uc3m | Universidad Carlos III de Madrid

Master Degree in Statistics for Data Science
2020-2021

Master Thesis

BAYESIAN ANALYSIS OF DAILY URGENCY ALERTS IN MADRID CITY

Marta Ilundain Martínez

Stefano Cabras

Madrid, September 2021

SUMMARY

The aim of this Master Thesis is to predict daily urgency alerts in Madrid City. To do so, we combine a Multinomial model and a Poisson model through a Bayesian analysis in order to obtain the counts of four different categories of calls. To predict the number of calls we first estimate the parameters of the multinomial which represent the proportion of the types of calls. Such proportion is further scaled using a Poisson model for the total number of calls. Estimations of posterior quantity have been performed using STAN, a probabilistic programming language that uses Monte Carlo Markov Chains to carry out Bayesian computation. The data used in this work consists of daily records of the counts of four types of calls taken since 2007 for 11 years until 2018, therefore the statistical unit is the day and the prediction of counts for each type of call is made daily. We will analyze the results in order to see if we obtain a good prediction and we will propose alternatives that help us improve our Bayesian model with the goal of obtaining a better forecast.

Keywords: Bayesian Inference, Multinomial model, RStan, Multinomial logit, Poisson Model, Gamma Model, Time Series Forecasting.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The study of the probability of an event occurring has always played an important role in humanity. Hundreds of years ago, the study of probability was based on intuitive interpretation, observation and experience, such as thinking that it might rain if there were black clouds in the sky or analyzing the chances of being killed by an animal threat based on your size or strength, and it was used to draw conclusions and act according to what was observed and the probabilities that such event would occur. Of course these examples correspond to the most primitive and simple part of life and history, just based on the sense of survival, but these and many more events that occurred and were studied in the past have guided us to where we are now, living in a world where almost everything is countable and can be predicted.

Among all the important developments that have occurred throughout history that have helped us reach the level of technology we have today, it is worth highlighting the birth of Bayesian Statistics in 1763, named after Thomas Bayes. The history of the birth of Bayesian statistics spans several hundred years, from Thomas Bayes formulating the basis and dying without publishing his work, then Richard Price, who was the one who found the archives and published them, and finally ending when Laplace developed the Bayesian interpretation of probability [Stigler, 1986].

Bayes' contributions to Bayesian statistics are specially the introduction of the probability using a belief, a prior, the formalization of the continuous expression of Bayes' theorem for the Bernoulli's distribution and also the introduction of the Bernoulli's distribution as a prior. On the other hand, many years later Laplace contributed with more complex mathematical objects [Gómez Villegas, 1994]. Among all the contributions, it should be noted the study of the proportion of birth, the first version of the Central Limit Theorem and the justification of least-squares estimators using Bayesian statistics [Gómez-Villegas, 2018].

Nowadays and specially in the last decade, the large development of Monte Carlo Markov Chains (MCMC) has had an incredible impact in Bayesian Statistics. In fact, the applications of Bayesian analysis in industry and government are increasing, but specially in Science, where this increase is even faster and there are now active groups and projects formed by Bayesian researchers. This improvement is due to the advance in computational areas, where very complex models usually can only be formulated computationally using Bayesian techniques. MCMC has become the most popular method of Bayesian computation because of its power in controlling very complex situations and because it is quite easy to program [Berger, 2000]. There are a lot of packages in R that can be implemented in order to estimate many Bayesian models using probabilistic languages, such as BUGS, JAGS, NIMBLE, RStan, Greta,... and in this thesis we use one of them: RStan.

The aim of this work is to use Bayesian statistics in order to predict the number of calls of an urgency call-center. The data consists on daily urgency alerts in Madrid, which are related to traffic accidents, and the alerts are distributed in four types. We do not know what each type of call corresponds to, but using a Bayesian analysis, we can study the data and develop beliefs based on what is observed. Given this beliefs, we create a model that can predict the total number of calls for each type. In this way, we implement our model in Stan and obtain several results, which are then used to make a simulation (together with a process by which we calculate the total number of daily calls) and thus obtain the prediction of the total number of calls for each type.

In this work we explain first the main Bayesian models we use in a theoretical way, then we analyze the data, doing a small pre-process, adding some new variables collected from the AEMET, showing the main characteristics of our dataset,... Next, we apply the Bayesian models to our data and finally we study the results obtained and draw conclusions from the analysis, evaluating how good the predictions are and how good our model fits the data.

2. THEORETICAL FRAMEWORK

The main Bayesian models we will see in this thesis are the Multinomial logit model and the Poisson model. The multinomial logit model will be introduced using RStan, therefore, in order to understand better how this program works, we will explain its main characteristics in this chapter.

2.1. Multinomial logistic regression

The Multinomial logistic regression is a classification method that generalizes logistic regression to multiclass problems, where we have more than two possible discrete outcomes [Greene, 2012]. It is is used to predict the probabilities of the different outcomes of a categorically distributed dependent variable, given a set of independent variables which can be dichotomous or continuous. Multinomial logistic regression uses maximum likelihood estimation to evaluate the probability of categorical membership.

In multinomial logistic regression, it is useful to evaluate the multicollinearity with simple correlations among the independent variables and check that the data has a correct sample size in order to work better and efficiently with it [Starkweather, 2011]. The main assumptions regarding the multinomial logistic regression model are:

- The assumption of independence among the dependent variable outcomes, which states that the choice of or membership in one category is not related to the choice or membership of another category, the categories are independent.
- Assumption of non-perfect separation, if the groups of the outcome variable are perfectly separated by the predictors, then we will obtain an estimation with unrealistic coefficients.

The multinomial logit regression is a generalized linear model that estimates the probabilities of the C categories of a qualitative dependent variables Y using a set of independent variables **X** [Carpita et al., 2014].
For a number of subjects $i \in \{1, ..., N\}$, we have the response variable of that subject i y_i, which is a vector of non negative integers of length C categories, therefore y_{ic} for $c = 1, ..., C$ represents the counts of categories observed for n_i (total number of independent observations of subject i) , each observation is classified in one of the C categories. Then, given p_{ic} the probability that an observation of the total n_i falls into category $c \in \{1, ..., C\}$, we have that for each subject i, y_i will be distributed given the multinomial distribution:

$$y_i|\cdot \sim MN(n_i, p_{i1}, ..., p_{iC})$$

Thus, in order to fit a logistic regression model to each of the categorical probabilities, we have to take into account a vector β_c of coefficients for each category which helps us model the probability that an observation from y_i falls into one of all the C categories. In this way, the probability follows the next formula [Fisher and McEvoy, 2020]:

$$p_{ic} = p_c(x_i) = \frac{exp(x_i^T \beta_c)}{\sum_{j=1}^{m} exp(x_j^T \beta_j)} \; with \; c = 1, 2, ..., C$$

Where $\boldsymbol{\beta}_c$ is the vector of the regression coefficients of X that correspond to category c of Y and each vector x_i contains the observed data of all the independent variables associated with the observation vector y_i.

For each subject i, we have a multinomial distribution of $\boldsymbol{Y} = (Y_1, ..., Y_C)$ with parameter n and probabilities $\boldsymbol{p} = (p_1, p_2, ..., p_C)$, with $p_c \geq 0$ c=1,...,C and $\sum_{c=1}^{C} p_c = 1$. The probability mass function of Y is:

$$L(\boldsymbol{p}|\boldsymbol{y}) = P(Y_1 = y_1, ..., Y_C = y_C) = \frac{n!}{y_1!...y_C!} p_1^{y_1} ... p_C^{y_C}$$

Where $\boldsymbol{y} = (y_1, ..., y_C)$, with y_c are non negative integers that satisfy $\sum_{c=1}^{C} y_c = n$. The PMF can be written in the form of an exponential family using $\theta_c = log(p_1/p_C)$ for c=1,...,C-1, where we establish one of the categories as the baseline, therefore:

$$P(Y_1 = y_1, ..., Y_C = y_C) = \frac{n!}{y_1!...y_C!} \left(1 + \sum_{c=1}^{C-1} e^{\theta_c}\right)^{-n} exp\left(\sum_{c=1}^{C-1} y_c \theta_c\right)$$

The maximum likelihood estimator of $\boldsymbol{p}$ is $\hat{\boldsymbol{p}} = (\hat{p_1}, ..., \hat{p_C}) = (Y_1/n, ..., Y_C/n)$ (for each subject i) [DasGupta and Zhang, 2005].

Regarding the Bayesian approach, we will have to establish a prior on $\boldsymbol{\beta}_j$ based on our beliefs. The selected prior will be a generic informative prior, a standard normal distribution. In this way, the prior of β has mean zero and a standard deviation chosen by us to give good results. The posterior distribution of the normal is also a normal, hence we know the posterior distribution of the coefficients.
Moreover, in this work we will use the logit in order to formulate the model in Stan and simulate the coefficients. We estimate C-1 logit equations, one for each category relative to the reference category c=C. In this way, we obtain C-1 logit equations, where p_{iC} corresponds to the baseline (category C as reference, but it could be any other) [Williams, 2021]:

$$logit(p_{ic}) = log\left(\frac{P(Y_i = c)}{P(Y_i = C)}\right) = \alpha_c + \sum_{d=1}^{D} \beta_{cd} X_{id} \; with \; k = 1, ..., C - 1$$

Being D the number of independent variables in the model. Therefore, we obtain a probability for each subject i and each category k.

