FinM 331/Stat 339 Financial Data Analysis, Winter 2010

Floyd B. Hanson, Visiting Professor

Email: fhanson@uchicago.edu

Master of Science in Financial Mathematics Program
University of Chicago

Lecture 4

6:30-9:30 pm, 25 January 2010, Ryerson 251 in Chicago

7:30-10:30 pm, 25 January 2010 at UBS in Stamford

7:30-10:30 am, 26 January 2010 at Spring in Singapore

4. Non-Normal Exploratory Data Analysis Tools:

4.1. Cauchy Distribution — A Pathological Example of a Fat Tail Distribution:

The **Cauchy distribution has a bell-shaped density**, like the normal density, but **that is a reciprocal quadratic**,

$$f_X^{(c)}(x;a,b) = \frac{b}{\pi(b^2+(x-a)^2)}, \tag{4.1}$$

and its distribution is related to the inverse tangent function, $\tan^{-1}(x)$ in mathematics or `atan(x)` on $(-\pi/2,+\pi/2)$ in **MATLAB**, i.e.,

$$F_X^{(c)}(x;a,b) = \frac{1}{\pi}\tan^{-1}\left(\frac{x-a}{b}\right)+\frac{1}{2}, \tag{4.2}$$

the integration technique being to let $z=(x-a)/b$ and note that the new integrand is the exact derivative of the tangent of z divided by π while $\tan^{-1}(-\infty)/\pi = -1/2$. Here, a is the **mode** or location of the density maximum as well as the **median**, and b is a scale parameter as well as the reciprocal of π times the height at the mode, $1/(\pi b)$.

- ***Cauchy Distribution and Pathology:***

  Since $f_X^{(c)}(x;a,b) = \mathrm{O}(1/x^2)$, the Cauchy density is integrable and probability is conserved, but $x f_X^{(c)}(x;a,b) = \mathrm{O}(1/x)$ as $|x| \to \infty$, so the mean of a Cauchy RV is $\mu^{(c)} = \mathrm{E}[X^{(c)}] = \infty$. Thus, the variance along with all other high moments are undefined with an infinite values. Now that is pathological since the tails are too fat for the Cauchy moments to be integrable. In fact, the Cauchy tails are so fat compared to normal tails, the relative size of the normal to Cauchy tails is still exponentially small as can be seen by an application of L'Hôpital's rule when x becomes large and using standard forms,

$$\frac{f_X^{(n)}(x;0,1)}{f_X^{(c)}(x;0,1)} = \frac{\sqrt{\pi}(1+x^2)}{\sqrt{2}\exp(x^2/2)} \xrightarrow{\mathrm{L'H}} \frac{\sqrt{2\pi}}{\exp(x^2/2)} \to 0^+, \quad (4.3)$$

  confirming that exponentials of large arguments will beat out powers, in most cases. Thus, **Cauchy tail asymptotic power law**:

$$f_X^{(c)}(x;a,b) \to \frac{b}{\pi \cdot x^2} \quad as \quad |x| \to \infty. \quad (4.4)$$

- ***Cauchy Distribution and Cauchy Principal Value (CPV):***

  The pathology is not so serious for practical reasons, since the data range is finite, the maximum log-return is never much more than 20% in absolute value and it is realistic to be interested in finite part of the tails anyway.

  Another reason is the improper integrals on the full infinite domain rigorously have to be treated in the limit on the finite domain $(-R_1, +R_2)$ as both values go to infinity. However, for random simulations, necessarily done in finite time, the results will be finite excluding numerical overflow, so another of Cauchy's ideas, the **Cauchy Principal Value**, takes advantage of antisymmetry of an integrand on $(-R, +R)$ as $R \to \infty$. Hence, the CPV value of the Cauchy mean for the standard distribution $(a, b) = (0, 1)$ is

$$\mu^{(cpv)} = \frac{1}{\pi} \lim_{R \to \infty} \int_{-R}^{+R} \frac{x dx}{1 + x^2} \equiv 0, \tag{4.5}$$

  by oddness, a practical mean. **However, in the standard case and the CPV variance, nothing can save the variance due to evenness.**

- ***Inverse Distribution and Distribution RNG by Uniform RNG:***

  For many CDFs, the following result permits defining one RNG in terms of the best known RNG, the **uniform RNG**:

  Theorem 4.1. Conversion of a CDF to Uniform CDF:
  If $F_X(x)$ *is* ***invertible****, and* X *is a RV with distribution* $F_X(x)$*, then*

$$F_X(x) \overset{\text{dist}}{=} F_U^{(u)}(u), \tag{4.6}$$

  where $F_U^{(u)}(u)$ *is the uniform cumulative distribution function.*

  "Sketch of Proof": For practical purposes, assume $F_X(x)$ is strictly increasing and continuous, so $(F_X)^{-1}(x)$ exists, but we ignore pathological cases. Let X be an RV with distribution $F_X(x)$ and that $U = F_X(X)$ is also an RV, then

$$F_{F_X}(X)(u) \equiv \text{Prob}[F_X(X) \le u] \overset{\text{incr.}}{=} \text{Prob}[X \le F_X^{-1}(u)] \tag{4.7}$$

$$\equiv F_X\left((F_X)^{-1}(u)\right)] \underset{\text{inverse}}{\overset{\text{defn}}{=}} u. \qquad \square \tag{4.8}$$

  The catch is that a practical and efficient computational formula for the inverse in needed for usefulness.

- ***Cauchy RNG Using Uniform RNG:***

  Since from (4.2),

$$u = F_X^{(c)}(x; a, b) = \tan^{-1}((x-a)/b)/\pi + 1/2, \tag{4.9}$$

  it takes some algebra to invert, so solving for x in terms of u,

$$x = \left(F_X^{(c)}\right)^{-1}(u; a, b) = b\tan(\pi(u-0.5)) + a, \tag{4.10}$$

  and by defining in MATLAB,

```
cauchyrnd(a,b,1,N)=b*tan(pi*(rand(1,N)-0.5))+a;
```
(4.11)

  which is **not** found in the **`Statistics Toolbox`**, but can easily be placed in a subfunction **`function`** within a **`function`** main **`m-file`** (note since **`tan`** and **`rand`** functions are vector function and everything else is a scalar the function should be vector and be efficient).

  Note: The same thing does NOT work for the Normal Distribution due to that lack of a inverse in terms of elementary function, although one exists on principle.

- ***Cauchy MATLAB Functions in Public Domain:***

  The function **cauchyrnd** is essentially one that can be can be found at **MathWorks Central File Exchange** in the downloaded public toolbox by Peder Axensten, *http://www.mathworks.com/matlabcentral/fileexchange/11749* . The package contains

  - **cauchycdf**: Cauchy cumulative distribution function (cdf).
  - **cauchyfit**: Parameter estimation for Cauchy data.
    {*Note: This constrained maximum likelihood m-file uses the Optimization Toolbox's fmincon, else uses the basic fminsearch.*}
  - **cauchyinv**: Inverse of the Cauchy cumulative distribution function (cdf).
  - **cauchypdf**: Cauchy probability density function (pdf).
  - **cauchyrnd**: Generate random numbers from the Cauchy distribution.

- ***Cauchy PDFs (height- and tail-adjusted) and 2008 Log-Returns:***

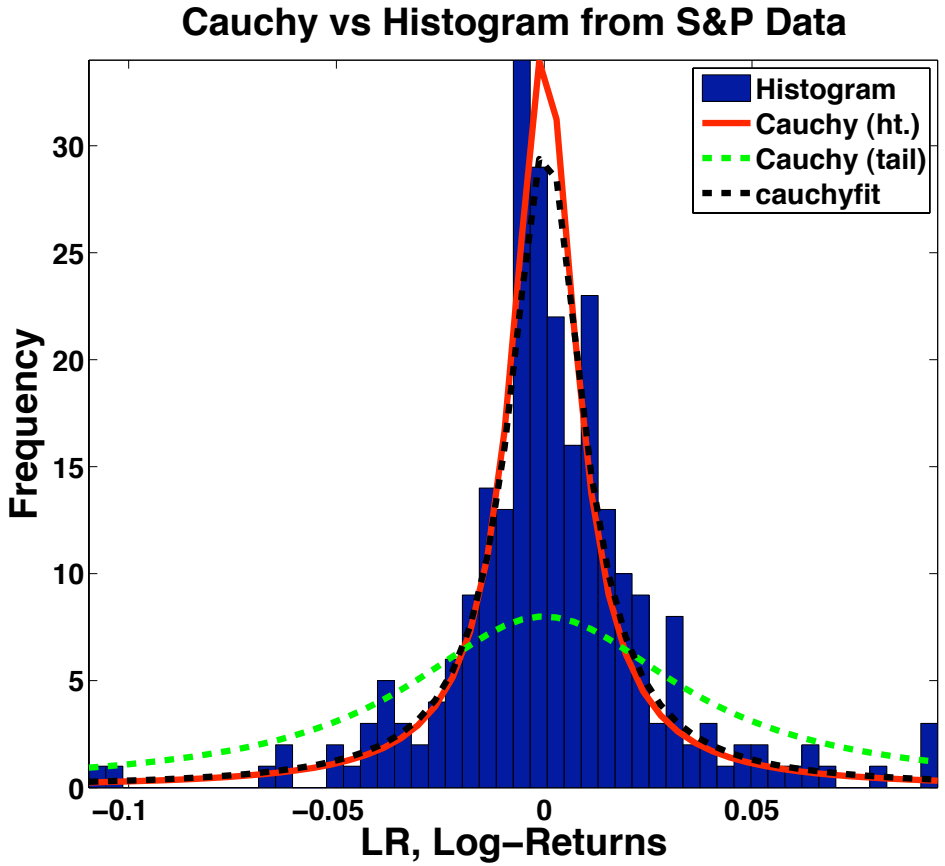


Figure 4.1: Combined plot of histogram of the 2008 S&P500 Index log-returns using **`cauchypdf`** with either height (**red —**) or tail (**green - - -**) to histogram's, but **do not fit in general (!), though look good for tails**. Also for comparison is the output of **`cauchyfit`** in (**black - - -**).

