RESEARCH REPORT SERIES
(Statistics #2015-04)

Selection of Predictors to Model Coverage Errors in the Master Address File

Andrew Raim
Marissa N. Gargano[1]

[1] RTI International

Center for Statistical Research & Methodology
Research and Methodology Directorate
U.S. Census Bureau
Washington, D.C. 20233

Report Issued: December 30, 2015

Selection of Predictors to Model Coverage Errors in the Master Address File

Andrew M. Raim* & Marissa N. Gargano†

Center for Statistical Research and Methodology, U.S. Census Bureau,
Washington, DC, 20233, U.S.A.

Abstract

The U.S. Census Bureau depends on the Master Address File (MAF) to prepare address lists for the decennial census and household surveys. Accuracy of the MAF is critical to these operations. The Census Bureau has considered statistical models to help characterize and predict errors on the MAF. This work follows Young, Raim, & Johnson (Accepted, 2015) and further investigates zero-inflated negative binomial regression to model adds from the 2010 Address Canvassing operation. We consider several supplemental data sources including the Planning Database, the Longitudinal Employer-Household Dynamics data, and land use data, in addition to the database with outcomes from the operation. Collection of the 2010 Address Canvassing data was subject to a variety of influences not captured in the data. These influences include variations in field representative behavior, in-office post-processing of field data, and other operational details not available at the time of data analysis. Therefore, it is not obvious which predictors explain outcomes from the operation, and variable selection is especially critical for this analysis. We carry out an exhaustive variable selection, consisting of forward and backward selection steps, and compare candidate models by several likelihood and prediction-based criteria. This method allows us to consider two-way interactions and to rank predictors by their contribution to the model. Our initial results find that predictors based on missing delivery point type, historical coverage on the Delivery Sequence File, and IRS 1040 forms with no block ID or no MAFID to be among the most useful. The model obtained from the variable selection is shown to fit well to a majority of the blocks, but the relatively small proportion of blocks which do not fit well tend to be those with the most observed adds. Therefore, future research is needed to identify other useful predictors or to permit more heterogeneity within the model. We stress that we are not making recommendations for future Census Bureau operations; our purpose is to obtain a plausible statistical model for MAF coverage error based on the 2010 Address Canvassing outcomes.

Keywords: zero-inflated counts; negative binomial; logistic regression; variable selection; address canvassing.

1 Introduction

The U.S. Census Bureau maintains a database called the Master Address File (MAF) that contains every known residential address in the United States and Puerto Rico. The MAF is used to prepare an address list for the decennial census and for more frequently conducted household sample surveys such as the American Community Survey (ACS) and the Current Population Survey (CPS). Thus, census and sample survey operations at the Census Bureau depend critically on the MAF containing accurate, up-to-date information.

When a housing unit exists in the field but not on the MAF, we say that undercoverage occurs. Alternatively, overcoverage occurs when a housing unit is listed in the MAF but does not actually exist or is not

*Email: andrew.raim@census.gov.

†Now with RTI International.

This paper is released to inform interested parties of ongoing research and to encourage discussion of work in progress. Any views expressed are those of the authors and not necessarily those of the U.S. Census Bureau.

residential. This terminology is taken from survey sampling,[1] and is appropriate because the MAF forms the basis of sampling frames used for sample surveys. We speak of coverage error here at the level of housing units, as opposed to person-level coverage error which is also associated with the census. Undercoverage can cause housing units of interest to be excluded from a survey or census and results to be compromised. Overcoverage can lead to costly nonresponse follow-up work when attempts are made to visit the address and contact inhabitants. Clearly, neither of these outcomes is desirable.

In practice, addresses are never removed from the MAF, and multiple entries can exist for the same housing unit. The Census Bureau regularly produces MAF extracts for usage within the agency; these extracts present a view of active addresses on the MAF where each address is represented by a single entry. Furthermore, consumers of the MAF are usually interested in only a subset of a MAF extract. For example, when drawing a sample of housing units for a survey, only those units eligible for the survey are needed. A set of rules to identify such a subset is called a filter (U.S. Census Bureau, 2014a, Chapter 3). The extracting and filtering processes are necessary to generate usable address lists from the MAF, but are also potential sources of overcoverage and undercoverage.

There are several processes that update the MAF regularly. The Delivery Sequence File (DSF), obtained from the United States Postal Service, provides the majority of city-style addresses in the MAF (Schar et al., 2012). The Postal Service collects DSF data primarily for the purpose of mail delivery and may not always identify the addresses of interest to the Census Bureau. The Census Bureau makes use of other address providers as well, such as local governments through the Local Update of Census Addresses (LUCA) program, and maintains programs such as Demographic Area Address Listing (DAAL) and Community Address Updating System (CAUS), which update addresses on a regular basis. Despite these efforts, the MAF can never completely reflect all habitable addresses in the field. The field is constantly changing, and it is unrealistic to expect that all updates will be caught immediately and recorded in the database without any error or ambiguity.

To prepare the MAF for the 2010 Decennial Census, the Census Bureau invested nearly half a billion dollars in the 2010 Address Canvassing (AdCan) operation. U.S. Census Bureau (2012) reports costs and other logistical results from this operation, which involved 111,105 field representatives (FRs) walking 5.9 million census blocks[2] in the United States and Puerto Rico. A universe of 144.9 million addresses called the "dependent list" was prepared from the MAF to be checked in the field. Each FR was given a hand-held computer containing the dependent list, and was tasked with verifying the addresses and suggesting corrections. If an address was found in the field and was not listed in the MAF (i.e., undercoverage), the address was considered an "add". Similarly, if an address was listed in the MAF but was not found in the field (i.e., overcoverage), the address was considered a "delete." The results of the AdCan operation are listed below in Table 1.1. For this report, we take "New Adds" to be the outcome of our interest. Another action which may be considered to be coverage error is "Matched to Records". This occurred when an FR attempted to add an address but later determined it to already be present in the MAF but not geocoded to a precise location or otherwise excluded from the dependent list. Some modelers within the Census Bureau have summed "New Adds" and "Matched to Records" together as the measurement of undercoverage, but we consider them as two fundamentally different actions and do not make use of "Matched to Records" in this work. We also do not model overcoverage in this report, but the main outcome of interest would be "Does not Exist - Double Delete".

Address canvassing provided assurance that the 2010 Decennial Census would consider all eligible households, but its high cost made it the second most expensive operation of the 2010 Census after nonresponse follow-up (Boies et al., 2012). The majority of housing units were confirmed to be accurately recorded in the MAF prior to AdCan; Table 1.1 shows that approximately 62% of considered housing units were simply verified during the AdCan operation. Furthermore, as discussed in Section 3, of the census blocks in our modeling universe, 77.88% had zero adds, 57.87% had zero deletes, and 50% of blocks had both zero adds and zero deletes. After 2010, the Census Bureau began research to replace some of the in-field canvassing with

[1] Coverage rates definitions. Accessed June 25, 2015. http://www.census.gov/programs-surveys/acs/methodology/sample-size-and-data-quality/coverage-rates-definitions.html.

[2] This block count refers to the 2010 census collection geography, as opposed to 2010 tabulation geography which is used in the remainder of the paper.

Table 1.1: 2010 Address Canvassing operation results. Source: U.S. Census Bureau (2012)

Action	# HUs	% HUs
New Add	6,624,155	4.23
Matched to Records	4,152,739	2.65
Change	19,608,785	12.51
Move	5,450,563	3.47
Verify	97,635,517	62.31
Does not Exist - Double Delete	15,819,921	10.10
Duplicate	4,085,556	2.61
Nonresidential	1,238,260	0.79
Uninhabitable	551,566	0.35
Unduplicated Rejected Records	1,536,094	0.98
Total	156,703,156	100.00

activities that could be carried out from the office in order to avoid some of the cost. One major component of this research has been statistical modeling. Boies et al. (2012) investigate the use of logistic regression models to predict coverage errors and to prioritize the canvassing operation after 2010. By ordering census blocks by the predicted probability of having a substantial amount of coverage error, we can produce a list of blocks most likely to differ from the MAF. A canvassing operation with a reduced in-field workload could deploy FRs to walk only the selected blocks, which could avoid a substantial proportion of cost of the 2010 operation. One caveat of logistic regression is that an event of interest, such as $\{\text{Adds} \geq 1 \text{ or Deletes} > 3\}$, must be determined to characterize "substantial" coverage error observed in address canvassing. Young et al. (2015) instead consider modeling add or delete counts at the census block level using zero-inflated negative binomial (ZINB) and zero-inflated Poisson (ZIP) models. As discussed in Hilbe (2011), such models are used to account for the high prevalence of zero counts. Non-statistical approaches to reduce in-field canvassing are also under consideration at Census Bureau. For example, the Geography Division (GEO) is considering aerial imagery and its ability to detect change over time (U.S. Census Bureau, 2014b).

This report follows Young et al. (2015), and considers an alternative method for variable selection under ZINB regression. Young et al. (2015) follow a procedure that includes screening predictors in the 2010 AdCan database and dropping those with low correlation to the outcome or a high degree of multicollinearity. In this report, we make use of forward and backward variable selection steps that allow us to determine which variables are most useful and which are extraneous or detrimental to the model. The data sources considered include the 2010 AdCan database along with the Planning Database, a dataset describing land use, DSF stability index variables, variables from the Longitudinal Employer-Household Dynamics program, foreclosure counts from the RealtyTrac data, and selected counts of IRS 1040 returns. These data sources are described in Section 2. We carry out selection in the count regression and zero-inflated regression parts of the ZINB model separately, using logistic regression as a surrogate for the zero-inflated part of the model and negative binomial regression as a surrogate for the count part. The forward selection allows us to see, at each step, which variable in the available candidates will yield the greatest improvement to the model. At each backward selection step, we can compare contributions from variables already in the model.

Variable selection is of particular importance in the modeling of address canvassing outcomes. Adds and deletes are obtained through a complicated sequence of field operations and in-office adjudication, and it has not been clear from the onset which variables serve as strong predictors. It is hypothesized that strong predictors would capture both change in the field and an inability to detect that change without canvassing (e.g. because the block has poor DSF coverage). We stress that the models obtained in this paper are not necessarily recommended for operations; rather, to investigate one possible method of variable selection and its ability to identify signal in a large set of candidate predictors.

The Census Bureau recently carried out the 2015 Address Validation Test (AVT) (U.S. Census Bureau, 2015).[3] For the AVT, 10,100 blocks were sampled from the U.S. and fully canvassed. Adds, deletes, and

[3]This work discussed in this paper was conducted outside of the AVT project. Namely, the Title 26 datasets used in the

other canvassing outcomes were recorded, validated, and eventually made available for analysis. The AVT and future tests provide one possible way for models fitted on past data to be evaluated for relevance to more current data and to be updated. This report focuses on the 2010 Address Canvassing operation data, which offers a comprehensive view of MAF coverage error during the months preceding the 2010 Census, and continues to be a natural starting point for modeling efforts. Our first objective is to adequately explain the 2010 Address Canvassing operation outcomes by fitting the 2010 Address Canvassing operation data.

The Census Bureau's consideration to replace or reduce an expensive labor-intensive in-field operation using in-office modeling is not limited to address canvassing. Investigations are also being carried out to replace nonresponse follow-up with administrative record data (Morris et al., 2015) and to improve the efficiency of fieldwork in surveys (Slud and Erdman, 2013). These initiatives have potentially common threads, and might somehow benefit from being considered together in the same light.

There are many questions to be addressed in future MAF modeling efforts. For example, once we arrive at a very good fitting model, the method of selecting or excluding blocks for canvassing may be considered further. One simple idea is illustrated in this report and Young et al. (2015), but this method makes use only of point estimates and ignores the uncertainty expressed in the model. It may also be of interest to move beyond our relatively simple count models to explore issues such as unobserved heterogeneity among blocks, the bivariate relationship between adds and deletes, and additional data sources. There are also important policy questions, such as how to integrate statistical modeling with aerial imagery analysis and other efforts underway at the Census Bureau, and how to determine if a model is "good enough" to be used operationally.

The rest of the report proceeds as follows. Section 2 discusses data sources used in the analysis. Section 3 shows some exploratory analysis on the two response variables, and also provides some insight into the available predictors and their relationships with the response variables. Section 4 discusses the methods used in the count regression analysis. In Section 5, we present a statistical model for adds and discuss in detail how it was obtained. Section 6 concludes the report and suggests future ideas for the modeling effort. Appendix A lists all variables used in the analysis and how they were coded into predictors, and gives some additional tables and figures.

2 Data Sources

The main source of variables for our model is the 2010 AdCan database. The variables considered in this report are listed in Table A.1. This database was prepared by the Decennial Statistical Studies Division (DSSD) of the Census Bureau for the purpose of statistical modeling (Tomaszewski, 2014). It contains block-level variables based on indicator variables at the address level. For example, there are variables that summarize the presence of seasonal housing units within each block. Block level summary variables are given as both sums and means of the address level indicators; we consider only the sum variables in this report. Also, six versions of these variables are given, corresponding to three "positive" filtering criteria: HUs sent out for AdCan, HUs eligible for ACS, geocoded HUs; and three "negative" filtering criteria: HUs not sent out for AdCan, HUs not eligible for ACS, HUs not geocoded. To avoid redundant variables, we considered only variables based on the positive filters. There are also several categorical indicators (e.g. an urban versus rural indicator and a type of enumeration area category) and continuous variables (e.g. measures of land area and water area). From this database, we define our modeling universe as blocks with valid addresses prior to the Address Canvassing operation in the 50 United States and Washington, D.C. This yields a universe of 6,539,119 blocks.

The Census Bureau Planning Database (PDB) contains variables correlated with mail nonresponse (Bruce and Robinson, 2004). The most recent version of the PDB prior to the 2010 Address Canvassing operation is the 2000 PDB, which is based on 2000 census tabulation geography and is publicly available at the tract level. Its variables include various housing attributes (e.g. crowded housing) and person attributes (e.g. language isolation). Based on these characteristics, each tract is assigned a hard-to-count score that explains the degree of enumeration difficulty. Table A.2 shows variables we considered from the PDB. Because our block

present work were not used in research to prepare the official AVT report.

universe is based on 2010 tabulation geography, we allocate the 2000 information to 2010 tracts in the following way. For each tract in the 2000 tabulation geography, we used public geography files[4] to identify all of the tracts from 2010 tabulation geography that intersect with it. Suppose J tracts from 2010, indexed $1, \ldots, J$, have land areas $a_1, \ldots, a_J$ intersecting with the 2000 tract. For each variable in the 2000 PDB, a proportion $a_k / \sum_{j=1}^{J} a_j$ of the 2000 tract's value was "donated" to the kth 2010 tract. In this way, variables for each 2010 tract are assigned as a sum of donations from all intersecting 2000 tracts.

The Land Use data shown in Table A.3 was provided by the Geography Division at Census Bureau. It contains block-level data regarding each block's distance to certain landmark areas and the percentage of physical geographical features on each block, provided by the National Land Cover Database (NLCD) (Homer et al., 2007).

Along with the Land Use data, the Geography Division (GEO) provided us with a "DSF stability index" at the block level. This variable represents an aggregate of the stability of the HUs within each block and is based on data from between spring 2000 and spring 2009. If a HU's stability index is close to 1, this suggests that the address has been present in the DSF for the given time period. This is referred to as "complete coverage" on the DSF. At the block level, a DSF Stability Index close to 1 indicates that the block contains a majority of housing units with complete coverage. Likewise, a DSF Stability Index close to 0 indicates that the block contains a majority of housing units with poor DSF coverage. The variables in this dataset are listed in Table A.4.

The Center for Economic Studies (CES) at Census Bureau is responsible for the Longitudinal Employer-Household Dynamics (LEHD) program, which provides a wealth of publicly available economic data.[5] The LEHD Origin-Destination Employment Statistics (LODES) data provides tract level information on workforce characteristics and growth. CES provided candidate predictors based on public LEHD/LODES data from 2007 and 2008, which is given at the tract level for the 2010 tabulation geography. These variables, listed in Table A.5, are based on residence area characteristics and workplace area characteristics for primary jobs of the workforce.

RealtyTrac is a provider of United States housing, including comprehensive data on home foreclosures.[6] We make use of variables based on counts of foreclosed homes at the block level in 2005, 2006, 2007 and 2008; these are shown in Table A.6.

Table A.7 lists three variables based on IRS records for tax year 2007, which were provided by CES. These variables capture counts of IRS 1040 returns that had no block ID, no MAFID, and both no block ID and no MAFID. This information was not directly available before address canvassing, and had to be computed by taking shares from counts that would have been available at the level of U.S. Postal Service ZIP code. By considering such counts, we hope to indirectly capture the ability of post offices to maintain the DSF for their associated region of mail delivery. It is suspected that some post offices are more effective than others at identifying and recording changes to the DSF, and that presence of IRS 1040 returns without block IDs or MAFIDs could be an indicator of less effective DSF updating. The January 2014 MAF extract was used to calculate each block's share of housing units in each ZIP code, which was used to allocate "no block ID", "no MAFID", and "no block ID and no MAFID" counts from the ZIP code level to the block level.

In order to obtain a predictive model, covariates should be based on information that would have been available before the outcome is observed. To the best of our knowledge, all candidate predictors are based on information which would have been available before the 2010 AdCan operation.

