Project 1: HO2 Analysis

Alex Hyman

7/13/2018

My Notes

- Heating Oil No. 2 is used in many of our investments. When the price goes up, the cost of producing products goes up
- When HO2 is volatile, so are our easrnings
- We would like to understand HO2 better so we can better forecast our earnings
- We want to reflect the ups and downs of price movements

Part 1: HO2 data preparation and exploration

Data is provided by the turing.manhattan.edu website and is accessed via the url function. The following R block will read the data, ensuring that the DATE column is not read as a factor. We will also preview the data with the head function and remove any pieces of partial data with the na.omit function. The na.omit function will get rid of any rows that have NA as a value in either column. Finally, the structure of the data frame is viewed to ensure the data is set up as we wish.

The R skills that are required in this block include: * Reading files * Previewing Data * Cleaning Data

```
# Getting the data from online at https://turing.manhattan.edu/~wfoote01/finalytics/data/
# Reading the nyhh02.csv file
# Setting strings as factors = FALSE so our dates remain as string, not factors
HO2 <- read.csv(url("https://turing.manhattan.edu/~wfoote01/finalytics/data/nyhh02.csv"),
                stringsAsFactors = F, header = T)
# Previewing the data
head(HO2)
```

```
##       DATE DHOILNYH
## 1 6/2/1986    0.402
## 2 6/3/1986    0.393
## 3 6/4/1986    0.378
## 4 6/5/1986    0.390
## 5 6/6/1986    0.385
## 6 6/9/1986    0.373
```

```
# Setting the new HO2 data frame to not include missing data
HO2 <- na.omit(HO2)
# looking at the structure of the data frames
str(HO2) # Two columns, DATE is a character, DHOILNYH is a numeric column
```

```
## 'data.frame':    7697 obs. of  2 variables:
##  $ DATE    : chr  "6/2/1986" "6/3/1986" "6/4/1986" "6/5/1986" ...
##  $ DHOILNYH: num  0.402 0.393 0.378 0.39 0.385 0.373 0.365 0.389 0.394 0.398 ...
```

Question 1. What is the nature of HO2 returns?

To evaluate the nature of returns, it is necessary to create visual graphics that tend to be simplier to process then looking at the data by itself. We will first create a line plot of the rate of returns and create a bar plot

of the absolute value of teh rate of returns. The absolute value of the rate of returns will give us a better idea how volatile HO2 prices are, and by default, our volatile our earnings are.

The following block of R code will take the time series pricing data we collected previously and convert it into a data frame with columns including the date, percent returns, magnitude of returns, direction, and the price. The data frame will have three numeric columns (price, returns, size), one factor column (direction), and one date column (Date). The direction column will be a factor with three levels: up, down, and same. These levels indicate whether the price has increased, decreased, or remained the same on a given day, compared to the previous day. This will allow us to separate the levels and conduct an analysis on each level.

The R skills that are required in this block include: * Data manipulation * Vector Creation * Date Formatting * Indexing on conditionals * Manipulating vectors * Data Frame Creation

```
# Calculating difference of logs returns to calculate approximate percent change
return <- as.numeric(diff(log(HO2$DHOILNYH))) * 100
# Getting the absolute value of the percent changes to get magnitude
size <- as.numeric(abs(return))
# creating a character vector that will hold information on whether the price went up, doum, or stayed
direction <- character(length(return))
direction[return > 0] <- "up"
direction[return < 0] <- "down"
direction[return == 0] <- "same"
# Converting the dates in HO2 to actual dates using the as.Date function. Deleting the first index of d
date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")
# Creating a vector of the prices. Deleting the first index so we can have same lengths in vectors.
price <- HO2$DHOILNYH[-1]
# Creating a data frame for HO2 that contains the returns percent change, date, direction, magnitude of
HO2.df <- na.omit(data.frame(
                            Date = date,
                            Price = price,
                            Return = return,
                            Size = size,
                            Direction = direction
                             ))
# Looking at the structure of the data frame. The Dates should be included as dates and the direction s
str(HO2.df)
```

```
## 'data.frame':    7696 obs. of  5 variables:
##  $ Date     : Date, format: "1986-06-03" "1986-06-04" ...
##  $ Price    : num  0.393 0.378 0.39 0.385 0.373 0.365 0.389 0.394 0.398 0.379 ...
##  $ Return   : num  -2.26 -3.89 3.13 -1.29 -3.17 ...
##  $ Size     : num  2.26 3.89 3.13 1.29 3.17 ...
##  $ Direction: Factor w/ 3 levels "down","same",..: 1 1 3 1 1 1 3 3 3 1 ...
```

