

# Session 11 The Outliers

2025-12-12

## 1. Load library

```
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(stringr)
library(purrr)
```

## 2. Read in dataset

```
df_A <- read_excel("A.xlsx") %>% mutate(Source = "A")
df_B <- read_excel("B.xlsx") %>% mutate(Source = "B")
df_P <- read_excel("P.xlsx") %>% mutate(Source = "P")

df_SE <- read_excel("SouthEast.xlsx") %>%
  rename(PH = Phusical, MH = Mental) %>%
  mutate(Source = "SE") %>%
  select(any_of(c("Age", "Region", "PH", "MH", "Smoker", "Belief", "SES5", "Gender", "ID", "Source")),
  everything())

df_B1 <- read_excel("Book1.xlsx") %>%
  rename(PH = `Physical Health`, MH = `Mental Health`, ID = `No.`) %>%
  mutate(
    Age = suppressWarnings(as.numeric(Age)),
    ID = as.character(ID),
    Source = "B1"
  ) %>%
  select(any_of(c("ID", "Age", "Region", "PH", "MH", "Smoker", "Belief", "SES5", "Gender", "Source")),
  everything())
```

## 3. Bind datasets

```
df_total <- bind_rows(df_A, df_B, df_P, df_SE, df_B1)
```

```
to01 <- function(x) {  
  x <- str_to_lower(str_trim(as.character(x)))  
  case_when(  
    x %in% c("1","y","yes","true","t") ~ 1,  
    x %in% c("0","n","no","false","f") ~ 0,  
    TRUE ~ NA_real_  
  )  
}
```

```
df_model <- df_total %>%  
  mutate(  
    Belief = to01(Belief),  
    PH      = to01(PH),  
    MH      = to01(MH)  
  ) %>%  
  filter(!is.na(Belief), !is.na(PH), !is.na(MH))  
  
print(table(df_model$Belief, useNA = "ifany"))
```

```
##  
##    0    1  
## 288 137
```

```
print(table(df_model$PH, useNA = "ifany"))
```

```
##  
##    0    1  
## 347   78
```

```
print(table(df_model$MH, useNA = "ifany"))
```

```
##  
##    0    1  
## 288 137
```

```
if (nrow(df_model) == 0) stop("No complete cases after cleaning: check encodings in Belief/PH/  
MH.")  
if (length(unique(df_model$Belief)) < 2) stop("Belief has <2 classes after cleaning.")  
if (length(unique(df_model$PH)) < 2) stop("PH has <2 classes after cleaning.")  
if (length(unique(df_model$MH)) < 2) stop("MH has <2 classes after cleaning.")
```

#### 4. Primary – binomial regression

```
model <- glm(Belief ~ PH * MH, data = df_model, family = binomial(link = "logit"))  
summary(model)
```

```
##
## Call:
## glm(formula = Belief ~ PH * MH, family = binomial(link = "logit"),
##      data = df_model)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.53408     0.13661  -3.910 9.24e-05 ***
## PH          -0.34628     0.31918  -1.085  0.2780
## MH          -0.48628     0.25009  -1.944  0.0518 .
## PH:MH       -0.01966     0.67695  -0.029  0.9768
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 534.34  on 424  degrees of freedom
## Residual deviance: 528.49  on 421  degrees of freedom
## AIC: 536.49
##
## Number of Fisher Scoring iterations: 4
```