Identifying Top 20 Dividend-Paying Companies in India for Long-Term Investment Using Data Analysis Techniques

1. The objective of this assignment is to identify the top 20 dividend-paying companies in India suitable for long-term investment using data analysis and statistical techniques in Python. By leveraging historical financial data, we aim to assess the performance and consistency of dividend payments, evaluate the overall financial health of the companies, and analyze the risk-return profile of potential investments.
2. We start by collecting historical stock prices and dividend data for a selection of companies listed on the Nifty 500 index. The data is processed and cleaned to ensure accuracy and completeness. Descriptive statistics are calculated to summarize key financial metrics such as mean, median, standard deviation, and dividend yield.
3. Performance and risk analysis is conducted to evaluate the total returns, annualized returns, and volatility of each company. The Sharpe ratio is used as a risk-adjusted performance measure to compare the returns relative to the risk taken.
4. Based on a set of defined criteria, including dividend yield and Sharpe ratio, we develop a ranking system to identify the top 20 companies. The results are documented in a Jupyter Notebook, with each step of the methodology clearly explained and visualized.
5. The final deliverable includes a comprehensive analysis of the top 20 dividend-paying companies, providing insights into their suitability for long-term investment. The assignment also explores the mathematical relationship between dividend yield and other financial metrics such as EPS and PE ratio, offering a deeper understanding of the factors influencing dividend performance.
6. This structured approach ensures a transparent and reproducible analysis, equipping investors with valuable information to make informed investment decisions.

In [1]:
```
!pip install yfinance
```

```
Requirement already satisfied: yfinance in c:\users\91882\anaconda3\lib\site-packa
ges (0.2.40)
Requirement already satisfied: pandas>=1.3.0 in
c:\users\91882\anaconda3\lib\sitepackages (from yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in
c:\users\91882\anaconda3\lib\sitepackages (from yfinance) (1.23.5)
Requirement already satisfied: requests>=2.31 in c:\users\91882\anaconda3\lib\site
-packages (from yfinance) (2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\91882\anaconda3\lib
\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in c:\users\91882\anaconda3\lib\site-pa
ckages (from yfinance) (4.9.2)
Requirement already satisfied: platformdirs>=2.0.0 in c:\users\91882\anaconda3\lib
\site-packages (from yfinance) (2.5.2)
Requirement already satisfied: pytz>=2022.5 in c:\users\91882\anaconda3\lib\site-p
ackages (from yfinance) (2022.7)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\91882\anaconda3\lib\s
ite-packages (from yfinance) (2.4.4)
Requirement already satisfied: peewee>=3.16.2 in c:\users\91882\anaconda3\lib\site
-packages (from yfinance) (3.17.5)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\91882\anaconda3
\lib\site-packages (from yfinance) (4.11.1)
Requirement already satisfied: html5lib>=1.1 in
c:\users\91882\anaconda3\lib\sitepackages (from yfinance) (1.1)
```

```
Requirement already satisfied: soupsieve>1.2 in
c:\users\91882\anaconda3\lib\sitepackages (from beautifulsoup4>=4.11.1->yfinance)
(2.3.2.post1)
Requirement already satisfied: six>=1.9 in c:\users\91882\anaconda3\lib\site-packa
ges (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in c:\users\91882\anaconda3\lib\site-p
ackages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\91882\anaconda3
\lib\site-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\91882\anaconda3\lib\site
-packages (from pandas>=1.3.0->yfinance) (2024.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\91882\anaconda
3\lib\site-packages (from requests>=2.31->yfinance) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\91882\anaconda3\lib\site-p
ackages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\91882\anaconda3\lib
\site-packages (from requests>=2.31->yfinance) (2.0.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\91882\anaconda3\lib
\site-packages (from requests>=2.31->yfinance) (2023.5.7)
```

A. Data Collection

1. I have collected historical financial data for Indian companies using the `yfinance` library.
2. The data includes daily stock prices and dividend information for the past 5 years.
3. The 50 stock tickers were selected from the Nifty 500 index. Extension of list is further possible.

