```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import matplotlib.font_manager as fm
from pylab import cm
from sklearn.linear_model import LinearRegression

#Get Data
I, P = np.loadtxt('CSV/data_transducer_mA_psi.csv', unpack=True,
delimiter=',',skiprows=0)


I_linSpace = np.linspace(I.min(), I.max(), I.size)
degreeOfPoly = 1
m, b = np.polyfit(I, P, degreeOfPoly) #np.poltfit(I, P, degreeOfPoly)
P_linReg = (m * I_linSpace) + b

I_colVec = I.reshape((-1,1))
P_LR = LinearRegression().fit(I_colVec, P)
P_linReg_rSquared = P_LR.score(I_colVec, P)
print("m = ", m, "\tb = ", b, "\tr^2 = ", P)
#Make Figure and plot it
fig = plt.figure(figsize=(6,6))
left, bottom, width, height = [0.15,0.1,0.80,.85]
ax = fig.add_axes( [left, bottom, width, height] )
thicc = 0.5
P_color = 'r'
# eq15_color = 'b'
# eq16_color = 'g'
ax.plot(I, P, '.',label='$PressureDrop[psi]$', color=P_color)
ax.plot(I_linSpace, P_linReg, linestyle='-',color=P_color,linewidth=thicc)



# Edit the major and minor ticks of the x and y axes
ax.xaxis.set_tick_params(which='major', size=4, width=1, direction='in', top='on')
ax.xaxis.set_tick_params(which='minor', size=4, width=1, direction='in', top='on')
ax.yaxis.set_tick_params(which='major', size=4, width=1, direction='in', right='on')
ax.yaxis.set_tick_params(which='minor', size=4, width=1, direction='in', right='on')

# Edit the major and minor tick locations
numberOfTicsPerAxis = 5

minTics_x = 2 #small tics per big tic
x_tic = 2#abs(I.max() - I.min()) / numberOfTicsPerAxis
ax.xaxis.set_major_locator(mpl.ticker.MultipleLocator(x_tic))
ax.xaxis.set_minor_locator(mpl.ticker.MultipleLocator(x_tic/minTics_x))
minPerMajTic_y = 2
y_tic = 2.5 #abs(P.max() - P.min()) / numberOfTicsPerAxis
ax.yaxis.set_major_locator(mpl.ticker.MultipleLocator(y_tic))
ax.yaxis.set_minor_locator(mpl.ticker.MultipleLocator(y_tic/minPerMajTic_y))

# Set the axis limit
min_y = np.min(P)
max_y = np.max(P)
min_x = np.min(I)
```

```
max_x = np.max(I)
axis_buffer = 0.20 #making the domain & range a % wider than the dataset
x_buffer = (max_x - min_x) * axis_buffer
y_buffer = (max_y - min_y) * axis_buffer
ax.set_xlim(min_x - x_buffer, max_x + x_buffer)
ax.set_ylim(min_y - y_buffer, max_y + y_buffer)

# Add the x and y-axis labels
ax.set_xlabel(r'$I[mA]$', labelpad=5)
ax.set_ylabel('Pressure Drop [psi]', labelpad=5)
# ax.set_title('Pressure / Current Relation')

# Collect all the font names available to matplotlib
font_names = [f.name for f in fm.fontManager.ttflist]
# print(font_names)
# Edit the font, font size, and axes width
mpl.rcParams['font.family'] = 'Avenir'
plt.rcParams['font.size'] = 10
plt.rcParams['axes.linewidth'] = .9


# Save figure
plt.savefig('IMG/transducer_psi_mA.png', dpi=900, transparent=False,
bbox_inches='tight')

#Show the Plot
plt.show()
print("PROGRAM COMPLETE")
```