

Exploring Statistical Processes with *Mathematica7*

Fred Spiring
The University of Manitoba
 CANADA

1. Introduction

Methods for estimating, assessing and monitoring processes are illustrated using the software package *Mathematica7* (Wolfram (2009)). Graphical techniques that allow the dynamic assessment of underlying distributional properties as well as capabilities are presented and illustrated. In addition, innovative procedures associated with compositional data in the L^3 space are examined and expanded to the L^1 constrained space for two variables and the L^2 space for three variables. Several new conventions are proposed that attempt to provide insights into a variety of processes, all with diagnostic tools useful for, but not limited to the manufacturing sector. Several estimation and inferential techniques are presented with tools for determining associated estimates and the resulting inferences. The manuscript is accompanied by a *Mathematica7* notebook best viewed using *Mathematica7* or *Mathematica7 Player*. *Mathematica7 Player* is a free download available at www.Wolfram.com/products/player/ that allows all features of the notebook to be viewed.

2. Creating Probability Plots

Probability plots are graphical expressions used in examining data structures. Plots provide insights into the suitability of a particular probability density function (pdf) in describing the stochastic behavior of the data and estimates of the unknown parameters of the pdf. Although generally very powerful, the inferences drawn from probability plots are subjective.

The underlying principle behind probability plots is simple and consistent. The order statistics, with $Y_{[i]}$ denoting the i th largest observation, such that

$$Y_{[1]} \leq Y_{[2]} \leq \dots \leq Y_{[i]} \leq \dots \leq Y_{[n]}$$

are plotted versus their expected values $E(Y_{[i]})$. A linear relationship between the order statistics and their expected values indicates the pdf used in determining the expected values provides a reasonable representation of the behavior of the observed data. A non-linear plot suggests that other pdf(s) may be more suitable in describing the stochastic structure of the data.

The expected value of the i th order statistic is

$$E(Y_{[i]}) = n! / [(i-1)!(n-i)!] \int_0^1 Y_{[i]} [F(y_{[i]})]^{i-1} [1 - F(y_{[i]})]^{n-i} dF(y_{[i]})$$

where $f(y)$ denotes the pdf being considered, $F(y)$ the associated cumulative distribution function (cdf) and n the size of the dataset under investigation. Because numerical solutions for this equation can be difficult, the approximation $E(Y_{[i]}) = F^{-1}[(i - c)/(n - 2c - 1)]$, where F^{-1} denotes the inverse cdf and c a constant ($0 \leq c \leq 1$) is frequently used. Setting $c=0.5$ (for discussion see Kimball (1960)) results in

$$E(Y_{[i]}) = F^{-1}[(i - 0.5)/n]$$

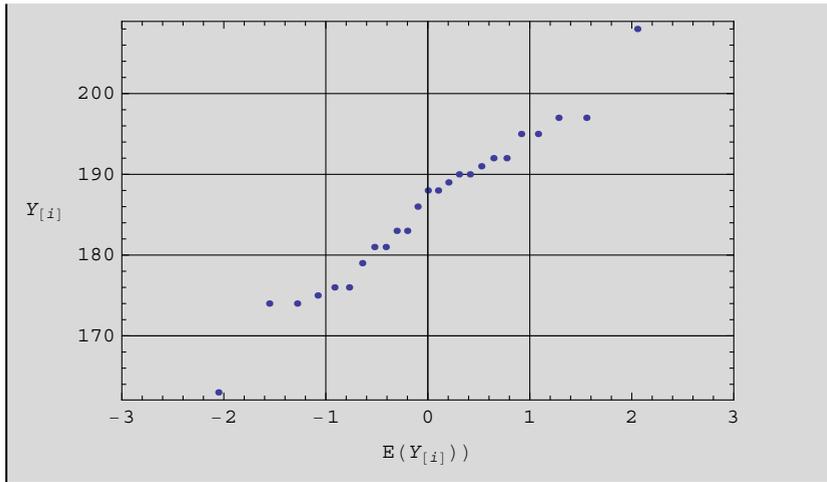
and is the approximation used here. *Mathematica* will be used to evaluate the $E(Y_{[i]})$, create the resulting probability plot, assist in assessing linearity and determine parameter estimates.

Mathematica's Quantile functions are used to find the $E(Y_{[i]})$'s for specific pdfs and create the plot of $Y_{[i]}$ versus $E(Y_{[i]})$. If the resulting plot is considered linear then the pdf used to determine the $E(Y_{[i]})$'s can be used to describe the stochastic structure of the data. Assuming the plot is deemed linear, estimates for the unknown parameters can be determined from the plot.

```

y = {163, 174, 174, 175, 176, 176, 179, 181, 181, 183, 183,
     186, 188, 188, 189, 190, 190, 191, 192, 192, 195, 195,
     197, 197, 208};
pdfs = NormalDistribution[0, 1];
EYpairs[y_, pdfs_] :=
  With[{n = Length[y]},
    Transpose[{Map[Quantile[pdfs, #] &, (Range[n] - 0.5)/n],
              Sort[y]}]]
linePlot :=
  Plot[Evaluate[Fit[EYpairs[y, pdfs], {1, x}, x],
        {x, -3.0, 3.0}], DisplayFunction -> Identity];
r2 := LinearModelFit[EYpairs[y, pdfs], {1, x}, x]
probabilityPlot[yList_, pdfList_] :=
  ListPlot[EYpairs[y, pdfs],
    FrameLabel -> {"E(Y[i])", "Y[i]"},
    RotateLabel -> False,
    PlotRange -> {{-3, 3}, Automatic},
    Frame -> True,
    GridLines -> {{-1, 0, 1}, Automatic},
    DisplayFunction -> Identity];
Show[probabilityPlot[yList, pdfList], Prolog -> AbsolutePointSize[4]]

```



For the normal family of density functions, $f(y) = (2\pi\sigma^2)^{-1/2}\exp(-(y - \mu)^2 / (2\sigma^2))$, $-\infty < y < \infty$, the standard pdf is identified in the routine by `NormalDistribution[0,1]` and the data denoted by `y`. If the resulting normal probability plot is considered linear, then the intersection of the plot with the $E(Y_{[i]}) = 0$ asymptote provides an estimate for the location parameter μ (in this case 187) and the slope provides an estimate for the scale parameter σ . Using the plot's intersection points with the vertical asymptotes ± 1 and dividing by 2 results in an estimate, in this case, of 10 for σ .

The addition of a least squares line and the resulting coefficient of determination (R^2) provide insights into the linearity of the probability plot. The least squares line provides visual assistance in assessing the linearity, while R^2 provides numerical assessment (as R^2 increases, the more linear the probability plot). The least squares line and R^2 are included in subsequent plots.

```

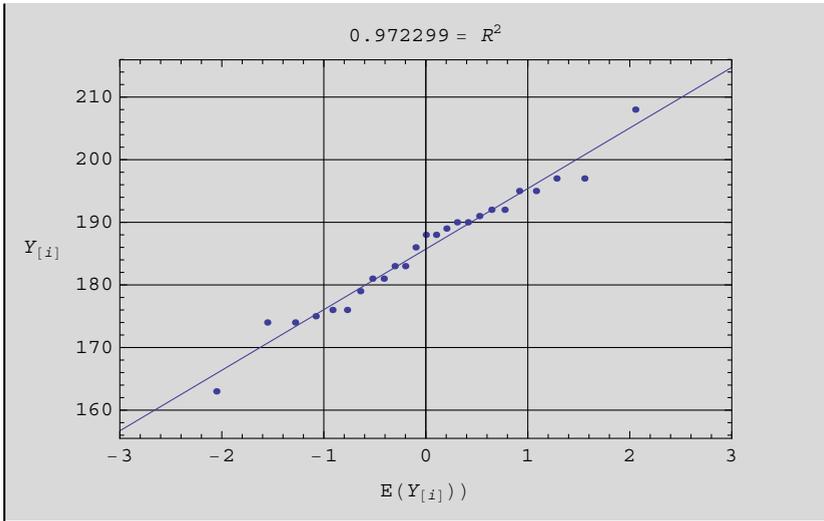
y = {163,174,174,175,176,176,179,181,181,183,183,
     186,188,188,189,190,190,191,192,192,195,195,
     197,197,208};
pdfs = NormalDistribution[0,1];
EYpairs[y_,pdfs_] :=
  With[{n = Length[y]},
    Transpose[{Map[Quantile[pdfs,#]&, (Range[n] - 0.5)/n],
              Sort[y]}]]
linePlot :=
  Plot[Evaluate[Fit[EYpairs[y,pdfs], {1,x}, x]],
        {x, -3.0, 3.0}, DisplayFunction->Identity];
r2 := LinearModelFit[EYpairs[y,pdfs], {1,x}, x]
probabilityPlot[yList_,pdfList_] :=
  ListPlot[EYpairs[y,pdfs],
    FrameLabel->{"E(Y_{[i]})", "Y_{[i]}", "= R^2 "r2["RSquared"]},
    RotateLabel->False,

```

```

PlotRange->{{-3,3},Automatic},
Frame->True,
GridLines->{{-1,0,1},Automatic},
DisplayFunction->Identity];
Show[probabilityPlot[yList,pdfList],linePlot,
Prolog->AbsolutePointSize[4]]

