PART 2: Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is commonly used within the data community as a foundational step to explore and understand the data before making assumptions or developing models. In the context of data analytics, EDA can play a crucial role in extracting general business-related insights from the data.

To begin, we'll focus on identifying general information and uncovering patterns that can serve as a basis for further analysis and decision-making.

- What is the overall structure of the dataset, and what types of information does it contain

```
SELECT * FROM layoff_cl2;
```

--This version of the dataset contains 3,849 records and 11 data columns.

company text	location text	industry text	layoff_num integer	layoff_perc numeric	date date	stage text	country text	funds_raised numeric	year integer	month text
Advata	Seattle	Healthcare	32	0.21	2022-10-28	Unknown	United States	[null]	2022	October
Gatherly	Atlanta	Marketing	[null]	0.5	2022-07-31	Unknown	United States	[null]	2022	July
Soundwide	Berlin	Other	[null]	0.08	2023-01-27	Unknown	Germany	[null]	2023	January
Verily	SF Bay Area	Healthcare	250	0.15	2023-01-11	Unknown	United States	3500.0	2023	January
Zapp	London	Food	[null]	0.1	2022-05-25	Unknown	United Kingdom	300.0	2022	May
Atmosphere	Austin	Other	[null]	[null]	2023-12-07	Series D	United States	214.0	2023	December
Beamery	London	HR	[null]	0.12	2023-01-10	Series D	United States	223.0	2023	January
Glovo	Madrid	Food	22	[null]	2024-05-25	Acquired	Spain	1200.0	2024	May
HP	Tel Aviv	Hardware	100	[null]	2023-02-19	Post-IPO	Israel	4200.0	2023	February
F-Secure	Helsinki	Security	70	[null]	2023-10-25	Post-IPO	Finland	[null]	2023	October

- Let’s examine some descriptive statistics of layoffs:

```
SELECT
        COUNT(company) AS layoff_count,
        MAX(layoff_num) AS max_layoff_num,
        MIN(layoff_num) AS min_layoff_num,
        ROUND(AVG(COALESCE(layoff_num, 0)), 2) AS Average_layoff_num
FROM layoff_cl2;
```

--The maximum number of layoffs recorded was **15,000**, the minimum was **3**, and the average layoff number was **177.46**.

layoff_count bigint	max_layoff_num integer	min_layoff_num integer	average_layoff_num numeric
3849	15000	3	177.46

- Which companies have laid off more than 50% of their employees?

```
SELECT
        company,
        ROUND(layoff_perc, 2) AS layoff_percentage
FROM layoff_cl2
        WHERE layoff_perc > 0.50
        ORDER BY layoff_perc ASC;
```

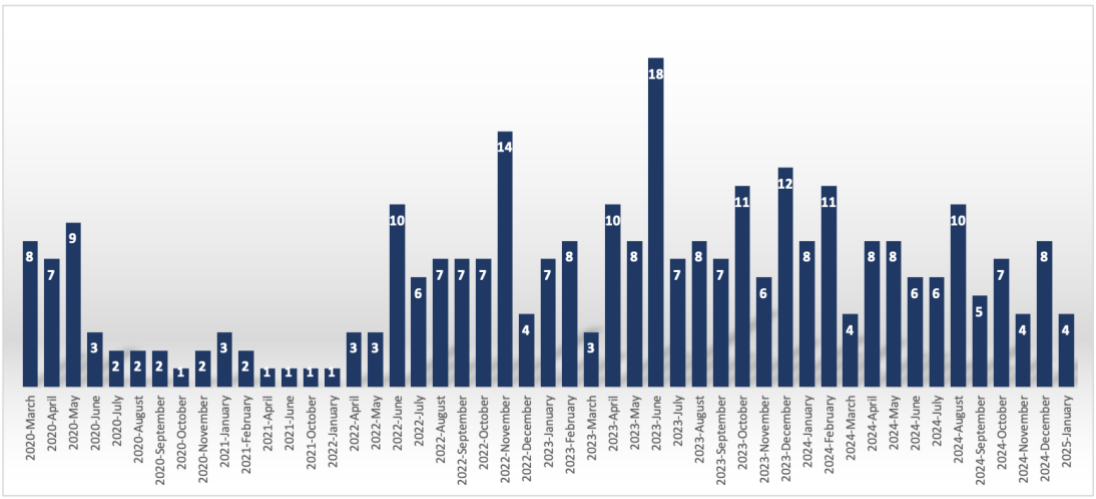
--A total of **379 companies (9.8%)** have laid off more than half of their employees since March 2020.

