# Mediational Models and Bootstrapping

*Andrew L. Stewart*

*June 7, 2017*

## Read in the data

```
cromley.dat <- read.table("./Cromley2017.dat", header=T)
library(lavaan)
```

```
## Warning: package 'lavaan' was built under R version 3.3.2
```

```
## This is lavaan 0.5-23.1097
```

```
## lavaan is BETA software! Please report any bugs.
```

## Specify the model

```
model <- '#direct effect
          comprehension ~ a*reading + b*vocab +c*inference

          #mediator
          inference ~ d*reading + e*vocab

          #correlation
          reading~~vocab

          #indirect effect
          dc := d*c
          ec := e*c

          #total effect
          total := a+b+(d*c)+(e*c)
'
```

## Fit the model

```
fit <- sem(model,se="bootstrap",bootstrap=5000,standardize=TRUE,data=cromley.dat)
summary(fit)
```

```
## lavaan (0.5-23.1097) converged normally after  33 iterations
##
##   Number of observations                           175
##
##   Estimator                                         ML
##   Minimum Function Test Statistic                0.000
##   Degrees of freedom                                 0
##   Minimum Function Value               0.0000000000000
```

```
##
## Parameter Estimates:
##
##   Information                                 Observed
##   Standard Errors                            Bootstrap
##   Number of requested bootstrap draws             5000
##   Number of successful bootstrap draws            5000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   comprehension ~
##     reading    (a)    0.042    0.164    0.260    0.795
##     vocab      (b)    0.275    0.029    9.391    0.000
##     inference  (c)    0.395    0.118    3.330    0.001
##   inference ~
##     reading    (d)    0.641    0.067    9.534    0.000
##     vocab      (e)    0.043    0.015    2.864    0.004
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   reading ~~
##     vocab            11.968    1.625    7.367    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .comprehension     8.006    0.838    9.549    0.000
##    .inference         2.789    0.295    9.447    0.000
##     reading           5.136    0.541    9.489    0.000
##     vocab            85.405    9.067    9.419    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     dc                0.253    0.080    3.160    0.002
##     ec                0.017    0.008    2.072    0.038
##     total             0.587    0.121    4.842    0.000
```

```
parameterEstimates(fit,boot.ci.type = "perc",standardized = TRUE)
```

```
##              lhs op             rhs label    est    se     z pvalue
## 1  comprehension  ~         reading     a  0.042 0.164 0.260  0.795
## 2  comprehension  ~           vocab     b  0.275 0.029 9.391  0.000
## 3  comprehension  ~       inference     c  0.395 0.118 3.330  0.001
## 4      inference  ~         reading     d  0.641 0.067 9.534  0.000
## 5      inference  ~           vocab     e  0.043 0.015 2.864  0.004
## 6        reading ~~           vocab       11.968 1.625 7.367  0.000
## 7  comprehension ~~   comprehension        8.006 0.838 9.549  0.000
## 8      inference ~~       inference        2.789 0.295 9.447  0.000
## 9        reading ~~         reading        5.136 0.541 9.489  0.000
## 10         vocab ~~           vocab       85.405 9.067 9.419  0.000
## 11            dc :=             d*c    dc  0.253 0.080 3.160  0.002
## 12            ec :=             e*c    ec  0.017 0.008 2.072  0.038
## 13         total := a+b+(d*c)+(e*c) total  0.587 0.121 4.842  0.000
##    ci.lower ci.upper std.lv std.all std.nox
## 1    -0.266    0.366  0.042   0.023   0.023
## 2     0.216    0.329  0.275   0.595   0.595
```

```
## 3     0.163    0.629  0.395   0.221   0.221
## 4     0.509    0.770  0.641   0.608   0.608
## 5     0.014    0.072  0.043   0.166   0.166
## 6     8.784   15.235 11.968   0.571   0.571
## 7     6.263    9.559  8.006   0.440   0.440
## 8     2.197    3.343  2.789   0.488   0.488
## 9     4.084    6.222  5.136   1.000   1.000
## 10   68.427  103.745 85.405   1.000   1.000
## 11    0.100    0.417  0.253   0.134   0.134
## 12    0.004    0.035  0.017   0.037   0.037
## 13    0.356    0.831  0.587   0.788   0.788
```