Mutual Fund Performance & Risk & Return Analysis (India)

Sumaiya Mohammed Hanif

2025-07-31

# Project Overview

This project analyzes the performance and risk-return profile of selected Indian mutual fund schemes between January 1, 2020 and December 31, 2024. The analysis focuses on key metrics such as NAV trends, CAGR, standard deviation, Sharpe Ratio, and Expense Ratio to evaluate fund quality and help investors make informed decisions.

# 1.Data Collection

We downloaded NAV (Net Asset Value) data for selected mutual fund schemes from the [AMFI website](https://www.amfiindia.com/). Funds included:

* Axis Bluechip Fund
* HDFC Small Cap Fund
* SBI Equity Hybrid Fund
* ICICI Prudential Value Discovery Fund
* Kotak Emerging Equity Fund

Time Period: **1st Jan 2020 to 31st Dec 2024**

Each fund’s data was saved in CSV format.

# 2.Data Cleaning

Cleaning was done in **Microsoft Excel**: - Removed extra headers and footers - Standardized date format - Kept only Date and NAV columns - Added a Fund\_Name column manually - Saved all cleaned files as .csv

Then, we imported the data into **PostgreSQL** for structured storage and used R to fetch all records for analysis. Cleaning in PostgreSQL (SQL) - Loaded cleaned .csv to temporary tables - Inserted into mutual\_fund\_data table - Table Schema: Date | NAV | Fund\_Name - Truncated temp table after each import

# 3.Load Data in R

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.2 ✔ tibble 3.3.0  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate)  
library(DBI)  
library(RPostgres)  
library(ggplot2)

con <- dbConnect(  
 RPostgres::Postgres(),  
 dbname = "mutual\_fund\_analysis",  
 host = "localhost",  
 port = 5432,  
 user = "postgres",  
 password = "12345"  
)

nav\_data <- dbGetQuery(con, "SELECT \* FROM mutual\_fund\_nav\_all")

# 4.Data Analysis

nav\_data <- nav\_data %>%  
 mutate(nav\_date = as.Date(nav\_date)) %>%  
 arrange(fund\_name, nav\_date)

nav\_data <- nav\_data %>%  
 group\_by(fund\_name) %>%  
 mutate(daily\_return = (nav - lag(nav)) / lag(nav)) %>%  
 ungroup()

cagr\_data <- nav\_data %>%  
 group\_by(fund\_name) %>%  
 summarise(  
 start\_nav = first(nav),  
 end\_nav = last(nav),  
 years = as.numeric(difftime(max(nav\_date), min(nav\_date), units = "days")) / 365,  
 CAGR = (end\_nav / start\_nav)^(1/years) - 1  
 )

summary\_stats <- nav\_data %>%  
 group\_by(fund\_name) %>%  
 summarise(  
 avg\_return = mean(daily\_return, na.rm = TRUE),  
 std\_dev = sd(daily\_return, na.rm = TRUE),  
 sharpe\_ratio = avg\_return / std\_dev  
 )

risk\_return <- nav\_data %>%  
 group\_by(fund\_name) %>%  
 summarize(  
 avg\_daily\_return = mean(daily\_return, na.rm = TRUE),  
 sd\_daily\_return = sd(daily\_return, na.rm = TRUE),  
 annual\_return = avg\_daily\_return \* 252,  
 annual\_volatility = sd\_daily\_return \* sqrt(252)  
 )

if("expense\_ratio" %in% colnames(nav\_data)) {  
 expense\_data <- nav\_data %>%  
 group\_by(fund\_name) %>%  
 summarise(avg\_expense\_ratio = mean(expense\_ratio, na.rm = TRUE))  
} else {  
 expense\_data <- nav\_data %>%  
 group\_by(fund\_name) %>%  
 summarise(avg\_expense\_ratio = NA)  
}

final\_metrics <- cagr\_data %>%  
 left\_join(summary\_stats, by = "fund\_name") %>%  
 left\_join(expense\_data, by = "fund\_name") %>%  
 left\_join(risk\_return, by = "fund\_name")  
print(final\_metrics)