- ***Q-Q Plot of Cauchy (height-adjusted) Simulations versus 2008 S&P500 Log-Returns:***

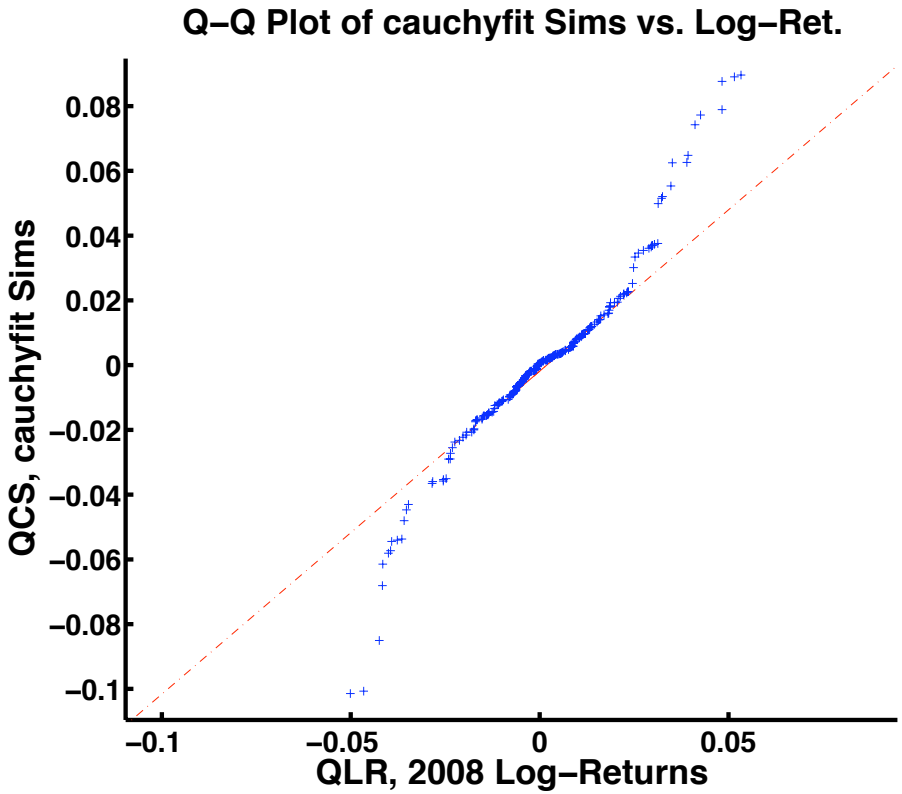


Figure 4.2: Q-Q Plot of `cauchyrnd` RNG simulations for height-adjusted Cauchy distribution versus 2008 S&P500 Index log-returns The simulations are a good representation of the tails of the data, any large deviations probably due to Cauchy theoretical infinite domain.

- ***Q-Q Plot of Cauchy (tail-adjusted) Simulations versus 2008 S&P500 Log-Returns:***

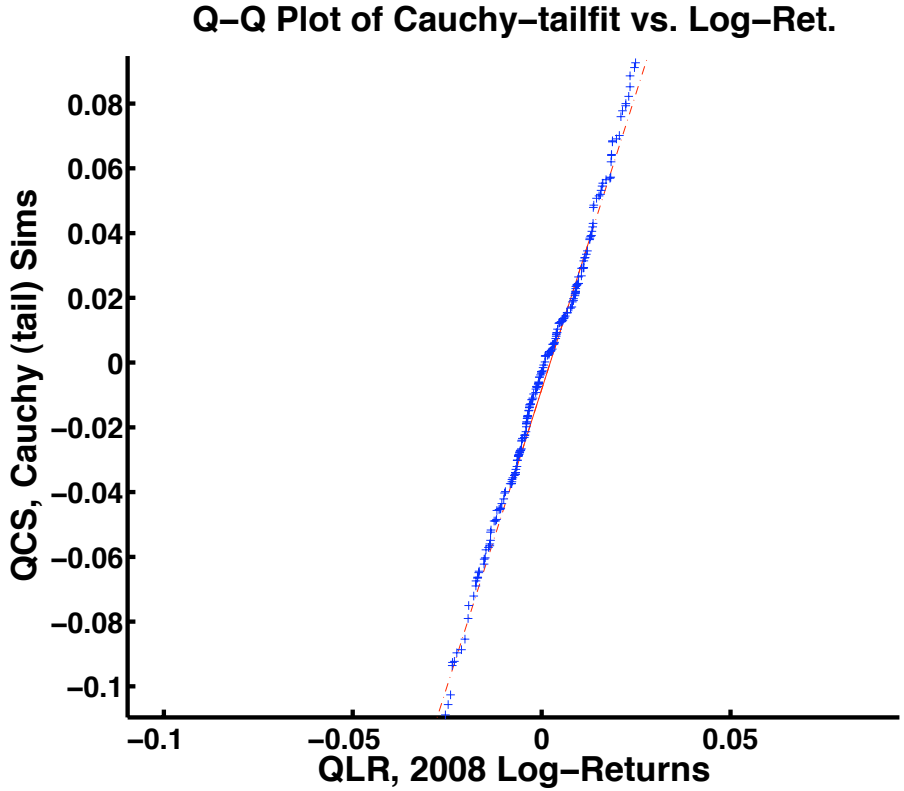


Figure 4.3: Q-Q Plot of the **`cauchyrnd`** RNG simulations for tail-adjusted Cauchy distribution versus 2008 S&P500 Index log-returns. The simulations are a better representation of the central part of the data and tails, excluding Cauchy large domain deviations.

- ***Cauchy and Normal Fits with height-adjusted Cauchy Sims Comparison of Fat and Very Thin Tails:***

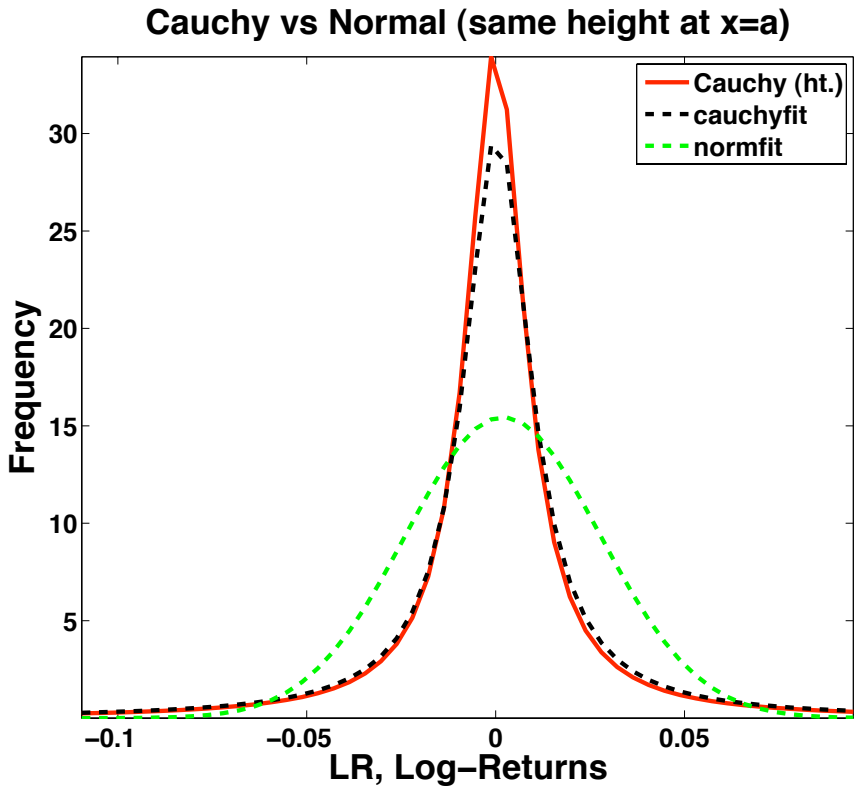


Figure 4.4: Comparison of Cauchy **`cauchyfit`** and Normal **`normfit`** fits with Cauchy (ht.) **`cauchypdf`** height-adjusted to LR height, with Cauchy (ht.) (**red** —), **`cauchyfit`** (**red** —) and **`normfit`** Normal (**green - - -**), showing differences in tail thickness.

