Random Behaviour of Assets

CQF Lecture One Explorations

CQF Lecture One exercises are explorations intended for you to do some experimentation with (1) returns estimation, (2) histogram and (3) Q-Q plots. The purpose is seeing how robust (or not) the estimates and rules are. By estimating returns over different timescales you re-discover the scaling of volatility σ with the square root of time $\sqrt{\delta t}$.

1. Multi-period projection with log-returns vs. linear returns for the asset price S_t

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

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

For log-returns (compounded returns),
A τ-period log return is simply the sum of the single-period log returns over τ/n periods, because it is additive in the exponent.

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

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

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 $[11\text{M}, 12\text{M}]$ (strictly following notation, we would have written R_t as $R_{11\text{M}}$). The multi-period returns are simple to obtain.

For linear returns (simple returns),
Projecting with linear returns from single periods over a longer time period (converting timescales) involves multiplication.

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

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

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

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.

The exercises have been edited by CQF Faculty, Richard Diamond, R.Diamond@fitchlearning.com

There are several rules about the use of returns that create common pitfals in estimation and portfolio management.

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 where for $R_{t+\tau} - R_t$ the $\tau = 1/252$.

The square-root rule for volatility $\sigma\sqrt{\delta t}$ applies under the assumption that the compounded returns are invariants – they behave identically and independently across time.

The familiar variance minimisation (MVP, Markowitz) objective function is defined for linear returns.

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

where subscript in μ_τ and Σ_τ means their estimation from market data of respective frequency,e.g., $\tau = 1/252$ daily.

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.

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

2. Histogram is a graphic representation of probability density function $f(x)$, a *pdf*. It is also a discrete representation. To built a histogram one actually engages in estimation of density as follows:

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

where n_j is the varying number of observations in a bucket, h is our bandwidth – bucket/window size on the scale of real numbers $\mathbb{R}$, and N is the sample size.

n_j/N gives frequency, while n_j/Nh is 'a chunk' of density also called probability mass. If we set the window size so small that it includes only one observation, then for each chunk the *pdf* is $f(x) = 1/Nh$, where $1/h$ is a normalising parameter.

With the proper 'kernel smoothing' methods small bandwidth setting h will produce a histogram that repeats the plotted data (low smoothness), while sufficiently high bandwidth will smooth the representation into a symmetric histogram as if the returns are Normally distributed.

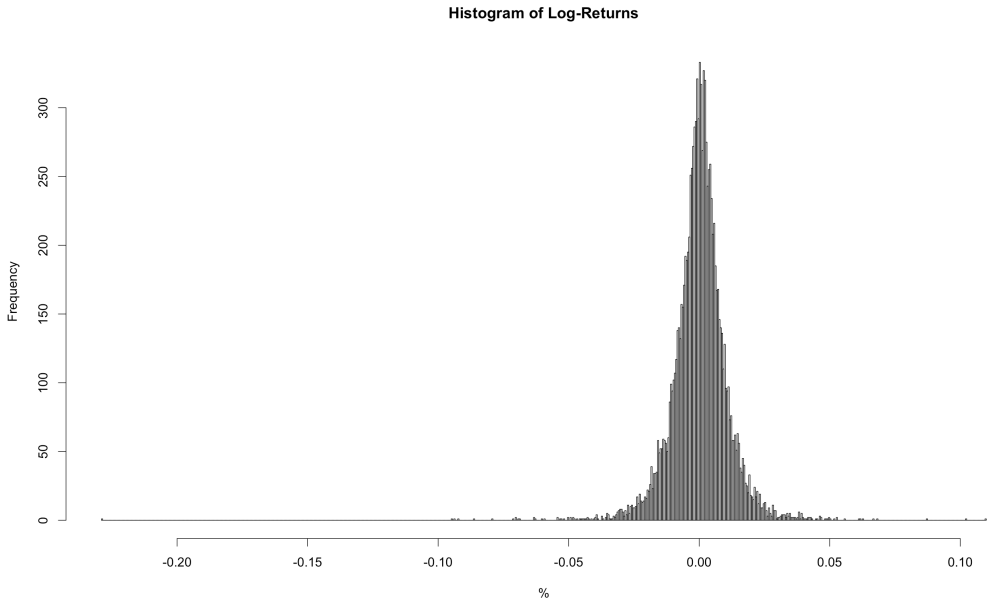

Figure 1: Histogram of SPX log-returns, built in R environment (code below).

