广州大学大学实验报告

学院 机械与电气工程学院

班级 机器192

姓名 何启海

学号 1907700047

实验名称 多源数据集成、清洗和统计

成绩

多源数据集成、清洗和统计 实验报告

开课实验室 2020 年­­­­ 12 月 4日

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学院 | 机械与电气工程学院 | | 年级、专业、班 | 机器192 | 姓名 | 何启海 | 学号 | 1907700047 |
| 实验课程名称 | | 机器学习和数据挖掘 | | | 成绩 | | |  |
| 实验项目名称 | | 多源数据集成、清洗和统计 | | | 指导老师 | | | 彭伟龙 |
| 1. 实验目的   对多源数据进行集成、清洗和统计   1. 实验原理  **题目** 广州大学某班有同学100人，现要从两个数据源汇总学生数据。第一个数据源在数据库中，第二个数据源在txt文件中，两个数据源课程存在缺失、冗余和不一致性，请用C/C++/Java程序实现对两个数据源的一致性合并以及每个学生样本的数值量化。  数据库表：ID (int), 姓名(string), 家乡(string:限定为Beijing / Guangzhou / Shenzhen / Shanghai), 性别（string:boy/girl）、身高（float:单位是cm)）、课程1成绩（float）、课程2成绩（float）、...、课程10成绩(float)、体能测试成绩（string：bad/general/good/excellent）；其中课程1-课程5为百分制，课程6-课程10为十分制。  txt文件：ID(string：6位学号)，性别（string:male/female）、身高（string:单位是m)）、课程1成绩（string）、课程2成绩（string）、...、课程10成绩(string)、体能测试成绩（string：差/一般/良好/优秀）；其中课程1-课程5为百分制，课程6-课程10为十分制。 **参考** 数据库中Stu表数据   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ID | Name | City | Gender | Height | C1 | ... | C10 | Constitution | | 1 | Sun | Beijing | boy | 160 | 87 |  | 9 | good | | 2 | Zhu | Shenzhen | girl | 177 | 66 |  | 8 | excellent | | ... | ... | ... | ... | ... | ... | ... | ... | ... |   student.txt中   |  | | --- | | ID Name City Gender Height C1 。。。 C10 Constitution  202001 Sun Beijing male 180 87 。。。 9 good  202003 Tang Hanghai male 156 91 。。。 10 general  ... ... ... .. ... .. ... ... ... |   两个数据源合并后读入内存，并统计：  学生中家乡在Beijing的所有课程的平均成绩。  学生中家乡在广州，课程1在80分以上，且课程9在9分以上的男同学的数量。(备注：该处做了修正，课程10数据为空，更改为课程9)  比较广州和上海两地女生的平均体能测试成绩，哪个地区的更强些？  学习成绩和体能测试成绩，两者的相关性是多少？（九门课的成绩分别与体能成绩计算相关性）   1. 使用仪器、材料   Vs2019，C语言，结构体，链表   1. 实验步骤 2. 首先创建结构体   struct students  {  char ID[7];//id  char name[30]; //名字  char city[15]; //城市  char gender[15]; //性别  char height[15]; //身高  char score1[6]; //成绩1  char score2[6];//成绩2  char score3[6];//成绩3  char score4[6];//成绩4  char score5[6];//成绩5  char score6[6];//成绩6  char score7[6];//成绩7  char score8[6];//成绩8  char score9[6];//成绩9  char score10[6];//成绩10  char constitution[20];//体能  int normative;//不规范性检测，当数据冲突时，填写得较为规范的可信度较高  struct students\* next;  };  因为考虑到读取题目（ID，city等等），所以都用char数组，normative就是不规范性，数字越小越规范   1. 进行文件的读取，这里主要涉及到链表的建立和读取，因为TXT和csv打开方式都一样，就只展示一个的，步骤就是p1和p2分别用malloc获取分配动态内存，p1读入数据，然后p1的next指向p2，然后p1=p2，p2再获取新的内存地址，循环往复。   struct students\* opentxt(struct students\* head)  {  struct students\* p1, \* p2,\*p3;  int checkheight;  printf("请输入打开的txt文件名\n");  char fname[30] = {0};  char c[100] ;  int m=0;//一行数据的第几个ID,Name,City,Gender,Height,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,Constitution  scanf\_s("%s", fname,30);  for (int i = 0; i < 30; i++)  {  if (fname[i] == 0)  {  fname[i] = '.';  fname[i+1] = 't';  fname[i+2] = 'x';  fname[i+3] = 't';  fname[i+4] = '\0';  break;  }  }  FILE\* fp = NULL;  if (NULL == (fp = fopen(fname, "r+")))  {  printf("文件打开失败，请检查!