Portfolio of Companies

Abdeali Arsiwala

2024-05-17

```
file_path <- "Earnings Quality Signals Analytics Project Data.xlsx"
financial_data <- read_excel(file_path, sheet = "Financial Statement Data")
head(financial_data)
```

GVKEY <dbl>	PERMNO <dbl>	Company Name <chr>	Ticker <chr>	Fiscal Year <dbl>	Current Assets <chr>
1004	54594	AAR CORP	AIR	2021	1007.2
1004	54594	AAR CORP	AIR	2022	1097.9000000000001
1004	54594	AAR CORP	AIR	2020	937
1045	21020	AMERICAN AIRLINES GROUP INC	AAL	2020	11095
1045	21020	AMERICAN AIRLINES GROUP INC	AAL	2023	13572
1045	21020	AMERICAN AIRLINES GROUP INC	AAL	2021	17336

6 rows | 1-6 of 27 columns

Filter the data to create dataframes of companies that have both 2020 and 2021 data and remove duplicates

```
# Filter financial data for 2020 and 2021
financial_data_2020 <- financial_data %>% filter(`Fiscal Year` == 2020)
financial_data_2021 <- financial_data %>% filter(`Fiscal Year` == 2021)

# Identify companies with financial data for both 2020 and 2021
common_permnos <- intersect(financial_data_2020$PERMNO, financial_data_2021$PERMNO)

# Filter the financial data for these companies
financial_data_2020 <- financial_data_2020 %>% filter(PERMNO %in% common_permnos)
financial_data_2021 <- financial_data_2021 %>% filter(PERMNO %in% common_permnos)

# Remove duplicates by keeping the first occurrence
financial_data_2020 <- financial_data_2020 %>%
  distinct(PERMNO, .keep_all = TRUE)

financial_data_2021 <- financial_data_2021 %>%
  distinct(PERMNO, .keep_all = TRUE)

# Verify dimensions
dim(financial_data_2020)
```

```
## [1] 7328   27
```

```
dim(financial_data_2021)
```

```
## [1] 7328   27
```

Merge and Clean Dataframe

```
# Merge the 2020 and 2021 financial data
financial_data_merged <- merge(financial_data_2021, financial_data_2020, by = c("GVKEY", "PERMNO", "Company Nam
e", "Ticker"), suffixes = c("_2021", "_2020"))

financial_data_merged <- financial_data_merged %>%
  select(-`Fiscal Year_2020`, -`Fiscal Year_2021`)

cols_to_convert <- setdiff(names(financial_data_merged), c("Company Name", "Ticker"))

financial_data_merged[cols_to_convert] <- lapply(financial_data_merged[cols_to_convert], as.numeric)

# Check the head of the merged dataframe
head(financial_data_merged)
```

	GVKEY <dbl>	PER... <dbl>	Company Name <chr>	Ticker <chr>	Current Assets_2021 <dbl>	Current Liabilities_2021 <dbl>
1	10000	39220	STANDARD MOTOR PRODS	SMP	711.205	476.369
2	10008	43481	STANDEX INTERNATIONAL CORP	SXI	374.366	143.709
3	10016	43350	STANLEY BLACK & DECKER INC	SWK	8526.400	8767.400
4	100243	18724	AMCOR PLC	AMCR	5266.000	4345.000
5	10030	18438	STARRETT (L.S.) CO -CL A	SCX	119.220	51.689
6	100338	81046	RELX PLC	RELX	3191.614	5075.167

6 rows | 1-7 of 49 columns

STEP 1 -> Calculate the Accrual Component for 2021

```
# Calculate changes (deltas) for required components
financial_data_merged <- financial_data_merged %>%
  mutate(
    delta_CA = `Current Assets_2021` - `Current Assets_2020`,
    delta_Cash = `Cash_2021` - `Cash_2020`,
    delta_CL = `Current Liabilities_2021` - `Current Liabilities_2020`,
    delta_STD = `Debt in Current Liabilities_2021` - `Debt in Current Liabilities_2020`,
    delta_TP = `Income Taxes Payable_2021` - `Income Taxes Payable_2020`,
    Avg_Total_Assets = (`Total Assets_2020` + `Total Assets_2021`) / 2

  )

# Calculate the Accrual Component
financial_data_merged <- financial_data_merged %>%
  mutate(
    Accrual_Component_2021 = (delta_CA - delta_Cash) - (delta_CL - delta_STD - delta_TP) - `Depreciation and Amor
tization_2021`
  )

# Calculate the Cash Component
financial_data_merged <- financial_data_merged %>%
  mutate(
    Cash_Component_2021 = `Operating Income After Depreciation_2021` - Accrual_Component_2021  + `Depreciation an
d Amortization_2021`
  )

dim(financial_data_merged)
```

```
## [1] 7244   56
```

```
# Drop rows with NA values in the Accrual_Component column
financial_data_merged <- financial_data_merged %>%
  filter(!is.na(Accrual_Component_2021) & !is.na(Cash_Component_2021) & !is.na(Avg_Total_Assets))

dim(financial_data_merged)
```

```
## [1] 3566   56
```

