MrP Illustration

Lucas Leemann and Fabio Wasserfallen – Illustrative Example, Book Chapter for the Sage Handbook of Research Methods in Political Science and International Relations

10/5/2018

This document provides an illustration of how Multilevel Regression with Post-Stratification (MrP) works. The leading example is a post-vote survey in Switzerland following the vote on a Minaret ban in 2009. More information on this vote can be found here and here.

Set Up

We first set the correct working directory and then load the libraries that we will use in this exercise.

```
setwd("/Users/lleemann/Dropbox/Democratic Deficit exchange folder/Book chapter/Data and Code")
library(foreign)
library(lme4)
library(arm)
library(extrafont)
data1 <- read.dta("Minaret.dta")
```

After loading the data we can take a look at the variables. We see that there is a variable *minaret* which indicates whether a respondent voted yes or no. The other variables indicate gender, education level, age group, and canton of the respondents. We delete all respondents that did not vote and then look at the distribution of survey answers.

```
head(data1)
```

```
##   female minaret canton educ agegroup
## 1      1       0      3    5        4
## 2      1       0     19    6        1
## 3      1       0     22    6        1
## 4      1       1      5    2        2
## 5      0       0      2    5        1
## 6      1      NA     23    2        1
```

```
data2 <- data1[-which(is.na(data1$minaret)),]
table(data2$minaret)
```

```
##
##   0   1
## 351 330
```

```
table(data2$minaret)/sum(table(data2$minaret))
```

```
##
##         0         1
## 0.5154185 0.4845815
```

The raw distribution in the data shows that 52% of the people indicate that they voted no. But the actual vote was supported by a clear majority of 58%. This is not unusual for a small survey sample that the estimate is somewhat far from the actual outcome.

Multilevel Regression with Post-Stratification (MRP)

Let's say that we want to use the survey data to estimate the support for the minaret intiative per canton. A survey with only 680 observations to estimate support in 26 different cantons is not straight-forward. The most obvious first step is to take the average support of all people living in a specific canton as a measure of that canton's support for the minaret ban. But we can also go beyong this and exploit a hierarchical model.

First Step: Estimate Response Model

We estimate a hierarchical model in which we include a random effect that varies over cantons. On the individual level, we only include the variables *female*, *agegroup*, and *educ*. Rather than including dummies, we include them through random effects that vary over the groups in these variables:

```
model1 <- glmer(minaret ~ 1 + (1|female) + (1|agegroup) + (1|educ) + (1|canton),
                data= data2, family=binomial("probit"))
summary(model1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( probit )
## Formula: minaret ~ 1 + (1 | female) + (1 | agegroup) + (1 | educ) + (1 |
##     canton)
##    Data: data2
##
##      AIC      BIC   logLik deviance df.resid
##    872.8    895.4   -431.4    862.8      676
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5191 -0.7530 -0.5975  0.7028  1.6738
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  canton   (Intercept) 0.000000 0.00000
##  educ     (Intercept) 0.177576 0.42140
##  agegroup (Intercept) 0.004633 0.06806
##  female   (Intercept) 0.000000 0.00000
## Number of obs: 681, groups:  canton, 26; educ, 6; agegroup, 4; female, 2
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1010     0.1866  -0.541    0.588
```

We see from the output that we do not get any variance accross the cantons or accross gender groups. This sometimes happens when we rely on the **lme4::glmer** to estimate the hierarchical model. As an alternative, one could fit the model with **stan**. We will now use this Model in our next steps.

Second Step: Generate Predictions for Ideal Types

The second step is to create predict support probabilities for specific ideal types. Since we have used here gender (2), educ (6), agegroup (4), and canton (26) we have 1248 ideal types that we need to model. On the individual level there are 48 ideal types and they live in 26 cantons.

