1 Asset Returns

Most of financial studies involve returns instead of prices, of assets. For average investors, return of an asset is a complete and scale-free summary of the investment opportunity.

Prices and Returns

Denote P_t the price of an asset at time index t and assume for now that this asset pays no dividends.

Simple return Holding the asset for one period from date $t-1$ to date t would result in one period simple *net* return

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1\,.$$

The one-period simple *gross* return is simply $1 + R_t$. Holding the asset for k periods between dates $t-k$ and t gives a k-period simple gross return

$$\begin{aligned} 1 + R_t(k) &= \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \cdot \frac{P_{t-1}}{P_{t-2}} \cdots \frac{P_{t-k+1}}{P_{t-k}} \\ &= (1+R_t)\cdots(1+R_{t-k+1}) = \prod_{j=0}^{k-1}(1+R_{t-j}). \end{aligned} \tag{1.1}$$

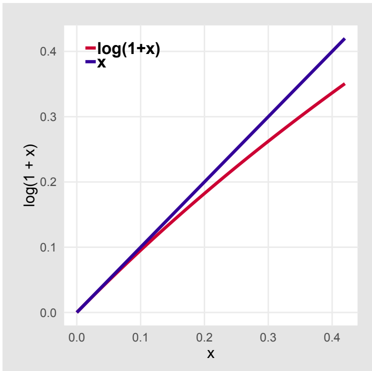
That is, the product of the k one-period simple gross returns involved. This is called a *compound return*. The k-period simple net return is simply $R_t(k) = (P_t - P_{t-k})/P_{t-k}$.

Continuous compounding Suppose that the annual interest rate of a bank deposit is i and the initial deposit is P_0. The net value a year later is $P_0(1+r)$ without compounding. It is $P_0(1+r/k)^k$ if compounded, where k is the number of times being compounded. The net value of continuous compounding a year later is P_0e^r by letting $k \to \infty$, and n years later P_0e^{nr}. If an investment after n years is P_n, then its net present value is P_ne^{-nr}. The net present value of an investment is a simple criterion for deciding whether to undertake an investment.

Continuously compounded return The *log return* or continuously compounded return is defined as

$$r_t = \log(1+R_t) = \log(P_t/P_{t-1}) = \log P_t - \log P_{t-1} = p_t - p_{t-1} \quad (1.2)$$

where $p_t = \log P_t$. When R_t is small, $r_t = \log(1+R_t) \approx R_t$ because $\log(1+x) \approx x$ for small $|x|$ by Taylor expansion. Hence, log returns and net returns are very similar for short period returns but less similar for longer period returns.



The advantage of continuously compounded returns become clear when we consider multiperiod returns,

$$r_t(k) = \log\{1+R_t(k)\} = r_t + \dots r_{t-k+1}. \quad (1.3)$$

from (1.1). Log returns are additive.

Eg 1.1. NVIDIA Corporation., monthly data Aug 18, 2023- Aug 18, 2024 from Yahoo! Finance, https://finance.yahoo.com/

NasdaqGS - Nasdaq Real Time Price • USD

NVIDIA Corporation (NVDA)

Aug 18, 2023 - Aug 18, 2024 | Historical Prices | Monthly

Date	Open	High	Low	Close	Adj Close	Volume
Aug 16, 2024	121.94	125.00	121.18	124.58	124.58	298,254,018
Aug 1, 2024	117.53	125.00	90.69	124.58	124.58	4,658,917,500
Jul 1, 2024	123.47	136.15	102.54	117.02	117.02	6,405,438,600
Jun 11, 2024	0.01 Dividend					
Jun 1, 2024	113.62	140.76	112.00	123.54	123.53	7,442,539,100
Jun 10, 2024	10:1 Stock Splits					
May 1, 2024	85.08	115.82	81.25	109.63	109.62	9,647,971,000
Apr 1, 2024	90.30	92.22	75.61	86.40	86.39	10,074,181,000
Mar 5, 2024	0.00 Dividend					
Mar 1, 2024	80.00	97.40	79.43	90.36	90.34	12,149,218,000
Feb 1, 2024	62.10	82.39	61.65	79.11	79.10	11,077,899,000
Jan 1, 2024	49.24	63.49	47.32	61.53	61.52	9,706,237,000
Dec 5, 2023	0.00 Dividend					
Dec 1, 2023	46.53	50.43	45.01	49.52	49.51	7,411,887,000
Nov 1, 2023	40.88	50.55	40.87	46.77	46.76	9,144,618,000
Oct 1, 2023	44.03	47.61	39.23	40.78	40.77	10,141,101,000
Sep 6, 2023	0.00 Dividend					
Sep 1, 2023	49.76	49.80	40.98	43.50	43.49	8,579,273,000

