

Name: Xunyi LU
ID: 201600688



1. How the programme work

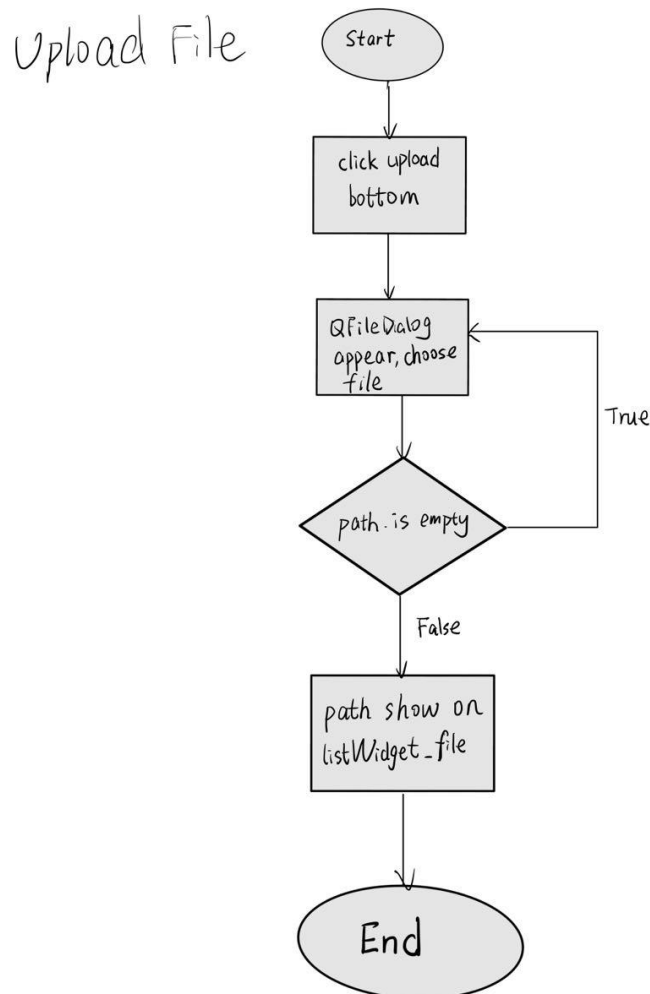
Generally, my programme has 4 main functions. 1. Upload databases (in this programme they can only be *.csv file) to the application. 2. Plot the selected databases. 3. Change colour of plots. 4. Database calculator: can perform add, minus, sqrt, Ln of databases, and plot the result.

Next, I would like to explain the software, following the four functions mentioned above.

1.1 Upload file

I use **QFileDialog** to select database, which will get the path of the selected file. And I use a filter "*.csv" to only select csv file. The dialog is triggered by clicking a button.

I use a **ListWidget listWidget_file** to show the uploaded files to user.



1.2 Plot

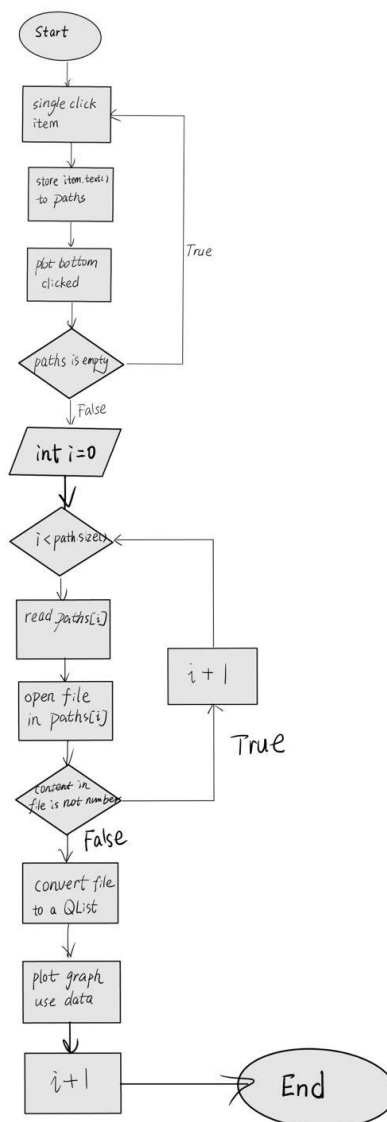
I use **QCustomPlot** to plot the graph.

After user uploaded some files, the paths of the files will be shown on the listWidget. When user want to plot a database, he needs to single clicked the line on the **listWidget_file**, then this signal sent, the slot is: The content of this line, which is the path of the file, will be stores in a global variable **QStringList paths**; There is an important function here, I use a new **QStringList distpaths** to avoid duplications. So if user clicked one line multiple times, the graph will still be plot one time.

When Plot button is clicked, programme will read every line is the **paths**. And put each path to the function **QList<QList<double>> read_csvFile(QString path)**. This function will open the file of this path, read the content of this file, then return a double type of list: **QList<QList<double>> data**.

We will use

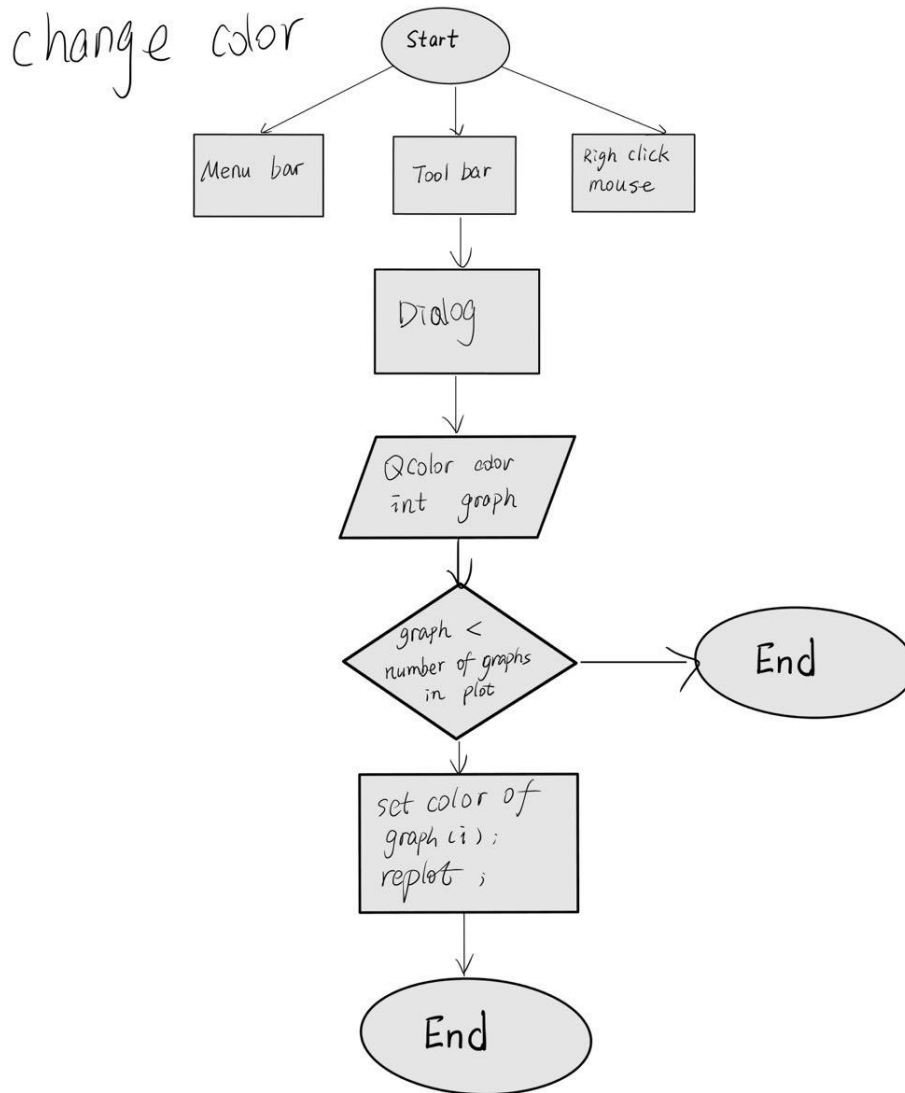
plot



1.3 Change colour

- Use menu bar
- Use tool bar
- Right clicked mouse

I use these three ways above to change the color of the graph. The 3 ways will create an action. In the action, it will create a dialog. Dialog will ask the user of the number of graph and the color. Here I use a button, when button clicked, a **QColorDialog color** appears ask for the user, and to return the colour the user select.



1.4 Database calculator

I create a new listWidget_function to store the equation. When user double clicked the item in the listWidget_file, the content will be added to listWidget_function. There are four buttons to perform +, -, $\sqrt{\text{sqrt}}$, Ln. When clicked the concerning button, the content will show on listWidget_function. When equal button clicked, it starts to perform the calculator function.

Here I use two important variable, int status and int symbol.

Uses of status:

1. set to 1 when read as files
2. set to 2 when read as addition or subtraction
3. set to 3 when reading log or sqrt

The 3 possible situations:

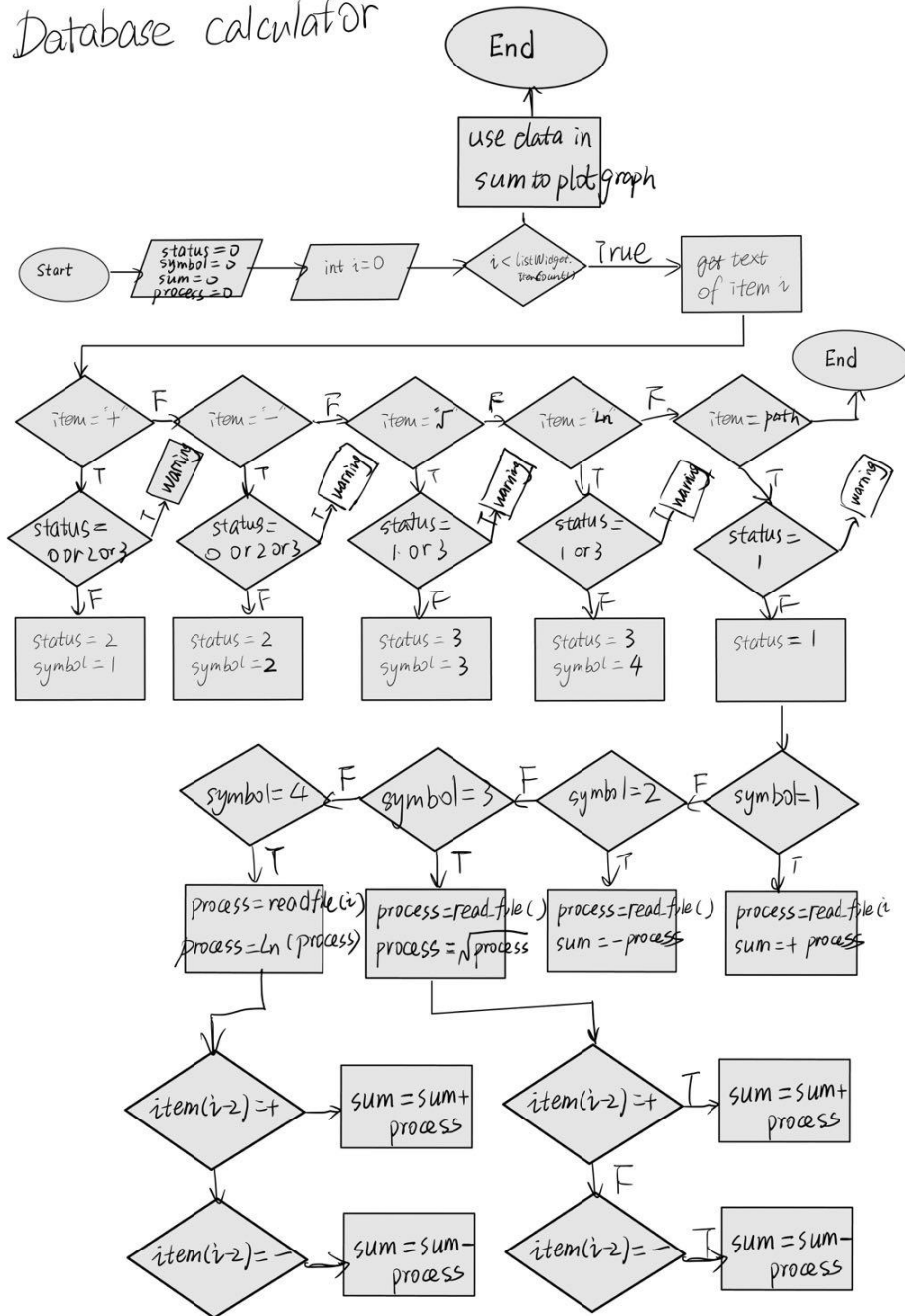
- a. When reading + -, the previous one must be path, so if status=1, you can proceed. Errors are reported for 2, 3 and 0
- b. When reading log sqrt, the previous one can be addition or subtraction, not path, and not itself. So when status=2, it can proceed. When it is 1, 3 gives an error
- c. When reading path, the previous one can be additive or subtractive and can be sqrt log, but cannot be path. so status = 0, 2, 3, can proceed, 1 gives an error.

