

# Chapter 1

## Descriptive Statistics for Financial Data

Updated: February 3, 2015

In this chapter we use graphical and numerical descriptive statistics to study the distribution and dependence properties of daily and monthly asset returns on a number of representative assets. The purpose of this chapter is to introduce the techniques of exploratory data analysis for financial time series and to document a set of stylized facts for monthly and daily asset returns that will be used in later chapters to motivate probability models for asset returns.

The R packages used in this Chapter are **corrplot**, **PerformanceAnalytics**, **tseries** and **zoo**. Make sure these packages are installed and loaded before running the R examples.

### 1.1 Univariate Descriptive Statistics

Let  $\{R_t\}$  denote a univariate time series of asset returns (simple or continuously compounded). Throughout this chapter we will assume that  $\{R_t\}$  is a covariance stationary and ergodic stochastic process such that

$$\begin{aligned} E[R_t] &= \mu \text{ independent of } t, \\ \text{var}(R_t) &= \sigma^2 \text{ independent of } t, \\ \text{cov}(R_t, R_{t-j}) &= \gamma_j \text{ independent of } t, \\ \text{corr}(R_t, R_{t-j}) &= \gamma_j / \sigma^2 = \rho_j \text{ independent of } t. \end{aligned}$$

## 2CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

In addition, we will assume that each  $R_t$  is identically distributed with unknown pdf  $f_R(r)$ .

An observed sample of size  $T$  of historical asset returns  $\{r_t\}_{t=1}^T$  is assumed to be a realization from the stochastic process  $\{R_t\}$  for  $t = 1, \dots, T$ . That is,

$$\{r_t\}_{t=1}^T = \{R_1 = r_1, \dots, R_T = r_T\}$$

The goal of *exploratory data analysis* is to use the observed sample  $\{r_t\}_{t=1}^T$  to learn about the unknown pdf  $f_R(r)$  as well as the time dependence properties of  $\{R_t\}$ .

### 1.1.1 Example Data

We illustrate the descriptive statistical analysis using daily and monthly adjusted closing prices on Microsoft stock and the S&P 500 index over the period January 1, 1998 and May 31, 2012.<sup>1</sup> These data are obtained from `finance.yahoo.com`. We first use the daily and monthly data to illustrate descriptive statistical analysis and to establish a number of stylized facts about the distribution and time dependence in daily and monthly returns.

**Example 1** *Getting daily and monthly adjusted closing price data from Yahoo! in R*

As described in chapter 1, historical data on asset prices from `finance.yahoo.com` can be downloaded and loaded into R automatically in a number of ways. Here we use the `get.hist.quote()` function from the `tseries` package to get daily adjusted closing prices and end-of-month adjusted closing prices on Microsoft stock (ticker symbol `msft`) and the S&P 500 index (ticker symbol `^gspc`):<sup>2</sup>

```
> msftPrices = get.hist.quote(instrument="msft", start="1998-01-01",  
+                             end="2012-05-31", quote="AdjClose",
```

---

<sup>1</sup>An adjusted closing price is adjusted for dividend payments and stock splits. Any dividend payment received between closing dates are added to the close price. If a stock split occurs between the closing dates then the all past prices are divided by the split ratio.

<sup>2</sup>The ticker symbol `^gspc` refers to the actual S&P 500 index, which is not a tradable security. There are several mutual funds (e.g., Vanguard's S&P 500 fund with ticker `VFINF`) and exchange traded funds (e.g., State Street's SPDR S&P 500 ETF with ticker `SPY`) which track the S&P 500 index that are investable.

```
+             provider="yahoo", origin="1970-01-01",
+             compression="m", retclass="zoo")
> sp500Prices = get.hist.quote(instrument="^gspc", start="1998-01-01",
+                             end="2012-05-31", quote="AdjClose",
+                             provider="yahoo", origin="1970-01-01",
+                             compression="m", retclass="zoo")
> msftDailyPrices = get.hist.quote(instrument="msft", start="1998-01-01",
+                                  end="2012-05-31", quote="AdjClose",
+                                  provider="yahoo", origin="1970-01-01",
+                                  compression="d", retclass="zoo")
> sp500DailyPrices = get.hist.quote(instrument="^gspc", start="1998-01-01",
+                                   end="2012-05-31", quote="AdjClose",
+                                   provider="yahoo", origin="1970-01-01",
+                                   compression="d", retclass="zoo")
> class(msftPrices)
[1] "zoo"
> colnames(msftPrices)
[1] "AdjClose"
> start(msftPrices)
[1] "1998-01-02"
> end(msftPrices)
[1] "2012-05-01"
> head(msftPrices, n=3)
      AdjClose
1998-01-02  13.53
1998-02-02  15.37
1998-03-02  16.24
> head(msftDailyPrices, n=3)
      AdjClose
1998-01-02  11.89
1998-01-05  11.83
1998-01-06  11.89
```

The objects `msftPrices`, `sp500Prices`, `msftDailyPrices`, and `sp500DailyPrices` are of class "zoo" and each have a column called `AdjClose` containing the end-of-month adjusted closing prices. Notice, however, that the dates asso-

## 4 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

ciated with the monthly closing prices are beginning-of-month dates.<sup>3</sup> It will be helpful for our analysis to change the column names in each object, and to change the class of the date index for the monthly prices to "yearmon"

```
> colnames(msftPrices) = colnames(msftDailyPrices) = "MSFT"
> colnames(sp500Prices) = colnames(sp500DailyPrices) = "SP500"
> index(msftPrices) = as.yearmon(index(msftPrices))
> index(sp500Prices) = as.yearmon(index(sp500Prices))
```

It will also be convenient to create merged "zoo" objects containing both the Microsoft and S&P500 prices

```
> msftSp500Prices = merge(msftPrices, sp500Prices)
> msftSp500DailyPrices = merge(msftDailyPrices, sp500DailyPrices)
> head(msftSp500Prices, n=3)
      MSFT  SP500
Jan 1998 13.53  980.3
Feb 1998 15.37 1049.3
Mar 1998 16.24 1101.8
> head(msftSp500DailyPrices, n=3)
      MSFT SP500
1998-01-02 11.89 975.0
1998-01-05 11.83 977.1
1998-01-06 11.89 966.6
```

We create "zoo" objects containing simple returns using the **PerformanceAnalytics** function `Return.calculate()`

```
> msftRetS = Return.calculate(msftPrices, method="simple")
> msftDailyRetS = Return.calculate(msftDailyPrices, method="simple")
> sp500RetS = Return.calculate(sp500Prices, method="simple")
> sp500DailyRetS = Return.calculate(sp500DailyPrices, method="simple")
> msftSp500RetS = Return.calculate(msftSp500Prices, method="simple")
> msftSp500DailyRetS = Return.calculate(msftSp500DailyPrices, method="simple")
```

We remove the first NA value of each object to avoid problems that some R functions have when missing values are encountered

---

<sup>3</sup>When retrieving monthly data from Yahoo!, the full set of data contains the open, high, low, close, adjusted close, and volume for the month. The convention in Yahoo! is to report the date associated with the open price for the month.

```
> msftRetS = msftRetS[-1]
> msftDailyRetS = msftDailyRetS[-1]
> sp500RetS = sp500RetS[-1]
> sp500DailyRetS = sp500DailyRetS[-1]
> msftSp500RetS = msftSp500RetS[-1]
> msftSp500DailyRetS = msftSp500DailyRetS[-1]
```

We also create "zoo" objects containing continuously compounded (cc) returns

```
> msftRetC = log(1 + msftRetS)
> sp500RetC = log(1 + sp500RetS)
> MSFTsp500RetC = merge(msftRetC, sp500RetC)
```



### 1.1.2 Time Plots

A natural graphical descriptive statistic for time series data is a *time plot*. This is simply a line plot with the time series data on the y-axis and the time index on the x-axis. Time plots are useful for quickly visualizing many features of the time series data.

**Example 2** *Time plots of monthly prices and returns.*

A two-panel plot showing the monthly prices is given in Figure 1.1, and is created using the `plot` method for "zoo" objects:

```
> plot(msftSp500Prices, main="", lwd=2, col="blue")
```

The prices exhibit random walk like behavior (no tendency to revert to a time independent mean) and appear to be non-stationary. Both prices show two large boom-bust periods associated with the dot-com period of the late 1990s and the run-up to the financial crisis of 2008. Notice the strong common trend behavior of the two price series.

A time plot for the monthly returns is created using:

```
> my.panel <- function(...) {
+   lines(...)
+   abline(h=0)
+ }
> plot(msftSp500RetS, main="", panel=my.panel, lwd=2, col="blue")
```

## 6 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

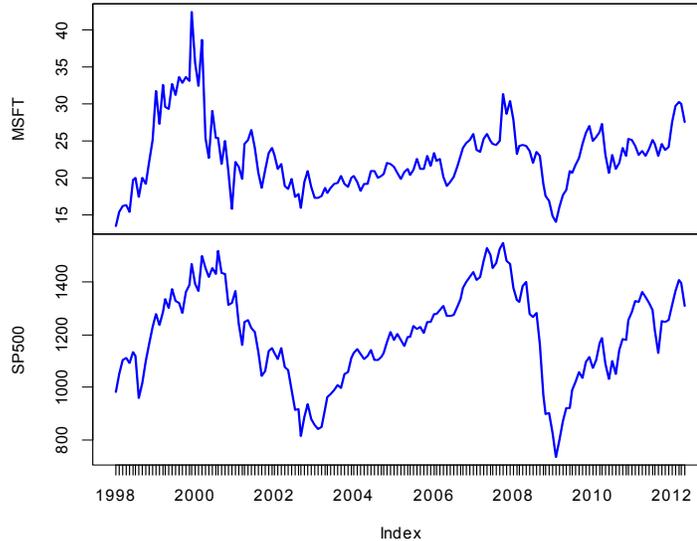


Figure 1.1: End-of-month closing prices on Microsoft stock and the S&P 500 index.

and is given in Figure 1.2. The horizontal line at zero in each panel is created using the custom panel function `my.panel()` passed to `plot()`. In contrast to prices, returns show clear mean-reverting behavior and the common monthly mean values look to be very close to zero. Hence, the common mean value assumption of covariance stationarity looks to be satisfied. However, the volatility (i.e., fluctuation of returns about the mean) of both series appears to change over time. Both series show higher volatility over the periods 1998 - 2003 and 2008 - 2012 than over the period 2003 - 2008. This is an indication of possible non-stationarity in volatility.<sup>4</sup> Also, the coincidence of high and low volatility periods across assets suggests a common driver to the time varying behavior of volatility. There does not appear to be any visual evidence of systematic time dependence in the returns. Later on we will see that the estimated autocorrelations are very close to zero. The returns for

---

<sup>4</sup>The returns can still be covariance stationary and exhibit time varying *conditional* volatility.

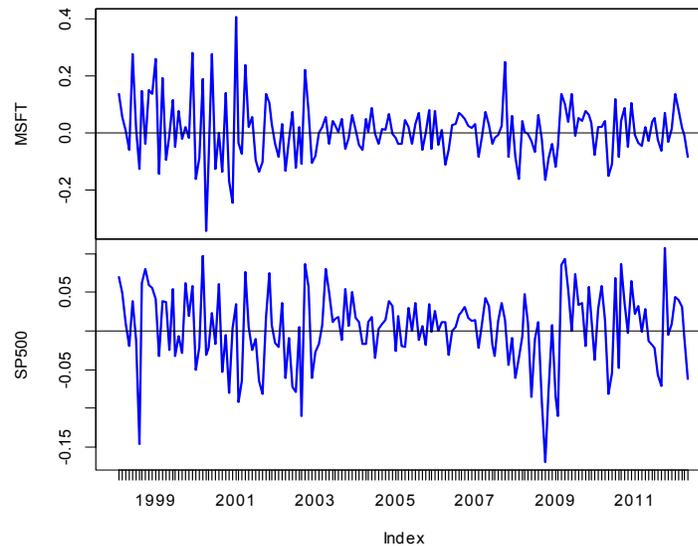


Figure 1.2: Monthly continuously compounded returns on Microsoft stock and the S&P 500 index.

