

One-Week U.S. Equity Prediction Trading System Analysis

Thomas Benacci

2025-01-13

1. Introduction

This analysis evaluates a Random Forest-based trading system designed to predict holiday returns. The model selects stocks predicted to outperform the SPY benchmark, forming a concentrated long-only portfolio. We compare performance against both the equal-weighted stock universe and SPY.

2. Setup

```
library(tidyverse)
library(lubridate)
library(zoo)
library(data.table)
library(scales)
```

3. Data

We merge the feature dataset with the Random Forest (trees = 2000, max depth = 7, splitting criterion = entropy) model predictions. The analysis covers 2009–2024, corresponding to the out-of-sample test period.

```
# Load feature dataset
feature_data <- fread("feature_data_holiday_20251112.csv", data.table = FALSE) %>%
  distinct(datadate, tic, .keep_all = TRUE) %>%
  mutate(model = "RF1")
# Load RF model predictions
results_data <- fread("rf_holiday_results_01132026.csv", data.table = FALSE) %>%
  select(datadate, tic, prediction)
# merge and clean
data <- merge(results_data, feature_data, all.x = TRUE) %>%
  filter(datadate <= "2025-01-01") %>%
  rename(spy_return = spy_holiday_return, return = holiday_return)
rm(feature_data, results_data)

cat("Observations:", nrow(data), "\n")
```

```
## Observations: 406500
```

```
cat("Date range:", as.character(min(data$datadate)), "to", as.character(max(data$datadate)),
"\n")
```

```

## Date range: 2009-01-08 to 2024-12-26

cat("Unique tickers:", length(unique(data$tic)), "\n")

## Unique tickers: 1013

# Load risk-free rate for potential Sharpe calculations
fedfunds <- fread("FEDFUNDS.csv", data.table = FALSE) %>%
  mutate(
    FEDFUNDS = FEDFUNDS / 100,
    rf_weekly = (1 + FEDFUNDS)^(1/52) - 1,
    rf_daily = (1 + FEDFUNDS)^(1/252) - 1
  )

```

4. Exploratory Data Analysis

4.1 Excess Return Distribution

Excess returns are calculated as stock return minus SPY return for each period. A distribution centered at zero indicates no systematic edge from random selection.

```

excess_returns <- data$return - data$spy_return

# summary statistics
data.frame(
  Statistic = c("Mean", "Median", "Std Dev", "Min", "Max"),
  Value = round(c(
    mean(excess_returns, na.rm = TRUE),
    median(excess_returns, na.rm = TRUE),
    sd(excess_returns, na.rm = TRUE),
    min(excess_returns, na.rm = TRUE),
    max(excess_returns, na.rm = TRUE)
  ), 4)
)

```

Statistic	Value
1 Mean	-0.0005
2 Median	-0.0008
3 Std Dev	0.0373
4 Min	-0.8742
5 Max	2.2742

```

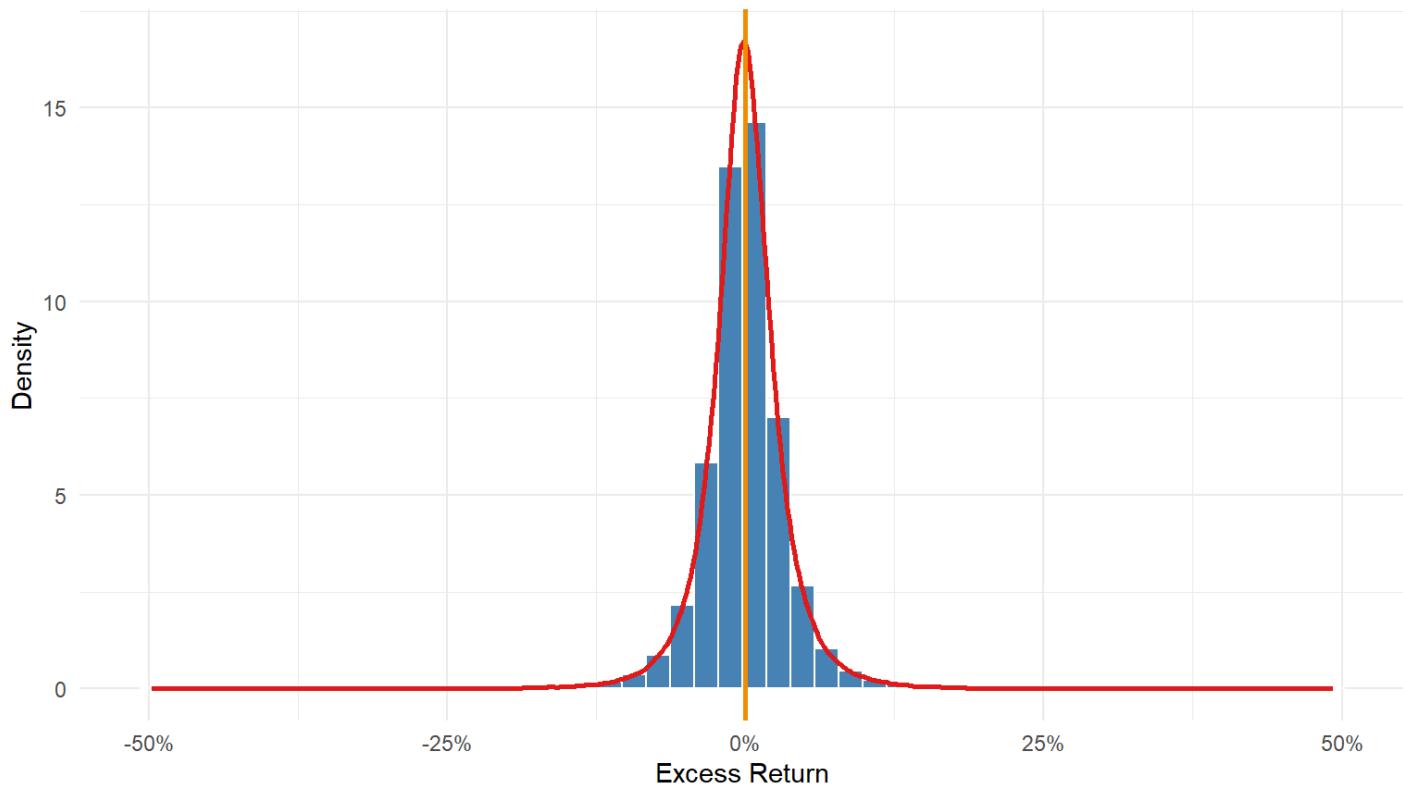
excess_trimmed <- excess_returns[excess_returns > -0.5 & excess_returns < 0.5]

ggplot(data.frame(x = excess_trimmed), aes(x = x)) +
  geom_histogram(aes(y = after_stat(density)), bins = 50, fill = "steelblue", color = "white") +
  geom_density(color = "#E41A1C", linewidth = 1) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "gray40") +
  geom_vline(xintercept = mean(excess_trimmed), color = "#F18F01", linewidth = 1) +
  labs(
    title = "Distribution of Excess Returns (Stock vs. SPY)",
    subtitle = "Trimmed to ±50% | Orange line = mean",
    x = "Excess Return", y = "Density"
  ) +
  theme_minimal() +
  scale_x_continuous(labels = percent_format())

```

Distribution of Excess Returns (Stock vs. SPY)

Trimmed to $\pm 50\%$ | Orange line = mean



The distribution centers near zero, consistent with weak-form market efficiency. Any edge must come from the prediction model, not the universe itself.

