答题纸

学号： 17263829　　姓名： 赵宜珺　　课程名：数据统计分析与 R

语言

一、选择题

1、A

2、D

3、A

4、C、D

5、B

6、B

7、B

8、B

9、C

二、判断题

1、×

2、√

3、√

4、√

5、√

6、×

7、√

8、√

9、×

10、×

11、×

12、√

13、√

三、填空题

1、生成 6 个符合正态分布的随机数

2、生成一个范围从 from 到 to 的，步长为 by 的向量

3、画以 dose 为横轴，drugA 为纵轴，点形状为 23，直线类型为 6，绘图区域的背景颜色为蓝色，默认绘图颜色为红色的点线图

4、txt、excel、csv、数据库文件

四、简答题

1、

```
Environment 中是所有变量
History 中是所有执行过的语句
```

2、

```
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
plot(x, xlab = "x", ylab = "y", main = "cex and font", ylim = c(1, 7), pch = 21)
```

3、

3、描述型统计量的函数: head()、summary()、mean()、length()、sum()、
sd()

分组统计量的函数:aggregate()、 describeBy()

4、

```
#冒泡排序

bubbleSort = function(vector){

    n = length(vector)

    for(i in 1 : (n - 1)){

        for(j in (i + 1) : n){

            if(vector[i] >= vector[j]){

                temp = vector[i]

                vector[i] = vector[j]

                vector[j] = temp

            }

        }

    }

    return (vector)

}

num <- c(3, 2, 4, 10, 1)

bubbleSort(num)
```
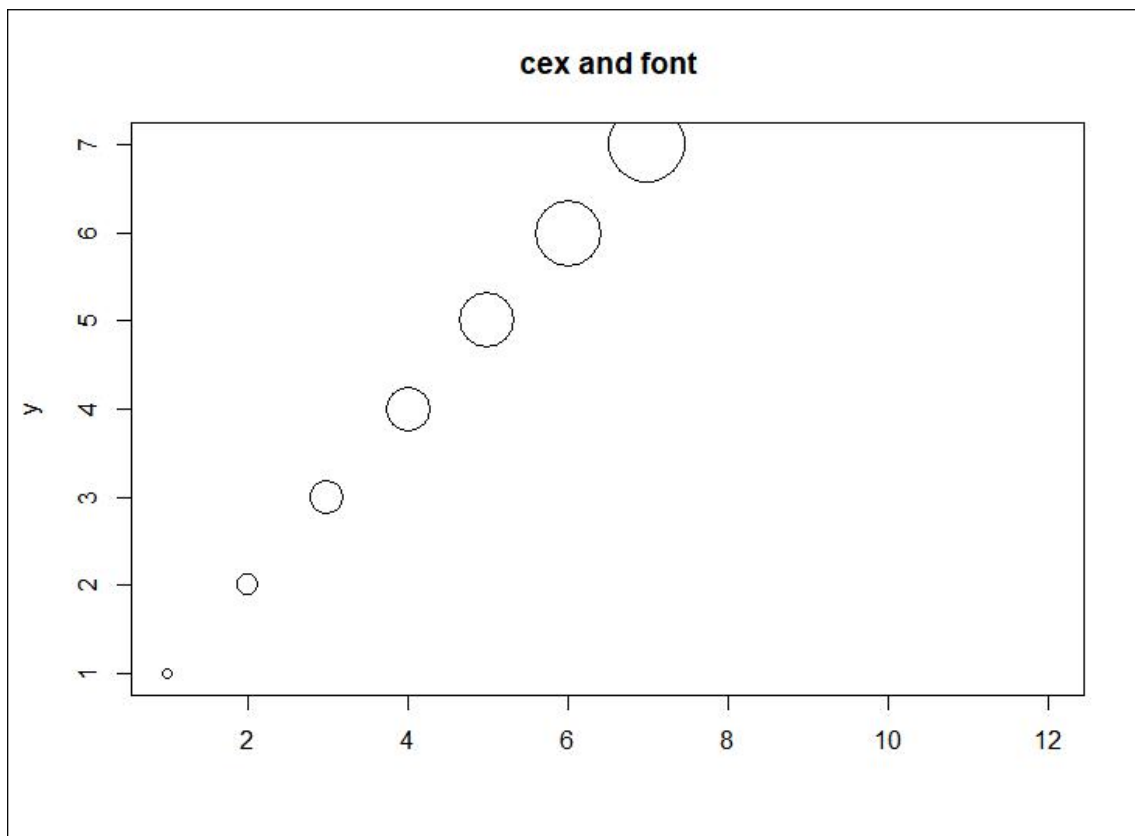
运行结果：

```
> num <- c(3, 2, 4, 10, 1)
> bubbleSort(num)
[1]  1  2  3  4 10
```