Microsoft and the S&P 500 tend to go up and down together suggesting a positive correlation. ■

**Example 3** *Plotting returns on the same graph*

In Figure 1.2, the volatility of the returns on Microsoft and the S&P 500 looks to be similar but this is illusory. The y-axis scale for Microsoft is much larger than the scale for the S&P 500 index and so the volatility of Microsoft returns is actually much larger than the volatility of the S&P 500 returns. Figure 1.3 shows both returns series on the same time plot created using

```
> plot(msftSp500RetS, plot.type="single", main="",
+      col = c("red", "blue"), lty=c("dashed", "solid"),
+      lwd=2, ylab="Returns")
> abline(h=0)
> legend(x="bottomright", legend=colnames(msftSp500RetS),
```

## 8CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

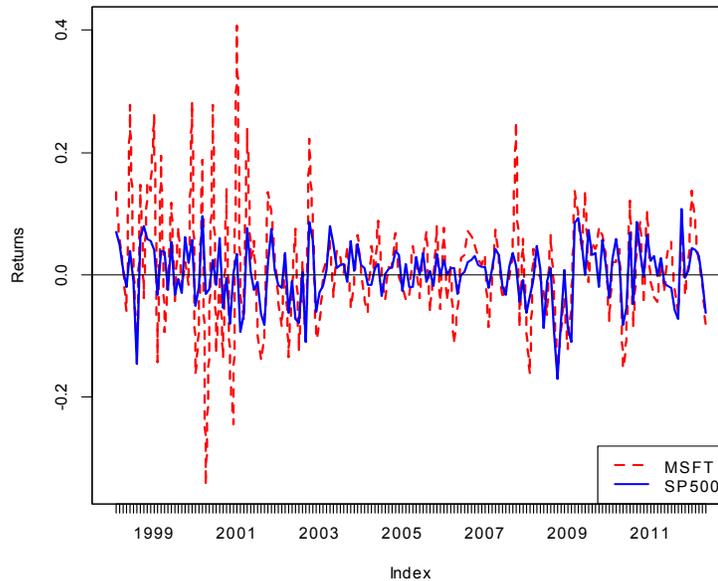


Figure 1.3: Monthly continuously compounded returns for Microsoft and S&P 500 index on the same graph.

```
+      lty=c("dashed", "solid"), lwd=2,  
+      col=c("red", "blue"))
```

Now the higher volatility of Microsoft returns, especially before 2003, is clearly visible. However, after 2008 the volatilities of the two series look quite similar. In general, the lower volatility of the S&P 500 index represents risk reduction due to holding a large diversified portfolio. ■

### **Example 4** *Comparing simple and continuously compounded returns*

Figure 1.4 compares the simple and cc monthly returns for Microsoft created using

```
> retDiff = msftRetS - msftRetC  
> dataToPlot = merge(msftRetS, msftRetC, retDiff)  
> plot(dataToPlot, plot.type="multiple", main="",
```

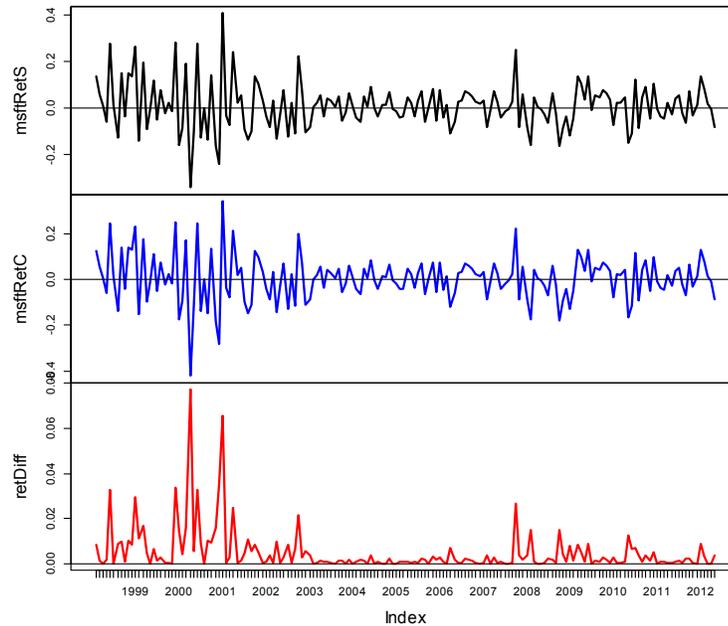


Figure 1.4: Monthly simple and cc returns on Microsoft. Top panel: simple returns; middle panel: cc returns; bottom panel: difference between simple and cc returns.

```
+ panel = my.panel,
+ lwd=2, col=c("black", "blue", "red"))
```

The top panel shows the simple returns, the middle panel shows the cc returns, and the bottom panel shows the difference between the simple and cc returns. Qualitatively, the simple and cc returns look almost identical. The main differences occur when the simple returns are large in absolute value (e.g., mid 2000, early 2001 and early 2008). In this case the difference between the simple and cc returns can be fairly large (e.g., as large as 0.08 in mid 2000). When the simple return is large and positive, the cc return is not quite as large; when the simple return is large and negative, the cc return is a larger negative number.

#### Example 5 *Plotting Daily Returns*

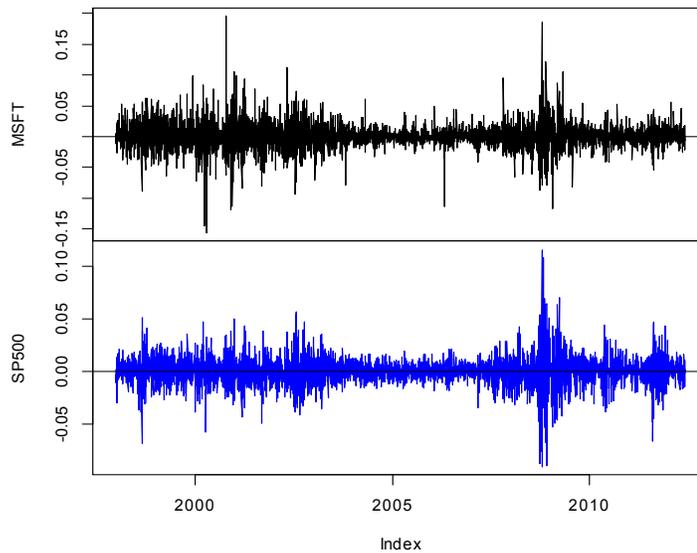


Figure 1.5: Daily returns on Microsoft and the S&P 500 index.

Figure 1.5 shows the daily simple returns on Microsoft and the S&P 500 index created with

```
> plot(msftSp500DailyRetS, main="",
+      panel=my.panel, col=c("black", "blue"))
```

Compared with the monthly returns, the daily returns are closer to zero and the magnitude of the fluctuations (volatility) is smaller. However, the clustering of periods of high and low volatility is more pronounced in the daily returns. As with the monthly returns, the volatility of the daily returns on Microsoft is larger than the volatility of the S&P 500 returns. Also, the daily returns show some large and small “spikes” that represent unusually large (in absolute value) daily movements (e.g., Microsoft up 20% in one day and down 15% on another day). The monthly returns do not exhibit such extreme movements relative to typical volatility.

## Equity Curves

To directly compare the investment performance of two or more assets, plot the simple multi-period cumulative returns of each asset on the same graph. This type of graph, sometimes called an *equity curve*, shows how a one dollar investment amount in each asset grows over time. Better performing assets have higher equity curves. For simple returns, the  $k$ -period returns are  $R_t(k) = \prod_{j=0}^{k-1} (1 + R_{t-j})$  and represent the growth of one dollar invested for  $k$  periods. For continuously compounded returns, the  $k$ -period returns are  $r_t(k) = \sum_{j=0}^{k-1} r_{t-j}$ . However, this cc  $k$ -period return must be converted to a simple  $k$ -period return, using  $R_t(k) = \exp(r_t(k)) - 1$ , to properly represent the growth of one dollar invested for  $k$  periods.

### Example 6 *Equity curves for Microsoft and S&P 500 monthly returns*

To create the equity curves for Microsoft and the S&P 500 index based on simple returns use:

```
> equityCurveMsft = cumprod(1 + msftRetS)
> equityCurveSp500 = cumprod(1 + sp500RetS)
> dataToPlot = merge(equityCurveMsft, equityCurveSp500)
> plot(dataToPlot, plot.type="single", ylab="Cumulative Returns",
+       col=c("black", "blue"), lwd=2)
> legend(x="topright", legend=c("MSFT", "SP500"),
+       col=c("black", "blue"), lwd=2)
```

The R function `cumprod()` creates the cumulative products needed for the equity curves. Figure 1.6 shows that a one dollar investment in Microsoft dominated a one dollar investment in the S&P 500 index over the given period. In particular, \$1 invested in Microsoft grew to about \$2.10 (over about 12 years) whereas \$1 invested in the S&P 500 index only grew to about \$1.40. Notice the huge increases and decreases in value of Microsoft during the dot-com bubble and bust over the period 1998 - 2001. ■

## Drawdowns

To be completed

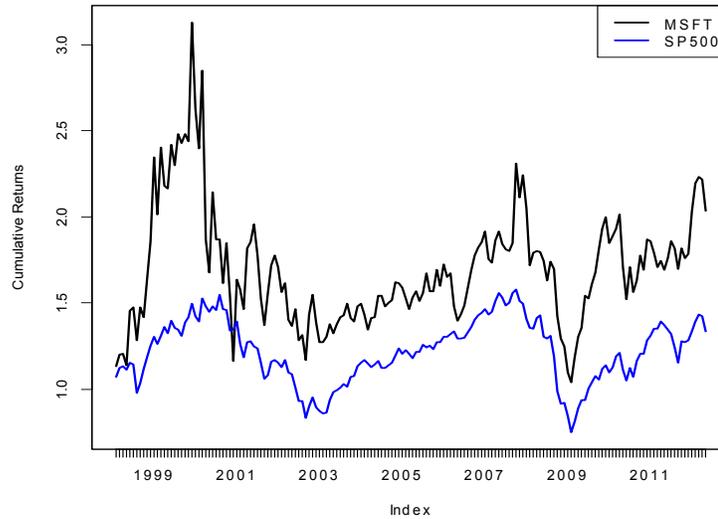


Figure 1.6: Monthly cumulative continuously compounded returns on Microsoft and the S&P 500 index.

### 1.1.3 Descriptive Statistics for the Distribution of Returns

In this section, we consider graphical and numerical descriptive statistics for the unknown marginal pdf,  $f_R(r)$ , of returns. Recall, we assume that the observed sample  $\{r_t\}_{t=1}^T$  is a realization from a covariance stationary and ergodic time series  $\{R_t\}$  where each  $R_t$  is a continuous random variable with common pdf  $f_R(r)$ . The goal is to use  $\{r_t\}_{t=1}^T$  to describe properties of  $f_R(r)$ .

We study returns and not prices because prices are non-stationary. Sample descriptive statistics are only meaningful for covariance stationary and ergodic time series.

#### Histograms

A *histogram* of returns is a graphical summary used to describe the general shape of the unknown pdf  $f_R(r)$ . It is constructed as follows. Order returns from smallest to largest. Divide the range of observed values into  $N$  equally

sized bins. Show the number or fraction of observations in each bin using a bar chart.

**Example 7** *Histograms for the daily and monthly returns on Microsoft and the S&P 500 index*

Figure 1.7 shows the histograms of the daily and monthly returns on Microsoft stock and the S&P 500 index created using the R function `hist()`:

```
> par(mfrow=c(2,2))
> hist(msftRetS, main="", col="cornflowerblue")
> hist(msftDailyRetS, main="", col="cornflowerblue")
> hist(sp500RetS, main="", col="cornflowerblue")
> hist(sp500DailyRetS, main="", col="cornflowerblue")
> par(mfrow=c(1,1))
```

All histograms have a bell-shape like the normal distribution. The histograms of daily returns are centered around zero and those for the monthly return are centered around values slightly larger than zero. The bulk of the daily (monthly) returns for Microsoft and the S&P 500 are between -5% and 5% (-20% and 20%) and -3% and 3% (-10% and 10%), respectively. The histogram for the S&P 500 monthly returns is slightly skewed left (long left tail) due to more large negative returns than large positive returns whereas the histograms for the other returns are roughly symmetric.

