STATS115_preliminary_analysis

Seiya Uno

Preliminary Analysis

```
Warning: package 'Hmisc' was built under R version 4.3.3
```

```
Warning: package 'corrplot' was built under R version 4.3.3
```

```
df <- read.csv("./data/binary_diabetes.csv")
sample_n(df, 3)
```

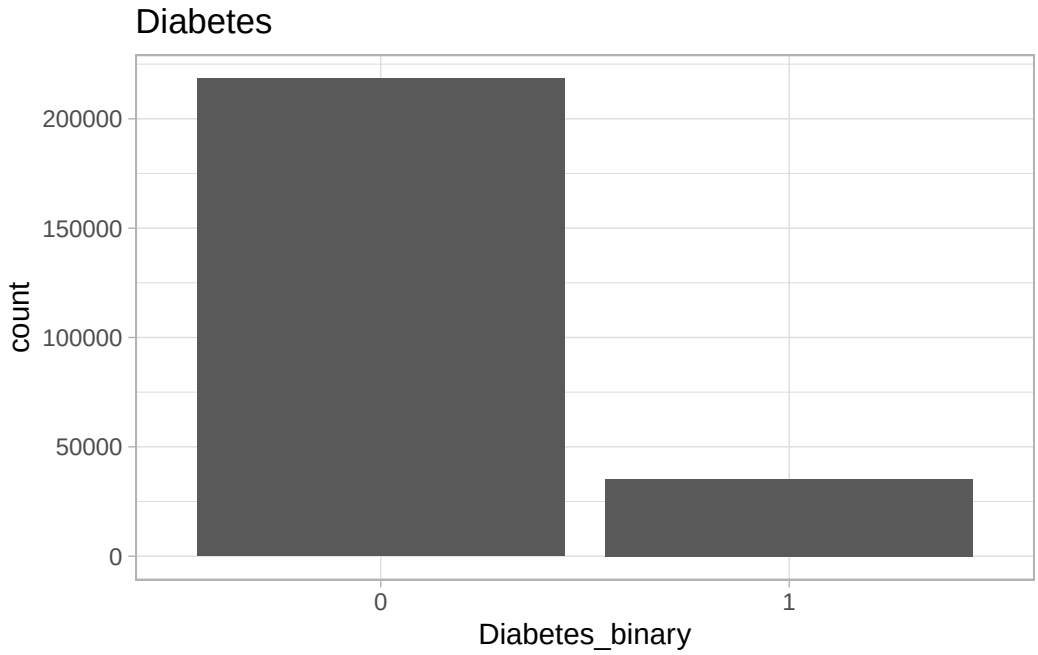
```
  Diabetes_binary HighBP HighChol CholCheck BMI Smoker Stroke
1               0      1        1         1  30      1      0
2               0      1        1         1  29      0      0
3               0      1        1         1  24      0      1
  HeartDiseaseorAttack PhysActivity Fruits Veggies HvyAlcoholConsump
1                    0            1      1       0                 0
2                    0            1      1       1                 0
3                    0            1      0       1                 0
  AnyHealthcare NoDocbcCost GenHlth MentHlth PhysHlth DiffWalk Sex Age
1             1           0       3        0        0        0   1  10
2             1           0       3        0       21        0   0   8
3             1           0       3        0        0        0   1  13
  Education Income
1         5      3
2         6      8
3         4      4
```

```
nrow(df)
```

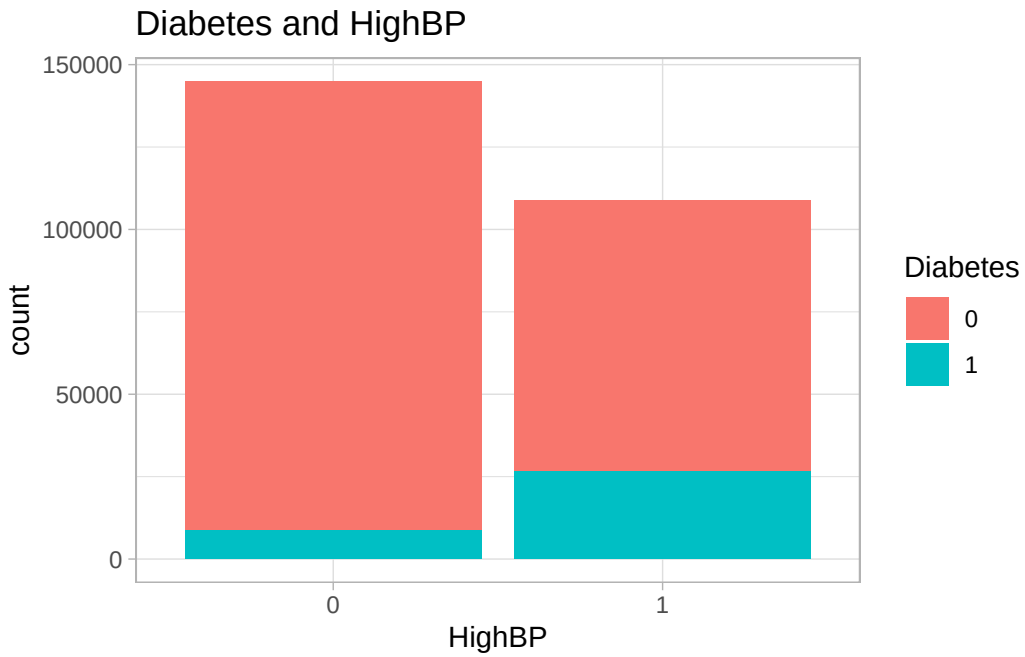
```
[1] 253680
```

```r
df_orig <- df
cols <- c("Diabetes_binary","HighBP","HighChol","CholCheck","Smoker","Stroke","HeartDiseas
df[cols] <- lapply(df[cols], factor)
```

```r
ggplot(df, aes(x=Diabetes_binary)) +
  geom_bar() +
  theme_light() +
  labs(title="Diabetes")
```




```r
ggplot(df, aes(x=HighBP, fill=Diabetes_binary, group=Diabetes_binary)) +
  geom_bar() +
  theme_light() +
  labs(title="Diabetes and HighBP", x="HighBP",
       fill="Diabetes")
```



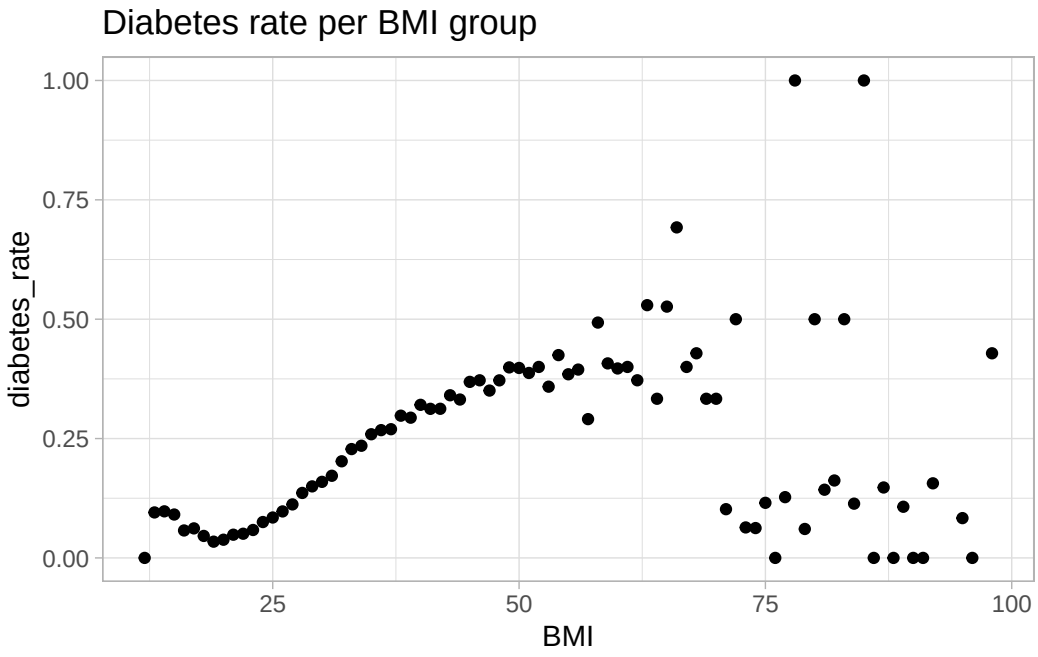

```
# cor() function want nuimerical values, not factors. Hence, we pass original df here.
cor_mtx <- cor(df_orig)
#corrplot(cor_mtx, type="upper", order = "hclust", tl.cex = 0.6, )
```

It seems that BMI, DiffWalk, GenHealth, PhysHealth, HighCol, HighBP, Age, and HeartDiseaseAttack are positively correlated with Diabetes_binary. DiffWalk, GenHealth, and PhysHealth are correlated with each other; hence we may not need all of them.

