

```
In [1]: install.packages("mediation")
library(mediation)
```

```
also installing the dependencies 'diffobj', 'brio', 'desc', 'pkgload', 'praise',
'processx', 'waldo', 'checkmate', 'testthat', 'gridExtra', 'htmlTable', 'viridis',
'Formula', 'minqa', 'nloptr', 'mvtnorm', 'sandwich', 'lpSolve', 'Hmisc', 'lme4'
```

```
Updating HTML index of packages in '.Library'
```

```
Making 'packages.html' ...
done
```

```
Loading required package: MASS
```

```
Loading required package: Matrix
```

```
Loading required package: mvtnorm
```

```
Loading required package: sandwich
```

```
mediation: Causal Mediation Analysis
Version: 4.5.0
```

```
In [2]: # Assume your dataset is named "Data" with columns: study_time, motivation, and exa
data <- read.csv("Pilot_modified_data_1.csv")
```

```
# Step 1: Check for normality in the data
```

```
# Shapiro-Wilk test for normality
```

```
shapiro_test_BFI_E <- shapiro.test(data$BFI..E.)
```

```
shapiro_test_PAQ <- shapiro.test(data$PAQ)
```

```
shapiro_test_Panas <- shapiro.test(data$Panas..)
```

```
# Check the results
```

```
print("Shapiro-Wilk Test for Normality:")
```

```
print(paste("BFI..E. p-value:", shapiro_test_BFI_E$p.value))
```

```
print(paste("PAQ p-value:", shapiro_test_PAQ$p.value))
```

```
print(paste("Panas.. p-value:", shapiro_test_Panas$p.value))
```

```
[1] "Shapiro-Wilk Test for Normality:"
[1] "BFI..E. p-value: 1.19656516587766e-10"
[1] "PAQ p-value: 4.34455538770965e-06"
[1] "Panas.. p-value: 1.04478453267043e-12"
```

```
In [6]: # Assume your dataset is named "Data" with columns: Panas.., Panas...1, BFI..E., BF
data <- read.csv("Pilot_modified_data_1.csv")
```

```
# Function to perform Shapiro-Wilk test for normality and display p-values
```

```
perform_shapiro_test <- function(column_name) {
```

```
  shapiro_test <- shapiro.test(data[[column_name]])
```

```
  cat("Shapiro-Wilk Test for Normality of", column_name, ":\n")
```

```
  print(paste("p-value:", shapiro_test$p.value))
```

```
}
```

```
# List of columns to test for normality
columns_to_test <- c('Panas..', 'Panas...1', 'BFI..E.', 'BFI..A.', 'BFI..C.', 'BFI..N.', 'BFI..O.', 'EI..Self.A.', 'EI..Self.M.', 'EI..Social.A.', 'EI..RM.', 'PAQ', 'CBCL')

# Perform Shapiro-Wilk test for normality for each column
for (column in columns_to_test) {
  perform_shapiro_test(column)
}
```

```
Shapiro-Wilk Test for Normality of Panas.. :
[1] "p-value: 1.04478453267043e-12"
Shapiro-Wilk Test for Normality of Panas...1 :
[1] "p-value: 5.95276089703684e-08"
Shapiro-Wilk Test for Normality of BFI..E. :
[1] "p-value: 1.19656516587766e-10"
Shapiro-Wilk Test for Normality of BFI..A. :
[1] "p-value: 2.14568037923264e-06"
Shapiro-Wilk Test for Normality of BFI..C. :
[1] "p-value: 7.52179676169919e-10"
Shapiro-Wilk Test for Normality of BFI..N. :
[1] "p-value: 5.11244061091894e-06"
Shapiro-Wilk Test for Normality of BFI..O. :
[1] "p-value: 9.01788050988435e-09"
Shapiro-Wilk Test for Normality of EI..Self.A. :
[1] "p-value: 0.00113402080211277"
Shapiro-Wilk Test for Normality of EI..Self.M. :
[1] "p-value: 0.395299339675702"
Shapiro-Wilk Test for Normality of EI..Social.A. :
[1] "p-value: 0.316562818856219"
Shapiro-Wilk Test for Normality of EI..RM. :
[1] "p-value: 0.202736089956561"
Shapiro-Wilk Test for Normality of Total :
[1] "p-value: 0.000315016029120591"
Shapiro-Wilk Test for Normality of PAQ :
[1] "p-value: 4.34455538770965e-06"
Shapiro-Wilk Test for Normality of CBCL :
[1] "p-value: 7.05797768586667e-11"
```

```
In [10]: #the code to check the p-values from the Shapiro-Wilk test and display "Check" if t

# Assume your dataset is named "Data" with columns: study_time, motivation, and exa
data <- read.csv("Pilot_modified_data_1.csv")

# Create a vector with column names to test
columns_to_test <- c('Panas..', 'Panas...1', 'BFI..E.', 'BFI..A.', 'BFI..C.', 'BFI..N.', 'BFI..O.', 'EI..Self.A.', 'EI..Self.M.', 'EI..Social.A.', 'EI..RM.', 'PAQ', 'CBCL')

# Function to perform Shapiro-Wilk test and display the result
check_normality <- function(column) {
  shapiro_test_result <- shapiro.test(data[[column]])
  if (shapiro_test_result$p.value < 0.05) {
    return("Check")
  } else {
    return("Okay")
  }
}
```

```
# Check for normality in each column
normality_results <- sapply(columns_to_test, check_normality)

# Display the results
print("Shapiro-Wilk Test Results:")
for (i in seq_along(columns_to_test)) {
  print(paste(columns_to_test[i], ":", normality_results[i]))
}
```

```
[1] "Shapiro-Wilk Test Results:"
[1] "Panas.. : Check"
[1] "Panas...1 : Check"
[1] "BFI..E. : Check"
[1] "BFI..A. : Check"
[1] "BFI..C. : Check"
[1] "BFI..N. : Check"
[1] "BFI..O. : Check"
[1] "EI..Self.A. : Check"
[1] "EI..Self.M. : Okay"
[1] "EI..Social.A. : Okay"
[1] "EI..RM. : Okay"
[1] "Total : Check"
[1] "PAQ : Check"
[1] "CBCL : Check"
```

In [5]:

```
Error in inherits(model.y, "gam"): argument "model.y" is missing, with no default
Traceback:
```

```
1. perform_sobel_mediation(dep_var)
2. mediate(formula_outcome, mediator = formula_mediator, data = data,
.      boot = TRUE, test = "Sobel") # at line 13 of file <text>
```

In [11]:

```
Error in inherits(model.y, "gam"): argument "model.y" is missing, with no default
Traceback:
```

```
1. mediate(formula_outcome, mediator = formula_mediator, data = Data,
.      boot = TRUE, test = "Sobel")
```

In [12]:

```
Updating HTML index of packages in '.Library'
```

```
Making 'packages.html' ...
done
```

```
Updating HTML index of packages in '.Library'
```

```
Making 'packages.html' ...
done
```

```
Error in inherits(model.y, "gam"): argument "model.y" is missing, with no default  
Traceback:
```

```
1. mediate(formula_outcome, mediator = formula_mediator, data = Data,  
  .      boot = TRUE, test = "Sobel")
```

In [ ]: