Bayesian Learning and MonteCarlo Simulation

Final project

Academic year 2021/2022
Professor: Bassetti Federico

Group members:
Castiglione Pasquale - 10657816
Job Charlotte - 10786162
Mariani Elisa - 10632876

Politecnico di Milano

TABLE OF CONTENTS

1. The dataset

1.1 Data Source

The data used in this paper have been taken from the site of the Italian Civil Protection. We used two different datasets: the first one ('Italy COVID-19 data'[1]) contains different features like the date, the number of deaths, the number of tests, the number of new positives, the number of hospitalized people etc. The second dataset ('Italy Region Color Zone'[2]) contains, for each day, the color of each region in Italy.

1.2 Feature choice and new variables

We decided to keep as features the ones that seemed the most interesting to the problems we want to model. Some of these variables are used directly in our final datasets, while others are used to create new features.

We created two datasets: the main one ("dataset.rds") will be used in paragraphs 2 and 3 and the other one ("weekly.rds") only in paragraph 4. In particular, both concern data for Lombardy but they cover different time periods. The main one contains daily data from 06/12/2020 to 05/07/2021. The second one contains weekly data (data averaged by week) from 01/06/2020 to 01/11/2021.

In the main dataset, we added 3 new columns: "newpos_av7D", "hospitalized_with_symptoms_av7D", "intensive_care_av7D". These columns contain the averages of the considered variable over the previous 7 days and they are created to easily access information to predict our outcomes (f.i. number of deaths at t+7). We have also added the column 'deathsH8' that will be the center of the analysis carried out in paragraph 2. 'deathsH8' contains the number of deaths 7 days ahead with respect to the current observation.

Finally, we added to this dataset the standardized version of the quantitative variables to have values that are all in the same range (useful in paragraph 2) and we merged it together with the dataset containing the color of the regions (created from "covid-19-zone.csv").

1.3 The variables

In the main dataset, at the end of the *data preparation* the variables are:

- **date**: Date of notification
- **new_positives**: Daily cases of infection
- **newpos_av7D**: Average number of the new positive cases in the 7 days before
- **color**: Current color of the region
- **deathsH8**: Number of deaths 7 days ahead
- **newpos_av7DSCALED**: scaled values of variable newpos_av7D
- **hosp_with_symptoms_av7DSCALED**: scaled values of variable hosp_with_symptoms_av7D (average number of hospitalized people in the 7 days before)
- **intensive_care_av7DSCALED**: scaled values of variable intensive_care_av7D (average number of people who entered the IC in the 7 days before)
- **deathsH8SCALED**: scaled values of variable deathsH8

In the smaller dataset, used in paragraph 4, the final variables are:

- **week:** incremental count of the number of weeks (1,2,..52)
- **intensive_care_week:** weekly average of the number of people who entered the IC
- **newpos_week:** weekly average of the number of people infected
- **hospitalized_with_symptoms_week:** weekly average of the number of people hospitalized (not in IC)

[1] *The link for the database is the following: "https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-regioni/dpc-covid19-ita-regioni.csv"*

[2] *The link for the database is the following: "https://raw.githubusercontent.com/tsiotas/covid-19-zone/main/covid-19-zone.csv"*

1.3.1 The categorical variable ‘color’

As regression requires numerical inputs, categorical variables as ‘color’ need to be recorded into a set of binary variables. In a case like this, where our categorical variable can have multiple values (N), we can use the ‘one-hot-encoding’ trick. This consists in recording our categorical variables with N levels into N-1 indicator variables which give the value ‘1’ if the observation is in the considered category and ‘0’ otherwise.

In our case we have 4 levels : ‘Bianca’, ‘Gialla’, ‘Arancione’, ‘Rossa’ and the ‘dummy columns’ to create are ‘Gialla’, ‘Arancione’, ‘Rossa’: when, for the same observation, all of them have the value ‘0’ means that the color for that observation is ‘Bianca’.

This is done automatically by R simply by considering the variable as a ‘factor’ with multiple levels. In the following paragraphs, doing regression, the first level statistics will be included in the ‘intercept’ and the other levels will be considered as autonomous variables.

Only when using JAGS (in paragraph 2.4) we actually created the 3 ‘dummy’ columns for convenience.

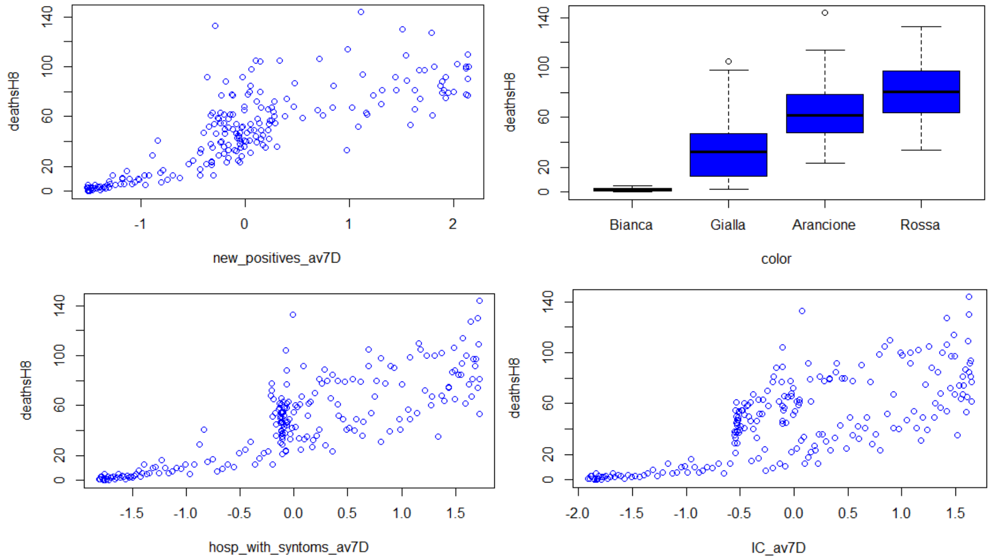
2. The first problem: a model for ‘deathsH8’

2.1. Description of the problem

The problem considered in this first part is that of estimating, on the basis of some variables at time t, the number of deaths on day t+7.

