Sector Rotation Strategy

Ronan Anderson

1) Summary of Task & Data

This file presents a long-only sector rotation strategy derived from a hierarchical clustering analysis. This strategy aims to outperform the market and is developed using only historical sector returns (i.e., no macro data, etc. used). The data used for this task is taken from the Kenneth French website (https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html).

The sector data used is the value-weighted returns from the 49 Industry Portfolios [Daily] data set and the market data is taken from the Fama/French 3 Factors [Daily] data set. Note that the risk-free rate, SMB and HML variables are also used.

2) Load in Data Set & Clean

After cutting the headings and equal-weighted returns from the sector returns data set, the returns were loaded. This file includes 49 different sectors composed from stocks on the NYSE, AMEX, and NASDAQ. The sectors daily returns begin in July 1, 1926 and as of this task, end in June 30, 2022. As indicated in the file, missing values are represented as "-0.9999"; therefore, these were assigned NA.

```
sectors = read.csv("49_Industry_Portfolios_Daily_Value.csv", na.strings=c(""," ","NA"),
                   stringsAsFactors=FALSE)

sectors$Date <- as.character(sectors$Date)
sectors$Date <- as.Date(sectors$Date, format = "%Y%m%d")
Date <- sectors$Date
sectors <- cbind(Date, sectors[2:50]/100) # returns converted to decimal.

sectors[2:50][sectors[2:50] == "-0.9999"] <- NA
```

For each trade in our sector rotation strategy to be effective, we must alot an adequate amount of time to each position. We are looking for a period long enough to overcome temporary volatility, but short enough to react to recent market trends. Therefore, we will work with quarterly periods. With this in mind, we convert our daily returns to quarterly returns.

```
sectors$Quarter <- as.yearqtr(sectors$Date)
sectors_qtrly <- sectors[2:51] %>%
  group_by(Quarter) %>%
  summarize(Agric=sum(Agric), Food=sum(Food), Soda=sum(Soda), Beer=sum(Beer),
            Smoke=sum(Smoke), Toys=sum(Toys), Fun=sum(Fun), Books=sum(Books),
            Hshld=sum(Hshld), Clths=sum(Clths), Hlth=sum(Hlth), MedEq=sum(MedEq),
            Drugs=sum(Drugs), Chems=sum(Chems), Rubbr=sum(Rubbr), Txtls=sum(Txtls),
            BldMt=sum(BldMt), Cnstr=sum(Cnstr), Steel=sum(Steel), FabPr=sum(FabPr),
            Mach=sum(Mach), ElcEq=sum(ElcEq), Autos=sum(Autos), Aero=sum(Aero),
            Ships=sum(Ships), Guns=sum(Guns), Gold=sum(Gold), Mines=sum(Mines),
```

```
            Coal=sum(Coal), Oil=sum(Oil), Util=sum(Util), Telcm=sum(Telcm),
            PerSv=sum(PerSv), BusSv=sum(BusSv), Hardw=sum(Hardw), Softw=sum(Softw),
            Chips=sum(Chips), LabEq=sum(LabEq), Paper=sum(Paper), Boxes=sum(Boxes),
            Trans=sum(Trans), Whlsl=sum(Whlsl), Rtail=sum(Rtail), Meals=sum(Meals),
            Banks=sum(Banks), Insur=sum(Insur), RlEst=sum(RlEst), Fin=sum(Fin),
            Other=sum(Other), .groups = 'drop')
```

Now, before we start into developing our strategy, we must separate our data into a training set, which the strategy can be developed with, and a test set, which the strategy will be tested on. In total, there are 384 quarters of which we will leave the last 30 as our test set (2015 Q1 to 2022 Q2).

```
training <- sectors_qtrly[1:354,]
test <-  sectors_qtrly[355:384,]
```

3) Clustering the Training Set into Portfolios

Hierarchical clustering is chosen for this task because it is less sensitive to outliers than other clustering methods (e.g., k-means). The clusters produced from other methods may be distorted if outliers are present.

Clustering techniques are mostly designed for cross-sectional data; therefore, we transpose the data to have the sectors on the rows and the time periods on the columns. You could think of this as each quarter being a different variable. This is stored in the data frame 't_training'.

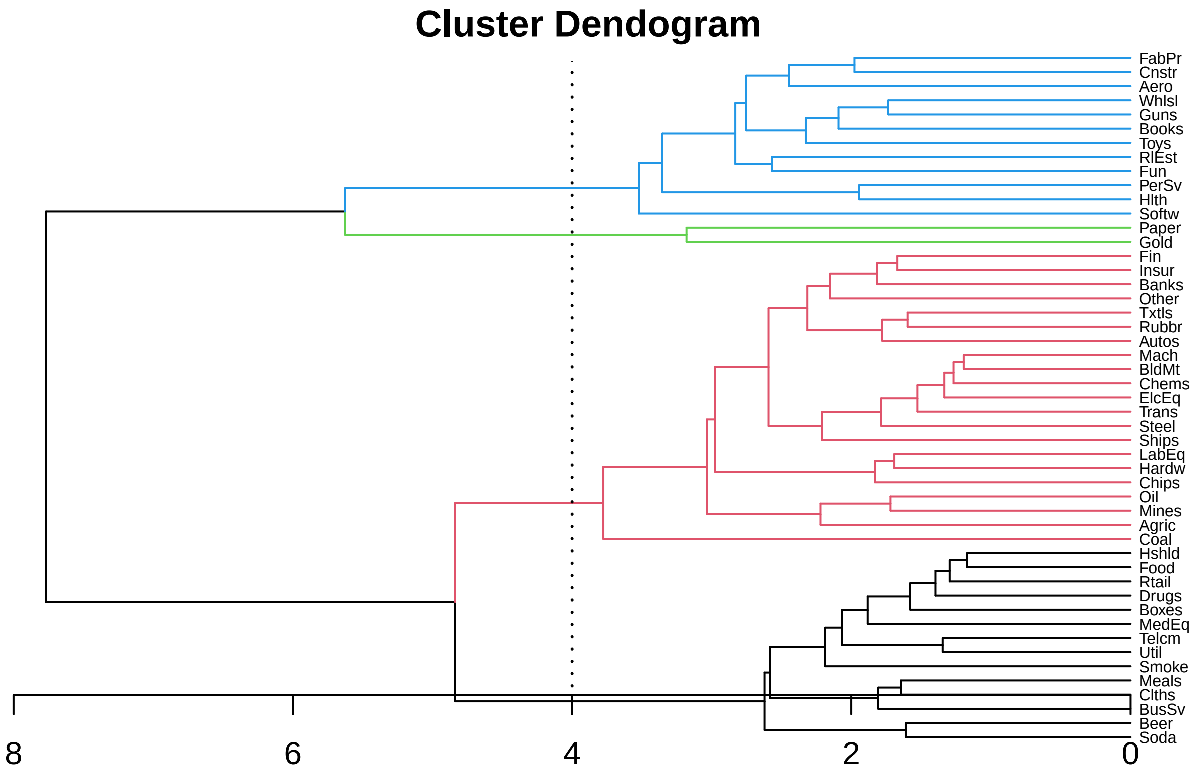
```
t_training <- data.frame(t(training))

colnames(t_training) <- t_training[1,]
t_training <- t_training[-1,]

Sector <- rownames(t_training)
rownames(t_training) <- 1:dim(t_training)[1]
t_training <- cbind(Sector, t_training)
```

One issue with Hierarchical clustering is that is can present dendograms with a lot of chaining. Typically, Wards method is used to produce clear clusters and this method is most commonly used in pair with the Euclidean dissimilarity measure. Single, Complete and Average linkage methods were also tried; however, Ward's linkage method gave a significantly clearer dendogram than the others.

