政治学方法論Ⅰ：課題４

作成者：宋財泫(ソン・ジェヒョン)　／　作成日：2014-10-27　／提出日：2014-10-28

全問共通

```
# Stata ファイルを読み込むために foreign パッケージ、
library(foreign)
# 作図のために ggplot2、
library(ggplot2)
# データ操作と wide 型→long 型への変換などのために reshape2 と plyr パッケージを読み込む
library(reshape2)
library(plyr)

# ggplot で日本語を表示するためにテーマを設定する
theme_set(theme_gray(base_size=12, base_family="HiraKakuProN-W3"))

# "hr96-09.dta"を読み込み、hw04 の中に書き込む
hw04 <- read.dta("hr96-09.dta")
head(hw04) # 最初の６つのケースを表示
##   year    ku kun party              name age     status nocand   wl rank
## 1 1996 aichi   1   NFP KAWAMURA, TAKASHI  47  incumbent      7  win    1
## 2 1996 aichi   1   LDP     IMAEDA, NORIO  72       moto      7 lose    2
## 3 1996 aichi   1   DPJ     SATO, TAISUKE  53  incumbent      7 lose    3
## 4 1996 aichi   1   JCP   IWANAKA, MIHOKO  43 challenger      7 lose    4
## 5 1996 aichi   1 bunka       ITO, MASAKO  51 challenger      7 lose    5
## 6 1996 aichi   1    NP  YAMADA, HIROSHIB  51 challenger      7 lose    6
##   previous  vote voteshare eligible turnout     exp
## 1        2 66876      40.0   346774   49.22 9828097
## 2        3 42969      25.7   346774   49.22 9311555
## 3        2 33503      20.1   346774   49.22 9231284
## 4        0 22209      13.3   346774   49.22 2177203
## 5        0   616       0.4   346774   49.22      NA
## 6        0   566       0.3   346774   49.22      NA
tail(hw04) # 最後の６つのケースを表示
##      year        ku kun party                name age     status nocand
## 5609 2009 yamanashi   2   msz    NAGASAKI, KOTARO  41  incumbent      4
## 5610 2009 yamanashi   2   LDP    HORIUCHI, MITSUO  79  incumbent      4
## 5611 2009 yamanashi   2     H MIYAMATSU, HIROYUKI  69 challenger      4
## 5612 2009 yamanashi   3   DPJ       GOTO, HITOSHI  52  incumbent      3
## 5613 2009 yamanashi   3   LDP           ONO, JIRO  56  incumbent      3
## 5614 2009 yamanashi   3     H   SAKURADA, DAISUKE  47 challenger      3
##        wl rank previous   vote voteshare eligible turnout      exp
## 5609 lose    2        1  57213      32.1   234746   77.09  7916556
## 5610 lose    3       10  52773      29.6   234746   77.09 11611677
## 5611 lose    4        0   1214       0.7   234746   77.09  1326378
## 5612  win    1        3 112894      62.7   248102   74.70  6795969
## 5613 lose    2        1  63611      35.3   248102   74.70 12876644
## 5614 lose    3        0   3663       2.0   248102   74.70  1953819
```

問題 1

- データセットから 2005 年の分のデータだけを取りだし、2 変数のクロス表を 2 つ作り、そのう ち 1 つをモザイクプロットにしなさい（モザイクプロットについては、ggplot2 を使わなくても よい）

```
# ======
# 問題 1
# ======

# ==============================
# クロス表 1 ：候補者の立場と当落
# ==============================

# 2005 年度だけのデータを指定し、hw04.2005 に subset として書き込む。
# hw04 の内容が変わると hw04.2005 の中身も変わる。
hw04.2005 <- subset(hw04, year==2005)

# status と wl の内容を日本語に変える
hw04.2005$status <- factor(hw04.2005$status, labels=c("新人", "現職", "元職"))
hw04.2005$wl <- factor(hw04.2005$wl, labels=c("落選", "当選", "復活当選"))

# 立候補時の候補者の立場(status)と当選有無(wl)のクロス表
prob1.table1 <- with(hw04.2005, table(wl, status))
print(prob1.table1)
##           status
## wl          新人 現職 元職
##   落選       445  104   23
##   当選        39  249   12
##   復活当選    43   62   12
```

