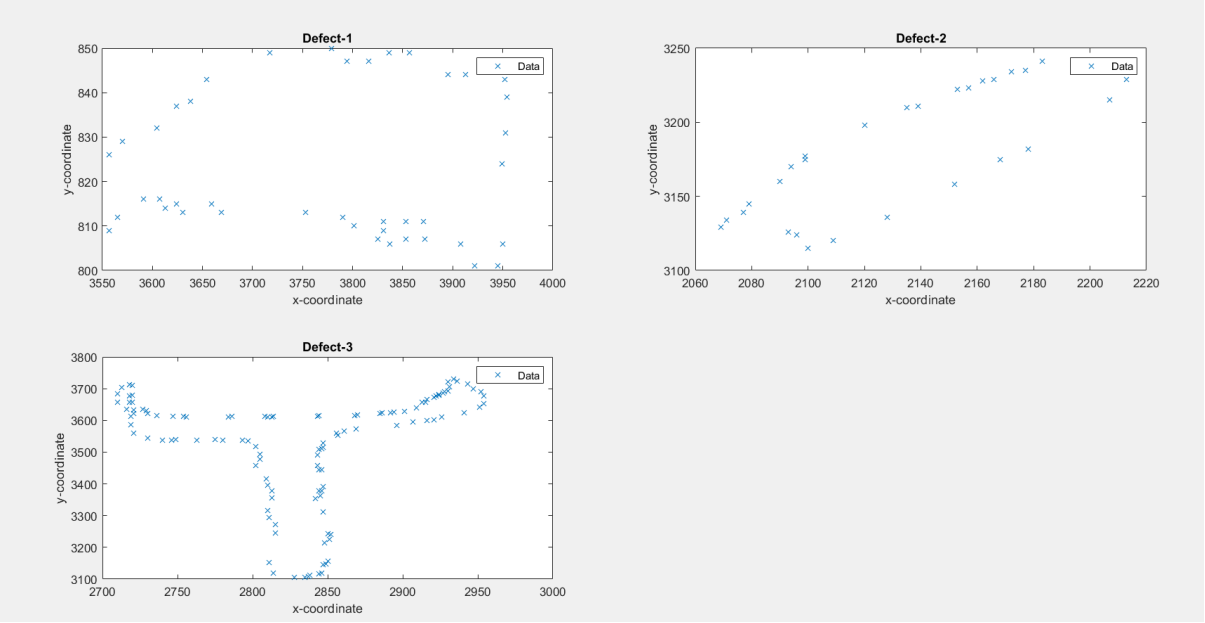
BY S.VISHAL
CH18B020

Plots

Question-4 plots



The above plots agree with the inferences concluded using OLS/TLS.

Defect-1: along horizontal axis

Defect-2: along y = x

Defect-3: along vertical axis

Code

OLS.m (function to perform OLS)

```
function [alpha, beta, uhat, yhat, s] = OLS(u,y)
    N = length(u);
    ybar = mean(y);
    ubar = mean(u);
    syy = var(y,1);
    suu = var(u,1);
    syu = 1/N*sum((y-ybar).*(u-ubar));
    uhat = u;
    alpha = syu/suu;
    beta = ybar - alpha*ubar;
    yhat = alpha*uhat + beta;
    s =
struct('alpha',alpha,'beta',beta,'ybar',ybar,'ubar',ubar,
'syy',syy,'syu',syu,'suu',suu);
end
```

BY S.VISHAL
CH18B020

IOLS.m (function to perform IOLS)

```
function [alpha, beta, uhat, yhat, s] = IOLS(u,y)
    N = length(u);
    ybar = mean(y);
    ubar = mean(u);
    syy = var(y,1);
    suu = var(u,1);
    syu = 1/N*sum((y-ybar).*(u-ubar));
    yhat = y;
    alpha = syy/syu;
    beta = ybar - alpha*ubar;
    uhat = (yhat - beta)/alpha;
    s =
struct('alpha',alpha,'beta',beta,'ybar',ybar,'ubar',ubar,
'syy',syy,'syu',syu,'suu',suu);
end
```

TLS.m (function to perform TLS)

```
function [alpha, beta, uhat, yhat, s] = TLS(u,y)
    N = length(u);
    ybar = mean(y);
    ubar = mean(u);
    syy = var(y,1);
    suu = var(u,1);
    syu = 1/N*sum((y-ybar).*(u-ubar));
    alpha = (syy - suu + sqrt((syy-suu)^2 +
4*syu^2))/2/syu;
    beta = ybar - alpha*ubar;
    uhat = (alpha*(y-beta)+u)/(alpha^2+1);
    yhat = alpha*uhat + beta;
    s =
struct('alpha',alpha,'beta',beta,'ybar',ybar,'ubar',ubar,
'syy',syy,'syu',syu,'suu',suu);
end
```

