```python
#Wesley Johanson
from re import I
import ChE
import numpy as np

#USER PARAMETERS
_filename = "CSV/plot_2.csv"
files = [ ]
#Data labels
labels = np.loadtxt(_filename, unpack=True, delimiter=',',dtype=str)[:, 0]
savePlotAs = "Newplot.png"
folder = "IMG/"
i = 2

# regressionVars = [0] #Variables to calculate Linear Regression R^2 values with respect to
customColors = None
customYLabel = None #"poop"
# First int is x-axis, next int's are the functions to plot w/ respect to x on
# the same plot

plots = None

plots = [
            [0, 1 ]
            ]

YLabels = None

YLabels = [ "$Vz/Vz_{Max}$",
            "friction factor",
            "$V^{+}$",
            "poop3"
            ]

_markers = [ ".",".", "+", "2", "", ""]

for plotData in plots:
    # print(plotData)
    #  Create Plotting Object: LOAD A LABELED CSV FILE
    plot = ChE.ChEplot()
    plot.loadCSV_str(_filename, labels, indepVars=1, skip=1)
    plot.printData()
    #Set Data
    plot.setDataLabel(labels)
    #Plotting
    plot.setDataColors(customColors)
    plot.setFxns2Plot(plotData)
    plot.plotData_str(width=6,height=6, markers=_markers)
    #Statistics & Regression
    # plot.plotLRegLines(width=0.5)
    # plot.printAllRSquared()
    #Plot Parameters
    xaxisLabel = labels[plotData[0]] #Don't Change
    print(xaxisLabel)
    # xaxisLabel = "$S^{+}$"
    yaxisLabel = YLabels[i] if YLabels else labels[plotData[1]]
```

```
    plot.setAxisLabels(xaxisLabel, yaxisLabel, xpadding=5, ypadding=5)
    plot.setTicProps()
    # plot.setNumTics(delta_x=10, delta_y=10, x_subTics=3, y_subTics=3)
    plot.showLegend()
    # plot.changeFont()
    #Presentation
    # plot.showPlot()
    temp = folder + str(i) + '_' + savePlotAs
    #TESTING!!!!!!!!!!!!!!!!!!!!!!!!1
    # plot.figure.xscale('log')
    plot.savePlot(filename=temp,_dpi=600)
    print(temp)
    plot.close()
    i += 1

# def   saveManyPlots(_dataSets: list):
#   for i in range(0, len(_dataSets)):
#       #Create Plotting Object: LOAD A LABELED CSV FILE
#       plot = ChE.ChEplot()
#       plot.loadCSV(_filename, labels, indepVars=1, skip=1)
#       #Set Data
#       plot.setDataLabel(labels)
#       #Plotting
#       plot.setDataColors(customColors)
#       plot.setFxns2Plot(_dataSets)
#       plot.plotData(width=6,height=6)
#       #Statistics & Regression
#       plot.plotLRegLines(width=0.5)
#       plot.printAllRSquared(vars=regressionVars)
#       #Plot Parameters
#       xaxisLabel = labels[_dataSets[0]] #Don't Change
#       yaxisLabel = customYAxisLabel if customYAxisLabel else labels[1]
#       plot.setAxisLabels(xaxisLabel, yaxisLabel, xpadding=5, ypadding=5)
#       plot.setTicProps()
#       # plot.setNumTics(delta_x=1.0, delta_y=1.0, x_subTics=3, y_subTics=3)
#       plot.showLegend()
#       plot.changeFont()
#       #Presentation
#       # plot.showPlot()
#       imgFileName = folder + str(i) + "_" + savePlotAs
#       plot.savePlot(filename=imgFileName,_dpi=600)


# saveManyPlots(dataSets2Plot)



# #Data
# plot.loadCSV('logRe_logf.csv', dataNames, indepVars=1)
# #Plotting
# # plot.setFnLabels(fnLabels)
# plot.setDataColors(['#89CFF0','#800020','#301934'])
# plot.plotData(width=6,height=6)
# #Regression
# plot.plotLRegLines(width=0.1)
# plot.printAllRSquared()
# #Plot Parameters
```

```
114 # plot.setAxisLabels("$Log_{10}(\mathcal{Re})$", "$Log_{10}(\mathcal{f})$", xpadding=5,
    ypadding=5)
115 # plot.setTicProps()
116 # plot.setNumTics(0.1, 0.25, 3,3)
117 # plot.showLegend()
118 # plot.changeFont()
119 # #Presentation
120 # plot.showPlot()
121 # plot.savePlot(filename="log(Re)_vs_log(f).png",_dpi=600)
122 # # plot.close()
123
124
125
126
127
128 #Best fit calibration line for the data, to modify the process control pressure.
129 #I need to determine a logical argument if the transducer should be implemented
130
131 #Data->
132 #   Err = +/- (ErrX, ErrY)->
133 #       Calibration line->
134 #           Calculate ΔP for the tech to modify the process control
135               #Should the transducer be used in the plant, given data
136
137 #Claibration Curve with 2 different method
138 #       (a)std regression
139 #            report errors/confidence intervals on the estimated parameters
140 #            (i.e., slope and intercept).
141 #       (b) Using χ2 minimization, also
142 #           reporting the slope and intercept.
143
144
145 print("Program Complete")
```