Group 4_Project1

Brandon Croarkin, Brian Pierce, Justin Longo, and TS Yeap

July 29, 2018

Project 1 - HO2 Analysis

Part 1: HO2 Data Preperation and Exploration

Problem

```
# Read in data package EIAdata
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
# use na.approx() as well
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 0.3
98 ...
```

Questions

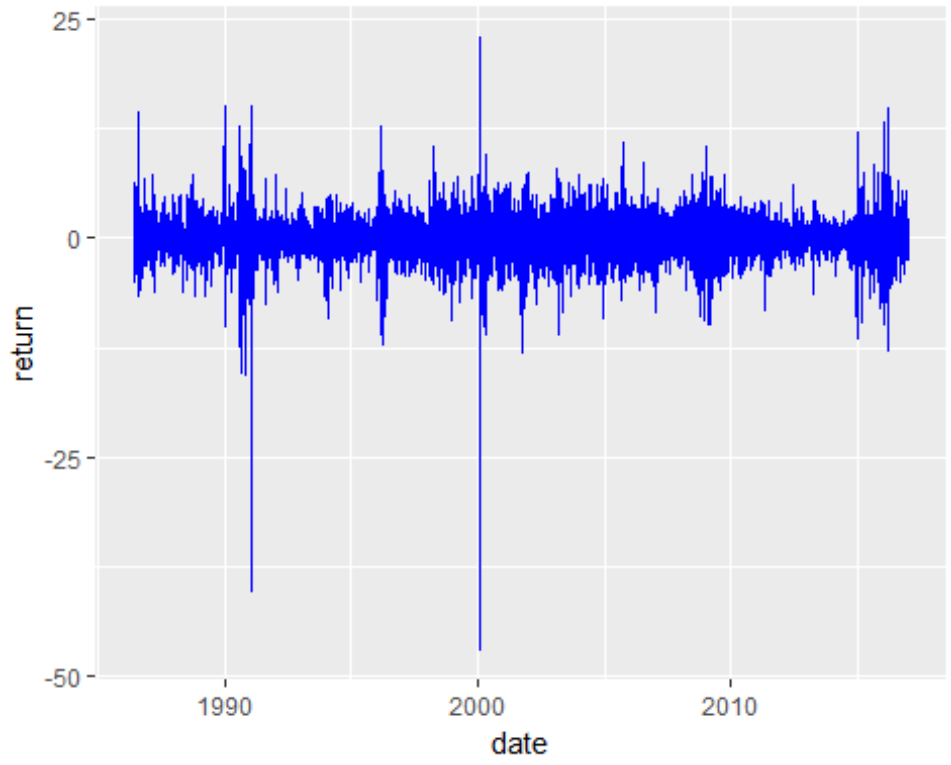
1. What is the nature of HO2 returns? We want to reflect the ups and downs of price movements, something of prime interest to management. First, we calculate percentage changes as log returns. Our interest is in the ups and downs. To look at that we use if and else statements to define a new column called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

```
# Construct expanded data frame
return <- as.numeric(diff(log(HO2$DHOILNYH))) *
    100  # Euler
size <- as.numeric(abs(return))  # size is indicator of volatility
direction <- ifelse(return > 0, "up",
    ifelse(return < 0, "down", "same"))  # another indicator of volatility
```

```
# =if(return > 0, 'up', if(return <
# 0, 'down', 'same'))
date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")  # length of DATE is length of retu
rn +1: omit 1st observation
price <- as.numeric(HO2$DHOILNYH[-1])  # length of DHOILNYH is length of retu
rn +1: omit first observation
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 ...
```

