```
clc
close all

% this script is for smoothing velocity
read_data

v1 = [];
t = [];
j = 1;
for i = 1:(length(noisy_flight1)-1)
    v1(j) = (noisy_flight1(i+1) - noisy_flight1(i))/0.01;
    t(j) = mtime1(i);
    j = j+1;
end

% take the moving average the velocity
sma_v1 = [];
N = 200; % number of data points in moving average
for i = N:length(v1)
    sma_v1(i) = mean(v1(i-N+1:i));
end
% from 10 to the length
% take the average from 1 to 10

% take the exponential moving average
smoothing = 0.3; % this is the smoothing factor applied to the data
ema_v1 = zeros(1,length(v1));
for i = N:length(v1)
    ema_v1(i) = smoothing*ema_v1(i-1) + (1-0.3)*sma_v1(i-2);
end

[sma_v1,ema_v1,t1,v1] = moving_average(noisy_flight1,mtime1);
[sma_v2,ema_v2,t2,v2] = moving_average(noisy_flight2,mtime2);
[sma_v3,ema_v3,t3,v3] = moving_average(noisy_flight3,mtime3);
[sma_v4,ema_v4,t4,v4] = moving_average(noisy_flight4,mtime4);

figure
plot(mtime1,m_velocity1,'g')
hold on
plot(mtime2,m_velocity2,'b')
plot(mtime3,m_velocity3,'r')
plot(mtime4,m_velocity4,'c')

plot(mtime1(2:length(mtime1)),v1,'g')
plot(mtime2(2:length(mtime2)),v2,'b')
plot(mtime3(2:length(mtime3)),v3,'r')
plot(mtime4(2:length(mtime4)),v4,'c')

xlabel('Time (s)','FontSize',16)
ylabel('Velocity (ft/s)','FontSize',16)
title('Velocity from Noisy Data','FontSize',20)
legend('TL 22','CL 22','TL 23','CL 23')
```

```
hold off


% plot the unadjusted filtered velocities
figure
plot(mtime1,m_velocity1,'g')
hold on
plot(mtime2,m_velocity2,'b')
plot(mtime3,m_velocity3,'r')
plot(mtime4,m_velocity4,'c')
plot(t(N:length(t1)),sma_v1(N:length(sma_v1)),'g')

%plot(t,v1,'r')
%plot(t(N+2:length(t)),ema_v1(N+2:length(ema_v1)),'c')


plot(t(N:length(t2)),sma_v2(N:length(sma_v2)),'b')


plot(t(N:length(t3)),sma_v3(N:length(sma_v3)),'r')


plot(t(N:length(t4)),sma_v4(N:length(sma_v4)),'c')

xlabel('Time (s)','FontSize',16)
ylabel('Velocity (ft/s)','FontSize',16)
title('Actual Velocity and Unadjusted Filtered Velocity','FontSize',20)
legend('TL 22','CL 22','TL 23','CL 23')
hold off

% create mega velocity vector
v = [m_velocity1(N:length(m_velocity1)-1) m_velocity2(N:length(m_velocity2)-1)
 m_velocity3(N:length(m_velocity3)-1) m_velocity4(N:length(m_velocity4)-1)];
v_sma = [sma_v1(N:length(sma_v1)) sma_v2(N:length(sma_v2))
 sma_v3(N:length(sma_v3)) sma_v4(N:length(sma_v4))];
figure
scatter(v_sma,v,'b');
xlabel('SMA Velocity','FontSize',16)
ylabel('Actual Velocity','FontSize',16)
title('SMA Velocity v. Actual Velocity','FontSize',20)

vmodel = polyfit(v_sma,v,1);

% adjust the sma of velocities using the model
sma_v1_adj = adj_v(sma_v1,vmodel);
sma_v2_adj = adj_v(sma_v2,vmodel);
sma_v3_adj = adj_v(sma_v3,vmodel);
sma_v4_adj = adj_v(sma_v4,vmodel);

figure
plot(mtime1,m_velocity1,'g')
hold on
plot(mtime2,m_velocity2,'b')
plot(mtime3,m_velocity3,'r')
```

```
plot(mtime4,m_velocity4,'c')
plot(t(N:length(t1)),sma_v1_adj(N:length(sma_v1_adj)),'g')

plot(t(N:length(t2)),sma_v2_adj(N:length(sma_v2_adj)),'b')

plot(t(N:length(t3)),sma_v3_adj(N:length(sma_v3_adj)),'r')

plot(t(N:length(t4)),sma_v4_adj(N:length(sma_v4_adj)),'c')

xlabel('Time (s)','FontSize',16)
ylabel('Velocity (ft/s)','FontSize',16)
title('Actual Velocity and Adjusted Filtered Velocity','FontSize',20)
legend('TL 22','CL 22','TL 23','CL 23')
hold off

figure
plot(t(N:length(t1)),sma_v1_adj(N:length(sma_v1_adj)).^2,'g')
hold on
plot(t(N:length(t2)),sma_v2_adj(N:length(sma_v2_adj)).^2,'b')
plot(t(N:length(t3)),sma_v3_adj(N:length(sma_v3_adj)).^2,'r')
plot(t(N:length(t4)),sma_v4_adj(N:length(sma_v4_adj)).^2,'c')
xlabel('Time')
ylabel('v^2')
legend('TL 22','CL 22','TL 23','CL 23')
hold off



function [sma,ema,t,v] = moving_average(noisy_flight,mtime)
N = 200; % number of points in the moving average
v = []; % will become the noisy flight velocity
t = [];
j = 1;
for i = 1:(length(noisy_flight)-1)
    v(j) = (noisy_flight(i+1) - noisy_flight(i))/0.01;
    t(j) = mtime(i);
    j = j+1;
end
sma = [];
N = 200; % number of data points in moving average
for i = N:length(v)
    sma(i) = mean(v(i-N+1:i));
end
smoothing = 0.3; % this is the smoothing factor applied to the data
ema = zeros(1,length(v));

for i = N:length(v)
    ema(i) = smoothing*ema(i-1) + (1-0.3)*sma(i-2);
end

end

function sma_adj = adj_v(sma,model)
% adjusts simple moving average of v using polyfit model
```

```
% sma is the simple moving average vector
% model is the first order model

sma_adj = sma*model(1) + model(2);

end
```

Warning: Column headers from the file were modified to make them valid MATLAB identifiers before creating variable names for the table. The original column headers are saved in the VariableDescriptions property. Set 'VariableNamingRule' to 'preserve' to use the original column headers as table variable names.