

# **BAYESIAN LOGISTIC REGRESSION FOR BREAST CANCER PREDICTION**

**BEENAPREET KAUR**

**ISYE 6420  
BAYESIAN PROJECT**

2018

**SPRING 2018**

# INDEX

## 1. Summary

- Project Description

## 2. Bayesian Logistic Regression

- Why use Bayesian? Why use frequentist
- Have evaluated both in the project

## 3. Exploratory Data Analysis

## 4. Model

- Model OpenBugs
- Model PYMC
- Model Selection

## 5. Results

- Comparison of Classical and Bayesian Approach
- Interpreting Coefficients
- Confidence Intervals and Credible sets

## 1. SUMMARY

The breast cancer dataset from UCI Machine learning repository consists of Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. These features describe the characteristics of the cell nuclei present in the digitized image. In this project I have attempted to use a Bayesian Logistic Regression model to understand what characteristics of a cell nuclei are important in deciding whether a cell is malignant.

## 2. BAYESIAN LOGISTIC REGRESSION

- In the Frequentist approach the data is said to be a random variable while the parameters are fixed, and we model how likely we would get the data given a particular set of the parameters
- In the Bayesian approach however, we assume that we have already observed the data and now we are trying to update our belief about the underlying parameters using the data, hence the Bayesian approach is more intuitive

To estimate the parameters ( $\theta$ ) by Bayesian Methods we need to model the conditional distribution for unobserved quantities given the data which is called Posterior Distribution

$$P(\theta|x) = p(\theta) * \frac{p(x|\theta)}{\int p(\theta) * p(x|\theta)} \propto p(\theta) * p(x|\theta)$$

The **prior** distribution ( $p(\theta)$ ) expresses our uncertainty about  $\theta$  before seeing the data and The **posterior** distribution ( $P(\theta|x)$ ) expresses our uncertainty about  $\theta$  after seeing the data

The posterior, which is our updated belief about the weights given evidence, is proportional to our prior (initial belief) times the likelihood. We can't evaluate the closed form posterior, but can approximate it by sampling or variational methods. This gives us a distribution over the weights.

This is the basic principal behind Bayesian Logistic Regression.

## 3. EXPLORATORY DATA ANALYSIS

- After examining the correlations we can see that there is a strong correlation of cell size, cell shape and chromatin with malignant cells.

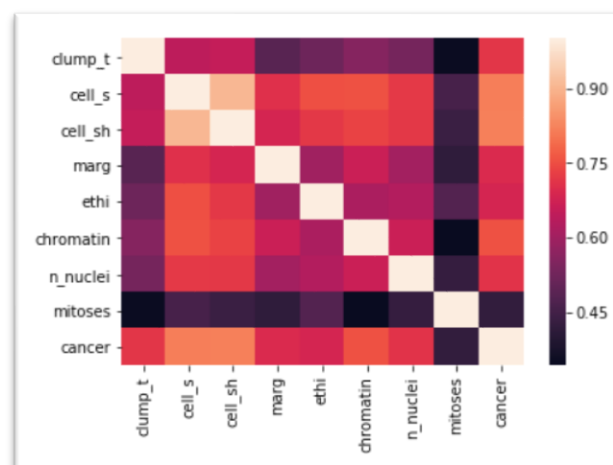


Figure 1 : Shows the Correlations between different variables in the data

To confirm the correlations we further examine the box plot graphs below.

b. Below I have examined the relationship of each variable with whether or not the cells are malignant(1)

