

Chapter 3: Testing alternative models of data

William Revelle

Northwestern University

Prepared as part of course on latent variable analysis ([Psychology 454](#))
and as a supplement to the [Short Guide to R for psychologists](#)

January 30, 2007

3.1	One factor — congeneric data model	2
3.1.1	Generating the data	2
3.1.2	Estimate a congeneric model	5
3.1.3	Estimate a tau equivalent model with equal true score and unequal error loadings	6
3.1.4	Estimate a parallel test model with equal true score and equal error loadings	8
3.1.5	Estimate a parallel test model with fixed loadings	9
3.1.6	Comparison of models	10
3.2	Two (perhaps correlated) factors	10
3.2.1	Generating the data	11
3.2.2	Exploratory Factor analysis of the data	12
3.2.3	Confirmatory analysis with a predicted structure	14
3.2.4	Confirmatory factor analysis with two independent factors with equal loadings within factors	16
3.2.5	Structure invariance, part I— unequal loadings within factors - matched across factors	18
3.2.6	Estimate two correlated factors	19
3.3	Hierarchical models	27
3.3.1	Two Correlated factors with a g factor	27
3.3.2	Generating the data for 3 correlated factors	30
3.3.3	Exploratory factor analysis with 3 factors	31
3.3.4	Three correlated factors with a g factor	34
3.3.5	Bifactor solutions	36
3.3.6	Schmid Leiman transformations to orthogonalize the factors	38
3.3.7	Omega as an estimate of reliability	40

In this chapter we consider how to test nested alternative models of some basic data types. Using the simulation tools introduced in the previous chapter, we generate a data set from a congeneric reliability model with unequal true score loadings and fit three alternative models to the data. Then we simulate a two factor data structure and consider a set of alternative

models. Finally, we consider ways of representing (and modeling) hierarchical data structures.

For these examples, as well as the other ones, we need to load the `psych` and `sem` packages.

```
> library(sem)
> library(psych)
```

3.1 One factor — congeneric data model

The classic test theory structure of 4 observed variables $V_1 \dots V_4$ all loading on a single factor, θ , (Figure 3.1 may be analyzed in multiple ways. The most restrictive model considers all the loadings to be fixed values (perhaps .7). A more reasonable model is to consider the four variables to be parallel, that is to say, that they have equal loadings on the latent variable and equal error variances. A less restrictive models would be tau equivalence where the paths are equal but the errors can be unequal, and then the least restrictive model is known as the “congeneric” model where all paths are allowed to vary.

We can generate data under a congeneric model and then test it with progressively more restricted models (i.e., start with the most unrestricted model, the congeneric model, fix some parameters for the tau equivalent model, add equality constraints for the parallel test model, and then fit arbitrarily fixed parameters). To do this, we first create a function, `sim.sem`, which we apply to make our data.

3.1.1 Generating the data

We create a function, **`sim.sem`**, to simulate data with a variety of possible structures. Although the function defaults to four variables with specific loadings on one factor, we can vary both the number of variables as well as the loadings and the number of factors. The function returns the pattern matrix used to generate the data and the implied structure matrix, or just the simulated raw data.

```
> sim.sem <- function(N = 1000, loads = c(0.8, 0.7, 0.6, 0.5), phi = NULL, obs = TRUE) {
+   if (!is.matrix.loads)) {
+     loading <- matrix.loads, ncol = 1)
+   }
+   else {
+     loading <- loads
+   }
+   nv <- dim.loading)[1]
+   nf <- dim.loading)[2]
+   error <- diag(1, nrow = nv)
+   diag(error) <- sqrt(1 - diag.loading %*% t.loading)))
+   if (is.null(phi))
+     phi <- diag(1, nrow = nf)
+   pattern <- cbind.loading, error)
```

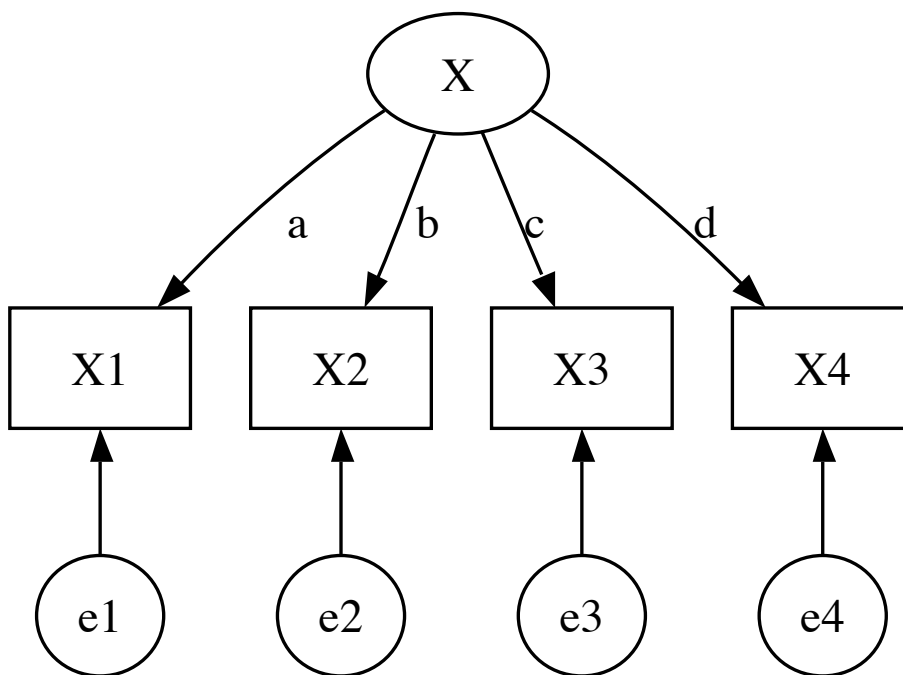


Figure 3.1: The basic congeneric model is one latent (true score) factor accounting for the correlations of multiple observed scores. If there are at least 4 observed variables, the model is identified. For fewer variables, assumptions need to be made (i.e., for two parallel tests, the path coefficients are all equal.)

```

+   colnames(pattern) <- c(paste("theta", seq(1:nf), sep = ""), paste("e", seq(1:nv),
+   sep = ""))
+   rownames(pattern) <- c(paste("V", seq(1:nv), sep = ""))
+   temp <- diag(1, nv + nf)
+   temp[1:nf, 1:nf] <- phi
+   phi <- temp
+   colnames(phi) <- c(paste("theta", seq(1:nf), sep = ""), paste("e", seq(1:nv),
+   sep = ""))
+   structure <- pattern %*% phi
+   latent <- matrix(rnorm(N * (nf + nv)), ncol = (nf + nv))
+   if (nf > 1) {
+     for (i in 1:nf) {
+       for (j in i + 1:nf) {
+         phi[i, j] <- 0
+       }
+     }
+   }
+   observed <- latent %*% t(pattern %*% phi)
+   if (obs) {
+     return(observed)
+   }
+   else {
+     ps <- list(pattern = pattern, structure = structure, phi)
+     return(ps)
+   }
+ }

```

Specifying a particular factor pattern, we can use the **sim.sem** function to show the extended pattern matrix, the implied population correlation matrix, and then take a sample of 1000 from that population. Note that even with 1000 simulated subjects the sample correlation matrix is not the same as the population matrix. As we develop our theory testing skills, it is useful to remember that we are trying to make inferences about the population based upon parameter estimates derived from the sample.

```

> N <- 1000
> sim <- sim.sem(obs = FALSE)
> round(sim$pattern, 2)

```

	theta1	e1	e2	e3	e4
V1	0.8	0.6	0.00	0.0	0.00
V2	0.7	0.0	0.71	0.0	0.00
V3	0.6	0.0	0.00	0.8	0.00
V4	0.5	0.0	0.00	0.0	0.87

```

> population <- (sim$pattern %*% t(sim$pattern))
> population

```

	V1	V2	V3	V4
V1	1.00	0.56	0.48	0.40

```

V2 0.56 1.00 0.42 0.35
V3 0.48 0.42 1.00 0.30
V4 0.40 0.35 0.30 1.00

> set.seed(42)
> data.f1 <- sim.sem()
> round(cor(data.f1), 2)

      V1   V2   V3   V4
V1 1.00 0.54 0.52 0.41
V2 0.54 1.00 0.41 0.32
V3 0.52 0.41 1.00 0.32
V4 0.41 0.32 0.32 1.00

```

3.1.2 Estimate a congeneric model

Using the simulated data generated above, we find the covariance matrix from the sample data and apply sem to the data. (The **sem** package needs to be loaded first.) Examine the statistics of fit as well as the residual matrix.

```

> S.congeneric <- cov(data.f1)
> model.congeneric <- matrix(c("theta -> V1", "a", NA, "theta -> V2", "b", NA,
+   "theta -> V3", "c", NA, "theta -> V4", "d", NA, "V1 <-> V1", "u", NA, "V2 <-> V2",
+   "v", NA, "V3 <-> V3", "w", NA, "V4 <-> V4", "x", NA, "theta <-> theta",
+   NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.congeneric) <- c("path", "label", "initial estimate")
> model.congeneric

      path                label initial estimate
[1,] "theta -> V1"         "a"      NA
[2,] "theta -> V2"         "b"      NA
[3,] "theta -> V3"         "c"      NA
[4,] "theta -> V4"         "d"      NA
[5,] "V1 <-> V1"           "u"      NA
[6,] "V2 <-> V2"           "v"      NA
[7,] "V3 <-> V3"           "w"      NA
[8,] "V4 <-> V4"           "x"      NA
[9,] "theta <-> theta"     NA       "1"

> sem.congeneric = sem(model.congeneric, S.congeneric, N)
> summary(sem.congeneric, digits = 3)

Model Chisquare = 0.46   Df = 2 Pr(>Chisq) = 0.795
Chisquare (null model) = 910   Df = 6
Goodness-of-fit index = 1
Adjusted goodness-of-fit index = 0.999
RMSEA index = 0   90% CI: (NA, 0.0398)
Bentler-Bonnett NFI = 1
Tucker-Lewis NNFI = 1.01

```

```
Bentler CFI = 1
```

```
BIC = -13.4
```

```
Normalized Residuals
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-0.177000	-0.032200	-0.000271	0.010600	0.017000	0.319000

```
Parameter Estimates
```

	Estimate	Std Error	z value	Pr(> z)	
a	0.829	0.0320	25.90	0	V1 <--- theta
b	0.657	0.0325	20.23	0	V2 <--- theta
c	0.632	0.0325	19.43	0	V3 <--- theta
d	0.503	0.0340	14.80	0	V4 <--- theta
u	0.316	0.0346	9.12	0	V1 <--> V1
v	0.580	0.0334	17.35	0	V2 <--> V2
w	0.604	0.0337	17.94	0	V3 <--> V3
x	0.776	0.0382	20.31	0	V4 <--> V4

```
Iterations = 13
```

```
> round(residuals(sem.congeneric), 2)
```

	V1	V2	V3	V4
V1	0	0.00	0.00	0.00
V2	0	0.00	-0.01	0.00
V3	0	-0.01	0.00	0.01
V4	0	0.00	0.01	0.00

3.1.3 Estimate a tau equivalent model with equal true score and unequal error loadings

A more constrained model, “Tau equivalence”, assumes that the theta paths in Figure 3.1 are equal but allows the error variances to be unequal.

