Project 2 - Group1

HO2 Analysis

Part 1

In this set we will build a data set using filters and `if` and `diff` statements. We will then answer some questions using plots and a pivot table report. We will then write a function to house our approach in case we would like to run the same analysis on other data sets.

Problem

Supply chain managers at our company continue to note we have a significant exposure to heating oil prices (Heating Oil No. 2, or HO2), specifically New York Harbor. The exposure hits the variable cost of producing several products. When HO2 is volatile, so is earnings. Our company has missed earnings forecasts for five straight quarters. To get a handle on Brent we download this data set and review some basic aspects of the prices.

```
HO2 <- read.csv("data/nyhh02.csv", header = T,
    stringsAsFactors = F)
# stringsAsFactors sets dates as
# character type
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
```

```
HO2 <- na.omit(HO2)  ## to clean up any missing data
str(HO2)  # review the structure of the data so far
```

```
## '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 ...
```

Questions

1. What is the nature of HO2 returns? We want to reflect the ups and downs of price movements, something of prime interest to management. First, we calculate percentage changes as log returns. Our interest is in the ups and downs. To look at that we use `if` and `else` statements to define a new column called `direction`. We will build a data frame to house this analysis.

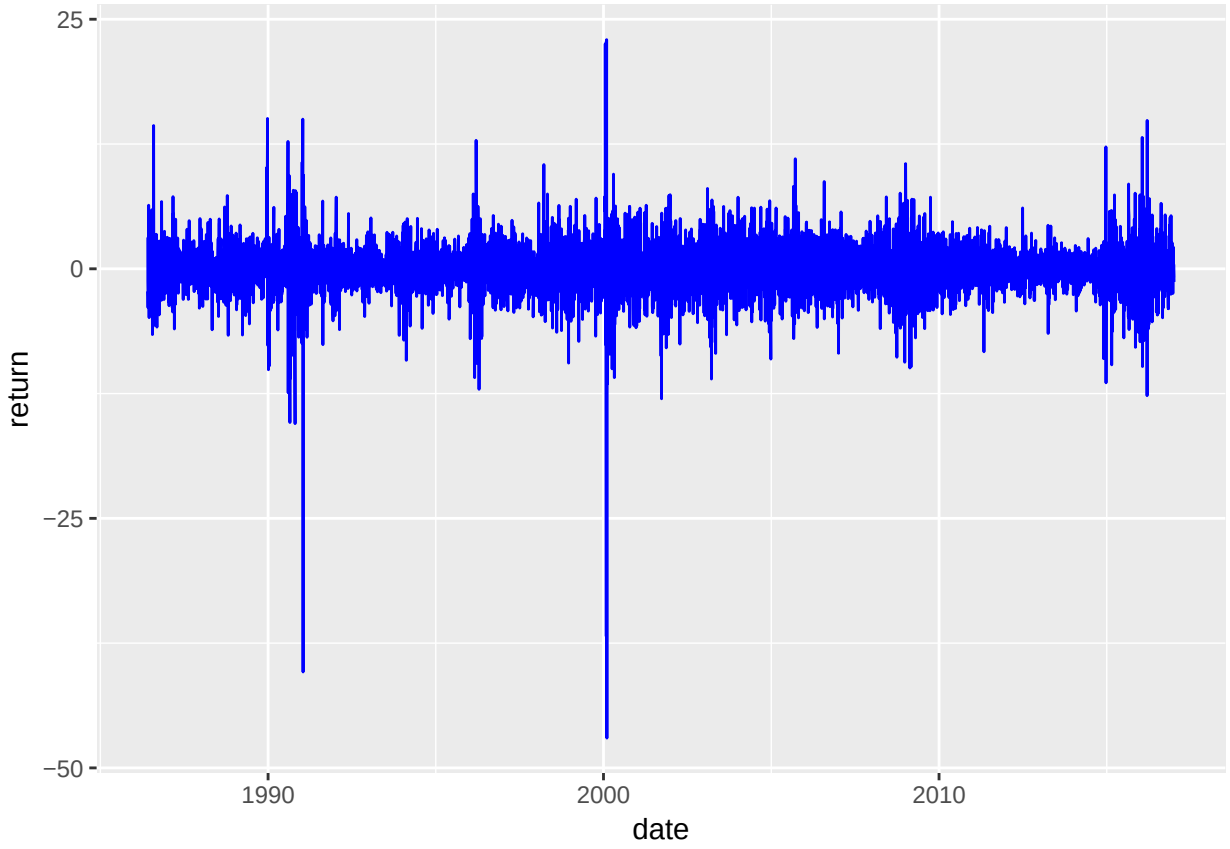
```
# Construct expanded data frame
return <- as.numeric(diff(log(HO2$DHOILNYH))) *
    100
size <- as.numeric(abs(return))  # size is indicator of volatility
direction <- ifelse(return > 0, "up",
    ifelse(return < 0, "down", "same"))  # another indicator of volatility
date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")  # length of DATE is length of return +1: omit 1st observatio
```

```
price <- as.numeric(HO2$DHOILNYH[-1])  # length of DHOILNYH is length of return +1: omit first observat
HO2.df <- na.omit(data.frame(date = date,
    price = price, return = return, size = size,
    direction = direction))  # clean up data frame by omitting NAs
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 ...
```

