# 对《Overperception of moral outrage in online social networks inflates beliefs about intergroup hostility》的可重复性研究——代码部分

沈玫霖、乔骊珠、吴惜之、包珺、李慧

2024-06-07

# 研究 1

## 前期准备

```
# 检查是否安装所需要的 R 包，如果没有就下载
packages <- c("tidyverse", "kableExtra", "psych", "Hmisc", "lmerTest")
 lapply(packages, function(pkg) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg, dependencies = TRUE)
  }
})

# 加载 R 包
library(tidyverse)
library(kableExtra)
library(psych)
library(Hmisc)
library(lmerTest)

# 设置工作路径
setwd("C:/Users/sml/Desktop/R/大作业/复刻代码")

# 读取文件
self_report <- read_csv("../osfstorage-archive/Data/study1_self_report.csv")
data <- read_csv("../osfstorage-archive/Data/study1_data_raw.csv")
```

```
# 函数 1: frequencies.table() 的作用是计算变量的频数、百分比，并将结果以美观的表格呈现
frequencies.table <- function(variable, label) {
  freq <- table(variable)
  prop <- prop.table(table(variable))
  perc <- prop*100
  combined <- cbind(freq, perc)
  kable((combined), format = "latex",col.names = c("Freq", "%"), digits = 2) %>%
    kable_styling(bootstrap_options = "striped",full_width = FALSE, position = "left")
}

# 函数 2: corr.matrix() 的作用是计算变量之间的相关系数，评估显著性，呈现在表格里
corr.matrix <- function(x){
  x <- as.matrix(x)
```

```
  R <- Hmisc::rcorr(x)$r
  p <- Hmisc::rcorr(x)$P
  mystars <- ifelse(p < .001, "***",
                    ifelse(p < .01, "** ", ifelse(p < .05, "* ", " ")))
  R <- format(round(cbind(rep(-1.11, ncol(x)), R), 2))[,-1]
  Rnew <- matrix(paste(R, mystars, sep=""), ncol=ncol(x))
  diag(Rnew) <- paste(diag(R), " ", sep="")
  rownames(Rnew) <- colnames(x)
  colnames(Rnew) <- paste(colnames(x), "", sep="")
  Rnew <- as.matrix(Rnew)
  Rnew[upper.tri(Rnew, diag = TRUE)] <- ""
  Rnew <- as.data.frame(Rnew)
  Rnew <- cbind(Rnew[1:length(Rnew)-1])
  return(Rnew)
}

# 函数 3: numextract() 的作用是从字符串中提取数字
numextract <- function(string){
  str_extract(string, "\\-*\\d+\\.*\\d*")
}
```

### 数据处理

```
# 添加性别标签的变量 gender_label 到 data 中
data <- data %>% mutate(gender_label =
                          ifelse(gender == 1, "Male",
                                 ifelse(gender == 2, "Female", "Other")))

# 添加党派标签 party_label 的变量到 data 中
data <- data %>% mutate(
  party_label =
    ifelse(party == 1, "Democrat",
           ifelse(party == 2, "Republican",
                  ifelse(party == 3, "Indepedent",
                         ifelse(party == 4, "Other","None")))))

# 添加党派认同程度的变量 p_identity 到 data 中
# 若 sis_dem 缺失，则使用 sis_rep 的数据
```

```r
data <- data %>% mutate(p_identity = ifelse(is.na(sis_dem) == TRUE,
                                            sis_rep, sis_dem))

# 添加意识形态极端程度 ideo_extr 的变量到 data 中，使用的是绝对值
data <- data %>% mutate(ideo_extr = abs(ideo))
```

```r
# 注意力检测的频数表
frequencies.table(data$comp_check)
```

| Freq | % |
|---|---|
| 133 | 95.00 |
| 6 | 4.29 |
| 1 | 0.71 |

```r
# 剔除未通过注意力检测的观察者
data_trim <- data %>% filter(comp_check == 1)

# 剔除党派类别除"Democrat" 和"Republican" 之外的观察者
data_trim <- data_trim %>% filter(party_label == "Democrat" |
                                  party_label == "Republican")
```

### 描述性统计

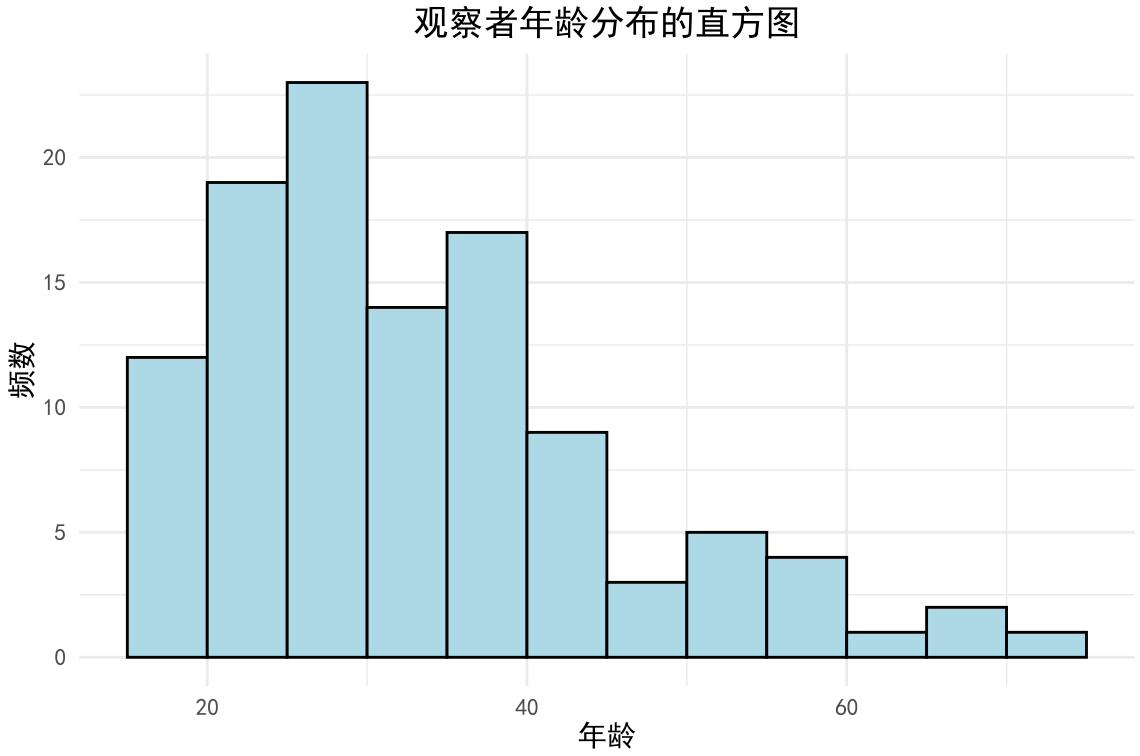
```r
# 观察者性别频数分布表
frequencies.table(data_trim$gender_label)
```

| | Freq | % |
|---|---|---|
| Female | 53 | 48.18 |
| Male | 56 | 50.91 |
| Other | 1 | 0.91 |

```r
# 观察者年龄的直方图
hist_data <- hist(data_trim$age,
                  plot = FALSE)

plot(hist_data,
     main = " 观察者年龄分布的直方图",
     xlab = " 年龄",
     ylab = " 频数",
```

```
     col = "lightblue",
     border = "black")
```

```
ggplot(data_trim, aes(x = age)) +
  geom_histogram(breaks = hist_data$breaks,
                 fill = "lightblue",
                 color = "black") +
  labs(title = " 观察者年龄分布的直方图",
       x = " 年龄",
       y = " 频数") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    text = element_text(family = "SimHei")
  )
```




```
# 观察者意识形态频数分布表
frequencies.table(data_trim$ideo)
```

| | Freq | % |
|---|---|---|
| -3 | 20 | 18.18 |
| -2 | 18 | 16.36 |
| -1 | 21 | 19.09 |
| 0 | 8 | 7.27 |
| 1 | 23 | 20.91 |
| 2 | 10 | 9.09 |
| 3 | 10 | 9.09 |

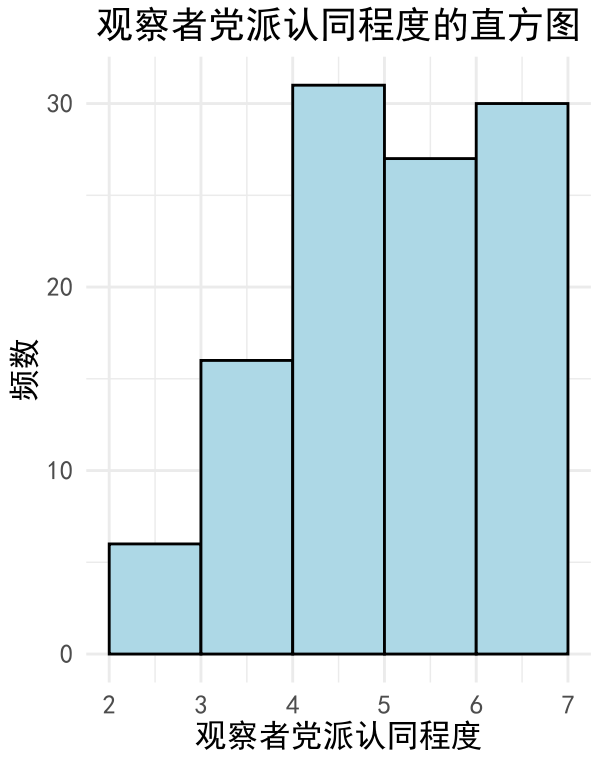
```r
# 观察者党派频数分布表
frequencies.table(data_trim$party_label)
```

| | Freq | % |
|---|---|---|
| Democrat | 53 | 48.18 |
| Republican | 57 | 51.82 |

```r
# 观察者党派认同程度的直方图和平均值
hist_data <- hist(data_trim$p_identity,
                  plot = FALSE)

plot(hist_data,
     main = " 观察者党派认同程度的直方图",
     xlab = " 观察者党派认同程度",
     ylab = " 频数",
     col = "lightblue",
     border = "black")
```

```r
ggplot(data_trim, aes(x = p_identity)) +
  geom_histogram(binwidth = 1,
                 fill = "lightblue",
                 color = "black",
                 boundary = 2) +  # 确保柱子连在一起
  labs(title = " 观察者党派认同程度的直方图",
       x = " 观察者党派认同程度",
       y = " 频数") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
```

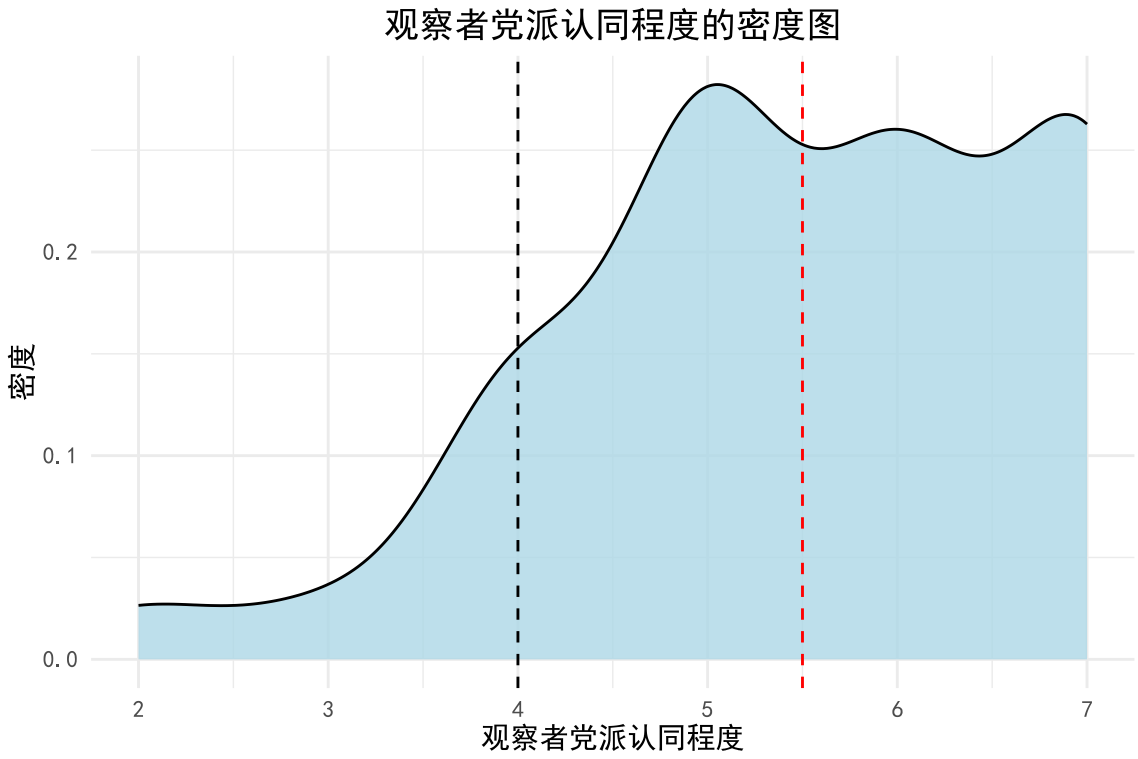
```
    text = element_text(family = "SimHei")
  ) +
  coord_fixed(ratio = 1/5)
```




```
round(mean(data_trim$p_identity, na.rm = TRUE),2)
```

```
[1] 5.51
```

```
# 观察者党派认同程度的密度图
data_trim %>%
  ggplot(aes(x = p_identity)) +
  geom_density(fill = "lightblue", alpha = .8) +
  labs(title = " 观察者党派认同程度的密度图", x = " 观察者党派认同程度",
       y = " 密度") +
  geom_vline(xintercept = 5.5, linetype = "dashed", color = "red") +
  geom_vline(xintercept = 4, linetype = "dashed") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    text = element_text(family = "SimHei")
  )
```




过度感知的分析

愤怒部分

```
# 选择 data_trim 表格中”or“结尾的列，代表与愤怒情绪 (or: outrage) 相关的评分
data_or_trans <- data_trim %>% select(ends_with("_or")) %>%
  t() %>% # 转置数据框，使得每一行代表一个推文的评分
  as.data.frame() %>% # 将矩阵转换为数据框
  mutate(tweet_id = as.integer(numextract(rownames(.))),
         tweet_id_char = rownames(.)) %>%
  # 使用 numextract 函数从行名中提取数字，创建新的推文 ID 列
  arrange(tweet_id)# 根据推文 ID 进行排列

# 计算每个推文被感知的平均愤怒程度
data_or_trans$mean_or <- data_or_trans %>%
  select(-c(tweet_id, tweet_id_char)) %>% rowMeans(., na.rm = TRUE)

# 引入自我报告评分进行比较，根据 tweet_id 进行合并
data_or_trans <- data_or_trans %>% left_join(self_report,
                                             by = "tweet_id")
```

```r
# 选择数据框 data_or_trans 中的列 V1 到 V110，以及 tweet_id 和 sr_outrage
data_mlm <- data_or_trans %>% select(V1:V110, tweet_id, sr_outrage) %>%
  pivot_longer(cols = -c(tweet_id, sr_outrage),
               values_to = "judgment",
               names_to = "pid",
               values_drop_na = TRUE)

# 为数据框 data_mlm 添加一个新列 name，并将其值设为"perceiver"（观察者）
data_mlm <- data_mlm %>% mutate(name = "perceiver")

# 从 self_report 数据框中选择 tweet_id 和 sr_outrage 列
# 为每一行添加一个 pid，其值为 tweet_id 的行号
self_report_trim <- self_report %>% select(tweet_id, sr_outrage) %>%
  mutate(pid = as.character(row_number(tweet_id)))

# 选择 tweet_id 列，然后添加一个新列 name 设为"author"
# 并与 self_report_trim 数据框进行左连接，再将 judgment 列的值设为 sr_outrage
data_mlm2 <- data_mlm %>% select(tweet_id) %>%
  mutate(name = "author") %>% left_join(self_report_trim,
                                        by = "tweet_id") %>%
  mutate(judgment = sr_outrage)

# 创建一个列的顺序顺序
col_order <- c("tweet_id", "sr_outrage", "pid",
               "judgment", "name")

# 根据列的顺序重新排列 data_mlm2 数据框
data_mlm2 <- data_mlm2[, col_order]

# 将 data_mlm 和 data_mlm2 合并，按 tweet_id 排序
# 添加一个新列 name_dum，其值为如果 name 为"author" 则为 0，否则为 1
data_mlm3 <- rbind(data_mlm, data_mlm2) %>% arrange(tweet_id) %>%
  mutate(name_dum = ifelse(name == "author", 0, 1))

# 设置 options 以控制科学计数法的显示，scipen=999 表示不使用科学计数法
options(scipen = 999)
```

```r
# 多层次模型固定效应部分包括变量 name，随机效应包括 tweet_id 和 pid 的随机截距
model_or <- lmer(
  judgment ~ name + (1 | tweet_id) + (1 | pid),
  data = data_mlm3
)
summary(model_or)
```

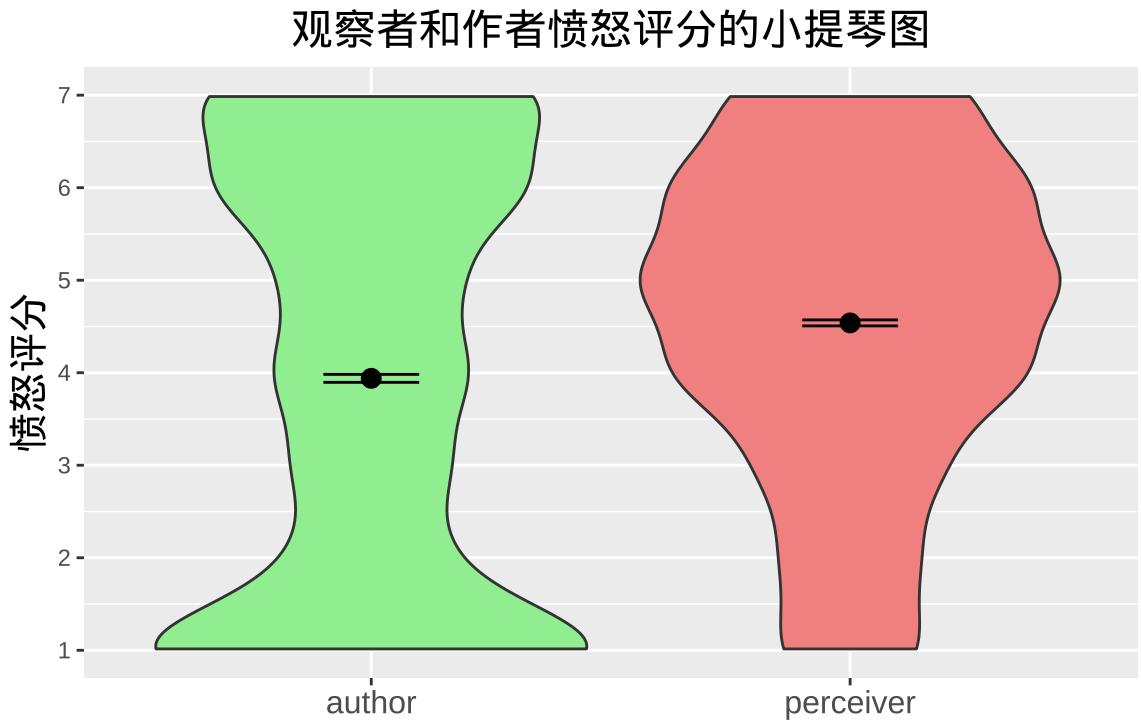
```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: judgment ~ name + (1 | tweet_id) + (1 | pid)
   Data: data_mlm3

REML criterion at convergence: 16402.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.4545 -0.0507  0.0042  0.0512  5.3369

Random effects:
 Groups   Name        Variance Std.Dev.
 pid      (Intercept) 3.1632   1.7785
 tweet_id (Intercept) 0.8944   0.9457
 Residual             0.7511   0.8666
Number of obs: 5822, groups:  pid, 243; tweet_id, 133

Fixed effects:
              Estimate Std. Error       df t value            Pr(>|t|)
(Intercept)     3.9548     0.1754 332.7316  22.546 <0.0000000000000002 ***
nameperceiver   0.5913     0.2308 235.2578   2.562               0.011 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
namepercevr -0.594
```

```r
round(confint(model_or, level = 0.95),2)
```

```
Computing profile confidence intervals ...

              2.5 % 97.5 %
```

```
.sig01          1.62   1.95
.sig02          0.84   1.08
.sigma          0.85   0.88
(Intercept)     3.61   4.30
nameperceiver   0.14   1.04
```

```r
# 观察者和作者愤怒评分的小提琴图
p_or <- ggplot(data_mlm3, aes(x = name, y = judgment, fill = name)) +
  geom_violin(trim = FALSE, bw = 0.5) +
  geom_point(stat = "summary", fun = "mean", color = "black", size = 3) +
  geom_errorbar(stat = "summary", fun.data = mean_se, width = 0.2) +
  scale_fill_manual(values = c("author" = "lightgreen",
                               "perceiver" = "lightcoral")) +
  scale_y_continuous(breaks = 1:7, limits = c(1, 7)) +
  labs(x = "", y = " 愤怒评分", title = " 观察者和作者愤怒评分的小提琴图") +
  theme(
    legend.position = "none",
    axis.title.y = element_text(size = 15),
    axis.text.x = element_text(size = 12),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
print(p_or)
```




