

# Modeling Missing Response

Of the 25,397 rides in the data set, 11,365 were not rated. With such a large amount of missing data, careful consideration should be made about what can be inferred from this data set. A common problem with missing responses in crowdsourced rating data sets is that the missingness of ratings is not independent of the ratings that the users would give. This worry motivated Ying, Feinberg, and Wedel's work on creating models for recommendation systems based on online ratings that explicitly modelled missing data<sup>1</sup>. In the case of rides, it's possible that cyclists are more likely to rate their ride if they had a bad experience than if their ride was uneventful. This kind of correlation between missingness and the response can cause strong biases in the estimates, as we will demonstrate.

In this chapter, we attempt to address the missing data issues by fitting a model that simultaneously models the missing data mechanism and the ride ratings. However, with the current state of the ride data, these models may be unable to come up with accurate estimates because of another problem in the data collection. As mentioned in Chapter 1, rides are often misclassified as bike rides when they are actually car rides or rides on public transit. We suspect that many of the unrated rides are rides that were misclassified as bike rides, and thus were not rated by the rider. (We assume that riders don't often go through the effort of correcting the classification of rides and know not to rate rides that weren't bike rides.) If this is the case, then it would be inappropriate to make use of the data with missing responses. If, however, *Ride Report* is able to improve their classification enough to make this a non-issue, these methods could be vital to accurately modeling ride rating.

## What could possibly go wrong?

We focus on the situation we have, where our response variable  $y_i$  has missing values. Define the vector  $R = (r_1, r_2, \dots, r_n)$  such that

$$r_i = \begin{cases} 1, & \text{if } y_i \text{ is missing;} \\ 0, & \text{if } y_i \text{ is observed;} \end{cases} \quad (1)$$

for  $i = 1, \dots, n$

Rubin classifies missing data into three situations<sup>2</sup>:

1. **Missing Completely at Random (MCAR)**, where  $R$  is independent of  $Y$  and the predictors  $X$ . *i.e.*  $\mathbb{P}(R=1|Y, X) = \mathbb{P}(R=1)$
2. **Missing at Random (MAR)**, where  $R$  is independent of  $Y$ , but may depend on  $X$ , *i.e.*  $\mathbb{P}(R=1|Y, X) = \mathbb{P}(R=1|X)$
3. **Nonignorable, or not MCAR nor MAR**, where  $R$  is dependent on  $Y$ .

As discussed in the introduction, we believe that rider ratings may be correlated with nonresponse and thus the missing ratings are non-ignorable.

If missing data is nonignorable, what could go wrong with our models? Let's look at a toy example. Define the data set of  $n$  observations with  $x \in \mathbb{R}^n$ ,  $y \in \{0, 1\}^n$ , and  $R$  defined as before, where

$$\begin{aligned} x_i &\sim \text{Normal}(0, 1), \\ y_i &\sim \text{Bernoulli}(\text{logit}^{-1}(4x_i)), \\ r_i &\sim \text{Bernoulli}(0.3 + 0.4y_i), \end{aligned}$$

for  $i = 1, \dots, n$ .

---

<sup>1</sup>@ying2006

<sup>2</sup>@little1987 (page 14)

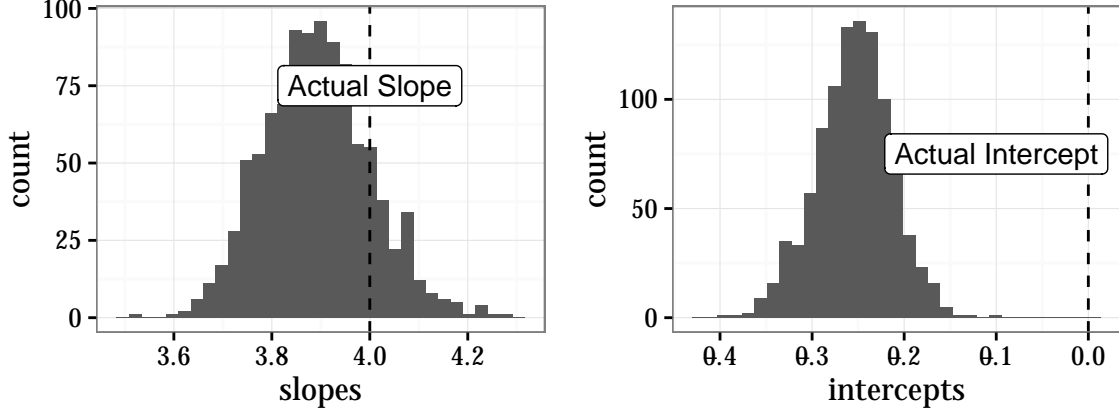


Figure 1: Simulated example of logistic regression fits to a model with nonignorable missing response. One data set of size  $n = 10^4$  was computed from the toy data model. We then recomputed  $R$  1,000 times, each time fitting a simple logistic regression model to  $y$  and  $X$ .