3 Exploratory Analysis

As previously mentioned, 77.88% of the 6,539,119 blocks in our universe had zero adds. Because of this, the data are heavily right-skewed. This is confirmed by the histogram in Figure 3.1, which is truncated to show only blocks with at most 10 adds. Taking the natural logarithm of these outcomes does not produce

[4] 2010 Census Block Relationship Files website. http://www.census.gov/geo/maps-data/data/rel_blk_download.html
[5] LEHD website. http://lehd.ces.census.gov
[6] RealtyTrac website. http://www.realtytrac.com

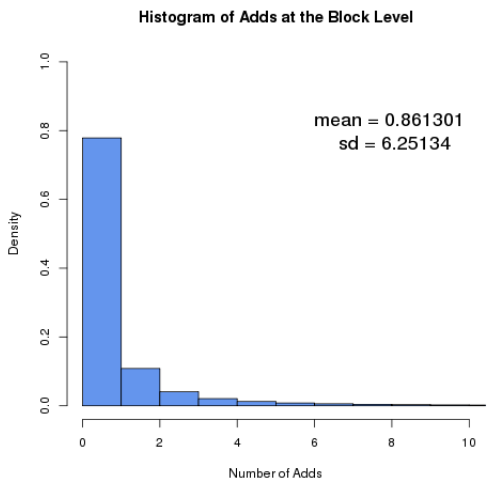


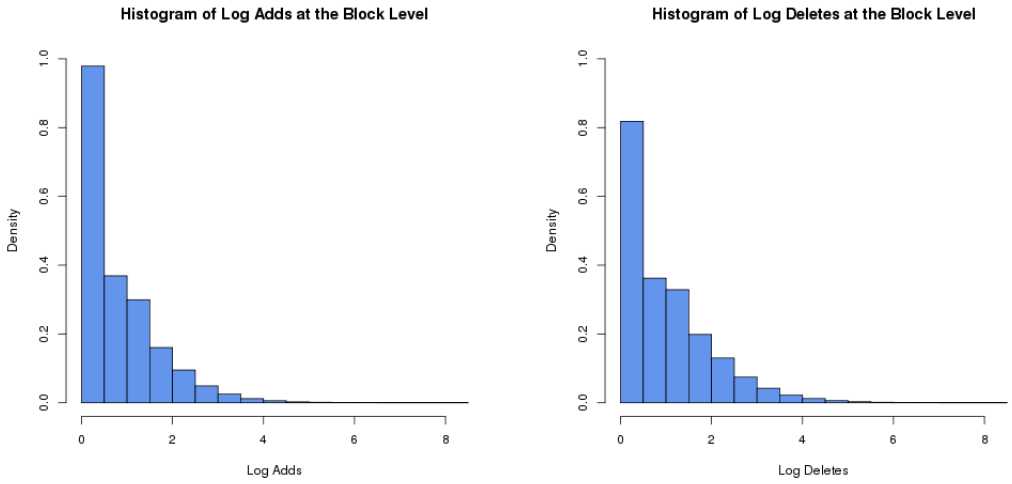
Figure 3.1: Histograms of add counts per block (truncated to 10 or less) from 2010 AdCan operation.

a bell-shaped distribution, as demonstrated in Figure 3.2. The overdispersion and high percentage of zeros suggests modeling the outcomes using a zero-inflated negative binomial distribution, which is discussed in Section 4.

This pattern of overdispersion also occurs regularly among the candidate predictors. For example, in the Planning Database, consider the percent of people unemployed in a given tract (`pct_unemploy_2010`). In Figure 3.3, the histogram of the original variable shows that it is highly right-skewed. However, the log-transformed version (`log_pct_unemploy`) follows a bell-shaped distribution. This transformation was applied liberally to count predictors and helped to avoid models which: (1) did not converge, (2) yielded a Hessian which could not be used to estimate standard errors, or (3) gave estimates with wildly varying magnitudes leading to ridiculously large prediction errors. To avoid taking the logarithm of zero, a small offset was added to most variables; the offset was taken to be 1 for most count variables, and a small increment was used for continuous variables.

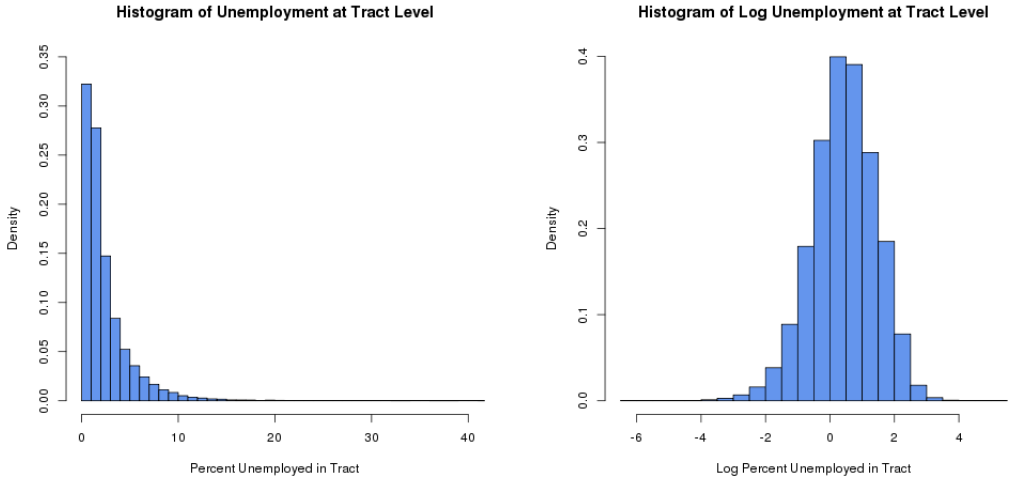
After coding the variables as described in Appendix A, we computed Pearson correlations between the candidate predictors versus `log(adds + 1)` in the hopes of identifying several very strong predictors. These correlations are displayed in Figure 3.4 along with basic summary statistics. The histogram and boxplots also display a 95% confidence interval for the mean and median of the correlations, respectively, which is centered near zero. From the boxplot, we can see that the variable `log_delptypeBk_sum` has a relatively large correlation (0.45) with the log of adds. This variable is a measure of housing units with a blank delivery type. Besides that one instance, it was difficult to identify very strong candidate predictors for adds based on these figures alone. We proceeded with variable selection in the context of statistical models as described in Section 4.

Figure A.2 shows a map of all observed adds from the 2010 Address Canvassing operation. We can see a few interesting patterns from this map. First, it appears that blocks with large adds, colored in blue and dark blue, seem to cluster together. Second, most of these clusters appear to be in major cities across the United States. For example, New York City and Long Island in New York contain a large amount of adds, as well as Philadelphia, Pennsylvania. One major exception is West Virginia, which also appears to contain large amounts of adds. Finally, the eastern half of the country appears to have more adds than the western half of the country. This might be explained by higher population densities in the eastern states.



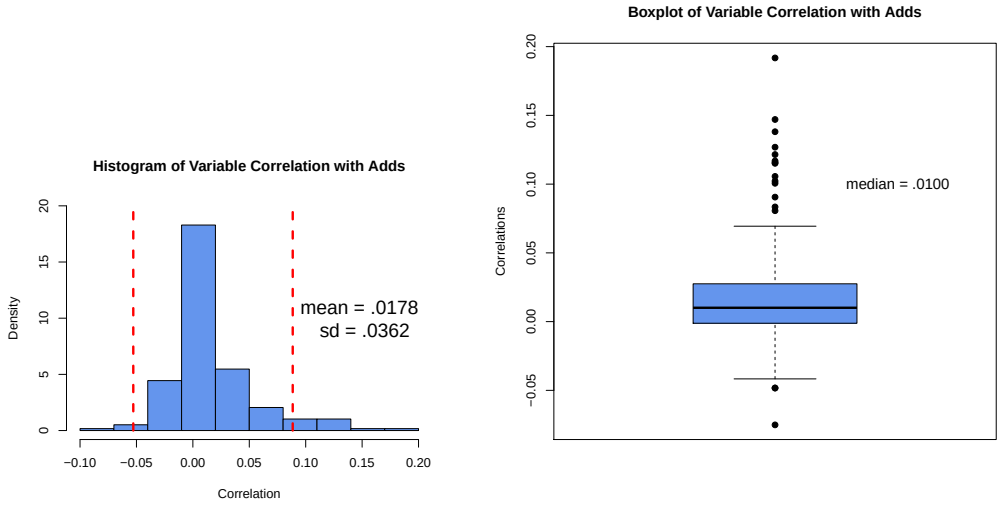
(a) (b)

Figure 3.2: Histograms of log-add and log-delete counts per block (truncated to 10 or less) from 2010 AdCan operation.



(a) (b)

Figure 3.3: Histogram of `pct_unemploy_2010`.



(a) (b)

Figure 3.4: Correlations between candidate predictors and adds.

4 Methodology

Of the 6,539,119 blocks in our modeling universe, we select a systematic sample of 100,000 blocks as the training set for modeling. All model selection and fitting is done on the training set, while the resulting model is evaluated on the entire block universe. This helps to ensure that we are not overfitting to the 2010 AdCan outcomes, and also keeps computational requirements manageable. The training set was selected by a systematic sample using `PROC SURVEYSELECT` in `SAS`. To obtain this sample, we sorted the universe of blocks by state FIPS code, then by number of pre-AdCan HUs within each state from largest to smallest count. After randomly selecting a starting point in the list, the sample is determined by skipping indices at a fixed interval so that a desired number $n = 100{,}000$ blocks are included.

Let $N = 6,539,119$ denote the number of blocks in the universe and $\mathcal{T} \subset \{1, \ldots, N\}$ denote the training set. The add count on the ith block will be denoted as y_i. Following Young et al. (2015), we consider a regression model for $\{y_1, \ldots, y_N\}$ based on the zero-inflated negative binomial (ZINB) distribution. ZINB is commonly used to model count data with large frequency of zeros that cannot be explained only by a count distribution; see Hilbe (2011) for details. We consider the parameterization of the ZINB density given by

$$f(y \mid \mu, \kappa, \pi) = \pi 1_{\{0\}}(y) + (1-\pi)\frac{\Gamma(y + 1/\kappa)}{\Gamma(y+1)\Gamma(1/\kappa)}\frac{(\kappa\mu)^y}{(1+\kappa\mu)^{y+1/\kappa}}, \quad y = 0, 1, \ldots \tag{4.1}$$

where $\mu > 0$, $\kappa > 0$, $\pi \in (0, 1)$, and 1_A represents the indicator function on the set A (i.e. $1_A(x) = 1$ if $x \in A$ and $1_A(x) = 0$ otherwise). To denote that a random variable Y is drawn from $f(y \mid \mu, \kappa, \pi)$, we will write $Y \sim \text{ZINB}(\mu, \kappa, \pi)$. In this case, $\text{E}(Y) = (1-\pi)\mu$ and $\text{Var}(Y) = (1-\pi)\mu\{1 + \mu(\kappa + \pi)\}$. When $\pi \to 0$, (4.1) becomes the negative binomial distribution which we will write as $Y \sim \text{NB}(\mu, \kappa)$. The Poisson($\mu$) distribution is a special case of NB(μ, κ) where $\kappa \to 0$. We will write $Z \sim \text{Ber}(\pi)$ for a random Z drawn from a Bernoulli distribution with probability of success π. The density (4.1) can be obtained by finding the marginal distribution of Y when $Z \sim \text{Ber}(\pi)$ and

$$Y \sim \begin{cases} \text{NB}(\mu, \kappa) & \text{if } Z = 0, \\ 0 & \text{if } Z = 1. \end{cases}$$

Here, $Z = 1$ represents a latent state where zero is always observed (i.e. a block is stable and no HUs will be added), and $Z = 0$ represents a latent state where a count (i.e. zero or more adds) could be observed. Consider a d_1-dimensional covariate $\boldsymbol{x}$ to be linked to the count mean μ and a d_2-dimensional covariate $\boldsymbol{w}$ to be linked to the probability of systematic zero π. A corresponding regression model can be written as

$$Y \sim \text{ZINB}(\mu, \kappa, \pi), \quad \log(\mu) = \boldsymbol{x}^T\boldsymbol{\beta} + \log t, \quad \text{logit}(\pi) = \boldsymbol{w}^T\boldsymbol{\gamma}, \tag{4.2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{d_1}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{d_2}$. The term $t > 0$ is an offset or exposure which facilitates interpretation by a count rate $\mu/t = \exp\{\boldsymbol{x}^T\boldsymbol{\beta}\}$. Suppose our data consists of $\{(y_i, \boldsymbol{x}_i, \boldsymbol{w}_i, t_i) : i \in \{1, \ldots, N\}\}$, and assume model (4.2) for each y_i independently. We can formulate the likelihood of the training set as

$$L(\boldsymbol{\beta}, \kappa, \boldsymbol{\gamma}) = \prod_{i \in \mathcal{T}} f\left(y_i \,\middle|\, \mu_i = \exp\{\boldsymbol{x}_i^T\boldsymbol{\beta} + t_i\}, \kappa, \pi_i = \frac{1}{1 + \exp\{-\boldsymbol{w}_i^T\boldsymbol{\gamma}\}}\right).$$

ZINB is a widely used model and off-the-shelf software packages are available to carry out estimation by maximum likelihood. We make use of the `COUNTREG` procedure in SAS to fit ZINB. The goal for this work is to determine a reasonable $\boldsymbol{x}$ and $\boldsymbol{w}$ from the currently available data sources to predict y, or to ascertain that no satisfactory $\boldsymbol{x}$ and $\boldsymbol{w}$ exist. For the remainder of the paper, we assume a ZINB model as our eventual goal and focus on the selection of predictors. The idea is that, once a reasonable set of predictors is determined, future work could change or relax the model assumptions (e.g. by considering more complex models) to perhaps make better use of the data (e.g. through dimension reduction or alternative codings of the predictors).

We carry out variable selection using forward and backward selection steps with customized code that we have implemented in `R`. We found programming such a procedure to be more natural in `R` than in `SAS`. We also found fitting a negative binomial or logistic regression in `R` to be much faster and more reliable than fitting a ZINB model in `R` with available packages. Therefore, variable selection is split into two phases. One phase takes y to be the outcome and makes use of negative binomial regression; this corresponds to the latent state where a count is observed. The other phase takes the event $[y = 0]$ to be the outcome and makes use of logistic regression. This corresponds to the state where systematic zeros are observed. In each phase, we consider two kinds of steps.

Add1. An Add1 step starts with an initial model and adds one candidate predictor at a time. The resulting model fits can then be compared side-by-side to decide which candidate, if any, should be added. Let $\boldsymbol{x} = (x_1, \ldots, x_p)$ be the covariates in the initial model and let $\boldsymbol{x}^* = (x_1^*, \ldots, x_q^*)$ be candidates which could be added. Fit q models using

$$(x_1, \ldots, x_p, x_1^*), \quad \ldots, \quad (x_1, \ldots, x_p, x_q^*),$$

plus the initial model using $\boldsymbol{x}$. For each model, compute the log-likelihood, Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), sum of squared prediction errors (SSPE), and sum of absolute prediction errors (APE). The models can then be compared by these criteria and the most helpful x_j^* can be added to the initial model in subsequent steps. We found that some x_j^* can improve the likelihood but can also have a detrimental effect on prediction error. We avoid automating the selection process and manually examine the criteria from the $q + 1$ model fits. As a rule of thumb, we sort the fits by log-likelihood, and select the variable which most improves the log-likelihood while also taking care to keep the prediction error as small as possible.

Drop1. A Drop1 step starts with an initial model and drops one predictor at a time. The resulting model fits can then be compared side-by-side to decide which predictor, if any, should be dropped. Let $\boldsymbol{x} = (x_1, \ldots, x_p)$ be the covariates in the initial model. Fit p models using

$$(x_2, x_3, \ldots, x_p), \quad (x_1, x_3, \ldots, x_p), \quad \ldots, \quad (x_1, x_2, \ldots, x_{p-1}),$$

plus the initial model using $\boldsymbol{x}$. For each model, compute the same criteria as in the Add1 step. Our rule of thumb is to drop the variable whose absence is the least detrimental to the log-likelihood, and where absence of the variable also does not have a detrimental effect on prediction error. If no such variables are present in the model, no change is made. A Drop1 step is useful for ranking the utility of predictors in a given model.

Add1 and Drop1 steps are sequenced together by the analyst to arrive at a satisfactory model; that is, a model where all candidate predictors have been considered, and where no variables in the model are extraneous or detrimental. This method of variable selection helps the analyst develop a strong intuition about available predictors, but can be tedious as it requires frequent manual intervention. Future work should consider automated methods which give the same kind of intuition. Note that all variable selection is carried out on the training set $\mathcal{T}$ to protect against overfitting.

The prediction error criteria are defined as

$$\text{SSPE} = \sum_{i \in \mathcal{S}} (y_i - \hat{y}_i)^2 \quad \text{and} \quad \text{APE} = \sum_{i \in \mathcal{S}} |y_i - \hat{y}_i|,$$

where $\hat{y}_i$ are predictions obtained from the model and $\mathcal{S}$ is some subset of the data (e.g. the model universe or the training set). We also define the mean-square prediction error (MSPE) and the mean absolute prediction error (MAPE) by dividing SSPE and APE, respectively, by the number of observations in $\mathcal{S}$. We do not consider statistical significance in our battery of criteria. In our experience, predictors with significant coefficients have been extraneous or detrimental to prediction error; on the other hand, nonsignificant coefficients have led to improved predictions.

A variable's utility or detriment can change dramatically given other variables in the model. For example, two predictors may explain roughly the same variability in the outcome. Predictors may also be dependent on each other, causing a potential collinearity problem. An interaction between two or more predictors may be a useful covariate, and its utility may further depend on the presence or absence of other covariates. The Add1/Drop1 framework allows two-way interactions and alternative codings of predictors (placing into categories, log-transforming, etc) to be considered without additional effort. We consider all pairs of two-way interactions between candidate predictors whose main effects are included in the model.[7]

To detect the presence of a multicollinearity problem, we consider the Generalized Variance Inflation Factor (GVIF) proposed by Fox and Monette (1992) and used by Young et al. (2015). Suppose predictors $(x_1, \ldots, x_p)$ correspond to estimated coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)$ and are partitioned into G groups by the analyst. As an example, for a categorical variable split into dummies, we may want to consider their collinearity as a unit. Trivially, each predictor can belong to its own group. Let $\hat{\boldsymbol{R}}$ denote the estimated correlation matrix of $\hat{\boldsymbol{\beta}}$. The GVIF for group g, for $g = 1, \ldots, G$, is computed as

$$\det(\hat{\boldsymbol{R}}_1) \det(\hat{\boldsymbol{R}}_2) / \det(\hat{\boldsymbol{R}}),$$

where $\hat{\boldsymbol{R}}_1$ contains the rows and columns of $\hat{\boldsymbol{R}}$ corresponding to coefficients in group g, and $\hat{\boldsymbol{R}}_2$ contains the remaining rows and columns. As with the traditional Variance Inflation Factor, a large GVIF indicates a problem with multicollinearity. The GVIF reflects the relationship between the volumes of two confidence regions for $\hat{\boldsymbol{\beta}}$: the estimated region versus an ideal region. A GVIF of 1 indicates no evidence of a problem due to multicollinearity.

