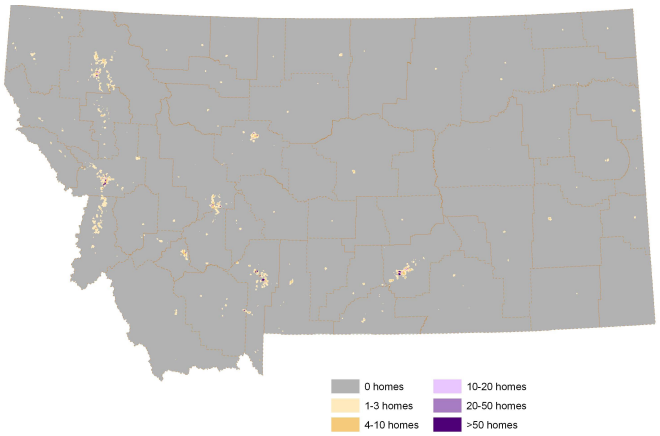
A Model for Housing Development in Montana



Prepared for:
Patty Gude
Headwaters Economics
P.O. Box 7059
Bozeman, MT 59771
Ph: 406-599-7425 ; Email: patty@headwaterseconomics.org

Prepared by:
Trent L. McDonald, Ph.D.
Western EcoSystems Technology, Inc.
2003 Central Avenue
Cheyenne, Wyoming
Ph: 307-634-1756 ; Email: tmcdonald@west-inc.com

December 1, 2010

WEST Inc.

Environmental and Statistical Consultants

1 Introduction

WEST Inc. was contracted by Headwaters Economics to construct a predictive model for the number of new housing units constructed between 2000 and 2008 in the state of Montana. The purpose of this model was first to identify landscape characteristics correlated with new house construction, and second to forecast the number of new houses during the next 9 years (2008 - 2016). These predictions can then be used to identify "hot-spots" or other areas of large growth.

For estimation of the model, Headwaters Economics provided a data set containing the number of new houses constructed during 2000 - 2008 in every quarter section of Montana. A description of the 17 variables contained in the data set appears in Table 1. This data set contained $362,585$ records, one for each quarter section containing private land in the state of Montana. The response variable of interest was the number of new homes built on any particular quarter section during 2000 - 2008 (i.e., `homeschg`, Table 1). This response took on values from 0 to 523 , with 96% of the responses statewide equal to zero.

In this report, we describe the statistical methods WEST Inc. used to construct the predictive model for number of homes on a quarter section. These methods include an initial exploratory data analysis, estimation of 4 types of models, model validation, and forecasting.

2 Methods

2.1 Exploratory Data Analysis

The purpose of exploratory data analysis was to identify general characteristics of the data. Two explorations were performed: mapping of non-zero counts, and characterization of missing values. All non-zero counts of the primary response variables, `homeschg`, were mapped and inspected for broad patterns. We looked for extreme clumping, sparsity, or broad trends that could influence estimation of the primary models. The second exploratory analysis identified the variables and observations with large numbers of missing values and the patterns of those missing values. This information was used to decide how best to treat missing values.

2.2 Models

Prior to model building, all covariates considered for inclusion in the model were scaled to have zero mean and unit variances. The equation for scaling a covariate was,

$$x^* = \frac{x - \bar{x}}{std(x)}, \tag{1}$$

where x^* was the scaled variable to be fitted in models. Scaling was performed because covariates in the data set measured drastically different quantities over differing ranges. Scaling stabilized model

estimation by making variation comparable among covariates and thus contributions to the model by different variables were comparable.

The following four regression models were fitted to `homeschg`: 1) a standard linear model, 2) a linear model after log transforming the response, 3) a generalized linear model (GLM) with a $\log$ link that assumed `homeschg` followed a Poisson distribution, and 4) a zero-inflated Poisson (ZIP) model. A GLM assuming `homeschg` followed a negative binomial likelihood was also considered but variation in `homeschg` was large and the numerical fitting routine would not converge. A ZIP model that assumed a negative binomial likelihood for non-zero values was also attempted, but resulted in fitted values that were many orders of magnitude (i.e., 10000$\times$) larger than observed. Each of the 4 successful models is described next.

2.2.1 Linear model

The linear model was,

$$E[y_i] = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$$

where $y_i =$ `homesch` for the i^{th} quarter section containing private land in Montana, x_{ij} was the value of the j^{th} covariate for the i^{th} quarter section, and β_j was the estimated coefficient for the j^{th} covariate. Here, the residuals $(y_i - E[y_i])$ were assumed to following a *normal* distribution with mean 0 and covariance matrix σI.

2.2.2 Log-linear model

The log-linear model was,

$$E[\ln(y_i)] = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}.$$

Here, the residuals $(\ln(y_i) - E[\ln(y_i)])$ were assumed to following a *normal* distribution with mean 0 and covariance matrix σI.

2.2.3 Generalized linear model

The generalized linear model relaxed the requirement that residuals follow a `normal` distribution and assumed a $\log$ link function. The generalized linear model was,

$$\ln(E[y_i]) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}.$$

Here, the observed counts y_i were assumed to be independent and follow a *Poisson* distribution with mean $E[y_i]$.

2.2.4 Zero-inflated Poisson model

The zero-inflated Poisson model attempted to account for the large proportion of zeros in the data. Under this model, the random process resulting in a certain number of houses being added to a quarter section was assumed to have two phases. First, the process "decides" whether to place additional houses

on a quarter section. If so, the process then "decides" how many to put on the quarter section. Under this model, zeros can arise in two ways. Zeros can occur because the process "decided" not to put additional houses on the quarter section. Or, zeros can occur because the process has not yet added houses to the quarter section even though it previously "decided" to place houses there.

Statistically, the ZIP model assumed that the distribution of house additions was a mixture of a point mass at $y_i = 0$ and a Poisson distribution. The ZIP mixture distribution had density function,

$$f(y_i) = \omega^{I(y_i=0)} + (1-\omega_i)^{1-I(y_i=0)} \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!},$$

where $I(y_i = 0)$ is an indicator function equal to 1 if $y_i = 0$, and 0 otherwise. Under this model the mean and variance of y_i were

$$\begin{aligned} E[y_i] &= (1-\omega_i)\lambda_i \\ Var(y_i) &= E[y_i] + \left(\frac{\omega_i}{1-\omega_i}\right) E[y_i]^2 \end{aligned}$$

As in the previous models, the ZIP model assumed counts y_i were independent.

