

暑假第一次 R 语言练习作业—第四小组

题目：

一、1 第一题（舒祥）

- 1.某日 Beijing 温度为 27C°，Chengdu 温度为 32C°，Haerbin 温度为 21C°，Guangzhou 温度为 33C°，用 for 和 it 命令找出温度大于 30C°的城市。
- 2.同上题数据，假定温度在 25C°以上为炎热，根据条件判断并判断 Beijing 温度是否炎热，保存在变量 is_hot 中
- 3.将题 1 中的温度构成向量保存在变量 tem 中，求平均温度

二、第二题（邱咏梅）

暑假大家都去了很多地方游玩，假如部分的同学的游玩情况如下：

names	qym	sx	chh	cbl	cy	jxj	czz
places	普吉岛	马尔代夫	苏州	扬州	九寨沟	西藏	香格里拉
partners	5	6	7	8	9	2	3
days	9	8	7	8	6	5	8

1. 请将上述表格储存于数据框 basic message 中。
2. 请另外生成一列表明该同学是国外还是国内游（1 表示国内，2 表示国外，用判断完成）
3. 请筛选出国内游，并且 partner 个数大于平均值，days 大于等于众数的同学。

三、第三题（陈嫣）

- 1, 已知 a,b,c 三人的身高分别为 170,160,180.将他们的身高数据储存到变量 x 中，并将该向量中各元素分别命名 a,b,c，再读取出身高最高的人的身高和姓名。
- 2, 构造矩阵 $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ ， $\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$ ，再由矩阵 $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ ， $\begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$

构造矩阵 A $\begin{pmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{pmatrix}$ ，B $\begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 2 & 2 \\ 2 & 2 \end{pmatrix}$ ，最后再求两矩阵的乘积 AB 和 BA

四、第四题（蔡铮铮）

- 1.求矩阵 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ 和 $\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$ 的加和。

用 Dataframe 制作表格,并把表格补充完整。

Name	Chinese	Maths	English	Summary
xiaoming	93	98	95	
xiaohong	83	99	89	
xiaomei	87	90	94	
xiaoli	89	92	88	
xiaojun	78	96	90	

- 3.在上述表格中添加一人的成绩，xiaoying 85 95 95

五、第五题（陈慧慧）

1. 请用一个函数生成一组等差数列 a_1, a_2, a_3, a_4 , 其中 $a_1 = 60, a_4 = 99$, 并将结果保存在 `scores` 这个变量中。

2. 一题 `scores` 变量中的元素分别对应于小高 2009, 2010, 2011, 2012 年期末数学成绩。

(1) 请以 `years` 和 `scores` 为列名, 生成一个数据框 `xiaogao_data0`

(2) 又知小高 2013, 2014, 2015 年的期末数学成绩为 95, 90, 91 分, 请将此数据加入 (1) 中所得的数据框中得到新的数据框 `xiaogao_data1`

3. 请用 `for` 和 `if` 语句, 找出第二题 2009—2015 年小高期末数学成绩下降的年份。

六、第六题（陈彦宏）

2016 里约奥运会正在进行, 其中备受关注的由我国游泳名将孙杨参加的男子 200 米自由泳决赛于北京时间 8 月 9 日上午九点二十一分在里约奥运水上场馆如期举行。

1) 其中参加决赛的运动员有: 1 泳道、南非选手查德·勒克洛斯, 2 泳道、俄罗斯联邦选手拉斯科, 3 泳道、美国选手德怀尔, 4 泳道、中国选手孙杨, 5 泳道、日本选手萩野公介, 6 泳道、德国选手比德尔曼, 7 泳道、美国选手哈斯, 8 泳道、英国选手詹姆斯·盖伊。请使用 `scan()` 命令, 按照运动员各自泳道的先后顺序, 将运动员的名字存储在变量 `names` 中, 将各自的国家存储到变量 `countries` 中, 并构造成数据框 `DF1`;

