WLSF

SGTB Khalsa College, University of Delhi

Preetpal Singh(2020PHY1140)(20068567043)

Ankur Kumar(2020PHY1113)(20068567010)

Unique Paper Code: 32221401

Paper Title: Mathematical Physics III

Submitted on: March 14, 2022

B.Sc(H) Physics Sem IV

Submitted to: Dr. Mamta

Theory

- What is the principle of maximum likelihood?
- Describe the method of weighted least squares for fitting a linear function to a given set of N data points $(x_i, [y_{ij}])\,(i = 1, \cdots, N$ and $j = 1, \cdots, N1)$, where $N1$ is the number of y-values observed for each x. Take weights to be $w_i = 1/\sigma_i^2, \sigma_i$ being the value of error in $\bar{y}_i$ and $\bar{y}_i$ is the mean of all $[y_{ij}]$ for each i.
  1. How is Least square fitting related to the principle of maximum likelihood?
  2. Derive the formulae for mean, error in mean, slope, intercept, errors in slope and intercept.
  3. Under what condition does weighted least square fitting (WLSF) reduces to ordinary least square fitting(OLSF)? Show that under this condition above formulae reduce to the formulae for OLSF.
- Define correlation coefficient and adjusted correlation coefficient.

Principle of maximum likelihood

The principle of maximum likelihood is a method of obtaining the optimum values of the parameters that define a model. The parameter values are found such that they maximise the likelihood that the process described by the model produced the data that were actually observed.

To understand this further, we have taken an example where the data points can be adequately described by a gaussian(normal) distribution. Recall that the Gaussian distribution has 2 parameters. The mean, μ, and the standard deviation, σ. Different values of these parameters result in different curves. Maximum likelihood estimation is a method that will find the values of μ and σ that result in the curve that best fits the data.

The values that we find are called the maximum likelihood estimates (MLE). What we want to calculate is the total probability of observing all of the data.

The first assumption is that each data point is generated independently of the others. The probability density of observing a single data point x, that is generated from a Gaussian distribution is given by:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

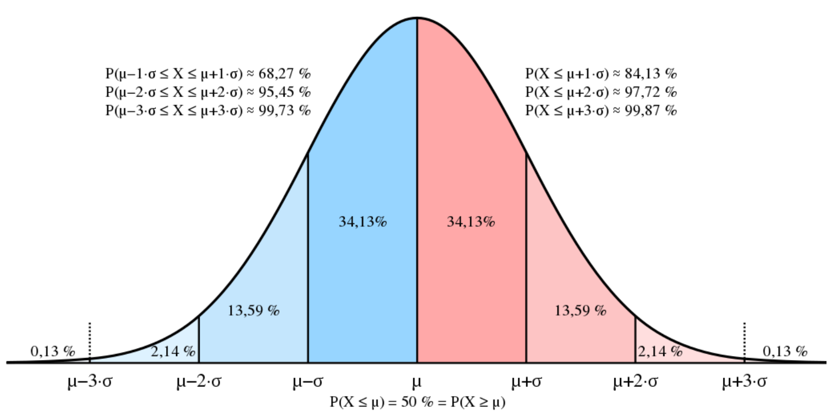

Figure 1: Gaussian Distribution

The total(joint) probability density of observing the data points is given by:

$$P(x) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x_i-\mu)^2/2\sigma^2}$$

To find the MLE values for our parameters we will differentiate the function with respect to the parameter and equate it to 0. We simplify the equation by taking the natural log on both sides which gives:

$$ln(P(x)) = ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2}$$

We differentiate this function with respect to the parameters one at a time and equate it to zero to find the values of the MLE.

$$\frac{\partial ln(P(x))}{\partial \mu} = 0$$

$$\frac{\partial ln(P(x))}{\partial \sigma} = 0$$

Weighted Least Squares

Least squares minimisation is another common method for estimating parameter values for a model. It turns out that when the model is assumed to be Gaussian as in the example above, the MLE estimates are equivalent to the least squares method. For least squares parameter estimation we want to find the line that minimises the total squared distance between the data points and the regression line. In maximum likelihood estimation we want to maximise the total probability of the data. When a Gaussian distribution is assumed, the maximum probability is found when the data points get closer to the mean value. Since the Gaussian distribution is symmetric, this is equivalent to minimising the distance between the data points and the mean value.