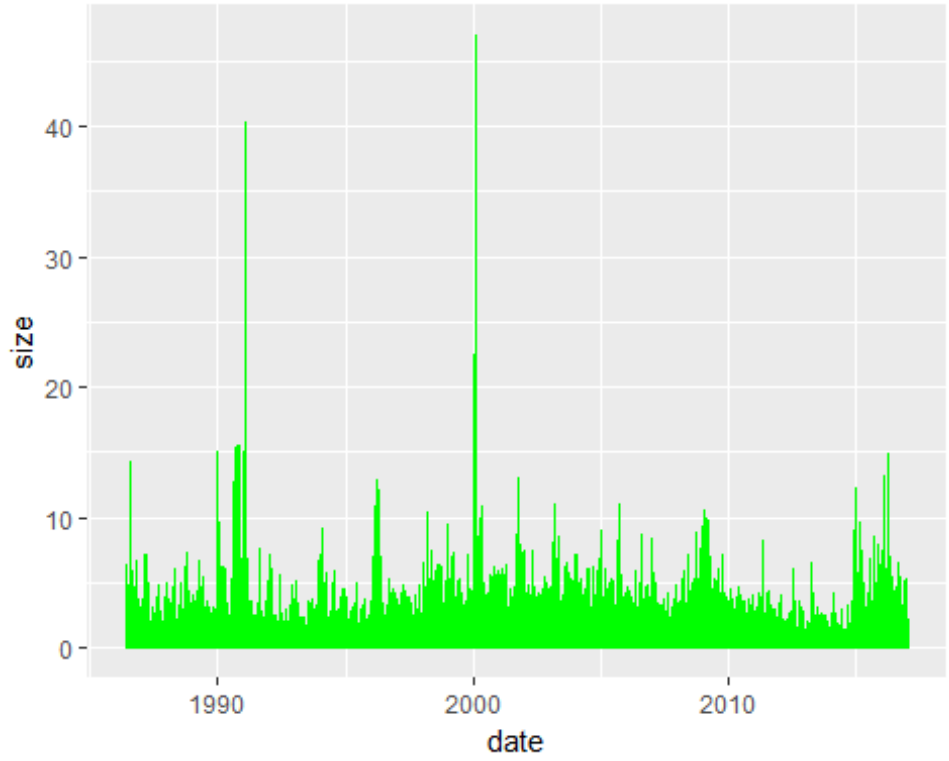
We can plot with the ggplot2 package. In the ggplot statements we use aes, “aesthetics”, to pick x (horizontal) and y (vertical) axes. Use group =1 to ensure that all data is plotted. The added (+) geom_line is the geometrical method that builds the line plot.

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.4

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



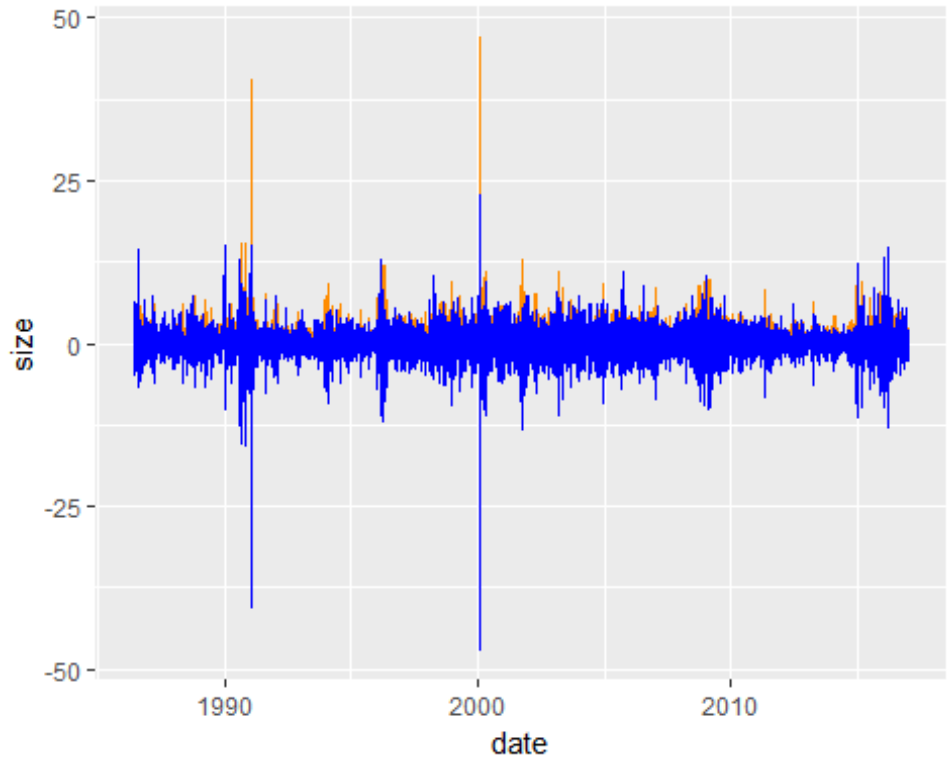
Let's try a bar graph of the absolute value of price rates. We use geom_bar to build this picture.

```
p <- ggplot(HO2.df, aes(x = date, y = size,
    group = 1)) + geom_bar(stat = "identity",
    colour = "green")
p
```



