```
# practiging fitting plots with various functionsimport numpy as np
import numpy as np
import LT.box as B

file_name = 'examples_data.py' #set name of data file
f = B.get_file(file_name) #open and read the file
# get the data
# current

#extract the data from the file and save them into their respective variables
A = B.get_data(f,'A')
b = B.get_data(f,'b')
db = B.get_data(f, 'db')
C = B.get_data(f,'C')
D = B.get_data(f, 'D')
# The following examples fitl to fit fits C vs A, using linear fit,
#polynomial fits, in either the whole ranges (fit1, fit2) or a subrange (fit 3, 4
B.plot_exp(A, C, db) #plot the data with error bars (db)
B.pl.show() #display the plot

#You should uncomment the next line to see what it does to the x-axi
#Can you set the y-axis title now?
B.pl.xlabel("x (unit)") # sets the x-axis label
B.pl.ylabel("y (unit)")

fit1 = B.linefit(A, C, db) #does the linear fit of the entire dataset
B.plot_line(fit1.xpl, fit1.ypl) #plots the fitted line

#the following two lines selecting the ranges
r1 = B.in_between(4.0, 18.0, C) #. in window should be changed to in _between
r2 = B.in_between(2.0, 12.0, C) #creates a variable with the data points of C
#between 1 and 12

#only fit the selected ranges as specified by r1 or r2
fit3 = B.linefit(A[r1], C[r1], db[r1]) #linear fit using data from range 1
B.plot_line(fit3.xpl, fit3.ypl) #plot it

#the fit below use second order-- defined by the "2" in the argument
#polynomial; you should change 2 to other integers and see how the
#fits are different
fit4 = B.polyfit(A[r2], C[r2], db[r2], 2)
B.plot_line(fit4.xpl, fit4.ypl)

#for polynomial 5 (example) within range r2
fit5 = B.polyfit(A[r2], C[r2], db[r2], 5)
B.plot_line(fit5.xpl, fit5.ypl)


speed = fit4.parameters[1].value #for speed = fit4.par[1]
#coefficient value (slope)
```

```
d_speed = fit4.parameters[1].err #or d_speed = fit4.sig_par[1]
#uncertainty value in the coefficient

#printing out the fitting parameters with proper significant digits
print("\nspeed is %3E +/- %3E m/s \n" % (speed, d_speed))
# Print results in scientific notation
```