Mutual Fund Plan

Mutual Fund

Mutual funds are investment plans that pool money from multiple investors to purchase a diversified portfolio of stocks, bonds, and other securities, managed by professional fund managers. A mutual fund plan is created by selecting the stocks where an investor can benefit in the long term.

About the project

This project focuses on creating a mutual fund plan aimed at benefiting long-term investors. The process starts by collecting historical stock data, including price trends over time. Based on this information, we evaluate the performance of various stocks to identify those that offer high returns while minimizing risk. The goal is to select a balanced set of stocks that promise consistent growth and stability. Finally, we estimate the potential future value of monthly investments to help investors make informed decisions about their long-term financial goals.

Gathering Data

```
import pandas as pd
import numpy as np

df=pd.read_csv('nifty50_closing_prices.csv')

print(df.head())
```

```
                       Date  RELIANCE.NS  HDFCBANK.NS
ICICIBANK.NS  \
0  2024-08-20 00:00:00+05:30  2991.899902  1637.699951   1179.449951

1  2024-08-21 00:00:00+05:30  2997.350098  1625.800049   1174.849976

2  2024-08-22 00:00:00+05:30  2996.250000  1631.300049   1191.099976

3  2024-08-23 00:00:00+05:30  2999.949951  1625.050049   1203.500000

4  2024-08-26 00:00:00+05:30  3025.199951  1639.949951   1213.300049


       INFY.NS       TCS.NS  KOTAKBANK.NS  HINDUNILVR.NS
ITC.NS  \
0  1872.199951  4523.299805   1805.650024    2751.050049  498.799988

1  1872.699951  4551.500000   1812.949951    2791.199951  505.399994

2  1880.250000  4502.000000   1821.500000    2792.800049  504.549988
```

```
3  1862.099976  4463.899902   1818.000000    2815.600098  505.799988

4  1876.150024  4502.450195   1812.500000    2821.149902  505.700012


         LT.NS  ...  HEROMOTOCO.NS   DRREDDY.NS   SHREECEM.NS
BRITANNIA.NS  \
0  3572.699951  ...    5244.399902  6965.350098  24730.550781
5765.799805
1  3596.050049  ...    5284.700195  7062.450195  24808.050781
5837.350098
2  3606.500000  ...    5329.950195  6969.049805  25012.400391
5836.799805
3  3598.550049  ...    5384.899902  6954.500000  24706.050781
5792.649902
4  3641.899902  ...    5343.750000  6943.299805  24906.449219
5796.950195

       UPL.NS  EICHERMOT.NS   SBILIFE.NS  ADANIPORTS.NS  BAJAJ-AUTO.NS
\
0  566.150024   4883.250000  1761.300049    1492.550049    9779.700195

1  568.299988   4913.549805  1800.599976    1503.500000    9852.000000

2  579.150024   4933.549805  1795.250000    1492.300049    9914.200195

3  573.700012   4898.100098  1789.300049    1491.300049   10406.450195

4  577.450012   4875.200195  1796.250000    1482.550049   10432.549805


   HINDALCO.NS
0   672.900024
1   685.599976
2   685.549988
3   685.099976
4   711.849976

[5 rows x 51 columns]
```

```
df.columns
```

```
Index(['Date', 'RELIANCE.NS', 'HDFCBANK.NS', 'ICICIBANK.NS',
'INFY.NS',
       'TCS.NS', 'KOTAKBANK.NS', 'HINDUNILVR.NS', 'ITC.NS', 'LT.NS',
'SBIN.NS',
       'BAJFINANCE.NS', 'BHARTIARTL.NS', 'HCLTECH.NS',
'ASIANPAINT.NS',
       'AXISBANK.NS', 'DMART.NS', 'MARUTI.NS', 'ULTRACEMCO.NS',
'HDFC.NS',
       'TITAN.NS', 'SUNPHARMA.NS', 'M&M.NS', 'NESTLEIND.NS',
```

```
'WIPRO.NS',
       'ADANIGREEN.NS', 'TATASTEEL.NS', 'JSWSTEEL.NS', 'POWERGRID.NS',
       'ONGC.NS', 'NTPC.NS', 'COALINDIA.NS', 'BPCL.NS', 'IOC.NS',
'TECHM.NS',
       'INDUSINDBK.NS', 'DIVISLAB.NS', 'GRASIM.NS', 'CIPLA.NS',
       'BAJAJFINSV.NS', 'TATAMOTORS.NS', 'HEROMOTOCO.NS',
'DRREDDY.NS',
       'SHREECEM.NS', 'BRITANNIA.NS', 'UPL.NS', 'EICHERMOT.NS',
'SBILIFE.NS',
       'ADANIPORTS.NS', 'BAJAJ-AUTO.NS', 'HINDALCO.NS'],
      dtype='object')
```

```
#Convert date column into date time
df['Date']=pd.to_datetime(df['Date'])
```

```
#Look for null values
print(df.isnull().sum())
```

```
Date               0
RELIANCE.NS        0
HDFCBANK.NS        0
ICICIBANK.NS       0
INFY.NS            0
TCS.NS             0
KOTAKBANK.NS       0
HINDUNILVR.NS      0
ITC.NS             0
LT.NS              0
SBIN.NS            0
BAJFINANCE.NS      0
BHARTIARTL.NS      0
HCLTECH.NS         0
ASIANPAINT.NS      0
AXISBANK.NS        0
DMART.NS           0
MARUTI.NS          0
ULTRACEMCO.NS      0
HDFC.NS           24
TITAN.NS           0
SUNPHARMA.NS       0
M&M.NS             0
NESTLEIND.NS       0
WIPRO.NS           0
ADANIGREEN.NS      0
TATASTEEL.NS       0
JSWSTEEL.NS        0
POWERGRID.NS       0
ONGC.NS            0
NTPC.NS            0
COALINDIA.NS       0
```

```
BPCL.NS           0
IOC.NS            0
TECHM.NS          0
INDUSINDBK.NS     0
DIVISLAB.NS       0
GRASIM.NS         0
CIPLA.NS          0
BAJAJFINSV.NS     0
TATAMOTORS.NS     0
HEROMOTOCO.NS     0
DRREDDY.NS        0
SHREECEM.NS       0
BRITANNIA.NS      0
UPL.NS            0
EICHERMOT.NS      0
SBILIFE.NS        0
ADANIPORTS.NS     0
BAJAJ-AUTO.NS     0
HINDALCO.NS       0
dtype: int64
```