5、

```
#因子
database <- c("type1", "type2", "type1", "type1")
database <- factor(database)
#矩阵
y <- matrix(1:20, nrow = 5, ncol = 4) #5 行 4 列的矩阵
#数组
z <- array(1:24, c(2, 3, 4))
#列表
g <- "list"
h <- c(1, 2, 3, 4)
j <- matrix(1:10, nrow = 5)
mylist <- list(title = g, h, j)
```

6、

```
#画图
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
plot(x, xlab = "x", ylab = "y", main = "cex and font", ylim = c(1, 7), pch = 21, cex = c(1, 2, 3, 4, 5, 6, 7))
```

cex and font

五、编程题

```
#编程题目
#文件位置
fileName <- 'D:/数据分析与 R 语言/R/考试/成绩.csv'
#数学满分 600
set.seed(1)
Math <- round(runif(50, 1, 600), 2)
Math
#English 满分 60
set.seed(2)
English <- round(runif(50, 1, 60), 2)
English
#java 分数满分 100
set.seed(3)
Java <- round(runif(50, 1, 100), 2)
Java
#语文满分 150
set.seed(4)
Chinese <- round(runif(50, 1, 150), 2)
Chinese
#体育满分 150
```

```
set.seed(5)
Sports <- round(runif(50, 1, 150), 2)
Sports
#id 生成
id <- rep(1:50)
#名字, 手动输入
name <- c()
#创建数据框
myGrade <- data.frame(id, Math, English, Java, Chinese, Sports)
#去行名写入文件
write.csv(myGrade, fileName, row.names = FALSE)
#手动添加 name 后重新导入
myTable <- read.csv(fileName, header = TRUE)
#抽取 10 行观测
set.seed(6)
attach(myTable)
newTable <- myTable[sample(1:50, 10, replace = FALSE), ]
newTable
detach(myTable)
#自定义标准化函数
fun <- function(x){
    x <- as.numeric(x)
    return ((x - min(x)) / (max(x) - min(x)))
}
#进行标准化
newTable$Math <- fun(newTable$Math)
newTable$English <- fun(newTable$English)
newTable$Java <- fun(newTable$Java)
newTable$Chinese <- fun(newTable$Chinese)
newTable$Sports <- fun(newTable$Sports)
#求均值
newTable <- transform(newTable, average = apply(newTable[, 3: 7], 1, mean))
#评分
#分位点
quantile <- quantile(newTable$average, c(0.8, 0.6, 0.4, 0.2))
newTable <- within(newTable, {
    score <- NA
    score[average >= quantile["80%"]] <- "优"
    score[average >= quantile["60%"] & average < quantile["80%"]] <- "良"
    score[average < quantile["60%"] & average >= quantile["40%"] ] <- "中"
    score[average < quantile["40%"] & average >= quantile["20%"] ] <- "差"
    score[average < quantile["20%"]] <- "非常差"
})
```
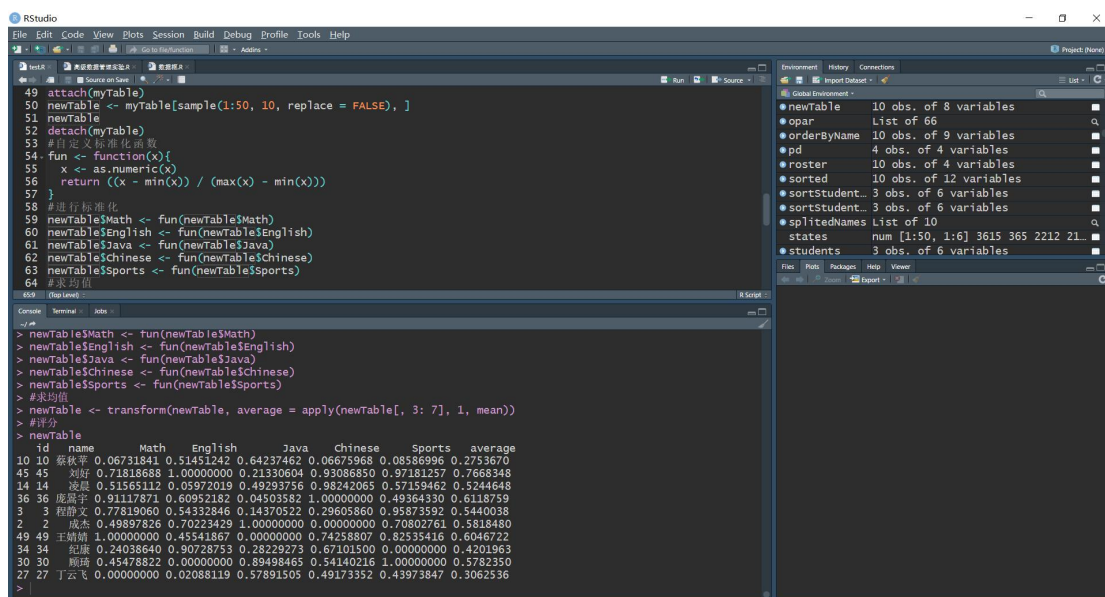
#根据姓排序

