```r
library(bbmle)
library(arm)
library(ggplot2)
library(reshape2)
library(dplyr)
data <- na.omit(read.delim("County Vote for McCain.txt", header=TRUE))
```

1

Estimate the following model via maximum likelihood using bbmle in R and interpret your output.

$$logodds_i \sim N(\mu_i, \sigma^2) \tag{1}$$
$$\mu_i = \beta_0 + \beta_1 pcollege_i + \beta_2 medinc_i \tag{2}$$

Specifically, do the following:

a. Write an R function for the log-likelihood that can be called from mle2 and optimized to estimate the model above.

```r
# Function to return negative loglikelihood
# Input:
#   params: a vector of parameters, i.e. \beta_0, \beta_1, ..., \sigma
#   y, X: vector of outcome, matrix of covariates
# Output: minus log likelihood
LL_normreg = function(params) {
  B = matrix(NA, nrow = length(params) - 1, ncol = 1)
  B[,1] = params[-length(params)] # dim(B) = K x 1, length(params) = K + 1
  sigma    = params[[length(params)]]
  minusll  = -sum(dnorm(y, X %*% B, sigma, log=T))
  return(minusll)
}
```

b. Use mle2 to optimize the function given the provided data.

Solution

```r
# Declare the names of the parameters (from B0 to B[# of predictors], and sigma):
parnames(LL_normreg) <- c("B0", "B1", "B2", "sigma")

# Fit the model using mle2 ('vecpar=TRUE' tells mle2 that the first argument passed to the
# LL function is a vector of all parameters with names declared in 'parnames' above and in

y <- data$logodds
X <- as.matrix(cbind(1, data[ , c("pcollege", "medinc")])) ; colnames(X) <- NULL
```

```r
m_1b <- mle2(LL_normreg, start = c(B0 = mean(y), B1 = 0, B2 = 0, sigma = sd(y)),
             data=list(y=y, X=X),
             vecpar = TRUE, control=list(maxit=5000))

## Warning in dnorm(y, X %*% B, sigma, log = T): NaNs produced
## Warning in dnorm(y, X %*% B, sigma, log = T): NaNs produced
## Warning in dnorm(y, X %*% B, sigma, log = T): NaNs produced

summary(m_1b)

## Maximum likelihood estimation
##
## Call:
## mle2(minuslogl = LL_normreg, start = c(B0 = mean(y), B1 = 0,
##     B2 = 0, sigma = sd(y)), data = list(y = y, X = X), vecpar = TRUE,
##     control = list(maxit = 5000))
##
## Coefficients:
##          Estimate  Std. Error     z value     Pr(z)
## B0     4.7461e-01  4.6417e-12  1.0225e+11 < 2.2e-16 ***
## B1    -5.9881e+00  4.3360e-13 -1.3810e+13 < 2.2e-16 ***
## B2     6.2079e-06  2.2720e-07  2.7324e+01 < 2.2e-16 ***
## sigma  5.5294e-01  3.4739e-18  1.5917e+17 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log L: 4361.867
```

c. Generate marginal effects on McCains share of the two-party vote (not the logodds) for both predictors. That is, calculate the difference in McCains predicted proportion comparing counties at the 95th percentile of each predictor to those at the 5th percentile, holding the other variable at its median value. [Note: you do not need to generate confidence intervals for the marginal effects for this problem].

Solution

TA's Note 1: The problem asks for the marginal effect on McCain's sahre of the two-party vote. Thus, we need to know how to convert logodds to the share as follows. (This is something worth understanding for logit model later).

Let p denotes McCain's share / proportion of vote. We have:

$$ln(\frac{p}{1-p}) = logodd \qquad \text{definition of logodd} \qquad (3)$$

$$\frac{p}{1-p} = \exp(logodd) \qquad \text{exponentiate both sides} \qquad (4)$$

$$p = \frac{\exp(logodd)}{1+\exp(logodd)} \qquad (5)$$

$$= \frac{1}{1-exp(-logodd)} \qquad \text{another expression of } p \qquad (6)$$

TA's Note 2: `predict` only works when the model is fit using a formula (which we're not doing). Thus, we have to calculate the predicted values by hand as follows.

