What makes wine great?

TU Dortmund University | Applied Bayesian Data Analysis | Prof. Dr. Bürkner, Prof. Dr. Ickstadt

Yuga Hikida

2024-03-17

1 Introduction

In this report, we construct Bayesian predictive model for quality of white wines using physicochemical variables. The main aim of our analysis are to 1. understand how each variables affect the quality and 2. evaluate suitable likelihood through comparison of predictive performance of models.

2 Data

The data is obtained from (Cortez and Reis 2009). It contains the quality of red and white wines which takes values from 1, 2 up to 10 (10 is the best quality), of which we only focus on the data for wine wines for this report (4898 observations). The histogram of quality is shown in Figure 1. It can be seen that mass concentrate in 5 and 6, and none of the wine receives the quality 1, 2 nor 10.

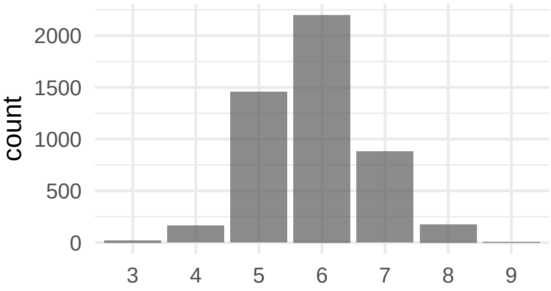


Figure 1: Histogram of response (quality)

We construct Bayesian model to predict quality of wine using following 7 physicochemical variables shown in Table 1 as predictors. We also give some interpretations of each variables for intuition.

Table 1: Description of predictive variables

Interpretations	Name of variable(s)
Acidity	citric.acid, volatile.acidity

Interpretations	Name of variable(s)
Sweetness	residual.sugar
Bitterness	sulphates
Saltiness	chlorides
Prevent oxidation and bacteria	total.sulfur.dioxide
Literally interpretable	alcohol

The histogram of predictive variables are shown in Figure 2. All the variables are continuous and the scale of predictive variables differs notably.

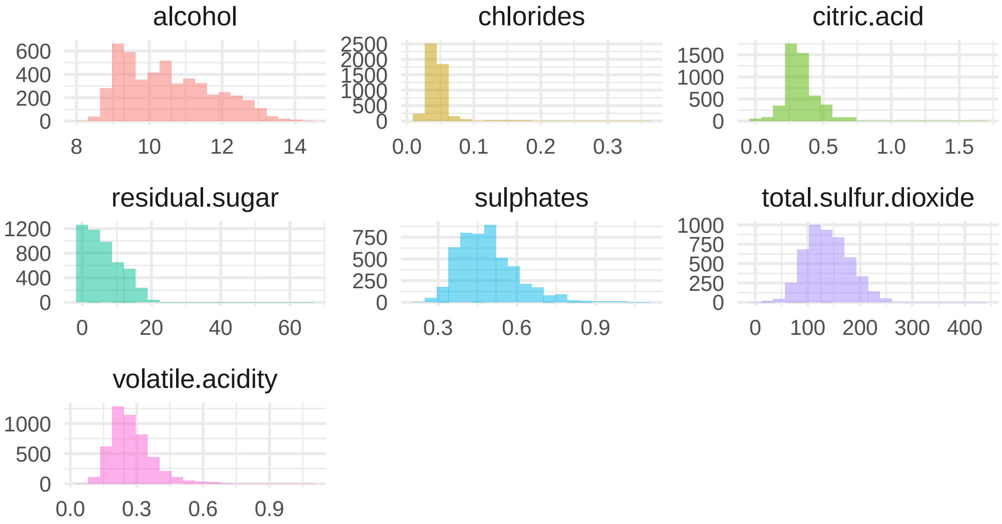


Figure 2: Histogram of predictive variables

3 Methodology

Depending on the choice of likelihood, we can model the response variable in three ways.

1. Categorical variable, $quality \in \{"1", ..., "10"\}$
2. Continuous (metric) variable, $quality \in \mathbb{R}$
3. Ordinal variable, $quality \in \{1, ..., 10\}$

The introductory paper of the data (Cortez and Reis 2009) focuses on the first approach where they built classification model using Support Vector Machine. Since we would like to retain ordered structure of data for interpretation, we use second and third approach. More precisely, we estimate Bayesian linear regression model for the second approach and Bayesian ordinal regression model for the third approach.

All the estimations are conducted using {R} package **brms** (Bürkner 2017). The package allows us to estimate Bayesian regression based model flexibly and efficiently by MCMC sampling method called Hamilton Monte Carlo using {Stan} (Carpenter et al. 2017) in backend.

Replication code for our analysis is available at https://github.com/1129hiki/abda_project.

4 Linear Regression

Firstly, we estimate linear regression model, which is arguably the simplest model when we have response and predictive variables. Hence, we estimate it as a baseline model. The model can be written as

$$\begin{aligned} y_i &\sim \text{normal}(\eta_i, \gamma) \\ \eta &= x_i^T \beta \\ \beta_j &\sim \text{normal}(0, \sigma_{\beta_j}) \\ \gamma &\sim \text{half-normal}(0, \sigma_\gamma) \end{aligned}$$

where y_i is *quality* and x_i is a vector of predictive variables for wine i. We use Gaussian likelihood even though we could choose other likelihood such as t distribution. One of our justification for the use of Gaussian likelihood is the central limit theorem which states that sum of large number of variables are approximately Gaussian. The β_j is the coefficient of linear regression for each predictive variables. Hence we have $j = 1, ..., 8$ including intercept and $\beta = [\beta_1, ..., \beta_8]^T$. The third line indicates the prior distribution for the coefficients, which we set separately for each coefficients so that we can incorporate difference in scale and/or informativeness into the priors.

Using **brms**, the model can be estimated by following code

```
f <- quality ~ citric.acid + volatile.acidity +
     residual.sugar + sulphates + chlorides +
     total.sulfur.dioxide + alcohol

linear_reg <- brm(f,
              data = d,
              family = gaussian(),
              prior = p_linear_reg,
              chains = 4,
              iter = 4000,
              warmup = 2000,
              save_pars = save_pars(all=TRUE))
```

where `f` specifies formula for linear regression, `family = gaussian()` specifies Gaussain likelihood with identity link, and `p_linear_reg` is a {R} vector containing prior specifications, which we will discuss in the next subsection. The `chains`, `iter`, and `warmup`, relating to sampling from posterior distribution, specify number of Markov chains, number of iteration per chain, and number of samples thrown away before start saving samples to avoid sampling from non-stationary part of the chain respectively. Lastly, we save all the posterior samples with `save_pars = save_pars(all=TRUE)` since we will use them for further analysis later.