上の出力結果をより見やすくすると

表 1. 候補者の立場と当落の関係

	新人	現職	元職
当選	445	104	23
落選	39	249	12
復活当選	43	62	12

以上のようになる。

次に、主要政党別の当選率、落選率、復活当選率をクロス表で表示する。ここで主要政党とは自民党、民主党、社民党、公明党、共産党を指す。しかし、政党変数は factor 型であるため、上記の政党のみのデータを抽出しても、他の政党も計算されるようになる。したがって、party 変数をもう一度 factor 型へ変換し、５つの政党のみに絞る。

```
# ==============================
# クロス表 2 ：候補者の政党と当落
# ==============================

# hw04.2005 のデータのうち、自民・公明・民主・社民・共産党のみを hw04.2005.5P へ
hw04.2005.5P <- subset(hw04.2005, party=="LDP" | party=="DPJ" | party=="JCP" |
                       party=="CGP" | party=="SDP")
```

```
# party 変数が factor 型であるため、上記の 5 つの政党のみに再調整し、日本語化する。
hw04.2005.5P$party <- factor(hw04.2005.5P$party,
                             labels=c("自民党", "民主党", "共産党",
                                      "公明党", "社民党"),
                             levels=c("LDP", "DPJ", "JCP", "CGP", "SDP"))

# 政党(party)と当選有無(wl)のクロス表
prob1.table2 <- with(hw04.2005.5P, table(wl, party))
print(prob1.table2)
##           party
## wl          自民党 民主党 共産党 公明党 社民党
##   落選          23    178    271      1     33
##   当選         219     52      0      8      1
##   復活当選      48     59      4      0      4
```

上の出力結果をより見やすくすると

表 2. 政党別の当落結果

	自民党	民主党	社民党	公明党	共産党
当選	219	52	1	8	0
落選	23	178	33	1	271
復活当選	48	59	4	0	4

以上のようになる。また、行と列を反転したものが表３である。

表 3. 政党別の当落結果（行列反転）

	当選	落選	復活当選
自民党	219	23	48
民主党	52	178	59
社民党	1	33	4
公明党	8	1	0
共産党	0	271	4

クロス表の行と列によって変わるのは

- 表２では当選者の中での特定の政党の候補者数が見やすく、
- 表３では特定の政党の中で落選者の数が見やすい

という違いがある。つまり、研究者が示したいものを最もよく表現できる行と列を選択すべきであろう。

次は２つのクロス表(prob1.table1 と prob1.table2)に基づいてモザイクプロットを作図する。

```
# ========================================
# モザイクプロットの作成１（prob1.table1）
# ========================================

# 立場ごとに立候補者数の合計を求める
table.status <- table(hw04.2005$status)
```

```r
# 表を行列に変換
prob1.table1 <- as.matrix(prob1.table1[1:3, 1:3])
table.status <- as.matrix(table.status[1:3])

# 各立場が全体に占める割合と、
# 各立場についての当選、復活当選、落選の割合を変数にする
prob1.df <- data.frame(status = levels(hw04.2005$status),
                       status.pct = 100 * table.status / sum(table.status),
                       win = 100 * prob1.table1[2,] / table.status,
                       zombie = 100 * prob1.table1[3,] / table.status,
                       lose = 100 * prob1.table1[1,] / table.status)

# x 軸上のカテゴリの境界値を計算する
prob1.df$xmax <- cumsum(prob1.df$status.pct)
prob1.df$xmin <- prob1.df$xmax - prob1.df$status.pct

# 今後使わない変数 status.pct 列を削除する
prob1.df$status.pct <- NULL

# prob1.df を形を変換し、ggplot で使いやすくする
prob1.dfm <- melt(prob1.df, id=c("status", "xmin", "xmax"))
print(prob1.dfm)
##   status     xmin      xmax variable     value
## 1   新人  0.00000  53.28615      win  7.400380
## 2   現職 53.28615  95.24772      win 60.000000
## 3   元職 95.24772 100.00000      win 25.531915
## 4   新人  0.00000  53.28615   zombie  8.159393
## 5   現職 53.28615  95.24772   zombie 14.939759
## 6   元職 95.24772 100.00000   zombie 25.531915
## 7   新人  0.00000  53.28615     lose 84.440228
## 8   現職 53.28615  95.24772     lose 25.060241
## 9   元職 95.24772 100.00000     lose 48.936170

# y 軸上のカテゴリ間の境界値を計算し、ymax と ymin 列に書き込む
prob1.dfm1 <- ddply(prob1.dfm, .(status), transform, ymax=cumsum(value))
prob1.dfm1 <- ddply(prob1.dfm1, .(status), transform, ymin=(ymax-value))

# 文字を表示する位置を決める
prob1.dfm1$xtext <- with(prob1.dfm1, xmin + (xmax - xmin)/2)
prob1.dfm1$ytext <- with(prob1.dfm1, ymin + (ymax - ymin)/2)

# モザイクプロットを作る
prob1.p <- ggplot(prob1.dfm1, aes(ymin=ymin, ymax=ymax,
                                  xmin=xmin, xmax=xmax,
                                  fill=variable))

prob1.p1 <- prob1.p + geom_rect(color = I("grey"))

prob1.p2 <- prob1.p1 + geom_text(aes(x=xtext, y=ytext,
                                     label = paste(round(value), "%")))
```

```
prob1.p3 <- prob1.p2 + geom_text(aes(x=xtext, y=103,
                                     label=c(rep("元職", 3),
                                             rep("新人", 3),
                                             rep("現職",3))),
                                 family="HiraKakuProN-W3")

prob1.p4 <- prob1.p3 + labs(x="", y="") +
  scale_fill_discrete(name="", labels=c("当選", "復活当選", "落選"))

print(prob1.p4)
# プロットを保存
ggsave("prob1.plot1.png")
## Saving 8 x 5 in image
```