To relate parameters of the ZIP distribution to study covariates, separate linear models were proposed for λ_i and ω_i, i.e,.

$$\begin{aligned} \log(\lambda_i) &= \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} \\ \log\left(\frac{\omega_i}{1-\omega_i}\right) &= \gamma_0 + \gamma_1 z_{i1} + \ldots + \gamma_q z_{iq}. \end{aligned}$$

2.3 Estimation

All estimates were computed by the method of maximum likelihood, and took place using the R statistical software package (`http://www.r-project.org`). The linear, log-linear, and generalized linear model used the built-in functions `lm()` and `glm()`. Estimation of coefficients in the ZIP model used the function `zeroinfl()` in the pscl library. The R code used to fit the models and perform model validation is included in Appendix A.

2.4 Model Selection

For the linear, log-linear, and GLM model, the set of variables included in the best fitting model was selected using backward stepwise elimination. Backward elimination utilized AIC to assess the predictive strength and utility of individual variables in the model, and stopped when it was no longer possible to remove a variable and decrease AIC. The initially full model contained `homes_1mi`, `home8yr1mi`, `water_dist`, `rd_dens`, `mjrd_dist`, `ecoregion`, `summit_den`, `pcinc`, `pop_dens`, `ap_tt`, and `town_tt` (Table 1). The two homes variables, `home` and `homes8yr`, were not considered for inclusion due to high correlation with their counterparts computed on a 1 mile buffer surrounding the quarter section ($r = 0.85$ between `home` and `homes_1mi`; $r = 0.78$ between `homes8yr` and `home8yr1mi`). Because these home

variables were deemed to be measuring similar characteristics, the values computed on a 1 mile buffer surrounding the quarter section were chosen for potential inclusion because they encompassed more area and contained more variation than their counterparts computed on a single quarter section. Geographic coordinates were not considered for inclusion because it was felt that the covariates already being considered adequately quantified spatial characteristics important to housing development in Montana.

Contrary to the other models, the ZIP model was computationally expensive and unstable. For these reasons, stepwise selection was not used to select variables in the ZIP model. Covariates in the ZIP model were chosen after backward selection of the other models was complete. The set of covariates in the other models was inspected and a judgement was made as to which were appropriate to include in a ZIP model designed to forecast future development. The covariates included in the ZIP model for λ were `home8yr1mi`, `water_dist`, `rd_dens`, `mjrd_dist`, `ecoregion`, `summit_den`, `pcinc`, `pop_dens`, `ap_tt`, and `town_tt`. The covariates included in the ZIP model for ω were `home8yr1mi` and `town_tt`.

Once a model of each type was estimated, a combination of *k*-fold cross validation and visual inspection was used to select the final model. *K*-fold cross validation first divided the data set into $k = 10$ equally sized sets of observations, then excluded each set in turn and re-estimated the model on the remaining $k - 1 = 9$ sets. Mean square predictive error (MSPE) of the refitted model for the excluded observations was then computed and summed over the *k* sets, i.e.,

$$MSPE = \sum_{j=1}^{k} \sqrt{\sum_{i=1}^{n_j} \frac{(y_{ij} - \hat{y}_{ij})^2}{n-1}}.$$

where n_j was the number of observations in the j^{th} excluded set ($n_j \approx n/k$), y_{ij} was the value of *homeschg* for the i^{th} observation in the j^{th} excluded set and $\hat{y}_{ij}$ was the predicted value for y_{ij} computed by a model that was estimated using a data set that excluded the j^{th} set of observations. Because observations were highly skewed, with large values in or near cities, and because prediction accuracy in rural or undeveloped areas was paramount, values of *homeschg* > 20 were excluded when computing MSPE. Observed values greater than 20 were including during model estimation.

The observed and predicted values of *homeschg* were also mapped and inspected. During inspection, spatial extent and appropriateness of the predictions were the primary qualities being assessed. Following k-fold cross validation and visual inspection, a final model from among the 4 types was chosen.

2.5 Forecasts

To map predictions of the future number of houses that will be built on a quarter section during 2008 - 2016, appropriate values of all covariates in the final model must be known or estimated. Of the variables potentially included in the final model, only the home count variables (`homes_1mi` and `home8yr1mi`) change drastically over the course of a decade. To make predictions of housing development between 2008 and 2016, the values of `homes_1mi` and `home8yr1mi` were updated to cover the period 2000 - 2008 and used to make predictions. Past values of all other covariates in the final model were propagated forward for predictions. All predicted values were truncated at zero if necessary.

In addition to mapping the predicted average number of houses built on a quarter section over an infinite number of future 9 year periods, a 95% lower prediction limit was also computed. The predicted

average constituted a best point estimate of the number of houses to be built on a quarter section during 2008 - 2016. The 95% lower prediction limit can be interpreted as follows: if L is the 95% lower prediction limit for a particular quarter section, researchers can be 95% sure that the true number of houses built on that quarter section during 2008 - 2016 will be greater than L. From predictions for periods 2000 - 2008 and 2008 - 2016, home building acceleration or deceleration can be computed. If $\hat{y}_i^{08}$ and $\hat{y}_i^{16}$ are the predicted number of homes added during the respective 9 year period, acceleration values were computed as the difference $\hat{y}_i^{16} - \hat{y}_i^{08}$.

3 Results

3.1 Exploratory Data analysis

Figure 1 shows a map of observed `homeschg` for Montana. Larger counts of additional homes were clustered in and around cities like Billings, Bozeman, Missoula, and Helena. Other than clustering around towns, the density of non-zero counts generally increased in the west and south-west parts of Montana relative to eastern Montana.

Five covariates in the data set contained smaller numbers of missing values due to representational errors inherent in the GIS datasets. The covariates `pcinc` and `pop_dens` each had 277 missing values (0.076%), while `rd_dens`, `mjrd_dist`, `ap_tt` each had 274 missing values. The covariates `water_dist`, `summit_den`, and `town_tt` were missing 273, 97, and 70 times, respectively. The missing values in these covariates were nested in a descending order. For example, all observations with missing values for `summit_den` also had missing values for `water_dist`, and all those observations had missing values for `ap_tt`, and so on. By removing observations (rows) that were missing for the variable with the largest number, all rows containing missing observations were removed. Consequently, all rows with missing values for `pcinc` were removed from the data set prior to modeling. The final modeling data set contained $n = 362,308$ records (quarter sections). Means and standard deviations used to standardize the variables in the final modeling data set appear in Table 2.

