test1

Chen Chen

2025-09-17

Import data

```
head(data)
```

```
## # A tibble: 6 x 4
##   subject_id treatment memory_baseline memory_6_months
##        <dbl>     <dbl>           <dbl>           <dbl>
## 1          1         1              15              15
## 2          2         1              12              13
## 3          3         1               7              10
## 4          4         1              13               8
## 5          5         0              13              13
## 6          6         1              12              14
```

Data Preprocessing

To handle extreme values in the baseline data (0 or 15 successes), applying a constant adjustment of 0.5 to all counts. The adjusted success rate was then calculated as (number of successes + 0.5) / (total number of trials + 1). The motivation of (total number of trials + 1) is to deal with the situation of 15 successes.

```
n_trials <- 15

# data for model A
data$baseline_prop <- data$memory_baseline / n_trials
data$month_6_prop <- data$memory_6_months / n_trials

# data for madel B
data$p_baseline_smoothed <- (data$memory_baseline + 0.5) / (n_trials+1)

# log-odds
data$baseline_logodds <- log(data$p_baseline_smoothed / (1 - data$p_baseline_smoothed))

summary(data$baseline_prop)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3333  0.6667  0.8000  0.7733  0.8667  1.0000
```

```r
summary(data$month_6_prop)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.7500  0.8667  0.8227  0.9333  1.0000
```

```r
summary(data$baseline_logodds)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.6466  0.6466  1.2730  1.3531  1.6864  3.4340
```

Fitting models

In this part, I fitted three models: a logistic model with untransformed baseline proportion, a logistic model with log-odds-transformed baseline proportion, and a spline model. The results indicate that the baseline covariate is statistically significant in all three models. However, the treatment effect—which is our primary variable of interest—was not statistically significant in any of the models. I suspect that the small sample size may have limited the statistical power to detect a meaningful treatment effect. # ModelA-nontransformed

```r
modelA <- glm(cbind(memory_6_months, n_trials - memory_6_months) ~
                treatment + baseline_prop,
              family = binomial(link = "logit"),
              data = data)

summary(modelA)
```

```
##
## Call:
## glm(formula = cbind(memory_6_months, n_trials - memory_6_months) ~
##     treatment + baseline_prop, family = binomial(link = "logit"),
##     data = data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.29282    0.43197  -0.678    0.498
## treatment      0.07747    0.19407   0.399    0.690
## baseline_prop  2.37369    0.55486   4.278 1.89e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188.69  on 49  degrees of freedom
## Residual deviance: 170.47  on 47  degrees of freedom
## AIC: 274.03
##
## Number of Fisher Scoring iterations: 5
```

ModelB-logodds

```
modelB <- glm(cbind(memory_6_months, n_trials - memory_6_months) ~
                treatment + baseline_logodds,
              family = binomial(link = "logit"),
              data = data)

summary(modelB)
```

```
##
## Call:
## glm(formula = cbind(memory_6_months, n_trials - memory_6_months) ~
##     treatment + baseline_logodds, family = binomial(link = "logit"),
##     data = data)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.99545    0.18365   5.420 5.95e-08 ***
## treatment          0.06316    0.19362   0.326    0.744
## baseline_logodds   0.41091    0.10503   3.912 9.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188.69  on 49  degrees of freedom
## Residual deviance: 172.37  on 47  degrees of freedom
## AIC: 275.93
##
## Number of Fisher Scoring iterations: 5
```