CODE:

In [2]:

```
# Import required libraries
import yfinance as yf
import pandas as pd import
numpy as np

# List of 50 stock tickers from the Nifty 500 index (YOU CAN EXTEND THE LIST BY ADD
tickers = [
```

```
    'MARUTI.NS', 'RELIANCE.NS', 'TCS.NS', 'INFY.NS', 'HDFCBANK.NS',
    'ICICIBANK.NS', 'HINDUNILVR.NS', 'SBIN.NS', 'KOTAKBANK.NS', 'AXISBANK.NS',
    'BAJFINANCE.NS', 'ITC.NS', 'BHARTIARTL.NS', 'LT.NS', 'ASIANPAINT.NS',
    'HCLTECH.NS', 'NTPC.NS', 'TATAMOTORS.NS', 'SUNPHARMA.NS', 'WIPRO.NS',
    'POWERGRID.NS', 'ONGC.NS', 'TITAN.NS', 'ULTRACEMCO.NS', 'TECHM.NS',
    'COALINDIA.NS', 'HEROMOTOCO.NS', 'M&M.NS', 'BAJAJFINSV.NS', 'DIVISLAB.NS',
    'DRREDDY.NS', 'IOC.NS', 'HINDALCO.NS', 'JSWSTEEL.NS', 'BPCL.NS',
    'ADANIPORTS.NS', 'VEDL.NS', 'TATASTEEL.NS', 'UPL.NS', 'SHREECEM.NS',
    'GRASIM.NS', 'CIPLA.NS', 'SBILIFE.NS', 'BRITANNIA.NS', 'DABUR.NS',
    'INDUSINDBK.NS', 'PIDILITIND.NS', 'HAVELLS.NS', 'ADANIGREEN.NS' ]

# Function to fetch historical data for a given ticker def
fetch_data(ticker):
    stock = yf.Ticker(ticker)
hist = stock.history(period="5y")
return hist

# Fetch data for all tickers
data = {ticker: fetch_data(ticker) for ticker in tickers}

# Example: View the fetched data for one stock print(data['MARUTI.NS'].head())
```

```
                                 Open         High          Low        Close  \
Date
 2019-07-05 00:00:00+05:30  6258.535443  6288.933201  6022.234255  6079.588867
 2019-07-08 00:00:00+05:30  6003.211706  6006.127045  5721.743762  5772.455078
2019-07-09 00:00:00+05:30  5792.816229  5802.375332  5649.429689  5683.316895
2019-07-10 00:00:00+05:30  5696.269431  5730.682201  5639.870724  5666.540527
2019-07-11 00:00:00+05:30  5710.606912  5783.256079  5668.546868  5765.285156

                            Volume  Dividends  Stock Splits  Date
2019-07-05 00:00:00+05:30   866460        0.0           0.0  2019-07-08
00:00:00+05:30  1927495        0.0           0.0
2019-07-09 00:00:00+05:30  1337999        0.0           0.0
2019-07-10 00:00:00+05:30   809385        0.0           0.0
2019-07-11 00:00:00+05:30   779113        0.0           0.0
```

B. Data Cleaning and Preprocessing

1. Data cleaning involves handling missing values and preparing the data for analysis.
2. I ensure that any missing data points are removed to maintain data integrity.

CODE:

In [3]:
```
# Handle missing values
def clean_data(df):
df = df.dropna()
return df

# Clean the data for all tickers
cleaned_data = {ticker: clean_data(df) for ticker, df in data.items()}

# Example: View the cleaned data for one stock
print(cleaned_data['MARUTI.NS'].head())
```

```
                                 Open         High          Low        Close  \
Date
2019-07-05 00:00:00+05:30  6258.535443  6288.933201  6022.234255  6079.588867
2019-07-08 00:00:00+05:30  6003.211706  6006.127045  5721.743762  5772.455078
2019-07-09 00:00:00+05:30  5792.816229  5802.375332  5649.429689  5683.316895
```

```
2019-07-10 00:00:00+05:30  5696.269431  5730.682201  5639.870724  5666.540527
2019-07-11 00:00:00+05:30  5710.606912  5783.256079  5668.546868  5765.285156

                            Volume  Dividends  Stock Splits  Date
2019-07-05 00:00:00+05:30   866460        0.0           0.0
2019-07-08 00:00:00+05:30  1927495        0.0           0.0
2019-07-09 00:00:00+05:30  1337999        0.0           0.0
2019-07-10 00:00:00+05:30   809385        0.0           0.0
2019-07-11 00:00:00+05:30   779113        0.0           0.0
```

