STA 360: Homework 10c

Samuel Eure

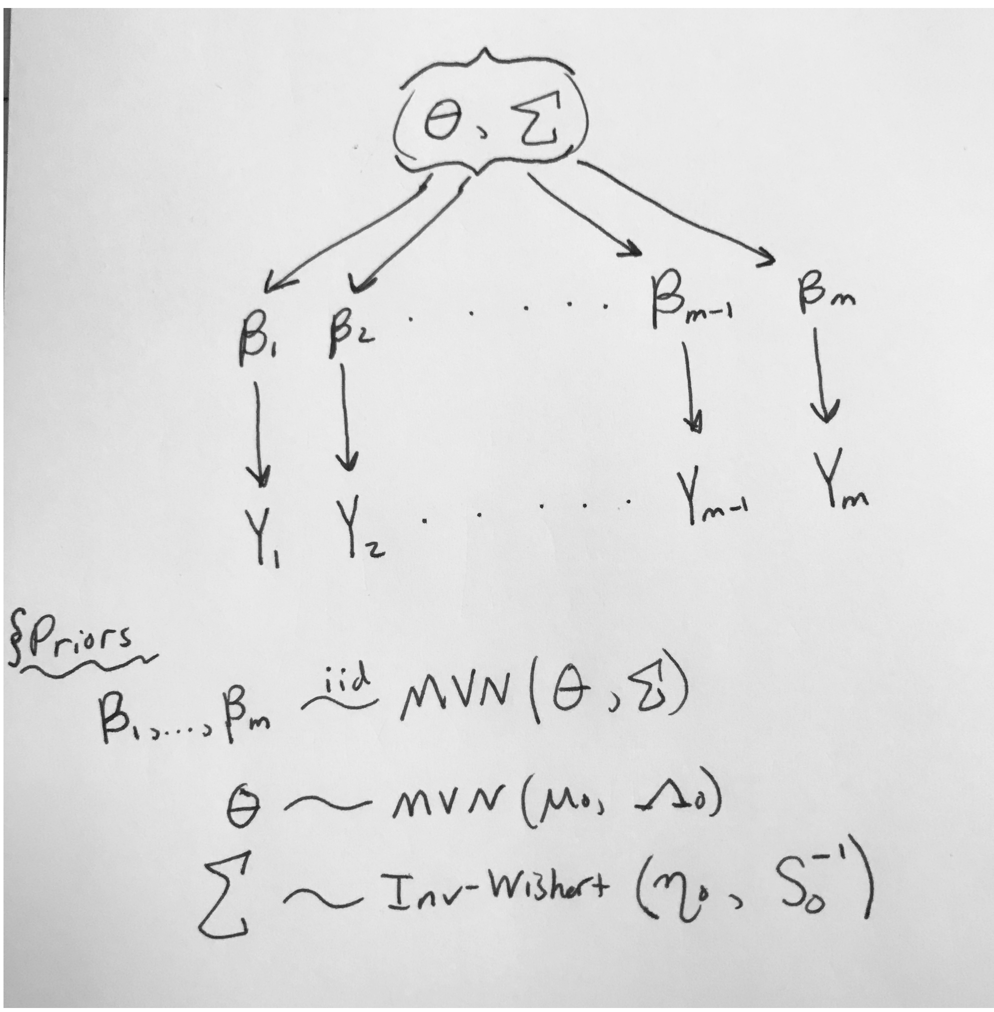
12/2/2018

Problem 11.4

The Data

```
data <- read.csv("mathstandard.txt", header = T)
counties <- unique(data$county)
Y <- data$metstandard;
X <- matrix(c(data$percentms), ncol = 1);
```

Part a



.

Part b

Logistic

```
BetaEstimate <- matrix(0,nrow = length(counties), ncol = 3);
colnames(BetaEstimate)<- c('intercept', 'percentMasters', 'samples');
rownames(BetaEstimate)<- counties;
plotXValues <- c();
countyToX   <- list();
countyToY   <- list();
countyToProbs <- list();
for(j in 1:length(counties)){
  countyX1         <- X[data$county == counties[j]];
  countyY          <- Y[data$county == counties[j]];
  model.j          <- glm(countyY ~ 1+ countyX1, family = binomial);
  beta.j           <- model.j$coef;
  BetaEstimate[j,c(1,2)] <- beta.j;
  if(length(countyY) == 1){BetaEstimate[j,2] <-0; beta.j[2]<-0}
  countyToX[[j]] <- countyX1; #This will be used in the MH step
  countyToY[[j]] <- countyY;  #This will be used in the MH step
  BetaEstimate[j,3] <- length(countyY)
  z <-countyX1*beta.j[2]+beta.j[1];
  countyToProbs[[j]] <- exp(z)/(1+exp(z));
}
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
BetaEstimate
```

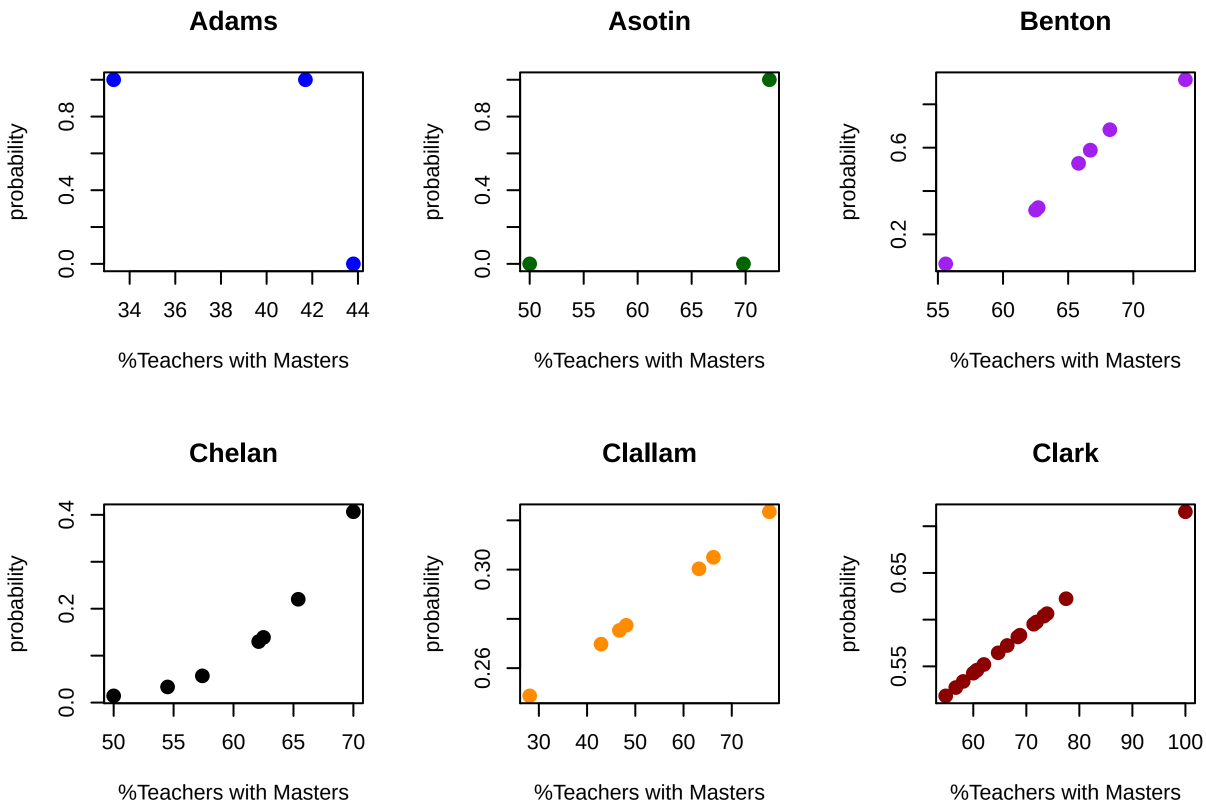
```
##                  intercept percentMasters samples
## Adams          931.2082644  -2.178458e+01       3
## Asotin       -1376.5980410   1.938916e+01       3
## Benton         -17.8895771   2.735615e-01       8
```

```
## Chelan         -13.8759019   1.928240e-01       7
## Clallam         -1.3131724   7.395515e-03       7
## Clark           -0.9551942   1.877388e-02      19
## Columbia        22.5660686   0.000000e+00       1
## Cowlitz         -0.3564859  -2.238540e-03       8
## Douglas        -23.5660685   1.698666e-16       3
## Ferry          -23.5660685   4.511809e-16       3
## Franklin     -2094.1646233   3.221858e+01       4
## Garfield       -22.5660685   0.000000e+00       1
## Grant           -1.7867982   3.004804e-03      12
## Grays Harbor    -8.4978525   1.174606e-01      10
## Island       -5398.6556245   8.886664e+01       3
## Jefferson       23.5660689  -6.671826e-12       3
## King            -2.8641607   5.537192e-02      64
## Kitsap          -3.4675513   5.926339e-02      10
## Kittitas     -3110.1242439   5.053044e+01       3
## Klickitat       -2.0628118   3.566456e-02       4
## Lewis           -3.5101909   3.372968e-02      12
## Lincoln       -233.7985949   3.054159e+00       5
## Mason         -153.9108965   2.600869e+00       5
## Okanogan       -14.3877504   2.177999e-01       7
## Pacific       -629.3599478   9.538342e+00       4
## Pend Oreille  -801.7189011   1.334867e+01       3
## Pierce          -7.3545109   1.064683e-01      27
## San Juan        23.5660689  -1.358459e-11       3
## Skagit          -2.3920152   4.282731e-02       7
## Skamania       -22.5660685   0.000000e+00       1
## Snohomish       -1.2003330   1.495265e-02      30
## Spokane          2.3035123  -3.555875e-02      27
## Stevens         -2.2734717   4.176312e-02       7
## Thurston        -1.5709685   3.233877e-02      13
## Wahkiakum      -22.5660685   0.000000e+00       1
## Walla Walla     -3.8581685   2.878328e-02       7
## Whatcom         -8.5641958   1.818284e-01      10
## Whitman        885.2327349  -1.007939e+01       9
## Yakima          -4.0847169   2.081292e-02      20
```

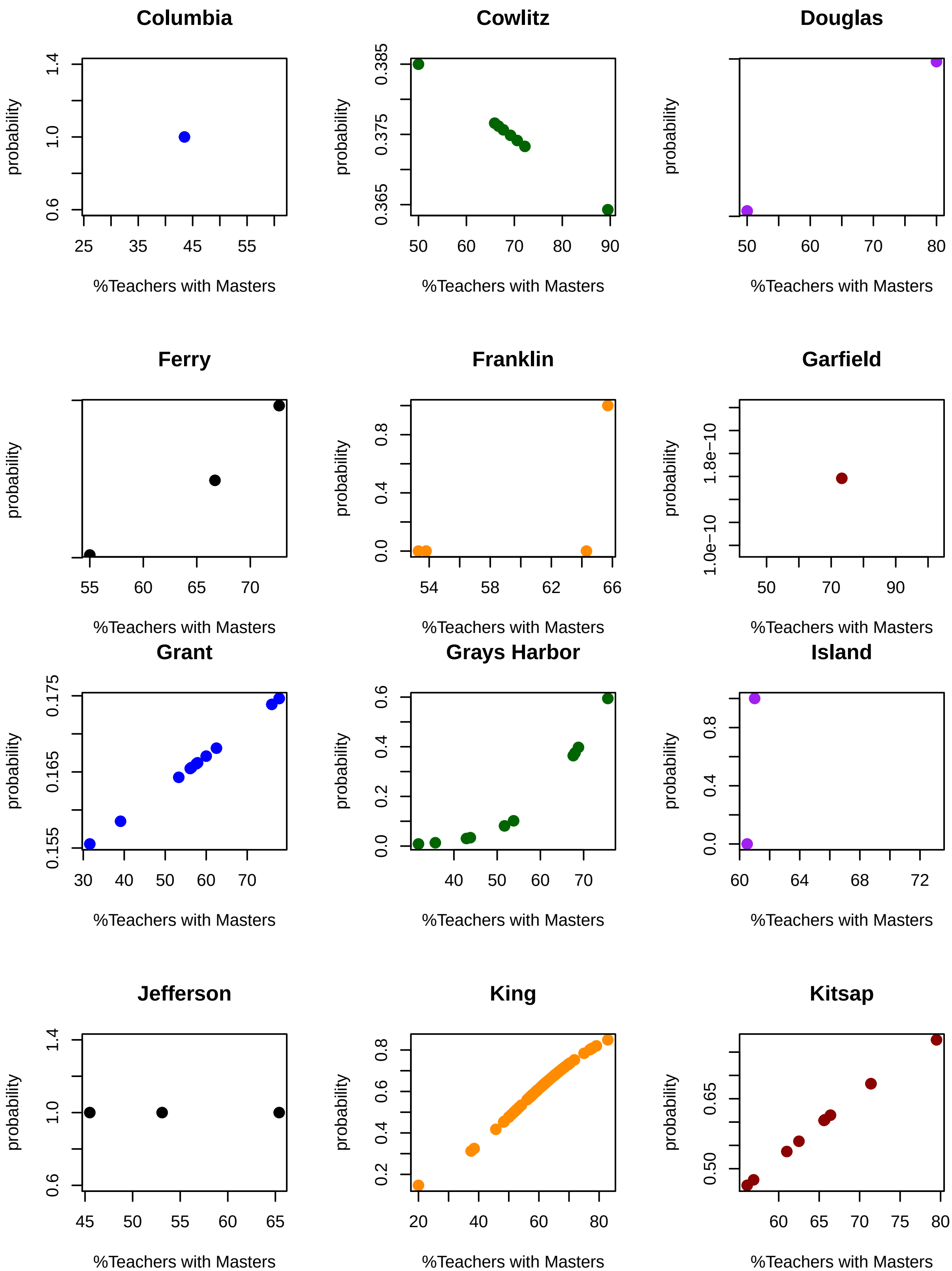
The table above shows my initial estimates of the values of β_j for each county j. As shown, some of these values are 0.0000, which may look like a computation error or data entry error. However, these values were initially NA, I simply changed them to zero for an important reason. First, notice that all the counties j which have a NA value for $x_{1,j}$ also only have one sample (or school) which represents them in the data set. Thus, a regression line can only be a constant (the intercept) value since one must have two data points to calculate slope.

```r
i=1
C <- c('darkred', 'blue', 'darkgreen','purple','black','darkorange')
while(i < length(counties)){
  par(mfrow = c(2,3))
  top <- min((i+5),length(counties))
  for(j in i:top){
    countyX1 <- X[data$county == counties[j]];
    plot(countyX1, countyToProbs[[j]], main = counties[j],
         xlab = '%Teachers with Masters', ylab = 'probability',
```

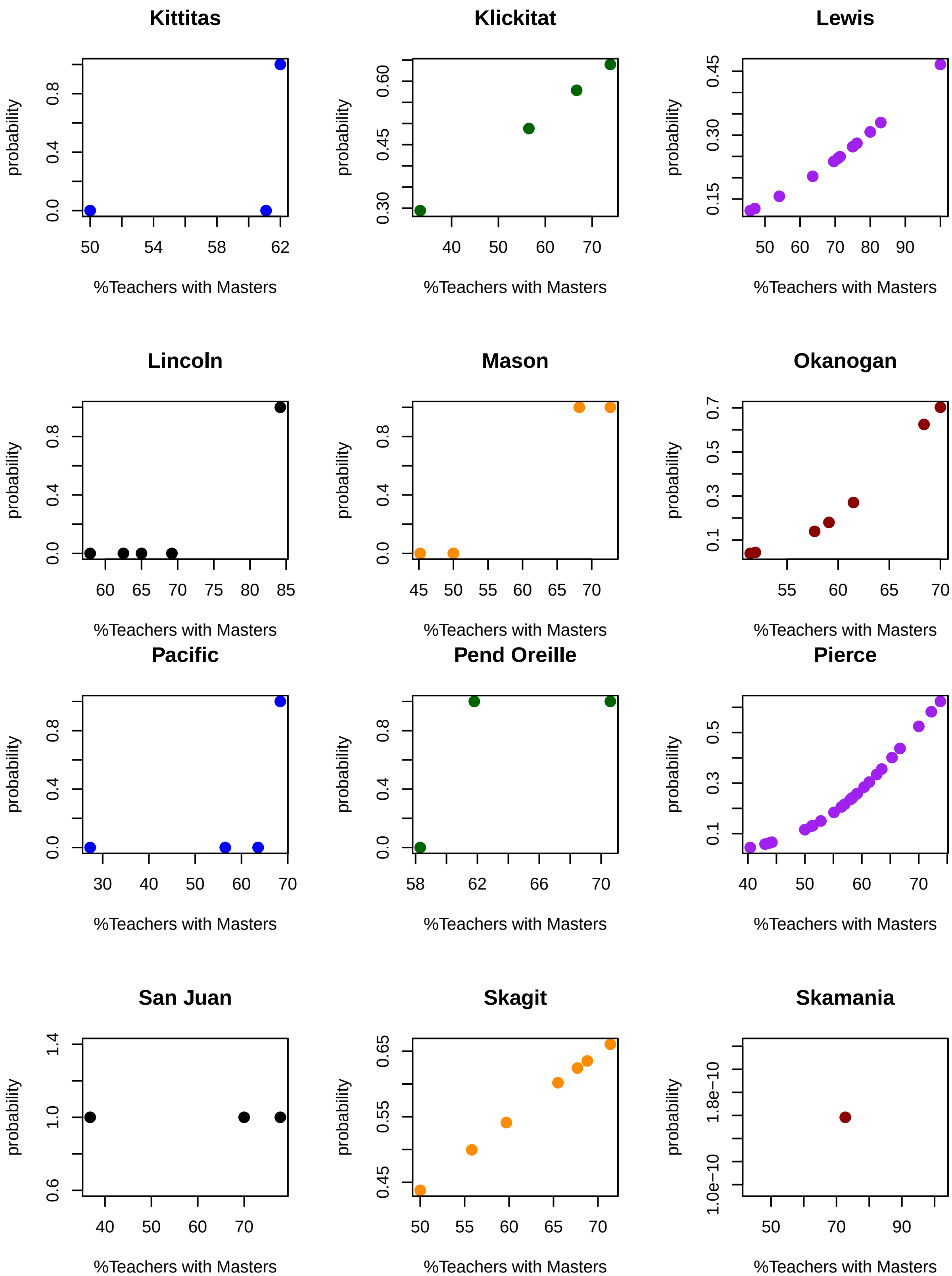
```
            col = C[j%%6+1], pch = 20, cex = 2)
  }
  i <- i + 6;
}
```