2.2. Poisson

In this work, we also use the Poisson model in order to obtain the total number of observations for any subject i n_i. Considering the Poisson model, which counts the number of events in a unit window of time $Y_1, Y_2, \ldots, Y_n|\theta \sim$ i.i.d. Poisson(θ):

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n|\theta) = c(y_1, \ldots, y_n) \cdot \theta^{\sum_{i=1}^{n} y_i} \cdot e^{-n\theta}$$

Where n is the total number of observations of our sample. The characterizing property of the Poisson is that the expectation and variance are equal to:

$$E[Y|\theta] = \theta$$

$$Var[Y|\theta] = \theta$$

Following the Bayesian procedure, we have to set a prior representing our beliefs and, in this work regarding the Poisson model, we have two options:

- For the Poisson sample model, an uncertain positive quantity θ has a gamma(a, b) distribution, the conjugate prior is the Gamma:

  $$p(\theta) = \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta}\, for\, \theta, a, b > 0$$

  Which leads to a Gamma posterior, so, based on our beliefs we need to select the proper values for the gamma distribution.

  $$\theta \sim gamma(a, b)$$

  And given the Poisson distribution of our variable Y: $Y_1, ..., Y_n|\theta \sim Poisson(\theta)$, we obtain the posterior distribution:

  $$\{\theta|Y_1, .., Y_n\} \sim gamma\left(a + \sum_{i=1}^{n} Y_i, b + n\right)$$

  Therefore, the posterior expectation is:

  $$E[\theta|y_1, ..., y_n] = \frac{a + \sum y_i}{b + n} = \frac{b}{b + n}\frac{a}{b} + \frac{n}{b + n}\frac{\sum y_i}{n}$$

  Where b can be interpreted as the prior sample size and a the number of success in that prior sample.

- If we consider that we do not have any prior knowledge, we select the Jeffrey's prior, which is an improper prior but the posterior distribution exists and is proper. It is a non-informative prior with the additional fact that it is invariant under reparametrization. Also, for a Poisson model with just one parameter θ, the Jeffrey's prior is a second order matching prior, which means that the credible interval we have in the Jeffrey's prior for θ is similar to the confidence interval for θ even at small samples. It follows the following formula:

  $$\pi_J(\theta) \propto \theta^{-1/2}$$

  And given the Poisson distribution of our variable Y: $Y_1, ..., Y_n|\theta \sim Poisson(\theta)$, we obtain the posterior distribution:

  $$\{\theta|Y_1, ..., Y_n\} \sim gamma\left(0.5 + \sum_{i=1}^{n} Y_i, n\right)$$

  This prior can be also viewed as the limit:

  $$\lim_{b \to 0} gamma(0.5, b)$$

  So the prior corresponds to gamma which has b tending to zero, and the posterior is also a gamma distribution.

  And finally, its posterior expectation is:

  $$E[\theta|y_1, ..., y_n] = \frac{0.5 + \sum y_i}{n}$$

The choice of one prior or the other comes from the study of the dataset and its characteristics, which will be chosen in following sections once we have presented our data. Nevertheless, the impact of choosing one prior over the other depends on the sample size, if we have a small sample the prior choices only influence very slightly on the posterior conclusions.

2.3. RStan

Stan is a probabilistic programming language written in C++ and released in 2012, named in honor of Stanislaw Ulam (1909-1984), co-inventor of the Monte Carlo method, which is used to specifying statistical and probability models for statistical inference. Stan was created by Andrew Gelman and Bob Carpenter, with a development team formed by 34 members. Using Markov Chain Monte Carlo methods, Stan is able to provide Bayesian inference for models with continuous variables.

Stan can be implemented in R using the RStan package and the interface supports sampling and optimization-based inference with posterior analysis. It declares data and lets us introduce parameters with constraints, it also defines a log posterior and it fits the model and predicts data thanks to the implementation of gradient-based Markov chain Monte Carlo algorithms. Unlike other similar programming languages (such as JAGS and BUGS), in RStan Markov Chain steps are generated with a method called Hamiltonian Monte Carlo (HMC), which can be more efficient than other samplers in JAGS and BUGS, particularly when fitting large and complex models (for data with highly correlated variables) [Marín, 2019].

The main fields in which Stan is used are social science, pharmaceutical statistics, market research and medical imaging.

2.3.1. Multinomial Logistic Regression in RStan

Given that this work is primarily Bayesian, it is worth mentioning how the multinomial logit regression model can be coded in Stan. The purpose of this section is only to give a concise idea of the code, without being too rigorous, since the RStan code that corresponds to multinomial logit regression model created specially for our dataset is in the Appendix (7) and will be explained with more detail later in this thesis.

The code we are going to use as a basis in this work can be obtained from the Stan User's guide. It lets us establish a prior on beta in a very easy way, coded in a vectorized form. To formulate the multinomial logit distribution, we use the *categorical_logit* function, which applies softmax internally to convert an arbitrary vector to a simplex:

$$\text{categorical \_ logit}(y \mid \alpha) = \text{categorical}(y \mid \text{softmax}(\alpha))$$

Where $\text{softmax}(u) = exp(u)/sum(exp(u))$. This softmax function is used to normalize the output of the logit to a probability distribution. We use this function because, although softmax is invariant under adding a constant to each component of its input and therefore the model is usually only identified if there is an acceptable prior on the coefficients, in this way we use K-vectors as parameters and the prior applies to all of the vectors, so we obtain estimation of the betas of all the categories and we are able to study all of them and its characteristics, with a posterior distribution related to the prior we set.

Another option that the Stan User's Guide [Stan Development Team, 2020] offers consists in using (K-1)-vectors by fixing one of them to be zero, but it requires a little bit more effort when coding and it does not allow us to study the estimates of the betas for all categories.

We have to take into account that this RStan code built with the function *categorical_logit* takes a total number of observations for any subject i which is unknown and random, therefore from the RStan code we obtain a vector of size C categories which corresponds to the probabilities and they sum up to 1, but they are normalized to a total number of observations which we don't know and can't access from this code. For this reason, in order to obtain the posterior distribution of our actual data, we use the Poisson model, which will help us obtain the total number of observations for any subject i.

3. DESCRIPTION OF THE DATASET

In this chapter we will describe our data set and study its main characteristics, at first with a little of pre-processing just to adjust our data to the described model, then explaining the main variables we have and the ones that we will use the most and also some properties of the data using graphs.

Our data consists on daily urgency alerts in Madrid related to traffic accidents. We have data from September 2007 to October 2018, that's eleven years of collecting data. In our dataset we have four main variables that are the subject of study in this work. They correspond to the number of calls in a day for the four types of call. We also have the date, three columns of day, month and year and a fourth column that corresponds to the weekday. These are all the variables we have in our original dataset.

3.1. Pre-process

In relationship to our dataset, it is interesting to add a new variable related to the frequency of traffic accidents in a day: the weather. This new variable can be obtained trough the State Meteorological Agency (AEMET), an agency of the Government of Spain responsible for providing weather forecast, among other things. We can access the data of many weather stations, in our case we will use the data from the Retiro station, from the AEMET OpenData. This station is located at 667m of altitude, 40º24'43"N, 03º40'41"W.
Among all the variables that we have (such as altitude, different measurements about wind and its direction, also about temperature and pressure,...), we will focus only on the precipitation. In the AEMET OpenData it is defined as the total amount of precipitation during the previous hour, expressed in mm. Therefore, we have a variable related to the rain that initially is a character. We turn it into factor in order to see if we have something abnormal, in this way we can study the values that this variable takes. We have data that goes from 0.0 to 45.3 and another value that corresponds to "Ip". In the AEMET OpenData, they explain that the Ip code means negligible precipitation, that is, an amount less than 0.1 mm, thus we will give it value zero, since this little precipitation does not affect on traffic accidents.

The second modification we have to do to our dataset corresponds to the variable weekday, which has seven levels (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday). In order to implement this variable in our model We have to convert this variable into six dummy variables, so we can indicate the weekday using just 0s or 1s. In this way, the logit equation would have six beta coefficients to express the day of the week, which is an improvement compared to just one beta coefficient for the whole weekday variable.

Finally, in order to adapt our data set to the Poisson model, we create a final variable which corresponds to the total amount of calls in a day, that is, the sum of all the calls of the four types. In this way, it will be easier to implement the Poisson model.

3.2. Explanation of the variables and visualization of the data

After this short pre-process, we can see the first seven observations of our final dataset in the following table:

d	m	y	type0	type1	type2	type3	Nt
1	9	2007	10	6	3	1	20
2	9	2007	12	6	3	1	22
3	9	2007	15	7	2	1	25
4	9	2007	21	8	3	1	33
5	9	2007	10	13	3	1	27
6	9	2007	10	10	4	2	26
7	9	2007	20	8	2	1	31

d	m	y	Mon	Tue	Wed	Thu	Fri	Sat	Prec
1	9	2007	0	0	0	0	0	1	0.0
2	9	2007	0	0	0	0	0	0	0.0
3	9	2007	1	0	0	0	0	0	0.0
4	9	2007	0	1	0	0	0	0	0.0
5	9	2007	0	0	1	0	0	0	0.0
6	9	2007	0	0	0	1	0	0	0.0
7	9	2007	0	0	0	0	1	0	0.0

Table 3.1

First seven observations of our final dataset

It is necessary to make clear that the statistical unit is the day, when predicting based on the Multinomial and Poisson models we do daily predictions, therefore the unit on which we base our results is a solar day.

We have information for a total of 4079 days, and although we have already slightly defined all our variables, we are going to define them again in order to study the main characteristics and their evolution throughout the years.