If we attempt to fit a logistic regression model to this data, our slopes will be unbiased but our estimate of the intercept will be biased. Figure 1 shows the results of a simulation, computing the slope and intercepts for 1,000 different patterns of missing data for the same generated data set generated from our toy data model.

It makes sense that we are underestimating our intercept. The intercept can be interpreted as the base rate, and if values of  $y_i = 1$  are more likely to be missing, the overall rate we observe will be lower.

Clearly, if we have nonignorable missing response, we are in a bad situation. Having missingness depend on  $Y$  leads to biased estimates of our intercepts when we fit models. But we do have all of our predictors of  $y$ , with no missingness. Could we leverage our understanding of how  $X$  predicts  $Y$  to understand the patterns of missing response?

## Modeling the Missing Data Mechanism with Expectation Maximization

Here we perform the expectation maximization algorithm using the weighting method proposed by Ibrahim and Lipsitz<sup>3</sup>. Let  $y$  be our binary response and  $X$  be our predictors. With these we have our complete data logistic regression model  $f(y | X, \beta)$ , where  $\beta$  is a vector of parameters in the complete data model.

We then specify a logistic regression model for missingness ( $R$ ):  $f(R | X, y, \alpha)$ , where  $\alpha$  is the vector of parameters in the missingness model.

We begin the algorithm by getting our first estimates of  $\alpha$  and  $\beta$ . We obtain  $\beta^{(1)}$  by estimating  $\beta$  with only the non-missing data. We can then estimate  $y$  for the missing data using  $\beta^{(1)}$ , and then use those estimates to compute  $\alpha^{(1)}$ .

For the E-step, we compute weights for each observation with missing response, representing the probability that the  $i$ th observation has response value  $y_i$ :

$$w_{iy_i}^{(t)} = f(y_i | r_i, x_i, \alpha^{(t)}, \beta^{(t)}) = \frac{f(y_i | x_i, \beta^{(t)})f(r_i | x_i, y_i, \alpha^{(t)})}{\sum_{y_i \in \{0,1\}} f(y_i | x_i, \beta^{(t)})f(r_i | x_i, y_i, \alpha^{(t)})}. \quad (2)$$

For observed responses,  $w_{iy_i}^{(t)} = 1$ . Note that for each observation  $i$ ,  $\sum_{y_i \in \{0,1\}} w_{iy_i} = 1$ . We can compute  $f(y_i | x_i, \beta^{(t)})$  and  $f(r_i | x_i, y_i, \alpha^{(t)})$  by making use of predictions from regression models. So in  $R$ , we can fit models and use the `predict()` function to get our probabilities from each of these models.

<sup>3</sup>@ibrahim1996

For the M-step, we find our next estimates of the parameters,  $\alpha^{(t+1)}$  and  $\beta^{(t+1)}$ , by maximizing

$$Q(\alpha, \beta \mid \alpha^{(t)}, \beta^{(t)}) = \sum_{i=1}^n \sum_{y_i \in \{0,1\}} w_{iy_i}^{(t)} \cdot l(\alpha, \beta \mid x_i, y_i, r_i). \quad (3)$$

We do this by first by estimating  $\beta^{(t+1)}$  using weighted maximum likelihood for the complete data model, and then estimating  $\alpha^{(t+1)}$  using the same method. To maximize  $l(\alpha, \beta \mid x_i, y_i, r_i)$ , we maximize the product of their likelihoods,

$$l(\alpha, \beta \mid x_i, y_i, r_i) = l(\beta \mid x_i, y_i) l(\alpha \mid r_i, x_i, y_i),$$

which we can maximize by maximize each of the likelihoods separately because our estimates of  $\alpha$  and  $\beta$  are only dependent on each other through  $x$  and  $y$ . This allows us to use any package that can fit models by maximum likelihood estimation using weights for the observations, which includes all of the model fitting packages we used in Chapter 3.

In order to create the data to fit these models, we create an augmented data set where each observation missing the response is recorded as two rows. These duplicate rows represent the two possible values of the response, and also contain the weights computed in the E-step. Figure 2 describes this process graphically.

Figure 2: Creation of augmented data set for the weighted method of the EM algorithm for missing response data.