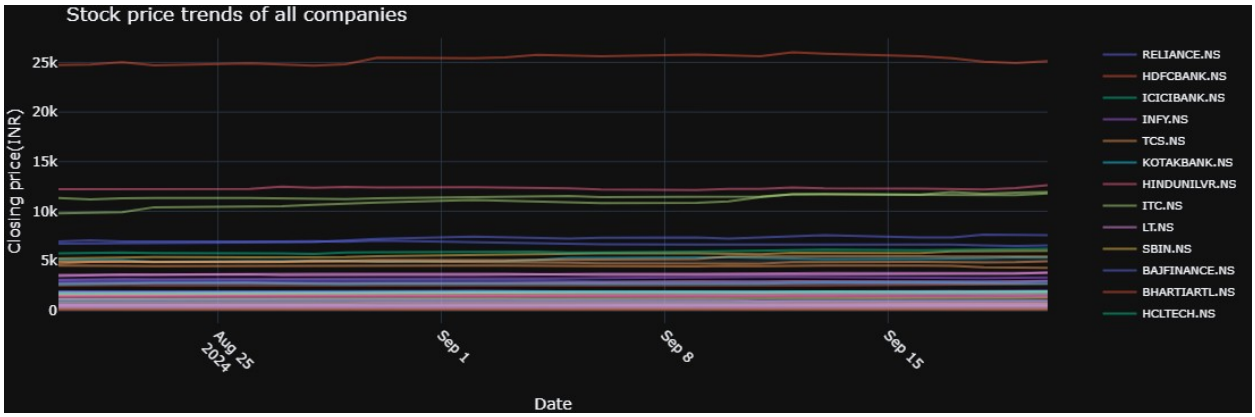
```
# We can see HDFC has 24 null values

df.ffill(inplace=True)

print(df['HDFC.NS'])
```

```
0    NaN
1    NaN
2    NaN
3    NaN
4    NaN
5    NaN
6    NaN
7    NaN
8    NaN
9    NaN
10   NaN
11   NaN
12   NaN
13   NaN
14   NaN
15   NaN
16   NaN
17   NaN
18   NaN
19   NaN
20   NaN
21   NaN
22   NaN
```

```
23   NaN
Name: HDFC.NS, dtype: float64

# All values are null
# Drop the column

df.drop(columns=['HDFC.NS'], inplace=True)

print(df.isnull().sum())

Date              0
RELIANCE.NS       0
HDFCBANK.NS       0
ICICIBANK.NS      0
INFY.NS           0
TCS.NS            0
KOTAKBANK.NS      0
HINDUNILVR.NS     0
ITC.NS            0
LT.NS             0
SBIN.NS           0
BAJFINANCE.NS     0
BHARTIARTL.NS     0
HCLTECH.NS        0
ASIANPAINT.NS     0
AXISBANK.NS       0
DMART.NS          0
MARUTI.NS         0
ULTRACEMCO.NS     0
TITAN.NS          0
SUNPHARMA.NS      0
M&M.NS            0
NESTLEIND.NS      0
WIPRO.NS          0
ADANIGREEN.NS     0
TATASTEEL.NS      0
JSWSTEEL.NS       0
POWERGRID.NS      0
ONGC.NS           0
NTPC.NS           0
COALINDIA.NS      0
BPCL.NS           0
IOC.NS            0
TECHM.NS          0
INDUSINDBK.NS     0
DIVISLAB.NS       0
GRASIM.NS         0
CIPLA.NS          0
BAJAJFINSV.NS     0
TATAMOTORS.NS     0
```

```
HEROMOTOCO.NS    0
DRREDDY.NS       0
SHREECEM.NS      0
BRITANNIA.NS     0
UPL.NS           0
EICHERMOT.NS     0
SBILIFE.NS       0
ADANIPORTS.NS    0
BAJAJ-AUTO.NS    0
HINDALCO.NS      0
dtype: int64
```

```
# Analyse Stock price trend for all companies

import plotly.graph_objs as go
import plotly.express as px

fig=go.Figure()
for company in df.columns[1:]:

fig.add_trace(go.Scatter(x=df['Date'],y=df[company],mode='lines',name=
company,opacity=0.5))
fig.update_layout(
    title='Stock price trends of all companies',
    xaxis_title='Date',