The dendogram given presents an appropriate cut around x=4, which leaves us with four clusters.

```
dist.mat = dist(t_training, method="euclidean")
cl.average <- hclust(dist.mat, method = "ward.D")

dend <- as.dendrogram(cl.average)

Order <- 1:49
frame <- data.frame(Sector = NA)

for(i in 1:49){
  j <- as.numeric(labels(dend)[i])
  frame[i, "Sector"] <- t_training$Sector[j]
}
labels(dend) <- frame$Sector

dend <- color_branches(dend, 4, col = c(1, 2, 3, 4))
```

```r
par(cex=0.5, mar=c(4, 2, 2, 2))
plot(dend, horiz=T, axes=FALSE)
par(cex=1, mar=c(4, 2, 2, 2))
title(main = 'Cluster Dendogram')
axis(1)
abline(v = 4, col = "black", lwd = 1.5, lty = 3)
```



We can then summarise these clusters and analyse their sectors.

```r
hcl.average <- cutree(cl.average, h = 4)
table(hcl.average)
```

```
## hcl.average
##  1  2  3  4
## 21 14 12  2
```

Cluster 1 contains 21 sectors, Cluster 2 contains 14 sectors, Cluster 3 contains 12 sectors and Cluster 4 contains 2 sectors. If we analyse the sectors in each cluster/portfolio, there does not appear to be any clear similarities between the sectors. For example, we might expect a portfolio that is mainly comprised of primary sectors or one that is predominantly services.

Perhaps if we revisit the data and only use a time period where all sectors have an equal period of returns (i.e., one with no NA's), this would give the clustering method a better chance of clustering similar sectors together. It would also be more appropriate to have consistency across the time period under study.

4) Modifying the Training Set

The test set will remain as is. For the training set, we need to establish the quarter in which the last NA appears.

```
nas <- which(is.na(sectors_qtrly), arr.ind=TRUE)
training2_start <- max(nas[,1]) + 1; training2_start
```

```
## [1] 173
```

```
training2_end <- dim(training)[1]; training2_end
```

```
## [1] 354
```

```
paste0(sectors_qtrly[training2_start, 1])
```

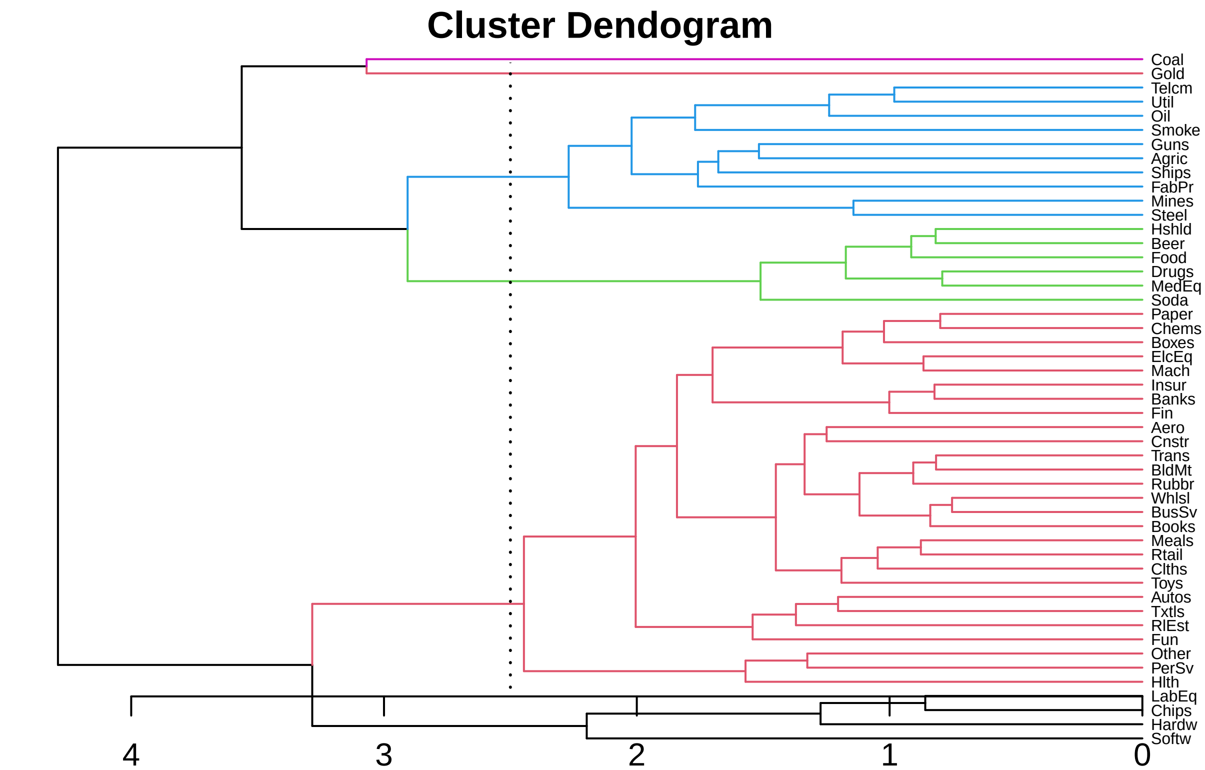
```
## [1] "1969.5"
```

The last NA appears in row 172 (1969 Q2/1969.25), which means that the modified trainings set will start in row 173 and end in row 354 (i.e, 1969 Q3 to 2014 Q4).

We transpose the data frame and repeat the same clustering process as previous. The dendogram of this modified data set is presented below, where a cut at x=2.5 appears most appropriate.

```
training2 <- sectors_qtrly[training2_start:training2_end,]

t_training2 <- data.frame(t(training2))

colnames(t_training2) <- t_training2[1,]
t_training2 <- t_training2[-1,]

Sector <- rownames(t_training2)
rownames(t_training2) <- 1:dim(t_training2)[1]
t_training2 <- cbind(Sector, t_training2)
```

```
dist.mat2 = dist(t_training2, method="euclidean")
cl.average2 <- hclust(dist.mat2, method = "ward.D")

dend2 <- as.dendrogram(cl.average2)

Order <- 1:49
frame <- data.frame(Sector = NA)

for(i in 1:49){
  j <- as.numeric(labels(dend2)[i])
  frame[i, "Sector"] <- t_training2$Sector[j]
}

labels(dend2) <- frame$Sector
dend2 <- color_branches(dend2, 6, col = c(1, 2, 3, 4, 10, 6))
par(cex=0.5, mar=c(4, 2, 2, 2))
plot(dend2, horiz=T, axes=FALSE)
par(cex=1, mar=c(4, 2, 2, 2))
title(main = 'Cluster Dendogram')
axis(1)
abline(v = 2.5, col = "black", lwd = 1.5, lty = 3)
```



A cut at this position gives us six different clusters (portfolios). This time, we have two outliers that have been assigned to their own portfolio. Their quarterly returns must be very different from the other sectors. Once again, lets analyse these to see if they offer a clearer pattern than before.

```
hcl.average2    <- cutree(cl.average2, h = 2.5)
table(hcl.average2)
```

```
## hcl.average2
##  1  2  3  4  5  6
## 10  6 27  1  1  4
```

```
x <- xtabs(~t_training2$Sector + hcl.average2); x
```

```
##                    hcl.average2
## t_training2$Sector 1 2 3 4 5 6
##              Aero  0 0 1 0 0 0
##              Agric 1 0 0 0 0 0
##              Autos 0 0 1 0 0 0
##              Banks 0 0 1 0 0 0
##              Beer  0 1 0 0 0 0
##              BldMt 0 0 1 0 0 0
##              Books 0 0 1 0 0 0
##              Boxes 0 0 1 0 0 0
##              BusSv 0 0 1 0 0 0
##              Chems 0 0 1 0 0 0
##              Chips 0 0 0 0 0 1
##              Clths 0 0 1 0 0 0
##              Cnstr 0 0 1 0 0 0
##              Coal  0 0 0 0 1 0
##              Drugs 0 1 0 0 0 0
##              ElcEq 0 0 1 0 0 0
```

```
##               FabPr 1 0 0 0 0 0
##               Fin   0 0 1 0 0 0
##               Food  0 1 0 0 0 0
##               Fun   0 0 1 0 0 0
##               Gold  0 0 0 1 0 0
##               Guns  1 0 0 0 0 0
##               Hardw 0 0 0 0 0 1
##               Hlth  0 0 1 0 0 0
##               Hshld 0 1 0 0 0 0
##               Insur 0 0 1 0 0 0
##               LabEq 0 0 0 0 0 1
##               Mach  0 0 1 0 0 0
##               Meals 0 0 1 0 0 0
##               MedEq 0 1 0 0 0 0
##               Mines 1 0 0 0 0 0
##               Oil   1 0 0 0 0 0
##               Other 0 0 1 0 0 0
##               Paper 0 0 1 0 0 0
##               PerSv 0 0 1 0 0 0
##               RlEst 0 0 1 0 0 0
##               Rtail 0 0 1 0 0 0
##               Rubbr 0 0 1 0 0 0
##               Ships 1 0 0 0 0 0
##               Smoke 1 0 0 0 0 0
##               Soda  0 1 0 0 0 0
##               Softw 0 0 0 0 0 1
##               Steel 1 0 0 0 0 0
##               Telcm 1 0 0 0 0 0
##               Toys  0 0 1 0 0 0
##               Trans 0 0 1 0 0 0
##               Txtls 0 0 1 0 0 0
##               Util  1 0 0 0 0 0
##               Whlsl 0 0 1 0 0 0
```