orderByName <- newTable[order(newTable$name),]


```
> myGrade
   id    Math English Java Chinese Sports
1   1 160.04   11.91 17.64   88.28  30.83
2   2 223.90   42.44 80.94    2.33 103.10
3   3 344.14   34.83 39.11   44.77 137.61
4   4 545.02   10.92 33.45   42.33  43.38
5   5 121.81   56.69 60.61  122.22  16.59
6   6 539.14   56.67 60.84   39.80 105.46
7   7 566.86    8.62 13.34  108.94  79.67
8   8 396.82   50.17 30.17  136.01 121.38
9   9 377.84   28.61 58.18  142.41 143.52
10 10  38.01   33.45 63.47   11.90  17.46
11 11 124.38   33.61 51.69  113.45  41.72
12 12 106.76   15.09 51.00   43.61  74.09
13 13 412.53   45.87 53.87   15.91  48.44
14 14 231.08   11.67 56.17  143.16  84.32
15 15 462.14   24.91 86.92   62.93  40.13
16 16 299.12   51.36 83.14   68.81  31.08
17 17 430.85   58.61 12.03  145.69  58.74
18 18 595.15   14.32 70.67   88.01 133.29
19 19 228.64   27.24 89.85  144.37  83.68
20 20 466.69    5.42 28.69  114.49 126.48
21 21 560.89   40.05 23.59  107.46 133.64
22 22 128.07   23.87  2.52  149.50 108.38
23 23 391.35   50.38 13.77   76.43  32.49
24 24  76.21    9.88 10.24   74.00  34.63
25 25 161.07   21.49 24.45   97.73  21.86
26 26 232.28   29.84 79.32  124.79  72.51
```

Console output (Window 1):

```
15 15     凌镜 462.14    24.91 86.92    62.93 40.13
16 16   沈陶榴 299.12    51.36 83.14    68.81 31.08
17 17     王颖 430.85    58.61 12.03   145.69 58.74
18 18   吴旭琴 595.15    14.32 70.67    88.01 133.29
19 19   杨荟蓉 228.64    27.24 89.85   144.37 83.68
20 20     叶雷 466.69     5.42 28.69   114.49 126.48
21 21   张沁心 560.89    40.05 23.59   107.46 133.64
22 22     张湘 128.07    23.87  2.52   149.50 108.38
23 23   郑欣跃 391.35    50.38 13.77    76.43 32.49
24 24   宗卫蓉  76.21     9.88 10.24    74.00 34.63
25 25   陈海俊 161.07    21.49 24.45    97.73 21.86
26 26   仇儒春 232.28    29.84 79.32   124.79 72.51
27 27   丁云飞   9.02     9.81 60.37    72.82 66.17
28 28   樊苏伟 230.05    22.07 91.10   126.42 144.93
29 29   葛文涛 521.94    57.80 56.48    77.54 22.15
30 30     顺琦 204.87     8.81 75.81    79.94 143.29
31 31   顾锦枫 289.77     1.61 38.54    85.50 67.26
32 32   胡湘伟 360.14    10.71 37.95    36.60  9.85
33 33     黄聪 296.63    48.80 17.86   131.82 42.00
34 34     纪康 112.54    52.26 45.88    98.52  5.64
35 35     姜勇 496.60    31.34 26.58    72.87  3.16
36 36   庞晟宇 401.41    38.00 34.29   145.68 73.59
37 37   沙红伟 476.75    50.82 89.07    69.54 89.68
38 38   宋金明  65.66    17.81 20.99    93.69 90.08
39 39   陶晨彦 434.50    40.37 58.34    58.87 60.25
40 40   邢佳伟 247.35     9.88 21.56     1.98 60.13
41 41     喻军 492.75    58.92 28.87   140.81 122.60
42 42   曹丹阳 388.59    18.52 78.84    37.08 36.04
43 43     陈玲 469.98     7.79 18.13    85.27 124.20
44 44   段红伟 332.27    10.63 57.50    27.95 80.49
45 45     刘妤 318.30    56.70 42.51   135.77 139.41
46 46     刘爽 473.82    47.90 27.49    13.56 82.92
47 47     陆星  14.98    58.51  5.73   135.27 114.26
48 48   施周莹 286.86    21.60 11.25   133.85 11.21
49 49   王婧婧 439.66    30.62 32.09   108.78 119.25
50 50   王子繁 415.95    48.81 80.26    85.16 95.29
>
```

