BDA Project: Predicting Default Throughout Mexico

Andres Potapczynski (ap3635), Jongwoo Choi (jc4816), Yi Chen (yc3356)

12/10/2018

Abstract

We help an small mortgage lending start-up in Mexico understand how several demographical and economic variables predict default in different cities throughout the country. We find that XXX. For this we tested over XXX models which showed us that XXX.

Note: All the code and STAN models can be found in the following link [...][Add GitHub link]

The results of our analysis is summarized in the table below.

[...][Finish Summary Table and Verify the Numbers]

Setting the preamble

In this project we are helping a small mortgage lending start-up in Mexico understand how different cities differ in their risk to default on their mortgage. This start-up has recently entered the market and wants us to help it asses how it should expand geographically throughout the country. Put differently, they want us to help them prioritize the expansion to cities that have shown the highest compliance as well as to understand how different variables predict such compliance.

In order to accomplish this, **we were provided with a large data set of 30,499 mortgages with over 90 covariates** (for a comprehensive explanation of the data set see the next section). Also, **we interviewed the startup to understand their main hypothesis** of why different cities might have a diverging default behavior as well as to retrieve all the domain knowledge possible for both our modeling and our prior assumptions. Our main findings were:

- **Some northern states in Mexico are fighting a drug war against different cartels and some cities have been specially damaged**. Thus, even though the northern region of Mexico is one of the wealthiest, cities belonging to the states of Sinaloa, Coahuila and Nuevo Leon might present higher default rates that cannot be explained with the covariates that we were given (since non are related to this event).
- **Poor southern states have a cultural tendency to fall easier into default**. States like Oaxaca might present a higher default rate even among high income individuals. The rational is that, given the low financial penetration in those states, individuals are less concern about their credit score and thus are more prone to abandon their properties if they fall into financial distress or even when they dislike their property.
- **The covariates related to employment should be the most predictive variables**. The start-up believes that the main driver of default is that people lose their jobs. Thus, the employment variables of our data set should be the most relevant.

Understanding the data

In this section we examine three main elements of the data set that we were given. First, we expose the set of covariates that we were given. Next, we show the most important plots from of exploratory data analysis. Finally, we detail the preprocessing steps that we took before pushing the data into `STAN`.

Covariates

As mentioned in the introduction, the data set consists of **30,499 mortgages with over 90 covariates** (where some of features are simply administrative and thus were not included in the analysis). **The average default rate is 6%**. Also, the data set provided was **their latest report available at August 2018**. We group the covariates in the following categories:

- [**Location**] `state`, `city` and `zip`: These features will allow me to understand if the client's behavior varies by geography. Also, I have this location features for both the house acquired and for the owner's location.
- [**Demographics**] `age`, `sex`, `ratio`, `risk_index`, `client_income` and `credit_score`. The feature `ratio` is the % of the client's income that the monthly payments for its mortgage represent. The `risk_index` is a variable that combines several metrics associated with his likelihood of paying a debt. The `client_income` feature is the monthly income that the person earned at that moment in time when she signed the mortgage. Finally, the ordinal variable `credit_score` evaluates how the client has performed in previous debts (related to `risk_index`).
- [**Asset features**] `vendor name`, `new_used`, and `appraisal_value`: These features tell us who was the vendor (either a construction company or an individual) and if the asset is new or not as well as the amount of money the person was given as a mortgage.
- [**Employment**] `employer_name` and `factor_employed`. The first feature conveys who is the employer and the second one information about the client's status of employment.
- [**Payment Records**] `days_pay`, and `y`. The first feature counts the number of days that have passed between the last payment date and the date on which the mortgage started. The next feature is the target variable that is 1 if the mortgage has at least one month without a payment and 0 otherwise.

Note that features like `interest_rate` and `contract_length` are present but have no variance. The current *product offer* in the data base is a 12% interest rate mortgage for 30 years. Thus it is impossible to pose counter factual questions for those variables.

Exploratory Data Analysis

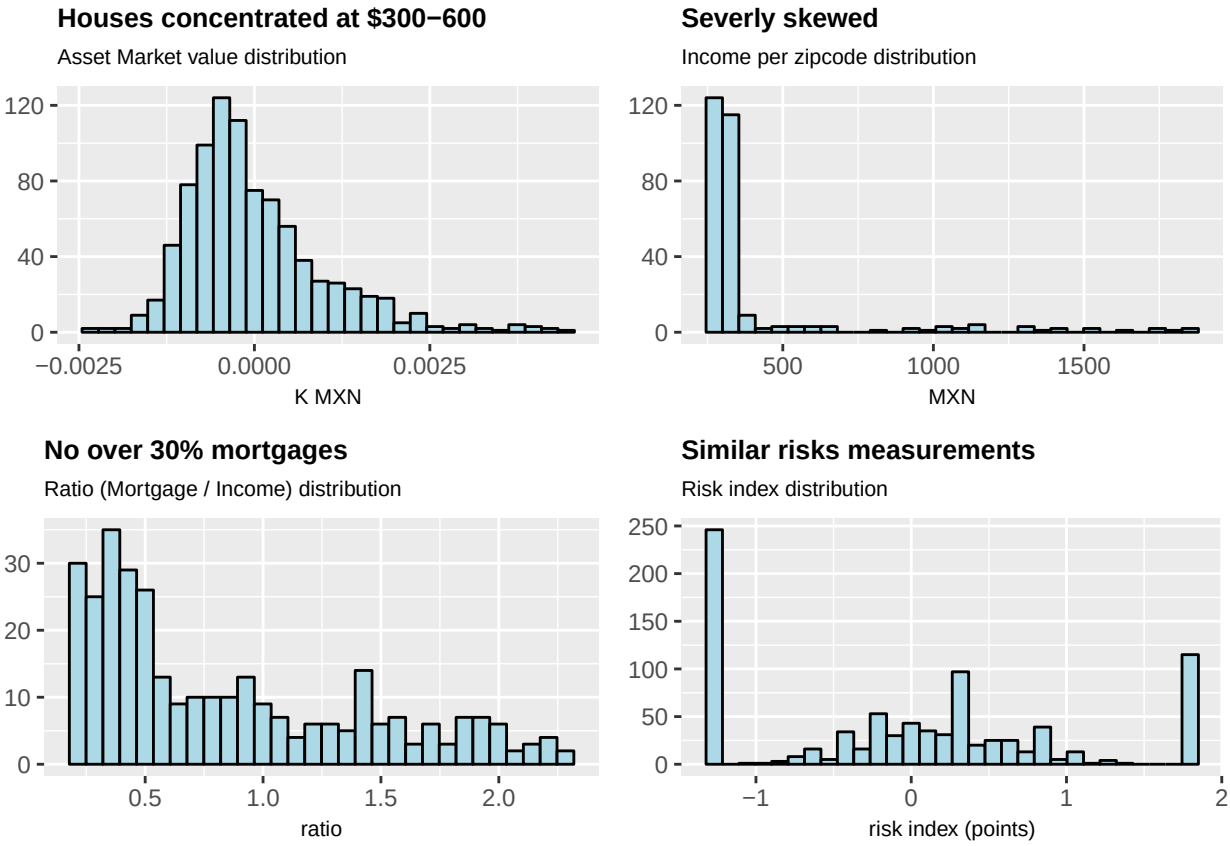
Our main tool for this analysis was to use some Machine Learning techniques as well as to plot the data in creative ways. The ML techniques that we used fall into two categories: (1) classification and (2) clustering. The data is highly unbalanced, thus predicting always that the person is not going to default ($Y = 0$) already achieves a 93% test accuracy. Thus, in terms of (1) we ran KNN and Decision Trees (both nonparametric classifier) in order to understand if the data that we were given contained the variables (and possible the interactions) which could increase the accuracy of default prediction. In terms of (2), we ran k-means which is a clustering technique that enable us to understand the different profile of persons in the data; these profile are detailed below as well as the graphs that we came up with.

