

# Homework 2

Joy Bao-Dai

## Reflection

This took me about 2-3 hours to reproduce. Overall, the data was pretty nice to work with and very clean already, with a relatively clear data dictionary attached. This was somewhat nicer than my expectations because I was expecting to have to deal with messy data. However, it didn't match my expectations in some ways because I wish that the methodology with fitting the models was described more clearly.

Figure 1 was fairly straight forward, as it was more EDA-like. The only thing that was difficult there involved making sure that the y-axis was log-scaled for the range and SD variables.

Table 2 was a different story, because it was hard to figure out exactly what the formulas were to fit the multinomial logistic regression models, as well as which transformations were used. For example, something that tripped me up was having to account for a "relative doubling" of the predictors by doing a log-transformation of the variables and then exponentiating and doubling the coefficient estimates to produce the final estimates and confidence intervals. By contrast, something that was clear and easy to follow was the researchers' decision to use a Bonferroni correction. I was able to include those easily, as it was clear how and where they included it.

Another issue with Table 2 is that while I was able to reproduce the numbers for TWA and range, I was not able to reproduce the numbers for SD. Since I was able to produce the other intervals almost exactly (with some slight variation in the hundredths decimal place due to rounding), I think that the issue is not in my code but in the study results themselves, or there was a different formula used for the SD calculations that wasn't explicitly mentioned in the study.

## Load data

```
# Load data
ddh <- read.csv(
  "https://raw.githubusercontent.com/sta440-fa23/class-files/main/homework-2/ddh.csv",
  header=TRUE)

# Rename some variables
ddh <- ddh %>%
  mutate(delayed_fn_1 = as.factor(delayed_fn))

levels(ddh$delayed_fn_1) <- c("IGF", "LGD")

# Look at data
#head(ddh)
```

