```
'''
A new py code to read strings (each line is a strng)
and seperate each line to two parts, and then convert
the second part to a float point numerical number.
8/25/2020. 9/4/2020, upgraded pip, downloaded numpy, and matplotlib,
and finished the whole program.9/5/2020, added the r-sqare value
The "for ... in" is really powerful and there is no need to
do indexing iteration (looping) anymore!
a [ ] list > integer; b [ ] list float.
'''

#ftype = input("Please choose the files type. For comma seperated file, press 1, for space seperated,
press 2.  ")
#xtype = input("Please choose the x data type. For numerical data, press 1, for stringd, press 2. ")
f = open("C:\LSFT\TXT\weight-3.txt", "r")
a = []                          # Original list a,  all strings
b = []                          # Original list b,  all strings
list_floats_a = []                  # Converted list, all float
list_floats_b = []                  # Converted list, all float

for line in f:
    #print("ftype = ",  ftype)
                                                            #line = f.readline()
    #print(line)
    #if ftype == "1":
    sp = line.rsplit(",")           # "," is very important. not ", " (no space after ",") for inv3.txt
    #if ftype == "2":
        #sp = line.rsplit(" ")

    #print(sp)                  # (and for inv11.csv), comma seperated txt files
    #x = sp[0]                  # the string is separated into two parts, before ","  and after it
    #y = sp[1]                  # the first part is sp[0], the second part is sp[1]
    #print (y)
    a.append(sp[0])             # creating list a, using .append method
    b.append(sp[1])             # creating list b, using .append method
#print(a, "\n")
#print(b, "\n")

for item in a:                  # operation in list a
    list_floats_a.append(int(item))         # convert strings in a to int (or, float) using int() or float()
function
                                #item is the value corresponding the counter in the list.
```

```
for item in b:                          # operation in list b;
    list_floats_b.append(float(item))   # convert strings in b to float, using float() function
#print(list_floats_a)                   # result is in a new list - list_floats_a
#print(list_floats_b)                   # result is in a new list - list_floats_b
#print("item = ", item)

import numpy
import matplotlib.pyplot as plt         # "as plt", so it can be called as "plt" later
from sklearn.metrics import r2_score

x = list_floats_a                       # x must be float number list_floats_a, a is only strings
y = list_floats_b                       # y must be float number list_floats_b, b is only strings

mymodel = numpy.poly1d(numpy.polyfit(x, y, 3))  # NumPy has a method that lets us make a polynomial
mode

print("\n The fitting r-squared is ")
print("\n  ", r2_score(y, mymodel(x)),"\n")      # print the r-sqared

myline = numpy.linspace(1, 420, 100)        # x curve (not x-axis) range: 1 - 420;

plt.scatter(x, y)                       # draw the original scatter plot
plt.plot(myline, mymodel(myline))       # draw the line of polynomial regression:
plt.show()                              # display the diagram

f.close()

print(list_floats_a, "\n")                  # contents in lists and variables can be carried out to
beyong f.close()
print(list_floats_b, "\n")

f2 = open("C:\\LSFT\DAT\weight-new.csv", "a")   # open file 2 to append ("a"), can not use write ("W")
f2.write(str(list_floats_a))                # write() argument must be str, not list
f2.write("\n \n")
f2.write(str(list_floats_b))
f2.write("\n \n")
f2.close()

'''
note: line 74 "f2 = open("C:\\LSFT\DAT\weight-new.csv", "a")", must use "a" (append), not "w" (write).
otherwise onle the lasr set of data would be left over, while all the rest of data would be all deleted.
```

R-Squared
It is important to know how well the relationship between the values of the x- and y-axis is, if there are no relationship the polynomial regression can not be used to predict anything.

The relationship between the values of the x- and y-axis is measured with a value called the r-squared. The r-squared value ranges from 0 to 1, where 0 means no relationship, and 1 means 100% related.

'''