Working

Libraries

```r
library(lme4)
```

```
Loading required package: Matrix
```

```
Warning in check_dep_version(): ABI version mismatch:
lme4 was built with Matrix ABI version 2
Current Matrix ABI version is 1
Please re-install lme4 from source or restore original 'Matrix' package
```

```r
library(knitr)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x tidyr::expand() masks Matrix::expand()
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
x tidyr::pack()   masks Matrix::pack()
x tidyr::unpack() masks Matrix::unpack()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becon
```

```
library(glmmTMB)
```

```
Warning in checkMatrixPackageVersion(getOption("TMB.check.Matrix", TRUE)): Package version i
TMB was built with Matrix ABI version 2
Current Matrix ABI version is 1
Please re-install 'TMB' from source using install.packages('TMB', type = 'source') or ask CR
```

```
library(broom)
library(emmeans)
```

```
Welcome to emmeans.
Caution: You lose important information if you filter this package's results.
See '? untidy'
```

```
library(ggplot2)
library(kableExtra)
```

```

Attaching package: 'kableExtra'

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

    group_rows
```

```
data <- read.table("data.txt",header = TRUE, as.is = TRUE)
```

```
data$Pass <- round(data$N * data$Pct)
data$Fail <- (data$N - data$Pass)
data$timeperiod <- rep(1, nrow(data))
data$timeperiod[data$Year > 2002] <- 2
data$timeperiod[data$Year > 2010] <- 3
data$timeperiod <- factor(data$timeperiod, levels = c(1, 2, 3), labels = c("tp1", "tp2", "tp
```

```
yP <- data.frame(Year = rep(data$Year, data$Pass), Pass = rep(1, sum(data$Pass)))
yF <- data.frame(Year = rep(data$Year, data$Fail), Pass = rep(0, sum(data$Fail)))
y <- rbind(yP, yF); rm(yP, yF)
y$timeperiod <- rep(1, nrow(y))
y$timeperiod[y$Year > 2002] <- 2
y$timeperiod[y$Year > 2010] <- 3
```

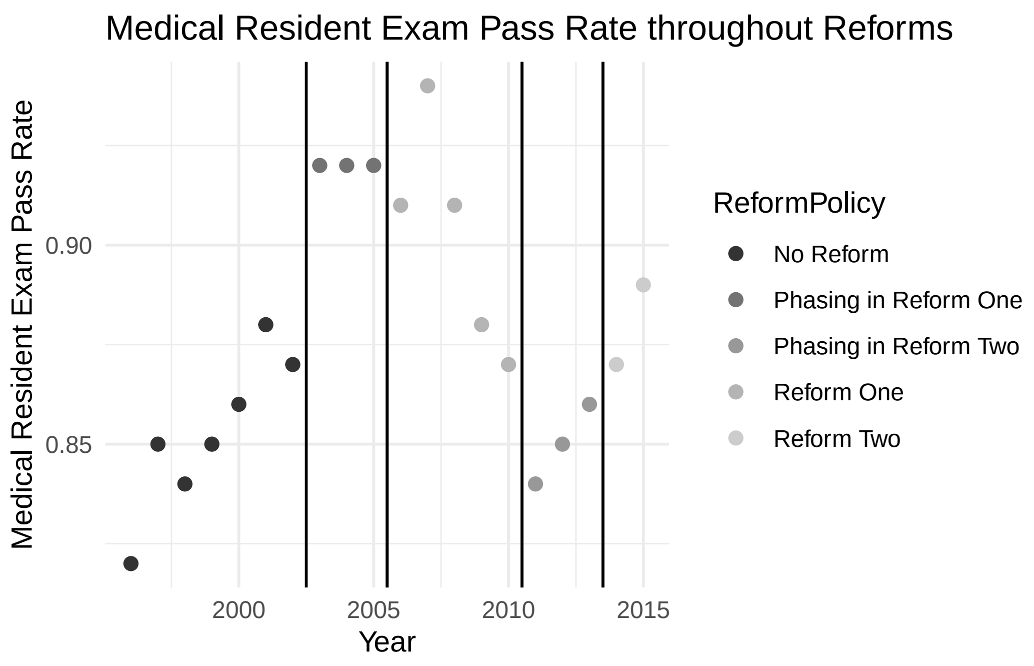
```r
y$timeperiod <- factor(y$timeperiod, levels = c(1, 2, 3), labels = c("tp1", "tp2", "tp3"))
y$timeperiod <- relevel(y$timeperiod, ref = "tp2")
data$timeperiod <- relevel(data$timeperiod, ref = "tp2")
```

```r
glm.out0 <- glm(Pass ~timeperiod, family  = binomial(link=logit), data=y)

summary(glm.out0)$coefficients
```

```
                Estimate Std. Error   z value      Pr(>|z|)
(Intercept)    2.2947928 0.01455794 157.63173  0.000000e+00
timeperiodtp1 -0.5392156 0.01928787 -27.95620 5.541470e-172
timeperiodtp3 -0.4588565 0.02088958 -21.96581 6.116367e-107
```

