Exam Solution

Xiang Li(4013115)

2024-01-04

Execise 1

1

```
polls_poland_df = read.csv("polls_poland.csv")
```

2

```
polls_poland_df$Fieldwork.End = as.Date(polls_poland_df$Fieldwork.End,
    format = "%Y-%m-%d")
polls_poland_df = polls_poland_df[(polls_poland_df$Fieldwork.End >=
    "2023-10-01") & (polls_poland_df$Fieldwork.End <= "2023-10-31"),
    c("Polling.Firm", "Fieldwork.End", "Sample.Size", "ZP",
        "KO", "TD", "Lewica", "Konfederacja")]
polls_poland_df
```

```
##                Polling.Firm Fieldwork.End   Sample.Size   ZP   KO   TD Lewica
## 1            United Surveys    2023-10-22          1000 33.9 30.9 19.3    5.6
## 2                      IBSP    2023-10-13          1000 37.4 27.8 13.0   10.3
## 3               PGB Opinium    2023-10-12          1100 33.2 30.6 11.9   11.6
## 4  Instytut Badań Pollster    2023-10-12 Not Available 35.8 31.4 10.7   10.5
## 5                    Kantar    2023-10-12          1702 34.0 28.0 12.0   13.0
## 6                     IBRiS    2023-10-11          1000 37.4 30.1 11.0   10.7
## 7                     IBRiS    2023-10-10          1100 36.0 30.1 11.7   10.9
## 8            Social Changes    2023-10-10          1094 36.0 28.0 12.0    9.0
## 9            United Surveys    2023-10-10          1000 36.7 30.5 10.2   11.1
## 10                    Ipsos    2023-10-10          1000 40.0 31.0  9.0    9.0
## 11         Research Partner    2023-10-09          1084 38.6 29.2  9.1    9.6
## 12 Instytut Badań Pollster    2023-10-07          1022 34.3 30.0 11.8    8.2
## 13                    IBRiS    2023-10-07          1000 37.6 30.4  8.3   12.4
## 14                Estymator    2023-10-06          1060 36.9 30.5  9.4    9.3
## 15              PGB Opinium    2023-10-04          1042 32.9 32.3 11.6   10.1
## 16                    IBRiS    2023-10-04          1000 35.7 30.4 11.1   10.5
## 17                 Opinia24    2023-10-04          1000 38.7 32.0  8.0   12.0
## 18                   Kantar    2023-10-04          1500 36.0 32.0  9.0   11.0
## 19           United Surveys    2023-10-02          1000 33.7 29.7 11.6   10.7
## 20           Social Changes    2023-10-02          1073 39.0 30.0  8.0    9.0
```

```
## 21          United Surveys    2023-10-01          1000 35.3 28.9 13.2    8.9
##    Konfederacja
## 1           6.7
## 2           8.3
## 3           8.5
## 4           8.7
## 5           9.0
## 6           8.9
## 7           9.9
## 8          11.0
## 9           9.1
## 10         10.0
## 11          9.7
## 12         10.5
## 13          8.4
## 14          9.6
## 15          8.9
## 16         10.2
## 17          6.0
## 18          8.0
## 19         10.5
## 20         10.0
## 21          9.8
```

```r
nrow(polls_poland_df)
```

```
## [1] 21
```

21 opinion polls are left now.

