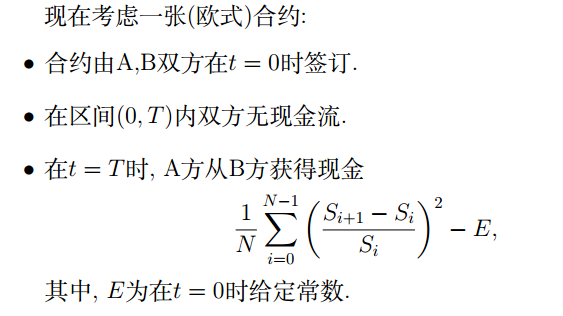
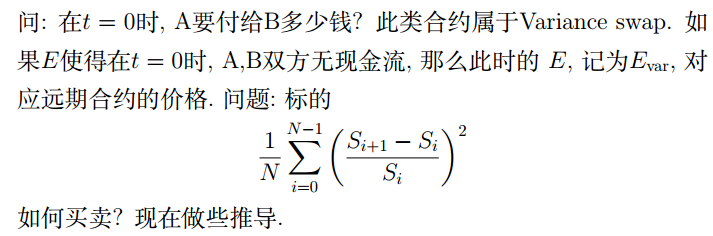
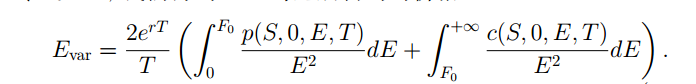
1. vsv问题叙述。





