```
read_data

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

m_altitude1 = model1(1)*time1.^3 + model1(2)*time1.^2 + model1(3)*time1 +
 model1(4);
m_altitude2 = model2(1)*time2.^3 + model2(2)*time2.^2 + model2(3)*time2 +
 model2(4);
m_altitude3 = model3(1)*time3.^3 + model3(2)*time3.^2 + model3(3)*time3 +
 model3(4);
m_altitude4 = model4(1)*time4.^3 + model4(2)*time4.^2 + model4(3)*time4 +
 model4(4);


figure % figure 1

scatter(time1,altitude1,'g')
hold on

scatter(time2,altitude2,'r')


scatter(time3,altitude3,'b')


scatter(time4,altitude4,'c')


plot(time1,m_altitude1,'k','LineWidth',2)
plot(time2,m_altitude2,'k','LineWidth',2)
plot(time3,m_altitude3,'k','LineWidth',2)
plot(time4,m_altitude4,'k','LineWidth',2)

hold off

legend('TL 22','CL 22','TL 23','CL 23','Location','Southeast','FontSize',16)
xlabel('Time (s)','FontSize',16)
ylabel('Altitude (ft)','FontSize',16)
title('Post-Burnout RRC3 Altitude Data for CRT Flights','FontSize',20)
%legend('Test Launch 22','Competition Launch 22','Test Launch 23','Competition
 Launch 23','Location','southeast','FontSize',16)

hold off

vmodel1 = [3*model1(1) 2*model1(2) model1(3)];
vmodel2 = [3*model2(1) 2*model2(2) model2(3)];
vmodel3 = [3*model3(1) 2*model3(2) model3(3)];
vmodel4 = [3*model4(1) 2*model4(2) model4(3)];
```

```
m_velocity1 = vmodel1(1)*time1.^2 + vmodel1(2)*time1 + vmodel1(3);
m_velocity2 = vmodel2(1)*time2.^2 + vmodel2(2)*time2 + vmodel2(3);
m_velocity3 = vmodel3(1)*time3.^2 + vmodel3(2)*time3 + vmodel3(3);
m_velocity4 = vmodel4(1)*time4.^2 + vmodel4(2)*time4 + vmodel4(3);

figure % figure 2
plot(time1,m_velocity1,'g','LineWidth',2)

hold on
plot(time2,m_velocity2,'r','LineWidth',2)
plot(time3,m_velocity3,'b','LineWidth',2)
plot(time4,m_velocity4,'c','LineWidth',2)

xlabel('Time into Flight (s)')
ylabel('Velocity (ft/s)')
title('Velocity v. Time into Flight for CRT Flights','FontSize',16)
legend('Test Launch 22','Competition Launch 22','Test Launch 23','Competition
 Launch 23','Location','northeast','FontSize',12)

hold off

R = 287; % ideal gas constant of air
g0 = 9.81; % gravitational constant
a = -0.0065;
rho_ref = 1.225;
T_ref = 288.15;

% densities for altitudes
rho1 = (1/16.0185)*(1+(a*(altitude1*0.3048)/T_ref)).^((-g0/(a*R))-1); % ft/
lb^3
rho2 = (1/16.0185)*(1+(a*(altitude2*0.3048)/T_ref)).^((-g0/(a*R))-1);
rho3 = (1/16.0185)*(1+(a*(altitude3*0.3048)/T_ref)).^((-g0/(a*R))-1);
rho4 = (1/16.0185)*(1+(a*(altitude4*0.3048)/T_ref)).^((-g0/(a*R))-1);

% distance to apogee
d1 = 10700 - altitude1; % ft
d2 = 10345 - altitude2;
d3 = 12315 - altitude3;
d4 = 10067 - altitude4;



TL_t = [time1 time3]; % single matrix for test launch times
CL_t = [time2 time4]; % single matrix for comp launch times
TL_v = [m_velocity1 m_velocity3]; % single matrix for test launch velocities
CL_v = [m_velocity2 m_velocity4]; % single matrix for competition launch
 velocities
TL_d = [d1 d3]; % single matrix for test launch distances to apogee
CL_d = [d2 d4]; % single matrix for comp launch distances to apogee