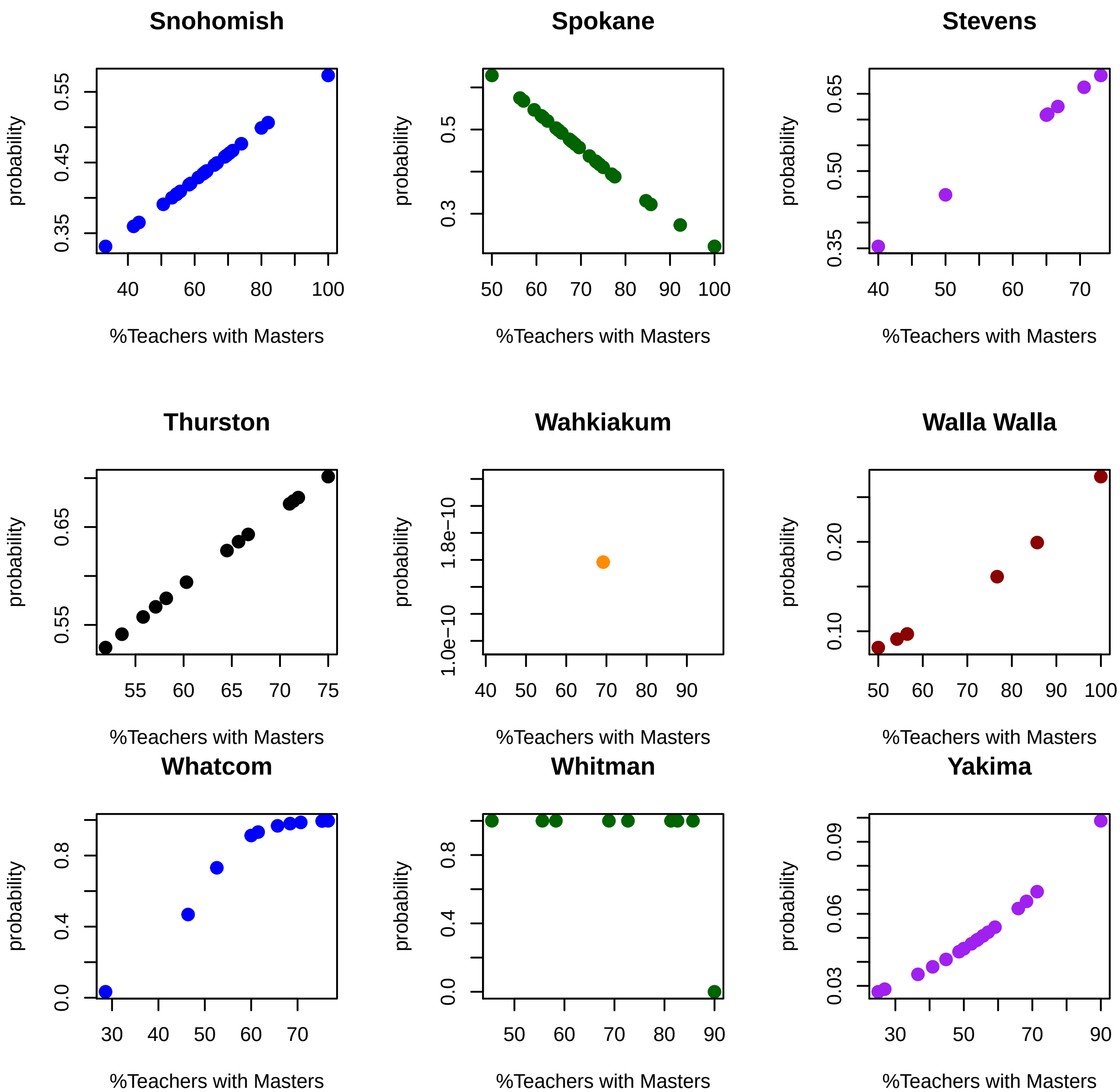



```
## Warning in plot.window(...): relative range of values = 30 * EPS, is small
## (axis 2)

## Warning in plot.window(...): relative range of values = 23 * EPS, is small
## (axis 2)
```



Columbia
probability
0.6
1.0
1.4
25
35
45
55
%Teachers with Masters
Cowlitz
probability
0.365
0.375
0.385
50
60
70
80
90
%Teachers with Masters
Douglas
probability
50
60
70
80
%Teachers with Masters
Ferry
probability
55
60
65
70
%Teachers with Masters
Franklin
probability
0.0
0.4
0.8
54
58
62
66
%Teachers with Masters
Garfield
probability
1.0e−10
1.8e−10
50
70
90
%Teachers with Masters
Grant
probability
0.155
0.165
0.175
30
40
50
60
70
%Teachers with Masters
Grays Harbor
probability
0.0
0.2
0.4
0.6
40
50
60
70
%Teachers with Masters
Island
probability
0.0
0.4
0.8
60
64
68
72
%Teachers with Masters
Jefferson
probability
0.6
1.0
1.4
45
50
55
60
65
%Teachers with Masters
King
probability
0.2
0.4
0.6
0.8
20
40
60
80
%Teachers with Masters
Kitsap
probability
0.50
0.65
60
65
70
75
80
%Teachers with Masters


Kittitas
probability
0.0 0.4 0.8
50 54 58 62
%Teachers with Masters
Klickitat
probability
0.30 0.45 0.60
40 50 60 70
%Teachers with Masters
Lewis
probability
0.15 0.30 0.45
50 60 70 80 90
%Teachers with Masters
Lincoln
probability
0.0 0.4 0.8
60 65 70 75 80 85
%Teachers with Masters
Mason
probability
0.0 0.4 0.8
45 50 55 60 65 70
%Teachers with Masters
Okanogan
probability
0.1 0.3 0.5 0.7
55 60 65 70
%Teachers with Masters
Pacific
probability
0.0 0.4 0.8
30 40 50 60 70
%Teachers with Masters
Pend Oreille
probability
0.0 0.4 0.8
58 62 66 70
%Teachers with Masters
Pierce
probability
0.1 0.3 0.5
40 50 60 70
%Teachers with Masters
San Juan
probability
0.6 1.0 1.4
40 50 60 70
%Teachers with Masters
Skagit
probability
0.45 0.55 0.65
50 55 60 65 70
%Teachers with Masters
Skamania
probability
1.0e−10 1.8e−10
50 70 90
%Teachers with Masters



Probit

```
pro.BetaEstimate <- matrix(0,nrow = length(counties), ncol = 3);
colnames(pro.BetaEstimate)<- c('intercept', 'percentMasters', 'samples');
rownames(pro.BetaEstimate)<- counties;
plotXValues <- c();
pro.countyToProbs <- list();
for(j in 1:length(counties)){
  countyX1          <- X[data$county == counties[j]];
  countyY           <- Y[data$county == counties[j]];
  model.j           <- glm(countyY ~ 1+ countyX1, family = binomial(link = "probit"));
  beta.j            <- model.j$coef;
  pro.BetaEstimate[j,c(1,2)] <- beta.j;
  if(length(countyY) == 1){pro.BetaEstimate[j,2] <-0; beta.j[2]<-0}
  pro.BetaEstimate[j,3] <- length(countyY)
```

```r
  z <-countyX1*beta.j[2]+beta.j[1];
  pro.countyToProbs[[j]] <- pnorm(z);
}
```

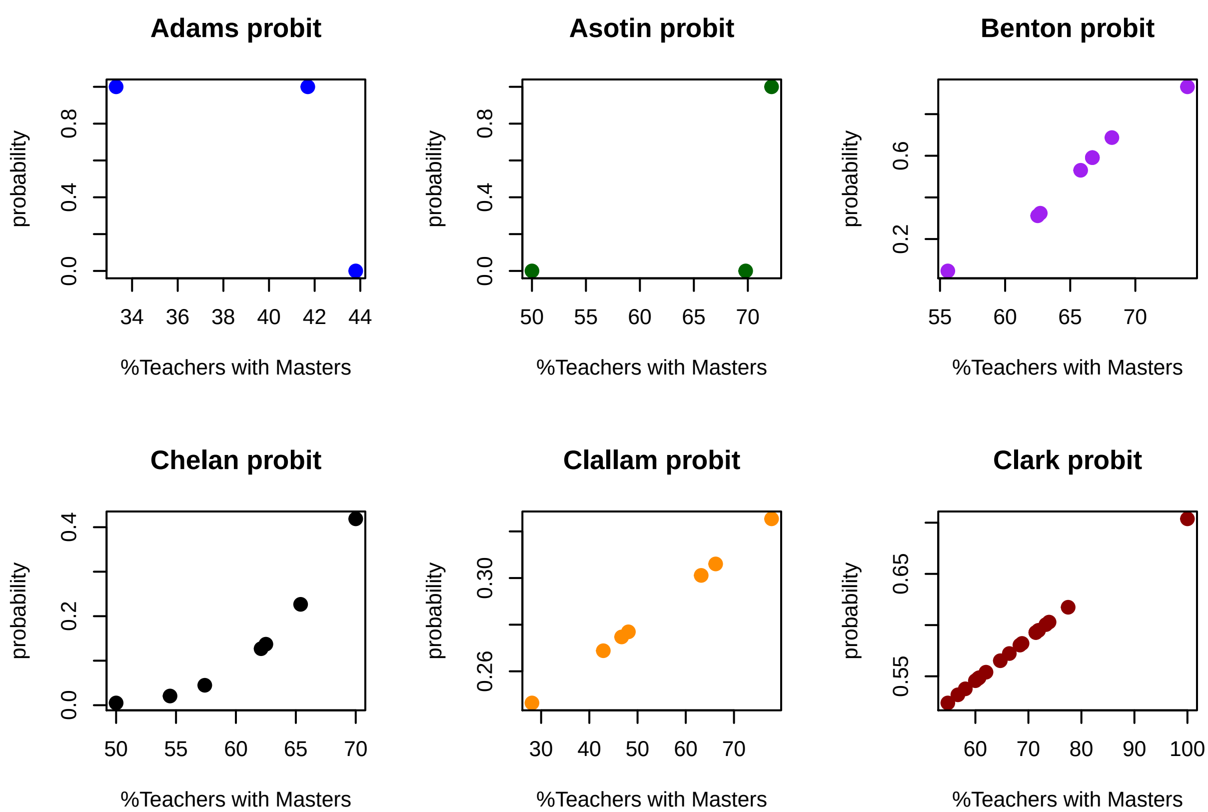
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
pro.BetaEstimate
```

```
##                   intercept percentMasters samples
## Adams           261.1533328  -6.109142e+00       3
## Asotin         -375.2196571   5.284843e+00       3
## Benton          -11.2237251   1.717281e-01       8
## Chelan           -8.4979976   1.184708e-01       7
## Clallam          -0.8175796   4.690423e-03       7
## Clark            -0.5159491   1.051194e-02      19
## Columbia          6.2437504   0.000000e+00       1
## Cowlitz          -0.2146536  -1.508529e-03       8
## Douglas          -6.4000838   4.573291e-17       3
## Ferry            -6.4000838   1.644081e-16       3
## Franklin       -591.3443818   9.097686e+00       4
## Garfield         -6.2437504   0.000000e+00       1
## Grant            -1.0827330   1.953467e-03      12
## Grays Harbor     -5.1217816   7.069872e-02      10
## Island        -1537.3016492   2.530536e+01       3
## Jefferson         6.4000838  -8.374934e-12       3
## King             -1.7068805   3.304999e-02      64
## Kitsap           -2.2073879   3.764587e-02      10
## Kittitas       -872.4594319   1.417487e+01       3
## Klickitat        -1.3208736   2.261177e-02       4
## Lewis            -2.2577675   2.209851e-02      12
## Lincoln         -65.6033716   8.563292e-01       5
## Mason           -41.6017930   7.033422e-01       5
## Okanogan         -8.9446635   1.356044e-01       7
## Pacific        -176.1340078   2.669124e+00       4
## Pend Oreille   -217.7896667   3.626472e+00       3
## Pierce           -4.4015101   6.368808e-02      27
```

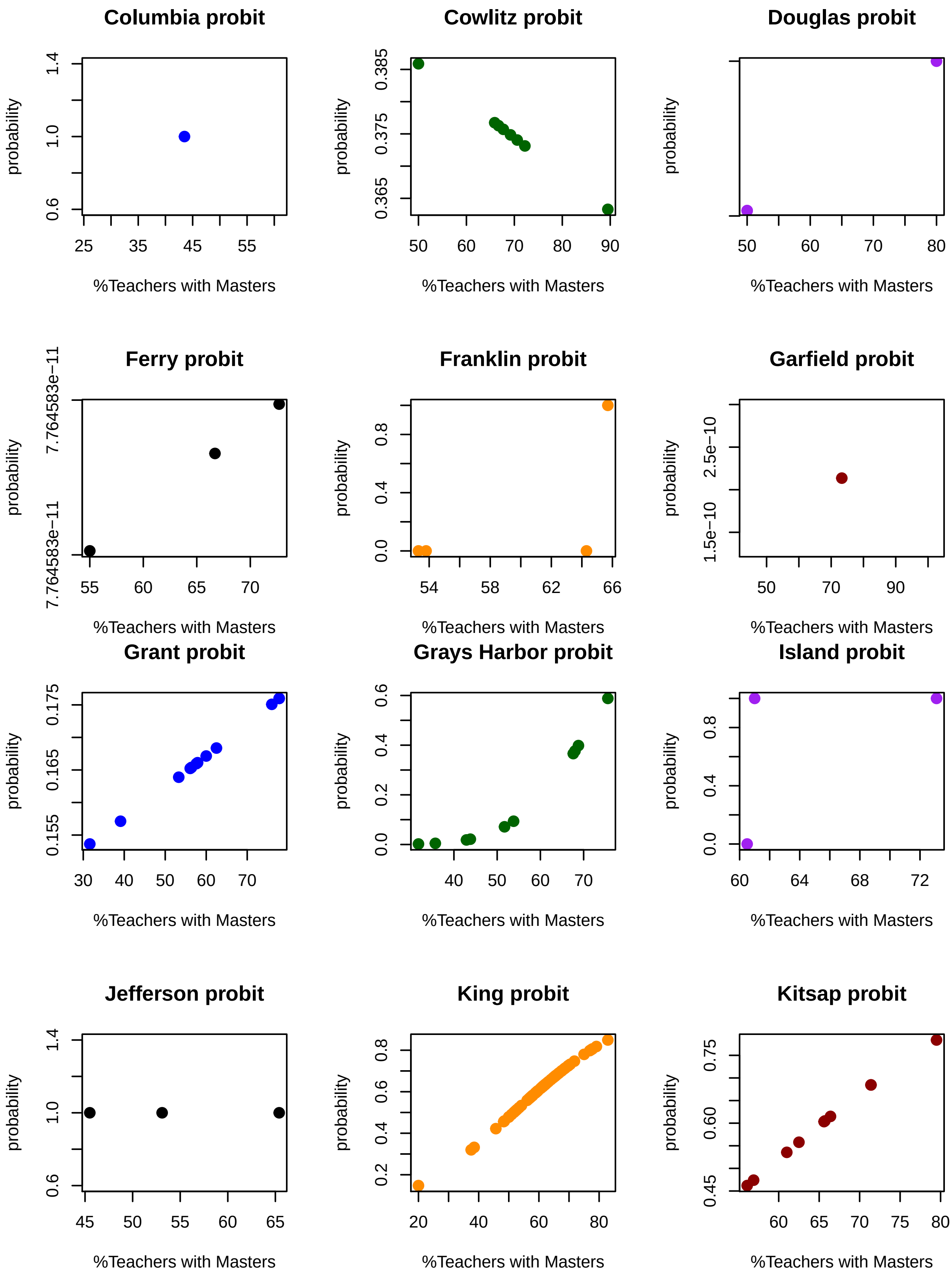
```
## San Juan         6.4000838  -1.177615e-11       3
## Skagit          -1.5076571   2.693985e-02       7
## Skamania        -6.2437504   0.000000e+00       1
## Snohomish       -0.7545658   9.399968e-03      30
## Spokane          1.4811569  -2.288736e-02      27
## Stevens         -1.4429036   2.639003e-02       7
## Thurston        -0.9102153   1.903244e-02      13
## Wahkiakum       -6.2437504   0.000000e+00       1
## Walla Walla     -2.4057579   1.861378e-02       7
## Whatcom         -5.1055123   1.081552e-01      10
## Whitman        252.1465727  -2.870719e+00       9
## Yakima          -2.2740561   1.148984e-02      20
```

