PROJECT

BLAL BLA BLA

anonymous

1 Introduction

- Background
- Problem formulation/scope
- Main modeling idea
- Some picture of the data?

2 Data Desctription

The "cath" dataset used in this report is obtained from Duke University Cardiovascular Disease Databank. It encapsulates a collection of 6 variables (@Data-table) that are closely related to cardiovascular health.

Table 1: TBD

sex	age	cad_dur	choleste	sigdz	tvdlm
0	73	132	268	1	1
0	68	85	120	1	1
0	54	45	NA	1	0
1	58	86	245	0	0
1	56	7	269	0	0
0	64	0	NA	1	0

The dataset consists of four explanatory variables (*sex*, *age*, *cad_dur*, *choleste*) and two response variables (*sigdz*, *tvdlm*) that provide an overview on patient demographics, clinical indicators, and critical outcomes related to coronary artery disease:

- **Sex** (*sex*): Categorized as 0 for male and 1 for female, this variable represents the gender distribution within our dataset.

- **Age** (*age*): Representing the age of patients in years, this variable serves as a demographic feature.
- **Chest Pain Duration** (*cad_dur*): The duration of chest pain symptoms in days.
- **Serum Cholesterol Level** (*choleste*): Measured in milligrams per deciliter, serum cholesterol levels are indicative of lipid metabolism and play a crucial role in cardiovascular health.
- **Significant Coronary Disease** (*sigdz*): A binary variable that captures the presence (1) or absence (0) of at least 75% blockage in one of the major coronary arteries.
- **Three Vessel Disease or Left Main Disease** (*tvdlm*): Denoting the presence (1) or absence (0) of blockage in either all three coronary vessels or in the left main coronary artery.

The univariate distributions of these variables are described in @Univariate-analysis.

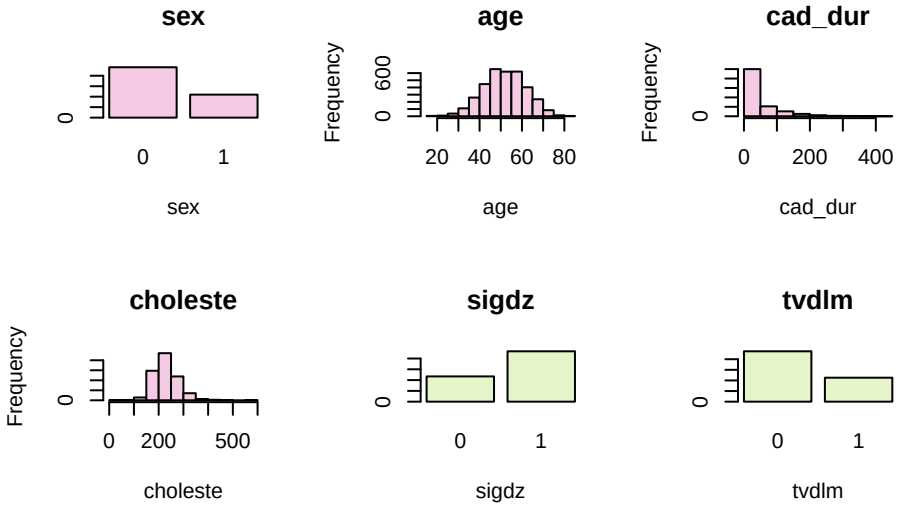


Figure 1: TBD 1

While constructing the Bayesian models to predict the probaility of significant coronary disease, the report strives to utilize the correlation between the explanatory variables (*sex*, *age*, *cad_dur*, *choleste*) and the desired response variable (*sigdz*). The *tvdlm* variable is not relevant in this report as the main focus is to predict the probability of significant coronary disease, independent of the type of the blockade.

Before the analysis, the data is preprocessed by removing *tvdlm* column and all rows that contain missing values, as well as by scaling the continuous variables to zero mean and unit variance. After this, we are left with $n = 2258$ observations.

