```r
### loading all the packages ###
library(readr)
library(dplyr)
library(lubridate)
library(tidyr)
install.packages("gt")
library(gt)
library(ggplot2)
library(purrr)



######## 1 A ##########


data <- read.csv('./Desktop/ASSIGNMENT MATERIALS-20231219/
exam2023-24.csv')
data$date <- as.Date(paste("01 ", data$date), format="%d
%b %Y")
data <- data |>
  mutate(
    year = year(date),
    month = month(date, label = TRUE, abbr = TRUE)
  )

avg_ret <- data |>
  group_by(FFI10_desc, month) |>
  summarise(avg_indret_ew = mean(indret_ew, na.rm = TRUE))
|>
  ungroup()

avg_ret_wide <- pivot_wider(avg_ret, names_from = month,
values_from = avg_indret_ew)

pretty_table <- gt(avg_ret_wide) %>%
  tab_header(title = "Average Monthly Returns by
Industry") %>%
  cols_label(
```

```r
    FFI10_desc = "Industry",
    Jan = "January", Feb = "February", Mar = "March",
    Apr = "April", May = "May", Jun = "June",
    Jul = "July", Aug = "August", Sep = "September",
    Oct = "October", Nov = "November", Dec = "December"
  ) %>%
  tab_options(
    heading.background.color = "gray",
    column_labels.font.size = "small",
    row_group.background.color = "lightgray"
  )

print(pretty_table)

##### 1b #######

std_dev <- data |>
  group_by(FFI10_desc, month) |>
  summarise(std_indret_ew = sd(indret_ew, na.rm = TRUE)) |
>
  ungroup()

std_dev_wide <- pivot_wider(std_dev, names_from = month,
 values_from = std_indret_ew)

std_table <- gt(std_dev_wide) %>%
  tab_header(title = "Time-Series Standard Deviations of
 indret ew by Industry") %>%
  cols_label(
    FFI10_desc = "Industry",
    Jan = "January", Feb = "February", Mar = "March",
    Apr = "April", May = "May", Jun = "June",
    Jul = "July", Aug = "August", Sep = "September",
    Oct = "October", Nov = "November", Dec = "December"
  ) %>%
  tab_options(
    heading.background.color = "gray",
    column_labels.font.size = "small",
    row_group.background.color = "lightgray"
```

```
  )

print(std_table)

sharpe_ratios <- data |>
  group_by(FFI10_desc, month) |>
  summarise(sharpe = mean(indret_ew, na.rm = TRUE) /
 sd(indret_ew, na.rm = TRUE)) |>
  ungroup()
sharpe_ratios_wide <- pivot_wider(sharpe_ratios,
 names_from = month, values_from = sharpe)

sharpe_table <- gt(sharpe_ratios_wide) %>%
  tab_header(title = "Sharpe Ratios of indret ew by
 Industry") %>%
  cols_label(
    FFI10_desc = "Industry",
    Jan = "January", Feb = "February", Mar = "March",
    Apr = "April", May = "May", Jun = "June",
    Jul = "July", Aug = "August", Sep = "September",
    Oct = "October", Nov = "November", Dec = "December"
  ) %>%
  tab_options(
    heading.background.color = "gray",
    column_labels.font.size = "small",
    row_group.background.color = "lightgray"
  )

print(sharpe_table)


### 1c ####


industries_ew <- data %>%
  select(date, FFI10_desc, indret_ew) %>%
  pivot_wider(names_from = FFI10_desc, values_from =
