Porteføljе samenseting

```
################----------ISIN--prices----------#####################3
library(lubridate)
library(openxlsx)
library(readxl)
library(dplyr)
library(tidyr)
library(zoo)



#####----Convert-the-Date-column-to-data-type----#####


titlon_data <- read_excel('Documents/høst 2023/Full_data.xlsx')

titlon_data <- titlon_data %>%
  filter(Sector != "ETF")

titlon_data <- titlon_data %>%
  filter(Name != "ABG Sundal Collier Holding")

titlon_data <- titlon_data %>%
  filter(!grepl("ser. B", Name, fixed = TRUE))

titlon_data <- titlon_data %>%
  filter(ISIN != "SE0009723034", ISIN != "NO0010441520", ISIN != "NO0010441512",
         ISIN != "NO0010489511", ISIN != "SE0009723000", ISIN != "SE0009723026",
         ISIN != "NO0010843998", ISIN != "NO0010489529")

titlon_data <- titlon_data %>%
  filter(Year != "2023")

titlon_data <- titlon_data %>%
  filter(!(lnDeltaP > 1.0 | lnDeltaP < -1.0))

titlon_data$Date <- as.Date(titlon_data$Date)

titlon_data <- titlon_data %>%
  filter(AdjustedPrice >= 1, mktcap >= 50000000)

titlon_data <- titlon_data %>%
  group_by(ISIN) %>%
  filter(n() >= 60)%>%
  ungroup()

number_of_unique_isins <- titlon_data %>%
  summarise(UniqueISINs = n_distinct(ISIN)) %>%
  pull(UniqueISINs)
```

```
number_of_unique_isins

# summarize lndeltaP

# Convert percentage to a multiplier
# Create a Quarter column

titlon_data <- titlon_data %>%
  mutate(lnDeltaP = lnDeltaP - bills_3month_Lnrate)

# Convert lnDeltaP from percent to multiplier

titlon_data <- titlon_data %>%
  mutate(Multiplier = 1 + (lnDeltaP),
         Quarter = paste0(year(Date), "-Q", quarter(Date)))

write.xlsx(titlon_data, file = "Documents/høst 2023/titlon_data.xlsx", rowNames = FALSE)


# Calculate total percentage change for each ISIN and Quarter
quarterly_summary <- titlon_data %>%
  group_by(ISIN, Quarter) %>%
  arrange(ISIN, Date) %>%
  summarise(Total_Percentage_Change = (prod(Multiplier, na.rm = TRUE) - 1),
            mktcap = last(mktcap),
            .groups = 'drop')

write.xlsx(quarterly_summary, file = "Documents/høst 2023/quarterly_summary.xlsx", rowNames =
  FALSE)


# Updating the transformation to monthly basis and creating a Month column
# Convert lnDeltaP from percent to multiplier
# Create a Month column in "YYYY-MM" format
titlon_data_month <- titlon_data %>%
  mutate(Multiplier = 1 + (lnDeltaP),
         Month = format(Date, "%Y-%m"))

# Writing the modified data to an Excel file
write.xlsx(titlon_data_month, file = "Documents/høst 2023/titlon_data_month.xlsx", rowNames =
  FALSE)

# Calculate total percentage change for each ISIN and Month
monthly_summary <- titlon_data_month %>%
  group_by(ISIN, Month) %>%
  arrange(ISIN, Date) %>%
  summarise(Total_Percentage_Change = (prod(Multiplier, na.rm = TRUE) - 1),
            mktcap = last(mktcap),
            .groups = 'drop')
```

```
write.xlsx(monthly_summary, file = "Documents/høst 2023/monthly_summary.xlsx", rowNames =
 FALSE)



##########----------reading-Excel-file-dataset----------


titlon_data <- read_excel('Documents/høst 2023/titlon_data.xlsx')
quarterly_summary <- read_excel('Documents/høst 2023/quarterly_summary.xlsx')
monthly_summary <- read_excel('Documents/høst 2023/monthly_summary.xlsx')

##################-------- portefolio 10% bottom---##################

#### Filter to keep only the bottom 20% of ISINs by market cap, wich leaves the 10% total marketcpt
 companies ####

bottom_20_percent <- quarterly_summary %>%
 group_by(Quarter) %>%
 mutate(market_cap_threshold_20 = quantile(mktcap, 0.2, na.rm = TRUE)) %>%
 filter(mktcap <= market_cap_threshold_20) %>%
 ungroup()

write.xlsx(bottom_20_percent, file = "Documents/høst 2023/bottom_20_percent.xlsx", rowNames =
 FALSE)

##################--------matching portefolio 50% bottom---##################


bottom_50_percent <- quarterly_summary %>%
 group_by(Quarter) %>%
 mutate(market_cap_threshold_50 = quantile(mktcap, 0.5, na.rm = TRUE)) %>%
 filter(mktcap <= market_cap_threshold_50) %>%
 ungroup()


write.xlsx(bottom_50_percent, file = "Documents/høst 2023/bottom_50_percent.xlsx", rowNames =
 FALSE)


#########--------Crete-data-in-bi-quarterly-form-----###########


bi_quarterly_data <- bottom_50_percent

bi_quarterly_data <- bi_quarterly_data %>%
 arrange(ISIN, Quarter) %>%
 mutate(Cumulative_Change = Total_Percentage_Change + 1) %>%
 group_by(ISIN) %>%
 mutate(Bi_Quarterly_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change) %>%
 ungroup()
```

```r
bi_quarterly_data <- bi_quarterly_data %>%
  filter(Quarter != '2009-Q2')
bi_quarterly_data <- bi_quarterly_data %>%
  filter(Quarter != '2009-Q1')

# Adjust the data changes back to percentages
bi_quarterly_Bottom50 <- bi_quarterly_data %>%
  mutate(Bi_Quarterly_Change = (Bi_Quarterly_Change - 1))

write.xlsx(bi_quarterly_Bottom50, file = "Documents/høst 2023/bi_quarterly_Bottom50.xlsx",
  rowNames = FALSE)


bi_quarterly_bottom20 <- bottom_20_percent

bi_quarterly_bottom20 <- bi_quarterly_bottom20 %>%
  arrange(ISIN, Quarter) %>%
  mutate(Cumulative_Change = Total_Percentage_Change + 1) %>%
  group_by(ISIN) %>%
  mutate(Bi_Quarterly_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change) %>%
  ungroup()

bi_quarterly_bottom20 <- bi_quarterly_bottom20 %>%
  filter(Quarter != '2009-Q1')

# Adjust the data by removing unnecessary rows and converting changes back to percentages
bi_quarterly_bottom20 <- bi_quarterly_bottom20 %>%
  mutate(Bi_Quarterly_Change = (Bi_Quarterly_Change - 1))


##############----------Crete-data-in-per-Q4-form----------################


Q4_bottom20 <- bottom_20_percent

Q4_bottom20 <- Q4_bottom20 %>%
  arrange(ISIN, Quarter) %>%
  mutate(Cumulative_Change = Total_Percentage_Change + 1) %>%
  group_by(ISIN) %>%
  mutate(Q4_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change,
    Q4_Change = lag(Cumulative_Change, n = 2, default = 1) * Q4_Change,
    Q4_Change = lag(Cumulative_Change, n = 3, default = 1) * Q4_Change) %>%
  ungroup()

Q4_bottom20 <- Q4_bottom20 %>%
  filter(Quarter != '2009-Q2')
Q4_bottom20 <- Q4_bottom20 %>%
  filter(Quarter != '2009-Q1')
Q4_bottom20 <- Q4_bottom20 %>%
  filter(Quarter != '2009-Q3')

# Adjust the data by removing unnecessary rows and converting changes back to percentages
```

```
Q4_bottom20 <- Q4_bottom20 %>%
  mutate(Q4_Change = (Q4_Change - 1))

write.xlsx(Q4_bottom20, file = "Documents/høst 2023/Q4_bottom20.xlsx", rowNames = FALSE)


Q4_bottom50 <- bottom_50_percent

Q4_bottom50 <- Q4_bottom50 %>%
  arrange(ISIN, Quarter) %>%
  mutate(Cumulative_Change = Total_Percentage_Change + 1) %>%
  group_by(ISIN) %>%
  mutate(Q4_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change,
    Q4_Change = lag(Cumulative_Change, n = 2, default = 1) * Q4_Change,
    Q4_Change = lag(Cumulative_Change, n = 3, default = 1) * Q4_Change) %>%
  ungroup()

Q4_bottom50 <- Q4_bottom50 %>%
  filter(Quarter != '2009-Q2')
Q4_bottom50 <- Q4_bottom50 %>%
  filter(Quarter != '2009-Q1')
Q4_bottom50 <- Q4_bottom50 %>%
  filter(Quarter != '2009-Q3')

#  removing unnecessary rows and converting changes back to percentages
Q4_bottom50 <- Q4_bottom50 %>%
  mutate(Q4_Change = (Q4_Change - 1))


write.xlsx(Q4_bottom50, file = "Documents/høst 2023/Q4_bottom50.xlsx", rowNames = FALSE)


###################------portfolio-performance-----####################3


# Step 1: Filter for 2009-Q1 and select top 20% ISINs based on lnDeltaP
top_20_isins <- bottom_20_percent %>%
  filter(Quarter == '2009-Q1') %>%
  mutate(lnDeltaP_rank = min_rank(desc(Total_Percentage_Change)) / n()) %>%
  filter(lnDeltaP_rank <= 0.20) %>%
  pull(ISIN)  # Extract ISINs of the top 15%

# Step 2: Filter the data for 2009-Q2 and for the ISINs in the top 2o% portfolio
portfolio_performance_Q2 <- bottom_50_percent %>%
  filter(ISIN %in% top_20_isins, Quarter == '2009-Q2') %>%
  summarize((Total_lnDeltaP_Q2 = sum(Total_Percentage_Change, na.rm =
  TRUE)/length(top_20_isins)))

# Print the combined lnDeltaP value for these ISINs in 2009-Q2
print(portfolio_performance_Q2*100)

list(top_20_isins)
```

```r
# Step 1: Filter for 2009-Q1 and select top 20% ISINs based on lnDeltaP
top_20Q1_isins <- bottom_20_percent %>%
  filter(Quarter == '2009-Q3') %>%
  mutate(lnDeltaP_rank = min_rank(desc(Total_Percentage_Change)) / n()) %>%
  filter(lnDeltaP_rank <= 0.20) %>%
  pull(ISIN)

# Step 2: Filter the data for 2009-Q3 and for the ISINs in the top 20% portfolio
portfolio_performance_Q2 <- bi_quarterly_Bottom50 %>%
  filter(ISIN %in% top_20Q1_isins, Quarter == '2010-Q1') %>%
  summarize((Total_lnDeltaP_Q2 = sum(Bi_Quarterly_Change, na.rm =
 TRUE)/length(top_20Q1_isins)))

# Print the combined lnDeltaP value hold periods of 2Q

print(portfolio_performance_Q2*100)

# Step 2: Filter the data for 2010-Q1 and for the ISINs in the top 20% portfolio
portfolio_performance_Q4 <- Q4_bottom50 %>%
  filter(ISIN %in% top_20Q1_isins, Quarter == '2010-Q1') %>%
  summarize((Total_lnDeltaP_Q4 = sum(Q4_Change, na.rm = TRUE)/length(top_20Q1_isins)))

# Print the combined lnDeltaP value hold periods of 4Q

print(portfolio_performance_Q4*100)


list(top_20Q1_isins)

# function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
  top_isins <- data %>%
    filter(Quarter == quarter) %>%
    mutate(lnDeltaP_rank = min_rank(desc(Total_Percentage_Change)) / n()) %>%
    filter(lnDeltaP_rank <= 0.20) %>%
    pull(ISIN)
  return(top_isins)
}

# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    summarize(Total_lnDeltaP = sum(Total_Percentage_Change, na.rm = TRUE) / length(isins))
  return(portfolio_performance * 100)  # Result scaled by 100
}
```

```r
# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q1', '2009-Q2', '2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3'),
 next_quarter = c('2009-Q2','2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3', '2022-Q4')
)

# Initialize a ResultsP1Q1 list
ResultsP1Q1 <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
 current_quarter <- quarters$start_quarter[i]
 next_quarter <- quarters$next_quarter[i]

 top_isins <- get_top_isins(bottom_20_percent, current_quarter)
 ResultsP1Q1[[next_quarter]] <- calculate_performance(bottom_50_percent, top_isins,
 next_quarter)
}

# Set names for the ResultsP1Q1 list to reflect the quarters for which performance was calculated
names(ResultsP1Q1) <- quarters$next_quarter



################3----------Q2-hold-period-forQ1-performance------################


# Helper function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
 top_isins <- data %>%
  filter(Quarter == quarter) %>%
  mutate(lnDeltaP_rank = min_rank(desc(Total_Percentage_Change)) / n()) %>%
  filter(lnDeltaP_rank <= 0.20) %>%
  pull(ISIN)
```

```
  return(top_isins)
}

# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    summarize(Total_lnDeltaP = sum(Bi_Quarterly_Change, na.rm = TRUE) / length(isins))
  return(portfolio_performance * 100)  # Result scaled by 100
}

# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                    '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                    '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                    '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                    '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                    '2021-Q3', '2022-Q1'),
  next_quarter = c('2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                   '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                   '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                   '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                   '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                   '2021-Q3', '2022-Q1', '2022-Q3')
)

# Initialize a resultsQ2 list
ResultsP1Q2 <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(bottom_20_percent, current_quarter)
  ResultsP1Q2[[next_quarter]] <- calculate_performance(bi_quarterly_Bottom50, top_isins,
  next_quarter)
}

# Set names for the resultsQ2 list to reflect the quarters for which performance was calculated
names(ResultsP1Q2) <- quarters$next_quarter



################3----------Q4-hold-period-forQ1-performance------################


# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
```

```
   summarize(Total_lnDeltaP = sum(Q4_Change, na.rm = TRUE) / length(isins))
 return(portfolio_performance)  # Result scaled by 100
}

# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q1', '2010-Q1', '2011-Q1',
                   '2012-Q1', '2013-Q1',
                   '2014-Q1', '2015-Q1', '2016-Q1',
                   '2017-Q1', '2018-Q1',
                   '2019-Q1', '2020-Q1', '2021-Q1'),
 next_quarter = c( '2010-Q1', '2011-Q1',
                   '2012-Q1', '2013-Q1',
                   '2014-Q1', '2015-Q1', '2016-Q1',
                   '2017-Q1', '2018-Q1',
                   '2019-Q1', '2020-Q1', '2021-Q1',
                   '2022-Q1')
)

# Initialize a resultsQ2 list
ResultsP1Q4 <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
 current_quarter <- quarters$start_quarter[i]
 next_quarter <- quarters$next_quarter[i]

 top_isins <- get_top_isins(bottom_20_percent, current_quarter)
 ResultsP1Q4[[next_quarter]] <- calculate_performance(Q4_bottom50, top_isins, next_quarter)
}

# Set names for the resultsQ4 list to reflect the quarters for which performance was calculated
names(ResultsP1Q4) <- quarters$next_quarter




################3----------H1-hold-period-for-K2-performance------################

# Helper function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
 top_isins <- data %>%
   filter(Quarter == quarter) %>%
   mutate(lnDeltaP_rank = min_rank(desc(Bi_Quarterly_Change)) / n()) %>%
   filter(lnDeltaP_rank <= 0.20) %>%
   pull(ISIN)
 return(top_isins)
}

# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
 portfolio_performance <- data %>%
```

```
  filter(ISIN %in% isins, Quarter == quarter) %>%
  summarize(Total_lnDeltaP = sum(Total_Percentage_Change, na.rm = TRUE) / length(isins))
 return(portfolio_performance * 100)  # Result scaled by 100
}


# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q2', '2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3'),
 next_quarter = c('2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3', '2022-Q4')
)

# Initialize a ResultsP2Q1 list
ResultsP2Q1 <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
 current_quarter <- quarters$start_quarter[i]
 next_quarter <- quarters$next_quarter[i]

 top_isins <- get_top_isins(bi_quarterly_bottom20, current_quarter)
 ResultsP2Q1[[next_quarter]] <- calculate_performance(bottom_50_percent, top_isins,
 next_quarter)
}

# Set names for the ResultsP2Q1 list to reflect the quarters for which performance was calculated
names(ResultsP2Q1) <- quarters$next_quarter

################3----------Q2-hold-period-forQ2-performance------################

# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
 portfolio_performance <- data %>%
  filter(ISIN %in% isins, Quarter == quarter) %>%
```

```
  summarize(Total_lnDeltaP = sum(Bi_Quarterly_Change, na.rm = TRUE) / length(isins))
 return(portfolio_performance * 100)  # Result scaled by 100
}


# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q2','2009-Q4', '2010-Q2', '2010-Q4', '2011-Q2',
                   '2011-Q4', '2012-Q2', '2012-Q4', '2013-Q2', '2013-Q4',
                   '2014-Q2', '2014-Q4', '2015-Q2', '2015-Q4', '2016-Q2',
                   '2016-Q4', '2017-Q2', '2017-Q4', '2018-Q2', '2018-Q4',
                   '2019-Q2', '2019-Q4', '2020-Q2', '2020-Q4', '2021-Q2',
                   '2021-Q4', '2022-Q2'),
 next_quarter = c('2009-Q4', '2010-Q2', '2010-Q4', '2011-Q2',
                  '2011-Q4', '2012-Q2', '2012-Q4', '2013-Q2', '2013-Q4',
                  '2014-Q2', '2014-Q4', '2015-Q2', '2015-Q4', '2016-Q2',
                  '2016-Q4', '2017-Q2', '2017-Q4', '2018-Q2', '2018-Q4',
                  '2019-Q2', '2019-Q4', '2020-Q2', '2020-Q4', '2021-Q2',
                  '2021-Q4', '2022-Q2', '2022-Q4')
)

# Initialize a ResultsP2Q2list
ResultsP2Q2<- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
 current_quarter <- quarters$start_quarter[i]
 next_quarter <- quarters$next_quarter[i]

 top_isins <- get_top_isins(bi_quarterly_bottom20, current_quarter)
 ResultsP2Q2[[next_quarter]] <- calculate_performance(bi_quarterly_Bottom50, top_isins,
 next_quarter)
}

# Set names for the ResultsP2Q2list to reflect the quarters for which performance was calculated
names(ResultsP2Q2) <- quarters$next_quarter


################3----------Q4-hold-period-forQ2-performance------################


# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
 portfolio_performance <- data %>%
  filter(ISIN %in% isins, Quarter == quarter) %>%
  summarize(Total_lnDeltaP = sum(Q4_Change, na.rm = TRUE) / length(isins))
 return(portfolio_performance * 100)  # Result scaled by 100
}


# Define quarter pairs
quarters <- data.frame(
```

```
  start_quarter = c('2009-Q2', '2010-Q2', '2011-Q2',
                    '2012-Q2', '2013-Q2',
                    '2014-Q2', '2015-Q2', '2016-Q2',
                    '2017-Q2', '2018-Q2',
                    '2019-Q2', '2020-Q2', '2021-Q2'),
  next_quarter = c( '2010-Q2', '2011-Q2',
                    '2012-Q2', '2013-Q2',
                    '2014-Q2', '2015-Q2', '2016-Q2',
                    '2017-Q2', '2018-Q2',
                    '2019-Q2', '2020-Q2', '2021-Q2',
                    '2022-Q2')
)

# Initialize a ResultsP2Q4list
ResultsP2Q4<- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(bi_quarterly_bottom20, current_quarter)
  ResultsP2Q4[[next_quarter]] <- calculate_performance(Q4_bottom50, top_isins, next_quarter)
}

# Set names for the ResultsP2Q4list to reflect the quarters for which performance was calculated
names(ResultsP2Q4) <- quarters$next_quarter


################3----------Q1-hold-period-forQ4-performance------################

# function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
  top_isins <- data %>%
    filter(Quarter == quarter) %>%
    mutate(lnDeltaP_rank = min_rank(desc(Q4_Change)) / n()) %>%
    filter(lnDeltaP_rank <= 0.20) %>%
    pull(ISIN)
  return(top_isins)
}

# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    summarize(Total_lnDeltaP = sum(Total_Percentage_Change, na.rm = TRUE) / length(isins))
  return(portfolio_performance * 100)  # Result scaled by 100
}


# Define quarter pairs
quarters <- data.frame(
```

```
  start_quarter = c('2009-Q4', '2010-Q1', '2010-Q2',
            '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
            '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
            '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
            '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
            '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
            '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
            '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
            '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
            '2022-Q3'),
  next_quarter = c('2010-Q1', '2010-Q2',
            '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
            '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
            '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
            '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
            '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
            '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
            '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
            '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
            '2022-Q3', '2022-Q4')
)

# Initialize a ResultsP4Q1 list
ResultsP4Q1 <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(Q4_bottom20, current_quarter)
  ResultsP4Q1[[next_quarter]] <- calculate_performance(bottom_50_percent, top_isins,
 next_quarter)
}

# Set names for the ResultsP4Q1 list to reflect the quarters for which performance was calculated
names(ResultsP4Q1) <- quarters$next_quarter


# function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
  top_isins <- data %>%
    filter(Quarter == quarter) %>%
    mutate(lnDeltaP_rank = min_rank(desc(Q4_Change)) / n()) %>%
    filter(lnDeltaP_rank <= 0.20) %>%
    pull(ISIN)
  return(top_isins)
}

calculate_performance <- function(data, isins, quarter) {
  if (length(isins) == 0) {
```

```
  return(data.frame(ISIN = character(), Individual_Performance = numeric()))  # Return empty
 dataframe if no ISINs
 }

 portfolio_performance <- data %>%
  filter(ISIN %in% isins, Quarter == quarter) %>% group_by(ISIN) %>%
  summarize(Total_lnDeltaP = sum(as.numeric(Total_Percentage_Change), na.rm = TRUE) /
length(isins)) %>%
  mutate(Total_lnDeltaP = Total_lnDeltaP * 100)  # Result scaled by 100

 return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3'),
 next_quarter = c('2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3', '2022-Q4')
)

# Initialize a ResultsP4Q1 list
ResultsP4Q1I <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
 current_quarter <- quarters$start_quarter[i]
 next_quarter <- quarters$next_quarter[i]

 top_isins <- get_top_isins(Q4_bottom20, current_quarter)
 ResultsP4Q1I[[next_quarter]] <- calculate_performance(bottom_50_percent, top_isins,
 next_quarter)
}

# Set names for the ResultsP4Q1 list to reflect the quarters for which performance was calculated
names(ResultsP4Q1I) <- quarters$next_quarter
```

```r
################----------Sharp-rate-P4Q1----------################
calculate_sharpe_ratio <- function(returns) {
  average_returns <- mean(returns)
  std_dev_returns <- sd(returns)
  sharpe_ratio <- average_returns / std_dev_returns
  return(sharpe_ratio)
}

# Initialize a list to store Sharpe ratios for each quarter
SharpeRatios <- list()

# Calculate Sharpe ratios using the stored portfolio performances
for (quarter_name in names(ResultsP4Q1I)) {
  # Extract the individual performances as numeric returns
  returns <- as.numeric(ResultsP4Q1I[[quarter_name]]$Total_lnDeltaP)

  # Calculate and store the Sharpe ratio for the quarter
  if (length(returns) > 1 && sd(returns) > 0) {  # Ensure there are enough data points and non-zero SD
    SharpeRatios[[quarter_name]] <- calculate_sharpe_ratio(returns)
  } else {
    SharpeRatios[[quarter_name]] <- NA  # Not enough data to calculate Sharpe ratio
  }
}

# Set names for the Sharpe Ratios list to match the quarters
names(SharpeRatios) <- names(ResultsP4Q1I)



################----------Q2-hold-period-forQ4-performance------################

# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    summarize(Total_lnDeltaP = sum(Bi_Quarterly_Change, na.rm = TRUE) / length(isins))
  return(portfolio_performance * 100)  # Result scaled by 100
}


# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q4', '2010-Q2', '2010-Q4', '2011-Q2',
                    '2011-Q4', '2012-Q2', '2012-Q4', '2013-Q2', '2013-Q4',
                    '2014-Q2', '2014-Q4', '2015-Q2', '2015-Q4', '2016-Q2',
                    '2016-Q4', '2017-Q2', '2017-Q4', '2018-Q2', '2018-Q4',
                    '2019-Q2', '2019-Q4', '2020-Q2', '2020-Q4', '2021-Q2',
                    '2021-Q4', '2022-Q2'),
```

```
  next_quarter = c('2010-Q2', '2010-Q4', '2011-Q2',
                   '2011-Q4', '2012-Q2', '2012-Q4', '2013-Q2', '2013-Q4',
                   '2014-Q2', '2014-Q4', '2015-Q2', '2015-Q4', '2016-Q2',
                   '2016-Q4', '2017-Q2', '2017-Q4', '2018-Q2', '2018-Q4',
                   '2019-Q2', '2019-Q4', '2020-Q2', '2020-Q4', '2021-Q2',
                   '2021-Q4', '2022-Q2', '2022-Q4')
)

# Initialize a ResultsP4Q2list
ResultsP4Q2<- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(Q4_bottom20, current_quarter)
  ResultsP4Q2[[next_quarter]] <- calculate_performance(bi_quarterly_Bottom50, top_isins,
 next_quarter)
}

# Set names for the ResultsP4Q2list to reflect the quarters for which performance was calculated
names(ResultsP4Q2) <- quarters$next_quarter

################3----------Q4-hold-period-forQ4-performance------################


# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    summarize(Total_lnDeltaP = sum(Q4_Change, na.rm = TRUE) / length(isins))
  return(portfolio_performance * 100)  # Result scaled by 100
}


# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q4', '2010-Q4', '2011-Q4',
                    '2012-Q4', '2013-Q4',
                    '2014-Q4', '2015-Q4', '2016-Q4',
                    '2017-Q4', '2018-Q4',
                    '2019-Q4', '2020-Q4', '2021-Q4'),
  next_quarter = c( '2010-Q4', '2011-Q4',
                    '2012-Q4', '2013-Q4',
                    '2014-Q4', '2015-Q4', '2016-Q4',
                    '2017-Q4', '2018-Q4',
                    '2019-Q4', '2020-Q4', '2021-Q4',
                    '2022-Q4')
)

# Initialize a ResultsP4Q4list
```

```
ResultsP4Q4<- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(Q4_bottom20, current_quarter)
  ResultsP4Q4[[next_quarter]] <- calculate_performance(Q4_bottom50, top_isins, next_quarter)
}

# Set names for the ResultsP4Q4list to reflect the quarters for which performance was calculated
names(ResultsP4Q4) <- quarters$next_quarter


results_df <- data.frame(
  Quarter = names(ResultsP4Q1),
  Return = unlist(ResultsP4Q1)
)

# Convert quarters to dates for easier manipulation
results_df$Year <- year(ymd(paste0(results_df$Quarter, "-15")))

# Aggregate quarterly returns to annual returns
annual_results <- results_df %>%
  group_by(Year) %>%
  summarize(Excess_Return = sum(Return), .groups = 'drop')
```

Alfa og SD

```
library(broom)

benchmark_data <- read_excel('Documents/høst 2023/Benchmark.xlsx')

benchmark_data_filtered <- benchmark_data %>%
  select(Date, OBX_adj)

# Merge datasets by Date, adding only OBX_adj to titlon_data
combined_data <- left_join(titlon_data, benchmark_data_filtered, by = "Date")

# Create a Quarter column
titlon_data_C <- combined_data %>%
  mutate(Multiplier_lnDeltaP = 1 + lnDeltaP,
         Multiplier_OBX_adj = OBX_adj,
         Quarter = paste0(year(Date), "-Q", quarter(Date)))

# Compound growth formula for lnDeltaP
# Compound growth for OBX adjusted multiplier
```

```r
# Last observed market cap in the quarter

quarterly_summary <- titlon_data_C %>%
 group_by(ISIN, Quarter) %>%
 arrange(ISIN, Date) %>%
 summarise(
  Total_Percentage_Change = (prod(Multiplier_lnDeltaP, na.rm = TRUE) - 1),
  Return_OBX = (prod(Multiplier_OBX_adj, na.rm = TRUE) - 1),
  mktcap = last(mktcap),
  .groups = 'drop'
 )


# Calculate quarterly SMB, HML, MOM, LIQ by aggregating these values from 'titlon_data_C'
market_factors_summary <- titlon_data_C %>%
 group_by(Quarter) %>%
 summarise(
  Quarterly_SMB = sum(SMB, na.rm = TRUE),  # Sum SMB values by quarter
  Quarterly_HML = sum(HML, na.rm = TRUE),  # Sum HML values by quarter
  Quarterly_MOM = sum(MOM, na.rm = TRUE),  # Sum MOM values by quarter
  Quarterly_LIQ = sum(LIQ, na.rm = TRUE),  # Sum LIQ values by quarter
  .groups = 'drop'
 )

# Merge the market factors back into the quarterly_summary
# This will add the market factors to each ISIN's quarterly data
quarterly_summary_smb <- quarterly_summary %>%
 left_join(market_factors_summary, by = "Quarter")

bottom_20_percent_smb <- quarterly_summary_smb %>%
 group_by(Quarter) %>%
 mutate(market_cap_threshold_20 = quantile(mktcap, 0.2, na.rm = TRUE)) %>%
 filter(mktcap <= market_cap_threshold_20) %>%
 ungroup()


bottom_50_percent_smb <- quarterly_summary_smb %>%
 group_by(Quarter) %>%
 mutate(market_cap_threshold_50 = quantile(mktcap, 0.5, na.rm = TRUE)) %>%
 filter(mktcap <= market_cap_threshold_50) %>%
 ungroup()


# Helper function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
 top_isins <- data %>%
  filter(Quarter == quarter) %>%
  mutate(lnDeltaP_rank = min_rank(desc(Total_Percentage_Change)) / n()) %>%
  filter(lnDeltaP_rank <= 0.20) %>%
  pull(ISIN)
 return(top_isins)
}
```

```r
# Function to calculate the performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
      Total_lnDeltaP = sum(Total_Percentage_Change, na.rm = TRUE),
      Sum_Return_OBX = sum(Return_OBX, na.rm = TRUE),
      Sum_Return_SMB = sum(Quarterly_SMB, na.rm = TRUE),
      Sum_Return_HML = sum(Quarterly_HML, na.rm = TRUE),
      Sum_Return_MOM = sum(Quarterly_MOM, na.rm = TRUE),
      Sum_Return_LIQ = sum(Quarterly_LIQ, na.rm = TRUE),
      .groups = 'drop'
    )
  return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2009-Q2', '2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
            '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
            '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
            '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
            '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
            '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
            '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
            '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
            '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
            '2022-Q3'),
  next_quarter = c('2009-Q2','2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
            '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
            '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
            '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
            '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
            '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
            '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
            '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
            '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
            '2022-Q3', '2022-Q4')
)

# Initialize an empty data frame for results
ResultsP1Q1T<- list()

for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(bottom_20_percent_smb, current_quarter)
  ResultsP1Q1T[[next_quarter]] <- calculate_performance(bottom_50_percent_smb, top_isins,
  next_quarter)
```

```
}

# Set names for the ResultsP1Q1 list to reflect the quarters for which performance was calculated
names(ResultsP1Q1T) <- quarters$next_quarter



ResultsP1Q1T <- do.call(rbind, lapply(names(ResultsP1Q1T), function(x) {
  data.frame(Quarter = x, Performance = ResultsP1Q1T[[x]])
}))


names(ResultsP1Q1T)


average_returns <- ResultsP1Q1T %>%
  summarise(
    Avg_Portfolio_Return = mean(Performance.Total_lnDeltaP, na.rm = TRUE),
    Avg_Benchmark_Return = mean(Performance.Sum_Return_OBX, na.rm = TRUE)
  )

# Calculate Beta using linear regression
regression_model <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data =
  ResultsP1Q1T)
beta <- summary(regression_model)$coefficients[2, 1]  # Extract the beta coefficient


risk_free_rate <- 0.02  #

# Calculate Alpha
alpha <- average_returns$Avg_Portfolio_Return - (risk_free_rate + beta *
  (average_returns$Avg_Benchmark_Return - risk_free_rate))

summary(alpha)

print(paste("Calculated Alpha: ", alpha))


summary(regression_model)

# Get a tidy summary of the model to extract coefficients and their significance
tidy_model <- tidy(regression_model)

# Extract Beta and its p-value
beta <- tidy_model$estimate[2]  # Coefficient for Sum_Return_OBX
beta_p_value <- tidy_model$p.value[2]  # P-value for the beta coefficient

# Display beta and its significance
print(paste("Beta: ", beta))
print(paste("Beta p-value: ", beta_p_value))

alpha <- tidy_model$estimate[1]
```

```
alpha_p_value <- tidy_model$p.value[1]

# Calculate confidence intervals for alpha
conf_int <- confint(regression_model, level = 0.95)  # 95% confidence interval

# Display alpha, its p-value, and confidence intervals
print(paste("Calculated Alpha: ", alpha))
print(paste("Alpha p-value: ", alpha_p_value))
print("Confidence Interval for Alpha:")
print(conf_int[1,])



# Calculate annualized returns and standard deviation
annualized_returns <- mean(ResultsP1Q1$Total_lnDeltaP) * 4  # Assuming quarterly returns
annualized_std_dev <- sd(ResultsP1Q1$Total_lnDeltaP) * sqrt(4)

# Assuming a risk-free rate (e.g., from Treasury bills)
risk_free_rate <- 0.01

# Calculate Sharpe Ratio
sharpe_ratio <- (annualized_returns - risk_free_rate) / annualized_std_dev
print(paste("Sharpe Ratio: ", sharpe_ratio))



regression <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
                 Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
                 Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data = ResultsP1Q1T)
summary(regression)

alfa <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data = ResultsP1Q1T)
summary(alfa)


benchmark_data$Date <- as.Date(benchmark_data$Date)

# Extract year from Date
benchmark_data$Year <- format(benchmark_data$Date, "%Y")

# Function to calculate compounded annual growth rate
compounded_return <- function(returns) {
  compounded = prod(1 + returns ) - 1
  return(compounded )  # Convert back to percentage
}

# Group by Year and calculate the compounded annual return
annual_returns <- benchmark_data %>%
  group_by(Year) %>%
  summarize(Yearly_Return = compounded_return(OBX_pure), .groups = 'drop')
```

```
# View the compounded annual returns
print(annual_returns)

write.xlsx(annual_returns, file = "Documents/høst 2023/annual_returns.xlsx", rowNames = FALSE)



#####################----------------#######################

ResultsP1Q2U

# Extract year from Quarter
ResultsP1Q2U$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q2U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP1Q2 <- ResultsP1Q2U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# If you need to convert log returns to simple percentage returns:
# annual_returns$Annual_Percent_Return <- (exp(annual_log_returns$Annual_Log_Return) - 1) *
  100
ResultsP1Q1U

# Extract year from Quarter
ResultsP1Q1U$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q1U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP1Q1 <- ResultsP1Q1U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

ResultsP1Q4U

ResultsP1Q4U <- ResultsP1Q4U %>%
  mutate(Total_lnDeltaP = Total_lnDeltaP / 100)

# Extract year from Quarter
ResultsP1Q4U$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q4U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP1Q4 <- ResultsP1Q4U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')


# View the summed annual log returns
print(annual_log_returnsP1Q2)
print(annual_log_returnsP1Q1)
print(annual_log_returnsP1Q4)

sd_value_P1Q1 <- sd(annual_log_returnsP1Q1$Annual_Log_Return)
```

```
sd_value_P1Q2 <- sd(annual_log_returnsP1Q2$Annual_Log_Return)
sd_value_P1Q4 <- sd(annual_log_returnsP1Q4$Annual_Log_Return)
sd_value_Bench <- sd(annual_returns$Yearly_Return)


# Print the standard deviation
print(sd_value_P1Q1)
print(sd_value_P1Q2)
print(sd_value_P1Q4)
print(sd_value_Bench)

average_return_obx <- mean(annual_returns$Yearly_Return)
average_return_P1Q1 <- mean(annual_log_returnsP1Q1$Annual_Log_Return)
average_return_P1Q2 <- mean(annual_log_returnsP1Q2$Annual_Log_Return)
average_return_P1Q4 <- mean(annual_log_returnsP1Q4$Annual_Log_Return)

print(average_return_obx)
print(average_return_P1Q1)
print(average_return_P1Q2)
print(average_return_P1Q4)

# Calculate Sharpe Ratio
sharpe_ratio_obx <- (average_return_obx) / sd_value_Bench
sharpe_ratio_P1Q1 <- (average_return_P1Q1) / sd_value_P1Q1
sharpe_ratio_P1Q2 <- (average_return_P1Q2) / sd_value_P1Q2
sharpe_ratio_P1Q4 <- (average_return_P1Q4) / sd_value_P1Q4

print(sharpe_ratio_obx)
print(sharpe_ratio_P1Q1)
print(sharpe_ratio_P1Q2)
print(sharpe_ratio_P1Q4)

ResultsP2Q2U$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q2U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP2Q2 <- ResultsP2Q2U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP2Q1U$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q1U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP2Q1 <- ResultsP2Q1U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP2Q4U$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q4U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP2Q4 <- ResultsP2Q4U %>%
```

```
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

sd_value_P2Q1 <- sd(annual_log_returnsP2Q1$Annual_Log_Return)
sd_value_P2Q2 <- sd(annual_log_returnsP2Q2$Annual_Log_Return)
sd_value_P2Q4 <- sd(annual_log_returnsP2Q4$Annual_Log_Return)

# View the summed annual log returns
print(annual_log_returnsP2Q2)
print(annual_log_returnsP2Q1)
print(annual_log_returnsP2Q4)

# Print the standard deviation
print(sd_value_P2Q1)
print(sd_value_P2Q2)
print(sd_value_P2Q4)

average_return_P2Q1 <- mean(annual_log_returnsP2Q1$Annual_Log_Return)
average_return_P2Q2 <- mean(annual_log_returnsP2Q2$Annual_Log_Return)
average_return_P2Q4 <- mean(annual_log_returnsP2Q4$Annual_Log_Return)

# Calculate Sharpe Ratio
sharpe_ratio_P2Q1 <- (average_return_P2Q1) / sd_value_P2Q1
sharpe_ratio_P2Q2 <- (average_return_P2Q2) / sd_value_P2Q2
sharpe_ratio_P2Q4 <- (average_return_P2Q4) / sd_value_P2Q4

print(sharpe_ratio_P2Q1)
print(sharpe_ratio_P2Q2)
print(sharpe_ratio_P2Q4)

ResultsP4Q2U$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q2U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP4Q2 <- ResultsP4Q2U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP4Q1U$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q1U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP4Q1 <- ResultsP4Q1U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP4Q4U$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q4U$Quarter))

# Group by Year and sum the log returns
annual_log_returnsP4Q4 <- ResultsP4Q4U %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')
```

```
sd_value_P4Q1 <- sd(annual_log_returnsP4Q1$Annual_Log_Return)
sd_value_P4Q2 <- sd(annual_log_returnsP4Q2$Annual_Log_Return)
sd_value_P4Q4 <- sd(annual_log_returnsP4Q4$Annual_Log_Return)

# View the summed annual log returns
print(annual_log_returnsP1Q2)
print(annual_log_returnsP1Q1)
print(annual_log_returnsP1Q4)

# Print the standard deviation
print(sd_value_P4Q1)
print(sd_value_P4Q2)
print(sd_value_P4Q4)

average_return_P4Q1 <- mean(annual_log_returnsP4Q1$Annual_Log_Return)
average_return_P4Q2 <- mean(annual_log_returnsP4Q2$Annual_Log_Return)
average_return_P4Q4 <- mean(annual_log_returnsP4Q4$Annual_Log_Return)

# Calculate Sharpe Ratio
sharpe_ratio_P4Q1 <- (average_return_P4Q1) / sd_value_P4Q1
sharpe_ratio_P4Q2 <- (average_return_P4Q2) / sd_value_P4Q2
sharpe_ratio_P4Q4 <- (average_return_P4Q4) / sd_value_P4Q4

print(sharpe_ratio_P4Q1)
print(sharpe_ratio_P4Q2)
print(sharpe_ratio_P4Q4)
```

Alfa og faktormodell

```
#####-----crete-data-with-risk-factors-----######

bottom_20_percent_smb <- quarterly_summary_smb %>%
  group_by(Quarter) %>%
  mutate(market_cap_threshold_20 = quantile(mktcap, 0.2, na.rm = TRUE)) %>%
  filter(mktcap <= market_cap_threshold_20) %>%
  ungroup()


bottom_50_percent_smb <- quarterly_summary_smb %>%
  group_by(Quarter) %>%
  mutate(market_cap_threshold_50 = quantile(mktcap, 0.5, na.rm = TRUE)) %>%
  filter(mktcap <= market_cap_threshold_50) %>%
  ungroup()


bi_quarterly_data_smb <- bottom_50_percent_smb

bi_quarterly_data_smb <- bi_quarterly_data_smb %>%
  arrange(ISIN, Quarter) %>%
```

```
  mutate(Cumulative_Change = Total_Percentage_Change + 1) %>%
  group_by(ISIN) %>%
  mutate(Bi_Quarterly_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change) %>%
  ungroup()

bi_quarterly_data_smb <- bi_quarterly_data_smb %>%
  filter(Quarter != '2009-Q2')
bi_quarterly_data_smb <- bi_quarterly_data_smb %>%
  filter(Quarter != '2009-Q1')

# Adjust the data changes back to percentages
bi_quarterly_Bottom50_smb <- bi_quarterly_data_smb %>%
  mutate(Bi_Quarterly_Change = (Bi_Quarterly_Change - 1))

bi_quarterly_Bottom50_smb <- bi_quarterly_Bottom50_smb %>%
  arrange(ISIN, Quarter) %>%
  group_by(ISIN) %>%
  mutate(Bi_Quarterly_Sum_OBX = lag(Return_OBX+1, default = 1) * Return_OBX,
         Bi_Quarterly_SMB = lag(Quarterly_SMB, default = 0) + Quarterly_SMB,
         Bi_Quarterly_HML = lag(Quarterly_HML, default = 0) + Quarterly_HML,
         Bi_Quarterly_MOM = lag(Quarterly_MOM, default = 0) + Quarterly_MOM,
         Bi_Quarterly_LIQ = lag(Quarterly_LIQ, default = 0) + Quarterly_LIQ) %>%
  ungroup()



bi_quarterly_bottom20_smb <- bottom_20_percent_smb

bi_quarterly_bottom20_smb <- bi_quarterly_bottom20_smb %>%
  arrange(ISIN, Quarter) %>%
  mutate(Cumulative_Change = Total_Percentage_Change + 1) %>%
  group_by(ISIN) %>%
  mutate(Bi_Quarterly_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change) %>%
  ungroup()

bi_quarterly_bottom20_smb <- bi_quarterly_bottom20_smb %>%
  filter(Quarter != '2009-Q1')

# remove unnecessary rows and back to percentages
bi_quarterly_bottom20_smb <- bi_quarterly_bottom20_smb %>%
  mutate(Bi_Quarterly_Change = (Bi_Quarterly_Change - 1))

bi_quarterly_bottom20_smb <- bi_quarterly_bottom20_smb %>%
  arrange(ISIN, Quarter) %>%
  group_by(ISIN) %>%
  mutate(Bi_Quarterly_Sum_OBX = lag(Return_OBX, default = 0) + Return_OBX,
         Bi_Quarterly_SMB = lag(Quarterly_SMB, default = 0) + Quarterly_SMB,
         Bi_Quarterly_HML = lag(Quarterly_HML, default = 0) + Quarterly_HML,
         Bi_Quarterly_MOM = lag(Quarterly_MOM, default = 0) + Quarterly_MOM,
         Bi_Quarterly_LIQ = lag(Quarterly_LIQ, default = 0) + Quarterly_LIQ) %>%
  ungroup()
```

```
##############-----------Crete-data-in-per-Q4-form----------###############


Q4_bottom20_smb <- bottom_20_percent_smb

Q4_bottom20_smb <- Q4_bottom20_smb %>%
 arrange(ISIN, Quarter) %>%
 mutate(Cumulative_Change = Total_Percentage_Change + 1, Return_OBX = Return_OBX + 1) %>%
 group_by(ISIN) %>%
 mutate(Q4_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change,
  Q4_Change = lag(Cumulative_Change, n = 2, default = 1) * Q4_Change,
  Q4_Change = lag(Cumulative_Change, n = 3, default = 1) * Q4_Change) %>%
 ungroup()

Q4_bottom20_smb <- Q4_bottom20_smb %>%
 filter(Quarter != '2009-Q2')
Q4_bottom20_smb <- Q4_bottom20_smb %>%
 filter(Quarter != '2009-Q1')
Q4_bottom20_smb <- Q4_bottom20_smb %>%
 filter(Quarter != '2009-Q3')

# remove unnecessary rows and back to percentages
Q4_bottom20_smb <- Q4_bottom20_smb %>%
 mutate(Q4_Change = (Q4_Change - 1))

names(Q4_bottom20_smb)

Q4_bottom20_smb <- Q4_bottom20_smb %>%
 arrange(ISIN, Quarter) %>%
 group_by(ISIN) %>%
 mutate(
  lag1_Return_OBX = lag(Return_OBX, n = 1, default = 0),
  lag2_Return_OBX = lag(Return_OBX, n = 2, default = 0),
  lag3_Return_OBX = lag(Return_OBX, n = 3, default = 0),
  Q4_Sum_Return_OBX = lag1_Return_OBX * lag2_Return_OBX * lag3_Return_OBX * Return_OBX,

  lag1_Quarterly_SMB = lag(Quarterly_SMB, n = 1, default = 0),
  lag2_Quarterly_SMB = lag(Quarterly_SMB, n = 2, default = 0),
  lag3_Quarterly_SMB = lag(Quarterly_SMB, n = 3, default = 0),
  Q4_Sum_Quarterly_SMB = lag1_Quarterly_SMB + lag2_Quarterly_SMB + lag3_Quarterly_SMB +
Quarterly_SMB,

  lag1_Quarterly_HML = lag(Quarterly_HML, n = 1, default = 0),
  lag2_Quarterly_HML = lag(Quarterly_HML, n = 2, default = 0),
  lag3_Quarterly_HML = lag(Quarterly_HML, n = 3, default = 0),
  Q4_Sum_Quarterly_HML = lag1_Quarterly_HML + lag2_Quarterly_HML + lag3_Quarterly_HML +
Quarterly_HML,

  lag1_Quarterly_MOM = lag(Quarterly_MOM, n = 1, default = 0),
  lag2_Quarterly_MOM = lag(Quarterly_MOM, n = 2, default = 0),
  lag3_Quarterly_MOM = lag(Quarterly_MOM, n = 3, default = 0),
```

```
  Q4_Sum_Quarterly_MOM = lag1_Quarterly_MOM + lag2_Quarterly_MOM +
 lag3_Quarterly_MOM + Quarterly_MOM,

  lag1_Quarterly_LIQ = lag(Quarterly_LIQ, n = 1, default = 0),
  lag2_Quarterly_LIQ = lag(Quarterly_LIQ, n = 2, default = 0),
  lag3_Quarterly_LIQ = lag(Quarterly_LIQ, n = 3, default = 0),
  Q4_Sum_Quarterly_LIQ = lag1_Quarterly_LIQ + lag2_Quarterly_LIQ + lag3_Quarterly_LIQ +
 Quarterly_LIQ
 ) %>%
 ungroup()

###-Q4-bottom-50-###

Q4_bottom50_smb <- bottom_50_percent_smb

Q4_bottom50_smb <- Q4_bottom50_smb %>%
 arrange(ISIN, Quarter) %>%
 mutate(Cumulative_Change = Total_Percentage_Change + 1,  Return_OBX = Return_OBX + 1) %>%
 group_by(ISIN) %>%
 mutate(Q4_Change = lag(Cumulative_Change, default = 1) * Cumulative_Change,
  Q4_Change = lag(Cumulative_Change, n = 2, default = 1) * Q4_Change,
  Q4_Change = lag(Cumulative_Change, n = 3, default = 1) * Q4_Change) %>%
 ungroup()

Q4_bottom50_smb <- Q4_bottom50_smb %>%
 filter(Quarter != '2009-Q2')
Q4_bottom50_smb <- Q4_bottom50_smb %>%
 filter(Quarter != '2009-Q1')
Q4_bottom50_smb <- Q4_bottom50_smb %>%
 filter(Quarter != '2009-Q3')

# remove unnecessary rows and back to percentages
Q4_bottom50_smb <- Q4_bottom50_smb %>%
 mutate(Q4_Change = (Q4_Change - 1))


Q4_bottom50_smb <- Q4_bottom50_smb %>%
 arrange(ISIN, Quarter) %>%
 group_by(ISIN) %>%
 mutate(
  lag1_Return_OBX = lag(Return_OBX, n = 1, default = 0),
  lag2_Return_OBX = lag(Return_OBX, n = 2, default = 0),
  lag3_Return_OBX = lag(Return_OBX, n = 3, default = 0),
  Q4_Sum_Return_OBX = lag1_Return_OBX * lag2_Return_OBX * lag3_Return_OBX * Return_OBX,

  lag1_Quarterly_SMB = lag(Quarterly_SMB, n = 1, default = 0),
  lag2_Quarterly_SMB = lag(Quarterly_SMB, n = 2, default = 0),
  lag3_Quarterly_SMB = lag(Quarterly_SMB, n = 3, default = 0),
  Q4_Sum_Quarterly_SMB = lag1_Quarterly_SMB + lag2_Quarterly_SMB + lag3_Quarterly_SMB +
 Quarterly_SMB,

  lag1_Quarterly_HML = lag(Quarterly_HML, n = 1, default = 0),
```

```
  lag2_Quarterly_HML = lag(Quarterly_HML, n = 2, default = 0),
  lag3_Quarterly_HML = lag(Quarterly_HML, n = 3, default = 0),
  Q4_Sum_Quarterly_HML = lag1_Quarterly_HML + lag2_Quarterly_HML + lag3_Quarterly_HML +
 Quarterly_HML,

  lag1_Quarterly_MOM = lag(Quarterly_MOM, n = 1, default = 0),
  lag2_Quarterly_MOM = lag(Quarterly_MOM, n = 2, default = 0),
  lag3_Quarterly_MOM = lag(Quarterly_MOM, n = 3, default = 0),
  Q4_Sum_Quarterly_MOM = lag1_Quarterly_MOM + lag2_Quarterly_MOM +
 lag3_Quarterly_MOM + Quarterly_MOM,

  lag1_Quarterly_LIQ = lag(Quarterly_LIQ, n = 1, default = 0),
  lag2_Quarterly_LIQ = lag(Quarterly_LIQ, n = 2, default = 0),
  lag3_Quarterly_LIQ = lag(Quarterly_LIQ, n = 3, default = 0),
  Q4_Sum_Quarterly_LIQ = lag1_Quarterly_LIQ + lag2_Quarterly_LIQ + lag3_Quarterly_LIQ +
 Quarterly_LIQ
 ) %>%
 ungroup()
```

```
# function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
 top_isins <- data %>%
  filter(Quarter == quarter) %>%
  mutate(lnDeltaP_rank = min_rank(desc(Total_Percentage_Change)) / n()) %>%
  filter(lnDeltaP_rank <= 0.20) %>%
  pull(ISIN)
 return(top_isins)
}

# Function to calculate the performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
 portfolio_performance <- data %>%
  filter(ISIN %in% isins, Quarter == quarter) %>%
  group_by(ISIN) %>%
  summarize(
   Total_lnDeltaP = sum(Total_Percentage_Change, na.rm = TRUE)/length(isins),
   Sum_Return_OBX = sum(Return_OBX, na.rm = TRUE)/length(isins),
   Sum_Return_SMB = sum(Quarterly_SMB, na.rm = TRUE)/length(isins),
   Sum_Return_HML = sum(Quarterly_HML, na.rm = TRUE)/length(isins),
   Sum_Return_MOM = sum(Quarterly_MOM, na.rm = TRUE)/length(isins),
   Sum_Return_LIQ = sum(Quarterly_LIQ, na.rm = TRUE)/length(isins),
   .groups = 'drop'
  )
 return(portfolio_performance)
}
```

```
# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2009-Q2', '2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
            '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
            '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
            '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
            '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
            '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
            '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
            '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
            '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
            '2022-Q3'),
  next_quarter = c('2009-Q2','2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
            '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
            '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
            '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
            '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
            '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
            '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
            '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
            '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
            '2022-Q3', '2022-Q4')
)

# Initialize an empty data frame for results
ResultsP1Q1T<- list()

for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(bottom_20_percent_smb, current_quarter)
  ResultsP1Q1T[[next_quarter]] <- calculate_performance(bottom_50_percent_smb, top_isins,
  next_quarter)
}

# Set names for the ResultsP1Q1 list to reflect the quarters for which performance was calculated
names(ResultsP1Q1T) <- quarters$next_quarter



ResultsP1Q1T <- do.call(rbind, lapply(names(ResultsP1Q1T), function(x) {
  data.frame(Quarter = x, Performance = ResultsP1Q1T[[x]])
}))


names(ResultsP1Q1T)


regression_P1Q1 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
              Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
              Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data = ResultsP1Q1T)
```

```
summary(regression_P1Q1)

alfa_P1Q1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP1Q1T)
summary(alfa_P1Q1)



# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
      Total_lnDeltaP = sum(Bi_Quarterly_Change, na.rm = TRUE)/length(isins),
      Sum_Return_OBX = sum(Bi_Quarterly_Sum_OBX, na.rm = TRUE)/length(isins),
      Sum_Return_SMB = sum(Bi_Quarterly_SMB, na.rm = TRUE)/length(isins),
      Sum_Return_HML = sum(Bi_Quarterly_HML, na.rm = TRUE)/length(isins),
      Sum_Return_MOM = sum(Bi_Quarterly_MOM, na.rm = TRUE)/length(isins),
      Sum_Return_LIQ = sum(Bi_Quarterly_LIQ, na.rm = TRUE)/length(isins),
      .groups = 'drop'
    )
  return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                    '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                    '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                    '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                    '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                    '2021-Q3', '2022-Q1'),
  next_quarter = c('2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                   '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                   '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                   '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                   '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                   '2021-Q3', '2022-Q1', '2022-Q3')
)

# Initialize a resultsQ2 list
ResultsP1Q2T <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(bottom_20_percent_smb, current_quarter)
  ResultsP1Q2T[[next_quarter]] <- calculate_performance(bi_quarterly_Bottom50_smb, top_isins,
 next_quarter)
}
```

```
# Set names for the resultsQ2 list to reflect the quarters for which performance was calculated
names(ResultsP1Q2T) <- quarters$next_quarter

ResultsP1Q2T <- do.call(rbind, lapply(names(ResultsP1Q2T), function(x) {
  data.frame(Quarter = x, Performance = ResultsP1Q2T[[x]])
}))


regression_P1Q2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
                Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
                Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data = ResultsP1Q2T)
summary(regression_P1Q2)

alfa_P1Q2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data = ResultsP1Q2T)
summary(alfa_P1Q2)



# Function to calculate the portfolio performance for a given list of ISINs and quarter
calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
      Total_lnDeltaP = sum(Q4_Change, na.rm = TRUE)/length(isins),
      Sum_Return_OBX = sum(Q4_Sum_Return_OBX, na.rm = TRUE)/length(isins),
      Sum_Return_SMB = sum(Q4_Sum_Quarterly_SMB, na.rm = TRUE)/length(isins),
      Sum_Return_HML = sum(Q4_Sum_Quarterly_HML, na.rm = TRUE)/length(isins),
      Sum_Return_MOM = sum(Q4_Sum_Quarterly_MOM, na.rm = TRUE)/length(isins),
      Sum_Return_LIQ = sum(Q4_Sum_Quarterly_LIQ, na.rm = TRUE)/length(isins),
      .groups = 'drop'
    )
  return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2010-Q1', '2011-Q1',
                '2012-Q1', '2013-Q1',
                '2014-Q1', '2015-Q1', '2016-Q1',
                '2017-Q1', '2018-Q1',
                '2019-Q1', '2020-Q1', '2021-Q1'),
  next_quarter = c( '2010-Q1', '2011-Q1',
                '2012-Q1', '2013-Q1',
                '2014-Q1', '2015-Q1', '2016-Q1',
                '2017-Q1', '2018-Q1',
                '2019-Q1', '2020-Q1', '2021-Q1',
                '2022-Q1')
)

# Initialize a resultsQ2 list
```

```
ResultsP1Q4T <- list()

# Loop through the quarters, calculating top ISINs and performance for each quarter pair
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(bottom_20_percent_smb, current_quarter)
  ResultsP1Q4T[[next_quarter]] <- calculate_performance(Q4_bottom50_smb, top_isins,
 next_quarter)
}

# Set names for the resultsQ2 list to reflect the quarters for which performance was calculated
names(ResultsP1Q4T) <- quarters$next_quarter

ResultsP1Q4T <- do.call(rbind, lapply(names(ResultsP1Q4T), function(x) {
  data.frame(Quarter = x, Performance = ResultsP1Q4T[[x]])
}))


regression_P1Q4 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
                Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
                Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data = ResultsP1Q4T)
summary(regression_P1Q4)

alfa_P1Q4 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data = ResultsP1Q4T)
summary(alfa_P1Q4)




###################-----------P2----------##################



get_top_isins <- function(data, quarter) {
  top_isins <- data %>%
    filter(Quarter == quarter) %>%
    mutate(lnDeltaP_rank = min_rank(desc(Bi_Quarterly_Change)) / n()) %>%
    filter(lnDeltaP_rank <= 0.20) %>%
    pull(ISIN)
  print(paste("Quarter:", quarter, "ISINs:", length(top_isins)))
  return(top_isins)
}

calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
```

```
    Total_lnDeltaP = sum(Total_Percentage_Change, na.rm = TRUE),
    Sum_Return_OBX = sum(Return_OBX, na.rm = TRUE),
    Sum_Return_SMB = sum(Quarterly_SMB, na.rm = TRUE),
    Sum_Return_HML = sum(Quarterly_HML, na.rm = TRUE),
    Sum_Return_MOM = sum(Quarterly_MOM, na.rm = TRUE),
    Sum_Return_LIQ = sum(Quarterly_LIQ, na.rm = TRUE),
    .groups = 'drop'
   )
 print(paste("Quarter:", quarter, "Performance rows:", nrow(portfolio_performance)))
 return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q1', '2009-Q2', '2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3'),
 next_quarter = c('2009-Q2','2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3', '2022-Q4')
)


ResultsP2Q1T <- list()
for (i in 1:nrow(quarters)) {
 current_quarter <- quarters$start_quarter[i]
 next_quarter <- quarters$next_quarter[i]

 top_isins <- get_top_isins(bi_quarterly_bottom20_smb, current_quarter)
 ResultsP2Q1T[[next_quarter]] <- calculate_performance(bottom_50_percent_smb, top_isins,
 next_quarter)
}
names(ResultsP2Q1T) <- quarters$next_quarter

ResultsP2Q1T <- ResultsP2Q1T[lapply(ResultsP2Q1T, nrow) > 0]

ResultsP2Q1T <- do.call(rbind, lapply(names(ResultsP2Q1T), function(x) {
 data.frame(Quarter = x, ResultsP2Q1T[[x]])
```

```
}))

regression_P2Q1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
                Sum_Return_SMB + Sum_Return_HML +
                Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q1T)
summary(regression_P2Q1)

alfa_P2Q1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q1T)
summary(alfa_P2Q1)




calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
      Total_lnDeltaP = sum(Bi_Quarterly_Change, na.rm = TRUE),
      Sum_Return_OBX = sum(Bi_Quarterly_Sum_OBX, na.rm = TRUE),
      Sum_Return_SMB = sum(Bi_Quarterly_SMB, na.rm = TRUE),
      Sum_Return_HML = sum(Bi_Quarterly_HML, na.rm = TRUE),
      Sum_Return_MOM = sum(Bi_Quarterly_MOM, na.rm = TRUE),
      Sum_Return_LIQ = sum(Bi_Quarterly_LIQ, na.rm = TRUE),
      .groups = 'drop'
    )
  print(paste("Quarter:", quarter, "Performance rows:", nrow(portfolio_performance)))
  return(portfolio_performance)
}


# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                    '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                    '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                    '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                    '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                    '2021-Q3', '2022-Q1'),
  next_quarter = c('2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                   '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                   '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                   '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                   '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                   '2021-Q3', '2022-Q1', '2022-Q3')
)


ResultsP2Q2T <- list()
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]
```

```
 top_isins <- get_top_isins(bi_quarterly_bottom20_smb, current_quarter)
 ResultsP2Q2T[[next_quarter]] <- calculate_performance(bi_quarterly_Bottom50_smb, top_isins,
 next_quarter)
}
names(ResultsP2Q2T) <- quarters$next_quarter

ResultsP2Q2T <- ResultsP2Q2T[lapply(ResultsP2Q2T, nrow) > 0]

ResultsP2Q2T <- do.call(rbind, lapply(names(ResultsP2Q2T), function(x) {
 data.frame(Quarter = x, ResultsP2Q2T[[x]])
}))

regression_P2Q2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
                Sum_Return_SMB + Sum_Return_HML +
                Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q2T)
summary(regression_P2Q2)

alfa_P2Q2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q2T)
summary(alfa_P2Q2)



calculate_performance <- function(data, isins, quarter) {
 portfolio_performance <- data %>%
  filter(ISIN %in% isins, Quarter == quarter) %>%
  group_by(ISIN) %>%
  summarize(
   Total_lnDeltaP = sum(Q4_Change, na.rm = TRUE),
   Sum_Return_OBX = sum(Q4_Sum_Return_OBX, na.rm = TRUE),
   Sum_Return_SMB = sum(Q4_Sum_Quarterly_SMB, na.rm = TRUE),
   Sum_Return_HML = sum(Q4_Sum_Quarterly_HML, na.rm = TRUE),
   Sum_Return_MOM = sum(Q4_Sum_Quarterly_MOM, na.rm = TRUE),
   Sum_Return_LIQ = sum(Q4_Sum_Quarterly_LIQ, na.rm = TRUE),
   .groups = 'drop'
  )
 print(paste("Quarter:", quarter, "Performance rows:", nrow(portfolio_performance)))
 return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q1', '2010-Q1', '2011-Q1',
                '2012-Q1', '2013-Q1',
                '2014-Q1', '2015-Q1', '2016-Q1',
                '2017-Q1', '2018-Q1',
                '2019-Q1', '2020-Q1', '2021-Q1'),
 next_quarter = c( '2010-Q1', '2011-Q1',
                '2012-Q1', '2013-Q1',
                '2014-Q1', '2015-Q1', '2016-Q1',
                '2017-Q1', '2018-Q1',
                '2019-Q1', '2020-Q1', '2021-Q1',
```

```
                '2022-Q1')
)

ResultsP2Q4T <- list()
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(bi_quarterly_bottom20_smb, current_quarter)
  ResultsP2Q4T[[next_quarter]] <- calculate_performance(Q4_bottom50_smb, top_isins,
 next_quarter)
}
names(ResultsP2Q4T) <- quarters$next_quarter

ResultsP2Q4T <- ResultsP2Q4T[lapply(ResultsP2Q4T, nrow) > 0]

ResultsP2Q4T <- do.call(rbind, lapply(names(ResultsP2Q4T), function(x) {
  data.frame(Quarter = x, ResultsP2Q4T[[x]])
}))

regression_P2Q4 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
                Sum_Return_SMB + Sum_Return_HML +
                Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q4T)
summary(regression_P2Q4)

alfa_P2Q4 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q4T)
summary(alfa_P2Q4)

###################------------P4----------------###################

# function to retrieve top 20% ISINs for a given quarter
get_top_isins <- function(data, quarter) {
  top_isins <- data %>%
    filter(Quarter == quarter) %>%
    mutate(lnDeltaP_rank = min_rank(desc(Q4_Change)) / n()) %>%
    filter(lnDeltaP_rank <= 0.20) %>%
    pull(ISIN)
  return(top_isins)
}

calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
      Total_lnDeltaP = sum(Total_Percentage_Change, na.rm = TRUE),
      Sum_Return_OBX = sum(Return_OBX, na.rm = TRUE),
      Sum_Return_SMB = sum(Quarterly_SMB, na.rm = TRUE),
      Sum_Return_HML = sum(Quarterly_HML, na.rm = TRUE),
      Sum_Return_MOM = sum(Quarterly_MOM, na.rm = TRUE),
      Sum_Return_LIQ = sum(Quarterly_LIQ, na.rm = TRUE),
      .groups = 'drop'
```

```
  )
 print(paste("Quarter:", quarter, "Performance rows:", nrow(portfolio_performance)))
 return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
 start_quarter = c('2009-Q1', '2009-Q2', '2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3'),
 next_quarter = c('2009-Q2','2009-Q3', '2009-Q4', '2010-Q1', '2010-Q2',
          '2010-Q3', '2010-Q4', '2011-Q1', '2011-Q2', '2011-Q3', '2011-Q4',
          '2012-Q1', '2012-Q2', '2012-Q3', '2012-Q4', '2013-Q1', '2013-Q2',
          '2013-Q3', '2013-Q4', '2014-Q1', '2014-Q2', '2014-Q3', '2014-Q4',
          '2015-Q1', '2015-Q2', '2015-Q3', '2015-Q4', '2016-Q1', '2016-Q2',
          '2016-Q3', '2016-Q4', '2017-Q1', '2017-Q2', '2017-Q3', '2017-Q4',
          '2018-Q1', '2018-Q2', '2018-Q3', '2018-Q4', '2019-Q1', '2019-Q2',
          '2019-Q3', '2019-Q4', '2020-Q1', '2020-Q2', '2020-Q3', '2020-Q4',
          '2021-Q1', '2021-Q2', '2021-Q3', '2021-Q4', '2022-Q1', '2022-Q2',
          '2022-Q3', '2022-Q4')
)

ResultsP4Q1T <- list()
for (i in 1:nrow(quarters)) {
 current_quarter <- quarters$start_quarter[i]
 next_quarter <- quarters$next_quarter[i]

 top_isins <- get_top_isins(Q4_bottom20_smb, current_quarter)
 ResultsP4Q1T[[next_quarter]] <- calculate_performance(bottom_50_percent_smb, top_isins,
 next_quarter)
}
names(ResultsP4Q1T) <- quarters$next_quarter

ResultsP4Q1T <- ResultsP4Q1T[lapply(ResultsP4Q1T, nrow) > 0]

ResultsP4Q1T <- do.call(rbind, lapply(names(ResultsP4Q1T), function(x) {
 data.frame(Quarter = x, ResultsP4Q1T[[x]])
}))

regression_P4Q1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
          Sum_Return_SMB + Sum_Return_HML +
          Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q1T)
summary(regression_P4Q1)

alfa_P4Q1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q1T)
```

```
summary(alfa_P4Q1)



calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
      Total_lnDeltaP = sum(Bi_Quarterly_Change, na.rm = TRUE),
      Sum_Return_OBX = sum(Bi_Quarterly_Sum_OBX, na.rm = TRUE),
      Sum_Return_SMB = sum(Bi_Quarterly_SMB, na.rm = TRUE),
      Sum_Return_HML = sum(Bi_Quarterly_HML, na.rm = TRUE),
      Sum_Return_MOM = sum(Bi_Quarterly_MOM, na.rm = TRUE),
      Sum_Return_LIQ = sum(Bi_Quarterly_LIQ, na.rm = TRUE),
      .groups = 'drop'
    )
  print(paste("Quarter:", quarter, "Performance rows:", nrow(portfolio_performance)))
  return(portfolio_performance)
}


# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                    '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                    '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                    '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                    '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                    '2021-Q3', '2022-Q1'),
  next_quarter = c('2009-Q3', '2010-Q1', '2010-Q3', '2011-Q1',
                   '2011-Q3', '2012-Q1', '2012-Q3', '2013-Q1', '2013-Q3',
                   '2014-Q1', '2014-Q3', '2015-Q1', '2015-Q3', '2016-Q1',
                   '2016-Q3', '2017-Q1', '2017-Q3', '2018-Q1', '2018-Q3',
                   '2019-Q1', '2019-Q3', '2020-Q1', '2020-Q3', '2021-Q1',
                   '2021-Q3', '2022-Q1', '2022-Q3')
)

ResultsP4Q2T <- list()
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(Q4_bottom20_smb, current_quarter)
  ResultsP4Q2T[[next_quarter]] <- calculate_performance(bi_quarterly_Bottom50_smb, top_isins,
  next_quarter)
}
names(ResultsP4Q2T) <- quarters$next_quarter

ResultsP4Q2T <- ResultsP4Q2T[lapply(ResultsP4Q2T, nrow) > 0]

ResultsP4Q2T <- do.call(rbind, lapply(names(ResultsP4Q2T), function(x) {
```

```
  data.frame(Quarter = x, ResultsP4Q2T[[x]])
}))

regression_P4Q2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
                Sum_Return_SMB + Sum_Return_HML +
                Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q2T)
summary(regression_P4Q2)

alfa_P4Q2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q2T)
summary(alfa_P4Q2)



calculate_performance <- function(data, isins, quarter) {
  portfolio_performance <- data %>%
    filter(ISIN %in% isins, Quarter == quarter) %>%
    group_by(ISIN) %>%
    summarize(
      Total_lnDeltaP = sum(Q4_Change, na.rm = TRUE),
      Sum_Return_OBX = sum(Q4_Sum_Return_OBX, na.rm = TRUE),
      Sum_Return_SMB = sum(Q4_Sum_Quarterly_SMB, na.rm = TRUE),
      Sum_Return_HML = sum(Q4_Sum_Quarterly_HML, na.rm = TRUE),
      Sum_Return_MOM = sum(Q4_Sum_Quarterly_MOM, na.rm = TRUE),
      Sum_Return_LIQ = sum(Q4_Sum_Quarterly_LIQ, na.rm = TRUE),
      .groups = 'drop'
    )
  print(paste("Quarter:", quarter, "Performance rows:", nrow(portfolio_performance)))
  return(portfolio_performance)
}

# Define quarter pairs
quarters <- data.frame(
  start_quarter = c('2009-Q1', '2010-Q1', '2011-Q1',
                '2012-Q1', '2013-Q1',
                '2014-Q1', '2015-Q1', '2016-Q1',
                '2017-Q1', '2018-Q1',
                '2019-Q1', '2020-Q1', '2021-Q1'),
  next_quarter = c( '2010-Q1', '2011-Q1',
                '2012-Q1', '2013-Q1',
                '2014-Q1', '2015-Q1', '2016-Q1',
                '2017-Q1', '2018-Q1',
                '2019-Q1', '2020-Q1', '2021-Q1',
                '2022-Q1')
)

ResultsP4Q4T <- list()
for (i in 1:nrow(quarters)) {
  current_quarter <- quarters$start_quarter[i]
  next_quarter <- quarters$next_quarter[i]

  top_isins <- get_top_isins(Q4_bottom20_smb, current_quarter)
```

```
  ResultsP4Q4T[[next_quarter]] <- calculate_performance(Q4_bottom50_smb, top_isins,
  next_quarter)
}
names(ResultsP4Q4T) <- quarters$next_quarter

ResultsP4Q4T <- ResultsP4Q4T[lapply(ResultsP4Q4T, nrow) > 0]

ResultsP4Q4T <- do.call(rbind, lapply(names(ResultsP4Q4T), function(x) {
  data.frame(Quarter = x, ResultsP4Q4T[[x]])
}))

regression_P4Q4 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
                Sum_Return_SMB + Sum_Return_HML +
                Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q4T)
summary(regression_P4Q4)

alfa_P4Q4 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q4T)
summary(alfa_P4Q4)



summary(regression_P1Q1)
summary(regression_P1Q2)
summary(regression_P1Q4)

summary(regression_P2Q1)
summary(regression_P2Q2)
summary(regression_P2Q4)

summary(regression_P4Q1)
summary(regression_P4Q2)
summary(regression_P4Q4)

summary(alfa_P1Q1)
summary(alfa_P1Q2)
summary(alfa_P1Q4)

summary(alfa_P2Q1)
summary(alfa_P2Q2)
summary(alfa_P2Q4)

summary(alfa_P4Q1)
summary(alfa_P4Q2)
summary(alfa_P4Q4)
```

Informationsraten

```
annual_returns_X <- annual_returns %>%
```

```
  mutate(Yearly_Return = Yearly_Return*100)

annual_log_returnsP1Q1 <- annual_log_returnsP1Q1 %>%
  mutate(Annual_Log_Return = Annual_Log_Return * 100)

combined_IR_P1Q1 <- merge(annual_returns_X, annual_log_returnsP1Q1, by = "Year")

# Calculate excess returns
combined_IR_P1Q1$Excess_Returns <- combined_IR_P1Q1$Annual_Log_Return -
  combined_IR_P1Q1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P1Q1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q1 <- mean(combined_IR_P1Q1$Excess_Returns) / std_excess

write.xlsx(combined_IR_P1Q1, file = "Documents/høst 2023/combined_IR_P1Q1.xlsx", rowNames =
  FALSE)


combined_IR_P1Q2 <- merge(annual_returns_X, annual_log_returnsP1Q2, by = "Year")

# Calculate excess returns
combined_IR_P1Q2$Excess_Returns <- combined_IR_P1Q2$Annual_Log_Return -
  combined_IR_P1Q2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess_P1Q2 <- sd(combined_IR_P1Q2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q2 <- mean(combined_IR_P1Q2$Excess_Returns) / std_excess_P1Q2


combined_IR_P1Q4 <- merge(annual_returns_X, annual_log_returnsP1Q4, by = "Year")

# Calculate excess returns
combined_IR_P1Q4$Excess_Returns <- combined_IR_P1Q4$Annual_Log_Return -
  combined_IR_P1Q4$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess_P1Q4 <- sd(combined_IR_P1Q4$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q4 <- mean(combined_IR_P1Q4$Excess_Returns) / std_excess_P1Q4


combined_IR_P2Q1 <- merge(annual_returns_X, annual_log_returnsP2Q1, by = "Year")

# Calculate excess returns
combined_IR_P2Q1$Excess_Returns <- combined_IR_P2Q1$Annual_Log_Return -
  combined_IR_P2Q1$Yearly_Return
```

```r
# Calculate standard deviation of the excess returns
std_excess_P2Q1 <- sd(combined_IR_P2Q1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q1 <- mean(combined_IR_P2Q1$Excess_Returns) / std_excess_P2Q1


combined_IR_P2Q2 <- merge(annual_returns_X, annual_log_returnsP2Q2, by = "Year")

# Calculate excess returns
combined_IR_P2Q2$Excess_Returns <- combined_IR_P2Q2$Annual_Log_Return -
 combined_IR_P2Q2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess_P2Q2 <- sd(combined_IR_P2Q2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q2 <- mean(combined_IR_P2Q2$Excess_Returns) / std_excess_P2Q2


combined_IR_P2Q4 <- merge(annual_returns_X, annual_log_returnsP2Q4, by = "Year")

# Calculate excess returns
combined_IR_P2Q4$Excess_Returns <- combined_IR_P2Q4$Annual_Log_Return -
 combined_IR_P2Q4$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess_P2Q4 <- sd(combined_IR_P2Q4$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q4 <- mean(combined_IR_P2Q4$Excess_Returns) / std_excess_P2Q4


combined_IR_P4Q1 <- merge(annual_returns_X, annual_log_returnsP4Q1, by = "Year")

# Calculate excess returns
combined_IR_P4Q1$Excess_Returns <- combined_IR_P4Q1$Annual_Log_Return -
 combined_IR_P4Q1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess_P4Q1 <- sd(combined_IR_P4Q1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q1 <- mean(combined_IR_P4Q1$Excess_Returns) / std_excess_P4Q1


combined_IR_P4Q2 <- merge(annual_returns_X, annual_log_returnsP4Q2, by = "Year")

# Calculate excess returns
combined_IR_P4Q2$Excess_Returns <- combined_IR_P4Q2$Annual_Log_Return -
 combined_IR_P4Q2$Yearly_Return
```

```r
# Calculate standard deviation of the excess returns
std_excess_P4Q2 <- sd(combined_IR_P4Q2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q2 <- mean(combined_IR_P4Q2$Excess_Returns) / std_excess_P4Q2


combined_IR_P4Q4 <- merge(annual_returns_X, annual_log_returnsP4Q4, by = "Year")

# Calculate excess returns
combined_IR_P4Q4$Excess_Returns <- combined_IR_P4Q4$Annual_Log_Return -
 combined_IR_P4Q4$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess_P4Q4 <- sd(combined_IR_P4Q4$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q4 <- mean(combined_IR_P4Q4$Excess_Returns) / std_excess_P4Q4


# Print the Information Ratio
print(information_ratio_P1Q1)
print(information_ratio_P1Q2)
print(information_ratio_P1Q4)

print(information_ratio_P2Q1)
print(information_ratio_P2Q2)
print(information_ratio_P2Q4)

print(information_ratio_P4Q1)
print(information_ratio_P4Q2)
print(information_ratio_P4Q4)


#########---------alfa


benchmark_quarterly <- benchmark_data %>%
 group_by(Quarter) %>%
 summarize(Benchmark_Return = mean(OBX_pure, na.rm = TRUE))

# Merge the benchmark returns with the results data

Alfa_regression_P1Q1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q1)
Alfa_regression_P1Q2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q2)
Alfa_regression_P1Q4 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q4)

Alfa_regression_P2Q1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q1)
Alfa_regression_P2Q2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q2)
Alfa_regression_P2Q4 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q4)
```

```
Alfa_regression_P4Q1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q1)
Alfa_regression_P4Q2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q2)
Alfa_regression_P4Q4 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q4)

summary(Alfa_regression_P1Q1)
summary(Alfa_regression_P1Q2)
summary(Alfa_regression_P1Q4)

summary(Alfa_regression_P2Q1)
summary(Alfa_regression_P2Q2)
summary(Alfa_regression_P2Q4)

summary(Alfa_regression_P4Q1)
summary(Alfa_regression_P4Q2)
summary(Alfa_regression_P4Q4)
```

Robusthetstest

```
# Filter the dataset from 2009 to 2015
ResultsP1Q1U_periode1 <- ResultsP1Q1U %>%
  filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP1Q1U_periode2 <- ResultsP1Q1U %>%
  filter(Year >= 2016 & Year <= 2022)

# Filter the dataset from 2009 to 2015
ResultsP1Q2U_periode1 <- ResultsP1Q2U %>%
  filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP1Q2U_periode2 <- ResultsP1Q2U %>%
  filter(Year >= 2016 & Year <= 2022)

# Filter the dataset from 2009 to 2015
ResultsP1Q4U_periode1 <- ResultsP1Q4U %>%
  filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP1Q4U_periode2 <- ResultsP1Q4U %>%
  filter(Year >= 2016 & Year <= 2022)

# OBX Benchmark
benchmark_periode1 <- benchmark_data %>%
  filter(Year >= 2009 & Year <= 2015)

benchmark_periode2 <- benchmark_data %>%
  filter(Year >= 2016 & Year <= 2022)
```

```r
benchmark_returns1 <- benchmark_periode1 %>%
  group_by(Year) %>%
  summarize(Yearly_Return = compounded_return(OBX_pure), .groups = 'drop')

benchmark_returns2 <- benchmark_periode2 %>%
  group_by(Year) %>%
  summarize(Yearly_Return = compounded_return(OBX_pure), .groups = 'drop')


# Extract year from Quarter
ResultsP1Q1U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q1U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP1Q1U_periode1 <- ResultsP1Q1U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP1Q1U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q1U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP1Q1U_periode2 <- ResultsP1Q1U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

##### P2
ResultsP1Q2U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q2U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP1Q2U_periode1 <- ResultsP1Q2U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP1Q2U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q2U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP1Q2U_periode2 <- ResultsP1Q2U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

##### Q4

ResultsP1Q4U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q4U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP1Q4U_periode1 <- ResultsP1Q4U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
```

```
ResultsP1Q4U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP1Q4U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP1Q4U_periode2 <- ResultsP1Q4U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')


# 2009 - 20015
# View the summed annual log returns
print(annual_log_ResultsP1Q1U_periode1)
print(annual_log_ResultsP1Q2U_periode1)
print(annual_log_ResultsP1Q4U_periode1)
print(benchmark_returns1)

sd_value_P1Q1_P1 <- sd(annual_log_ResultsP1Q1U_periode1$Annual_Log_Return)
sd_value_P1Q2_P1 <- sd(annual_log_ResultsP1Q2U_periode1$Annual_Log_Return)
sd_value_P1Q4_P1 <- sd(annual_log_ResultsP1Q4U_periode1$Annual_Log_Return)
sd_value_Bench_P1 <- sd(benchmark_returns1$Yearly_Return)

print(sd_value_P1Q1_P1)
print(sd_value_P1Q2_P1)
print(sd_value_P1Q4_P1)
print(sd_value_Bench_P1)


# average return

average_return_P1Q1_P1 <- mean(annual_log_ResultsP1Q1U_periode1$Annual_Log_Return)
average_return_P1Q2_P1 <- mean(annual_log_ResultsP1Q2U_periode1$Annual_Log_Return)
average_return_P1Q4_P1 <- mean(annual_log_ResultsP1Q4U_periode1$Annual_Log_Return)
average_return_Bench_P1 <- mean(benchmark_returns1$Yearly_Return)


print(average_return_P1Q1_P1)

# Calculate Sharpe Ratio
sharpe_ratio_P1Q1_P1 <- (average_return_P1Q1_P1) / sd_value_P1Q1_P1
sharpe_ratio_P1Q2_P1 <- (average_return_P1Q2_P1) / sd_value_P1Q2_P1
sharpe_ratio_P1Q4_P1 <- (average_return_P1Q4_P1) / sd_value_P1Q4_P1
sharpe_ratio_Bench_P1 <- (average_return_Bench_P1) / sd_value_Bench_P1

print(sharpe_ratio_P1Q1_P1)
print(sharpe_ratio_P1Q2_P1)
print(sharpe_ratio_P1Q4_P1)
print(sharpe_ratio_Bench_P1)

# 2016 - 2022
print(annual_log_ResultsP1Q1U_periode2)
print(annual_log_ResultsP1Q2U_periode2)
print(annual_log_ResultsP1Q4U_periode2)
print(benchmark_returns2)
```

```
sd_value_P1Q1_P2 <- sd(annual_log_ResultsP1Q1U_periode2$Annual_Log_Return)
sd_value_P1Q2_P2 <- sd(annual_log_ResultsP1Q2U_periode2$Annual_Log_Return)
sd_value_P1Q4_P2 <- sd(annual_log_ResultsP1Q4U_periode2$Annual_Log_Return)
sd_value_Bench_P2 <- sd(benchmark_returns2$Yearly_Return)

print(sd_value_P1Q1_P2)
print(sd_value_P1Q2_P2)
print(sd_value_P1Q4_P2)
print(sd_value_Bench_P2)

# average return

average_return_P1Q1_P2 <- mean(annual_log_ResultsP1Q1U_periode2$Annual_Log_Return)
average_return_P1Q2_P2 <- mean(annual_log_ResultsP1Q2U_periode2$Annual_Log_Return)
average_return_P1Q4_P2 <- mean(annual_log_ResultsP1Q4U_periode2$Annual_Log_Return)
average_return_Bench_P2 <- mean(benchmark_returns2$Yearly_Return)

print(average_return_P1Q1_P2)
# Calculate Sharpe Ratio
sharpe_ratio_P1Q1_P2 <- (average_return_P1Q1_P2) / sd_value_P1Q1_P2
sharpe_ratio_P1Q2_P2 <- (average_return_P1Q2_P2) / sd_value_P1Q2_P2
sharpe_ratio_P1Q4_P2 <- (average_return_P1Q4_P2) / sd_value_P1Q4_P2
sharpe_ratio_Bench_P2 <- (average_return_Bench_P2) / sd_value_Bench_P2

print(sharpe_ratio_P1Q1_P2)
print(sharpe_ratio_P1Q2_P2)
print(sharpe_ratio_P1Q4_P2)
print(sharpe_ratio_Bench_P2)


# infromasjonsrate
# 2009 - 2015
combined_IR_P1Q1_P1 <- merge(benchmark_returns1, annual_log_ResultsP1Q1U_periode1, by =
  "Year")

# Calculate excess returns
combined_IR_P1Q1_P1$Excess_Returns <- combined_IR_P1Q1_P1$Annual_Log_Return -
  combined_IR_P1Q1_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P1Q1_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q1_P1 <- mean(combined_IR_P1Q1_P1$Excess_Returns) / std_excess

# 2016-2022
combined_IR_P1Q1_P2 <- merge(benchmark_returns2, annual_log_ResultsP1Q1U_periode2, by =
  "Year")

# Calculate excess returns
```

```
combined_IR_P1Q1_P2$Excess_Returns <- combined_IR_P1Q1_P2$Annual_Log_Return -
 combined_IR_P1Q1_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P1Q1_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q1_P2 <- mean(combined_IR_P1Q1_P2$Excess_Returns) / std_excess

#####------Q2-------######

# 2009 - 2015
combined_IR_P1Q2_P1 <- merge(benchmark_returns1, annual_log_ResultsP1Q2U_periode1, by =
 "Year")

# Calculate excess returns
combined_IR_P1Q2_P1$Excess_Returns <- combined_IR_P1Q2_P1$Annual_Log_Return -
 combined_IR_P1Q2_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P1Q2_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q2_P1 <- mean(combined_IR_P1Q2_P1$Excess_Returns) / std_excess

# 2016-2022
combined_IR_P1Q2_P2 <- merge(benchmark_returns2, annual_log_ResultsP1Q2U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P1Q2_P2$Excess_Returns <- combined_IR_P1Q2_P2$Annual_Log_Return -
 combined_IR_P1Q2_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P1Q2_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q2_P2 <- mean(combined_IR_P1Q2_P2$Excess_Returns) / std_excess

#####-----Q4-------#######

# 2009 - 2015
combined_IR_P1Q4_P1 <- merge(benchmark_returns1, annual_log_ResultsP1Q4U_periode1, by =
 "Year")

# Calculate excess returns
combined_IR_P1Q4_P1$Excess_Returns <- combined_IR_P1Q4_P1$Annual_Log_Return -
 combined_IR_P1Q4_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P1Q4_P1$Excess_Returns)
```

```r
# Calculate the Information Ratio
information_ratio_P1Q4_P1 <- mean(combined_IR_P1Q4_P1$Excess_Returns) / std_excess

# 2016-2022
combined_IR_P1Q4_P2 <- merge(benchmark_returns2, annual_log_ResultsP1Q4U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P1Q4_P2$Excess_Returns <- combined_IR_P1Q4_P2$Annual_Log_Return -
 combined_IR_P1Q4_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P1Q4_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P1Q4_P2 <- mean(combined_IR_P1Q4_P2$Excess_Returns) / std_excess

# Print the Information Ratio
print(information_ratio_P1Q1_P1)
print(information_ratio_P1Q2_P1)
print(information_ratio_P1Q4_P1)

# Print the Information Ratio
print(information_ratio_P1Q1_P2)
print(information_ratio_P1Q2_P2)
print(information_ratio_P1Q4_P2)


############----------P2_Porteføljer--------#################

# Filter the dataset from 2009 to 2015
ResultsP2Q1U_periode1 <- ResultsP2Q1U %>%
  filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP2Q1U_periode2 <- ResultsP2Q1U %>%
  filter(Year >= 2016 & Year <= 2022)

# Filter the dataset from 2009 to 2015
ResultsP2Q2U_periode1 <- ResultsP2Q2U %>%
  filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP2Q2U_periode2 <- ResultsP2Q2U %>%
  filter(Year >= 2016 & Year <= 2022)

# Filter the dataset from 2009 to 2015
ResultsP2Q4U_periode1 <- ResultsP2Q4U %>%
  filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP2Q4U_periode2 <- ResultsP2Q4U %>%
```

```
  filter(Year >= 2016 & Year <= 2022)

# OBX Benchmark
benchmark_periode1 <- benchmark_data %>%
  filter(Year >= 2009 & Year <= 2015)

benchmark_periode2 <- benchmark_data %>%
  filter(Year >= 2016 & Year <= 2022)

benchmark_returns1 <- benchmark_periode1 %>%
  group_by(Year) %>%
  summarize(Yearly_Return = compounded_return(OBX_pure), .groups = 'drop')

benchmark_returns2 <- benchmark_periode2 %>%
  group_by(Year) %>%
  summarize(Yearly_Return = compounded_return(OBX_pure), .groups = 'drop')


# Extract year from Quarter
ResultsP2Q1U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q1U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP2Q1U_periode1 <- ResultsP2Q1U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP2Q1U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q1U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP2Q1U_periode2 <- ResultsP2Q1U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

##### P2
ResultsP2Q2U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q2U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP2Q2U_periode1 <- ResultsP2Q2U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP2Q2U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q2U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP2Q2U_periode2 <- ResultsP2Q2U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

##### Q4
```

```r
ResultsP2Q4U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q4U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP2Q4U_periode1 <- ResultsP2Q4U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP2Q4U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP2Q4U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP2Q4U_periode2 <- ResultsP2Q4U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')


# 2009 - 20015
# View the summed annual log returns
print(annual_log_ResultsP2Q1U_periode1)
print(annual_log_ResultsP2Q2U_periode1)
print(annual_log_ResultsP2Q4U_periode1)
print(benchmark_returns1)

sd_value_P2Q1_P1 <- sd(annual_log_ResultsP2Q1U_periode1$Annual_Log_Return)
sd_value_P2Q2_P1 <- sd(annual_log_ResultsP2Q2U_periode1$Annual_Log_Return)
sd_value_P2Q4_P1 <- sd(annual_log_ResultsP2Q4U_periode1$Annual_Log_Return)
sd_value_Bench_P1 <- sd(benchmark_returns1$Yearly_Return)

print(sd_value_P2Q1_P1)
print(sd_value_P2Q2_P1)
print(sd_value_P2Q4_P1)
print(sd_value_Bench_P1)


# average return

average_return_P2Q1_P1 <- mean(annual_log_ResultsP2Q1U_periode1$Annual_Log_Return)
average_return_P2Q2_P1 <- mean(annual_log_ResultsP2Q2U_periode1$Annual_Log_Return)
average_return_P2Q4_P1 <- mean(annual_log_ResultsP2Q4U_periode1$Annual_Log_Return)
average_return_Bench_P1 <- mean(benchmark_returns1$Yearly_Return)

# Calculate Sharpe Ratio
sharpe_ratio_P2Q1_P1 <- (average_return_P2Q1_P1) / sd_value_P2Q1_P1
sharpe_ratio_P2Q2_P1 <- (average_return_P2Q2_P1) / sd_value_P2Q2_P1
sharpe_ratio_P2Q4_P1 <- (average_return_P2Q4_P1) / sd_value_P2Q4_P1
sharpe_ratio_Bench_P1 <- (average_return_Bench_P1) / sd_value_Bench_P1

print(sharpe_ratio_P2Q1_P1)
print(sharpe_ratio_P2Q2_P1)
print(sharpe_ratio_P2Q4_P1)
print(sharpe_ratio_Bench_P1)
```

```
# 2016 - 2022
print(annual_log_ResultsP2Q1U_periode2)
print(annual_log_ResultsP2Q2U_periode2)
print(annual_log_ResultsP2Q4U_periode2)
print(benchmark_returns2)

sd_value_P2Q1_P2 <- sd(annual_log_ResultsP2Q1U_periode2$Annual_Log_Return)
sd_value_P2Q2_P2 <- sd(annual_log_ResultsP2Q2U_periode2$Annual_Log_Return)
sd_value_P2Q4_P2 <- sd(annual_log_ResultsP2Q4U_periode2$Annual_Log_Return)
sd_value_Bench_P2 <- sd(benchmark_returns2$Yearly_Return)

print(sd_value_P2Q1_P2)
print(sd_value_P2Q2_P2)
print(sd_value_P2Q4_P2)
print(sd_value_Bench_P2)

# average return

average_return_P2Q1_P2 <- mean(annual_log_ResultsP2Q1U_periode2$Annual_Log_Return)
average_return_P2Q2_P2 <- mean(annual_log_ResultsP2Q2U_periode2$Annual_Log_Return)
average_return_P2Q4_P2 <- mean(annual_log_ResultsP2Q4U_periode2$Annual_Log_Return)
average_return_Bench_P2 <- mean(benchmark_returns2$Yearly_Return)

# Calculate Sharpe Ratio
sharpe_ratio_P2Q1_P2 <- (average_return_P2Q1_P2) / sd_value_P2Q1_P2
sharpe_ratio_P2Q2_P2 <- (average_return_P2Q2_P2) / sd_value_P2Q2_P2
sharpe_ratio_P2Q4_P2 <- (average_return_P2Q4_P2) / sd_value_P2Q4_P2
sharpe_ratio_Bench_P2 <- (average_return_Bench_P2) / sd_value_Bench_P2

print(sharpe_ratio_P2Q1_P2)
print(sharpe_ratio_P2Q2_P2)
print(sharpe_ratio_P2Q4_P2)
print(sharpe_ratio_Bench_P2)


# infromasjonsrate
# 2009 - 2015
combined_IR_P2Q1_P1 <- merge(benchmark_returns1, annual_log_ResultsP2Q1U_periode1, by =
 "Year")

# Calculate excess returns
combined_IR_P2Q1_P1$Excess_Returns <- combined_IR_P2Q1_P1$Annual_Log_Return -
 combined_IR_P2Q1_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P2Q1_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q1_P1 <- mean(combined_IR_P2Q1_P1$Excess_Returns) / std_excess

# 2016-2022
```

```
combined_IR_P2Q1_P2 <- merge(benchmark_returns2, annual_log_ResultsP2Q1U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P2Q1_P2$Excess_Returns <- combined_IR_P2Q1_P2$Annual_Log_Return -
 combined_IR_P2Q1_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P2Q1_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q1_P2 <- mean(combined_IR_P2Q1_P2$Excess_Returns) / std_excess

#####------Q2-------######

# 2009 - 2015
combined_IR_P2Q2_P1 <- merge(benchmark_returns1, annual_log_ResultsP2Q2U_periode1, by =
 "Year")

# Calculate excess returns
combined_IR_P2Q2_P1$Excess_Returns <- combined_IR_P2Q2_P1$Annual_Log_Return -
 combined_IR_P2Q2_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P2Q2_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q2_P1 <- mean(combined_IR_P2Q2_P1$Excess_Returns) / std_excess

# 2016-2022
combined_IR_P2Q2_P2 <- merge(benchmark_returns2, annual_log_ResultsP2Q2U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P2Q2_P2$Excess_Returns <- combined_IR_P2Q2_P2$Annual_Log_Return -
 combined_IR_P2Q2_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P2Q2_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q2_P2 <- mean(combined_IR_P2Q2_P2$Excess_Returns) / std_excess

#####-----Q4-------#######

# 2009 - 2015
combined_IR_P2Q4_P1 <- merge(benchmark_returns1, annual_log_ResultsP2Q4U_periode1, by =
 "Year")

# Calculate excess returns
combined_IR_P2Q4_P1$Excess_Returns <- combined_IR_P2Q4_P1$Annual_Log_Return -
 combined_IR_P2Q4_P1$Yearly_Return
```

```
# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P2Q4_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q4_P1 <- mean(combined_IR_P2Q4_P1$Excess_Returns) / std_excess

# 2016-2022
combined_IR_P2Q4_P2 <- merge(benchmark_returns2, annual_log_ResultsP2Q4U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P2Q4_P2$Excess_Returns <- combined_IR_P2Q4_P2$Annual_Log_Return -
 combined_IR_P2Q4_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P2Q4_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P2Q4_P2 <- mean(combined_IR_P2Q4_P2$Excess_Returns) / std_excess

# Print the Information Ratio
print(information_ratio_P2Q1_P1)
print(information_ratio_P2Q2_P1)
print(information_ratio_P2Q4_P1)

# Print the Information Ratio
print(information_ratio_P2Q1_P2)
print(information_ratio_P2Q2_P2)
print(information_ratio_P2Q4_P2)

####################----------P4_porteføljer----------###############

# Filter the dataset from 2009 to 2015
ResultsP4Q1U_periode1 <- ResultsP4Q1U %>%
 filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP4Q1U_periode2 <- ResultsP4Q1U %>%
 filter(Year >= 2016 & Year <= 2022)

# Filter the dataset from 2009 to 2015
ResultsP4Q2U_periode1 <- ResultsP4Q2U %>%
 filter(Year >= 2009 & Year <= 2015)

# Filter the dataset from 2016 to 2022
ResultsP4Q2U_periode2 <- ResultsP4Q2U %>%
 filter(Year >= 2016 & Year <= 2022)

# Filter the dataset from 2009 to 2015
ResultsP4Q4U_periode1 <- ResultsP4Q4U %>%
 filter(Year >= 2009 & Year <= 2015)
```

```r
# Filter the dataset from 2016 to 2022
ResultsP4Q4U_periode2 <- ResultsP4Q4U %>%
  filter(Year >= 2016 & Year <= 2022)

# OBX Benchmark
benchmark_periode1 <- benchmark_data %>%
  filter(Year >= 2009 & Year <= 2015)

benchmark_periode2 <- benchmark_data %>%
  filter(Year >= 2016 & Year <= 2022)

benchmark_returns1 <- benchmark_periode1 %>%
  group_by(Year) %>%
  summarize(Yearly_Return = compounded_return(OBX_pure), .groups = 'drop')

benchmark_returns2 <- benchmark_periode2 %>%
  group_by(Year) %>%
  summarize(Yearly_Return = compounded_return(OBX_pure), .groups = 'drop')


# Extract year from Quarter
ResultsP4Q1U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q1U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP4Q1U_periode1 <- ResultsP4Q1U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP4Q1U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q1U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP4Q1U_periode2 <- ResultsP4Q1U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

##### P2
ResultsP4Q2U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q2U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP4Q2U_periode1 <- ResultsP4Q2U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP4Q2U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q2U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP4Q2U_periode2 <- ResultsP4Q2U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')
```

```
##### Q4

ResultsP4Q4U_periode1$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q4U_periode1$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP4Q4U_periode1 <- ResultsP4Q4U_periode1 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')

# Extract year from Quarter
ResultsP4Q4U_periode2$Year <- as.numeric(sub("-Q.*", "", ResultsP4Q4U_periode2$Quarter))

# Group by Year and sum the log returns
annual_log_ResultsP4Q4U_periode2 <- ResultsP4Q4U_periode2 %>%
  group_by(Year) %>%
  summarize(Annual_Log_Return = sum(Total_lnDeltaP), .groups = 'drop')


# 2009 - 20015
# View the summed annual log returns
print(annual_log_ResultsP4Q1U_periode1)
print(annual_log_ResultsP4Q2U_periode1)
print(annual_log_ResultsP4Q4U_periode1)
print(benchmark_returns1)

sd_value_P4Q1_P1 <- sd(annual_log_ResultsP4Q1U_periode1$Annual_Log_Return)
sd_value_P4Q2_P1 <- sd(annual_log_ResultsP4Q2U_periode1$Annual_Log_Return)
sd_value_P4Q4_P1 <- sd(annual_log_ResultsP4Q4U_periode1$Annual_Log_Return)
sd_value_Bench_P1 <- sd(benchmark_returns1$Yearly_Return)

print(sd_value_P4Q1_P1)
print(sd_value_P4Q2_P1)
print(sd_value_P4Q4_P1)
print(sd_value_Bench_P1)


# average return

average_return_P4Q1_P1 <- mean(annual_log_ResultsP4Q1U_periode1$Annual_Log_Return)
average_return_P4Q2_P1 <- mean(annual_log_ResultsP4Q2U_periode1$Annual_Log_Return)
average_return_P4Q4_P1 <- mean(annual_log_ResultsP4Q4U_periode1$Annual_Log_Return)
average_return_Bench_P1 <- mean(benchmark_returns1$Yearly_Return)

# Calculate Sharpe Ratio
sharpe_ratio_P4Q1_P1 <- (average_return_P4Q1_P1) / sd_value_P4Q1_P1
sharpe_ratio_P4Q2_P1 <- (average_return_P4Q2_P1) / sd_value_P4Q2_P1
sharpe_ratio_P4Q4_P1 <- (average_return_P4Q4_P1) / sd_value_P4Q4_P1
sharpe_ratio_Bench_P1 <- (average_return_Bench_P1) / sd_value_Bench_P1

print(sharpe_ratio_P4Q1_P1)
print(sharpe_ratio_P4Q2_P1)
```

```
print(sharpe_ratio_P4Q4_P1)
print(sharpe_ratio_Bench_P1)

# 2016 - 2022
print(annual_log_ResultsP4Q1U_periode2)
print(annual_log_ResultsP4Q2U_periode2)
print(annual_log_ResultsP4Q4U_periode2)
print(benchmark_returns2)

sd_value_P4Q1_P2 <- sd(annual_log_ResultsP4Q1U_periode2$Annual_Log_Return)
sd_value_P4Q2_P2 <- sd(annual_log_ResultsP4Q2U_periode2$Annual_Log_Return)
sd_value_P4Q4_P2 <- sd(annual_log_ResultsP4Q4U_periode2$Annual_Log_Return)
sd_value_Bench_P2 <- sd(benchmark_returns2$Yearly_Return)

print(sd_value_P4Q1_P2)
print(sd_value_P4Q2_P2)
print(sd_value_P4Q4_P2)
print(sd_value_Bench_P2)

# average return

average_return_P4Q1_P2 <- mean(annual_log_ResultsP4Q1U_periode2$Annual_Log_Return)
average_return_P4Q2_P2 <- mean(annual_log_ResultsP4Q2U_periode2$Annual_Log_Return)
average_return_P4Q4_P2 <- mean(annual_log_ResultsP4Q4U_periode2$Annual_Log_Return)
average_return_Bench_P2 <- mean(benchmark_returns2$Yearly_Return)

# Calculate Sharpe Ratio
sharpe_ratio_P4Q1_P2 <- (average_return_P4Q1_P2) / sd_value_P4Q1_P2
sharpe_ratio_P4Q2_P2 <- (average_return_P4Q2_P2) / sd_value_P4Q2_P2
sharpe_ratio_P4Q4_P2 <- (average_return_P4Q4_P2) / sd_value_P4Q4_P2
sharpe_ratio_Bench_P2 <- (average_return_Bench_P2) / sd_value_Bench_P2

print(sharpe_ratio_P4Q1_P2)
print(sharpe_ratio_P4Q2_P2)
print(sharpe_ratio_P4Q4_P2)
print(sharpe_ratio_Bench_P2)


# infromasjonsrate
# 2009 - 2015
combined_IR_P4Q1_P1 <- merge(benchmark_returns1, annual_log_ResultsP4Q1U_periode1, by =
  "Year")

# Calculate excess returns
combined_IR_P4Q1_P1$Excess_Returns <- combined_IR_P4Q1_P1$Annual_Log_Return -
  combined_IR_P4Q1_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P4Q1_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q1_P1 <- mean(combined_IR_P4Q1_P1$Excess_Returns) / std_excess
```

```
# 2016-2022
combined_IR_P4Q1_P2 <- merge(benchmark_returns2, annual_log_ResultsP4Q1U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P4Q1_P2$Excess_Returns <- combined_IR_P4Q1_P2$Annual_Log_Return -
 combined_IR_P4Q1_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P4Q1_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q1_P2 <- mean(combined_IR_P4Q1_P2$Excess_Returns) / std_excess

#####------Q2-------######

# 2009 - 2015
combined_IR_P4Q2_P1 <- merge(benchmark_returns1, annual_log_ResultsP4Q2U_periode1, by =
 "Year")

# Calculate excess returns
combined_IR_P4Q2_P1$Excess_Returns <- combined_IR_P4Q2_P1$Annual_Log_Return -
 combined_IR_P4Q2_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P4Q2_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q2_P1 <- mean(combined_IR_P4Q2_P1$Excess_Returns) / std_excess

# 2016-2022
combined_IR_P4Q2_P2 <- merge(benchmark_returns2, annual_log_ResultsP4Q2U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P4Q2_P2$Excess_Returns <- combined_IR_P4Q2_P2$Annual_Log_Return -
 combined_IR_P4Q2_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P4Q2_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q2_P2 <- mean(combined_IR_P4Q2_P2$Excess_Returns) / std_excess

#####-----Q4-------#######

# 2009 - 2015
combined_IR_P4Q4_P1 <- merge(benchmark_returns1, annual_log_ResultsP4Q4U_periode1, by =
 "Year")

# Calculate excess returns
```

```
combined_IR_P4Q4_P1$Excess_Returns <- combined_IR_P4Q4_P1$Annual_Log_Return -
 combined_IR_P4Q4_P1$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P4Q4_P1$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q4_P1 <- mean(combined_IR_P4Q4_P1$Excess_Returns) / std_excess

# 2016-2022
combined_IR_P4Q4_P2 <- merge(benchmark_returns2, annual_log_ResultsP4Q4U_periode2, by =
 "Year")

# Calculate excess returns
combined_IR_P4Q4_P2$Excess_Returns <- combined_IR_P4Q4_P2$Annual_Log_Return -
 combined_IR_P4Q4_P2$Yearly_Return

# Calculate standard deviation of the excess returns
std_excess <- sd(combined_IR_P4Q4_P2$Excess_Returns)

# Calculate the Information Ratio
information_ratio_P4Q4_P2 <- mean(combined_IR_P4Q4_P2$Excess_Returns) / std_excess

## start
# Print the Information Ratio
print(information_ratio_P4Q1_P1)
print(information_ratio_P4Q2_P1)
print(information_ratio_P4Q4_P1)

# Print the Information Ratio
print(information_ratio_P4Q1_P2)
print(information_ratio_P4Q2_P2)
print(information_ratio_P4Q4_P2)

# P2 periode 1
print(information_ratio_P2Q1_P1)
print(information_ratio_P2Q2_P1)
print(information_ratio_P2Q4_P1)

# Periode 2
print(information_ratio_P2Q1_P2)
print(information_ratio_P2Q2_P2)
print(information_ratio_P2Q4_P2)

# P4 periode 1
print(information_ratio_P1Q1_P1)
print(information_ratio_P1Q2_P1)
print(information_ratio_P1Q4_P1)

# Periode 2
print(information_ratio_P1Q1_P2)
print(information_ratio_P1Q2_P2)
```

```
print(information_ratio_P1Q4_P2)

## start
#### sharpe ---#####
print(sharpe_ratio_P4Q1_P1)
print(sharpe_ratio_P4Q2_P1)
print(sharpe_ratio_P4Q4_P1)
print(sharpe_ratio_P4Q1_P2)
print(sharpe_ratio_P4Q2_P2)
print(sharpe_ratio_P4Q4_P2)

print(sharpe_ratio_Bench_P1)
print(sharpe_ratio_Bench_P2)
#### P2
print(sharpe_ratio_P2Q1_P1)
print(sharpe_ratio_P2Q2_P1)
print(sharpe_ratio_P2Q4_P1)
print(sharpe_ratio_P2Q1_P2)
print(sharpe_ratio_P2Q2_P2)
print(sharpe_ratio_P2Q4_P2)

#### P1
print(sharpe_ratio_P1Q1_P1)
print(sharpe_ratio_P1Q2_P1)
print(sharpe_ratio_P1Q4_P1)
print(sharpe_ratio_P1Q1_P2)
print(sharpe_ratio_P1Q2_P2)
print(sharpe_ratio_P1Q4_P2)

########----------Results-and-SD------#############


print(sd_value_P1Q1_P1)
print(sd_value_P1Q2_P1)
print(sd_value_P1Q4_P1)
print(sd_value_Bench_P1)

print(sd_value_P1Q1_P2)
print(sd_value_P1Q2_P2)
print(sd_value_P1Q4_P2)
print(sd_value_Bench_P2)

average_return_P1Q4_P1 <- mean(annual_log_ResultsP1Q4U_periode1$Annual_Log_Return)
average_return_Bench_P1 <- mean(benchmark_returns1$Yearly_Return)


print(average_return_P1Q1_P1)
print(sd_value_P1Q1_P1)

print(average_return_P1Q1_P2)
print(sd_value_P1Q1_P2)
```

```
print(average_return_P1Q2_P1)
print(sd_value_P1Q2_P1)

print(average_return_P1Q2_P2)
print(sd_value_P1Q2_P2)

print(average_return_P1Q4_P1)
print(sd_value_P1Q4_P1)

print(average_return_P1Q4_P2)
print(sd_value_P1Q4_P2)


print(average_return_Bench_P1)
print(average_return_Bench_P2)

#ny

print(average_return_P2Q1_P1)
print(sd_value_P2Q1_P1)

print(average_return_P2Q1_P2)
print(sd_value_P2Q1_P2)

print(average_return_P2Q2_P1)
print(sd_value_P2Q2_P1)

print(average_return_P2Q2_P2)
print(sd_value_P2Q2_P2)

print(average_return_P2Q4_P1)
print(sd_value_P2Q4_P1)

print(average_return_P2Q4_P2)
print(sd_value_P2Q4_P2)

## NYYY

print(average_return_P4Q1_P1)
print(sd_value_P4Q1_P1)

print(average_return_P4Q1_P2)
print(sd_value_P4Q1_P2)

print(average_return_P4Q2_P1)
print(sd_value_P4Q2_P1)

print(average_return_P4Q2_P2)
print(sd_value_P4Q2_P2)

print(average_return_P4Q4_P1)
print(sd_value_P4Q4_P1)
```

```
print(average_return_P4Q4_P2)
print(sd_value_P4Q4_P2)


############------Alfa------###############

Alfa_regression_P1Q1_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q1_P1)
Alfa_regression_P1Q2_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q2_P1)
Alfa_regression_P1Q4_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q4_P1)
Alfa_regression_P1Q1_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q1_P2)
Alfa_regression_P1Q2_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q2_P2)
Alfa_regression_P1Q4_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P1Q4_P2)

summary(Alfa_regression_P1Q1_P1)
summary(Alfa_regression_P1Q2_P1)
summary(Alfa_regression_P1Q4_P1)
summary(Alfa_regression_P1Q1_P2)
summary(Alfa_regression_P1Q2_P2)
summary(Alfa_regression_P1Q4_P2)

Alfa_regression_P2Q1_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q1_P1)
Alfa_regression_P2Q2_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q2_P1)
Alfa_regression_P2Q4_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q4_P1)
Alfa_regression_P2Q1_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q1_P2)
Alfa_regression_P2Q2_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q2_P2)
Alfa_regression_P2Q4_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P2Q4_P2)

summary(Alfa_regression_P2Q1_P1)
summary(Alfa_regression_P2Q2_P1)
summary(Alfa_regression_P2Q4_P1)
summary(Alfa_regression_P2Q1_P2)
summary(Alfa_regression_P2Q2_P2)
summary(Alfa_regression_P2Q4_P2)

Alfa_regression_P4Q1_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q1_P1)
Alfa_regression_P4Q2_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q2_P1)
Alfa_regression_P4Q4_P1 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q4_P1)
Alfa_regression_P4Q1_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q1_P2)
Alfa_regression_P4Q2_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q2_P2)
Alfa_regression_P4Q4_P2 <- lm(Annual_Log_Return ~ Yearly_Return, data = combined_IR_P4Q4_P2)

summary(Alfa_regression_P4Q1_P1)
summary(Alfa_regression_P4Q2_P1)
summary(Alfa_regression_P4Q4_P1)
summary(Alfa_regression_P4Q1_P2)
summary(Alfa_regression_P4Q2_P2)
summary(Alfa_regression_P4Q4_P2)

ResultsP4Q4T <- ResultsP4Q4T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))
```

```
ResultsP4Q2T <- ResultsP4Q2T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

ResultsP4Q1T <- ResultsP4Q1T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

ResultsP2Q1T <- ResultsP2Q1T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

ResultsP2Q2T <- ResultsP2Q2T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

ResultsP2Q4T <- ResultsP2Q4T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

ResultsP1Q1T <- ResultsP1Q1T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

ResultsP1Q2T <- ResultsP1Q2T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

ResultsP1Q4T <- ResultsP1Q4T %>%
  mutate(Year = as.numeric(substr(Quarter, 1, 4)))

# Split the data into two periods
ResultsP4Q4T_Periode1 <- ResultsP4Q4T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP4Q4T_Periode2 <- ResultsP4Q4T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP4Q2T_Periode1 <- ResultsP4Q2T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP4Q2T_Periode2 <- ResultsP4Q2T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP4Q1T_Periode1 <- ResultsP4Q1T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP4Q1T_Periode2 <- ResultsP4Q1T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP2Q4T_Periode1 <- ResultsP2Q4T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP2Q4T_Periode2 <- ResultsP2Q4T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP2Q2T_Periode1 <- ResultsP2Q2T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP2Q2T_Periode2 <- ResultsP2Q2T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP2Q1T_Periode1 <- ResultsP2Q1T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP2Q1T_Periode2 <- ResultsP2Q1T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP1Q4T_Periode1 <- ResultsP1Q4T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP1Q4T_Periode2 <- ResultsP1Q4T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP1Q2T_Periode1 <- ResultsP1Q2T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP1Q2T_Periode2 <- ResultsP1Q2T %>% filter(Year >= 2016 & Year <= 2022)

ResultsP1Q1T_Periode1 <- ResultsP1Q1T %>% filter(Year >= 2009 & Year <= 2015)
ResultsP1Q1T_Periode2 <- ResultsP1Q1T %>% filter(Year >= 2016 & Year <= 2022)
```

```
regression_P4Q4_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q4T_Periode1)
summary(regression_P4Q4_Periode1)

alfa_P4Q4_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q4T_Periode1)
summary(alfa_P4Q4_Periode1)

regression_P4Q4_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q4T_Periode2)
summary(regression_P4Q4_Periode2)

alfa_P4Q4_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q4T_Periode2)
summary(alfa_P4Q4_Periode2)

regression_P4Q2_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q2T_Periode1)
summary(regression_P4Q2_Periode1)

alfa_P4Q2_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q2T_Periode1)
summary(alfa_P4Q2_Periode1)

regression_P4Q2_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q2T_Periode2)
summary(regression_P4Q2_Periode2)

alfa_P4Q2_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q2T_Periode2)
summary(alfa_P4Q2_Periode2)

regression_P4Q1_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q1T_Periode1)
summary(regression_P4Q1_Periode1)

alfa_P4Q1_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q1T_Periode1)
summary(alfa_P4Q1_Periode1)

regression_P4Q1_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP4Q1T_Periode2)
summary(regression_P4Q1_Periode2)

alfa_P4Q1_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP4Q1T_Periode2)
summary(alfa_P4Q1_Periode2)

regression_P2Q4_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q4T_Periode1)
```

```
summary(regression_P2Q4_Periode1)

alfa_P2Q4_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q4T_Periode1)
summary(alfa_P2Q4_Periode1)

regression_P2Q4_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q4T_Periode2)
summary(regression_P2Q4_Periode2)

alfa_P2Q4_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q4T_Periode2)
summary(alfa_P2Q4_Periode2)

regression_P2Q2_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q2T_Periode1)
summary(regression_P2Q2_Periode1)

alfa_P2Q2_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q2T_Periode1)
summary(alfa_P2Q2_Periode1)

regression_P2Q2_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q2T_Periode2)
summary(regression_P2Q2_Periode2)

alfa_P2Q2_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q2T_Periode2)
summary(alfa_P2Q2_Periode2)

regression_P2Q1_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q1T_Periode1)
summary(regression_P2Q1_Periode1)

alfa_P2Q1_Periode1 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q1T_Periode1)
summary(alfa_P2Q1_Periode1)

regression_P2Q1_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX +
            Sum_Return_SMB + Sum_Return_HML +
            Sum_Return_MOM + Sum_Return_LIQ, data = ResultsP2Q1T_Periode2)
summary(regression_P2Q1_Periode2)

alfa_P2Q1_Periode2 <- lm(Total_lnDeltaP ~ Sum_Return_OBX, data = ResultsP2Q1T_Periode2)
summary(alfa_P2Q1_Periode2)



names(ResultsP1Q4T_Periode1)

regression_P1Q4_Periode1 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
            Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
```

```
            Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data =
 ResultsP1Q4T_Periode1)
summary(regression_P1Q4_Periode1)

alfa_P1Q4_Periode1 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data =
 ResultsP1Q4T_Periode1)
summary(alfa_P1Q4_Periode1)

regression_P1Q4_Periode2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
            Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
            Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data =
 ResultsP1Q4T_Periode2)
summary(regression_P1Q4_Periode2)

alfa_P1Q4_Periode2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data =
 ResultsP1Q4T_Periode2)
summary(alfa_P1Q4_Periode2)

regression_P1Q2_Periode1 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
            Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
            Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data =
 ResultsP1Q2T_Periode1)
summary(regression_P1Q2_Periode1)

alfa_P1Q2_Periode1 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data =
 ResultsP1Q2T_Periode1)
summary(alfa_P1Q2_Periode1)

regression_P1Q2_Periode2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
            Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
            Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data =
 ResultsP1Q2T_Periode2)
summary(regression_P1Q2_Periode2)

alfa_P1Q2_Periode2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data =
 ResultsP1Q2T_Periode2)
summary(alfa_P1Q2_Periode2)

regression_P1Q1_Periode1 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
           Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
            Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data =
 ResultsP1Q1T_Periode1)
summary(regression_P1Q1_Periode1)

alfa_P1Q1_Periode1 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data =
 ResultsP1Q1T_Periode1)
summary(alfa_P1Q1_Periode1)

regression_P1Q1_Periode2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX +
           Performance.Sum_Return_SMB + Performance.Sum_Return_HML +
            Performance.Sum_Return_MOM + Performance.Sum_Return_LIQ, data =
 ResultsP1Q1T_Periode2)
```

```
summary(regression_P1Q1_Periode2)

alfa_P1Q1_Periode2 <- lm(Performance.Total_lnDeltaP ~ Performance.Sum_Return_OBX, data =
 ResultsP1Q1T_Periode2)
summary(alfa_P1Q1_Periode2)


summary(regression_P1Q1_Periode1)
summary(regression_P1Q2_Periode1)
summary(regression_P1Q4_Periode1)

summary(regression_P2Q1_Periode1)
summary(regression_P2Q2_Periode1)
summary(regression_P2Q4_Periode1)

summary(regression_P4Q1_Periode1)
summary(regression_P4Q2_Periode1)
summary(regression_P4Q4_Periode1)

summary(alfa_P1Q1_Periode1)
summary(alfa_P1Q2_Periode1)
summary(alfa_P1Q4_Periode1)

summary(alfa_P2Q1_Periode1)
summary(alfa_P2Q2_Periode1)
summary(alfa_P2Q4_Periode1)

summary(alfa_P4Q1_Periode1)
summary(alfa_P4Q2_Periode1)
summary(alfa_P4Q4_Periode1)


summary(regression_P1Q1_Periode2)
summary(regression_P1Q2_Periode2)
summary(regression_P1Q4_Periode2)

summary(regression_P2Q1_Periode2)
summary(regression_P2Q2_Periode2)
summary(regression_P2Q4_Periode2)

summary(regression_P4Q1_Periode2)
summary(regression_P4Q2_Periode2)
summary(regression_P4Q4_Periode2)

summary(alfa_P1Q1_Periode2)
summary(alfa_P1Q2_Periode2)
summary(alfa_P1Q4_Periode2)

summary(alfa_P2Q1_Periode2)
summary(alfa_P2Q2_Periode2)
summary(alfa_P2Q4_Periode2)
```

```
summary(alfa_P4Q1_Periode2)
summary(alfa_P4Q2_Periode2)
summary(alfa_P4Q4_Periode2)
```

Results to excel

```
##################-P4=K4-##################

ResultsP4Q4U <- do.call(rbind, lapply(names(ResultsP4Q4), function(x) {
  data.frame(Quarter = x, Performance = ResultsP4Q4[[x]])
}))

ResultsP4Q2U <- do.call(rbind, lapply(names(ResultsP4Q2), function(x) {
  data.frame(Quarter = x, Performance = ResultsP4Q2[[x]])
}))

ResultsP4Q1U <- do.call(rbind, lapply(names(ResultsP4Q1), function(x) {
  data.frame(Quarter = x, Performance = ResultsP4Q1[[x]])
}))

write.xlsx(ResultsP4Q1U, file = "Documents/høst 2023/ResultsP4Q1U.xlsx", rowNames = FALSE)
write.xlsx(ResultsP4Q4U, file = "Documents/høst 2023/ResultsP4Q4U.xlsx", rowNames = FALSE)
write.xlsx(ResultsP4Q2U, file = "Documents/høst 2023/ResultsP4Q2U.xlsx", rowNames = FALSE)


##################-P2=K2-##################

ResultsP2Q4U <- do.call(rbind, lapply(names(ResultsP2Q4), function(x) {
  data.frame(Quarter = x, Performance = ResultsP2Q4[[x]])
}))

ResultsP2Q2U <- do.call(rbind, lapply(names(ResultsP2Q2), function(x) {
  data.frame(Quarter = x, Performance = ResultsP2Q2[[x]])
}))

ResultsP2Q1U <- do.call(rbind, lapply(names(ResultsP2Q1), function(x) {
  data.frame(Quarter = x, Performance = ResultsP2Q1[[x]])
}))

write.xlsx(ResultsP2Q1U, file = "Documents/høst 2023/ResultsP2Q1U.xlsx", rowNames = FALSE)
write.xlsx(ResultsP2Q4U, file = "Documents/høst 2023/ResultsP2Q4U.xlsx", rowNames = FALSE)
write.xlsx(ResultsP2Q2U, file = "Documents/høst 2023/ResultsP2Q2U.xlsx", rowNames = FALSE)


##################-P1=K1-##################

ResultsP1Q4U <- do.call(rbind, lapply(names(ResultsP1Q4), function(x) {
  data.frame(Quarter = x, Performance = ResultsP1Q4[[x]])
}))
```

```r
ResultsP1Q2U <- do.call(rbind, lapply(names(ResultsP1Q2), function(x) {
  data.frame(Quarter = x, Performance = ResultsP1Q2[[x]])
}))

ResultsP1Q1U <- do.call(rbind, lapply(names(ResultsP1Q1), function(x) {
  data.frame(Quarter = x, Performance = ResultsP1Q1[[x]])
}))

write.xlsx(ResultsP1Q1U, file = "Documents/høst 2023/ResultsP1Q1U.xlsx", rowNames = FALSE)
write.xlsx(ResultsP1Q4U, file = "Documents/høst 2023/ResultsP1Q4U.xlsx", rowNames = FALSE)
write.xlsx(ResultsP1Q2U, file = "Documents/høst 2023/ResultsP1Q2U.xlsx", rowNames = FALSE)
```