

Stat 471: Lecture 13

Monte-Carlo Inference.

Moo K. Chung
mchung@stat.wisc.edu

October 5, 2003

1. Quantile function. For given probability α , the quantile is the point q such that $q_\alpha = F^{-1}(\alpha)$. In lecture 11, we showed how to estimate the CDF based on Monte-Carlo integration. It was shown that $F(x)$ can be estimated by $\sum_{i=1}^n \mathbb{I}_{X_i \leq x}/n$, i.e. the proportion of random sample that is smaller than x . Now consider the order statistics $X_{(i)}$ of X_i . If $X_{(j)} \leq x < X_{(j+1)}$, there are j samples that are smaller than x so $F(x) = j/n$. Given $j/n \leq \alpha < (j+1)/n$, $q_\alpha \approx X_{(j)}$.

```
function q=quantile(x,p)
sorted=sort(x);
q=sorted(round(size(x,1)*p));
```

A different possibly more sophisticated estimation can be found in Frigge *et al.* (1989) The American Statistician. 43:50-54.

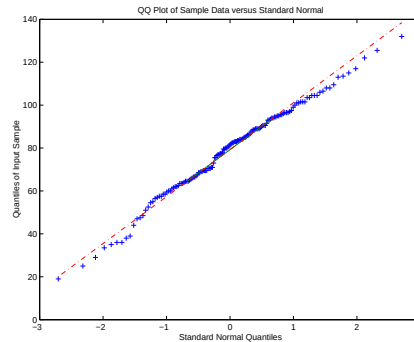
2. We are interested in testing if the arm strength x_i of the construction workers is 80 pounds (load strength.data). Let μ be the mean arm strength. Then

$$H_0 : \mu = 80 \text{ vs. } H_1 : \mu \neq 80.$$

If x_i are from normal, under H_0 , the appropriate test statistic is t -statistic

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t_{n-1}.$$

We reject H_0 if T is either large or small. Suppose $P(-c < T < c) = \alpha$. We reject H_0 if $|t| > c$ with $100(1 - \alpha)\%$ significance. Assume that you do not know the distributional form of T in which case you do now know the value of c and in turn there is no inference.



3. From QQ-plot, the arm strength data x_i seems to follow $N(\mu, \sigma)$ so we use $N(\mu, \sigma)$ as the pseudo-population. We generate m random samples from the pseudo-population and compute T statistic for each random sample.

```
n=size(load_strength,1); m=10000; t=zeros(m,1);
mu=mean(load_strength); sigma=std(load_strength);
>>[mu sigma]
ans =
    78.7517    21.1093
>>tob =(mu-80)/(sigma/sqrt(n))
tob =
   -0.7170
for i=1:m x=sigma*snrnd(n)+mu;
t(i)=(mean(x) - 80)/(sigma/sqrt(n));
end;
>> quantile(t,[0.05 0.95])
ans =
   -2.3720    0.9107
>> normcdf([-1.65, 1.65])
ans =
    0.0495    0.9509
```