2) 比赛结束, 各选手的成绩按泳道先后顺序分别为: 1:45:20\1:45:91\1:45:23\

\1:44:65\1:45:90\1:45:84\1:45:58\1:45:49, 请将这组数据存储在变量 `time` 中, 构造成数据框并与 1) 中数据框合并为 `DF2`, 同时判断出哪个国家的哪一名选手取得了冠军(虽然大家都知道了哈哈~);

3) 赛后数据显示, 每名运动员在发令枪响后的反应时按泳道先后顺序分别为: 0.63\0.66\0.65\0.76\0.65\0.84\0.76\0.66, 将上述数据存储在变量 `rtime` 中, 构成数据框合并到 2) 中数据框 `DF2` 中成为 `DF3`, 分析哪个国家的哪个选手实际平均游泳速度最快, 并用适当方法分析反应时的长短是否会对运动员的成绩产生较大影响。

第七题（陈彬林）

1: 在多列数据中, 如何在中间插一列 (数据不用 excel 处理)

2: 题目描述如下:

有一个数据框如下:

	姓名	Tag1	Tag2	Tag3	Tag4	Tag5
1	xiaohong	租车	美图	聊天	租车	拍照
2	xiaoming	聊天	聊天	美图	租车	美食
3	xiaolv	美图	租车	美图	美食	聊天

其中 `tag` 代表使用工具类型 (可以看做是这一类型的 APP)

那么如何在 `tag2—tag5` 中寻找与 `tag1` 中 APP 类型相同的 APP 类型?

(即: 在第一个用户上, 他的 `tag1` 是租车, 则在 `tag2—tag6` 中寻找租车, 需要设计代码寻找。同样, 其他用户也需要通过这个算法来寻找与其相同的 APP 类型。请设计算法。)

参考答案：

第一题：

代码：

```
#1(1)
city=c("beijing","chengdu","haerbin","guangzhou")
temp=c(27,32,21,33)
data1=data.frame(city,temp)

for (i in 1:4)
{
  if( temp[i]>30)
  {print(city[i])}
}
```

结果：

	city	temp
1	beijing	27
2	chengdu	32
3	haerbin	21
4	guangzhou	33

```
+ }
[1] "chengdu"
[1] "guangzhou"
```

代码：

```
#1(2)
is.hot=(data1[1,2]>25)

#1(3)
tem=temp
mean(tem)
```

结果：

```
> tem=temp
> mean(tem)
[1] 28.25
```

第二题：

代码：

```
#2(1)
names1=c("qym","sx","chh","cbl","cy","jxj","czz")
place1=c("普吉岛","马尔代夫","苏州","扬州","九寨沟","西藏","香格里拉")
partners=c(5,6,7,8,9,2,3)
days=c(9,8,7,8,6,5,8)
basic.message=data.frame(names1,place1,partners,days)
```

结果：

	names1	place1	partners	days
1	qym	普吉岛	5	9
2	sx	马尔代夫	6	8
3	chh	苏州	7	7
4	cbl	扬州	8	8
5	cy	九寨沟	9	6
6	jxj	西藏	2	5
7	czz	香格里拉	3	8

代码：

```
#2(2)
travel=c("o","o","in","in","in","in","in")
basic.message=data.frame(basic.message,travel)
basic.message$tra = rep(0,length(names1));
for (i in 1:length(names1))
{
  if(basic.message$travel[i]=="o")
  {
    basic.message$tra[i]=2
  }else{
    basic.message$tra[i]=1
  }
}
basic.message$travel=NULL
```

结果：

	names1	place1	partners	days	tra
1	qym	普吉岛	5	9	2
2	sx	马尔代夫	6	8	2
3	chh	苏州	7	7	1
4	cbl	扬州	8	8	1
5	cy	九寨沟	9	6	1
6	jxj	西藏	2	5	1
7	czz	香格里拉	3	8	1