Also following Young et al. (2015), we consider the randomized quantile residuals proposed by Dunn and Smyth (1996). Suppose Y_i is drawn independently from cumulative distribution function (cdf) $F_{\boldsymbol{\theta}_i}$ for $i = 1, \ldots, n$. Denote Φ^{-1} as the quantile function for $N(0, 1)$ and let

$$F_i^{(L)} = \lim_{z \uparrow Y_i} F_{\hat{\boldsymbol{\theta}}_i}(z) \quad \text{and} \quad F_i^{(U)} = F_{\hat{\boldsymbol{\theta}}_i}(Y_i).$$

[7] If an interpretable model is desired, the analyst can choose to follow common best practices for adding and dropping predictors. For example, if a two-way interaction is present in the model, it is usually suggested for both main effects to be present. The analyst may also wish to avoid dropping dummies that compose a categorical predictor. In this work, we are more interested in prediction than interpretability.

where $\hat{\boldsymbol{\theta}}_i$ is an estimate of $\boldsymbol{\theta}_i$. The residual is computed as $r_i = \Phi^{-1}(U_i)$ using a draw U_i from the uniform distribution on the interval $(F_i^{(L)}, F_i^{(U)})$. When $F_{\boldsymbol{\theta}_i}$ is a continuous cdf, $F_i^{(L)} = F_i^{(U)}$ and the residual simplifies to $r_i = \Phi^{-1}(F_{\hat{\boldsymbol{\theta}}_i}(Y_i))$. The idea for these residuals is that, if the estimated models $F_{\hat{\boldsymbol{\theta}}_i}$ fit well, then $r_1, \ldots, r_n$ should be distributed approximately as a random sample from $\mathrm{N}(0, 1)$. In practice, the residuals can be used to diagnose the fit of the model by checking boxplots, Normal Q-Q plots, and so on. The randomness of the residuals may be seen as a potential downside, as their values partially depend on the state of a random number generator. It can also be seen as an advantage, providing some jitter and leading to more interpretable plots when analyzing discrete data.

5 ZINB Model for Adds

In this section we obtain a ZINB model for adds using the set of candidate predictors introduced in Section 2. Variables are selected using the Add1/Drop1 method discussed in Section 4. First, Section 5.1 considers the event $[y = 0]$ as the outcome and selects variables for a logistic regression model. Section 5.2 selects variables for a negative binomial regression model using y as the response. These two results are combined in Section 5.3 to obtain a single ZINB regression model: the zero-inflated regression using the predictors obtained in Section 5.1, and the count regression using the predictors obtained in Section 5.2. The ZINB model is then evaluated for goodness-of-fit to the training dataset and its ability to extrapolate to the full block universe in Section 5.4.

5.1 Bernoulli Selection

Model selection began with an initial set of variables from the AdCan database, listed in Table A.8, based on prior modeling experience. The table shows the fit for the initial regression model, including the estimates for the coefficients and associated standard error, z-statistics, and p-values. Starting from this initial model, Add1 and Drop1 steps were taken as described in Table 5.1. First, candidate predictors from the AdCan DB were considered. Next, variables from the supplementary data sources described in Section 2 were considered as candidates. Finally, all possible two-way interactions from the selected predictors were considered as candidates. Note that, if a two-way interaction was selected in one step, its interaction with a third variable would be considered in a subsequent step; in this way, higher order interactions were considered as well.

The first row of Table 5.1 shows that we first ran a Drop1 step on our initial model. This process dropped every variable from the initial model in turn and computed the log-likelihood, AIC, BIC, SSPE and APE. The output was displayed in a table with rows corresponding to models with one predictor dropped, sorted by log-likelihood in ascending order. From this output, we noted that dropping `log_acs_hu_ratio` led to a slightly simpler model whose fit was almost equivalent. The predictor `log_acs_hu_ratio` was therefore dropped, followed by several other predictors in subsequent steps. In the fourth step, an Add1 step helped us to determine that the candidate `log_mafsrc1_sum` should be added to the model.

Table 5.2 shows details for a Drop1 step using predictors selected to be in the final model; this table was computed after Table 5.1. We also list the GVIF for each variable, excluding the full model and the intercept. We can see, for example, that dropping `log landmeters2` yields a very similar fit to the full model. However, we decided to keep `log landmeters2` because the quadratic term `log_landmeters2_sq` is also present in the model, and is providing a more substantial improvement to the fit. We see that `log_delptypeBk_sum` appears to be the most helpful predictor; if dropped, there is a significant degrade in fit. The set of predictors notated as `log_devel*_pct` is associated with a high GVIF. This was noted in case problems were encountered later in the model building process. For completeness, the estimates for the final model are listed in Table A.9.

5.2 Negative Binomial Selection

Results for variable selection of the negative binomial regression model are analogous to those in Section 5.1. Table A.10 shows the initial model selected from the AdCan database from previous experience. Table 5.3

Table 5.1: Bernoulli Add1/Drop1 selection with AdCan DB variables.

	AdCan DB Steps	LogLik	AIC	BIC	SSPE	APE
0	Initial	-43815.61	87663.22	87815.43	13846.82	27670.01
1	Drop log_acs_hu_ratio	-43815.76	87661.52	87804.21	13846.78	27669.60
2	Drop log_gc_sum	-43816.05	87660.09	87793.27	13846.44	27669.61
3	Drop log_business_sum	-43816.95	87659.91	87783.57	13845.82	27669.11
4	Add log_mafsrc1_sum	-43684.80	87397.60	87530.78	13806.06	27579.27
5	Add log_compcity1_sum	-43649.36	87328.73	87471.42	13793.74	27549.87
	Supplemental Steps	LogLik	AIC	BIC	SSPE	APE
1	Add log_forest*_pct	-43403.55	86843.09	87014.33	13702.57	27347.94
2	Add log_irs1040ng	-43147.98	86333.96	86514.71	13613.40	27176.52
3	Add log_pct_crowd_occp_u	-42996.04	86032.09	86222.35	13565.01	27085.40
4	Add log_crops_pct	-42894.10	85830.21	86029.98	13527.67	27011.14
5	Add log_dsf_si_spr09	-42812.80	85669.60	85878.89	13503.96	26949.49
6	Add log_shrub_pct	-42749.23	85544.47	85763.27	13481.73	26901.20
7	Add log_devel*_pct	-42701.94	85457.89	85714.74	13466.32	26873.57
8	Add stability_index	-42662.65	85381.30	85647.66	13454.58	26853.09
9	Add hu_block2tract_ratio	-42636.46	85330.91	85606.79	13445.67	26838.67
10	Add log_pct_pop_0_17	-42611.10	85282.20	85567.59	13436.57	26822.35
11	Add log_irs1040nb	-42542.48	85146.95	85441.86	13409.99	26774.47
12	Add log_irs1040nm	-42466.18	84996.35	85300.77	13386.86	26729.95
13	Add log_htc	-42427.21	84920.42	85234.35	13374.92	26709.55
14	Add log_pct_mlt_u_10p_str	-42403.57	84875.15	85198.59	13368.77	26696.56
15	Add log_pct_not_single_u_strc	-42378.69	84827.39	85160.34	13360.71	26681.74
16	Add log_pct_black	-42356.64	84785.28	85127.75	13352.51	26665.80
17	Drop log_hu_density_ratio	-42356.64	84783.29	85116.24	13352.50	26665.83
	Interaction Steps	LogLik	AIC	BIC	SSPE	APE
1	Add I1	-42222.04	84516.08	84858.55	13305.19	26562.30
2	Add I2	-42145.33	84364.66	84716.64	13276.19	26509.46
3	Add I3	-42070.75	84217.50	84578.99	13244.43	26437.04
4	Add I4	-42008.87	84095.73	84466.73	13226.09	26404.02
5	Add I5	-41946.73	83973.47	84353.98	13202.67	26363.23
6	Add I6	-41908.02	83898.05	84288.08	13190.87	26339.09
7	Drop urbanZERO	-41908.02	83898.05	84288.08	13190.87	26339.09
8	Drop teaUER	-41912.09	83902.17	84273.17	13191.83	26341.53

Variable/Group Definitions

I1: log_compcity1_sum:log_devel1_pct
I2: log_dep_list:log_dsf_si_spr09
I3: log_landmeters2:log_dsf_si_spr09
I4: log_delptypeBk_sum:log_dsf_si_spr09
I5: log_dsf_si_spr09:log_irs1040nm
I6: log_devel2_pct:log_irs1040nb
log_forest*_pct: log_forest1_pct, log_forest2_pct, log_forest3_pct
log_devel*_pct: log_devel0_pct, log_devel1_pct, log_devel2_pct, log_devel3_pct

Table 5.2: Drop1 for final selected Bernoulli model.

Drop	LogLik	AIC	BIC	SSPE	APE	GVIF
<FULL MODEL>	-41912.09	83902.17	84273.17	13191.83	26341.53	---
log_landmeters2	-41912.15	83900.30	84261.79	13191.93	26341.54	8.3035
log_irs1040nm	-41912.17	83900.33	84261.82	13191.78	26341.46	2.5217
log_compcity1_sum	-41916.20	83908.40	84269.89	13192.75	26346.62	15.2929
hu_block2tract_ratio	-41917.14	83910.27	84271.77	13194.03	26346.33	2.7258
hasSeasonalY	-41919.59	83915.18	84276.67	13194.58	26347.93	1.0476
log_unitstat1_sum	-41920.94	83917.88	84279.37	13193.28	26342.98	19.9653
teaMOM	-41929.74	83935.48	84296.97	13197.91	26351.72	1.7314
log_dep_list	-41930.84	83937.69	84299.18	13195.97	26352.39	17.1194
log_pct_not_single_u_strc	-41931.13	83938.26	84299.75	13198.36	26352.72	11.7245
log_htc	-41932.17	83940.34	84301.83	13197.33	26352.83	7.6980
log_pct_mlt_u_10p_strc	-41939.81	83955.62	84317.11	13198.17	26353.99	2.6670
log_pct_black	-41940.83	83957.66	84319.15	13202.76	26361.12	1.7512
log_shrub_pct	-41949.37	83974.74	84336.23	13204.72	26369.67	1.4111
I1	-41949.57	83975.13	84336.62	13203.23	26364.63	11.1165
log_forest*_pct	-41954.51	83981.02	84323.49	13203.95	26371.48	2.4973
Intercept	-41955.00	83986.01	84347.50	13205.35	26383.00	---
log_devel*_pct	-41961.01	83992.02	84324.97	13206.73	26373.57	95.6188
I2	-41963.58	84003.16	84364.66	13208.29	26379.30	9.6059
log_mafsrc1_sum	-41965.68	84007.36	84368.85	13208.85	26375.61	2.1558
log_landmeters2_sq	-41969.28	84014.56	84376.06	13213.10	26367.67	4.7501
I3	-41970.24	84016.47	84377.96	13213.48	26379.76	18.4979
log_isVacantY_sum	-41971.26	84018.52	84380.01	13215.65	26388.27	1.5412
I4	-41980.05	84036.10	84397.59	13212.25	26377.59	6.9441
log_eds_res_sum	-41994.06	84064.12	84425.62	13213.91	26393.10	1.8156
log_pct_pop_0_17	-41994.56	84065.12	84426.61	13219.72	26391.34	6.7107
log_crops_pct	-41996.13	84068.26	84429.75	13224.24	26407.01	1.8849
I5	-41998.72	84073.43	84434.92	13225.13	26407.14	26.4756
log_irs1040nb	-42001.31	84078.61	84440.10	13223.37	26398.75	2.0666
stability_index	-42004.05	84084.10	84445.59	13223.06	26399.24	2.7371
I6	-42041.89	84159.79	84521.28	13241.89	26453.13	6.4004
log_pct_crowd_occp_u	-42048.89	84173.78	84535.27	13237.98	26428.97	1.9901
log_irs1040ng	-42056.02	84188.03	84549.52	13246.17	26442.34	1.8875
log_dsf_si_spr09	-42255.39	84586.77	84948.26	13312.72	26582.05	56.2278
log_delptypeBk_sum	-42323.42	84722.85	85084.34	13323.03	26629.75	19.9653

Variable/Group Definitions

I1: log_devel2_pct:log_irs1040nb
I2: log_compcity1_sum:log_devel1_pct
I3: log_dsf_si_spr09:log_irs1040nm
I4: log_delptypeBk_sum:log_dsf_si_spr09
I5: log_dep_list:log_dsf_si_spr09
I6: log_landmeters2:log_dsf_si_spr09
log_forest*_pct: log_forest1_pct,log_forest2_pct,log_forest3_pct
log_devel*_pct: log_devel0_pct, log_devel1_pct, log_devel2_pct, log_devel3_pct

describes the Add1/Drop1 selection process for the negative binomial model. Table 5.4 shows a Drop1 step for the final selected model along with GVIFs. Estimates for the final model are listed in Table A.11. In our experience, the negative binomial model was much more prone to the problem of added variables giving a large improvement to the log-likelihood but causing a detrimental effect to prediction error. For example, selecting `log_delptypeBk_sum` as a predictor increased SSPE by an order of magnitude (not shown). If we instead coded the underlying variable `dpreac_a9_delptypeBk_sum` as the indicator `has_delptypeBk`, we obtained a large improvement to the log-likelihood without the increase in prediction error. Indeed, this turned out to be the most useful predictor according to Table 5.4.

5.3 Zero-Inflated Negative Binomial Model

A zero-inflated negative binomial model was obtained by taking the final model from Section 5.1 as the covariate $\boldsymbol{w}$ in the ZINB regression model (4.2), and by taking the final model from Section 5.2 as the covariate $\boldsymbol{x}$. We used the Newton-Raphson algorithm in `PROC COUNTREG`, and checked that the algorithm converged with a positive definite Hessian (so that covariances could be estimated). Estimates for the count regression coefficients and dispersion parameter are listed in Table 5.6, while Table 5.7 shows estimates for the zero-inflation coefficients. Each table gives standard errors, p-values and 95% confidence intervals along with the estimates.

There are a few interesting things to note in Table 5.6. First, we can see that the variable with the strongest influence on mean add count is `stability_index`. This variable represents a measure of consistency for a block's DSF history. Recall from Section 2 that values closer to 1 suggest HUs on the block have complete and accurate histories in the DSF, while values closer to 0 suggest poor DSF coverage. Our estimate of -0.8793 in the count regression suggests that higher values of `stability_index` are associated with a smaller mean count, which is consistent with our intuition. We also note that certain variables, including `teaMOM`, `log_unitstat1_sum`, `log_forest2_pct` and `log_irs1040nm`, are not significant at the 5% level. We did not exclude these variables from the model because they were seen to improve the log-likelihood and prediction error in Section 5.2.

From Table 5.7, we see `hu_block2tract_ratio` has the largest influence on the probability of systematic zero for a block. This seems intuitive because this variable is measuring the ratio of pre-AdCan HUs in a block to the count in the tract. If this ratio is high, we might expect to see an area with more activity than its surrounding areas, and therefore more adds. Our estimate of -4.4435 suggests that the probability of a systematic zero decreases for larger values of the ratio, which is consistent with our intuition. As in Table 5.6, there are several variables that are not significant at the 5% level, including `log_crops_pct`, `log_devel0_pct`, `log_irs1040nm`, and the interaction term `log_devel1_pct:log_compcity1_sum`.

5.4 ZINB Model Evaluation

Table 5.8 shows the model fit statistics for our zero-inflated adds model. The log-likelihood, AIC and BIC were computed using the training set, and thus cannot be used for direct comparison with other models unless the same training set is used. Prediction errors were computed, however, using all blocks in the universe, facilitating direct comparison against other models. Fit statistics for negative binomial and Poisson regression models are shown as well, using the same predictors in the count regression as ZINB. We can see that negative binomial is fairly close to ZINB in terms of log-likelihood, AIC, and BIC, while Poisson is worse by an order of magnitude. Alternatively, the Poisson SSPE and APE rivals ZINB. The decreased magnitude of SSPE and APE for Poisson was due to the distribution's tendency to produce smaller predictions and not because it produces more accurate predictions than negative binomial.

Figure 5.1 shows plots of the randomized quantile residuals computed on the training set. The Q-Q plot shows that our model is not capturing extreme values well. Furthermore, the plot of the residuals vs. log-predictions shows a trend where the residuals become larger in magnitude as log-prediction approaches zero. Figure A.1 shows quantile residuals for the negative binomial and Poisson models, respectively, for comparison. It is apparent from these plots that the negative binomial model fit is comparable to ZINB while Poisson is notably worse.

Table 5.3: NegBin Add1/Drop1 selection with AdCan DB variables.

	AdCan DB Steps	LogLik	AIC	BIC	SSPE	APE
0	Initial	-88685.72	177405.4	177567.2	2241029	109489.5
1	Add log_mafsrc2_sum	-88683.88	177403.8	177575.0	2212585	109220.2
	Supplemental Steps	**LogLik**	**AIC**	**BIC**	**SSPE**	**APE**
1	Add stability_index	-87993.59	176023.2	176194.4	2322641	108110.0
2	Add log_irs1040ng	-87719.45	175476.9	175657.6	2212165	107215.4
3	Add log_irs1040nb	-87480.24	175000.5	175190.7	2187939	106409.2
4	Add log_devel*_pct	-87366.96	174781.9	175010.2	2151479	105895.5
5	Add log_crops_pct	-87205.61	174461.2	174699.1	2146343	105429.5
6	Add log_pct_crowd_occp_u	-87101.30	174254.6	174501.9	2131212	104890.0
7	Add log_pct_pop_0_17	-87034.80	174123.6	174380.5	2131989	104932.3
8	Add log_pct_not_single_u_strc	-86944.05	173944.1	174210.5	2122462	104622.0
9	Add log_forest*_pct	-86898.75	173859.5	174154.4	2108162	104299.6
10	Add log_dsf_si_spr00	-86830.08	173724.2	174028.6	2124208	104610.7
11	Add log_shrub_pct	-86780.09	173626.2	173940.1	2123697	104602.3
12	Add log_dsf_si_spr09	-86699.76	173467.5	173791.0	2180394	105472.2
13	Add pct_unemploy_zero	-86658.59	173387.2	173720.1	2167083	105419.5
14	Add log_pct_li_hh_indo_europe	-86608.32	173288.6	173631.1	2166127	105399.7
15	Add log_irs1040nm	-86563.72	173201.4	173553.4	2165367	105140.1
16	Add log_pct_mlt_u_2p_strc	-86522.99	173122.0	173483.5	2173598	105081.9
17	Add realtrac_*_2007	-86472.60	173027.2	173417.2	2189715	105348.9
18	Add log_pct_api	-86442.92	172969.8	173369.4	2193304	105528.8
19	Add uni_dist*	-86411.79	172919.6	173376.2	2198854	105687.2
20	Drop log_acs_hu_ratio	-86412.05	172918.1	173365.2	2199715	105703.0
21	Drop uni_dist3	-86412.32	172916.6	173354.2	2200831	105712.8
22	Drop urbanZERO	-86412.91	172915.8	173343.9	2201206	105687.4
23	Drop realtrac_6_10_2007	-86413.60	172915.2	173333.8	2201698	105677.6
24	Drop uni_dist5	-86414.42	172914.8	173323.9	2201086	105675.5
25	Drop uni_dist1	-86416.15	172916.3	173315.8	2201842	105665.4
26	Drop uni_dist4	-86419.00	172920.0	173310.0	2200246	105649.4
	Interaction Steps	**LogLik**	**AIC**	**BIC**	**SSPE**	**APE**
1	Add I1	-86208.79	172501.6	172901.1	2208588	105394.5
2	Add I2	-86118.12	172322.2	172731.3	2204527	104928.7
3	Add I3	-86031.04	172150.1	172568.7	2195509	105283.9
4	Add I4	-85970.86	172031.7	172459.8	2116908	104301.2

Variable/Group Definitions

I1: log_dep_list:log_devel1_pct
I2: log_landmeters2:log_dsf_si_spr00
I3: log_unitstat1_sum:log_hu_density_ratio
I4: log_eds_res_sum:stability_index
log_devel*_pct: log_devel0_pct, log_devel1_pct, log_devel2_pct, log_devel3_pct
log_forest*_pct: log_forest1_pct, log_forest2_pct, log_forest3_pct
realtrac_*_2007: realtrac_1_5_2007, realtrac_6_10_2007, realtrac_11plus_2007
uni_dist*: uni_dist0, uni_dist1, uni_dist2, uni_dist3, uni_dist4, uni_dist5

Table 5.4: Drop1 for final selected Negbin model.