We read out the realizations of the random effects:

```r
re.female <- ranef(model1)$female[[1]]
re.agegroup <- ranef(model1)$agegroup[[1]]
re.educ <- ranef(model1)$educ[[1]]
re.canton <- ranef(model1)$canton[[1]]
```

In a next step, we will build the 48 ideal types (disregarding for the moment that they may vary over cantons). To do so, we create a vector with 48 elements and repeat the realizations such that all combinations are given:

```r
female.re <- rep(re.female,24)
age.re <- rep(kronecker(re.agegroup,c(1,1)), 6)
educ.re <- kronecker(re.educ,rep(1, 8))
ind.re <- rowSums(cbind(female.re, age.re, educ.re))
ind.re <- ind.re + fixef(model1)
```

The object **ind.re** contains now 48 elements and each element has all the information on the individual level for a specific ideal type. Now, we must add the cantonal part to it (Note: since we have no context-level variables in the response model and since the estimated variance was accross cantons was 0, we could skip this step. But usually, one will have conetxt-level variables and the unit-random-effects will vary). Hence, we create a vector of length 48*26:

```r
y.lat1 <- rep(NA,1248)
for (i in 1:26){
  a <- ((i-1)*48)+1
  b <- a + 47
  y.lat1[a:b] <- ind.re + re.canton[i]
}
```

Everything we have done so far was on the latent variable. The last step is to transform these scores to predicted probabilities.

```r
p1 <- pnorm(y.lat1)
```

Third Step: Post-Stratify by Ideal Types

We first load an object that contains for each of the 48 ideal types the number of people living in a specifi canton.

```r
load("Census.Rda")
dim(Censusobject)
```

```
## [1] 48 26
```

```r
head(Censusobject)
```

```
##       [,1]  [,2]  [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,] 10742  9114  3250  336 1269  321  305  253  902  2354  2190  1816
## [2,] 10635  9448  3696  526 1544  351  330  332  825  2705  2212  1685
## [3,] 11343 11569  4665  751 2232  540  553  591  936  3932  2984  2176
## [4,] 17568 18249  7624 1300 3685  945  989  914 1607  7342  5026  2806
## [5,] 11203 13667  5841 1046 2623  722  677  667 1098  5524  3284  2092
## [6,] 24240 26592 10712 1586 4377 1064 1256 1153 2343  9590  7341  4279
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
## [1,]  2030   566   467   118  3893  1706  4545  1971  2377  5383  2719
## [2,]  1933   558   511   130  4515  1948  4396  2132  2763  4915  2855
## [3,]  2362   689   816   255  5212  2379  5972  2513  2902  5600  3875
## [4,]  4180  1086  1183   396  9610  4093 10086  4428  5492  9221  8209
## [5,]  2805   743   984   440  6792  3116  6833  2970  4200  7278  5564
```

```
## [6,]  7053  2066  1636   560 14044  5883 13987  6325  9540 14888 12171
##      [,24] [,25] [,26]
## [1,]  1569  3896   831
## [2,]  1581  3535   788
## [3,]  1985  3362  1248
## [4,]  3372  5178  2405
## [5,]  2253  3637  1696
## [6,]  4938  7301  3137
```

We can now first estimate support in Switzerland to see if we will still be so far off. To do so, we multiply the vector of predicted probabilities with the number of people of that type and divide by the population:

