计算编程在心理学研究中的应用课程作业

谷菲 201928012503005

2020-04-26

目录

1 前言

鉴于老师说过可以使用其它软件，并且我发现对于本次作业来说使用 R 会很方便快捷，尤其是可以用 Rmarkdown 很方便地将代码和结果整合到一个文档中，所以决定使用 R 来完成本次作业。后来我也学习了一下 Unix 的脚本，用 awk 程序和 bash 的循环又完成了一次作业 1，结果和前面是一致的。

2 作业 1

2.1 提取每名被试每个 trial 的时长

提取结果到数据框 log 里，这里展示了一下前 10 行

```
# 读取数据
log <- read.table("log.txt",header = TRUE)
# 计算trial时长放到trial_time列
# trial时长是Text2.Onset-Fixation.Onset+5000，即反应前的加反应阶段的5s
log <- mutate(log,trial_time=Text2.Onset-Fixation.Onset+5000)
```

表 1: 被试每个试次的时长（部分）

Subjectindex	trial_time
1	18000
1	19500
1	18000
1	19500
1	17000
1	19500
1	16000
1	17000
1	18000
1	16000

2.2 计算每名被试在 4 个条件下的正确率

```
# 将反应是否正确转换为数值
log$Corre.Response <- as.numeric(log$Corre.Response)
# 计算fixation的时长
log <- mutate(log,fixation=Text1.Onset-Fixation.Onset)
# 根据fixation时长判断字母长度,3000对应6
log <- mutate(log, word.length=ifelse(log$fixation>2900 &
                  log$fixation<3100, 6, 9))
# 计算正确率
acc <- 100*tapply(log$Corre.Response,
            list(log$Subjectindex,
                 log$Audio.Onset,
                 log$word.length),
            mean)
```

表 2: 被试在四种条件下的正确率（%）

	6 字母-无干扰	6 字母-有干扰	9 字母-无干扰	9 字母-有干扰
1	82.00	76.47	69.23	44.68
2	81.16	77.50	73.08	41.03
3	82.35	65.31	79.31	50.00
4	82.35	51.79	80.49	44.23
5	72.55	66.67	72.92	37.74
6	82.09	60.47	75.51	43.90
7	82.14	72.97	68.85	39.13
8	81.40	59.26	79.17	52.73
9	85.71	61.70	74.14	43.40
10	78.00	56.14	61.54	36.59
11	89.58	70.21	74.51	46.30
12	88.68	82.35	64.29	61.11
13	89.06	65.91	71.43	32.00
14	80.00	57.14	70.00	50.94
15	80.85	70.91	68.00	64.58
16	84.00	65.38	81.25	46.00
17	91.67	77.08	68.42	46.81
18	87.18	58.49	72.73	45.28
19	85.37	54.69	68.63	40.91
20	78.43	67.31	72.55	41.30
21	78.85	62.30	66.67	47.62
22	88.33	56.52	76.60	44.68
23	86.27	73.81	74.58	52.08
24	81.82	73.68	62.50	43.75
25	85.00	54.55	70.97	32.56
26	89.13	57.50	78.85	38.71
27	81.63	69.39	72.73	48.94
28	85.71	60.00	65.31	46.00
29	82.61	80.56	63.33	36.21
30	79.63	60.87	76.00	40.00

2.3 提取 4 个条件下被试的反应时和正确反应的反应时

提取全部反应的反应时，通过 If.Response 等于 TRUE 进行提取，这里展示了提取结果的前 10 行。

```
# 计算反应时
log <- mutate(log,response.time=Response.Onset-Text2.Onset)
# 提取全部有反应的反应时
rt_all <- subset(log,If.Response==TRUE,
                 select = c("Subjectindex",
                            "Audio.Onset",
                            "word.length",
                            "response.time"))
```

表 3: 全部反应的反应时（部分）

Subjectindex	Audio.Onset	word.length	response.time
1	TRUE	9	2615
1	FALSE	6	1492
1	FALSE	9	1721
1	FALSE	9	1839
1	TRUE	6	2611
1	FALSE	9	1855
1	FALSE	9	1964
1	TRUE	6	2609
1	TRUE	9	2602
1	TRUE	9	2500

接着是正确反应的反应时，通过 Corre.Response 等于 1 进行提取，这里展示了提取结果的前 10 行。

```
# 提取正确反应的反应时
rt_correct <- subset(log,Corre.Response==1,
                 select = c("Subjectindex",
                            "Audio.Onset",
                            "word.length",
                            "response.time"))
```