When comparing two or more return distributions, it is useful to use the same bins for each histogram. Figure ?? shows the histograms for Microsoft and S&P 500 returns using the same 15 bins, created with the R code:

```
> msftHist = hist(msftRetS, plot=FALSE, breaks=15)
> par(mfrow=c(2,2))
> hist(msftRetS, main="", col="cornflowerblue")
> hist(msftDailyRetS, main="", col="cornflowerblue",
+       breaks=msftHist$breaks)
> hist(sp500RetS, main="", col="cornflowerblue",
+       breaks=msftHist$breaks)
> hist(sp500DailyRetS, main="", col="cornflowerblue",
+       breaks=msftHist$breaks)
> par(mfrow=c(1,1))
```

## 14 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

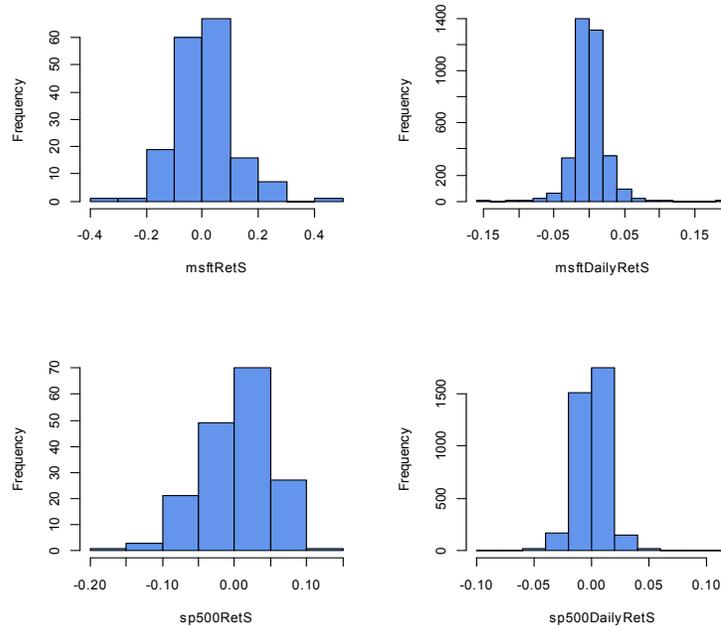


Figure 1.7: Histograms of monthly continuously compounded returns on Microsoft stock and S&P 500 index.

Using the same bins for all histograms allows us to see more clearly that the distribution of monthly returns is more spread out than the distributions for daily returns, and that the distribution of S&P 500 returns is more tightly concentrated around zero than the distribution of Microsoft returns. ■

**Example 8** *Are Microsoft returns normally distributed? A first look.*

The shape of the histogram for Microsoft returns suggests that a normal distribution might be a good candidate for the unknown distribution of Microsoft returns. To investigate this conjecture, we simulate random returns from a normal distribution with mean and standard deviation calibrated to the Microsoft daily and monthly returns using:

```
> set.seed(123)
> gwnDaily = rnorm(length(msftDailyRetS), mean=mean(msftDailyRetS),
```

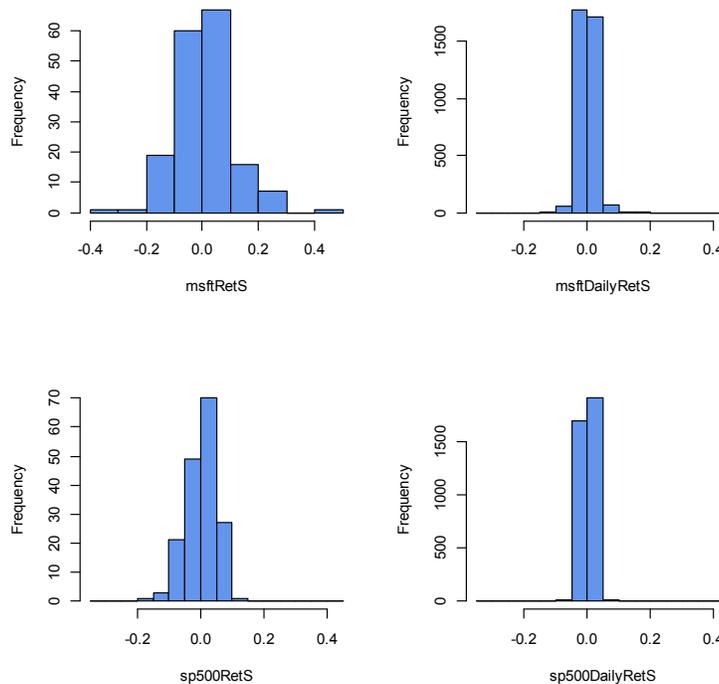


Figure 1.8: Histograms for Microsoft and S&P 500 returns using the same bins.

```

+           sd=sd(msftDailyRetS))
> gwnDaily = zoo(gwnDaily, index(msftDailyRetS))
> gwnMonthly = rnorm(length(msftRetS), mean=mean(msftRetS),
+                   sd=sd(msftRetS))
> gwnMonthly = zoo(gwnMonthly, index(msftRetS))

```

Figure 1.9 shows the Microsoft monthly returns together with the simulated normal returns created using

```

> par(mfrow=c(2,2))
> plot(msftRetS, main="Monthly Returns on MSFT",
+      lwd=2, col="blue", ylim=c(-0.4, 0.4))
> abline(h=0)

```

## 16 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

```
> plot(gwnMonthly, main="Simulated Normal Returns",
+      lwd=2, col="blue", ylim=c(-0.4, 0.4))
> abline(h=0)
> hist(msftRetS, main="", col="cornflowerblue",
+      xlab="returns")
> hist(gwnMonthly, main="", col="cornflowerblue",
+      xlab="returns", breaks=msftHist$breaks)
> par(mfrow=c(1,1))
```

The simulated normal returns shares many of the same features as the Microsoft returns: both fluctuate randomly about zero. However, there are some important differences. In particular, the volatility of Microsoft returns appears to change over time (large before 2003, small between 2003 and 2008, and large again after 2008) whereas the simulated returns has constant volatility. Additionally, the distribution of Microsoft returns has fatter tails (more extreme large and small returns) than the simulated normal returns. Apart from these features, the simulated normal returns look remarkably like the Microsoft monthly returns.

Figure 1.10 shows the Microsoft daily returns together with the simulated normal returns. The daily returns look much less like GWN than the monthly returns. Here the constant volatility of simulated GWN does not match the volatility patterns of the Microsoft daily returns, and the tails of the histogram for the Microsoft returns are “fatter” than the tails of the GWN histogram. ■

### Smoothed histogram

Histograms give a good visual representation of the data distribution. The shape of the histogram, however, depends on the number of bins used. With a small number of bins, the histogram often appears blocky and fine details of the distribution are not revealed. With a large number of bins, the histogram might have many bins with very few observations. The `hist()` function in R smartly chooses the number of bins so that the resulting histogram typically looks good.

The main drawback of the histogram as descriptive statistic for the underlying pdf of the data is that it is discontinuous. If it is believed that the underlying pdf is continuous, it is desirable to have a continuous graphical summary of the pdf. The *smoothed histogram* achieves this goal. Given a

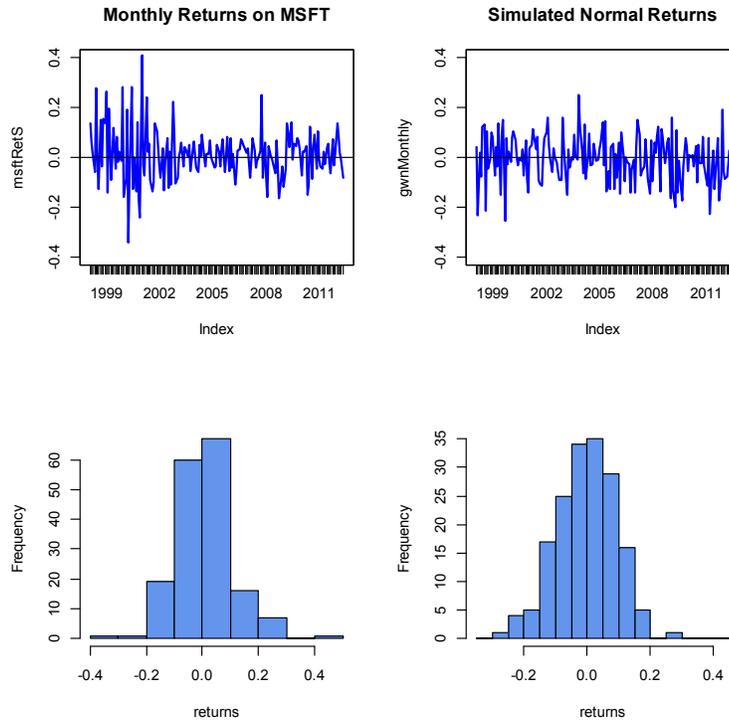


Figure 1.9: Comparison of Microsoft monthly returns with simulated normal returns with the same mean and standard deviation as the Microsoft returns.

sample of data  $\{x_t\}_{t=1}^T$  the R function `density()` computes a smoothed estimate of the underlying pdf at each point  $x$  in the bins of the histogram using the formula

$$\hat{f}_X(x) = \frac{1}{Tb} \sum_{t=1}^T k\left(\frac{x - x_t}{b}\right),$$

where  $k(\cdot)$  is a continuous smoothing function (typically a standard normal distribution) and  $b$  is a bandwidth (or bin-width) parameter that determines the width of the bin around  $x$  in which the smoothing takes place. The resulting pdf estimate  $\hat{f}_X(x)$  is a two-sided weighted average of the histogram values around  $x$ .

**Example 9** *Smoothed histogram for Microsoft monthly returns*

## 18CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

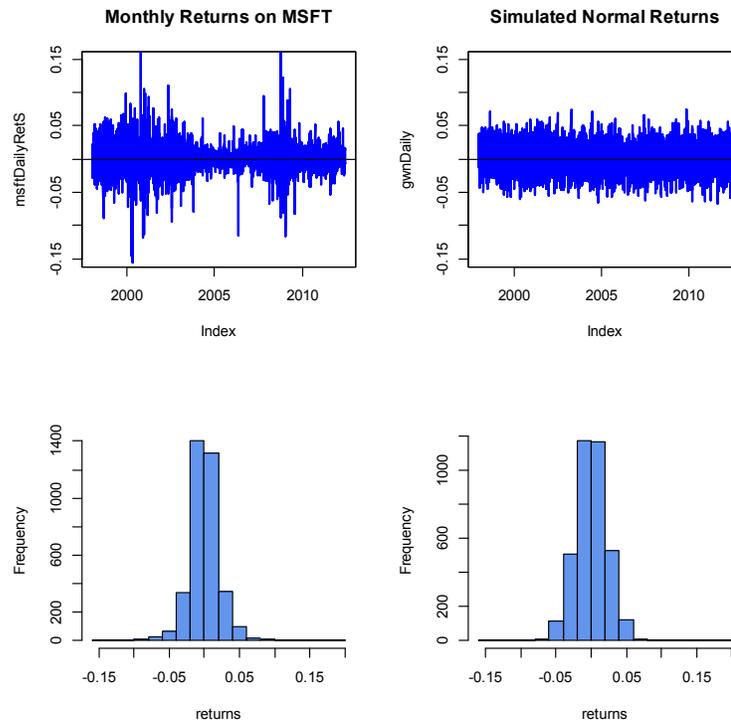


Figure 1.10: Comparison of Microsoft daily returns with simulated normal returns with the same mean and standard deviation as the Microsoft returns.

