Bayesian Synthesis Models part 1

Jingchen (Monika) Hu

Vassar College

Statistical Data Privacy

Outline

Outline

Recap

- Lecture 1 Overview of synthetic data: various aspects of creating synthetic data for microdata privacy protection

Recap

- Lecture 1 Overview of synthetic data: various aspects of creating synthetic data for microdata privacy protection
- Lectures 2 & 3 Introduction to Bayesian modeling: nuts and bolts of Bayesian modeling
  - The foundation of Bayesian inference: prior, likelihood, Bayes' rule (discrete and continuous), and posterior
  - Markov chain Monte Carlo (MCMC): estimation and diagnostics
  - Posterior predictive and synthetic data
  - Case study: gamma-Poisson conjugate model, Poisson regression model, posterior prediction, prior choices, posterior inference, MCMC, MCMC diagnostics, using the `brms` package

Synthesis approaches

- Two general approaches to synthetic data creation
  - Sequential synthesis
  - Joint synthesis

- Sequential synthesis
  - More commonly used and easier to estimate
  - The main focus of this lecture and next

Synthesis approaches

- Two general approaches to synthetic data creation
  - Sequential synthesis
  - Joint synthesis
- Sequential synthesis
  - More commonly used and easier to estimate
  - The main focus of this lecture and next
- Joint synthesis
  - Less commonly used and usually more challenging to estimate
  - The DPMPM model for multivariate nominal categorical data (Hu, Reiter, and Wang (2014)), next lecture

The CE sample

- Our sample is from the 1st quarter of 2019, containing five variables on 5133 CUs

Variable Name	Variable information
UrbanRural	Binary; the urban / rural status of CU: 1 = Urban, 2 = Rural.
Income	Continuous; the amount of CU income before taxes in past 12 months (in *USD*).
Race	Categorical; the race category of the reference person: 1 = White, 2 = Black, 3 = Native American, 4 = Asian, 5 = Pacific Islander, 6 = Multi-race.
Expenditure	Continuous; CU's total expenditures in last quarter (in *USD*).
KidsCount	Count; the number of CU members under age 16.

The CE sample

```
CEdata <- readr::read_csv(file = "CEdata.csv")
CEdata[1:3, ]
```

```
## # A tibble: 3 x 5
##   UrbanRural Income  Race Expenditure KidsCount
##        <dbl>  <dbl> <dbl>       <dbl>     <dbl>
## 1          1  73720     1      27542.         3
## 2          1  12000     1       7416.         2
## 3          1  20000     1       8608.         0
```

Plan for this lecture

- Synthesis models for continuous outcome variables (e.g., Expenditure and Income)
- Synthesis models for binary outcome variables (e.g., UrbanRural)

Outline

Log transformation

- Suppose we want to use the Income variable to perform a partial synthesis for Expenditure
- Both Income and Expenditure are highly skewed, we apply the logarithm transformation for both in our model building, creating two new variables: LogIncome and LogExpenditure

```
CEdata$LogIncome <- log(CEdata$Income)
CEdata$LogExpenditure <- log(CEdata$Expenditure)
```

A Bayesian simple linear regression: model specification

- Y_i is LogExpenditure and X_i is LogIncome for CU i
- A Bayesian simple linear regression model:

$$Y_i \overset{ind}{\sim} \text{Normal}(\mu_i, \sigma) \quad (1)$$

$$\mu_i = \beta_0 + \beta_1 X_i \quad (2)$$

A Bayesian simple linear regression: model specification

- Y_i is LogExpenditure and X_i is LogIncome for CU i
- A Bayesian simple linear regression model:

$$Y_i \overset{ind}{\sim} \text{Normal}(\mu_i, \sigma) \quad (1)$$

$$\mu_i = \beta_0 + \beta_1 X_i \quad (2)$$

- The expected LogExpenditure of CU i is μ_i, which is a linear function of LogIncome X_i through the **intercept parameter** β_0 and the **slope parameter** β_1

Discussion question: How would you interpret the intercept β_0 and the slope β_1 in this CE context?

