

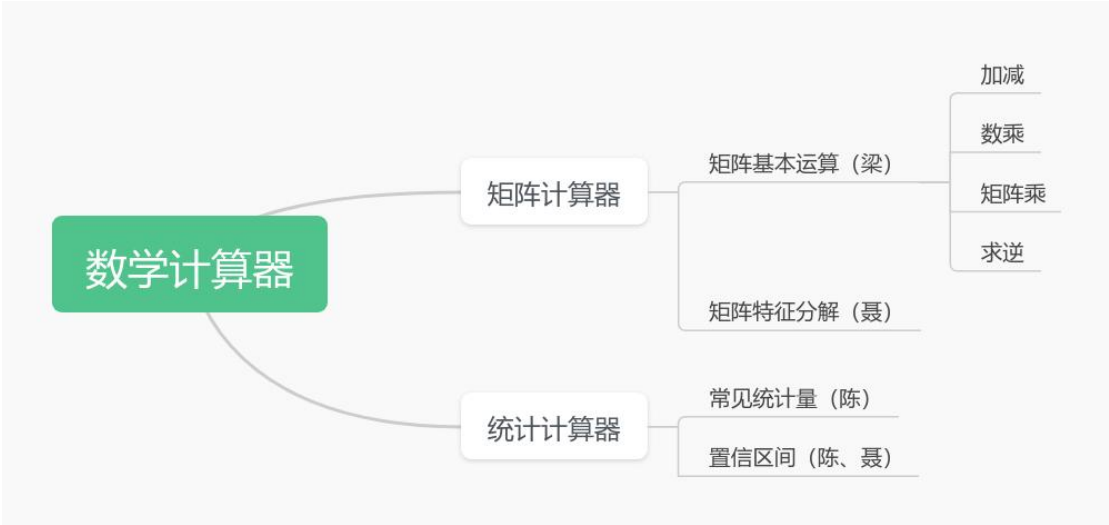
QT 大作业报告

——31 组星辰锦梁队

（1）程序功能介绍

项目名称——基于 QT 的数学计算器

项目功能：



特别地，

在“矩阵基本运算”中，我们小组实现了对矩阵的存储，以便进行多次运算；

在“统计计算器”中，我们小组实现了对.txt 文档的读入，以便进行大样本的统计计算。

（2）项目各模块与类设计细节

1. 矩阵基本运算：

A. 指针

所有的矩阵均以指针方式存储，只发送存储地址，主窗口与各运算子窗口共享相同的数个矩阵。在运算窗口进行保存，会直接修改 A-F 矩阵，省去矩阵的传输，提升效率；也避免 A-F 矩阵同时出现多个值，简化系统。

B. 多态

```
struct st_base_mat_wid: public QDialog {
    Q_OBJECT
public:
    QVector<st_matrix*> & mat_vec;
    QVector<QPushButton*> mat_bt_vec;
    QLabel * what_to_do_label;
    st_matrix * ans_mat;
    QVector<QLabel *> ans_display_vec;
    QPushButton * save;
    QPushButton * cancel;
    QPushButton * conf_bt;

    void display_ans_mat();
    explicit st_base_mat_wid(QVector<st_matrix*> & m, mainmatrix * p);
    ~st_base_mat_wid() {if (ans_mat) delete ans_mat;};

protected slots:
    virtual void save_press_down();
    virtual void save_mat_pressed();
    void mat_wid_cancel() {this -> close();};
};
```

使用了 st_base_mat_wid 作为各运算子窗口的基类。

mat_vec	A-F 矩阵的指针
mat_bt_vec	六个需要显示的矩阵按钮
display_ans_mat	显示计算结果
save_press_down	进入存储模式
save_mat_pressed	在存储模式下，保存至该矩阵(A-F)

各类运算的子窗口均继承自此类，精简代码。

2. 矩阵特征分解

设计了 eigendecomposition（特征分解）类：

```
class eigendecomposition : public QDialog
{
    Q_OBJECT

public:
    explicit eigendecomposition(QWidget *parent = nullptr);
    ~eigendecomposition();

private slots:
    void calculateEigen();

private:
    Ui::eigendecomposition *ui;

    int matrixSize;
    QLineEdit *sizeInput;
    QTextEdit *resultsTextEdit;
    QVector<QVector<QLineEdit*>> matrixInputs;
    QPushButton *calculateButton;

    Eigen::MatrixXd matrix;
    Eigen::SelfAdjointEigenSolver<Eigen::MatrixXd> solver;
};
```