The results of the first exercise are the following.

Classifier	5 fold Test Accuracy
KNN	94 %
Decision Tree	93 %
Always Predict Zero	93 %

Were we see that it appears futile to perform our analysis at the individual level since the variables that we were given do not separate the data. However, as our baseline model, we do try this alternative.

The plots of the variables that we included in the analysis are:



Data Preprocessing

Below we list the main preprocessing steps that we did with the data

- **Aggregated the variables at the city level**. The data set that we were given was at the individual mortgage level. In order to make it suitable for the geographical analysis we aggregated the relevant variables at the city level. The majority by the mean but some by their sum. For example, individual income and age were transformed into mean income and mean age in that city were as the binary response when people default was summed into the number of people that defaulted in that city.
- **Transform to log space the variables related to money**. As it is well-known, income distributions tend to be skewed and to resemble extreme value densities. Thus, we transformed those values into the log space to bring closer together the large values that were present.
- **Z-score all the variables**. Finally we, z-scored all the variables to put them in the same scale and thus assuring that placing a similar prior in their slope coefficients (β_j) made sense.

Trying out Logistic Regression

What appears most natural is to model this data with a logistic regression since we are trying to estimate a binary variable with information from several variables. However, as it was discussed previously, we find that the data is not informative at this level. Even with nonparametric methods it is not possible to uncover a clear relationship. However, we ran the model and show below where it falls short.

Specification

The generative process is the following:

$$\alpha \sim N(0, 5)$$

$$\beta \sim N(0, 5)$$

$$y_i \sim Ber(logit^{-1}(\alpha + X_i\beta))$$

the variables that are going to be included for the X are the following: `client_income`: β_1, `appraisal_value`: β_2, `app_2_inc`: β_3, `mar_2_app`: β_4, `sex_F`: β_5, `age`: β_6, `risk_index`: β_7, `employed_30`: β_8 and `condition_U`: β_9.

