# LMX_mediation_model

Allama Ikbal Sijan

2023-09-22

## Installing and Loading Packages

```
chooseCRANmirror(ind = 1)
install.packages("tidyverse")
install.packages("psych")
install.packages("ggplot2")
install.packages("lavaan")
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(ggplot2)
library(lavaan)
```

```
## This is lavaan 0.6-16
## lavaan is FREE software! Please report any bugs.
##
## Attaching package: 'lavaan'
##
## The following object is masked from 'package:psych':
##
##     cor2cov
```

## Loading the dataset

```
df <- read_csv("GenderMatchData.csv")
```

```
## Rows: 808 Columns: 59
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): otenure_l, ttenure_l, otenure, ttenure
## dbl (55): tmkey, tlkey, office, gender, age, gender_l, office_l, age_l, lc1,...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

## Data Cleaning

```
df <- df[ , c("tmkey", "tlkey", "gender", "age", "gender_l", "age_l",
              "lc1", "lc2", "lc3", "lc4", "lc5", "lc6", "lc7", "lc8",
              "lmx1", "lmx2", "lmx3", "lmx4", "lmx5", "lmx6", "lmx7",
              "as1", "as2", "as3", "as4", "as5", "as6", "as7")]

df <- df %>%
  mutate(mean_lc = (lc1 + lc2 + lc3 + lc4 + lc5 + lc6 + lc7 + lc8) / 8,
         mean_lmx = (lmx1 + lmx2 + lmx3 + lmx4 + lmx5 + lmx6 + lmx7) / 7,
         mean_as = (as1 + as2 + as3 + as4 + as5 + as6 + as7) / 7)

sum(is.na(df$mean_lmx))
```

```
## [1] 223
```

```
sum(is.na(df$mean_as))
```

```
## [1] 223
```

```
sum(is.na(df$mean_lc))
```

```
## [1] 25
```

```
df <- df %>%
  filter(mean_lc != "NA") %>%
  filter(mean_lmx != "NA") %>%
  filter(mean_as != "NA")

df <- df %>%
  mutate(gender_match = case_when((gender == 3 & gender_l == 1) |
                                  (gender == 4 & gender_l == 2) ~ 1, TRUE ~ 0))

# 565 observation after cleaning (final sample size)
```

# Simple modeling (Trial)

```r
model_1 <- lm(mean_as ~ mean_lc, data = df)
summary(model_1)
```

```
##
## Call:
## lm(formula = mean_as ~ mean_lc, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18574 -0.31644  0.02499  0.31070  1.18711
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.24649    0.12803  25.358  < 2e-16 ***
## mean_lc      0.17428    0.03191   5.461  7.1e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.4904 on 563 degrees of freedom
## Multiple R-squared:  0.05031,    Adjusted R-squared:  0.04862
## F-statistic: 29.83 on 1 and 563 DF,  p-value: 7.102e-08
```

```r
model_2 <- lm(mean_as ~ mean_lc + mean_lmx, data = df)
summary(model_2)
```

```
##
## Call:
## lm(formula = mean_as ~ mean_lc + mean_lmx, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.24603 -0.29999  0.02186  0.28623  1.27130
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.67243    0.15148  17.642  < 2e-16 ***
## mean_lc      0.06695    0.03488   1.919   0.0554 .
## mean_lmx     0.24193    0.03696   6.546 1.33e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.4731 on 562 degrees of freedom
## Multiple R-squared:  0.1176, Adjusted R-squared:  0.1145
## F-statistic: 37.45 on 2 and 562 DF,  p-value: 5.403e-16
```

```r
model_3 <- lm(mean_as ~ mean_lc + mean_lmx * gender_match, data = df)
summary(model_3)
```

```
##
```

```
## Call:
## lm(formula = mean_as ~ mean_lc + mean_lmx * gender_match, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22695 -0.28474  0.00477  0.30145  1.25498
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.20185    0.23296  13.744  < 2e-16 ***
## mean_lc                 0.07403    0.03465   2.137  0.03307 *
## mean_lmx                0.11603    0.05879   1.974  0.04890 *
## gender_match           -0.89688    0.28315  -3.168  0.00162 **
## mean_lmx:gender_match   0.20099    0.06892   2.916  0.00369 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.469 on 560 degrees of freedom
## Multiple R-squared:  0.1361, Adjusted R-squared:  0.1299
## F-statistic: 22.05 on 4 and 560 DF,  p-value: < 2.2e-16
```

## Multiple Group Path Model using LAVAAN (Final Model)