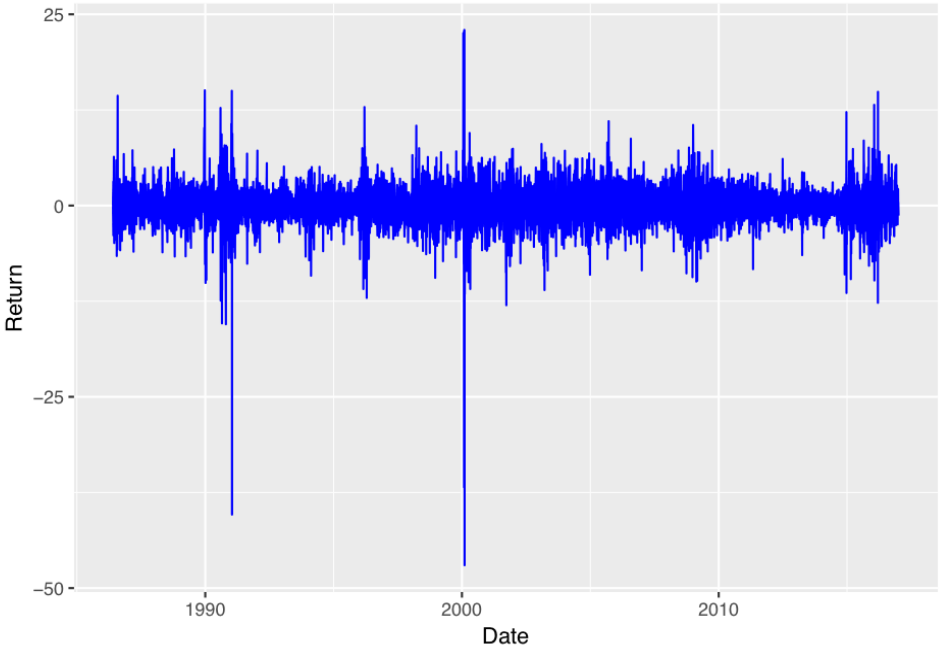
Now that all of the data has been cleaned and stored in a manner that is function for analysis, we will plot the returns of HO2 over time via the ggplot2 library and the ggplot function. The aesthetic function inside the ggplot function will have set x to the date and set y to the return. The geom_line function will be added to the ggplot to create the line plot, and the colour argument inside the geom_line function will be set to blue to display the plot in an applealing manner.

The R skills that are required in this block include: * Plotting with ggplot * Using geom_line function * Displaying plots

1. What is the nature of HO2 returns? Clustered and with some extreme returns, especially on the negative end.

```
# Loading the ggplot2 library for graphics
library(ggplot2)
```

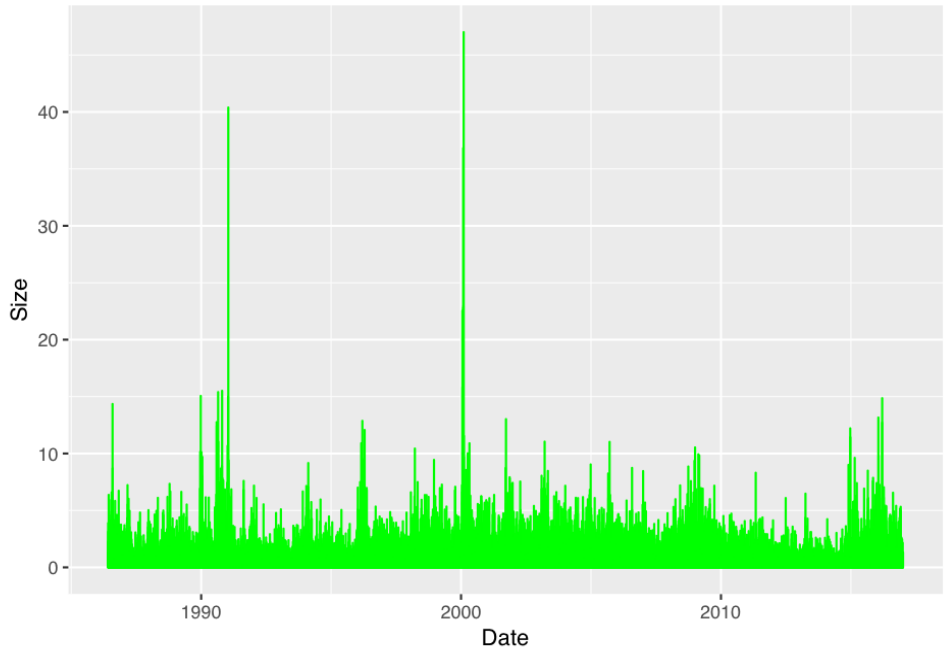
```r
# Setting the variable p to the plot fo the returns over time
# The ggplot function initializes the plot object, HO2.df is the data for the plot
# aes function is setting the asthetic for the plot including x as the Date and y as the returns
# The group = 1 argument sets all the data in the same group to ensure all data are plotted
# The geom_line function is creating the line plot on the ggplot object with the colour set
# to blue to display a blue line
p <- ggplot(HO2.df, aes(x = Date, y = Return, group = 1)) + geom_line(colour = 'blue')
# the p is printing out the ggplot object to actually see the plot
p
```



