Diagnostics

Andrew

OutlierRemovalMethod	TransformationMethod	SW_Test	BF_Test
None	None	0	0.0009874
Boxplot	None	0	0.0000000
Studentized	None	0	0.0009874
None	PPCC	0	0.0000000
Boxplot	PPCC	0	0.0000000
Studentized	PPCC	0	0.0000000
None	Shapiro-Wilk	0	0.0000000
Boxplot	Shapiro-Wilk	0	0.0000000
Studentized	Shapiro-Wilk	0	0.0000000
None	Log-Likelihood	0	0.0000000
Boxplot	Log-Likelihood	0	0.0000000
Studentized	Log-Likelihood	0	0.0000000

```r
#  Functions
sw_test <- function(data) {
  model <- aov(Y ~ A * B, data = data)
  return(shapiro.test(model$residuals)$p.value)
}

bf_test <- function(data) {
  model <- aov(Y ~ A * B, data = data)
  return(car::leveneTest(model$residuals ~ paste(data$A, data$B))[1, 3])
}

# Assuming data is TFA and has Y, A, B
do_transform <- function(objective, data) {
  if (objective == "None") {
    return(data)
  } else if (objective == "PPCC" || objective == "Shapiro-Wilk") {
    model <- aov(Y ~ A * B, data = data)
    L <- EnvStats::boxcox(model, objective.name = objective,
                          optimize = TRUE)$lambda
  } else if (objective == "Log-Likelihood") {
    L <- EnvStats::boxcox(data$Y, objective.name = objective,
                          optimize = TRUE)$lambda
  } else {
    warning("Invalid objective")
    return()
  }

  YT <- (data$Y ^ L - 1) / L
```

```r
  t.data <- data.frame(Y = YT, A = data$A, B = data$B)
  return(t.data)
}

# Assuming data is TFA and has Y, A, B
do_transform_and_model <- function(objective, data) {
  t.data <- do_transform(objective, data)
  t.model <- aov(Y ~ A * B, data = t.data)
  return(t.model)
}

# Assuming data is TFA and has Y, A, B
remove_outlier <- function(objective, data, alpha = 0.05) {
  data <- force(data)

  if (objective == "None") {
    return(data)
  }else if (objective == "Boxplot") {
    plot <- boxplot(Y ~ A * B, data = data, plot = FALSE)
    outliers <- plot$out
    return(data[!data$Y %in% outliers, ])
  } else if (objective == "Studentized") {
    model <- lm(Y ~ A * B, data = data)
    rij <- rstandard(model)
    nt <- nrow(data)
    a <- length(unique(data$A))
    b <- length(unique(data$B))
    t.cutoff <- qt(1 - alpha / (2 * nt), nt - a - b + 1)
    CO.rij <- which(abs(rij) > t.cutoff)

    if (length(CO.rij) == 0) {
      # No outlier
      return(data)
    } else {
      return(data[-CO.rij, ])
    }
  } else {
    warning("Invalid objective")
    return()
  }
}

do_diagnostic <- function(o_objective, t_objective, data, alpha = 0.05) {
  data <- remove_outlier(o_objective, data, alpha = alpha)
  return(do_transform(t_objective, data))
}
```

```r
#  Main
data <- read.csv("insurance.csv")
selected_columns <- c("charges", "children", "region")
data <- subset(data, select = selected_columns)
colnames(data) <- c("Y", "A", "B")
```

```r
res <- expand.grid(OutlierRemovalMethod = c("None", "Boxplot", "Studentized"),
                   TransformationMethod = c("None", "PPCC", "Shapiro-Wilk",
                                            "Log-Likelihood"))

res$data <- apply(res, 1, function(row) {
  return(do_diagnostic(row["OutlierRemovalMethod"],
                       row["TransformationMethod"], data))
})

res$SW_Test <- apply(res, 1, function(row) {
  return(sw_test(row["data"][[1]]))
})

res$BF_Test <- apply(res, 1, function(row) {
  return(bf_test(row["data"][[1]]))
})

res <- res[, -which(names(res) == "data")]

knitr::kable(res)
```