```r
i=1
while(i < length(counties)){
  par(mfrow = c(2,3))
  top <- min((i+5),length(counties))
  for(j in i:top){
    countyX1 <- X[data$county == counties[j]];
    plot(countyX1, pro.countyToProbs[[j]], main = paste(counties[j],'probit'),
         xlab = '%Teachers with Masters', ylab = 'probability',
         col = C[j%%6+1], pch = 20, cex = 2)
  }
  i <- i + 6;
}
```

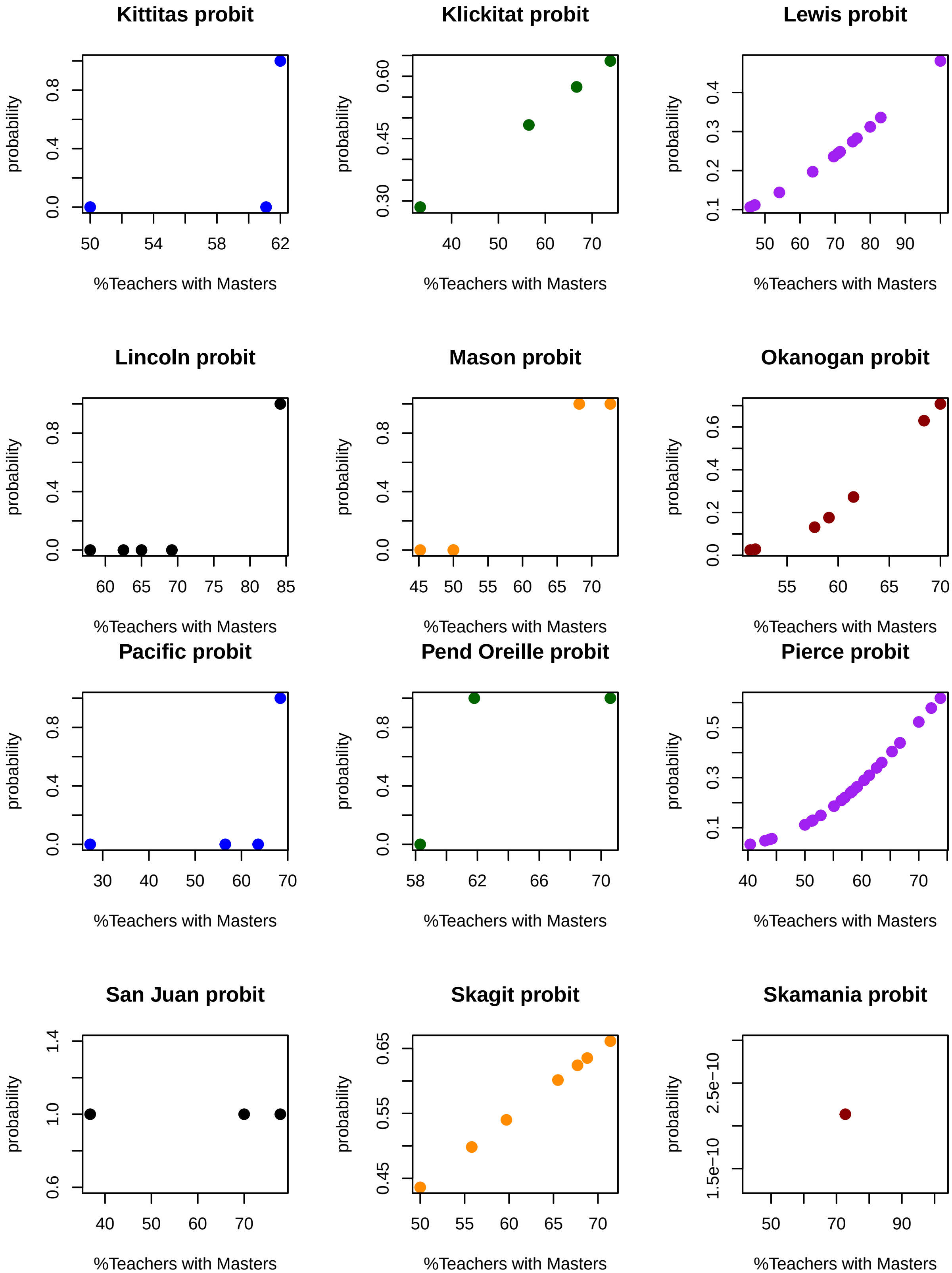


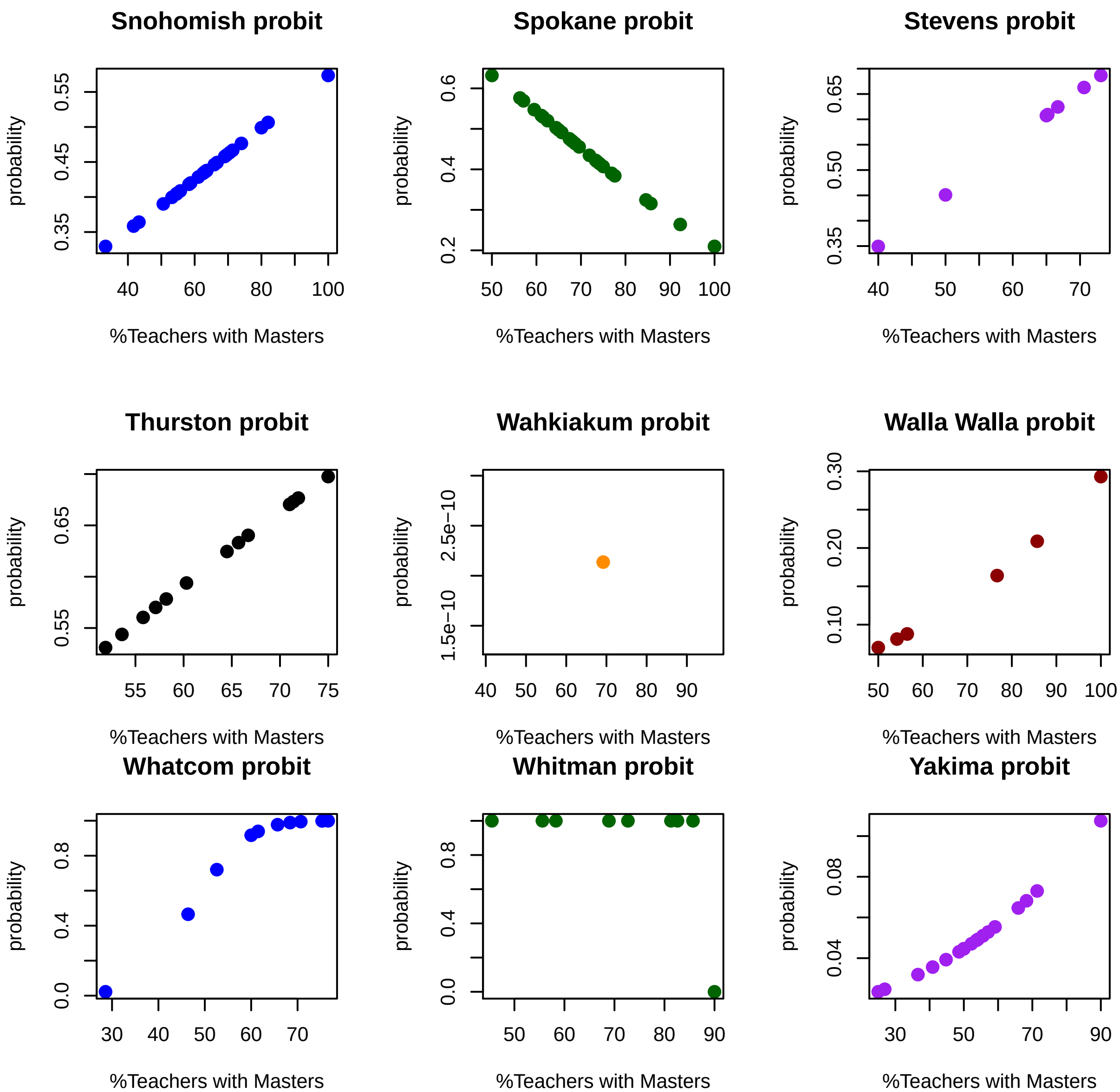

```
## Warning in plot.window(...): relative range of values = 23 * EPS, is small
## (axis 2)

## Warning in plot.window(...): relative range of values = 70 * EPS, is small
```

```
## (axis 2)
```



Columbia probit
Cowlitz probit
Douglas probit
Ferry probit
Franklin probit
Garfield probit
Grant probit
Grays Harbor probit
Island probit
Jefferson probit
King probit
Kitsap probit
probability
%Teachers with Masters
0.6
1.0
1.4
25
35
45
55
0.365
0.375
0.385
50
60
70
80
90
7.764583e−11
7.764583e−11
65
0.0
0.4
0.8
54
58
62
66
1.5e−10
2.5e−10
0.155
0.165
0.175
30
40
0.2
0.6
64
68
72
20
0.45
0.60
0.75
75


Kittitas probit
probability
%Teachers with Masters
Klickitat probit
probability
%Teachers with Masters
Lewis probit
probability
%Teachers with Masters
Lincoln probit
probability
%Teachers with Masters
Mason probit
probability
%Teachers with Masters
Okanogan probit
probability
%Teachers with Masters
Pacific probit
probability
%Teachers with Masters
Pend Oreille probit
probability
%Teachers with Masters
Pierce probit
probability
%Teachers with Masters
San Juan probit
probability
%Teachers with Masters
Skagit probit
probability
%Teachers with Masters
Skamania probit
probability
%Teachers with Masters

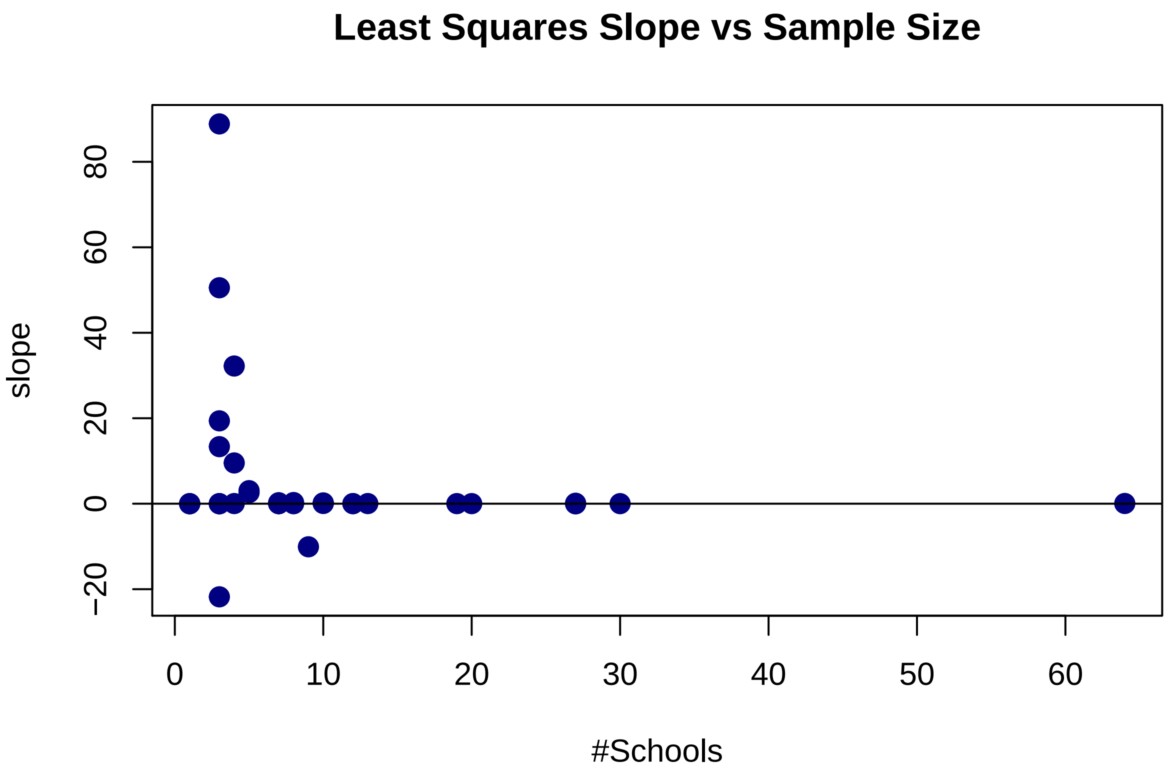


Comments

It appears that some of the counties (Yakima and Clark for example) show positive linear trends between the probability and X_1, while others (Spokane and Cowlitz for example) show negative linear trends, while others (San Juan and Whitman for example) don't show any evident trends in the data. This supports the idea that the relationship between the predictor $X1$ (percent of teachers with a master's degree) and Y (if more than 50 percent of students passed the test) differ from county to county. The probit regression method and the default method yeild similar in terms of the shape of the plots, however the probit regression method yeilds less extreme coefficients. Note the differences in the y axis in the two plots below.

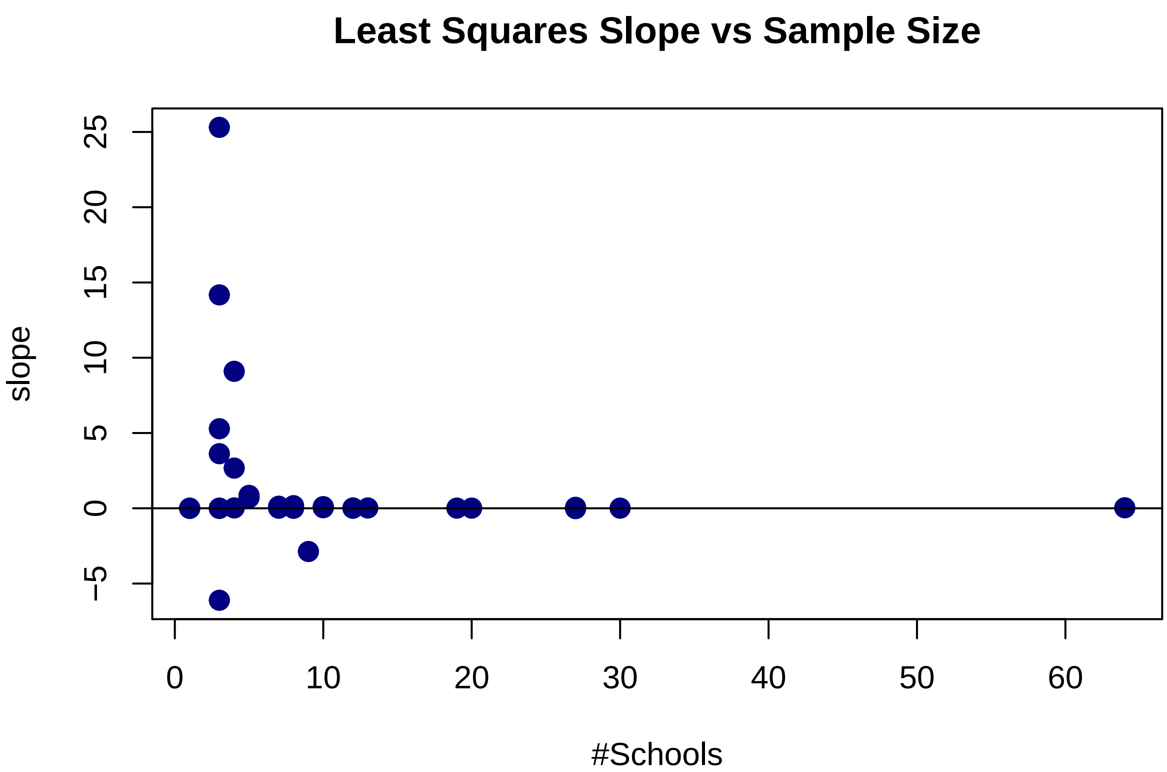
Logisic

```r
plot(BetaEstimate[,3], BetaEstimate[,2], pch = 20, cex = 2, col = 'navy',
     main = "Least Squares Slope vs Sample Size", xlab = '#Schools', ylab = 'slope')
abline(0,0)
```



Probit

```r
plot(pro.BetaEstimate[,3], pro.BetaEstimate[,2], pch = 20, cex = 2, col = 'navy',
     main = "Least Squares Slope vs Sample Size", xlab = '#Schools', ylab = 'slope')
abline(0,0)
```



As can be seen above, schools with more extreme slope value have small sample sizes.

Ad hoc estimates $\hat{\theta}, \hat{\Sigma}$ for θ, Σ can be obtained by taking the average and covariance of $\beta_{j,MLE}$ values which have samples ≥ 10, which is performed below.

Logistic

I'll use this subset

```
moreThan10 <- BetaEstimate[BetaEstimate[,3]>9,]
moreThan10
```

```
##               intercept percentMasters samples
## Clark        -0.9551942    0.018773879      19
## Grant        -1.7867982    0.003004804      12
## Grays Harbor -8.4978525    0.117460555      10
## King         -2.8641607    0.055371915      64
## Kitsap       -3.4675513    0.059263387      10
## Lewis        -3.5101909    0.033729679      12
## Pierce       -7.3545109    0.106468262      27
## Snohomish    -1.2003330    0.014952651      30
## Spokane       2.3035123   -0.035558754      27
## Thurston     -1.5709685    0.032338772      13
## Whatcom      -8.5641958    0.181828364      10
## Yakima       -4.0847169    0.020812918      20
```

```
theta0.hat <- mean(moreThan10[,1]);
theta1.hat <- mean(moreThan10[,2]);
```

```r
theta.hat  <- c(theta0.hat, theta1.hat);
print("Theta.hat")
```

```
## [1] "Theta.hat"
```

```r
theta.hat
```

```
## [1] -3.46274671  0.05070387
```

```r
Sigma.hat <- var(moreThan10[,c(1,2)])
print("Sigma.hat")
```

```
## [1] "Sigma.hat"
```

```r
print(Sigma.hat)
```

```
##                 intercept percentMasters
## intercept      10.755821     -0.1793610
## percentMasters -0.179361      0.0034898
```

Thus with logistic regression

$\hat{\theta} = [-3.46275, 0.05070]$ and $\hat{\Sigma} = \begin{bmatrix} 10.755821 & -0.1793610 \\ -0.179361 & 0.0034898 \end{bmatrix}$

Probit

```r
pro.moreThan10 <- pro.BetaEstimate[pro.BetaEstimate[,3]>9,]
pro.moreThan10
```