We have N data points $(x_i, [y_{ij}])(i = 1, ...N)$ and $(j = 1, ...N_1)$ where N_1 is the number of y-values observed for each x. Take weights to be $w_i = 1/\sigma_i^2, \sigma_i$ being the value of error in $\bar{y}_i$ and $\bar{y}_i$ is the mean of all $[y_{ij}]$ for each i.

Mean

The mean of y values for a given x_i is written as:

$$y_i = \frac{\sum_{j=1}^{N_1} y_{ij}}{N_1}$$

Error in Mean

The standard deviation in y_i is given by:

$$s_i = \sqrt{\frac{1}{N_1 - 1}\sum_{j=1}^{N_1}(y_{ij} - y_i)^2}$$

The standard error will be given by:

$$\sigma_i = \frac{s_i}{\sqrt{N_1}}$$

Parameters

The usual linear regression model assumes that all the random error components are identically and independently distributed with constant variance. When this assumption is violated, then ordinary least squares estimator of regression coefficient looses its property of minimum variance in the class of linear and unbiased estimators. The violation of such assumption can arise in anyone of the following situations:

- The variance of random error components is not constant.
- The random error components are not independent.
- The random error components do not have constant variance as well as they are not independent.

In such cases, the covariance matrix of random error components does not remain in the form of an identity matrix but can be considered as any positive definite matrix. Under such assumption, the OLSE does not remain efficient as in the case of identity covariance matrix. The generalized or weighted least squares method is used in such situations to estimate the parameters of the model. In this method, the deviation between the observed and expected values of y_i is multiplied by a weight w_i where w_i is chosen to be inversely proportional to the variance of y_i.

For simple linear regression model, the weighted least squares function is:

$$\chi^2 = \sum_{i=1}^{N} w_i (y_i - c - mx_i)^2$$

In this, minimise χ^2

$$\chi^2 = \sum \omega (y_i - mx_i - c)^2 = \sum \omega_i \left(y_i^2 + m^2 x_i^2 + c^2 + 2mx_i c - 2mx_i y_i - 2cy_i\right)$$

$$\chi^2 = \sum \omega_i y_i^2 + \sum \omega_i m^2 x_i^2 + \sum \omega_i c^2 + 2\sum \omega_i m x_i c - 2\sum \omega_i m x_i y_i - 2\sum \omega_i y_i c$$

$$\frac{\partial \chi^2}{\partial m} = 2m\sum \omega_i x_i^2 + 2c\sum \omega_i x_i - 2\sum \omega_i x_i y_i$$

$$\text{for minimum}\chi^2 : \frac{\partial \chi^2}{\partial m} = 0$$

$$m\sum \omega_i + c\sum \omega_i x_i = \sum \omega_i y_i \tag{1}$$

$$\text{for minimum}\chi^2 : \frac{\partial \chi^2}{\partial c} = 0$$

$$\frac{\partial \chi^2}{\partial c} = 2c\sum \omega_i + 2m\sum \omega_i x_i - 2\sum \omega_i y_i$$

$$c\sum \omega_i + m\sum \omega_i x_i = \sum \omega_i y_i \tag{2}$$

On multiplying equation(1) by $\sum \omega_i$ and equation (2) by $\sum \omega_i x_i$ and then substracting

$$m(\sum \omega_i)\left(\sum \omega_i x_i^2\right) - m\left(\sum \omega_i x_i\right)^2 = \left(\sum \omega_i \left(x_i y_i\right)\left(\sum \omega_i\right) - \left(\sum \omega_i x_i\right)\left(\sum \omega_i y_i\right)\right.$$

Slope

$$m = \frac{\sum \left(\omega_i x_i y_i\right)\left(\sum \omega_i\right) - \left(\sum \omega_i x_i\right)\left(\sum \omega_i y_i\right)}{\left(\sum \omega_i\right)\left(\sum \omega_i x_i^2\right) - \left(\sum \omega_i x_i\right)^2}$$