表 4: 正确反应的反应时（部分）

Subjectindex	Audio.Onset	word.length	response.time
1	TRUE	9	2615
1	FALSE	6	1492
1	FALSE	9	1721
1	FALSE	9	1839
1	TRUE	6	2611
1	FALSE	9	1855
1	FALSE	9	1964
1	TRUE	6	2609
1	TRUE	9	2602
1	TRUE	9	2500

2.4 计算 4 个条件下被试的平均反应时和正确反应的平均反应时

计算 4 个条件每名被试所有反应的平均反应时。

```
# 每名被试平均反应时（所有反应的）
meanrt_all <- tapply(rt_all$response.time,
                     list(rt_all$Subjectindex,
                          rt_all$Audio.Onset,
                          rt_all$word.length),
                     mean)

# 所有人的平均
mean_all <- tapply(rt_all$response.time,
                   list(rt_all$Audio.Onset,
                        rt_all$word.length),
                   mean)
```

计算 4 个条件每名被试正确反应的平均反应时。

```
# 平均反应时（正确反应的）
meanrt_correct <- tapply(rt_correct$response.time,
                     list(rt_correct$Subjectindex,
                          rt_correct$Audio.Onset,
                          rt_correct$word.length),
                     mean)
```

```
# 所有人的平均
mean_correct <- tapply(rt_correct$response.time,
                       list(rt_correct$Audio.Onset,
                            rt_correct$word.length),
                       mean)
```

首先将所有被试的平均结果输出，然后是每名被试的平均结果。

表 5: 四种条件下的平均反应时（ms）

	6 字母-无干扰	6 字母-有干扰	9 字母-无干扰	9 字母-有干扰
全部反应	1498.80	2600.90	1902.90	2602.05
正确反应	1499.05	2601.03	1901.29	2601.70

表 6: 被试在四种条件下全部反应的反应时（ms）

	6 字母-无干扰	6 字母-有干扰	9 字母-无干扰	9 字母-有干扰
1	1500.00	2620.90	1902.71	2610.20
2	1509.14	2583.42	1932.94	2578.91
3	1508.59	2619.71	1887.91	2614.86
4	1485.07	2615.65	1899.97	2635.79
5	1472.60	2559.30	1893.30	2600.98
6	1479.45	2602.94	1906.04	2593.89
7	1484.65	2608.71	1925.54	2579.46
8	1526.89	2611.90	1884.27	2639.13
9	1500.24	2580.55	1890.30	2578.25
10	1482.38	2604.98	1876.86	2617.35
11	1486.61	2594.95	1862.60	2590.39
12	1497.17	2601.96	1883.03	2586.33
13	1480.65	2604.97	1937.62	2611.00
14	1500.25	2578.28	1885.77	2624.56
15	1504.33	2593.92	1882.20	2597.89
16	1490.64	2612.46	1931.52	2613.98
17	1497.24	2595.77	1891.88	2576.05
18	1507.68	2585.88	1936.15	2594.43
19	1505.03	2603.80	1921.72	2586.46
20	1498.07	2604.96	1868.98	2595.41
21	1508.36	2622.71	1923.92	2578.68
22	1523.92	2598.07	1914.76	2588.55
23	1486.39	2578.36	1904.21	2583.28
24	1504.51	2589.08	1906.61	2623.88
25	1512.97	2592.41	1916.42	2614.70
26	1488.44	2628.00	1899.83	2619.19
27	1503.49	2599.64	1926.20	2592.36
28	1506.10	2596.25	1888.30	2611.10
29	1515.46	2618.52	1894.13	2597.34
30	1511.49	2615.68	1898.38	2620.07