快乐部分 (同愤怒部分)

```
data_hap_trans <- data_trim %>% select(ends_with("_hap")) %>%
  t() %>%
  as.data.frame() %>%
  mutate(tweet_id = as.integer(numextract(rownames(.))),
         tweet_id_char = rownames(.)) %>%
  arrange(tweet_id)

data_hap_trans$mean_hap <- data_hap_trans %>%
  select(-c(tweet_id, tweet_id_char)) %>% rowMeans(., na.rm = TRUE)

data_hap_trans <- data_hap_trans %>% left_join(self_report,
                                               by = "tweet_id")

data_mlmh <- data_hap_trans %>% select(V1:V110, tweet_id, sr_happy) %>%
  pivot_longer(cols = -c(tweet_id, sr_happy),
               values_to = "judgment",
               names_to = "pid",
               values_drop_na = TRUE)
```

```r
data_mlmh <- data_mlmh %>% mutate(name = "perceiver")


self_report_trimh <- self_report %>% select(tweet_id, sr_happy) %>%
  mutate(pid = as.character(row_number(tweet_id)))


data_mlm2h <- data_mlmh %>% select(tweet_id) %>%
  mutate(name = "author") %>% left_join(self_report_trimh,
                                        by = "tweet_id") %>%
  mutate(judgment = sr_happy)


col_orderh <- c("tweet_id", "sr_happy", "pid",
                "judgment", "name")

data_mlm2h <- data_mlm2h[, col_orderh]


data_mlm3h <- rbind(data_mlmh, data_mlm2h) %>% arrange(tweet_id) %>%
  mutate(name_dum = ifelse(name == "author", 0, 1))


options(scipen = 999)
```

```r
model_hap<- lmer(judgment ~ name + (1 | tweet_id) + (1 |pid),
                 data = data_mlm3h)
summary(model_hap)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: judgment ~ name + (1 | tweet_id) + (1 | pid)
   Data: data_mlm3h

REML criterion at convergence: 15379.5

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.9962 -0.0923 -0.0112  0.0416  6.5457
```

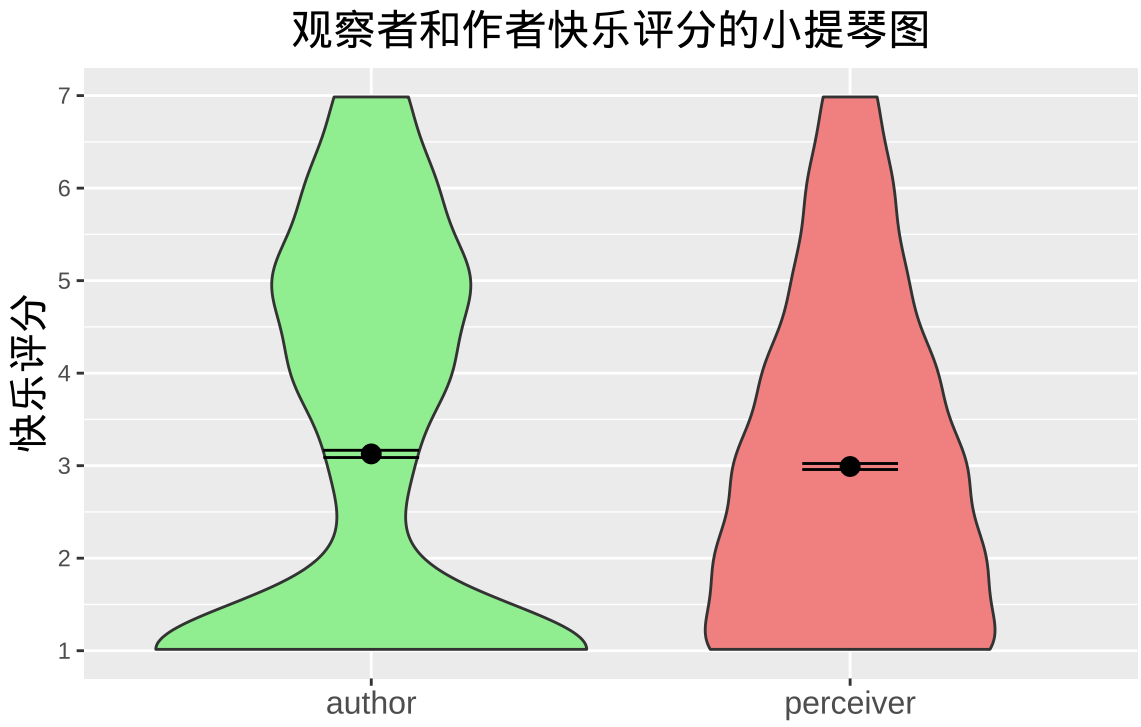
```
Random effects:
 Groups   Name        Variance Std.Dev.
 pid      (Intercept) 2.6483   1.6274
 tweet_id (Intercept) 0.6438   0.8024
 Residual             0.6322   0.7951
Number of obs: 5822, groups:  pid, 243; tweet_id, 133

Fixed effects:
              Estimate Std. Error       df t value            Pr(>|t|)
(Intercept)     3.1278     0.1580 325.5464  19.794 <0.0000000000000002 ***
nameperceiver  -0.1301     0.2111 236.2927  -0.616               0.538
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
namepercevr -0.603
```

```r
round(confint(model_hap, level = 0.95),2)
```

```
Computing profile confidence intervals ...

              2.5 % 97.5 %
.sig01         1.48   1.78
.sig02         0.71   0.91
.sigma         0.78   0.81
(Intercept)    2.82   3.44
nameperceiver -0.54   0.28
```

```r
p_hap <- ggplot(data_mlm3h, aes(x = name, y = judgment, fill = name)) +
  geom_violin(trim = FALSE, bw = 0.5) +
  geom_point(stat = "summary", fun = "mean", color = "black", size = 3) +
  geom_errorbar(stat = "summary", fun.data = mean_se, width = 0.2) +
  scale_fill_manual(values = c("author" = "lightgreen", "perceiver" = "lightcoral")) +
  scale_y_continuous(breaks = 1:7, limits = c(1, 7)) +
  labs(x = "", y = " 快乐评分", title = " 观察者和作者快乐评分的小提琴图") +
  theme(
    legend.position = "none",
    axis.title.y = element_text(size = 15),
    axis.text.x = element_text(size = 12),
```

```
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
print(p_hap)
```




## 研究 2

### 数据预处理

```
# 读取文件储存在变量中
self_report_study2 <- read_csv("../osfstorage-archive/Data/study2_self_report.csv")
data_study2 <- read_csv("../osfstorage-archive/Data/study2_data_raw.csv")

# 数据预处理
# 性别标签（age）
data_study2 <- data_study2 %>% mutate(
  gender_label = ifelse(gender == 1, "Male",
                        ifelse(gender == 2, "Female", "Other")))
# 党派标签（party）
data_study2 <- data_study2 %>%
```

```r
  mutate(party_label =
           ifelse(party == 1, "Democrat",
                  ifelse(party == 2, "Republican",
                         ifelse(party == 3, "Indepedent",
                                ifelse(party == 4, "Other", "None")))))

# 使用 mutate() 和 ifelse() 函数为数据框 data 添加一个新列 p_identity
# 如果 sis_dem 列的值是 NA，则取 sis_rep 列的值；否则取 sis_dem 列的值
data_study2 <- data_study2 %>%
  mutate(p_identity = ifelse(is.na(sis_dem) == TRUE, sis_rep, sis_dem))

# 使用 mutate() 和 abs() 函数为数据框 data 添加一个新列 ideo_extr
# 计算 ideo 列的绝对值
data_study2 <- data_study2 %>% mutate(ideo_extr = abs(ideo))

# 使用之前定义的 frequencies.table 函数来查看 comp_check 列的分布情况
frequencies.table(data_study2$comp_check)
```

| Freq | % |
|---|---|
| 181 | 95.77 |
| 3 | 1.59 |
| 2 | 1.06 |
| 3 | 1.59 |

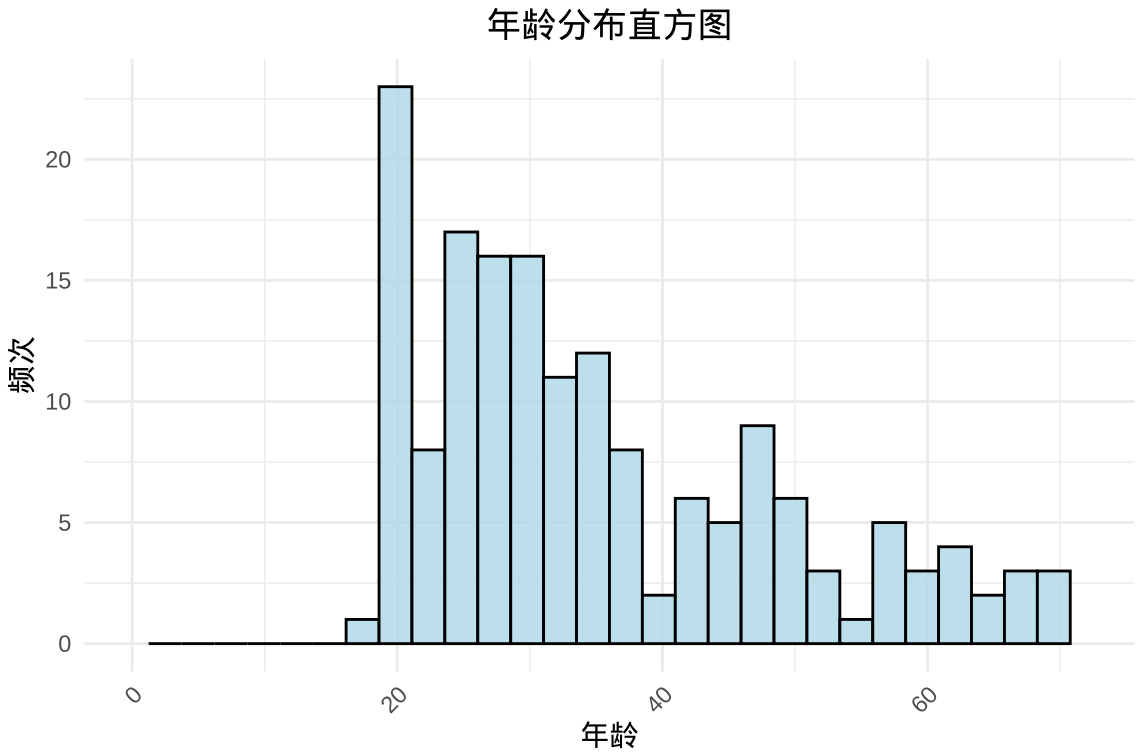
```r
# 使用 filter() 函数从 data 中移除 comp_check 列值为非 1 的行，即移除未通过理解检查的行
data_trim_study2 <- data_study2 %>% filter(comp_check == 1)

# 剔除 Democrat.Republican 外的党派
data_trim_study2 <- data_trim_study2 %>%
  filter(party_label == "Democrat" | party_label == "Republican")
```

### 描述性统计

```r
# 使用 frequencies.table 函数查看 data_trim 中 gender_label 列的分布情况
frequencies.table(data_trim_study2$gender_label)
```

| | Freq | % |
|---|---|---|
| Female | 79 | 47.88 |
| Male | 86 | 52.12 |

```r
# 使用 ggplot() 函数为 data_trim 中 age 列的值绘制直方图
ggplot(data_trim_study2, aes(x = age)) +
  geom_histogram(bins = 30, fill = "lightblue", alpha = 0.8, color = "black") +
  labs(x = " 年龄", y = " 频次", title = " 年龄分布直方图") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_continuous(limits = c(0, max(data_trim_study2$age))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



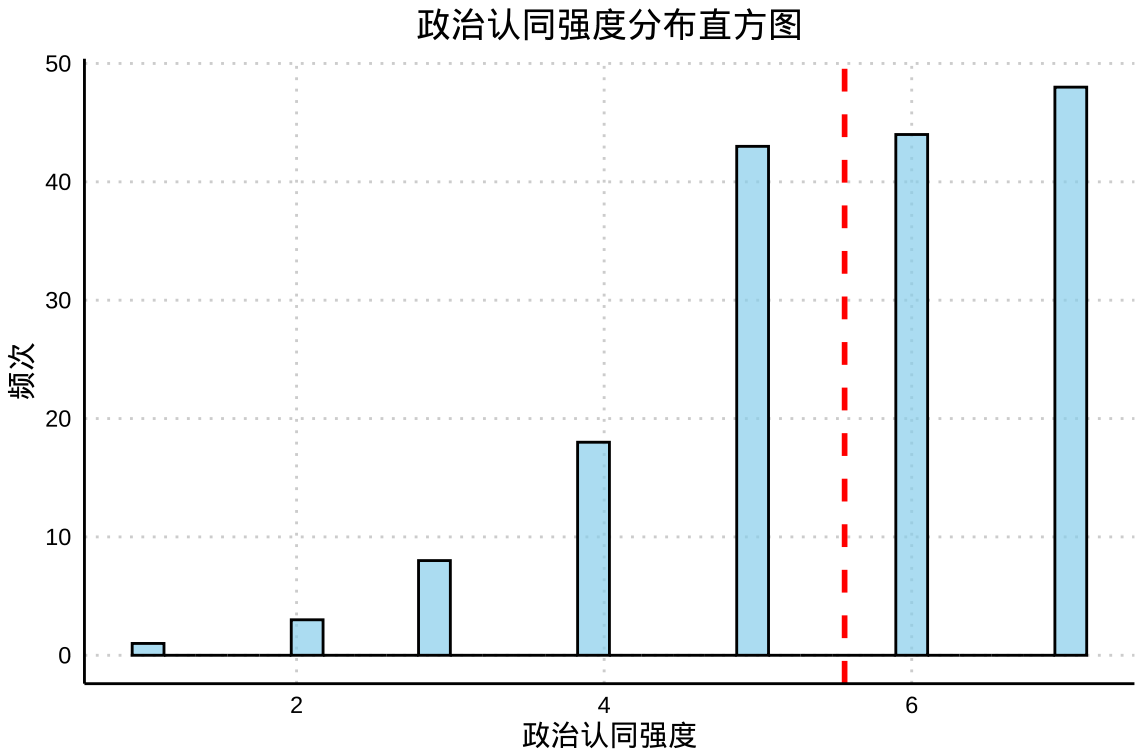

```r
# 使用 frequencies.table 函数查看 data_trim 中 ideo 列的分布情况
frequencies.table(data_trim_study2$ideo)
```

| | Freq | % |
|---|---|---|
| -3 | 26 | 15.76 |
| -2 | 36 | 21.82 |
| -1 | 25 | 15.15 |
| 0 | 7 | 4.24 |
| 1 | 39 | 23.64 |
| 2 | 22 | 13.33 |
| 3 | 10 | 6.06 |

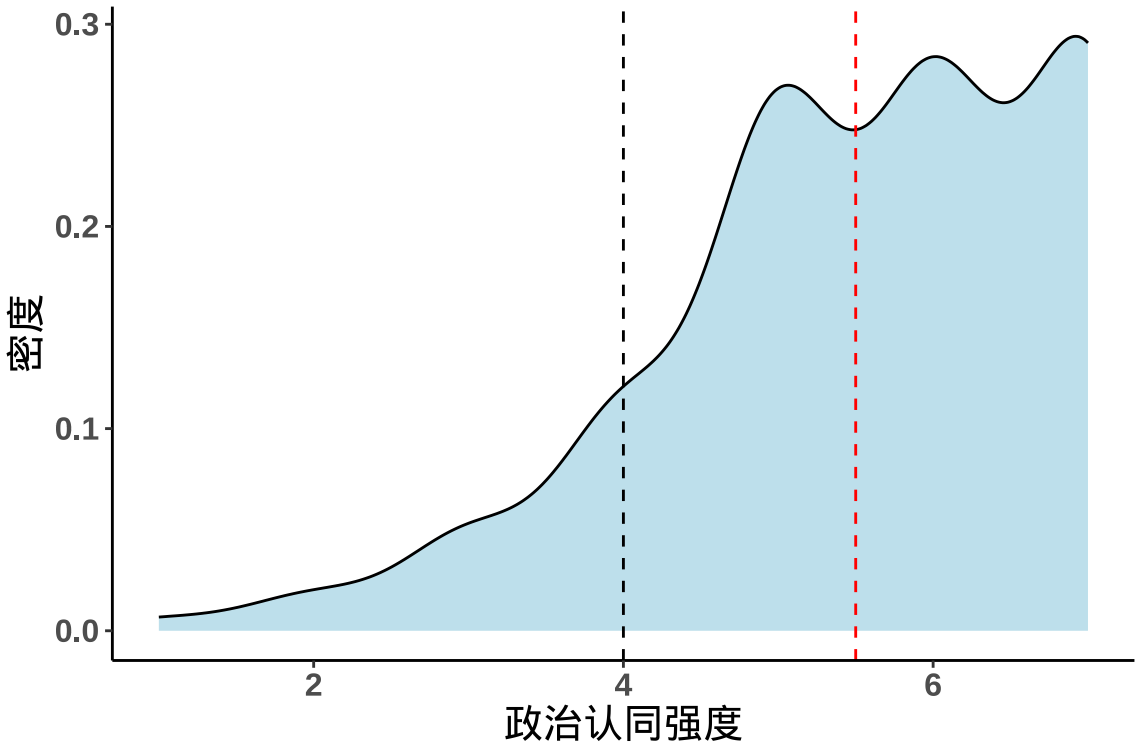
```r
# 使用 frequencies.table 函数查看 data_trim 中 party_label 列的分布情况
frequencies.table(data_trim_study2$party_label)
```

| | Freq | % |
|---|---|---|
| Democrat | 80 | 48.48 |
| Republican | 85 | 51.52 |

```r
#p_identity 列的值绘制直方图并计算均值
mean_p_identity_study2 <- mean(data_trim_study2$p_identity, na.rm = TRUE)

ggplot(data_trim_study2, aes(x = p_identity)) +
  geom_histogram(bins = 30, fill = "skyblue", alpha = 0.7, color = "black") + # 绘制直方图
  geom_vline(aes(
    xintercept = mean_p_identity_study2),
    color = "red", linetype = "dashed", size = 1) + # 添加表示均值的虚线
  labs(x = " 政治认同强度", y = " 频次",
       title = " 政治认同强度分布直方图") + # 设置轴标签和图标题
  theme_minimal() + # 使用简洁主题
  theme(
    plot.title = element_text(hjust = 0.5), # 居中图标题
    axis.title.x = element_text(face = "bold"), # 粗体 X 轴标题
    axis.title.y = element_text(face = "bold"), # 粗体 Y 轴标题
    axis.text = element_text(color = "black"), # 轴文本颜色
    axis.line = element_line(color = "black"), # 轴线颜色
    panel.grid.major = element_line(
      color = "grey80", linetype = "dotted"), # 主网格线样式
    panel.grid.minor = element_blank(), # 移除次网格线
    panel.border = element_blank() # 移除面板边框
  ) +
  theme(panel.grid = element_blank()) # 移除面板网格线
```




```
# 使用 ggplot2 包绘制 p_identity 列的密度图，并添加一些额外的可视化元素
data_trim_study2 %>% ggplot(aes(x = p_identity)) +
  geom_density(fill = "lightblue", alpha = .8) +
  xlab(" 政治认同强度") +
  ylab(" 密度") +
  geom_vline(xintercept = 5.5,
             linetype = "dashed", color = "red") +  # 添加虚线表示特定值
  geom_vline(xintercept = 4, linetype = "dashed") +
  theme_bw() +  # 设置主题为黑白，并移除边框和轴线
  theme(panel.border = element_blank(), axis.line = element_line()) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) + # 移除主要和次要的网格线
  theme(text=element_text(size = 15, face = 'bold'))  # 设置文本大小和粗体
```




主要的过度感知分析

