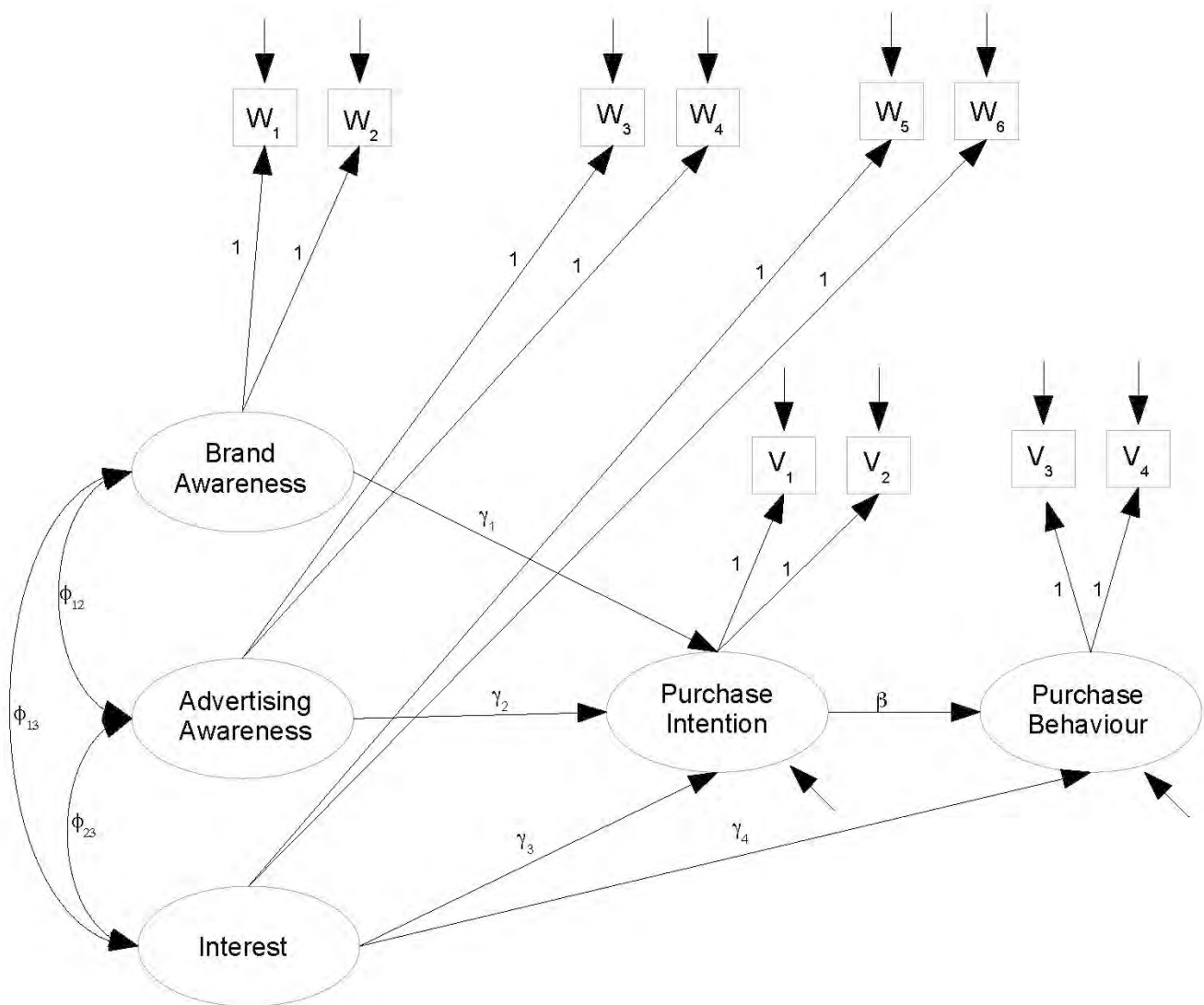


Brand Awareness

A major Canadian coffee shop chain is trying to break into the U.S. Market. They assess the following variables twice on a random sample of coffee-drinking adults. The two measurements of each variable are conducted at different times by different interviewers asking somewhat different questions, in such a way that the errors of measurement may be assumed independent. The variables are

- Brand Awareness: Familiarity with the coffee shop chain
- Advertising Awareness: Recall for advertising of the coffee shop chain
- Interest in the product category: Mostly this was how much they say they like doughnuts.
- Purchase Intention: Expressed willingness to go to an outlet of the coffeeshop chain and make an order.
- Purchase behaviour: Reported dollars spent at the chain during the 2 months following the interview, based on a later telephone interview.

