Normalitycheck

Chang Lu

2025-02-15

```
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(nortest)
```

Data clean

```
df <- read_excel("MAMS and Dental Combined for Analysis.xlsx", sheet = "Sheet1")
```

```
## New names:
## * `` -> `...1`
## * `E.` -> `E....10`
## * `E.` -> `E....12`
```

```
head(df)
```

```
## # A tibble: 6 x 30
##   ...1      Central diabetes ins~1 Decreased conduction~2 Mitral valve stenosi~3
##   <chr>     <chr>                  <chr>                  <chr>
## 1 Student   Q1                     Q2                     Q3
## 2 Student 1 1.0                    0.0                    0.0
## 3 Student 2 1.0                    1.0                    0.0
## 4 Student 3 0.0                    1.0                    0.0
## 5 Student 4 0.0                    0.0                    0.0
## 6 Student 5 1.0                    0.0                    1.0
## # i abbreviated names: 1: `Central diabetes insipidus`,
```

```
## #   2: 'Decreased conduction rate along the bundle branches',
## #   3: 'Mitral valve stenosis'
## # i 26 more variables:
## #   'Decreased pulmonary capillary hydrostatic fluid pressure' <chr>,
## #   Oxytocin <chr>, 'Graves' disease' <chr>, B. <chr>,
## #   'Increased serum aldosterone concentration' <chr>, E....10 <chr>, ...
```

```
summary(df)
```

```
##      ...1           Central diabetes insipidus
##  Length:36          Length:36
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##  Decreased conduction rate along the bundle branches Mitral valve stenosis
##  Length:36                                           Length:36
##  Class :character                                    Class :character
##  Mode  :character                                    Mode  :character
##  Decreased pulmonary capillary hydrostatic fluid pressure   Oxytocin
##  Length:36                                                  Length:36
##  Class :character                                           Class :character
##  Mode  :character                                           Mode  :character
##  Graves' disease         B.
##  Length:36          Length:36
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##  Increased serum aldosterone concentration   E....10
##  Length:36                                  Length:36
##  Class :character                           Class :character
##  Mode  :character                           Mode  :character
##  Arterial O2 concentration   E....12          Blocked urethra
##  Length:36                 Length:36          Length:36
##  Class :character          Class :character   Class :character
##  Mode  :character          Mode  :character   Mode  :character
##  Excess maternal androgens
##  Length:36
##  Class :character
##  Mode  :character
##  The elastic recoil of the stretched arterial walls provides the force to continue blood flow in the
##  Length:36
##  Class :character
##  Mode  :character
##  Mutations that result in inactive IGF-1 receptors
##  Length:36
##  Class :character
##  Mode  :character
##  A decrease in Ca2+ resorption from bone Absence of a Y chromosome
##  Length:36                               Length:36
##  Class :character                        Class :character
##  Mode  :character                        Mode  :character
##  Testosterone stimulates GnRH from the hypothalamus
##  Length:36
##  Class :character
##  Mode  :character
##  Plasma angiotensin II concentration increases
```

```
##  Length:36
##  Class :character
##  Mode  :character
##  Its production is enhanced by cortisol. Total Score         Quiz Time
##  Length:36                               Length:36          Length:36
##  Class :character                        Class :character   Class :character
##  Mode  :character                        Mode  :character   Mode  :character
##  Accomodations          Sex            Race/Ethnicity     English Proficiency
##  Length:36          Length:36          Length:36          Length:36
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##    Born USA         Home Language      Age arrive USA
##  Length:36          Length:36          Length:36
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

```
df_students <- df[2:32, ]

head(df_students)
```

```
## # A tibble: 6 x 30
##   ...1      Central diabetes ins~1 Decreased conduction~2 Mitral valve stenosi~3
##   <chr>     <chr>                  <chr>                  <chr>
## 1 Student 1 1.0                    0.0                    0.0
## 2 Student 2 1.0                    1.0                    0.0
## 3 Student 3 0.0                    1.0                    0.0
## 4 Student 4 0.0                    0.0                    0.0
## 5 Student 5 1.0                    0.0                    1.0
## 6 Student 6 1.0                    1.0                    0.0
## # i abbreviated names: 1: 'Central diabetes insipidus',
## #   2: 'Decreased conduction rate along the bundle branches',
## #   3: 'Mitral valve stenosis'
## # i 26 more variables:
## #   'Decreased pulmonary capillary hydrostatic fluid pressure' <chr>,
## #   Oxytocin <chr>, 'Graves' disease' <chr>, B. <chr>,
## #   'Increased serum aldosterone concentration' <chr>, E....10 <chr>, ...
```

```
# Define the new column names for the questions
question_names <- paste0("Q", 1:20)  # Generates Q1, Q2, ..., Q20