Model C-spline

```
modelC <- gam(cbind(memory_6_months, n_trials - memory_6_months) ~
                treatment + s(baseline_prop),
              family = binomial(link = "logit"),
              method = "REML",
              data = data)

summary(modelC)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## cbind(memory_6_months, n_trials - memory_6_months) ~ treatment +
##     s(baseline_prop)
##
```

```
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.5069     0.1423   10.59   <2e-16 ***
## treatment     0.1466     0.1979    0.74    0.459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq  p-value
## s(baseline_prop) 3.516  4.385  27.86 2.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.133   Deviance explained = 16.5%
## -REML = 134.84  Scale est. = 1         n = 50
```

```
treatment_effect_A <- summary(modelA)$coefficients["treatment", ]
treatment_effect_B <- summary(modelB)$coefficients["treatment", ]


treatment_effect_C <- summary(modelC)$p.table["treatment", ]


treatment_comparison <- data.frame(
  Model = c("Model A", "Model B", "Model C"),
  Estimate = c(treatment_effect_A["Estimate"], treatment_effect_B["Estimate"], treatment_effect_C["Esti
  Std_Error = c(treatment_effect_A["Std. Error"], treatment_effect_B["Std. Error"], treatment_effect_C[
  p_value = c(treatment_effect_A["Pr(>|z|)"], treatment_effect_B["Pr(>|z|)"], treatment_effect_C["Pr(>|z
)
print(treatment_comparison)
```

```
##     Model   Estimate Std_Error   p_value
## 1 Model A 0.07746559 0.1940674 0.6897691
## 2 Model B 0.06316259 0.1936211 0.7442598
## 3 Model C 0.14657014 0.1979470 0.4590260
```

Comparison

In this situation, I found spline model has lowest AIC and BIC, but the differences between these three models are not obvious.

```
model_comparison <- data.frame(
  Model = c("A: Raw Proportion", "B: Log-Odds", "C: Spline"),
  AIC = c(AIC(modelA), AIC(modelB), AIC(modelC)),
  BIC = c(BIC(modelA), BIC(modelB), BIC(modelC))
)

print(model_comparison)
```

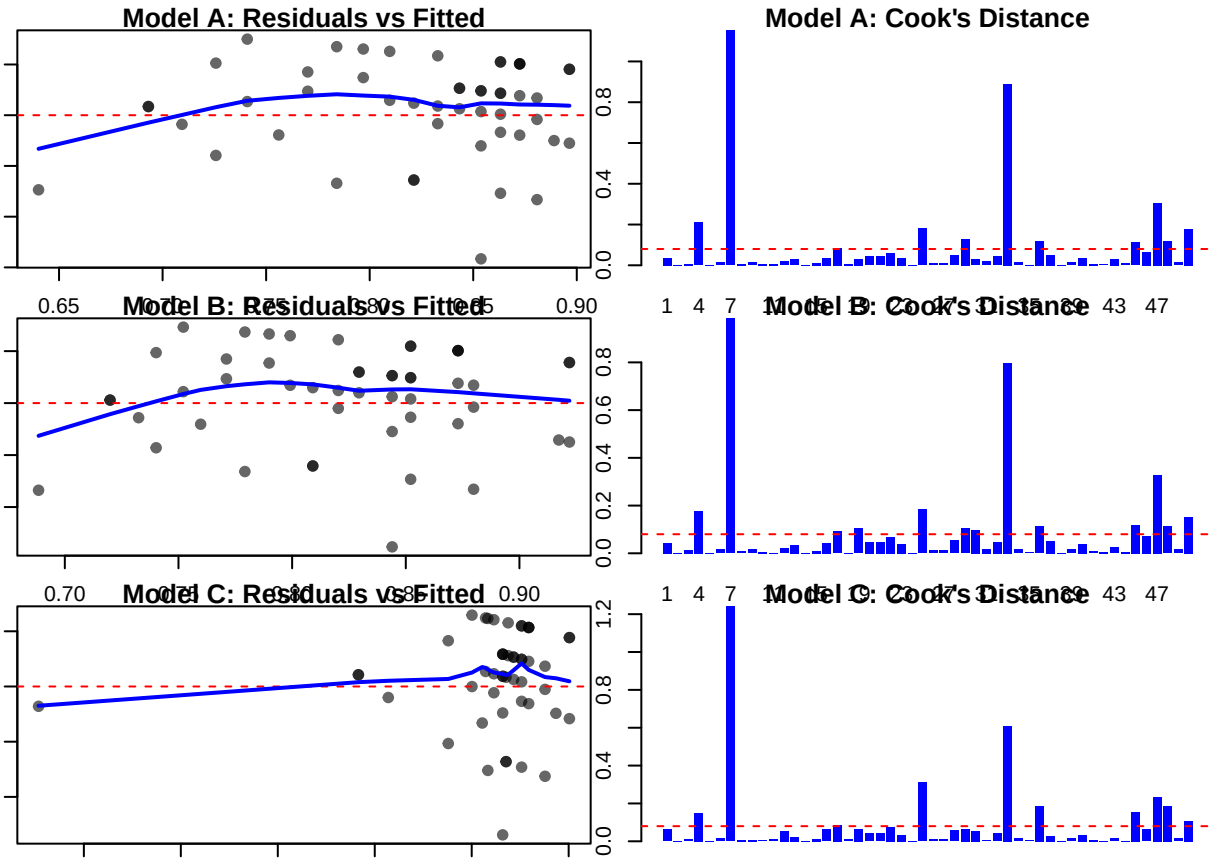
```
##               Model      AIC      BIC
## 1 A: Raw Proportion 274.0272 279.7632
## 2       B: Log-Odds 275.9310 281.6671
## 3         C: Spline 267.6111 279.6343
```