Thus, we use as y_i the variable ‘*deathsH8*’ and as covariates $x_{i1},x_{i2},x_{i3},x_{i4}$ respectively the variables ‘*new_positives_av7D*’, ‘*color*’, ‘*hospitalized_with_symptoms_av7D*’, ‘*IC_av7D*’.

We can start by plotting the data to have an idea of what we can expect.



As we can see from the plots, for the covariates we used standardized data to ensure that all variables had the same magnitude. In this way we can then have a direct comparison between their coefficients and those related to the categorical variable color which, as it was announced before, can only take values 1 or 0.

It’s clear that these variables will be somehow related with the number of deaths 7 days after: the victims of COVID19 are a part of the people found positive and it is very likely that they were hospitalized with symptoms in regular patients' rooms or in IC. The color of the region will be also related to the number of deaths since it is an indicator of the severity of the pandemic situation.

We are considering different days for the covariates with respect to the response variable to take into consideration the incubation time of the virus and the period of time between hospitalization and death[3].

[3] *According to the analysis of deceased patients published by ‘Instituto Superiore di Sanità’ (https://www.epicentro.iss.it/coronavirus/sars-cov-2-decessi-italia), based on data prior to 10/01/2022, the median time between hospitalization and death was 8 days.*

2.2 Frequentist approach

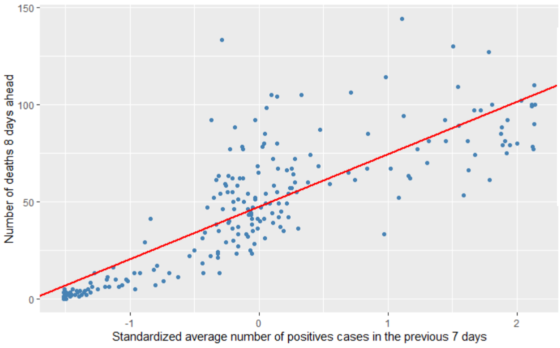
Even though we are working with count data (and we should use discrete models) we want to start by considering them as continuous and then we will move to a more suitable approach. We start with a *Frequentist OLS Linear Regression* approach. The considered model is:

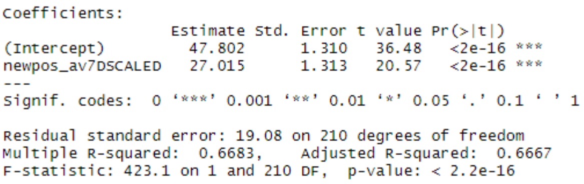
$$y \ = \ X\beta \ + \ \varepsilon \qquad\qquad \varepsilon \sim N(0,1)$$

where y was **deathsH8** and X is the vector of the covariates considered.

2.2.1 Simple Linear Regression

We try to estimate with only the covariate newpos_av7DSCALED the number of deaths at t+7.
Here is reported the plot and the main statistics:





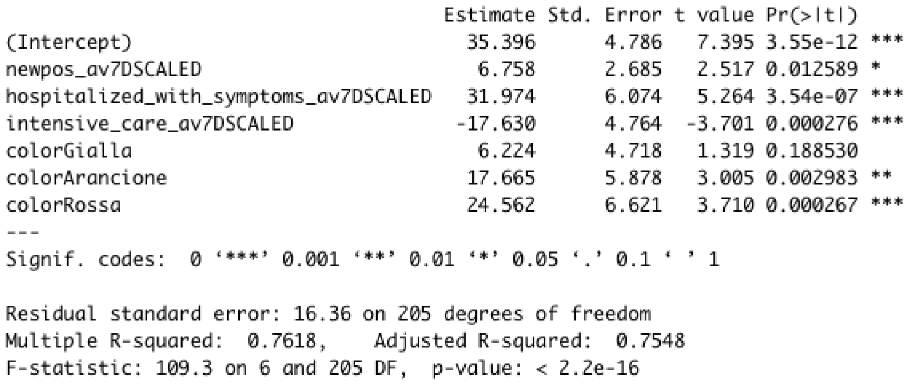
```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)         47.802      1.310   36.48   <2e-16 ***
newpos_av7DSCALED   27.015      1.313   20.57   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.08 on 210 degrees of freedom
Multiple R-squared:  0.6683,	Adjusted R-squared:  0.6667 
F-statistic: 423.1 on 1 and 210 DF,  p-value: < 2.2e-16
```

We observe, by looking at the p-values, that both the intercept and the coefficients of the covariate are significant for the problem.

2.2.2 Multiple Linear Regression

Now we use ‘*newpos_av7DSCALED*’, ‘*hospitalized_with_symptoms_av7DSCALED*’, ‘*intensive_care_av7DSCALED*’ and ‘*color*’ as covariates.



```
                                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)                             35.396      4.786   7.395 3.55e-12 ***
newpos_av7DSCALED                        6.758      2.685   2.517 0.012589 *  
hospitalized_with_symptoms_av7DSCALED   31.974      6.074   5.264 3.54e-07 ***
intensive_care_av7DSCALED              -17.630      4.764  -3.701 0.000276 ***
colorGialla                              6.224      4.718   1.319 0.188530    
colorArancione                          17.665      5.878   3.005 0.002983 ** 
colorRossa                              24.562      6.621   3.710 0.000267 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.36 on 205 degrees of freedom
Multiple R-squared:  0.7618,	Adjusted R-squared:  0.7548 
F-statistic: 109.3 on 6 and 205 DF,  p-value: < 2.2e-16
```

From the summary we can see how the different covariates affect the model. In the column estimate, we observe the weight given to the variables. The standard error tells how precisely the estimate is measured. The t value is used to test whether or not the coefficient is significantly different from 0. If it is not significant (PR>t), then the coefficient will not add anything to the model and the variable could be dropped.
In particular, looking at the *p-values*, we can observe that our covariates are all significant except the ‘colorGialla’ one. The R-squared value, that is preferred as closer to 1 as possible, here is quite high.
QQ-plots and residual plots are available in the Appendix A.1.

