

amyloid2.0

Liwen Yin

2025-02-19

```
amypopu <- read.csv("Amyloid All Patients.csv")
amypopu$sample <- factor(ifelse(seq_len(nrow(amypopu)) <= 189, 1, 0))
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
```

```
amypopu <- amypopu %>%
  mutate(
    Severity = ifelse(Grade %in% c("0", "mild", "moderate"), "low",
                      ifelse(Grade == "severe", "high", NA)),
    Severity = factor(Severity, levels = c("low", "high")),
    amyloid_status = ifelse(Amyloid == "Y", 1, 0),
    amyloid_status = factor(amyloid_status, levels = c(0, 1))
  ) %>%
  dplyr::select(-Amyloid, -Grade) %>%
  dplyr::select(-matches("X"))
amyloid <- read.csv("amyloid.csv")
amyloid$amyloid_status <- as.factor(amyloid$amyloid_status)
```

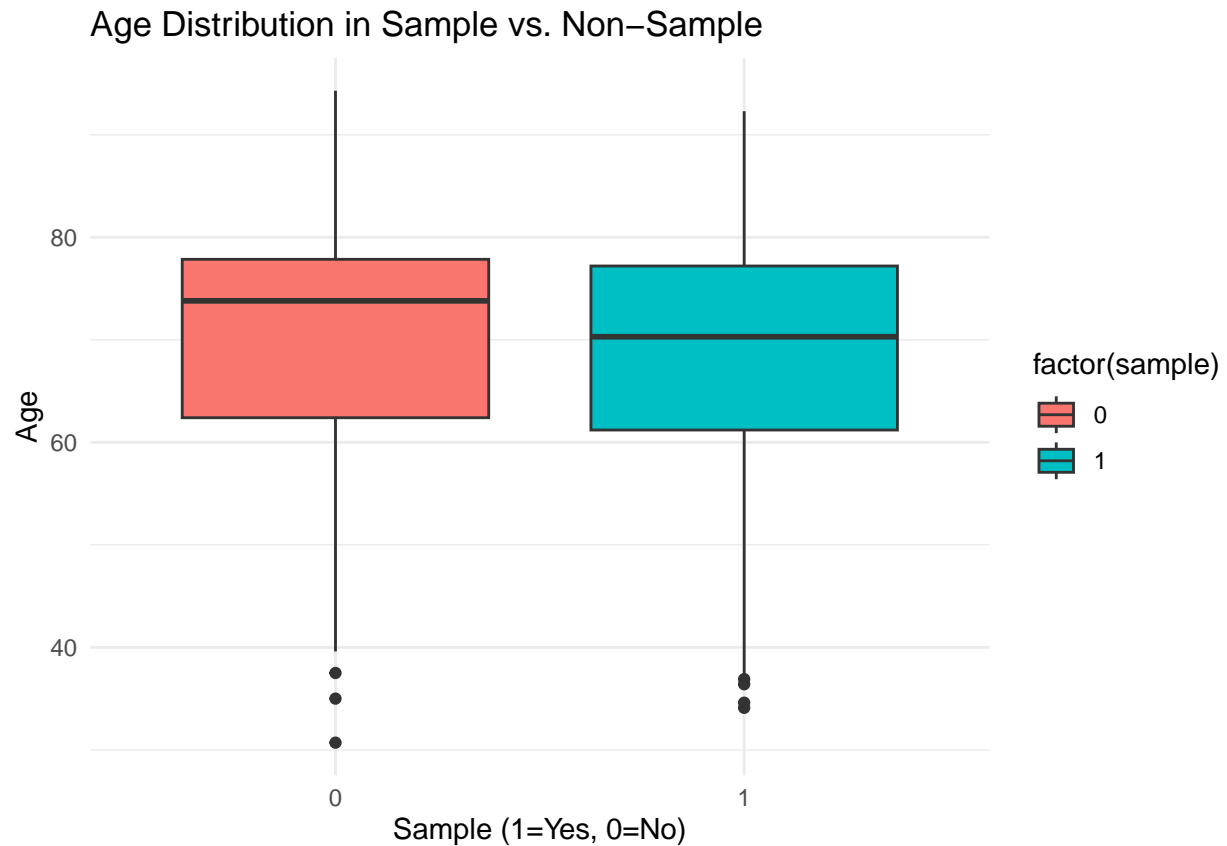
```
summ <- aggregate(cbind(Age, Points) ~ sample, data = amypopu, summary)
summ
```

```
##   sample Age.Min. Age.1st Qu. Age.Median Age.Mean Age.3rd Qu. Age.Max.
## 1      0 30.70000   62.40000   73.80000 70.25395   77.85000 94.30000
## 2      1 34.10000   61.20000   70.30000 68.34339   77.20000 92.30000
##   Points.Min. Points.1st Qu. Points.Median Points.Mean Points.3rd Qu.
## 1    31.00000    62.00000    74.00000    70.23684    78.00000
## 2    59.00000    99.00000   116.00000   114.94180   129.00000
##   Points.Max.
## 1    94.00000
## 2   180.00000
```

```
library(ggplot2)
```

