```
% THIS SCRIPT IS USED TO READ DATA FROM THE RRC3 CSV TO SOME VECTORS

clc
clear
close all

data = readtable('Flight_Data.csv');

time = table2array(data(:,1));
altitude = table2array(data(:,2));
velocity = table2array(data(:,4));
flight = table2cell(data(:,9));

L = length(time);

time1 = [];
time2 = [];
time3 = [];
time4 = [];

w = 1;
x = 1;
y = 1;
z = 1;

for i = 1:L
    launch = flight(i);
    if char(launch) == 'TL 22'
        time1(w) = time(i);
        altitude1(w) = altitude(i);
        rrc3_v1(w) = velocity(i);
        w = w + 1;
    elseif char(launch) == 'CL 22'
        time2(x) = time(i);
        altitude2(x) = altitude(i);
        rrc3_v2(x) = velocity(i);
        x = x + 1;
    elseif char(launch) == 'TL 23'
        time3(y) = time(i);
        altitude3(y) = altitude(i);
        rrc3_v3(y) = velocity(i);
        y = y + 1;
    else
        time4(z) = time(i);
        altitude4(z) = altitude(i);
        rrc3_v4(z) = velocity(i);
        z = z + 1;
    end

end
```

```
mtime1 = linspace(8,28,(28-8)*100);
mtime2 = linspace(8,28,(28-8)*100);
mtime3 = linspace(8,28,(28-8)*100);
mtime4 = linspace(8,28,(28-8)*100);
% create simulated number of data points equivalent to 100 Hz readings

model1 = polyfit(time1,altitude1,3);
model2 = polyfit(time2,altitude2,3);
model3 = polyfit(time3,altitude3,3);
model4 = polyfit(time4,altitude4,3);

m_altitude1 = model1(1)*mtime1.^3 + model1(2)*mtime1.^2 + model1(3)*mtime1 +
 model1(4);
m_altitude2 = model2(1)*mtime2.^3 + model2(2)*mtime2.^2 + model2(3)*mtime2 +
 model2(4);
m_altitude3 = model3(1)*mtime3.^3 + model3(2)*mtime3.^2 + model3(3)*mtime3 +
 model3(4);
m_altitude4 = model4(1)*mtime4.^3 + model4(2)*mtime4.^2 + model4(3)*mtime4 +
 model4(4);

m_velocity1 = 3*model1(1)*mtime1.^2 + 2*model1(2)*mtime1 + model1(3);
m_velocity2 = 3*model2(1)*mtime2.^2 + 2*model2(2)*mtime2 + model2(3);
m_velocity3 = 3*model3(1)*mtime3.^2 + 2*model3(2)*mtime3 + model3(3);
m_velocity4 = 3*model4(1)*mtime4.^2 + 2*model4(2)*mtime4 + model4(3);

% simulate noisy data
noise_factor = 15; % higher the number, the noisier the data

for i = 1:length(m_altitude1)
    if rand() > 0.5
        noisy_flight11(i) = m_altitude1(i) + noise_factor*rand();
    else
        noisy_flight11(i) = m_altitude1(i) - noise_factor*rand();
    end
end

for i = 1:length(m_altitude2)
    if rand() > 0.5
        noisy_flight2(i) = m_altitude2(i) + noise_factor*rand();
    else
        noisy_flight2(i) = m_altitude2(i) - noise_factor*rand();
    end
end

for i = 1:length(m_altitude3)
    if rand() > 0.5
        noisy_flight3(i) = m_altitude3(i) + noise_factor*rand();
    else
        noisy_flight3(i) = m_altitude3(i) - noise_factor*rand();
    end
end

for i = 1:length(m_altitude4)
    if rand() > 0.5
```

```
        noisy_flight4(i) = m_altitude4(i) + noise_factor*rand();
    else
        noisy_flight4(i) = m_altitude4(i) - noise_factor*rand();
    end
end

noisy_flight1 = nosify(m_altitude1);
noisy_flight2 = nosify(m_altitude2);
noisy_flight3 = nosify(m_altitude3);
noisy_flight4 = nosify(m_altitude4);

plot(mtime1,noisy_flight11,'r')
hold on
plot(mtime1,noisy_flight1,'b')
legend('Bad Model','Better Model')
xlabel('Time into Flight (s)')
ylabel('Altitude')
hold off

%bad_residual = noisy_flight11 - m_altitude1;
%good_residual = noisy_flight1 - m_altitude1;

%figure
%histogram(bad_residual)
%title('Histogram of Residuals with Random Noise','FontSize',20)
%xlabel('Noisy Altitude - Real Altitude (ft)')
%ylabel('Quantity of Residuals')

%figure
%histogram(good_residual)
%title('Histogram of Residuals with Random Gaussian Noise','FontSize',20)
%xlabel('Noisy Altitude - Real Altitude')
%ylabel('Quantity of Residuals (ft)')


function noisy_flight = nosify(m_altitude)
% m_altitude is the modeled altitude
% sigma is the standard deviation of measurements
% std dev is 6.448 if 99% of values fall within +/- 15 ft of actual

sigma = 0; % standard deviation of measurements
noisy_flight = zeros(1,length(m_altitude));
for i = 1:length(m_altitude)
    mean = m_altitude(i);
    r = rand(); % generate random point
    z = -sqrt(2) * erfcinv(r*2); % generate z-score based on random value
    noisy_flight(i) = mean + z*sigma;
end

end
```

Warning: Column headers from the file were
modified to make them valid MATLAB identifiers
before creating variable names for the table.