```
# 选择数据框 data_trim 中以"_or" 结尾的列，这些列代表推特级别的评分
# 然后转置数据，转换为数据框，并按推文 ID 排序
data_or_trans_study2 <- data_trim_study2 %>% select(ends_with("_or")) %>%
  t() %>%
  as.data.frame() %>%
  mutate(tweet_id = as.integer(numextract(rownames(.))), tweet_id_char = rownames(.)) %>%
  arrange(tweet_id)

# 对于幸福感 ("_hap") 的评分，执行与愤怒 ("_or") 相同的操作
data_hap_trans_study2 <- data_trim_study2 %>% select(ends_with("_hap")) %>%
  t() %>%
  as.data.frame() %>%
  mutate(tweet_id = as.integer(numextract(rownames(.))), tweet_id_char = rownames(.)) %>%
  arrange(tweet_id)

# 计算每个推文被感知的愤怒的平均值
data_or_trans_study2$mean_or <- data_or_trans_study2 %>%
  select(-c(tweet_id, tweet_id_char)) %>% rowMeans(., na.rm = TRUE)
```

```r
# 计算每个推文被感知的幸福感的平均值
data_hap_trans_study2$mean_hap <- data_hap_trans_study2 %>%
  select(-c(tweet_id, tweet_id_char)) %>% rowMeans(., na.rm = TRUE)

# 将自我报告评分 (self_report) 通过推文 ID 与之前的数据合并，以便进行比较
data_or_trans_study2 <- data_or_trans_study2 %>% left_join(self_report_study2, by = "tweet_id")
data_hap_trans_study2 <- data_hap_trans_study2 %>% left_join(self_report_study2, by = "tweet_id")

# 针对愤怒情绪进行多层次模型分析（MLM）
data_mlm_study2 <- data_or_trans_study2 %>% select(V1:V165, tweet_id, sr_outrage) %>%
  pivot_longer(cols = -c(tweet_id, sr_outrage),
               values_to = "judgment",
               names_to = "pid",
               values_drop_na = TRUE)
# 添加一个表示评判者是观察者还是作者的变量
data_mlm_study2 <- data_mlm_study2 %>% mutate(name = "perceiver")
# 准备自我报告数据，以便与 MLM 数据合并
self_report_trim_study2 <- self_report_study2 %>% select(tweet_id, sr_outrage) %>%
  mutate(pid = as.character(row_number(tweet_id)))
# 将自我报告数据添加到 MLM 数据中，并将评判值设置为自我报告的愤怒评分
data_mlm2_study2 <- data_mlm_study2 %>% select(tweet_id) %>%
  mutate(name = "author") %>% left_join(self_report_trim_study2, by = "tweet_id") %>%
  mutate(judgment = sr_outrage)
# 指定列的顺序
col_order_study2 <- c("tweet_id", "sr_outrage", "pid",
               "judgment", "name")
# 按指定的列顺序对数据 _mlm2 进行重排
data_mlm2_study2 <- data_mlm2_study2[, col_order]
# 合并观察者和作者的数据，并添加一个虚拟变量来区分两者
data_mlm3_study2 <- rbind(data_mlm_study2, data_mlm2_study2) %>% arrange(tweet_id) %>%
  mutate(name_dum = ifelse(name == "author", 0, 1))
# 设置 R 的选项，以控制科学记数法的显示
options(scipen = 999)

# 拟合愤怒情绪的多层次模型
model_or_study2 <- lmer(judgment ~ name + (1 | tweet_id) + (1 |pid), data = data_mlm3_study2)
# 获取模型摘要
summary(model_or_study2)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: judgment ~ name + (1 | tweet_id) + (1 | pid)
   Data: data_mlm3_study2

REML criterion at convergence: 26517.3

Scaled residuals:
    Min      1Q  Median      3Q     Max
-6.0978 -0.0484  0.0018  0.0788  5.8834

Random effects:
 Groups   Name        Variance Std.Dev.
 pid      (Intercept) 2.5762   1.605
 tweet_id (Intercept) 1.4697   1.212
 Residual             0.7886   0.888
Number of obs: 9304, groups:  pid, 360; tweet_id, 195

Fixed effects:
             Estimate Std. Error       df t value             Pr(>|t|)
(Intercept)     4.0099     0.1446 529.9875  27.724 < 0.0000000000000002 ***
nameperceiver   0.5839     0.1708 346.5754   3.418             0.000706 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
namepercevr -0.542
```

```r
# 获取置信区间
confint(model_or_study2, level = 0.95)
```

```
Computing profile confidence intervals ...

                  2.5 %    97.5 %
.sig01        1.4880514 1.7296384
.sig02        1.0961018 1.3455810
.sigma        0.8750365 0.9013501
(Intercept)   3.7264705 4.2934122
nameperceiver 0.2488372 0.9189277
```

```r
# 为幸福感情绪重复上述步骤，准备数据进行多层次模型分析
data_mlmh_study2 <- data_hap_trans_study2 %>% select(V1:V165, tweet_id, sr_happy) %>%
  pivot_longer(cols = -c(tweet_id, sr_happy),
               values_to = "judgment",
               names_to = "pid",
               values_drop_na = TRUE)
# 添加一个表示评判者是观察者还是作者的变量
data_mlmh_study2 <- data_mlmh_study2 %>% mutate(name = "perceiver")

# 准备自我报告数据，以便与 MLM 数据合并
self_report_trimh_study2 <- self_report_study2 %>% select(tweet_id, sr_happy) %>%
  mutate(pid = as.character(row_number(tweet_id)))

# 将自我报告数据添加到 MLM 数据中，并将评判值设置为自我报告的幸福感评分
data_mlm2h_study2 <- data_mlmh_study2 %>% select(tweet_id) %>%
  mutate(name = "author") %>% left_join(self_report_trimh_study2, by = "tweet_id") %>%
  mutate(judgment = sr_happy)
# 指定列的顺序
col_orderh_study2 <- c("tweet_id", "sr_happy", "pid",
                "judgment", "name")
# 按指定的列顺序对数据 _mlm2h 进行重排
data_mlm2h_study2 <- data_mlm2h_study2[, col_orderh]
# 合并观察者和作者的数据，并添加一个虚拟变量来区分两者
data_mlm3h_study2 <- rbind(data_mlmh_study2, data_mlm2h_study2) %>% arrange(tweet_id) %>%
  mutate(name_dum = ifelse(name == "author", 0, 1))
# 设置 R 的选项，以控制科学记数法的显示
options(scipen = 999)

# 拟合幸福感情绪的多层次模型
model_hap_study2 <- lmer(judgment ~ name + (1 | tweet_id) + (1 |pid), data = data_mlm3h_study2)
# 获取模型摘要
summary(model_hap_study2)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: judgment ~ name + (1 | tweet_id) + (1 | pid)
   Data: data_mlm3h_study2

REML criterion at convergence: 23286.3
```

```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-7.6647 -0.1573 -0.0106  0.0395  6.7880

Random effects:
 Groups   Name        Variance Std.Dev.
 pid      (Intercept) 2.3222   1.5239
 tweet_id (Intercept) 0.6561   0.8100
 Residual             0.5572   0.7465
Number of obs: 9304, groups:  pid, 360; tweet_id, 195

Fixed effects:
              Estimate Std. Error       df t value            Pr(>|t|)
(Intercept)     2.8051     0.1241 497.5800  22.608 <0.0000000000000002 ***
nameperceiver  -0.1698     0.1620 350.8545  -1.048               0.295
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
namepercevr -0.599
```

```r
# 获取置信区间
confint(model_hap_study2, level = 0.95)
```

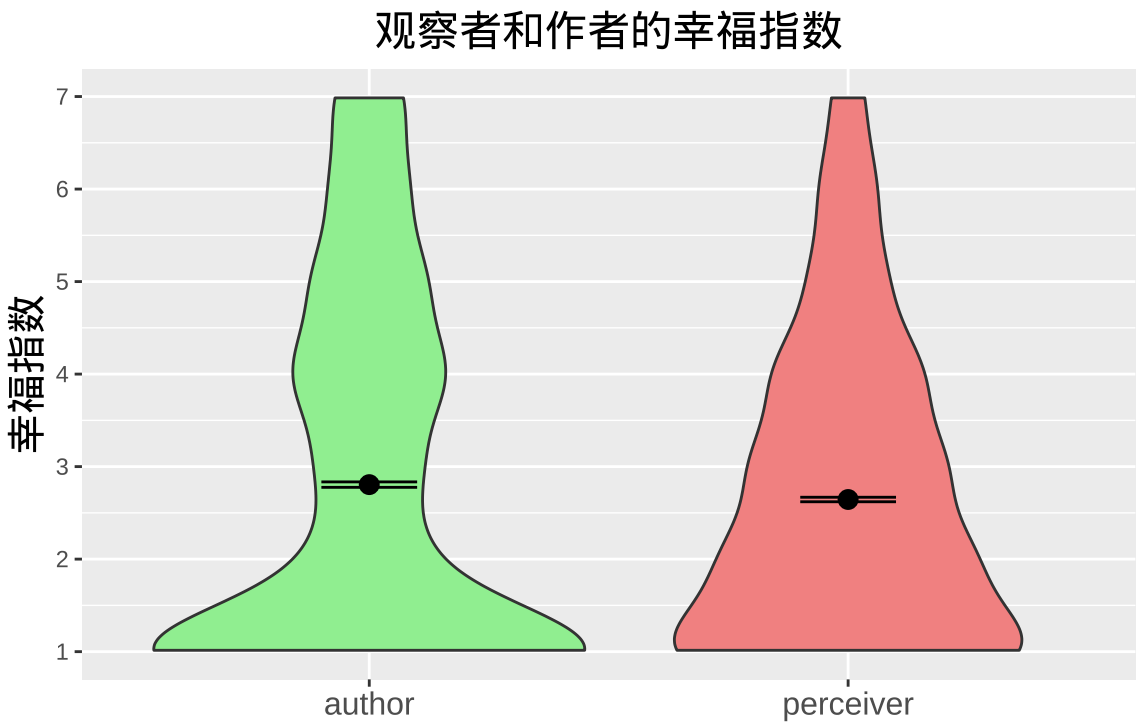
```
Computing profile confidence intervals ...

                   2.5 %    97.5 %
.sig01         1.4134396 1.6407204
.sig02         0.7316158 0.9005256
.sigma         0.7355299 0.7576467
(Intercept)    2.5619737 3.0482749
nameperceiver -0.4873870 0.1477717
```

### 小提琴图

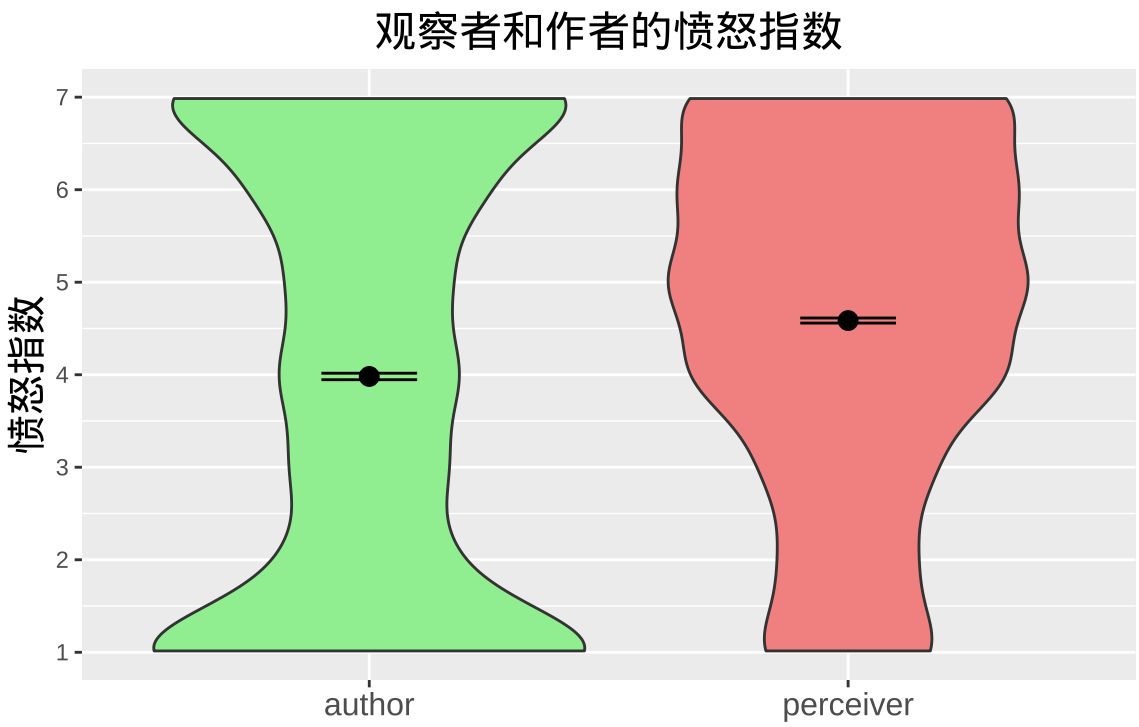
```r
# 创建一个 ggplot 对象，名为 p_hap，使用数据集 data_mlm3h
p_hap_study2 <- ggplot(data_mlm3h_study2, aes(x = name, y = judgment, fill = name)) +
```

```r
# 添加小提琴图层，展示数据分布
# trim = FALSE 表示不修剪小提琴图的边缘
# bw = 0.5 设置小提琴图的带宽
geom_violin(trim = FALSE, bw = 0.5) +

# 添加点图层，展示每个组的平均值
# stat = "summary" 表示使用汇总统计
# fun = "mean" 指定使用平均值作为点的位置
# 点的颜色、大小设置
geom_point(stat = "summary", fun = "mean", color = "black", size = 3) +

# 添加误差条图层，展示每个组平均值的标准误差
# fun.data 指定自定义函数，计算平均值的标准误差
# width 设置误差条的宽度
geom_errorbar(stat = "summary", fun.data = mean_se, width = 0.2) +

# 手动设置填充颜色，为不同的组分配不同的颜色
scale_fill_manual(values = c("author" = "lightgreen", "perceiver" = "lightcoral")) +

# 设置 y 轴的刻度和范围
# breaks 定义 y 轴的刻度
# limits 定义 y 轴的范围
scale_y_continuous(breaks = 1:7, limits = c(1, 7)) +

# 设置图表的标签和标题
# x 轴标签为空
# y 轴标签为" 幸福指数"
# 图表标题为" 观察者和作者的幸福指数"
labs(x = "", y = " 幸福指数", title = " 观察者和作者的幸福指数") +

# 设置图表的主题，包括图例位置、轴标题大小、轴文本大小、标题样式等
theme(
  legend.position = "none",  # 不显示图例
  axis.title.y = element_text(size = 15),  # y 轴标题文字大小
  axis.text.x = element_text(size = 12),  # x 轴文本大小
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5)  # 标题样式
)
```

```r
# 打印或显示图表
print(p_hap_study2)
```




```r
# 使用 ggplot 作图，复现文献 Fig2 中的 b 图
# 创建一个 ggplot 对象，名为 p__or_study2，使用数据集 data_mlm3h_study2
p_or_study2 <- ggplot(data_mlm3_study2, aes(x = name, y = judgment, fill = name)) +
geom_violin(trim = FALSE, bw = 0.5) +
geom_point(stat = "summary", fun = "mean", color = "black", size = 3) +
geom_errorbar(stat = "summary", fun.data = mean_se, width = 0.2) +
scale_fill_manual(values = c("author" = "lightgreen", "perceiver" = "lightcoral")) +
scale_y_continuous(breaks = 1:7, limits = c(1, 7)) +
labs(x = "", y = " 愤怒指数", title = " 观察者和作者的愤怒指数") +
theme(
  legend.position = "none",
  axis.title.y = element_text(size = 15),
  axis.text.x = element_text(size = 12),
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
)

# 打印或显示图表
print(p_or_study2)
```




## 研究 1 和研究 2 数据合并部分

过度感受愤怒和政治社交媒体使用的相关和回归

```
# 读取文件储存
op1 <- read_csv("../osfstorage-archive/Data/study1_overperception.csv")
op2 <- read_csv("../osfstorage-archive/Data/study2_overperception.csv")

# 将两个数据框按行合并，存储在 op_final 中
op_final <- rbind(op1, op2)

# 对变量'sm_use_politics_slider'和'overperception'进行皮尔逊相关性检验
cor.test(op_final$sm_use_politics_slider, op_final$overperception, method = "pearson")
```

```
    Pearson's product-moment correlation

data:  op_final$sm_use_politics_slider and op_final$overperception
t = 2.8552, df = 222, p-value = 0.004709
```

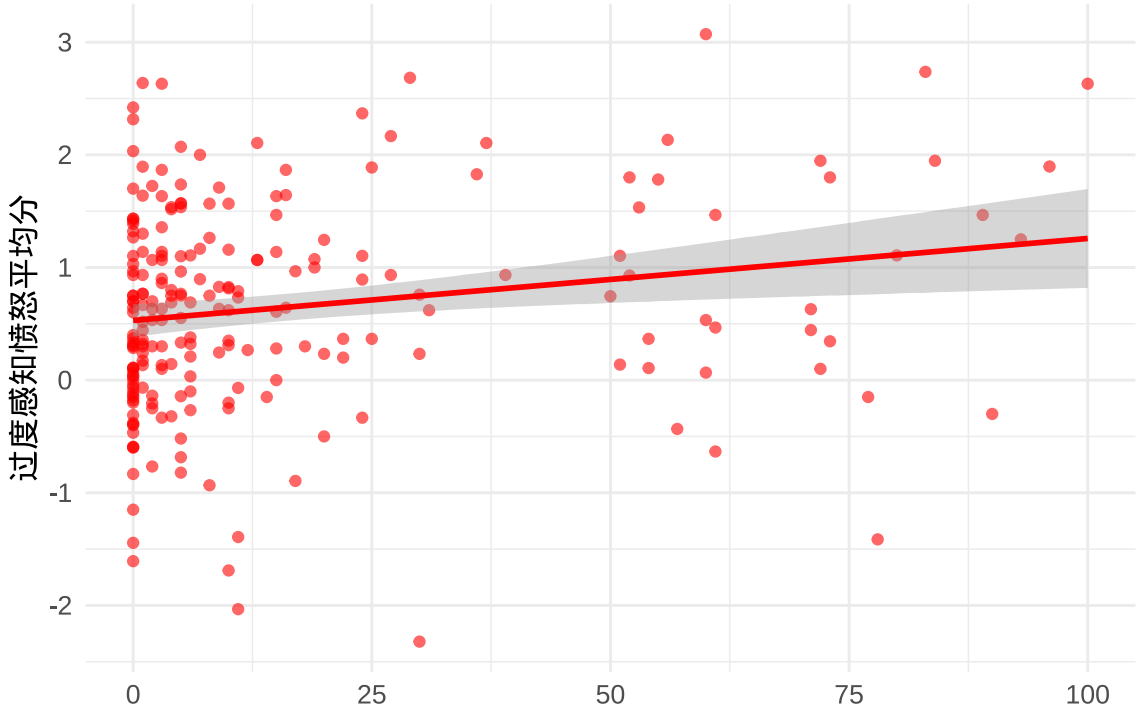
```
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.05856476 0.31159842
sample estimates:
      cor
0.1882028
```

```r
# 构建线性回归模型，解释变量为'overperception'
# 自变量为标准化后的'sm_use_politics_slider'、'ideo_extr'和'p_identity'
# 使用 op_final 数据框中的数据
summary(lm(overperception ~ scale(sm_use_politics_slider) +
             scale(ideo_extr) + scale(p_identity), data = op_final))
```

```
Call:
lm(formula = overperception ~ scale(sm_use_politics_slider) +
    scale(ideo_extr) + scale(p_identity), data = op_final)

Residuals:
     Min       1Q   Median       3Q      Max
-3.12131 -0.55868  0.01541  0.58270  2.07471

Coefficients:
                             Estimate Std. Error t value             Pr(>|t|)
(Intercept)                  0.644662   0.060924  10.581 < 0.0000000000000002 ***
scale(sm_use_politics_slider) 0.168631   0.063744   2.645              0.00875 **
scale(ideo_extr)              0.046878   0.066740   0.702              0.48318
scale(p_identity)            -0.009182   0.069938  -0.131              0.89566
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9108 on 220 degrees of freedom
  (因为不存在，51个观察量被删除了)
Multiple R-squared:  0.03771,   Adjusted R-squared:  0.02459
F-statistic: 2.874 on 3 and 220 DF,  p-value: 0.03713
```

```r
# 过度感知愤怒和政治社交媒体使用的关系图
p_cor1 <- ggplot(op_final, aes(x = sm_use_politics_slider,
                               y = overperception)) +
  geom_point(color = "red", alpha = 0.6) +
```

