Weld_Model

Data Pre-Processing

```
## ============================================================
## Data import + cleaning + descriptives + Table 1 export
## ============================================================

## ---- Libraries (keep only what you use here) ----
library(readxl)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(tidyr)
library(forcats)
library(tableone)
library(tibble)
library(knitr)
library(kableExtra)
```

```
Attaching package: 'kableExtra'

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

    group_rows
```

```
library(webshot2)
library(car)
```

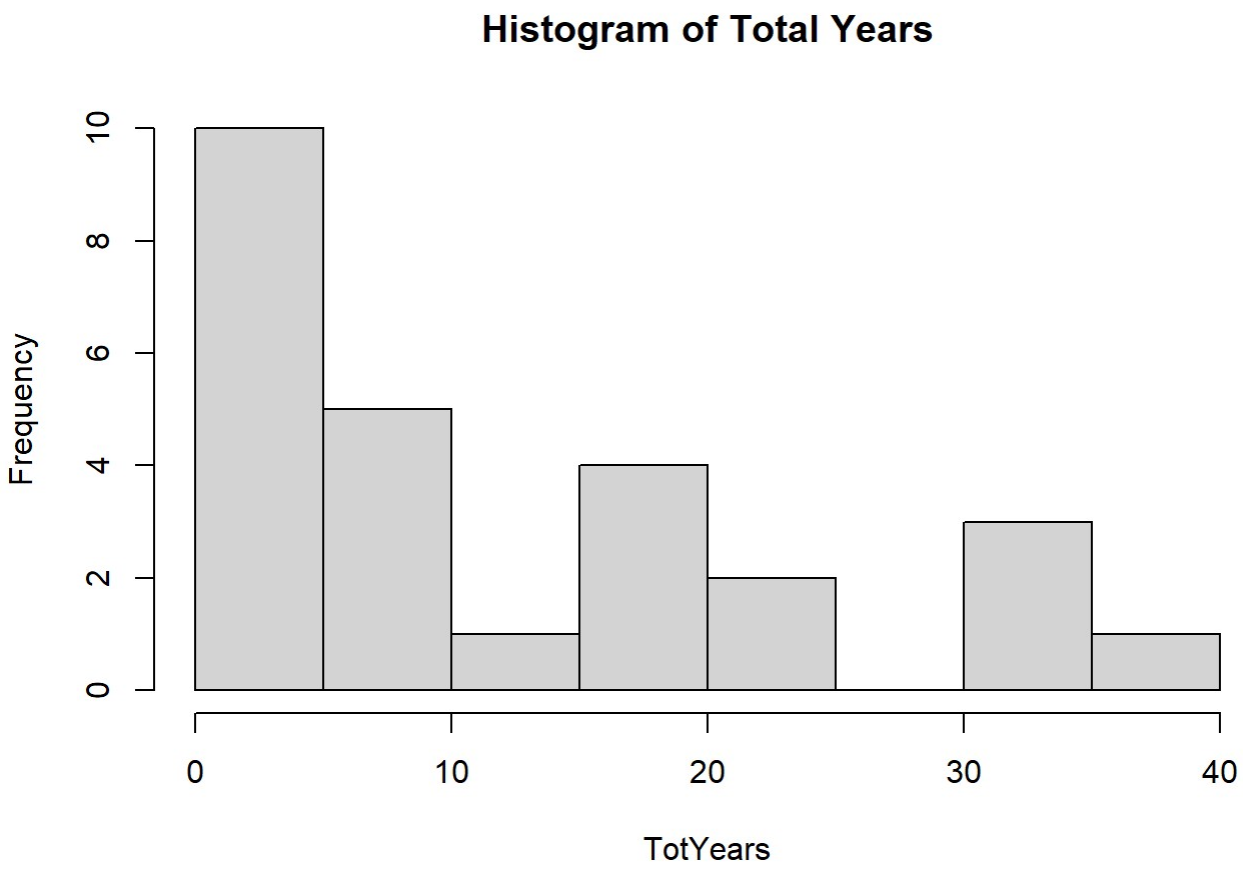
```
Loading required package: carData


Attaching package: 'car'

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

recode

```
library(corrr)
## ---- Read data ----
wdat <- read_excel("C:/Users/vijay/Box/Manuscript in Prep/Welders/wdat1.xlsx")

## ---- Convert selected variables to factors ----
wdat <- wdat %>%
  mutate(
    FallHist    = as.factor(FallHist),
    NoiseExp    = as.factor(NoiseExp),
    VibExp      = as.factor(VibExp),
    FumesExp    = as.factor(FumesExp),
    PPEHearing  = as.factor(PPEHearing),
    PPEResp     = as.factor(PPEResp),
    PPEEye      = as.factor(PPEEye),
    Smoking     = as.factor(Smoking),
    Alcohol     = as.factor(Alcohol),
    Group       = as.factor(Group),
    Marital     = as.factor(Marital),
    EduLevel    = as.factor(EduLevel),
    WeldingTyp  = as.factor(WeldingTyp),
    ExFrequency = as.factor(ExFrequency),
    Resid       = as.factor(Resid),
    LivCond     = as.factor(LivCond),
    CoMorb      = as.factor(CoMorb)
  )

## ---- Drop Gender if present (optional) ----
if ("Gender" %in% names(wdat)) {
  wdat <- wdat %>% select(-Gender)
}

## ---- FIX Group labels robustly ----
## If your file uses "Welder" (singular), convert to "Welders"
wdat <- wdat %>%
  mutate(
    Group = fct_recode(Group, "Welders" = "Welder"),
    Group = factor(Group, levels = c("Control", "Welders"))
  )
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `Group = fct_recode(Group, Welders = "Welder")`.
Caused by warning:
! Unknown levels in `f`: Welder
```

```
## ---- Quick descriptives (optional sanity checks) ----
hist(wdat$TotYears, main = "Histogram of Total Years", xlab = "TotYears")
```




```
gtab <- table(wdat$Group, useNA = "ifany")
gtab
```

```
Control Welders
     10      16
```

```
prop.table(gtab) * 100
```

```
 Control  Welders
38.46154 61.53846
```

```
cont_table <- xtabs(~ Group + FallHist, data = wdat)
cont_table
```

```
         FallHist
Group     0 1
  Control 7 3
  Welders 9 7
```

```
prop.table(cont_table, margin = 1) * 100  # row %
```

```
         FallHist
Group         0     1
  Control 70.00 30.00
  Welders 56.25 43.75
```

```
prop.table(cont_table, margin = 2) * 100  # col %
```

```
         FallHist
Group         0     1
  Control 43.75 30.00
  Welders 56.25 70.00
```

```
## ============================================================
## Recode exposure variables (as you requested)
## Const -> Constant
## Freq  -> Frequent
## Occ   -> Occasional
## Cont  -> Control   (NOTE: your dataset uses Cont = Control)
## None  -> None
## ============================================================
exp_vars <- c("NoiseExp", "VibExp", "FumesExp")

wdat <- wdat %>%
  mutate(across(all_of(exp_vars), ~ fct_recode(.x,
    "Constant"   = "Const",
    "Frequent"   = "Freq",
    "Occasional" = "Occ",
    "Control"    = "Cont",
    "None"       = "None"
  )))
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `across(...)`.
Caused by warning:
! Unknown levels in `f`: None
```

```
## Optional: consistent order in tables
wdat <- wdat %>%
  mutate(across(all_of(exp_vars), ~ factor(.x,
    levels = c("None", "Occasional", "Frequent", "Constant", "Control")
  )))

## ---- Variable label map (OK as-is) ----
name_to_label <- c(
  Group = "Group",
  Age = "Age (mean + SD)"
```

```
  Age = "Age (mean ± SD)",
  Marital = "Marital Status",
  EduLevel = "Education Level",
  TotYears = "Total Years of Work (mean ± SD)",
  WeldingTyp = "Welding Type",
  Whours = "Weekly Hours",
  NoiseExp = "Noise Exposure",
  VibExp = "Vibration Exposure",
  FumesExp = "Fumes Exposure",
  Smoking = "Smoking Status",
  Alcohol = "Alcohol Consumption",
  ExFrequency = "Exposure Frequency",
  CoMorb = "Comorbidities",
  PPEHearing = "Hearing Protection",
  PPEResp = "Respirator Use",
  PPEEye = "Eye Protection",
  Resid = "Residency",
  LivCond = "Living Conditions",
  BBS = "Balance Score",
  MBEST = "mBEST Score",
  FES = "FES Score",
  FallHist = "Fall History"
)
```

Table 1

```
## ============================================================
## Table 1 (TableOne) -> HTML -> PNG
## Stratified by Group, includes p-values
## ============================================================

factor_vars <- c(
  "Marital","EduLevel","Group","WeldingTyp",
  "NoiseExp","VibExp","FumesExp","Smoking","Alcohol",
  "PPEHearing","PPEResp","PPEEye",
  "FallHist"
)

vars <- c(
  "Age","Marital","EduLevel","TotYears","WeldingTyp",
  "NoiseExp","VibExp","FumesExp","Smoking","Alcohol",
  "PPEHearing","PPEResp","PPEEye",
  "FallHist","Whours"
)

table1 <- CreateTableOne(
  vars = vars,
  data = wdat,
  factorVars = factor vars,
```

```
  strata = "Group",
  test = TRUE,
  includeNA = FALSE   # set TRUE if you want NA rows shown
)

tab_mat <- print(
  table1,
  showAllLevels = TRUE,
  quote = FALSE,
  noSpaces = TRUE,
  printToggle = FALSE
)

tab_df <- as.data.frame(tab_mat) %>%
  rownames_to_column(var = "Variable") %>%
  mutate(
    Variable = gsub("^X\\.?[0-9]*\\s*", "", Variable),
    Variable = trimws(Variable)
  ) %>%
  select(Variable, everything())

kbl_obj <- kbl(
  tab_df,
  format = "html",
  caption = "Table 1. Participant Characteristics Stratified by Group",
  escape = FALSE,
  align = "l"
) %>%
  kable_styling(
    full_width = FALSE,
    position = "center",
    bootstrap_options = c("striped", "condensed", "hover")
  ) %>%
  row_spec(0, bold = TRUE)

html_file <- "Table1_publication.html"
save_kable(kbl_obj, file = html_file, self_contained = TRUE)

png_file <- "Table1_publication.png"
webshot(
  url = html_file,
  file = png_file,
  zoom = 2,
  vwidth = 1800,
  vheight = 1200,
  delay = 0.3
)
```

```
file:///C:/Users/vijay/Box/Manuscript in Prep/Welders/Welders/Table1_publication.html
```

```
screensnot completed
```

Table 1. Participant Characteristics Stratified by Group

Variable	level	Control	Welders	p	test
n		10	16		
Age..mean..SD..		40.80 (8.90)	41.81 (6.27)	0.736	
Marital....	Married	8 (80.0)	15 (93.8)	0.398	
	Other	1 (10.0)	0 (0.0)		
	Single	1 (10.0)	1 (6.2)		
EduLevel....	Diploma	2 (20.0)	2 (12.5)	0.183	
	Graduate	2 (20.0)	1 (6.2)		
	Postgraduate	2 (20.0)	0 (0.0)		
	Primary school	3 (30.0)	8 (50.0)		
	Secondary school	1 (10.0)	5 (31.2)		
TotYears..mean..SD..		0.00 (0.00)	19.69 (10.63)	<0.001	
WeldingTyp....	Arc	0 (0.0)	7 (43.8)	<0.001	
	MIG	0 (0.0)	5 (31.2)		
	None	10 (100.0)	0 (0.0)		
	TIG	0 (0.0)	4 (25.0)		
NoiseExp....	None	0 (0.0)	1 (6.2)	<0.001	
	Occasional	0 (0.0)	5 (31.2)		
	Frequent	0 (0.0)	8 (50.0)		
	Constant	0 (0.0)	2 (12.5)		
	Control	10 (100.0)	0 (0.0)		
VibExp....	Occasional	0 (0.0)	5 (31.2)	<0.001	
	Frequent	0 (0.0)	8 (50.0)		
	Constant	0 (0.0)	3 (18.8)		
	Control	10 (100.0)	0 (0.0)		
FumesExp....	None	0 (0.0)	1 (6.2)	<0.001	
	Occasional	0 (0.0)	2 (12.5)		
	Frequent	0 (0.0)	9 (56.2)		
	Constant	0 (0.0)	4 (25.0)		
	Control	10 (100.0)	0 (0.0)		
Smoking....	Current	1 (10.0)	6 (37.5)	0.154	
	Former	4 (40.0)	2 (12.5)		
	Never	5 (50.0)	8 (50.0)		
Alcohol....	Never	4 (40.0)	6 (37.5)	0.360	
	Occ	4 (40.0)	3 (18.8)		
	Regular	2 (20.0)	7 (43.8)		
PPEHearing....	Never	10 (100.0)	11 (68.8)	0.146	
	Sometimes	0 (0.0)	5 (31.2)		
PPEResp....	Always	0 (0.0)	3 (18.8)	0.003	
	Never	10 (100.0)	5 (31.2)		
	Sometimes	0 (0.0)	8 (50.0)		
PPEEye....	Always	0 (0.0)	7 (43.8)	<0.001	
	Never	10 (100.0)	2 (12.5)		
	Sometimes	0 (0.0)	7 (43.8)		
FallHist....	0	7 (70.0)	9 (56.2)	0.774	
	1	3 (30.0)	7 (43.8)		
Whours..mean..SD..		0.00 (0.00)	8.19 (2.07)	<0.001	

```
message("Saved HTML: ", html_file)
```

```
Saved HTML: Table1_publication.html
```

```
message("Saved PNG:  ", png_file)
```

```
Saved PNG:  Table1_publication.png
```

```
csv_file <- "Table1_publication.csv"

write.csv(
  tab_df,
  file = csv_file,
```

```
  row.names = FALSE,
  na = ""
)

message("Saved CSV: ", csv_file)
```

```
Saved CSV: Table1_publication.csv
```

Table 2

```
library(dplyr)

# Outcomes to analyze
outcomes <- c("FES", "BBS", "MBEST")

# Helper functions
fmt_mean_sd <- function(x) {
  sprintf("%.2f \u00B1 %.2f", mean(x, na.rm = TRUE), sd(x, na.rm = TRUE))
}

fmt_med_iqr <- function(x) {
  qs <- quantile(x, probs = c(0.25, 0.50, 0.75), na.rm = TRUE, names = FALSE)
  sprintf("%.2f [%.2f, %.2f]", qs[2], qs[1], qs[3])
}

# Calculate rank-biserial correlation
rank_biserial_wilcox <- function(x, g, ref = "Control") {
  df <- data.frame(x = x, g = g) %>%
    filter(!is.na(x), !is.na(g))

  df$g <- droplevels(as.factor(df$g))
  if (nlevels(df$g) != 2) stop("Group must have exactly 2 levels.")
  if (!ref %in% levels(df$g)) stop("Reference group not found.")

  df$g <- relevel(df$g, ref = ref)
  g1 <- levels(df$g)[1]
  g2 <- levels(df$g)[2]

  x1 <- df$x[df$g == g1]
  x2 <- df$x[df$g == g2]
  n1 <- length(x1)
  n2 <- length(x2)
  if (n1 < 2 || n2 < 2) return(NA_real_)

  wt <- wilcox.test(x1, x2, exact = FALSE)
  W <- as.numeric(wt$statistic)
  U <- wt$statistic
  # Calculate rank-biserial correlation
```

```
  rb <- (2 * (n1 * n2 - U)) / (n1 * n2) - 1
  return(rb)
}

# Calculate Hedges' g (standardized mean difference)
cohen_g <- function(x, g, ref_levels = c("Control", "Welders")) {
  df <- data.frame(x = x, g = g) %>% filter(!is.na(x), !is.na(g))
  df$g <- factor(df$g, levels = ref_levels)
  x1 <- df$x[df$g == ref_levels[1]]
  x2 <- df$x[df$g == ref_levels[2]]
  n1 <- length(x1)
  n2 <- length(x2)
  if (n1 < 2 || n2 < 2) return(NA_real_)
  m1 <- mean(x1)
  m2 <- mean(x2)
  s1 <- sd(x1)
  s2 <- sd(x2)
  sp <- sqrt(((n1 - 1)*s1^2 + (n2 - 1)*s2^2) / (n1 + n2 - 2))
  g <- (m2 - m1) / sp
  # Hedges' correction for small samples
  J <- 1 - (3 / (4*(n1 + n2) - 9))
  g_corrected <- g * J
  return(g_corrected)
}

# Perform analysis for each outcome
results <- lapply(outcomes, function(v) {
  df <- wdat %>%
    select(Group, all_of(v)) %>%
    rename(Value = all_of(v)) %>%
    filter(!is.na(Value))

  x_control <- df %>% filter(Group == "Control") %>% pull(Value)
  x_welders <- df %>% filter(Group == "Welders") %>% pull(Value)

  # Wilcoxon test
  wilcox_test <- wilcox.test(Value ~ Group, data = df, exact = FALSE)
  p_wilcox <- signif(wilcox_test$p.value, 3)
  W_stat <- as.numeric(wilcox_test$statistic)

  # Effect size: rank-biserial correlation
  rb <- rank_biserial_wilcox(df$Value, df$Group, ref = "Control")

  # Welch's t-test
  t_test <- t.test(Value ~ Group, data = df)
  p_ttest <- signif(t_test$p.value, 3)
  t_stat <- as.numeric(t_test$statistic)

  # Effect size: Hedges' g
  g <- cohen_g(df$Value, df$Group)
```

```
  list(
    Outcome = v,
    Wilcoxon_p = p_wilcox,
    Rank_biserial_r = rb,
    Welch_t = t_stat,
    Welch_p = p_ttest,
    Hedges_g = g
  )
})

# Compile results into a data frame
results_df <- do.call(rbind, lapply(results, as.data.frame))

# Adjust p-values for multiple comparisons
results_df$Wilcoxon_p_adj <- p.adjust(results_df$Wilcoxon_p, method = "bonferroni")
results_df$Welch_p_adj <- p.adjust(results_df$Welch_p, method = "bonferroni")

# Display the results
print(results_df)
```

```
   Outcome Wilcoxon_p Rank_biserial_r   Welch_t Welch_p   Hedges_g
W      FES    0.00467         0.66875 -1.993057  0.0577  0.6974523
W1     BBS    0.02130        -0.54375  2.385066  0.0274 -0.7641659
W2   MBEST    0.04620        -0.47500  2.253067  0.0338 -0.7718341
   Wilcoxon_p_adj Welch_p_adj
W         0.01401      0.1731
W1        0.06390      0.0822
W2        0.13860      0.1014
```

