**Code print out via text file:**

import matplotlib.pyplot as plt, numpy as np

# Create Data with noise, plot

x = np.linspace(0.,50.,num=50)

y = np.random.normal(0,5,50) + 2\*x

# Do least squares to get y\_prime

c = (np.mean(y)\*np.mean(np.power(x,2)) - np.mean(x)\*np.mean(x\*y))/(np.mean(np.power(x,2))-np.power(np.mean(x),2))

m = (np.mean(x\*y) - np.mean(x)\*np.mean(y))/(np.mean(np.power(x,2)) - np.power(np.mean(x),2))

# Plot y\_prime vs x

y\_prime = m\*x + c

plt.plot(x,y\_prime,label = "My Least Square Line", linewidth = '5',color = 'black')

plt.xlabel("x")

plt.ylabel("y")

plt.title("Least Squares Methods")

# Pseudo-inverse

A = np.array([x,np.ones(50)]).T

myPInvA = np.linalg.inv((A.T).dot(A)).dot(A.T)

myPInvM,myPInvC = myPInvA.dot(y)

plt.plot(x,myPInvM\*x+myPInvC,linestyle = 'dotted',label = "Solving pinv Line", linewidth = '10',color = 'blue')

# linalg.pinv

pInvA = np.linalg.pinv(A)

pInvM,pInvC = pInvA.dot(y)

plt.plot(x,pInvM\*x+pInvC,linestyle = 'dashdot',label = "linalg.pinv Line", linewidth = '4',color = 'red')

# linalg.lstsq

linalgM, linalgC = np.linalg.lstsq(A,y)[0]

plt.plot(x,linalgM\*x+linalgC,linestyle = 'dashed',label = "lstsq Line", linewidth = '3', color = 'pink')

# plot data, move legend, and show plot

plt.plot(x,y,"o",label = "Data",color = 'green')

plt.legend(bbox\_to\_anchor=(.5, .9),

bbox\_transform=plt.gcf().transFigure)

plt.show()