- Type0, Type1, Type2, Type3: these variables correspond to the number of calls in a day for each type. We don't know what each type of call corresponds to, but we can observe the frequency with which each type occurs. In fact, in the following table we can see the average of total calls per day for each type along with the total amount of calls over the years:

	type0	type1	type2	type3
Average calls per day	21	8	2	1
Total calls	87087	34116	8983	4520

Table 3.2

Average number of calls per day and total calls for all the types

As we can see, type0 is the most frequent type, followed by type1. For type2 and type3, we have a huge difference compared to the others since they don't have the same magnitude order. Although type2 has twice the incidence of type3, these are the less frequent types.

It is interesting to study the evolution of the total amount of calls throughout the years, in case the trend has changed. This can be seen in the following graphs:

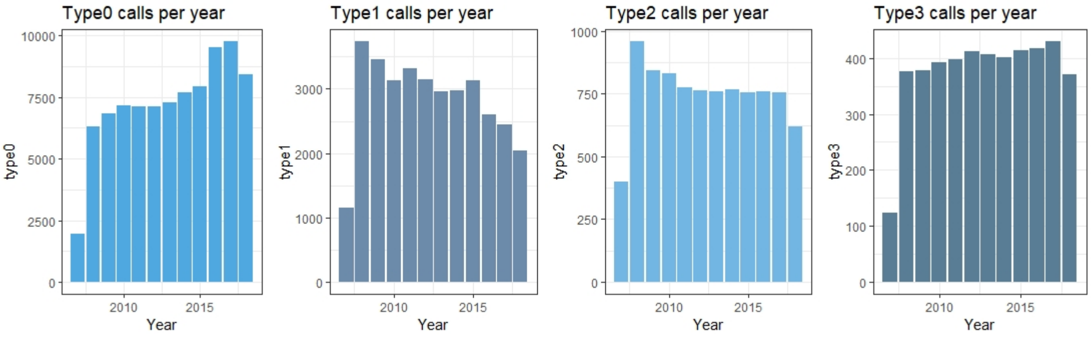

Fig. 3.1. Total calls per year for the four types

In figure 3.1 we can see that type2 and type3 are quite constant over the years, taking into account that the first observation which corresponds to 2007 is referred to the last four months of the year, so it is expected to appear with less volume of calls, the same happens for the last year, where we only have data until October 31, 2018. Regarding type0 and type1, we can see that type0 increases and type1 decreases, both of them in approximately a 25%. Given that this is a noticeable change in the number of calls, it would be better to compare the percent of calls for each type over the years in order to see if the trend does not change. This can be seen in the following table:

	2007	2008	2009	2010	2011	2012
Proportion type0	0.54	0.56	0.60	0.63	0.61	0.62
Proportion type1	0.32	0.33	0.30	0.27	0.29	0.28
Proportion type2	0.11	0.08	0.07	0.07	0.07	0.07
Proportion type3	0.03	0.03	0.03	0.03	0.03	0.03

	2013	2014	2015	2016	2017	2018
Proportion type0	0.64	0.65	0.65	0.72	0.73	0.74
Proportion type1	0.26	0.26	0.26	0.20	0.18	0.18
Proportion type2	0.07	0.06	0.06	0.05	0.06	0.05
Proportion type3	0.03	0.03	0.03	0.03	0.03	0.03

Table 3.3

Proportion of every type of call per year

Now it is easier to see if the total number of calls for each type has changed over the years and we have confirmed our suspicions: since 2016, type0 calls have increased and type1 calls have decreased, being this the biggest change regarding the proportion of calls.

This affects directly to the multinomial part of our model. After seeing that the trend of the proportion of calls has two steps (one before 2016 and another one from then on), we could suggest that in order to train our data correctly we would have to separate the data and estimate two different groups of betas. Nevertheless, this will be discussed in the results section, where we will see if it is necessary to separate the fitting of the model in order to obtain better results, or if this change over the years does not affect to the final results.

- Nt, Mon, Tue, Wed, Thy, Fri, Sat: these variables are defined all in this block because it is interesting to see if there is some change in the total number of calls over the week, and if it has changed over the years too. As we have defined in the pre-process, the weekday variable was turned into six dummy variables that indicate with 0s and 1s the day of the week, where Sunday corresponds to the case in which all of them are zero. The variable Nt corresponds to the total number of calls for each day.

  We can study the relationship between these variables using a simple boxplot, where we can see the number of calls for each day of the week:

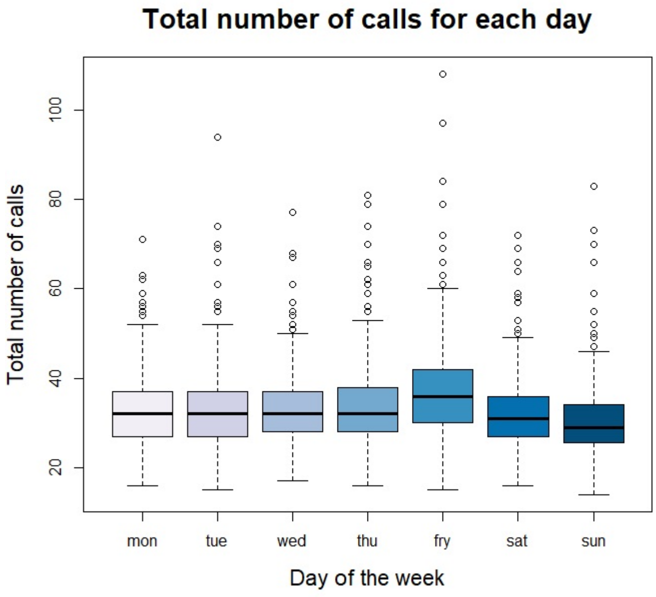

Fig. 3.2. Total calls for each weekday

As we can see, the amount of total calls over the week is stable, except for Friday, where we have more incidence. We have a growth of approximately 20%, which is not much but still remarkable, although we conclude that we will not add this variable to the Poisson model since just one day of the week has a different boxplot.

We can also study how many calls we have per year and its evolution in the following table:

	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Total number of calls	3620	11376	11502	11526	11604	11425	11424	11848	12223	13298	13406	11454

Table 3.4

Total number of calls per year

The total number of calls per year remains more or less constant during the 11 years of data, with values around 11000 calls per year. We can notice the increase since year 2016, there is a growth of approximately the 20%, but it happens only for two years and it may not be important.

Since we have stated that variables weekday and year won't be added to the Poisson model, we will chose the Jeffrey's prior considering that we only have one parameter and we prefer to use a non-informative prior.

- Prec: this variable, related to the weather of a certain day, has already been defined as the total amount of precipitation during the previous hour, expressed in mm, but in order to implement it in an optimal way in our multinomial model, we must turn it into a dummy variable with two levels that represent the strength of the rain. The AEMET says that we have weak rain for less than 2mm/h, normal rain for 2mm/h to 15mm/h and heavy rain for more than 15mm/h. Using this information, we create a new variable called Rain which says whether it is raining or not, being no rain for precipitation less than 2mm/h (taking value 0) and rain for higher than that (taking value 1).

It is interesting to see if, in the case that we have rain, which type of call is increased. Therefore, in the following two figures we can observe the relationship between each type of call and variables Prec and Rain, given the day of the week:

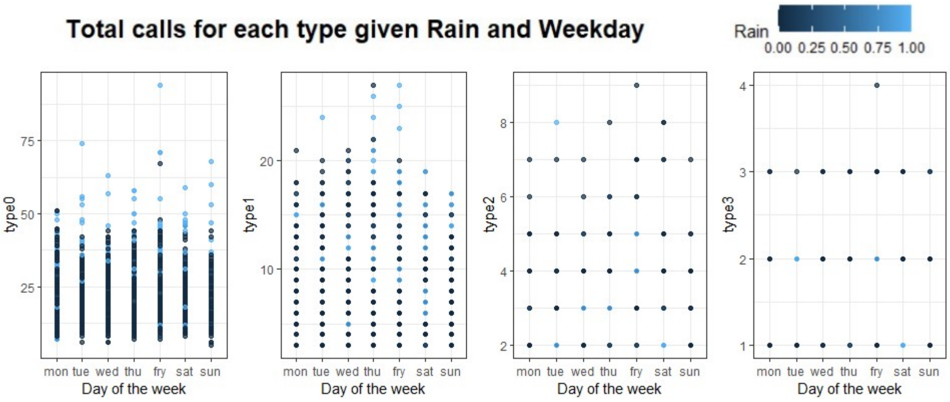

Fig. 3.3. Total calls given weekday and rain

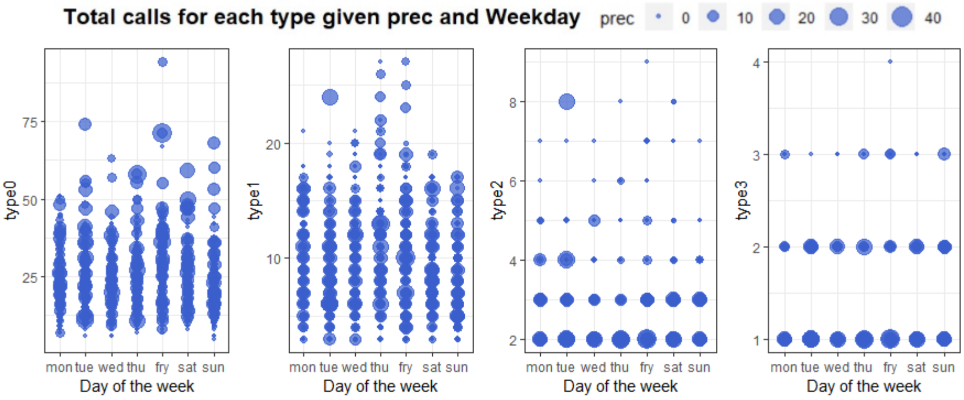

Fig. 3.4. Total calls given weekday and prec

As we can see in these two figures, it seems that there is no relation between Rain and more calls of type2 and type3, but it does seem it for calls of type0 and type1, where we have more calls if it is raining.

