SYR-MBA FIN 654 Financial Analytics Practice Set #2

Khan and Khalifa

January 29, 2017

Practice Sets for Data frames, Pivot tables and Metrics

These practice sets practice various `R` features in this chapter. reading in data, constructing data frames, pivoting information, developing metrics, writing functions, and tables. And then, the estimation of distribution parameters and plots using `ggplot2`, followed by summarization of findings.

Set A

Build a data set using filters and `if` and `diff` statements. Answer questions using plots and a pivot table report. Review a function as approach to run some of 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 the data set and review some basic aspects of the prices.

```
# Read in data - stringsAsFactors sets dates as character type
HO2 <- read.csv("data/nyhh02.csv", header = T, stringsAsFactors = F)
HO2 <- na.omit(HO2)  ## to clean up any missing data
head(HO2, n = 3)
```

```
##       DATE DHOILNYH
## 1 6/2/1986    0.402
## 2 6/3/1986    0.393
## 3 6/4/1986    0.378
```

```
str(HO2)  ## review the structure of the data
```

```
## '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

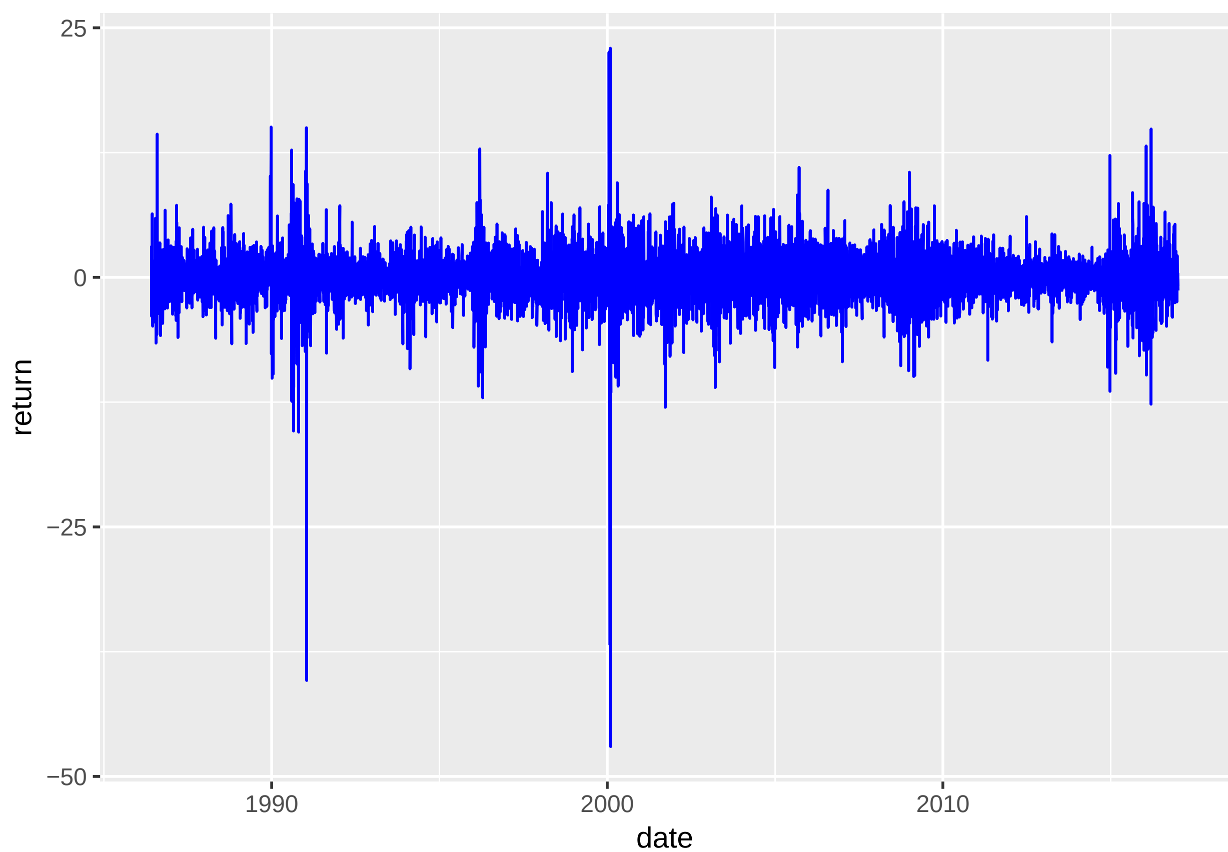
1. What is the nature of HO2 returns? Reflect on the ups and downs of price movements, something of prime interest to management. First, calculate percentage changes as log returns. Focus on the ups and downs - use `if` and `else` statements to define a new column called `direction`. Build a data frame to house this analysis.

