Appendix

```
t = [3.4935 4.2853 5.1374 5.8181 6.8632 8.1841]';

x = [6 10.1333 14.2667 18.4000 22.5333 26.6667]';

sig = 0.1;

m = length(t);

G = [ones(m,1) x]; % matrix G

tolr = 1e-8;
tolx = 1e-16;
p = 1;
maxiter = 20;

ML1 = irls(G, t, tolr, tolx, p, maxiter); % irls function
to = ML1(1)
s2 = ML1(2)

err = sig*ones(size(t));

% fitted model
T = zeros(m,1);
for i =1:m
   T(i) = to + s2*x(i);
end

%plot
figure1=figure(1);

%L1 estimate
%subplot(2,1,1)
plot(x,t,'.','MarkerSize',20)
hold on
errorbar(x,t,err)
hold on
plot(x,T,'r',LineWidth=2)
hold off
legend('data','fitted model','Location','southeast')
ylabel('Arrival times')
xlabel('Distance (km)')
title('L1 estimate')

q = 1e4; % no 0f monte carlo simulations
n = 2; % no of parameters

noise = sig*randn(m,q);

%W = (1/sig)*eye(m);
```

```
%Gw = W*G;

db = G*ML1;

M = zeros(q,n);

for i = 1:q
    di = (db + noise(:,i));
    ML1i = G\di;
    M(i,1) = ML1i(1); M(i,2) = ML1i(2);
end

%figure2=figure('Position', [100, 100, 1024, 1200]);
figure(2)
subplot(2,1,1)
qqplot(M(:,1))
ylabel('Quantiles of Input Sample');
xlabel('Standard Normal Quantiles');
title('Q-Q plot for to')
subplot(2,1,2)
qqplot(M(:,2))
ylabel('Quantiles of Input Sample');
xlabel('Standard Normal Quantiles');
title('Q-Q plot for s2')

mbar = mean(M); % mean
A = M - repmat(mbar,[q],[1]);
C = (A'*A)/q; % empirical estimate

%for to
z = 1.96;
ct1 = mbar(1) - sqrt(C(1,1)*z*sqrt(n));
ct2 = mbar(1) + sqrt(C(2,2)*z)*sqrt(n);

 Confidence_interval_to = [ct1 ct2]
%for s2
ct11 = mbar(2) - sqrt(C(1,1)*z*sqrt(n));
ct22 = mbar(2) + sqrt(C(2,2)*z*sqrt(n));

Confidence_interval_s2 = [ct11 ct22]
```