選舉民意調查資料分析報告

高嘉妤、柯堯珹、吳承恩、趙友誠

2024-10-07

Table of contents


```
library(haven)
library(Hmisc)
library(dplyr)
library(ggplot2)
library(MASS)
pollsav <- read_sav("poll.sav")
write.csv(pollsav, file = "poll.csv", row.names = FALSE)
pollcsv <- read.csv("poll.csv")
```

資料簡介

This is a complete data with no actual missing value while some might be labeled as missing. Dimension of the Data : ***1671 samples × 15 columns***

Table 1: 變數解釋

Variables	Explanation	remark
V1	District	1: 北區, 2: 中西區
V2、V3	Li	
V4_1~V4_8	Candidate known	1~10 號
V5	Candidate supported	1~10 號
V6	Age	1:20 到 29 歲,2:30 到 39 歲,3:40 到 49 歲,4:50 到 59 歲,5:60 歲以上
V7	Education level	1: 小學, 2: 國中, 3: 高中, 4: 專科, 5: 大學以上
V8	Sex	1:male, 2:female

Table 2: 遺失值定義

Variables	Missing
V1	98,99
V2、V3	44,98,99
V4_1~V4_8	98,99
V5	98,99
V6	6,99
V7	95,99
V8	99

有些數據在轉換時發生錯誤，因此將 91 視為與 98、99 同類。而我們發現有 42 筆從第一題開始就是遺失值的數據，因此將他們移除後，資料維度是 ***1629 samples × 15 columns***。

```
pollcsv <- data.frame(
  t(apply(pollcsv,MARGIN = 1, FUN = function(row){
    row[row==99 | row==98 | row==91 | row==95 | row==44 | row[13]==6] <- 0
    return(row)
  }))
)
pollcsv <- pollcsv[pollcsv$v1!=0,]
pollcsv[] <- lapply(pollcsv, function(item) return(as.factor(item)))
n <- dim(pollcsv)[1]   # 樣本數
```

一、分析所有候選人的知名度、支持度

指標定義:

1. 知名度 = $\frac{\text{第四題出現次數}}{\text{樣本數}}$
2. 支持度 = $\frac{\text{第五題出現次數}}{\text{有效樣本數}} = \frac{\text{第五題出現次數}}{\text{樣本數-無表態者}}$

```
# 計算 1~10 號候選人在複選題出現的次數
count4 <- unlist(lapply(factor(1:10), function(x){
  return(
    sum(unlist(
      apply(
        pollcsv[,4:11],
        MARGIN = 1,
        function(row) if(x %in% row) return (TRUE)))))
}))
# 計算 1~10 號候選人在第 5 題出現的次數
count5 <- unlist(lapply(factor(0:10),function(x){
  return(sum(pollcsv$v5==x))
} ))
# 有效樣本數
n_prime <- n-count5[1]
# 知名度 = 出現次數/sample size
p <- data.frame(factor(1:10), popularity=round(count4/n,3), count4)
# 支持度 = 出現次數/(sample size-沒回答的)
s <- data.frame(
  factor(1:10),
  support.level=round(count5[2:11]/n_prime,3),
  count5[2:11]
```

```
)
# 將候選人依據知名度與支持度排序
p <- p[order(p$popularity, decreasing = TRUE ),]
s <- s[order(s$support.level, decreasing = TRUE ),]
df.total <- cbind(p,s)
row.names(df.total) <- 1:10
# 依據區將資料分成 Dist1 與 Dist2
Dist1 <-subset(pollcsv,pollcsv$v1==1)
Dist2 <-subset(pollcsv,pollcsv$v1==2)
n1 <- dim(Dist1)[1]
n2 <- dim(Dist2)[1]
# 第一區的計算
# 候選人在複選題出現的次數
count4_1 <- unlist(lapply(factor(1:10), function(x){
  return(
    sum(unlist(
      apply(
        Dist1[,4:11],
        MARGIN = 1,
        function(row) if(x %in% row) return (TRUE)))))
}))
# 候選人在第 5 題出現的次數
count5_1 <- unlist(lapply(factor(0:10),function(x){
  return(sum(Dist1$v5==x))
} ))
n1_prime <- n1-count5_1[1]
# 知名度 = 出現次數/sample size
p_1 <- data.frame(factor(1:10), popularity=round(count4_1/n1,3), count4_1)
# 支持度 = 出現次數/(sample size-沒回答的)
s_1 <- data.frame(
  factor(1:10),
  support.level=round(count5_1[2:11]/n1_prime,3),
  count5_1[2:11]
)
# 將候選人依據知名度與支持度排序
p_1 <- p_1[order(p_1$popularity, decreasing = TRUE ),]
s_1 <- s_1[order(s_1$support.level, decreasing = TRUE ),]
df.R1 <- cbind(p_1,s_1)
row.names(df.R1) <- 1:10
# 第二區的計算
# 候選人在複選題出現的次數
count4_2 <- unlist(lapply(factor(1:10), function(x){
  return(
    sum(unlist(
      apply(
        Dist2[,4:11],
        MARGIN = 1,
        function(row) if(x %in% row) return (TRUE)))))
}))
# 候選人在第 5 題出現的次數
count5_2 <- unlist(lapply(factor(0:10),function(x){
  return(sum(Dist2$v5==x))
} ))
```