Results

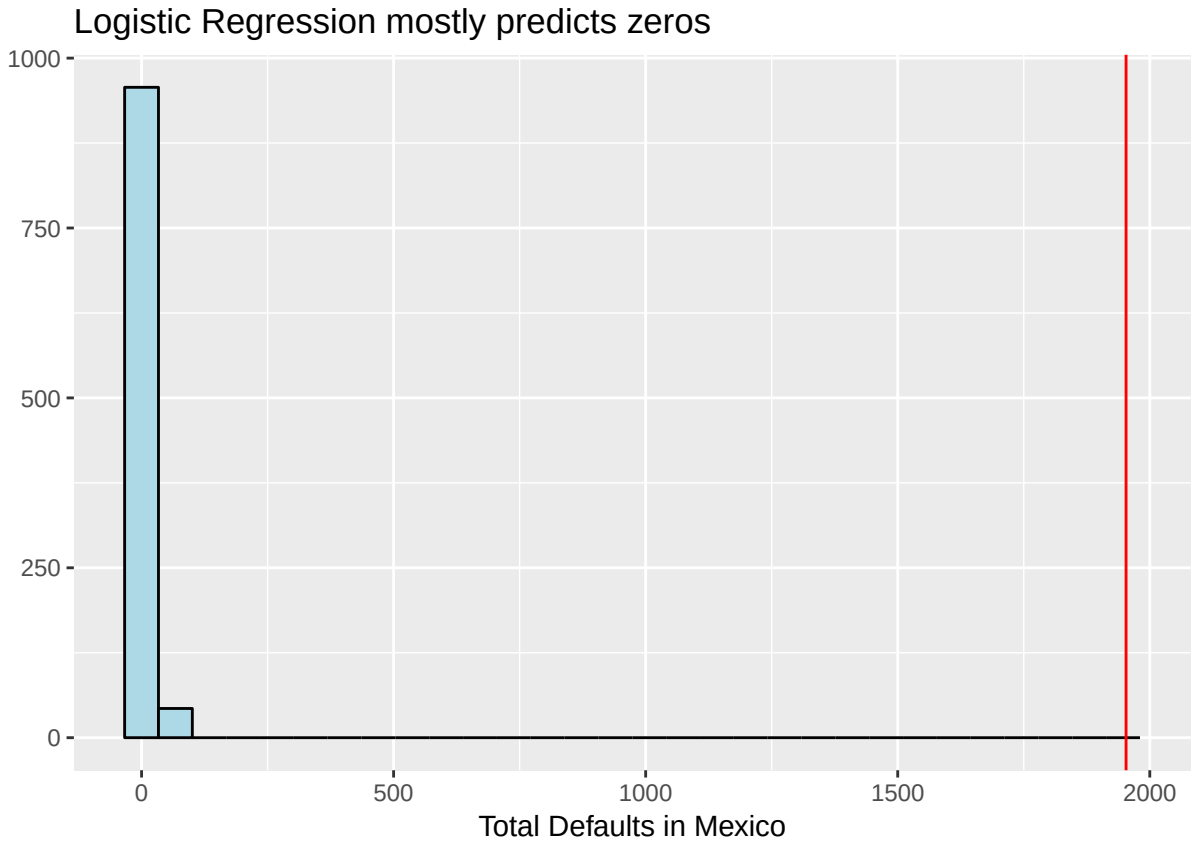
The results from fitting that previous model in STAN is

```
## Inference for Stan model: logistic.
## 1 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=1000.
##
##          mean se_mean   sd   2.5%   50% 97.5% n_eff Rhat
## alpha    3.30    0.19 4.76  -5.78  3.27 12.96   616 1.00
## beta[1]  0.00    0.00 0.00  -0.01  0.00  0.00   785 1.00
## beta[2] -5.11    0.19 4.60 -13.83 -5.34  3.92   604 1.00
## beta[3]  1.03    0.02 0.48   0.09  1.01  2.01   734 1.01
## beta[4]  0.00    0.00 0.03  -0.07  0.00  0.05   752 1.00
## beta[5]  0.06    0.02 0.53  -1.00  0.08  1.08   804 1.00
## beta[6] -1.00    0.04 0.76  -2.87 -0.85 -0.01   316 1.00
## beta[7]  0.00    0.00 0.00  -0.01  0.00  0.00   488 1.00
## beta[8] -0.10    0.00 0.03  -0.16 -0.10 -0.05   734 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Dec  7 22:28:28 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

As it can be seen above, there is a lot of uncertainty for the β's. This will remain a constant theme throughout this analysis, however, in this case it is particularly true that all the coefficients include zero and that the intercept α takes the leading role. Now, without further discussing the model, let's see how it is not suitable for this analysis.

Evaluation

The total number of people that default is a critical quantity to estimate for the start-up since there is a direct link to its profits. The following PPC shows how the logistic regression model has an unacceptable prediction for this quantity of interest.



as it can be seen above the value observe is completely off what the model predicts. Thus we now change our approach to the problem.

Changing the approach

We changed our approach in the following manner. Rather than keep modelling the individual data, we aggregated it at the level on which the start-up was interested: city and state. The aggregation was mostly done by averaging the different variables for each individual. In this way, since we are now working with variables which are the combination of many small events (the individuals), the CLT then provides us with a more manageable data set. Now part of the noise was pruned on the aggregation process.

With this new approach, we now focus on modelling a count variable: the number of defaults per city. For this type of data, we tried different models such as binomial, poisson and negative binomial. All of them yielded similar results: both for the values of β and for held-out RMSE in 5 fold CV. We opted to continue the analysis with the binomial specification since it was converging fast and with not numerical problems (in contrast with the poisson model when even when using the function `poisson_log` in `STAN` we had numerical problems for some chains).

Specification

The generative process for our main model of this section is

$$\alpha \sim N(0, 5)$$
$$\beta \sim N(0, 5)$$

$$y_j \sim Bin(n_j, logit^{-1}(\alpha + X_j\beta))$$

where now $j = 1, \ldots, 880$ (total number of cities in the data set) and X_j is equal to the average of all the value of that variables for all the individuals in that city. The other model specifications that we tried only changed how y_j is distributed. For the poisson model we have

$$y_j \sim Poi(n_j \exp(\alpha + X_j\beta))$$

whereas for the negative binomial we have to add another positive parameter ϕ. We employed the mean-variance parameterization and thus

$$y_j \sim NegBin(\exp(\alpha + X_j\beta), \phi)$$

Finally, as part of a prior check, we also substitute the normal priors for

$$\alpha \sim Cauchy(0, 5)$$

$$\beta \sim Cauchy(0, 5)$$

Parameter Recovery

[. . .][Add how the model recovers the parameters]

Results

We now present the results for the binomial model (as a `STAN` print) but also in a comparative table the result for the rest of the models.

```
## Inference for Stan model: binomial.
## 8 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=8000.
##
##          mean se_mean   sd  2.5%   50% 97.5% n_eff Rhat
## alpha   -2.73    0.00 0.04 -2.82 -2.73 -2.65  5199    1
## beta[1] -0.27    0.01 0.28 -0.83 -0.27  0.27  3108    1
## beta[2]  0.36    0.00 0.25 -0.12  0.36  0.87  3354    1
## beta[3] -0.20    0.00 0.15 -0.50 -0.20  0.10  3434    1
## beta[4] -0.10    0.00 0.07 -0.25 -0.10  0.04  6036    1
## beta[5]  0.04    0.00 0.08 -0.11  0.04  0.19  7279    1
## beta[6]  0.09    0.00 0.09 -0.09  0.09  0.27  5676    1
## beta[7] -0.19    0.00 0.07 -0.32 -0.19 -0.06  6747    1
## beta[8]  0.25    0.00 0.08  0.08  0.24  0.41  5838    1
## beta[9]  0.11    0.00 0.04  0.02  0.11  0.20  6767    1
##
## Samples were drawn using NUTS(diag_e) at Fri Dec  7 20:41:54 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Coeff	Binomial	Binomial Cauchy	Poisson	Negative Binomial
α	-2.73	-2.73	-2.8	-2.69
β_1	-0.27	-0.20	-0.23	-0.32
β_2	0.36	0.31	0.33	0.39
β_3	-0.2	-0.17	-0.17	-0.25
β_4	-0.10	-0.10	-0.09	-0.11

β_5 | 0.04 | 0.04 | 0.03 | 0.04 β_6 | 0.09 | 0.09 | 0.08 | 0.1 β_7 | -0.19 | -0.19 | -0.17 | -0.2 β_8 | 0.25 | 0.24 | 0.23 | 0.23 β_9 | 0.11 | 0.11 | 0.1 | 0.13

What we can conclude from the previous table is that our estimates are not perturbed by different model specifications. Moreover, the effect of adding a "robust" prior to the problem has little effect.