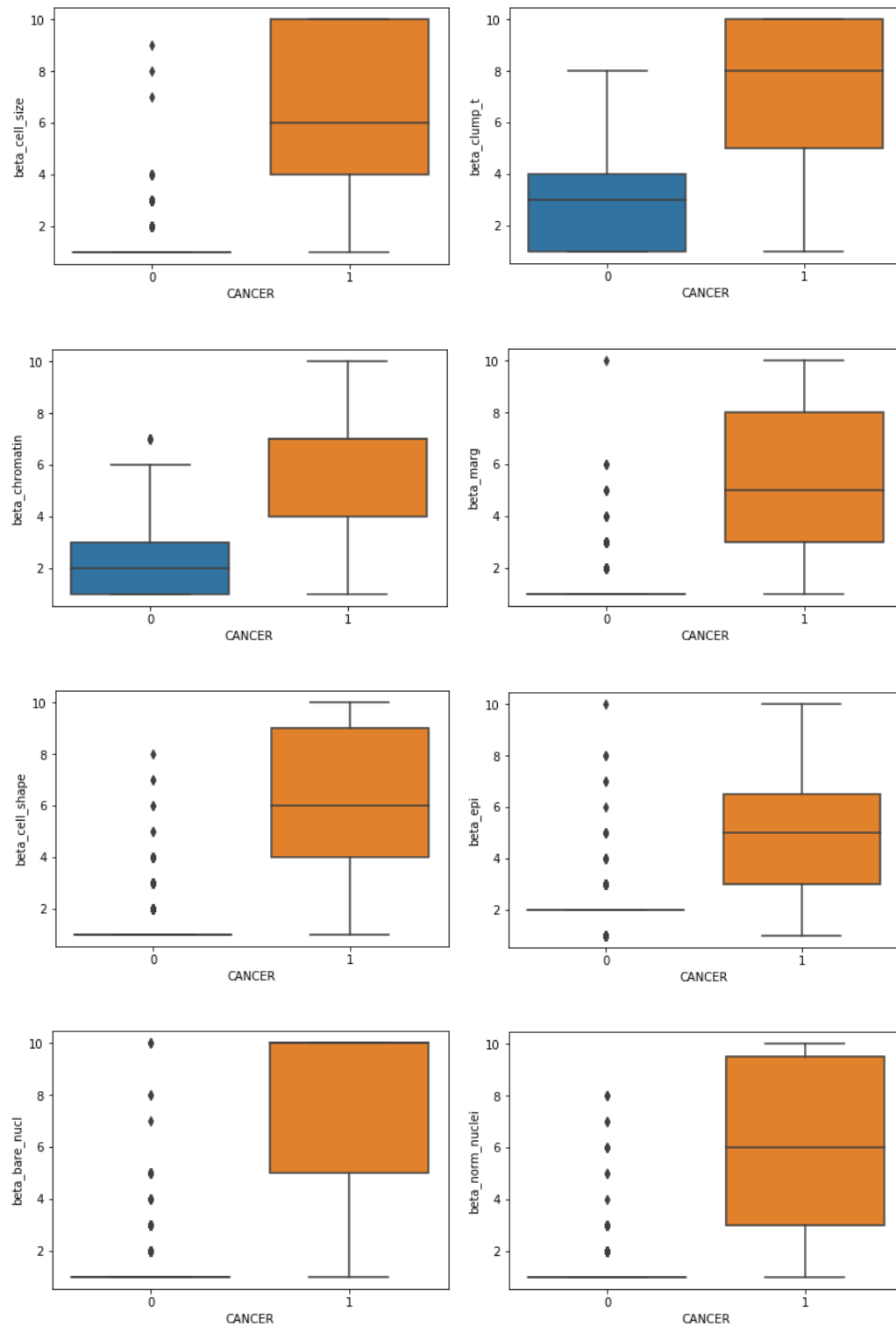


Figure 2: Boxplots showing variation of variables with whether or not the cell is malignant

As can be seen from the above charts, almost all of the variables have a significant contribution to the cell being malignant. There is a **strong correlation between the coefficient for bland chromatin and the probability of the cell being malignant**. There is also a strong indication that **higher the clump thickness of the cell, higher is the probability of it being malignant**.