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

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



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:
# vector OUTPUTS: list of scalars
# (mean, sd, median, skewness,
# kurtosis)
data_moments <- function(data) {
    library(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
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.4

##
## 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

## 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))

## Warning: package 'bindrcpp' was built under R version 3.4.4

# 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
library(xtable)

## Warning: package 'xtable' was built under R version 3.4.4

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"))

## % latex table generated in R 3.4.3 by xtable 1.8-2 package
## % Tue Jul 31 20:14:21 2018
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrr}
##   \hline
##  & 1 & 2 & 3 \\
##   \hline
## 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 \\
##    \hline
## \end{tabular}
## \caption{Heating Oil No. 2: 1986-2016}
## \end{table}

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

```
## % latex table generated in R 3.4.3 by xtable 1.8-2 package
## % Tue Jul 31 20:14:21 2018
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrr}
##   \hline
##  & mean & std\_dev & median & skewness & kurtosis \\
##   \hline
## 1 & 0.02 & 2.52 & 0.00 & -1.44 & 38.26 \\
##    \hline
## \end{tabular}
## \end{table}
```

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

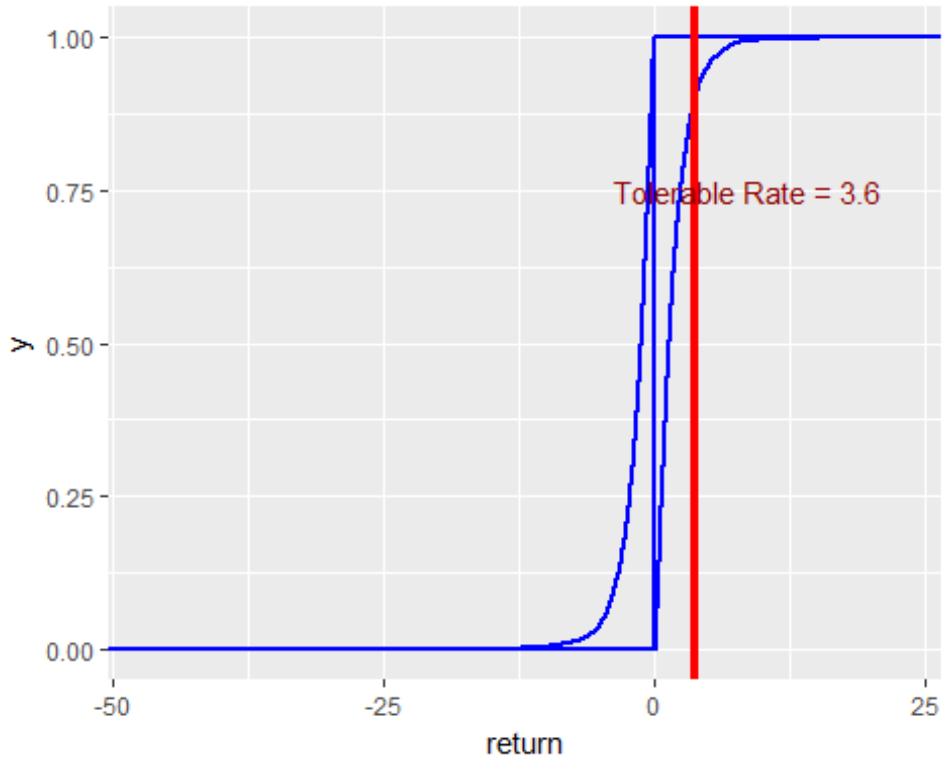
Problem

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? We can use the HO2.df data frame with ggplot2 and the cumulative relative frequency function stat_ecdf to begin to understand this data.

```
HO2.tol.pct <- 0.95
HO2.tol <- quantile(HO2.df$return, HO2.tol.pct)
HO2.tol.label <- paste("Tolerable Rate = ",
    round(HO2.tol, 2), sep = "")
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 + 5,
        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 volatilit
```

```
y
    date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")  # length of DATE is length of
return +1: omit 1st observation
    price <- as.numeric(HO2$DHOILNYH[-1])  # length of DHOILNYH is length of
return +1: omit first observation
    HO2.df <- na.omit(data.frame(date = date,
        price = price, return = return,
        size = size, direction = direction))  # clean up data frame by omitti
ng 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. We will use the gamma distribution to simulate future heating oil #2 price scenarios.

```
library(MASS)

##
## Attaching package: 'MASS'

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

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)

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

fit.gamma.up

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

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

## 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

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)
```

Appendix

List in the text the 'R' skills needed to complete this project

Expanded data frame construction Graphical interface construction Data "moments" - input/manipulation Identify/define data subsets Function creation

Explain each of the functions (e.g., ggplot()) used to compute and visualize results.