From the following plot we see that we have a lot of uncertainty in our estimates. [. . .][Add that coefficients plot]

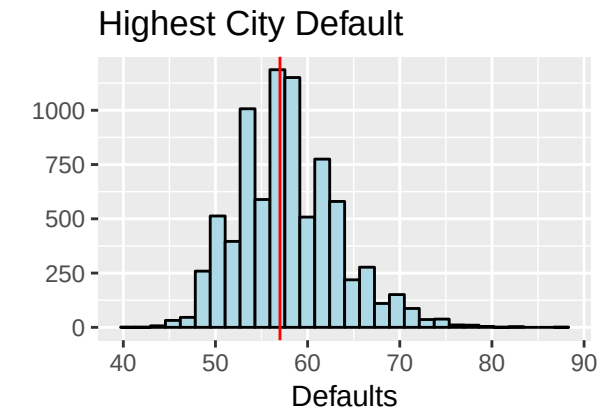
Evaluation

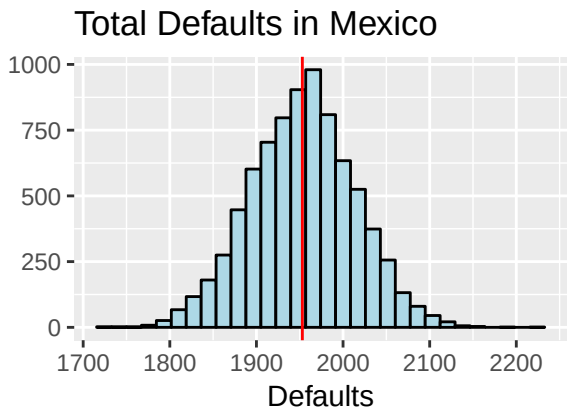
Before jumping into the PPCs of the binomial model, we will present yet another comparative table between the models. Below is the RMSE for 5 fold CV with an 80 / 20 split each time.

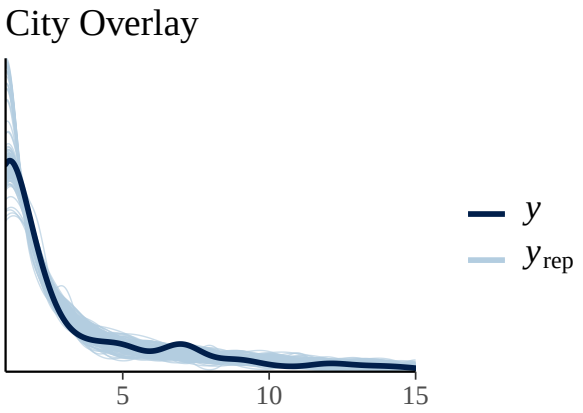
Model	5 fold RMSE (sd RMSE) at City
Predict always zero	7.174 (1.58)
Binomial	1.87 (0.239)
Binomial Cauchy	1.866 (0.24)
Poisson	1.87 (0.242)
Negative Binomial	1.8513 (0.2383)

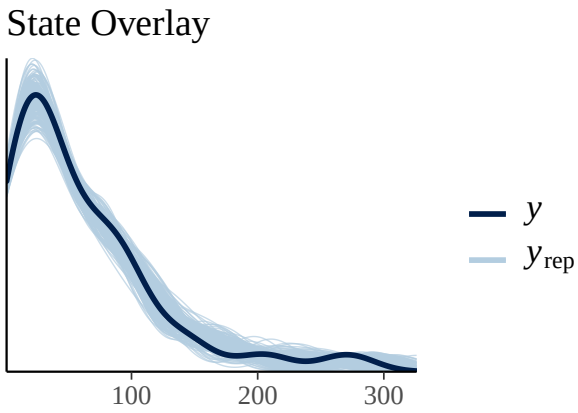
Model	5 fold RMSE (sd RMSE) at State
Predict always zero	23.63 (4.45)
Binomial	4.69 (0.981)
Binomial Cauchy	4.69 (0.977)
Poisson	4.69 (0.9814)
Negative Binomial	4.657 (0.9742)

Again, we start by looking at the tail value of the sum of the total defaults predicted by the model. Now we check how well the model is able to replicate the per city counts.



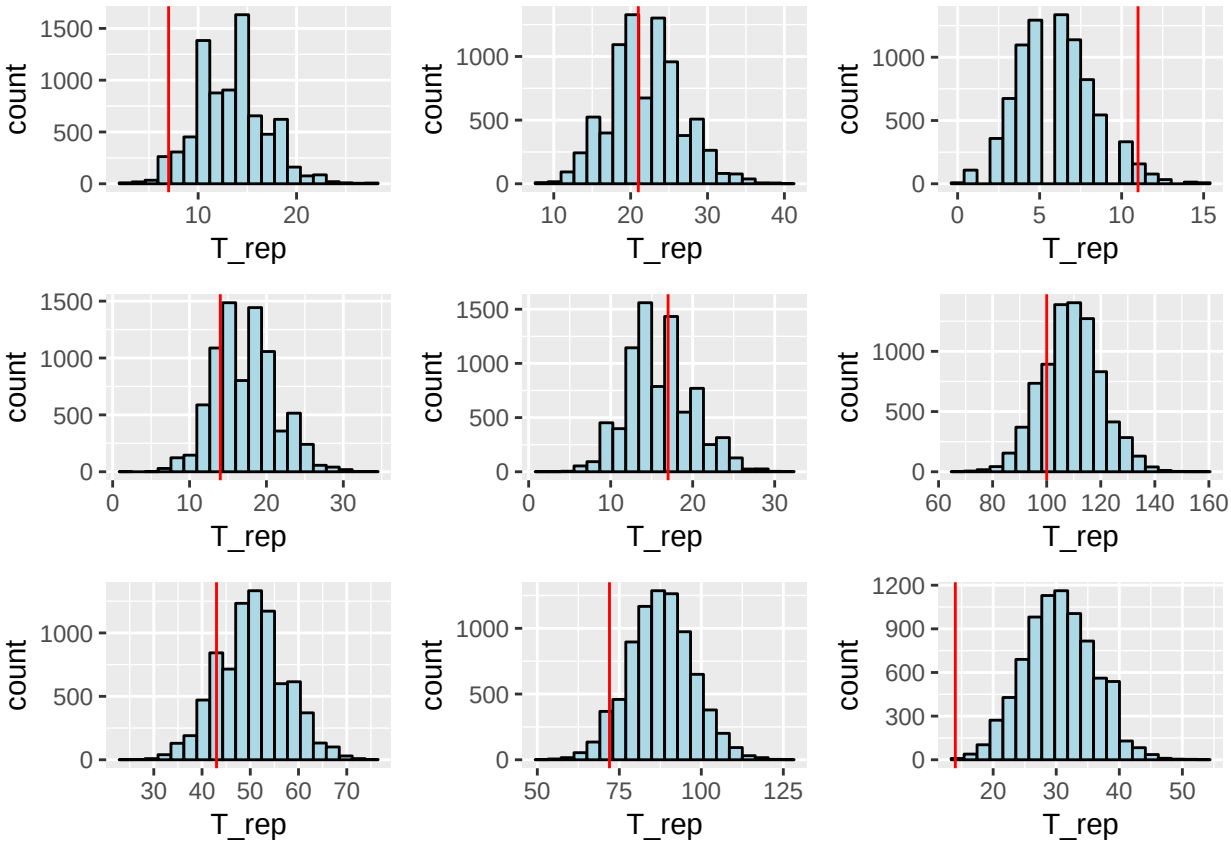






We observe that the model is able to satisfactorily mimic the distribution of the data. In addition, the model is also able to emulate the distribution of the data if we aggregate it at the state level.