## 4. MODEL

### 4.1 Open Bugs

The below model with Logit link function was used to fit the Data

```
model{
  for(i in 1:683)
  {

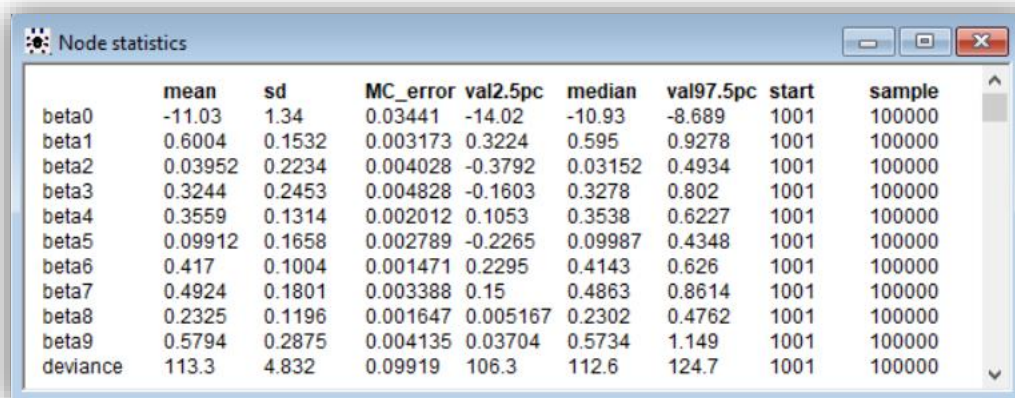
    y[i]~dbern(p[i])
    logit(p[i])<-
    beta0+beta1*x1[i]+beta2*x2[i]+beta3*x3[i]+beta4*x4[i]+beta5*x5[i]+beta6*x6[i]+beta7*x7[i]+beta8*x8[i]+
    beta9*x9[i]

  }

  beta1~dnorm(0,0.001)
  beta2~dnorm(0,0.001)
  beta0~dnorm(0,0.001)
  beta3~dnorm(0,0.001)
  beta4~dnorm(0,0.001)
  beta5~dnorm(0,0.001)
  beta6~dnorm(0,0.001)
  beta7~dnorm(0,0.001)
  beta8~dnorm(0,0.001)
  beta9~dnorm(0,0.001)

}
```

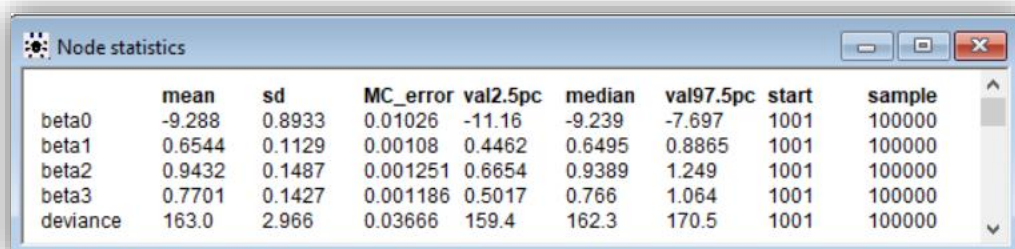
Figure 3: Logistic Regression - Full Model



A screenshot of a 'Node statistics' window from a statistical software package. The window displays a table with 9 columns: node name, mean, sd, MC\_error, val2.5pc, median, val97.5pc, start, and sample. The rows represent the model parameters beta0 through beta9, and the deviance. All parameters have a sample size of 100,000.

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta0	-11.03	1.34	0.03441	-14.02	-10.93	-8.689	1001	100000
beta1	0.6004	0.1532	0.003173	0.3224	0.595	0.9278	1001	100000
beta2	0.03952	0.2234	0.004028	-0.3792	0.03152	0.4934	1001	100000
beta3	0.3244	0.2453	0.004828	-0.1603	0.3278	0.802	1001	100000
beta4	0.3559	0.1314	0.002012	0.1053	0.3538	0.6227	1001	100000
beta5	0.09912	0.1658	0.002789	-0.2265	0.09987	0.4348	1001	100000
beta6	0.417	0.1004	0.001471	0.2295	0.4143	0.626	1001	100000
beta7	0.4924	0.1801	0.003388	0.15	0.4863	0.8614	1001	100000
beta8	0.2325	0.1196	0.001647	0.005167	0.2302	0.4762	1001	100000
beta9	0.5794	0.2875	0.004135	0.03704	0.5734	1.149	1001	100000
deviance	113.3	4.832	0.09919	106.3	112.6	124.7	1001	100000

Figure 4: Logistic Regression - Full Model Results



A screenshot of a 'Node statistics' window for a specific model (Model 1). The window displays a table with 9 columns: node name, mean, sd, MC\_error, val2.5pc, median, val97.5pc, start, and sample. The rows represent the model parameters beta0 through beta4, and the deviance. All parameters have a sample size of 100,000.

