# Amyloid

Liwen Yin

2025-02-05

## data cleaning

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
neg <- read.csv("Amyloid_negative.csv")
pos <- read.csv("Amyloid_positive.csv")
amyloid <- rbind(neg, pos)
colnames(amyloid)[which(names(amyloid) == "Amyloid")] <- "amyloid_status"

amyloid <- amyloid %>%
  dplyr::select(-matches("X")) %>%
  mutate(amyloid_status = factor(ifelse(amyloid_status == "Y", 1, 0))) %>%
  mutate(
    Laterality = as.factor(Laterality),
    Race = as.factor(Race),
    Monoclonal.Gammopathy = as.factor(Monoclonal.Gammopathy),
    Rheumatoid.Arthritis = as.factor(Rheumatoid.Arthritis),
    Coronary.Artery.Disease = as.factor(Coronary.Artery.Disease),
    Afib = as.factor(Afib),
    Degenerative.Spine.Disease = as.factor(Degenerative.Spine.Disease),
    Diabetes = as.factor(Diabetes),
    Tendinopathy = as.factor(Tendinopathy),
    EMG = as.factor(EMG),
    Bilateral = as.factor(Bilateral.)
  )
amyloid$Grade <- factor(amyloid$Grade,
                        levels = c(0, "mild", "moderate", "severe"),
                        labels = c("none", "mild", "moderate", "severe"))
amyloid<- amyloid %>% filter(!is.na(Grade))
```

### logistic regression model

```
model1 <- glm(amyloid_status ~ Points + Grade, data = amyloid, family = binomial)
summary(model1)
```

```
##
## Call:
## glm(formula = amyloid_status ~ Points + Grade, family = binomial,
##     data = amyloid)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -21.38240 3225.24298  -0.007  0.99471
## Points           0.02816    0.00988   2.850  0.00437 **
## Grademild        0.02632 3591.37812   0.000  0.99999
## Grademoderate   15.81842 3225.24285   0.005  0.99609
## Gradesevere     17.35876 3225.24284   0.005  0.99571
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 186.91  on 188  degrees of freedom
## Residual deviance: 150.50  on 184  degrees of freedom
## AIC: 160.5
##
## Number of Fisher Scoring iterations: 17
```

### Bayesian generalized linear models

```
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.32.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
library(arm)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: Matrix

## Loading required package: lme4

##
## arm (Version 1.14-4, built: 2024-4-1)

## Working directory is /Users/vivien/Desktop/MSSP/676/consulting2/consulting2025

##
## Attaching package: 'arm'

## The following objects are masked from 'package:rstanarm':
##
##     invlogit, logit
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
bayes_model <- stan_glm(amyloid_status ~ Grade,
                        data = amyloid,
                        family = binomial,
                        prior = student_t(3, 0, 2.5),  #weakly informative
                        chains = 4, iter = 2000, seed = 123,refresh = 0)
```

```
## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: Examine the pairs() plot to diagnose sampling problems
```

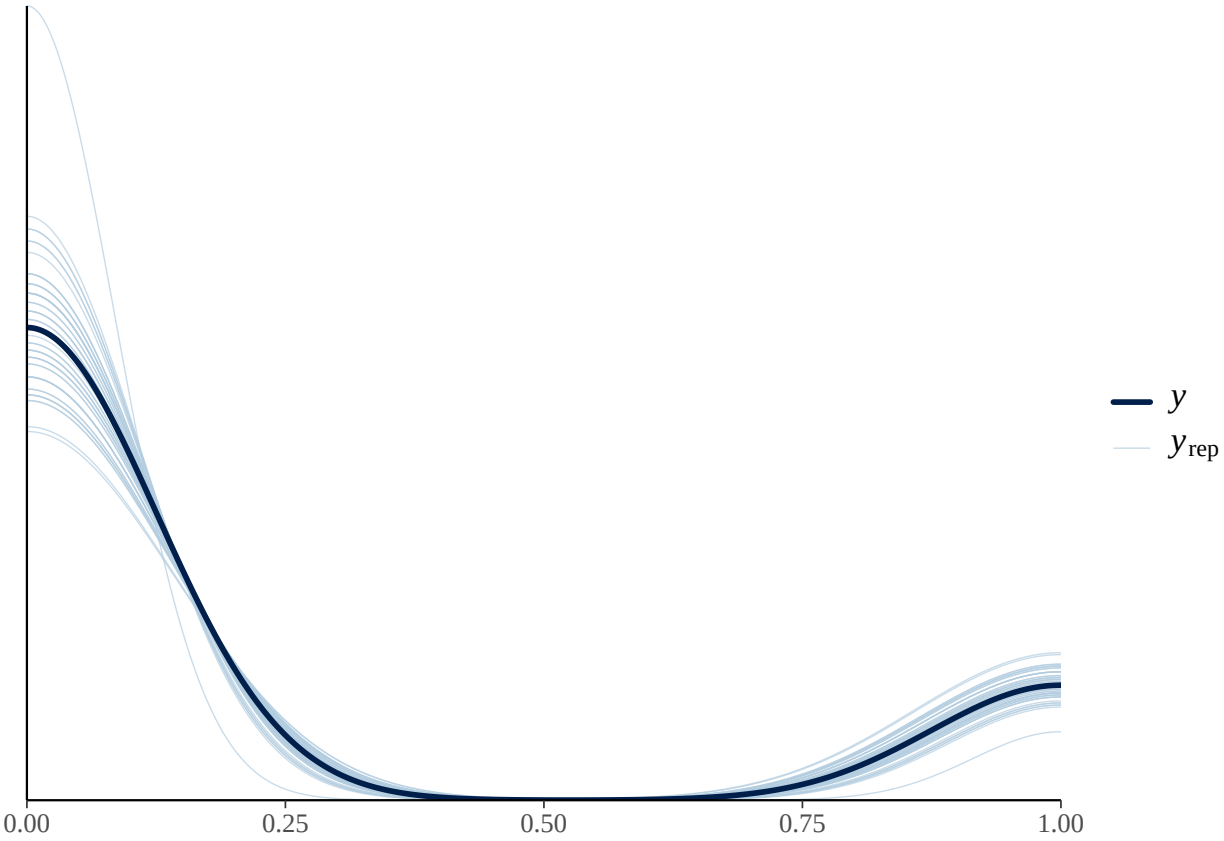
```
summary(bayes_model)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      amyloid_status ~ Grade
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 189
##  predictors:   4
##
## Estimates:
##                mean   sd   10%   50%   90%
## (Intercept)   -3.4    1.6 -5.5  -3.2  -1.6
## Grademild     -2.2    2.9 -5.6  -1.8   0.7
## Grademoderate  1.1    1.6 -0.7   1.0   3.2
## Gradesevere    2.9    1.6  1.0   2.7   4.9
##
## Fit Diagnostics:
##           mean   sd   10%   50%   90%
## mean_PPD 0.2    0.0  0.2   0.2   0.2
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  1092
## Grademild     0.1  1.0  1104
## Grademoderate 0.0  1.0  1121
## Gradesevere   0.0  1.0  1103
## mean_PPD      0.0  1.0  3299
## log-posterior 0.0  1.0  1167
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
#Credible Interval
posterior_interval(bayes_model, prob = 0.95)
```