 indret_ew) %>%
  select(-date)
```

```
varcov_ew <- cov(industries_ew, use =
"pairwise.complete.obs")

industries_vw <- data %>%
  select(date, FFI10_desc, indret_vw) %>%
  pivot_wider(names_from = FFI10_desc, values_from =
indret_vw) %>%
  select(-date)

varcov_vw <- cov(industries_vw, use =
"pairwise.complete.obs")

cov_difference <- varcov_ew - varcov_vw

varcov_ew_table <- as.data.frame(varcov_ew)

varcov_vw_table <- as.data.frame(varcov_vw)

cov_difference_table <- as.data.frame(cov_difference)

print("Covariance Matrix for Equal-Weighted Industry
Returns:")
print(varcov_ew_table)

print("Covariance Matrix for Value-Weighted Industry
Returns:")
print(varcov_vw_table)

print("Difference between Covariance Matrices (varcov_ew -
varcov_vw):")
print(cov_difference_table)

varcov_ew_vector <- as.numeric(varcov_ew)
varcov_vw_vector <- as.numeric(varcov_vw)

cov_comparison_data <- data.frame(
  Matrix = rep(c("varcov_ew", "varcov_vw"), each =
length(varcov_ew)),
```

```
  Value = c(varcov_ew_vector, varcov_vw_vector)
)
cov_comparison_data <- data.frame(
  Matrix = rep(c("Matrix A", "Matrix B"), each = 10),
  Value = c(rnorm(10, mean = 5, sd = 2), rnorm(10, mean =
 10, sd = 3))
)
ggplot(cov_comparison_data, aes(x = Matrix, y = Value,
 fill = Matrix)) +
  geom_boxplot() +
  labs(title = "Boxplot Comparison of Covariance
 Matrices",
       x = "Matrix", y = "Covariance Value") +
  theme_minimal()

calculate_beta <- function(return_type) {
  betas <- data %>%
    group_by(FFI10_desc) %>%
    do(tidy(lm(reformulate("rM", response = return_type),
 data = .))) %>%
    filter(term == "rM") %>%
    select(FFI10_desc, beta = estimate) %>%
    ungroup() %>%
    arrange(desc(beta)) # Sorting in descending order of
 beta

  return(betas)
}
beta_ew <- calculate_beta("indret_ew")
beta_vw <- calculate_beta("indret_vw")

plot_betas <- function(betas, title) {
  ggplot(betas, aes(x = reorder(FFI10_desc, -beta), y =
 beta)) + # Reorder based on beta
    geom_bar(stat = "identity", fill = "steelblue") +
    coord_flip() +  # Flips the axes for better visibility
    theme_minimal() +
    xlab("Industry") +
    ylab("Market Beta") +
```

```
    ggtitle(title) +
    theme(axis.text.x = element_text(angle = 45, hjust =
 1))
}
plot_ew <- plot_betas(beta_ew, "Market Betas for 'indret
 ew'")
plot_vw <- plot_betas(beta_vw, "Market Betas for 'indret
 vw'")

print(plot_ew)
print(plot_vw)

industries <- unique(data$FFI10_desc)
financial_ratios <- colnames(data)[6:ncol(data)]  #
 Assuming financial ratios start from the 6th column

results <- list()

for (industry in industries) {
  for (ratio in financial_ratios) {
    # Filter data for the industry and create a lagged
 version of the financial ratio
    industry_data <- data %>%
      filter(FFI10_desc == industry) %>%
      arrange(date) %>%
      mutate(!!paste0("lagged_", ratio) := lag(!!
