STOCK MARKET ANALYSIS

SQL PROJECT

BY -
Saurabh Kumar Soni

CONTENTS

- Introduction to Case Study
- Objective
- Data Analysis
- Insights

CASE STUDY

- Data Source https://www.kaggle.com/datasets/debashis74017/stock-market-index-data-india-1990-2022?rvi=1
- This Case Study contains 3 Datasets.
- Attributes(Same in all 3 Datasets) – Date, Open, High, Close, Sector.
- Data in Dataset is from 2011-08-01 To 2022-01-31.
- 1st Dataset - Nifty IT Dataset.
- 2nd Dataset - Nifty FMCG Dataset.
- 3rd Dataset - Nifty BANK Dataset.

OBJECTIVE

- Analyse IT Industry, FMCG Industry and BANK Industry on following parameters -
  - Volatility.
  - Drawdown percentage (Time Frame – 2020-02-20 and 2020-03-23).
  - Recovery days.
  - Number of days when price closed above its previous day’s close.
  - CAGR(Returns).
  - Scoring based on the above criteria.

IT INDUSTRY

- The **IT** sector is a field which is changing the shape of Indian business standards.
- This sector includes –
  - software development, consultancies, software management.
  - online services and business process outsourcing (BPO).
- The IT industry accounted for 7.4% of India's GDP in FY22, and it is expected to contribute 10% to India's GDP by 2025.

FMCG INDUSTRY

- The Fast-moving consumer goods (FMCG) sector is the 4th largest sector of the Indian economy.
- Fast-moving consumer goods are nondurable products that sell quickly at relatively low costs.
- FMCGs have low profit margins and high-volume sales.
- Examples of FMCGs include milk, gum, fruit and vegetables etc.

BANK INDUSTRY

- Banking is an industry that deals with credit facilities, storage for cash, investments, and other financial transactions.
- A bank is an institution that accepts customer deposits and offers loans to individuals and corporate clients.
- The banking industry is one of the key drivers of most economies because it channels funds to borrowers with productive investments.

DATA ANALYSIS

- **Data analysis** is the process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.
- The process of data analysis uses to organize the data in a logical way. It helps to analyze data from different outlooks and a variety of statistical perspectives.

Query Query History

Execute/Refresh
F5

```
1  SELECT * FROM public.bank_datase
2  SELECT * FROM public.fmcg_dataset
3  SELECT * FROM public.it_dataset
4
5
6
7
8
```

Data Output Messages Notifications

	Date date	Open real	High real	Low real	Close real
1	2011-02-01	6992	6999.75	6824.75	6898.6
2	2011-02-02	6964.4	7063.3	6928.9	6961.6
3	2011-02-03	6978.75	7061.8	6928.55	6998.9
4	2011-02-04	6965.65	7001.4	6824.2	6850.6
5	2011-02-07	6916.85	6920.35	6839.8	6861.2
6	2011-02-08	6879.55	6915.45	6830.75	6847.5
7	2011-02-09	6825.15	6867.15	6759.05	6828.9
8	2011-02-10	6799.5	6811.85	6697.3	6720.4

Query Query History

```
6   ALTER TABLE public.bank_dataset
7   ADD COLUMN "sector" VARCHAR(50)
8
9   ALTER TABLE public.fmcg_dataset
10  ADD COLUMN "sector" VARCHAR(50)
11
12  ALTER TABLE public.it_dataset
13  ADD COLUMN "sector" VARCHAR(50)
14
15  UPDATE public.bank_dataset SET sector = 'BANK'
16  UPDATE public.fmcg_dataset SET sector = 'FMCG'
17  UPDATE public.bank_dataset SET sector = 'IT'
18
19
20
```

Data Output Messages Notifications

```
UPDATE 2851

Query returned successfully in 123 msec.
```