```
> S.congeneric <- cov(data.f1)
> model.tau <- matrix(c("theta -> V1", "a", NA, "theta -> V2", "a", NA, "theta -> V3",
+ "a", NA, "theta -> V4", "a", NA, "V1 <-> V1", "u", NA, "V2 <-> V2", "v",
+ NA, "V3 <-> V3", "w", NA, "V4 <-> V4", "x", NA, "theta <-> theta", NA, 1),
+ ncol = 3, byrow = TRUE)
> colnames(model.tau) <- c("path", "label", "initial estimate")
> model.tau
```

	path	label	initial estimate
[1,]	"theta -> V1"	"a"	NA
[2,]	"theta -> V2"	"a"	NA
[3,]	"theta -> V3"	"a"	NA
[4,]	"theta -> V4"	"a"	NA
[5,]	"V1 <-> V1"	"u"	NA

```

[6,] "V2 <-> V2"      "v"    NA
[7,] "V3 <-> V3"      "w"    NA
[8,] "V4 <-> V4"      "x"    NA
[9,] "theta <-> theta" NA     "1"

> sem.tau = sem(model.tau, S.congeneric, N)
> summary(sem.tau, digits = 3)

Model Chisquare = 56.1   Df = 5 Pr(>Chisq) = 7.64e-11
Chisquare (null model) = 910   Df = 6
Goodness-of-fit index = 0.974
Adjusted goodness-of-fit index = 0.947
RMSEA index = 0.101   90% CI: (0.0783, 0.126)
Bentler-Bonnett NFI = 0.938
Tucker-Lewis NNFI = 0.932
Bentler CFI = 0.943
BIC = 21.6

Normalized Residuals
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.160 -2.890 -0.967 -0.418  2.290  3.000

Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
a 0.668      0.0202   33.2    0      V1 <--- theta
u 0.448      0.0270   16.6    0      V1 <--> V1
v 0.565      0.0315   18.0    0      V2 <--> V2
w 0.576      0.0319   18.1    0      V3 <--> V3
x 0.730      0.0386   18.9    0      V4 <--> V4

Iterations = 10

> round(residuals(sem.tau), 2)

      V1    V2    V3    V4
V1  0.11  0.10  0.08 -0.03
V2  0.10  0.00 -0.04 -0.12
V3  0.08 -0.04 -0.02 -0.12
V4 -0.03 -0.12 -0.12 -0.15

```

Note that this model has a much worse fit (as it should), with a very large change in the χ^2 that far exceeds the benefit of greater parsimony (the change in degrees of freedom from 2 to 5). However, note that some traditional measures of fit (e.g., the GFI) seem reasonable. The RMSEA and NFI suggest a poor fit, as do the residuals.

3.1.4 Estimate a parallel test model with equal true score and equal error loadings

An even more unrealistic model would a model of parallel tests where the true score variances are the same for all tests, as are the error variances.

```
> model.parallel <- matrix(c("theta -> V1", "a", NA, "theta -> V2", "a", NA,
+ "theta -> V3", "a", NA, "theta -> V4", "a", NA, "V1 <-> V1", "u", NA, "V2 <-> V2",
+ "u", NA, "V3 <-> V3", "u", NA, "V4 <-> V4", "u", NA, "theta <-> theta",
+ NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.parallel) <- c("path", "label", "initial estimate")
> model.parallel
```

	path	label	initial estimate
[1,]	"theta -> V1"	"a"	NA
[2,]	"theta -> V2"	"a"	NA
[3,]	"theta -> V3"	"a"	NA
[4,]	"theta -> V4"	"a"	NA
[5,]	"V1 <-> V1"	"u"	NA
[6,]	"V2 <-> V2"	"u"	NA
[7,]	"V3 <-> V3"	"u"	NA
[8,]	"V4 <-> V4"	"u"	NA
[9,]	"theta <-> theta"	NA	"1"

```
> sem.parallel = sem(model.parallel, S.congeneric, N)
> summary(sem.parallel, digits = 3)
```

Model Chisquare = 91.2 Df = 8 Pr(>Chisq) = 2.22e-16
Chisquare (null model) = 910 Df = 6
Goodness-of-fit index = 0.959
Adjusted goodness-of-fit index = 0.949
RMSEA index = 0.102 90% CI: (0.0838, 0.121)
Bentler-Bonnett NFI = 0.9
Tucker-Lewis NNFI = 0.931
Bentler CFI = 0.908
BIC = 36.0

Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-2.78e+00	-1.04e+00	-2.74e-01	7.11e-07	9.99e-01	3.51e+00

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z)	
a	0.652	0.0198	32.9	0	V1 <--- theta
u	0.586	0.0152	38.7	0	V1 <--> V1

Iterations = 11

```
> round(residuals(sem.parallel), 2)
```


	V1	V2	V3	V4
V1	-0.01	0.12	0.10	-0.01
V2	0.12	0.00	-0.02	-0.09
V3	0.10	-0.02	-0.01	-0.10
V4	-0.01	-0.09	-0.10	0.02

3.1.5 Estimate a parallel test model with fixed loadings

The most restrictive model estimates the fewest parameters and considers the case where all loadings are fixed at a particular value. (This is truly a stupid model). Notice how large the residuals are.

```
> model.fixed <- matrix(c("theta -> V1", NA, 0.6, "theta -> V2", NA, 0.6, "theta -> V3",
+      NA, 0.6, "theta -> V4", NA, 0.6, "V1 <-> V1", "u", NA, "V2 <-> V2", "u",
+      NA, "V3 <-> V3", "u", NA, "V4 <-> V4", "u", NA, "theta <-> theta", NA, 1),
+      ncol = 3, byrow = TRUE)
> colnames(model.fixed) <- c("path", "label", "initial estimate")
> model.fixed
```

	path	label	initial estimate
[1,]	"theta -> V1"	NA	"0.6"
[2,]	"theta -> V2"	NA	"0.6"
[3,]	"theta -> V3"	NA	"0.6"
[4,]	"theta -> V4"	NA	"0.6"
[5,]	"V1 <-> V1"	"u"	NA
[6,]	"V2 <-> V2"	"u"	NA
[7,]	"V3 <-> V3"	"u"	NA
[8,]	"V4 <-> V4"	"u"	NA
[9,]	"theta <-> theta"	NA	"1"

```
> sem.fixed = sem(model.fixed, S.congeneric, N)
> summary(sem.fixed, digits = 3)

Model Chisquare = 98.6   Df = 9 Pr(>Chisq) = 0
Chisquare (null model) = 910   Df = 6
Goodness-of-fit index = 0.957
Adjusted goodness-of-fit index = 0.952
RMSEA index = 0.0998   90% CI: (0.0826, 0.118)
Bentler-Bonnett NFI = 0.892
Tucker-Lewis NNFI = 0.934
Bentler CFI = 0.9
BIC = 36.4
```

Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-0.980	0.633	1.510	1.850	2.590	5.790

Parameter Estimates

```

      Estimate Std Error z value Pr(>|z|)
u 0.594      0.0153    38.9    0      V1 <--> V1

```

```
Iterations = 8
```

```
> round(residuals(sem.fixed), 2)
```

```

      V1      V2      V3      V4
V1 0.05  0.19  0.16  0.05
V2 0.19  0.06  0.05 -0.03
V3 0.16  0.05  0.05 -0.03
V4 0.05 -0.03 -0.03  0.08

```

3.1.6 Comparison of models

We can examine the degrees of freedom in each of previous analyses and compare them to the goodness of fit. The difference in χ^2 of a nested model is itself a χ^2 with degrees of freedom = the difference of the two different degrees of freedom. We form a list of the different analyses, and then show the summary statistics.

```

> summary.list <- list()
> summary.list[[1]] <- summary(sem.congeneric)[1:2]
> summary.list[[2]] <- summary(sem.tau)[1:2]
> summary.list[[3]] <- summary(sem.parallel)[1:2]
> summary.list[[4]] <- summary(sem.fixed)[1:2]
> summary.data <- matrix(unlist(summary.list), nrow = 4, byrow = TRUE)
> rownames(summary.data) <- c("congeneric", "tau", "parallel", "fixed")
> colnames(summary.data) <- c("chisq", "df")
> summary.data

```

```

              chisq df
congeneric  0.4597646 2
tau         56.1290414 5
parallel    91.2250565 8
fixed       98.5734749 9

```

A simple conclusion is that although the congeneric model has estimated the most parameters, the χ^2 difference between congeneric and tau equivalence models justifies rejecting tau equivalence in favor of the less restrictive congeneric model.

3.2 Two (perhaps correlated) factors

We now consider more interesting problems. The case of two correlated factors sometimes appears as a classic prediction problem (multiple measures of X, multiple measures of Y, what is the correlation between the two latent constructs) and sometimes as a measurement problem (multiple subfactors of X). The generation structure is similar.

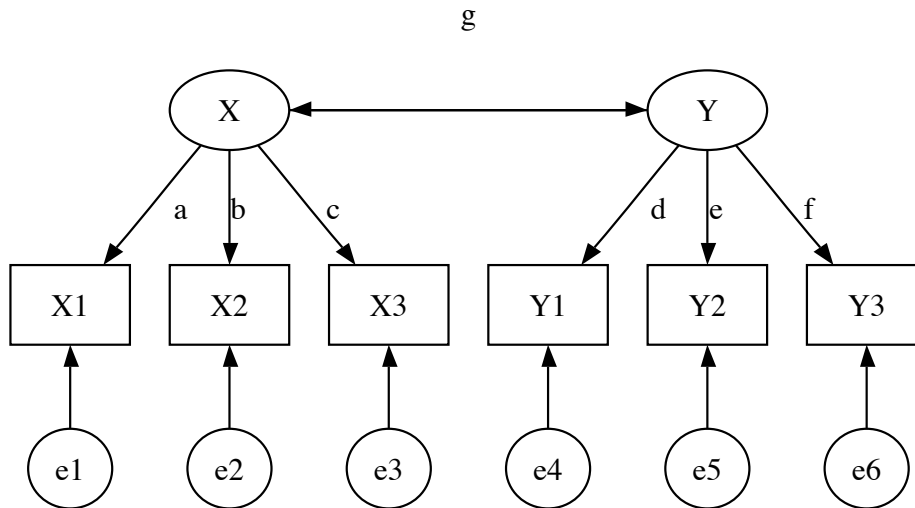


Figure 3.2: Six variables with two factors. This notation shows the error of measurement in the observed and latent variables. If $g > 0$, then the two factors are correlated.

3.2.1 Generating the data

We use the `sim.sem` function from before, and specify a two factor, uncorrelated structure. Figure 3.2 shows the general case where the two factors could be correlated. By specifying the path between the two latent variables to be 0, we make them uncorrelated.