```
##                     2.5%      97.5%
## (Intercept)   -6.9257975 -0.8996405
## Grademild     -9.1879833  1.8046039
## Grademoderate -1.4321039  4.7099111
## Gradesevere    0.2697847  6.4219404
```

```
#pp check
pp_check(bayes_model)
```



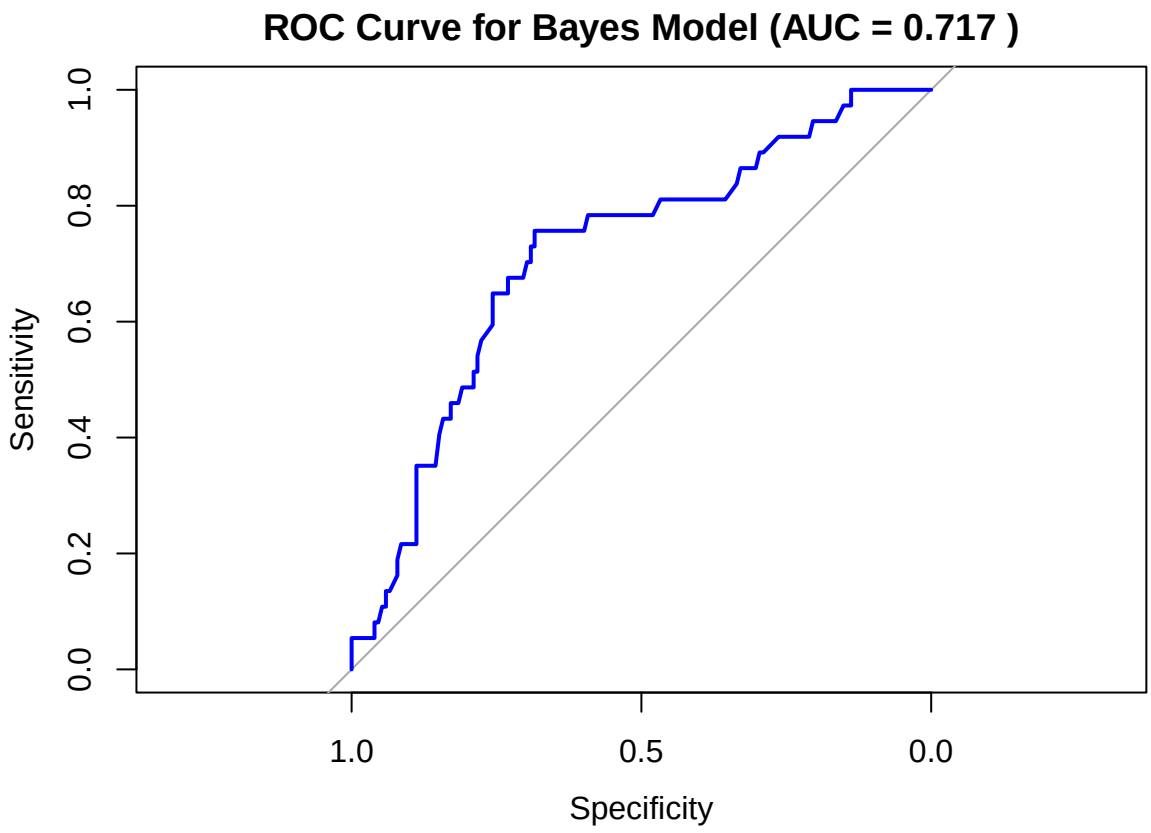

```
#Odds ratio
exp(posterior_interval(bayes_model, prob = 0.95))
```

```
##                        2.5%       97.5%
## (Intercept)   0.0009821196   0.4067159
## Grademild     0.0001022609   6.0775635
## Grademoderate 0.2388059779 111.0422827
## Gradesevere   1.3096824350 615.1956973
```

```
#ROC & AUC
set.seed(1)
predicted_probs <- posterior_predict(bayes_model, type = "response") %>% colMeans()
actual_values <- amyloid$amyloid_status
roc_curve <- roc(actual_values, predicted_probs,levels = c(0, 1))
```

