# HW4

高嘉妤、柯堯琙、吳承恩、趙友誠

2024-10-30

## Table of contents

**0.** 資料簡介 **1**

**1.** 資料整理、資料清洗、**missing values** 診斷、資料視覺化 **2**
遺失值比例圖 . . . . . 4

**2.** 分析所有候選人的支持率 **4**

**3. 3** 號候選人的競選策略 **(**需在何地、對何人進行拉票**)** **6**
地圖視覺化 . . . . . 6
借助統計模型分析顯著因子 . . . . . 7

**4.** 以 **V4** 回答出候選人人數來評估受訪者「政治熱衷程度」，建立合適統計模型分析該變數並說明使用該方法的原因 **8**

**5. 3** 號候選人支持率 **(**具資料不平衡特性**)** 的預測模式與資料不平衡的處理 **12**

```
library(rio)          #read sav file
library(labelled)     #remove attribute of sav data
library(Hmisc)        #describe
library(sf)           #render map
#remotes::install_github("shihjyun/twmap")
library(twmap)        #map data
library(showtext)     #show zw-tw in ggplot2
library(dplyr);library(ggplot2);library(MASS)
pollsav <- import("poll.sav")
```

# 0. 資料簡介

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 | v2: 33 個里, v3: 20 個里 |
| 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 |

| Variables | Explanation | remark |
|---|---|---|
| V8 | Sex | 1:male, 2:female |

# 1. 資料整理、資料清洗、missing values 診斷、資料視覺化

```
pollcsv <- data.frame(
  apply(pollsav,2,
        function(col){
          as.factor(remove_attributes(col,
                                      attributes = c("label","format.spss","display_width","labels")))
})) #sav 格式的" 屬性" 會造成 describe 的 bug，因此將標籤移除
pollcsv <- remove_attributes(pollcsv, "dimnames")
n <- dim(pollcsv)[1]
latex(describe(pollcsv), file="")
```

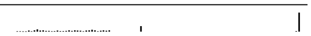
**pollcsv**
**15 Variables    1671 Observations**

**v1**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 2 |

```
Value          1     2
Frequency   1107   564
Proportion 0.662 0.338
```

**v2**



| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 36 |

```
lowest : 1  10 11 12 13, highest: 7  8  9  98 99
```

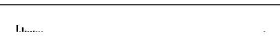
**v3**



| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 23 |

```
lowest : 1  10 11 12 13, highest: 7  8  9  98 99
```

**v4_1**



| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 12 |

```
Value          1    10     2     3     4     5     6     7     8     9    91    98
Frequency    328    11     5   214    43    27    38    47     4     1    14   939
Proportion 0.196 0.007 0.003 0.128 0.026 0.016 0.023 0.028 0.002 0.001 0.008 0.562
```

**v4_2**



| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 10 |

```
Value         10     2     3     4     5     6     7     8     9    99
Frequency     15     6   189    59    32    75    99     2     4  1190
Proportion 0.009 0.004 0.113 0.035 0.019 0.045 0.059 0.001 0.002 0.712
```

**v4_3**



| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 9 |

```
Value         10     3     4     5     6     7     8     9    99
Frequency     19     6    60    36    61    91     1     2  1395
Proportion 0.011 0.004 0.036 0.022 0.037 0.054 0.001 0.001 0.835
```

**v4_4**



| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 8 |

```
Value         10     4     5     6     7     8     9    99
Frequency     20     4    28    41    52     3     4  1519
Proportion 0.012 0.002 0.017 0.025 0.031 0.002 0.002 0.909
```

**v4_5**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 7 |

| Value | 10 | 5 | 6 | 7 | 8 | 9 | 99 |
|---|---|---|---|---|---|---|---|
| Frequency | 15 | 3 | 14 | 38 | 4 | 3 | 1594 |
| Proportion | 0.009 | 0.002 | 0.008 | 0.023 | 0.002 | 0.002 | 0.954 |

**v4_6**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 6 |

| Value | 10 | 6 | 7 | 8 | 9 | 99 |
|---|---|---|---|---|---|---|
| Frequency | 20 | 3 | 12 | 6 | 7 | 1623 |
| Proportion | 0.012 | 0.002 | 0.007 | 0.004 | 0.004 | 0.971 |

**v4_7**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 5 |

| Value | 10 | 7 | 8 | 9 | 99 |
|---|---|---|---|---|---|
| Frequency | 12 | 3 | 2 | 3 | 1651 |
| Proportion | 0.007 | 0.002 | 0.001 | 0.002 | 0.988 |

**v4_8**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 3 |

| Value | 10 | 8 | 99 |
|---|---|---|---|
| Frequency | 4 | 1 | 1666 |
| Proportion | 0.002 | 0.001 | 0.997 |

**v5**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 13 |

| Value | 1 | 10 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 91 | 98 | 99 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Frequency | 158 | 53 | 9 | 205 | 79 | 33 | 98 | 195 | 6 | 8 | 10 | 269 | 548 |
| Proportion | 0.095 | 0.032 | 0.005 | 0.123 | 0.047 | 0.020 | 0.059 | 0.117 | 0.004 | 0.005 | 0.006 | 0.161 | 0.328 |

