Scientist #3: Code Workflow Explanation.

Workflow:

The required libraries were introduced, numpy, math and matplotlib. Then from the ex3 jupyter notebook file, the mass of the rocket without fuel, the mass of the fuel and the combined masses or the total mass of the rocket at its initial phase. These are random numbers generated based on a seed. Then the rest of the parameters were defined like gravity and drag coefficients, etc. To get the area for the drag force calculation, a basic area calculation was performed. The last step before the first while loop was to initialize lists and the final initial values. For this problem it was ideal to split the numerical calculations / iterations into two different while loops, each for one of the two stages of the rocket launch. Stage one is the while the rocket still has fuel. This is the condition for the while loop. Then a change in mass is defined using time and the rate of fuel from the problem specifications. Next the drag force, gravity force and thrust force is calculated and combined into a total net force. This net force is then converted into acceleration by dividing out the mass. Finally, the Euler Numerical iterations occur by updating the velocity based on the acceleration and change in time as well as updating the altitude based on the velocity and change in time. The masses of each the total mass of the rocket and just the fuel is then decremented using the change in mass. The last step in the while loop is to append the lists of time, velocity, and altitude. From here stage two is entered. This is the point from where the thrust is cut off until the rocket hits the ground. So, the while loop condition is while the rocket is above the ground or height is greater than or equal to zero. At the beginning of the loop there is an if statement checking if the velocity is negative to set the sign of the air drag to always oppose the motion of the rocket. The drag and gravity forces are calculated and combined to create a net force. This net force had the mass divided out again to give a net acceleration. Then the Euler numerical methods are implemented again, updating the velocity using acceleration and the change in time and then updating the altitude using the velocity and change in time. The last step is to append the lists once again. From here there are a few last lines to finish up the program. The maximum height, maximum velocity and time taken to achieve terminal velocity are all calculated and printed. Then the plots are made, first a plot of Velocity vs Time with a point plotted at max velocity and then a plot of Altitude vs Time with a max altitude point.

References:

- John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib Development team, 2002-2012, Matplotlib Documentation, Version 3, 3.4, https://matplotlib.org/stable/contents.html