs_staging2
WHERE percentage_laid_off = 1;
```

- **Insight**: Many of these companies are startups that went out of business.

5. Companies with the Most Funding but 100% Layoffs

To assess how large these companies were, we order the 100% layoffs by the amount of funding they raised:

sql

Copy code

```
SELECT *
FROM world_layoffs.layoffs_staging2
WHERE percentage_laid_off = 1
ORDER BY funds_raised_millions DESC;
```

- **Insight**: Companies like **BritishVolt** and **Quibi** raised billions yet completely failed.

6. Largest Single Layoff Events

We identify companies with the largest layoffs in a single event:

sql

Copy code

```
SELECT company, total_laid_off
```

```
FROM world_layoffs.layoffs_staging2

ORDER BY total_laid_off DESC

LIMIT 5;
```

- **Insight**: This lists the top 5 companies with the highest layoffs on a single day.

7. Companies with the Most Total Layoffs

Summing up layoffs across multiple events for each company, we find the companies with the highest total layoffs:

```
sql

Copy code

SELECT company, SUM(total_laid_off)

FROM world_layoffs.layoffs_staging2

GROUP BY company

ORDER BY SUM(total_laid_off) DESC

LIMIT 10;
```

- **Insight**: This gives us the top 10 companies with the most cumulative layoffs.

8. Layoffs by Location

We sum layoffs by location (e.g., cities) to find areas that were most affected:

```
sql

SELECT location, SUM(total_laid_off)

FROM world_layoffs.layoffs_staging2

GROUP BY location

ORDER BY SUM(total_laid_off) DESC

LIMIT 10;
```

- **Insight**: Identifies the top 10 cities affected by layoffs.

9. Layoffs by Country

Similarly, we sum layoffs by country to identify countries most impacted:

sql

```
SELECT country, SUM(total_laid_off)
FROM world_layoffs.layoffs_staging2
GROUP BY country
ORDER BY SUM(total_laid_off) DESC;
```

- **Insight**: Shows the total number of layoffs per country.

10. Layoffs by Year

Breaking down the layoffs year-by-year gives us an understanding of how layoffs evolved over time:

sql

```
SELECT YEAR(date), SUM(total_laid_off)
FROM world_layoffs.layoffs_staging2
GROUP BY YEAR(date)
ORDER BY YEAR(date) ASC;
```

- **Insight**: This highlights any trends in layoffs over different years.

11. Layoffs by Industry

We group layoffs by industry to see which sectors were hardest hit:

sql

```
SELECT industry, SUM(total_laid_off)
FROM world_layoffs.layoffs_staging2
GROUP BY industry
ORDER BY SUM(total_laid_off) DESC;
```

- **Insight**: Industries like technology, finance, and others can be compared.

12. Layoffs by Company Stage

Analyzing layoffs by the stage of the company (e.g., startup, growth, mature) provides insight into how businesses of different sizes or maturity levels were affected:

sql

Copy code

```
SELECT stage, SUM(total_laid_off)
FROM world_layoffs.layoffs_staging2
GROUP BY stage
ORDER BY SUM(total_laid_off) DESC;
```

- **Insight**: Shows how startups, growth-stage, and mature companies were impacted.

Advanced Queries

13. Layoffs by Company and Year

This query identifies the companies with the most layoffs each year:

sql

Copy code

```
WITH Company_Year AS
(
  SELECT company, YEAR(date) AS years, SUM(total_laid_off) AS total_laid_off
  FROM world_layoffs.layoffs_staging2
  GROUP BY company, YEAR(date)
),
Company_Year_Rank AS (
```

```
  SELECT company, years, total_laid_off, DENSE_RANK() OVER (PARTITION BY years ORDER BY total_laid_off DESC) AS ranking
  FROM Company_Year
)
SELECT company, years, total_laid_off, ranking
FROM Company_Year_Rank
WHERE ranking <= 3
ORDER BY years ASC, total_laid_off DESC;
```

- **Insight**: Lists the top 3 companies with the highest layoffs per year.

14. Rolling Total Layoffs by Month

This query calculates the total number of layoffs each month and computes a rolling total over time:

sql

Copy code

```
WITH DATE_CTE AS
(
  SELECT SUBSTRING(date, 1, 7) AS dates, SUM(total_laid_off) AS total_laid_off
  FROM world_layoffs.layoffs_staging2
  GROUP BY dates
  ORDER BY dates ASC
)
SELECT dates, SUM(total_laid_off) OVER (ORDER BY dates ASC) AS rolling_total_layoffs
FROM DATE_CTE
ORDER BY dates ASC;
```

- **Insight**: Provides a rolling sum of layoffs over time, which helps track the cumulative effect of layoffs.

Conclusion

This EDA has revealed important insights about global layoffs, such as companies with the most layoffs, industries and countries most affected, and trends over time. The analysis demonstrates the extent of the impact, particularly on startups and companies with significant funding. Moreover, advanced queries provided deeper insights into layoffs on a per-year basis and tracked rolling totals.