代码：

```
#2(3)
mean(basic.message$partners)
sort(table(basic.message$days))
mean1=5.71
mode1=8

for (i in 1:length(names1))
{if(basic.message$tra[i]==1&basic.message$partners[i]>mean(basic.message$partners)&basic.message$days[i]>=8)
{
  print(basic.message$names1[i])
}
}
```

结果：

```
> mode1=0
>
> for (i in 1:length(names1))
+ {if(basic.message$tra[i]==1&basic.message$par
+ {
+   print(basic.message$names1[i])
+ }
+ }
[1] cb1
Levels: cb1 chh cy czz jxj qym sx
> |
```

第三题：

代码：

```
#3(1)
names2=c("a","b","c")
x=c(170,160,180)
data2=data.frame(names2,x)
for (i in 1:length(names2)) {if(x[i]==max(x))
{
  print(names2[i])
}
|
}
```

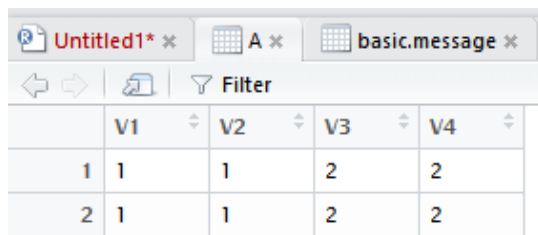
结果：

```
> #3(1)
> names2=c("a","b","c")
> x=c(170,160,180)
> data2=data.frame(names2,x)
> for (i in 1:length(names2)) {if(x[i]==max(x))
+ {
+   print(names2[i])
+ }
+ }
[1] "c"
```

代码：

```
#3(2)
matrix1=matrix(c(1,1,1,1),nrow = 2,ncol = 2)
matrix2=matrix(c(2,2,2,2),nrow = 2,ncol = 2)
A=cbind(matrix1,matrix2)
B=rbind(matrix1,matrix2)
AB=A%%B
BA=B%%A
```

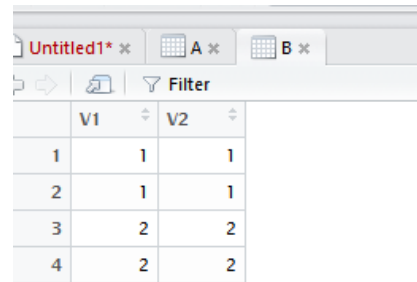
结果：



Matrix A is displayed in the RStudio environment. The window title is 'Untitled1*' and the active pane shows the matrix A. The matrix has 2 rows and 4 columns, with columns labeled V1, V2, V3, and V4. The data is as follows:

	V1	V2	V3	V4
1	1	1	2	2
2	1	1	2	2

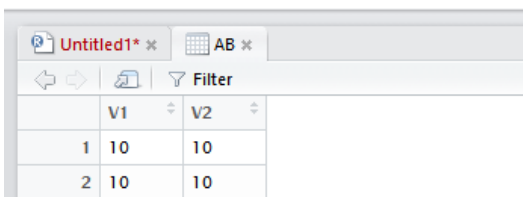
A :



Matrix B is displayed in the RStudio environment. The window title is 'Untitled1*' and the active pane shows the matrix B. The matrix has 4 rows and 2 columns, with columns labeled V1 and V2. The data is as follows:

	V1	V2
1	1	1
2	1	1
3	2	2
4	2	2

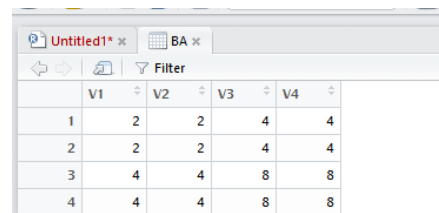
B :



Matrix AB is displayed in the RStudio environment. The window title is 'Untitled1*' and the active pane shows the matrix AB. The matrix has 2 rows and 2 columns, with columns labeled V1 and V2. The data is as follows:

	V1	V2
1	10	10
2	10	10

AB:



Matrix BA is displayed in the RStudio environment. The window title is 'Untitled1*' and the active pane shows the matrix BA. The matrix has 4 rows and 4 columns, with columns labeled V1, V2, V3, and V4. The data is as follows:

	V1	V2	V3	V4
1	2	2	4	4
2	2	2	4	4
3	4	4	8	8
4	4	4	8	8

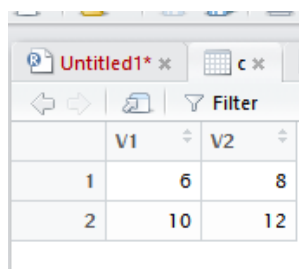
BA:

第四题：

代码：

```
#4(1)
matrix3=matrix(c(1,2,3,4),nrow = 2,ncol = 2,byrow = TRUE)
matrix4=matrix(c(5,6,7,8),nrow = 2,ncol = 2,byrow = TRUE)
c=matrix3+matrix4
```

结果：



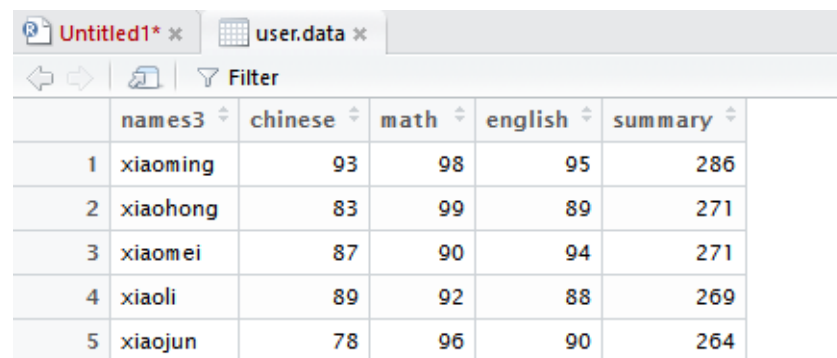
Matrix c is displayed in the RStudio environment. The window title is 'Untitled1*' and the active pane shows the matrix c. The matrix has 2 rows and 2 columns, with columns labeled V1 and V2. The data is as follows:

	V1	V2
1	6	8
2	10	12

代码：

```
#4(2)
names3=c("xiaoming","xiaohong","xiaomei","xiaoli","xiaojun")
chinese=c(93,83,87,89,78)
math=c(98,99,90,92,96)
english=c(95,89,94,88,90)
user.data=data.frame(names3,chinese,math,english)
user.data$summary=rep(0,length(names3))
for (i in 1:length(names3))
{user.data$summary[i]=user.data$chinese[i]+user.data$math[i]+user.data$english[i]}
}
```

结果：

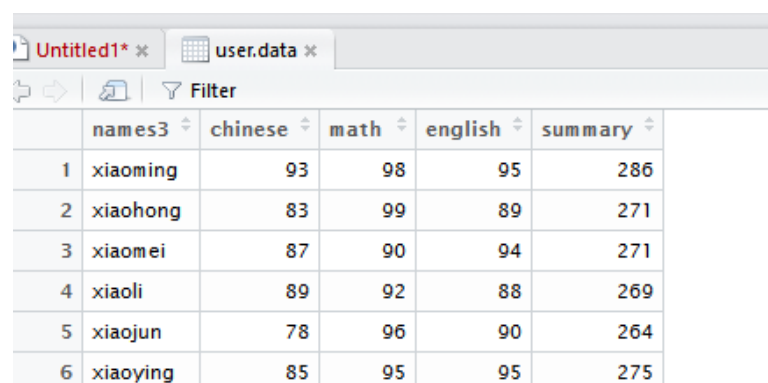


	names3	chinese	math	english	summary
1	xiaoming	93	98	95	286
2	xiaohong	83	99	89	271
3	xiaomei	87	90	94	271
4	xiaoli	89	92	88	269
5	xiaojun	78	96	90	264

代码：

```
#4(3)
names3=c("xiaoying")
chinese=c(85)
math=c(95)
english=c(95)
data_new=data.frame(names3,chinese,math,english)
data_new$summary=data_new$chinese+data_new$math+data_new$english
user.data=rbind(user.data,data_new)
```