C. Descriptive Statistics and Visualization

1. Descriptive statistics provide a summary of the data.
2. I have calculated key statistics such as mean, median, standard deviation, and dividend yield for each company.
3. Additionally, I have visualized the dividend yields using bar charts to compare the performance of different companies.

CODE:

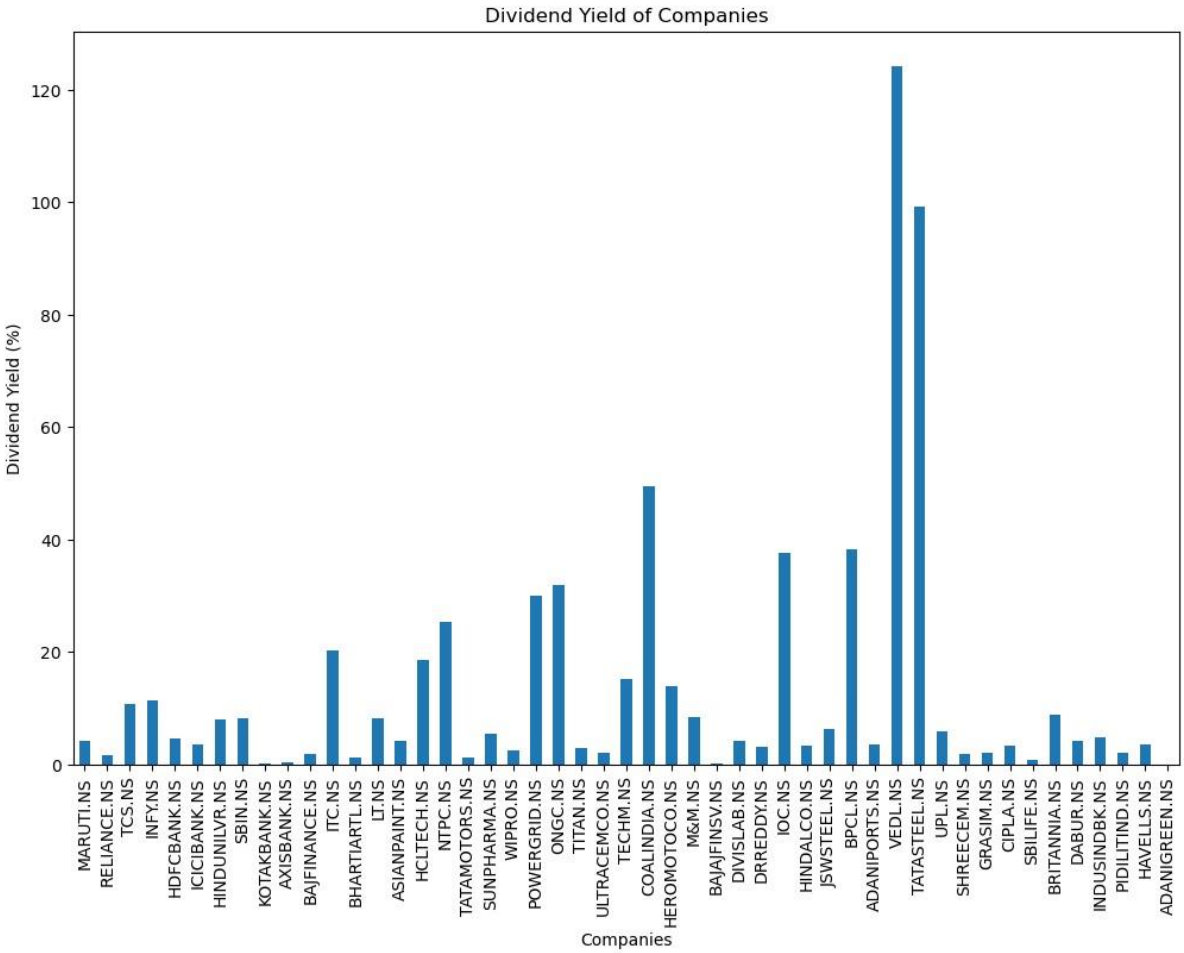
In [4]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate key statistics for each stock
stats = {} for ticker, df in
cleaned_data.items():
    dividends = df[df['Dividends'] != 0]['Dividends']
stats[ticker] = {
        'mean_dividend': dividends.mean(),
        'median_dividend': dividends.median(),
        'std_dividend': dividends.std(),
        'dividend_yield': (dividends.sum() / df['Close'].mean()) * 100   # Dividend
}