4.1 Prior specification

For prior specification, we focus on the variable *alcohol*. As it can be seen in Figure 3, *alcohol* takes the value from 8% to 14% (the range is 6%), and *quality* takes value from 3 to 9 (the range is 6 as well). Following one of the principles for prior specification discussed in (Simpson et al. 2017), we assume the simplest model where none of the other variables have predictive power and hence corresponding coefficients takes zero. Then we expect absolute value of $\beta_{alcohol}$ to be smaller than 1 since we would not have $quality = 0$ nor $quality = 11$. We reflect this information with the prior $\beta_{alcohol} \sim \text{normal}(0, 0.36)$ such that density outside ± 1 is small. Note that we set the mean to be zero since we have no knowledge about the effect of *alcohol* on *quality* upon estimation of model.

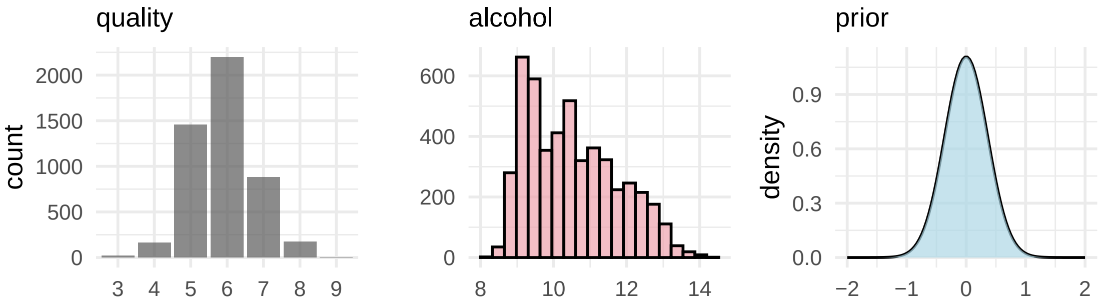


Figure 3: Distribution of (response / predictive) variables and prior distribution.

We can express prior for $\beta_{alcohol}$ as

$$\begin{aligned} \beta_{alcohol} &\sim \text{normal}(0, 0.36) \\ &:= \text{normal}(0, \alpha SD(y)/SD(\text{alcohol})) \end{aligned}$$

where $SD(\cdot)$ denotes standard deviation. We now get scale free informativeness parameter for the prior which is calculated as $\alpha \approx 0.5$. This can be used for prior specification for other coefficients such that we set same informativeness for all the coefficients. In particular, we set the priors as

$$\beta_j \sim \text{normal}(0, \alpha SD(y)/SD(x_j))$$

.

4.2 Result

For our report, we mainly focus on two predictive variables *alcohol* and *citric.acid*. The results are shown in Figure 4. The model is fitted without any convergence issue as shown by a trace plot and R-hat (see Appendix). The left column shows the posterior distribution of the coefficients corresponding to the two variables. The blue line is the median of the posterior and shadowed area indicates 50% credible interval centered around its median. It can be seen that 1% increase in *alcohol* is expected to lead around 0.35 to 0.4 unit increase in *quality*, while $1g/L$ increase in *citric.acid* is expected to lead around -0.4 to 0 unit increase (decrease) in *quality*. The right column shows posterior distribution of conditional effect of each predictive variable with other variables fixed at their mean. The shadoewd area shows 95% credible interval. It can be seen that there is strong linear relationship between *alcohol* and *critic.acid* while the effect of *citric.acid* is limited which is implied by the size of coefficient and its uncertainty.

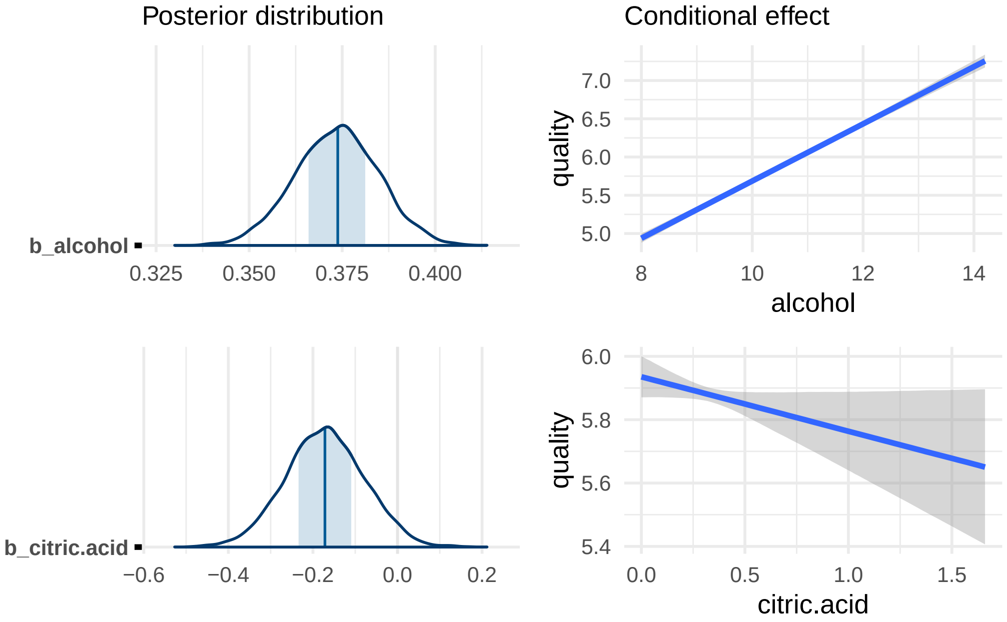


Figure 4: Result for linear regression (only for alcohol and citric.acid)

5 Ordinal Regression

Now we move on to an ordinal regression model. The discussion here will be based on (Bürkner and Vuorre 2019). In particular, we will use ordinal regression model called cumulative model. In cumulative model, we consider continuous latent variable $\tilde{y}$ which determine the quality y through thresholds τ_c. The model can be expressed as follow.

$$
\begin{aligned}
&\text{For } c = 2, .., C-1: \\
&\qquad Pr(y = c) = Pr(y \leq c) - Pr(y \leq c-1) \\
&\qquad\qquad := Pr(\tilde{y} \leq \tau_c) - Pr(\tilde{y} \leq \tau_{c-1}) \\
&\qquad\qquad \tilde{y} = \eta + \epsilon,\ \epsilon \sim \text{normal}(0, 1)
\end{aligned}
$$

In cumulative model, probabilities for each categories c is defined as probability that latent variable $\tilde{y}$ takes the value between τ_{c-1} and τ_c or $Pr(\tau_{c-1} < \tilde{y} \leq \tau_c)$. We then model $\tilde{y}$ with the linear predictor η which remains same as linear regression model, and noise term which we assume to be standard normal. This also defines the distribution of $\tilde{y}$ to be normal.

