Amyloid Analysis 2

Jin Wen Lin

2025-02-19

Data Prepration

```
# load data
amy_data <- read.csv("Amyloid All Patients.csv")
# sample indicator (1 for sample, 0 for unsample)
amy_data$sample_ind <- factor(ifelse(1:nrow(amy_data) <= 189, 1, 0))

amyloid <- amy_data %>%
  select(-c(X, X28, X42, X22, X11, X14, X8, X16)) %>%  # delete the columns that are not informative
  mutate(
    Amyloid = factor(ifelse(Amyloid == "Y", 1, 0)), # 1 for Y, 0 for N
    Laterality = factor(Laterality),
    Sex = factor(Sex),
    Race = factor(ifelse(Race == "White", 1, 0)), # 1 for white, 0 for else
    Afib = factor(Afib),
    Tendinopathy = factor(Tendinopathy),
    EMG = factor(EMG),
    Bilateral. = factor(Bilateral.),
    Bifringence = factor(Bifringence),
    Grade = factor(Grade, levels = c("mild", "moderate", "severe")),
    severity = factor(ifelse(Grade == "severe", 1, 0)) # 1 for severe, 0 for others
  )
```

```
# load data
positive <- read.csv("Amyloid_positive.csv")
negative <- read.csv("Amyloid_negative.csv")
# combined data
overall <- rbind(positive, negative)
amy <- overall %>%
  select(-c(X, X28, X42, X22, X11, X14, X8, X16)) %>%  # delete the columns that are not informative
  mutate(
    Amyloid = factor(ifelse(Amyloid == "Y", 1, 0)), # 1 for Y, 0 for N
    Laterality = factor(Laterality),
    Sex = factor(Sex),
    Race = factor(ifelse(Race == "White", 1, 0)), # 1 for white, 0 for else
    Monoclonal.Gammopathy = factor(Monoclonal.Gammopathy),
    Rheumatoid.Arthritis = factor(Rheumatoid.Arthritis),
    Coronary.Artery.Disease = factor(Coronary.Artery.Disease),
    Afib = factor(Afib),
    Degenerative.Spine.Disease = factor(Degenerative.Spine.Disease),
```