Original Data				Augmented Data			
$y_i$	$x_i$	$r_i$		$y_i$	$x_i$	$r_i$	$w_i$
1	2.4	0	$\rightarrow$	1	2.4	0	1
0	1.3	0		0	1.3	0	1
NA	-0.4	0		1	-0.4	0	0.2
				0	-0.4	0	0.8

We repeat the E and M step until the joint loglikelihood converge to within some tolerance. An example implementation of this algorithm can be found in ??.

As an example, we simulated a dataset from the same model we presented earlier of size  $10^4$ . Of those observations, 6,252 were missing. As shown in Table 1, the estimate for the intercept in the model that only considers the complete data is way off, but the model resulting from the EM algorithm are nearly as accurate as the model computed fit to the full data (with missing values filled in from the original data model.) The missing data model is also able to get accurate estimates of the parameters that define the missing data mechanism, but the estimates are quite uncertain.

Table 1: Coefficients for models fit to simulated data set ( $\pm$  twice the standard error.)

Model	$\hat{\beta}_0$	$2 \cdot SE_{\hat{\beta}_0}$	$\hat{\beta}_X$	$2 \cdot SE_{\hat{\beta}_X}$
Actual	0	—	4	—
Full Data Model	−0.009	0.065	3.881	0.080
Complete Data Model	−0.278	0.106	3.819	0.259
EM Final Model	0.042	0.065	3.814	0.157

## EM Algorithm for the Ride Data

In order to perform the algorithm, we need to specify a model for nonresponse. We will the same predictors that we do in Model 4 for ride rating—including a smoothing spline for time of day for weekdays and

Table 2: Estimates for missing data mechanism.

Model	$\hat{\alpha}_0$	$2 \cdot SE_{\hat{\alpha}_0}$	$\hat{\alpha}_Y$	$2 \cdot SE_{\hat{\alpha}_Y}$
Actual	0.3	—	0.4	—
EM Missing Data Model	0.263	0.132	0.530	0.268

weekends—except we do not use random rider intercepts. For the EM algorithm, we use Model 4 as our ride rating model and use the following model for the rating nonresponse mechanism:

$$r_i \sim \text{Bernoulli}(\text{logit}^{-1}(\alpha_0 + y_i \alpha_y + X_i \alpha_x + \mathcal{S}(t_i))). \quad (4)$$

Table 3: Fit summaries for Model 4 and the EM Model

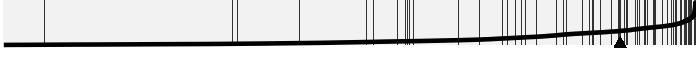

Model	Separation Plot	AUC <sup>4</sup>
Model 4		0.802
EM Model		0.763

Table 4: Estimates for ride rating nonresponse mechanism. The Basic Nonresponse Model is estimated based on the data with  $y$  predicted by Model 4. The EM Nonresponse Model is estimated with the EM algorithm, which uses the same model specifications.

Parameter	Basic Nonresponse Model	EM Nonresponse Model
$y$	0.730 (0.235, 1.224)	1.035 (0.493, 1.577)
Log(Length)	-0.297 (-0.362, -0.232)	-0.327 (-1.163, -0.771)
Mean Temperature	0.200 (0.139, 0.262)	0.139 (0.077, -0.262)
Mean Wind Speed	0.032 (0.003, 0.060)	0.031 (0.001, 0.061)
Max Gust Speed	-0.003 (-0.016, 0.010)	-0.007 (-0.021, 0.006)
Rainfall	0.007 (-0.028, 0.041)	-0.024 (-0.057, 0.009)
Rainfall 4-Hour	-0.002 (-0.012, 0.009)	0.010 (-0.001, 0.021)
Intercept	-0.927 (-1.124, -0.729)	-0.967 (-1.163, -0.771)

Table 5: Ride rating model estimates after EM algorithm

<b>Parameter</b>	<b>Model 4</b>	<b>EM Model</b>
Log(Length)	-0.147 (-0.290, -0.005)	0.205 (-3.603, -2.684)
Mean Temperature	0.142 (0.004, 0.281)	0.100 (0.005, 0.196)
Mean Wind Speed	0.002 (-0.054, 0.057)	-0.026 (-0.069, 0.016)
Max Gust Speed	-0.005 (-0.031, 0.021)	0.020 (0.001, 0.039)
Rainfall	0.050 (-0.017, 0.117)	0.051 (0.009, 0.093)
Rainfall 4-Hour	0.022 (0.003, 0.041)	0.017 (0.003, 0.030)
Intercept	-2.792 (-3.334, -2.250)	-3.144 (-3.604, -2.684)