计算与编程第一次作业报告

王超 12031012

#第一题

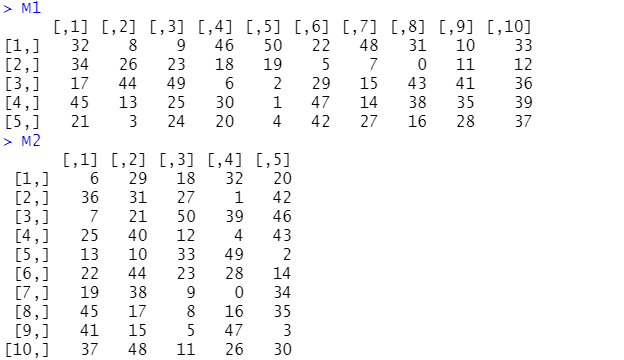
思路：

为了实现流程图，我使用了（if else）语句，嵌套了三层。参数a，b和c有很多大小顺序，甚至可能相等。该流程图仅显示了四种可能性，因此我的代码也反映了这四种类型的情况。具体注释见相关脚本。

#第二题

思路：

#2.1首先生成0到50之间的五十个随机整数，再将这两组数赋值给所需要的矩阵。

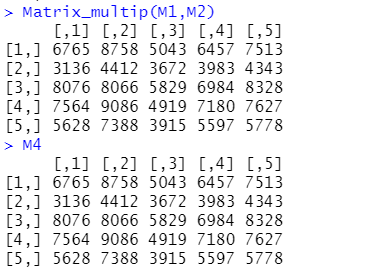


#2.2 定义Matrix\_multip函数，计算M1\*M2 = M3。

M1\*M2 会进行250次乘积，每十次乘积之和会生成M3的一个元素。用sum来记录每十次乘积之和，并编号记录，每记录好一个sum值，需要对其进行初始化。

用for循环来实现250次乘积，嵌套三层循环能更好的实现M1和M2中每个元素相乘。最终会生成25个sum值，将其赋值给M3中相应的元素即可得到M3。

用M4 <- M1 %\*% M2 来验证M3，发现它们相等。具体注释见相关脚本。



#第三题

思路：

第一二层直接用元素和向量来表示，很好输出。当k大于2层时，考虑生成k x k的矩阵，用来记录3~k行的数。矩阵的第i行会有i个有用的数。将第1，2行的前1，2个数赋值为1。同时第i行的第1个和第i个数总是为1。用双重for循环来实现对第i行2~（i-1）个元素的赋值。函数返回矩阵的第k行即可。具体注释见相关脚本。



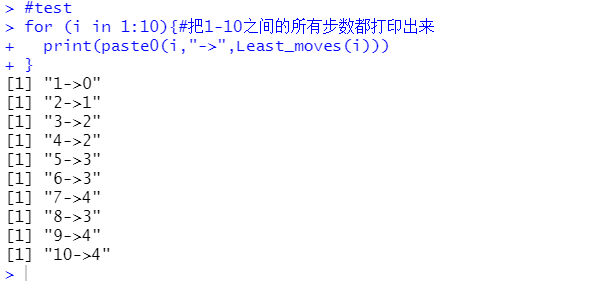
#第四题

思路：

尝试使用递归求解，因为每次移动都只会倍增或者加1，在1~100之间，第n（n>2）个奇数的步数是第n-1个偶数的步数再加一。

即有steps（2i+1）=steps（2i）+1，而且steps（2i） = steps(i)+1。

即有当i大于3时，如果为偶数，则计算i/2的步数加一即可得到i的步数；如果为奇数，则计算i-1的步数，再加一即可得到i的步数。一直递归到i=1，2，3时的情形即可。具体注释见相关脚本。