STEP 2 -> Standardize the Accrual Component

```
# Standardize the Accrual Component
financial_data_merged <- financial_data_merged %>%
  mutate(
    Per_Asset_Accrual_2021 = Accrual_Component_2021 / Avg_Total_Assets,
    Per_Asset_Cash_2021 = Cash_Component_2021 / Avg_Total_Assets,
    Per_Asset_Operating_2021 = `Operating Income After Depreciation_2021` / Avg_Total_Assets
  )

financial_data_merged <- financial_data_merged[,c(1:4, 57:59)]

dim(financial_data_merged)
```

```
## [1] 3566    7
```

```
head(financial_data_merged)
```

	GVKEY <dbl>	PER... <dbl>	Company Name <chr>	Ticker <chr>	Per_Asset_Accrual_2021 <dbl>	Per_Asset_Cash_2021 <dbl>
1	10000	39220	STANDARD MOTOR PRODS	SMP	0.027381746	0.11960171
2	10008	43481	STANDEX INTERNATIONAL CORP	SXI	-0.028837341	0.14420150
3	10016	43350	STANLEY BLACK & DECKER INC	SWK	0.050392009	0.04736184
4	10030	18438	STARRETT (L.S.) CO -CL A	SCX	0.002799795	0.12777145
5	100338	81046	RELX PLC	RELX	-0.044608092	0.22893685
6	1004	54594	AAR CORP	AIR	0.014709661	0.08119219

6 rows | 1-7 of 8 columns

Calculate Size-Adjusted Cumulative Abnormal Returns

```
company_stock_returns <- read_excel(file_path, sheet = "Company Stock Returns", col_names = TRUE)
size_matched_returns <- read_excel(file_path, sheet = "Size-matched Stock Returns", col_names = TRUE)

# Specify column names
column_names <- c("PERMNO", "Company Name", "Ticker",
                  "Jan_2020", "Feb_2020", "Mar_2020", "Apr_2020",
                  "May_2020", "Jun_2020", "Jul_2020", "Aug_2020",
                  "Sep_2020", "Oct_2020", "Nov_2020", "Dec_2020",
                  "Jan_2021", "Feb_2021", "Mar_2021", "Apr_2021",
                  "May_2021", "Jun_2021", "Jul_2021", "Aug_2021",
                  "Sep_2021", "Oct_2021", "Nov_2021", "Dec_2021",
                  "Jan_2022", "Feb_2022", "Mar_2022", "Apr_2022",
                  "May_2022", "Jun_2022", "Jul_2022", "Aug_2022",
                  "Sep_2022", "Oct_2022", "Nov_2022", "Dec_2022",
                  "Jan_2023", "Feb_2023", "Mar_2023", "Apr_2023",
                  "May_2023", "Jun_2023", "Jul_2023", "Aug_2023",
                  "Sep_2023", "Oct_2023", "Nov_2023", "Dec_2023")

# Set column names
colnames(company_stock_returns) <- column_names
colnames(size_matched_returns) <- column_names

head(company_stock_returns)
```

PER... <dbl>	Company Name <chr>	Ticker <chr>	Jan_2020 <chr>	Feb_2020 <chr>
10026	J & J SNACK FOODS CORP	JJSF	-0.10001599999999999	-3.0269999999999998E-2
10028	ENVELA CORP	ELA	0.60740700000000003	0.22580600000000001
10032	PLEXUS CORP	PLXS	-7.5643000000000002E-2	-6.7070000000000005E-2
10044	ROCKY MOUNTAIN CHOC FACT INC	RMCF	-9.8591999999999999E-2	-6.4904000000000003E-2
10051	HANGER INC	HNGR	-0.115176	-5.5669000000000003E-2
10065	ADAMS DIVERSIFIED EQUITY FD	ADX	5.7070000000000003E-3	-9.0163999999999994E-2