# Rename only the question columns
colnames(df_students)[2:21] <- question_names

head(df_students)
```

```
## # A tibble: 6 x 30
##   ...1   Q1    Q2    Q3    Q4    Q5    Q6    Q7    Q8    Q9    Q10   Q11   Q12
##   <chr>  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Stude~ 1.0   0.0   0.0   0.0   0.0   1.0   1.0   0.0   1.0   0.0   1.0   1.0
## 2 Stude~ 1.0   1.0   0.0   0.0   1.0   1.0   1.0   1.0   1.0   0.0   0.0   1.0
## 3 Stude~ 0.0   1.0   0.0   1.0   1.0   1.0   1.0   1.0   0.0   0.0   0.0   0.0
## 4 Stude~ 0.0   0.0   0.0   1.0   1.0   1.0   1.0   1.0   1.0   0.0   0.0   0.0
## 5 Stude~ 1.0   0.0   1.0   1.0   1.0   0.0   1.0   1.0   1.0   0.0   0.0   1.0
```

```
## 6 Stude~ 1.0   1.0   0.0   0.0   0.0   1.0   1.0   1.0   1.0   0.0   1.0   0.0
## # i 17 more variables: Q13 <chr>, Q14 <chr>, Q15 <chr>, Q16 <chr>, Q17 <chr>,
## #   Q18 <chr>, Q19 <chr>, Q20 <chr>, `Total Score` <chr>, `Quiz Time` <chr>,
## #   Accomodations <chr>, Sex <chr>, `Race/Ethnicity` <chr>,
## #   `English Proficiency` <chr>, `Born USA` <chr>, `Home Language` <chr>,
## #   `Age arrive USA` <chr>
```

```
colnames(df_students)
```

```
##  [1] "...1"                "Q1"                  "Q2"
##  [4] "Q3"                  "Q4"                  "Q5"
##  [7] "Q6"                  "Q7"                  "Q8"
## [10] "Q9"                  "Q10"                 "Q11"
## [13] "Q12"                 "Q13"                 "Q14"
## [16] "Q15"                 "Q16"                 "Q17"
## [19] "Q18"                 "Q19"                 "Q20"
## [22] "Total Score"         "Quiz Time"           "Accomodations"
## [25] "Sex"                 "Race/Ethnicity"      "English Proficiency"
## [28] "Born USA"            "Home Language"       "Age arrive USA"
```

```
# Convert value format to numeric
df_students$`Total Score` <- as.numeric(df_students$`Total Score`)
df_students$`Quiz Time` <- as.numeric(df_students$`Quiz Time`)
df_students$Accomodations <- as.numeric(df_students$Accomodations)
df_students$Sex <- as.numeric(df_students$Sex)
df_students$`Race/Ethnicity` <- as.numeric(df_students$`Race/Ethnicity`)
df_students$`English Proficiency` <- as.numeric(df_students$`English Proficiency`)
df_students$`Born USA` <- as.numeric(df_students$`Born USA`)
df_students$`Home Language` <- as.numeric(df_students$`Home Language`)
df_students$`Age arrive USA`<- as.numeric(df_students$`Age arrive USA`)


# Convert Quiz Time from seconds to minutes
df_students$`Quiz Time Minutes` <- df_students$`Quiz Time` / 60 # for later test
# Create Quiz Time Bins
df_students$`Quiz Time Group` <- cut(df_students$`Quiz Time Minutes`,
                                     breaks = c(0, 10, 20, Inf),
                                     labels = c("0-10 min", "10-20 min", "20+ min"),
                                     include.lowest = TRUE)

# Check new groups
table(df_students$`Quiz Time Group`)
```

```
##
##  0-10 min 10-20 min   20+ min
##         7         7        17
```

```
# Verify changes
str(df_students)
```

```
## tibble [31 x 32] (S3: tbl_df/tbl/data.frame)
##  $ ...1               : chr [1:31] "Student 1" "Student 2" "Student 3" "Student 4" ...
```