Exploratory Data Analysis

```r
# Plot One: Breaking Down Subsets

# Add Reform Phasing Information
reform_phase_data <- data |>
  mutate(
    ReformPolicy = case_when(
      Year < 2003 ~ "No Reform",
      Year < 2006 ~ "Phasing in Reform One",
      Year < 2011 ~ "Reform One",
      Year < 2014 ~ "Phasing in Reform Two",
      Year < 2016 ~ "Reform Two"
    )
  )

# Create the Plot
reform_phase_data |>
  ggplot(aes(x = Year, y = Pct, color = ReformPolicy)) +
  geom_point(size = 2) +
    labs(
      x = "Year",
      y = "Medical Resident Exam Pass Rate",
      title = "Medical Resident Exam Pass Rate throughout Reforms"
    ) +
  geom_vline(xintercept = c(2002.5, 2005.5, 2010.5, 2013.5)) +
  geom_vline(aes(xintercept = 2005.5)) +
  geom_vline(aes(xintercept = 2010.5)) +
  geom_vline(aes(xintercept = 2013.5)) +
  scale_color_grey() +
  theme_minimal()
```



Random Effects

Below I made a random effect for the year. This is consider a binomial mixture model.

```r
model <- glmer(
  Pass ~ timeperiod + (1 | Year),
  data = y,
  family = binomial(link = "logit")
)
```

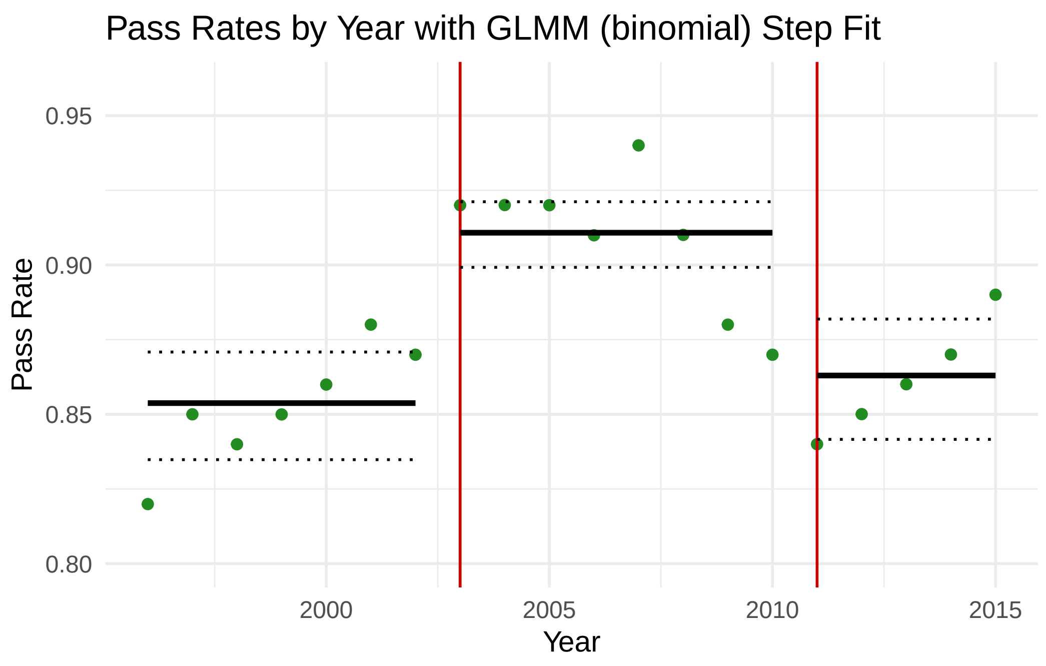
```r
summary(model)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
```

```
Formula: Pass ~ timeperiod + (1 | Year)
   Data: y

      AIC       BIC    logLik -2*log(L)  df.resid
 105933.2  105972.7  -52962.6  105925.2    143987

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9073  0.2957  0.3707  0.4194  0.4671

Random effects:
 Groups Name        Variance Std.Dev.
 Year   (Intercept) 0.0371   0.1926
Number of obs: 143991, groups:  Year, 20

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.32341    0.06885  33.744  < 2e-16 ***
timeperiodtp1 -0.55911    0.10042  -5.568 2.58e-08 ***
timeperiodtp3 -0.48314    0.11034  -4.379 1.19e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) tmprd1
timeperdtp1 -0.681
timeperdtp3 -0.617  0.420
```

```r
emm <- emmeans(model, ~ timeperiod, type = "response", re.form = NA)
step <- as.data.frame(emm) %>%
  mutate(period = c("tp2","tp1","tp3")) %>%
  arrange(period)

nm_mean  <- intersect(names(step), c("response","prob","emmean"))[1]
nm_lower <- intersect(names(step), c("lower.CL","asymp.LCL"))[1]
nm_upper <- intersect(names(step), c("upper.CL","asymp.UCL"))[1]

step$response <- step[[nm_mean]]
step$lower.CL <- step[[nm_lower]]
step$upper.CL <- step[[nm_upper]]

step$xmin <- c(1996, 2003, 2011)
```

```r
step$xmax <- c(2002, 2010, 2015)

ggplot(data, aes(x = Year, y = Pass/N)) +
  geom_point(color = "forestgreen") +
  geom_vline(xintercept = c(2003, 2011), color = "red3") +
  geom_segment(data = step,
               aes(x = xmin, xend = xmax, y = response, yend = response),
               inherit.aes = FALSE, color = "black", linewidth = 1) +
  geom_segment(data = step,
               aes(x = xmin, xend = xmax, y = lower.CL, yend = lower.CL),
               inherit.aes = FALSE, color = "black", linetype = "dotted") +
  geom_segment(data = step,
               aes(x = xmin, xend = xmax, y = upper.CL, yend = upper.CL),
               inherit.aes = FALSE, color = "black", linetype = "dotted") +
  labs(title = "Pass Rates by Year with GLMM (binomial) Step Fit",
       y = "Pass Rate") +
  coord_cartesian(ylim = c(0.80, 0.96)) +
  theme_minimal()
```