```
## Setting direction: controls < cases
```

```
auc_value <- auc(roc_curve)
plot(roc_curve,  main = paste("ROC Curve for Bayes Model (AUC =", round(auc_value, 3), ")"),
     col = "blue",
     lwd = 2)
```




```
print(paste("AUC:", round(auc_value, 3)))
```

```
## [1] "AUC: 0.717"
```

For every unit increase in the linear component of Grade, the log-odds of amyloid positivity increase by 3.2. mean_posterior predictive distribution:the mean is 0.2, suggesting the model predicts an average probability of positive as approximately 20% across the dataset. 95% Bayesian credible intervals : The linear effect of Grade is strongly positive. The quadratic effect of Grade includes zero, suggesting no significant quadratic effect. odds ratio: Linear Effect of Grade OR: [0.01, 0.21] : The baseline odds of amyloid positivity are very low. Quadratic Effect of Grade OR: [2.51, 1211.14] : Indicates that higher Grade strongly increases the odds of amyloid positivity. pp check: Posterior predictive distribution compared to the observed data distribution; Slight deviations at very low probabilities.

```
bayes_model2 <- stan_glm(
  amyloid_status ~ Grade,
  data = amyloid,
  family = binomial,
  prior = normal(0, 1, autoscale = TRUE),  # Ordered effect prior
  chains = 4, iter = 2000, seed = 123,refresh = 0
)
summary(bayes_model2)
```

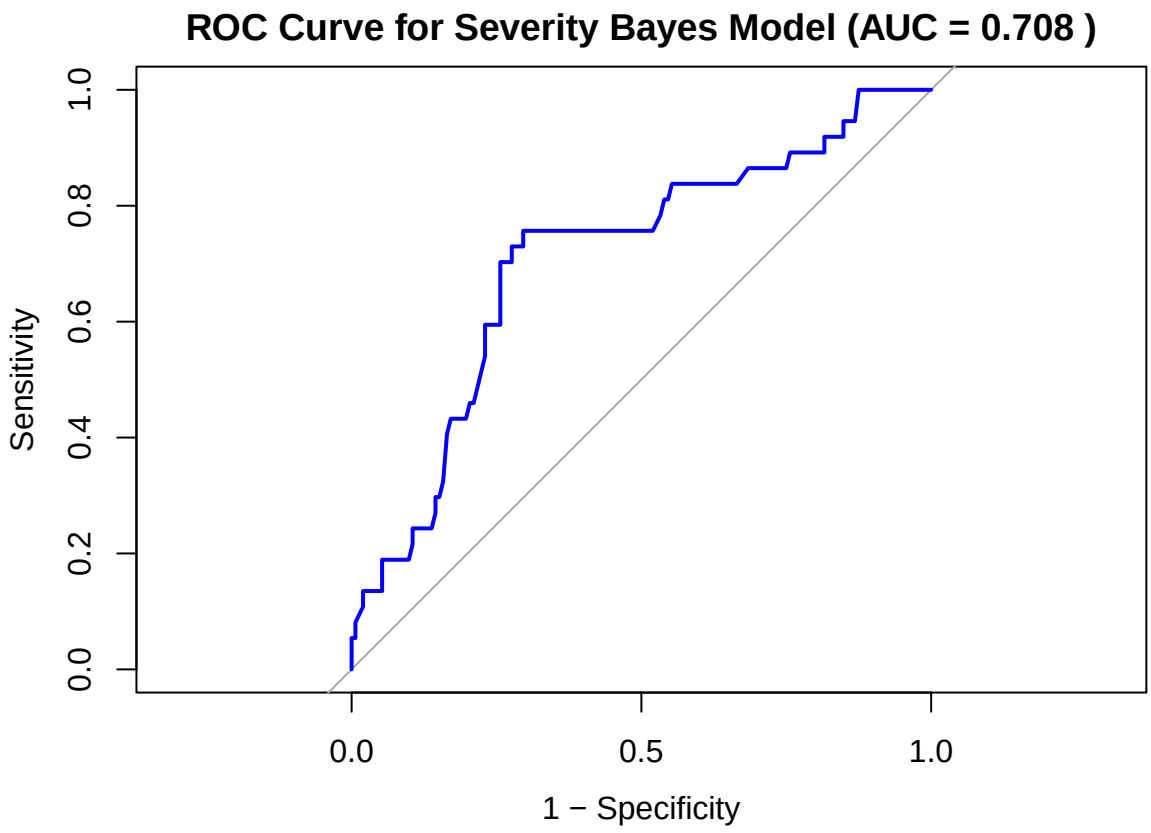
```
##
## Model Info:
```

```
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      amyloid_status ~ Grade
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 189
##  predictors:   4
##
## Estimates:
##                 mean   sd   10%   50%   90%
## (Intercept)   -2.7    1.1 -4.2  -2.7  -1.3
## Grademild     -3.0    2.3 -6.0  -2.8  -0.2
## Grademoderate  0.4    1.2 -1.0   0.4   1.9
## Gradesevere    2.1    1.1  0.7   2.1   3.6
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.2    0.0  0.1   0.2   0.2
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  1514
## Grademild     0.1  1.0  1521
## Grademoderate 0.0  1.0  1592
## Gradesevere   0.0  1.0  1527
## mean_PPD      0.0  1.0  3771
## log-posterior 0.0  1.0  1319
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
set.seed(1)
predicted_probs <- posterior_predict(bayes_model2, type = "response") %>% colMeans()
actual_values <- amyloid$amyloid_status
roc_curve <- roc(actual_values, predicted_probs,levels = c(0, 1))
```

```
## Setting direction: controls < cases
```

```
auc_value <- auc(roc_curve)
plot(roc_curve,  main = paste("ROC Curve for Severity Bayes Model (AUC =", round(auc_value, 3), ")"),
     col = "blue",
     lwd = 2, legacy.axes = TRUE)
```




```
print(paste("AUC:", round(auc_value, 3)))
```

```
## [1] "AUC: 0.708"
```

Autoscaling adjusts the prior standard deviation relative to the data scale, ensuring that the prior matches the actual predictor variability. The normal prior with scaling ensures coefficients are regularized to prevent extreme estimates in case of sparse data.