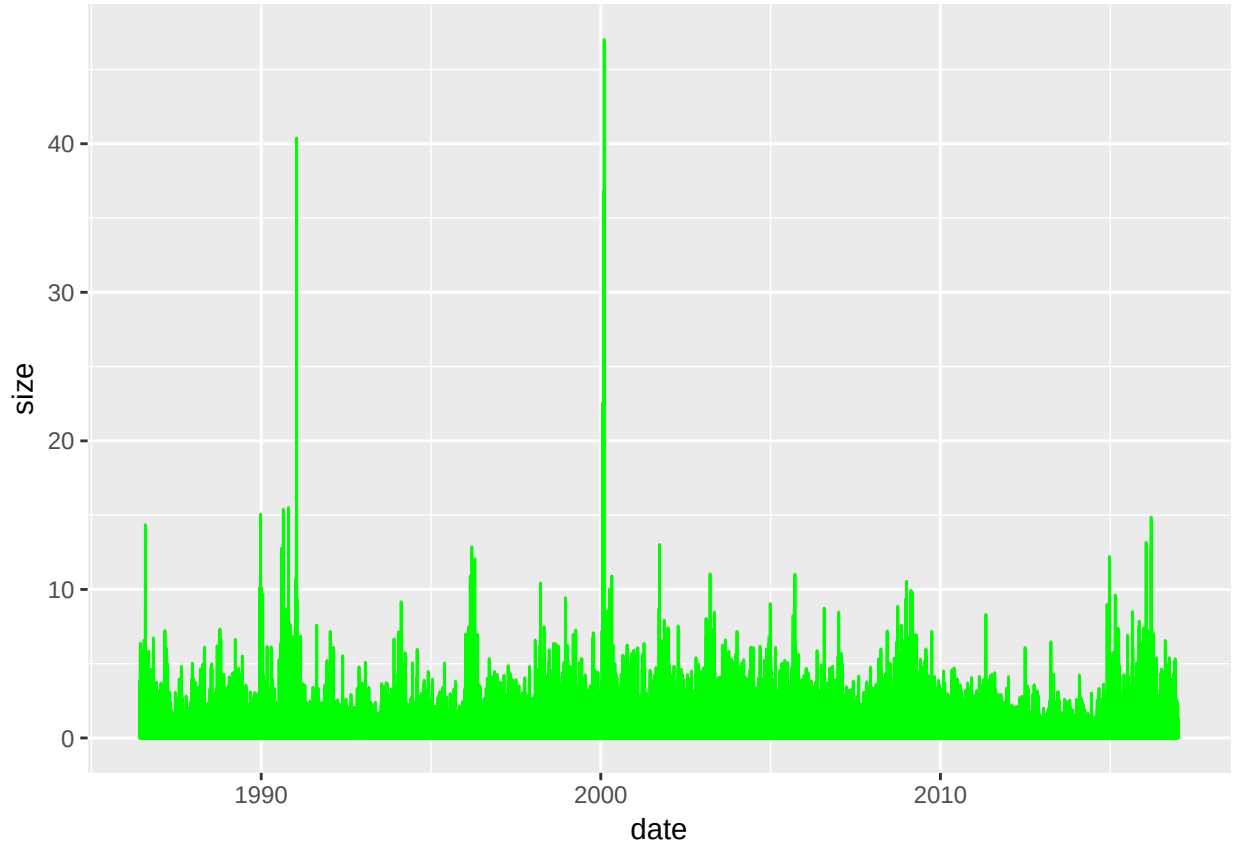
We can plot with the `ggplot2` package. In the `ggplot` statements we use `aes`, "aesthetics", to pick `x` (horizontal) and `y` (vertical) axes. Use `group =1` to ensure that all data is plotted. The added (`+`) `geom_line` is the geometrical method that builds the line plot.

```
require(ggplot2)
ggplot(HO2.df, aes(x = date, y = return,
    group = 1)) + geom_line(colour = "blue")
```



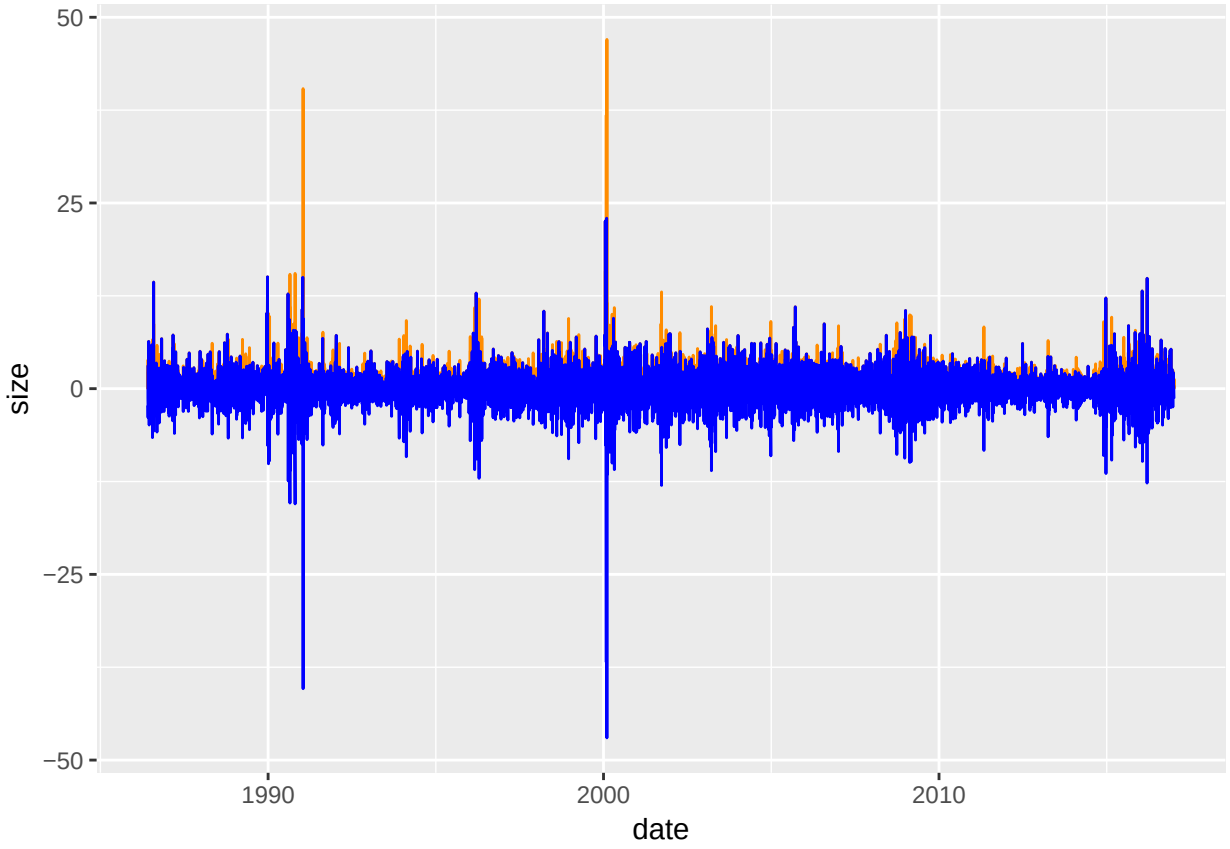
Let's try a bar graph of the absolute value of price rates. We use `geom_bar` to build this picture.

```
# require(ggplot2)
ggplot(HO2.df, aes(x = date, y = size,
    group = 1)) + geom_bar(stat = "identity",
    colour = "green")
```



Now let's build an overlay of `return` on `size`.

```
ggplot(HO2.df, aes(date, size)) + geom_bar(stat = "identity",
    colour = "darkorange") + geom_line(data = HO2.df,
    aes(date, return), colour = "blue")
```