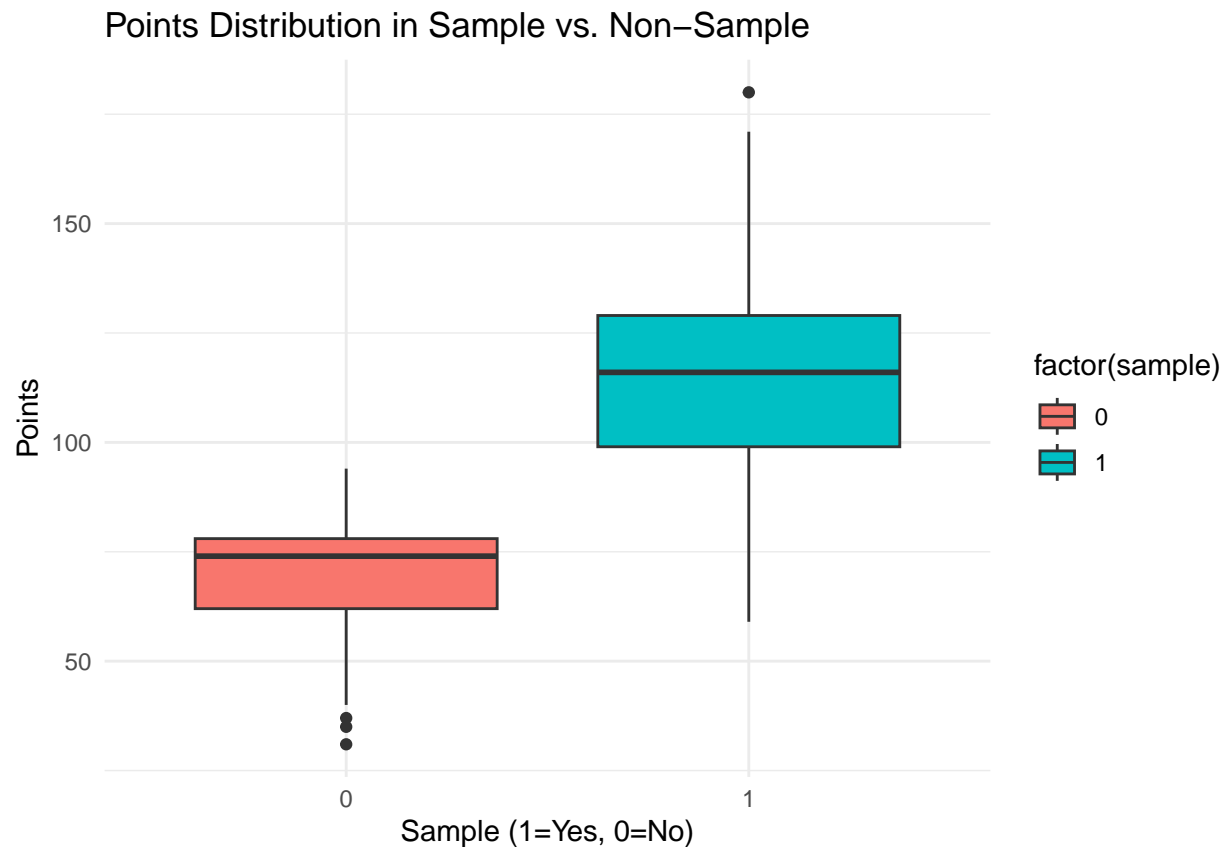
```
#Age
```

```
ggplot(ampopu, aes(x = factor(sample), y = Age, fill = factor(sample))) +  
  geom_boxplot() +  
  labs(title = "Age Distribution in Sample vs. Non-Sample", x = "Sample (1=Yes, 0=No)", y = "Age") +  
  theme_minimal()
```



```
#Points
```

```
ggplot(ampopu, aes(x = factor(sample), y = Points, fill = factor(sample))) +  
  geom_boxplot() +  
  labs(title = "Points Distribution in Sample vs. Non-Sample", x = "Sample (1=Yes, 0=No)", y = "Points") +  
  theme_minimal()
```



```
library(pROC)
```

