Mediation_count_DV

This PDF is used in the following YouTube tutorial.

https://youtu.be/9UnPjISWO5s

Warning:

a. Please make sure to check the video description to read a published paper. Since I am not the author of the paper, I cannot guarantee that the R code shown in this video correctly reflects the idea of that paper. Please read the paper by yourself and make your own judgment.

Geldhof et al. (2017) Accommodating binary and count variables in mediation: A case for conditional indirect effects https://journals.sagepub.com/doi/abs/10.1177/0165025417727876

b. Please do NOT cite this video as a reference, if you are writing an academic paper. Further, I do not provide consulting services. But, you can leave a comment down below, and I am more than happy to help. The correctness and quality of the R code presented in this tutorial are not guaranteed.

c. The R code here uses quasi-Poisson. X is continuous in the R code. The final mediation effect is calculated based on the mean of X.

Main R Code:

1. Data Simulation

X: Mean = 5, SD=4

M = 0.3+0.5X

Y =exp(0.2 + 0.2M+0.08X)

```
###########################################
###  data simulation
###########################################

# set the size of the sample
n=500
# set seed
set.seed(123)

# simulate x (normal distribution)
X <- rnorm(n, 5, 4)
# calculate the mean of X
mean_x=mean(X)
```

```r
# simulate a residual for M
residual_1<-rnorm(n,0,1)
M<-0.3+0.5*X+residual_1

# mu for Poisson regression via a log link
mu_1 <- exp(0.2 + 0.2*M+0.08*X)
# use rpois to generate Y
Y <- rpois(n, lambda=mu_1)

# combine into a dataframe and print out the first 6 rows
data <- data.frame(X=X, M=M, Y=Y)
head(data)
```

```
##           X        M  Y
## 1  2.758097 1.077156  1
## 2  4.079290 1.345946  1
## 3 11.234833 6.944202  8
## 4  5.282034 3.692078  4
## 5  5.517151 1.549409  2
## 6 11.860260 6.134983 17
```

2. Key Function

```r
####################################################
###  Serial Mediation Analysis for Count data
####################################################
library(boot)
```

```
## Warning: package 'boot' was built under R version 4.1.3
```

```r
set.seed(123)


Mediation_function_poisson<-function(data_used,i,x_predetermined=0)
{
  # Sample a data
  data_temp=data_used[i,]

  # Deciding which X value to use
  if(x_predetermined==0){x_predetermined=mean(data_temp$X)}
  else if (x_predetermined==-1){x_predetermined=mean(data_temp$X)-sd(data_temp$X)}
  else(x_predetermined=mean(data_temp$X)+sd(data_temp$X))

  # a path
  result_a<-lm(M~X, data = data_temp)
  a_0<-result_a$coefficients[1]
  a_1<-result_a$coefficients[2]

  # b path
  result_b<-glm(Y~M+X, data = data_temp, family = quasipoisson)
  b_0<-result_b$coefficients[1]
```

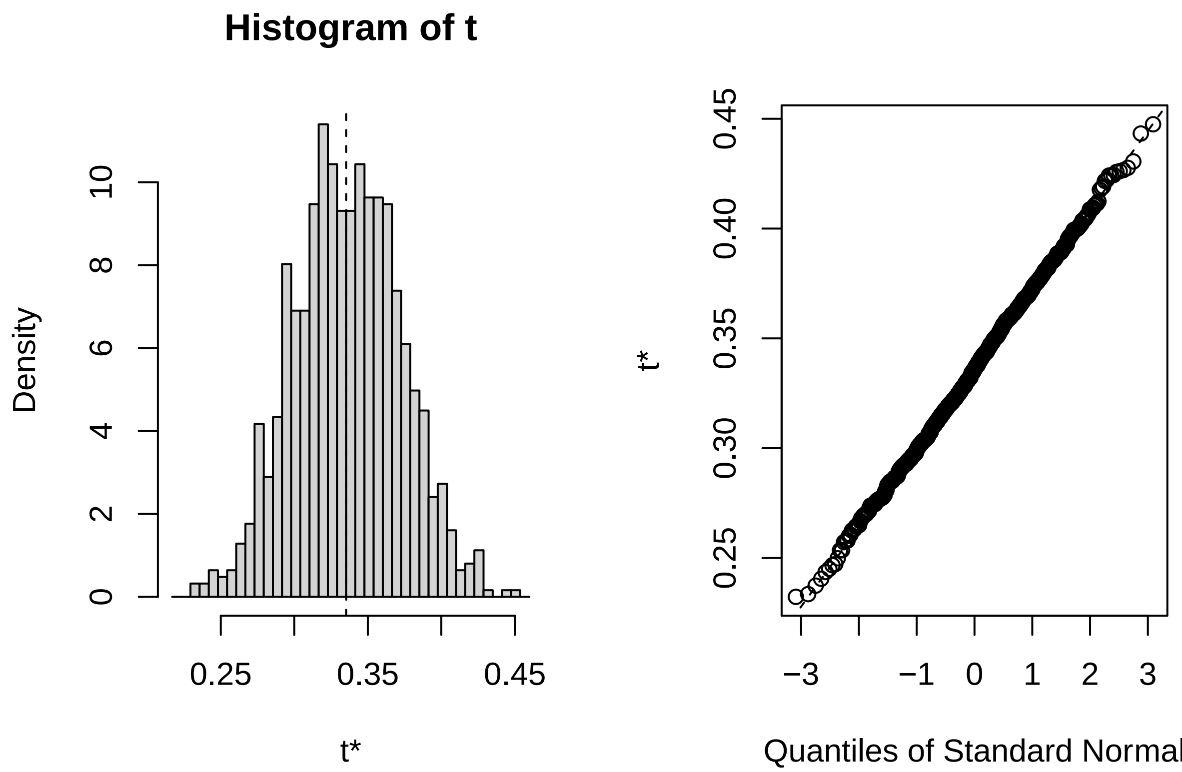
```r
  b_1<-result_b$coefficients[2]
  c_1_apostrophe<-result_b$coefficients[3]

  #calculating the indirect effect
  M_estimated=a_0+a_1*x_predetermined
  indirect_effect<-a_1*b_1*exp(b_0+b_1*M_estimated+c_1_apostrophe*x_predetermined)
  return(indirect_effect)
}
```

3. Use the function

```r
# use boot() to do bootstrapping mediation analysis
boot_mediation <- boot(data, Mediation_function_poisson, R=1000)
boot_mediation
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data, statistic = Mediation_function_poisson, R = 1000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.3352801 0.0009118087  0.03610191
```

```r
# plot the 1000 indirect effects
plot(boot_mediation)
```




```r
# print out confidence intervals
boot.ci(boot.out = boot_mediation, type = c("norm", "perc"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_mediation, type = c("norm", "perc"))
##
## Intervals :
## Level      Normal              Percentile
## 95%   ( 0.2636,  0.4051 )   ( 0.2681,  0.4062 )
## Calculations and Intervals on Original Scale
```

4. Check

```r
##################################################
###  calculate the theoretical value
##################################################

mean_x<-mean(data$X)
theoretical_indirect_effect<-0.5*0.2*exp(0.2+0.2*(0.3+0.5*mean_x)+0.08*mean_x)
theoretical_indirect_effect
```

```
## [1] 0.3270377
```