```



If the resulting probability plot is not considered linear then alternative pdfs may be considered. Simply changing the pdf used in determining the $E(Y_{[i]})$'s will allow different stochastic structures to be examined. Replacing NormalDistribution[0,1] with ExponentialDistribution[1] in the routine determines the $E(Y_{[i]})$'s for the pdf $f(y) = (1/\theta) \exp[-y/\theta]$, $0 < y < \infty$. Altering the PlotRange and position of the asymptotes results in an exponential probability plot.

```

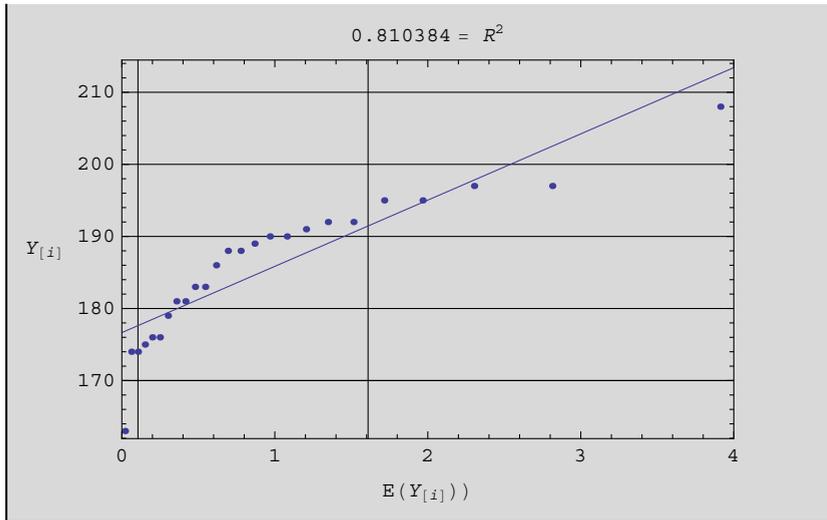
pdfs = ExponentialDistribution[1];
EYpairs[y_, pdfs_] :=
  With[{n = Length[y]},
    Transpose[{Map[Quantile[pdfs, #] &, (Range[n] - 0.5)/n],
      Sort[y]}]]]
linePlot :=
  Plot[Evaluate[Fit[EYpairs[y, pdfs], {1, x}, x],
    {x, 0, 4.0}], DisplayFunction->Identity];
r2 := LinearModelFit[EYpairs[y, pdfs], {1, x}, x]
probabilityPlot[yList_, pdfList_] :=
  ListPlot[EYpairs[y, pdfs],
    FrameLabel->{"E(Y_{[i]})", "Y_{[i]}", "= R^2 " r2["RSquared"]},
    RotateLabel->False,
    PlotRange->{{0, 4}, Automatic},

```

```

Frame->True,
GridLines->{{Quantile[pdfs,.1],
Quantile[pdfs,.8]},Automatic},
DisplayFunction->Identity];
Show[probabilityPlot[yList,pdfList],linePlot,
Prolog->AbsolutePointSize[4]]

```



In the case of the standard exponential distribution, if the resulting probability plot is considered linear then an estimate of θ can be determined as $1.504/(y_{.8} - y_{.1})$ (Shapiro (1980)), where $y_{.8}$ is the 80th percentile and $y_{.1}$ is the 10th percentile of the distribution. Vertical asymptotes have been included at the 10th and 80th percentiles to facilitate determining the points of intersection with these asymptotes.

Creating a probability plot for the standard uniform distribution, $f(y) = 1/\theta, -\theta/2 \leq y \leq \theta/2$ requires changing the routine to `UniformDistribution[0, 1]`. In addition the `PlotRange` is altered to (0, 1) and asymptotes added at .25, .5, .75. If the resulting uniform probability plot is considered linear, then an estimate of θ is determined using the plot's intersection with the 25th and 75th percentiles (i.e., $y_{.25}, y_{.75}$) as follows $(y_{.75} - y_{.25}) / .5$.

```

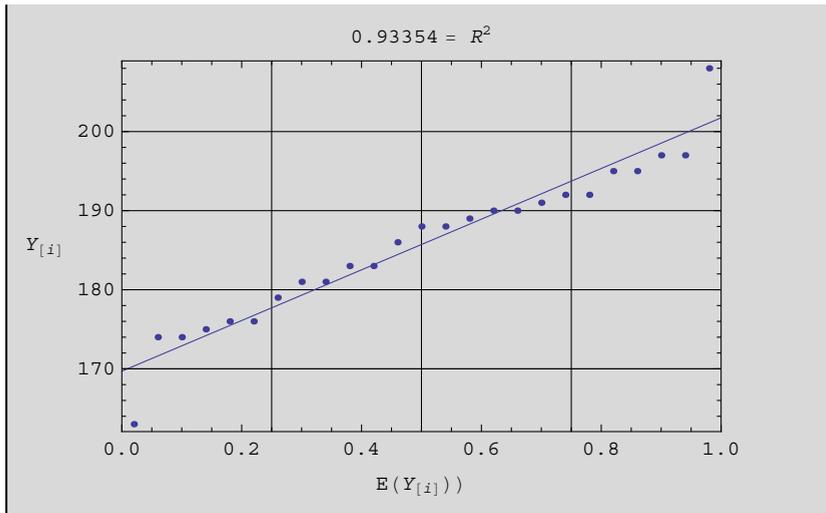
pdfs = UniformDistribution[{0,1}];
EYpairs[y_,pdfs_] :=
  With[{n = Length[y]},
    Transpose[{Map[Quantile[pdfs,#]&, (Range[n] - 0.5)/n],
    Sort[y]}]]]
linePlot:=
  Plot[Evaluate[Fit[EYpairs[y,pdfs], {1,x},x],
  {x,0,1.0}],DisplayFunction->Identity];
r2:=LinearModelFit[EYpairs[y,pdfs], {1,x},x]

```

```

probabilityPlot[yList_, pdfList_]:=
  ListPlot[EYpairs[y, pdfs],
    FrameLabel->{"E(Y[i])", "Y[i]", "= R2"*r2["RSquared"]},
    RotateLabel->False,
    PlotRange->{{0, 1}, Automatic},
    Frame->True,
    GridLines->{{.25, .5, .75}, Automatic},
    DisplayFunction->Identity];
Show[probabilityPlot[yList, pdfList], linePlot, Prolog->AbsolutePointSize[4]]

```



Other pdfs can be examined by changing the distribution function specified in the routine. Probability plots for the LogNormalDistribution[0,1] and WeibullDistribution[1, 3.25] distributions are illustrated.

```

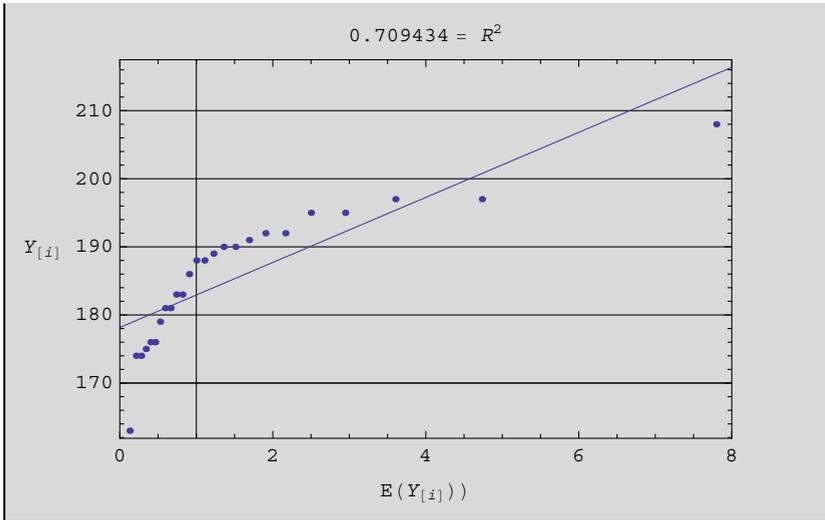
pdfs = LogNormalDistribution[0, 1];
EYpairs[y_, pdfs_]:=
  With[{n = Length[y]},
    Transpose[{Map[Quantile[pdfs, #]&, (Range[n] - 0.5)/n],
      Sort[y]}]]
linePlot:=
  Plot[Evaluate[Fit[EYpairs[y, pdfs], {1, x}, x],
    {x, 0, 8}], DisplayFunction->Identity];
r2:=LinearModelFit[EYpairs[y, pdfs], {1, x}, x]
probabilityPlot[yList_, pdfList_]:=
  ListPlot[EYpairs[y, pdfs],
    FrameLabel->{"E(Y[i])", "Y[i]", "= R2"*r2["RSquared"]},
    RotateLabel->False,
    PlotRange->{{0, 8}, Automatic},