**v6**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 6 |

| Value | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Frequency | 52 | 94 | 201 | 336 | 946 | 42 |
| Proportion | 0.031 | 0.056 | 0.120 | 0.201 | 0.566 | 0.025 |

**v7**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 6 |

| Value | 1 | 2 | 3 | 4 | 5 | 95 |
|---|---|---|---|---|---|---|
| Frequency | 292 | 165 | 431 | 198 | 520 | 65 |
| Proportion | 0.175 | 0.099 | 0.258 | 0.118 | 0.311 | 0.039 |

**v8**

| n | missing | distinct |
|---|---|---|
| 1671 | 0 | 2 |

| Value | 1 | 2 |
|---|---|---|
| Frequency | 682 | 989 |
| Proportion | 0.408 | 0.592 |

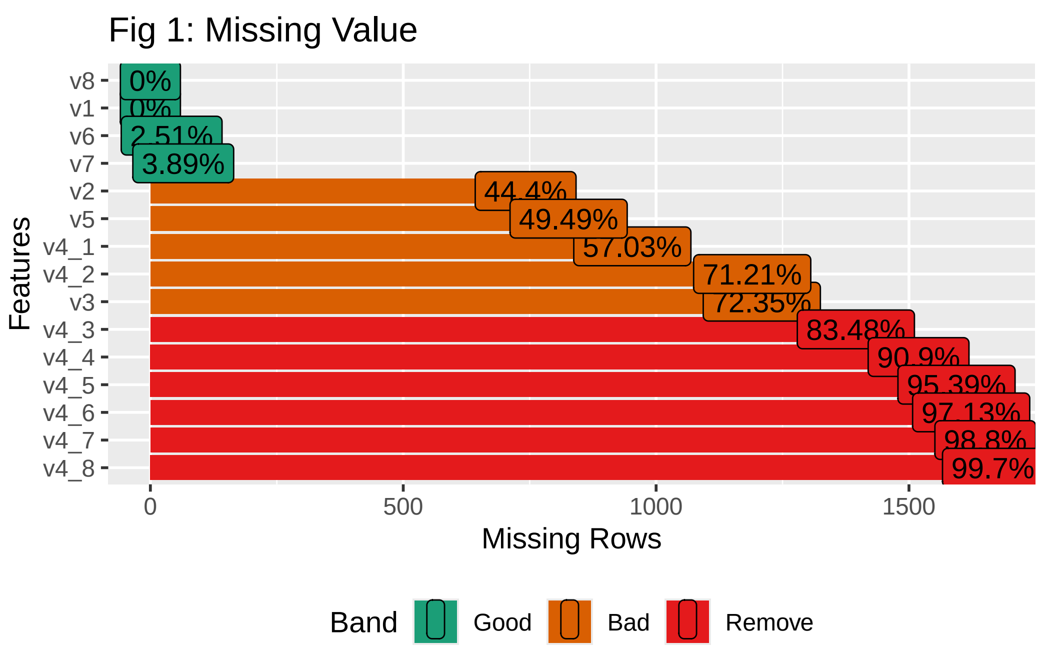
Table 2: 遺失值定義

| Variables | Missing |
|---|---|
| V1 | 98,99 |
| V2、V3 | 44,98,99 |
| V4_1~V4_8 | 91,98,99 |
| V5 | 91,98,99 |
| V6 | 6,99 |
| V7 | 95,99 |
| V8 | 99 |

遺失值比例圖

將定義的遺失值轉換成 NA 並以遺失值比例圖 (by variable) 的方式呈現。考量到遺失值的性質，我們並未刪除任何資料，決定後續對不同變數分析時再移除。

```
pollcsv <- data.frame(
  t(apply(pollcsv,MARGIN = 1, FUN = function(row){
    row[row==99 | row==98 | row==95 | row==91 | row==44] <- NA
    return(row)
  }))
)
pollcsv$v6[pollcsv$v6==6] <- NA
DataExplorer::plot_missing(pollcsv, title = "Fig 1: Missing Value",)
```




