*//multiplelinear(XA,y,alpha)*

dd="C:\data\Backup HP June 2017\Copy of most important engine build stuff\avgtperci.txt";*//row vector c for linear set of equations f(x)= Ax+c; goal is to solve for x that will minimize f(x)~0*

y=fscanfMat(dd);

ee="C:\data\Backup HP June 2017\Copy of most important engine build stuff\A6All.txt";*// matrix A for equation above each line in the text file is a*

XA=fscanfMat(ee);

alpha= 0.1;

*//ff="C:\Copy of most important engine build stuff\x0.txt";// matrix x0 initial gues of solution, must have as many rows as c and A*

*//x0=fscanfMat(ff);*

*// check matrix sizes for compatibility*

*// [nA, mA] = size (A);*

*//[nx, mx] = size (x0);*

*//[nc, mc] = size (c);*

*// if nA<> nc|mA<> nx then*

*// error ('Incompatible dimensions");*

*//abort;*

*//end;*

[m n] = size(XA); *//Size of original X matrix*

X = [ones(m,1) XA]; *//Augmenting matrix X*

b=inv(X'\*X)\*X'\*y; *//Coefficients of function*

yh = X\*b; *//Fitted value of y*

e=y-yh; *//Errors or residuals*

SSE=e'\*e; *//Sum of squared errors*

MSE = SSE/(m-n-1); *//Mean square error*

se = sqrt(MSE); *//Standard error of estimate*

C = MSE\*inv(X'\*X); *//Covariance matrix*

[nC mC]=size(C);

seb = []; *//Standard errors for coefficients*

for i = 1:nC

seb = [seb; sqrt(C(i,i))];

end;

ta2 = cdft('T',m-n,1-alpha/2,alpha/2); *//t\_alpha/2*

sY = []; sYp = []; *//Terms involved in C.I. for Y, Ypred*

for i=1:m

sY = [sY; sqrt(X(i,:)\*C\*X(i,:)')];

sYp = [sYp; se\*sqrt(1+X(i,:)\*(C/se)\*X(i,:)')];

end;

CIYL = yh-sY; *//Lower limit for C.I. for mean Y*

CIYU = yh+sY; *//Upper limit for C.I. for mean Y*

CIYpL = yh-sYp; *//Lower limit for C.I. for predicted Y*

CIYpU = yh+sYp; *//Upper limit for C.I. for predicted Y*

CIbL = b-ta2\*seb; *//Lower limit for C.I. for coefficients*

CIbU = b+ta2\*seb; *//Upper limit for C.I. for coefficients*

t0b = b./seb; *//t parameter for testing H0:b(i)=0*

decision = []; *//Hypothesis testing for H0:b(i)=0*

for i = 1:n+1

if t0b(i)>ta2 | t0b(i)<-ta2 then

decision = [decision; ' reject '];

else

decision = [decision; ' do not reject'];

end;

end;

ybar = mean(y); *//Mean value of y*

SST = sum((y-ybar)^2); *//Total sum of squares*

SSR = sum((yh-ybar)^2); *//Residual sum of squares*

MSR = SSR/n; *//Regression mean square*

MSE = SSE/(m-n-1); *//Error mean square*

F0 = MSR/MSE; *//F parameter for significance of regression*

Fa = cdff('F',n,m-n-1,1-alpha,alpha); *//F\_alpha*

R2 = 1-SSE/SST; R = sqrt(R2); *//Coeff. of multiple regression*

R2a = 1-(SSE/(m-n-1))/(SST/(m-1)); *//Adj. Coeff. of multiple regression*

*//Printing of results*

printf(' ');

printf('Multiple linear regression');

printf('==========================\n');

printf(' ');

printf('Table of coefficients');

printf('-------------------------------------------------------------------------\n');

printf(' i b(i) se(b(i)) Lower Upper t0 H0:b(i)=0\n');

printf('-------------------------------------------------------------------------\n');

for i = 1:n+1

printf('%4.0f %10g %10g %10g %10g %10g '+decision(i),...

i-1,b(i),seb(i),CIbL(i),CIbU(i),t0b(i));

printf('\n')

end;

printf('-------------------------------------------------------------------------\n');

printf(' t\_alpha/2 = %g\n',ta2);

printf('-------------------------------------------------------------------------\n');

printf(' ');printf(' ');

printf('Table of fitted values and errors\n');

printf('---------------------------------------------------------------------------------\n');

printf(' i y(i) yh(i) e(i) C.I. for Y C.I. for Ypred\n');

printf('---------------------------------------------------------------------------------\n');

for i = 1:m

printf('%4.0f %10.6g %10.6g %10.6g %10.6g %10.6g %10.6g %10.6g',...

i,y(i),yh(i),e(i),CIYL(i),CIYU(i),CIYpL(i),CIYpU(i));

printf('\n')

end;

printf('---------------------------------------------------------------------------------\n');

printf(' ');printf(' ');

printf('-------Analysis of variance-------\n');

printf('--------------------------------------------------------\n');

printf('Source of variation Sum of squares Degrees of freedom Mean square F0\n')

*//printf('variation squares freedom square F0');*

printf('--------------------------------------------------------\n');

printf('Regression %10.6g %10.0f %10.6g %10.6g\n',SSR,n,MSR,F0');

printf('Residual %10.6g %10.0f %10.6g \n',SSE,m-n-1,MSE);

printf('Total %10.6g %10.0f \n',SST,m-1);

printf('--------------------------------------------------------\n');

printf('----With F0 = %g and F\_alpha = %g,',F0,Fa);

if F0>Fa then

printf(' reject the null hypothesis \n');

else

printf(' do not reject the null hypothesis \n');

end;

printf('--------------------------------------------------------\n');

disp(' ');

printf('Additional information\n');

printf('---------------------------------------------------------\n');

printf('Standard error of estimate (se) = %g\n',se);

printf('Coefficient of multiple determination (R^2) = %g\n',R2);

printf('Multiple correlation coefficient (R) = %g\n',R);

printf('Adjusted coefficient of multiple determination = %g\n',R2a);

printf('---------------------------------------------------------\n');

printf(' ');

printf('Covariance matrix:');

disp(C);

printf(' ');

printf('---------------------------------------------------------\n');

*//Plots of residuals - several options*

for j = 1:n

xset('window',j);xset('mark',0,14);*//xbasc(j);*

plot2d(XA(:,j),e,-9)

xtitle('Residual plot - error vs. x'+string(j),'x'+string(j),'error');

end;

xset('window',n+1);xset('mark',0,14);

plot2d(y,e,-9);

xtitle('Residual plot - error vs. y','y','error');

xset('window',n+2);xset('mark',0,14);

plot2d(yh,e,-9);

xtitle('Residual plot - error vs. yh','yh','error');