```

```

Frame->True,
GridLines->{{1}, Automatic},
DisplayFunction->Identity];
Show[probabilityPlot[yList, pdfList], linePlot,
Prolog->AbsolutePointSize[4]]

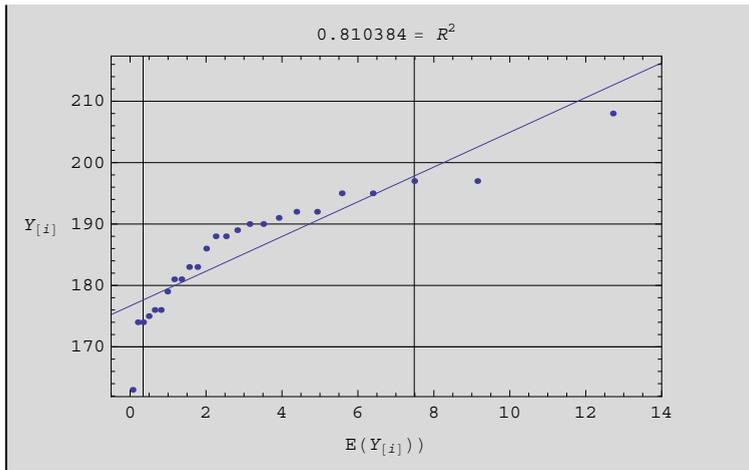
```



```

pdfs = WeibullDistribution[1, 3.25];
EYpairs[y_, pdfs_] :=
  With[{n = Length[y]},
    Transpose[{Map[Quantile[pdfs, #] & (Range[n] - 0.5)/n],
      Sort[y]}]]]
linePlot :=
  Plot[Evaluate[Fit[EYpairs[y, pdfs], {1, x}, x],
    {x, -.5, 14}], DisplayFunction->Identity];
r2 := LinearModelFit[EYpairs[y, pdfs], {1, x}, x]
probabilityPlot[yList_, pdfList_] :=
  ListPlot[EYpairs[y, pdfs],
    FrameLabel->{"E(Y[i])", "Y[i]", "= R^2 " r2["RSquared"]},
    RotateLabel->False,
    PlotRange->{{-.5, 14}, Automatic},
    Frame->True,
    GridLines->{{Quantile[pdfs, .1],
      Quantile[pdfs, .9]}, Automatic},
    DisplayFunction->Identity];
Show[probabilityPlot[yList, pdfList], linePlot, Prolog->AbsolutePointSize[4]]

```



Probability plots are generally restricted to the class of pdfs characterized by location and scale parameters. The Weibull distribution of the form $f(y) = \alpha \beta y^{\beta-1} \exp(-\alpha y^\beta)$, $\alpha, \beta > 0, 0 \leq y < \infty$, is an exception in that α and β are considered scale and shape parameters. In the previous example the values of the parameters were set at 1 and 3.25 respectively. A linear relationship in a Weibull probability plot suggests that the Weibull distribution with specific parameter values is appropriate in describing the stochastic nature of the data. However a non-linear relationship does not necessarily rule out the Weibull family of distributions but may be a reflection on the value(s) of the parameters chosen.

In the case of the Weibull distribution, taking the natural logarithm twice and plotting allows the distribution (with scale and shape parameters) to be examined analogous to those distributions characterized by location and scale parameters (Hahn and Shapiro (1967)). However, in general, probability plots can assess only those distributions with no (or at least a constant) shape parameter. Cheng and Spiring (1990) used rotation to illustrate techniques that extend the use of probability plots to a class of pdfs characterized by location, scale and shape parameters. Of particular interest were the Weibull and Tukey's- λ distributions as both are characterized by a location, scale and single shape parameter.

2.1 Creating & Interpreting 3-D Probability Surfaces

Dynamic graphic techniques have opened new frontiers in data display and analysis. With a basic understanding of simple probability plots, subjective interpretation of distributional assumptions can be made for families of distributions that contain a shape parameter. Strong visual results are possible for relatively small sample sizes. In the examples that follow, sample sizes of 25 provide good insights into the distributional properties of the observed data.

Let Y denote a random variable with pdf $f(y; \mu, \sigma, \lambda)$ and cdf $F(y; \mu, \sigma, \lambda)$ where μ, σ and λ denote the location, scale and shape parameters of the distribution respectively. Cheng and Spiring (1990) defined the X-axis as $E(Y_{[i]}; \lambda)$, scaled the Z-axis arithmetically and defined it as the order statistics $Y_{[i]}$ and let the Y axis denote values of the shape parameter λ , to create

a surface in 3 space. Examination of the resulting surface allowed inferences regarding the stochastic nature of the data as well as estimates for location, scale and shape parameters of the associated pdf.

The resulting surface is essentially an infinite number of traditional probability plots laid side by side. These probability plots are ordered by the value of the shape parameter used in calculating the $E(Y_{[j]})$'s. Slicing the surface along planes parallel to the XZ plane at various points along the Y axis, allows viewing of the "linearity" of the surface by considering the resultant projection on the XZ plane. The projection is a univariate probability plot of the data for a particular value of the shape parameter. The goal then is to slice the surface such that the most linear projection on the XZ plane is found.

Rotation allows viewing of the created surface from several perspectives, enhancing the ability to determine where the surface appears most linear and the associated value of the shape parameter. From the most linear portion of the surface, estimates for the location, scale and shape parameters can be determined. The 50th percentile (or midpoint of the X-axis provides an estimate for the location, the value of the Y-axis where the surface is most linear provides an estimate for the shape parameter and the slope of the surface (in the X-direction) an estimate of the scale.

In practice the order statistics are plotted versus the expected value of the ordered statistics for various values of the shape parameter. Then examining various views of the surface allows one to determine the value of the shape parameter associated with the most linear portion of the curve. From there estimates for the location and scale parameters are possible.

Animation permits a series of univariate probability plots (for specific values of the shape parameter) to be viewed in a sequential fashion, highlighting changes in the probability plots resulting from changes in the shape parameter. This results in a quick and reliable method for determining the most linear portion of the surface. The procedure creates a series of univariate probability plots representing various values of the shape parameter. The observer must determine which of the plots (if any) is most linear. If the surface provides no linear results, then one concludes that the data do not arise from the family of distributions considered.

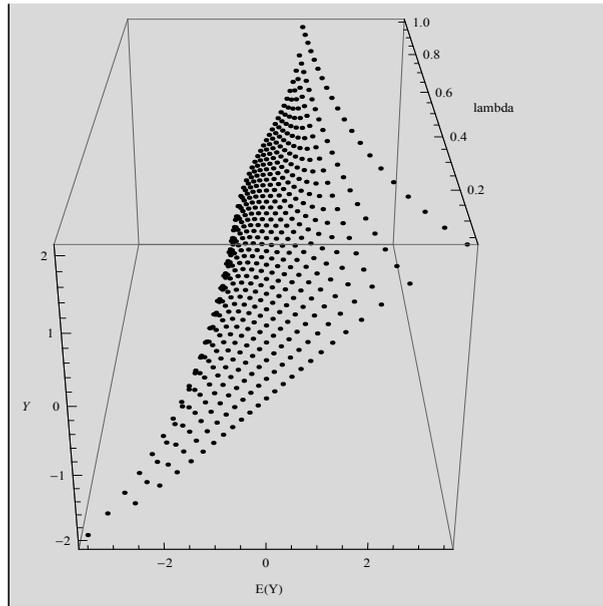
2.2 Example

Letting y denote the 25 simulated normal distribution results, $E(Y_{[j]}; \lambda)$ the expected value of the associated order statistics and λ the shape parameter, a surface in three space can be created using the following routine.

```

y = {.1, .2025, .3045, .4124, .523, .6433, .7723, .9154, 1.08, 1.282, 1.555, 2.054, -2.054,
-1.555, -1.282, -1.08, -.9154, -.7723, -.6433, -.523, -.4124, -.3045, -.2025, -.1, 0};
n = Count[y, _]; s = Min[y]; l = Max[y]; d = 4(l - s)/n; x = Sort[y];
t[lambda_] = Table[Point[{x[[j]], (((j - .5)/n)^lambda - (1 - ((j - .5)/n))^lambda)
/lambda}, lambda}], {j, 1, n}];
Show[Graphics3D[Table[{t[lambda]}, {lambda, 0.05, 1, .05}], Axes->True,
AxesLabel->{Y, "E(Y)", "lambda"}, BoxRatios->{2, 2, 4},
ViewPoint->{1, 0, -2}]]

