Reticular Action Model (RAM) Notation Notes

Ivan Jacob Agaloos Pesigan

2021-01-19

Contents

Chapter 1

Description

This is a collection of my personal notes on the Reticular Action Model (RAM) notation that accompanies the `ramR` package (Pesigan, 2021). You can install the released version of `ramR` from GitHub with:

```
remotes::install_github("jeksterslab/ramR")
```

These notes are based on the following resources:

- Boker and McArdle (2005)
- McArdle and McDonald (1984)
- McArdle (2005)

See GitHub Pages for the html deployment.

Chapter 2

Reticular Action Model (RAM) Matrix Notation

2.1 Full Model

Definition 2.1.

$$\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u} \tag{2.1}$$

where

- $\mathbf{v}$ and $\mathbf{u}$ are $t \times 1$ vectors of random variables
- $\mathbf{A}$ is a $t \times t$ matrix of *directed* or *asymmetric* relationship from column variable v_j to row variable v_i
  - $\mathbf{A}$ represent the regression of each of the t variables $\mathbf{v}$ on the other $t-1$ variables
  - diagonal $a_{i,i}$ is zero
  - u_i represent the residual of v_i
  - if all regression coefficients on other variables are zero, then the variable v_i is considered the same as its own residual u_i

Definition 2.2.

$$\mathbf{S} = \mathbb{E}\left\{\mathbf{u}\mathbf{u}'\right\}, \tag{2.2}$$

where

- $\mathbf{S}$ is a $t \times t$ matrix of *undirected* or *symmetric* relationship
  - the notation Ω is used in other sources for $\mathbf{S}$
- $\mathbb{E}$ is the expectation operator

Definition 2.3.

$$\mathbf{C} = \mathbb{E}\left\{\mathbf{v}\mathbf{v}'\right\}, \tag{2.3}$$

where

- $\mathbf{C}$ is a $t \times t$ variance-covariance matrix
  - the notation Σ is used in other sources for $\mathbf{C}$

Definition 2.4.

$$\mathbf{v} = \mathbf{A} \mathbf{v} + \mathbf{u}$$

can be rewritten as

$$\begin{aligned} \mathbf{v} - \mathbf{A} \mathbf{v} &= \mathbf{u} \\ \mathbf{u} &= \mathbf{v} - \mathbf{A} \mathbf{v} \\ \mathbf{u} &= \left(\mathbf{I} - \mathbf{A} \right) \mathbf{v} \end{aligned} \tag{2.4}$$

assuming that $\left(\mathbf{I} - \mathbf{A} \right)$ is non-singular,

$$\mathbf{E} = \left(\mathbf{I} - \mathbf{A} \right)^{-1} \tag{2.5}$$

then

$$\begin{aligned} \mathbf{v} &= \left(\mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{u} \\ &= \mathbf{E} \mathbf{u} . \end{aligned} \tag{2.6}$$

Using the definitions above, $\mathbf{S}$ and $\mathbf{C}$ are given by

$$\begin{aligned} \mathbf{S} &= \left(\mathbf{I} - \mathbf{A} \right) \mathbf{C} \left(\mathbf{I} - \mathbf{A} \right)^{-1} \\ &= \mathbf{E}^{-1} \mathbf{C} \left(\mathbf{E}^{-1} \right)^{\mathsf{T}} \end{aligned} \tag{2.7}$$

$$\begin{aligned} \mathbf{C} &= \left(\mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{S} \left[\left(\mathbf{I} - \mathbf{A} \right)^{-1} \right]^{\mathsf{T}} \\ &= \mathbf{E} \mathbf{S} \mathbf{E}^{\mathsf{T}} \end{aligned} \tag{2.8}$$

2.2 Observed/Manifest/Given Variables vs. Unobserved/Latent/Hidden Variables

Definition 2.5.