Next, we will create a bar plot of the absolute value of the rate of returns. This will be done using the ggplot function with the Date as the X-axis and the absolute value of the rate of returns as the y-axis. The plot is displyed as a bar chart, adding the geom_bar tool to the ggplot. Because we are not interested in count data, the stat argument within the geom_bar tool is set to identity to show the actual size of the absolute value of rate of returns over time.

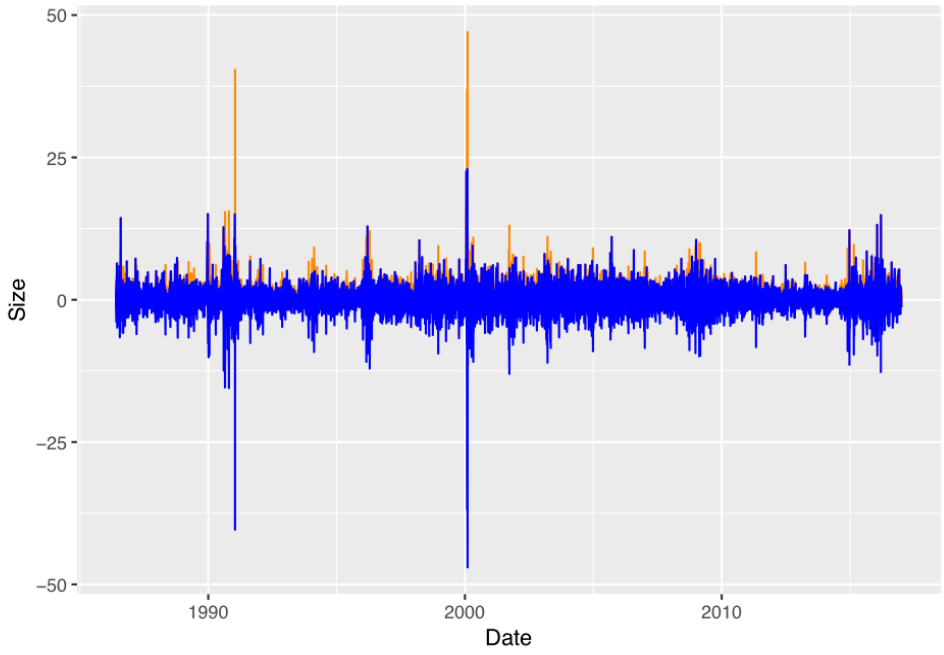
The R skills that are required in this block include:

```r
# Setting the variable p to the plot fo the returns over time
# The ggplot function initializes the plot object, HO2.df is the data for the plot
# aes function is setting the asthetic for the plot including x as the Date and y as the size
# The group = 1 argument sets all the data in the same group to ensure all data are plotted.
# The geom_bar function is used show the volatility in returns as a bar plot over time. The stat
# argument is set to identity so the bar is the same size as the returns, and is not counting the
# number of entries
p <- ggplot(HO2.df, aes(x = Date, y = Size, group = 1)) +
    geom_bar(stat = "identity", colour = "green")
p # Displaying the plot
```



Next, we will provide a plot that contains both the absolute value of the returns in an orange color and the actual returns in blue. Wherever the orange line is visible, the returns were negative for that day.

```
# Plotting the orange size of returns geom_bar plot first (on bottom) and then overlaying the blue geom
# Creating the ggplot object, setting the aesthetic X to the data and the size as the y
p <- ggplot(HO2.df, aes(x = Date, y = Size)) +
    geom_bar(stat = 'identity', colour = 'darkorange') +
    # Drawing the line plot and making it the blue color
    geom_line(data = HO2.df, aes(x = Date, y = Return), colour = 'blue')
p
```



Question 2

Let's dig deeper and compute mean, standard deviation, etc. Load the data_moments() function. Run the function using the HO2.df$return subset of the data and write a knitr::kable() report.