```
df <- df %>%
  mutate(gen_mat4 = case_when(
    gender == 3 & gender_l == 1 ~ "MM",
    gender == 4 & gender_l == 2 ~ "FF",
    gender == 3 & gender_l == 2 ~ "MF",
    gender == 4 & gender_l == 1 ~ "FM"
  ))
df$gender_match <- as.numeric(df$gender_match)

df$gen_mat4 <- factor(df$gen_mat4, ordered = TRUE, levels = c("MM", "FF", "MF", "FM"))

MGmodel1 <- '
  # Direct effects
  mean_lmx ~ a*mean_lc
  mean_as ~ mean_lc + b*mean_lmx

  # indirect effects
  indirect := a*b
'
fitMG1 <- sem(MGmodel1, data = df, group = "gen_mat4")
```

```
## Warning in lav_data_full(data = data, group = group, cluster = cluster, : lavaan WARNING: group varia

## Warning in lavaanify(model = FLAT, constraints = constraints, varTable = DataOV, : lavaan WARNING: us
##   setting implies imposing equality constraints across all the groups;
##   If this is not intended, either remove the label(s), or use a vector
##   of labels (one for each group);
##   See the Multiple groups section in the man page of model.syntax.
```

```r
summary(fitMG1,  standardized = TRUE, rsq = TRUE)
```

```
## lavaan 0.6.16 ended normally after 40 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        28
##   Number of equality constraints                     6
##
##   Number of observations per group:
##     FF                                             339
##     MF                                              68
##     FM                                             105
##     MM                                              52
##
## Model Test User Model:
##
##   Test statistic                                11.249
##   Degrees of freedom                                 6
##   P-value (Chi-square)                           0.081
##   Test statistic for each group:
##     FF                                           0.841
##     MF                                           5.389
##     FM                                           4.671
##     MM                                           0.349
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
##
## Group 1 [FF]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000    0.437    0.427
##   mean_as ~
##     mean_lc           0.172    0.045    3.783    0.000    0.172    0.200
##     mean_lmx   (b)    0.235    0.036    6.560    0.000    0.235    0.280
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          2.420    0.144   16.763    0.000    2.420    4.150
##    .mean_as           2.212    0.194   11.371    0.000    2.212    4.517
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          0.278    0.021   13.019    0.000    0.278    0.818
##    .mean_as           0.200    0.015   13.019    0.000    0.200    0.834
##
```

```
## R-Square:
##                    Estimate
##     mean_lmx          0.182
##     mean_as           0.166
##
##
## Group 2 [MF]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000    0.437    0.383
##   mean_as ~
##     mean_lc           0.040    0.095    0.424    0.672    0.040    0.049
##     mean_lmx   (b)    0.235    0.036    6.560    0.000    0.235    0.325
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          2.326    0.153   15.167    0.000    2.326    3.655
##    .mean_as           2.924    0.377    7.747    0.000    2.924    6.355
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          0.345    0.059    5.831    0.000    0.345    0.853
##    .mean_as           0.186    0.032    5.831    0.000    0.186    0.880
##
## R-Square:
##                    Estimate
##     mean_lmx          0.147
##     mean_as           0.120
##
##
## Group 3 [FM]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000    0.437    0.542
##   mean_as ~
##     mean_lc          -0.052    0.063   -0.828    0.408   -0.052   -0.081
##     mean_lmx   (b)    0.235    0.036    6.560    0.000    0.235    0.291
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          2.302    0.144   16.033    0.000    2.302    3.554
##    .mean_as           3.170    0.253   12.516    0.000    3.170    6.052
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          0.296    0.041    7.246    0.000    0.296    0.706
##    .mean_as           0.256    0.035    7.246    0.000    0.256    0.934
##
## R-Square:
##                    Estimate
##     mean_lmx          0.294
##     mean_as           0.066
##
##
## Group 4 [MM]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000    0.437    0.570
##   mean_as ~
##     mean_lc           0.168    0.082    2.050    0.040    0.168    0.256
##     mean_lmx   (b)    0.235    0.036    6.560    0.000    0.235    0.273
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          2.570    0.148   17.383    0.000    2.570    4.418
##    .mean_as           2.570    0.324    7.927    0.000    2.570    5.139
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .mean_lmx          0.228    0.045    5.099    0.000    0.228    0.675
##    .mean_as           0.195    0.038    5.099    0.000    0.195    0.780
##
## R-Square:
##                    Estimate
##     mean_lmx          0.325
##     mean_as           0.220
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     indirect          0.103    0.018    5.808    0.000    0.103    0.119
```

## Checking Model Fit