A Bayesian simple linear regression: model estimation

- We use the `brms` R package to estimate our chosen Bayesian simple linear regression model

A Bayesian simple linear regression: model estimation

- We use the `brms` R package to estimate our chosen Bayesian simple linear regression model
- We will obtain pre-specified number of **posterior parameter draws** from the output, a process similar to what we have seen for the bike sharing rental case study
- These posterior parameter draws will be used for synthetic data generation through the **posterior predictive distribution**

A Bayesian simple linear regression: model estimation

- Make sure to install and load the `brms` library

```
library(brms)
```

- To streamline our synthesis process, we create the **design matrix** $\boldsymbol{X}$ based on the chosen model

```
SLR_ff <- stats::as.formula(LogExpenditure ~ 1 + LogIncome)
SLR_model <- stats::model.frame(SLR_ff, CEdata)
SLR_X <- data.frame(stats::model.matrix(SLR_ff, SLR_model))
```

A Bayesian simple linear regression: model estimation

- We use the **default priors**
- The information of default priors can be extracted by the `get_prior()` function

```
brms::get_prior(data = CEdata,
                family = gaussian,
                LogExpenditure ~ 1 + LogIncome)
```

```
##                   prior     class      coef group resp dpar nlpar bound
##                  (flat)         b
##                  (flat)         b LogIncome
##  student_t(3, 8.9, 2.5) Intercept
##    student_t(3, 0, 2.5)     sigma
##         source
##        default
##  (vectorized)
##        default
##        default
```

A Bayesian simple linear regression: model estimation

- We use `family = gaussian` in the `brm()` function to fit the simple linear regression model

```
SLR_fit <- brms::brm(data = CEdata,
                     family = gaussian,
                     LogExpenditure ~ 1 + LogIncome,
                     iter = 5000,
                     warmup = 3000,
                     thin = 1,
                     chains = 1,
                     seed = 539)
```

A Bayesian simple linear regression: model estimation

- The key to applying Bayesian synthesis models is to save posterior parameter draws of estimated parameters
- These draws will be used to generate synthetic data given the posterior predictive distribution
- We use the `posterior_samples()` function to retrieve the posterior parameter draws in `post_SLR`

```
post_SLR <- brms::posterior_samples(x = SLR_fit)
post_SLR[1:3, ]
```

```
##   b_Intercept b_LogIncome     sigma      lp__
## 1    5.030315   0.3578751 0.7613313 -5799.006
## 2    5.037303   0.3570114 0.7330382 -5799.746
## 3    5.079390   0.3537355 0.7455257 -5797.711
```

Discussion question: What is the dimension of `post_SLR`?

A Bayesian simple linear regression: MCMC diagnostics

```
bayesplot::mcmc_trace(x = SLR_fit,
                      pars = c("b_Intercept", "b_LogIncome", "sigma"))
bayesplot::mcmc_acf(x = SLR_fit,
                    pars = c("b_Intercept", "b_LogIncome", "sigma"))
```

Discussion question: What are being plotted by the commands above? What features of the plots indicate issues with convergence? What remedies do you have to improve the MCMC convergence?

A Bayesian simple linear regression: synthesis

- To predict the LogExpenditure, Y_i^* for a CU given its LogIncome, X_i and a set of parameter draws, denoted as $\{\beta_0^*, \beta_1^*, \sigma^*\}$:

$$Y_i^* \mid \beta_0^*, \beta_1^*, \sigma^* \overset{ind}{\sim} \text{Normal}(\beta_0^* + \beta_1^* X_i, \sigma^*). \tag{3}$$

A Bayesian simple linear regression: synthesis

- Now for each of the n CUs, we create a predicted value

$$\begin{aligned}
\text{compute } E[Y_1^*] = \beta_0^* + \beta_1^* X_1 \quad &\rightarrow \quad \text{sample } Y_1^* \sim \text{Normal}(E[Y_1^*], \sigma^*) \\
&\vdots \\
\text{compute } E[Y_i^*] = \beta_0^* + \beta_1^* X_i \quad &\rightarrow \quad \text{sample } Y_i^* \sim \text{Normal}(E[Y_i^*], \sigma^*) \\
&\vdots \\
\text{compute } E[Y_n^*] = \beta_0^* + \beta_1^* X_n \quad &\rightarrow \quad \text{sample } Y_n^* \sim \text{Normal}(E[Y_n^*], \sigma^*)
\end{aligned}$$