4.2 Target Variable Over Time

The target is binary: 1 if the stock beats SPY, 0 otherwise. A baseline of ~50% means random selection has no edge.

```
cat("Overall target mean:", round(mean(data$target), 3), "\n")
```

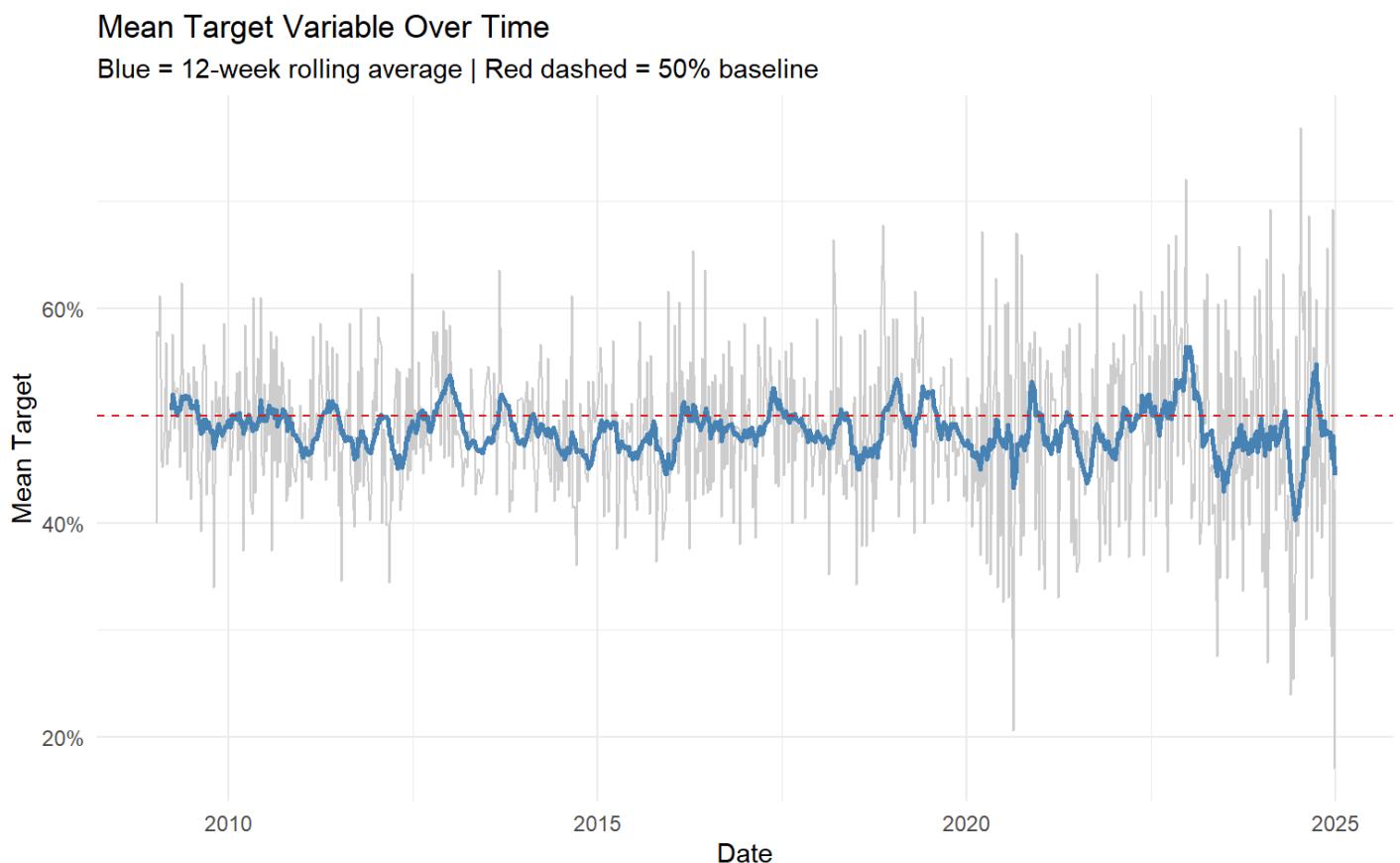
```
## Overall target mean: 0.486
```

```

df_target <- data %>%
  group_by(datadate) %>%
  summarize(mean_target = mean(target), .groups = "drop") %>%
  mutate(
    datadate = as.Date(datadate),
    rolling_mean = zoo::rollmean(mean_target, k = 12, fill = NA, align = "right")
  )

ggplot(df_target, aes(x = datadate)) +
  geom_line(aes(y = mean_target), color = "gray70", alpha = 0.6) +
  geom_line(aes(y = rolling_mean), color = "steelblue", linewidth = 1) +
  geom_hline(yintercept = 0.5, linetype = "dashed", color = "#E41A1C") +
  labs(
    title = "Mean Target Variable Over Time",
    subtitle = "Blue = 12-week rolling average | Red dashed = 50% baseline",
    x = "Date", y = "Mean Target"
  ) +
  theme_minimal() +
  scale_y_continuous(labels = percent_format())

```



Post-2020 shows higher dispersion, suggesting a regime change—possibly due to increased SPY concentration in mega-cap tech.