Vsv问题简单来说，就是以上述表达式为标的，求出一个E，使得在0时刻这个合约不产生现金流。

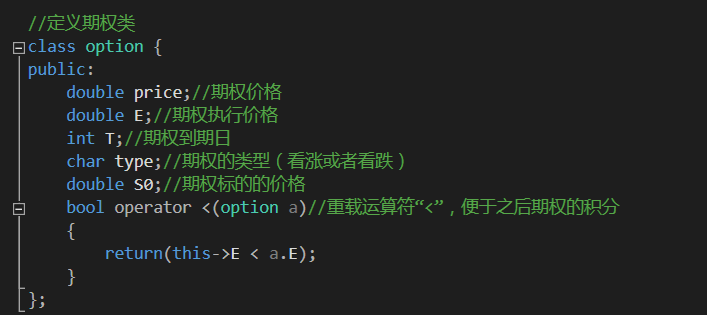
通过一些简单的推导，得到了如下表达式：



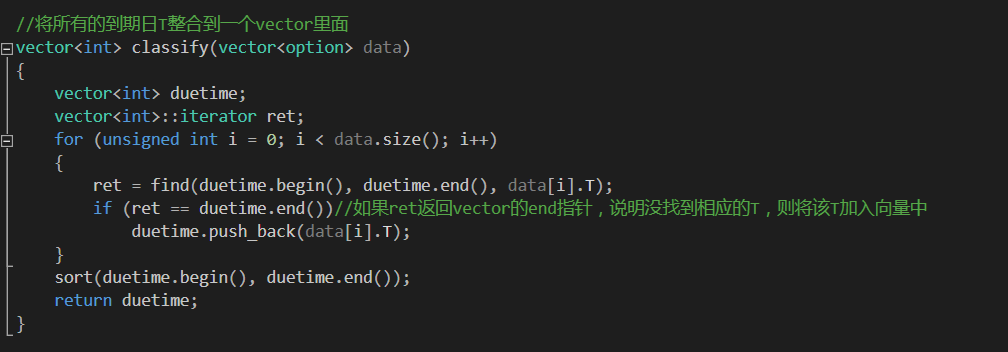
1. 无风险利率r的确定。

这里取i=0.034788为无风险利率，该利率为短期国债收益率。

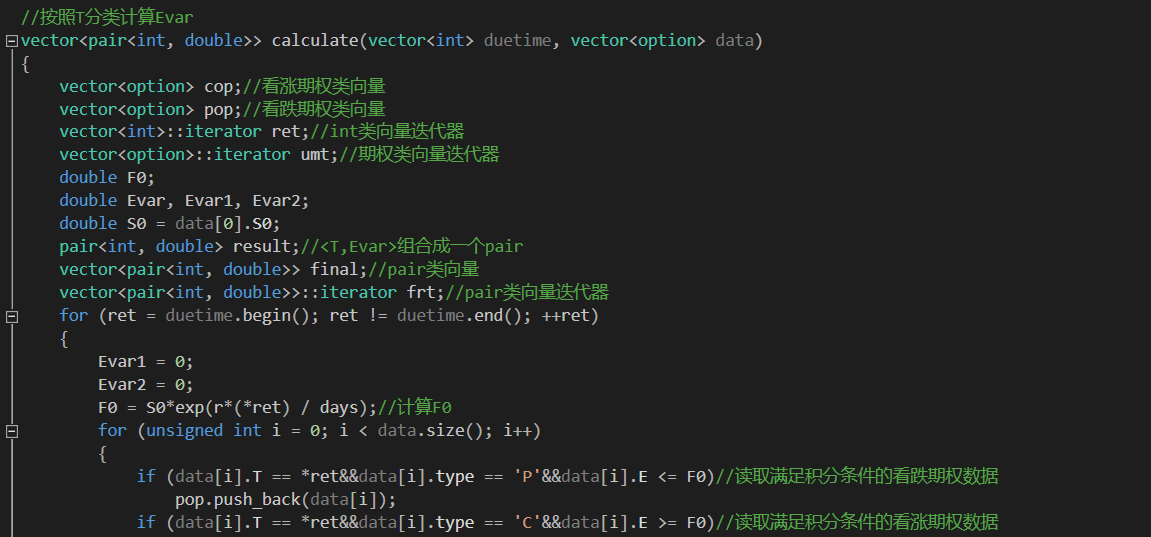
1. 数据说明：网上下载的数据较为繁杂，原始数据存储在50ETF期权文件夹中，由于该数据集中有很多变量没有实际的作用，故先运用python进行数据的清洗，将清洗好的数据输出到工作目录中的data文件夹。本次作业用到了20171010-20171013四天的数据。每次只需要运行C++程序就可以得到结果。
2. C++程序设计思路。
3. 定义期权类，类成员变量包括期权的价格price、期权的执行价E、期权的到期日T、期权的类型（看涨期权或者看跌期权）、期权标的指数的价格S0。重载 “<”为期权的E的大小比较（因为积分是对E积分，故为了方便之后积分，这里重载“<”运算符便于提取出满足积分条件的期权）

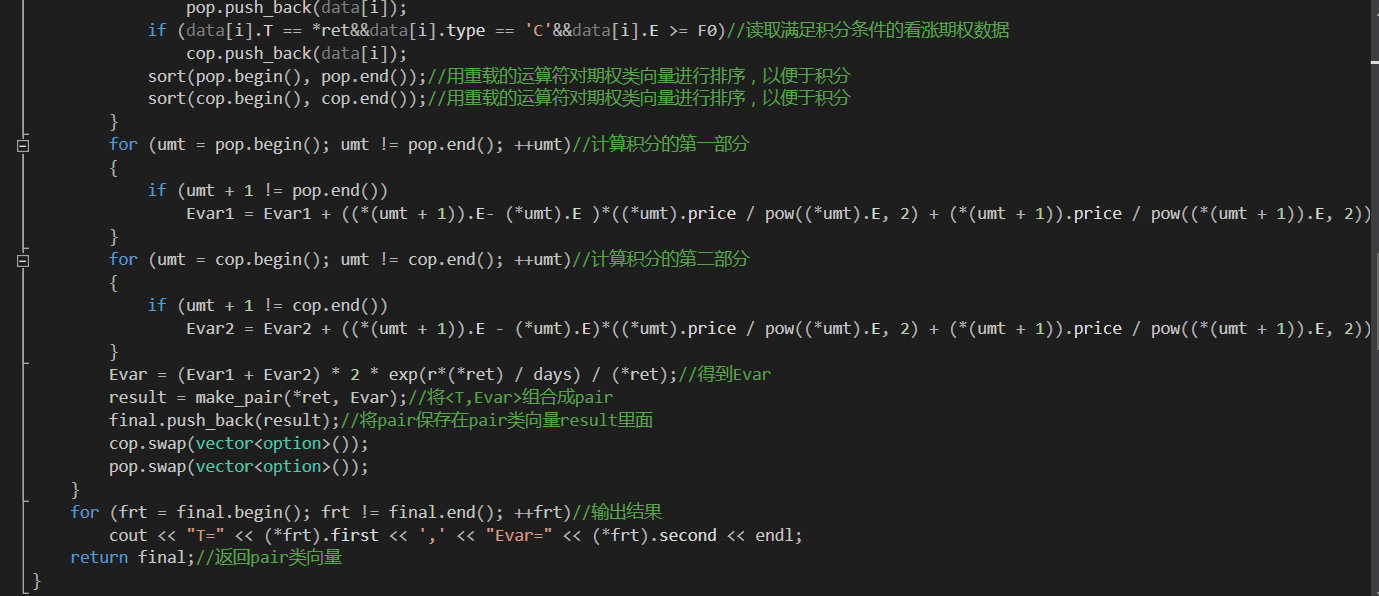


1. 由于每天都有很多期权进行交易，而这些期权的到期日T不同，而Evar的计算中，需要保证积分的时候T一样，所以定义vector<int> classify(vector<option> data)函数，将每天的数据读入data之后，提取出所有不同的T，返回一个vector<int>类型的变量duetime，其中储存所有的T的可能取值。

