Solutions to Laboratory Exercise 2

Simple Data Handling

Problem 2.1 Examine the Excel data file provided to determine the range of cells with the relevant observations. Import the Excel data by using an R package i.e. `readxl`.

Sol: Note that when you use data downloaded from Morningstar Direct®, the generated Excel data file will contain some Summary Statistics at the bottom of the spreadsheet. Given this you should always examine the file before it is downloaded and identify the range of cells that contain the desired information. In this lab we have removed these Summary Statistics from the file. The cells that contain the relevant observations and variables are within the range A1:F629.

Note also that the following is the simplest method of uploading and cleaning the data, since it does not adjust the character arrays containing the stock name, ticker, or exchange information. Future exercises will require us to consider these qualitative characteristics.

R does not have a default function to import Excel data, but there are multiple packages that can help you import external data into R. This solution requires function `read_excel` in the package `readxl`. Feel free to explore alternative options in other packages. If you have never used `readxl` before, you need to install the package first. The code: `install.packages("readxl")` will do it for you. And you will need `library(readxl)` to call this package into R. For more information about `readxl`, you can refer to its website at CRAN https://cran.r-project.org/web/packages/readxl.

```r
# 2.1
library(readxl)
raw = read_excel('lab2_data.xlsx',
                 sheet = 'New Search Criteria',
                 range = 'A1:F629')

head(raw)
```

```
## # A tibble: 6 x 6
##                              Name Ticker           Exchange
##                             <chr>  <chr>              <chr>
## 1                   Ablest, Inc.    AIH NYSE AMEX EQUITIES
## 2 Accelr8 Technology Corporation    AXK NYSE AMEX EQUITIES
## 3        Acme United Corporation    ACU NYSE AMEX EQUITIES
## 4                 ACR Group, Inc.    BRR NYSE AMEX EQUITIES
## 5       Adams Resources & Energy     AE NYSE AMEX EQUITIES
## 6    AdCare Health Systems, Inc.    ADK NYSE AMEX EQUITIES
## # ... with 3 more variables: `Shares Outstanding (mil) - Yearly Year2004` <dbl>,
## #   `Daily Closing Price 2004-12-31 USD` <dbl>, `P/E - Daily 2004-12-31` <dbl>
```

Problem 2.2 Adjust for any missing or erroneous data. To preserve the structure of the variables, R represents most missing or erroneous data with a special NA value. Normal R arithmetic operations yield in NAs when operands are NAs. To avoid this from happening, create a temporary dataset and remove the NAs from the data by using `na.omit`.

Sol: The basic unit of a dataset in R is `data.frame`. You are able to do many kinds of data manipulation in `data.frame`, but the syntax could be verbose and the implementation could be inefficient. Two data handling packages, `data.table` and `dplyr` have earned very high popularity among R users. `data.table` is an extension of `data.frame`, but its implementation is very fast and memory efficient. For more information about `data.table`, see its CRAN website https://cran.r-project.org/web/packages/data.table. Alternately, `dplyr` is not as efficient as `data.table` in handling data of large scale, but its syntax is more familiar to many R users. For more information about `dplyr`, see its CRAN website https://cran.r-project.org/web/packages/dplyr.

This solution requires `data.table` and we will heavily rely on its embedded syntax and functions in the following laboratory exercises. However, there are always alternative approaches to solve the same problem, so again feel free to write your own programs, use other packages and customize your workflow.

Use `setDT` to convert the dataset to `data.table`, and remove any rows containing NAs. You will be left with a 320-by-3 dateset.

```
# 2.2
library(data.table)
tempData = na.omit(setDT(raw))

head(tempData)
```

```
##                               Name Ticker           Exchange
## 1:                    Ablest, Inc.    AIH NYSE AMEX EQUITIES
## 2: Accelr8 Technology Corporation    AXK NYSE AMEX EQUITIES
## 3:        Acme United Corporation    ACU NYSE AMEX EQUITIES
## 4:                ACR Group, Inc.    BRR NYSE AMEX EQUITIES
## 5:       Adams Resources & Energy     AE NYSE AMEX EQUITIES
## 6:         Advanced Photonix Inc.    API NYSE AMEX EQUITIES
##    Shares Outstanding (mil) - Yearly Year2004 Daily Closing Price 2004-12-31 USD
## 1:                                   2876615                             7.38
## 2:                                   9961210                             2.10
## 3:                                   3405921                            15.70
## 4:                                  10712294                             3.05
## 5:                                   4217596                            17.64
## 6:                                  13430750                             1.82
##    P/E - Daily 2004-12-31
## 1:               7.028571
## 2:             -11.666670
## 3:              21.805560
## 4:               8.970588
## 5:               9.639344
## 6:              22.750000
```

