Appendix: Codes and extra material

1 Credit Card Data

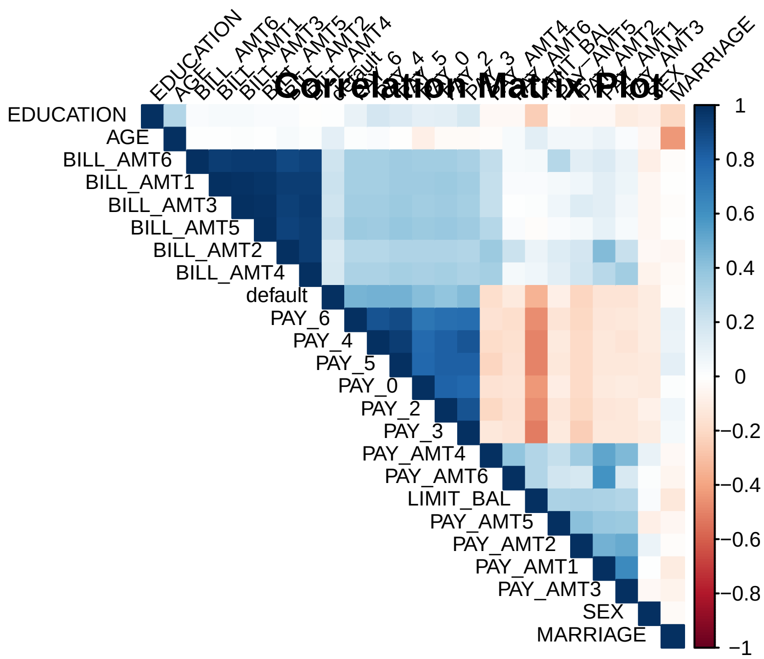
1.1 data cleaning and extraction of small subset

```
set.seed(123)

file_path_credit <- "default-credit-card.csv"
file_path_credit_names <- "names-default-credit-card.csv"
data_credit <- read.csv(file_path_credit)
data_credit_names <- read.csv(file_path_credit_names)

credit_short_colnames <- colnames(data_credit)
credit_long_colnames <- colnames(data_credit_names)
colnames(data_credit) <- credit_long_colnames

data_credit_unclean=data_credit

data_credit <- data_credit %>%
  filter(PAY_0 != -2) %>% filter(PAY_0 != 0) %>%
  filter(PAY_2 != -2)  %>% filter(PAY_2 != 0)  %>%
  filter(PAY_3 != -2)  %>% filter(PAY_3 != 0)  %>%
  filter(PAY_4 != -2)  %>% filter(PAY_4 != 0)  %>%
  filter(PAY_5 != -2)  %>% filter(PAY_5 != 0)  %>%
  filter(PAY_6 != -2)  %>% filter(PAY_6 != 0)  %>%
  filter(EDUCATION > 0) %>% filter(EDUCATION < 4)

data_credit_edu <-function(edu,def){data_credit %>% filter(EDUCATION == edu) %>% filter(default == d

combined_df <- rbind(data_credit_edu(1,0), data_credit_edu(1,1),
                     data_credit_edu(2,0), data_credit_edu(2,1),
                     data_credit_edu(3,0), data_credit_edu(3,1))

write.csv(combined_df, file = "credit_data_education_short.csv")

data_credit_full = data_credit
data_credit = combined_df
```