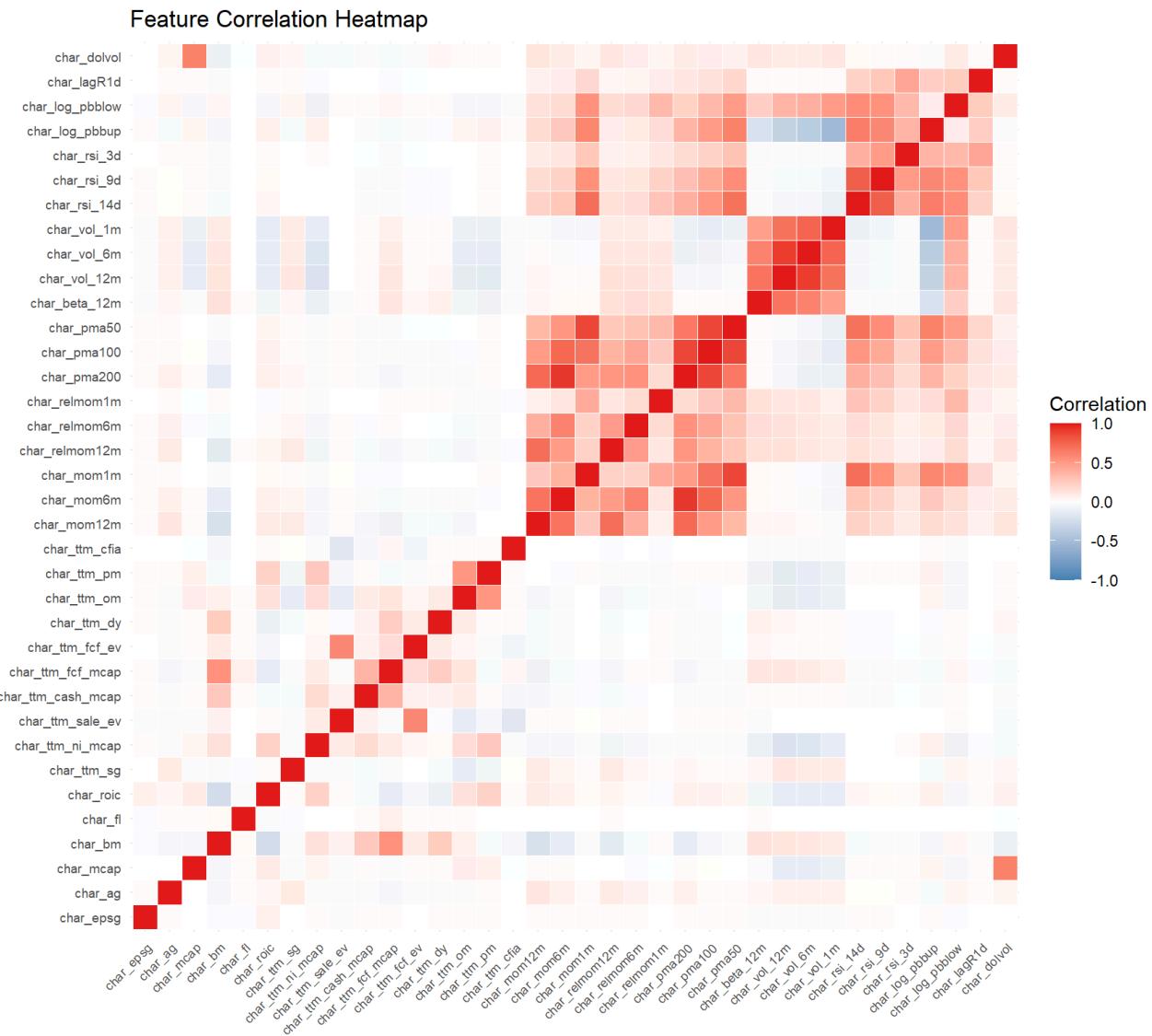
5. Feature Analysis

5.1 Feature Correlation Heatmap

```
feature_cor_mat <- data %>%
  select(starts_with("char_")) %>%
  cor(use = "pairwise.complete.obs")

cor_long <- as.data.frame(as.table(feature_cor_mat)) %>%
  rename(Feature1 = Var1, Feature2 = Var2, Correlation = Freq)

ggplot(cor_long, aes(x = Feature1, y = Feature2, fill = Correlation)) +
  geom_tile(color = "white", linewidth = 0.1) +
  scale_fill_gradient2(low = "steelblue", mid = "white", high = "#E41A1C", midpoint = 0, limits
= c(-1, 1)) +
  labs(title = "Feature Correlation Heatmap", x = NULL, y = NULL) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 7),
        axis.text.y = element_text(size = 7)) +
  coord_fixed()
```



Technical indicators show high inter-correlation; fundamentals are largely orthogonal. This suggests potential for dimensionality reduction.