```
##               intercept percentMasters samples
## Clark        -0.5159491    0.010511938      19
## Grant        -1.0827330    0.001953467      12
## Grays Harbor -5.1217816    0.070698724      10
## King         -1.7068805    0.033049985      64
## Kitsap       -2.2073879    0.037645873      10
## Lewis        -2.2577675    0.022098506      12
## Pierce       -4.4015101    0.063688080      27
## Snohomish    -0.7545658    0.009399968      30
## Spokane       1.4811569   -0.022887357      27
## Thurston     -0.9102153    0.019032443      13
## Whatcom      -5.1055123    0.108155224      10
## Yakima       -2.2740561    0.011489838      20
```

```r
theta0.hat <- mean(pro.moreThan10[,1]);
theta1.hat <- mean(pro.moreThan10[,2]);
pro.theta.hat  <- c(theta0.hat, theta1.hat);
print("Pro.Theta.hat")
```

```
## [1] "Pro.Theta.hat"
```

```r
pro.theta.hat
```

```
## [1] -2.07143352  0.03040306
```

```r
pro.Sigma.hat <- var(pro.moreThan10[,c(1,2)])
print("pro.Sigma.hat")
```

```
## [1] "pro.Sigma.hat"
```

```r
print(pro.Sigma.hat)
```

```
##                intercept percentMasters
## intercept      3.93338187   -0.065478933
## percentMasters -0.06547893    0.001261682
```

Part c

With priors of

$$\theta \sim MVN(\hat{\theta}, \hat{\Sigma}), \quad \Sigma^{-1} \sim Wishart(4, \hat{\Sigma}^{-1})$$

their full conditional distributions are simply

$$(\theta \mid \dots) \sim MVN\Big((\hat{\Sigma}^{-1} + m\Sigma^{-1})^{-1}(\hat{\Sigma}^{-1}\hat{\theta} + m\Sigma^{-1}\bar{\beta}), (\hat{\Sigma}^{-1} + m\Sigma^{-1})^{-1}\Big), \quad \bar{\beta} := \frac{1}{m}\sum_{j}^{m}\beta_j$$

$$\Sigma \sim Wishart\Big(4 + m, \big[\hat{\Sigma} + S_\theta\big]^{-1}\Big), \quad S_\theta := \sum_{j}^{m}(\beta_j - \theta)(\beta_j - \theta)^T$$

and for each β_j, $\beta_j^{(s+1)}$ can be obtained from $\beta_j^{(s)}$ and $\Sigma^{(s+1)}$ through proposing new values with

$$\beta_j^{*} \sim MVN(\beta_j^{(s)}, \Sigma^{(s+1)})$$

which makes the proposals symmetric, while

$$p(y_j \mid \beta_j^{(s)}, x_j) = \prod_{i}^{n_j} p_{i,j,(s)}^{y_{i,j}}(1 - p_{i,j,(s)})^{1-y_{i,j}}$$

where

$$p_{i,j,s} = \frac{e^{\beta_j^{(s)T}x_{i,j}}}{1 + e^{\beta_j^{(s)T}x_{i,j}}}$$

Thus, the acceptance probability r can be calculated as

$$p_{j,*} = \frac{exp\{\beta_{i,j}^{(*)T}x_{i,j}\}}{1 + exp\{\beta_{i,j}^{(*)T}x_{i,j}\}}$$

$$p_{j,s} = \frac{exp\{\beta_{j}^{(s)T}x_{i,j}\}}{1 + exp\{\beta_{j}^{(s)T}x_{i,j}\}}$$

$$r = \min\left(1, \frac{\prod_i^{n_j} \text{dbern}(y_{i,j}, p_{i,j,*})}{\prod_i^{n_j} \text{dbern}(y_{i,j}, p_{i,j,(s)})}\right), \quad \log(r) = \min\left(0, \sum_i^{n_j} \log(\text{dbern}(y_{i,j}, p_{i,j,*})) - \sum_i^{n_j} \log(\text{dbern}(y_{i,j}, p_{i,j,(s)}))\right)$$

since

$$\log(\text{dbern}(y,p)) = \log\left(p^y(1-p)^{1-y}\right) = y\log(p) + (1-y)\log(1-p) = y\log\left(\frac{e^{\beta^T x}}{1+e^{\beta^T x}}\right) + (1-y)\log\left(\frac{1}{1+e^{\beta^T x}}\right)$$

$$= y\log\left(e^{\beta^T x}\right) - y\log\left(1+e^{\beta^T x}\right) + (1-y)\log\left(1\right) - (1-y)\log\left(1+e^{\beta^T x}\right)$$

$$= y\beta^T x - y\log(1+e^{\beta^T x}) - \log\left(1+e^{\beta^T x}\right) + y\log\left(1+e^{\beta^T x}\right) = y\beta^T x - \log(1+e^{\beta^T x})$$

$$\Rightarrow r = \min\left(0, \sum_i^{n_j}\left(y_{i,j}\beta_{j,*}^T x_{i,j} - \log(1+e^{\beta_{j,*}^T x_{i,j}}) + \log \text{dMVN}(\beta_{j,*}, \theta^{(s+1)}, \Sigma^{(s+1)})\right) - \sum_i^{n_j}\left(y_{i,j}\beta_{j,(s)}^T x_{i,j} - \log(1+e^{\beta_{j,(s)}^T x_{i,j}}) + \log \text{d}\right.\right.$$

The Metropolis-Hastings Algorithm

Functions

```r
logLikeBeta <- function(Betas, x, y) {
  z <- x*(Betas[2]) + Betas[1]
  p <- exp(z)/(1+exp(z))
  probs<- dbinom(y,1,p)
  return (sum(log(probs)))
}

logLikeProbit <- function(Betas, x, y) {
  z <- x*(Betas[2]) + Betas[1]
  p <- pnorm(z)
  probs<- dbinom(y,1,p)
  return (sum(log(probs)))
}

sigmaToVec <- function(Sigma){
  a <- Sigma[1,1];
  b <- Sigma[1,2];
  c <- Sigma[2,2];
  return(c(a,b,c));
}
vecToSigma <- function(sigmaVec){
  Sigma <- matrix(c(sigmaVec[1],sigmaVec[2],
                    sigmaVec[2],sigmaVec[3]),
                  nrow = 2, ncol = 2, byrow = T)
  return(Sigma);
}
```

```r
library(MASS)
library(mvtnorm)
library(coda)
```

```r
ARBetas <- rep(0,length(counties));
set.seed(1)
#Data Structures
Samples <- 5000; m <- length(counties); thinning <- 1;
SAMPLES <- Samples*thinning;
BETAS   <- list();
SIGMA   <- matrix(0, nrow = 1, ncol  = 3);
THETA   <- matrix(0, nrow = 1, ncol = 2);
#Initial Values
for(j in 1:m){
  BETAS[[j]] <- matrix(0, nrow = 1, ncol = 2);
}
currentBetas  <- BetaEstimate[,c(1,2)]*0;
currentBetas[,1] <- theta.hat[1]
currentBetas[,2] <- theta.hat[2]
SIGMA[1,]     <- sigmaToVec(Sigma.hat);
currentSigma  <- Sigma.hat;
THETA[1,]     <- theta.hat;
currentTheta  <- theta.hat;
Sigma.hat.inv <- solve(Sigma.hat)

pro.BETAS   <- list();
pro.SIGMA   <- matrix(0, nrow = 1, ncol  = 3);
pro.THETA   <- matrix(0, nrow = 1, ncol = 2);
#Initial Values
for(j in 1:m){
  pro.BETAS[[j]] <- matrix(0, nrow = 1, ncol = 2);
}
pro.currentBetas  <- pro.BetaEstimate[,c(1,2)]*0;
pro.currentBetas[,1] <- pro.theta.hat[1]
pro.currentBetas[,2] <- pro.theta.hat[2]
pro.SIGMA[1,]     <- sigmaToVec(pro.Sigma.hat);
pro.currentSigma  <- pro.Sigma.hat;
pro.THETA[1,]     <- pro.theta.hat;
pro.currentTheta  <- pro.theta.hat;
pro.Sigma.hat.inv <- solve(pro.Sigma.hat)


for(i in 2:SAMPLES){
  #Update theta
  sigma.s.inv <- solve(currentSigma);
  #pro.sigma.s.inv <- solve(pro.currentSigma);
  beta.mean   <- colSums(currentBetas)/m;
  #pro.beta.mean   <- colSums(pro.currentBetas)/m;
  t.var       <- solve(Sigma.hat.inv + m*sigma.s.inv);
  #pro.t.var       <- solve(pro.Sigma.hat.inv + m*pro.sigma.s.inv);
  t.mean      <- t.var%*%(Sigma.hat.inv%*%theta.hat + m*sigma.s.inv%*%beta.mean);
  #pro.t.mean      <- pro.t.var%*%(pro.Sigma.hat.inv%*%pro.theta.hat + m*pro.sigma.s.inv%*%pro.beta.mea
  theta.new   <- mvrnorm(1, t.mean, t.var);
  #pro.theta.new   <- mvrnorm(1, pro.t.mean, pro.t.var);
  currentTheta<- theta.new;
  #pro.currentTheta<- pro.theta.new;
  #Update Sigma
```

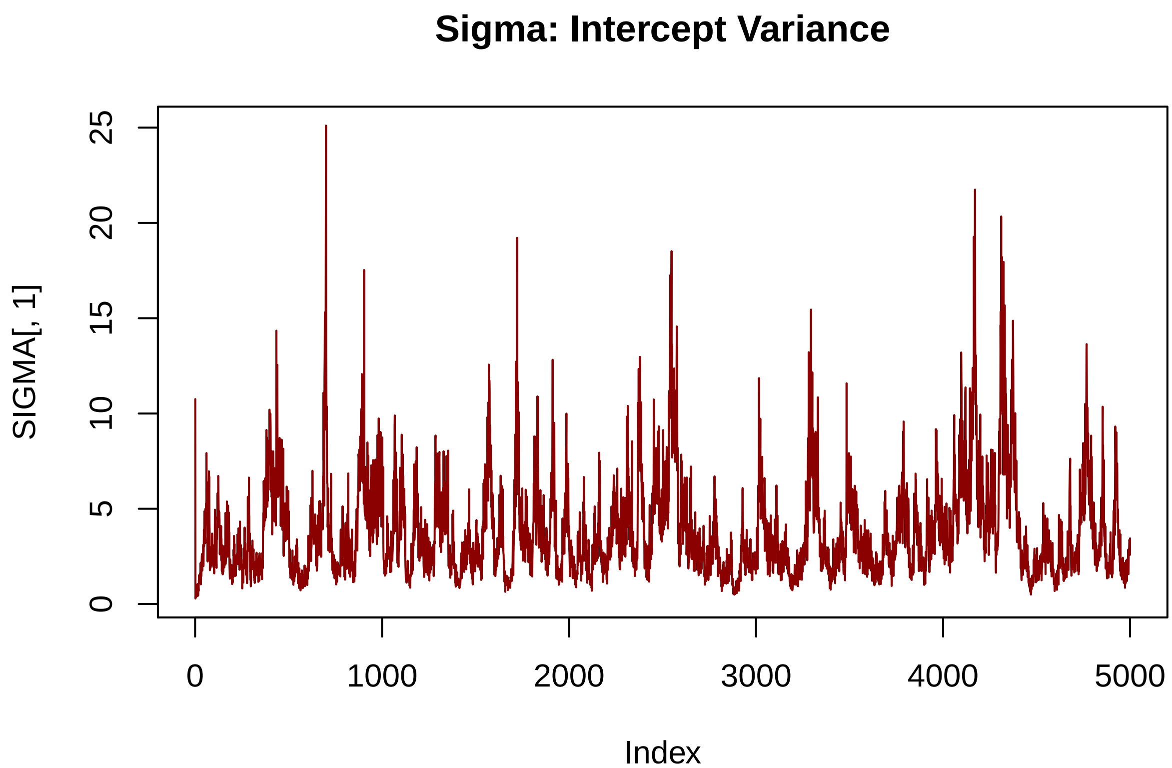
```
  S.theta <- matrix(0, nrow = 2, ncol =2)
  #pro.S.theta <- matrix(0, nrow = 2, ncol =2)
  for(k in 1:length(counties)){
    S.theta <- S.theta + (currentBetas[k,]-currentTheta)%*%t(currentBetas[k,]-currentTheta);
    #pro.S.theta <- pro.S.theta + (pro.currentBetas[k,]-pro.currentTheta)%*%t(pro.currentBetas[k,]-pro.
  }
  Sigma.new.inv <- rWishart(1, 4+m, solve(Sigma.hat + S.theta))[,,1];
  #pro.Sigma.new.inv <- rWishart(1, 4+m, solve(pro.Sigma.hat + pro.S.theta))[,,1];
  Sigma.new     <- solve(Sigma.new.inv);
  #pro.Sigma.new     <- solve(pro.Sigma.new.inv);
  currentSigma  <- Sigma.new;
  #pro.currentSigma  <- pro.Sigma.new;
  for(j in 1:m){ #We can itterate through these individually.
    countyX  <- countyToX[[j]];
    countyY  <- countyToY[[j]];
    beta.s   <- currentBetas[j,];
    beta.pro <- mvrnorm(1, beta.s, 2*currentSigma);
    logP.s   <- logLikeBeta(beta.s, countyX, countyY);
    logP.pro <- logLikeBeta(beta.pro, countyX, countyY);
    prior.pro<- log(dmvnorm(beta.pro,currentTheta, currentSigma));
    prior.s  <- log(dmvnorm(beta.s,currentTheta, currentSigma));
    logDiff  <- logP.pro - logP.s + prior.pro - prior.s;
    logU     <- log(runif(1,0,1));
    accept   <- (logDiff >= logU);
    if(accept){ARBetas[j] <- ARBetas[j]+1};
    currentBetas[j,] <- beta.pro*as.vector(accept) + beta.s*as.vector((!accept))
  }
  #for(j in 1:m){ #We can itterate through these individually.
  #  countyX  <- countyToX[[j]];
  #  countyY  <- countyToY[[j]];
  #  pro.beta.s   <- pro.currentBetas[j,];
  #  pro.beta.pro <- mvrnorm(1, pro.beta.s, pro.currentSigma);
  #  pro.logP.s   <- logLikeProbit(pro.beta.s, countyX, countyY);
  #  pro.logP.pro <- logLikeProbit(pro.beta.pro, countyX, countyY);
  #  pro.prior.pro<- log(dmvnorm(pro.beta.pro, pro.currentTheta, pro.currentSigma));
  #  pro.prior.s  <- log(dmvnorm(pro.beta.s, pro.currentTheta, pro.currentSigma));
  #  pro.logDiff  <- logP.pro - pro.logP.s + pro.prior.pro - pro.prior.s;
  #  pro.logU     <- log(runif(1,0,1));
  #  pro.accept   <- (pro.logDiff >= pro.logU);
  #  pro.currentBetas[j,] <- pro.beta.pro*as.vector(pro.accept) + pro.beta.s*as.vector((!pro.accept))
  #}
  SIGMA  <- rbind(SIGMA,sigmaToVec(Sigma.new));
  THETA  <- rbind(THETA,theta.new);
  for(j in 1:m){
    BETAS[[j]] <- rbind(BETAS[[j]],currentBetas[j,]);
  }
  #pro.SIGMA  <- rbind(pro.SIGMA,sigmaToVec(pro.Sigma.new));
  #pro.THETA  <- rbind(pro.THETA, pro.theta.new);
  #for(j in 1:m){
  #  pro.BETAS[[j]] <- rbind(pro.BETAS[[j]],pro.currentBetas[j,]);
  #}
}
print("done")
```

```
## [1] "done"
```

Part d

Logistic

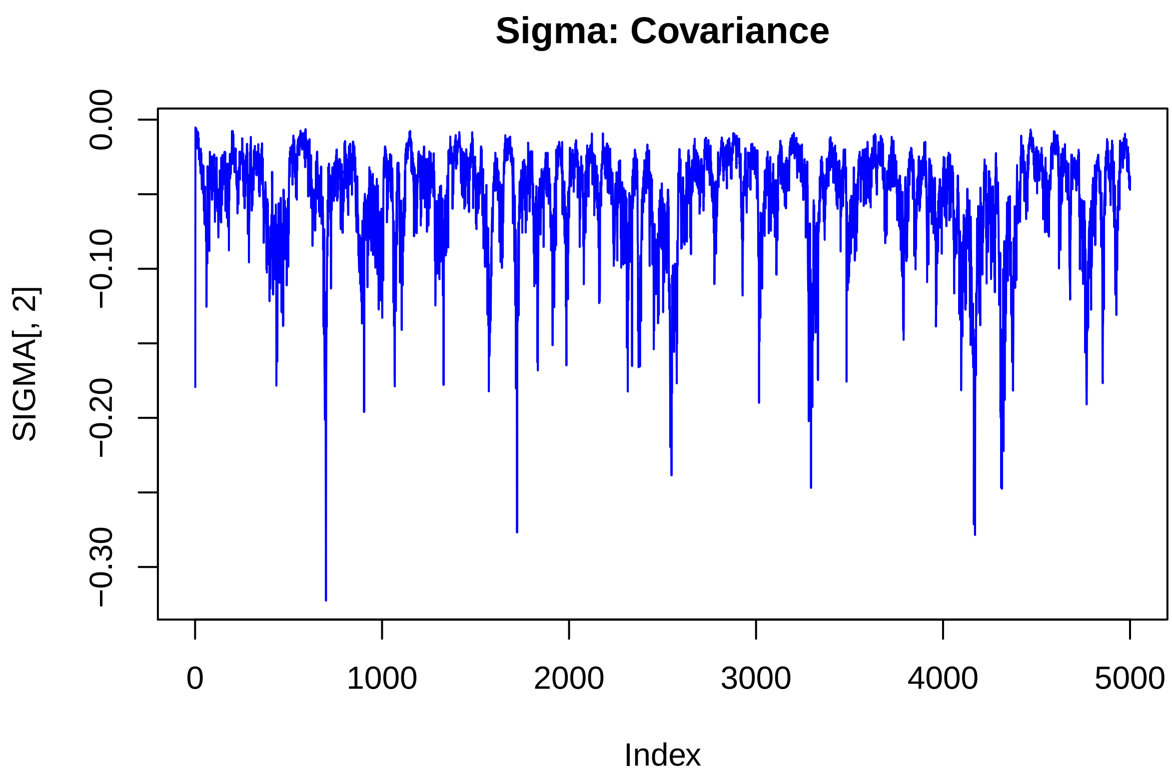
```r
plot(SIGMA[,1], type = 'l',col = C[1], main = 'Sigma: Intercept Variance')
```