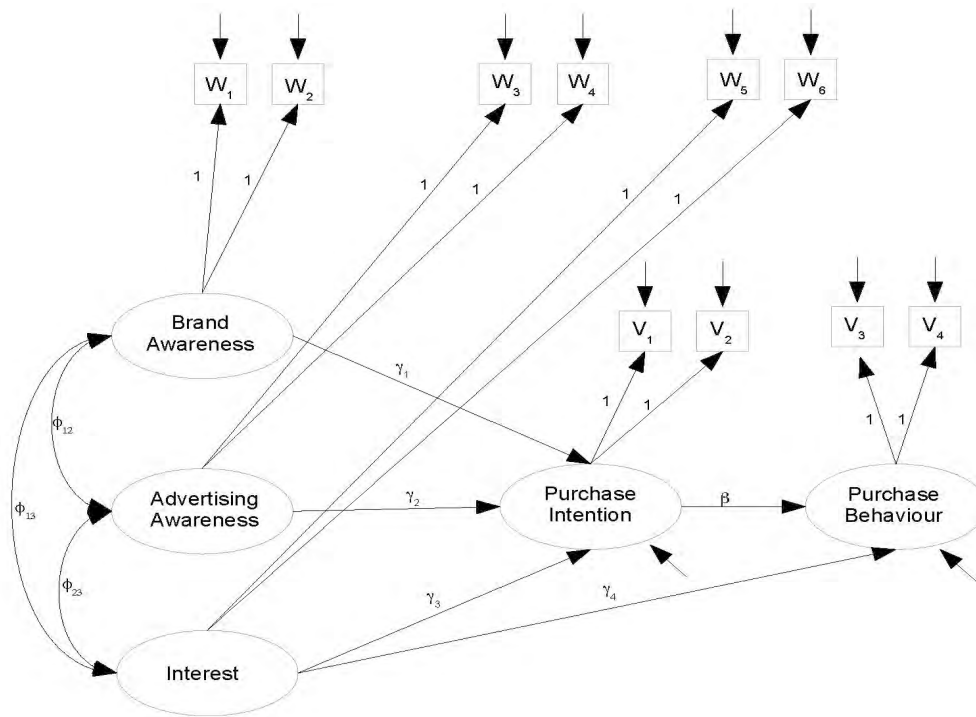
All variables were measured on a scale from 0 to 100 except purchase behaviour, which is in dollars.



```

> rm(list=ls()); options(scipen=999)
> # install.packages("lavaan", dependencies = TRUE) # Only need to do this once
> library(lavaan)
This is lavaan 0.6-11
lavaan is FREE software! Please report any bugs.
>
> coffee = read.table("http://www.utstat.toronto.edu/brunner/openSEM/data/timmy1.data.txt")
> head(coffee)
  w1 w2 w3 w4 w5 w6 v1 v2 v3 v4
1 40 23 26 21 48 38 22 22 15 15
2 45 24 29 23 49 48 26 13  8 13
3 29 21 21 13 42 37 18 12 13 13
4 38 26 18 19 47 42 20  9 12 10
5 47 31 30 18 48 52 26 16 22 16
6 31 24 18 13 39 40 20 12 16 18
>
> # Observed variables
> #   w1 = Brand Awareness 1
> #   w2 = Brand Awareness 2
> #   w3 = Ad Awareness 1
> #   w4 = Ad Awareness 2
> #   w5 = Interest 1
> #   w6 = Interest 2
> #   v1 = Purchase Intention 1
> #   v2 = Purchase Intention 2
> #   v3 = Purchase Behaviour 1
> #   v4 = Purchase Behaviour 2
> # Latent variables
> #   L_BrAw = True brand awareness
> #   L_AdAw = True advertising awareness
> #   L_Inter = True interest in the product category
> #   L_PI = True purchase intention
> #   L_PBeh = True purchase behaviour
>

```



```

> torus1 =
+ '
+ # Latent variable model
+   L_PI ~ gamma1*L_BrAw + gamma2*L_AdAw + gamma3*L_Inter
+   L_PBeh ~ gamma4*L_Inter + beta*L_PI
+ # Measurement model (simple double measurement)
+   L_BrAw =~ 1*w1 + 1*w2
+   L_AdAw =~ 1*w3 + 1*w4
+   L_Inter =~ 1*w5 + 1*w6
+   L_PI =~ 1*v1 + 1*v2
+   L_PBeh =~ 1*v3 + 1*v4
+ # Variances and covariances
+ # Exogenous latent variables
+   L_BrAw ~~ phi11*L_BrAw # Var(L_BrAw) = phi11
+   L_BrAw ~~ phi12*L_AdAw # Cov(L_BrAw,L_AdAw) = phi12
+   L_BrAw ~~ phi13*L_Inter # Cov(L_BrAw,L_Inter) = phi13
+   L_AdAw ~~ phi22*L_AdAw # Var(L_AdAw) = phi22
+   L_AdAw ~~ phi23*L_Inter # Cov(L_AdAw,L_Inter) = phi23
+   L_Inter ~~ phi33*L_Inter # Var(L_Inter) = phi33
+ # Errors in the latent model (epsilons)
+   L_PI =~ psi1*L_PI # Var(epsilon1) = psi1
+   L_PBeh =~ psi2*L_PBeh # Var(epsilon2) = psi2
+ # Measurement errors
+   w1 =~ omega1*w1 # Var(e1) = omega1
+   w2 =~ omega2*w2 # Var(e2) = omega2
+   w3 =~ omega3*w3 # Var(e3) = omega3
+   w4 =~ omega4*w4 # Var(e4) = omega4
+   w5 =~ omega5*w5 # Var(e5) = omega5
+   w6 =~ omega6*w6 # Var(e6) = omega6
+   v1 =~ omega7*v1 # Var(e7) = omega7
+   v2 =~ omega8*v2 # Var(e8) = omega8
+   v3 =~ omega9*v3 # Var(e9) = omega9
+   v4 =~ omega10*v4 # Var(e10) = omega10
+ # Bounds (Variances are positive)
+   phi11 > 0; phi22 > 0; phi33 > 0
+   psi1 > 0; psi2 > 0
+   omega1 > 0; omega2 > 0; omega3 > 0; omega4 > 0; omega5 > 0
+   omega6 > 0; omega7 > 0; omega8 > 0; omega9 > 0; omega10 > 0
+ ' # End of model torus1

```

```

> fit1 = lavaan(torus1, data=coffee)
> show(fit1)
lavaan 0.6-11 ended normally after 113 iterations

```

Estimator	ML
Optimization method	NLMINB
Number of model parameters	23
Number of inequality constraints	15
Number of observations	200

