Project 1 – HO2 Analysis

Live Sessions: weeks 1 and 2

Purpose, Process, Product

With this project we will practice reading, cleaning, and exploring data, building data frames, pivot tables, and plots. We will also use functions to perform repeatable tasks. We will fit a distribution to data. Throughout we will visualize results with plots using `ggplot2` in this project to explore the data and gain further insight into the results of our analysis. We will summarize our findings in a report documented with an `R markdown` file and output.

Assignment

Submit into **Coursework > Assignments and Grading > Project 2 > Submission** an `RMD` file with filename **lastname-firstname_Project2.Rmd**. If you have difficulties submitting a `.Rmd` file, then submit a `.txt` file.

1. Use headers (##), r-chunks for code, and text to build a report that addresses the two parts of this project.
2. List in the text the 'R' skills needed to complete this project.
3. Explain each of the functions (e.g., `ggplot()`) used to compute and visualize results.
4. Discuss how well did the results begin to answer the business questions posed at the beginning of each part of the project.

R Markdown set up

1. Open a new `R Markdown` pdf (or Word) document file and save it with file name **lastname-firstname_Project2** to your working directory. The `Rmd` file extension will automatically be appended to the file name. Create a new folder called `data` in this working directory and deposit the `.csv` file for Project #2 to this directory.
2. Modify the `YAML` header in the `Rmd` file to reflect the name of this project, your name, and date.
3. Replace the `R Markdown` example in the new file with the following script. Modify this script to reflect the parts and questions in the project. The ones below are illlustrative only.

```
# Part 1: HO2 data preparation and exploration
(INSERT explanatory text here)
(INSERT r chunks here)
## Question 1 - title
(INSERT explanatory text here)
(INSERT r chunks here)
## Question 2 - title
(INSERT explanatory text here)
(INSERT r chunks here)
# Part 2: HO2 analysis
(INSERT explanatory text here)
## Further questions as needed
(INSERT explanatory text here)
(INSERT r chunks here)
```

4. Click `knit` in the `Rstudio` command bar to produce the `pdf` or `Word` document.

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 HO2 we download this data set and review some basic aspects of the prices.

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

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 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 omitting NAs
str(HO2.df)
```

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)
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.

```
# library(ggplot2)
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)
```

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$return < 0)  # one way
table(HO2.df$return > 0)
table(HO2.df$direction)  # this counts 0 returns as negative
table(HO2.df$return == 0)
# Pivoting
library(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)
# Here is how we can produce a LaTeX
# formatted and rendered table
library(xtable)
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"))
print(xtable(answer), digits = 2)
```

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 volatility
    date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")  # length of DATE is length of re
turn +1: omit 1st observation
    price <- as.numeric(HO2$DHOILNYH[-1])  # length of DHOILNYH is length of re
turn +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
    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)
```

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)
HO2.data <- HO2_movement(file = "data/nyhh02.csv",
    caption = "HO2 NYH")$df
str(HO2.data)
fit.gamma.up <- fitdistr(HO2.data[HO2.data$direction ==
    "up", "return"], "gamma", hessian = TRUE)
fit.gamma.up
# 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)
fit.t.down
fit.gamma.down <- fitdistr(-HO2.data[HO2.data$direction ==
    "down", "return"], "gamma", hessian = TRUE)  # gamma distribution defined f
or data >= 0
fit.gamma.down
```