$$\mathbf{v} = \begin{bmatrix} \mathbf{g}_{p \times 1} \\ \mathbf{h}_{q \times 1} \end{bmatrix} \tag{2.9}$$

$$t = p + q \tag{2.10}$$

- $\mathbf{g}$ may be considered observed, manifest or *given* variables
- $\mathbf{h}$ may be considered unobserved, latent, or *hidden* variables

Definition 2.6.

$$\mathbf{F} = \left[\mathbf{I}_{p \times p} : \mathbf{0}_{p \times q} \right] \tag{2.11}$$

- the $\mathbf{F}$ matrix acts as a *filter* to select the manifest variables out of the full set of manifest and latent variables

$$\mathbf{g} = \mathbf{F}\mathbf{v} \tag{2.12}$$

$$\begin{aligned} \mathbf{g} &= \mathbf{F}\left(\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{u} \\ &= \mathbf{F}\mathbf{E}\mathbf{u} \end{aligned} \tag{2.13}$$

Definition 2.7.

$$\mathbf{M} = \mathbb{E}\left\{\mathbf{g}\mathbf{g}^{\mathsf{T}}\right\} \tag{2.14}$$

$$\begin{aligned} \mathbf{M} &= \mathbf{F}\left(\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{S}\left[\left(\mathbf{I} - \mathbf{A}\right)^{-1}\right]^{\mathsf{T}}\mathbf{F}^{\mathsf{T}} \\ &= \mathbf{F}\mathbf{E}\mathbf{S}\mathbf{E}^{\mathsf{T}}\mathbf{F}^{\mathsf{T}} \\ &= \mathbf{F}\mathbf{C}\mathbf{F}^{\mathsf{T}} \end{aligned} \tag{2.15}$$

- when components of $\mathbf{v}$ are permuted, the columns of $\mathbf{F}$ can be correspondingly permuted
- the rows and columns of $\mathbf{C}$ that are filtered out by $\mathbf{F}$ contain useful information about the latent variable structure.

The equations above completely define RAM.

Chapter 3

Reticular Action Model (RAM) Path Diagram

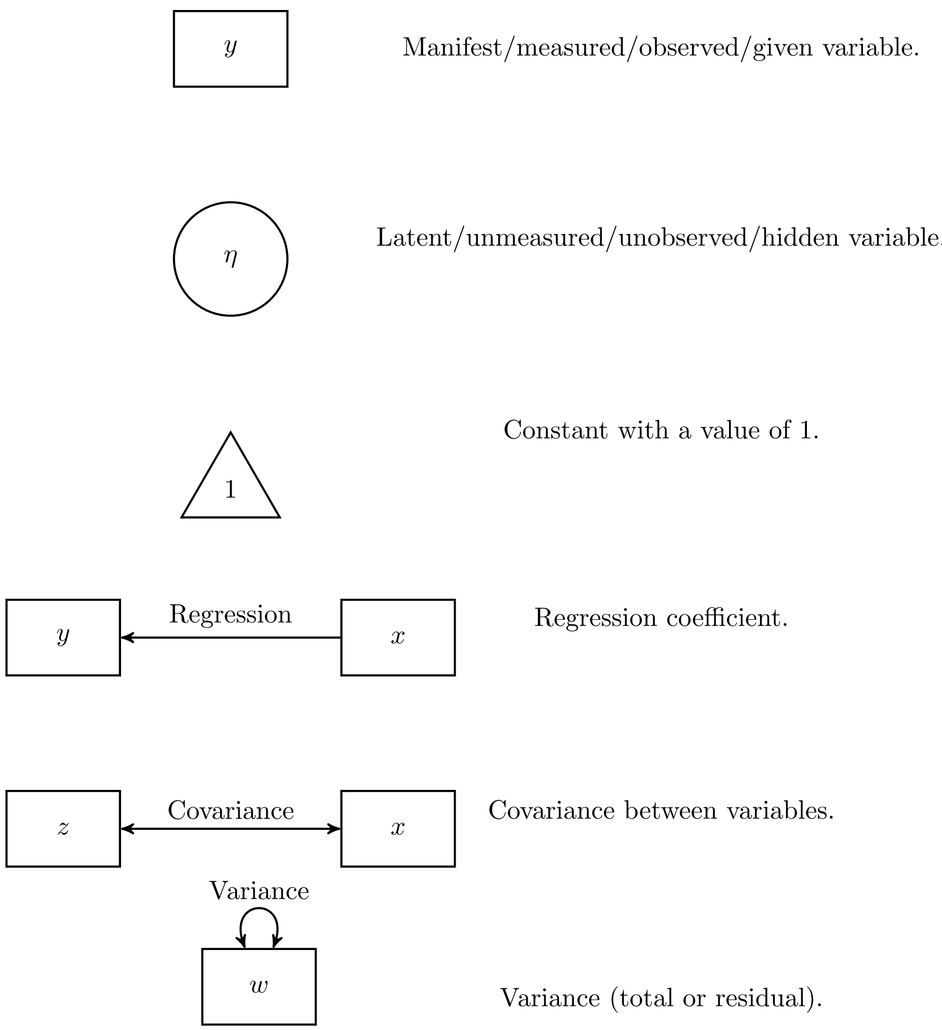
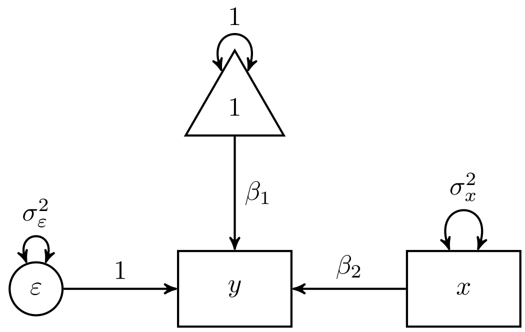