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.

```
# Load the data_moments() function
# data_moments function INPUTS: r
# vector OUTPUTS: list of scalars
# (mean, sd, median, skewness,
# kurtosis)
data_moments <- function(data) {
    require(moments)
    mean.r <- mean(data)
    sd.r <- sd(data)
    median.r <- median(data)
    skewness.r <- skewness(data)
    kurtosis.r <- kurtosis(data)
    result <- data.frame(mean = mean.r,
        std_dev = sd.r, median = median.r,
        skewness = skewness.r, kurtosis = kurtosis.r)
    return(result)
}
# Run data_moments()
```

```
answer <- data_moments(HO2.df$return)
# Build pretty table
answer <- round(answer, 4)
knitr::kable(answer)
```

mean	std_dev	median	skewness	kurtosis
0.0179	2.5236	0	-1.4353	38.2595

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?

```
# Counting
table(HO2.df$return < 0)  # one way
```

```
##
## FALSE  TRUE
##  4039  3657
```

```
table(HO2.df$return > 0)
```

```
##
## FALSE  TRUE
##  3936  3760
```

```
table(HO2.df$direction)  # this counts 0 returns as negative
```

```
##
## down same   up
## 3657  279 3760
```

```
table(HO2.df$return == 0)
```

```
##
## FALSE  TRUE
##  7417   279
```

```
# Pivoting
require(dplyr)
## 1: filter to those houses with
## fairly high prices pivot.table <-
## filter(HO2.df, size >
## 0.5*max(size)) 2: set up data frame
## for by-group processing
pivot.table <- group_by(HO2.df, direction)
## 3: calculate the summary metrics
options(dplyr.width = Inf)  ## to display all columns
HO2.count <- length(HO2.df$return)
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) *
        100, 2))
# Build visual
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

```
# Here is how we can produce a LaTeX
# formatted and rendered table
require(xtable)
options(xtable.comment = FALSE)
HO2.caption <- "Heating Oil No. 2: 1986-2016"
print(xtable(t(pivot.table), digits = 2,
    caption = HO2.caption, align = rep("r",
        4), table.placement = "V"))
```

	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

Heating Oil No. 2: 1986-2016

```
print(xtable(answer), digits = 2)
```

	mean	std_dev	median	skewness	kurtosis
1	0.02	2.52	0.00	-1.44	38.26

Part 2

We will use the data from Part 1 to investigate the distribution of returns we generated. This will entail fitting the data to some parametric distributions as well as

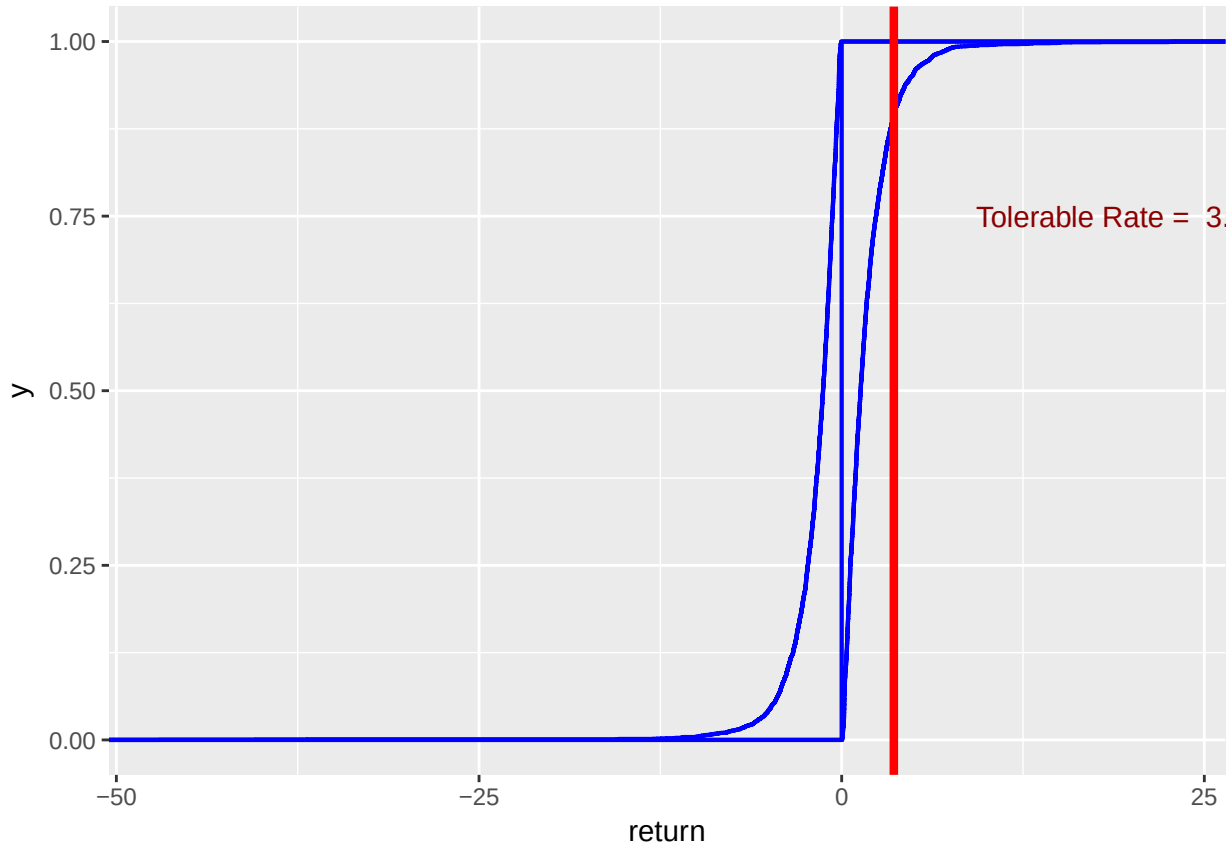
Problem

We want to further characterize the distribution of up and down movements visually. Also we would like to repeat the analysis periodically for inclusion in management reports.

Questions

1. How can we show the differences in the shape of ups and downs in HO2, especially given our tolerance for risk? Let's use the `HO2.df` data frame with `ggplot2` and the cumulative relative frequency function `stat_ecdf`.

```
HO2.tol.pct <- 0.95
HO2.tol <- quantile(HO2.df$return, HO2.tol.pct)
HO2.tol.label <- paste("Tolerable Rate = ",
    round(HO2.tol, 2))
ggplot(HO2.df, aes(return, fill = direction)) +
    stat_ecdf(colour = "blue", size = 0.75) +
    geom_vline(xintercept = HO2.tol,
        colour = "red", size = 1.5) +
    annotate("text", x = HO2.tol + 15,
        y = 0.75, label = HO2.tol.label,
        colour = "darkred")
```



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()`.

