Analysing Results of Questionnaire

aiswary-a

2024-04-28

Contents

0. Data Prep

0.1. Workspace Prep

```
# loading libraries
library(pacman)
pacman::p_load(tidyverse,
               car) # for accessing Levene's test
```

0.2. Cleaning the Data

```
# insert CSV
df <-
  read_csv(
    "C:/Users/aisro/Desktop/UNI/BSc CogSci - 2nd Sem/Applied/VSCode + Git/Form Responses/CogSci_Applied_
  ) # to be updated with final data, this is just a placeholder

# renaming the df columns
## NOTE: [1] pre-tool usage, [2] post-tool usage, [char] textual response, [num] numerical response
```

```r
renaming_dfCols <-
  c(
    "timestamp",
    "age",
    "vote_eligibility",
    "preTool_biasPerception",
    "preTool_biasAwareness",
    "preTool_morality",
    "preTool_valueCommitment",
    "email",
    "postTool_biasPerception",
    "postTool_biasAwareness",
    "postTool_morality",
    "postTool_valueCommitment",
    "char_engagement",
    "num_decision_enjoyment",
    "num_likely_to_recommend",
    "num_ranking_agreement",
    "char_ranking_selfperception",
    "char_past_candidate_test",
    "char_future_candidate_test",
    "char_comment_tool_selfperception",
    "char_comment_tool_design",
    "char_comment_other",
    "gender",
    "empty"
  )

colnames(df) <- renaming_dfCols

# deleting email column
df <- df %>%
  select(-email, -empty)

# removing first (test) row
df <- df[df$timestamp != "16/04/2024 14:07:03",]

# ID'ing non-numeric columnsss and making those all lowercase!
df <- df %>%
  mutate_if(~ !is.numeric(.), tolower)
```

```r
# groups for stat tests!

# vars. of interest :))
groupSingles <- c(
  "preTool_biasPerception",
  "postTool_biasPerception",
  "preTool_biasAwareness",
  "postTool_biasAwareness",
  "preTool_morality",
  "postTool_morality",
  "preTool_valueCommitment",
  "postTool_valueCommitment"
```

```
)

# defining the group pairings
groupPairs <- list(
  c("preTool_biasPerception", "postTool_biasPerception"),
  c("preTool_biasAwareness", "postTool_biasAwareness"),
  c("preTool_morality", "postTool_morality"),
  c("preTool_valueCommitment", "postTool_valueCommitment")
)
```

1. Demographics

```
print(paste("N =", nrow(df)))
```

```
## [1] "N = 16"
```

```
# age distribution
stats_dfAge <- df %>%
  summarise(mean = mean(age), sd = sd(age)) # insert age column

print(paste("The mean age:", round(stats_dfAge$mean, digits = 3)))
```

```
## [1] "The mean age: 28.438"
```

```
print(paste("The SD of the age:", round(stats_dfAge$sd, digits = 3)))
```

```
## [1] "The SD of the age: 14.278"
```

```
# gender distribution
stats_dfGender <- df %>%
  count(gender) # insert gender column

print(paste("The gender distribution of the sample:", stats_dfGender))
```

```
## [1] "The gender distribution of the sample: c(\"female\", \"male\")"
## [2] "The gender distribution of the sample: c(9, 7)"
```