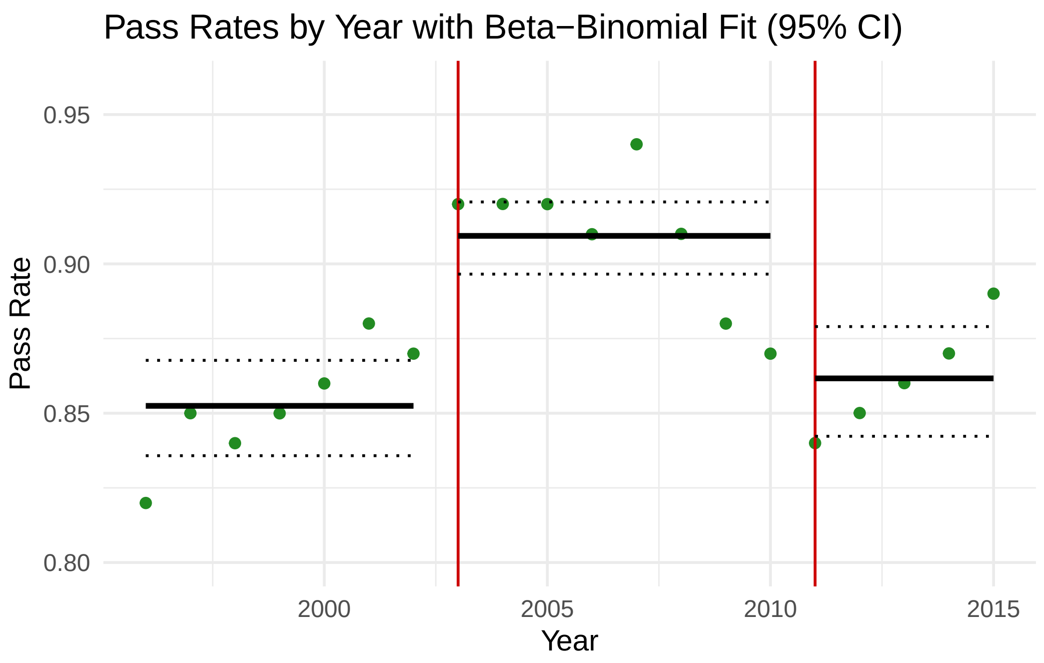
Beta-Binomial

```r
beta_binomial_model <- glmmTMB(
  cbind(Pass, Fail) ~ timeperiod,
  data = data,
  family = betabinomial(link = "logit")
)

summary(beta_binomial_model)
```

```
 Family: betabinomial  ( logit )
Formula:          cbind(Pass, Fail) ~ timeperiod
Data: data

      AIC       BIC    logLik -2*log(L)  df.resid
    262.6     266.6    -127.3     254.6        16


Dispersion parameter for betabinomial family ():  280

Conditional model:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.30613    0.07465  30.891  < 2e-16 ***
timeperiodtp1 -0.55215    0.09863  -5.598 2.16e-08 ***
timeperiodtp3 -0.47694    0.10817  -4.409 1.04e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
period_map <- data.frame(
  timeperiod = factor(c("tp1","tp2","tp3"), levels = levels(data$timeperiod)),
  xmin = c(1996, 2003, 2011),
  xmax = c(2002, 2010, 2015)
)

newdat <- tibble(timeperiod = c("tp1","tp2","tp3")) |>
  left_join(period_map, by = "timeperiod")

pred_link <- predict(beta_binomial_model, newdata = newdat, type = "link", se.fit = TRUE)
newdat$fit <- plogis(pred_link$fit)
newdat$lo  <- plogis(pred_link$fit - 1.96 * pred_link$se.fit)
newdat$hi  <- plogis(pred_link$fit + 1.96 * pred_link$se.fit)
```

```r
# plot
ggplot(data, aes(x = Year, y = Pass/N)) +
  geom_point(color = "forestgreen") +
  geom_vline(xintercept = c(2003, 2011), color = "red3") +
  geom_segment(data = newdat, aes(x = xmin, xend = xmax, y = fit, yend = fit),
               inherit.aes = FALSE, color = "black", linewidth = 1) +
  geom_segment(data = newdat, aes(x = xmin, xend = xmax, y = lo, yend = lo),
               inherit.aes = FALSE, color = "black", linetype = "dotted") +
  geom_segment(data = newdat, aes(x = xmin, xend = xmax, y = hi, yend = hi),
               inherit.aes = FALSE, color = "black", linetype = "dotted") +
  labs(title = "Pass Rates by Year with Beta-Binomial Fit (95% CI)",
       y = "Pass Rate") +
  coord_cartesian(ylim = c(0.80, 0.96)) +
  theme_minimal()
```



