

Comobidities of Iraqi children with ASD: Reproducible Report

Contents

1	Settings	2
1.1	Clear working environment	2
1.2	Set a random seed	2
1.3	Required packages	2
1.4	Import and inspect data attributes	3
2	Difference in sample characteristics across healthcare centers	4
2.1	Define Variables and result objects	4
2.2	Epsilon-squared effect size for continuous characteristics	5
2.3	Cramer's V effect size for categorical characteristics	9
3	Difference in BMI across potentially-associated factors	14
3.1	Define Variables and result objects	14
3.2	Wilixicon R effect size for binary grouping variables	15
3.3	Epsilon-squared effect size for multi-level grouping variables	20

1 Settings

1.1 Clear working environment

```
rm(list = ls())
```

1.2 Set a random seed

```
set.seed(123)
```

1.3 Required packages

rcompanion : Compute effect sizes following nonparametric tests

ggplot2 : Create jitter and median plots

scales : Format percents

```
packages <- c("rcompanion", "ggplot2", "scales")

for (package in packages) {

  if (!requireNamespace(package, quietly = TRUE))
    install.packages(package)

  suppressPackageStartupMessages(library(package, character.only = TRUE))

}
```

```
## Warning: package 'rcompanion' was built under R version 4.4.3
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'scales' was built under R version 4.4.3
```

1.4 Import and inspect data attributes

```
df <- read.csv("ASD_comorbidity_dataset.csv")  
  
attributes(df)[names(attributes(df)) != "row.names"]
```

```
## $names  
## [1] "ID" "source" "age"  
## [4] "gender" "dx_age" "asd_fhx"  
## [7] "epilepsy" "sleep_disorder" "sleep_duration"  
## [10] "bmi" "calorie" "chronic_condition"  
## [13] "family_hx" "daytime_activity" "family_lifestyle"  
## [16] "daily_screentime" "excessive_screentime" "resperidone"  
##  
## $class  
## [1] "data.frame"
```

2 Difference in sample characteristics across healthcare centers

2.1 Define Variables and result objects

```
#Specify continuous and grouping variables
scales  <- c("age", "dx_age", "bmi")
centre  <- "source"
cat_vars <- c("gender", "asd_fhx", "epilepsy", "sleep_disorder")
```

```
#Prepare an empty data frame for Kruskal-Wallis test results
results_char1 <- data.frame(scale      = character(),
                             H_value   = numeric(),
                             df        = integer(),
                             p_value   = numeric(),
                             eps_sq    = numeric(),
                             CI_lower_95 = numeric(),
                             CI_upper_95 = numeric(),
                             stringsAsFactors = FALSE
                           )
```

```
#Prepare an empty data frame for chi-square test results
results_char2 <- data.frame(variable    = character(),
                             chisq_stat = numeric(),
                             df         = integer(),
                             p_value    = numeric(),
                             cramers_V  = numeric(),
                             CI_lower_95 = numeric(),
                             CI_upper_95 = numeric(),
                             stringsAsFactors = FALSE
                           )
```

2.2 Epsilon-squared effect size for continuous characteristics

```
#Loop over continuous variables
for (y in scales){

  # Subset and clean the data
  sub      <- df[, c(y, centre)]
  names(sub) <- c("y", "g")
  sub      <- na.omit(sub)
  sub$y    <- as.numeric(sub$y)
  sub$g    <- factor(sub$g, levels = c("al_subtain", "al_husseini", "baghdad"))

  # Kruskal-Wallis test
  kw <- kruskal.test(y ~ g, data = sub)

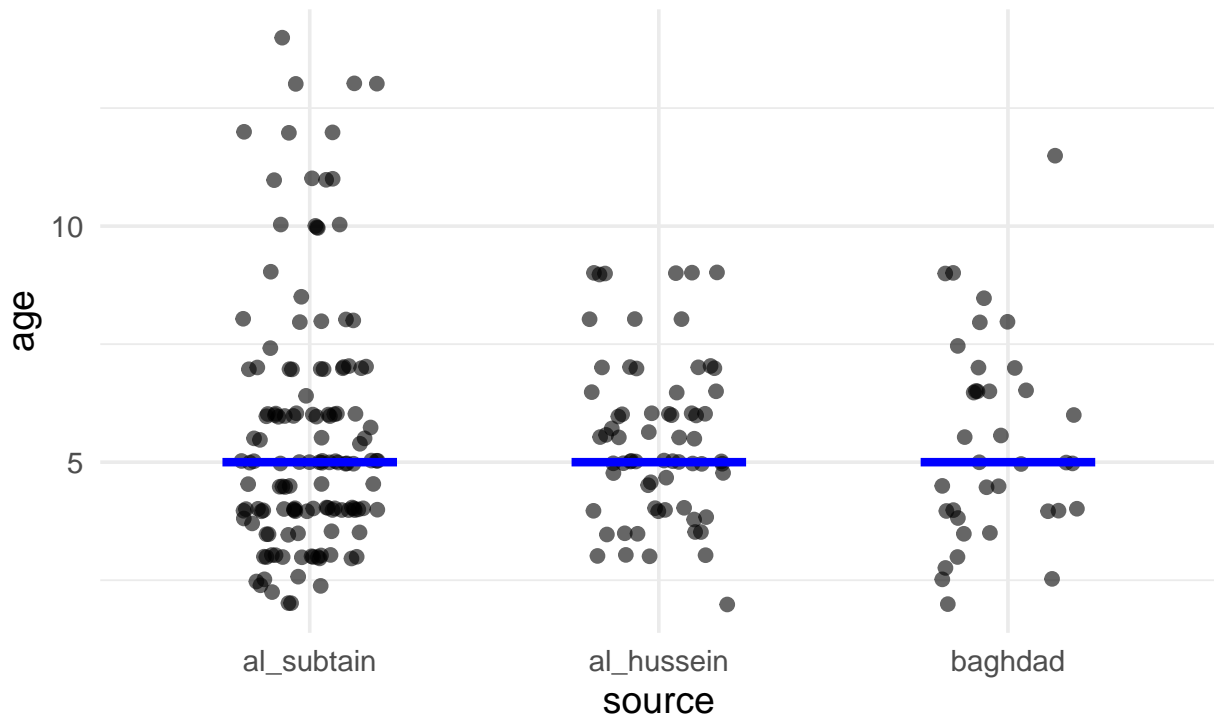
  # Epsilon-squared effect size:
  eps <- epsilonSquared(x      = sub$y,
                        g      = sub$g,
                        ci      = TRUE,
                        conf.level = 0.95,
                        type     = "bca",
                        R        = 10000
                        )

  # Jitter and median crossbar plot
  p <- ggplot(sub, aes(x = g, y = y)) +
    geom_jitter(width = 0.2, alpha = 0.6) +
    stat_summary(fun = median,
                 geom = "crossbar",
                 width = 0.5,
                 color = "blue",
                 linewidth = 0.6
                 ) +
    labs(title = paste0("Scatter of ", y, " by ", centre),
          subtitle = paste0("Epsilon-squared = ", round(eps, 3)),
          x = centre,
          y = y
          ) +
    theme_minimal(base_size = 14)
  print(p)

  # Append results
  results_char1 <- rbind(results_char1,
                         data.frame(scale = y,
                                    H_value = as.numeric(kw$statistic),
                                    df = kw$parameter,
                                    p_value = kw$p.value,
                                    eps_sq = eps,
                                    stringsAsFactors = FALSE
                                    )
                         )
}
```