```r
effectiveSize(SIGMA[,1])
```

```
##     var1
## 118.8534
```

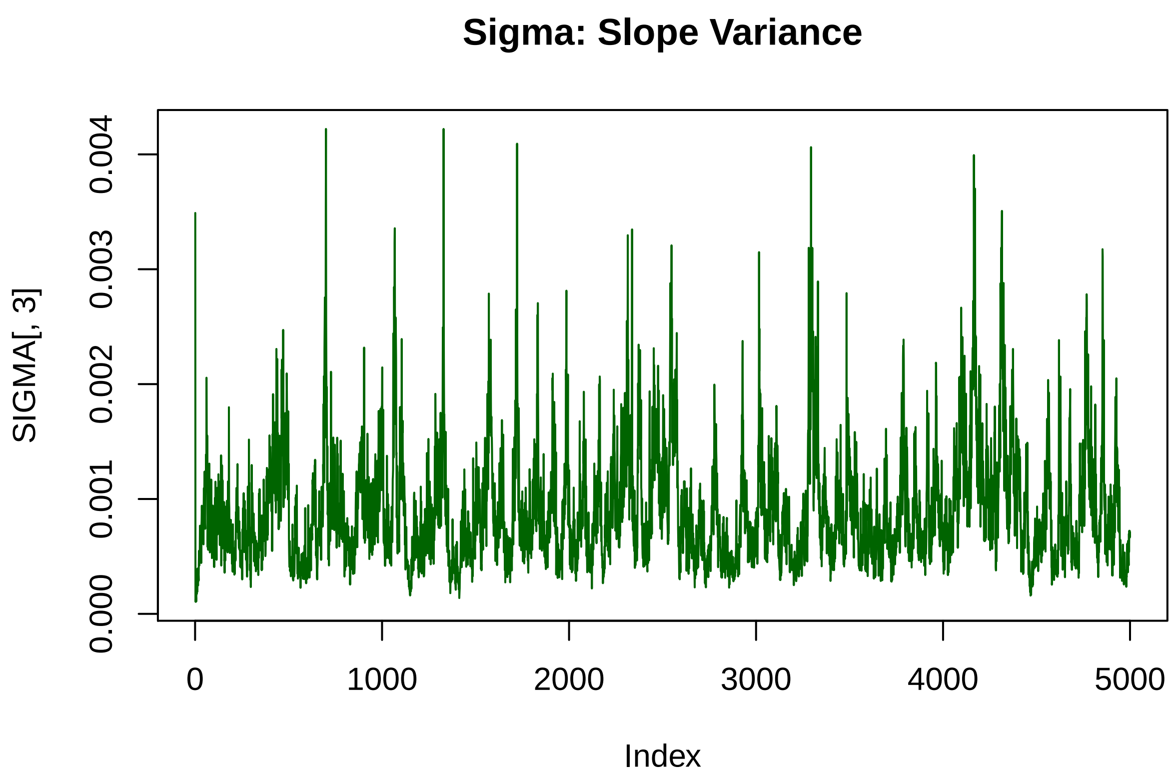
```r
plot(SIGMA[,2], type = 'l',col = C[2], main = 'Sigma: Covariance')
```




```r
effectiveSize(SIGMA[,2])
```

```
##     var1
## 147.0702
```

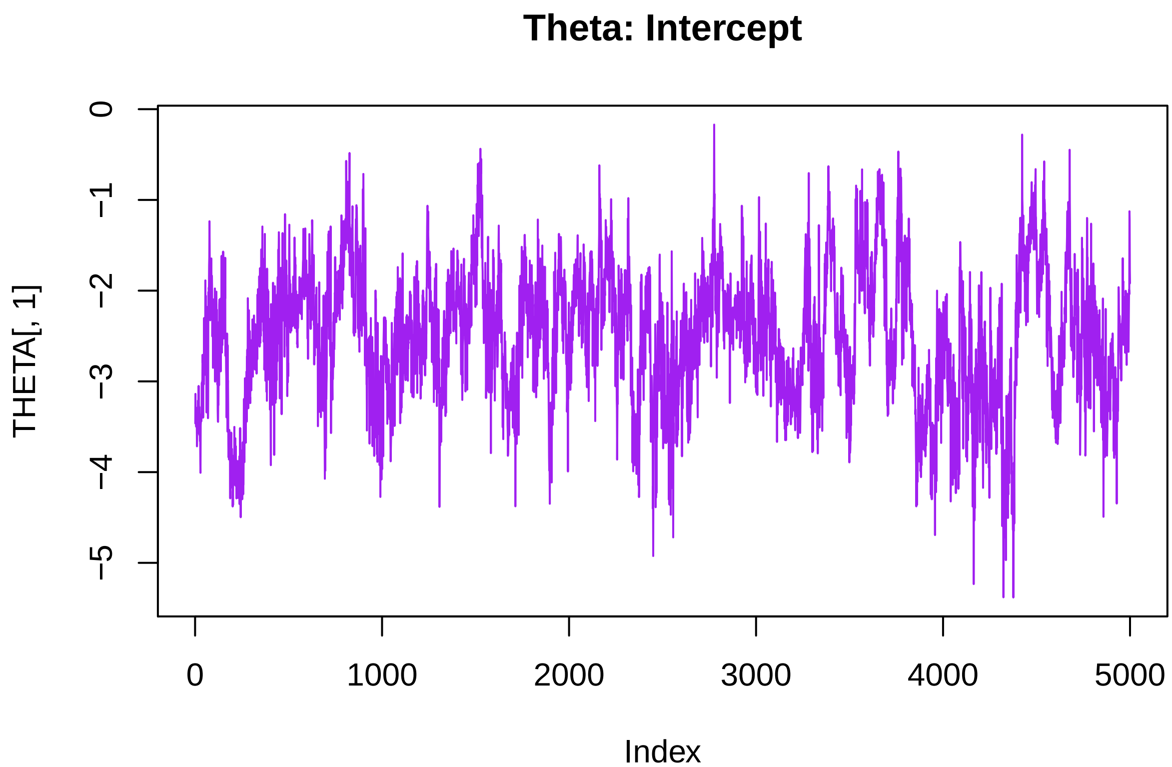
```r
plot(SIGMA[,3], type = 'l',col = C[3], main = 'Sigma: Slope Variance')
```



```r
effectiveSize(SIGMA[,3])
```

```
##     var1
## 179.2111
```

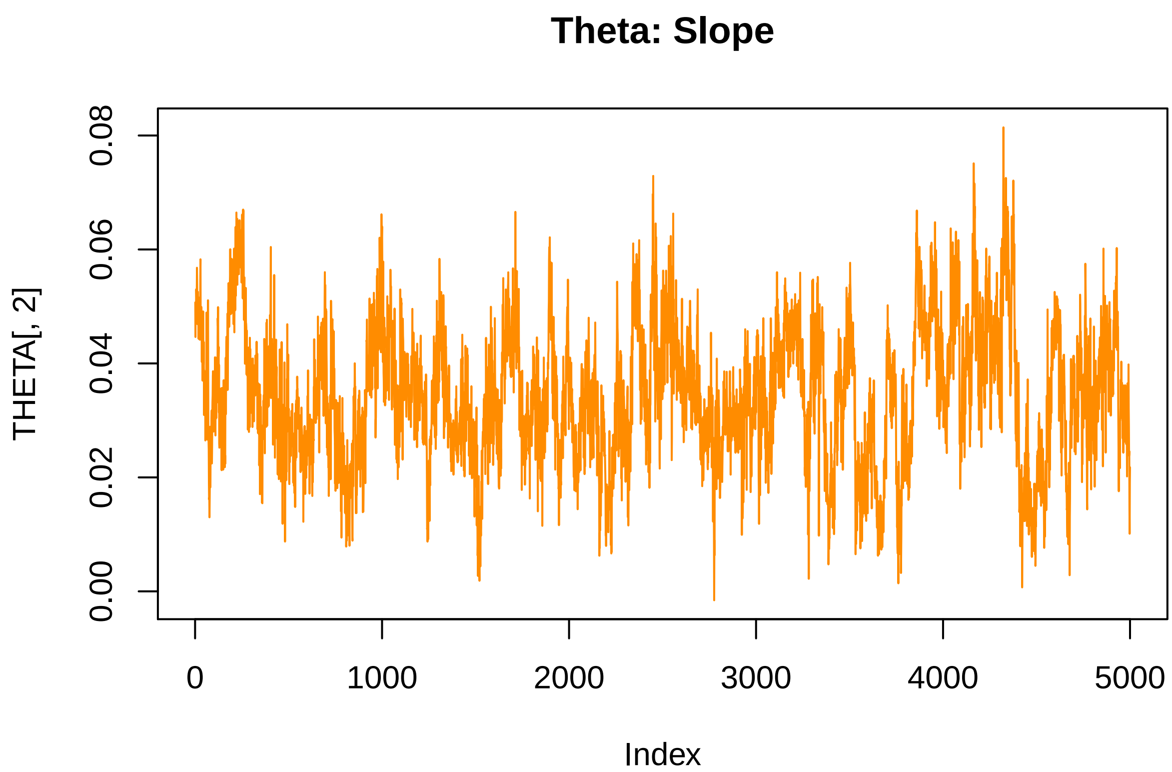
```r
plot(THETA[,1], type = 'l',col = C[4], main = 'Theta: Intercept')
```




```r
effectiveSize(THETA[,1])
```

```
##     var1
## 78.23242
```

```r
plot(THETA[,2], type = 'l',col = C[6], main = 'Theta: Slope')
```




```r
effectiveSize(THETA[,2])
```

```
##     var1
## 79.27545
```

Part e

Posterior expectation of β

```r
postBeta <- matrix(0, nrow = length(counties), ncol = 2)
colnames(postBeta)  <- c("Beta_0,j","Beta_1,j")
colN     <- c();
for(j in 1:length(counties)){
  countyB <- BETAS[[j]];
  b0 <- countyB[,1]; b1 <- countyB[,2];
  postBeta[j,1] <- mean(b0); postBeta[j,2] <- mean(b1);
  colN <- c(colN,paste("Beta_", j, sep = ""));
}
rownames(postBeta) <- colN;
postBeta
```

