

FinM 331/Stat 339 Financial Data Analysis

(Applied Statistical Analysis of Financial Data in MATLAB)

Winter 2009

Floyd B. Hanson, Visiting Professor

Email: t-9fhans@uchicago.edu

Master of Science in Financial Mathematics Program

University of Chicago

Lecture 4

6:30-9:30 pm, 26 January 2009, Ryerson 251 in Chicago

7:30-10:30 pm, 26 January 2009 at UBS Stamford

8:30-11:30 am, 27* January 2009, #02-01 Spring Singapore

*

Gong Xi Fa Cai!

Unfortunately conflicts with Second Day of the Chinese New Year holiday in Singapore, but this is an international program subject to many holidays. Sorry.

4. *More and Non-Normal Exploratory Data Analysis Tools:*

4.1 *Shortfall Statistics: Other Risk Measures Beyond VaR:*^a

- *Basel Accords on Financial Institutions:*

These are a series of agreements about the responsibilities of international financial institutions to control their exposures to risk (see Hull ('06) so-called "*Traders' Bible*" Chapter 18, for a short sidebar summary; also Wikipedia, but since it is usually anonymous, check further). "The first pillar (namely *Basel I*) deals with maintenance of regulatory capital calculated for three major components of risk that a bank faces: credit risk, operational risk and market risk. . . . *Basel II* uses a "three pillars" concept (1) minimum capital requirements (addressing risk), (2) supervisory review and (3) market discipline . . ." (Wikipedia, time ordered).

^aAs previously, this lecture will be a hybrid of Carmona's ('04) and Hanson's ('00-'09) financial data analysis, but with the former more clarified.

- ***Profit & Loss Or Loss & Profit Distributions:***

Since the portfolio value returns ΔV_i for $i = 1:n$ can range from positive to negative, their distribution is both a profit and loss (***P&L***) distribution. Since our main risk is the losses, let the random variable ***X*** represent the data $-\Delta V_i$ to emphasize the losses and let $F_X(x)$ be the distribution of X , the ***"Loss & Profit" distribution***(☺). Note that if the RV ***Y*** represents $+\Delta V_i$, then $Y = -X$ with distribution ***F_Y(y)***, so by the law for changing densities, say on (a, b) under changes of variables, conserving probability

$$1 = \int_a^b f_Y(y)dy = + \int_{-b}^{-a} f_X(x)dx,$$

which is equivalent to using the absolute value of the Jacobian of the transformation, i.e., $f_X(x) = f_Y(y)|dy/dx|$ and similarly for the distribution at least in the continuous case when we can assume the densities exist.

- **Shortfall Distribution:**

Again let α be the level of risk corresponding to the Value at Risk VaR_α , then the Shortfall distribution (SF) is defined as

$$F_X^{(\text{sf})}(x; \alpha) = \text{Prob}[X \leq x \mid X > \text{VaR}_\alpha],$$

where the form of the conditioning is explained by the original definition of VaR in loose terms using the data $X = -\Delta V_i$ loosely as an RV,

$$\begin{aligned} \alpha &\stackrel{\text{def}}{=} \text{Prob}[\Delta V_i < -\text{Var}_\alpha] = \text{Prob}[-\Delta V_i > \text{Var}_\alpha] \\ &= \text{Prob}[X > \text{Var}_\alpha]. \end{aligned}$$

However, the shortfall distribution, since conditional, is related to a conditional truncation of the "Loss & Profit" distribution $F_X(x)$ with renormalization to conservation of probability, i.e., in terms of the differential distribution (called measure in abstract, but $dF_X(x) = f_X(x)dx$ in usual practice),

$$dF_X^{(\text{sf})}(x; \alpha) = \left\{ \begin{array}{ll} \frac{dF_x(x | x > \text{VaR}_\alpha)}{\text{Prob}[X > \text{VaR}_\alpha]}, & x > \text{VaR}_\alpha \\ 0, & \textit{else} \end{array} \right\},$$

so, probability is conserved,

$$\begin{aligned} \int_{-\infty}^{+\infty} dF_X^{(\text{sf})}(x; \alpha) &= \int_{\text{VaR}_\alpha}^{+\infty} dF_X^{(\text{sf})}(x; \alpha) = \frac{1}{\alpha} \int_{\text{VaR}_\alpha}^{+\infty} dF_X(x) \\ &\equiv \frac{1}{\alpha} \text{Prob}[X > \text{VaR}_\alpha] = 1. \end{aligned}$$

Hence, the Expected Shortfall $\text{ES}(\alpha)$ is

$$\begin{aligned} \text{ES}(\alpha) &\equiv \mathbf{E}[X | X > \text{VaR}_\alpha] = \int_{-\infty}^{+\infty} x dF_X^{(\text{sf})}(x; \alpha) \\ &= \frac{1}{\alpha} \int_{\text{VaR}_\alpha}^{+\infty} x dF_X(x), \end{aligned}$$

completing a what should be a coherent explanation of Carmona's (p. 27) less than credible presentation, stripped below the essentials.

- **4.2 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*,

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

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.,

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

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 *pi* times the height at the mode, $1/(\pi b)$.