Uses of symbol:

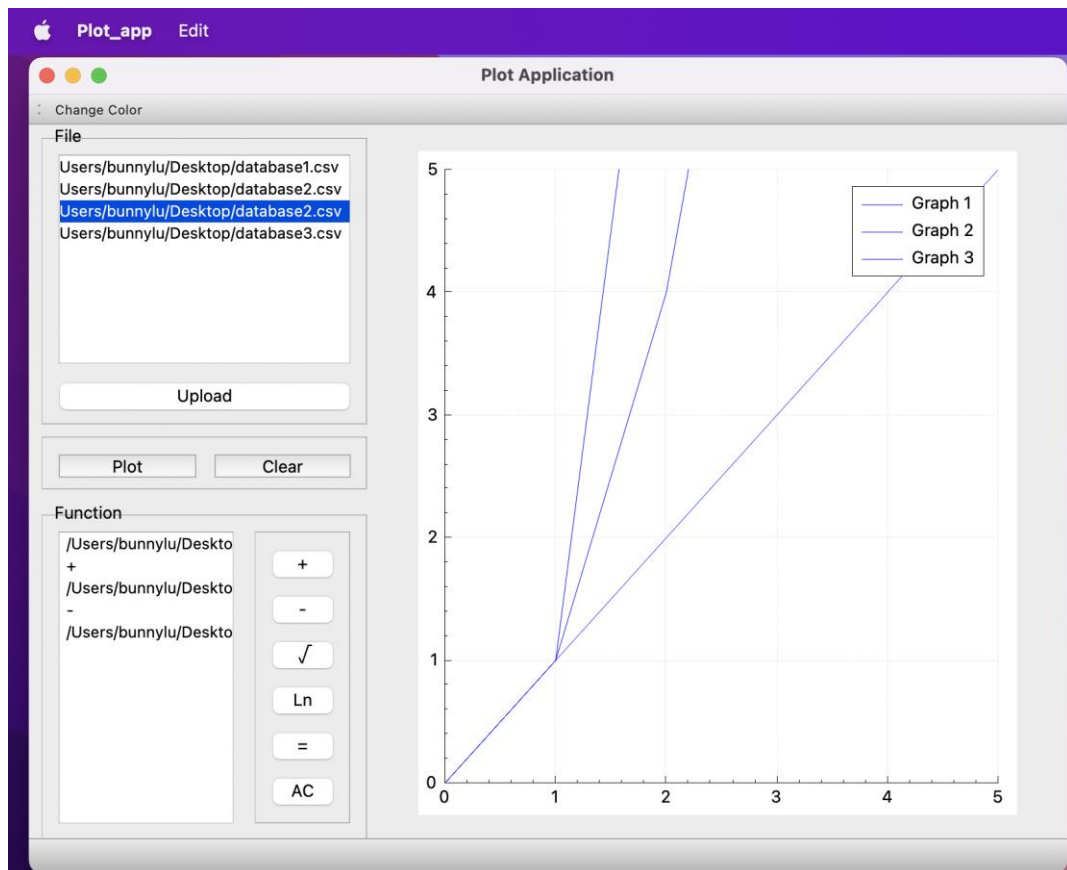
- + is 1;
- is 2;
- $\sqrt{\text{sqrt}}$ is 3;
- Ln is 4;

When the address is read, look at the symbol to select the sum and the current paths(files) for calculation.

Database calculator

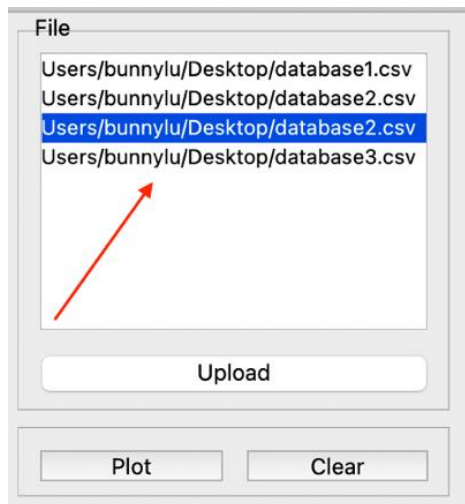


2. User Instructions

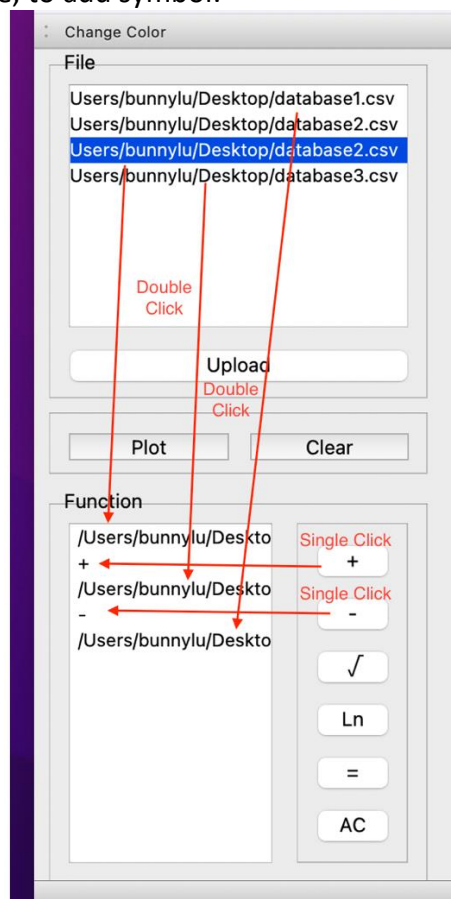


1. Upload file:
Click the Upload button, choose a (*.csv) file. The chosen file will be shown on File listwidget.
2. Plot:
Single click the line in File listWidget which you want to plot. And click Plot button, the graph will be plotted. Then click Clear button. The graph will disappear. Here is one thing that, if you user clicked Plot button, and you want to click Plot bottom again, you MUST click Clear button before you click Plot button, or the graph will be something wrong.
And after user clicked Plot bottom, the "item clicked" information will be deleted. User should re-click each database he wanted to plot. It is a potential runtime error. However, it is still unsolved.

If you want to plot database1, database2, database3, single clicked database1/2/3, and click plot. If you click the three lines multiple times by mistake, or the file is the same, it will only be plot once to avoid duplications.



3. You can click Clear button to clear all the graphs. And user would have to clicked Clear bottom before user wants to clear Plot bottom again.
4. When you use the Function, you can double click the line in File widget, and the content you clicked will appear on the Function widget. And single click the button on the right side, to add symbol.



When click “=” bottom, the result will be plot.

When “AC” button clicked, all content in the Function widget will be remove. If you want to re-use calculator, you should re-double click them.

You may follow the calculator rule:

- (1) "+", and "-" cannot be the first item.
- (2) 2 or more than 2 items should be entered.
- (3) "+", and "-" cannot be followed by a "+", and "-", such as Data1 + +Data2 will be incorrect.
- (4) A path cannot be followed by a path, such as Data1 data2 – data2 will be incorrect.
- (5) A "√" or a "Ln" cannot be followed by them.
- (6) If you want to use √ the numbers in the file cannot less than zero.
- (7) If you want to use Ln, the numbers in the file cannot ≤ 0 .

You can enter the equation like this:

Path of database1

+

Path of database2

-

Ln

Path of database2

Or

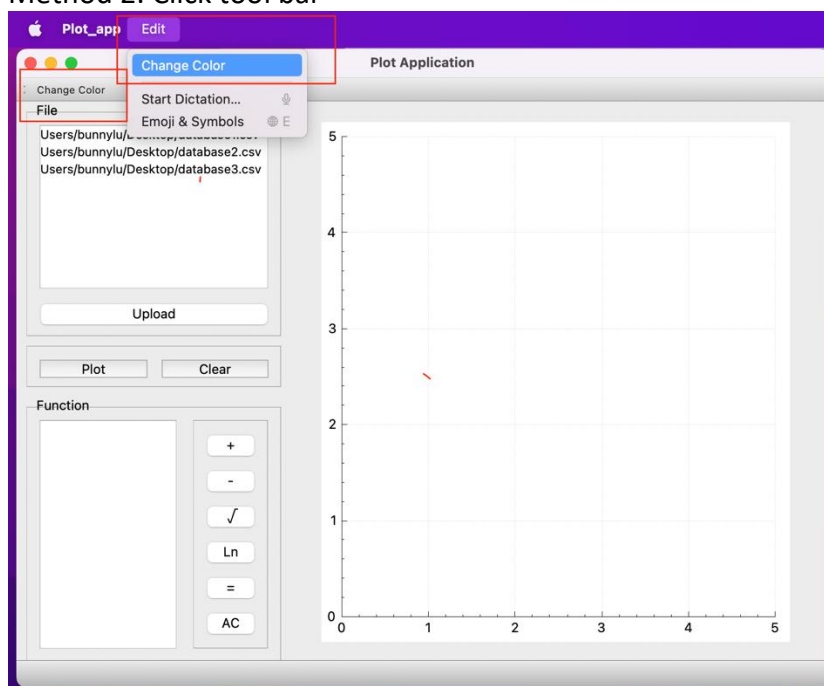
√

Path of database

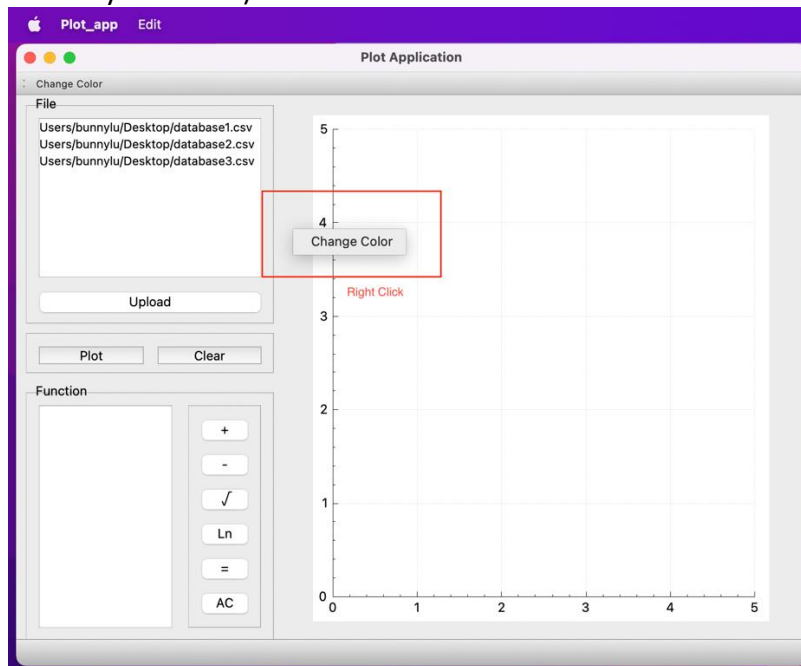
5. Change colour

Method 1: Click menu bar

Method 2: Click tool bar



Method 3: Right clicked window. (If right click the plot area or the widget area, this may not work)

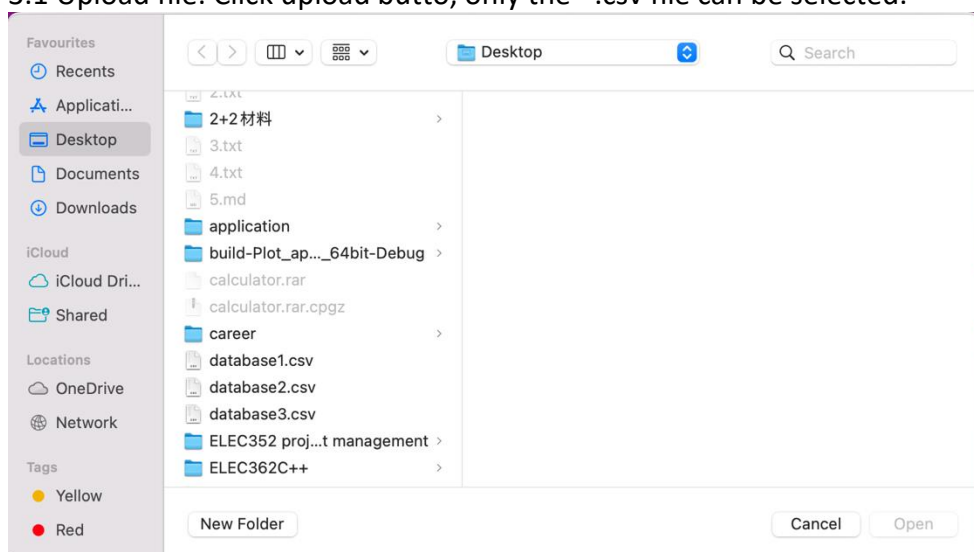


Then a dialog will appear, you should use spin box to enter the graph you want to choose. For example, you want to re-color graph1, you should enter 1 in spin box. Click the bottom Color, a color dialog will appear, and you can choose a color. Then click Ok.