```
> set.seed(42)
> N <- 1000
> pattern <- matrix(c(0.9, 0, 0.8, 0, 0.7, 0, 0, 0.8, 0, 0.7, 0, 0.6), ncol = 2,
+   byrow = TRUE)
> phi <- matrix(c(1, 0, 0, 1), ncol = 2)
> population <- sim.sem(loads = pattern, phi = phi, obs = FALSE)
> round(population$pattern, 2)

   theta1 theta2  e1  e2  e3  e4  e5  e6
V1    0.9    0.0 0.44 0.0 0.00 0.0 0.00 0.0
V2    0.8    0.0 0.00 0.6 0.00 0.0 0.00 0.0
V3    0.7    0.0 0.00 0.0 0.71 0.0 0.00 0.0
V4    0.0    0.8 0.00 0.0 0.00 0.6 0.00 0.0
V5    0.0    0.7 0.00 0.0 0.00 0.0 0.71 0.0
V6    0.0    0.6 0.00 0.0 0.00 0.0 0.00 0.8

> pop.cor <- round(population$structure %*% t(population$pattern), 2)
> pop.cor

   V1  V2  V3  V4  V5  V6
V1 1.00 0.72 0.63 0.00 0.00 0.00
V2 0.72 1.00 0.56 0.00 0.00 0.00
V3 0.63 0.56 1.00 0.00 0.00 0.00
```

```
> pairs.panels(data.f2)
```

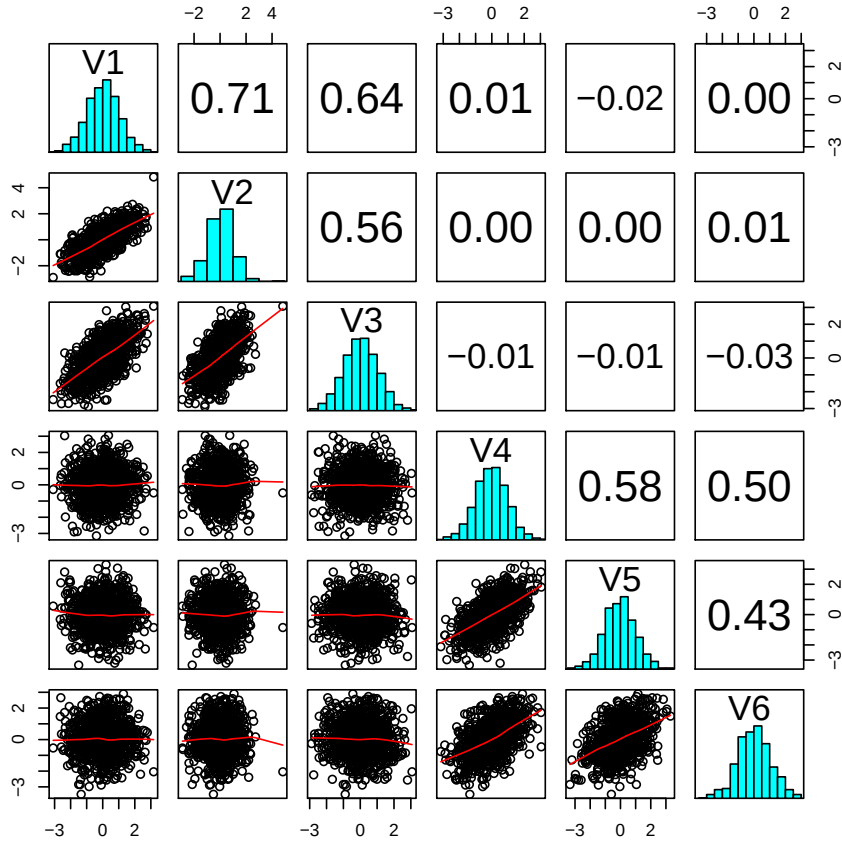


Figure 3.3: A ScatterPlot Matrix, SPLOM, of the six variables.

```
V4 0.00 0.00 0.00 1.00 0.56 0.48
V5 0.00 0.00 0.00 0.56 1.00 0.42
V6 0.00 0.00 0.00 0.48 0.42 1.00
```

```
> data.f2 <- sim.sem(loads = pattern, phi = phi)
```

The SPLOM suggests two separate factors in the data.

3.2.2 Exploratory Factor analysis of the data

This structure may be analyzed in a variety of different ways, including exploratory factor analysis. A “scree” plot of the eigen values of the matrix suggests a two factor solution. Based upon this “prior” hypotheses, we extract two factors using the **factanal** function.

```
> f2 <- factanal(data.f2, 2)
> f2
```

```
> VSS.scrree(cor(data.f2))
```

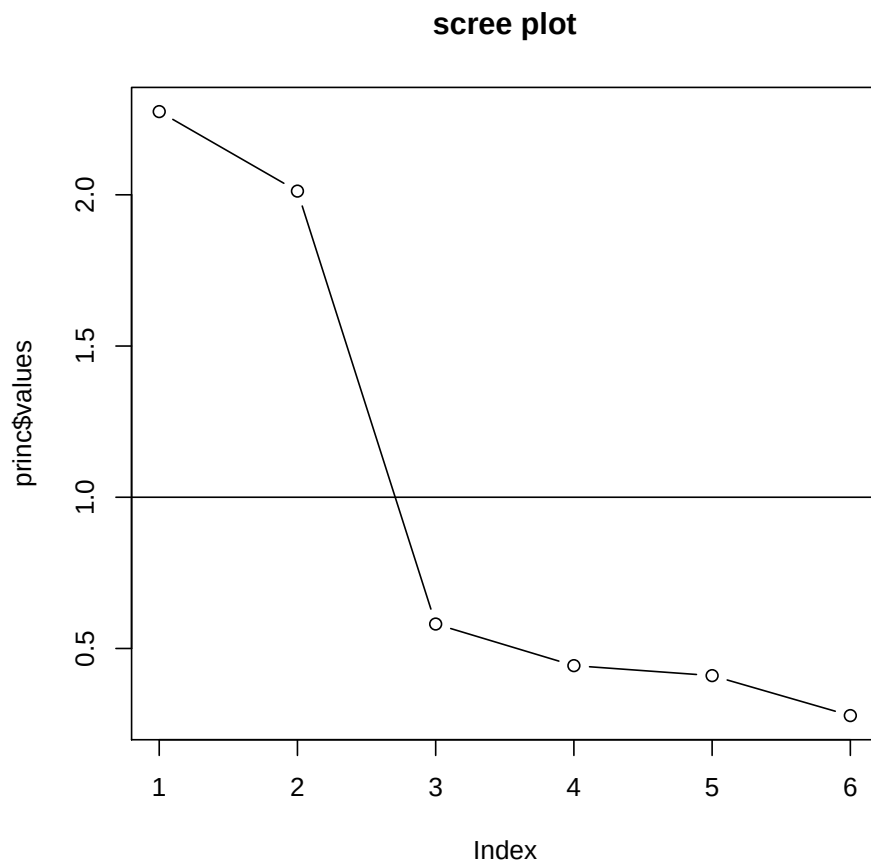


Figure 3.4: A scree plot of the eigen values of the simulated data suggests that two factors are the best representation of the data. Compare this to the two correlated factor problem, Figure 3.6, and the three correlated factor problem, Figure 3.9

Call:

```
factanal(x = data.f2, factors = 2)
```

Uniquenesses:

	V1	V2	V3	V4	V5	V6
	0.201	0.374	0.491	0.329	0.496	0.626

Loadings:

	Factor1	Factor2
V1	0.894	
V2	0.791	
V3	0.713	
V4		0.819
V5		0.709
V6		0.611

	Factor1	Factor2
SS loadings	1.934	1.548
Proportion Var	0.322	0.258
Cumulative Var	0.322	0.580

Test of the hypothesis that 2 factors are sufficient.

The chi square statistic is 3.97 on 4 degrees of freedom.

The p-value is 0.41

The factor loadings nicely capture the population values specified in section [3.2.1](#).

3.2.3 Confirmatory analysis with a predicted structure

We can also analyze these data taking a confirmatory, proposing that the first 3 variables load on one factor, and the second 3 variables load on a second factor.

```
> S.f2 <- cov(data.f2)
> model.two <- matrix(c("theta1 -> V1", "a", NA, "theta1 -> V2", "b", NA, "theta1 -> V3",
+ "c", NA, "theta2 -> V4", "d", NA, "theta2 -> V5", "e", NA, "theta2 -> V6",
+ "f", NA, "V1 <-> V1", "u", NA, "V2 <-> V2", "v", NA, "V3 <-> V3", "w", NA,
+ "V4 <-> V4", "x", NA, "V5 <-> V5", "y", NA, "V6 <-> V6", "z", NA, "theta1 <-> theta1",
+ NA, 1, "theta2 <-> theta2", NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.two) <- c("path", "label", "initial estimate")
> model.two
```

	path	label	initial	estimate
[1,]	"theta1 -> V1"	"a"	NA	
[2,]	"theta1 -> V2"	"b"	NA	
[3,]	"theta1 -> V3"	"c"	NA	
[4,]	"theta2 -> V4"	"d"	NA	
[5,]	"theta2 -> V5"	"e"	NA	

```

[6,] "theta2 -> V6"      "f"   NA
[7,] "V1 <-> V1"        "u"   NA
[8,] "V2 <-> V2"        "v"   NA
[9,] "V3 <-> V3"        "w"   NA
[10,] "V4 <-> V4"       "x"   NA
[11,] "V5 <-> V5"       "y"   NA
[12,] "V6 <-> V6"       "z"   NA
[13,] "theta1 <-> theta1" NA    "1"
[14,] "theta2 <-> theta2" NA    "1"

> sem.two = sem(model.two, S.f2, N)
> summary(sem.two, digits = 3)

Model Chisquare = 4.91   Df = 9 Pr(>Chisq) = 0.842
Chisquare (null model) = 2004   Df = 15
Goodness-of-fit index = 0.998
Adjusted goodness-of-fit index = 0.996
RMSEA index = 0   90% CI: (NA, 0.0205)
Bentler-Bonnett NFI = 0.998
Tucker-Lewis NNFI = 1.00
Bentler CFI = 1
BIC = -57.3

Normalized Residuals
      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
-8.02e-01 -4.43e-02 -7.84e-06 -8.03e-02  2.63e-05  3.51e-01

Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
a 0.898      0.0283   31.66  0.00e+00 V1 <--- theta1
b 0.770      0.0282   27.31  0.00e+00 V2 <--- theta1
c 0.727      0.0301   24.18  0.00e+00 V3 <--- theta1
d 0.817      0.0340   24.03  0.00e+00 V4 <--- theta2
e 0.731      0.0345   21.19  0.00e+00 V5 <--- theta2
f 0.644      0.0348   18.50  0.00e+00 V6 <--- theta2
u 0.204      0.0267    7.64  2.22e-14 V1 <--> V1
v 0.354      0.0244   14.53  0.00e+00 V2 <--> V2
w 0.510      0.0282   18.11  0.00e+00 V3 <--> V3
x 0.327      0.0391    8.36  0.00e+00 V4 <--> V4
y 0.527      0.0375   14.06  0.00e+00 V5 <--> V5
z 0.696      0.0385   18.09  0.00e+00 V6 <--> V6

Iterations = 22

> round(residuals(sem.two), 2)

      V1  V2  V3  V4  V5  V6
V1  0.00 0.00 0.00 0.01 -0.02 0.00
V2  0.00 0.00 0.00 0.00 0.00 0.01

```

```
V3  0.00 0.00  0.00 -0.01 -0.01 -0.03
V4  0.01 0.00 -0.01  0.00  0.00  0.00
V5 -0.02 0.00 -0.01  0.00  0.00  0.00
V6  0.00 0.01 -0.03  0.00  0.00  0.00
```

```
> std.coef(sem.two)
```

```
      Std. Estimate
a a 0.89320      V1 <--- theta1
b b 0.79150      V2 <--- theta1
c c 0.71341      V3 <--- theta1
d d 0.81904      V4 <--- theta2
e e 0.70975      V5 <--- theta2
f f 0.61113      V6 <--- theta2
```

It is useful to compare these “confirmatory” factor loadings with the factor loadings obtained by the exploratory factor analysis in section 3.2.2. Although the unstandardized loadings differ, the standardized loadings are identical to 3 decimals.

3.2.4 Confirmatory factor analysis with two independent factors with equal loadings within factors

The previous model allowed the factor loadings (and hence the quality of measurement of the variables) to differ. A more restrictive model (e.g., tau equivalence) forces the true score loadings to be equal within each factor. Note that although the χ^2 suggests that the model is not adequate, the more standard measures of adequacy of fit (e.g., RMSEA and the NFI) incorrectly show a good fit.