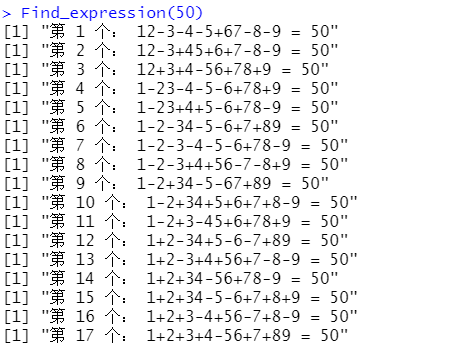
#第五题

思路：

#5.1

采用穷举法 在123456789内插入“ ” “+” “-” 三种操作符，因此有 3^8 种情况，直接生成（1：6561）的数组，用来循环。将1：6561化为3进制的数储存到6561\*8的矩阵中，从右往左储存。不满8位的其余位置用0赋值。矩阵的每一行就代表了一种运算方案，然后进行6561次循环，用矩阵每一行的8个数对123456789逐一处理，0.1.2分别代表不加符号/-号/+号。将见矩阵记录好的运算符插入到123456789中，0,1和2分别代表不插符号，减号和加号。用字符串str来记录123456789之间的操作符号，用paste0 函数来从1到9遍历，并形成所需要的字符串str。然后用eval(parse(text=str))函数计算得到sum值。

对每个sum值和目标值对比，如果相等，则输出对应的等式（str）。具体注释见相关脚本。



#5.2

重新定义了个函数Find\_solution，函数主体部分和5.1中Find\_expression（）相同，只是在return处做了修改，返回值变为满足条件数的个数。执行最后一次循环时，返回t值。函数Find\_solution测试1-100内各个数的满足条件的个数，用Total\_solutions[i]记录。用for循环遍历Find\_solution(1~100），并记录其中满足条件数的最大值和最小值；记录Total\_solutions[1~100]中等于最大值和最小值的数，然后输出。具体注释见相关脚本。

