weighted least square fitting
Lab Report for Assignment No. 6

SHASHVAT JAIN
(2020PHY1114)

HARSH SAXENA
(2020PHY1162)

S.G.T.B. Khalsa College, University of Delhi, Delhi-110007, India.

March 14, 2022

Submitted to Dr. Mamta
"32221401 - MATHEMATICAL PHYSICS III"

Contents

1 Theory **1**

1.1 Maximum likelihood and Least squares 1

1.2 Weighted Mean and error in mean 2

1.3 Derivation of Slope and Intercept 3

1.4 Derivation of Error in Slope and Intercept 4

1.5 Weighted least square fitting reduces to ordinary LSF 6

2 Programming **6**

3 Analysis **8**

1 Theory

1.1 Maximum likelihood and Least squares

The method of least squares can be derived from the formalism of **maximum likelihood** in conjunction with the central limit theorem, which motivates the statement that each data point that we measure, y_i , is drawn from a Gaussian distribution with a width given by the standard error,σ_i
The y-coordinate of the line of best fit at x_i , $y(x_i)$,is the most probable value of the mean of the parent distribution. We can use the parent distribution to calculate the probability of obtaining the value y_i , given the parameters m and c, which is proportional to the value of the probability density function at y_i.The assumption that we make is that the parent distribution is described by the **Gaussian probability** density function.
the data set has a uncertainity σ_i in y_i.
Let $y = f(x : m, c)$ are the set of parameters to be estimated.Then from the central limit theoram,distribution of measured y values about their ideal value is gaussian.
Then probability for a particular y_i for given x_i is-

$$P(y_i : m, c) = \frac{1}{\sigma_i\sqrt{2\pi}} exp\left(\frac{-(y_i - f(x_i : m, c)}{\sigma_i}\right) \tag{1}$$

Thus,Maximizing the maximum likelihood function of the estimators $\hat{m}$ and $\hat{c}$ is similar to minimizing -

$$\chi^2 = \sum \left[\frac{(y_i - f(x_i : m, c)}{\sigma_i}\right]^2 \tag{2}$$

Using this χ^2 we will determine the estimators $\hat{m}$ and $\hat{c}$ for a function given by,$y = mx + c$.

$$\chi^2 = \sum \left[\frac{(y_i - mx_i - c)}{\sigma_i}\right]^2$$
$$\chi^2 = \sum w_i(y_i - mx_i - c)^2, w_i = \frac{1}{\sigma_i^2}$$

1.2 Weighted Mean and error in mean

When taking the weighted mean of a series of measurements, it is important that the compatibility of the results is considered: Combine multiple measurements of the same quantity only if they are consistent with each other.

Let the result for one experiment is $x_i \pm \alpha_i$ and those of another be $x_j \pm \alpha_j$. If the two results have similar error than the mean of the two results would be the best estimated value as this give equal importance to both the data. And it is given by:

$$\bar{x}_{i,j} = \frac{1}{2}(x_i + x_j)$$

It can be shown that the error in the weighted mean is given by:

$$\frac{1}{\alpha_{(\bar{x}_i,j)}} = \sqrt{\frac{1}{\alpha_i^2} + \frac{1}{\alpha_j^2}}$$

The best combined estimate, x_{CE} , incorporating all of the available data is the sum of the weighted means, normalised by the sum of the weightings:

$$x_{CE} = \frac{\sum_i w_i x_i}{\sum_i w_i}$$

Consider N measurement of the same quantity, x_i, with $i = 1, 2, 3, ...N$
we can take σ to be the error in one of these measurements. As the error is the same for each x_i , each measurement carries the same weight,

$$\therefore x_{CE} = \frac{\sum_i^N w_i x_i}{\sum_i^N w_i}$$

$$x_{CE} = \frac{\sum_i^N w_i}{N} = \bar{x}$$

This gives the expected result that the combined estimate is the mean of the measurements. In a similar manner we can calculate the standard error of the weighted mean to be

$$\frac{1}{\alpha_{CE}^2} = \sum \frac{1}{\sigma^2} = \frac{N}{\sigma^2}$$

$$\boxed{\alpha_{CE} = \frac{\sigma}{\sqrt{N}}}$$

1.3 Derivation of Slope and Intercept

We have to minimize χ^2 w.r.t m and c then.