```
> model.twotau <- matrix(c("theta1 -> V1", "a", NA, "theta1 -> V2", "a", NA,
+   "theta1 -> V3", "a", NA, "theta2 -> V4", "d", NA, "theta2 -> V5", "d", NA,
+   "theta2 -> V6", "d", NA, "V1 <-> V1", "u", NA, "V2 <-> V2", "v", NA, "V3 <-> V3",
+   "w", NA, "V4 <-> V4", "x", NA, "V5 <-> V5", "y", NA, "V6 <-> V6", "z", NA,
+   "theta1 <-> theta1", NA, 1, "theta2 <-> theta2", NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.twotau) <- c("path", "label", "initial estimate")
> model.twotau
```

	path	label	initial estimate
[1,]	"theta1 -> V1"	"a"	NA
[2,]	"theta1 -> V2"	"a"	NA
[3,]	"theta1 -> V3"	"a"	NA
[4,]	"theta2 -> V4"	"d"	NA
[5,]	"theta2 -> V5"	"d"	NA
[6,]	"theta2 -> V6"	"d"	NA
[7,]	"V1 <-> V1"	"u"	NA
[8,]	"V2 <-> V2"	"v"	NA
[9,]	"V3 <-> V3"	"w"	NA
[10,]	"V4 <-> V4"	"x"	NA
[11,]	"V5 <-> V5"	"y"	NA


```

[12,] "V6 <-> V6"          "z"    NA
[13,] "theta1 <-> theta1" NA      "1"
[14,] "theta2 <-> theta2" NA      "1"

> sem.twotau = sem(model.twotau, S.f2, N)
> summary(sem.twotau, digits = 3)

Model Chisquare = 46.1   Df = 13 Pr(>Chisq) = 1.38e-05
Chisquare (null model) = 2004   Df = 15
Goodness-of-fit index = 0.985
Adjusted goodness-of-fit index = 0.976
RMSEA index = 0.0505   90% CI: (0.0352, 0.0666)
Bentler-Bonnett NFI = 0.977
Tucker-Lewis NNFI = 0.98
Bentler CFI = 0.983
BIC = -43.7

Normalized Residuals
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.3600 -0.7210 -0.0444 -0.2320  0.2160  1.8500

Parameter Estimates
  Estimate Std Error z value Pr(>|z|)
a 0.808    0.0215   37.5    0      V1 <--- theta1
d 0.739    0.0221   33.5    0      V4 <--- theta2
u 0.280    0.0198   14.2    0      V1 <--> V1
v 0.321    0.0212   15.1    0      V2 <--> V2
w 0.480    0.0275   17.5    0      V3 <--> V3
x 0.400    0.0270   14.8    0      V4 <--> V4
y 0.510    0.0309   16.5    0      V5 <--> V5
z 0.648    0.0366   17.7    0      V6 <--> V6

Iterations = 15

> round(residuals(sem.twotau), 2)

      V1    V2    V3    V4    V5    V6
V1 0.08 0.04 0.00 0.01 -0.02 0.00
V2 0.04 -0.03 -0.09 0.00 0.00 0.01
V3 0.00 -0.09 -0.09 -0.01 -0.01 -0.03
V4 0.01 0.00 -0.01 0.05 0.05 -0.02
V5 -0.02 0.00 -0.01 0.05 0.01 -0.08
V6 0.00 0.01 -0.03 -0.02 -0.08 -0.08

> std.coef(sem.twotau)

Std. Estimate
1 a 0.83640      V1 <--- theta1
2 a 0.81881      V2 <--- theta1
3 a 0.75897      V3 <--- theta1

```

```

4 d 0.75999      V4 <--- theta2
5 d 0.71919      V5 <--- theta2
6 d 0.67637      V6 <--- theta2

```

3.2.5 Structure invariance, part I— unequal loadings within factors - matched across factors

Are the two factors measured the same way? That is, are the loadings for the first factor the same as those for the second factor? We can test the model that the ordered loadings are the same across the two factors. We allow the errors to differ.

```

> model.two.invar <- matrix(c("theta1 -> V1", "a", NA, "theta1 -> V2", "b", NA,
+   "theta1 -> V3", "c", NA, "theta2 -> V4", "a", NA, "theta2 -> V5", "b", NA,
+   "theta2 -> V6", "c", NA, "V1 <-> V1", "u", NA, "V2 <-> V2", "v", NA, "V3 <-> V3",
+   "w", NA, "V4 <-> V4", "x", NA, "V5 <-> V5", "y", NA, "V6 <-> V6", "z", NA,
+   "theta1 <-> theta1", NA, 1, "theta2 <-> theta2", NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.two.invar) <- c("path", "label", "initial estimate")
> model.two.invar

```

	path	label	initial estimate
[1,]	"theta1 -> V1"	"a"	NA
[2,]	"theta1 -> V2"	"b"	NA
[3,]	"theta1 -> V3"	"c"	NA
[4,]	"theta2 -> V4"	"a"	NA
[5,]	"theta2 -> V5"	"b"	NA
[6,]	"theta2 -> V6"	"c"	NA
[7,]	"V1 <-> V1"	"u"	NA
[8,]	"V2 <-> V2"	"v"	NA
[9,]	"V3 <-> V3"	"w"	NA
[10,]	"V4 <-> V4"	"x"	NA
[11,]	"V5 <-> V5"	"y"	NA
[12,]	"V6 <-> V6"	"z"	NA
[13,]	"theta1 <-> theta1"	NA	"1"
[14,]	"theta2 <-> theta2"	NA	"1"

```

> sem.two.invar = sem(model.two.invar, S.f2, N)
> summary(sem.two.invar, digits = 3)

```

```

Model Chisquare = 10.7   Df = 12 Pr(>Chisq) = 0.557
Chisquare (null model) = 2004   Df = 15
Goodness-of-fit index = 0.996
Adjusted goodness-of-fit index = 0.994
RMSEA index = 0   90% CI: (NA, 0.0293)
Bentler-Bonnett NFI = 0.995
Tucker-Lewis NNFI = 1
Bentler CFI = 1
BIC = -72.2

```

```

Normalized Residuals
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.7400 -0.8010 -0.0444 -0.0818  0.4770  1.6000

Parameter Estimates
  Estimate Std Error z value Pr(>|z|)
a 0.862    0.0214   40.26  0      V1 <--- theta1
b 0.750    0.0215   34.91  0      V2 <--- theta1
c 0.690    0.0225   30.67  0      V3 <--- theta1
u 0.211    0.0249    8.48  0      V1 <--> V1
v 0.350    0.0235   14.86  0      V2 <--> V2
w 0.513    0.0277   18.53  0      V3 <--> V3
x 0.312    0.0315    9.89  0      V4 <--> V4
y 0.536    0.0330   16.24  0      V5 <--> V5
z 0.692    0.0371   18.66  0      V6 <--> V6

```

```
Iterations = 18
```

```
> round(residuals(sem.two.invar), 2)
```

	V1	V2	V3	V4	V5	V6
V1	0.06	0.04	0.06	0.01	-0.02	0.00
V2	0.04	0.03	0.04	0.00	0.00	0.01
V3	0.06	0.04	0.05	-0.01	-0.01	-0.03
V4	0.01	0.00	-0.01	-0.06	-0.05	-0.07
V5	-0.02	0.00	-0.01	-0.05	-0.04	-0.05
V6	0.00	0.01	-0.03	-0.07	-0.05	-0.06

```
> std.coef(sem.two.invar)
```

	Std. Estimate	
1 a	0.88236	V1 <--- theta1
2 b	0.78527	V2 <--- theta1
3 c	0.69405	V3 <--- theta1
4 a	0.83927	V4 <--- theta2
5 b	0.71551	V5 <--- theta2
6 c	0.63875	V6 <--- theta2

What is both interesting and disappointing from this example is that although the true loadings (refer back to 3.2.1) are not matched across the two factors, estimating a model that they are equivalent across factors can not be rejected, even with 1000 subjects. In addition, the correct population values are not included in the normal confidence intervals of the estimated values of a,b, and c.

3.2.6 Estimate two correlated factors

This next example is a bit more subtle, in that we generate data with a particular causal structure. The matrix of intercorrelations of the two factors leads to correlations between the

variables, but reflects the idea of a path coefficient from the first latent variable to the second one.¹

```
> set.seed(42)
> N <- 1000
> pattern <- matrix(c(0.9, 0, 0.8, 0, 0.7, 0, 0, 0.8, 0, 0.7, 0, 0.6), ncol = 2,
+   byrow = TRUE)
> phi <- matrix(c(1, 0.4, 0.4, 1), ncol = 2)
> population <- sim.sem(loads = pattern, phi = phi, obs = FALSE)
> round(population$pattern, 2)

   theta1 theta2  e1  e2  e3  e4  e5  e6
V1    0.9    0.0 0.44 0.0 0.00 0.0 0.00 0.0
V2    0.8    0.0 0.00 0.6 0.00 0.0 0.00 0.0
V3    0.7    0.0 0.00 0.0 0.71 0.0 0.00 0.0
V4    0.0    0.8 0.00 0.0 0.00 0.6 0.00 0.0
V5    0.0    0.7 0.00 0.0 0.00 0.0 0.71 0.0
V6    0.0    0.6 0.00 0.0 0.00 0.0 0.00 0.8

> round(population$structure, 2)

   theta1 theta2  e1  e2  e3  e4  e5  e6
V1    0.90    0.36 0.44 0.0 0.00 0.0 0.00 0.0
V2    0.80    0.32 0.00 0.6 0.00 0.0 0.00 0.0
V3    0.70    0.28 0.00 0.0 0.71 0.0 0.00 0.0
V4    0.32    0.80 0.00 0.0 0.00 0.6 0.00 0.0
V5    0.28    0.70 0.00 0.0 0.00 0.0 0.71 0.0
V6    0.24    0.60 0.00 0.0 0.00 0.0 0.00 0.8

> pop.cor <- population$structure %*% t(population$pattern)
> round(pop.cor, 2)

      V1  V2  V3  V4  V5  V6
V1 1.00 0.72 0.63 0.29 0.25 0.22
V2 0.72 1.00 0.56 0.26 0.22 0.19
V3 0.63 0.56 1.00 0.22 0.20 0.17
V4 0.29 0.26 0.22 1.00 0.56 0.48
V5 0.25 0.22 0.20 0.56 1.00 0.42
V6 0.22 0.19 0.17 0.48 0.42 1.00

> data.f2 <- sim.sem(loads = pattern, phi = phi)
```

The scree test for this problem also suggests two factors, although not as clearly as in example 3.2.1. We first conduct an exploratory factor analysis of the data. Rather than accepting the default value of a VARIMAX rotation, we examine the unrotated solution. For comparisons to a confirmatory factor analysis, we repeat the exploratory analysis with a VARIMAX rotation to simple structure.

¹Once again, we set the seed for the random number generator to a particular value in order to have reproducible results.