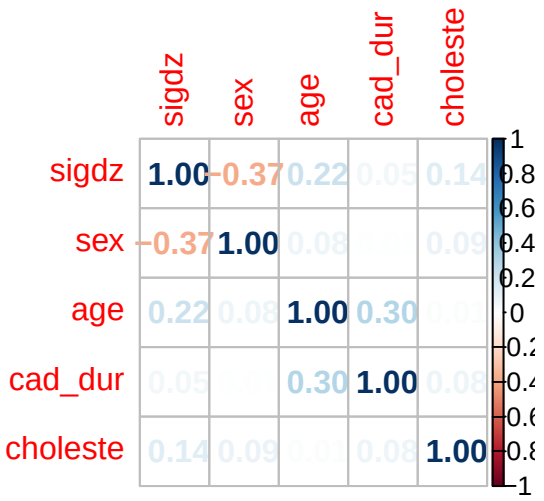


Figure 2: TBD 3

Seems that sex and age have the most significant bivariate correlation to the responsive variable sigdz.

3 Mathematical Model

- Bayesian logistic regression
- Likelihood
- Priors
- Posteriors

3.1 The generalized linear model and priors

For the binary response variable y, the likelihood for the binomial GLM can be written as a conditionally binomial PMF.

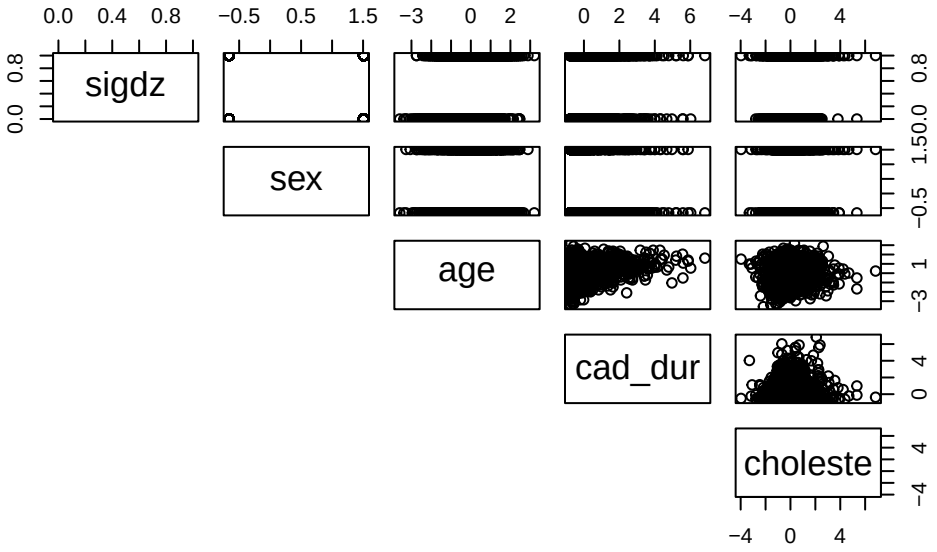


Figure 3: TBD 3

$$\binom{n}{y}\pi(1-\pi)^{n-y},$$

where n is the total amount of trials, $\pi = g^{-1}(\eta)$ is the probability of success (patient presenting with significant coronary disease) and $\eta = \alpha + x^T\beta$ is a linear predictor. For a sample of size N, the likelihood is the product of the N individual likelihoods.

The link function g maps the probability π between the unit interval and the set of real numbers $\mathbb{R}$, and when applying the inverse link function to the linear predictor ηH the output will be a valid probability between 0 and 1.

This project utilizes the logit link function for the GLM, which makes it a logistic regression model. The likelihood expressed with the logit link function $g(x) = \ln(\frac{x}{1-x})$ for a single observation can be written as:

$$\binom{n}{y}(\text{logit}^{-1}(\eta))^y(1-\text{logit}^{-1})^{n-y}$$

Priors are set for the intercept and vector of regression coefficients α and β, from the linear predictor $\eta = \alpha + x^T\beta$,

$$\alpha \sim t_v(\mu, \sigma^2)$$
$$\beta_k \sim t_v(\mu, \sigma^2)$$

where v is the degrees of freedom, μ is the location and σ is the scale.