$$\frac{\partial \chi^2}{\partial m} = 0$$
$$\sum \frac{\partial [w_i(y_i - mx_i - c)^2]}{\partial m} = 0$$
$$-2\sum [w_i x_i (y_i - mx_i - c)] = 0$$
$$\sum w_i x_i y_i - m\sum w_i x_i^2 - c\sum w_i x_i = 0$$
$$\frac{\partial \chi^2}{\partial c} = 0$$
$$-2\sum [w_i (y_i - mx_i - c)] = 0$$
$$\sum w_i y_i - m\sum w_i x_i - c\sum w_i = 0$$

$$\implies c = \frac{\sum w_i y_i - m\sum w_i x_i}{\sum w_i}$$
$$\boxed{c = \overline{Y} - m\overline{X}}$$
$$\overline{Y} = \frac{\sum w_i y_i}{w_i}, \overline{X} = \frac{\sum w_i x_i}{w_i}$$

Putting c in eq 8.

$$\sum w_i x_i y_i - m\sum w_i x_i^2 - (\overline{Y} - m\overline{X})\sum w_i x_i = 0$$
$$m = \frac{\sum w_i x_i y_i - \overline{Y}\sum x_i}{\sum w_i x_i^2 - \overline{X}\sum w_i x_i}$$
$$\boxed{m = \frac{\sum w_i \sum w_i x_i y_i - \sum w_i y_i \sum x_i}{\sum w_i \sum w_i x_i^2 - (\sum w_i x_i)^2}}$$
$$\boxed{c = \frac{\sum w_i x_i^2 \sum w_i y_i - \sum w_i x_i \sum w_i x_i y_i}{\sum w_i \sum w_i x_i^2 - (\sum w_i x_i)^2}}$$

$$\Delta = \sum w_i \sum w_i x_i^2 - (\sum w_i x_i)^2, S_{xy} = \sum w_i x_i y_i$$
$$S_x = \sum w_i x_i, S_{x^2} = \sum w_i x_i^2, S_y = \sum w_i y_i$$

1.4 Derivation of Error in Slope and Intercept

Since,the constant m depends on x_i and y_i but only y_i have uncertainity around therefore,by propagation of errors.

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

$$\frac{\partial m}{\partial y_i} = \frac{(\sum w_i) w_i x_i - (\sum w_i x_i) w_i}{\Delta}$$

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

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

$$\sigma_m^2 = \sum \frac{w_i \left[(\sum w_i)^2 x_i^2 + (\sum w_i x_i)^2 - 2(\sum w_i \sum w_i x_i) x_i\right]}{\Delta^2}$$

$$\sigma_m^2 = \sum \frac{w_i \left[(\sum w_i)^2 x_i^2 + (\sum w_i \overline{X})^2 - 2(\sum w_i)^2 \overline{X} x_i\right]}{\Delta^2}$$

$$\sigma_m^2 = \sum \frac{w_i \left[(\sum w_i)^2 (x_i^2 + \overline{X}^2 - 2\overline{X} x_i)\right]}{\Delta^2}$$

$$\sigma_m^2 = \sum \frac{w_i \left[(\sum w_i)^2 (x_i - \overline{X})^2\right]}{\Delta^2}$$

$$\sigma_m^2 = \frac{(\sum w_i)^2 \sum w_i (x_i - \overline{X})^2}{\Delta^2}$$

$$\sigma_m^2 = \frac{\sum w_i}{\Delta}$$

$$\sigma_m = \sqrt{\frac{\sum w_i}{\Delta}}$$

Similarly for intercept we can write.