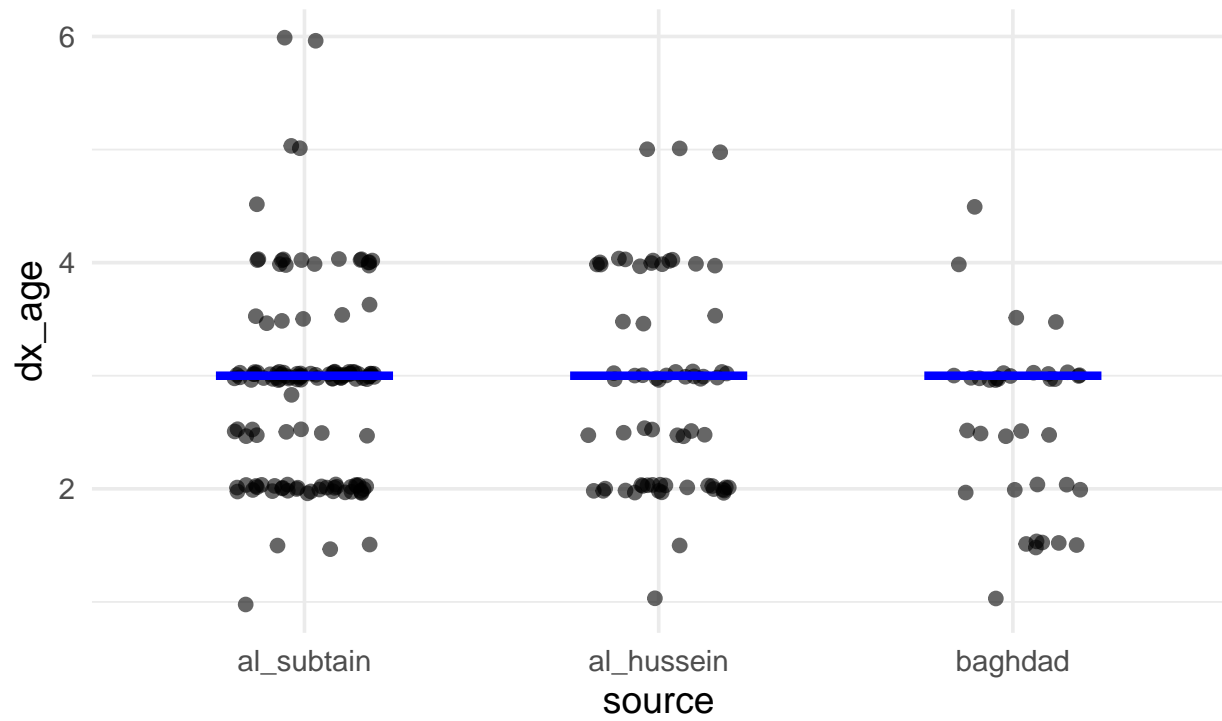
Scatter of age by source

Epsilon-squared = 0.004



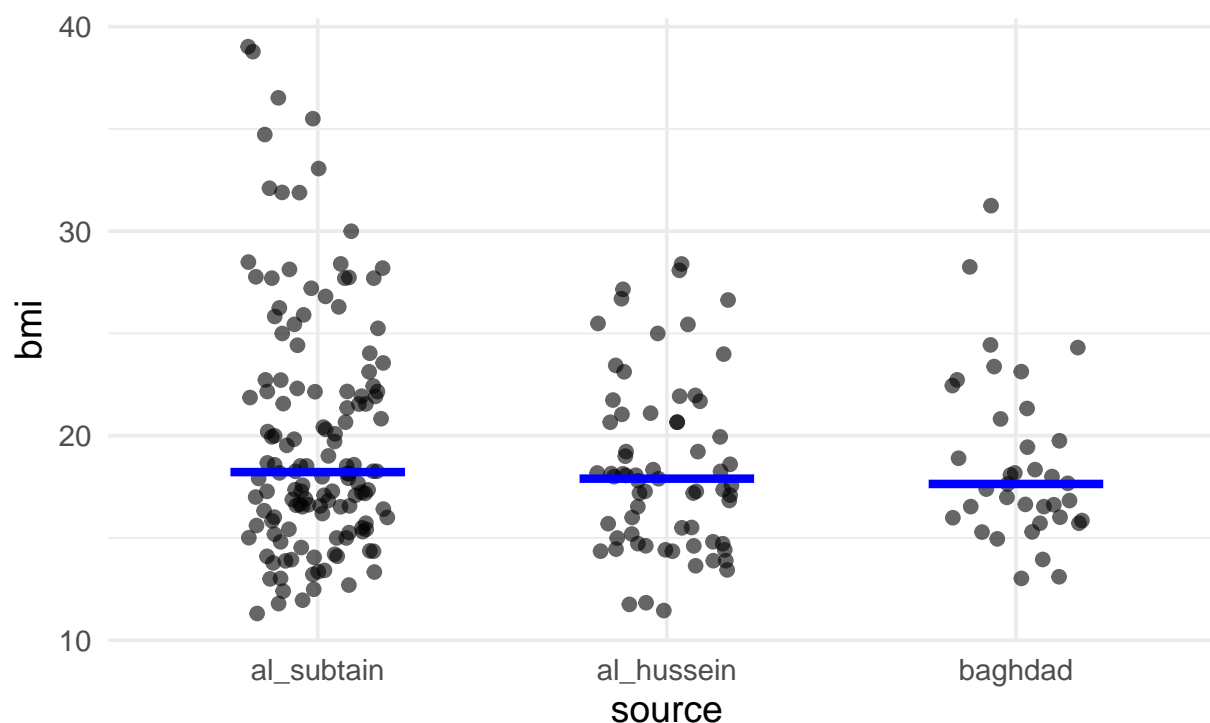
Scatter of dx_age by source

Epsilon-squared = 0.011



Scatter of bmi by source

Epsilon-squared = 0.008



```
# Print results
print(results_char1)
```

```
##      scale  H_value df    p_value eps_sq.epsilon.squared eps_sq.lower.ci
## df      age 0.8462549 2 0.6549952          0.00354      3.22e-06
## df1 dx_age 2.5953899 2 0.2731607          0.01090      7.97e-05
## df2  bmi 1.9653671 2 0.3743053          0.00822      6.91e-05
##      eps_sq.upper.ci
## df              0.0146
## df1             0.0390
## df2             0.0335
```