Beta Binomial With Subset

```r
# Grouping Subsets

# Reform Data
```

```r
reform_data <- data %>%
  filter(Year < 2003 | (Year >= 2006 & Year <= 2010) | (Year >= 2014 & Year <= 2015))
reform_data$timeperiod <- factor(reform_data$timeperiod, levels = c("tp1","tp2","tp3"))
reform_data$timeperiod <- relevel(reform_data$timeperiod, ref = "tp2")
```

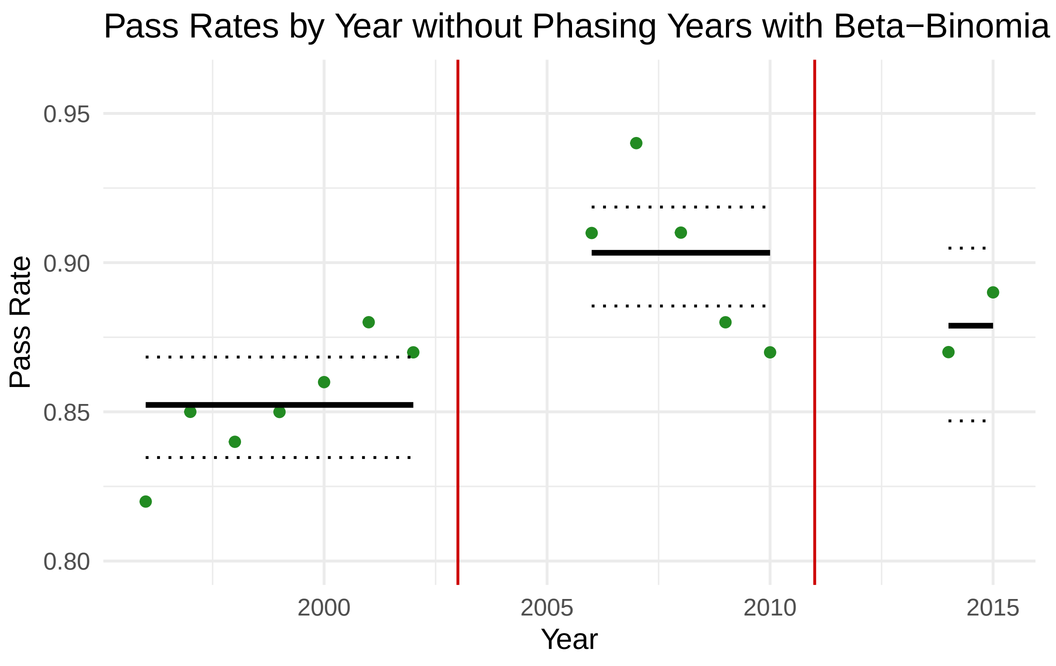
Creating Beta Binomial Model with Subsets

```r
beta_binomial_model_subsets <- glmmTMB(
  cbind(Pass, Fail) ~ timeperiod,
  data = reform_data,
  family = betabinomial(link = "logit")
)

summary(beta_binomial_model_subsets)
```

```
 Family: betabinomial  ( logit )
Formula:          cbind(Pass, Fail) ~ timeperiod
Data: reform_data

      AIC       BIC    logLik -2*log(L)  df.resid
    188.2     190.7     -90.1     180.2        10


Dispersion parameter for betabinomial family ():  251

Conditional model:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.23459    0.09658  23.137  < 2e-16 ***
timeperiodtp1 -0.48172    0.11798  -4.083 4.45e-05 ***
timeperiodtp3 -0.25281    0.16833  -1.502    0.133
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
period_map_sub <- tibble(
  timeperiod = factor(c("tp1","tp2","tp3"), levels = levels(reform_data$timeperiod)),
  xmin = c(1996, 2006, 2014),
  xmax = c(2002, 2010, 2015)
)
```

```r
# Keep only the periods that actually exist in reform_data
present <- unique(reform_data$timeperiod)
period_map_sub <- semi_join(period_map_sub,
                                  tibble(timeperiod = present),
                                  by = "timeperiod")

# 5) Predict on the link scale for present periods, then transform to response
newdat <- select(period_map_sub, timeperiod)
pred_link <- predict(beta_binomial_model_subsets, newdata = newdat, type = "link", se.fit = 

newdat$fit <- plogis(pred_link$fit)