Loading financial data in R With the R package `quantmod`, we can load data from a variety of sources including Yahoo Finance. First, install the package if it has not been installed using the R command: `install.packages("quantmod")`

```
library(quantmod) ## load package each session
```

See the package documentation for the list of R functions and their help files. The package website, though outdated, contains useful examples.

```r
getSymbols("NVDA", from = "2023-08-18", to = "2024-08-18")

## [1] "NVDA"
```

An object NVDA has been created. The default of `getSmbols()` will load daily data from 2007-01-03 or the IPO date of the company to the current date. Use the R functions `head()` and `tail()` to show the first and last few entries.

```r
head(NVDA,n = 3)  ## default n = 6

##            NVDA.Open NVDA.High NVDA.Low NVDA.Close NVDA.Volume NVDA.Adjusted
## 2023-08-18    42.635    43.578   41.660     43.299   583768000      43.28604
## 2023-08-21    44.494    47.065   44.222     46.967   692573000      46.95295
## 2023-08-22    48.135    48.187   45.333     45.668   755293000      45.65434

tail(NVDA)

##            NVDA.Open NVDA.High NVDA.Low NVDA.Close NVDA.Volume NVDA.Adjusted
## 2024-08-09    105.64    106.60   103.43     104.75   290844200        104.75
## 2024-08-12    106.32    111.07   106.26     109.02   325559900        109.02
## 2024-08-13    112.44    116.23   111.58     116.14   312646700        116.14
## 2024-08-14    118.53    118.60   114.07     118.08   339246400        118.08
## 2024-08-15    118.76    123.24   117.47     122.86   318086700        122.86
## 2024-08-16    121.94    125.00   121.18     124.58   301838900        124.58
```

Daily data can become weekly, monthly, or yearly with `quantmod`'s R functions, `to.weekly()`, `to.monthly()`, `to.quarterly()` and `to.yearly()`. Below is monthly NVDA quotes of the past year.

```r
NVDA.monthly = to.monthly(NVDA); NVDA.monthly;

##          NVDA.Open NVDA.High NVDA.Low NVDA.Close NVDA.Volume NVDA.Adjusted
## Aug 2023     42.63     50.27    41.66      49.35  7542430000         49.34
## Sep 2023     49.76     49.80    40.98      43.50  8579273000         43.49
## Oct 2023     44.03     47.61    39.23      40.78 10141101000         40.77
## Nov 2023     40.88     50.55    40.87      46.77  9144618000         46.76
```



```r
## Dec 2023     46.53     50.43    45.01      49.52  7411887000         49.52
## Jan 2024     49.24     63.49    47.32      61.53  9706237000         61.52
## Feb 2024     62.10     82.39    61.65      79.11 11077899000         79.10
## Mar 2024     80.00     97.40    79.43      90.36 12149218000         90.35
## Apr 2024     90.30     92.22    75.61      86.40 10074181000         86.39
## May 2024     85.08    115.82    81.25     109.63  9647971000        109.62
## Jun 2024    113.62    140.76   112.00     123.54  7442539100        123.54
## Jul 2024    123.47    136.15   102.54     117.02  6405438600        117.02
## Aug 2024    117.53    125.00    90.69     124.58  4658917500        124.58
```

The output has 11 monthly (Sep 2023 – July 2024) quotes and 2 incomplete month quotes in the first and last rows. The column labels are self-explained. Adjustment is often used when examining historical returns or performing a detailed analysis on historical returns. Both `.close` and `.adjusted` are adjusted closing prices, the former is adjusted for stock splits, the later both stock splits and stock dividends.