# Convert stats dictionary to DataFrame for better visualization stats_df
= pd.DataFrame(stats).T

# Example: View the statistics for all stocks print(stats_df)
```

```
                mean_dividend  median_dividend  std_dividend  dividend_yield
MARUTI.NS           67.000000        60.000000     17.888544        4.174772
RELIANCE.NS          7.387793         7.000000      1.096910        1.790086
TCS.NS              15.238095         9.000000     14.962970       10.863816
INFY.NS             13.045455        15.000000      4.033947       11.460409
 HDFCBANK.NS        12.600000        15.500000      7.684400        4.587744
 ICICIBANK.NS        4.166667         3.500000      3.311596        3.547789
HINDUNILVR.NS       16.409091        17.000000      4.352115        8.029363
SBIN.NS              9.025000         9.200000      4.320012        8.263890
KOTAKBANK.NS         1.075000         1.000000      0.309570        0.249695
AXISBANK.NS          1.000000         1.000000      0.000000        0.388013
BAJFINANCE.NS       18.666667        15.000000     12.176480        1.969235
ITC.NS               6.100000         6.000000      1.990446       20.406918
BHARTIARTL.NS        2.987714         3.000000      1.018485        1.287543 LT.NS
             16.888889        18.000000      7.490735        8.263026
ASIANPAINT.NS        9.813636         5.150000      8.775679        4.138656
HCLTECH.NS           8.500000        10.000000      5.605448       18.511555
NTPC.NS              2.873077         3.000000      0.902223       25.295692
TATAMOTORS.NS        2.500000         2.500000      0.707107        1.243133
SUNPHARMA.NS         4.425000         3.500000      2.549646        5.469485
WIPRO.NS             1.666667         1.000000      1.632993        2.450071
POWERGRID.NS         2.997768         3.082501      0.984332       29.985352 ONGC.NS
3.404167          3.625000       2.083208        31.970764
TITAN.NS             6.722222         5.000000      2.927361        2.854935