Portfolio 1 appears to have a lot of primary industry based sectors, such as Agriculture, Mines, Oil and it also dips into sectors related to production and manufacturing. Portfolio 2 is clearly composed of sectors related to (Non)-Discretionary services. Portfolio 3 consists of 27 sectors and predominantly consists of tertiary industry sectors. Portfolio 4 is Gold and Portfolio 5 is Coal, which are our two outliers. Lastly, Portfolio 6 is can be classed as a 'tech' portfolio.

These portfolios are clearly more modern and more defined than before, as shown by the newfound 'tech' portfolio. The next step is to create these individual portfolios with their respective sectors.

5) Creating the Portfolios

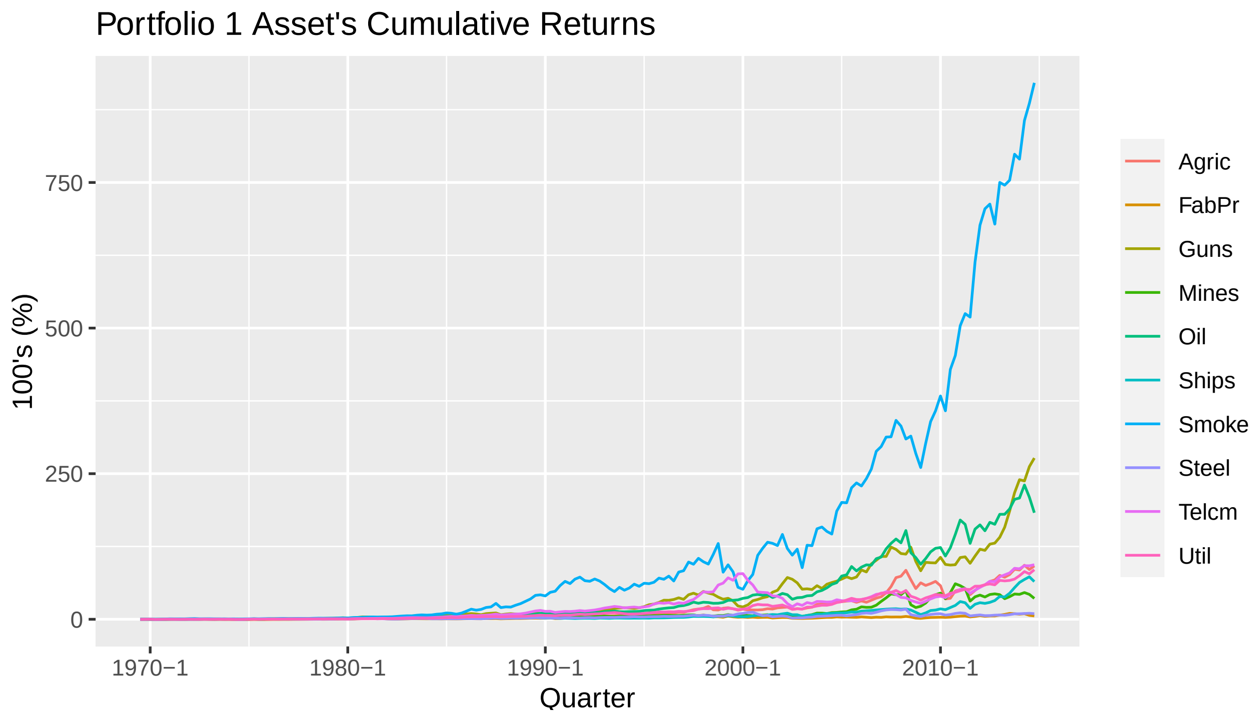
We can use table 'x' from the above code excerpt to do this. First, we need to arrange the sectors alphabetically similar to 'x'. Then we can run the following function to create six individual portfolios with the cumulative returns for each sector stored. The cumulative returns are required to graph each portfolio as given below.

```
new_order = sort(colnames(training2)[2:50])
df <- training2[, new_order]
training2 <- cbind("Quarter"=training2$Quarter, df)
```

```r
Portfolio <- function(z, x, training2) {

  cluster <- which(x[,z] == 1, arr.ind=TRUE)
  cluster <- data.frame(cluster)
  sec <- data.frame("head" = rownames(cluster))
  names <- rbind(data.frame(head = "Quarter"), sec)

  Returns <- data.frame(matrix(nrow = dim(training2)[1], ncol = dim(cluster)[1]+1))
  Returns[1] <- data.frame(training2$Quarter)

  colnames(Returns) <- names$head

  for (i in 1:dim(cluster)[1]) {

  j <- cluster[i,]

  Returns[i+1] <- training2[j+1]

  }

  for (i in 2:dim(Returns)[2]) {

    r <- Returns[i]
    Returns[i] <- cumprod(1+r)-1
}
  return(Returns)
}
```

Cumulative Returns of Portfolios 1's Sectors

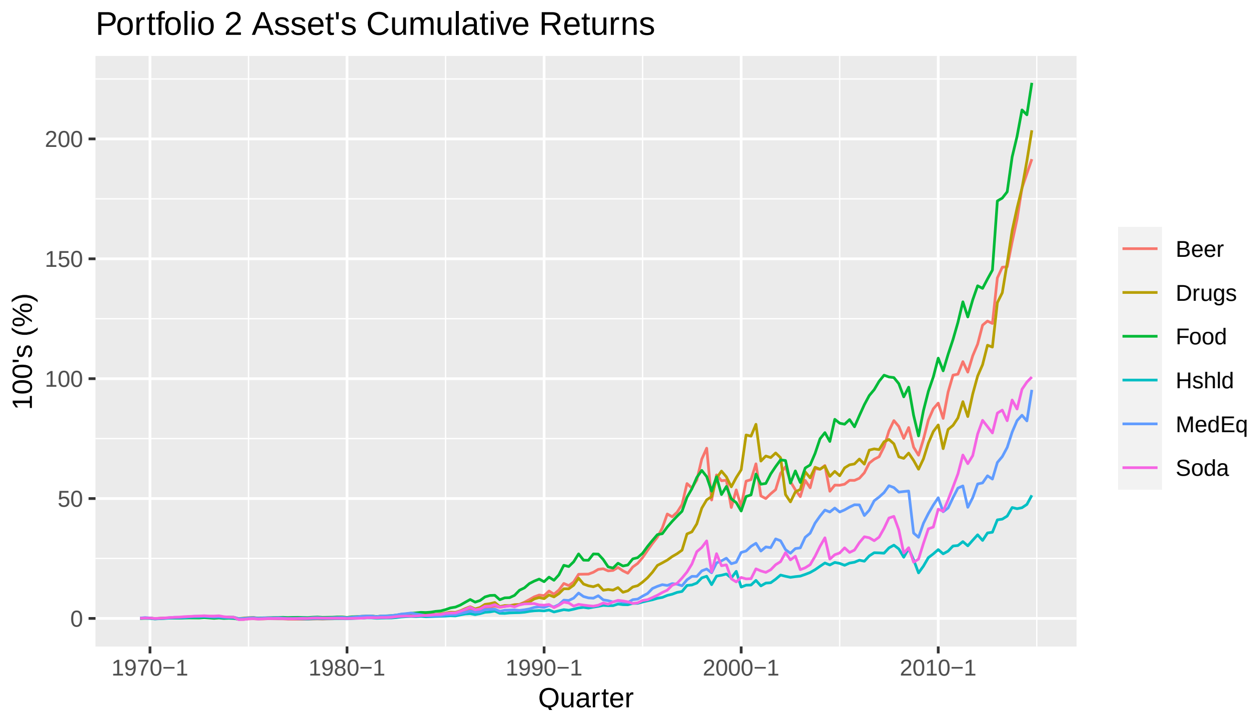
```r
Portfolio_1 <- Portfolio(1, x, training2)

meltdf <- melt(Portfolio_1, id="Quarter")
ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Portfolio 1 Asset's Cumulative Returns") +
  theme(legend.title=element_blank())
```



