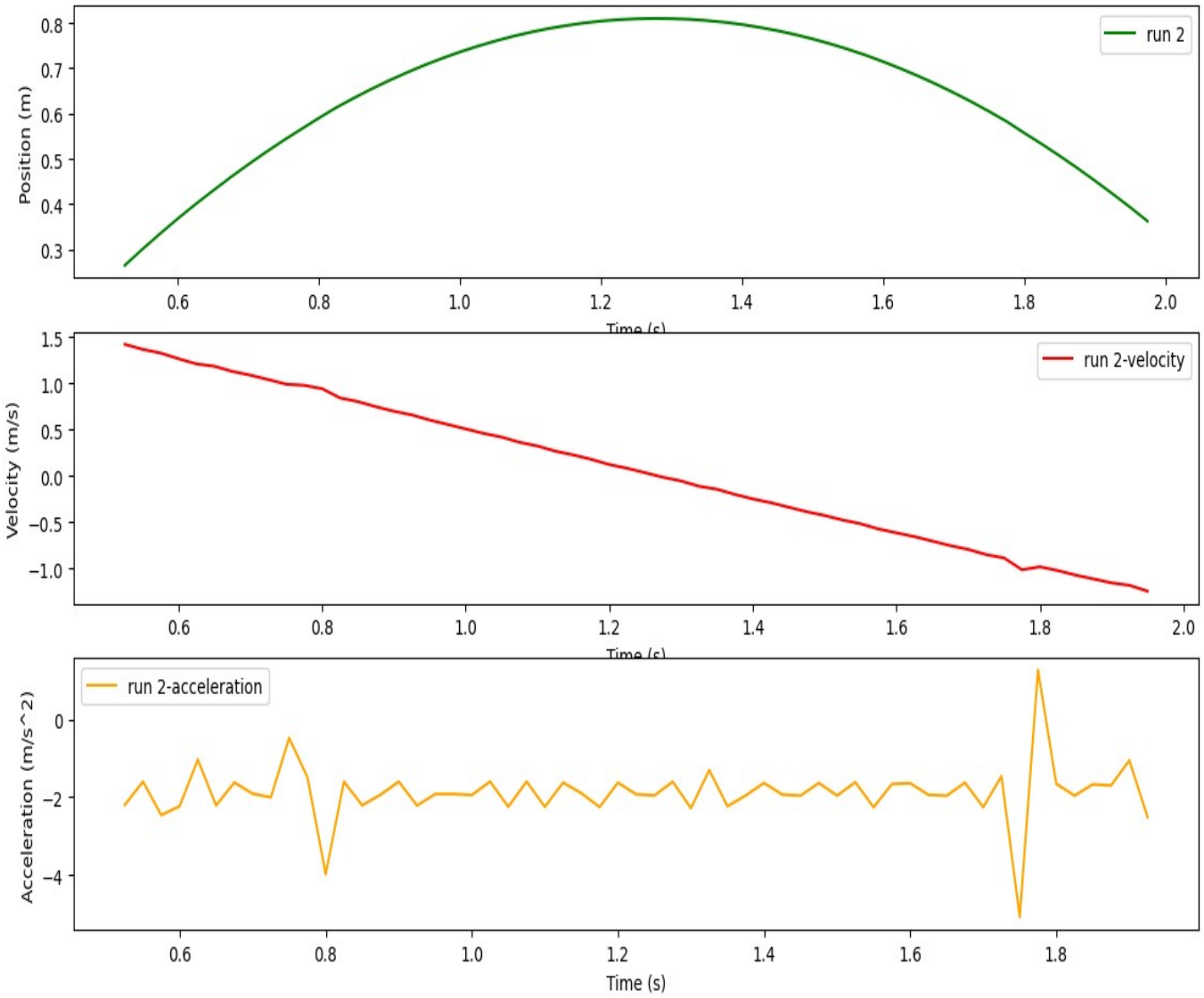
```
import matplotlib.pyplot as pyplot
import numpy as np

# This is a mess but it works
fig, axes = pyplot.subplots(3,1)
def plot_trimmed(path: str, label: str, mintime: float, minpos: float, maxtime: float) -> None:
pyplot.subplot(3,1,1)
data = np.loadtxt(path, delimiter=",", skiprows=1) # does this path syntax work on windows? who knows
data = data[ # not sure if there's a "smarter" way of doing this
(data[:, 0] > mintime) &
(data[:, 1] > minpos) &
(data[:, 0] < maxtime)
]
pos_times = data[:, 0]
positions = data[:, 1]
pyplot.plot(pos_times, positions, color = "green", label=label)

velocity_point_count = pos_times.size-1
velocities = np.ndarray((velocity_point_count,)) # initialize array with shape N-1
velo_times = np.ndarray((velocity_point_count,))
for i in range(0,velocity_point_count):
velo_times[i] = pos_times[i]

velocities[i] = (positions[i+1]-positions[i])/((pos_times[i+1]-pos_times[i]))

pyplot.subplot(3,1,2)
pyplot.plot(velo_times, velocities, color = "red", label=label+"-velocity")

accel_point_count = velocity_point_count-1
accels = np.ndarray((accel_point_count,))
accel_times = np.ndarray((accel_point_count,))
for i in range(0,accel_point_count):
accel_times[i] = (pos_times[i])

accels[i] = ((velocities[i+1]-velocities[i])/((velo_times[i+1]-velo_times[i])))
avg_accel = accels.mean()
std_accel = accels.std() / np.sqrt(accel_point_count)

print("Avg acceleration = " + str(np.round(avg_accel,1)) + "+-" + str(np.round(std_accel,2)) + "m/s^2")
pyplot.subplot(3,1,3)
pyplot.plot(accel_times, accels, color = "orange", label=label+"-acceleration")

# plot_trimmed("./data/run1.csv", "run 1", .45, .1, 2.5)
plot_trimmed("./data/run2.csv", "run 2", .5, .1, 2)
# plot_trimmed("./data/run3.csv", "run 3", 1, .1, 3)
```

```
axes[0].set(xlabel= "Time (s)", ylabel="Position (m)")
axes[1].set(xlabel= "Time (s)", ylabel="Velocity (m/s)")
axes[2].set(xlabel= "Time (s)", ylabel="Acceleration (m/s^2)")
axes[0].legend()
axes[1].legend()
axes[2].legend()
pyplot.show()
```



The acceleration looks mostly constant. Though the line is very jagged most peaks are accompanied by trough which even out when taking the average which ends up being $a_{avg} = -1.9 \pm 0.1 \frac{m}{s^2}$. This can be seen more easily in the slope of the velocity line which has some minor irregularity but is overall linear. I ended up having to trim my data slightly more aggressively because I had previously left in a very slight change in position (which I didn't notice) at either end that compounded very severely once I had calculated acceleration. Wish I could've done this in LaTeX but I don't have it installed on my laptop :(