newdat$lo  <- plogis(pred_link$fit - 1.96 * pred_link$se.fit)
newdat$hi  <- plogis(pred_link$fit + 1.96 * pred_link$se.fit)

# Add plotting spans for the *subset* ranges
newdat <- dplyr::left_join(newdat, period_map_sub, by = "timeperiod")

ggplot(reform_data, aes(x = Year, y = Pass/N)) +
  geom_point(color = "forestgreen") +
  geom_vline(xintercept = c(2003, 2011), color = "red3") +

  geom_segment(data = newdat,
               aes(x = xmin, xend = xmax, y = fit, yend = fit),
               inherit.aes = FALSE, color = "black", linewidth = 1) +
  geom_segment(data = newdat,
               aes(x = xmin, xend = xmax, y = lo, yend = lo),
               inherit.aes = FALSE, color = "black", linetype = "dotted") +
  geom_segment(data = newdat,
               aes(x = xmin, xend = xmax, y = hi, yend = hi),
               inherit.aes = FALSE, color = "black", linetype = "dotted") +

  labs(title = "Pass Rates by Year without Phasing Years with Beta-Binomial Fit (95% CI)",
       y = "Pass Rate") +
  coord_cartesian(ylim = c(0.80, 0.96)) +
  theme_minimal()
```



~~~
## Goodness of Fit


::: {.cell}

```{.r .cell-code}
AIC(model, beta_binomial_model, beta_binomial_model_subsets)


Warning in AIC.default(model, beta_binomial_model,
beta_binomial_model_subsets): models are not all fitted to the same number of
observations


                            df         AIC
model                        4 105933.2270
beta_binomial_model          4    262.6229
beta_binomial_model_subsets  4    188.1705


:::
~~~

```r
library(dplyr)
library(knitr)
library(kableExtra)

# Extract AICs
aic_vals <- AIC(model, beta_binomial_model, beta_binomial_model_subsets)
```

```
Warning in AIC.default(model, beta_binomial_model,
beta_binomial_model_subsets): models are not all fitted to the same number of
observations
```