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

```
##
```

```
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## cov, smooth, var
```

```
Model1 <- glm(sample ~ Points, data = amypopu, family = binomial)
summary(Model1)
```

```
##
```

```
## Call:
```

```
## glm(formula = sample ~ Points, family = binomial, data = amypopu)
```

```
##
```

```
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -15.61627    1.89884  -8.224  < 2e-16 ***
```

```
## Points      0.18007    0.02224   8.095 5.71e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 468.70  on 340  degrees of freedom
## Residual deviance: 151.13  on 339  degrees of freedom
## AIC: 155.13
##
## Number of Fisher Scoring iterations: 7

amypopu$prob_sample <- predict(Model1, type = "response")
amypopu$weight <- ifelse(amypopu$sample == 1, 1 / amypopu$prob_sample, NA)
amyloid <- amyloid %>%
  mutate(weight = amypopu$weight[1:189])
fit_weighted <- glm(amyloid_status ~ Points + Severity,
                    data = amyloid,
                    family = quasibinomial,
                    weights = weight)
summary(fit_weighted)

##
## Call:
## glm(formula = amyloid_status ~ Points + Severity, family = quasibinomial,
##      data = amyloid, weights = weight)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.07338    1.56068  -5.173 5.93e-07 ***
## Points         0.04271    0.01285   3.325 0.001066 **
## SeveritySevere 2.19651    0.55614   3.950 0.000111 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 2.276017)
##
##      Null deviance: 463.31  on 188  degrees of freedom
## Residual deviance: 351.62  on 186  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6

library(arm)

## Loading required package: MASS

##
## Attaching package: 'MASS'

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

## Loading required package: Matrix

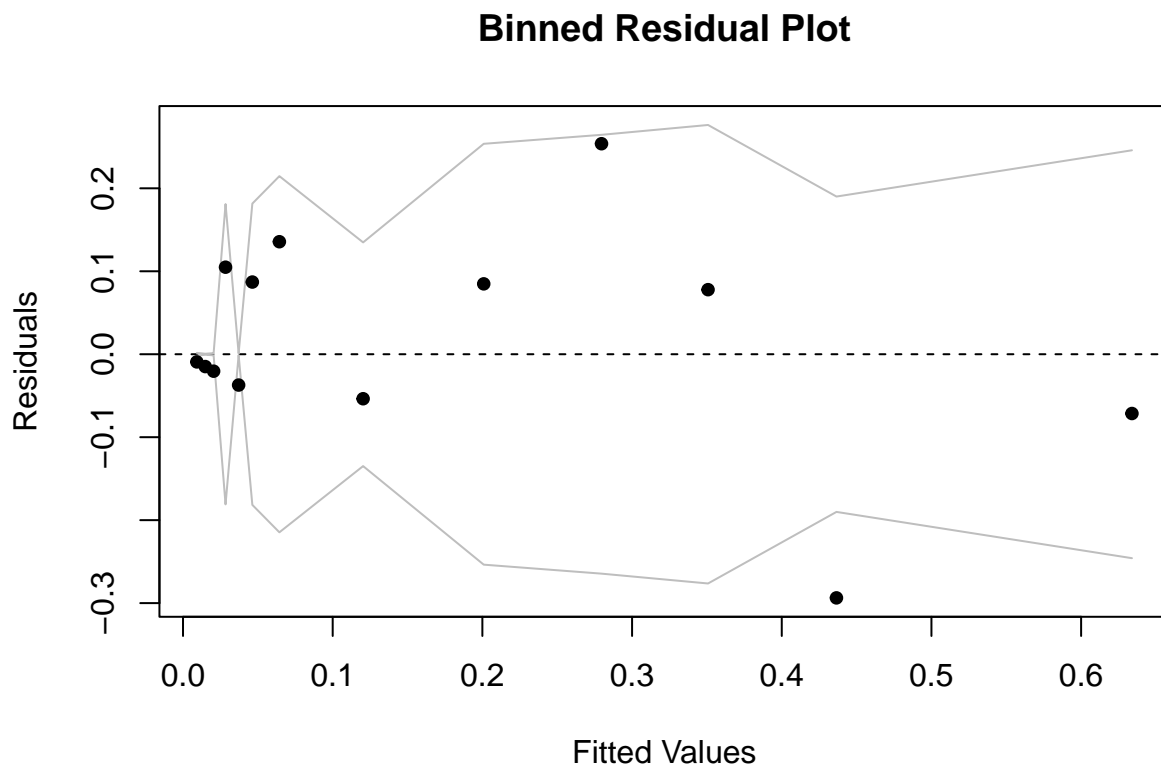
```

```
## Loading required package: lme4

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

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