```
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  theme_minimal() +
  labs(x = "", y = " 过度感知愤怒平均分") +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    axis.title.y = element_text(size = 12),
    axis.text.y = element_text(size = 10),
    axis.text.x = element_text(size = 10)
  )
print(p_cor1)
```




## 研究 3

### 读取数据与数据清洗

```
# 读取数据
self_report_study3 <-
  read_csv("../osfstorage-archive/Data/study2_self_report.csv")
data_study3 <- read_csv("../osfstorage-archive/Data/study3_data_raw.csv")
```

```r
# 数据清洗
# 性别
data_study3 <- data_study3 %>% mutate(gender_label =
                           ifelse(gender == 1, "Male",
                                  ifelse(gender == 2, "Female", "Other")))

# 党派
data_study3 <- data_study3 %>% mutate(
  party_label =
    ifelse(party == 1, "Democrat",
           ifelse(party == 2, "Republican",
                  ifelse(party == 3, "Indepedent",
                         ifelse(party == 4, "Other", "None")))))

# 政治身份强度
data_study3 <- data_study3 %>% mutate(
  p_identity = ifelse(is.na(sis_dem) == TRUE, sis_rep, sis_dem))

# 意识形态
data_study3 <- data_study3 %>% mutate(ideo_extr = abs(ideo))

# 数据剔除
frequencies.table(data_study3$comp_check)
```

| | Freq | % |
|---|---|---|
| 1 | 350 | 97.22 |
| 3 | 9 | 2.50 |
| 4 | 1 | 0.28 |

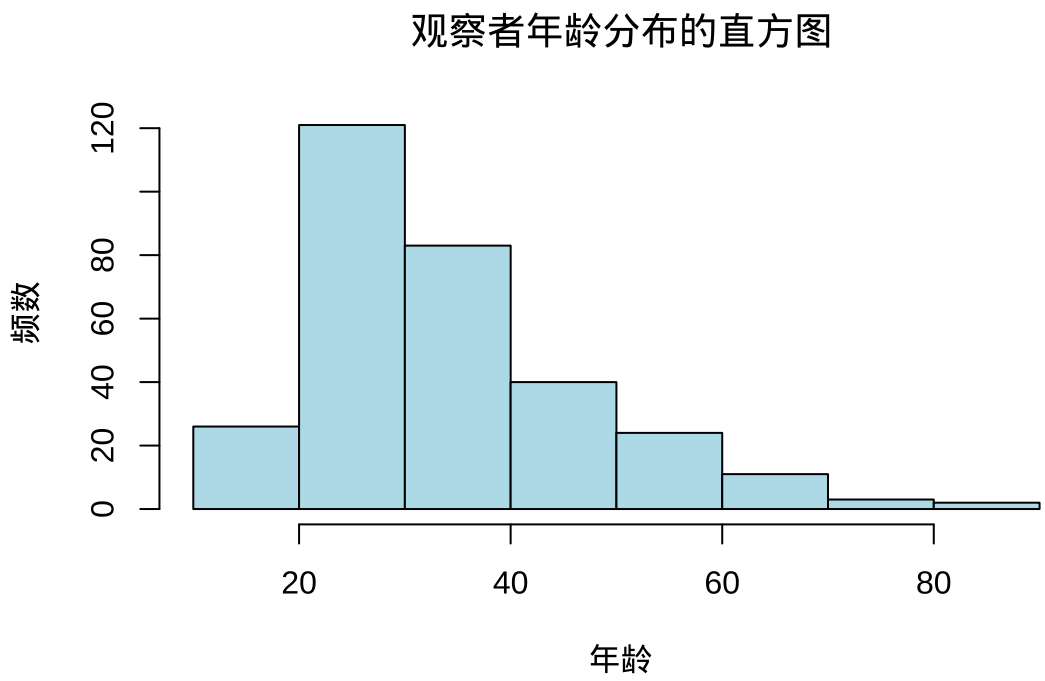
```r
data_trim_study3 <- data_study3 %>% filter(comp_check == 1)

data_trim_study3 <- data_trim_study3 %>% filter(
  party_label == "Democrat" | party_label == "Republican")
```

### 描述性统计（作图）

```r
# 观察者性别频数分布表
frequencies.table(data_trim_study3$gender_label)
```

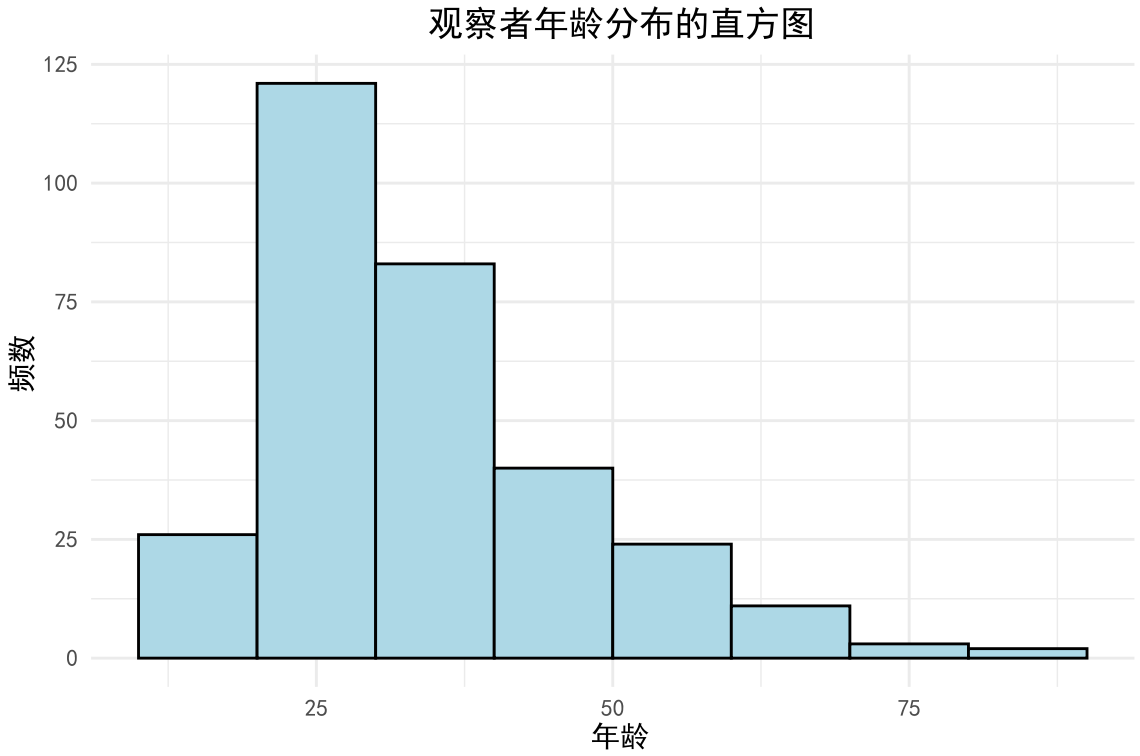
| | Freq | % |
|---|---|---|
| Other | 310 | 100 |

```
# 观察者年龄的直方图
hist_data_study3 <- hist(data_trim_study3$age,
                         plot = FALSE)
plot(hist_data_study3,
     main = " 观察者年龄分布的直方图",
     xlab = " 年龄",
     ylab = " 频数",
     col = "lightblue",
     border = "black")
```




```
ggplot(data_trim_study3, aes(x = age)) +
  geom_histogram(breaks = hist_data_study3$breaks,
                 fill = "lightblue",
                 color = "black") +
  labs(title = " 观察者年龄分布的直方图",
       x = " 年龄",
       y = " 频数") +
  theme_minimal() +
  theme(
```

```
    plot.title = element_text(hjust = 0.5),
    text = element_text(family = "SimHei")
  )
```




```
# 意识形态
frequencies.table(data_trim_study3$ideo)
```

| | Freq | % |
|---|---|---|
| -3 | 50 | 16.13 |
| -2 | 50 | 16.13 |
| -1 | 57 | 18.39 |
| 0 | 17 | 5.48 |
| 1 | 66 | 21.29 |
| 2 | 44 | 14.19 |
| 3 | 26 | 8.39 |

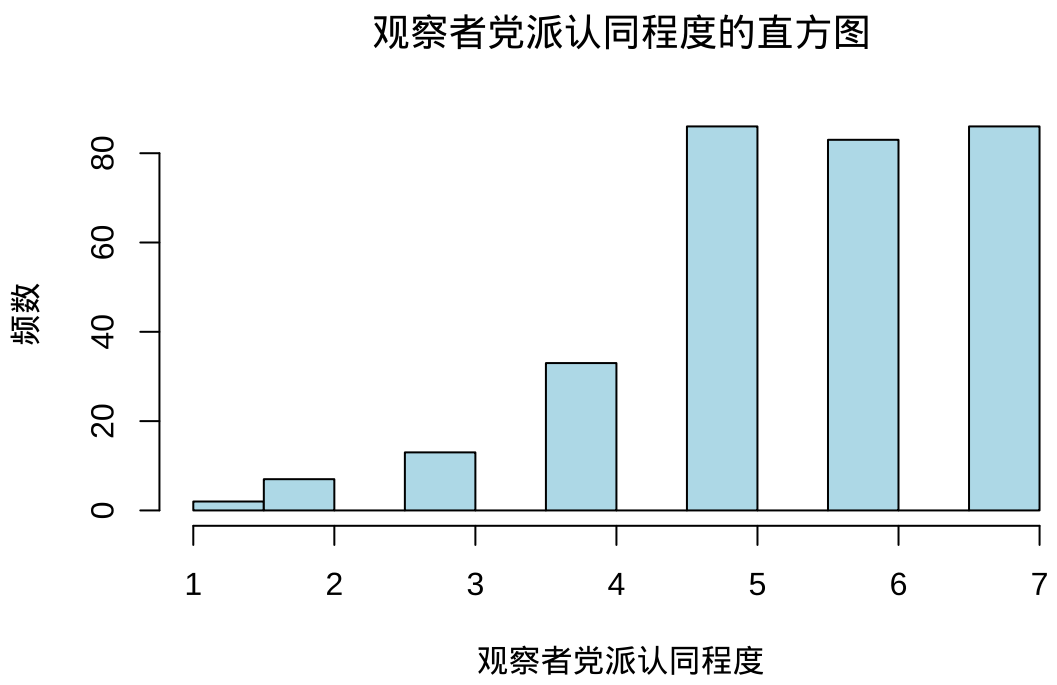
```
# 党派
frequencies.table(data_trim_study3$party_label)
```

| | Freq | % |
|---|---|---|
| Democrat | 161 | 51.94 |
| Republican | 149 | 48.06 |

```r
# 党派认同程度
p_identity_summary_study3 <-
  summary(data_trim_study3$p_identity)
print(p_identity_summary_study3)
```

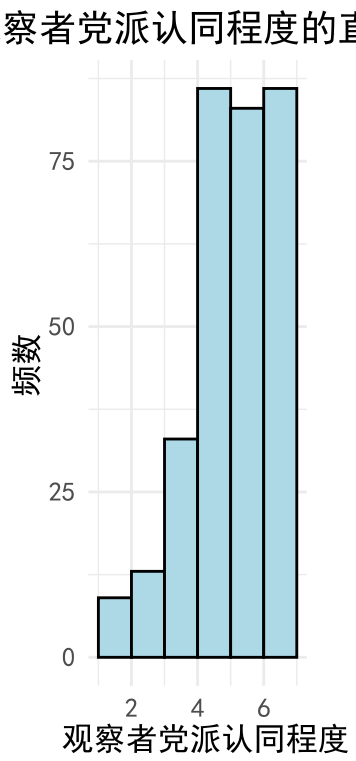
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   5.000   6.000   5.539   7.000   7.000
```

```r
# 观察者党派认同程度的直方图和平均值
hist_data_study3 <- hist(data_trim_study3$p_identity,
                  plot = FALSE)
plot(hist_data_study3,
     main = " 观察者党派认同程度的直方图",
     xlab = " 观察者党派认同程度",
     ylab = " 频数",
     col = "lightblue",
     border = "black")
```




```r
ggplot(data_trim_study3, aes(x = p_identity)) +
  geom_histogram(binwidth = 1,
                 fill = "lightblue",
```

```
                 color = "black",
                 boundary = 2) + # 确保柱子连在一起
  labs(title = " 观察者党派认同程度的直方图",
       x = " 观察者党派认同程度",
       y = " 频数") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10),
    text = element_text(family = "SimHei")
  ) +
  coord_fixed(ratio = 1/5)
```

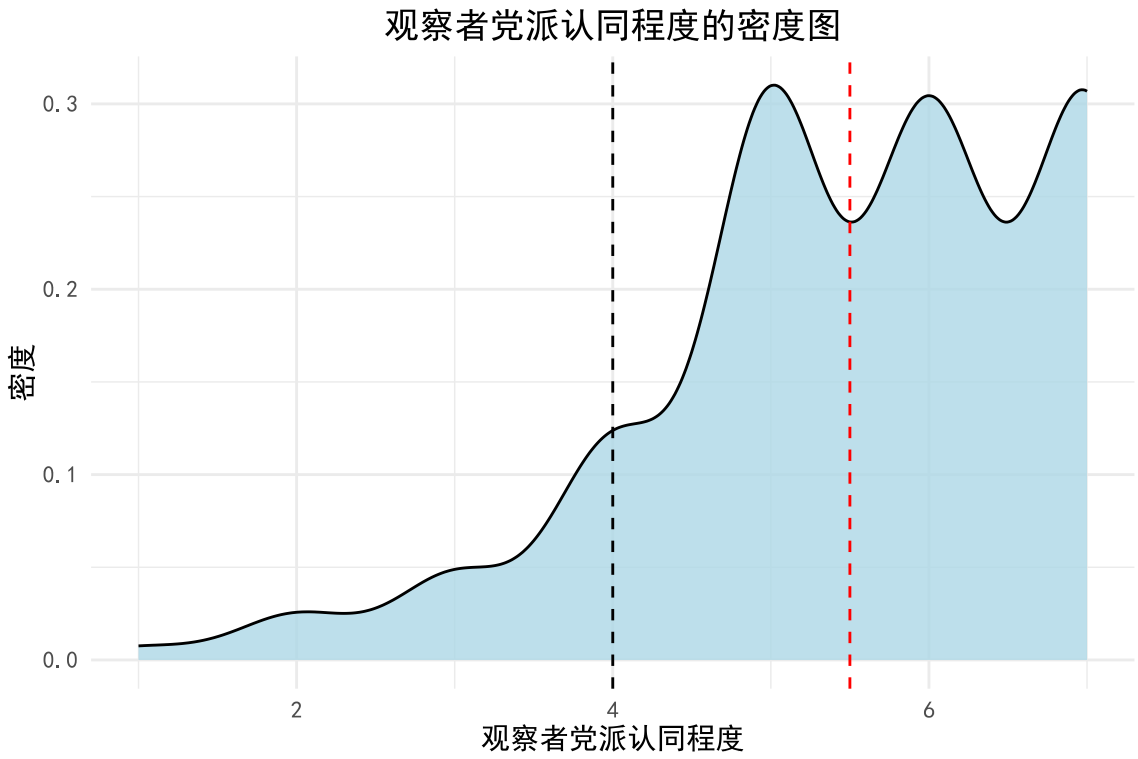



```
round(mean(data_trim_study3$p_identity, na.rm = TRUE),2)
```

```
[1] 5.54
```

```
# 党派认同程度密度图
data_trim_study3 %>%
  ggplot(aes(x = p_identity)) +
```

```
  geom_density(fill = "lightblue", alpha = .8) +
  labs(title = " 观察者党派认同程度的密度图", x = " 观察者党派认同程度", y = " 密度") +
  geom_vline(xintercept = 5.5, linetype = "dashed", color = "red") +
  geom_vline(xintercept = 4, linetype = "dashed") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5), text = element_text(family = "SimHei"))
```




## 选择、转置、计算平均和合并过度感知情绪相关的数据

```
# 选择与愤怒情绪相关的推文级别评分
# 然后转置、转换为数据框，并增加新列
data_or_trans_study3 <- data_trim_study3 %>%
  select(ends_with("_or")) %>%
  t() %>%
  as.data.frame() %>%
  mutate(tweet_id = as.integer(numextract(rownames(.))),
         tweet_id_char = rownames(.)) %>%
  arrange(tweet_id)

# 选择与快乐情绪相关的推文级别评分，执行与愤怒情绪相同的操作
data_hap_trans_study3 <- data_trim_study3 %>%
```

```
  select(ends_with("_hap")) %>%
  t() %>%
  as.data.frame() %>%
  mutate(tweet_id = as.integer(numextract(rownames(.))),
         tweet_id_char = rownames(.)) %>%
  arrange(tweet_id)

# 计算每个推文的愤怒情绪平均感知值
data_or_trans_study3$mean_or <- data_or_trans_study3 %>%
  select(-c(tweet_id, tweet_id_char)) %>%
  rowMeans(., na.rm = TRUE)

# 计算每个推文的快乐情绪平均感知值
data_hap_trans_study3$mean_hap <- data_hap_trans_study3 %>%
  select(-c(tweet_id, tweet_id_char)) %>%
  rowMeans(., na.rm = TRUE)

# 将自我报告的评分数据加入到愤怒和快乐情绪的数据中
# 以便于比较（join by tweet_id）
data_or_trans_study3 <- data_or_trans_study3 %>%
  left_join(self_report, by = "tweet_id")
data_hap_trans_study3 <- data_hap_trans_study3 %>%
  left_join(self_report, by = "tweet_id")

# 建立多层次模型
ncol(data_or_trans_study3)
```

```
[1] 319
```

```
# 数据准备
data_mlm_study3 <- data_or_trans_study3 %>% select(V1:V310, tweet_id, sr_outrage) %>%
  pivot_longer(cols = -c(tweet_id, sr_outrage),
               values_to = "judgment",
               names_to = "pid",
               values_drop_na = TRUE)

# 添加被试类型标签
data_mlm_study3 <- data_mlm_study3 %>% mutate(name = "perceiver")
```

```r
# 准备自我报告数据
self_report_trim_study3 <- self_report_study3 %>%
  select(tweet_id, sr_outrage) %>%
  mutate(pid = as.character(row_number(tweet_id)))

# 创建作者数据集
data_mlm2_study3 <- data_mlm_study3 %>%
  select(tweet_id) %>%
  mutate(name = "author") %>%
  left_join(self_report_trim_study3, by = "tweet_id") %>%
  mutate(judgment = sr_outrage)

# 确定列顺序并合并数据集
col_order <- c("tweet_id", "sr_outrage", "pid",
               "judgment", "name")

data_mlm2_study3 <- data_mlm2_study3[, col_order]

data_mlm3_study3 <-
  rbind(data_mlm_study3, data_mlm2_study3) %>%
  arrange(tweet_id) %>%
  mutate(name_dum = ifelse(name == "author", 0, 1))


options(scipen = 999)

# 模型 1 拟合
model_or_study3 <- lmer(
  judgment ~ name + (1 | tweet_id) + (1 | pid),
  data = data_mlm3_study3
)
summary(model_or_study3)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: judgment ~ name + (1 | tweet_id) + (1 | pid)
   Data: data_mlm3_study3

REML criterion at convergence: 49510.9
```