- ***MATLAB Code for Cauchy PDFs with S&P500 Log-Return `histogram`, Two Q-Q Plots of Cauchy Simulations Against Log-Returns, and Cauchy-Normal Fits Comparison:***

```
function histspc2008cauchy
% Get Cauchy vs Histogram for Log-Returns Density
% for 2008 S&P500 ^GSPC (Yahoo Finance) Data;
%   Dates 2007/12/31-2008/12/31, Daily Adjusted Closings.
clc
load -ASCII S08.mat; % Note:  load function use.
fprintf('\nhistspc2008cauchy.m Output for Log-Returns (%s):',datestr(now));
L = length(S08);
LR = log(S08(2:L))-log(S08(1:L-1)); % Note: Vector Log Difference!
NLR = L-1;
minLR = min(LR); maxLR = max(LR); dLR = maxLR-minLR;
fprintf('\nminLR = %7.5f; maxLR = %5.3f;',minLR,maxLR);
a = median(LR); h1 = 34.5; b1 = 1/(pi*h1); % h = height from histogram
fprintf('\ncauchy height fit: a = %7.3e; b1 = %7.5f; h1 = %7.5f;',a,b1,h1);
h2 = 8; b2 = 1/(pi*h2); % h = height from histogram
fprintf('\ncauchy tail fit: a = %9.3e; b2 = %7.5f; h2 = %7.5f;',a,b2,h2);
%
figure(1); nfig=1;
nb = 50;
hist(LR,nb); hold on; % hold hist for adding;
```

```
dz = dLR/(nb-1); z = minLR + dz*(0:nb-1);
fzc1 = cauchypdf(z,a,b1);
fzc2 = cauchypdf(z,a,b2);
[mleparms,res] = cauchyfit(LR);
fprintf('\ncauchy output: mleparms,res; size(mleparms)=[%i,%i];'...
    ,size(mleparms));
afit = mleparms(1,1); bfit = mleparms(1,2);
fprintf('\n[afit,bfit]=[%9.3e,%9.3e]',afit,bfit);
fzcfit = cauchypdf(z,afit,bfit);
res
scrsize = get(0,'ScreenSize'); % figure spacing for target screen
ss = [5.0,4.5,4.0,3.5]; % figure spacing factors
plot(z,fzc1,'-r',z,fzc2,'--g',z,fzcfit,'--k','LineWidth',4);
axis tight; hold off;
title('Cauchy vs Histogram from S&P Data'...;
    ,'Fontsize',24,'FontWeight','Bold');
xlabel('LR, Log-Returns','Fontsize',24,'FontWeight','Bold');
ylabel('Frequency','Fontsize',24,'FontWeight','Bold');
legend('Histogram','Cauchy (ht.)','Cauchy (tail)','cauchyfit');
set(gca,'Fontsize',18,'FontWeight','Bold');
set(gcf,'Color','White','Position' ...
   ,[scrsize(3)/ss(nfig) 60 scrsize(3)*0.60 scrsize(4)*0.80]); %[l,b,w,h]
%
figure(2); nfig=2;
```

```
Xcsims1 = cauchyrnd(afit,bfit,1,NLR);
qqplot(LR,Xcsims1); axis([minLR maxLR minLR maxLR]) % tight;
title('Q-Q Plot of cauchyfit Sims vs. Log-Ret.'...;
    ,'Fontsize',24,'FontWeight','Bold');
xlabel('QLR, 2008 Log-Returns','Fontsize',24,'FontWeight','Bold');
ylabel('QCS, cauchyfit Sims','Fontsize',24,'FontWeight','Bold');
set(gca,'Fontsize',20,'FontWeight','Bold','LineWidth',3);
set(gcf,'Color','White','Position' ...
   ,[scrsize(3)/ss(nfig) 60 scrsize(3)*0.60 scrsize(4)*0.80]); %[l,b,w,h]
%
figure(3); nfig=3;
Xcsims2 = cauchyrnd(a,b2,1,NLR);
qqplot(LR,Xcsims2);  axis([minLR maxLR minLR maxLR]) % tight;
title('Q-Q Plot of Cauchy-tailfit vs. Log-Ret.'...;
    ,'Fontsize',24,'FontWeight','Bold');
xlabel('QLR, 2008 Log-Returns','Fontsize',24,'FontWeight','Bold');
ylabel('QCS, Cauchy (tail) Sims','Fontsize',24,'FontWeight','Bold');
set(gca,'Fontsize',20,'FontWeight','Bold','LineWidth',3);
set(gcf,'Color','White','Position' ...
   ,[scrsize(3)/ss(nfig) 60 scrsize(3)*0.60 scrsize(4)*0.80]); %[l,b,w,h]
%
figure(4); nfig=4;
%sigma = 1/(sqrt(2*pi)*h1); % same height h1 at x = a;
[muhat,sigmahat,muci,sigmaci] = normfit(LR); % Special MLE
```

```
fprintf('\nnormfit: muhat=%9.3e; sigmahat=%9.3e;'...
    ,muhat,sigmahat);
fprintf('\nnormfit: muci=[%8.2e,%8.2e]; sigmaci=[%8.2e,%8.2e];'...
    ,muci,sigmaci);
fznormfit = normpdf(z,muhat,sigmahat);
plot(z,fzc1,'-r',z,fzcfit,'--k',z,fznormfit,'--g','LineWidth',3);
axis tight;
title('Cauchy vs Normal (same height at x=a)'...;
    ,'Fontsize',24,'FontWeight','Bold');
xlabel('LR, Log-Returns','Fontsize',24,'FontWeight','Bold');
ylabel('Frequency','Fontsize',24,'FontWeight','Bold');
legend('Cauchy (ht.)','cauchyfit','normfit');
set(gca,'Fontsize',18,'FontWeight','Bold');
set(gcf,'Color','White','Position' ...
   ,[scrsize(3)/ss(nfig) 60 scrsize(3)*0.60 scrsize(4)*0.80]); %[l,b,w,h]
fprintf('\n');
```

```
==================Output====================

histspc2008cauchy.m Output for Log-Returns (20-Jan-2010 21:40:29):
minLR = -0.10957; maxLR = 0.095;
cauchy height fit: a = 0.000e+00; b1 = 0.00923; h1 = 34.50000;
cauchy tail fit: a = 0.000e+00; b2 = 0.03979; h2 = 8.00000;
cauchy output: mleparms,res; size(mleparms)=[1,2];
[afit,bfit]=[4.518e-04,1.058e-02]
res =       iterations: 40
        funcCount: 41
     cgiterations: 29
    firstorderopt: 6.0140e-04
        algorithm: 'large-scale: trust-region reflective Newton'
          message: [1x444 char]
             call: 'fmincon'
         exitflag: 2
normfit: muhat=1.921e-03; sigmahat=2.583e-02;
normfit: muci=[-1.28e-03,5.12e-03]; sigmaci=[2.38e-02,2.83e-02];
>>
```

4.2 Data Estimation of Distribution Functions:

Again let $\vec{X} = [x_i]_{n\times 1}$ denote the data observations, hopefully IID with common CDF $F_X(x)$ and the corresponding CDF estimate by $\widehat{F}_n(x) \simeq F_X(x)$ where

$$\widehat{F}_n(x) \equiv \frac{1}{n}\sum_{i=1}^{n} 1_{\{x_i \le x\}} = \frac{K(x)}{n}, \tag{4.12}$$

where $1_{\{x_i \le x\}}$ is the indicator function for the set $\{x_i \le x\}$ for each $i = 1:n$, one in the set and zero out of the set, so the above sum represents the cumulative count, $K(x)$, of all the observations less than or equal x. This averaged count will be **piece-wise continuous** but by the **Fundamental Law of Statistics**[a],

$$\widehat{F}_n(x) \to F_X(x) \ \ as \ \ n \to \infty. \tag{4.13}$$

For the corresponding estimate of some example financial functionals, thus for example, we write $\widehat{\mathrm{VaR}}_N[\widehat{F}_n](\alpha) \simeq \mathrm{VaR}_N[F_X](\alpha)$ for value at risk or $\widehat{\mathrm{ES}}[\widehat{F}_n](\alpha) \simeq \mathrm{ES}[F_X](\alpha)$ for the **expected shortfall**.

[a] Carmona (2004), p. 29.

4.3 Order Statistics:

Ordered observations of samples $\vec{X} = [X_i]_{n\times 1}$ of size n satisfy,

$$\min(\vec{X}) \equiv x_n^{(1)} \le x_n^{(2)} \le \cdots \le x_n^{(n-1)} \le x_n^{(n)} \equiv \max(\vec{X}), \quad (4.14)$$

upon sorting (see **`MATLAB function sort`**), allowing for nonunique values due to ties. They play an essential role in constructing **quantiles**, $[q_i]_{m\times 1}$, **empirically estimated CDFs**, $\widehat{F}_n(x)$, and other statistical quantities, usually in the background with computational software.

It is assumed that the ordered observations, $x_n^{(k)}$ for for $k = 1:n$, correspond to realizations of RVs $X_n^{(k)}$ called ***k*th order statistics**. For example, the quantile marks q_k associated with probabilities p_k can be reformulated as

$$\widehat{q}_k = q_k[\widehat{F}_n] = x_n^{(k)}, \quad for \ \frac{k-1}{n} < p_k \le \frac{k}{n}. \quad (4.15)$$

Thus, with $F_X(q) = p$ and F_X invertible,

$$\begin{aligned}
\mathbf{Prob}\left[\sum_{i=1}^{n} 1_{\{x_i \in (q,1]\}} = k\right] &= \mathbf{Prob}\left[q \in [X_n^{(n-k)}, X_n^{(n-k+1)})\right] \\
&\overset{\text{inverse}}{\overset{F_X}{=}} \mathbf{Prob}[p \in [F_X(X_n^{(n-k)}), F_X(X_n^{(n-k+1)}))] \\
&= \mathbf{Prob}[k \text{ objects in } n \text{ bins}] \qquad (4.16) \\
&\overset{\text{prob}}{\overset{\text{bino}}{=}} \binom{n}{k} (1-p)^k p^{n-k},
\end{aligned}$$

noting that the usual binomial p is replaced by $(1-p)$ since we are counting bins from the right rather than from the left.

4.4 Extreme Values, Fat Tails, Pareto Distributions, and POTs (Peaks Over Threshold):

The Cauchy distribution is a simple example of fat tails attached to a bell-shaped central distribution, but once the median or mode is determined there is only one parameter that specifies the shape to fit to the tail. Then there is the **generalized Pareto (GP) distribution** of power distributions that are used to **fit just the tail part of the distribution**. There are many variations of the Pareto distribution, so we just list the form of the density or PDF used by **MATLAB**,

$$f_X^{(\mathrm{gp})}(x;K,b,\theta)=\frac{1}{b}\left(1+\frac{K}{b}(x-\theta)\right)^{-1-1/K}, \qquad (4.17)$$

where K is the **shape parameter** (if $K=0$, then a **limiting exponential form** is used as $K\to 0$), b is the **scale parameter** and θ is the **location parameter**. The θ is usually not used, so a usual form it more like

$$f_X^{(\mathrm{gp})}(x;K,b,0)=\frac{1}{b}\left(1+\frac{Kx}{b}\right)^{-1-1/K}. \qquad (4.18)$$

There are many restrictions to the **generalized Pareto (GP) parameters** and the user can check **`MATLAB Help`** for them.

`MATLAB` has the typical family of **GP** support functions named for functionality, such a **`gppdf`**, **`gpcdf`**, **`gpinv`**, **`gpstat`**, **`gprnd`**, maximum likelihood function **`[pg2parms,gpci]=gpfit(Xpg);`** gives **`[K,sigma]=pg2parms`** with 95% confidence interval given threshold **`theta`**, so use **`Xpg=X-theta`**, and **`neglogLH=gplike(pg3parms,Xpg)`** separately gives the negative log-likelihood value given all 3 parameters.

For separating a tail from the central part of the distribution, the technique of **Peak Over Threshold (POT)** is used by picking a value a location where the ordered observations differ markedly from the normal distribution, often by **eye-balling** the **Q-Q plot** of the observation quantiles against the normal quantiles for the observation value that markedly differs from the linear line, **but sample size must be sufficient**.

For example, examining the Q-Q plot comparing the 2008 S&P 500 log-returns against the normal distribution in Lecture 3 on page 25 indicates that $\mathbf{LRpot} = -0.04$ looks like a good **POT** value, but $\mathbf{LRpot} = -0.02$ yields just a sufficient number of points for fitting. Then the user can sort the observation date in ascending values (e.g., using **MATLAB `sort`** function), next peel off the tail observations that do not exceed $\mathbf{LRpot}$ and finally storing the positive parts (the **GP fitting functions** and **economist/sociologist Vilfredo Pareto expect a positive tail**) into another vector, say

$$\mathbf{LRtail} = \{-\mathbf{LR}_i : \mathbf{LR}_i < \mathbf{LRpot}\}. \tag{4.19}$$

In **MATLAB** it can be implemented using logical expressions as an indicator function (**without a `*` sign**), i.e.,

```
LRtail=-LR(LR<=LRpot);
```
(4.20)

However, **`gpfit.m` did not fit** with a boundary warning for **`POTs`** too short for sufficient sample size.[a]

[a] The public domain **MATLAB** code **`expan.m`** for ***fast exponential analysis: Expan.m*** of **`POT`** data by Pieter Van Gelder was tested and be edited for class presentation purposes, results follows several pages later.