```
library(rstanarm)
amyloid$Age_100 <- amyloid$Age/100
bayes_model3 <- stan_glm(
  amyloid_status ~ Race + Age + Afib + Degenerative.Spine.Disease + Bilateral. + Monoclonal.Gammopathy +
  data = amyloid,
  family = binomial,
 prior = student_t(1, 0, 5),  # prior for coefficients
  chains = 4, iter = 2000, seed = 123, refresh = 0
)
```

```
## Warning: There were 4 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

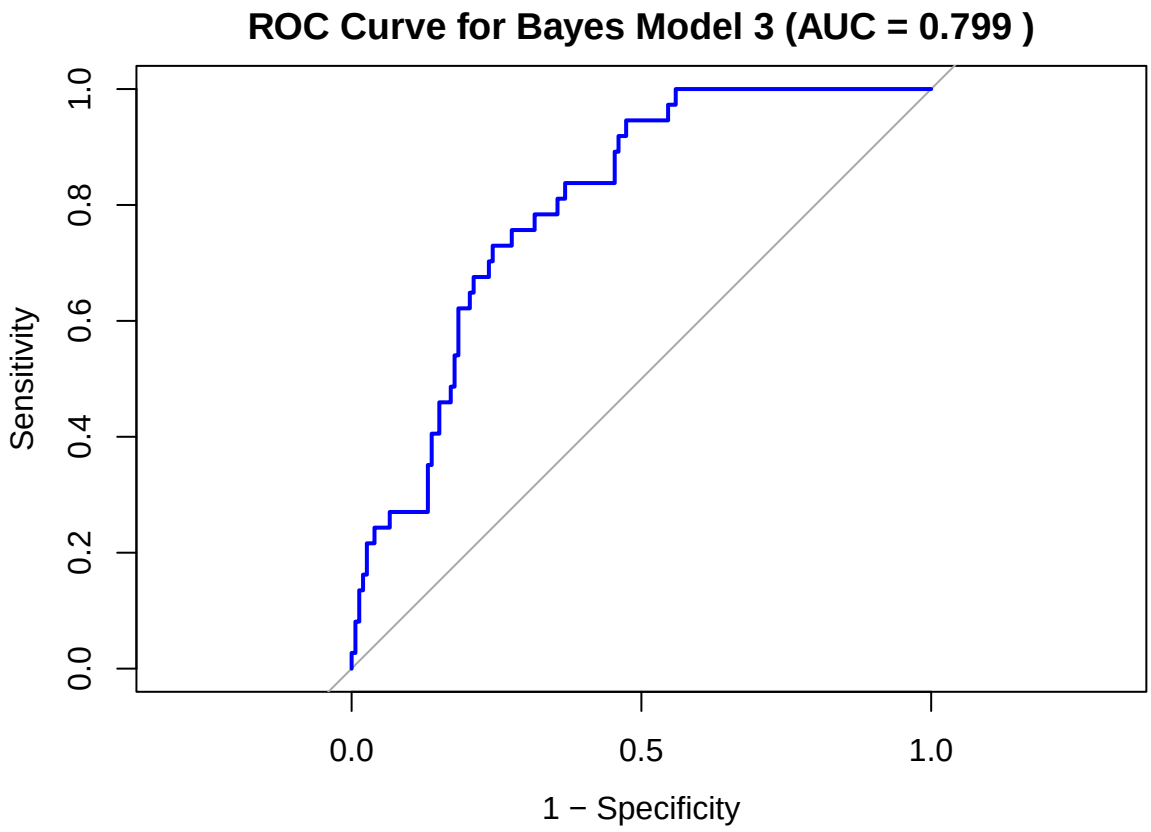
```r
print(summary(bayes_model3), digits=4)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      amyloid_status ~ Race + Age + Afib + Degenerative.Spine.Disease +
##     Bilateral. + Monoclonal.Gammopathy + Rheumatoid.Arthritis +
##     Grade
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 189
##  predictors:   11
##
## Estimates:
##                               mean     sd       10%      50%      90%
## (Intercept)                  -7.9534   3.5439 -12.5152  -7.4835  -4.0258
## RaceWhite                    -0.3420   0.7917  -1.3058  -0.3713   0.6482
## Age                           0.0359   0.0215   0.0089   0.0356   0.0629
## AfibY                         0.5707   0.5701  -0.1606   0.5841   1.2850
## Degenerative.Spine.DiseaseY   0.3685   0.4345  -0.1779   0.3641   0.9398
## Bilateral.Y                   1.2021   0.6308   0.4125   1.1838   2.0390
## Monoclonal.GammopathyY       -0.0511   1.0618  -1.3894  -0.0292   1.2784
## Rheumatoid.ArthritisY         1.1030   1.8248  -1.1696   1.0646   3.4178
## Grademild                    -4.5736   6.6607 -12.5947  -3.0955   1.4326
## Grademoderate                 2.3013   3.2083  -1.0171   1.7081   6.3356
## Gradesevere                   3.7536   3.2143   0.3974   3.1475   7.7684
##
## Fit Diagnostics:
##            mean   sd     10%    50%    90%
## mean_PPD 0.1973 0.0353 0.1534 0.1958 0.2434
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                              mcse   Rhat   n_eff
## (Intercept)                  0.1036 1.0021 1171
## RaceWhite                    0.0131 1.0009 3663
## Age                          0.0003 0.9991 5094
## AfibY                        0.0085 0.9992 4458
## Degenerative.Spine.DiseaseY 0.0066 0.9995 4380
## Bilateral.Y                  0.0102 1.0004 3843
## Monoclonal.GammopathyY       0.0158 0.9994 4497
## Rheumatoid.ArthritisY        0.0299 1.0010 3727
## Grademild                    0.1939 1.0023 1180
## Grademoderate                0.1009 1.0030 1011
## Gradesevere                  0.1018 1.0031  997
## mean_PPD                     0.0006 1.0003 4008
## log-posterior                0.0610 1.0016 1423
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
set.seed(1)
predicted_probs <- posterior_predict(bayes_model3, type = "response") %>% colMeans()
actual_values <- amyloid$amyloid_status
roc_curve <- roc(actual_values, predicted_probs,levels = c(0, 1))
```

```
## Setting direction: controls < cases
```