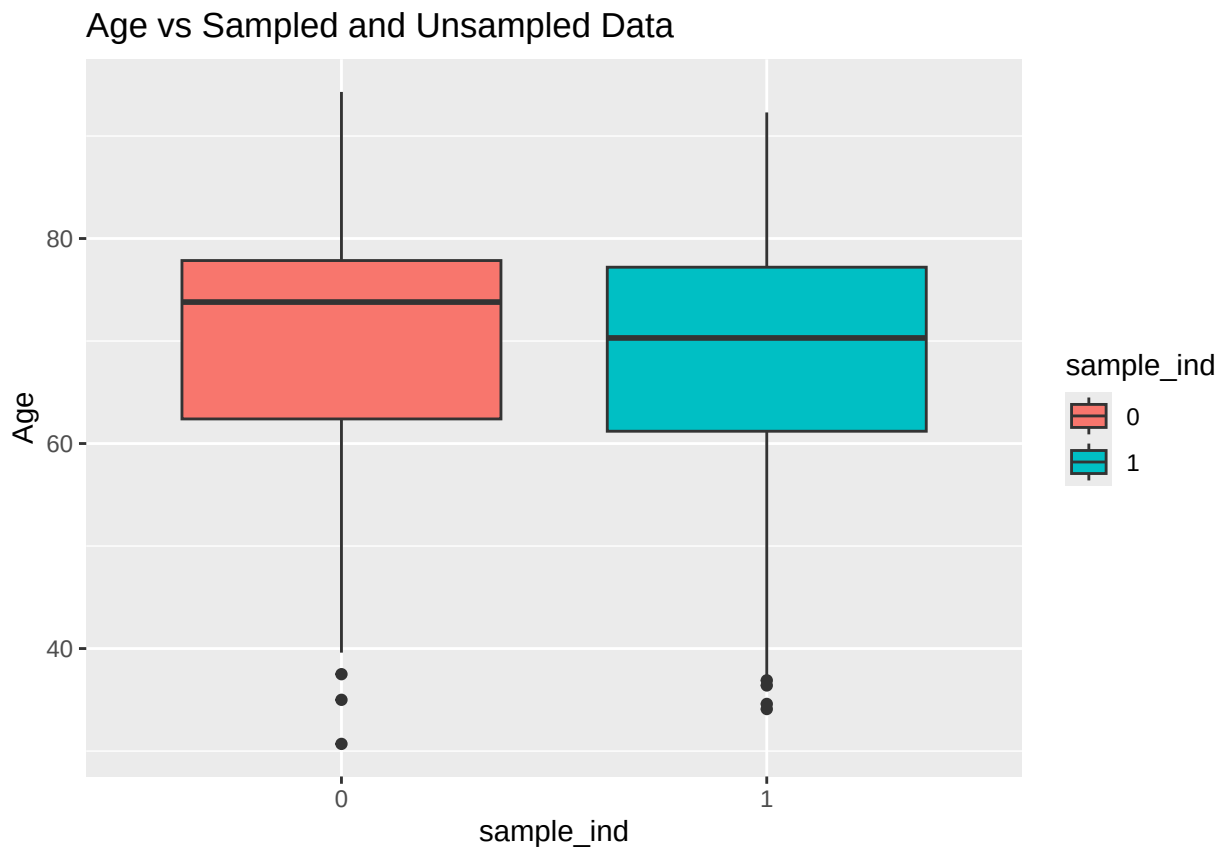
```
    Diabetes = factor(Diabetes),
    Tendinopathy = factor(Tendinopathy),
    EMG = factor(EMG),
    Bilateral. = factor(Bilateral.),
    Bifringence = factor(Bifringence),
    Grade = factor(Grade, levels = c("mild", "moderate", "severe")),
    severity = factor(ifelse(Grade == "severe", 1, 0)), # 1 for severe, 0 for others,
    sample_ind = factor(ifelse(1:nrow(overall) <= 189, 1, 0))
  )
```

EDA

Age Vs Sampled and Unsampled Data

```
# boxplot of Age vs sample indicator
ggplot(amyloid, aes(x = sample_ind, y = Age, fill = sample_ind)) +
  geom_boxplot() +
  ggtitle("Age vs Sampled and Unsampled Data")
```