- ***Q-Q Plot, Reprinted from Lecture 3, page 25, Comparing 2008 Log-Returns Against Corresponding Normal Distribution:***

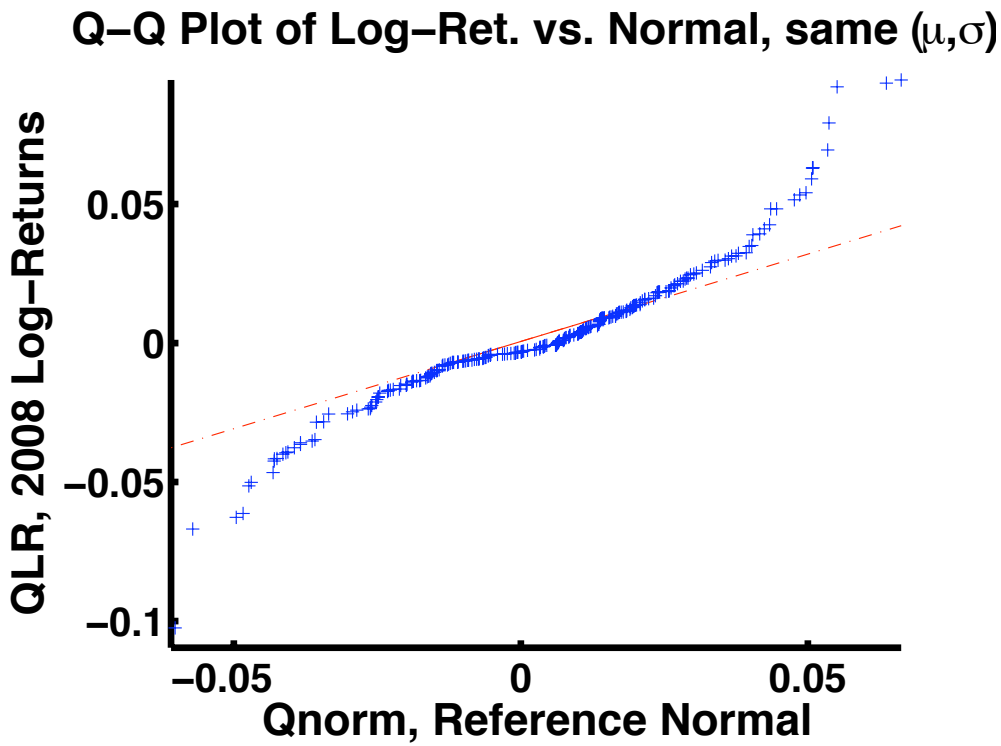


Figure 4.5: Q-Q Plot of **S&P500 Index log-returns** for the whole year 2008 compared to a simulated normal distribution with the **same** mean (μ) and standard deviation or volatility (σ). ***Look at those really fat tails!***

- ***Histogram of Negative Tail of 2008 S&P 500 Log-Returns Up To the POT:***

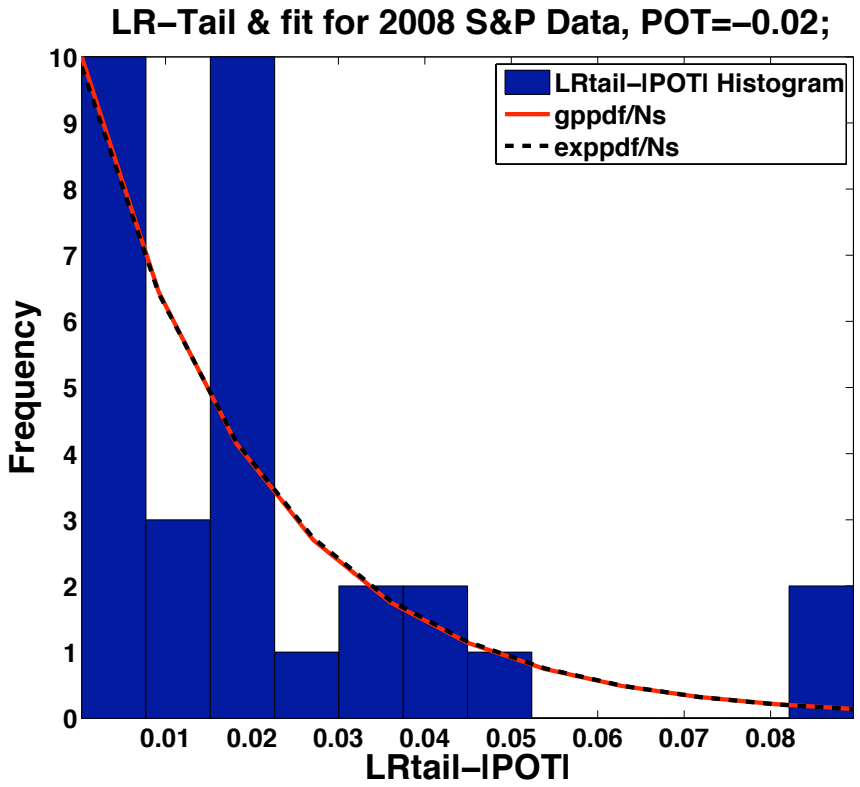


Figure 4.6: The histogram display the negative tail of the Log-Returns with sign reversed on $[-\mathbf{POT}, -\mathbf{LRmin}]$, where $\mathbf{POT} = -\mathbf{0.02}$ (big, count=31). The fitted ***red*** — line is from `gpfit` (Only works if $\mathbf{POT} \geq -\mathbf{0.02}$ for sufficient sample, so transition jump-tail.) and closely fitting **black - - -** line is from `expfit`.

- **MATLAB Code for Histogram of Negative Tail (Sign Reversed) of 2008 S&P 500 Log-Returns, with POT, gplike and expplike:**

```
function tailfitspc2008potvary
% Get Tail Fit on Left for Log-Returns Density
% for 2008 S&P500 ^GSPC (Yahoo Finance) Data;
%   Dates 2007/12/31-2008/12/31, Daily Adjusted Closings.
clc
load -ASCII S08.mat; %  load function used for 2008 data.
fprintf('\ntailfitspc2008potvary Output, (%s):',datestr(now));
L = length(S08); % NLR = L-1;
LR = log(S08(2:L))-log(S08(1:L-1)); % Note: Vector Log Difference!
minLR = min(LR); maxLR = max(LR);
fprintf('\nminLR = %7.5f; maxLR = %5.3f;',minLR,maxLR);
% LRsort = sort(LR)'; % get order statistics transpose
POT = -0.02; % "guestimate" Peak Over Threshold from Q-Q plot
LRtail = -LR(LR<=POT); % gpfit data must be positive;(LR<=POT)=indicator;
Npot = length(LRtail);
fprintf('\nPOT = %7.5f; Npot = %3i;',POT,Npot);
fprintf('\nLRtail=');
for i=1:Npot,fprintf('%5.3f;',LRtail(i));end, % fprintf('\n ');
theta = -POT; % 0; % -(minLR+POT)/2;
fprintf('\ntheta = %7.4f;',theta);
Xgp = LRtail-theta; % gpfit assumes theta known, gives K and sigma;
[parmhat,parmci] = gpfit(Xgp); %  Works for POT >= -0.2, else
```

```
% Warning: Else Max likelihood has converged to a boundary point;
K = parmhat(1); sigma = parmhat(2);
fprintf('\ngpfit: K = %7.4f; KCI = [%7.4f,%7.4f];',K,parmci(1:2,1));
fprintf('\ngpfit: sigma = %7.4f; sigmaCI = [%7.4f,%7.4f];'...
    ,sigma,parmci(1:2,2));
[muhat,expci] = expfit(Xgp); %  Works for POT >= -0.2, else
% Warning: Else Max likelihood has converged to a boundary point;
fprintf('\nexpfit: muhat = %7.4f; expCI = [%7.4f,%7.4f];' ...
    ,muhat,expci(1:2,1));
%
figure(1);
nbins = 12;
hist(Xgp,nbins);  hold on;
xmin = min(Xgp); xmax = max(Xgp);
fprintf('\nnbins = %2i; xmin = %7.5f; xmax = %5.3f;',nbins,xmin,xmax);
X = xmin+(xmax-xmin)*(0:10)/10;
fgp = gppdf(X,K,sigma,0); % theta_arg=0, since in X.
Ns = max(fgp)/10; % normalize to hist height.
fprintf('\nFit Rescaling: Ns = %7.5f;',Ns);
plot(X,fgp/Ns,'-r','LineWidth',3); axis tight; hold on;
fexp = exppdf(X,muhat); %
plot(X,fexp/Ns,'--k','LineWidth',3); axis tight; hold off;
title('LR-Tail & fit for 2008 S&P Data, POT=-0.02;','Fontsize',24,...
    'FontWeight','Bold');
```

```
xlabel('LRtail-|POT|','Fontsize',24,'FontWeight','Bold');
ylabel('Frequency','Fontsize',24,'FontWeight','Bold');
legend('LRtail-|POT| Histogram','gppdf/Ns','exppdf/Ns');
set(gca,'Fontsize',18,'FontWeight','Bold');
fprintf('\n');

==========OUTPUT ====================
ttailfitspc2008.m Output for Log-Returns, (22-Jan-2010 16:47:29):
minLR = -0.10957; maxLR = 0.095;
POT = -0.02000; Npot =  31;
LRtail=0.024;0.050;0.038;0.036;0.026;0.039;0.035;0.063;0.061; ...
theta =  0.0200;
gpfit: K =  0.0153; KCI = [-0.3655, 0.3960];
gpfit: sigma =  0.0205; sigmaCI = [ 0.0122, 0.0345];
expfit: muhat =  0.0208; expCI = [ 0.0151, 0.0307];
nbins = 12; xmin = 0.00030; xmax = 0.090;
Fit Rescaling: Ns = 4.79952;
>>
```

- ***Exponential Analysis of LR Left-Tail POT Data:***

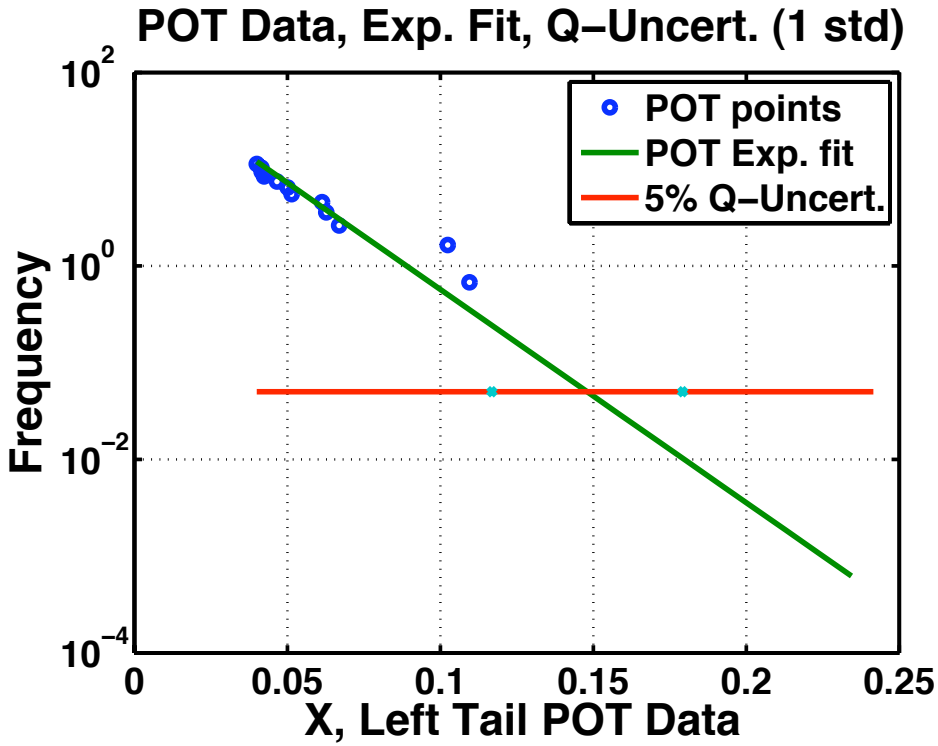


Figure 4.7: **Semi-Log plot of exponential fit** to S&P 500 log-return left-tail POT data (sign reversed) using modified `expan.m` code from MATLAB Central File Exchange. Note that near center points (left) fit the ***exponential*** very well whereas the two largest jumps are not as close to the fit, but are within **Q-uncertainty** of $\alpha = 0.05$ or **5%**.