1.2 correlations in subset

```
colnames(data_credit) <- credit_long_colnames
df = data_credit
correlations <- sapply(df[, -which(names(df) == "default")], function(x) biserial.cor(x, df$default)
list_cor = sort(round(abs(correlations),3))
correlations_df <- data.frame(
  Variable = names(list_cor),
  Correlation = as.vector(list_cor)
)

#correlations_df %>% filter(Correlation > 0)
```

```
# Calculate the correlation matrix
cor_matrix <- cor(data_credit[, -1])

# Create a correlation matrix plot
corrplot(cor_matrix, method = "color", type = "upper", order = "hclust", tl.col = "black", tl.srt =
title("Correlation Matrix Plot")
```




```
correlations <- cor(data_credit[, 'default'], data_credit[, -which(names(data_credit) == 'default')]

round(correlations,3)
```

```
         ID LIMIT_BAL    SEX EDUCATION MARRIAGE  AGE PAY_0 PAY_2 PAY_3 PAY_4
[1,] -0.035    -0.344 -0.104         0   -0.015 0.12 0.428 0.393 0.432  0.47
     PAY_5 PAY_6 BILL_AMT1 BILL_AMT2 BILL_AMT3 BILL_AMT4 BILL_AMT5 BILL_AMT6
[1,] 0.475 0.466     0.213     0.164     0.204      0.18     0.223     0.204
     PAY_AMT1 PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5 PAY_AMT6
[1,]    -0.14   -0.218   -0.147   -0.179   -0.086   -0.118
```

1.3 Stan model PAY_0 + PAY_2

```r
#Time  1.36 mins
start.time <- Sys.time()
fit1 <- brm(default ~ PAY_0 + PAY_2,
            data = data_credit,
            refresh = 0,
            prior=c(
              prior(normal(0,100), class="Intercept")
              , prior(normal(0, 100), class = b)
            ),
            family = bernoulli(),
           file = "pooled",
           backend = "cmdstanr",
           seed = 123
)
end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken
```

```
Time difference of 0.01 secs
```

```r
predictions <- predict(fit1, newdata = data_credit, type = "response")
binary_predictions <- ifelse(predictions[,1] > 0.5, 1, 0)
conf_matrix <- confusionMatrix(factor(binary_predictions), factor(data_credit$default))
conf_matrix
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 94 36
         1 26 84

               Accuracy : 0.7417
                 95% CI : (0.6814, 0.7958)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 1.785e-14

                  Kappa : 0.4833

 Mcnemar's Test P-Value : 0.253

            Sensitivity : 0.7833
            Specificity : 0.7000
         Pos Pred Value : 0.7231
         Neg Pred Value : 0.7636
             Prevalence : 0.5000
         Detection Rate : 0.3917
   Detection Prevalence : 0.5417
      Balanced Accuracy : 0.7417
```

```
      'Positive' Class : 0
```

```r
start.time <- Sys.time()
batch_size = 1000;
start = 1;
binary_predictions_unclean = c();
predictions_batch_list = c();
for (i in 1:30) {
  if (i%%10 == 0) {
    print(i);
  }
  end = start + batch_size - 1;
  predictions_batch <- predict(fit1, newdata = data_credit_unclean[start:end,],
                type = "response", allow_new_levels = TRUE)
  predictions_batch_list = c(predictions_batch_list, predictions_batch)
  start = end + 1
  binary_predictions_batch <- ifelse(predictions_batch[,1] > 0.5, 1, 0)
  binary_predictions_unclean = c(binary_predictions_unclean, binary_predictions_batch);
}
```

```
[1] 10
[1] 20
[1] 30
```

```r
end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken
```

```
Time difference of 45.19 secs
```

```r
conf_matrix <- confusionMatrix(factor(binary_predictions_unclean), reference = factor(data_credit_un
conf_matrix
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 20637  3452
         1  2727  3184

               Accuracy : 0.794
                 95% CI : (0.7894, 0.7986)
    No Information Rate : 0.7788
    P-Value [Acc > NIR] : 7.634e-11

                  Kappa : 0.3779

 Mcnemar's Test P-Value : < 2.2e-16
```

```
            Sensitivity : 0.8833
            Specificity : 0.4798
         Pos Pred Value : 0.8567
         Neg Pred Value : 0.5387
             Prevalence : 0.7788
         Detection Rate : 0.6879
   Detection Prevalence : 0.8030
      Balanced Accuracy : 0.6815

       'Positive' Class : 0
```