On multiplying equation(1) by $\sum \omega_i x_i$ and equation (2) by $\sum \omega_i x_i^2$ and then substracting

$$c\left(\sum \omega_i x_i\right)^2 - c(\sum \omega_i)\left(\sum \omega_i x_i^2\right) = \left(\sum \omega_i x_i\right)\left(\sum \omega_i x_i y_i\right) - \sum\left(\omega_i x_i^2\right)\sum\left(\omega_i y_i\right)$$

Intercept

$$c = \frac{\sum\left(\omega_i x_i^2\right)\sum\left(\omega_i y_i\right) - \left(\sum \omega_i x_i\right)\left(\sum \omega_i x_i y_i\right)}{\left(\sum \omega_i\right)\left(\sum \omega_i x_i^2\right) - \left(\sum \omega_i x_i\right)^2}$$

Error in Slope

Let

$$\left(\sum \omega_i\right)\left(\sum \omega_i x_i^2\right) - \left(\sum \omega_i x_i\right)^2 = \Delta$$

By propogation of Errors,

$$\sigma_m^2 = \sum\left(\frac{\partial m}{\partial y_i}\sigma_i\right)^2$$

$$\frac{\partial m}{\partial y_i} = \frac{\left(\omega_i x_i\right)\left(\sum \omega_i\right) - \left(\sum \omega_i x_i\right)\left(\omega_i\right)}{\Delta}$$

$$\left(\frac{\partial m}{\partial y_i}\right)\sigma_i = \frac{\sum \omega_i \frac{x_i}{\sigma_i} - \frac{\sum \omega_i x_i}{\sigma_i}}{\Delta}$$

$$\begin{aligned}
\sigma_m^2 &= \sum \frac{\left[\sum \omega_i \frac{x_i}{\sigma_i} - \frac{\sum \omega_i x_i}{\sigma_i}\right]^2}{\Delta^2} \\
\sigma_m^2 &= \sum \frac{w_i[(\sum \omega_i)^2 x_i^2 - (\sum \omega_i x_i)^2 - 2(\sum \omega_i \sum \omega_i x_i)x_i]}{\Delta^2} \\
\sigma_m^2 &= \sum \frac{w_i[(\sum \omega_i)^2 x_i^2 - (\sum \omega_i \bar{X})^2 - 2(\sum \omega_i)^2 \bar{(}X)x_i]}{\Delta^2} \\
\sigma_m^2 &= \sum \frac{w_i[(\sum \omega_i)^2 (x_i^2 + \bar{X}^2 - 2\bar{X}x_i)]}{\Delta^2} \\
\sigma_m^2 &= \sum \frac{w_i[(\sum \omega_i)^2 (x_i - \bar{X})^2]}{\Delta^2} \\
\sigma_m^2 &= \frac{(\sum \omega_i)^2 \sum \omega_i (x_i - \bar{X})^2}{\Delta^2} \\
\sigma_m^2 &= \frac{\sum \omega_i}{\Delta} \\
\sigma_m &= \sqrt{\frac{\sum w_i}{\sum w_i \sum w_i x_i^2 - \left(\sum w_i x_i\right)^2}}
\end{aligned}$$

Error in Intercept

Let

$$\left(\sum \omega_i\right)\left(\sum \omega_i x_i^2\right) - \left(\sum \omega_i x_i\right)^2 = \Delta$$

In similar fashion,

$$\sigma_c^2 = \sum\left(\frac{\partial c}{\partial y_i}\sigma_i\right)^2$$

$$\frac{\partial c}{\partial y_i} = \frac{(\omega_i)\left(\sum \omega_i x_i^2\right) - \left(\sum \omega_i x_i\right)(\omega_i x_i)}{\Delta}$$

$$\left(\frac{\partial c}{\partial y_i}\right)\sigma_i = \frac{\sum \omega_i \frac{x_i}{\sigma_i} - \frac{\sum \omega_i x_i}{\sigma_i}}{\Delta}$$

$$\sigma_c^2 = \sum \frac{\left[\frac{\sum \omega_i x_i^2}{\sigma_i} - \frac{\sum (\omega_i x_i) x_i}{\sigma_i}\right]^2}{\Delta^2}$$

$$\sigma_c^2 = \sum \frac{\omega_i[(\sum \omega_i x_i^2)^2 + (\sum \omega_i x_i)^2 x_i^2 - 2(\sum \omega_i x_i^2)(\sum \omega_i x_i) x_i]}{\Delta^2}$$