Figure 3.1: Path Diagram Elements



$$y = \alpha + \beta x + \varepsilon$$

Figure 3.2: Two-Variable Regression Model

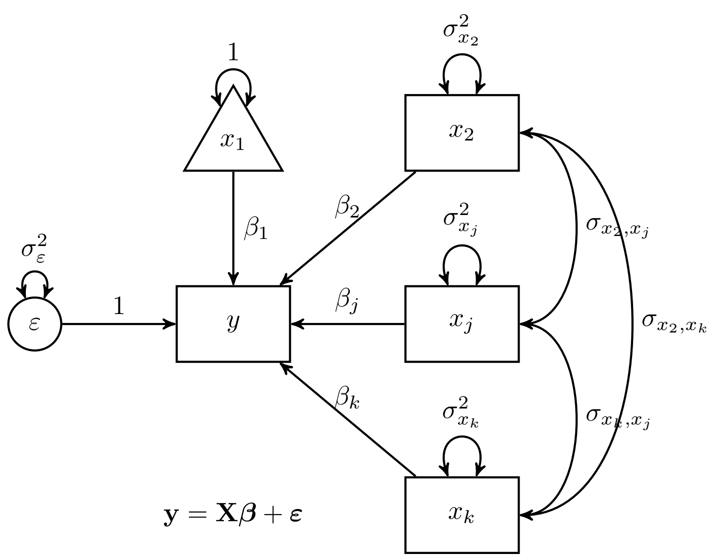


Figure 3.3: k-Variable Regression Model

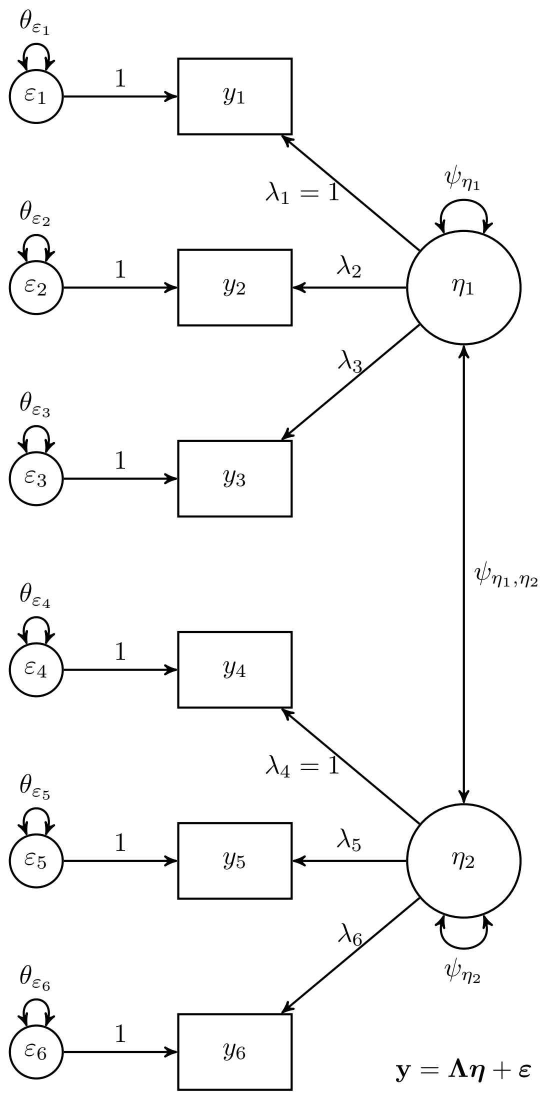


Figure 3.4: Two-Factor Confirmatory Factor Analysis Model

Chapter 4

Student's t-test

In this section, the Student's t-test is presented as a structural equation model using the RAM notation. Let y be a continuous dependent variable, x be a dichotomous independent variable ($x = \{0, 1\}$), and ε be the stochastic error term with mean 0 and constant variance of σ^2_ε across the values of x. The associations of the variables are given by

$$y = \alpha + \beta x + \varepsilon$$

where

- α is the expected value of y when $x = 0$
- β is the unit change in y for unit change in x
- $\alpha + \beta$ is the expected value of y when $x = 1$