1000 breakpoints were used. It is recommended that you experiment by setting the number of breakpoints higher (gives low smoothness) and lower but not too low.

3. Building a Q-Q plot in explicit steps is more straightforward in Excel because you can follow the transformations and organise the data in matching columns.

   Implementing the steps, you will obtain a result alike the following table. 'Historic' stands for the actual S&P 500 return, followed by 'Scaled' return (Z-score), indexes and the Standard Normal Percentile which corresponds to the cumulative probability given by i/N. Notice one past negative return that was in excess of 21 standard deviations!

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 to be the Normal variables $Z_t = \frac{r_t - \mu}{\sigma}$. For a market index the average daily return $\mu \approx 0$, that is increasingly valid for a large 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.

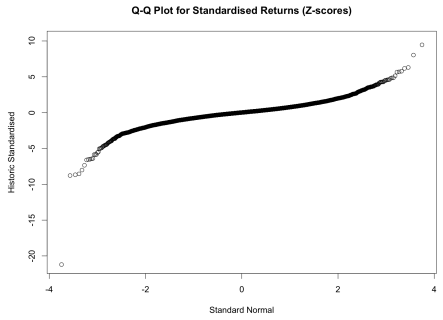

Figure 2: Q-Q Plot built step by step.

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

We have selected R environment for its convenience in manipulating the time series and intuitive facilities of programming language. It is also noticeable that statistical tests were implemented by specialists – the output is organised to present what is important. A quant job requires the active knowledge of the chosen libraries, e.g., 'zoo' objects for time series analysis, 'quantlib' main library of the functions for a financial quant, and 'urca' library for working with non-stationary time series (cointegration testing).

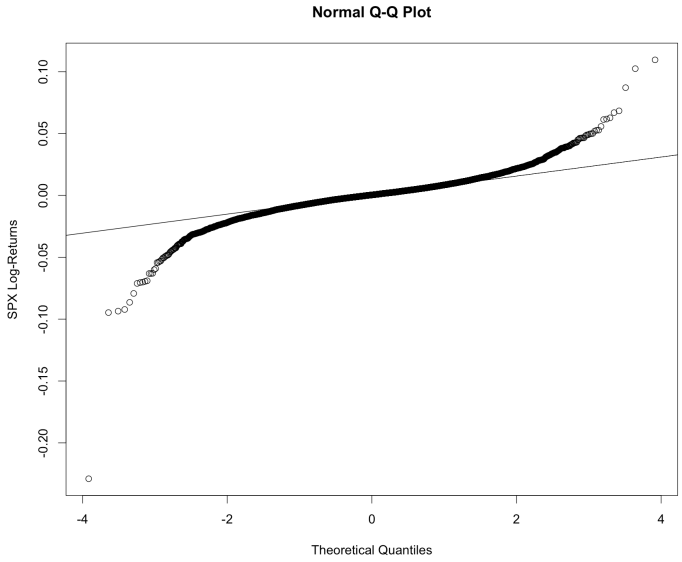

Figure 3: Q-Q Plot for SPX log-returns.

R Code: The code below reads from *SPX.xls* file distributed with Lecture One (saved as *.csv*). It selects certain sub-sample and plots a histogram and Q-Q plot.

```
#######################################################################
# 2015. Richard Diamond. Quieries to r.diamond@cqf.com                 #
# CQF Lecture One                                                     #
#######################################################################

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)
returns.this = diff(log(prices.this$Close))
hist(returns.this, breaks=1000, main = "Histogram of SPX Log-Returns", xlab="%")

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

The following section of the R code replicates the steps that you might like to do in Excel in the first instance. It provides an example of matrix manipulation in R. Code comments abridged.

```
#########################################################################
# 2015. Richard Diamond. Quieries to r.diamond@cqf.com                   #
# CQF Lecture One. Q-Q Plot Step-by-step                                 #
#########################################################################

# 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 the 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")
```