1. Calculate ICC

```
###### First, identify patients whose evaluation (PhyGA) did not change significantly in the first two weeks of treatment ######

# Load necessary libraries
library(openxlsx)
library(dplyr)
library(tidyverse)
library(psych)

# Read Excel file
file_path <- "F:/RStudio/MCID of CSU.xlsx"
data <- read.xlsx(file_path)

head(data)

# Define function to classify UAS7 scores
classify_uas7 <- function(score) {
  if (score == 0) {
    return("0")
  } else if (score >= 1 && score <= 6) {
    return("1")
  } else if (score >= 7 && score <= 15) {
    return("2")
  } else if (score >= 16 && score <= 27) {
    return("3")
  } else if (score >= 28 && score <= 42) {
    return("4")
  } else {
    return("Unknown")
  }
}

# Classify baseline UAS7 scores
data$`Baseline activity level` <- sapply(data$`Baseline.UAS7`, classify_uas7)

# Classify UAS7 scores after 4 weeks of treatment
data$`Activity level at 2 Weeks` <- sapply(data$`UAS7.at.2.Weeks`, classify_uas7)

# Convert column data to numeric and calculate UAS7 level change (PhyGA)
data$`Baseline activity level` <- as.numeric(data$`Baseline activity level`)
data$`Activity level at 2 Weeks` <- as.numeric(data$`Activity level at 2 Weeks`)
data$PhyGA <- data$`Baseline activity level` - data$`Activity level at 2 Weeks`
```

```
# Create a new Excel workbook and save PhyGA data
output_file <- "F:/RStudio/MCID of CSU/PhyGA.xlsx"
write.xlsx(data, output_file, rowNames = FALSE)

# Print success message
cat("PhyGA evaluation data has been saved to", output_file, "\n")

######### Step 2: Compare ICC coefficients for the first two weeks for patients with minimal change in evaluation (PhyGA) #########
# Specify the path and filename of the Excel file
file_path <- "F:/RStudio/MCID of CSU/PhyGA.xlsx"

library(readxl)
library(irr)

# Read Excel file
data_icc <- read_excel(file_path)

# View the first row to confirm accurate data import
head(data_icc)

# Filter patients with PhyGA = 0
data_icc_zero <- subset(data_icc, PhyGA == 0)

# Assign baseline and week 2 scores for the waiting treatment group
baseline_scores <- data_icc_zero$`Baseline.UAS7`
week2_scores <- data_icc_zero$`UAS7.at.2.Weeks`

# Check for missing values and remove them (if any)
valid_data <- complete.cases(baseline_scores, week2_scores)
baseline_scores <- baseline_scores[valid_data]
week2_scores <- week2_scores[valid_data]

# Calculate ICC for UAS7
icc_result <- ICC(cbind(baseline_scores, week2_scores))

# Print ICC results
print(icc_result)
```

2. Measure the correlation between the calibration (PatGA) and the outcome variable (UAS7 improvement value)

```
# Specify the path and filename of the Excel file
file_path <- "F:/RStudio/MCID of CSU.xlsx"

# Read the Excel file
data <- read_excel(file_path)

# Here we use sample data
data <- data.frame(
    Patient_ID = data$`Patient ID`,
    Baseline_UAS7 = data$`Baseline UAS7`,
    UAS7_at_2_Weeks = data$`UAS7 at 2 Weeks`,
    UAS7_at_4_Weeks = data$`UAS7 at 4 Weeks`,
    PatGA = data$`PatGA`
)

# Calculate the change in UAS7
data <- data %>%
    mutate(UAS7_change_4_weeks = Baseline_UAS7 - UAS7_at_4_Weeks)

# Calculate Spearman correlation coefficient
correlation <- cor(data$PatGA, data$UAS7_change_4_weeks, method = "spearman")

# Output the correlation coefficient and its significance level
cat("Spearman Correlation Coefficient:", correlation, "\n")

# Test the significance of the correlation coefficient
p_value <- cor.test(data$PatGA, data$UAS7_change_4_weeks, method = "spearman")$p.value
cat("P-value:", p_value, "\n")

# Determine if the correlation is significant based on the significance level
if (p_value < 0.05) {
    cat("The correlation is statistically significant.\n")
} else {
    cat("The correlation is not statistically significant.\n")
}
```

3. MCID Value Calculation (Anchor-based method: Mean Change Method): Using patients' self-assessment category (1 = Significant Improvement) as an example ######

```
# The number after "PatGA ==" corresponds to the level of self-assessment

# Install and load necessary packages
library(readxl)
library(dplyr)

# Specify the path and filename of the Excel file
file_path <- "F:/RStudio/MCID of CSU.xlsx"

# Read the Excel file
data <- read_excel(file_path)

# Select required columns and calculate UAS7 change value
data_processed <- data %>%
  select(Patient_ID = `Patient ID`,
         Baseline_UAS7 = `Baseline UAS7`,
         UAS7_at_2_Weeks = `UAS7 at 2 Weeks`,
         UAS7_at_4_Weeks = `UAS7 at 4 Weeks`,
         PatGA = `PatGA`) %>%
  mutate(UAS7_change_4_weeks = Baseline_UAS7 - UAS7_at_4_Weeks)

# Filter patients with PatGA = 1 (Significant Improvement)
data_improved <- filter(data_processed, PatGA == 1)

# Calculate the mean change in UAS7 for these patients (using 4-week data)
mcid_uas7_mean_change_4_weeks <- mean(data_improved$UAS7_change_4_weeks, na.rm = TRUE)

# Print the MCID value
print(paste("MCID for UAS7 at 4 Weeks (Mean Change Method, PatGA = 1):", mcid_uas7_mean_change_4_weeks))
```

4. Calculating Confidence Intervals Using Bootstrap Method Based on Anchor-Based Mean Change Method

```
# Install and load necessary packages
library(readxl)
library(dplyr)
library(boot)

# Specify the path and filename of the Excel file
file_path <- "F:/RStudio/MCID of CSU.xlsx"

# Read the Excel file
data <- read_excel(file_path)

# Select required columns and calculate UAS7 change value
data_processed <- data %>%
  select(Patient_ID = `Patient ID`,
         Baseline_UAS7 = `Baseline UAS7`,
         UAS7_at_2_Weeks = `UAS7 at 2 Weeks`,
         UAS7_at_4_Weeks = `UAS7 at 4 Weeks`,
         PatGA = `PatGA`) %>%
  mutate(UAS7_change_4_weeks = Baseline_UAS7 - UAS7_at_4_Weeks)

# Filter patients with PatGA = 1 (Significant Improvement)
data_improved <- filter(data_processed, PatGA == 1)

# Define a function to calculate MCID (for 4-week change)
mcid_function <- function(data, indices) {
  sample_data <- data[indices, ]
  # Ensure NA values are handled
  mean_change <- ifelse(any(is.na(sample_data$UAS7_change_4_weeks)), NA,
                        mean(sample_data$UAS7_change_4_weeks, na.rm = TRUE))
  return(mean_change)
}

# Use Bootstrap method
set.seed(123) # Set random seed for reproducibility
bootstrap_results <- boot(data = data_improved, statistic = mcid_function, R = 1000)

# Calculate confidence interval
confidence_interval <- boot.ci(bootstrap_results, type = "perc")

# Print confidence interval
print(confidence_interval)
```