表 7: 被试在四种条件下正确反应的反应时（ms）

	6 字母-无干扰	6 字母-有干扰	9 字母-无干扰	9 字母-有干扰
1	1500.54	2611.13	1905.81	2591.00
2	1511.21	2576.58	1934.11	2560.56
3	1511.26	2610.38	1889.11	2641.67
4	1484.24	2622.00	1886.85	2635.04
5	1470.51	2555.56	1900.46	2599.25
6	1479.49	2614.92	1899.68	2599.33
7	1482.65	2609.59	1923.14	2582.33
8	1528.06	2612.38	1883.97	2651.55
9	1503.81	2576.07	1891.65	2604.61
10	1483.10	2598.44	1865.38	2613.07
11	1486.33	2597.06	1860.29	2583.20
12	1497.17	2600.83	1865.19	2584.64
13	1476.58	2604.55	1953.27	2625.75
14	1500.93	2582.71	1898.91	2617.00
15	1503.03	2591.15	1883.41	2591.35
16	1487.02	2609.53	1938.31	2630.70
17	1498.23	2584.19	1888.51	2562.55
18	1507.68	2576.61	1935.08	2554.83
19	1505.00	2600.63	1929.06	2567.39
20	1498.97	2608.23	1860.78	2583.84
21	1511.02	2630.42	1905.80	2573.80
22	1523.92	2582.04	1916.14	2599.48
23	1488.98	2582.35	1900.89	2580.28
24	1503.36	2605.40	1904.80	2645.00
25	1512.97	2601.73	1912.48	2660.57
26	1485.98	2633.39	1906.73	2610.42
27	1514.40	2595.94	1925.15	2595.78
28	1504.46	2621.11	1883.38	2605.57
29	1513.63	2618.07	1883.32	2594.81
30	1510.42	2623.07	1895.08	2613.95

3 作业 2

3.1 描述统计

分别计算男、女被试在前、后测中，反应时的最大值、最小值、中位数、均值、标准差

```
# 读取数据
matlab_data <- read.table("matlab_data.txt",header = TRUE)
# 分类统计
analyze <- describeBy(matlab_data[c("Pre","Post")],list(matlab_data$Gender),mat = TRUE)
# 转换一下名称
analyze$vars <- factor(analyze$vars,levels = c(1,2),labels = c("前测","后测"))
analyze$group1 <- factor(analyze$group1,
                         levels = c("Female","Male"),
                         labels = c("女","男"))
# 提取数据
result <- analyze[c("vars","group1","max","min","median","mean","sd")]
```

表 8: 描述统计结果

测试阶段	性别	最大值	最小值	中位数	平均数	标准差
前测	女	360.86	212.93	280.47	280	30
前测	男	354.27	216.16	280.28	280	30
后测	女	335.95	178.21	257.42	260	30
后测	男	367.96	203.86	302.57	300	30

3.2 假设检验

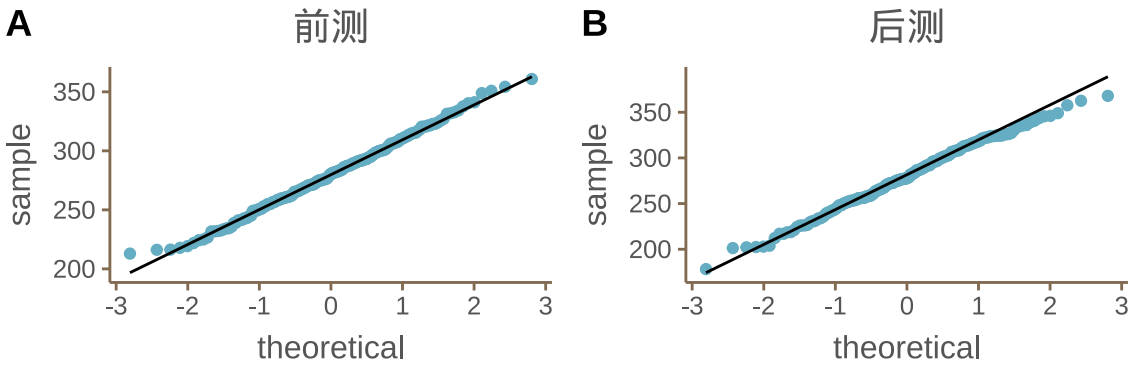
3.2.1 检验是否符合正态分布

通过 Kolmogorov-Smirnov 检验发现前测和后测反应时都是符合正态分布的，Q-Q 图也可以看出是符合的。

```
cat("前测")
# Kolmogorov-Smirnov 检验
ks.test(scale(matlab_data$Pre),pnorm)
# 绘图主题
ggthemr('fresh',layout = "clean")
plot_pre <- ggplot(matlab_data, aes(sample = Pre)) +
  labs(title = "前测")+#设置坐标轴
```

```
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_qq() +
  geom_qq_line()
cat("后测")
# Kolmogorov-Smirnov 检验
ks.test(scale(matlab_data$Post),pnorm)
plot_post <- ggplot(matlab_data, aes(sample = Post)) +
  labs(title = "后测")+#设置坐标轴
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_qq() +
  geom_qq_line()
# 组合两个Q-Q图
ggarrange(plot_pre,plot_post,ncol = 2,labels = c("A","B"))
```

```
## 前测
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(matlab_data$Pre)
## D = 0.026237, p-value = 0.9991
## alternative hypothesis: two-sided
##
## 后测
##  One-sample Kolmogorov-Smirnov test
##
## data:  scale(matlab_data$Post)
## D = 0.037872, p-value = 0.9365
## alternative hypothesis: two-sided
```