Figure 1.11 shows the histogram of Microsoft returns overlaid with the smoothed histogram created using

```
> MSFT.density = density(msftRetS)
> hist(msftRetS, main="", xlab="Microsoft Monthly Returns",
+      col="cornflowerblue", probability=T, ylim=c(0,5))
> points(MSFT.density,type="l", col="orange", lwd=2)
```

In Figure 1.11, the histogram is normalized (using the argument `probability=TRUE`), so that its total area is equal to one. The smoothed density estimate transforms the blocky shape of the histogram into a smooth continuous graph.



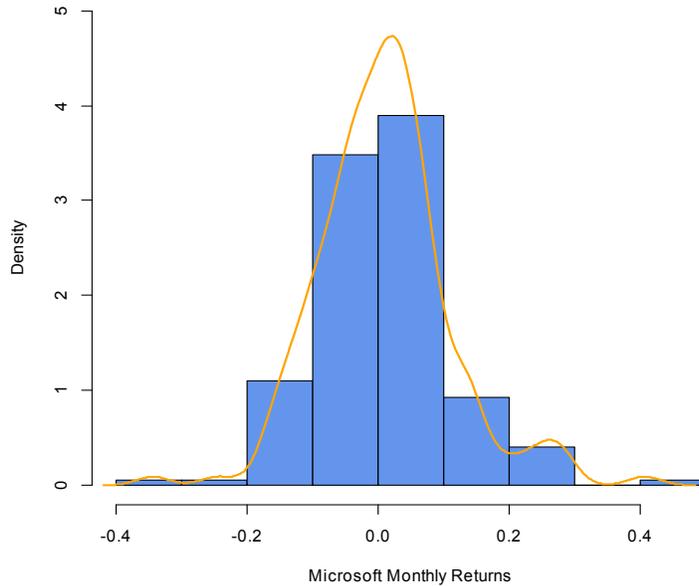


Figure 1.11: Histogram and smoothed density estimate for the monthly returns on Microsoft.

### Empirical CDF

Recall, the CDF of a random variable  $X$  is the function  $F_X(x) = \Pr(X \leq x)$ . The *empirical CDF* of a data sample  $\{x_t\}_{t=1}^T$  is the function that counts the fraction of observations less than or equal to  $x$  :

$$\begin{aligned}\hat{F}_X(x) &= \frac{1}{T}(\#x_i \leq x) \\ &= \frac{\text{number of } x_i \text{ values } \leq x}{\text{sample size}}\end{aligned}$$

**Example 10** *Empirical CDF for daily and monthly returns on Microsoft*

To be completed

**Empirical quantiles/percentiles**

Recall, for  $\alpha \in (0, 1)$  the  $\alpha \times 100\%$  quantile of the distribution of a continuous random variable  $X$  with CDF  $F_X$  is the point  $q_\alpha^X$  such that  $F_X(q_\alpha^X) = \Pr(X \leq q_\alpha^X) = \alpha$ . Accordingly, the  $\alpha \times 100\%$  *empirical quantile* (or  $100 \times \alpha^{\text{th}}$  *percentile*) of a data sample  $\{x_t\}_{t=1}^T$  is the data value  $\hat{q}_\alpha$  such that  $\alpha \cdot 100\%$  of the data are less than or equal to  $\hat{q}_\alpha$ . Empirical quantiles can be easily determined by ordering the data from smallest to largest giving the ordered sample (also known as *order statistics*)

$$x_{(1)} < x_{(2)} < \cdots < x_{(T)}.$$

The empirical quantile  $\hat{q}_\alpha$  is the order statistic closest to  $\alpha \times T$ .<sup>5</sup>

The *empirical quartiles* are the empirical quantiles for  $\alpha = 0.25, 0.5$  and  $0.75$ , respectively. The second empirical quartile  $\hat{q}_{.50}$  is called the *sample median* and is the data point such that half of the data is less than or equal to its value. The *interquartile range* (IQR) is the difference between the 3rd and 1st quartile

$$\text{IQR} = \hat{q}_{.75} - \hat{q}_{.25},$$

and shows the size of the middle of the data distribution.

**Example 11** *Empirical quantiles of the Microsoft and S&P 500 monthly returns*

The R function `quantile()` computes empirical quantiles for a single data series. By default, `quantile()` returns the empirical quartiles as well as the minimum and maximum values:

```
> quantile(msftRetS)
      0%      25%      50%      75%      100%
-0.343450 -0.048561  0.008377  0.057303  0.407878
> quantile(sp500RetS)
      0%      25%      50%      75%      100%
-0.16942 -0.02094  0.00809  0.03532  0.10772
```

---

<sup>5</sup>There is no unique way to determine the empirical quantile from a sample of size  $N$  for all values of  $\alpha$ . The R function `quantile()` can compute empirical quantile using one of seven different definitions.

The left (right) quantiles of the Microsoft monthly returns are smaller (larger) than the respective quantiles for the S&P 500 index.

To compute quantiles for a specified  $\alpha$  use the `probs` argument. For example, to compute the 1% and 5% quantiles of the monthly returns use

```
> quantile(msftRetS,probs=c(0.01,0.05))
      1%      5%
-0.1892 -0.1371
> quantile(sp500RetS,probs=c(0.01,0.05))
      1%      5%
-0.12040 -0.08184
```

Here we see that 1% of the Microsoft cc returns are less than -18.9% and 5% of the returns are less than -13.7%, respectively. For the S&P 500 returns, these values are -12.0% and -8.2%, respectively.

To compute the median and IQR values for monthly returns use the R functions `median()` and `IQR()`, respectively

```
> apply(msftSp500RetS, 2, median)
      MSFT      SP500
0.008377 0.008090
> apply(msftSp500RetS, 2, IQR)
      MSFT      SP500
0.10586 0.05626
```

The median returns are similar (about 0.8% per month) but the IQR for Microsoft is about twice as large as the IQR for the S&P 500 index. ■

### Historical/Empirical VaR

Recall, the  $\alpha \times 100\%$  value-at-risk (VaR) of an investment of  $\$W$  is  $\text{VaR}_\alpha = \$W \times q_\alpha^R$ , where  $q_\alpha^R$  is the  $\alpha \times 100\%$  quantile of the probability distribution of the investment simple rate of return  $R$ . The  $\alpha \times 100\%$  *historical VaR* (sometimes called *empirical VaR* or *Historical Simulation VaR*) of an investment of  $\$W$  is defined as

$$\text{VaR}_\alpha^{HS} = \$W \times \hat{q}_\alpha^R.$$

where  $\hat{q}_\alpha^R$  is the empirical  $\alpha$  quantile of a sample of simple returns  $\{R_t\}_{t=1}^T$ . For a sample of continuously compounded returns  $\{r_t\}_{t=1}^T$  with empirical  $\alpha$  quantile  $\hat{q}_\alpha^r$ ,

$$\text{VaR}_\alpha^{HS} = \$W \times (\exp(\hat{q}_\alpha^r) - 1).$$

## 22 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

Historical VaR is based on the distribution of the observed returns and not on any assumed distribution for returns (e.g., the normal distribution).

**Example 12** *Using empirical quantiles to compute historical Value-at-Risk*

Consider investing  $W = \$100,000$  in Microsoft and the S&P 500 over a month. The 1% and 5% historical VaR values for these investments based on the historical samples of monthly returns are:

```
> W = 100000
> msftQuantiles = quantile(msftRetS, probs=c(0.01, 0.05))
> sp500Quantiles = quantile(sp500RetS, probs=c(0.01, 0.05))
> msftVaR = W*msftQuantiles
> sp500VaR = W*sp500Quantiles
> msftVaR
      1%      5%
-18923 -13711
> sp500VaR
      1%      5%
-12040  -8184
```

Based on the empirical distribution of the monthly returns, a \$100,000 monthly investment in Microsoft will lose \$13,711 or more with 5% probability and will lose \$18,923 or more with 1% probability. The corresponding values for the S&P 500 are \$8,184 and \$12,040, respectively. The historical VaR values for the S&P 500 are considerable smaller than those for Microsoft. In this sense, investing in Microsoft is a riskier than investing in the S&P 500 index.

### 1.1.4 QQ-plots

Often it is of interest to see if a given data sample could be viewed as a random sample from a specified probability distribution. One easy and effective way to do this is to compare the empirical quantiles of a data sample to those from a reference probability distribution. If the quantiles match up, then this provides strong evidence that the reference distribution is appropriate for describing the distribution of the observed data. If the quantiles do not match up, then the observed differences between the empirical quantiles and the reference quantiles can be used to determine a more appropriate reference

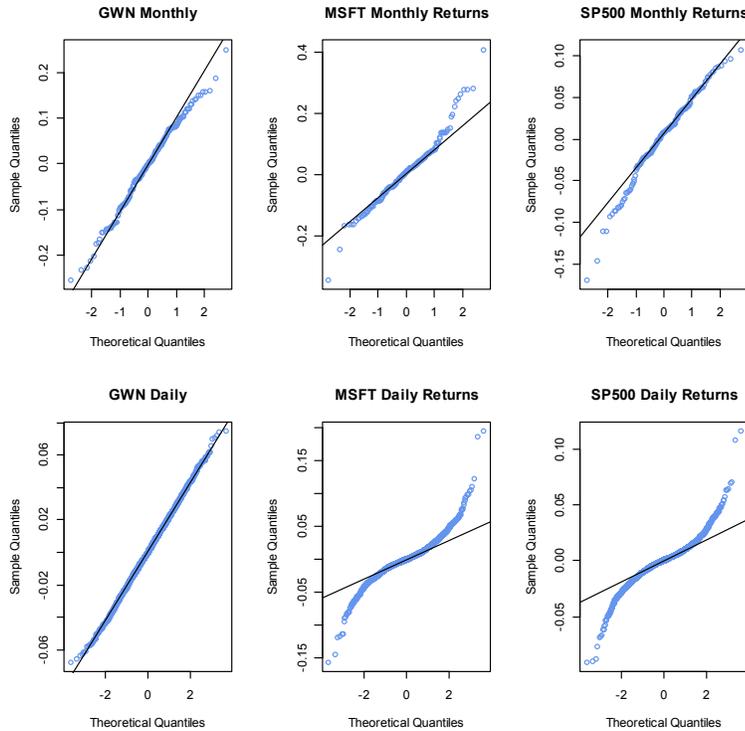


Figure 1.12: Normal QQ-plots for GWN, Microsoft returns and S&P 500 returns.

distribution. It is common to use the normal distribution as the reference distribution, but any distribution can, in principle be used.

The quantile-quantile plot (QQ-plot) gives a graphical comparison of the empirical quantiles of a data sample to those from a specified reference distribution. The QQ-plot is an  $xy$ -plot with the reference distribution quantiles on the  $x$ -axis and the empirical quantiles on the  $y$ -axis. If the quantiles exactly match up then the QQ-plot is a straight line. If the quantiles do not match up, then the shape of the QQ-plot indicates which features of the data are not captured by the reference distribution.