2.3 Cramer's V effect size for categorical characteristics

```
# Loop over categorical variables
for (v in cat_vars) {

  # Subset and clean the data
  sub <- df[, c(v, centre)]
  names(sub) <- c("y", "g")
  sub <- na.omit(sub)
  sub$y <- factor(sub$y)
  sub$g <- factor(sub$g, levels = c("al_subtain", "al_hussein", "baghdad"))

  tbl <- table(sub$y, sub$g)

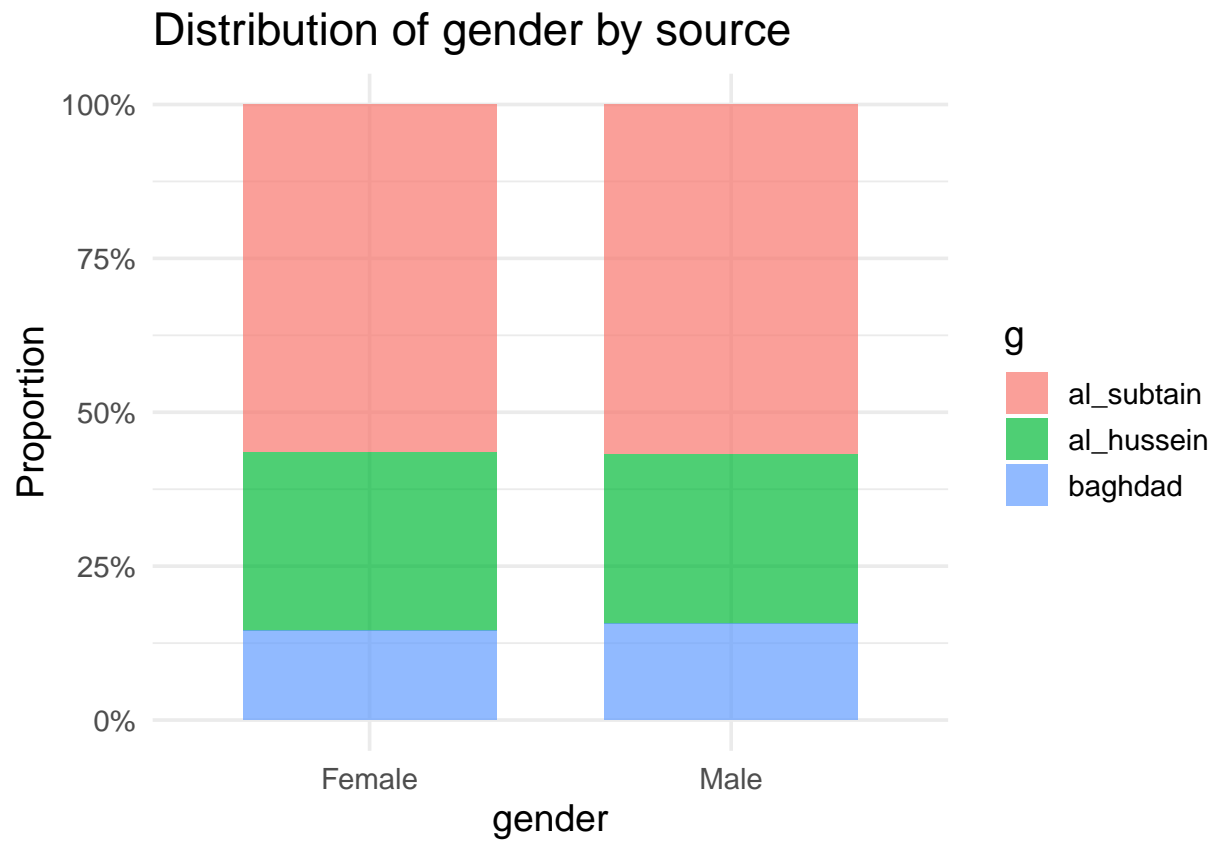
  # Chi-square test
  chi <- chisq.test(tbl, correct = FALSE)

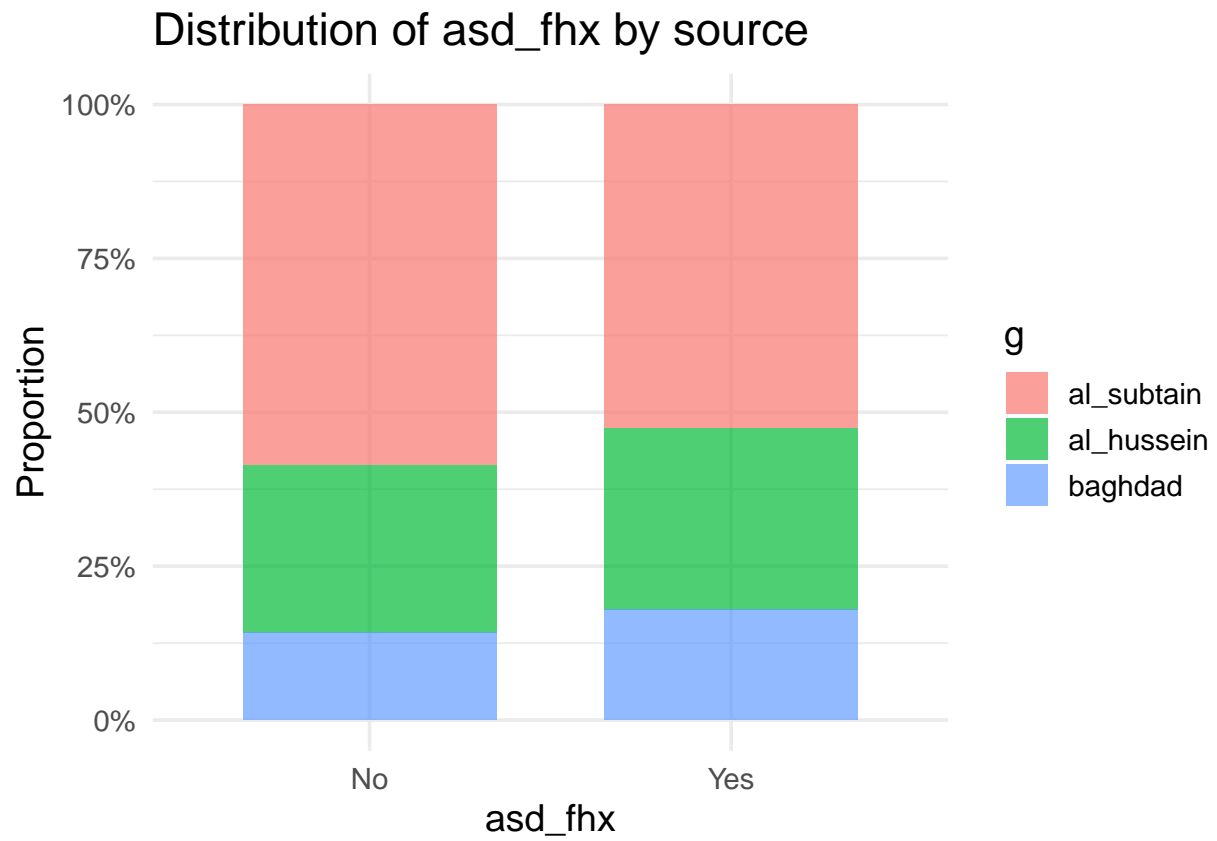
  # Cramer's V effect size
  cv <- cramerV(tbl,
                ci = TRUE,
                conf = 0.95,
                type = "bca",
                R = 10000
  )

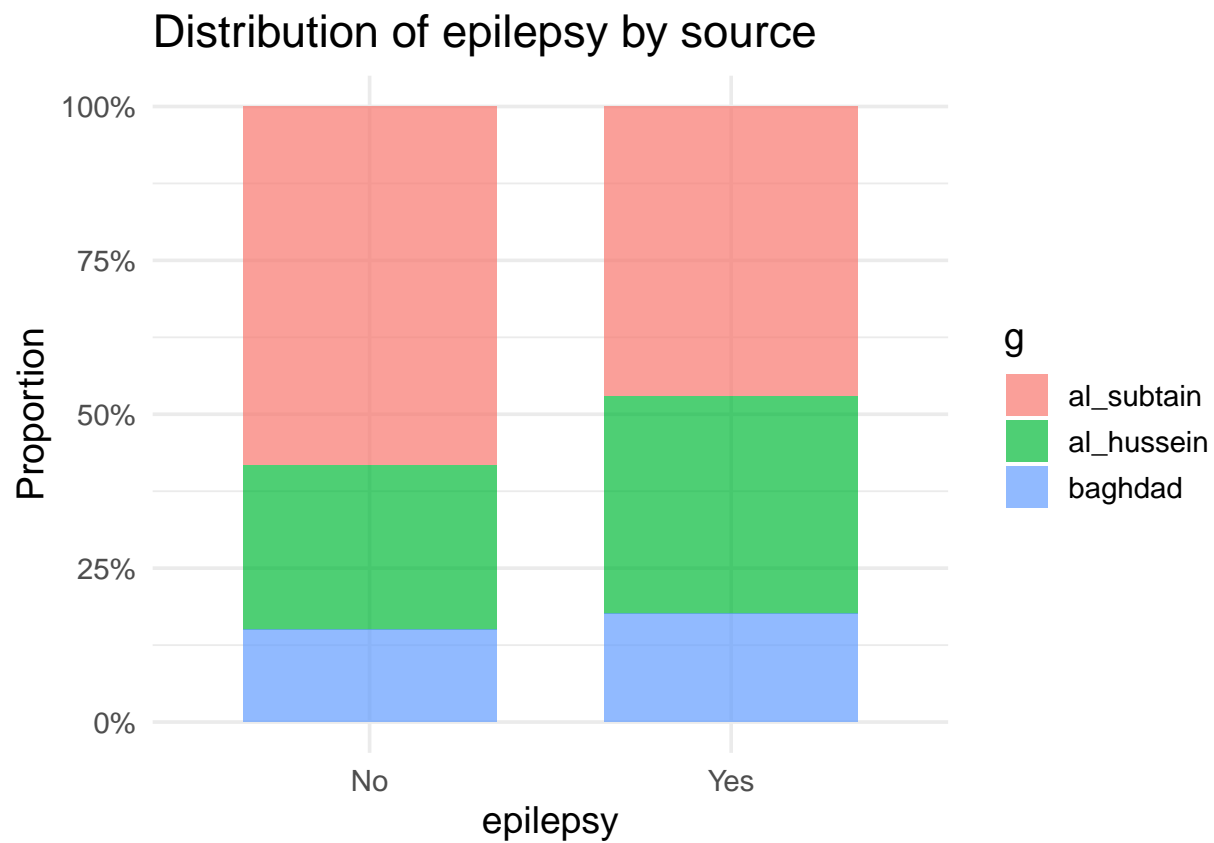
  # Stacked bar chart
  p <- ggplot(sub, aes(x = y, fill = g)) +
    geom_bar(position = "fill", width = 0.7, alpha = 0.7) +
    scale_y_continuous(labels = percent_format()) +
    labs(title = sprintf("Distribution of %s by %s", v, centre),
         x = v,
         y = "Proportion"
    ) +
    theme_minimal(base_size = 14)
  print(p)

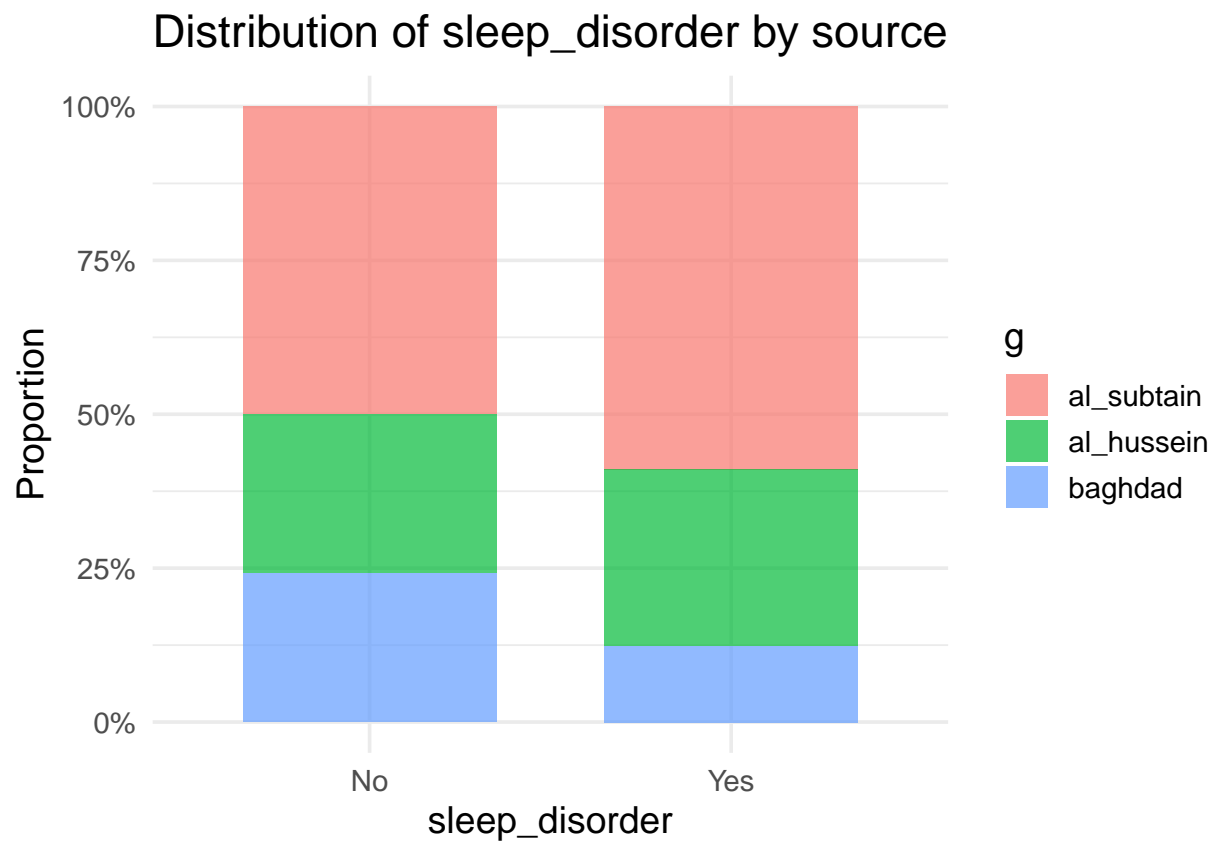
  # Append results
  results_char2 <- rbind(results_char2,
                        data.frame(variable = v,
                                   chisq_stat = as.numeric(chi$statistic),
                                   df = as.integer(chi$parameter),
                                   p_value = chi$p.value,
                                   crammers_V = cv,
                                   stringsAsFactors = FALSE
                        )
  )
}
```

```
## Warning in norm.inter(t, adj.alpha): extreme order statistics used as endpoints
```









```
# Print results
print(results_char2)
```

```
##      variable chisq_stat df    p_value cramers_V.Cramer.V cramers_V.lower.ci
## 1      gender 0.08192373  2 0.95986573      0.01848      0.000000
## 2      asd_fhx 0.92587489  2 0.62943201      0.06211      0.003493
## 3      epilepsy 1.54537230  2 0.46177101      0.08024      0.005074
## 4 sleep_disorder 4.96607621  2 0.08348919      0.14380      0.025680
## cramers_V.upper.ci
## 1      0.0217
## 2      0.1321
## 3      0.1649
## 4      0.2693
```