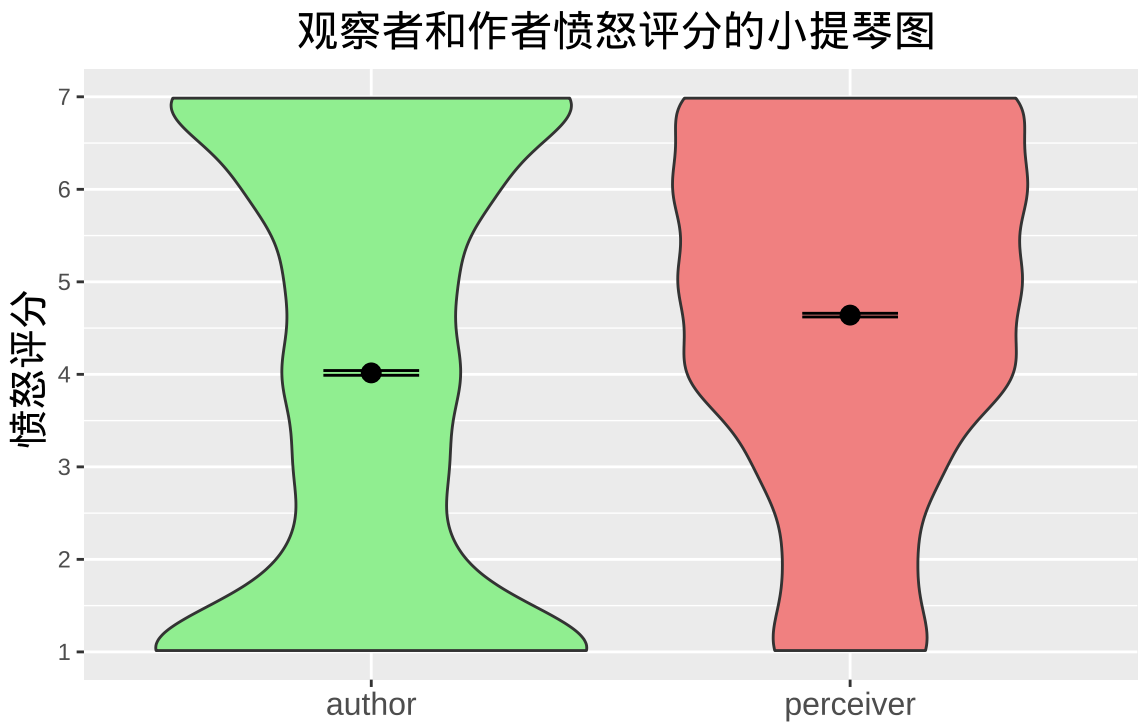
```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-6.2514 -0.0363  0.0037  0.0820  5.8345

Random effects:
 Groups   Name        Variance Std.Dev.
 pid      (Intercept) 1.9790   1.4068
 tweet_id (Intercept) 1.4054   1.1855
 Residual             0.8399   0.9164
Number of obs: 17444, groups:  pid, 507; tweet_id, 197

Fixed effects:
             Estimate Std. Error       df t value             Pr(>|t|)
(Intercept)     4.0051     0.1314 623.8705  30.471 < 0.0000000000000002 ***
nameperceiver   0.6231     0.1291 495.9253   4.825           0.00000187 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
namepercevr -0.598
```

```r
round(confint(model_or_study3, level = 0.95),2)
```

```
             2.5 % 97.5 %
.sig01        1.32   1.50
.sig02        1.07   1.31
.sigma        0.91   0.93
(Intercept)   3.75   4.26
nameperceiver 0.37   0.88
```

```r
# 观察者和作者愤怒评分的小提琴图
p_or_study3 <- ggplot(data_mlm3_study3, aes(
  x = name, y = judgment, fill = name)) +
  geom_violin(trim = FALSE, bw = 0.5) +
  geom_point(stat = "summary", fun = "mean",
             color = "black", size = 3) +
  geom_errorbar(stat = "summary",
                fun.data = mean_se, width = 0.2) +
  scale_fill_manual(
```

```
    values = c("author" = "lightgreen",
               "perceiver" = "lightcoral")) +
  scale_y_continuous(breaks = 1:7, limits = c(1, 7)) +
  labs(x = "", y = " 愤怒评分",
       title = " 观察者和作者愤怒评分的小提琴图") +
  theme(
    legend.position = "none",
    axis.title.y = element_text(size = 15),
    axis.text.x = element_text(size = 12),
    plot.title = element_text(size = 16,
                              face = "bold", hjust = 0.5)
  )
print(p_or_study3)
```




```
# 建立多层次模型：和上面步骤一样，只是分析的情绪是 happiness

data_mlmh_study3 <- data_hap_trans_study3 %>%
  select(V1:V310, tweet_id, sr_happy) %>%
  pivot_longer(cols = -c(tweet_id, sr_happy),
               values_to = "judgment",
               names_to = "pid",
```

```
                values_drop_na = TRUE)

data_mlmh_study3 <- data_mlmh_study3 %>%
  mutate(name = "perceiver")

self_report_trimh_study3 <- self_report_study3 %>%
  select(tweet_id, sr_happy) %>%
  mutate(pid = as.character(row_number(tweet_id)))

data_mlm2h_study3 <- data_mlmh_study3 %>%
  select(tweet_id) %>%
  mutate(name = "author") %>%
  left_join(self_report_trimh_study3, by = "tweet_id") %>%
  mutate(judgment = sr_happy)


col_orderh <- c("tweet_id", "sr_happy", "pid",
                "judgment", "name")

data_mlm2h_study3 <- data_mlm2h_study3[, col_orderh]


data_mlm3h_study3 <- rbind(data_mlmh_study3,
                           data_mlm2h_study3) %>%
  arrange(tweet_id) %>%
  mutate(name_dum = ifelse(name == "author", 0, 1))

options(scipen = 999)

# 模型 2 拟合
model_hap_study3<- lmer(
  judgment ~ name + (1 | tweet_id) + (1 |pid),
  data = data_mlm3h_study3)
summary(model_hap_study3)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method ['lmerModLmerTest']
Formula: judgment ~ name + (1 | tweet_id) + (1 | pid)
   Data: data_mlm3h_study3
```

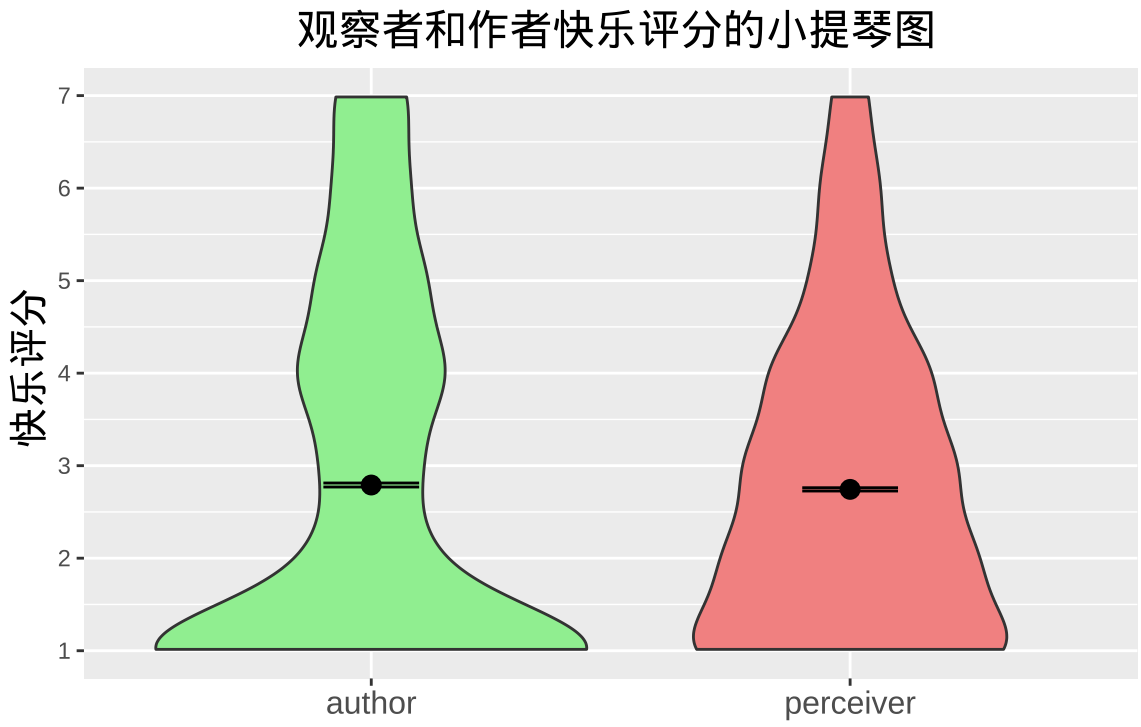
```
REML criterion at convergence: 44872.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-6.9977 -0.1355 -0.0083  0.0298  7.2384

Random effects:
 Groups   Name        Variance Std.Dev.
 pid      (Intercept) 1.9096   1.3819
 tweet_id (Intercept) 0.5815   0.7626
 Residual             0.6439   0.8024
Number of obs: 17444, groups:  pid, 507; tweet_id, 197

Fixed effects:
               Estimate Std. Error        df t value            Pr(>|t|)
(Intercept)     2.79186    0.11278 674.50195  24.755 <0.0000000000000002 ***
nameperceiver  -0.05651    0.12664 498.70651  -0.446               0.656
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr)
namepercevr -0.684
```

```
round(confint(model_hap_study3, level = 0.95),2)
```

```
              2.5 % 97.5 %
.sig01         1.30   1.47
.sig02         0.69   0.85
.sigma         0.79   0.81
(Intercept)    2.57   3.01
nameperceiver -0.30   0.19
```

```
# 观察者和作者快乐评分的小提琴图
p_hap_study3 <-
  ggplot(data_mlm3h_study3, aes(
    x = name, y = judgment, fill = name)) +
  geom_violin(trim = FALSE, bw = 0.5) +
  geom_point(stat = "summary",
             fun = "mean", color = "black", size = 3) +
```

```
  geom_errorbar(stat = "summary",
                fun.data = mean_se, width = 0.2) +
  scale_fill_manual(values =
                      c("author" = "lightgreen",
                        "perceiver" = "lightcoral")) +
  scale_y_continuous(breaks = 1:7, limits = c(1, 7)) +
  labs(x = "", y = " 快乐评分",
       title = " 观察者和作者快乐评分的小提琴图") +
  theme(
    legend.position = "none",
    axis.title.y = element_text(size = 15),
    axis.text.x = element_text(size = 12),
    plot.title = element_text(size = 16,
                              face = "bold", hjust = 0.5)
  )
print(p_hap_study3)
```



政治媒体使用对过度感知的相关和回归

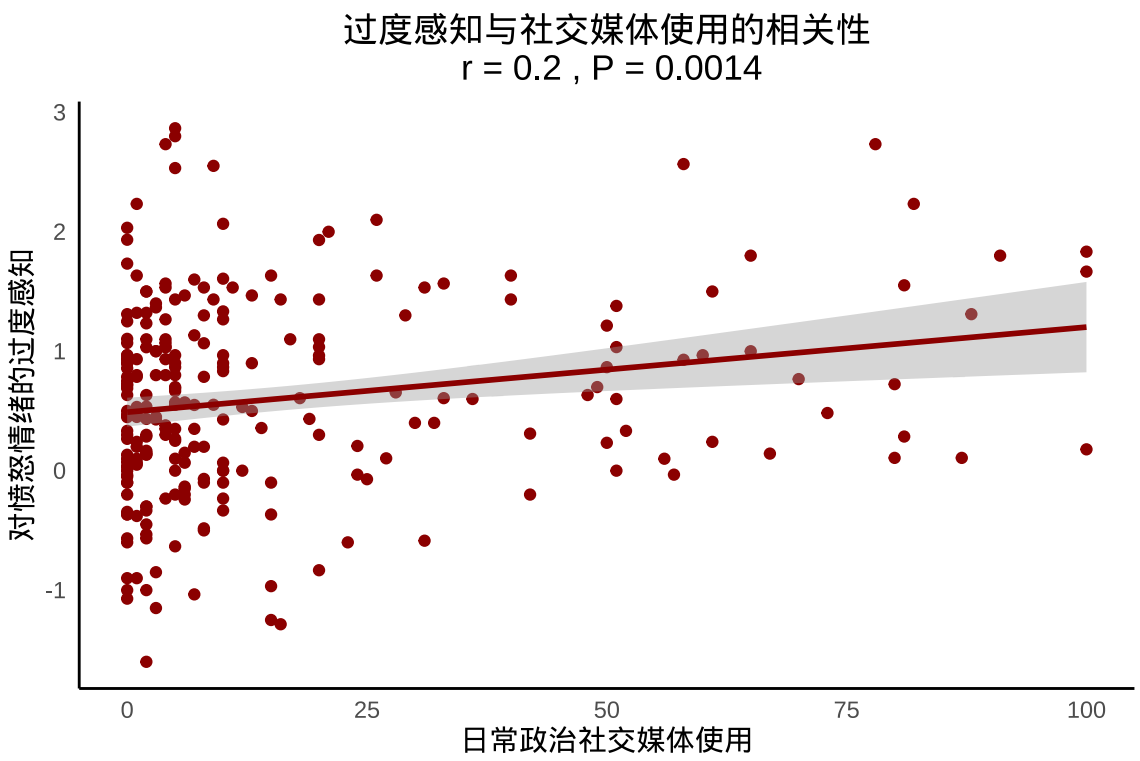
```
# 定义暗红色
dark_red <- "#8B0000"
# 读取数据
op3 <- read_csv(
  "../osfstorage-archive/Data/study3_overperception.csv")

# 计算 Pearson 相关系数并保存结果
correlation_result_study3 <-
  cor.test(op3$sm_use_politics_slider,
           op3$overperception, method = "pearson")

# 查看相关系数和 P 值
r_value_study3 <- correlation_result_study3$estimate
p_value_study3 <- correlation_result_study3$p.value

# 打印相关系数和 P 值
print(correlation_result_study3)
```

```
    Pearson's product-moment correlation

data:  op3$sm_use_politics_slider and op3$overperception
t = 3.2413, df = 248, p-value = 0.001353
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.07952028 0.31771867
sample estimates:
      cor
0.2015983
```

```
# 执行回归分析
model_study3 <- lm(overperception ~
                     scale(sm_use_politics_slider) +
                     scale(ideo_extr) + scale(p_identity),
                   data = op3)
summary(model_study3)
```

```
Call:
lm(formula = overperception ~ scale(sm_use_politics_slider) +
    scale(ideo_extr) + scale(p_identity), data = op3)

Residuals:
     Min       1Q   Median       3Q      Max
-2.04908 -0.55293 -0.01526  0.53870  2.26888

Coefficients:
                              Estimate Std. Error t value             Pr(>|t|)
(Intercept)                    0.60173    0.05105  11.787 < 0.0000000000000002 ***
scale(sm_use_politics_slider)  0.16255    0.05239   3.103              0.00214 **
scale(ideo_extr)              -0.03796    0.05402  -0.703              0.48294
scale(p_identity)              0.00670    0.05501   0.122              0.90316
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8071 on 246 degrees of freedom
  (因为不存在，60个观察量被删除了)
Multiple R-squared:  0.04266,   Adjusted R-squared:  0.03099
F-statistic: 3.654 on 3 and 246 DF,  p-value: 0.01318
```

```r
# 绘制散点图与回归线，使用暗红色
ggplot(op3, aes(x = sm_use_politics_slider,
                y = overperception)) +
  geom_point(color = dark_red) +  # 散点图，颜色设置为暗红色
  geom_smooth(method = "lm",
              formula = y ~ x, se = TRUE,
              color = dark_red) +
  # 线性回归线和置信区间，颜色设置为暗红色
  theme_minimal() +  # 使用简洁主题
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid = element_blank(),  # 去除网格线
        axis.line = element_line(color = "black")) +
  # 保留横轴和纵轴为黑色
  labs(title = paste(" 过度感知与社交媒体使用的相关性\n",
                     "r =", round(r_value_study3, 2),
                     ", P =", round(p_value_study3, 4)),
       x = " 日常政治社交媒体使用",
       y = " 对愤怒情绪的过度感知")
```




```
round(confint(model_study3, level = 0.95),2)
```

```
                              2.5 % 97.5 %
(Intercept)                    0.50   0.70
scale(sm_use_politics_slider)  0.06   0.27
scale(ideo_extr)              -0.14   0.07
scale(p_identity)             -0.10   0.12
```

## 研究 4

### 研究 4 用到的包、函数

```
packages <- c("lsr","rstatix","car")
 lapply(packages, function(pkg) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg, dependencies = TRUE)
  }
})
```

```
[[1]]
NULL

[[2]]
NULL

[[3]]
NULL
```

```r
library(lsr)
library(rstatix)
library(car)

data_study4 <-
  read_csv("../osfstorage-archive/Data/study4_data_raw.csv")
stim_study4 <-
  read_csv("../osfstorage-archive/Data/stim_descriptives_data.csv")

# 计算汇总统计量的标准误差（Standard Error, SE）
summarySE <- function(
    data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                      conf.interval=.95, .drop=TRUE) {
  library(plyr)# 对 R 语言中内置的 engths 函数的一个扩展
# 增加了对 NA 的处理选项
  length2 <- function (x, na.rm=FALSE) {
    if (na.rm) sum(!is.na(x))
    else length(x)
  }
# 指定的组（由 groupvars 定义）计算
# 每个变量（由 measurevar 定义）的 N、mean 和 sd
  datac <- ddply(data, groupvars, .drop=.drop,
                 .fun = function(xx, col) {
                   c(N    = length2(xx[[col]], na.rm=na.rm),
                     mean = mean   (xx[[col]], na.rm=na.rm),
                     sd   = sd     (xx[[col]], na.rm=na.rm)
                   )
                 },
                 measurevar
  )
 # 重命名一个列，以及计算均值的标准误差
```

```
  datac <- rename(datac, c("mean" = measurevar))
  datac$se <- datac$sd / sqrt(datac$N)
  # 计算均值的置信区间
  ciMult <- qt(conf.interval/2 + .5, datac$N-1)
  datac$ci <- datac$se * ciMult
  return(datac)
}
```

### 数据清洗

```
# 计算原始数据集 data 的行数，即记录数
data_study4 %>% nrow()
```

```
[1] 602
```

```
# 经过过滤后剩余的数据集
data_trim_study4 <- data_study4 %>%
  filter(political_party == "Republican" | political_party == "Democrat")

# 被移除的记录数；除去没有政治党派倾向的参与者 (N = 27)
nrow(data_study4) - nrow(data_trim_study4)
```

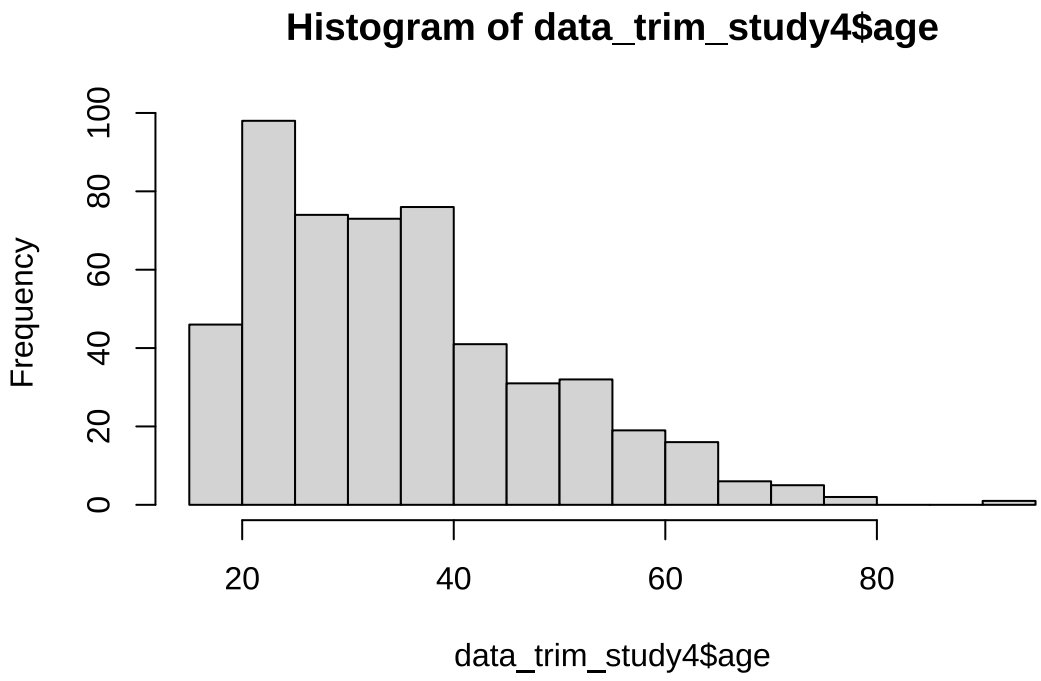
```
[1] 27
```

```
# 进一步过滤 data_trim，未通过理解性检查的参与者 (N = 52)
data_trim_study4 <- data_trim_study4 %>% filter(barr_check + barrett_check == 2)

# final N = 523 剩余的样本大小
nrow(data_trim_study4)
```

```
[1] 523
```