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

Since $x f_X^{(c)}(x; a, b) = O(1/x)$ but not sufficiently $O(1/x)$ as $|x| \rightarrow \infty$, the mean of a Cauchy RV is $\mu^{(c)} = \mathbf{E}[X^{(c)}] = \infty$ and the variance along with all other high moments are thus undefined with an infinite mean. 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,

$$\frac{f_X^{(n)}(x; 0, 1)}{f_X^{(c)}(x; 0, 1)} = \frac{\sqrt{\pi}(1+x^2)}{\sqrt{2}\exp(x^2/2)} \xrightarrow{\text{L'H}} \frac{\sqrt{2\pi}}{\exp(x^2/2)} = \sqrt{2\pi}e^{-x^2/2} \rightarrow 0^+,$$

confirming that exponentials of large arguments will beat out powers, in most cases.

Cauchy tail asymptotic power law: $f_X^{(c)}(x; a, b) \rightarrow \frac{b}{\pi \cdot x^2}$ as $|x| \rightarrow \infty$.

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

The pathology is not so serious for practical reasons, since the data range is finite, perhaps the order of a quadrillion dollars for a while and if we are just 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 $R \rightarrow \infty$. Hence, the CPV value of the Cauchy mean for the standard distribution $(a, b) = (0, 1)$ is

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

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: If $F_X(x)$ is invertible, and X is a random variable with distribution $F_X(x)$, then

$$F_X(x) \stackrel{\text{dist}}{=} F_U^{(u)}(u),$$

where $F_U^{(u)}(u)$ is the uniform random number generator.

”Sketch of Proof”: For all 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 $F_X(X)$ is also an RV, then

$$\begin{aligned} F_X(u) &\equiv \text{Prob}[F_X(X) \leq u] \stackrel{\text{incr.}}{=} \text{Prob}[X \leq F_X^{-1}(u)] \\ &\equiv F_X((F_X)^{-1}(u)) \stackrel[\text{inverse}]{\text{defn}}{=} u. \end{aligned}$$

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

- *Cauchy and Uniform Random Number Generators:*

Since letting

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

on page 6, it takes mostly algebra to invert, so solving for x in terms of u ,

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

and in MATLAB

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

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 work for the Normal Distribution due to that lack of a inverse in terms of elementary function, although one exists on principle.

The function **cauchyrnd** is essentially one that 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.

{Caution: The maximum likelihood m-file needs the Optimization Toolbox, but the toolbox does not mention it, but user will get error message if run.}

- **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:*

Cauchy vs Histogram from S&P Data

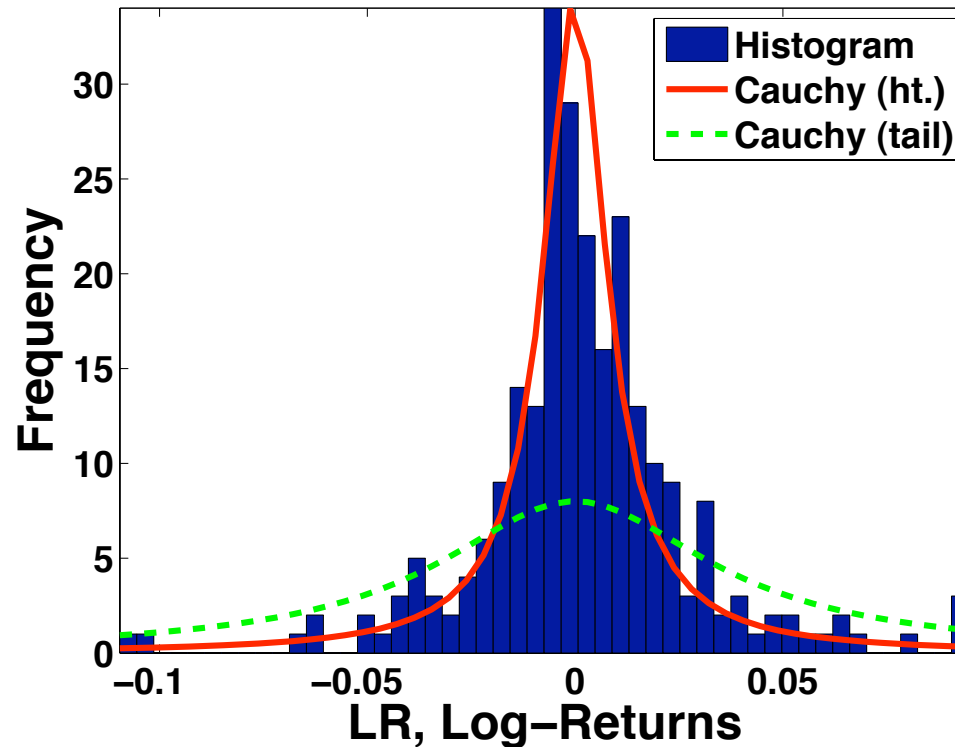


Figure 1: Combined plot of the histogram of the 2008 S&P500 Index log-returns along with two graphs using `cauchypdf` with `nbins` bin counts of the histogram and with either height (*red —*) or tail (*green - - -*) to histogram's, but do not fit in general (!).