Thus far everything is looking great. However, if we look at the per individual state graphs. We acknowledge some deficiency which leave some room for improvement. [. . .][Add multiple plots here]



Including some Hierarchy

The easiest way to fix a PPC is to model it directly. It appears as cheating. In the sense that we keep expanding the model to every eventuality that we encounter; nonetheless, if this eventuality is actually an aspect of the model that we do care then there is not evident harm on expanding the model to fit this aspect as well. This is what we do for the first model expansion. We introduce a hierarchical structure at the state level to each of the intercepts.

Specification

The generative process now is the following

$$\alpha \sim N(0, 5)$$

then for each $s = 1, \ldots, 32$

$$\sigma_s \sim lognormal(\log(1), 1)$$

$$\alpha_s \sim N(\alpha, \sigma_j)$$

$$\beta \sim N(0, 5)$$

$$y_j \sim Bin(n_j, logit^{-1}(\alpha_{s_j} + X_j\beta))$$

where again $j = 1, \ldots, 880$. Now we have an offset that should give us sufficient flexibility to estimate what we want to.

Parameter Recovery

[. . .][See if this is actually needed]

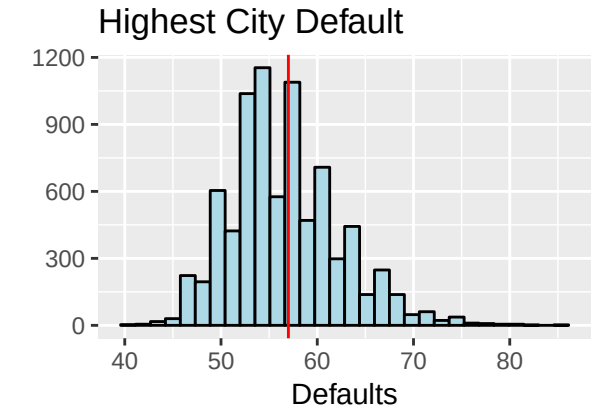
Results

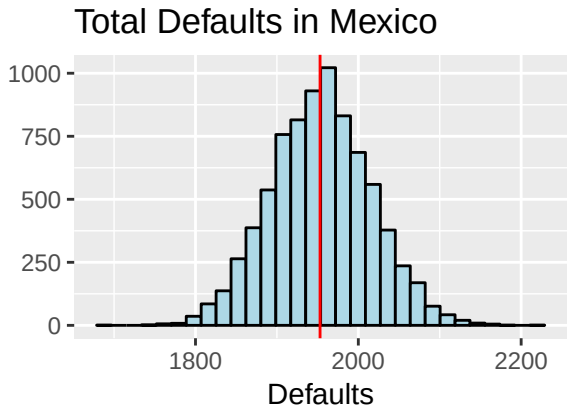
The results of incorporating this hierarchical structure can be seen below.