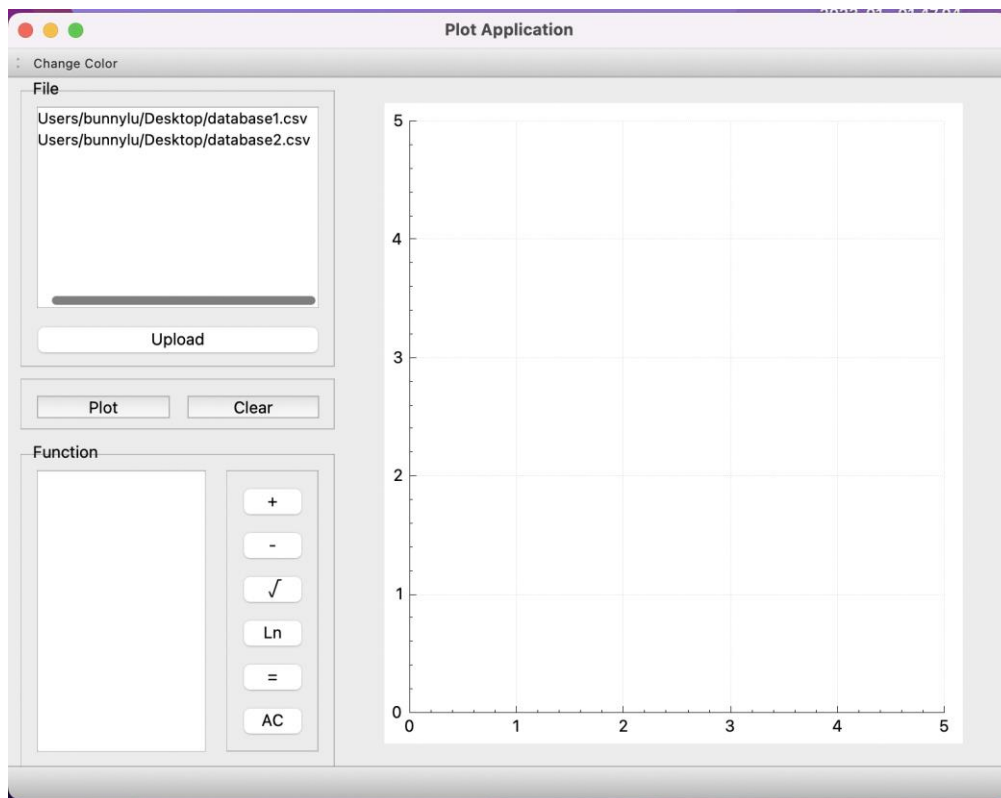
3. Testing and verification

****Important**** For simplification, all the test data has same x value. And the programme cannot deal with the situation if the data has different x value.

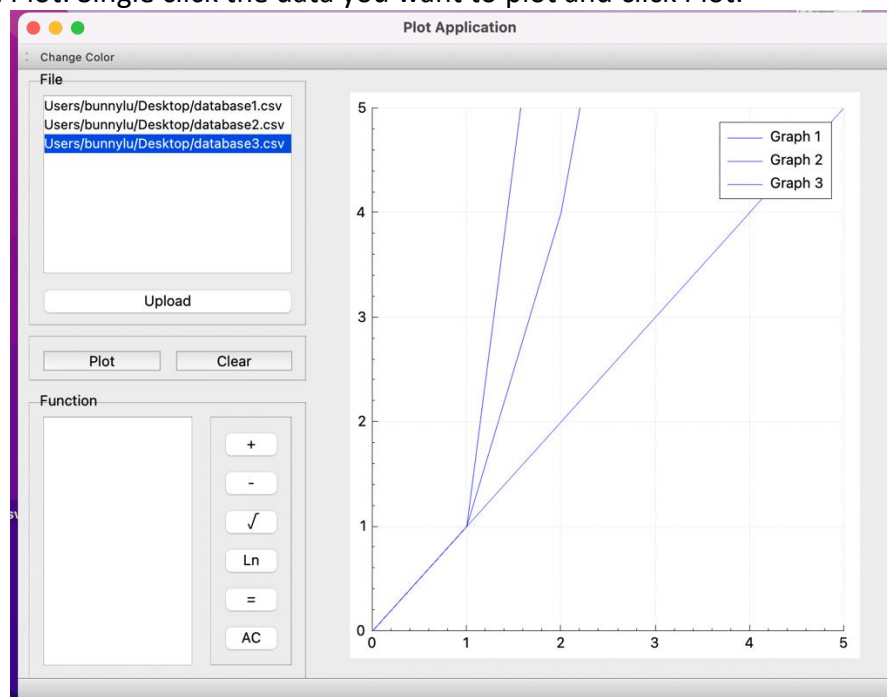
3.1 Upload file: Click upload butto, only the *.csv file can be selected.



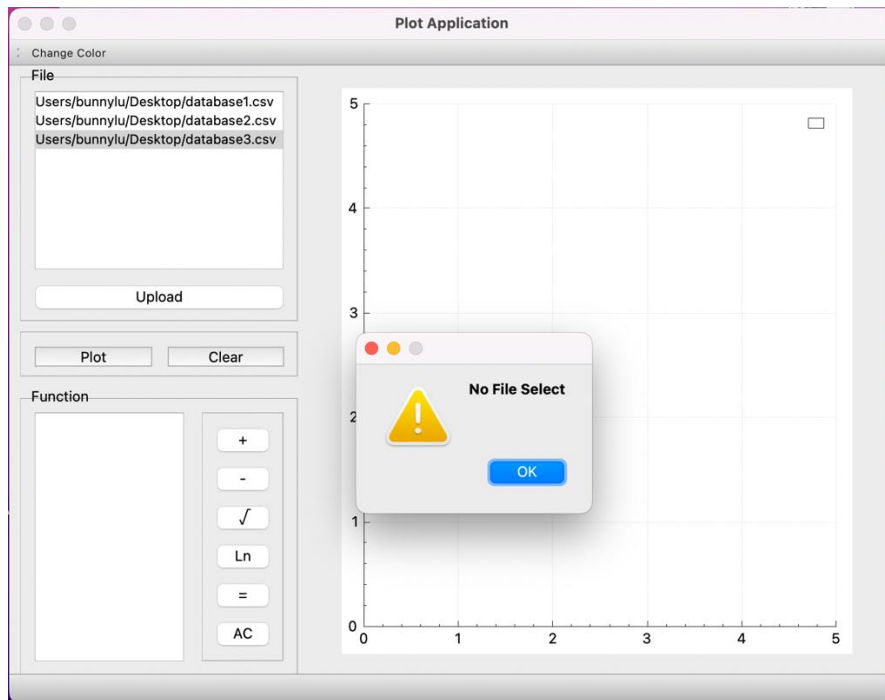
When select a file, it will show in list widget like this.



3.2 Plot: Single click the data you want to plot and click Plot.

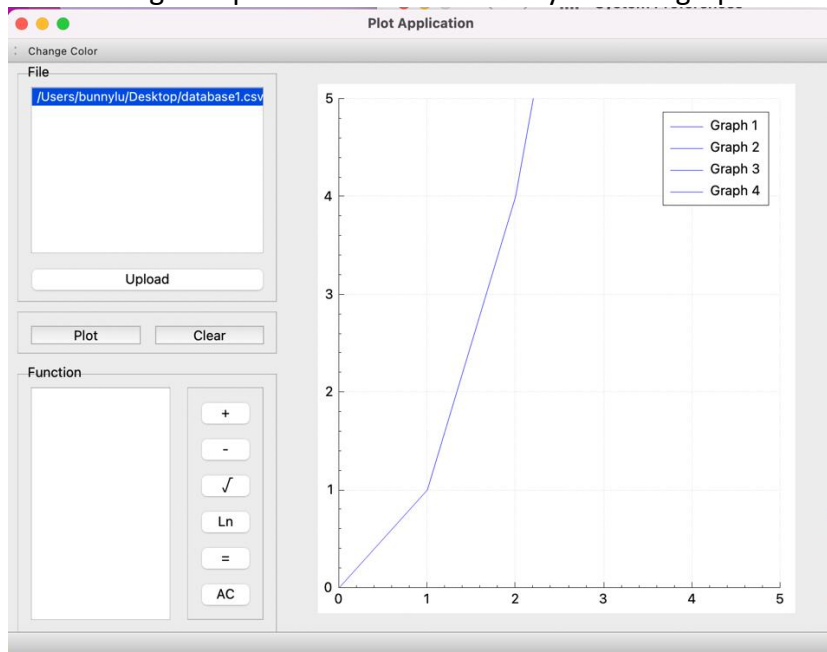


3.3 (Run-time error 1) Upload file: If Click Plot bottom without click any file, there will be something wrong and message box appear.

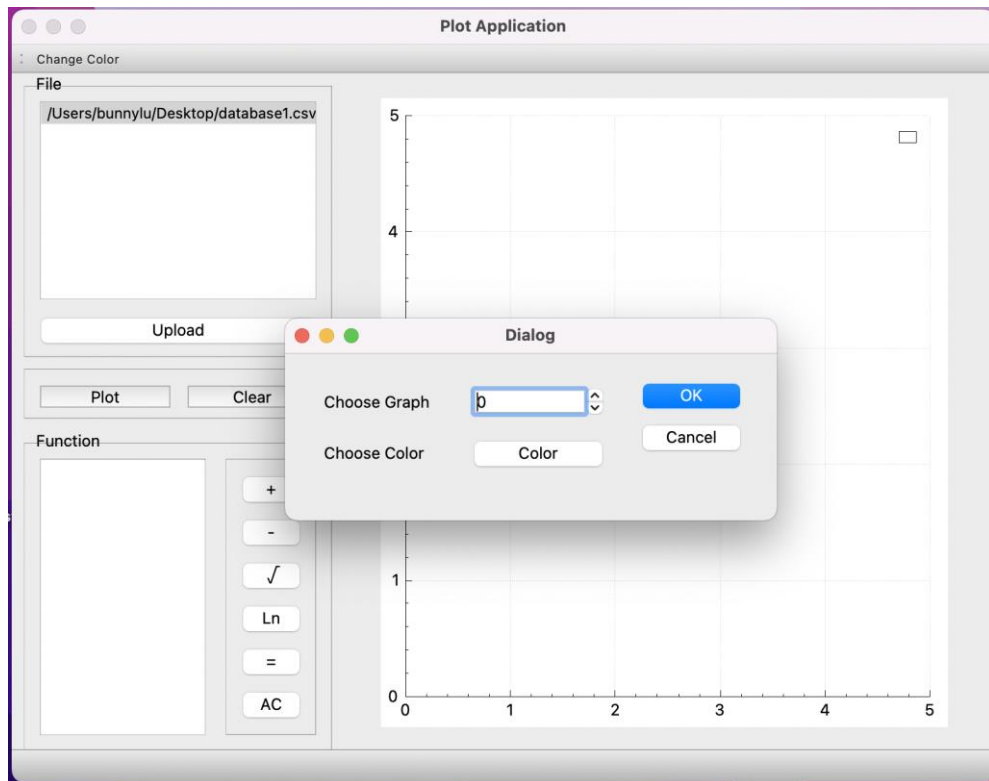


3.4 (Run-time error 2) Plot: Click one file multiple times, then click plot, the graph will only be plot once.

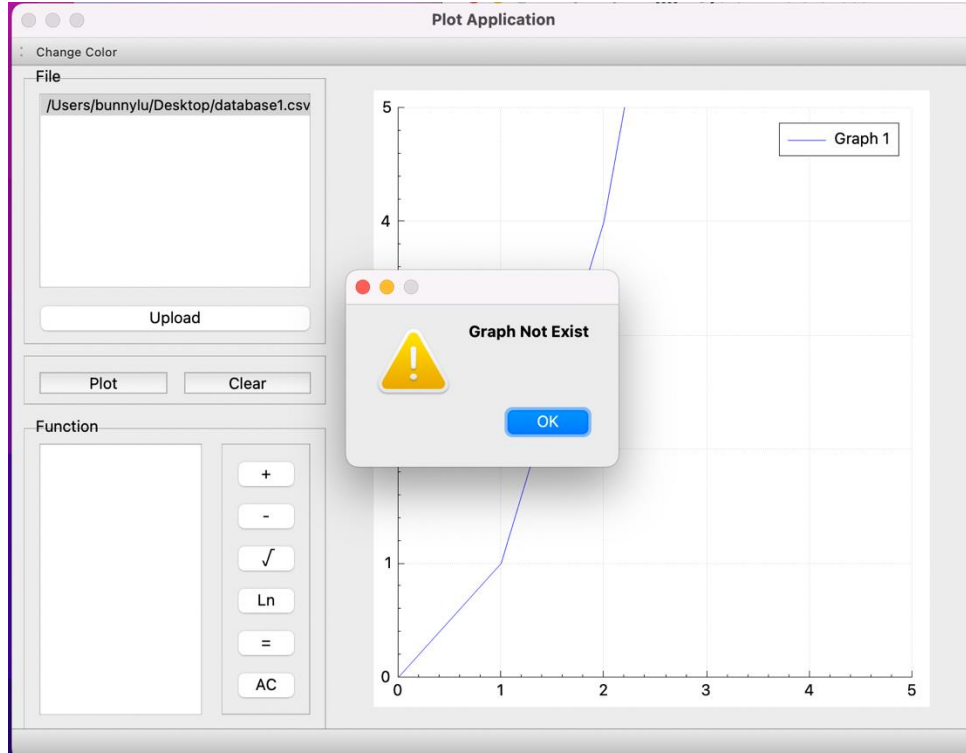
3.5 (Run time error 3) Plot: If the user clicks on File->Click Plot->Click File->Click Plot, the drawing is duplicated. This is solved by clear all graphs before plot.