We can also express $Pr(\tilde{y} \leq \tau_c)$ as

$$
\begin{aligned}
Pr(\tilde{y} \leq \tau_c) &= Pr(\eta + \epsilon \leq \tau_c) \\
&= Pr(\epsilon \leq \tau_c - \eta) \\
&= \Phi(\tau_c - \eta)
\end{aligned}
$$

where Φ is a cumulative distribution function of standard normal also known as Probit. This lead to a simpler expression for the probability without $\tilde{y}$ as

$$
Pr(y = c) = \Phi(\tau_c - \eta) - \Phi(\tau_{c-1} - \eta)
$$

Note that for $c = 1$ and $c = C$, we have

$$
\begin{aligned}
Pr(y = 1) &= \Phi(\tau_1 - \eta) \\
Pr(y = C) &= 1 - \Phi(\tau_{C-1} - \eta)
\end{aligned}
$$

Using **brms**, the model can be estimated as follow.

```
cumlat <- brm(f,
              data = d,
              family = cumulative("probit"),
              prior = p_cumlat)
```

We specify cumulative model with `family = cumulative()` with link function `"probit"` which corresponds to normal likelihood for $\tilde{y}$.

We set prior in a same way as linear regression assuming the simplest model where linear predictor η does not exist and hence $SD(\tilde{y}) = 1$. As discussed in (Bürkner and Vuorre 2019), assuming ordinal variable as a metric variable (as in liner regression) would cause problems such as distortion in size of coefficients. Hence we expect cumulative model to perform better, which we will evaluate in further analysis later.

Note that we could also consider category specific effect where effect of predictive variables differs across categories (e.g. *alcohol* being important factor for lower *quality* but not for higher *quality*). However, we do not take this approach for our analysis given small number of data for some of the categories (especially for $quality = 3, 9$).

5.1 Sampling strategy for thresholds

By the construction of cumulative model, we have constraint on the thresholds

$$\tau_1 < \tau_2 < \cdots < \tau_{C-1}.$$

If this does not hold, we get negative probability $Pr(y = c) < 0$. This can be avoided by introducing unconstrained thresholds $\tilde{\tau}_c$, and define the (constrained) thresholds as

$$\tau_c = \begin{cases} \tilde{\tau}_1 & \text{if } c = 1 \\ \tau_{c-1} + \exp(\tilde{\tau}_c) & \text{if } 1 < c \leq C - 1. \end{cases}$$

Every time $\tilde{\tau}_c$ are sampled, we need to apply this transformation to obtain τ_c which will be then stored as posterior samples. Using **Stan**, this can be done by declaring the thresholds as ordered vector.

```
parameters {
  ...
  ordered[nthres] thresholds;
}
```

This is done in {Stan} code created by **brms**.

5.2 Result

The result for cumulative model is shown in Figure 5. We again focus on the two variables. The left column shows the conditional effect of each variables on $\tilde{y}$. It can be seen that the effects are similar to the one from linear regression. The right column shows the conditional effects on the probability of each qualities. For *alcohol*, since it has positive effect, larger *alcohol* corresponds to higher probability for higher *quality* such as 8 and 9 compare to smaller *alcohol*. We also see the opposite (e.g. 8% *alcohol* corresponds to high probability for *quality* 3 to 5). For *citric.acid*, the conditional probabilities are flatter since the effect is limited.

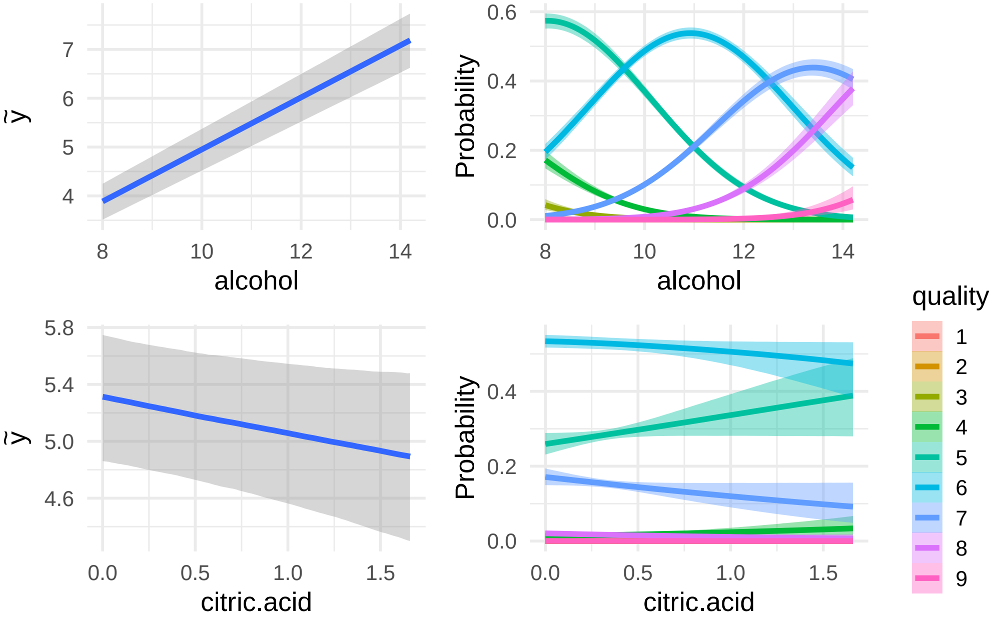


Figure 5: Result for cumulative model (only for alcohol and citric.acid)

In linear regression, we implicitly assumed equidistant among each qualities by assuming that *quality* is a metric variable. In cumulative model, we infer the distances by the threshold τ_c. The posterior distribution of τ_c is shown in Figure 6. It can be seen that the large part of posteriors for τ_3 and τ_4 overlaps, which implies lower *quality* wine are more similar to each other in terms of *quality*. We also see larger gap between the posterior for τ_6 and τ_7. This indicates that there is larger difference between *average quality* wine ($y = 6$) and *above average quality* wine ($y = 7$) [1].