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta0	-9.288	0.8933	0.01026	-11.16	-9.239	-7.697	1001	100000
beta1	0.6544	0.1129	0.00108	0.4462	0.6495	0.8865	1001	100000
beta2	0.9432	0.1487	0.001251	0.6654	0.9389	1.249	1001	100000
beta3	0.7701	0.1427	0.001186	0.5017	0.766	1.064	1001	100000
deviance	163.0	2.966	0.03666	159.4	162.3	170.5	1001	100000

Figure 5: Logistic Regression - Model 1 (beta0 + beta\_clump\_t\*x1 + beta\_cell\_size\*x2 + beta\_cell\_shape\*x3 + beta\_chromatin\*x4) results

## 4.2 PYMC (Python)

In PYMC first the hyperparameters are defined, as in the case of openbugs, I used weak Gaussian priors for all the parameters.

```
### hyperpriors
tau = pm.Gamma('tau', 1.e-3, 1.e-3, value=10.)
sigma = pm.Lambda('sigma', lambda tau=tau: tau**-.5)

### parameters
# fixed effects
beta0 = pm.Normal('beta0', 0., 1e-6, value=0.)
beta_clump_t = pm.Normal('beta_clump_t', 0., 1e-6, value=0.)
beta_cell_size = pm.Normal('beta_cell_size', 0., 1e-6, value=0.)
beta_cell_shape = pm.Normal('beta_cell_shape', 0., 1e-6, value=0.)
beta_marg = pm.Normal('beta_marg', 0., 1e-6, value=0.)
beta_epi = pm.Normal('beta_epi', 0., 1e-6, value=0.)
beta_bare_nucl = pm.Normal('beta_bare_nucl', 0., 1e-6, value=0.)
beta_chromatin = pm.Normal('beta_chromatin', 0., 1e-6, value=0.)
beta_norm_nuclei = pm.Normal('beta_norm_nuclei', 0., 1e-6, value=0.)
beta_mito = pm.Normal('beta_mito', 0., 1e-6, value=0.)
```

Figure 6: Hyperparameter initialization in PYMC

For y values it is important to declare that this value is observed - and our known quantities. After this we run the Markov Chain Monte Carlo simulation with 100,000 iterations, with the initial 1000 values burned so that we consider values only once they've reached steady state

```
logit_p = (beta0 + beta_clump_t*x1 + beta_cell_size*x2 + beta_cell_shape*x3 +
          beta_marg*x4 + beta_epi*x5 + beta_bare_nucl*x6 + beta_chromatin*x7 +
          beta_norm_nuclei*x8 + beta_mito*x9)
@pm.observd
def y(logit_p=logit_p, value=df[11]):
    return pm.bernoulli_like(df[11], pm.invlogit(logit_p))

m = pm.MCMC([beta0, beta_clump_t, beta_cell_size, beta_cell_shape, beta_marg,
            beta_epi, beta_bare_nucl, beta_chromatin, beta_norm_nuclei, beta_mito,
            tau, sigma, logit_p, y], calc_deviance=True)
a=time.time()
m.sample(100000, 1000)
b=time.time()
Time_full_model=b-a
D1=m.deviance
```

Figure 7: Model Definition and sampling

```
beta0 -11.1139002299 1.30551310309
beta_clump_t 0.608049412978 0.152847439241
beta_cell_size 0.0356127568966 0.230011551645
beta_cell_shape 0.325855446744 0.244623048796
beta_marg 0.361259032265 0.132041511104
beta_epi 0.0999496924643 0.169762669733
beta_bare_nucl 0.41383829328 0.100548735751
beta_chromatin 0.502110820548 0.178805835974
beta_norm_nuclei 0.231967987396 0.119031331185
beta_mito 0.58524691275 0.290418035772
```