MATLAB Code for Fast Exponential Analysis Fit 0f Left-Tail POT Data, with semi-log function of y `semilogy`*, so log(exp) fit* $= -kx$*:*

```
function tailexpanspc2008
% Get Tail Exponential (expan.m) Fit on Left (POT) for Log-Returns
% for 2008 S&P500 ^GSPC (Yahoo Finance) Data;
%   Dates 2007/12/31-2008/12/31, Daily Adjusted Closings.
clc
load -ASCII S08.mat; % Note: load function used for 2008 data;
fprintf('\ntailexpanspc2008.m Output for Log-Returns, 1/16/2009:');
L = length(S08); NLR = L-1;
tp = 1; p = 0.05;% One year's collection time, p = alpha = 5\%.
LR = log(S08(2:L))-log(S08(1:L-1)); % Note: Vector Log Difference!
minLR = min(LR); maxLR = max(LR);
fprintf('\nminLR = %7.5f; maxLR = %5.3f;',minLR,maxLR);
POT = -0.04; % "guestimate" Peak Over Threshold from Q-Q plot
LRtail = -LR(LR<=POT); % (LR<=POT)=indicator;
Npot = length(LRtail);
fprintf('\nPOT = %7.5f; Npot = %3i;',POT,Npot)
%
figure(1); % expan figure with POT data, exponential fit, uncertainty
%       bounds by crosses around the extrapolation to the p-value
fprintf('\nexpan Input: tp = %3.1f years; alpha = %4.2f',tp,p);
[xp,sp,mu] = expan(LRtail,tp,p); %[xp,sp] = expan(X,tp,p); Add exp. mean
fprintf('\nexpan Output: xp = %5.3f Q; sp = %4.2f uncert.;',xp,sp);
```

```
fprintf('\nexpdist Output: mean mu=%6.4f; rate lambda=%5.1f;',mu,1/mu);
fprintf('\n');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [xp,sp,l] = expan(X,tp,p)
% expan    Performs a fast exponential analysis on the POT dataset
%              stored in array X
% INPUT:
%       X: array or scalar values with POT data (peaks over threshold)
%       tp:  time period over which the data is collected
%       p: probability for which quantile has to be calculated
% OUTPUT:
%       xp: quantile value corresponding to exceedance probability p
%       sp: uncertainty in quantile value expressed as 1 standard dev.
%       Figure, showing the POT data, exponential fit, and uncertainty
%       bounds by crosses around the extrapolation to the p-value
% EXAMPLE:
%       Dataset e, generated from an exponential dist. with scale 1:
%       for i=1:10, e(i)=2-log(rand(1)); end
%       Assume the data are the peaks above 2 (meters) measured during
%       100 years.  Perform exponential analysis for the 10^-4 quantile
%       with:  expan(e,100,10^(-4))
%
% Author: P.H.A.J.M. van Gelder
% eMail: p.vangelder@ct.tudelft.nl
% Website:
```

```
%    http://www.hydraulicengineering.tudelft.nl/public/gelder/homepg.htm
% $Revision: 1.0 $ $Date: 2004/08/28 $; FINM Revision 2009/01/23 FBH
% Source FBH: http://www.mathworks.com/matlabcentral/fileexchange/5808
% ************************************************************
%
n=length(X);
l=mean(X-min(X));
ratio=n/tp;
xp=min(X)-l*log(p/ratio);
sp=-log(p/ratio)*l/sqrt(n);
xx=min(X):(xp-min(X))/10:xp+3*sp;
Fx=1-exp(-(xx-min(X))/l);
%
semilogy(sort(X),ratio*(1-([1:n]-0.3)/(n+0.4)),'o'...
    ,xx,ratio*(1-Fx),[min(X),xp+3*sp]...
    ,[p,p],[xp-sp,xp+sp],[p,p],'x','LineWidth',3);
title('POT Data, Exp. Fit, Q-Uncert. (1 std)'...
    ,'Fontsize',24,'FontWeight','Bold');
xlabel('X, Left Tail POT Data','Fontsize',24,'FontWeight','Bold');
ylabel('Frequency','Fontsize',24,'FontWeight','Bold');
legend('POT points','POT Exp. fit','5% Q-Uncert.');
set(gca,'Fontsize',20,'FontWeight','Bold','LineWidth',2);
grid
```

```
==================Output====================

tailexpanspc2008.m Output for Log-Returns, 1/16/2009:
minLR = -0.10957; maxLR = 0.095;
POT = -0.04000; Npot =  12;
expan Input: tp = 1.0 years; alpha = 0.05
expan Output: xp = 0.148 Q; sp = 0.03 uncert.;
expdist Output: mean mu = 0.0197; rate lambda =  50.7;
>>
```

4.5 Bivariate ($m = 2$ dimensions) and Multivariate ($m \geq 2$ dimensions) Distributions[a]*:*

- ***Bivariate Distributions:***

In this case, our observations are n sets of 2-tuples or 2-vectors of **observations** in two variables x and y,

$$\mathbf{x} = [(x_i, y_i)]_{n\times 1} = [\vec{x}_i]_{n\times 1}, \tag{4.21}$$

in effect at $n \times 2$ array. The corresponding two-dimensional random variables have the same form:

$$\mathbf{X} = [(X_i, Y_i)]_{n\times 1} = [\vec{X}_i]_{n\times 1}. \tag{4.22}$$

The bivariate distribution is a joint distribution defined by a joint probability,

$$F_{\vec{X}}(\vec{x}) \equiv F_{X,Y}(x, y) \equiv \mathbf{Prob}[X \leq x, Y \leq y] \tag{4.23}$$

and if a joint density exists, we usually assume it will except in pathological situations, then

$$F_{X,Y}(x, y) = \int_{-\infty}^{x}\int_{-\infty}^{y} f_{X,Y}(x', y')dx'dy'. \tag{4.24}$$

[a] See also Carmona ('04), Chapter 2.

It follows from taking two partial derivatives, one with respect to x and one with respect to y, that

$$f_{X,Y}(x,y) = \frac{\partial^2 F_{X,Y}}{\partial x \partial y}(x,y). \tag{4.25}$$

Note, for students uncomfortable with partial derivatives, that we only take one partial derivative with respect to one variable at a time while holding the other fixed as if calculating an ordinary derivative, i.e., first,

$$\frac{\partial}{\partial x}\left(F_{X,Y}|_{y\ fixed}\right)(x,y) = \int_{-\infty}^{y} f_{X,Y}(x,y')dy', \tag{4.26}$$

using the fundamental theorem of calculus and second,

$$\frac{\partial}{\partial y}\left(\frac{\partial F_{X,Y}}{\partial x}\bigg|_{x\ fixed}\right)(x,y) = f_{X,Y}(x,y), \tag{4.27}$$

again using the fundamental theorem of calculus. Note also that the evaluation at (x,y) is always logically done after the differentiation, otherwise could lead to errors, e.g., $\partial(F_{X,Y}(1,2))/\partial x \equiv 0$.

Sometimes we need just the density with regard to only one of the two variables and these are called **marginal densities**, so

$$f_X(x) \equiv \int_{-\infty}^{+\infty} f_{X,Y}(x, y')dy',$$
$$f_Y(y) \equiv \int_{-\infty}^{+\infty} f_{X,Y}(x', y)dx'. \quad (4.28)$$

• ***Correlation Coefficients and Covariances:***
Much of the random variables that we have considered so far were assumed to be independence, but that may not be strictly true in financial markets due to large and rapid electronic trading. Important announcements can trigger **herd behavior**. So it would be important to have **measures of interdependence**.

The **correlation coefficient between RVs X and Y** is defined as the volatility normalized, hence dimensionless, covariance,

$$\rho_{X,Y} \equiv \frac{\mathrm{Cov}[X,Y]}{\sigma_X \sigma_Y}, \tag{4.29}$$

which is also called **Pearson's correlation coefficient**. The **covariance** between X and Y is defined as

$$\mathrm{Cov}[X,Y] \equiv \sigma_{X,Y} \equiv \mathrm{E}[(X-\mu_X)(Y-\mu_Y)] = E[XY] - \mu_X \mu_Y, \tag{4.30}$$

where $\mu_X = \mathrm{E}_X[X]$ and $\mu_Y = \mathrm{E}_Y[Y]$ are the respective means of X and Y. The standard deviations or volatilities are $\sigma_X = \sqrt{\mathrm{E}_X[(X-\mu_X)^2]}$ and $\sigma_Y = \sqrt{\mathrm{E}_Y[(Y-\mu_Y)^2]}$.

• ***Sample Correlation Coefficients and Related Sample Moments:***

Recall the unbiased sample means,

$$(\overline{x}, \overline{y}) = (\widehat{\mu}_X, \widehat{\mu}_Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i, y_i), \tag{4.31}$$

and the sample (biased) variances,

$$(\widehat{\sigma}^2_X, \widehat{\sigma}^2_Y) = \frac{1}{n}\sum_{i=1}^{n}((x_i - \overline{x})^2, (y_i - \overline{y})^2), \tag{4.32}$$

standard deviations or volatilities the square roots of the corresponding quantities. Whereas, the sample covariances are

$$\widehat{\mathrm{Cov}}_{X,Y} = \widehat{\sigma}_{X,Y} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) \tag{4.33}$$

and the sample correlation coefficient is

$$\widehat{\rho}_{X,Y} = \frac{\widehat{\mathrm{Cov}}_{X,Y}}{\widehat{\sigma}_X \widehat{\sigma}_Y} = \frac{\widehat{\sigma}_{X,Y}}{\widehat{\sigma}_X \widehat{\sigma}_Y}. \tag{4.34}$$

The correlation coefficients $\rho_{X,Y}$ and $\widehat{\rho}_{X,Y}$ are bounded in $[-1,+1]$:

* If $\rho_{X,Y} = \pm 1$, then $Y = \pm\sigma_Y X/\sigma_X + \alpha$[a], an affine function of X.