    yaxis_title='Closing price(INR)',
    xaxis=dict(tickangle=45),
    legend=dict(
        x=1.05,
        y=1,
        traceorder='normal',
        font=dict(size=10),
        orientation='v'
    ),
    margin=dict(l=0,r=0,t=30,b=0),
    hovermode='x',
    template='plotly_dark'
)
fig.show()
```




```
# Companies with highest risks

all_companies=df.columns[1:]

volitality_all_companies=df[all_companies].std()

volitality_all_companies.sort_values(ascending=False).head(10)
```

```
BAJAJ-AUTO.NS    659.810841
SHREECEM.NS      429.919834
BAJFINANCE.NS    306.658594
DIVISLAB.NS      247.674895
HEROMOTOCO.NS    247.092728
DRREDDY.NS       175.124908
ULTRACEMCO.NS    172.673053
DMART.NS         155.593701
BRITANNIA.NS     144.164343
MARUTI.NS        109.587342
dtype: float64
```

```
# Companies with high growth rate

growth_all_companies=df[all_companies].pct_change()*100

avg_growth_all_companies=growth_all_companies.mean()

avg_growth_all_companies.sort_values(ascending=False).head(10)
```

```
BAJAJ-AUTO.NS    0.883421
BAJAJFINSV.NS    0.791730
BHARTIARTL.NS    0.735219
DIVISLAB.NS      0.634851
HEROMOTOCO.NS    0.602192
ICICIBANK.NS     0.557742
BAJFINANCE.NS    0.536819
TITAN.NS         0.393800
HINDUNILVR.NS    0.351634
BRITANNIA.NS     0.327747
dtype: float64
```

```
# Companies with highest return on investment

initial_price=df[all_companies].iloc[0]
final_price=df[all_companies].iloc[-1]
roi_all_companies=((final_price-initial_price)/initial_price)*100
roi_all_companies.sort_values(ascending=False).head(10)
```

```
BAJAJ-AUTO.NS    22.107017
BAJAJFINSV.NS    19.642973
BHARTIARTL.NS    18.120965
DIVISLAB.NS      15.404976
HEROMOTOCO.NS    14.660402
ICICIBANK.NS     13.480860
BAJFINANCE.NS    12.797149
TITAN.NS          9.275089
HINDUNILVR.NS     8.235039
BRITANNIA.NS      7.713587
dtype: float64
```