Table 3: 第一區候選人知名度與支持度

	號碼	知名度	計數	號碼	支持度	計數
1	3	0.293	317	3	0.297	163
2	7	0.228	247	7	0.259	142
3	6	0.180	195	6	0.151	83
4	1	0.160	173	1	0.100	55
5	4	0.098	106	4	0.058	32
6	5	0.089	96	10	0.055	30
7	10	0.064	69	5	0.053	29
8	8	0.014	15	2	0.013	7
9	9	0.013	14	9	0.007	4
10	2	0.006	7	8	0.005	3

```
n2_prime <- n2-count5_2[1]
# 知名度 = 出現次數/sample size
p_2 <- data.frame(factor(1:10), popularity=round(count4_2/n2,3), count4_2)
# 支持度 = 出現次數/(sample size-沒回答的)
s_2 <- data.frame(
  factor(1:10),
  support.level=round(count5_2[2:11]/n2_prime,3),
  count5_2[2:11]
)
# 將候選人依據知名度與支持度排序
p_2 <- p_2[order(p_2$popularity, decreasing = TRUE ),]
s_2 <- s_2[order(s_2$support.level, decreasing = TRUE ),]
df.R2 <- cbind(p_2,s_2)
row.names(df.R2) <- 1:10
```

```
alist <- list(df.R1, df.R2, df.total)
blist <- c(" 第一區候選人知名度與支持度"," 第二區候選人知名度與支持度"," 兩區合併候選人知名度與支持度")
for(i in 1:3){
  latex(
    alist[[i]],
    title="",
    file = "",
    caption = blist[i],
    booktabs = TRUE,
    colheads = c('號碼','知名度', '計數'," 號碼",'支持度','計數')
  )
}
```

表三至表五是所有候選人的分區及合併之知名度與支持度。

二、候選人 3 的知名度與支持度

在第二題，我們將用年齡、性別、居住里和教育水平來對這筆資料進行分層，用以觀察 3 號候選人在不同情況下的知名度和支持度，特別注意我們還計算了"supportknown" 此項變數，來計算選票轉換率 (支持度/知名度)，也就是在所有知道 3 號候選人的選民中，支持他的有多少人，用來進行更好的競選策略。

Table 4: 第二區候選人知名度與支持度

	號碼	知名度	計數	號碼	支持度	計數
1	1	0.282	154	1	0.347	102
2	7	0.168	92	7	0.177	52
3	3	0.165	90	4	0.160	47
4	4	0.110	60	3	0.143	42
5	10	0.086	47	10	0.078	23
6	6	0.068	37	6	0.051	15
7	5	0.055	30	5	0.014	4
8	9	0.018	10	9	0.014	4
9	8	0.015	8	8	0.010	3
10	2	0.007	4	2	0.007	2

Table 5: 兩區合併候選人知名度與支持度

	號碼	知名度	計數	號碼	支持度	計數
1	3	0.250	407	3	0.243	205
2	7	0.208	339	7	0.230	194
3	1	0.201	327	1	0.186	157
4	6	0.142	232	6	0.116	98
5	4	0.102	166	4	0.094	79
6	5	0.077	126	10	0.063	53
7	10	0.071	116	5	0.039	33
8	9	0.015	24	2	0.011	9
9	8	0.014	23	9	0.010	8
10	2	0.007	11	8	0.007	6

Table 6: 候選人 3 依年齡分層之知名度與支持度

	年齡	該年齡之樣本數	知名度計數	知名度	支持度計數	支持度	supportknown
1	20-29	52	6	0.12	5	0.10	0.83
2	30-39	94	23	0.24	12	0.13	0.54
3	40-49	201	62	0.31	31	0.15	0.48
4	50-59	336	100	0.30	52	0.15	0.50
5	60+	946	216	0.23	105	0.11	0.48