3.2.2 前测与后测的反应时是否显著差异

通过配对样本 t 检验发现前测和后测的反应时没有显著差异。

```r
t.test(matlab_data$Pre,matlab_data$Post,paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  matlab_data$Pre and matlab_data$Post
## t = 3.0199e-06, df = 199, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.529772  6.529792
## sample estimates:
## mean of the differences
##                   1e-05
```

3.2.3 前测任务反应时上是否存在显著的性别差异

通过独立样本 t 检验发现前测反应时没有显著的性别差异。

```r
t.test(data=matlab_data,Pre~Gender,var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  Pre by Gender
## t = 0, df = 198, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.366574  8.366574
## sample estimates:
## mean in group Female   mean in group Male
##                  280                  280
```

3.2.4 认知训练的效果是否存在显著的性别差异

计算后测和前测的差值作为训练效果的度量，通过独立样本 t 检验发现认知训练的效果有显著性别差异。

```r
# 计算差值
matlab_data <- mutate(matlab_data,effect=Post-Pre)
t.test(data=matlab_data,effect~Gender,var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  effect by Gender
## t = -6.6667, df = 198, p-value = 2.544e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -51.83212 -28.16788
## sample estimates:
## mean in group Female   mean in group Male
##            -20.00001             19.99999
```