```
library(kableExtra)
library(webshot)
```

```
Registered S3 method overwritten by 'webshot':
  method        from
  print.webshot webshot2


Attaching package: 'webshot'

The following objects are masked from 'package:webshot2':

    appshot, resize, rmdshot, shrink, webshot
```

```
# Prepare your results table as a data frame, e.g., 'tab_df'
# For example:
# tab_df <- results_df %>%
#   select(Outcome, Wilcoxon_p, Hedges_g) %>%
#   rename(`Outcome` = Outcome, `Wilcoxon p` = Wilcoxon_p, `Hedges' g` = Hedges_g)
```

```r
# Create HTML table with kableExtra
kbl_obj1 <- kbl(
  results_df,
  format = "html",
  caption = "Table 2.Effect of Welding on BBS, FES and MBEST",
  escape = FALSE,
  align = "l"
) %>%
  kable_styling(
    full_width = FALSE,
    position = "center",
    bootstrap_options = c("striped", "condensed", "hover")
  ) %>%
  row_spec(0, bold = TRUE)

# Save the HTML file
html_file1 <- "Table2_publication.html"
save_kable(kbl_obj1, file = html_file, self_contained = TRUE)

# Save as PNG using webshot
png_file1 <- "Table2_publication.png"
webshot(
  url = html_file1,
  file = png_file1,
  zoom = 2,
  vwidth = 1800,
  vheight = 1200,
  delay = 0.3
)
```

Table 2.Effect of Welding on BBS, FES and MBEST

	Outcome	Wilcoxon_p	Rank_biserial_r	Welch_t	Welch_p	Hedges_g	Wilcoxon_p_adj	Welch_p_adj
W	FES	0.00467	0.66875	-1.993057	0.0577	0.6974523	0.01401	0.1731
W1	BBS	0.02130	-0.54375	2.385066	0.0274	-0.7641659	0.06390	0.0822
W2	MBEST	0.04620	-0.47500	2.253067	0.0338	-0.7718341	0.13860	0.1014

```
# Message outputs
message("Saved HTML: ", html_file1)
```

```
Saved HTML: Table2_publication.html
```

```
message("Saved PNG:  ", png_file1)
```

```
Saved PNG:  Table2_publication.png
```

```
csv_file <- "Table2_publication.csv"

write.csv(
  results_df,
  file = csv_file,
  row.names = FALSE,
  na = ""
)

message("Saved CSV: ", csv_file)
```

```
Saved CSV: Table2_publication.csv
```

Effect of Balance and postural stability in Welders

```
library(dplyr)
library(tidyr)
library(ggplot2)

# ---- 0) Ensure Group matches your actual levels and order ----
wdat <- wdat %>%
  mutate(
    Group = factor(Group, levels = c("Control", "Welders"))
  )

# Optional quick checks
print(unique(wdat$Group))
```

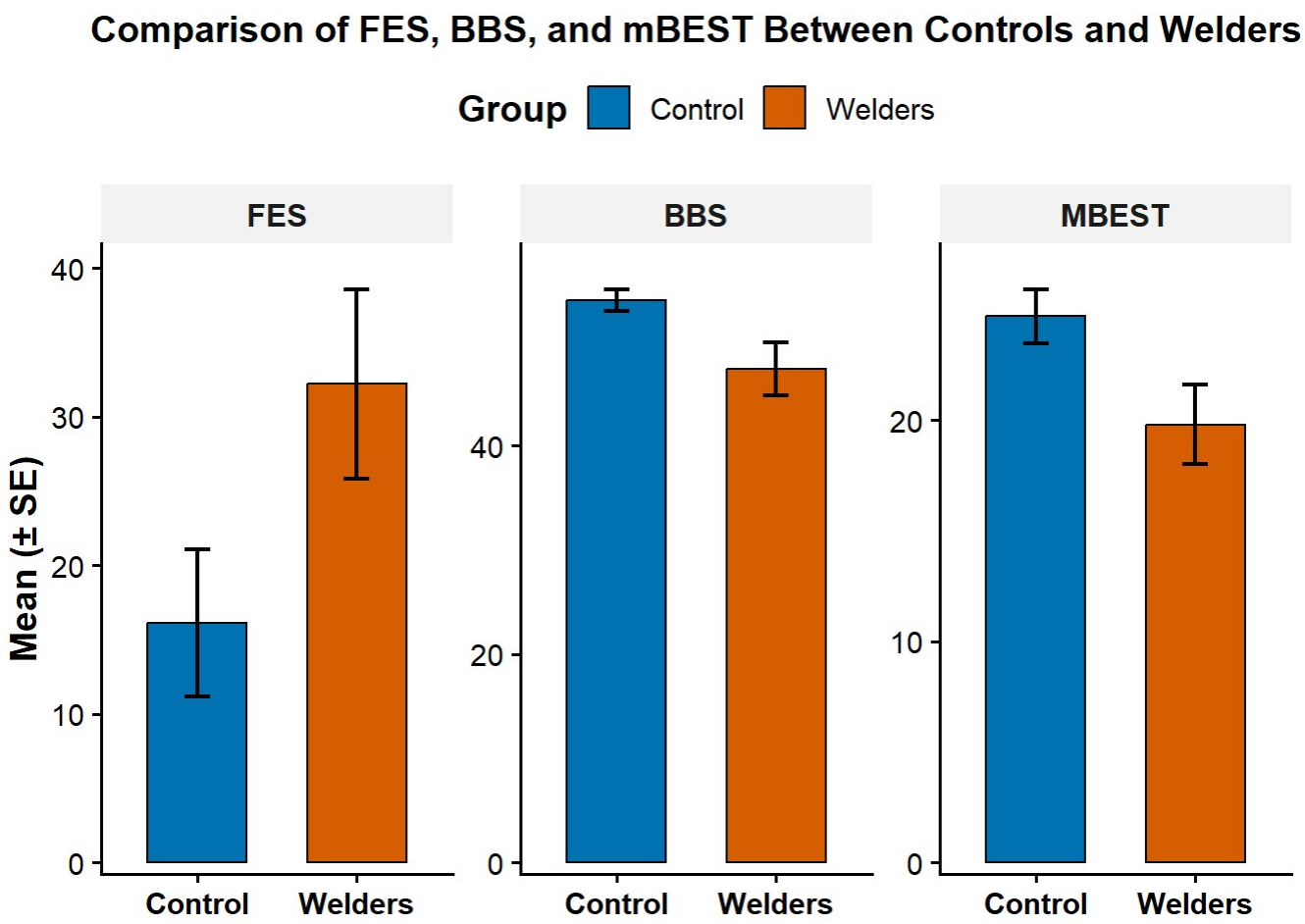
```
[1] Welders Control
Levels: Control Welders
```

```
print(table(wdat$Group, useNA = "ifany"))
```

```
Control Welders
     10      16
```

```
# ---- 1) Reshape to long format ----
combined_data <- wdat %>%
  select(Group, FES, BBS, MBEST) %>%
  pivot_longer(
    cols = c(FES, BBS, MBEST),
    names_to = "Variable",
    values_to = "Score"
  ) %>%
  mutate(
    Variable = factor(Variable, levels = c("FES", "BBS", "MBEST"))
  )

# ---- 2) Compute Mean and SE ----
plot_data <- combined_data %>%
  group_by(Group, Variable) %>%
  summarise(
    n = n(),
    Mean = mean(Score),
    SE   = sd(Score) / sqrt(n),
    .groups = "drop"
  )

# Optional: add n to facet labels (publishable)
plot_data <- plot_data %>%
  mutate(Variable_lab = paste0(Variable, " (n=", n, ")"))

# ---- 3) Publishable faceted bar plot ----
# Color-blind friendly palette (Okabe-Ito inspired)
group_cols <- c("Control" = "#0072B2", "Welders" = "#D55E00")

p <- ggplot(plot_data, aes(x = Group, y = Mean, fill = Group)) +
  geom_col(width = 0.62, color = "black", linewidth = 0.45) +
  geom_errorbar(
    aes(ymin = Mean - SE, ymax = Mean + SE),
    width = 0.16, linewidth = 0.75
  ) +
  facet_wrap(~ Variable, scales = "free_y") +
  scale_fill_manual(
    values = group_cols,
    breaks = c("Control", "Welders"),
    labels = c("Control", "Welders")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0.02, 0.08))) +
```

```
  labs(
    title = "Comparison of FES, BBS, and mBEST Between Controls and Welders",
    x = NULL,
    y = "Mean (± SE)",
    fill = "Group"
  ) +
  theme_classic(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
    axis.title.y = element_text(face = "bold"),
    axis.text.x = element_text(face = "bold"),
    legend.position = "top",
    legend.title = element_text(face = "bold"),
    strip.text = element_text(face = "bold", size = 12),
    strip.background = element_rect(fill = "grey95", color = NA),
    axis.line = element_line(linewidth = 0.6),
    axis.ticks = element_line(linewidth = 0.6)
  )

print(p)
```




```
# ---- 4) Save (journal-friendly) ----
ggsave("balancetest.png", plot = p, width = 8, height = 5.5, dpi = 600, bg = "white")
```

```
ggsave("balance_biplot.png", plot = p, width = 8, height = 5.5, dpi = 300, bg = "white")
```

PCA of Balance, Exposure and PPE Variables

```
# Load required libraries
library(dplyr)
library(fastDummies)
library(FactoMineR)
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
## ============================================================
## 4) PCA #1: Balance Metrics
## ============================================================
bal_df <- wdat %>%
  select(BBS, FES, MBEST) %>%
  mutate(across(everything(), as.numeric)) %>%
  drop_na()

pc_bal <- prcomp(bal_df, center = TRUE, scale. = TRUE)

## Add PC1 back to full dataset (align by row indices that survived drop_na)
wdat$Balance_PC1 <- NA_real_
wdat$Balance_PC1[as.integer(rownames(bal_df))] <- pc_bal$x[, 1]

cat("\n===== Balance PCA summary =====\n")
```

```
===== Balance PCA summary =====
```

```
print(summary(pc_bal))
```

```
Importance of components:
                          PC1    PC2     PC3
Standard deviation     1.5232 0.6938 0.44551
Proportion of Variance 0.7734 0.1605 0.06616
Cumulative Proportion  0.7734 0.9338 1.00000
```

```
## ============================================================
## 5) PCA #2: Exposure variables (dummy-coded) + fix zero-variance columns
## ============================================================
exp_df <- wdat %>%
  select(NoiseExp, VibExp, FumesExp) %>%
  drop_na()

exp_dummy <- exp_df %>%
```

```
  fastDummies::dummy_cols(remove_selected_columns = TRUE)

## Remove zero-variance columns (fixes: cannot rescale a constant/zero column)
zero_var <- sapply(exp_dummy, function(x) sd(x, na.rm = TRUE) == 0)
exp_dummy <- exp_dummy[, !zero_var, drop = FALSE]

pc_exp <- prcomp(exp_dummy, center = TRUE, scale. = TRUE)

wdat$Exp_PC1 <- NA_real_
wdat$Exp_PC1[as.integer(rownames(exp_df))] <- pc_exp$x[, 1]

cat("\n===== Exposure PCA summary =====\n")
```

```
===== Exposure PCA summary =====
```

```
print(summary(pc_exp))
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     2.1419 1.8386 1.5394 1.4275 0.96304 0.70584 0.36134
Proportion of Variance 0.3277 0.2415 0.1693 0.1456 0.06625 0.03559 0.00933
Cumulative Proportion  0.3277 0.5692 0.7384 0.8840 0.95023 0.98582 0.99515
                           PC8       PC9      PC10      PC11      PC12
Standard deviation     0.26065 4.965e-16 3.573e-16 2.527e-16 6.729e-17
Proportion of Variance 0.00485 0.000e+00 0.000e+00 0.000e+00 0.000e+00
Cumulative Proportion  1.00000 1.000e+00 1.000e+00 1.000e+00 1.000e+00
                           PC13      PC14
Standard deviation     8.409e-19 1.127e-46
Proportion of Variance 0.000e+00 0.000e+00
Cumulative Proportion  1.000e+00 1.000e+00
```

```
## ============================================================
## 6) PCA #3: PPE variables (dummy-coded) + fix zero-variance columns
## ============================================================
ppe_df <- wdat %>%
  select(PPEHearing, PPEResp, PPEEye) %>%
  drop_na()

ppe_dummy <- ppe_df %>%
  fastDummies::dummy_cols(remove_selected_columns = TRUE)

zero_var2 <- sapply(ppe_dummy, function(x) sd(x, na.rm = TRUE) == 0)
ppe_dummy <- ppe_dummy[, !zero_var2, drop = FALSE]

pc_ppe <- prcomp(ppe_dummy, center = TRUE, scale. = TRUE)

wdat$PPE_PC1 <- NA_real_
wdat$PPE_PC1[as.integer(rownames(ppe_df))] <- pc_ppe$x[, 1]
```

```
cat("\n===== PPE PCA summary =====\n")
```

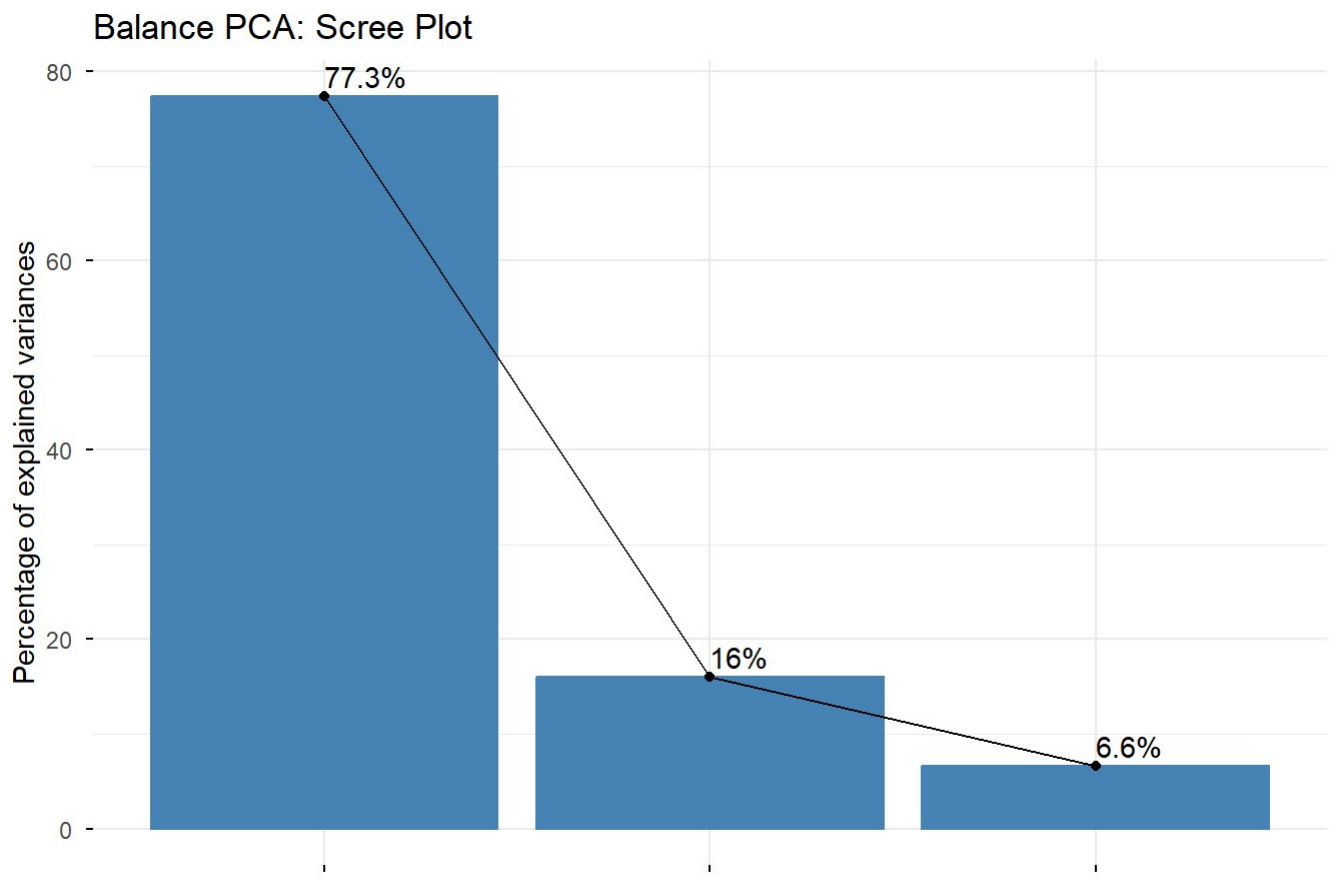
```
===== PPE PCA summary =====
```

```
print(summary(pc_ppe))
```

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5      PC6       PC7
Standard deviation     1.9053 1.6694 1.0248 0.56933 0.45643 9.19e-16 1.648e-16
Proportion of Variance 0.4538 0.3484 0.1313 0.04052 0.02604 0.00e+00 0.000e+00
Cumulative Proportion  0.4538 0.8022 0.9334 0.97396 1.00000 1.00e+00 1.000e+00
                           PC8
Standard deviation     7.551e-17
Proportion of Variance 0.000e+00
Cumulative Proportion  1.000e+00
```