```
##           Beta_0,j    Beta_1,j
## Beta_1  -1.0484037 0.016926857
## Beta_2  -2.9074281 0.038735088
## Beta_3  -3.0102136 0.044392673
## Beta_4  -3.3422520 0.038031387
## Beta_5  -2.3078772 0.027276412
## Beta_6  -2.2636841 0.036215773
```

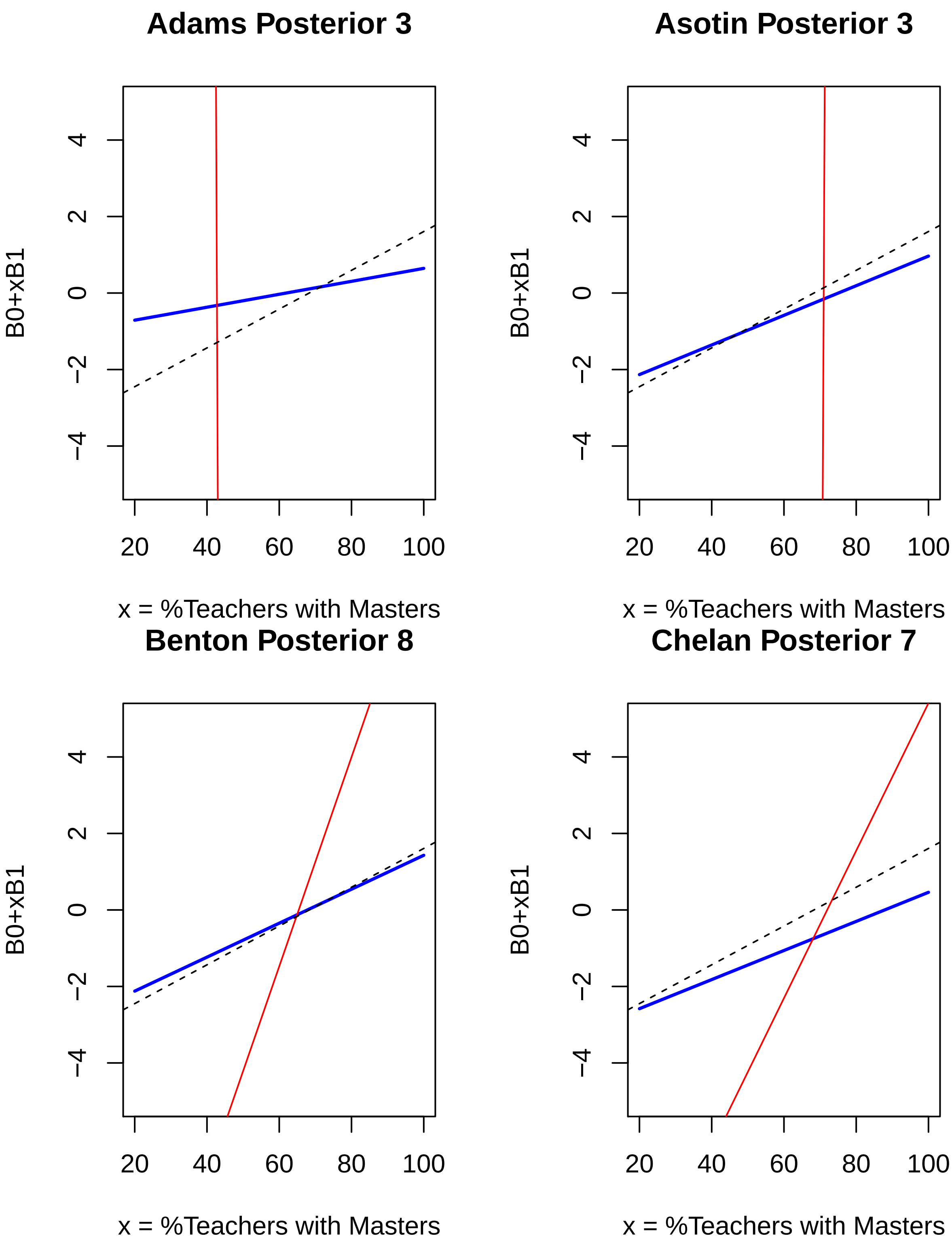
```
## Beta_7  -1.5791156 0.025518948
## Beta_8  -2.2242517 0.026635791
## Beta_9  -2.7643559 0.028749876
## Beta_10 -2.7874777 0.028414655
## Beta_11 -2.9538822 0.037688558
## Beta_12 -2.4574342 0.028188769
## Beta_13 -2.7425245 0.026750786
## Beta_14 -3.6129112 0.045513634
## Beta_15 -2.5265783 0.039782341
## Beta_16 -1.3100116 0.028870461
## Beta_17 -2.3736204 0.045862134
## Beta_18 -2.3442052 0.038229892
## Beta_19 -2.8292338 0.038602909
## Beta_20 -2.3325947 0.035034805
## Beta_21 -2.8530271 0.029836748
## Beta_22 -3.3942032 0.039658661
## Beta_23 -3.3930415 0.050240687
## Beta_24 -3.0642483 0.039623537
## Beta_25 -3.1601721 0.041968792
## Beta_26 -2.5949183 0.041021131
## Beta_27 -3.6628984 0.046423184
## Beta_28 -1.5925618 0.032224107
## Beta_29 -2.2757931 0.036158095
## Beta_30 -2.6288759 0.030252155
## Beta_31 -2.0230707 0.027712427
## Beta_32 -0.7788214 0.006995258
## Beta_33 -2.3404848 0.037412155
## Beta_34 -2.3391851 0.040412685
## Beta_35 -2.5841489 0.030790579
## Beta_36 -2.5524771 0.023082119
## Beta_37 -2.5256009 0.052572245
## Beta_38 -1.0116718 0.028004027
## Beta_39 -3.4545094 0.029324757
```

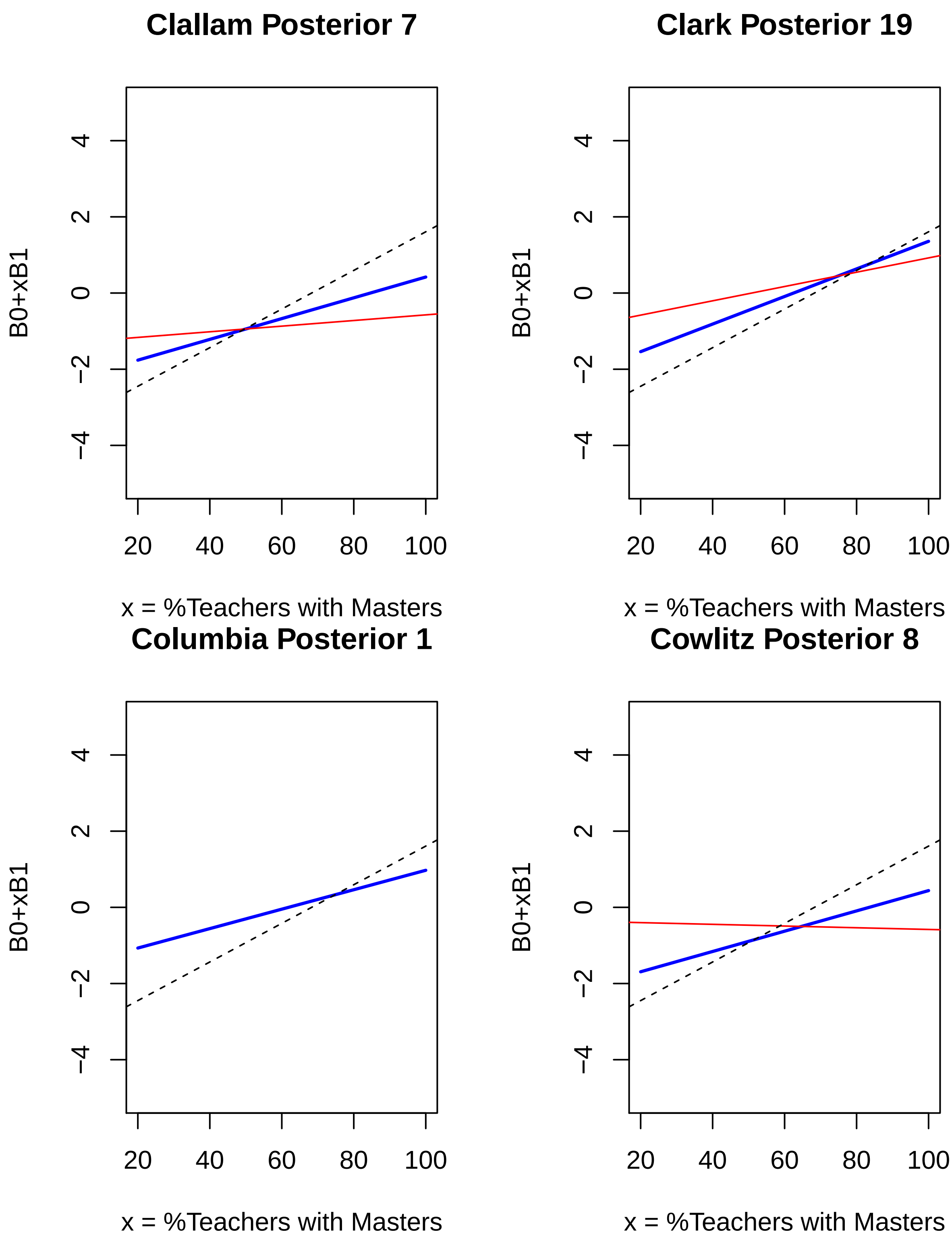
Regression Plot Comparisons

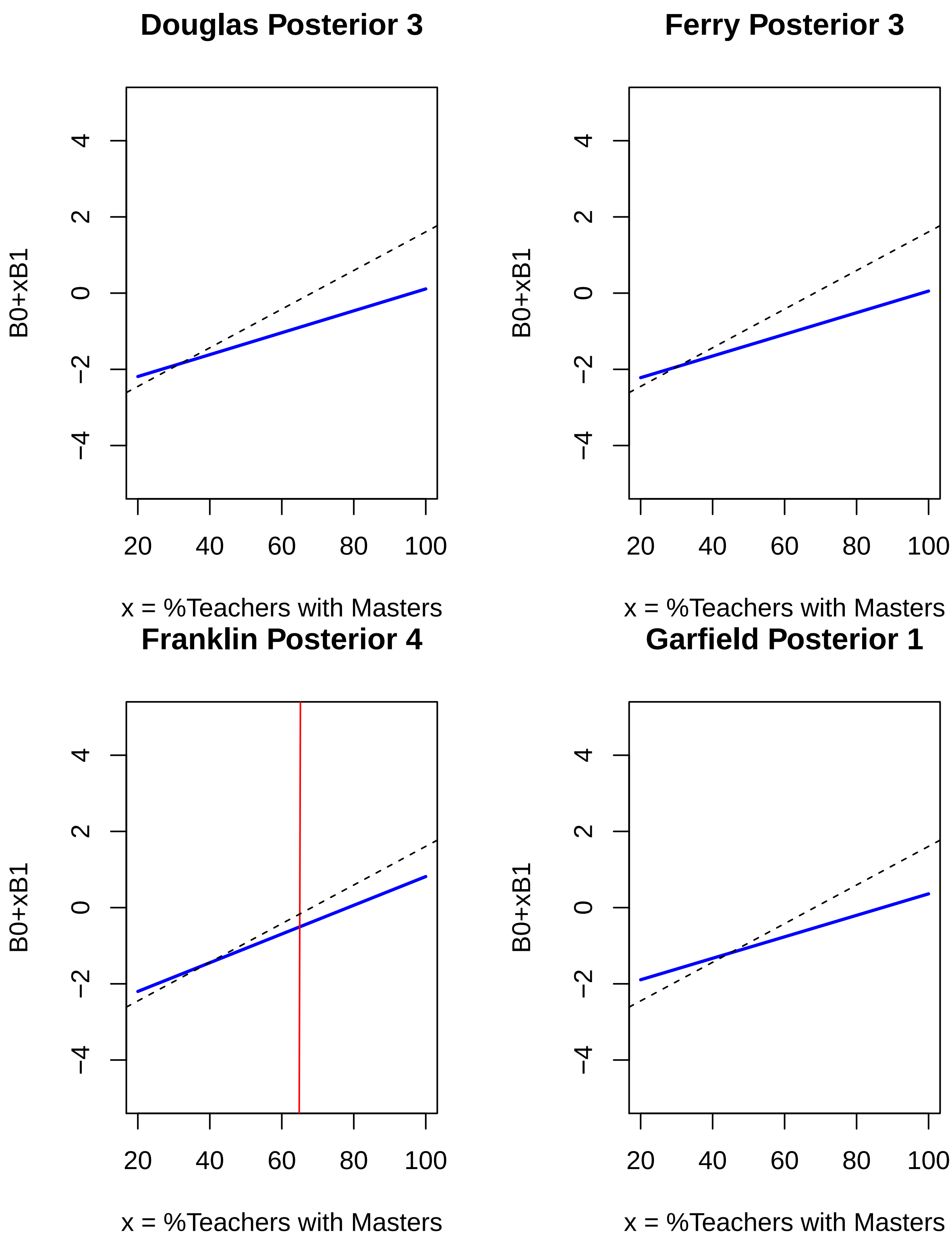
In the plots below, the red line represents the value of $\beta_{0,j} + \beta_{1,j}$ from the estiamtes found in part b, while the blue line represents the posteior estimates of these values. Plotted in black also is my value of $\hat{\theta}$, which was used as the prior mean of each θ. The lines are plotted over the range of 20 to 100, which is the range of values observed in the dataset. Plotted in the titles of each of these pots is an number which represents the number of schools sampled in this county.

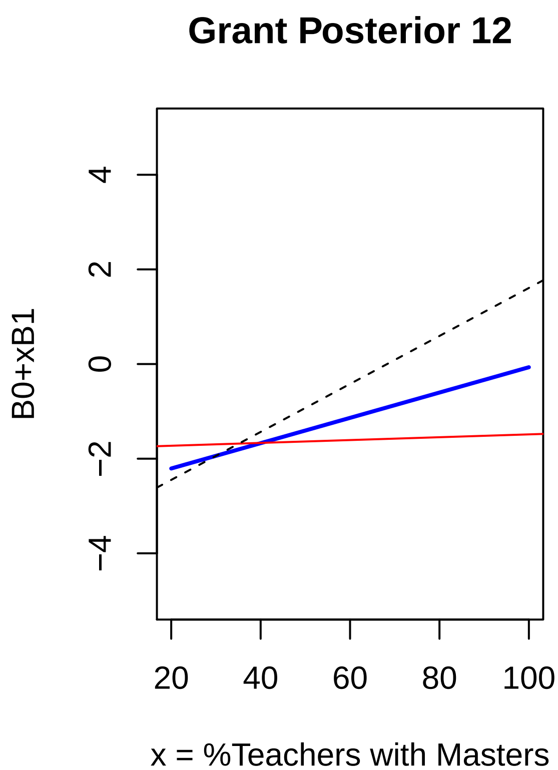
```
i=1
while(i < length(counties)){
  par(mfrow = c(1,2))
  top <- min((i+5),length(counties))
  for(j in i:top){
    countyX1 <- 20:100
    plot(countyX1, countyX1*postBeta[j,2]+postBeta[j,1], main = paste(counties[j], "Posterior", BetaEst
         xlab = 'x = %Teachers with Masters', ylab = 'B0+xB1',
         col = 'blue', type = 'l', ylim = c(-5,5), lwd = 2)
    abline(BetaEstimate[j,1], BetaEstimate[j,2], col = 'red')
```

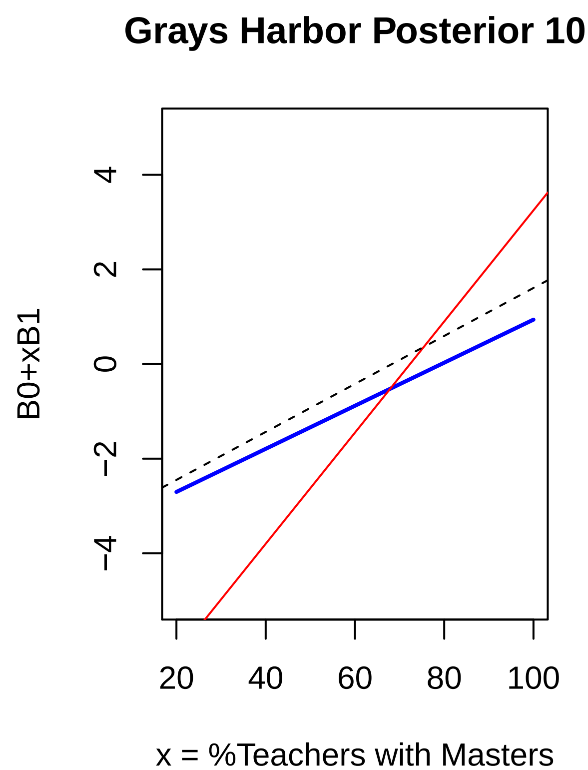
```
    abline(theta.hat[1], theta.hat[2], lty =2)
  }
  i <- i + 6;
}
```

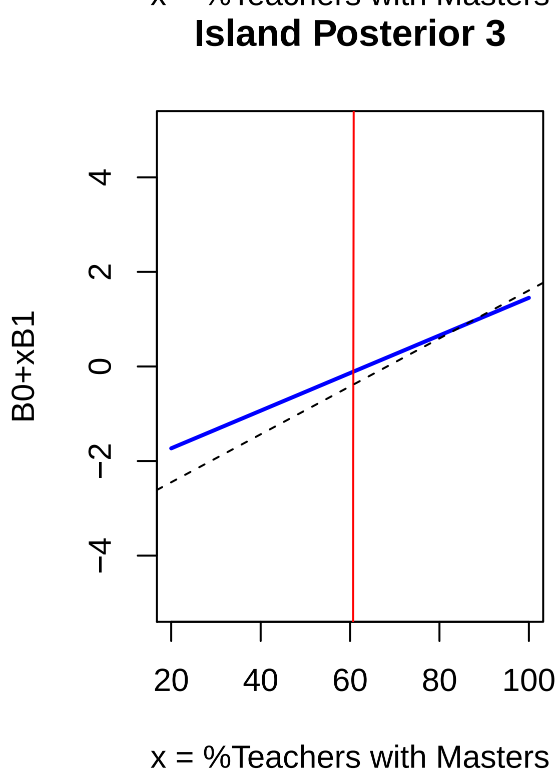


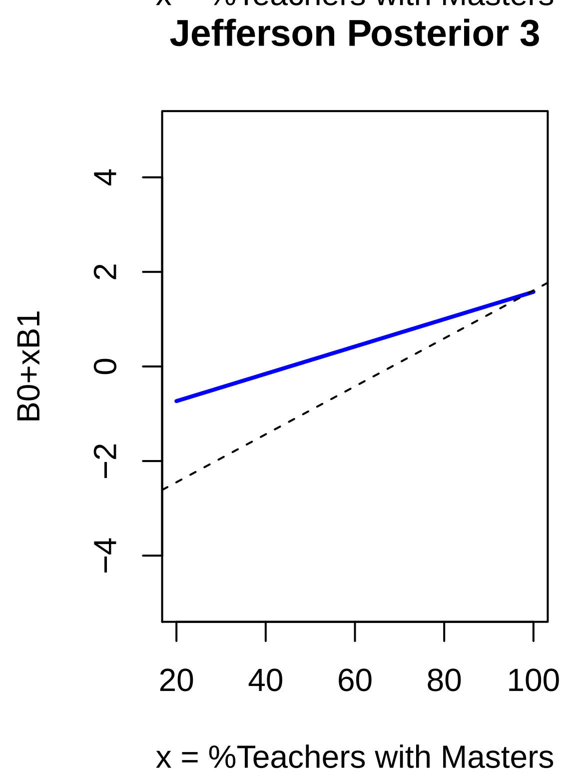

Clallam Posterior 7
Clark Posterior 19
Columbia Posterior 1
Cowlitz Posterior 8
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters

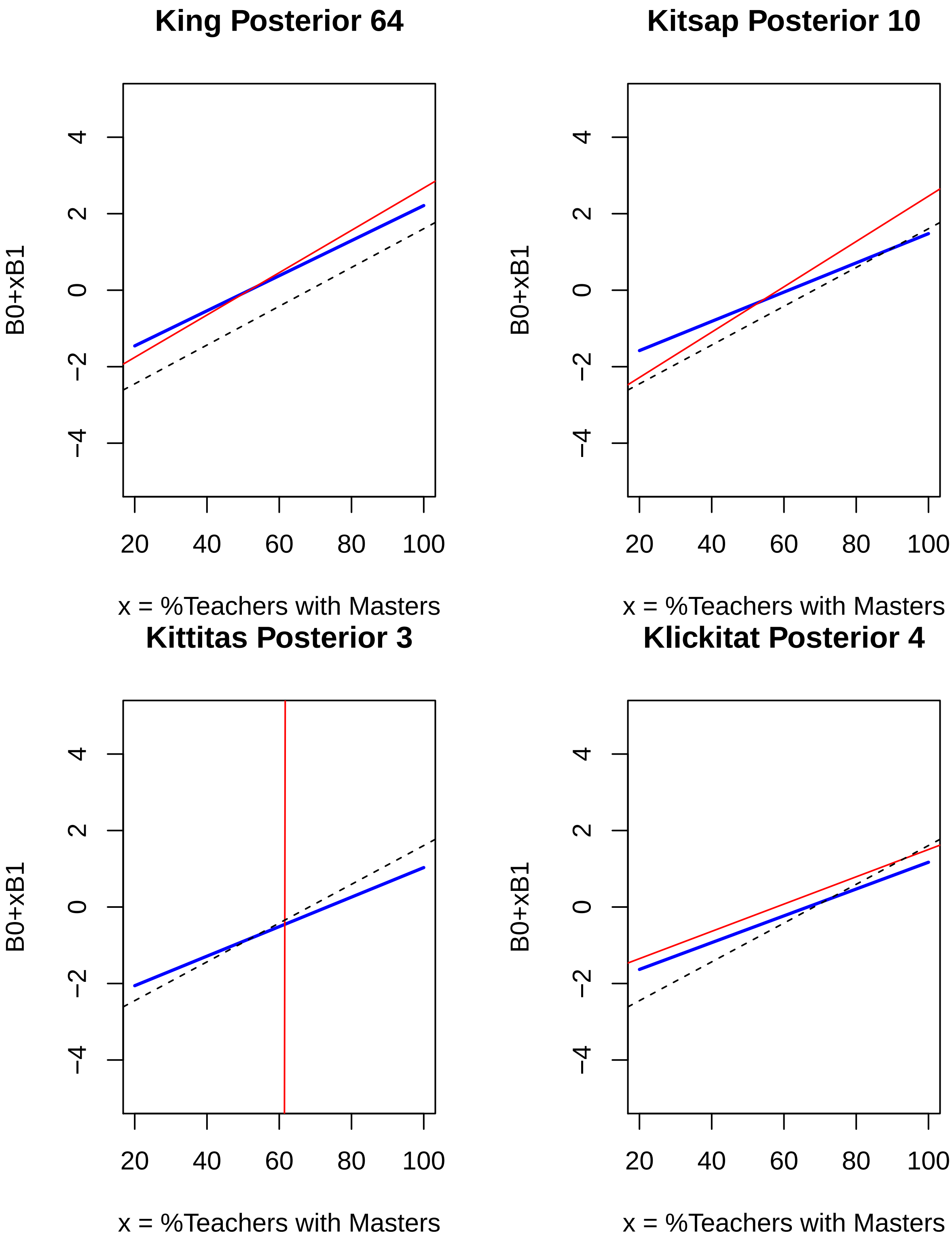

Douglas Posterior 3
Ferry Posterior 3
Franklin Posterior 4
Garfield Posterior 1
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters

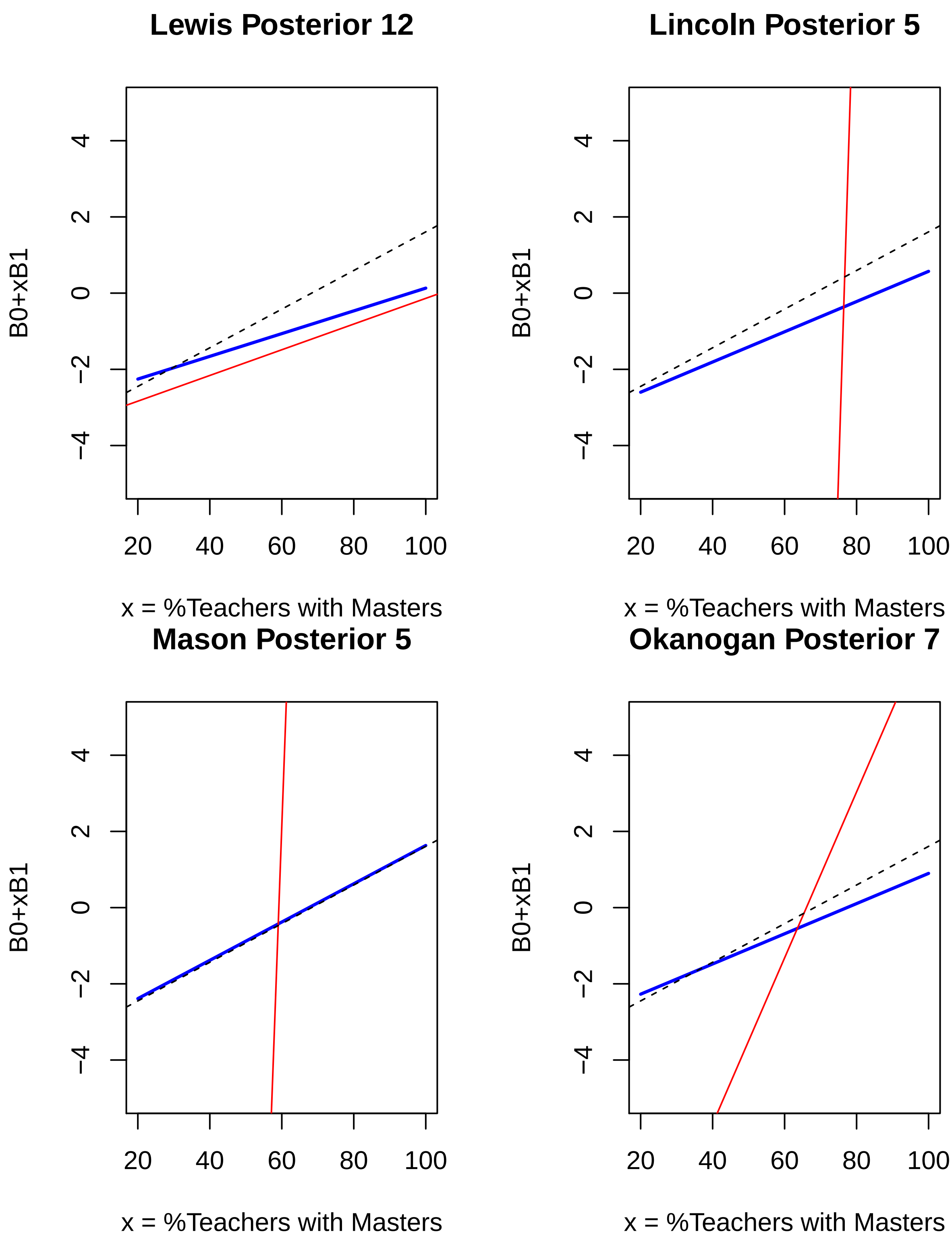

Grant Posterior 12
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters

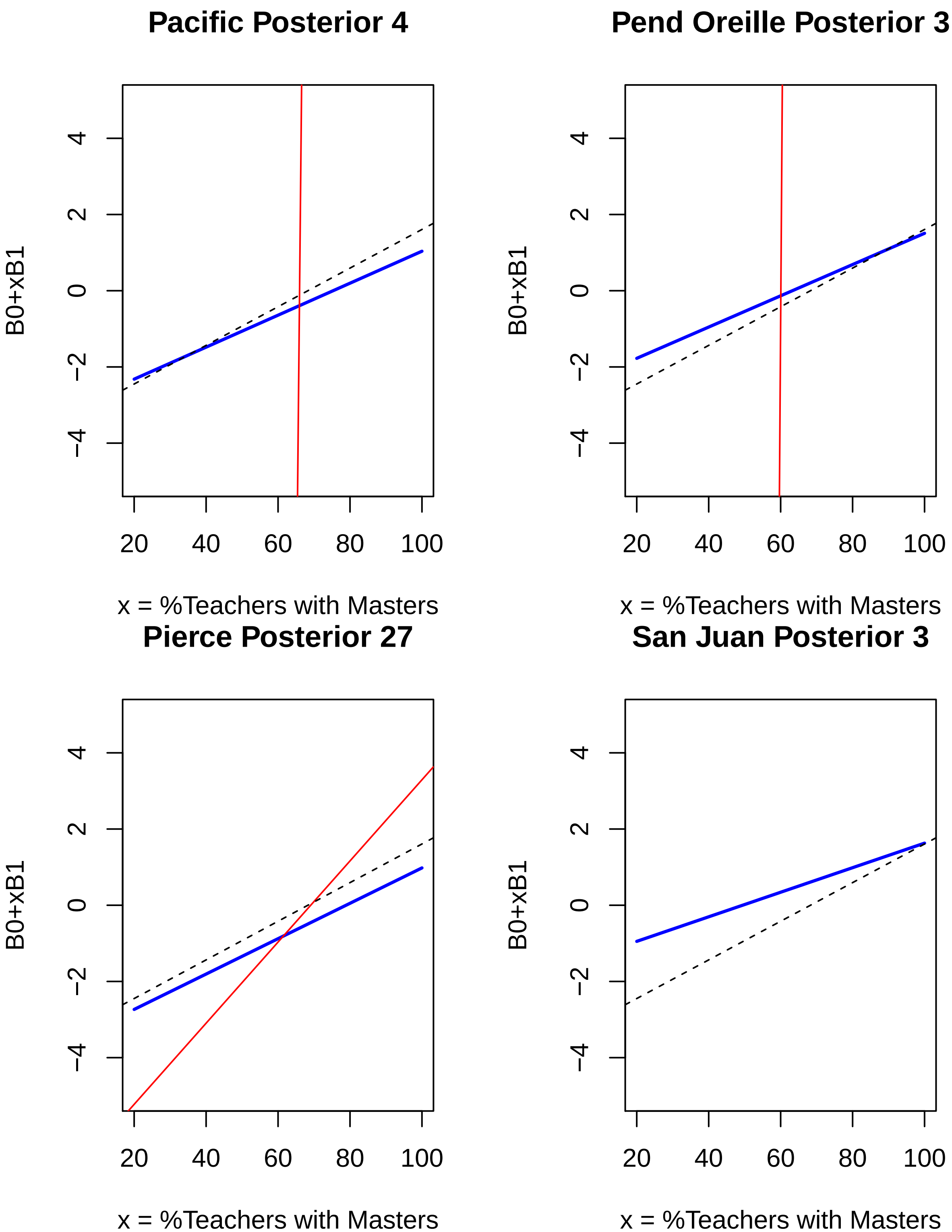

Grays Harbor Posterior 10
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters

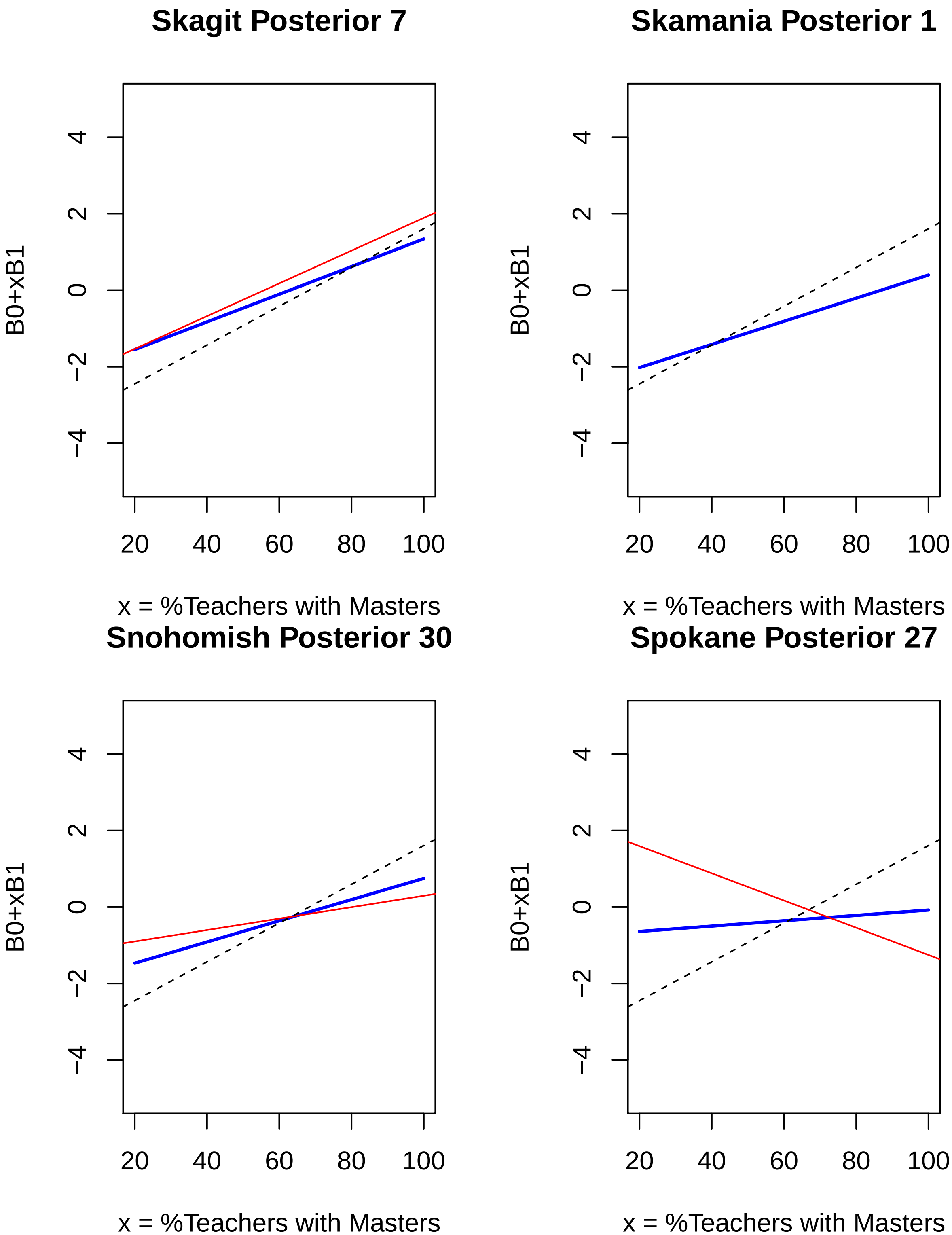

Island Posterior 3
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters

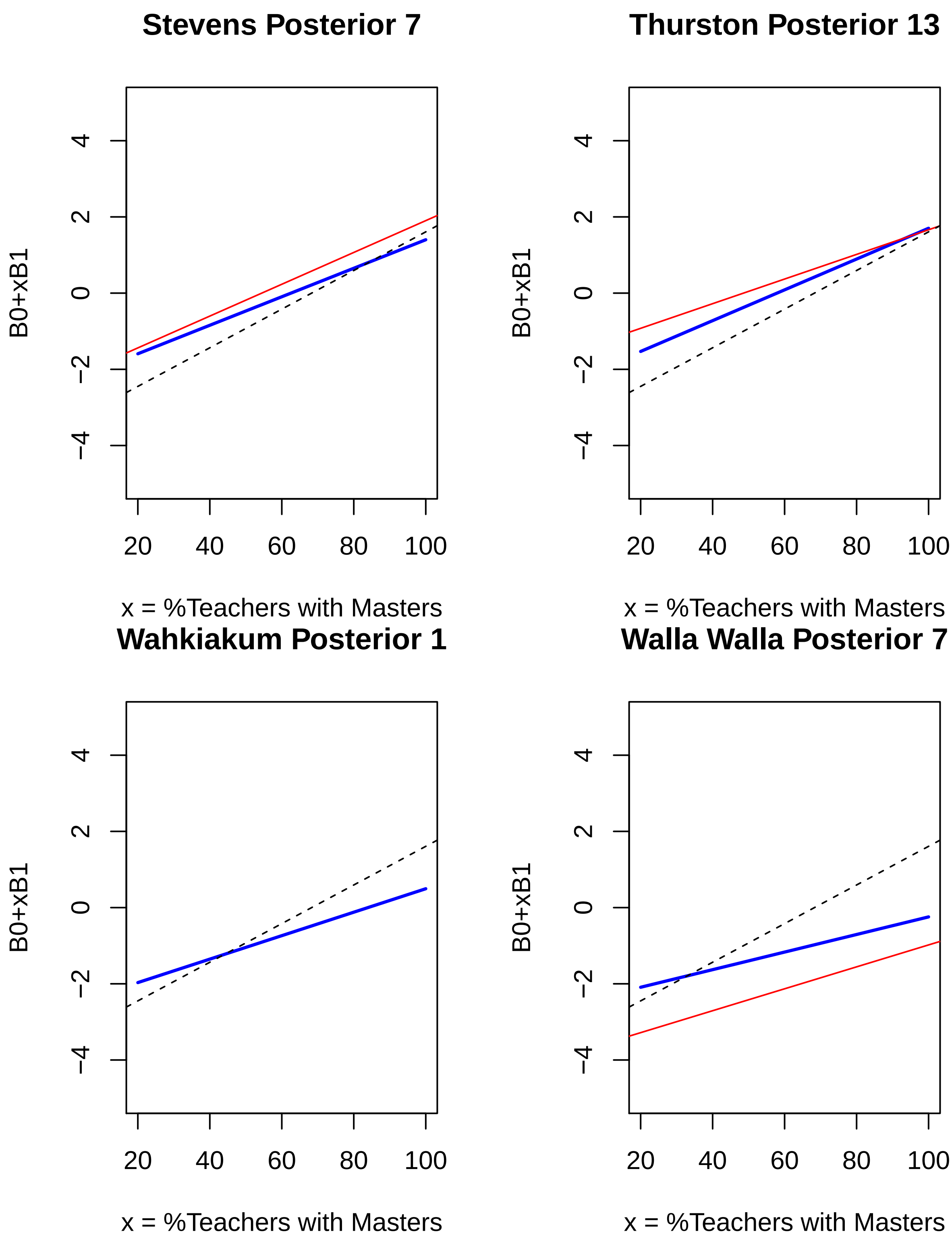

Jefferson Posterior 3
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters

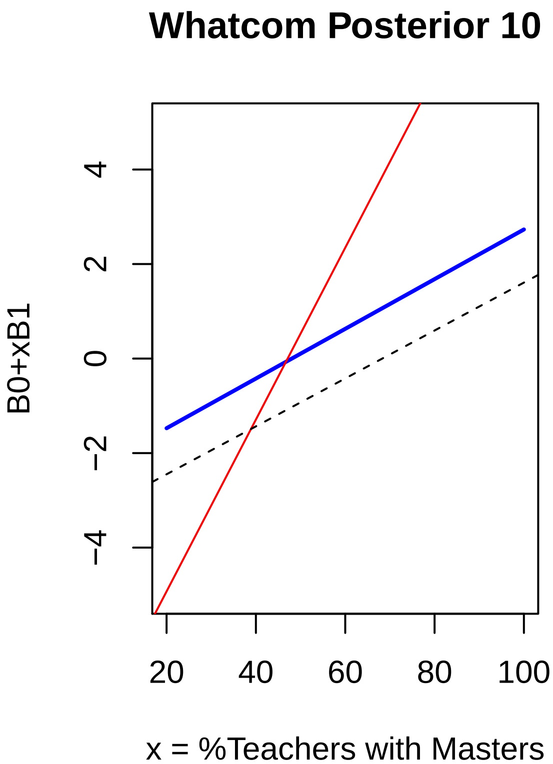

King Posterior 64
Kitsap Posterior 10
Kittitas Posterior 3
Klickitat Posterior 4
B0+xB1
x = %Teachers with Masters
−4
−2
0
2
4
20
40
60
80
100

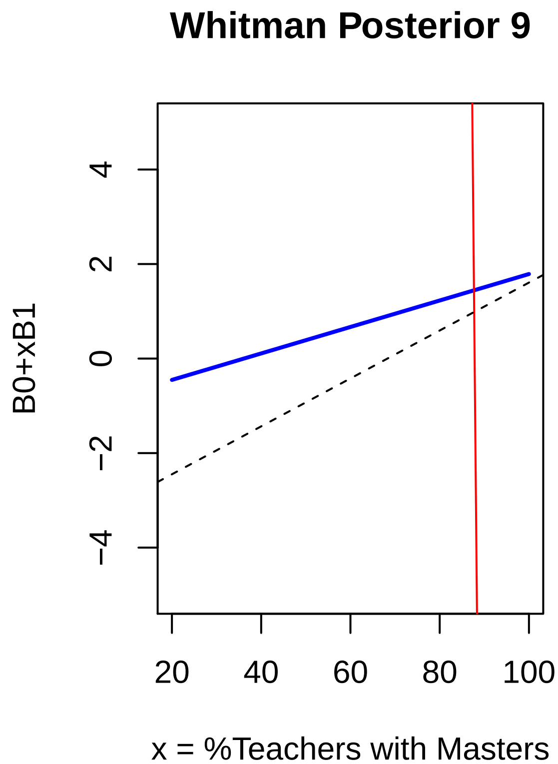

Lewis Posterior 12
Lincoln Posterior 5
Mason Posterior 5
Okanogan Posterior 7
B0+xB1
x = %Teachers with Masters
−4
−2
0
2
4
20
40
60
80
100

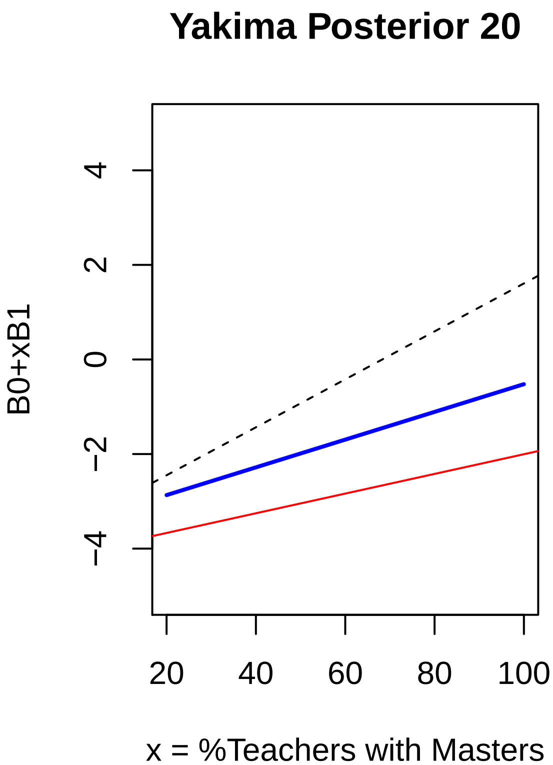

Pacific Posterior 4
Pend Oreille Posterior 3
Pierce Posterior 27
San Juan Posterior 3
B0+xB1
x = %Teachers with Masters
−4
−2
0
2
4
20
40
60
80
100


Skagit Posterior 7
Skamania Posterior 1
Snohomish Posterior 30
Spokane Posterior 27
B0+xB1
x = %Teachers with Masters
−4
−2
0
2
4
20
40
60
80
100


Stevens Posterior 7
Thurston Posterior 13
Wahkiakum Posterior 1
Walla Walla Posterior 7
B0+xB1
x = %Teachers with Masters
−4
−2
0
2
4
20
40
60
80
100


Whatcom Posterior 10
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters


Whitman Posterior 9
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters


Yakima Posterior 20
B0+xB1
−4
−2
0
2
4
20
40
60
80
100
x = %Teachers with Masters

Comments

In the plots above, some counties have red (old) and blue (new) lines which look quite similar, some have lines that are different but only slightly so, and some have lines which are extremely different (in some cases, the red line is not even visible on the plot. This is due to the extreme intercept values $\beta_{0,j}$ for these estimates). One may notice that for plots with lines which are very different from each other, the sample sizes tent to be quite small, while those with large sample sizes tent ot have simlar lines. Also note that for values with small sample sizes, the blue lines and the black dashed line tend to look quite similar. This is due to shinkage.

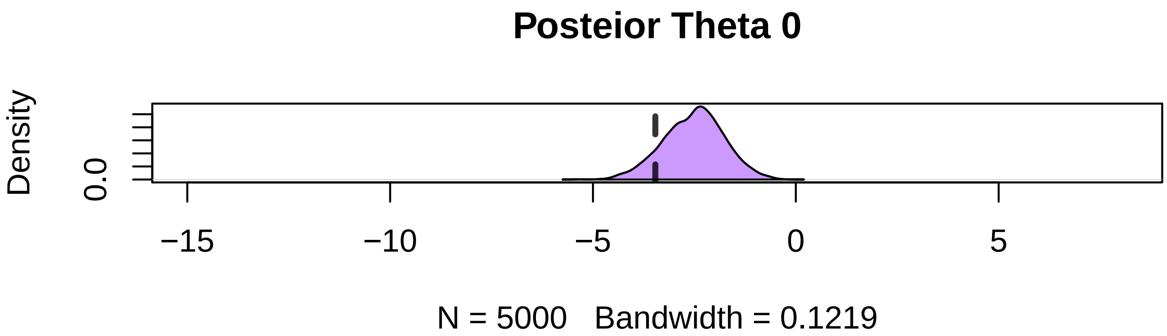
Part f

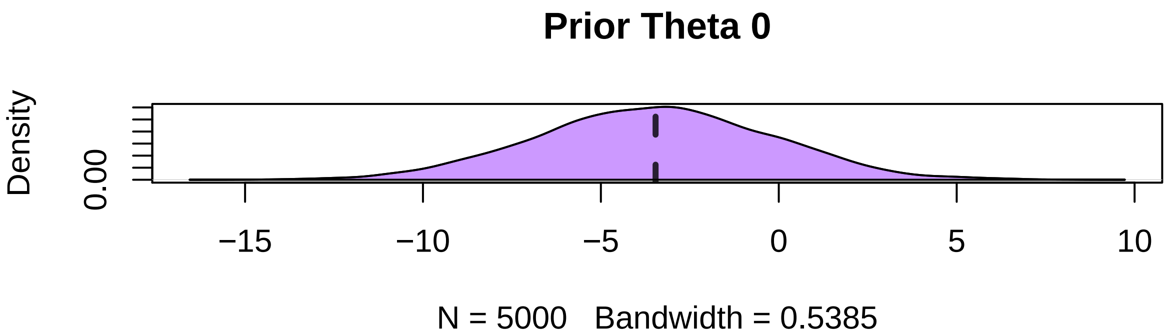
Below, the black dashed line represents my ad hoc estimates calculated in part bfor these values.

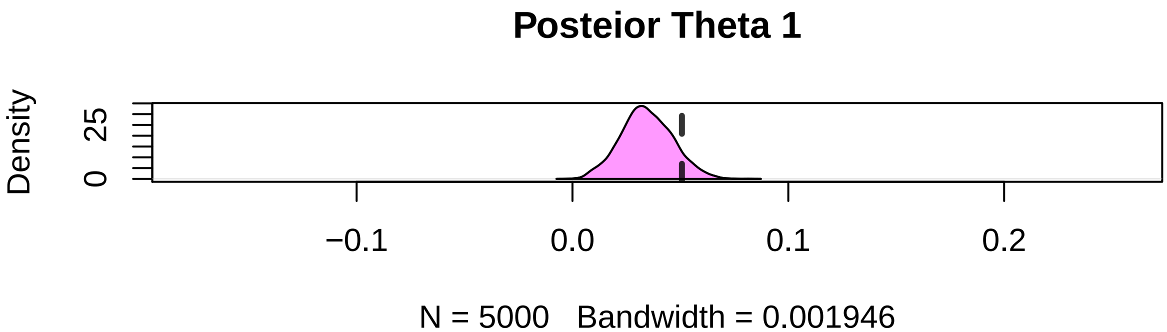
```
library(MCMCpack)
```

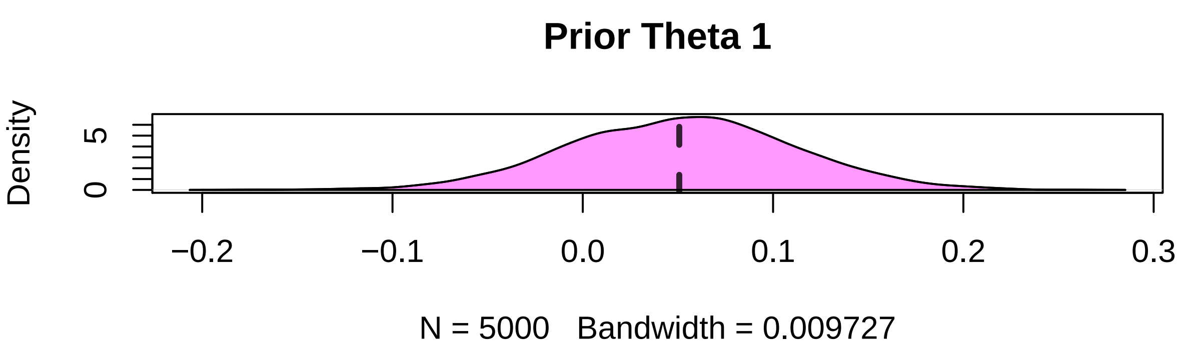
```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
## ## Copyright (C) 2003-2018 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
## ##
## ## Support provided by the U.S. National Science Foundation
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
set.seed(10)
priorTheta <- rmvnorm(Samples, theta.hat, Sigma.hat)
priorSigma <- matrix(0, nrow = Samples, ncol = 3);
priorMean <- solve(Sigma.hat);
for(i in 1:Samples){
   priorSigma[i,]<- sigmaToVec(solve(rWishart(1, 4, priorMean)[,,1]));
}