- *Q-Q Plot of Cauchy (height-adjusted) Simulations versus 2008 S&P500 Log>Returns (**cauchyfit** could not be used to fit, yet):*

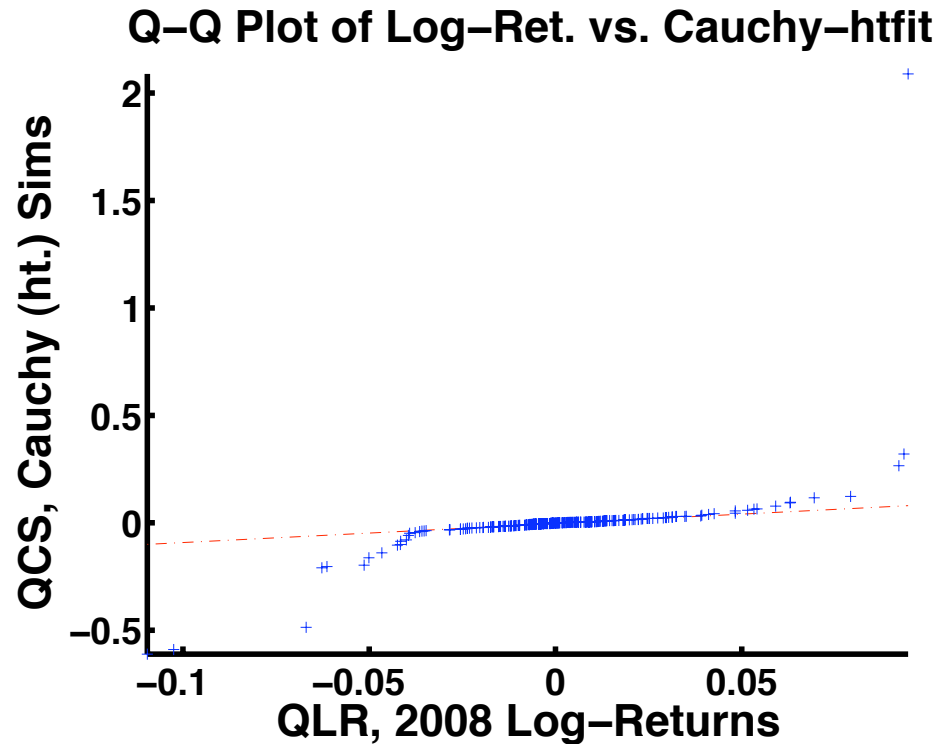


Figure 2: Q-Q Plot of the 2008 S&P500 Index log-returns versus the **cauchyrnd** RNG simulations for height-adjusted Cauchy distribution and the simulations are a good representation of the tails of the data, any large deviations probably due to Cauchy theoretical infinite domain.

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

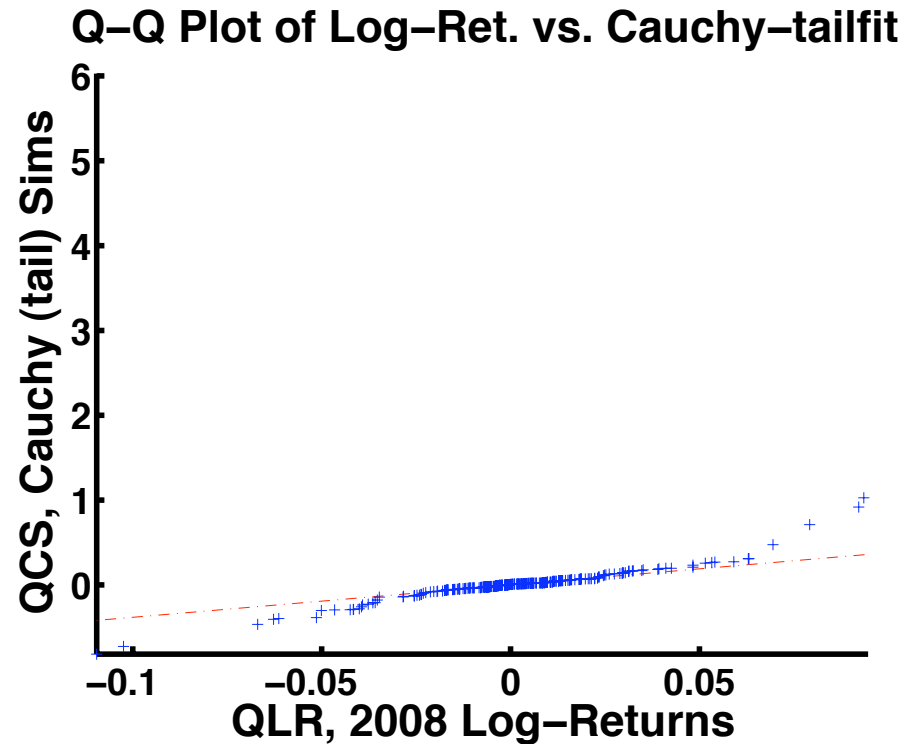


Figure 3: Q-Q Plot of the 2008 S&P500 Index log-returns versus the `cauchyrnd` RNG simulations for tail-adjusted Cauchy distribution and the simulations are a better representation of the central part of the data and tails, excluding Cauchy large domain deviations. (No `cauchyfit`; *Caution*: L3 `qqplot` axes mislabeling.)

