# QUANTITATIVE VALUE INVESTING STRATEGY

## Importing Libraries & Data

In [1]:
```
import pandas as pd
import numpy as np
import math
```

In [2]:
```
df = pd.read_csv('FUNDAMENTALratios.csv')
```

In [3]:
```
df
```

Out[3]:

| | symbol | revenuePerShare | trailingPE | earningsQuarterlyGrowth | previousClose | open | dayLow | dayHigh | volume | trailingEps | pegRatio | ebitda | totalDebt | totalRevenue | debtToEquity | revenuePerSha |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | RELIANCE.BO | 1296.823 | 28.012129 | 0.093 | 2901.30 | 2897.05 | 2895.35 | 2920.00 | 562484.0 | 103.88 | NaN | 1.503867e+12 | 3.166970e+12 | 8.773650e+12 | 36.100 | 1296 |
| 1 | RELIANCE.NS | 1296.823 | 28.010878 | 0.093 | 2901.95 | 2899.95 | 2894.70 | 2920.00 | 9763420.0 | 103.89 | NaN | 1.503867e+12 | 3.166970e+12 | 8.773650e+12 | 36.100 | 1296 |
| 2 | HDFCBANK.BO | 293.740 | 16.260424 | 0.359 | 1445.10 | 1437.30 | 1437.30 | 1450.70 | 783356.0 | 88.74 | NaN | NaN | 7.996598e+12 | 1.933224e+12 | NaN | 293 |
| 3 | HINDUNILVR.NS | 263.412 | 51.285230 | 0.014 | 2242.35 | 2239.05 | 2232.05 | 2266.00 | 3507581.0 | 44.00 | NaN | 1.421525e+11 | 1.272000e+10 | 6.190100e+11 | 2.515 | 263 |
| 4 | ICICIBANK.NS | 197.399 | 18.315136 | 0.257 | 1081.80 | 1081.15 | 1078.70 | 1093.70 | 17212189.0 | 59.53 | 1.03 | NaN | 2.009669e+12 | 1.380849e+12 | NaN | 197 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 5905 | NIESSPM.BO | NaN | NaN | NaN | 2.16 | 2.16 | 2.16 | 2.16 | 200.0 | NaN | NaN | NaN | NaN | NaN | NaN | |
| 5906 | QGOLDHALF.NS | NaN | NaN | NaN | 56.27 | 56.20 | 55.43 | 56.20 | 14972.0 | NaN | NaN | NaN | NaN | NaN | NaN | |
| 5907 | HDFCMFGETF.NS | NaN | NaN | NaN | 51.67 | 51.81 | 51.51 | 51.81 | 385162.0 | NaN | NaN | NaN | NaN | NaN | NaN | |
| 5908 | UTIBANKETF.NS | NaN | 19.812199 | NaN | 47.53 | 47.85 | 47.20 | 48.35 | 136993.0 | NaN | NaN | NaN | NaN | NaN | NaN | |
| 5909 | KOTAKGOLD.NS | NaN | NaN | NaN | 57.02 | 56.64 | 56.22 | 56.70 | 416314.0 | NaN | NaN | NaN | NaN | NaN | NaN | |