## 2. 分析所有候選人的支持率

支持度定義: 支持度 $= \frac{\text{第五題出現次數}}{\text{樣本數}}$

```
# 計算總體支持度
count5.total <- sapply(1:11,function(x){
  if(x==11) return(sum(is.na(pollcsv$v5))/n)
  else return(sum(pollcsv$v5[!is.na(pollcsv$v5)]==x)/n)
} )
# 計算分區支持度（北區中西區）v1
support.district <- do.call(rbind, lapply(1:2,function(i){
  tempdata <- pollcsv[pollcsv$v1==i,]
  n.temp <- dim(tempdata)[1]
  return(sapply(1:11, function(x){
    if(x==11) return(sum(is.na(tempdata$v5))/n.temp)
    else return(sum(tempdata$v5[!is.na(tempdata$v5)]==x)/n.temp)
    }))
}))
# 計算性別支持度 v8
support.sex <- do.call(rbind, lapply(1:2,function(i){
```

```
  tempdata <- pollcsv[pollcsv$v8==i,]
  n.temp <- dim(tempdata)[1]
  return(sapply(1:11, function(x){
    if(x==11) return(sum(is.na(tempdata$v5))/n.temp)
    else return(sum(tempdata$v5[!is.na(tempdata$v5)]==x)/n.temp)
    }))
}))
# 計算年齡支持度 v6
support.age <- do.call(rbind, lapply(1:5,function(i){
  tempdata <- pollcsv[pollcsv$v6==i,]
  n.temp <- dim(tempdata)[1]
  return(sapply(1:11, function(x){
    if(x==11) return(sum(is.na(tempdata$v5))/n.temp)
    else return(sum(tempdata$v5[!is.na(tempdata$v5)]==x)/n.temp)
    }))
}))
# 計算教育程度支持度 v7
support.edu <- do.call(rbind, lapply(1:5,function(i){
  tempdata <- pollcsv[pollcsv$v7==i,]
  n.temp <- dim(tempdata)[1]
  return(sapply(1:11, function(x){
    if(x==11) return(sum(is.na(tempdata$v5))/n.temp)
    else return(sum(tempdata$v5[!is.na(tempdata$v5)]==x)/n.temp)
    }))
}))
table.support <- rbind(
  count5.total,
  support.district,
  support.sex,
  support.age,
  support.edu
)
table.support <- data.frame(
  apply(table.support, 2, function(col) paste0(round(col,3)*100,"%"))
)
rownames(table.support) <- c(
  "",
  "北區","中西區",
  "男性","女性",
  "20 到 29 歲","30 到 39 歲","40 到 49 歲","50 到 59 歲","60 歲以上",
  "小學","國中","高中","專科","大學以上")
colnames(table.support) <- c(1:10,"沒決定")
latex(table.support, file = "",title="",
      rgroup = c("總計","分區","性別","年齡","學歷"),
      n.rgroup = c(1,2,2,5,5),
      caption = "候選人支持度整理表"
)
```

Table 3: 候選人支持度整理表

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 沒決定 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 總計 | | | | | | | | | | | |
| | 9.5% | 0.5% | 12.3% | 4.7% | 2% | 5.9% | 11.7% | 0.4% | 0.5% | 3.2% | 49.5% |
| 分區 | | | | | | | | | | | |
| 北區 | 5.1% | 0.6% | 14.7% | 2.9% | 2.6% | 7.5% | 12.9% | 0.3% | 0.4% | 2.7% | 50.3% |
| 中西區 | 18.1% | 0.4% | 7.4% | 8.3% | 0.7% | 2.7% | 9.2% | 0.5% | 0.7% | 4.1% | 47.9% |
| 性別 | | | | | | | | | | | |
| 男性 | 9.8% | 0.9% | 12.9% | 5.6% | 2.5% | 7.3% | 11.6% | 0.7% | 0.3% | 4% | 44.4% |
| 女性 | 9.2% | 0.3% | 11.8% | 4.1% | 1.6% | 4.9% | 11.7% | 0.1% | 0.6% | 2.6% | 53% |
| 年齡 | | | | | | | | | | | |
| 20 到 29 歲 | 3.2% | 1.1% | 5.3% | 3.2% | 0% | 1.1% | 11.7% | 1.1% | 0% | 1.1% | 72.3% |
| 30 到 39 歲 | 5.9% | 1.5% | 8.8% | 1.5% | 2.2% | 4.4% | 11.8% | 1.5% | 0.7% | 2.9% | 58.8% |
| 40 到 49 歲 | 4.5% | 1.2% | 12.8% | 4.5% | 3.3% | 5.3% | 16% | 0% | 0.8% | 1.2% | 50.2% |
| 50 到 59 歲 | 10.6% | 0.8% | 13.8% | 5% | 2.6% | 5.8% | 11.4% | 0.3% | 0.5% | 1.9% | 47.4% |
| 60 歲以上 | 9.6% | 0% | 10.6% | 4.5% | 1.2% | 5.7% | 8.6% | 0.2% | 0.3% | 3.8% | 55.5% |
| 學歷 | | | | | | | | | | | |
| 小學 | 8.7% | 0% | 7.6% | 1.4% | 0.6% | 3.4% | 5% | 0.3% | 0% | 1.1% | 72% |
| 國中 | 7.8% | 0% | 11.3% | 2.6% | 1.3% | 2.2% | 7.4% | 0% | 0% | 3% | 64.3% |
| 高中 | 9.1% | 0% | 12.9% | 5% | 2.6% | 6.5% | 9.5% | 0.4% | 0.8% | 3.2% | 50% |
| 專科 | 7.2% | 0.4% | 11.8% | 3.8% | 2.3% | 6.1% | 7.6% | 0% | 0% | 2.3% | 58.6% |
| 大學以上 | 7.2% | 1.4% | 9.7% | 5.3% | 1.5% | 5.6% | 15.4% | 0.5% | 0.7% | 3.4% | 49.2% |

