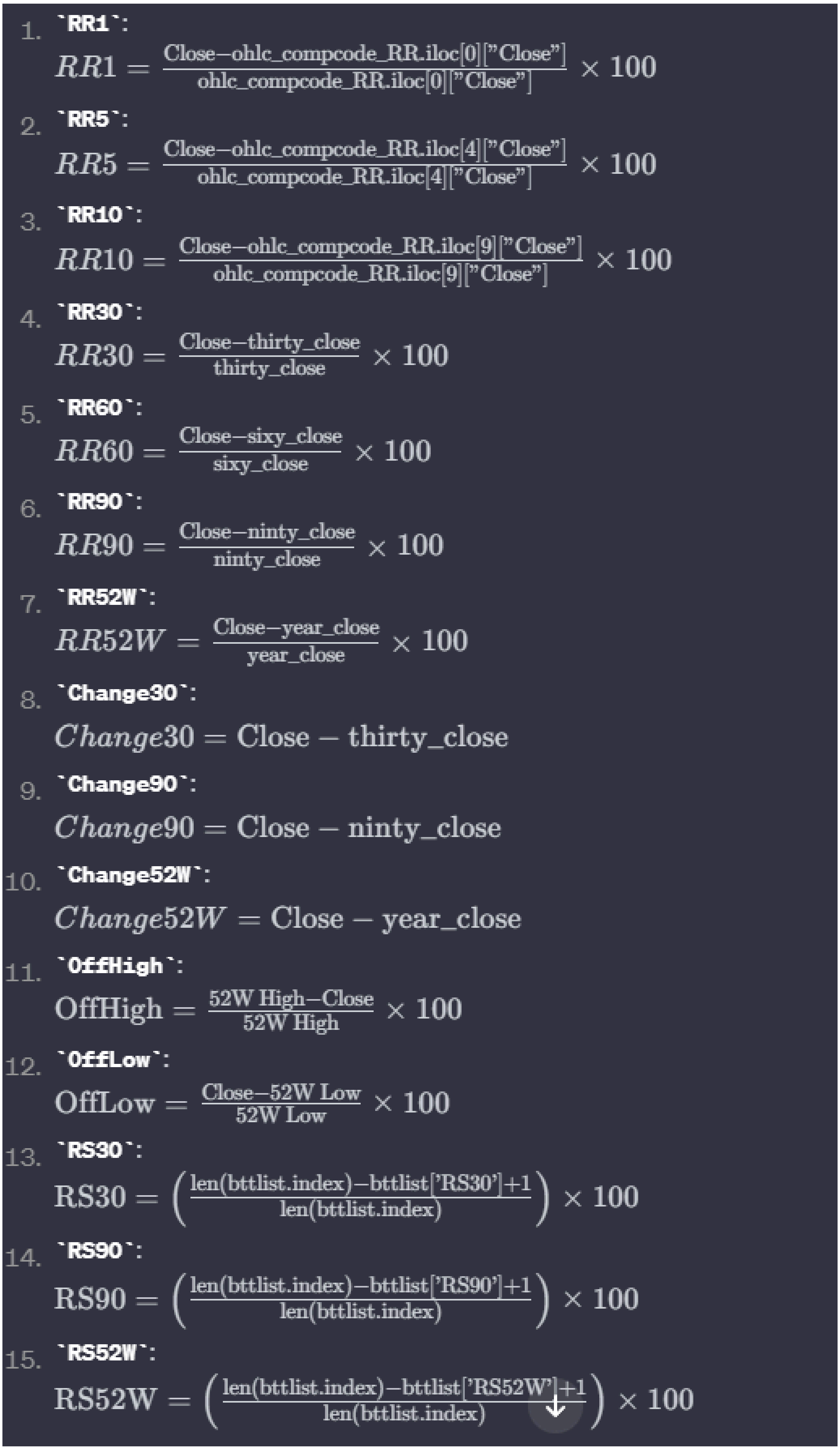
```
   Script running for  2023-12-26
Initiating QuarterlyEPS & TTM Process.... Date: 2023-12-26

Compiling quarterly EPS list
End Qtr:  2023-09-30
CompanyCode NULL List:  3
prev_4year:  2019-09-30
Calculating quarterly one EPS Sales growth
Calculating quarterly two EPS Sales growth
Inserting Quarterly EPS Results
Calculating TTM values

Compiling quarterly EPS list
End Qtr:  2023-09-30
CompanyCode NULL List:  0
Data not present for quarterly EPS list for ....Date: 2023-12-26
BTT Back: 2023-12-01 BTT Next: 2024-01-01
Today:  2023-12-26
Calculating EPS Rating
Calculating Consolidated EPS Rating
Number of entries from consolidated data:  695
Merging Null Set Back into list
Calculating Stock Percentile Ranking
Inserting into EPS..
Compiling Ratios Merge List for today 2023-12-26
Compiling list from RatiosBanking and NonBanking...
Calculating sales growth and npm for ttm year...

Starting loop---

2662    2023-09-30
Name: YearEnding, dtype: object
Merging background info to Ratios list
Inserting values into Ratios merge list
Inserting Ratios Merge List:
Fetching BTT stocks for SMR Ranking
number of stocks: 994
Calculating percentile values for NPM, ROE and Sales Growth
Calculating SMR Rank
Inserting into SMR...

Generating MF List for today: 2023-12-26
Filtering schemes from Scheme Master
Length of scheme master list: 1229
Merging filtered list with schemewise portfolio
Merging with Indutry mapping
Calculating Market Cap
```

```
C:\Users\dsram\OneDrive\Desktop\Braviza\app\reports\FRS.py:172: FutureWarning: Setting
an item of incompatible dtype is deprecated and will raise in a future error of pandas. Value
'Large Cap' has dtype incompatible with float64, please explicitly cast to a compatible dtype
first.
  scheme_mf_list.loc[index, 'Market Cap'] = market_cap
Inserting into MFList
Compiled MF merge list
Calculating MF Exposure and Rank
Index Size for MF:  1003
Inserting into table
Completed MF Rank generation

Generating NAV Rank and category average for today: 2023-12-26
Filtering schemes from scheme master
Merging with SchemeNavMaster
Getting BTT MF Categories
Getting Scheme NAV current prices data
Calculating Scheme Rank
Inserting Scheme NAV Rank in table
Inserting Scheme NAV List:
Calculating average of each group
Inserting NAV Category Average in table
Inserting Scheme NAV category average List:
Completed NAV Rank and category average
Getting NAV data from FRS-NAVRank
Set MF OHLC from NAV data
Getting category averages for schemes
Frs_nav Rank and Frs_nav Average row wise
Inserting into the DB

                OHLC Fetch Service Started..........



NSE Fetch invoked ....



BSE Fetch invoked ....
Csv file:
C:\Users\dsram\OneDrive\Desktop\Braviza\app\OHLCFiles\EQ_ISINCODE_261223.CSV