```
# Construct expanded data frame
return <- as.numeric(diff(log(HO2$DHOILNYH))) * 100
# size is indicator of volatility
size <- as.numeric(abs(return))
```

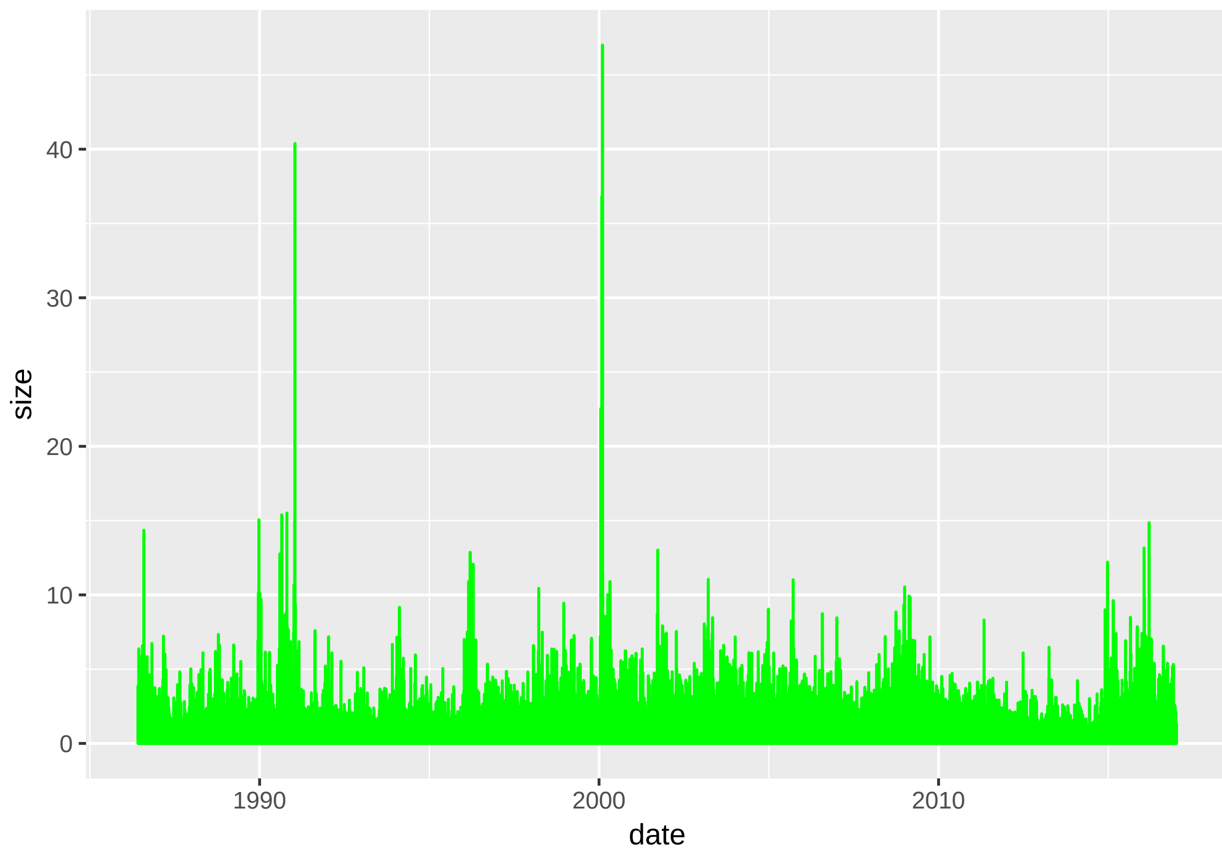
```r
# direction is another indicator of volatility
direction <- ifelse(return > 0, "up", ifelse(return < 0, "down",
    "same"))
# length of DATE is length of return +1: omit 1st observation
date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")