3 Difference in BMI across potentially-associated factors

3.1 Define Variables and result objects

```
#Specify continuous and grouping variables
y_var      <- "bmi"
binary_groups <- c("chronic_condition", "family_hx", "excessive_screentime", "resperidone")
multi_groups <- c("calorie", "daytime_activity", "family_lifestyle")
```

```
#Prepare an empty data frame for Man-Whitney U test results
results_binary <- data.frame(group      = character(),
                             U_stat    = numeric(),
                             p_value    = numeric(),
                             r_value    = numeric(),
                             CI_lower_95 = numeric(),
                             CI_upper_95 = numeric(),
                             stringsAsFactors = FALSE
                             )
```

```
#Prepare an empty data frame for Kruskal-Wallis test results
results_multi <- data.frame(group      = character(),
                             H_value    = numeric(),
                             df         = integer(),
                             p_value    = numeric(),
                             eps_sq     = numeric(),
                             CI_lower_95 = numeric(),
                             CI_upper_95 = numeric(),
                             stringsAsFactors = FALSE
                             )
```

3.2 Wilcoxon R effect size for binary grouping variables

```
# Loop over binary grouping variables
for (g in binary_groups) {

  # Subset and clean the data
  sub <- df[, c(y_var, g)]
  names(sub) <- c("y", "g")
  sub <- na.omit(sub)
  sub$g <- factor(sub$g, levels = c("No", "Yes"))

  # Mann-Whitney test
  wt <- wilcox.test(y ~ g, data = sub, exact = FALSE)

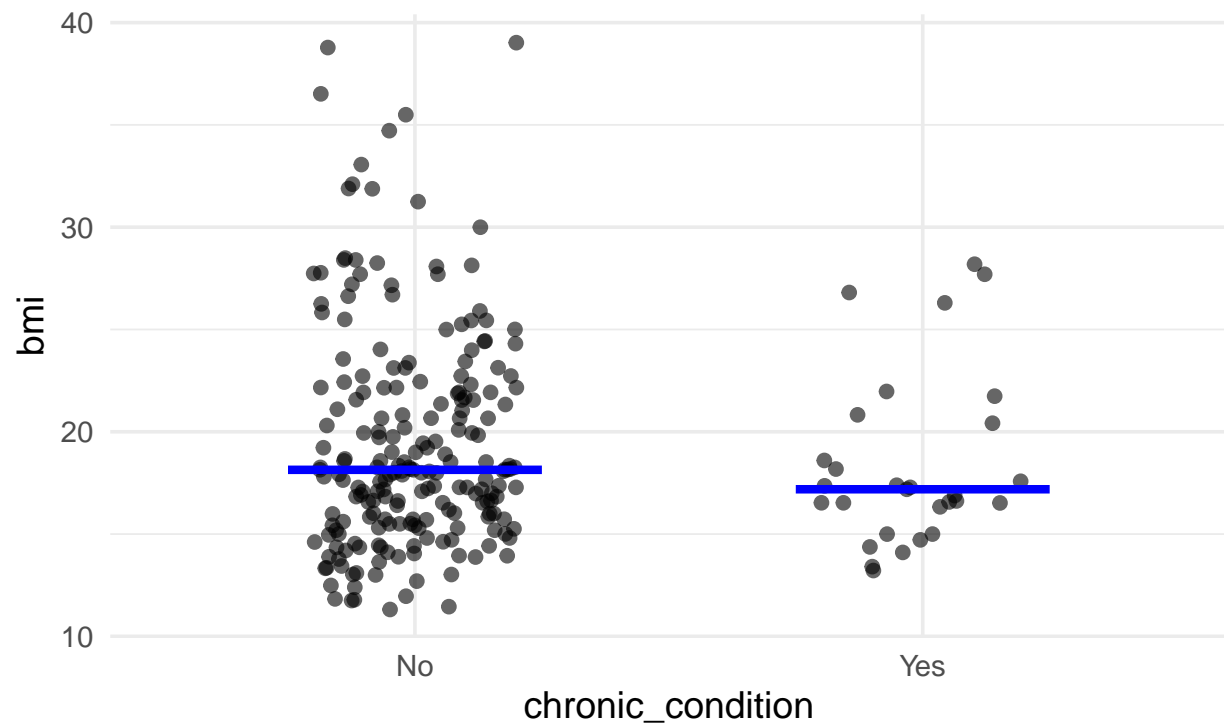
  # R-values effect size
  r <- -1*wilcoxonR(x      = sub$y,
                    g      = sub$g,
                    ci      = TRUE,
                    conf.level = 0.95,
                    type     = "bca",
                    R        = 10000
                    )
  r[c(2, 3)] <- r[c(3, 2)]

  # Jitter and median crossbar plot
  p <- ggplot(sub, aes(x = g, y = y)) +
    geom_jitter(width = 0.2, alpha = 0.6) +
    stat_summary(fun = median,
                 geom = "crossbar",
                 width = 0.5,
                 color = "blue",
                 linewidth = 0.6
                 ) +
    labs(title = paste0("Scatter of ", y_var, " by ", g),
         subtitle = paste0("r = ", round(r, 3)),
         x = g,
         y = y_var
         ) +
    theme_minimal(base_size = 14)
  print(p)

  #Append result
  results_binary <- rbind(results_binary,
                          data.frame(group = g,
                                     U_stat = as.numeric(wt$statistic),
                                     p_value = wt$p.value,
                                     r_value = r,
                                     stringsAsFactors = FALSE
                          )
  )
}
```