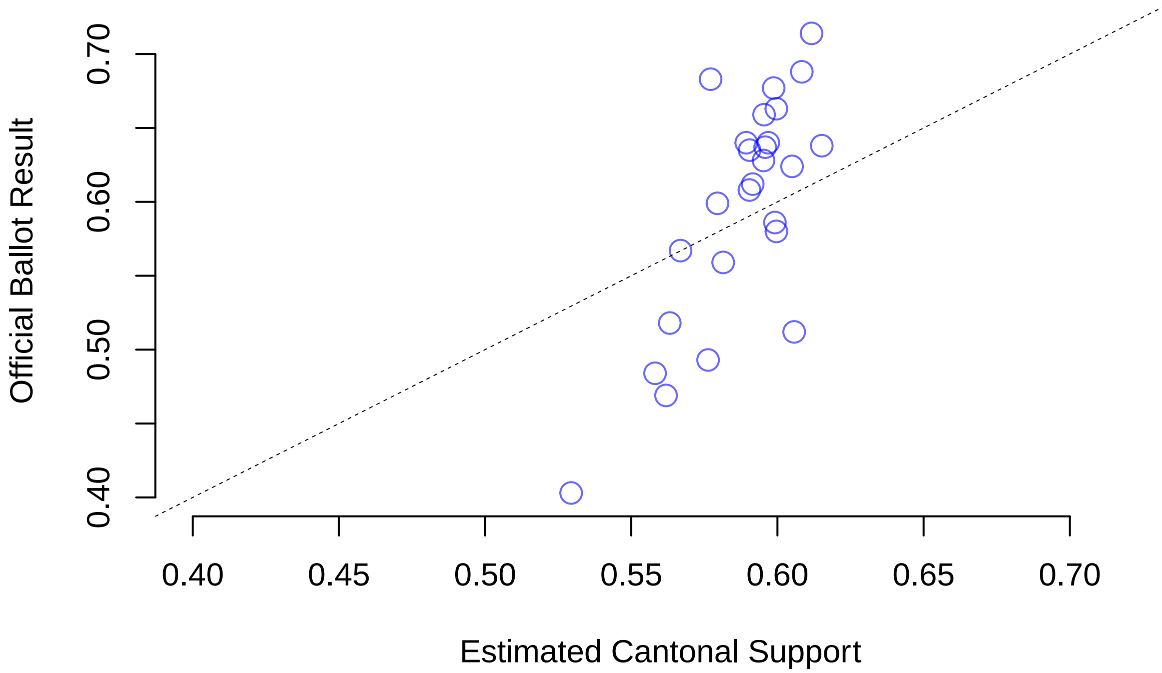
```
a <- c(Censusobject)
# Estimate
sum(p1*a)/sum(a)
```

```
## [1] 0.5800503
```

This is a stunning result. The official outcome was 57.5%, the raw data showed 48.5% support and the MrP estimate delivers an almost perfect estimate of 58%! So, in this example, MrP helped with a biased sample that over-counted well-educated people and under-counted people with less education. We can also exploit this to see our cantonal estimates.

We essentially go through our vector with the 1248 ideal types and cut for each canton its 48 ideal types out. We then post-stratify by the census information from that canton and store the estimate. We can plot this against the true outcome data for that vote.

```
mrp.minaret1 <- rep(NA,26)
for (i in 1:26){
  a1 <- ((i-1)*48)+1
  a2 <- a1 + 47
  p1 <- pnorm(y.lat1[a1:a2])
  a <- Censusobject[,i]
  mrp.minaret1[i] <-  sum(p1*a)/sum(a)
}

MINARET <- c(51.80,60.80,61.20,63.80,66.30,62.40,62.80,68.80,56.70,55.90,64.00,48.40,
             59.90,63.50,63.70,71.40,65.90,58.60,64.00,67.70,68.30,46.90,58.00,49.30,
             40.30,51.20)/100


plot(mrp.minaret1, MINARET, pch=21, cex=1.5, col=rgb(0,0,255,150,maxColorValue=255),
     bty="n", ylab="Official Ballot Result", xlab="Estimated Cantonal Support",
     ylim=c(.4,.72), xlim=c(.4,.72))
abline(c(0,1), lty=2, lwd=.5)
```



We see immediately that we have estimates that do not vary sufficiently. We under-estimate the high outcomes and over-estimate the low outcomes. But since the model above estimated the variance of the cantonal random effect to be 0 (which it is most likely not), all variation is the consequence of a different social make-up of the population. We can considerally improve the estimates by adding context-level information into the response model.