$$\sigma_c^2 = \sum \left(\frac{\partial c}{\partial y_i}\right)^2 \sigma_i^2$$
$$\frac{\partial c}{\partial y_i} = \frac{(\sum w_i x_i^2) w_i - (\sum w_i x_i) w_i x_i}{\Delta}$$
$$\left(\frac{\partial c}{\partial y_i}\right) \sigma_i = \frac{\frac{\sum w_i x_i^2}{\sigma_i} - \frac{(\sum w_i x_i) x_i}{\sigma_i}}{\Delta}$$
$$\sigma_c^2 = \sum \frac{\left[\frac{\sum w_i x_i^2}{\sigma_i} - \frac{(\sum w_i x_i) x_i}{\sigma_i}\right]^2}{\Delta^2}$$
$$\sigma_c^2 = \sum \frac{w_i \left[(\sum w_i x_i^2)^2 + (\sum w_i x_i)^2 x_i^2 - 2(\sum w_i x_i^2)(\sum w_i x_i) x_i\right]}{\Delta^2}$$
$$\sigma_c^2 = \sum \frac{w_i \left[(\sum w_i x_i^2)^2 - (\sum w_i x_i^2)(\sum w_i x_i) x_i + (\sum w_i x_i)^2 x_i^2 - (\sum w_i x_i^2)(\sum w_i x_i) x_i\right]}{\Delta^2}$$
$$\sigma_c^2 = \sum \frac{w_i \left[\sum w_i x_i^2 (\sum w_i x_i^2 - (\sum w_i x_i) x_i) + \sum w_i x_i ((\sum w_i x_i) x_i^2 - (\sum w_i x_i^2) x_i)\right]}{\Delta^2}$$
$$\sigma_c^2 = \sum \frac{\left[\sum w_i x_i^2 (\sum w_i \sum w_i x_i^2 - (\sum w_i x_i)^2) + \sum w_i x_i ((\sum w_i x_i) \sum w_i x_i^2 - (\sum w_i x_i^2) w_i x_i)\right]}{\Delta^2}$$
$$\sigma_c^2 = \frac{\sum w_i x_i^2}{\Delta}$$
$$\sigma_c = \sqrt{\frac{\sum w_i x_i^2}{\Delta}}$$

Correlation coefficient

Correlation Coefficient is a statistical concept, which helps in establishing a relation between predicted and actual values obtained in a statistical experiment. The calculated value of the correlation coefficient explains the exactness between the predicted and actual values. Possible values of the correlation coefficient range from -1 to +1, with -1 indicating a perfectly linear negative, i.e., inverse, correlation (sloping downward) and +1 indicating a perfectly linear positive correlation (sloping upward) and a correlation coefficient close to 0 suggests that there is no specific relation between two variables.

$$r = \frac{\sum w_i (x_i - \overline{X})(y_i - \overline{Y})}{\sqrt{\sum w_i (x_i - \overline{X})^2 \sum w_i (y_i - \overline{Y})^2}}$$

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.

$$r_{x,y}(adjusted) = \frac{r_{x,y}(Calculated)}{r_{x,y}(positiverematch)}$$

1.5 Weighted least square fitting reduces to ordinary LSF

takeing here weights to be constant i.e, $w = \frac{1}{\sigma}$.
Then the formula for slope and intecept changes to-

$$m = \frac{\sum w \sum wx_i y_i - \sum wy_i \sum x_i}{\sum w \sum wx_i^2 - (\sum wx_i)^2}$$
$$m = \frac{N \sum x_i y_i - \sum y_i \sum x_i}{N \sum x_i^2 - (\sum x_i)^2}$$

similarly

$$c = \frac{\sum wx_i^2 \sum wy_i - \sum wx_i \sum wx_i y_i}{\sum w \sum wx_i^2 - (\sum wx_i)^2}$$
$$c = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{N \sum x_i^2 - (\sum x_i)^2}$$

2 Programming

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress
from scipy.optimize import curve_fit
from matplotlib import use
plt.style.use("bmh")
use("WebAgg")
dat= np.loadtxt("data-lsf.csv",delimiter=",")

inptdat = np.zeros((dat.shape[0],3))

inptdat[:,0]= dat[:,0]
inptdat[:,1]= np.mean(dat[:,1:],axis = 1)**2
std = 2*np.sqrt(inptdat[:,1])*np.std(dat[:,1:],axis = 1,ddof=1)/np.sqrt(dat.shape
    [1]-1)
inptdat[:,2]= 1/(std)**2