Problem 2.3 Rename the following variables in the dataset: **shsOut** for shares outstanding, **clsgPrice** for closing price, and **pE** for price-to-earning (P/E) ratio. Create two additional variables: for the natural logarithm of the market value of equity (or market capitalization) called **lnSize** from shsOut and **clsgPrice**, and for the earnings-to-price (E/P) ratio, which is the inverse of the P/E ratio called **eP**. Plot the data in a histogram to visualize the distribution of the **lnSize** and **eP** variables.

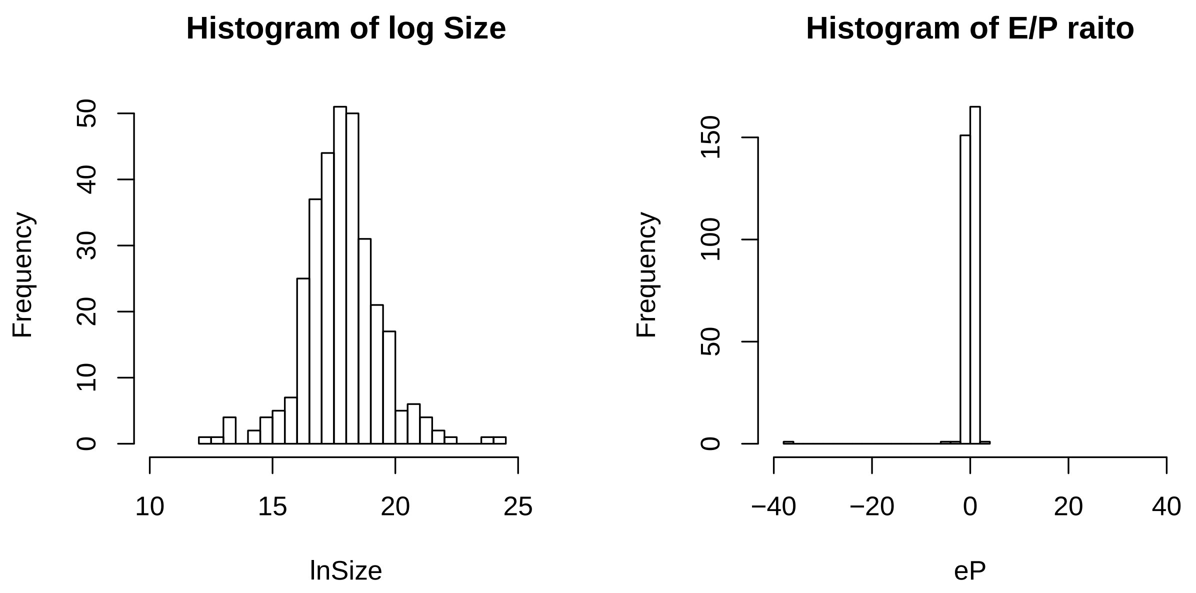
Sol: Rename the variables in the temporary dataset, and create a subset that only includes the required ones.

```
# 2.3
setnames(tempData, c('Name', 'Ticker', 'Exchange',
                     'shsOut', 'clsgPrice', 'pE'))

Data = tempData[, .(Ticker,
                    lnSize = log(shsOut * clsgPrice),
                    eP = 1 / pE)]

par(mfrow = c(1, 2))
```

```r
hist(Data[, lnSize],
     breaks = 20, xlim = c(10, 26),
     xlab = 'lnSize',
     main = 'Histogram of log Size')
hist(Data[, eP],
     breaks = 20, xlim = c(-40, 40),
     xlab = 'eP',
     main = 'Histogram of E/P raito')
```



```r
par(mfrow = c(1, 1))
```

Problem 2.4 Calculate the following basic descriptive statistics for **lnSize** and **eP** in this sample data: (a) measures of location and central tendency: mean and median, (b) measures of scale or dispersion: variance and standard deviation, and interquartile range and (c) measure of distribution or shape: percentile i.e. 5%, 25%, 50%, 75%, and 95%.