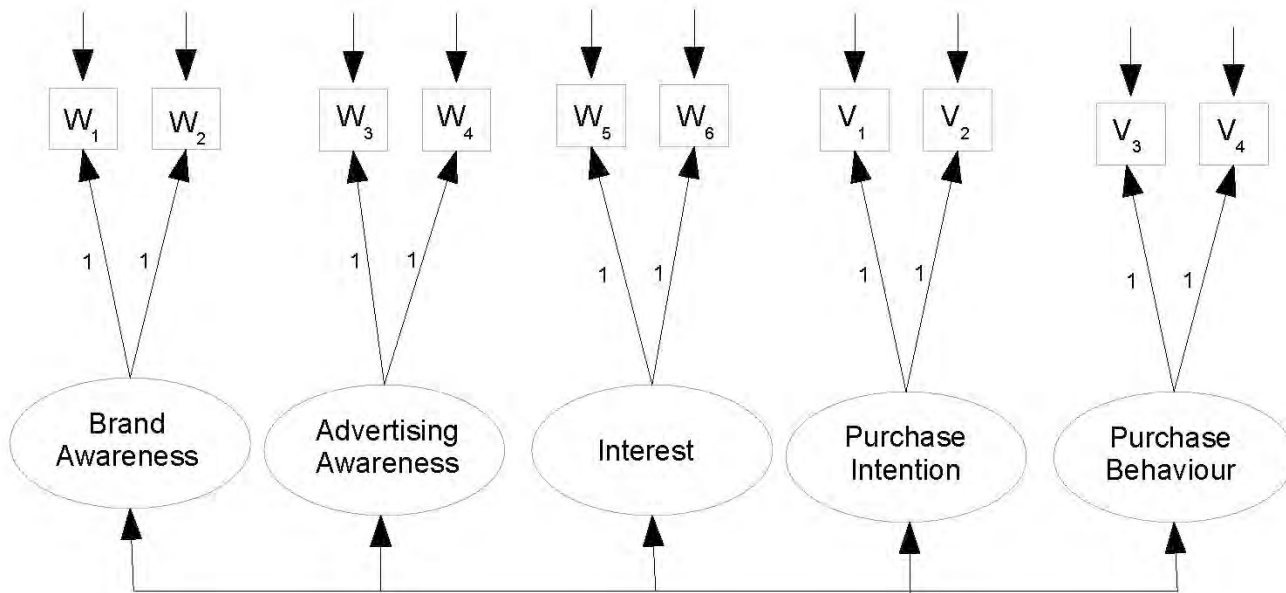
Model Test User Model:

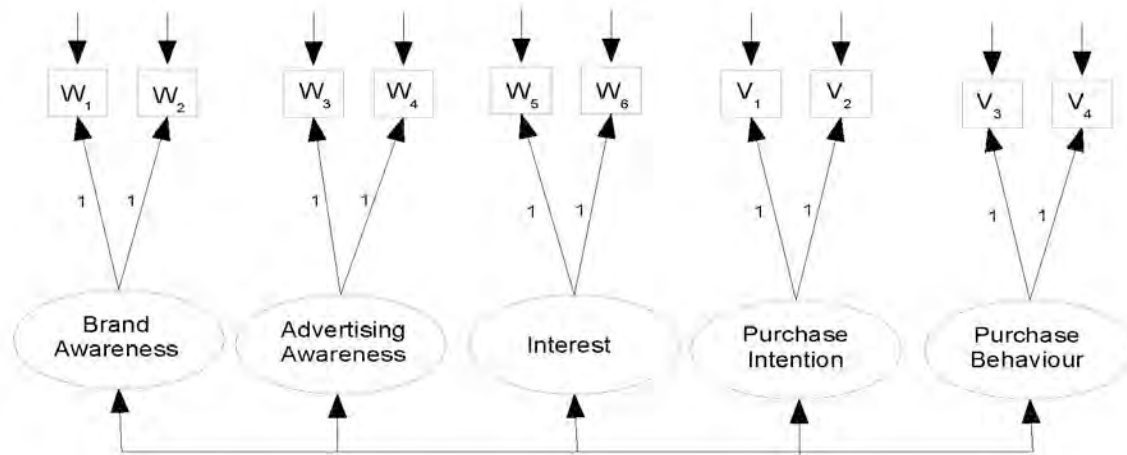
Test statistic	77.752
Degrees of freedom	32
P-value (Chi-square)	0.000

```

>
> # It did not fit, and matched SAS.
> # Split the problem up into parts. Look first at the measurement model.
>

```





```

> torus2 =
+ '
+ # Measurement model (still simple double measurement)
+   L_BrAw =~ 1*w1 + 1*w2
+   L_AdAw =~ 1*w3 + 1*w4
+   L_Inter =~ 1*w5 + 1*w6
+   L_PI =~ 1*v1 + 1*v2
+   L_PBeh =~ 1*v3 + 1*v4
+ # Variances and covariances
+   # Latent variables
+   L_BrAw =~ phi11*L_BrAw      # Var(L_BrAw)           = phi11
+   L_BrAw =~ phi12*L_AdAw     # Cov(L_BrAw, L_AdAw) = phi12
+   L_BrAw =~ phi13*L_Inter    # Cov(L_BrAw, L_Inter) = phi13
+   L_BrAw =~ phi14*L_PI       # Cov(L_BrAw, L_PI)   = phi14
+   L_BrAw =~ phi15*L_PBeh     # Cov(L_BrAw, L_PBeh) = phi15
+
+   L_AdAw =~ phi22*L_AdAw     # Var(L_AdAw)         = phi22
+   L_AdAw =~ phi23*L_Inter    # Cov(L_AdAw, L_Inter) = phi23
+   L_AdAw =~ phi24*L_PI       # Cov(L_AdAw, L_PI)   = phi24
+   L_AdAw =~ phi25*L_PBeh     # Cov(L_AdAw, L_PBeh) = phi25
+
+   L_Inter =~ phi33*L_Inter    # Var(L_Inter)        = phi33
+   L_Inter =~ phi34*L_PI       # Cov(L_Inter, L_PI)  = phi34
+   L_Inter =~ phi35*L_PBeh     # Cov(L_Inter, L_PBeh) = phi35
+
+   L_PI =~ phi44*L_PI          # Var(L_PI)           = phi44
+   L_PI =~ phi45*L_PBeh        # Cov(L_PI, L_PBeh)   = phi45
+
+   L_PBeh =~ phi55*L_PBeh      # Var(L_PBeh)         = phi55
+ # Measurement errors
+   w1 =~ omega1*w1            # Var(e1)             = omega1
+   w2 =~ omega2*w2            # Var(e2)             = omega2
+   w3 =~ omega3*w3            # Var(e3)             = omega3
+   w4 =~ omega4*w4            # Var(e4)             = omega4
+   w5 =~ omega5*w5            # Var(e5)             = omega5
+   w6 =~ omega6*w6            # Var(e6)             = omega6
+   v1 =~ omega7*v1            # Var(e7)             = omega7
+   v2 =~ omega8*v2            # Var(e8)             = omega8
+   v3 =~ omega9*v3            # Var(e9)             = omega9
+   v4 =~ omega10*v4           # Var(e10)            = omega10
+ # Bounds (Variances are positive)
+   phi11 > 0; phi22 > 0; phi33 > 0; phi44 > 0; phi55 > 0
+   omega1 > 0; omega2 > 0; omega3 > 0; omega4 > 0; omega5 > 0
+   omega6 > 0; omega7 > 0; omega8 > 0; omega9 > 0; omega10 > 0
+ ' # End of model torus2