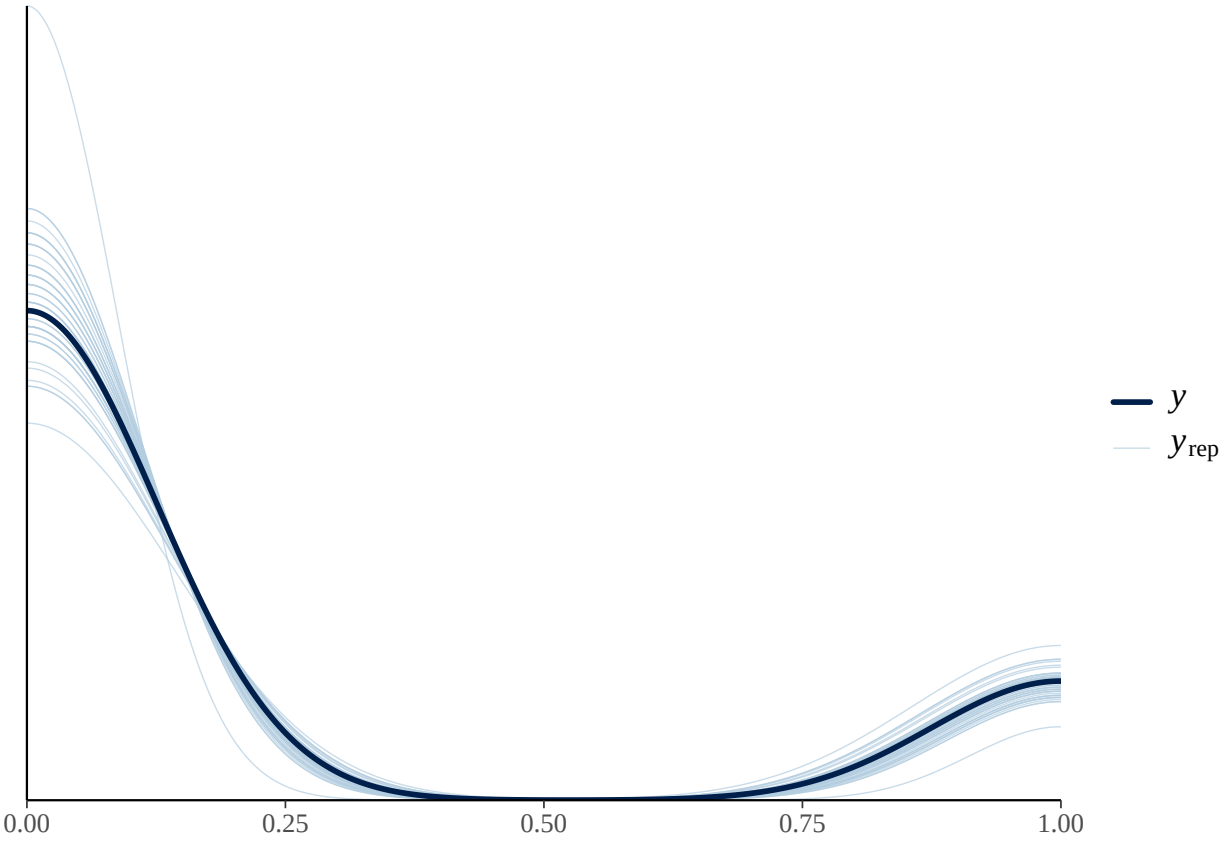
```r
auc_value <- auc(roc_curve)
plot(roc_curve,  main = paste("ROC Curve for Bayes Model 3 (AUC =", round(auc_value, 3), ")"),
     col = "blue",
     lwd = 2, legacy.axes = TRUE)
```




```r
print(paste("AUC:", round(auc_value, 3)))
```

```
## [1] "AUC: 0.799"
```

```r
pp_check(bayes_model3)
```




normal(0, 1) indicates no strong prior belief about the direction (positive or negative) of the effect of the predictors on the log-odds of amyloid_status. A standard deviation of 1 allows for moderate variability in the coefficients. prior_intercept = normal(0, 5):Allows for wide baseline probabilities, accommodating substantial uncertainty in the initial prevalence of the outcome.

```
bayes_model4 <- stan_glm(
  amyloid_status ~ Points,
  data = amyloid,
  family = binomial,
  prior = normal(0, 1),      # Weakly informative
  prior_intercept = normal(0, 5),
  chains = 4, iter = 2000, seed = 123, refresh = 0
)
summary(bayes_model4)
```