For negative correlations, PhysActivity, Education, and Income seem more significant than other variables.

```
# ggplot(df, aes(x=BMI, fill=Diabetes_binary, group=Diabetes_binary)) +
#   geom_bar(position = "fill") +
#   theme_light() +
#   labs(title="Diabetes and HighBP", x="BMI",
#        fill="Diabetes")

diabetes_rate_df <- df %>% group_by(BMI) %>%
                      summarise(diabetes_rate = mean(Diabetes_binary==1))

ggplot(diabetes_rate_df, aes(x=BMI, y=diabetes_rate)) +
  geom_point() +
```

```
  theme_light() +
  labs(title="Diabetes rate per BMI group", x="BMI")
```



Model using default prior

Using subset of data since it takes very long time to fit the model with full dataset. Split dataset into 10 chunks

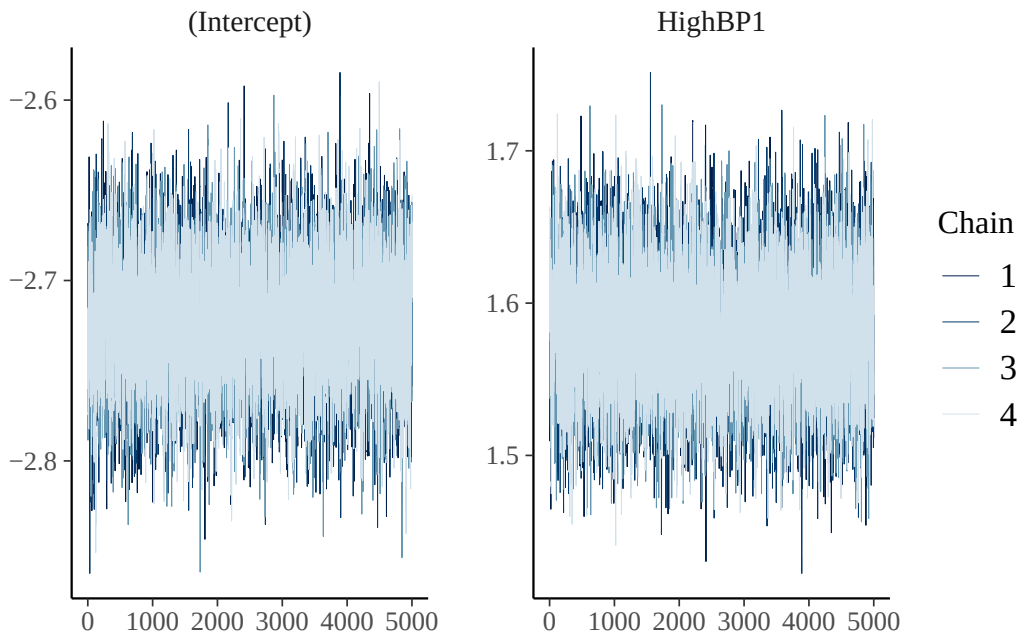
```
adf <- function(x) {
  as.data.frame(x[1])
}

set.seed(RANDOM_STATE)
df <- df[sample(1:nrow(df)), ]
s <- split(df, (seq(nrow(df))-1) %/% floor(nrow(df) * 0.1))
train_df <- s[[1]]
```

HighBP

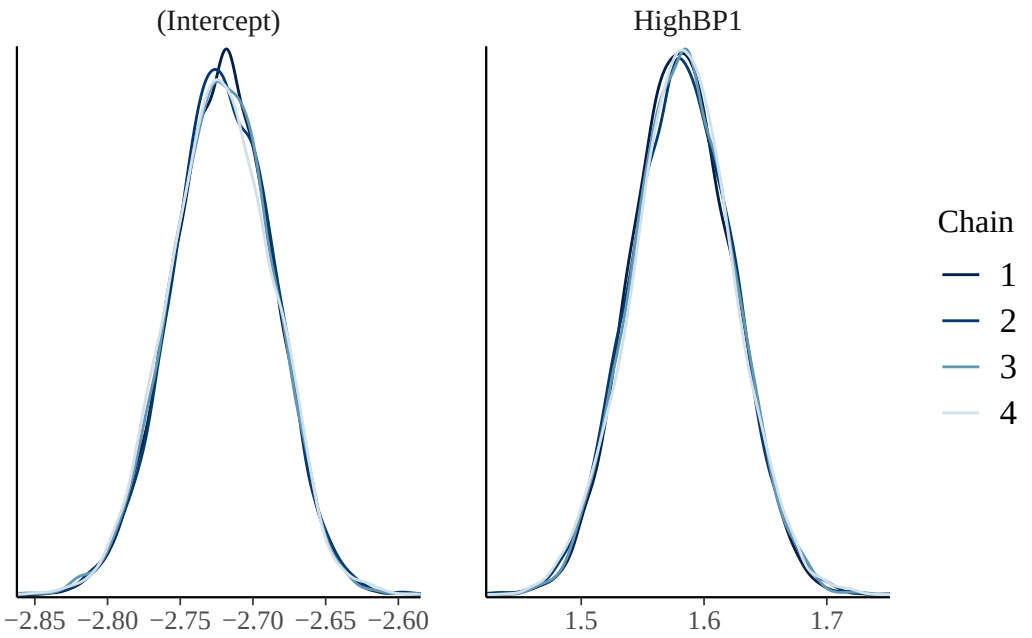
```
cutoff <- 0.15
```

We do not pass prior information to use default prior.

```
diabetes_model_def_HB <- stan_glm(Diabetes_binary ~ HighBP,
                                  data = train_df,
                                  family = binomial,
                                  chains = 4, iter = 5000*2,
                                  seed = RANDOM_STATE,
                                  prior_PD = FALSE, refresh=FALSE)
mcmc_trace(diabetes_model_def_HB, size=0.1)
```

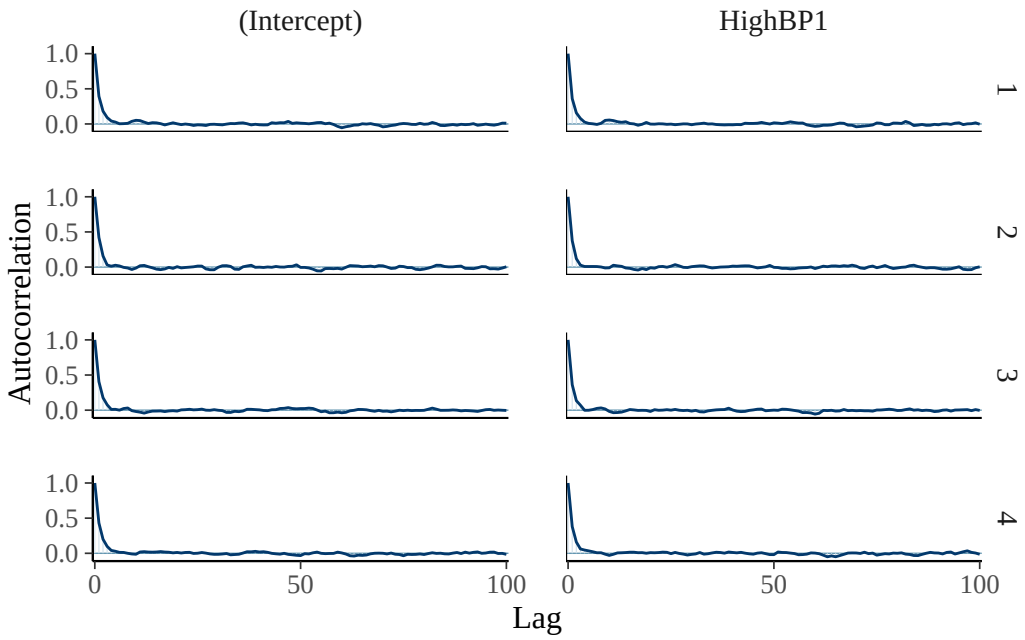



```
mcmc_dens_overlay(diabetes_model_def_HB)
```




```
mcmc_acf(diabetes_model_def_HB, lags = 100)
```

```
Warning: The `facets` argument of `facet_grid()` is deprecated as of ggplot2 2.2.0.
i Please use the `rows` argument instead.
i The deprecated feature was likely used in the bayesplot package.
  Please report the issue at <https://github.com/stan-dev/bayesplot/issues/>.
```




```
posterior_interval(diabetes_model_def_HB, prob = 0.80)
```

```
                  10%       90%
(Intercept) -2.766239 -2.676640
HighBP1      1.529148  1.635685
```