A Bayesian simple linear regression: synthesis

- Suppose we use one set of posterior draws at the `index` iteration of the MCMC

```
SLR_synthesize <- function(X, post_draws, index, n, seed){
  set.seed(seed)
  mean_Y <- as.matrix(X) %*%
    t(data.matrix(post_draws[index, c("b_Intercept", "b_LogIncome")]))
  synthetic_Y <- stats::rnorm(n, mean_Y, post_draws[index, "sigma"])
  data.frame(X, synthetic_Y)
}
```

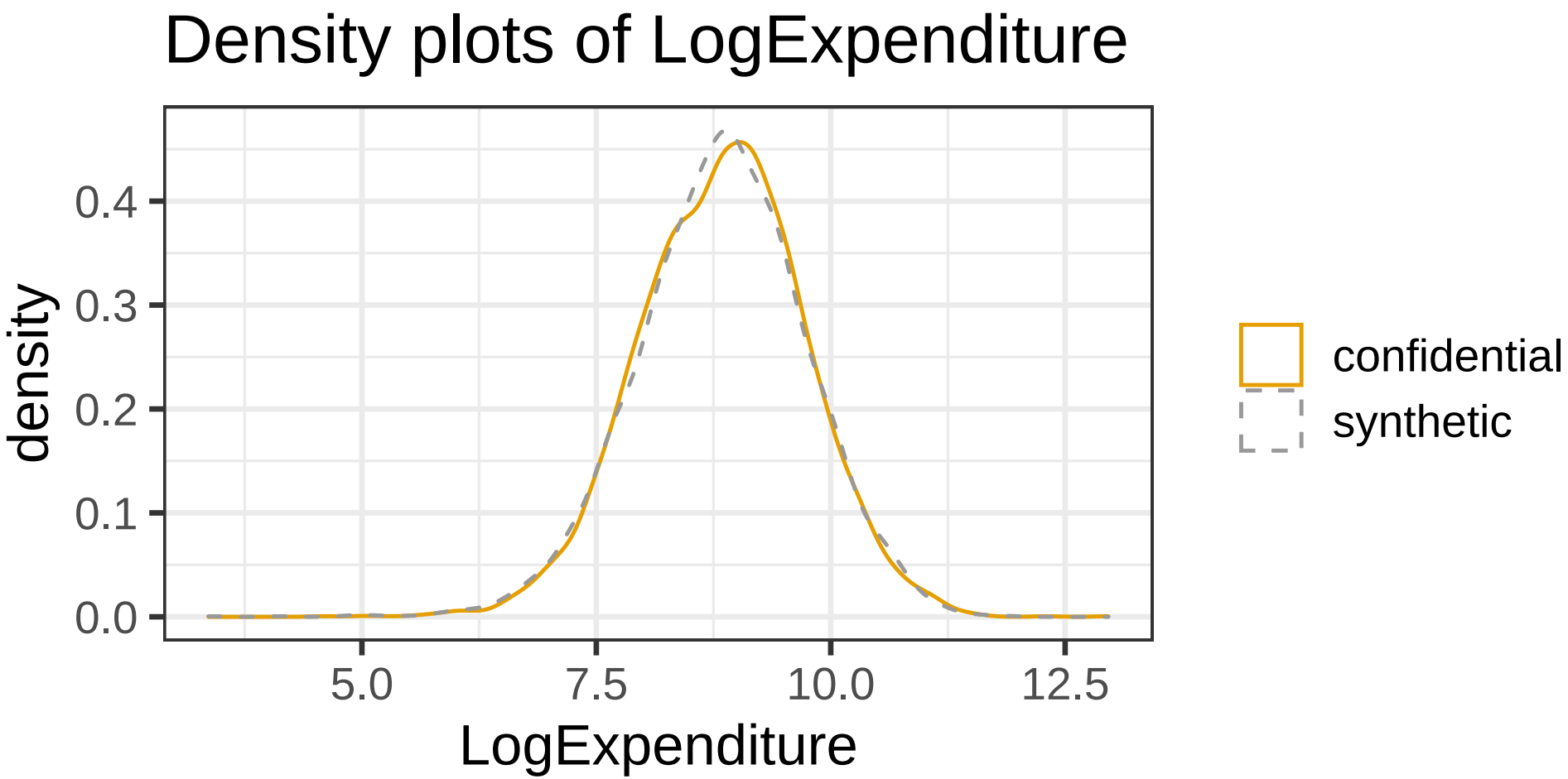
Dicussion question: What are the inputs and outputs of this `SLR_synthesize()` function?