for(i in 1:2){
  par(mfrow = c(2,1))
  plot(density(THETA[,i]), main = paste('Posteior Theta', i-1),
       xlim =c(min(priorTheta[,i]),max(priorTheta[,i])))
  polygon(density(THETA[,i]), col=rgb(.5*i,0,1,.4), border="black")
  abline(v=theta.hat[i], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
  plot(density(priorTheta[,i]), main = paste('Prior Theta', i-1))
  polygon(density(priorTheta[,i]), col=rgb(.5*i,0,1,.4), border="black")
  abline(v=theta.hat[i], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
}
```



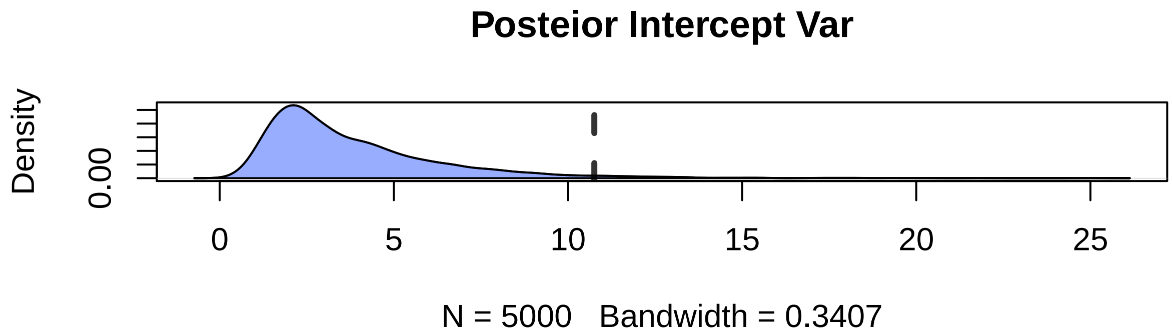


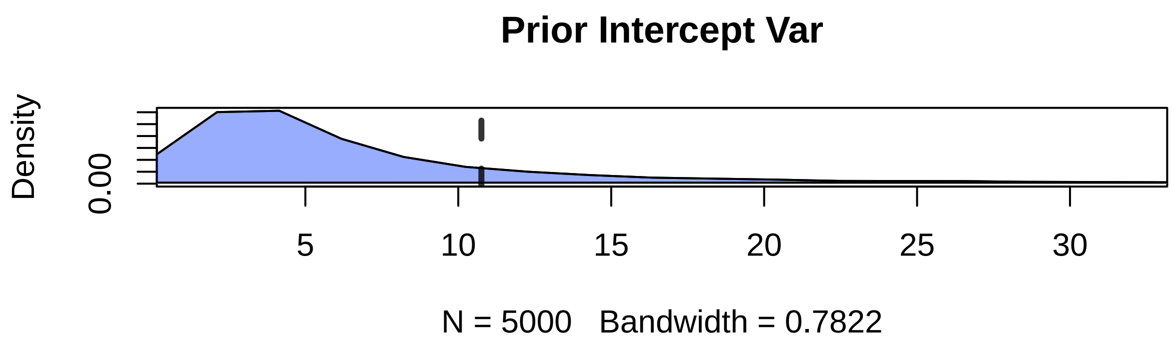




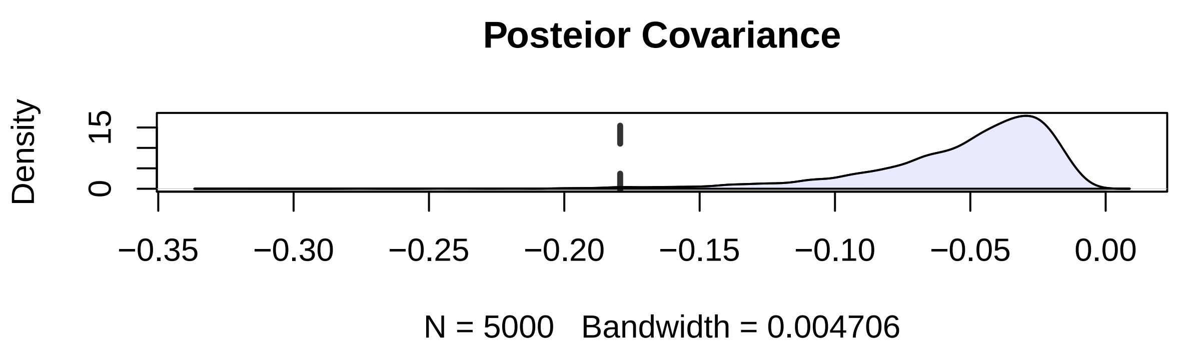

