# Load necessary libraries

library(dplyr)

library(haven)

library(ggplot2)

library(broom)

library(gridExtra)

library(grid)

library(gtable)

library(stringr)

# Load your dataset

data <- read\_dta("C:/Users/c23086899/OneDrive - Cardiff University/Documents/nsw\_2021-23\_anonymised\_respondent\_file.dta")

# Recode variables according to the provided mappings

data\_recoded <- data %>%

mutate(

HealthStatus = factor(case\_when(

Dvgenhealth3 == 1 ~ "Good/general Health",

Dvgenhealth3 %in% c(2, 3) ~ "Fair/bad health",

TRUE ~ NA\_character\_

)),

BMI = factor(case\_when(

Dvbmimorb2 %in% c(1, 2) ~ "Not overweight or obese",

Dvbmimorb2 == 3 ~ "Overweight",

Dvbmimorb2 == 4 ~ "Obese",

Dvbmimorb2 %in% c(-99, -98, -88, -9) ~ "Unknown",

TRUE ~ NA\_character\_

), levels = c("Not overweight or obese", "Overweight", "Obese", "Unknown")),

AlcoholConsumption = factor(case\_when(

DvCvalcusgrp == 1 ~ "None in last year/nondrinker",

DvCvalcusgrp == 2 ~ "Low risk",

DvCvalcusgrp == 3 ~ "High risk",

DvCvalcusgrp %in% c(-99, -98, -88, -9, -8) ~ "Unknown",

TRUE ~ NA\_character\_

), levels = c("None in last year/nondrinker", "Low risk", "High risk", "Unknown")),

SmokingStatus = factor(case\_when(

Dvsmokstat == 3 ~ "Never smoked",

Dvsmokstat == 2 ~ "Former smoker",

Dvsmokstat == 1 ~ "Current smoker",

Dvsmokstat %in% c(-99, -98, -88, -9, -8) ~ "Unknown",

TRUE ~ NA\_character\_

), levels = c("Never smoked", "Former smoker", "Current smoker", "Unknown")),

PhysicalActivity = factor(case\_when(

Dvmvpagrp2 == 1 ~ "Meets MVPA Guidelines",

Dvmvpagrp2 == 2 ~ "Low or some activity",

Dvmvpagrp2 == 3 ~ "Inactive",

TRUE ~ NA\_character\_

), levels = c("Meets MVPA Guidelines", "Low or some activity", "Inactive"))

)

# Calculate N for each category manually

n\_counts <- list(

BMI = table(data\_recoded$BMI, useNA = "ifany"),

AlcoholConsumption = table(data\_recoded$AlcoholConsumption, useNA = "ifany"),

SmokingStatus = table(data\_recoded$SmokingStatus, useNA = "ifany"),

PhysicalActivity = table(data\_recoded$PhysicalActivity, useNA = "ifany")

)

# Fit the GLM model

GLM\_Object <- glm(HealthStatus == "Fair/bad health" ~ BMI + AlcoholConsumption + SmokingStatus + PhysicalActivity,

data = data\_recoded, family = binomial, na.action = na.exclude)

# Get model summary

model\_summary <- tidy(GLM\_Object, conf.int = TRUE, exponentiate = TRUE)

# Helper function to map variable names

map\_variable\_name <- function(name) {

switch(name,

"BMI" = "BMI",

"Alcohol Consumption" = "AlcoholConsumption",

"Smoking Status" = "SmokingStatus",

"Physical Activity" = "PhysicalActivity",

name

)

}

# Helper function to clean up level names

clean\_level\_name <- function(name) {

name %>%

str\_remove("^BMI") %>%

str\_remove("^AlcoholConsumption") %>%

str\_remove("^SmokingStatus") %>%

str\_remove("^PhysicalActivity") %>%

str\_trim()

}

# Modify the plot\_data creation

plot\_data <- model\_summary %>%

filter(term != "(Intercept)") %>%

mutate(

Variable = case\_when(

grepl("BMI", term) ~ "BMI",

grepl("AlcoholConsumption", term) ~ "Alcohol Consumption",

grepl("SmokingStatus", term) ~ "Smoking Status",

grepl("PhysicalActivity", term) ~ "Physical Activity",

TRUE ~ NA\_character\_

),

Level = str\_remove(term, paste0(Variable, "")) %>% str\_trim()

) %>%

bind\_rows(

tibble(

term = c("BMINot overweight or obese", "AlcoholConsumptionNone in last year/nondrinker",

"SmokingStatusNever smoked", "PhysicalActivityMeets MVPA Guidelines"),

estimate = 1,

conf.low = 1,

conf.high = 1,

p.value = NA,

Variable = c("BMI", "Alcohol Consumption", "Smoking Status", "Physical Activity"),

Level = c("Not overweight or obese", "None in last year/nondrinker", "Never smoked", "Meets MVPA Guidelines")

)

) %>%

mutate(

N = sapply(seq\_along(term), function(i) {

var\_name <- map\_variable\_name(Variable[i])

level\_name <- clean\_level\_name(Level[i])

if (!is.null(n\_counts[[var\_name]])) {

matched\_level <- names(n\_counts[[var\_name]])[str\_detect(names(n\_counts[[var\_name]]), fixed(level\_name, ignore\_case = TRUE))]

if (length(matched\_level) > 0) {

return(n\_counts[[var\_name]][matched\_level[1]])

}

}

return(NA\_integer\_)

}),

Odds\_ratio = ifelse(estimate == 1, "Reference", sprintf("%.2f (%.2f, %.2f)", estimate, conf.low, conf.high)),

p\_value = ifelse(is.na(p.value), "Reference",

ifelse(p.value < 0.001, "<0.001", sprintf("%.3f", p.value)))