In order to compare the results, we have created the following graph, in which we have the evolution for each type of call over the years.

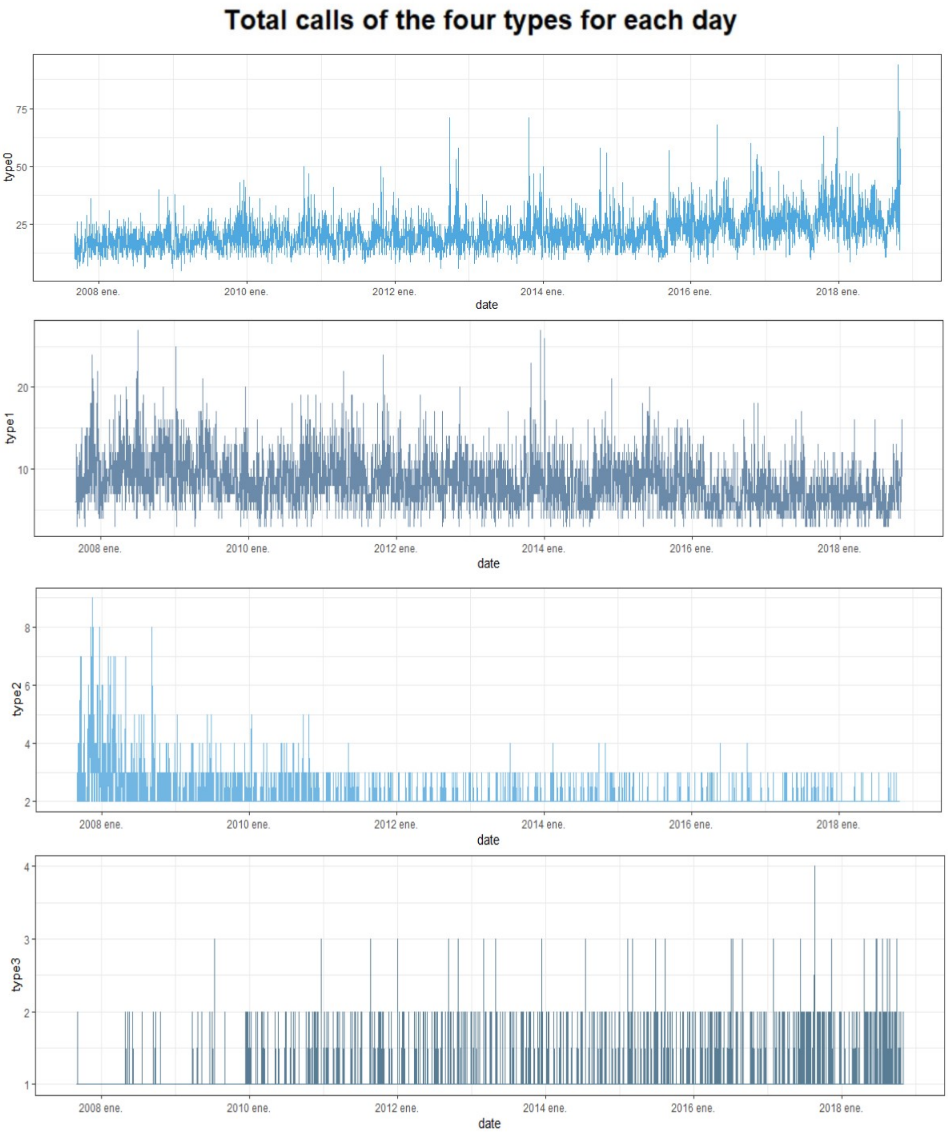

Fig. 3.5. Total calls for the four types over the years

In these plots we have the total number of calls of each type for each observation, that is, for the 4079 days. It is useful to plot the data in this way in order to compare them with the predicted number of calls for each type, which will be done in section 5. The goal is to plot the actual data and the prediction together in order to see how good the predictions are.

Once we have defined all the variables and seen its main characteristics, taking into account how to implement them correctly to our model, we can start to apply the theoretical framework to our data. In the following sections we will establish the Bayesian model we will use to predict data, and such results will be studied later.

4. MODEL

We will assume that the counts of the different categories of Y (number of calls) follow a multinomial distribution, with explaining variables Rain (dummy variable that says if it rains (1) or not (0)), the weekday (which are six dummy variables that indicate the day of the week) and the number of calls of each type for the previous day. The goal is to predict the response probability of variable Y, which has 4 categories Y_0, Y_1, Y_2, Y_3, in relationship to these explaining variables and also the previous counts of Y (number of calls) for every type of call, thus the prediction of for example number of calls of type1 depends on the weather, the weekday and the number of calls of all the types in the previous day.

Given a day t, where t=1,...,T with T=4079, $\boldsymbol{Y}_{j,t}$ the vector of calls of the four types (j=0,1,2,3 types) and the total amount of calls in day t N_t (which is fixed and unknown in this multinomial model), we have that the joint distribution of $\mathbf{Y}_t = (Y_{0,t}, Y_{1,t}, Y_{2,t}, Y_{3,t})$ for a certain day t given the total number of calls $\mathbf{Y}_t|N_t$ follows a multinomial distribution, with probabilities $\mathbf{p}_t = (p_{0,t}, p_{1,t}, p_{2,t}, p_{3,t})$, denoted by:

$$\mathbf{Y}_t|N_t, \mathbf{p}_t \sim Multinomial(p_{0,t}, p_{1,t}, p_{2,t}, p_{3,t})$$

The probability mass function of Y is (we omit the t subindex in order to shorten the notation, but it corresponds to any day in the data):

$$P(Y_0 = y_0, Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) = \frac{n!}{y_0!\, y_1!\, y_2!\, y_3!} p_0^{y_0}\, p_1^{y_1}\, p_2^{y_2}\, p_3^{y_3}$$

where $\mathbf{y} = (y_0, y_1, y_2, y_3)$ with y_j's non negative integers that satisfy $\sum_{j=1}^{4} y_1 = N$ for any day in our data.

One of these probabilities p_j can be implicit in the multinomial model, since all probabilities add up to 1, $p_j \geq 0$, $\sum_{j=1}^{4} p_j = 1$. Therefore, in order to predict the proportion of calls in a certain day t we calculate the correspondent parameters of p_0, p_1, p_2 using the logit, that is $\theta_0, \theta_1, \theta_2$. We also include the explanatory variables Rain and the six dummy variables for weekday in the prediction. Given this, we have the logit:

$$\begin{aligned} Logit(\theta_j) = \alpha + \beta \cdot X = \alpha + \beta_1 Y_{t-1}^{(0)} + \beta_2 Y_{t-1}^{(1)} + \beta_3 Y_{t-1}^{(2)} + \beta_4 Y_{t-1}^{(3)} + \beta_5 rain_t + \beta_6 Mon_t + \\ + \beta_7 Tue_t + \beta_8 Wed_t \ + \beta_9 Thu_t + \beta_{10} Fri_t + \beta_{11} Sat_t \ for \ j = 0, 1, 2 \end{aligned} \tag{1}$$

Therefore, for a given day t, we can predict the number of calls of type 0, 1, 2 and 3 using this formula, where they depend on the number of calls of each type of the previous day and the precipitation and weekday that corresponds to day t, being a total of 11 explanatory variables.

The parameters α and β are simply the regression coefficients of the linear regression of the logit(θ_j) in terms of X, being θ_j the parameter of a Multinomial distribution, which indicates the proportion of calls of type j (for a certain day t). Regarding α and β, our prior beliefs are represented in this case by means of proper priors, where both of them are normally distributed with mean 0 and variance 0.1:

$$\alpha \sim N(0, 0.1) \text{ and } \beta \sim N(0, 0.1)$$

Also, the matrix X, which represents the predictors of this multinomial logistic regression, has a special form, in which the variables Type0, Type1, Type2 and Type3 (or $Y^{(j)}$ as represented in the formula) depend on the previous observation of day t (which is t-1) and for the variables that show the weather and the weekday we have that they correspond to that day t. Therefore, we will be predicting 4078 observations, since for the first day we don't have data of the previous day.

To sum up, and comparing the written model with the RStan code (7), the code is constructed in a way in which we start with 11 predictors, 4 types of calls and t number of days. In this way, we have our matrix X, and using the priors, we have a categorical logit of t vectors that correspond to the actual data, we use the function categorical logit in order to estimate the coefficients of the logit. What we do in RStan is to obtain the matrix of posterior betas given the code, therefore we obtain a matrix of 11x4 which corresponds to the 11 predictors for the 4 types of calls.
So, if y is a vector of 4 elements, the matrix X has dimensions tx11, $X \cdot \beta$ is tx4, and we would obtain t days vectors of y with the 4 proportions. Therefore, we have to take into account that the posterior matrix β and parameter α obtained (which are normally distributed) are the ones we use to obtain the proportion for each type of call of any given day t. We only have 2500 simulations for the matrix β and the parameter α, but the logit is carried out using the mean of all these simulations (we simulate the betas just one time for all the t chosen days, there is not a group of betas for each day given the restrictions of the RStan code). Therefore, the posterior of the logit of the probabilities is referred to the posterior distribution of the betas and alpha, which is a normal, but using the posterior mean of β and α. Nevertheless, we have to take into account that the logit is not a linear operator, so the posterior of the proportions does not correspond to the posterior distribution of the coefficients βs and α. Although we calculate them using these parameters, the proportions have a different posterior distribution, which is a logistic distribution.