```



The pdf of the lambda distribution can be determined for specific values of λ , however it is generally given as the pdf of Z under the transformation $Z = ((X^\lambda) - (1 - X)^\lambda)/\lambda$ where $X \sim U[0, 1]$. The transformation is also the percentile function for the distribution and results in the expected value of the order statistics being of the form $E(Y_{[i]}; \lambda) = (((i - 0.5)/n)^\lambda - (1 - (i - 0.5)/n)^\lambda)/\lambda$. While rotation allows different views of this surface, determining the most linear portion can still be difficult. Rather than rotating, slicing and viewing the resulting plots, the following routine creates a series of probability plots enhanced with a regression line and R^2 , as well as the associated value of λ that can be viewed using the animation function of *Mathematica*.

```

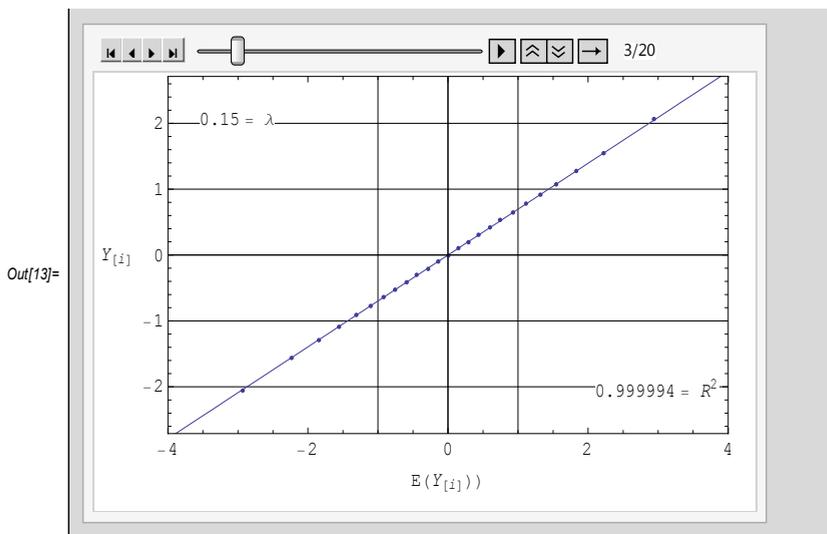
y = {.1, .2025, .3045, .4124, .523, .6433, .7723, .9154, 1.08, 1.282, 1.555, 2.054,
     -2.054, -1.555, -1.282, -1.08, -.9154, -.7723, -.6433, -.523, -.4124, -.3045,
     -.2025, -.1, 0};
n = Length[y];
s = Min[y];
t = Max[y];
d = 4(t - s)/n;
l:=Text[Style[lambda " = λ^n"], {-3, t}];
pdfs[j_]:=((((j - .5)/n)^lambda - (1 - ((j - .5)/n))^lambda)/lambda;
EYpairs[y_, pdfs_]:=With[{n = Length[y]},
  Transpose[{Map[pdfs[#]&, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
    11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25}], Sort[y]}]];
linePlot:=Plot[Evaluate[Fit[EYpairs[y, pdfs], {1, x}, x]],
  {x, -4.0, 4.0}, DisplayFunction->Identity];
r2:=LinearModelFit[EYpairs[y, pdfs], {1, x}, x];
probabilityPlot[yList_, pdfList_]:=

```

```

ListPlot[EYpairs[y, pdfs],
FrameLabel->{"E(Y[i])", "Y[i]"}, RotateLabel->False,
PlotRange->{{-4, 4}, {s - d, t + d}}, Frame->True,
GridLines->{{-1, 0, 1}, Automatic},
DisplayFunction->Identity];
rr:=Text[Style["= R2"], {3, -t}];
SlideView[Table[Show[probabilityPlot[yList, pdfList], linePlot,
Graphics[l], Graphics[rr],
DisplayFunction->$DisplayFunction, ImageSize -> Scaled[0.9],
Prolog->AbsolutePointSize[4], {lambda, 0.05, 1.00, 0.05}],
AppearanceElements -> All]

```



The resulting series of plots represent the projections associated with slices of the surface taken at $\lambda=0.05(.05)1.0$. Using the animation keys to sequentially examine the plots, it quickly becomes apparent that the most linear probability plot occurs at $\lambda=0.15$. The R^2 value supports the visual assessment reaching its maximum of 0.999994 at $\lambda=0.15$. The asymptote $E(Y_{[i]}) = 0$ suggests an estimated mean of 0, while the slope of the plot suggests an estimated standard deviation of $(0.8 - (-0.8))/2 = 0.8$. This example highlights the relationship that exists between the normal and the symmetric lambda families. The symmetric lambda distribution with $\lambda=0.14$ is used as an approximation to the normal distribution.

The pdf of the standard Weibull distribution is of the form $f(y) = \lambda y^{\lambda-1} e^{-y^\lambda}$, $0 < y < \infty$, and the expected value of the order statistics can be approximated by $\left(-\ln\left[\frac{2n+1-2i}{2n}\right]\right)^{\frac{1}{\lambda}}$. The subsequent routine creates a series of univariate probability plots that permits examination of the Weibull family of distributions for values of the shape parameter $\lambda=1(.25)5$. Again the goal is to find the most linear portion of the surface or most linear slice of the surface for the values of the shape parameter considered. In those cases where the “most” linear probability

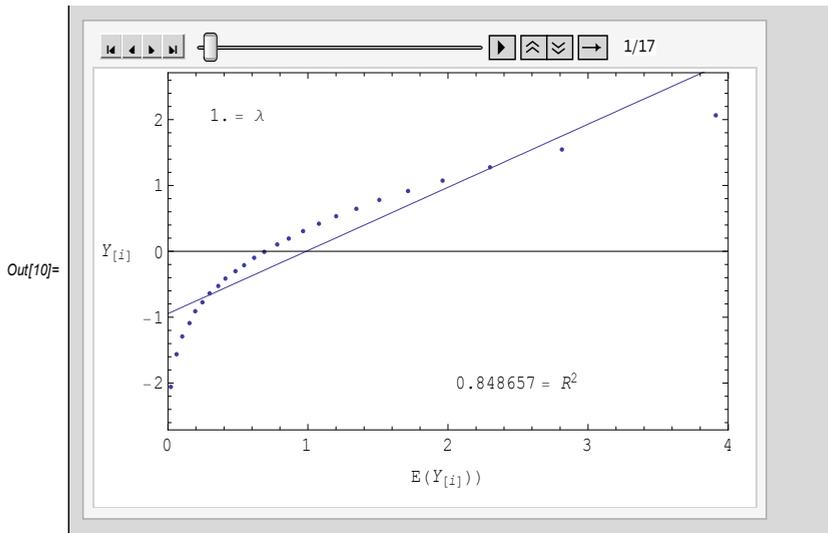
plot is deemed non-linear then either the Weibull family is inappropriate and/or the value of the shape parameter has not been included.

A Weibull distribution with shape parameter of approximately 3.25 is often cited as a reasonable approximation to the normal distribution. Again using the animation keys to sequentially examine the plots, it quickly becomes apparent that the most linear plot of the series visually appears to occur at $\lambda=3.25$ or $\lambda=3.50$ while the R^2 value suggests that the most linear plot occurs at $\lambda=3.50$.

```

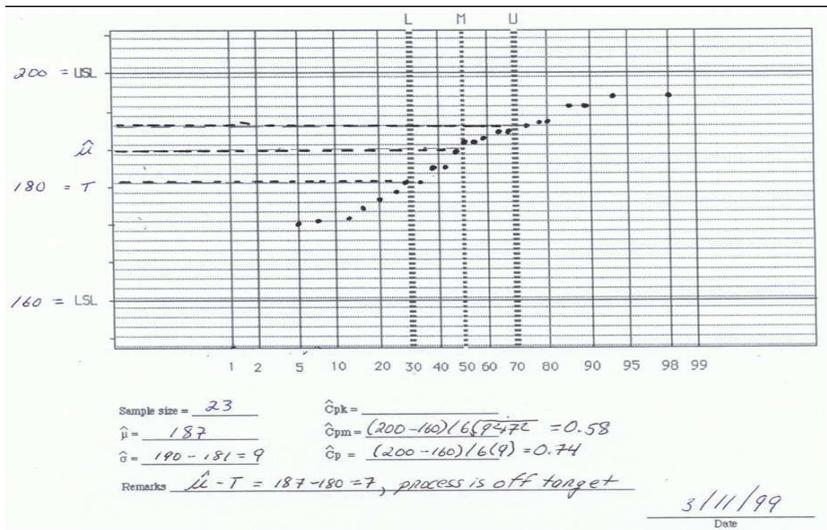
y = {0.1, 0.2025, 0.3045, 0.4124, 0.523, 0.6433, 0.7723, 0.9154, 1.08, 1.282, 1.555, 2.054,
-2.054, -1.555, -1.282, -1.08, -0.9154, -0.7723, -0.6433, -0.523, -0.4124,
-0.3045, -0.2025, -0.1, 0};
n = Length[y]; s = Min[y]; t = Max[y]; d =  $\frac{4(t-s)}{n}$ ;
l := Text[Style[lambda =  $\lambda$ "], {"0.5", t}];
pdfs[j_] :=  $\left(-\text{Log}\left[\frac{2n+1-2j}{2n}\right]\right)^{1/\text{lambda}}$ 
EYpairs[y_, pdfs_] :=
With[{n = Length[y]},
Transpose[
{(pdfs[#1]&)/@{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25}, Sort[y]}]]
linePlot := Plot[Evaluate[Fit[EYpairs[y, pdfs], {1, x}, x]], {x, 0, "4."}, DisplayFunction -> Identity];
r2 := LinearModelFit[EYpairs[y, pdfs], {1, x}, x];
probabilityPlot[yList_, pdfList_] :=
ListPlot[EYpairs[y, pdfs], FrameLabel -> {"E(Y[i])", "Y[i]"}, RotateLabel -> False,
PlotRange -> {{0, 4}, {s - d, t + d}}, Frame -> True, DisplayFunction -> Identity];
rr := Text[Style["= R2" r2["RSquared"], {2.5, -2}];
SlideView[
Table[Show[probabilityPlot[yList, pdfList], linePlot, Graphics[rr], Graphics[l],
DisplayFunction -> $DisplayFunction, ImageSize -> Scaled[0.9],
Prolog -> AbsolutePointSize[4], {lambda, "1.", "5.", "0.25"}], AppearanceElements -> All]

```



3. Process Capability Paper

Chan, Cheng and Spiring (1988) proposed a graphical technique for examining process capability by combining the concepts that process capability indices assume the underlying distribution is normal and the graphical assessment of normality derived from normal probability plots. The result was Process Capability Paper. An example where 23 observations were gathered from a process with $USL=200$, $T=180$ and $LSL=160$ is illustrated below.