6 Model Comparison

6.1 ELPD-LOO

We conduct model comparison using ELPD-LOO (Vehtari, Gelman, and Gabry 2017). It approximate the expected value of predictive density for each observation. Especially, the ELPD-LOO estimate the quantity

[1] We use the term *average* as general term other than arithmetic mean.

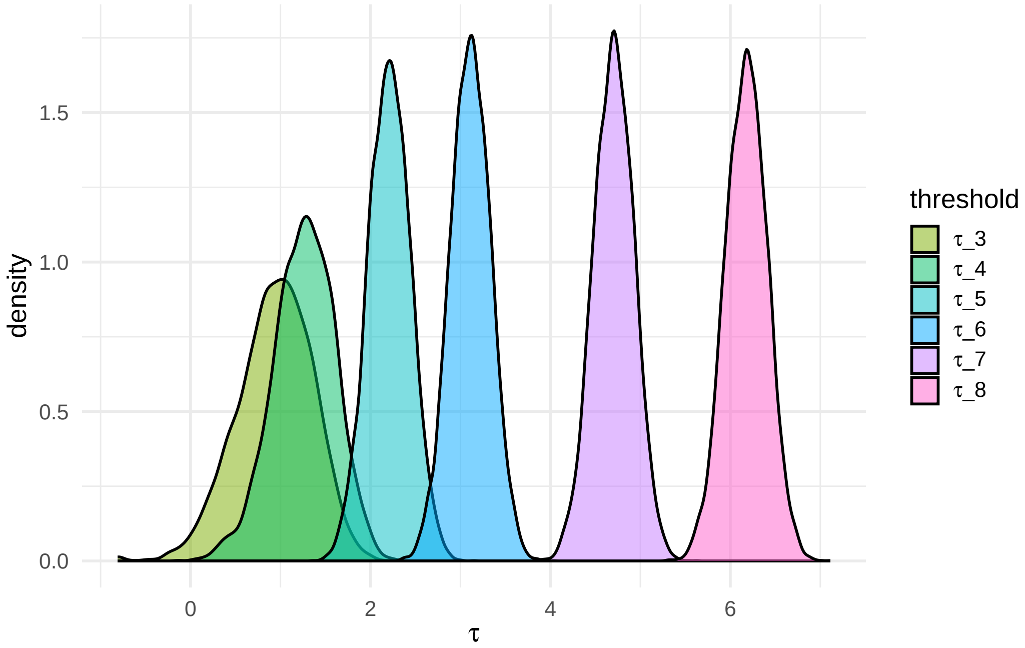


Figure 6: Posterior distribution for the thresholds

$$
\begin{aligned}
elpd_{loo} &= \sum_{i=1}^{n} \log p(y_i|y_{-i}) \\
&= \sum_{i=1}^{n} \log \int p(y_i|\theta)p(\theta|y_{-i})d\theta
\end{aligned}
$$

where $y_{-i} = y \backslash y_i$ and θ is a vector containing all the parameters. Note that we will not be able to approximate the integral directly with MCMC samples at hand since we do not have samples from $p(\theta|y_{-i})$ but instead we only have posterior samples $\theta^{(s)} \sim p(\theta|y)$. Hence, importance sampling is used to approximate the calculation. Then we can approximate $P(y_i|y_{-i})$ as

$$
\begin{aligned}
p(y_i|y_{-i}) &= \int p(y_i|\theta)p(\theta|y_{-i})d\theta \\
&= \int p(y_i|\theta)\frac{p(\theta|y_{-i})}{p(\theta|y)}p(\theta|y)d\theta \\
&:= \int p(y_i|\theta)r_i p(\theta|y)d\theta \\
&\approx \frac{1}{S}\sum_{s=1}^{S} p(y_i|\theta^{(s)})r_i^{(s)}
\end{aligned}
$$

where S is the number of MCMC samples. Using Bayes theorem and assuming the factorisation of likelihood, the importance ratio $r_i^{(s)}$ can be expressed as

$$
r_i^{(s)} \propto \frac{\prod_{j\in J, j\neq i} p(y_j|\theta^{(s)})p(\theta^{(s)})}{\prod_{j\in J} p(y_j|\theta^{(s)})p(\theta^{(s)})} = \frac{1}{p(y_i|\theta^{(s)})}
$$

where $J = \{1, ..., n\}$. Note that $r_i^{(s)}$ tends to take extreme value, which is stabilised by applying Pareto smoothing. This gives an estimation of $P(y_i|y_{-i})$ [2] and subsequently for $elpd_{loo}$.

6.2 Result

Since we modeled the response *quality* as continuous (metric) variable for linear regression, and discrete (ordinal) variable for cumulative model, we get posterior predictive density and posterior predictive probabilities respectively, which is not comparable directly in general (Vehtari, n.d.a). However, when response variable is an integer counts, density can be interpreted as approximate probability value and hence direct comparison is possible (Vehtari, n.d.b). This apply to our response variable *quality* and therefore we compare ELPD-LOO directly as follow.

```
loo_compare(linear_reg, cumlat)
```

```
           elpd_diff se_diff
cumlat       0.0       0.0  
linear_reg -37.8      10.0  
```

It can be seen that cumulative model takes larger ELPD-LOO by 37.8 or around 3.8 times of its standard error, which indicates that cumulative model perform better in terms of posterior predictive performance. This could be due to the choice of discrete model which is consistent with the response variable, or non-equidistant among categories modelled by the threshold τ_c.

6.3 Posterior predictive check

Additionally, we perform a posterior predictive check to evaluate the fit of our models. This involves the generation of simulated datasets y^{rep} derived from posterior samples. These simulated datasets are then used to inspect the model fit, specifically by checking the consistency between y^{rep} and the observed data y. The result is shown in Figure 7. We simulated 10 new datasets for each models. It can be seen obviously that linear regression does not fit observed data due to the choice of continuous model; it have density for value like 5.5 which will never occur in observed data, and it underestimate the density for observed values which is integer. For cumulative model, the fit looks much better although density tends to be overestimated for $y = 6$ and underestimated for $y = 5$.