5910 rows × 18 columns

## Data Wrangling

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
symbol                    0
revenuePerShare        2675
trailingPE             2528
earningsQuarterlyGrowth    3584
previousClose           580
open                    578
dayLow                  578
dayHigh                 578
volume                  578
trailingEps            2668
pegRatio               5557
ebitda                 2898
totalDebt              2839
totalRevenue           2676
debtToEquity           3170
revenuePerShare.1      2675
earningsGrowth         3622
revenueGrowth          2739
dtype: int64
```

In [5]:
```
mean_values = df.select_dtypes(include=[np.number]).mean()
df.fillna(mean_values, inplace=True)          # Fill missing values with the mean of each column
```

In [6]:
```
df['trailingPE'] = df['trailingPE'].replace([np.inf, -np.inf], 1000)
# Replace infinite P/E ratios with a high value (e.g., 1000) and handle negative earnings (P/E should be very high)
```

In [7]:
```
df.duplicated().sum()
```

Out[7]:
```
81
```

In [8]:
```
df = df.drop_duplicates()     # Removing duplicates
```

In [9]:
```
df = df.assign(BaseCompany=df['symbol'].str.split('.').str[0])
df = df.drop_duplicates(subset='BaseCompany', keep='first')
# Keeping only one copy of companies that appear twice as they are listed on both nse and bse (eg reliance.bo and reliance.ns)
```

In [10]:
```
df = df.drop(columns = ['previousClose', 'dayLow', 'dayHigh', 'volume', 'BaseCompany'])
# Removing non-essential columns
```

In [11]:
```
df.rename(columns={'open': 'sharePrice'}, inplace=True)
```

In [12]:
```
df
```

Out[12]:

| | symbol | revenuePerShare | trailingPE | earningsQuarterlyGrowth | sharePrice | trailingEps | pegRatio | ebitda | totalDebt | totalRevenue | debtToEquity | revenuePerShare.1 | earningsGrowth | revenueGrowth |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | RELIANCE.BO | 1296.823000 | 28.012129 | 0.093000 | 2897.05 | 103.880000 | 7.752125 | 1.503867e+12 | 3.166970e+12 | 8.773650e+12 | 36.100000 | 1296.823000 | 0.093000 | 0.036000 |
| 2 | HDFCBANK.BO | 293.740000 | 16.260424 | 0.359000 | 1437.30 | 88.740000 | 7.752125 | 1.233124e+10 | 7.996598e+12 | 1.933224e+12 | 118.158228 | 293.740000 | -0.001000 | 1.211000 |
| 3 | HINDUNILVR.NS | 263.412000 | 51.285230 | 0.014000 | 2239.05 | 44.000000 | 7.752125 | 1.421525e+11 | 1.272000e+10 | 6.190100e+11 | 2.515000 | 263.412000 | 0.014000 | -0.002000 |
| 4 | ICICIBANK.NS | 197.399000 | 18.315136 | 0.257000 | 1081.15 | 59.530000 | 1.030000 | 1.233124e+10 | 2.009669e+12 | 1.380849e+12 | 118.158228 | 197.399000 | 0.253000 | 0.204000 |
| 6 | INFY.NS | 4.481000 | 25.573729 | -0.084000 | 1521.00 | 59.000000 | 2.960000 | 4.250000e+09 | 1.051000e+09 | 1.855200e+10 | 10.871000 | 4.481000 | -0.068000 | 0.001000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 5894 | DIGJAMLTD.BO | 13.545000 | 1000.000000 | 1.272781 | 83.10 | -6.000000 | 7.752125 | -9.186000e+07 | 5.974000e+08 | 2.711140e+08 | 292.556000 | 13.545000 | 1.072592 | -0.353000 |
| 5895 | BHATEXT.BO | 368.680045 | 1000.000000 | 1.272781 | 8.40 | 18.468205 | 7.752125 | 1.233124e+10 | 7.261417e+10 | 8.765737e+10 | 118.158228 | 368.680045 | 1.072592 | 0.490909 |
| 5897 | COCKERILL.BO | 368.680045 | 1000.000000 | 1.272781 | 3432.95 | 18.468205 | 7.752125 | 1.233124e+10 | 7.261417e+10 | 8.765737e+10 | 118.158228 | 368.680045 | 1.072592 | 0.490909 |
| 5899 | RITCO.BO | 356.090000 | 19.266376 | 0.349000 | 224.40 | 11.450000 | 7.752125 | 6.125690e+08 | 2.619718e+09 | 8.779346e+09 | 159.960000 | 356.090000 | 0.337000 | 0.228000 |
| 5905 | NIESSPM.BO | 368.680045 | 1000.000000 | 1.272781 | 2.16 | 18.468205 | 7.752125 | 1.233124e+10 | 7.261417e+10 | 8.765737e+10 | 118.158228 | 368.680045 | 1.072592 | 0.490909 |

4207 rows × 14 columns

## Calculation of Financial Ratios

We are calculating the following ratios and representing them in percentiles for comparison :

- EBITDA Margin
- Debt to Equity Ratio
- Earnings Growth
- Revenue Growth
- PE Ratio
- Earnings Quarterly Growth
- PEG Ratio
- Revenue Per Share
- EPS

In [13]:
```
df['EBITDA Margin'] = df['ebitda'] / df['totalRevenue']
```

In [14]:
```
df['debtToEquity'] = 1 / df['debtToEquity']
```

In [15]:
```
df['trailingPE'] = 1 / df['trailingPE']
```

In [16]:
```
df['revenuePerShare'] = (df['revenuePerShare'] / df['sharePrice']) * 100
```

In [17]:
```
df['pegRatio'] = 1 / df['pegRatio']
```

In [18]:
```
df['trailingEps'] = (df['trailingEps'] / df['sharePrice']) * 100
```

In [19]:
```
ratios = ['EBITDA Margin', 'debtToEquity', 'earningsGrowth', 'revenueGrowth', 'trailingPE', 'earningsQuarterlyGrowth', 'pegRatio', 'revenuePerShare', 'trailingEps']