年齡分層

```
# 知名度
knownC3 <- data.frame(
  yes_no = apply(
    pollcsv[,4:11], 1, function(row){
      if("3" %in% row){return(1)}
      else{return(0)}}
  ),
  age = pollcsv$v6
)
# 支持度
pollC3 <- data.frame(
  yes_no = unlist(
    lapply(pollcsv[,12], function(x){
      if(x=="3"){return(1)}
      else{return(0)}
  })),
  age = pollcsv$v6
)
Age_Stratified <- data.frame(
  知名度計數 = t(table(knownC3))[,2],
  支持度計數 = t(table(pollC3))[,2],
  num_total = table(pollcsv$v6)
)
Age_Stratified[,3] <- NULL

Age_Stratified$知名度 <-
  round(Age_Stratified$知名度計數/Age_Stratified$num_total.Freq,2)
Age_Stratified$支持度 <-
  round(Age_Stratified$支持度計數/Age_Stratified$num_total.Freq,2)
Age_Stratified$supportknown <-
  round(Age_Stratified$支持度/Age_Stratified$知名度,2)
Age_Stratified <- Age_Stratified%>%
  rename(
    '該年齡之樣本數'='num_total.Freq'
  )
Age_Stratified$年齡 <- c("20-29","30-39","40-49","50-59","60+")
Age_Stratified <- Age_Stratified[,c(7,3,1,4,2,5,6)]
latex(data.table::data.table(Age_Stratified),title="",file = "",caption = " 候選人 3 依年齡分層之知名度與
```

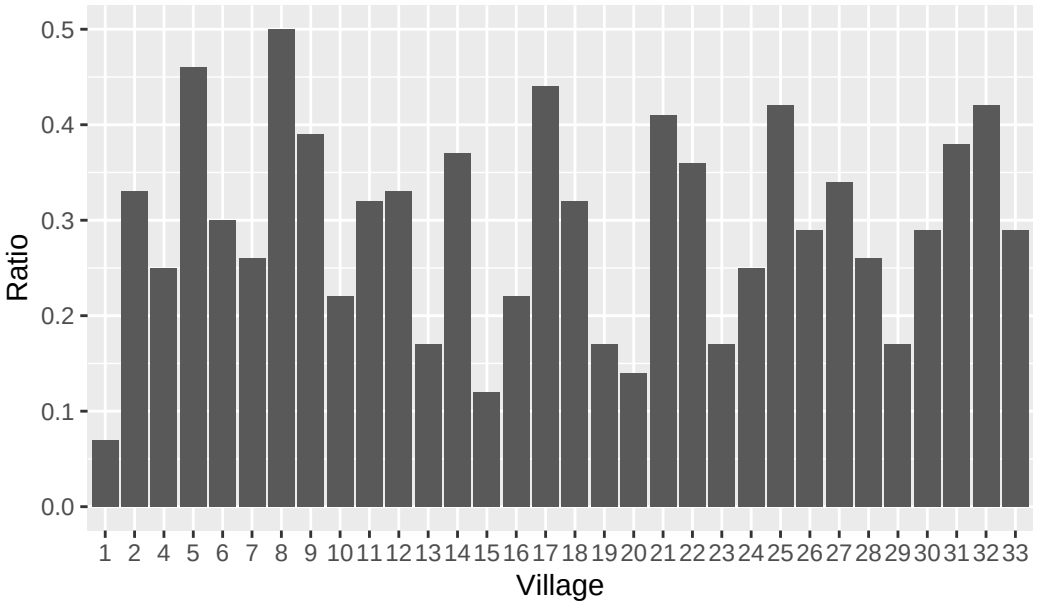
依照年齡分層的知名度與支持度如表六所示:

30 歲以下的支持度有效轉換率比其他年齡層高上不少，因此可以致力於提升在年輕族群中的知名度。在網路社群方面下功夫應該是不錯的投資。

里分層

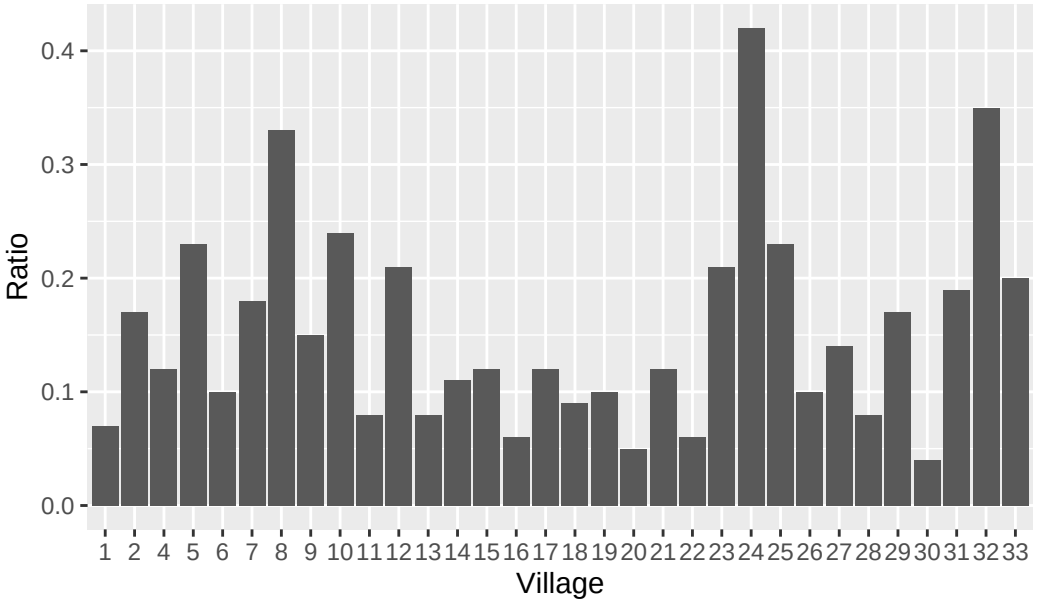
```
# 將 v4_1~v4_10 有 3 的新增 True False
pollcsv$has_3 <- apply(pollcsv[, 4:11], 1, function(row) any(row == 3))
library(dplyr)
library(ggplot2)
# 北區人
north_number <- pollcsv %>% count(v2)
north_number <- subset(north_number,v2!=0)
# 中西區人
west_number <- pollcsv %>% count(v3)
west_number <- subset(west_number,v3!=0)

# 知名 北區
result1_3_2 <- pollcsv%>%
  filter(has_3 == TRUE)%>%
  count(v2)
result1_3_2 <- subset(result1_3_2, v2 != 0)
# 知名 中西
result1_3_3 <-pollcsv%>%
  filter(has_3==TRUE)%>%
  count(v3)
result1_3_3 <- subset(result1_3_3,v3!=0)
# 支持 北
resultv5_v2 <-pollcsv%>%
  filter(v5 ==3)%>%
  count(v2)
resultv5_v2 <- subset(resultv5_v2,v2!=0)
# 支持 中西
resultv5_v3 <-pollcsv%>%
  filter(v5 ==3)%>%
  count(v3)
resultv5_v3 <- subset(resultv5_v3,v3!=0)
# 北區相除
merged_north <- merge(
  north_number,
  result1_3_2,
  by = "v2", suffixes = c("_north", "_known"))
merged_north <- merge(
  merged_north,
  resultv5_v2,
  by = 'v2')

merged_north$ratio_known <- round(merged_north$n_known / merged_north$n_north,2)
merged_north$ratio_support <- round(merged_north$n / merged_north$n_north,2)

ggplot(merged_north, aes(x = v2, y = ratio_known))+
  geom_bar(stat = 'identity')+
  scale_x_discrete(breaks = 1:44)+
  labs(title = 'Fig 1:North with known ratio in village', x = 'Village', y = 'Ratio')+
  theme(plot.title = element_text(hjust = 0.5))
```

Fig 1:North with known ratio in village




```
ggplot(merged_north, aes(x = v2, y = ratio_support))+
  geom_bar(stat = 'identity')+
  scale_x_discrete(breaks = 1:44)+
  labs(title = 'Fig 2:North with support ratio in village', x = 'Village', y = 'Ratio')+
  theme(plot.title = element_text(hjust = 0.5))
```

Fig 2:North with support ratio in village




```
# 中西相除
merged_west <- merge(
  west_number, result1_3_3,
  by = "v3", suffixes = c("_west", "_known"))
merged_west <- merge(
  merged_west, resultv5_v3,
```

Table 7: 候選人 3 於中西區各里之知名度與支持度

	里 (中西區)	里樣本數	知名度計數	知名度	支持度計數	支持度	supportknown
1	1	17	2	0.12	1	0.06	0.50
2	10	15	4	0.27	1	0.07	0.26
3	11	9	2	0.22	2	0.22	1.00
4	12	29	6	0.21	2	0.07	0.33
5	13	34	7	0.21	1	0.03	0.14
6	14	17	2	0.12	1	0.06	0.50
7	16	18	2	0.11	1	0.06	0.55
8	17	43	11	0.26	8	0.19	0.73
9	18	20	5	0.25	1	0.05	0.20
10	19	34	6	0.18	5	0.15	0.83
11	2	25	8	0.32	1	0.04	0.12
12	4	51	4	0.08	5	0.10	1.25
13	5	42	8	0.19	4	0.10	0.53
14	7	14	3	0.21	2	0.14	0.67

```
  by = 'v3')
merged_west$ratio_known <- round(merged_west$n_known / merged_west$n_west,2)
merged_west$ratio_support <- round(merged_west$n / merged_west$n_west,2)
# 重新命名 west_table
west_table<-merged_west%>%
  rename(
    '里 (中西區)'='v3',
    '里樣本數'='n_west',
    '知名度計數'='n_known',
    '支持度計數'='n',
    '支持度'="ratio_support",
    " 知名度"="ratio_known"
  )
west_table$supportknown <-
  round(west_table$支持度/west_table$知名度,2)
west_table <- west_table[,c(1,2,3,5,4,6,7)]
latex(data.table::data.table(west_table),title="",file = "",caption=" 候選人 3 於中西區各里之知名度與支持
```