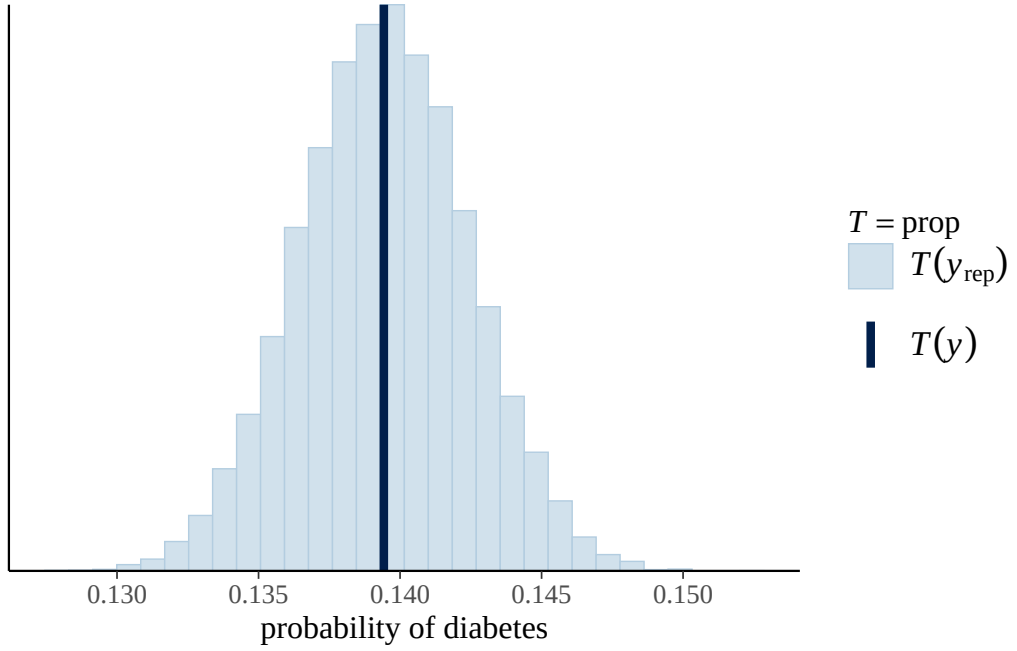
```
exp(posterior_interval(diabetes_model_def_HB, prob = 0.80))
```

```
                  10%        90%
(Intercept) 0.06289813 0.06879391
HighBP1     4.61424249 5.13297190
```

```
prop <- function(x){mean(x == 1)}
pp_check(diabetes_model_def_HB, nreps = 100,
         plotfun = "stat", stat = "prop") +
  xlab("probability of diabetes")
```

```
Warning: 'nreps' is ignored for this PPC
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
get_summary <- function(model, data, cutoff, start, end) {
  res <- c()
  for(i in start:end) {
    res <- append(res, classification_summary(model, data[[i]], cutoff=cutoff))
  }
  return(res)
}
summary_HB <- get_summary(diabetes_model_def_HB, s, 0.15, 2, 6)
summary_HB
```

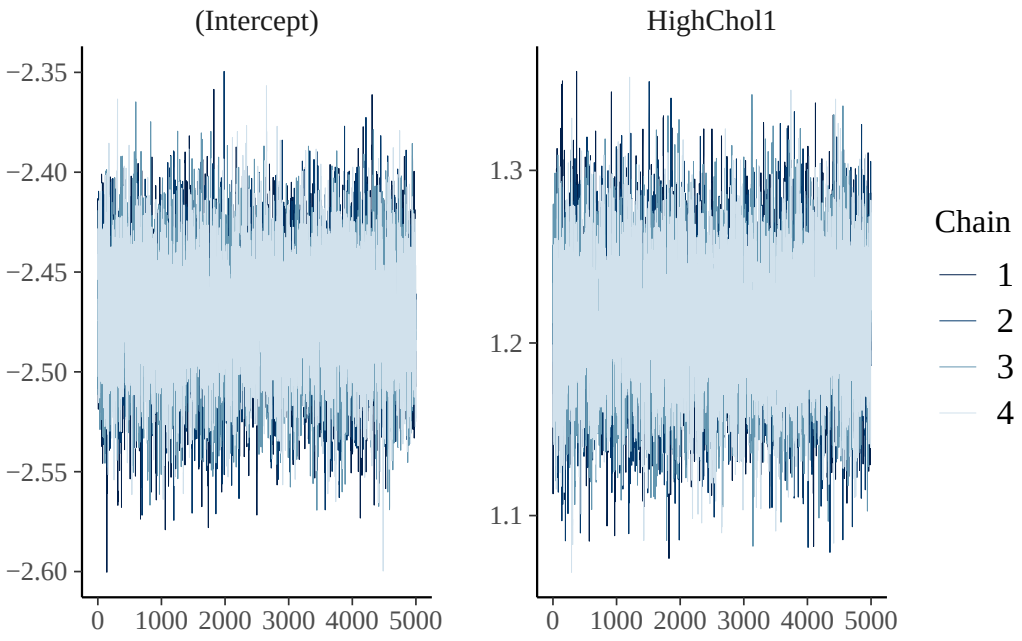
```
$confusion_matrix
 y     0    1
 0 13617 8154
 1   881 2716

$accuracy_rates

sensitivity      0.7550737
specificity      0.6254651
```

```
overall_accuracy 0.6438426

$confusion_matrix
 y     0    1
 0 13735 8003
 1   899 2731

$accuracy_rates

sensitivity      0.7523416
specificity      0.6318429
overall_accuracy 0.6490855

$confusion_matrix
 y     0    1
 0 13627 8180
 1   914 2647

$accuracy_rates

sensitivity      0.7433305
specificity      0.6248911
overall_accuracy 0.6415169

$confusion_matrix
 y     0    1
 0 13485 8345
 1   840 2698

$accuracy_rates

sensitivity      0.7625777
specificity      0.6177279
overall_accuracy 0.6379297

$confusion_matrix
 y     0    1
 0 13529 8305
 1   820 2714

$accuracy_rates

sensitivity      0.7679683
```