Read.csv (data input) - reads in a table format and creates a data frame from it. By default, it reads the first line as a header and uses "," as a separator. Head (reading data) - returns in the first x rows of the dataframe Str (exploring data) - displays the internal structure of

the an R object including the number of observations, number of variables, and data types for each of the columns As.numeric (converting data function) - convert objects to a numeric data type Log - computes logarithms, by default natural logarithms. Abs - computes the absolute value of a number, which converts all numbers to positive as.Date - converts objects to a date data type Na.omit - removes rows with missing values on columns specified Data.frame - converts a list of vectors into a data frame ggplot (graphical function) - Lets you create layered plots with a more detailed look at the data Geom_line - creates a line plot Geom_bar - creates a bar plot Geom_vline - creates a vertical intercept Annotate - adds text to a ggplot graph data_moments (description function) - custom function used for calculating mean, standard deviation, median, kurtosis, and skewness. Outputs the results in a dataframe. Round - rounds the values in its first argument to the specified number of decimal places (default of 0). Table - uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels pivot.table (sort/filter) - used to organize/scale data for broader or more specific analysis Length - gets the length of vectors and factors Summarise (dplyr function) - used on grouped data created by group_by and outputs custom aggregation metrics with one row for each group Quantile - produces sample quantiles corresponding to the given probabilities, with the smallest observation corresponding to a probability of 0 and the largest to a probability of 1. xtable (conversion function) - integrated regression analysis, framing/matrix data assortments to allow printer-friendly versions of output. kable (html table function) - purposefully simplified table generating function. Paste - concatenates vectors after converting to character.
Stat_ecdf - the empirical cumulative distribution function (ECDF) is a non-parametric estimator of the underlying CDF of a random variable. It assigns a probability of to each datum, orders the data from smallest to largest in value, and calculates the sum of the assigned probabilities up to and including each datum.

Discuss how well did the results begin to answer the business questions posed at the beginning of each part of the project.

As part of the initial problem setup, and results, the output of this project clearly answers each business questions in an organized and repeatable fashion. The instructions were followed and completed in order to best represent the most critical data as efficiently as possible in multiple versions, to include : the nature of HO2 returns (volatility), duration, and size. Further analysis included successful computations around statistical measurement. Primarily, the values of expected variables such as mean, median, skewness, kurtosis, etc. Addressing organization and dissecting the data, the pivot table function successfully sorted and filtered the appropriate sets in order to quickly focus on required values. Additionally, the summaries specifically describe HO2's movements while also being available for additional queries, as is expected from the pivot tables' utility.