- Which companies have laid off 100% of their employees (shut down)?

```
SELECT
        company,
        ROUND(layoff_perc, 2) AS layoff_percentage
FROM layoff_cl2
        WHERE layoff_perc= 1
        ORDER BY company;
```

--A total of **300 companies (7.8%)** have laid off all of their employees since 2020, indicating that most of these companies have shut down. Let see the date pattern of companies shutdown since March 2020:

```
SELECT
        CONCAT(year,'-', month_name) as year_month,
        COUNT(company) companies_shutdown FROM
(SELECT
        company,
        layoff_perc,
        EXTRACT(YEAR FROM date) AS year,
        EXTRACT(MONTH FROM date) AS month_num,
        TO_CHAR(date, 'FMMonth') AS month_name
FROM layoff_cl2
        WHERE layoff_perc = 1) date_layoff
GROUP BY year, month_name, month_num
ORDER BY year ASC, month_num ASC
```



- What are the top 5 largest layoff events since 2020, and which companies were involved?

```
SELECT
        company,
        layoff_num
FROM layoff_cl2
        WHERE layoff_num IS NOT NULL
        ORDER BY layoff_num DESC
        LIMIT 5;
```

--The top 5 layoffs involve well-known giants: **Intel, Tesla, Google, and Meta** and **Amazon.** Each of these companies has unfortunately laid off more than 10,000 employees in the found layoff round.

company text	layoff_num integer
Intel	15000
Tesla	14000
Google	12000
Meta	11000
Amazon	10000

- In general, which companies have laid off the most employees in total (not just a single event) since 2020?

```
SELECT
        company,
        COALESCE(SUM(layoff_num), 0) AS total_laid_off
FROM layoff_cl2
```

```
        GROUP BY company
        ORDER BY total_laid_off DESC
        LIMIT 5;
```

--This series of giants — **Amazon, Meta, Intel, Microsoft, and Tesla** — have laid off approximately **100,000 employees** in total over the last 4 years

company text	total_laid_off bigint
Amazon	27840
Meta	24600
Intel	16057
Microsoft	14708
Tesla	14500

- **Which countries have the highest total number of employees laid off?**

```
SELECT
        country,
        SUM(layoff_num) AS total_laid_off
FROM layoff_cl2
        WHERE layoff_num IS NOT NULL
        GROUP BY country
        ORDER BY total_laid_off DESC
        LIMIT 5;
```

country text	total_laid_off bigint
United States	460166
India	56631
Germany	30412
United Kingdom	20090
Netherlands	19005

```
SELECT
        country,
        companies_num,
        ROUND(total_laid_off/companies_num, 1) AS per_company_layoff
```

```
FROM (SELECT
          country,
          COUNT(DISTINCT(company)) AS companies_num,
          COALESCE(SUM(layoff_num), 0) AS total_laid_off
      FROM layoff_cl2
          GROUP BY country) AS companies_layoff
      ORDER BY per_company_layoff DESC
      LIMIT 10;
```

--Based on the total number of employees laid off, the **US** ranks first, with nearly **8 times more jobs cut** than **India**, which holds the second rank. However, to make a more statistically sound comparison, it's also useful to consider the number of companies involved (per company values). In this context, **Netherlands** ranks first, with **18 companies**, followed by the **Japan** with **5 companies**.