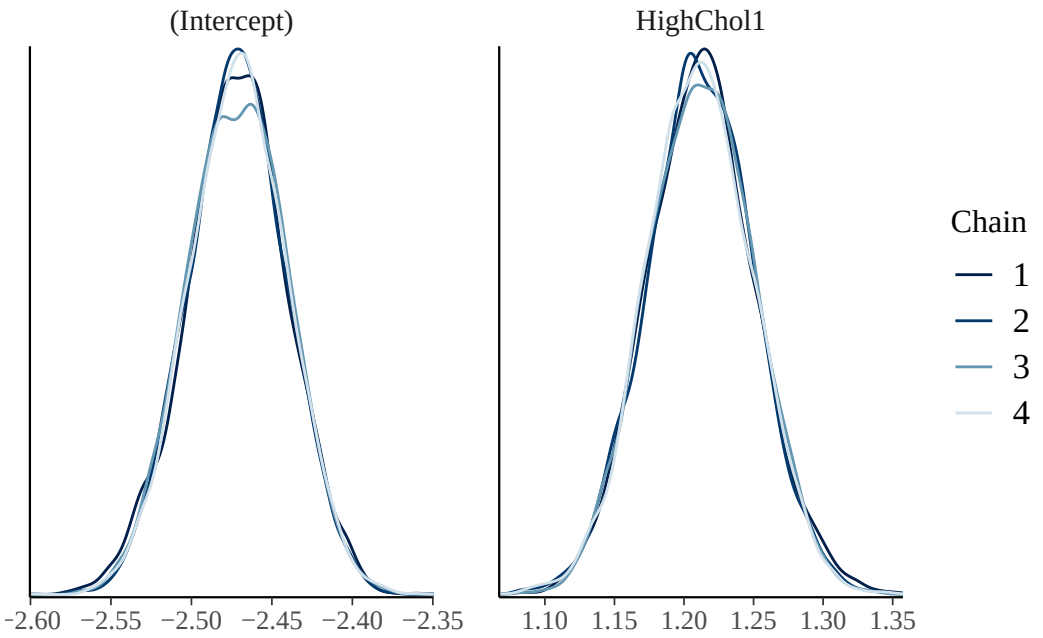
```
specificity      0.6196299
overall_accuracy 0.6402949
```

HighChol

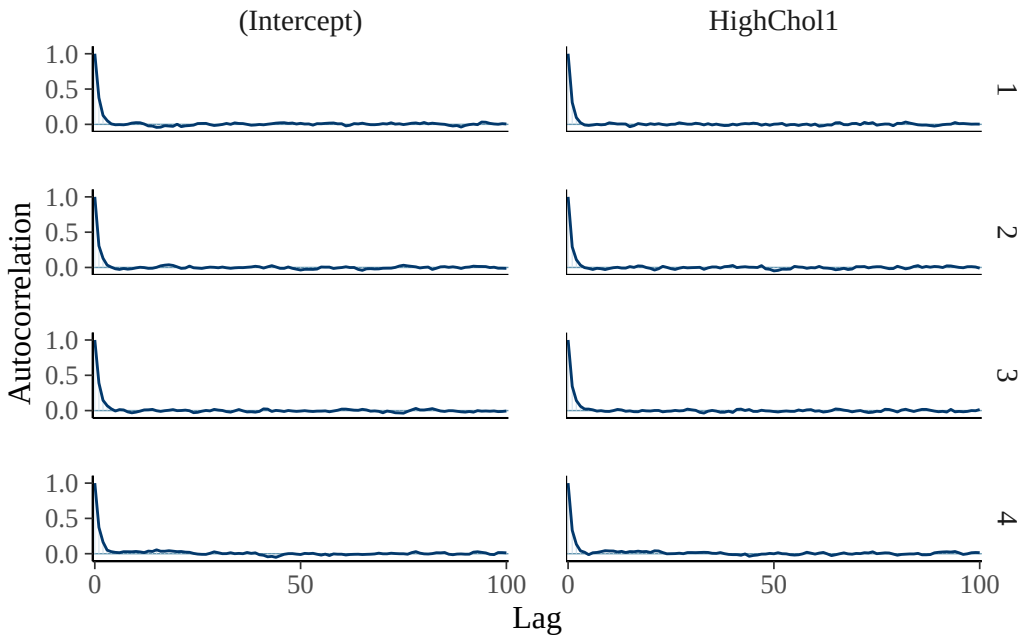
```
diabetes_model_def_HC <- stan_glm(Diabetes_binary ~ HighChol,
                                  data = train_df,
                                  family = binomial,
                                  chains = 4, iter = 5000*2,
                                  seed = RANDOM_STATE,
                                  prior_PD = FALSE, refresh=FALSE)
mcmc_trace(diabetes_model_def_HC, size=0.1)
```




```
mcmc_dens_overlay(diabetes_model_def_HC)
```




```
mcmc_acf(diabetes_model_def_HC, lags = 100)
```



```
posterior_interval(diabetes_model_def_HC, prob = 0.80)
```

```
                 10%       90%
(Intercept) -2.510899 -2.431643
HighChol1    1.163176  1.261122
```

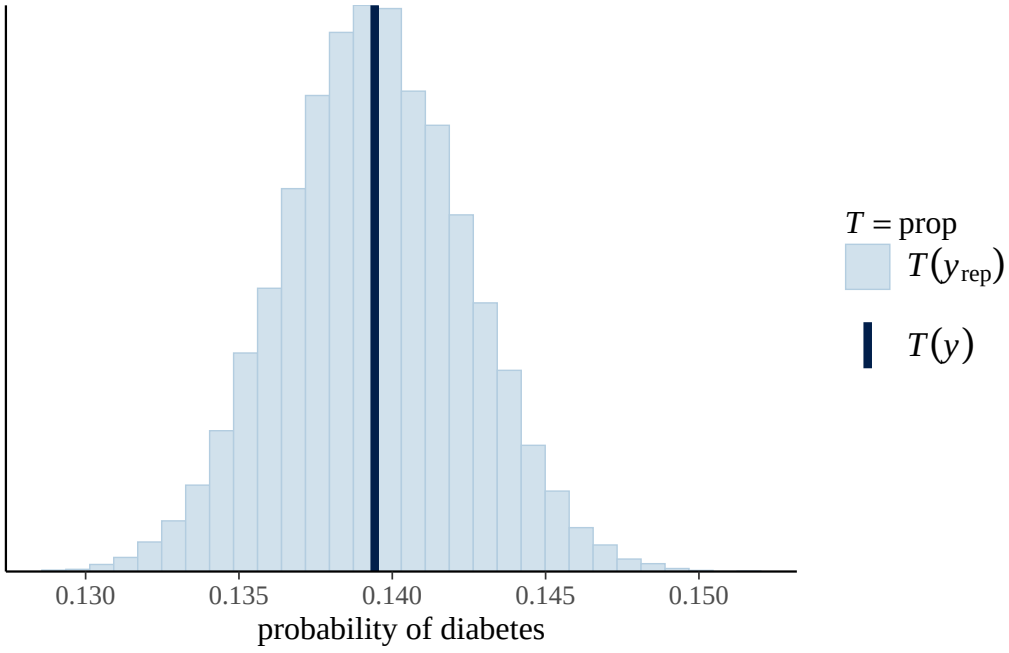
```
exp(posterior_interval(diabetes_model_def_HC, prob = 0.80))
```

```
                 10%        90%
(Intercept) 0.0811952 0.08789234
HighChol1   3.2000805 3.52937789
```

```
pp_check(diabetes_model_def_HC, nreps = 100,
         plotfun = "stat", stat = "prop") +
  xlab("probability of diabetes")
```

```
Warning: 'nreps' is ignored for this PPC
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



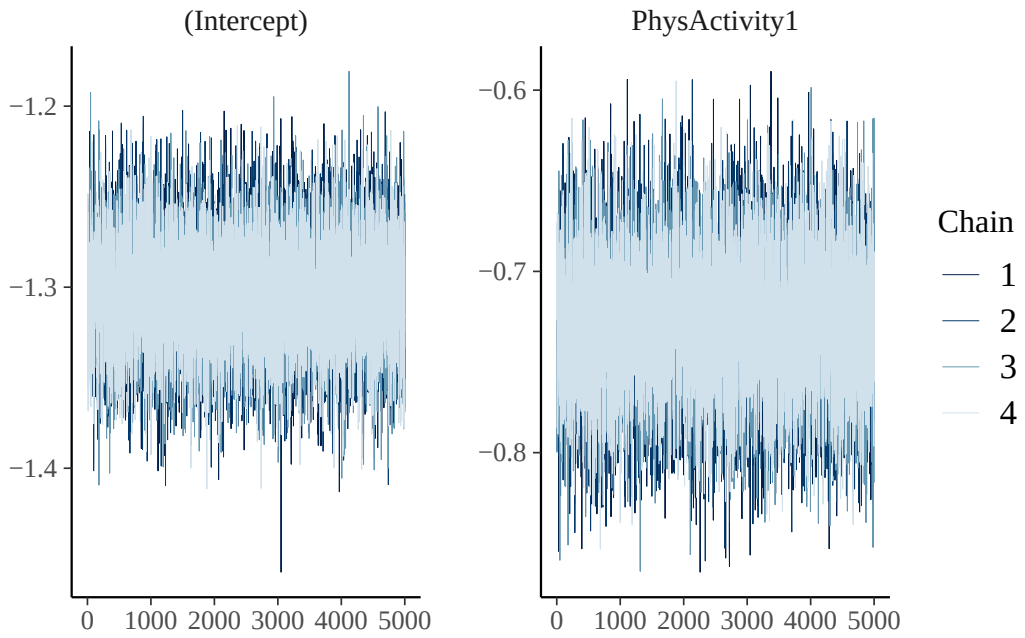
```r
summary_HC <- get_summary(diabetes_model_def_HC, s, cutoff, 2, 6)
summary_HC
```