* If X and Y are **independent**, then $\rho_{X,Y} = 0$, but the converse is not true, in general.

---

[a] Hanson (2007), Online Appendix B, Th. B.59.

• ***Multivariate Normal Density:*** For jointly distributed normal RVs, $\vec{X} = [X_i]_{m\times 1}$, then the multivariate normal density can be written in compact form through linear algebra,

$$f_{\vec{X}}^{(n)}(\vec{x}) = \frac{1}{\sqrt{(2\pi)^m(\det)[\Sigma]}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^\top \Sigma^{-1}(\vec{x}-\vec{\mu})\right), \quad (4.35)$$

where $\vec{\mu} = \mathbf{E}[\vec{X}] = [\mu_i]_{m\times 1}$ is the $m\times 1$ **mean** and

$$\Sigma = \mathbf{Cov}[\vec{X}, \vec{X}^\top] = [\sigma_{i,j}]_{m\times m} \quad (4.36)$$

is the $m \times m$ **covariance matrix**, with determinant $\mathbf{Det}[\Sigma]$ and $\sigma_{i,i} \equiv \sigma_i^2$ are the diagonal variance terms for $i = 1:m$. For notational simplicity, we say $\vec{X} \overset{\mathbf{dist}}{=} \mathcal{N}(\vec{\mu}, \Sigma)$ for $\vec{X}$ is normally distributed with mean $\vec{\mu}$ and covariance matrix Σ. By linear transformation this can be decomposed into **multivariate standard normal form**[a], $\vec{X} = \vec{\mu} + \sqrt{\Sigma}\vec{Z}$ where $\vec{Z} \overset{\mathbf{dist}}{=} \mathcal{N}(\vec{0}, I_m)$ is the standard multivariate random variable, I_m being the $m\times m$ identity matrix.

[a] Carmona (2004), Chapter 1, Appendix 1, p. 92ff.

• ***Independent Normal Multivariate Random Variables:***

If the $\vec{X}$ are **pairwise independent** then the distribution and density by definition of independence must be separable, so $\mathbf{\Sigma} = V_m \equiv [\sigma_i^2 \delta_{i,j}]_{m\times m}$ where V_m is a diagonal matrix with the individual variances along the diagonal and $\delta_{i,j}$ is the **Kronecker** delta, 1 if $j = i$ and otherwise 0. The multivariate normal distribution takes the form,

$$f_{\vec{X}}^{(n)}(\vec{x}) = \prod_{i=1}^{m} f_{X_i}^{(n)}(x_i) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right). \quad (4.37)$$

• ***Normal Bivariate Density Example:***[a]

The **bivariate normal distribution**, i.e., the two-dimensional case, needs several conditions to keep the density well-defined: $\sigma_i > 0$ for $= 1:2$, $\sigma_{1,2} = \rho\sigma_1\sigma_2$, where $\rho = \rho_{1,2}$ is the correlation coefficient between state 1 and state 2 such that $-1 < \rho < +1$. Thus,

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}, \tag{4.38}$$

$$\Sigma^{-1} = \frac{1}{1-\rho^2} \begin{bmatrix} 1/\sigma_1^2 & -\rho/(\sigma_1\sigma_2) \\ -\rho/(\sigma_1\sigma_2) & 1/\sigma_2^2 \end{bmatrix}. \tag{4.39}$$

The Σ^{-1} follows upon calculating the two-dimensional inverse of Σ, while substituting for Σ^{-1} and

$$\mathrm{Det}[\Sigma] = (1-\rho^2)\sigma_1^2\sigma_2^2, \tag{4.40}$$

[a]This Section from Hanson (2007), Online Appendix B, pp. B.47ff.

Substitution yields the more explicit density form:

$$f_{\vec{X}}^{(n)}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \vec{\mu}, \Sigma\right) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{0.5}{1-\rho^2}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - \frac{2\rho(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2} + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]\right). \tag{4.41}$$

Remark: *The bivariate normal density becomes* ***singular*** *when* $\sigma_1 \longrightarrow 0^+$ *or* $\sigma_2 \longrightarrow 0^+$ *or* $\rho^2 \longrightarrow 1^-$ *and the density becomes degenerate. If* $\rho > 0$, *then* X_1 *and* X_2 *are* ***positively correlated****, while if* $\rho < 0$, *then* X_1 *and* X_2 *are* ***negatively correlated****.*

Some of the first few moments are tabulated (results from the Maple symbolic computation system) in Table 1[a].

Table 1: ***Some expected moments of bivariate normal distribution.***

Some Binormal Expectations
$\mathbf{E}[1]=1$
$\mathbf{E}[x_i]=\mu_i,\ i=1:2$
$\mathbf{Var}[x_i]=\sigma_i^2,\ i=1:2$
$\mathbf{Cov}[x_1,x_2]=\rho\sigma_1\sigma_2$
$\mathbf{E}[(x_i-\mu_i)^3]=0,\ i=1:2$
$\mathbf{E}[(x_i-\mu_i)^4]=3\sigma_i^4,\ i=1:2$
$\mathbf{E}[(x_1-\mu_1)^2(x_2-\mu_2)^2]=(1+2\rho^2)\sigma_1^2\sigma_2^2$

[a]This Table from Hanson (2007), Online Appendix B, p. B.48.

4.6 Bivariate Graphical Analysis:

• *Bivariate Histograms:*

The *MATLAB Stat Toolbox* bivariate histogram `hist3([X,Y], [nbin1,nbins2]);` can be used for comparing two data column-vectors `X` and `Y`, with user selected bin sizes, `nbin1` and `nbins2`. This gives a projected three dimensional qualitative graphical display of the density on a rectangular grids. It also comes in several forms, letting `XY=[X,Y]`, such as `hist3(XY);` displays on in 3-dimensions on a default 10×10 grid. Also, there is `hist3(XY,centers);` that allows the user to select bin centers with array `centers=[xcenters,ycenters]` where `xcenters` and `ycenters` are the center vectors.

● ***Bivariate Histogram (`hist3`) Merging 2008 Log-Returns and Reference Normal Simulations:***

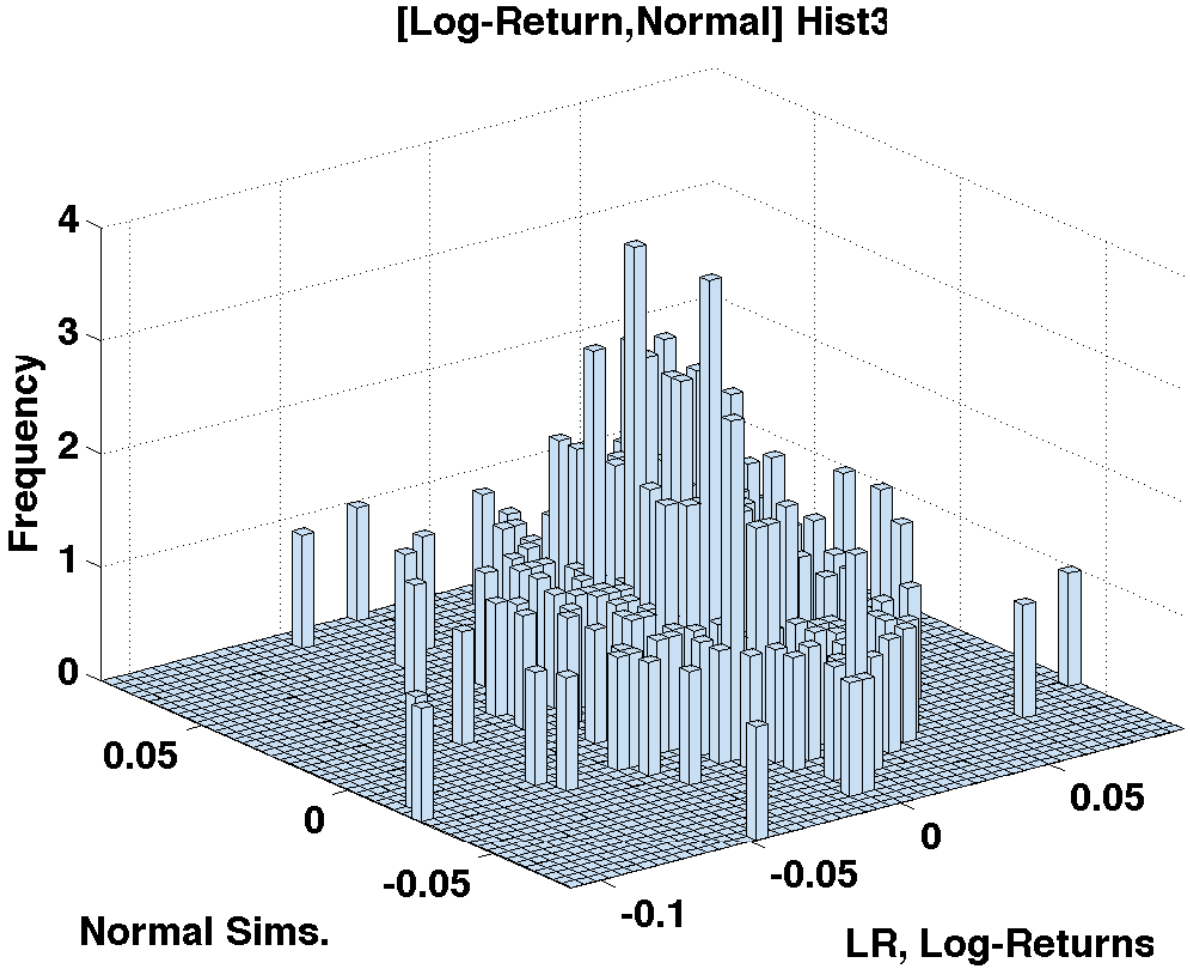
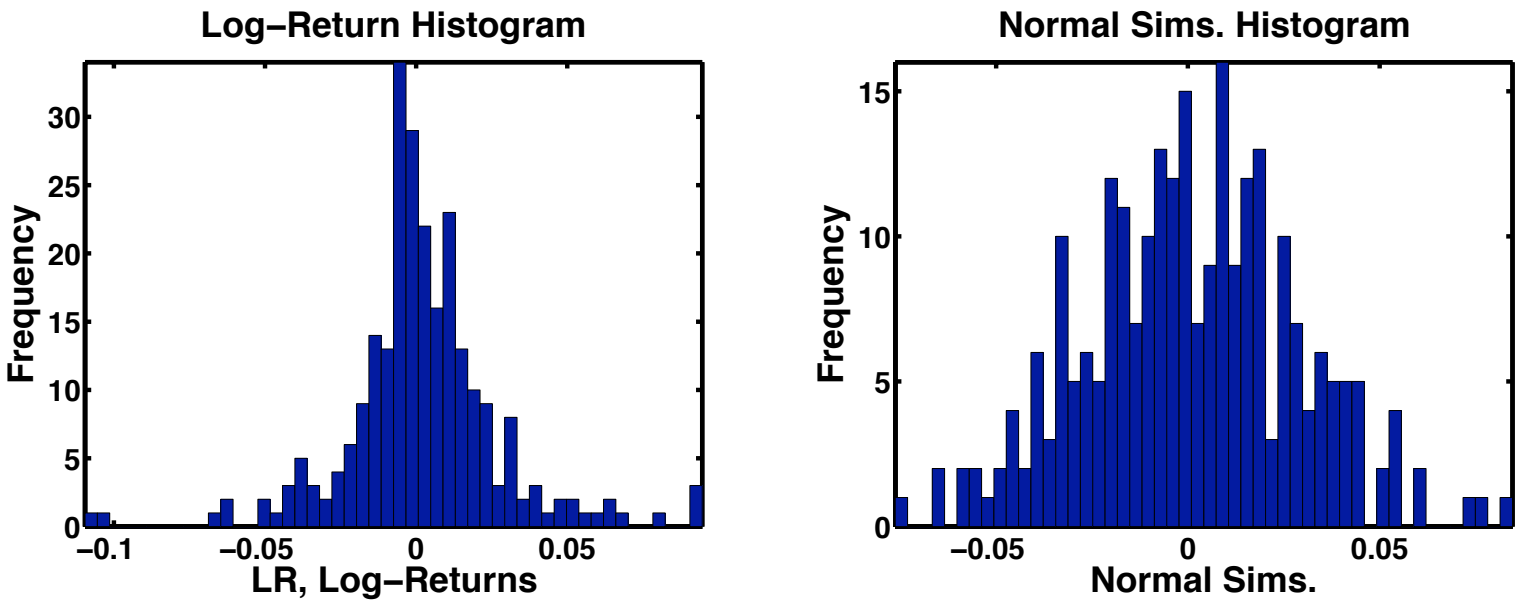