Drop	LogLik	AIC	BIC	SSPE	APE	GVIF
<FULL MODEL>	-85970.86	172031.7	172459.8	2116908	104301.2	---
log_landmeters2	-85975.77	172039.5	172458.1	2109846	104093.4	9.2939
log_business_sum	-85976.07	172040.1	172458.7	2111715	104130.4	2.2585
Intercept	-85983.26	172054.5	172473.1	2120100	104308.6	---
teaUER	-85984.96	172057.9	172476.5	2117310	104345.4	1.1121
log_gc_sum	-85985.54	172059.1	172477.6	2117925	104185.3	31.0442
teaMOM	-85990.41	172068.8	172487.4	2115589	104309.4	1.8585
uni_dist*	-85992.94	172071.9	172480.9	2107489	104125.7	1.1063
log_forest*_pct	-85995.01	172074.0	172473.6	2130724	104584.9	2.5406
log_pct_api	-85995.98	172080.0	172498.5	2112394	104178.7	2.0437
log_eds_res_sum	-85996.06	172080.1	172498.7	2148298	104831.1	8.2536
log_unitstat1_sum	-85997.23	172082.5	172501.0	2110715	103995.9	19.9242
hasSeasonalY	-85999.01	172086.0	172504.6	2122769	104305.4	1.0885
log_dsf_si_spr00	-86000.35	172088.7	172507.3	2153736	104917.1	6.7937
pct_unemploy_zero	-86002.79	172093.6	172512.1	2119111	104357.6	6.4168
log_shrub_pct	-86008.07	172104.1	172522.7	2114421	104302.0	1.4539
realtrac_*_2007	-86011.64	172109.3	172518.3	2120530	104171.3	1.2003
log_irs1040nm	-86015.24	172118.5	172537.0	2121257	104539.0	1.2678
log_pct_mlt_u_2p_strc	-86016.01	172120.0	172538.6	2120598	104420.4	5.8404
log_pct_li_hh_indo_europe	-86018.48	172125.0	172543.5	2117652	104341.4	1.4805
I1	-86031.04	172150.1	172568.7	2195509	105283.9	8.6869
log_pct_not_single_u_strc	-86036.58	172161.2	172579.7	2119094	104507.2	9.3992
log_landmeters2_sq	-86047.09	172182.2	172600.8	2146784	104952.0	4.6563
log_crops_pct	-86053.84	172195.7	172614.2	2117041	104482.7	1.8612
I2	-86062.70	172213.4	172632.0	2094190	103836.2	6.2153
log_isVacantY_sum	-86074.54	172237.1	172655.7	2120337	104247.6	1.5819
log_dsf_si_spr09	-86081.78	172251.6	172670.1	2125316	103928.2	9.0949
log_irs1040nb	-86095.41	172278.8	172697.4	2136627	105213.2	1.4427
log_pct_crowd_occp_u	-86100.99	172290.0	172708.5	2132394	104728.3	1.9708
log_dep_list	-86105.48	172299.0	172717.5	2118011	104864.7	26.5083
I3	-86107.98	172304.0	172722.5	2117005	104402.1	4.9722
log_devel*_pct	-86116.55	172315.1	172705.1	2128059	104691.7	19.7455
log_pct_pop_0_17	-86131.23	172350.5	172769.0	2124477	104634.4	9.5865
log_hu_density_ratio	-86137.27	172362.5	172781.1	2089070	103620.0	8.3270
I4	-86159.75	172407.5	172826.1	2128475	104896.0	8.1616
stability_index	-86189.57	172467.1	172885.7	2147184	105481.3	2.8988
log_irs1040ng	-86193.97	172475.9	172894.5	2122896	104799.9	1.9294
has_delptypeBk	-86252.72	172593.4	173012.0	2159779	105825.5	1.5264

Variable/Group Definitions

I1: log_eds_res_sum:stability_index
I2: log_unitstat1_sum:log_hu_density_ratio
I3: log_landmeters2:log_dsf_si_spr00
I4: log_dep_list:log_devel1_pct
log_devel*_pct: log_devel0_pct, log_devel1_pct, log_devel2_pct, log_devel3_pct
log_forest*_pct: log_forest1_pct, log_forest2_pct, log_forest3_pct
uni_dist*: uni_dist0, uni_dist2

We take a closer look at the randomized quantile residuals for ZINB in Figure 5.2. The boxplot on the left shows the residuals indicating a "good" fit with absolute value no larger than 3, while the boxplot on the right shows "bad" residuals with absolute value greater than 3. Recall that the residuals follow a standard normal distribution under a good fitting model; therefore, most residuals are expected to be between -3 and 3. Most of the 99,512 good residuals have observed add counts close to zero. On the other hand, only 488 out of 100,000 residuals are considered bad; however, these blocks tend to have larger observed add counts.

The results in Figures 5.1 and 5.2 are restricted to the training set and are based on a single computation of the residuals (which have a random component and will vary slightly from computation to computation). We next compute each residual 1,000 times on the full universe using the estimates from Tables 5.6 and 5.7, yielding $r_i^{(1)}, \ldots, r_i^{(1000)}$ for each $i = 1, \ldots, N$. The following results use averaged residuals $\bar{r}_i = \sum_{\ell=1}^{1000} r_i^{(\ell)}/1000$ for $i = 1, \ldots, N$. Out of 6,539,119 blocks, 25,237 (or about 0.39%) have "bad" $\bar{r}_i$ such that their absolute value is greater than 3. However, this small proportion of blocks contains 1,059,624 (about 18.81%) of the total 5,632,150 adds. Of the 3,182 counties in the universe, 2,573 counties contain at least one bad $\bar{r}_i$, demonstrating that the lack of fit is scattered across the nation. Finally, we note that all of the bad $\bar{r}_i$ exhibit large observed counts and small predicted counts; this indicates that we are missing covariates to detect add activity in these cases. Future work could determine if there is a fundamental difference between the good and bad blocks. Initial attempts to make this distinction using classification trees did not provide additional insight. We suggest that the phenomenon of bad residuals be better understood before using a MAF error model in an operational setting.

Table 5.9 provides a more in-depth look at the raw residuals $|y_i - \hat{y}_i|$ for $i = 1, \ldots, N$. Here, we view the distribution of raw residuals, grouped by the outcome y. For example, the first row shows the quantiles of raw residuals for all blocks in which 0 adds were observed. The last column shows that there were 5,092,781 blocks with 0 adds. Of these blocks, we can see that the smallest raw residual is 0.001, as indicated by the 0 quantile. We can also see that, of these blocks, the median raw residual was 0.236, and 97.5% of the residuals were less than 2.682. If we observe all of the values in each quantile, excluding the 100% quantile, we can see that in general, the residuals are increasing for blocks with higher numbers of adds. Clearly, the model does a better job at predicting the smaller values, which is not surprising based on the residual plots and the intuition that larger counts will be associated with larger errors.

Table 5.10 shows state level estimates of adds based on our zero-inflated count model. The model predicts approximately 98% of adds at the address level nationwide, and also performs well in certain states, including predicting approximately 96% of observed adds in Arizona and approximately 104% of observed adds in New York. However, the model does not do as well for other states. For example, the model predicts only 63% of the observed adds in Washington D.C., and predicts 58% more than the observed adds in Iowa.

We can also see these state level estimates in Figure A.3, which can be compared to Figure A.2. Figure A.3 verifies that the model does not predict as many large adds as were observed in the field. We can also see that where the count is low, and where there is a lower population density, the prediction accuracy improves. Figure A.4 shows "bad" quantile residuals at the block level. It is interesting, yet not surprising, to note that these bad residuals appear to be clustered around major cities, such as New York City, Philadelphia, and Atlanta.

Table 5.11 shows results of a simulated 2010 address canvassing if we were to canvass blocks based on our predictions. To do this, we first sorted blocks in descending order based on the number of predicted adds. The column "% HU Canvassed" represents a threshold for the amount of canvassing we would allow. For example, 20% means that we would select the first k blocks in our sorted list containing at most 20% of the HUs that existed before AdCan.[8] Based on the blocks we select for canvassing, we calculate the false positive rate (FPR) and true positive rate (TPR). In this case, a "false positive" is a block that we select to canvass because our model predicted there would be at least one add, but when canvassed, no adds are found. Thus, the FPR is the percentage of all blocks selected for the canvassing operation in which no adds were found during canvassing. Similarly, a "true positive" is a block that we decide to canvass because our model

[8] A similar method of selecting blocks has commonly been used by AdCan modelers at Census Bureau, but other choices are possible. A threshold of 20% represents a large reduction to the canvassing operation. We could consider a larger threshold as a more conservative approach, or set a threshold for the amount of allowed undercoverage instead.

Table 5.5: Comparison of final model from this paper ("RG") versus Young et al. (2015) ("YRJ").

	Metric	RG Model	YRJ Model
1	Sum of squared prediction errors	235,779,143	572,745,208
2	Mean of squared prediction errors	36.0567	87.5875
3	Sum of absolute prediction errors	6,897,446	7,769,202
4	Mean of absolute prediction errors	1.0548	1.1881
5	Number of blocks in universe with average residual greater than 3 ("bad residuals")	25,237	24,943
6	Number of observed adds in bad blocks	1,059,624	937,907
7	Number of counties containing at least one bad block from the universe	2,573	2,512
8	2010 adds captured by predicted add count†	1,870,340	3,219,174
9	2010 blocks canvassed by predicted add count†	232,971	870,610
10	2010 adds captured by predicted add rate†	3,600,931	3,295,730
11	2010 blocks canvassed by predicted add rate†	2,688,849	1,114,702

†Refers to a simulated address canvassing operation where blocks containing up to 20% of pre-AdCan housing units may be selected for canvassing. Metrics 8 and 9 are based on selecting blocks with the largest add count predicted by the model, while metrics 10 and 11 are based on selecting blocks with the largest predicted add rate.

predicts at least one add, and when canvassed we find at least one add. Therefore, the TPR is the percentage of all blocks containing at least one add which are selected for the canvassing operation. Figure 5.3 shows the add capture rate curve and receiver operating characteristic (ROC) curve corresponding to Table 5.11. Figure 5.3(a) shows that, with perfect knowledge of the locations of the adds, and sorting blocks in the same way as our model, we would only have to canvass blocks containing about 40% of the pre-AdCan housing units in order to capture all adds. Using 40% as the threshold for our model would allow us to capture about 60% of the adds. Figure 5.3(b) tells the same story in terms of the TPR and FPR; to canvass 80% of the blocks with at least one add would require an FPR of 42%.

Table 5.12 is similar to Table 5.11, except now the housing units are sorted in descending order based on the ratio of predicted number of adds to pre-AdCan HUs. Figure 5.4 shows the canvassing and ROC curves corresponding to this method of selecting blocks. This method tends to capture more adds than sorting by the raw prediction, but also tends to produce a larger number of false positives. For example, if we were to canvass blocks containing 40% of housing units using the ratio, we would capture around 81% of the adds. However, to canvass 80% of the blocks with at least one add would require an FPR of 59%.

Results for the ZINB model in this paper can be compared to Young et al. (2015) to gauge the effectiveness of our variable selection method versus the simpler screening method. Both models were developed and evaluated using the same universe of blocks. Table 5.5 shows such a comparison. Results from this paper are denoted as "RG" and results from Young et al. (2015) as "YRJ". Metrics 5, 6, and 7 are based on randomized quantile residuals averaged over 1,000 repetitions, as described earlier in this section. RG provides a noticeably better overall fit, but somewhat fewer bad residuals are obtained under YRJ. This suggests that RG is missing covariates that were effective in YRJ to explain add counts in some blocks. Note that the design matrix for the RG count regression has $d_1 = 39$ columns and the ZI regression has $d_2 = 44$ columns. On the other hand, YRJ has $d_1 = 31$ and $d_2 = 4$, and many of the variables in the count regressions of RG and YRJ do not overlap. The number of bad residuals could potentially be used as a criterion for future model selection to improve upon the RG approach. Metrics 8–11 compare results for canvassing selection between the models. RG produces a much smaller list when sorting by predictions (metrics 8 and 9), while YRG tends to capture many more adds at the cost of a larger list. In addition, RG captures more adds when sorting by prediction rate (metrics 10 and 11), at the cost of a much larger block list. By choosing between the two sorting methods, better results are possible with RG for one criterion. On the other hand, YRG appears to producing more balanced results.

Table 5.6: Parameter estimates for ZINB adds model, count regression and dispersion portions of model.

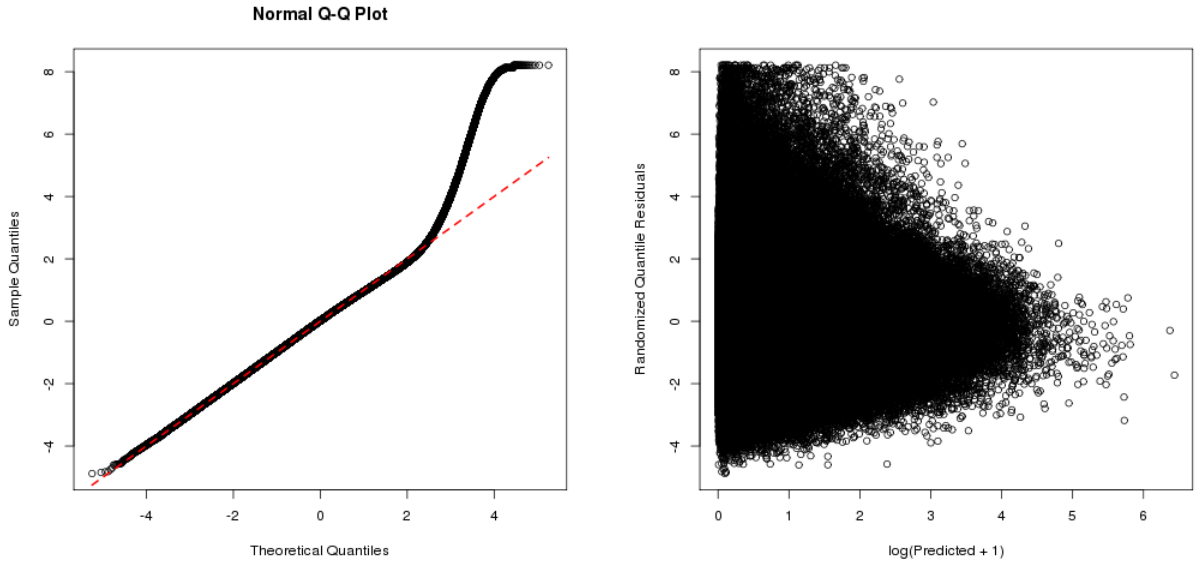
Coefficient	Estimate	SE	p-value	CL Lo	CL Hi
Intercept	0.6101	0.0980	<.0001	0.4180	0.8022
log_dep_list	-0.6519	0.0369	<.0001	-0.7243	-0.5796
log_landmeters2	-0.0226	0.0113	0.0463	-0.0448	-4e-04
log_eds_res_sum	-0.1806	0.0328	<.0001	-0.2448	-0.1163
log_landmeters2_sq	-0.0109	0.0021	<.0001	-0.0150	-0.0067
log_business_sum	0.0349	0.0165	0.0338	0.0027	0.0672
teaMOM	-0.0405	0.0274	0.1399	-0.0943	0.0133
teaUER	0.3025	0.0519	<.0001	0.2007	0.4043
log_gc_sum	0.2504	0.0426	<.0001	0.1670	0.3339
hasSeasonalY	0.4119	0.0541	<.0001	0.3059	0.5179
log_unitstat1_sum	-0.0344	0.0329	0.2967	-0.0989	0.0302
has_delptypeBk	-0.0501	0.0244	0.0404	-0.0980	-0.0022
log_isVacantY_sum	0.1727	0.0132	<.0001	0.1467	0.1986
log_hu_density_ratio	-0.2706	0.0138	<.0001	-0.2976	-0.2436
stability_index	-0.8793	0.0430	<.0001	-0.9636	-0.7950
log_irs1040ng	0.1855	0.0139	<.0001	0.1583	0.2126
log_irs1040nb	0.0849	0.0063	<.0001	0.0726	0.0972
log_devel0_pct	0.0259	0.0082	0.0016	0.0098	0.0419
log_devel1_pct	0.2119	0.0147	<.0001	0.1831	0.2406
log_devel2_pct	0.1082	0.0091	<.0001	0.0904	0.1260
log_devel3_pct	0.0499	0.0134	0.0002	0.0236	0.0761
log_crops_pct	-0.1080	0.0084	<.0001	-0.1243	-0.0916
log_pct_crowd_occp_u	0.1911	0.0220	<.0001	0.1481	0.2341
log_pct_pop_0_17	-0.3190	0.0230	<.0001	-0.3641	-0.2740
log_pct_not_single_u_strc	0.1678	0.0206	<.0001	0.1274	0.2082
log_forest1_pct	-0.0154	0.0080	0.0556	-0.0311	4e-04
log_forest2_pct	-0.0094	0.0095	0.3232	-0.0279	0.0092
log_forest3_pct	0.0305	0.0121	0.0118	0.0067	0.0542
log_dsf_si_spr00	-0.1037	0.0142	<.0001	-0.1315	-0.0759
log_shrub_pct	0.0411	0.0092	<.0001	0.0232	0.0591
log_dsf_si_spr09	0.1471	0.0162	<.0001	0.1154	0.1788
pct_unemploy_zero	-0.3067	0.0456	<.0001	-0.3961	-0.2173
log_pct_li_hh_indo_europe	0.2187	0.0320	<.0001	0.1559	0.2815
log_irs1040nm	0.0012	0.0104	0.9084	-0.0192	0.0216
log_pct_mlt_u_2p_strc	-0.1110	0.0160	<.0001	-0.1425	-0.0796
realtrac_1_5_2007	-0.2369	0.0334	<.0001	-0.3024	-0.1715
realtrac_11plus_2007	0.2460	0.0857	0.0041	0.0781	0.4139
log_pct_api	0.2139	0.0236	<.0001	0.1677	0.2602
uni_dist0	0.4486	0.1102	<.0001	0.2326	0.6646
uni_dist2	0.1033	0.0445	0.0203	0.0161	0.1904
log_dep_list:log_devel1_pct	-0.0820	0.0048	<.0001	-0.0914	-0.0725
log_landmeters2:log_dsf_si_spr00	0.0480	0.0033	<.0001	0.0415	0.0545
log_unitstat1_sum:log_hu_density_ratio	0.0670	0.0048	<.0001	0.0576	0.0765
log_eds_res_sum:stability_index	0.2609	0.0401	<.0001	0.1823	0.3394
Dispersion	1.9918	0.0328	<.0001	1.9276	2.0560

Table 5.7: Parameter estimates for ZINB adds model, zero-inflated regression portion of model.