```
## HO2_movement(file, caption) input:
## HO2 csv file from /data directory
## output: result for input to kable
## in $table and xtable in $xtable;
## data frame for plotting and further
## analysis in $df.  Example: HO2.data
## <- HO2_movement(file =
## 'data/nyhh02.csv', caption = 'HO2
## NYH')
HO2_movement <- function(file = "data/nyhh02.csv",
    caption = "Heating Oil No. 2: 1986-2016") {
    # Read file and deposit into variable
    HO2 <- read.csv(file, header = T,
        stringsAsFactors = F)
    # stringsAsFactors sets dates as
    # character type
    HO2 <- na.omit(HO2)  ## to clean up any missing data
    # Construct expanded data frame
    return <- as.numeric(diff(log(HO2$DHOILNYH))) *
        100
    size <- as.numeric(abs(return))  # size is indicator of volatility
    direction <- ifelse(return > 0, "up",
        ifelse(return < 0, "down", "same"))  # another indicator of volatility
    date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")  # length of DATE is length of return +1: omit 1st observ
```

```
    price <- as.numeric(HO2$DHOILNYH[-1])  # length of DHOILNYH is length of return +1: omit first obse
    HO2.df <- na.omit(data.frame(date = date,
        price = price, return = return,
        size = size, direction = direction))  # clean up data frame by omitting NAs
    require(dplyr)
    ## 1: filter if necessary pivot.table
    ## <- filter(HO2.df, size >
    ## 0.5*max(size)) 2: set up data frame
    ## for by-group processing
    pivot.table <- group_by(HO2.df, direction)
    ## 3: calculate the summary metrics
    options(dplyr.width = Inf)  ## to display all columns
    HO2.count <- length(HO2.df$return)
    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)
    # Construct transpose of pivot table
    # with xtable()
    require(xtable)
    pivot.xtable <- xtable(t(pivot.table),
        digits = 2, caption = HO2.caption,
        align = rep("r", 4), table.placement = "V")
    HO2.caption <- "Heating Oil No. 2: 1986-2016"
    output.list <- list(table = pivot.table,
        xtable = pivot.xtable, df = HO2.df)
    return(output.list)
}
```

Test `HO2_movement()` with data and display results in a table with 2 decimal places.

```
knitr::kable(HO2_movement(file = "data/nyhh02.csv")$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

Morale: more work today (build the function) means less work tomorrow (write yet another report).

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.

```
require(MASS)
HO2.data <- HO2_movement(file = "data/nyhh02.csv",
    caption = "HO2 NYH")$df
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 ...
```

```
fit.gamma.up <- fitdistr(HO2.data[HO2.data$direction ==
    "up", "return"], "gamma", hessian = TRUE)
fit.gamma.up
```

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

```
fit.gamma.up$estimate/fit.gamma.up$sd
```

```
##    shape     rate
## 48.13896 39.68608
```

```
# fit.t.same <-
# fitdistr(HO2.data[HO2.data$direction
# == 'same', 'return'], 'gamma',
# hessian = TRUE) # a problem here is
# all observations = 0
fit.t.up <- fitdistr(HO2.data[HO2.data$direction ==
    "up", "return"], "t", hessian = TRUE)
fit.t.up
```

```
##        m             s            df
##   1.33270760   0.95179926   2.71456370
##  (0.02213093) (0.02087303) (0.13999289)
```

```
fit.t.down <- fitdistr(HO2.data[HO2.data$direction ==
    "down", "return"], "t", hessian = TRUE)
fit.t.down
```

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

```
fit.gamma.down <- fitdistr(-HO2.data[HO2.data$direction ==
    "down", "return"], "gamma", hessian = TRUE)  # gamma distribution defined for data >= 0
fit.gamma.down
```

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

```
# using std.dist
require(fGarch)
Fit.std.down <- stdFit(HO2.data[HO2.data$direction ==
    "down", "return"])
Fit.std.down
```

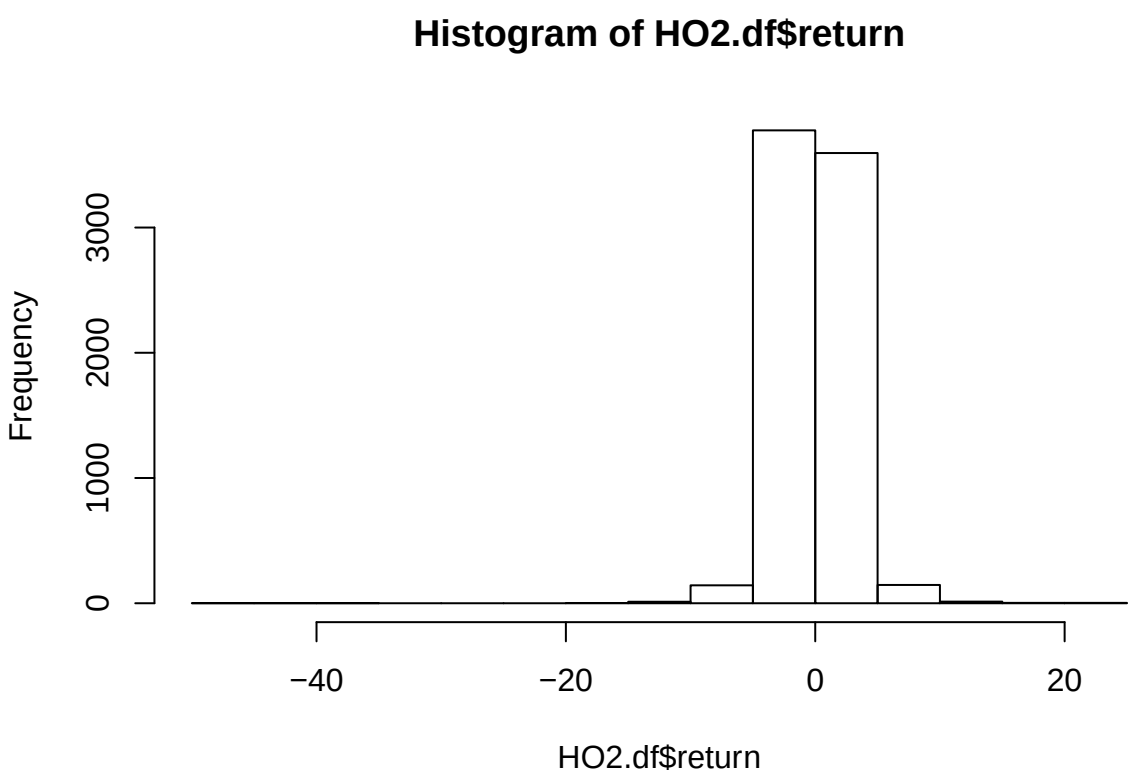
```
## $par
##      mean        sd        nu
## -1.305654  2.027300  2.508942
##
## $objective
## [1] 6418.87
```

```
##
## $convergence
## [1] 0
##
## $iterations
## [1] 28
##
## $evaluations
## function gradient
##       35      101
##
## $message
## [1] "relative convergence (4)"
```