Creating mutual fund plan based on high ROI and low risk

```
roi_threshold=roi_all_companies.median()

volatility_threshold=volitality_all_companies.median()

selected_companies=roi_all_companies[(roi_all_companies>roi_threshold)
& (volitality_all_companies<volatility_threshold)]

selected_companies.sort_values(ascending=False)
```

```
ICICIBANK.NS     13.480860
INDUSINDBK.NS     7.159914
JSWSTEEL.NS       7.021748
AXISBANK.NS       6.592466
HDFCBANK.NS       6.319839
SUNPHARMA.NS      5.627425
KOTAKBANK.NS      5.474481
CIPLA.NS          4.850117
NTPC.NS           4.356926
dtype: float64
```

For allocation we will use inverse volatility ratio, Companies with lower volatility will get higher allocation

```
selected_volatility=volitality_all_companies[selected_companies.index]

inverse_volatility=1/selected_volatility

investment_ratio=(inverse_volatility/inverse_volatility.sum())*100
```

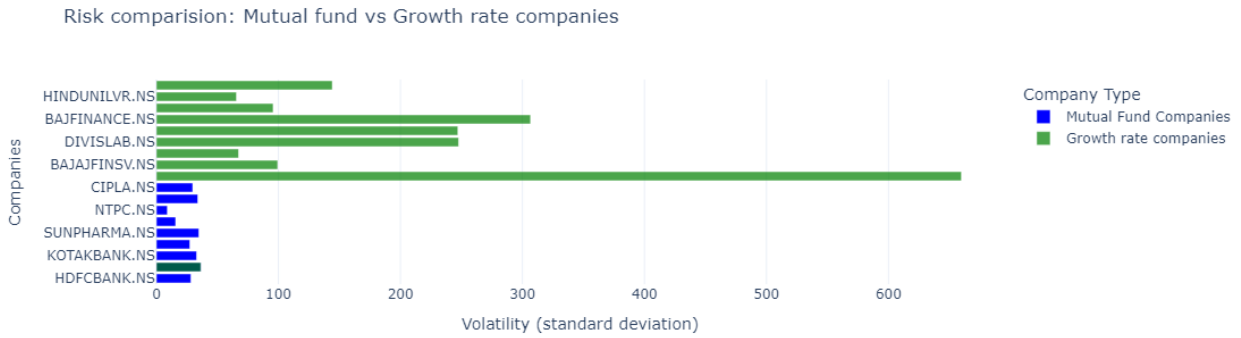
```
investment_ratio.sort_values(ascending=False)
```

```
NTPC.NS          28.076767
JSWSTEEL.NS      15.998503
AXISBANK.NS       9.223133
HDFCBANK.NS       8.933035
CIPLA.NS          8.478347
KOTAKBANK.NS      7.664235
INDUSINDBK.NS     7.443153
SUNPHARMA.NS      7.255261
ICICIBANK.NS      6.927566
dtype: float64
```