```
> pairs.panels(data.f2)
```

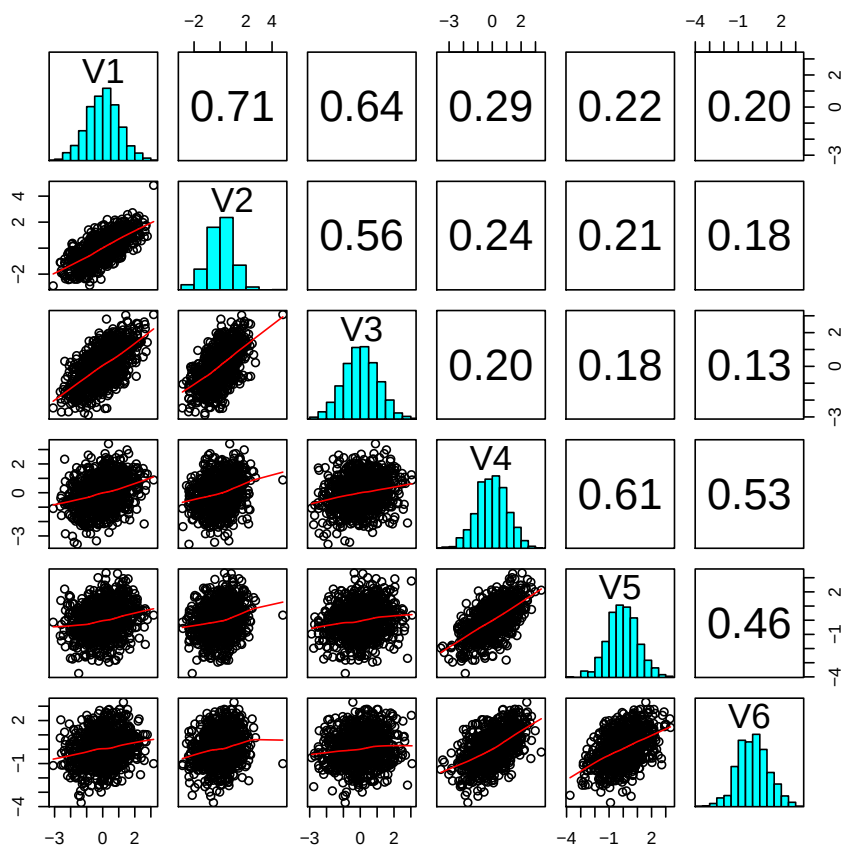


Figure 3.5: Six variables loading on 2 correlated factors

```
> VSS.scree(cor(data.f2))
```

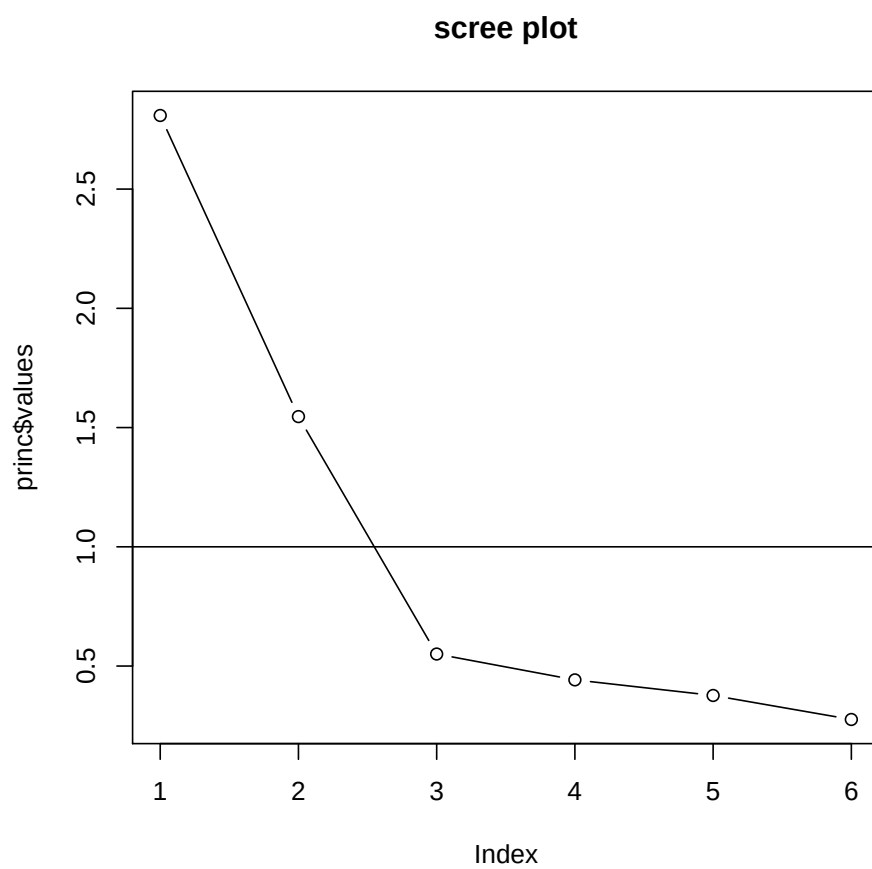


Figure 3.6: Scree plot of two correlated factors. Compare to Figure [3.4](#)

```

> f2 <- factanal(data.f2, 2, rotation = "none")
> f2

Call:
factanal(x = data.f2, factors = 2, rotation = "none")

Uniquenesses:
      V1      V2      V3      V4      V5      V6
0.201 0.374 0.491 0.295 0.463 0.600

Loadings:
      Factor1 Factor2
V1  0.845  -0.291
V2  0.749  -0.256
V3  0.667  -0.254
V4  0.552   0.633
V5  0.463   0.568
V6  0.404   0.487

      Factor1 Factor2
SS loadings      2.401  1.175
Proportion Var   0.400  0.196
Cumulative Var   0.400  0.596

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 3.99 on 4 degrees of freedom.
The p-value is 0.407

> f2v <- factanal(data.f2, 2)
> f2v

Call:
factanal(x = data.f2, factors = 2)

Uniquenesses:
      V1      V2      V3      V4      V5      V6
0.201 0.374 0.491 0.295 0.463 0.600

Loadings:
      Factor1 Factor2
V1 0.888
V2 0.786
V3 0.711
V4 0.226  0.809
V5 0.174  0.711
V6 0.156  0.613

      Factor1 Factor2

```

SS loadings	2.018	1.558
Proportion Var	0.336	0.260
Cumulative Var	0.336	0.596

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 3.99 on 4 degrees of freedom.
The p-value is 0.407

The sem for uncorrelated factors does not fit very well (as it should not!)

```
> S.f2 <- cov(data.f2)
> model.two <- matrix(c("theta1 -> V1", "a", NA, "theta1 -> V2", "b", NA, "theta1 -> V3",
+ "c", NA, "theta2 -> V4", "d", NA, "theta2 -> V5", "e", NA, "theta2 -> V6",
+ "f", NA, "V1 <-> V1", "u", NA, "V2 <-> V2", "v", NA, "V3 <-> V3", "w", NA,
+ "V4 <-> V4", "x", NA, "V5 <-> V5", "y", NA, "V6 <-> V6", "z", NA, "theta1 <-> theta1",
+ NA, 1, "theta2 <-> theta2", NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.two) <- c("path", "label", "initial estimate")
> model.two
```

	path	label	initial estimate
[1,]	"theta1 -> V1"	"a"	NA
[2,]	"theta1 -> V2"	"b"	NA
[3,]	"theta1 -> V3"	"c"	NA
[4,]	"theta2 -> V4"	"d"	NA
[5,]	"theta2 -> V5"	"e"	NA
[6,]	"theta2 -> V6"	"f"	NA
[7,]	"V1 <-> V1"	"u"	NA
[8,]	"V2 <-> V2"	"v"	NA
[9,]	"V3 <-> V3"	"w"	NA
[10,]	"V4 <-> V4"	"x"	NA
[11,]	"V5 <-> V5"	"y"	NA
[12,]	"V6 <-> V6"	"z"	NA
[13,]	"theta1 <-> theta1"	NA	"1"
[14,]	"theta2 <-> theta2"	NA	"1"

```
> sem.two = sem(model.two, S.f2, N)
> summary(sem.two, digits = 3)

Model Chisquare = 101 Df = 9 Pr(>Chisq) = 0
Chisquare (null model) = 2206 Df = 15
Goodness-of-fit index = 0.969
Adjusted goodness-of-fit index = 0.927
RMSEA index = 0.101 90% CI: (0.0838, 0.119)
Bentler-Bonnett NFI = 0.954
Tucker-Lewis NNFI = 0.93
Bentler CFI = 0.958
BIC = 38.5
```

Normalized Residuals

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-1.56e-05	-3.80e-06	2.10e+00	3.26e+00	6.43e+00	9.05e+00

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z)	
a	0.898	0.0283	31.66	0.00e+00	V1 <--- theta1
b	0.770	0.0282	27.31	0.00e+00	V2 <--- theta1
c	0.727	0.0301	24.18	0.00e+00	V3 <--- theta1
d	0.883	0.0341	25.88	0.00e+00	V4 <--- theta2
e	0.782	0.0344	22.74	0.00e+00	V5 <--- theta2
f	0.682	0.0346	19.72	0.00e+00	V6 <--- theta2
u	0.204	0.0267	7.64	2.22e-14	V1 <--> V1
v	0.354	0.0244	14.53	0.00e+00	V2 <--> V2
w	0.510	0.0282	18.11	0.00e+00	V3 <--> V3
x	0.328	0.0401	8.18	2.22e-16	V4 <--> V4
y	0.527	0.0376	14.02	0.00e+00	V5 <--> V5
z	0.696	0.0383	18.19	0.00e+00	V6 <--> V6

Iterations = 22

```
> std.coef(sem.two)
```

	Std. Estimate	
a a	0.89320	V1 <--- theta1
b b	0.79150	V2 <--- theta1
c c	0.71341	V3 <--- theta1
d d	0.83905	V4 <--- theta2
e e	0.73296	V5 <--- theta2
f f	0.63266	V6 <--- theta2

```
> round(residuals(sem.two), 2)
```

	V1	V2	V3	V4	V5	V6
V1	0.00	0.00	0.00	0.30	0.23	0.22
V2	0.00	0.00	0.00	0.25	0.22	0.19
V3	0.00	0.00	0.00	0.22	0.20	0.15
V4	0.30	0.25	0.22	0.00	0.00	0.00
V5	0.23	0.22	0.20	0.00	0.00	0.00
V6	0.22	0.19	0.15	0.00	0.00	0.00

and so we allow the two factors to be correlated.

```
> S.f2 <- cov(data.f2)
> model.two <- matrix(c("theta1 -> V1", "a", NA, "theta1 -> V2", "b", NA, "theta1 -> V3",
+ "c", NA, "theta2 -> V4", "d", NA, "theta2 -> V5", "e", NA, "theta2 -> V6",
+ "f", NA, "theta1 <-> theta2", "g", NA, "V1 <-> V1", "u", NA, "V2 <-> V2",
+ "v", NA, "V3 <-> V3", "w", NA, "V4 <-> V4", "x", NA, "V5 <-> V5", "y", NA,
+ "V6 <-> V6", "z", NA, "theta1 <-> theta1", NA, 1, "theta2 <-> theta2", NA,
+ 1), ncol = 3, byrow = TRUE)
```

```
> colnames(model.two) <- c("path", "label", "initial estimate")
> model.two
```

	path	label	initial estimate
[1,]	"theta1 -> V1"	"a"	NA
[2,]	"theta1 -> V2"	"b"	NA
[3,]	"theta1 -> V3"	"c"	NA
[4,]	"theta2 -> V4"	"d"	NA
[5,]	"theta2 -> V5"	"e"	NA
[6,]	"theta2 -> V6"	"f"	NA
[7,]	"theta1 <-> theta2"	"g"	NA
[8,]	"V1 <-> V1"	"u"	NA
[9,]	"V2 <-> V2"	"v"	NA
[10,]	"V3 <-> V3"	"w"	NA
[11,]	"V4 <-> V4"	"x"	NA
[12,]	"V5 <-> V5"	"y"	NA
[13,]	"V6 <-> V6"	"z"	NA
[14,]	"theta1 <-> theta1"	NA	"1"
[15,]	"theta2 <-> theta2"	NA	"1"

```
> sem.two = sem(model.two, S.f2, N)
> summary(sem.two, digits = 3)
```

```
Model Chisquare = 5.39   Df = 8 Pr(>Chisq) = 0.715
Chisquare (null model) = 2206   Df = 15
Goodness-of-fit index = 0.998
Adjusted goodness-of-fit index = 0.995
RMSEA index = 0   90% CI: (NA, 0.0278)
Bentler-Bonnett NFI = 0.998
Tucker-Lewis NNFI = 1.00
Bentler CFI = 1
BIC = -49.9
```

Normalized Residuals

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-0.888000	-0.057500	-0.000009	-0.062200	0.086200	0.443000

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z)	
a	0.899	0.0280	32.10	0e+00	V1 <--- theta1
b	0.770	0.0280	27.46	0e+00	V2 <--- theta1
c	0.726	0.0300	24.20	0e+00	V3 <--- theta1
d	0.890	0.0332	26.84	0e+00	V4 <--- theta2
e	0.776	0.0338	22.98	0e+00	V5 <--- theta2
f	0.679	0.0344	19.74	0e+00	V6 <--- theta2
g	0.359	0.0337	10.65	0e+00	theta2 <--> theta1
u	0.201	0.0256	7.86	4e-15	V1 <--> V1
v	0.354	0.0239	14.83	0e+00	V2 <--> V2