 sym(ratio)))
    formula <- as.formula(paste("indret_ew ~",
 paste0("lagged_", ratio)))
    model <- lm(formula, data = industry_data)
    tidy_model <- tidy(model)
    result_name <- paste(industry, ratio, sep = "_")
    results[[result_name]] <- tidy_model
  }
}
analyze_results <- function(results) {
  significant_predictors <- list()
  common_predictors <- list()
```

```
  for (result_name in names(results)) {
    result <- results[[result_name]]
    significant_terms <- result %>% filter(term !=
 "(Intercept)", p.value < 0.05)

    if (nrow(significant_terms) > 0) {
      industry_ratio <- strsplit(result_name, " ")[[1]]
      significant_predictors[[result_name]] <-
 significant_terms
      common_predictors[[industry_ratio[2]]] <-
 c(common_predictors[[industry_ratio[2]]],
 industry_ratio[1])
    }
  }

  list(significant_predictors = significant_predictors,
 common_predictors = common_predictors)
}
final_results <- analyze_results(results)
print(final_results)

####### 2 b RMSE #######

industries <- unique(data$FFI10_desc)
financial_ratios <- colnames(data)[6:ncol(data)]  #
 Assuming financial ratios start from the 6th column

calculate_rmse <- function(actual, predicted) {
  sqrt(mean((predicted - actual)^2, na.rm = TRUE))
}
rmse_results <- matrix(NA, nrow = length(industries), ncol
 = length(financial_ratios))
rownames(rmse_results) <- industries
colnames(rmse_results) <- financial_ratios
for (industry in industries) {
  for (ratio in financial_ratios) {
    # Prepare the data
    industry_data <- data %>%
      filter(FFI10_desc == industry) %>%
```

```
      arrange(date) %>%
      mutate(!!rlang::sym(paste0("lagged_", ratio)) :=
dplyr::lag(!!rlang::sym(ratio))) %>%
      na.omit()
    formula <- as.formula(paste("indret_ew ~",
paste0("lagged_", ratio)))
    model <- lm(formula, data = industry_data)
    predictions <- predict(model, newdata = industry_data)
    rmse_results[industry, ratio] <-
calculate_rmse(industry_data$indret_ew, predictions)
  }
}
best_predictors <- apply(rmse_results, 1, which.min)
best_predictors <- colnames(rmse_results)[best_predictors]
print(best_predictors)
best_predictors_df <- data.frame(
  Industry = industries,
  Best_Predictor = best_predictors
)
print(best_predictors_df)

##### PART B #######

run_rolling_regression <- function(data, industry,
predictor) {
  start_index <- 1
  end_index <- 120
  sse_values <- numeric()

  while (end_index < nrow(data)) {

    rolling_data <- data[start_index:end_index, ]


    formula <- as.formula(paste("indret_ew ~", predictor))
    model <- lm(formula, data = rolling_data)
    if (end_index + 1 <= nrow(data)) {
      next_month_data <- data[end_index + 1, ]
      predicted_value <- predict(model, newdata =
```

```
next_month_data)
      actual_value <- next_month_data[["indret_ew"]]


      squared_error <- (predicted_value - actual_value)^2
      sse_values <- c(sse_values, squared_error)
    }


    start_index <- start_index + 1
    end_index <- end_index + 1
  }

  cumulative_sse <- cumsum(sse_values)
  return(data.frame(Month = (121:(120 +
length(cumulative_sse))), Cumulative_SSE =
cumulative_sse, Industry = industry))
}

cumulative_sse_results <- data.frame()


for (industry in industries) {
  industry_data <- data %>% filter(FFI10_desc == industry)
  best_predictor <-
best_predictors_df[best_predictors_df$Industry ==
industry, "Best_Predictor"]


  if (best_predictor %in% colnames(industry_data)) {
    industry_sse <- run_rolling_regression(industry_data,
industry, best_predictor)
    cumulative_sse_results <-
rbind(cumulative_sse_results, industry_sse)
  }
}


ggplot(cumulative_sse_results, aes(x = Month, y =
```

```
Cumulative_SSE, color = Industry)) +
  geom_line() +
  labs(title = "Cumulative SSE for Predictors by
Industry",
       x = "Month", y = "Cumulative SSE") +
  theme_minimal()

run_rolling_regression_benchmark <- function(data,
industry) {
  mean_return <- mean(data$indret_ew[1:120], na.rm = TRUE)
  sse_values <- numeric()

  for (end_index in 121:nrow(data)) {
    actual_value <- data[end_index, "indret_ew"]
    squared_error <- (mean_return - actual_value)^2
    sse_values <- c(sse_values, squared_error)
  }

  cumulative_sse <- cumsum(sse_values)
  month_sequence <- 121:(120 + length(cumulative_sse))
  return(data.frame(Month = month_sequence,
Cumulative_SSE_Benchmark = cumulative_sse, Industry =
industry))
}

cumulative_sse_benchmark_results <- data.frame()

for (industry in industries) {
  industry_data <- data %>% filter(FFI10_desc == industry)
  industry_sse_benchmark <-
run_rolling_regression_benchmark(industry_data, industry)
  cumulative_sse_benchmark_results <-
rbind(cumulative_sse_benchmark_results,
industry_sse_benchmark)
}

final_results <- merge(cumulative_sse_results,
cumulative_sse_benchmark_results, by = c("Month",
"Industry"))
```

```
final_results$SSE_Difference <-
 final_results$Cumulative_SSE -
 final_results$Cumulative_SSE_Benchmark


ggplot(final_results, aes(x = Month, y = SSE_Difference,
 color = Industry)) +
  geom_line() +
  labs(title = "Cumulative Difference in SSE (Model vs.