如圖 1、圖 2 所示:

在中西區，3 號候選人在第 1,14,16,4 里的知名度皆不高；第 2 里的知名度最高；而三號候選人在中西區的支持度大致上偏低，除了第 11 和 17 里的支持度比較高。

我們認為 3 號候選人可以鞏固第 17 里的選民，因為此里的人有高選票轉換率，而 3 號候選人不用特別針對第 2 里的選民進行拉票活動，因為在第 2 里已經有較高比率的知名度的情況下，支持度卻不高。

```
# 重新命名 north table
north_table<-merged_north%>%
  rename(
    '里 (北區)'='v2',
    '里樣本數'='n_north',
    '知名度計數'='n_known',
    '支持度計數'='n',
    '支持度'="ratio_support",
    " 知名度"="ratio_known"
  )
```

Table 8: 候選人 3 於北區各里之知名度與支持度

	里 (北區)	里樣本數	知名度計數	知名度	支持度計數	支持度	supportknown
1	1	14	1	0.07	1	0.07	14.29
2	10	37	8	0.22	9	0.24	40.91
3	11	25	8	0.32	2	0.08	6.25
4	12	24	8	0.33	5	0.21	15.15
5	13	12	2	0.17	1	0.08	5.88
6	14	19	7	0.37	2	0.11	5.41
7	15	32	4	0.12	4	0.12	33.33
8	16	18	4	0.22	1	0.06	4.55
9	17	16	7	0.44	2	0.12	4.55
10	18	22	7	0.32	2	0.09	6.25
11	19	40	7	0.17	4	0.10	23.53
12	2	12	4	0.33	2	0.17	6.06
13	20	21	3	0.14	1	0.05	7.14
14	21	41	17	0.41	5	0.12	12.20
15	22	33	12	0.36	2	0.06	5.56
16	23	29	5	0.17	6	0.21	35.29
17	24	12	3	0.25	5	0.42	20.00
18	25	31	13	0.42	7	0.23	16.67
19	26	42	12	0.29	4	0.10	13.79
20	27	56	19	0.34	8	0.14	23.53
21	28	39	10	0.26	3	0.08	11.54
22	29	12	2	0.17	2	0.17	11.76
23	30	28	8	0.29	1	0.04	3.45
24	31	37	14	0.38	7	0.19	18.42
25	32	31	13	0.42	11	0.35	26.19
26	33	41	12	0.29	8	0.20	27.59
27	4	8	2	0.25	1	0.12	4.00
28	5	35	16	0.46	8	0.23	17.39
29	6	10	3	0.30	1	0.10	3.33
30	7	38	10	0.26	7	0.18	26.92
31	8	70	35	0.50	23	0.33	46.00
32	9	33	13	0.39	5	0.15	12.82

```
north_table$supportknown <- round(north_table$支持度計數/north_table$知名度,2)
north_table <- north_table[,c(1,2,3,5,4,6,7)]
latex(data.table::data.table(north_table),title="",file = "",caption=" 候選人 3 於北區各里之知名度與支持度
```

如表 7、表 8 所示，在北區中，3 號候選人在第 8 里的知名度最高，在第 1 里的支持度最低。而 3 號候選人在第 24 里的支持度最高，第 30 里的支持度最低。

我們認為 3 號候選人可以鞏固第 8 里的選民，因為第 8 里的支持度和知名度都偏高，而第 22 和 30 里不用進行特別的拜票活動，因為這兩個里的支持度和知名度都偏低。