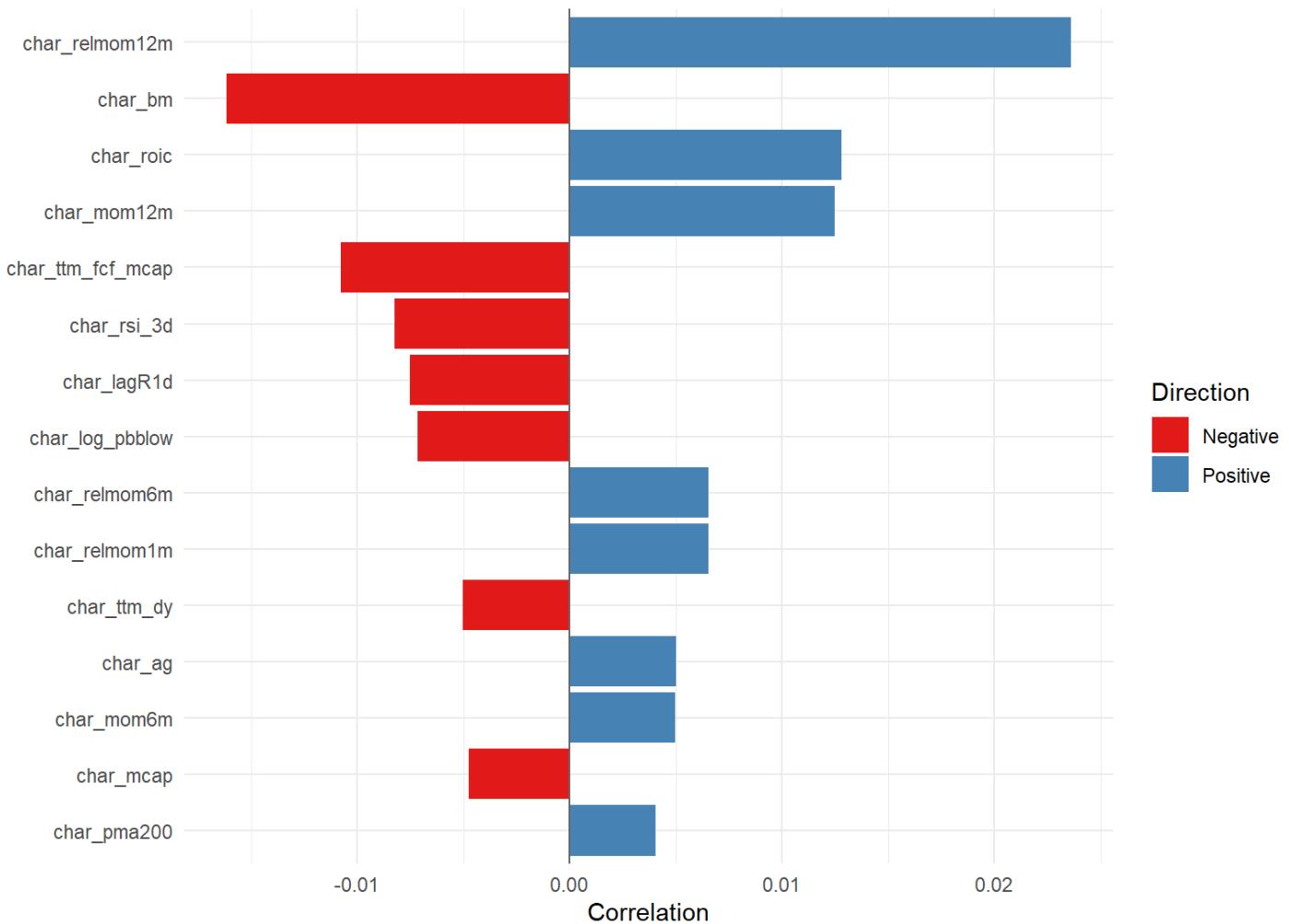
5.2 Feature-Target Correlations

```
target_cors <- sapply(data[grep("char_", names(data))], function(x) cor(data$target, x, use = "complete.obs"))

cor_df <- data.frame(Feature = names(target_cors), Correlation = as.numeric(target_cors)) %>%
  arrange(desc(abs(Correlation))) %>%
  head(15) %>%
  mutate(Direction = ifelse(Correlation > 0, "Positive", "Negative"))

ggplot(cor_df, aes(x = reorder(Feature, abs(Correlation)), y = Correlation, fill = Direction)) +
  geom_col() +
  coord_flip() +
  scale_fill_manual(values = c("Positive" = "steelblue", "Negative" = "#E41A1C")) +
  geom_hline(yintercept = 0, color = "gray40") +
  labs(title = "Top 15 Features by Correlation with Target", x = NULL, y = "Correlation") +
  theme_minimal()
```

Top 15 Features by Correlation with Target



Technical indicators dominate the top predictors, reflecting the short-term (weekly) trading horizon.

5.3 Feature Importance Summary Statistics

```
# full correlation table sorted by absolute value
feature_cors <- sapply(data[grep("char_", names(data))], function(x) cor(data$target, x, use = "complete.obs"))

importance_df <- data.frame(
  Feature = names(feature_cors),
  Correlation = as.numeric(feature_cors)
) %>%
  mutate(
  Abs_Corr = abs(Correlation),
  Direction = ifelse(Correlation > 0, "Positive", "Negative"),
  Rank = rank(-Abs_Corr)
) %>%
  arrange(Rank)

# summary of correlations
cat("== Feature-Target Correlation Summary ==\n")
```

```
## === Feature-Target Correlation Summary ===
```

```
cat("Total features:", nrow(importance_df), "\n")
```

```
## Total features: 36
```

```
cat("Positive correlations:", sum(importance_df$Correlation > 0), "\n")
```

```
## Positive correlations: 17
```

```
cat("Negative correlations:", sum(importance_df$Correlation < 0), "\n")
```

```
## Negative correlations: 19
```

```
cat("Mean |correlation|:", round(mean(importance_df$Abs_Corr), 4), "\n")
```

```
## Mean |correlation|: 0.005
```

```
cat("Max |correlation|:", round(max(importance_df$Abs_Corr), 4), "\n")
```

```
## Max |correlation|: 0.0236
```

```
cat("Features with |corr| > 0.05:", sum(importance_df$Abs_Corr > 0.05), "\n")
```

```
## Features with |corr| > 0.05: 0
```

```
cat("Features with |corr| > 0.02:", sum(importance_df$Abs_Corr > 0.02), "\n")
```

```
## Features with |corr| > 0.02: 1
```

```
# full ranked table
importance_df %>%
  select(Rank, Feature, Correlation, Direction) %>%
  mutate(Correlation = round(Correlation, 4))
```

##	Rank	Feature	Correlation	Direction
## 1	1	char_relmom12m	0.0236	Positive
## 2	2	char_bm	-0.0162	Negative
## 3	3	char_roic	0.0128	Positive
## 4	4	char_mom12m	0.0125	Positive
## 5	5	char_ttm_fcf_mcav	-0.0108	Negative
## 6	6	char_rsi_3d	-0.0082	Negative
## 7	7	char_lagR1d	-0.0075	Negative
## 8	8	char_log_pbblow	-0.0072	Negative
## 9	9	char_relmom6m	0.0065	Positive
## 10	10	char_relmom1m	0.0065	Positive
## 11	11	char_ttm_dy	-0.0050	Negative
## 12	12	char_ag	0.0050	Positive
## 13	13	char_mom6m	0.0050	Positive
## 14	14	char_mcav	-0.0048	Negative
## 15	15	char_pma200	0.0040	Positive
## 16	16	char_beta_12m	0.0040	Positive
## 17	17	char_ttm_cash_mcav	-0.0040	Negative
## 18	18	char_vol_1m	-0.0036	Negative
## 19	19	char_mom1m	-0.0035	Negative
## 20	20	char_pma50	-0.0027	Negative
## 21	21	char_ttm_sale_ev	0.0026	Positive
## 22	22	char_ttm_fcf_ev	-0.0024	Negative
## 23	23	char_ttm_om	-0.0024	Negative
## 24	24	char_epsg	0.0024	Positive
## 25	25	char_dolvol	0.0023	Positive
## 26	26	char_ttm_pm	0.0022	Positive
## 27	27	char_ttm_sg	0.0021	Positive
## 28	28	char_ttm_ni_mcav	-0.0018	Negative
## 29	29	char_vol_6m	-0.0017	Negative
## 30	30	char_rsi_9d	-0.0017	Negative
## 31	31	char_log_pbbup	-0.0012	Negative
## 32	32	char_rsi_14d	0.0012	Positive
## 33	33	char_vol_12m	-0.0011	Negative
## 34	34	char_ttm_cfia	-0.0002	Negative
## 35	35	char_pma100	0.0000	Positive
## 36	36	char_f1	0.0000	Positive