Adding Context-Level Variables

As mentioned above, we will often want to include context-level variables and this part illustrates how we can do so. One variable we can use here is a vote that took place a year prior to this vote. It was aimed at over-turning a decision of the courts regarding naturalization decisions and very popular among the right.

```
rightP <- c(39.30,36.70,44.30,46.50,59.90,47.10,49.10,48.90,44.30,
            27.00,41.40,28.50,35.20,42.80,42.60,48.30,48.30,34.90,
            46.80,48.90,42.20,19.00,25.00,18.00,17.90,19.80)/100
# source: https://www.bk.admin.ch/ch/d/pore/va/20080601/can532.html
canton <- c(1:26)
data3 <- cbind(rightP,canton)
data4 <- merge(data2,data3,by="canton")

#### MRP
model2 <- glmer(minaret ~ rightP + (1|female) + (1|agegroup) + (1|educ) + (1|canton),

                data= data4, family=binomial("probit"))
summary(model2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( probit )
## Formula: minaret ~ rightP + (1 | female) + (1 | agegroup) + (1 | educ) +
##     (1 | canton)
##    Data: data4
```

```
##
##      AIC      BIC   logLik deviance df.resid
##    870.3    897.5   -429.2    858.3      675
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6711 -0.7568 -0.5160  0.7554  1.9568
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  canton   (Intercept) 0.000000 0.00000
##  educ     (Intercept) 0.176852 0.42054
##  agegroup (Intercept) 0.004606 0.06787
##  female   (Intercept) 0.000000 0.00000
## Number of obs: 681, groups:  canton, 26; educ, 6; agegroup, 4; female, 2
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.4712     0.2565  -1.837   0.0662 .
## rightP        1.0211     0.4860   2.101   0.0356 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##        (Intr)
## rightP -0.687
```

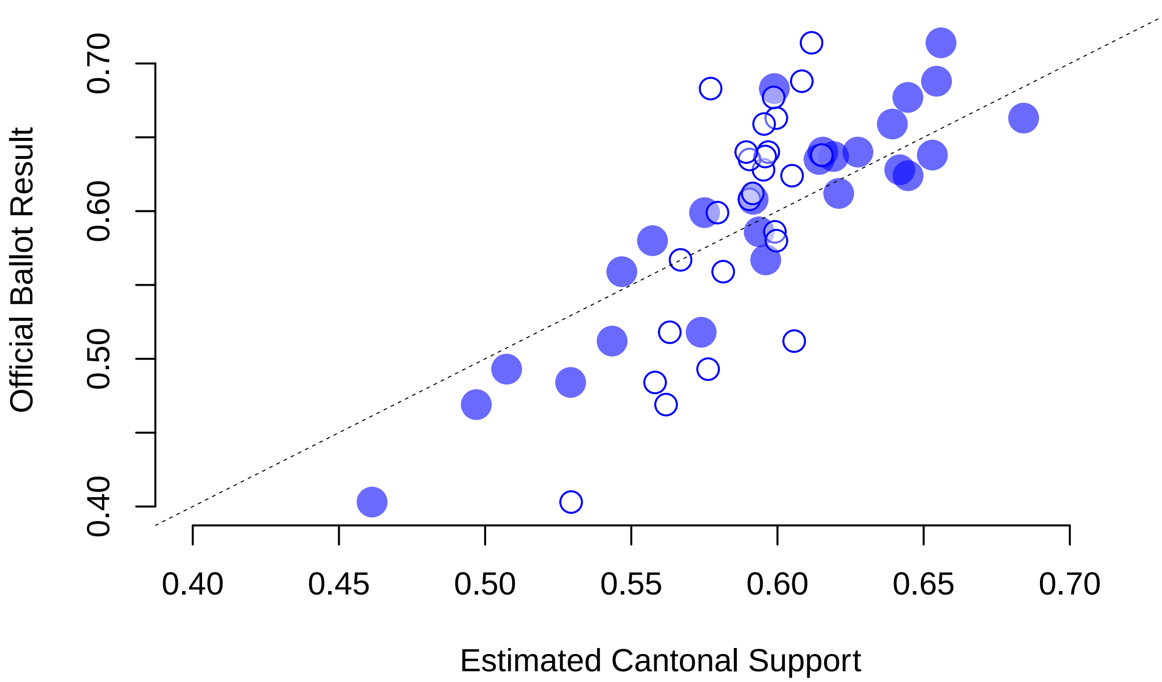
We can take these model estimates and carry out steps 2 and 3 from above but changing step 2 slightly to account for the context-level explanatory factor:

```r
re.female <- ranef(model2)$female[[1]]
re.agegroup <- ranef(model2)$agegroup[[1]]
re.educ <- ranef(model2)$educ[[1]]
re.canton <- ranef(model2)$canton[[1]]

female.re <- rep(re.female,24)
age.re <- rep(kronecker(re.agegroup,c(1,1)), 6)
educ.re <- kronecker(re.educ,rep(1, 8))
ind.re <- rowSums(cbind(female.re, age.re, educ.re))
ind.re <- ind.re + fixef(model2)[1]
beta1 <- fixef(model2)[2]

y.lat2 <- rep(NA,1248)
for (i in 1:26){
  a <- ((i-1)*48)+1
  b <- a + 47
  y.lat2[a:b] <- ind.re + beta1 * rightP[i] + re.canton[i]
}

p2 <- pnorm(y.lat2)
a <- c(Censusobject)
sum(p2*a)/sum(a)
```

```
## [1] 0.581083
```