Sol: We can use the `apply` family in R to calculate descriptive statistics for all the numeric columns in the dataset.

```r
# 2.4 (type = 2 is MatLab algorithm)
sapply(Data[, .(lnSize, eP)],
       function(x) c(mean = mean(x), median = median(x),
                     var = var(x), sd = sd(x),
                     igr = IQR(x, type = 2),
                     qtl = quantile(x, type = 2,
                                    probs = c(0.05, 0.25, 0.50, 0.75, 0.95))))
```

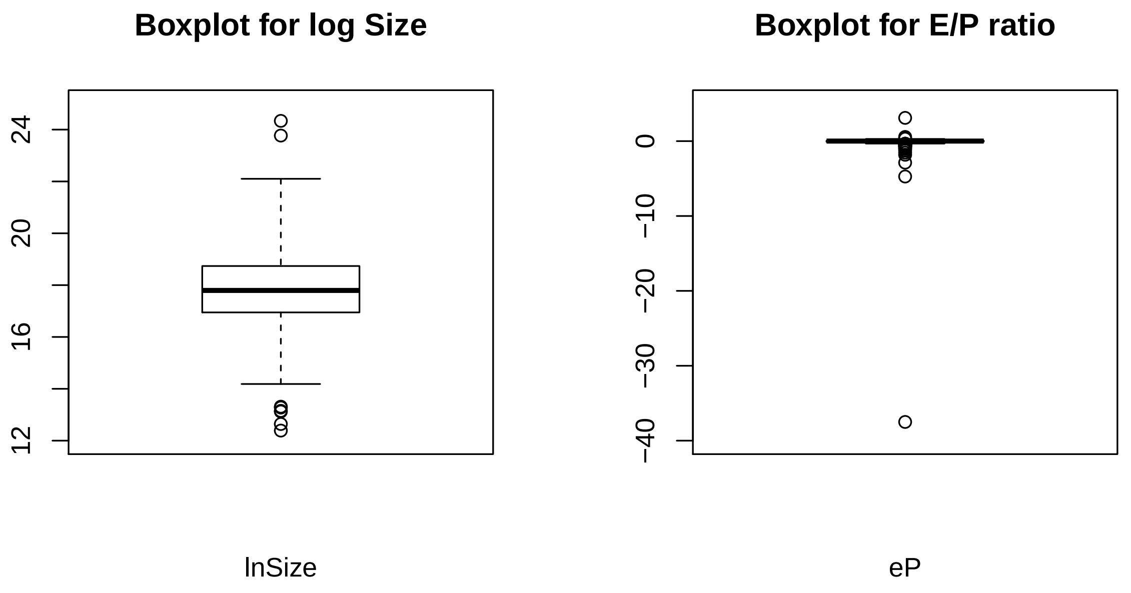
```
##              lnSize           eP
## mean      17.825871 -0.184287294
## median    17.795192  0.005164414
## var        2.437361  4.558459938
## sd         1.561205  2.135055020
## igr       1.789630  0.129073828
## qtl.5%   15.330371 -0.515317387
## qtl.25% 16.948057 -0.074166647
## qtl.50% 17.795192  0.005164414
## qtl.75% 18.737687  0.054907181
## qtl.95% 20.285780  0.114093150
```

Problem 2.5 Outliers are data values that are dramatically different from patterns in the rest of the data, i.e., 3 standard deviations away from their mean.

(a) Plot a box plot and identify any presence of outliers.

Sol: Plot a box plot for each of the variables with the boxplot function. Box plots also describe the distribution of data values and they mark any outliers with a circle. By default, for the boxplot function an outlier is a value that it is more than 1.5 times the interquartile range away from the top and/or bottom of the box. However, we have adjusted this value to be 2 times the interquartile range that it is approximately 3 standard deviations from the mean. Both boxplots show the presence of outliers in our sample.

```
# 2.5 (a)
par(mfrow = c(1, 2))
boxplot(Data[, lnSize],
        range = 2, ylim = c(12, 25),
        main = 'Boxplot for log Size',
        xlab = 'lnSize')
boxplot(Data[, eP],
        range = 2, ylim = c(-40, 5),
        main = 'Boxplot for E/P ratio',
        xlab = 'eP')
```



```
par(mfrow = c(1, 1))
```

(b) To confirm the results of the box plot, calculate the number of outliers for **lnSize** and **eP**, create two new columns that can differentiate the outliers and non-outliers (i.e. keep the non-outliers as the original observations but replace the outliers as missing), and make the box plots and histograms again for the new columns.