'''
for i in range(0,dat.shape[0]): # for each row
    yi = dat[i,1:dat.shape[1]]
    inptdat[i,1]= np.mean(yi)
    inptdat[i,2]= len(yi)/(np.sum((yi-inptdat[i,1])**2)/(len(yi)-1))
'''

def Mywlsf(inpt,weights=False):
    if inpt.shape[1]>3:
        raise ValueError("shape of input matrix must be (N,2) or (N,3) for weighted."
    )
    x,y,w = inpt[:,0],inpt[:,1],inpt[:,2]
    if not weights or inpt.shape[1] == 2 :
        w = np.ones(x.shape,dtype=float)
    x_mean,y_mean = np.average(x,0,w),np.average(y,0,w)
    ss_xx = w.dot((x-x_mean)**2)
    ss_yy = w.dot((y-y_mean)**2)
    ss_xy = w.dot((x-x_mean)*(y-y_mean))
    [ss_x,s_x,s_w] = np.sum([w*x**2,w*x,w],axis=1)
    delta = (s_w*ss_x - s_x**2)
    a = np.array([ss_xy/ss_xx,y_mean-ss_xy/ss_xx*x_mean])
    resi = y - a[0]*x - a[1]
```

```
    r_arr = np.sum([resi,resi**2],axis=1)
    #a = np.array([(s_w*s_xy - s_x*s_y),(ss_x*s_y - s_x*s_xy)])/delta
    if not weights:
        s = np.sqrt(r_arr[1]/(len(x)-2))
        e_a = np.array([s/np.sqrt(ss_xx),s*np.sqrt(1/len(x)+x_mean**2/ss_xx),])
    else :
        e_a = np.sqrt(np.array([s_w,ss_x])/delta)
    corr = [np.sqrt(ss_xy**2/(ss_xx*ss_yy)),(resi**2*w).sum()]
    return(a,e_a,r_arr,corr)

if __name__ == "__main__":
    def get_km(p,e):
        k= 4*np.pi**2/p[0]
        m = 3*p[1]/p[0]
        err_k = 4*np.pi**2/p[0]**2 * e[0]
        err_m = m*np.sqrt((e[0]/p[0])**2 + (e[1]/p[1])**2)
        return(k,m,err_k,err_m)


    np.savetxt("1114.csv",inptdat)
    se = np.sqrt(1/inptdat[:,2])
    #--------------lsf
    (params2,err2,resi2,corr2) = Mywlsf(inptdat)
    k1,m1,err_k1,err_m1 = get_km(params2,err2)
    print("\nlsf : Parameters => ",params2,"\n Error in params => ",err2,"\n Sum of
    residuals and sum of square of residuals => ",resi2,"\n corr coef and chi^2 => ",
    corr2)
    inbuilt = linregress(inptdat[:,0],inptdat[:,1])
    print(inbuilt)
    print("Value of spring constant(k) => ",k1,"+-",err_k1,"\nValue of effective mass
    of spring(3m) => ",m1,"+-",err_m1)
    #--------------wlsf
    (params,err,resi,corr) = Mywlsf(inptdat,weights=True)
    k,m,err_k,err_m = get_km(params,err)
    print("\nWeighted lsf : Parameters => ",params,"\n Error in params => ",err,"\n
    Sum of residuals and sum of square of residuals => ",resi,"\n corr coef and chi^2
    => ",corr)

    print("Value of spring constant(k) => ",k,"+-",err_k,"\nValue of effective mass
    of spring(3m) => ",m,"+-",err_m)
    p,pcov = curve_fit(lambda x,a,b:a*x+b,inptdat[:,0],inptdat[:,1],sigma=std ,
    absolute_sigma=True)
    print("\nInbuilt scipy.optimize.curve_fit => ",p,np.sqrt(np.diag(pcov)))

    #np.savetxt("1114.out",[k,m])
    fig,ax = plt.subplots(1,1)
    ax.errorbar(inptdat[:,0],inptdat[:,1],se,fmt= ".",color='black',
                ecolor='red', elinewidth=3, capsize=0,label = "Data-points, error
    bars 1STE")
    ax.plot(inptdat[:,0],inptdat[:,0]*params[0]+params[1],c ='cyan',label = "weighted
    regression line")
    ax.plot(inptdat[:,0],inptdat[:,0]*params2[0]+params2[1],ls = '--',c='green',label
    = "Ordinary regression line")
    ax.set_xlabel("$M$");ax.set_ylabel("$T^2$");
    ax.legend()
    fig,ax = plt.subplots(1,1)
    ax.scatter(inptdat[:,0],inptdat[:,1],marker= ".",color='black',label = "Data-
    points")
    ax.plot(inptdat[:,0],inptdat[:,0]*params2[0]+params2[1],c='cyan',label = "
    Ordinary regression line")
    ax.set_xlabel("$M$");ax.set_ylabel("$T^2$")
    ax.legend()
```

