Explorations in Asset Returns

with Examples in R

by Richard Diamond

richard.diamond@pcacentral.com

Abstract

Robustness of common estimates (mean μ, volatility σ) from returns data depends on the period τ. It is common that we have a dataset of daily returns but would like to estimate and optimise portfolio allocations for the monthly horizon or longer. Statistical projection is required. To develop an intuition it is recommended to rediscover the rules taught about asset returns, the most common is the scaling of volatility σ with the square root of time period $\sqrt{\tau}$ and how that affects probability density. Building histograms with varying bucket size as well as constructing Q-Q plots by-step help to gain appreciation about densities and empirical nature of probability distribution.

This tutorial comes in three sections covering projection with asset returns, density histogram, and empirical Q-Q plot construction, for which R code is provided in the Appendix.

Keywords: asset returns, projection, histogram, percentiles, Q-Q plot, Normality tests, iid, R

JEL Classification: C5, G11

Projection with Asset Returns

This section gives multi-period projection with log-returns vs. linear returns for the asset price S_t

$$R_t^s = \frac{S_{t+1}}{S_t} - 1 \qquad \text{vs.} \qquad R_t^c = \ln \frac{S_{t+1}}{S_t}$$

where compound returns are converted into simple using

$$R_t^s = \exp R_t^c \; - 1$$

For log-returns (compound returns), a τ-period log return is the sum of single-period log returns over τ/n periods, because it is additive in the exponent. The multi-period returns are simple to obtain.

$$R_{t+\tau} \quad = \quad R_t + R_{t+1} + R_{t+2} + \cdots + R_{t+n-1}$$

$$R_{1Y} \quad = \quad R_{1\text{M}} + R_{2\text{M}} + \cdots + R_{12\text{M}}$$

where $R_{1\text{M}}$ is monthly return for $[0, 1\text{M}]$, $R_{2\text{M}}$ is monthly return for $[1\text{M}, 2\text{M}]$ and so on to $R_{12\text{M}}$ as monthly return for $[11\text{M}, 12\text{M}]$.

For linear returns (simple returns), projecting with linear returns from single periods over a longer time period involves multiplication of factors.

$$1 + R_{t+\tau} \quad = \quad (1 + R_t) \times (1 + R_{t+1}) \times \cdots \times (1 + R_{t+n-1})$$

$$1 + R_{1Y} \quad = \quad (1 + R_{1\text{M}}) \times (1 + R_{2\text{M}}) \times \cdots \times (1 + R_{12\text{M}})$$

$$1 + r_{1Y} \quad = \quad (1 + r_{1\text{M}}) \times (1 + f_{2\text{M}}) \times \cdots \times (1 + f_{12\text{M}})$$

This is also a relationship between the long term simple rate and a set of forward rates. The notation is fitting because $r_{1\text{M}}$ applies over $[0, 1\text{M}]$ and $f_{12\text{M}}$ is a *forward* rate known at the start of $[11\text{M}, 12\text{M}]$ period.

There are several rules about the use of returns ignorance of which creates common pitfals in estimation of portfolio allocations and risk measures.

While returns are free from a scale in dollars, they are not unitless – their unit is time. Returns are always estimated over some 'timescale', most common is daily timescale from R_t to $R_{t+\tau}$ the $\tau = 1/252$. If $\tau > 1M$ then a projection from shorter time periods over the longer horizon gives the effect of larger range (see Figure 1 below). A particular value of return, sampled from weekly or monthly period, can be as small as daily return. But we expect R_{1M} *to vary* within the larger range than R_{1D}.

The square-root rule for volatility $\sigma\sqrt{\tau}$ applies under the assumption that the compound returns are invariants – they behave identically and independently across time. The scaling of volatility with time is done as $\sigma_{10D} = \sqrt{\sigma_{1D}^2 + \sigma_{1D}^2 + \sigma_{1D}^2 + \ldots} = \sigma_{1D}\sqrt{10}$. As a rule of thumb, if $\sigma_{1D} = 0.01$ then annualised $\sigma \approx 0.16$ often expressed without an index.

Linear returns are to use in portfolio calculations... the familiar variance minimisation (MVP, Markowitz) objective function is defined for linear returns.