The column can be retrieved by the usual way from a R matrix: `NVDA.monthly[, 6]` or `NVDA.monthly[,"NVDA.Adjusted"]`. The `quantmod` also provides convenient functions to retrieve individual column, `Op()`, `Hi()`, `Lo()`, `Cl()`, `Vo()`, `Ad()`.

Returns can be computed with either closing price or the fully adjusted closing price in the last column, to compute returns. We should always use the latter in analysis. Below is the monthly log returns r_t of NVDA in the past year applying (1.2).

```r
diff(log(Ad(NVDA.monthly)))

##          NVDA.Adjusted
## Aug 2023            NA
## Sep 2023   -0.12621886
```

```
## Oct 2023   -0.06454611
## Nov 2023    0.13705022
## Dec 2023    0.05726297
## Jan 2024    0.21705905
## Feb 2024    0.25138850
## Mar 2024    0.13293977
## Apr 2024   -0.04474659
## May 2024    0.23812763
## Jun 2024    0.11950866
## Jul 2024   -0.05422017
## Aug 2024    0.06260326
```

The return computation can be done by using `quantmod`'s *R* functions. Any periodic returns can be easily computed with `quantmod`'s `R` functions, `dailyReturn()`, `weeklyReturn()`, etc.

```
args(monthlyReturn)

## function (x, subset = NULL, type = "arithmetic", leading = TRUE,
##     ...)
```

The default type of `monthlyReturn()` `"arithmetic"` gives simple return. Setting `type = "log"` will return log returns.

```
monthlyReturn(Ad(NVDA), type = "log", leading = F)

##            monthly.returns
## 2023-08-31              NA
## 2023-09-29     -0.12621886
## 2023-10-31     -0.06454611
## 2023-11-30      0.13705022
## 2023-12-29      0.05726297
## 2024-01-31      0.21705905
## 2024-02-29      0.25138850
## 2024-03-28      0.13293977
## 2024-04-30     -0.04474659
## 2024-05-31      0.23812763
## 2024-06-28      0.11950866
## 2024-07-31     -0.05422017
## 2024-08-16      0.06260326
```


In this class, we will always use log returns based on the fully adjusted closing prices unless state otherwise.

The `R` functions `monthlyReturn()` and `yearlyRetrn()`, compute monthly returns of calendar months and yearly returns of calendar. Setting `leading = F` excludes the monthly return based on the first incomplete month. However, the last return, which is based on the period 2024-08-01 to 2024 -08-18, is included.

Dividends and Splits Adjusted close is the closing price after adjustments for all applicable splits and dividend distributions. All adjustments are backward-adjusted using appropriate split and dividend multipliers on dates prior to the split and prior to the dividend declared. See the details of calculating the adjustments in Yahoo! Finance.

The Yahoo! Finance display of NVDA in Eg. 1.1 shows that NVDA had dividend payments and stock splits during the past year. Splits and Dividends can also be loaded using `quantmod`'s `R` functions `getSplits()` and `getDividends()`.

```
getSplits("NVDA", from = "2023-08-18", to = "2024-08-18")

##            NVDA.spl
## 2024-06-10      0.1
```

The value $0.1 = 1/10$ which the Yahoo! Finance expressed as 10:1. The default setting for dividends has adjusted for split as in the Yahoo! Finance.

```r
getDividends("NVDA", from = "2023-08-18", to = "2024-08-18")
```

```
##            NVDA.div
## 2023-09-06    0.004
## 2023-12-05    0.004
## 2024-03-05    0.004
## 2024-06-11    0.010
```

Dividends without the split adjustment can also be obtained by setting `split.adjust = F`.

```r
getDividends("NVDA", from = "2023-08-18", to = "2024-08-18", split.adjust = F)
```

```
##            NVDA.div
## 2023-09-06     0.04
## 2023-12-05     0.04
## 2024-03-05     0.04
## 2024-06-11     0.01
```

Efficient Market Hypothesis and Random Walks

One of the most interesting questions in analyzing financial data is whether financial asset prices are forecastable. A related question if the asset returns are predicable. The null hypothesis of no predicability of asset returns is related to the random walk hypothesis of the price model.