The intercept and the regression coefficients are believed to be as likely positive as they would be negative, but likely relatively close to zero. This can be represented with normal distribution with a mean of zero and a small standard deviation. For example, $\mathcal{N}(0, 1)$. The priors can also be represented with the Students t-distribution if there is less priori confidence that the parameters will be close to zero. The Students t-distribution includes a larger standard deviation and has heavier tails than the normal distribution and would therefore be suitable for this purpose.

3.2 The additive non-linear model

The additive non-linear model on the other hand combines multiple functions in a way that isn't strictly linear. This allows for a more flexible relationship between the explanatory and response variable y. In this report, the non-linear model uses the same link function, and the logistic regression model can be written as:

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 f_1(x_1) + \beta_2 f_2(x_2) ... \beta_n f_n(x_n),$$

where f_i are non-linear functions that transform the explanatory features individually. In this report, all functions are smooth functions except f_{age}, which would only be the variable itself. The smooth functions utilize penalized splines, allowing the model to create a curved relationship between the features. The shape of the smooth functions is estimated from the data and the penalty helps avoid overfitting.nThe smooth function works by minimizing the sum of the model fit with the smoothness / penalty (here, thin plate regression splines (default)).

The likelihood would then become:

$$\prod_{i=1}^{N} \text{Bernoulli}(y|\text{logit}^{-1}(\pi_i), \sigma)$$

4 Model Definitions and Implementation

- Tie mathematical model to our implementation
- Ndraws, warmup, etc.

4.1 Linear model

```
# Make response variable a factor
cath$sigdz <- as.factor(cath$sigdz)

x <- model.matrix(sigdz ~ . - 1, data=cath)
y <- cath$sigdz

# Formula
formula_linear <- formula(sigdz ~ sex + age + cad_dur + choleste)

# Prior
prior_linear <- student_t(df = 3, location = 0, scale = 2.5)

# The model
# model_linear <- stan_glm(formula_linear, data = cath,
#                  family = binomial(link = "logit"),
#                  prior = prior_linear, prior_intercept = prior_linear, QR=TRUE,
#                  refresh=0)

# saveRDS(model_linear, file = "./additional_files/model_linear.rds")
model_linear <- readRDS("./additional_files/model_linear.rds")
```

4.2 Nonlinear model

```
# Formula
formula_nonlinear <- formula(sigdz ~ sex + s(age) + s(cad_dur) + s(choleste))

# Model definition
# model_nonlinear <- stan_gamm4(
#   formula_nonlinear, data = cath,
#   family = binomial(link = "logit"),
#   prior = prior_linear, prior_intercept = prior_linear,
#   refresh = 0
# )

# saveRDS(model_nonlinear, file = "./additional_files/model_nonlinear.rds")
model_nonlinear <- readRDS("./additional_files/model_nonlinear.rds")
```

5 Model Evaluation

- Rhat, ESS, HMC divergences, pp_check(), loo_compare(), classification accuracy
- Prior sensitivity analysis