3.2 Model Selection

The estimated models and cross-validation scores for each of the estimated models appear in Table 3. The smallest cross-validation prediction error resulted from the GLM model (MSPE $= 1.78$), followed by the ZIP (MSPE $= 2.04$) and linear model (MSPE $= 2.15$). The high value of MSPE for the log-linear model (MSPE $= 22.7$) was caused by a few very large predictions. Despite small values of average prediction error, inspection of the observed and predicted maps (Figure 2) revealed that the GLM and ZIP model did not predict housing growth in isolated areas away from cities and roads. Housing growth, however, does occur in isolated areas away from cities (Figure 1), and the linear model produced better predictions in these areas. Because accurate predictions away from cites was paramount to the study, the linear model was chosen as the best and final forecasting model to use.

In the final linear model, the number of houses built between 2000 and 2008 was **positively** related to the following variables:

- Number of homes built during the previous 9 year period calculated over a 1 mile buffer surrounding the quarter section (`homes8yr1mi`),
- Mean per capita income in 2000 (`pcinc`),
- Road density within 1 mile of the quarter section (`rd_dens`), and
- Travel time to the nearest town (`town_tt`).

The number of homes was **negatively** related to the following variables:

- Absolute number of homes in 2000 in the quarter section and a 1 mile buffer surrounding it (`homes_1mi`),
- Population density in 2000 (`pop_dens`), and
- The number of mountain peaks within 10 miles of the quarter section (`summit_den`).

By ecoregion, growth was highest in the "NW Glaciated" region, followed by "NW Plains", followed by "N Rockies".

Because all variables were standardized prior to modeling, Wald t-ratios ($Coef/SE$) can be used to assess the relative strength of each variable for prediction. Among variables in the final model, past home growth (`homes8yr1mi`) was by far the strongest predictor of future growth ($t = 275.8$). The number of homes already in the surrounding area (`homes_1mi`), and road density (`rd_dens`) were also strong predictors of future growth ($t = -11.4$ for `homes_1mi`; $t = 10.3$ for `rd_dens`). The next highest Wald t-ratio was 5.8 for `town_tt`.

3.3 Forecasts

By updating the home count variables to cover the period 2000 to 2008 (rather than 1999 to 2000), a forecast of growth during the period 2008 to 2016 was made. A map of predicted growth in number of houses during 2008 - 2016 appears in Figure 3. From Figure 3, it is clear that the majority of growth is predicted to occur around existing towns. Other areas predicted to have high growth are those in relatively developed valleys (e.g., near Missoula or Helena) where road density is relatively high. Acceleration values for the area around Bozeman (Figure 4) shows approximately equal proportions of areas with increased and decreased rates of home building.

4 Tables and Figures

Table 1: All variables in the data base used to develop a model for number of homes built in quarter sections of Montana.

Variable	Description
home	Number of homes present in 2000 in each quarter section
homes8yr	Number of homes built between 1992 and 2000 in each quarter section
homeschg	(*response*) Number of homes built between 2000 and 2008 in each quarter section
homes_1mi	Number of homes per quarter section in 2000 within 1 mile of each quarter section
home8yr1mi	Number of homes per quarter section built between 1992 and 2000 within 1 mile of each quarter section
water_dist	Mean Euclidian distance in meters to nearest major body of water
rd_dens	Mean road density within 1 mile of each quarter section, expressed as miles of road per square mile of area
mjrd_dist	Mean Euclidian distance in meters to nearest major road (interstates, state highways, primary and secondary roads)
ecoregion	1 = Northwestern Glaciated Plains (N and NE Montana); 2 = Northwestern Great Plains (E and parts of central Montana); 3 = Northern Rockies (all of W Montana)
summit_den	Mean number of mountain peaks per square mile within 10 miles of each quarter section
pcinc	Mean per capita income in 2000 in each quarter section
pop_dens	Mean population density per square mile in 2000 in each quarter section
ap_tt	Mean travel time in minutes to the nearest town with large commercial airport from each quarter section
town_tt	Mean travel time in minutes to the nearest town from each quarter section
x_coord	X coordinate of quarter section centroid, in meters, State Plane Coordinate System 1983
y_coord	Y coordinate of quarter section centroid, in meters, State Plane Coordinate System 1983

Table 2: Means and standard deviations used to standardize variables in preparation for modeling.

Variable	Mean	Standard Dev.
homes	0.9277	12.9928
homes8yr	0.1398	2.2850
homes_1mi	3.7854	35.7437
home8yr1mi	0.5663	5.2460
water_dist	12438.1609	10912.3369
rd_dens	0.9154	1.3298
mjrd_dist	7252.6585	6914.3259
summit_den	0.0047	0.0103
pcinc	15639.7070	3029.9990
pop_dens	10.8174	329.7255
ap_tt	163.1384	102.5372
town_tt	61.2857	42.8334
x_coord	643273.0600	226175.2461
y_coord	315260.3681	125141.6184

Table 3: Standardized coefficients, standard errors, and cross-validation score (MSPE) for the 4 primary house models. Standardized coefficients result from fitting variables that have been standardized using the appropriate mean and variance contained in Table 2.

	Linear		**Log-linear**		**GLM**		**ZIP**	
Variable	Coef	SE	Coef	SE	Coef	SE	Coef	SE
(Intercept)	0.1700	0.0086	0.0233	0.0006	-6.7110	0.0295	-2.5683	0.0426
ap_tt					-0.7494	0.0133	-0.6858	0.0142
ecoregionN Rockies	-0.0656	0.0173	0.0365	0.0013	0.6767	0.0208	0.3648	0.0234
ecoregionNW Plains	-0.0045	0.0109	0.0193	0.0008	0.6755	0.0206	0.4843	0.0234
home8yr1mi	1.5803	0.0057	0.0968	0.0004	0.0457	0.0003	0.0390	0.0002
homes_1mi	-0.0776	0.0068	0.0295	0.0005	0.0006	0.0016		
mjrd_dist			-0.0036	0.0004	0.0889	0.0139	0.1474	0.0144
pcinc	0.0211	0.0049	0.0100	0.0004	0.1163	0.0025	0.0905	0.0027
pop_dens	-0.0192	0.0049	-0.0053	0.0004	-0.1405	0.0030	-0.1207	0.0023
rd_dens	0.0735	0.0072	0.0381	0.0005	0.2743	0.0026	0.2012	0.0021
summit_den	-0.0116	0.0058	0.0008	0.0004	0.0298	0.0039	0.0421	0.0041
town_tt	0.0317	0.0055	-0.0052	0.0005	-3.3033	0.0247	-1.4771	0.0313
water_dist			-0.0049	0.0004	-0.4787	0.0114	-0.3754	0.0114
MSPE	2.1571		22.7008		1.7871		2.0412	