A Bayesian simple linear regression: synthesis

- Perform synthesis for the CE dataset

```
n <- nrow(CEdata)
SLR_synthetic_one <- SLR_synthesize(X = SLR_X,
                                    post_draws = post_SLR,
                                    index = 1,
                                    n = nrow(SLR_X),
                                    seed = 246)
names(SLR_synthetic_one) <- c("Intercept", "LogIncome", "LogExpenditure")
```

A Bayesian simple linear regression: utility check



A Bayesian simple linear regression: $m > 1$ synthetic datasets

```
n <- nrow(CEdata)
m <- 20
SLR_synthetic_m <- vector("list", m)
for (l in 1:m){
  SLR_synthetic_one <- SLR_synthesize(X = SLR_X,
                                      post_draws = post_SLR,
                                      index = 1980 + l,
                                      n = nrow(SLR_X),
                                      seed = m + l)
  names(SLR_synthetic_one) <- c("Intercept", "LogIncome", "LogExpenditure")
  SLR_synthetic_m[[l]] <- SLR_synthetic_one
}
```

A Bayesian multiple linear regression: overview

- Model expression

$$Y_i \overset{ind}{\sim} \text{Normal}(\mu_i, \sigma) \tag{4}$$

$$\mu_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_r X_{ir} \tag{5}$$

- Design matrix

```
MLR_2vars_ff <- stats::as.formula(LogExpenditure ~ 1 + LogIncome +
                                    as.factor(Race))
MLR_2vars_model <- stats::model.frame(MLR_2vars_ff, CEdata)
MLR_2vars_X <- data.frame(stats::model.matrix(MLR_2vars_ff,
                                              MLR_2vars_model))
```

Discussion question: What are the predictors in this model?

Outline

A Bayesian logistic regression: overview

- Binary outcome variables
  - labor participation (0 or 1)
  - loan default (yes or no)
  - UrbanRural in our CE sample

A Bayesian logistic regression: overview

- Binary outcome variables
  - labor participation (0 or 1)
  - loan default (yes or no)
  - UrbanRural in our CE sample
- When modeling a binary outcome variable, regular linear regression models would not work (linear regression models are used to model continuous outcome variables through a normal data model)
- The idea of modeling the outcome variable as a function of predictor variables in linear regression can be extended to modeling other types of outcome variables
  - Categorical and count will be covered later

A Bayesian logistic regression: model specification

- When working with a binary outcome variable, Y_i, we can think of it as a **Bernoulli random variable**:

$$Y_i \overset{ind}{\sim} \text{Bernoulli}(p_i) \tag{6}$$

  where p_i is the success probability of observation i taking $Y_i = 1$
- $p_i \in (0, 1)$

A Bayesian logistic regression: model specification

- When working with a binary outcome variable, Y_i, we can think of it as a **Bernoulli random variable**:
$$Y_i \overset{ind}{\sim} \text{Bernoulli}(p_i) \tag{6}$$
where p_i is the success probability of observation i taking $Y_i = 1$
- $p_i \in (0, 1)$

- The **odds**, $p_i/1 - p_i$ is then a positive, continuous quantity
- If we take its log, $\log(p_i/1 - p_i)$, typically known as the **log odds**, we have an unknown continuous quantity which is on the real line, i.e., $\log(p_i/1 - p_i) \in (-\infty, \infty)$
- This log odds, also known as the **logit of** p_i, $\log(p_i/1 - p_i)$, can then be modeled as a linear function of predictor variables