用plot(seq(1,100,1),Total\_solutions,type="o",xlab="number",ylab="solutions")作图即可

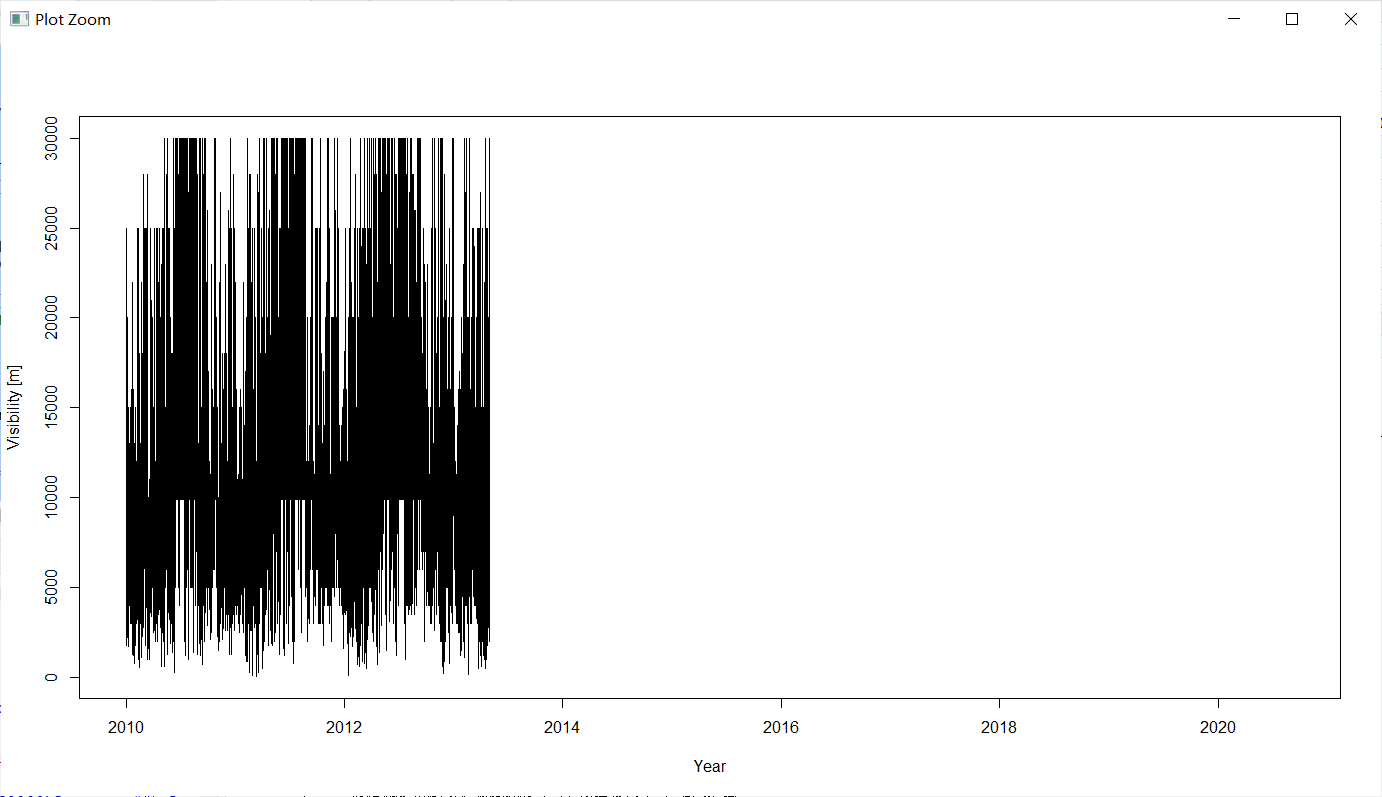
#第六题

思路：

#6.1 绘制时间序列图表

考虑到数据中有字符和逗号，首先将x转化为日期格式数据，将导入y的数据按照逗号分割为列表，将列表转化为更好操作的字符矩阵。整理数据，去除不合格的数据，将处理后的字符矩阵转换为数值矩阵。

此时发现因为在2013年之后数据不符合要求，都被筛掉了！作图数据仅仅到2013年。具体注释见相关脚本。



6.2 分析可见度的变化趋势

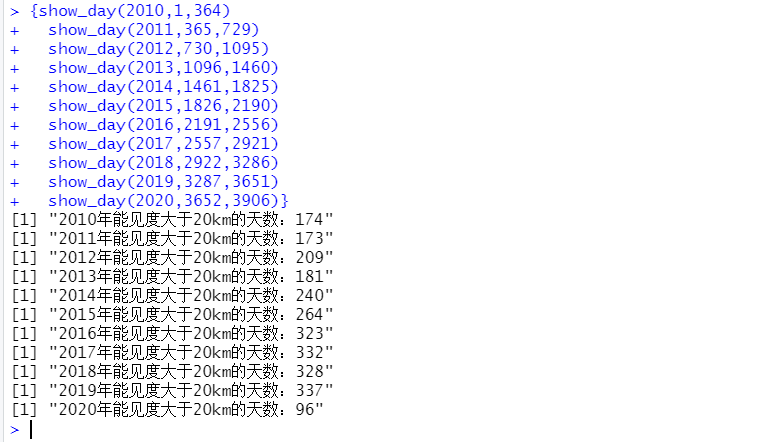
分辨率变为每天，需要对数据再次处理，把日期的转化为日期数据，统一格式后转化为数值型，并且把数据起始日期作为第一天。将导入y2的数据按照逗号分割为列表，将列表转化为更好操作的字符矩阵，将处理后的字符矩阵转换为数值矩阵，将y[,1]中的所有大于160000的值剔除。因为发现数据中有异常值，如999999。

用for循环，逐一找到每天内的最高可见度，并将每一天的最高可见度储存起来，最后分别判断每一年中位于不同可见度区间的天数 ，并打印出来，进行分析。具体注释见相关脚本。





假定能见度20km以上认为当天的能见度较好，从2010年到2020年十年来，能见度较好的天数如下图所示，数据显示十年来深圳的能见度在逐年变好，另外2020年数据仅记录到9月11号。