1.4 Stan model EDUCATION Hierarchical

```r
start.time <- Sys.time()
fit2 <- brm(default ~  PAY_0 + PAY_2 + (1| EDUCATION),   # short dataset
            data = data_credit,
            refresh = 0,
            prior=c(
                prior(normal(0,100),  class="Intercept"),
                prior(normal(0,100),  class="b"),
                prior(exponential(.02),  class="sd")
            ),
            family = bernoulli(),
            file = "model2_education_small_data_simple",
            backend = "cmdstanr",
            iter = 5000,
            warmup = 2500,
            seed = 123
)
end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken
```

```
Time difference of 0.02 secs
```

```r
predictions <- predict(fit2, newdata = data_credit, type = "response", allow_new_levels = TRUE)
binary_predictions <- ifelse(predictions[,1] > 0.5, 1, 0)
conf_matrix <- confusionMatrix(factor(binary_predictions), factor(data_credit$default))
conf_matrix
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 95 37
         1 25 83

               Accuracy : 0.7417
                 95% CI : (0.6814, 0.7958)
    No Information Rate : 0.5
```

```
   P-Value [Acc > NIR] : 1.785e-14

                  Kappa : 0.4833

 Mcnemar's Test P-Value : 0.1624

            Sensitivity : 0.7917
            Specificity : 0.6917
         Pos Pred Value : 0.7197
         Neg Pred Value : 0.7685
             Prevalence : 0.5000
         Detection Rate : 0.3958
   Detection Prevalence : 0.5500
      Balanced Accuracy : 0.7417

       'Positive' Class : 0
```

```r
start.time <- Sys.time()
batch_size = 1000;
start = 1;
binary_predictions_unclean = c();
predictions_batch_list = c();
for (i in 1:30) {
  if (i%%10 == 0) {
    print(i);
  }
  end = start + batch_size - 1;
  predictions_batch <- predict(fit2, newdata = data_credit_unclean[start:end,],
                type = "response", allow_new_levels = TRUE)
  predictions_batch_list = c(predictions_batch_list, predictions_batch)
  start = end + 1
  binary_predictions_batch <- ifelse(predictions_batch[,1] > 0.5, 1, 0)
  binary_predictions_unclean = c(binary_predictions_unclean, binary_predictions_batch);
}
```

```
[1] 10
[1] 20
[1] 30
```

```r
end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken
```

```
Time difference of 1.28 mins
```

```r
confusionMatrix(factor(binary_predictions_unclean), reference = factor(data_credit_unclean$default))
```

```
Confusion Matrix and Statistics
```

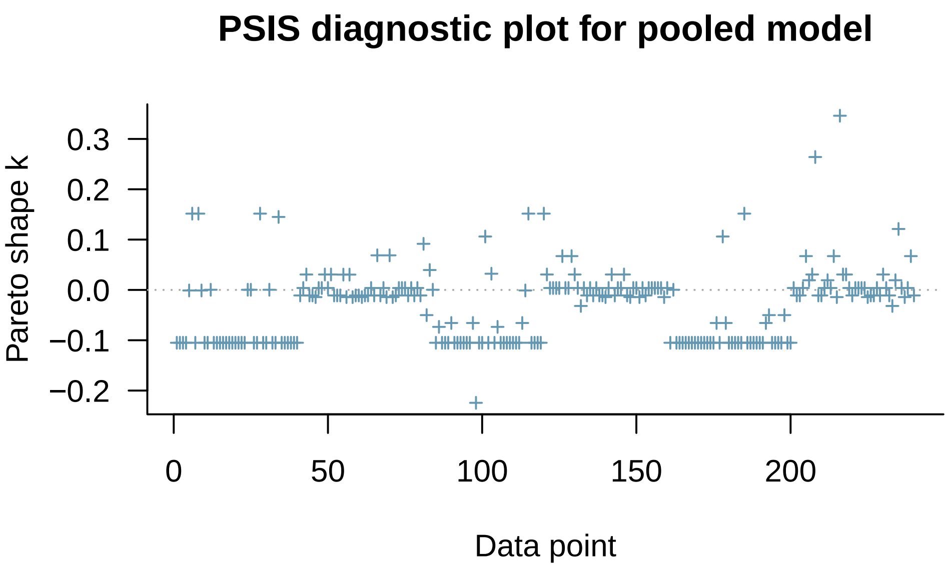
```
          Reference
Prediction     0     1
         0 20409  3372
         1  2955  3264

               Accuracy : 0.7891
                 95% CI : (0.7844, 0.7937)
    No Information Rate : 0.7788
    P-Value [Acc > NIR] : 8.003e-06

                  Kappa : 0.3738

 Mcnemar's Test P-Value : 1.696e-07

            Sensitivity : 0.8735
            Specificity : 0.4919
         Pos Pred Value : 0.8582
         Neg Pred Value : 0.5248
             Prevalence : 0.7788
         Detection Rate : 0.6803
   Detection Prevalence : 0.7927
      Balanced Accuracy : 0.6827

       'Positive' Class : 0
```