2. Pre-Tool vs. Post-Tool Usage

2.1. Checking Assumptions

a. Normality

```r
# Shapiro-Wilk test for Normality!

stat_nResults <- list() # an empty list to store the results

# normality testing for-loop, going through each indexed column
for (i in seq_along(groupSingles)) {
  single <- groupSingles[i]  # extract column name using index

  # taking the group and forcing it to numeric if necessary
  item <-
    as.numeric(df[[single]])

  # the Shapiro-Wilk normality test!
  shapiro_result <-
    shapiro.test(item)

  # storing the results
  test_name <-
    paste("Shapiro-Wilk Test of Normality for", single)
  stat_nResults[[test_name]] <- shapiro_result

  # printing the result!
  cat(test_name, ":\n")
  print(shapiro_result)

  # performing the hypothesis test using results!
  if (!is.null(shapiro_result$p.value) &&
      shapiro_result$p.value > 0.05) {
    cat(
      "Fail to reject null hypothesis that data significantly differs from a normal distribution\n\n"
    )
  } else if (!is.null(shapiro_result$p.value) &&
             shapiro_result$p.value <= 0.05) {
    cat(
      "Reject the null hypothesis that data significantly differs from a normal distribution\n\n"
    )
  } else {
    cat("Unable to compute p-value for the test.\n\n")
  }
}
```

```
## Shapiro-Wilk Test of Normality for preTool_biasPerception :
##
##  Shapiro-Wilk normality test
##
## data:  item
## W = 0.8929, p-value = 0.06192
##
## Fail to reject null hypothesis that data significantly differs from a normal distribution
##
## Shapiro-Wilk Test of Normality for postTool_biasPerception :
##
##  Shapiro-Wilk normality test
```

```
##
## data:  item
## W = 0.92252, p-value = 0.2104
##
## Fail to reject null hypothesis that data significantly differs from a normal distribution
##
## Shapiro-Wilk Test of Normality for preTool_biasAwareness :
##
##  Shapiro-Wilk normality test
##
## data:  item
## W = 0.9227, p-value = 0.1864
##
## Fail to reject null hypothesis that data significantly differs from a normal distribution
##
## Shapiro-Wilk Test of Normality for postTool_biasAwareness :
##
##  Shapiro-Wilk normality test
##
## data:  item
## W = 0.95996, p-value = 0.6917
##
## Fail to reject null hypothesis that data significantly differs from a normal distribution
##
## Shapiro-Wilk Test of Normality for preTool_morality :
##
##  Shapiro-Wilk normality test
##
## data:  item
## W = 0.80769, p-value = 0.003452
##
## Reject the null hypothesis that data significantly differs from a normal distribution
##
## Shapiro-Wilk Test of Normality for postTool_morality :
##
##  Shapiro-Wilk normality test
##
## data:  item
## W = 0.79027, p-value = 0.002765
##
## Reject the null hypothesis that data significantly differs from a normal distribution
##
## Shapiro-Wilk Test of Normality for preTool_valueCommitment :
##
##  Shapiro-Wilk normality test
##
## data:  item
## W = 0.9269, p-value = 0.2176
##
## Fail to reject null hypothesis that data significantly differs from a normal distribution
##
## Shapiro-Wilk Test of Normality for postTool_valueCommitment :
##
##  Shapiro-Wilk normality test
```

```
##
## data:  item
## W = 0.88253, p-value = 0.05176
##
## Fail to reject null hypothesis that data significantly differs from a normal distribution
```

b. Homogeneity of Variance

```r
# Levene's test for homoscedasticity (homogeneity of variance)

stat_lResults <- list() # empty list to store results

# Levene's test for-loop, going through each indexed column
for (pair in groupPairs) {

  # printing the pair to see the comparison
  print(pair)

  # extracting the pair!
  group1 <- df[[pair[1]]] # member no. 1 of pair!
  group2 <- df[[pair[2]]] # member no. 2 of pair!

  # performing Levene's test
  lResults <-
    leveneTest(group1, group2) # funky naming to deal with overwriting issues :/

  # storing results in the pre-established list
  test_name <- paste("Levene test between", pair[1], "and", pair[2])
  stat_lResults[[test_name]] <- lResults

  # see the result!
  cat(test_name, ":\n")
  print(lResults)

  # performing the hypothesis test using results!
  if (!is.null(lResults$`Pr(>F)`[1]) &&
      lResults$`Pr(>F)`[1] > 0.05) {
    cat(
      "Fail to reject null hypothesis that there is no significant difference between the groups\n\n"
    )
  } else if (!is.null(lResults$`Pr(>F)`[1]) &&
             lResults$`Pr(>F)`[1] <= 0.05) {
    cat(
      "Reject the null hypothesis that there is no significant difference between the groups\n\n"
    )
  } else {
    cat("Unable to compute p-value for the test.\n\n")
  }
}
```

```
## [1] "preTool_biasPerception"  "postTool_biasPerception"
## Levene test between preTool_biasPerception and postTool_biasPerception :
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value   Pr(>F)
## group  5    8.85 0.002778 **
##        9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Reject the null hypothesis that there is no significant difference between the groups
##
## [1] "preTool_biasAwareness"  "postTool_biasAwareness"
## Levene test between preTool_biasAwareness and postTool_biasAwareness :
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  5  0.5551 0.7322
##        9
## Fail to reject null hypothesis that there is no significant difference between the groups
##
## [1] "preTool_morality"  "postTool_morality"
## Levene test between preTool_morality and postTool_morality :
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  1.5364 0.2546
##       12
## Fail to reject null hypothesis that there is no significant difference between the groups
##
## [1] "preTool_valueCommitment"  "postTool_valueCommitment"
## Levene test between preTool_valueCommitment and postTool_valueCommitment :
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  3  0.3847 0.7662
##       11
## Fail to reject null hypothesis that there is no significant difference between the groups
```

