CP 2

Advanced Model Fitting and Plotting

Nathan Ngqwebo

NGQLIN011

Department of Science, University of Cape Town

PHY2004W | Intermediate Physics

Dr Spencer Wheaton

6 March 2023

CP 2

Advanced Model Fitting and Plotting

This computational activity is an exercise in fitting linear and non-linear models to data and presenting information about the quality of the computed fit solutions with data plotting and uncertainty analyses. The following are the results of two regression tasks fitting linear models to data with and without uncertainty in the dependent variable. All code was produced using Python (Van Rossum & Drake, 2009) with the packages: SciPy (Seabold & Perktold, 2010), NumPy(Harris et al., 2020), and Matplotlib(Hunter, 2007) used for analysis and plotting.

Unweighted Linear Regression

Since we are performing an unweighted regression, the weights associated with each deviation in y-value should be equal to one. Thus, in the SciPy module: curve_fit, the uncertainties in the y-values: "sigma" should be None, as in Figure 1.

```
popt, pcov = curve_fit(f, data[:, 0], data[:, 1],
p0, sigma=None)
```

Figure 1

Python Code Sample of the curve_fit Function Call with "sigma" = None

Performing the unweighted linear regression on the data found in *"LinearNoErrors.txt"* yielded the results shown in Table 1 and the plot in Figure 2.

Table 1

Tabulated Results of Unweighted Linear Regression on "LinearNoErrors.txt" Data

Parameter	Value	Uncertainty
m	0.590	0.026
c	1.07	0.19

Note: These values agree with the values obtained in CP1: $m = 0.58 \pm 0.03, c = 1 \pm 0.2$