```

```
> fit2 = lavaan(torus2, data=coffee)
> show(fit2)
```

lavaan 0.6-11 ended normally after 124 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	25
Number of inequality constraints	15
Number of observations	200

Model Test User Model:

Test statistic	76.380
Degrees of freedom	30
P-value (Chi-square)	0.000

```
>
> # Specify the model more briefly, without all those variances and covariances.
> torus2b =
+ '
+ # Measurement model (still simple double measurement)
+   L_BrAw =~ 1*w1 + 1*w2
+   L_AdAw =~ 1*w3 + 1*w4
+   L_Inter =~ 1*w5 + 1*w6
+   L_PI =~ 1*v1 + 1*v2
+   L_PBeh =~ 1*v3 + 1*v4
+ ' # End of model torus2b
> fit2b = cfa(torus2b, data=coffee) # Using cfa (confirmatory factor analysis)
>
> show(fit2b)
```

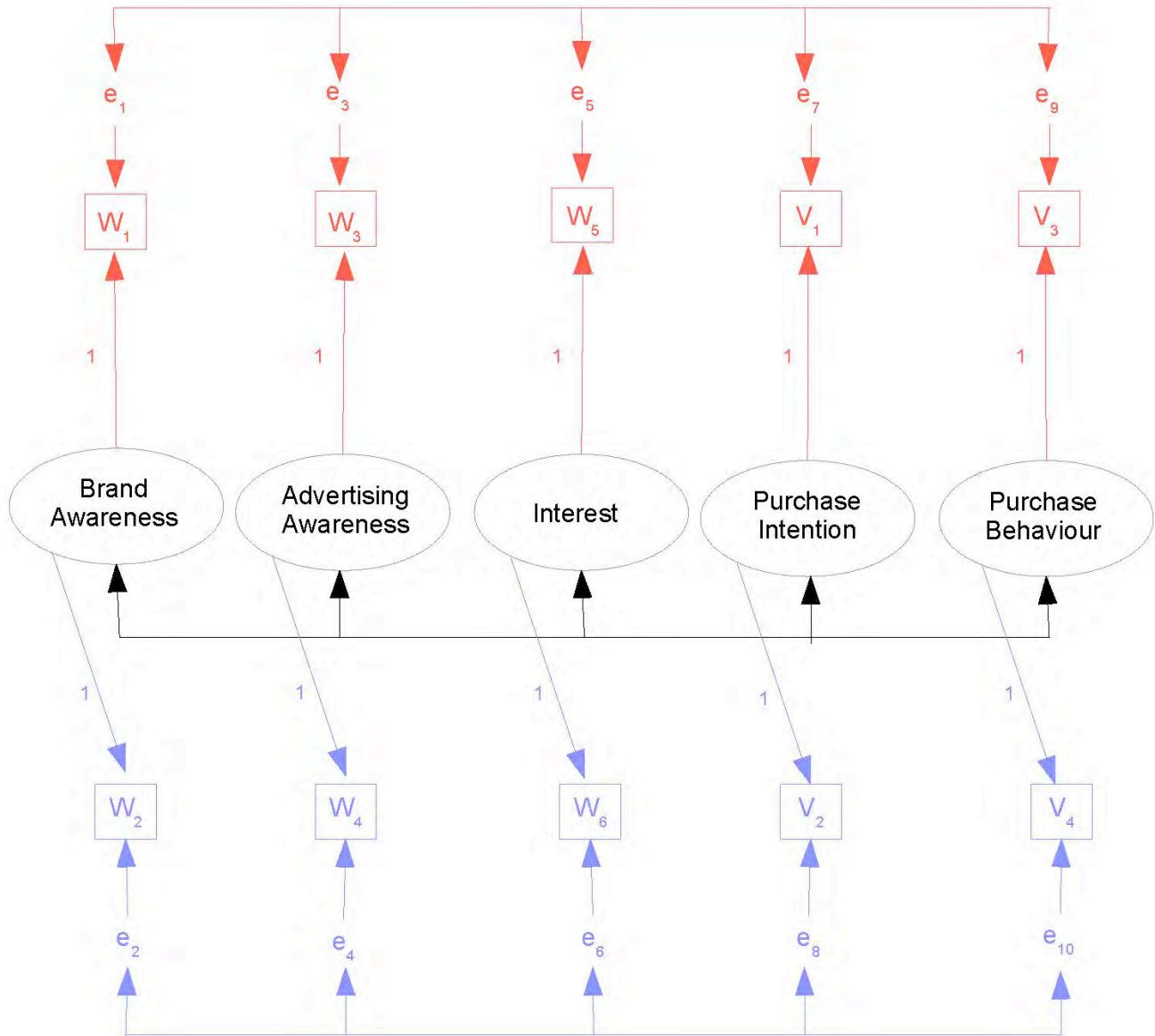
lavaan 0.6-11 ended normally after 139 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	25
Number of observations	200

Model Test User Model:

Test statistic	76.380
Degrees of freedom	30
P-value (Chi-square)	0.000

```
> # The measurement model does not fit. Try a true double measurement model, allowing
covariances within sets.
```



```

> torus3 =
+ '
+ # Measurement model (still simple double measurement)
+   L_BrAw  =~ 1*w1 + 1*w2
+   L_AdAw  =~ 1*w3 + 1*w4
+   L_Inter =~ 1*w5 + 1*w6
+   L_PI    =~ 1*v1 + 1*v2
+   L_PBeh  =~ 1*v3 + 1*v4
+ # Add covariances between measurement error terms, without naming them
+   w1 ~~ w3; w1 ~~ w5; w1 ~~ v1; w1 ~~ v3
+           w3 ~~ w5; w3 ~~ v1; w3 ~~ v3
+           w5 ~~ v1; w5 ~~ v3
+           v1 ~~ v3
+   w2 ~~ w4; w2 ~~ w6; w2 ~~ v2; w2 ~~ v4
+           w4 ~~ w6; w4 ~~ v2; w4 ~~ v4
+           w6 ~~ v2; w6 ~~ v4
+           v2 ~~ v4
+ ' # End of model torus3

```

```
> fit3 = cfa(torus3, data=coffee)
```

```
Warning message:
```

```
In lav_object_post_check(object) :
```

```
lavaan WARNING: the covariance matrix of the residuals of the observed  
variables (theta) is not positive definite;  
use lavInspect(fit, "theta") to investigate.
```

```
>
```

```
> lavInspect(fit3, "theta")
```

```
      w1      w2      w3      w4      w5      w6      v1      v2      v3      v4  
w1 10.617  
w2  0.000 10.477  
w3  2.700  0.000 11.704  
w4  0.000 -1.726  0.000 11.263  
w5  1.246  0.000  0.475  0.000  8.786  
w6  0.000 -3.239  0.000 -1.904  0.000  5.053  
v1  3.208  0.000  2.999  0.000  3.933  0.000 13.013  
v2  0.000 -2.484  0.000 -1.490  0.000 -3.382  0.000  6.854  
v3  0.555  0.000 -0.485  0.000  1.049  0.000  0.875  0.000  4.699  
v4  0.000 -1.408  0.000 -1.756  0.000 -0.663  0.000 -1.499  0.000  3.911
```

```
> eigen(lavInspect(fit3, "theta"))$values
```

```
[1] 19.183427 12.752307 11.726917 10.823940  9.555380  8.270761  6.184626  4.599928  4.356615  
[10] -1.076578
```

```
> # Maybe model is wrong, or maybe bad starting values.
```

```
> # Obtain MOM estimates for use as starting values.
```

```
>
```

```
> d1 = as.matrix(coffee[,c(1,3,5,7,9)]) # Measurement set one
```

```
> d2 = as.matrix(coffee[,c(2,4,6,8,10)]) # Measurement set two
```

```
> Phi_hat = cov(d1,d2); Phi_hat
```

```
      w2      w4      w6      v2      v4  
w1 10.186131  6.670427 15.123116 11.928618  8.162688  
w3  6.655075  8.684598 12.766332 11.339975  6.893844  
w5  7.627940  6.536859 16.409548 10.881683  6.290829  
v1  8.347940  7.563392 16.891960 15.024598 10.119975  
v3  4.674573  3.738015  7.650754  6.998216 17.746859
```

```
> # Make it symmetric
```

```
> Phi_hat = (Phi_hat + t(Phi_hat) )/2; Phi_hat
```

```
      w2      w4      w6      v2      v4  
w1 10.186131  6.662751 11.375528 10.138279  6.418631  
w3  6.662751  8.684598  9.651595  9.451683  5.315930  
w5 11.375528  9.651595 16.409548 13.886822  6.970791  
v1 10.138279  9.451683 13.886822 15.024598  8.559095  
v3  6.418631  5.315930  6.970791  8.559095 17.746859
```

```
> eigen(Phi_hat)$values # Is Phi-hat positive definite?
```

```
[1] 50.164191 12.097980  2.925981  1.668071  1.195511
```

```
> Omega1_hat = cov(d1) - Phi_hat
```

```
> Omega2_hat = cov(d2) - Phi_hat
```

```
> eigen(Omega1_hat)$values # Is Omega1_hat positive definite?
```

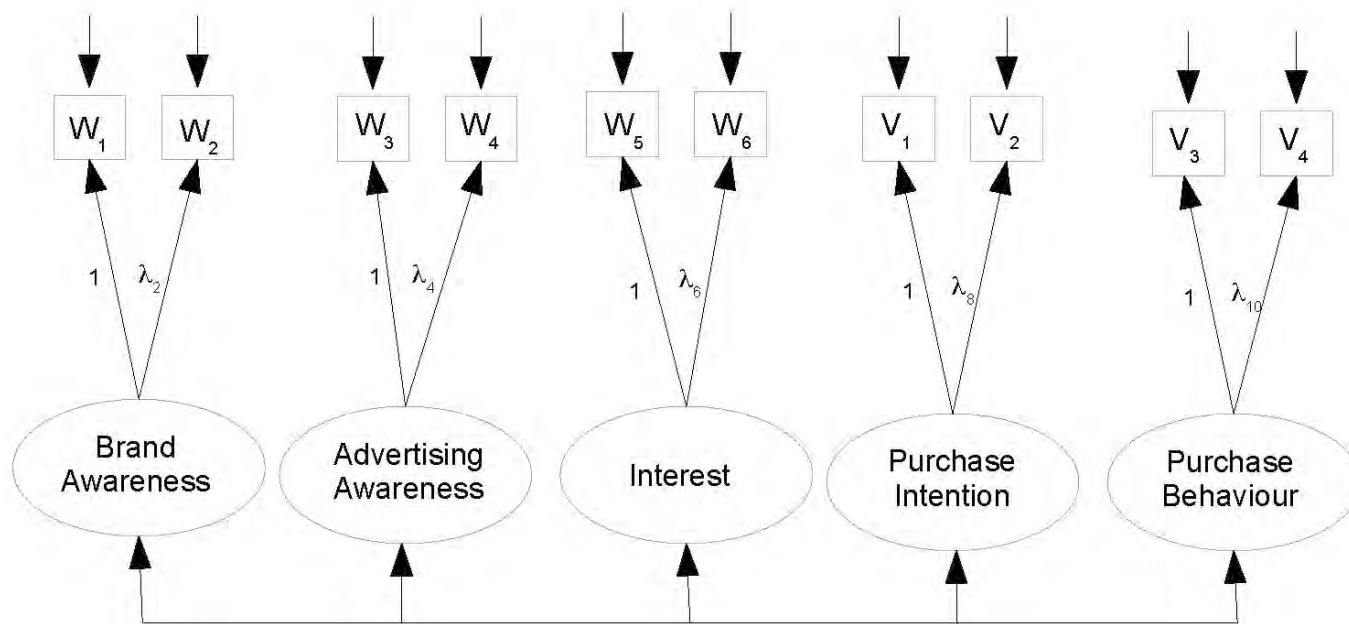
```
[1] 26.402687  9.301147  8.288868  5.106178  2.868356
```

```
> eigen(Omega2_hat)$values # Is Omega2_hat positive definite?
```

```
[1] 12.867799 11.828405  9.847771  4.712254 -3.393667
```

```
> # Method of moments estimate is outside the parameter space
```