The “Smoke’ sector, which is solely comprised of tobacco products sales, resulted in a cumulative return x3 greater than the next sector (Guns) in this portfolio.

Cumulative Returns of Portfolios 2’s Sectors

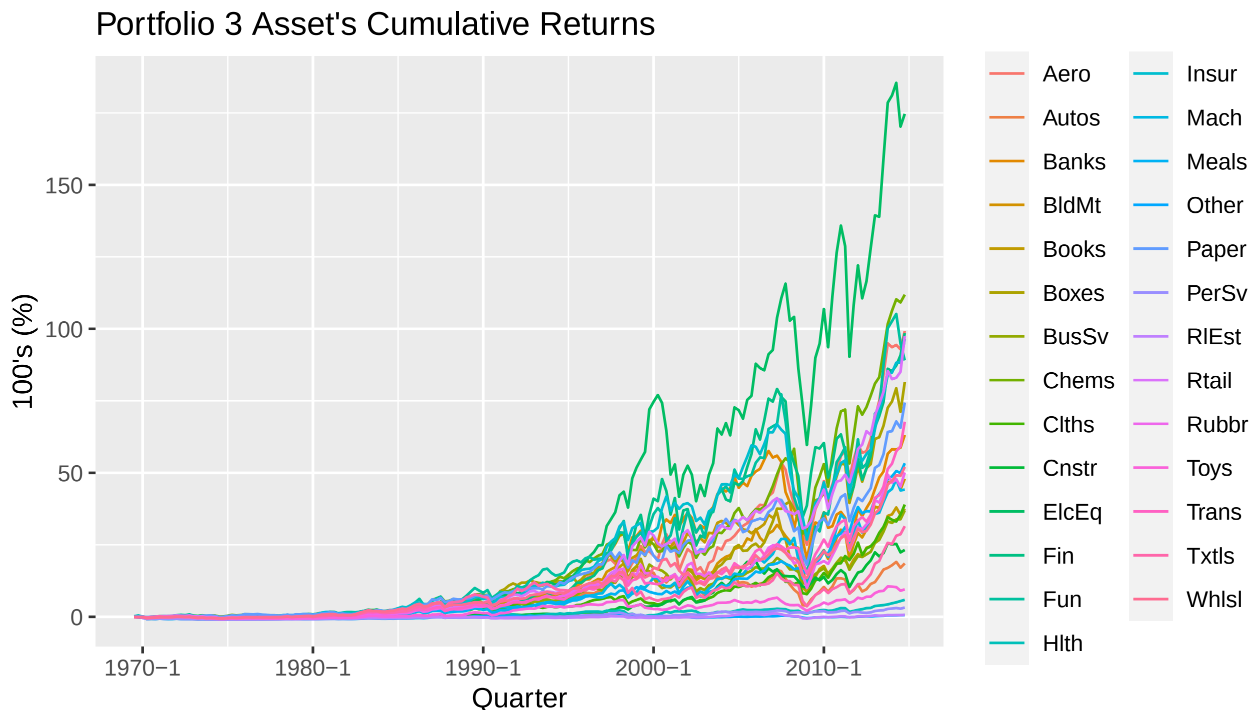
```r
Portfolio_2 <- Portfolio(2, x, training2)

meltdf <- melt(Portfolio_2, id="Quarter")
ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Portfolio 2 Asset's Cumulative Returns") +
  theme(legend.title=element_blank())
```



The ‘Food’, ‘Drugs’ and ‘Beer’ sectors recorded the largest returns in this portfolio, with the ‘Hshld’ sector (comprised of consumer goods) recording the lowest return.

Cumulative Returns of Portfolios 3’s Sectors

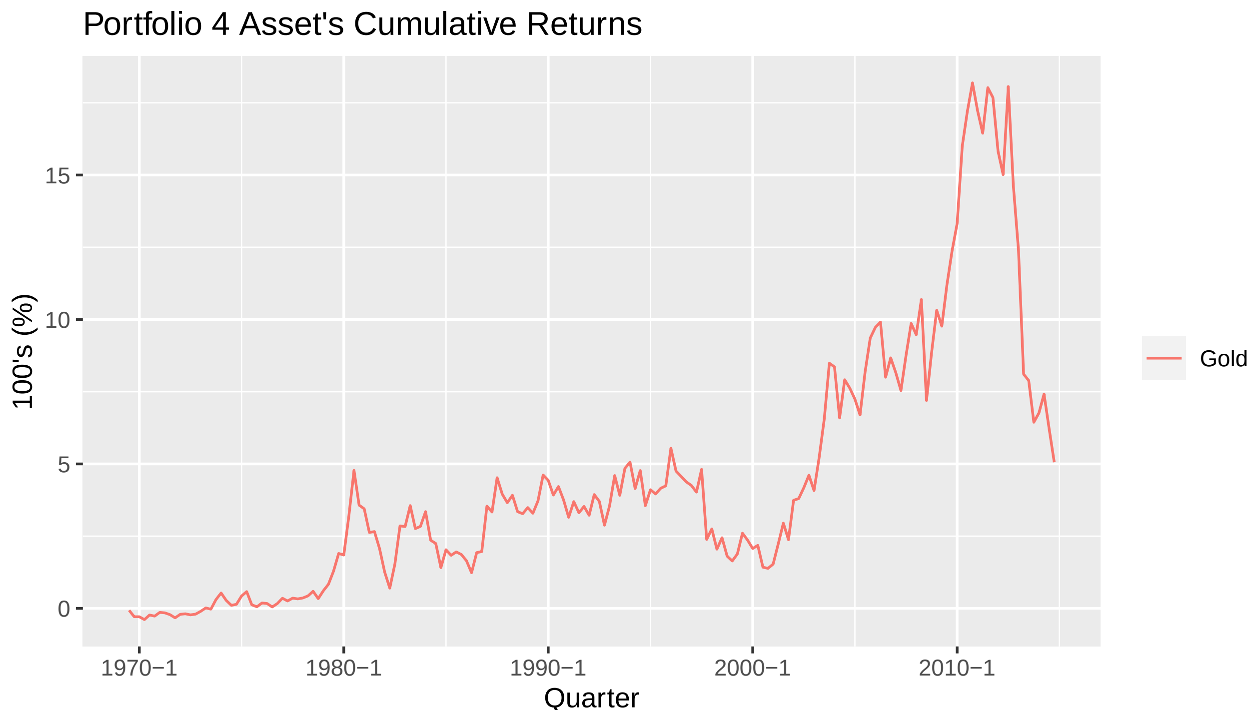
```
Portfolio_3 <- Portfolio(3, x, training2)

meltdf <- melt(Portfolio_3,id="Quarter")
ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Portfolio 3 Asset's Cumulative Returns") +
  theme(legend.title=element_blank())
```



This portfolio contains a lot of similar return levels; however, 'ElcEq' records the highest cumulative return by over 5,000%, while 'Other' and 'RlEst' record remarkably low cumulative returns at 64% and 72%, respectively.