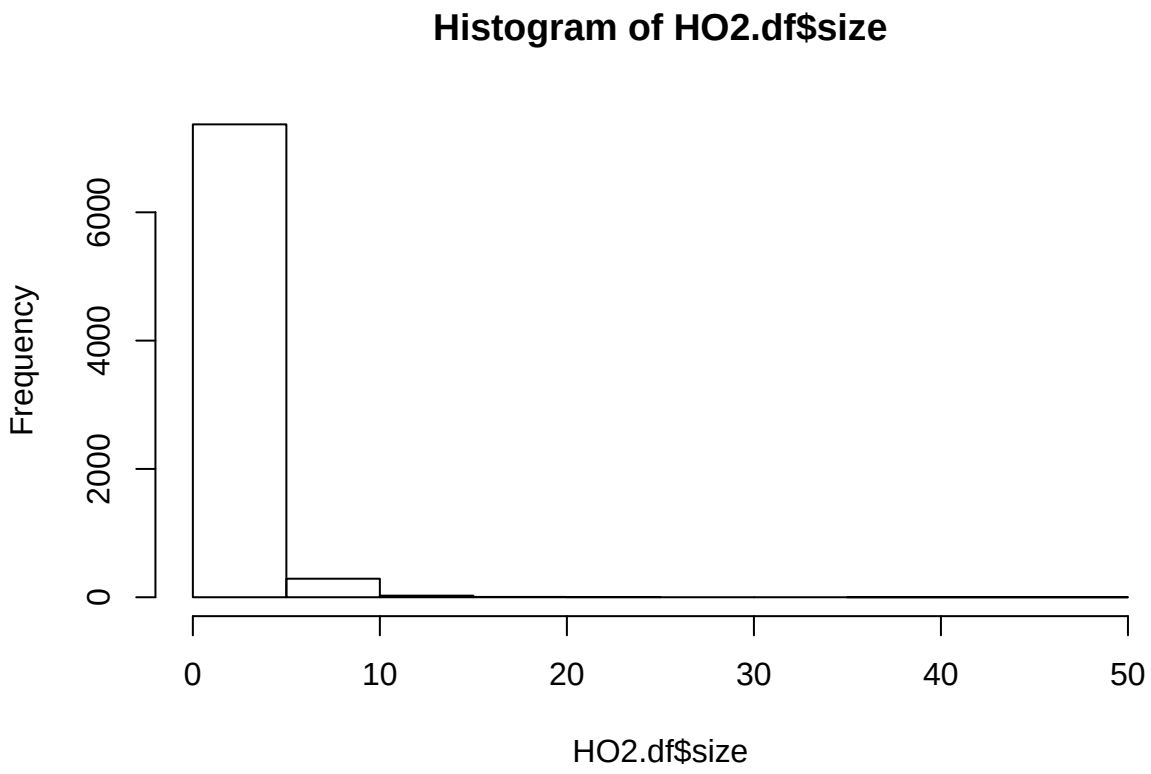
```
Fit.std.up <- stdFit(HO2.data[HO2.data$direction ==
    "up", "return"])
Fit.std.up
```

```
## $par
##     mean       sd       nu
## 1.332708 1.855135 2.714560
##
## $objective
## [1] 6629.047
##
## $convergence
## [1] 0
##
## $iterations
## [1] 16
##
## $evaluations
## function gradient
##       27       61
##
## $message
## [1] "relative convergence (4)"
```

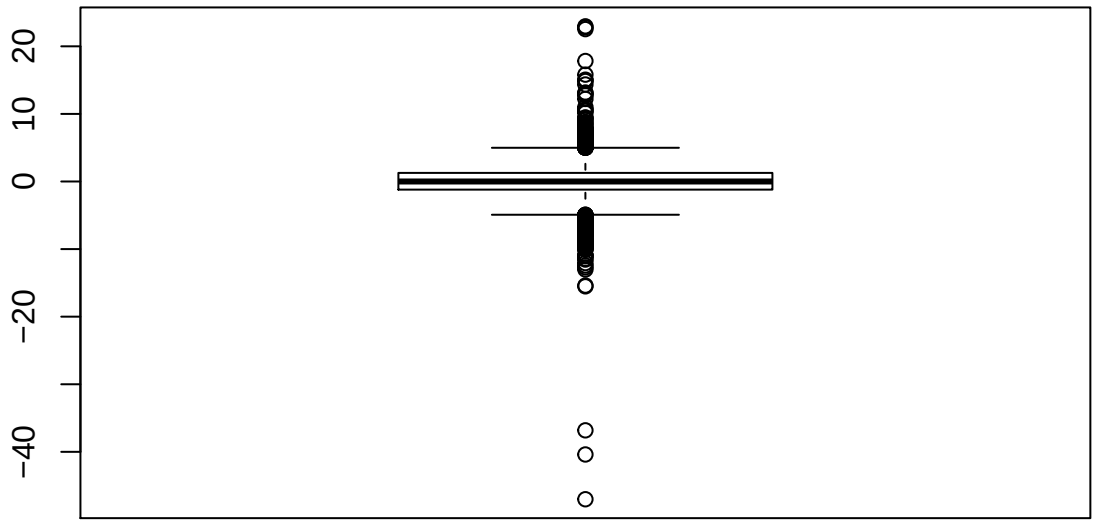
```
# Messing around w/ plots

# Histograms
hist(HO2.df$return)
```