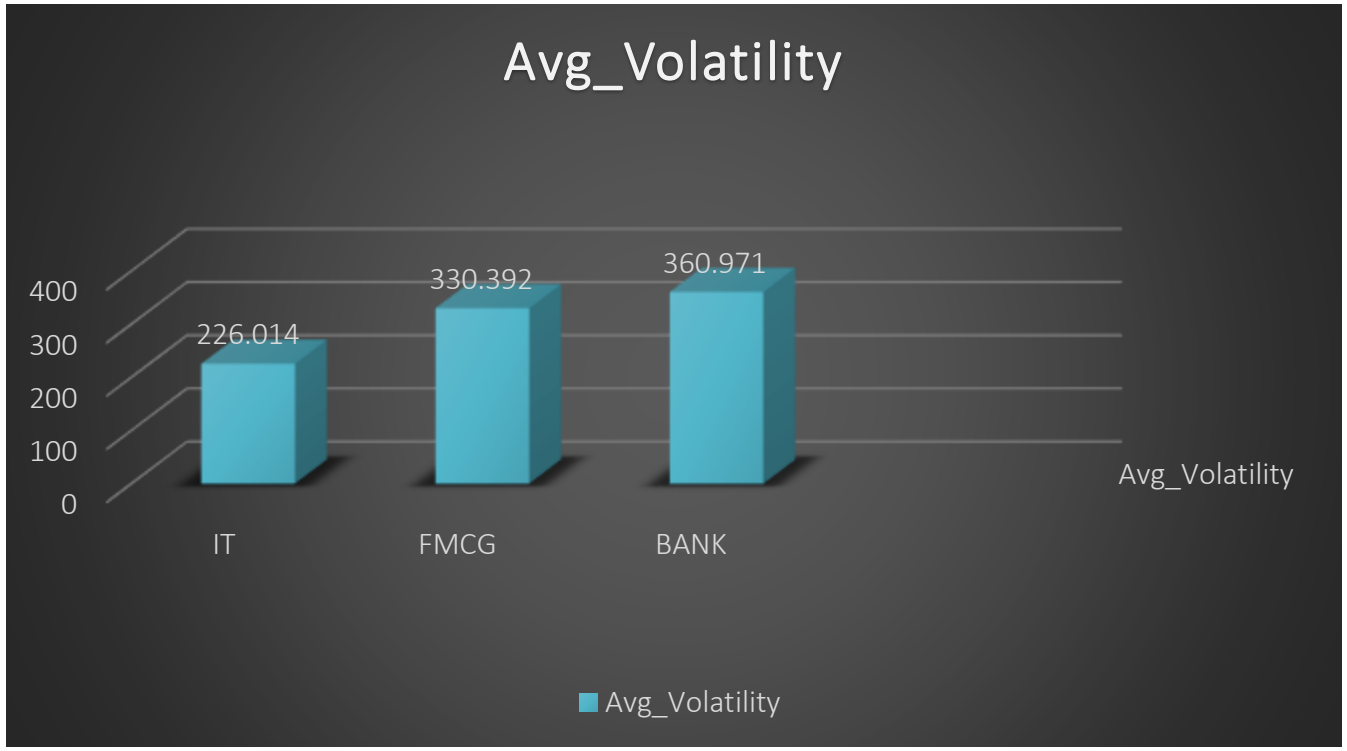
- Retrieving data from the table.
- Adding a new column sector.

VOLATILITY –

Volatility – In finance , Volatility is the rate at which the price of a stock increases or decreases over a particular period.

Higher stock price volatility often means higher risk and helps an investor to estimate the fluctuations that may happen in the future.

Formula Used – **Average(Day High – Day Low)** over a period of time.



INSIGHTS -

Volatility – Lower is the better

IT is the least volatile sector

Banking sector is the most volatile sector

```sql
CREATE VIEW A1 as (SELECT sector,high,low FROM public.bank_dataset b
UNION
SELECT sector,high,low FROM public.fmcg_dataset f
UNION
SELECT sector,high,low FROM public.it_dataset i);
```

```sql
SELECT sector,AVG(High - Low) as avg_volatility,
       dense_rank() OVER(ORDER BY avg(high - low) asc) AS ranking
FROM A1
GROUP BY sector
```