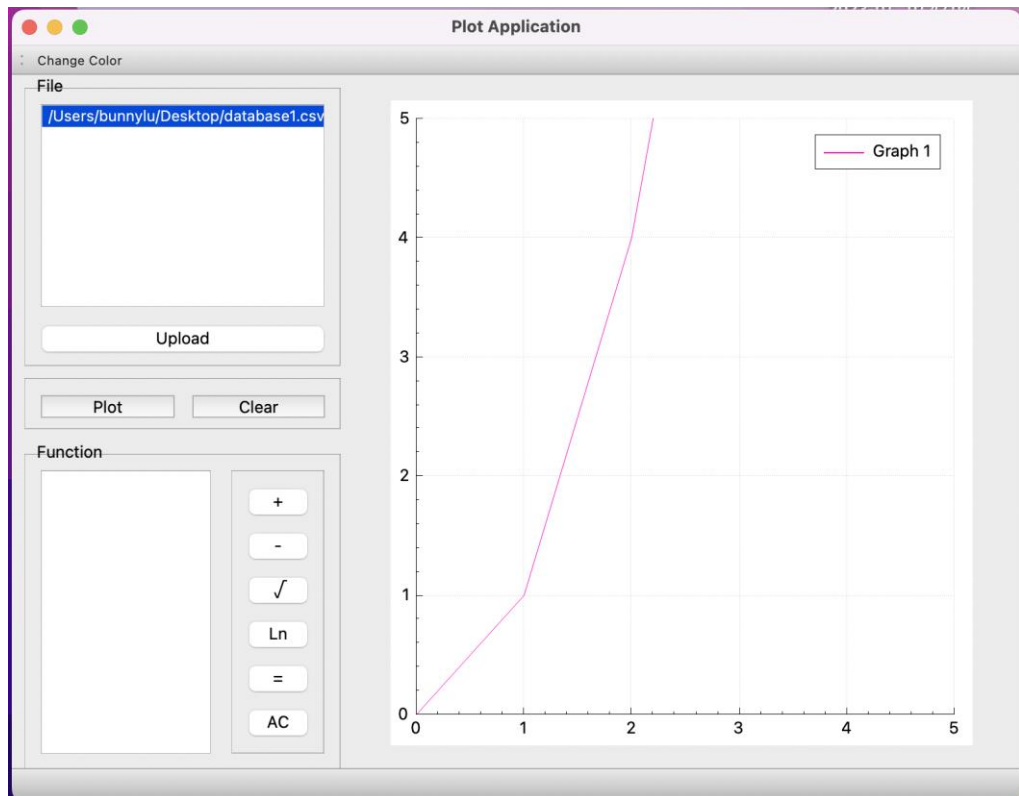
3.6 (Runtime Error 4) Colour change: When click tool bar, or menu bar, or right click mouse, the dialog will appear.



If the number in spin box is out of range, for example, there are 5 graphs, but user enter 6, or there is no graph6, there will have errors and a message box will occur.

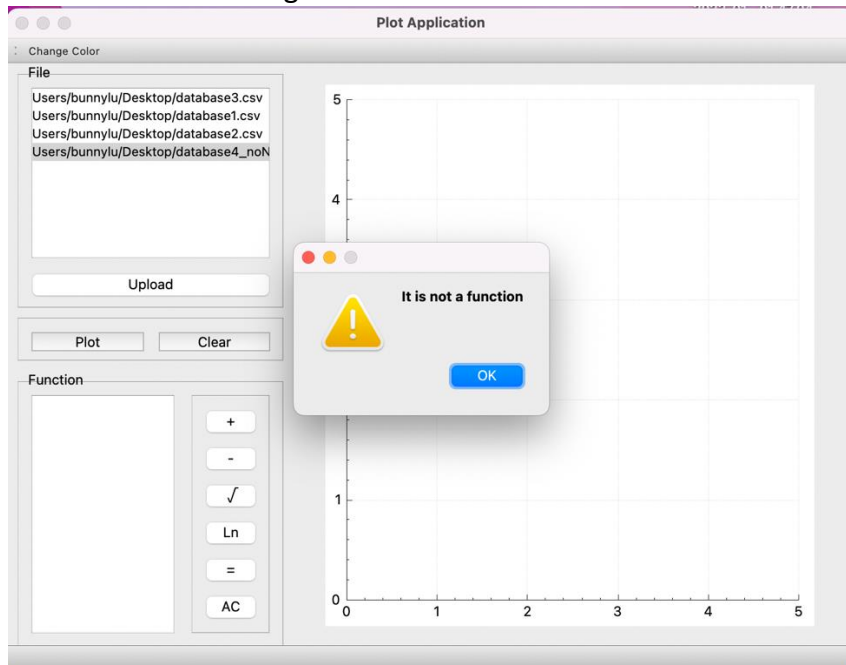


If user enter right graph number, the result will be correct.

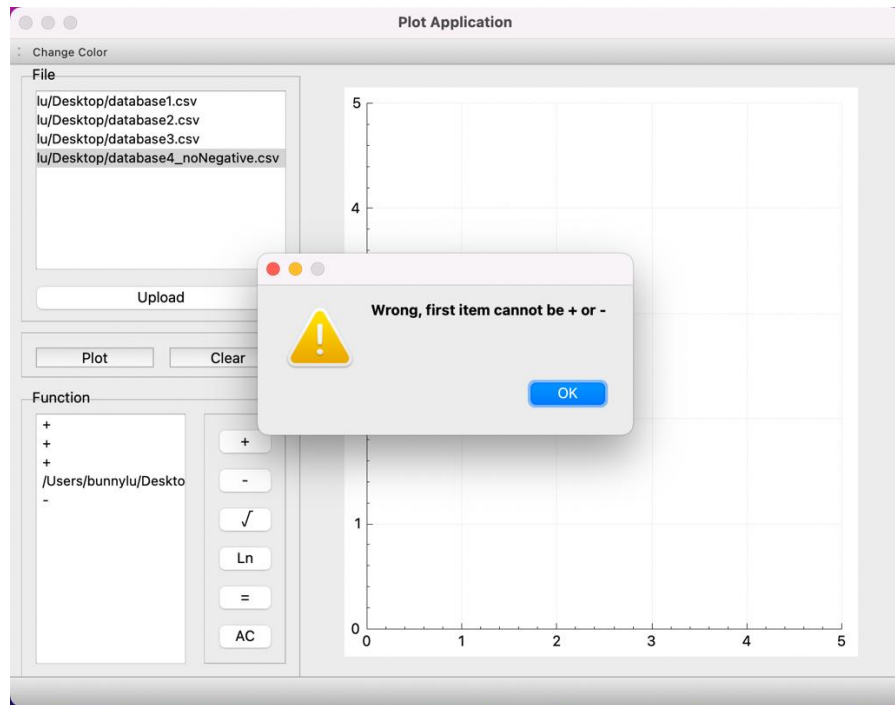


3.7 (Run-time error 5 - 14) Calculator: When using calculator, if user enter a wrong format equation, there is something wrong and a message box occur.

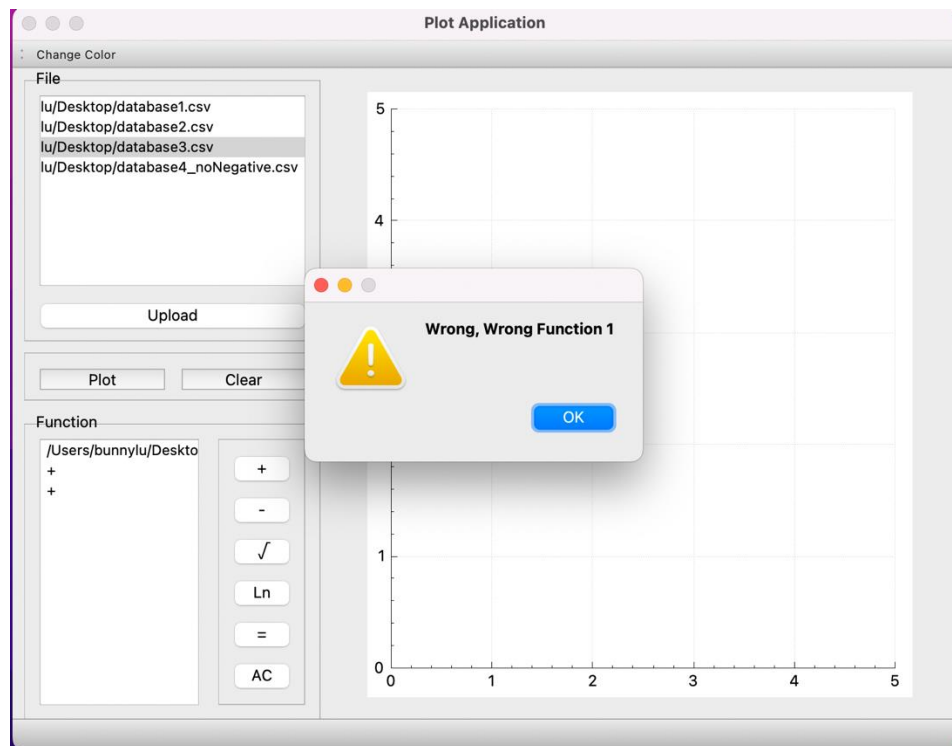
(1) When there is nothing in the calculator