# length of DHOILNYH is length of return +1: omit first
# observation
price <- as.numeric(HO2$DHOILNYH[-1])
# clean up data frame by omitting NAs
HO2.df <- na.omit(data.frame(date = date, price = price, return = return,
    size = size, direction = direction))
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 ...
```

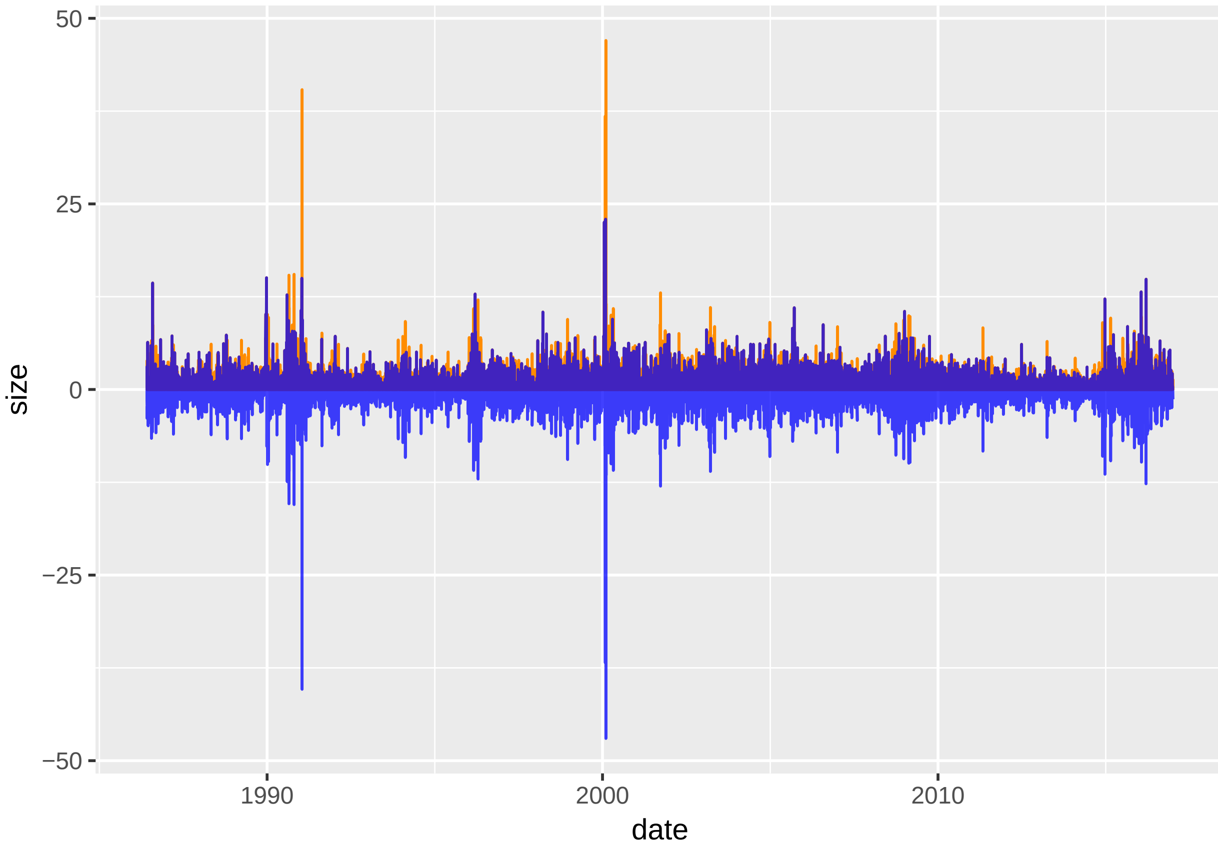
```r
require(ggplot2)
# Plot line graph of return in blue
ggplot(HO2.df, aes(x = date, y = return, group = 1)) + geom_line(colour = "blue")
```