1.5 Model comparison (e.g. with LOO-CV).

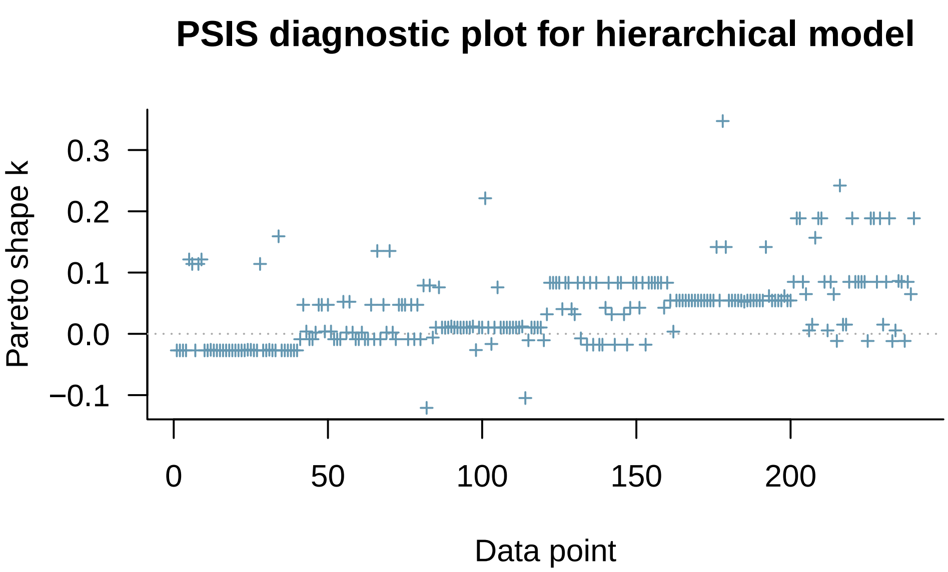
```r
l1 = loo(fit1)
l2 = loo(fit2)
loo_compare(l1,l2)
```

```
     elpd_diff se_diff
fit1  0.0       0.0   
fit2 -0.4       1.0   
```

```r
plot(l1, label_points = TRUE, main = 'PSIS diagnostic plot for pooled model')
```




```r
plot(l2, label_points = TRUE, main = 'PSIS diagnostic plot for hierarchical model')
```



1.6 Convergence diagnostic

```r
summarize_draws(fit1)
```

```
# A tibble: 5 x 10
  variable      mean   median     sd   mad       q5     q95  rhat ess_bulk
  <chr>        <dbl>    <dbl>  <dbl> <dbl>    <dbl>   <dbl> <dbl>    <dbl>
1 b_Intercept -0.213   -0.212 0.150  0.151 -4.58e-1 2.94e-2  1.00    2592.
2 b_PAY_0      0.521    0.518 0.154  0.157  2.79e-1 7.81e-1  1.00    2058.
3 b_PAY_2      0.165    0.165 0.131  0.131 -5.02e-2 3.85e-1  1.00    2075.
```