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \{\boldsymbol{w'\mu_\tau} - \boldsymbol{\lambda w'\Sigma_\tau w}\}$$

where subscript in μ_τ and Σ_τ means their estimation from market data of respective frequency, eg, $\tau = 1/252$ for daily and $\tau = 1/52$ for weekly. For instance, if allocations are reviewed weekly then it makes sense to operate with linear returns sampled weekly too.

The linear returns (not log-returns) aggregate across assets. Return calculation for a portfolio of N assets is done as a weighted average of their individual returns.

$$R_{t,\Pi} = w_1 R_{t,1} + w_2 R_{t,2} + w_3 R_{t,3} + \cdots + w_N R_{t,N}$$

but it is compound returns that are suited to make projections. The projection is important when making performance attributions. For example, when we have a sample of quarterly returns for a hedge fund and would like to make a projection for the distribution of its potential annual returns. Compound returns have a distribution that behaves better on projection: increasing $\sigma\sqrt{\tau}$ means simply more uncertainty associated with the longer horizons τ. The uncertainty is symmetric wrt positive and negative outcomes.

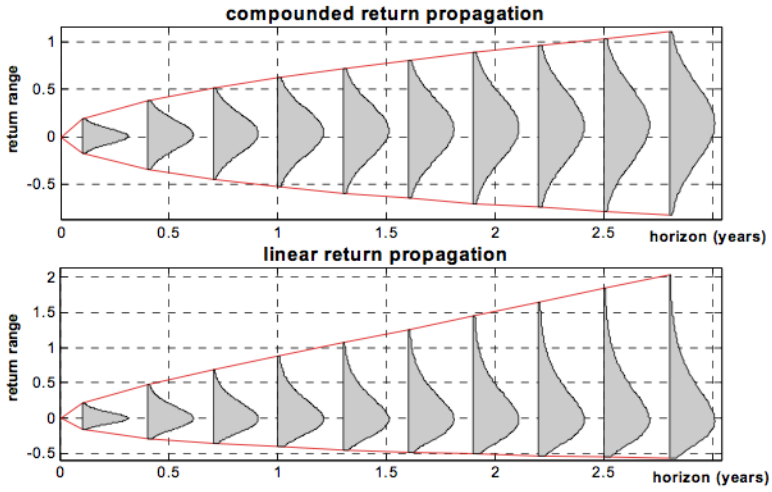

Figure 1: From: Linear versus Compounded Returns by Attillio Meucci (2010).

To give more detail on Figure 1, the analytical distribution is propagated without generating any random values. Instead, the *pdf* 'bell' is recalculated for a fixed range of quantiles $\in [0.01, 0.99]$ at each time step $\tau = \tau + 0.30$. Projection over long-term horizons of 2.5 and 2.8 years reveals distribution mismatch between simple and compound returns that are regarded interchangeable over small timescale.

Inferring the future distribution of (returns) performance over horizon that is longer than timescale (frequency) of observation is a case of statistical projection. The most common timescale for returns data is 'daily'. Therefore making statements about weekly, monthly or annual return would require a projection.

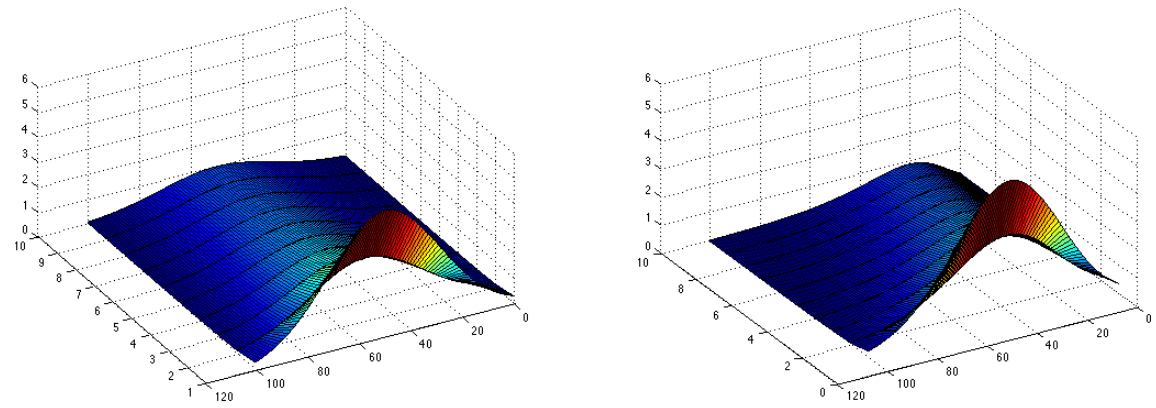

Figure 2: Linear versus Compound Returns: distribution propagation ($\mu = 0.05$, $\sigma = 0.25$).

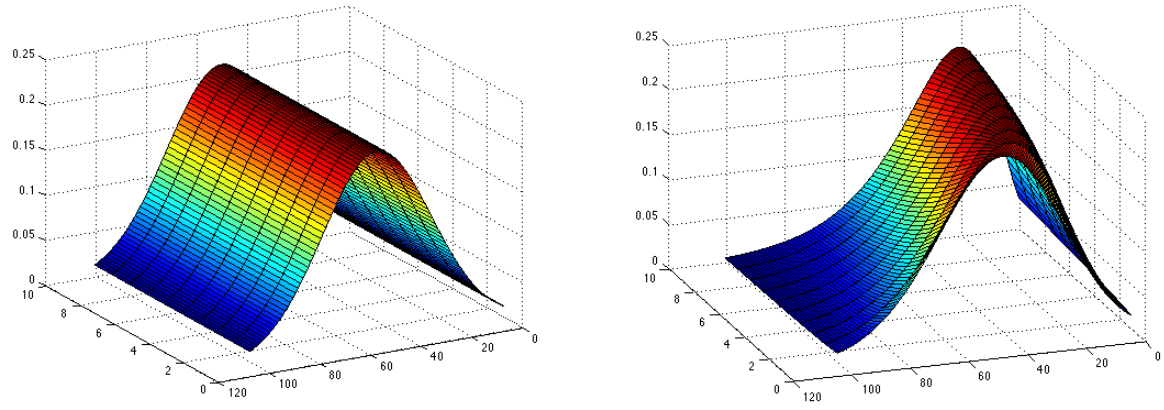

Figure 3: Distributons as above, standardised. Time horizon is axe scaled 0..10.

The standardisation of the propagated distribution at each time step reveals (bottom plots) that the difference is between the Normal distribution for compound returns and Lognormal distribution for simple returns. This follows from the exact relationship $R_t^s = \exp R_t^c - 1$ so, if R_t^c are Normal (and identically distributed) then simple returns are necessarily Lognormal. The Lognormal distribution propagates with a prominent asymmetry.

The usual direction of modelling is to condense information from time series (of returns) into a distribution. Then, the projection focuses on estimating the moments of a future distribution $f_{t+\tau}$ in one step (via an analytical procedure). The full projection requires evolving each initial observation as an asset path over the multiple steps (via Monte-Carlo). Therefore, **we understand 'the full projection' as a procedure that is opposite to the estimation of a distribution for invariants** (e.g., standardised returns is an example of iid invariants). The full projection always implies time series $\{R_{t\to t+1}\}_{t=1..T}$ as its result.

Histogram and Density Estimation

Histogram is a graphic representation of probability density function $f(x)$, a *pdf*, given the right bucket size. It is also a discrete representation. To built a histogram one actually engages in estimation of density,

$$f(x) = \frac{1}{Nh} n_j$$

where n_j is the varying number of observations in a bucket (count frequency), h is our bandwidth – bucket/window size on the scale of real numbers $\mathbb{R}$ (not count!), and N is the number of observations. Nh is an expected average amount of observation per bucket and n_j/Nh is density of actual/average observations, which is low in the tail regions and high in the middle, creating the 'bell' curvature for Normal density.

More formally, n_j/N gives percentage frequency, while n_j/Nh is 'a chunk of density' also called probability mass. If we set the bucket size such as to include one observation only then for each chunk, the *pdf* is $f(x) = 1/Nh$, where $1/h$ is a standardising parameter.

The proper **kernel smoothing** methods refer to h as bandwidth. The small setting will produce a histogram that repeats the plotted data (low smoothness), while sufficiently high bandwidth gives the large bucket size and smooths the representation into symmetric histogram as if the returns are Normally distributed or close.

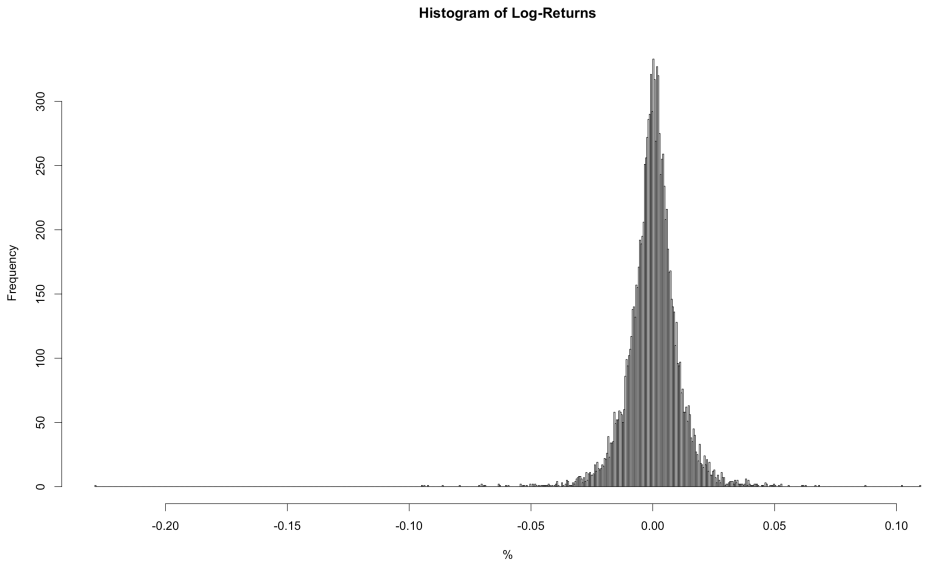

Figure 4: Simple histogram of SPX log-returns, built in R environment (code in Appendix).

To build Figure 4 histogram the parameter of 1000 breakpoints was used, the more breakpoints the less observations are in the bucket (low smoothness). To obtain a proper density that conforms to the analytical Normal distribution, experiment by setting the number of breakpoints lower but not too low.

Figure 4 histogram shows a count of observations per bucket. The density histogram is visually the same but vertical axe would have the standardised scale: density instead of frequency, However, if there is a need to determine the shape of density curve (i.e., is it like Normal or Lognormal?) or conform to an analytical distribution (usually done to convert the observations into uniformly distributed $[0, 1]$ pseudo-observations) it is recommended to use kernel smoothing methodology and kernel density plots in R in particular.

Quantile-Quantile Plot

Building a Q-Q plot in explicit steps is a straightforward task for Excel because for each data transformation it is possible to create a matching column.

Implementing step-by-step, you will obtain a result alike the following table. 'Historic' stands for the compound return of S&P 500 index in this case, followed by 'Scaled' return (Z-score), indexes and the Standard Normal Percentile which corresponds to the cumulative probability given by i/N. Notice the outlier return, the procedure allows to see that it is in excess of 21 standard deviations but its corresponding percentile and cumulative density are not as revealing.

Historic	Scaled	i	i/N	Standard
-0.22900	-21.20462	1	0.00009	-3.74534
-0.09470	-8.78146	2	0.00018	-3.56758
-0.09354	-8.67429	3	0.00027	-3.45987
-0.09219	-8.54970	4	0.00036	-3.38162
-0.08642	-8.01584	5	0.00045	-3.31983
-0.07922	-7.35036	6	0.00054	-3.26858
...	...	...	...	...

PDF $1/N$	CDF i/N
0.00009	0.00009
0.00009	0.00018
0.00009	0.00027
0.00009	0.00036
0.00009	0.00045
0.00009	0.00054
...	...

Table 1: Left: Inputs for a Q-Q plot. Right: Empirical PDF and CDF.

3.1 Q-Q Plot Step-by-Step

a. Scale empirical log-returns R_t^c to be the Normal variables $Z_t = \frac{R_t^c - \mu_\tau}{\sigma_\tau}$. For a market index, the average daily return $\mu_{1D} \approx 0$, increasingly so for a multi-year sample.

b. Sort the scaled returns in the ascending order and create an index column $i = 1..N$.

c. For each observation, the individual probability density (probability mass) is $1/N$ and the cumulative density is i/N. This allows to obtain percentiles.

d. The standardised percentile is obtained with the inverse cumulative density $\Phi^{-1}(i/N)$.

Plot the scaled returns (Z-scores) from step (a) against the theoretical percentiles from step (d). For the perfectly Normal log-returns, the Q-Q plot would be a straight line.