性別分層

```
sex1 <-subset(pollcsv,pollcsv$v8==1)# 男生
sex2 <-subset(pollcsv,pollcsv$v8==2)# 女生
n3 <- dim(sex1)[1]
n4 <- dim(sex2)[1]
# 男生的計算
```

```
count4_sex1 <- unlist(lapply(factor(1:10), function(x){
  return(
    sum(unlist(
      apply(
        sex1[,4:11],
        MARGIN = 1,
        function(row) if(x %in% row) return (TRUE)))))
}))
count5_sex1 <- unlist(lapply(factor(0:10),function(x){
  return(sum(sex1$v5==x))
} ))
n3_prime <- n3-count5_sex1[1]
# 知名度 = 出現次數/sample size
p_sex1 <- data.frame(
  factor(1:10),
  popularity=round(count4_sex1/n3,3),
  count4_sex1
)
# 支持度 = 出現次數/(sample size-沒回答的)
s_sex1 <- data.frame(
  factor(1:10),
  support.level=round(count5_sex1[2:11]/n3_prime,3),
  count5_sex1[2:11]
)
# 女生的計算
# 候選人在複選題出現的次數
count4_sex2 <- unlist(lapply(factor(1:10), function(x){
  return(
    sum(unlist(
      apply(
        sex2[,4:11],
        MARGIN = 1,
        function(row) if(x %in% row) return (TRUE)))))
}))
# 候選人在第 5 題出現的次數
count5_sex2 <- unlist(lapply(factor(0:10),function(x){
  return(sum(sex2$v5==x))
} ))
n4_prime <- n4-count5_sex2[1]
# 知名度 = 出現次數/sample size
p_sex2 <- data.frame(
  factor(1:10),
  popularity=round(count4_sex2/n4,3),
  count4_sex2
)
# 支持度 = 出現次數/(sample size-沒回答的)
s_sex2 <- data.frame(
  factor(1:10),
  support.level=round(count5_sex2[2:11]/n4_prime,3),
  count5_sex2[2:11]
)
data1=data.frame(
  sex=c('male','female'),
```

Table 9: 候選人 3 依性別分層之知名度與支持度

	性別	該性別樣本數	知名度計數	知名度	支持度計數	支持度	supportknown
1	male	668	190	0.28	88	0.13	0.46
2	female	961	217	0.23	117	0.12	0.54

```
  total = c(n3,n4),
  popularity_count=c(
    count4_sex1[3],
    count4_sex2[3]),
  popularity_ratio=c(
    round(count4_sex1[3]/n3,2),
    round(count4_sex2[3]/n4,2)),
  support_count=c(
    count5_sex1[4],
    count5_sex2[4]),
  support_ratio=c(
    round(count5_sex1[4]/n3,2),
    round(count5_sex2[4]/n4,2)),
  change_ratio=c(
    round(count5_sex1[4]/count4_sex1[3],2),
    round(count5_sex2[4]/count4_sex2[3],2)))
latex(
  data1,
  title="",file = "",caption = " 候選人 3 依性別分層之知名度與支持度",
  booktabs = TRUE,
  colheads = c('性別','該性別樣本數','知名度計數',
               '知名度','支持度計數','支持度',
               'supportknown')
)
```

根據表 9 可以發現，3 號候選人在男性選民中的知名度是大於女性選民的，但男性選民的支持度卻小於女性選民，在所有認識 3 號候選人的男性選民中，有 46.3% 的人支持他當選；而在所有認識 3 號候選人的女性選民中，有 53.4% 的人支持他當選。

因此建議 3 號候選人可以多多向女性選民進行拉票，以此拉高在女性選民中的知名度，並藉此來提高支持度。

教育程度分層

```
# 支持三號的教育程度和人數
support3toedulevel <- pollcsv[pollcsv$v5 == 3,"v7"]
supportcount<-summary(support3toedulevel)

# 知道三號的教育程度和人數
known3toedulevel <- apply(pollcsv[,4:11],1,function(row)any(row==3))
k32edu <- pollcsv[known3toedulevel,14]
popularitycount <- summary(k32edu)

totalnumber <- summary(pollcsv$v7)
edu_table <- data.frame(
  cbind(totalnumber,popularitycount,supportcount)
)
edu_table$popularityratio <- round(edu_table$popularitycount/edu_table$totalnumber ,2)
```

Table 10: 候選人 3 依學歷分層之知名度與支持度

	教育程度	樣本數	知名度計數	知名度	支持度計數	支持度	supportknown
1	小學	291	51	0.18	27	0.09	0.53
2	國中	164	40	0.24	26	0.16	0.65
3	高中	429	121	0.28	64	0.15	0.53
4	專科	198	53	0.27	31	0.16	0.58
5	大學及以上	519	141	0.27	57	0.11	0.40

```
edu_table$supportratio <- round(edu_table$supportcount/edu_table$totalnumber ,2)
edu_table$supportknown <- round(edu_table$supportcount/edu_table$popularitycount,2)
edu_table <- edu_table[2:6,]
edu_table$edulevel <- c(" 小學"," 國中"," 高中"," 專科"," 大學及以上")
edu_table <- edu_table[,c(7,1,2,4,3,5,6)]
latex(data.table::data.table(edu_table),
      title="",file = "", caption=" 候選人 3 依學歷分層之知名度與支持度",
      colheads=c(" 教育程度"," 樣本數"," 知名度計數"," 知名度"," 支持度計數"," 支持度","supportknown"))
```

如表 10 所示:

3 號候選人在各教育水平情況下的知名度和支持度皆相差不大，因此不需要針對不同的教育水平進行不同的競選策略。

總結來說，我們認為 3 號候選人可以用特定的競選方法來針對不同族群，像是可以舉辦文創市集來針對年輕族群和女性族群，或是在網路上投放廣告和進行網路宣傳活動來吸引年輕族群的票，或是針對特定的里來讓喜愛他的選民獲得更多關注，透過街訪鄰居的力量獲得特定里裡面的更多票。

三、候選人 3 的支持率預測模式

資料切割

在分割資料後，由於每個變數都是類別變數，因此我們對每個變數進行卡方同質性檢定，以此檢視資料分割的品質:

在信心水準為 0.05 之下，若拒絕虛無假設，則有足夠證據支持訓練集與測試集在此變數上結構不同。檢定結果如表 11 所示。

```
mydata <- pollcsv[,c(1,2,3,13,14,15,16)]
mydata$support <- ifelse(pollcsv$v5=="3",1,0)
mydata$support <- factor(mydata$support)
```

```
library(caret)
start = 1
yes3 <- mydata[mydata$has_3,]
no3 <- mydata[!mydata$has_3,]
set.seed(i)
yes_index <- createDataPartition(yes3$support, p = 0.8, list = FALSE)
no_index <- createDataPartition(no3$support, p = 0.8, list = FALSE)
yes3_train_data <- mydata[yes_index, ]  # yes 訓練集
yes3_test_data <- mydata[-yes_index, ]  # yes 測試集
no3_train_data <- mydata[no_index, ]  # no 訓練集
no3_test_data <- mydata[-no_index, ]  # no 測試集

train_data <- rbind(yes3_train_data, no3_train_data)
test_data <- rbind(yes3_test_data, no3_test_data)
```

```r
variables <- names(train_data)
testHomo <- lapply(variables, function(var){
  table_test_train <- data.frame(table(train_data[[var]]),table(test_data[[var]]))
  table_test_train[,c(1,3)]<-NULL
  chi_test <- chisq.test(table_test_train, correct = FALSE)
  return(c(chi_test$statistic, chi_test$p.value))
})
testHomo <- as.data.frame(do.call(rbind, testHomo))
colnames(testHomo) <- c("X.squared","P.value")
testHomo$var <- unlist(variables)
testHomo <- testHomo[,c(3,1,2)]
testHomo$P.value <- round(testHomo$P.value, digits = 5)
testHomo$X.squared <- round(testHomo$X.squared, digits = 5)
while(sum(testHomo[,3]>=0.1)!=8){
  i <- i+1
  set.seed(i)
  yes_index <- createDataPartition(yes3$support, p = 0.8, list = FALSE)
  no_index <- createDataPartition(no3$support, p = 0.8, list = FALSE)
  yes3_train_data <- mydata[yes_index, ]  # yes 訓練集
  yes3_test_data <- mydata[-yes_index, ]  # yes 測試集
  no3_train_data <- mydata[no_index, ]  # no 訓練集
  no3_test_data <- mydata[-no_index, ]  # no 測試集

  train_data <- rbind(yes3_train_data, no3_train_data)
  test_data <- rbind(yes3_test_data, no3_test_data)
  variables <- names(train_data)
  testHomo <- lapply(variables, function(var){
    table_test_train <- data.frame(table(train_data[[var]]),table(test_data[[var]]))
    table_test_train[,c(1,3)]<-NULL
    chi_test <- chisq.test(table_test_train, correct = FALSE)
    return(c(chi_test$statistic, chi_test$p.value))
  })
  testHomo <- as.data.frame(do.call(rbind, testHomo))
  colnames(testHomo) <- c("X.squared","P.value")
  testHomo$var <- unlist(variables)
  testHomo <- testHomo[,c(3,1,2)]
  testHomo$P.value <- round(testHomo$P.value, digits = 5)
  testHomo$X.squared <- round(testHomo$X.squared, digits = 5)
}
cat(" 適合的種子:",i)
```

適合的種子: 6

```r
latex(testHomo, file = "", caption=" 同質性檢定結果")
```

```
\begin{table}[!tbp]
\caption{同質性檢定結果\label{testHomo}}
\begin{center}
\begin{tabular}{llrr}
\hline\hline
\multicolumn{1}{l}{testHomo}&\multicolumn{1}{c}{var}&\multicolumn{1}{c}{X.squared}&\multicolumn{1}{c}{P.
\hline
1&v1&$ 2.47861$&$0.11540$\tabularnewline
2&v2&$27.35031$&$0.74416$\tabularnewline
3&v3&$21.21025$&$0.38485$\tabularnewline
```

```
4&v6&$ 3.12099$&$0.53779$\tabularnewline
5&v7&$ 4.73928$&$0.44852$\tabularnewline
6&v8&$ 0.74351$&$0.38854$\tabularnewline
7&has_3&$ 0.32947$&$0.56597$\tabularnewline
8&support&$ 0.00695$&$0.93355$\tabularnewline
\hline
\end{tabular}\end{center}
\end{table}
```