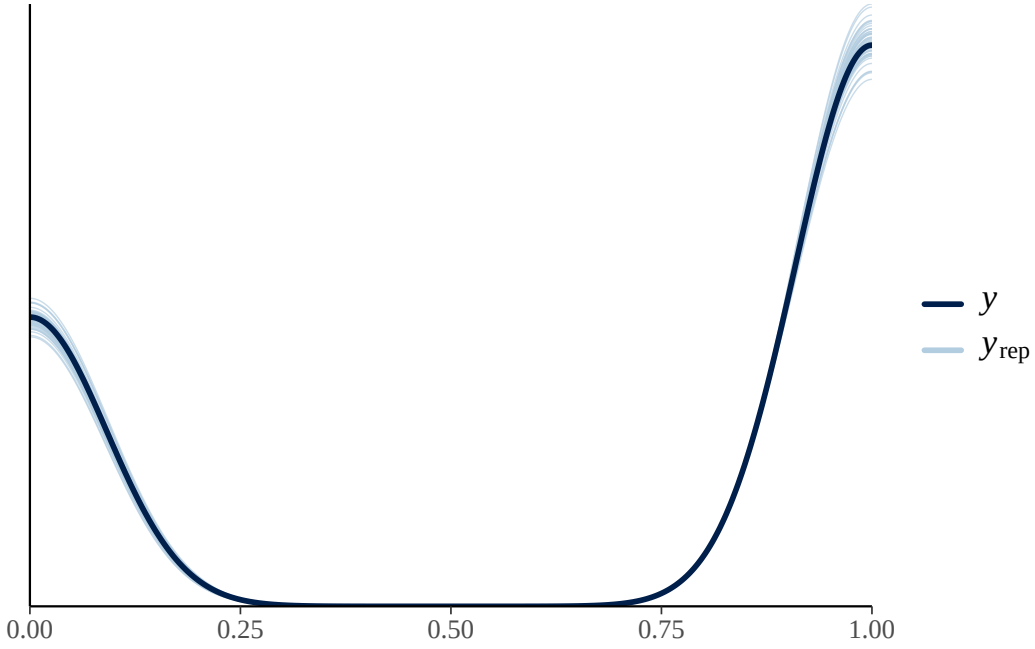
5.1 Convergence diagnostics & posterior predictive checks

```
summary(model_linear)
```

```
Model Info:
 function:     stan_glm
 family:       binomial [logit]
 formula:      sigdz ~ sex + age + cad_dur + choleste
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 2258
 predictors:   5

Estimates:
              mean   sd   10%   50%   90%
(Intercept)  0.8    0.1  0.7   0.8   0.9
sex         -1.0    0.1 -1.0  -1.0  -0.9
age          0.7    0.1  0.6   0.7   0.8
cad_dur     -0.1    0.1 -0.2  -0.1   0.0
choleste     0.5    0.1  0.4   0.5   0.5

Fit Diagnostics:
           mean   sd   10%   50%   90%
mean_PPD 0.7    0.0  0.6   0.7   0.7

The mean_ppd is the sample average posterior predictive distribution of the outcome variable

MCMC diagnostics
              mcse Rhat n_eff
(Intercept)   0.0  1.0  5042
sex           0.0  1.0  4810
age           0.0  1.0  5221
cad_dur       0.0  1.0  5889
```

```
choleste      0.0  1.0  4825
mean_PPD      0.0  1.0  5017
log-posterior 0.0  1.0  1902

For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective
```

```
pp_check(model_linear)
```



Nonlinear model:

```
summary(model_nonlinear)
```

```
Model Info:
 function:     stan_gamm4
 family:       binomial [logit]
 formula:      sigdz ~ sex + s(age) + s(cad_dur) + s(choleste)
 algorithm:    sampling
 sample:       4000 (posterior sample size)
 priors:       see help('prior_summary')
 observations: 2258
```

Estimates:

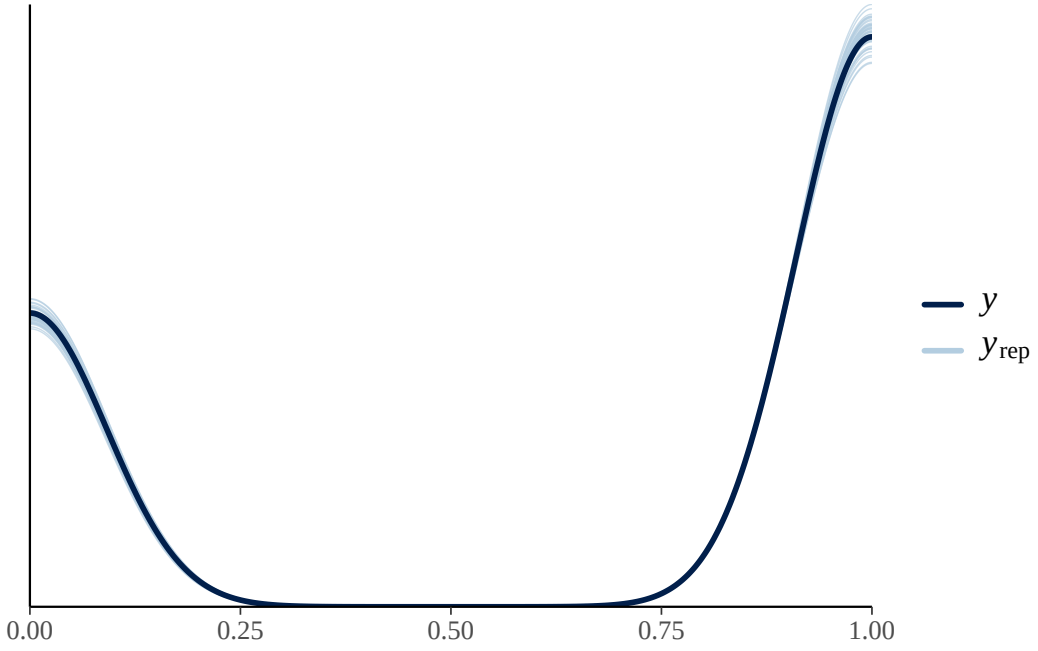
	mean	sd	10%	50%	90%
(Intercept)	0.8	0.1	0.7	0.8	0.9
sex	-1.0	0.1	-1.0	-1.0	-0.9
s(age).1	-0.8	2.2	-3.6	-0.4	1.6
s(age).2	-0.8	2.5	-4.1	-0.3	1.8
s(age).3	-2.1	2.4	-5.4	-1.6	0.3
s(age).4	1.1	2.0	-1.0	0.7	3.7
s(age).5	3.2	2.3	0.1	3.1	6.2
s(age).6	-0.6	1.0	-1.9	-0.4	0.5
s(age).7	1.1	0.9	-0.1	1.3	2.2
s(age).8	-1.3	2.3	-4.2	-0.8	0.9
s(age).9	2.1	2.9	-0.4	0.8	6.7
s(cad_dur).1	-0.2	1.4	-1.7	-0.1	1.2
s(cad_dur).2	0.2	1.3	-1.1	0.1	1.7
s(cad_dur).3	0.4	1.1	-0.9	0.2	1.7
s(cad_dur).4	-0.4	1.4	-2.1	-0.2	0.9
s(cad_dur).5	-0.9	1.1	-2.4	-0.6	0.2
s(cad_dur).6	0.8	1.1	-0.3	0.7	2.2
s(cad_dur).7	0.5	1.0	-0.5	0.3	1.7
s(cad_dur).8	-0.1	0.7	-0.9	0.0	0.8
s(cad_dur).9	-0.2	1.0	-1.4	-0.1	0.6
s(choleste).1	0.1	2.1	-2.5	0.1	2.6
s(choleste).2	-0.3	2.2	-3.0	-0.2	2.3
s(choleste).3	-1.0	2.1	-3.6	-0.9	1.3
s(choleste).4	-1.2	2.0	-3.9	-1.1	1.1
s(choleste).5	-0.1	1.7	-2.2	0.0	2.0
s(choleste).6	0.4	1.2	-1.1	0.4	1.9
s(choleste).7	3.5	1.1	2.2	3.4	4.9
s(choleste).8	0.4	1.9	-1.8	0.2	2.9
s(choleste).9	0.2	1.4	-0.9	0.0	1.6
smooth_sd[s(age)1]	2.1	1.3	0.4	2.1	3.9
smooth_sd[s(age)2]	1.8	1.6	0.2	1.4	4.1
smooth_sd[s(cad_dur)1]	1.1	0.7	0.3	1.0	2.1
smooth_sd[s(cad_dur)2]	0.9	0.9	0.1	0.6	2.1
smooth_sd[s(choleste)1]	2.2	0.8	1.3	2.0	3.3
smooth_sd[s(choleste)2]	1.0	1.0	0.1	0.7	2.3