### 描述性统计

```
# 绘制数据集 data_trim 中 age 字段（年龄）的直方图，用于可视化年龄的分布情况
hist(data_trim_study4$age)
```



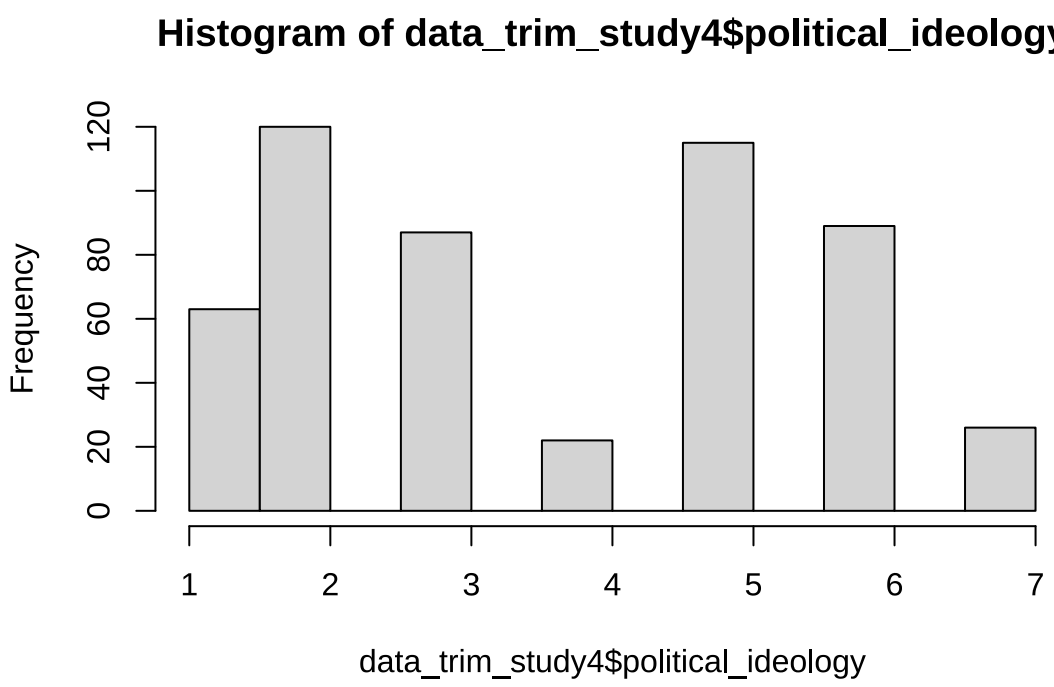

```
mean(data_trim_study4$age,
     na.rm = TRUE)#age 字段的平均值，忽略 NA 值
```

```
[1] 35.61923
```

```
sd(data_trim_study4$age, na.rm = TRUE)# 计算 age 字段的标准差
```

```
[1] 13.33535
```

```
# ideology; political_ideology 字段（政治意识形态）的直方图
hist(data_trim_study4$political_ideology)
```




```r
# party ; 使用之前定义的 frequencies.table 函数为 political_party 字段生成频率表
frequencies.table(data_trim_study4$political_party)
```

| | Freq | % |
|---|---|---|
| Democrat | 276 | 52.77 |
| Republican | 247 | 47.23 |

```r
# familiar with Barr;（对 Barr 的熟悉程度）生成频率表和百分比。
frequencies.table(data_trim_study4$barr_familiar)
```

| | Freq | % |
|---|---|---|
| -3 | 79 | 15.37 |
| -2 | 75 | 14.59 |
| -1 | 75 | 14.59 |
| 0 | 111 | 21.60 |
| 1 | 87 | 16.93 |
| 2 | 55 | 10.70 |
| 3 | 32 | 6.23 |

```r
# familiar with Barrett; 对 barrett_familiar 字段（对 Barrett 的熟悉程度）
frequencies.table(data_trim_study4$barrett_familiar)
```

| | Freq | % |
|---|---|---|
| -3 | 34 | 6.60 |
| -2 | 27 | 5.24 |
| -1 | 34 | 6.60 |
| 0 | 108 | 20.97 |
| 1 | 94 | 18.25 |
| 2 | 114 | 22.14 |
| 3 | 104 | 20.19 |

```
# gender; gender 字段（性别）生成频率表和百分比
frequencies.table(data_trim_study4$gender)
```

| | Freq | % |
|---|---|---|
| 23 | 1 | 0.19 |
| 27 | 1 | 0.19 |
| agender | 1 | 0.19 |
| f | 3 | 0.57 |
| F | 5 | 0.96 |
| feamle | 1 | 0.19 |
| female | 170 | 32.57 |
| Female | 111 | 21.26 |
| FEMALE | 5 | 0.96 |
| Femlae | 1 | 0.19 |
| Fenale | 1 | 0.19 |
| Frmale | 1 | 0.19 |
| genderfluid | 1 | 0.19 |
| genderqueer | 1 | 0.19 |
| Helicopter | 1 | 0.19 |
| m | 2 | 0.38 |
| M | 3 | 0.57 |
| male | 99 | 18.97 |
| Male | 104 | 19.92 |
| MALE | 3 | 0.57 |
| non-binary | 1 | 0.19 |
| Non binary | 1 | 0.19 |
| nonbinary | 3 | 0.57 |
| Nonbinary | 1 | 0.19 |
| Perfer not to disclose | 1 | 0.19 |

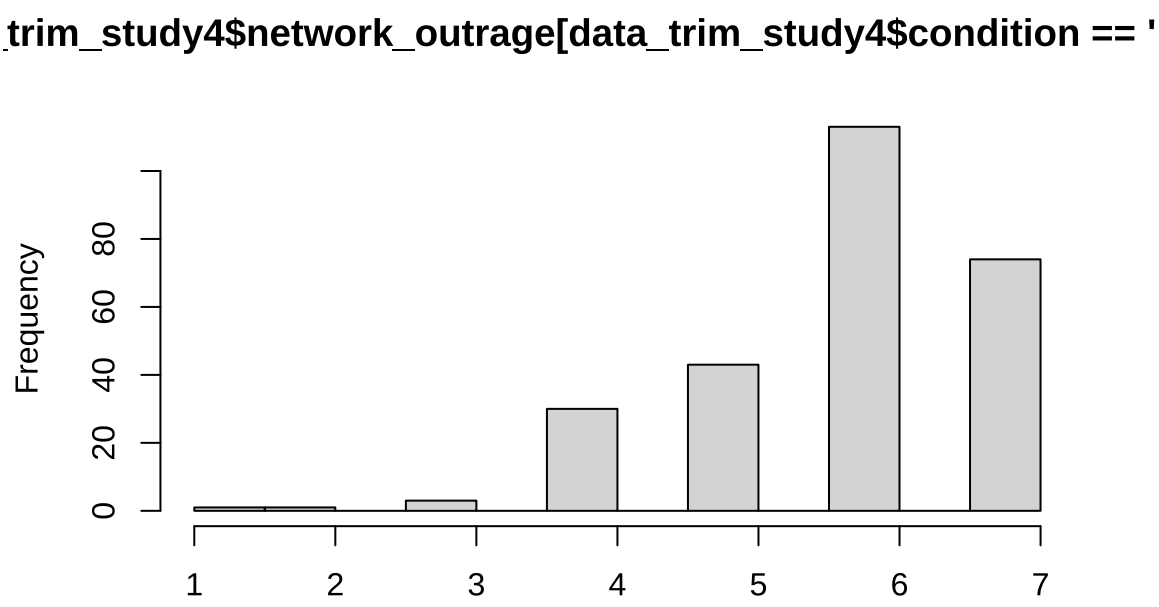
设置数据集和变量，以便进行进一步的分析或可视化

```
# 将原始数据集 data 写入名为"study4_data_raw.csv" 的 CSV 文件
data_study4 %>% write_csv("../osfstorage-archive/Data/study4_data_raw.csv")

# make factor for plotting
# 创建一个新变量 condition_fac，将现有的 condition 变量转换为因子（factor）类型的变量
# 并指定其水平（levels）为"High Overperception" 和"Low Overperception" 来创建的
data_trim_study4 <- data_trim_study4 %>%
  mutate(condition_fac =
           factor(condition,
                  levels = c("High Overperception",
                             "Low Overperception")))
```

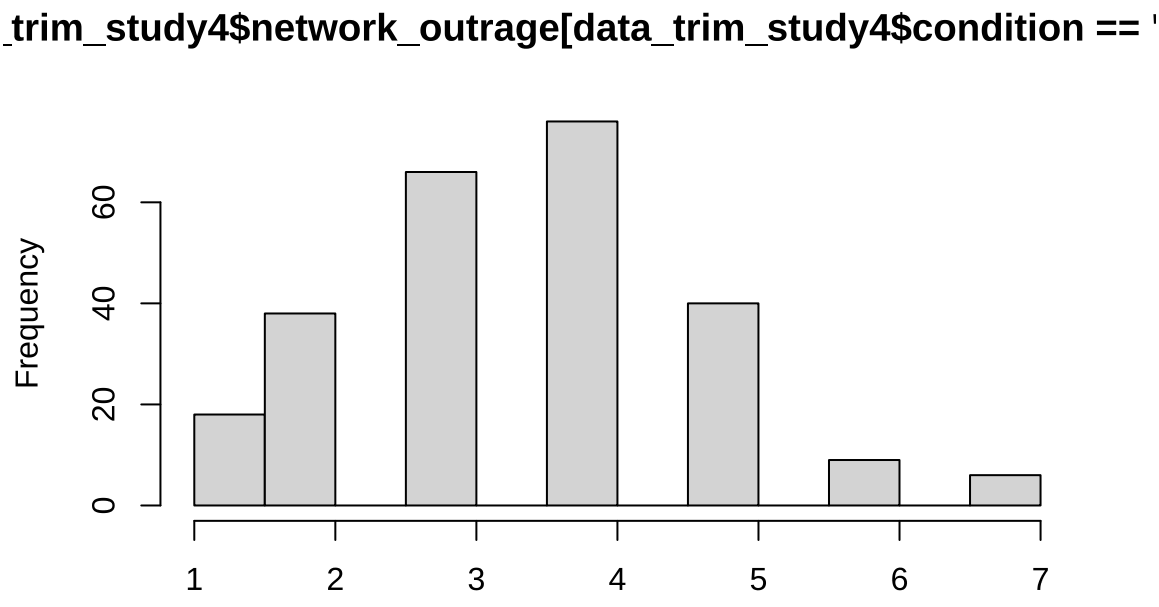
### t 检验

检验两个组别（高过度感知、低过度感知）在因变量（network_outrage）上是否存在显著差异，并且检查数据的正态性和方差齐性

```
# 绘制直方图
hist(data_trim_study4$network_outrage[data_trim_study4$condition == "High Overperception"])
```




```
hist(data_trim_study4$network_outrage[data_trim_study4$condition == "Low Overperception"])
```



```r
# 方差齐性检验
leveneTest(network_outrage ~ condition, data = data_trim_study4)
```

```
Warning in leveneTest.default(y = y, group = group, ...): group coerced to factor.

Levene's Test for Homogeneity of Variance (center = median)
       Df F value      Pr(>F)
group   1  20.276 0.000008293 ***
      516
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# t 检验
t.test(network_outrage ~ condition, data = data_trim_study4, var.equal = FALSE)
```

```
    Welch Two Sample t-test

data:  network_outrage by condition
t = 21.563, df = 479.5, p-value < 0.00000000000000022
alternative hypothesis: true difference in means between group High Overperception and group Low O
95 percent confidence interval:
 2.087642 2.506258
sample estimates:
mean in group High Overperception  mean in group Low Overperception
                         5.822642                          3.525692
```

```r
# compute cohen's d 效应量计算
cohensD(network_outrage ~ condition, data = data_trim_study4)
```

```
[1] 1.905544
```

两个单样本 t 检验，以比较两组（高过度感知和低过度感知）的 **network_outrage** 变量的均值与某个已知的均值（**mu**）是否存在显著差异

```r
# 数据准备
# 筛选 stim 数据集中 over_under 字段等于"Overperceived" 的记录，并存储为 over_stim
over_stim <- stim_study4 %>%
```

```r
  filter(over_under == "Overperceived")
# 筛选字段等于"Underperceived" 的记录，并存储为 under_stim
under_stim <- stim_study4 %>% filter(over_under == "Underperceived")

# 从 data_trim 数据集中筛选出 condition 字段等于"High Overperception" 的记录，并存储为 over
over <- data_trim_study4 %>% filter(condition == "High Overperception")
# 筛选出 condition 字段等于"Low Overperception" 的记录，并存储为 under
under <- data_trim_study4 %>% filter(condition == "Low Overperception")

# 单样本 t 检验和 Cohen's d 计算
# 对高过度感知组的 network_outrage 变量进行单样本 t 检验
# 比较其均值与 over_stim 中 or_mean 变量的均值是否存在显著差异
t.test(over$network_outrage, mu = mean(over_stim$or_mean), alternative = "two.sided")
```

```
    One Sample t-test

data:  over$network_outrage
t = 8.0562, df = 264, p-value = 0.00000000000002726
alternative hypothesis: true mean is not equal to 5.298027
95 percent confidence interval:
 5.694423 5.950860
sample estimates:
mean of x
 5.822642
```

```r
cohensD(over$network_outrage, mu = mean(over_stim$or_mean)) #Cohen's d 效应量
```

```
[1] 0.4948911
```

```r
# 低过度感知组
t.test(under$network_outrage, mu = mean(under_stim$or_mean), alternative = "two.sided")
```

```
    One Sample t-test

data:  under$network_outrage
t = 1.3797, df = 252, p-value = 0.1689
alternative hypothesis: true mean is not equal to 3.409379
```

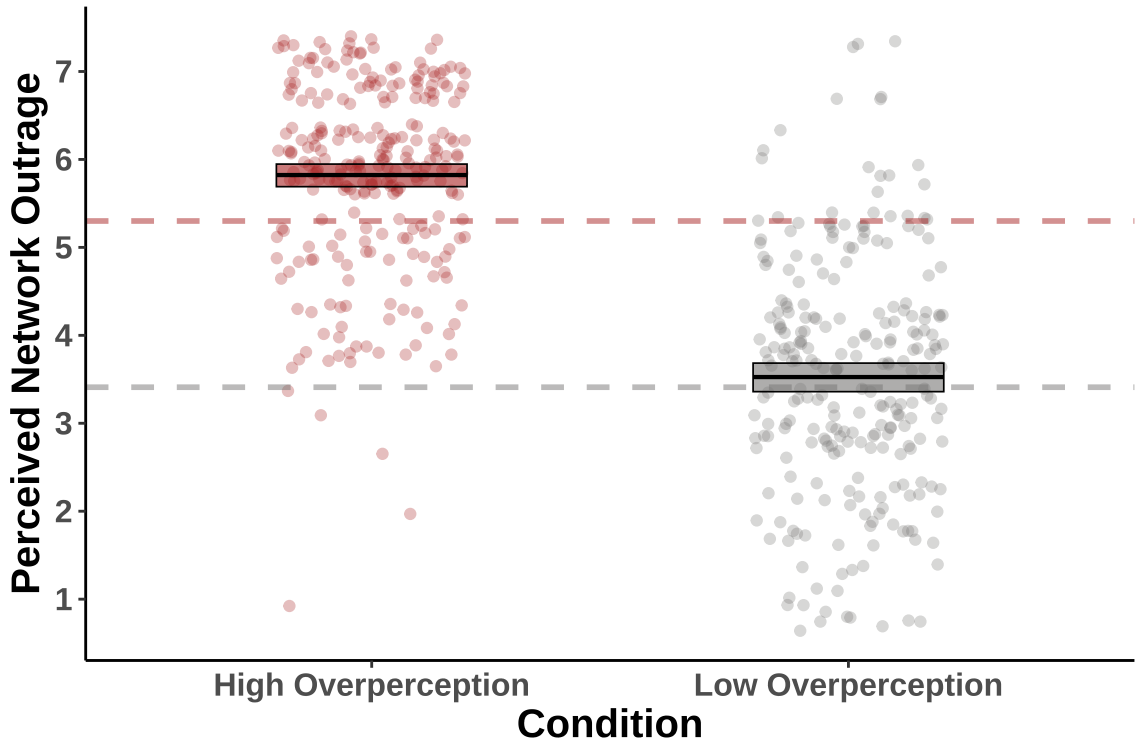
```
95 percent confidence interval:
 3.359669 3.691714
sample estimates:
mean of x
 3.525692
```

```r
cohensD(under$network_outrage, mu = mean(under_stim$or_mean))
```

```
[1] 0.08674411
```

**plot group outrage judgments —-可视化**

```r
data_trim_study4 %>%
  ggplot(aes(x = condition_fac, y = network_outrage, fill = condition_fac)) + #x 轴代表条件，y 轴代
  geom_jitter(aes(color = condition_fac),
              alpha = .3,
              width = .2) +
  stat_summary(fun.data = "mean_cl_boot", geom = "crossbar",
               position = position_dodge(width=1),
               size=.3, width=.4, alpha=.6) +
  xlab("Condition") +
  ylab("Perceived Network Outrage") +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line()) +
  scale_fill_manual(values=c("#a82424", "#787776")) +
  scale_color_manual(values=c("#a82424", "#787776")) +
  scale_y_continuous(breaks = seq(1, 7, by = 1)) +
  geom_hline(yintercept = 5.30, linetype = "dashed", color = "#a82424", size = 1, alpha = .5) +
  geom_hline(yintercept = 3.41, linetype = "dashed", color = "#787776", size = 1, alpha = .5) +# 添
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  theme(text=element_text(size = 15, face = 'bold')) +
  #theme(legend.position = c(.82, .92)) +
  theme(legend.position = "none") +
  theme(legend.title = element_blank())
```




## 研究 5

读取文件和数据清洗

```
# 读取文件
data_study5 <- read_csv("../osfstorage-archive/Data/study5_data_raw.csv")

# 计算原始数据中的观测数
data_study5 %>% group_by(id) %>% dplyr::summarize(n = n()) %>% nrow(.)
```

```
[1] 1200
```

```
# 过滤数据移除非共和党或者民主党的被试
data_trim_study5 <- data_study5 %>%
  filter(political_party == "Republican" | political_party == "Democrat")

# 再次计算过滤后数据中的观测数
data_trim_study5 %>% group_by(id) %>% dplyr::summarize(n = n()) %>% nrow(.)
```

```
[1] 1100
```

```
# 移除了 87 个不符合上面 check 观测数的被试，现在被试量剩下 1013
data_trim_study5 <- data_trim_study5 %>% filter(barr_check + barrett_check == 2)

# 计算最终的观测数
data_trim_study5 %>% group_by(id) %>% dplyr::summarize(n = n()) %>% nrow(.)
```

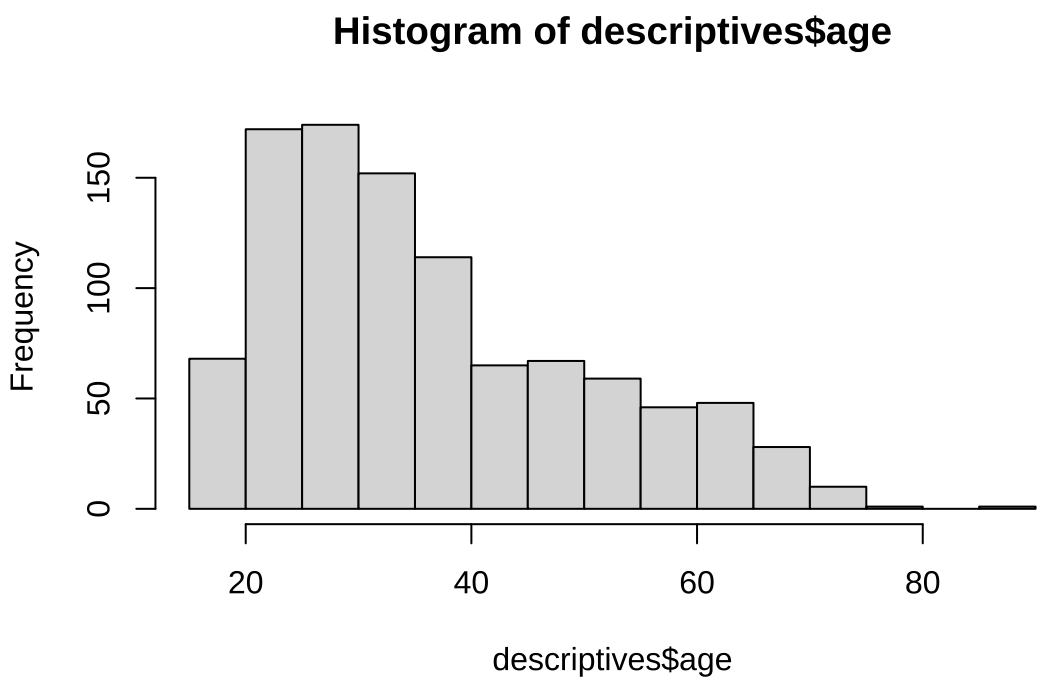
```
[1] 1013
```

### 描述统计

```
# 计算描述性统计量
descriptives <- data_trim_study5 %>% group_by(id) %>%
  summarize_all(list(first))

# 性别频率表
frequencies.table(descriptives$gender)
```

| | Freq | % |
|---|---|---|
| 21 | 1 | 0.10 |
| 22 | 1 | 0.10 |
| agender | 1 | 0.10 |
| agender/afab | 1 | 0.10 |
| Cis-female | 1 | 0.10 |
| cis female | 1 | 0.10 |
| cisgender woman | 1 | 0.10 |
| f | 4 | 0.39 |
| F | 2 | 0.20 |
| female | 298 | 29.42 |
| Female | 212 | 20.93 |
| FEMALE | 5 | 0.49 |
| Genderfluid | 1 | 0.10 |
| genderqueer | 1 | 0.10 |
| m | 3 | 0.30 |
| M | 2 | 0.20 |
| male | 195 | 19.25 |
| Male | 250 | 24.68 |
| MAle | 1 | 0.10 |
| MALE | 1 | 0.10 |
| Male (red-blooded) | 1 | 0.10 |
| MALE, XY, IT'S SCIENTIFIC, SO FOLLOW THE SCIENCE AND NOT NONSENSE | 1 | 0.10 |
| man | 3 | 0.30 |
| non-binary | 1 | 0.10 |
| Non-binary | 3 | 0.30 |
| Nonbinary | 1 | 0.10 |
| nonbinary trans woman | 1 | 0.10 |
| Transgender Female | 1 | 0.10 |
| woman | 12 | 1.18 |
| Woman | 5 | 0.49 |
| WOMAN | 1 | 0.10 |
| woman (or woman-adjacent) | 1 | 0.10 |

```
# 年龄直方图
hist(descriptives$age)
```