Figure 4.8: Bivariate plot of the bivariate histogram of the 2008 S&P500 Index log-returns merged with the reference normal simulations.



(a) Reprinted 2008 Log-Returns.

(b) Reprinted Reference Normal.

Figure 4.9: Histograms of previous 2008 Log-Returns and the Reference Normal Simulations to show the components that are combined to make the bivariate histogram by **`hist3`** in Figure 4.10.

• *Bivariate Histogram (`hist3`) Merging 2008 Log-Return Data and Reference Normal Simulations MATLAB Code:*

```
function hist3kde2spc2008normal
% Get Bivariate Histogram for Log-Returns Density
%     Similar Normal Simulation.
% for 2008 S&P500 ^GSPC (Yahoo Finance) Data;
%   Dates 2007/12/31-2008/12/31, Daily Adjusted Closings.
clc
load -ASCII S08.mat; % Note: load function used for 2800 data;
fprintf('\nhist3kde2spc2008normal Output (%s):',datestr(now));
L = length(S08); fprintf('\nsizeS = [%3i,%3i];',size(S08));
LR = log(S08(2:L))-log(S08(1:L-1)); % Note: Vector Log Difference!
NLR = L-1; sizeLR = size(LR); % size=NLR X 1
meanLR = mean(LR); stdLR = std(LR);
fprintf('\nmeanLR = %7.5f; stdLR = %5.3f;',meanLR,stdLR);
figure(1);
nb = 50;
Ynorm = normrnd(meanLR,stdLR,NLR,1);
sizeY = size(Ynorm);
fprintf('\nsizeLR = [%3i,%3i]; sizeY = [%3i,%3i];',sizeLR,sizeY);
XY = [LR,Ynorm]; % Caution: both must be column vectors of same size.
nbins = [nb,nb];
hist3(XY,nbins); axis tight;
title('[Log-Return,Normal] Hist3'...;
```

```
    ,'Fontsize',16,'FontWeight','Bold');
xlabel('LR, Log-Returns','Fontsize',16,'FontWeight','Bold');
ylabel('Normal Sims.','Fontsize',16,'FontWeight','Bold');
zlabel('Frequency','Fontsize',16,'FontWeight','Bold');
set(gca,'Fontsize',16,'FontWeight','Bold');
figure(2)
hist(LR,nb); axis tight;
title('Log-Return Histogram','Fontsize',24,'FontWeight','Bold');
xlabel('LR, Log-Returns','Fontsize',24,'FontWeight','Bold');
ylabel('Frequency','Fontsize',24,'FontWeight','Bold');
set(gca,'Fontsize',20,'FontWeight','Bold','LineWidth',3);
figure(3)
hist(Ynorm,nb); axis tight;
title('Normal Sims. Histogram','Fontsize',24,'FontWeight','Bold');
xlabel('Normal Sims.','Fontsize',24,'FontWeight','Bold');
ylabel('Frequency','Fontsize',24,'FontWeight','Bold');
set(gca,'Fontsize',20,'FontWeight','Bold','LineWidth',3);
%
%
%
%
%
%
%
```

```
figure(4);
[bandwidth,density,X,Y]=kde2d(XY);
%   % plot the data and the density estimate
contour3(X,Y,density,50), hold on;
plot(XY(:,1),XY(:,2),'r.','MarkerSize',5);
title('[Log-Ret.,Normal] kde2d','Fontsize',16,'FontWeight','Bold');
xlabel('LR, Log-Returns','Fontsize',16,'FontWeight','Bold');
ylabel('Normal Sims.','Fontsize',16,'FontWeight','Bold');
zlabel('Frequency','Fontsize',16,'FontWeight','Bold');
set(gca,'Fontsize',16,'FontWeight','Bold');
fprintf('\n');
```

- ***Bivariate Kernel Smoothing Density Estimator:***

The **bivariate kernel density estimator** is similar to that of the univariate case except for the extra dimension and the corresponding extra scaling so letting the scaled coordinates be $(\widetilde{x},\widetilde{y})=((x-x_i)/x_{bw},(y-y_i)/y_{bw})$ with positive bandwidth vector (x_{bw},y_{bw}) and change of variables $d\widetilde{x}d\widetilde{y}=dxdy/(x_{bw}y_{bw})$, a common $dxdy$ canceling out between original and tranformed densities, i.e., $f_{X,Y}(x,y)=f_{\widetilde{X},\widetilde{Y}}(\widetilde{x},\widetilde{y};x_{bw},y_{bw})/(x_{bw}y_{bw})$. The **sample estimator** with **nonnegative kernel** K is

$$\widehat{f}_{X,Y}(x,y;x_{bw},y_{bw})=\frac{1}{nx_{bw}y_{bw}}\sum_{i=1}^{n}K\left(\frac{x-x_i}{x_{bw}},\frac{y-y_i}{y_{bw}}\right). \quad (4.42)$$

If the variables (x,y) are similar in physical dimensions and other attributes, the bandwidth could be the same, i.e., $x_{bw}=y_{bw}$ then very simplified form (Carmona's (2004)) could be used,

$$\widehat{f}_{X,Y}(x,y;x_{bw},x_{bw})=\frac{1}{nx_{bw}^2}\sum_{i=1}^{n}K\left(\frac{(x-x_i,y-y_i)}{x_{bw}}\right). \quad (4.43)$$

The `[fs,xs]=ksdensity(X)`, from the **`Statistical Toolbox`** used in the univariate case, **ONLY takes single Vector data `X`**, so cannot be used in the bivariate case.

However, **Zdravko I. Botev**, the University of Queensland in Australia, has created a nice bivariate kernel density estimated that does assume normal or Gaussian data or mixtures of Gaussian data and is named **`kde2d.m`** that contains very clear instructions about the input, output and several very good examples, infact better than that of **`ksdensity`**. It can be found at

Bivariate Kernel Density Estimation Code Webpage

on the **Mathworks Matlab Central File public domain directory**. Included in the preface comments of **`kde2d.m`** is Botev's technical report on the univariate version,

A Novel Nonparametric Density Estimator.

• *Bivariate Kernel Smoothing (`kde2d`) of 2008 Log-Returns and Normal Simulations:*

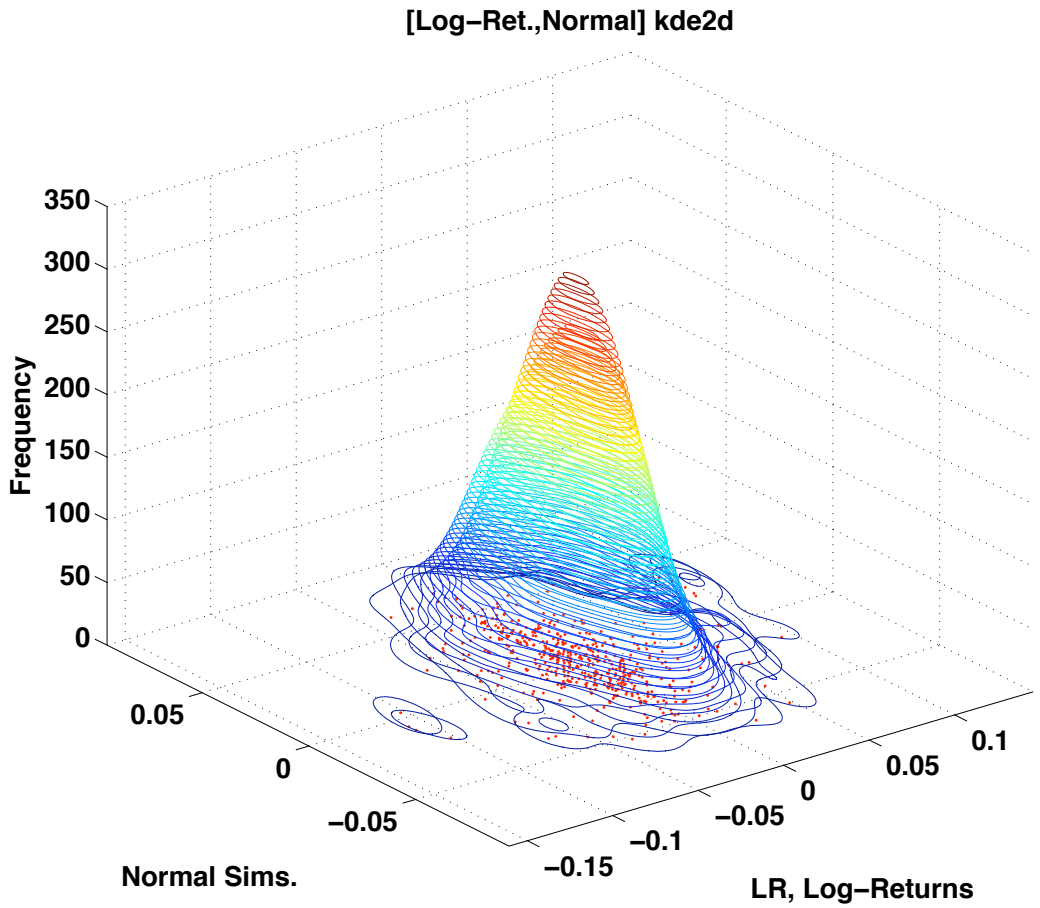


Figure 4.10: Bivariate plot of the kernel smoothed bivariate density of the 2008 S&P500 Index log-returns together with the reference normal simulations. Extreme jumps are presented by the small islands.