2.3 Bayesian linear regression - Gaussian likelihood

Considering the response variable as Normal, like in the previous paragraph, we want to start by performing a Bayesian normal linear regression. Therefore, now we assume that the model is the following:

$$y = X\beta + \varepsilon \qquad \varepsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$$

Accordingly, the likelihood we want to initially use is a simple Gaussian distribution:

$$y=(y_1, ..., y_n) \sim N_n(\beta_0\mathbf{1}+\beta\mathbf{X},\sigma^2 I_n) \quad \beta = (\beta_1, ..., \beta_k)$$

where **1** is a vector of ones of length n and I_n is the identity matrix.

Note that the error term has now a prior distribution and that we are assuming that our observations are i.i.d. Our goal is to update the unknown parameters β and σ^2 based on the data (y_1, ..., y_n) and the covariates (x_1, ..., x_n) where n is the number of observations.

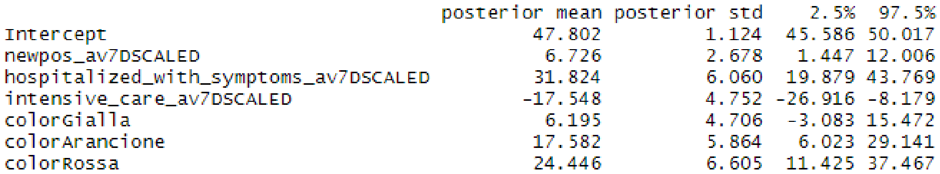
2.3.1 Zellner's informative G-prior

Using the parametrization that divides β_0 from $\beta=(\beta_1, ..., \beta_k)$, as used in BAS, the (centered) Zellner's informative G-prior is:

$$\beta = (\beta_1, \ldots, \beta_k)$$
$$\beta|\sigma^2, X \sim \mathcal{N}_k(0, \alpha\sigma^2(X^tX)^{-1})$$
$$(\beta_0, \sigma^2)|X \sim \pi(\beta_0, \sigma^2) = \sigma^{-2} \qquad \alpha > 0$$

The hyperparameter α can be interpreted as a measure of the amount of information available in the prior relative to the sample. Usually, when no information is available, α is fixed to n (number of samples) since it is interpreted as adding prior information equivalent to just one observation.Thus, we start by assuming a Zellner G-prior with constant α=212, as the number of observations we are considering.

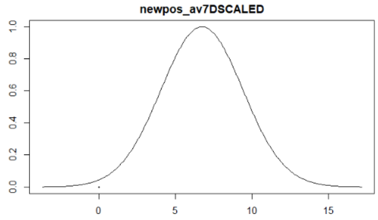
The posterior distribution, in this case, is known in closed formula and, using the BAS package to fit the model, we automatically compute it. We obtain the following posterior estimates and credible intervals of level 5%:

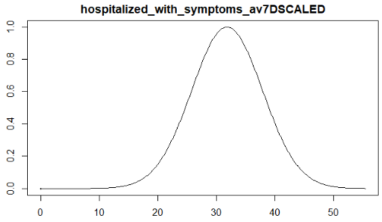


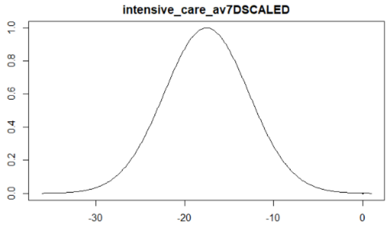
```
                                     posterior mean posterior std    2.5%  97.5%
Intercept                                    47.802         1.124  45.586 50.017
newpos_av7DSCALED                             6.726         2.678   1.447 12.006
hospitalized_with_symptoms_av7DSCALED        31.824         6.060  19.879 43.769
intensive_care_av7DSCALED                   -17.548         4.752 -26.916 -8.179
colorGialla                                   6.195         4.706  -3.083 15.472
colorArancione                               17.582         5.864   6.023 29.141
colorRossa                                   24.446         6.605  11.425 37.467
```

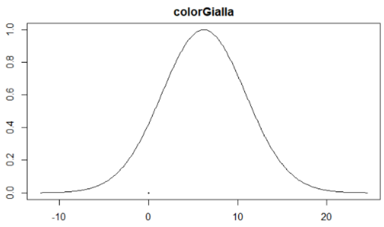
We can see from the table the posterior mean and the posterior standard deviation. As in the frequentistic case, all the posterior means are far from being zero so it is unlikely that the coefficients of these covariates are null. Looking at the credible intervals, the only variable with a coefficient that could be null is 'ColorGialla', which was not relevant even from the frequentist analysis. For the other coefficients, we are 95% confident that they will not be zero.