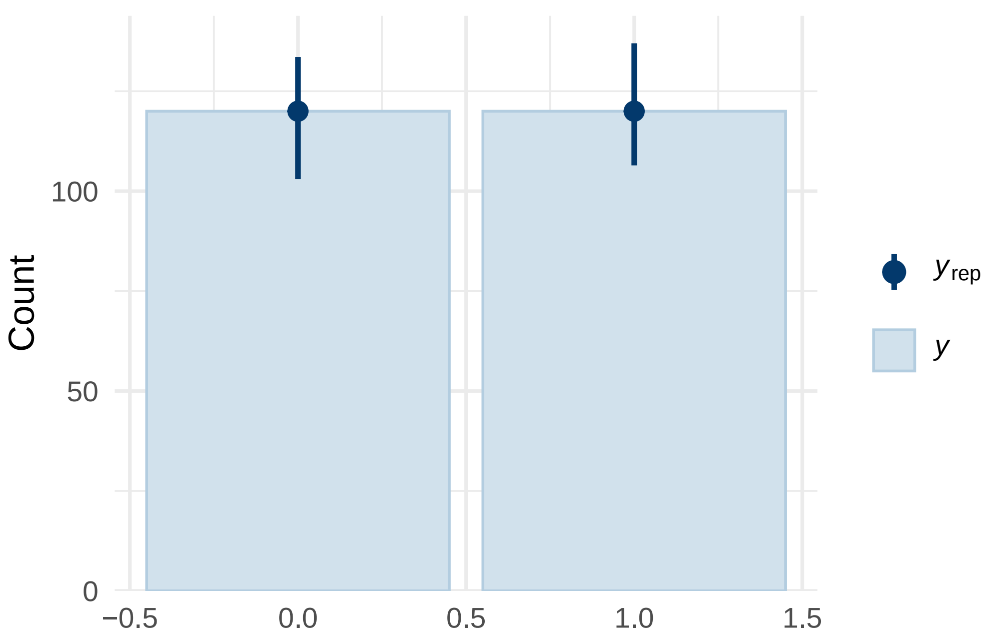
```
4 lprior       -16.6    -16.6    0.0000398 0      -1.66e+1 -1.66e+1  1.00    2418.
5 lp__        -158.    -158.     1.21      0.981 -1.61e+2 -1.57e+2  1.00    1876.
# i 1 more variable: ess_tail <dbl>
```

```
summarize_draws(fit2)
```

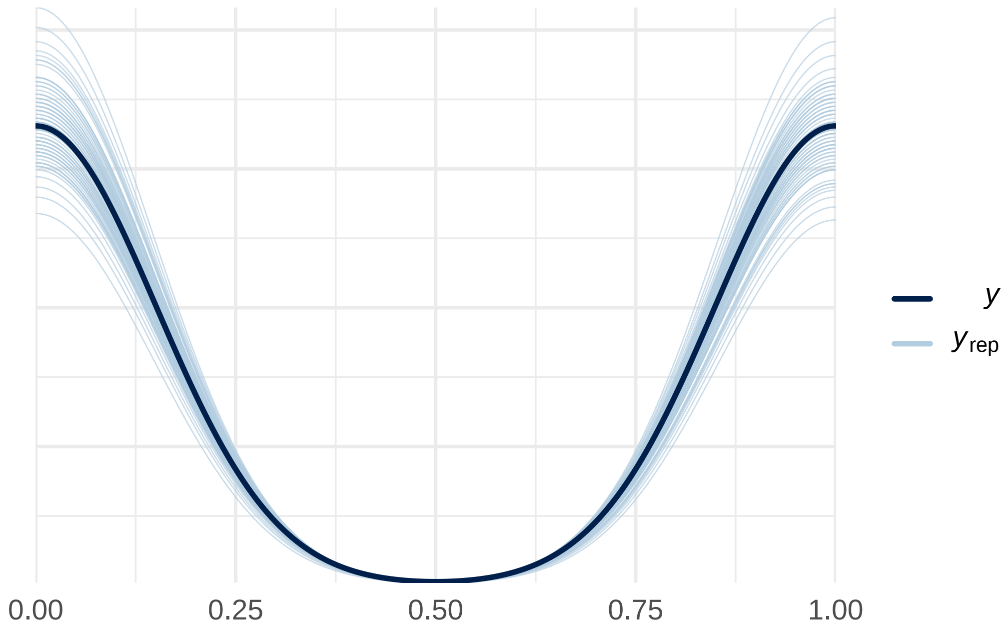
```
# A tibble: 9 x 10
  variable          mean   median     sd     mad       q5      q95  rhat ess_bulk
  <chr>            <dbl>    <dbl>  <dbl>   <dbl>    <dbl>    <dbl> <dbl>    <dbl>
1 b_Intercept  -2.49e-1 -2.25e-1 0.364  0.269   -9.08e-1    0.293  1.00     1622.
2 b_PAY_0       5.51e-1  5.48e-1 0.157  0.156    2.95e-1    0.811  1.00     3162.
3 b_PAY_2       1.64e-1  1.65e-1 0.130  0.130   -4.99e-2    0.380  1.00     3102.
4 sd_EDUCATIO~  5.73e-1  3.90e-1 0.605  0.368    4.01e-2    1.69   1.01     1170.
5 r_EDUCATION~  2.08e-1  1.25e-1 0.392  0.258   -3.01e-1    0.982  1.01     1601.
6 r_EDUCATION~ -1.61e-1 -1.16e-1 0.384  0.256   -8.30e-1    0.407  1.00     1859.
7 r_EDUCATION~  2.89e-2  5.26e-3 0.377  0.235   -5.62e-1    0.699  1.00     1774.
8 lprior       -2.05e+1 -2.05e+1 0.0121 0.00741 -2.05e+1  -20.5     1.01     1167.
9 lp__         -1.67e+2 -1.66e+2 2.33   2.26    -1.71e+2 -163.      1.01     1550.
# i 1 more variable: ess_tail <dbl>
```