From these results, we continue further analysis with cumulative model.

7 Adding nonlinearity with spline

We further try to improve our cumulative model by adding non-linearity. Intuitively, it is natural that there is non-linear relationship between physicochemical variables and the response variable. For example, too much *alcohol* or too much sweetness would lead bad tasting and lower *quality*. Therefore, we expect there exist some *optimal value* which maximizes the *quality* of wine and if the *optimal value* exist within the data range observed, we would want to model the non-linear

[2] Normalising term for $r_i^{(s)}$ is $\frac{1}{S}\sum_{s=1}^{S} r_i^{(s)}$

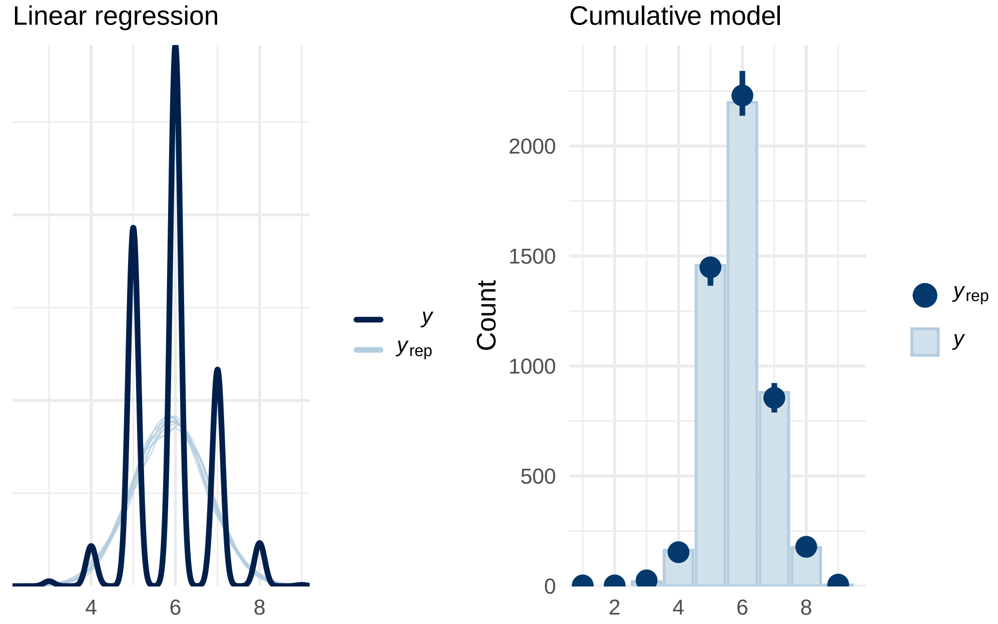


Figure 7: Density of y and 10 simulated datasets

relationship. For computational reason, we add non-linearity to the two variables *residual.sugar* and *total.sulfur.dioxide*. The posterior distributions of coefficients for these variable from our previous models are shown in Figure 8. It can be seen that posterior distribution concentrate close to zero. Although this might be due to the scale or simply small effect of variables, we suspect this might be due to non-linearity.

7.1 Spline

We add non-linearity with spline. The basic idea of spline is to approximate non-linear function with linear combination of basis functions. The discussion here will be based on (Wood 2017) and (Pedersen et al. 2019).

In univariate case for simplicity, spline model can be expressed as

$$\begin{aligned} y_i &= f(x_i) + \epsilon_i \\ &\approx \sum_{k=1}^{K} \beta_k B_k(x_i) + \epsilon_i \\ y &= B\beta + \epsilon \end{aligned}$$

where $B_k(x_i)$ is a basis function, β_k is a weight for each basis, and K is the number of basis. The third line gives matrix representation with $B_{(i,k)} = B_k(x_i)$ and $\beta = [\beta_1, ..., \beta_k]^T$. In general, fitting this model directly would give too wiggly approximation of function and lead to overfitting. Hence, one need to penalise for the wiggliness. In frequentist spline, this can be done by minimising following objective function

$$\hat{\beta} = \text{argmin}_\beta(\|y - B\beta\|^2 + \lambda \beta^T S \beta)$$

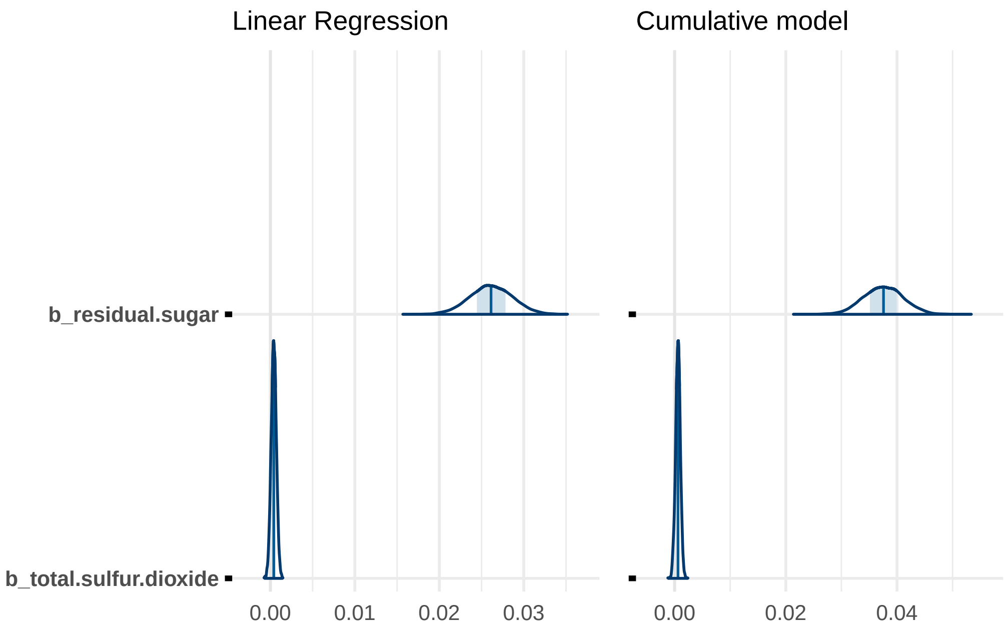


Figure 8: Posterior distribution of *b_residual.sugar* and *b_total.sulfur.dioxide*.

where S is a measurement of wiggliness and λ is a (hyper) parameter controlling the strength of penalisation. It turns out that this model can be expressed as a Bayesian multilevel model

$$\begin{aligned} y &= \Theta a + \Psi b + \epsilon \\ a &\sim \text{normal}(0, \sigma_a I) \\ \sigma_a &\sim p(\sigma_a) \\ b &\sim \prod_m p(b_m) \end{aligned}$$

where $\Theta = \Theta(B,S), \Psi = \Psi(B,S)$ are transformed basis and a, b are corresponding coefficients. The term Θa are penalised such that a is partially pooled towards global mean zero through penalisation parameter $\sigma_a \in \mathbb{R}_+$, which is also learnt by data. On the contrary, the penalisation is not applied for the term Ψb such that we set independent priors.