A Bayesian logistic regression: model specification

- Assume r predictor variables, $X_{i1}, \cdots, X_{ir}$
- A linear function of X_i for the logit of p_i can be expressed as:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_r X_{ir} \tag{7}$$

with $r+1$ parameters, $\{\beta_0, \cdots, \beta_r\}$

A Bayesian logistic regression: model specification

- Assume r predictor variables, $X_{i1}, \cdots, X_{ir}$
- A linear function of X_i for the logit of p_i can be expressed as:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_r X_{ir} \qquad (7)$$

  with $r+1$ parameters, $\{\beta_0, \cdots, \beta_r\}$

- p_i is fixed once $\{\beta_0, \cdots, \beta_r\}$ are known, therefore not considered as a parameter
- From a Bayesian perspective, we specify prior distributions for $\{\beta_0, \cdots, \beta_r\}$ and from MCMC estimation, we obtain posterior draws of these parameters

A Bayesian logistic regression: synthesis details

- Once posterior draws of parameters $\{\beta_0, \cdots, \beta_r\}$ are available, we can then simulate a posterior predictive draw of Y_i^* given predictor variable X_i and a set of posterior parameter draws, denoted as $\{\beta_0^*, \cdots, \beta_r^*\}$:

$$\text{logit}(p_i^*) = \log\left(\frac{p_i^*}{1-p_i^*}\right) = \beta_0^* + \beta_1^* X_{i1} + \cdots + \beta_r^* X_{ir}, \quad (8)$$

$$Y_i^* \overset{ind}{\sim} \text{Bernoulli}(p_i^*). \quad (9)$$

A Bayesian logistic regression: synthesis details

- Note that to calculate p_i^* from $\beta_0^*, \cdots, \beta_r^*$ and X_i, we need the following algebra transformation

$$\begin{aligned} \log\left(\frac{p_i^*}{1-p_i^*}\right) &= \beta_0^* + \beta_1^* X_{i1} + \cdots + \beta_r^* X_{ir} \\ \frac{p_i^*}{1-p_i^*} &= \exp(\beta_0^* + \beta_1^* X_{i1} + \cdots + \beta_r^* X_{ir}) \\ p_i^* &= \frac{\exp(\beta_0^* + \beta_1^* X_{i1} + \cdots + \beta_r^* X_{ir})}{1+\exp(\beta_0^* + \beta_1^* X_{i1} + \cdots + \beta_r^* X_{ir})} \end{aligned} \tag{10}$$

A Bayesian logistic regression: synthesis details

- For notation simplicity, we let $p_i^* = h(\beta_0^*, \cdots, \beta_r^*, \boldsymbol{X}_i)$ represent the previous expression, where $\boldsymbol{X}_i$ is the vector of predictors for observation i
- Then to simulate synthetic values for all n observations using the following procedure

$$\begin{aligned}
\text{compute } p_1^* = h(\beta_0^*, \cdots, \beta_r^*, \boldsymbol{X}_1) &\rightarrow \text{sample } Y_1^* \sim \text{Bernoulli}(p_1^*) \\
&\vdots \\
\text{compute } p_i^* = h(\beta_0^*, \cdots, \beta_r^*, \boldsymbol{X}_i) &\rightarrow \text{sample } Y_i^* \sim \text{Bernoulli}(p_i^*) \\
&\vdots \\
\text{compute } p_n^* = h(\beta_0^*, \cdots, \beta_r^*, \boldsymbol{X}_n) &\rightarrow \text{sample } Y_n^* \sim \text{Bernoulli}(p_n^*)
\end{aligned}$$

- This process creates one synthetic binary vector $(Y_i^*)_{i=1,\cdots,n}$

A Bayesian logistic regression: model estimation

- We wish to synthesize binary UrbanRural with continuous LogExpenditure as the single predictor
- UrbanRural is coded as 1 = Urban and 2 = Rural: we need to create a 0 / 1 version of Y_i as $\tilde{Y}_i = Y_i - 1$

$$\tilde{Y}_i \mid p_i \overset{ind}{\sim} \text{Bernoulli}(p_i) \tag{11}$$

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_i \tag{12}$$

Discussion question: Is $\tilde{Y}_i = 0$ referring to an urban CU or rural CU?

