# 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\_

)),

Age = factor(case\_when(

DvAgeGrp7 == 1 ~ "16-24",

DvAgeGrp7 == 2 ~ "25-34",

DvAgeGrp7 == 3 ~ "35-44",

DvAgeGrp7 == 4 ~ "45-54",

DvAgeGrp7 == 5 ~ "55-64",

DvAgeGrp7 %in% c(6, 7) ~ "65+",

TRUE ~ NA\_character\_

), levels = c("16-24", "25-34", "35-44", "45-54", "55-64", "65+")),

Sex = factor(case\_when(

Sex == 1 ~ "Male",

Sex == 2 ~ "Female",

TRUE ~ NA\_character\_

), levels = c("Male", "Female")),

MaritalStatus = factor(case\_when(

MarStat %in% c(2, 3, 4, 7, 9) ~ "Married/civil partnership/cohabiting",

MarStat == 1 ~ "Single",

MarStat %in% c(5, 6, 8) ~ "Divorced/widowed/separated",

TRUE ~ NA\_character\_

), levels = c("Married/civil partnership/cohabiting", "Single", "Divorced/widowed/separated")),

Deprivation = factor(case\_when(

DvWIMDOvr5 == 5 ~ "Least deprived - 0.48 ->8.37",

DvWIMDOvr5 == 4 ~ "8.37 -> 13.92",

DvWIMDOvr5 == 3 ~ "13.92 ->21.43",

DvWIMDOvr5 == 2 ~ "21.43 -> 33.88",

DvWIMDOvr5 == 1 ~ "Most deprived - 33.88 -> 92.60",

TRUE ~ NA\_character\_

), levels = c("Least deprived - 0.48 ->8.37", "8.37 -> 13.92", "13.92 ->21.43", "21.43 -> 33.88", "Most deprived - 33.88 -> 92.60")),

Income = factor(case\_when(

DvWIMDInc5 == 5 ~ "Highest quintile",

DvWIMDInc5 == 4 ~ "2nd quintile",

DvWIMDInc5 == 3 ~ "3rd quintile",

DvWIMDInc5 == 2 ~ "4th quintile",

DvWIMDInc5 == 1 ~ "Lowest quintile",

TRUE ~ "unknown"

), levels = c("Highest quintile", "2nd quintile", "3rd quintile", "4th quintile", "Lowest quintile", "unknown")),

Education = factor(case\_when(

Educat2 == 1 ~ "Degree or equivalent",

Educat2 %in% c(2, 3, 4, 5, 6, 7) ~ "Below degree",

Educat2 == 8 ~ "None",

TRUE ~ NA\_character\_

), levels = c("Degree or equivalent", "Below degree", "None"))

)

# ... [rest of the code remains the same until the arrange() function]

# ... [rest of the code remains the same]

# Hardcode N values

n\_counts <- list(

Age = c("16-24" = 917, "25-34" = 2567, "35-44" = 3005, "45-54" = 3331, "55-64" = 4774, "65+" = 9126),

Sex = c("Male" = 10204, "Female" = 13499),

MaritalStatus = c("Married/civil partnership/cohabiting" = 11339, "Single" = 6435, "Divorced/widowed/separated" = 5940),

Deprivation = c("Least deprived - 0.48 ->8.37" = 4927, "8.37 -> 13.92" = 5498, "13.92 ->21.43" = 5088, "21.43 -> 33.88" = 4318, "Most deprived - 33.88 -> 92.60" = 3889),

Income = c("Highest quintile" = 5163, "2nd quintile" = 5529, "3rd quintile" = 4942, "4th quintile" = 4223, "Lowest quintile" = 3863),

Education = c("Degree or equivalent" = 7371, "Below degree" = 12779, "None" = 3474)

)

# Fit the GLM model

GLM\_Object <- glm(HealthStatus == "Fair/bad health" ~ Age + Sex + MaritalStatus + Deprivation + Income + Education,

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

# Get model summary

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

# Create a function to get the correct order for each variable

get\_variable\_order <- function(variable) {

switch(variable,

"Age" = c("16-24", "25-34", "35-44", "45-54", "55-64", "65+"),

"Sex" = c("Male", "Female"),

"MaritalStatus" = c("Married/civil partnership/cohabiting", "Single", "Divorced/widowed/separated"),

"Deprivation" = c("Least deprived - 0.48 ->8.37", "8.37 -> 13.92", "13.92 ->21.43", "21.43 -> 33.88", "Most deprived - 33.88 -> 92.60"),

"Income" = c("Highest quintile", "2nd quintile", "3rd quintile", "4th quintile", "Lowest quintile", "unknown"),

"Education" = c("Degree or equivalent", "Below degree", "None")

)

}

# ... [previous data loading and recoding code remains unchanged]

# Special handling for MaritalStatus N values

special\_marital\_status\_handling <- function(levels) {

sapply(levels, function(level\_name) {

if (grepl("Single", level\_name, ignore.case = TRUE)) {

return(n\_counts$MaritalStatus["Single"])

} else if (grepl("Divorced|widowed|separated", level\_name, ignore.case = TRUE)) {

return(n\_counts$MaritalStatus["Divorced/widowed/separated"])