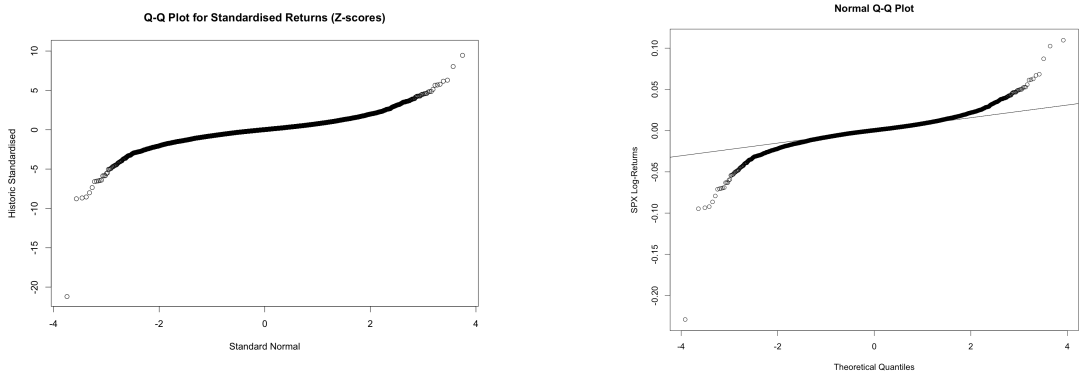

Figure 5: Q-Q Plot for SPX log-returns step by step (left) vs. built with *qqnorm()* and *qqline()* (right).

3.2 Q-Q Plot with R Libraries Time series experiments are best conducted in the suitable environment. Most tools recommended to students and analysts (Eviews, Stata, SPSS, SAS) package modelling into standardised statistical tests and procedures. They allow you to modify parameters but rarely the procedure itself.

R environment, on the other hand, offers convenience in manipulating time series and intuitive facilities of programming language. It is evident when statistical tests are implemented by specialists – the output organises important details. A quant job requires the active knowledge of the chosen libraries, which for R are 'zoo' objects for time series analysis, 'quantlib' main library of the functions for a financial quant, and 'urca' library for non-stationary time series (cointegration testing) which most of financial time series are.

3.3 Checking for IID-ness The purpose of a Q-Q plot is to test the data for conformance to the Normal Distribution. More precisely, the check is for observations to be independent while sampled from identical distributions, so the actual testing is for 'iid-ness' of returns series $\{R_{t\to t+1}\}_{t=1..T}$.

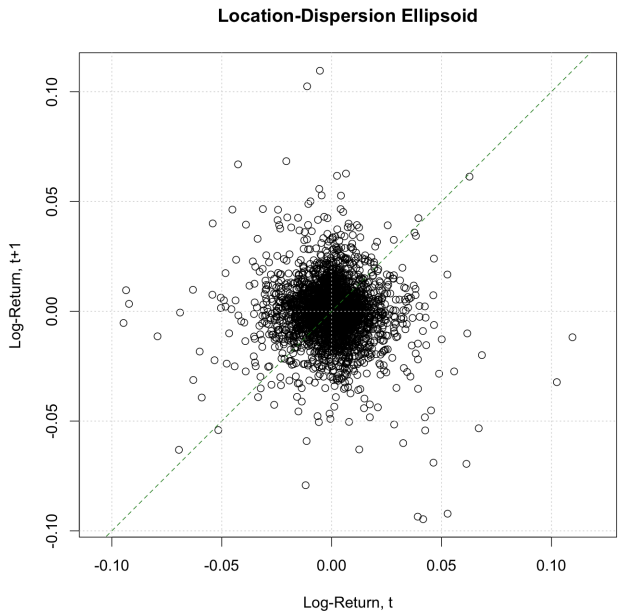

Figure 6: Lag Plot: an efficient iid check for a market index returns.

There is no formal method to test for iid-ness, and the named statistical tests for data being Normally distributed (e.g., Kolmogorov-Smirnov test) serve as proxies and must be supplemented by testing for serial correlation (e.g., LM test, ACF and PACF plots of significant autocorrelation values). An informal but robust iid check is performed by plotting R_{t+1} vs. R_t, the result is known as location-dispersion ellipsoid (Figure 6). The true and perfect iid observations would be evenly spread within a circle on Cartesian coordinates.

