---

title: "MED33 (Versiov 4 anaylysis) - using data up to April 2025"

author: "Isa Akbarzadeh"

output:

 html_document:

  theme: cerulean

  toc: true

  toc_float: true

  code_folding: hide

```{r setup, include=FALSE}

knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)

library(readxl)

library(dplyr)

library(tidyr)

library(lubridate)

library(geepack)

library(ggplot2)

library(kableExtra)

library(shiny)

```

```{r data-import, include=FALSE}

df <- read_excel("C:/Users/NightMan/Downloads/MED33 8th_april_2025_update.xlsx")

baseline_data <- read_excel("C:/Users/NightMan/Downloads/MED33 6th_jan_2025_update.xlsx", sheet = "baseline data")

#MED33 6th_jan_2025_update
```

```

```{r dataset-prep1, include=FALSE}
# Clean and format main dataset
df <- df %>%
  mutate(
    `Castor Participant ID` = as.character(`Castor Participant ID`),
    Survey = as.Date(Survey, format = "%Y.%m.%d"),
    YearMonth = format(Survey, "%Y-%m"),
    `Survey Progress` = as.numeric(as.character(`Survey Progress`))
  ) %>%
  filter(`Survey Progress` >= 95)

# Clean participant IDs
df$`Castor Participant ID` <- as.numeric(gsub("-", "", df$`Castor Participant ID`))
```

```{r dataset-prep2, include=FALSE}
# Subset for participants with at least two surveys
df_twosurveys <- df %>%
  group_by(`Castor Participant ID`) %>%
  filter(n() >= 2) %>%
  ungroup()

# Summarize survey completion and participant counts
surveys_completed <- df %>%
  group_by(`Castor Participant ID`) %>%
  summarise(Survey_Count = n())
```
```

```
monthly_counts <- df %>%
 group_by(YearMonth) %>%
 summarise(Unique_IDs = n_distinct(`Castor Participant ID`))

monthly_mean_scores <- df %>%
 group_by(YearMonth) %>%
 summarise(
  n        = sum(!is.na(score)),
  Mean_Score = mean(score, na.rm = TRUE),
  SD_Score   = sd(score, na.rm = TRUE)
 ) %>%
 arrange(YearMonth)
```

```
```{r dataset-prep3, include=FALSE}
# Prepare baseline data
baseline_data <- baseline_data %>%
 mutate(`Castor Participant ID` = as.character(`Castor Participant ID`))
baseline_data$`Castor Participant ID` <- as.numeric(gsub("-", "",
baseline_data$`Castor Participant ID`))

survey_counts_distribution <- surveys_completed %>%
    group_by(Survey_Count) %>%
    summarise(Num_Participants = n()) %>%
    arrange(Survey_Count)
```
```

```
monthly_stats <- df %>%
    group_by(YearMonth) %>%
    summarise(mean_score = mean(score, na.rm = TRUE),
              median_score = median(score, na.rm = TRUE))


  list(
    df = df,
    df_twosurveys = df_twosurveys,
    surveys_completed = surveys_completed,
    survey_counts_distribution = survey_counts_distribution,
    monthly_counts = monthly_counts,
    monthly_mean_scores = monthly_mean_scores,
    monthly_stats = monthly_stats
  )

# Merge baseline data with the main dataset
merged_df <- left_join(df, baseline_data, by = "Castor Participant ID") %>%
 mutate(
   `Date of positive test/First episode of symptoms` =
    as.Date(`Date of positive test/First episode of symptoms`, origin = "1899-12-30")
 )
```

```{r dataset-prep4, include=FALSE}
# Calculate first survey date for each participant
first_survey_df <- merged_df %>%
 group_by(`Castor Participant ID`) %>%
```

```
  summarise(First_Survey_Date = min(Survey), .groups = 'drop')

merged_df <- left_join(merged_df, first_survey_df, by = "Castor Participant ID") %>%
  mutate(
    `Long COVID Exposure Period` = as.numeric(difftime(
      First_Survey_Date,
      `Date of positive test/First episode of symptoms`,
      units = "weeks"
    )) / 4.348
  )
```

```{r dataset-prep5, include=FALSE}
# Prepare data for GEE
merged_df <- merged_df %>%
  group_by(`Castor Participant ID`) %>%
  mutate(
    First_Survey_Date = min(Survey),
    Survey_Number = as.numeric(difftime(Survey, First_Survey_Date, units = "days"))
  ) %>%
  ungroup() %>%
  select(-First_Survey_Date)

gee_df <- merged_df %>%
  select(`Castor Participant ID`, Survey_Number, score, `Long COVID Exposure Period`,  age, sex, BMI, Hosp) %>%
  rename(
    ID = `Castor Participant ID`,
```

```
  Time = Survey_Number,
  Score = score,
  Exposure = `Long COVID Exposure Period`
 ) %>%
 mutate(
  Exposure = ifelse(Exposure < 1, 1, Exposure)
 )

gee_df_clean <- gee_df %>% drop_na(Exposure, Time, sex, age, BMI, Hosp)
gee_df_clean$ID <- as.factor(gee_df_clean$ID)
gee_df_clean$Time <- as.numeric(gee_df_clean$Time)
gee_df_clean$Score <- as.numeric(gee_df_clean$Score)
gee_df_clean$Exposure <- as.numeric(gee_df_clean$Exposure)
gee_df_clean$age <- as.numeric(gee_df_clean$age)
gee_df_clean$BMI <- as.numeric(gee_df_clean$BMI)
gee_df_clean$sex <- as.factor(gee_df_clean$sex)
gee_df_clean$Hosp <- as.factor(gee_df_clean$Hosp)

cases_with_12_months <- gee_df_clean %>%
 filter(`Time` >= 350) %>%
 summarise(Count = n_distinct(`ID`))
```