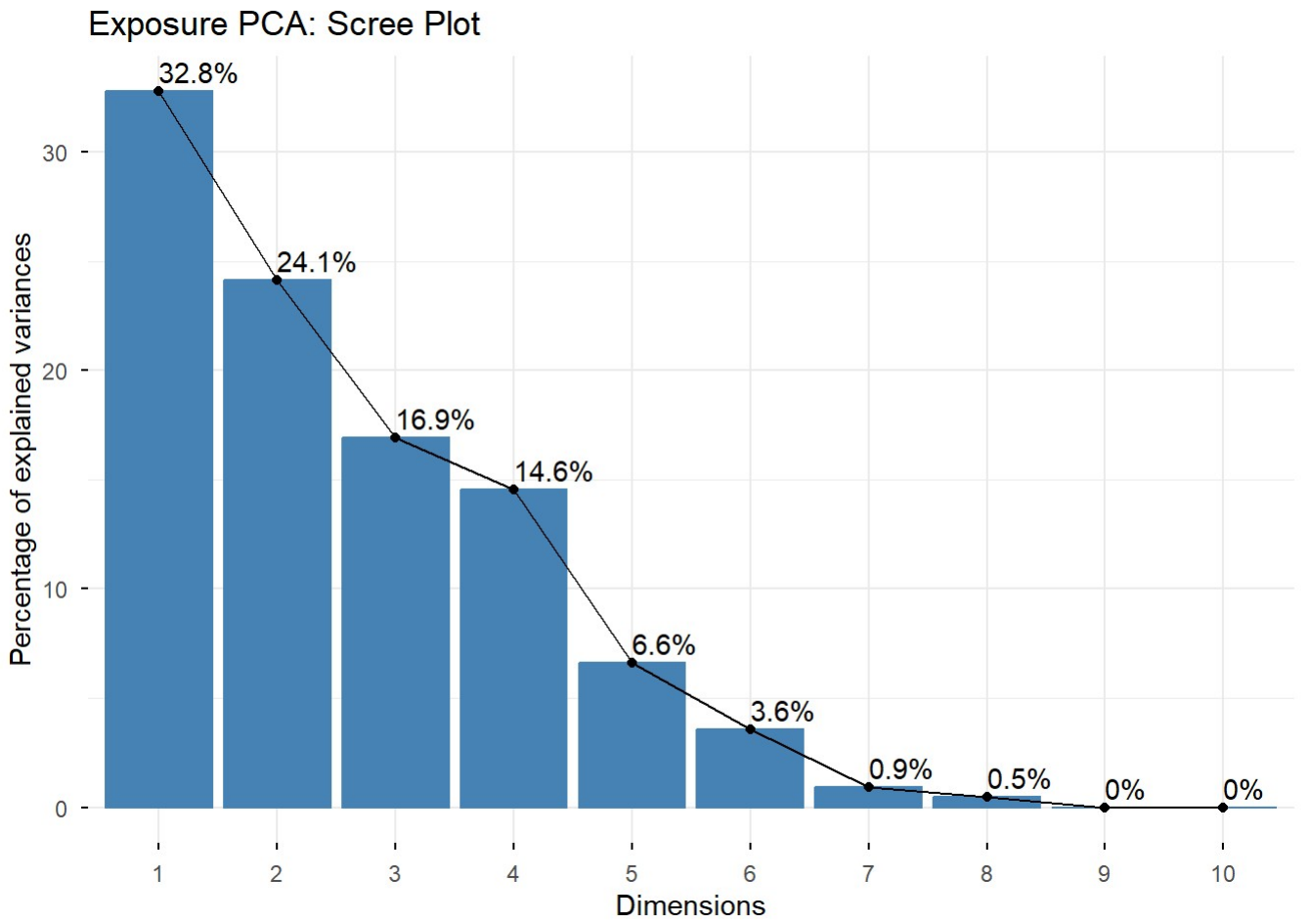
```
## ============================================================
## 7) Scree plots (variance explained)
## ============================================================
fviz_eig(pc_bal, addlabels = TRUE, main = "Balance PCA: Scree Plot")
```

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```



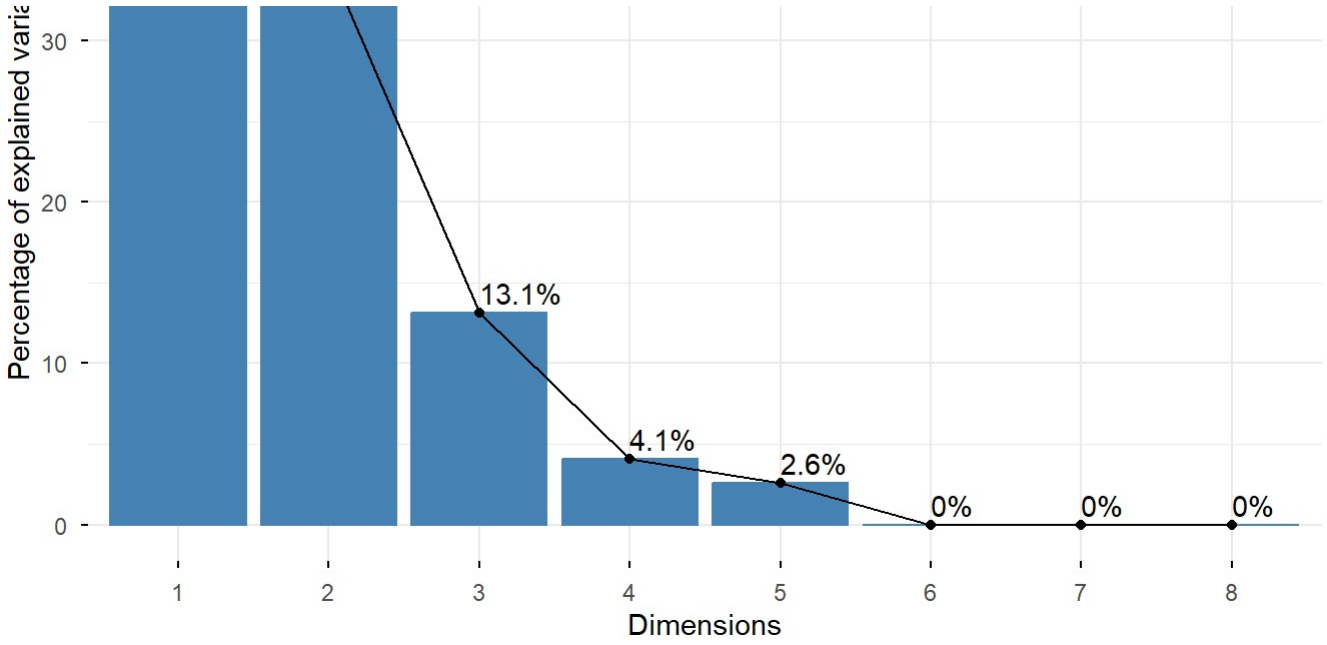

```
fviz_eig(pc_exp, addlabels = TRUE, main = "Exposure PCA: Scree Plot")
```

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```



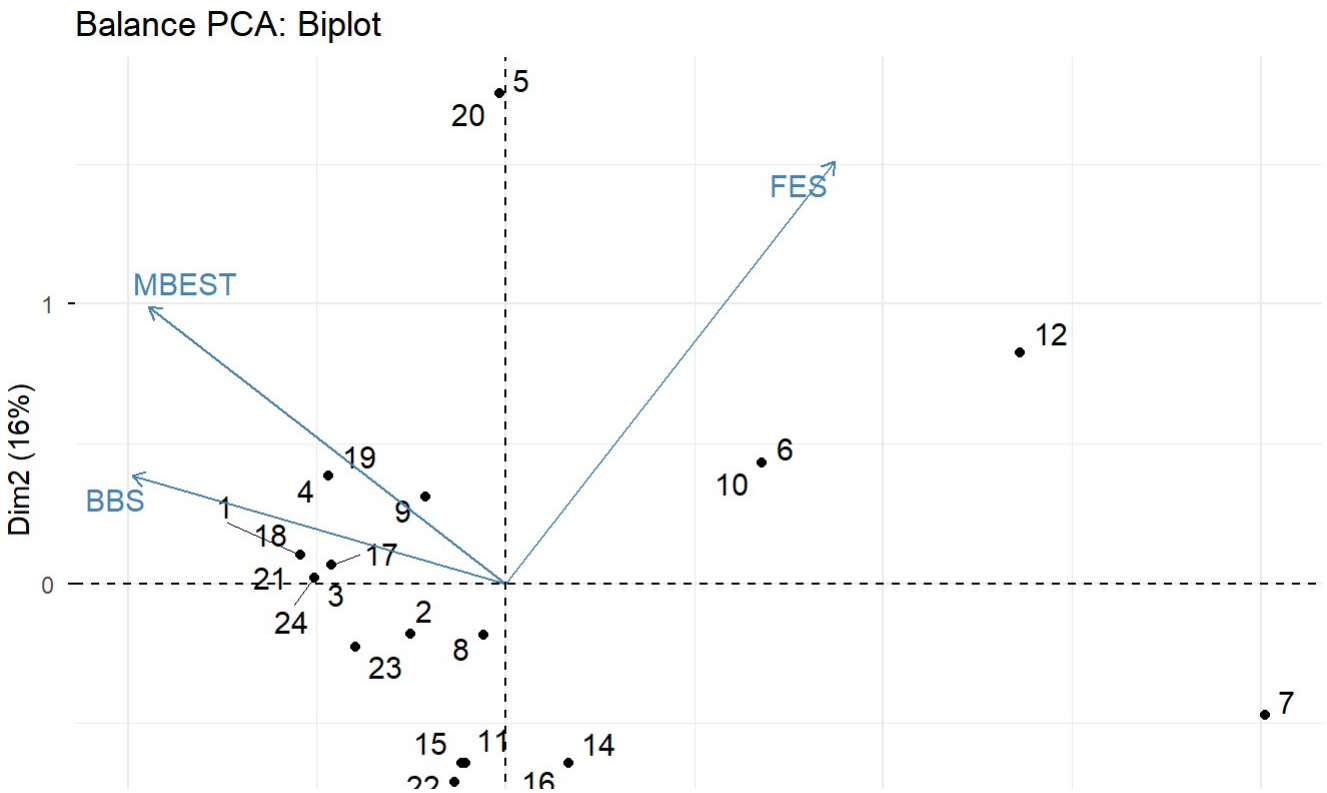

```
fviz_eig(pc_ppe, addlabels = TRUE, main = "PPE PCA: Scree Plot")
```

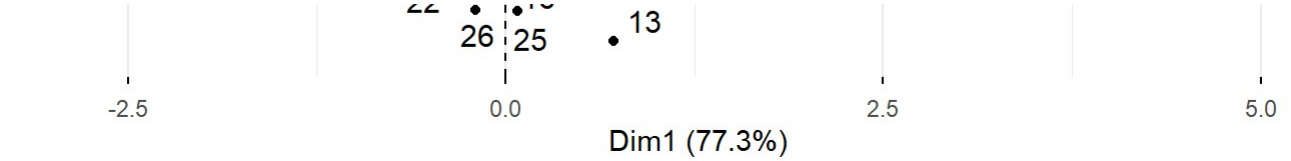
```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```



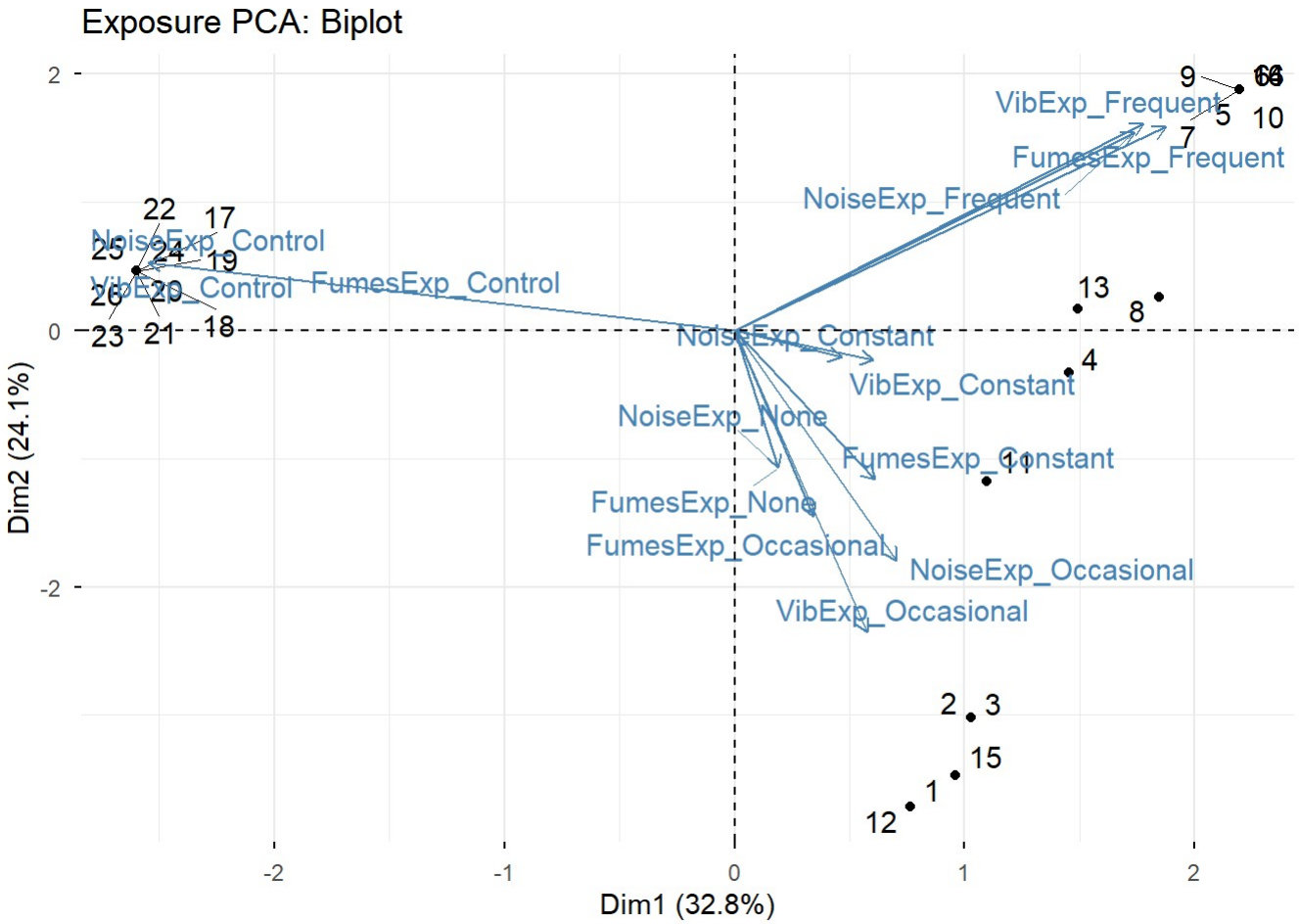

```
## ========================================================
## 8) Biplots
## ========================================================
fviz_pca_biplot(pc_bal, repel = TRUE, title = "Balance PCA: Biplot")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
ℹ The deprecated feature was likely used in the ggpubr package.
  Please report the issue at <https://github.com/kassambara/ggpubr/issues>.
```

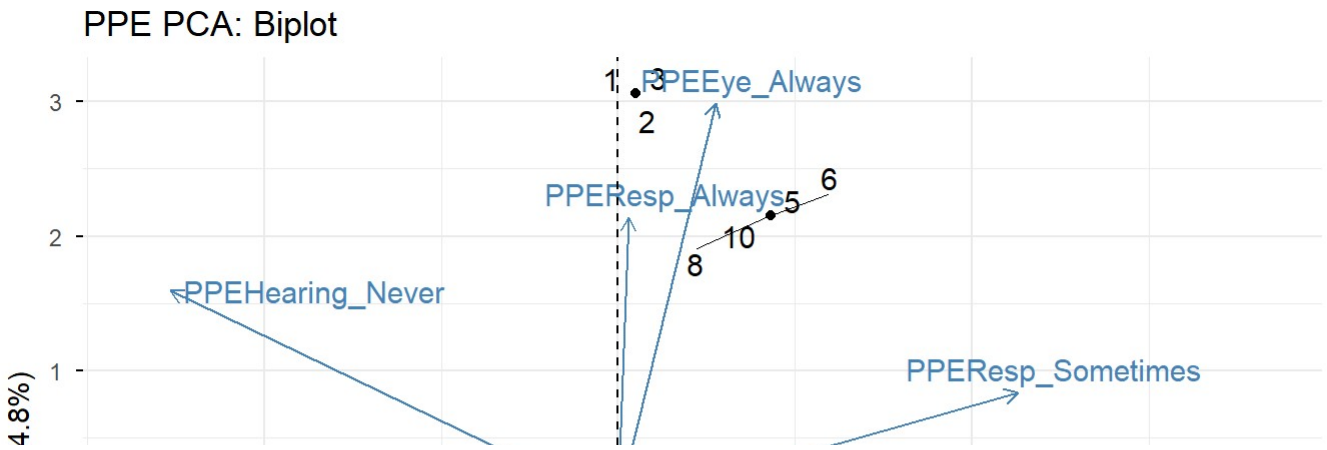


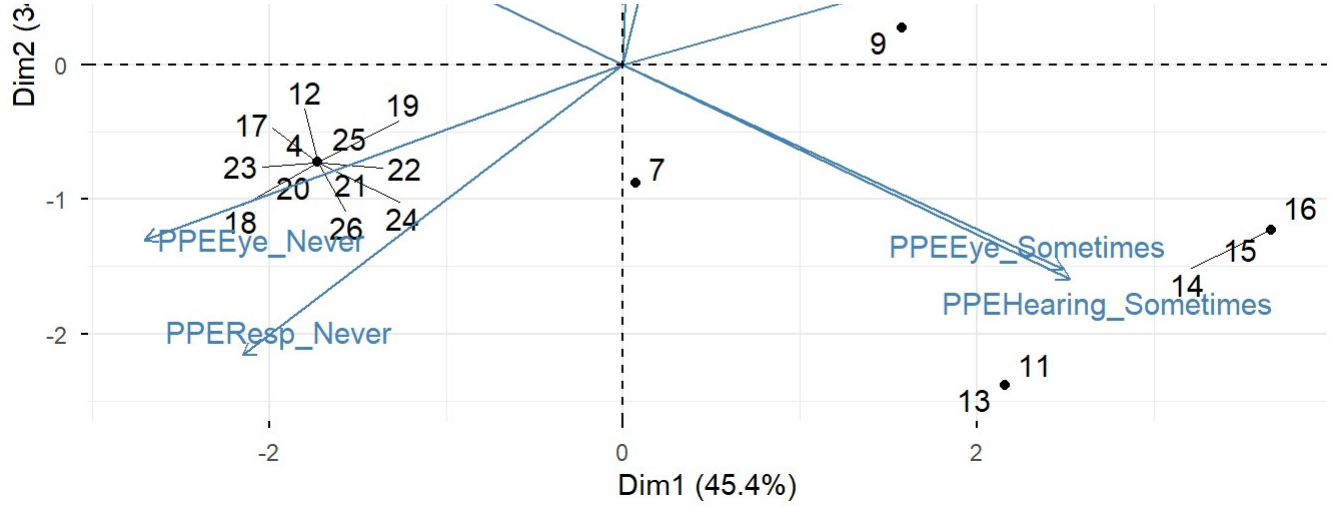



```
fviz_pca_biplot(pc_exp, repel = TRUE, title = "Exposure PCA: Biplot")
```




```
fviz_pca_biplot(pc_ppe, repel = TRUE, title = "PPE PCA: Biplot")
```






```
## ============================================================
## 9) Loadings (PC1)
## ============================================================
cat("\n--- Balance PC1 loadings ---\n")
```