```
binnedplot(fitted(fit_weighted), residuals(fit_weighted, type = "response"),
  main = "Binned Residual Plot",
  xlab = "Fitted Values", ylab = "Residuals")
```

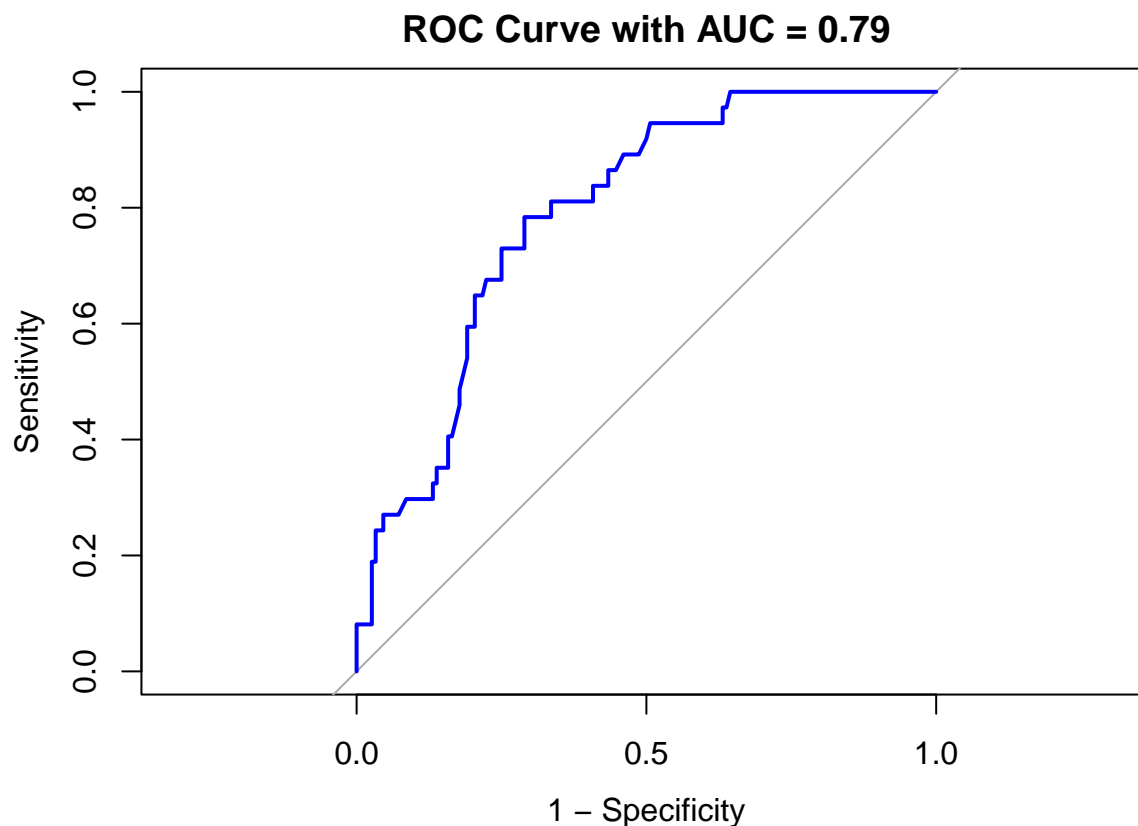


```
predicted_probs <- predict(fit_weighted, type = "response")
roc_curve <- roc(amyloid$amyloid_status, predicted_probs)
```

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

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

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



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