$$\sigma_c^2 = \sum \frac{\omega_i[(\sum \omega_i x_i^2)^2 - (\sum \omega_i x_i^2)(\sum \omega_i x_i) x_i + (\sum \omega_i x_i)^2 x_i^2 - (\sum \omega_i x_i^2)(\sum \omega_i x_i) x_i]}{\Delta^2}$$

$$\sigma_c^2 = \sum \frac{w_i[\sum \omega_i x_i^2(\sum \omega_i x_i^2 - (\sum \omega_i x_i) x_i) + \sum \omega_i x_i((\sum \omega_i x_i) x_i^2 - (\sum \omega_i x_i^2) x_i)]}{\Delta^2}$$

$$\sigma_c^2 = \sum \frac{[\sum \omega_i x_i^2(\sum \omega_i \sum \omega_i x_i^2 - (\sum \omega_i x_i)^2) + \sum \omega_i x_i((\sum \omega_i x_i)\sum \omega_i x_i^2 - (\sum \omega_i x_i^2)\omega_i x_i)]}{\Delta^2}$$

$$\sigma_c^2 = \frac{\sum \omega_i x_i^2}{\Delta}$$

$$\sigma_c = \sqrt{\frac{\sum w_i x_i^2}{\sum w_i \sum w_i x_i^2 - \left(\sum w_i x_i\right)^2}}$$

Ordinary Least Square Fitting

The method of ordinary least squares assumes that there is constant variance in the errors (which is called homoscedasticity). This means, when all the weights ω_i are same and equal to 1 weighted least square fitting (WLSF) reduces to ordinary least square fitting(OLSF). More specifically, we assume that the measurement of each y_i is governed by the Gauss distribution, with the same width parameter σ for all measurements.

Since $w_i = 1$, so $\sum w_i = N$, The above formulas will reduce to OLSF:

Slope

$$m = \frac{(1)\sum(x_i y_i)(N) - (1)\left(\sum x_i\right)(1)\left(\sum y_i\right)}{(N)(1)\left(\sum x_i^2\right) - (1)\left(\sum x_i\right)^2}$$

$$m = \frac{N\sum(x_i y_i) - \left(\sum x_i\right)\left(\sum y_i\right)}{N\left(\sum x_i^2\right) - \left(\sum x_i\right)^2}$$

Intercept

$$c = \frac{(1)\sum\left(x_i^2\right)(1)\sum(y_i) - (1)\left(\sum x_i\right)(1)\left(\sum x_i y_i\right)}{(N)(1)\left(\sum x_i^2\right) - (1)\left(\sum x_i\right)^2}$$

$$c = \frac{\sum\left(x_i^2\right)\sum(y_i) - \left(\sum x_i\right)\left(\sum x_i y_i\right)}{N\left(\sum x_i^2\right) - \left(\sum x_i\right)^2}$$

Error in Slope

$$\sigma_m = \sqrt{\frac{(N)}{(N)(1)\sum x_i^2 - (1)\left(\sum x_i\right)^2}}$$

$$\sigma_m = \sqrt{\frac{N}{N\sum x_i^2 - \left(\sum x_i\right)^2}}$$

Error in Intercept

$$\alpha_c = \sqrt{\frac{(1)\sum x_i^2}{N(1)\sum x_i^2 - (1)\left(\sum x_i\right)^2}}$$

$$\alpha_c = \sqrt{\frac{\sum x_i^2}{N\sum x_i^2 - \left(\sum x_i\right)^2}}$$

Correlation Coefficient

The correlation coefficient is a measure of the association between two variables. It is used to find the relationship is between data and a measure to check how strong it is. The formulas return a value between -1 and 1, where -1 shows negative correlation and +1 shows a positive correlation.