Sol: To quantify the presence of outliers, first create a function, `is.out`, that identifies the outliers from a series of numeric. Then, apply this function to each column and sum the result. Using this function in the dataset, we then create two new variables, **lnSize1** and **eP1**, that identify outliers as NAs but keep non-outliers as the original observations. Draw the histograms and boxplots of the new variables, respectively.

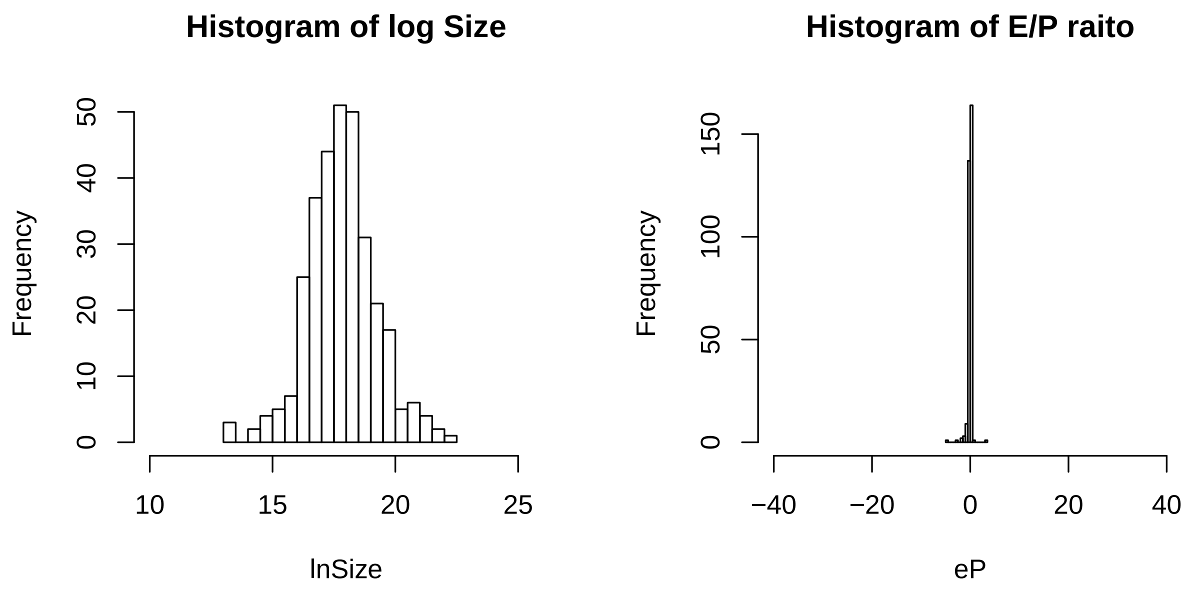
```r
# 2.5 (b)
is.out = function(x) abs(x - mean(x)) > 3 * sd(x)
sapply(Data[, .(lnSize, eP)], function(x) c(Nout = sum(is.out(x))))
```

```
## lnSize.Nout     eP.Nout
##           5           1
```