country text	companies_num bigint	per_company_layoff numeric
Netherlands	18	1055.0
Japan	5	833.0
Sweden	22	693.0
Cayman Islands	2	530.0
Russia	1	400.0
China	22	372.0
Germany	84	362.0
Saudi Arabia	1	340.0
Senegal	1	300.0
India	204	277.0

To examine the situation in the **Netherlands**, we can run the following query:

```
SELECT
        company,
        COALESCE(SUM(layoff_num), 0) AS total_laid_off
FROM layoff_cl2
        WHERE country = 'Netherlands'
        GROUP BY company
        ORDER BY total_laid_off DESC;
```

--It appears that **Philips, Booking.com, and OLX Group** are the key players contributing to the layoffs in the Netherlands.

company text	total_laid_off bigint
Philips	10000
Booking.com	4375
OLX Group	2300
TomTom	500
Just Eat Takeaway	390
Just Eat	300
Bol.com	300
WeTransfer	295
Uber	225
PayU	150
MessageBird	90
Sketch	80
Reynen Court	0
TicketSwap	0
Mollie	0
BUX	0
Change Invest	0
Prosus	0

- **Which industries have have experienced the highest total layoffs?**

```
SELECT
        industry,
        COALESCE(SUM(layoff_num), 0) AS total_laid_off,
        COUNT(company) AS companies_num
FROM layoff_cl2
        GROUP BY industry
        ORDER BY total_laid_off DESC
        LIMIT 5;
```

industry text	total_laid_off bigint	companies_num bigint
Consumer	74646	222
Retail	72819	312
Other	62167	238
Transportation	60568	240
Hardware	54870	52

--The **Consumer** and **Retail** industries have seen the highest impact from layoffs since 2020. Let's now examine the top 3 companies in these industries (query below):

```
WITH company_industry_cte AS (
        SELECT
                company,
                industry,
                SUM(layoff_num) AS total_laid_off
```

```
        FROM layoff_cl2
                WHERE layoff_num IS NOT NULL
                GROUP BY industry, company
                ORDER BY total_laid_off DESC
),
        rank_within_indust AS (
        SELECT *,
                dense_rank() OVER (PARTITION BY industry ORDER BY total_laid_off DESC)
AS rank_in_indust
        FROM company_industry_cte
        )
SELECT * FROM rank_within_indust
        WHERE rank_in_indust < 4 AND industry IN ('Consumer', 'Retail');
```

company text	industry text	total_laid_off bigint	rank_in_indust bigint
Meta	Consumer	24600	1
Google	Consumer	13472	2
Bytedance	Consumer	5700	3
Amazon	Retail	27840	1
Wayfair	Retail	5000	2
Groupon	Retail	3800	3

--However, when considering the layoff rate per company, the **Hardware** industry ranks first, with a rate three times higher than that of the **Consumer** industry, which ranks second

```
SELECT
        industry,
        companies_num,
        ROUND(total_laid_off/companies_num, 1) AS per_company_layoff
FROM (SELECT
          industry,
          COUNT(DISTINCT(company)) AS companies_num,
          COALESCE(SUM(layoff_num), 0) AS total_laid_off
      FROM layoff_cl2
          GROUP BY industry) AS industries_layoff
      ORDER BY per_company_layoff DESC
      LIMIT 10;
```

- **Research suggests that having a single round of layoffs is often a better solution for both companies and employees, compared to multiple rounds. Let's take a look at which companies have had more than 5 rounds of layoff in the past 4 years.**

```
SELECT company,
        COUNT(company) AS layoff_rounds,
        COALESCE(SUM(layoff_num), 0) AS total_laid_off
FROM layoff_cl2
        GROUP BY company
```

```
    HAVING COUNT(company) > 5
    ORDER BY layoff_rounds DESC, total_laid_off DESC;
```

--The following companies have had more than 5 rounds of layoffs among them **Amazon** (12 rounds), **Google** (12 rounds), **Microsoft (**11 rounds), **Rivian** (8 rounds), **Salesforce**, **Spotify**, **Bytedance** (7 rounds each) rank top.