Cumulative model with spline term can be estimated using **brms** as follow

```r
f_s <- quality ~ citric.acid + volatile.acidity +
       sulphates + chlorides + alcohol +
       s(residual.sugar) + s(total.sulfur.dioxide)

cumlat_s <- brm(f_s,
                data = d,
                family = cumulative("probit"),
                prior = p_cumlat_s)
```

where `s()` construct the transformed basis using **mgcv** package (Wood 2017). [3]

[3] We used default values for construction of basis.

7.2 Result

The result for two variables with spline term is shown in Figure 9. For *residual.sugar*, it can be seen that non-linearity is unclear given considerable amount of uncertainty, which implies that small linear coefficient can be simply due to small effect or its scale. On the other hand, we can see that there is concave-like relationship between *residual.sugar* and *quality* where around 100 to 250 mg/L of *residual.sugar* leads to high *quality* and as the value goes further away from this range, *quality* decreases.

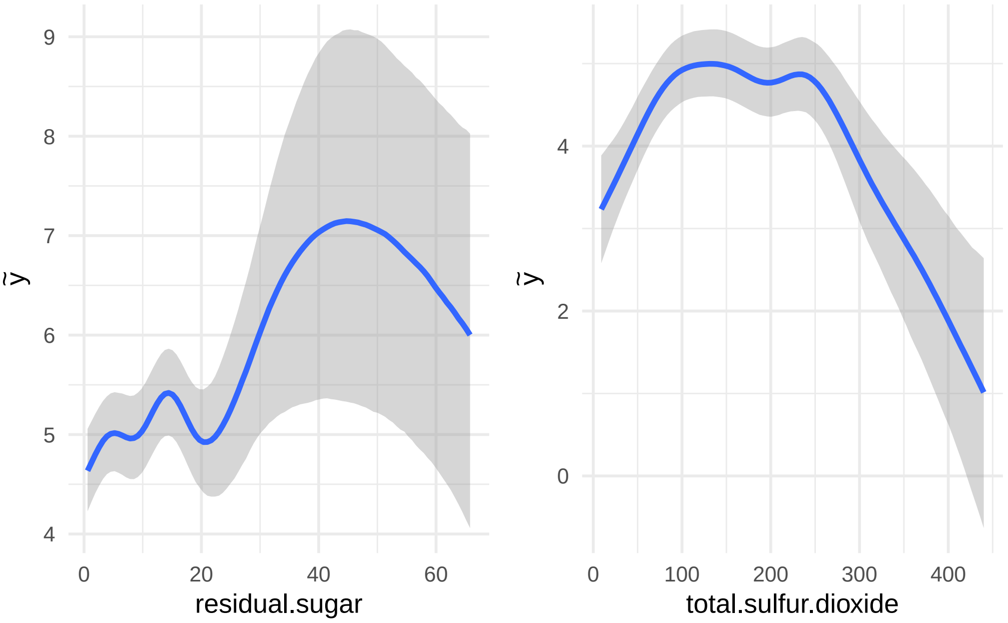


Figure 9: Conditional effect for cumulative model with spline (only variables with spline term)

Finally, we compare predictive performance of all three models using ELPD-LOO. The result is following. It can be seen that adding spline term improved the ELPD by 91.9 around 6 times of its standard error, which indicates that predictive performance improved notably.

```
loo_compare(linear_reg, cumlat, cumlat_s)
```

```
           elpd_diff se_diff
cumlat_s      0.0       0.0 
cumlat      -91.9      15.4 
linear_reg -129.8      19.0 
```

8 Conclusion

In this report, we estimated three Bayesian models to predict *quality* of wine using 7 physicochemical variables. Focusing on two variables, we found that positive effect of *alcohol* and negative effect of *citric.acid* although the latter effect is limited. Evaluating predictive performance of model with the ELPD, it was shown that cumulative model with spline is the best performing. Comparing with linear regression model, this can be due to the choice of discrete model which is consistent with the response, or non-equidistant of categories modelled by the thresholds. Adding spline term also improved the model further, which indicates the non linear functional form between response and predictive variables. For further analysis, we could add non-linearity to more variables, and also consider synergy effect by adding interaction terms or using tensor product spline.

References

Bürkner, Paul-Christian. 2017. "brms: An R Package for Bayesian Multilevel Models Using Stan." *Journal of Statistical Software* 80 (1): 1–28. https://doi.org/10.18637/jss.v080.i01.

Bürkner, Paul-Christian, and Matti Vuorre. 2019. "Ordinal Regression Models in Psychology: A Tutorial." *Advances in Methods and Practices in Psychological Science* 2 (1): 77–101. https://doi.org/10.1177/2515245918823199.

Carpenter, Bob, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. "Stan: A Probabilistic Programming Language." *Journal of Statistical Software* 76 (1).

Cortez, Cerdeira, Paulo, and J. Reis. 2009. "Wine Quality." UCI Machine Learning Repository.

Pedersen, Eric J, David L Miller, Gavin L Simpson, and Noam Ross. 2019. "Hierarchical Generalized Additive Models in Ecology: An Introduction with Mgcv." *PeerJ* 7: e6876.

Simpson, Daniel, Håvard Rue, Andrea Riebler, Thiago G. Martins, and Sigrunn H. Sørbye. 2017. "Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors." *Statistical Science* 32 (1): 1–28. https://doi.org/10.1214/16-STS576.

Vehtari, Aki. n.d.a. "Cross-Validation FAQ." https://users.aalto.fi/~ave/CV-FAQ.html.

———. n.d.b. "Nabiximols Treatment Efficiency." https://users.aalto.fi/~ave/casestudies/Nabiximols/nabiximols.html.