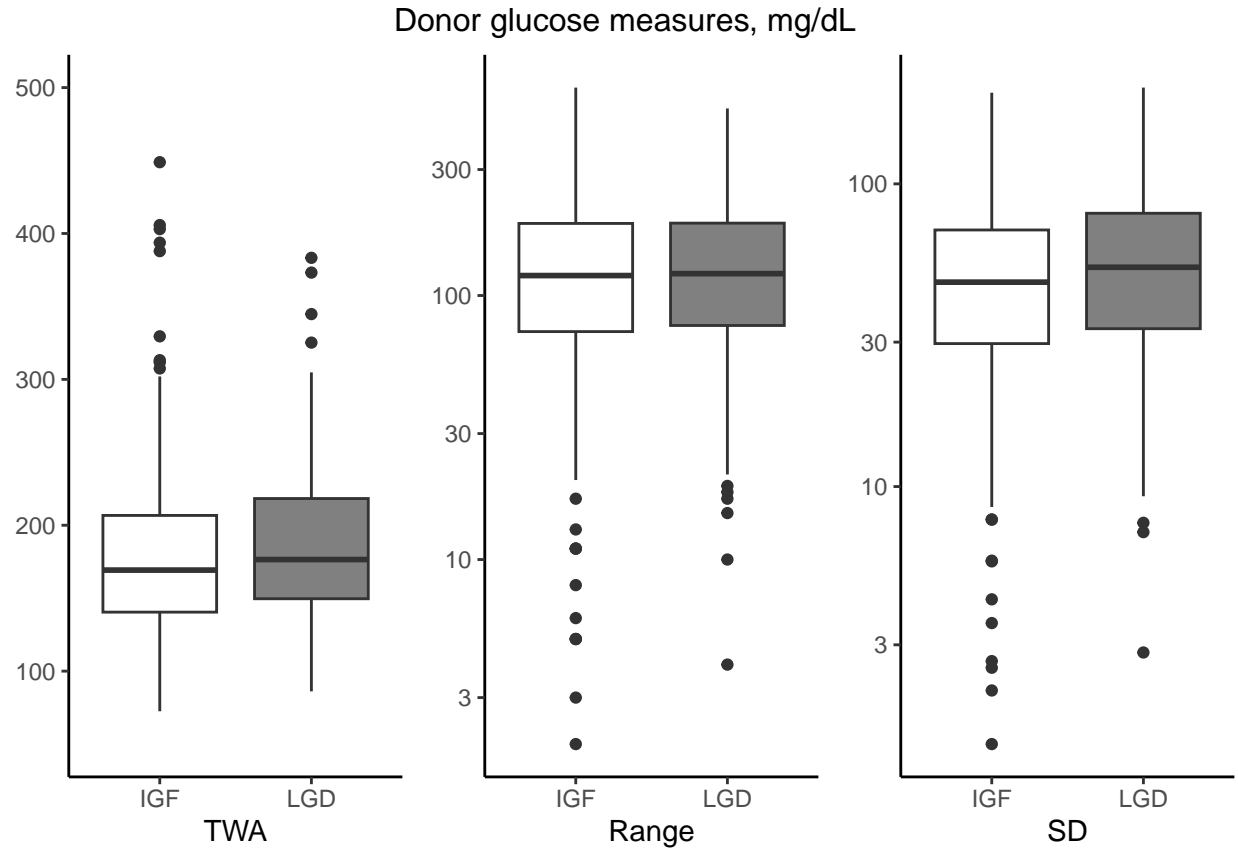
Figure 1: Deceased donor hyperglycemia and liver graft dysfunction

```
# Boxplot for TWA
twa_plot <- ggplot(data=ddh,
  mapping=aes(y=glutwa, x=delayed_fn_1, fill=delayed_fn_1)) +
  geom_boxplot() +
  xlab("TWA") +
  ylab(NULL) +
  scale_x_discrete(limits=c("IGF", "LGD")) +
  guides(fill=FALSE) +
  ylim(50, 500) +
  theme_classic() +
  scale_fill_manual(values=c("#FFFFFF", "#808080"))

# Boxplot for Range
range_plot <- ggplot(data=ddh,
  mapping=aes(y=glurange, x=delayed_fn_1, fill=delayed_fn_1)) +
  geom_boxplot() +
  xlab("Range") +
  ylab(NULL) +
  scale_x_discrete(limits=c("IGF", "LGD")) +
  guides(fill=FALSE) +
  ylim(1, 1000) +
  scale_y_log10() +
  theme_classic() +
  scale_fill_manual(values=c("#FFFFFF", "#808080"))

# Boxplot for SD
sd_plot <- ggplot(data=ddh,
  mapping=aes(y=glusd, x=delayed_fn_1, fill=delayed_fn_1)) +
  geom_boxplot() +
  xlab("SD") +
  ylab(NULL) +
  scale_x_discrete(limits=c("IGF", "LGD")) +
  guides(fill=FALSE) +
  ylim(1, 300) +
  scale_y_log10() +
  theme_classic() +
  scale_fill_manual(values=c("#FFFFFF", "#808080"))

# Print plots
grid.arrange(twa_plot, range_plot, sd_plot, ncol = 3, top='Donor glucose measures, mg/dL')
```



**Table 2: Primary exposure**

```
# Fit models
unadjusted_model <- glm(delayed_fn ~ log(glutwa),
  data = ddh,
  family = "binomial")

adjusted_model <- glm(delayed_fn ~ log(glutwa) + d_age + as.factor(d_cod) + hemo_instability,
  data = ddh,
  family = "binomial")

unadjusted_results <- tidy(unadjusted_model)
adjusted_results <- tidy(adjusted_model)

# Function to calculate results
calc_results <- function(coeff, a=0.05) {
  alpha <- a
  z_value <- qnorm(1 - alpha / 2) # Calculate z-value for CI

  # Odds Ratio and Confidence Intervals
  odds_ratio <- exp(coeff$estimate * log(2))
  lower_ci <- exp((coeff$estimate - z_value * coeff$std.error) * log(2))
  upper_ci <- exp((coeff$estimate + z_value * coeff$std.error) * log(2))
  p_value <- coeff$p.value
```

```

    return(data.frame(Odds_Ratio = odds_ratio, Lower_CI = round(lower_ci, 2), Upper_CI = round(upper_ci, 2)))
  }

  # Extracting and formatting results
  unadj_twa <- unadjusted_results %>%
    filter(term == "log(glutwa)") %>%
    calc_results(a=0.05)
  adj_twa <- adjusted_results %>%
    filter(term == "log(glutwa)") %>%
    calc_results(a=0.05)

  # Combine and format table
  results_table <- rbind(
    data.frame(Exposure = "Time-weighted average", Model = "Unadjusted", unadj_twa),
    data.frame(Exposure = "Time-weighted average", Model = "Adjusted", adj_twa)
  )

  results_table$CI <- paste0("(", results_table$Lower_CI, ", ", results_table$Upper_CI, ")")

  final_table <- results_table %>%
    select(Exposure, Model, Odds_Ratio, CI, P_Value)

  # Print the table
  kable(final_table, digits = 2, format = "latex", booktabs = TRUE) %>%
    kable_styling(latex_options = "hold_position")