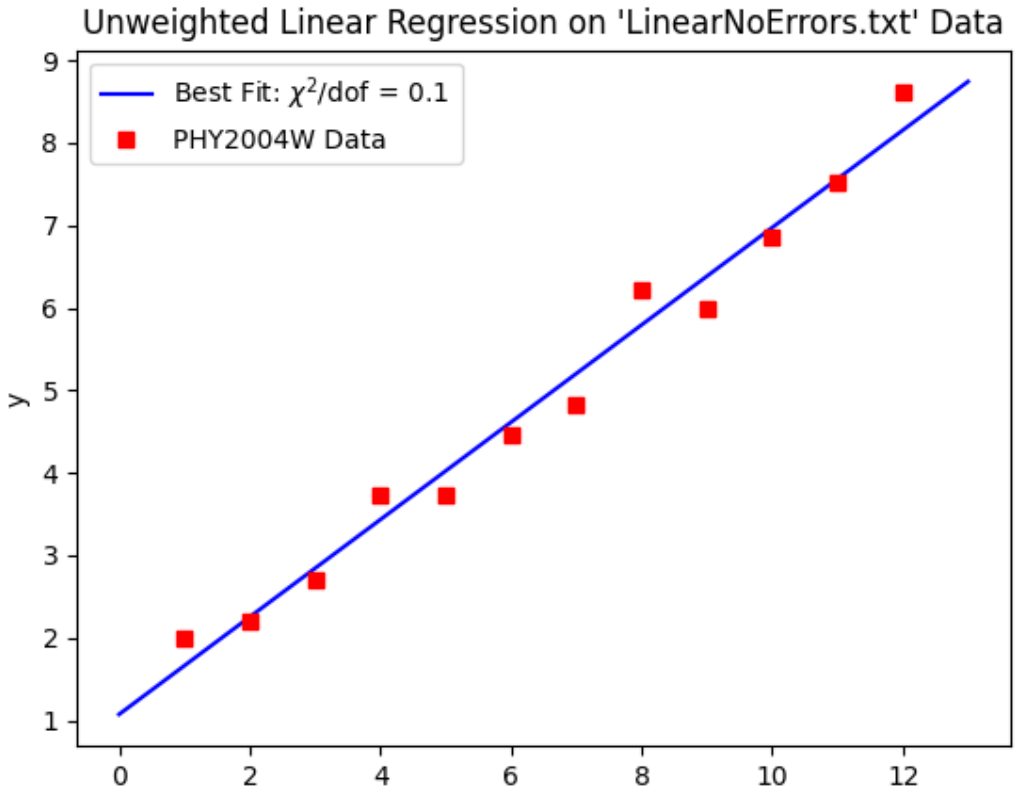

Figure 2

Scatter Plot of "LinearNoErrors.txt" Data with Line of Best Fit

Fit Evaluation

Using the χ^2/dof goodness of fit estimate, we found that the fit scored 0.1, a significant deviation from 1. Hence, we quoted the uncertainty results in Table 1 as the square root of the covariance matrix diagonals, as specified in the SciPy documentation, with an added correction factor of $\sqrt{\chi^2/dof}$ (Press et al., 2007).

Given that lower χ^2/dof estimates suggest that the model is trying to fit noise, and that we saw a significant anticorrelation[1] between m and c, given by the "curve_fit" method we do not find these uncertainty estimates to be reliable.

[1] The covariance matrix returned by the "curve_fit" method showed an anticorrelation between m and c of -0.88

Figure 2 is still identical to the one obtained in CP1. This suggests that there is agreement between the SciPy method, and the method described in the theory.

Fit Visualization

Figure 3 is a contour plot which describes the χ^2/dof goodness of fit estimate on the $m \times c$ parameter space where the lowest values indicate a better combination of the parameters m and c.