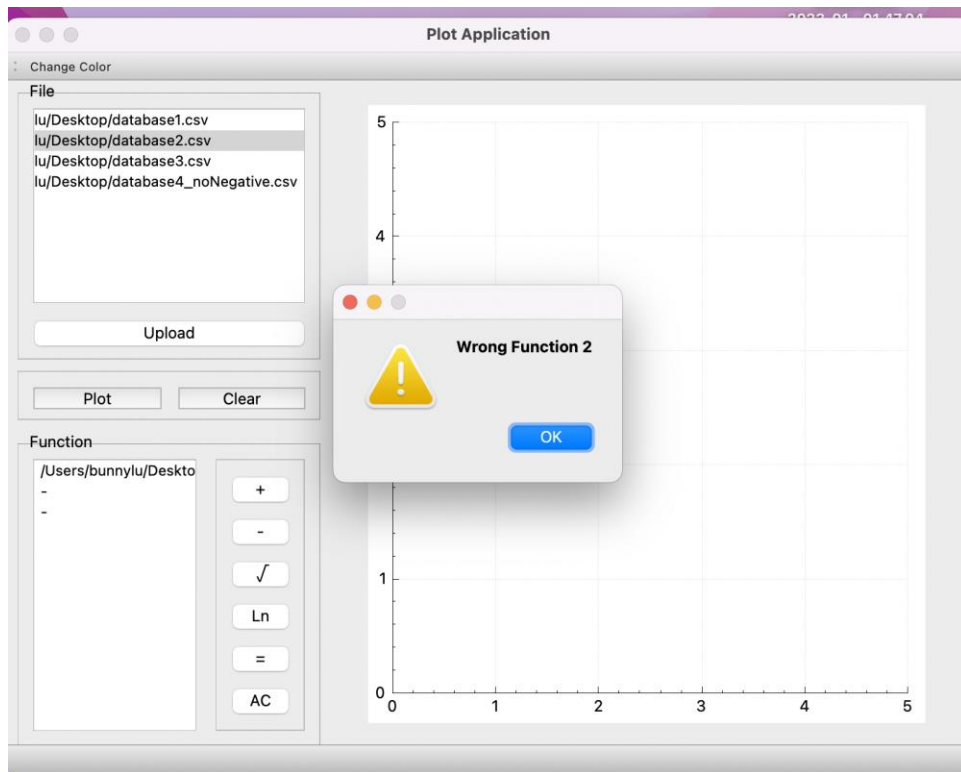
(2) When "+" or "-" is the first item



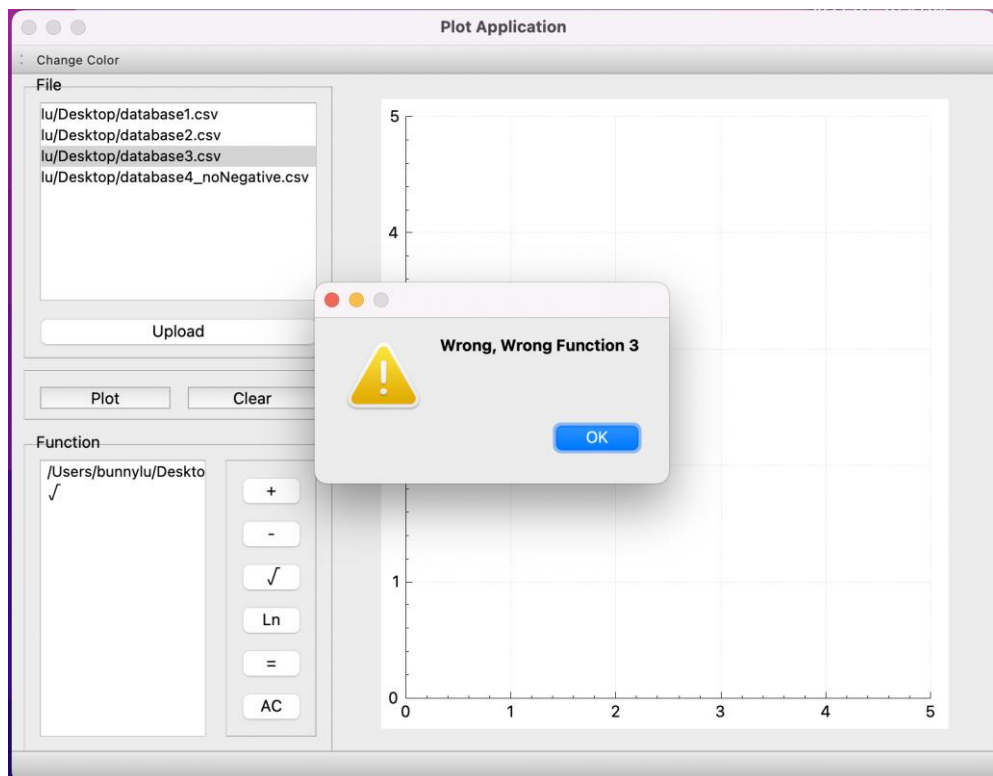
(3) When “+” is followed by a “+”, “-”, “√”, “Ln”



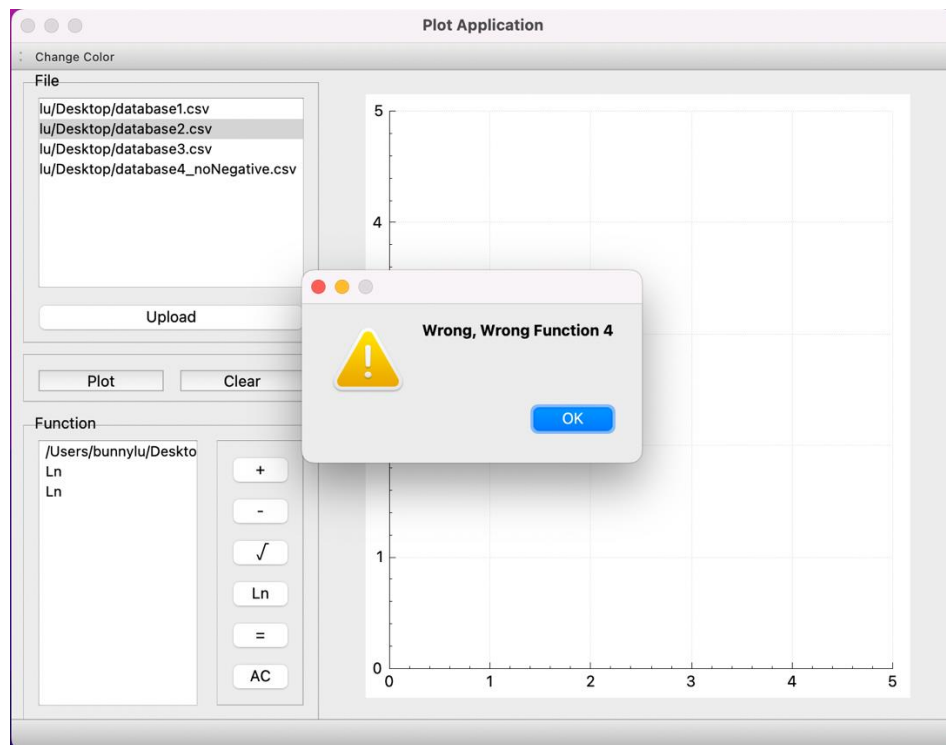
(4) When “-” is followed by a “+”, “-”, “√”, “Ln”



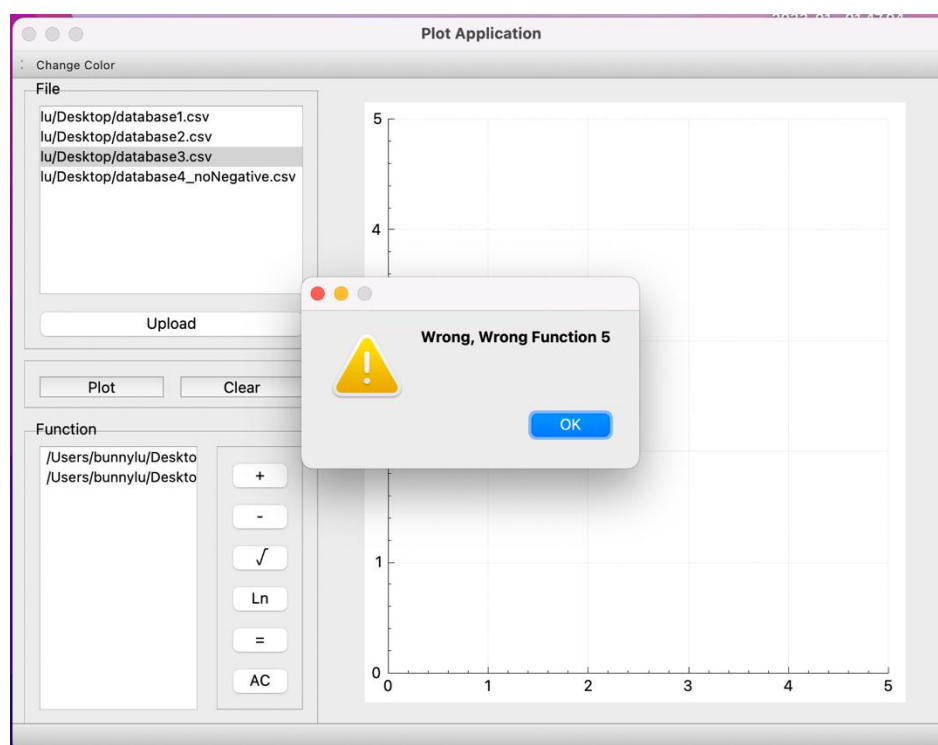
(5) When “v” is followed by a path, “v”, “Ln”



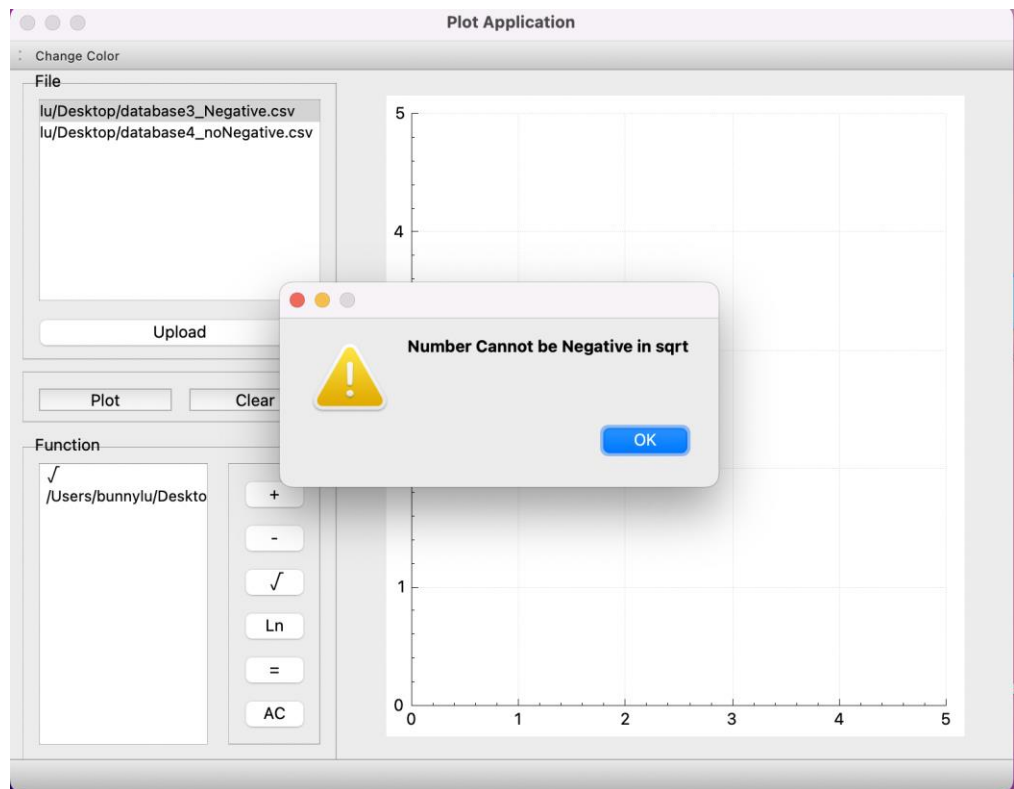
(6) When “Ln” is followed by a path, “v”, “Ln”



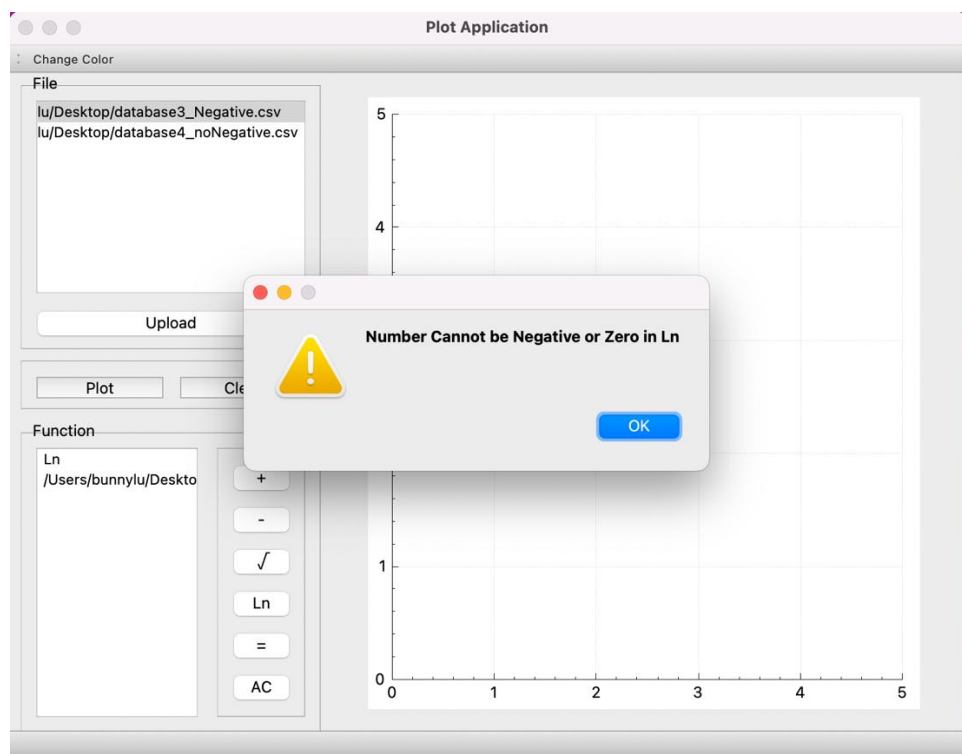
(7) When a path is followed by a path,



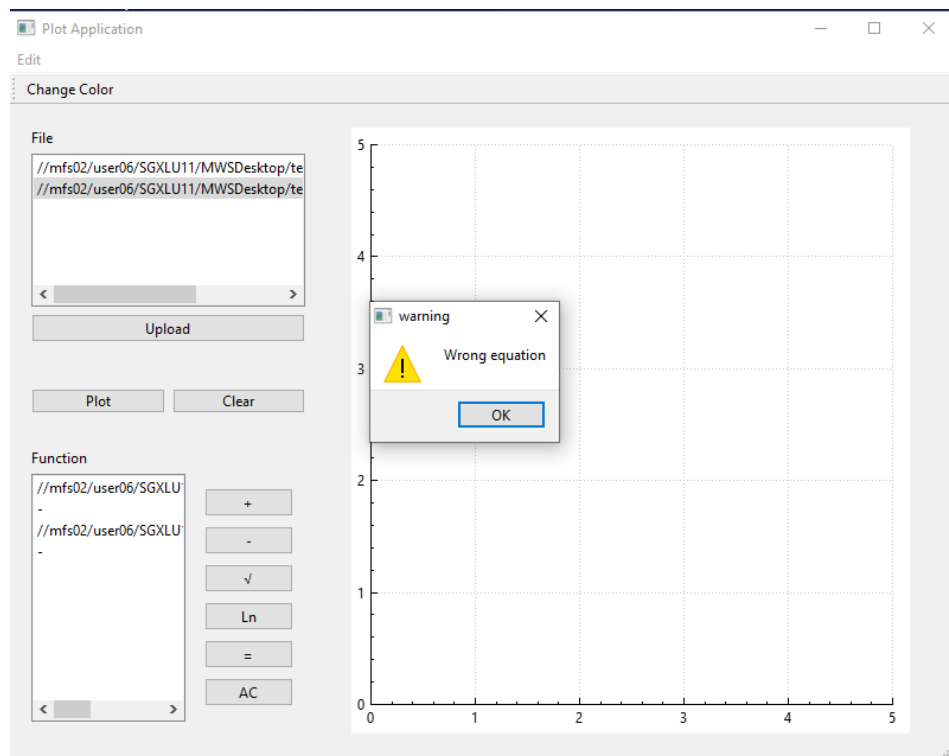
(8) When using "v" (sqrt), and the data is less than 0



(9) When using “Ln”, and the data is less than or equal to 0



(10) When entering +, -, sqrt, Ln to the last of the equation, there should be something wrong and a message box will appear.