Coefficient	Estimate	SE	p-value	CL Lo	CL Hi
Intercept	0.0221	0.2162	0.9185	-0.4016	0.4459
log_dep_list	-0.1813	0.0463	<.0001	-0.2721	-0.0904
log_landmeters2	0.0888	0.0286	0.0019	0.0327	0.1450
log_eds_res_sum	-0.2973	0.0292	<.0001	-0.3546	-0.2400
log_landmeters2_sq	0.0127	0.0049	0.0092	0.0031	0.0223
teaMOM	0.3981	0.0662	<.0001	0.2683	0.5279
hasSeasonalY	-0.8338	0.3716	0.0248	-1.5621	-0.1055
log_unitstat1_sum	0.2108	0.0570	0.0002	0.0990	0.3225
log_delptypeBk_sum	-1.6071	0.0725	<.0001	-1.7492	-1.4650
log_isVacantY_sum	-0.1201	0.0312	0.0001	-0.1812	-0.0590
log_mafsrc1_sum	-0.1385	0.0175	<.0001	-0.1729	-0.1041
log_compcity1_sum	0.4212	0.0652	<.0001	0.2934	0.5490
log_forest1_pct	-0.0891	0.0190	<.0001	-0.1263	-0.0519
log_forest2_pct	-0.1231	0.0268	<.0001	-0.1756	-0.0706
log_forest3_pct	-0.1355	0.0387	0.0005	-0.2114	-0.0596
log_irs1040ng	-0.2034	0.0301	<.0001	-0.2623	-0.1444
log_pct_crowd_occp_u	-0.3167	0.0384	<.0001	-0.3919	-0.2414
log_crops_pct	0.0281	0.0201	0.1630	-0.0114	0.0675
log_dsf_si_spr09	-1.8498	0.0943	<.0001	-2.0347	-1.6650
log_shrub_pct	-0.1365	0.0238	<.0001	-0.1833	-0.0898
log_devel0_pct	-0.0018	0.0142	0.8996	-0.0296	0.0261
log_devel1_pct	0.0873	0.0260	0.0008	0.0364	0.1382
log_devel2_pct	0.2889	0.0277	<.0001	0.2347	0.3432
log_devel3_pct	0.0852	0.0188	<.0001	0.0484	0.1221
stability_index	-0.3301	0.0824	<.0001	-0.4915	-0.1686
hu_block2tract_ratio	-4.4435	1.4134	0.0017	-7.2136	-1.6734
log_pct_pop_0_17	0.0850	0.0328	0.0095	0.0208	0.1493
log_irs1040nb	-0.0752	0.0156	<.0001	-0.1059	-0.0446
log_irs1040nm	-0.0234	0.0324	0.4693	-0.0868	0.0400
log_htc	-0.0732	0.0175	<.0001	-0.1075	-0.0390
log_pct_mlt_u_10p_strc	0.2275	0.0292	<.0001	0.1703	0.2848
log_pct_not_single_u_strc	-0.1191	0.0409	0.0036	-0.1993	-0.0389
log_pct_black	0.1233	0.0194	<.0001	0.0854	0.1613
log_devel1_pct:log_compcity1_sum	0.0083	0.0091	0.3635	-0.0096	0.0262
log_dep_list:log_dsf_si_spr09	0.1356	0.0135	<.0001	0.1091	0.1621
log_landmeters2:log_dsf_si_spr09	-0.0996	0.0083	<.0001	-0.1159	-0.0833
log_dsf_si_spr09:log_delptypeBk_sum	0.1493	0.0239	<.0001	0.1024	0.1962
log_dsf_si_spr09:log_irs1040nm	-0.1092	0.0131	<.0001	-0.1350	-0.0835
log_irs1040nb:log_devel2_pct	0.0384	0.0052	<.0001	0.0282	0.0486

Table 5.8: Fit statistics for ZINB Adds model. Log-likelihood, AIC and BIC are computed on the training set. Prediction error measures SSPE, MSPE, APE, and MAPE are computed using all blocks in the universe.

		ZINB	NegBin	Poisson
Training Set	LogLik	-83,113	-85,971	-152,561
	AIC	166,393	172,032	305,210
	BIC	167,192	172,460	305,629
Universe	SSPE	235,779,143	240,267,978	232,626,457
	MSPE	36.0567	36.7432	35.5746
	APE	6,897,446	7,054,974	6,900,898
	MAPE	1.0548	1.0789	1.0553



(a) (b)

Figure 5.1: Randomized quantile residuals from ZINB model computed on the training set.

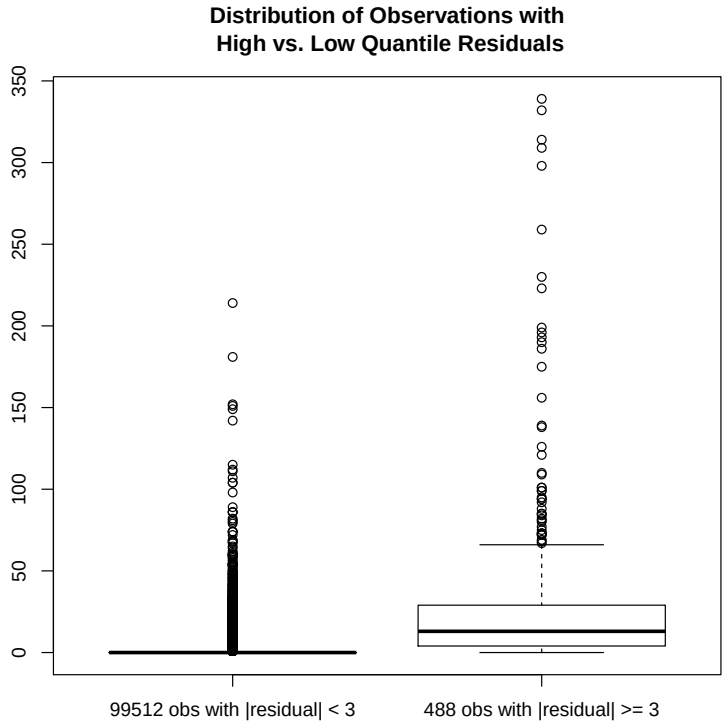


Figure 5.2: Residuals with absolute value no greater than 3 is shown in left box. Right box shows residuals with absolute value greater than 3.

Table 5.9: Quantiles of absolute residuals for observed counts.

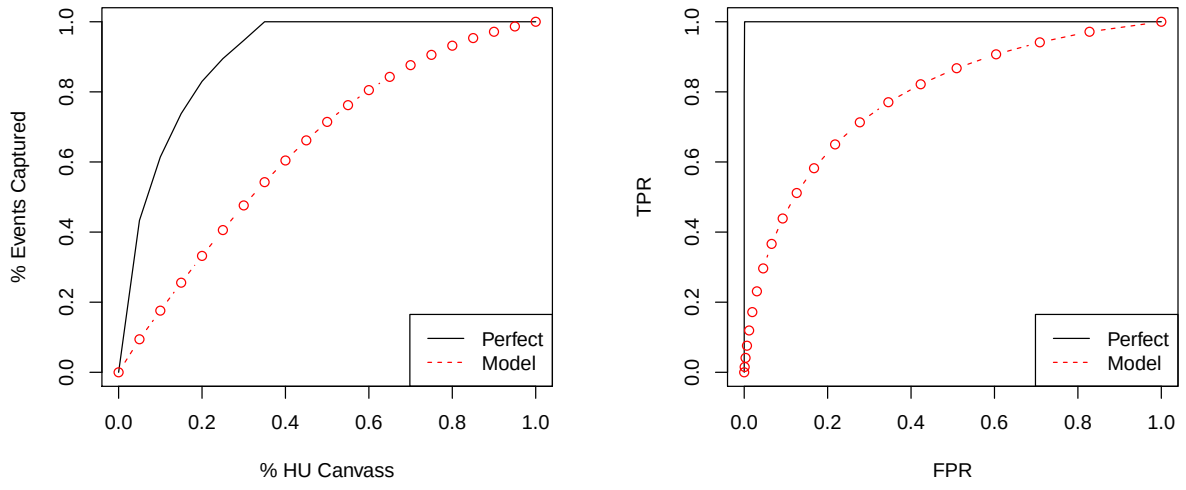
	Quantile for Absolute Residual							
Observed	0	0.025	0.05	0.5	0.95	0.975	1	n
0	0.001	0.034	0.047	0.236	1.813	2.682	623.076	5,092,781
1	0.000	0.037	0.073	0.649	2.594	3.978	274.625	707,952
2	0.000	0.073	0.145	1.249	2.932	4.696	298.860	266,872
3	0.000	0.117	0.233	1.867	3.144	5.227	266.697	135,526
4	0.000	0.170	0.349	2.539	3.909	5.534	132.903	81,136
5	0.000	0.239	0.465	3.221	4.869	5.845	122.354	52,276
6	0.000	0.300	0.577	3.965	5.834	5.959	131.859	37,024
7	0.001	0.380	0.760	4.673	6.812	6.927	192.298	26,704
8	0.001	0.451	0.925	5.518	7.806	7.922	88.949	20,720
9	0.001	0.547	1.125	6.199	8.771	8.891	63.426	16,147
$[10, 15)$	0.001	0.899	1.728	8.214	12.524	13.252	282.965	44,953
$[15, 20)$	0.006	1.732	3.394	12.668	17.608	18.267	244.095	19,788
$[20, 25)$	0.007	2.654	4.855	17.285	22.648	23.347	145.006	10,771
$[25, 30)$	0.003	3.676	6.959	21.601	27.435	27.986	306.231	6,411
$[30, 35)$	0.032	5.197	9.567	26.631	32.550	33.136	217.028	4,342
$[35, 40)$	0.060	7.133	11.952	31.296	37.377	37.983	126.742	2,854
$[40, 45)$	0.069	7.514	13.137	36.009	42.433	43.025	113.617	2,156
$[45, 50)$	0.673	11.838	17.812	41.590	47.718	48.133	282.034	1,654
$[50, 55)$	0.455	14.055	21.105	45.880	52.344	53.002	79.372	1,216
$[55, 60)$	0.105	19.110	25.006	51.343	57.611	58.322	177.431	942
$[60, 65)$	0.634	13.721	24.864	56.501	62.901	63.661	74.981	846
$[65, 70)$	1.468	21.747	31.446	60.736	67.570	68.173	68.888	627
$[70, 75)$	5.063	21.417	33.018	66.964	72.802	73.377	122.100	578
$[75, 80)$	4.897	24.306	36.500	71.605	77.339	78.116	78.872	456
$[80, 85)$	31.625	42.064	49.810	76.988	83.140	83.591	83.917	421
$[85, 90)$	1.997	27.365	46.929	81.176	87.748	87.995	88.897	344
$[90, 95)$	8.838	41.817	54.864	86.751	92.439	92.927	93.809	273
$[95, 100)$	11.960	55.444	65.641	92.876	98.216	98.592	98.832	315
$[100, 200)$	1.224	61.084	77.171	119.357	182.155	188.807	437.275	2,084
$[200, 300)$	1.955	157.030	179.236	226.089	284.299	289.063	298.584	544
$[300, 400)$	248.534	281.976	291.277	329.913	383.152	390.189	398.628	221
$[400, 500)$	149.312	357.271	384.476	435.238	491.178	495.371	495.691	99
$[500, 1000)$	453.317	469.041	481.368	625.933	896.527	950.374	988.716	73
$[1000, 5000)$	1041.198	1043.260	1045.322	1270.179	2389.017	3011.024	3633.032	13

Table 5.10: State level coverage estimates.

State	Observed Adds	Predicted Adds	Ratio	HUs	Blocks
AL	157,450	151,847.60	0.96	2,392,330	143,079
AK	38,131	25,452.23	0.67	323,647	10,858
AZ	140,553	134,911.10	0.96	3,079,198	118,362
AR	87,925	100,239.10	1.14	1,494,997	103,004
CA	386,516	411,125.60	1.06	14,786,176	425,696
CO	74,115	86,457.54	1.17	2,481,927	116,085
CT	30,045	36,770.49	1.22	1,561,327	48,583
DE	20,564	18,685.54	0.91	390,686	15,563
DC	9,409	5,916.85	0.63	316,722	4,567
FL	308,930	321,416.50	1.04	10,072,338	317,818
GA	222,328	214,016.20	0.96	4,721,702	182,212
HI	48,322	33,879.41	0.70	510,781	11,066
ID	48,621	42,870.88	0.88	714,804	57,946
IL	118,561	132,829.90	1.12	5,725,571	315,050
IN	75,160	76,860.22	1.02	2,983,379	188,390
IA	23,857	37,725.09	1.58	1,424,149	144,205
KS	28,465	42,946.07	1.51	1,332,228	134,564
KY	147,040	106,649.60	0.73	2,137,716	94,669
LA	118,691	99,365.27	0.84	2,191,785	107,510
ME	88,957	71,771.20	0.81	758,022	37,306
MD	46,200	40,594.83	0.88	2,576,329	85,114
MA	73,992	83,283.95	1.13	2,989,361	99,964
MI	134,967	146,420.50	1.08	4,937,906	219,787
MN	84,706	90,327.32	1.07	2,476,363	159,975
MS	109,252	107,851.10	0.99	1,382,160	90,537
MO	139,851	144,472.00	1.03	2,912,338	193,592
MT	44,117	47,774.81	1.08	552,090	52,246
NE	21,041	30,919.65	1.47	841,438	104,353
NV	48,756	42,715.47	0.88	1,289,125	36,940
NH	48,409	43,622.85	0.90	617,883	29,837
NJ	92,253	92,007.15	1.00	3,760,709	122,371
NM	96,314	85,377.00	0.89	1,064,238	65,249
NY	318,808	331,593.90	1.04	9,019,977	255,527
NC	220,956	228,837.60	1.04	5,088,628	198,757
ND	15,519	18,800.71	1.21	350,168	55,705
OH	100,342	106,774.90	1.06	5,527,999	251,222
OK	128,419	131,399.90	1.02	1,739,117	140,420
OR	65,116	61,638.39	0.95	1,738,982	89,920
PA	254,214	226,768.10	0.89	5,996,659	302,288
RI	14,235	14,395.27	1.01	473,918	17,938
SC	109,847	114,368.50	1.04	2,557,421	118,836
SD	20,683	23,209.02	1.12	368,986	48,275
TN	129,803	119,201.70	0.92	3,044,881	151,188
TX	540,799	508,966.30	0.94	10,663,218	474,444
UT	57,992	40,824.88	0.70	952,217	45,970
VT	20,463	27,469.29	1.34	431,035	19,451
VA	121,289	124,589.40	1.03	3,709,883	153,940
WA	146,613	109,016.70	0.74	2,890,612	122,440
WV	155,323	101,414.20	0.65	934,461	65,825
WI	76,759	102,233.30	1.33	2,770,549	161,698
WY	21,472	20,956.14	0.98	287,471	28,777
US	5,632,150	5,519,561	0.98	143,345,607	6,539,119

Table 5.11: Canvassing using predicted number of adds.

% HU Canvassed	FPR	TPR	Captured Adds	Blocks Canvassed
0.00	0.00	0.00	0	0
0.05	0.00	0.02	531,049	27,341
0.10	0.00	0.04	990,657	75,593
0.15	0.01	0.08	1,439,819	143,439
0.20	0.01	0.12	1,870,340	232,971
0.25	0.02	0.17	2,285,421	347,464
0.30	0.03	0.23	2,681,182	488,725
0.35	0.05	0.30	3,055,350	660,373
0.40	0.07	0.37	3,403,383	864,590
0.45	0.09	0.44	3,726,573	1,104,423
0.50	0.13	0.51	4,023,194	1,380,848
0.55	0.17	0.58	4,292,387	1,694,572
0.60	0.22	0.65	4,534,161	2,049,923
0.65	0.28	0.71	4,746,765	2,442,790
0.70	0.35	0.77	4,935,117	2,874,633
0.75	0.42	0.82	5,100,564	3,343,100
0.80	0.51	0.87	5,247,746	3,848,260
0.85	0.60	0.91	5,370,259	4,387,645
0.90	0.71	0.94	5,471,435	4,970,682
0.95	0.83	0.97	5,556,841	5,621,326
1.00	1.00	1.00	5,632,150	6,539,119



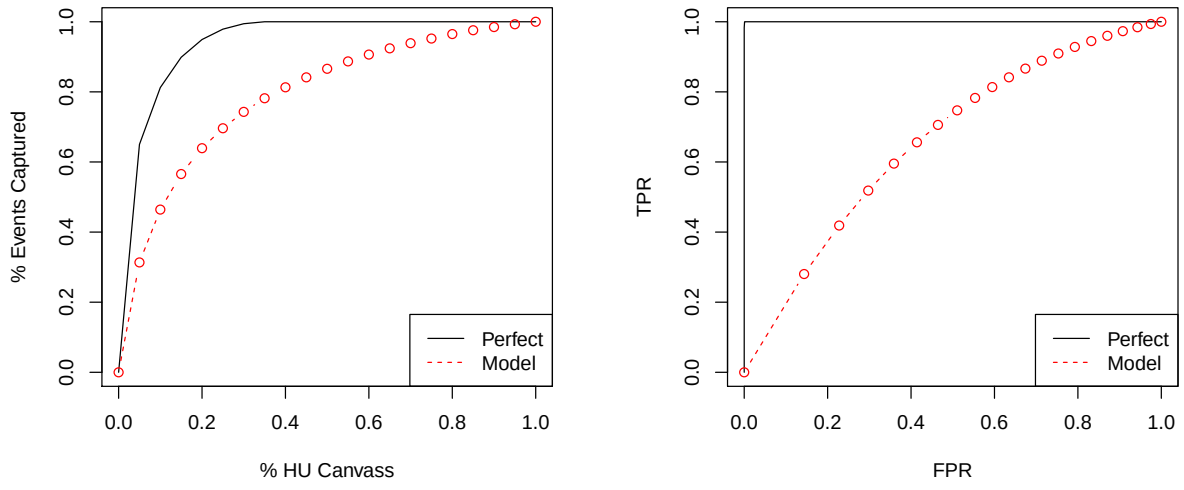
(a) Add capture rate.