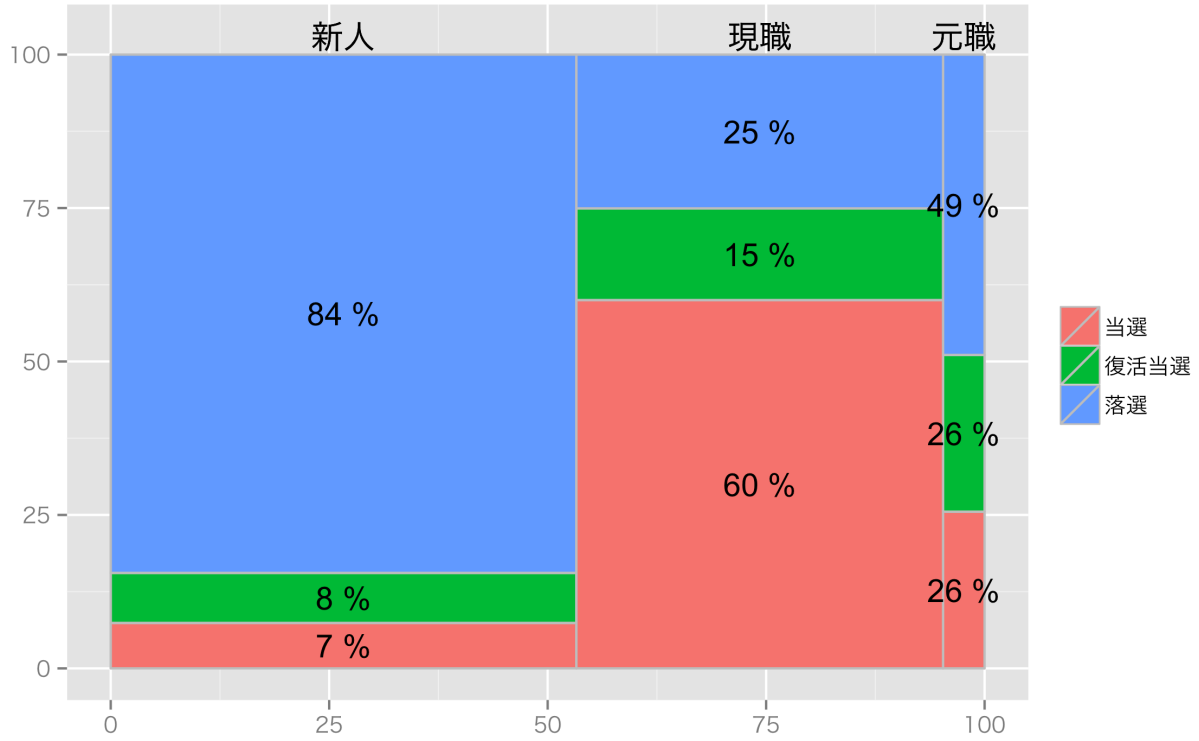

図 1. prob1.table1 のモザイクプロット

```
# ========================================
# モザイクプロットの作成２（prob1.table2）
# ========================================

# 政党ごとに立候補者数の合計を求める
table.party <- table(hw04.2005.5P$party)

# 表を行列に変換する
prob1.table2 <- as.matrix(prob1.table2[1:3, 1:5])
table.party <- as.matrix(table.party[1:5])
```

```r
# 各政党が全体に占める割合と、
# 各政党についての当選、復活当選、落選の割合を変数にする
prob2.df <- data.frame(party = levels(hw04.2005.5P$party),
                       party.pct = 100 * table.party / sum(table.party),
                       win = 100 * prob1.table2[2,] / table.party,
                       zombie = 100 * prob1.table2[3,] / table.party,
                       lose = 100 * prob1.table2[1,] / table.party)

# x 軸上のカテゴリの境界値を計算する
prob2.df$xmax <- cumsum(prob2.df$party.pct)
prob2.df$xmin <- prob2.df$xmax - prob2.df$party.pct

# 今後使わない変数 party.pct 列を削除する
prob2.df$party.pct <- NULL

# prob2.df を形を変換し、ggplot で使いやすくする。
prob2.dfm <- melt(prob2.df, id=c("party", "xmin", "xmax"))
print(prob2.dfm)
##     party     xmin      xmax variable     value
## 1  自民党  0.00000  32.18646      win 75.517241
## 2  民主党 32.18646  64.26193      win 17.993080
## 3  共産党 64.26193  94.78357      win  0.000000
## 4  公明党 94.78357  95.78246      win 88.888889
## 5  社民党 95.78246 100.00000      win  2.631579
## 6  自民党  0.00000  32.18646   zombie 16.551724
## 7  民主党 32.18646  64.26193   zombie 20.415225
## 8  共産党 64.26193  94.78357   zombie  1.454545
## 9  公明党 94.78357  95.78246   zombie  0.000000
## 10 社民党 95.78246 100.00000   zombie 