Cumulative Returns of Portfolios 4's Sectors

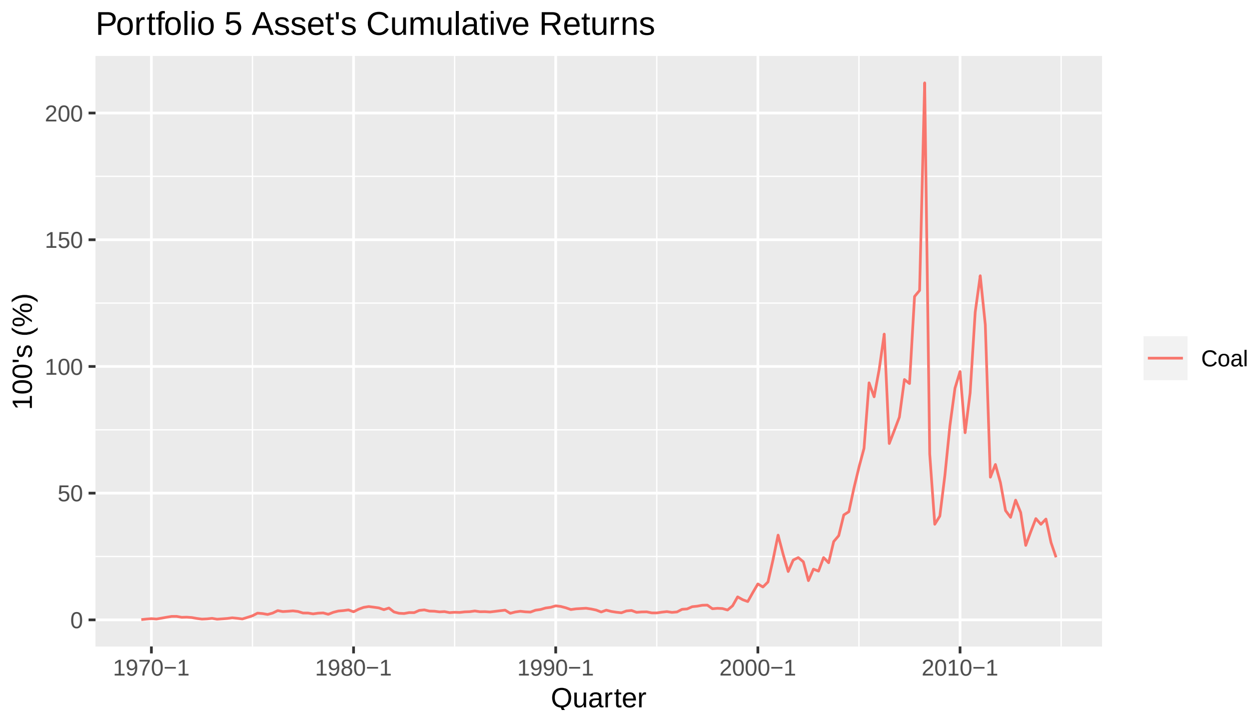
```
Portfolio_4 <- Portfolio(4, x, training2)

meltdf <- melt(Portfolio_4,id="Quarter")
ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Portfolio 4 Asset's Cumulative Returns") +
  theme(legend.title=element_blank())
```



It is not surprising that the 'Gold' sector has been placed in its own portfolio. This sector accounts for gold and silver ore prices, which are often used for diversification reasons or for hedging, due to their market characteristics. Unlike the sectors in the previous three portfolios, this sector has posted negative returns over the past two years.

Cumulative Returns of Portfolios 5's Sectors

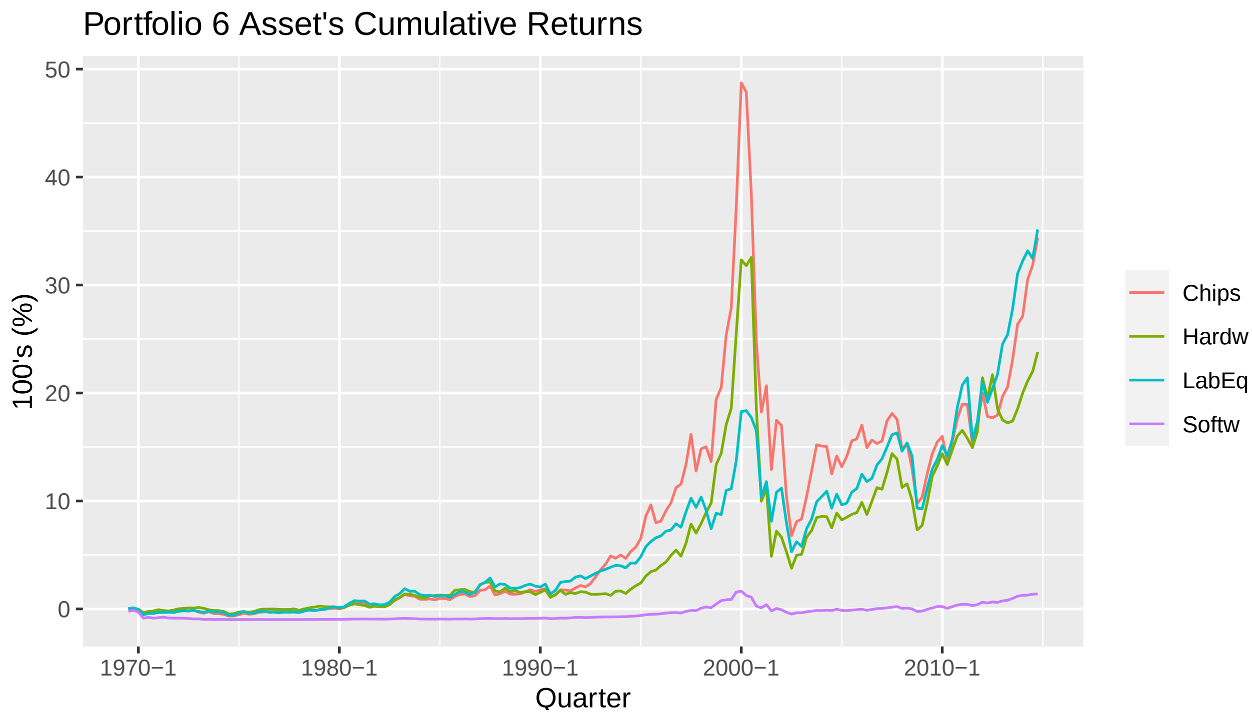
```
Portfolio_5 <- Portfolio(5, x, training2)

meltdf <- melt(Portfolio_5,id="Quarter")
ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Portfolio 5 Asset's Cumulative Returns") +
  theme(legend.title=element_blank())
```



We can see here that the 'Coal' portfolio is incredibly volatile with a peak cumulative return of over 20,000%. The volatility here, and the downward trend shown over the last few years are likely to be two of the main reasons that this sector has its own portfolio.

Cumulative Returns of Portfolios 6's Sectors

```
Portfolio_6 <- Portfolio(6, x, training2)

meltdf <- melt(Portfolio_6,id="Quarter")
ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Portfolio 6 Asset's Cumulative Returns") +
  theme(legend.title=element_blank())
```



This graph is proof of our theory that this portfolio is technology based with the boom in the sector returns around the dotcom era evident. Although the 'Software' sector appears different, it still follows the same trend as the other three sectors, but just at a lower scale of returns.

6) Calculating the Returns of each Portfolio

We require the quarterly returns so that we can compute a mean return of the portfolios for each quarter (i.e., the portfolio returns). It should be noted that by using the mean to calculate portfolio returns, we are assuming that the sectors are equal-weighted, which is correct in this circumstance.