Fit Diagnostics:

	mean	sd	10%	50%	90%
mean_PPD	0.7	0.0	0.6	0.7	0.7

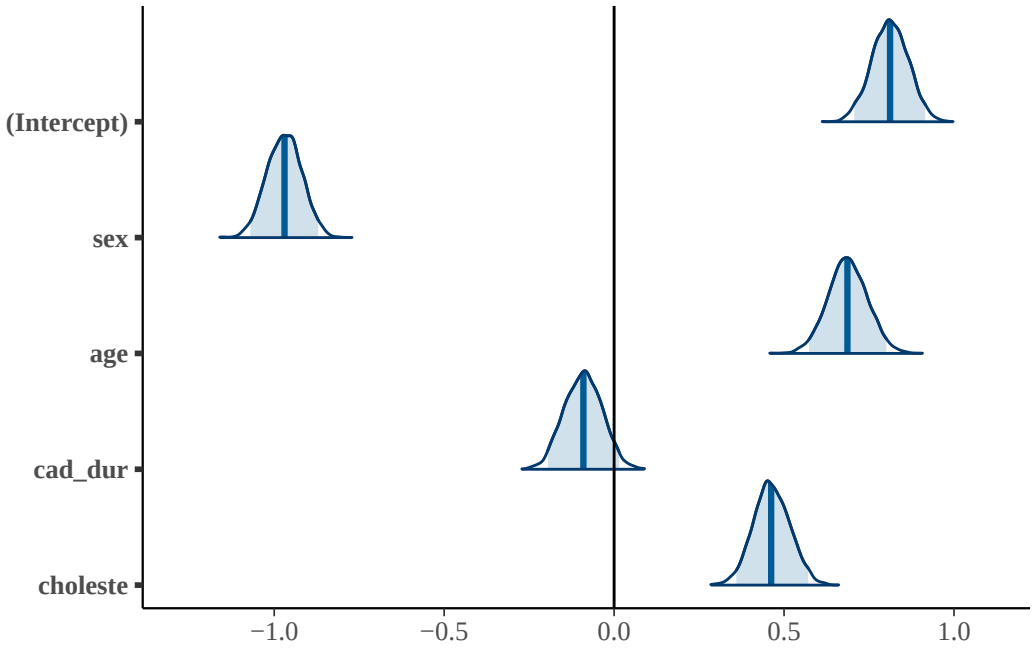
```
The mean_ppd is the sample average posterior predictive distribution of the outcome variable

MCMC diagnostics
                        mcse Rhat n_eff
(Intercept)             0.0  1.0  3912
sex                     0.0  1.0  4064
s(age).1                0.0  1.0  3263
s(age).2                0.0  1.0  2942
s(age).3                0.1  1.0  1210
s(age).4                0.0  1.0  2419
s(age).5                0.1  1.0   574
s(age).6                0.0  1.0  2018
s(age).7                0.0  1.0   566
s(age).8                0.1  1.0  1925
s(age).9                0.1  1.0   432
s(cad_dur).1            0.0  1.0  4061
s(cad_dur).2            0.0  1.0  4282
s(cad_dur).3            0.0  1.0  3874
s(cad_dur).4            0.0  1.0  3359
s(cad_dur).5            0.0  1.0  2938
s(cad_dur).6            0.0  1.0  2890
s(cad_dur).7            0.0  1.0  3508
s(cad_dur).8            0.0  1.0  3698
s(cad_dur).9            0.0  1.0  2818
s(choleste).1           0.0  1.0  4074
s(choleste).2           0.0  1.0  3904
s(choleste).3           0.0  1.0  3681
s(choleste).4           0.0  1.0  3634
s(choleste).5           0.0  1.0  4740
s(choleste).6           0.0  1.0  3436
s(choleste).7           0.0  1.0  2143
s(choleste).8           0.0  1.0  3006
s(choleste).9           0.0  1.0  1444
smooth_sd[s(age)1]      0.1  1.0   542
smooth_sd[s(age)2]      0.1  1.0   679
smooth_sd[s(cad_dur)1]  0.0  1.0  1730
smooth_sd[s(cad_dur)2]  0.0  1.0  3642
smooth_sd[s(choleste)1] 0.0  1.0  2121
smooth_sd[s(choleste)2] 0.0  1.0  2998
mean_PPD                0.0  1.0  4238
log-posterior           0.2  1.0   766

For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective
```

```
pp_check(model_nonlinear)
```