```
--- Balance PC1 loadings ---
```

```
print(pc_bal$rotation[, 1, drop = FALSE])
```

```
             PC1
BBS    -0.6098831
FES     0.5371141
MBEST -0.5827101
```

```
cat("\n--- Exposure PC1 loadings ---\n")
```

```
--- Exposure PC1 loadings ---
```

```
print(pc_exp$rotation[, 1, drop = FALSE])
```

```
                          PC1
NoiseExp_None        0.03405914
NoiseExp_Occasional  0.12638649
NoiseExp_Frequent    0.31126406
NoiseExp_Constant    0.08310222
NoiseExp_Control    -0.45665579
VibExp_Occasional    0.10295859
VibExp_Frequent      0.31854764
VibExp_Constant      0.10819152
VibExp_Control      -0.45665579
FumesExp_None        0.03405914
```

```
FumesExp_None         0.03405914
FumesExp_Occasional   0.06159470
FumesExp_Frequent     0.33586211
FumesExp_Constant     0.10925134
FumesExp_Control     -0.45665579
```

```
cat("\n--- PPE PC1 loadings ---\n")
```

```
--- PPE PC1 loadings ---
```

```
print(pc_ppe$rotation[, 1, drop = FALSE])
```

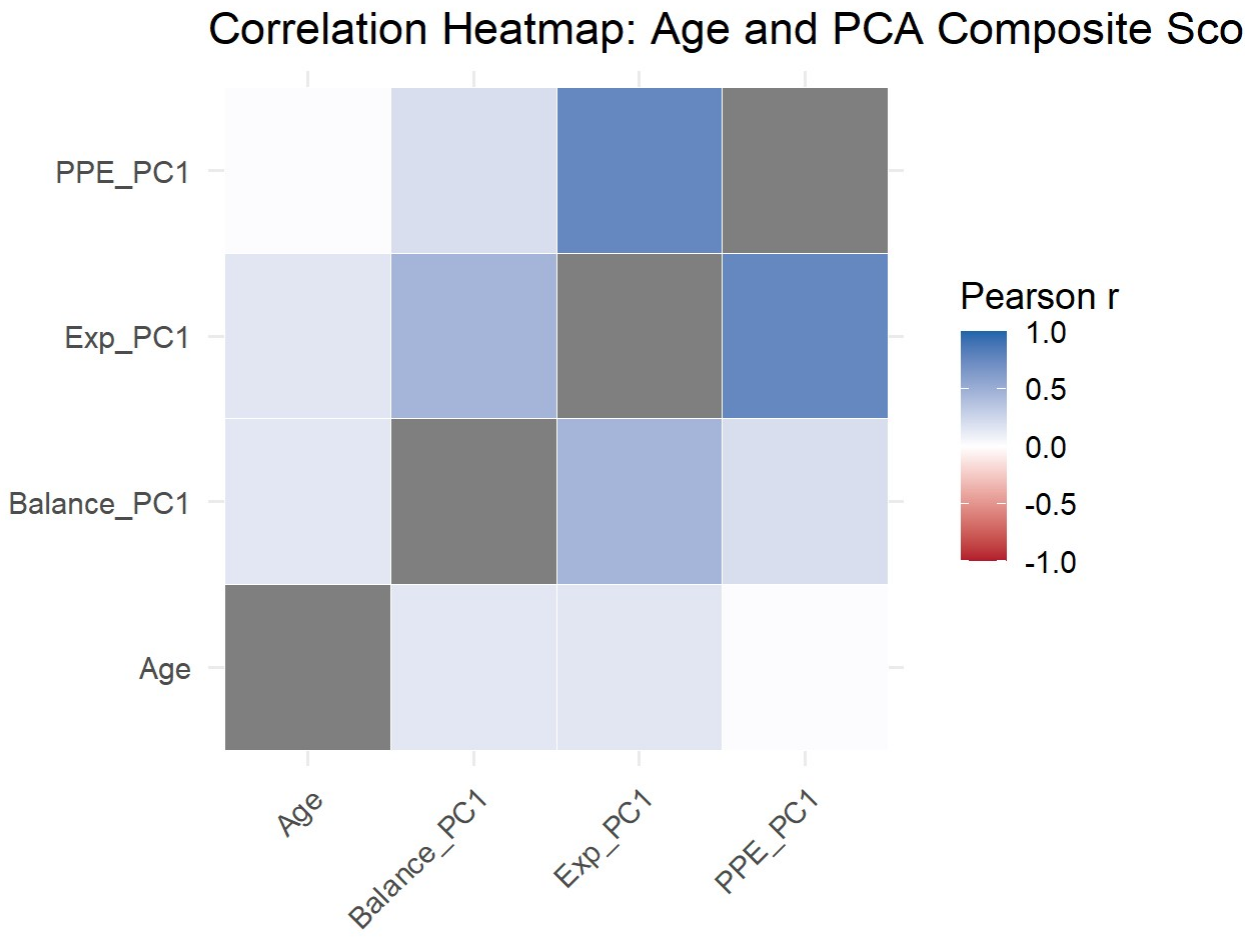
```
                             PC1
PPEHearing_Never     -0.419499101
PPEHearing_Sometimes  0.419499101
PPEResp_Always        0.009966904
PPEResp_Never        -0.356779377
PPEResp_Sometimes     0.375010229
PPEEye_Always         0.091398924
PPEEye_Never         -0.448617073
PPEEye_Sometimes      0.412803139
```

```
## ============================================================
## 10) Correlation heatmap: Age + composite PCs
## ============================================================
corr_df <- wdat %>%
  select(Age, Balance_PC1, Exp_PC1, PPE_PC1) %>%
  mutate(across(everything(), as.numeric)) %>%
  drop_na()

cor_matrix <- correlate(corr_df, method = "pearson", use = "pairwise.complete.obs")
```

```
Correlation computed with
• Method: 'pearson'
• Missing treated using: 'pairwise.complete.obs'
```

```
cor_long <- cor_matrix %>%
  pivot_longer(-term, names_to = "variable2", values_to = "correlation") %>%
  rename(variable = term)

ggplot(cor_long, aes(x = variable, y = variable2, fill = correlation)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(
    low = "#B2182B",
    mid = "white",
    high = "#2166AC",
    midpoint = 0,
    limits = c(-1, 1)
```

```
  ) +
  coord_fixed() +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title = element_blank()) +
  labs(title = "Correlation Heatmap: Age and PCA Composite Scores",
       fill = "Pearson r")
```




```
## ============================================================
## 11) Save wdat with PCs (optional)
## ============================================================
write.csv(wdat, "wdat_with_PCs.csv", row.names = FALSE)
cat("\nSaved: wdat_with_PCs.csv\n")
```

```
Saved: wdat_with_PCs.csv
```

Plot for each PCA

```
library(factoextra)
```

```r
library(ggplot2)

pca_fviz_pack <- function(pc, prefix, group = NULL, title_prefix = NULL) {

  tp <- ifelse(is.null(title_prefix), prefix, title_prefix)

  # 1) Scree plot
  p_scree <- fviz_eig(
    pc, addlabels = TRUE,
    main = paste0(tp, ": Scree plot")
  ) +
    theme_classic(base_size = 12)

  # 2) Variable contributions (PC1)
  p_var_contrib1 <- fviz_contrib(
    pc, choice = "var", axes = 1, top = 20
  ) +
    labs(title = paste0(tp, ": Variable contributions (PC1)")) +
    theme_classic(base_size = 12)

  # 3) Variable contributions (PC2)
  p_var_contrib2 <- fviz_contrib(
    pc, choice = "var", axes = 2, top = 20
  ) +
    labs(title = paste0(tp, ": Variable contributions (PC2)")) +
    theme_classic(base_size = 12)

  # 4) Variables plot (correlation circle)
  p_vars <- fviz_pca_var(
    pc, repel = TRUE
  ) +
    labs(title = paste0(tp, ": Variables (correlation circle)")) +
    theme_classic(base_size = 12)

  # 5) Individuals plot (optionally color by group)
  if (!is.null(group)) {
    # group should be a vector aligned to rows used in the PCA
    p_ind <- fviz_pca_ind(
      pc,
      geom = "point",
      habillage = group,      # color by group
      addEllipses = TRUE,
      ellipse.level = 0.95,
      repel = TRUE
    ) +
      labs(title = paste0(tp, ": Individuals by group")) +
      theme_classic(base_size = 12)
  } else {
    p_ind <- fviz_pca_ind(
      pc, geom = "point", repel = TRUE
    )
```

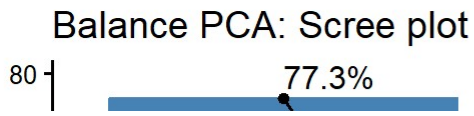
```
    ) +
      labs(title = paste0(tp, ": Individuals")) +
      theme_classic(base_size = 12)
  }

  # 6) Biplot (optional; can be busy but sometimes useful)
  p_biplot <- fviz_pca_biplot(
    pc, repel = TRUE
  ) +
    labs(title = paste0(tp, ": Biplot")) +
    theme_classic(base_size = 12)

  # Return as a named list
  list(
    scree = p_scree,
    var_contrib_pc1 = p_var_contrib1,
    var_contrib_pc2 = p_var_contrib2,
    vars_circle = p_vars,
    individuals = p_ind,
    biplot = p_biplot
  )
}
```

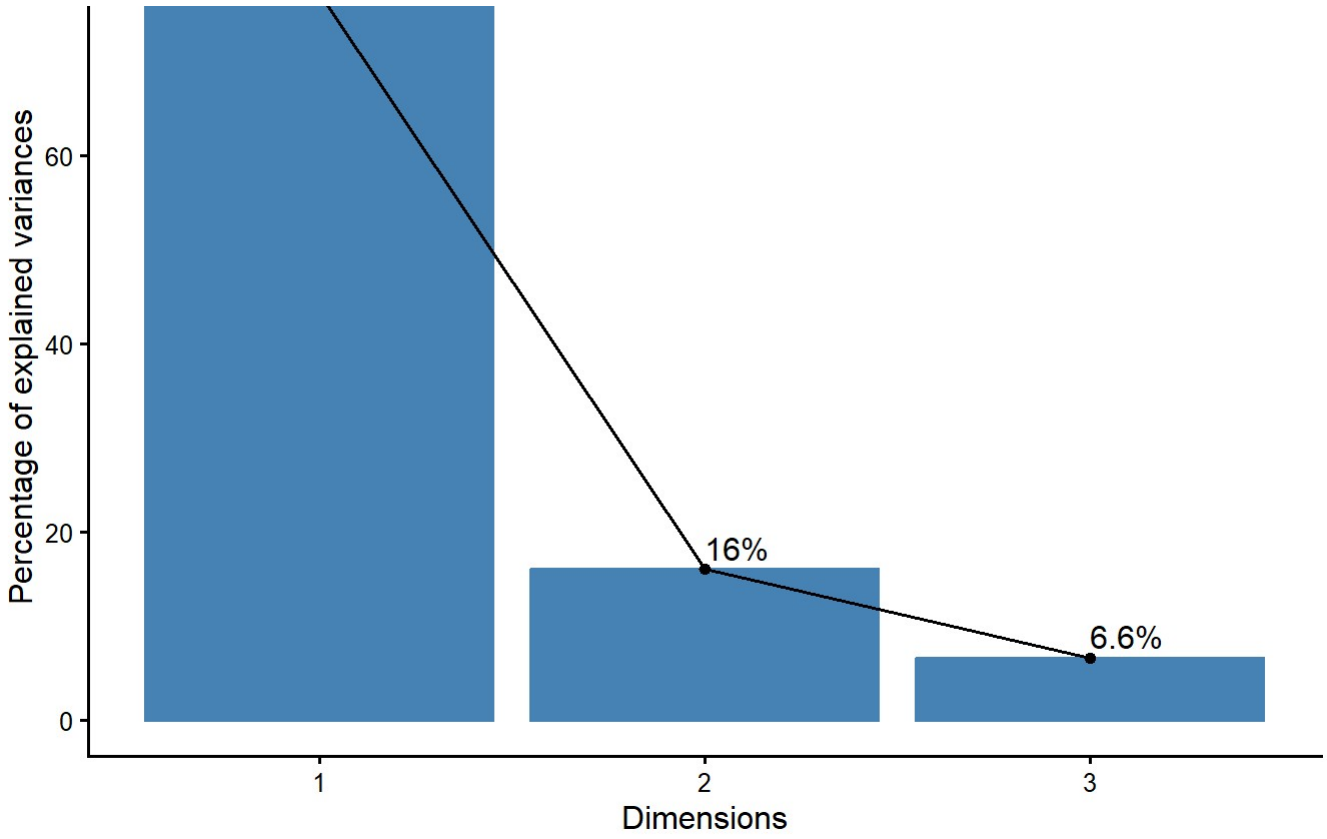
```
# Group vectors aligned to each PCA input data
grp_bal <- wdat$Group[as.integer(rownames(bal_df))]
grp_exp <- wdat$Group[as.integer(rownames(exp_df))]
grp_ppe <- wdat$Group[as.integer(rownames(ppe_df))]

# Build plot packs
plots_bal <- pca_fviz_pack(pc_bal, prefix = "Balance_PCA", group = grp_bal, title_prefix = "Bal
```

```
Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
ℹ Please use tidy evaluation idioms with `aes()`.
ℹ See also `vignette("ggplot2-in-packages")` for more information.
ℹ The deprecated feature was likely used in the factoextra package.
  Please report the issue at <https://github.com/kassambara/factoextra/issues>.
```

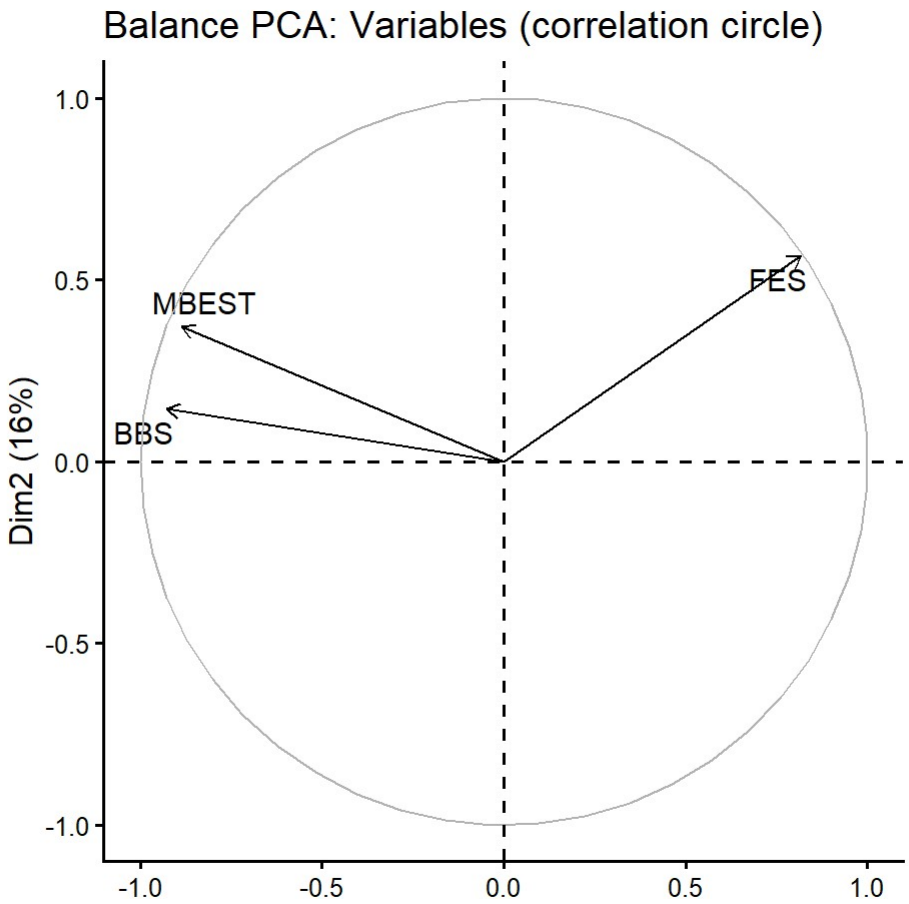
```
plots_exp <- pca_fviz_pack(pc_exp, prefix = "Exposure_PCA", group = grp_exp, title_prefix = "Ex
plots_ppe <- pca_fviz_pack(pc_ppe, prefix = "PPE_PCA", group = grp_ppe, title_prefix = "PPE PCA

# Print any plot to screen
plots_bal$scree
```

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```



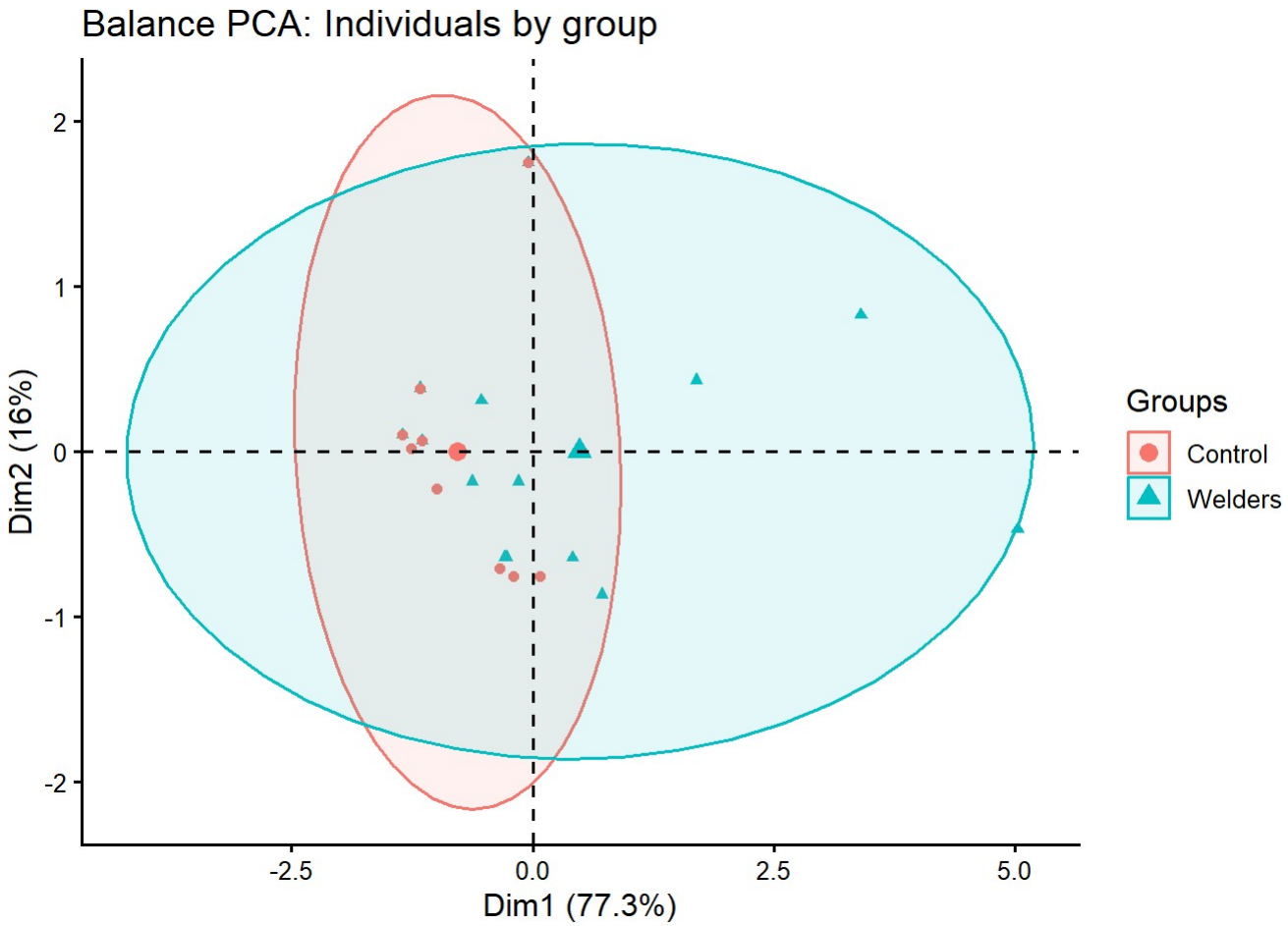