结果：



	names3	chinese	math	english	summary
1	xiaoming	93	98	95	286
2	xiaohong	83	99	89	271
3	xiaomei	87	90	94	271
4	xiaoli	89	92	88	269
5	xiaojun	78	96	90	264
6	xiaoying	85	95	95	275

第五题：

代码：

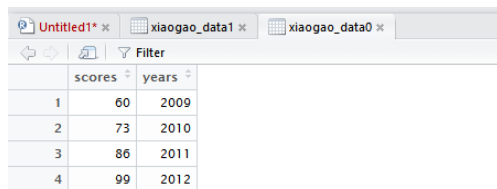
```
#5 1(1)
scores=seq(60,99,13)
```

代码：

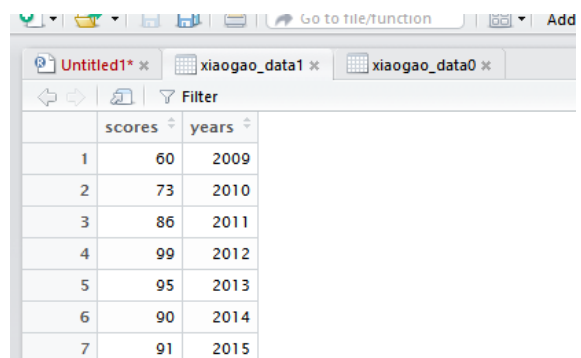
```
#5 2(2)
scores=c(95,90,91)
years=c(2013,2014,2015)
data_new1=data.frame(scores,years)
xiaogao_data1=rbind(xiaogao_data0,data_new1)

#5 3|
for (i in 1:7)
{if(xiaogao_data1[i+1,1]<xiaogao_data1[i,1])
{
  print(xiaogao_data1[i+1,2])
}
}
```

结果：



	scores	years
1	60	2009
2	73	2010
3	86	2011
4	99	2012



	scores	years
1	60	2009
2	73	2010
3	86	2011
4	99	2012
5	95	2013
6	90	2014
7	91	2015

而数学成绩下降的年份结果如下：

```
>
> #5 3
> for (i in 1:7)
+ {if(xiaogao_data1[i+1,1]<xiaogao_data1[i,1])
+ {
+   print(xiaogao_data1[i+1,2])
+ }
+ }
[1] 2013
[1] 2014
```


第六题：

代码：（没有理解题设中 scan 函数的用法和意义，只能用笨方法？？）

```
#6(1)
names=c("查德·勒克洛斯","拉斯科","德怀尔","孙杨","萩野公介","比德尔曼","哈斯","詹姆斯·盖伊")
countries=c("南非","俄罗斯联邦","美国","中国","日本","德国","美国","英国")
DF1=data.frame(names,countries)
```

结果：



	names	countries
1	查德·勒克洛斯	南非
2	拉斯科	俄罗斯联邦
3	德怀尔	美国
4	孙杨	中国
5	萩野公介	日本
6	比德尔曼	德国
7	哈斯	美国
8	詹姆斯·盖伊	英国

代码：

```
#6(2)
time=c("105.20","105.91","105.23","104.65","105.90","105.84","105.58","105.49")

DF2=cbind(DF1,time)
for (i in 1:length(names))

{if(DF2[i,3]==min(time))
{print(DF2[i,1])}}
}
```

结果：

```
> DF2=cbind(DF1,time)
> for (i in 1:length(names))
+
+ {if(DF2[i,3]==min(time))
+ {print(DF2[i,1])}}
+
+ }
[1] 孙杨
Levels: 比德尔曼 查德·勒克洛斯 德怀尔 哈斯 拉斯科 萩野公介 孙杨 詹姆斯·盖伊
```