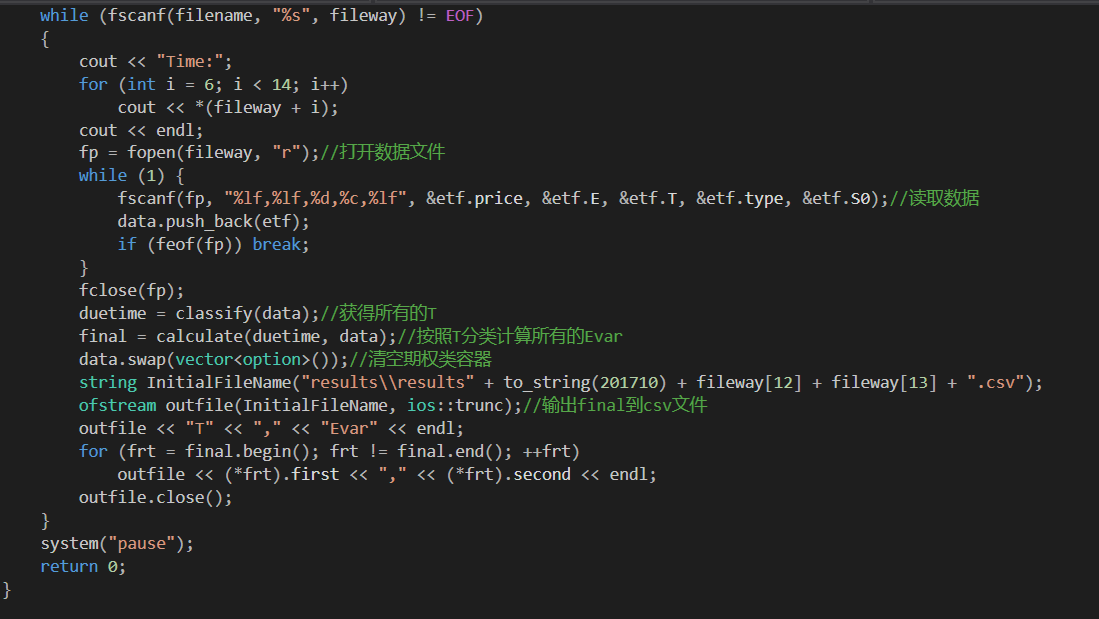


1. 定义函数vector<pair<int, double>> calculate(vector<int> duetime, vector<option> data)来计算每天不同到期日T对应的Evar。duetime中储存了所有T的可能取值，data中储存了所有的期权数据，按照文献定义的方法，可以计算出Evar。将<T，Evar>用一个pair来储存，然后将所有的这样的pair存在一个vector<pair<int, double>>里作为该函数的返回值。





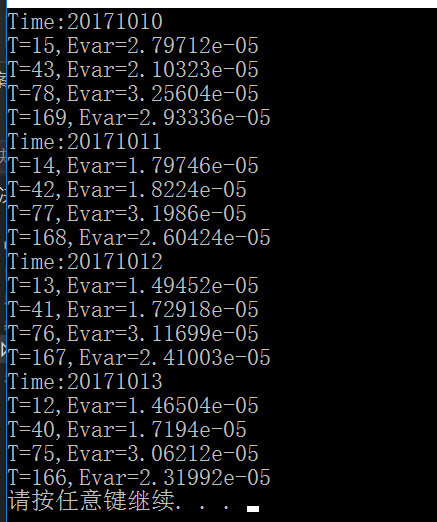
1. 主函数包括了读取数据部分、计算部分和输出到文件部分。从文件“filename.txt”读取数据的路径，再按照天为单位，从文件夹data每次读入一天的数据，然后调用函数计算不同T对应的Evar，然后输出结果到文件夹results。



1. 计算结果讨论。

注意到每天都有不同到期日的期权进行交易，所以本次程序分析了20171010-20171013四天的数据，按照天为单位来进行计算，得到了每天不同到期日T对应的Evar，一共4\*4=16个Evar。计算结果已经输出到了results文件夹中。

其中C++运行出来的结果如下：



分析这些Evar的计算结果，可以看出：

1. 到期日相同的期权，随着到期日的临近，（例如20171010日T=15的期权和20171011日T=14的期权是相同的期权），Evar在变小。
2. 同一天内，Evar的大小和T并没有显著的关联，T=78左右的期权Evar是最大的，可能的原因是：一般意义上，时间越长，期权的总体波动率可能越大，可是T太大了导致波动率已经可以通过统计规律来进行预测，所以T=166的Evar反而较小。
3. Evar均为10e-05这个量级，可见Evar是很小的一个量级的数。

注释后截图代码嵌入 3 ，4 最后word转为pdf提交