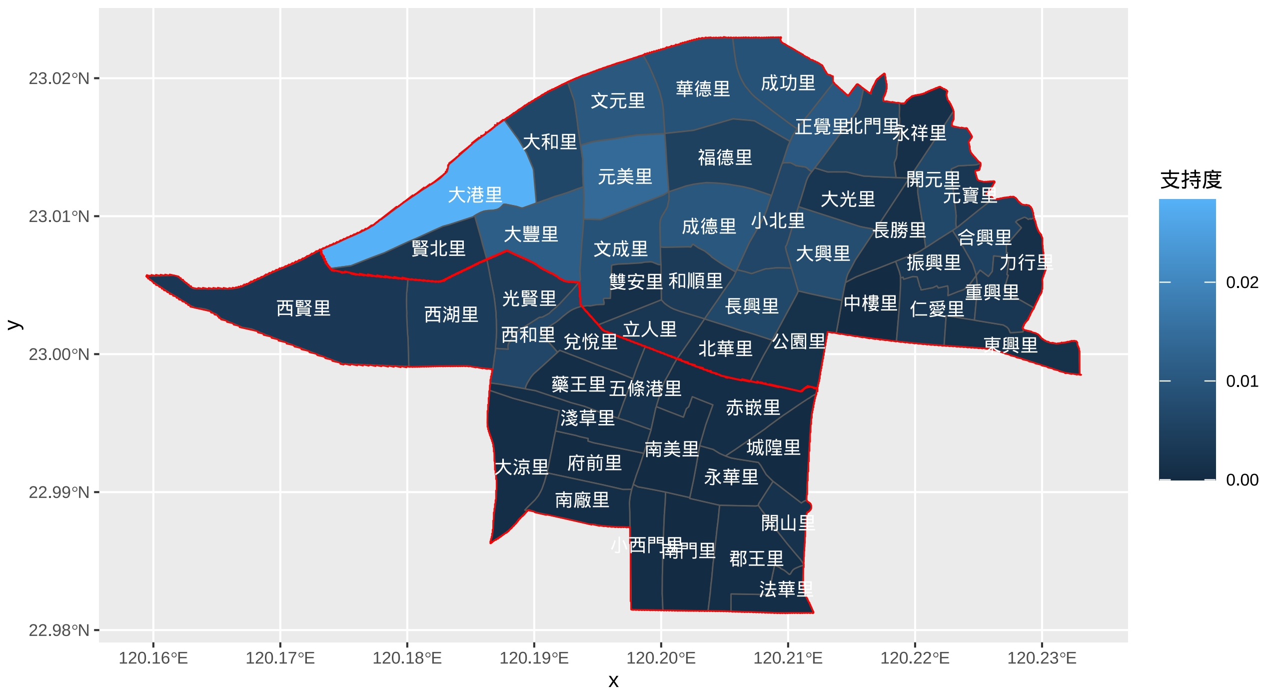
## 3. 3 號候選人的競選策略 (需在何地、對何人進行拉票)

### 地圖視覺化

```
# 計算三號候選人對於里的支持度
support.li_north <- data.frame(
  support = sapply(1:33, function(i){
    tempdata <- pollcsv[pollcsv$v2==i,]
    n.temp <- dim(tempdata)[1]
    return(sum(tempdata$v5[!is.na(tempdata$v5)]==3)/n.temp)}
  ),
  VILLNAME = names(attr(pollsav$v2,"labels"))[1:33]
)
support.li_midwest <- data.frame(
  support = sapply(1:20, function(i){
    tempdata <- pollcsv[pollcsv$v3==i,]
    n.temp <- dim(tempdata)[1]
    return(sum(tempdata$v5[!is.na(tempdata$v5)]==3)/n.temp)
  }),
  VILLNAME = names(attr(pollsav$v3,"labels"))[1:20]
)
# 從台灣地圖選取中西區與北區里層級的地圖資料
myMap <- tw_village[
  tw_village$COUNTYNAME == " 臺南市" &
  (tw_village$TOWNNAME==" 中西區"| tw_village$TOWNNAME==" 北區") ,]
myMap <- merge(x = myMap, y = rbind(support.li_midwest, support.li_north), by = "VILLNAME")
showtext_auto()
ggplot(data = myMap) +
  geom_sf(aes(fill = support)) +  # 填充區域
```

```r
  geom_sf(
    data = summarize(
      group_by(myMap,TOWNNAME),
      geometry = st_union(st_buffer(geometry,dist = 0.01))) , fill = NA, color = 'red') +
      #st_buffer 是為了解決 union 之後內部還有線條的問題 (地圖資料有問題)
  geom_sf_text(aes(label=VILLNAME), size = 2, color = "white")+
  ggtitle("Fig 2: 三號候選人支持度熱區圖")+
  labs(fill = " 支持度")+
  theme_gray(base_size = 6.5)
```