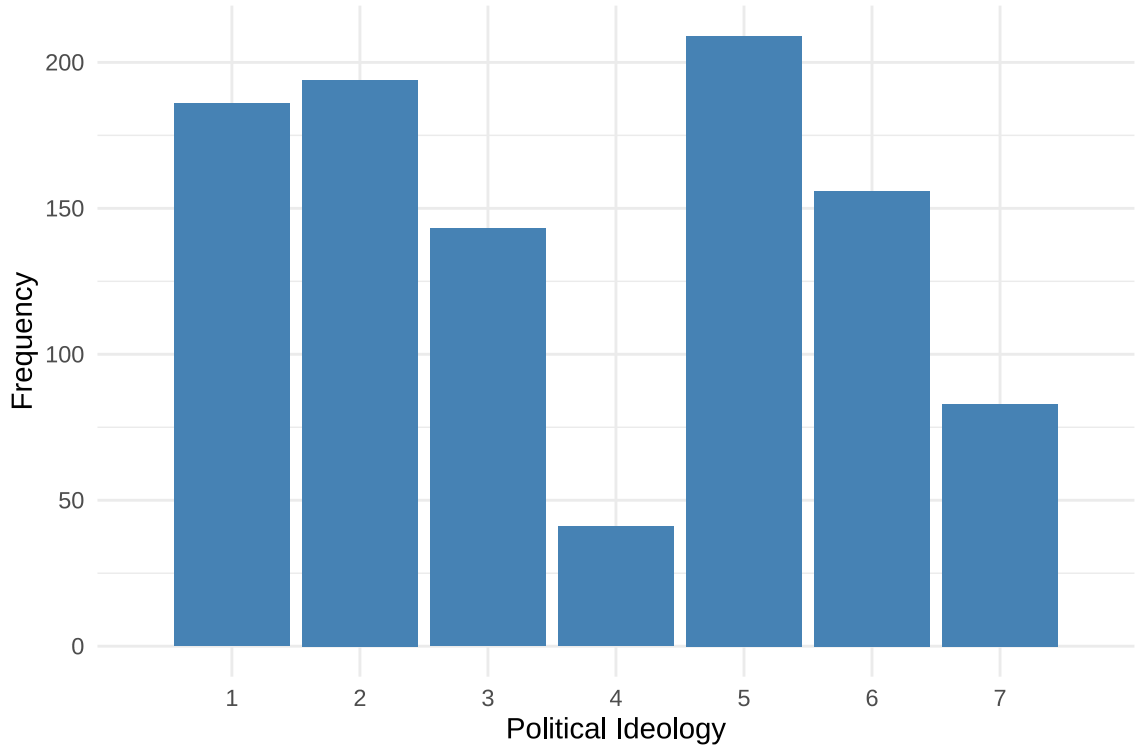



```
# 政治意识形态直方图
frequency <- descriptives %>%
  count(political_ideology)
frequency_df <- as.data.frame(frequency)
print(frequency_df)
```

```
  political_ideology   n
1                  1 186
2                  2 194
3                  3 143
4                  4  41
5                  5 209
6                  6 156
7                  7  83
8                 NA   1
```

```
ggplot(frequency, aes(x = political_ideology, y = n)) +
  geom_bar(stat = "identity", width = 0.9, fill = "steelblue") +
  scale_x_discrete(limits = 1:7) +  # 确保 x 轴包括所有水平
  theme_minimal() +
  labs(x = "Political Ideology", y = "Frequency")  # 设置轴标签
```

```
Warning in scale_x_discrete(limits = 1:7): Continuous limits supplied to discrete scale.
i Did you mean `limits = factor(...)` or `scale_*_continuous()`?

Warning: Removed 1 row containing missing values or values outside the scale range (`geom_bar()`).
```




```
  theme_minimal()  # 使用简洁的主题
```

```
List of 136
 $ line                           :List of 6
  ..$ colour       : chr "black"
  ..$ linewidth    : num 0.5
  ..$ linetype     : num 1
  ..$ lineend      : chr "butt"
  ..$ arrow        : logi FALSE
  ..$ inherit.blank: logi TRUE
  ..- attr(*, "class")= chr [1:2] "element_line" "element"
 $ rect                           :List of 5
  ..$ fill         : chr "white"
  ..$ colour       : chr "black"
  ..$ linewidth    : num 0.5
  ..$ linetype     : num 1
```

```
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_rect" "element"
$ text                            :List of 11
 ..$ family       : chr ""
 ..$ face         : chr "plain"
 ..$ colour       : chr "black"
 ..$ size         : num 11
 ..$ hjust        : num 0.5
 ..$ vjust        : num 0.5
 ..$ angle        : num 0
 ..$ lineheight   : num 0.9
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : logi FALSE
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ title                           : NULL
$ aspect.ratio                    : NULL
$ axis.title                      : NULL
$ axis.title.x                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 2.75points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.x.top                :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
```

```
 ..$ vjust        : num 0
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 2.75points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.x.bottom             : NULL
$ axis.title.y                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : num 90
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.y.left               : NULL
$ axis.title.y.right              :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : num -90
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text                       :List of 11
```

```
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : chr "grey30"
 ..$ size         : 'rel' num 0.8
 ..$ hjust        : NULL
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 1
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 2.2points 0points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.top                :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : NULL
 ..$ vjust        : num 0
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 2.2points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
```

```
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.bottom              : NULL
$ axis.text.y                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 1
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 0points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.y.left                : NULL
$ axis.text.y.right               :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.theta                 : NULL
$ axis.text.r                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0.5
 ..$ vjust        : NULL
```

```
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 2.2points
 .. ..- attr(*, "unit")= int 8
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.ticks                      : list()
 ..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ axis.ticks.x                    : NULL
$ axis.ticks.x.top                : NULL
$ axis.ticks.x.bottom             : NULL
$ axis.ticks.y                    : NULL
$ axis.ticks.y.left               : NULL
$ axis.ticks.y.right              : NULL
$ axis.ticks.theta                : NULL
$ axis.ticks.r                    : NULL
$ axis.minor.ticks.x.top          : NULL
$ axis.minor.ticks.x.bottom       : NULL
$ axis.minor.ticks.y.left         : NULL
$ axis.minor.ticks.y.right        : NULL
$ axis.minor.ticks.theta          : NULL
$ axis.minor.ticks.r              : NULL
$ axis.ticks.length               : 'simpleUnit' num 2.75points
 ..- attr(*, "unit")= int 8
$ axis.ticks.length.x             : NULL
$ axis.ticks.length.x.top         : NULL
$ axis.ticks.length.x.bottom      : NULL
$ axis.ticks.length.y             : NULL
$ axis.ticks.length.y.left        : NULL
$ axis.ticks.length.y.right       : NULL
$ axis.ticks.length.theta         : NULL
$ axis.ticks.length.r             : NULL
$ axis.minor.ticks.length         : 'rel' num 0.75
$ axis.minor.ticks.length.x       : NULL
$ axis.minor.ticks.length.x.top   : NULL
$ axis.minor.ticks.length.x.bottom: NULL
$ axis.minor.ticks.length.y       : NULL
$ axis.minor.ticks.length.y.left  : NULL
```

```
$ axis.minor.ticks.length.y.right : NULL
$ axis.minor.ticks.length.theta   : NULL
$ axis.minor.ticks.length.r       : NULL
$ axis.line                       : list()
 ..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ axis.line.x                     : NULL
$ axis.line.x.top                 : NULL
$ axis.line.x.bottom              : NULL
$ axis.line.y                     : NULL
$ axis.line.y.left                : NULL
$ axis.line.y.right               : NULL
$ axis.line.theta                 : NULL
$ axis.line.r                     : NULL
$ legend.background               : list()
 ..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ legend.margin                   : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
 ..- attr(*, "unit")= int 8
$ legend.spacing                  : 'simpleUnit' num 11points
 ..- attr(*, "unit")= int 8
$ legend.spacing.x                : NULL
$ legend.spacing.y                : NULL
$ legend.key                      : list()
 ..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ legend.key.size                 : 'simpleUnit' num 1.2lines
 ..- attr(*, "unit")= int 3
$ legend.key.height               : NULL
$ legend.key.width                : NULL
$ legend.key.spacing              : 'simpleUnit' num 5.5points
 ..- attr(*, "unit")= int 8
$ legend.key.spacing.x            : NULL
$ legend.key.spacing.y            : NULL
$ legend.frame                    : NULL
$ legend.ticks                    : NULL
$ legend.ticks.length             : 'rel' num 0.2
$ legend.axis.line                : NULL
$ legend.text                     :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
```

```
 ..$ size         : 'rel' num 0.8
 ..$ hjust        : NULL
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ legend.text.position            : NULL
$ legend.title                    :List of 11
 ..$ family       : NULL
 ..$ face         : NULL
 ..$ colour       : NULL
 ..$ size         : NULL
 ..$ hjust        : num 0
 ..$ vjust        : NULL
 ..$ angle        : NULL
 ..$ lineheight   : NULL
 ..$ margin       : NULL
 ..$ debug        : NULL
 ..$ inherit.blank: logi TRUE
 ..- attr(*, "class")= chr [1:2] "element_text" "element"
$ legend.title.position           : NULL
$ legend.position                 : chr "right"
$ legend.position.inside          : NULL
$ legend.direction                : NULL
$ legend.byrow                    : NULL
$ legend.justification            : chr "center"
$ legend.justification.top        : NULL
$ legend.justification.bottom     : NULL
$ legend.justification.left       : NULL
$ legend.justification.right      : NULL
$ legend.justification.inside     : NULL
$ legend.location                 : NULL
$ legend.box                      : NULL
$ legend.box.just                 : NULL
$ legend.box.margin               : 'margin' num [1:4] 0cm 0cm 0cm 0cm
 ..- attr(*, "unit")= int 1
```

```
 $ legend.box.background           : list()
  ..- attr(*, "class")= chr [1:2] "element_blank" "element"
 $ legend.box.spacing              : 'simpleUnit' num 11points
  ..- attr(*, "unit")= int 8
  [list output truncated]
 - attr(*, "class")= chr [1:2] "theme" "gg"
 - attr(*, "complete")= logi TRUE
 - attr(*, "validate")= logi TRUE
```

```r
# party 的频数
frequencies.table(descriptives$political_party)
```

| | Freq | % |
|---|---|---|
| Democrat | 533 | 52.62 |
| Republican | 480 | 47.38 |

```r
# Barr 的熟悉度的频数
frequencies.table(descriptives$barr_familiar)
```

| | Freq | % |
|---|---|---|
| -1 | 127 | 12.75 |
| -2 | 106 | 10.64 |
| -3 Not at all familiar | 117 | 11.75 |
| 0 Somewhat familiar | 249 | 25.00 |
| 1 | 150 | 15.06 |
| 2 | 140 | 14.06 |
| 3 Very familiar | 107 | 10.74 |

```r
# Barrett 的熟悉度的频数
frequencies.table(descriptives$barrett_familiar)
```

| | Freq | % |
|---|---|---|
| -1 | 36 | 3.59 |
| -2 | 36 | 3.59 |
| -3 Not at all familiar | 39 | 3.89 |
| 0 Somewhat familiar | 183 | 18.26 |
| 1 | 159 | 15.87 |
| 2 | 243 | 24.25 |
| 3 Very familiar | 306 | 30.54 |

### 创建新变量

```
# 重新编码政治意识形态网络
descriptives <- descriptives %>%
  mutate(ideo_network_recode =
           dplyr::recode(ideo_network,
                         `1` = -3L,
                         `2`= -2L,
                         `3` = -1L,
                         `4` = 0L,
                         `5` = 1L,
                         `6` = 2L,
                         `7` = 3L))

# 创建自我党派和其他党派的情感温度评分
# 如果是"Democrat"，那么 ownparty_temp 列使用 dem_network_temp 列的值
# 如果不是民主党人，那么将使用 rep_network_temp 列的值。
descriptives <- descriptives %>%
  mutate(ownparty_temp =
           ifelse(political_party == "Democrat",
                  dem_network_temp, rep_network_temp),
         otherparty_temp =
           ifelse(political_party == "Democrat",
                  rep_network_temp, dem_network_temp),
         ideo_extr_network = abs(ideo_network_recode))


# 为绘图设置因子
descriptives <- descriptives %>%
  mutate(condition_fac =
           factor(condition, levels =
                    c("High Overperception", "Low Overperception")))


# 过滤数据集，只保留民主党和共和党参与者的数据，然后按条件分组，并计算三个变量的均值
descriptives_plot <- descriptives %>%
  filter(political_party == "Democrat" | political_party == "Republican") %>%
  group_by(condition) %>%
  dplyr::summarize(ownparty_temp = mean(ownparty_temp),
```

```
                    otherparty_temp = mean(otherparty_temp),
                    ideo_extr_network = mean(ideo_extr_network))
```

画图前对数据框的一些整理

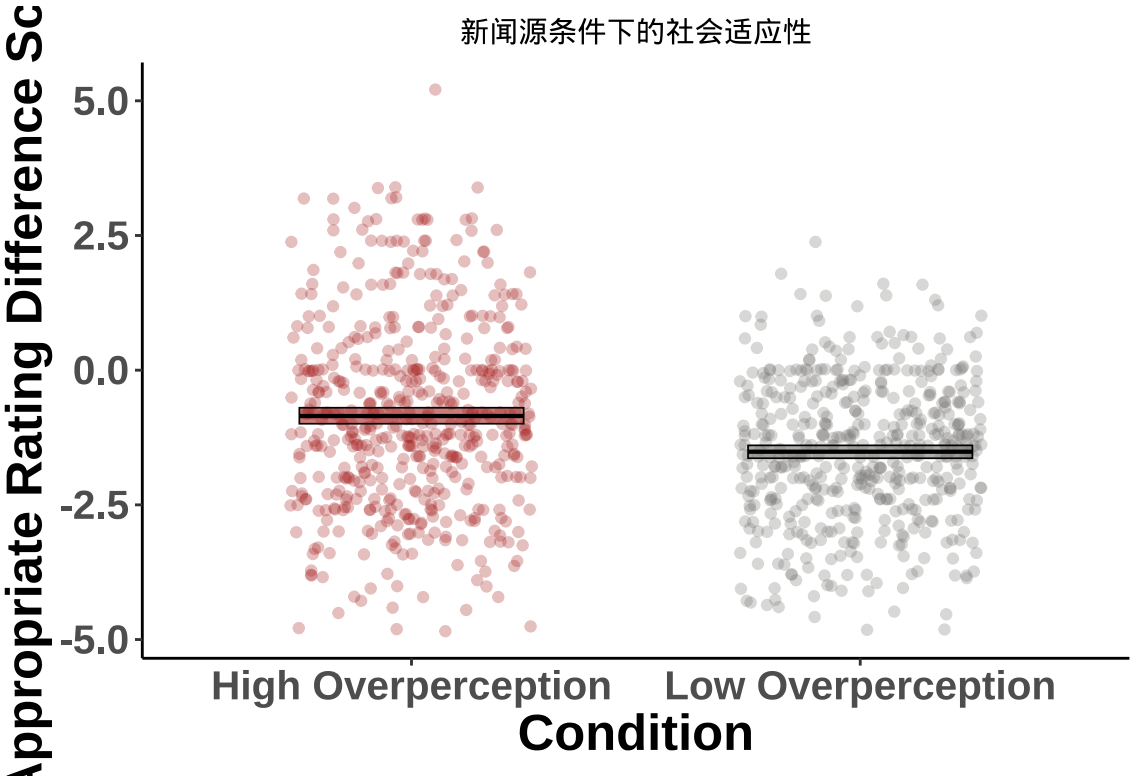
```
# 推文的分组替换成更加简洁的标签
data_trim_study5 <- data_trim_study5 %>%
  mutate(norm_stim_label_group =
           case_when(grepl("dem_high", norm_stim) ~ "dem_high",
                     grepl("dem_low", norm_stim ) ~"dem_low",
                     grepl("rep_high", norm_stim ) ~"rep_high",
                     grepl("rep_low", norm_stim ) ~"rep_low"))

# remove author tweet who opted out after experiment was run
# 移除在实验运行后选择退出的作者的推文
data_trim_study5 <- data_trim_study5 %>% filter(norm_stim != "rep_high_64_1")


stim_approp <- data_trim_study5 %>%
  filter(political_party == "Democrat" | political_party == "Republican") %>%
  group_by(id, condition, norm_stim_label_group) %>%
  dplyr::summarize(mean = mean(appropriate_rating))

# spread() 函数将 stim_approp 数据框从长格式转换为宽格式
stim_approp_w <- spread(stim_approp, norm_stim_label_group, mean) %>%
  mutate(dem_diff = dem_high - dem_low,
         rep_diff = rep_high - rep_low) %>%
  mutate(diff = ifelse(is.na(dem_diff), rep_diff, dem_diff)) %>%
  ungroup() %>%
  select(id, diff)


stim_approp <- stim_approp %>%
  left_join(stim_approp_w, by = "id")

stim_approp_plot <- stim_approp %>%
  group_by(id) %>% dplyr::summarize(condition = first(condition),
                                    diff = first(diff))
```

```r
stim_approp_plot <- stim_approp_plot %>%
  mutate(condition_fac =
           factor(condition, levels =
                    c("High Overperception", "Low Overperception")))
```

画图

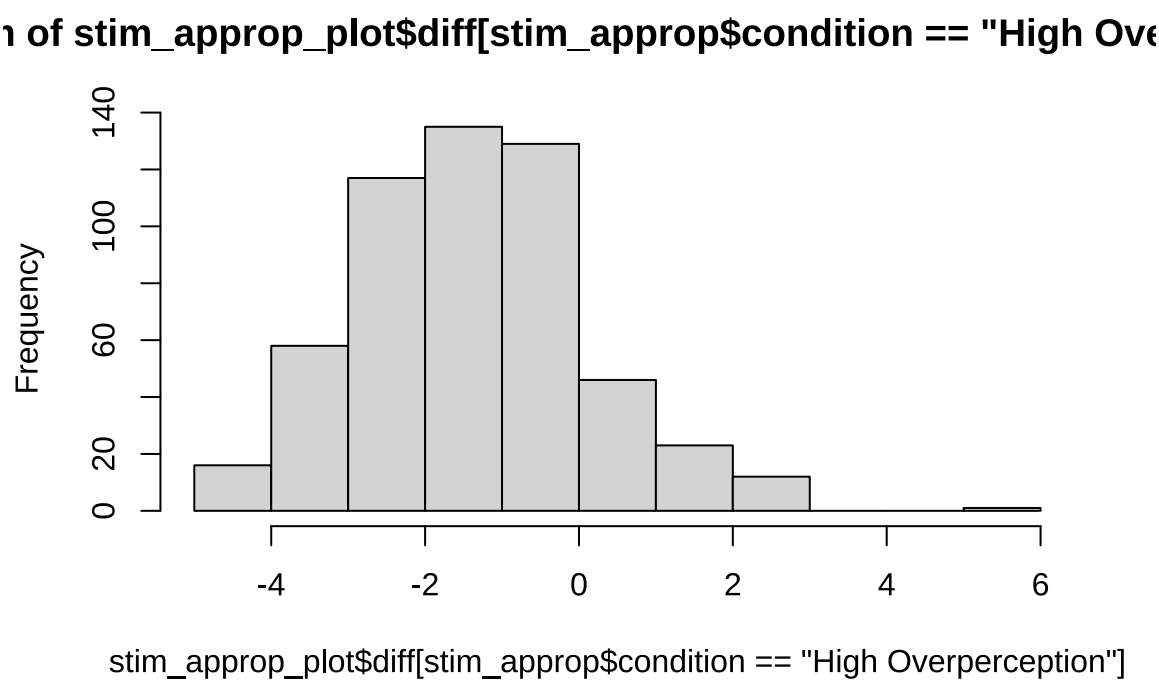
**x：高低感知 y：评分差 x 轴代表条件（condition_fac）也就是接受的信息是高愤怒感知和低愤怒感知组，y 轴代表适当性评分差异分数（diff）**

```r
# plot
stim_approp_plot %>%
  ggplot(aes(x = condition_fac, y = diff, fill = condition_fac)) +
  geom_jitter(aes(color = condition_fac),
              alpha = .3,
              width = .27) +
  stat_summary(fun.data = "mean_cl_boot", geom = "crossbar",
               position = position_dodge(width=1),
               size=.3, width=.5, alpha=.6) +
  xlab("Condition") +
  ylab("Appropriate Rating Difference Score") +
  ggtitle(" 新闻源条件下的社会适应性") +
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line()) +
  scale_fill_manual(values=c("#a82424", "#787776")) +
  scale_color_manual(values=c("#a82424", "#787776")) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  theme(text = element_text(size = 19, face = 'bold')) +
  theme(legend.position = "none") +
  theme(legend.title = element_blank()) +
  theme(plot.title =
          element_text(size = 11, hjust = 0.5,
                       face = "bold", color = "black")) # 设置标题格式
```