for ratio in ratios:
    df[f'{ratio} Percentile'] = df[f'{ratio}'].rank(pct = True) * 100
```

We create a 'compositeScore' which is an average of all the ratios

In [20]:
```
ratiosPercentile = [ratio + ' Percentile' for ratio in ratios]
```

In [21]:
```
df['compositeScore'] = df[ratiosPercentile].mean(axis = 1)
```

In [22]:
```
df
```

Out[22]:

| | symbol | revenuePerShare | trailingPE | earningsQuarterlyGrowth | sharePrice | trailingEps | pegRatio | ebitda | totalDebt | totalRevenue | ... | EBITDA Margin Percentile | debtToEquity Percentile | earningsGrowth Percentile | revenueGrowth Percentile | trailingPE Percentile |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | RELIANCE.BO | 44.763570 | 0.035699 | 0.093000 | 2897.05 | 3.585717 | 0.128997 | 1.503867e+12 | 3.166970e+12 | 8.773650e+12 | ... | 86.522463 | 80.817685 | 13.025909 | 17.435227 | 76.515332 |
| 2 | HDFCBANK.BO | 20.436930 | 0.061499 | 0.359000 | 1437.30 | 6.174076 | 0.128997 | 1.233124e+10 | 7.996598e+12 | 1.933224e+12 | ... | 6.013787 | 37.746613 | 10.316140 | 98.597575 | 88.067507 |
| 3 | HINDUNILVR.NS | 11.764454 | 0.019499 | 0.014000 | 2239.05 | 1.965119 | 0.128997 | 1.421525e+11 | 1.272000e+10 | 6.190100e+11 | ... | 91.371524 | 94.889470 | 10.708343 | 14.511528 | 63.845971 |
| 4 | ICICIBANK.NS | 18.258244 | 0.054600 | 0.257000 | 1081.15 | 5.506174 | 0.970874 | 1.233124e+10 | 2.009669e+12 | 1.380849e+12 | ... | 6.251486 | 37.746613 | 16.686475 | 29.819349 | 85.785595 |
| 6 | INFY.NS | 0.294609 | 0.039103 | -0.084000 | 1521.00 | 3.879027 | 0.337838 | 4.250000e+09 | 1.051000e+09 | 1.855200e+10 | ... | 91.300214 | 89.137152 | 9.032565 | 14.618493 | 78.892322 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 5894 | DIGJAMLTD.BO | 16.299639 | 0.001000 | 1.272781 | 83.10 | -7.220217 | 0.128997 | -9.186000e+07 | 5.974000e+08 | 2.711140e+08 | ... | 2.056097 | 1.937247 | 60.339910 | 3.018778 | 24.768243 |
| 5895 | BHATEXT.BO | 4389.048160 | 0.001000 | 1.272781 | 8.40 | 219.859577 | 0.128997 | 1.233124e+10 | 7.261417e+10 | 8.765737e+10 | ... | 53.791300 | 37.746613 | 60.339910 | 66.401236 | 24.768243 |
| 5897 | COCKERILL.BO | 10.739453 | 0.001000 | 1.272781 | 3432.95 | 0.537969 | 0.128997 | 1.233124e+10 | 7.261417e+10 | 8.765737e+10 | ... | 53.791300 | 37.746613 | 60.339910 | 66.401236 | 24.768243 |
| 5899 | RITCO.BO | 158.685383 | 0.051904 | 0.349000 | 224.40 | 5.102496 | 0.128997 | 6.125690e+08 | 2.619718e+09 | 8.779346e+09 | ... | 13.120989 | 3.613026 | 18.576183 | 30.995959 | 84.668410 |
| 5905 | NIESSPM.BO | 17068.520622 | 0.001000 | 1.272781 | 2.16 | 855.009468 | 0.128997 | 1.233124e+10 | 7.261417e+10 | 8.765737e+10 | ... | 53.791300 | 37.746613 | 60.339910 | 66.401236 | 24.768243 |

4207 rows × 25 columns

In [23]:
```
topStocks = df.sort_values (by = 'compositeScore', ascending = False).head(10).reset_index (drop = True)
topStocks
```

Out[23]:

| | symbol | revenuePerShare | trailingPE | earningsQuarterlyGrowth | sharePrice | trailingEps | pegRatio | ebitda | totalDebt | totalRevenue | ... | EBITDA Margin Percentile | debtToEquity Percentile | earningsGrowth Percentile | revenueGrowth Percentile | trailingPE Percentile |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | JINDCOT.NS | 14747.201818 | 9.100000 | 10.086 | 2.50 | 910.000000 | 0.128997 | -6.656150e+07 | 3.876200e+08 | -1.000000e+03 | ... | 100.000000 | 84.882339 | 99.381982 | 66.401236 | 99.976230 |