company text	layoff_rounds bigint	total_laid_off bigint
Amazon	12	27840
Google	12	13472
Microsoft	11	14708
Rivian	8	1170
Salesforce	7	11140
Bytedance	7	5700
Spotify	7	2355
Meta	6	24600
Intel	6	16057
Uber	6	7785
Better.com	6	3900

company text	layoff_rounds bigint	total_laid_off bigint
Swiggy	6	3280
Unity	6	3199
Stitch Fix	6	2688
Gopuff	6	2400
Redfin	6	1897
Expedia	6	1600
Loft	6	1289
ShareChat	6	958
Convoy	6	620
New Relic	6	545
Jellysmack	6	273

- **Which of the top five countries have companies raised the most funds per company?"**

```
WITH country_fund_cte AS (
    SELECT
        country,
        ROUND(SUM(funds_raised) / COUNT(company), 1) AS funds_per_company
    FROM layoff_cl2
        WHERE funds_raised IS NOT NULL
        GROUP BY country
),
country_rank_cte AS (
    SELECT
        *,
        DENSE_RANK() OVER (ORDER BY funds_per_company DESC) AS country_fund_rank
    FROM country_fund_cte
)
SELECT *
FROM country_rank_cte
WHERE country_fund_rank <= 5;
```

The top five countries with the highest funds raised per company are **Lithuania, China, the Netherlands, Turkey, and Singapore**. Lithuania and Turkey made the list mainly due to their small number of companies—**two in Lithuania and one in Turkey**—and the significant funds raised by **Uber in Lithuania and Getir in Turkey**.

If we ignore per-company funding, the **United States ranks first, followed by India and China**, which aligns with expectations.

To explore further, we can analyze specific countries like Lithuania and Turkey with the following query:

```
SELECT * FROM layoff_cl2
WHERE country IN ('Lithuania', 'Turkey');
```

- **A rolling total is an important metric that help us to catch a clear picture of trends in our data. Let's take a look at year_month rolling totals:**

```
WITH month_series AS (
    SELECT
        generate_series(
            (SELECT MIN(DATE_TRUNC('month', date)) FROM layoff_cl2),
            (SELECT MAX(DATE_TRUNC('month', date)) FROM layoff_cl2),
            INTERVAL '1 month'
        )::DATE AS month_start
       ), month_year_cte AS (
    SELECT
        DATE_PART('year', month_start) AS year,
       DATE_PART('month', month_start) AS month_num,
        TO_CHAR(month_start, 'FMMonth') AS month_name,
        COALESCE(SUM(layoff_num), 0) AS total_laid_off
    FROM month_series
    LEFT JOIN layoff_cl2
        ON DATE_TRUNC('month', layoff_cl2.date) = month_series.month_start
    GROUP BY month_start
)
SELECT
    year,
    month_name,
    total_laid_off,
    SUM(total_laid_off) OVER (ORDER BY year, month_num
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS rolling_total
FROM month_year_cte
ORDER BY year, month_num;
```

--There was a significant increase in tech layoffs at the beginning of 2022, a trend that continued into 2023 and 2024 as well

year double precision	month_name text	total_laid_off bigint	rolling_total numeric
2020	March	9628	9628
2020	April	26710	36338
2020	May	25804	62142
2020	June	7627	69769
2020	July	7112	76881
2020	August	1969	78850
2020	September	609	79459
2020	October	450	79909
2020	November	237	80146
2020	December	852	80998
2021	January	6813	87811
2021	February	868	88679
2021	March	47	88726
2021	April	261	88987
2021	May	0	88987
2021	June	2434	91421
2021	July	80	91501
2021	August	1867	93368
2021	September	161	93529
2021	October	22	93551