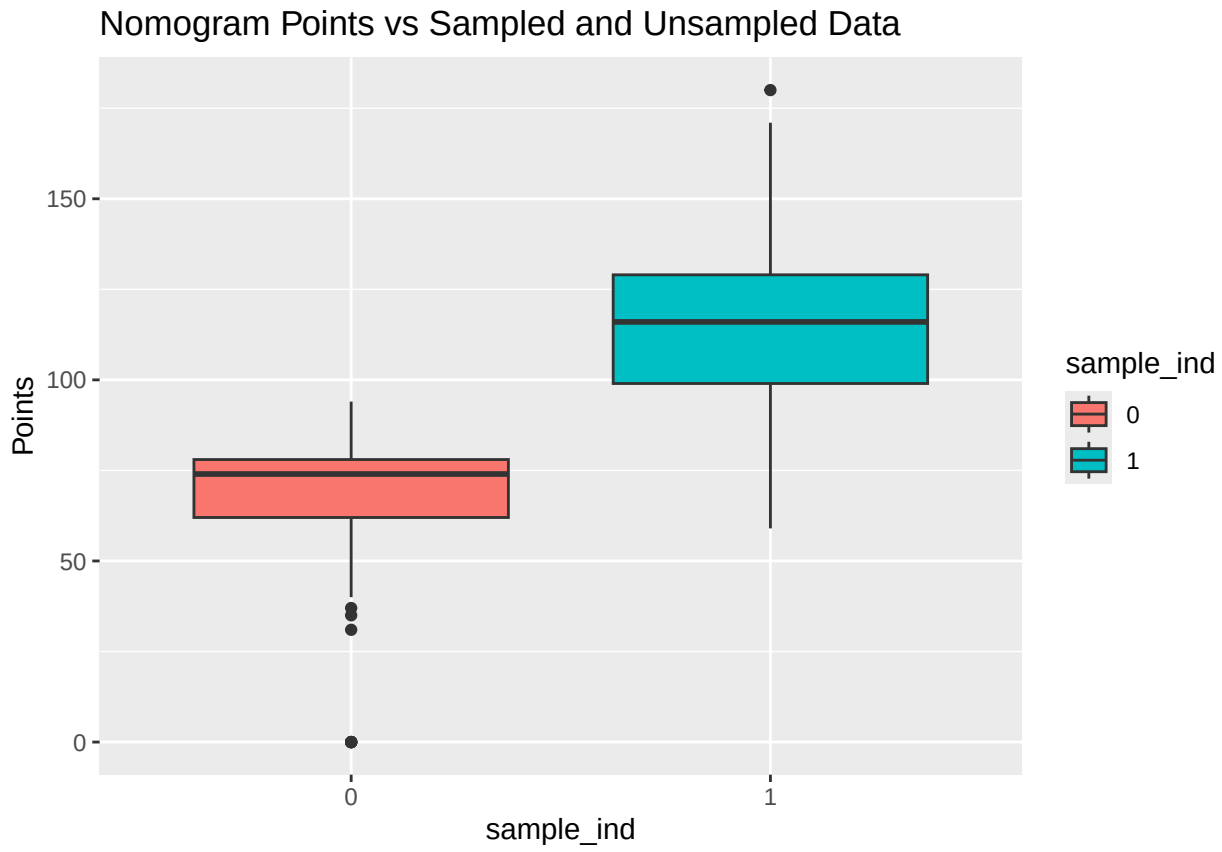
```
## Warning: Removed 6 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



The above graph is the box plot of patients' age Vs whether the data is sampled. The median age for the unsampled groups is slightly above 70, where the sampled group has the median age near 70. There seems having no big difference between the two groups.

Nomogram Points Vs Sampled and Unsampled Data

```
# boxplot of Nomogram points vs sample indicator
ggplot(amyloid, aes(x = sample_ind, y = Points, fill = sample_ind)) +
  geom_boxplot() +
  ggtitle("Nomogram Points vs Sampled and Unsampled Data")
```



The above graph is the box plot of Nomogram Points Vs whether the data is sampled. The median nomogram points for the unsampled groups is near 75, where the sampled group has the median points above 100. There seems having a big difference between the two groups. This might indicate a selection bias where the sampling tends to select patients with higher nomogram points. This sample might not represent all the population, hence further analysis would be conducted, which is IPW (Inverse Probability Weight) to see if this is needed to correct the selection bias.

What is Inverse Probability Weight?

Inverse probability weight is a strategy to make correction for the selection bias. The idea behind it is to assign a weight to each observation where the weight is equal to the inverse of the selection probability. The selection probability here refers to the probability of being included in the sample. The weighting adjustment

for each observation of the sample might better represent the target population, which in this case is the VA patients with carpal tunnel syndrome. To calculate the selection probability, a logistic regression will be fitted with the response variable sample indicator and the independent variable nomogram points.

```
# logistic regression for the selection probability
prob_select <- glm(sample_ind ~ Points, data = amyloid, family = binomial)
# extract the probability
amyloid$p_sampled <- predict(prob_select, type = "response")
sampled <- amyloid[amyloid$sample_ind == 1, ]  # sampled data
# IPW
sampled$weight <- 1 / sampled$p_sampled

# model
final_model <- glm(Amyloid ~ Points + severity, data = sampled, weights = weight,
                   family = binomial)
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(final_model)
```

```
##
## Call:
## glm(formula = Amyloid ~ Points + severity, family = binomial,
##     data = sampled, weights = weight)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.205878   0.966235  -9.528  < 2e-16 ***
## Points       0.055100   0.007258   7.592 3.15e-14 ***
## severity1    1.561050   0.450111   3.468 0.000524 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 272.08  on 179  degrees of freedom
## Residual deviance: 164.40  on 177  degrees of freedom
##   (9 observations deleted due to missingness)
## AIC: 168.64
##
## Number of Fisher Scoring iterations: 6
```