- *Cauchy and Normal (both height-adjusted) PDF Comparison of Their Fat and Very Thin Tails:*

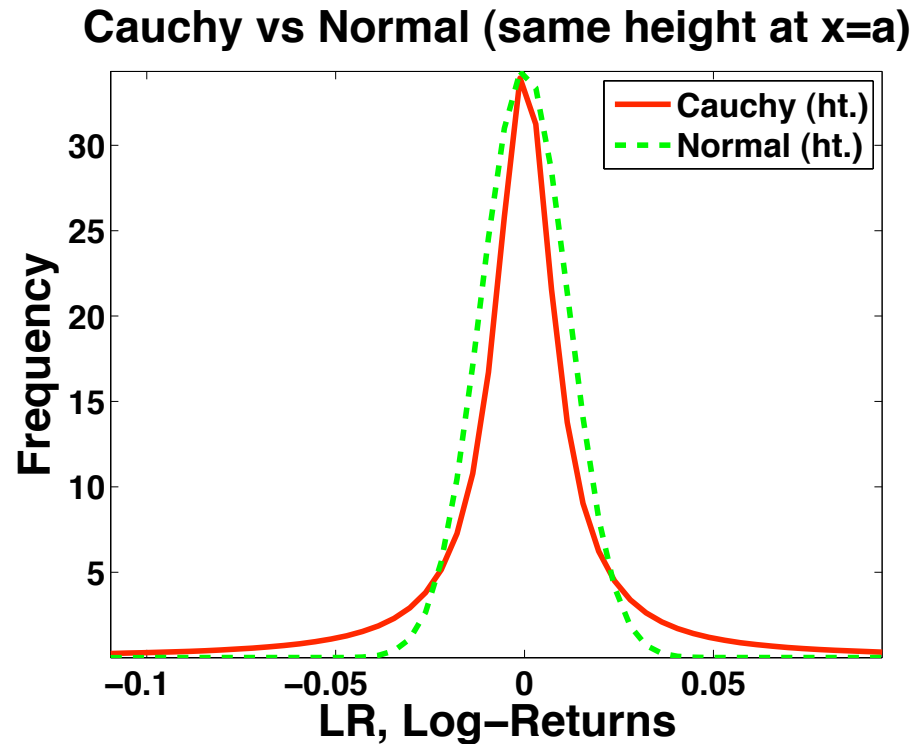


Figure 4: Comparison of Cauchy (`cauchypdf`) and Normal (`normpdf`) PDFs adjusted to same heights, using `nbins` bin counts of the histogram and with either Cauchy (*red —*) or Normal (*green - - -*), showing large difference in tail thickness.

- ***MATLAB Code for Two Cauchy PDFs with S&P500 Log-Return histogram, Two Q-Q Plots of Cauchy Simulations Against Log>Returns, and Cauchy-Normal (same height) 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 S.mat; % Note: Change GSPC2008adjC.txt name for load function.
fprintf('\nhistspc2008cauchy.m Output for Log>Returns, 1/16/2009:');
L = length(S);
LR = log(S(2:L))-log(S(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 guess 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 guess from histogram
fprintf('\ncauchy tail fit: a = %7.3e; b2 = %7.5f; h2 = %7.5f;', a, b2, h2);
figure(1);
nb = 50;
hist(LR, nb); hold on; % hold hist for adding;
dz = dLR/(nb-1); z = minLR + dz*(0:nb-1);
fzcl = cauchypdf(z, a, b1);
```

```

fzc2 = cauchypdf(z,a,b2);
plot(z,fzc1,'-r',z,fzc2,'--g','LineWidth',3); 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)');
set(gca,'FontSize',18,'FontWeight','Bold');
figure(2)
Xcsims1 = cauchyrnd(a,b1,1,NLR);
qqplot(LR,Xcsims1); axis tight;
title('Q-Q Plot of Log-Ret. vs. Cauchy-htfit'...;
      , 'FontSize',24,'FontWeight','Bold');
xlabel('QLR, 2008 Log>Returns','FontSize',24,'FontWeight','Bold');
ylabel('QCS, Cauchy (ht.) Sims','FontSize',24,'FontWeight','Bold');
set(gca,'FontSize',20,'FontWeight','Bold','LineWidth',3);
figure(3)
Xcsims2 = cauchyrnd(a,b2,1,NLR);
qqplot(LR,Xcsims2); axis tight;
title('Q-Q Plot of Log-Ret. vs. Cauchy-tailfit'...;
      , '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);

```

```

figure(4);
sigma = 1/(sqrt(2*pi)*h1); % same height h1 at x = a;
fprintf('\nnormal height fit: a = %7.3e; sigma = %7.5f; h1 = %7.5f;'...
    ,a,sigma,h1);
fznorm = normpdf(z,a,sigma);
plot(z,fzcl,'-r',z,fznorm,'--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.)','Normal (ht.)');
set(gca,'FontSize',18,'FontWeight','Bold');
fprintf('\n');

```

=====**Output**=====