Tables

Table 1. Number of completed surveys in each month

```{r summary, echo= FALSE}
kable(monthly_counts, format = "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Table 2. Number of completed surveys for each unique IDs

```{r summary2, echo= FALSE}
surveys_completed <- surveys_completed %>%
  arrange(desc(Survey_Count))
kable(surveys_completed, format = "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Table 3. Number IDs with at least 12 months of follow up

```{r summary3, echo= FALSE}
kable(cases_with_12_months, format = "html", col.names = c("Cases with ≥12 Months Follow-up")) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

\

\

\

\

Graphs

```{r vis}
ggplot(monthly_counts, aes(x = YearMonth, y = Unique_IDs)) +
    geom_line(group = 1) +
    geom_point() +
    theme_minimal() +
    labs(x = "Date (Months)", y = "Number of Completed Surveys", title = "Figure 1. Number of Completed Surveys for Each Unique ID per Month") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))


ggplot(survey_counts_distribution, aes(x = Survey_Count, y = Num_Participants)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    geom_text(aes(label = Num_Participants), vjust = -0.5, color = "black") +
    theme_minimal() +
    labs(x = "Number of Surveys Completed", y = "Number of Participants", title = "Figure 2. Number of Surveys Completed by Participants") +
    scale_x_continuous(breaks = 1:18) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))


ggplot(monthly_mean_scores, aes(x = YearMonth, y = Mean_Score)) +
    geom_line(group = 1) +
    geom_point() +
    theme_minimal() +
    labs(x = "Date (Months)", y = "Mean PAC Score", title = "Figure 3. Mean PAC Score for Each Month (Reveresed Trend)") +
    scale_y_reverse() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
ggplot(monthly_stats, aes(x = YearMonth)) +
  geom_point(aes(y = mean_score), color = "blue", size = 2, alpha = 0.5) +
  geom_point(aes(y = median_score), color = "red", size = 2, alpha = 0.5) +
  theme_minimal() +
  labs(x = "Date (Months)", y = "Score", title = "Figure 4. Mean (Blue) and Median (Red) PAC Scores for Each Month") +
  scale_y_reverse() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

Prediction Model

In these GEE models using three different algorithms, variables of exposure, time, sex, age, BMI, and having hospitalization due to COVID-19 was entered./

```{r model, include=FALSE}
gee_df <- gee_df %>%
 mutate(
  ID      = as.factor(ID),
  Time    = as.numeric(Time),
  Score   = as.numeric(Score),
  Exposure = as.numeric(Exposure),
  age     = as.numeric(age),
  BMI     = as.numeric(BMI),
  sex     = as.factor(sex),
  Hosp    = as.factor(Hosp)
 )
```

```
gee_df_clean <- gee_df %>%
 tidyr::drop_na(Exposure, Time, sex, age, BMI, Hosp, Score) %>%
 mutate(
  sex  = droplevels(sex),
  Hosp = droplevels(Hosp)
 )

# 0-variance numerics
num_vars <- c("Exposure","Time","age","BMI","Score")
sapply(gee_df_clean[num_vars], function(v) c(var=stats::var(v),
unique=length(unique(v))))

# Single-level factors after cleaning
lapply(gee_df_clean[c("sex","Hosp")], function(f) table(f, useNA="ifany"))



# Fit GEE models
model_exchangeable <- geeglm(
 Score ~ Exposure * Time + sex + age + BMI + Hosp,
 data = gee_df_clean,
 id = ID,
 family = gaussian,
 corstr = "exchangeable"
)

model_ar1 <- geeglm(
 Score ~ Exposure * Time,
 data = gee_df_clean,
```

```
 id = ID,
 family = gaussian,
 corstr = "ar1"
)

model_independence <- geeglm(
 Score ~ Exposure * Time,
 data = gee_df_clean,
 id = ID,
 family = gaussian,
 corstr = "independence"
)
```

model results: exchangeable

```{r model results1}
summary(model_exchangeable)
est <- coef(summary(model_exchangeable))
ci_exchangeable <- data.frame(
 Estimate  = est[, "Estimate"],
 SE       = est[, "Std.err"],
 CI_lower  = est[, "Estimate"] - 1.96 * est[, "Std.err"],
 CI_upper  = est[, "Estimate"] + 1.96 * est[, "Std.err"]
)
ci_exchangeable
plot(model_exchangeable$fitted.values, residuals(model_exchangeable),
    xlab = "Fitted Values", ylab = "Residuals", main = "Residuals vs. Fitted") +
 abline(h = 0, col = "red")
```

```

## model results: ar1

```{r model results2}
summary(model_ar1)
plot(model_ar1$fitted.values, residuals(model_ar1),
                    xlab = "Fitted Values", ylab = "Residuals", main = "Residuals vs. Fitted") +
 abline(h = 0, col = "red")
```

model results: independence

```{r model results3}
summary(model_independence)
plot(model_independence$fitted.values, residuals(model_independence),
                xlab = "Fitted Values", ylab = "Residuals", main = "Residuals vs. Fitted") +
 abline(h = 0, col = "red")
```

##Goodness of Fit metrics for three models

```{r QIC}
qic_results <- data.frame(
 Correlation_Structure = c("Exchangeable", "AR(1)", "Independence"),
 QIC = c(
  sum((model_exchangeable$y - model_exchangeable$fitted.values)^2),
  sum((model_ar1$y - model_ar1$fitted.values)^2),
  sum((model_independence$y - model_independence$fitted.values)^2)
 )
)
```
```

```
kable(qic_results, format = "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```