```r
# Plot bar graph of size in green
ggplot(HO2.df, aes(x = date, y = size, group = 1)) + geom_bar(stat = "identity",
    colour = "green")
```




```r
# Plot combined bar and line graphs, overlaying 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",
        alpha = 0.75)
```



2. Dig deeper and compute mean, standard deviation, etc.

```
# Define the data_moments() function using moments package
# functions INPUTS: data 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)  # add a quantile?
    result <- data.frame(mean = mean.r, std_dev = sd.r, median = median.r,
        skewness = skewness.r, kurtosis = kurtosis.r)
    return(result)
}
```

Run the function using the `HO2.df$return` subset and write a `knitr::kable()` report.

```
# 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. Pivot `size` and `return` on `direction`. What is the average and range of returns by direction? How often are there 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)
```

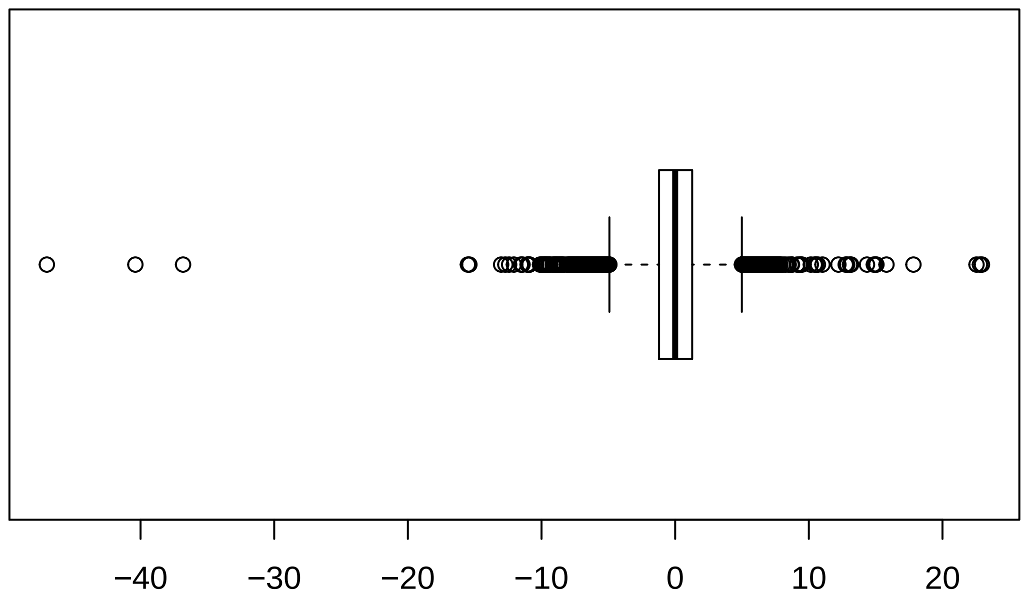
```
##
## FALSE  TRUE
##  7417   279
```

```
# 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
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)
```

```
# 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

```
# Show the statistics as a boxplot.
boxplot(HO2.df$return, horizontal = TRUE)
```



Set B

Use the data from Set A to investigate the distribution of returns we generated.

Problem

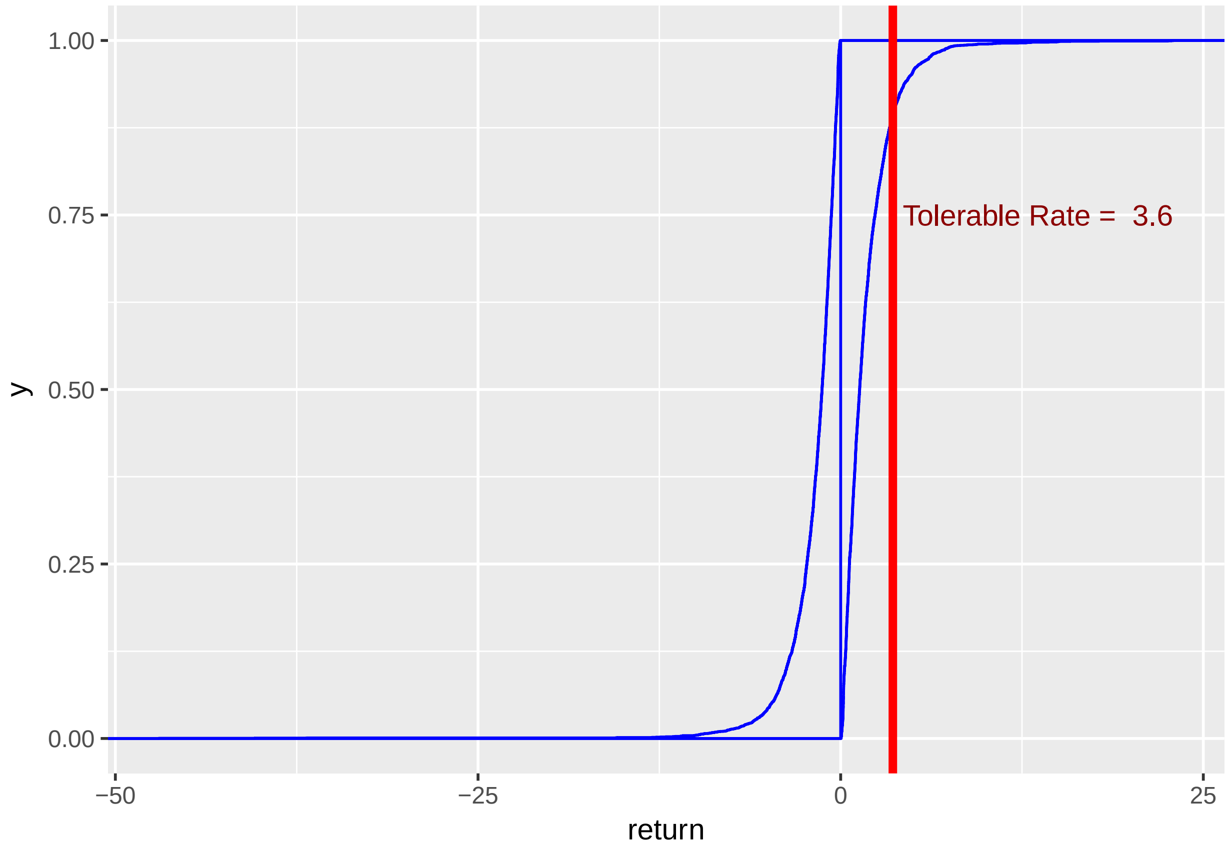
This will entail fitting the data to some parametric distributions as well as writing a function to house results from the previous set.