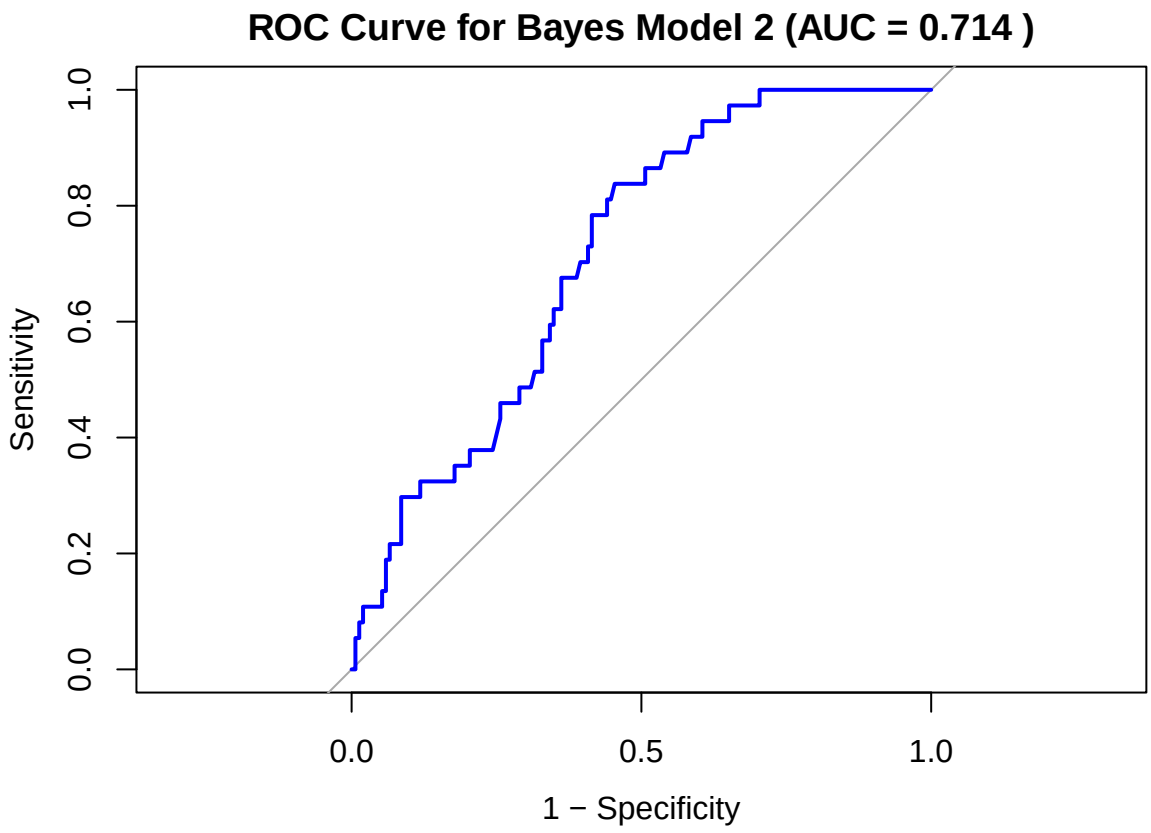
```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      amyloid_status ~ Points
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 189
##  predictors:   2
##
```

```
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) -5.8    1.1 -7.4  -5.8  -4.4
## Points       0.0    0.0  0.0   0.0   0.0
##
## Fit Diagnostics:
##           mean   sd   10%   50%   90%
## mean_PPD 0.2    0.0  0.1   0.2   0.2
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  1626
## Points        0.0  1.0  1746
## mean_PPD      0.0  1.0  2906
## log-posterior 0.0  1.0  1669
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
set.seed(1)
predicted_probs <- posterior_predict(bayes_model4, type = "response") %>% colMeans()
actual_values <- amyloid$amyloid_status
roc_curve <- roc(actual_values, predicted_probs,levels = c(0, 1))
```

```
## Setting direction: controls < cases
```