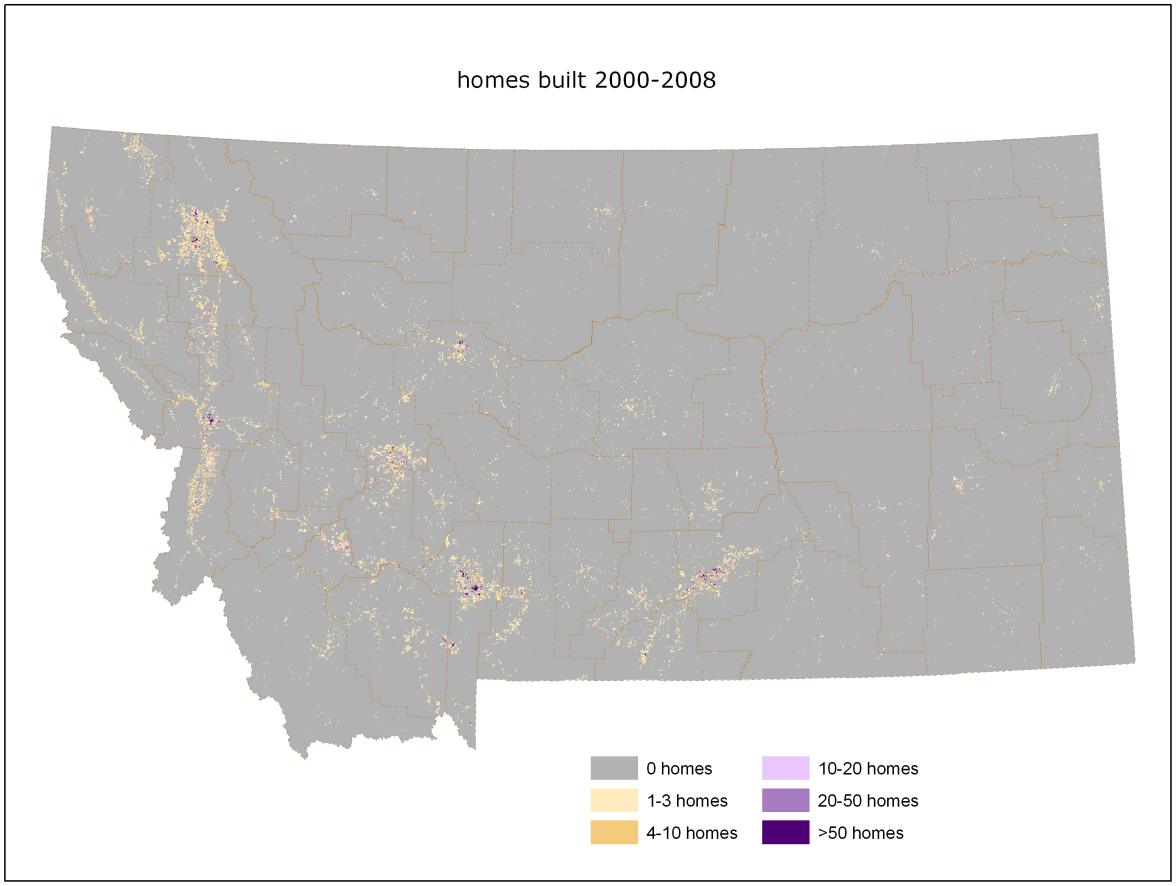
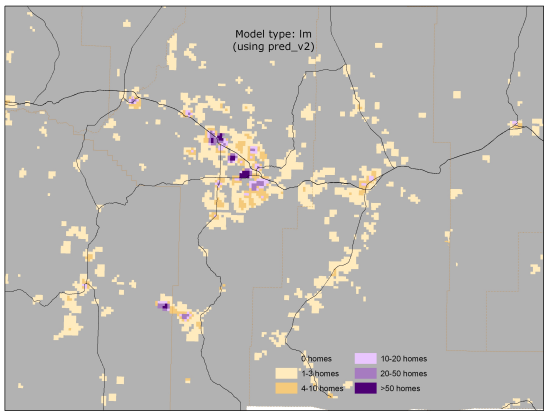
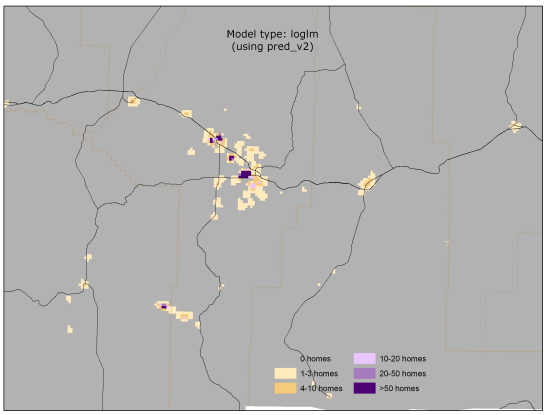


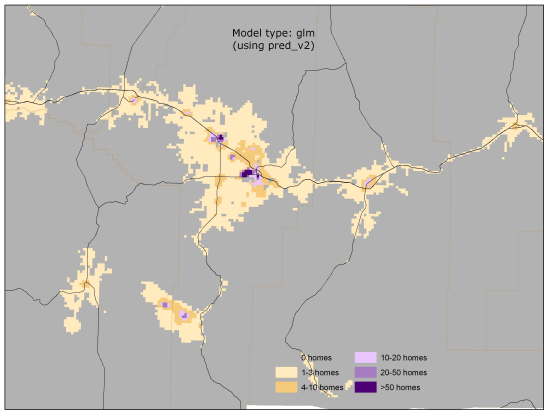
Figure 1: Observed number of homes built between 2000 and 2008 (i.e., `homeschg`) for all quarter sections in Montana.