Attempting to insert NSE data into DB
NSE Insert Completed
Attempting to insert BSE data into DB
BSE Insert Completed
OHLC Insert Completed

                OHLC Fetch Completed.
                Index OHLC Fetch Service
index OHLC date
 2023-12-26
Fetching IndexOHLC from NSE
Merging with index BTTCode
Filling empty values
Inserting into index OHLC table
C:\Users\dsram\OneDrive\Desktop\Braviza\app\lib\index_ohlc.py:158:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
rsus-a-copy
  index_ohlc[index_ohlc.columns[7:]] = index_ohlc[index_ohlc.columns[7:]].replace(r'\-', '-1',
regex=True).astype(float)
C:\Users\dsram\OneDrive\Desktop\Braviza\app\lib\index_ohlc.py:159:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
rsus-a-copy
  index_ohlc[index_ohlc.columns[7:]] = index_ohlc[index_ohlc.columns[7:]].replace(-1,
np.nan)
Index OHLC Insert Completed
Merging with OHLC
Merged with NSE

                Index OHLC Fetch Completed.
Getting splits data for date 2023-12-26
Getting bonus data for date 2023-12-26
Updating OHLC
No split data for today
No bonus data for today
Updating Shareholding
No split data for today
No Bonus data for today
Getting OHLC list for Date: 2023-12-26
Calculating PE for OHLC list
PE high/low calc
Inserting PE
Inserted PE for date:  2023-12-26
year_back
 2022-12-25
Today date: 2023-12-26
```

```
--------------------------------------------------------------------------------
Fetching BTTList for Date:  2023-12-26
Fetching OHLC Data
        ISIN    NSECode   BSECode        CompanyName  CompanyCode 52W High 52W
Low 52W High Date 52W Low Date  ... Off-High Off-Low     Open     High     Low   Close
Date         Value      Volume
0  INE208A01029   ASHOKLEY  500477.0  Ashok Leyland Ltd.   10510001.0      NaN
NaN          NaN         NaN  ...      NaN     NaN   173.25   174.90   172.5   173.9  2023-12-26
2.227437e+09  12825482.0
1  INE208A01029   ASHOKLEY  500477.0  Ashok Leyland Ltd.   10510001.0      NaN
NaN          NaN         NaN  ...      NaN     NaN   173.25   174.90   172.5   173.9  2023-12-26
2.227437e+09  12825482.0
2  INE451A01017   FORCEMOT  500033.0   Force Motors Ltd.   10510002.0      NaN     NaN
NaN         NaN  ...      NaN     NaN      NaN      NaN     NaN     NaN  2023-12-26           NaN
NaN
3  INE066A01021  EICHERMOT  505200.0  Eicher Motors Ltd.   10510004.0      NaN     NaN
NaN         NaN  ...      NaN     NaN  4024.80  4069.65  4012.0  4040.4  2023-12-26
1.922477e+09    475107.0
4  INE066A01021  EICHERMOT  505200.0  Eicher Motors Ltd.   10510004.0      NaN     NaN
NaN         NaN  ...      NaN     NaN  4024.80  4069.65  4012.0  4040.4  2023-12-26
1.922477e+09    475107.0

[5 rows x 42 columns]
Calculating High/Low & Value Avg
```



