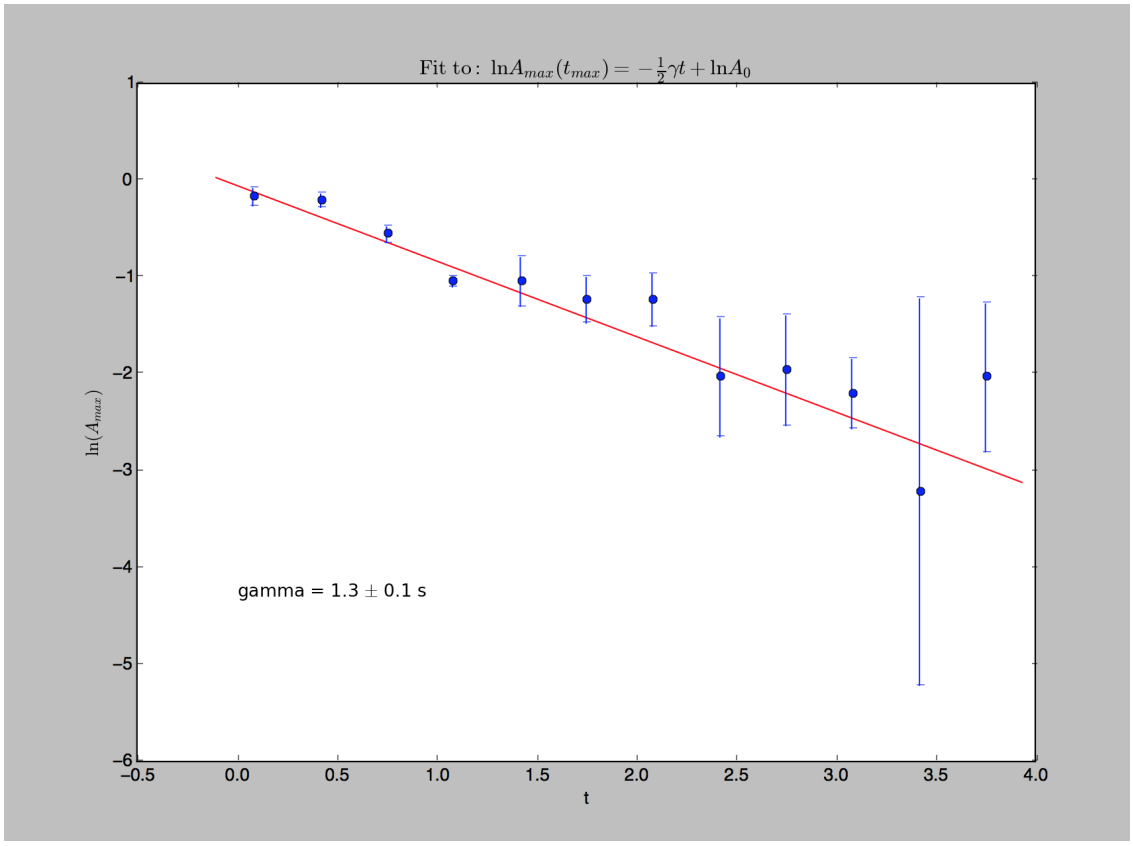
```
import numpy as np;
import matplotlib.pyplot as plot
import math;
from scipy.optimize import curve_fit

# load csv
data = "Amaxdata.csv";
t, Amax, dAmax = np.loadtxt(data, skiprows=0, unpack = True, delimiter=',');

# this function comes straight from Intro to Python for Science
def LineFitWt(x, y, sig):
    sig2 = sig**2
    norm = (1./sig2).sum()
    xhat = (x/sig2).sum() / norm
    yhat = (y/sig2).sum() / norm
    slope = ((x-xhat)*y/sig2).sum()/((x-xhat)*x/sig2).sum()
    yint = yhat - slope*xhat
    sig2_slope = 1./((x-xhat)*x/sig2).sum()
    sig2_yint = sig2_slope * (x*x/sig2).sum() / norm
    return slope, yint, np.sqrt(sig2_slope), np.sqrt(sig2_yint)

X = t;              # t does not change
Y = np.log(Amax);   # scaling to log(yint)
dY = dAmax/Amax;    # uncertainty in log(yint)

slope, yint, d_slope, d_yint = LineFitWt(X, Y, dY)

#fitting parameters
A_0 = np.exp(yint)
gamma = -1.0/slope

#uncertainties
dA_0 = A_0 * d_yint
Dgamma = gamma**2 * d_slope

# two points deterine a line
Xext = 0.05*(X.max()-X.min())
Xfit = np.array([X.min()-Xext, X.max()+Xext])
Yfit = yint + slope*Xfit

plot.errorbar(X, Y, dY, fmt="bo")       #plot points with new error
plot.plot(Xfit, Yfit, "r-", zorder=-1)  #plot fit

plot.title("$\mathrm{Fit\\ to:}\\ \ln A_{max}(t_{max}) = -\\frac{1}{2}\gamma t +\ln A_0$")
plot.xlabel("t")
plot.ylabel("$\ln(A_{max})$")

plot.text(0, .5-4.8, "gamma = {0:0.1f} $\pm$ {1:0.1f} s".format(gamma, Dgamma))

plot.show();
```


Fit to: $\ln A_{max}(t_{max}) = -\frac{1}{2}\gamma t + \ln A_0$
gamma = 1.3 ± 0.1 s
$\ln(A_{max})$
t
1
0
−1
−2
−3
−4
−5
−6
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0