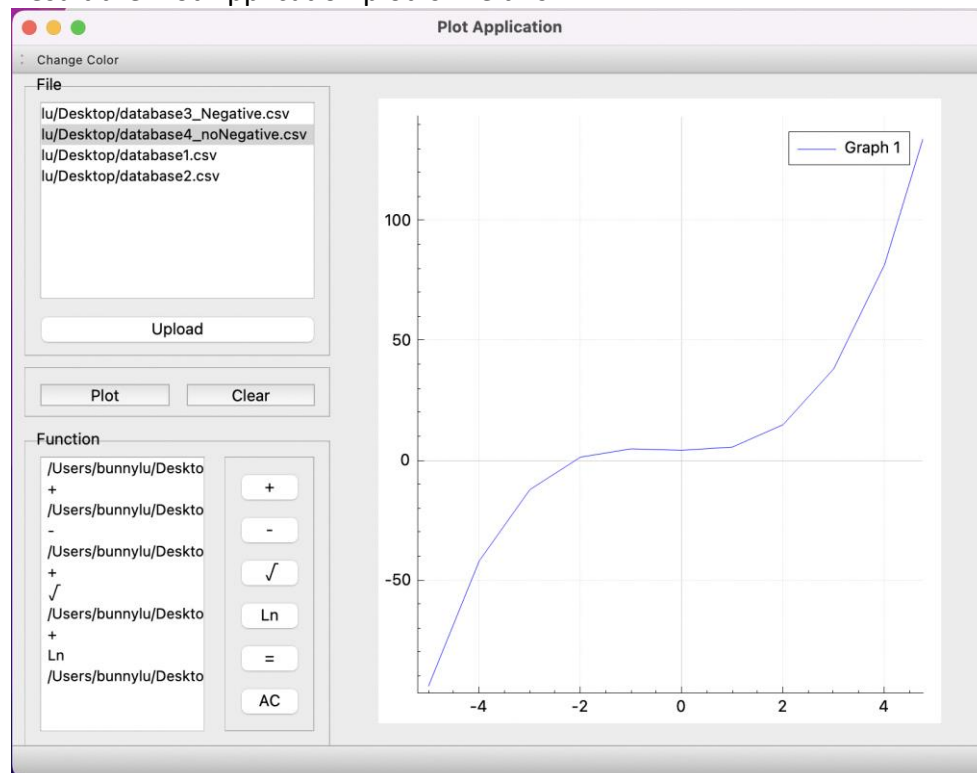


3.8 Calculator: If enter correct equation, the plot will be like this:

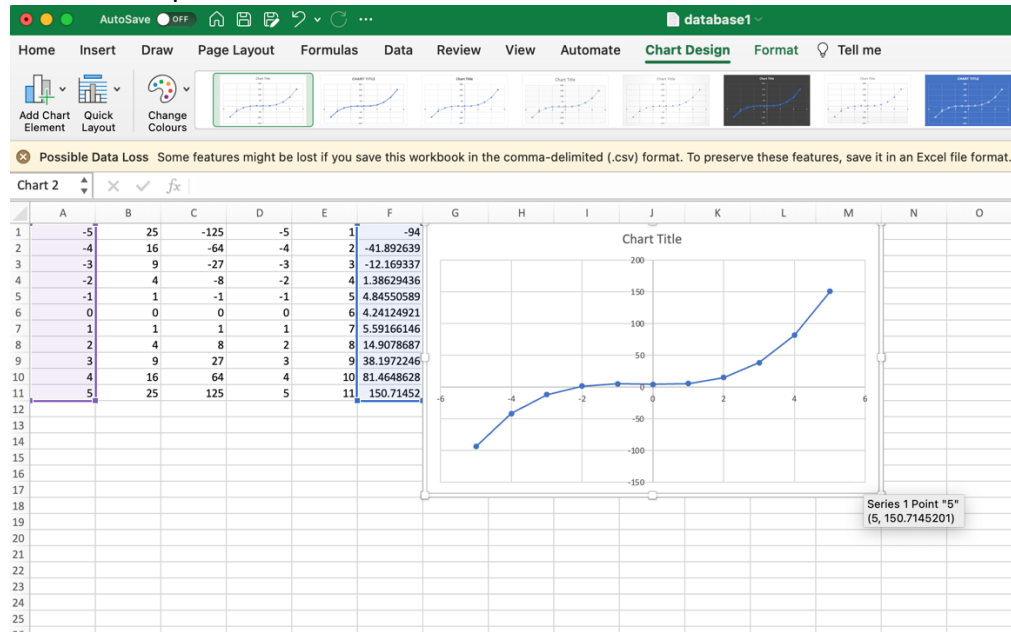
The equation is:

Database1 + Database2 – Database3_Negative + √Database4_noNegative + Ln Database4_noNegative

Result the Plot Application plot is like this:

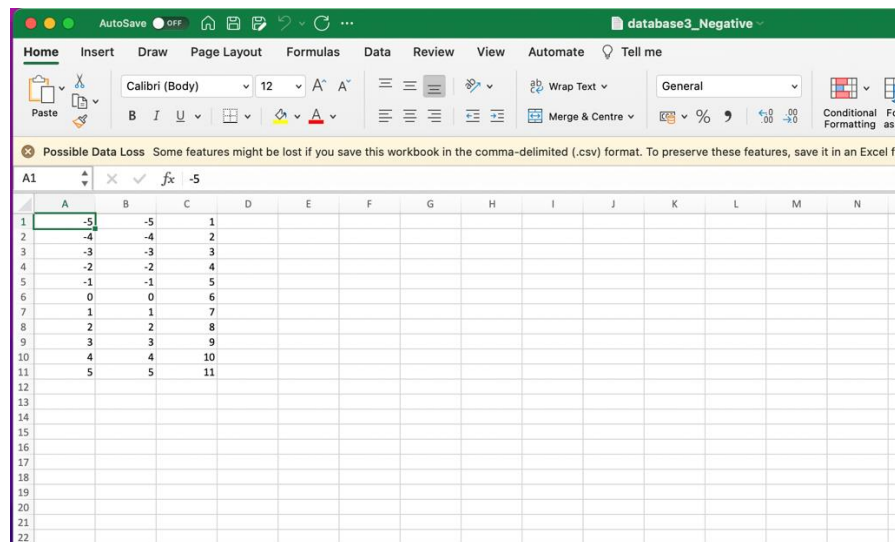


Result Excel plot is like this.



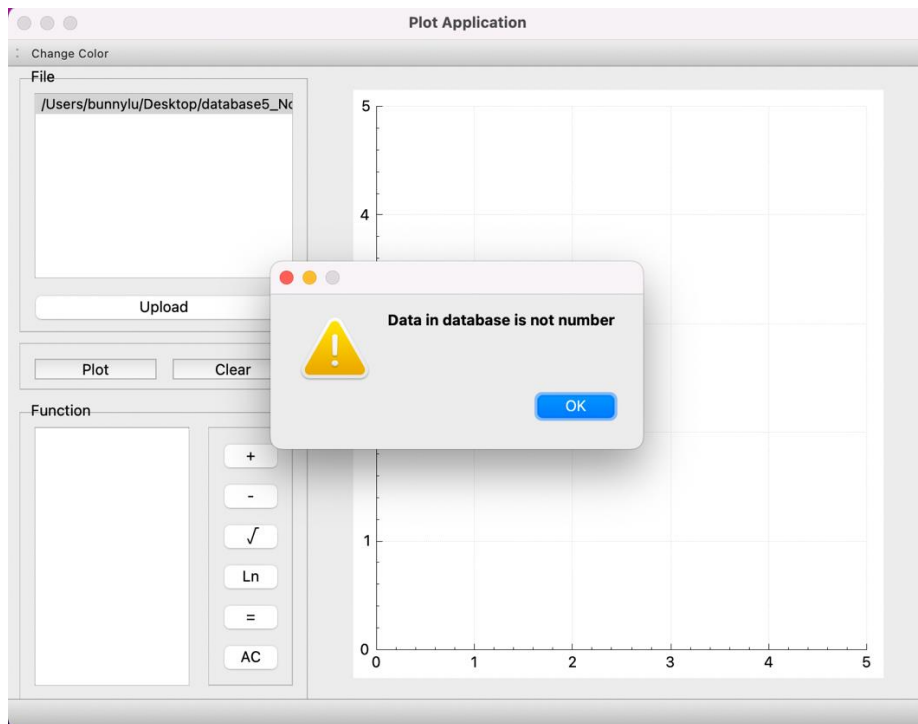
It can be concluded that the calculator function of Plot Application is correct.

3.9 (Run-time error 15) Upload file: If user upload a database which has more than 2 columns, programme will only read the first and the second column and plot. When database3 is plot which has database like this:

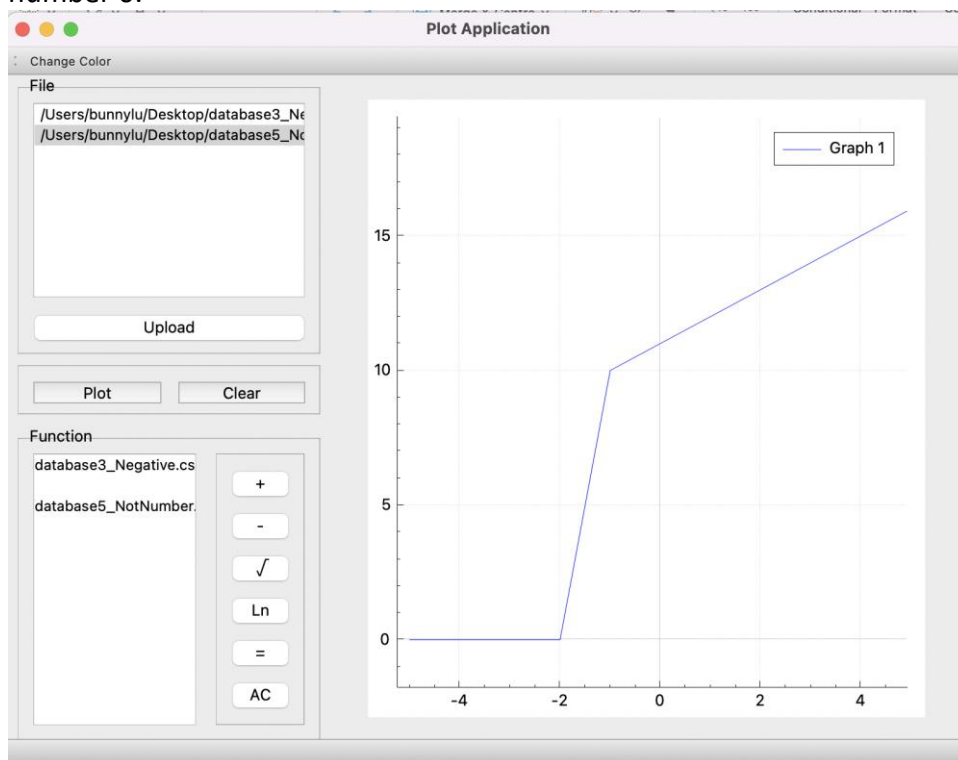


The plot will be like this. It can be seen that only column1 can column2 is plot.