```

w 0.513    0.0280    18.34    0e+00    V3 <--> V3
x 0.315    0.0378     8.34    0e+00    V4 <--> V4
y 0.536    0.0361    14.82    0e+00    V5 <--> V5
z 0.700    0.0380    18.42    0e+00    V6 <--> V6

```

```
Iterations = 22
```

```
> std.coef(sem.two)
```

```

Std. Estimate
a a 0.89472    V1 <--- theta1
b b 0.79114    V2 <--- theta1
c c 0.71163    V3 <--- theta1
d d 0.84578    V4 <--- theta2
e e 0.72766    V5 <--- theta2
f f 0.63007    V6 <--- theta2

```

```
> round(residuals(sem.two), 2)
```

	V1	V2	V3	V4	V5	V6
V1	0.00	0.00	0.00	0.02	-0.02	0.00
V2	0.00	0.00	0.00	0.00	0.00	0.01
V3	0.00	0.00	0.00	-0.01	-0.01	-0.03
V4	0.02	0.00	-0.01	0.00	0.00	0.00
V5	-0.02	0.00	-0.01	0.00	0.00	0.01
V6	0.00	0.01	-0.03	0.00	0.01	0.00

3.3 Hierarchical models

The two correlated factors of section 3.2.1 may be thought of as representing two lower level factors each of which loads on a higher level factor. With just two lower level factors, the loadings on the higher level factor are not unique (one correlation, r , between the two factors may be represented in an infinite number of ways as the product of loadings ga and gb).

There are several ways of representing hierarchical models, including correlated level one factors with a g factor and uncorrelated lower level factors with a g factor (a bifactor solution). The latter may be estimated directly from the data, or may be found by using the Schmid-Leiman transformation of the correlated factors.

3.3.1 Two Correlated factors with a g factor

The hierarchical model of a g factor is underidentified unless we specify one of the g paths. Here we set it to 1 and then estimate the rest of the model.

```

> S.g2 <- cov(data.f2)
> model.g2 <- matrix(c("theta1 -> V1", "a", NA, "theta1 -> V2", "b", NA, "theta1 -> V3",
+ "c", NA, "theta2 -> V4", "d", NA, "theta2 -> V5", "e", NA, "theta2 -> V6",
+ "f", NA, "g -> theta1", NA, 1, "g -> theta2", "g2", NA, "V1 <-> V1", "u",

```

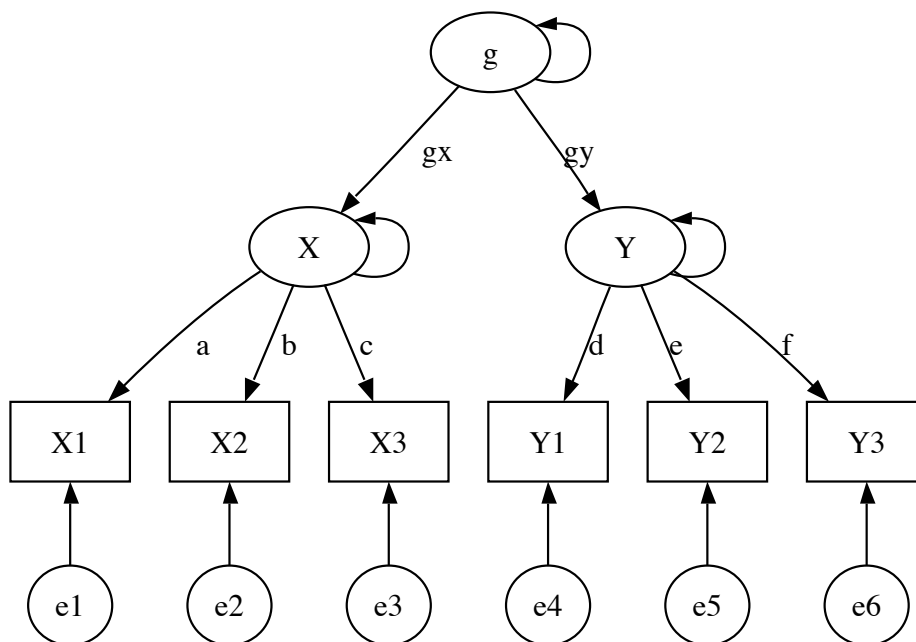


Figure 3.7: The correlation between two factors may be modeled by a g , general, factor. This representation shows all the errors that need to be estimated.

```
+      NA, "V2 <-> V2", "v", NA, "V3 <-> V3", "w", NA, "V4 <-> V4", "x", NA, "V5 <-> V5",
+      "y", NA, "V6 <-> V6", "z", NA, "theta1 <-> theta1", NA, 1, "theta2 <-> theta2",
+      NA, 1, "g <-> g", NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.g2) <- c("path", "label", "initial estimate")
> model.g2
```

	path	label	initial estimate
[1,]	"theta1 -> V1"	"a"	NA
[2,]	"theta1 -> V2"	"b"	NA
[3,]	"theta1 -> V3"	"c"	NA
[4,]	"theta2 -> V4"	"d"	NA
[5,]	"theta2 -> V5"	"e"	NA
[6,]	"theta2 -> V6"	"f"	NA
[7,]	"g -> theta1"	NA	"1"
[8,]	"g -> theta2"	"g2"	NA
[9,]	"V1 <-> V1"	"u"	NA
[10,]	"V2 <-> V2"	"v"	NA
[11,]	"V3 <-> V3"	"w"	NA
[12,]	"V4 <-> V4"	"x"	NA
[13,]	"V5 <-> V5"	"y"	NA
[14,]	"V6 <-> V6"	"z"	NA
[15,]	"theta1 <-> theta1"	NA	"1"
[16,]	"theta2 <-> theta2"	NA	"1"
[17,]	"g <-> g"	NA	"1"

```

> sem.g2 = sem(model.g2, S.g2, N)
> summary(sem.g2, digits = 3)

Model Chisquare = 5.39   Df = 8 Pr(>Chisq) = 0.715
Chisquare (null model) = 2206   Df = 15
Goodness-of-fit index = 0.998
Adjusted goodness-of-fit index = 0.995
RMSEA index = 0   90% CI: (NA, 0.0278)
Bentler-Bonnett NFI = 0.998
Tucker-Lewis NNFI = 1.00
Bentler CFI = 1
BIC = -49.9

Normalized Residuals
      Min.    1st Qu.    Median      Mean    3rd Qu.      Max.
-8.88e-01 -5.75e-02  6.72e-07 -6.22e-02  8.61e-02  4.43e-01

Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
a  0.636      0.0198   32.10  0.00e+00 V1 <--- theta1
b  0.544      0.0198   27.46  0.00e+00 V2 <--- theta1
c  0.513      0.0212   24.20  0.00e+00 V3 <--- theta1
d  0.767      0.0361   21.21  0.00e+00 V4 <--- theta2
e  0.669      0.0345   19.36  0.00e+00 V5 <--- theta2
f  0.585      0.0337   17.33  0.00e+00 V6 <--- theta2
g2 0.590      0.0747    7.90  2.89e-15 theta2 <--- g
u  0.201      0.0256    7.86  4.00e-15 V1 <--> V1
v  0.354      0.0239   14.83  0.00e+00 V2 <--> V2
w  0.513      0.0280   18.34  0.00e+00 V3 <--> V3
x  0.315      0.0378    8.34  0.00e+00 V4 <--> V4
y  0.536      0.0361   14.82  0.00e+00 V5 <--> V5
z  0.700      0.0380   18.42  0.00e+00 V6 <--> V6

Iterations = 26

> std.coef(sem.g2)

      Std. Estimate
a  a  0.89472      V1 <--- theta1
b  b  0.79114      V2 <--- theta1
c  c  0.71163      V3 <--- theta1
d  d  0.84578      V4 <--- theta2
e  e  0.72766      V5 <--- theta2
f  f  0.63007      V6 <--- theta2
      0.70711      theta1 <--- g
g2 g2 0.50800      theta2 <--- g

> round(residuals(sem.g2), 2)

```

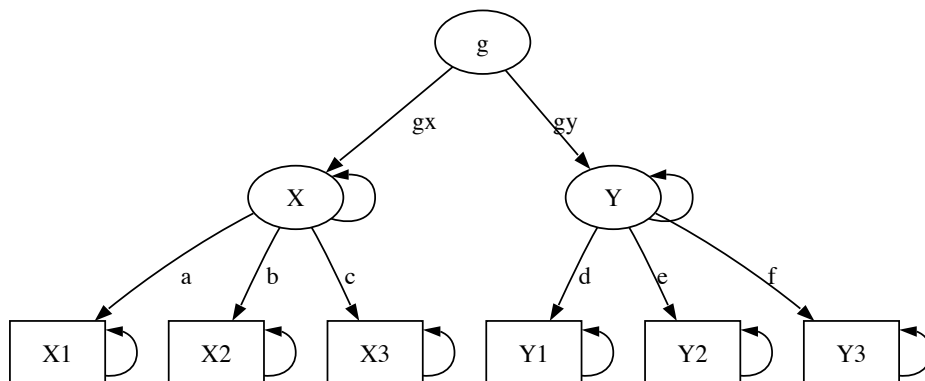


Figure 3.8: The correlation between two factors may be modeled by a g , general, factor. This representation is somewhat more compact than the previous figure (3.7.)

	V1	V2	V3	V4	V5	V6
V1	0.00	0.00	0.00	0.02	-0.02	0.00
V2	0.00	0.00	0.00	0.00	0.00	0.01
V3	0.00	0.00	0.00	-0.01	-0.01	-0.03
V4	0.02	0.00	-0.01	0.00	0.00	0.00
V5	-0.02	0.00	-0.01	0.00	0.00	0.01
V6	0.00	0.01	-0.03	0.00	0.01	0.00

3.3.2 Generating the data for 3 correlated factors

We have two demonstrations: the first is the two correlated factor data from section 3.2.1, the second is a three correlated factors. To create the later we use the `sim.sem` function with three latent variables.

```
> pattern <- matrix(c(0.9, 0.8, 0.7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.8, 0.7, 0.6,
+ 0, 0, 0, 0, 0, 0, 0, 0, 0.6, 0.5, 0.4), ncol = 3)
> colnames(pattern) <- c("F1", "F2", "F3")
> rownames(pattern) <- paste("V", 1:dim(pattern)[1], sep = "")
> pattern
```

	F1	F2	F3
V1	0.9	0.0	0.0
V2	0.8	0.0	0.0
V3	0.7	0.0	0.0
V4	0.0	0.8	0.0
V5	0.0	0.7	0.0
V6	0.0	0.6	0.0
V7	0.0	0.0	0.6
V8	0.0	0.0	0.5
V9	0.0	0.0	0.4

```
> VSS.screer(cor(data.f3))
```

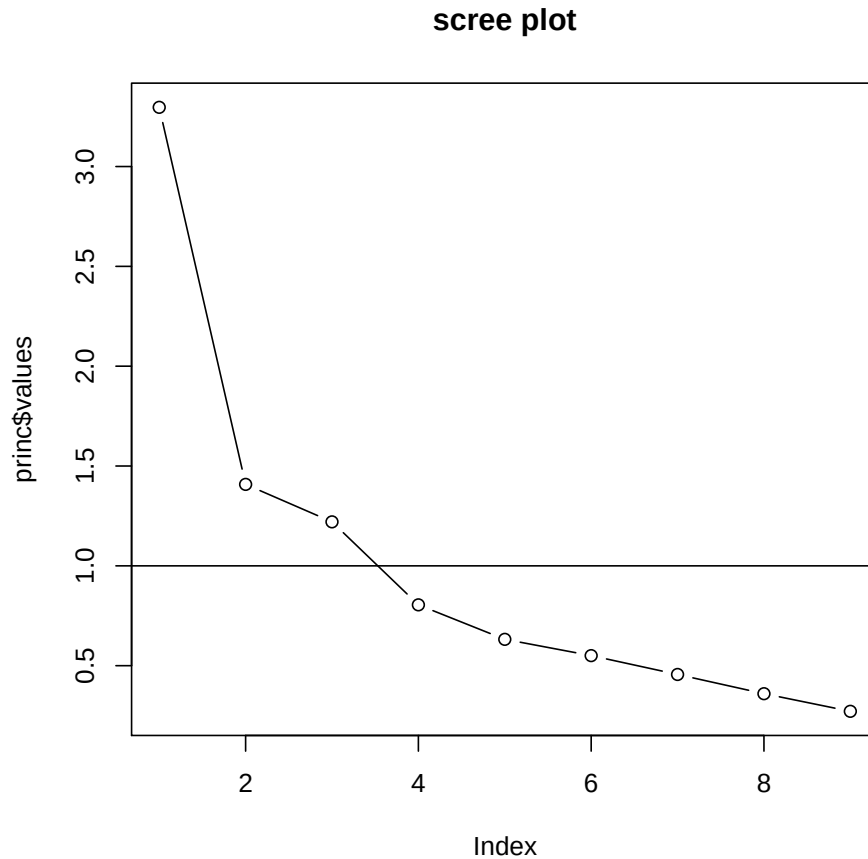


Figure 3.9: Scree plot of three correlated factors. Compare to the two uncorrelated factors, Figure 3.4, and the two correlated factors, ??