Window 2 - Source editor:

```
41  #创建数据框
42  myGrade <- data.frame(id, Math, English, Java, Chinese, Sports)
43  #去行名写入文件
44  write.csv(myGrade, fileName, row.names = FALSE)
45  #手动添加name后重新导入
46  myTable <- read.csv(fileName, header = TRUE)
47  #抽取10行观测
```

Window 2 - Console:

```
42 42   曹丹阳 388.59    18.52 78.84    37.08 36.04
43 43     陈玲 469.98     7.79 18.13    85.27 124.20
44 44   段红伟 332.27    10.63 57.50    27.95 80.49
45 45     刘妤 318.30    56.70 42.51   135.77 139.41
46 46     刘爽 473.82    47.90 27.49    13.56 82.92
47 47     陆星  14.98    58.51  5.73   135.27 114.26
48 48   施周莹 286.86    21.60 11.25   133.85 11.21
49 49   王婧婧 439.66    30.62 32.09   108.78 119.25
50 50   王子繁 415.95    48.81 80.26    85.16 95.29
> set.seed(6)
> attach(myTable)
The following objects are masked _by_ .GlobalEnv:

    Chinese, English, id, Java, Math, name, Sports

> newTable <- myTable[sample(1:50, 10, replace = FALSE), ]
> newTable
    id   name   Math English  Java Chinese Sports
10 10 蔡秋苹  38.01   33.45 63.47   11.90  17.46
45 45   刘妤 318.30   56.70 42.51  135.77 139.41
14 14   凌晨 231.08   11.67 56.17  143.16  84.32
36 36 庞晟宇 401.41   38.00 34.29  145.68  73.59
3   3 程静文 344.14   34.83 39.11   44.77 137.61
2   2   成杰 223.90   42.44 80.94    2.33 103.10
49 49 王婧婧 439.66   30.62 32.09  108.78 119.25
34 34   纪康 112.54   52.26 45.88   98.52   5.64
30 30   顺琦 204.87    8.81 75.81   79.94 143.29
27 27 丁云飞   9.02    9.81 60.37   72.82  66.17
> detach(myTable)
>
```

Window 3 - Source editor:

```
49  attach(myTable)
50  newTable <- myTable[sample(1:50, 10, replace = FALSE), ]
51  newTable
52  detach(myTable)
53  #自定义标准化函数
54 ▾ fun <- function(x){
55    x <- as.numeric(x)
56    return ((x - min(x)) / (max(x) - min(x)))
57  }
58  #进行标准化
59  newTable$Math <- fun(newTable$Math)
60  newTable$English <- fun(newTable$English)
61  newTable$Java <- fun(newTable$Java)
62  newTable$Chinese <- fun(newTable$Chinese)
63  newTable$Sports <- fun(newTable$Sports)
64  #求均值
```