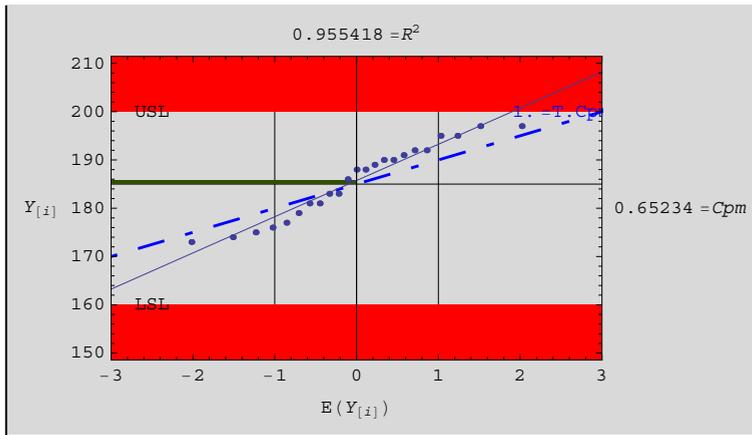
Mathematica can be used to a) evaluate the $E(Y_{[i]})$'s, b) create the resulting Process Capability Paper plot and c) assist in assessing linearity and determining parameter estimates. The

addition of a least squares line and the resulting coefficient of determination (R^2) provide insights into linearity of the probability plot. The least squares line provides visual assistance in assessing the linearity, while R^2 provides numerical assessment. The least squares line and R^2 (i.e., RSquared) are included in subsequent plots. If the resulting capability plot is not considered linear then the various process capability indices may not provide valid indications of process capability.

Mathematica can be used to create the basic format for the Process Capability Paper by inputting the basic information from the process including the study results (data), upper specification limit (USL), lower specification limit (LSL), Target and Target C_{pm} (TCpm). The following *Mathematica* code will create an updated version of the Process Capability Paper.

```
data = {173, 174, 175, 176, 177, 179, 181, 181, 183, 183, 186, 188, 188, 189, 190, 190,
191, 192, 192, 195, 195, 197, 197};
```

```
USL:=200; LSL:=160; Target:=185; TCpm:=1.00;
(* code to create plots *)
m:=Mean[data]; s:=StandardDeviation[data];
cpm:=Min[{USL - Target}, {Target - LSL}]/(3 * (s^2 + (m - Target)^2)^(1/2));
sig:=(((Min[{USL - Target}, {Target - LSL}])^2)/(9 * (TCpm^2)))^(1/2);
mut:=(m - Target);
pdfs = NormalDistribution[0, 1];
EYpairs[data_, pdfs_] :=
With[{n = Length[data]},
Transpose[{Map[Quantile[pdfs, #]&, (Range[n] - 0.5)/n], Sort[data]}]]
linePlot:=Plot[Evaluate[Fit[EYpairs[data, pdfs], {1, x}, x],
{x, -3.0, 3.0}];
r2:=LinearModelFit[EYpairs[data, pdfs], {1, x}, x]
probabilityPlot[dataList_, pdfList_] :=
ListPlot[EYpairs[data, pdfs],
FrameLabel -> {"E(Y[i])", "Y[i]", r2["RSquared"] == "", "=Cpm" N[cpm]}},
RotateLabel -> False,
PlotRange -> {{-3.0, 3.0}, {LSL - 1.5 * s, USL + 1.5 * s}}, Frame -> True,
GridLines -> {{-1, 0, 1}, {Target}}];
Show[probabilityPlot[dataList, pdfList],
Graphics[{RGBColor[.2, .3, 0], Rectangle[{-3, Target}, {0, m}]}],
Graphics[{RGBColor[1, 0, 0], Rectangle[{-3, LSL - 1.5 * s}, {3, LSL}]}],
Graphics[{RGBColor[1, 0, 0], Rectangle[{-3, USL}, {3, USL + 1.5 * s}]}],
Graphics[Text["USL", {-2.5, USL}], Graphics[Text["LSL", {-2.5, LSL}]],
Graphics[{RGBColor[0, 0, 1], Text["=T.Cpm" N[TCpm], {2.5, USL}]}],
Graphics[{RGBColor[0, 0, 1], AbsoluteThickness[1.5],
Dashing[ {.01, .05, .05, .05} ],
Line[{{-3, Target - 3 * sig}, {3, Target + 3 * sig}]}]],
probabilityPlot[dataList, pdfList], linePlot, ImageSize -> Scaled[1]]
```



The resulting plot represents 23 observations from a process with a target (T) of 185, upper specification limits (USL) of 200, lower specification limit (LSL) of 160. A Target Cpm (TCpm) value of 1 is used to illustrate some of the features included in this enhanced version of Process Capability paper. The enhanced Process Capability paper continues to be a normal probability plot with the y-axis representing the value of the order statistics ($Y_{[i]}$) and the x-axis the expected value of the order statistics ($E(Y_{[i]})$) assuming the underlying distribution is normal.

The resulting plot includes T , USL and LSL with the areas beyond the specification limits highlighted in red. The difference between the process target (T) and the process average (μ) is indicated by the green box. An ordinary least squares (OLS) line (solid line) and the R^2 value (top of the plot frame) are included in order to facilitate the assessment of normality through the linearity of the points and their strength of association. The value of Cpm associated with the data is included along with the OLS line (dashed) representing a process that is on target with a $Cpm = 1$.

3.1 Process Capability Paper Enhancements

The *Mathematica*-produced Process Capability Paper has several enhancements. The output includes the basic features of Process Capability Paper including a normal probability plot of the data including asymptotes at -1, 0 and 1; an OLS line and R^2 in an attempt to enhance the "linearity" assessment of the probability plot; a dashed line reflecting the the slope of the line associated with the target Cpm ; identified regions beyond the specification limits highlighted in red; and a graphics box indicating the distance the mean is from the target highlighted in green. A second example is illustrated below.

```

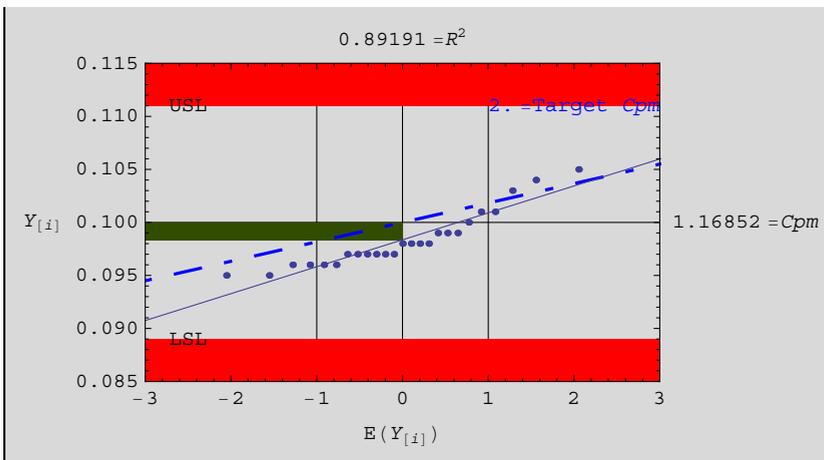
data = {0.101,0.105,0.099,0.098,0.097,0.101,0.098,0.095,
0.099,0.103,0.096,0.104,0.096,0.098,0.097,0.096,"0.097",
0.099,0.098,0.097,0.095,0.096,0.1,0.097,0.097};
USL:="0.111";LSL:="0.089";Target:="0.1";TCpm:=2;
m:=Mean[data];
s:=StandardDeviation[data];
cpm:= $\frac{\text{Min}\{\text{USL}-\text{Target}\},\{\text{Target}-\text{LSL}\}}{3\sqrt{s^2+(m-\text{Target})^2}}$ 