} else if (grepl("Married|civil partnership|cohabiting", level\_name, ignore.case = TRUE)) {

return(n\_counts$MaritalStatus["Married/civil partnership/cohabiting"])

} else {

return(NA)

}

})

}

# Correct hardcoding of N values, including Marital Status

n\_counts <- list(

Age = c("16-24" = 917, "25-34" = 2567, "35-44" = 3005, "45-54" = 3331, "55-64" = 4774, "65+" = 9126),

Sex = c("Male" = 10204, "Female" = 13499),

MaritalStatus = c("Married/civil partnership/cohabiting" = 11339, "Single" = 6435, "Divorced/widowed/separated" = 5940),

Deprivation = c("Least deprived - 0.48 ->8.37" = 4927, "8.37 -> 13.92" = 5498, "13.92 ->21.43" = 5088, "21.43 -> 33.88" = 4318, "Most deprived - 33.88 -> 92.60" = 3889),

Income = c("Highest quintile" = 5163, "2nd quintile" = 5529, "3rd quintile" = 4942, "4th quintile" = 4223, "Lowest quintile" = 3863),

Education = c("Degree or equivalent" = 7371, "Below degree" = 12779, "None" = 3474)

)

# Prepare the data for plotting with updated N value for "Single"

plot\_data <- model\_summary %>%

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

mutate(

Variable = case\_when(

grepl("Age", term) ~ "Age",

grepl("Sex", term) ~ "Sex",

grepl("MaritalStatus", term) ~ "Marital Status",

grepl("Deprivation", term) ~ "Deprivation",

grepl("Income", term) ~ "Income",

grepl("Education", term) ~ "Education",

TRUE ~ NA\_character\_

),

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

) %>%

bind\_rows(

tibble(

term = c("Age16-24", "SexMale", "MaritalStatusMarried/civil partnership/cohabiting",

"DeprivationLeast deprived - 0.48 ->8.37", "IncomeHighest quintile", "EducationDegree or equivalent"),

estimate = 1,

conf.low = 1,

conf.high = 1,

p.value = NA,

Variable = c("Age", "Sex", "Marital Status", "Deprivation", "Income", "Education"),

Level = c("16-24", "Male", "Married/civil partnership/cohabiting",

"Least deprived - 0.48 ->8.37", "Highest quintile", "Degree or equivalent")

)

) %>%

mutate(

N = case\_when(

Variable == "Marital Status" ~ special\_marital\_status\_handling(Level),

Variable == "Age" ~ n\_counts$Age[Level],

Variable == "Sex" ~ n\_counts$Sex[Level],

Variable == "Deprivation" ~ n\_counts$Deprivation[Level],

Variable == "Income" ~ n\_counts$Income[Level],

Variable == "Education" ~ n\_counts$Education[Level],

TRUE ~ 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))),

Variable\_order = factor(Variable, levels = c("Age", "Sex", "Marital Status", "Deprivation", "Income", "Education")),

Level\_order = factor(Level, levels = unlist(lapply(unique(Variable), get\_variable\_order)))

) %>%

arrange(Variable\_order, Level\_order) %>%

mutate(

row\_id = row\_number(),

Variable\_display = ifelse(!duplicated(Variable), as.character(Variable), "")

)

# Create the forest plot without resizing the table rows

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

geom\_point(size = 2) +

geom\_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.2) + # Keep error bar heights smaller

geom\_vline(xintercept = 1, linetype = "dashed") +

scale\_x\_continuous(limits = c(0, max(plot\_data$conf.high, na.rm = TRUE) \* 1.1),

breaks = seq(0, ceiling(max(plot\_data$conf.high, na.rm = TRUE)), by = 0.5)) +

labs(x = "", y = "") +

theme\_minimal() +

theme(

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

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

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

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

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

plot.margin = margin(5.5, 20, 5.5, 5.5)

)

# Convert ggplot to grob

forest\_grob <- ggplotGrob(forest\_plot)

# Create the table structure without adjusting row heights manually

table\_data <- plot\_data %>%

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

# Create the gtable for the 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"),

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

))

# Add vertical and horizontal lines

for(i in seq\_len(ncol(gt))) {

gt <- 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 = 1.5)),

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

}

for(i in seq\_len(nrow(gt))) {

gt <- 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 = 1.5)),

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

}

# Adjust column widths and add plot column

gt$widths <- unit(c(2, 3, 1, 4, 2, 1), "inches")

gt <- gtable\_add\_cols(gt, unit(4, "inches"), 3)

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

# Adjust the x-axis scale positioning

scale\_grob <- textGrob(paste(seq(0, 3.5, by = 0.5), collapse = " "),

gp = gpar(fontsize = 8),

x = unit(0.18, "npc"), # Move the scale text to the left

just = "left") # Align the text to the left

# Add x-axis scale below the plot

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")

# Add "Adjusted Odds Ratio" label below the x-axis scale

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

gt <- gtable\_add\_rows(gt, heights = unit(0.1, "cm"))

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

# Add title

title <- textGrob("Adjusted Odds Ratios for Fair/Bad Health in Socio-demographic factors",

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

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))

# Display the final plot

grid.newpage()

grid.draw(gt)

# Save the plot

ggsave("correct\_forest\_plot\_ordered\_with\_scales.png", gt, width = 18, height = 10, dpi = 300)