```
##  $ Q1                : chr [1:31] "1.0" "1.0" "0.0" "0.0" ...
##  $ Q2                : chr [1:31] "0.0" "1.0" "1.0" "0.0" ...
##  $ Q3                : chr [1:31] "0.0" "0.0" "0.0" "0.0" ...
##  $ Q4                : chr [1:31] "0.0" "0.0" "1.0" "1.0" ...
##  $ Q5                : chr [1:31] "0.0" "1.0" "1.0" "1.0" ...
##  $ Q6                : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Q7                : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Q8                : chr [1:31] "0.0" "1.0" "1.0" "1.0" ...
##  $ Q9                : chr [1:31] "1.0" "1.0" "0.0" "1.0" ...
##  $ Q10               : chr [1:31] "0.0" "0.0" "0.0" "0.0" ...
##  $ Q11               : chr [1:31] "1.0" "0.0" "0.0" "0.0" ...
##  $ Q12               : chr [1:31] "1.0" "1.0" "0.0" "0.0" ...
##  $ Q13               : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Q14               : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Q15               : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Q16               : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Q17               : chr [1:31] "1.0" "0.0" "1.0" "1.0" ...
##  $ Q18               : chr [1:31] "1.0" "0.0" "1.0" "1.0" ...
##  $ Q19               : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Q20               : chr [1:31] "1.0" "1.0" "1.0" "1.0" ...
##  $ Total Score       : num [1:31] 0.7 0.7 0.7 0.7 0.7 0.7 0.6 0.6 0.6 0.6 ...
##  $ Quiz Time         : num [1:31] 1463 2574 2744 93675 38755 ...
##  $ Accomodations     : num [1:31] 2 2 2 1 2 2 2 2 2 2 ...
##  $ Sex               : num [1:31] 0 1 0 1 1 0 1 1 1 1 ...
##  $ Race/Ethnicity    : num [1:31] 1 1 1 0 0 1 0 1 1 0 ...
##  $ English Proficiency: num [1:31] 0 1 1 0 0 0 0 0 1 1 ...
##  $ Born USA          : num [1:31] 0 0 1 0 1 0 0 0 0 0 ...
##  $ Home Language     : num [1:31] 0 1 1 0 1 0 0 0 1 0 ...
##  $ Age arrive USA    : num [1:31] 0 0 4 0 3 0 0 0 0 0 ...
##  $ Quiz Time Minutes : num [1:31] 24.4 42.9 45.7 1561.2 645.9 ...
##  $ Quiz Time Group   : Factor w/ 3 levels "0-10 min","10-20 min",..: 3 3 3 3 3 2 3 2 3 2 ...
```

Quiz_Time = check_normality_by_group(df_students, "Quiz Time Minutes","Total Score"),

Accomodations = check_normality_by_group(df_students, "Accomodations","Total Score"),

Age_Arrive_USA = check_normality_by_group(df_students, "Age arrive USA","Total Score")

Shapiko-Wilk Test

```
# function for normality check using the Shapiro-Wilk test
# if the result of p-value >0.05, the data is normal
check_normality_by_group <- function(data, group_col, test_col) {
  data %>%
    group_by(!!sym(group_col)) %>%
```

```
    summarise(
      Shapiro_Statistic = shapiro.test(na.omit(!!sym(test_col)))$statistic,
      P_Value = shapiro.test(na.omit(!!sym(test_col)))$p.value
    ) %>%
    arrange(P_Value)
}
```

```
# Check normality for Total Score by demographic group
normality_total_score <- list(
  Sex = check_normality_by_group(df_students, "Sex", "Total Score"),
  Race_Ethnicity = check_normality_by_group(df_students, "Race/Ethnicity", "Total Score"),
  English_Proficiency = check_normality_by_group(df_students, "English Proficiency", "Total Score"),
  Born_USA = check_normality_by_group(df_students, "Born USA", "Total Score"),
  Home_Language = check_normality_by_group(df_students, "Home Language", "Total Score")
)
```

```
# Print results
print("Normality Test for Total Score by Group")
```

```
## [1] "Normality Test for Total Score by Group"
```

```
print(normality_total_score)
```

```
## $Sex
## # A tibble: 2 x 3
##     Sex Shapiro_Statistic P_Value
##   <dbl>             <dbl>   <dbl>
## 1     1             0.903  0.0297
## 2     0             0.841  0.0777
##
## $Race_Ethnicity
## # A tibble: 2 x 3
##   `Race/Ethnicity` Shapiro_Statistic P_Value
##              <dbl>             <dbl>   <dbl>
## 1                1             0.898  0.0539
## 2                0             0.932  0.363
##
## $English_Proficiency
## # A tibble: 2 x 3
##   `English Proficiency` Shapiro_Statistic P_Value
##                   <dbl>             <dbl>   <dbl>
## 1                     0             0.910  0.0868
## 2                     1             0.905  0.159
##
## $Born_USA
## # A tibble: 2 x 3
##   `Born USA` Shapiro_Statistic P_Value
##        <dbl>             <dbl>   <dbl>
## 1          0             0.914  0.0428
## 2          1             0.899  0.326
##
## $Home_Language
```

```
## # A tibble: 2 x 3
##   `Home Language` Shapiro_Statistic P_Value
##             <dbl>             <dbl>   <dbl>
## 1               0             0.907  0.0651
## 2               1             0.919  0.279
```

Both groups p > 0.05, we use t-test. At least one group p <= 0.05, we use Mann-Whitney U test (nonparametric).

Sex: MWU Race: T-test English: t-test BornUSA: MWU homeLanguage: t-test

Kolmogorov-Smirnov Test