```

```

sig:= $\sqrt{\frac{\text{Min}\{\text{USL}-\text{Target},\text{Target}-\text{LSL}\}^2}{9\text{TCpm}^2}}$ 
mut:=m - Target
pdfs = NormalDistribution[0, 1];
EYpairs[data_, pdfs_] :=
With[{n = Length[data]},
Transpose [ { (Quantile[pdfs, #1] &)/@ $\frac{\text{Range}[n]-"0.5"}{n}$ , Sort[data] } ] ]
linePlot:=Plot[Evaluate[Fit[EYpairs[data, pdfs], {1, x}, x]],
{x, -"4.", "4."}, DisplayFunction -> Identity];
r2:=LinearModelFit[EYpairs[data, pdfs], {1, x}, x]
probabilityPlot[dataList_, pdfList_] := ListPlot[EYpairs[data, pdfs],
FrameLabel -> { "E(Y[i])", "Y[i]", r2["RSquared"] == "", "=Cpm" N[Cpm] },
RotateLabel -> False,
PlotRange -> {{ -"3.", "3."}, {LSL - "1.5"s, USL + "1.5"s}}, Frame -> True,
Axes -> None, GridLines -> {{ -1, 0, 1}, {Target}},
DisplayFunction -> Identity];
Show[probabilityPlot[dataList, pdfList],
Graphics[{RGBColor["0.2", "0.3", 0], Rectangle[{ -3, Target}, {0, m}]}],
Graphics[{RGBColor[1, 0, 0], Rectangle[{ -3, LSL - "1.5"s}, {3, LSL}]}],
Graphics[{RGBColor[1, 0, 0], Rectangle[{ -3, USL}, {3, USL + "1.5"s}]}],
Graphics[Text["USL", { -"2.5", USL}], Graphics[Text["LSL", { -"2.5", LSL}]],
Graphics[{RGBColor[0, 0, 1], Text["=Target Cpm" N[TCpm], { "2.", USL}]}],
Graphics[{RGBColor[0, 0, 1], AbsoluteThickness["1.5"],
Dashing[{ "0.01", "0.05", "0.05", "0.05" }],
Line[{ { -3, Target - 3sig}, {3, Target + 3sig} }]}],
probabilityPlot[dataList, pdfList], linePlot,
DisplayFunction -> $DisplayFunction, Prolog -> AbsolutePointSize[8],
ImageSize -> Scaled[1]]

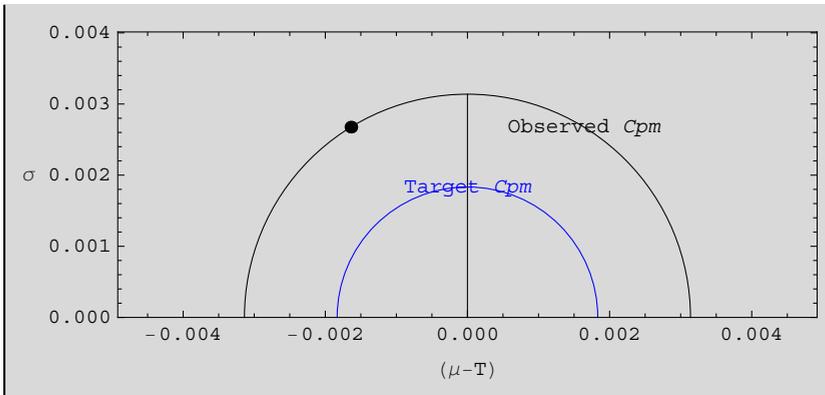
```



3.2 Additional Views of Process Capability

Mathematica can be used to enhance the inferences from the Enhanced Process Capability Paper by examining departures from the target and incorporating target Capability values with their associated curves.

```
Show[Graphics[{RGBColor[0,0,1],Text["Target Cpm",{0,sig}],
Circle[{0,0}, $\frac{1}{3*TCpm}(\text{Min}[\{\text{USL}-\text{Target}\},\{\text{Target}-\text{LSL}\}])$ ,
{0,180°}]}],
Graphics[{Black,Text["Observed Cpm",{Abs[mut],s}],
Circle[{0,0}, $\frac{1}{3*cpm}(\text{Min}[\{\text{USL}-\text{Target}\},\{\text{Target}-\text{LSL}\}])$ ,
{0,180°}]}],Frame->True,
PlotRange->{{-3*Abs[mut],3*Abs[mut]},{0,1.5*s}},
FrameLabel->{"(μ-T)", "σ"},RotateLabel->False,
Prolog->{AbsolutePointSize[6],Point[{mut,s}],
Line[{0,0},{0,(1/(3*cpm))(\text{Min}[\{\text{USL}-\text{Target}\},\{\text{Target}-\text{LSL}\}])]}],
ImageSize->Scaled[1]]
```



This plot illustrates the various combinations of a process’s variability and off-targetness associated with a particular value of *Cpm*. All points lying on the blue (Target *Cpm*) semi-circle represent combinations of off-tagetness ($\mu - T$) and variability (σ) that result in the Target $Cpm= 2$. The black semi-circle represents all combinations of off-targetness and variability that have a *Cpm* value equivalent to that exhibited by the process under investigation. The point represents the observed off-targetness and variability combination associated with the process.

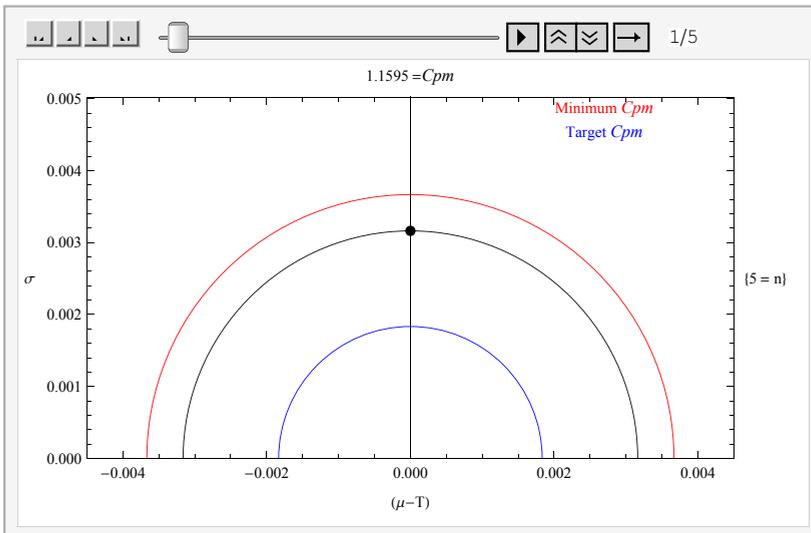
Animation permits a series of plots to be viewed in a sequential fashion resulting in a reliable method for examining a) different views of a single sample or b) multiple samples from comparable processes. The following creates an animated view of multiple samples from a single process. The plot includes the observed *Cpm*, min *Cpm* (red semicircle) and Target *Cpm* for five samples of size five taken from a process.

```
mincpm = 1; TCpm = 2; USL = 0.111; LSL = 0.089; Target = 0.1; groups = 5;
data[1] = {0.101,0.105,0.099,0.098,0.097};
```

```

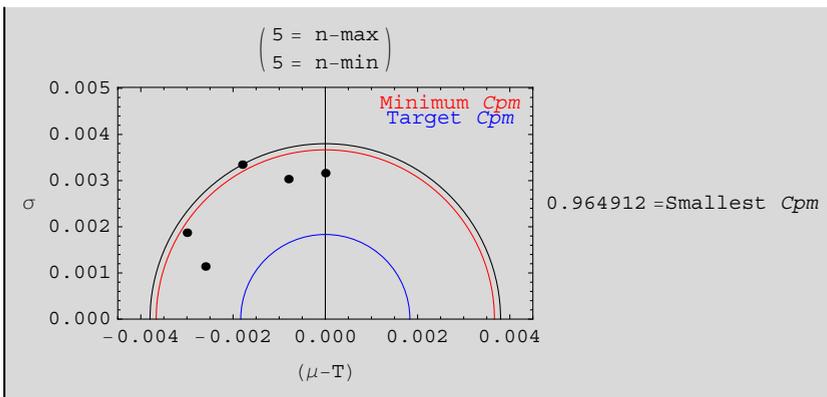
data[2] = {0.101, 0.098, 0.095, 0.099, 0.103};
data[3] = {0.096, 0.104, 0.096, 0.098, 0.097};
data[4] = {0.096, 0.097, 0.099, 0.098, 0.097};
data[5] = {0.095, 0.096, 0.1, 0.097, 0.097};
Do[{mu[i] = Mean[data[i]], s[i] = StandardDeviation[data[i]], mut[i] = mu[i] - Target,
cpm[i] = (Min[{USL - Target}, {Target - LSL}]/(3 * (s[i]^2 + (mu[i] - Target)^2)^(1/2))),
n[i] = Length[data[i]]}, {i, groups}];
maxmut:=Max[{Abs[mut[1]], Abs[mut[2]], Abs[mut[3]], Abs[mut[4]], Abs[mut[5]]}];
maxs:=Max[{s[1], s[2], s[3], s[4], s[5]}];
SlideView[
Table[
Show[
Graphics[Circle[{0, 0}, Min[{USL - Target}, {Target - LSL}]/(3cpm[j]),
{0, 180°}], Frame -> True,
Prolog -> {AbsolutePointSize[6], Point[{mut[j], s[j]}]},
Line[{0, 0}, {0, 1.5 * maxs}],
{RGBColor[1, 0, 0], Text["Minimum Cpm", {0.9 * maxmut, 1.45 * maxs}]},
Circle[{0, 0}, Min[{USL - Target}, {Target - LSL}]/(3 * mincpm),
{0, 180°}], {RGBColor[0, 0, 1],
Text["Target Cpm", {0.9 * maxmut, 1.35 * maxs}]},
Circle[{0, 0}, Min[{USL - Target}, {Target - LSL}]/(3 * TCpm),
{0, 180°}],}],
PlotRange -> {{-1.5 * maxmut, 1.5 * maxmut}, {0, 1.5 * maxs}},
FrameLabel -> {"(μ-T)", "σ", "=Cpm"cpm[j], {"= n"n[j]}},
RotateLabel -> False, ImageSize -> Scaled[1], {j, groups}],
AppearanceElements -> All]