Vehtari, Aki, Andrew Gelman, and Jonah Gabry. 2017. "Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and WAIC." *Statistics and Computing* 27: 1413–32.

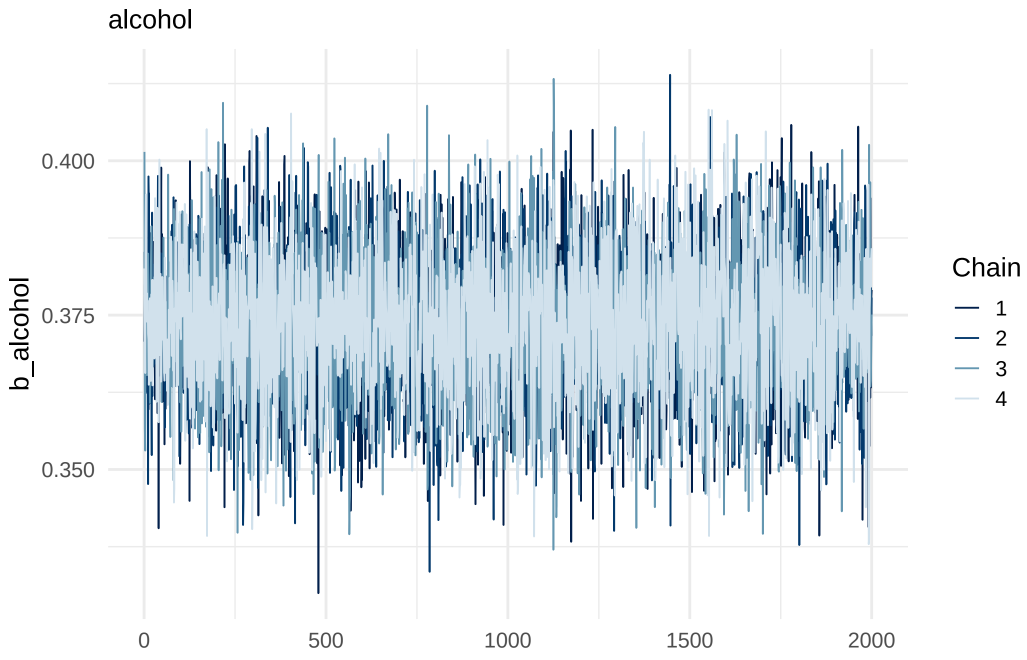
Wood, S. N. 2017. *Generalized Additive Models: An Introduction with r*. 2nd ed. Chapman; Hall/CRC.

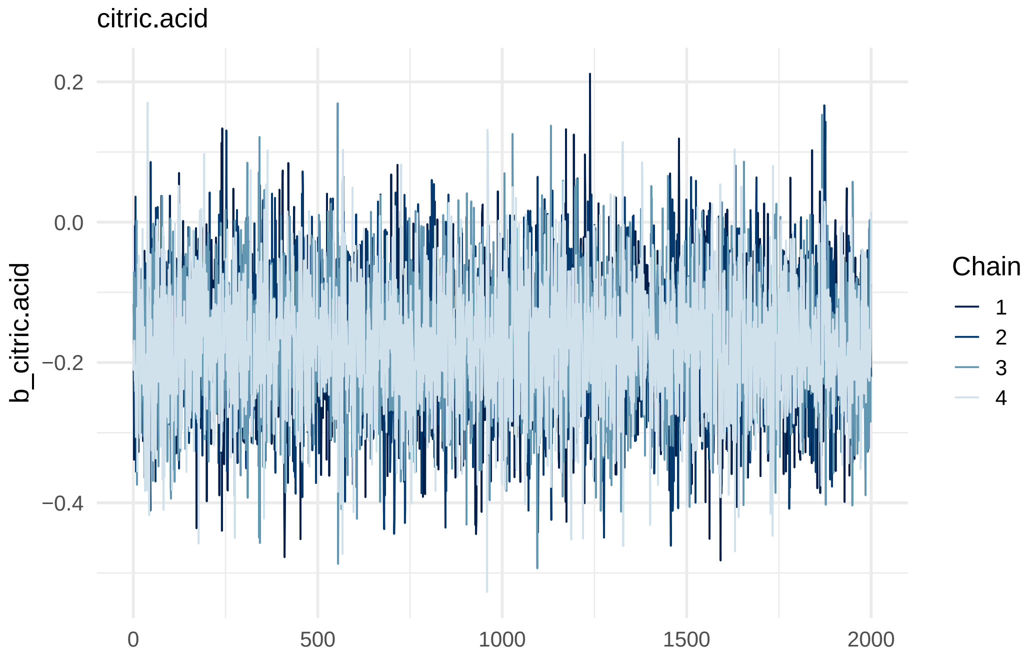
Code Repository: https://github.com/1129hiki/abda_project

Appendix

A.1 Model diagnosis for selected parameters

Trace plots for coefficients of *alcohol* and *citric.acid* in linear regression.





A.2 Model summaries

Linear Regression

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: quality ~ citric.acid + volatile.acidity + residual.sugar + sulphates + chlorides + total.sulfur.dioxide + alcohol
   Data: d (Number of observations: 4898)
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Population-Level Effects:
                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept                2.20      0.15     1.90     2.50 1.00     8186     6717
citric.acid             -0.17      0.09    -0.35     0.01 1.00     8908     5856
volatile.acidity        -2.12      0.11    -2.34    -1.90 1.00     8456     5703
residual.sugar           0.03      0.00     0.02     0.03 1.00    10690     6525
sulphates                0.44      0.10     0.25     0.64 1.00     9337     5876
chlorides               -0.87      0.54    -1.94     0.20 1.00     7324     5997
```

```
total.sulfur.dioxide     0.00      0.00    -0.00     0.00 1.00     8627     6665
alcohol                  0.37      0.01     0.35     0.40 1.00     7269     5937

Family Specific Parameters: 
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     0.76      0.01     0.75     0.78 1.00     9825     5584

