MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Civil and Environmental Engineering

1.017/1.010 Computing and Data Analysis for Environmental Applications / Uncertainty in Engineering

---

Problem Set 6: Estimates and Confidence Intervals (Solutions provided at end of each problem)
Due: Thursday, Oct. 30, 2003

Please turn in a hard copy of your MATLAB program as well as all printed outputs (tables, plots, etc.) required to solve the problem.

Problem 1: Comparing Alternative Estimates of Population Properties

Suppose that you have a sample of 10 observations of a random variable ***x*** which you believe to be exponentially distributed. Your objective is to estimate the 90 percentile value x_{90} of this variable. This value of the solution of the equation $F_x(x_{90}) = 0.9$.

Propose at least two different methods for estimating x_{90} from the 10 observations.

Compare the performance of these alternative estimators with a stochastic simulation that performs the following steps:

1. Generate many (e.g. 1000) replicates, each consisting of 10 observations drawn from an exponential distribution with parameter $a = E[x]$ specified by you.
2. For each replicate derive an estimate $\hat{x}_{90}$ of x_{90} from each of your two proposed estimators.
3. For each estimator compute the sample mean and variance of $\hat{x}_{90}$ over all replicates. Also, for each estimator construct an $\hat{x}_{90}$ histogram and an $\hat{x}_{90}$ CDF plot (using MATLAB's `normplot` function).
4. Determine whether your estimators are unbiased and consistent (check consistency by plotting the rerunning your simulation for a much larger number of observations).

Which of your estimators is better? Explain your reasoning.

Problem 1 Solution:

```
% Problem set 6 Problem 1
clear all
close all
nrep=1000;
% True x90 value, a=5:
x90true = expinv(0.9,5)
% Method 1: Pick the 9th value of the 10 ranked values
samples1=exprnd(5,nrep,10);
sorted=sort(samples1,2);
x90_1=sorted(:,9);
mu1=mean(x90_1)
var1=var(x90_1)
% Method 2: Take the mean of the sample and use that
% as a to compute x90=Finv(0.9)
samples2=exprnd(5,nrep,10);
means=mean(samples2,2);
x90_2 = expinv(0.9,means);
mu2=mean(x90_2)
var2=var(x90_2)
figure
subplot(2,2,1), hist(x90_1)
subplot(2,2,2), normplot(x90_1)
subplot(2,2,3), hist(x90_2)
subplot(2,2,4), normplot(x90_2)
% The second method is better.  This makes sense because
% you're using all the data, not just one data point per set.
```

Problem 2: The Bivariate Normal Distribution

Two dependent normally distributed random variables (parameters μ_x, μ_y, σ_x, σ_y, and ρ):

$$f_{xy}(x,y) = \frac{1}{2\pi |C|^{0.5}} \exp\left\{ -\left[\frac{(Z-\mu)' C^{-1} (Z-\mu)}{2} \right] \right\}$$

Z = vector of **random variables** = $[x \;\; y]'$

μ = vector of **means** = $[\, E(\boldsymbol{x}) \;\; E(\boldsymbol{y}) \,]'$

C = **covariance matrix** = $C = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}$

$\sigma_x = Std(\boldsymbol{x})$, $\sigma_y = Std(\boldsymbol{y})$, $\rho = Correl(x,y)$

$|C|$ = **determinant** of $C = \sigma_x^2 \sigma_y^2 (1-\rho^2)$

C^{-1} = **inverse** of $C = \dfrac{1}{|C|}\begin{bmatrix} \sigma_y^2 & -\rho\sigma_x\sigma_y \\ -\rho\sigma_x\sigma_y & \sigma_x^2 \end{bmatrix}$

Note that the ′ symbol is used to indicate the vector transpose in the bivariate normal probability density expression. The argument of the exponential in this expression is a scalar.

In this problem you will use the MATLAB function `mvnrnd` to generate scatterplots of correlated bivariate normal samples. This function takes as arguments the means of x and y and the covariance matrix defined above (called `SIGMA` in the MATLAB documentation).

Assume $E[x] = 0$, $E[y] = 0$, $\sigma_x = 1$, $\sigma_y = 0$. Use `mvnrnd` to generate 100 (x, y) realizations . Use `plot` to plot each of these as a point on the (x,y) plane (do not connect the points). Vary the correlation coefficient ρ to examine its effect on the scatter. Consider $\rho = 0.$, 0.5, 0.9. Use subplot to put plots for all 3 ρ values on one page.

Problem 2 Solution:

```
% Problem Set 6 -- Problem 2
clear all
close all
% The Bivariate Normal Distribution
rho=-.5;
sigmax=1;
sigmay=1;
muxy=[0 0];
C=[sigmax^2, rho*sigmax*sigmay; rho*sigmax*sigmay,sigmay^2];
values = mvnrnd(muxy,C,100);
plot(values(:,1),values(:,2),'*')
```

Problem 3: Effect of Sample Size on Estimate Accuracy

Reconsider the arsenic data set from Problem Set 4. Estimate the mean of the complete data set (population) from smaller samples of size N, randomly selected (without replacement) from the complete data set. Compute the sample mean and standard deviation for N = 4, 8, 32, 64, and 128. Plot the differences between the sample and population means and standard deviations (on two different plots) as functions of N. Explain your results.

Problem 3 Solution:

```
% Problem Set 6 -- Problem 3
clear all
close all
load arsenicdata.txt
```

```
N=[4 8 32 64 128];
popmu=mean(arsenicdata);
pops=std(arsenicdata);
for i=1:5
    number=N(i);
    index = randperm(length(arsenicdata));
    sample=arsenicdata(index(1:number));
    mu(i)=mean(sample);
    s(i)=std(sample);
end
mudiff=abs(popmu-mu);
sdiff=abs(pops-s);
figure
plot(N,mudiff,'*')
figure
plot(N,sdiff,'*')
```

Problem 4: Confidence Intervals

The following random sample was drawn from a continuous probability distribution $F_X(x)$. Estimate the mean and standard deviation of this distribution and specify 90%, 95%, and 99% confidence intervals for the mean.

x = [2.6287 7.0923 2.3959 0.4207 2.8124 4.1257 3.1121 0.8913

1.2885 0.1863 0.5489 2.2652 1.3867 8.5322 1.8364 2.3576

0.4417 0.4693 2.2507 0.7189]

Problem 4 Solution:

```
% Problem Set 6 Problem 4
clear all
close all
load p4_sample.txt
data=p4_sample;
mu=mean(data)
s=std(data)
N=length(data);
% 90% confidence interval
zl=norminv(.05);
zu=norminv(.95);
lowerbound90=-(zu*s/sqrt(N)-mu)      %   1.4836
upperbound90=-(zl*s/sqrt(N)-mu)      %   3.0907
% 95% confidence interval
zl=norminv(.025);
zu=norminv(.975);
```

```
lowerbound95=-(zu*s/sqrt(N)-mu)     %   1.3297
upperbound95=-(zl*s/sqrt(N)-mu)     %   3.2446
% 99% confidence interval
zl=norminv(.005);
zu=norminv(.995);
lowerbound99=-(zu*s/sqrt(N)-mu)     %   1.0289
upperbound99=-(zl*s/sqrt(N)-mu)     %   3.5454
```