The national estimate does not change much, but we want to see whether this changes the cantonal estimates since we now gave the model more structure.

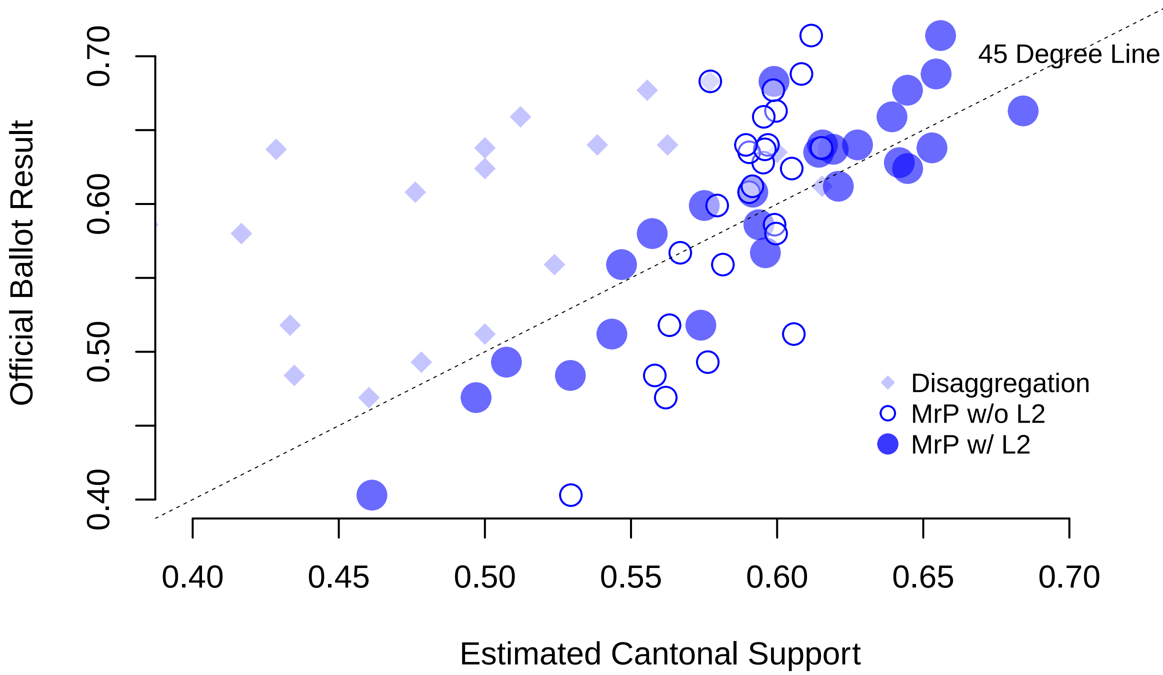
```
# Cantonal estimates
mrp.minaret2 <- rep(NA,26)
for (i in 1:26){
  a1 <- ((i-1)*48)+1
  a2 <- a1 + 47
  p2 <- pnorm(y.lat2[a1:a2])
  a <- Censusobject[,i]
  mrp.minaret2[i] <-  sum(p2*a)/sum(a)
}

#par(family="CMU Serif")
plot(mrp.minaret2, MINARET, pch=20, cex=3, col=rgb(0,0,255,150,maxColorValue=255),
     bty="n", ylab="Official Ballot Result", xlab="Estimated Cantonal Support",
     ylim=c(.4,.72), xlim=c(.4,.72))
points(mrp.minaret1,MINARET, col="blue", pch=21, cex=1.5,
       bg=rgb(255,255,255,100,maxColorValue = 255))
abline(c(0,1), lty=2, lwd=.5)
```