A Bayesian logistic regression: model estimation

- To streamline our synthesis process, we create the **design matrix X** based on the chosen model

```
logistic_ff <- stats::as.formula((UrbanRural - 1) ~ 1 + LogExpenditure)
logistic_model <- stats::model.frame(logistic_ff, CEdata)
logistic_X <- data.frame(stats::model.matrix(logistic_ff, logistic_model))
```

A Bayesian logistic regression: model estimation

- To streamline our synthesis process, we create the **design matrix X** based on the chosen model

```
logistic_ff <- stats::as.formula((UrbanRural - 1) ~ 1 + LogExpenditure)
logistic_model <- stats::model.frame(logistic_ff, CEdata)
logistic_X <- data.frame(stats::model.matrix(logistic_ff, logistic_model))
```

- We use the **default priors**

A Bayesian logistic regression: model estimation

- We use `family = bernoulli(link = "logit")` in the `brm()` function to fit the simple linear regression model

```
logistic_fit <- brms::brm(data = CEdata,
                          family = bernoulli(link = "logit"),
                          (UrbanRural - 1) ~ 1 + LogExpenditure,
                          iter = 5000,
                          warmup = 3000,
                          thin = 1,
                          chains = 1,
                          seed = 720)
```

A Bayesian logistic regression: model estimation

- The key to applying Bayesian synthesis models is to save posterior parameter draws of estimated parameters
- These draws will be used to generate synthetic data given the posterior predictive distribution
- We use the `posterior_samples()` function to retrieve the posterior parameter draws in `post_logistic`

```
post_logistic <- brms::posterior_samples(x = logistic_fit)
post_logistic[1:3, ]
```

```
##   b_Intercept b_LogExpenditure      lp__
## 1   1.7386479       -0.5095163 -1236.636
## 2  -0.0674308       -0.2962953 -1232.647
## 3  -0.1452401       -0.2913404 -1232.939
```

A Bayesian logistic regression: MCMC diagnostics

- Don't forget to do them!

A Bayesian logistic regression: synthesis

- Suppose we use one set of posterior draws at the `index` iteration of the MCMC

```
logistic_synthesize <- function(X, post_draws, index, n, seed){
  set.seed(seed)
  log_p <- as.matrix(X) %*%
    t(data.matrix(post_draws[index, c("b_Intercept", "b_LogExpenditure")]))
  p <- exp(log_p) / (1 + exp(log_p))
  synthetic_Y <- stats::rbinom(n, size = 1, prob = p) + 1
  data.frame(X, synthetic_Y)
}
```