```
$confusion_matrix
 y     0    1
 0 13386 8385
 1  1162 2435

$accuracy_rates

sensitivity      0.6769530
specificity      0.6148546
overall_accuracy 0.6236597

$confusion_matrix
 y     0    1
 0 13503 8235
 1  1209 2421

$accuracy_rates

sensitivity      0.6669421
specificity      0.6211703
overall_accuracy 0.6277200

$confusion_matrix
 y     0    1
 0 13452 8355
 1  1210 2351

$accuracy_rates

sensitivity      0.6602078
specificity      0.6168661
overall_accuracy 0.6229502

$confusion_matrix
 y     0    1
 0 13355 8475
 1  1168 2370

$accuracy_rates
```

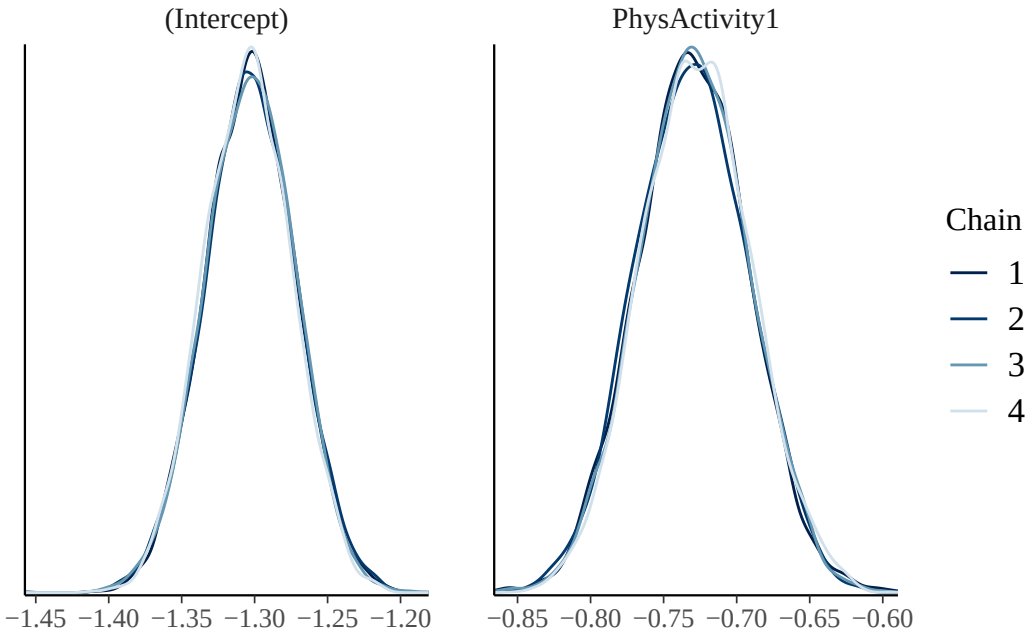
```
sensitivity      0.6698700
specificity      0.6117728
overall_accuracy 0.6198754

$confusion_matrix
 y     0    1
 0 13445 8389
 1  1145 2389

$accuracy_rates

sensitivity      0.6760045
specificity      0.6157827
overall_accuracy 0.6241722
```

PhysActivity

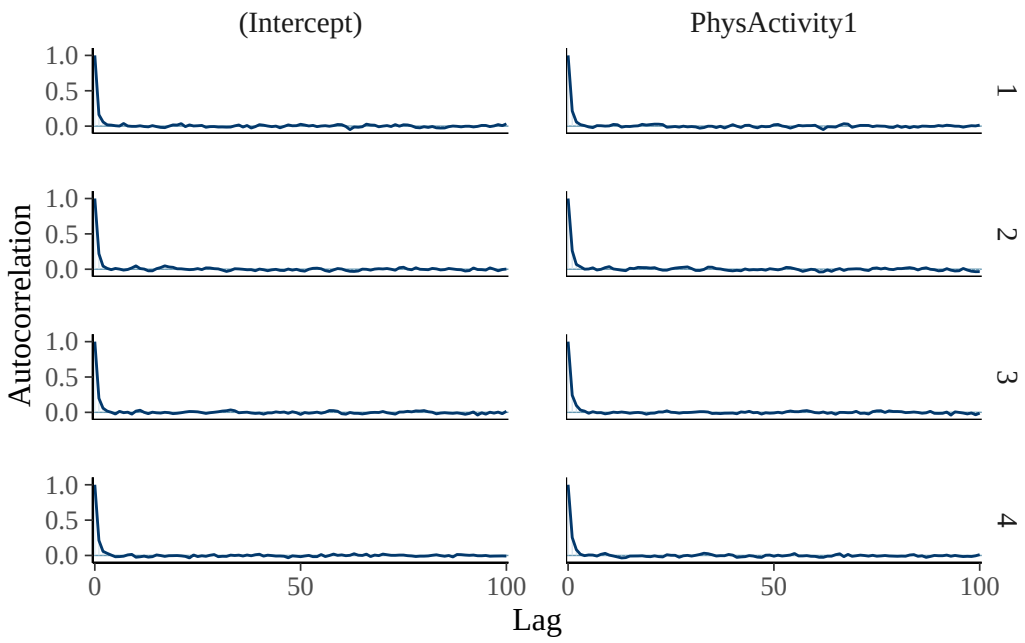
```r
diabetes_model_def_PA  <- stan_glm(Diabetes_binary ~ PhysActivity,
                                   data = train_df,
                                   family = binomial,
                                   chains = 4, iter = 5000*2,
                                   seed = RANDOM_STATE,
                                   prior_PD = FALSE, refresh=FALSE)
mcmc_trace(diabetes_model_def_PA, size=0.1)
```




```
mcmc_dens_overlay(diabetes_model_def_PA)
```



```
mcmc_acf(diabetes_model_def_PA, lags = 100)
```




```
posterior_interval(diabetes_model_def_PA, prob = 0.80)
```

```
                    10%        90%
(Intercept)   -1.3430365 -1.2638132
PhysActivity1 -0.7773386 -0.6799376
```

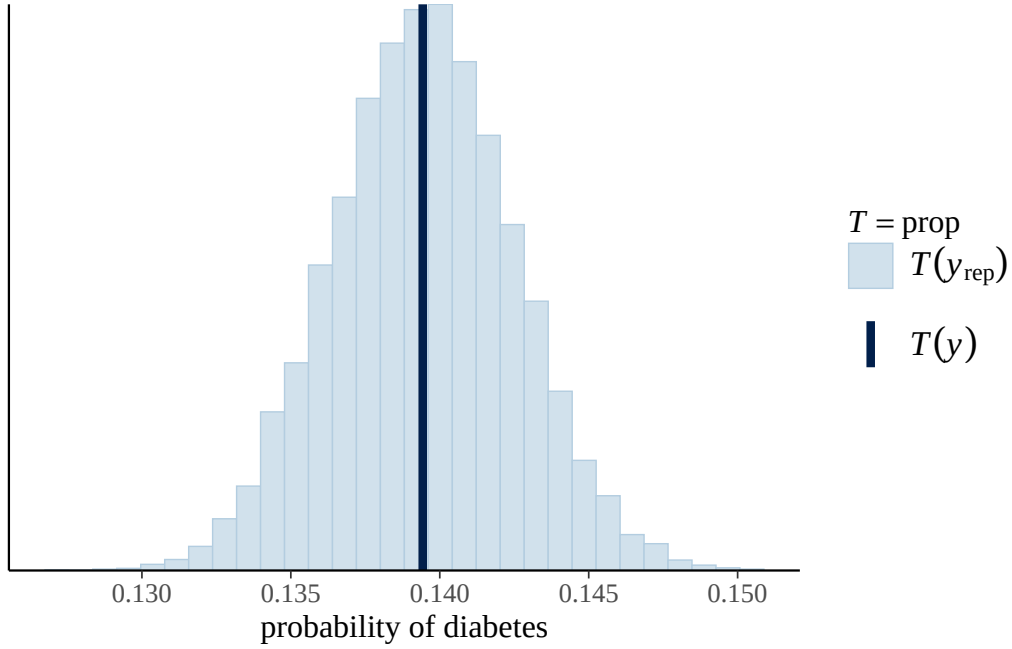
```
exp(posterior_interval(diabetes_model_def_PA, prob = 0.80))
```