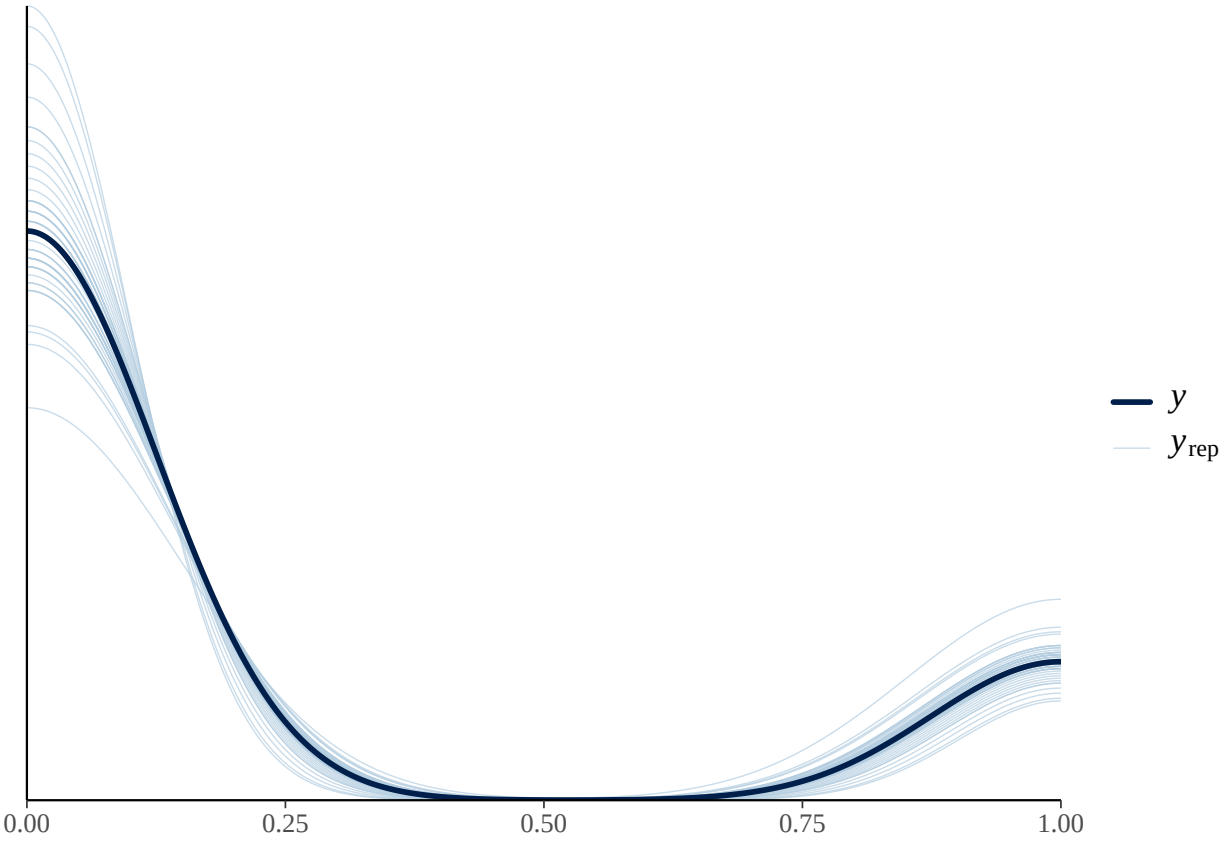
```
auc_value <- auc(roc_curve)
plot(roc_curve,  main = paste("ROC Curve for Bayes Model 2 (AUC =", round(auc_value, 3), ")"),
     col = "blue",
     lwd = 2, legacy.axes = TRUE)
```




```
print(paste("AUC:", round(auc_value, 3)))
```

```
## [1] "AUC: 0.714"
```

```
pp_check(bayes_model4)
```




```
reg <- glm(amyloid_status ~ Points, family = binomial, data = amyloid)
summary(reg)
```

```
##
## Call:
## glm(formula = amyloid_status ~ Points, family = binomial, data = amyloid)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.715152   1.156982   -4.94 7.82e-07 ***
## Points       0.035890   0.009202    3.90 9.60e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 186.91  on 188  degrees of freedom
## Residual deviance: 169.65  on 187  degrees of freedom
## AIC: 173.65
##
## Number of Fisher Scoring iterations: 4
```

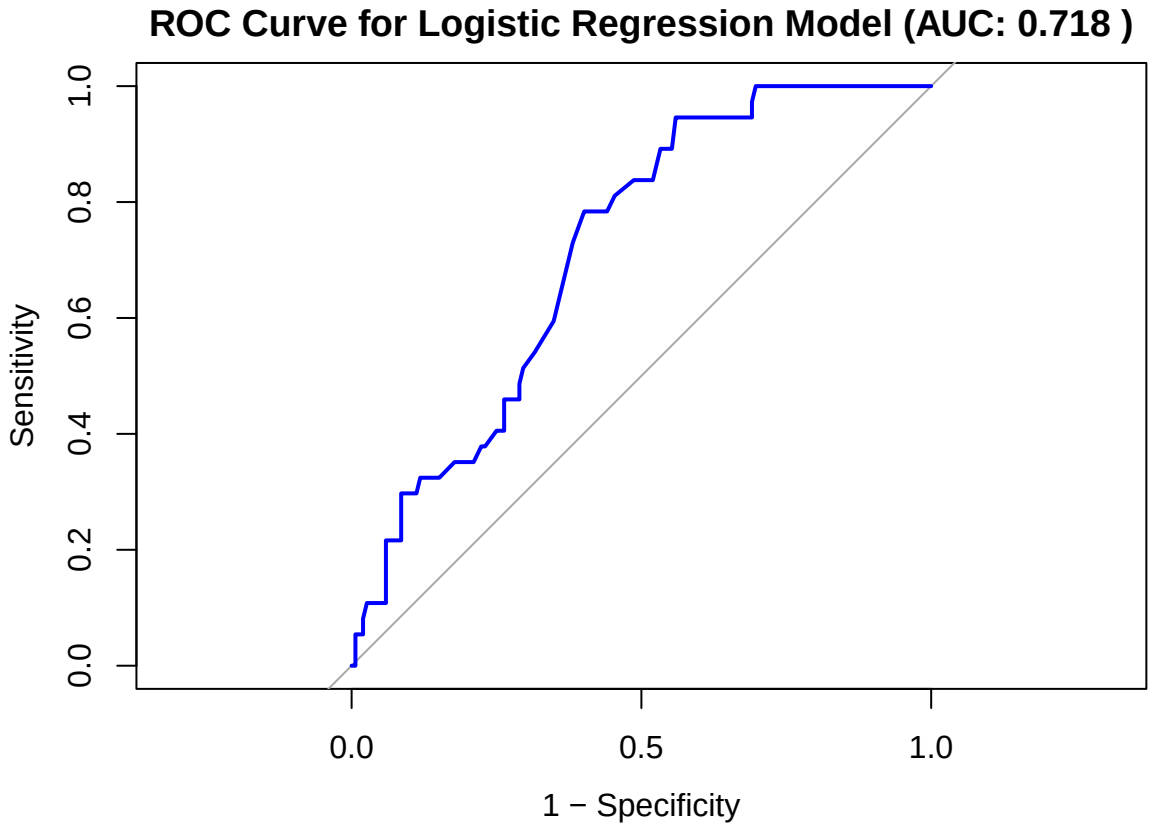
```
library(pROC)
set.seed(1)
predicted_probs <- predict(reg, type = "response")
```

```
actual_values <- amyloid$amyloid_status
roc_curve <- roc(actual_values, predicted_probs)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_value <- auc(roc_curve)
auc_value_rounded <- round(auc_value, 3)
plot(roc_curve, main = paste("ROC Curve for Logistic Regression Model (AUC:", auc_value_rounded, ")"), c
```




```
print(paste("AUC=", auc_value_rounded))
```

```
## [1] "AUC= 0.718"
```