#PPC

```
pp_check(fit1, type = "bars", ndraws = 150)
```




```
pp_check(fit1, type = "dens_overlay", ndraws = 100)
```

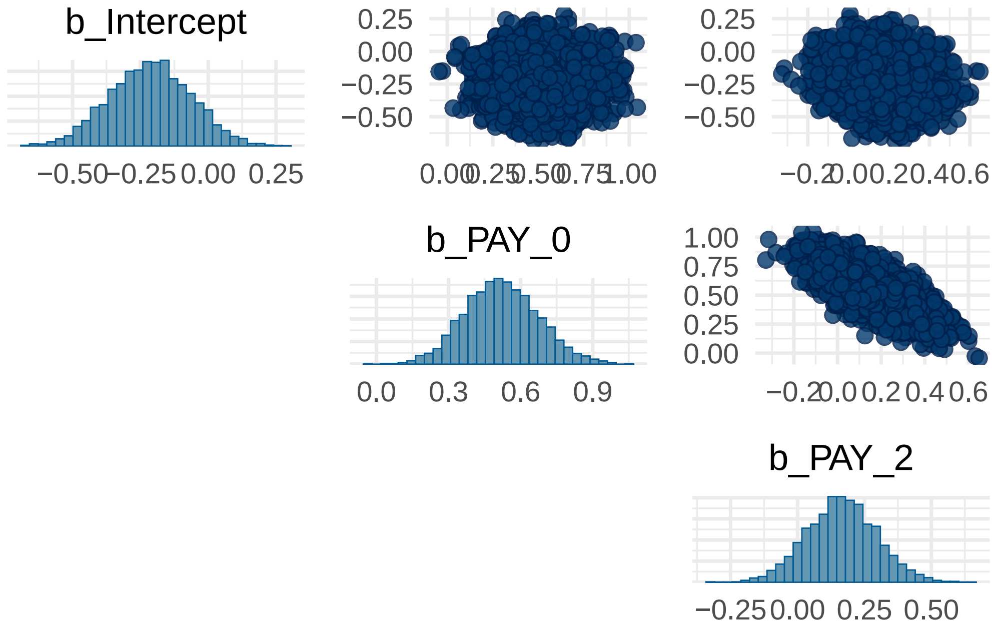



```r
posterior_samples <- posterior_samples(fit1)
```

