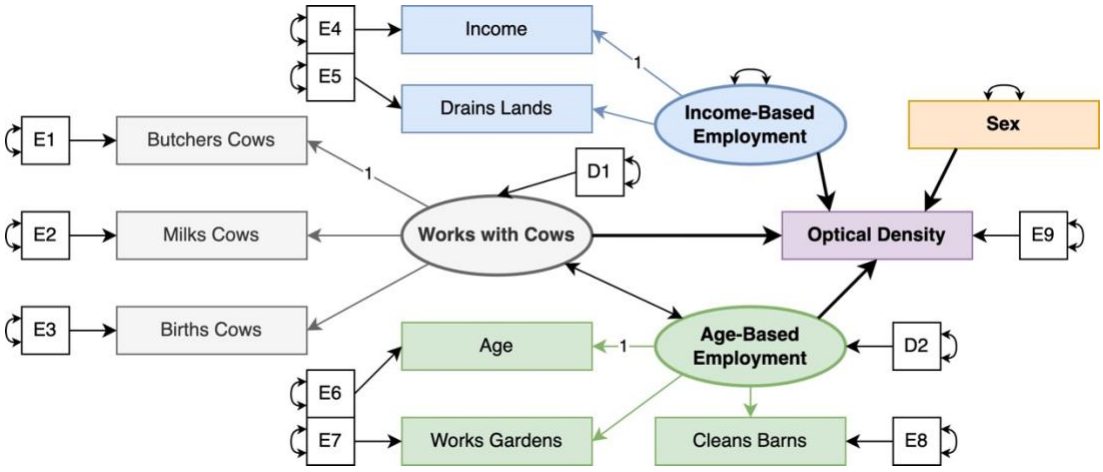
A.3 | Numerical Optimization in Structural Equation Modeling

Having explored how systems of matrix equations can be used to reconstruct the covariability structure among the original dimensions, we can now turn out attention to the mathematical framework of identifying the optimal set of parameter estimates that minimize the difference between this reconstructed covariance matrix and the originally observed covariance matrix. As with other similar analyses, this will rely heavily on numerical optimization. First, however, we need a thorough understanding of how to quantify the difference between these two matrices, something that actually changes depending on certain characteristics of the original data.

A.3.1 | Reconstructing the Covariance Matrix (A Review)

Recall that the reconstructed covariance matrix $\hat{\boldsymbol{\Sigma}}$ represents the variances of and the covariances between each of the original dimensions based solely on the information contained within the path diagram and the corresponding matrices of regression coefficients and variances. The unfixed values in this diagram and these matrices constitute parameters to be estimated such that the resulting reconstructed covariance matrix is as close as possible to the original observed covariance matrix $\boldsymbol{\Sigma}$.

Consider again the path diagram used to represent the covariance structure between the ten dimensions in our example data.



We can represent each of the unfixed pathways in this diagram using one of three parameter matrices.

The matrix of regression coefficients among the DVs $\boldsymbol{\beta}$, with starting values of 0.5, for example, can be constructed using the `as.matrix()` function in R.

```
BT <- matrix(c(  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5,
                 0,  0,  0,  0,  0,  0,  0,  0,  0, .5, .5,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5,
                 0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0),
            nrow = 11, ncol = 11, byrow = T,
            dimnames = list(c("ButchersCows","MilksCows",
                              "BirthsCows","Income","DrainsLands",
                              "Age","WorksGardens","CleansBarns",
                              "OpticalDensity","WorksWithCows",
                              "AgeBasedEmployment"),
                            c("ButchersCows","MilksCows",
                              "BirthsCows","Income","DrainsLands",
                              "Age","WorksGardens","CleansBarns",
                              "OpticalDensity","WorksWithCows",
                              "AgeBasedEmployment")))
```

Recall that any numerical quantity can be used as a starting value for a parameter; 0.5 is just one simple choice.