```

histspc2008cauchy.m Output for Log>Returns, 1/16/2009:
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;
normal height fit: a = 0.000e+00; sigma = 0.01156; h1 = 34.50000;
>>

```

- **4.3 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 estimation by $\hat{F}_n(x) \simeq F_X(x)$ where

$$\hat{F}_n(x) \equiv \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\{x_i \leq x\}} = \frac{K(x)}{n},$$

where $\mathbf{1}_{\{x_i \leq x\}}$ is the indicator function for the set $\{x_i \leq x\}$ for each $i = 1:n$, one in the set and zero out of the set, so represent 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 (Carmona ('04), p. 29), $\hat{F}_n(x) \rightarrow F_X(x)$ as $n \rightarrow \infty$.

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

- **4.4 Order Statistics:**

Ordered observations of samples of size n ,

$$\min(\vec{X}) \equiv x_n^{(1)} \leq x_n^{(2)} \leq \dots \leq x_n^{(n-1)} \leq x_n^{(n)} \equiv \max(\vec{X}),$$

allowing for nonunique values due to ties, play a big role in constructing quantiles, $[q_i]_{m \times 1}$, empirically estimated CDFs, $\hat{F}_n(x)$, and other statistical quantities, usually in the background with computational software.

It is assumed that the ordered observations, $x_n^{(k)}$ for $k = 1:n$, correspond to realizations of RVs $X_n^{(k)}$ called *kth order statistics*.

For example, the quantile marks q_k associated with probabilities p_k can be reformulated as

$$\hat{q}_k = q_k[\hat{F}_n] = x_n^{(k)}, \text{ for } \frac{k-1}{n} < p_k \leq \frac{k}{n}.$$

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

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

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

- **4.5 Extreme Values, Fat Tails, Pareto Distributions, and POTs:**

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,

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

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