**Example 13** *Normal QQ-plots for GWN, Microsoft and S&P 500 returns*

The R function `qqnorm()` creates a QQ-plot for a data sample using the normal distribution as the reference distribution. Figure 1.12 shows nor-

## 24 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

mal QQ-plots for the simulated GWN data, Microsoft returns and S&P 500 returns created using

```
> par(mfrow=c(2,3))
> qqnorm(gwnMonthly, main="GWN Monthly", col="cornflowerblue")
> qqline(gwnMonthly)
> qqnorm(msftRetS, main="MSFT Monthly Returns", col="cornflowerblue")
> qqline(msftRetS)
> qqnorm(sp500RetS, main="SP500 Monthly Returns", col="cornflowerblue")
> qqline(sp500RetS)
> qqnorm(gwnDaily, main="GWN Daily", col="cornflowerblue")
> qqline(gwnDaily)
> qqnorm(msftDailyRetS, main="MSFT Daily Returns", col="cornflowerblue")
> qqline(msftDailyRetS)
> qqnorm(sp500DailyRetS, main="SP500 Daily Returns", col="cornflowerblue")
> qqline(sp500DailyRetS)
> par(mfrow=c(1,1))
```

The normal QQ-plot for the simulated GWN data is very close to a straight line, as it should be since the data are simulated from a normal distribution. The `qqline()` function draws a straight line through the points to help determine if the quantiles match up. The normal QQ-plots for the Microsoft and S&P 500 monthly returns are linear in the middle of the distribution but deviate from linearity in the tails of the distribution. In the normal QQ-plot for Microsoft monthly returns, the theoretical normal quantiles on the x-axis are too small in both the left and right tails because the points fall below the straight line in the left tail and fall above the straight line in the right tail. Hence, the normal distribution does not match the empirical distribution of Microsoft monthly returns in the extreme tails of the distribution. In other words, the Microsoft returns have fatter tails than the normal distribution. For the S&P 500 monthly returns, the theoretical normal quantiles are too small only for the left tail of the empirical distribution of returns (points fall below the straight line in the left tail only). This reflects the long left tail (negative skewness) of the empirical distribution of S&P 500 returns. The normal QQ-plots for the daily returns on Microsoft and the S&P 500 index both exhibit a pronounced tilted S-shape with extreme departures from linearity in the left and right tails of the distributions. The daily returns clearly have much fatter tails than the normal distribution. ■

**Example 14** *Student's t QQ-plot for Microsoft returns*

The function `qqPlot()` from the R package `car` can be used to create a QQ-plot against any reference distribution that has a corresponding quantile function implemented in R. For example, a QQ-plot for the Microsoft returns using a Student's t reference distribution with 5 degrees of freedom can be created using

```
> library(car)
> qqPlot(coredata(msftRetS), distribution="t", df=5,
+        ylab="MSFT quantiles", envelope=FALSE)
```

In the function `qqPlot()`, the argument `distribution="t"` specifies that the quantiles are to be computed using the R function `qt()`.<sup>6</sup> Figure 1.13 shows the resulting graph. Here, with a reference distribution with fatter tails than the normal distribution the QQ-plot for Microsoft returns is closer to a straight line. This indicates that the Student's t distribution with 5 degrees of freedom is a better reference distribution for Microsoft returns than the normal distribution. ■

### 1.1.5 Shape Characteristics of the Empirical Distribution

Recall, for a random variable  $X$  the measures of center, spread, asymmetry and tail thickness of the pdf are:

$$\begin{aligned} \text{center: } & \mu_X = E[X], \\ \text{spread : } & \sigma_X^2 = \text{var}(X) = E[(X - \mu_X)^2], \\ \text{spread : } & \sigma_X = \sqrt{\text{var}(X)} \\ \text{asymmetry : } & \text{skew}_X = E[(X - \mu_X)^3]/\sigma^3, \\ \text{tail thickness : } & \text{kurt}_X = E[(X - \mu_X)^4]/\sigma^4. \end{aligned}$$

The corresponding shape measures for the empirical distribution (e.g., as measured by the histogram) of a data sample  $\{x_t\}_{t=1}^T$  are the sample statis-

---

<sup>6</sup>The `coredata()` function is used to extract the data as a “numeric” object because `qqPlot()` does not work correctly with “zoo” objects.

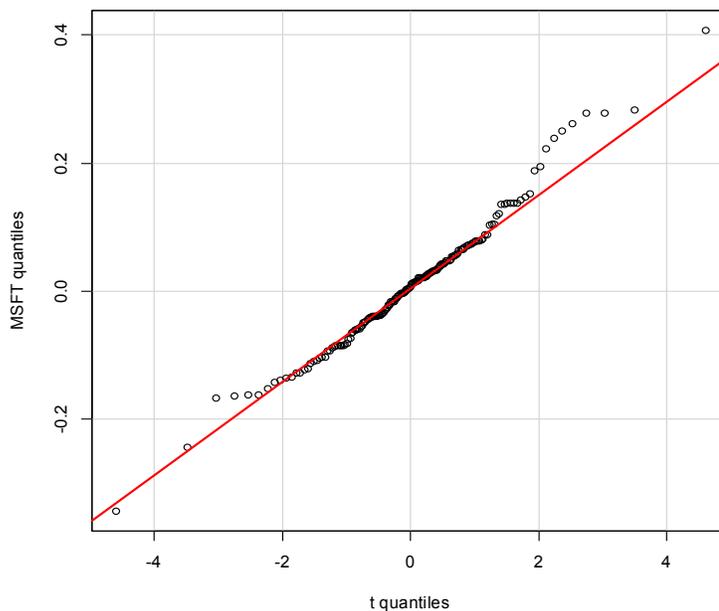


Figure 1.13: QQ-plot of Microsoft returns using Student's  $t$  distribution with 5 degrees of freedom as the reference distribution.

tics:<sup>7</sup>

$$\hat{\mu}_x = \bar{x} = \frac{1}{T} \sum_{t=1}^T x_t, \quad (1.1)$$

$$\hat{\sigma}_x^2 = s_x^2 = \frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})^2, \quad (1.2)$$

$$\hat{\sigma}_x = \sqrt{\hat{\sigma}_x^2} \quad (1.3)$$

$$\widehat{\text{skew}}_x = \frac{\frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})^3}{s_x^3}, \quad (1.4)$$

$$\widehat{\text{kurt}}_x = \frac{\frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})^4}{s_x^4}. \quad (1.5)$$

<sup>7</sup>Values with hats  $\hat{\cdot}$  denote sample estimates of the corresponding population quantity. For example, the sample mean  $\hat{\mu}_x$  is the sample estimate of the population expected value  $\mu_X$ .

The sample mean,  $\hat{\mu}_X$ , measures the center of the histogram; the sample standard deviation,  $\hat{\sigma}_x$ , measures the spread of the data about the mean in the same units as the data; the sample skewness,  $\widehat{\text{skew}}_x$ , measures the asymmetry of the histogram; the sample kurtosis,  $\widehat{\text{kurt}}_x$ , measures the tail-thickness of the histogram. The sample excess kurtosis, defined as the sample kurtosis minus 3

$$\widehat{\text{ekurt}}_x = \widehat{\text{kurt}}_x - 3, \quad (1.6)$$

measures the tail thickness of the data sample relative to that of a normal distribution.

Notice that the divisor in (1.2)-(1.5) is  $T - 1$  and not  $T$ . This is called a *degrees-of-freedom* correction. In computing the sample variance, skewness and kurtosis, one degree-of-freedom in the sample is used up in the computation of the sample mean so that there are effectively only  $T - 1$  observations available to compute the statistics.<sup>8</sup>

**Example 15** *Sample shape statistics for the returns on Microsoft and S&P 500*

The R functions for computing (1.1) - (1.5) are `mean()`, `var()` and `sd()`, respectively. There are no functions for computing (1.4) and (1.5) in base R. The functions `skewness()` and `kurtosis()` in the **PerformanceAnalytics** package compute (1.4) and the sample excess kurtosis (1.6), respectively.<sup>9</sup> The sample statistics for the Microsoft and S&P 500 monthly returns are:

```
> statsMat = rbind(apply(msftSp500RetS, 2, mean),
+                   apply(msftSp500RetS, 2, var),
+                   apply(msftSp500RetS, 2, sd),
+                   apply(msftSp500RetS, 2, skewness),
+                   apply(msftSp500RetS, 2, kurtosis))
> rownames(statsMat) = c("Mean", "Variance", "Std Dev",
+                       "Skewness", "Excess Kurtosis")
> round(statsMat, digits=4)
      MSFT   SP500
```

---

<sup>8</sup>If there is only one observation in the sample then it is impossible to create a measure of spread in the sample. You need at least two observations to measure deviations from the sample average. Hence the effective sample size for computing the sample variance is  $T - 1$ .

<sup>9</sup>Similar functions are available in the **moments** package.

## 28 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

Mean	0.0092	0.0028
Variance	0.0103	0.0023
Std Dev	0.1015	0.0478
Skewness	0.4853	-0.5631
Excess Kurtosis	1.9150	0.6545

The mean and standard deviation for Microsoft monthly returns are 0.9% and 10%, respectively. Annualized, these values are approximately 10.08% ( $.009 \times 12$ ) and 34.6% ( $.10 \times \sqrt{12}$ ), respectively. The corresponding monthly and annualized values for S&P 500 returns are .3% and 4.7% and 3.6% and 16.3%, respectively. Microsoft has a higher mean and volatility than S&P 500. The lower volatility for the S&P 500 reflects risk reduction due to diversification. The sample skewness for Microsoft, 0.485, is slightly positive and reflects the approximate symmetry in the histogram in Figure 1.7. The skewness for S&P 500, however, is moderately negative at -0.563 which reflects the somewhat long left tail of the histogram in Figure 1.7. The sample excess kurtosis values for Microsoft and S&P 500 are 1.915 and 0.654, respectively, and indicate that the tails of the histograms are slightly fatter than the tails of a normal distribution.

The sample statistics for the daily returns are

```
> statsMatDaily = rbind(apply(msftSp500DailyRetS, 2, mean),
+                           apply(msftSp500DailyRetS, 2, var),
+                           apply(msftSp500DailyRetS, 2, sd),
+                           apply(msftSp500DailyRetS, 2, skewness),
+                           apply(msftSp500DailyRetS, 2, kurtosis))
> rownames(statsMatDaily) = rownames(statsMat)
> round(statsMatDaily, digits=4)
              MSFT  SP500
Mean          0.0005 0.0002
Variance      0.0005 0.0002
Std Dev       0.0217 0.0135
Skewness      0.2485 0.0012
Excess Kurtosis 7.3824 6.9971
```

For a daily horizon, the sample mean for both series is zero (to three decimals). As with the monthly returns, the sample standard deviation for Microsoft, 2.2%, is about twice as big as the sample standard deviation for

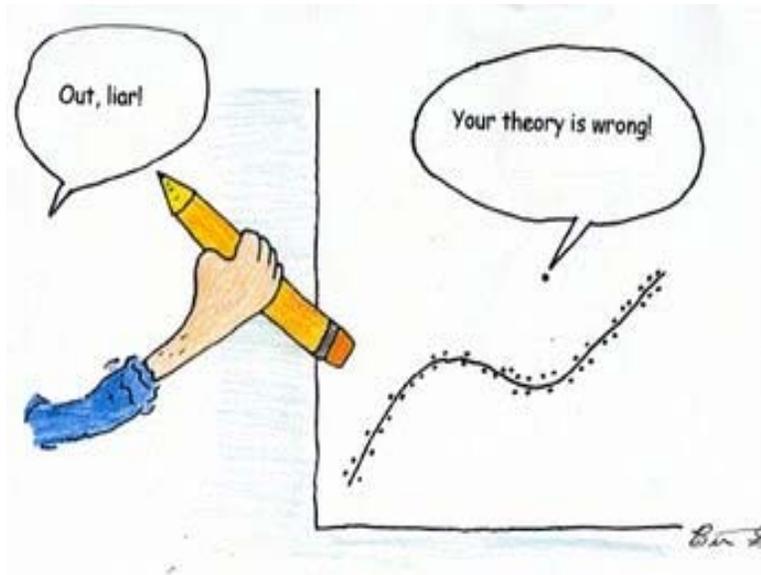


Figure 1.14: Illustration of an “outlier” in a data sample.

the S&P 500 index, 1.4%. Neither daily return series exhibits much skewness, but the excess kurtosis values are much larger than the corresponding monthly values indicating that the daily returns have much fatter tails and are more non-normally distributed than the monthly return series.

The daily simple returns are very close to the daily cc returns so the square-root-of-time rule can be applied to the sample mean and standard deviation of the simple returns:

```
> 12*statsMatDaily["Mean", ]
      MSFT    SP500
0.005597 0.002069
> sqrt(12)*statsMatDaily["Std Dev", ]
      MSFT    SP500
0.07518 0.04670
```

Comparing these values to the corresponding monthly sample statistics shows that the square-root-of-time-rule underestimates both the monthly mean and standard deviation. ■

Statistic	GWN	GWN with Outlier	% $\Delta$
Mean	-0.0054	-0.0072	32%
Variance	0.0086	0.0101	16%
Std. Deviation	0.0930	0.1003	8%
Skewness	-0.2101	-1.0034	377%
Kurtosis	2.7635	7.1530	159%
Median	-0.0002	-0.0002	0%
IQR	0.1390	0.1390	0%

Table 1.1: Sample statistics for GWN with and without outlier.