This now looks much better. Although the national estimate did not improve, the cantonal-level estimates are much better than in the emtpty model. We can now also contrast this with the approach of disaggregation.

```
TAB <- table(attributes(model2)$frame$canton,attributes(model2)$frame$minaret)
dis.minaret <- TAB[,2]/rowSums(TAB)

#par(family="CMU Serif")
plot(mrp.minaret2, MINARET, pch=20, cex=3, col=rgb(0,0,255,150,maxColorValue=255),
     bty="n", ylab="Official Ballot Result", xlab="Estimated Cantonal Support",
     ylim=c(.4,.72), xlim=c(.4,.72))
points(dis.minaret,MINARET, pch=18, col=rgb(0,0,255,60,maxColorValue = 255),
       cex=1.5)
points(mrp.minaret1,MINARET, col="blue", pch=21, cex=1.5,bg=rgb(255,255,255,100,maxColorValue = 255))
abline(c(0,1), lty=2, lwd=.5)
```

```r
legend(.63,.5,legend = c("Disaggregation","MrP w/o L2", "MrP w/ L2"), pch=c(18,21,20),
       col=c(rgb(0,0,255,60,maxColorValue = 255),"blue",
             rgb(0,0,255,200,maxColorValue = 255)), bty="n", pt.cex=c(1,1,2), cex=0.8)
text(.7,.7,"45 Degree Line", cex=0.8)
```