Both:

```
pplot_linear<-plot(model_linear, "areas", prob = 0.95, prob_outer = 1)
pplot_linear + geom_vline(xintercept = 0)
```




```
# pplot_nonlinear<-plot(model_nonlinear, "areas", prob = 0.95, prob_outer = 1)
# pplot_nonlinear + geom_vline(xintercept = 0)
```

5.2 Model comparison using LOO-CV

```
# LOO of our models
loo_linear <- loo(model_linear, save_psis = TRUE)
loo_nonlinear <- loo(model_nonlinear, save_psis = TRUE)
print(loo_linear)
```

```
Computed from 4000 by 2258 log-likelihood matrix

         Estimate   SE
elpd_loo  -1177.4 24.4
p_loo         5.2  0.2
looic      2354.8 48.7
------
Monte Carlo SE of elpd_loo is 0.0.
```

```
All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.
```

```r
print(loo_nonlinear)
```

```
Computed from 4000 by 2258 log-likelihood matrix

         Estimate   SE
elpd_loo  -1170.6 24.3
p_loo        11.6  0.7
looic      2341.3 48.6
------
Monte Carlo SE of elpd_loo is 0.1.

All Pareto k estimates are good (k < 0.5).
See help('pareto-k-diagnostic') for details.
```

```r
# Baseline model of the linear model
model_baseline <- update(model_linear, formula = sigdz ~ 1, QR = FALSE)

# LOO of the baseline model
loo_baseline <- loo(model_baseline, save_psis = TRUE)
```

Comparing the LOOs:

```r
loo_compare(loo_linear, loo_nonlinear, loo_baseline)
```

```
               elpd_diff se_diff
model_nonlinear    0.0       0.0
model_linear      -6.8       3.4
model_baseline  -278.1      21.8
```

The nonlinear model seems to be the best!

5.3 Posterior predictive performance: Classification accuracy

5.3.1 Linear model:

```
## Predicted probabilities
linpred_linear <- posterior_linpred(model_linear)
preds_linear <- posterior_epred(model_linear)
pred_linear <- colMeans(preds_linear)
pr_linear <- as.integer(pred_linear >= 0.5)

## posterior classification accuracy
round(mean(xor(pr_linear,as.integer(y==0))),2)
```

```
[1] 0.75
```

```
# posterior balanced classification accuracy
round((mean(xor(pr_linear[y==0]>0.5,as.integer(y[y==0])))+mean(xor(pr_linear[y==1]<0.5,as.
```

```
[1] 0.69
```

LOO balanced (Better estimate, maybe let's include only these?):

```
# LOO predictive probabilities
ploo_linear=E_loo(preds_linear, loo_linear$psis_object, type="mean", log_ratios = -log_lik

# LOO classification accuracy
round(mean(xor(ploo_linear>0.5,as.integer(y==0))),2)
```

```
[1] 0.75
```

```
# LOO balanced classification accuracy
round((mean(xor(ploo_linear[y==0]>0.5,as.integer(y[y==0])))+mean(xor(ploo_linear[y==1]<0.5
```

```
[1] 0.69
```

5.3.2 Nonlinear model

```
## Predicted probabilities
linpred_nonlinear <- posterior_linpred(model_nonlinear)
preds_nonlinear <- posterior_epred(model_nonlinear)
pred_nonlinear <- colMeans(preds_nonlinear)
pr_nonlinear <- as.integer(pred_nonlinear >= 0.5)

## posterior classification accuracy
round(mean(xor(pr_nonlinear,as.integer(y==0))),2)
```

```
[1] 0.76
```

```
# posterior balanced classification accuracy
round((mean(xor(pr_linear[y==0]>0.5,as.integer(y[y==0])))+mean(xor(pr_nonlinear[y==1]<0.5,
```

```
[1] 0.69
```

LOO balanced (Better estimate, maybe let's include only these?):

```
# LOO predictive probabilities
ploo_nonlinear=E_loo(preds_nonlinear, loo_nonlinear$psis_object, type="mean", log_ratios =

# LOO classification accuracy
round(mean(xor(ploo_nonlinear>0.5,as.integer(y==0))),2)
```

```
[1] 0.76
```

```
# LOO balanced classification accuracy
round((mean(xor(ploo_nonlinear[y==0]>0.5,as.integer(y[y==0])))+mean(xor(ploo_nonlinear[y==
```

```
[1] 0.69
```