To obtain the predictions of the proportions from the multinomial model, we just have to substitute in equation 1 the correspondent data for t days of matrix of predictors X along with the mean of all the 2500 simulations of β and α. In this way, we simplify the θ_j of equation 1, obtaining finally a matrix of tx4 that corresponds to the proportions of calls for each day t and the 4 types of calls.

Nevertheless, we have to take into account that this multinomial model takes a fixed number of calls N for each day t which is unknown. In order to implement correctly this parameter in the model we must know that it is random, since the total number of daily calls follows a random Poisson distribution with an unknown mean parameter, therefore we would have to establish a prior on N.

We formulate the Poisson model as:

$$N_1, N_2, \ldots, N_t|\theta \sim \text{i.i.d. Poisson}(\theta), \text{ i.e.,}$$
$$\Pr(N_1 = n_1, \ldots, N_t = n_t|\theta) = c(n_1, \ldots, n_t) \cdot \theta^{\sum_{i=1}^{t} n_i} \cdot e^{-t\theta}$$

Here we have t=4079 days, the number of observations in our sample. We assume then that data result from a Poisson process which counts the number of events in a unit window of time, being a day such unit in this context.

Following the Bayesian procedure, we have studied our data and we have seen that the total number of calls changes in approximately a 20% over the years. Since we haven't carried out a hard study regarding the total number of calls, we have finally selected the Jeffrey's prior, which was already explained in section 2.2. Therefore, we set:

$$\pi_J(\theta) \propto \theta^{-1/2}$$

Finally, using the Bayes' formula, we know that the posterior distribution of θ will be:

$$\{\theta|N_1, N_2, \ldots, N_t\} \sim \text{gamma}\left(0.5 + \sum_{i=1}^{t} N_i, t\right) \tag{2}$$

Then, the posterior mean:

$$E(\theta|n_1, \ldots, n_t) = \frac{0.5 + \sum_{i=1}^{t} N_i}{4079}. \tag{3}$$

In this way, to model correctly the number of calls of each type, we have to implement this parameter N in the model. To implement it correctly, we do a certain number of simulations of the total number of calls for every day t that we are predicting. That is, we use the variable Nt, which shows the total number of calls for each day, and we calculate the posterior mean using the formula 3 obtained from the posterior gamma distribution 2. We calculate the posterior mean for each day t and then we simulate a hundred observations from this mean, we take the mean of these hundred simulations and we obtain a list of t elements, where each one of them corresponds to the total number of calls for that day.

Combining both models using the parameters θ_j, proportions for each type of call (obtained from the multinomial model), and N_t, total number of calls in a day (obtained from the Poisson model), we are able to carry out a certain number of simulations of a multinomial distribution. In this way, we will obtain the final predictions of the number of calls of each type of call for any day t.

5. RESULTS

In this section, we analyze the final results. Although the methodology has already been explained in the previous chapter 4, it is necessary to make clear some aspects about the process and the correspondent code. Therefore, in this chapter we are going to explain step by step what to do in order to obtain the final predictions.

First, as described previously in the multinomial model, we need to obtain the estimations for matrix of coefficients β and coefficient α. The inputs of the RStan code (7) are K=4 types of calls, t the selected number of days that we use to fit the model, y[t,K] that corresponds to t rows of the four type of calls of our data, D=11 the number of predictors in the logit and finally the matrix X of size [t,D] obtained from the dataset. This matrix X, as we have mentioned before, has a special form. For the four columns that correspond to the four types of calls, we have that the observations correspond to the day before t-1, therefore we erase the last observation (t=4079) since we will not predict a 4080th day (it does not exist). The other seven variables correspond to the day we are predicting (six for the day of the week and one for rain), so we erase the first observation because we don't have information about the number of calls for each type of the previous day.
Once we have all our inputs, we run the fit of the model and we obtain posterior of the coefficients $\boldsymbol{\beta}$ and α. We have four groups of results, one for each type of call, and each group has 11 columns that correspond to the number of variables we use in the model. There are 2500 rows in each group, but in order to use this elements, we work with the posterior mean of each column, therefore for each group we obtain 11 means. In this way, we obtain a final matrix β of 11 rows and 4 columns, where for example the element (1,1) corresponds to the coefficient β_1 for the first type of call, that is, j=0 in equation 1 and element (8,3) corresponds to β_8 for j=3. Working with this matrix and the coefficient α is very easy since in order to calculate the estimations of the proportions, we just substitute in equation 1. In this way, we obtain a matrix of tx4 that shows the proportion of each call for every day t.

The following step consists on estimating the total number of calls using the Poisson model. In order to do that, first we calculate the posterior mean for each day using the correspondent value of variable Nt and equation 3. With this value, we are able to simulate 100 values of Nt following a Poisson distribution, but we take the mean of these hundred values in order to obtain the total number of calls for each day t.

Using the proportion of each type of call and the total number of calls of a certain day t, we can obtain the final prediction of the number of calls for each type. To do so, for every day t we simulate a certain number of observations that follow a multinomial distribution, using as parameters the total number of calls and the type proportion for day t. The final predictions correspond to the mean of these simulations, therefore we obtain the results, where we have 4078 days (one day less than the original dataset) and the predictions of the four types of calls, represented as an integer and not as a proportion as we had previously obtained.

The goodness or badness of these final results depend on the models described, with the selected priors for the multinomial and Poisson models.

Given that these results are obtained from first a fitting using RStan and then two simulations of Nt and the final results, we can see that getting good or bad results depend on the selected number of days that we use to fit the RStan model or the prior of the betas (the prior of the betas can be changed by changing the standard deviation of the normal (informative prior), but the Jeffrey's prior of the Poisson model is non-informative and it does not work in the same way). One thing or the other makes the matrix of betas (and also the coefficient α, but it is much less important) change, and this is the cause of a good or a bad result. The selection of the prior has been made by trying different values and seeing which one is the best (so we are not going to change it), but we can still change the number of days to fit the model. Because of this, we have created 5 different cases, where we fit the model with a different number of selected days t. Now, we proceed to see the results of these processes:

5.1. Fit using 4078 days

First, we fit our model with the whole dataset, that is, we use 4078 days in order to obtain the posterior matrix of betas. In the appendix we can see the matrix of the posterior means for every β that we obtain (7), but in this section we will only study the final results, not the outcomes that we get for every process or simulation. Nevertheless, it is worth highlighting these beta matrices, since they are the outcome of the RStan process. In each matrix we only show the posterior mean of the βs that are significant, that is, their credibility interval does not contain value zero, since they are the ones that most contribute to the obtaining of the parameters of the multinomial. We know that they have a normal posterior distribution, but this posterior distribution changes when we substitute in the multinomial logit since the logit is not a linear operator. In this way, the posterior distribution of the proportions is not a normal. Anyway, analyzing the betas we can see that just the predictors regarding the number of calls of the previous day are significant, and variables weekday and rain are not important in the model. Also, although it has not been mentioned in every case, the coefficient α is zero in every fit.

We are going to show the table of the first seven days and a plot of the evolution of the 4 types of calls over the years (similar to plot 3.5) and we calculate the MAE of the four type of calls and the total number of calls in order to compare how good the predictions are compared to the actual data.

The Mean Absolute Error (MAE) is a measurement that shows the error between the predictions and the data, and we include it in this analysis only from a frequentist point of view to analyze the error, but it is not a typical way of proceeding in Bayesian analysis, we just include it to compare the results. In fact, we include the MAE which is expressed in calls, but we write it with two decimal places to give the specific result. The interpretation of these errors is expressed as an integer, which makes more sense since we are analyzing calls.

Therefore, in this case for a fit of 4078 days, we obtain:

d	m	y	type0	type1	type2	type3	Nt
2	9	2007	11	6	3	1	21
3	9	2007	11	6	3	1	21
4	9	2007	11	8	3	0	22
5	9	2007	13	9	3	0	25
6	9	2007	15	9	2	1	27
7	9	2007	13	9	3	1	26
8	9	2007	13	11	3	0	27

Table 5.1

First seven predictions for a fit using 4078 days

As we can see, we obtain similar results as the original data, where type0 and type1 are the most common calls, and type2 and type3 do not occur so often. In comparison to table 3.1, we have that these results seem to be more constant, and we have less amount of total calls. In the next figure, we can see how the number of calls change over the years:

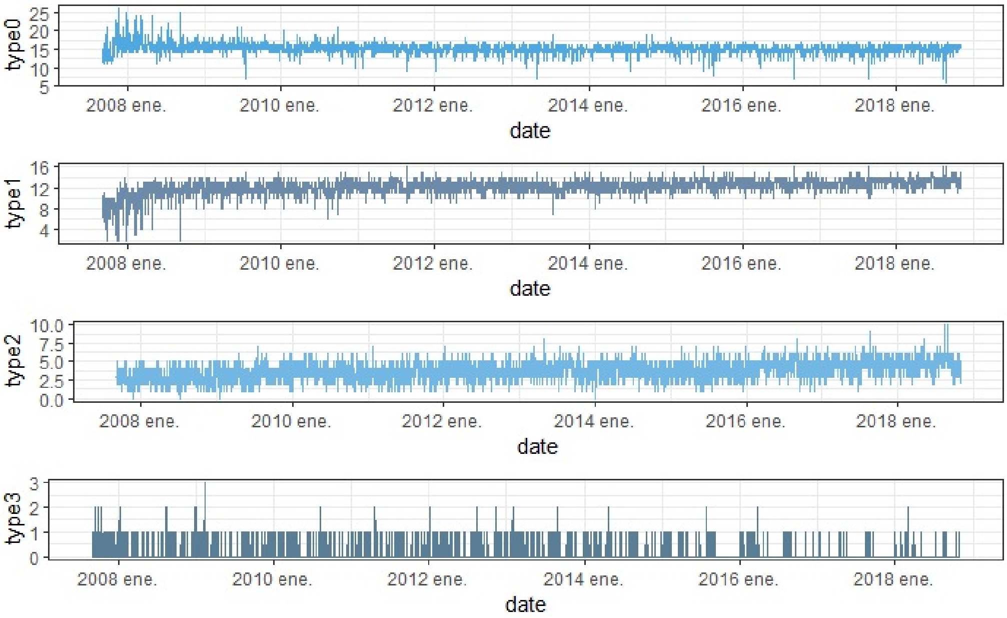

Fig. 5.1. Total calls for the four types over the years using a 4078 days fit

In these plots we observe that it is true that the predictions are more constant than the original dataset. We don't have an increase in the number of calls of type0 and type1 over the years, they stay in values that go from 15 to 25 calls for type0 and 12-16 for type1. Type2 seem to increase over the years, and type3 has a maximum value of 3 calls per day, therefore these results are very similar to the actual data.