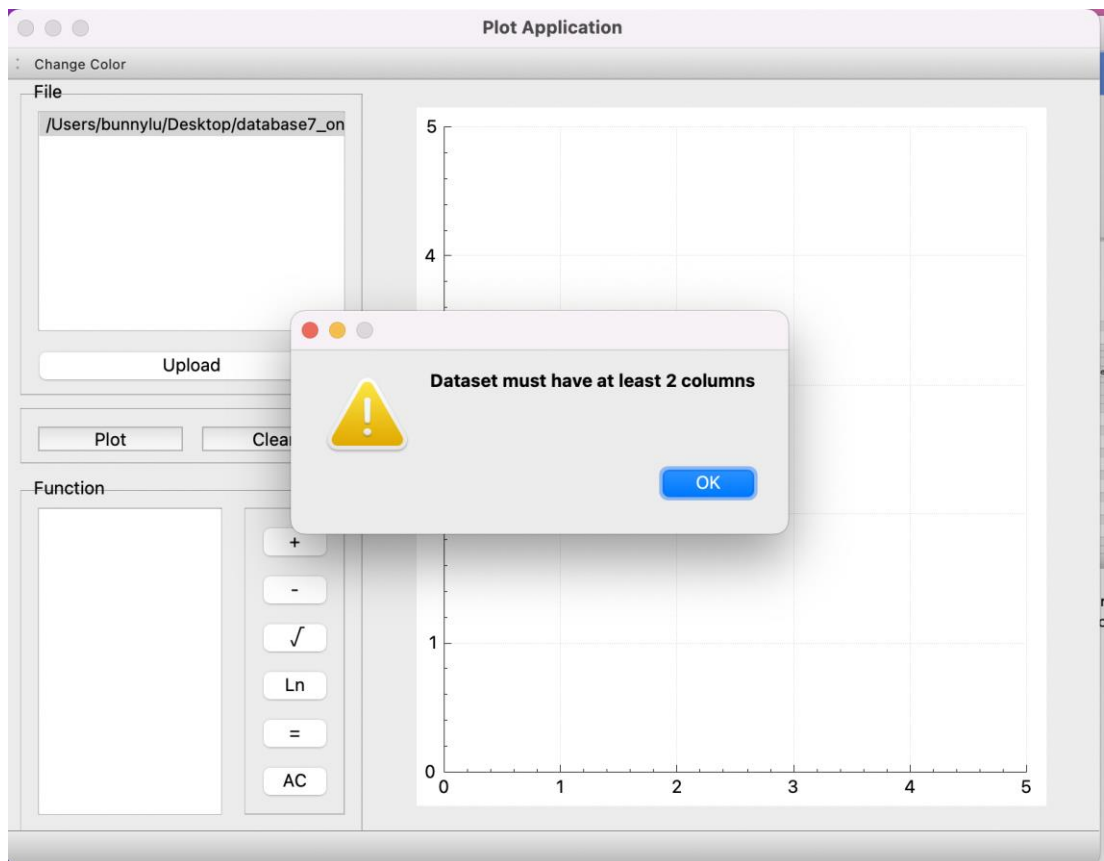
When plot, a message box will occur.



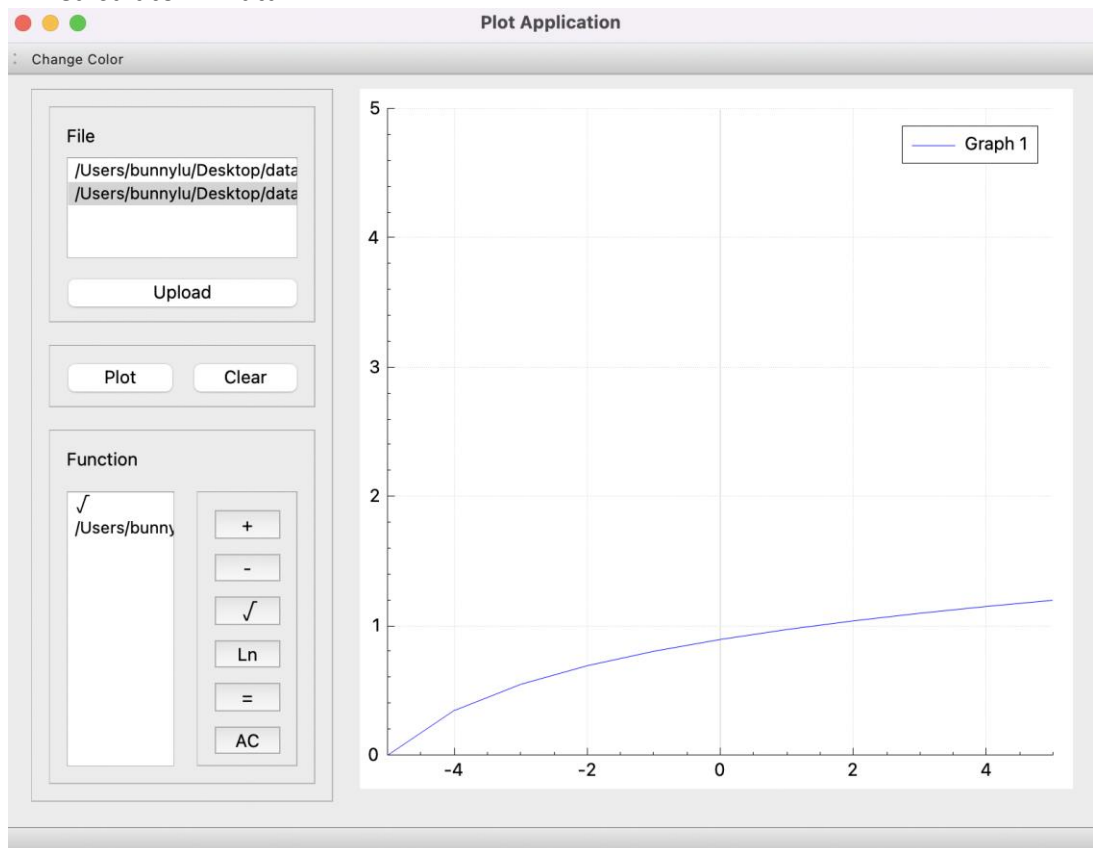
In calculator function, if the content is not number, it will be converted to number 0.



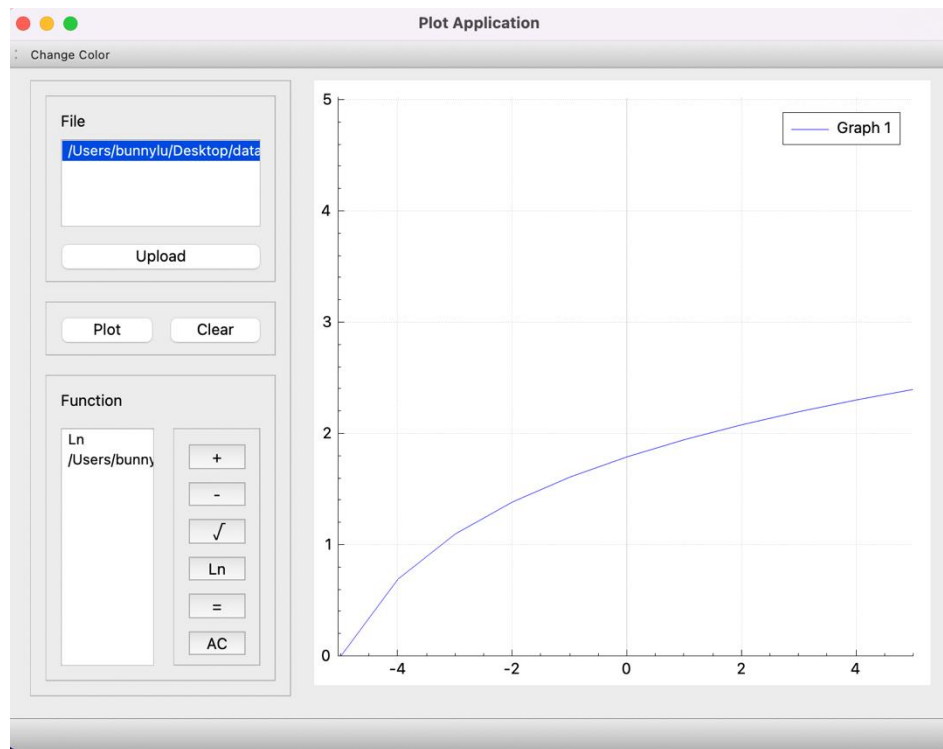
3.11 (Runtime error 17) If uploaded database has less than 2 columns, when click plot, a message box will occur.



3.12 Calculator: vData



3.13 Calculator: LnData



3.14 (Runtime error18) This error is not solved.
When upload file is null, which means there is no data in it, when Plot button clicked, the application will collapse.

3.15

4. Note:

The most important parts in the code are

a) Avoid duplications

Always clear all graphs after plotting, but do not use `qcustomplot->replot()` function. When "Plot" button clicked, the paths and graphs are cleared. When "=" button clicked, all graphs should be cleared.

b) Calculator algorithm, the use of **variable int status and int symbol**

//Uses of status:

//1. set to 1 when read as files

//2. set to 2 when read as addition or subtraction

//3. set to 3 when reading log or sqrt

//a. When reading + -, the previous one must be path, so if status=1, you can proceed. Errors are reported for 2, 3 and 0

//b. When reading log sqrt, the previous one can be addition or subtraction, not path, and not itself. So when status=2, it can proceed. When it is 1, 3 gives an error

//c. When reading path, the previous one can be additive or subtractive and can be sqrt log, but cannot be path. so status = 0, 2, 3, can proceed, 1 gives an error.

//Uses of symbol:

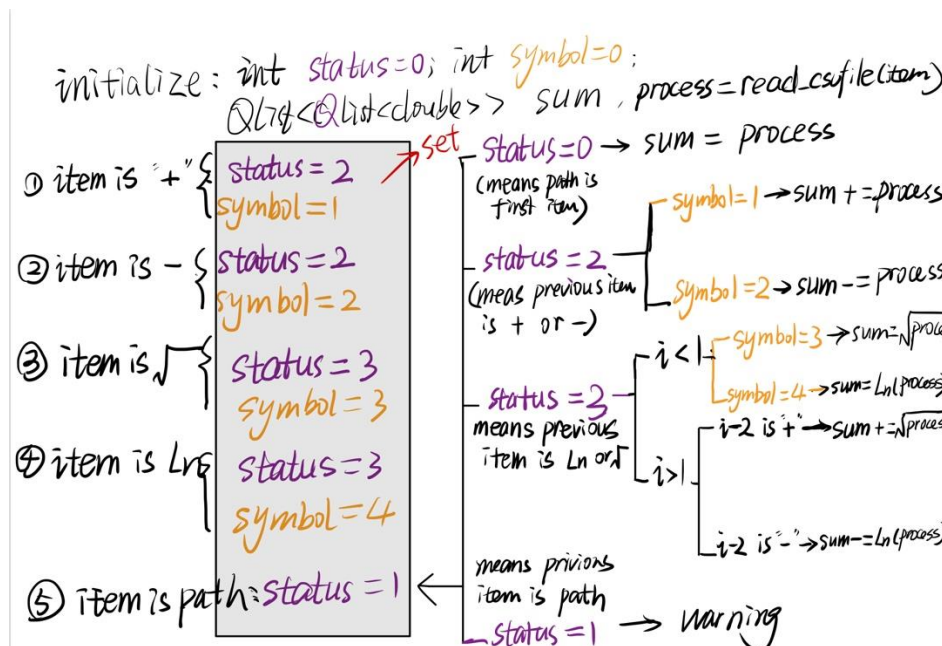
// + is 1;

// - is 2;

// $\sqrt{\quad}$ is 3;

// Ln is 4

//When the address is read, look at the symbol to select the sum and the current address for calculation.