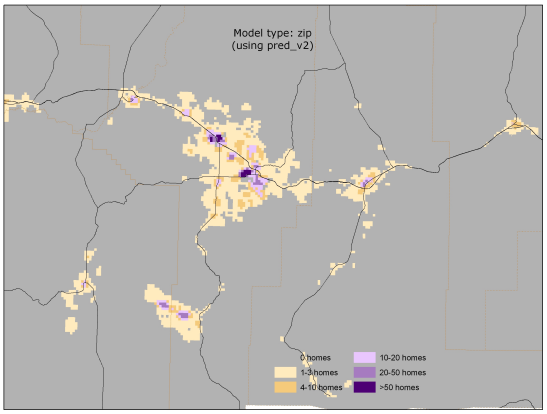
(a) Linear model.



(b) Log-linear model.



(c) GLM.



(d) ZIP model.

Figure 2: Predictions of homes built between 2000 and 2008 by the 4 primary model types in an area around Bozeman, MT.

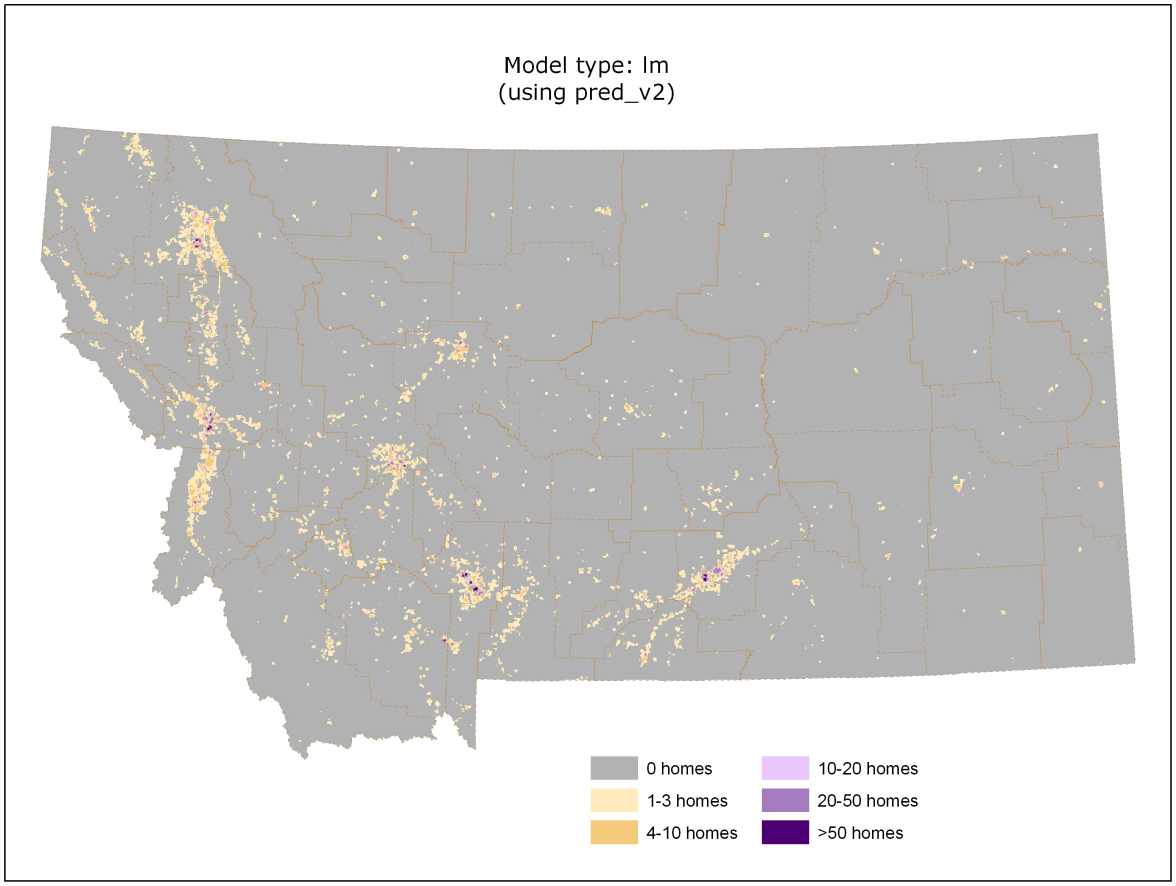


Figure 3: Predicted number of homes to be built between 2008 and 2016 in Montana based on the final linear model.