Moreover, we obtain a MAE for each type of call and the total of calls of:

	type0	type1	type2	type3	Nt
MAE	7.30	4.54	1.78	0.96	6.09

Table 5.2

MAE for a fit using 4078 days

We obtain that the biggest difference in comparison to the original data corresponds to type0 calls, where we have an error of 7 calls, and type1, with an error of 4 calls. For type2 and type3 this error is not so important, but it is clear that the model does not predict well the type0 and type1 calls. This could come from the total number of calls error, since apparently we are predicting 6 calls less per day, and these loss of calls affects to the number of calls of type0 and type1.

5.2. Fit using 100 days

Next, we are going to use a small amount of days in order to obtain the matrix of coefficients, just to see if we obtain better results than in the previous case. The correspondent matrix of betas can be found in the appendix 7, and now the only significant coefficients are β_{11}, β_{13},β_{21} and β_{23}, therefore the number of calls of type2 and type3 of the previous day and weekday and rain are not important variables in our model.

Although we fit the model with a hundred days, we still predict the 4078 days of the data, and we present the results as in the previous case, in this way we are able to compare them similarly and analyze which fit is the best one. Therefore, we obtain:

d	m	y	type0	type1	type2	type3	Nt
2	9	2007	11	4	3	3	21
3	9	2007	12	4	2	2	20
4	9	2007	14	4	2	2	22
5	9	2007	17	4	2	2	25
6	9	2007	15	5	3	3	26
7	9	2007	15	5	3	3	26
8	9	2007	19	5	2	2	28

Table 5.3

First seven predictions for a fit using 100 days

In table 5.3 we have similar predictions as for the fit using 4078 days, although it looks like we have more calls of type0 and type3, and it seems to be more total calls too. This can be seen in a clearer way in the following figure:

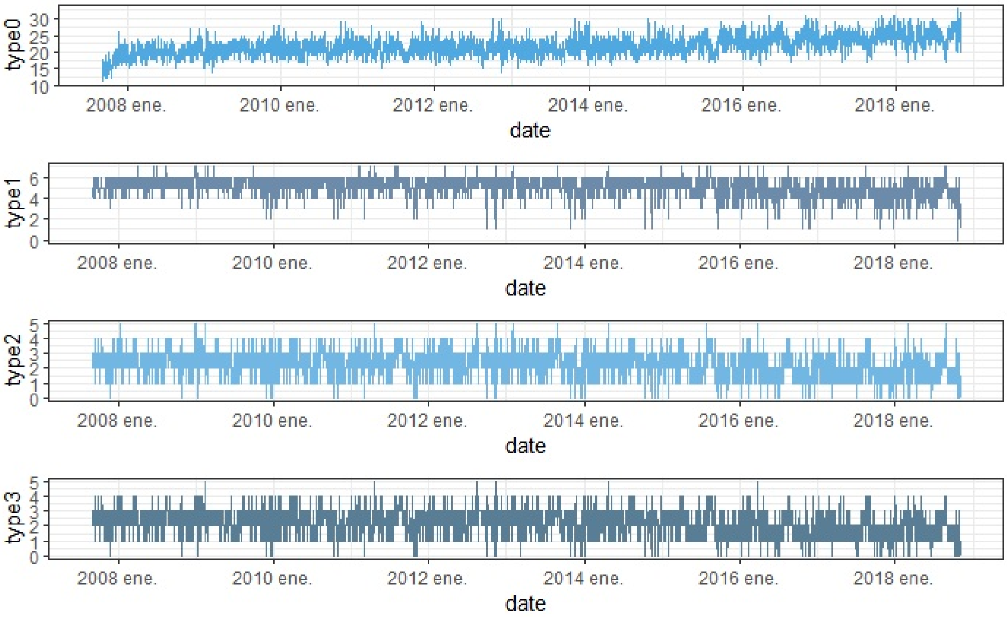

Fig. 5.2. Total calls for the four types over the years using a 100 days fit

For type0, we obtain an increase as in the actual data, and now the maximum value of calls is around 30 calls per day, which is much better than in the previous case in section 5.1. For type1, we have clearly less calls, which is reflected in the number of calls of type3, where we have more counts. On the other hand, predictions of type2 are very similar to the data. This directly affect the MAE:

	type0	**type1**	**type2**	**type3**	**Nt**
MAE	5.18	3.56	0.72	1.13	6.08

Table 5.4
MAE for a fit using 100 days

As expected, compared to the previous case we have a smaller MAE for type0, type1 and type2 but a higher MAE for type3. Regarding the total number of calls, the error has also been decreased but not by much. Nonetheless, this table 5.4 shows that we have obtain a good prediction since these errors are not worrying.

5.3. Fit using 1000 days

Since we have obtained quite well results using a hundred days to fit the model, we have also tried to use a thousand and see if we obtained even better predictions (the correspondent estimated matrix beta can be found in the appendix 7, where again just the first four predictors are significant). In the following table, we have the first seven prediction using this fit of t=1000 days:

d	m	y	type0	type1	type2	type3	Nt
2	9	2007	10	7	2	1	20
3	9	2007	11	8	1	1	21
4	9	2007	11	9	1	1	22
5	9	2007	13	11	1	0	25
6	9	2007	14	9	1	1	25
7	9	2007	14	9	2	1	26
8	9	2007	13	12	1	0	26

Table 5.5

First seven predictions for a fit using 1000 days

At first sight, it does not seem that we obtain better results than in previous cases, the number of calls for each type is quite similar to the table obtained using t=4078 days (5.1). Compared to the previous case (5.3), it looks like we have improved the predictions of type3 calls using t=1000 days, but type0 and type1 have worsened. Let's see the prediction over the years in the following figure:

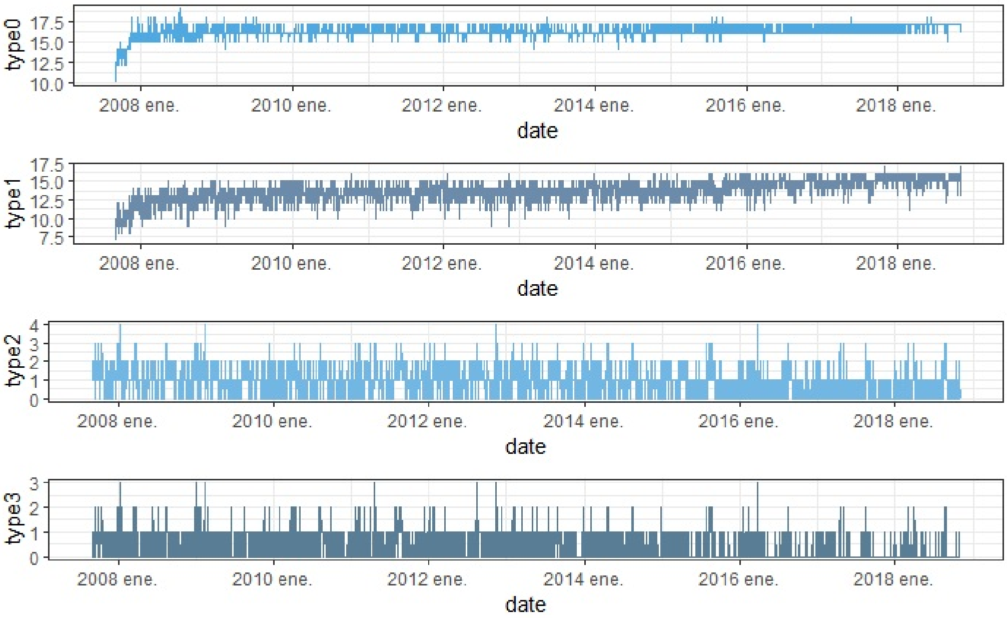

Fig. 5.3. Total calls for the four types over the years using a 1000 days fit

	type0	type1	type2	type3	Nt
MAE	6.46	5.68	1.33	0.77	6.12

Table 5.6

MAE for a fit using 1000 days

Analyzing figure 5.3 and table 5.6, we can see that the predictions for type0 are not good and we obtain an error of 6 calls, the same occurs for type1 with an error of 5 calls. Type2 and type3 are not that bad, in fact we obtain better predictions for type3 in comparison to the previous cases, but overall figure 5.3 does not resemble to the same figure for the actual data 3.5, the plots don't have the same structure for type0 and type1, and the predictions are bad.