➢ Volatility

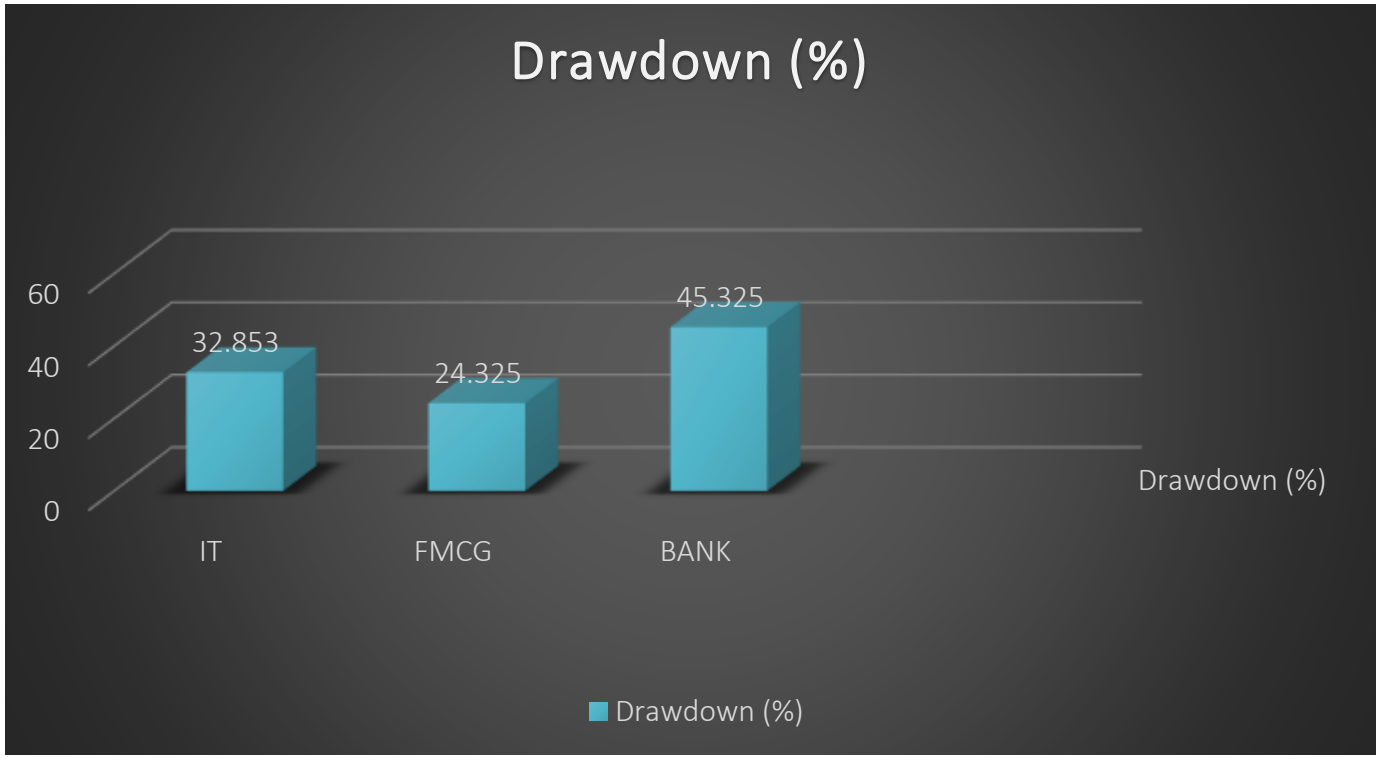
sector character varying (50)	avg_volatility double precision	ranking bigint
IT	226.01372719413143	1
FMCG	330.39177060269424	2
BANK	360.97107922658716	3

DRAWDOWN (FALL IN PRICE) – (2020-02-20 to 2020-03-23)

Drawdown – A drawdown is defined as the percentage of decline in the value over a period before it bounces back to the original value or beyond. It is expressed as the difference between the highest, i.e., the peak and the lowest.

Formula Used –

$$\text{Percent Increase} = \frac{\text{Final Value} - \text{Initial Value}}{\text{Initial Value}} \times 100\ \%$$



INSIGHTS -

- Drawdown – Lower is the better
- FMCG is the least Drawdown sector
- Banking is the most Drawdown sector

```sql
SET @pre_covid_price_it = (SELECT close FROM it_dataset WHERE DATE = '2020-02-20');
SET @post_covid_price_it = (SELECT close FROM it_dataset WHERE DATE = '2020-03-23');

SELECT ROUND(100.0*(@post_covid_price_it - @pre_covid_price_it)/@pre_covid_price_it,4) AS it_DrawDown
```

- IT Drawdown

it_DrawDown
-32.8528

```sql
SET @pre_covid_price_fmcg = (SELECT close FROM fmcg_dataset WHERE DATE = '2020-02-20');
SET @post_covid_price_fmcg = (SELECT close FROM fmcg_dataset WHERE DATE = '2020-03-23');

SELECT ROUND(100.0*(@post_covid_price_fmcg - @pre_covid_price_fmcg)/@pre_covid_price_fmcg,4) AS fmcg_DrawDown
```