5. Appendix

```
#include "dialog_colorchoose.h"
```

```
#include "ui_dialog_colorchoose.h"
```

```
#include <QColorDialog>
```

```
Dialog_colorChoose::Dialog_colorChoose(QWidget *parent) :
```

```
    QDialog(parent),
```

```
    ui(new Ui::Dialog_colorChoose)
```

```
{
```

```
    ui->setupUi(this);
```

```
}
```

```

Dialog_colorChoose::~~Dialog_colorChoose()
{
    delete ui;
}

void Dialog_colorChoose::on_color_clicked()
{
    QColor c = QColorDialog::getColor(Qt::white, this, "Choose Color");
    if(c.isValid()){
        col = c;
    }
}

void Dialog_colorChoose::on_Dialog_colorChoose_accepted()
{
    graph = ui->spinBox->text().toInt();
}

void Dialog_colorChoose::on_buttonBox_rejected()
{
    graph = -1;
}

#ifndef DIALOG_COLORCHOOSE_H
#define DIALOG_COLORCHOOSE_H
#include <QDialog>
#include <QColor>

namespace Ui {
class Dialog_colorChoose;
}

class Dialog_colorChoose : public QDialog
{
    Q_OBJECT

public:
    explicit Dialog_colorChoose(QWidget *parent = nullptr);

    QColor col;

```

```

    int graph = 0;
    ~Dialog_colorChoose();

private slots:
    void on_color_clicked();

    void on_Dialog_colorChoose_accepted();

    void on_buttonBox_rejected();

private:
    Ui::Dialog_colorChoose *ui;
};

#endif // DIALOG_COLORCHOOSE_H

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}

#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QByteArray>
#include <QByteArray>
#include <QByteArray>
#include <QColor>
#include <QDebug>
#include <QDialog>
#include <QFile>
#include <QFileDialog>
#include <QMenu>
#include <QtMath>
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    this->setWindowTitle("Plot Application");
}

```

```

// QString test = "4.55555555555555555555555555555555";
// double test2 = test.toDouble()
// qDebug()<< "Test 2 is " << test2;

}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButton_upload_clicked()
{
    // ui->widget->clearGraphs();
    QString path = QFileDialog::getOpenFileName(this, "Upload File", "", "*.csv" );
    if(path.isEmpty()){
        return;
    }

    ui->listWidget_file->addItem(path);
}

void MainWindow::on_listWidget_file_itemClicked(QListWidgetItem *item)
{
    paths.append(item->text());
}

void MainWindow::on_pushButton_plot_clicked()
{
    if(paths.isEmpty()){
        QMessageBox::warning(this, "No File Select", "No File Select");
        return;
    }

    plot(paths);
    // qDebug()<<paths;
    // paths.clear();
    // ui->widget->clearGraphs();
}

```

```

void MainWindow::plot(QStringList paths)
{
    ui->widget->clearGraphs();
    if(paths.isEmpty()){
        ui->widget->legend->setVisible(false);
        ui->widget->clearGraphs();
        ui->widget->replot();
    }

    // Avoid Duplication!!! Very important
    QStringList distpaths;
    for(int i = 0; i < paths.length(); i++){
        if(!distpaths.contains(paths[i])){
            distpaths.append(paths[i]);
        }
    }
    // qDebug()<<distpaths.size();

    for(int i = 0; i < distpaths.size(); i++){

        QList<QList<double>> data = read_csvFile(distpaths[i]);
        if(data.at(0).size() < 2){
            QMessageBox::warning(this, "Wrong", "Dataset must have at least 2 columns");
            return;
        }
        // qDebug()<< data;
        QVector<double> x(data.size()), y(data.size());
        // qDebug() << "2";
        // qDebug()<<data.size();
        for (int i=0; i<data.size(); ++i)
        {
            x[i] = data.at(i).at(0);
            y[i] = data.at(i).at(1);

        }

        qDebug()<<'3';
        ui->widget->legend->setVisible(true);
        ui->widget->setInteractions(QCP::iRangeDrag | QCP::iRangeZoom);
        ui->widget->addGraph();
        ui->widget->graph(i);
        ui->widget->graph(i)->addData(x,y);
        ui->widget->graph(i)->rescaleAxes(true);
        ui->widget->replot();

    }
}

```

```

}

QList<QList<double>> MainWindow::read_csvFile(QString path){
    QFile a(path);
    if(!a.exists()){
        QMessageBox::warning(this, "Wrong", "File Not Exist");
    }
    QList<QList<double>> db_list;

    QStringList lines; /*行数据*/
    QFile inFile(path);
    if (inFile.open(QIODevice::ReadOnly))
    {
        QTextStream stream_text(&inFile);
        while (!stream_text.atEnd())
        {
            lines.push_back(stream_text.readLine());
        }
        for (int j = 0; j < lines.size(); j++)
        {
            QString line = lines.at(j);
            QStringList split = line.split(","); /*列数据*/

            QList<double> db;
            for (int col = 0; col < split.size(); col++)
            {
                qDebug() << split.at(col).toDouble() << " ";
                QString temp01 = split.at(col);
                QByteArray test01 = temp01.toLatin1();
                if(test01 != "0" && temp01.toDouble() == 0){
                    QMessageBox::warning(this, "Wrong", "Data in database is not number");
                    return db_list;
                }

                db.append(temp01.toDouble());

            }
            //      qDebug()<<"/n";
            //      str_list.append(str);
            db_list.append(db);
        }
        inFile.close();
    }
}

```

```

        return db_list;
    }

void MainWindow::on_pushButton_Clear_clicked()
{
    paths.clear();
    plot(paths);
}

void MainWindow::on_actionChange_Color_triggered()
{
    Dialog_colorChoose color_dlg;
    color_dlg.exec();
    QColor colorByUser = color_dlg.col;
    int temp = color_dlg.graph;
    if(temp != -1){
        int graphByUser = color_dlg.graph-1;

        if(temp > ui->widget->graphCount() || temp <= 0){
            QMessageBox::warning(this, "Warning", "Graph Not Exist");
            return;
        }

        ui->widget->graph(graphByUser)->setPen(colorByUser);
        ui->widget->replot();
    }
}

void MainWindow::contextMenuEvent(QContextMenuEvent *event){
    QMenu contextMenu;
    contextMenu.addAction(ui->actionChange_Color);
    contextMenu.exec(event->globalPos());
}

void MainWindow::on_listWidget_file_itemDoubleClicked(QListWidgetItem *item)
{
    ui->listWidget_function->addItem(item->text());
}

void MainWindow::on_pushButton_add_clicked()
{
    ui->listWidget_function->addItem("+");
}

```

```

}

void MainWindow::on_pushButton_min_clicked()
{
    ui->listWidget_function->addItem("-");
}

```

```

void MainWindow::on_pushButton_sqrt_clicked()
{
    ui->listWidget_function->addItem("√ ");
}

```

```

void MainWindow::on_pushButton_Ln_clicked()
{
    ui->listWidget_function->addItem("Ln");
}

```

```

void MainWindow::on_pushButton_eq_clicked()
{
    QList<QList<double>> sum;

    QList<QList<double>> process;

```

```

//Uses of status:
//1. set to 1 when read as files
//2. set to 2 when read as addition or subtraction
//3. set to 3 when reading log or sqrt

```

//a. When reading + -, the previous one must be path, so if status=1, you can proceed.
Errors are reported for 2, 3 and 0

//b. When reading log sqrt, the previous one can be addition or subtraction, not path, and not itself. So when status=2, it can proceed. When it is 1, 3 gives an error

//c. When reading path, the previous one can be additive or subtractive and can be sqrt log, but cannot be path. so status = 0, 2, 3, can proceed, 1 gives an error.
int status = 0;

```

//Uses of symbol:
// + is 1;
// - is 2;
// √ is 3;
// Ln is 4

```

//When the address is read, look at the symbol to select the sum and the current address for calculation.


```
int symbol = 0;
qDebug() << "3. There are " << ui->widget->graphCount() << "graphs in the plot before plotting";
```

```
//When only 1 or none database added to the function
if(ui->listWidget_function->count() == 0 || ui->listWidget_function->count() == 1){
    QMessageBox::warning(this, "wrong", "It is not a function" );
    on_pushButton_AC_clicked();
    return;
}
```

```
qDebug() << "2. There are " << ui->widget->graphCount() << "graphs in the plot before plotting";
```

```
//When + or - comes first in the function
QByteArray temp = ui->listWidget_function->item(0)->text().toLatin1();
if(temp == "+" || temp == "-"){
    QMessageBox::warning(this, "wrong", "Wrong, first item cannot be + or -" );
    on_pushButton_AC_clicked();
    return;
}
```

```
//read each item in the listwidget_function
for(int i = 0; i < ui->listWidget_function->count(); i++){
    QString item = ui->listWidget_function->item(i)->text();
    if(item == "+"){
        qDebug() << "+";
        if(status == 0 || status == 2 || status == 3){
            QMessageBox::warning(this, "wrong", "Wrong, Wrong Function 1" );
            on_pushButton_AC_clicked();
            return;
        }
        status = 2;
        symbol = 1;
    }
```

```
}else if(item == "-"){
    qDebug() << "-";
    if(status == 0 || status == 2 || status == 3){
        QMessageBox::warning(this, "wrong", "Wrong Function 2" );
        on_pushButton_AC_clicked();
        return;
    }
    status = 2;
    symbol = 2;
}else if(item == "√"){
    qDebug() << "√ ";
    if(status == 1 || status == 3){
```

```

        QMessageBox::warning(this, "wrong", "Wrong, Wrong Function 3" );
        on_pushButton_AC_clicked();
        return;
    }
    status = 3;
    symbol = 3;
} else if(item == "Ln"){
//    qDebug() << "log";
    if(status == 1 || status == 3){
        QMessageBox::warning(this, "wrong", "Wrong, Wrong Function 4" );
        on_pushButton_AC_clicked();
        return;
    }
    status = 3;
    symbol = 4;
} else{ //In this case, the item is a Path
    qDebug() << "This is a path";
    if(status == 1){
//        qDebug() << "This is path";
        QMessageBox::warning(this, "wrong", "Wrong, Wrong Function 5" );
        on_pushButton_AC_clicked();
        return;
    } else if(status == 0){
        sum = read_csvFile(item);
    } else if(status == 2){ // Symbo is - or +
        process = read_csvFile(item);
        switch(symbol)
        {
            case 1:
                for(int i = 0; i < process.size(); i++){
                    sum[i][0] = process[i][0];
                    sum[i][1] += process[i][1];
                }
                break;
            case 2:
                for(int i = 0; i < process.size(); i++){
                    sum[i][0] = process[i][0];
                    sum[i][1] = sum[i][1]-process[i][1];
                }
                break;

        }
    }
} else if(status == 3){ //Symbol is sqrt or ln; y cannot be negative or zero
    process = read_csvFile(item);
    switch(symbol)
    {
        case 3: //sqrt

```

```

    for(int i = 0; i < process.size(); i++){
        if(process[i][1]<0){
            QMessageBox::warning(this, "Wrong", "Number Cannot be Negative in
sqrt");
            on_pushButton_AC_clicked();
            return;
        }
        process[i][0] = process[i][0];
        process[i][1] = qSqrt(process[i][1]);
    }
    break;
case 4: //Ln

    for(int i = 0; i < process.size(); i++){
        if(process[i][1]<=0){
            QMessageBox::warning(this, "Wrong", "Number Cannot be Negative or Zero
in Ln");
            on_pushButton_AC_clicked();
            return;
        }
        process[i][0] = process[i][0];
        process[i][1] = qLn(process[i][1]);
    }
    break;
}

if(i < 2){ //Means sqrt or Ln is the first symbol
    sum = process;
}
if(i >= 2){ //Means sqrt or Ln is not first symbol
    QByteArray temp2 = ui->listWidget_function->item(i-2)->text().toLatin1();
    if(temp2 == "+"){
        for(int i = 0; i < process.size(); i++){
            sum[i][0] = process[i][0];
            sum[i][1] = sum[i][1] + process[i][1];
        }
    }
    if(temp2 == "-"){
        for(int i = 0; i < process.size(); i++){
            sum[i][0] = process[i][0];
            sum[i][1] = sum[i][1] - process[i][1];
        }
    }
}
}

//    qDebug()<< "TEST TEST TEST";

```

```

        status = 1;

    }
}

//After finish the calculating, the final item in the equation should not be + - sqrt Ln
//which means the status should not equal to 2, 3
if(status == 2 || status == 3){
    QMessageBox::warning(this, "warning", "Wrong equation");
    return;
}
qDebug() << "1,There are " << ui->widget->graphCount() << "graphs in the plot before
plotting";

QVector<double> x(sum.size()), y(sum.size());
for (int j=0; j<sum.size(); j++)
{
    x[j] = sum.at(j).at(0);
    y[j] = sum.at(j).at(1);
}

qDebug() << "There are " << ui->widget->graphCount() << "graphs in the plot before
plotting";
//If there is no graph
if(ui->widget->graphCount() == 0){
    ui->widget->legend->setVisible(true);
    ui->widget->setInteractions(QCP::iRangeDrag | QCP::iRangeZoom);
    ui->widget->addGraph();
    ui->widget->graph(0);
    ui->widget->graph(0)->addData(x,y);
    ui->widget->graph(0)->rescaleAxes();
    ui->widget->replot();
}else{//When there are graphs
    ui->widget->addGraph();
    ui->widget->setInteractions(QCP::iRangeDrag | QCP::iRangeZoom);
    ui->widget->graph(ui->widget->graphCount()-1);
    ui->widget->graph(ui->widget->graphCount()-1)->addData(x,y);
    ui->widget->graph(ui->widget->graphCount()-1)->rescaleAxes(true);
    ui->widget->replot();
}
qDebug() << "There are " << ui->widget->graphCount() << "graphs in the plot after
plotting";
}

```

```

void MainWindow::on_pushButton_AC_clicked()
{
    ui->listWidget_function->clear();
}

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QListWidget>
#include <QMainWindow>
#include <QContextMenuEvent>
#include <QColor>
#include "dialog_colorchoose.h"
#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);

    QStringList paths;

    QColor m_color;

    QList<QList<double> > read_csvFile(QString path);

    ~MainWindow();

    void plot(QStringList paths);
private slots:
    void on_pushButton_upload_clicked();

    void on_listWidget_file_itemClicked(QListWidgetItem *item);

    void on_pushButton_plot_clicked();

    void on_pushButton_Clear_clicked();

    void on_actionChange_Color_triggered();

```

```
void contextMenuEvent(QContextMenuEvent *event);

void on_listWidget_file_itemDoubleClicked(QListWidgetItem *item);

void on_pushButton_add_clicked();

void on_pushButton_min_clicked();

void on_pushButton_sqrt_clicked();

void on_pushButton_Ln_clicked();

void on_pushButton_eq_clicked();

void on_pushButton_AC_clicked();

private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```