### 1.1.6 Outliers

Figure 1.14 nicely illustrates the concept of an outlier in a data sample. All of the points are following a nice systematic relationship except one - the “outlier”. Outliers can be thought of in two ways. First, an outlier can be the result of a data entry error. In this view, the outlier is not a valid observation and should be removed from the data sample. Second, an outlier can be a valid data point whose behavior is seemingly unlike the other data points. In this view, the outlier provides important information and should not be removed from data sample. For financial market data, outliers are typically extremely large or small values that could be the result of a data entry error (e.g. price entered as 10 instead of 100) or a valid outcome associated with some unexpected bad or good news. Outliers are problematic for data analysis because they can greatly influence the value of certain sample statistics.

#### **Example 16** *Effect of outliers on sample statistics*

To illustrate the impact of outliers on sample statistics, the simulated monthly GWN data is polluted by a single large negative outlier:

```
> gwnMonthlyOutlier = gwnMonthly
> gwnMonthlyOutlier[20] = -sd(gwnMonthly)*6
```

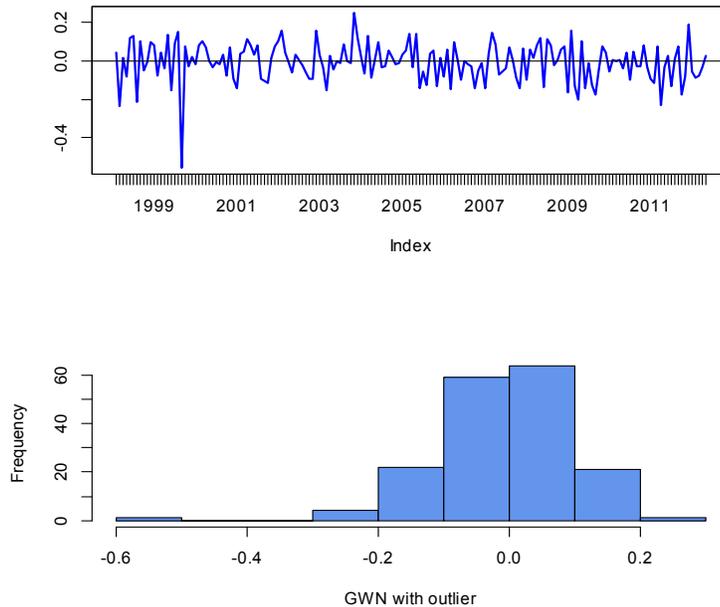


Figure 1.15: Monthly GWN data polluted by a single outlier.

Figure 1.15 shows the resulting data. Visually, the outlier is much smaller than a typical negative observation and creates a pronounced asymmetry in the histogram. Table 1.1 compares the sample statistics (1.1) - (1.5) of the unpolluted and polluted data. All of the sample statistics are influenced by the single outlier with the skewness and kurtosis being influenced the most. The mean decreases by 32% and the variance and standard deviation increase by 16% and 8%, respectively. The skewness changes by 377% and the kurtosis by 159%. Table 1.1 also shows the median and the IQR, which are quantile-based statistics for the center and spread, respectively. Notice that these statistics are unaffected by the outlier. ■

The previous example shows that the common sample statistics (1.1) - (1.5) based on the sample average and deviations from the sample average can be greatly influenced by a single outlier, whereas quantile-based sample statistics are not. Sample statistics that are not greatly influenced by a single outlier are called (outlier) *robust statistics*.

### Defining Outliers

A commonly used rule-of-thumb defines an outlier as an observation that is beyond the sample mean plus or minus three times the sample standard deviation. The intuition for this rule comes from the fact that if  $X \sim N(\mu, \sigma^2)$  then  $\Pr(\mu - 3\sigma \leq X \leq \mu + 3\sigma) \approx 0.99$ . While the intuition for this definition is clear, the practical implementation of the rule using the sample mean,  $\hat{\mu}$ , and sample standard deviation,  $\hat{\sigma}$ , is problematic because these statistics can be heavily influenced by the presence of one or more outliers (as illustrated in the previous sub-section). In particular, both  $\hat{\mu}$  and  $\hat{\sigma}$  become larger (in absolute value) when there are outliers and this may cause the rule-of-thumb rule to miss identifying an outlier or to identify a non-outlier as an outlier. That is, the rule-of-thumb outlier detection rule is not robust to outliers!

A natural fix to the above outlier detection rule-of-thumb is to use outlier robust statistics, such as the median and the IQR, instead of the mean and standard deviation. To this end, a commonly used definition of a moderate outlier in the right tail of the distribution (value above the median) is a data point  $x$  such that

$$\hat{q}_{.75} + 1.5 \cdot IQR < x < \hat{q}_{.75} + 3 \cdot IQR. \quad (1.7)$$

To understand this rule, recall that the IQR is where the middle 50% of the data lies. If the data were normally distributed then  $\hat{q}_{.75} \approx \hat{\mu} + 0.67 \times \hat{\sigma}$  and  $IQR \approx 1.34 \times \hat{\sigma}$ . Then (1.7) is approximately

$$\hat{\mu} + 2.67 \times \hat{\sigma} < x < \hat{\mu} + 4.67 \times \hat{\sigma}$$

Similarly, a moderate outlier in the left tail of the distribution (value below the median) is a data point such that

$$\hat{q}_{.25} - 3 \cdot IQR < x < \hat{q}_{.25} - 1.5 \cdot IQR.$$

Extreme outliers are those observations  $x$  even further out in the tails of the distribution and are defined using

$$\begin{aligned} x &> \hat{q}_{.75} + 3 \cdot IQR, \\ x &< \hat{q}_{.25} - 3 \cdot IQR. \end{aligned}$$

For example, with normally distributed data an extreme outlier would be an observation that is more than 4.67 standard deviations above the mean.

### 1.1.7 Box Plots

A boxplot of a data series describes the distribution using outlier robust statistics. The basic boxplot is illustrated in Figure. The middle of the distribution is shown by a rectangular box whose top and bottom sides are the first and third quartiles,  $\hat{q}_{.25}$  and  $\hat{q}_{.75}$ , respectively. The line roughly in the middle of the box is the sample median,  $\hat{q}_{.5}$ . The top whisker is a horizontal line that represents either the largest data value or  $\hat{q}_{.75} + 1.5 \cdot IQR$ , whichever is smaller. Similarly, the bottom whisker is located at either the smallest data value or  $\hat{q}_{.25} - 3 \cdot IQR$ , whichever is larger.

#### Example 17 *Boxplots of return distributions*

Box plots are a great tool for visualizing the return distribution of one or more assets. Figure 1.16 shows the boxplots of the monthly continuously compounded and simple returns on Microsoft and the S&P500 index computed using<sup>10</sup>

```
> boxplot(coredata(msftRetS), coredata(msftRetC),
+         coredata(sp500RetS), coredata(sp500RetC),
+         names=c("msftRetS", "msftRetC",
+                 "sp500RetS", "sp500RetC"),
+         col="cornflowerblue")
```

The Microsoft returns have both positive and negative outliers whereas the S&P 500 index returns only have negative outliers. The boxplots also clearly show that the distributions of the continuously compounded and simple returns are very similar except for the extreme tails of the distributions. ■

## 1.2 Time Series Descriptive Statistics

In this section, we consider graphical and numerical descriptive statistics for summarizing the linear time dependence in a time series.

---

<sup>10</sup>The `coredata()` function is used to extract the data from the `zoo` object because `boxplot()` doesn't like `zoo` objects.

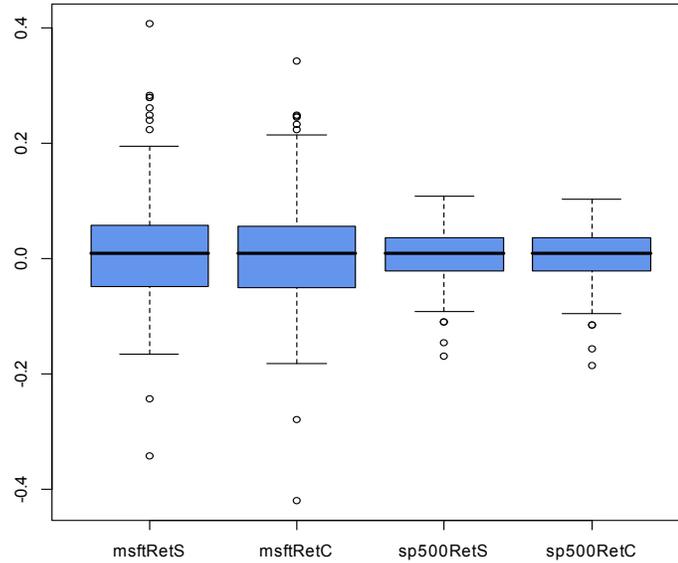


Figure 1.16: Boxplots of the continuously compounded and simple monthly returns on Microsoft stock and the S&P 500 index.

### 1.2.1 Sample Autocovariances and Autocorrelations

For a covariance stationary time series process  $\{X_t\}$ , the autocovariances  $\gamma_j = \text{cov}(X_t, X_{t-j})$  and autocorrelations  $\rho_j = \text{cov}(X_t, X_{t-j}) / \sqrt{\text{var}(X_t)\text{var}(X_{t-j})} = \gamma_j / \sigma^2$  describe the linear time dependences in the process. For a sample of data  $\{x_t\}_{t=1}^T$ , the linear time dependences are captured the sample autocovariances and autocorrelations:

$$\hat{\gamma}_j = \frac{1}{T-j} \sum_{t=j+1}^T (x_t - \bar{x})(x_{t-j} - \bar{x}), \quad j = 1, 2, \dots, T - j + 1,$$

$$\hat{\rho}_j = \frac{\hat{\gamma}_j}{\hat{\sigma}^2}, \quad j = 1, 2, \dots, T - j + 1,$$

where  $\hat{\sigma}^2$  is the sample variance (1.2). The sample autocorrelation function (SACF) is a plot of  $\hat{\rho}_j$  vs.  $j$ , and gives a graphical view of the linear time

dependences in the observed data.

**Example 18** *SACF for the Microsoft and S&P 500 returns.*

The R function `acf()` computes and plots the sample autocorrelations  $\hat{\rho}_j$  of a time series for a specified number of lags. For example, to compute  $\hat{\rho}_j$  for  $j = 1, \dots, 5$  for the monthly and daily returns on Microsoft use

```
> acf(coredata(msftDailyRetS), lag.max=5, plot=FALSE)
```

Autocorrelations of series 'coredata(msftDailyRetS)', by lag

```
      0      1      2      3      4      5
1.000 -0.044 -0.018 -0.010 -0.042  0.018
```

```
> acf(coredata(msftRetS), lag.max=5, plot=FALSE)
```

Autocorrelations of series 'coredata(msftRetS)', by lag

```
      0      1      2      3      4      5
1.000 -0.193 -0.114  0.193 -0.140 -0.023
```

```
> acf(coredata(msftDailyRetS), lag.max=5, plot=FALSE)
```

Autocorrelations of series 'coredata(msftDailyRetS)', by lag

```
      0      1      2      3      4      5
1.000 -0.044 -0.018 -0.010 -0.042  0.018
```

None of these sample autocorrelations are very big and indicate lack of linear time dependence in the monthly and daily returns.