1. `RR1`:

$$RR1 = \frac{\text{Close}-\text{ohlc\_compcode\_RR.iloc[0][”Close”]}}{\text{ohlc\_compcode\_RR.iloc[0][”Close”]}} \times 100$$

2. `RR5`:

$$RR5 = \frac{\text{Close}-\text{ohlc\_compcode\_RR.iloc[4][”Close”]}}{\text{ohlc\_compcode\_RR.iloc[4][”Close”]}} \times 100$$

3. `RR10`:

$$RR10 = \frac{\text{Close}-\text{ohlc\_compcode\_RR.iloc[9][”Close”]}}{\text{ohlc\_compcode\_RR.iloc[9][”Close”]}} \times 100$$

4. `RR30`:

$$RR30 = \frac{\text{Close}-\text{thirty\_close}}{\text{thirty\_close}} \times 100$$

5. `RR60`:

$$RR60 = \frac{\text{Close}-\text{sixy\_close}}{\text{sixy\_close}} \times 100$$

6. `RR90`:

$$RR90 = \frac{\text{Close}-\text{ninty\_close}}{\text{ninty\_close}} \times 100$$

7. `RR52W`:

$$RR52W = \frac{\text{Close}-\text{year\_close}}{\text{year\_close}} \times 100$$

8. `Change30`:

$$Change30 = \text{Close} - \text{thirty\_close}$$

9. `Change90`:

$$Change90 = \text{Close} - \text{ninty\_close}$$

10. `Change52W`:

$$Change52W = \text{Close} - \text{year\_close}$$

11. `OffHigh`:

$$\text{OffHigh} = \frac{\text{52W High}-\text{Close}}{\text{52W High}} \times 100$$

12. `OffLow`:

$$\text{OffLow} = \frac{\text{Close}-\text{52W Low}}{\text{52W Low}} \times 100$$

13. `RS30`:

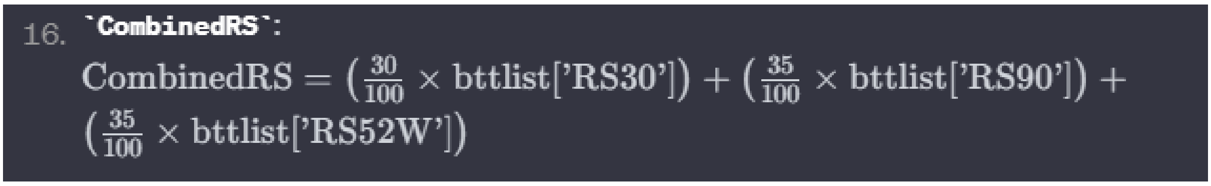
$$\text{RS30} = \left(\frac{\text{len(bttlist.index)}-\text{bttlist['RS30']}+1}{\text{len(bttlist.index)}}\right) \times 100$$

14. `RS90`:

$$\text{RS90} = \left(\frac{\text{len(bttlist.index)}-\text{bttlist['RS90']}+1}{\text{len(bttlist.index)}}\right) \times 100$$

15. `RS52W`:

$$\text{RS52W} = \left(\frac{\text{len(bttlist.index)}-\text{bttlist['RS52W']}+1}{\text{len(bttlist.index)}}\right) \times 100$$



16. **`CombinedRS`**:

$$\text{CombinedRS} = \left(\frac{30}{100} \times \text{bttlist['RS30']}\right) + \left(\frac{35}{100} \times \text{bttlist['RS90']}\right) + \left(\frac{35}{100} \times \text{bttlist['RS52W']}\right)$$