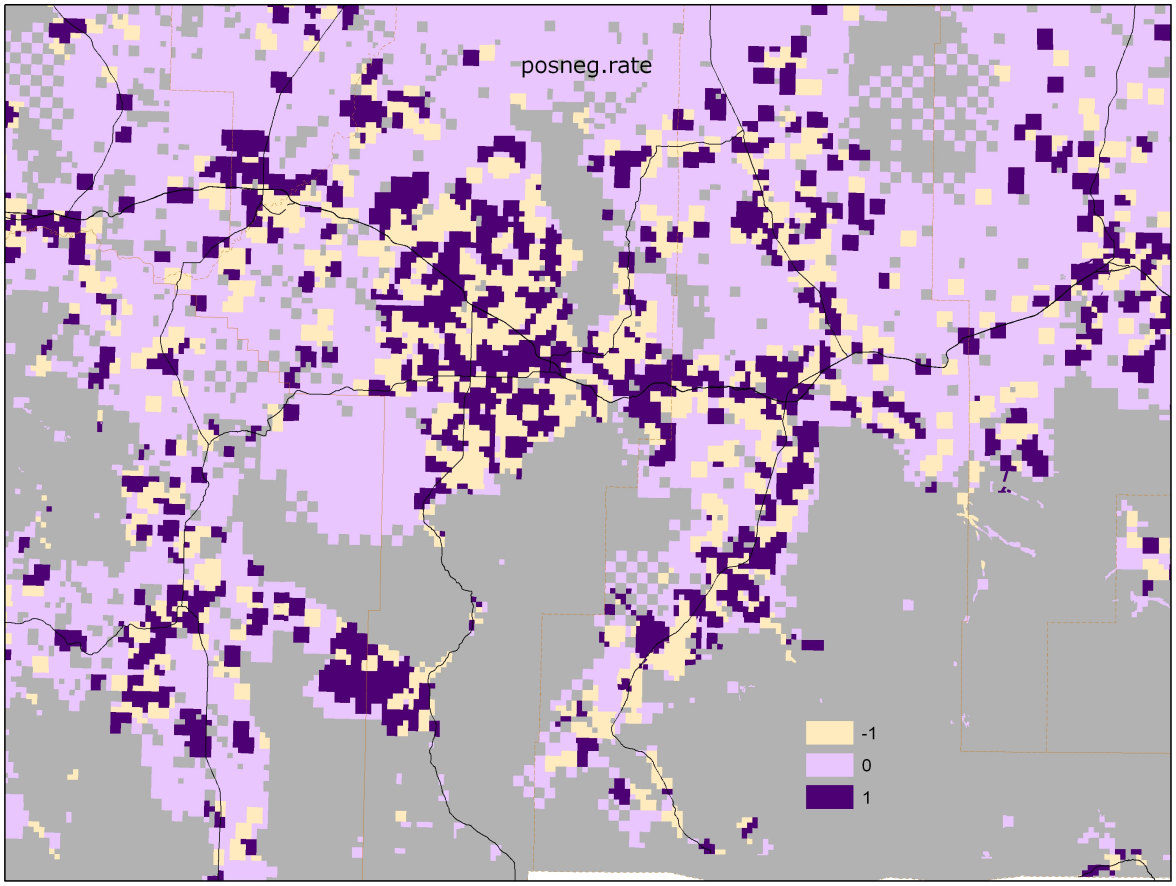


Figure 4: Predicted acceleration (dark purple; 1) or deceleration (tan; -1) in number of homes built between 2008 and 2016 in an area surrounding Bozeman, MT based on the final linear model. Light purple (0) are areas predicted to have steady growth between 2000-2008 and 2008-2016. Many of these (light purple) areas did not experience any growth in 2000-2008 and are not predicted to experience any growth during 2008-2016. Quarter sections that contain no private lands are shown in gray.