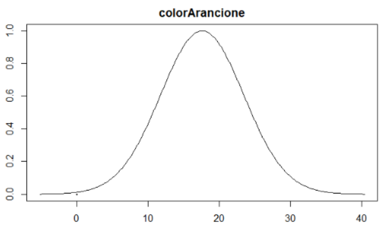
Of course, being the posterior a multi dimensional distribution we can plot just the marginal posterior relative to each covariate:

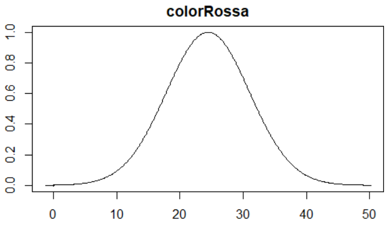






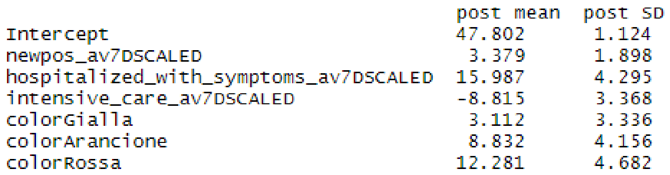






By looking at the plots we can get a graphical representation of what we noticed by looking at the previous summary: all of them except 'colorGialla' have the value '0' in the tail of the distribution thus there is a small posterior probability that the parameter under examination is null (and therefore that the feature is not relevant). These results are consistent with the p-values ('*') in the frequentistic case.

Trying to fit the same model with a smaller α (e.g. α=1), we get posterior means closer to zero (both for negative and positive coefficients). This is understandable because, with a small α we are forcing a prior centered in 0 with low variance: more informative than the previous case.



```
                                        post mean  post SD
Intercept                               47.802     1.124
newpos_av7DSCALED                        3.379     1.898
hospitalized_with_symptoms_av7DSCALED   15.987     4.295
intensive_care_av7DSCALED               -8.815     3.368
colorGialla                              3.112     3.336
colorArancione                           8.832     4.156
colorRossa                              12.281     4.682
```

Using a bigger α instead (e.g.α=400) we are considering a prior with larger variance(less informative) and we get results nearly equal to the first case. Having no prior information on the problem, this is a desirable result. The numerical results of this case can be found in Appendix A.2

2.3.2 Zellner-Siow prior (mixture of g-prior)

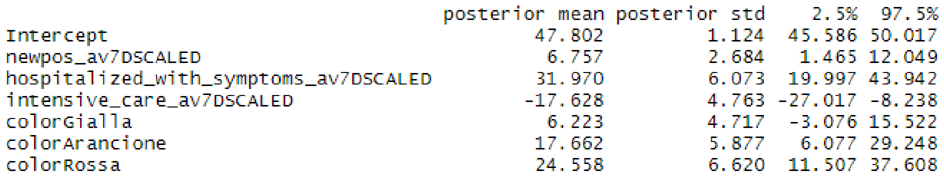
For comparison, we use now another prior, the Zellner-Siow prior that is simply a mixture of Zellner priors:

$$\beta = (\beta_1, \dots, \beta_k)$$
$$\beta|\sigma^2, X \sim \mathcal{N}_k(0, \alpha\sigma^2(X^t X)^{-1})$$
$$\sigma^2|X \sim \pi(\sigma^2) = \sigma^{-2}$$
$$1/\alpha \sim \pi_0 = Gamma(1/2, n/2)$$

In this case we have a latent parameter, 1/α, that is not directly present in the likelihood but it is used to define the prior for the parameter β. It is sampled from a Gamma(0.5,n*0.5) where n is the number of observations. We have no additional hyper parameters.

In this case, the posterior is not known in closed form. Using the BAS package to fit the model, the function 'bas.lm' is not performing just an analytical computation like in the previous paragraph, in this case it performs a sampling procedure. Within this function, it is possible to set which sampling methods to use. Here we use the default one (method= "BAS") that is "Bayesian Adaptive Sampling" algorithm[4].

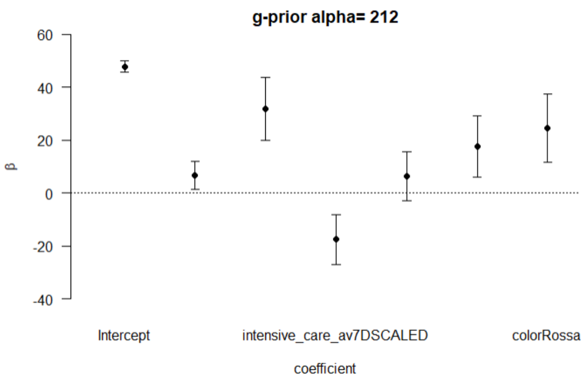
The results obtained are substantially equal to the ones obtained using the Zellner's informative prior. Here the posterior estimates and credible intervals:

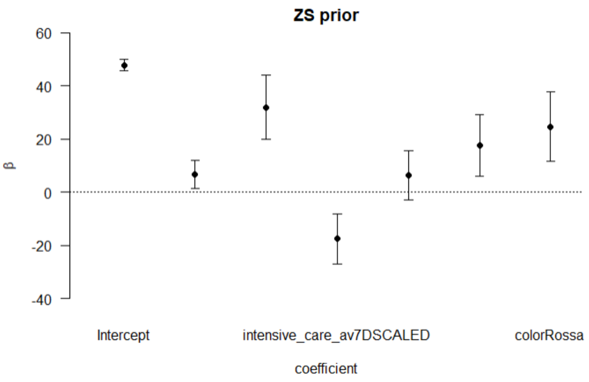


```
                                      posterior mean posterior std    2.5%  97.5%
Intercept                                     47.802         1.124  45.586 50.017
newpos_av7DSCALED                              6.757         2.684   1.465 12.049
hospitalized_with_symptoms_av7DSCALED         31.970         6.073  19.997 43.942
intensive_care_av7DSCALED                    -17.628         4.763 -27.017 -8.238
colorGialla                                    6.223         4.717  -3.076 15.522
colorArancione                                17.662         5.877   6.077 29.248
colorRossa                                    24.558         6.620  11.507 37.608
```