```
    plt.legend()
    plt.show()
```

3 Analysis

The obtained values of weights were of the order of 3-4. The weighted least squares produces parameters(slope and intercept) that give relatively less preference to the data-points in our sample with larger values of standard error, that is, the obtained regression line is such that the absolute value of residuals is greater near the data-points with smaller standard error, whereas unweighted least squares regression treats all points the same, thus resulting in smaller standard deviation in residuals.

Finally, we obtain the following result

```
lsf : Parameters =>  [0.00332881 0.06574135]
 Error in params =>  [1.27473604e-05 4.01136969e-03]
 Sum of residuals and sum of square of residuals =>  [3.33066907e-16 2.68117073e-04]
 corr coef and chi^2 =>  [0.9999413478000886, 0.0002681170733194866 7]
LinregressResult(slope=0.003328809947878788, intercept=0.0657413495939394, rvalue=0.9999413478000887, pva
lue=5.1770780785256146e-17, stderr=1.2747360352259549e-05, intercept_stderr=0.004011369688645547)
Value of spring constant(k) =>  11859.618969690417 +- 45.415280239674104
Value of effective mass of spring(3m) =>  59.24761457393965 +- 3.622251019875521

Weighted lsf : Parameters =>  [0.00334723 0.06334312]
 Error in params =>  [3.38121269e-05 1.05068265e-02]
 Sum of residuals and sum of square of residuals =>  [-0.02760486  0.00041433]
 corr coef and chi^2 =>  [0.9999271195187265, 1.4286171475946885]
Value of spring constant(k) =>  11794.340758078899 +- 119.14068569245693
Value of effective mass of spring(3m) =>  56.77205767354707 +- 9.434320070641816