```

Exposure	Model	Odds_Ratio	CI	P_Value
Time-weighted average	Unadjusted	1.48	(0.94, 2.33)	0.09
Time-weighted average	Adjusted	1.46	(0.91, 2.35)	0.11

Table 2: Secondary exposure: Range

```

# Fit models
unadjusted_model <- glm(delayed_fn ~ log(glurange),
  data = ddh,
  family = "binomial")

adjusted_model <- glm(delayed_fn ~ log(glurange) + d_age + as.factor(d_cod) + hemo_instability,
  data = ddh,
  family = "binomial")

unadjusted_results <- tidy(unadjusted_model)
adjusted_results <- tidy(adjusted_model)

# Extracting and formatting results (use same function as above)
unadj_twa <- unadjusted_results %>%
  filter(term == "log(glurange)") %>%
  calc_results(a=0.025)
adj_twa <- adjusted_results %>%

```

```

filter(term == "log(glurange)") %>%
calc_results(a=0.025)

# Combine and format table
results_table <- rbind(
  data.frame(Exposure = "Range", Model = "Unadjusted", unadj_twa),
  data.frame(Exposure = "Range", Model = "Adjusted", adj_twa)
)

results_table$CI <- paste0("(", results_table$Lower_CI, ", ", results_table$Upper_CI, ")")

final_table <- results_table %>%
  select(Exposure, Model, Odds_Ratio, CI, P_Value)

# Print the table
kable(final_table, digits = 2, format = "latex", booktabs = TRUE) %>%
  kable_styling(latex_options = "hold_position")

```

Exposure	Model	Odds_Ratio	CI	P_Value
Range	Unadjusted	1.03	(0.86, 1.24)	0.69
Range	Adjusted	1.06	(0.87, 1.28)	0.53

**Table 2: Secondary exposure: SD**

```

# Fit models
unadjusted_model <- glm(delayed_fn ~ log(glusd),
  data = ddh,
  family = "binomial")

adjusted_model <- glm(delayed_fn ~ log(glusd) + d_age + as.factor(d_cod) + hemo_instability,
  data = ddh,
  family = "binomial")

unadjusted_results <- tidy(unadjusted_model)
adjusted_results <- tidy(adjusted_model)

# Extracting and formatting results (use same function as above)
unadj_twa <- unadjusted_results %>%
  filter(term == "log(glusd)") %>%
  calc_results(a=0.025)
adj_twa <- adjusted_results %>%
  filter(term == "log(glusd)") %>%
  calc_results(a=0.025)

# Combine and format table
results_table <- rbind(
  data.frame(Exposure = "SD", Model = "Unadjusted", unadj_twa),
  data.frame(Exposure = "SD", Model = "Adjusted", adj_twa)
)

```

```

results_table$CI <- paste0("(", results_table$Lower_CI, ", ", results_table$Upper_CI, ")")

final_table <- results_table %>%
  select(Exposure, Model, Odds_Ratio, CI, P_Value)

# Print the table
kable(final_table, digits = 2, format = "latex", booktabs = TRUE) %>%
  kable_styling(latex_options = "hold_position")

```

Exposure	Model	Odds_Ratio	CI	P_Value
SD	Unadjusted	1.12	(0.91, 1.38)	0.24
SD	Adjusted	1.13	(0.91, 1.41)	0.20