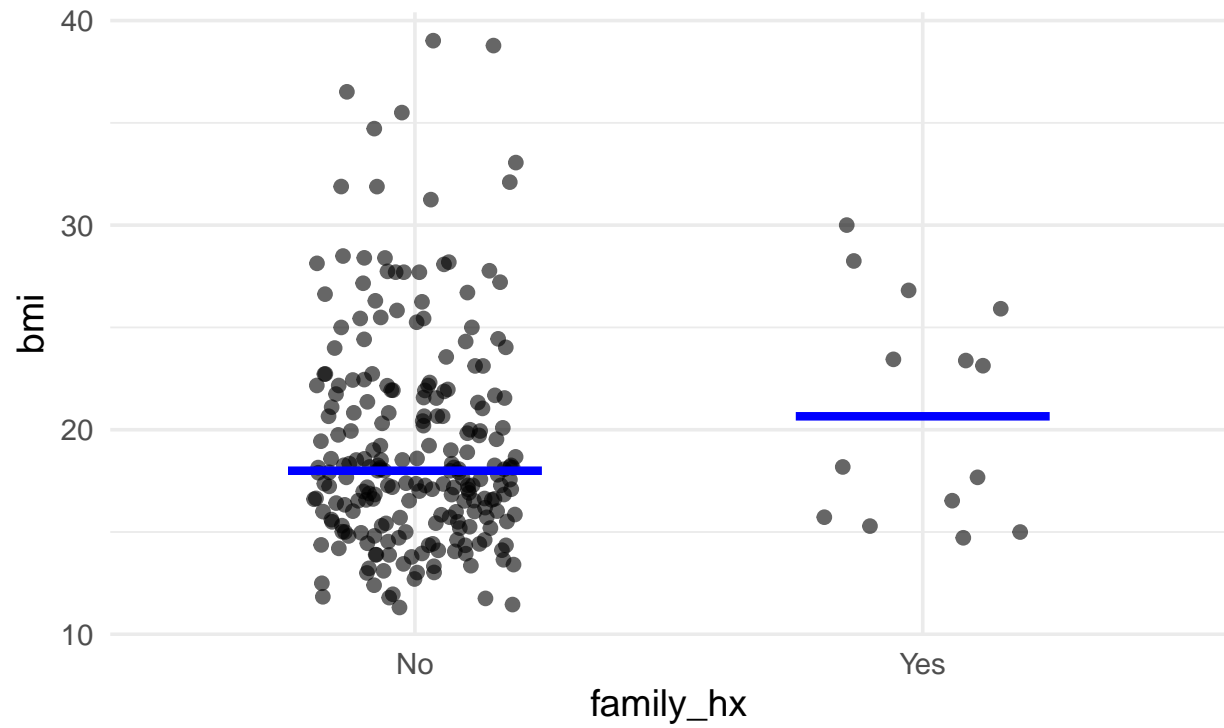
Scatter of bmi by chronic_condition

$r = -0.066$



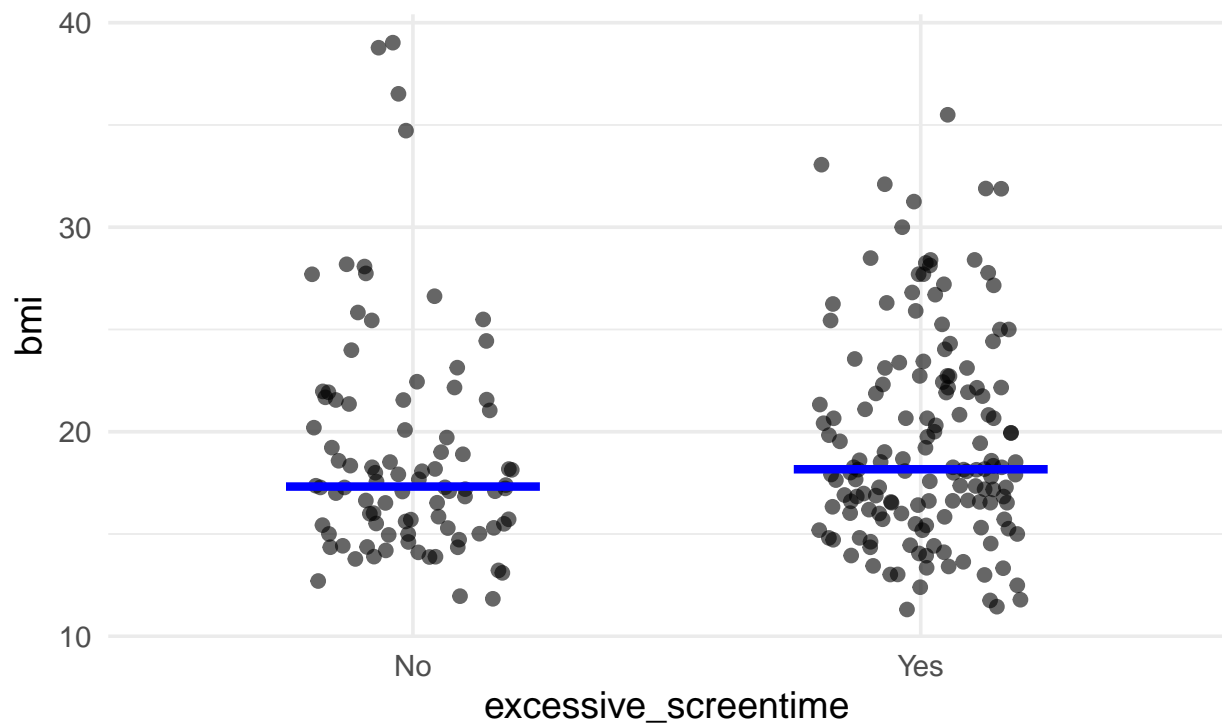
Scatter of bmi by family_hx

$r = 0.078$



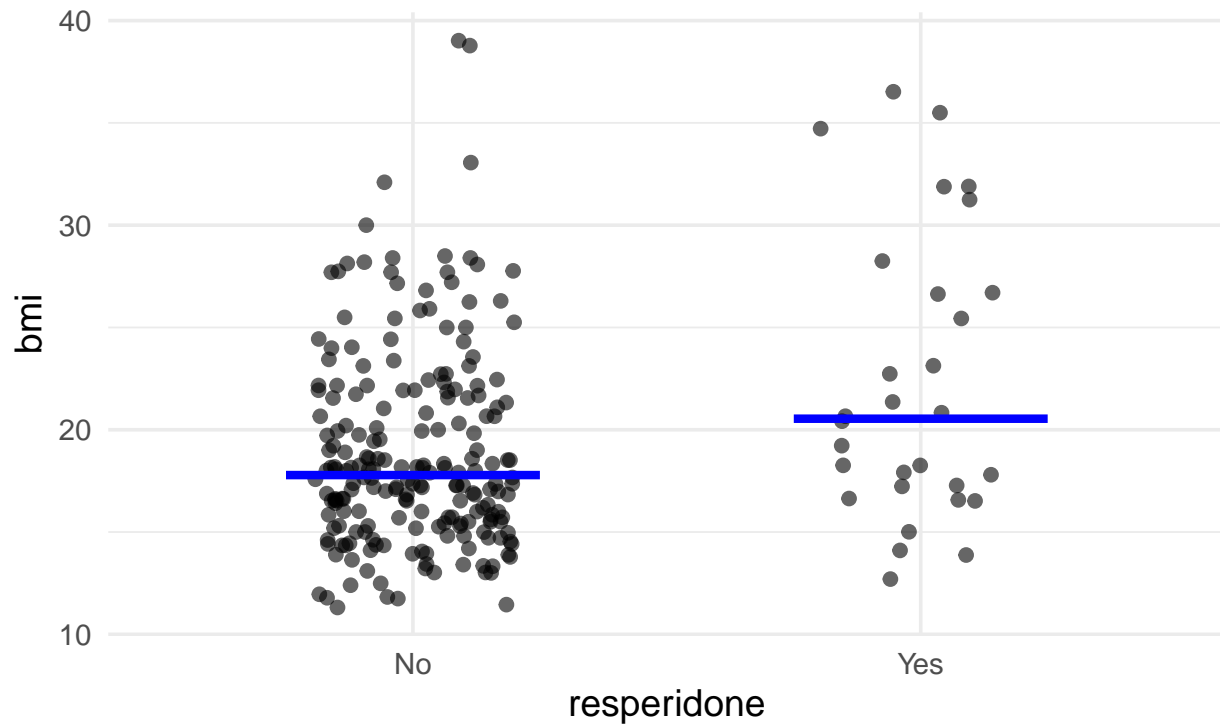
Scatter of bmi by excessive_screentime

$r = 0.087$



Scatter of bmi by resperidone

$r = 0.163$



```
# Print results
print(results_binary)
```

```
##           group U_stat  p_value r_value.r r_value.lower.ci
## 1 chronic_condition 3415.5 0.3105225 -0.0658      -0.1800
## 2      family_hx 1280.5 0.2324386  0.0775      -0.0566
## 3 excessive_screentime 6045.5 0.1763537  0.0871     -0.0422
## 4      resperidone 2254.0 0.0118155  0.1630       0.0308
## r_value.upper.ci
## 1          0.0553
## 2          0.2110
## 3          0.2110
## 4          0.2880
```

3.3 Epsilon-squared effect size for multi-level grouping variables

```
# Loop over non-binary grouping variables
for (g in multi_groups) {

  # Subset and clean the data
  sub      <- df[, c(y_var, g)]
  names(sub) <- c("y", "g")
  sub      <- na.omit(sub)
  sub$y    <- as.numeric(sub$y)
  sub$g    <- factor(sub$g, levels = c("Low", "Moderate", "High"))

  # Kruskal-Wallis test
  kw <- kruskal.test(y ~ g, data = sub)

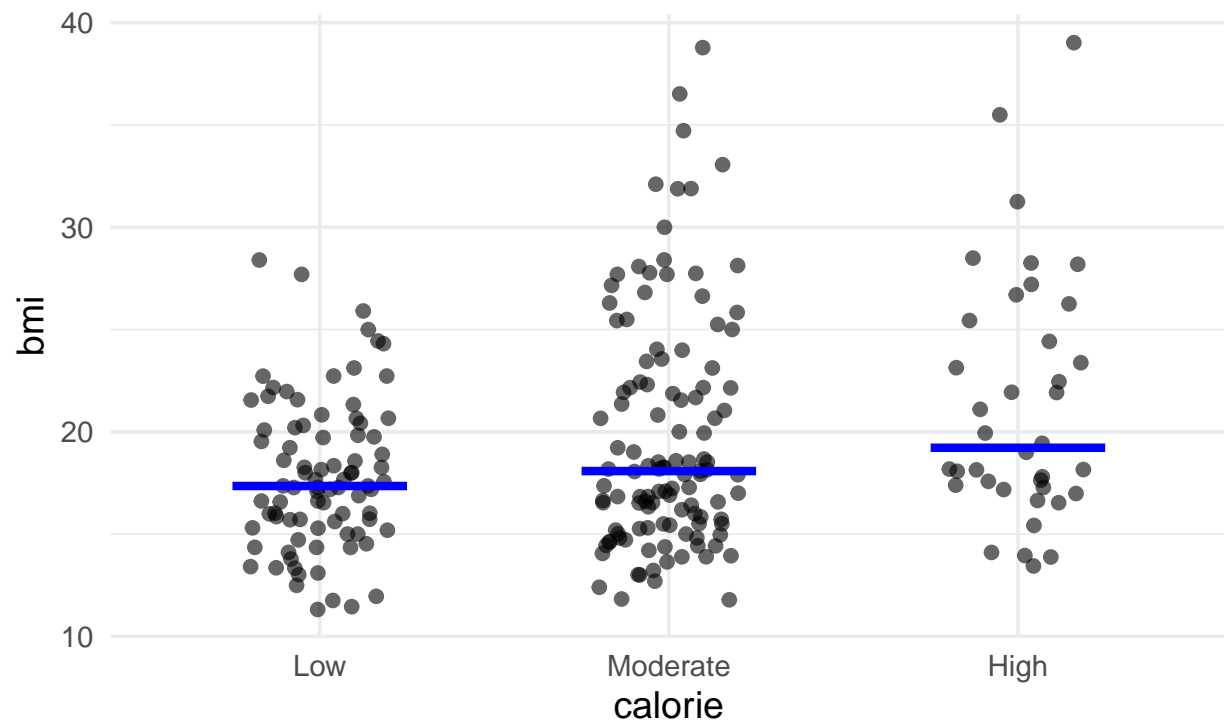
  # Epsilon-squared effect size:
  eps <- epsilonSquared(x      = sub$y,
                        g      = sub$g,
                        ci      = TRUE,
                        conf.level = 0.95,
                        type     = "bca",
                        R        = 10000
                        )

  # Jitter and median crossbar plot
  p <- ggplot(sub, aes(x = g, y = y)) +
    geom_jitter(width = 0.2, alpha = 0.6) +
    stat_summary(fun = median,
                 geom = "crossbar",
                 width = 0.5,
                 color = "blue",
                 linewidth = 0.6
                 ) +
    labs(title = paste0("Scatter of ", y_var, " by ", g),
          subtitle = paste0("Epsilon-squared = ", round(eps, 3)),
          x = g,
          y = y_var
          ) +
    theme_minimal(base_size = 14)
  print(p)

  # Append results
  results_multi <- rbind(results_multi,
                         data.frame(group = g,
                                    H_value = as.numeric(kw$statistic),
                                    df = kw$parameter,
                                    p_value = kw$p.value,
                                    eps_sq = eps,
                                    stringsAsFactors = FALSE
                                    )
                         )
}
```