Inbuilt scipy.optimize.curve_fit =>  [0.00334723 0.06334312] [3.38121332e-05 1.05068274e-02]
```

Figure 1: Weighted and ordinary least squares regression for given data(M, T^2)

Relation between σ_y^2 and $\sigma_{\overline{T}}2$

Since we know that $y = \overline{T}^2$

$$\sigma_y^2 = \left(\frac{\partial y}{\partial \overline{T}}\right)^2 \sigma_{\overline{T}}^2$$
$$\sigma_y^2 = 4\overline{T}^2 \sigma_{\overline{T}}^2$$

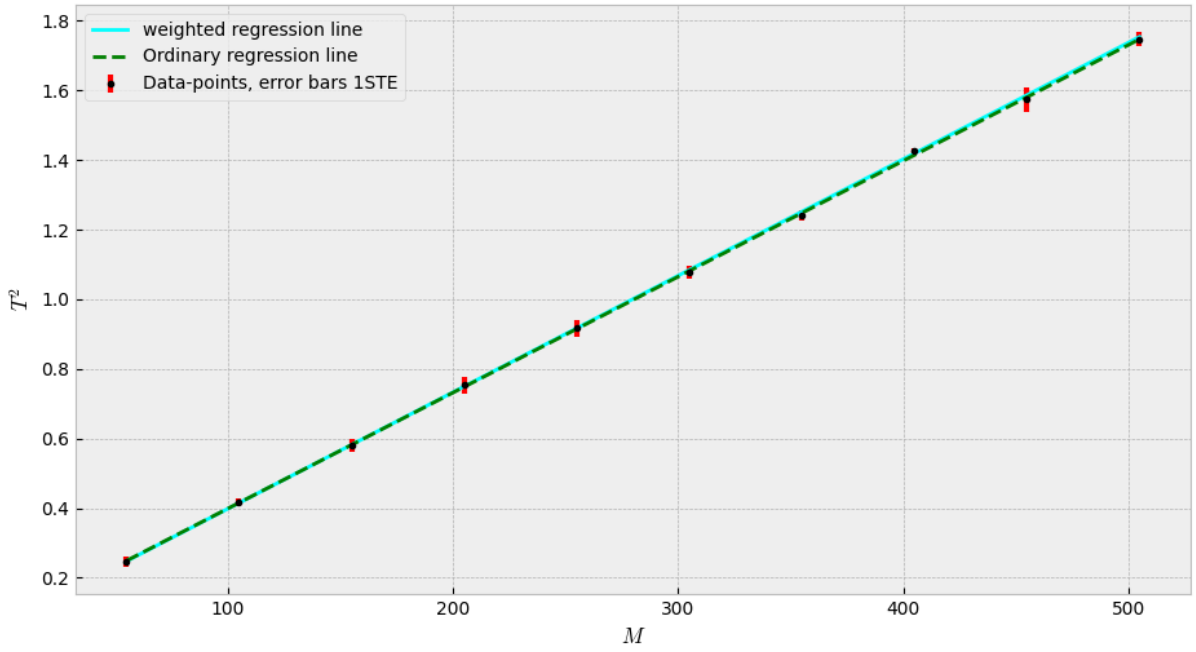

Figure 2: Weighted and ordinary least squares fit for given data(M, T^2)

We re-iterate our above mentioned point, the regression line for Weighted least squares passes nearer to the data-point with smaller error bar(larger weight) and farther away from the point with larger error bar(smaller weight) as compared to the Ordinary least squares regression line.

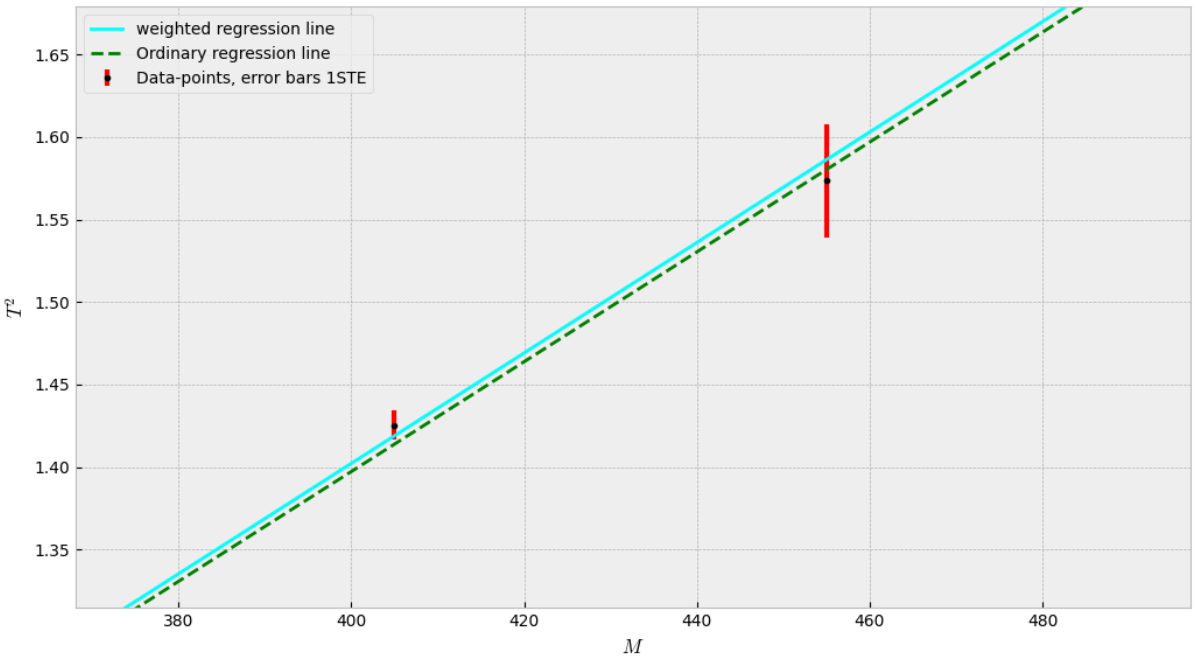

Figure 3: Weighted and ordinary least squares fit for given data(M, T^2)

This observation is the only but a very important difference that weighted least squares offers over ordinary least squares, it is able to account for an extra information associated with the data-points and accordingly set weights, prioritizing measurements made with relatively smaller uncertainty.