Further characterize the distribution of up and down movements visually. Also, repeat the analysis periodically for inclusion in management reports.

Questions

1. Show the differences in the shape of ups and downs in HO2, especially given the tolerance for risk? 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 = 0.5) + geom_vline(xintercept = HO2.tol, colour = "red",
    size = 1.5) + annotate("text", x = HO2.tol + 10, y = 0.75, label = HO2.tol.label,
    colour = "darkred")
```



2. Write a function similar to `data_moments` in order to regularly, and reliably, analyze HO2 price movements.

```
# Define the HO2_movement(file, caption) function 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 is indicator of volatility
    size <- as.numeric(abs(return))
    # direction is another indicator of volatility
    direction <- ifelse(return > 0, "up", ifelse(return < 0, "down",
        "same"))
    # length of DATE is length of return +1: omit 1st observation
    date <- as.Date(HO2$DATE[-1], "%m/%d/%Y")
    # length of DHOILNYH is length of return +1: omit first
    # observation
    price <- as.numeric(HO2$DHOILNYH[-1])
    # clean up data frame by omitting NAs
    HO2.df <- na.omit(data.frame(date = date, price = price, return = return,
```

```
        size = size, direction = direction))
    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)
    output.list <- list(table = pivot.table, df = HO2.df)
    return(output.list)
}
```

```
# Test HO2_movement().
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

3. Use the `MASS` package's `fitdistr()` function to find the optimal fit of the HO2 data to a parametric distribution, Student t. This will simulate future movements in HO2 returns.

```
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 ...
```

Find the optimal fit of the HO2 data to Student t parametric distribution.

```
require(MASS)
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.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)
```

Compare with `fGarch` package `stdFit()` function output.

```r
require(fGarch)
Fit.t.down <- stdFit(HO2.data[HO2.data$direction == "down", "return"])
Fit.t.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)"
```

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

The results from both fitdistr() and stdFit() are comparable. The computation sd * sqrt((nu-2)/nu) matches with the value s.

Practice Set Debrief

List the R skills needed to complete these practice sets.

R skills needed include the ability to write conditional statements (ifelse), include (require) packages to access available functions, write and use new functions for repeated use, and understand how to manipulate tables and matrices before passing them to different functions that will produce statistical results and graphs. This includes the use of an interesting function stat_ecdf (Empirical Cumulative Density Function). Further, knowing a command such as lsf.str("package:MASS") provides a quick overview of functions available in package MASS, for example.

What are the packages used to compute and graph results. Explain each of them.

Set A uses packages ggplot2, moments and dplyr. Set B uses packages dplyr and MASS. Package `ggplot2` is a system for creating elegant graphics. You provide the data and the variables that need to be mapped to aesthetics, and what kind of charts to plot (bar, line, pie, etc.) and `ggplot2` does the rest. You can even plot multiple charts on the same graph. Package `moments` provides many of the descriptive statistics functions that are commonly used to understand a data set: mean, median, mode, quartile, standard deviation, skewness and kurtosis. Package `dplyr` houses many tasks for efficient data manipulation, working on data frames and sql-like objects. These tasks include: filter, group_by, select and summarise. Package `MASS` contains a rich set of functions related to distributions. Problem Set B uses the ?maximum likelihood? approach in the fitdistr estimating function to estimate the parameters of the Student t distribution. Another package `fGarch` provides stdFit function that yields similar results.

How well did the results begin to answer the business questions posed at the beginning of each practice set?

The nature of HO2 returns was easily described by first calculating percentage changes as log returns, and then applying descriptive statistics functions and graphs. The ups and downs of price movements could be assessed with a simple conditional check, labeling directions as up/down/same based on relative change. Summary metrics based on the direction grouping was then feasible with the moments package. Finally, fitdistr and stdFit functions helped in establishing the optimal fit of the data to a parametric distribution, Student t, simulating future movements.