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

There are many restrictions to the generalized Pareto parameters and the user can check MATLAB Help for them.

MATLAB has the typical family of support functions named for functionality, such a **gppdf**, **gpcdf**, **gpinv**, **gpstat**, **gprnd**, maximum likelihood function **gpfit**, and its log-likelihood counterpart **gplike**.

Sometimes a pure reciprocal power law like

$$f_X^{(\text{pwr})}(x; K) = Cx^{-K}$$

may be used rather than the generalized form.

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.

For example, examining the Q-Q plot comparing the 2008 S&P 500 log-returns against the normal distribution in Lecture 3 on page 10 indicates that $\mathbf{LRpot} = -0.04$ looks like a good POT value, although others could be chosen. 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 Vilfredo Pareto expect a positive tail*) into another vector, say \mathbf{LRtail} .

Since **gpfit.m** *did not fit*, 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. The presentation of results follows several pages later.

- *Q-Q Plot, Reprinted and Axes Corrected from Lecture 3, page 10, Comparing 2008 Log>Returns Against Corresponding Normal Distribution:*

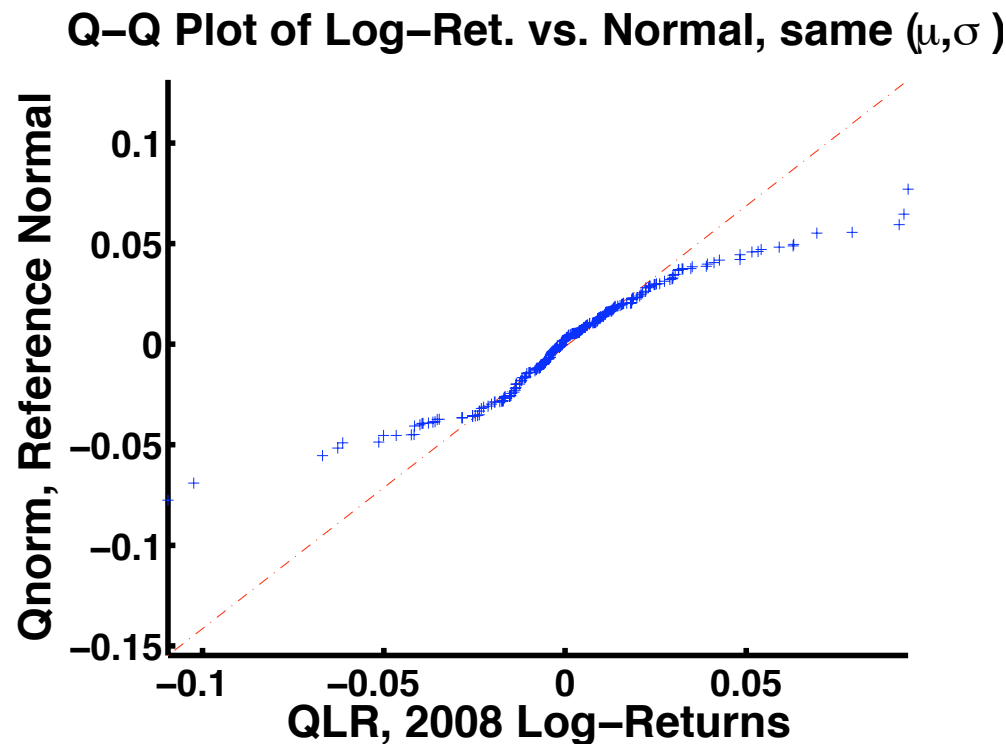


Figure 5: Q-Q Plot of *S&P500 Index log-returns* for the whole year 2008 compared to a simulated normal distribution with the *same* mean (μ) and standard deviation or volatility (σ). *Look at those really fat tails!*

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

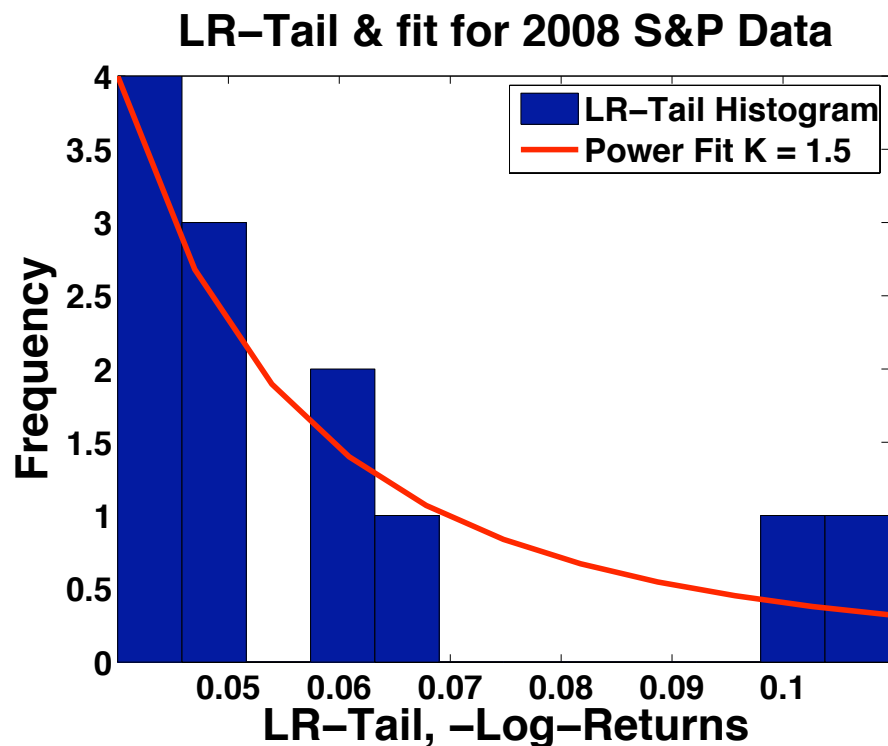


Figure 6: The histogram display the negative tail of the Log>Returns with sign reversed on $[-LR_{pot}, -LR_{max}]$. The fitted *red* — line is really an *eye-ball fit* since *gpf* could not fit it.

- ***MATLAB Code for Histogram of Negative Tail (Sign Reversed) of 2008 S&P 500 Log>Returns:***