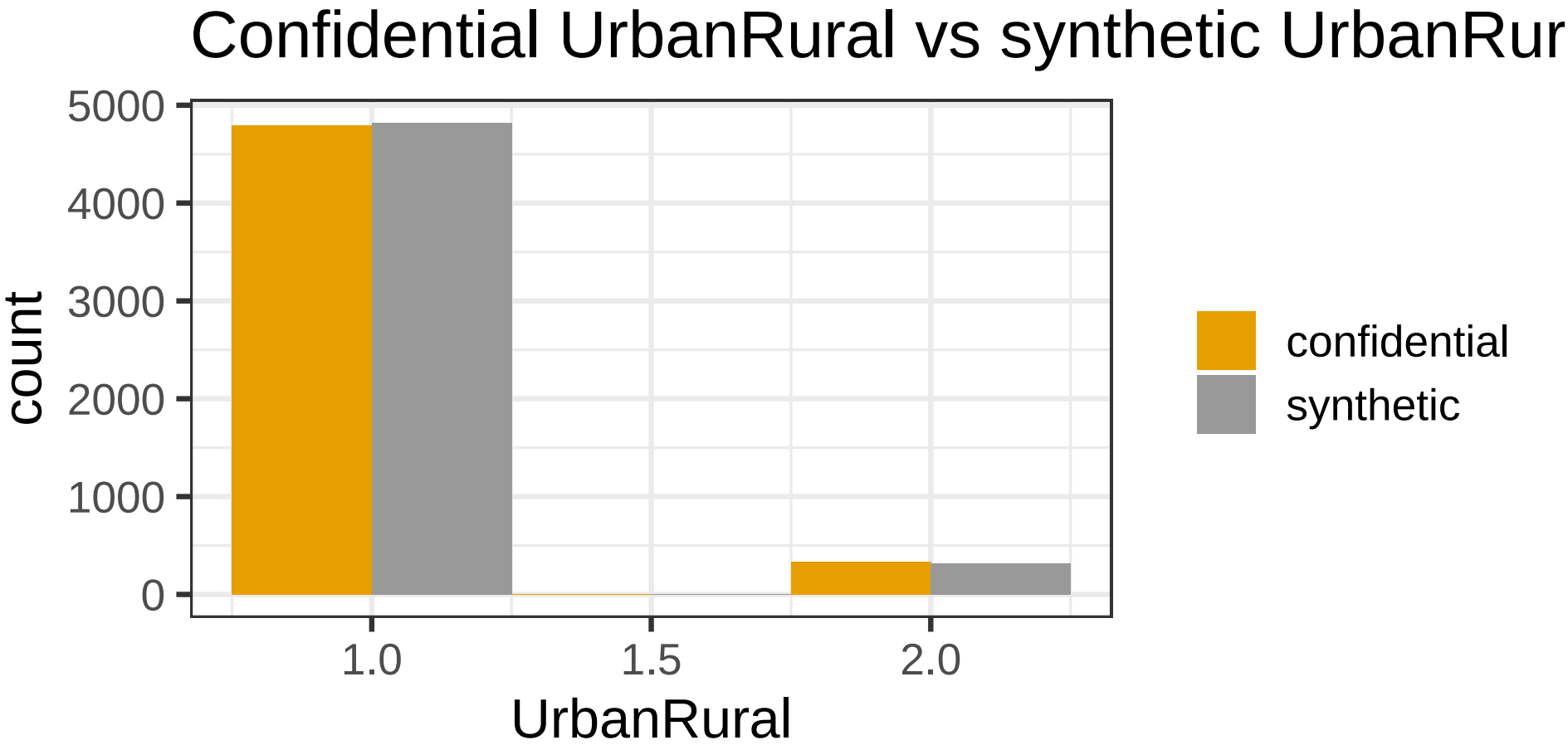
Dicussion question: What are the inputs and outputs of this `SLR_synthesize()` function?

A Bayesian logistic regression: synthesis

- Perform synthesis for the CE dataset

```
n <- nrow(CEdata)
logistic_synthetic_one <- logistic_synthesize(X = logistic_X,
                                              post_draws = post_logistic,
                                              index = 1,
                                              n = nrow(logistic_X),
                                              seed = 902)
names(logistic_synthetic_one) <- c("Intercept", "LogExpenditure",
                                   "UrbanRural")
```

A Bayesian logistic regression: utility check



Outline

Summary

- Synthesizing continuous outcome variables
  - Bayesian simple / multiple linear regressions
- Synthesizing binary outcome variables
  - Bayesian logistic regression
- In both cases
  - Prior (default priors in `brms`)
  - MCMC estimation and diagnostics
  - Posterior predictions for synthetic data generation (writing R functions)
  - Simple utility checks

Summary

- Synthesizing continuous outcome variables
  - Bayesian simple / multiple linear regressions
- Synthesizing binary outcome variables
  - Bayesian logistic regression
- In both cases
  - Prior (default priors in `brms`)
  - MCMC estimation and diagnostics
  - Posterior predictions for synthetic data generation (writing R functions)
  - Simple utility checks

- Homework 4: a few R programming exercises
  - Submission on Moodle and prepare to discuss next time
- Lecture 5: Bayesian synthesis models part 2: categorical, count (sequential synthesis), and DPMPM
  - Section 3 of Kinney et al. (2011)
  - Section 2 of Hu, Reiter, and Wang (2014)

References I

Hu, J., J. P. Reiter, and Q. Wang. 2014. "Disclosure Risk Evaluation for Fully Synthetic Categorical Data." Privacy in Statistical Databases, 185–99.

Kinney, S. K., J. P. Reiter, A. P. Reznek, J. Miranda, R. S. Jarmin, and J. M. Abowd. 2011. "Towards Unrestricted Public Use Business Microdata: The Synthetic Longitudinal Business Database." International Statistical Review 79 (3): 362–84.