Appendix A: R code

```
#
# Read headwaters data and process to ready it for modeling.
# load data
dat.full <- read.csv("C:\\Users\\trent\\Documents\\Projects\\Headwaters
    \\rawdata\\developmentforecast2.csv", header=T)
#   The missing values for homes_1mi and homes8yr1mi should be zeros
#   per email from Patty 4 Aug.
#   Her email:
#   hi trent & darren.  when you rerun model selection, please do include
#   the homes_1mi & homes8yr1mi variables.  i understand you dropped them
#   because 61.4% of the records were null, but these should be zeros.
#   i've investigated & the nulls are a result of the way we calculated
#   these variables in GIS, but they truly are zeros. -patty
dat.full$homes_1mi[ is.na(dat.full$homes_1mi) ] <- 0
dat.full$home8yr1mi[ is.na(dat.full$home8yr1mi) ] <- 0
# ------------------------------------------
# Missing values
# default summary method.  Computes number of missing values
summary(dat.full)
# Note that a majority of Easement is missing.
# also, a few of the water_dist, rd_dens, mjrd_dist, summit_dens, pcinc,
# pop_dens, apt_tt, and town_tt are missing
# Below we see that the few missing covariates are subsetted, and only
# consist of 277 observations,
## 277 missing from the same observations
g1 <- which(is.na(dat.full$pcinc))
g2 <- which(is.na(dat.full$pop_dens))
## 274 missing from the same observations
g3 <- which(is.na(dat.full$mjrd_dist))
g4 <- which(is.na(dat.full$rd_dens))
g5 <- which(is.na(dat.full$ap_tt))
print(sum( g4!=g3 ))  ## = 0
print(sum( g4!=g5 ))  ## = 0
## other missing covariates.  If these are missing, then the above are also
## missing
g6 <- which(is.na(dat.full$water_dist))
g7<- which(is.na(dat.full$town_tt))
g8 <- which(is.na(dat.full$summit_den))
# checking subsets.  This verifies that rows with missing
# values are subset of those in g1 (and g2).
cat("All these numbers should be 0:\n")
print(sum( g2!=g1 ))       ## is 0, means g1 and g2 are same rows
print(sum(!(g3 %in% g1)))
print(sum(!(g4 %in% g1)))
print(sum(!(g5 %in% g1)))
print(sum(!(g6 %in% g1)))
print(sum(!(g7 %in% g1)))
print(sum(!(g8 %in% g1)))
```

```
#   All the above are 0, implies all missing rows are subset of those listed in g2
#-----------------------------------------
# update dataframe by removing Easement and the rows identified in g1
# remove missing rows
dat <- dat.full[-g1,]
# remove Easement,
dat <- dat[,-which( names(dat) %in% c("Easement"))]
#   Clean up. Careful: this erases everything but dat and dat.full
#remove( list= ls()[!(ls() %in% c("dat", "dat.full"))] )
cat("Final dimesion of data frame is:\n")
print(dim(dat))
save.image("Headwaters_raw.RData")
#-----------------------------------------
# put covariates on the same scale for modeling
covars <- c("homes", "homes8yr", "homes_1mi", "home8yr1mi",
    "water_dist", "rd_dens", "mjrd_dist",
    "summit_den", "pcinc", "pop_dens", "ap_tt", "town_tt",
    "x_coord", "y_coord")
desc.stats <- cbind(colMeans(dat[,covars]), apply(dat[,covars],2,sd))
for( j in covars){
    if( j != "ecoregion" ){
        ind <- row.names(desc.stats) == j
        dat[,j] <- (dat[,j] - desc.stats[ind,1]) / desc.stats[ind,2]
    }
}
# make factor covars.
dat$ecoregion <- factor(dat$ecoregion, levels=c(1,2,3),
    labels=c("NW Glacated","NW Plains","N Rockies"))
# compute log homeschg
dat$log.homeschg <- log( dat$homeschg + 1 )
save(dat, desc.stats, file="dat.Rdata" )
```

```
#   Model estimation for Headwater Economics housing development
#   project.
#
#   Trent McDonald - 23Sep10
#
#   Input:
#   dat = the data frame containing all covariates and the response
#
#-----------------------------------------
F.lm.model <- function(fit.dat=dat){
#
#   Fit a normal linear model to dat .
#
ff <- homeschg~homes_1mi + home8yr1mi + water_dist + rd_dens +
        mjrd_dist + ecoregion + summit_den + pcinc + pop_dens +
        ap_tt + town_tt
g.lm <- lm(ff,data=fit.dat)
g.lm <- step( g.lm )    # backward elimination using AIC
cat("Final linear model:\n")
```

```
print( summary(g.lm))
g.lm
}
# ------------------------------------------
F.loglm.model <- function(fit.dat=dat){
#
#   Fit a log normal linear model to dat .
#
fit.dat$log.homeschg <- log( fit.dat$homeschg + 1 )
ff <- log.homeschg~homes_1mi + home8yr1mi + water_dist + rd_dens +
    mjrd_dist + ecoregion + summit_den + pcinc + pop_dens + ap_tt + town_tt
g.lm <- lm(ff,data=fit.dat)
g.lm <- step( g.lm )    # backward elimination using AIC
cat("Final log linear model:\n")
print( summary(g.lm))
g.lm
}
# ------------------------------------------
F.glm.model <- function(fit.dat=dat){
#
#   Fit a generalized (Poisson) linear model to dat .
#
ff <- homeschg~homes_1mi + home8yr1mi + water_dist + rd_dens + mjrd_dist +
    ecoregion + summit_den + pcinc + pop_dens + ap_tt + town_tt
g.lm <- glm(ff,data=fit.dat, family=poisson)
g.lm <- step( g.lm )    # backward elimination using AIC
cat("Final Poisson linear model:\n")
print( summary(g.lm))
g.lm
}
# ------------------------------------------
F.zip.model <- function(fit.dat=dat){
#
#   Fit a Zero inflated Poisson model
#
library(pscl)
g.zip <- zeroinfl( homeschg ~ home8yr1mi + water_dist + rd_dens +
    mjrd_dist + ecoregion + summit_den + pcinc + pop_dens + ap_tt +
    town_tt  | home8yr1mi + town_tt,
    data = dat, dist="poisson", link="logit")
cat("Final Zip model:\n")
print( summary(g.zip))
g.zip
}
# ------------------------------------------
F.score.model <- function(mod, data = dat, y="homeschg", exclude=20,
    cost = function(y, yhat) mean((y - yhat)^2), K = n,
    ilink=function(x){x}){
#
#   score the model using k-fold cross-validation
#
```

```
#   Inputs:
#   mod = a fitted model object
#   data = fitting data frame.
#   y = response variable in data
#   exlude = value of response, above which we exclude when we calculate score
#   cost = function to computed on predictions measuring "closeness" of
#       predicted values to observations
#   K = number of equal sized "folds" to use for validation
#   ilink = the inverse link function
#
#   This code was largely plagerized from library(boot) routine cv.glm
n <- nrow(data)
out <- NULL
if ((K > n) || (K <= 1))
    stop("K outside allowable range")
K.o <- K
K <- round(K)
kvals <- unique(round(n/(1:floor(n/2))))
temp <- abs(kvals - K)
if (!any(temp == 0))
    K <- kvals[temp == min(temp)][1L]
if (K != K.o)
    warning("K has been set to ", K)
cat(paste("K set to", K, "\n"))
f <- ceiling(n/K)
s <- sample(rep(1:K, f), n)   # A permutation of 1:K rep-ed f times
n.s <- table(s)
glm.y <- data[,y]    # response vector
ind <- glm.y <= exclude
ms <- max(s)
CV <- V <- 0
for (i in 1:ms) {
    cat(paste("---- Fold", i, "\n"))
    j.out <- c(1:n)[s == i]
    j.in <- c(1:n)[s != i]
    data.in <- data[j.in, ]
    d.glm <- update( mod, data=data.in )
    ind.i <- glm.y[j.out] <= exclude
    p.alpha <- sum(ind.i)/(sum(ind))
    mu.hat <- predict(d.glm, data[j.out, , drop = FALSE], type = "response")
    mu.hat <- ilink(mu.hat)
    mu.hat[ mu.hat < 0 ] <- 0  # negative predictions possible with identity link.