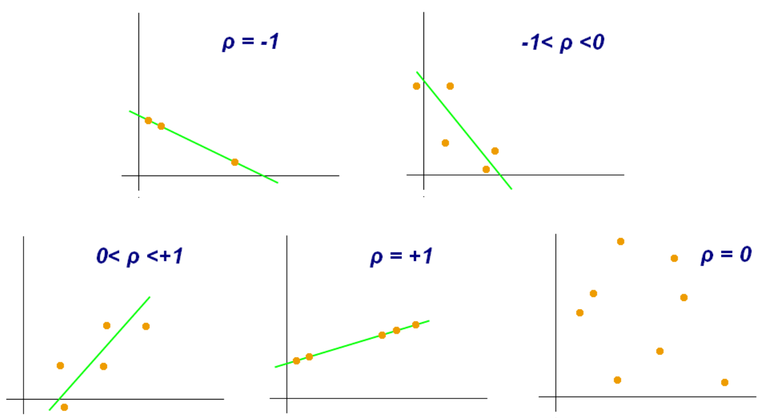

Figure 2: Correlation Coefficient

It is given by:

$$r = \frac{\sum \omega_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum \omega_i (x_i - \bar{x})^2 \sum \omega_i (y_i - \bar{y})^2}}$$

Adjusted Correlation Coefficient

The adjusted correlation coefficient is obtained by dividing the original correlation coefficient by the rematched correlation coefficient, whose sign is that of the sign of original correlation coefficient. The sign of adjusted correlation coefficient is the sign of original correlation coefficient. If the sign of the original r is negative, then the sign of the adjusted r is negative, even though the arithmetic of dividing two negative numbers yields a positive number. The expression below provides only the numerical value of the adjusted correlation coefficient.
It is given by:

$$r(adjusted) = \frac{r(original)}{r_{+ve/-ve}(rematch)}$$

Note: **Rematching**
As mentioned above, the correlation coefficient theoretically assumes values in the interval between +1 and -1, including the end values +1 or -1 (an interval that includes the end values is called a closed interval, and is denoted with left and right square brackets: [, and], respectively. Accordingly, the correlation coefficient assumes values in the closed interval [-1, +1]). However, it is not

well known that the correlation coefficient closed interval is restricted by the shapes (distributions) of the individual X data and the individual Y data. The extent to which the shapes of the individual X and individual Y data differ affects the length of the realised correlation coefficient closed interval, which is often shorter than the theoretical interval. Clearly, a shorter realised correlation coefficient closed interval necessitates the calculation of the adjusted correlation coefficient (to be discussed below).

The length of the realised correlation coefficient closed interval is determined by the process of 'rematching'. Rematching takes the original (X, Y) paired data to create new (X, Y) 'rematched-paired' data such that all the rematched-paired data produce the strongest positive and strongest negative relationships. The correlation coefficients of the strongest positive and strongest negative relationships yield the length of the realised correlation coefficient closed interval. The rematching process is as follows:

- The strongest positive relationship comes about when the highest X-value is paired with the highest Y-value; the second highest X-value is paired with the second highest Y-value, and so on until the lowest X-value is paired with the lowest Y-value.
- The strongest negative relationship comes about when the highest, say, X-value is paired with the lowest Y-value; the second highest X-value is paired with the second lowest Y-value, and so on until the highest X-value is paired with the lowest Y-value.