2.2. Stat. Testing!

a. Paired t-Test

```r
# paired t-Tests comparing pre-tool and post-tool usage !

stat_tResults <- list() # empty list to store the results

# Paired t-Test for-loop, iterating through each indexed column
for (pair in groupPairs) {

  # extracting the group pairs and performing numeric conversions where necessary
  group1 <- as.numeric(c(df[[pair[1]]]))
  group2 <- as.numeric(c(df[[pair[2]]]))

  # perfomring the paired t-test!
  stat_tResults <- t.test(group1, group2, paired = TRUE,
                          alternative = "two.sided")

  # storing the result in the pre-established list!
```

```r
  test_name <- paste("Paired t-test between", pair[1], "and", pair[2])
  stat_tResults[[test_name]] <- stat_tResults

  # seeing the result of our lovely test!
  cat(test_name, ":\n")
  print(stat_tResults)

  # performing the hypothesis test using results!
  if (!is.null(stat_tResults$p.value[1]) && stat_tResults$p.value[1] > 0.05) {
    cat("Fail to reject null hypothesis that there is no significant difference between the groups\n\n")
  } else if (!is.null(stat_tResults$p.value[1]) && stat_tResults$p.value[1] <= 0.05) {
    cat("Reject the null hypothesis that there is no significant difference between the groups\n\n")
  } else {
    cat("Unable to compute p-value for the test.\n\n")
  }
}
```

```
## Paired t-test between preTool_biasPerception and postTool_biasPerception :
##
##  Paired t-test
##
## data:  group1 and group2
## t = -1.8353, df = 14, p-value = 0.08779
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -1.15659390  0.08992723
## sample estimates:
## mean difference
##      -0.5333333
##
## Fail to reject null hypothesis that there is no significant difference between the groups
##
## Paired t-test between preTool_biasAwareness and postTool_biasAwareness :
##
##  Paired t-test
##
## data:  group1 and group2
## t = -1.526, df = 14, p-value = 0.1493
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -2.084746  0.351413
## sample estimates:
## mean difference
##      -0.8666667
##
## Fail to reject null hypothesis that there is no significant difference between the groups
##
## Paired t-test between preTool_morality and postTool_morality :
##
##  Paired t-test
##
## data:  group1 and group2
## t = 0.32323, df = 14, p-value = 0.7513
```

```
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.3756988  0.5090321
## sample estimates:
## mean difference
##      0.06666667
##
## Fail to reject null hypothesis that there is no significant difference between the groups
##
## Paired t-test between preTool_valueCommitment and postTool_valueCommitment :
##
##  Paired t-test
##
## data:  group1 and group2
## t = 1.5718, df = 14, p-value = 0.1383
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -0.145813  0.945813
## sample estimates:
## mean difference
##             0.4
##
## Fail to reject null hypothesis that there is no significant difference between the groups
```

3. Unused Code I'm Too Paranoid to Delete

```r
# making all text in df lowercase
df <- df %>%
  mutate(
    vote_eligibility = tolower(vote_eligibility),
    char_engagement = tolower(char_engagement),
    char_ranking_selfperception = tolower(char_engagement),
    char_past_candidate_test = tolower(char_past_candidate_test),
    char_future_candidate_test = tolower(char_future_candidate_test),
    char_comment_tool_selfperception = tolower(char_comment_tool_selfperception),
    char_comment_tool_design = tolower(char_comment_tool_design),
    char_comment_other = tolower(char_comment_other),
    gender = tolower(gender)
  )

# - - - -

# group pairings
gr_dfBiasPerception <- c(df$preTool_biasPerception,df$postTool_biasPerception)
gr_dfBiasAwareness <- c(df$preTool_biasAwareness, df$postTool_biasAwareness)
gr_dfMorality <- c(df$preTool_morality, df$postTool_morality)
gr_dfValCommitment <- c(df$preTool_valueCommitment, df$postTool_valueCommitment)
```