```
summary(fitMG1, fit.measures = TRUE)
```

```
## lavaan 0.6.16 ended normally after 40 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        28
##   Number of equality constraints                     6
##
##   Number of observations per group:
##     FF                                             339
##     MF                                              68
##     FM                                             105
##     MM                                              52
##
## Model Test User Model:
##
```

```
##   Test statistic                                11.249
##   Degrees of freedom                                 6
##   P-value (Chi-square)                           0.081
##   Test statistic for each group:
##     FF                                           0.841
##     MF                                           5.389
##     FM                                           4.671
##     MM                                           0.349
##
## Model Test Baseline Model:
##
##   Test statistic                               239.099
##   Degrees of freedom                                12
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.977
##   Tucker-Lewis Index (TLI)                       0.954
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)               -801.387
##   Loglikelihood unrestricted model (H1)       -795.762
##
##   Akaike (AIC)                                1646.774
##   Bayesian (BIC)                              1742.145
##   Sample-size adjusted Bayesian (SABIC)       1672.305
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.079
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.149
##   P-value H_0: RMSEA <= 0.050                    0.213
##   P-value H_0: RMSEA >= 0.080                    0.550
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.042
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
##
## Group 1 [FF]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000
##   mean_as ~
##     mean_lc           0.172    0.045    3.783    0.000
##     mean_lmx   (b)    0.235    0.036    6.560    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          2.420    0.144   16.763    0.000
##    .mean_as           2.212    0.194   11.371    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          0.278    0.021   13.019    0.000
##    .mean_as           0.200    0.015   13.019    0.000
##
##
## Group 2 [MF]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000
##   mean_as ~
##     mean_lc           0.040    0.095    0.424    0.672
##     mean_lmx   (b)    0.235    0.036    6.560    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          2.326    0.153   15.167    0.000
##    .mean_as           2.924    0.377    7.747    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          0.345    0.059    5.831    0.000
##    .mean_as           0.186    0.032    5.831    0.000
##
##
## Group 3 [FM]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000
##   mean_as ~
##     mean_lc          -0.052    0.063   -0.828    0.408
##     mean_lmx   (b)    0.235    0.036    6.560    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          2.302    0.144   16.033    0.000
##    .mean_as           3.170    0.253   12.516    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          0.296    0.041    7.246    0.000
##    .mean_as           0.256    0.035    7.246    0.000
##
##
## Group 4 [MM]:
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   mean_lmx ~
##     mean_lc    (a)    0.437    0.035   12.489    0.000
##   mean_as ~
##     mean_lc           0.168    0.082    2.050    0.040
##     mean_lmx   (b)    0.235    0.036    6.560    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          2.570    0.148   17.383    0.000
##    .mean_as           2.570    0.324    7.927    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .mean_lmx          0.228    0.045    5.099    0.000
##    .mean_as           0.195    0.038    5.099    0.000
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     indirect          0.103    0.018    5.808    0.000
```

**The CFI value of 0.977 and TL value of 0.954 suggests that model provides a good fit to the data compared to baseline model. The RMSEA value of 0.079 suggests reasonable fit to the data. The SRMR value of 0.042 suggests the model's residuals are relatively small indicating good model fit**

**In Group 1 (FF), characterized by female coaches and female employees, as well as in Group 4 (MM), where both coach and employee genders are male, our findings demonstrate statistically significant mediation effects ($p < 0.05$). LMX serves as a mediator in the relationship between leadership coaching and adaptive selling for these gender-matched groups.**

**In contrast, Group 2 (MF), where the employee is male and the manager is female, and Group 3 (FM), where the employee is female and the manager is male, reveal distinct patterns. In cases of gender mismatch between coach and employee, the p-values associated with mediation effects notably increase ($p = 0.672$ for Group 2 (MF) and $p = 0.408$ for Group 3 (FM)). These elevated p-values suggest insignificant relationships between leadership coaching and adaptive selling when mediated by LMX in situations of gender mismatch.**

**Thus, the results of the model supports the hypothesis and suggests that LMX mediates the relationship between coaching and adaptive selling, but only when genders match.**