Programming

```
import matplotlib.pyplot as plt
import numpy as np

File_data = np.loadtxt("/home/hinton/Semester_4/MP3/Practical/p/1140.dat",skiprows
    =1,usecols=[1,2,3,4,5,6,7,8,9,10,11],delimiter=',',dtype='float')
p = [];q = [];r=[];s_dev=[];
for i in np.arange(0,10):
    mean = ((np.average((File_data[i][1:np.shape(File_data)[1]]))))
    standard_deviation = ((np.sum((File_data[i][1:np.shape(File_data)[1]]-mean)**2)
    /90))*4*(mean)**2
    weight =(1)/(standard_deviation)
    q.append(mean**2)
    r.append(File_data[i][0])
    p.append(weight)
    s_dev.append(standard_deviation)
print("masses",r)
print("y_mean",q)
print("weight",p)
N = np.array([r,q,p])
np.savetxt('/home/hinton/Semester_4/MP3/Practical/p/1140.csv',N,delimiter=',')

def Mywlsf(x_i,y_i,w_i):
    Dell = (((np.sum(w_i))*(np.sum(w_i*x_i*x_i)))-(np.sum(w_i*x_i))**2)
    c = (((np.sum(w_i*(x_i**2)))*(np.sum(w_i*y_i)))-((np.sum(w_i*x_i))*(np.sum(w_i*
    y_i*x_i))))/Dell
    m = (((np.sum(w_i))*(np.sum(w_i*y_i*x_i)))-((np.sum(w_i*x_i))*(np.sum(w_i*y_i))
    ))/Dell
    Error_in_intercept = np.sqrt((np.sum(w_i*x_i*x_i))/Dell)
    Error_in_slope = np.sqrt((np.sum(w_i))/Dell)
    chi_sq = np.sum(((y_i-(m*x_i+c))**2)*w_i)
    Y_cal = m*x_i+c
    print("intercept = ",c)
    print("slope = ",m)
    print("Error_in_intercept = ",Error_in_intercept)
    print("Error_in_slope = ",Error_in_slope)
    print("chi_sq = ",chi_sq)
    print("Errors = ",Y_cal-y_i)
    print("Sum of residuals = ",(np.sum((Y_cal-y_i)*w_i)))
    print("Sum of square of residuals", (np.sum(w_i*(Y_cal-y_i))**2))
    print("Fitted Values = ",Y_cal)
    print("---------------------------------------------------------")
    print("spring constant = ", 4*np.pi**2/m)
    print("m = ", (4*(np.pi**2)/m)*(c/(4*np.pi**2)))
    fig1, (ax1,ax2) = plt.subplots(2,1)
    ax1.scatter(x_i,y_i,label="observed points")
    ax1.plot(x_i,Y_cal,label="fitted points")
    ax1.set_ylabel('$T^2$')
    ax1.set_title("Wlsf")
    ax1.errorbar(x_i, y_i, (Y_cal-y_i),fmt='o',label="error")
    ax1.legend(loc='upper left')
    ax1.grid(True)
    ax2.scatter(x_i,y_i,label="observed points")
    ax2.plot(x_i,Y_cal,label="fitted points")
    ax2.set_ylabel('$T^2$')
    ax2.set_xlabel('$M$')
    ax2.set_ylim(0.57,0.59)
    ax2.set_xlim(154,156)
    ax2.errorbar(x_i, y_i, (Y_cal-y_i),fmt='o',label="error")
    ax2.legend(loc='upper left')
    ax2.grid(True)
    plt.savefig("/home/hinton/Semester_4/MP3/Practical/p/1140.png")
    plt.show()

def Mylsf(x_i,y_i):
    print("--------------------------------------------------------")
```

```
    print("Least Square Fitting")
    Dell = np.size(x_i)*(np.sum(x_i**2))-(np.sum(x_i))**2
    c = (((np.sum(x_i**2))*(np.sum(y_i)))-((np.sum(x_i))*(np.sum(y_i*x_i))))/Dell
    m = (np.size(x_i)*np.sum(x_i*y_i)-np.sum(x_i)*np.sum(y_i))/Dell
    Y_cal = m*x_i+c
    S = np.sqrt(np.sum((y_i-Y_cal)**2)/(np.size(x_i)-2))
    Error_in_intercept = S*np.sqrt((np.sum(x_i**2)/(np.size(x_i)*np.sum(x_i**2)-(np.sum(x_i))**2)))
    Error_in_slope = S*np.sqrt((np.size(x_i)/(np.size(x_i)*np.sum(x_i**2)-(np.sum(x_i)**2))))
    chi_sq = np.sum(((y_i-(m*x_i+c))**2))
    Y_cal = m*x_i+c
    print("intercept = ",c)
    print("slope = ",m)
    print("Error_in_intercept = ",Error_in_intercept)
    print("Error_in_slope =",Error_in_slope)
    print("chi_sq = ",chi_sq)
    print("Errors = ",Y_cal-y_i)
    print("Sum of residuals = ",np.sum(y_i-(m*x_i+c)))
    print("Sum of square of residuals", np.sum(((y_i-(m*x_i+c))**2)))
    print("Fitted Values =",Y_cal)
    fig1, (ax1,ax2) = plt.subplots(2,1)
    ax1.scatter(x_i,y_i,label="observed points")
    ax1.plot(x_i,Y_cal,label="fitted points")
    ax1.set_ylabel('$T^2$')
    ax1.set_title("LSF")
    ax1.errorbar(x_i, y_i, Y_cal-y_i,fmt='o',label="error")
    ax1.legend(loc='upper left')
    ax1.grid(True)
    ax2.scatter(x_i,y_i,label="observed points")
    ax2.plot(x_i,Y_cal,label="fitted in y")
    ax2.set_ylabel('$T^2$')
    ax2.set_xlabel('$M$')
    ax2.set_ylim(0.57,0.59)
    ax2.set_xlim(154,156)
    ax2.errorbar(x_i, y_i, (Y_cal-y_i),fmt='o',label="error")
    ax2.legend(loc='upper left')
    plt.grid(True)
    plt.savefig("/home/hinton/Semester_4/MP3/Practical/p/1140_1.png")
    plt.show()


Mywlsf(N[0],N[1],N[2])
Mylsf(N[0],N[1])
```