```
function tailfitspc2008
% 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 S.mat; % Note: Change GSPC2008adjC.txt name for load function.
fprintf('\ntailfitspc2008.m Output for Log>Returns, 1/16/2009:');
L = length(S); NLR = L-1;
LR = log(S(2:L))-log(S(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 = -0.04; % "guestimate" Peak Over Threshold from Q-Q plot
Npot = 0;
for i = 1:NLR
    if LRsort(1,i) > POT % get tail data size
        Npot = i-1;
        break;
    end
end
fprintf('\nPOT = %7.5f; Npot = %3i;',POT,Npot);
LRtail = -LRsort(1,1:Npot); % gpfitt data must be positive
```

```

gpparms = gpfit(LRtail);
figure(1);
nbins = 12;
hist(LRtail,nbins); axis tight; hold on;
xmin = min(LRtail); xmax = max(LRtail);
fprintf('\nnbins = %2i; xmin = %7.5f; xmax = %5.3f;',nbins,xmin,xmax);
X = xmin+(xmax-xmin)*(0:10)/10;
K = 1.5; Scale = 4*xmin^(K+1);
fx = Scale./X.^(K+1);
plot(X,fx,'-r','LineWidth',3); axis tight; hold off;
title('LR-Tail & fit for 2008 S&P Data','FontSize',24,'FontWeight','Bold');
xlabel('LR-Tail, -Log>Returns','FontSize',24,'FontWeight','Bold');
ylabel('Frequency','FontSize',24,'FontWeight','Bold');
legend('LR-Tail Histogram','Power Fit K = 1.5');
set(gca,'FontSize',18,'FontWeight','Bold');
tailfitspc2008.m Output for Log>Returns, 1/16/2009:
minLR = -0.10957; maxLR = 0.095;
POT = -0.04000; Npot = 12;Warning: Maximum likelihood has converged
to a boundary point of the parameter space. Confidence intervals and
standard errors can not be computed reliably.
> In gpfit at 122
    In tailfitspc2008 at 23
nbins = 12; xmin = 0.04001; xmax = 0.110;

```

4.6 Exponential Analysis of Log-Return Left-Tail POT Data:

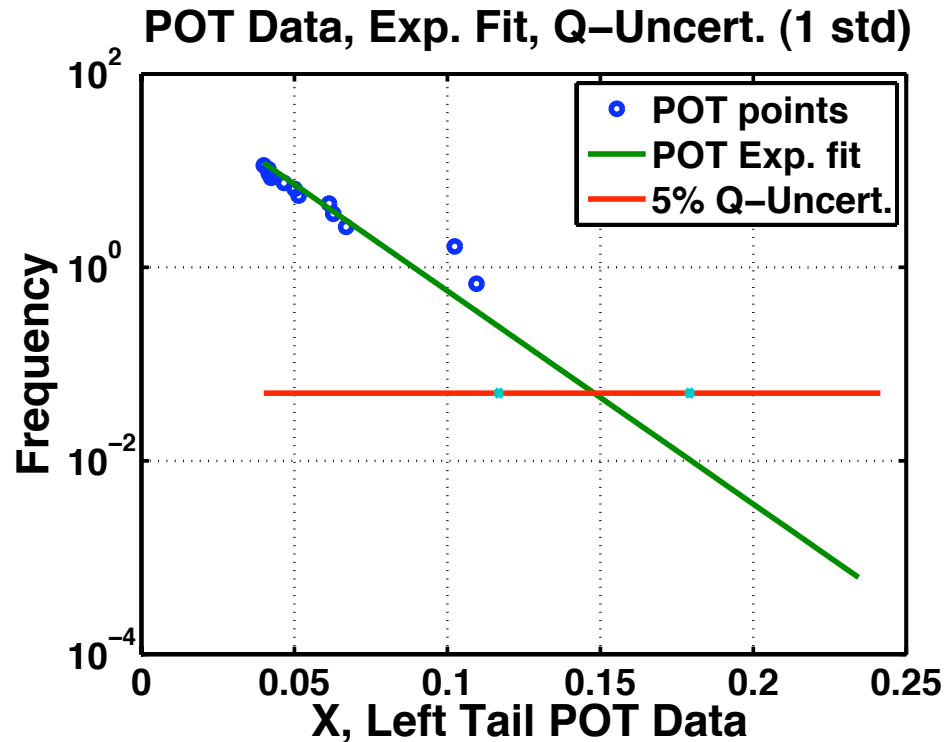


Figure 7: Semi-Log plot of exponential fit to S&P 500 log-return left-tail POT data (sign reversed) using modified `expan.m` code from MATLAB Central File Exchange. Note that the near center points (now on left) fit the *exponential* very well whereas the two largest jumps not on the right are not as close to the fit, but are within Q-uncertainty of $\alpha = 0.05$ or 5%.

MATLAB Code Driver for Fast Exponential Analysis Fit for Left-Tail POT Data:

```
function tailexpc2008
% 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 S.mat; % Note: Change GSPC2008adjC.txt name for load fn.
fprintf('\ntailexpc2008.m Output for Log>Returns, 1/16/2009:');
L = length(S); NLR = L-1;
tp = 1; p = 0.05;% One year's collection time, p = alpha = 5\%.
LR = log(S(2:L))-log(S(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 = -0.04; % "guestimate" Peak Over Threshold from Q-Q plot
Npot = 0;
for i = 1:NLR
    if LRsort(1,i) > POT % get tail data size
        Npot = i-1;
        break;
    end
end
fprintf('\nPOT = %7.5f; Npot = %3i;',POT,Npot);
```