\n");  return(head);  }  else  {  printf("txt文件打开成功\n");  head = NULL;  }  p3 = NULL;  p2 = (struct students\*)malloc(LEN);  p1 = (struct students\*)malloc(LEN);  if (p1 == NULL) {  printf("内存分配不成功！\n");  }  head = p1;  rewind(fp);  while (!feof(fp))  {    for (int j = 0; j < 100; j++)  {  if (m==16)  {  m = 0;  p3 = p1;//防止最后溢出多一个结构体  p1->next = p2;  p1 = p2;  p2 = (struct students\*)malloc(LEN);  }  fscanf\_s(fp, "%c", &c[j],1);//逐个获取输入字符（包括空格和换行符）  if (c[j]==','||c[j]=='\n')  {  c[j] = '\0';  switch (m)  {    case 0:  {  p1->normative = 0;  strcpy(p1->ID, c);  break;  }  case 1:  {  strcpy(p1->name, c);  break;  }  case 2:  {  strcpy(p1->city, c);  break;  }  case 3:  {//先不按照TXT的规范来，  if (!strcmp("male", c))  {  strcpy(p1->gender, "boy");    }  else if (!strcmp("female", c))  {  strcpy(p1->gender, "girl");    }  else  {  strcpy(p1->gender, c);  p1->normative += 1;  }  break;  }  case 4:  {//先不按照TXT的规范来  /\*if (strcmp("\0", c))  {  checkheight = (int)c;  if (checkheight > 3)  {  c[3] = c[3];  c[2] = c[1];  c[1] = ',';  c[ 4] ='\0';  p1->normative += 1;  }  }\*/  if (strcmp("\0", c))  {  checkheight = atoi(c);  if (checkheight < 3)  {    c[1] = c[2];  c[2] = c[3];  c[3] = '\0';  }  else  {  p1->normative += 1;  }    }  else  {  p1->normative += 1;  }  strcpy(p1->height, c);  break;  }  case 5:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score1, c);  break;  }  case 6:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score2, c);  break;  }  case 7:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score3, c);  break;  }  case 8:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score4, c);  break;  }  case 9:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score5, c);  break;  }  case 10:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score6, c);  break;  }  case 11:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score7, c);  break;  }  case 12:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score8, c);  break;  }  case 13:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score9, c);  break;  }  case 14:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->score10, c);  break;  }  case 15:  {  if (!strcmp("\0", c))  {  p1->normative += 1;  }  strcpy(p1->constitution, c);  break;  }  }  m++;  j = 0;  break;  }  }    }  rewind(fp);  p1 = p3;  p1->next = NULL;  return head;  }   1. 读取完两个数据源的文件后，先进行合并，即把txt的文件与csv的相同ID的作比较，进行覆盖或是补全。之后进行插入，把csv里没有而TXT里有的数据插入csv里，最后是去重，把重复id的数据进行比较后决定留下来的那一个。 2. 统计，主要就是用atoi进行强制类型转换，把char转换为int类型，然后进行各种统计，体能成绩则由bad/general/good/excellent分别为0,1,3,4分。考虑到general到good应该是提升比较大的地方 ，所以相差两分。同时统计相关性时用了int\* b[9], \* bk[9];这两个指针数组，因为是第一次使用所以写起来磕磕绊绊的。 3. 实验过程原始记录（数据、图表、计算等）            1. 实验结果及分析   这次的作业，对我来说是很有挑战性的，因为我的专业并没有对C语言有过多的学习，而且从大一上学期学了C语言入门到现在已经快一年了，本来不怎么样的C语言基础更是忘了不少，这次几乎是从头开始学习结构体，结构体指针，动态内存分配，强制类型转换，文件的打开和读取，字符串比较和字符串复制（因为不能直接p1.score1=c这样子），还有指针数组等等，工程很大，写了好久才弄出来，期间遇到不少的bug和匪夷所思的问题，不过收获也是很大的，C语言的水平提高了不少。不过还是不知道怎么打开xlsl的Excel文件，只能转为csv来打开。 | | | | | | | | |