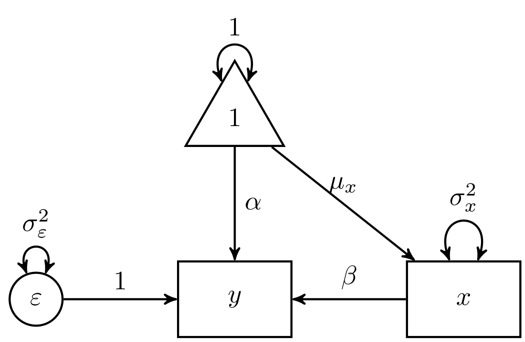


Figure 4.1: Student's t-test

4.1 Symbolic

Let $\{y, x, \varepsilon\}$ be the variables of interest.

$$\mathbf{A} = \begin{pmatrix} 0 & \beta & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sigma_x^2 & 0 \\ 0 & 0 & \sigma_\varepsilon^2 \end{pmatrix}$$

$$\begin{aligned}
\mathbf{C} &= (\mathbf{I}-\mathbf{A})^{-1}\,\mathbf{S}\left[(\mathbf{I}-\mathbf{A})^{-1}\right]^{\mathsf{T}} \\
&= \mathbf{E}\mathbf{S}\mathbf{E}^{\mathsf{T}} \\
&= \begin{pmatrix} 1 & \beta & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sigma_x^2 & 0 \\ 0 & 0 & \sigma_\varepsilon^2 \end{pmatrix} \begin{pmatrix} 1 & \beta & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}} \\
&= \begin{pmatrix} \sigma_x^2\beta^2+\sigma_\varepsilon^2 & \beta\sigma_x^2 & \sigma_\varepsilon^2 \\ \sigma_x^2\beta & \sigma_x^2 & 0 \\ \sigma_\varepsilon^2 & 0 & \sigma_\varepsilon^2 \end{pmatrix}
\end{aligned}$$