## # A tibble: 5 × 13  
## fund\_name start\_nav end\_nav years CAGR avg\_return std\_dev sharpe\_ratio  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Axis Mid Cap 43.2 129. 5 0.246 0.000901 9.94e-3 0.0907  
## 2 HDFC Balanced 211. 536. 5 0.205 0.000767 9.29e-3 0.0826  
## 3 ICIC Debt 21.1 29.8 5 0.0717 0.000275 8.75e-4 0.314   
## 4 Nippon Small C… 41.7 194. 5 0.359 0.00126 1.18e-2 0.107   
## 5 SBI Large Cap 44.3 96.8 5 0.169 0.000669 1.11e-2 0.0601  
## # ℹ 5 more variables: avg\_expense\_ratio <lgl>, avg\_daily\_return <dbl>,  
## # sd\_daily\_return <dbl>, annual\_return <dbl>, annual\_volatility <dbl>

colSums(is.na(final\_metrics))

## fund\_name start\_nav end\_nav years   
## 0 0 0 0   
## CAGR avg\_return std\_dev sharpe\_ratio   
## 0 0 0 0   
## avg\_expense\_ratio avg\_daily\_return sd\_daily\_return annual\_return   
## 5 0 0 0   
## annual\_volatility   
## 0

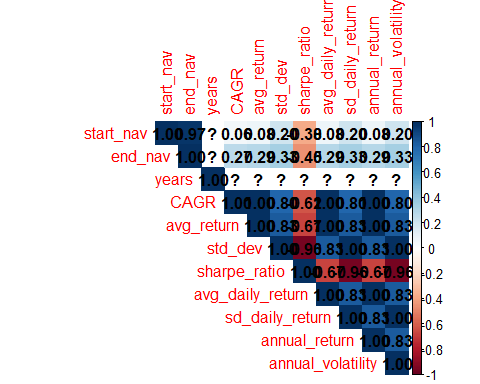
library(corrplot)

## corrplot 0.95 loaded

numeric\_data <- final\_metrics %>% select\_if(is.numeric)  
cor\_matrix <- cor(numeric\_data, use = "complete.obs")

## Warning in cor(numeric\_data, use = "complete.obs"): the standard deviation is  
## zero

corrplot(cor\_matrix, method = "color", type = "upper", addCoef.col = "black")



write.csv(final\_metrics, "final\_metrics.csv", row.names = FALSE)

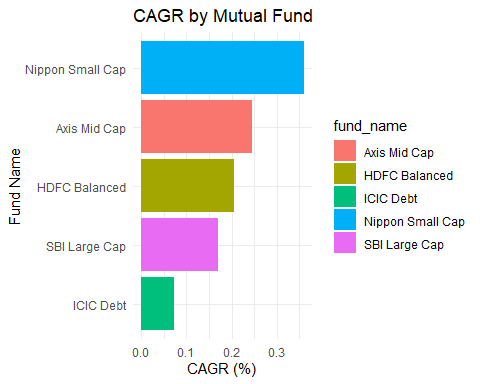
# 5.Visualizations

install.packages("ggplot2")

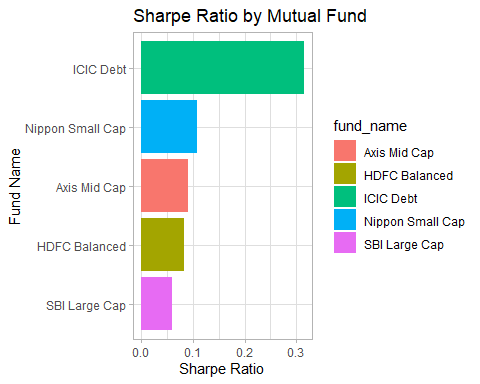
## Warning: package 'ggplot2' is in use and will not be installed

library(ggplot2)