```
plots_bal$vars_circle
```



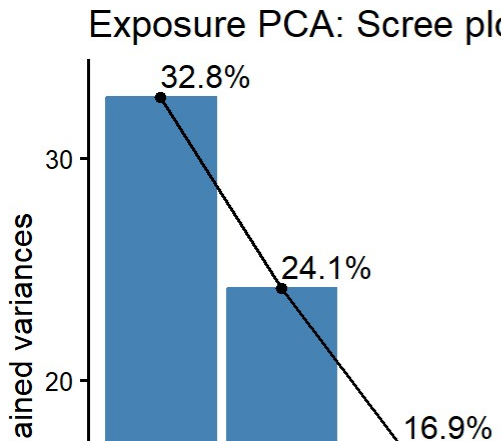
Dim1 (77.3%)

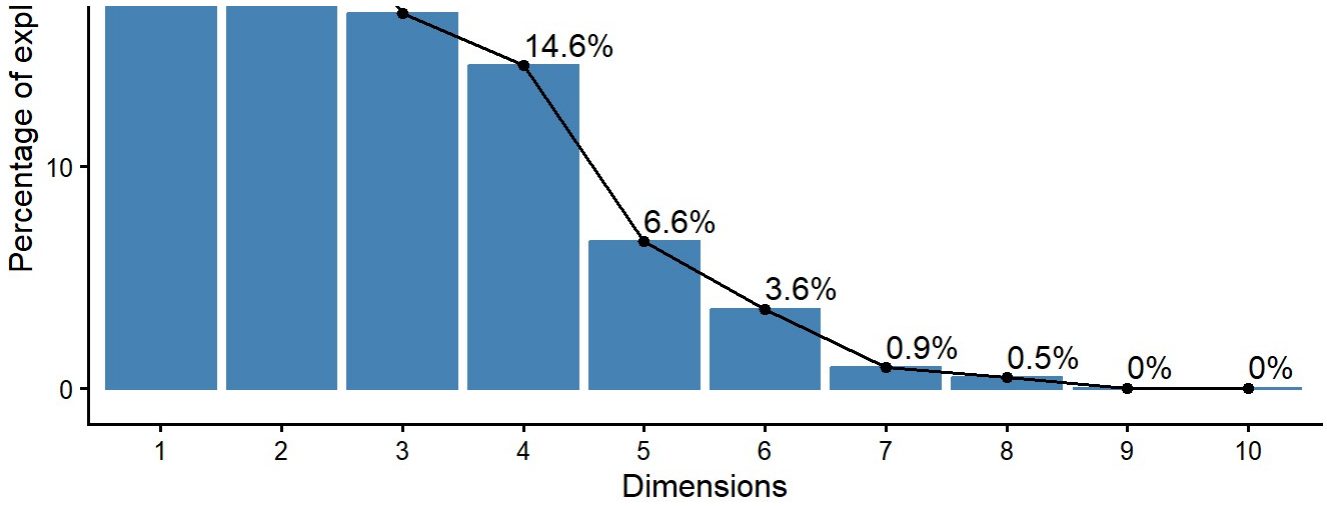
```
plots_bal$individuals
```




```
plots_exp$scree
```

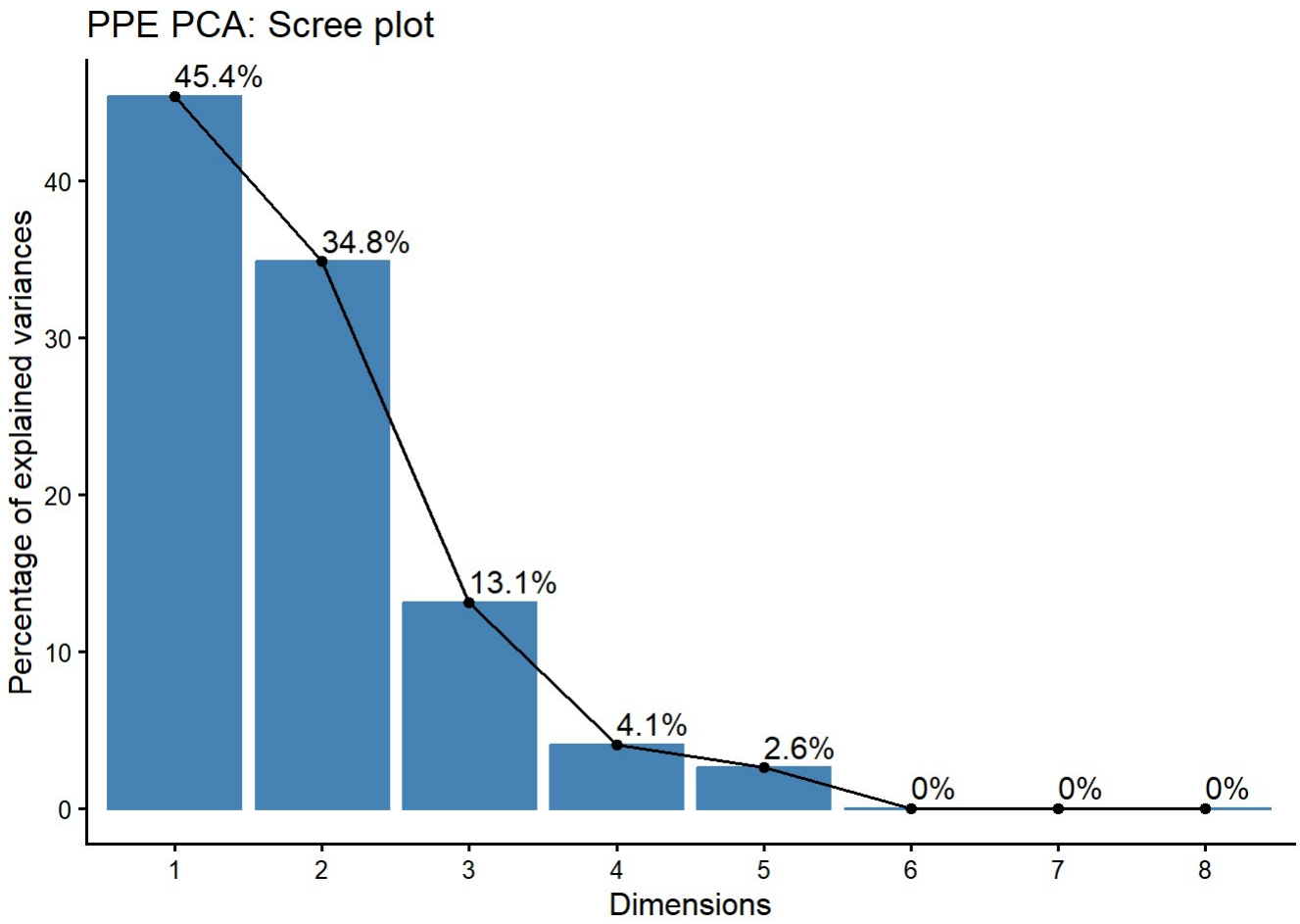
```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```






```
plots_ppe$scree
```

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
Ignoring empty aesthetic: `width`.
```

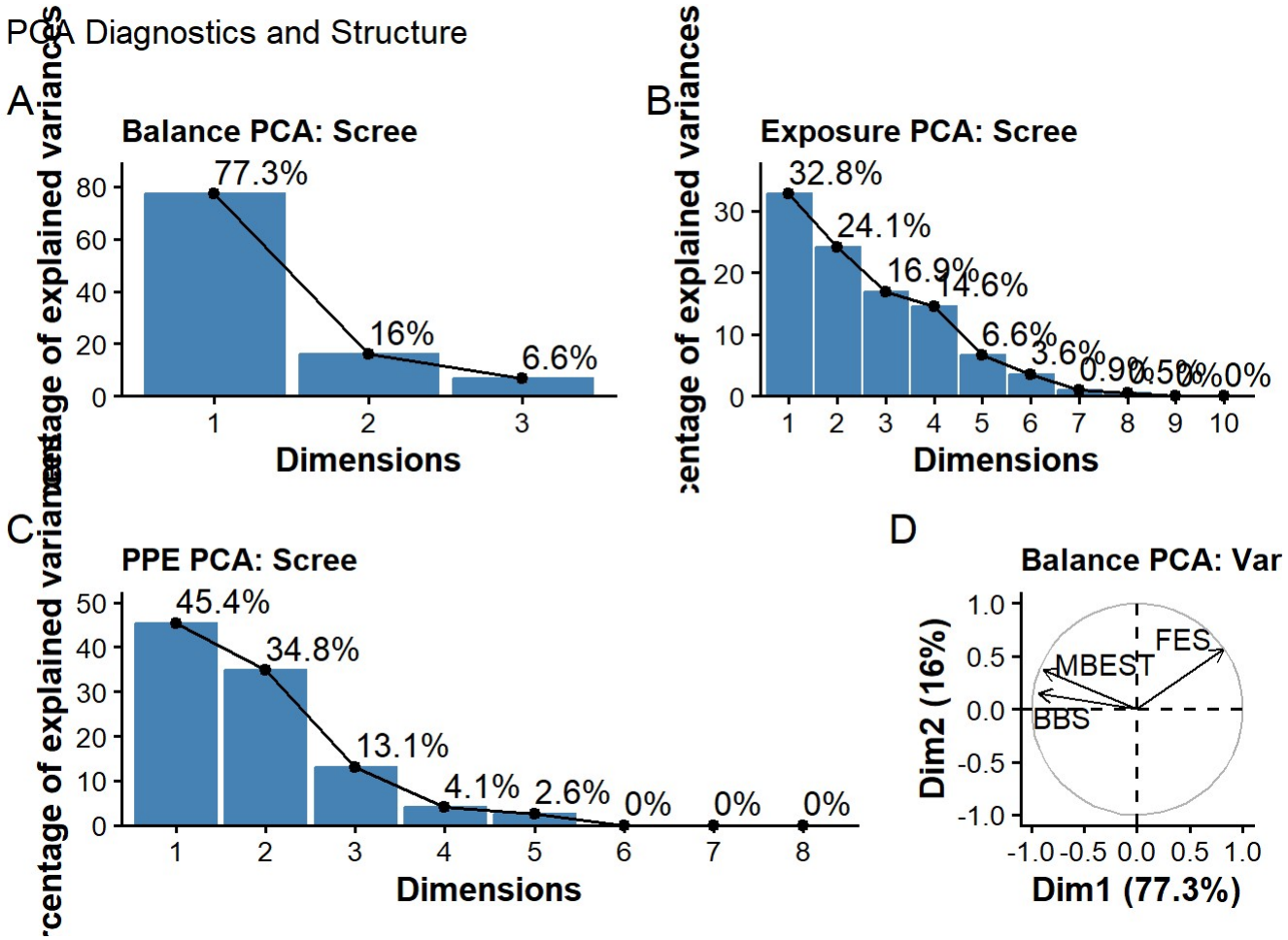


Multi-panel

```
## ============================================================
## PUBLISHABLE MULTI-PANEL FIGURE (Scree + Variable Plots)
## Balance / Exposure / PPE
## ============================================================

library(patchwork)
library(factoextra)
library(ggplot2)

## ---- Global theme ----
base_theme <- theme_classic(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 12, hjust = 0),
    axis.title = element_text(face = "bold"),
    legend.position = "none",
    plot.margin = margin(6, 6, 6, 6)
  )

## ---- Panels ----
p_scree_bal <- fviz_eig(pc_bal, addlabels = TRUE, main = "Balance PCA: Scree") +
  scale_y_continuous(expand = expansion(mult = c(0, 0.15))) +
  coord_cartesian(clip = "off") +
  base_theme + theme(legend.position = "none")

p_scree_exp <- fviz_eig(pc_exp, addlabels = TRUE, main = "Exposure PCA: Scree") +
  scale_y_continuous(expand = expansion(mult = c(0, 0.15))) +
  coord_cartesian(clip = "off") +
  base_theme + theme(legend.position = "none")

p_scree_ppe <- fviz_eig(pc_ppe, addlabels = TRUE, main = "PPE PCA: Scree") +
  scale_y_continuous(expand = expansion(mult = c(0, 0.15))) +
  coord_cartesian(clip = "off") +
  base_theme + theme(legend.position = "none")


p_var_bal <- fviz_pca_var(pc_bal, repel = TRUE, title = "Balance PCA: Variables") + base_theme

## ---- 2x2 layout ----
multi_panel_2x2 <- (p_scree_bal | p_scree_exp) /
                   (p_scree_ppe | p_var_bal) +
  plot_annotation(
    title = "PCA Diagnostics and Structure",
    tag_levels = "A"
  )

print(multi_panel_2x2)
```

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, : Ignoring empty
```

```
aesthetic: `width`.
Ignoring empty aesthetic: `width`.
Ignoring empty aesthetic: `width`.
```




```
## ---- Save ----
ggsave(
  filename = "Fig_PCA_Multipanel_2x2.png",
  plot = multi_panel_2x2,
  width = 10, height = 7.5,
  dpi = 600, bg = "white"
)
```

```
Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, : Ignoring empty
aesthetic: `width`.
Ignoring empty aesthetic: `width`.
Ignoring empty aesthetic: `width`.
```

Section 4: Firth Logistic Regeression on PC variables

```
library(logistf)

## Ensure outcome is coded correctly
```

```
## Ensure outcome is coded correctly
wdat <- wdat %>%
  mutate(FallHist = factor(FallHist, levels = c(0, 1)))

## ------------------------------
## Model 1: Balance PC1
## ------------------------------
firth_bal <- logistf(
  FallHist ~ Age + Balance_PC1,
  data = wdat
)

## ------------------------------
## Model 2: Exposure PC1
## ------------------------------
firth_exp <- logistf(
  FallHist ~ Age + Exp_PC1,
  data = wdat
)

## ------------------------------
## Model 3: PPE PC1
## ------------------------------
firth_ppe <- logistf(
  FallHist ~ Age + PPE_PC1,
  data = wdat
)

## ------------------------------
## Model 4: Full model
## ------------------------------
firth_all <- logistf(
  FallHist ~ Age + Balance_PC1 + Exp_PC1 + PPE_PC1,
  data = wdat
)

## ------------------------------
## Inspect results
## ------------------------------
summary(firth_bal)
```

```
logistf(formula = FallHist ~ Age + Balance_PC1, data = wdat)

Model fitted by Penalized ML
Coefficients:
                  coef   se(coef)   lower 0.95 upper 0.95    Chisq           p
(Intercept) -7.0521802 3.16571410 -15.04642550 -1.5053352 6.747971 0.009385435
Age          0.1564332 0.07300583   0.02735203  0.3396896 5.983045 0.014444043
Balance_PC1 -0.1190299 0.28303722  -0.72188776  0.5395535 0.151781 0.696839168
            method
```

```
(Intercept)      2
Age              2
Balance_PC1      2

Method: 1-Wald, 2-Profile penalized log-likelihood, 3-None

Likelihood ratio test=6.053633 on 2 df, p=0.0484697, n=26
Wald test = 5.329744 on 2 df, p = 0.06960827
```

```
summary(firth_exp)
```

```
logistf(formula = FallHist ~ Age + Exp_PC1, data = wdat)

Model fitted by Penalized ML
Coefficients:
                  coef   se(coef)   lower 0.95 upper 0.95     Chisq          p
(Intercept) -6.7050958 2.99426333 -14.03563191 -1.4163228 6.5245427 0.01063959
Age          0.1479171 0.06859604   0.02527792  0.3135937 5.7912419 0.01610621
Exp_PC1      0.1323288 0.20666473  -0.27914864  0.5778884 0.4008197 0.52666622
            method
(Intercept)      2
Age              2
Exp_PC1          2

Method: 1-Wald, 2-Profile penalized log-likelihood, 3-None

Likelihood ratio test=6.458066 on 2 df, p=0.03959577, n=26
Wald test = 5.391789 on 2 df, p = 0.06748198
```

```
summary(firth_ppe)
```

```
logistf(formula = FallHist ~ Age + PPE_PC1, data = wdat)

Model fitted by Penalized ML
Coefficients:
                   coef   se(coef)   lower 0.95 upper 0.95     Chisq
(Intercept) -6.83730599 3.02267958 -14.35060978 -1.5444018 6.8492294
Age          0.15184265 0.06964521   0.02847426  0.3224511 6.0929864
PPE_PC1      0.07553396 0.22856765  -0.40205392  0.5536211 0.1048552
                     p method
(Intercept) 0.008867951      2
Age         0.013571950      2
PPE_PC1     0.746079196      2

Method: 1-Wald, 2-Profile penalized log-likelihood, 3-None

Likelihood ratio test=6.120347 on 2 df, p=0.04687957, n=26
Wald test = 5.303257 on 2 df, p = 0.07053625
```

```
summary(firth_all)
```

```
logistf(formula = FallHist ~ Age + Balance_PC1 + Exp_PC1 + PPE_PC1,
    data = wdat)

Model fitted by Penalized ML
Coefficients:
                  coef   se(coef)   lower 0.95 upper 0.95      Chisq          p
(Intercept) -6.3633649 3.09739509 -15.28858083 -0.9880406 5.70670200 0.01690026
Age          0.1398874 0.07065278   0.01526569  0.3388758 5.00724691 0.02524142
Balance_PC1 -0.2225962 0.31092509  -0.94137540  0.4702157 0.43092324 0.51153628
Exp_PC1      0.2756537 0.34059247  -0.39511415  1.0537775 0.64286245 0.42267616
PPE_PC1     -0.1155883 0.36307458  -0.98404169  0.6272631 0.08997165 0.76421321
            method
(Intercept)      2
Age              2
Balance_PC1      2
Exp_PC1          2
PPE_PC1          2

Method: 1-Wald, 2-Profile penalized log-likelihood, 3-None

Likelihood ratio test=6.336027 on 4 df, p=0.1754197, n=26
Wald test = 5.276077 on 4 df, p = 0.2601247
```