```
                   10%       90%
(Intercept)   0.2610518 0.2825745
PhysActivity1 0.4596277 0.5066486
```

```
pp_check(diabetes_model_def_PA, nreps = 100,
         plotfun = "stat", stat = "prop") +
  xlab("probability of diabetes")
```

```
Warning: 'nreps' is ignored for this PPC
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



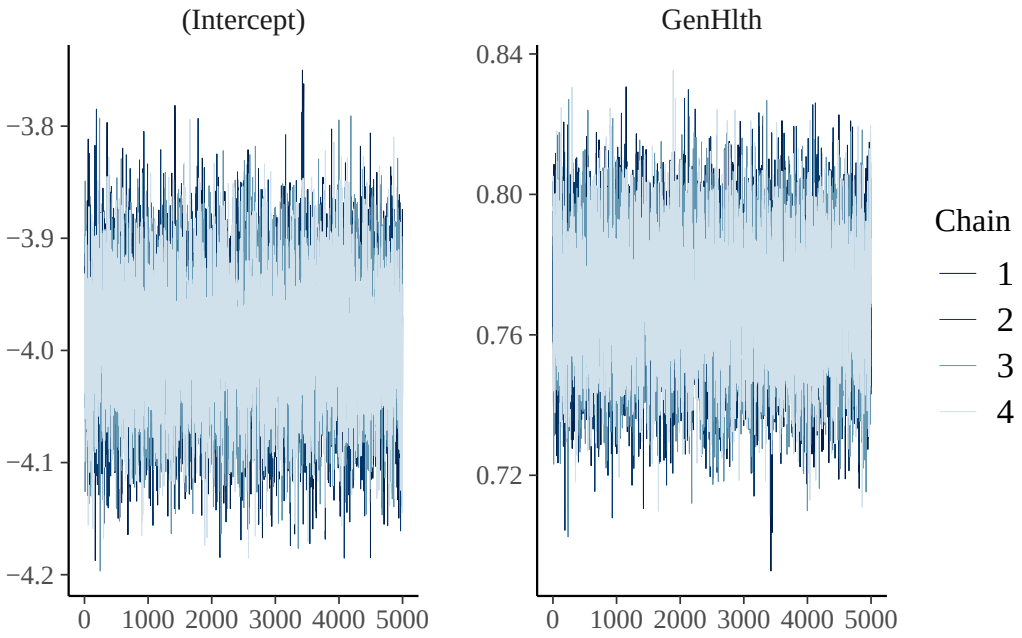

```
summary_PA <- get_summary(diabetes_model_def_PA, s, cutoff, 2, 6)
summary_PA
```

```
$confusion_matrix
 y     0    1
 0 16949 4822
 1  2200 1397

$accuracy_rates

sensitivity      0.3883792
specificity      0.7785127
overall_accuracy 0.7231946

$confusion_matrix
 y     0    1
```

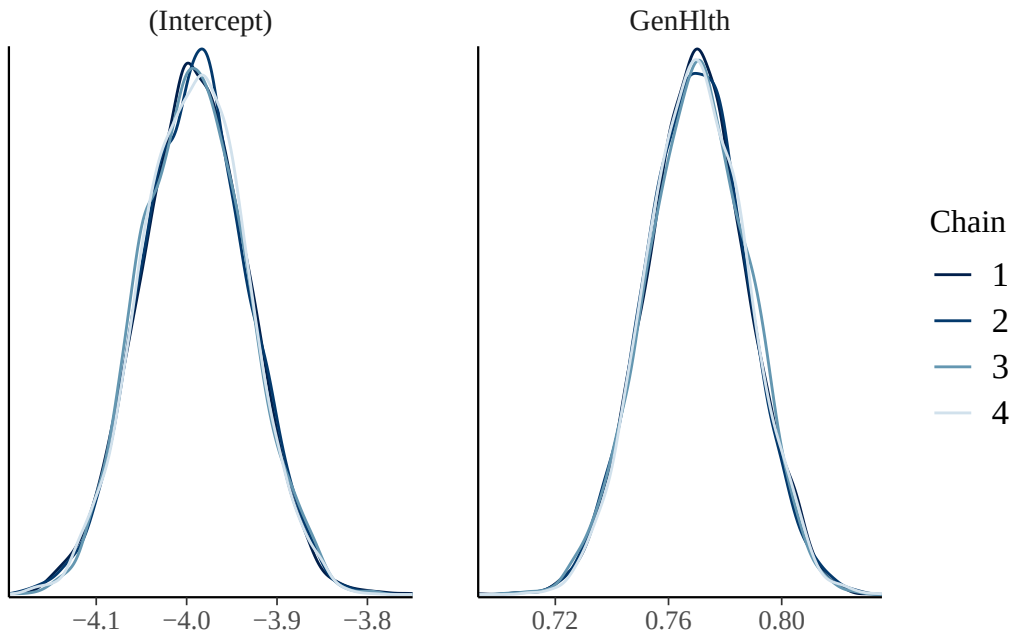
```
 0 16941 4797
 1  2286 1344

$accuracy_rates

sensitivity      0.3702479
specificity      0.7793265
overall_accuracy 0.7207900

$confusion_matrix
 y     0    1
 0 16959 4848
 1  2257 1304

$accuracy_rates

sensitivity      0.3661893
specificity      0.7776861
overall_accuracy 0.7199227

$confusion_matrix
 y     0    1
 0 16872 4958
 1  2237 1301

$accuracy_rates

sensitivity      0.3677219
specificity      0.7728814
overall_accuracy 0.7163750

$confusion_matrix
 y     0    1
 0 16961 4873
 1  2253 1281

$accuracy_rates

sensitivity      0.3624788
specificity      0.7768160
overall_accuracy 0.7190949
```

GenHlth

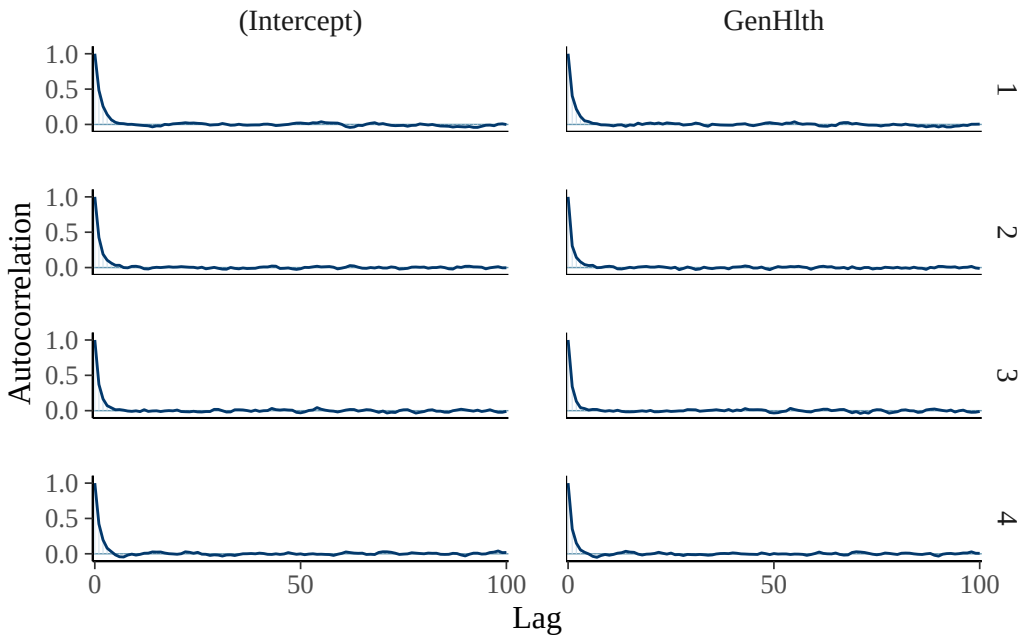
```
diabetes_model_def_GH  <- stan_glm(Diabetes_binary ~ GenHlth,
                                   data = train_df,
                                   family = binomial,
                                   chains = 4, iter = 5000*2,
                                   seed = RANDOM_STATE,
                                   prior_PD = FALSE, refresh=FALSE)
mcmc_trace(diabetes_model_def_GH, size=0.1)
```




```
mcmc_dens_overlay(diabetes_model_def_GH)
```




```
mcmc_acf(diabetes_model_def_GH, lags = 100)
```



```
posterior_interval(diabetes_model_def_GH, prob = 0.80)
```