- FMCG Drawdown

fmcg_DrawDown
-24.3248

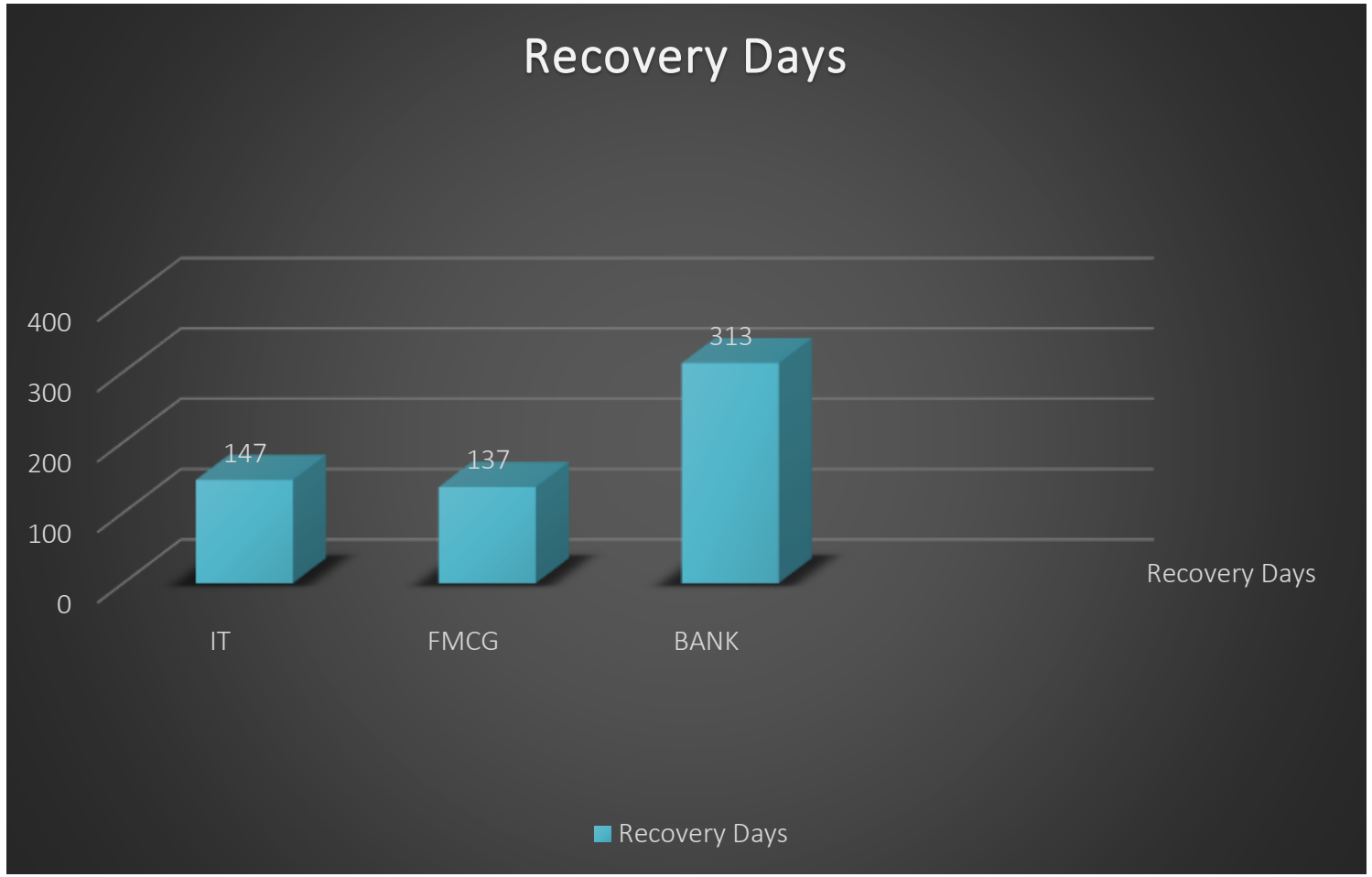
```sql
SET @pre_covid_price_bank = (SELECT close FROM bank_dataset WHERE DATE = '2020-02-20');
SET @post_covid_price_bank = (SELECT close FROM bank_dataset WHERE DATE = '2020-03-23');

SELECT ROUND(100.0*(@post_covid_price_bank - @pre_covid_price_bank)/@pre_covid_price_bank,4) AS bank_DrawDown
```

- Bank Drawdown

bank_DrawDown
-45.3253

RECOVERY DAYS–

Recovery Days – Number of days has to taken to close above it's pre covid price.



INSIGHTS -

Recovery Days – Lesser the better

FMCG – Fastest Recovery

Banking - Slowest Recovery

```sql
SET @recovery_date_it = (SELECT date FROM it_dataset WHERE close > @pre_covid_price_it AND date > '2020-02-20'
                         ORDER BY date LIMIT 1)
SELECT DATEDIFF(@recovery_date_it,'2020-02-20') AS recovery_date_it
```

➢ IT Recovery Days

recovery_date_it
147

```sql
SET @recovery_date_fmcg = (SELECT date FROM fmcg_dataset WHERE close > @pre_covid_price_fmcg AND date > '2020-02-20'
                           ORDER BY date LIMIT 1)
SELECT DATEDIFF(@recovery_date_fmcg,'2020-02-20') AS recovery_date_fmcg
```