10.526316
## 11 自民党  0.00000  32.18646     lose  7.931034
## 12 民主党 32.18646  64.26193     lose 61.591696
## 13 共産党 64.26193  94.78357     lose 98.545455
## 14 公明党 94.78357  95.78246     lose 11.111111
## 15 社民党 95.78246 100.00000     lose 86.842105

# y 軸上のカテゴリ間の境界値を計算し、ymax と ymin 列に書き込む
prob2.dfm1 <- ddply(prob2.dfm, .(party), transform, ymax=cumsum(value))
prob2.dfm1 <- ddply(prob2.dfm1, .(party), transform, ymin=(ymax-value))

# 文字を表示する位置を決める
prob2.dfm1$xtext <- with(prob2.dfm1, xmin + (xmax - xmin)/2)
prob2.dfm1$ytext <- with(prob2.dfm1, ymin + (ymax - ymin)/2)

# モザイクプロットを作る
prob2.p <- ggplot(prob2.dfm1, aes(ymin=ymin, ymax=ymax,
                                  xmin=xmin, xmax=xmax,
                                  fill=variable))

prob2.p1 <- prob2.p + geom_rect(color = I("white"))

prob2.p2 <- prob2.p1 + geom_text(aes(x=xtext, y=ytext,
                                     label = paste(round(value), "%")))
```

```
prob2.p3 <- prob2.p2 + geom_text(aes(x=xtext, y=103,
                                     label=c(rep("公明党", 3),
                                             rep("共産党", 3),
                                             rep("民主党", 3),
                                             rep("社民党", 3),
                                             rep("自民党", 3))),
                                 family="HiraKakuProN-W3")

prob2.p4 <- prob2.p3 + labs(x="", y="") +
  scale_fill_discrete(name="", labels=c("当選", "復活当選", "落選"))

# モザイクプロットを表示
print(prob2.p4)

# プロットを保存
ggsave("prob1.plot2.png")
## Saving 8 x 5 in image
```