6 rows | 1-5 of 51 columns

```
# Select the necessary columns for calculation
required_columns <- c("PERMNO", "Company Name", "Ticker",
                      "May_2022", "Jun_2022", "Jul_2022", "Aug_2022",
                      "Sep_2022", "Oct_2022", "Nov_2022", "Dec_2022",
                      "Jan_2023", "Feb_2023", "Mar_2023", "Apr_2023")

company_stock_returns <- company_stock_returns[, required_columns]
size_matched_returns <- size_matched_returns[, required_columns]


# Merge the two datasets
merged_returns <- merge(company_stock_returns, size_matched_returns, by = c("PERMNO", "Company Name", "Ticker"),
suffixes = c("_Individual", "_Size_Matched"))

dim(merged_returns)
```

```
## [1] 2772   27
```

```
# Remove duplicates by keeping the first occurrence
merged_returns <- merged_returns %>%
  distinct(PERMNO, .keep_all = TRUE)

dim(merged_returns)
```

```
## [1] 2772   27
```

```
# Convert columns to numeric
cols_to_convert <- setdiff(names(merged_returns), c("Company Name", "Ticker"))
merged_returns[cols_to_convert] <- lapply(merged_returns[cols_to_convert], as.numeric)

# Drop NA values except for "Company Name" and "Ticker"
merged_returns <- merged_returns %>%
  drop_na(-`Company Name`, -`Ticker`)

# Display the result
dim(merged_returns)
```

```
## [1] 2414   27
```

```
head(merged_returns)
```

	PERMNO <dbl>	Company Name <chr>	Ticker <chr>	May_2022_Individual <dbl>	Jun_2022_Individual <dbl>
1	10026	J & J SNACK FOODS CORP	JJSF	-0.143487	0.094159
2	10032	PLEXUS CORP	PLXS	0.045107	-0.074293
3	10104	ORACLE CORP	ORCL	-0.020164	-0.028504
4	10107	MICROSOFT CORP	MSFT	-0.018125	-0.055321
5	10158	AMERESCO INC	AMRC	0.164155	-0.224114
6	10200	REPLIGEN CORP	RGEN	0.045981	-0.012586

6 rows | 1-6 of 28 columns

```
# Define a function to calculate cumulative returns
calculate_cumulative_returns <- function(df, start_col, end_col) {
  # Convert column names to numeric
  start_col_index <- match(start_col, colnames(df))
  end_col_index <- match(end_col, colnames(df))

  # Calculate cumulative returns
  returns <- apply(df[, start_col_index:end_col_index], 1, function(row) prod(1 + row))
  returns
}

# Calculate Cumulative_Return_Individual
merged_returns$Cumulative_Return_Individual <- calculate_cumulative_returns(
  merged_returns, "May_2022_Individual", "Apr_2023_Individual"
)

# Calculate Cumulative_Return_Size_Matched
merged_returns$Cumulative_Return_Size_Matched <- calculate_cumulative_returns(
  merged_returns, "May_2022_Size_Matched", "Apr_2023_Size_Matched"
)

# Calculate Size_Adjusted_Abnormal_Returns
merged_returns$Size_Adjusted_Abnormal_Returns <- merged_returns$Cumulative_Return_Individual - merged_returns$Cum
ulative_Return_Size_Matched

merged_returns <- merged_returns[, c("PERMNO", "Company Name", "Ticker", "Cumulative_Return_Individual", "Cumulat
ive_Return_Size_Matched", "Size_Adjusted_Abnormal_Returns")]

# Display the result
head(merged_returns)
```

	PERMNO <dbl>	Company Name <chr>	Ticker <chr>	Cumulative_Return_Individual <dbl>
1	10026	J & J SNACK FOODS CORP	JJSF	1.0433136
2	10032	PLEXUS CORP	PLXS	1.0780114
3	10104	ORACLE CORP	ORCL	1.3112995
4	10107	MICROSOFT CORP	MSFT	1.1183256
5	10158	AMERESCO INC	AMRC	0.8247409
6	10200	REPLIGEN CORP	RGEN	0.9643223

6 rows | 1-5 of 7 columns

```
merged_final <- merge(financial_data_merged, merged_returns, by = c("PERMNO", "Company Name", "Ticker"))
dim(merged_final)
```

```
## [1] 1472   10
```

```
head(merged_final)
```