Figure 8: Results of the Full Model obtained from PYMC

### 4.3 Model Selection

To compare the full model with other models, I calculated the deviance and BIC for each of the models mentioned in table .The Bayesian Information Criterion (BIC) for a model m with number of predictors  $p_m$  and sample size n is defined as:

$$BIC_m = D_m + (p_m + 1) * \log(n)$$

$\Delta BIC$	Evidence against higher BIC
0 to 2	Not worth more than a bare mention
2 to 6	Positive
6 to 10	Strong
>10	Very Strong

*Table 1: Guideline for model selection using BIC*

Model	Deviance	BIC
<b>Full Model</b> (all 9 variables and intercept)	<b>113.16</b>	<b>184.87</b>
<b>Model 1</b> (beta0 + beta_clump_t*x1 + beta_cell_size*x2 + beta_cell_shape*x3 + beta_chromatin*x4)	<b>165</b>	<b>191.10</b>
<b>Model 2</b> (beta0 + beta_clump_t*x1 + beta_cell_size*x2 + beta_cell_shape*x3)	<b>205.97</b>	<b>232.08</b>
<b>Model 3</b> (beta0 + beta_clump_t*x1 + beta_cell_size*x2 )	<b>337.85</b>	<b>357.43</b>
<b>Model 4</b> (beta0 + beta_clump_t*x1)	<b>1122.58</b>	<b>1135.63</b>
<b>Model 5</b> (beta0 + beta_cell_size*x2 )	<b>1133.17</b>	<b>1146.22</b>

*Table 2: Summary of Deviance and BIC values for different models*

From the above table results we can see that Full Model and Model 1 are the best models since they have the lowest BIC values Since  $\Delta BIC = BIC_{(Full\ Model)} - BIC_{(Model\ 1)} = 6.23 \Rightarrow$  from table we can say that we have **strong** evidence to choose the Full Model over Model 1.

## 5. RESULTS

### a) Time Comparison

After selecting the best model I ran a Logistic regression using the classical approach and the Bayesian approach, in both OpenBugs and PYMC, as shown below, from Table 3 results we can see that running the model on OpenBugs takes almost half the time as it does in PYMC.

Model	Time
OpenBugs	247s
PYMC	512s

Table 3: Time taken to run the full model in Openbugs and PYMC

Coefficient	GLM_coeff	GLM_stddev	Openbugs_coeff	Openbugs_stddev	PYMC_coeff	PYMC_stddev
Intercept	-10.1039	1.175	-11.03	1.34	-11.11	1.31
beta_clump_t	0.535	0.142	0.6004	0.1532	0.61	0.15
beta_cell_size	-0.0063	0.209	0.03952	0.2234	0.04	0.23
beta_cell_shape	0.3227	0.231	0.3244	0.2453	0.33	0.24
beta_marg	0.3306	0.123	0.3559	0.1314	0.36	0.13
beta_epi	0.0966	0.157	0.09912	0.1658	0.10	0.17
beta_bare_nucl	0.383	0.094	0.417	0.1004	0.41	0.10
beta_chromatin	0.4472	0.171	0.4924	0.1801	0.50	0.18
beta_norm_nuclei	0.213	0.113	0.2325	0.1196	0.23	0.12
beta_mito	0.5348	0.329	0.5794	0.2875	0.59	0.29

Table 4: Comparison of coefficient values across Classical Logistic Regression and Bayesian Logistic Regression

### b) Comparison of Classical and Bayesian Approach

From Table 4, it can be observed that even though the size of a cell is positively correlated with malignant cells – in the GLM logistic regression the size of cell has a weak but negative coefficient whereas in the Bayesian approach because of our prior reinforcement, the cell size still shows a positive correlation with malignant tumors. In the Classical approach as can be seen from the correlation diagram in Figure 1, there is a strong correlation between cell size and cell shape, hence multicollinearity may have caused the sign to flip.