Window 3 - Console:

```
> newTable$Math <- fun(newTable$Math)
> newTable$English <- fun(newTable$English)
> newTable$Java <- fun(newTable$Java)
> newTable$Chinese <- fun(newTable$Chinese)
> newTable$Sports <- fun(newTable$Sports)
> #求均值
> newTable <- transform(newTable, average = apply(newTable[, 3: 7], 1, mean))
> #评分
> newTable
    id   name       Math    English       Java   Chinese     Sports   average
10 10 蔡秋苹 0.06731841 0.51451242 0.64237462 0.06675968 0.08586996 0.2753670
45 45   刘妤 0.71818668 1.00000000 0.21330604 0.93086850 0.97181257 0.7668348
14 14   凌晨 0.51565112 0.05972019 0.49293756 0.98242065 0.57159462 0.5244648
36 36 庞晟宇 0.91117871 0.60952182 0.04503582 1.00000000 0.49364330 0.6118759
3   3 程静文 0.77819060 0.54332846 0.14370522 0.29605860 0.95873592 0.5440038
2   2   成杰 0.49897826 0.70223429 1.00000000 0.70802761 0.61880480 0.5818480
49 49 王婧婧 1.00000000 0.45541867 0.00000000 0.74258807 0.82535416 0.6046722
34 34   纪康 0.24038640 0.90728753 0.28229273 0.67101500 0.00000000 0.4201963
30 30   顺琦 0.45478822 0.00000000 0.89498465 0.54140216 1.00000000 0.5782350
27 27 丁云飞 0.00000000 0.02088119 0.57891505 0.49173352 0.43973847 0.3062536
>
```

```
> #分位点
> quantile <- quantile(newTable$average, c(0.8, 0.6, 0.4, 0.2))
> newTable <- within(newTable, {
+   score <- NA
+   score[average >= quantile["80%"]] <- "优"
+   score[average >= quantile["60%"] & average < quantile["80%"]] <- "良"
+   score[average < quantile["60%"] & average >= quantile["40%"] ] <- "中"
+   score[average < quantile["40%"] & average >= quantile["20%"] ] <- "差"
+   score[average < quantile["20%"]] <- "非常差"
+ })
> newTable
   id   name       Math    English       Java    Chinese     Sports    average  score
10 10 蔡秋苹 0.06731841 0.51451242 0.64237462 0.06675968 0.08586996 0.2753670 非常差