Question-2

```
clear; close all;
%% Open data
A = readmatrix('CO2.csv');
time = A(:,1);
CO2 = A(:,2);
temp = A(:,3);
temp_cut = 3.6;
%% Predict maximum permissible level of CO2 - OLS
% y -> temperature deviation
% u -> CO2
```

```
u = CO2;
y = temp;
[alpha, beta, uhat, yhat, s] = OLS(u,y);
disp(s);
CO2_cut_OLS = (temp_cut-beta)/alpha;
%% Predict maximum permissible level of CO2 - TLS
[alpha_TLS, beta_TLS, uhat_TLS, yhat_TLS,s_TLS] =
TLS(u,y);
CO2_cut_TLS = (temp_cut-beta_TLS)/alpha_TLS;
%% Predict year
ut = time;
yt = CO2;
[alpha_t, beta_t, uhat_t, yhat_t, s_t] = OLS(ut,yt);
t_OLS = (CO2_cut_OLS-beta_t)/alpha_t;
tpred_OLS = ceil(t_OLS);
t_TLS = (CO2_cut_TLS-beta_t)/alpha_t;
tpred_TLS = ceil(t_TLS);
```

Question-3

```
close all; clear;
%% Data
EP =
[1.98,2.31,3.29,3.56,1.23,1.57,2.05,0.66,0.31,2.82,0.13,3
.15,2.72,2.31,1.92,1.56,0.94,2.27,3.17,2.36]';
CF =
[1.87,2.2,3.15,3.42,1.1,1.41,1.84,0.68,0.27,2.8,0.14,3.2,
2.7,2.43,1.78,1.53,0.84,2.21,3.10,2.34]';
% u - EP, y - CF
u = EP;
y = CF;
N = length(u);
%% Part a)
% OLS
[alpha_OLS, beta_OLS, uhat_OLS, yhat_OLS,s_OLS] =
OLS(u,y);
sigma_e_OLS = 1/(N-2)*sum((y-alpha_OLS*u-beta_OLS).^2);
CI_OLS = [alpha_OLS-
2.16*sigma_e_OLS,alpha_OLS+2.16*sigma_e_OLS];
% IOLS
[alpha_IOLS, beta_IOLS, uhat_IOLS, yhat_IOLS,s_IOLS] =
IOLS(u,y);
sigma_e_IOLS = 1/(N-2)*sum((y-alpha_IOLS*u-
beta_IOLS).^2);
CI_IOLS = [alpha_IOLS-
2.16*sigma_e_IOLS,alpha_OLS+2.16*sigma_e_IOLS];
% TLS
```

```
[alpha_TLS, beta_TLS, uhat_TLS, yhat_TLS,s_TLS] =
TLS(u,y);
sigma_e_TLS = 1/(N-2)*sum((y-alpha_TLS*u-beta_TLS).^2);
CI_TLS = [alpha_TLS-
2.16*sigma_e_TLS,alpha_TLS+2.16*sigma_e_TLS];
%% Part b)
ui = 2.31;
yi = 2.20;
% OLS: u is perfect
OLS_pred = ui;
% IOLS: y is perfect
IOLS_pred = yi;
% TLS: both imperfect. doing a perpendicular projection
will give us an
% estimate for EP and CF. Since we need a single estimate
we will assume
% alpha = 1 and beta = 0. So yhat = uhat = (u+y)/2
TLS_pred = (ui+yi)/2;
```