(b) ROC curve.

Figure 5.3: Canvassing using predicted number of adds from ZINB Adds model.

Table 5.12: Canvassing using ratio of predicted number of adds to HUs.

% HU Canvassed	FPR	TPR	Captured Adds	Blocks Canvassed
0.00	0.00	0.00	0	0
0.05	0.14	0.28	1,765,560	1,136,726
0.10	0.23	0.42	2,614,613	1,765,485
0.15	0.30	0.52	3,185,617	2,264,339
0.20	0.36	0.60	3,600,931	2,688,849
0.25	0.41	0.66	3,921,279	3,058,043
0.30	0.46	0.71	4,184,387	3,386,628
0.35	0.51	0.75	4,403,136	3,681,057
0.40	0.55	0.78	4,579,042	3,951,077
0.45	0.59	0.81	4,739,085	4,205,379
0.50	0.63	0.84	4,876,692	4,449,650
0.55	0.67	0.87	4,995,490	4,685,952
0.60	0.71	0.89	5,105,193	4,918,795
0.65	0.75	0.91	5,204,995	5,151,096
0.70	0.79	0.93	5,287,798	5,379,376
0.75	0.83	0.94	5,362,654	5,604,186
0.80	0.87	0.96	5,434,725	5,822,045
0.85	0.91	0.97	5,496,226	6,030,328
0.90	0.94	0.98	5,547,080	6,225,538
0.95	0.97	0.99	5,591,990	6,402,255
1.00	1.00	1.00	5,632,150	6,539,119



(a) Add capture rate.

(b) ROC curve.

Figure 5.4: Canvassing using ratio of predicted number of adds to HUs using ZINB Adds model.

6 Conclusions

This report discussed the modeling of addresses added to the MAF which were obtained through the 2010 AdCan operation. The add counts indicated where our usual MAF updating procedures did not match the field before the 2010 Census. The mechanism that caused the adds was complex and not fully known at the time of data analysis. For example, determination of a housing unit as an add was subject to FR behavior, in-office adjudication, and other operational details which were not available in the data. Therefore, an exhaustive variable selection was needed to find predictors that could explain the observed adds. We made use of a stepwise method that is simple and intuitive, but also quite time-consuming. This method allowed us to consider several supplemental datasets and determine the most useful predictors. It also allowed us to consider two-way interactions and to rank the selected variables by their contribution to the model. Evaluation of the resulting zero-inflated negative binomial model found that a small proportion (0.39%) of blocks are not well explained by the model, but these tend to be the ones with the most adds.

Future work should investigate the nature of the poor-fitting blocks. We may be missing important predictors to explain adds, or perhaps their mechanism for producing adds is fundamentally different. We could also consider ways to better handle the heterogeneity from a purely modeling standpoint. Random effects could be used to allow additional variability in the observed counts (McCulloch et al., 2008) or to induce spatial dependence between nearby blocks (Banerjee et al., 2003). Finite mixtures of regressions (Frühwirth-Schnatter, 2006) could be used to model several heterogeneous subpopulations with fundamentally different regression functions.

One reviewer points out that our training set contains very few blocks with large numbers of housing units, relative to the sample size of 100,000. For example, 30,608 blocks in the modeling universe had more than 300 pre-AdCan housing units. These blocks contained 318,412 adds. Of these 30,608 blocks, 469 were included in our sample, and only 230 of these 469 had at least one add. Blocks with many pre-AdCan housing units had a high potential for large add counts, and perhaps including more in the training set could improve the model's ability to detect adds. More careful selection of the training set could be considered in future work.

Aside from improving the model, other issues mentioned briefly in this report also remain to be addressed. These include evaluating models for use in operations and determining precisely how the models would be used in an operation. We have not developed a model for deletes or matched adds in this report, but such efforts could potentially be carried out using similar methodology, or perhaps combined into a unified methodology.

Acknowledgements

The authors are grateful to a number of colleagues at Census Bureau whose support made this work possible. Tommy Wright (CSRM) provided constant encouragement and helped to overcome a number of obstacles in completing the work. Eric Slud (CSRM) helped us to identify the major objectives of this report: the selection of predictors and interactions for the model, and identification of the most useful predictors. We thank DSSD managers Pat Cantwell, Debbie Fenstermaker, and Laura Ferreira for funding our research and for including us in meetings on the operational side of the project. Derek Young (U. of Kentucky) and Kim Sellers (Georgetown U.) facilitated helpful conversations on count modeling methodology and its application to this problem. Additionally, Derek Young shared all details about his initial work on the MAF error model with us. John Boies, Kevin Shaw, and Christine Tomaszewski from DSSD engaged us in valuable discussions on the subject matter, nuances of the data, and previous work. Reid Rottach (DSSD) gave the recommendation to select blocks for canvassing based on prediction ratio rather than raw prediction. David Brown (CES) engaged us in a number of discussions on the modeling effort and provided us with the IRS and Land Use predictors, as well as other datasets not discussed in the report. Mark Kutzbach (CES) provided us with carefully crafted LEHD predictors. We are grateful to April Avnayim, Shonin Anacker, and Mike Ratcliffe from GEO for the DSF stability index and land use data, along with their documentation. Thanks to Krista Heim and Kevin Shaw from DSSD for providing a peer review of this manuscript, and to Tommy

Wright and Lauren Emanuel from CSRM for editorial feedback. A number of other colleagues provided input as well, and we greatly appreciate their feedback and encouragement.

References

Sudipto Banerjee, Brad P. Carlin, and Alan E. Gelfand. *Hierarchical Modeling and Analysis for Spatial Data*. Chapman & Hall, CRC Press, 2003.

John L. Boies, Kevin M. Shaw, and Jonathan P. Holland. 2010 Census Program for Evaluations and Experiments Address Canvassing Targeting and Cost Reduction Evaluation Report. In *2010 Census Planning Memoranda Series*, 2012.

Antonio Bruce and J. Gregory Robinson. Tract Level Planning Database with Census 2000 Data. 2004. URL https://www.census.gov/2010census/partners/pdf/TractLevelCensus2000Apr_2_09.pdf. Accessed: June 25, 2015.

Peter K. Dunn and Gordon K. Smyth. Randomized Quantile Residuals. *Journal of Computational and Graphical Statistics*, 5(3):236–244, 1996.

John Fox and Georges Monette. Generalized Collinearity Diagnostics. *Journal of the American Statistical Association*, 87(417):178–183, 1992.

Sylvia Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models*. Springer, 2006.

Joseph M. Hilbe. *Negative Binomial Regression*. Cambridge University Press, 2nd edition, 2011.

Collin Homer, Jon Dewitz, Joyce Fry, Michael Coan, Nazmul Hossian, Charles Larson, Nate Herold, Alexa McKerrow, J. Nick VanDriel, and James Wickham. Completion of the 2001 National Land Cover Database for the conterminous United States. *Photogrammetric Engineering and Remote Sensing*, 73(4):337 – 341, 2007.

Charles E. McCulloch, Shayle R. Searle, and John M. Neuhaus. *Generalized, Linear, and Mixed Models*. Wiley-Interscience, 2nd edition, 2008.

Darcy Steeg Morris, Andrew Keller, and Brian Clark. An approach for using administrative records to reduce contacts in the 2020 census. In *JSM Proceedings, Government Statistics Section. Alexandria, VA: American Statistical Association*, pages 3278–3292, 2015.

Bryan Schar, James Lawrence, Star Ying, and Jim Hartman. An Investigation into Expanding the Community Address Updating System Universe. In *DSSD 2012 American Community Survey Memorandum Series*, 2012.

Eric Slud and Chandra Erdman. Adaptive Curtailment of Survey Followup Based on Contact History Data. In *Proceedings of the Federal Committee on Statistical Methodology (FCSM) Research Conference*, 2013.

Christine Gibson Tomaszewski. 2009 Targeted Address Canvassing User Documentation. In *DSSD 2020 Decennial Census RT Memorandum Series*, 2014.

U.S. Census Bureau. 2010 Census Address Canvassing Operational Assessment. In *2010 Census Planning Memoranda Series: 2010 Census Program for Evaluations and Experiments*. U.S. Census Bureau, 2012. URL https://www.census.gov/2010census/pdf/2010_Census_AC_Operational_Assessment.pdf.

U.S. Census Bureau. American Community Survey Design and Methodology, January 2014a. URL http://www.census.gov/programs-surveys/acs/methodology/design-and-methodology.html.

U.S. Census Bureau. Geography Division Address Canvassing Recommendation, 2014b. URL http://www2.census.gov/geo/pdfs/gssi/Address_Canvassing_Recommendation.pdf.

U.S. Census Bureau. Analysis of Statistical Models using the 2015 Master Address File Model Validation Test Results, 2015. In progress.

Derek S. Young, Andrew M. Raim, and Nancy R. Johnson. Zero-Inflated Modeling for Characterizing Coverage Errors of Extracts from the U.S. Census Bureau's Master Address File, 2015. Accepted for publication to *Journal of the Royal Statistical Society: Series A*.

A Additional Tables and Figures

Table A.1: Variables from AdCan database.

Variable Name	Description
actionA_sum	Blocks with zero adds. y = I(actionA_sum = 0)
d_flag_ac_count_sum	Number of housing units in a block as a result of canvassing. log_dep_list = log(d_flag_ac_count_sum)
gqv_tea_cat	Type of enumeration area.. MOM: mailout/mailback or military. UER: list/enumerate for 2000, remote update/enumerate for 2010 or remote Alaska or update/enumerate. teaMOM = I(gqv_tea_cat = 'MOM') teaUER = I(gqv_tea_cat = 'UER')
dacs09s_ur_urban	Block is in an urban or rural area. urbanZERO = I(dacs09s_ur_urban = 0)
log_landmeters2	Land area in square kilometers. log_landmeters2 = log(a12_LandMeters2 + 0.0001) log_landmeters2_sq = log_landmeters2^2
log_acs_hu_ratio	ACS Housing unit ratio. log_acs_hu_ratio = log(d_flag_a9_count_sum / d_flag_ac_count_sum + 1)
d_flag_gc_count_sum	Number of geocoded housing units prior to Address Canvassing. log_gc_sum = log(d_flag_gc_count_sum + 1)
dpreac_gc_eds_res_sum	Number of excluded from delivery statistics Delivery Point Type housing units. log_eds_res_sum = log(dpreac_gc_eds_res_sum + 1)
dpreac_gc_business_sum	Number of business related DSF Delivery Point Type housing units. log_business_sum = log(dpreac_gc_business_sum + 1)
dpreac_gc_isSeasonalY_sum	Seasonal housing units on block, i.e. unit is only occupied at certain times of the year. log_isSeasonalY_sum = log(dpreac_gc_isSeasonalY_sum + 1) hasSeasonalY = I(dpreac_gc_isSeasonalY_sum > 0)
dpreac_gc_unitstat1_sum	Number of valid living quarters. log_unitstat1_sum = log(dpreac_gc_unitstat1_sum + 1)
dpreac_a9_delptypeBk_sum	Number of blank Delivery Point Type housing units. log_delptypeBk_sum = log(dpreac_a9_delptypeBk_sum + 1)
dpreac_a9_mafsrc1_sum	Number of housing units with MAF Source Code of DSF - Delivery Sequence File. log_mafsrc1_sum = log(dpreac_a9_mafsrc1_sum + 1)
dpreac_gc_adcanaf0_sum	Number of housing units not valid for Address Canvassing delivery. log_adcanaf0_sum = log(dpreac_gc_adcanaf0_sum + 1)
dpreac_a9_isVacantY_sum	Number of vacant housing units. log_isVacantY_sum = log(dpreac_a9_isVacantY_sum + 1)
dpreac_a9_compcity1_sum	Number of housing units with complete city style address information. log_compcity1_sum = log(dpreac_a9_compcity1_sum + 1)
dpreac_a9_delptypeBk_sum	Indicator of whether or not the block contains at least one blank Delivery Point Type housing. has_delptypeBk = I(dpreac_a9_delptypeBk_sum > 0)
dpreac_a9_mafsrc2_sum	Number of housing units with MAF Source Code of 1990 ACF. log_mafsrc2_sum = I(dpreac_a9_mafsrc2_sum + 1)

(Derived)	Housing unit density at the block level.
	`block_hu_density = d_flag_ac_count_sum / (a12_LandMeters2 + 0.0001)`
	
	Housing unit density at the tract level
	`tract_hu_density = tract_ac_count_sum / (tract_landmeters2 + 0.0001)`
	
	Housing unit density ratio from block to tract level
	`log_hu_density_ratio = log(block_hu_density / tract_hu_density)`
	
	Total housing unit ratio from block to tract level
	`hu_block2tract_ratio = d_flag_ac_count_sum / tract_ac_count_sum`

Table A.2: Variables from 2000 Census Planning Database. The 2010 suffix indicates that the data originally from 2000 census geography has been transformed to 2010 geography.

Variable Name	**Description**

`htc_2010`	"Hard-to-Count" score.
	`log_htc = log(htc_2010 + 0.1)`
	`htc_zero = I(htc_2010 = 0)`

`mail_return_rate_2010`	Mail return rate.
	`log_mail_return_rate = log(mail_return_rate_2010 + 1)`
	`mail_return_rate_zero = I(mail_return_rate_2010 = 0)`

`pct_vacant_hu_2010`	Percent vacant units.
	`log_pct_vacant_hu = log(pct_vacant_hu_2010 + 1)`
	`pct_vacant_hu_zero = I(pct_vacant_hu_2010 = 0)`

`pct_single_u_strc_2010`	Percent single detached or attached housing units in structure.
	`log_pct_single_u_strc = log(pct_single_u_strc_2010 + 1)`
	`pct_single_u_strc_zero = I(pct_single_u_strc_2010 = 0)`

`pct_not_single_u_strc_2010`	Percent of housing units that are not single detached or attached units.
	`log_pct_not_single_u_strc = log(pct_not_single_u_strc_2010 + 1)`
	`pct_not_single_u_strc_zero = I(pct_not_single_u_strc_2010 = 0)`

`pct_mlt_u_10p_strc_2010`	Percent 10 or more housing units in structure.
	`log_pct_mlt_u_10p_strc = log(pct_mlt_u_10p_strc_2010 + 1)`
	`pct_mlt_u_10p_strc_zero = I(pct_mlt_u_10p_strc_2010 = 0)`

`pct_mlt_u_2p_strc_2010`	Percent 2 or more housing units in structure.
	`log_pct_mlt_u_2p_strc = log(pct_mlt_u_2p_strc_2010 + 1)`
	`pct_mlt_u_2p_strc_zero = I(pct_mlt_u_2p_strc_2010 = 0)`

`pct_mobile_home_2010`	Percent trailer/mobile home.
	`log_pct_mobile_home = log(pct_mobile_home_2010 + 1)`
	`pct_mobile_home_zero = I(pct_mobile_home_2010 = 0)`

`pct_renter_occp_hu_2010`	Percent renter occupied.
	`log_pct_renter_occp_hu = log(pct_renter_occp_hu_2010 + 1)`
	`pct_renter_occp_hu_zero = I(pct_renter_occp_hu_2010 = 0)`

`pct_crowd_occp_u_2010`	Percent occupied units with $\geq$ 1.5 persons per room.
	`log_pct_crowd_occp_u = log(pct_crowd_occp_u_2010 + 1)`
	`pct_crowd_occp_u_zero = I(pct_crowd_occp_u_2010 = 0)`

`pct_not_hb_wf_hh_2010`	Percent households that are not husband/wife families.
	`log_pct_not_hb_wf_hh = log(pct_not_hb_wf_hh_2010 + 1)`
	`pct_not_hb_wf_hh_zero = I(pct_not_hb_wf_hh_2010 = 0)`