➢ FMCG Recovery Days

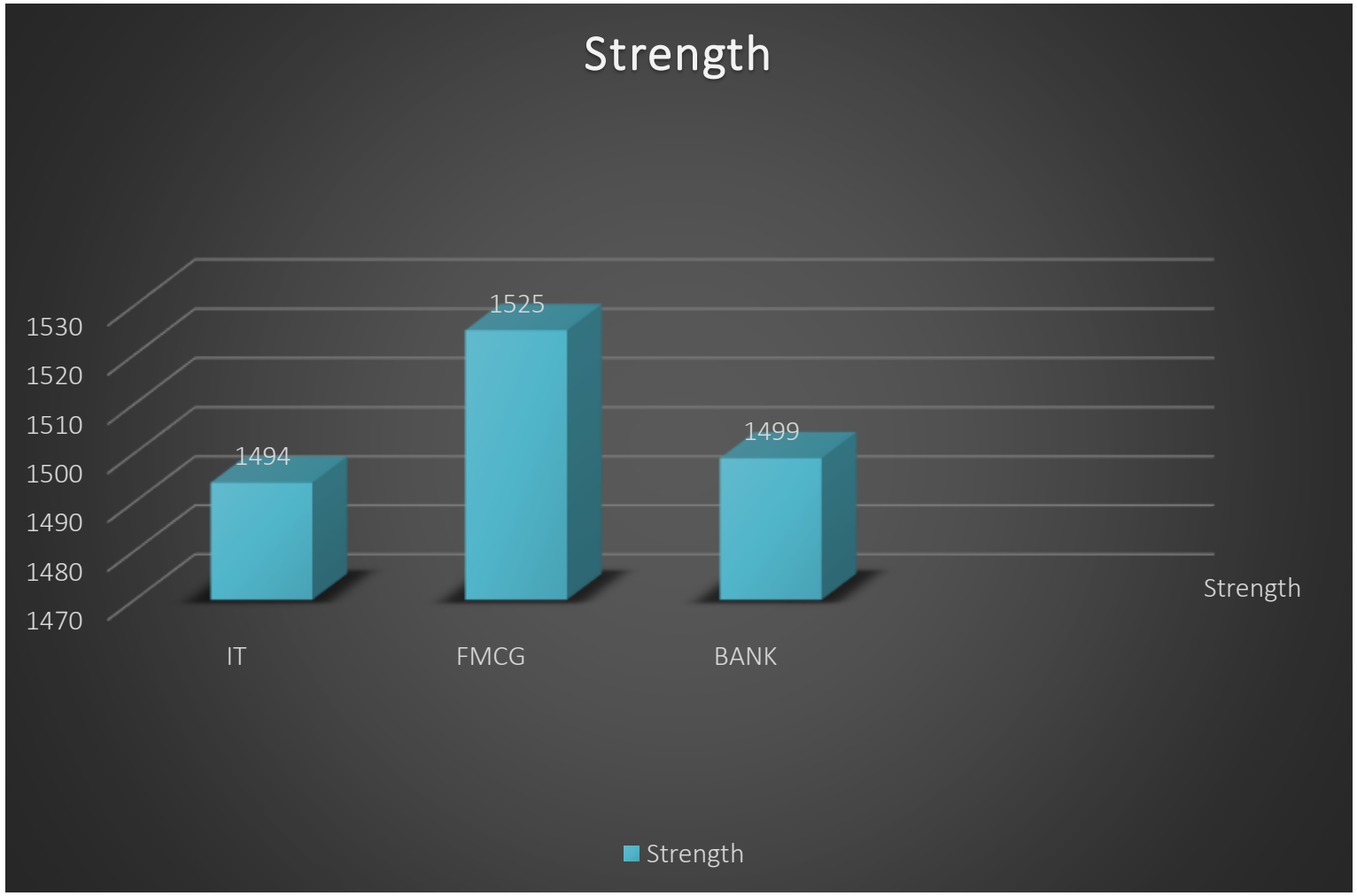
recovery_date_fmcg
137

```sql
SET @recovery_date_bank = (SELECT date FROM bank_dataset WHERE close > @pre_covid_price_bank AND date > '2020-02-20'
                           ORDER BY date LIMIT 1)
SELECT DATEDIFF(@recovery_date_bank,'2020-02-20') AS recovery_date_bank
```

➢ BANK Recovery Days

recovery_date_bank
313

NUMBER OF DAYS WHEN STOCK PRICE CLOSED ABOVE ITS PREVIOUS DAY CLOSED PRICE (STRENGTH)–



INSIGHTS -

Strength– Higher the better

IT – minimum

FMCG - maximum

```sql
SELECT sector,SUM(IF((close>prvs_close),1,0)) AS higher_closed_price_days
FROM (SELECT sector,date,close,(LAG(close) OVER(ORDER BY date)) AS prvs_close FROM it_dataset) a
GROUP BY sector
UNION
SELECT sector,SUM(IF((close>prvs_close),1,0)) AS higher_closed_price_days
FROM (SELECT sector,date,close,(LAG(close) OVER(ORDER BY date)) AS prvs_close FROM fmcg_dataset) a
GROUP BY sector
UNION
SELECT sector,SUM(IF((close>prvs_close),1,0)) AS higher_closed_price_days
FROM (SELECT sector,date,close,(LAG(close) OVER(ORDER BY date)) AS prvs_close FROM bank_dataset) a
GROUP BY sector)
SELECT * FROM CTE ORDER BY higher_closed_price_days;
```