Discussion

As the error is less for approximated values and given data points for Weighted Least Square Fitting compared to Least Square Fitting. So, Weighted Least Square Fitting gives better approximations.

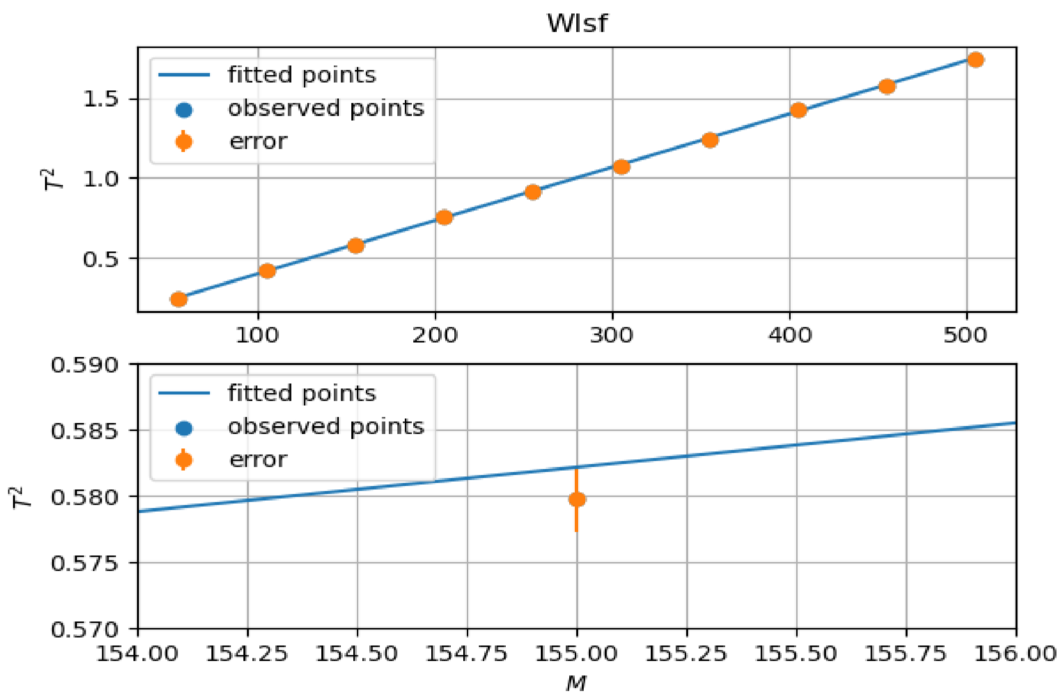

Figure 3: Weighted Least Square Fitting

```
(base) hinton@hinton-VirtualBox:~$ conda run -n base --no-capture-output --live-stream python /home/hinton/Semester_4/MP3/Practical/p/wlsf.py
masses [55.0, 105.0, 155.0, 205.0, 255.0, 305.0, 355.0, 405.0, 455.0, 505.0]
y_mean [0.2457184900000001, 0.4173160000000002, 0.5797299600000001, 0.7530768400000002, 0.9168062500000003, 1.07702884, 1.24233316, 1.4253972099999999
, 1.57351936, 1.7471552399999997]
weight [4470.208633396492, 7669.4135620073685, 2721.223173889535, 1661.0014254719658, 1736.5446673410365, 3293.2097683798224, 4672.140085010225, 12679
.792506904005, 856.113897649963, 2150.791162573701]
intercept =  0.06334311929953071
slope =  0.003347233933131487
Error_in_intercept =  0.01050682647979881
Error_in_slope =  3.3812126862044396e-05
chi_sq =  1.4286171475946805
Errors =  [ 1.72249562e-03 -2.51331772e-03  2.43441893e-03 -3.55076441e-03
  8.15222481e-05  7.22062890e-03  9.27800556e-03 -6.42434778e-03