If the effect of Points is relatively small (statistically significant in glm, but with a small actual effect size), the prior distribution tends to shrink the coefficient of Points toward 0.

amyloid_status ~ Points + grade

```
bayes_model5 <- stan_glm(
  amyloid_status ~ Grade + Points,
  data = amyloid,
```

```r
  family = binomial,
  prior = normal(0, 1),      # Weakly informative
  prior_intercept = normal(0, 5),
  chains = 4, iter = 2000, seed = 123, refresh = 0
)
print(summary(bayes_model5),digit = 4)
```

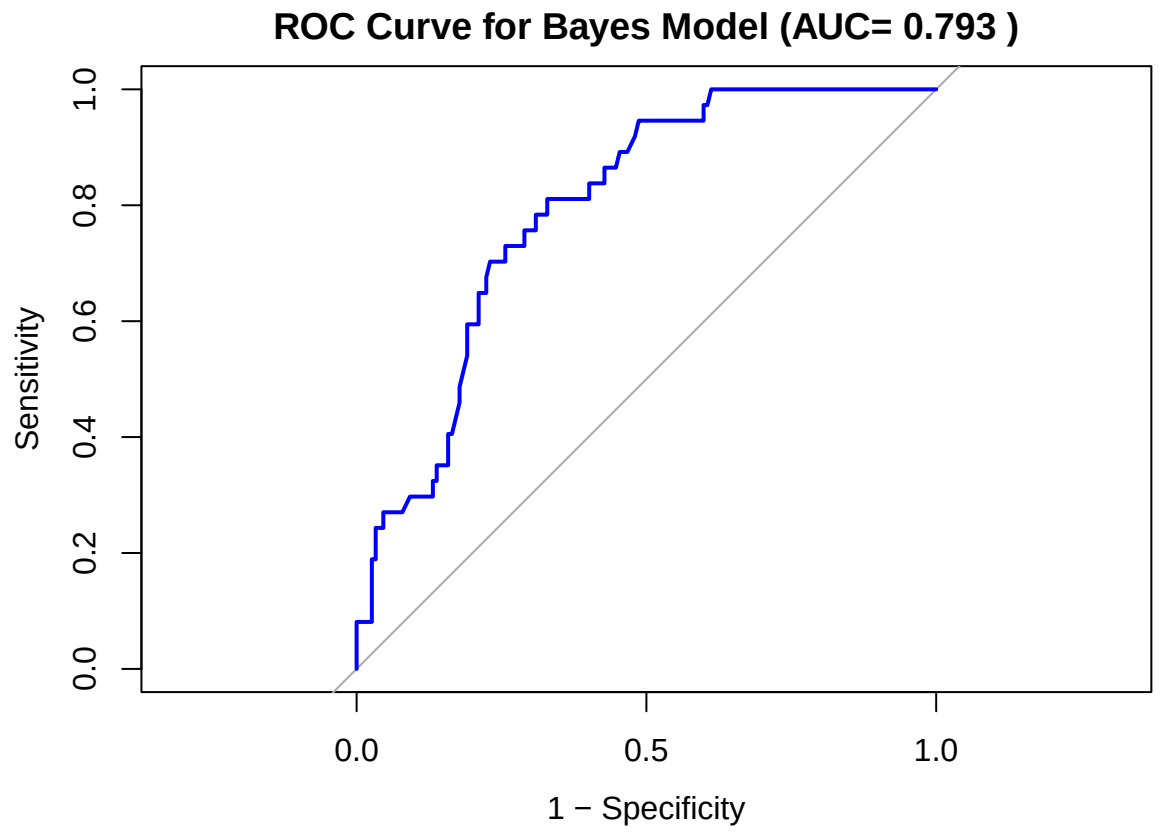
```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      amyloid_status ~ Grade + Points
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 189
##  predictors:   5
##
## Estimates:
##                 mean    sd      10%     50%     90%
## (Intercept)   -5.7304  1.3975 -7.5070 -5.7033 -3.9881
## Grademild     -0.7533  0.8106 -1.8017 -0.7428  0.2869
## Grademoderate -0.2261  0.6512 -1.0493 -0.2541  0.6204
## Gradesevere    1.2484  0.6443  0.4375  1.2409  2.0782
## Points         0.0311  0.0102  0.0183  0.0310  0.0440
##
## Fit Diagnostics:
##            mean   sd     10%    50%    90%
## mean_PPD 0.1969 0.0375 0.1481 0.1958 0.2434
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse   Rhat   n_eff
## (Intercept)   0.0262 1.0013 2847
## Grademild     0.0146 1.0000 3098
## Grademoderate 0.0134 1.0004 2359
## Gradesevere   0.0132 1.0003 2365
## Points        0.0002 1.0004 2814
## mean_PPD      0.0006 1.0013 4076
## log-posterior 0.0413 1.0017 1552
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
library(pROC)
set.seed(1)
predicted_probs <- predict(bayes_model5, type = "response")
actual_values <- amyloid$amyloid_status
roc_curve <- roc(actual_values, predicted_probs)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
auc_value <- auc(roc_curve)
auc_value_rounded <- round(auc_value, 3)
plot(roc_curve, main = paste("ROC Curve for Bayes Model (AUC=", auc_value_rounded, ")"), col = "blue", l
```




```r
print(paste("AUC=", auc_value_rounded))
```

```
## [1] "AUC= 0.793"
```