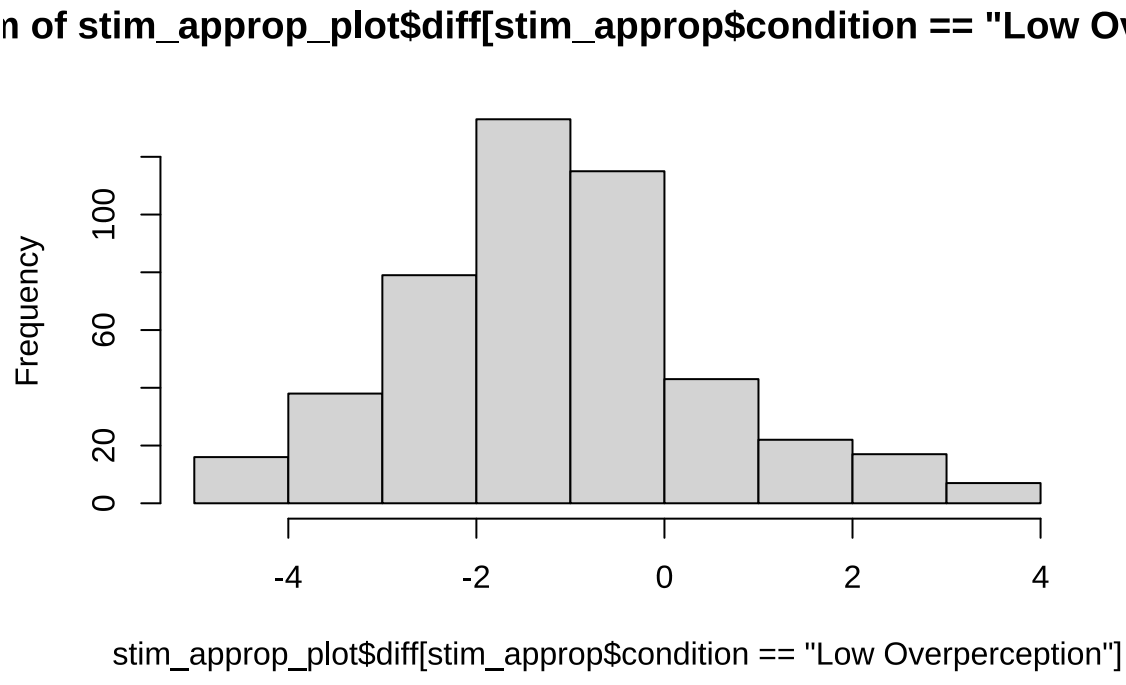



结果

```
# test network norms 检查因变量的分布----
hist(stim_approp_plot$diff[stim_approp$condition == "High Overperception"])
```




```
hist(stim_approp_plot$diff[stim_approp$condition == "Low Overperception"])
```



```r
# test for homogeneity of variance (violated) 方差齐性检验
leveneTest(diff ~ condition, data = stim_approp_plot)
```

```
Levene's Test for Homogeneity of Variance (center = median)
        Df F value       Pr(>F)
group    1  25.061 0.0000006556 ***
      1005
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# t-test, not assume equal variances 独立样本 t 检验
t.test(diff ~ condition, data = stim_approp_plot, var.equal = FALSE)
```

```
    Welch Two Sample t-test

data:  diff by condition
t = 6.8929, df = 964.58, p-value = 0.000000000009867
alternative hypothesis: true difference in means between group High Overperception and group Low O
95 percent confidence interval:
 0.4740220 0.8513649
sample estimates:
mean in group High Overperception  mean in group Low Overperception
                       -0.8509615                        -1.5136550
```

```r
# compute cohen's d 计算效应量值
cohensD(diff ~ condition, data = stim_approp_plot)
```

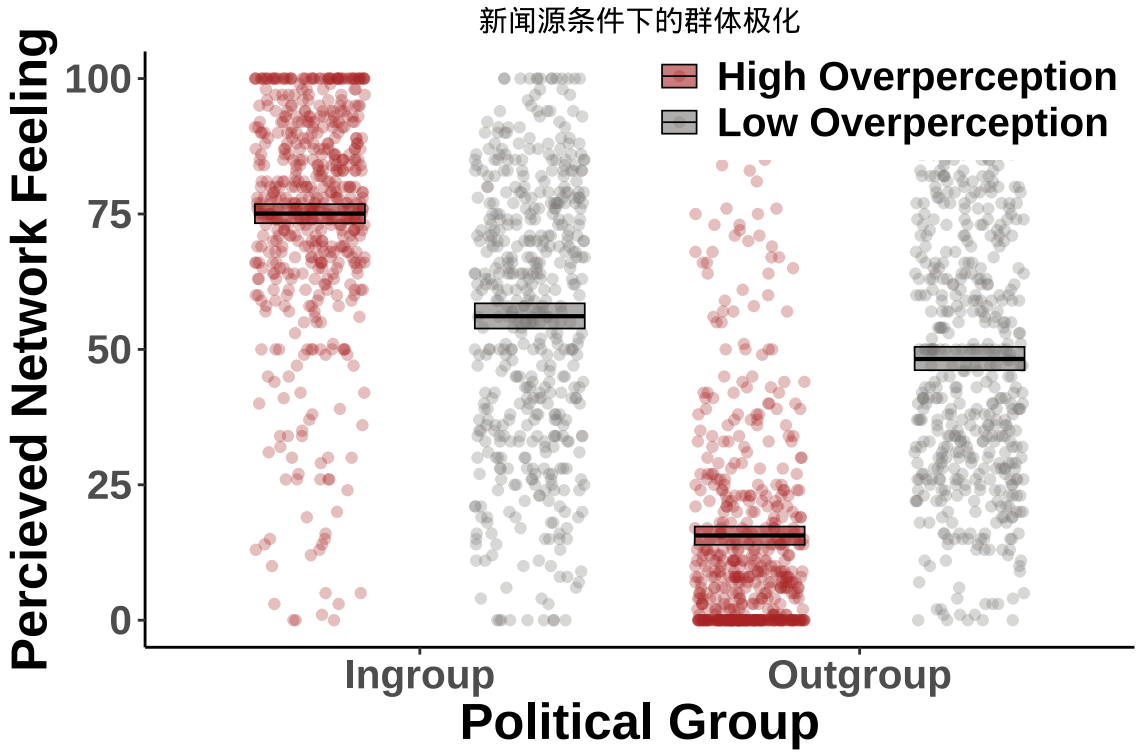
```
[1] 0.4308528
```

画图

x 轴：内外群体 y 轴：感知到的情感温度

```r
## plot network therm ----

ownparty <- descriptives %>% select(id, ownparty_temp) %>%
  rename(temp = ownparty_temp) %>%
```

```r
  mutate(network = "Ingroup") %>%
  cbind(descriptives$condition) %>%
  rename(condition = `descriptives$condition`)

otherparty <- descriptives %>% select(id, otherparty_temp) %>%
  rename(temp = otherparty_temp) %>%
  mutate(network = "Outgroup") %>%
  cbind(descriptives$condition) %>%
  rename(condition = `descriptives$condition`)

plot_therm <- rbind(ownparty, otherparty)

plot_therm <- plot_therm %>%
  mutate(condition_fac =
           factor(condition,
                  levels = c("High Overperception", "Low Overperception")))

plot_therm %>%
  ggplot(aes(x = network, y = temp, fill = condition_fac)) +
  geom_point(position = position_jitterdodge(dodge.width = 1, jitter.width = .5),
             aes(color = condition_fac, fill = condition_fac),
              alpha = .3) +
  stat_summary(fun.data = "mean_cl_boot", geom = "crossbar",
               position = position_dodge(width=1),
               size=.3, width=.5, alpha=.6) +
  xlab("Political Group") +
  ylab("Percieved Network Feeling") +
  ggtitle(" 新闻源条件下的群体极化")+
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line()) +
  scale_fill_manual(values=c("#a82424", "#787776")) +
  scale_color_manual(values=c("#a82424", "#787776")) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  theme(text=element_text(size = 19, face = 'bold')) +
  theme(legend.position = c(.75, .92)) +
  theme(legend.title = element_blank())+
  theme(plot.title = element_text(size = 11, hjust = 0.5, face = "bold", color = "black"))
```




```
class(plot_therm$condition_fac)
```

```
[1] "factor"
```

两因素混合方差分析

```
# two-way mixed ANOVA, network therm ----
anova <- anova_test(
  data = plot_therm, dv = temp, wid = id,
  between = condition, within = network, effect.size = "pes")

get_anova_table(anova)
```

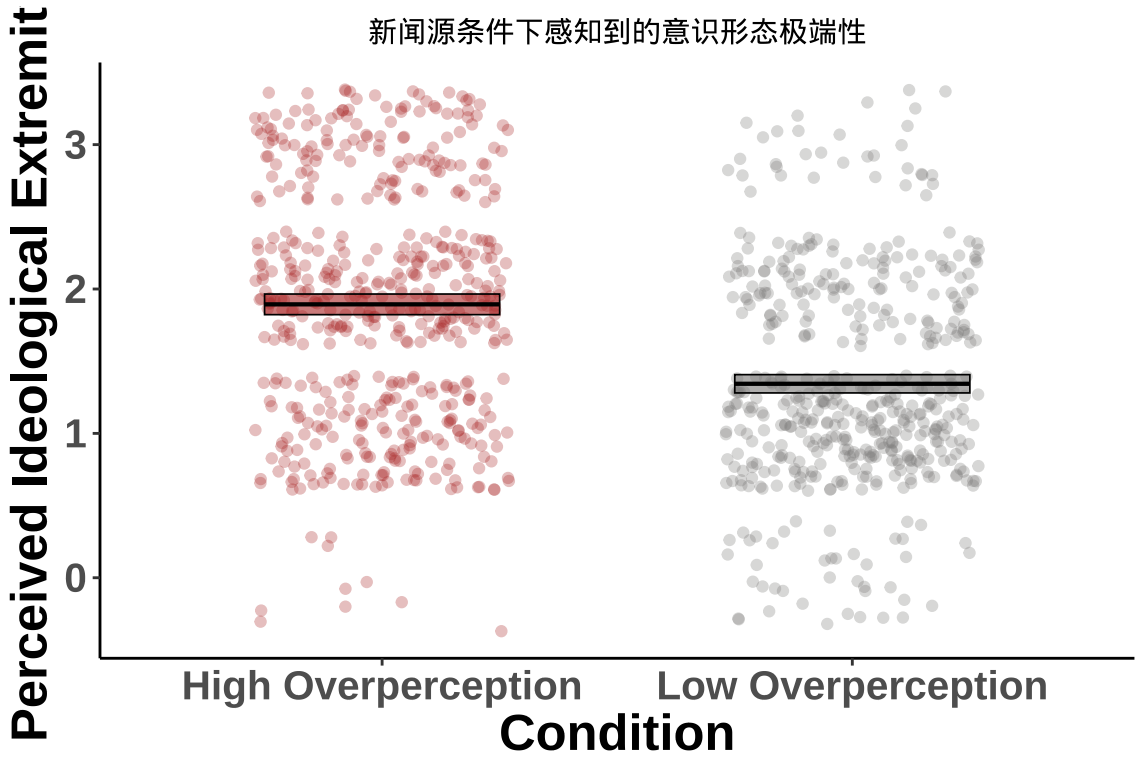
```
ANOVA Table (type III tests)

              Effect DFn  DFd       F
1          condition   1 1010 185.000
2            network   1 1010 630.697
3 condition:network   1 1010 369.577
```

```
1 0.000000000000000000000000000000000000000819999999999999586094978631933827273314818739891052246
2 0.000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
3 0.000000000000000000000000000000000000000000000000000000000000000000000000000000000196999999999999668043315
  p<.05   pes
1     * 0.155
2     * 0.384
3     * 0.268
```

画图

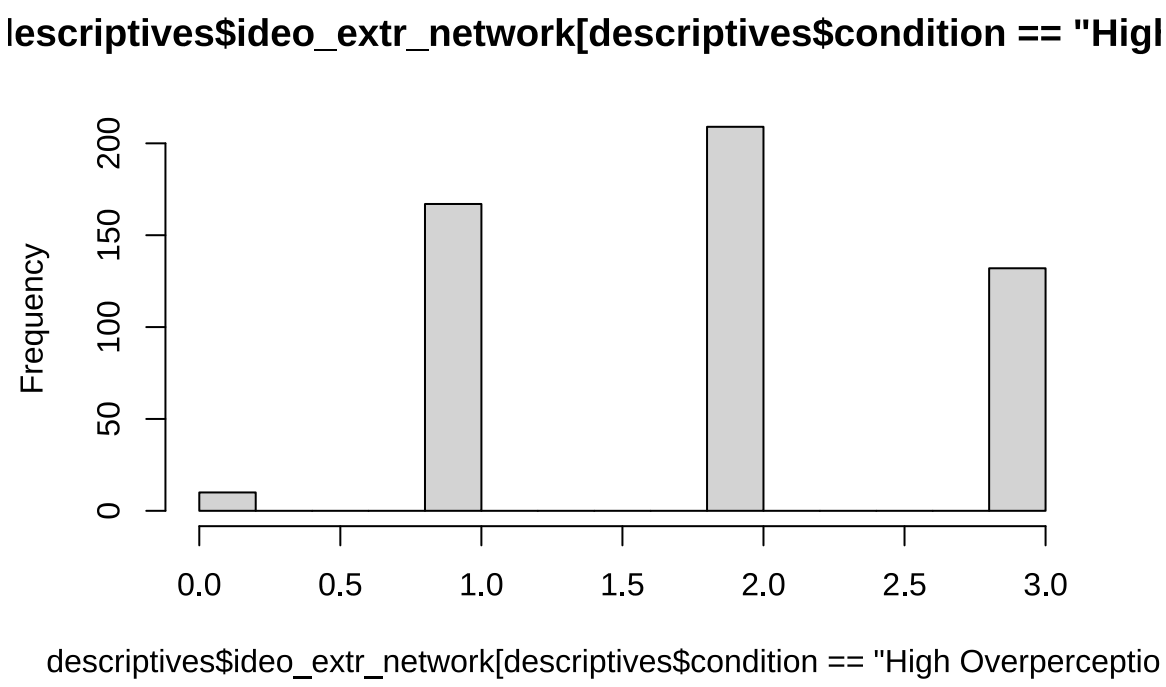
x: 内外群体 y：意识形态的极端性

```
## plot ideo_extr ----

descriptives %>%
  ggplot(aes(x = condition_fac, y = ideo_extr_network, fill = condition_fac)) +
  geom_jitter(aes(color = condition_fac),
              alpha = .3,
              width = .27) +
  stat_summary(fun.data = "mean_cl_boot", geom = "crossbar",
               position = position_dodge(width=1),
               size=.3, width=.5, alpha=.6) +
  xlab("Condition") +
  ylab("Perceived Ideological Extremity") +
  ggtitle(" 新闻源条件下感知到的意识形态极端性") + # 添加图表标题
  theme_bw() +
  theme(panel.border = element_blank(), axis.line = element_line()) +
  scale_fill_manual(values=c("#a82424", "#787776")) +
  scale_color_manual(values=c("#a82424", "#787776")) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  theme(text=element_text(size = 19, face = 'bold')) +
  theme(legend.position = "none") +
  theme(legend.title = element_blank()) +
  theme(plot.title =
          element_text(size = 11, hjust = 0.5,
                       face = "bold", color = "black")) # 设置标题格式
```

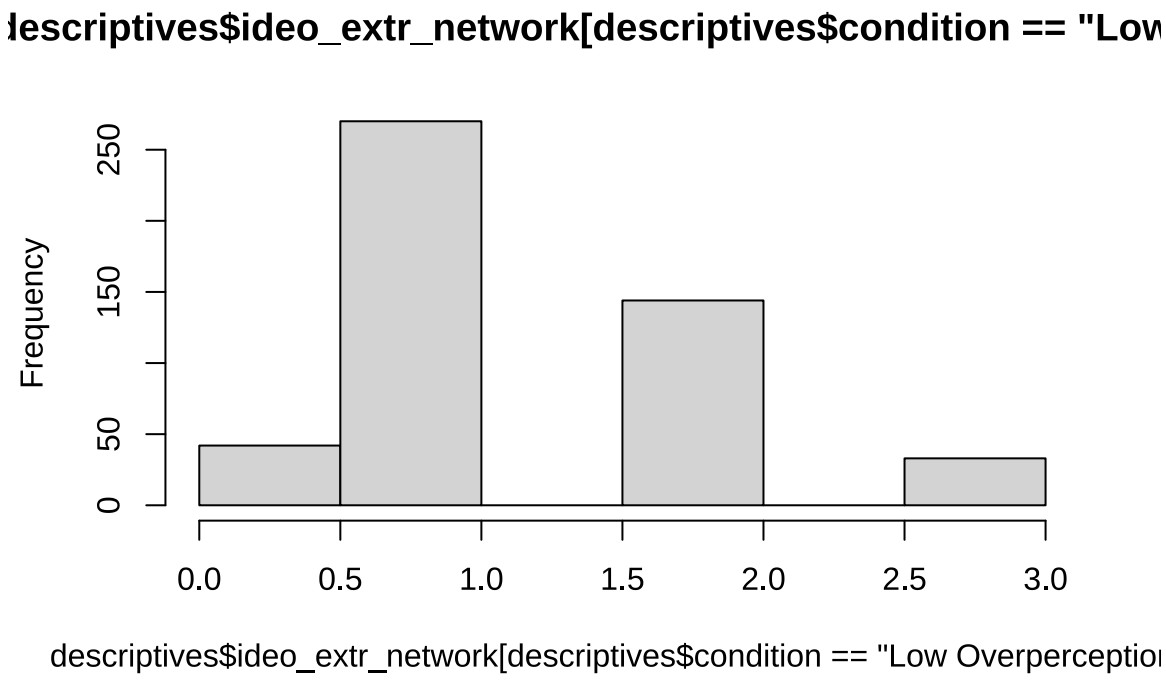



t 检验

```
# t-test ideo_extr ----

# examine distribution of DV
# overperception looks normal, note outlier in accurate perception
hist(descriptives$ideo_extr_network[descriptives$condition == "High Overperception"])
```




```r
hist(descriptives$ideo_extr_network[descriptives$condition == "Low Overperception"])
```



```r
# test for homogeneity of variance (violated)
leveneTest(ideo_extr_network ~ condition, data = descriptives)
```

```
Levene's Test for Homogeneity of Variance (center = median)
        Df F value   Pr(>F)
group    1  7.7202 0.005562 **
      1005
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# t-test, not assume equal variances
t.test(ideo_extr_network ~ condition, data = descriptives, var.equal = FALSE)
```

```
    Welch Two Sample t-test

data:  ideo_extr_network by condition
t = 11.388, df = 1003.6, p-value < 0.00000000000000022
alternative hypothesis: true difference in means between group High Overperception and group Low O
95 percent confidence interval:
 0.4554455 0.6450827
sample estimates:
mean in group High Overperception  mean in group Low Overperception
                         1.893822                          1.343558
```

```r
# compute cohen's d
cohensD(ideo_extr_network ~ condition, data = descriptives)
```

```
[1] 0.7160856
```