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{aligned}
\mathbf{M} &= \mathbf{F}\,(\mathbf{I}-\mathbf{A})^{-1}\,\mathbf{S}\left[(\mathbf{I}-\mathbf{A})^{-1}\right]^{\mathsf{T}}\mathbf{F}^{\mathsf{T}} \\
&= \mathbf{F}\mathbf{E}\mathbf{S}\mathbf{E}^{\mathsf{T}}\mathbf{F}^{\mathsf{T}} \\
&= \mathbf{F}\mathbf{C}\mathbf{F}^{\mathsf{T}} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \sigma_x^2\beta^2+\sigma_\varepsilon^2 & \beta\sigma_x^2 & \sigma_\varepsilon^2 \\ \sigma_x^2\beta & \sigma_x^2 & 0 \\ \sigma_\varepsilon^2 & 0 & \sigma_\varepsilon^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{\mathsf{T}} \\
&= \begin{pmatrix} \sigma_x^2\beta^2+\sigma_\varepsilon^2 & \beta\sigma_x^2 \\ \sigma_x^2\beta & \sigma_x^2 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
\mathbf{v} &= (\mathbf{I}-\mathbf{A})^{-1}\,\mathbf{u} \\
&= \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & \beta & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right]^{-1} \begin{pmatrix} \alpha \\ \mu_x \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \alpha+\beta\mu_x \\ \mu_x \\ 0 \end{pmatrix}
\end{aligned}$$

$$
\begin{aligned}
\mathbf{u} &= \left(\mathbf{I} - \mathbf{A} \right) \mathbf{v} \\
&= \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & \beta & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right] \begin{pmatrix} \alpha + \beta \mu_x \\ \mu_x \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \alpha \\ \mu_x \\ 0 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{g} &= \mathbf{F} \left(\mathbf{I} - \mathbf{A} \right)^{-1} \mathbf{u} \\
&= \left[\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & \beta & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right]^{-1} \begin{pmatrix} \alpha \\ \mu_x \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \alpha + \beta \mu_x \\ \mu_x \end{pmatrix}
\end{aligned}
$$

4.1.1 Using the `ramR` Package

```
A
```

```
##      [,1] [,2]   [,3]
## [1,] "0"  "beta" "1"
## [2,] "0"  "0"    "0"
## [3,] "0"  "0"    "0"
```

```
S
```

```
##      [,1] [,2]         [,3]
## [1,] "0"  "0"          "0"
## [2,] "0"  "sigma[x]^2" "0"
## [3,] "0"  "0"          "sigma[varepsilon]^2"
```

```
u
```

```
##      [,1]
## [1,] "alpha"
## [2,] "mu[x]"
## [3,] "0"
```

```
filter
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
```

The covariance expectations can be symbolically derived using the `ramR::C_sym()` function.

```
ramR::C_sym(A, S)
```

```
## {{sigma[x]^2*beta^2+sigma[varepsilon]^2,                         beta*sigma[x]^2,                sig
##  {                       sigma[x]^2*beta,                             sigma[x]^2,
##  {                    sigma[varepsilon]^2,                                     0,                sig
```

$$\mathbf{C} = \begin{pmatrix} \sigma_x^2\beta^2 + \sigma_\varepsilon^2 & \beta\sigma_x^2 & \sigma_\varepsilon^2 \\ \sigma_x^2\beta & \sigma_x^2 & 0 \\ \sigma_\varepsilon^2 & 0 & \sigma_\varepsilon^2 \end{pmatrix}$$

The covariance expectations for the observed variables can be symbolically derived using the `ramR::M_sym()` function.

```
ramR::M_sym(A, S, filter)
```

```
## {{sigma[x]^2*beta^2+sigma[varepsilon]^2,                         beta*sigma[x]^2},
##  {                       sigma[x]^2*beta,                             sigma[x]^2}}
```

$$\mathbf{M} = \begin{pmatrix} \sigma_x^2\beta^2 + \sigma_\varepsilon^2 & \beta\sigma_x^2 \\ \sigma_x^2\beta & \sigma_x^2 \end{pmatrix}$$