Next an analysis of the returns will be conducted. This analysis will include the: * Mean * Standard Deviation * Median * Skewness * Kurtosis

```
# Creating a function to calculate all the analysis data listed above
data_moments <- function(data) {
    # Need the moments library for skewness and kurtosis
    library(moments)
    #Calculate mean
    mean.r <- mean(data)
    # Calculate standard deviation
    sd.r <- sd(data)
    # Calculate median
    median.r <- median(data)
    # Calculate skewness (shifted lef or right)
    skewness.r <- skewness(data)
    # Calculte the kurtosis (peakedness)
    kurtosis.r <- kurtosis(data)
    # Creating a data frame with all the moments
    result <- data.frame(Mean = mean.r, 'Standard Deviation' = sd.r,
                         Median = median.r, Skewness = skewness.r,
                         Kurtosis = kurtosis.r
                         )
```

```
    # Returning the data frame
    return(result)
}
# Using the function we created on the returns
answer <- data_moments(HO2.df$Return)
# Rounding our data frame to only four decimal points
answer <- round(answer, 4)
# Creating table for display
knitr::kable(answer)
```

Mean	Standard.Deviation	Median	Skewness	
0.0179	2.5236	0	-1.4353	
What do t	he numbers signify?			
The avera	ge returns is approxi	mately ze	ro, but the	re are extreme tails with the kurtosis around 38 and a fairly l

Question 3 - Let's pivot size and return on direction. What is the average and range of returns by direction? How often might we view positive or negative movements in HO2?

To gain a better understanding of how the returns behave, we will make a summary table that provides the average returns, standard deviation of returns, 5th percentile, 95th percentile, and the percentage of instances for each of the three different returns categories ("up", "down", "same"). Logically, the returns that are the same should have an average of zero, standard deviation of zero, and

```
#Set results to "asis" from karl broman from kbroman.org
#Count number of results less than 0 (equal to zero is false)
table(HO2.df$Return < 0)
```

FALSE TRUE 4039 3657

```
#Count number of results greater than 0 (equal to zero is false)
table(HO2.df$Return > 0)
```

FALSE TRUE 3936 3760

```
#Count number of instances in each category
table(HO2.df$Direction)
```

down same up 3657 279 3760

```
#Counts number of returns that ended where they finished
table(HO2.df$Return == 0)
```

FALSE TRUE 7417 279

```
#Import the dplyr library for pivot table analysis
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Starting the pivot table by separtating by direction
pivot.table <- group_by(HO2.df, Direction)
# Making all columns visible
options(dplyr.width = Inf)
# Getting the number of entries so percantage can be calculated
HO2.count <- length(HO2.df$Return)
#Creating the pivot table based on the grouped direction. This data includes average return, standard d
pivot.table <- summarise(pivot.table,
                         return.avg = round(mean(Return), 4),
                         return.sd = round(sd(Return), 4),
                         quantile.5 = round(quantile(Return, 0.05), 4),
                         quantile.95 = round(quantile(Return, 0.95), 4),
                         percent = round(length(Return) / HO2.count, 4) * 100
                         )
#Displaying the pivot table
knitr::kable(pivot.table, digits = 2)
```

Direction	return.avg	return.sd	quantile.5	quantile.95	percent
down	-1.77	1.99	-4.78	-0.19	47.52
same	0.00	0.00	0.00	0.00	3.63
up	1.76	1.75	0.18	4.82	48.86

```
#Importing the xtable library
library(xtable)
#Initializing the caption for another table
HO2.caption <- "Heating Oil No. 2: 1986 - 2016"
#Printing the pivot table in LaTeX
print(xtable(t(pivot.table), digits = 2, caption = HO2.caption, align = rep("r", 4), table.placement =
```

% latex table generated in R 3.5.0 by xtable 1.8-2 package % Thu Jul 26 19:54:21 2018