Residuals and Cook's distance graph

The figure shows that there is no significant fitting problem among the three models. The blue trend line of model C is most consistent with the red line. In the residual plot of model C, the scatter plots tend to be evenly distributed on the right side. This may be because most of the memory success rates in the data are high.

The outlier detection shows that three models have some same strong influence points.

```
# format
par(mfrow = c(3, 2), mar = c(1, 1, 1, 1))

fitted_A <- fitted(modelA)
fitted_B <- fitted(modelB)
fitted_C <- fitted(modelC)

#residuals and Cook's distance
resid_A <- resid(modelA, type = "deviance")
resid_B <- resid(modelB, type = "deviance")
resid_C <- resid(modelC, type = "deviance")

cooks_A <- cooks.distance(modelA)
cooks_B <- cooks.distance(modelB)
cooks_C <- cooks.distance(modelC)

# model A
plot(fitted_A, resid_A, main = "Model A: Residuals vs Fitted",
     xlab = "Fitted Values", ylab = "Deviance Residuals", pch = 19, col = alpha("black", 0.6))
abline(h = 0, col = "red", lty = 2)
lines(lowess(fitted_A, resid_A), col = "blue", lwd = 2)

barplot(cooks_A, main = "Model A: Cook's Distance",
        ylab = "Cook's Distance", col = "blue", border = NA)
abline(h = 4/length(cooks_A), col = "red", lty = 2)

# model B
plot(fitted_B, resid_B, main = "Model B: Residuals vs Fitted",
     xlab = "Fitted Values", ylab = "Deviance Residuals", pch = 19, col = alpha("black", 0.6))
abline(h = 0, col = "red", lty = 2)
lines(lowess(fitted_B, resid_B), col = "blue", lwd = 2)

barplot(cooks_B, main = "Model B: Cook's Distance",
        ylab = "Cook's Distance", col = "blue", border = NA)
abline(h = 4/length(cooks_B), col = "red", lty = 2)

# model C
plot(fitted_C, resid_C, main = "Model C: Residuals vs Fitted",
     xlab = "Fitted Values", ylab = "Deviance Residuals", pch = 19, col = alpha("black", 0.6))
abline(h = 0, col = "red", lty = 2)
lines(lowess(fitted_C, resid_C), col = "blue", lwd = 2)

barplot(cooks_C, main = "Model C: Cook's Distance",
        ylab = "Cook's Distance", col = "blue", border = NA)
abline(h = 4/length(cooks_C), col = "red", lty = 2)
```



Simulation study

I have some questions about data generation. Is the dataset simulated by randomly drawing from a binomial distribution? Also, how should we interpret the situation where the covariate treatment is not statistically significant? Could this simply be due to the sample size?