The mean expectations can be symbolically derived using the `ramR::v_sym()` function.

```
ramR::v_sym(A, u)
```

```
## {{alpha+beta*mu[x]},
##  {           mu[x]},
##  {               0}}
```

$$\mathbf{v} = \begin{pmatrix} \alpha + \beta\mu_x \\ \mu_x \\ 0 \end{pmatrix}$$

The mean expectations for the observed variables can be symbolically derived using the `ramR::g_sym()` function.

```
ramR::g_sym(A, u, filter)
```

```
## {{alpha+beta*mu[x]},
##  {           mu[x]}}
```

$$\mathbf{g} = \begin{pmatrix} \alpha + \beta\mu_x \\ \mu_x \end{pmatrix}$$

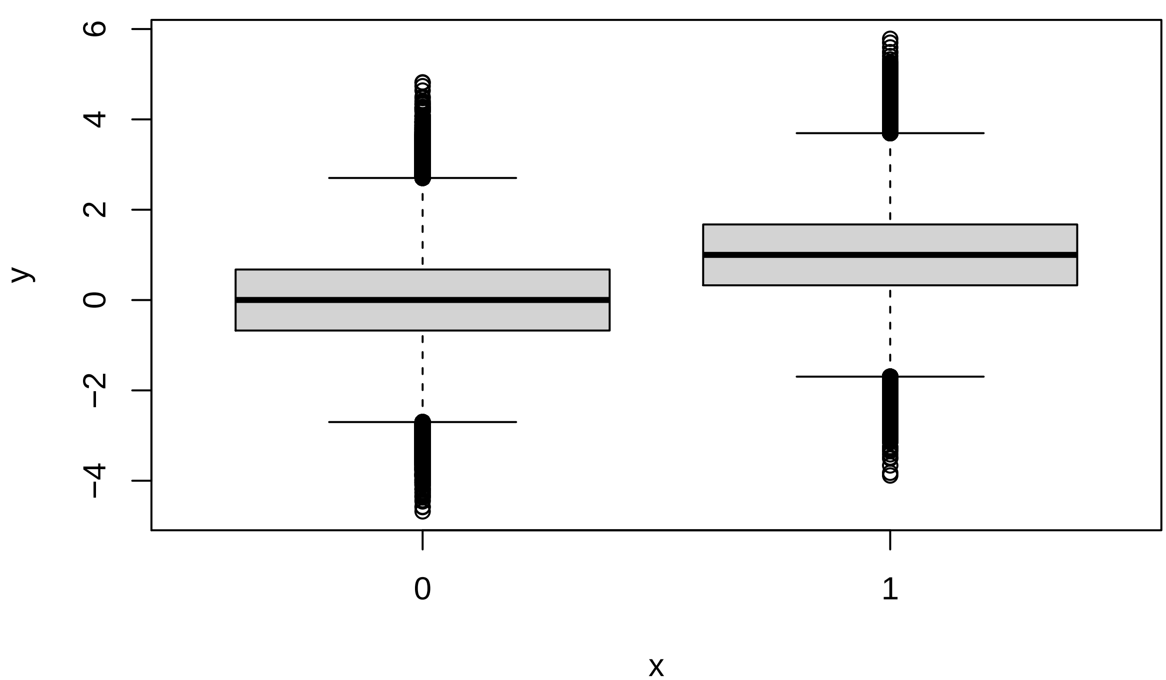
4.2 Numerical Example

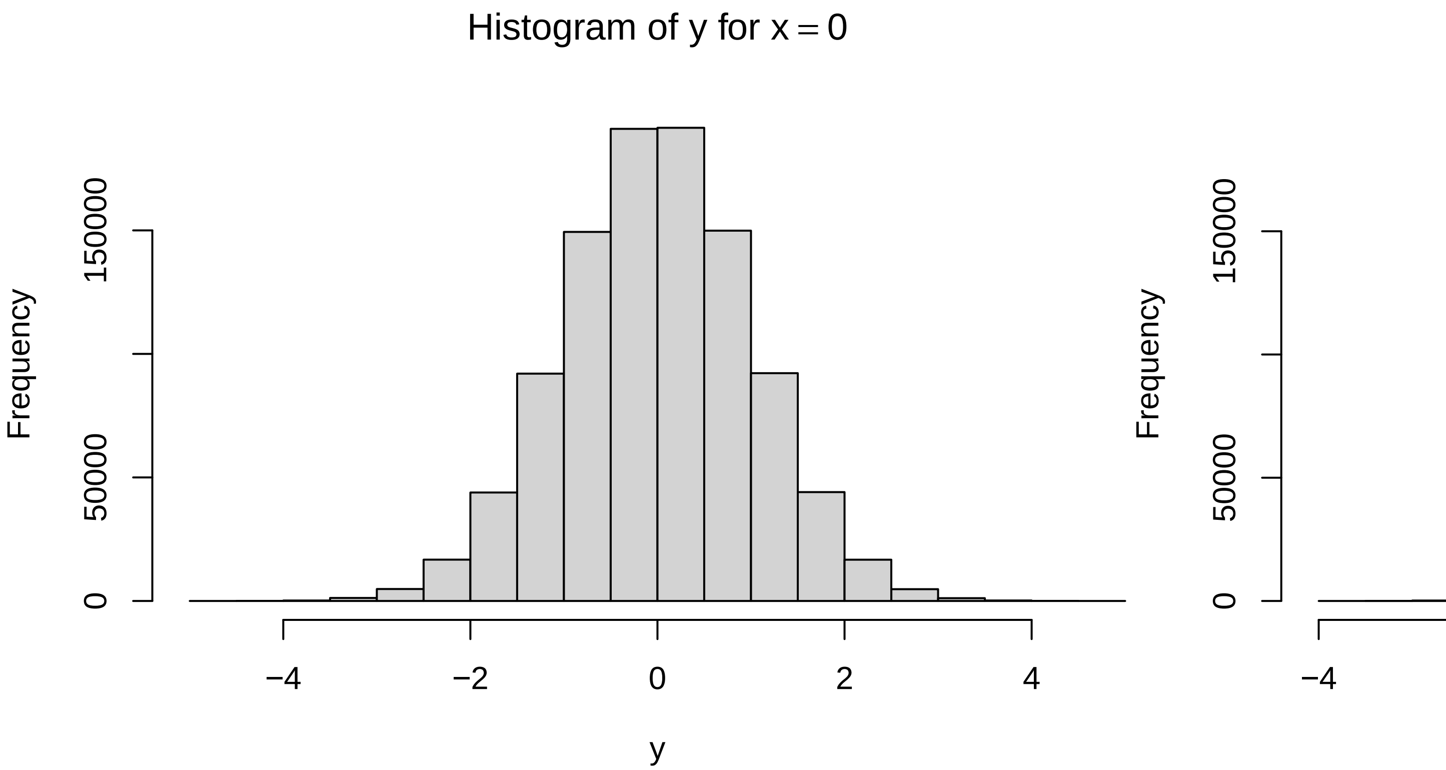
```
head(df)
```

```
##            y x
## 1  1.3709584 0
## 2 -0.5646982 0
## 3  0.3631284 0
## 4  0.6328626 0
## 5  0.4042683 0
## 6 -0.1061245 0
```

```
summary(df)
```

```
##        y                  x
##  Min.   :-4.6785   Min.   :0.0
##  1st Qu.:-0.2622   1st Qu.:0.0
##  Median : 0.5013   Median :0.5
##  Mean   : 0.5000   Mean   :0.5
##  3rd Qu.: 1.2618   3rd Qu.:1.0
##  Max.   : 5.7839   Max.   :1.0
```





4.2.1 t-test

```r
t.test <- t.test(y ~ x, data = df)
t.test
```

```
##
##  Welch Two Sample t-test
##
## data:  y by x
## t = -706.06, df = 2e+06, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.0016565 -0.9961108
## sample estimates:
## mean in group 0 mean in group 1
##    0.0005737398    0.9994574009
```

```r
t.test$estimate
```

```
## mean in group 0 mean in group 1
##    0.0005737398    0.9994574009
```