➢ Strength

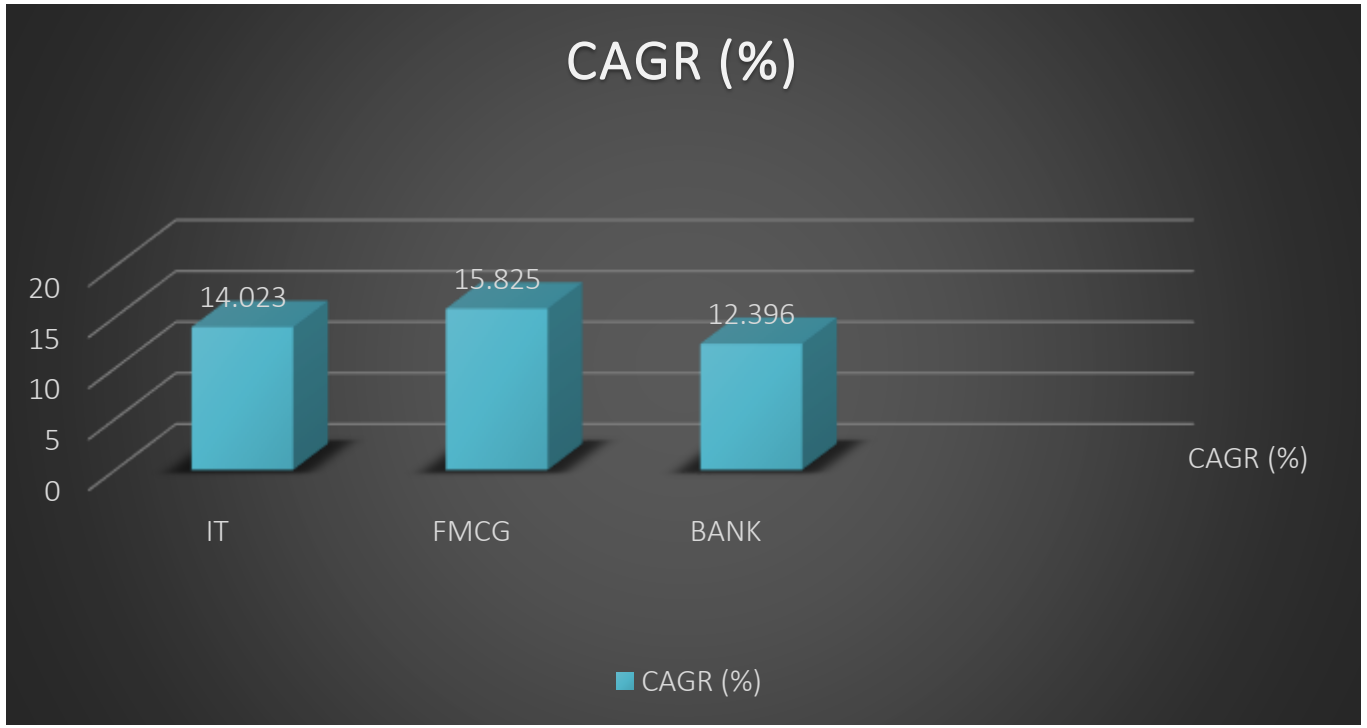
sector	higher_closed_price_days
IT	1494
BANK	1499
FMCG	1525

CAGR (COMPOUNDED ANNUAL GROWTH RATE)–

CAGR– The compounded annual growth rate (CAGR) is the rate of return (RoR) that would be required for an investment to grow from its beginning to its ending balance, assuming the profit ware reinvested at the end of each specific period of the investment's life plan.

Formula Used – $\text{CAGR} = \left(\frac{EV}{Bv}\right)^{\left(\frac{1}{n}\right)} - 1 \times 100$

