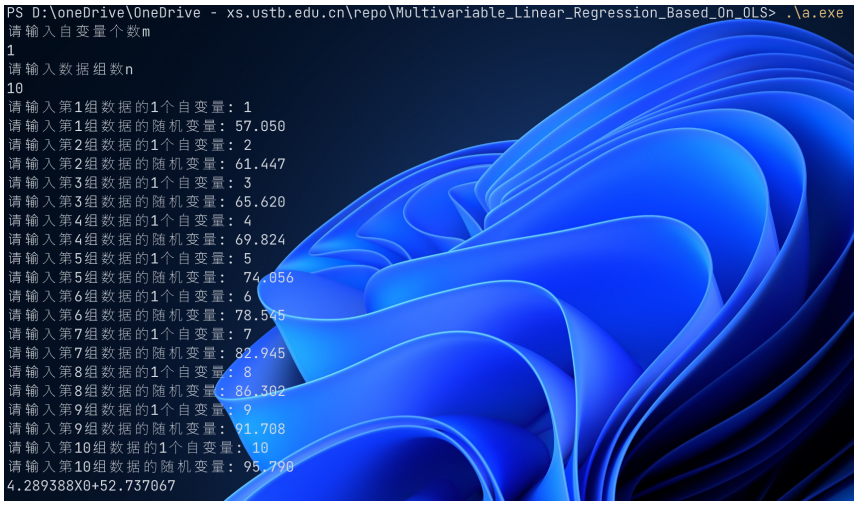
数据处理过程

赵方程

1	2	3	4	5	6	7	8	9	10
57.050	61.447	65.620	69.824	74.056	78.545	82.945	86.302	91.708	95.790

行波法测量声速实验数据表



使用基于最小二乘法的线性回归分析可得

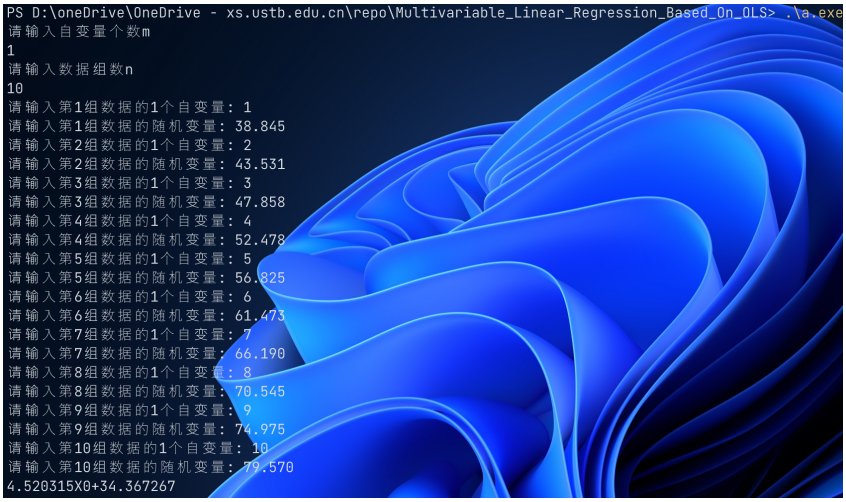
$$b_{行波法} = 4.28mm$$
$$\lambda_{行波法} = 8.578mm$$
$$U_{\lambda行波法} = 0.17mm$$
$$S_{b行波法} = 0.035mm$$

计算可得

$$V = 342m/s$$
$$U_V = 7m/s$$

]同理

1	2	3	4	5	6	7	8	9	10
38.845	43.531	47.858	52.478	56.825	61.473	66.190	70.545	74.975	79.570



使用基于最小二乘法的线性回归分析可得

$$
\begin{aligned}
b_{\text{驻波法}} &= 4.520mm \\
\lambda_{\text{驻波法}} &= 9.04mm \\
U_{\lambda\text{驻波法}} &= 0.046mm \\
S_{b\text{驻波法}} &= 0.01mm
\end{aligned}
$$

计算可得

$$
\begin{aligned}
V &= 360.4m/s \\
U_V &= 7m/s
\end{aligned}
$$

由公式

$$
v = \sqrt{\frac{\gamma RT_0}{\mu}}
$$

可得

$$
\gamma = 1.40
$$

部分计算所用的cpp源代码如下

repo地址:https://github.com/Equationzhao/Multivariable_Linear_Regression_Based_On_OLS.git

```cpp
#include "Matrix.h"
#include "Square.h"
#include "LinerEquation.h"
#include "OLS_MLR.h"

using namespace std;

auto get( double** x, double* y, const int& m, const int& n ) -> void
{
    for (int i = 0; i < n; ++i)
    {
```

```
        printf("请输入第%d组数据的%d个自变量: ", i + 1, m);
        for (int j = 0; j < m; ++j)
        {
            cin >> x[j][i];
        }
        printf("请输入第%d组数据的随机变量: ", i + 1);
        cin >> y[i];
    }
}

auto printAns(const OLS_MLR& mlr )
{
    for (int i = 0, size = mlr.getAns().size(); i < size; ++i)
    {
        if (!i)
        {
            printf("%lfX%d", mlr.getAnsOf(i), i);
        }
        else
        {
            if (mlr.getAnsOf(i) > 0)
            {
                cout << "+";
            }
            if (i == size - 1)
            {
                printf("%lf", mlr.getAnsOf(i));
            }
            else
            {
                printf("%lfX%d", mlr.getAnsOf(i), i);
            }
        }
    }
}

auto main() -> int
{

    int m, n;
    cout << "请输入自变量个数m\n";
    cin >> m;
    cout << "请输入数据组数n\n";
    cin >> n;

    const auto x = new double *[m];
    for (int i = 0; i < m; ++i)
    {
```

```
        x[i] = new double[n];
    }
    const auto y = new double[n];

    get(x, y, m, n);

    OLS_MLR mlr(x, y, m, n);

    printAns(mlr);

    printf
    (
        "\n\n\n偏差平方和为 %lf\n平均标准偏差为 %lf\n复相关系数为 %lf\n",
        mlr.getSumOfSquares(),
        mlr.getStdDeviation(),
        mlr.getMultiple_Correlation_Coefficient()
    );

    for (int i = 0; i < m; ++i)
    {
        delete[]x[i];
    }
    delete[]x;
    delete[]y;

    system("pause");
}
```