	1	2	3
Direction	down	same	up
return.avg	-1.7718	0.0000	1.7598
return.sd	1.9862	0.0000	1.7460
quantile.5	-4.7761	0.0000	0.1817
quantile.95	-0.1894	0.0000	4.8203
percent	47.52	3.63	48.86

Table 3: Heating Oil No. 2: 1986 - 2016

```
#Printing the details of all the returns
print(xtable(answer), digits = 2)
```

% latex table generated in R 3.5.0 by xtable 1.8-2 package % Thu Jul 26 19:54:21 2018

	Mean	Standard.Deviation	Median	Skewness	Kurtosis
1	0.02	2.52	0.00	-1.44	38.26

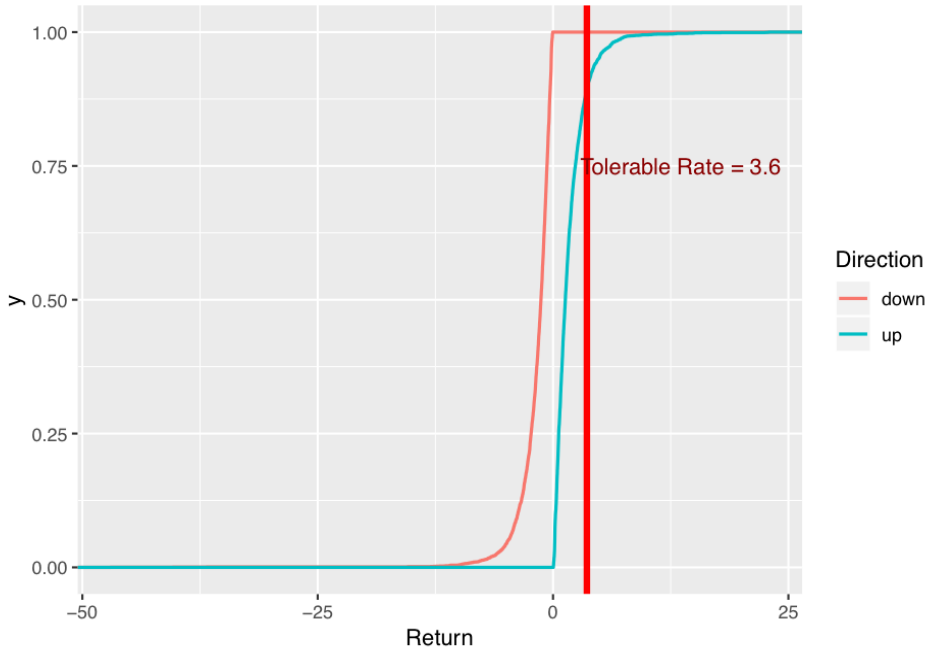
Part 2: HO2 analysis

Question 1. How can we show the differences in the shape of ups and downs in HO2, especially given our tolerance for risk? We can use the HO2.df data frame with ggplot2 and the cumulative relative frequency function stat_ecdf to begin to understand this data.