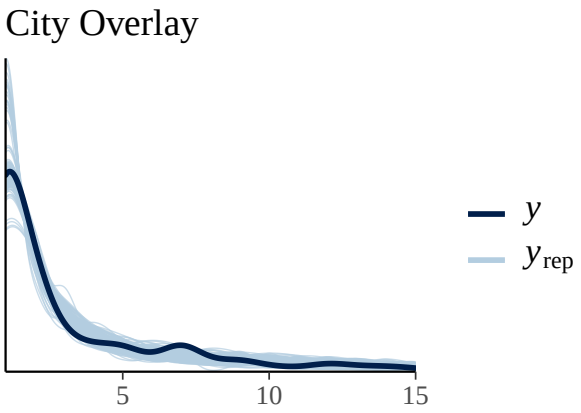
```
## Inference for Stan model: hier_binomial.
## 8 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=8000.
##
##          mean se_mean   sd  2.5%   50% 97.5% n_eff Rhat
## alpha   -2.73    0.01 0.15 -3.03 -2.73 -2.43   755 1.01
## beta[1] -0.09    0.00 0.31 -0.68 -0.09  0.52  4262 1.00
## beta[2]  0.10    0.00 0.28 -0.45  0.10  0.63  4322 1.00
## beta[3] -0.16    0.00 0.17 -0.48 -0.16  0.18  4504 1.00
## beta[4] -0.08    0.00 0.08 -0.25 -0.08  0.08  6589 1.00
## beta[5]  0.02    0.00 0.08 -0.15  0.02  0.17  8828 1.00
## beta[6] -0.12    0.00 0.11 -0.34 -0.13  0.09  5563 1.00
## beta[7] -0.09    0.00 0.08 -0.24 -0.09  0.06  7248 1.00
## beta[8]  0.22    0.00 0.09  0.05  0.22  0.39  6508 1.00
## beta[9]  0.10    0.00 0.06 -0.03  0.10  0.22  7791 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Dec  7 19:34:09 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

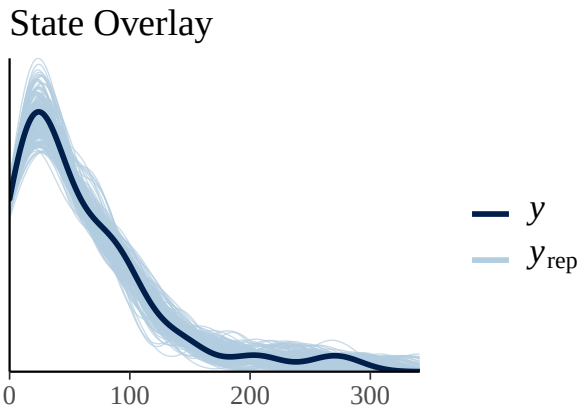
Evaluation

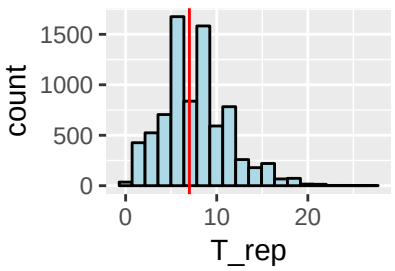
As always, we start with the PPC for the total number of defaults

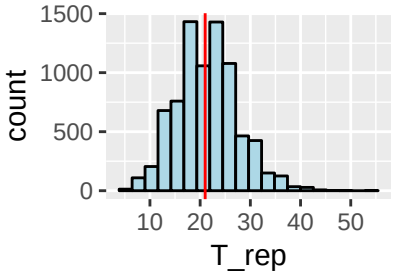

Highest City Default
1200
900
600
300
0
40
50
60
70
80
Defaults

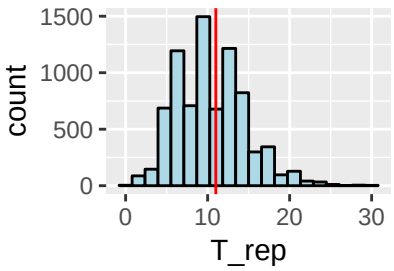

Total Defaults in Mexico
1000
750
500
250
0
1800
2000
2200
Defaults

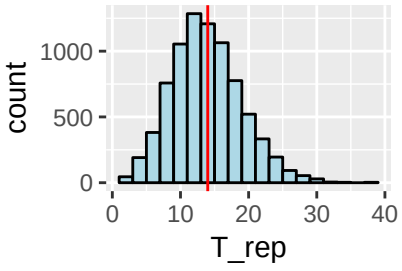

City Overlay
5
10
15
y
y_rep

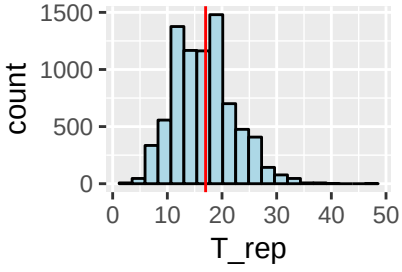

State Overlay
0
100
200
300
y
y_rep

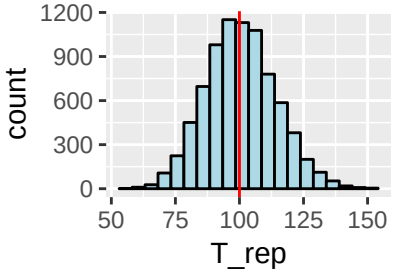

count
1500
1000
500
0
0
10
20
T_rep

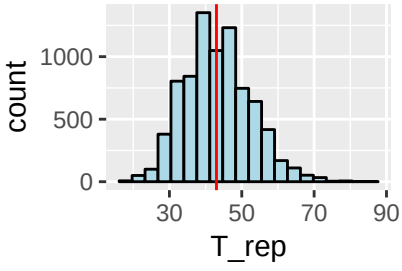

count
1500
1000
500
0
10
20
30
40
50
T_rep

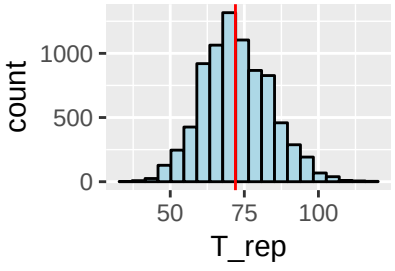

count
1500
1000
500
0
0
10
20
30
T_rep

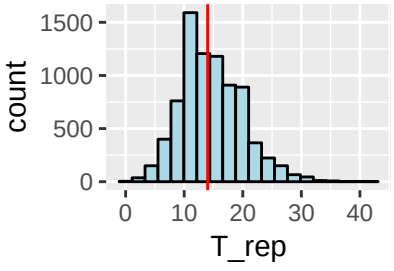

count
1000
500
0
0
10
20
30
40
T_rep


count
1500
1000
500
0
0
10
20
30
40
50
T_rep


count
1200
900
600
300
0
50
75
100
125
150
T_rep


count
1000
500
0
30
50
70
90
T_rep


count
1000
500
0
50
75
100
T_rep


count
1500
1000
500
0
0
10
20
30
40
T_rep

Now it is worth pointing out that in terms of 5 fold CV this extension worsens the performance. The 5 fold RMSE is: 2.1594 (0.61). Even though held out scores are a common currency to compare models, it is not necessarily true that we should always pick the one with the lowest error especially when the differences are not as substantial. Nonetheless, our next model fixes this loss of predictibility and estimates the state default means as well as the model of this section.

Leveraging from Spatial information

In this section we yet again expand the previous model. Now by incorporating spatial information. As seen before, the addition of a hierarchical structure into our model made the y^{rep} at the state level fit much better. Nonetheless we suffered a decrease in held-out accuracy. Thus we explore if adding spatial information is able to alleviate that. The idea is to upgrade the hierarchical interaction of the α_s's where now they share information between neighboring states.

Specification

One of the most popular ways to incorporate spatial random effects it to use Conditional Auto regressive (CAR) priors. Now instead of α_s we are going to switch to ϕ_s. Now the process is the following:

$$\phi_s \mid \phi_j \sim N(\alpha \sum_{j=1}^{n} b_{sj}\phi_j, \tau_s^{-1})$$

where each b_{sj} is an element that encodes the adjacency matrix information and τ_s is a spatially varying precision. By using Brook's lemma we can express simplify the previous expression as:

$$\phi \sim N(0, [D_\tau(I - \alpha B)]^{-1})$$

where

- $D = diag(m_s)$ is a 32×32 diagonal matrix where m_s is the number of the neighbors for the state s
- $D_\tau = \tau D$ and τ is the variance hyperparameter for the ϕs
- α is the parameter that controls spatial dependence.
- $B = D^{-1}W$ is the scaled adjacency matrix (discussed above). And W is the adjacency matrix. ($w_{ss} = 0$, $w_{ij} = 1$ if the state s is a neighbor of state j and zero otherwise)

However, to alleviate the computational burden, it is common to use the IC AR prior where, in the previous expression α is set to 1. The only problem with the ICAR prior is that it is improper.

The generative process is

$$\alpha \sim Cauchy(0, 2.5)$$
$$\beta \sim Cauchy(0, 5)$$
$$\tau \sim Gamma(2, 2)$$

then for each $s = 1, \ldots, 32$

$$\phi_s \sim N(0, [D_\tau(I - \alpha B)]^{-1})$$

and finally

$$y_j \sim Bin(n_j, logit^{-1}(\alpha + \phi_{j_s} + X_j\beta))$$

where again $j = 1, \ldots, 880$.

Before moving forward, it is important to make two comments:

- This prior on the ϕ_s is more restrictive than the previous prior on α_s. Thus, we expect more regularization.
- This design of the model is better for interpretation and for the questions that the start-up wants to answer (since they have geographical concerns). With this model we can *learn* about geographical patterns.

Parameter Recovery

[...][Show that it recovers the parameters]

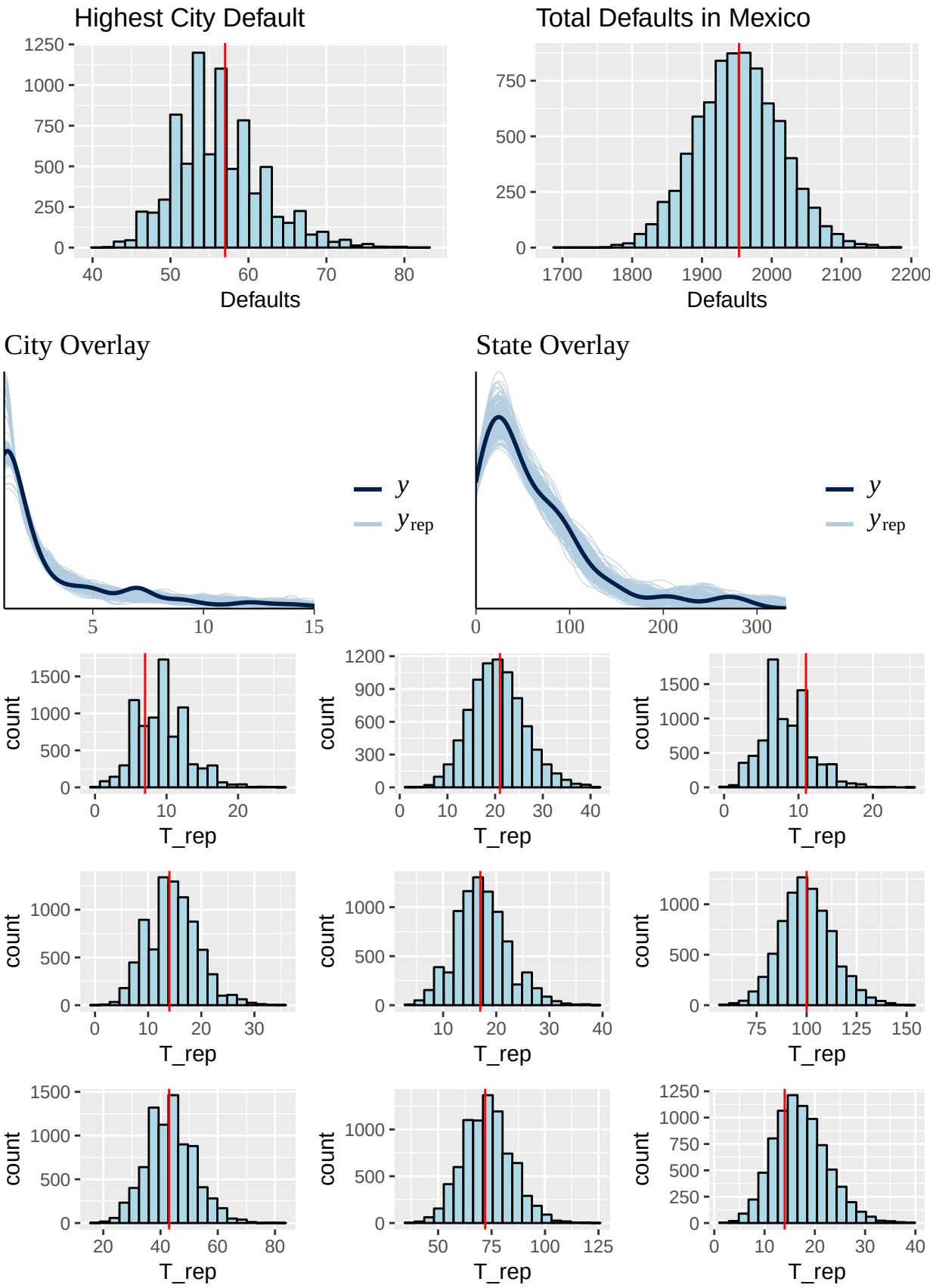
Results