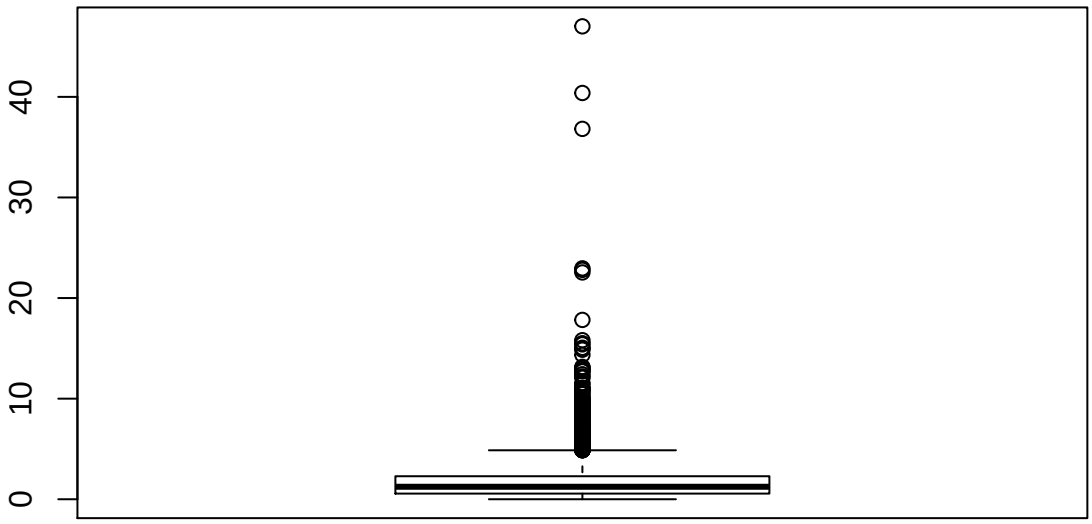

```
hist(HO2.df$size)
```



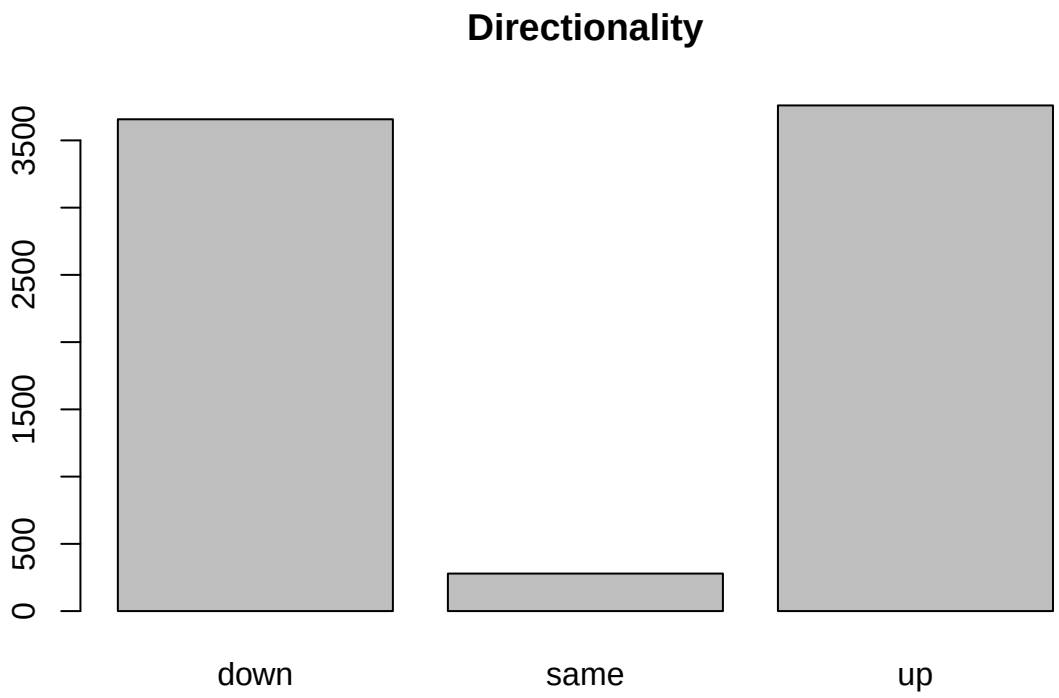

```
# Boxplots
boxplot(HO2.df$return)
```