Table #3 OR/CI of the all the Models

```
library(dplyr)
library(kableExtra)

## ----------------------------
## Helper: format p-values
## ----------------------------
fmt_p <- function(p) {
  ifelse(is.na(p), "",
         ifelse(p < 0.001, "<0.001", sprintf("%.3f", p)))
}

## ----------------------------
## Extract results from logistf
## ----------------------------
extract_firth_results <- function(model, model_name, drop_intercept = TRUE) {

  # coefficients and p-values
  coef <- model$coefficients
  pval <- model$prob

  # 95% CI (profile likelihood from logistf)
```

```
  # Note: logistf stores ci.lower/ci.upper (already on log-odds scale)
  # These are typically more stable than calling confint()
  ci_low <- model$ci.lower
  ci_up  <- model$ci.upper

  out <- tibble(
    Model = model_name,
    Predictor = names(coef),
    OR = exp(coef),
    CI_Lower = exp(ci_low),
    CI_Upper = exp(ci_up),
    p_value = pval
  )

  if (drop_intercept) {
    out <- out %>% filter(Predictor != "(Intercept)")
  }

  out %>%
    mutate(
      OR = sprintf("%.2f", OR),
      `95% CI` = paste0(sprintf("%.2f", CI_Lower), "–", sprintf("%.2f", CI_Upper)),
      p_value = fmt_p(p_value)
    ) %>%
    select(Model, Predictor, OR, `95% CI`, p_value)
}

## ----------------------------
## Your fitted models
## ----------------------------
models <- list(
  Balance  = firth_bal,
  Exposure = firth_exp,
  PPE      = firth_ppe,
  Full     = firth_all
)

## ----------------------------
## Build Table 3
## ----------------------------
table3_df <- bind_rows(lapply(names(models), function(nm) {
  extract_firth_results(models[[nm]], nm, drop_intercept = TRUE)
}))

## Optional: nicer predictor labels (edit to your liking)
predictor_labels <- c(
  "Age" = "Age (years)",
  "Balance_PC1" = "Balance PC1",
  "Exp_PC1" = "Exposure PC1",
  "PPE_PC1" = "PPE PC1"
```

```
)

table3_df <- table3_df %>%
  mutate(
    Predictor = ifelse(Predictor %in% names(predictor_labels),
                       predictor_labels[Predictor],
                       Predictor)
  )

## ----------------------------
## Display Table 3
## ----------------------------
kbl_table3 <- table3_df %>%
  kbl(
    format = "html",
    caption = "Table 3. Firth Logistic Regression Results (Odds Ratios, 95% CI, p-values)",
    align = "l",
    escape = TRUE
  ) %>%
  kable_styling(full_width = FALSE, position = "center") %>%
  row_spec(0, bold = TRUE)

kbl_table3
```

Table 3. Firth Logistic Regression Results (Odds Ratios, 95% CI, p-values)

Model	Predictor	OR	95% CI	p_value
Balance	Age (years)	1.17	1.03–1.40	0.014
Balance	Balance PC1	0.89	0.49–1.72	0.697
Exposure	Age (years)	1.16	1.03–1.37	0.016
Exposure	Exposure PC1	1.14	0.76–1.78	0.527
PPE	Age (years)	1.16	1.03–1.38	0.014
PPE	PPE PC1	1.08	0.67–1.74	0.746
Full	Age (years)	1.15	1.02–1.40	0.025
Full	Balance PC1	0.80	0.39–1.60	0.512
Full	Exposure PC1	1.32	0.67–2.87	0.423
Full	PPE PC1	0.89	0.37–1.87	0.764

```
library(webshot2)

html_file <- "Table3_firth.html"
png_file  <- "Table3_firth.png"

save_kable(kbl_table3, file = html_file, self_contained = TRUE)
```

```
webshot(
  url = html_file,
  file = png_file,
  zoom = 2,
  vwidth = 1600,
  vheight = 900,
  delay = 0.3
)
```

Table 3. Firth Logistic Regression Results (Odds Ratios, 95% CI, p-values)

Model	Predictor	OR	95% CI	p_value
Balance	Age (years)	1.17	1.03–1.40	0.014
Balance	Balance PC1	0.89	0.49–1.72	0.697
Exposure	Age (years)	1.16	1.03–1.37	0.016
Exposure	Exposure PC1	1.14	0.76–1.78	0.527
PPE	Age (years)	1.16	1.03–1.38	0.014
PPE	PPE PC1	1.08	0.67–1.74	0.746
Full	Age (years)	1.15	1.02–1.40	0.025
Full	Balance PC1	0.80	0.39–1.60	0.512
Full	Exposure PC1	1.32	0.67–2.87	0.423
Full	PPE PC1	0.89	0.37–1.87	0.764

```
message("Saved HTML: ", html_file)
```

```
Saved HTML: Table3_firth.html
```

```
message("Saved PNG:  ", png_file)
```

```
Saved PNG:  Table3_firth.png
```

Comparison of Models

```
library(dplyr)

summarize_firth <- function(model, model_name) {
  coef_names <- names(model$coefficients)
  coefs <- round(model$coefficients, 3)
  pvals <- signif(model$prob, 3)
```

```
  tibble(
    Model = model_name,
    LogLikelihood = model$loglik[2],  # log-likelihood of fitted model
    AIC = model$aic,
    Coefficients = paste(coef_names, coefs, collapse = ", "),
    P_values = paste(coef_names, pvals, collapse = ", ")
  )
}

results <- bind_rows(
  summarize_firth(firth_bal, "Balance_PC1"),
  summarize_firth(firth_exp, "Exp_PC1"),
  summarize_firth(firth_ppe, "PPE_PC1"),
  summarize_firth(firth_all, "All_factors")
)

print(results)
```

```
# A tibble: 4 × 4
  Model       LogLikelihood Coefficients                                  P_values
  <chr>               <dbl> <chr>                                         <chr>
1 Balance_PC1         -12.2  (Intercept) -7.052, Age 0.156, Balance_PC1… (Interc…
2 Exp_PC1             -11.9  (Intercept) -6.705, Age 0.148, Exp_PC1 0.1… (Interc…
3 PPE_PC1             -12.0  (Intercept) -6.837, Age 0.152, PPE_PC1 0.0… (Interc…
4 All_factors          -9.63 (Intercept) -6.363, Age 0.14, Balance_PC1 … (Interc…
```

Model Comparison

```
## ============================================================
## FIRTH LOGISTIC REGRESSION — MODEL COMPARISON (ROBUST)
## ============================================================

library(dplyr)
library(tibble)
library(logistf)
library(knitr)
library(kableExtra)

## ---- Ensure outcome coding ----
wdat <- wdat %>%
  mutate(FallHist = factor(FallHist, levels = c(0, 1)))

## ---- Fit Firth models ----
firth_bal <- logistf(FallHist ~ Age + Balance_PC1, data = wdat)
firth_exp <- logistf(FallHist ~ Age + Exp_PC1,     data = wdat)
firth_ppe <- logistf(FallHist ~ Age + PPE_PC1,     data = wdat)
```

```r
firth_all <- logistf(FallHist ~ Age + Balance_PC1 + Exp_PC1 + PPE_PC1, data = wdat)

## ============================================================
## TABLE 3 — Odds Ratios (OR), 95% CI, p-values
## ============================================================
extract_firth_or <- function(model, model_name) {
  coefs <- model$coefficients
  ci    <- suppressMessages(confint(model))
  p     <- model$prob

  tibble(
    Model = model_name,
    Predictor = names(coefs),
    OR = round(exp(coefs), 3),
    CI_Lower = round(exp(ci[, 1]), 3),
    CI_Upper = round(exp(ci[, 2]), 3),
    p_value = signif(p, 3)
  )
}

table3_or <- bind_rows(
  extract_firth_or(firth_bal, "Age + Balance_PC1"),
  extract_firth_or(firth_exp, "Age + Exp_PC1"),
  extract_firth_or(firth_ppe, "Age + PPE_PC1"),
  extract_firth_or(firth_all, "Full model")
)

kbl(table3_or,
    caption = "Table 3. Firth logistic regression results (OR, 95% CI, p-values)",
    align = "l") %>%
  kable_styling(full_width = FALSE, position = "center",
                bootstrap_options = c("striped", "condensed")) %>%
  row_spec(0, bold = TRUE)
```

Table 3. Firth logistic regression results (OR, 95% CI, p-values)

Model	Predictor	OR	CI_Lower	CI_Upper	p_value
Age + Balance_PC1	(Intercept)	0.001	0.000	0.222	0.00939
Age + Balance_PC1	Age	1.169	1.028	1.405	0.01440
Age + Balance_PC1	Balance_PC1	0.888	0.486	1.715	0.69700
Age + Exp_PC1	(Intercept)	0.001	0.000	0.243	0.01060
Age + Exp_PC1	Age	1.159	1.026	1.368	0.01610
Age + Exp_PC1	Exp_PC1	1.141	0.756	1.782	0.52700
Age + PPE_PC1	(Intercept)	0.001	0.000	0.213	0.00887
Age + PPE_PC1	Age	1.164	1.029	1.381	0.01360
Age + PPE_PC1	PPE_PC1	1.078	0.669	1.740	0.74600

Full model	(Intercept)	0.002	0.000	0.372	0.01690
Full model	Age	1.150	1.015	1.403	0.02520
Full model	Balance_PC1	0.800	0.390	1.600	0.51200
Full model	Exp_PC1	1.317	0.674	2.868	0.42300
Full model	PPE_PC1	0.891	0.374	1.872	0.76400

```
## ============================================================
## MODEL SUMMARY TABLE — LogLik, AIC (computed), LRT, Wald
## ============================================================
summarize_firth <- function(model, model_name) {

  ll  <- as.numeric(model$loglik[2])              # fitted log-likelihood
  k   <- length(model$coefficients)               # number of parameters (incl intercept)
  aic <- 2 * k - 2 * ll                           # AIC definition

  # Some logistf versions store probtest as a matrix with rownames
  pt <- model$probtest

  # Safe extraction in case naming differs
  get_pt <- function(row, col) {
    if (is.null(pt)) return(NA)
    if (!row %in% rownames(pt)) return(NA)
    if (!col %in% colnames(pt)) return(NA)
    as.numeric(pt[row, col])
  }

  tibble(
    Model = model_name,
    n = model$n,
    k = k,
    LogLikelihood = round(ll, 3),
    AIC = round(aic, 3),

    LRT_stat = round(get_pt("likelihood ratio", "Chisq"), 3),
    LRT_df   = get_pt("likelihood ratio", "df"),
    LRT_p    = signif(get_pt("likelihood ratio", "p"), 3),

    Wald_stat = round(get_pt("wald", "Chisq"), 3),
    Wald_df   = get_pt("wald", "df"),
    Wald_p    = signif(get_pt("wald", "p"), 3)
  )
}

model_summary <- bind_rows(
  summarize_firth(firth_bal, "Age + Balance_PC1"),
  summarize_firth(firth_exp, "Age + Exp_PC1"),
  summarize_firth(firth_ppe, "Age + PPE_PC1"),
```

```
  summarize_firth(firth_all, "Full model")
)

kbl(model_summary,
    caption = "Firth logistic regression model fit statistics",
    align = "l") %>%
  kable_styling(full_width = FALSE, position = "center",
                bootstrap_options = c("striped", "condensed", "hover")) %>%
  row_spec(0, bold = TRUE)
```

Firth logistic regression model fit statistics

Model	n	k	LogLikelihood	AIC	LRT_stat	LRT_df	LRT_p	Wald_stat	Wald_df	Wald_p
Age + Balance_PC1	26	3	-12.238	30.476	NA	NA	NA	NA	NA	NA
Age + Exp_PC1	26	3	-11.898	29.796	NA	NA	NA	NA	NA	NA
Age + PPE_PC1	26	3	-12.004	30.008	NA	NA	NA	NA	NA	NA
Full model	26	5	-9.632	29.263	NA	NA	NA	NA	NA	NA

```
## ============================================================
## NESTED MODEL COMPARISONS (vs Full model)
## ============================================================
lr_comp <- function(smaller, larger, name_small, name_large) {
  chisq <- 2 * (as.numeric(larger$loglik[2]) - as.numeric(smaller$loglik[2]))
  df <- length(larger$coefficients) - length(smaller$coefficients)
  p <- pchisq(chisq, df = df, lower.tail = FALSE)

  tibble(
    Comparison = paste0(name_large, " vs ", name_small),
    Chisq = round(chisq, 3),
    df = df,
    p_value = signif(p, 3)
  )
}

lrt_table <- bind_rows(
  lr_comp(firth_bal, firth_all, "Age + Balance_PC1", "Full model"),
  lr_comp(firth_exp, firth_all, "Age + Exp_PC1", "Full model"),
  lr_comp(firth_ppe, firth_all, "Age + PPE_PC1", "Full model")
)

kbl(lrt_table,
    caption = "Likelihood-ratio tests comparing reduced models to the full model",
    align = "l") %>%
  kable_styling(full_width = FALSE, position = "center",
                bootstrap_options = c("striped", "condensed")) %>%
  row_spec(0, bold = TRUE)
```

Likelihood-ratio tests comparing reduced models to the full model

Comparison	Chisq	df	p_value
Full model vs Age + Balance_PC1	5.213	2	0.0738
Full model vs Age + Exp_PC1	4.533	2	0.1040
Full model vs Age + PPE_PC1	4.745	2	0.0932

```
## ============================================================
## EXPORTS
## ============================================================
write.csv(table3_or, "Table3_Firth_ORs.csv", row.names = FALSE)
write.csv(model_summary, "Firth_Model_Summaries.csv", row.names = FALSE)
write.csv(lrt_table, "Firth_Model_LRT_Comparisons.csv", row.names = FALSE)
```

Forest Plot

```
library(dplyr)
library(ggplot2)
library(stringr)

#--------------------------------
# Helper: build a tidy OR table from a logistf model
#--------------------------------
tidy_logistf_or <- function(model, model_name = "Model", drop_intercept = TRUE) {

  coefs <- model$coefficients
  ci_l  <- model$ci.lower
  ci_u  <- model$ci.upper
  pval  <- model$prob

  df <- tibble(
    Model = model_name,
    Term = names(coefs),
    OR = exp(coefs),
    CI_low = exp(ci_l),
    CI_high = exp(ci_u),
    p = as.numeric(pval)
  )

  if (drop_intercept) df <- df %>% filter(Term != "(Intercept)")

  df
}

#--------------------------------
```

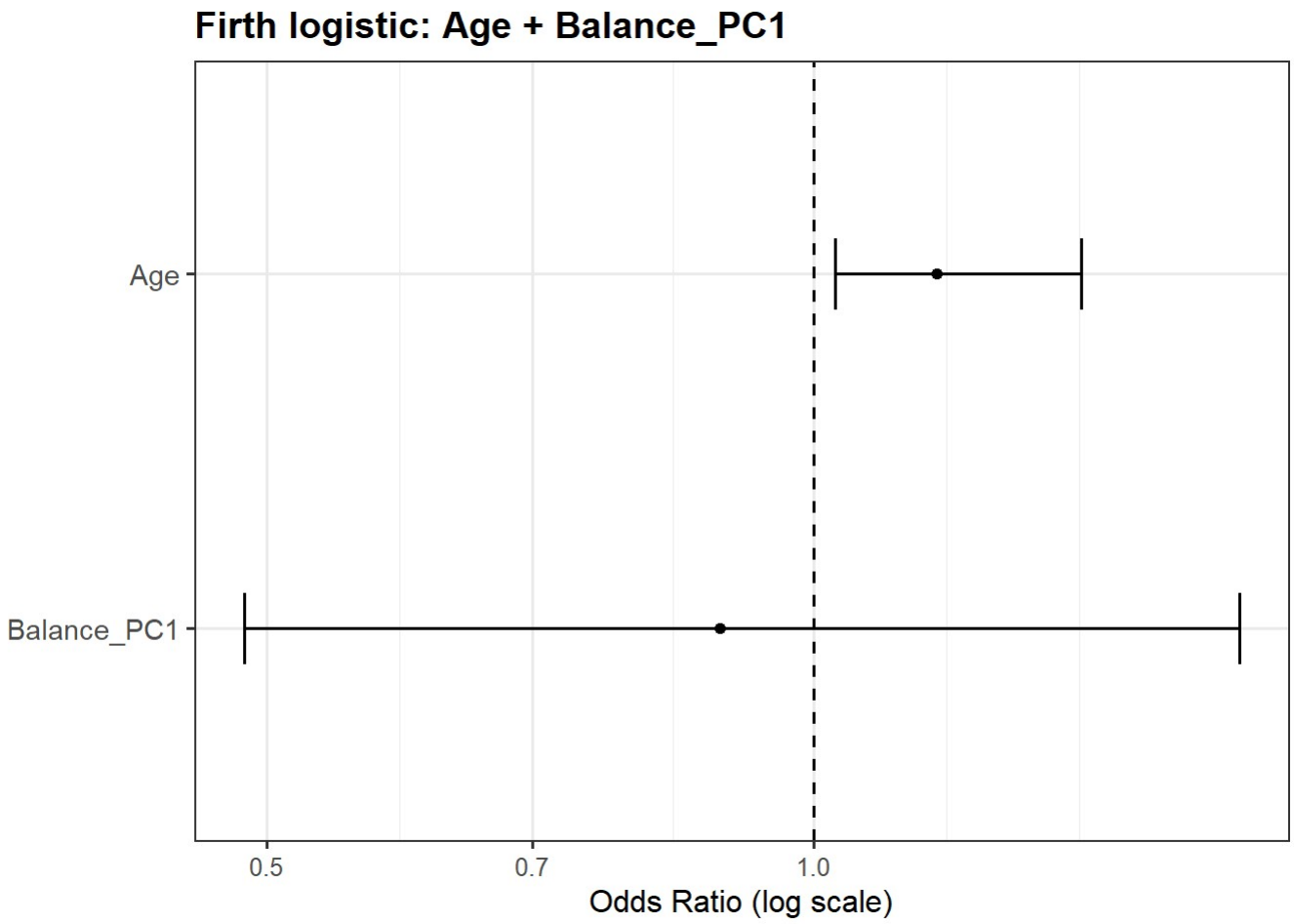
```
# Helper: forest plot for one model
#------------------------------
plot_firth_forest <- function(df, title = NULL) {

  df <- df %>%
    mutate(
      Term = factor(Term, levels = rev(Term)),  # top-to-bottom order
      p_lab = case_when(
        is.na(p) ~ "",
        p < 0.001 ~ "p<0.001",
        TRUE ~ paste0("p=", sprintf("%.3f", p))
      ),
      or_lab = paste0(sprintf("%.2f", OR),
                      " (", sprintf("%.2f", CI_low), "–", sprintf("%.2f", CI_high), ")")
    )

  ggplot(df, aes(x = OR, y = Term)) +
    geom_vline(xintercept = 1, linetype = 2) +
    geom_point() +
    geom_errorbarh(aes(xmin = CI_low, xmax = CI_high), height = 0.2) +
    scale_x_log10() +
    labs(
      title = title,
      x = "Odds Ratio (log scale)",
      y = NULL
    ) +
    theme_bw(base_size = 12) +
    theme(
      plot.title = element_text(face = "bold"),
      axis.text.y = element_text(size = 11)
    )
}

#------------------------------
# Build plot data for each model
#------------------------------
df_bal <- tidy_logistf_or(firth_bal, "Age + Balance_PC1")
df_exp <- tidy_logistf_or(firth_exp, "Age + Exp_PC1")
df_ppe <- tidy_logistf_or(firth_ppe, "Age + PPE_PC1")
df_all <- tidy_logistf_or(firth_all, "Full model")

#------------------------------
# Create plots
#------------------------------
p_bal <- plot_firth_forest(df_bal, "Firth logistic: Age + Balance_PC1")
```

```
Warning: `geom_errorbarh()` was deprecated in ggplot2 4.0.0.
ℹ Please use the `orientation` argument of `geom_errorbar()` instead.
```

```
p_exp <- plot_firth_forest(df_exp, "Firth logistic: Age + Exp_PC1")
```