We can see the similarity between the coefficients estimates obtained under the two different priors also with the following plots:

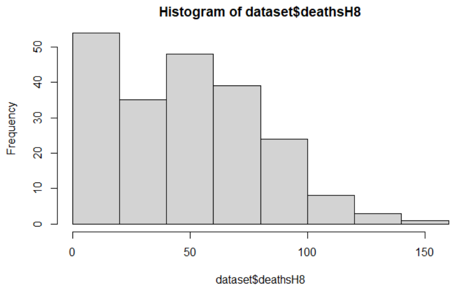
[4] *Further details on the Bayesian Adaptive Sampling algorithm can be found in the R documentation (https://www.rdocumentation.org/packages/BAS/versions/1.6.2/topics/bas.lm)*





2.4 Bayesian Poisson Regression

Now we want to consider a model where the distribution of our response variable, deathsH8, is no more Gaussian. Having count data, the distributions most suited to our case are the Poisson one and the Negative Binomial one. The main difference between the two is that the Poisson distribution assumes that the mean and the variance are equal, while the Negative Binomial one relaxes this assumption.



We model the response variable with a simple Poisson random variable, where the Y_i are no more i.i.d but they are still independent. The parameter λ depend, in terms of a linear combination, on the covariates X_i and their coefficients.

$$Y_i|X_i \sim Pois\,(k|\lambda(X_i))$$

$$log(\lambda(X_i)) = \alpha + \sum_{j=1}^{p} \beta_j X_{ij}$$

We consider the logarithm of the parameter because, as it is defined, it could be both positive and negative values (for example if all the βs happen to be negative and $\alpha = 0$) and the Poisson function can not have a negative parameter. Therefore, the log operates as a link function that always returns a positive number.

We need to choose prior distributions for the βs and α. We choose to use a simple Normal distribution for α (non-informative,with large variance) and a hierarchical distribution for β:

$$\alpha \sim N(0, 100)$$

$$\beta_j \sim N(0, 1/s_j)$$

$$s_j \sim Gamma(0.01, 0.01)$$

$$(i = 1... 6 = indices\ of\ the\ covariates)$$

Having no prior information on the problem, we chose these prior distributions because of their simplicity and the hyper-parameters in order to make them as uninformative as possible.

The covariates are 'enumerated' as follows:

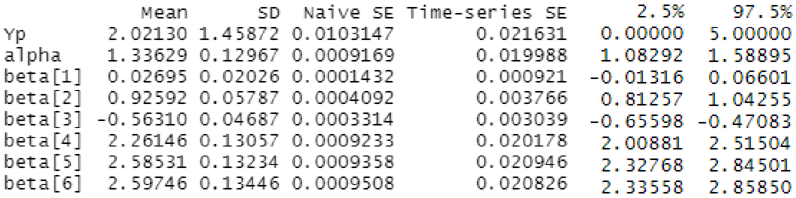
X_1=*newpos_av7DSCALED* X_2=*hospitalized_with_symptoms_av7DSCALED*

X_3=*intensive_care_av7DSCALED* X_4=*colorGialla*

X_5=*colorArancione* X_6=*colorRossa*

We fit the model using JAGS[5]. In particular, we trained the model with the first 211 rows (all except the last one) and kept the last row of our dataset, the 212-th row, to do a prediction exercise (results and details in the following sub-paragraph, 2.4.1).

We sample the MCMC using 3000 burn-in steps and 100000 steps for the sampling of each chain, saving all the βs, α and Yp (the latter is used in the prediction exercise). We obtain the following posterior statistics and credible intervals at level 5%.



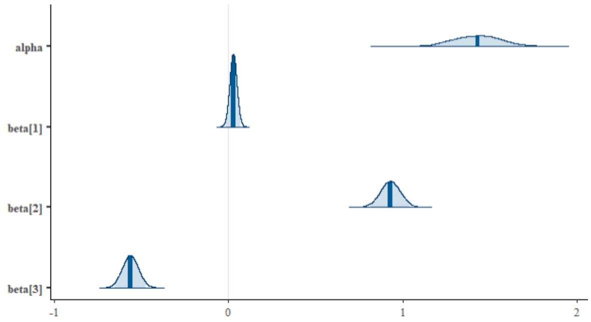
```
              Mean      SD  Naive SE Time-series SE      2.5%    97.5%
Yp         2.02130 1.45872 0.0103147       0.021631  0.00000  5.00000
alpha      1.33629 0.12967 0.0009169       0.019988  1.08292  1.58895
beta[1]    0.02695 0.02026 0.0001432       0.000921 -0.01316  0.06601
beta[2]    0.92592 0.05787 0.0004092       0.003766  0.81257  1.04255
beta[3]   -0.56310 0.04687 0.0003314       0.003039 -0.65598 -0.47083
beta[4]    2.26146 0.13057 0.0009233       0.020178  2.00881  2.51504
beta[5]    2.58531 0.13234 0.0009358       0.020946  2.32768  2.84501
beta[6]    2.59746 0.13446 0.0009508       0.020826  2.33558  2.85850
```

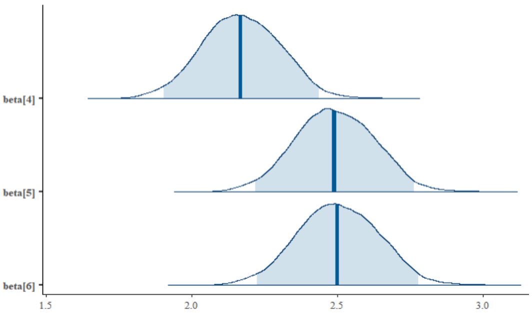
These results are quite different from the ones previously obtained. First, we see that the magnitude of these elements is very low compared to the previous ones but this is to be expected because we are using the logarithm of lambda and not λ itself. In other words, in the previous models the term $\alpha + \sum_{j=1}^{p} \beta_j X_{ij}$ was the mean of the distribution of y, now that term is the logarithm of the mean.