References

[1] *Statistics and Data Analysis for Financial Engineering* (2nd edition)
by David Ruppert & David Matteson, Springer, 2015.

[2] *Linear vs. Compounded Returns* by Attilio Meucci - The Quant Classroom, GARP Risk Professional Magazine, April 2010.

Appendix: R Code

The code below reads from *SPX.csv*, a file with two columns: date and closing price level. $\mathbb{P}2$ Estimation section of the code offers by-step procedure that one might replicate in Excel if their command of R is insufficient. The code also gives simple cases of matrix manipulation in R, with explanations in comments.

```
# 2015. Richard Diamond.
prices.zoo = read.zoo("SPX.csv", header=TRUE, sep=",", format = "%d/%m/%y")
prices.this = window(prices.zoo, start=as.Date("1950-01-03"), end=as.Date("2013-01-03"))
plot(prices.this$Close, main = "SPX Closing Prices", ylab = "Price (USD)", xlab = "Date")

# P1 Invariants. IID check (Histogram, Density Histogram, Dispersion Ellipsoid)
returns.this = diff(log(prices.this$Close))
hist(returns.this, breaks=1000, main = "Histogram of SPX Log-Returns", xlab="%") #or set freq=FALSE
lag.plot(returns.this, main = "Location-Dispersion Ellipsoid (Lag Plot)", diag=TRUE)

# P2 Estimation. Normality Check (Q-Q plot)
qqnorm(returns.this, ylab = "SPX Log-Returns")
qqline(returns.this)

# P2.1 Empirical Density
sreturns.this = (returns.this - mean(returns.this))/ sd(returns.this) #the long way
sreturns.this = scale(returns.this, center=TRUE, scale=TRUE) # each value is a Z-score

plot(sreturns.this, main = "SPX Standardised Returns (Z-scores)",
 ylab = "Standard Deviation", xlab = "Date")

# P2.2 Building a data matrix (table for presentation)
sreturns.N = length(sreturns.this)
sreturns.data = matrix(rep(NA, sreturns.N), sreturns.N, 3) #an empty structure
# dim(sreturns.data)

for(i in 1:sreturns.N)
  {
  sreturns.data[i,1]=i # This can be using seq(1:sreturns.N)
  sreturns.data[i,2]=i/sreturns.N
  sreturns.data[i,3]= qnorm(i/sreturns.N) # Can this be more efficient computationally?
  }
sreturns.data = cbind(sort(as.vector(sreturns.this)), sreturns.data)
# This is a tricky line: sorting a vector of scaled returns and appending it to data matrix

# P2.3 Q-Q Plot
plot(sreturns.data[,4],sreturns.data[,1], main = "Q-Q Plot for Standardised Returns (Z-scores)",
 ylab = "Historic Standardised", xlab = "Standard Normal")
```

```
#########################################################################
# REFERENCE Testing for Normality in R                                  #
# The following summary adopted from stackexchange.com                  #
#########################################################################

library(moments)
library(nortest)
library(e1071)

# Skewness and Kurtosis. Standard Normal values are 0 and 3.
skewness(returns.this)
kurtosis(returns.this)

# Kolmogorov-Smirnov test
ks.test(sreturns.this,"pnorm", mean(returns.this), sqrt(var(returns.this)) )
# Shapiro-Wilks test
shapiro.test(returns.this)
# Anderson-Darling test
ad.test(returns.this)

# Q-Q Plot: The points must be on the line or very close
qqnorm(returns.this)
qqline(returns.this)

# P Plot: The straight line must be a good fit for Normal data
probplot(returns.this, qdist=qnorm)

# Exercise: try Normality checks with empirical values rescaled to Z-scores
sreturns.this = scale(returns.this, center=TRUE, scale=TRUE)
probplot(sreturns.this, qdist=qnorm)

# Generate Normal random numbers
set.seed(111)
normal.sample <- rnorm(500, mean = 10, sd=1)

# True Normal Density for reference
f.density <- function(t) dnorm(t, mean(normal.sample), sqrt(var(normal.sample)))
curve(f.density, xlim=c(6,14))
hist(normal.sample, prob=T, add=T)

probplot(normal.sample, qdist=qnorm)
qqnorm(normal.sample)
qqline(normal.sample)
```