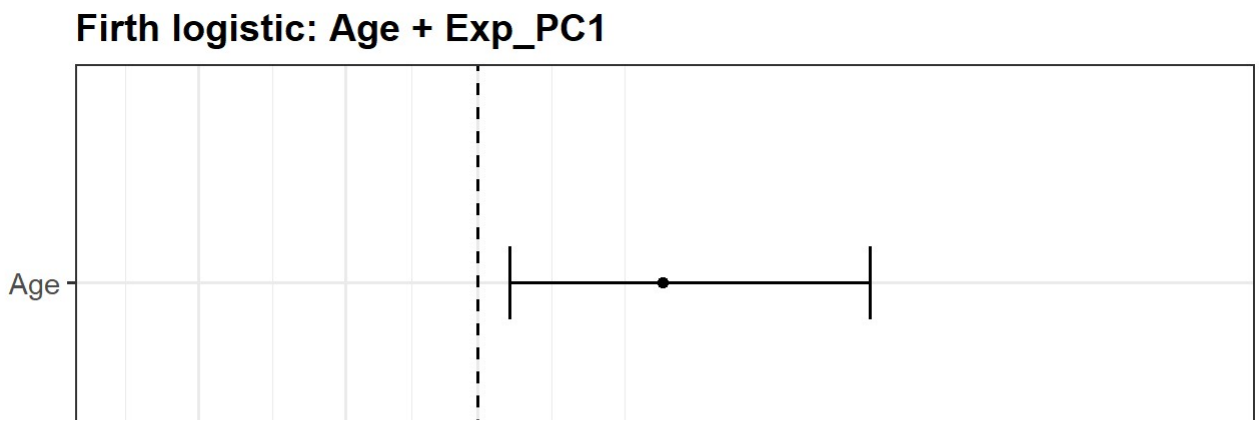
```
p_exp <- plot_firth_forest(df_exp, "Firth logistic: Age + Exp_PC1")
p_ppe <- plot_firth_forest(df_ppe, "Firth logistic: Age + PPE_PC1")
p_all <- plot_firth_forest(df_all, "Firth logistic: Full model")

p_bal
```

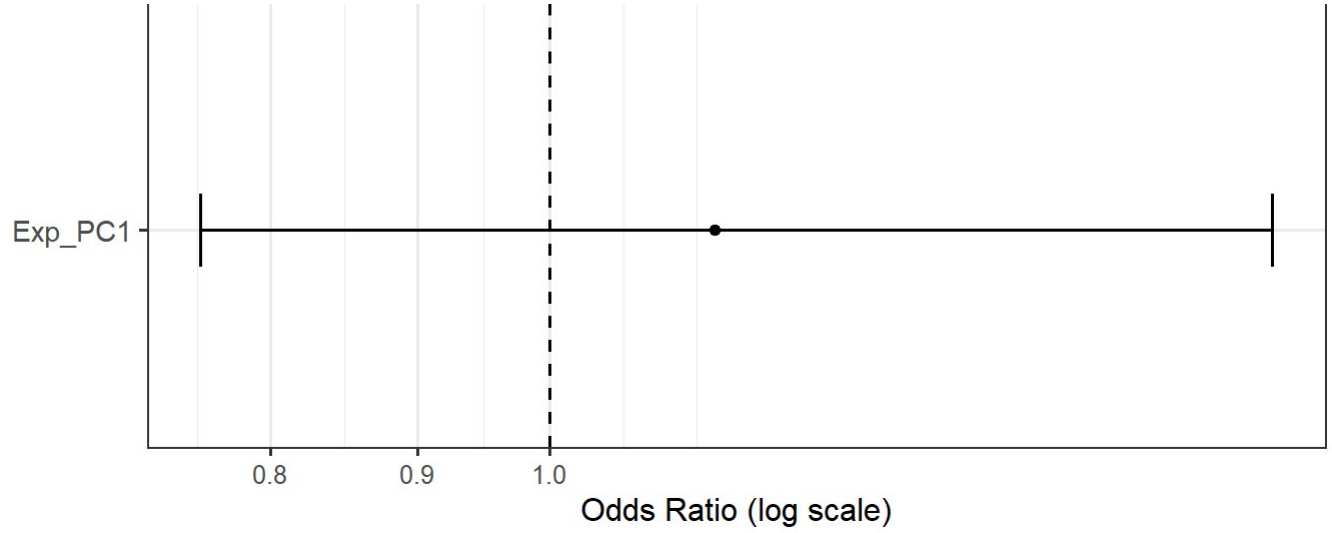
```
`height` was translated to `width`.
```




```
p_exp
```

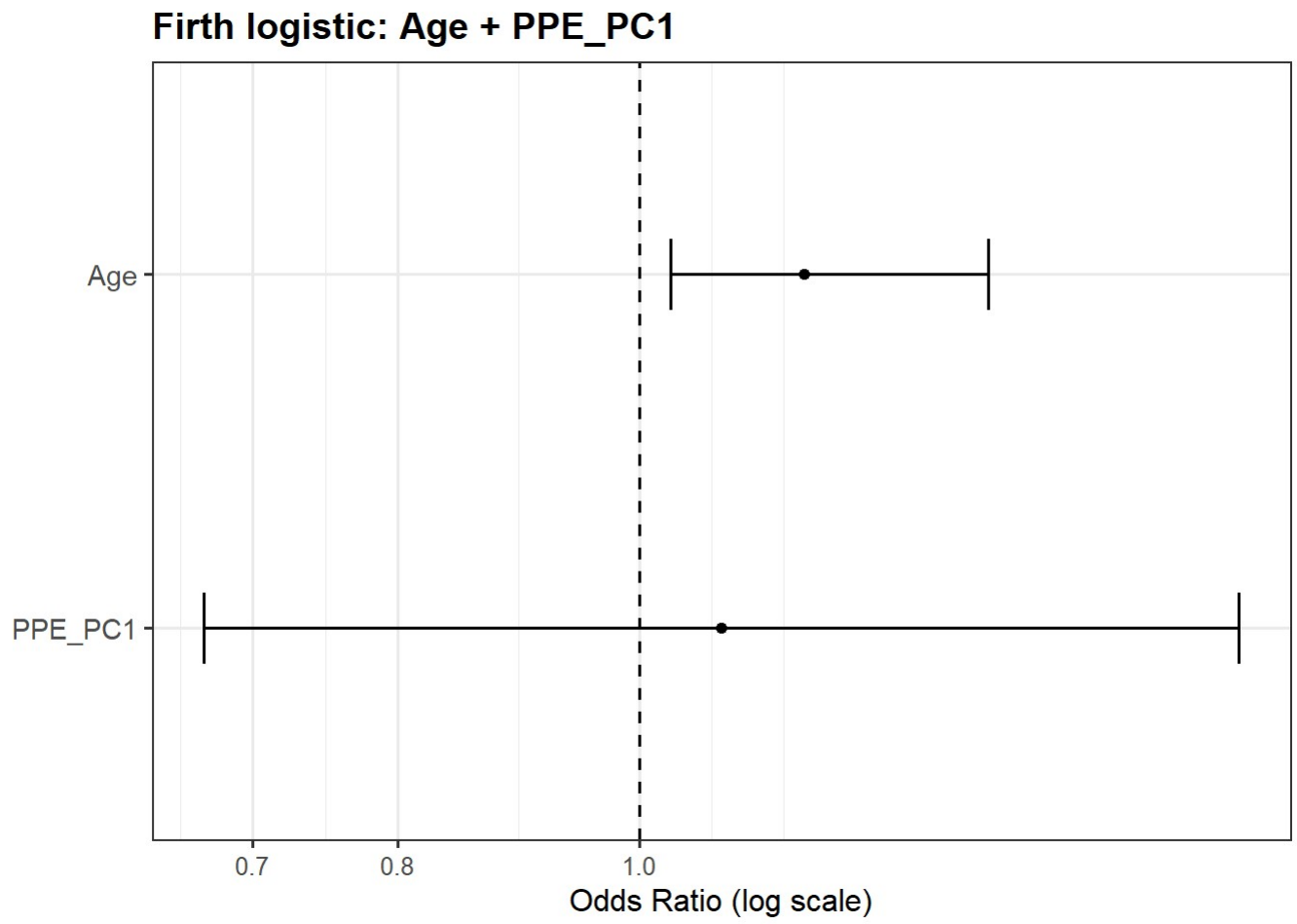
```
`height` was translated to `width`.
```





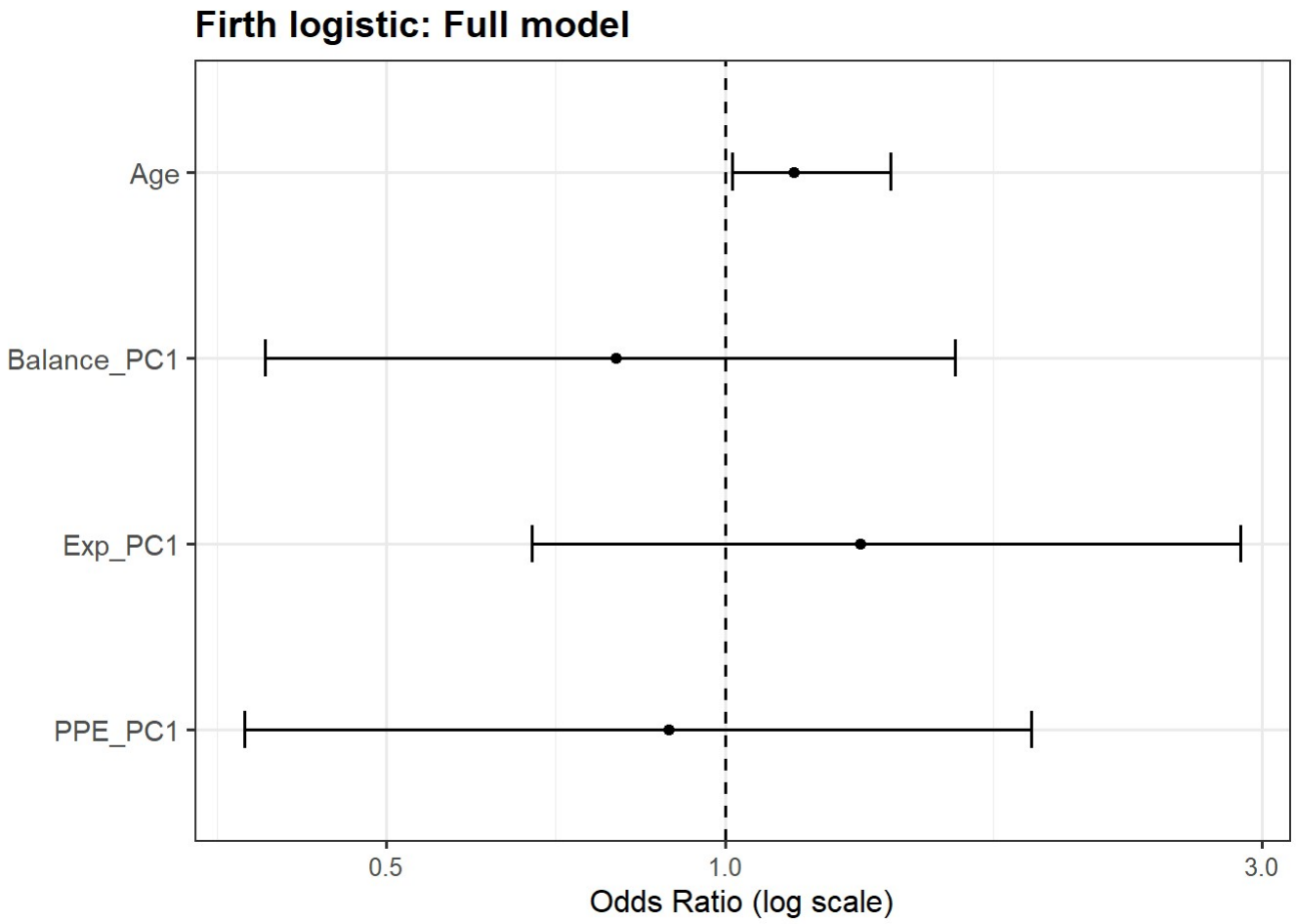

```
p_ppe
```

```
`height` was translated to `width`.
```




```
p_all
```

```
`height` was translated to `width`.
```




```
#------------------------------
# Save high-res images (optional)
#------------------------------
ggsave("Firth_Forest_Balance.png", p_bal, width = 7, height = 4.5, dpi = 300)
```

```
`height` was translated to `width`.
```

```
ggsave("Firth_Forest_Exposure.png", p_exp, width = 7, height = 4.5, dpi = 300)
```

```
`height` was translated to `width`.
```

```
ggsave("Firth_Forest_PPE.png",     p_ppe, width = 7, height = 4.5, dpi = 300)
```

```
`height` was translated to `width`.
```

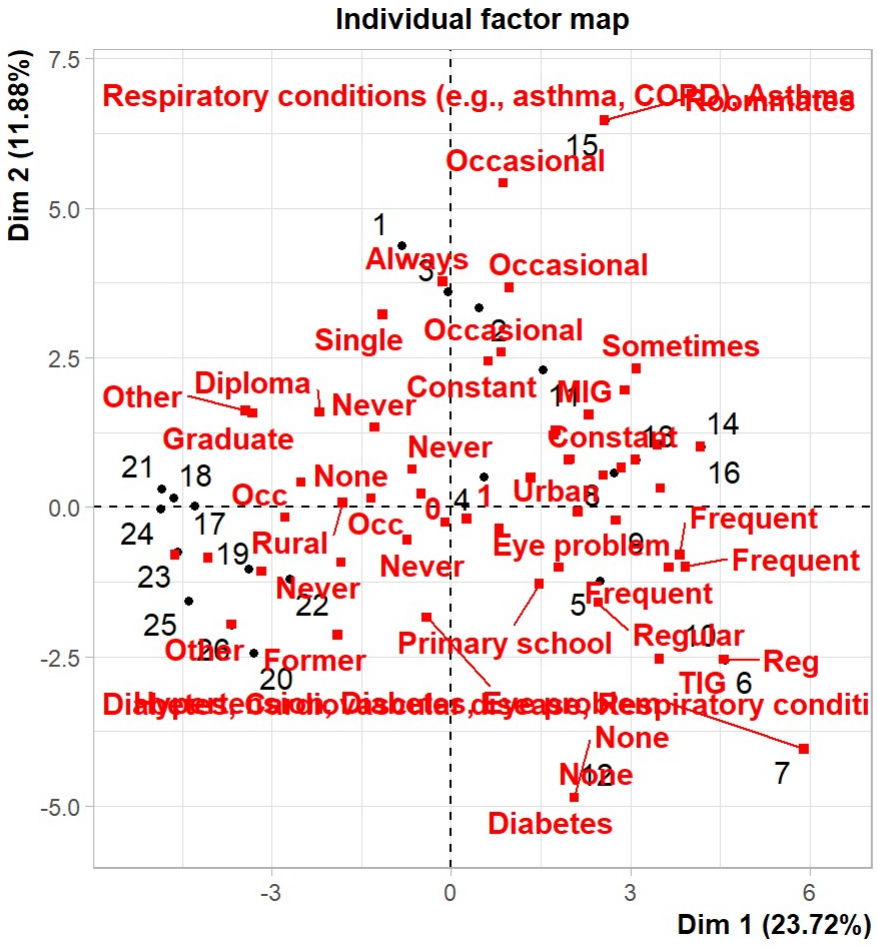
```
ggsave("Firth_Forest_Full.png",    p_all, width = 7, height = 5.0, dpi = 300)
```

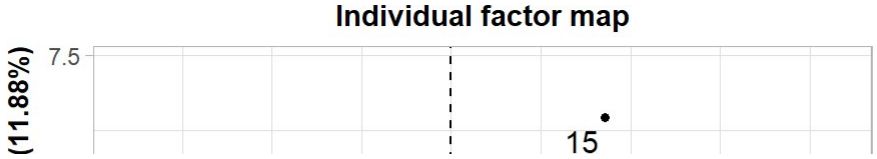
```
`height` was translated to `width`.
```