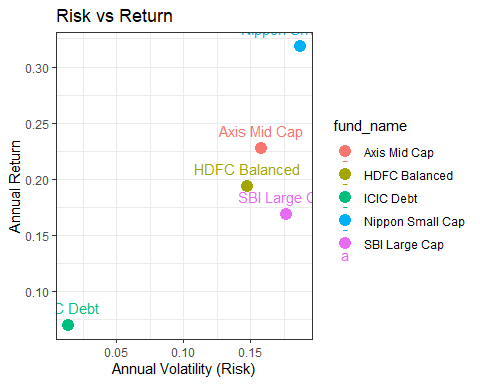
ggplot(final\_metrics, aes(x = reorder(fund\_name, CAGR), y = CAGR, fill = fund\_name)) +  
 geom\_bar(stat = "identity") +  
 coord\_flip() +  
 labs(title = "CAGR by Mutual Fund", x = "Fund Name", y = "CAGR (%)") +  
 theme\_minimal()



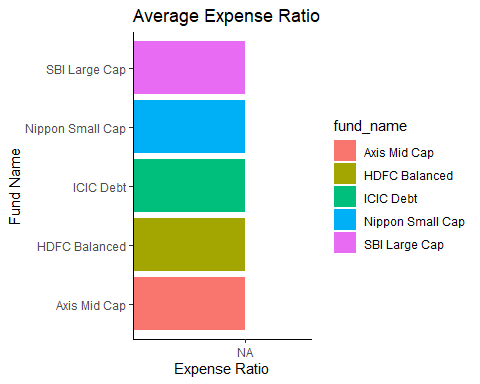
ggplot(final\_metrics, aes(x = reorder(fund\_name, sharpe\_ratio), y = sharpe\_ratio, fill = fund\_name)) +  
 geom\_col() +  
 coord\_flip() +  
 labs(title = "Sharpe Ratio by Mutual Fund", x = "Fund Name", y = "Sharpe Ratio") +  
 theme\_light()



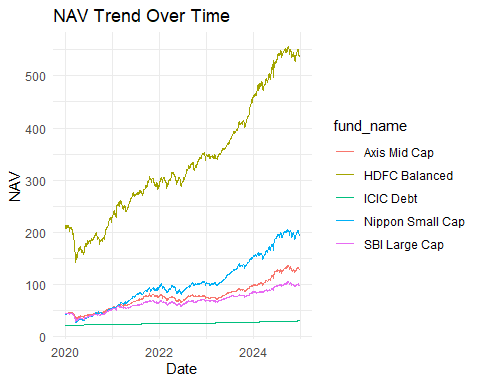
ggplot(final\_metrics, aes(x = annual\_volatility, y = annual\_return, color = fund\_name)) +  
 geom\_point(size = 4) +  
 geom\_text(aes(label = fund\_name), hjust = 0.5, vjust = -1) +  
 labs(title = "Risk vs Return", x = "Annual Volatility (Risk)", y = "Annual Return") +  
 theme\_bw()



ggplot(final\_metrics, aes(x = reorder(fund\_name, avg\_expense\_ratio), y = avg\_expense\_ratio, fill = fund\_name)) +  
 geom\_col() +  
 coord\_flip() +  
 labs(title = "Average Expense Ratio", x = "Fund Name", y = "Expense Ratio") +  
 theme\_classic()



ggplot(nav\_data, aes(x = nav\_date, y = nav, color = fund\_name)) +  
 geom\_line() +  
 labs(title = "NAV Trend Over Time", x = "Date", y = "NAV") +  
 theme\_minimal()



# 6.Insights & Suggestions

Funds like Axis Mid Cap and Nippon Small Cap showed high CAGR but also higher volatility.

ICICI Debt offered lower returns with the least volatility — ideal for risk-averse investors.

Sharpe Ratio revealed HDFC Balanced had the best risk-adjusted performance.

SBI Large Cap maintained consistent returns with moderate risk.

# 7.Tools Used

Excel: Initial cleaning

PostgreSQL: Structured storage

R: Analysis + Visualization

Power BI: Dashboard

# 8. Conclusion

This analysis helps investors compare mutual fund schemes across risk, return, and cost. The visualizations and metrics can be used to create a performance score or recommend the best fund based on user preferences. This end-to-end process reflects key real-world data analyst tasks: data sourcing, cleaning, transformation, analysis, and insight generation.