Fig 2: 三號候選人支持度熱區圖




借助統計模型分析顯著因子

```r
pollcsv$sup3 <- ifelse(pollcsv$v5=="3",1,0)
pollcsv$sup3[is.na(pollcsv$sup3)] <- 0
data.adjust <- na.omit(pollcsv[,c(1,13:16)])  # 將 v6~v8 有 NA 的資料刪除
data.adjust$sup3 <- as.factor(data.adjust$sup3)
count_sup3 <- table(data.adjust$sup3)
w <- ifelse(data.adjust$sup3 == 1,
                  count_sup3[1] / (count_sup3[1]+count_sup3[2]),
                  count_sup3[2] / (count_sup3[1]+count_sup3[2]))
w <- round(round(100*w)/min(round(100*w)))
wlogit_fulldata <- glm(sup3 ~ v1 + v6 + v7 + v8,
                       data = data.adjust,
                       family = binomial(),
                       weights = w)
summary(wlogit_fulldata)
```

```
Call:
glm(formula = sup3 ~ v1 + v6 + v7 + v8, family = binomial(),
```

```
    data = data.adjust, weights = w)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.04888    0.27404  -0.178 0.858445
v12         -0.72456    0.08754  -8.277  < 2e-16 ***
v62          0.25664    0.27921   0.919 0.358007
v63          0.42026    0.25221   1.666 0.095658 .
v64          0.31509    0.24559   1.283 0.199488
v65         -0.04643    0.24378  -0.190 0.848954
v72          0.58620    0.15223   3.851 0.000118 ***
v73          0.35934    0.12513   2.872 0.004082 **
v74          0.34832    0.15520   2.244 0.024810 *
v75         -0.08018    0.14002  -0.573 0.566864
v82         -0.06042    0.08019  -0.753 0.451182
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3924.1  on 1600  degrees of freedom
Residual deviance: 3792.2  on 1590  degrees of freedom
AIC: 3814.2

Number of Fisher Scoring iterations: 5
```