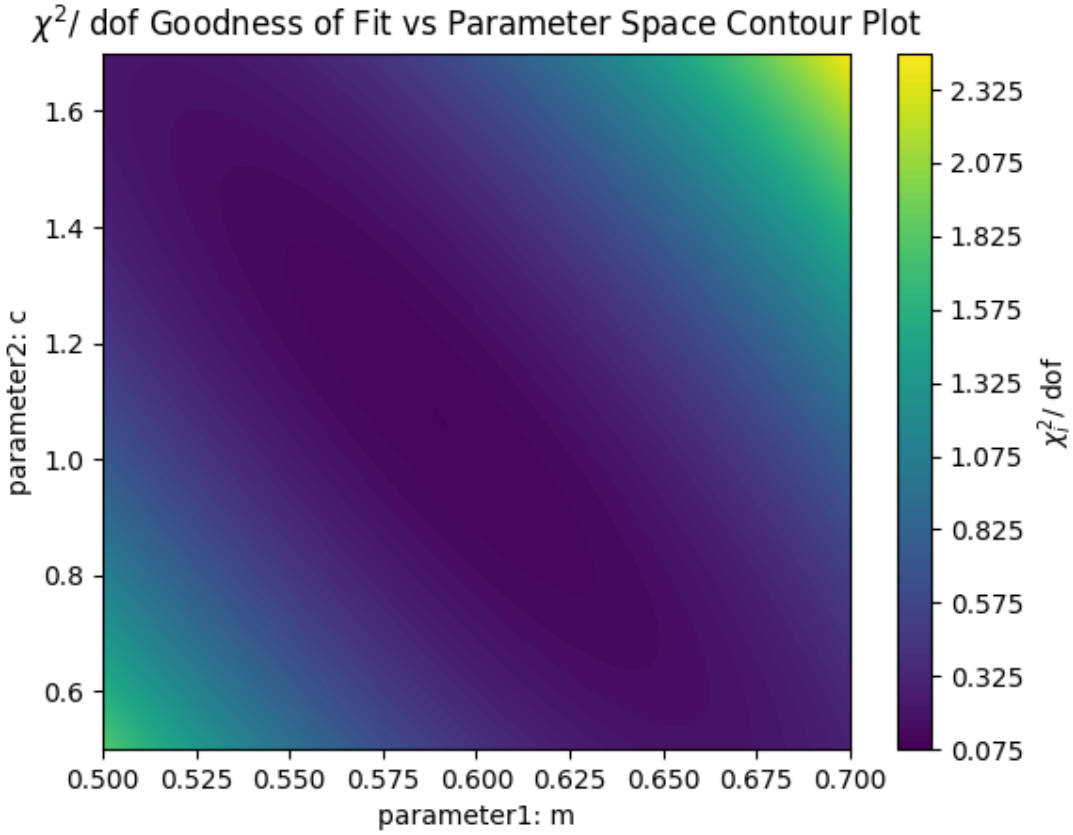

Figure 3

Contour Plot with Height Showing χ^2/dof as Height in the $m \times c$ Parameter Space

Weighted Linear Regression

The same procedure as above was followed to perform a weighted linear regression on the *"LinearWithErrors.txt"* data. The calculated parameters and their standard uncertainty are presented in Table 2 and the plot in Figure 4**.**

Table 2

Tabulated Results of Weighted Linear Regression on "LinearWithErrors.txt" Data

Parameter	Value	Uncertainty
m	0.606	0.017
c	0.86	0.11

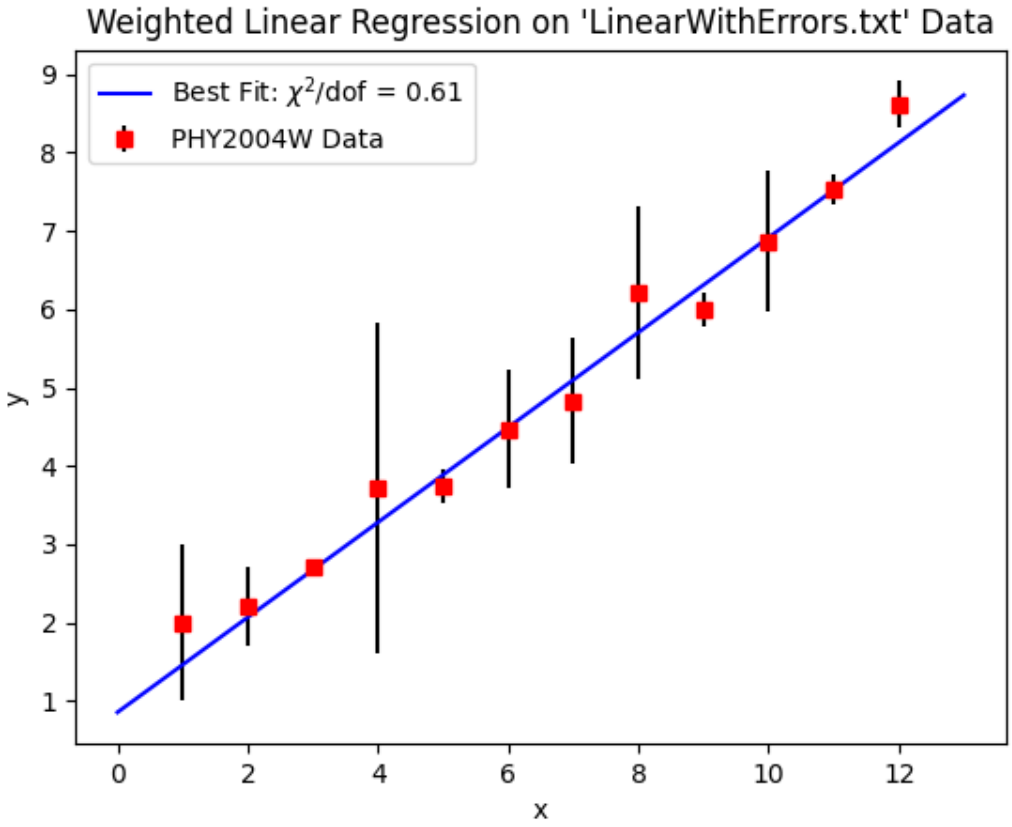

Figure 4

Scatter Plot of "LinearWithErrors.txt" Data with Best Fit Line

Note: The non-uniform weight distribution of this data compared to the unweighted data resulted in an increase in slope to conform more to data points with lower uncertainties.