We can then construct the matrix of regression coefficients between the IVs and DVs $\boldsymbol{\gamma}$ by using a similar approach.

```
GM <- matrix(c(  0,  0, .5,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                 0,  0,  0, .5,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                 0,  0,  0,  0, .5,  0,  0,  0,  0,  0,  0,  0,  0,
                 0,  1,  0,  0,  0, .5,  0,  0,  0,  0,  0,  0,  0,
                 0, .5,  0,  0,  0,  0, .5,  0,  0,  0,  0,  0,  0,
                 0,  0,  0,  0,  0,  0,  0, .5,  0,  0,  0,  0,  0,
                 0,  0,  0,  0,  0,  0,  0,  0, .5,  0,  0,  0,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,  0,  0,
                .5, .5,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,
                 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5),
            nrow = 11, ncol = 13, byrow = T,
            dimnames = list(c("ButchersCows","MilksCows",
                              "BirthsCows","Income","DrainsLands",
                              "Age","WorksGardens","CleansBarns",
                              "OpticalDensity","WorksWithCows",
                              "AgeBasedEmployment"),
                            c("Sex","IncomeBasedEmployment",
                              "E1","E2","E3","E4","E5","E6",
                              "E7","E8","E9","D1","D2")))
```

Finally, we can construct the matrix of variances among the IVs $\boldsymbol{\Phi}$, which together with the two matrices of regression coefficients constructed above, should represent every fixed and unfixed pathway in our original path diagram.

```
PH <- matrix(c( .5,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0, .5,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0, .5,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0, .5,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0, .5,  0,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0, .5,  0,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0, .5,  0,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0, .5,  0,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0, .5,  0,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5,  0,
                0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, .5),
            nrow = 13, ncol = 13, byrow = T,
            dimnames = list(c("Sex","IncomeBasedEmployment",
                              "E1","E2","E3","E4","E5","E6",
                              "E7","E8","E9","D1","D2"),
                            c("Sex","IncomeBasedEmployment",
                              "E1","E2","E3","E4","E5","E6",
                              "E7","E8","E9","D1","D2")))
```

Before using these matrices to reconstruct the covariance matrix, we need to also define the selection matrix for the dependent variables $\boldsymbol{G_y}$ and the selection matrix for the independent variables $\boldsymbol{G_x}$ in R.

```
Gy <- matrix(0, nrow = 9, ncol = 11,
             dimnames = list(c("ButchersCows","MilksCows",
                               "BirthsCows","Income","DrainsLands",
                               "Age","WorksGardens","CleansBarns",
                               "OpticalDensity"),
                             c("ButchersCows","MilksCows",
                               "BirthsCows","Income","DrainsLands",
                               "Age","WorksGardens","CleansBarns",
                               "OpticalDensity","WorksWithCows",
                               "AgeBasedEmployment")))
diag(Gy) <- 1

Gx <- matrix(0, nrow = 1, ncol = 13,
             dimnames = list(c("Sex"),
                             c("Sex","IncomeBasedEmployment",
                               "E1","E2","E3","E4","E5","E6",
                               "E7","E8","E9","D1","D2")))
diag(Gx) <- 1
```

Finally, we need to define the identity matrix $\boldsymbol{I}$ that is the same size as the matrix of regression coefficients among the dependent variables $\boldsymbol{\beta}$.

```
I <- diag(nrow(BT))
```