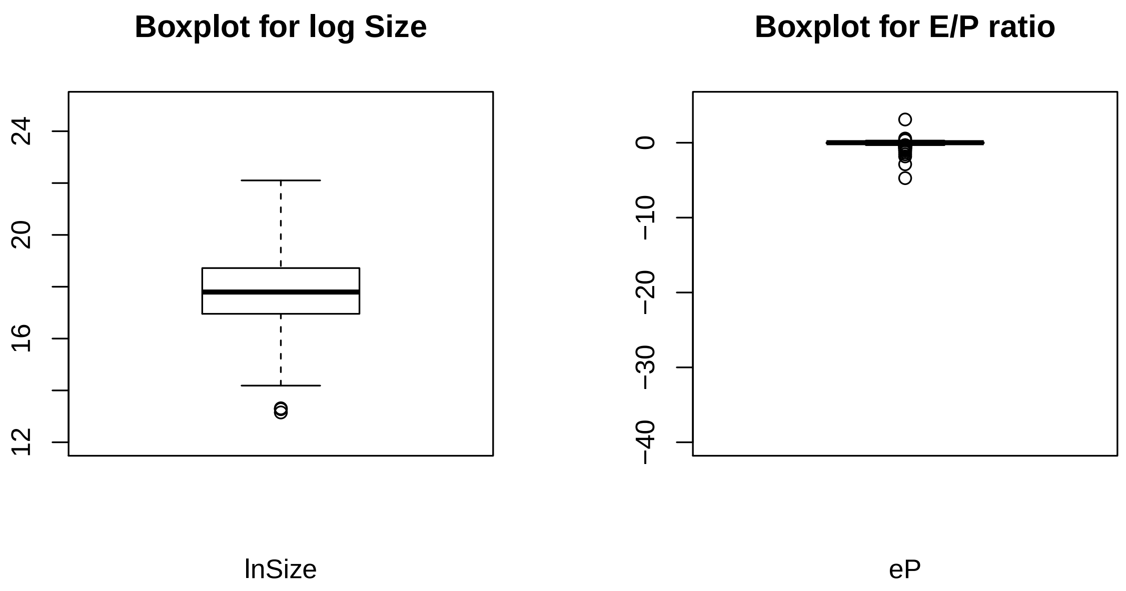
```r
Data[!is.out(lnSize), lnSize1 := lnSize]
Data[!is.out(eP), eP1 := eP]

w3sd = function(x) c(mean(x) - 3 * sd(x), mean(x) + 3 * sd(x))
.range = t(Data[, lapply(.SD, w3sd), .SDcols = c("lnSize", "eP")])

par(mfrow = c(1, 2))
hist(Data[, lnSize1], breaks = 20, xlim = c(10, 26),
     xlab = 'lnSize',
     main = 'Histogram of log Size')
hist(Data[, eP1], breaks = 20, xlim = c(-40, 40),
     xlab = 'eP',
     main = 'Histogram of E/P raito')
```



```r
boxplot(Data[, lnSize1], range = 2, ylim = c(12, 25),
        main = 'Boxplot for log Size',
        xlab = 'lnSize')
boxplot(Data[, eP1], range = 2, ylim = c(-40, 5),
        main = 'Boxplot for E/P ratio',
        xlab = 'eP')
```



```
par(mfrow = c(1, 1))
```

lnSize has 5 outliers, and **eP** has 1 outlier. For both **lnSize1** and **eP1** the histograms show a much tighter distribution, while the boxplots demonstrate that their individual values are within the 3 standard deviations, i.e., within [13.14, 22.51] for **lnSize** and within [-6.59, 6.22] for **eP**.

Problem 2.6 Calculate the z-score for each of the variables, plot the results, and explain the reason for standardizing factor exposures.

Sol: Create z-scores in the dataset. We standardize factor exposures in order to make the data comparable in magnitude and dispersion before further analysis. A z-score is the stock's i value of factor k minus the cross-sectional average value of factor k, normalized by the standard deviation of stock's i value of factor k. Since the result of this standardization process is a standard normal random variable z with mean zero and standard deviation of one, a z-score expresses the exposure relative to the standard deviation. This tells us how many standard deviation units away from the average a particular stock's exposure is, which facilitates stock comparison across multiple characteristics.

```
# 2.6
Data[, zlnSize1 := scale(lnSize1)]
Data[, zeP1 := scale(eP1)]

head(Data)
```

```
##    Ticker   lnSize          eP  lnSize1         eP1    zlnSize1        zeP1
## 1:    AIH 16.87090  0.14227643 16.87090  0.14227643 -0.68542575  0.49394576
## 2:    AXK 16.85615 -0.08571426 16.85615 -0.08571426 -0.69591376 -0.04338168
## 3:    ACU 17.79469  0.04585986 17.79469  0.04585986 -0.02865332  0.26671163
## 4:    BRR 17.30204  0.11147541 17.30204  0.11147541 -0.37890025  0.42135406
## 5:     AE 18.12494  0.10374150 18.12494  0.10374150  0.20614564  0.40312681
## 6:    API 17.01189  0.04395604 17.01189  0.04395604 -0.58518423  0.26222472
```