3

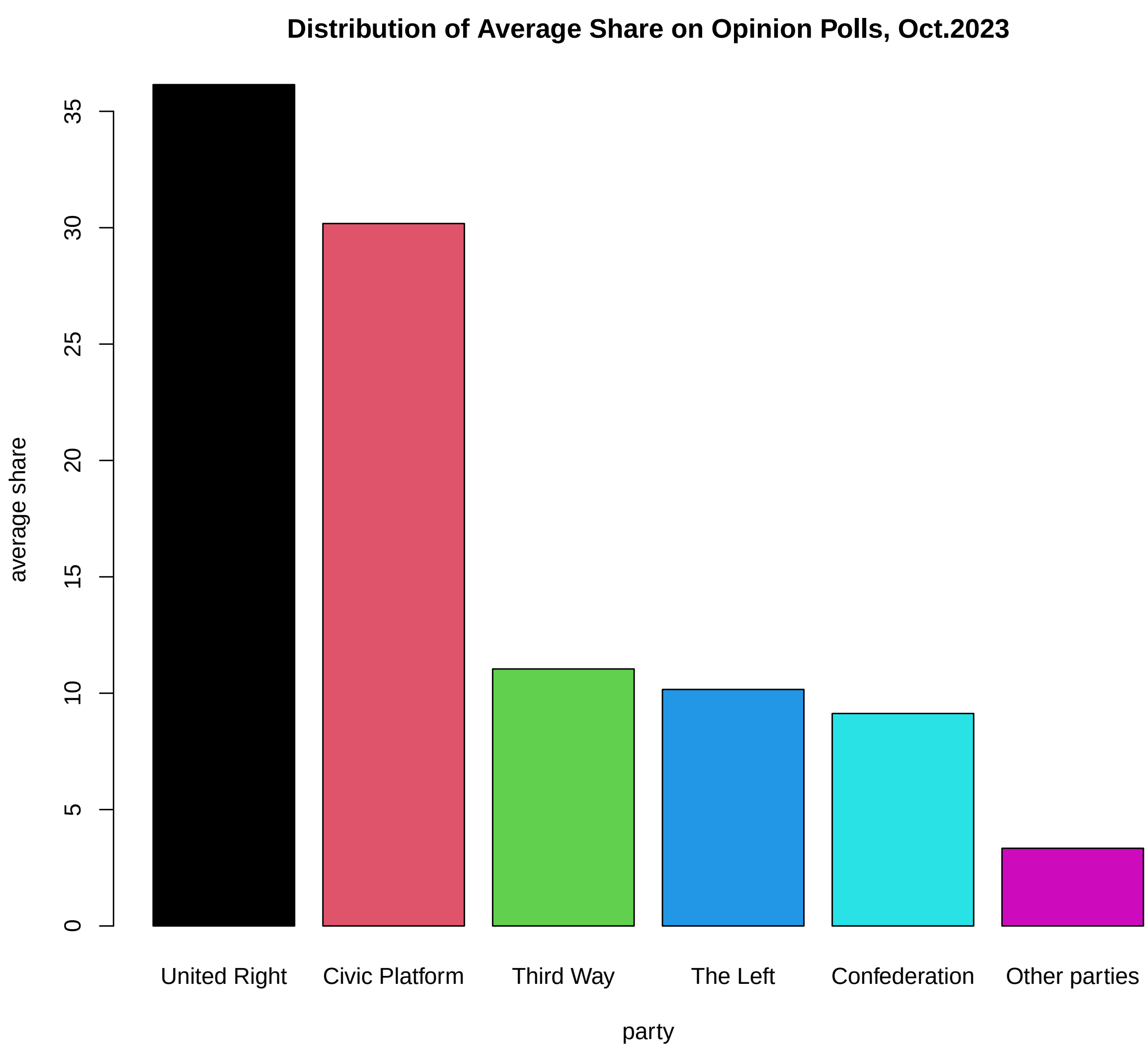
```r
mean_df = as.data.frame(apply(polls_poland_df[, 4:8], 2,
    mean, na.rm = T))
colnames(mean_df) = c("Ave_Share")
mean_df
```

```
##               Ave_Share
## ZP            36.147619
## KO            30.180952
## TD            11.042857
## Lewica        10.161905
## Konfederacja   9.128571
```

4

```r
mean_df["Other parties", "Ave_Share"] = 100 - sum(mean_df$Ave_Share)
mean_df$Name = c("United Right", "Civic Platform", "Third Way",
```

```r
    "The Left", "Confederation", "Other parties")
mean_df = mean_df[order(mean_df$Ave_Share, decreasing = T),
    ]
barplot(height = mean_df$Ave_Share, names.arg = mean_df$Name,
    col = 1:6, main = "Distribution of Average Share on Opinion Polls, Oct.2023",
    xlab = "party", ylab = "average share")
```



5

```r
act_share = c(35.4, 30.7, 14.4, 8.6, 7.2)
mean_df$Act_Share = c(act_share, 100 - sum(act_share))
mean_df$compare = (mean_df$Act_Share > mean_df$Ave_Share) *
    1
mean_df
```

```
##                Ave_Share           Name Act_Share compare
## ZP             36.147619   United Right      35.4       0
## KO             30.180952 Civic Platform      30.7       1
## TD             11.042857      Third Way      14.4       1
## Lewica         10.161905       The Left       8.6       0
## Konfederacja    9.128571   Confederation      7.2       0
## Other parties   3.338095  Other parties       3.7       1
```

Based on the table above, Civic Platform and Third Way obtained more votes than predicted by the opinion polls, United Right, The Left and Confederation obtained less votes than expected.