Another thing we can notice is that now 'newpos_av7D' is the only parameter that has the value '0' in the credible interval of level 5%, instead 'colorGialla' is far from it. We can notice that even in previous models 'newpos_av7D' was the second coefficient with the mean closest to zero.

This model is certainly more suitable for our problem than using a Gaussian Linear Regression. In fact, in general, using a continuous distribution to model count data takes risks: it is quite likely that the regression model would produce decimal or even negative predicted values(especially if we have observations ~ 0), which are theoretically impossible with count data (f.i. in our case, the number of deaths can not be negative).

We plot below the marginal posterior densities with the 95% HPD interval highlighted. Other plots relative to this posterior density can be found in appendix A.3





It's clear we created a MC using JAGS to sample from the posterior, thus the result can not be smooth: these plots are just 'smoothed' histograms (smoothed automatically by JAGS). These graphs are consistent with the considerations made above, based on the summary.

Comparing our results with the ML estimates we get from the function *'glm'*, they are very similar. This is understandable as we are using completely uninformative priors. For the numerical results of 'glm' please refer to Appendix A.4.

We can also try to use another prior and compare the new results with the previous ones. The new chosen prior is:

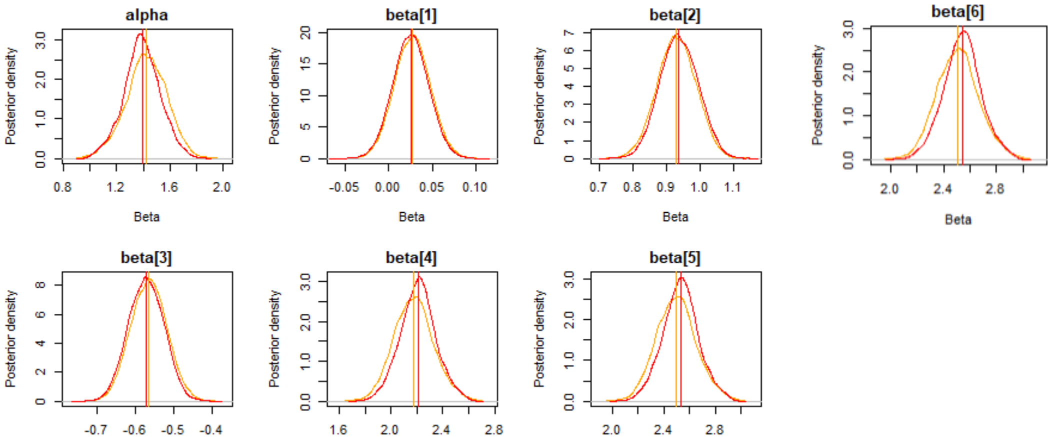
$$\alpha \sim N(0, 100)$$
$$\beta_j \sim N(m, 100)$$
$$m \sim N(0, 100)$$

[5] *For convenience, to work with the categorical variable 'color' we built additional dummy columns. This is because, having to provide in input to the model a matrix of covariates, it was simpler to create additional columns than using the R data object 'factor' (used in the previous paragraphs).*

The prior for β is still hierarchical and all the β_i will have a common mean that has a N(0,100) prior. Using a hierarchical prior is a good practice to have some dependence between the parameters.
We also fit this model using JAGS and keeping the same number of steps for sampling and 'burn-in' phase.

The posterior statistics and plots are very similar to the ones obtained using the previous prior. Therefore we omitted them in this paragraph but they are available in the Appendix A.5.

Here we compare graphically the posterior marginal distributions obtained with the two different priors.



We can see that the plots using the two models are very similar and, as we have already noticed by the summaries of the model the only parameter that could be null is β_1 associated with the covariate 'newpos_av7D' (that was the variable closest to '0' also in the previously seen models).

2.4.1 Prediction exercise

As already anticipated, we kept the last row to do a prediction exercise: we want to predict the value of the variable deathsH8 on the last day considered by our dataset using the covariates associated to that row and the model previously fitted (using the data of the other 211 days).

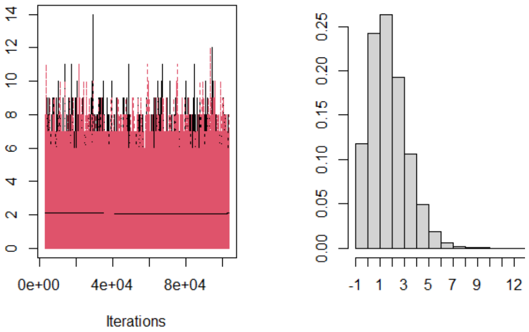
Thus, we defined the variable to predict Y_p that differently from the other Y_i is not observed and the variable X_p containing the values of the covariates of the last row of the dataset.

$$Y_p|X_p \sim Pois\,(k|\lambda(X_p))$$

$$log(\lambda(X_p)) = \alpha + \sum_{j=1}^{p} \beta_j X_{pj}$$

The results obtained from the sampling are the following:

```
          Mean      SD  Naive SE Time-series SE
Yp     2.02130 1.45872 0.0103147       0.021631
```



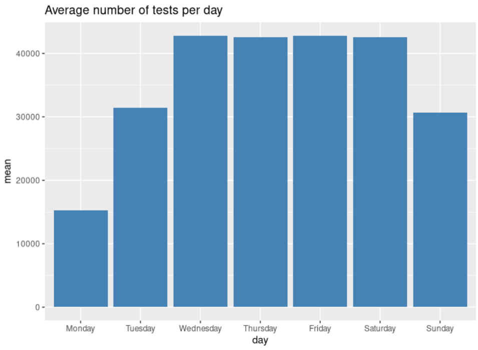
The mean is around 2 with a standard deviation of ~1.5. This seems plausible since in the last days of our dataset the values of 'deathsH8' have been in the range [0,5] and the real value of that variable on that day is '1'. We can conclude that the prediction exercise has an acceptable result.