First we need to undo the cumulative returns. We can use the same function as above with the last 'for' loop removed (which changed quarterly returns to cumulative). This modified function is called Portfolio_Basic.

```
Portfolio_Basic <- function(z, x, training2) {

  cluster <- which(x[,z] == 1, arr.ind=TRUE)
  cluster <- data.frame(cluster)
  sec <- data.frame("head" = rownames(cluster))
  names <- rbind(data.frame(head = "Quarter"), sec)

  Returns <- data.frame(matrix(nrow = dim(training2)[1], ncol = dim(cluster)[1]+1))
  Returns[1] <- data.frame(training2$Quarter)

  colnames(Returns) <- names$head

  for (i in 1:dim(cluster)[1]) {

  j <- cluster[i,]

  Returns[i+1] <- training2[j+1]
```

```
  }
  return(Returns)
}

Portfolio_1_Basic <- Portfolio_Basic(1, x, training2)
Portfolio_2_Basic <- Portfolio_Basic(2, x, training2)
Portfolio_3_Basic <- Portfolio_Basic(3, x, training2)
Portfolio_4_Basic <- Portfolio_Basic(4, x, training2)
Portfolio_5_Basic <- Portfolio_Basic(5, x, training2)
Portfolio_6_Basic <- Portfolio_Basic(6, x, training2)
```

With the returns now back to quarterly, we can calculate the mean of all sectors per quarter and add them as a new variable to each portfolio's data frame using the 'Portfolio_M' function.

```
Portfolio_M <- function(portfolio , training2) {

mod <- portfolio[2:dim(portfolio)[2]]
for (i in 1:dim(training2)[1]) {

  portfolio[i, 'Mean'] <- rowMeans(mod[i,])

}
return(portfolio)
}


Portfolio_1_Basic <- Portfolio_M(Portfolio_1_Basic, training2)
Portfolio_2_Basic <- Portfolio_M(Portfolio_2_Basic, training2)
Portfolio_3_Basic <- Portfolio_M(Portfolio_3_Basic, training2)
Portfolio_6_Basic <- Portfolio_M(Portfolio_6_Basic, training2)
```

Next, we can extract all of these mean variables into one data frame. This data frame is named 'Portfolio_Test_Means' and it can be used to graph their returns comparatively. We can then cumulate the returns again for comparison.

```
Portfolio_Means <- data.frame("Quarter"=training2$Quarter,
                              "Portfolio 1"=Portfolio_1_Basic$Mean,
                              "Portfolio 2"=Portfolio_2_Basic$Mean,
                              "Portfolio 3"=Portfolio_3_Basic$Mean,
                              "Portfolio 4"=Portfolio_4_Basic$Gold,
                              "Portfolio 5"=Portfolio_5_Basic$Coal,
                              "Portfolio 6"=Portfolio_6_Basic$Mean)



for (i in 2:7) {
  r <- Portfolio_Means[i]
  Portfolio_Means[i] <- cumprod(1+r)-1
}
```

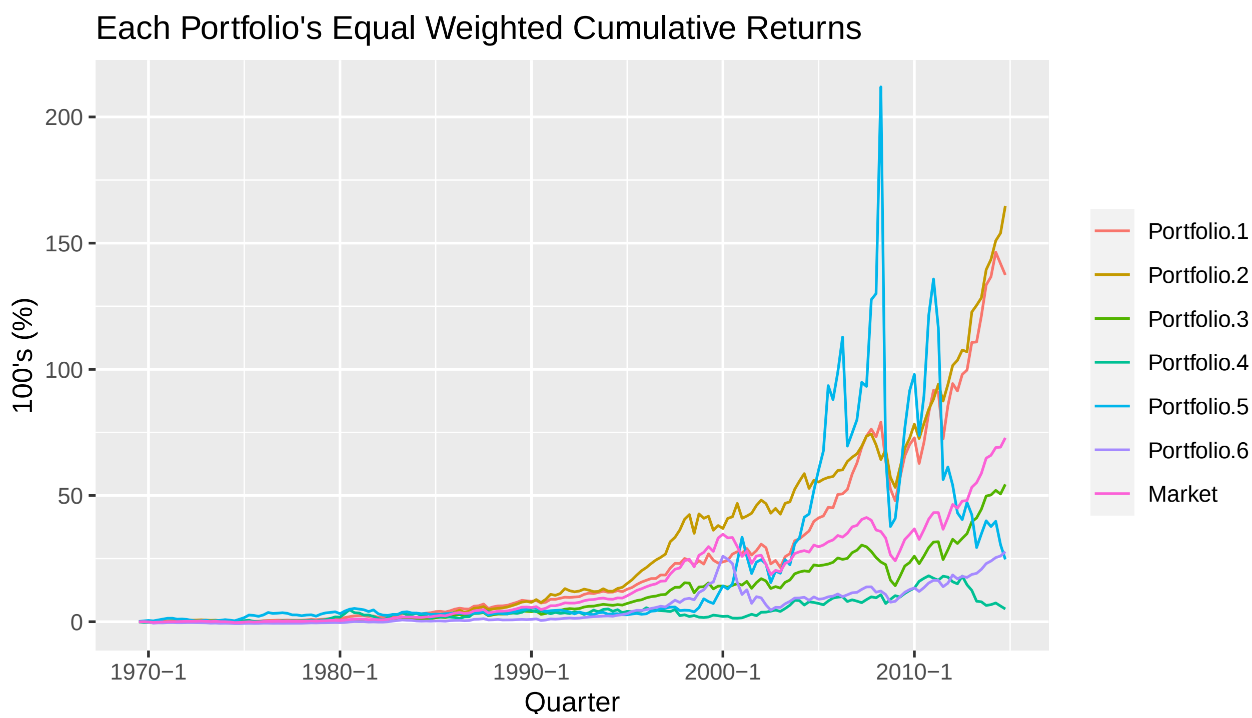
Before we graph these returns, we can also extract the market returns from the Fama/French 3 Factors [Daily] data set. This data is loaded in, converted to quarterly returns, and split into a training and test set.

```
FF=read.csv("F-F_Research_Data_Factors_daily.csv", na.strings=c(""," ","NA"),
            stringsAsFactors=FALSE)
FF$Date <- as.character(FF$Date)
FF$Date <- as.Date(FF$Date, format = "%Y%m%d")
```

```
Date <- FF$Date
FF <- cbind(Date, FF[2:5]/100) # set as decimal returns similar to sector data.

FF$Quarter <- as.yearqtr(FF$Date)
FF_qtrly <- FF[2:6] %>%
  group_by(Quarter) %>%
  summarize(Mkt.RF=sum(Mkt.RF), SMB=sum(SMB), HML=sum(HML), RF=sum(RF), .groups = 'drop')

FF_qtrly <- na.omit(FF_qtrly) # used to delete a line of NAs at tail

trainingFF <- FF_qtrly[training2_start:training2_end,] # same as sector training set
testFF <-  FF_qtrly[355:384,] # same as sector test set
```

Now lets get a market return from this data set. We first need to add back the risk free rate as these are excess returns. Next, we cumulate the market returns so that we can add them to the Portfolio_Means data frame. A graph of all six portfolio returns and the market returns is then generated.

```
training_market <- trainingFF[2] + trainingFF[5]
test_market <- data.frame("Market Returns" = testFF$Mkt.RF + testFF$RF)

Portfolio_Means["Market"] <- cumprod(1+training_market)-1

meltdf <- melt(Portfolio_Means, id="Quarter")
ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Each Portfolio's Equal Weighted Cumulative Returns") +
  theme(legend.title=element_blank())
```



From the graph, we can see that only Portfolio 1 and Portfolio 2 have outperformed the market over this time period. The highest performing portfolio is the consumer goods based portfolio, while the lowest performing portfolio is solely comprised of the ‘Gold’ sector.