Question-4

```
clear; close all;
%% Open data
A = readmatrix('defects_annotation_data.csv');
x1 = rem_NaN(A(:,1)); y1 = rem_NaN(A(:,2));
x2 = rem_NaN(A(:,4)); y2 = rem_NaN(A(:,5));
x3 = rem_NaN(A(:,7)); y3 = rem_NaN(A(:,8));
%% Defect-1
N = length(x1);
% TLS
[alpha_TLS, beta_TLS, uhat_TLS, yhat_TLS,s_TLS] =
TLS(x1,y1);
sigma_e_TLS = 1/(N-2)*sum((y1-alpha_TLS*x1-beta_TLS).^2);
CI_TLS = [alpha_TLS-
2.16*sigma_e_TLS,alpha_TLS+2.16*sigma_e_TLS];
% TLS - inverted
[alpha_TLS2, beta_TLS2, uhat_TLS2, yhat_TLS2,s_TLS2] =
TLS(y1,x1);
sigma_e_TLS2 = 1/(N-2)*sum((y1-alpha_TLS2*x1-
beta_TLS2).^2);
CI_TLS2 = [alpha_TLS2-
2.16*sigma_e_TLS2,alpha_TLS2+2.16*sigma_e_TLS2];
% OLS
[alpha_OLS, beta_OLS, uhat_OLS, yhat_OLS,s_OLS] =
OLS(x1,y1);
sigma_e_OLS = 1/(N-2)*sum((y1-alpha_OLS*x1-beta_OLS).^2);
```

```
CI_OLS = [alpha_OLS-
2.16*sigma_e_OLS,alpha_OLS+2.16*sigma_e_OLS];
% OLS - inverted
[alpha_OLS2, beta_OLS2, uhat_OLS2, yhat_OLS2,s_OLS2] =
OLS(y1,x1);
sigma_e_OLS2 = 1/(N-2)*sum((y1-alpha_OLS2*x1-
beta_OLS2).^2);
CI_OLS2 = [alpha_OLS2-
2.16*sigma_e_OLS2,alpha_OLS2+2.16*sigma_e_OLS2];
% Plot
subplot(2,2,1);
plot(x1,y1,'x');
title('Defect-1'); xlabel('x-coordinate'); ylabel('y-
coordinate');
legend('Data');
%% Defect-2
% TLS
[alpha_TLS_def2, beta_TLS_def2, uhat_TLS_def2,
yhat_TLS_def2,s_TLS_def2] = TLS(x2,y2);
sigma_e_TLS_def2 = 1/(N-2)*sum((y2-alpha_TLS_def2*x2-
beta_TLS_def2).^2);
CI_TLS_def2 = [alpha_TLS_def2-
2.16*sigma_e_TLS_def2,alpha_TLS_def2+2.16*sigma_e_TLS_def
2];
% TLS - inverted
[alpha_TLS2_def2, beta_TLS2_def2, uhat_TLS2_def2,
yhat_TLS2_def2,s_TLS2_def2] = TLS(y2,x2);
sigma_e_TLS2_def2 = 1/(N-2)*sum((y2-alpha_TLS2_def2*x2-
beta_TLS2_def2).^2);
CI_TLS2_def2 = [alpha_TLS2_def2-
2.16*sigma_e_TLS2_def2,alpha_TLS2_def2+2.16*sigma_e_TLS2_
def2];
% OLS
[alpha_OLS_def2, beta_OLS_def2, uhat_OLS_def2,
yhat_OLS_def2,s_OLS_def2] = OLS(x2,y2);
sigma_e_OLS_def2 = 1/(N-2)*sum((y2-alpha_OLS_def2*x2-
beta_OLS_def2).^2);
CI_OLS_def2 = [alpha_OLS_def2-
2.16*sigma_e_OLS_def2,alpha_OLS_def2+2.16*sigma_e_OLS_def
2];
% OLS - inverted
[alpha_OLS2_def2, beta_OLS2_def2, uhat_OLS2_def2,
yhat_OLS2_def2,s_OLS2_def2] = OLS(y2,x2);
sigma_e_OLS2_def2 = 1/(N-2)*sum((y2-alpha_OLS2_def2*x2-
beta_OLS2_def2).^2);
```

```
CI_OLS2_def2 = [alpha_OLS2_def2-
2.16*sigma_e_OLS2_def2,alpha_OLS2_def2+2.16*sigma_e_OLS2_
def2];
% Plot
subplot(2,2,2);
plot(x2,y2,'x');
title('Defect-2'); xlabel('x-coordinate'); ylabel('y-
coordinate');
legend('Data');
%% Defect-3
% TLS
[alpha_TLS_def3, beta_TLS_def3, uhat_TLS_def3,
yhat_TLS_def3,s_TLS_def3] = TLS(x3,y3);
sigma_e_TLS_def3 = 1/(N-2)*sum((y3-alpha_TLS_def3*x3-
beta_TLS_def3).^2);
CI_TLS_def3 = [alpha_TLS_def3-
2.16*sigma_e_TLS_def3,alpha_TLS_def3+2.16*sigma_e_TLS_def
3];
% TLS - inverted
[alpha_TLS2_def3, beta_TLS2_def3, uhat_TLS2_def3,
yhat_TLS2_def3,s_TLS2_def3] = TLS(y3,x3);
sigma_e_TLS2_def3 = 1/(N-2)*sum((y3-alpha_TLS2_def3*x3-
beta_TLS2_def3).^2);
CI_TLS2_def3 = [alpha_TLS2_def3-
2.16*sigma_e_TLS2_def3,alpha_TLS2_def3+2.16*sigma_e_TLS2_
def3];
% OLS
[alpha_OLS_def3, beta_OLS_def3, uhat_OLS_def3,
yhat_OLS_def3,s_OLS_def3] = OLS(x3,y3);
sigma_e_OLS_def3 = 1/(N-2)*sum((y3-alpha_OLS_def3*x3-
beta_OLS_def3).^2);
CI_OLS_def3 = [alpha_OLS_def3-
2.16*sigma_e_OLS_def3,alpha_OLS_def3+2.16*sigma_e_OLS_def
3];
% OLS - inverted
[alpha_OLS2_def3, beta_OLS2_def3, uhat_OLS2_def3,
yhat_OLS2_def3,s_OLS2_def3] = OLS(y3,x3);
sigma_e_OLS2_def3 = 1/(N-2)*sum((y2-alpha_OLS2_def3*x2-
beta_OLS2_def3).^2);
CI_OLS2_def3 = [alpha_OLS2_def3-
2.16*sigma_e_OLS2_def3,alpha_OLS2_def3+2.16*sigma_e_OLS2_
def3];
% Plot
subplot(2,2,3);
plot(x3,y3,'x');
title('Defect-3'); xlabel('x-coordinate'); ylabel('y-
coordinate');
```

```
legend('Data');
%% function to remove NaN values
function vnew = rem_NaN(v)
    vnew = v(~isnan(v));
end
```