In this case we see a proper histogram and not a smoothed one because JAGS recognizes Y_p as a discrete random variable and thus it does not smooth the histogram.

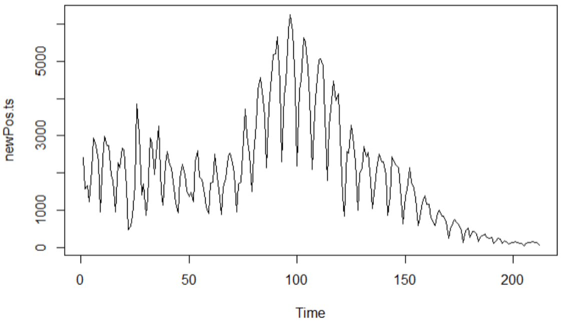
3. Time series model for `newpos_av7D`

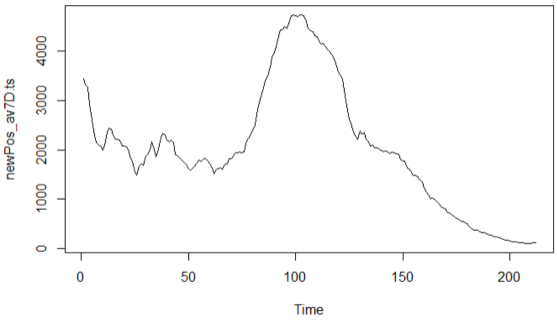
In this section, we focus on the analysis of the number of new positive cases using a time series model. We want to use an AR(1), a linear model that predicts the present value of a time series using the immediately prior value in time:

$$y_{t+1} = \mu + \alpha y_t + \epsilon_t \qquad \epsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$$



In particular, we don't consider as y the number of 'new_positives' but the variable 'newpos_av7D' to take into account the 'seasonality' of the number of positives. As we can see from the graph on the right and below there are weekdays that have a lower number of new positives (probably due to the lower amount of tests performed). Using 'newpos_av7D', we are considering an average of the 7 days before the day considered thus the seasonality is mitigated.





We assume the following prior because of its simplicity and the fact that is non-informative:

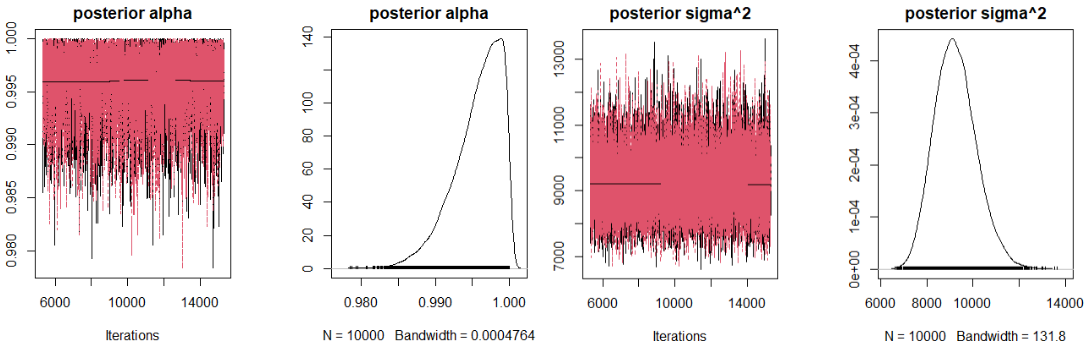
$$\alpha \sim U(1, 1)$$

$$\mu \sim N(0,\ 1000)$$

$$\tau = \frac{1}{\sigma^2} \sim G(0.001,\ 0.001)$$

and we write the model in JAGS. We sample the MCMC using 5000 burn-in steps and 10000 steps for the sampling for each chain, keeping track of the parameters α, μ, and τ.

Now we can plot the posterior density and trace plots of α and σ^2:

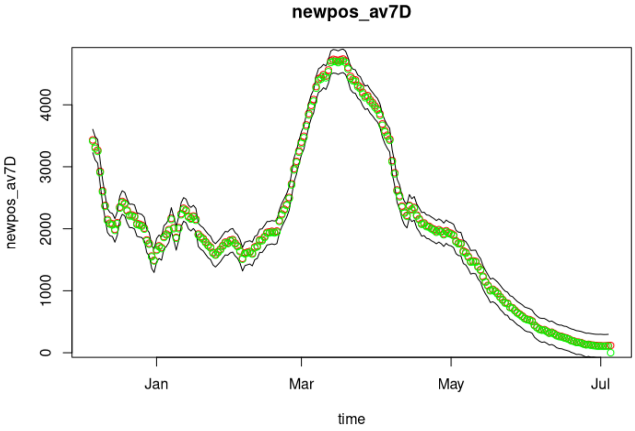


The posterior parameter trace plots show the evolution of the sampled parameters values over the iterations of the two markov chains.

The posterior alpha trace plot shows that the chains get stuck for high values of alpha which means that we have some problems for high values of alpha. With lower values of alpha, we have sufficient states changes as the MCMC algorithm runs

The posterior sigma^2 trace plot displays a stable and horizontal band indicating convergence of the MCMC algorithm in distribution. We observe that our two chains converge at the same point and seem to explore the same region of parameter values which is a good sign.