```
a1 <- quantile(priorSigma[,1], c(.05, .95))
a2 <- quantile(priorSigma[,2], c(.05, .95))
a3 <- quantile(priorSigma[,3], c(.05, .95))
par(mfrow = c(2,1))
plot(density(SIGMA[,1]), main = paste('Posteior Intercept Var'))
```

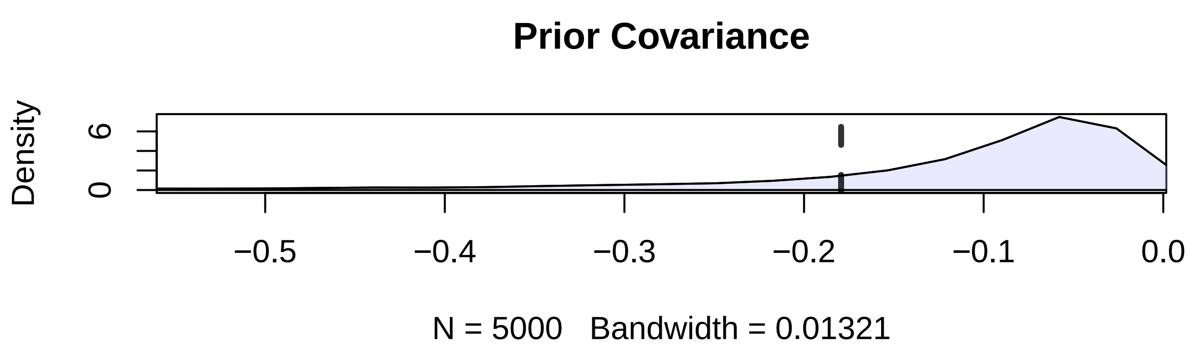
```
polygon(density(SIGMA[,1]), col=rgb(0,.2,1,.4), border="black")
abline(v=Sigma.hat[1,1], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
plot(density(priorSigma[,1]), main = paste('Prior Intercept Var'),
     xlim=c(a1[1], a1[2]))
polygon(density(priorSigma[,1]), col=rgb(0,.2,1,.4), border="black")
abline(v=Sigma.hat[1,1], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
```



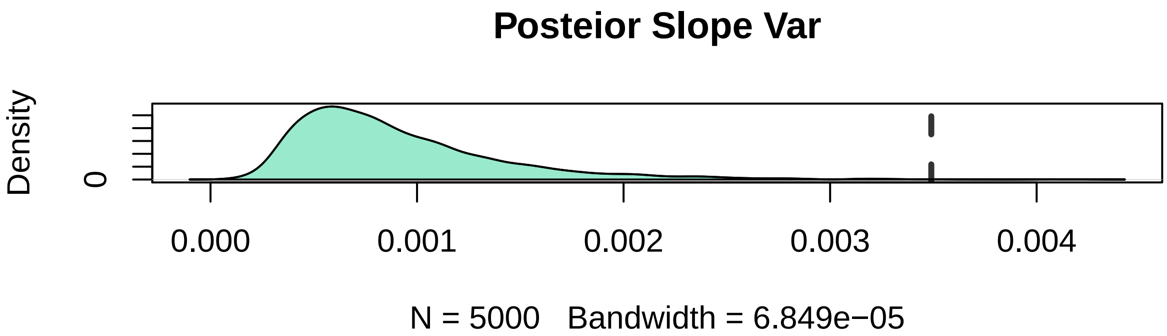


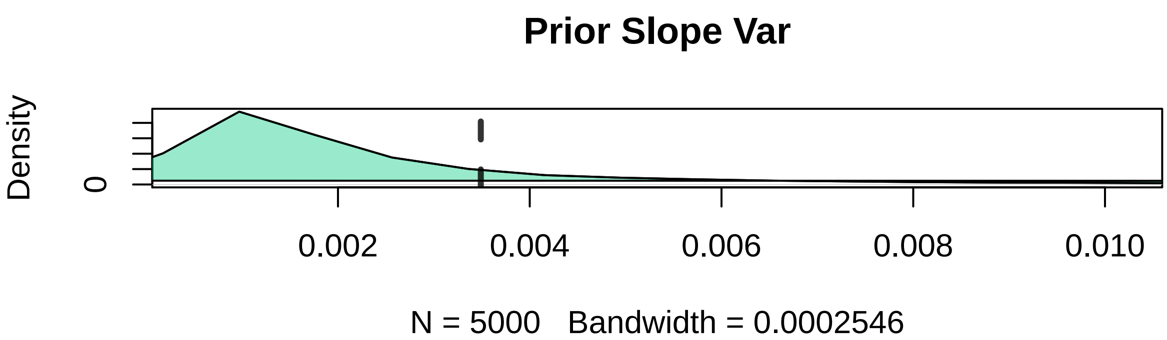

```
par(mfrow = c(2,1))
plot(density(SIGMA[,2]), main = paste('Posteior Covariance'))
polygon(density(SIGMA[,2]), col=rgb(.8,.8,1,.4), border="black")
abline(v=Sigma.hat[1,2], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
plot(density(priorSigma[,2]), main = paste('Prior Covariance'),
     xlim=c(a2[1], a2[2]))
polygon(density(priorSigma[,2]), col=rgb(.8,.8,1,.4), border="black")
abline(v=Sigma.hat[1,2], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
```






```r
par(mfrow = c(2,1))
plot(density(SIGMA[,3]), main = paste('Posteior Slope Var'))
polygon(density(SIGMA[,3]), col=rgb(0,.8,.5,.4), border="black")
abline(v=Sigma.hat[2,2], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
plot(density(priorSigma[,3]), main = paste('Prior Slope Var'),
     xlim=c(a3[1], a3[2]))
polygon(density(priorSigma[,3]), col=rgb(0,.8,.5,.4), border="black")
abline(v=Sigma.hat[2,2], lwd = 3, col = rgb(0,0,0,.8), lty = 2)
```





Given the size of the spread of my variences of my Σ values, I conclude that there is reasonable evidence to believe that there are differences between the slopes and intercepts across groups.