year double precision	month_name text	total_laid_off bigint	rolling_total numeric
2021	November	2070	95621
2021	December	1200	96821
2022	January	510	97331
2022	February	3685	101016
2022	March	5714	106730
2022	April	4128	110858
2022	May	12985	123843
2022	June	17394	141237
2022	July	16238	157475
2022	August	13055	170530
2022	September	5881	176411
2022	October	20471	196882
2022	November	53594	250476
2022	December	10664	261140
2023	January	89709	350849
2023	February	39902	390751
2023	March	37963	428714
2023	April	20100	448814
2023	May	15127	463941
2023	June	11206	475147

year double precision	month_name text	total_laid_off bigint	rolling_total numeric
2023	July	10690	485837
2023	August	10178	496015
2023	September	4707	500722
2023	October	8076	508798
2023	November	8373	517171
2023	December	8189	525360
2024	January	34137	559497
2024	February	15729	575226
2024	March	7403	582629
2024	April	22423	605052
2024	May	11011	616063
2024	June	10083	626146
2024	July	9051	635197
2024	August	25944	661141
2024	September	3941	665082
2024	October	3659	668741
2024	November	6455	675196
2024	December	2268	677464
2025	January	5575	683039

- **In order to take a detailed look at layoff trend, I decided to see yearly and monthly pattern of layoffs.**

```
WITH yearly_sum AS(
        SELECT
                year,
                COALESCE(SUM(layoff_num),0) as year_sum
        FROM layoff_cl2
                GROUP BY year
        ), year_percent AS(
        SELECT
                year,
                year_sum,
                (year_sum *100)/SUM(year_sum) OVER() as percent
        FROM yearly_sum)
SELECT year, year_sum, ROUND(percent, 1) as yearly_percent
        FROM year_percent
        ORDER BY year

--------

WITH monthly_sum AS (
    SELECT
        TO_CHAR(date, 'Month') AS month,
        EXTRACT(MONTH FROM date) AS month_num,
        COALESCE(SUM(layoff_num), 0) AS month_sum
    FROM layoff_cl2
    GROUP BY TO_CHAR(date, 'Month'), EXTRACT(MONTH FROM date)
), monthly_percent  AS (
        SELECT
                *,
                (month_sum * 100.0) / SUM(month_sum) OVER() AS percent
        FROM monthly_sum)
SELECT
```

```
        month,
        month_sum,
        ROUND(percent, 1) AS month_percent
FROM monthly_percent
        ORDER BY month_num

--------

SELECT
        CONCAT(year,'-',EXTRACT(QUARTER FROM date)) AS quarter,
        COALESCE(SUM(layoff_num), 0) as total_laid_off
FROM layoff_cl2
        GROUP BY year, quarter
        ORDER BY total_laid_off DESC;
```

According to the dataset, 2023—especially its first quarter—was particularly challenging for tech employees. The number of layoffs in the tech sector surged significantly, with **264,220** workers let go, marking a nearly **60% increase** from the **164,319** layoffs in 2022. Moreover, the majority of these layoffs occurred at the beginning of the year, with **January alone accounting for 20%** of the total.

quarter text	total_laid_off bigint
2023-1	167574
2022-4	84729
2020-2	60141
2024-1	57269
2023-2	46433
2024-2	43517
2024-3	38936
2022-3	35174
2022-2	34507
2023-3	25575
2023-4	24638
2024-4	12382
2022-1	9909
2020-3	9690
2020-1	9628
2021-1	7728
2025-1	5575
2021-4	3292
2021-2	2695
2021-3	2108
2020-4	1539

year integer	year_sum bigint	yearly_percent numeric
2020	80998	11.9
2021	15823	2.3
2022	164319	24.1
2023	264220	38.7
2024	152104	22.3
2025	5575	0.8

month text	month_sum bigint	month_percent numeric
January	136744	20.0
February	60184	8.8
March	60755	8.9
April	73622	10.8
May	64927	9.5
June	48744	7.1
July	43171	6.3
August	53013	7.8
September	15299	2.2
October	32678	4.8
November	70729	10.4
December	23173	3.4