Both the posterior distributions indicate that the parameters will be null with a very low probability because the ‘0’ value is in the tail of the distribution. A model of this type obviously cannot compete with a model in which there are some covariates as the previous ones: the amount of information provided by regressors is an important factor for a good model of the problem.
Below we plot the true data (red), the sample estimates (green) and the corresponding confidence intervals.



We can see that the true data are very close to the samples estimated by the model: they are always inside the confidence interval. The model slightly underestimates but the difference is very low.

For autocorrelation plots of α and σ^2 please refer to the appendix A.6.

4. ARMAX model for ‘intensive_care_H8’

In this section we want to use an extended version of a normal AR(1) process: to predict the present value of a time series we use not only the immediately prior value in time but also other covariates. In other words, our dependent variable is modeled in terms of a linear combination of independent variables, its past values and disturbances.

The model we want to use is known as ARMAX (or ARIMAX):

$$y_{t+1} = \mu + \sum_{j=1}^{m} \beta_j X_t^j + \alpha y_t + \epsilon_t \qquad \epsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

where m is the number of used covariates, X_t^j is the value of the covariate j for the observation t. Each covariate has a coefficient β_j and then we keep an intercept term μ.

The prior we use is the same as the previous paragraph with the addition of only a prior (non informative) distribution for the β parameters:

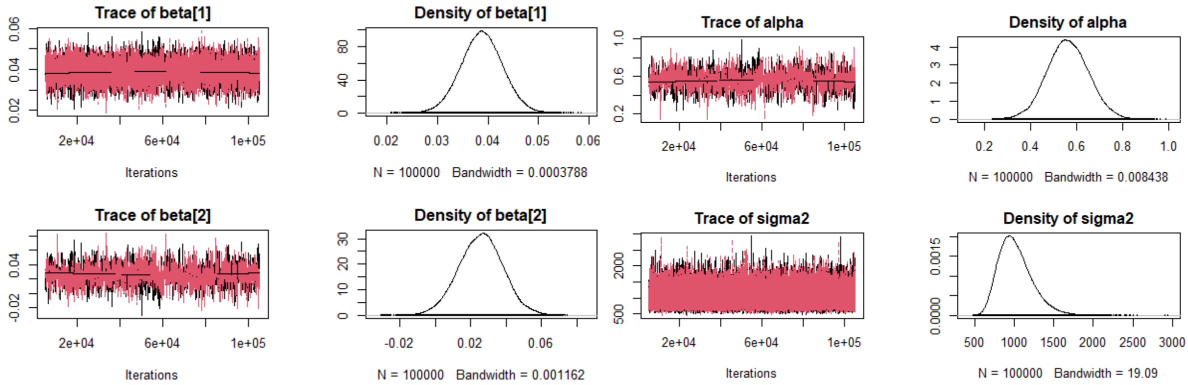
$$\alpha \sim U(1, 1)$$
$$\mu \sim N(0,\ 1000)$$
$$\tau = \frac{1}{\sigma^2} \sim G(0.001,\ 0.001)$$
$$\beta \sim N(0, 1000)$$

4.1 The ‘weekly’ dataset

The dataset used in this paragraph is different from the one previously used. To avoid the effect of the ‘seasonality’ of the data we are going to use weekly averaged data. In particular we are going to use the dataset ‘weekly.rds’ already presented in paragraph 1. In this way, the period of time between y_{t+1} and y_t is of one week. In particular we are going to use only 3 variables: ‘intensive_care_week’ as our response variable, “hospitalized_with_symptoms_week” (X^1) and “new_positives_week” (X^2) as covariates. Thus, the coefficients for these two covariates are β_1 and β_2.

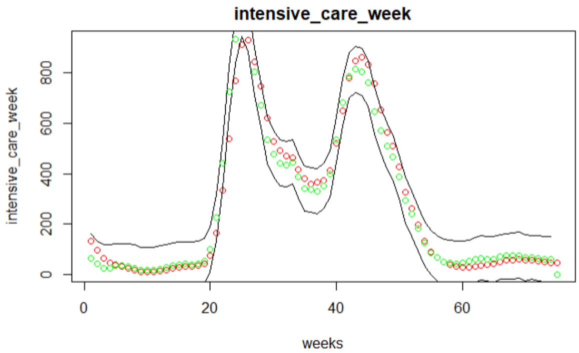
4.2 Posterior analysis and interpretation of results

We write the model in JAGS. We sample the MCMC using 5000 burn-in steps and 10000 steps for the sampling, keeping track of the parameter α, μ, τ, β*s*. We obtain the posterior distributions:



We can see that the trace plots do not present any periodicity or 'flat part'. The posterior distributions have the value '0' in the tail except for β_2 . Thus, for the other parameters, there is a small posterior probability that they are null.

Below we plot the true data (red), the sample estimates (green) and the corresponding confidence intervals



We can see that the true data are very close to the samples estimated by the model: they are almost always inside the confidence interval. The model overestimates and underestimates the true data in different time periods, however the difference is within reasonable limits.

5. Conclusions

This project was an opportunity to put in place the theoretical knowledge learned during the course. Starting from a single dataset, we modeled different variables with different types of models. We started by estimating the number of deaths, first approximating it with a normal model and then using a discrete one. Then we analyzed a time-series model for the number of positives. Finally, we combined the idea of regression with that of autoregressive models, modeling the number of patients entering the IC department with an ARMAX process.

We have seen that when using noninformative Prior the frequentistic model gives results similar to the Bayesian one. We have also seen that the AR(1) model, despite carrying way less information on the problem than a regression with other covariates, in our case produced reasonable estimates.

We could also see in several situations that using different priors led to the same results if you kept their non-informativeness.