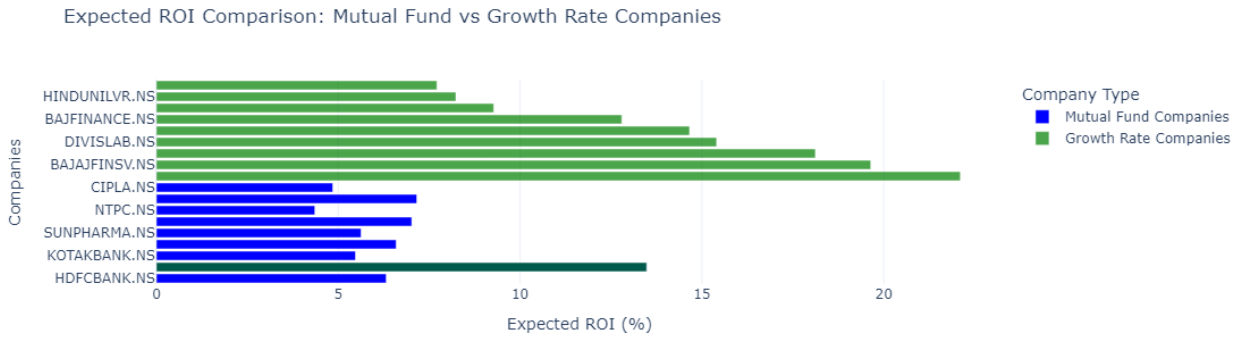
Analyze Mutual Fund Plan

```
#comparing our mutual fund plan by comparing with high performance
companies in stock market
#compare the risk of our mutual fund plan and high growth companies

top_growth_companies=avg_growth_all_companies.sort_values(ascending=Fa
lse).head(10)
risk_growth_rate_companies=volitality_all_companies[top_growth_compani
es.index]
risk_mutual_fund_companies=volitality_all_companies[selected_companies
.index]
fig=go.Figure()
fig.add_trace(go.Bar(
    y=risk_mutual_fund_companies.index,
    x=risk_mutual_fund_companies,
    orientation='h',
    name='Mutual Fund Companies',
    marker=dict(color='blue')
))
fig.add_trace(go.Bar(
    y=risk_growth_rate_companies.index,
    x=risk_growth_rate_companies,
    orientation='h',
    name='Growth rate companies',
    marker=dict(color='green'),
    opacity=0.7
))
fig.update_layout(
    title='Risk comparision: Mutual fund vs Growth rate companies',
    xaxis_title='Volatility (standard deviation)',
    yaxis_title='Companies',
    barmode='overlay',
    legend=dict(title='Company Type'),
    template='plotly_white'
```

```
)
fig.show()
```




```
#Comparing returns of mutual fund and Growth rate companies

expected_roi_mutual_fund=roi_all_companies[selected_companies.index]
expected_roi_growth_companies=roi_all_companies[top_growth_companies.i
ndex]
fig = go.Figure()
fig.add_trace(go.Bar(
    y=expected_roi_mutual_fund.index,
    x=expected_roi_mutual_fund,
    orientation='h',
    name='Mutual Fund Companies',
    marker=dict(color='blue')
))
fig.add_trace(go.Bar(
    y=expected_roi_growth_companies.index,
    x=expected_roi_growth_companies,
    orientation='h',
    name='Growth Rate Companies',
    marker=dict(color='green'),
    opacity=0.7
))
fig.update_layout(
    title='Expected ROI Comparison: Mutual Fund vs Growth Rate
Companies',
    xaxis_title='Expected ROI (%)',
    yaxis_title='Companies',
    barmode='overlay',
    legend=dict(title='Company Type'),
    template='plotly_white'
)
fig.show()
```