The Efficient Market Hypothesis

The efficient markets hypothesis (EMH) can be traced back to Bachelier (1900) or ever earlier. Fama (1970) is often credited with providing constructive definition. A financial market is said to be effi-

cient if it fully and correctly reflects all currently available information in determining security prices. The efficient market hypothesis (EMH) says that stocks always trade at their fair value, it is impossible for an investor to consistently beat the market.

This leads to the random walk hypothesis by earlier statisticians. The random walk hypothesis however is more restrictive than what the EMH assumes. It states that increments of prices are independently distributed.

The Random Walk models

Any sequence S_t, $t \geq 0$ of real valued random variables with $i.i.d$ increments $Z_1, Z_2, \ldots$ and initial value S_0 is called random walk and S_0 its *delay*. That is, assuming $E(Z_t^2) < \infty$, S_t satisfies,

$$\begin{aligned} S_t - S_{t-1} &= Z_t\,, \\ Z_t &= \mu + \varepsilon_t\,, \qquad \varepsilon_t \sim i.i.d.(0, \sigma^2)\,. \end{aligned} \tag{1.4}$$

The mean of Z_t, μ is called the *drift* of $S_t, t > 0$.

Simple random walks The simplest random walk is zero-delay zero-drift random walk, *i.e.*, $S_0 = 0$ and $\mu = 0$. In this case, $S_t = \sum_{s=1}^{t} \varepsilon_s$, $E(S_t) = 0$ and $\mathrm{Cov}(S_t, S_{t+i}) = E(S_t S_{t+i}) = t\sigma^2$ for $i = 0, 1, \ldots$, since Z_t are uncorrelated. In particular, $\mathrm{var}(S_t) = t\sigma^2$.

In general, the model solves (1.4) is

$$S_t = S_0 + \sum_{s=1}^{t} Z_s$$

with conditional mean and variance,

$$E(S_t|S_0) = S_0 + \mu t \qquad \text{and} \qquad \mathrm{var}(S_t|S_0) = t\sigma^2 .$$

Conditioning on S_0, the variance (and covariance) is the same as the simple random walk.

Geometric Random Walks A positive random sequence X_t, $t \geq 0$ is a geometric random walk if $\log X_t$ is a random Walk.

To model P_t, the price of a financial asset at time t, we note that P_t is the price of an asset holding for t period with initial price P_0. Thus, from (1.1),

$$P_t = P_0\{1 + R_t(t)\} = P_0 \prod_{j=0}^{t-1} (1 + R_{t-j}) .$$

Following from (1.3),

$$\log P_t = \log P_0 + \log\{1 + R_t(t)\} = \log P_0 + r_1 + \cdots + r_t .$$

If $r_1, \ldots, r_t$ are $i.i.d.$ then P_t is a geometric random walk. If $r_1, \ldots, r_t$ are $i.i.d. N(\mu, \sigma^2)$, P_t is a lognormal geometric random walk.

The random walk hypothesis The random walk hypothesis is the hypothesis that stock prices behave at least approximately like a geometric random walk.

But is it or is it not a true random walk? If it is, then stock prices are unpredictable except in terms of long-run-average risk and return.

If we write the return sequence, r_t as

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t \sim i.i.d.(0, \sigma^2) . \tag{1.5}$$

The random walk hypothesis implies that $E(\varepsilon_{t+1} \,|\, \mathscr{F}_t) = 0$, where $\mathscr{F}_t$ is the available information generated by $\varepsilon_s, s \leq t$. That is, the increments r_{t+1} are not forecastable at time t as the EMH implies. The model 1.5, however goes beyond the first moment, it also implies that $\mathrm{var}(\varepsilon_{t+1} \,|\, \mathscr{F}_t) = \sigma^2$, a constant.