7) Fama-French Risk Factor Models on Portfolios

Before we get into the sector rotation strategy, lets first analyse the risk factors of these six portfolios. Using the Fama-French data loaded in earlier, we can run Fama-French 3 Factors models on each portfolio to analyse their risk exposures. These models use basic quarterly returns.

```
model1 <- lm(Portfolio_1_Basic$Mean - trainingFF$RF ~ Mkt.RF + SMB + HML, data=trainingFF)
model2 <- lm(Portfolio_2_Basic$Mean - trainingFF$RF ~ Mkt.RF + SMB + HML, data=trainingFF)
model3 <- lm(Portfolio_3_Basic$Mean - trainingFF$RF ~ Mkt.RF + SMB + HML, data=trainingFF)
model4 <- lm(Portfolio_4_Basic$Gold - trainingFF$RF ~ Mkt.RF + SMB + HML, data=trainingFF)
model5 <- lm(Portfolio_5_Basic$Coal - trainingFF$RF ~ Mkt.RF + SMB + HML, data=trainingFF)
model6 <- lm(Portfolio_6_Basic$Mean - trainingFF$RF ~ Mkt.RF + SMB + HML, data=trainingFF)

export_summs(model1, model2, model3, model4, model5, model6, scale = FALSE)
```

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
(Intercept)	0.00	0.01	-0.00 *	0.01	0.01	0.00
	(0.00)	(0.00)	(0.00)	(0.01)	(0.01)	(0.00)
Mkt.RF	0.86 ***	0.89 ***	1.10 ***	0.15	0.96 ***	1.14 ***
	(0.04)	(0.05)	(0.03)	(0.16)	(0.17)	(0.05)
SMB	0.11	-0.20 **	0.43 ***	0.42	0.14	0.59 ***
	(0.06)	(0.07)	(0.04)	(0.25)	(0.26)	(0.08)
HML	0.24 ***	0.01	0.27 ***	-0.15	0.14	-0.45 ***
	(0.05)	(0.06)	(0.04)	(0.22)	(0.23)	(0.07)
N	182	182	182	182	182	182
R2	0.79	0.71	0.94	0.04	0.21	0.86

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.

Portfolio 1: Significantly exposed to value stocks ($p<0.001$).
Portfolio 2: Significantly exposed to large market cap stocks ($p<0.01$).
Portfolio 3: Significantly exposed to small market cap stocks ($p<0.001$) and value stocks ($p<0.001$).
Portfolio 4: The Gold sector is not exposed to any of these risk factors.
Portfolio 5: The Coal sector is not exposed to any of these risk factors.
Portfolio 6: Significantly exposed to small market cap stocks ($p<0.001$) and growth stocks ($p<0.001$).

These results are somewhat expected for most portfolios. For example, Portfolio 6 is exposed to small market cap stocks and growth stocks, which many stocks in the 'tech' industry fall under due to its recent boom.

8) Sector Rotation Strategy

First, we need to recreate the portfolios for the test data to test our sector rotation strategy on. We can use an adaptation of the previous functions. Note that this function does not generate the cumulative returns as before. Instead, it gives the quarterly returns as recorded.

```
Portfolio_Test <- function(z, x, test) {

  cluster <- which(x[,z] == 1, arr.ind=TRUE)
  cluster <- data.frame(cluster)
  sec <- data.frame("head" = rownames(cluster))
  names <- rbind(data.frame(head = "Quarter"), sec)

  Returns <- data.frame(matrix(nrow = dim(test)[1], ncol = dim(cluster)[1]+1))
  Returns[1] <- data.frame(test$Quarter)

  colnames(Returns) <- names$head

  for (i in 1:dim(cluster)[1]) {

  j <- cluster[i,]

  Returns[i+1] <- test[j+1]

  }
  return(Returns)
}

Portfolio_1_Test <- Portfolio_Test(1, x, test)
Portfolio_2_Test <- Portfolio_Test(2, x, test)
Portfolio_3_Test <- Portfolio_Test(3, x, test)
Portfolio_4_Test <- Portfolio_Test(4, x, test)
Portfolio_5_Test <- Portfolio_Test(5, x, test)
Portfolio_6_Test <- Portfolio_Test(6, x, test)
```

With the portfolios created, we can then get the means of each portfolio (i.e., the portfolio returns) and add them to a new data frame 'Portfolio_Test_Means'. Note that this data frame has three extra columns, which are required to test the strategy in the next step.

```
Portfolio_Test_M <- function(portfolio, test) {

mod <- portfolio[2:dim(portfolio)[2]]
for (i in 1:dim(test)[1]) {

  portfolio[i, 'Mean'] <- rowMeans(mod[i,])

}
return(portfolio)
}

Portfolio_1_Test <- Portfolio_Test_M(Portfolio_1_Test, test)
Portfolio_2_Test <- Portfolio_Test_M(Portfolio_2_Test, test)
Portfolio_3_Test <- Portfolio_Test_M(Portfolio_3_Test, test)
Portfolio_6_Test <- Portfolio_Test_M(Portfolio_6_Test, test)


Portfolio_Test_Means <- data.frame("Quarter"=test$Quarter,
                                   "Portfolio 1"=Portfolio_1_Test$Mean,
                                   "Portfolio 2"=Portfolio_2_Test$Mean,
                                   "Portfolio 3"=Portfolio_3_Test$Mean,
```

```
                                    "Portfolio 4"=Portfolio_4_Test$Gold,
                                    "Portfolio 5"=Portfolio_5_Test$Coal,
                                    "Portfolio 6"=Portfolio_6_Test$Mean,
                                    "No. 1 Portfolio"=NA,
                                    "No. 2 Portfolio"=NA,
                                    "Total Returns"=NA)
```

As previously stated, the test data is from 2015 Q1 to 2022 Q2 (30 periods). The long-only sector rotation strategy is as follows:

- Determine which of the six portfolios records the highest return in Quarter X-1.
- If either Portfolio 4 or Portfolio 5 record the highest return, then also determine the portfolio with the second highest return. This is important for diversification purposes.
- Invest funds in this #1 portfolio for Quarter X, and if required, also in the #2 portfolio. If the funds are to be split, then they shall be split 50/50.

This trading strategy was applied to the test data below and the first six rows are printed to confirm that it executed accurately.

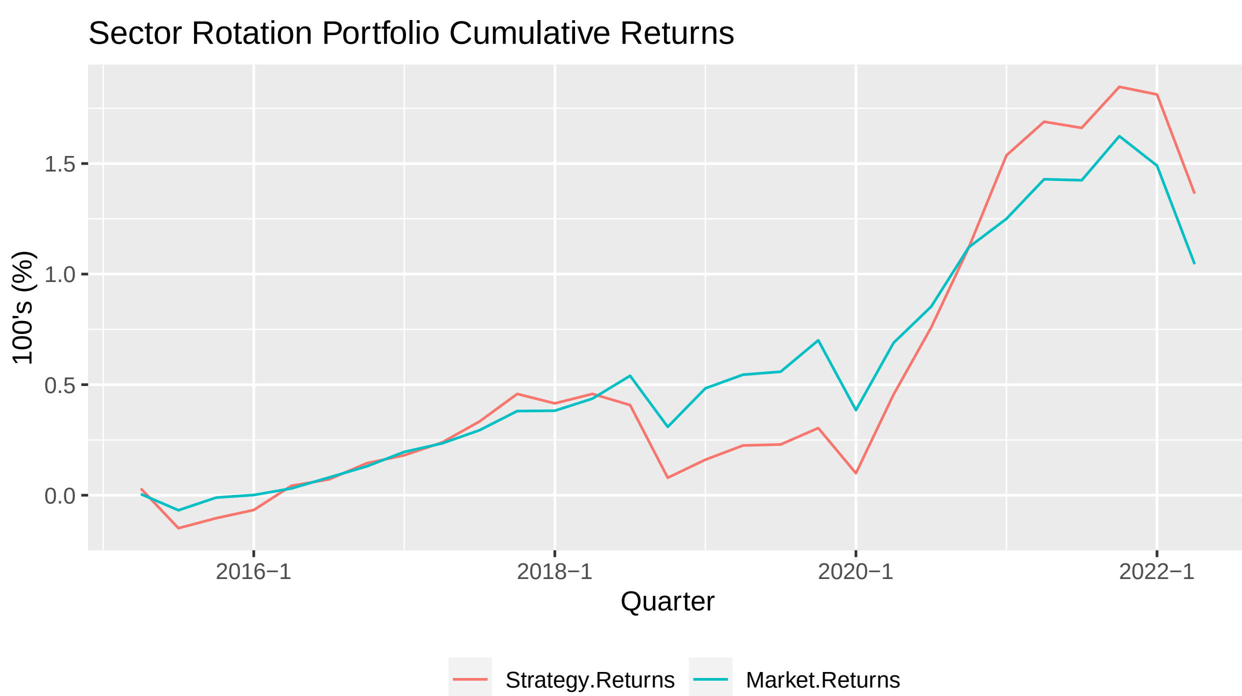
```
for (i in 1:(dim(test)[1]-1)){

  max <- sort(Portfolio_Test_Means[i,2:7], decreasing = TRUE)[1]
  j <- which(Portfolio_Test_Means[i,2:7] == max[,1], arr.ind=TRUE)

  Portfolio_Test_Means[i+1,8] <- Portfolio_Test_Means[i+1,j[2]+1]

  if (j[2] == 4 ||j[2] == 5){

    max2 <- sort(Portfolio_Test_Means[i,2:7], decreasing = TRUE)[2]
    c <- which(Portfolio_Test_Means[i,2:7] == max2[1,], arr.ind=TRUE)

    Portfolio_Test_Means[i+1,9] <- Portfolio_Test_Means[i+1,c[2]+1]

  }

  Portfolio_Test_Means[i+1,10] <- mean(as.numeric(Portfolio_Test_Means[i+1,8:9]),
                                        na.rm=TRUE)

}

kable(head(Portfolio_Test_Means[,c(1,8:10)], 6))
```