4.7 Multivariate Sample Statistics and Principal Component Analysis (PCA):[a]

As the **number of variates or states**, say variables x_j for $j = 1:m$, gets large and the **number of observations per variate**, say observations $X_{i,j}$ for $i = 1:n_j$, becomes large, some times large with m, then we may need to find some way to **reduce the dimension** to $\widehat{m} < m$ by looking for some combinations of variates that are most important for the problem, i.e., the **principal components** contributing the most variance. Often the techniques involve finding appropriate orthogonal transformations for the reduction. In order to keep things simple, it will be assumed the number of observations per variate are constant or $n_j = n,\ \forall j$ with $X_{i,j}$ data observations.[b]

---

[a] See Carmona (2004), p. 84ff for PCA or these notes for clarification.

[b] Note that we have a choice of since $\mathcal{X} = [X_{i,j}]_{n\times m}$ or its transpose $\mathcal{X}^\top = [X_{j,i}]_{m\times n}$, but having former with the jth state sample data in the jth column is consistent with prior use in univariate sample in a column.

• ***Curse of Dimensionality with Computational Complexity:***

The computational statistic problem is not just due to the total data dimension, $\mathcal{X} = [X_{i,j}]_{n\times m}$, of $n \cdot m$. Depending on the application, we will be looking at the data array $\mathcal{X}$ in at least two ways, one as an observation vector $\vec{X}_j^{(o)} = [X_{i,j}]_{n\times 1}$ for fixed j or as a variate vector $\vec{X}_i^{(v)} = [X_{i,j}]_{1\times m}$ for fixed i. The real **curse of dimensionality** appears as large scale computational complexity in many multivariable problems due to functions and integrals that arise. In computational statistics, we usually have to consider the **total frequencies of all observations**,

$$f_{\mathcal{X}} = f_X(\mathcal{X}) = f_X\left(\left[\vec{x}_i^{(v)}\right]_{n\times 1}\right), \tag{4.44}$$

so depending on how we look at all this information, we either have m^n or n^m for the total amount of information and usually the former $m^n = \exp(n\log(m))$ is the most serious since usually $n >> m$. This **exponential computational complexity** is called

The Curse of Dimensionality![a]

[a] ***For a general computational or numerical account, see Hanson (Book '07), p. 230-231 or Hanson (CDC03), p. 5, http://www.math.uic.edu/ hanson/pub/CDC03/cdc03web.pdf.***

• ***Multivariate Sample Means:***

The sample mean depends on how the variables and observations are interpreted, so for example, the sample mean could be the average score for one student i with respect to all *for* $j = 1:m$ questions or variables (v), so

$$\overline{X}_i^{(v)} \equiv \frac{1}{m} \sum_{j=1}^{m} X_{i,j}, \tag{4.45}$$

or the sample mean could be average of all observations (o), e.g., *for* $i = 1:n$ *students*, with average grade on question j,

$$\overline{X}_j^{(o)} \equiv \frac{1}{n} \sum_{i=1}^{n} X_{i,j}. \tag{4.46}$$

Often in **PCA** we are looking at variances or covariances and other quadratic forms, in general, so it is convenient to use **normalized linear combinations (NLCs)** such as

$$\widetilde{\overline{X}}_i^{(v)} \equiv \frac{1}{\sqrt{m}} \sum_{j=1}^{m} X_{i,j}. \tag{4.47}$$

• ***Variance-Covariance Matrices:***

0em The **estimated, observational covariance matrix** is defined by the elements,

$$\widehat{C}^{(o)}_{x,j,j'} \equiv \widehat{C}_x\left[\vec{X}^{(o)}_j, \vec{X}^{(o)}_{j'}\right] = \frac{1}{n}\sum_{k=1}^{n}\left(X_{k,j} - \overline{X}^{(o)}_j\right)\left(X_{k,j'} - \overline{X}^{(o)}_{j'}\right), \quad (4.48)$$

for $j, j' = 1:m$ and estimated covariance matrix $\widehat{C}^{(o)}_x \equiv \left[\widehat{C}^{(o)}_{x,j,j'}\right]_{m\times m}$ and **estimated (biased) variances** $\widehat{\sigma}^{(o)}_j \equiv \widehat{C}^{(o)}_{x,j,j}$. Clearly, this matrix is **symmetric** since the transpose is $\left(\widehat{C}^{(o)}_x\right)^\top = \widehat{C}^{(o)}_x$ with transposed element $\widehat{C}^{(o)\top}_{x,j,j'} \equiv \widehat{C}^{(o)}_{x,j',j} = \widehat{C}^{(o)}_{x,j,j'}$. Also, the matrix is **non-negative definite (NND)**, $\widehat{C}^{(o)}_x \geq 0$ by the **Cauchy-Schwarz inequality**, i.e., $|\vec{x}^\top\vec{y}| \leq |\vec{x}|\cdot|\vec{y}|$, for the same reason that the correlation coefficient is bounded,

$$|\widehat{\rho}_{j,j'}| = \frac{\left|\widehat{C}^{(o)}_{x,j,j'}\right|}{\widehat{\sigma}^{(o)}_j\widehat{\sigma}^{(o)}_{j'}} \leq 1, \quad \text{with } \widehat{C}^{(o)}_{x,j,j'} = \left(\frac{\delta\vec{X}^{(o)}_j}{\sqrt{n}}\right)^\top\left(\frac{\delta\vec{X}^{(o)}_j}{\sqrt{n}}\right), \quad (4.49)$$

in normalized form, provided $\widehat{\sigma}^{(o)}_j > 0$ and $\delta\vec{X}^{(o)}_j \equiv \left[X_{i,j} - \overline{X}^{(o)}_j\right]_{m\times 1}$.

• ***Covariance Eigenvalue (Characteristic Value) Problem:***

From the **NND property**, $\widehat{C}_x^{(o)} \geq 0$, it follows for the **eigenvalue problem (EVP)** if full rank $\mathcal{X}$, (else singular value decomposition (SVD)),

$$\widehat{C}_x^{(o)} \vec{C}_k = \lambda_k \vec{C}_k, \tag{4.50}$$

that the eigenvalues have the same property, $\lambda_k \geq 0$ for $k = 1:m$ (not to be confused with the Poisson jump rate) and can be sorted into descending order, taking λ_1 as the principal eigenvalue,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0, \tag{4.51}$$

with associated eigenvectors, $\vec{C}_k = [C_{i,k}]_{m\times 1}$, (called **loadings**), having the same index order, assumed normalized to unit modulus $\left|\vec{C}_k\right| = 1$ and **pairwise orthogonal** ("$\perp$" : $\vec{C}_i^{\top}\vec{C}_j = \delta_{i,j}$), subject to no multiple eigenvalues. If we load the eigenvectors into a matrix $C \equiv [C_{i,j}]_{m\times m}$ by columns and use the eigvalues to form a diagonal matrix $\Lambda = [\lambda_i \delta_{i,j}]_{m\times m}$ then we have the following **orthogonal decomposition** of the estimated covariance matrix,

$$\widehat{C}_x^{(o)} = C^{\top}\Lambda C, \tag{4.52}$$

modulo the sign of the eigenvectors C with regard to uniqueness.

- **MATLAB** ***Principal Component Analysis Functions:***

```
[pcacoeffs,score,latent] = princomp(X);
```

Given input of $n \times m$ observation data matrix **X**, with rows containing observed data and columns containing variables, **princomp** outputs:

1. $m \times m$ matrix **pcacoeffs** of PCA Coefficients where the jth column, $[\mathbf{pcacoeffs}(i,j)]_{m\times 1}$, contains the jth **nonstandardized principal component or loadings or eigenvectors** $\vec{C}_j$ in decreasing order of component variance σ_j;
2. $n \times m$ vector **score** of the **representation of X in principal component space** with rows corresponding to observations and columns to variables;
3. $m \times 1$ vector **latent** of **principal component variances or eigenvalues** $\vec{\lambda}$ of the covariance matrix **Cx** of the data matrix **X**.

If a fully standardized analysis is needed, then the data **X** should be replaced by the standardizing **zscore(X)** , scaled by the element-wise standard deviation.

(Another form for data input function besides **`princomp`**[a].)

`[pcacoeffs,latent,explained]=pcacov(C);`

Given input of $m \times m$ Covariance matrix **`Cx`**, **`pcacov`** outputs:

1. $m \times m$ matrix **`pcacoeffs`** of PCA Coefficients where the jth column, $[\texttt{pcacoeffs}(i,j)]_{m\times 1}$, contains the jth **nonstandardized principal component or loadings or eigenvectors** $\vec{C}_j$ in decreasing order of component variance σ_j;
2. $m \times 1$ vector **`latent`** of **principal component variances or eigenvalues** $\vec{\lambda}$ of **`Cx`**;
3. $m \times 1$ vector **`explained`** of the **percentage of the total variance explained** by each principal component.

---

[a]Carmona (2004), pp. 87-93, using the **`S-Plus princomp`** version, gives several financial examples of a sets of related assets that can be reduced to smaller sets of the most important assets in the relevant markets.

`[residuals,reconstructed]=pcares(X,npca);`

Given input of $n \times m$ observation data matrix **`X`**, with rows containing observed data and columns containing variables, and also given **`npca`** retained principal components of **`X`**, **`npca`**=$\widehat{\mathtt{m}} \leq$**`m`**, **`pcares`** outputs:

1. $n \times m$ residual matrix **`residuals`** of PCA Coefficients where the jth column, $[\mathtt{pcacoeffs}(i,j)]_{m\times 1}$, contains the jth **nonstandardized principal component or loadings or eigenvectors** $\vec{C}_j$ in decreasing order of component variance σ_j;
2. $n \times m$ optional vector **`reconstructed`** of the **reconstructed observations**, approximating PCA by the **`npca`** principal components;

If standardized analysis is needed, then the data **`X`** should be replaced by the **`standardized Z-scores`** form with **`zscore(X)`**, centering with respect to the mean and scaled by the element-wise standard deviation.

** Reminder: Lecture 4 Homework Posted in Chalk Assignments, due by Lecture 5 in Chalk Assignments!*

*** Summary of Lecture 3:**

1. **Cauchy Distribution**
2. **Sample CDFs**
3. **Order Statistics**
4. **Pareto Distribution and POTs in Tail Distributions**
5. **Multivariate and Bivariate Distributions**
6. **Bivariate Graphical Analysis**
7. **N-Dim. Sample Statistics & Principal Component Analysis**