5.4 Technical vs Fundamental Features

```
# categorize features based on the data_prep naming conventions
# Technical: mom, relmom, pma, beta, vol, rsi, pbb, LagR, dolvol
# Fundamental: mcap, bm, fl, roic, epsg, ag, ttm_*
importance_df <- importance_df %>%
  mutate(
    Category = ifelse(
      grepl("mom|relmom|pma|beta|vol|rsi|pbb|lagR|dolvol", Feature, ignore.case = TRUE),
      "Technical",
      "Fundamental"
    )
  )

category_summary <- importance_df %>%
  group_by(Category) %>%
  summarise(
    N_Features = n(),
    Mean_Abs_Corr = round(mean(Abs_Corr), 4),
    Max_Abs_Corr = round(max(Abs_Corr), 4),
    Pct_Positive = round(mean(Correlation > 0) * 100, 1),
    .groups = "drop"
  ) %>%
  arrange(desc(Mean_Abs_Corr))

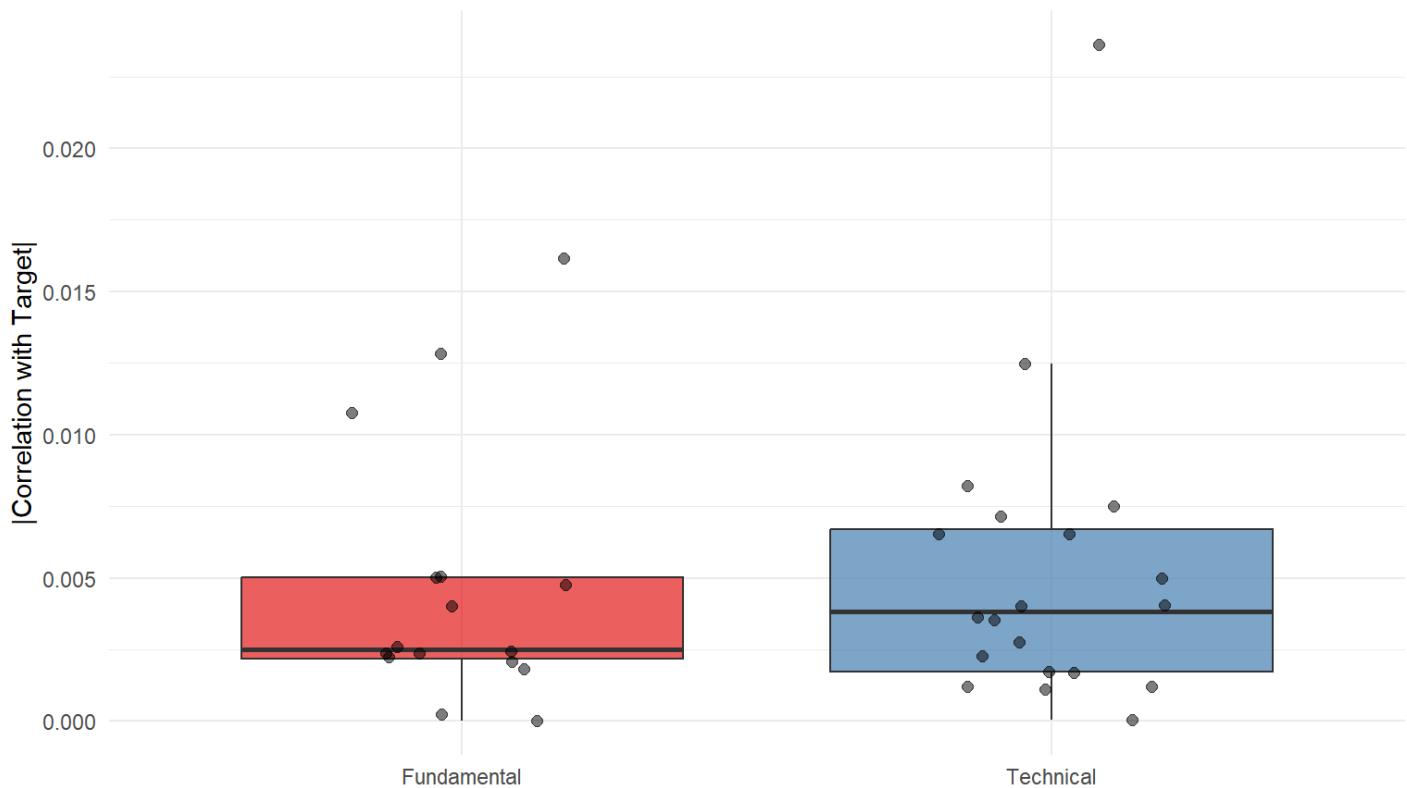
category_summary
```

```
## # A tibble: 2 × 5
##   Category     N_Features Mean_Abs_Corr Max_Abs_Corr Pct_Positive
##   <chr>           <int>        <dbl>        <dbl>        <dbl>
## 1 Technical       20        0.0052       0.0236       50
## 2 Fundamental     16        0.0047       0.0162      43.8
```

```
ggplot(importance_df, aes(x = Category, y = Abs_Corr, fill = Category)) +
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.5, size = 2) +
  scale_fill_manual(values = c("Technical" = "steelblue", "Fundamental" = "#E41A1C")) +
  labs(
    title = "Feature Importance: Technical vs Fundamental",
    subtitle = "Distribution of |correlation| with target",
    x = NULL, y = "|Correlation with Target|"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```

Feature Importance: Technical vs Fundamental

Distribution of |correlation| with target



```
# top 5 features in each category
importance_df %>%
  group_by(Category) %>%
  slice_head(n = 5) %>%
  select(Category, Rank, Feature, Correlation) %>%
  mutate(Correlation = round(Correlation, 4)) %>%
  arrange(Rank)
```

```
## # A tibble: 10 × 4
## # Groups:   Category [2]
##   Category     Rank Feature       Correlation
##   <chr>     <dbl> <chr>           <dbl>
## 1 Technical     1 char_relmom12m    0.0236
## 2 Fundamental   2 char_bm        -0.0162
## 3 Fundamental   3 char_roic      0.0128
## 4 Technical     4 char_mom12m    0.0125
## 5 Fundamental   5 char_ttm_fcf_mc cap -0.0108
## 6 Technical     6 char_rsi_3d     -0.0082
## 7 Technical     7 char_lagR1d     -0.0075
## 8 Technical     8 char_log_pbb low -0.0072
## 9 Fundamental   11 char_ttm_dy     -0.005
## 10 Fundamental  12 char_ag        0.005
```

Technical indicators show higher average predictive power than fundamental factors, consistent with the short-term trading horizon.

6. Portfolio Construction

6.1 Parameters