Quarter	No..1.Portfolio	No..2.Portfolio	Total.Returns
2015 Q1	NA	NA	NA
2015 Q2	0.030375	NA	0.030375
2015 Q3	-0.174175	NA	-0.174175
2015 Q4	0.053050	NA	0.053050
2016 Q1	0.036200	0.04651	0.041355
2016 Q2	0.117425	NA	0.117425

Naturally, the first row does not record a strategy return as it is only used to observe the best performing portfolio. We can see that row 5 (2016 Q1) has invested in two portfolios, as the best performing portfolio in row 4 (2015 Q4) was Portfolio 5.

With the strategy implemented on the test data, we can now extract it and compare it to the market returns for that period. Once again, we must get the total cumulative return of this strategy and the market.

```r
Strategy_Returns <- data.frame("Quarter" = test$Quarter,
                               "Strategy Returns" = Portfolio_Test_Means$Total.Returns,
                               "Market Returns"= test_market)


Strategy_Returns <- Strategy_Returns[-1,] #2015 Q1 has no values for strategy.


Strategy_Returns_Cum <- Strategy_Returns
for (i in 2:3) {
  r <- Strategy_Returns_Cum[i]
  Strategy_Returns_Cum[i] <- cumprod(1+r)-1
}

meltdf <- melt(Strategy_Returns_Cum,id="Quarter")
p <- ggplot(meltdf, aes(x=Quarter, y=value, colour=variable, group=variable)) + geom_line() +
  labs(y = "100's (%)", title = "Sector Rotation Portfolio Cumulative Returns") +
  theme(legend.title=element_blank())
p + theme(legend.position="bottom")
```



The strategy returns are relatively similar to the market returns. The cumulative strategy returns are below the market initially, and from 2018 Q2 to 2020 Q3, but they outperform the market over the given test period.

9) Fama-French Risk Factor Models on Strategy Returns

Lets analyse the strategy risk exposures using the different Fama-French factors. We need to use the excess returns here.

```r
Strategy_Returns[2] <- Strategy_Returns[2] - testFF$RF[2:30]
Strategy_Returns[3] <- Strategy_Returns[3] - testFF$RF[2:30]

model <- lm(Strategy_Returns$Strategy.Returns ~ Mkt.RF[2:30] + SMB[2:30] + HML[2:30],
            data = testFF)
summary(model)
```

```
##
## Call:
## lm(formula = Strategy_Returns$Strategy.Returns ~ Mkt.RF[2:30] +
##     SMB[2:30] + HML[2:30], data = testFF)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.095723 -0.042522 -0.002236  0.026126  0.117428
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.009133   0.011006   0.830   0.4145
## Mkt.RF[2:30] 1.013107   0.144365   7.018 2.35e-07 ***
## SMB[2:30]    0.456537   0.261444   1.746   0.0931 .
## HML[2:30]    0.102187   0.131928   0.775   0.4459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05428 on 25 degrees of freedom
## Multiple R-squared:  0.8143, Adjusted R-squared:  0.792
## F-statistic: 36.53 on 3 and 25 DF,  p-value: 2.717e-09
```

The strategy portfolio records a beta of 1.01, which is significant ($p<0.001$); however, it is quite indifferent from the market beta of 1 (i.e., statistically significant, but not economically significant). It is also exposed to small market cap stocks at a 10% significance level. The alpha of this portfolio is positive; however, it is not significant at any level. A significantly positive alpha here would indicate that the portfolio is skilled at beating the market after accounting for both size and value risk.

10) Risk/Return Metrics for the Strategy.

```r
sharpe_p <- mean(Strategy_Returns$Strategy.Returns)/sd(Strategy_Returns$Market.Returns)
sharpe_m <- mean(Strategy_Returns$Market.Returns)/sd(Strategy_Returns$Market.Returns)

CAPM <- lm(Strategy_Returns$Strategy.Returns ~ Mkt.RF[2:30], data = testFF)
beta <- CAPM$coefficients[2]
treynor_p <- mean(Strategy_Returns$Strategy.Returns)/as.numeric(CAPM$coefficients[2])
treynor_m <- mean(Strategy_Returns$Market.Returns) # portfolio beta of market is 1

sortino_p <- mean(Strategy_Returns$Strategy.Returns)/
  sd(Strategy_Returns$Strategy.Returns[Strategy_Returns$Strategy.Returns<0])
sortino_m <- mean(Strategy_Returns$Market.Returns)/
  sd(Strategy_Returns$Market.Returns[Strategy_Returns$Market.Returns<0])


paste0("The sharpe ratio for the strategy portfolio is ",round(sharpe_p,3),".")
```

```r
paste0("The sharpe ratio for the market is ",round(sharpe_m,3),".")
paste0("The treynor ratio for the strategy portfolio is ",round(treynor_p,3),".")
paste0("The treynor ratio for the market is ",round(treynor_m,3),".")
paste0("The sortino ratio for the strategy portfolio is ",round(sortino_p,3),".")
paste0("The sortino ratio for the market is ",round(sortino_m,3),".")
```

```
## [1] "The sharpe ratio for the strategy portfolio is 0.39."
## [1] "The sharpe ratio for the market is 0.302."
## [1] "The treynor ratio for the strategy portfolio is 0.03."
## [1] "The treynor ratio for the market is 0.027."
## [1] "The sortino ratio for the strategy portfolio is 0.387."
## [1] "The sortino ratio for the market is 0.333."
```

Sharpe Ratio: This ratio measures excess return per unit of total risk. The Sharpe ratio of the portfolio is greater than that of the market ($0.390 > 0.302$).

Treynor Ratio: This ratio measures excess return per unit of systematic risk. The Treynor ratio of the portfolio is greater than that of the market ($0.030 > 0.027$).

Sortino Ratio: This ratio measures excess return per unit of downside risk (i.e., penalises for downside volatility). The Sortino ratio of the portfolio is greater than that of the market ($0.387 > 0.333$).

In summary, these ratios confirm that our sector rotation strategy outperforms the market on a risk-adjusted basis. Therefore, by investing in this portfolio, we can obtain greater returns than the market for the same level of risk (be it total, systematic or downside risk).