Recall that the two measurements of each latent variable are different. One of the interviews is in-person, and the other is by telephone call-back. Maybe they're not really equivalent. Perhaps one in each set (say number two, the call-backs) should have a coefficient not equal to one.



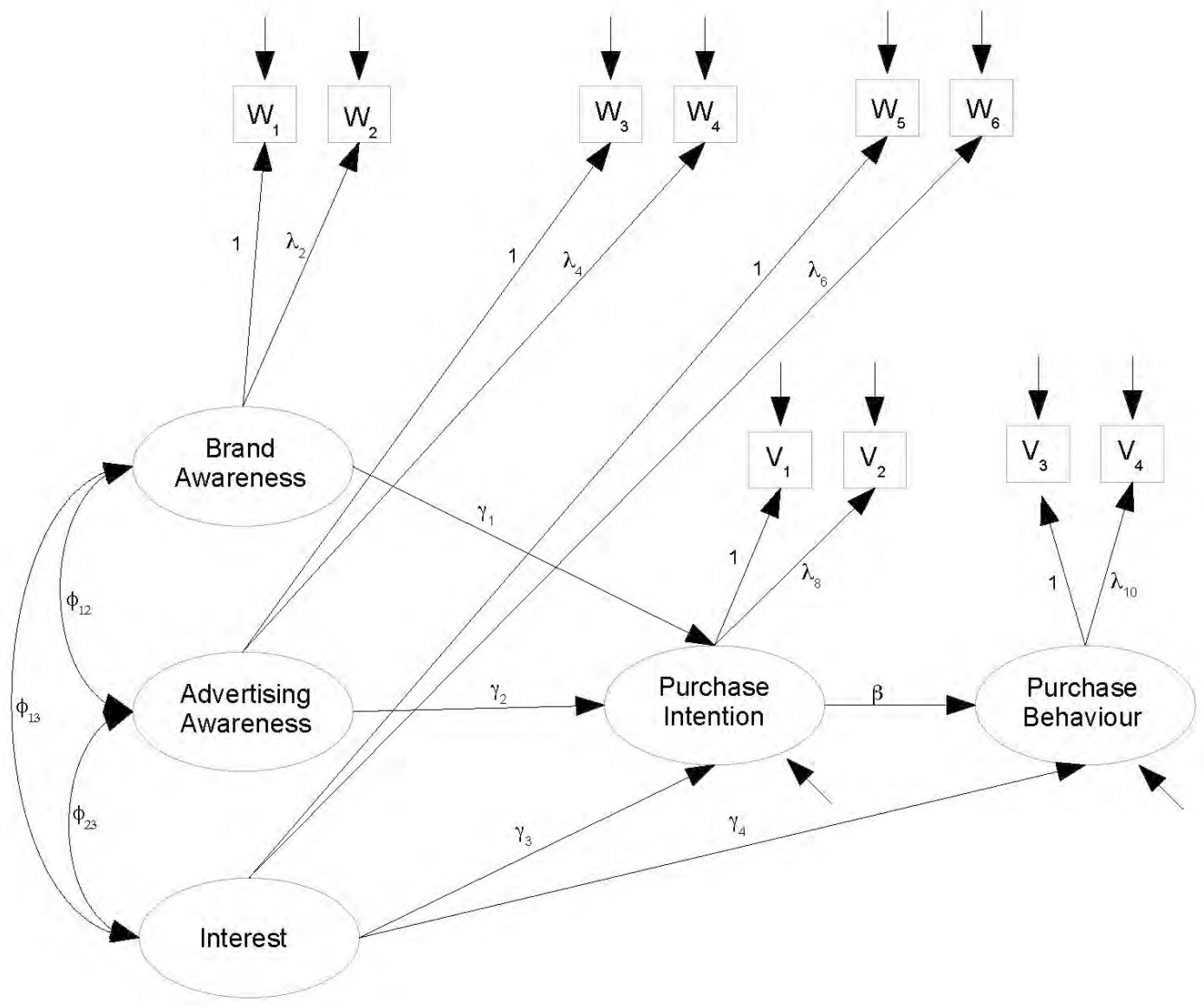
```
> torus4 =
+ '
+ # Measurement model
+   L_BrAw  =~ 1*w1 + lambda2*w2
+   L_AdAw  =~ 1*w3 + lambda4*w4
+   L_Inter =~ 1*w5 + lambda6*w6
+   L_PI    =~ 1*v1 + lambda8*v2
+   L_PBeh  =~ 1*v3 + lambda10*v4
+ ' # End of model torus4
> fit4 = cfa(torus4, data=coffee)
> show(fit4)
```

lavaan 0.6-11 ended normally after 161 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	30
Number of observations	200

Model Test User Model:

Test statistic	17.837
Degrees of freedom	25
P-value (Chi-square)	0.849



```

> # Just edit the measurement model part of model 1
> torus5 =
+ '
+ # Latent variable model
+   L_PI ~ gamma1*L_BrAw + gamma2*L_AdAw + gamma3*L_Inter
+   L_PBeh ~ gamma4*L_Inter + beta*L_PI
+ # Measurement model
+   L_BrAw =~ 1*w1 + lambda2*w2
+   L_AdAw =~ 1*w3 + lambda4*w4
+   L_Inter =~ 1*w5 + lambda6*w6
+   L_PI =~ 1*v1 + lambda8*v2
+   L_PBeh =~ 1*v3 + lambda10*v4
+ # Variances and covariances
+ # Exogenous latent variables
+   L_BrAw ~~ phi11*L_BrAw # Var(L_BrAw) = phi11
+   L_BrAw ~~ phi12*L_AdAw # Cov(L_BrAw,L_AdAw) = phi12
+   L_BrAw ~~ phi13*L_Inter # Cov(L_BrAw,L_Inter) = phi13
+   L_AdAw ~~ phi22*L_AdAw # Var(L_AdAw) = phi22
+   L_AdAw ~~ phi23*L_Inter # Cov(L_AdAw,L_Inter) = phi23
+   L_Inter ~~ phi33*L_Inter # Var(L_Inter) = phi33
+ # Errors in the latent model (epsilons)
+   L_PI ~~ psi1*L_PI # Var(epsilon1) = psi1
+   L_PBeh ~~ psi2*L_PBeh # Var(epsilon2) = psi2
+ # Measurement errors
+   w1 ~~ omega1*w1 # Var(e1) = omega1
+   w2 ~~ omega2*w2 # Var(e2) = omega2
+   w3 ~~ omega3*w3 # Var(e3) = omega3
+   w4 ~~ omega4*w4 # Var(e4) = omega4
+   w5 ~~ omega5*w5 # Var(e5) = omega5
+   w6 ~~ omega6*w6 # Var(e6) = omega6
+   v1 ~~ omega7*v1 # Var(e7) = omega7
+   v2 ~~ omega8*v2 # Var(e8) = omega8
+   v3 ~~ omega9*v3 # Var(e9) = omega9
+   v4 ~~ omega10*v4 # Var(e10) = omega10
+ # Bounds (Variances are positive)
+   phi11 > 0; phi22 > 0; phi33 > 0
+   psi1 > 0; psi2 > 0
+   omega1 > 0; omega2 > 0; omega3 > 0; omega4 > 0; omega5 > 0
+   omega6 > 0; omega7 > 0; omega8 > 0; omega9 > 0; omega10 > 0
+ ' # End of model torus5
> fit5 = lavaan(torus5, data=coffee)

```

Warning messages:

- 1: In lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats, :
lavaan WARNING:
Could not compute standard errors! The information matrix could not be inverted. This may be a symptom that the model is not identified.
- 2: In lav_object_post_check(object) :
lavaan WARNING: covariance matrix of latent variables is not positive definite;
use lavInspect(fit, "cov.lv") to investigate.

```
> summary(fit5)
lavaan 0.6-11 ended normally after 2096 iterations
```

```
Estimator ML
Optimization method NLMINB
Number of model parameters 28
Number of inequality constraints 15

Number of observations 200
```

```
Model Test User Model:
```

```
Test statistic 31.127
Degrees of freedom 27
P-value (Chi-square) 0.266
```

```
Parameter Estimates:
```

```
Standard errors Standard
Information Expected
Information saturated (h1) model Structured
```

```
Latent Variables:
```

	Estimate	Std.Err	z-value	P(> z)
L_BrAw =~				
w1	1.000			
w2 (lmb2)	0.535	NA		
L_AdAw =~				
w3	1.000			
w4 (lmb4)	0.552	NA		
L_Inter =~				
w5	1.000			
w6 (lmb6)	1.094	NA		
L_PI =~				
v1	1.000			
v2 (lmb8)	0.708	NA		
L_PBeh =~				
v3	1.000			
v4 (lmb10)	1.034	NA		

```
(to be continued)
```

```
Comparing the estimates from the good measurement model,
```

```
> coef(fit4)
```

	lambda2	lambda4	lambda6	lambda8	lambda10
	0.530	0.543	1.090	0.708	1.029
w1~~w1	5.106	12.955	7.034	13.401	6.205
w6~~w6	6.134	8.322	10.301	4.440	3.993
L_BrAw~~L_BrAw	19.135	15.914	14.980	21.128	17.155
L_BrAw~~L_AdAw	12.297	13.502	16.248	7.883	11.306
L_AdAw~~L_PI	15.070	6.144	15.564	6.533	9.619

Continuing to look at the output of `summary(fit5)`,

Regressions:

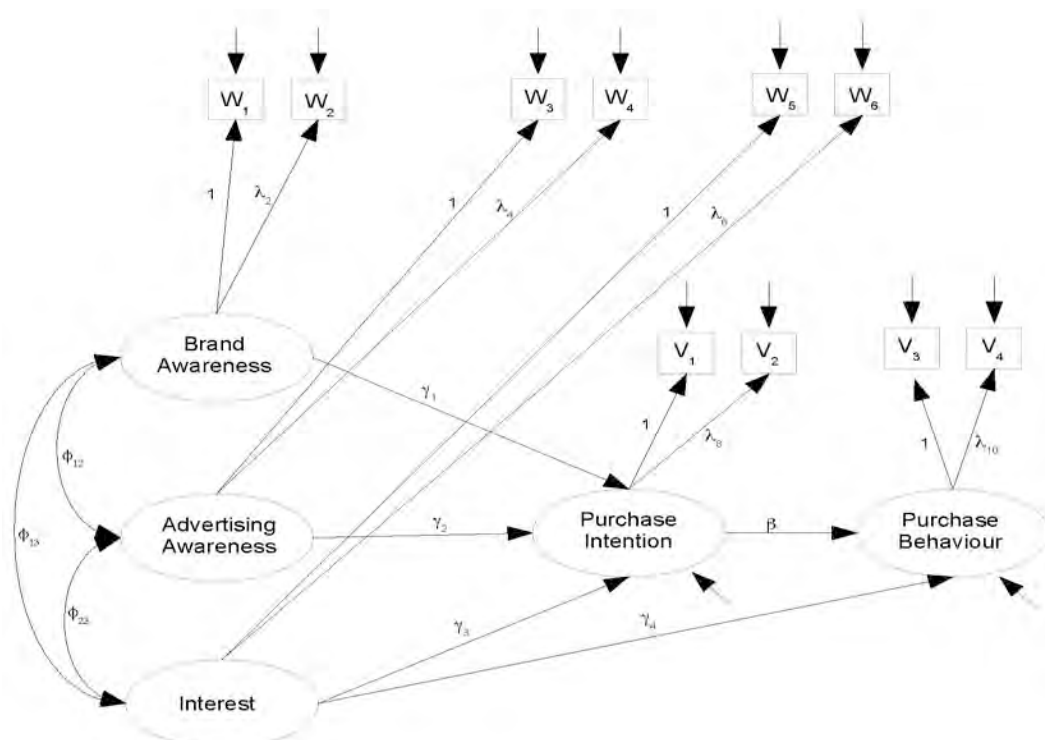
		Estimate	Std.Err	z-value	P(> z)
L_PI ~					
L_BrAw	(gmm1)	47.719	NA		
L_AdAw	(gmm2)	-156.406	NA		
L_Inter	(gmm3)	80.361	NA		
L_PBeh ~					
L_Inter	(gmm4)	-0.156	NA		
L_PI	(beta)	0.570	NA		