**My Plot:**

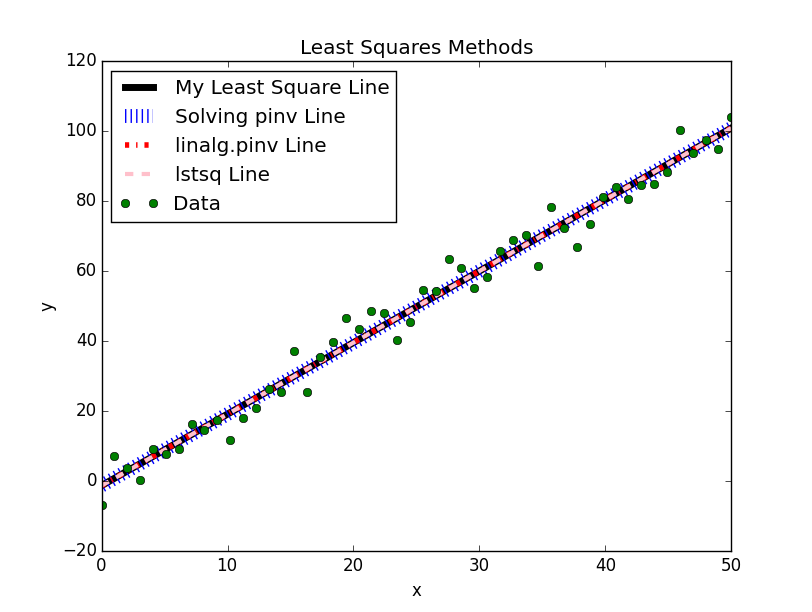
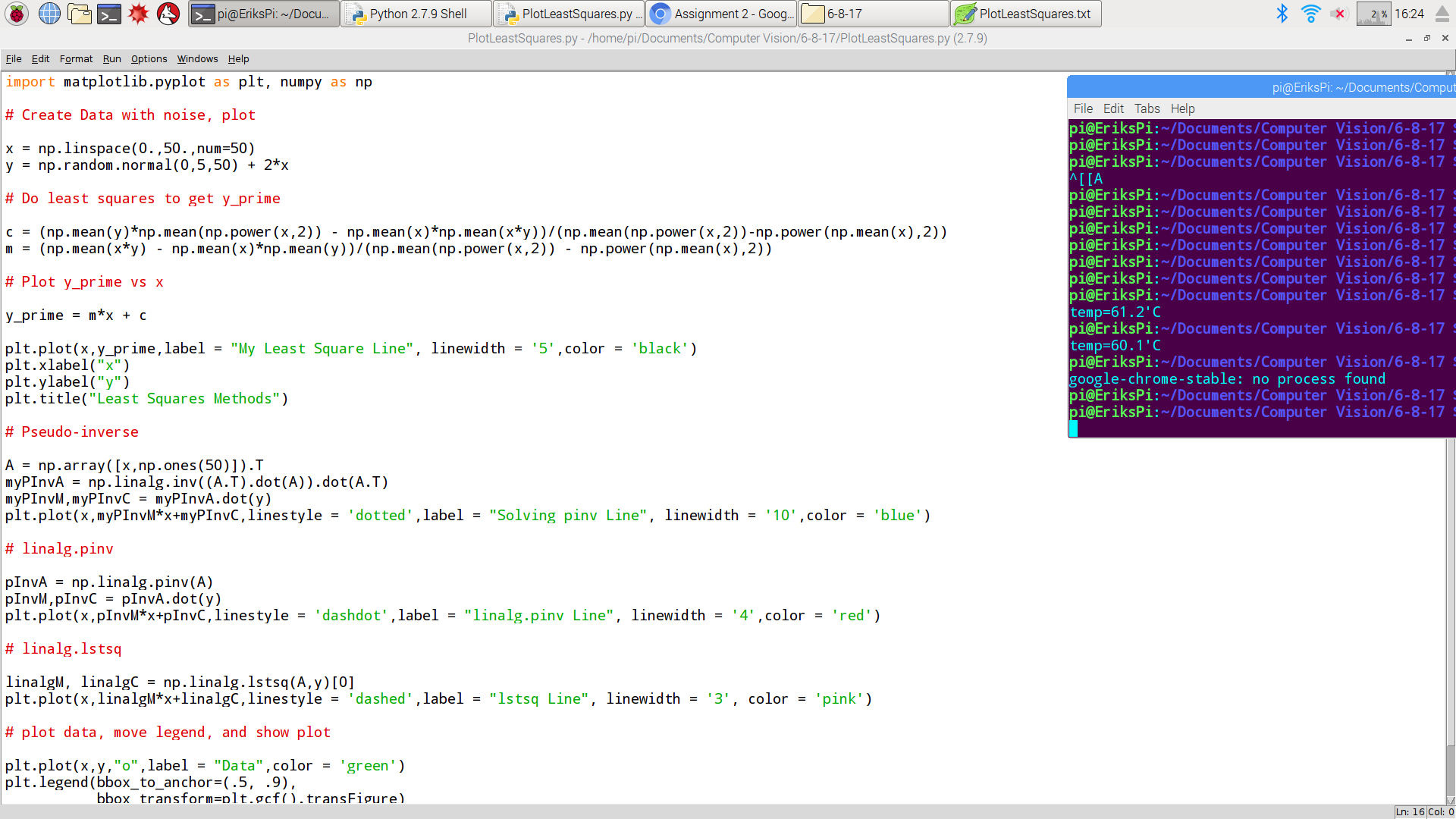
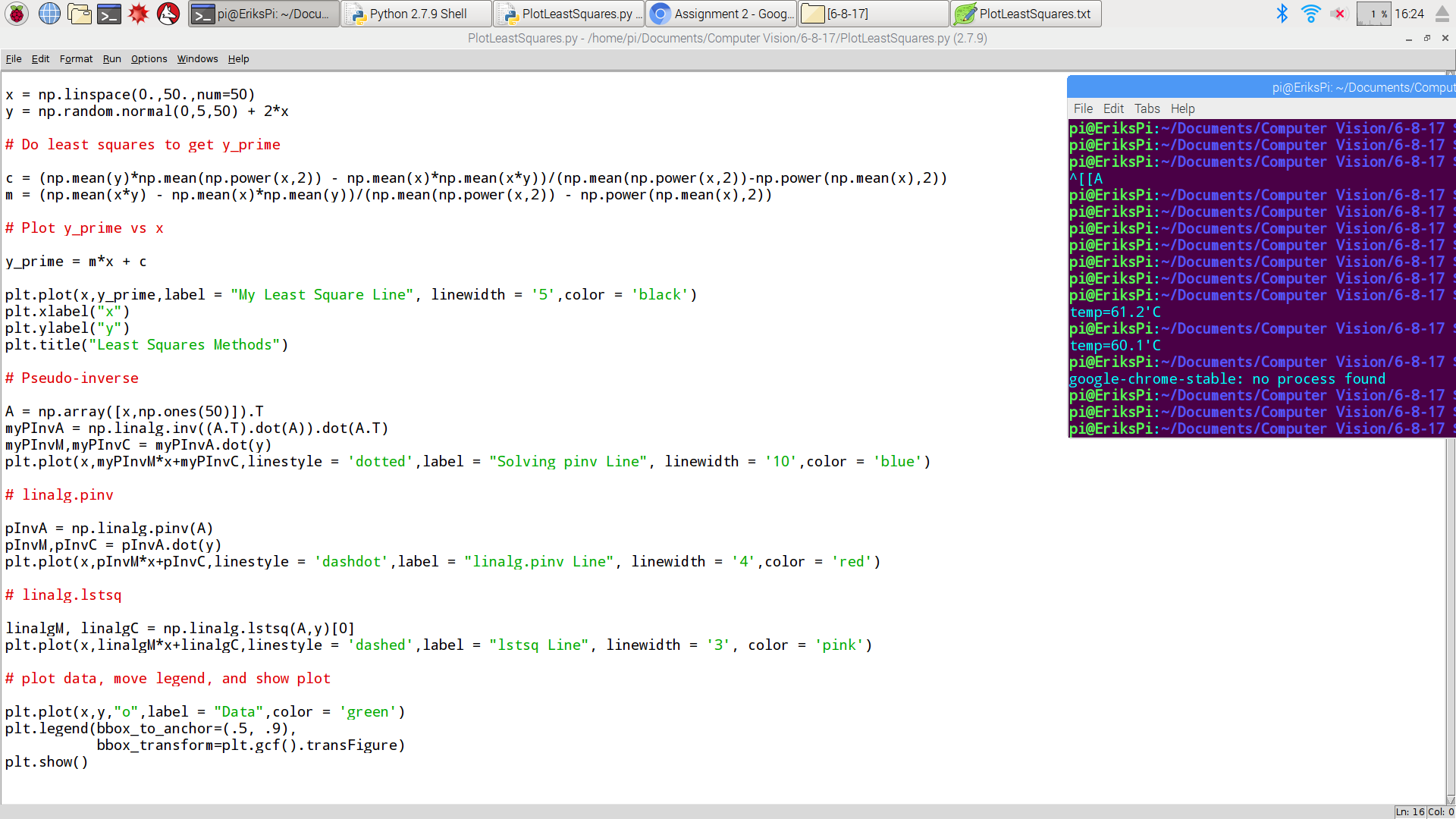
****

Figure 1: Four lines plotted on top of one another with different line styles, widths, and colors to show that each line is roughly equal to the other lines

**Screenshot of Code for Additional Clarity:**

****

****