TL_a = [m_altitude1 m_altitude3];
CL_a = [m_altitude1 m_altitude3];
TL_rrc3v = [rrc3_v1 rrc3_v3]; % filtered velocity from test launch rrc3's
CL_rrc3v = [rrc3_v2 rrc3_v4]; % filtered velocity from comp launch rrc3's
```

```
v = linspace(0,750,750*10);

TL_model = polyfit(TL_v,TL_d,4);
CL_model = polyfit(CL_v,CL_d,4);

TL_model_prediction = run_model(TL_model,v); % test launch model curve for
 graph
CL_model_prediction = run_model(CL_model,v); % competition launch model curve
 for graph

TL_model_prediction = TL_model(1)*v.^4 + TL_model(2)*v.^3 + TL_model(3)*v.^2+
 TL_model(4)*v + TL_model(5);
CL_model_prediction = CL_model(1)*v.^4 + CL_model(2)*v.^3 + CL_model(3)*v.^2+
 CL_model(4)*v + CL_model(5);

figure % figure 3
scatter(m_velocity1,d1,'g','o')
hold on
scatter(m_velocity2,d2,'r','o')
scatter(m_velocity3,d3,'b','*')
scatter(m_velocity4,d4,'c','*')
plot(v,TL_model_prediction,'k')
plot(v,CL_model_prediction,'k')

xlabel('Velocity (ft/s)','FontSize',16)
ylabel('Distance to Apogee (ft)','FontSize',16)
title('Velocity v. Distance to Apogee','FontSize',20)
legend('Test Launch 22','Competition Launch 22','Test Launch 23','Competition
 Launch 23','Location','southeast','FontSize',12)

hold off

%TLMP = TL_model(1)*TL_v.^3 + TL_model(2)*TL_v.^2+ TL_model(3)*TL_v +
 TL_model(4); % predictions for every TL velocity data point
%CLMP = CL_model(1)*CL_v.^3 + CL_model(2)*CL_v.^2+ CL_model(3)*CL_v +
 CL_model(4); % predictions for every CL velocity data point
%TLMP = TL_model(1)*TL_v.^4 + TL_model(2)*TL_v.^3 + TL_model(3)*TL_v.^2+
 TL_model(4)*TL_v + TL_model(5);
%CLMP = CL_model(1)*CL_v.^4 + CL_model(2)*CL_v.^3 + CL_model(3)*CL_v.^2+
 CL_model(4)*CL_v + CL_model(5);

v1 = [];
v2 = [];
v3 = [];
v4 = [];

% obtain actual velocities from RRC3 rather than ones from derivative of
% fit
for i = 1:(length(time1)-1)
    v1(i) = (altitude1(i+1) - altitude1(i))/0.05;
end

for i = 1:(length(time2)-1)
    v2(i) = (altitude2(i+1) - altitude2(i))/0.05;
```

```
end

for i = 1:(length(time3)-1)
    v3(i) = (altitude3(i+1) - altitude3(i))/0.05;
end

for i = 1:(length(time4)-1)
    v4(i) = (altitude4(i+1) - altitude4(i))/0.05;
end

TL_vexp = [v1 v3];
CL_vexp = [v2 v4];

TLMP = TL_model(1)*TL_rrc3v.^4 + TL_model(2)*TL_rrc3v.^3 +
 TL_model(3)*TL_rrc3v.^2+ TL_model(4)*TL_rrc3v + TL_model(5);
CLMP = CL_model(1)*CL_rrc3v.^4 + CL_model(2)*CL_rrc3v.^3 +
 CL_model(3)*CL_rrc3v.^2+ CL_model(4)*CL_rrc3v + CL_model(5);

TL_residuals = TL_d - TLMP;
CL_residuals = CL_d - CLMP;

Altitude_residuals = altitude1 - m_altitude1;

figure
scatter(TL_v,TL_residuals)
xlabel('Velocity (ft)')
ylabel('Actual Distance to Apogee - Predicted Distance to Apogee (ft)')
title('Residuals v. Live Velocity','FontSize',20)

figure
scatter(TL_rrc3v,TL_v)
xlabel('Filtered RRC3 Velocity Readings (ft/s)')
ylabel('Modeled Velocity (ft/s)')

function model_prediction = run_model(model,data)
    model_prediction = model(1)*data.^3 + model(2)*data.^2 + model(1)*data
 +model(4);

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.
```