Coefficient	GLM_coeff	C1_25%	C1_97.5%	Openbugs_coeff	HPD_25%	HPD_97.5%
Intercept	-10.1039	-12.407	-7.801	-11.03	0.03441	-10.93
beta_clump_t	0.535	0.257	0.813	0.6004	0.003173	0.595
beta_cell_size	-0.0063	-0.416	0.404	0.03952	0.004028	0.03152
beta_cell_shape	0.3227	-0.129	0.775	0.3244	0.004828	0.3278
beta_marg	0.3306	0.089	0.573	0.3559	0.002012	0.3538
beta_epi	0.0966	-0.21	0.404	0.09912	0.002789	0.09987
beta_bare_nucl	0.383	0.199	0.567	0.417	0.001471	0.4143
beta_chromatin	0.4472	0.111	0.783	0.4924	0.003388	0.4863
beta_norm_nuclei	0.213	-0.008	0.434	0.2325	0.001647	0.2302
beta_mito	0.5348	-0.11	1.179	0.5794	0.004135	0.5734

Table 5: Highest Posterior Density interval of Bayesian Model and Confidence Interval of Classical Model



c) Odds Ratio Interpretation

The largest Coefficients are for the cell clump thickness (beta\_clump\_t), chromatin and mitosis

⇒ Increasing cell clump thickness by one unit increases the odds of the cell being malignant by  $(e^{0.6} - 1) * 100\% = 82.2\%$

⇒ Similarly increasing bland chromatin texture by one unit increases odds of the cell being malignant by

63.2 % and by increasing mitosis by 1 unit the odds increase by 78.4%

d) Credible Set and CI interpretation (GLM and OpenBugs)

In the Frequentist approach, a 95% CI is the interval that will contain the true value on 95% of the occasions, if a study were repeated many times using samples from the same population.

For clump thickness the 95% Confidence Interval is [0.257, 0.813], this implies that on repeated sampling, the parameter estimate lies in the interval 95% of the times.

In the Bayesian Approach, a 95% credible set implies that Given the data and the model, there is a 95% chance the true values of the estimate lie in that interval.

For Clump Thickness a 95% Credible set of [0.003173, 0.595], implies that there is a 95% chance that the true values of the estimate lie in this interval. The Bayesian way of establishing credibility in the estimate is definitely more reliable however, in my model the credible interval is very large (the lower bound seems off) hence it is worthwhile to play around more with the priors of the Bayesian estimates to establish a more credible - Credible Interval

## 6. APPENDIX

### - Attributes Description :

Clump thickness: Benign cells tend to be grouped in monolayers, while cancerous cells are often grouped in multilayer.

Uniformity of cell size/shape: Cancer cells tend to vary in size and shape. That is why these parameters are valuable in determining whether the cells are cancerous or not.

Marginal adhesion: Normal cells tend to stick together. Cancer cells tend to loose this ability. So loss of adhesion is a sign of malignancy.