```
n_long <- 20 # number of long positions
prob_weighted <- TRUE # weight by prediction probability
periods_per_year <- 252 / 5 # weekly trading periods
# date-to-year mapping for labels
unique_dates <- unique(data$datadate)
year_labels <- seq(2009, 2025, length.out = length(unique_dates))
date_to_year <- setNames(year_labels, unique_dates)
```

6.2 RF Portfolio Formation

We select the top N stocks by predicted probability each period and form a probability-weighted portfolio.

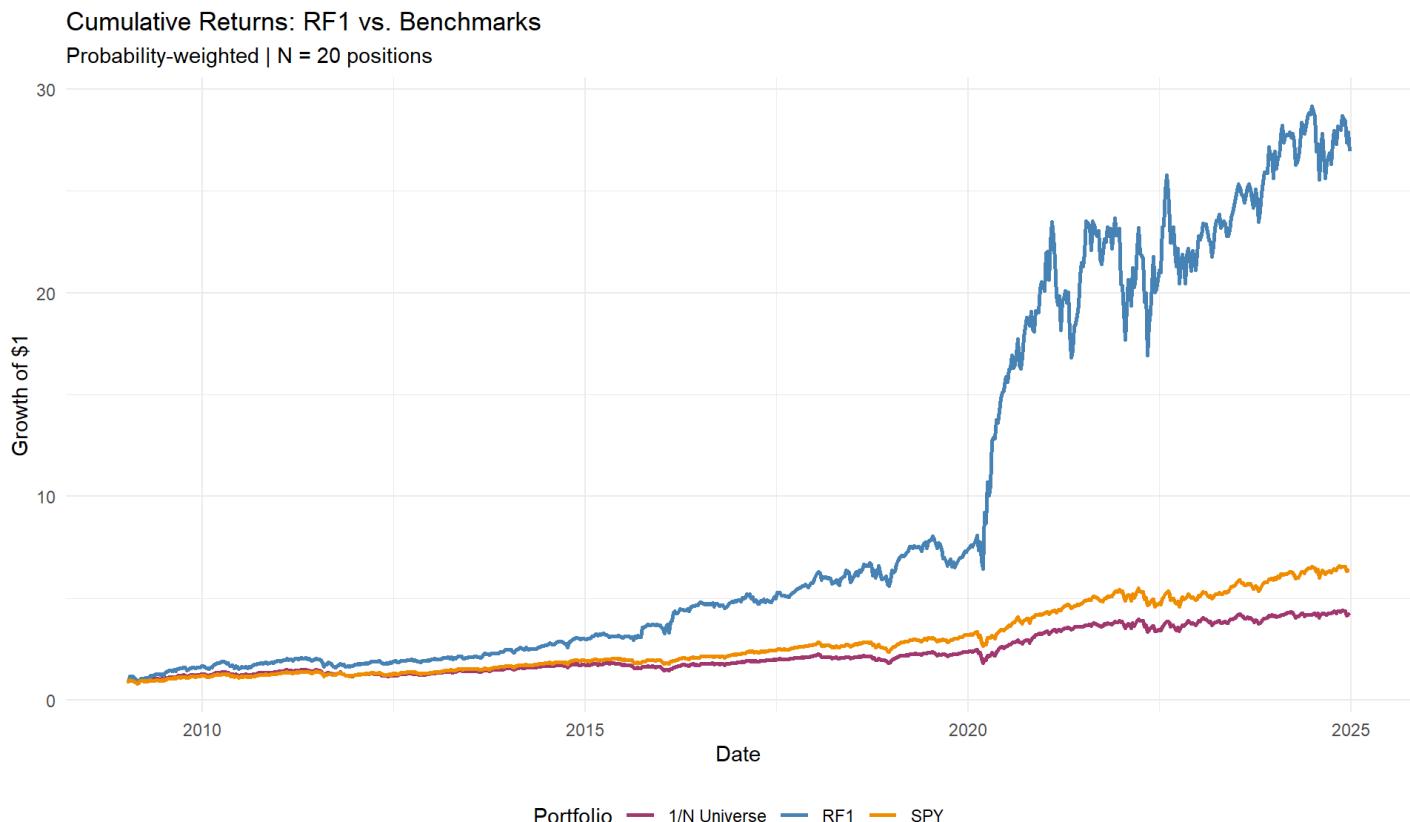
```