```
# function using the Kolmogorov-Smirnov Test
check_ks_normality_by_group <- function(data, group_col, test_col) {
  data %>%
    group_by(!!sym(group_col)) %>%
    summarise(
      Sample_Size = n(),
      KS_Statistic = ifelse(Sample_Size >= 3,
                            ks.test(na.omit(!!sym(test_col)), "pnorm",
                                    mean = mean(na.omit(!!sym(test_col))),
                                    sd = sd(na.omit(!!sym(test_col))))$statistic,
                            NA),
      P_Value = ifelse(Sample_Size >= 3,
                       ks.test(na.omit(!!sym(test_col)), "pnorm",
                               mean = mean(na.omit(!!sym(test_col))),
                               sd = sd(na.omit(!!sym(test_col))))$p.value,
                       NA)
    ) %>%
    arrange(P_Value)
}
```

```
# Apply the KS test to different groups
ks_normality_accomodations <- check_ks_normality_by_group(df_students, "Accomodations", "Total Score")
```

```
## Warning: There were 2 warnings in `summarise()`.
## The first warning was:
## i In argument: `KS_Statistic = ifelse(...)`.
## i In group 2: `Accomodations = 2`.
## Caused by warning in `ks.test.default()`:
## ! ties should not be present for the one-sample Kolmogorov-Smirnov test
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
print("Kolmogorov-Smirnov Normality Test for Total Score by Quiz Time Group")
```

```
## [1] "Kolmogorov-Smirnov Normality Test for Total Score by Quiz Time Group"
```

```
print(ks_normality_accomodations)
```

```
## # A tibble: 2 x 4
##   Accomodations Sample_Size KS_Statistic P_Value
##           <dbl>       <int>        <dbl>   <dbl>
## 1             2          29        0.145   0.578
## 2             1           2       NA      NA
```

Lilliefors Test

```
#  Function using Lilliefors Test
check_lilliefors_normality_by_group <- function(data, group_col, test_col) {
  data %>%
    group_by(!!sym(group_col)) %>%
    summarise(
      Sample_Size = n(),
      Lilliefors_Statistic = ifelse(Sample_Size >= 3,
                                    lillie.test(na.omit(!!sym(test_col)))$statistic,
                                    NA),
      P_Value = ifelse(Sample_Size >= 3,
                       lillie.test(na.omit(!!sym(test_col)))$p.value,
                       NA)
    ) %>%
    arrange(P_Value)
}
```

```
lilliefors_normality_accomodations <- check_lilliefors_normality_by_group(df_students, "Accomodations",
lilliefors_normality_quiz_time <- check_lilliefors_normality_by_group(df_students, "Quiz Time Group", "
lilliefors_normality_Age_Arrive_USA <- check_lilliefors_normality_by_group(df_students, "Age arrive USA
```

```
print("Lilliefors Normality Test for Total Score by different Group")
```

```
## [1] "Lilliefors Normality Test for Total Score by different Group"
```

```
print(lilliefors_normality_accomodations)
```

```
## # A tibble: 2 x 4
##   Accomodations Sample_Size Lilliefors_Statistic P_Value
##           <dbl>       <int>               <dbl>   <dbl>
## 1             2          29               0.145   0.125
## 2             1           2              NA      NA
```

```
print(lilliefors_normality_quiz_time)
```

```
## # A tibble: 3 x 4
##   `Quiz Time Group` Sample_Size Lilliefors_Statistic P_Value
##   <fct>                   <int>               <dbl>   <dbl>
## 1 0-10 min                    7               0.243   0.239
## 2 20+ min                    17               0.166   0.240
## 3 10-20 min                   7               0.178   0.721
```

```r
print(lilliefors_normality_Age_Arrive_USA)
```

```
## # A tibble: 5 x 4
##   `Age arrive USA` Sample_Size Lilliefors_Statistic P_Value
##              <dbl>       <int>                <dbl>   <dbl>
## 1                0          24                0.163  0.0978
## 2                2           2               NA     NA
## 3                3           2               NA     NA
## 4                4           2               NA     NA
## 5                5           1               NA     NA
```

I suggest using MWU test for accomodations and Age arrive USA, for one of the group sample size is too small(< 3), fails to using normality tests.

The quiz time group(categorized by 10 minutes) is suitable to use normality test.

Conclusion:

T-test:

1. quiz time(categorized by 10 minutes)
2. English Proficiency
3. Race
4. Home Language

Mann-Whitney U Test

1. Accomodations
2. Age Arrive USA
3. Sex
4. Born in USA