	PER... <dbl>	Company Name <chr>	Ticker <chr>	GVKEY <dbl>	Per_Asset_Accrual_2021 <dbl>	Per_Asset_Cash_2021 <dbl>
1	10026	J & J SNACK FOODS CORP	JJSF	12825	-0.07803478	0.18991615
2	10032	PLEXUS CORP	PLXS	12945	0.00167728	0.09907021
3	10104	ORACLE CORP	ORCL	12142	-0.15079616	0.30843081
4	10107	MICROSOFT CORP	MSFT	12141	-0.06436883	0.31887134
5	10158	AMERESCO INC	AMRC	185128	0.01791062	0.05427222
6	10200	REPLIGEN CORP	RGEN	12181	0.03104776	0.07769615

6 rows | 1-7 of 11 columns

Forming the Long/Short Portfolio

```
# Sort the merged_final dataframe based on Per Asset Accruals
portfolio <- merged_final[order(merged_final$Per_Asset_Accrual), ]

# Select the top 25 companies with the lowest Per Asset Accruals (L group)
L_group <- portfolio[1:25, ]

# Select the top 25 companies with the highest Per Asset Accruals (H group)
H_group <- portfolio[(nrow(portfolio) - 24):nrow(portfolio), ]

# Calculate average Per Asset Accrual for L and H groups
avg_accrual_L <- mean(L_group$Per_Asset_Accrual)
avg_accrual_H <- mean(H_group$Per_Asset_Accrual)

# Calculate average Size-Adjusted Cumulative Abnormal Returns for L and H groups
avg_returns_L <- mean(L_group$Size_Adjusted_Abnormal_Returns)
avg_returns_H <- mean(H_group$Size_Adjusted_Abnormal_Returns)

# Compute Hedge Return
hedge_return <- avg_returns_L - avg_returns_H

# Print results
cat("Average Per Asset Accrual for L group:", avg_accrual_L, "\n")
```

```
## Average Per Asset Accrual for L group: -0.5511338
```

```
cat("Average Per Asset Accrual for H group:", avg_accrual_H, "\n")
```

```
## Average Per Asset Accrual for H group: 0.7991447
```

```
cat("Average Size-Adjusted Cumulative Abnormal Returns for L group:", avg_returns_L, "\n")
```

```
## Average Size-Adjusted Cumulative Abnormal Returns for L group: -0.03263934
```

```
cat("Average Size-Adjusted Cumulative Abnormal Returns for H group:", avg_returns_H, "\n")
```

```
## Average Size-Adjusted Cumulative Abnormal Returns for H group: -0.1633618
```

```
cat("Hedge Return:", hedge_return, "\n")
```

```
## Hedge Return: 0.1307225
```

Descriptive Statistics

```
summary(merged_final[,c("Per_Asset_Accrual_2021","Per_Asset_Cash_2021","Per_Asset_Operating_2021")])
```

```
##  Per_Asset_Accrual_2021 Per_Asset_Cash_2021 Per_Asset_Operating_2021
##  Min.   :-0.81633       Min.   :-4.72145    Min.   :-4.58890
##  1st Qu.:-0.06662       1st Qu.:-0.17963    1st Qu.:-0.20321
##  Median :-0.02405       Median : 0.07239    Median : 0.01593
##  Mean   :-0.00211       Mean   :-0.05267    Mean   :-0.08502
##  3rd Qu.: 0.02958       3rd Qu.: 0.16100    3rd Qu.: 0.09231
##  Max.   : 1.19470       Max.   : 1.97188    Max.   : 1.76285
```

Satitistics of Lower Group Portfolio

```
cat("STATISTICS FOR THE LOWER GROUP: \n\n")
```

```
## STATISTICS FOR THE LOWER GROUP:
```

```
summary(L_group[,c("Per_Asset_Accrual_2021","Per_Asset_Cash_2021","Per_Asset_Operating_2021", "Size_Adjusted_Abno
rmal_Returns")])
```

```
##  Per_Asset_Accrual_2021 Per_Asset_Cash_2021 Per_Asset_Operating_2021
##  Min.   :-0.8163        Min.   :-1.23591    Min.   :-1.6895
##  1st Qu.:-0.6717        1st Qu.:-0.11580    1st Qu.:-0.6062
##  Median :-0.5007        Median : 0.11779    Median :-0.4759
##  Mean   :-0.5511        Mean   : 0.07276    Mean   :-0.4916
##  3rd Qu.:-0.4328        3rd Qu.: 0.25007    3rd Qu.:-0.2941
##  Max.   :-0.3774        Max.   : 0.83221    Max.   : 0.3984
##  Size_Adjusted_Abnormal_Returns
##  Min.   :-0.79826
##  1st Qu.:-0.42555
##  Median :-0.16595
##  Mean   :-0.03264
##  3rd Qu.: 0.10643
##  Max.   : 3.89787
```

Satitistics of Higher Group Portfolio

```
cat("STATISTICS FOR THE HIGHER GROUP: \n\n")
```

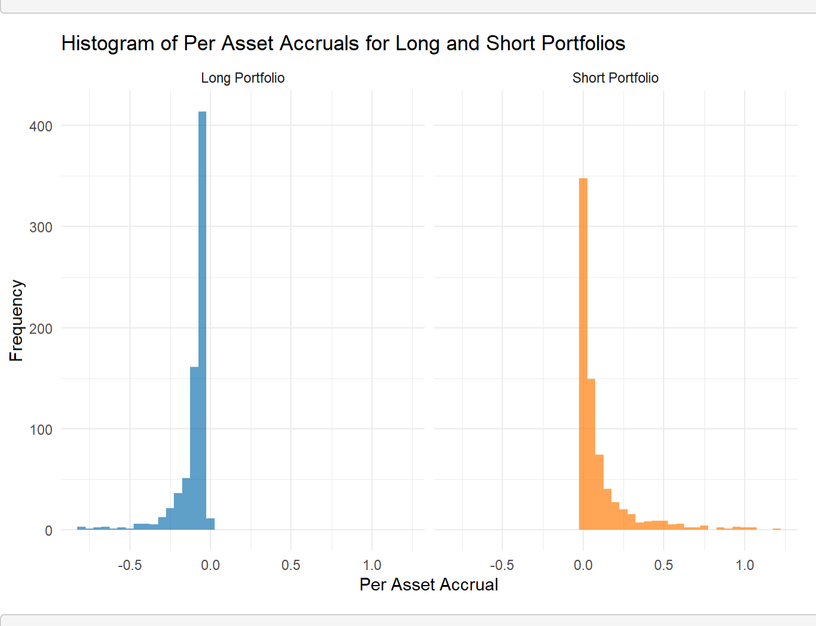
```
## STATISTICS FOR THE HIGHER GROUP:
```

```
summary(H_group[,c("Per_Asset_Accrual_2021","Per_Asset_Cash_2021","Per_Asset_Operating_2021", "Size_Adjusted_Abno
rmal_Returns")])
```

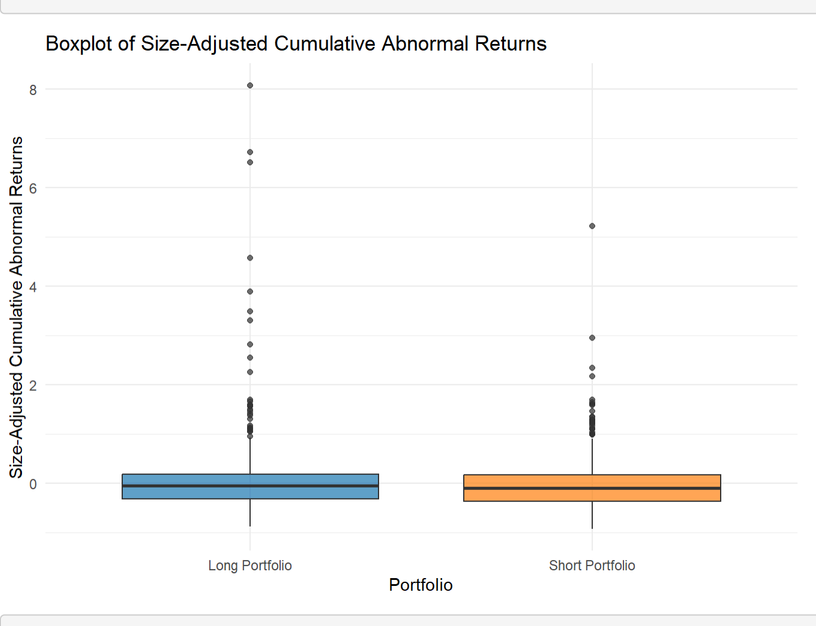
```
##  Per_Asset_Accrual_2021 Per_Asset_Cash_2021 Per_Asset_Operating_2021
##  Min.   :0.5768         Min.   :-1.8034     Min.   :-0.8149
##  1st Qu.:0.6381         1st Qu.:-1.2583     1st Qu.:-0.4061
##  Median :0.7635         Median :-1.0131     Median :-0.2386
##  Mean   :0.7991         Mean   :-1.0536     Mean   :-0.2590
##  3rd Qu.:0.9601         3rd Qu.:-0.8527     3rd Qu.:-0.1343
##  Max.   :1.1947         Max.   :-0.4428     Max.   : 0.1365
##  Size_Adjusted_Abnormal_Returns
##  Min.   :-0.7824
##  1st Qu.:-0.4945
##  Median :-0.3852
##  Mean   :-0.1634
##  3rd Qu.: 0.2027
##  Max.   : 1.0054
```

Visualizations

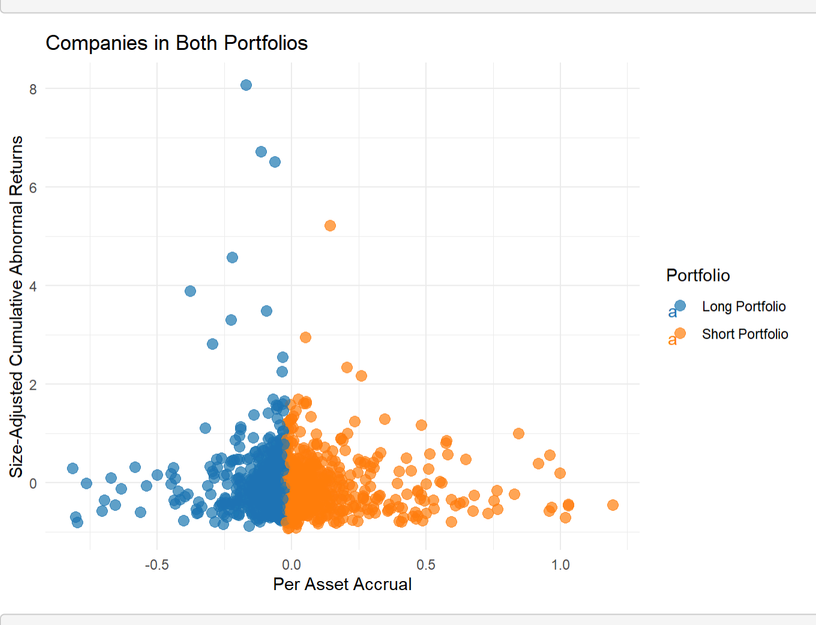
```
# Add a column to indicate portfolio group
merged_final <- merged_final %>%
  mutate(Portfolio = ifelse(Per_Asset_Accrual_2021 < median(Per_Asset_Accrual_2021), "Long Portfolio", "Short Por
tfolio"))

# Histogram of Per Asset Accruals for Long and Short portfolios
ggplot(merged_final, aes(x = Per_Asset_Accrual_2021, fill = Portfolio)) +
  geom_histogram(binwidth = 0.05, alpha = 0.7, position = "identity") +
  scale_fill_manual(values = c("Long Portfolio" = "#1f77b4", "Short Portfolio" = "#ff7f0e")) +
  facet_wrap(~ Portfolio) +
  labs(title = "Histogram of Per Asset Accruals for Long and Short Portfolios", x = "Per Asset Accrual", y = "Fre
quency") +
  theme_minimal() +
  theme(legend.position = "none")
```




```
# Boxplot of Size-Adjusted Cumulative Abnormal Returns for Long and Short portfolios
ggplot(merged_final, aes(x = Portfolio, y = Size_Adjusted_Abnormal_Returns, fill = Portfolio)) +
  geom_boxplot(alpha = 0.7) +
  scale_fill_manual(values = c("Long Portfolio" = "#1f77b4", "Short Portfolio" = "#ff7f0e")) +
  labs(title = "Boxplot of Size-Adjusted Cumulative Abnormal Returns", x = "Portfolio", y = "Size-Adjusted Cumula
tive Abnormal Returns") +
  theme_minimal() +
  theme(legend.position = "none")
```




```
# Scatter plot of Per Asset Accrual vs. Size-Adjusted Cumulative Abnormal Returns for companies in both portfolio
s
ggplot(merged_final, aes(x = Per_Asset_Accrual_2021, y = Size_Adjusted_Abnormal_Returns, color = Portfolio, label
= `Company Name`)) +
  geom_point(alpha = 0.7, size = 3) +
  geom_text(aes(label=ifelse(Portfolio == "Long Portfolio" & Size_Adjusted_Abnormal_Returns > 1, as.character(`Co
mpany Name`),'')), hjust=0, vjust=0, check_overlap = TRUE) +
  scale_color_manual(values = c("Long Portfolio" = "#1f77b4", "Short Portfolio" = "#ff7f0e")) +
  labs(title = "Companies in Both Portfolios", x = "Per Asset Accrual", y = "Size-Adjusted Cumulative Abnormal Re
turns") +
  theme_minimal()
```




```
library(kableExtra)

# Create a table for the Long Portfolio
long_portfolio_table <- L_group %>%
  select(`Company Name`, `Ticker`, Per_Asset_Accrual_2021, `Size_Adjusted_Abnormal_Returns`) %>%
  arrange(Per_Asset_Accrual_2021)

kable(long_portfolio_table, format = "markdown", caption = "Long Portfolio: Companies with Lowest Per Asset Accru
al") %>%
  kable_styling(latex_options = c("striped", "hold_position"), full_width = F)
```

Long Portfolio: Companies with Lowest Per Asset Accrual

Company Name	Ticker	Per_Asset_Accrual_2021	Size_Adjusted_Abnormal_Returns
SPECTRUM PHARMACEUTICALS INC	SPPI	-0.8163297	0.2979816
SIYATA MOBILE INC	SYTA	-0.8046241	-0.6942267
AGILE THERAPEUTICS INC	AGRX	-0.7975088	-0.7982572
EMERSON RADIO CORP	MSN	-0.7633621	-0.0115275
EXICURE INC	XCUR	-0.7063613	-0.5640285
ATRECA INC	BCEL	-0.6980626	-0.3543373
MALACCA STRAITS ACQ CO LTD	MLAC	-0.6716796	0.1064311
DATASEA INC	DTSS	-0.6569028	-0.4405528
ALLOVIR INC	ALVR	-0.6338401	-0.1190435
BLUEBIRD BIO INC	BLUE	-0.5839026	0.3232583
WORKHORSE GROUP INC	WKHS	-0.5623617	-0.5917117
GENERATION BIO CO	GBIO	-0.5404332	-0.0589034
ELECTROCORE INC	ECOR	-0.5006792	0.1611900
GEOVAX LABS INC	GOVX	-0.4500924	-0.0144665
VOYAGER THERAPEUTICS INC	VYGR	-0.4482313	0.1935787
SILK ROAD MEDICAL INC	SILK	-0.4404431	0.3086590
KALA BIO INC	KALA	-0.4369462	-0.3449764
CORCEPT THERAPEUTICS INC	CORT	-0.4331448	0.0866449
ITEOS THERAPEUTICS INC	ITOS	-0.4327788	-0.4255481
CIBUS INC	CBUS	-0.4212851	-0.1659528
COMMUNICATIONS SYSTEMS INC	JCS	-0.4156552	-0.3399637
POLYPID LTD	PYPD	-0.4031578	-0.7679706
NOVA LIFESTYLE INC	NVFY	-0.3970319	-0.2782562
KARYOPHARM THERAPEUTICS INC	KPTI	-0.3860921	-0.2218698
CYMABAY THERAPEUTICS INC	CBAY	-0.3774383	3.8978660

```
# Create a table for the Short Portfolio
short_portfolio_table <- H_group %>%
  select(`Company Name`, `Ticker`, Per_Asset_Accrual_2021, `Size_Adjusted_Abnormal_Returns`) %>%
  arrange(desc(Per_Asset_Accrual_2021))

kable(short_portfolio_table, format = "markdown", caption = "Short Portfolio: Companies with Highest Per Asset Ac
crual") %>%
  kable_styling(latex_options = c("striped", "hold_position"), full_width = F)
```

Short Portfolio: Companies with Highest Per Asset Accrual

Company Name	Ticker	Per_Asset_Accrual_2021	Size_Adjusted_Abnormal_Returns
ATYR PHARMA INC	LIFE	1.1947000	-0.4403999
RED CAT HOLDINGS INC	RCAT	1.0293680	-0.4285803
BIONANO GENOMICS INC	BNGO	1.0290801	-0.4530626
FREIGHT TECHNOLOGIES INC	FRGT	1.0179584	-0.6986785
BRILLIANT ACQUISITION CORP	BRLI	0.9988877	0.2027016
ARVINAS INC	ARVN	0.9675789	-0.4945299
PRELUDE THERAPEUTICS INC	PRLD	0.9600596	0.5712507
LIPOCINE INC	LPCN	0.9574613	-0.5697229
CARDIFF ONCOLOGY INC	CRDF	0.9189048	0.3893586
OLEMA PHARMACEUTICALS INC	OLMA	0.8445935	1.0054104
ACLARIS THERAPEUTICS INC	ACRS	0.8286526	-0.2214482
KARTOON STUDIOS INC	TOON	0.7655863	-0.5278169
ASTROTECH CORP	ASTC	0.7634951	-0.1550311
INSPIREMD INC	NSPR	0.7519004	-0.3637906
FULCRUM THERAPEUTICS INC	FULC	0.7302819	-0.6152121
URANIUM ROYALTY CORP	UROY	0.6784900	-0.2515849
GEVO INC	GEVO	0.6756449	-0.5624186
XENON PHARMACEUTICALS INC	XENE	0.6475388	0.4775684
OCEAN POWER TECHNOLOGIES INC	OPTT	0.6380547	-0.3852015
ENLIVEX THERAPEUTICS LTD	ENLV	0.6245424	-0.4058380
EVOKE PHARMA INC	EVOK	0.6097866	-0.4644639
FARMMI INC	FAMI	0.5947645	-0.7823975
FOGHORN THERAPEUTICS INC	FHTX	0.5944893	-0.3395985
IMMERSION CORP	IMMR	0.5799521	0.5759905
CELSIUS HOLDINGS INC	CELH	0.5768457	0.8534504

Conclusion

The analysis aimed to investigate the earnings quality signal by calculating the accrual component of operating income and constructing a long/short portfolio based on this metric. The descriptive statistics provided insights into the financial characteristics and performance of the sampled firms, while the performance of the portfolios gave a clearer picture of the strategy's effectiveness.

Key Findings:

1. **Per Asset Accrual and Portfolio Formation:**
   - **Long Portfolio (L group):** Consisted of firms with the lowest Per Asset Accrual, averaging -0.5511. The average Size-Adjusted Cumulative Abnormal Return was -0.0326.
   - **Short Portfolio (H group):** Comprised of firms with the highest Per Asset Accrual, averaging 0.7991. The average Size-Adjusted Cumulative Abnormal Return was -0.1634.
   - **Hedge Return:** Calculated as the difference between the average Size-Adjusted Cumulative Abnormal Returns of the L and H groups, yielding a positive hedge return of 0.1307.
2. **Descriptive Statistics:**
   - The sample included a wide range of Per Asset Accrual values, from -0.8163 to 1.1947, with a mean of -0.0021.
   - Operating Income and Cash Flow metrics also exhibited significant variability, indicating diverse financial performance across firms.

Portfolio Companies:

- **Long Portfolio (L group):** Includes companies like Spectrum Pharmaceuticals Inc (SPPI), Siyata Mobile Inc (SYTA), and Agile Therapeutics Inc (AGRX).
- **Short Portfolio (H group):** Includes companies like Celsius Holdings Inc (CELH), Immersion Corp (IMMR), and Foghorn Therapeutics Inc (FHTX).

Interpretation:

- **Performance of Long Portfolio (L group):** The firms with the lowest accruals, while expected to perform better due to high earnings quality, yielded an average negative Size-Adjusted Cumulative Abnormal Return of -0.0326.
- **Performance of Short Portfolio (H group):** Firms with the highest accruals, typically seen as having lower earnings quality, also had negative returns but significantly lower at -0.1634.
- **Hedge Return:** A positive hedge return of 0.1307 indicates that a strategy of going long on the L group and short on the H group could yield positive returns despite both groups having negative average returns individually.

Insights:

- The positive hedge return supports the hypothesis that firms with lower accruals (L group) tend to outperform those with higher accruals (H group) in terms of risk-adjusted returns.
- The analysis confirms the potential utility of accrual-based metrics in constructing effective long/short equity strategies.
- Further investigation could enhance the robustness of the strategy by incorporating additional financial metrics and testing over different time periods or market conditions.

Limitations and Future Research:

- The presence of outliers and extreme values in both groups highlights the need for robust statistical methods to handle such anomalies.
- Missing values and data quality issues should be addressed to improve the accuracy of the analysis.
- Expanding the dataset and incorporating more firms across different sectors could provide more generalized insights.
- Further research could involve exploring other financial indicators or a multi-factor model to enhance the predictive power of the investment strategy.

Overall, this project demonstrates the application of financial analytics in constructing investment strategies and provides valuable insights into the relationship between accruals and stock performance.