5.4. Fit using data until 31/12/2015

Next, we are going to use t=3044 days, which corresponds to data until the last day of 2015 (included). Again, the matrix of β can be found in the appendix (7) with just the first four variables significant.

It is interesting to study this case since it corresponds to the 75% of the data and also because, as we have seen previously in table 3.3 from this year on, the proportions of calls change noticeably. Therefore, we have split the fit of the 4078 days into two periods: first a fit using data until the end of 2015 and another fit using data since 2016 (the last fit we will analyze), where the trend of the proportion of calls changes.

For the fit until 31/12/2015, we obtain the following results:

d	m	y	type0	type1	type2	type3	Nt
2	9	2007	10	6	3	1	20
3	9	2007	11	7	3	0	21
4	9	2007	11	9	3	0	23
5	9	2007	12	10	3	0	25
6	9	2007	13	8	3	0	24
7	9	2007	12	9	4	0	26
8	9	2007	12	11	3	0	27

Table 5.7

First seven predictions for a fit using data prior to 2016

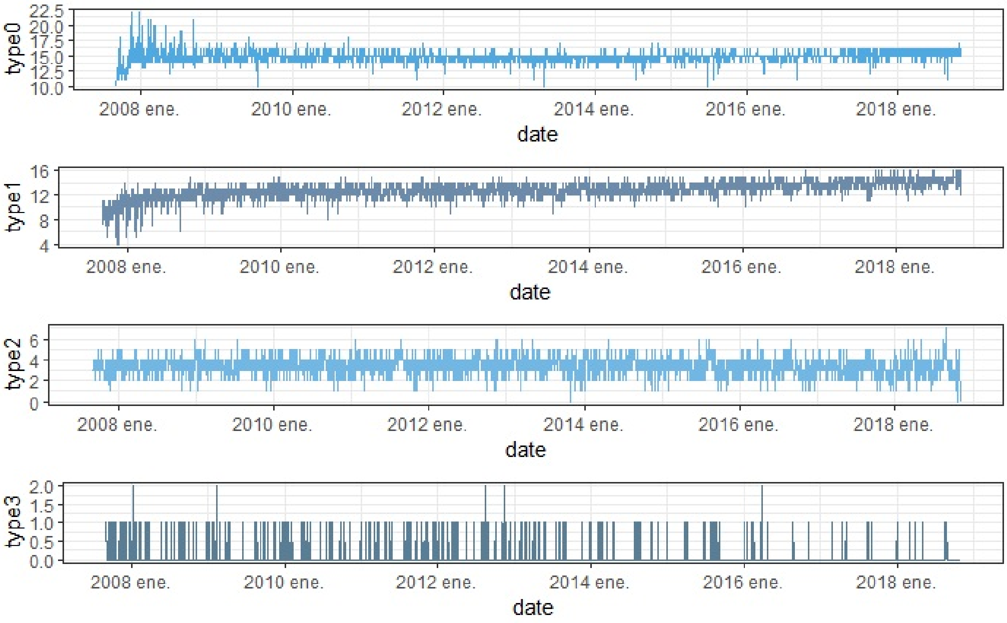

Fig. 5.4. Total calls for the four types over the years using data prior to 2016

	type0	**type1**	**type2**	**type3**	**Nt**
MAE	7.28	5.04	1.33	1.04	6.07

Table 5.8

MAE for a fit using data prior to 2016

As we can see, these results are not very good. The predictions on table 5.7 are similar to the ones obtained in the first case (section 5.1), the plots aren't analogous to the ones of the original data (3.5) and the MAE is higher than before for type0 and type1, therefore we conclude that this is not a good fit and the predictions are bad.

5.5. Fit using data since 01/01/2016

Analogously to the previous case, now we fit the model with the last 1034 observations in order to see if we obtain better predictions, although we don't expect them to be much better since the results of the previous case were bad. This is the last fit we will analyze in this work, the correspondent matrix of betas can be found in the appendix (7), with the same conclusion as before, we only have four significant predictors and weekday and rain are not important. In this case, we obtain the following predictions and errors:

d	m	y	type0	type1	type2	type3	Nt
2	9	2007	10	6	3	2	21
3	9	2007	10	7	3	1	21
4	9	2007	10	8	3	1	22
5	9	2007	13	9	3	0	25
6	9	2007	13	9	1	2	25
7	9	2007	12	9	2	2	25
8	9	2007	13	10	3	1	25

Table 5.9

First seven predictions for a fit using data since 2016

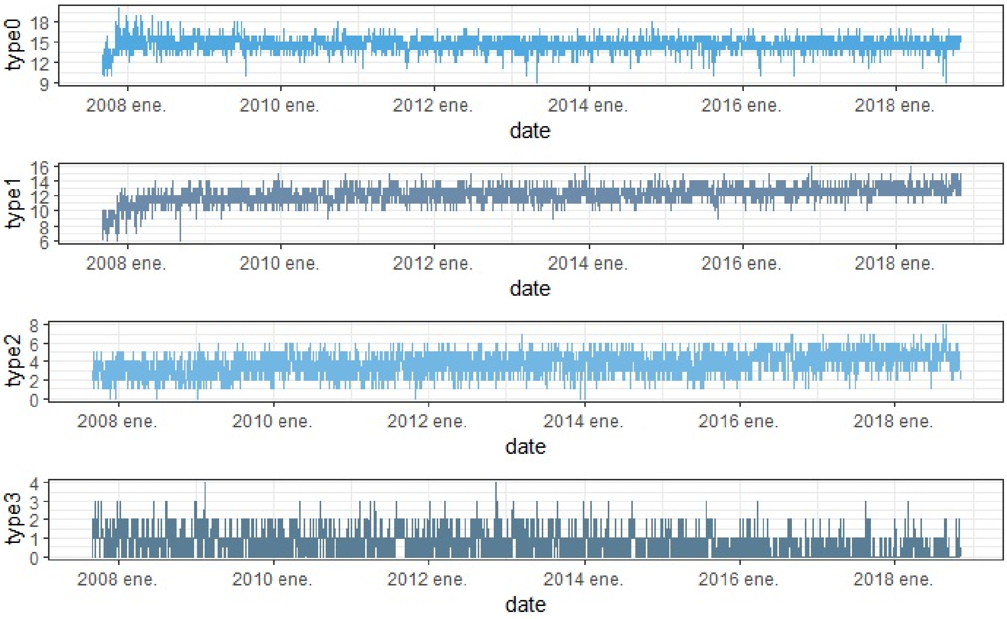

Fig. 5.5. Total calls for the four types over the years using data since 2016

	type0	type1	type2	type3	Nt
MAE	7.42	4.40	1.77	0.62	6.14

Table 5.10

MAE for a fit using data since 2016

Studying the results, we observe that it is not a good prediction. Table 5.9 is very lookalike to previous tables that showed a bad result, the figure 5.5 is not similar to the original data in figure 3.5 and the errors of type0 and type2 and Nt are quite noticeable, although we have obtained the lowest error regarding type3.

Thus, what we mentioned in previous section 3.2 about separating the fit in order to obtain better results is not true, and we obtain even worse predictions. It made sense to do it this way in order to obtain a higher proportion of type0 calls, given that the proportion of this type of calls increases as of 2016, and thus we could have obtained a better prediction. But, in the end, this fit turned out not to be the right one.

5.6. Best fit

By far, the best predictions correspond to the hundred days fit, in section 5.2. In this case, we have obtained the lowest MAE for type0, with a considerable difference compared to the second best fit, and the plots of each type of call over were the ones that most closely resembled the real data. In the following figure, we can see the predictions obtained (blue) together with the data (red):

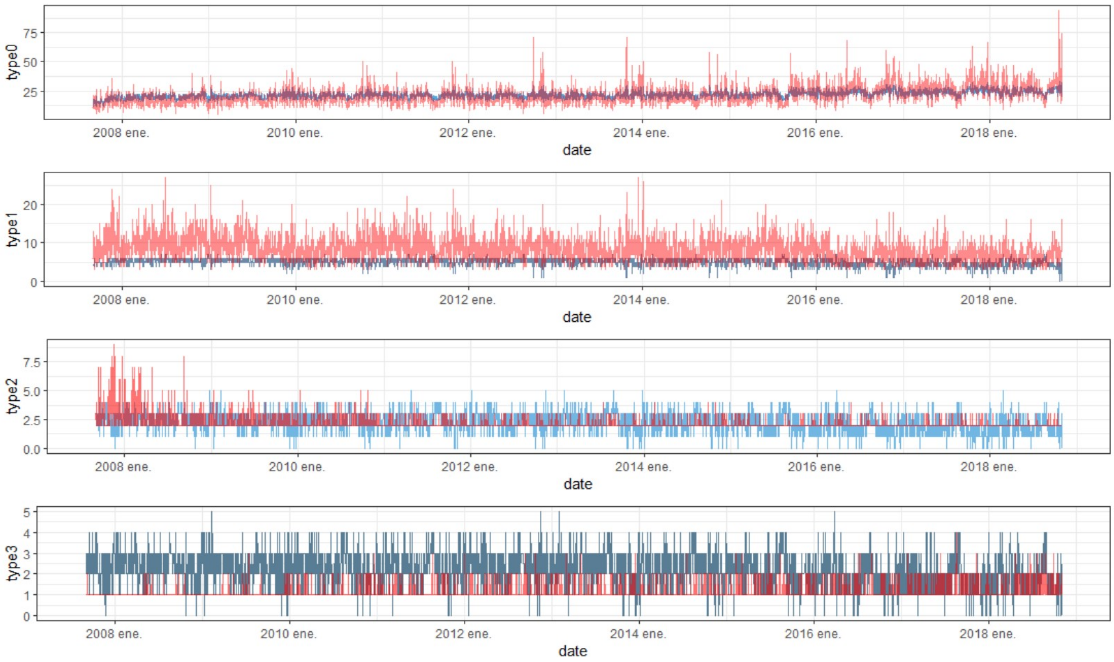

Fig. 5.6. Predictions of number of calls for each type vs Data

We can see that for type0, the prediction corresponds quite well to the data, although it is much more constant and smooth, with less ups and downs. For type1, the prediction is below the original data, so this prediction could be much better. For type2, the prediction fits very well the data except for the first year, where it is very different. Finally, for type3 we have obtained a good prediction, although it slightly exceeds the real values. In summary, the obtained prediction is satisfactory.