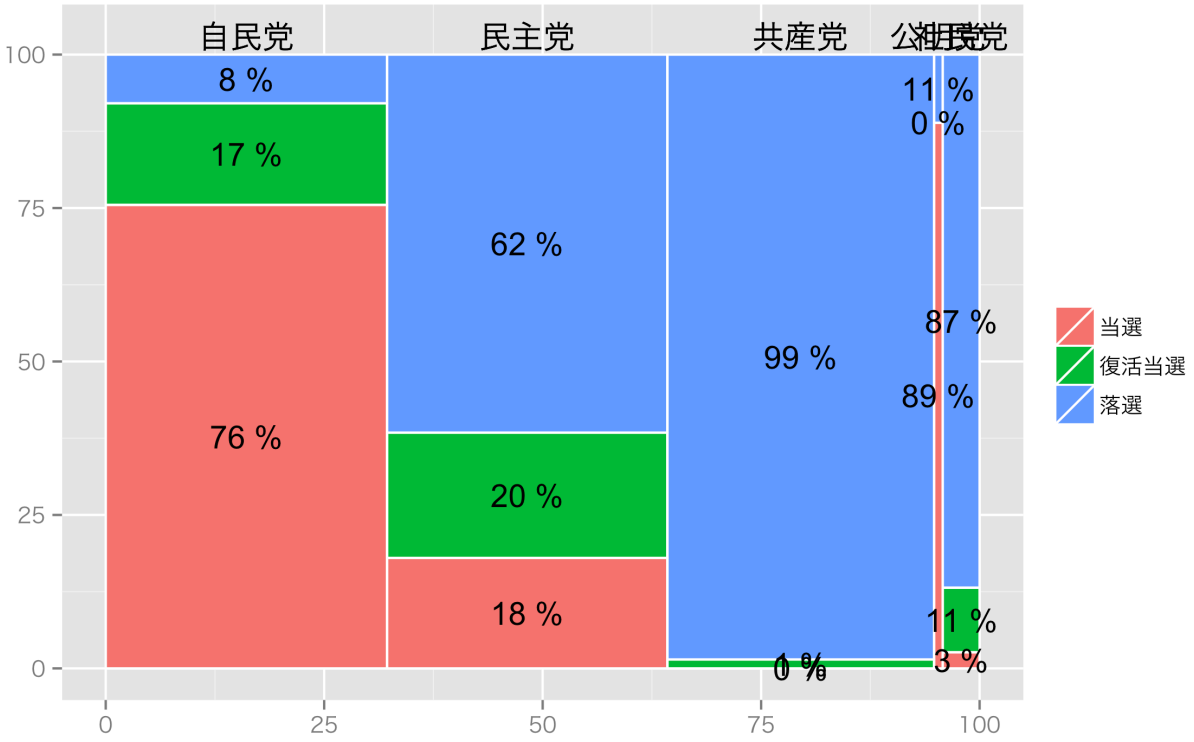

図 2. prob1.table2 のモザイクプロット

問題２

- データセットを使って回帰分析（OLS、ロジット、プロビット等、使えるものなら何でも良い）を行い、結果を奇麗な表にまとめなさい。また、同じ結果をキャタピラプロットで示しなさい

問題２では当選有無を従属変数とし、選挙費用、当選回数、年齢、立場が当選にどのような影響を与えるかを探索的に分析することを目的とする。従属変数はバイナリ変数であるため、プロビットモデルでパラメータを推定する。

```r
# ======
# 問題２
# ======

# 当選したか否かを従属変数とするため、当選ダミーを作る
hw04$win <- as.numeric(hw04$wl == "win")

# １円単位の場合、係数が０に近くなることがあるため
# 解釈の便宜のために１００万単位で再調整
hw04$exp.100man <- hw04$exp / 1000000

# プロビット分析の結果をprob2.probitへストック
# 従属変数：当選ダミー
# 独立変数：選挙費用（１００万円）
#           当選回数
#           年齢
#           立場（現職、新人、元職）

prob2.probit <- glm(win ~ exp.100man + previous + age + status, data=hw04,
                    family=binomial(link="probit"))
# 結果を表示する
summary(prob2.probit)

##
## Call:
## glm(formula = win ~ exp.100man + previous + age + status,
##     family = binomial(link = "probit"), data = hw04)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7655  -0.5258  -0.2746   0.4617   2.7243
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.443291   0.114424  -3.874 0.000107
## exp.100man       0.065208   0.004504  14.476  < 2e-16
## previous         0.237168   0.013316  17.810  < 2e-16
## age             -0.031844   0.002474 -12.872  < 2e-16
## statusincumbent  0.622989   0.066063   9.430  < 2e-16
## statusmoto       0.494990   0.095448   5.186 2.15e-07
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6405.2  on 5479  degrees of freedom
## Residual deviance: 4144.5  on 5474  degrees of freedom
##   (134 observations deleted due to missingness)
## AIC: 4156.5
##
## Number of Fisher Scoring iterations: 6
```

以上の結果を表でまとめると

表 4. プロビット分析の結果

	係数 （標準誤差）
切片	-0.4433***
	(0.1144)
選挙費用(100 万円)	0.0652***
	(0.0045)
当選回数	0.2372***
	(0.0133)
年齢	-0.0318***
	(0.0025)
立場	
現職	0.6230***
	(0.0661)
元職	0.4950***
	(0.0954)