```



A stationary plot of the observed *Cpm*, min *Cpm* and Target *Cpm* for five samples of size five taken from a process can be created using the following code.

```
Show[
Graphics[
Circle[{0,0},
(((Min[{USL - Target}, {Target - LSL}]))/
(3 * Min[{cpm[1], cpm[2], cpm[3], cpm[4], cpm[5]}])), {0, 180Degree}],
Graphics[{RGBColor[1,0,0], Text["Minimum Cpm", {.9 * maxmut, 1.4 * maxs}],
Circle[{0,0}, (((Min[{USL - Target}, {Target - LSL}]))/(3 * mincpm)), {0, 180Degree}]}],
Graphics[{RGBColor[0,0,1], Text["Target Cpm", {.9 * maxmut, 1.3 * maxs}],
Circle[{0,0}, (((Min[{USL - Target}, {Target - LSL}]))/(3 * TCpm)), {0, 180Degree}]}],
Frame -> True,
FrameLabel->{"(μ-T)", "σ", {"= n-max" {Max[n[1], n[2], n[3], n[4], n[5]]},
"= n-min" {Min[n[1], n[2], n[3], n[4], n[5]]}},
"=Smallest Cpm"Min[{cpm[1], cpm[2], cpm[3], cpm[4], cpm[5]}]},
PlotRange->{{-1.5 * maxmut, 1.5 * maxmut}, {0, 1.5 * maxs}}, RotateLabel->False,
Prolog -> {AbsolutePointSize[4], Point[{mut[1], s[1]}], Point[{mut[2], s[2]}],
Point[{mut[3], s[3]}], Point[{mut[4], s[4]}], Point[{mut[5], s[5]}]},
Line[{0,0}, {0, (1.5 * maxs)}]}], ImageSize -> Scaled[1]]
```



4. Compositional Data

Compositional data refers to the group of constrained space metrics that take the form

$$X_1 + X_2 + X_3 + X_4 + \dots + X_d = a$$

where $0 \leq X_i \leq a$ for all i and each X_i represents a proportion of the total composition a . Setting $d=2$ results in all possible combinations of R^{+2} (shaded region in Figure 4.1) that satisfy the equation $X_1 + X_2 = a$. Graphically these combinations represent a line in R^{+2} and referred to as the L^1 space.

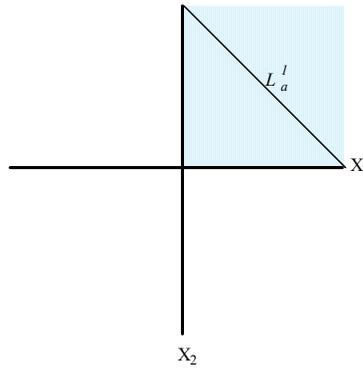


Figure 4.1 The L^1_a Constrained Space

All observations in the L^1_a space lie on the line $X_1 + X_2 = a$, with different values of a , moving the line either closer or further from the origin. The Euclidean distance along the perpendicular from the origin to the L^1_a constrained space is $a\sqrt{2}$. In addition the points where the L^1_a space intersects the axes are exactly a units from the origin (see Figure 4.2).

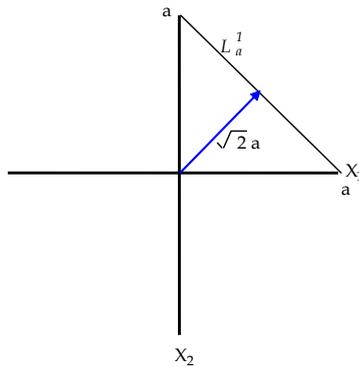


Figure 4.2 Distance from Origin to L^1_a along perpendicular

4.1 The L^2 Space

The triple X_1, X_2, X_3 subject to the constraint $X_1 + X_2 + X_3 = a$, represents a point in R^{+3} space. Ternary paper, also referred to as Triangular coordinate paper, is available for observations of the form X_1, X_2, X_3 where $X_1 + X_2 + X_3 = a$ and uses a planar view (see Figure 4.3) of the constrained space. Most commercial ternary paper adds scaling and axes to enhance the plotting procedure with the resulting region referred to as the L^2 space.

An alternative view of the L^2 space rewrites the equation in the form $X_1 + X_2 = a - X_3$ and makes use of the L^1 space. The points X_1, X_2 are located on the $L^1_{a-X_3}$ line, which is $\sqrt{2}(a - X_3)$ units (along the perpendicular) from the origin. As the compositional make-up varies (i.e., as we observe different values of the triple X_1, X_2, X_3), the $L^1_{a-X_3}$ line will vary as

will the location (i.e., X_1, X_2) on the line. The L^2 space consists of the set of subspaces $L^1_{a-X_3}$ where $0 \leq X_3 \leq a$. When using normal arithmetic paper, the values can be determined directly from the plot (see Figure 4.4). X_1 and X_2 are the usual projections onto their appropriate axes, while X_3 is the distance from the intersection of $L^1_{a-X_3}$ (with either of the axes) to a on the same axis.

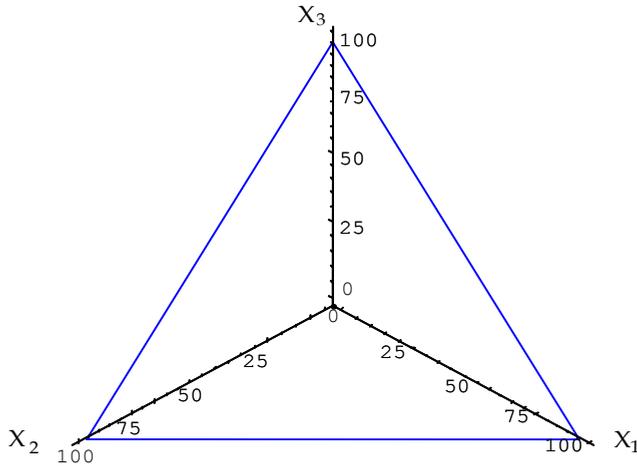


Figure 4.3 The L^2 space as a plane in Three Space

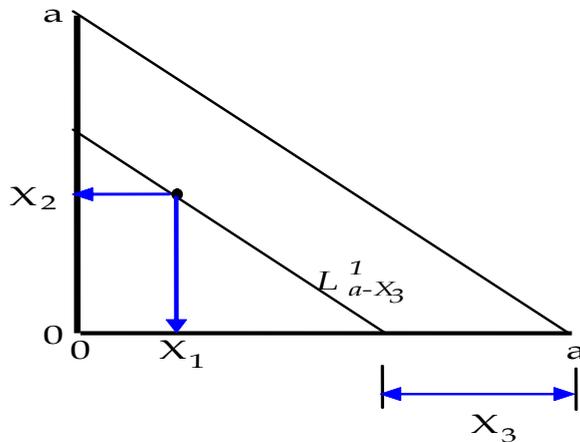


Figure 4.4 Geometric Interpretations of the Constrained Triple

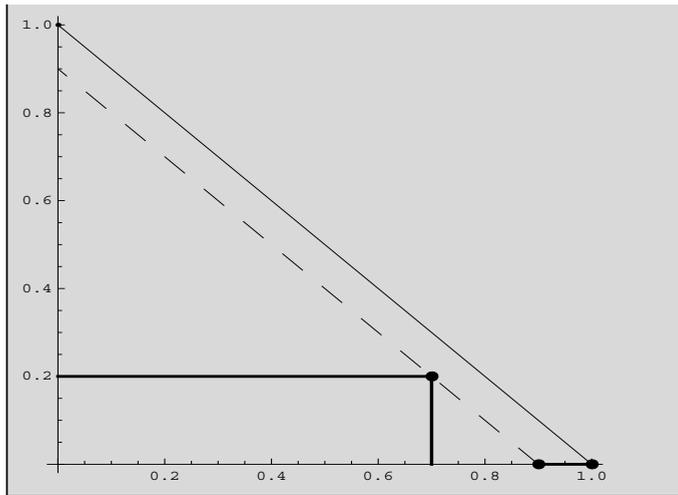
The general L^2_a space is easily created using arithmetic graph paper requiring no special scaling or plotting procedures. The plot is easily generalized to allow for various values of a

and the creation of a general form of the constrained space paper. The *Mathematica* code to create the L^2_a constrained space paper and plot an observed points (i.e., 0.7, 0.2, 0.1 with a = 1.0) follows.

```

a = 1; x1 = 0.7; x2 = 0.2; x3 = a - x1 - x2;
Show[Graphics[{Point[{0, a}], AbsolutePointSize[7],
Point[{x1, x2}], Point[{a - x3, 0}], Point[{a, 0}],
Line[{a, 0}, {0, a}],
{Dashing[{0.05, 0.05}], Line[{a - x3, 0}, {0, a - x3}]},
AbsoluteThickness[2],
Line[{x1, x2}, {x1, 0}], Line[{x1, x2}, {0, x2}],
Line[{a, 0}, {a - x3, 0}], AxesOrigin->{0.0, 0.0}, Axes->True]]

```



The L^2 space is the area bounded by the X_1, X_2 axes and the solid line that intersects the axes at the value of $a=1$. The $L^1_{a-X_3}$ space is denoted by the dashed line parallel to solid line intersecting the axes exactly X_3 units from a (again 1 in this case). A heavier line has been drawn along the X_1 axis from a towards the origin that is exactly X_3 units in length (0.1 in this case). X_1, X_2, X_3 (in this case 0.7, 0.2, 0.1) has been highlighted at the appropriate point on the solid line (i.e., the $L^1_{a-X_3}$ space). In addition the projections onto the axes have been included to facilitate reading the values of X_1, X_2 directly from the plot.