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Cumulative model

```
 Family: cumulative 
  Links: mu = probit; disc = identity 
Formula: quality ~ citric.acid + volatile.acidity + residual.sugar + sulphates + chlorides + total.sulfur.dioxide + alcohol 
   Data: d (Number of observations: 4898) 
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Population-Level Effects: 
                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept[1]             0.94      0.42     0.08     1.67 1.00     3026     2646
Intercept[2]             1.26      0.34     0.54     1.89 1.00     4158     3398
Intercept[3]             2.21      0.23     1.75     2.66 1.00     5834     5648
Intercept[4]             3.11      0.22     2.66     3.54 1.00     5747     5589
Intercept[5]             4.70      0.22     4.26     5.14 1.00     5618     5566
Intercept[6]             6.18      0.23     5.72     6.62 1.00     5395     5627
Intercept[7]             7.34      0.24     6.88     7.79 1.00     5348     5466
Intercept[8]             8.77      0.27     8.23     9.30 1.00     5629     5625
citric.acid             -0.25      0.13    -0.51     0.01 1.00     5557     5392
volatile.acidity        -3.11      0.16    -3.42    -2.79 1.00     5336     5236
residual.sugar           0.04      0.00     0.03     0.04 1.00     6909     5780
sulphates                0.63      0.14     0.37     0.90 1.00     5755     4027
chlorides               -1.26      0.78    -2.79     0.29 1.00     6024     4579
total.sulfur.dioxide     0.00      0.00    -0.00     0.00 1.00     8104     6304
alcohol                  0.53      0.02     0.50     0.57 1.00     4876     5522

Family Specific Parameters: 
     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
disc     1.00      0.00     1.00     1.00   NA       NA       NA

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Cumulative model with spline

```
 Family: cumulative 
  Links: mu = probit; disc = identity 
Formula: quality ~ s(residual.sugar) + s(total.sulfur.dioxide) + citric.acid + volatile.acidity + sulphates + chlorides + alcohol 
   Data: d (Number of observations: 4898) 
  Draws: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
         total post-warmup draws = 8000

Smooth Terms: 
                             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sds(sresidual.sugar_1)           9.23      2.80     4.97    15.72 1.00     2470     4390
sds(stotal.sulfur.dioxide_1)     3.47      1.18     1.78     6.37 1.00     3041     4189

Population-Level Effects: 
                        Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept[1]                0.34      0.45    -0.66     1.12 1.00     4096     1960
Intercept[2]                0.68      0.36    -0.09     1.32 1.00     5887     3241
Intercept[3]                1.73      0.22     1.31     2.16 1.00     6835     5375
Intercept[4]                2.69      0.20     2.30     3.09 1.00     7458     5016
Intercept[5]                4.33      0.20     3.93     4.74 1.00     7462     4941
Intercept[6]                5.83      0.21     5.43     6.25 1.00     7700     5253
Intercept[7]                7.02      0.22     6.60     7.45 1.00     7773     5383
Intercept[8]                8.46      0.26     7.97     8.98 1.00     7144     3505
citric.acid                -0.18      0.14    -0.44     0.08 1.00     9277     4803
volatile.acidity           -3.04      0.16    -3.35    -2.72 1.00     8595     6104
sulphates                   0.60      0.14     0.33     0.87 1.00     8753     5654
chlorides                  -1.58      0.78    -3.12    -0.06 1.00     9246     6296
alcohol                     0.53      0.02     0.49     0.56 1.00     7822     5927
sresidual.sugar_1           2.40      2.57    -2.60     7.39 1.00     5352     2885
stotal.sulfur.dioxide_1    -0.06      2.36    -4.81     4.54 1.00     6721     4474

Family Specific Parameters: 
     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
disc     1.00      0.00     1.00     1.00   NA       NA       NA

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

A.3 Prior summaries

Linear Regression

```
           prior     class                 coef group resp dpar nlpar lb ub  source
          (flat)         b                                                  default
  normal(0,0.36)         b              alcohol                                user
normal(0,20.268)         b            chlorides                                user
 normal(0,3.659)         b          citric.acid                                user
 normal(0,0.087)         b       residual.sugar                                user
  normal(0,3.88)         b            sulphates                                user
  normal(0,0.01)         b total.sulfur.dioxide                                user
 normal(0,4.393)         b     volatile.acidity                                user
    normal(6, 5) Intercept                                                     user
    normal(0, 5)     sigma                                           0         user
```

Cumulative model

```
               prior     class                 coef group resp dpar nlpar lb ub  source
              (flat)         b                                                  default
     normal(0,0.406)         b              alcohol                                user
    normal(0,22.885)         b            chlorides                                user
    normal(0, 4.132)         b          citric.acid                                user
     normal(0,0.099)         b       residual.sugar                                user
      normal(0,3.88)         b            sulphates                                user
     normal(0,0.012)         b total.sulfur.dioxide                                user
     normal(0,4.961)         b     volatile.acidity                                user
student_t(3, 0, 2.5) Intercept                                                  default
       normal(-2, 1) Intercept                    1                                user
    normal(-1.43, 1) Intercept                    2                                user
    normal(-0.86, 1) Intercept                    3                                user
    normal(-0.29, 1) Intercept                    4                                user
     normal(0.29, 1) Intercept                    5                                user
     normal(0.86, 1) Intercept                    6                                user
     normal(1.43, 1) Intercept                    7                                user
        normal(2, 1) Intercept                    8                                user
```

Cumulative model with Spline

```
               prior     class                    coef group resp dpar nlpar lb ub       source
              (flat)         b                                                          default
     normal(0,0.406)         b                 alcohol                                     user
    normal(0,22.885)         b               chlorides                                     user
    normal(0, 4.132)         b             citric.acid                                     user
        normal(0, 3)         b       sresidual.sugar_1                                     user
        normal(0, 3)         b stotal.sulfur.dioxide_1                                     user
      normal(0,3.88)         b               sulphates                                     user
     normal(0,4.961)         b        volatile.acidity                                     user
student_t(3, 0, 2.5) Intercept                                                          default
       normal(-2, 1) Intercept                       1                                     user
    normal(-1.43, 1) Intercept                       2                                     user
    normal(-0.86, 1) Intercept                       3                                     user
    normal(-0.29, 1) Intercept                       4                                     user
     normal(0.29, 1) Intercept                       5                                     user
     normal(0.86, 1) Intercept                       6                                     user
     normal(1.43, 1) Intercept                       7                                     user
        normal(2, 1) Intercept                       8                                     user
student_t(3, 0, 2.5)       sds                                                0         default
student_t(3, 0, 2.5)       sds       s(residual.sugar)                        0    (vectorized)
student_t(3, 0, 2.5)       sds s(total.sulfur.dioxide)                        0    (vectorized)
```