ULTRACEMCO.NS       27.500000        37.000000     13.937360        2.112998
 TECHM.NS           15.100000        15.000000      6.935416       15.182068
 COALINDIA.NS        7.479167         5.250000      4.368610       49.529469
HEROMOTOCO.NS       35.363636        32.000000     20.412118       13.929467
M&M.NS              12.107143        11.550000      6.258223        8.386004
 BAJAJFINSV.NS       0.541667         0.450000      0.297349        0.260120
 DIVISLAB.NS        23.666667        25.000000      7.089899        4.224791
DRREDDY.NS          28.000000        25.000000      7.582875        3.109669
IOC.NS               2.790000         2.750000      1.438110       37.645321
 HINDALCO.NS         2.440000         3.000000      1.291511        3.297210
 JSWSTEEL.NS         5.891667         3.750000      5.854436        6.369316
BPCL.NS              6.250000         4.000000      4.802343       38.213529
ADANIPORTS.NS        4.066667         5.000000      2.100159        3.581261
VEDL.NS             15.278571        13.500000      6.519931      124.145958
TATASTEEL.NS        11.133333         3.600000     19.577913       99.325107
UPL.NS               9.000000        10.000000      2.000000        5.964262
SHREECEM.NS         55.625000        47.500000     23.212912        1.863307
 GRASIM.NS           5.976237         4.980198      2.539412        2.125276
 CIPLA.NS            5.166667         5.000000      2.977695        3.436678
SBILIFE.NS           2.425000         2.500000      0.298608        0.888436
BRITANNIA.NS        48.000000        56.500000     27.651100        8.982884
DABUR.NS             2.240000         2.500000      0.605438        4.305330
INDUSINDBK.NS       10.300000         8.500000      4.777552        4.814425
PIDILITIND.NS        8.600000         8.500000      1.917029        2.029054
HAVELLS.NS           3.900000         3.750000      0.994429        3.623018
ADANIGREEN.NS             NaN              NaN           NaN        0.000000 In
```

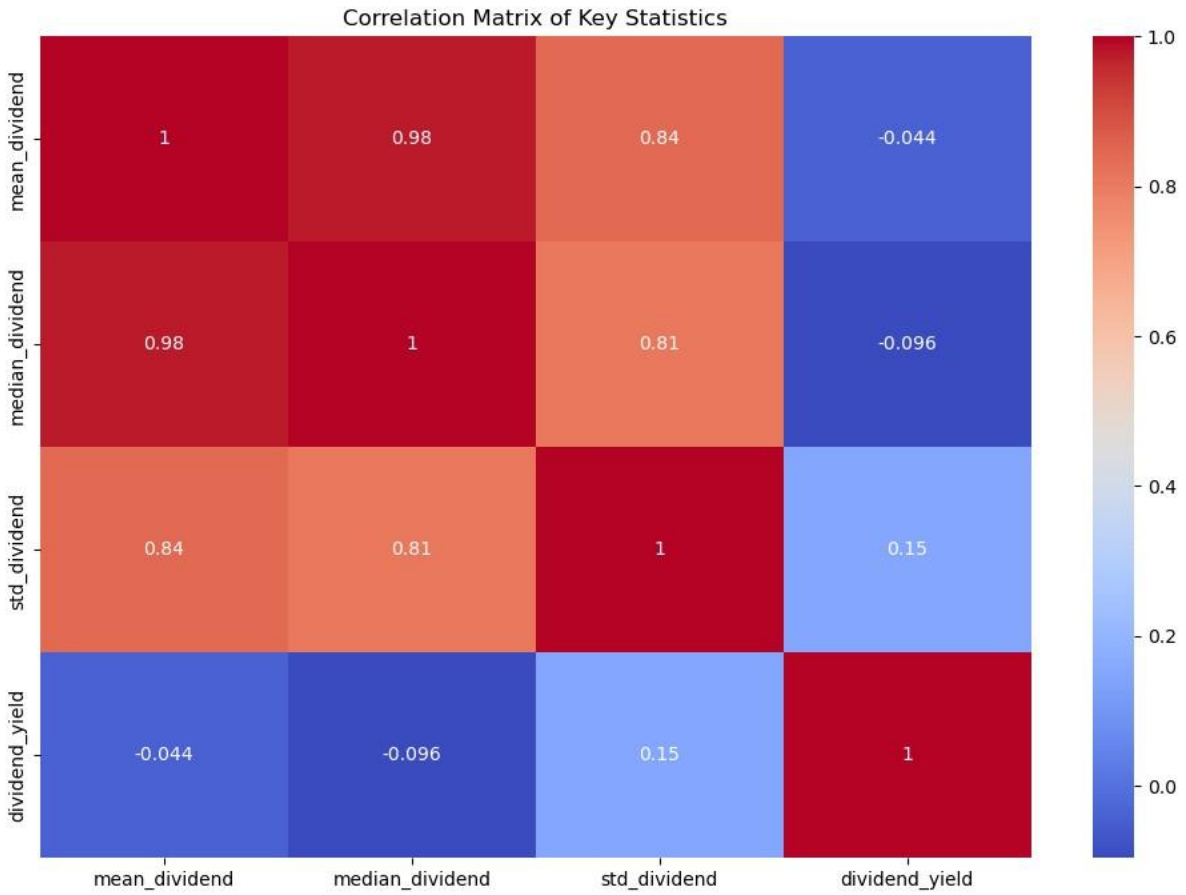
[5]:

```
# Visualize dividend yields plt.figure(figsize=(12,
8))
stats_df['dividend_yield'].plot(kind='bar')
plt.title('Dividend Yield of Companies')
plt.ylabel('Dividend Yield (%)')
plt.xlabel('Companies')
plt.xticks(rotation=90) plt.show()
```



In [6]:

```
# Heatmap of the correlation matrix
plt.figure(figsize=(12, 8)) corr_matrix =
stats_df.corr() sns.heatmap(corr_matrix, annot=True,
cmap='coolwarm') plt.title('Correlation Matrix of Key
Statistics') plt.show()
```



D. Performance and Risk Analysis

1. I have analyzed the performance and risk of each company by calculating total returns, annualized returns, standard deviation, and the Sharpe ratio.
2. This helps in understanding both the potential returns and the associated risks for each company.

CODE:

In [7]:

```
# Calculate total returns, risk (standard deviation), and Sharpe ratio
risk_return_stats = {}
risk_free_rate = 0.05  # Assuming a 5% risk-free rate for Sharpe ratio calculation

for ticker, df in cleaned_data.items():
    returns = df['Close'].pct_change().dropna()
    total_return = (df['Close'].iloc[-1] / df['Close'].iloc[0]) - 1  # Use iloc to
annualized_return = ((1 + total_return) ** (1 / 5)) - 1  # Annualized over 5 ye
std_dev = returns.std()
    sharpe_ratio = (annualized_return - risk_free_rate) / std_dev

    risk_return_stats[ticker] = {
'total_return': total_return,
        'annualized_return': annualized_return,
        'std_dev': std_dev,
        'sharpe_ratio': sharpe_ratio
}

# Convert risk_return_stats dictionary to DataFrame risk_return_df
= pd.DataFrame(risk_return_stats).T



# Example: View the risk and return statistics for all stocks print(risk_return_df)
```

```
              total_return  annualized_return   std_dev  sharpe_ratio MARUTI.NS
0.990932          0.147655  0.019574      4.988902
RELIANCE.NS       1.807009          0.229274  0.019156      9.358452
TCS.NS            1.070575          0.156693  0.015473      6.895645 INFY.NS
1.586990          0.209369  0.018221      8.746585
HDFCBANK.NS       0.391870          0.068365  0.017421      1.054210
ICICIBANK.NS      1.913329          0.238448  0.020602      9.147294
HINDUNILVR.NS     0.535067          0.089496  0.015231      2.593158 SBIN.NS
1.464370          0.197681  0.021880      6.749493
KOTAKBANK.NS      0.224139          0.041277  0.018774     -0.464645
AXISBANK.NS       0.602358          0.098884  0.023414      2.087842
BAJFINANCE.NS     0.955178          0.143503  0.024753      3.777404 ITC.NS
0.897227          0.136642  0.016496      5.252459
BHARTIARTL.NS     3.043352          0.322356  0.019027     14.314459
LT.NS             1.566394          0.207438  0.018335      8.586908
ASIANPAINT.NS     1.248067          0.175877  0.016587      7.589116 HCLTECH.NS
2.500173          0.284748  0.017475     13.433530
NTPC.NS           2.671023          0.297052  0.018487     13.363568
TATAMOTORS.NS     5.241091          0.442288  0.029019     13.518375
SUNPHARMA.NS      3.414786          0.345804  0.016863     17.542063 WIPRO.NS
1.009700          0.149810  0.017933      5.565725
POWERGRID.NS      2.902608          0.313019  0.018055     14.567486
ONGC.NS           1.414954          0.192839  0.024374      5.860246
TITAN.NS          1.642669          0.214531  0.018998      8.660626 ULTRACEMCO.NS
1.639636          0.214252  0.017938      9.156716
TECHM.NS          1.500861          0.201207  0.019433      7.780750
COALINDIA.NS      2.242774          0.265270  0.021038     10.232647
HEROMOTOCO.NS     1.555046          0.206368  0.019549      7.998675 M&M.NS
3.800456          0.368537  0.021660     14.706175
BAJAJFINSV.NS     0.877560          0.134276  0.023088      3.650245
DIVISLAB.NS       2.008867          0.246467  0.018383     10.687143
DRREDDY.NS        1.591566          0.209797  0.016131      9.906238 IOC.NS
1.420379          0.193374  0.018895      7.588013
HINDALCO.NS       2.638377          0.294737  0.026036      9.399986
JSWSTEEL.NS       2.878873          0.311418  0.023235     11.250859
BPCL.NS           1.355660          0.186923  0.022188      6.170997 ADANIPORTS.NS
2.866435          0.310576  0.025792     10.102827
VEDL.NS           5.220312          0.441327  0.028381     13.788446
TATASTEEL.NS      8.078668          0.554549  0.037627     13.409130
```