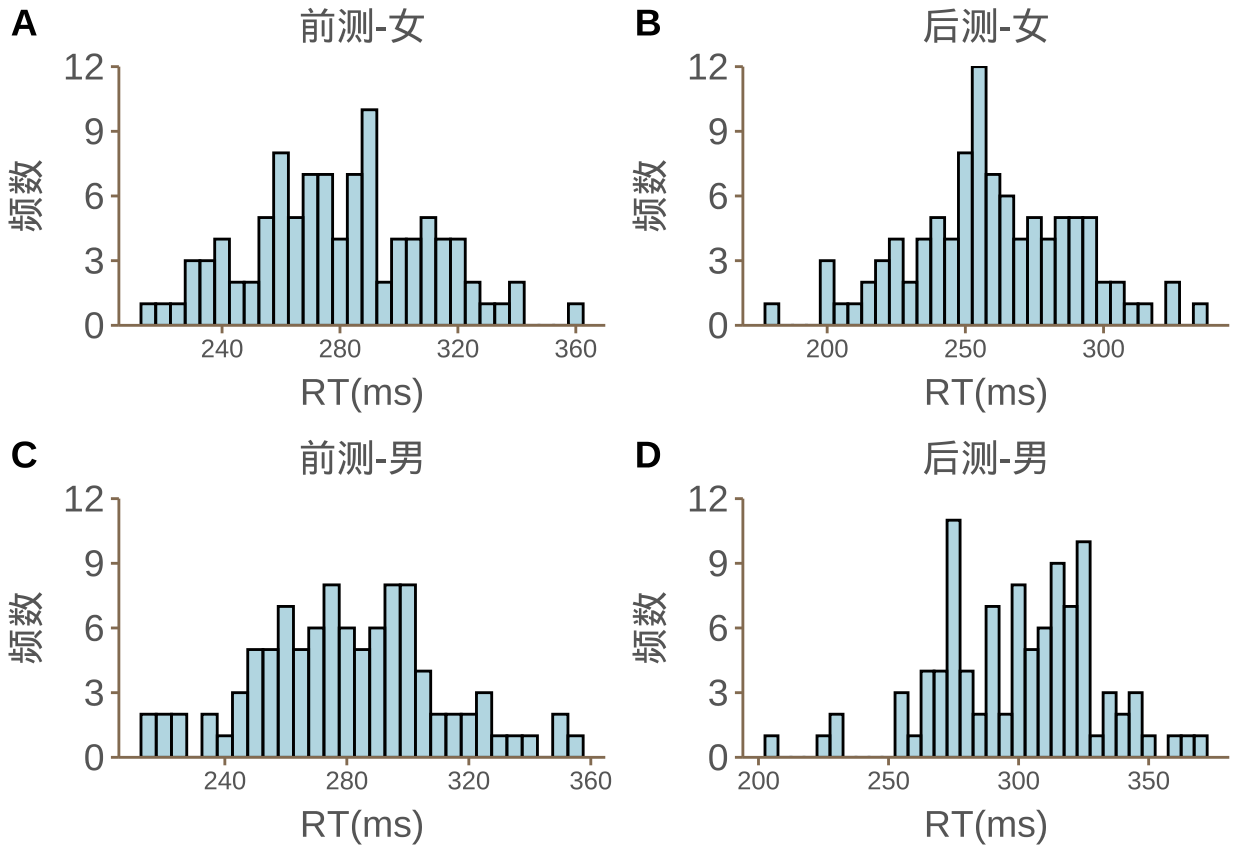
3.3 数据可视化

3.3.1 男女在前后测任务中反应时的直方图

```
# 画图时的字体大小
WORD_SIZE = 14
# 绘图主题
ggthemr('fresh',layout = "clean")
# 转换数据格式
plot_data <- reshape2::melt(matlab_data[1:4],
                            measure.vars=c("Pre","Post"))
plot_data$variable <- factor(plot_data$variable,
                             levels = c("Pre","Post"),
                             labels = c("前测","后测"))
plot_data$Gender <- factor(plot_data$Gender,
                           levels = c("Female","Male"),
                           labels = c("女","男"))
# 循环画图再拼接
i <- 1
figurelist <- list()
for (gen in c("女","男")) {
  for (test in c("前测","后测")){
    # 图的标题
    plottitle=paste0(test,'-',gen)
    # 取出画图的数据
    ploti <- subset(plot_data,Gender==gen&variable==test)
    figurelist[[i]] <- ggplot(ploti, aes(x=value)) +
```

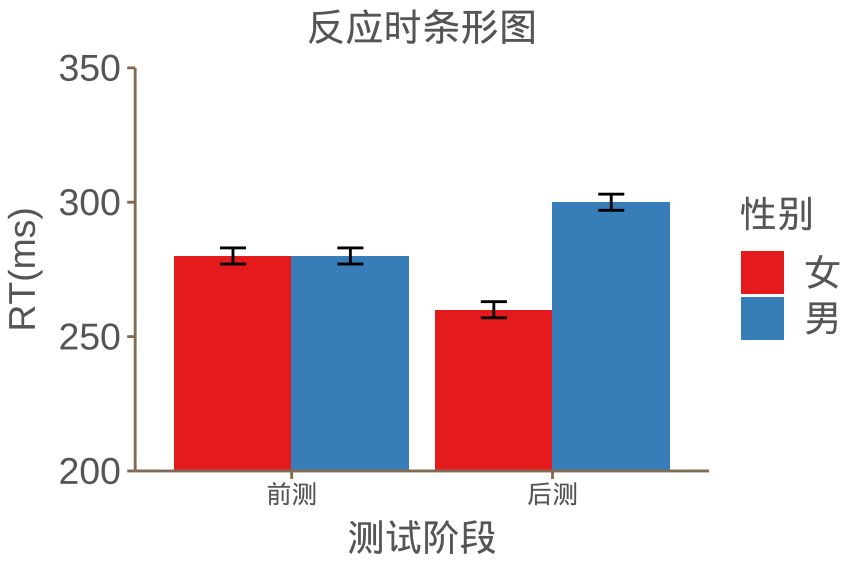
```
      geom_histogram(binwidth=5, alpha=0.5, color="black",position="identity")+
      labs(title = plottitle ,x='RT(ms)',y='频数')+#设置坐标轴
      coord_cartesian(ylim=c(0,12)) +  # 设置y轴坐标范围
      theme(axis.text.y = element_text(size=WORD_SIZE),
            axis.title.x = element_text(size=WORD_SIZE),
            axis.title.y = element_text(size=WORD_SIZE),
            legend.title = element_text(size=WORD_SIZE),
            legend.text = element_text(size=WORD_SIZE),
            plot.title = element_text(hjust = 0.5)) +
      scale_y_continuous(expand = c(0,0))

      i <- i+1
  }
}
# 组合四个图
ggarrange(plotlist = figurelist,nrow = 2,ncol = 2,labels=c("A","B","C","D"))
```



