mediation_example

Creating R square mediated function

```r
rsquare_med <- function(data, x, m, y) {
  # Compute correlations among the variables
  rxm <- cor(data[[x]], data[[m]])
  rxy <- cor(data[[x]], data[[y]])
  rmy <- cor(data[[m]], data[[y]])

  # Regression: m ~ x (to get alpha, first indirect path)
  model1 <- lm(as.formula(paste(m, "~", x)), data = data)
  alpha <- coef(model1)[[x]]

  # Regression: y ~ x + m (to get 'tauprime' and 'beta')
  model2 <- lm(as.formula(paste(y, "~", x, "+", m)), data = data)
  # Indirect effect of x on y via M = alpha*beta
  tauprime <- coef(model2)[[x]]
  beta <- coef(model2)[[m]]

  # Compute total effect of x on y: tau = tauprime + (alpha*beta)
  total <- tauprime + (alpha*beta)


  # Compute effect-size measures
  mediatedeffect <- alpha * beta         # the mediated effect (a * b)
  rxmsquared <- rxm^2                      # squared correlation between x and m
  partialrxy_msquared <- ((rxy - rmy * rxm) / sqrt((1 - rmy^2) * (1 - rxmsquared)))^2
  partialrmy_xsquared <- ((rmy - rxy * rxm) / sqrt((1 - rxy^2) * (1 - rxmsquared)))^2
  overallrsquared <- (((rxy^2) + (rmy^2)) - (2 * rxy * rmy * rxm)) / (1 - rxmsquared)
  rsquaredmediated <- (rmy^2) - (overallrsquared - (rxy^2))

  # Create a list of results
  results <- list(
    alpha = alpha,
    beta = beta,
    tauprime = tauprime,
    total = total,
    mediatedeffect = mediatedeffect,
    rxm = rxm,
    rxmsquared = rxmsquared,
    rxy = rxy,
    rmy = rmy,
    partialrxy_msquared = partialrxy_msquared,
    partialrmy_xsquared = partialrmy_xsquared,
    overallrsquared = overallrsquared,
    rsquaredmediated = rsquaredmediated
  )

  return(results)
}
```

Getting example data

```
library("MEPS")
library("dplyr")          # Data wrangling
```

```
Warning: package 'dplyr' was built under R version 4.3.3


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("gtsummary")      # Create tables
```

```
Warning: package 'gtsummary' was built under R version 4.3.3
```

```r
library("bda")            # Perform Sobel text
```

```
Warning: package 'bda' was built under R version 4.3.3
```

```
Loading required package: boot
```

```
bda - 18.3.2
```

```r
library("mediation")       # Perform the bootstrap approach
```

```
Warning: package 'mediation' was built under R version 4.3.3
```

```
Loading required package: MASS
```

```

Attaching package: 'MASS'
```

```
The following object is masked from 'package:gtsummary':

    select
```

```
The following object is masked from 'package:dplyr':

    select
```

```
Loading required package: Matrix
```

```
Warning: package 'Matrix' was built under R version 4.3.3
```

```
Loading required package: mvtnorm
```

```
Warning: package 'mvtnorm' was built under R version 4.3.3
```

```
Loading required package: sandwich
```

```
Warning: package 'sandwich' was built under R version 4.3.3
```

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

```r
library(tidyr)
```

```
Warning: package 'tidyr' was built under R version 4.3.3
```

```

Attaching package: 'tidyr'
```

```
The following objects are masked from 'package:Matrix':

    expand, pack, unpack
```

```r
####Load data from AHRQ MEPS website
hc2021 = read_MEPS(file = "h233")


### Step 5: Change column names to lowercase
names(hc2021) <- tolower(names(hc2021))

### Step 6: Select specific variables
### 2021
hc2021p = hc2021 %>%
  rename(
    workdays = ddnwrk21,
    diabetes = diabdx_m18,
    health_status = rthlth31) %>%
  dplyr::select(
    dupersid,
    workdays,
    diabetes,
    health_status,
    sex)
hc2021p$year <- 2021

### Step 7: Clean data (We don't want to include any missing or NA responses)
hc2021p = hc2021p %>%
  filter(workdays >= 0,
         diabetes >= 1,
         health_status >= 1)

# We want "No diabetes" to have a value of 0 because it will make interpreting the model e
```

```r
hc2021p$diabetes[hc2021p$diabetes == 2] = 0
```

Running example

```r
results <- rsquare_med(data = hc2021p, x = "diabetes", m = "health_status", y = "workdays"


# Print the results
print(results %>% as.data.frame()  %>% t() %>% round(4))
```

```
                      [,1]
alpha               0.7328
beta                1.7250
tauprime            0.5224
total               1.7864
mediatedeffect      1.2640
rxm                 0.2163
rxmsquared          0.0468
rxy                 0.0483
rmy                 0.1610
partialrxy_msquared 0.0002
partialrmy_xsquared 0.0238
overallrsquared     0.0261
rsquaredmediated    0.0021
```