隨機森林

```
Loading required package: foreach

Loading required package: iterators

Loading required package: parallel

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1708  245
         1    0    0

               Accuracy : 0.8746
                 95% CI : (0.859, 0.8889)
    No Information Rate : 0.8746
    P-Value [Acc > NIR] : 0.517

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.8746
         Neg Pred Value :    NaN
             Prevalence : 0.8746
         Detection Rate : 0.8746
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 0
```

```r
library(randomForest)
set.seed(123)
F1_score <- c()
temp <- data.frame(
  mtry = c(),
  ntree = c(),
  x = c()
)
for(mtry in 1:7){
  for(ntree in c(200,300,500,750,1000,1500)){
    for(x in c(4,7.9,10,12)){
```

```
      fit.rf <- randomForest(
        support ~., data = train_data,
        weights = ifelse(train_data$support=="1", x,1),
        importance = TRUE, ntree = ntree, mtry = mtry
      )
      cm <- confusionMatrix(predict(fit.rf, test_data),reference = test_data$support)
      f1 <- (2 * cm[[4]]['Sensitivity'] * cm[[4]]['Precision']) /
              (cm[[4]]['Sensitivity'] + cm[[4]]['Precision'])
      F1_score <- c(F1_score,f1)
      temp <- rbind(temp, c(mtry,ntree,x))
    }
  }
}
set.seed(123)
fit.rf <- randomForest(
        support ~., data = train_data,
        weights = ifelse(train_data$support=="1", temp[which.max(F1_score),3],1),
        importance = TRUE, ntree = temp[which.max(F1_score),2], mtry = temp[which.max(F1_score),1]
      )
confusionMatrix(predict(fit.rf, test_data),reference = test_data$support)
```

```
      Reference
```

Prediction 0 1 0 1535 133 1 173 112

```
           Accuracy : 0.8433
             95% CI : (0.8264, 0.8592)
No Information Rate : 0.8746
P-Value [Acc > NIR] : 0.99998

              Kappa : 0.3326
```

Mcnemar's Test P-Value : 0.02578

```
        Sensitivity : 0.8987
        Specificity : 0.4571
     Pos Pred Value : 0.9203
     Neg Pred Value : 0.3930
         Prevalence : 0.8746
     Detection Rate : 0.7860
```

Detection Prevalence : 0.8541
Balanced Accuracy : 0.6779

```
   'Positive' Class : 0
```

廣義線性模型 (羅吉斯)

```
fit <- glm(support~v1+v6+v7+v8,family = binomial(logit),data = train_data)
fit.aic<-stepAIC(fit,trace = FALSE)
summary(fit.aic)
```

```
Call:
glm(formula = support ~ v1 + v7, family = binomial(logit), data = train_data)
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -16.1953   558.7996  -0.029 0.976879
v12          -0.7492     0.1997  -3.752 0.000176 ***
v71          14.1692   558.7996   0.025 0.979771
v72          14.9838   558.7996   0.027 0.978608
v73          14.5092   558.7996   0.026 0.979285
v74          14.6933   558.7996   0.026 0.979022
v75          14.4020   558.7996   0.026 0.979438
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 990.64  on 1304  degrees of freedom
Residual deviance: 963.32  on 1298  degrees of freedom
AIC: 977.32

Number of Fisher Scoring iterations: 15
```

```
predictions<-predict(fit.aic,newdata = test_data,type='response')
library(pROC)
```

```
Type 'citation("pROC")' for a citation.


Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```

```
roc <-roc(test_data$support,predictions)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```
optimal_cut_point_logit <- coords(roc, "best", ret = "threshold") # 找出 cut point
optimal_cut_point_logit
```

```
  threshold
1 0.1494759
```

```
pre_logit<-ifelse(predictions>0.1553628,1,0) # 就能分為 support or do not support
confusionMatrix(as.factor(pre_logit), as.factor(test_data$support))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1183  124
         1  525  121

               Accuracy : 0.6677
                 95% CI : (0.6463, 0.6886)
    No Information Rate : 0.8746
    P-Value [Acc > NIR] : 1
```

```
                 Kappa : 0.1096

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6926
            Specificity : 0.4939
         Pos Pred Value : 0.9051
         Neg Pred Value : 0.1873
             Prevalence : 0.8746
         Detection Rate : 0.6057
   Detection Prevalence : 0.6692
      Balanced Accuracy : 0.5933

       'Positive' Class : 0
```

我們先使用 stepAIC 選取 AIC 最小的模型。而從模型當中去確認相對顯著的變數，可以得出中西區的勝算是北區的 exp(-0.8391)=0.432 倍。