3.3.2 男女在前后测任务中反应时的条形图

```
ggplot(analyze, aes(x=vars, y=mean,fill=group1)) +
  labs(title = "反应时条形图" ,x='测试阶段',y='RT(ms)',fill='性别')+#设置坐标轴
  coord_cartesian(ylim=c(200,350)) +  # 设置y轴坐标范围
  theme(axis.text.y = element_text(size=WORD_SIZE),
        axis.title.x = element_text(size=WORD_SIZE),
        axis.title.y = element_text(size=WORD_SIZE),
        legend.title = element_text(size=WORD_SIZE),
        legend.text = element_text(size=WORD_SIZE),
        plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(expand = c(0,0))+
  geom_bar(position="dodge", stat="identity") +
  scale_fill_brewer(palette = "Set1",direction = 1)+ #颜色
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se),
                width=.2,color='black',       # 误差线
                position=position_dodge(.9))
```



4 作业 1（Unix 脚本）

主要使用了 awk，外加一个 bash 的循环。由于前面已经展示过完整的结果，这部分的结果和之前是一样的，所以就只是输出了结果的前几行。## 提取每名被试每个 trial 的时长

通过 awk 计算，这里展示了结果的前 5 行

```
# 计算trial时长
# trial时长是Text2.Onset-Fixation.Onset+5000，即反应前的加反应阶段的5s
awk 'NR!=1{print $7-$3+5000}' log.txt | head -n 5
```

```
## 18000
## 19500
## 18000
## 19500
## 17000
```

4.1 计算每名被试在 4 个条件下的正确率

```
# 每个被试循环
for sub in $(awk 'NR!=1{print $1}' log.txt | uniq)
do
# "'$var'"使用系统变量,或者"'"$var"'"
awk '$1=="'$sub'"{
  #根据fixation时长判断条件
  if ($4-$3==3000)
  {
    #结果以百分数表示，乘100
    sixcount[$6]+=0.01
    #统计正确的数量
    if ($9=="TRUE")
    {six[$6]+=1}
  }
  else
  {
    ninecount[$6]+=0.01

    if ($9=="TRUE")
    {nine[$6]+=1}
  }
```

```
}
END {

  printf "%d \t %.2f \t %.2f \t %.2f \t %.2f \n", "'$sub'",
  six["FALSE"]/sixcount["FALSE"], six["TRUE"]/sixcount["TRUE"],
  nine["FALSE"]/ninecount["FALSE"], nine["TRUE"]/ninecount["TRUE"]

}' log.txt

done | head -n 5
```

```
## 1     82.00   76.47   69.23   44.68
## 2     81.16   77.50   73.08   41.03
## 3     82.35   65.31   79.31   50.00
## 4     82.35   51.79   80.49   44.23
## 5     72.55   66.67   72.92   37.74
```

4.2 提取 4 个条件下被试的反应时和正确反应的反应时

提取全部反应的反应时，通过 If.Response 等于 TRUE 进行提取，这里展示了提取结果的前 5 行。

```
# 每个被试循环
for sub in $(awk 'NR!=1{print $1}' log.txt | uniq)
do
# "'$var'"使用系统变量,或者"'"$var"'"
awk '$1=="'$sub'"{
  if ($8=="TRUE")
  {
  subject=$1
  audio=$6
  rt=$10-$7
    if ($4-$3==3000)
    {
      word=6
    }
    else
    {
      word=9
```

```
    }
  printf "%d \t %s \t %d \t %d \n", subject,audio,word,rt
  }
}' log.txt

done > rt_all.txt
head -n 5 rt_all.txt
# 打印总的行数
cat rt_all.txt | wc -l
```

```
## 1     TRUE    9   2615
## 1     FALSE   6   1492
## 1     FALSE   9   1721
## 1     FALSE   9   1839
## 1     TRUE    6   2611
## 5306
```