```r
# Convert to data frame
aic_tbl <- as.data.frame(aic_vals) %>%
  rename(Model = 1, AIC = 2) %>%
  mutate(AIC = ifelse(Model == "beta_binomial_model_subsets",
                      paste0(round(AIC, 2), " *"),
                      round(AIC, 2)))

# Make a styled table
kable(aic_tbl, align = "c",
      caption = "AIC values across models (* = cannot compare, different n)") %>%
  kable_styling(full_width = FALSE, position = "center")
```

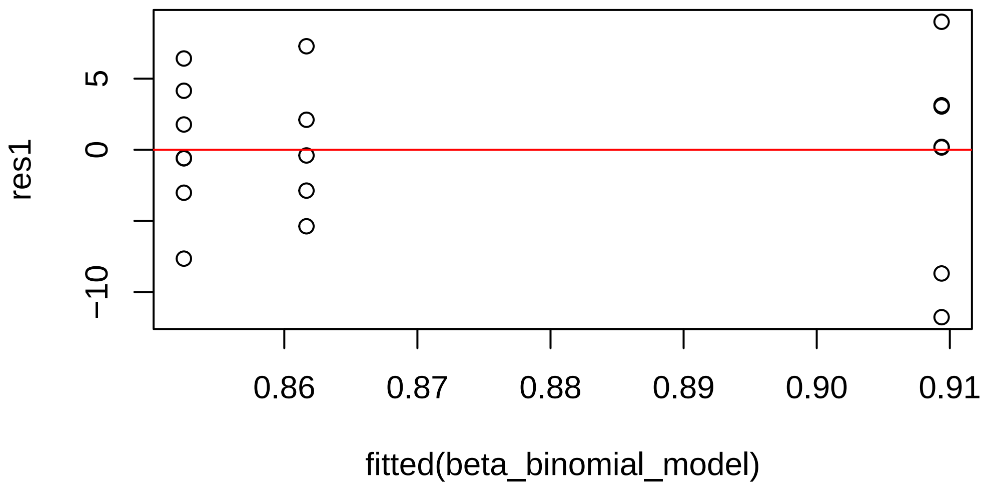
Table 1: AIC values across models (* = cannot compare, different n)

	Model	AIC
model	4	105933.23
beta_binomial_model	4	262.62
beta_binomial_model_subsets	4	188.17

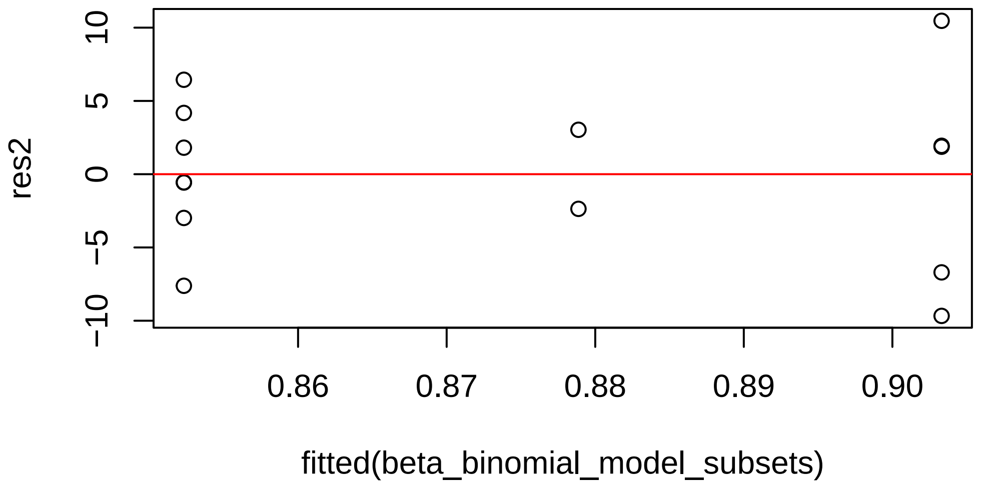
```r
#residuals(model, type = "pearson")
```