#第七题

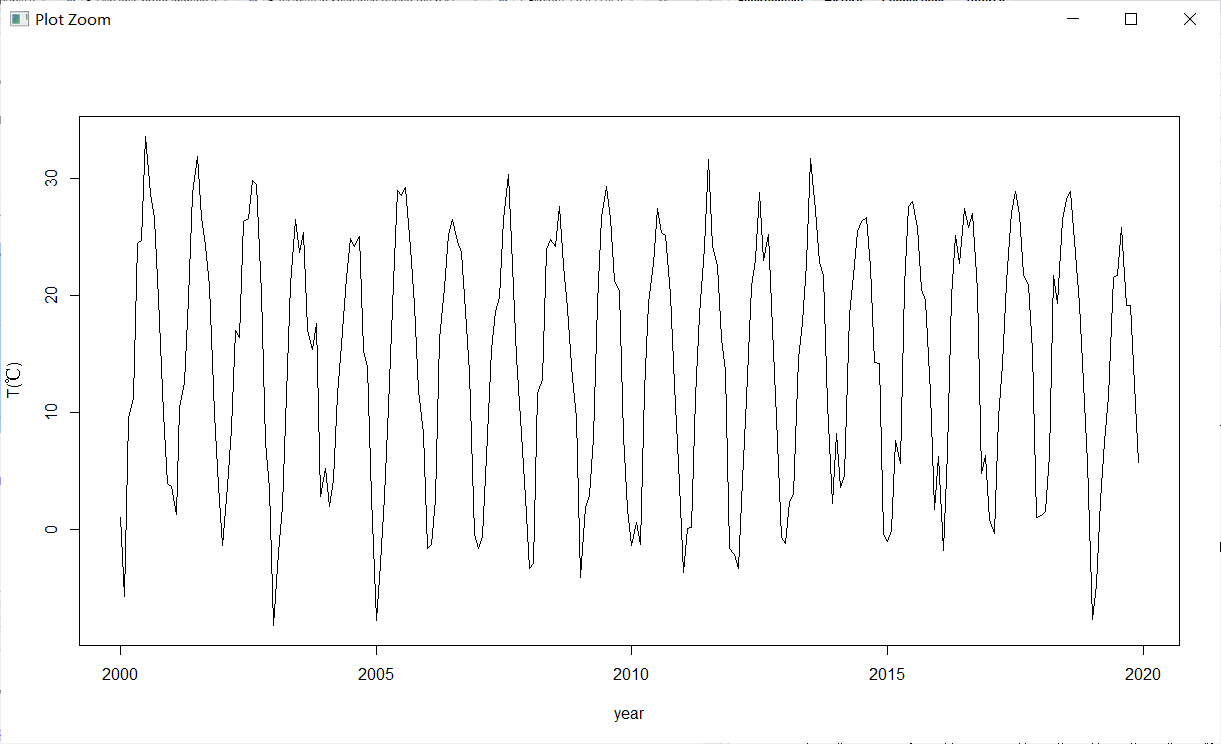
思路：

#7.1读入数据并整理

导入文件河北漳卫河流域2000~2019年逐月平均温度 Temperature.csv，探究河北漳卫河流域P1处温度的变化情况，发现数据中温度的单位为0.1℃，将单位变为1℃，再整理数据中异常值。

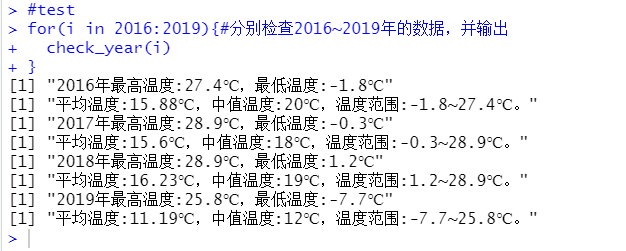
#7.2 绘制p1点近年来温度的时间序列图

定义一个函数可以plot任意年份的年度温度变化图，分别做出逐月平均温度的波动情况。最后做出2000年到2019年整个期间逐月平均温度的变化情况。



#7.3 分析数据的均值，范围等

定义一个检查数据的函数，分别检查2000~2019年的数据，并输出。具体注释见相关脚本。



从2000年到2019年，每年月平均最高温度变化不大，但月平均最低温度变化较大，2019年月平均最低温度为-7.7℃，2004年月平均最低温度为2℃。各年平均温度都在15~16摄氏度左右，2019年年平均温度相对较低，仅为11.19℃。