```
> phi <- matrix(c(1, 0, 0, 0.5, 1, 0, 0.4, 0.4, 1), ncol = 3, byrow = TRUE)
> phi
      [,1] [,2] [,3]
[1,]  1.0  0.0  0
[2,]  0.5  1.0  0
[3,]  0.4  0.4  1
> data.f3 <- sim.sem(loads = pattern, phi = phi)
```

3.3.3 Exploratory factor analysis with 3 factors

As a first approximation to these data, we can do a three factor exploratory analysis to try to understand the structure of the data.

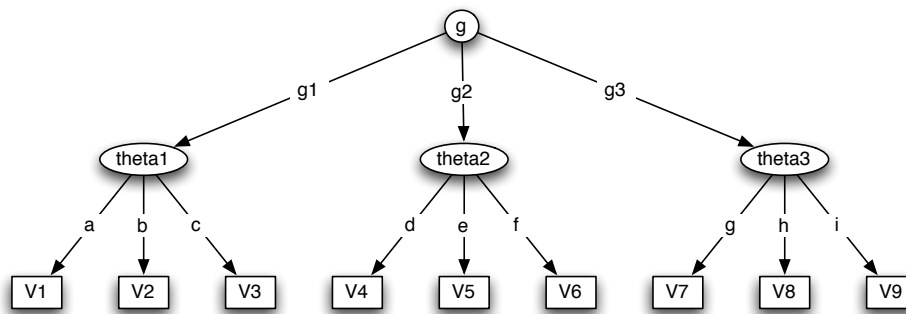


Figure 3.10: The correlation between three factors may be modeled by a g, general, factor.

```
> f3 <- factanal(data.f3, 3, rotation = "none")
> f3
```

Call:

```
factanal(x = data.f3, factors = 3, rotation = "none")
```

Uniquenesses:

V1	V2	V3	V4	V5	V6	V7	V8	V9
0.203	0.357	0.516	0.319	0.412	0.622	0.485	0.746	0.839

Loadings:

	Factor1	Factor2	Factor3
V1	0.835	-0.316	
V2	0.745	-0.297	
V3	0.659	-0.221	
V4	0.648	0.497	-0.121
V5	0.572	0.486	-0.154
V6	0.459	0.399	
V7	0.322	0.188	0.613
V8	0.226	0.196	0.406
V9	0.196	0.148	0.317

	Factor1	Factor2	Factor3
SS loadings	2.837	0.974	0.688
Proportion Var	0.315	0.108	0.076
Cumulative Var	0.315	0.424	0.500

Test of the hypothesis that 3 factors are sufficient.

The chi square statistic is 11.91 on 12 degrees of freedom.

The p-value is 0.453

Orthogonal Rotation

The loadings from this factor analysis are not particularly easy to understand and can be rotated to a more somewhat more understandable structure using the VARIMAX rotation (which is actually the default for factanal). We use the **GPArotation** package.

```
> library(GPArotation)
> f3v <- Varimax(loadings(f3))
> round(loadings(f3v), 2)
```

	Factor1	Factor2	Factor3
V1	0.87	0.20	0.07
V2	0.78	0.17	0.05
V3	0.67	0.17	0.09
V4	0.26	0.77	0.12
V5	0.20	0.74	0.07
V6	0.16	0.59	0.09
V7	0.14	0.14	0.69
V8	0.06	0.16	0.48
V9	0.07	0.13	0.37

The structure is more easy to understand than the original one, but still is somewhat hard to understand.

Oblique Rotation

By allowing the factors to be correlated, we are able to find a more simple representation of the factor pattern. However, we need to report both the factor loadings as well as the factor intercorrelations.

```
> f3o <- oblimin(loadings(f3))
> round(loadings(f3o), 2)
```

	Factor1	Factor2	Factor3
V1	0.89	0.00	0.00
V2	0.81	-0.01	-0.01
V3	0.68	0.02	0.03
V4	0.03	0.80	0.02
V5	-0.01	0.78	-0.03
V6	-0.02	0.62	0.01
V7	0.02	-0.02	0.72
V8	-0.04	0.06	0.49
V9	-0.02	0.05	0.38

The alternatives to exploratory factor analysis is to apply a confirmatory model specifying the “expected” structure. We do this with both a hierarchical g factor model as well as a bifactor model.

3.3.4 Three correlated factors with a g factor

```

> S.g3 <- cov(data.f3)
> model.g3 <- matrix(c("theta1 -> V1", "a", NA, "theta1 -> V2", "b", NA, "theta1 -> V3",
+ "c", NA, "theta2 -> V4", "d", NA, "theta2 -> V5", "e", NA, "theta2 -> V6",
+ "f", NA, "theta3 -> V7", "g", NA, "theta3 -> V8", "h", NA, "theta3 -> V9",
+ "i", NA, "g -> theta1", "g1", NA, "g -> theta2", "g2", NA, "g -> theta3",
+ "g3", NA, "V1 <-> V1", "u", NA, "V2 <-> V2", "v", NA, "V3 <-> V3", "w",
+ NA, "V4 <-> V4", "x", NA, "V5 <-> V5", "y", NA, "V6 <-> V6", "z", NA, "V7 <-> V7",
+ "s", NA, "V8 <-> V8", "t", NA, "V9 <-> V9", "r", NA, "theta1 <-> theta1",
+ NA, 1, "theta2 <-> theta2", NA, 1, "theta3 <-> theta3", NA, 1, "g <-> g",
+ NA, 1), ncol = 3, byrow = TRUE)
> colnames(model.g3) <- c("path", "label", "initial estimate")
> model.g3

      path          label initial estimate
[1,] "theta1 -> V1"      "a"      NA
[2,] "theta1 -> V2"      "b"      NA
[3,] "theta1 -> V3"      "c"      NA
[4,] "theta2 -> V4"      "d"      NA
[5,] "theta2 -> V5"      "e"      NA
[6,] "theta2 -> V6"      "f"      NA
[7,] "theta3 -> V7"      "g"      NA
[8,] "theta3 -> V8"      "h"      NA
[9,] "theta3 -> V9"      "i"      NA
[10,] "g -> theta1"      "g1"     NA
[11,] "g -> theta2"      "g2"     NA
[12,] "g -> theta3"      "g3"     NA
[13,] "V1 <-> V1"        "u"      NA
[14,] "V2 <-> V2"        "v"      NA
[15,] "V3 <-> V3"        "w"      NA
[16,] "V4 <-> V4"        "x"      NA
[17,] "V5 <-> V5"        "y"      NA
[18,] "V6 <-> V6"        "z"      NA
[19,] "V7 <-> V7"        "s"      NA
[20,] "V8 <-> V8"        "t"      NA
[21,] "V9 <-> V9"        "r"      NA
[22,] "theta1 <-> theta1" NA      "1"
[23,] "theta2 <-> theta2" NA      "1"
[24,] "theta3 <-> theta3" NA      "1"
[25,] "g <-> g"          NA      "1"

> sem.g3 = sem(model.g3, S.g3, N)
> summary(sem.g3, digits = 3)

Model Chisquare = 20.5   Df = 24 Pr(>Chisq) = 0.665
Chisquare (null model) = 2647   Df = 36
Goodness-of-fit index = 0.995

```

Adjusted goodness-of-fit index = 0.991
 RMSEA index = 0 90% CI: (NA, 0.0211)
 Bentler-Bonnett NFI = 0.992
 Tucker-Lewis NNFI = 1.00
 Bentler CFI = 1
 BIC = -145

Normalized Residuals

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-1.27e+00	-1.64e-01	1.63e-05	3.16e-02	3.43e-01	1.19e+00

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z)	
a	0.699	0.0380	18.40	0.00e+00	V1 <--- theta1
b	0.642	0.0361	17.77	0.00e+00	V2 <--- theta1
c	0.542	0.0327	16.55	0.00e+00	V3 <--- theta1
d	0.543	0.0744	7.29	3.02e-13	V4 <--- theta2
e	0.482	0.0664	7.26	3.79e-13	V5 <--- theta2
f	0.379	0.0535	7.09	1.34e-12	V6 <--- theta2
g	0.618	0.0485	12.72	0.00e+00	V7 <--- theta3
h	0.461	0.0392	11.77	0.00e+00	V8 <--- theta3
i	0.377	0.0379	9.95	0.00e+00	V9 <--- theta3
g1	0.788	0.0985	8.00	1.11e-15	theta1 <--- g
g2	1.370	0.2804	4.89	1.02e-06	theta2 <--- g
g3	0.583	0.0758	7.69	1.47e-14	theta3 <--- g
u	0.204	0.0238	8.54	0.00e+00	V1 <--> V1
v	0.375	0.0251	14.92	0.00e+00	V2 <--> V2
w	0.503	0.0268	18.78	0.00e+00	V3 <--> V3
x	0.342	0.0364	9.39	0.00e+00	V4 <--> V4
y	0.524	0.0356	14.71	0.00e+00	V5 <--> V5
z	0.702	0.0365	19.21	0.00e+00	V6 <--> V6
s	0.575	0.0609	9.43	0.00e+00	V7 <--> V7
t	0.781	0.0475	16.44	0.00e+00	V8 <--> V8
r	0.925	0.0480	19.26	0.00e+00	V9 <--> V9

Iterations = 35

> std.coef(sem.g3)

	Std. Estimate	
a a	0.89200	V1 <--- theta1
b b	0.80019	V2 <--- theta1
c c	0.69736	V3 <--- theta1
d d	0.84402	V4 <--- theta2
e e	0.74887	V5 <--- theta2
f f	0.60882	V6 <--- theta2
g g	0.68604	V7 <--- theta3
h h	0.51711	V8 <--- theta3

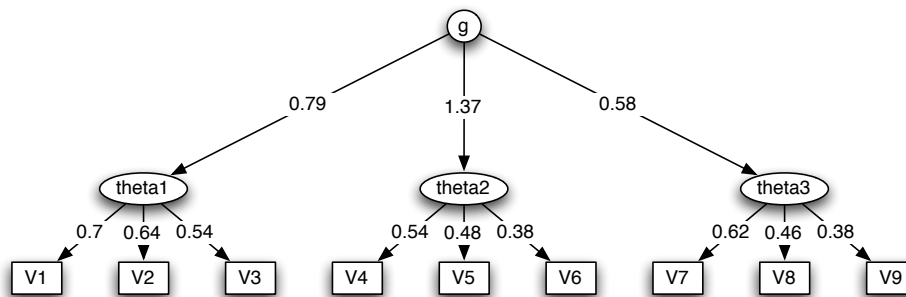


Figure 3.11: A hierarchical solution to the three correlated factors problem.