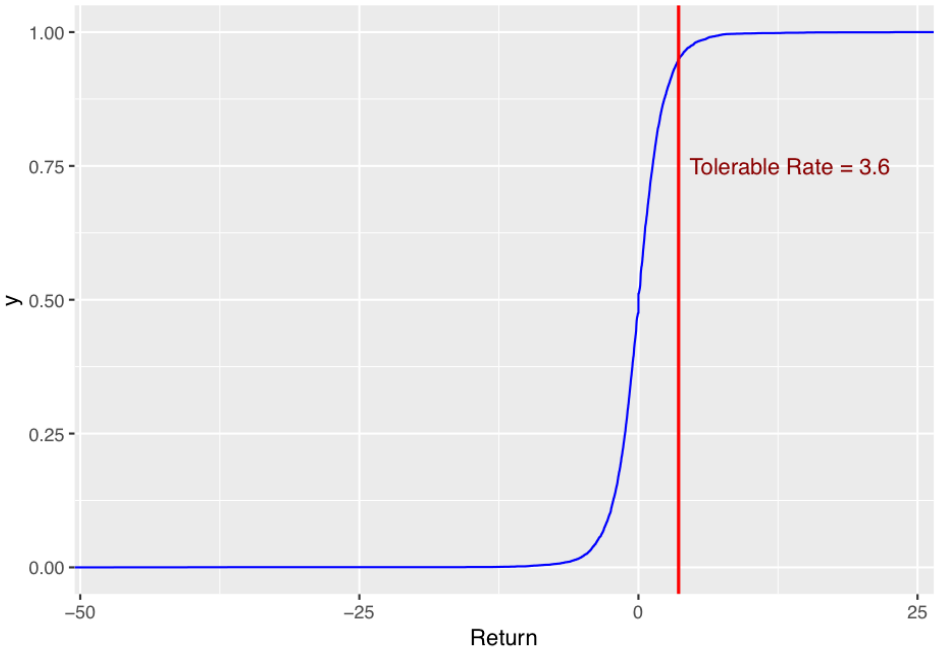
Further questions as needed

Plotting the shapes of the positive and negative returns. **Why are our returns on a daily rate? Our tolerance is based on this

```
#Setting the tolerance to 5%
HO2.tol.pct <- 0.95
#Figuring out what the 95th percentile of returns would be
HO2.tol <- quantile(HO2.df$Return, HO2.tol.pct)
#Creating the label with the past function that says where our tolerable limit is
HO2.tol.label <- paste("Tolerable Rate = ", round(HO2.tol, 2), sep = "")
#Plotting the cumulative density function for negative, positive, and same returns. Same returns should
ggplot(HO2.df[HO2.df$Direction == "up" | HO2.df$Direction == "down",], aes(Return, colour = Direction))
    # starting the plot with returns as our data and a different density function for each direction
    stat_ecdf(size = 0.75) + # plotting the density functions in blue
    geom_vline(xintercept = HO2.tol, # drawing a vertical redline at the tolerance limit
        colour = "red", size = 1.5) +
    annotate("text", x = HO2.tol + 10, # adding the label to the plot
        y = 0.75, label = HO2.tol.label,
        colour = "darkred")
```




```
# Creating a plot that shows overall cumulative distribution
ggplot(HO2.df, aes(Return)) + stat_ecdf(colour = "blue") + geom_vline(xintercept = HO2.tol, colour = "r
    size = 0.75) + annotate("text", x = HO2.tol + 10, y = 0.75, colour = "darkred", label = HO2.tol.lab
```



Question 2. How can we regularly, and reliably, analyze HO2 price movements? For this requirement, let's write a function similar to data_moments. Name this new function HO2_movement().

```
#First we are going to create a function that will automatically create the pivot table, xtable and dat
HO2_movement <- function(file = "data/nyhh02.csv",
                         caption = "Heating Oil No. 2: 1986-2016"){
    # The file is extracted from the manhattan site. It is also pasted to create a single string
    file <- paste("https://turing.manhattan.edu/~wfoote01/finalytics/", file, sep = "")
    # Reading the csv file and keeping the strings as string
    HO2 <- read.csv(url(file), header = TRUE, stringsAsFactors = FALSE)
    # Deleting the missing data
    HO2 <- na.omit(HO2)
    # using the log difference as percent change of returns
    return <- as.numeric(diff(log(HO2$DHOILNYH))) * 100
    # Calculating the absolute value of the returns to measure volitility
    size <- abs(return)
    # Creating an empty character vector the length of return
    direction <- character(length(return))
    # Where percent change is greater than 0, put up in the character vector
    direction[return > 0] <- "up"
    # Where percent change is less than 0, put down in the character vector
    direction[return < 0] <- "down"
    # Where percent change is 0, put same in the character vector
    direction[return == 0] <- "same"
```

```
    # Converting the dates that were imported as string into the type date and deleting teh first index
    date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")
    # Converting the price to a number and deleting the first index because needs to be the same length
    price <- as.numeric(HO2$DHOILNYH[-1])
    # Creating the data frame that has the data, price, return, size (absolute value of return), and  d
    HO2.df <- data.frame(Date = date, Price = price, Return = return, Size = size, Direction = directio
    # importing the library to create the pivot tables
    require(dplyr)
    # Creating a grouped table that is grouped by the direction of returns
    pivot.table <- group_by(HO2.df, Direction)
    # Making all columns visible
    options(dplyr.width = Inf)
    # Seeing how many entries are in the table so a percentage can be gathered for each category
    HO2.count <- length(HO2.df$Return)
    # Creating the pivot table that returns the average return, return standard deviation, 5% quantile,
    pivot.table <- summarise(pivot.table, return.avg = mean(Return), return.sd = sd(Return),
                             quantile.5 = quantile(Return, 0.05), quantile.95 = quantile(Return, 0.95),
                             percent = length(Return) / HO2.count * 100)
    require(xtable)
    # Creating a nice LaTeX table of the transposed pivot table using the caption argument of the funct
    pivot.xtable <- xtable(t(pivot.table), # transposing the pivot table
                           digits = 2, caption = caption, # Limiting to 2 digits, and wring the given c
                           align = rep("r", 4), table.placement = "V") # alligning the columns to the r
    # Creating a list to contain all the woutputs of the function
    output.list <- list(table = pivot.table,
                        xtable = pivot.xtable, df = HO2.df)
    # Returning the output list
    return(output.list)
    }
```