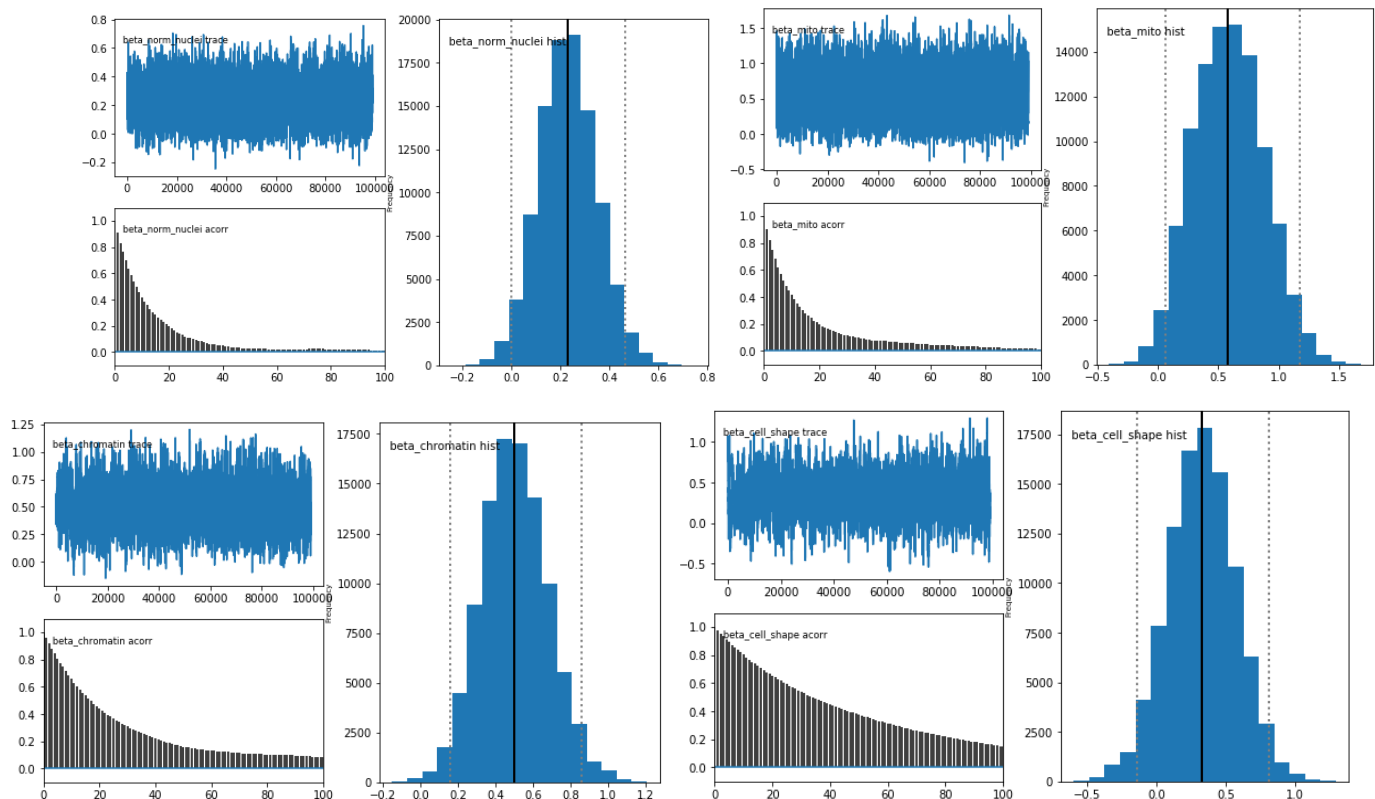
Single epithelial cell size: Is related to the uniformity mentioned above. Epithelial cells that are significantly enlarged may be a malignant cell.

Bare nuclei: This is a term used for nuclei that is not surrounded by cytoplasm (the rest of the cell). Those are typically seen in benign tumors.

Bland Chromatin: Describes a uniform "texture" of the nucleus seen in benign cells. In cancer cells the chromatin tends to be coarser.

Normal nucleoli: Nucleoli are small structures seen in the nucleus. In normal cells the nucleolus is usually very small if visible at all. In cancer cells the nucleoli become more prominent, and sometimes there are more of them.

### - PYMC Results:



## *References*

<https://stats.stackexchange.com/questions/58564/help-me-understand-bayesian-prior-and-posterior-distributions>

<https://stats.stackexchange.com/questions/163034/bayesian-logit-model-intuitive-explanation>

<http://statweb.stanford.edu/~ckirby/brad/papers/2005BayesFreqSci.pdf>

<https://stats.stackexchange.com/questions/49209/thinning-chains-in-bugs-jags>

<https://dsaber.com/2014/05/28/bayesian-regression-with-pymc-a-brief-tutorial/>

<https://stats.stackexchange.com/questions/161101/what-is-the-difference-between-logistic-regression-and-bayesian-logistic-regress>