Marginal effect of `pcollege` on `logodds`:

```
q90_pcollege <- quantile(data$pcollege, probs = c(0.05, 0.95))

(pred_logodd_college <- coef(m_1b)[1] + coef(m_1b)[2] * q90_pcollege +
  coef(m_1b)[3] * median(data$medinc))

##         5%        95%
##  0.5059787 -0.1834521
```

Marginal effect of `pcollege` on `pmccain`:

```
# Convert logodds to original variable vote share
(pred_pmccain_college <- exp(pred_logodd_college) / (1 + exp(pred_logodd_college)))

##        5%       95%
## 0.6238633 0.4542652

# Alternatively, use a built-in function to convert logodds to original var
(plogis(pred_logodd_college))

##        5%       95%
## 0.6238633 0.4542652

(me_pmccain_college <- pred_pmccain_college["95%"] - pred_pmccain_college["5%"])

##        95%
## -0.1695981
```

Similarly, marginal effect of `medinc` on `pmccain`:

```
q90_medinc <- quantile(data$medinc, probs = c(0.05, 0.95))
pred_pmccain_medinc <- plogis(coef(m_1b)[1] + coef(m_1b)[2] * median(data$pcollege) +
                              coef(m_1b)[3] * q90_medinc)
(me_pmccain_medinc <- pred_pmccain_medinc["95%"] - pred_pmccain_medinc["5%"])
```

```
##        95%
## 0.05672926
```

d. Interpret each of these effects in substantive terms: what do the results say about the predictors of McCain support and their influence relative to one another? Describe the results in an intuitive way with respect to the scales of the predictors, such that your reader can get a sense of how these variables relate to the DV.

Solution

Comparing counties with `pcollege`, i.e. county proportion with college degree, at 5% and 95% percentile (i.e. 0.04, 0.16), county proportion vote for McCain changes from 0.62 to 0.45.

Comparing counties with `medinc`, i.e. median income, at 5% and 95% percentile (i.e. $31000 vs $69000), county proportion vote for McCain changes from 0.55 to 0.6.

2

Estimate the following model via maximum likelihood using bbmle in R and interpret your output for the variance equation.

$$pmccain_i \sim N(\mu_i, \sigma_i^2) \tag{7}$$
$$\mu_i = \beta_0 + \beta_1 pcollege_i + \beta_2 medinc_i \tag{8}$$
$$\sigma_i^2 = \gamma_0 + \gamma_1 ginicnty_i \tag{9}$$

a. Write an R function for the log-likelihood that can be called from mle2 and optimized.

```
LL_normreg_hetero <- function(b0, b1, b2, g0, g1) {
  - sum(dnorm(y, b0 + b1*x1 + b2*x2, sqrt(g0 + g1*x3), log = T))
}
```

b. Use mle2 to optimize the function given the provided data.

```
m_2b <- mle2(LL_normreg_hetero,
             start = list(b0 = mean(y), b1 = 0, b2 = 0, g0 = var(y), g1 = 0),
             data = list(y = data$pmccain,
                         x1 = data$pcollege, x2 = data$medinc,
                         x3 = data$ginicnty))
summary(m_2b)

## Maximum likelihood estimation
##
## Call:
```

```
## mle2(minuslogl = LL_normreg_hetero, start = list(b0 = mean(y),
##     b1 = 0, b2 = 0, g0 = var(y), g1 = 0), data = list(y = data$pmccain,
##     x1 = data$pcollege, x2 = data$medinc, x3 = data$ginicnty))
##
## Coefficients:
##       Estimate  Std. Error     z value     Pr(z)
## b0  6.1724e-01  1.9885e-07  3.1041e+06 < 2.2e-16 ***
## b1 -1.2850e+00  3.9076e-08 -3.2885e+07 < 2.2e-16 ***
## b2  1.1600e-06  4.9914e-08  2.3241e+01 < 2.2e-16 ***
## g0 -1.7827e-02  3.4753e-04 -5.1294e+01 < 2.2e-16 ***
## g1  7.7207e-02  1.4532e-04  5.3128e+02 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log L: -3513.17
```

c. Calculate the marginal effect of a 5-95% change in ginicnty on the standard deviation of the error term for the model.