4.2.2 Linear Regression

```r
lm <- lm(y ~ x, data = df)
summary(lm)
```

```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8838 -0.6745  0.0005  0.6749  4.8195
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0005737  0.0010004   0.574    0.566
## x           0.9988837  0.0014147 706.057   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1 on 1999998 degrees of freedom
## Multiple R-squared:  0.1995, Adjusted R-squared:  0.1995
## F-statistic: 4.985e+05 on 1 and 1999998 DF,  p-value: < 2.2e-16
```

```r
coef(lm)
```

```
##  (Intercept)            x
## 0.0005737398 0.9988836611
```

4.2.3 Structural Equation Modeling

```r
model <- "
  y ~ x
  y ~ 1
  x ~ 1
"
fit <- lavaan::sem(model, data = df)
lavaan::summary(fit)
```

```
## lavaan 0.6-7 ended normally after 15 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                          5
##
##   Number of observations                       2000000
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   y ~
##     x                 0.999    0.001  706.057    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .y                 0.001    0.001    0.574    0.566
##     x                 0.500    0.000 1414.214    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .y                 1.001    0.001 1000.000    0.000
##     x                 0.250    0.000 1000.000    0.000
```

```
lavaan::coef(fit)
```

```
##   y~x   y~1   x~1  y~~y  x~~x
## 0.999 0.001 0.500 1.001 0.250
```

label	parameter
α	0
β	1
σ^{2}_{x}	0.25
σ^{2}_{ε}	0.25
μ_x	0.5

4.2.4 Using the `ramR` Package

```
A
```

```
##   y         x e
## y 0 0.9988837 1
## x 0 0.0000000 0
## e 0 0.0000000 0
```

```
S
```

```
##   y         x         e
## y 0 0.0000000 0.0000000
## x 0 0.2500001 0.0000000
## e 0 0.0000000 0.2494423
```

```
u
```

```
##             [,1]
## y 0.0005737398
## x 0.5000000000
## e 0.0000000000
```

```
filter
```

```
##   y x e
## y 1 0 0
## x 0 1 0
```

The covariance expectations can be numerically derived using the `ramR::C_num()` function.

```
ramR::C_num(A, S)
```

```
##           y         x         e
## y 0.4988845 0.2497210 0.2494423
## x 0.2497210 0.2500001 0.0000000
## e 0.2494423 0.0000000 0.2494423
```

The covariance expectations for the observed variables can be numerically derived using the `ramR::M_num()` function.

```
ramR::M_num(A, S, filter)
```

```
##           y         x
## y 0.4988845 0.2497210
## x 0.2497210 0.2500001
```

The mean expectations can be numerically derived using the `ramR::v_num()` function.

```
ramR::v_num(A, u)
```

```
##           v
## y 0.5000156
## x 0.5000000
## e 0.0000000
```

The mean expectations for the observed variables can be numerically derived using the `ramR::v_num()` function.

```
ramR::g_num(A, u, filter)
```

```
##           g
## y 0.5000156
## x 0.5000000
```

Bibliography

Boker, S. M. and McArdle, J. J. (2005). Path analysis and path diagrams. In Everitt, B. S. and Howell, D. C., editors, *Encyclopedia of Statistics in Behavioral Science*, pages 1529–1531. John Wiley & Sons, Ltd, Chichester, UK.

McArdle, J. J. (2005). The development of the RAM rules for latent variable structural equation modeling. In Maydeu-Olivares, A. and McArdle, J. J., editors, *Contemporary psychometrics: A festschrift for Roderick P. McDonald*, Multivariate applications book series, pages 225–273. Lawrence Erlbaum Associates, Mahwah, NJ.

McArdle, J. J. and McDonald, R. P. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology*, 37(2):234–251.

Pesigan, I. J. A. (2021). *ramR: Reticular Action Model (RAM) Notation.* R package version 0.9.0.