We can now use a series of matrix equations to reconstruct each quadrant of the covariance matrix, and then bind these quadrants together into the reconstructed covariance matrix $\widehat{\boldsymbol{\Sigma}}$.

```
SIG_yy <- Gy %*% solve(I - BT) %*% GM %*% PH %*%
          t(GM) %*% t(solve(I - BT)) %*% t(Gy)
SIG_yx <- Gy %*% solve(I - BT) %*% GM %*% PH %*% t(Gx)
SIG_xy <- t(SIG_yx)
SIG_xx <- Gx %*% PH %*% t(Gx)

SIG_hat <- rbind(cbind(SIG_yy, SIG_yx),
                 cbind(SIG_xy, SIG_xx))

SIG_hat
              ButchersCows  MilksCows BirthsCows Income DrainsLands
ButchersCows     0.4027778 0.13888889 0.13888889  0.000       0.000
MilksCows        0.1388889 0.19444444 0.06944444  0.000       0.000
BirthsCows       0.1388889 0.06944444 0.19444444  0.000       0.000
Income           0.0000000 0.00000000 0.00000000  0.625       0.250
DrainsLands      0.0000000 0.00000000 0.00000000  0.250       0.250
Age              0.2222222 0.11111111 0.11111111  0.000       0.000
WorksGardens     0.1111111 0.05555556 0.05555556  0.000       0.000
CleansBarns      0.1111111 0.05555556 0.05555556  0.000       0.000
OpticalDensity   0.2500000 0.12500000 0.12500000  0.250       0.125
Sex              0.0000000 0.00000000 0.00000000  0.000       0.000
                      Age WorksGardens CleansBarns OpticalDensity  Sex
ButchersCows     0.222222   0.11111111  0.11111111          0.250 0.00
MilksCows        0.111111   0.05555556  0.05555556          0.125 0.00
BirthsCows       0.111111   0.05555556  0.05555556          0.125 0.00
Income           0.000000   0.00000000  0.00000000          0.250 0.00
DrainsLands      0.000000   0.00000000  0.00000000          0.125 0.00
Age              0.402778   0.13888889  0.13888889          0.250 0.00
WorksGardens     0.138889   0.19444444  0.06944444          0.125 0.00
CleansBarns      0.138889   0.06944444  0.19444444          0.125 0.00
OpticalDensity   0.250000   0.12500000  0.12500000          0.625 0.25
Sex              0.000000   0.00000000  0.00000000          0.250 0.50
```

Given that the goal of structural equation models is to find the set of parameter estimates that best recreates the original covariance structure, the next step is to compare this reconstructed covariance matrix $\widehat{\boldsymbol{\Sigma}}$ to the original observed covariance matrix $\boldsymbol{\Sigma}$.

A.3.2 | Quantifying the Difference in the Covariance Matrices

The most common approach to quantify the difference between the observed and reconstructed covariance matrix, which was introduced in Section 3.3.2, is based on the *maximum likelihood* principle:

$$Q = \log|\widehat{\boldsymbol{\Sigma}}| - \log|\boldsymbol{\Sigma}| + \text{tr}\left(\boldsymbol{\Sigma} \times \widehat{\boldsymbol{\Sigma}}^{-1}\right) - d \tag{A.7}$$

Recall that $|\boldsymbol{\Sigma}|$ represent the determinant of the observed covariance matrix, the $\text{tr}(\,)$ operator refers to the trace of a matrix, and d refers to the dimensionality of the original data matrix itself.

We can quickly implement this objective function in R, using the reconstructed covariance matrix $\widehat{\boldsymbol{\Sigma}}$ (based on the starting values for the set of parameters defined above).

```
Q <- log(det(SIG_hat)) - log(det(SIG)) +
     sum(diag(SIG %*% solve(SIG_hat))) - nrow(SIG)
Q
[1] 100.595
```

Although this technique is by far the most common, several alternative methods have been proposed to quantify this difference between observed and reconstructed covariance matrices.

One such alternative method is based on using an *unweighted least squares* technique:

$$Q = \frac{1}{2}\text{tr}\left(\left(\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\right)^2\right) \tag{A.8}$$

Much like the maximum likelihood technique described above, the goal is to find the absolute smallest value of this difference and the optimized parameter estimates that correspond to it. Unfortunately, this technique does not usually yield standard errors and does not typically follow a chi-square distribution. As such, it is rarely used in the analysis of real data.

Another approach is to use a using a *generalized least squares* technique:

$$Q = \frac{1}{2}\text{tr}\left(\left(\left(\boldsymbol{\Sigma} - \widehat{\boldsymbol{\Sigma}}\right) \times \boldsymbol{\Sigma}^{-1}\right)^2\right) \tag{A.9}$$

We can quickly implement either of these objective functions in R.

```
Q1 <- 0.5 * sum(diag((SIG - SIG_hat)^2))
Q1
[1] 59.67969

Q2 <- 0.5 * sum(diag(((SIG - SIG_hat) * solve(SIG))^2))
Q2
[1] 4.26324
```

Several other alternative techniques have also been proposed, including those based on *elliptical distribution theory* and an *asymptotically distribution free* principle. Despite the presence of these alternative approaches, the maximum likelihood technique remains a reasonable choice so long as the sample size exceeds 500. When sample sizes are smaller or there is substantial lack of normality among the original dimensions, one or more of these alternative techniques may provide a more robust estimate of model parameters.