45 45   刘好 0.71818688 1.00000000 0.21330604 0.93086850 0.97181257 0.7668348     优
14 14   凌晨 0.51565112 0.05972019 0.49293756 0.98242065 0.57159462 0.5244648     差
36 36 庞昰宇 0.91117871 0.60952182 0.04503582 1.00000000 0.49364330 0.6118759     优
3   3 程静文 0.77819060 0.54332846 0.14370522 0.29605860 0.95873592 0.5440038     中
2   2   成杰 0.49897826 0.70223429 1.00000000 0.00000000 0.70802761 0.5818480     良
49 49 王婧婧 1.00000000 0.45541867 0.00000000 0.74258807 0.82535416 0.6046722     良
34 34   纪康 0.24038640 0.90728753 0.28229273 0.67101500 0.00000000 0.4201963     差
30 30   顾琦 0.45478822 0.00000000 0.89498465 0.54140216 1.00000000 0.5782350     中
27 27 丁云飞 0.00000000 0.02088119 0.57891505 0.49173352 0.43973847 0.3062536 非常差
```
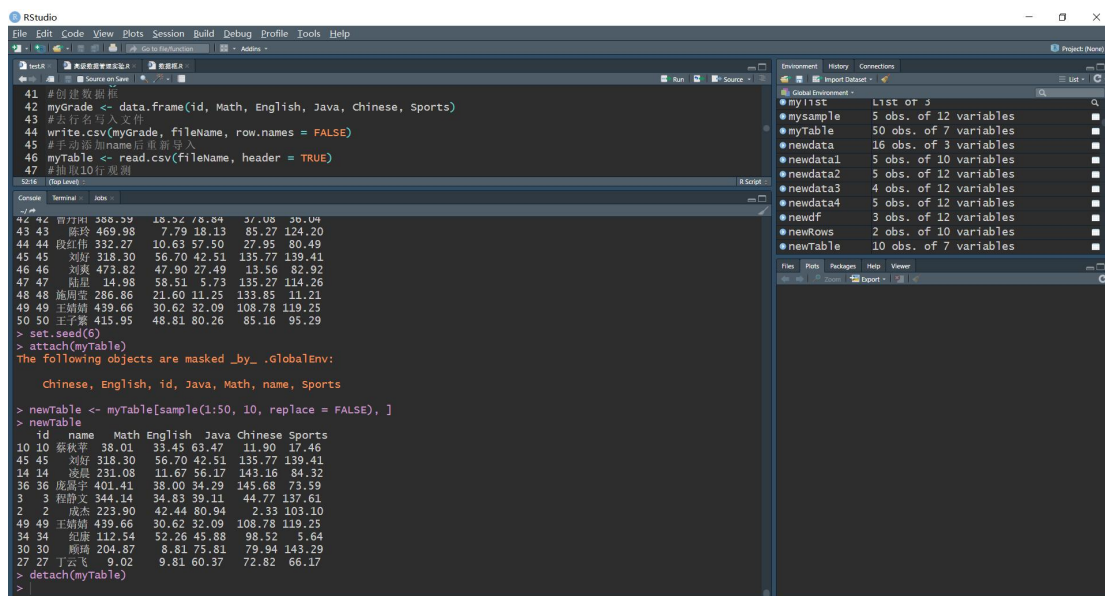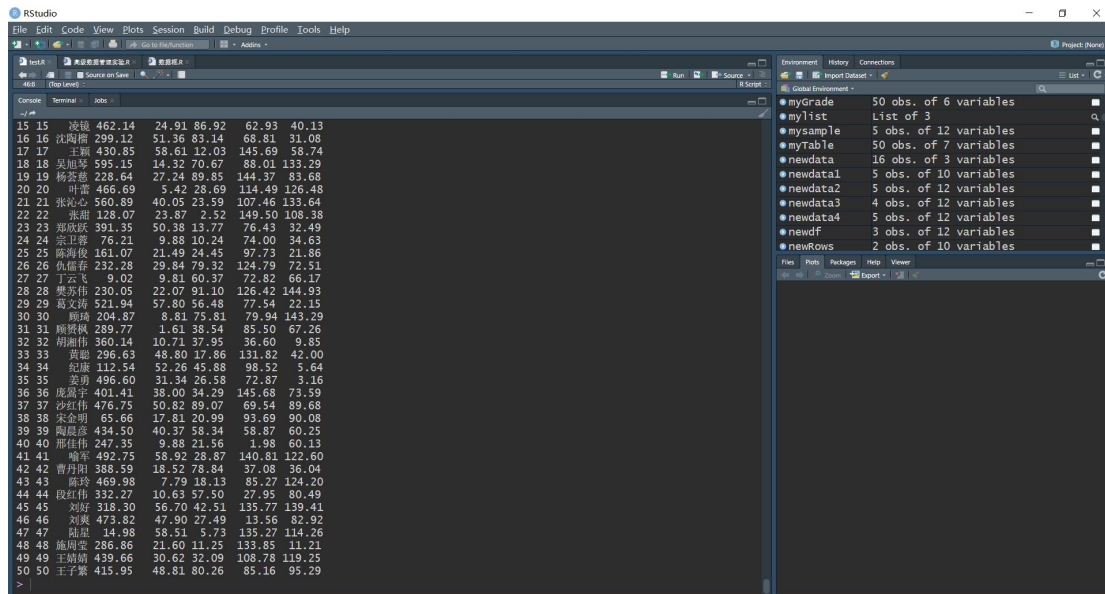
```
> #根据姓排序
> orderByName <- newTable[order(newTable$name), ]
> orderByName
   id   name       Math    English       Java    Chinese     Sports    average  score
10 10 蔡秋苹 0.06731841 0.51451242 0.64237462 0.06675968 0.08586996 0.2753670 非常差
2   2   成杰 0.49897826 0.70223429 1.00000000 0.00000000 0.70802761 0.5818480     良
3   3 程静文 0.77819060 0.54332846 0.14370522 0.29605860 0.95873592 0.5440038     中
27 27 丁云飞 0.00000000 0.02088119 0.57891505 0.49173352 0.43973847 0.3062536 非常差
30 30   顾琦 0.45478822 0.00000000 0.89498465 0.54140216 1.00000000 0.5782350     中
34 34   纪康 0.24038640 0.90728753 0.28229273 0.67101500 0.00000000 0.4201963     差
14 14   凌晨 0.51565112 0.05972019 0.49293756 0.98242065 0.57159462 0.5244648     差
45 45   刘好 0.71818688 1.00000000 0.21330604 0.93086850 0.97181257 0.7668348     优
36 36 庞昰宇 0.91117871 0.60952182 0.04503582 1.00000000 0.49364330 0.6118759     优
49 49 王婧婧 1.00000000 0.45541867 0.00000000 0.74258807 0.82535416 0.6046722     良
```