代码：

```
#6(3)
rtime=c(0.63,0.66,0.65,0.76,0.65,0.84,0.76,0.66)
DF3=cbind(DF2,rtime)
DF3$time=as.character(DF3$time)
DF3$rtime=as.character(DF3$rtime)
DF3$rtime=as.numeric(DF3$rtime)
DF3$time=as.numeric(DF3$time)
DF3$rtime=as.numeric(DF3$rtime)
DF3$ctimes=rep(0,length(names))

for (i in 1:length(names))
{
  DF3[i,5]=DF3[i,3]-DF3[i,4]
  if(DF3[i,5]==min(DF3$ctimes))
  {print(DF3[i,1])}
}
```

在这里，先将 time 转化成 character，再把其转化成 numeric，可避免出现转化形式出错的问题。

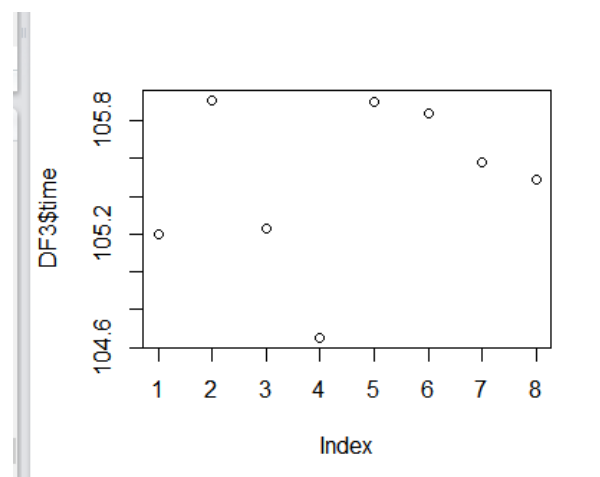
结果：

```
> for (i in 1:length(names))
+ {
+   DF3[i,5]=DF3[i,3]-DF3[i,4]
+   if(DF3[i,5]==min(DF3$ctimes))
+     {print(DF3[i,1])}
+ }
[1] 孙杨
Levels: 比德尔曼 查德·勒克洛斯 德怀尔 哈斯 拉斯科 萩野公介 孙杨 詹姆斯·盖伊
>
```

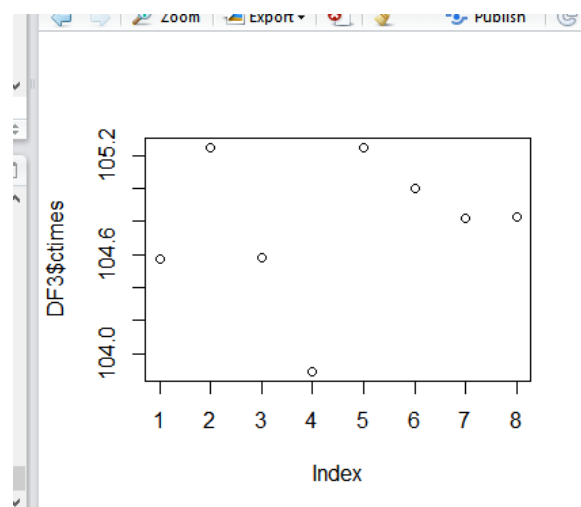
平均速度最快的仍是孙杨

接下来比较反应时间对选手影响的大小：

A=plot(DF3\$time)



C=plot(DF3\$ctimes)



通过比较以上图，C 图和 A 图差异不大，因此判断反应时间影响不大，原因可能是不同选手反应时间的差别不大（当然，反应时间越小越好）。

第七题：

代码：

```
#7(1)
a=c(0,0,0,0,0)
b=c(1,1,1,1,1)
df=data.frame(a,b)
df$c=rep(2,length(a))
df$d=rep(3,length(a))
df$e=rep(3,length(a))
#在c和d之间插上一列
x=rep(5,length(a))
df_new=cbind(df[,1:3],x,df[,4:5])
```