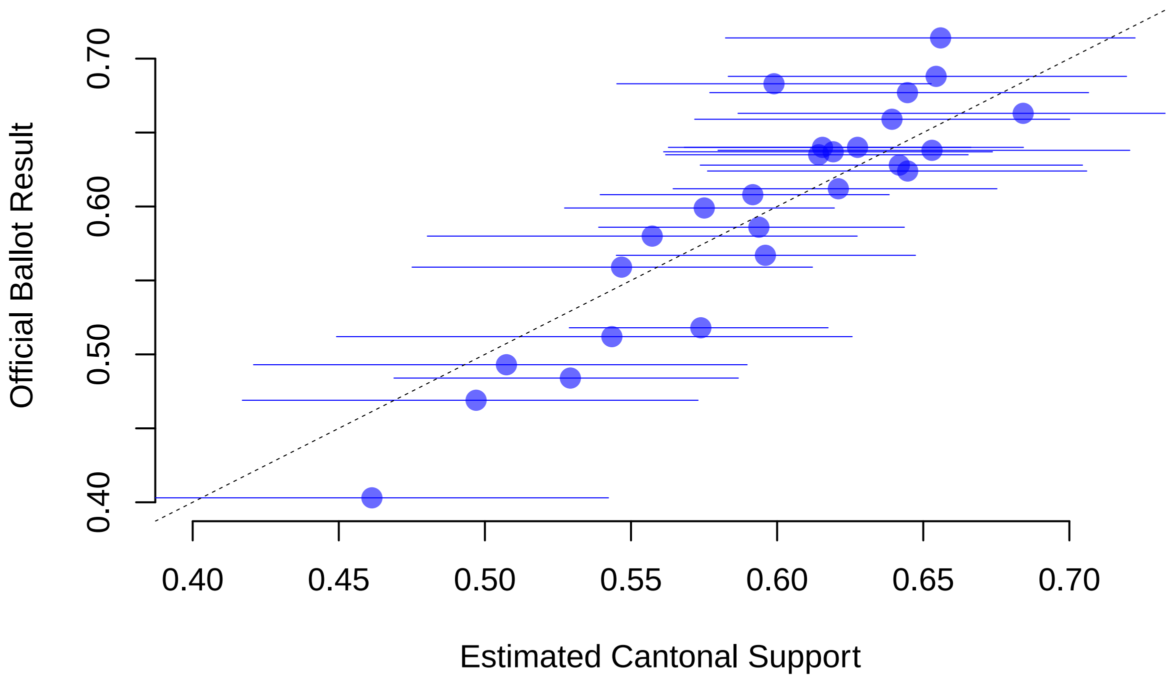


Adding Uncertainty

To add uncertainty, we rely on a simulation approach. We use all the model-uncertainty via simulation to illustrate how precide our estimates are.

```r
BLOCK <- sim(model2, n=1000)

re.female <- attributes(BLOCK)$ranef$female
re.agegroup <- attributes(BLOCK)$ranef$agegroup
re.educ <- attributes(BLOCK)$ranef$educ
re.canton <- attributes(BLOCK)$ranef$canton


female.re <- matrix(NA, 48,1000)
  for (i in 1:1000){
    female.re[,i] <- rep(re.female[i,,1],24)
  }

age.re <- matrix(NA, 48,1000)
  for (i in 1:1000){
    age.re[,i] <- rep(kronecker(re.agegroup[i,,1],c(1,1)), 6)
  }

educ.re <- matrix(NA, 48,1000)
for (i in 1:1000){
  educ.re[,i] <-   kronecker(re.educ[i,,1],rep(1, 8))
```

```
}

ind.re <- female.re + age.re + educ.re


y.lat2 <- matrix(NA,1248,1000)
for (i in 1:26){
  a <- ((i-1)*48)+1
  b <- a + 47
  level2 <- attributes(BLOCK)$fixef %*% matrix(c(1,rightP[i]),2,1) + re.canton[i]
  level2.48 <- matrix(rep(level2,48),48,1000, byrow = TRUE)
  y.lat2[a:b,] <- ind.re + level2.48
}


# Cantonal estimates
mrp.minaret2.unc <- matrix(NA,26,1000)
for (i in 1:26){
  a1 <- ((i-1)*48)+1
  a2 <- a1 + 47
  p2 <- pnorm(y.lat2[a1:a2,])
  a <- Censusobject[,i]
  mrp.minaret2.unc[i,] <-  t(p2)%*%a/sum(a)
}

size.canton <- table(data2$canton)


plot(mrp.minaret2, MINARET, pch=20, cex=2,
     col=rgb(0,0,255,150,maxColorValue=255),
     bty="n", ylab="Official Ballot Result", xlab="Estimated Cantonal Support",
     ylim=c(.4,.72), xlim=c(.4,.72))
  for (i in 1:26){
    draws <- mrp.minaret2.unc[i,]
    CI <- quantile(draws,c(0.025,0.975))
    segments(CI[1],MINARET[i],CI[2],MINARET[i], col="blue", lwd=0.5)
  }
abline(c(0,1), lty=2, lwd=.5)
```


Official Ballot Result
0.40
0.50
0.60
0.70
0.40
0.45
0.50
0.55
0.60
0.65
0.70
Estimated Cantonal Support