```

LRtail = -LRsort(1,1:Npot); % gpfit data must be positive
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;  
>>
```

4.7 Bivariate ($m = 2$ dimensions) and Multivariate ($m \geq 2$ dimensions) Distributions and Exploratory Data Analysis^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 ,

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

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

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

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

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

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

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

^aSee also Carmona ('04), Chapter 2.

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

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

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,

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

using the fundamental theorem of calculus and second,

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

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

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

The *MATLAB Stat Toolbox* bivariate histogram **hist3** can be used for a qualitative graphical display of the density on a rectangular grid, e.g., **hist3(XY)**; where **XY=[x, y]** is a $n \times 2$ matrix or array of data vectors **x** and **y** and the bivariate histogram is displayed in 3-dimensions on a default 10×10 grid. The user can specify the number of bins in both dimensions with the 2-vector **nbins = [nbins(1), nbins(2)]** and the form **hist3(XY, nbins)**; . Similarly, **hist3(XY, centers)**; allows the user to select bin centers with array **centers=[xcenters, ycenters]** where **xcenters** and **ycenters** are the center vectors. See MATLAB Help or the Lect. 5, p.2.

- **4.8 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 $(\tilde{x}, \tilde{y}) = ((x - x_i)/x_{bw}, (y - y_i)/y_{bw})$ with positive bandwidth vector (x_{bw}, y_{bw}) and change of variables $d\tilde{x}d\tilde{y} = dxdy/(x_{bw}y_{bw})$, a common $dxdy$ canceling out between original and transformed densities, i.e.,

$f_{X,Y}(x, y) = f_{\tilde{X},\tilde{Y}}(\tilde{x}, \tilde{y}; x_{bw}, y_{bw})/(x_{bw}y_{bw})$. The *sample estimator* with *nonnegative kernel* K is

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

If the variable (x, y) are similar in physical dimensions and other attributes, the bandwidth could be the same, i.e., $x_{bw} = y_{bw}$ then Carmona's ('04) very simplified form could be used,

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

The **ksdensity**, from the Statistical Toolbox used in the univariate case, ONLY takes Vector values 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.

- **4.9 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 do to rapid electronic communication. 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,

$$\rho_{X,Y} \equiv \frac{\text{Cov}[X, Y]}{\sigma_X \sigma_Y},$$

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

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

where $\mu_X = \mathbf{E}[X]$ and $\mu_Y = \mathbf{E}[Y]$ are the respective means of X and Y.

The standard deviations or volatilities are $\sigma_X = \sqrt{\mathbf{E}[(X - \mu_X)^2]}$ and $\sigma_Y = \sqrt{\mathbf{E}[(Y - \mu_Y)^2]}$.

- *Sample Correlation Coefficients and Related Sample Moments:*

Recall the unbiased sample means,

$$(\bar{x}, \bar{y}) = (\hat{\mu}_X, \hat{\mu}_Y) = \frac{1}{n} \sum_{i=1}^n (x_i, y_i),$$

and the sample (biased) variances,

$$(\hat{\sigma}_X^2, \hat{\sigma}_Y^2) = \frac{1}{n} \sum_{i=1}^n ((x_i - \bar{x})^2, (y_i - \bar{y})^2),$$

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

$$\widehat{\text{Cov}}_{X,Y} = \hat{\sigma}_{X,Y} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

and the sample correlation coefficient is

$$\hat{\rho}_{X,Y} = \frac{\widehat{\text{Cov}}_{X,Y}}{\hat{\sigma}_X \hat{\sigma}_Y} = \frac{\hat{\sigma}_{X,Y}}{\hat{\sigma}_X \hat{\sigma}_Y}.$$

The correlation coefficients ρ and $\hat{\rho}_{X,Y}$ are bounded in $[-1, +1]$; if $\rho = \pm 1$, then $Y = \pm \sigma_2 X / \sigma_1 + \alpha^a$, an affine function of X ; if X and Y are independent, then $\rho = 0$, but the converse is not true, in general.

^aHanson ('07), 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,

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

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

$$\Sigma = \text{Cov}[\vec{X}, \vec{X}^\top] = [\sigma_{i,j}]_{m \times m}$$

is the $m \times m$ **covariance matrix**, with determinant $\det[\Sigma]$ and

$\sigma_{i,i} \equiv \sigma_i^2$ are the diagonal variance terms for $i = 1:m$. For notational

simplicity, we say $\vec{X} \stackrel{\text{dist}}{=} \mathcal{N}(\vec{\mu}, \Sigma)$ for \vec{X} is normally distributed with mean $\vec{\mu}$ and covariance matrix Σ . By linear transformation this can be

decomposed into multivariate standard form^a, $\vec{X} = \vec{\mu} + \sqrt{\Sigma} \vec{Z}$ where

$\vec{Z} \stackrel{\text{dist}}{=} \mathcal{N}(\vec{0}, I_m)$ is the standard multivariate random variable, I_m being the $m \times m$ identity matrix.

^aCarmona ('04), Chapt. 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

$\Sigma = V_m \equiv [\sigma_i^2 \delta_{i,j}]_{m \times m}$ 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,

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

- *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 $i = 1 : 2$, $\sigma_{1,2} = \rho\sigma_1\sigma_2$, where $\rho = \rho_{1,2}$ is the correlation coefficient between state 1 and state 2 such that $-1 < \rho < +1$. Thus,

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

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

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

$$\det[\Sigma] = (1 - \rho^2)\sigma_1^2\sigma_2^2,$$

^aThis Section from Hanson ('07), Online Appendix B, pp. B.47ff.

yields the more explicit density form:

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

Remark: The bivariate normal density becomes *singular* when $\sigma_1 \rightarrow 0^+$ or $\sigma_2 \rightarrow 0^+$ or $\rho^2 \rightarrow 1^-$ 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 |
|---|
| $E[1] = 1$ |
| $E[x_i] = \mu_i, i = 1 : 2$ |
| $\text{Var}[x_i] = \sigma_i^2, i = 1 : 2$ |
| $\text{Cov}[x_1, x_2] = \rho\sigma_1\sigma_2$ |
| $E[(x_i - \mu_i)^3] = 0, i = 1 : 2$ |
| $E[(x_i - \mu_i)^4] = 3\sigma_i^4, i = 1 : 2$ |
| $E[(x_1 - \mu_1)^2(x_2 - \mu_2)^2] = (1 + 2\rho^2)\sigma_1^2\sigma_2^2$ |

^aThis Table from Hanson ('07), Online Appendix B, pp. B.48.