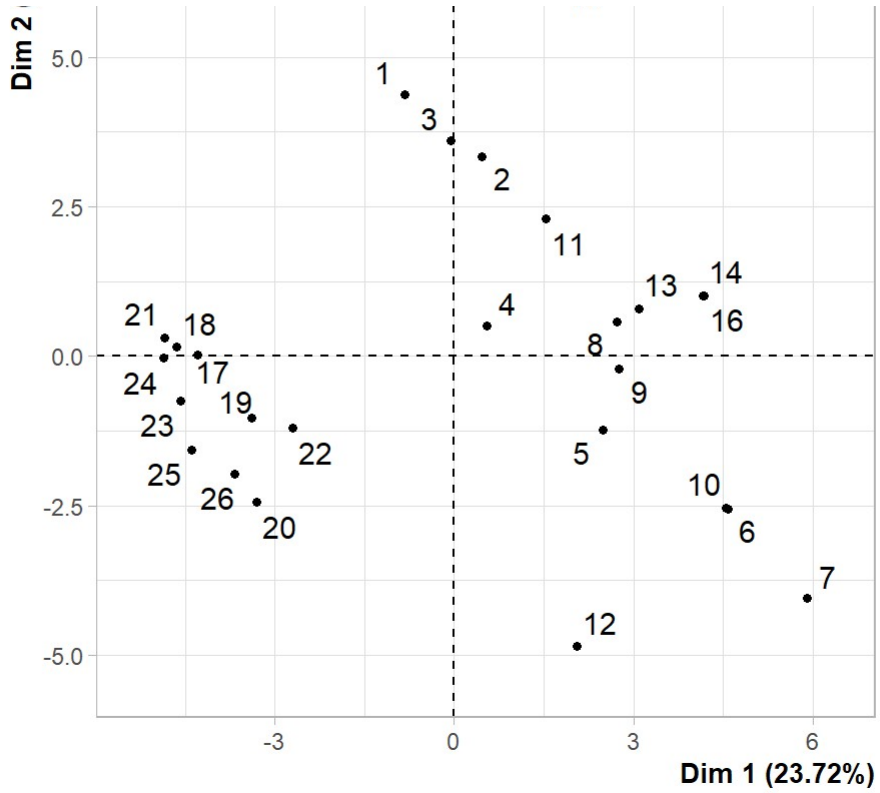
FAMD without indicator variables

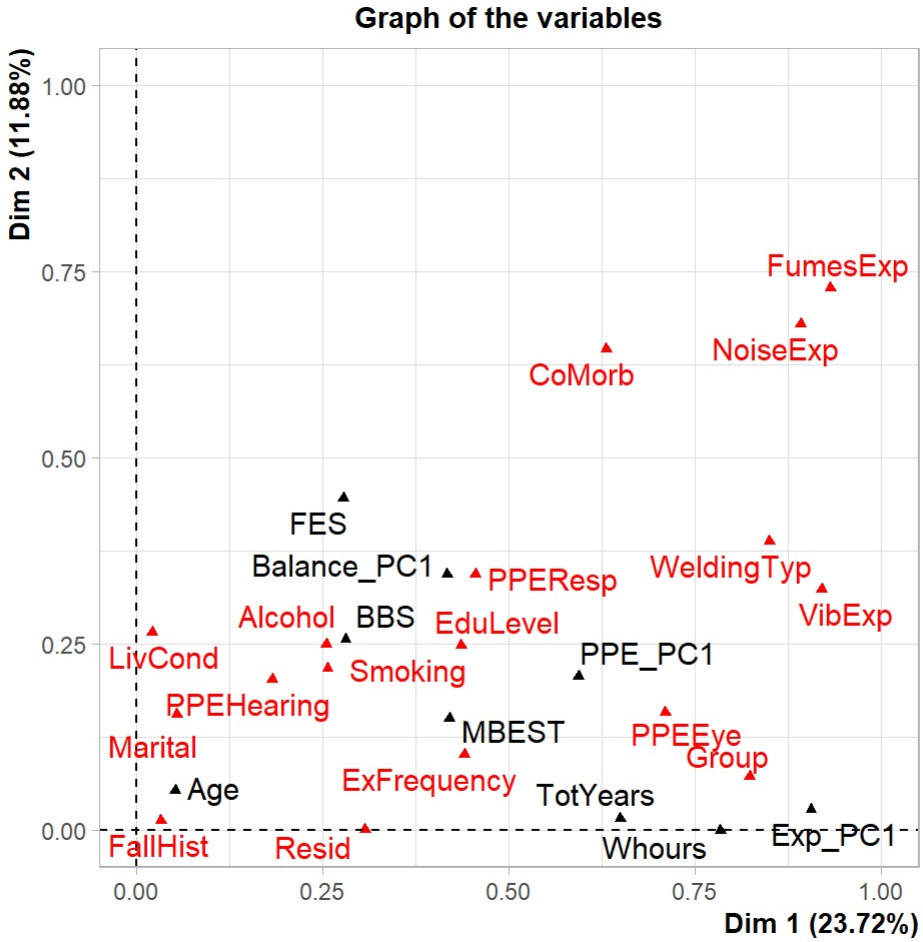
```
# Load libraries
library(FactoMineR)
library(factoextra)
library(dplyr)

# Prepare data: convert characters to factors, remove ID column if present
wdat_famd <- wdat %>%
  mutate_if(is.character, as.factor) %>%
  dplyr::select(-Id)  # remove Id column if it exists, adjust as needed

# Run FAMD (no graphs during run)
famd_result <- FAMD(wdat_famd, graph = TRUE)
```

```
Warning: ggrepel: 21 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

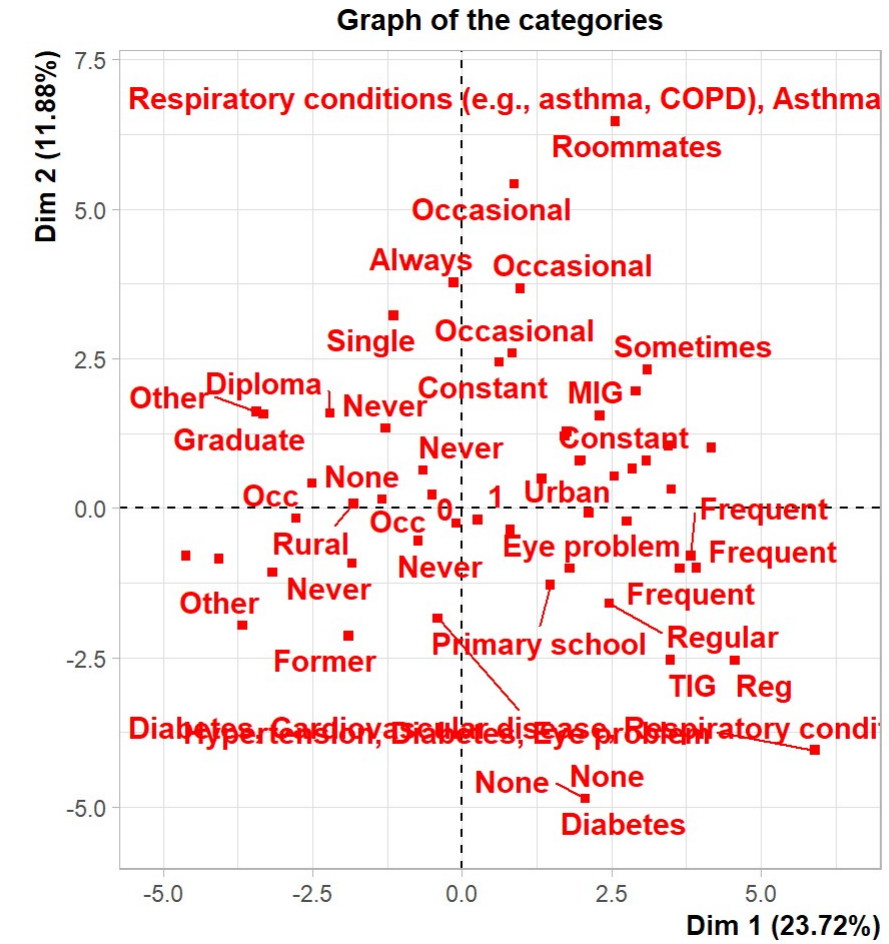


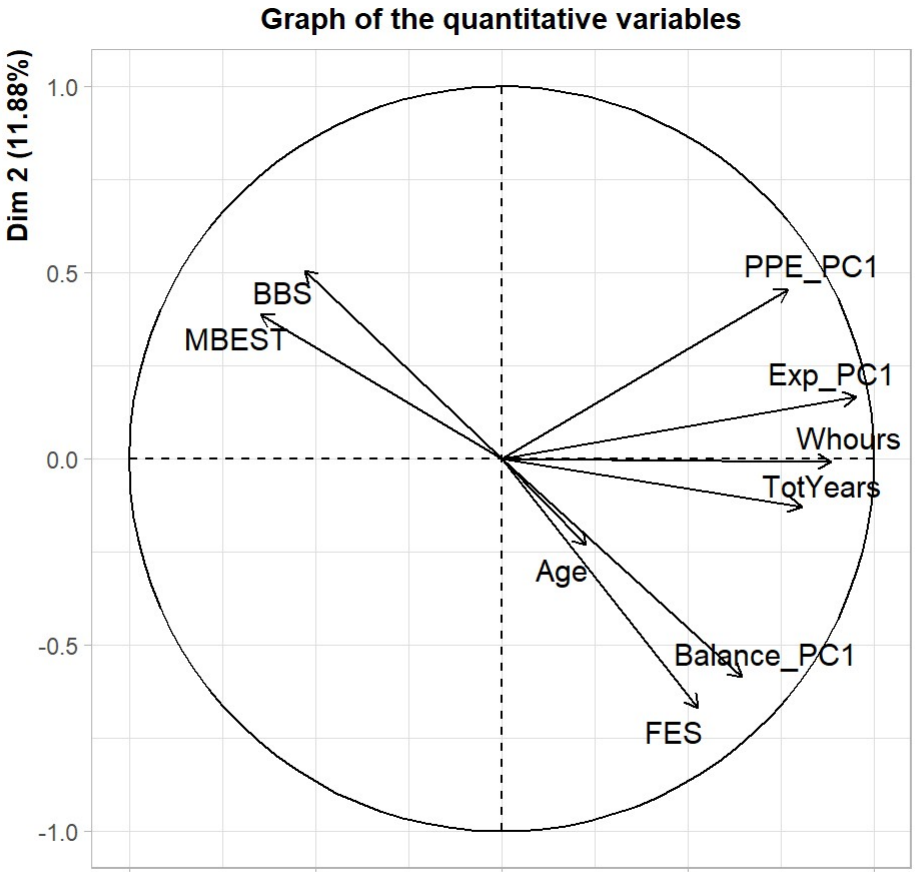


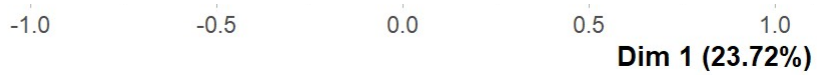




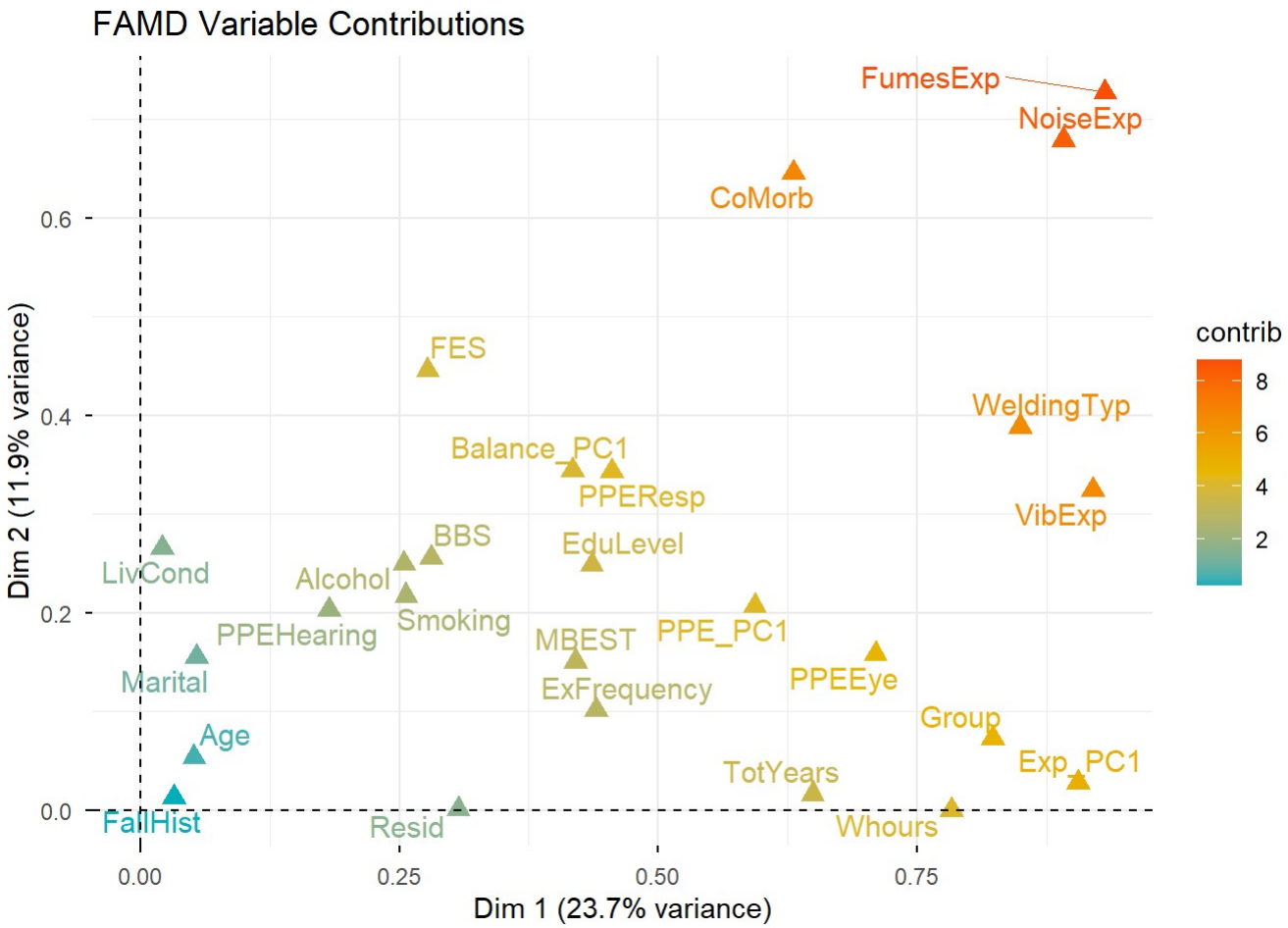

```
Warning: ggrepel: 21 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```


Graph of the categories
Dim 2 (11.88%)
Dim 1 (23.72%)
Respiratory conditions (e.g., asthma, COPD), Asthma
Roommates
Occasional
Always
Occasional
Single
Occasional
Sometimes
Other
Diploma
Never
Constant
MIG
Graduate
Never
Constant
None
Occ
0
1
Urban
Frequent
Occ
Rural
Eye problem
Frequent
Never
Never
Other
Frequent
Primary school
Regular
Former
TIG
Reg
Diabetes
Hypertension, Diabetes, Eye problem
None
None
Diabetes


Graph of the quantitative variables
Dim 2 (11.88%)
PPE_PC1
BBS
MBEST
Exp_PC1
Whours
TotYears
Age
Balance_PC1
FES




```r
# Visualize variable contributions on first two dimensions
fviz_famd_var(
  famd_result,
  repel = TRUE,                        # Avoid overlapping text
  col.var = "contrib",                 # Color by contribution
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), # Color gradient
  axes = c(1, 2),                      # Plot first two dimensions
  geom = c("point", "text"),           # Show points and labels
  pointsize = 3,
  ggtheme = theme_minimal()
) +
  labs(
    title = "FAMD Variable Contributions",
    x = paste0("Dim 1 (", round(famd_result$eig[1, 2], 1), "% variance)"),
    y = paste0("Dim 2 (", round(famd_result$eig[2, 2], 1), "% variance)")
  )
```




```r
# Extract contributions for Dim 1 and Dim 2
```

```
contrib_dim1 <- famd_result$var$contrib[, "Dim.1"]
contrib_dim2 <- famd_result$var$contrib[, "Dim.2"]

# Sort and display top contributors for Dim 1
cat("Top 10 contributors to Dimension 1:\n")
```

```
Top 10 contributors to Dimension 1:
```

```
print(sort(contrib_dim1, decreasing = TRUE)[1:10])
```

```
  FumesExp     VibExp    Exp_PC1   NoiseExp WeldingTyp      Group     Whours 
  7.408800   7.316871   7.201777   7.091131   6.757387   6.547308   6.231721 
    PPEEye   TotYears      CoMorb 
  5.646188   5.163387   5.015344 
```

```
# Sort and display top contributors for Dim 2
cat("\nTop 10 contributors to Dimension 2:\n")
```

```

Top 10 contributors to Dimension 2:
```

```
print(sort(contrib_dim2, decreasing = TRUE)[1:10])
```

```
   FumesExp    NoiseExp      CoMorb         FES  WeldingTyp Balance_PC1 
  11.562493  10.789529  10.259048   7.087828   6.172266   5.466447 
    PPEResp      VibExp     LivCond         BBS 
   5.457096   5.145731   4.221741   4.065304 
```

```
# Extract contributions for first 3 dimensions
var_contrib <- famd_result$var$contrib[, 1:2]

# Function to get top n variables for a dimension
top_n_vars <- function(contrib_vector, n = 10) {
  sorted <- sort(contrib_vector, decreasing = TRUE)
  head(sorted, n)
}

# Get top 5 variables per dimension
top10_dim1 <- top_n_vars(var_contrib[, "Dim.1"], 10)
top10_dim2 <- top_n_vars(var_contrib[, "Dim.2"], 10)
#top5_dim3 <- top_n_vars(var_contrib[, "Dim.3"], 10)

# Print results
list(Top10_Dim1 = top10_dim1,Top10_Dim2 = top10_dim2)
```

```
$Top10_Dim1
  FumesExp     VibExp    Exp_PC1   NoiseExp WeldingTyp      Group     Whours 
  7.408800   7.316871   7.201777   7.091131   6.757387   6.547308   6.231721 
```

```
   PPEEye   TotYears     CoMorb
 5.646188   5.163387   5.015344

$Top10_Dim2
   FumesExp    NoiseExp      CoMorb         FES  WeldingTyp Balance_PC1
 11.562493   10.789529   10.259048    7.087828    6.172266    5.466447
   PPEResp      VibExp     LivCond         BBS
  5.457096    5.145731    4.221741    4.065304
```