```r
res1 <- resid(beta_binomial_model, type = "pearson")
res2 <- resid(beta_binomial_model_subsets, type = "pearson")
```

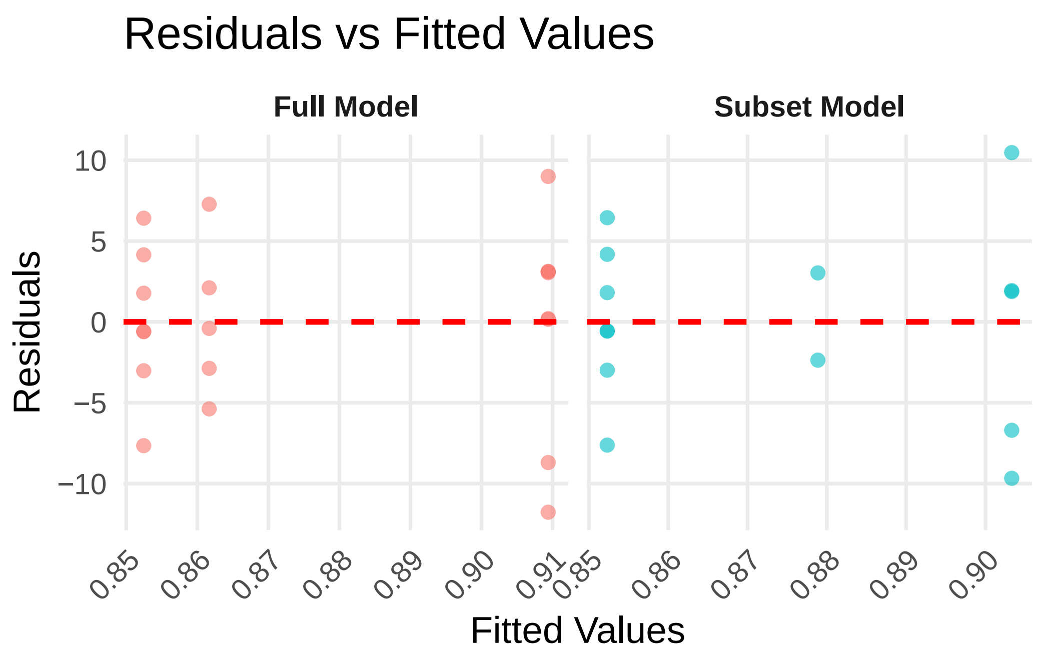
```r
plot(fitted(beta_binomial_model), res1)
abline(h=0, col="red")
```



```
plot(fitted(beta_binomial_model_subsets), res2)
abline(h=0, col="red")
```



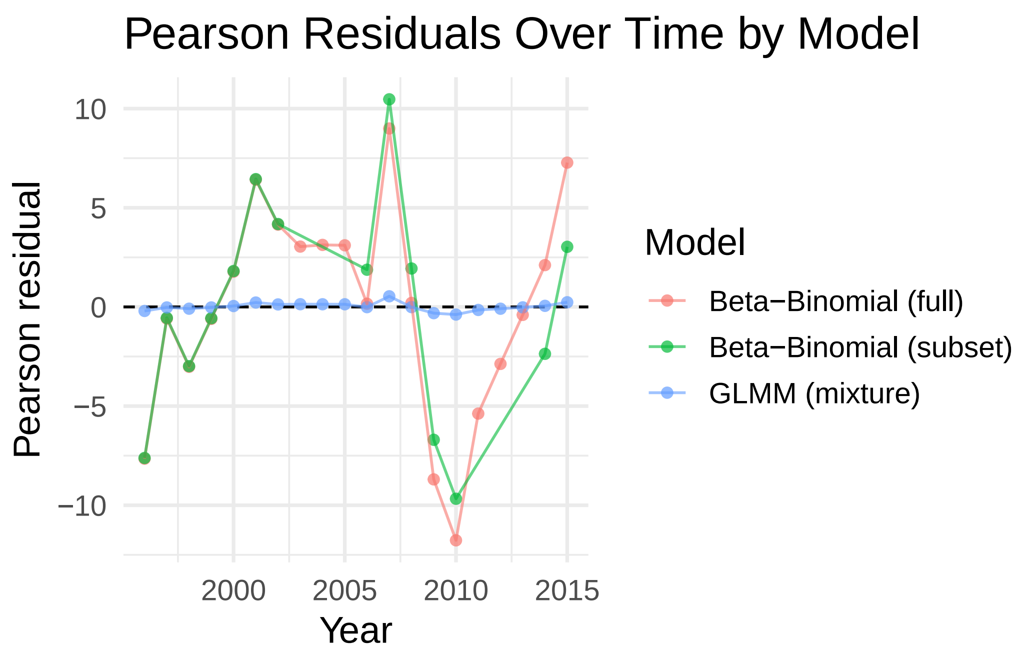
```
df<-data.frame(fitted =c(fitted(beta_binomial_model),fitted(beta_binomial_model_subsets)),res

ggplot(df, aes(x = fitted, y = residuals, color = model)) +
  geom_point(alpha = 0.6, size = 2) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", linewidth = 1) +
  facet_wrap(~ model, scales = "free_x") +
  theme_minimal(base_size = 14) +
  theme(
    panel.grid.minor = element_blank(),
    legend.position = "none",
    strip.text = element_text(face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1)  # tilt x-axis labels
  ) +
  labs(
    title = "Residuals vs Fitted Values",
    x = "Fitted Values",
    y = "Residuals"
  )
```