`pct_occp_u_no_ph_srvc_2010`	Percent occupied units with no telephone service.
	`log_pct_occp_u_no_ph_srvc = log(pct_occp_u_no_ph_srvc_2010 + 1)`
	`pct_occp_u_no_ph_srvc_zero = I(pct_occp_u_no_ph_srvc_2010 = 0)`

pct_not_hs_grad_2010	Percent of ages 25+ who are not high school graduates. log_pct_not_hs_grad = log(pct_not_hs_grad_2010 + 1) pct_not_hs_grad_zero = I(pct_not_hs_grad_2010 = 0)
pct_prs_blw_pov_lev_2010	Percent people below poverty. log_pct_prs_blw_pov_lev = log(pct_prs_blw_pov_lev_2010 + 1) pct_prs_blw_pov_lev_zero = I(pct_prs_blw_pov_lev_2010 = 0)
pct_pub_asst_inc_2010	Percent households with public assistance income. log_pct_pub_asst_inc = log(pct_pub_asst_inc_2010 + 1) pct_pub_asst_inc_zero = I(pct_pub_asst_inc_2010 = 0)
pct_unemploy_2010	Percent of people unemployed. log_pct_unemploy = log(pct_unemploy_2010 + 1) pct_unemploy_zero = I(pct_unemploy_2010 = 0)
pct_li_hh_2010	Percent linguistically isolated households. log_pct_li_hh = log(pct_li_hh_2010 + 1) pct_li_hh_zero = I(pct_li_hh_2010 = 0)
pct_li_hh_span_2010	Percent linguistically isolated Spanish households. log_pct_li_hh_span = log(pct_li_hh_span_2010 + 1) pct_li_hh_span_zero = I(pct_li_hh_span_2010 = 0)
pct_li_hh_indo_europe_2010	Percent linguistically isolated Indo-European households. log_pct_li_hh_indo_europe = log(pct_li_hh_indo_europe_2010 + 1) pct_li_hh_indo_europe_zero = I(pct_li_hh_indo_europe_2010 = 0)
pct_li_hh_api_2010	Percent linguistically isolated Asian and Pacific Islander households. log_pct_li_hh_api = log(pct_li_hh_api_2010 + 1) pct_li_hh_api_zero = I(pct_li_hh_api_2010 = 0)
pct_li_hh_other_2010	Percent isolated other language households. log_pct_li_hh_other = log(pct_li_hh_other_2010 + 1) pct_li_hh_other_zero = I(pct_li_hh_other_2010 = 0)
pct_occp_hu_moved_2010	Percent occupied units where householder moved into unit in 1999–2000. log_pct_occp_hu_moved = log(pct_occp_hu_moved_2010 + 1) pct_occp_hu_moved_zero = I(pct_occp_hu_moved_2010 = 0)
pct_white_2010	Percent white. log_pct_white = log(pct_white_2010 + 1) pct_white_zero = I(pct_white_2010 = 0)
pct_black_2010	Percent black/african american. log_pct_black = log(pct_black_2010 + 1) pct_black_zero = I(pct_black_2010 = 0)
pct_aian_2010	Percent American indian and Alaska native, log_pct_aian = log(pct_aian_2010 + 1) pct_aian_zero = I(pct_aian_2010 = 0)
pct_asian_2010	Percent Asian log_pct_asian = log(pct_asian_2010 + 1) pct_asian_zero = I(pct_asian_2010 = 0)
pct_nhpi_2010	Percent native Hawaiian and pacific islander. log_pct_nhpi = log(pct_nhpi_2010 + 1) pct_nhpi_zero = I(pct_nhpi_2010 = 0)
pct_api_2010	Percent asian and pacific islander. log_pct_api = log(pct_api_2010 + 1) pct_api_zero = I(pct_api_2010 = 0)
pct_2p_race_2010	Percent two or more races. log_pct_2p_race = log(pct_2p_race_2010 + 1) pct_2p_race_zero = I(pct_2p_race_2010 = 0)
pct_sor_2010	Percent some other race. log_pct_sor = log(pct_sor_2010 + 1) pct_sor_zero = I(pct_sor_2010 = 0)

pct_hsp_2010	Percent Hispanic origin (of any race). log_pct_hsp = log(pct_hsp_2010 + 1) pct_hsp_zero = I(pct_hsp_2010 = 0)
pct_non_hisp_white_2010	Percent non-Hispanic white. log_pct_non_hisp_white = log(pct_non_hisp_white_2010 + 1) pct_non_hisp_white_zero = I(pct_non_hisp_white_2010 = 0)
pct_gq_2010	Percent population in group quarters. log_pct_gq = log(pct_gq_2010 + 1) pct_gq_zero = I(pct_gq_2010 = 0)
pct_gq_inst_2010	Percent institutionalized population in group quarters. log_pct_gq_inst = log(pct_gq_inst_2010 + 1) pct_gq_inst_zero = I(pct_gq_inst_2010 = 0)
pct_gq_noninst_2010	Percent noninstitutionalized population in group quarters. log_pct_gq_noninst = log(pct_gq_noninst_2010 + 1) pct_gq_noninst_zero = I(pct_gq_noninst_2010 = 0)
pct_pop_0_17_2010	Percent population under age 18. log_pct_pop_0_17 = log(pct_pop_0_17_2010 + 1) pct_pop_0_17_zero = I(pct_pop_0_17_2010 = 0)
pct_pop_65_over_2010	Percent population aged 65 and over. log_pct_pop_65_over = log(pct_pop_65_over_2010 + 1) pct_pop_65_over_zero = I(pct_pop_65_over_2010 = 0)

Table A.3: Variables from Land Use data.

Variable Name	**Description**
blk_mil_distance	Distance from block (in meters) from a military area landmark feature. mil_dist0 = I(blk_mil_distance = 0) mil_dist1 = I(0 < blk_mil_distance >= 1000) mil_dist2 = I(1000 < blk_mil_distance >= 2000) mil_dist3 = I(2000 < blk_mil_distance >= 3000) mil_dist4 = I(3000 < blk_mil_distance <= 4000) mil_dist5 = I(4000 < blk_mil_distance < 5000) mil_dist6 = I(blk_mil_distance = 5000)
blk_ua_distance	Distance from block (in meters) from the nearest urban area. ua_dist0 = I(blk_ua_distance = 0) ua_dist1 = I(blk_ua_distance = 1000) ua_dist2 = I(blk_ua_distance = 2000) ua_dist3 = I(blk_ua_distance = 3000) ua_dist4 = I(blk_ua_distance = 4000) ua_dist5 = I(blk_ua_distance = 5000)
blk_uni_distance	Distance from block (in meters) from the nearest university area landmark feature. uni_dist0 = I(blk_uni_distance = 0) uni_dist1 = I(0 < blk_uni_distance <= 1000) uni_dist2 = I(1000 < blk_uni_distance <= 2000) uni_dist3 = I(2000 < blk_uni_distance <= 3000) uni_dist4 = I(3000 < blk_uni_distance <= 4000) uni_dist5 = I(4000 < blk_uni_distance <= 5000) uni_dist6 = I(blk_uni_distance = 5000)
blk_mil_adj	Block is adjacent to a military area landmark feature.
blk_uni_adj	Block is adjacent to a university area landmark feature.
blk_mil_part	Block is partial within a military area landmark feature.
blk_uni_part	Block is partial within a university area landmark feature.
blk_mil_full	Block is completely within a military area landmark feature.
blk_uni_full	Block is completely within a university area landmark feature.

blk_mil_pct	Percent of block covered by the nearest military area landmark feature. log_mil_pct = log(blk_mil_pct + 1)
blk_np_pct	Percent of block covered by a national park area landmark feature. log_np_pct = log(blk_np_pct + 1)
blk_uni_pct	Percent of block covered by a university area landmark feature. log_uni_pct = log(blk_uni_pct + 1)
open_water_pct	Percent of block covered by NLCD open water. log_open_water_pct = log(open_water_pct + 1)
devel0_pct	Percent of block covered by NLCD developed, open space. log_devel0_pct = log(devel0_pct + 1)
devel1_pct	Percent of block covered by NLCD developed, low intensity. log_devel1_pct = log(devel1_pct + 1)
devel2_pct	Percent of block covered by NLCD developed, medium intensity. log_devel2_pct = log(devel2_pct + 1)
devel3_pct	Percent of block covered by NLCD developed, high intensity. log_devel3_pct = log(devel3_pct + 1)
barren_pct	Percent of block covered by NLCD barren land. log_barren_pct = log(barren_pct + 1)
forest1_pct	Percent of block covered by NLCD deciduous forest. log_forest1_pct = log(forest1_pct + 1)
forest2_pct	Percent of block covered by NLCD evergreen forest. log_forest2_pct = log(forest2_pct + 1)
forest3_pct	Percent of block covered by NLCD mixed forest. log_forest3_pct = log(forest3_pct + 1)
shrub_pct	Percent of block covered by NLCD shrub/scrub. log_shrub_pct = log(shrub_pct + 1)
grassland_pct	Percent of block covered by NLCD grassland/herbaceous. log_grassland_pct = log(grassland_pct + 1)
pasture_pct	Percent of block covered by NLCD pasture/hay. log_pasture_pct = log(pasture_pct + 1)
crops_pct	Percent of block covered by NLCD cultivated crops. log_crops_pct = log(crops_pct + 1)
wetlands1_pct	Percent of block covered by NLCD woody wetlands. log_wetlands1_pct = log(wetlands1_pct + 1)
wetlands2_pct	Percent of block covered by NLCD emergent herbaceous wetlands. log_wetlands2_pct = log(wetlands2_pct + 1)
public_pct	Percent of block for public lands. log_public_pct = log(public_pct + 1)

Table A.4: Variables from DSF stability data.

Variable Name	Description
dsf_si_*	DSF Stability Indices for spring/fall 2009 DSF refreshes in 2000–2009. log_dsf_si_spr09 = log(dsf_si_spr09 + 1) log_dsf_si_fal08 = log(dsf_si_fal08 + 1) log_dsf_si_spr08 = log(dsf_si_spr08 + 1) log_dsf_si_fal07 = log(dsf_si_fal07 + 1) log_dsf_si_spr07 = log(dsf_si_spr07 + 1) log_dsf_si_fal06 = log(dsf_si_fal06 + 1) log_dsf_si_spr06 = log(dsf_si_spr06 + 1) log_dsf_si_fal05 = log(dsf_si_fal05 + 1) log_dsf_si_spr05 = log(dsf_si_spr05 + 1) log_dsf_si_fal04 = log(dsf_si_fal04 + 1) log_dsf_si_spr04 = log(dsf_si_spr04 + 1) log_dsf_si_fal03 = log(dsf_si_fal03 + 1) log_dsf_si_spr03 = log(dsf_si_spr03 + 1) log_dsf_si_fal02 = log(dsf_si_fal02 + 1) log_dsf_si_spr02 = log(dsf_si_spr02 + 1) log_dsf_si_fal01 = log(dsf_si_fal01 + 1) log_dsf_si_spr01 = log(dsf_si_spr01 + 1) log_dsf_si_fal00 = log(dsf_si_fal00 + 1) log_dsf_si_spr00 = log(dsf_si_spr00 + 1)
stability_index	Overall stability index.

Table A.5: Variables from LEHD Origin-Destination Employment Statistics (LODES) data.

Variable Name	Description
grow_rac_c000	Annual growth of all jobs from 2007 to 2008 (residential tabulation).
grow_rac_ca01	Annual growth of $<$ age 30 working from 2007 to 2008 (residential tabulation).
grow_rac_ca03	Annual growth of $\geq$ age 55 working from 2007 to 2008 (residential tabulation).
grow_rac_ce01	Annual growth of low earnings group from 2007 to 2008 (residential tabulation).
grow_rac_ce03	Annual growth of low earnings group from 2007 to 2008 (residential tabulation).
grow_wac_c000	Total job growth from 2007 to 2008 (workplace tabulation).
grow_wac_cns07	Annual growth of retail sector from 2007 to 2008 (workplace tabulation).
share_rac_ca01	Share of all jobs that were low earnings in 2008.
share_rac_ca03	Share of all jobs that were $\geq$ age 55 in 2008.
share_rac_ce01	Share of all jobs that were low earnings in 2008.
share_rac_ce03	Share of all jobs that were high earnings in 2008.
lodes_gini	Gini coefficient for LODES earnings groups.

Table A.6: Variables from RealtyTrac data.

Variable Name	Description
realtrac_sum_2005	Number of foreclosed homes within the block in 2005. realtrac_0_2005 = I(realtrac_sum_2005 = 0) realtrac_1_5_2005 = I(1 <= realtrac_sum_2005 <= 5) realtrac_6_10_2005 = I(6 <= realtrac_sum_2005 <= 10) realtrac_11plus_2005 = I(realtrac_sum_2005 >= 11)

realtrac_sum_2006	Number of foreclosed homes within the block in 2006. realtrac_0_2006 = I(realtrac_sum_2006 = 0) realtrac_1_5_2006 = I(1 <= realtrac_sum_2006 <= 5) realtrac_6_10_2006 = I(6 <= realtrac_sum_2006 <= 10) realtrac_11plus_2006 = I(realtrac_sum_2006 >= 11)
realtrac_sum_2007	Number of foreclosed homes within the block in 2007. realtrac_0_2007 = I(realtrac_sum_2007 = 0) realtrac_1_5_2007 = I(1 <= realtrac_sum_2007 <= 5) realtrac_6_10_2007 = I(6 <= realtrac_sum_2007 <= 10) realtrac_11plus_2007 = I(realtrac_sum_2007 >= 11)
realtrac_sum_2008	Number of foreclosed homes within the block in 2008. realtrac_0_2008 = I(realtrac_sum_2008 = 0) realtrac_1_5_2008 = I(1 <= realtrac_sum_2008 <= 5) realtrac_6_10_2008 = I(6 <= realtrac_sum_2008 <= 10) realtrac_11plus_2008 = I(realtrac_sum_2008 >= 11)

Table A.7: Variables from IRS 1040 data.

Variable Name	**Description**
irs1040nm	Number of 1040 forms in a block that had no block ID. log_irs1040nm = log(irs1040nm + 0.001)
irs1040nb	Number of 1040 forms in a block that had no MAFID. log_irs1040n = log(irs1040nb + 0.0005)
irs1040ng	Number of 1040 forms in a block that had no block ID and no MAFID. log_irs1040ng = log(irs1040ng + 0.005)

Table A.8: Initial Bernoulli model, with LogLik = -43,815.61 and AIC = 87,663.22.

	Estimate	SE	z-value	p-value
Intercept	1.7590	0.0695	25.296	<2e-16
log_dep_list	-0.2998	0.0346	-8.667	<2e-16
log_landmeters2	-0.1254	0.0096	-13.021	<2e-16
log_eds_res_sum	-0.2051	0.0133	-15.394	<2e-16
log_landmeters2_sq	0.0176	0.0019	9.346	<2e-16
log_business_sum	-0.0171	0.0159	-1.075	0.2824
log_acs_hu_ratio	-0.0426	0.0783	-0.544	0.5863
urbanZERO	-0.2929	0.0295	-9.938	<2e-16
teaMOM	0.2884	0.0247	11.670	<2e-16
teaUER	-0.2748	0.0622	-4.419	9.93e-06
log_gc_sum	-0.0377	0.0433	-0.869	0.3848
hasSeasonalY	-0.3350	0.0782	-4.284	1.83e-05
log_unitstat1_sum	0.1546	0.0375	4.121	3.77e-05
log_delptypeBk_sum	-0.6691	0.0112	-55.783	<2e-16
log_isVacantY_sum	-0.1393	0.0143	-9.706	<2e-16
log_hu_density_ratio	-0.0137	0.0074	-1.865	0.0622

Table A.9: Selected Bernoulli model, with LogLik = -41,912.09 and AIC = 83,902.17.

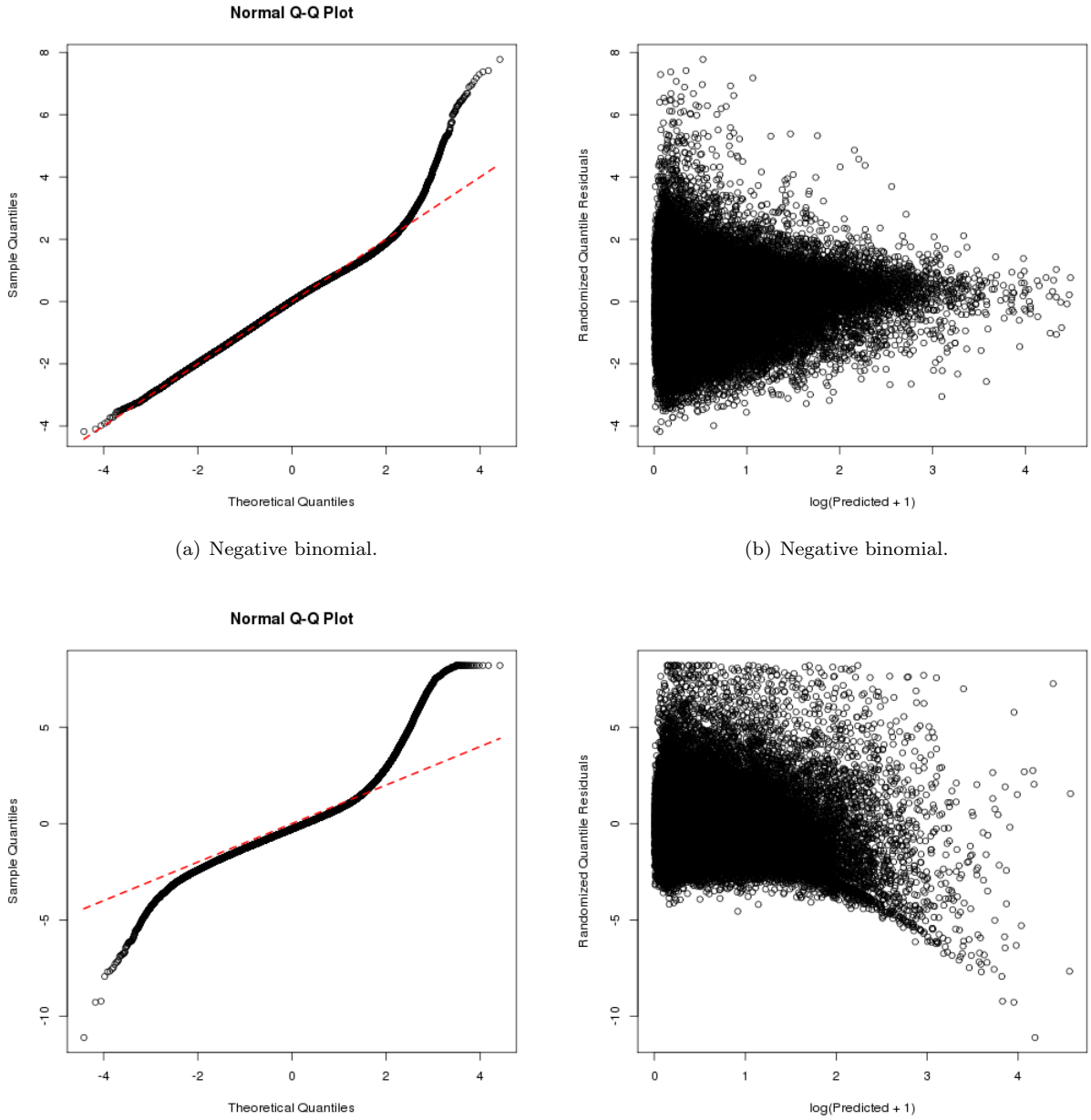
	Estimate	SE	z-value	p-value
Intercept	0.8684	0.0940	9.236	<2e-16
log_dep_list	-0.1708	0.0275	-6.215	5.13e-10
log_landmeters2	0.0042	0.0119	0.358	0.7206
log_eds_res_sum	-0.1859	0.0145	-12.848	<2e-16
log_landmeters2_sq	0.0228	0.0021	10.624	<2e-16
teaMOM	0.1582	0.0266	5.958	2.56e-09
hasSeasonalY	-0.3073	0.0799	-3.848	0.0001
log_unitstat1_sum	0.1384	0.0331	4.182	2.89e-05
log_delptypeBk_sum	-0.8565	0.0307	-27.879	<2e-16
log_isVacantY_sum	-0.1648	0.0151	-10.896	<2e-16
log_mafsrc1_sum	-0.1096	0.0106	-10.362	<2e-16
log_compcity1_sum	0.0754	0.0263	2.870	0.0041
log_forest1_pct	-0.0333	0.0079	-4.208	2.57e-05
log_forest2_pct	-0.0446	0.0099	-4.519	6.22e-06
log_forest3_pct	-0.0568	0.0130	-4.368	1.26e-05
log_irs1040ng	-0.2191	0.0129	-16.991	<2e-16
log_pct_crowd_occp_u	-0.3511	0.0210	-16.732	<2e-16
log_crops_pct	0.1069	0.0083	12.882	<2e-16
log_dsf_si_spr09	-1.2184	0.0469	-25.953	<2e-16
log_shrub_pct	-0.0801	0.0092	-8.685	<2e-16
log_devel0_pct	-0.0223	0.0071	-3.136	0.0017
log_devel1_pct	-0.0614	0.0133	-4.625	3.74e-06
log_devel2_pct	0.1226	0.0162	7.588	3.26e-14
log_devel3_pct	0.0105	0.0108	0.971	0.3317
stability_index	0.5374	0.0397	13.539	<2e-16
hu_block2tract_ratio	-1.6410	0.5177	-3.169	0.0015
log_pct_pop_0_17	0.2240	0.0177	12.663	<2e-16
log_irs1040nb	-0.0942	0.0071	-13.264	<2e-16
log_irs1040nm	-0.0057	0.0142	-0.401	0.6881
log_htc	-0.0638	0.0101	-6.290	3.18e-10
log_pct_mlt_u_10p_strc	0.1187	0.0160	7.432	1.07e-13
log_pct_not_single_u_strc	-0.1359	0.0221	-6.139	8.28e-10
log_pct_black	0.0807	0.0107	7.539	4.72e-14
log_compcity1_sum:log_devel1_pct	0.0488	0.0048	10.169	<2e-16
log_dep_list:log_dsf_si_spr09	0.0869	0.0066	13.099	<2e-16
log_landmeters2:log_dsf_si_spr09	-0.0621	0.0039	-16.050	<2e-16
log_delptypeBk_sum:log_dsf_si_spr09	0.1042	0.0089	11.665	<2e-16
log_dsf_si_spr09:log_irs1040nm	-0.0714	0.0066	-10.786	<2e-16
log_devel2_pct:log_irs1040nb	0.0269	0.0031	8.653	<2e-16

Table A.10: Initial negative binomial, with LogLik = -88,685.72 and AIC = 177,405.40. Theta parameter represents $1/\kappa$.