```
## Inference for Stan model: car.
## 8 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=8000.
##
##          mean se_mean   sd  2.5%   50% 97.5% n_eff Rhat
## alpha   -2.73    0.00 0.05 -2.83 -2.73 -2.63   376 1.02
## beta[1] -0.13    0.02 0.31 -0.74 -0.13  0.46   258 1.03
## beta[2]  0.14    0.02 0.28 -0.39  0.14  0.70   268 1.03
## beta[3] -0.17    0.01 0.16 -0.49 -0.17  0.15   342 1.02
## beta[4] -0.09    0.00 0.08 -0.24 -0.09  0.07   600 1.01
## beta[5]  0.01    0.00 0.08 -0.15  0.01  0.17   797 1.01
## beta[6] -0.09    0.01 0.10 -0.30 -0.09  0.11   361 1.01
## beta[7] -0.12    0.00 0.08 -0.26 -0.13  0.03   581 1.00
## beta[8]  0.22    0.00 0.09  0.06  0.23  0.40   525 1.02
## beta[9]  0.09    0.00 0.07 -0.05  0.09  0.21   554 1.03
##
## Samples were drawn using NUTS(diag_e) at Fri Dec  7 19:50:50 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

[...][Show the phi's graphs]

Evaluation

As always, we start with the PPC for the total number of defaults


Highest City Default
Defaults
Total Defaults in Mexico
Defaults
City Overlay
State Overlay
y
y_rep
count
T_rep

[. . .][Replicate Yi's PPCs plots]

Also, we were glad to find that the RMSE for the CAR model is 1.9284 (0.28) which is a bit above of the other regression models but not by much. Moreover, it does reduce the standard held-out deviation to from 0.6 to 0.28.