```

i   i   0.41359      V9 <--- theta3
g1  g1  0.61912      theta1 <--- g
g2  g2  0.80777      theta2 <--- g
g3  g3  0.50360      theta3 <--- g

```

3.3.5 Bifactor solutions

An alternative to the correlated lower level factors and a g factor is a “bifactor” model where each item is represented by two factors, a lower level, group, factor and a higher level, g, factor. This may be found directly through sem - cfa, or may be done indirectly by using a Schmid-Leiman transformation of the correlated factors. We use the same three factor data set as in the two previous sections (3.3.2, 3.3.4)

	path	label	initial	estimate
[1,]	"theta1 -> V1"	"a"	NA	
[2,]	"theta1 -> V2"	"b"	NA	
[3,]	"theta1 -> V3"	"c"	NA	
[4,]	"theta2 -> V4"	"d"	NA	
[5,]	"theta2 -> V5"	"e"	NA	
[6,]	"theta2 -> V6"	"f"	NA	
[7,]	"theta3 -> V7"	"g"	NA	
[8,]	"theta3 -> V8"	"h"	NA	
[9,]	"theta3 -> V9"	"i"	NA	
[10,]	"g -> V1"	"g1"	NA	
[11,]	"g -> V2"	"g2"	NA	
[12,]	"g -> V3"	"g3"	NA	
[13,]	"g -> V4"	"g4"	NA	
[14,]	"g -> V5"	"g5"	NA	
[15,]	"g -> V6"	"g6"	NA	
[16,]	"g -> V7"	"g7"	NA	
[17,]	"g -> V8"	"g8"	NA	
[18,]	"g -> V9"	"g9"	NA	
[19,]	"V1 <-> V1"	"u"	NA	
[20,]	"V2 <-> V2"	"v"	NA	

[21,]	"V3 <-> V3"	"w"	NA
[22,]	"V4 <-> V4"	"x"	NA
[23,]	"V5 <-> V5"	"y"	NA
[24,]	"V6 <-> V6"	"z"	NA
[25,]	"V7 <-> V7"	"s"	NA
[26,]	"V8 <-> V8"	"t"	NA
[27,]	"V9 <-> V9"	"r"	NA
[28,]	"theta1 <-> theta1"	NA	"1"
[29,]	"theta2 <-> theta2"	NA	"1"
[30,]	"theta3 <-> theta3"	NA	"1"
[31,]	"g <-> g"	NA	"1"

Model Chisquare = 16.8 Df = 18 Pr(>Chisq) = 0.536
 Chisquare (null model) = 2647 Df = 36
 Goodness-of-fit index = 0.996
 Adjusted goodness-of-fit index = 0.99
 RMSEA index = 0 90% CI: (NA, 0.0263)
 Bentler-Bonnett NFI = 0.994
 Tucker-Lewis NNFI = 1
 Bentler CFI = 1
 BIC = -108

Normalized Residuals

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-9.92e-01	-5.80e-02	-1.59e-05	1.55e-02	1.22e-01	8.96e-01

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z)	
a	0.697	0.0412	16.91	0.00e+00	V1 <--- theta1
b	0.658	0.0406	16.21	0.00e+00	V2 <--- theta1
c	0.517	0.0416	12.44	0.00e+00	V3 <--- theta1
d	0.501	0.0842	5.95	2.63e-09	V4 <--- theta2
e	0.533	0.0755	7.06	1.69e-12	V5 <--- theta2
f	0.409	0.0710	5.77	8.14e-09	V6 <--- theta2
g	0.642	0.0745	8.62	0.00e+00	V7 <--- theta3
h	0.451	0.0579	7.78	7.33e-15	V8 <--- theta3
i	0.355	0.0512	6.94	4.06e-12	V9 <--- theta3
g1	0.552	0.0477	11.57	0.00e+00	V1 <--- g
g2	0.490	0.0474	10.34	0.00e+00	V2 <--- g
g3	0.455	0.0466	9.77	0.00e+00	V3 <--- g
g4	0.758	0.0603	12.56	0.00e+00	V4 <--- g
g5	0.639	0.0591	10.81	0.00e+00	V5 <--- g
g6	0.503	0.0560	8.98	0.00e+00	V6 <--- g
g7	0.356	0.0426	8.37	0.00e+00	V7 <--- g
g8	0.268	0.0411	6.53	6.37e-11	V8 <--- g
g9	0.237	0.0423	5.60	2.15e-08	V9 <--- g
u	0.206	0.0272	7.57	3.84e-14	V1 <--> V1

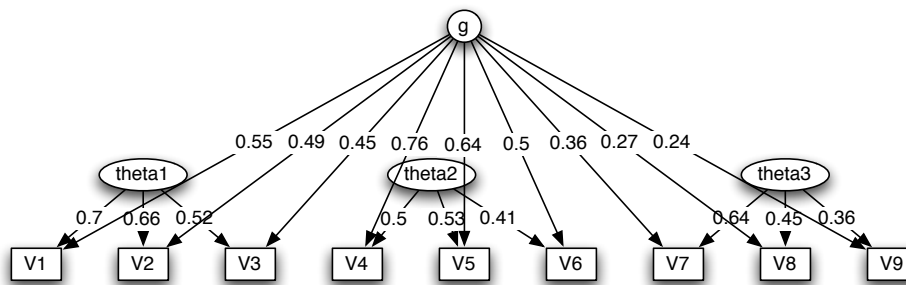


Figure 3.12: A bifactor solution to the three correlated factors problem.

```

v  0.369    0.0283    13.06    0.00e+00 V2 <--> V2
w  0.504    0.0267    18.87    0.00e+00 V3 <--> V3
x  0.364    0.0370     9.84    0.00e+00 V4 <--> V4
y  0.500    0.0431    11.60    0.00e+00 V5 <--> V5
z  0.695    0.0378    18.37    0.00e+00 V6 <--> V6
s  0.546    0.0898     6.09    1.15e-09 V7 <--> V7
t  0.791    0.0553    14.30    0.00e+00 V8 <--> V8
r  0.933    0.0494    18.88    0.00e+00 V9 <--> V9

```

Iterations = 54

		Std. Estimate	
a	a	0.69823	V1 <--- theta1
b	b	0.64468	V2 <--- theta1
c	c	0.52288	V3 <--- theta1
d	d	0.45929	V4 <--- theta2
e	e	0.48816	V5 <--- theta2
f	f	0.38775	V6 <--- theta2
g	g	0.61642	V7 <--- theta3
h	h	0.43658	V8 <--- theta3
i	i	0.33635	V9 <--- theta3
g1	g1	0.55298	V1 <--- g
g2	g2	0.47981	V2 <--- g
g3	g3	0.45993	V3 <--- g
g4	g4	0.69494	V4 <--- g
g5	g5	0.58531	V5 <--- g
g6	g6	0.47594	V6 <--- g
g7	g7	0.34180	V7 <--- g
g8	g8	0.25990	V8 <--- g
g9	g9	0.22427	V9 <--- g

3.3.6 Schmid Leiman transformations to orthogonalize the factors

An alternative to a confirmatory hierarchical analysis or bifactor solution is to extract at least 3 factors from a correlation matrix, transform them obliquely to a simple pattern solution, and

then extract the first factor from the correlations of these factors. From the resulting matrices, it is possible to find the g loading for each of the original variables (based upon the product of the g loadings of the factors and the loadings of the variables on these factors. Functionally, this is an alternative way of estimating a bifactor solution.

The **schmid** function found in the **psych** package finds the g factor and group factor loadings by doing a Schmid Leiman transformation. Here we show the Schmid Leiman analysis applied to the 3 correlated factors problem of section 3.3.4.

```
$sl
      g factor Factor1 Factor2 Factor3   h2   u2
V1      0.56    0.692   0.0016   0.0031 0.80 0.20
V2      0.50    0.629   0.0053   0.0122 0.66 0.34
V3      0.46    0.525   0.0098   0.0290 0.46 0.54
V4      0.66    0.022   0.5005   0.0161 0.65 0.35
V5      0.59    0.011   0.4874   0.0226 0.61 0.39
V6      0.48    0.017   0.3859   0.0129 0.38 0.62
V7      0.34    0.014   0.0150   0.6298 0.52 0.48
V8      0.26    0.034   0.0376   0.4296 0.25 0.75
V9      0.22    0.012   0.0308   0.3361 0.15 0.85
```

```
$orthog
      Factor1 Factor2 Factor3
V1      0.892   0.0025 -0.0035
V2      0.810  -0.0085 -0.0139
V3      0.677   0.0157   0.0332
V4      0.028   0.8039   0.0185
V5     -0.014   0.7829  -0.0258
V6     -0.022   0.6198   0.0148
V7      0.018  -0.0240   0.7207
V8     -0.044   0.0603   0.4916
V9     -0.016   0.0495   0.3846
```

```
$fcor
      [,1] [,2] [,3]
[1,] 1.00 0.49 0.31
[2,] 0.49 1.00 0.38
[3,] 0.31 0.38 1.00
```

```
$gloading
```

```
Loadings:
      Factor1
[1,] 0.631
[2,] 0.783
[3,] 0.486
```

```
Factor1
```

SS loadings	1.247
Proportion Var	0.416

Although not identical to the results of the bifactor solution, the results agree to two decimal places. Why are these estimates not the same? Because in the case of the confirmatory model, the loadings of the variables on one factor on other factors are set to 0, while in the Schmid Leiman case, they are allowed to be non-zero. The loadings on the general factor are used to calculate the ω_h coefficient discussed by Zinbarg et al. 2006.

3.3.7 Omega as an estimate of reliability

Many scales are assumed by their developers and users to be primarily a measure of one latent variable. When it is also assumed that the scale conforms to the effect indicator model of measurement (as is almost always the case in psychological assessment), it is important to support such an interpretation with evidence regarding the internal structure of that scale. In particular, it is important to examine two related properties pertaining to the internal structure of such a scale. The first property relates to whether all the indicators forming the scale measure a latent variable in common.

The second internal structural property pertains to the proportion of variance in the scale scores (derived from summing or averaging the indicators) accounted for by this latent variable that is common to all the indicators (Cronbach, 1951; McDonald, 1999; Revelle, 1979). That is, if an effect indicator scale is primarily a measure of one latent variable common to all the indicators forming the scale, then that latent variable should account for the majority of the variance in the scale scores. Put differently, this variance ratio provides important information about the sampling fluctuations when estimating individuals' standing on a latent variable common to all the indicators arising from the sampling of indicators (i.e., when dealing with either Type 2 or Type 12 sampling, to use the terminology of Lord, 1956). That is, this variance proportion can be interpreted as the square of the correlation between the scale score and the latent variable common to all the indicators in the infinite universe of indicators of which the scale indicators are a subset. Put yet another way, this variance ratio is important both as reliability and a validity coefficient. This is a reliability issue as the larger this variance ratio is, the more accurately one can predict an individual's relative standing on the latent variable common to all the scale's indicators based on his or her observed scale score. At the same time, this variance ratio also bears on the construct validity of the scale given that construct validity encompasses the internal structure of a scale." (Zinbarg, Yovel, Revelle, and McDonald, 2006). McDonald has proposed coefficient omega as an estimate of the general factor saturation of a test. Zinbarg, Revelle, Yovel and Li (2005) compare McDonald's Omega to Cronbach's alpha and Revelle's beta. They conclude that omega is the best estimate. (See also Zinbarg et al., 2006)

One way to find omega is to do a factor analysis of the original data set, rotate the factors obliquely, do a Schmid-Leiman (schmid) transformation, and then find omega. The **psych** package function **omega** does that.