	Estimate	SE	z-value	p-value
Intercept	-1.1624	0.0683	-17.026	<2e-16
log_dep_list	0.0973	0.0300	3.247	0.0012
log_landmeters2	0.0435	0.0098	4.421	9.82e-06
log_eds_res_sum	0.2051	0.0134	15.311	<2e-16
log_landmeters2_sq	-0.0208	0.0018	-11.297	<2e-16
log_business_sum	0.1102	0.0157	7.024	2.15e-12
log_acs_hu_ratio	-0.4374	0.0723	-6.052	1.43e-09
urbanZERO	0.1826	0.0300	6.095	1.09e-09
teaMOM	-0.5804	0.0257	-22.625	<2e-16
teaUER	0.3695	0.0643	5.747	9.08e-09
log_gc_sum	0.6218	0.0408	15.245	<2e-16
hasSeasonalY	0.7288	0.0728	10.011	<2e-16
log_unitstat1_sum	-0.3775	0.0355	-10.645	<2e-16
has_delptypeBk	0.8436	0.0200	42.281	<2e-16
log_isVacantY_sum	0.2474	0.0142	17.486	<2e-16
log_hu_density_ratio	-0.0754	0.0075	-10.103	<2e-16
Theta	0.2093	0.0022	---	---

Table A.11: Selected negative binomial model, with LogLik = -85,970.86 and AIC = 172,031.7. Theta parameter represents $1/\kappa$.

	Estimate	SE	z-value	p-value
Intercept	0.4500	0.0899	5.003	5.64e-07
log_dep_list	0.5199	0.0317	16.377	<2e-16
log_landmeters2	-0.0356	0.0118	-3.016	0.002559
log_eds_res_sum	-0.2224	0.0285	-7.792	6.61e-15
log_landmeters2_sq	-0.0239	0.0020	-11.930	<2e-16
log_business_sum	0.0521	0.0162	3.208	0.0013
teaMOM	-0.1676	0.0269	-6.239	4.40e-10
teaUER	0.3175	0.0611	5.201	1.99e-07
log_gc_sum	0.2122	0.0388	5.472	4.44e-08
hasSeasonalY	0.4885	0.0679	7.191	6.41e-13
log_unitstat1_sum	-0.2137	0.0307	-6.949	3.67e-12
has_delptypeBk	0.4839	0.0206	23.545	<2e-16
log_isVacantY_sum	0.1919	0.0138	13.895	<2e-16
log_hu_density_ratio	-0.2365	0.0132	-17.930	<2e-16
stability_index	-0.7996	0.0384	-20.815	<2e-16
log_irs1040ng	0.2644	0.0126	20.998	<2e-16
log_irs1040nb	0.0878	0.0056	15.543	<2e-16
log_devel0_pct	0.0492	0.0067	7.291	3.08e-13
log_devel1_pct	0.1526	0.0118	12.979	<2e-16
log_devel2_pct	0.0702	0.0071	9.894	<2e-16
log_devel3_pct	0.0196	0.0105	1.870	0.0615
log_crops_pct	-0.1031	0.0082	-12.595	<2e-16
log_pct_crowd_occp_u	0.3097	0.0200	15.489	<2e-16
log_pct_pop_0_17	-0.3540	0.0202	-17.493	<2e-16
log_pct_not_single_u_strc	0.2156	0.0188	11.479	<2e-16
log_forest1_pct	0.0277	0.0077	3.599	0.0003
log_forest2_pct	0.0181	0.0097	1.867	0.0618
log_forest3_pct	0.0511	0.0128	3.999	6.36e-05
log_dsf_si_spr00	-0.1136	0.0145	-7.854	4.03e-15
log_shrub_pct	0.0777	0.0092	8.484	<2e-16
log_dsf_si_spr09	0.2290	0.0177	12.962	<2e-16
pct_unemploy_zero	-0.3620	0.0456	-7.936	2.08e-15
log_pct_li_hh_indo_europe	0.3034	0.0321	9.460	<2e-16
log_irs1040nm	0.0916	0.0098	9.338	<2e-16
log_pct_mlt_u_2p_strc	-0.1537	0.0161	-9.557	<2e-16
realtrac_1_5_2007	-0.2445	0.0325	-7.534	4.91e-14
realtrac_11plus_2007	0.3688	0.0828	4.453	8.45e-06
log_pct_api	0.1499	0.0212	7.069	1.56e-12
uni_dist0	0.5695	0.0990	5.752	8.81e-09
uni_dist2	0.1318	0.0392	3.366	0.0008
log_dep_list:log_devel1_pct	-0.0798	0.0041	-19.247	<2e-16
log_landmeters2:log_dsf_si_spr00	0.0554	0.0033	16.615	<2e-16
log_unitstat1_sum:log_hu_density_ratio	0.0632	0.0048	13.284	<2e-16
log_eds_res_sum:stability_index	0.4239	0.0359	11.815	<2e-16
Theta	0.2580	0.0028	---	---



(a) Negative binomial.

(b) Negative binomial.

(c) Poisson.

(d) Poisson.

Figure A.1: Randomized quantile residuals computed on the training set using negative binomial and Poisson regressions.

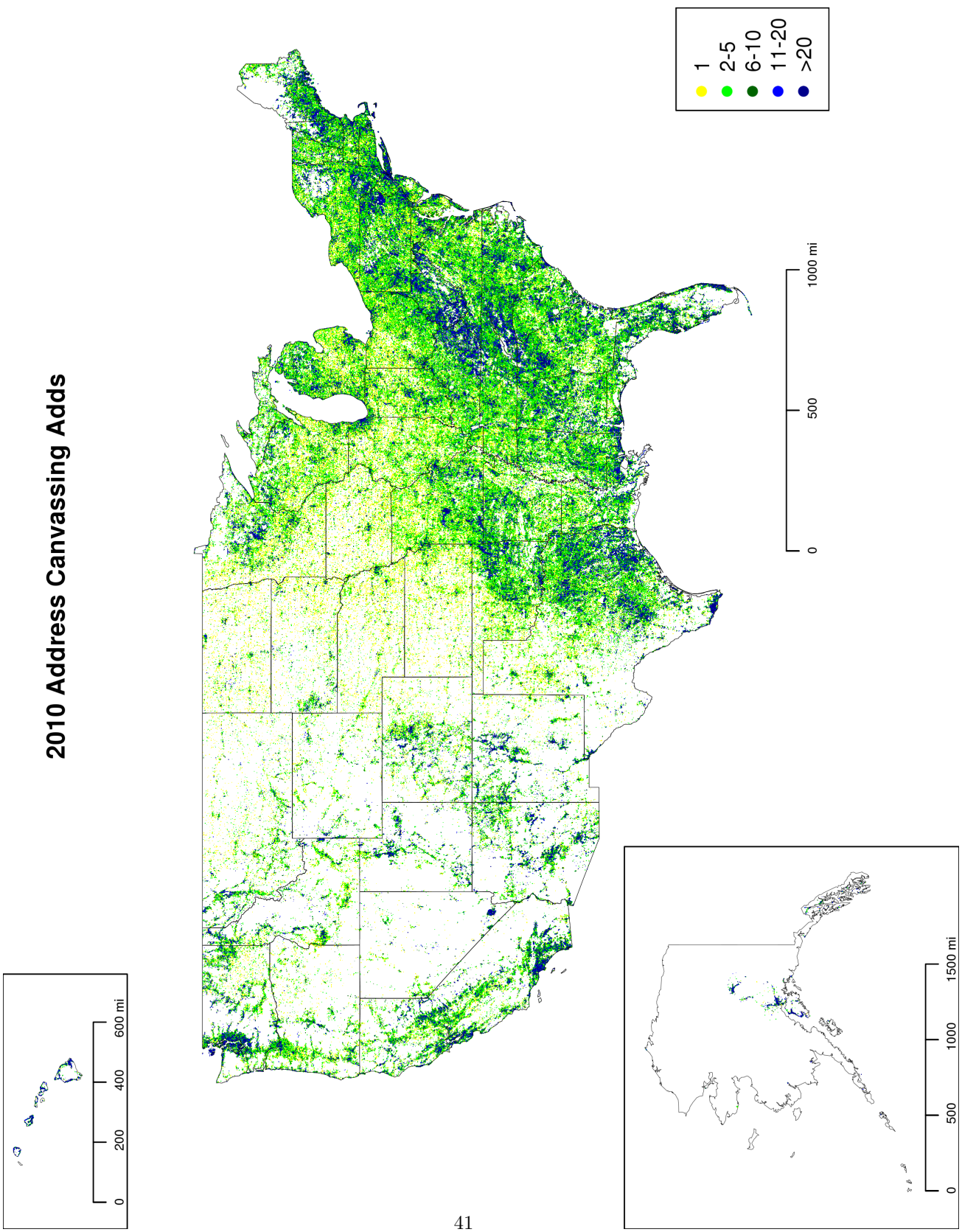


Figure A.2: 2010 Address Canvassing adds.

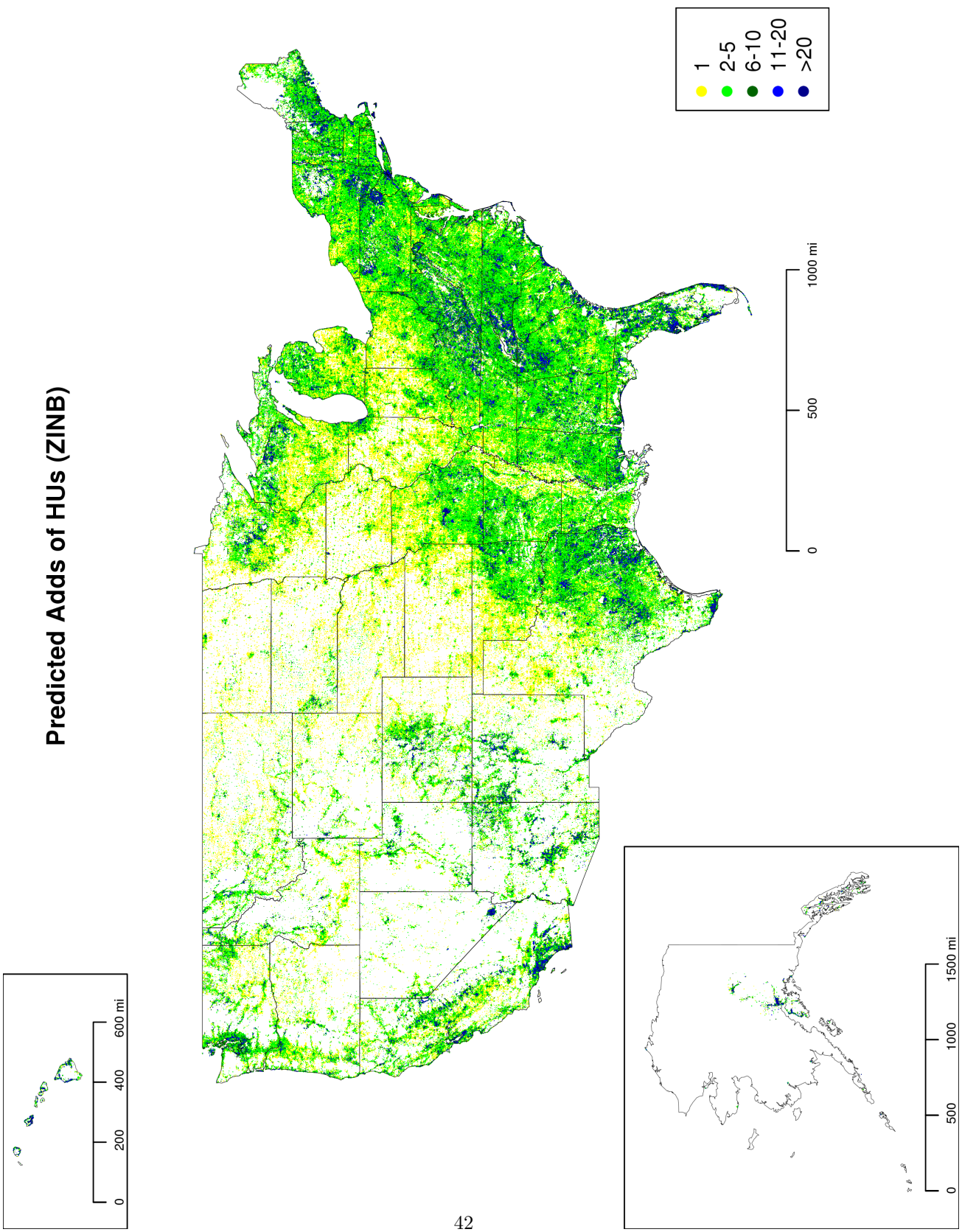


Figure A.3: Predicted adds based on ZINB model.

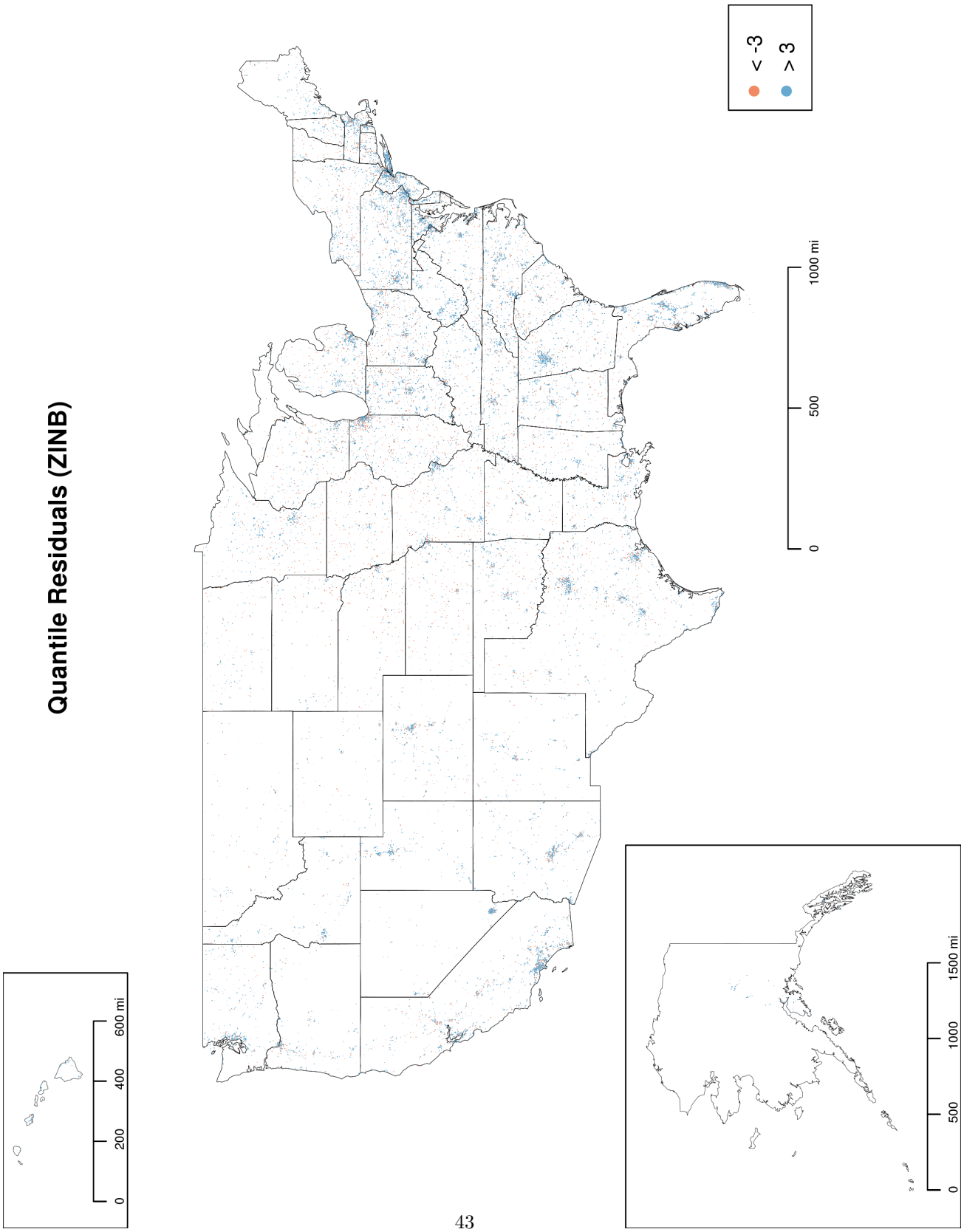


Figure A.4: "Bad" quantile residuals.