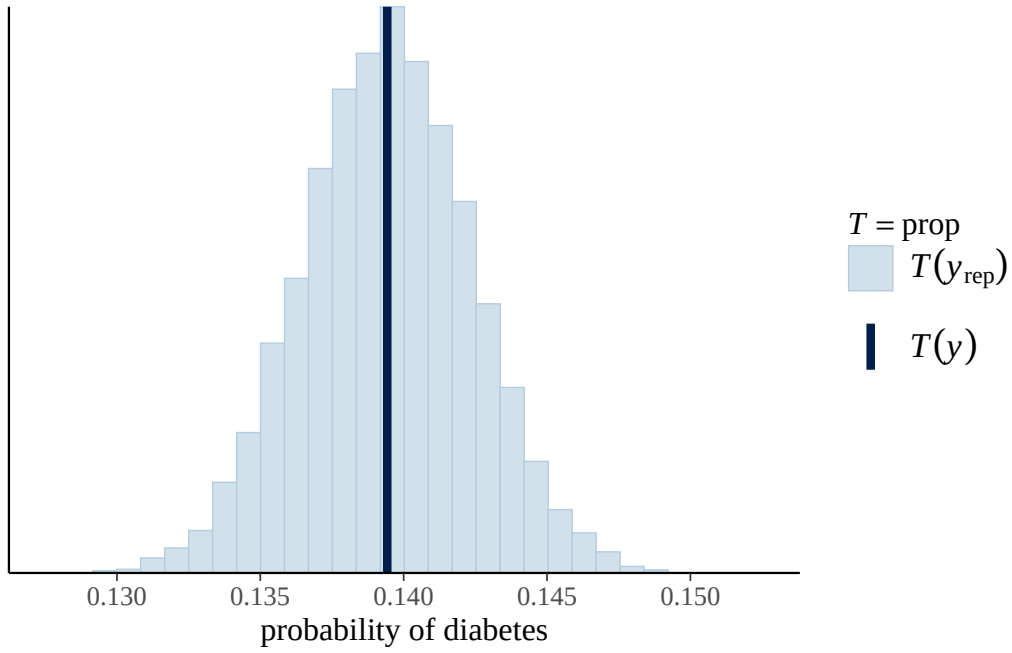
```
                  10%        90%
(Intercept) -4.0661146 -3.9160894
GenHlth      0.7468968  0.7931291
```

```
exp(posterior_interval(diabetes_model_def_GH, prob = 0.80))
```

```
                  10%        90%
(Intercept) 0.01714387 0.01991884
GenHlth     2.11044068 2.21030192
```

```
pp_check(diabetes_model_def_GH, nreps = 100,
         plotfun = "stat", stat = "prop") +
  xlab("probability of diabetes")
```

```
Warning: 'nreps' is ignored for this PPC

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
summary_GH <- get_summary(diabetes_model_def_GH, s, cutoff, 2, 6)
summary_GH
```

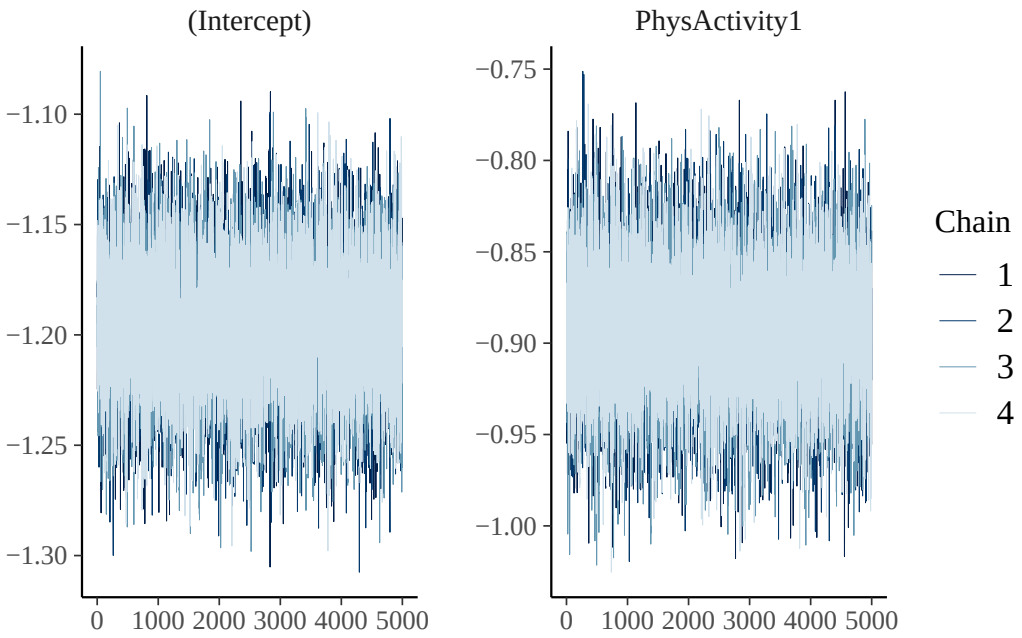
```
$confusion_matrix
 y     0    1
 0 12712 9059
 1   742 2855

$accuracy_rates

sensitivity      0.7937170
specificity      0.5838960
overall_accuracy 0.6136471

$confusion_matrix
 y     0    1
 0 12629 9109
 1   803 2827

$accuracy_rates

sensitivity      0.7787879
specificity      0.5809642
overall_accuracy 0.6092715

$confusion_matrix
 y     0    1
 0 12675 9132
 1   724 2837

$accuracy_rates

sensitivity      0.7966863
specificity      0.5812354
overall_accuracy 0.6114790

$confusion_matrix
 y     0    1
 0 12687 9143
 1   760 2778

$accuracy_rates
```

```
sensitivity      0.7851894
specificity      0.5811727
overall_accuracy 0.6096263

$confusion_matrix
 y     0    1
 0 12780 9054
 1   789 2745

$accuracy_rates

sensitivity      0.7767402
specificity      0.5853256
overall_accuracy 0.6119915
```