A.3.3 | Numerical Optimization of in Orthogonal Factor Rotation

Now that we have a better understanding of how to quantify the difference between the observed and reconstructed covariance matrices, let's turn our attention back to how we can

use numerical optimization to systematically evaluate different sets of parameter estimates (the input values) until the absolute smallest measure of this difference (the objective function) is achieved.

In order to demonstrate this computational approach, we first need to define an appropriate function in R to return the difference between the observed and reconstructed covariance matrix based on a concatenated set of parameter estimates. Using, for instance the matrix representation of the parameters defined in the path diagram of our example data, this function should update each of the 34 unfixed parameters in the two matrices of regression coefficients and the one matrix of variances before calculating the corresponding difference.

```
parameterizedDifference <- function(coeff = rep(0,34)) {
   BT["MilksCows","WorksWithCows"] <- coeff[1]
   BT["BirthsCows","WorksWithCows"] <- coeff[2]
   BT["OpticalDensity","WorksWithCows"] <- coeff[3]
   BT["AgeBasedEmployment","WorksWithCows"] <- coeff[4]
   BT["WorksGardens","AgeBasedEmployment"] <- coeff[5]
   BT["CleansBarns","AgeBasedEmployment"] <- coeff[6]
   BT["OpticalDensity","AgeBasedEmployment"] <- coeff[7]
   BT["WorksWithCows","AgeBasedEmployment"] <- coeff[4]

   GM["OpticalDensity","Sex"] <- coeff[8]
   GM["DrainsLands","IncomeBasedEmployment"] <- coeff[9]
   GM["OpticalDensity","IncomeBasedEmployment"] <- coeff[10]
   GM["ButchersCows","E1"] <- coeff[11]
   GM["MilksCows","E2"] <- coeff[12]
   GM["BirthsCows","E3"] <- coeff[13]
   GM["Income","E4"] <- coeff[14]
   GM["DrainsLands","E5"] <- coeff[15]
   GM["Age","E6"] <- coeff[16]
   GM["WorksGardens","E7"] <- coeff[17]
   GM["CleansBarns","E8"] <- coeff[18]
   GM["OpticalDensity","E9"] <- coeff[19]
   GM["WorksWithCows","D1"] <- coeff[20]
   GM["AgeBasedEmployment","D2"] <- coeff[21]

   diag(PH) <- coeff[22:34]

   SIG_yy <- Gy %*% solve(I - BT) %*% GM %*% PH %*%
          t(GM) %*% t(solve(I - BT)) %*% t(Gy)
   SIG_yx <- Gy %*% solve(I - BT) %*% GM %*% PH %*% t(Gx)
   SIG_xy <- t(SIG_yx)
   SIG_xx <- Gx %*% PH %*% t(Gx)

   SIG_hat <- rbind(cbind(SIG_yy, SIG_yx),
                    cbind(SIG_xy, SIG_xx))

   Q <- 0.5 * sum(diag(((SIG - SIG_hat) * solve(SIG))^2))
   return(Q)
}
```

Note that because there is a single oblique path between AgeBasedEmployment and WorksWithCows, both the elements corresponding to this pathway in the parameter matrices

are set to the same value (as highlighted above). Note also that we have used the generalized least squares technique for the purposes of this illustration.

We can now use this function to compare different sets of 34 parameter estimates in terms of which set results in a smaller difference between the observed and reconstructed covariance matrices. The question then becomes how we can find the set of parameter estimates that provides the absolute smallest difference possible.

In R, we can again use the psoptim() function in the pso package to perform particle swarm optimization to find this optimal solution.

```
library(pso)
psoptim(par = rep(0.5, 34), fn = parameterizedDifference,
        lower = 0.1, upper = 0.9, control = list(maxit = 1000))
$par
 [1] 0.8442101 0.9000000 0.2427981 0.4694333 0.8805888 0.3055150
 [7] 0.2437345 0.4497085 0.9000000 0.7414661 0.8831524 0.5882253
[13] 0.9000000 0.7806639 0.9000000 0.1000004 0.7525026 0.7711479
[19] 0.1140184 0.9000000 0.1000002 0.9000000 0.8999999 0.4362002
[25] 0.1011386 0.9000000 0.7054989 0.8999996 0.1000008 0.8543083
[31] 0.8999437 0.7651818 0.9000000 0.1000244

$value
[1] 0.8690069

$counts
 function iteration  restarts
    21000      1000         0

$convergence
[1] 2

$message
[1] "Maximal number of iterations reached"
```