 Benchmark) by Industry",
       x = "Month", y = "Cumulative SSE Difference") +
  theme_minimal()



#### Part B 4 A ######

library(quadprog)
library(PerformanceAnalytics)
library(dplyr)
library(tidyr)
library(ggplot2)

returns <- data %>%
  select(date, FFI10_desc, indret_ew) %>%
  spread(key = FFI10_desc, value = indret_ew)
returns <- returns[,-1]
cov_matrix <- cov(returns, use = "complete.obs")
dvec <- rep(0, ncol(cov_matrix))
amat <- t(matrix(1, ncol(cov_matrix)))
bvec <- 1
optimal_weights <- solve.QP(cov_matrix, dvec, amat, bvec,
 meq = 1)$solution
portfolio_returns <- as.matrix(returns) %*%
 optimal_weights
cumulative_returns <- cumprod(1 + portfolio_returns)
cumulative_returns_df <- data.frame(date = data$date,
 CumulativeReturns = cumulative_returns)
```

```
ggplot(cumulative_returns_df, aes(x = date, y =
 CumulativeReturns)) +
  geom_line(color = 'blue') +
  theme_minimal() +
  labs(title = "Cumulative Returns of the Weighted
 Portfolio",
       x = "Date",
       y = "Cumulative Returns")
print(plot)


data$date <- as.Date(as.yearmon(data$date, "%b %Y"))
returns <- data %>%
  select(date, FFI10_desc, indret_ew) %>%
  spread(key = FFI10_desc, value = indret_ew)

portfolio_returns <- vector("numeric", length =
 nrow(returns) - 120)

for (i in 121:nrow(returns)) {
  past_data <- returns[(i-120):(i-1), -1]

  # Calculate the covariance matrix
  cov_matrix <- cov(past_data, use = "complete.obs")

  dvec <- rep(0, ncol(cov_matrix))
  amat <- t(matrix(1, ncol(cov_matrix)))
  bvec <- 1
  optimal_weights <- solve.QP(cov_matrix, dvec, amat,
 bvec, meq = 1)$solution

  next_month_returns <- returns[i, -1]
  portfolio_returns[i-120] <- sum(optimal_weights *
 next_month_returns)
}

cumulative_returns <- cumprod(1 + portfolio_returns)

cumulative_returns_df <- data.frame(date =
```

```
returns$date[121:nrow(returns)], CumulativeReturns =
cumulative_returns)

cumulative_plot <- ggplot(cumulative_returns_df, aes(x =
date, y = CumulativeReturns)) +
  geom_line(color = 'blue') +
  theme_minimal() +
  labs(title = "Cumulative Returns of the Rolling Window
Portfolio",
       x = "Date",
       y = "Cumulative Returns")

annualized_sharpe <-
SharpeRatio.annualized(portfolio_returns, Rf = 0, scale =
252) # assuming 252 trading days

print(cumulative_plot)

print(paste("Annualized Sharpe Ratio:",
annualized_sharpe))
```