主要包括了两个函数：构造函数、calculateEigen 槽函数；

前者主要用于数据的输入；

而后者通过使用第三方库 Eigen，完成了特征分解。

3. 常见统计量、置信区间的计算

设计了 largesample 类：

```
class largesample : public QDialog
{
    Q_OBJECT

public:
    explicit largesample(QWidget *parent = nullptr);
    ~largesample();

private slots:
    void on_openButton_clicked();
    void on_newButton_clicked();
    void on_jisuanButton_clicked();

private:
    Ui::largesample *ui;
    QLabel *resultLabel;

    double calculateMean(const QVector<double> &samples);
    double calculateVariance(const QVector<double> &samples);
    double calculateStdDeviation(const QVector<double> &samples);
    double calculateMedian(const QVector<double> &samples);
    double calculateMin(const QVector<double> &samples);
    double calculateMax(const QVector<double> &samples);
};
```

主要包括了两个函数：on_openButton_clicked 槽函数、on_jisuanButton_clicked 槽函数

前者用于打开并读取.txt 文件中的数据；

而后者通过调用六个用于计算统计量的普通函数，完成了常见统计量、置信区间的计算。

（3）小组成员分工情况（可参考（1）中项目功能的思维导图）

梁天睿：实现了矩阵计算器的大部分功能——包括所有的矩阵基本运算；

陈锦泓：实现了统计计算器的大部分功能——包括常见统计量、置信区间的计算；

聂延辰：实现了上述之外的小部分功能——包括矩阵特征分解、置信区间的计算，并进行了功能测试与界面设计优化。

（4）项目总结与反思

1. 总结

“实用性”是本项目的核心出发点也是本项目的最大特色。例如矩阵分解有很多种，本项目选择实现了最常见且最重要的一种——特征分解；又如置信区间的构造有很多种，本项

目选择实现了最常见且最重要的三种——正态总体下的均值（方差已知/未知）以及方差的置信区间构造。

此外，本项目还有两大“创新点”——一是实现了对矩阵的存储，以便进行多次运算（组长本人之前用过网上的几种矩阵计算器，它们都没有存储矩阵的功能。如果需要进行多次的矩阵运算，就需要自己把运算的中间结果记录下来，十分麻烦）；二是实现了大/小样本数据的“同时”读入，即共用同一个输入框（这样可以随时切换大样本/小样本输入，甚至可以两者混用——如从文档中读入大样本数据之后，再手动输入一些小样本数据作为补充）。这两大“创新点”也增强了本项目的“实用性”。

2. 反思

受限于时间，本项目最终没有完全实现项目初期设计文档的全部内容，而是选择性地实现了一些最实用的功能。如果有机会的话，我们小组还希望实现一些其他较为实用的功能，如矩阵的奇异值分解、统计直方图的构造等。

（5）补充

受限于时长限制，本项目实际运行的录屏没有完全展示出已实现的所有功能。

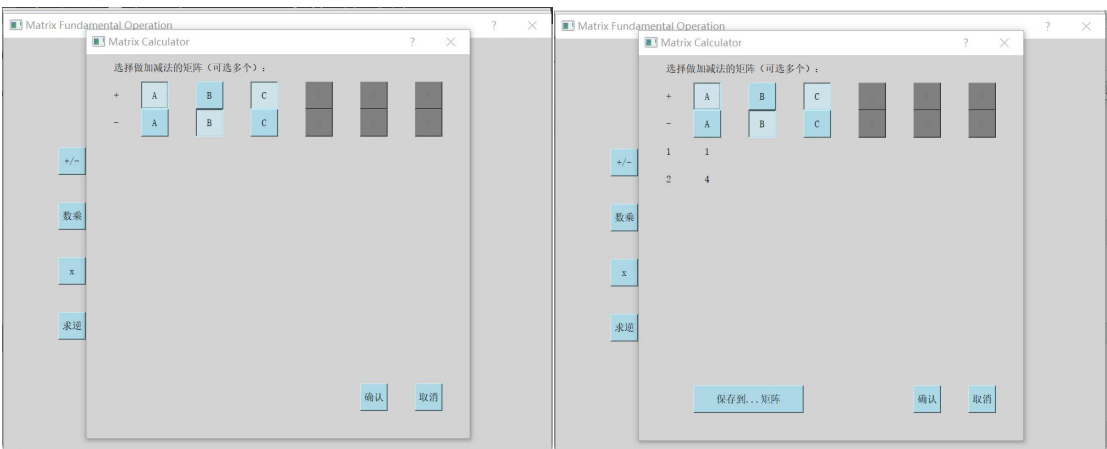
因此，这里提供几张未在录屏中展示的功能截图以作补充：

1. 矩阵基本运算——加减

矩阵 A、B、C 都提前设置好了，分别为：

$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 、 $B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ 、 $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 。

左下图表示要进行 $A-B+C$ 的运算，右下图为计算结果：



2. 矩阵基本运算——数乘

矩阵 A 提前设置好了，为：A=[[1,2],[3,4]] 。

左下图表示要进行-3.14 数乘 A 的运算，右下图为计算结果：