Recall that the $par aspect of the output identifies the set of parameter values that the algorithm determined would produce the absolute smallest function result, which in turn is contained in $value. Using our example output, we can see that this set of parameter estimates produces a difference of approximately 0.87, which is a substantial improvement over the difference of 4.26 produced by our unoptimized starting values.

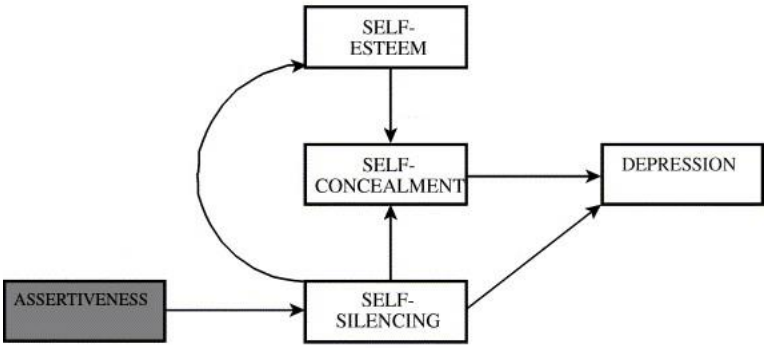
Note that, for the purposes of this illustration, we have set the lower and upper arguments to a very narrow range of possible coefficient estimates. A proper optimization would use a much wider range of possible values for each parameter, but would also incur substantially higher computational time, require a much higher number of iterations (as denoted by the maxit argument), and would likely involve additional troubleshooting of errors. As such, functions such as sem() provide a much simpler means of identifying optimal sets of parameter estimates. The underlying optimization that they rely on, however, is exactly the same as that demonstrated here.

Exercises for Section A.3

A.05 Self-Silencing and Depression. Studies consistently show that women report higher incidence rates of depression compared to men. An article published in 2005 in *Personality and Individual Differences* investigated how three personality and sociocultural variables mediate the documented relationship between self-silencing and depression in both men and women.

	Self-Esteem	Self-Concealment	Self-Silencing	Depression	Assertiveness
Self-Esteem	7.73	4.14	6.36	5.24	-1.60
Self-Concealment	4.14	86.49	80.54	27.42	-17.36
Self-Silencing	6.36	80.54	267.00	46.84	-79.78
Depression	5.24	27.42	46.84	67.08	-12.94
Assertiveness	-1.60	-17.36	-79.78	-12.94	206.21

The covariance matrix above was calculated using data on $N = 598$ women from an introductory psychology class at the University of Windsor in Southwestern Ontario, Canada.



The path diagram above represents the structural framework for this analysis.

a. Construct the covariance matrix in R with the proper row and column names
b. Use the path diagram to construct the matrix of regression coefficients among the dependent variables, including any hidden variables, using values of `NA` for every unfixed parameter
c. Use the path diagram to construct the matrix of regression coefficients between the independent and the dependent variables, including any hidden variables, using values of `NA` for every unfixed parameter
d. Use the path diagram to construct the matrix of variances among the independent variables, including any hidden variables,, fixing each of the variances to a value of `1.0`
e. Construct the selection matrix for the dependent variables and the selection matrix for the independent variables
f. Construct an identity matrix that is the same size as the matrix of regression coefficients among the dependent variables
g. Use `function()` to create a function that returns the difference using the maximum likelihood technique based on a concatenated set of ten regression coefficients

h. Use particle swarm optimization via `psoptim()` to find the best set of regression coefficients to minimize this objective function, using starting values of `1.0` for each coefficient with a lower limit of `-20` and an upper limit of `20`.
i. Find and interpret the effect of `Assertiveness` on `Self-Silencing`
j. Explain if `Self-Concealment` or `Self-Silencing` had a stronger direct effect on `Depression`