注：両側検定
* : p≦0.05　** : p≦0.01　*** : p≦0.001

以上のようである。summary()を通じた一般化線形モデルの分析結果の要約から表示されない擬似決定係数やχ^2は省略した。

続いて、表４をキャタピラプロットで表示する。

```
# 各変数ごとの係数、９５％信頼区間（両側）をデータフレームとして保存
probit.df <- data.frame(variable = c("切片", "選挙費用(百万円)", "当選回数",
                                     "年齢", "現職", "元職"),
                        coef = coef(prob2.probit),
                        lower.se = confint(prob2.probit)[, 1],
                        upper.se = confint(prob2.probit)[, 2])
## Waiting for profiling to be done...
## Waiting for profiling to be done...

# 使わない行の名前を削除する
row.names(probit.df) <- NULL

# キャタピラプロットの表示
# 係数が高い方から表示する
coef.plot <- ggplot(probit.df, aes(x = reorder(variable, coef),
                                   y = coef,
                                   ymin = lower.se,
                                   ymax = upper.se)) +
  # 点と線の太さを指定
  geom_pointrange(size = 1.5) +
  # 基準となる０に直線を描く
  geom_hline(aes(intercept=0), linetype="dotted") +
  # x, y 軸のラベル設定
  xlab("独立変数") + ylab("係数の推定値") +
  # プロットを反時計回りする
  coord_flip()
```

```
# プロットを表示する。
print(coef.plot)
# プロットを保存
ggsave("prob2.plot.png")
## Saving 5 x 6 in image
```

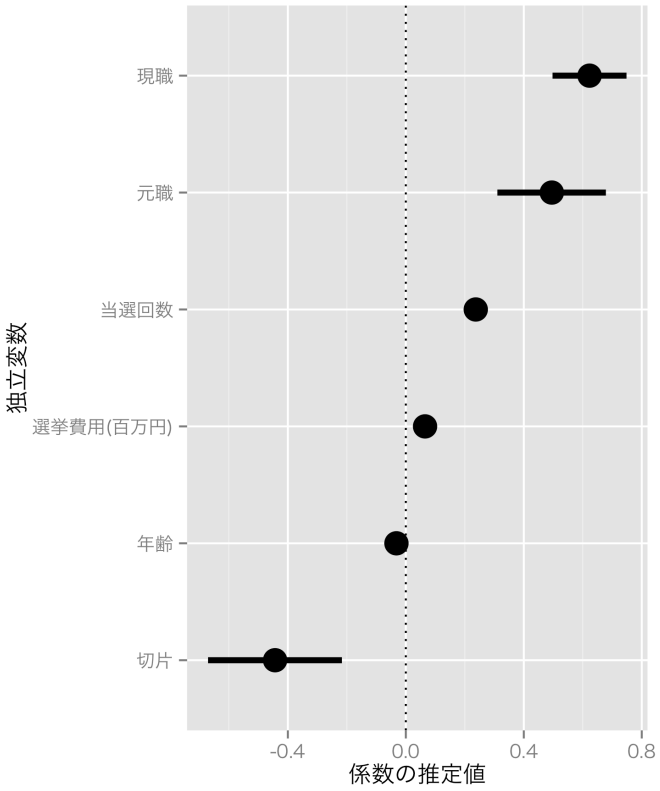

図 3. 表 4 のキャタピラプロット

問題３

- 自分で作った表と図を比較し、それぞれの長所と短所について論じなさい

1. クロス表とモザイクプロットの比較
   モザイクプロットの場合、各軸に対する割合が面積で総合的に確認できる長所があり、可読性の面から優れていることが分かる。しかし、図２の公明党や共産党のように１つの軸だけでも占める割合が小さい場合、文字が重なり可読性が落ちることもあろう。
2. プロビット分析の表とキャタピラプロットの比較
   この場合もクロス表とモザイクプロットの比較とほぼ同様である。可読性では大変優れているキャタピラプロットではあるが、有意確率などを直感的には分かるものの、正確には把握できない短所がある。
3. 全体的な比較をすると表からグラフへは完全にリプリケーションすることが出来るが、反対は近似的な形でしか再現できないという短所がある。キャタピラプロットの場合、点ごとに推定値を上書きする方法もあろうが、可読性を最大のメリットとするグラフ本来の目的が失われる恐れもあろう。