5.3.3 Calibration curves

```
cplot_linear <- ggplot(data = data.frame(pred=pred_linear,loopred=ploo_linear,
  y=as.numeric(y)-1), aes(x=loopred, y=y)) +
  stat_smooth(method='glm', formula = y ~ ns(x, 5), fullrange=TRUE, color="deeppink") +
```

```
  geom_abline(linetype = 'dashed') +
  labs(x = "Predicted (LOO)",y = "Observed",title = "Linear model - Calibration plot") +
  geom_jitter(height=0.02, width=0, alpha=0.05) +
  scale_y_continuous(breaks=seq(0,1,by=0.1)) +
  xlim(c(0,1)) +
  theme_minimal()

cplot_nonlinear <- ggplot(data = data.frame(loopred=ploo_nonlinear,
  y=as.numeric(y)-1), aes(x=loopred, y=y)) +
  stat_smooth(method='glm', formula = y ~ ns(x, 5), fullrange=TRUE, color="deeppink") +
  geom_abline(linetype = 'dashed') +
  labs(x = "Predicted (LOO)",y = "Observed",title = "Nonlinear model - Calibration plot")
  geom_jitter(height=0.02, width=0, alpha=0.05) +
  scale_y_continuous(breaks=seq(0,1,by=0.1)) +
  xlim(c(0,1)) +
  theme_minimal()

grid.arrange(cplot_linear, cplot_nonlinear, ncol=2)
```

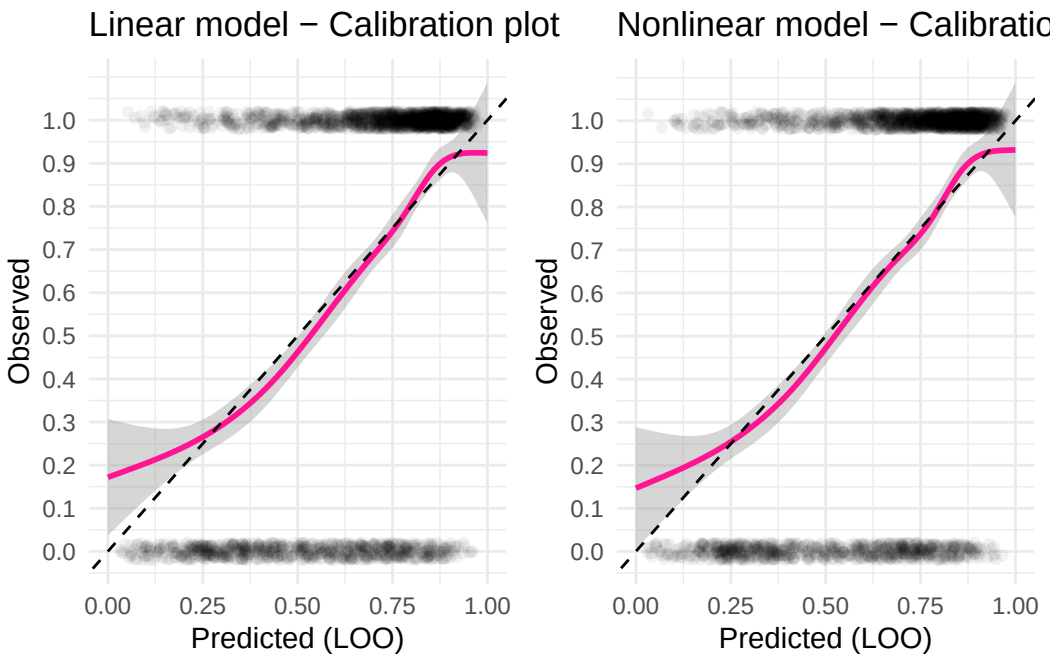


Figure 4: TBD X

6 Discussion

- Problems, potential improvements?
- Conclusion of analysis

7 Lessons Learned

8 References