| 1 | HBSL.NS | 61.268414 | 0.452469 | 3.140 | 80.10 | 45.755306 | 0.128997 | 3.233400e+08 | 1.097100e+07 | 3.503700e+08 | ... | 98.098407 | 96.600903 | 98.454956 | 99.477062 | 99.239363 |
| 2 | KEYFINSERV.NS | 56.682779 | 0.201628 | 2.476 | 148.95 | 19.959718 | 0.128997 | 1.233124e+10 | 6.268000e+07 | 4.096370e+08 | ... | 99.477062 | 92.179700 | 98.074638 | 98.859044 | 98.027098 |
| 3 | UTIAMC.NS | 15.474596 | 0.064606 | 2.094 | 819.75 | 6.577615 | 1.219512 | 9.048775e+09 | 1.373600e+09 | 1.621580e+10 | ... | 96.886142 | 94.081293 | 97.492275 | 96.054195 | 88.685524 |
| 4 | SOUTHBANK.NS | 75.968142 | 0.169257 | 1.965 | 28.25 | 17.345133 | 0.657895 | 1.233124e+10 | 4.212540e+10 | 4.490380e+10 | ... | 93.391966 | 37.746613 | 97.314000 | 96.767293 | 97.432850 |
| 5 | 63MOONS.BO | 29.065686 | 0.102898 | 3.218 | 408.00 | 10.139706 | 0.128997 | 1.611769e+09 | 5.520700e+07 | 5.461806e+09 | ... | 94.033753 | 99.025434 | 98.502496 | 97.302116 | 94.580461 |
| 6 | ALMONDZ.NS | 32.086364 | 0.102437 | 3.887 | 110.00 | 10.509091 | 0.128997 | 1.233124e+10 | 4.512000e+08 | 9.544060e+08 | ... | 99.310673 | 85.072498 | 98.621345 | 97.004992 | 94.532921 |
| 7 | IIFLSEC.NS | 51.834050 | 0.115274 | 1.324 | 116.30 | 11.745486 | 0.128997 | 1.233124e+10 | 9.011582e+09 | 1.842287e+10 | ... | 97.385310 | 76.586641 | 96.149275 | 96.838602 | 95.792726 |
| 8 | WELCORP.NS | 123.049772 | 0.076705 | 11.559 | 526.40 | 7.809650 | 1.075269 | 1.730230e+10 | 1.989440e+10 | 1.694858e+11 | ... | 18.279059 | 80.199667 | 99.453292 | 98.288567 | 91.347754 |
| 9 | NATCOPHARM.NS | 22.184535 | 0.074397 | 2.414 | 957.00 | 7.401254 | 0.128997 | 1.631375e+10 | 2.475000e+09 | 3.829800e+10 | ... | 96.173045 | 92.702638 | 98.050868 | 96.220585 | 90.896126 |

10 rows × 25 columns

In [24]:
```
investmentAmount = float(input("Enter the investment amount: "))
```
```
Enter the investment amount: 20000
```

In [25]:
```
totalCompositeScore = topStocks['compositeScore'].sum()
```

In [26]:
```
topStocks['allocatedAmount'] = (topStocks['compositeScore'] / totalCompositeScore) * investmentAmount
```

In [27]:
```
topStocks['numberOfStocks'] = (topStocks['allocatedAmount'] / topStocks['sharePrice']).apply(math.floor)
```

In [28]:
```
topStocks['investedAmount'] = topStocks['numberOfStocks'] * topStocks['sharePrice']
```

In [29]:
```
total_invested_amount = topStocks['investedAmount'].sum()
```

In [30]:
```
output_df = topStocks[['symbol', 'sharePrice', 'numberOfStocks', 'investedAmount']]
total_row = pd.DataFrame({['', '', '', total_invested_amount]], columns=output_df.columns, index=['Total'])
output_df = pd.concat([output_df, total_row])
```

In [31]:
```
display (output_df)
output_file_path = 'investmentAllocation.xlsx'
output_df.to_excel(output_file_path, index=False)

print(f"Investment allocation saved to {output_file_path}")
```

| | symbol | sharePrice | numberOfStocks | investedAmount |
|---|---|---|---|---|
| 0 | JINDCOT.NS | 2.5 | 879 | 2197.50 |
| 1 | HBSL.NS | 80.1 | 26 | 2082.60 |
| 2 | KEYFINSERV.NS | 148.95 | 13 | 1936.35 |
| 3 | UTIAMC.NS | 819.75 | 2 | 1639.50 |
| 4 | SOUTHBANK.NS | 28.25 | 71 | 2005.75 |
| 5 | 63MOONS.BO | 408.0 | 4 | 1632.00 |
| 6 | ALMONDZ.NS | 110.0 | 17 | 1870.00 |
| 7 | IIFLSEC.NS | 116.3 | 16 | 1860.80 |
| 8 | WELCORP.NS | 526.4 | 3 | 1579.20 |
| 9 | NATCOPHARM.NS | 957.0 | 1 | 957.00 |
| **Total** | | | | 17760.70 |

```
Investment allocation saved to investmentAllocation.xlsx
```