Execise 2

1

The answer is on the attached exam paper.

2

```r
x_seq = c(10.4, -8.74, 3.58, -1.98)
y = sum(x_seq)
k = length(x_seq)
while (y <= 172000) {
    k = k + 1
    x_seq[k] = (3 * x_seq[k - 1] - 5 * x_seq[k - 3])/4
    y = sum(x_seq)
}
length(x_seq)
```

```
## [1] 67
```

I need to cumulate 67 elements from x_k to make y_k exceed 172000.

3

```r
k
```

```
## [1] 67
```

```r
y
```

```
## [1] 400344.6
```

$k = 67$ and $y_k = 400344.6$.

Execise 3

1

```
f_X = function(x, p, alpha) {
    (-1/log(p)) * (alpha * (1 - p) * exp(-alpha * x))/(1 -
        (1 - p) * exp(-alpha * x))
}
f_X(x = 2, p = 0.7, alpha = 1)
```

```
## [1] 0.118648
```

P(X = 2) is greater than 0.5 is False, because X is a continuous random variable and the probability of that a continuous random variable gets a single value is 0, so $P(X = 2) = 0$.

2

```
log_lik = function(x_v, p, alpha) {
    if (any(x_v < 0)) {
        stop("There is negative value in x!!!")
    }
    if ((p <= 0) | (p >= 1) | (alpha <= 0)) {
        stop("Argument p or alpha is invalid!!!")
    }
    n = length(x_v)
    result = n * log(-alpha * (1 - p)/log(p)) - sum(log(exp(alpha *
        x_v) + p - 1))
    return(result)
}
```

3

```
sample_v = c(0.1, 0.2, 0.4, 1.3, 0.1, 1.2, 1.6, 0.5, 0.4,
    0.1, 0.1, 0.1)
```

```
neg_log_lik = function(para, x_v) {
    p = para[1]
    alpha = para[2]
    return(-1 * log_lik(x_v, p, alpha))
}
mle = optim(c(0.1, 1), neg_log_lik, method = "L-BFGS-B",
    lower = c(1e-05, 1e-05), upper = c(1 - 1e-05, Inf),
    x_v = sample_v)
c(p_hat = pnorm(mle$par[1]), alpha_hat = exp(mle$par[2]),
    loglikelihood = -1 * mle$value)
```

```
##         p_hat     alpha_hat loglikelihood
##     0.6647309     4.9145432    -3.7534383
```

$\hat{p} = 0.6647309$, $\hat{\alpha} = 4.9145432$ and $loglikelihood = -3.7534383$.

4

The answer is on the attached exam paper.

5

```
neg_log_lik1 = function(para, x_v) {
    p = pnorm(para[1])
    alpha = exp(para[2])
    return(-1 * log_lik(x_v, p, alpha))
}
mle1 = optim(c(0.1, 1), neg_log_lik1, x_v = sample_v)
c(p_hat = pnorm(mle1$par[1]), alpha_hat = exp(mle1$par[2]),
    loglikelihood = -1 * mle1$value)
```

```
##         p_hat     alpha_hat loglikelihood
##      0.425438      1.592248     -3.753438
```

$\hat{p} = 0.425438$, $\hat{\alpha} = 1.592248$ and $loglikelihood = -3.753438$.

6

```
list(mle_const = c(p_hat = pnorm(mle$par[1]), alpha_hat = exp(mle$par[2]),
    loglikelihood = -1 * mle$value), mle_repara = c(p_hat = pnorm(mle1$par[1]),
    alpha_hat = exp(mle1$par[2]), loglikelihood = -1 * mle1$value))
```

```
## $mle_const
##         p_hat     alpha_hat loglikelihood
##     0.6647309     4.9145432    -3.7534383
##
## $mle_repara
##         p_hat     alpha_hat loglikelihood
##      0.425438      1.592248     -3.753438
```

I can't say which one is better.

I could say the optimization which has larger loglikelihood value is better than the other one. But based on the results above, the final loglikelihood values from the two methods are almost same, so I can't say which one is better.