Testing function

```
# We are now going to test to see if the function works, and knit the table to look nice
knitr::kable(HO2_movement(file = "data/nyhh02.csv")$table, # knit table and extract the pivot table "ta
    digits = 2) # Set teh table to display only two digits
```

Direction	return.avg	return.sd	quantile.5	quantile.95	percent
down	-1.77	1.99	-4.78	-0.19	47.52
same	0.00	0.00	0.00	0.00	3.63
up	1.76	1.75	0.18	4.82	48.86

Question 3.

Suppose we wanted to simulate future movements in HO2 returns. What distribution might we use to run those scenarios? Here, let's use the MASS package's fitdistr() function to find the optimal fit of the HO2 data to a parametric distribution. We will use the gamma distribution to simulate future heating oil #2 price scenarios.

```
# Importing the statistical package MASS
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
# Using our function to convert the csv file into usable data and extracting the data frame from the li
HO2.data <- HO2_movement(caption = "HO2 NYH")$df
# Looking at the structure of the data frame
# Date is as typr date, Price, Return, and Size are type numeric, and the Direction is a factor, "up",
str(HO2.data)
```

```
## 'data.frame':    7696 obs. of  5 variables:
##  $ Date     : Date, format: "1986-06-03" "1986-06-04" ...
##  $ Price    : num  0.393 0.378 0.39 0.385 0.373 0.365 0.389 0.394 0.398 0.379 ...
##  $ Return   : num  -2.26 -3.89 3.13 -1.29 -3.17 ...
##  $ Size     : num  2.26 3.89 3.13 1.29 3.17 ...
##  $ Direction: Factor w/ 3 levels "down","same",..: 1 1 3 1 1 1 3 3 3 1 ...
```

```
# Now we will model the distribution of returns when the returns are positive
# Using the fitdistr function in the MASS package, we will retrieve the estimated shape and scale for f
fit.gamma.up <- fitdistr(HO2.data[HO2.data$Direction == "up", "Return"], "gamma") # The rows with the d
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```
# Displaying the shape and rate (inverse scale) of the estimated fitted gamma distribution
fit.gamma.up
```

```
##      shape         rate
##   1.30753665   0.74299635
##  (0.02716171) (0.01872184)
```

```
# Not going to model the returns that are the same because all have a value of 0
# Model the negative returns with the t-distribution
# Using fitdistr from MASS, filtering instances that only have the "down" direction
fit.t.down <- fitdistr(HO2.data[HO2.data$Direction == "down", "Return"], "t", hessian = T)
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
# Looking at the mean, standard deviation, and degrees of freedom in the t-distribution of negative ret
fit.t.down
```

```
##         m             s            df
##   -1.30565487    0.91307703    2.50894659
##  ( 0.02170850) ( 0.02061868) ( 0.12442996)
```

```
# Fitting a gamma distribution to the negative returns
# Must make the negative returns positive by multiplying by -1 or use the size column of direction == '
fit.gamma.down <- fitdistr(-HO2.data[HO2.data$Direction == "down", "Return"], "gamma", hessian = TRUE)
# Looking at the shape and the rate (inverse scale) of the distribution for negative returns
fit.gamma.down
```

```
##      shape         rate
##   1.31056202   0.73969342
##  (0.02761041) (0.01889467)
```

```
# I want to fit a t-distribution for all returns
t.dist.all <- fitdistr(HO2.data$Return, "t", hessian = T)
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
## Warning in log(s): NaNs produced
```

```
t.dist.all
```

```
##        m             s            df
##   0.02423709   1.66640769   3.72264703
##  (0.02265778) (0.02407289) (0.16718933)
```

Big questions:

Why using gamma distribution? Gamma distribution when separating between positive and negative. t-distribution when evaluating the potential nature of the entire index/