Where, EV = Ending value, BV = Beginning value, n = Number of years



INSIGHTS -

CAGR– Higher the better

Bank – Lowest

FMCG - Highest

```
SET @number_of_years = (SELECT (MAX(year) - MIN(year))
                        FROM (SELECT YEAR(date) as year FROM it_dataset GROUP BY YEAR(date)) a);
SELECT @number_of_years;

SET @begin_price_it = (SELECT close FROM it_dataset WHERE date = '2011-02-01');
SET @end_price_it = (SELECT close FROM it_dataset WHERE date = '2022-08-01');

SET @begin_price_fmcg = (SELECT close FROM fmcg_dataset WHERE date = '2011-02-01');
SET @end_price_fmcg = (SELECT close FROM fmcg_dataset WHERE date = '2022-08-01');

SET @begin_price_bank = (SELECT close FROM bank_dataset WHERE date = '2011-02-01');
SET @end_price_bank = (SELECT close FROM bank_dataset WHERE date = '2022-08-01');
```

```
WITH CTE1  AS(
SELECT 'it_CAGR' AS Category,ROUND((POWER((@end_price_it/@begin_price_it),(1/@number_of_years)) - 1)*100,4) AS CAGR
UNION
SELECT 'fmcg_CAGR' AS Category,ROUND((POWER((@end_price_fmcg/@begin_price_fmcg),(1/@number_of_years)) - 1)*100,4) AS
UNION
SELECT 'bank_CAGR' AS Category,ROUND((POWER((@end_price_bank/@begin_price_bank),(1/@number_of_years)) - 1)*100,4) AS
SELECT * FROM CTE1 ORDER BY CAGR;
```

➢ CAGR Return.

Category	CAGR
bank_CAGR	12.3963
it_CAGR	14.0231
fmcg_CAGR	15.8251

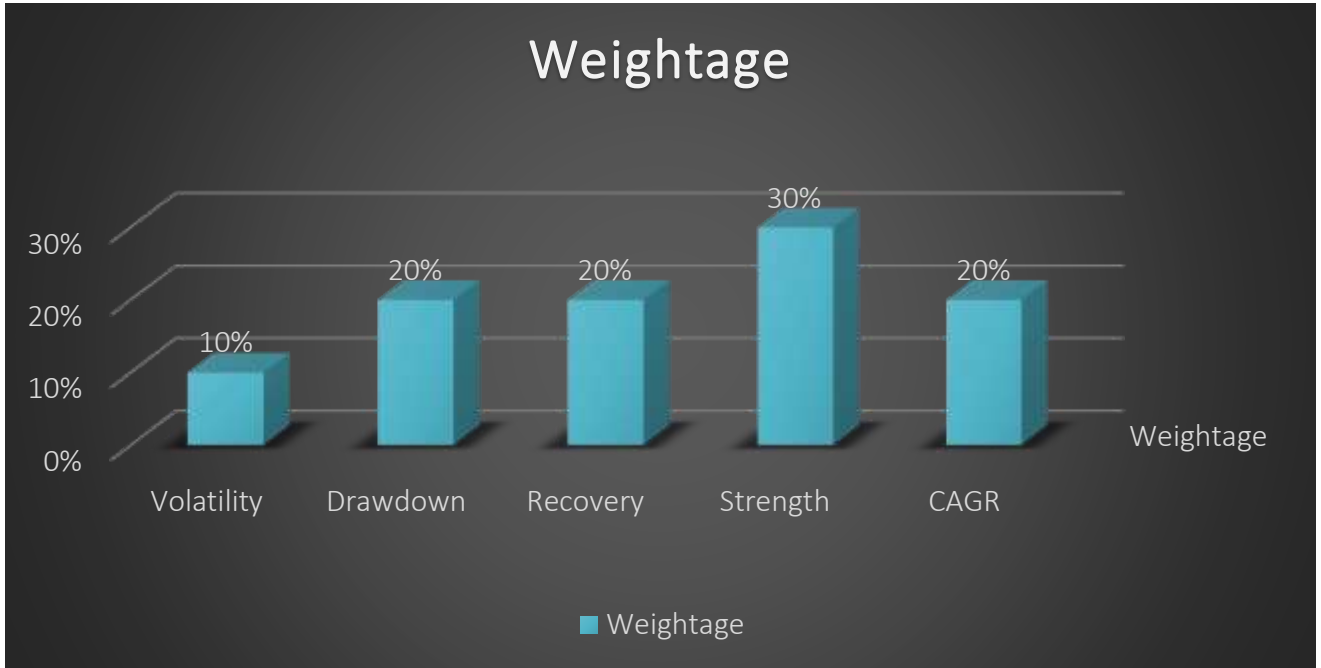
WEIGHTAGES AND FINAL SCORE

Weightages

Description	Weightage
Volatility	10%
Drawdown	20%
Recovery	20%
Higher_close_above	30%
CAGR	20%

Individual Score

Sector	Volatility	Drawdown	Recovery	Higher_close_above	CAGR
IT	3	2	2	1	2
FMCG	2	3	3	3	3
BANK	1	1	1	2	1