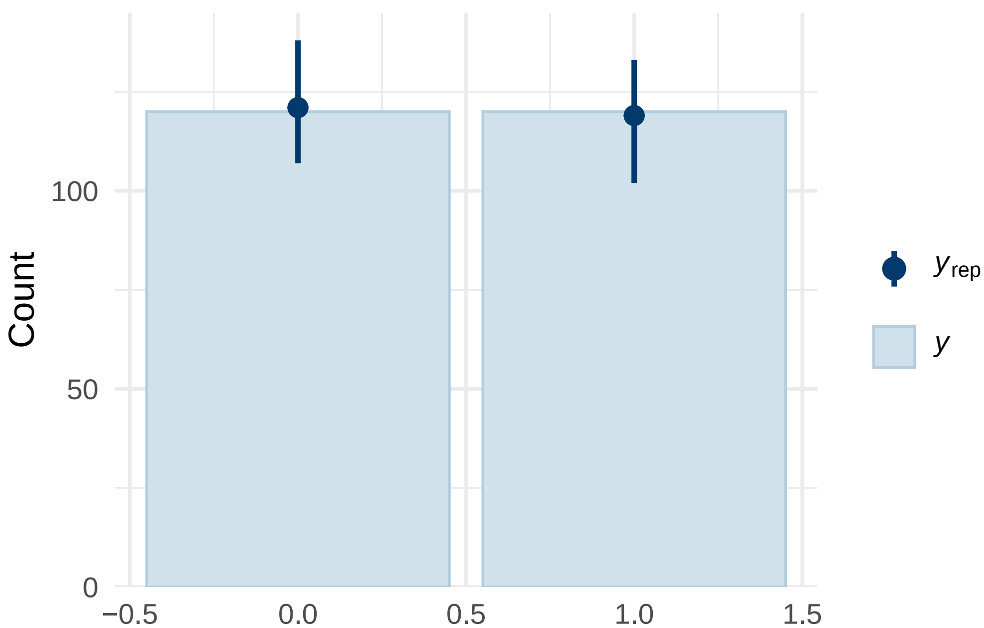
```
Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
recommended alternatives.
```

```r
# Use mcmc_pairs to create a scatterplot matrix for selected parameters
# Select a subset of parameters if you have many;
selected_params <- posterior_samples[, c("b_Intercept", "b_PAY_0", "b_PAY_2")]
np <- nuts_params(fit1)


# Create the pairs plot
mcmc_pairs(
  selected_params,
  np = np,
  pars = c("b_Intercept", "b_PAY_0", "b_PAY_2")
)
```

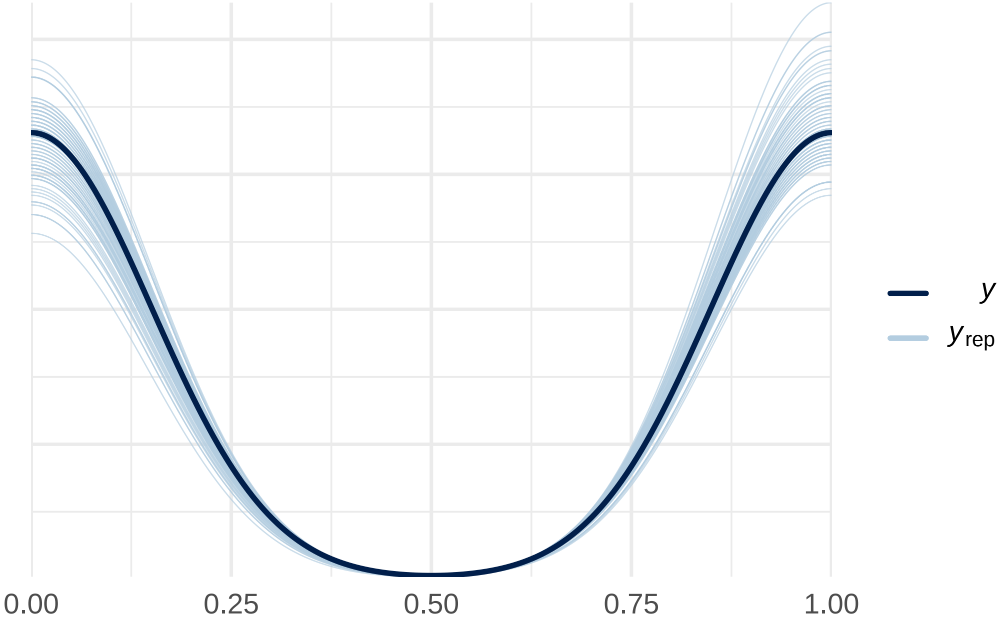
```
Warning: Only one chain in 'x'. This plot is more useful with multiple chains.
```




```r
# Bar plot of the observed outcomes compared to the replicated datasets
pp_check(fit2, type = "bars", ndraws = 100)
```




```r
# Density plot of the predicted probabilities
pp_check(fit2, type = "dens_overlay", ndraws = 100)
```