Solution

```
q90_ginicnty <- quantile(data$ginicnty, probs = c(0.05, 0.95))

# Predicted value of error sd when ginicnty is at 5% and 95% percentile
(pred_errorsd <- sqrt(coef(m_2b)["g0"] + coef(m_2b)["g1"] * q90_ginicnty))

##        5%       95%
## 0.1084036 0.1422845
```

The marginal effect of a 5-95% change in ginicnty on the standard deviation of the error term is 0.03.

d. Describe and interpret this marginal effect. Does inequality have a statistically significant effect on the model errors? How do changes in Gini relate to changes in the SD? What is the substantive significance of this effect (if any)?

Solution

Inequality does have a statistically significant effect on the model errors with very small p value. If Gini changes from 5% to 95%, i.e. from 0.3831 to 0.49311, the standard error changes from 0.11 to 0.14.

Given that the unconditional SD is 0.13, this change is substantively not very big.

3

Estimate the following model via OLS (use lm) and use a simulation-based approach (use the sim function in R) to generate point estimates and 95% confidence intervals for all predictors. Plot the estimates and their associated

confidence intervals in a pretty graph, and interpret each effect in substantive terms.

$$pmccain \sim N(\mu_i, \sigma^2) \tag{10}$$

$$\mu_i = \beta_0 + \beta_1 pcollege_i + \beta_2 medinc_i + \beta_3 pblack_i + \beta_4 phisp_i + \beta_5 ginicnty_i \tag{11}$$

```
m_3 <- lm(pmccain ~ pcollege + medinc + pblack + phisp + ginicnty, data = data)
sim_3 <- sim(m_3, n.sims = 1000)
```

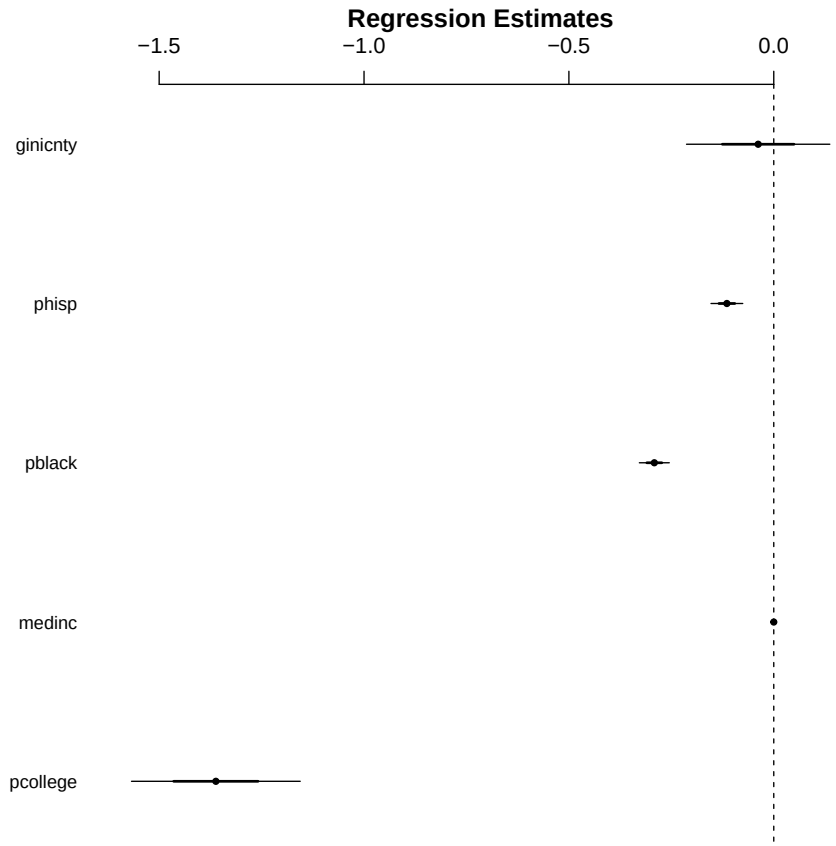
Point estimates and 95% confidence interval:

```
(est_3 <- apply(coef(sim_3), 2, quantile, probs = c(0.025, 0.5, 0.975)))
```

```
##       (Intercept)  pcollege       medinc     pblack       phisp
## 2.5%    0.6154837 -1.587850 3.593557e-08 -0.3267790 -0.15236837
## 50%     0.6961334 -1.359520 6.804561e-07 -0.2919090 -0.11537871
## 97.5%   0.7760588 -1.161847 1.353915e-06 -0.2527688 -0.07767121
##          ginicnty
## 2.5%  -0.20052282
## 50%   -0.04038777
## 97.5%  0.13172740
```