What do the empirical data show?

The i.i.d. assumption The empirical literature however shows that the assumption of $i.i.d.$ r_t is often violated. For example, the volatility in one period might depend on the volatility in recent periods. Stock return series exhibits volatility clustering, where time series show periods of high volatility and periods of low volatility.

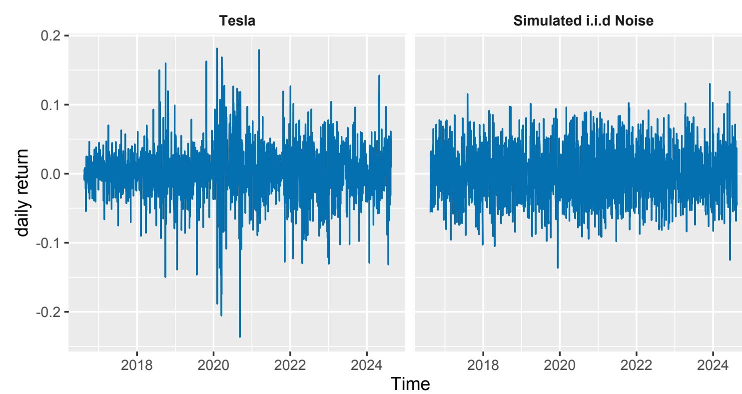


Figure 1.1: Log daily returns of Tesla Aug. 15, 2016 – Aug. 16, 2024 (left) and simulated Gaussian Noises (right).

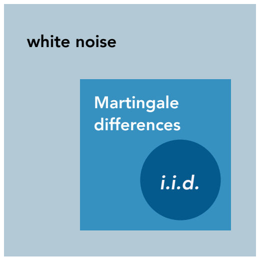
The left plot of Figure 1.1 is the daily log returns of Tesla, the right plot is the simulated Gaussian white noise of the same length $n = 2015$, with the same standard deviation 0.0373 as those of log returns of Tesla. The log return data appear to be more volatile and spiky at some periods. The local volatility which is the square root of conditional variance appears to be dependent on t. That is, $\mathrm{var}(\varepsilon_{t+1} \mid \mathscr{F}_t) = \sigma_t^2$.

The class of zero mean sequences satisfies only $E(\varepsilon_{t+1} \mid \mathscr{F}_t) = 0$ is called *Martingale differences*. The ARCH/GARCH models are examples of Martingale differences.

The $i.i.d.$ noise term in the random walk model (1.4) and the stock return model (1.1) can then be replaced by Martingale differences.

An even weaker version of random walk is that ε_t are *white noise*. That is,

$$E(\varepsilon_t) = 0\,, \quad \mathrm{var}(\varepsilon_t) = \sigma^2\,, \quad \mathrm{Cov}(\varepsilon_s, \varepsilon_t) = 0$$



The white noise process though is not linear forecastable, it is forecastable.

Nonforecastability In addition to time-varying volatility, there is also empirical evidence that the market is not fully efficient and the stock prices are partially predictable.

Short-term Momentum. More recent work by Lo and MacKinlay (1999) finds that short-run serial correlations are not zero and that the existence of too many successive moves in the same direction en-

able them to reject the hypothesis that stock prices behave as true random walks.

Long-Run Return Reversals. The short-term momentum does not continue, in fact there is evidence of negative autocorrelation in the long run. This has been dubbed "mean reversion". For example, Fama and French (1988) found that 25 to 40 percent of the variation in long holding period returns can be predicted in terms of a negative correlation with past returns.

Slow or Underreaction to New Information. If the full impact of an important news announcement is grasped over a period of time, stock prices will exhibit the positive serial correlation found by investigators. In such cases, the asset prices do not follow even the weak form of random walk.

Firm Size. Fama and French (1993) found evidence of correlation between the size of a firm and its return. It appears that smaller, perhaps more liquid firms, garner a greater return than larger firms.

A class of models for short-term predictable data is the ARIMA models with ARCH/GARCH errors. These models will be introduced later.