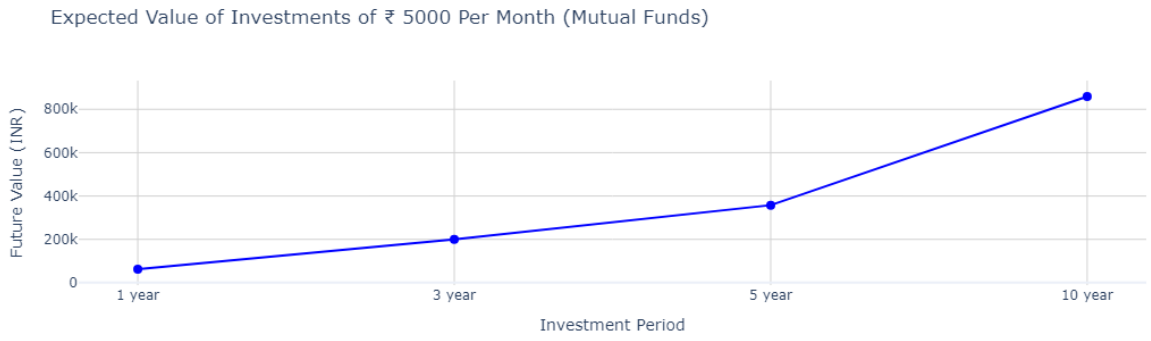
The comparison between the risk (volatility) and expected ROI for mutual fund companies (in blue) and growth rate companies (in green) shows a clear trade-off. Mutual fund companies offer lower volatility, meaning they are less risky, but also provide lower expected returns. In contrast, growth rate companies demonstrate higher volatility, indicating more risk, but they offer much higher potential returns, especially companies like Bajaj Auto and Bajaj Finserv. This highlights a common investment dilemma: lower risk comes with a lower reward, while higher risk could yield higher returns.

For long-term investments, the goal is typically to find companies that offer a balance of stable returns and manageable risk. The companies in our mutual fund exhibit low volatility, meaning they are less risky, and their moderate returns make them solid choices for long-term, stable growth. They are well-suited for conservative investors who want steady returns without significant fluctuations in value.

Calculating expected return

```
#Lets calculate expected return for a person investing 5000 rupees per
month over a period of 1 yr,3 yr,5 yr,10 yr

monthly_investment=5000
years=[1,3,5,10]
n=12 #interest compounding times
avg_roi=expected_roi_mutual_fund.mean()/100
def future_value(P,r,n,t):
    return P*(((1+r/n)**(n*t)-1)/(r/n))*(1+r/n)
future_values=[future_value(monthly_investment,avg_roi,n,t) for t in
years]
fig=go.Figure()
fig.add_trace(go.Scatter(
    x=[str(year)+" year" for year in years],
    y=future_values,
    mode='lines+markers',
    line=dict(color='blue'),
    marker=dict(size=8),
    name='Future value'
))
fig.update_layout(
```

```
    title="Expected Value of Investments of ₹ 5000 Per Month (Mutual
Funds)",
    xaxis_title="Investment Period",
    yaxis_title="Future Value (INR)",
    xaxis=dict(showgrid=True, gridcolor='lightgrey'),
    yaxis=dict(showgrid=True, gridcolor='lightgrey'),
    template="plotly_white",
    hovermode='x'
)
fig.show()
```



Summary

So, this is how a mutual fund plan is designed by investment companies for long-term investors. Mutual funds are investment plans that pool money from multiple investors to purchase a diversified portfolio of stocks, bonds, and other securities, managed by professional fund managers.