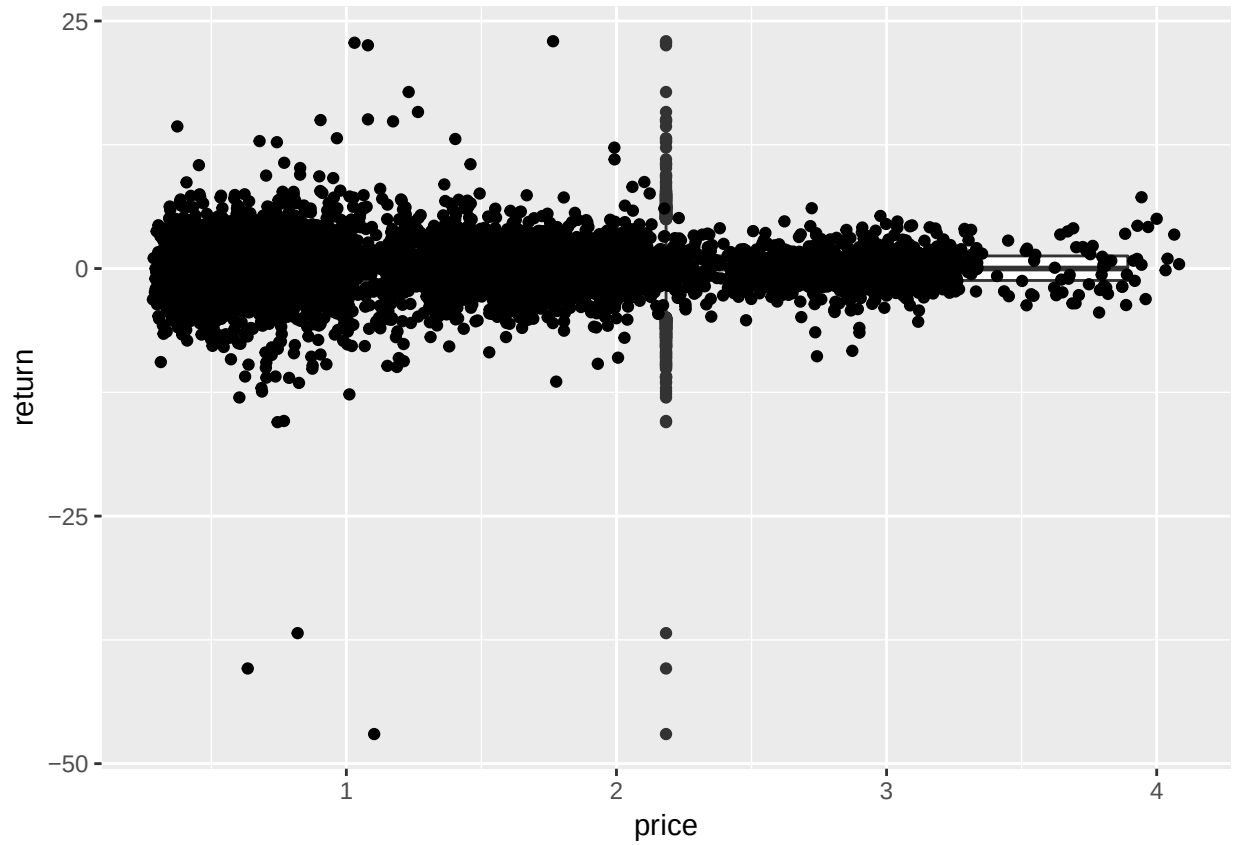

```
boxplot(HO2.df$size)
```



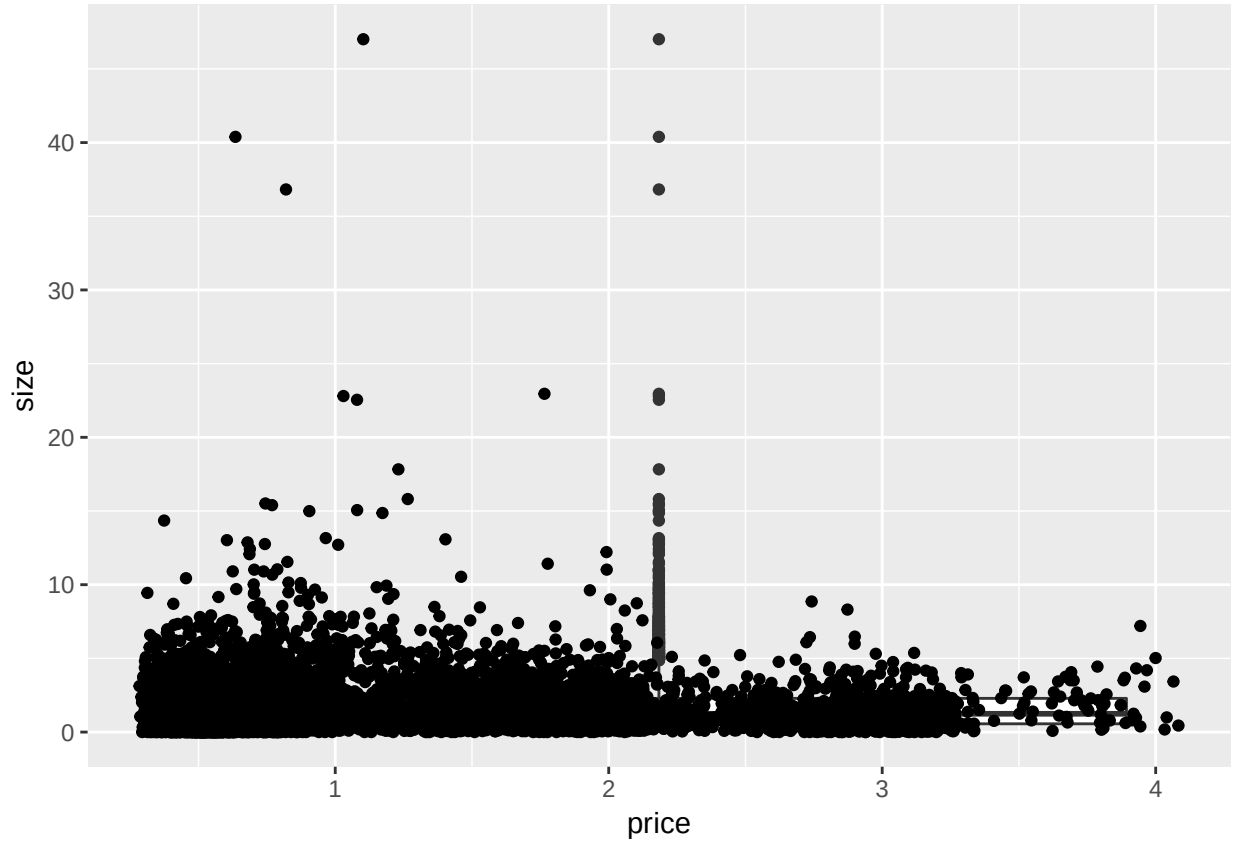

```
# Barplot of directionality counts
counts <- table(HO2.df$direction)
barplot(counts, main = "Directionality")
```




```
qplot(price, return, data = HO2.df, geom = c("boxplot",
    "jitter"))
```




```
qplot(price, size, data = HO2.df, geom = c("boxplot",
    "jitter"))
```

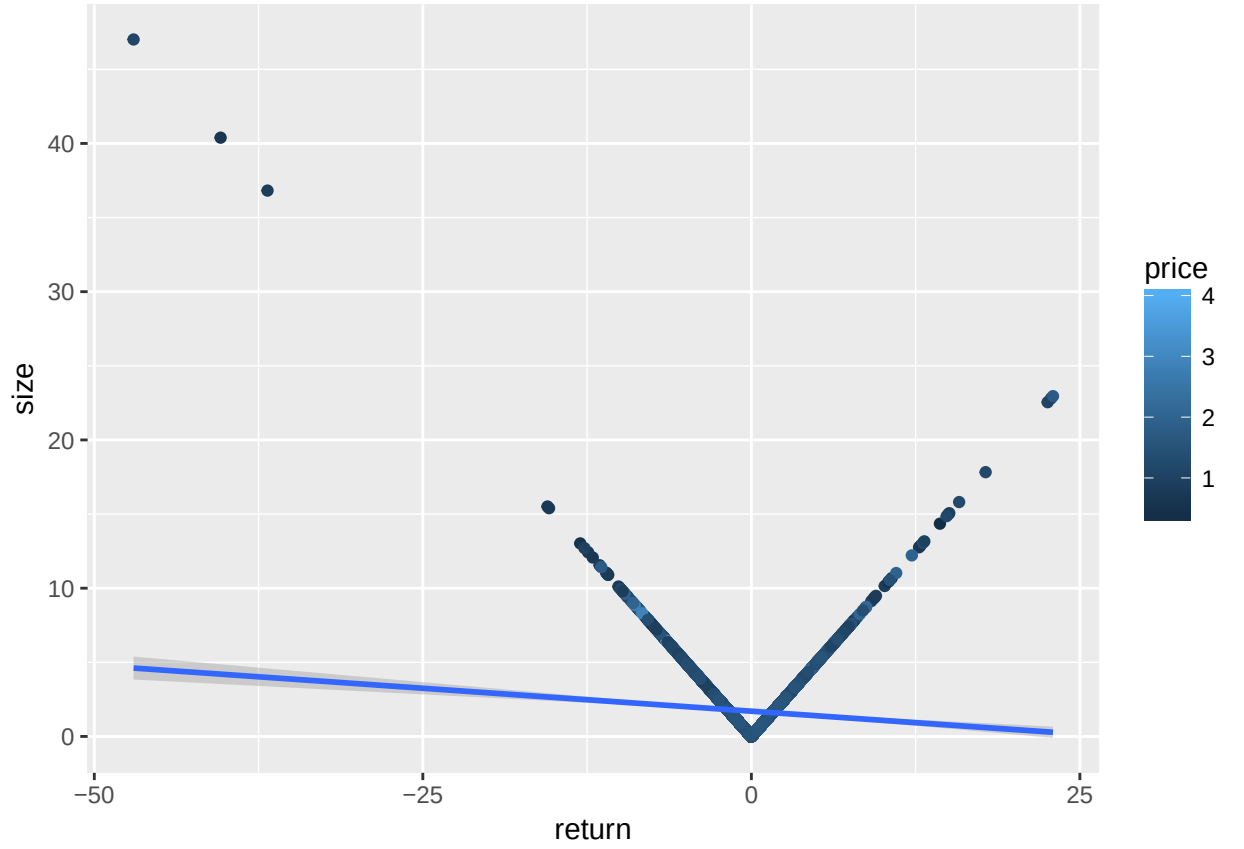



```
# Extracting months out from Date
HO2.df$month <- months(HO2.df$date)

# Sample of why not to use OLS for a
# problem of this nature
model <- lm(formula = price ~ month,
    data = HO2.df)
summary(model)
```

```
##
## Call:
## lm(formula = price ~ month, data = HO2.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9445 -0.7055 -0.4692  0.5735  2.8463
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.266292   0.036257  34.925   <2e-16 ***
## monthAugust    -0.010184   0.050025  -0.204    0.839
## monthDecember  -0.057781   0.050641  -1.141    0.254
## monthFebruary  -0.037785   0.052223  -0.724    0.469
## monthJanuary   -0.060073   0.051338  -1.170    0.242
## monthJuly      -0.029567   0.050566  -0.585    0.559
## monthJune      -0.033968   0.050417  -0.674    0.500
```

```
## monthMarch     -0.034005   0.050528  -0.673    0.501
## monthMay       -0.006782   0.050991  -0.133    0.894
## monthNovember  -0.017162   0.051234  -0.335    0.738
## monthOctober    0.001921   0.050025   0.038    0.969
## monthSeptember -0.005218   0.050951  -0.102    0.918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9021 on 7684 degrees of freedom
## Multiple R-squared:  0.0005106,  Adjusted R-squared:  -0.0009202
## F-statistic: 0.3568 on 11 and 7684 DF,  p-value: 0.972
```

```
qplot(return, size, data = HO2.df, geom = c("point",
    "smooth"), method = "lm", formula = y ~
    x, color = price)
```