Sample autocorrelations can also be visualized using the function `acf()`. Figure 1.17 shows the SACF for the monthly and daily returns on Microsoft and the S&P 500 created using

```
> par(mfrow=c(2,2))
> acf(coredata(msftRetS), main="msftRetS", lwd=2)
> acf(coredata(sp500RetS), main="sp500RetS", lwd=2)
> acf(coredata(msftDailyRetS), main="msftDailyRetS", lwd=2)
> acf(coredata(sp500DailyRetS), main="sp500DailyRetS", lwd=2)
> par(mfrow=c(1,1))
```

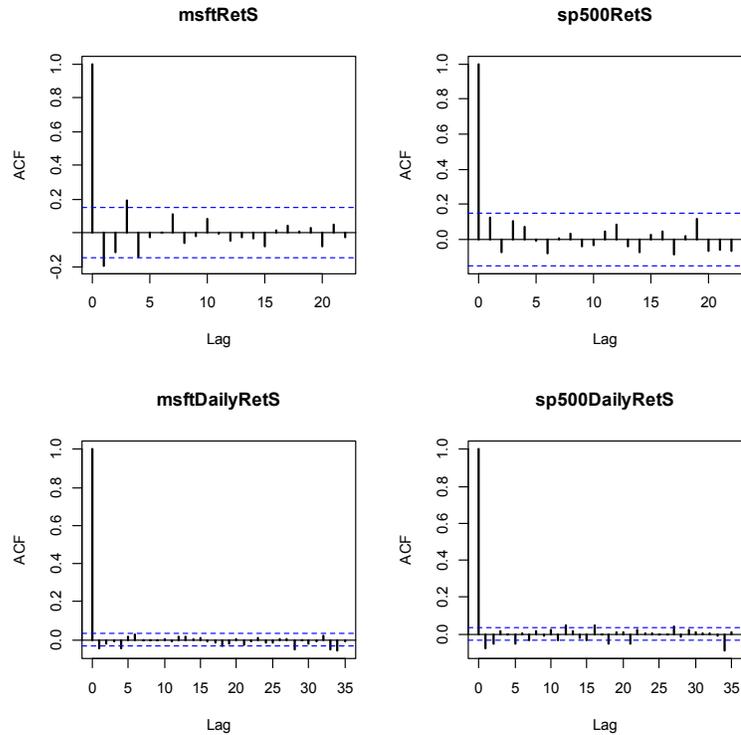


Figure 1.17: SACFs for the monthly and daily returns on Microsoft and the S&P 500 index.

None of the series show strong evidence for linear time dependence. The monthly returns have slightly larger sample autocorrelations at low lags than the daily returns. The dotted lines on the SACF plots are 5% critical values for testing the null hypothesis that the true autocorrelations  $\rho_j$  are equal to zero. If  $\hat{\rho}_j$  crosses the dotted line then the null hypothesis that  $\rho_j = 0$  can be rejected at the 5% significance level.

## 1.2.2 Time Dependence in Volatility

The daily and monthly returns of Microsoft and the S&P 500 index do not exhibit much evidence for linear time dependence. Their returns are essentially uncorrelated over time. However, the lack of autocorrelation does not mean that returns are independent over time. There can be nonlinear time

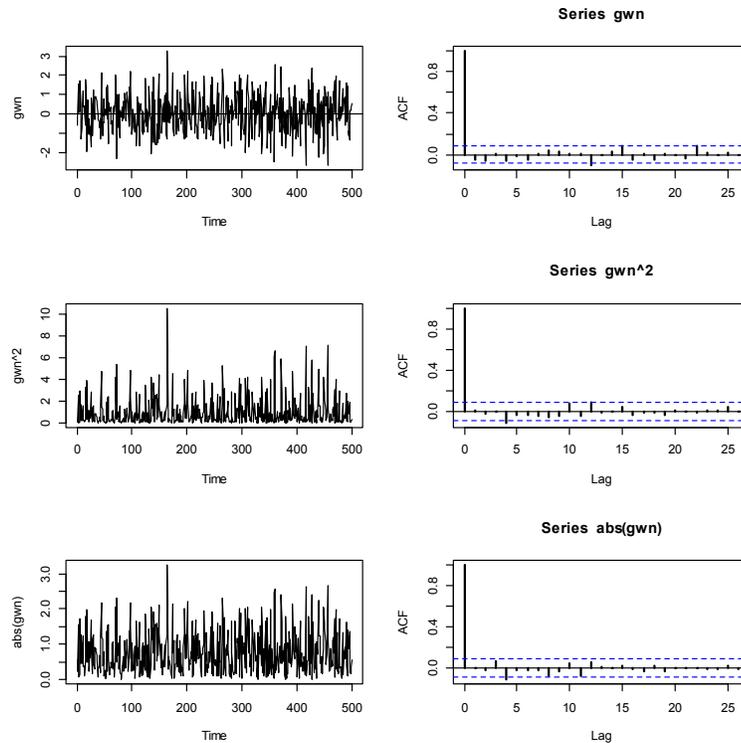


Figure 1.18:

dependence. Indeed, there is strong evidence for a particular type of nonlinear time dependence in the daily returns of Microsoft and the S&P 500 index that is related to time varying volatility. This nonlinear time dependence, however, is less pronounced in the monthly returns.

**Example 19** *GWN and Time Independence*

Let  $y_t \sim \text{GWN}(0, \sigma^2)$ . Then  $y_t$  is, by construction, independent over time. This implies that any function of  $y_t$  is also independent over time. To illustrate, Figure 1.18 shows  $y_t$ ,  $y_t^2$  and  $|y_t|$  together with their SACFs where  $y_t \sim \text{GWN}(0, 1)$  for  $t = 1, \dots, 500$ . Notice that all of the sample autocorrelations are essentially zero which confirms the lack of linear time dependence in  $y_t$ ,  $y_t^2$  and  $|y_t|$ . ■

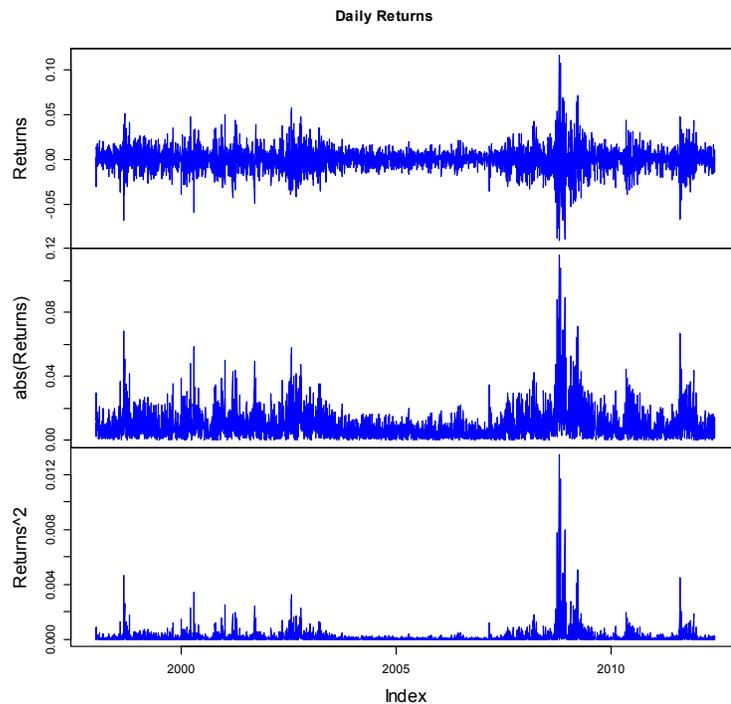


Figure 1.19: Daily returns, absolute returns, and squared returns on the S&P 500 index.

### Nonlinear Time Dependence in Asset Returns

Figures 1.5 and 1.2 show that the daily and monthly returns on Microsoft and the S&P 500 index appear to exhibit periods of volatility clustering. That is, high periods of volatility tend to be followed by periods of high volatility and periods of low volatility appear to be followed by periods of low volatility. In other words, the volatility of the returns appears to exhibit some time dependence. This time dependence in volatility causes linear time dependence in the absolute value and squared value of returns.

#### **Example 20** *Nonlinear time dependence in the S&P 500 returns*

To illustrate nonlinear time dependence in asset returns, Figure 1.19 shows the daily returns, absolute returns and squared returns for the S&P 500 index

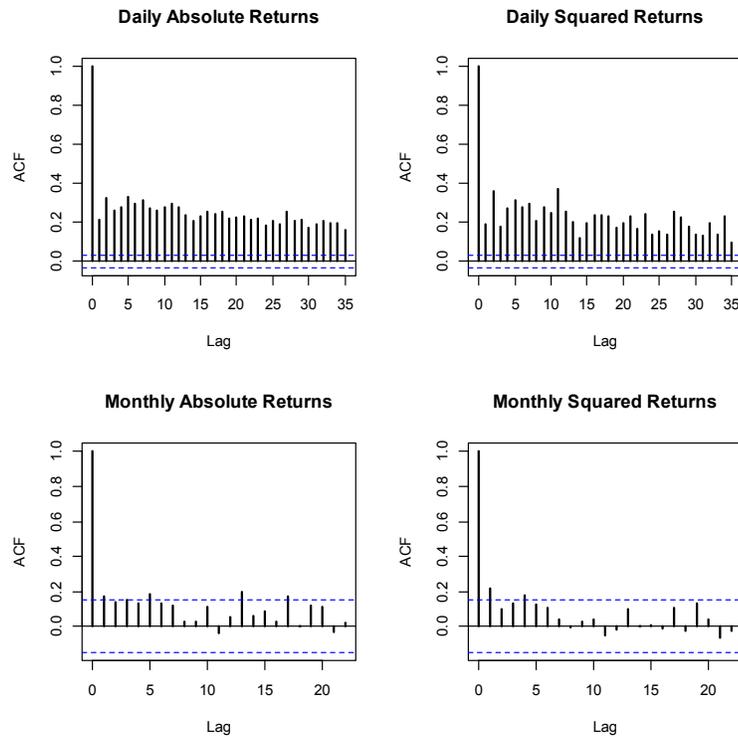


Figure 1.20: SACFs of daily and monthly absolute and squared returns on the S&P 500 index.

and Figure shows the SACFs for these series. Notice how the absolute and squared returns are large (small) when the daily returns are more (less) volatile. Figure 1.20 shows the SACFs for the absolute and squared daily and monthly returns. There is clear evidence of time dependence in the daily absolute and squared returns while there is some evidence of time dependence in the monthly absolute and squared returns. Since the absolute and squared daily returns reflect the volatility of the daily returns, the SACFs show a strong positive linear time dependence in daily return volatility.

## 1.3 Bivariate Descriptive Statistics

In this section, we consider graphical and numerical descriptive statistics for summarizing two or more data series.

### 1.3.1 Scatterplots

The contemporaneous dependence properties between two data series  $\{x_t\}_{t=1}^T$  and  $\{y_t\}_{t=1}^T$  can be displayed graphically in a scatterplot, which is simply an xy-plot of the bivariate data.