# select top N stocks by prediction
selected <- data %>%
  arrange(datadate, desc(prediction)) %>%
  group_by(datadate, model) %>%
  slice_head(n = n_long) %>%
  ungroup()
# calculate portfolio returns
portfolios <- selected %>%
  group_by(datadate, model) %>%
  summarise(
    return = if (prob_weighted) weighted.mean(return, prediction) else mean(return),
    .groups = "drop"
  ) %>%
  group_by(model) %>%
  arrange(datadate) %>%
  mutate(cumulative_return = cumprod(1 + return), datadate = as.Date(datadate)) %>%
  ungroup() %>%
  mutate(year = date_to_year(as.character(datadate)))
# add 1/N benchmark (equal-weighted universe)
equal_weighted <- data %>%
  group_by(datadate) %>%
  summarise(return = mean(return), .groups = "drop") %>%
  mutate(
    model = "1/N Universe",
    cumulative_return = cumprod(1 + return),
    year = date_to_year(as.character(datadate)),
    datadate = as.Date(datadate)
  )
# add SPY benchmark
spy_benchmark <- data %>%
  group_by(datadate) %>%
  summarise(return = mean(spy_return), .groups = "drop") %>%
  mutate(
    model = "SPY",
    cumulative_return = cumprod(1 + return),
    year = date_to_year(as.character(datadate)),
    datadate = as.Date(datadate)
  )
portfolios <- bind_rows(portfolios, equal_weighted, spy_benchmark)

```

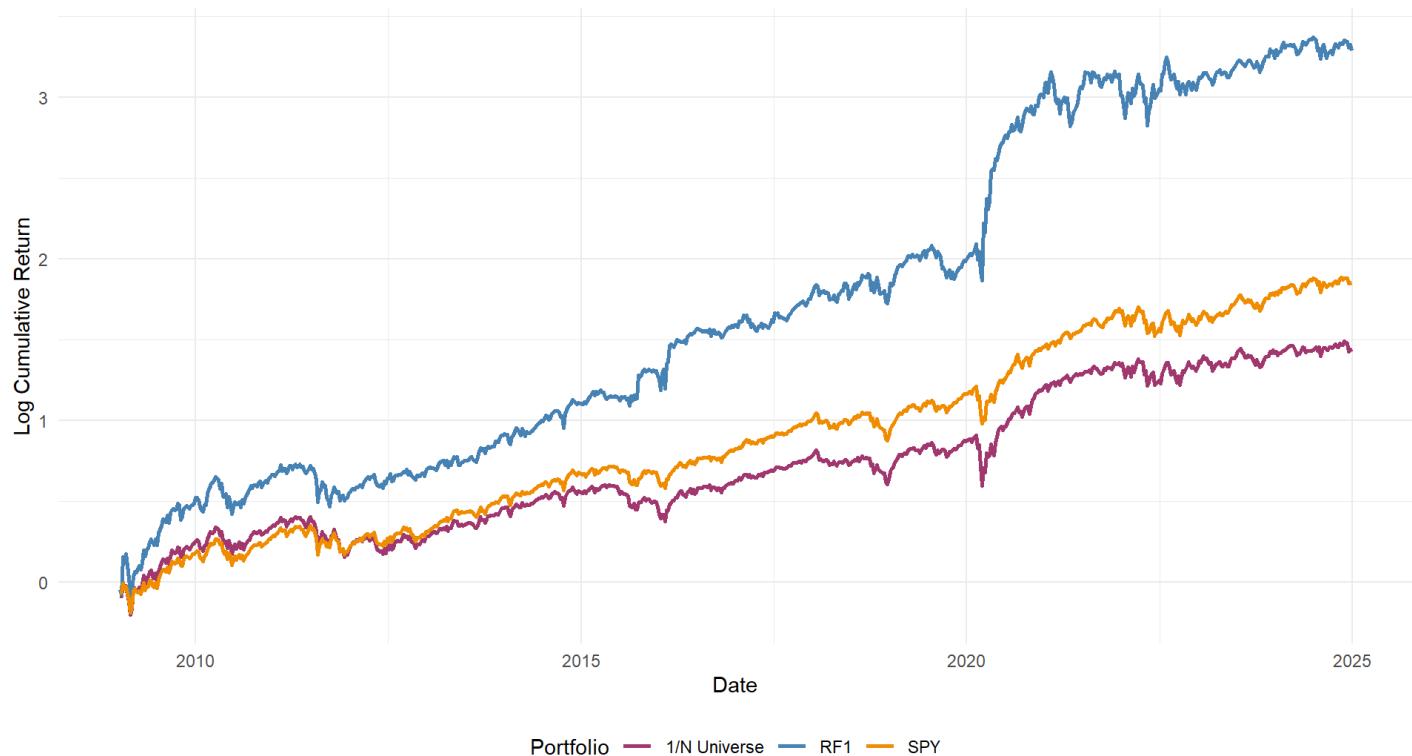
6.3 Cumulative Performance

```
ggplot(portfolios, aes(x = datadate, y = cumulative_return, color = model)) +  
  geom_line(linewidth = 1) +  
  scale_color_manual(values = c("RF1" = "steelblue", "1/N Universe" = "#A23B72", "SPY" = "#F18F0  
1")) +  
  scale_y_continuous(labels = comma) +  
  labs(  
    title = "Cumulative Returns: RF1 vs. Benchmarks",  
    subtitle = paste0("Probability-weighted | N = ", n_long, " positions"),  
    x = "Date", y = "Growth of $1", color = "Portfolio"  
) +  
  theme_minimal() +  
  theme(legend.position = "bottom")
```



```
ggplot(portfolios, aes(x = datadate, y = log(cumulative_return), color = model)) +  
  geom_line(linewidth = 1) +  
  scale_color_manual(values = c("RF1" = "steelblue", "1/N Universe" = "#A23B72", "SPY" = "#F18F0  
1")) +  
  labs(  
    title = "Log Cumulative Returns",  
    subtitle = "Log scale reveals compound growth rate stability over time",  
    x = "Date", y = "Log Cumulative Return", color = "Portfolio"  
) +  
  theme_minimal() +  
  theme(legend.position = "bottom")
```

Log Cumulative Returns
Log scale reveals compound growth rate stability over time



The log plot is useful for identifying performance decay or regime changes in strategy alpha.

7. Performance Metrics

```
spy_returns <- portfolios %>%
  filter(model == "SPY") %>%
  select(datadate, spy_return = return)

metrics <- portfolios %>%
  left_join(spy_returns, by = "datadate") %>%
  group_by(model) %>%
  summarise(
    Ann_Return = mean(return, na.rm = TRUE) * periods_per_year,
    Ann_Vol = sd(return, na.rm = TRUE) * sqrt(periods_per_year),
    Sharpe = Ann_Return / Ann_Vol,
    Max_DD = min((cumprod(1 + return) - cummax(cumprod(1 + return))) / cummax(cumprod(1 + return))),
    Info_Ratio = if (unique(model) == "SPY") NA_real_ else {
      active <- return - spy_return
      mean(active, na.rm = TRUE) * periods_per_year / (sd(active, na.rm = TRUE) * sqrt(periods_per_year))
    },
    .groups = "drop"
  ) %>%
  mutate(across(where(is.numeric), ~round(., 3)))

metrics
```

```

## # A tibble: 3 × 6
##   model      Ann_Return Ann_Vol Sharpe Max_DD Info_Ratio
##   <chr>       <dbl>     <dbl>   <dbl>     <dbl>
## 1 1/N Universe  0.103    0.17    0.606   -0.266    -0.496
## 2 RF1          0.241    0.277   0.87    -0.285     0.609
## 3 SPY          0.127    0.151   0.836   -0.203      NA

```

The RF system delivers higher returns than both benchmarks but with proportionally higher volatility. The Information Ratio quantifies risk-adjusted alpha relative to SPY—values between 0.5 and 1.0 are considered good.

8. Portfolio Size Sensitivity

We examine how performance changes with the number of positions held (N = 5, 10, 20, 50, 100, 250).