Conclusion

Skills used

: There were many skills that were learned/utilized in this project, starting with the use of RMD, and Miktex. The first step in this document is reading the data in from a csv, which is a relatively easy function of R to understand and equip. From there, we start to clean the data, deciding to omit any observations where a single column's value is null/na.

During question 1, we are formulating a dataframe that contains a number of features, which are all derived. Return is found by taking the log difference *100 of the original numeric column, and size is found by taking the absolute value of the 'return' column. Direction is really just a binary representation, telling us if our return is up or down based on boolean logic. The date feature is formatted, and omits the first obs. of the initial log function, which is blank due to lag. We've turned a simple 2feature df into one that has 5.

GGplot is R's version of Matplotlib or Seaborn, and is used to produce graphical representations/simulations of data. In Question1, we first show our return on a time series plot. Then we do the same thing with size (abs value of return, used for volatility tracking).

Lastly we generate on overlapping chart that shows both size and return in relation to date. In 1.2, we use a predetermined function to compute summary statistics on our df , then lay that into a table and knitting it for aesthetic purposes. This enables a quick look at some of the more important 'at a glance' data associated with our data's distribution. The use of functions in EDA is essential because it allows for quick, reproducible results and minimizes redundancy in a workflow.

The next few lines just deal with basic tables, where we want to show the frequency distributions responding to certain filters and logic. For example, we can see how many 'dates' are moving in a positive, negative or stagnant direction. Moving into pivoting, we use the Dplyr package/distribution. We follow the three rules of pivoting: filtering, grouping and then summarizing. In this step we are filtering on size being greater than half the max size, grouping by direction and summarizing the table with basic descriptive techniques, followed by running a knit function for aesthetics. We now have a 3 row table, sans header, displaying average return, standard deviation, 5% and 95% quantiles, and percent of the observations within each grouping. We can then render a LaTeX formatted table by using the xtable package and transposing the aforementioned pivot table. We also print a latex table of our summary stats rendered above, while rounding to 2 decimal places using the round function. This concludes part 1.

In part 2, we find our tollerable rate by running the quantile function on our return column, alongside a tolerable pct. of .95. We then plot our data with return on our x axis and filling with direction (y axis). This helps to show us the trends of the dataset with a threshold risk rate displayed. Next, we display our knowledge of functions by wrapping all of the cleansing and work with Dplyer into a single executable function for reproducibility. This function reads the data, cleans it, transforms it and then displays both pivot table and LaTeX tables. Now we can simply run this function on our data and produce these same meaningful insights, day in and day out. Finally, we move into the simulation of future return movements. we utilize both gamma and t distributions to find the optimal fit to our data. :

Data Insights

: The precursor to this lab was to understand returns on Brent over a series of time, as it was noted that earnings forecasts have been missed for over a year. Looking at our summary statistics derived from our data moments function, we can start seeing the distribution of the data over time. Most notably, returns is negatively skewed, at -1.43, which means that our mean(average) is to the left of the peak.

Speaking of peaks, we can see that our kurtosis is 38.25, well above the standard threshold of 13, which tells us that we like have heavy tails, or outliers present. Which leads us into our generated plot using absolute value of returns, something that visually depicts the volatility of our returns, which was noted earlier. By visual representing distributions, we can see periodic changes in actual returns over a period of time, and we can attempt to cluster volatility using absolute returns (disregarding cardinality).

We notice a few outliers, particularly in the early 90's and 00's that are likely distorting the distributions of our data, and wonder if, for this analysis, they would be better served as isolated instances and removed. We also notice that the dawning of 2015 heighted volatility of actualized returns up to the present day. Grouping and summarizing the data within a table helps us to derive meaning behind the directionality of the time series.

One of the most telling statistic derived from this table is the difference between the .05 quantile for up and down periods of time - -4.78% return during a down period, and 0.18% during a good period. This speaks to

the risk-reward of dealing with oil, but could potentially be cleaned up by omitting historical data. Plotting our tolerance for risk helps paint a picture of how risky Brent truly is. We have a downside of >50%, and an upside of 26%, with an equal frequency distribution of up and down time periods. Fitting the data upward and downward moving data with a gamma distribution estimates the gamma parameters, alpha and beta (shape and rate). Constructing the ratio of estimate to the standard error of estimates, we compute the number of standard deviations away from zero our estimates are. Ranging from 39-48, we see that they are pretty far from zero, so we can reject the null hypothesis that the estimates are no different from zero. We start out trying to find the optimal fit for our data using gamma and t distributions because, due to emperical evidence, we know that oil tends to minimize the error moreso than a normal distribution. We also utilize the stddist from the FGarch distribution and notice very similar results to our T distribution, noted above. :

Business Summary

: From a business perspective we are focused on the company's tie to Brent, and how that tie often leaves them exposed with the persistent volatility of crude in relation to variable costs. The manufacturing process of this business is impacted directly by #2 heating oil prices. The variable cost is an important component of product manufacturing as either labor, material or overhead cost that changes according to the change in the volume of production units produced. In this case, oil is identified as a variable cost component of manufacturing overhead. Although unclear, HO2 is most likely being used to generate heat for plant operations, and as the production increases, so does the use of heating oil. Variable costs are also the sum of marginal costs over all units produced. The supply chain mangers should be able to create a more accurate forecast of the variable cost based on the future movement analysis results. This analysis procedure should become part of normal analysis practice and should be performed on a more frequent basis to capture recent data in order to adjusted forecast.

This prediction analysis will be useful in making business process decisions. For example, we can explore entering into HO2 supplier contracts. These contracts are an agreement to purchase a set amount of HO2 at a specific price executed in advance. The idea is to circumvent volatility in HO2 resulting in the contract HO2 price being lower than current price. The company could also adjust the timing of the manufacturing process by possibly scheduling the work to performed at a time where HO2 prices are predicted to be lower based on the forecast.
:

Sources: https://bookdown.org/wfoote01/faur/r-data-modeling.html#estimate-until-morale-improves