模型中，v1(區) 以北區 (1) 最為 baseline，此項的係數為顯著且小於 0。中西區 vs. 北區的 odds ratio 是

$$e^{-0.72456} \approx 0.485$$

v7(教育程度) 的 2,3,4(國中～專科) 較為顯著且係數為正，各自對於只有國小學歷的人的 odds ratio 為

$$e^{0.58620} \approx 1.797$$

$$e^{0.35934} \approx 1.432$$

$$e^{0.34832} \approx 1.417$$

v6(年齡) 只有 3(40~49 歲) 稍微顯著，對於 20~29 歲的人的 odds ratio 是

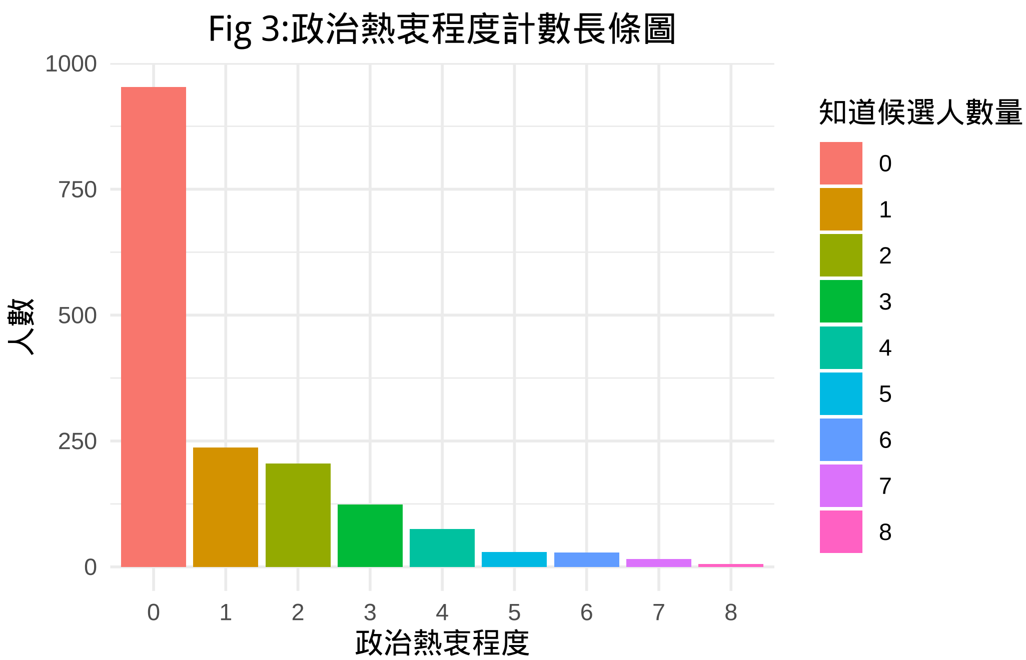
$$e^{0.42026} \approx 1.522$$

結合圖 2 的地圖資訊，建議三號候選人積極對北區、學歷在國中到專科之間、年齡在 40~49 歲的民眾積極拓展知名度。

## 4. 以 V4 回答出候選人人數來評估受訪者「政治熱衷程度」，建立合適統計模型分析該變數並說明使用該方法的原因

```r
pollcsv$known_count <- rowSums(!is.na(pollcsv[,c("v4_1", "v4_2", "v4_3", "v4_4", "v4_5", "v4_6", "v4_7",
count_data <- data.frame(
  Times = factor(0:8),
  Values = sapply(0:8,function(x){
    sum(pollcsv$known_count==x)
  })
)
```

```
# 建立次數圖
ggplot(count_data, aes(x = Times, y = Values , fill = Times ))+
  geom_bar(stat = 'identity')+
  scale_x_discrete(breaks = 0:8)+
  labs(title='Fig 3: 政治熱衷程度計數長條圖', x = '政治熱衷程度', y = '人數', fill = " 知道候選人數量")+
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))
```




```
poiv4<-glm(known_count~v1+v6+v7+v8, data = pollcsv, family = poisson())
AER::dispersiontest(poiv4)
```

```
	Overdispersion test

data:  poiv4
z = 12.524, p-value < 2.2e-16
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  2.321796
```

```
nbv4 <- glm.nb(known_count~v1+v6+v7+v8, data = pollcsv)
lmtest::lrtest(poiv4,nbv4) # 決定要用 Poisson 還是 Negative binomial
```

```
Likelihood ratio test

Model 1: known_count ~ v1 + v6 + v7 + v8
Model 2: known_count ~ v1 + v6 + v7 + v8
  #Df  LogLik Df  Chisq Pr(>Chisq)
1  11 -2635.0
2  12 -2280.5  1 709.13  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(nbv4)
```

```
Call:
glm.nb(formula = known_count ~ v1 + v6 + v7 + v8, data = pollcsv,
    init.theta = 0.6258750942, link = log)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.07825    0.30518  -3.533 0.000411 ***
v12         -0.14018    0.08591  -1.632 0.102753
v62          0.77007    0.31503   2.444 0.014506 *
v63          0.89173    0.29142   3.060 0.002213 **
v64          1.01004    0.28303   3.569 0.000359 ***
v65          0.93793    0.27922   3.359 0.000782 ***
v72          0.25893    0.16467   1.572 0.115855
v73          0.50668    0.13195   3.840 0.000123 ***
v74          0.51789    0.15992   3.239 0.001201 **
v75          0.50471    0.14010   3.603 0.000315 ***
v82         -0.19371    0.08241  -2.350 0.018749 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.6259) family taken to be 1)

    Null deviance: 1531.0  on 1600  degrees of freedom
Residual deviance: 1485.5  on 1590  degrees of freedom
  (因為不存在，70 個觀察量被刪除了)
AIC: 4584.9

Number of Fisher Scoring iterations: 1

              Theta:  0.6259
          Std. Err.:  0.0473

 2 x log-likelihood:  -4560.9200
```

```r
library(pscl)     # 建立 zero-inflated negative model
zinb_model <- zeroinfl(known_count ~ v1 +v6+v7+v8, data = pollcsv, dist = "negbin")
lmtest::lrtest(nbv4,zinb_model) # 決定要用 Negative 或 zero-inflated
```

```
Likelihood ratio test

Model 1: known_count ~ v1 + v6 + v7 + v8
Model 2: known_count ~ v1 + v6 + v7 + v8
  #Df  LogLik Df  Chisq Pr(>Chisq)
1  12 -2280.5
2  23 -2238.9 11 83.162  3.599e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(zinb_model)
```

```
Call:
```

```
zeroinfl(formula = known_count ~ v1 + v6 + v7 + v8, data = pollcsv, dist = "negbin")

Pearson residuals:
    Min      1Q  Median      3Q     Max 
-0.8393 -0.7052 -0.5357  0.4613  4.6745 

Count model coefficients (negbin with log link):
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -0.587285   0.399595  -1.470  0.14164    
v12         -0.094520   0.079076  -1.195  0.23196    
v62          1.043282   0.398262   2.620  0.00880 ** 
v63          1.081087   0.384695   2.810  0.00495 ** 
v64          1.185464   0.382191   3.102  0.00192 ** 
v65          1.240723   0.383341   3.237  0.00121 ** 
v72         -0.008993   0.158309  -0.057  0.95470    
v73          0.162114   0.124789   1.299  0.19391    
v74          0.201598   0.146656   1.375  0.16925    
v75          0.254145   0.131547   1.932  0.05336 .  
v82         -0.058140   0.072576  -0.801  0.42308    
Log(theta)   1.476716   0.266200   5.547  2.9e-08 ***

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)   
(Intercept)  -0.7381     1.2734  -0.580  0.56219   
v12           0.1319     0.1521   0.867  0.38574   
v62           0.7652     1.2615   0.607  0.54415   
v63           0.5669     1.2510   0.453  0.65042   
v64           0.5615     1.2538   0.448  0.65428   
v65           0.8606     1.2597   0.683  0.49447   
v72          -0.4869     0.2819  -1.727  0.08408 . 
v73          -0.6751     0.2189  -3.084  0.00204 **
v74          -0.6209     0.2714  -2.288  0.02215 * 
v75          -0.4509     0.2308  -1.954  0.05070 . 
v82           0.2961     0.1477   2.005  0.04496 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Theta = 4.3785 
Number of iterations in BFGS optimization: 69 
Log-likelihood: -2239 on 23 Df
```