Even so, these results can be improved. There are many changes we could do to our model in order to improve our results, for example, we could change the prior of Nt in the Poisson model. In this way, instead of using a non-informative prior as is the Jeffrey's prior, we would use the conjugate prior of the Poisson model, which is the Gamma. Since we don't know if in fact this would improve our model, we have tried to change it in order to check if the Jeffrey's prior was the right choice. We have hypothesized that for one day b=1 (which is the statistical unit as we stated in the beginning of this work), we have a=30 calls, which is more or less equal to the mean of Nt of the real data, we have calculated the posterior mean for each day and finally the results of this model with the correspondent MAE. We have not included the results in this analysis since they were very bad, but since we've done it, and to illustrate that Jeffrey's prior was indeed a good choice, in the following table we can see the obtained MAE:

	type0	**type1**	**type2**	**type3**	**Nt**
MAE	9.90	15.51	5.20	0.8	28.24

Table 5.11

MAE for a fit using the conjugate prior

Definitely, this is not a good way to improve the model, so we propose other options, which we do not know if they would work or not, but for which we have not done the test (unlike for the conjugate prior).

More alternatives would be to add the dependency on the weekday of the total number of calls in the Poisson model (although in that case we wouldn't be able to use the Jeffrey's prior since for more than one parameter it is difficult to use) or to add more predictors in the multinomial logit, such as the year or the month, although as we have seen, rain and weekday were not significant in our model and maybe the added variables wouldn't be either. Nevertheless, this options result in making our model more complex, and we don't know if they would improve our final predictions a lot, therefore, we stay with the results obtained from section 5.2, which are a good prediction of the number of calls for each type.

6. CONCLUSIONS

In this work we have predicted the number of calls of four types of calls from a daily urgency call-center in Madrid. The original data consisted in 4079 days of information, in which we had the number of calls of type0, type1, type2 and type3, along with three variables that represented the date. Using this date, we were able to obtain data from the AEMET OpenData, and add one more variable that represented the rain on that certain day.

In order to carry out the prediction, we created a model that fit our data. To do so, we used a multinomial model where the parameters were the proportion of each type of call, and they could be calculated using the multinomial logit. With a Bayesian analysis, we were able to simulate the posterior coefficients of the predictors, and in this way we obtained a group of proportions that corresponded to the 4 types for each day we predicted. The multinomial model was developed using RStan, a programming language that has helped us to obtain the coefficients of the predictors using a Bayesian statistical model. Moreover, we also modelled the total number of calls for each day using a Poisson distribution and Bayesian inference in order to select the prior of Nt. Combining both models, we were able to obtain the predictions of the number of calls for each type.

Once we defined the whole process, we began to show different results based on the selected t days that we used to fit the multinomial model, therefore we obtained five different matrices of the posterior mean of the coefficients and five different groups of results. After analyzing all of them, we got that the best prediction was the one obtained using t=100 days to fit, which showed an MAE not so far away from the actual data.

The results of this best fit are good, but they can still be improved. We observed that the highest MAE of 6 calls corresponded to the total number of calls in a day Nt, so the Poisson model could be changed in order to obtain better results. For example, we could change the prior (although we have seen that using the conjugate prior didn't work) or the dependency on the weekday (in which case we couldn't use the Jeffrey's prior since we would have more than one parameter). Also, another change that could be done would be to add the variables year and month to the multinomial logit, but nothing assures us that we obtain better results and we would be making our model more complex. Maybe, in order to improve the predictions, we could change the β and α priors, but since these are beliefs regarding the data, it is a more subjective issue.

In conclusion, the predictions obtained are similar to the actual data, so it is a good result and the aim of this work has been accomplished successfully. In fact, this approach is not the only one that works with this kind of data. Another alternative that we could use would be to combine Long short-term memory (LSTM, an artificial recurrent neural network) and Bayesian Regression.

Studying these type of events is very useful, since having a good prediction on the total number of calls and each type of call can help us to improve different aspects regarding the urgency alerts in Madrid City. Since we do not know what each type of call corresponds to, we propose an example where predicting how many calls there will be can be very useful. Imagine for example that type0 and type1 calls correspond to minor accidents or incidents, such as a car in the median strip of the highway that needs a tow truck or a minor crash between two vehicles, and type2 and type3 correspond to major accidents, where an ambulance and paramedic help is needed. In this case, we can anticipate and have the necessary resources available when the incident occurs, so everyone can receive the help they need in an optimal way. This case, and many others, show that we must keep improving the field of statistics, building new tools that allow us to predict with greater accuracy what will happen tomorrow.

7. APPENDIX

Rstan Code

```
mlogit_model = "
data {
  int K; // 4 types of calls
  int t; // days to fit the model
  int D; // number of predictors
  int y[t,K]; // t rows of calls from data
  matrix[t, D] x;// matrix of predictors
}
parameters {
  matrix[D, K] beta;
  real alpha;
}
model {
  matrix[t, K] x_beta = x * beta;

  to_vector(beta) ~ normal(0, 0.1);

  alpha ~ normal(0,0.1);

  for (i in 1:t)
    y[i,K] ~ categorical_logit(alpha + x_beta[i]’);
}"
```

Matrices β

$$\beta_{t=4078} = \begin{bmatrix} 0.10 & 0.06 & ... & -0.17 \\ 0.21 & ... & -0.15 & ... \\ 0.86 & -0.63 & -0.19 & ... \\ -1.51 & 1.15 & 0.35 & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \end{bmatrix}$$

$$\beta_{t=100}=\begin{bmatrix} 0.22 & ... & -0.08 & ... \\ 0.11 & ... & -0.05 & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \end{bmatrix}$$

$$\beta_{t=1000}=\begin{bmatrix} 0.16 & 0.09 & -0.11 & -0.15 \\ 0.18 & ... & ... & ... \\ 0.20 & -0.15 & ... & ... \\ -0.23 & 0.23 & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \end{bmatrix}$$

$$\beta_{t=\text{until } 31/12/2015}=\begin{bmatrix} 0.15 & 0.09 & ... & -0.20 \\ 0.16 & ... & -0.05 & -0.11 \\ 0.62 & -0.48 & ... & ... \\ -1.16 & 0.97 & 0.20 & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \end{bmatrix}$$

$$\beta_{t=\text{since } 01/01/2016} = \begin{bmatrix} 0.09 & ... & ... & -0.14 \\ 0.18 & ... & -0.16 & ... \\ 0.40 & -0.29 & ... & ... \\ -0.83 & 0.65 & 0.17 & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & ... & ... & ... \end{bmatrix}$$

BIBLIOGRAPHY

Berger, J. O. (2000). Bayesian analysis: A look at today and thoughts of tomorrow. *Journal of the American Statistical Association*, *95*(452), 1269–1276. https://doi.org/10.2307/2669768 (cit. on p. 1)

Carpita, M., Sandri, M., Simonetto, A., & Zuccolotto, P. (2014). *Chapter 14 - football mining with r* (Y. Zhao & Y. Cen, Eds.). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-411511-8.00015-3. (Cit. on p. 3)

DasGupta, A., & Zhang, T. (2005). Inference for binomial and multinomial parameters: A review and some open problems (cit. on p. 4).

Fisher, J. D., & McEvoy, K. R. (2020). Bayesian multinomial logistic regression for numerous categories [work in progress]. Last updated December 11, 2020 (cit. on p. 4).

Gómez Villegas, M. (1994). *El problema de la probabilidad inversa: Bayes y laplace*. Siglo XXI de España Editores. (Cit. on p. 1).

Gómez-Villegas, M. S., M. A. & De Mora Charles. (2018). *Historia de la probabilidad y de la estadística*. UNED. (Cit. on p. 1).

Greene, W. H. (2012). *Econometric analysis (seventh ed.)* Boston: Pearson Education. (Cit. on p. 3).

Marín, J. M. (2019). An introduction to stan with r [Online; posted 22-January-2019]. https://codingclubuc3m.rbind.io/post/2019-01-22/. (Cit. on p. 7)

Stan Development Team. (2020). RStan: The R interface to Stan [R package Version 2.27]. http://mc-stan.org/. (Cit. on p. 7)

Starkweather, A. K., J. & Moske. (2011). Multinomial logistic regression. https://it.unt.edu/sites/default/files/mlr_jds_aug2011.pdf. Retrieved from url (cit. on p. 3)

Stigler, S. (1986). *The history of statistics the measurement of uncertainty before 1900*. Harvard University Press. (Cit. on p. 1).

Williams, R. (2021). Multinomial logit models - overview. Last revised March 6, 2021 (cit. on p. 4).