Commercial Ternary paper scales the plane characterized by the points $(a, 0, 0)$, $(0, a, 0)$ and $(0, 0, a)$ in a triangular co-ordinate system. Analogous to the L^1 case where we added a third variable to the mix, the L^3_a space can be considered when we add a fourth variable. Similar to the L^2 space development, X_1, X_2, X_3, X_4 where $0 \leq X_i \leq a$ for all i such that $X_1 + X_2 + X_3 + X_4 = a$ can be written as $X_1 + X_2 + X_3 = a - X_4$ and the perpendicular distance between the origin and the L^2 plane to reflect the magnitude of X_4 .

Alternatively we could use other techniques to provide the inference regarding Ternary plots in the $L^2 \times D_1$ domain. Consider the case where X_1, X_2, X_3, X_4 where $0 \leq X_i \leq a$ for all i

such that $X_1 + X_2 + X_3 + X_4 = 100$ and where $X_4 = 0$. This is equivalent to looking at the standard L^2 Ternary plot and in this case would be scaled similar to commercial Ternary paper. The point $(30, 30, 40, 0)$ would appear as follows (see Figure 4.5). This plane would be $\sqrt{2}(100 - 0)$ units along the perpendicular from the origin.

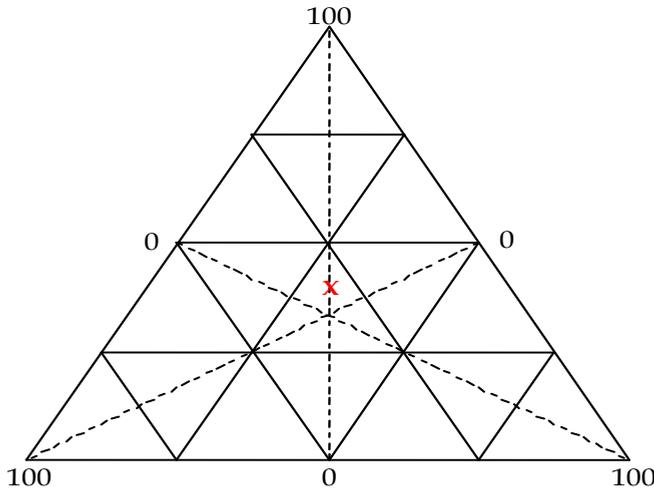
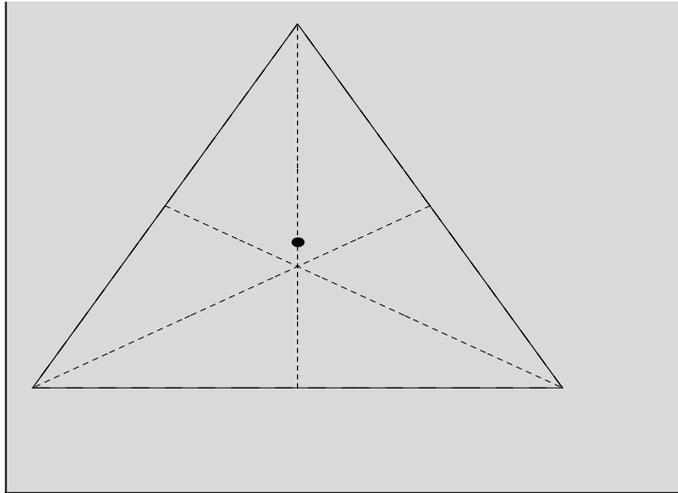


Figure 4.5 Planar view of L^3_a Space with point $(30, 30, 40, 0)$

The following *Mathematica* code results in a plane $\sqrt{2}(100 - 0)$ units along the perpendicular from the origin and point at $(30, 30, 40, 0)$.

```
x = 30; y = 30; z = 40; constraint = 100; a = constraint - x - y - z;
pts = {{0, 100 - a, 0}, {100 - a, 0, 0}, {0, 0, 100 - a}, {0, 100 - a, 0}};
oldpts = {{0, 100, 0}, {100, 0, 0}, {0, 0, 100}, {0, 100, 0}};
oldpts11:={{50 - a/2, 0, 50 - a/2}, {0, 100 - a, 0}}
oldpts12:={{0, 50 - a/2, 50 - a/2}, {100 - a, 0, 0}}
oldpts13:={{50 - a/2, 50 - a/2, 0}, {0, 0, 100 - a}}
Show[Graphics3D[{AbsolutePointSize[7], Point[{x, y, z}],
Line[pts], {Dashing[{0.01, 0.01]},
Line[oldpts11]}], {Dashing[{0.01, 0.01]},
{Line[oldpts12]}], {Dashing[{0.01, 0.01]},
{Line[oldpts13]}], {Dashing[{0.03, 0.03]}, Line[oldpts]}],
{Boxed->False, Ticks->None, Axes->False,
AxesEdge->{{1, 1}, {1, 1}, {1, 1}}, ViewPoint->{-2, -2, -2}]]
```

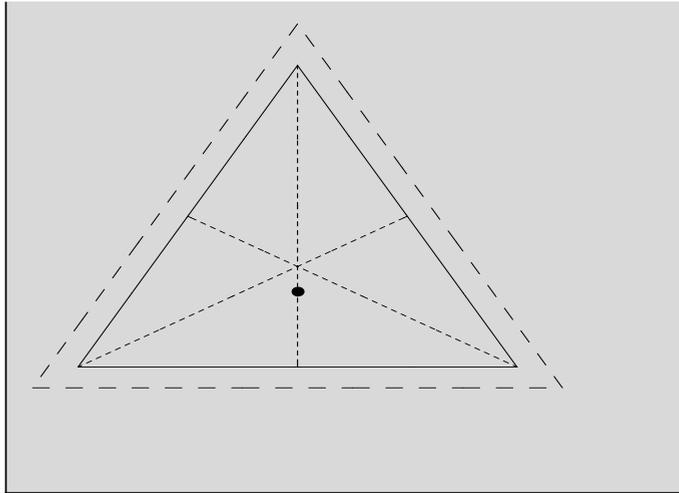


Suppose now that we observe the point $(30, 30, 20, 20)$ under the constraint that $X_1 + X_2 + X_3 + X_4 = 100$. The quadruple is represented by a point on the plane that is $\sqrt{2}(100 - 20)$ units from the origin. The point and its associated plane can be depicted as wholly contained within the plane associated with $X_4 = 0$ (dashed triangle below). *Mathematica* produces the above plot of the Planar view of the L^3_a space with point $(30, 30, 40, 20)$.

```

x = 30; y = 30; z = 20; constraint = 100; a = constraint - x - y - z;
pts = {{0, 100 - a, 0}, {100 - a, 0, 0}, {0, 0, 100 - a}, {0, 100 - a, 0}};
oldpts = {{0, 100, 0}, {100, 0, 0}, {0, 0, 100}, {0, 100, 0}};
oldpts11 := {{50 - a/2, 0, 50 - a/2}, {0, 100 - a, 0}}
oldpts12 := {{0, 50 - a/2, 50 - a/2}, {100 - a, 0, 0}}
oldpts13 := {{50 - a/2, 50 - a/2, 0}, {0, 0, 100 - a}}
Show[Graphics3D[{AbsolutePointSize[7], Point[{x, y, z}],
Line[pts], {{Dashing[{0.01, 0.01}],
Line[oldpts11]}}, {Dashing[{0.01, 0.01}],
{Line[oldpts12]}}, {Dashing[{0.01, 0.01}],
{Line[oldpts13]}}, {Dashing[{0.03, 0.03}], Line[oldpts]}},
{Boxed->False, Ticks->None, Axes->False,
AxesEdge->{{1, 1}, {1, 1}, {1, 1}}, ViewPoint->{-2, -2, -2}]]]

```



5. Comments

We have attempted to show how *Mathematica* can provide practitioners with the ability to quickly and simply examine data. In conjunction with functions found in *Mathematica*, the graphical methods developed may provide powerful inferences when assessing distributional forms, estimating parameter values, investigating process capability and examining constrained data.

6. References

- Chan, L.K., Cheng, S.W. & Spiring, F.A. (1988). A Graphical Technique for Process Capability. ASQ 42nd Annual Quality Congress Transactions.
- Cheng, S.W. & Spiring, F.A. (1990). Some Applications of 3-D scatter plots in data analysis, *Computational Statistics & Data Analysis*, 10, pp. 47-61.
- Hahn, G.J. & Shapiro, S.S. (1967). *Statistical Models in Engineering*. John Wiley & Sons, New York.
- Kimball, B.F. (1960). On the Choice of Plotting Positions on Probability Paper, *Journal of the American Statistical Association*, 55, pp. 546-.
- Wolfram, S. (2009). *Mathematica7*. Wolfram Media, Illinois and Cambridge University Press, New York.



Stochastic Control

Edited by Chris Myers

ISBN 978-953-307-121-3

Hard cover, 650 pages

Publisher Sciyo

Published online 17, August, 2010

Published in print edition August, 2010

Uncertainty presents significant challenges in the reasoning about and controlling of complex dynamical systems. To address this challenge, numerous researchers are developing improved methods for stochastic analysis. This book presents a diverse collection of some of the latest research in this important area. In particular, this book gives an overview of some of the theoretical methods and tools for stochastic analysis, and it presents the applications of these methods to problems in systems theory, science, and economics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Fred Spiring (2010). Exploring Statistical Processes with Mathematica7, Stochastic Control, Chris Myers (Ed.), ISBN: 978-953-307-121-3, InTech, Available from: <http://www.intechopen.com/books/stochastic-control/exploring-statistical-processes-with-mathematica>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.