结果：

	a	b	c	d	e
1	0	1	2	3	3
2	0	1	2	3	3
3	0	1	2	3	3
4	0	1	2	3	3
5	0	1	2	3	3

中间插入一列后：

	a	b	c	x	d	e
1	0	1	2	5	3	3
2	0	1	2	5	3	3
3	0	1	2	5	3	3
4	0	1	2	5	3	3
5	0	1	2	5	3	3

代码：（两个方法得到数据框，第一种是懒人做法啦）

```
#7(2)
df1=read.table(file = "表格.txt",sep = " ",header = TRUE);
#huo zhe
names=c("xiaohong","xiaoming","xiaolv")
Tag1=c("租车","聊天","美图")
Tag2=c("美图","聊天","租车")
Tag3=c("聊天","美图","美图")
Tag4=c("租车","租车","美食")
Tag5=c("拍照","美食","聊天")
df2=data.frame(names,Tag1,Tag2,Tag3,Tag4,Tag5)
```

	names	Tag1	Tag2	Tag3	Tag4	Tag5
1	xiaohong	租车	美图	聊天	租车	拍照
2	xiaoming	聊天	聊天	美图	租车	美食
3	xiaolv	美图	租车	美图	美食	聊天

代码：

（思路：使用循环，把每个满足条件的 tag1 和 tag2-tag5 进行匹配，得出结果）
 首先把 tag 都转化成 character

```
df2$Tag1=as.character(df2$Tag1)
df2$Tag2=as.character(df2$Tag2)
df2$Tag3=as.character(df2$Tag3)
df2$Tag4=as.character(df2$Tag4)
df2$Tag5=as.character(df2$Tag5)
```

再使用循环

```
for (i in 1:3)
{if(df2[i,2]==df2[i,3])
{
  print(df2[i,1])
}
}#得出第一个满足题设的情况
```

第一个满足题设情况如下：

```
> for (i in 1:3)
+ {if(df2[i,2]==df2[i,3])
+ {
+   print(df2[i,1])
+ }
+ }#得出第一个满足题设的情况
[1] xiaoming
Levels: xiaohong xiaolv xiaoming
>
```

再循环：

```
for (i in 1:3)
{if(df2[i,2]==df2[i,4])
{
  print(df2[i,1])
}
}#可知结果
```

第二个满足题设情况：

```

>
> for (i in 1:3)
+ {if(df2[i,2]==df2[i,4])
+ {
+   print(df2[i,1])
+ }
+ }#可知结果
[1] xiaolv
Levels: xiaohong xiaolv xiaoming
~

```

再循环：

```

for (i in 1:3)
{if(df2[i,2]==df2[i,5])
{
  print(df2[i,1])
}
}#

```

第三个满足题设情况：

```

> for (i in 1:3)
+ {if(df2[i,2]==df2[i,5])
+ {
+   print(df2[i,1])
+ }
+ }#kezhijieguo
[1] xiaohong
Levels: xiaohong xiaolv xiaoming
>

```

再循环：

```

for (i in 1:3)
{if(df2[i,2]==df2[i,6])
{
  print(df2[i,1])
}
}#

```

没有结果（说明此条件无满足题设的情况）

```

> for (i in 1:3)
+ {if(df2[i,2]==df2[i,6])
+ {
+   print(df2[i,1])
+ }
+ }#kezhimeiyou duiyingqingkuang
> |

```

第二种做法：(一步解决)

```
#第二种做法|
for (i in 1:3)
{
  for (j in 3:6)
  {
    if (df2[i,2]==df2[i,j])
    { print(paste("(",i,",",j,")",sep=""))}
  }
}
```

结果：

```
<
Error: unexpected '}' in "}"
> for (i in 1:3)
+ {
+   for (j in 3:6)
+   {
+     if (df2[i,2]==df2[i,j])
+     { print(paste("(",i,",",j,")",sep=""))}
+   }
+ }
[1] "(1,5)"
[1] "(2,3)"
[1] "(3,4)"
> |
```

小组热议：(争论)

第六小题第一小问，不能理解题目意思，不会用 scan

第六小题第三小问，答案不同，难以达成一致。