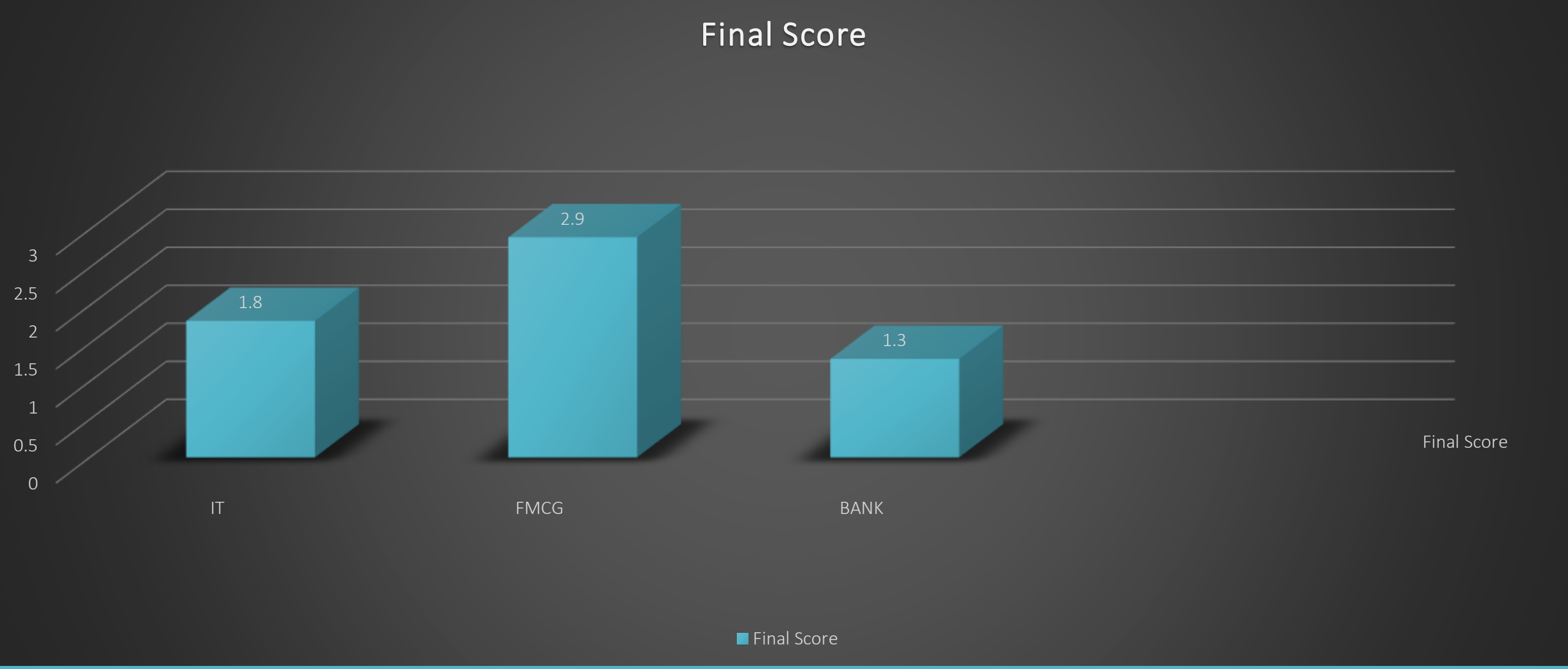


```sql
CREATE TABLE Score_Table (Sector varchar(50), `Description` VARCHAR(100), Score INT);

INSERT INTO Score_Table (Sector, `Description`, Score)
VALUES
("IT","Volatility",3),("FMCG","Volatility",2),("BANK","Volatility",1),
("IT","Drawdown",2),("FMCG","Drawdown",3),("BANK","Drawdown",1),
("IT","Recovery",2),("FMCG","Recovery",3),("BANK","Recovery",1),
("IT","Higher_close_above",1),("FMCG","Higher_close_above",3),("BANK","Higher_close_above",2),
("IT","CAGR",2),("FMCG","CAGR",3),("BANK","CAGR",1);


CREATE TABLE weightage_table(`Description` VARCHAR(100), Weightage decimal(2,2));

INSERT INTO weightage_table (`Description`,Weightage)
VALUES
("Volatility",0.1),
("Drawdown",0.2),
("Recovery",0.2),
("Higher_close_above",0.3),
("CAGR",0.2);
```

```sql
SELECT Sector, SUM(ROUND((Score*Weightage),10)) AS final_score FROM Score_Table st INNER JOIN weightage_table wt
ON st.`Description` = wt.`Description`
GROUP BY Sector
ORDER BY final_score DESC;
```

➢ Final Score

Sector	final_score
FMCG	2.90
IT	1.80
BANK	1.30



Conclusion -

After careful consideration of all parameters and assigning individual scores to each parameter, we proceeded to calculate the final score for each category based on the given percentage of parameter weightage. As a result, FMCG, IT, and BANK obtained the 1st, 2nd, and 3rd rankings, respectively.

References -

- Wikipedia, < https://en.wikipedia.org/wiki/Data_analysis>
- Fidelity International, https://www.fidelity.com.sg/beginners/what-is-volatility/market-volatility
- Wallstreetmojo team, https://www.wallstreetmojo.com/drawdown/

THANK YOU