**Example 21** *Scatterplot of Microsoft and S&P 500 returns*

Figure 1.21 shows the scatterplot between the Microsoft and S&P 500 returns created using

```
> plot(sp500RetS,msftRetS,
+      main="Monthly cc returns on MSFT and SP500",
+      xlab="S&P500 returns", ylab="MSFT returns",
+      lwd=2, pch=16, cex=1.25, col="blue")
> abline(v=mean(sp500RetS))
> abline(h=mean(msftRetS))
```

The S&P 500 returns are put on the x-axis and the Microsoft returns on the y-axis because the “market”, as proxied by the S&P 500, is often thought as an independent variable driving individual asset returns. The upward sloping orientation of the scatterplot indicates a positive linear dependence between Microsoft and S&P 500 returns. ■

**Exercise 22** *Pair-wise scatterplots for multiple series*

For more than two data series, the R function `pairs()` plots all pair-wise scatterplots in a single plot. For example, to plot all pair-wise scatterplots for the GWN, Microsoft returns and S&P 500 returns use:

```
> pairs(cbind(gwnMonthly,msftRetS,sp500RetS), col="blue",
+       pch=16, cex=1.25, cex.axis=1.25)
```

The top row of Figure 1.22 shows the scatterplots between the pairs (MSFT, GWN) and (SP500, GWN), the second row shows the scatterplots between the pairs (GWN, MSFT) and (SP500, MSFT), the third row shows the scatterplots between the pairs (GWN, SP500) and (MSFT, SP500). ■

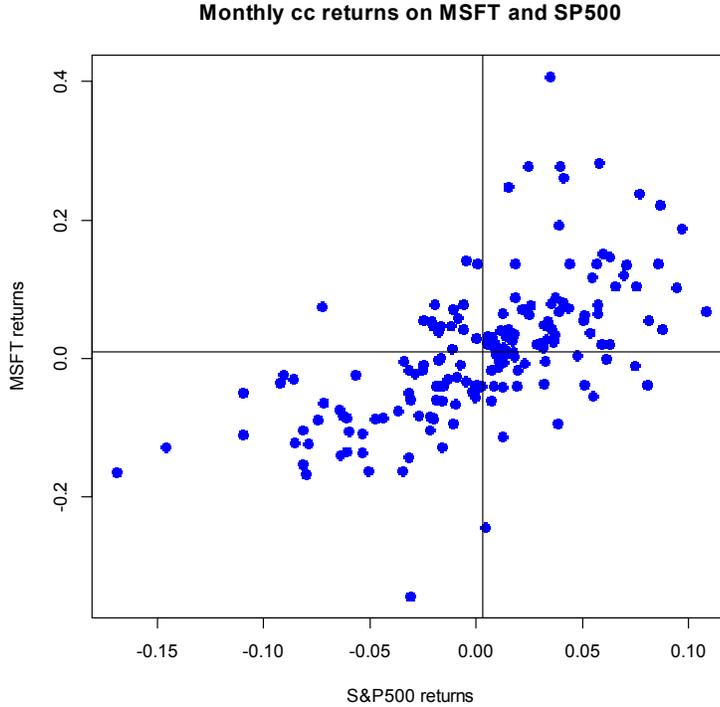


Figure 1.21: Scatterplot of Monthly returns on Microsoft and the S&P 500 index.

### 1.3.2 Sample Covariance and Correlation

For two random variables  $X$  and  $Y$ , the direction of linear dependence is captured by the covariance,  $\sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)]$ , and the direction and strength of linear dependence is captured by the correlation,  $\rho_{XY} = \sigma_{XY}/\sigma_X\sigma_Y$ . For two data series  $\{x_t\}_{t=1}^T$  and  $\{y_t\}_{t=1}^T$ , the sample covariance,

$$\hat{\sigma}_{xy} = \frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})(y_t - \bar{y}), \quad (1.8)$$

measures the direction of linear dependence, and the sample correlation,

$$\hat{\rho}_{xy} = \frac{\hat{\sigma}_{xy}}{\hat{\sigma}_x \hat{\sigma}_x}, \quad (1.9)$$

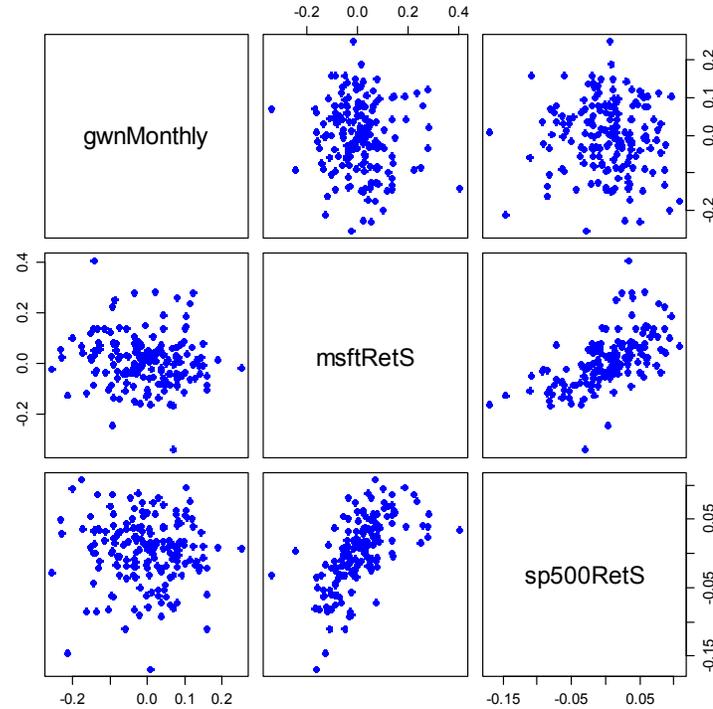


Figure 1.22: Pair-wise scatterplots between simulated GWN, Microsoft returns and S&P 500 returns.

measures the direction and strength of linear dependence. In (1.9),  $\hat{\sigma}_x$  and  $\hat{\sigma}_y$  are the sample standard deviations of  $\{x_t\}_{t=1}^T$  and  $\{y_t\}_{t=1}^T$ , respectively, defined by (1.3).

When more than two data series are being analyzed, it is often convenient to compute all pair-wise covariances and correlations at once using matrix algebra. Recall, for a vector of  $N$  random variables  $\mathbf{X} = (X_1, \dots, X_N)'$  with mean vector  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_N)'$  the  $N \times N$  variance-covariance matrix is

defined as

$$\boldsymbol{\Sigma} = \text{var}(\mathbf{X}) = \text{cov}(\mathbf{X}) = E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_N^2 \end{pmatrix}.$$

For  $N$  data series  $\{x_t\}_{t=1}^T$ , where  $x_t = (x_{1t}, \dots, x_{Nt})'$ , the sample covariance matrix is computed using

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{T-1} \sum_{t=1}^T (\mathbf{x}_t - \hat{\boldsymbol{\mu}})(\mathbf{x}_t - \hat{\boldsymbol{\mu}})' = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1N} \\ \hat{\sigma}_{12} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\sigma}_{1N} & \hat{\sigma}_{2N} & \cdots & \hat{\sigma}_N^2 \end{pmatrix}, \quad (1.10)$$

where  $\hat{\boldsymbol{\mu}}$  is the  $N \times 1$  sample mean vector. Define the  $N \times N$  diagonal matrix

$$\hat{\mathbf{D}} = \begin{pmatrix} \hat{\sigma}_1 & 0 & \cdots & 0 \\ 0 & \hat{\sigma}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\sigma}_N \end{pmatrix}.$$

Then the  $N \times N$  sample correlation matrix  $\hat{\mathbf{R}}$  is computed as

$$\hat{\mathbf{R}} = \hat{\mathbf{D}}^{-1} \hat{\boldsymbol{\Sigma}} \hat{\mathbf{D}}^{-1} = \begin{pmatrix} 1 & \hat{\rho}_{12} & \cdots & \hat{\rho}_{1N} \\ \hat{\rho}_{12} & 1 & \cdots & \hat{\rho}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\rho}_{1N} & \hat{\rho}_{2N} & \cdots & 1 \end{pmatrix}. \quad (1.11)$$

**Example 23** *Sample covariance and correlation between Microsoft and S&P 500 returns*

## 44 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

The scatterplot of Microsoft and S&P 500 returns in Figure 1.21 suggests a positive linear relationship in the data. We can confirm this by computing the sample covariance and correlation using the R functions `cov()` and `cor()`:

```
> cov(sp500RetS, msftRetS)
[1] 0.002979
> cor(sp500RetS, msftRetS)
[1] 0.6139
```

Indeed, the sample covariance is positive and the sample correlation shows a moderately strong linear relationship.

When passed a matrix of data, the `cov()` and `cor()` functions can also be used to compute the sample covariance and correlation matrices  $\hat{\Sigma}$  and  $\hat{\mathbf{R}}$ , respectively:

```
> cov(msftSp500RetS)
      MSFT  SP500
MSFT 0.010311 0.002979
SP500 0.002979 0.002284
> cor(msftSp500RetS)
      MSFT  SP500
MSFT 1.0000 0.6139
SP500 0.6139 1.0000
```

The function `cov2cor()` transforms a sample covariance matrix to a sample correlation matrix using (1.11):

```
> cov2cor(cov(msftSp500RetS))
      MSFT  SP500
MSFT 1.0000 0.6139
SP500 0.6139 1.0000
```



### Example 24 *Visualizing correlation matrices*

The R package **corrplot** contains functions for visualizing correlation matrices. This is particularly useful for summarizing the linear dependencies among many data series. For example, Figure 1.23 shows the correlation plot from the **corrplot** function `corrplot.mixed()` created using

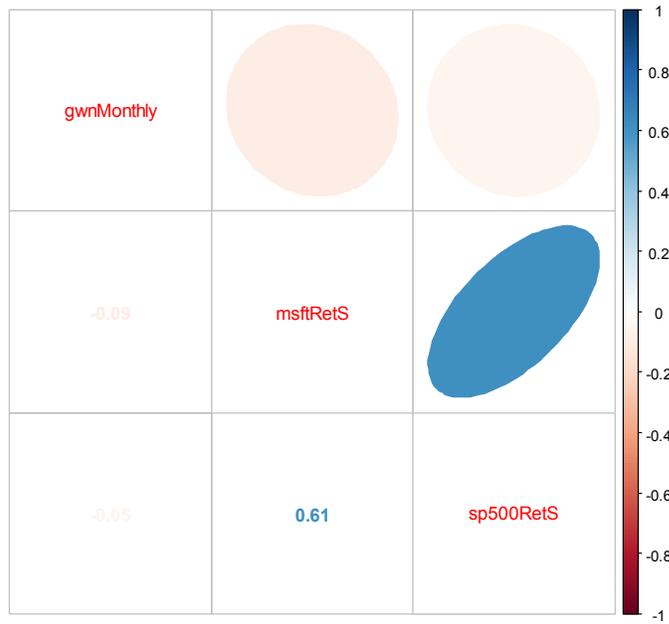


Figure 1.23:

```
> dataToPlot = merge(gwnMonthly, msftRetS, sp500RetS)
> cor.mat = cor(dataToPlot)
> corrplot.mixed(cor.mat, lower="number", upper="ellipse")
```

The color scheme shows the magnitudes of the correlations (blue for positive and red for negative) and the orientation of the ellipses show the magnitude and direction of the linear associations.

### 1.3.3 Stylized Facts for Daily and Monthly Asset Returns

A stylized fact is something that is generally true but not always. From the data analysis of the daily and monthly returns on Microsoft and the S&P 500 index we observe a number of stylized facts that tend to be true for other

## 46 CHAPTER 1 DESCRIPTIVE STATISTICS FOR FINANCIAL DATA

individual assets and portfolios of assets. For monthly data we observe the following stylized facts:

- M1. Prices appear to be random walk non-stationary and returns appear to be mostly covariance stationary. There is evidence that return volatility changes over time.
- M2. Returns appear to be approximately normally distributed. There is some negative skewness and excess kurtosis.
- M3. Assets that have high average returns tend to have high standard deviations (volatilities) and vice-versa. This is the no free lunch principle.
- M4. Returns on individual assets (stocks) have higher standard deviations than returns on diversified portfolios of assets (stocks).
- M5. Returns on different assets tend to be positively correlated. It is unusual for returns on different assets to be negatively correlated.
- M6. Asset returns are approximately uncorrelated over time. That is, there is little evidence of linear time dependence in asset returns.

Daily returns have some features in common with monthly returns and some not. The common features are M1 and M3-M6 above. The stylized facts that are specific to daily returns are :

- D2. Returns are not normally distributed. Empirical distributions have much fatter tails than the normal distribution (excess kurtosis).
- D7. Returns are not independent over time. Absolute and squared returns are positively autocorrelated and the correlation dies out very slowly. Volatility appears to be autocorrelated and, hence, predictable.

These stylized facts of daily and monthly asset returns are the main features of returns that models of assets returns should capture. A good model is one that can explain many stylized facts of the data. A bad model does not capture important stylized facts.

### 1.4 Further Reading

To be completed

## 1.5 Problems

**Exercise 25** *Histogram for returns with different number of bins*

**Exercise 26** *Smoothed density with different bandwidth parameters*

**Exercise 27** *Histogram overlaid with normal density*

Show PerformanceAnalytics function `chart.Histogram()`.

**Exercise 28** *Extracting unique covariance elements from covariance matrix*

**Exercise 29** *Visualizing correlation with heatmaps*

## 1.6 References

Ruppert, D. *Statistics and Data Analysis for Financial Engineering*. Springer-Verlag, New York.