    cost.i <- cost(glm.y[j.out][ind.i], mu.hat[ind.i])
    sep.i <- var( mu.hat[ind.i] )
    CV <- CV + p.alpha * cost.i
    V <- V + p.alpha * (sum(ind.i) - 1) * sep.i / sum(ind.i)
    cat(paste("\t", c("p.alpha=","CV=","V="), c(p.alpha, cost.i,
        sep.i*(sum(ind.i)-1)/sum(ind.i)), "\n"))
}
out <- list(K = K, cv = CV, v=V, score=CV + V)
cat("\nScore of final model using cross-validation:\n")
```

```
cat(paste("\tMean square cross-validation prediction error   =", CV, "\n"))
cat(paste("\t      Variance of cross-valication predictions =", V, "\n"))
cat(paste("\t                                        Score (CV + V) =", CV + V, "\n"))
out
}
# ------------------------------------
#   A utility function to show size of objects.  The model objects are
#   HUGE, and its good to know their size
F.object.size<-function(){
rev(sort(sapply( ls(envir=.GlobalEnv), function(x)
    {object.size(get(x, envir=.GlobalEnv))})))
}
# ------------------------------------
F.cut.size <- function(x){
#
#   cut down the size of a modeling object by erasing some components
#   of the model.
#
if( "lm" %in% class(x) ){
    x$residuals <- NULL
    x$fitted.values <- NULL
    x$effects <- NULL
    x$model <- NULL
}
if( "glm" %in% class(x) ){
    x$linear.predictors <- NULL
    x$weights <- NULL
    x$prior.weights <- NULL
    x$data <- NULL
    x$y <- NULL
}
if( "zeroinfl" %in% class(x) ){
    x$residuals <- NULL
    x$fitted.values <- NULL
    x$y <- NULL
}
x
}
# ------------------------------------
#   Call the above functions to do estimation
#
#   The linear model
g.lm <- F.lm.model()
g.lm.score <- F.score.model(g.lm, dat, y="homeschg", exclude = 20,
    K=10, ilink=function(x){x})
g.lm <- F.cut.size(g.lm)

#   The log linear model
g.loglm <- F.loglm.model()
g.loglm.score <- F.score.model(g.loglm, dat, y="homeschg", exclude = 20,
    K=10, ilink=function(x){exp(x)} )
```

```
g.loglm <- F.cut.size(g.loglm)

#   The Poisson model
g.glm <- F.glm.model()
g.glm.score <- F.score.model(g.glm, dat, y="homeschg", exclude = 20,
    K=10, ilink=function(x){x} )
g.glm <- F.cut.size(g.glm)

#   The ZIP model
g.zip <- F.zip.model()
g.zip.score <- F.score.model(g.zip, dat, y="homeschg", exclude = 20,
    K=10, ilink=function(x){x} )
g.zip <- F.cut.size(g.zip)

#   Results
ans <- data.frame( model=c("Normal","Log","Glm","Zip"),
    score=c(g.lm.score$score, g.loglm.score$score,
        g.glm.score$score, g.zip.score$score))
print(ans)

save.image( file="Models.RData" )
```

```
#
# Read new (updated) headwaters data prep it for prediction
#
# load data
new.dat <- read.csv("C:\\Users\\trent\\Documents\\Projects\\
    Headwaters\\rawdata\\Homes2008data.csv", header=T)
new.dat$homes_1mi[ is.na(new.dat$homes_1mi) ] <- 0
new.dat$home8yr1mi[ is.na(new.dat$home8yr1mi) ] <- 0
cat(paste("Original number of rows in 'new.dat':", nrow(new.dat), "\n"))
cat(paste("Original number of rows in 'dat.full':", nrow(dat.full), "\n"))
#   Merge new home counts with old values in 'dat'
#   First, drop the columns from 'dat' that will be replaced by data in new.dat
dat.full <- dat.full[, !(names(dat.full) %in% c("homes_1mi", "home8yr1mi"))]
new.dat <- merge(new.dat, dat.full, by="PLSS", all=T)
#   Checking...
cat(paste("New number of rows in 'new.dat':", nrow(new.dat), "\n"))
cat(paste("Number of rows in 'new.dat', not in 'dat.full':",
    sum(is.na(new.dat$homeschg)), "\n"))
cat(paste("Number of rows in 'dat.full', not in 'new.dat':",
    sum(is.na(new.dat$home8yr1mi)), "\n"))
#   Scale the variables as we did for fitting.
#   Scaling means and std's must be in desc.stats.
for( j in covars){
    if( j != "ecoregion" ){
        ind <- row.names(desc.stats) == j
        new.dat[,j] <- (new.dat[,j] - desc.stats[ind,1]) / desc.stats[ind,2]
    }
}
## make factor covars.
```

```
new.dat$ecoregion <- factor(new.dat$ecoregion, levels=c(1,2,3),
    labels=c("NW Glacated","NW Plains","N Rockies"))
save( new.dat, file="new_dat.RData")
```

```
#
#   R Code used to predict future growth, compute lower prediction limits,
#   and accelerations
#
#   Trent McDonald, 23Sep10
#
F.predict.model <- function(mod, new.data, ilink=function(x){x}){
#
#   Predict model responses
#
mu.hat <- predict(mod, new.data, type = "response")
mu.hat <- ilink(mu.hat)
mu.hat[ mu.hat < 0 ] <- 0
as.data.frame(list(PLSS = new.data$PLSS, predicted = mu.hat))
}
#-----------------------------------------
F.output <-function( df, file ){
#
#   Output the predicted values for plotting
write.table( df, file, sep=",", col.names=T, row.names=F )
}
#-----------------------------------------
#   Because we erased some stuff from the models, to cut down size, we
#   must re-estimate models
#
#   dat and dat.new have different dimensions because 277 rows with
#   missing values were deleted from dat.  See read_data.r
#   This makes predictions using dat and new.dat different sizes
#load( "dat.Rdata" )
#load( "new_dat.Rdata" )
library(pscl)
tmp <- g.lm
g.lm <- update( g.lm, data= dat )
pred <- F.predict.model( g.lm, dat )
#   -- These are predictions of past growth
F.output( pred, "Pred_lm_v2.csv" )
#   -- Future predicted values and other things for the LM model
#   Use 0.9 for confidence in next statemetn to get 0.95 one-sided intervals
pred1 <- as.data.frame( predict(g.lm, new.dat,
    interval="prediction", level=0.9, type = "response"))
pred1 <- pred1[,c("fit", "lwr")]  # drop upper limit
pred1$fit[ pred1$fit < 0 ] <- 0
pred1$lwr[ pred1$lwr < 0 ] <- 0
pred1 <- data.frame(PLSS = new.dat$PLSS, pred1, stringsAsFactors=F)
#   -- Merge so we can compute differences
names(pred)[ names(pred) == "predicted" ] <- "past.predicted"
names(pred1)[ names(pred1) == "fit" ] <- "future.predicted"
```

```
names(pred1)[ names(pred1) == "lwr" ] <- "future.low95"
pred1$sure.growth <- as.numeric( pred1$future.low95 > 0 )
tmp <- merge( pred, pred1, by="PLSS" )
tmp$growth.rate <- tmp$future.predicted - tmp$past.predicted
tmp$growth.rate[ abs(tmp$growth.rate) < 0.00001 ] <- 0 # Zeros out numbers like 1e-16.
tmp$posneg.rate <- tmp$growth.rate
tmp$posneg.rate[ tmp$growth.rate > 0.01 ] <- 1
tmp$posneg.rate[ -0.01 <= tmp$growth.rate & tmp$growth.rate <= 0.01 ] <- 0
tmp$posneg.rate[ tmp$growth.rate < -0.01 ] <- -1
#   -- Merge in observed values
tmp2 <- dat[, c("PLSS", "homeschg")]
tmp <- merge( tmp, tmp2, by="PLSS" )
F.output( tmp, "PastFuture_lm_v1.csv" )
# ----------------------
tmp <- g.loglm
g.loglm <- update( g.loglm, data= dat )
print(rbind( g.loglm$coef , tmp$coef ))
pred <- F.predict.model( g.loglm, dat, function(x){exp(x) - 1} )
F.output( pred, "Pred_loglm_v2.csv" )
# ----------------------
tmp <- g.glm
g.glm <- update( g.glm, data= dat )
print(rbind( g.glm$coef , tmp$coef ))
pred <- F.predict.model( g.glm, dat )
F.output( pred, "Pred_glm_v2.csv" )
# ----------------------
tmp <- g.zip
g.zip <- update( g.zip, data= dat )
print(rbind( g.zip$coef , tmp$coef ))
# ----------------------
pred <- F.predict.model( g.zip, dat )
F.output( pred, "Pred_zip_v2.csv" )
```