```

sizes <- c(5, 10, 20, 50, 100, 250)

make_portfolio <- function(n) {
  data %>%
    arrange(datadate, desc(prediction)) %>%
    group_by(datadate) %>%
    slice_head(n = n) %>%
    summarise(return = if (prob_weighted) weighted.mean(return, prediction) else mean(return), .
groups = "drop") %>%
    mutate(
      model = paste0("Top_", n),
      datadate = as.Date(datadate),
      cumulative_return = cumprod(1 + return)
    )
}

size_portfolios <- map_dfr(sizes, make_portfolio)
size_portfolios <- bind_rows(size_portfolios, equal_weighted, spy_benchmark)

```

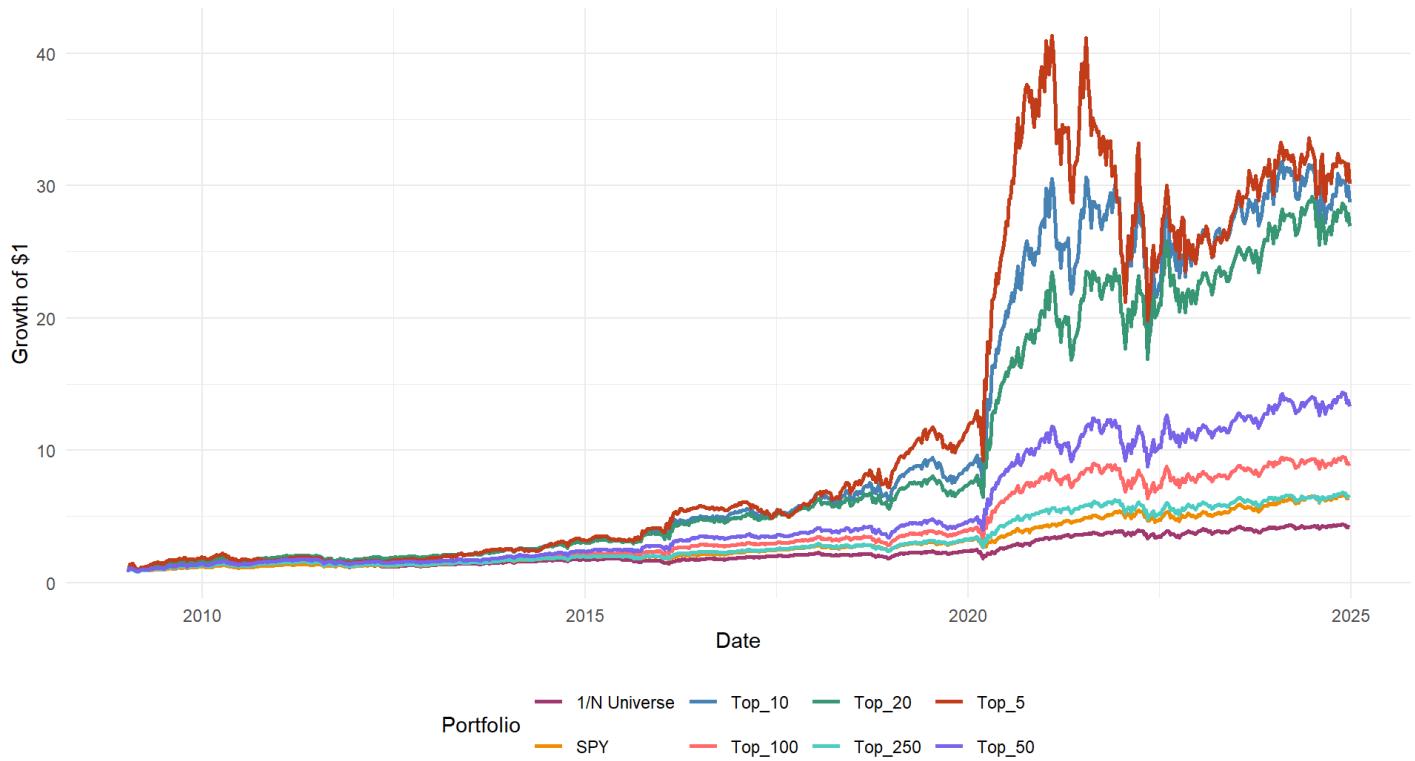
```

size_colors <- c("Top_5" = "#C73E1D", "Top_10" = "steelblue", "Top_20" = "#3B9A74",
                 "Top_50" = "#7B68EE", "Top_100" = "#FF6B6B", "Top_250" = "#4ECDC4",
                 "1/N Universe" = "#A23B72", "SPY" = "#F18F01")

ggplot(size_portfolios, aes(x = datadate, y = cumulative_return, color = model)) +
  geom_line(linewidth = 1) +
  scale_color_manual(values = size_colors) +
  scale_y_continuous(labels = comma) +
  labs(
    title = "Cumulative Returns by Portfolio Size",
    subtitle = "Smaller N = higher concentration, higher volatility",
    x = "Date", y = "Growth of $1", color = "Portfolio"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")

```

Cumulative Returns by Portfolio Size
Smaller N = higher concentration, higher volatility



```

size_metrics <- size_portfolios %>%
  left_join(spy_returns, by = "date") %>%
  group_by(model) %>%
  summarise(
    Ann_Return = mean(return, na.rm = TRUE) * periods_per_year,
    Ann_Vol = sd(return, na.rm = TRUE) * sqrt(periods_per_year),
    Sharpe = Ann_Return / Ann_Vol,
    Max_DD = min((cumprod(1 + return) - cummax(cumprod(1 + return))) / cummax(cumprod(1 + return))),
    Info_Ratio = if (unique(model) == "SPY") NA_real_ else {
      active <- return - spy_return
      mean(active, na.rm = TRUE) * periods_per_year / (sd(active, na.rm = TRUE) * sqrt(periods_per_year))
    },
    .groups = "drop"
  ) %>%
  mutate(across(where(is.numeric), ~round(., 3)))

size_metrics

```

```

## # A tibble: 8 × 6
##   model      Ann_Return Ann_Vol Sharpe Max_DD Info_Ratio
##   <chr>        <dbl>    <dbl>   <dbl>    <dbl>       <dbl>
## 1 1/N Universe 0.103     0.17   0.606   -0.266     -0.496
## 2 SPY           0.127     0.151   0.836   -0.203       NA
## 3 Top_10        0.254     0.313   0.812   -0.368     0.553
## 4 Top_100       0.158     0.219   0.725   -0.294     0.272
## 5 Top_20        0.241     0.277   0.87    -0.285     0.609
## 6 Top_250       0.133     0.187   0.708   -0.231     0.084
## 7 Top_5         0.269     0.356   0.756   -0.52      0.502
## 8 Top_50        0.19      0.244   0.775   -0.294     0.42

```

All RF portfolio configurations show positive Information Ratios. Concentrated portfolios (smaller N) have higher return potential but also higher drawdown risk.

9. Conclusion

The RF trading system demonstrates positive alpha generation across multiple portfolio configurations. Key findings:

- The model delivers higher returns than both the 1/N universe and SPY benchmarks
- This comes at the cost of higher volatility and larger drawdowns
- Information Ratios remain positive across all position sizes tested
- Concentrated portfolios ($N < 20$) show the highest return potential but also the most tail risk

The drawdown of ~30% for the best long portfolio size ($N = 20$) by information ratio represents significant risk that must be considered when allocating capital. The post-2020 regime change (increased target dispersion) warrants monitoring for potential model degradation and research into new features as data became more accessible over time.