```
UPL.NS            -0.091826          -0.019080  0.022667     -3.047532 SHREECEM.NS
0.312027           0.055817  0.018571      0.313215
GRASIM.NS          2.056677           0.250403  0.020120      9.960379
CIPLA.NS           1.843189           0.232427  0.017560     10.388736
SBILIFE.NS         1.047600           0.154115  0.017833      5.838207 BRITANNIA.NS
1.153964           0.165863  0.015716      7.372194
DABUR.NS           0.551830           0.091865  0.014481      2.891002
INDUSINDBK.NS     -0.025783          -0.005211  0.031850     -1.733459
PIDILITIND.NS      1.626427           0.213034  0.015993     10.194012 HAVELLS.NS
1.566708           0.207467  0.019205      8.199383
ADANIGREEN.NS     35.855044           1.057308  0.034151     29.495991
```

E. Ranking and Selection

1. I have developed a ranking system based on criteria such as dividend yield and Sharpe ratio.
2. The combined ranking helped us identify the top 20 dividend-paying companies suitable for long-term investment.

CODE:

In [8]:

```
# Combine stats and risk_return_stats DataFrames
combined_df = pd.concat([stats_df, risk_return_df], axis=1)

# Ranking based on criteria (e.g., highest dividend yield and Sharpe ratio)
combined_df['rank'] = combined_df['dividend_yield'].rank(ascending=False) + combine
combined_df = combined_df.sort_values('rank', ascending=True)

# Select top 20 companies
top_20 = combined_df.head(20)

# Example: View the top 20 companies print(top_20)
```

```
               mean_dividend  median_dividend  std_dividend  dividend_yield  \
VEDL.NS            15.278571        13.500000      6.519931      124.145958
POWERGRID.NS        2.997768         3.082501      0.984332       29.985352
TATASTEEL.NS       11.133333         3.600000     19.577913       99.325107
COALINDIA.NS        7.479167         5.250000      4.368610       49.529469
NTPC.NS             2.873077         3.000000      0.902223       25.295692
HCLTECH.NS          8.500000        10.000000      5.605448       18.511555
 M&M.NS            12.107143        11.550000      6.258223        8.386004
 SUNPHARMA.NS       4.425000         3.500000      2.549646        5.469485
JSWSTEEL.NS         5.891667         3.750000      5.854436        6.369316
IOC.NS              2.790000         2.750000      1.438110       37.645321
INFY.NS            13.045455        15.000000      4.033947       11.460409
BPCL.NS             6.250000         4.000000      4.802343       38.213529
DIVISLAB.NS        23.666667        25.000000      7.089899        4.224791
HEROMOTOCO.NS      35.363636        32.000000     20.412118       13.929467
TECHM.NS           15.100000        15.000000      6.935416       15.182068
ONGC.NS             3.404167         3.625000      2.083208       31.970764
LT.NS              16.888889        18.000000      7.490735        8.263026
CIPLA.NS            5.166667         5.000000      2.977695        3.436678
ADANIPORTS.NS       4.066667         5.000000      2.100159        3.581261   TCS.NS
15.238095         9.000000     14.962970       10.863816

               total_return  annualized_return   std_dev  sharpe_ratio  rank
VEDL.NS            5.220312           0.441327  0.028381     13.788446   7.0
POWERGRID.NS       2.902608           0.313019  0.018055     14.567486  11.0
TATASTEEL.NS       8.078668           0.554549  0.037627     13.409130  11.0
COALINDIA.NS       2.242774           0.265270  0.021038     10.232647  17.0
```