```
## 1) Beta-binomial (full data): per-row Pearson residuals already at 'data' level
res_bb <- residuals(beta_binomial_model, type = "pearson")
bb_df <- data %>%
```

```r
  mutate(resid = as.numeric(res_bb),
         model = "Beta-Binomial (full)") %>%
  select(Year, resid, model)

## 2) Beta-binomial (subset): per-row residuals on 'reform_data'
res_bbs <- residuals(beta_binomial_model_subsets, type = "pearson")
bbs_df <- reform_data %>%
  mutate(resid = as.numeric(res_bbs),
         model = "Beta-Binomial (subset)") %>%
  select(Year, resid, model)

## 3) GLMM "mixture" (binomial with random Year effect):
##    Compute Year-level Pearson residuals by aggregating individual-level predictions.
y$fit_p <- predict(model, type = "response", re.form = NULL)

mix_year <- y %>%
  group_by(Year) %>%
  summarise(
    observed = sum(Pass),
    expected = sum(fit_p),
    var = sum(fit_p * (1 - fit_p)),
    .groups = "drop"
  ) %>%
  mutate(
    var = pmax(var, 1e-8),  # guard against division by ~0
    resid = (observed - expected) / sqrt(var),
    model = "GLMM (mixture)"
  ) %>%
  select(Year, resid, model)

## Combine all models (note: subset model will have gaps for omitted years)
res_long <- bind_rows(bb_df, bbs_df, mix_year) %>%
  arrange(model, Year)
```

```r
ggplot(res_long, aes(x = Year, y = resid, color = model)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point(alpha = 0.7) +
  geom_line(alpha = 0.6) +
  labs(
    title = "Pearson Residuals Over Time by Model",
    y = "Pearson residual",
    color = "Model"
  ) +
  theme_minimal(base_size = 14)
```



```r
library(ggplot2)
library(dplyr)

# Use res_long from before
# Pairwise comparisons
res_full_vs_subset <- res_long %>%
  filter(model %in% c("Beta-Binomial (full)", "Beta-Binomial (subset)"))

res_full_vs_mix <- res_long %>%
  filter(model %in% c("Beta-Binomial (full)", "GLMM (mixture)"))

# Define consistent palette
model_colors <- c(
  "Beta-Binomial (full)" = "#1b9e77",
  "Beta-Binomial (subset)" = "#d95f02",
  "GLMM (mixture)" = "#7570b3"
)
```

```r
# Comparison 1: Full vs Subset Beta-Binomial
p1 <- ggplot(res_full_vs_subset, aes(x = Year, y = resid, color = model)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey40") +
  geom_point(size = 2, alpha = 0.7) +
  geom_line(linewidth = 1, alpha = 0.7) +
  scale_color_manual(values = model_colors) +
  labs(
    title = "Residuals Over Time: Full vs Subset Beta-Binomial",
    y = "Pearson residual", x = "Year", color = "Model"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "top",
    panel.grid.minor = element_blank(),
    plot.title = element_text(face = "bold")
  )

# Comparison 2: Full vs Mixture
p2 <- ggplot(res_full_vs_mix, aes(x = Year, y = resid, color = model)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey40") +
  geom_point(size = 2, alpha = 0.7) +
  geom_line(linewidth = 1, alpha = 0.7) +
  scale_color_manual(values = model_colors) +
  labs(
    title = "Residuals Over Time: Beta-Binomial vs GLMM Mixture",
    y = "Pearson residual", x = "Year", color = "Model"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "top",
    panel.grid.minor = element_blank(),
    plot.title = element_text(face = "bold")
  )
```

```r
# Save first plot: Full vs Subset Beta-Binomial
ggsave("residuals_full_vs_subset.png",
       plot = p1,
       width = 8, height = 5, dpi = 300)

# Save second plot: Full vs Mixture
ggsave("residuals_full_vs_mixture.png",
       plot = p2,
       width = 8, height = 5, dpi = 300)
```