Testing out Nonparametric methods

Finally we incorporated a Gaussian Process as our last analysis. Rather than an extension of the previous model, we see this as an alternative route that we took to understand if there were any interaction or nonlinearities in the data. Moreover, since this model fits well to the data, we leave as a future extension the addition of the ICAR prior to this model (although it would probably take days to fit).

Specification

Nonparametrics do not follow the same generative interpretation from the previous discussions. Now, instead of a generative process over numbers, we have a probability distribution over functions. As the name suggests, the Gaussian process relies on the multivariate normal distibution where now instead of a mean and covariance matrix we have a mean *function* and a covariance *function*. Mathematically,

$$y \sim MVN(a + f(x), K(x|\theta))$$

where a is an intercept value we included, $f(x)$ is the realization of a function over the N inputs and the positive-definite matrix takes the common form of

$$K(x|\alpha, \rho, \sigma)_{i,j} = \alpha^2 \exp\left(-\frac{1}{2\rho^2}\sum_{d=1}^{D}(x_{i,d} - x_{j,d})^2\right) + \delta_{i,j}\sigma^2,$$

where α, ρ and σ are the hyperparametrs. In this context α is called the *marginal standard deviation* and it controls the magnitude of the range of the function modeled by the GP. Moreover, ρ is called the *length-scale* parameters and links to the smoothness of the function represented by the GP.

Parameter Recovery

We do not test for parameter recovery since the model was taking too long to run. We distributed the large running time for CV and results.

Results

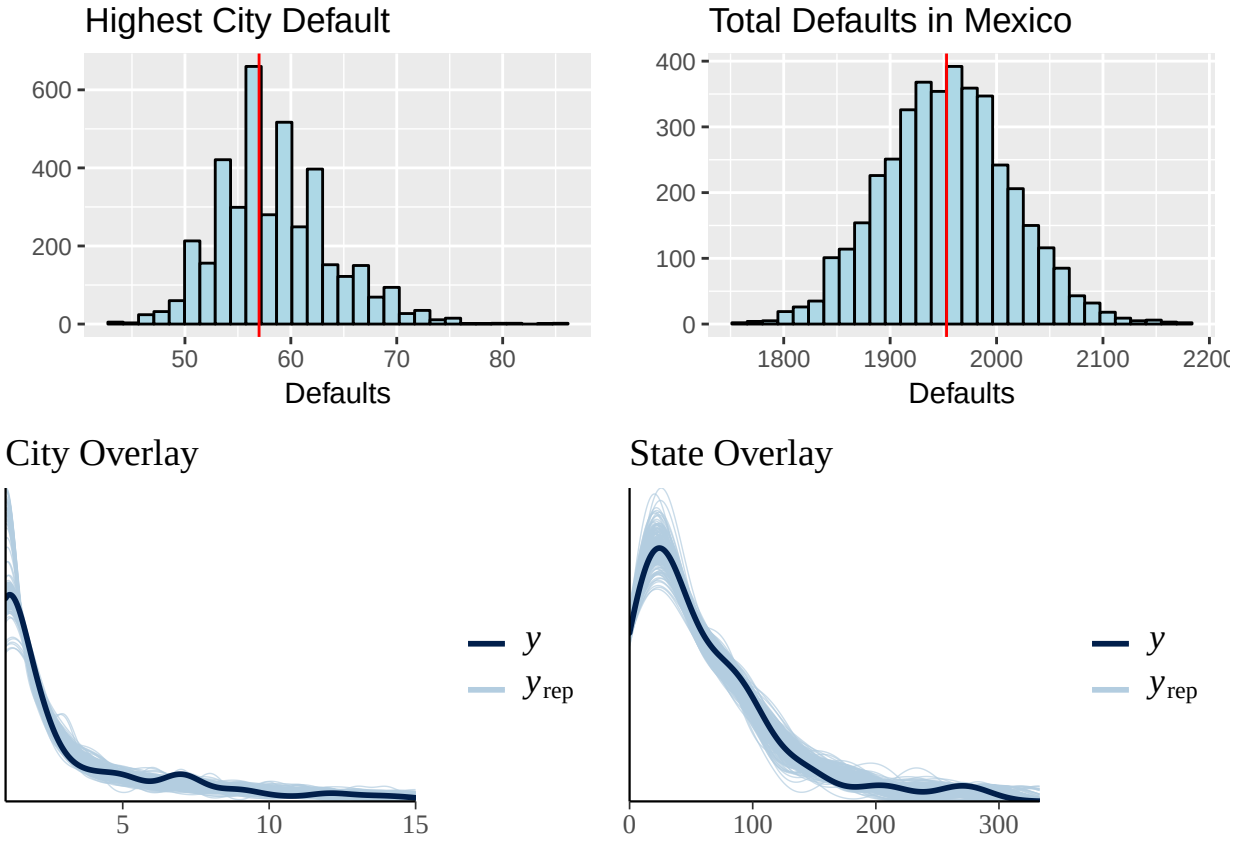
```
## Inference for Stan model: binomial_GP.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##        mean se_mean   sd  2.5%   50% 97.5% n_eff Rhat
## alpha  0.93    0.01 0.40  0.33  0.87  1.89  1423    1
## rho    1.43    0.01 0.51  0.69  1.35  2.61  2427    1
## a     -1.52    0.02 0.75 -2.68 -1.63  0.21  2168    1
##
## Samples were drawn using NUTS(diag_e) at Sat Dec  8 00:27:54 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
```

```
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

From the value of ρ we see that it is close to 1.5 and does indicates certain level of smoothness (which is evidence that our linear models were a defensible modelling assumption). Furthermore, the intercept a is much lower than in the previous models (where it was denoted as α). This is relevant to our analysis since it suggests that possible interactions of the given variables make take some of the explanatory effect from the intercept. This is reinforced by the fact that it is not the nonlinearities that are taking the weight-off the intercept (since ρ is close to 2) but rather some interactions.

Evaluation

Below is the performance of the GP over our set of four relevant PPCs



were we see that the model, at this levels, equally as good as the previous models. We were afraid that adding much more complexity could distort this. Moreover, we did a a single CV fold where the value was 1.71. It is lower than the values seen in the previous models however are not sure if this is just a result of chance, since maybe it was a train / test partition that was favorable for this model.

At this point we were unsure on how to proceed extracting more value out of our GP analysis. We also leave for future work to understand if there were some interactions suggested by the model as well as the nonlinearities (if present).

Conclusion

[. . .][Add the main learning that we got from this analysis]

Stan Code

[. . .][Add the GitHub link where they can find our code]