  1.28151989e-02  6.54101553e-03]
Sum of residuals =  -2.69420041831836e-11
Sum of square of residuals 7.258715894066825e-22
Fitted Values =  [0.24744099 0.41480268 0.58216438 0.74952608 0.9168877 1.08424947
 1.25161117 1.41897286 1.58633456 1.75369626]
```

Figure 4: Weighted Least Square Fitting Results

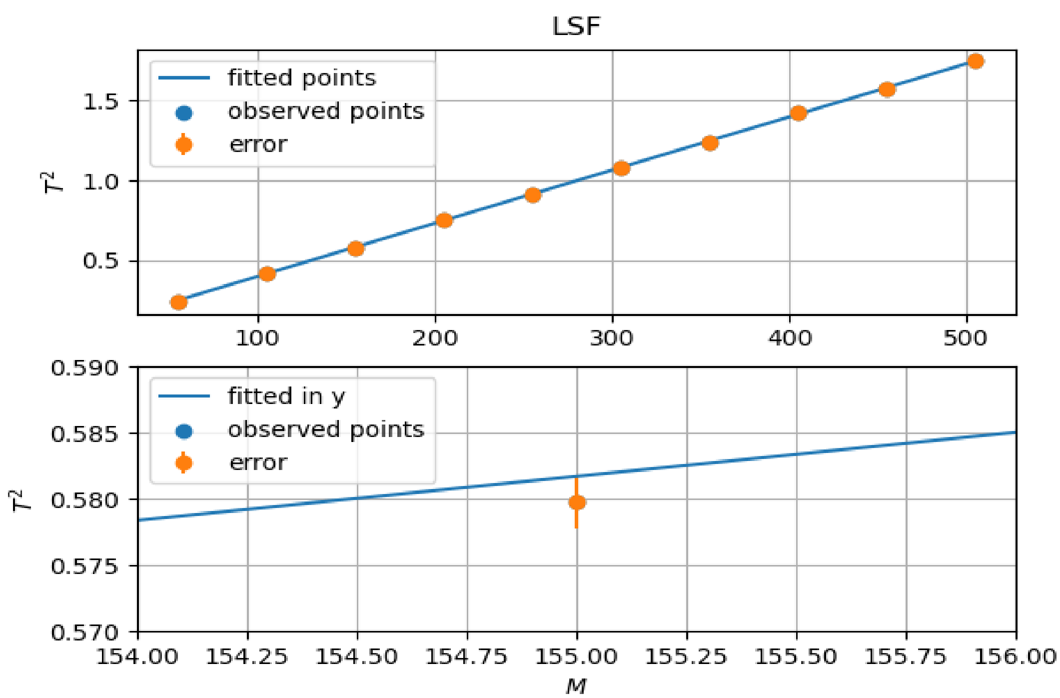

Figure 5: Least Square Fitting

```
Least Square Fitting
intercept =  0.06574134959393921
slope =  0.0033288099478787875
Error_in_intercept =  0.004011369688655981
Error_in_slope = 1.2747360352292707e-05
chi_sq =  0.0002681170733194866
Errors =  [ 0.00310741 -0.00204961  0.00197693 -0.00492945 -0.00221836  0.00399954
  0.00513572 -0.01148783  0.00683052 -0.00036487]
Sum of residuals =  3.58046925441613e-15
Sum of square of residuals 0.0002681170733194866
Fitted Values = [0.2488259  0.41526639 0.58170689 0.74814739 0.91458789 1.08102838
 1.24746888 1.41390938 1.58034988 1.74679037]
```

Figure 6: Least Square Fitting Results

```
...........................................................
spring constant =  11794.340758078899
m =  18.92401922451547
...........................................................
```