) %>%

group\_by(Variable) %>%

arrange(desc(Odds\_ratio == "Reference"), Level, .by\_group = TRUE) %>%

ungroup() %>%

mutate(row\_id = row\_number())

# Clean up the Level names

plot\_data$Level <- str\_remove(plot\_data$Level, "^BMI")

plot\_data$Level <- str\_remove(plot\_data$Level, "^AlcoholConsumption")

plot\_data$Level <- str\_remove(plot\_data$Level, "^SmokingStatus")

plot\_data$Level <- str\_remove(plot\_data$Level, "^PhysicalActivity")

plot\_data$Level <- str\_trim(plot\_data$Level)

# Remove the row with "Adjusted Odds Ratio" more aggressively

plot\_data <- plot\_data %>%

filter(!grepl("Adjusted Odds Ratio", Level, ignore.case = TRUE)) %>%

filter(!grepl("Adjusted Odds Ratio", Variable, ignore.case = TRUE))

# Update the repetitive variable names to show once per variable group

plot\_data <- plot\_data %>%

mutate(Variable\_display = ifelse(!duplicated(Variable), Variable, ""))

# Modify the forest plot creation

forest\_plot <- ggplot(plot\_data, aes(y = reorder(interaction(Variable, Level), desc(row\_id)), x = estimate)) +

geom\_point(size = 2, color = "black") +

geom\_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.2, color = "black") +

geom\_vline(xintercept = 1, linetype = "dashed", color = "black") + # This line is now correctly at x=1

scale\_x\_continuous(limits = c(0, 4),

breaks = seq(0, 4, by = 0.5),

expand = c(0, 0)) +

labs(x = NULL, y = NULL) +

theme\_minimal() +

theme(

axis.text.y = element\_blank(),

panel.grid.minor = element\_blank(),

panel.grid.major.y = element\_blank(),

axis.text.x = element\_blank(),

plot.margin = margin(5.5, 5.5, 5.5, 5.5),

text = element\_text(color = "black")

)

# Convert ggplot to grob

forest\_grob <- ggplotGrob(forest\_plot)

# Create the table structure

table\_data <- plot\_data %>%

dplyr::select(Variable\_display, Level, N, Odds\_ratio, p\_value)

# Create a gtable for the main structure

gt <- tableGrob(table\_data, rows = NULL,

theme = ttheme\_minimal(

core = list(fg\_params = list(hjust = 0, x = 0.1),

bg\_params = list(fill = "white")),

colhead = list(fg\_params = list(hjust = 0, x = 0.1, fontface = "bold"))

))

# Adjust column widths

gt$widths <- unit(c(2, 3, 1, 3, 2), "inches")

# Add forest plot column to the table

gt <- gtable\_add\_cols(gt, unit(3.5, "inches"), 3)

gt <- gtable\_add\_grob(gt, forest\_grob, t = 2, b = nrow(gt), l = 4, r = 4)

# Add title

title <- textGrob("Adjusted Odds Ratios for Fair/Bad Health in behavioural factor",

gp = gpar(fontface = "bold", fontsize = 14, col = "black"))

gt <- gtable\_add\_rows(gt, heights = grobHeight(title) + unit(0.5, "line"), pos = 0)

gt <- gtable\_add\_grob(gt, title, t = 1, l = 1, r = ncol(gt))

# Add lines between all rows and columns

gt <- gtable::gtable\_add\_grob(gt,

grobs = rectGrob(gp = gpar(fill = NA, lwd = 0.5, col = "black")),

t = 1, b = nrow(gt), l = 1, r = ncol(gt))

for (i in 1:(nrow(gt) - 1)) {

gt <- gtable::gtable\_add\_grob(gt,

grobs = segmentsGrob(x0 = unit(0, "npc"), x1 = unit(1, "npc"),

y0 = unit(0, "npc"), y1 = unit(0, "npc"),

gp = gpar(lwd = 0.5, col = "black")),

t = i, l = 1, r = ncol(gt))

}

for (i in 1:(ncol(gt))) {

gt <- gtable::gtable\_add\_grob(gt,

grobs = segmentsGrob(x0 = unit(0, "npc"), x1 = unit(0, "npc"),

y0 = unit(0, "npc"), y1 = unit(1, "npc"),

gp = gpar(lwd = 0.5, col = "black")),

t = 1, b = nrow(gt), l = i)

}

# Update the scale\_grob creation with adjusted positioning

scale\_grob <- textGrob(paste(c("0", "0.5", "1", "1.5", "2", "2.5", "3", "3.5", "4"), collapse = " "),

gp = gpar(fontsize = 8, col = "black"),

x = unit(0.11, "npc"), just = "left") # Adjusted x position from 0.02 to 0.1

# Add x-axis scale below the plot column with adjusted positioning

gt <- gtable\_add\_rows(gt, heights = unit(1, "cm"))

gt <- gtable\_add\_grob(gt, scale\_grob, t = nrow(gt), l = 4, r = 4, clip = "off")

# Adjust the "Adjusted Odds Ratio" label position

axis\_label <- textGrob("Adjusted Odds Ratio", gp = gpar(fontsize = 10, col = "black"),

x = unit(0.1, "npc"), just = "left") # Adjusted x position from 0.02 to 0.1

gt <- gtable\_add\_rows(gt, heights = unit(1, "cm"))

gt <- gtable\_add\_grob(gt, axis\_label, t = nrow(gt), l = 4, r = 4, clip = "off")

# Display the final plot

grid.newpage()

grid.draw(gt)

# Save the plot

ggsave("updated\_health\_factors\_forest\_plot\_table.png", gt, width = 15, height = 10, dpi = 300)