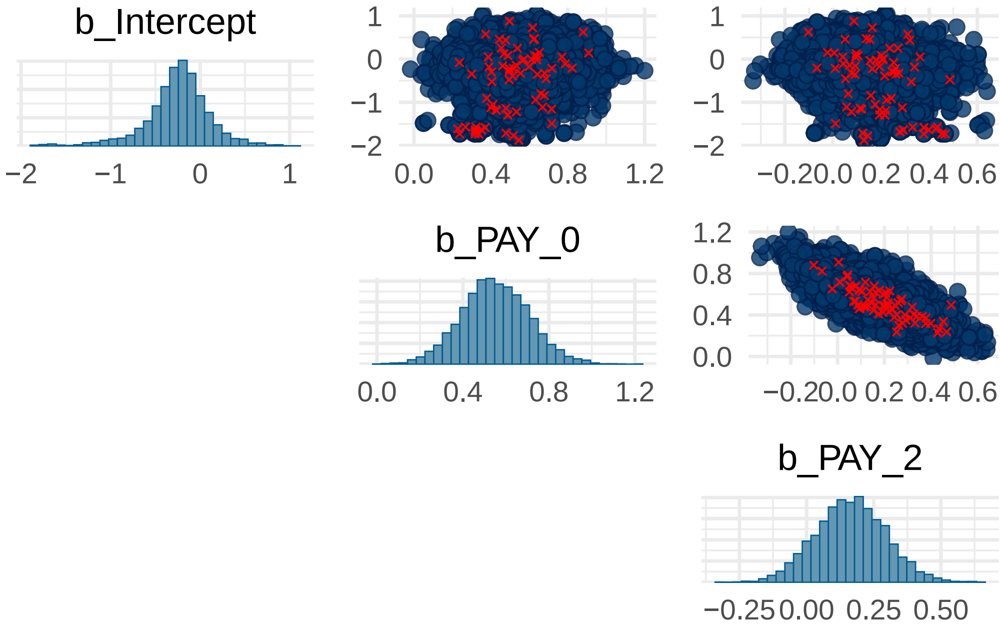



```
posterior_samples <- posterior_samples(fit2)
```

```
Warning: Method 'posterior_samples' is deprecated. Please see ?as_draws for
recommended alternatives.
```

```
# Use mcmc_pairs to create a scatterplot matrix for selected parameters
# Select a subset of parameters if you have many;
selected_params <- posterior_samples[, c("b_Intercept", "b_PAY_0", "b_PAY_2")]
np <- nuts_params(fit2)

# Create the pairs plot
mcmc_pairs(
  selected_params,
  np = np,
  pars = c("b_Intercept", "b_PAY_0", "b_PAY_2")
)
```

```
Warning: Only one chain in 'x'. This plot is more useful with multiple chains.
```



b_Intercept
b_PAY_0
b_PAY_2
−2
−1
0
1
0.0
0.4
0.8
1.2
−0.20.0
0.2
0.4
0.6
−0.25
0.00
0.25
0.50