先將每位受訪者知道的候選人用計數的方式去呈現出政治熱忠程度，而這些資料就轉變成 count data，也因此先使用 Poisson model 去做模型。然而在使用 Poisson model 之後並且去做 Dispersion test 時，可以發現這個模型有 Overdispersion 的情形產生，並且 Likelihood ratio test 的結果也建議我們使用 Negative binomial 的模型。在做出資料的分布圖後，可以發現受訪者完全不知道候選人的比例偏高，也就是 0 的資料，也因此想要使用 Zero-inflated negative binomial model 去解決 0 所帶來的問題。由 ZINB 的報表可以得知，在 count model 底下，也就是有講出候選人的受訪者中，30~39 歲，40~49 歲，50~59 歲以及 60 歲以上，他們相較於 20~29 歲是顯著的，並且他們的係數是逐步提高的，因此我們可以認為隨著年齡提高，政治熱忠程度也會隨之提高。而在零膨脹模型，教育程度的變數當中，高中及專科相較於國小是顯著的，也代表著高中及專科的受訪者更可能出現非零值，也就是說他們相較於教育程度只有國小的受訪者是更可能回答出候選人的。而在性別的部分，可以發現女性相較於男性是顯著的，藉由係數我們可以解釋成女性相較於男性較可能回答不出候選人，也就是說女性提高了結構性零的機率。