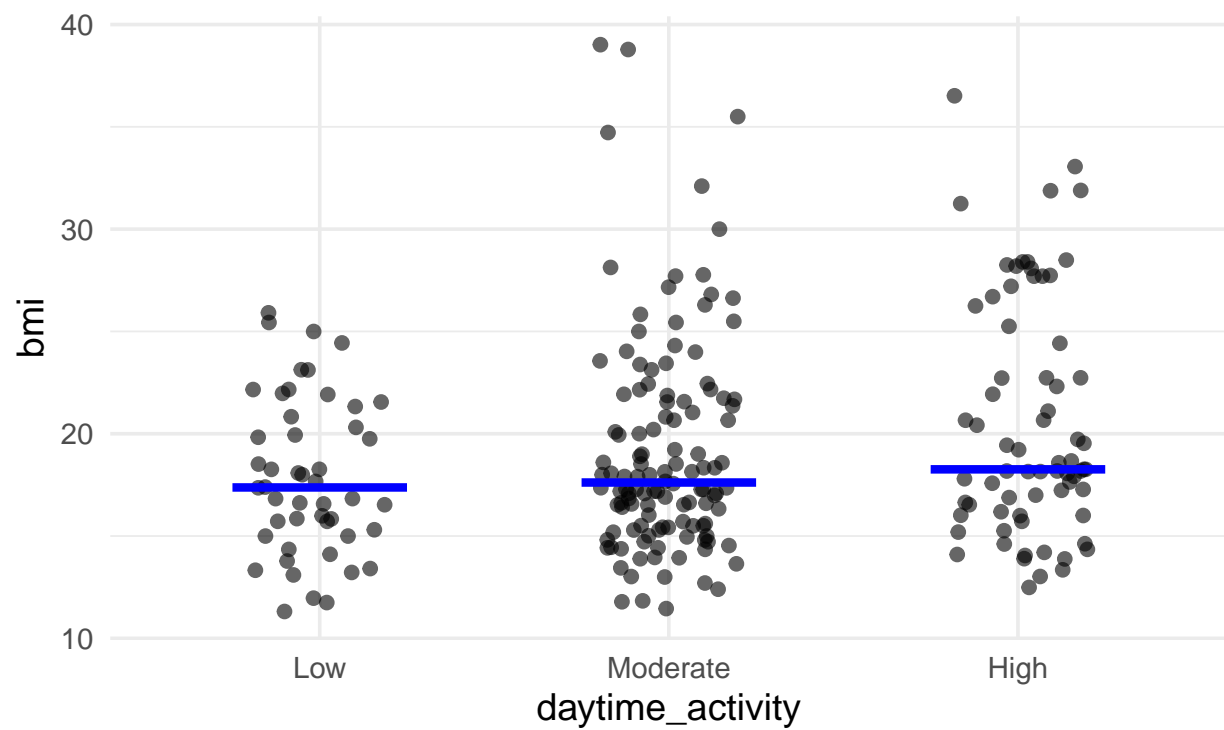
Scatter of bmi by calorie

Epsilon-squared = 0.038



Scatter of bmi by daytime_activity

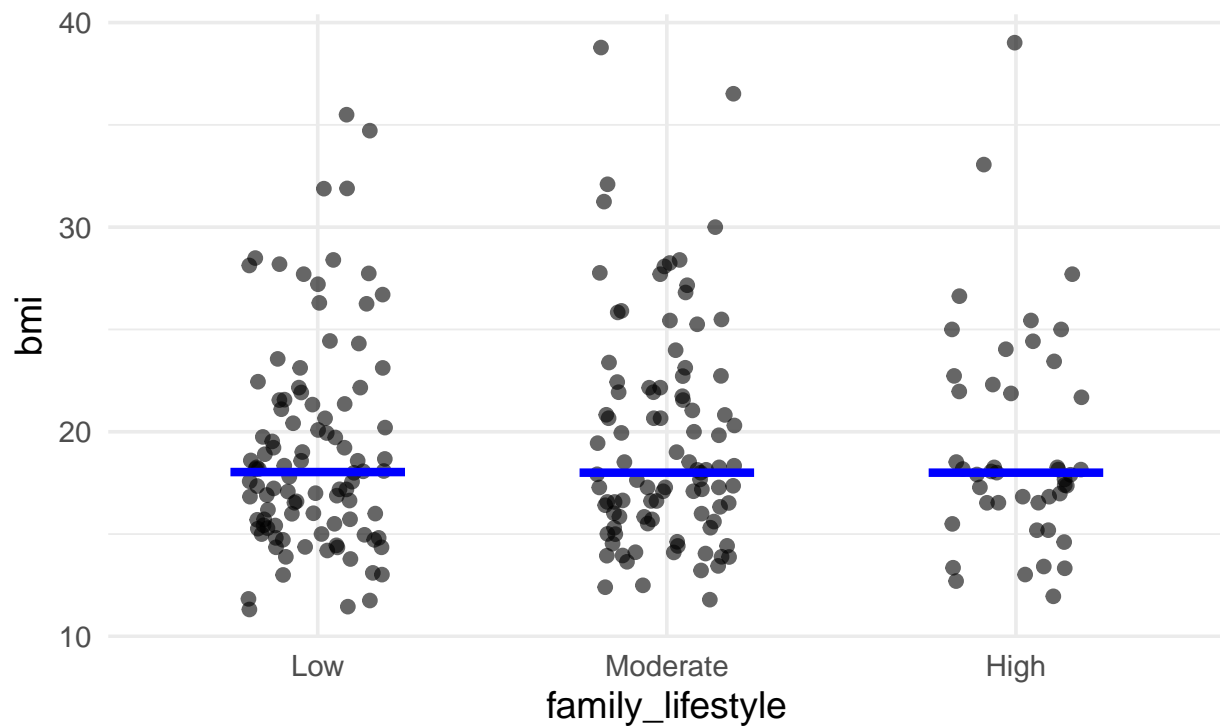
Epsilon-squared = 0.023



```
## Warning in norm.inter(t, adj.alpha): extreme order statistics used as endpoints
```

Scatter of bmi by family_lifestyle

Epsilon-squared = 0.001



```
# Print results
print(results_multi)
```

```
##           group  H_value df    p_value eps_sq.epsilon.squared
## df          calorie 8.9746725  2 0.01125057          0.03760
## df1 daytime_activity 5.4465608  2 0.06565901          0.02280
## df2 family_lifestyle 0.2686765  2 0.87429430          0.00112
##   eps_sq.lower.ci eps_sq.upper.ci
## df          5.25e-03          0.08720
## df1          9.60e-04          0.06800
## df2          2.94e-08          0.00362
```