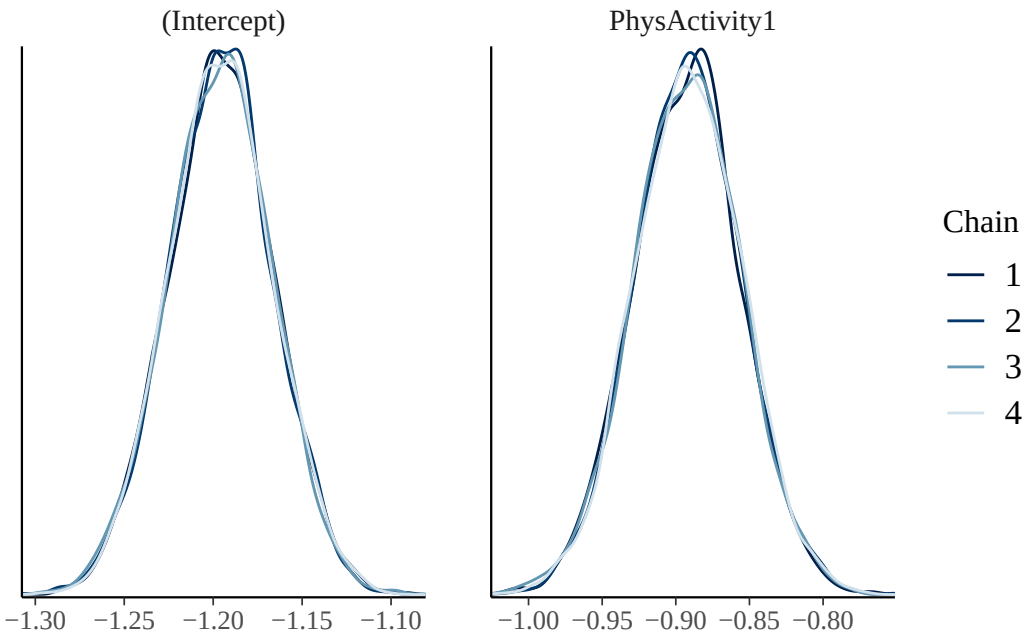
Informative prior

physactivity

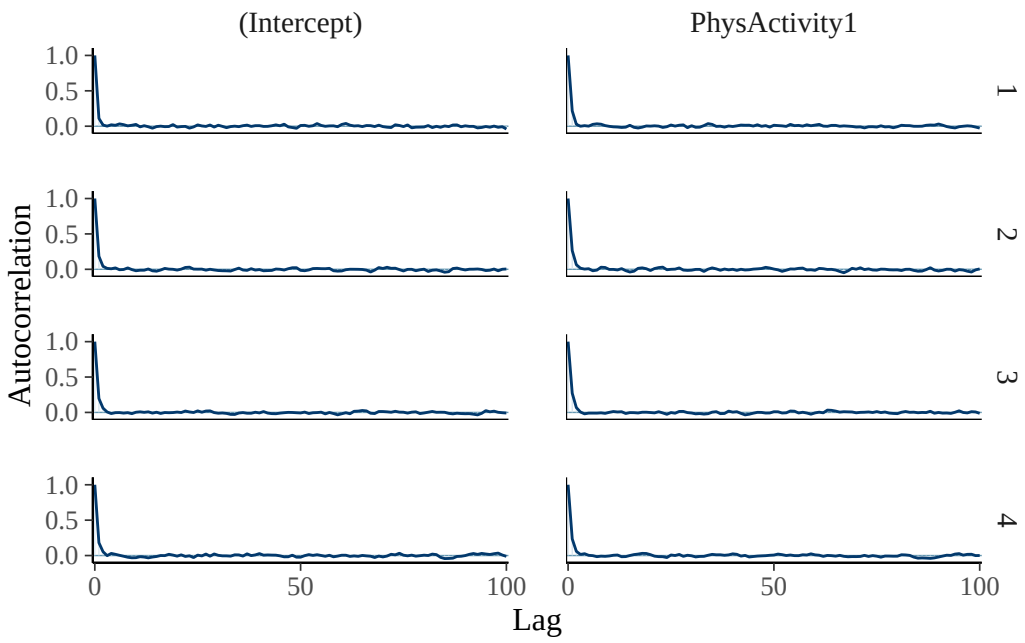
```
diabetes_model_I_PA <- stan_glm(Diabetes_binary ~ PhysActivity,
                                data = train_df,
                                family = binomial,
                                prior_intercept = normal(-1.7585, 0.1273, autoscale = FALSE
                                prior = normal(-2.4511, 0.1178, autoscale = FALSE),
                                chains = 4, iter = 5000*2,
                                seed = RANDOM_STATE,
                                prior_PD = FALSE, refresh=FALSE)
mcmc_trace(diabetes_model_I_PA, size=0.1)
```




```
mcmc_dens_overlay(diabetes_model_I_PA)
```



```
mcmc_acf(diabetes_model_I_PA, lags = 100)
```




```
posterior_interval(diabetes_model_I_PA, prob = 0.80)
```

```
                   10%        90%
(Intercept)   -1.233716 -1.1577675
PhysActivity1 -0.938310 -0.8457743
```

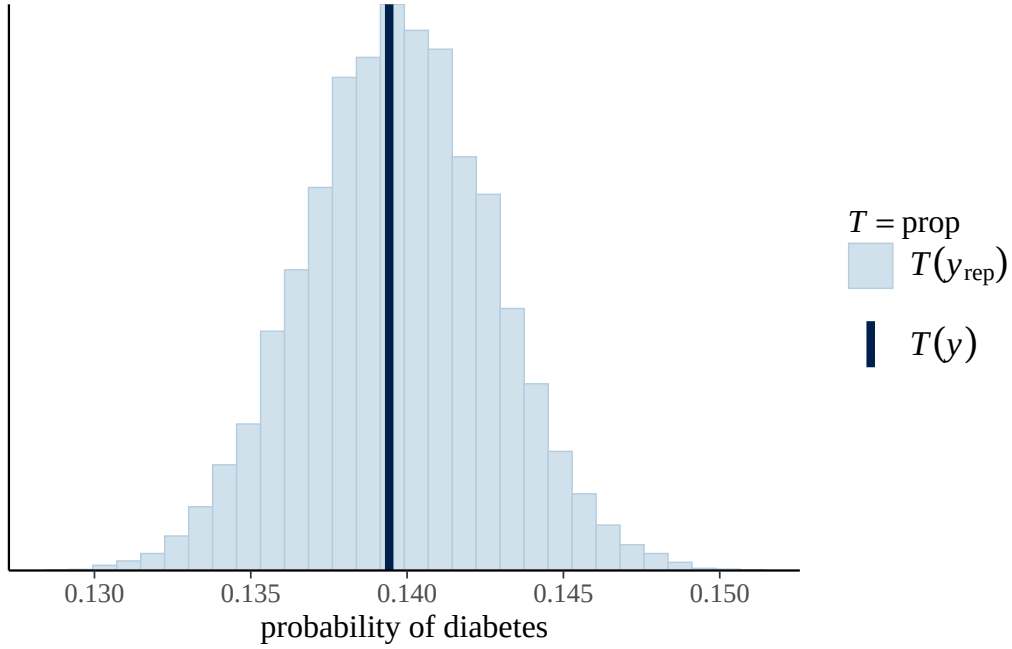
```
exp(posterior_interval(diabetes_model_I_PA, prob = 0.80))
```

```
                   10%       90%
(Intercept)   0.2912085 0.3141868
PhysActivity1 0.3912886 0.4292249
```

```
pp_check(diabetes_model_I_PA, nreps = 100,
         plotfun = "stat", stat = "prop") +
  xlab("probability of diabetes")
```

```
Warning: 'nreps' is ignored for this PPC
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
summary_I_PA <- get_summary(diabetes_model_I_PA, s, cutoff, 2, 6)
summary_I_PA
```

```
$confusion_matrix
 y     0    1
 0 16949 4822
 1  2200 1397

$accuracy_rates

sensitivity      0.3883792
specificity      0.7785127
overall_accuracy 0.7231946

$confusion_matrix
 y     0    1
```

```
 0 16941 4797
 1  2286 1344

$accuracy_rates

sensitivity      0.3702479
specificity      0.7793265
overall_accuracy 0.7207900

$confusion_matrix
 y     0    1
 0 16959 4848
 1  2257 1304

$accuracy_rates

sensitivity      0.3661893
specificity      0.7776861
overall_accuracy 0.7199227

$confusion_matrix
 y     0    1
 0 16872 4958
 1  2237 1301

$accuracy_rates

sensitivity      0.3677219
specificity      0.7728814
overall_accuracy 0.7163750

$confusion_matrix
 y     0    1
 0 16961 4873
 1  2253 1281

$accuracy_rates

sensitivity      0.3624788
specificity      0.7768160
overall_accuracy 0.7190949
```

For PhysActivity, the results are exactly the same for models with default and informative

prior. The model has low sensitivity (true positive rate), which is very bad for detecting diabetes because the model predicts a lots of patients as negative when they actually have diabetes.