## 5. 3 號候選人支持率 (具資料不平衡特性) 的預測模式與資料不平衡的處理

```
set.seed(123)  # For reproducibility
library(smotefamily)
library(dplyr)
new_poll <- pollcsv[,c(1,13,14,15,16)]
new_poll_wei <- pollcsv[,c(1,13,14,15,16)]
# 由於在 V1 V6 V7 V8 當中，缺失值並不多，因此我選擇刪除缺失值
new_poll <- na.omit(new_poll)
new_poll_wei <- na.omit(new_poll_wei)
new_poll_wei$sup3 <- as.numeric(new_poll_wei$sup3)
# 轉換為數值才能使用 smote
new_poll <- new_poll %>%
  mutate_if(is.character, as.factor) %>%
  mutate_if(is.factor, as.numeric)

poll_balanced <- SMOTE(X = new_poll[, -which(names(new_poll) == "sup3")],
                    target = new_poll$sup3,
                    K = 5)
table(poll_balanced$data$class)
```

```
   0    1
1396 1230
```

```
poll_balanced <- poll_balanced$data
# 由於 smote 過後他新增的資料可能會有不是整數的狀況，然而在這筆資料當中應該要為整數，也因此我選擇將那些有小數點
poll_balanced[] <- lapply(poll_balanced, function(x) if(is.numeric(x)) round(x) else x)
poll_balanced$class <- as.numeric(poll_balanced$class)
# 轉換為 factor
poll_balanced[c("v1", "v6", "v7", "v8")] <- lapply(poll_balanced[c("v1", "v6", "v7", "v8")], as.factor)
describe(poll_balanced)
```

```
poll_balanced

 5  Variables      2626  Observations
--------------------------------------------------------------------------------
v1
       n  missing distinct
    2626        0        2

Value          1     2
Frequency   1897   729
Proportion 0.722 0.278
--------------------------------------------------------------------------------
v6
       n  missing distinct
    2626        0        5

Value          1     2     3     4     5
Frequency     72   144   350   593  1467
Proportion 0.027 0.055 0.133 0.226 0.559
--------------------------------------------------------------------------------
v7
       n  missing distinct
```

```
    2626        0        5

Value          1     2     3     4     5
Frequency    422   295   752   349   808
Proportion 0.161 0.112 0.286 0.133 0.308
--------------------------------------------------------------------------------
v8
       n  missing distinct
    2626        0        2

Value          1     2
Frequency   1093  1533
Proportion 0.416 0.584
--------------------------------------------------------------------------------
class
       n  missing distinct     Info      Sum     Mean      Gmd
    2626        0        2    0.747     1230   0.4684   0.4982

--------------------------------------------------------------------------------
```

```r
train_nrow <- floor(0.7 * nrow(poll_balanced))
train_idx <- sample(seq_len(nrow(poll_balanced)), size=train_nrow)

poll_training <- poll_balanced[train_idx, ]
cat("Training set size:", nrow(poll_training))
```

```
Training set size: 1838
```

```r
poll_testing <- poll_balanced[-train_idx, ]
cat("Test set size:", nrow(poll_testing))
```

```
Test set size: 788
```

```r
sup3_log <- glm(class ~ ., data = poll_training, family = binomial)
summary(sup3_log)
```

```

Call:
glm(formula = class ~ ., family = binomial, data = poll_training)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.44350    0.34793  -1.275 0.202430
v12         -0.78399    0.11064  -7.086 1.38e-12 ***
v62          0.40154    0.35760   1.123 0.261490
v63          0.64459    0.32069   2.010 0.044430 *
v64          0.47261    0.31161   1.517 0.129344
v65          0.15989    0.30980   0.516 0.605791
v72          0.67576    0.19177   3.524 0.000425 ***
v73          0.47732    0.15746   3.031 0.002434 **
v74          0.34460    0.19479   1.769 0.076885 .
v75          0.01447    0.17780   0.081 0.935133
v82         -0.02889    0.10027  -0.288 0.773281
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2543.4  on 1837  degrees of freedom
Residual deviance: 2448.5  on 1827  degrees of freedom
AIC: 2470.5

Number of Fisher Scoring iterations: 4
```

```
pred_prob <- predict(sup3_log, poll_testing, type = "response")

pred_class <- ifelse(pred_prob > 0.5, 1, 0)

library(caret)
confusionMatrix(factor(pred_class), factor(poll_testing$class))
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 245 143
         1 186 214

               Accuracy : 0.5825
                 95% CI : (0.5472, 0.6172)
    No Information Rate : 0.547
    P-Value [Acc > NIR] : 0.02431

                  Kappa : 0.1662

 Mcnemar's Test P-Value : 0.02058

            Sensitivity : 0.5684
            Specificity : 0.5994
         Pos Pred Value : 0.6314
         Neg Pred Value : 0.5350
             Prevalence : 0.5470
         Detection Rate : 0.3109
   Detection Prevalence : 0.4924
      Balanced Accuracy : 0.5839

       'Positive' Class : 0
```

```
#Weighted logistic regression

class_counts <- table(new_poll_wei$sup3)
class_counts
```

```

   0    1
1396  205
```

```
train_nrow_wei <- floor(0.7 * nrow(new_poll_wei))
train_idx_wei <- sample(seq_len(nrow(new_poll_wei)), size=train_nrow_wei)

poll_training_wei <- new_poll_wei[train_idx_wei, ]
```

```r
weights <- ifelse(poll_training_wei$sup3 == 1,
                  class_counts[1] / (class_counts[1]+class_counts[2]),
                  class_counts[2] / (class_counts[1]+class_counts[2]))
weights <- round(round(100*weights)/min(round(100*weights)))
poll_testing_wei <- new_poll_wei[-train_idx_wei, ]

weighted_logit_model <- glm(sup3 ~ v1 + v6 + v7 + v8,
                            data = poll_training_wei,
                            family = binomial,
                            weights = weights)
summary(weighted_logit_model)
```

```
Call:
glm(formula = sup3 ~ v1 + v6 + v7 + v8, family = binomial, data = poll_training_wei,
    weights = weights)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.04798    0.34015  -0.141   0.8878
v12         -0.84581    0.10415  -8.121 4.62e-16 ***
v62          0.43011    0.34612   1.243   0.2140
v63          0.62291    0.31600   1.971   0.0487 *
v64          0.48978    0.30904   1.585   0.1130
v65          0.12221    0.30732   0.398   0.6909
v72          0.30541    0.17948   1.702   0.0888 .
v73          0.28374    0.14586   1.945   0.0517 .
v74          0.07682    0.18535   0.414   0.6785
v75         -0.15759    0.16374  -0.962   0.3358
v82         -0.01816    0.09580  -0.190   0.8497
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2789.4  on 1119  degrees of freedom
Residual deviance: 2690.5  on 1109  degrees of freedom
AIC: 2712.5

Number of Fisher Scoring iterations: 5
```

```r
pred_prob_wei <- predict(weighted_logit_model, poll_testing_wei, type = "response")

library(pROC)
# 使用 roc curve 找出最佳的閥值
roc_curve <- roc(poll_testing_wei$sup3, pred_prob_wei)
best_coords <- coords(roc_curve, "best", best.method = "youden")
pred_class_wei <- ifelse(pred_prob_wei > best_coords$threshold, 1, 0)

confusionMatrix(factor(pred_class_wei), factor(poll_testing_wei$sup3))
```

```
Confusion Matrix and Statistics

          Reference
```

```
Prediction   0   1
         0 323  28
         1 102  28

               Accuracy : 0.7297
                 95% CI : (0.6877, 0.7689)
    No Information Rate : 0.8836
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1652

 Mcnemar's Test P-Value : 1.528e-10

            Sensitivity : 0.7600
            Specificity : 0.5000
         Pos Pred Value : 0.9202
         Neg Pred Value : 0.2154
             Prevalence : 0.8836
         Detection Rate : 0.6715
   Detection Prevalence : 0.7297
      Balanced Accuracy : 0.6300

       'Positive' Class : 0
```

在處理不平衡資料的時候，我選擇使用 smote 以及 weighted logistic regression 來處理。我分別將他們都切成訓練集以及測試集，並且對他們做 confusion matrics 以此來判斷哪個模型較佳。在這過程當中，weighted logistic regression 在最後分類的 accuracy rate 高達 0.7297，而 smote 的 accuracy rate 則僅有 0.5825，因此我認為使用 weighted logistic regression 在這裡是較佳的。此外，weighted 權重的部分我是以比例去決定的，以此減少資料的不平衡。