接着是正确反应的反应时，通过 Corre.Response 等于 1 进行提取，这里展示了提取结果的前 5 行。虽然前面 5 个的结果是一样的，但是通过总的行数可以看出来确实去掉了不正确的反应。

```
# 每个被试循环
for sub in $(awk 'NR!=1{print $1}' log.txt | uniq)
do
# "'$var'"使用系统变量,或者"'"$var"'"
awk '$1=="'$sub'"{
  #增加一个条件$9=="TRUE"
  if ($8=="TRUE" && $9=="TRUE")
  {
  subject=$1
  audio=$6
  rt=$10-$7
    if ($4-$3==3000)
    {
      word=6
    }
    else
    {
      word=9
    }
```

```
  printf "%d \t %s \t %d \t %d \n", subject,audio,word,rt
  }
}' log.txt

done > rt_correct.txt
head -n 5 rt_correct.txt
# 打印总的行数
cat rt_correct.txt | wc -l
```

```
## 1     TRUE    9   2615
## 1     FALSE   6   1492
## 1     FALSE   9   1721
## 1     FALSE   9   1839
## 1     TRUE    6   2611
## 3992
```

4.3 计算 4 个条件下被试的平均反应时和正确反应的平均反应时

计算 4 个条件每名被试所有反应的平均反应时。

```
# 每个被试循环
for sub in $(awk '{print $1}' rt_all.txt | uniq)
do
# "'$var'"使用系统变量,或者"'"$var"'"
awk '$1=="'$sub'"{
  #增加一个条件$9=="TRUE"
  rtcount[$3$2]+=1
  rt[$3$2]+=$4
}
END {

  printf "%d \t %.2f \t %.2f \t %.2f \t %.2f \n", "'$sub'",
  rt["6FALSE"]/rtcount["6FALSE"], rt["6TRUE"]/rtcount["6TRUE"],
  rt["9FALSE"]/rtcount["9FALSE"], rt["9TRUE"]/rtcount["9TRUE"]

}' rt_all.txt

done | head -n 5
```

```
## 1     1500.00     2620.90     1902.71     2610.20
```

```
## 2     1509.14     2583.42     1932.94     2578.91
## 3     1508.59     2619.71     1887.91     2614.86
## 4     1485.07     2615.65     1899.97     2635.79
## 5     1472.60     2559.30     1893.30     2600.98
```

计算 4 个条件每名被试正确反应的平均反应时。

```
# 每个被试循环
for sub in $(awk '{print $1}' rt_correct.txt | uniq)
do
# "'$var'"使用系统变量,或者"'"$var"'"
awk '$1=="'$sub'"{
  #增加一个条件$9=="TRUE"
  rtcount[$3$2]+=1
  rt[$3$2]+=$4
}
END {

  printf "%d \t %.2f \t %.2f \t %.2f \t %.2f \n", "'$sub'",
  rt["6FALSE"]/rtcount["6FALSE"], rt["6TRUE"]/rtcount["6TRUE"],
  rt["9FALSE"]/rtcount["9FALSE"], rt["9TRUE"]/rtcount["9TRUE"]

}' rt_correct.txt

done | head -n 5
```

```
## 1     1500.54     2611.13     1905.81     2591.00
## 2     1511.21     2576.58     1934.11     2560.56
## 3     1511.26     2610.38     1889.11     2641.67
## 4     1484.24     2622.00     1886.85     2635.04
## 5     1470.51     2555.56     1900.46     2599.25
```

4.4 计算 4 个条件下被试的平均反应时和正确反应的平均反应时

先计算所有被试 4 个条件所有反应的平均反应时，然后是正确反应的平均反应时。

```
# 所有反应的
awk '{
  #增加一个条件$9=="TRUE"
  rtcount[$3$2]+=1
```

```
  rt[$3$2]+=$4
}
END {

  printf "全部反应 \t %.2f \t %.2f \t %.2f \t %.2f \n",
  rt["6FALSE"]/rtcount["6FALSE"], rt["6TRUE"]/rtcount["6TRUE"],
  rt["9FALSE"]/rtcount["9FALSE"], rt["9TRUE"]/rtcount["9TRUE"]

}' rt_all.txt
# 正确反应的
awk '{
  #增加一个条件$9=="TRUE"
  rtcount[$3$2]+=1
  rt[$3$2]+=$4
}
END {

  printf "正确反应 \t %.2f \t %.2f \t %.2f \t %.2f \n",
  rt["6FALSE"]/rtcount["6FALSE"], rt["6TRUE"]/rtcount["6TRUE"],
  rt["9FALSE"]/rtcount["9FALSE"], rt["9TRUE"]/rtcount["9TRUE"]

}' rt_correct.txt
```

```
## 全部反应      1498.80     2600.90     1902.90     2602.05
## 正确反应      1499.05     2601.03     1901.29     2601.70
```