```
## [1] "AUC: 0.79"
```

```
library(pROC)
fit_unweighted <- glm(amyloid_status ~ Points + Severity, data = amyloid, family = binomial)
summary(fit_unweighted)
```

```
##
## Call:
## glm(formula = amyloid_status ~ Points + Severity, family = binomial,
##      data = amyloid)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.93505    1.25526  -4.728 2.27e-06 ***
## Points         0.03001    0.00985   3.047 0.002312 **
## SeveritySevere 1.67919    0.43402   3.869 0.000109 ***
## ---
## 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
```

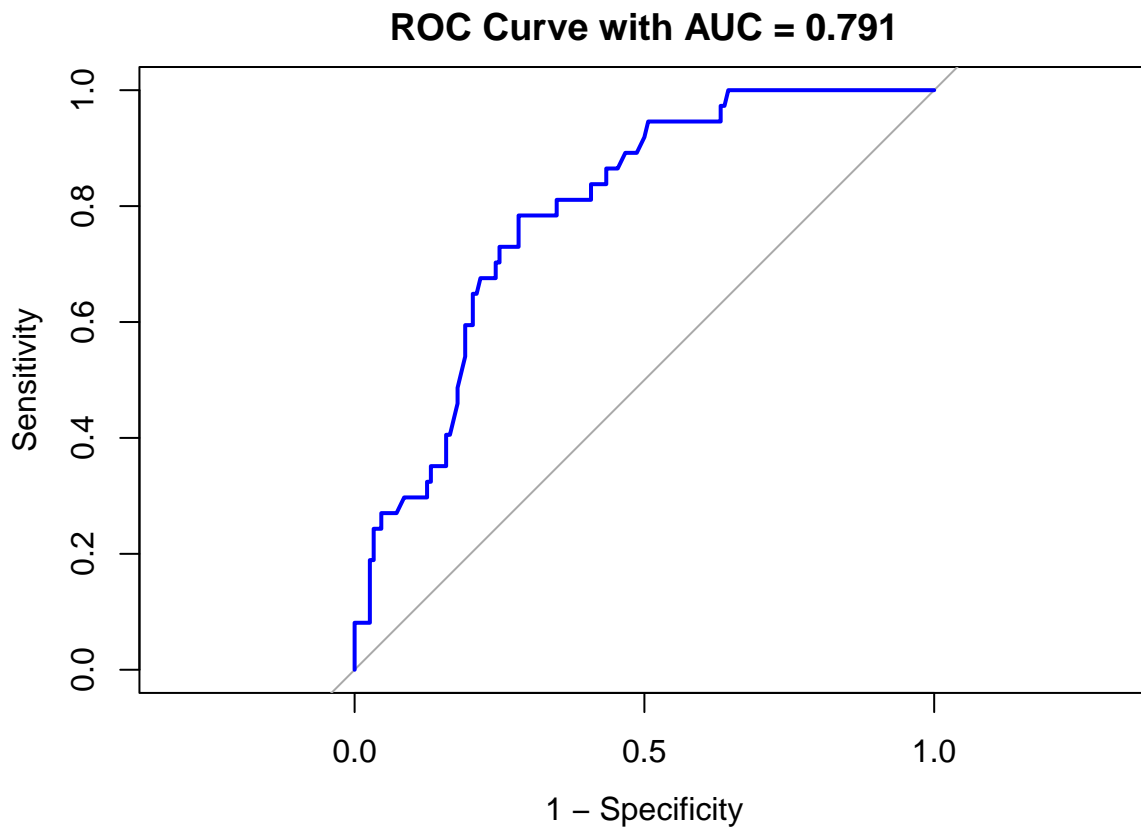
```
## Residual deviance: 152.84 on 186 degrees of freedom
## AIC: 158.84
##
## Number of Fisher Scoring iterations: 5
```

```
predicted_probs <- predict(fit_unweighted, type = "response")
roc_curve <- roc(amyloid$amyloid_status, predicted_probs)
```

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

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

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



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

```
## [1] "AUC: 0.791"
```

```
binnedplot(fitted(fit_unweighted), residuals(fit_unweighted, type = "response"),
           main = "Binned Residual Plot",
           xlab = "Fitted Values", ylab = "Residuals")
```

Binned Residual Plot