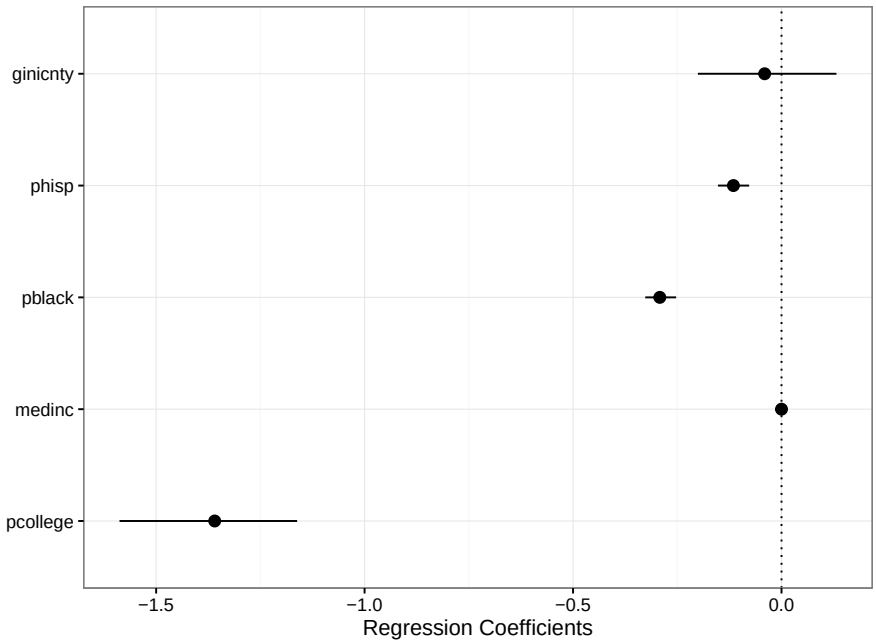
Plot:

```
coefplot(m_3)
```



That was too easy, so here's another plot

```
pd <- melt(est_3) %>% filter(Var2 != "(Intercept)") %>%
  dcast(Var2 ~ Var1)
ggplot(data = pd) +
  geom_pointrange(aes(x = Var2, y = `50%`, ymin = `2.5%`, ymax = `97.5%`)) +
  geom_hline(aes(yintercept = 0), linetype = "dotted") +
  coord_flip() + theme_bw() +
  labs(x = "", y = "Regression Coefficients")
```


ginicnty
phisp
pblack
medinc
pcollege
−1.5
−1.0
−0.5
0.0
Regression Coefficients