Fit Evaluation

As with the unweighted regression, we found that the χ^2/dof goodness of fit estimate value of 0.61 was a significant enough deviation from 1 to warrant using the correction factor to quote the uncertainties in the fitted parameters. This fit also showed a significant anticorrelation between m and c of -0.85. Thus, the assumption that one of these parameters can be fixed while the other is varied to find the optimal value does not hold well and the uncertainties inferred from this process should not be trusted.

Fit Visualisation

Figure 5 presents the χ^2/dof contour plot for the weighted linear regression.

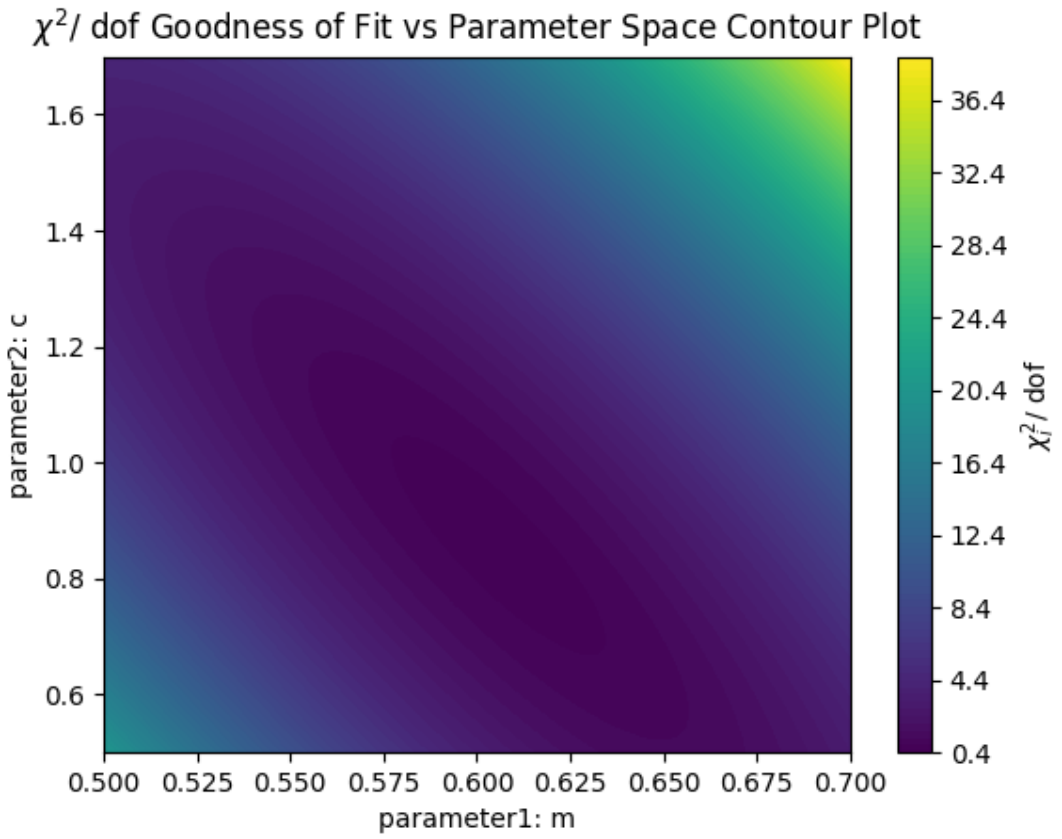

Figure 5

Contour Plot with Height Showing χ^2/dof as Height in the $m \times c$ Parameter Space

Note: There is a much steeper gradient in the goodness of fit estimate for the same parameter range as the unweighted case.

References

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science \& Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (3rd ed.). Cambridge University Press.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and Statistical Modeling with Python. *Proceedings of the 9th Python in Science Conference*, 92–96. https://doi.org/10.25080/majora-92bf1922-011

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.