Nonlinear Fit

Kinzel	Python
`f[t_] := a Sin[om t + phi] Exp[-b t]`	`def f(t, a, om, phi, b):` `    return a*np.sin(om*t + phi)*np.exp(-b*t)`
a `= {1,1,0,0.1}`	`y = f(t,1,1,0,0.1)`

`data = Table[{t, Sin[t]Exp[-t/10.] + 0.4*Random[] - 0.2}//N, {t,0,3Pi,0.3Pi}]`	`t = np.arange(0, 3*np.pi+0.1, 0.3*np.pi)`
	
	Note that `arange` returns points in the half-open interval $[a, b)$.
	
	`noise = 0.4*random.random() - 0.2`
	`data = y + noise`

`NonlinearRegress[data, f[t], t, {{a,1.1}, {om,1.1}, {phi,.1}, {b,.2}}, ShowProgress -> True]`	`popt,pcov = scipy.optimize.curve_fit(f, t, data)` `print(popt)`
`Quantile[ChiSquareDistribution[7], 0.95]`	`scipy.stats.chi2.ppf(0.95, df=7))`

`limit[x_] = Quantile[ChiSquareDistribution[7], x]`	`def limit(x):`
	`    return scipy.stats.chi2.ppf(x, df=7)`
	
	To get the distribution, use `scipy.stats.chi2.pdf()`

$\chi_0^2 = \sum_{i=1}^{N} \left[\frac{Y_i - g(\boldsymbol{a_0}, t_i)}{\sigma_i}\right]^2$ where $\boldsymbol{a_0} = \text{popt}$	`def chi0squared(y,f):` `    return sum((data - f(t,*popt))/(2/150))**2`

$\chi^2(\boldsymbol{a}) = \sum_{i=1}^{N} \left[\frac{Y_i - g(\boldsymbol{a}, t_i)}{\sigma_i}\right]^2$	For 1st contour plot (ab-space):
	
	`def chisquared(y,f):`
	`    return sum((data - f(t, A, popt[1], popt[2], B))/(2/150))**2`
	
	where `A` & `B` are the corresponding meshgrids. Note that `popt` is a vector of the best fit parameters in the same order as defined by the fitting function `f`

$P_M(\chi^2)$	`def P(chisquared,M):`
	`    return sps.chi2.cdf(chisquared, df=M)`
	
	Or, alternatively:
	
	`def P(chisquared,M):`
	`    return scipy.special.gammainc(chisquared/2, M/2)`