April 7 assignment

MATH 301 Data Confidentiality

Henrik Olsson

April 7, 2020

CE sample synthesis

```
CEdata <- read.csv(file = "CEdata.csv")
CEdata$LogIncome <- log(CEdata$Income)
CEdata$LogExpenditure <- log(CEdata$Expenditure)
```

```
n <- dim(CEdata)[1]
synthetic_one <- synthesize_loginc(CEdata$LogExpenditure,
                                   1, n, seed = 123)
names(synthetic_one) <- c("LogExpenditure", "LogIncome")
CEdata_org <- CEdata[, 1:4]
CEdata_syn <- as.data.frame(cbind(CEdata_org[, "UrbanRural"],
                                  exp(synthetic_one
                                      [, "LogIncome"]),
                                  cbind(CEdata_org
                                        [, c("Race",
                                             "Expenditure")])))
names(CEdata_syn) <- c("UrbanRural", "Income",
                       "Race", "Expenditure")
```

```
CEdata_org$LogIncome <- round(log(CEdata_org$Income),
                              digits = 1)
CEdata_org$LogExpenditure <- round(log(CEdata_org$Expenditure),
                                   digits = 1)
CEdata_syn$LogIncome <- round(log(CEdata_syn$Income),
                              digits = 1)
CEdata_syn$LogExpenditure <- round(log(CEdata_syn$Expenditure),
                                   digits = 1)

i <- 8
y_i <- CEdata_org$LogIncome[i]
y_i_guesses <- seq((y_i - 2.5), (y_i + 2.5), 0.5)
X_i <- CEdata_syn$LogExpenditure[i]
G <- length(y_i_guesses)

compute_logsumexp <- function(log_vector){
  log_vector_max <- max(log_vector)
  exp_vector <- exp(log_vector - log_vector_max)
  sum_exp <- sum(exp_vector)
  log_sum_exp <- log(sum_exp) + log_vector_max
  return(log_sum_exp)
}

H <- 50
beta0_draws <- post[1:H, "beta0"]
```

```r
beta1_draws <- post[1:H, "beta1"]
sigma_draws <- post[1:H, "sigma"]

CU_i_logZ_all <- rep(NA, G)
for (g in 1:G){
  q_sum_H <- sum((dnorm(y_i_guesses[g],
                        mean = (beta0_draws + beta1_draws * X_i),
                        sd = sigma_draws)) /
            (dnorm(y_i, mean = (beta0_draws + beta1_draws * X_i),
                   sd = sigma_draws)))
  log_pq_h_all <- rep(NA, H)
  for (h in 1:H){
    log_p_h <- sum(log(dnorm(CEdata_syn$LogIncome,
                             mean = (beta0_draws[h] + beta1_draws[h] *
                                       CEdata_syn$LogExpenditure),
                             sd = sigma_draws[h])))

    log_q_h <- log(((dnorm(y_i_guesses[g],
                           mean = (beta0_draws[h] + beta1_draws[h] * X_i),
                           sd = sigma_draws[h])) /
            (dnorm(y_i, mean = (beta0_draws[h] + beta1_draws[h] * X_i),
                   sd = sigma_draws[h]))) / q_sum_H)
    log_pq_h_all[h] <- log_p_h + log_q_h
  }
  CU_i_logZ_all[g] <- compute_logsumexp(log_pq_h_all)
}

prob <- exp(CU_i_logZ_all - max(CU_i_logZ_all)) /
  sum(exp(CU_i_logZ_all - max(CU_i_logZ_all)))
outcome <- as.data.frame(cbind(y_i_guesses, prob))
names(outcome) <- c("guess", "probability")
outcome[order(outcome$probability, decreasing = TRUE), ]
```

```
##    guess probability
## 8   12.6  0.09231563
## 7   12.1  0.09228750
## 9   13.1  0.09204320
## 6   11.6  0.09203442
## 5   11.1  0.09160126
## 10  13.6  0.09136939
## 4   10.6  0.09099926
## 3   10.1  0.09020571
## 11  14.1  0.09017674
## 2    9.6  0.08916632
## 1    9.1  0.08780057
```