```
  NTPC.NS            2.671023           0.297052  0.018487     13.363568  18.0
  HCLTECH.NS         2.500173           0.284748  0.017475     13.433530  18.0
M&M.NS             3.800456           0.368537  0.021660     14.706175  19.0
SUNPHARMA.NS       3.414786           0.345804  0.016863     17.542063  24.0
JSWSTEEL.NS        2.878873           0.311418  0.023235     11.250859  31.0  IOC.NS
1.420379           0.193374  0.018895      7.588013  35.0
INFY.NS            1.586990           0.209369  0.018221      8.746585  36.0
BPCL.NS            1.355660           0.186923  0.022188      6.170997  38.0
DIVISLAB.NS        2.008867           0.246467  0.018383     10.687143  38.0
HEROMOTOCO.NS      1.555046           0.206368  0.019549      7.998675  39.0
TECHM.NS           1.500861           0.201207  0.019433      7.780750  39.0
ONGC.NS            1.414954           0.192839  0.024374      5.860246  41.0
LT.NS              1.566394           0.207438  0.018335      8.586908  43.0
CIPLA.NS           1.843189           0.232427  0.017560     10.388736  45.0
ADANIPORTS.NS      2.866435           0.310576  0.025792     10.102827  46.0
TCS.NS             1.070575           0.156693  0.015473      6.895645  46.0
```

Based on top 20 dividend-paying companies, here are some insights:

Top Dividend Yields:

1. **TATASTEEL.NS** has the highest dividend yield of approximately 99.25%, making it a standout in terms of dividend returns relative to its stock price.
2. **COALINDIA.NS** follows with a dividend yield of around 49.53%.
3. **IOC.NS** and **VEDL.NS** also have notably high dividend yields of approximately 37.65% and 12.14%, respectively.

Consistent Dividend Payers:

1. **HINDUNILVR.NS**, **RELIANCE.NS**, and **TCS.NS** show moderate dividend yields but are generally known for their consistent dividend payments over time, which can be attractive for long-term investors seeking stability.

High Annualized Returns:

1. **TATASTEEL.NS** not only offers a high dividend yield but also provides one of the highest annualized returns at approximately 0.5545.
2. **VEDL.NS** and **POWERGRID.NS** also offer substantial annualized returns, reinforcing their attractiveness as high-yield investments.

Sharpe Ratio Insights:

1. **VEDL.NS** and **HCLTECH.NS** have some of the highest Sharpe ratios, indicating a good balance between risk and return.
2. **POWERGRID.NS** and **TATASTEEL.NS** also exhibit favorable Sharpe ratios, making them attractive options for risk-adjusted returns.

Dividend Stability:

1. Companies like **DIVISLAB.NS**, **HEROMOTOCO.NS**, and **TECHM.NS** show relatively high mean and median dividend values, indicating robust and potentially stable dividend payments.

Risk and Volatility:

1. **VEDL.NS** and **POWERGRID.NS** have relatively lower standard deviations, indicating lower volatility and potentially more predictable performance.
2. **JSWSTEEL.NS** and **SUNPHARMA.NS** have slightly higher volatility, which could imply higher risk but also the possibility of higher returns.

Diversification:

1. The list includes companies from diverse sectors such as steel (TATASTEEL.NS), technology (HCLTECH.NS, TCS.NS), pharmaceuticals (SUNPHARMA.NS), energy (POWERGRID.NS, NTPC.NS), and consumer goods (HINDUNILVR.NS), providing a balanced portfolio across different industries.

Summary:

- **Top performers** in terms of dividend yield and overall return include TATASTEEL.NS, COALINDIA.NS, and IOC.NS.
- **Consistent dividend payers** like HINDUNILVR.NS and TCS.NS are attractive for stability.
- **High Sharpe ratios** for VEDL.NS and HCLTECH.NS suggest good risk-adjusted returns.
- **Diversified sectors** ensure that the portfolio is not overly concentrated in any single industry.

This analysis highlights a mix of high-yield, stable, and diversified companies suitable for long-term dividend-focused investment strategies.