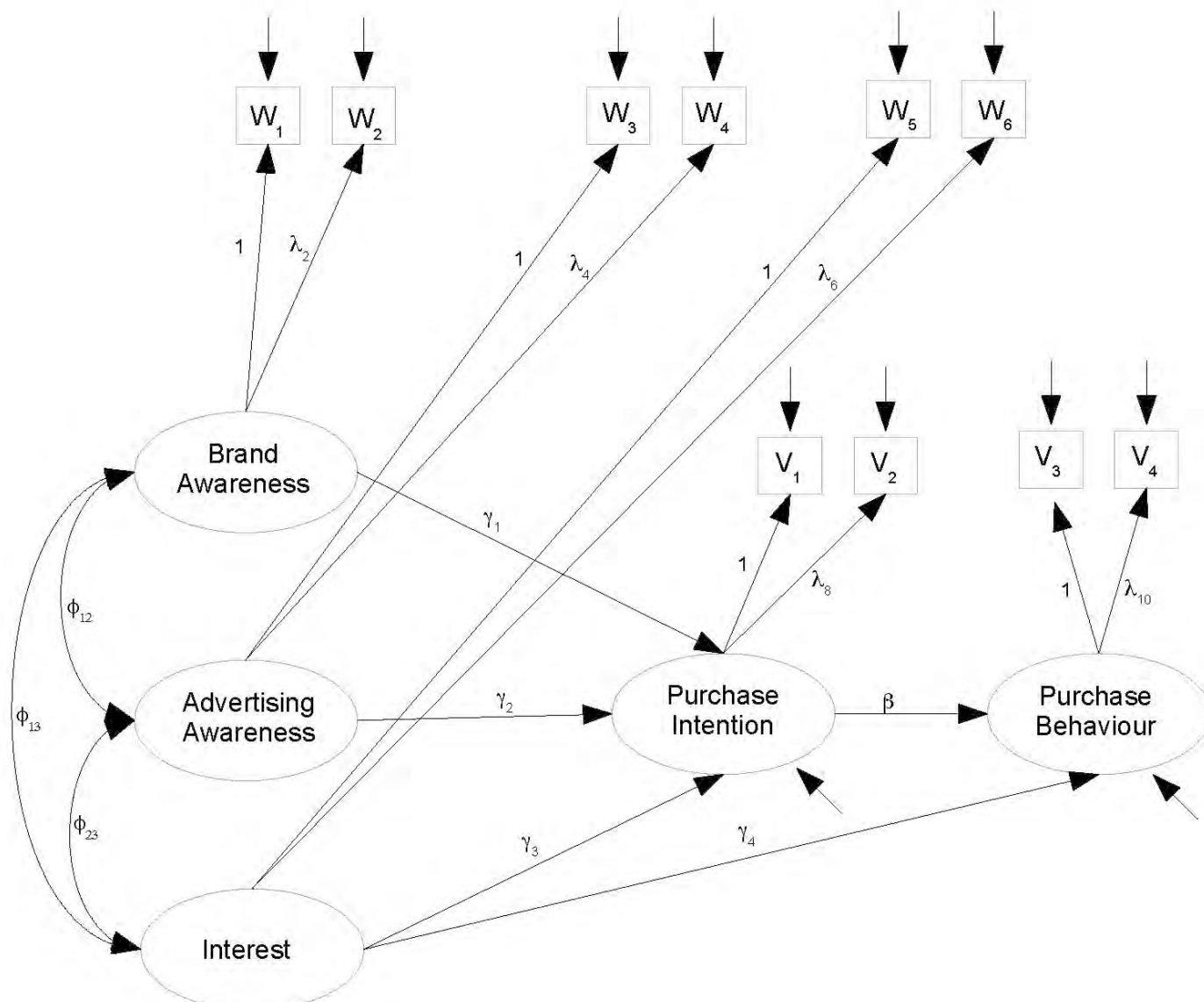
Covariances:

		Estimate	Std.Err	z-value	P(> z)
L_BrAw ~~					
L_AdAw	(ph12)	12.498	NA		
L_Inter	(ph13)	13.407	NA		
L_AdAw ~~					
L_Inter	(ph23)	11.621	NA		

Variances:

		Estimate	Std.Err	z-value	P(> z)
L_BrAw	(ph11)	18.730	NA		
L_AdAw	(ph22)	9.691	NA		
L_Inter	(ph33)	14.851	NA		
.L_PI	(psi1)	260.320	NA		
.L_PBeh	(psi2)	12.623	NA		
.w1	(omg1)	5.511	NA		
.w2	(omg2)	12.959	NA		
.w3	(omg3)	13.263	NA		
.w4	(omg4)	15.139	NA		
.w5	(omg5)	6.335	NA		
.w6	(omg6)	6.158	NA		
.v1	(omg7)	8.341	NA		
.v2	(omg8)	10.301	NA		
.v3	(omg9)	4.524	NA		
.v4	(om10)	3.903	NA		

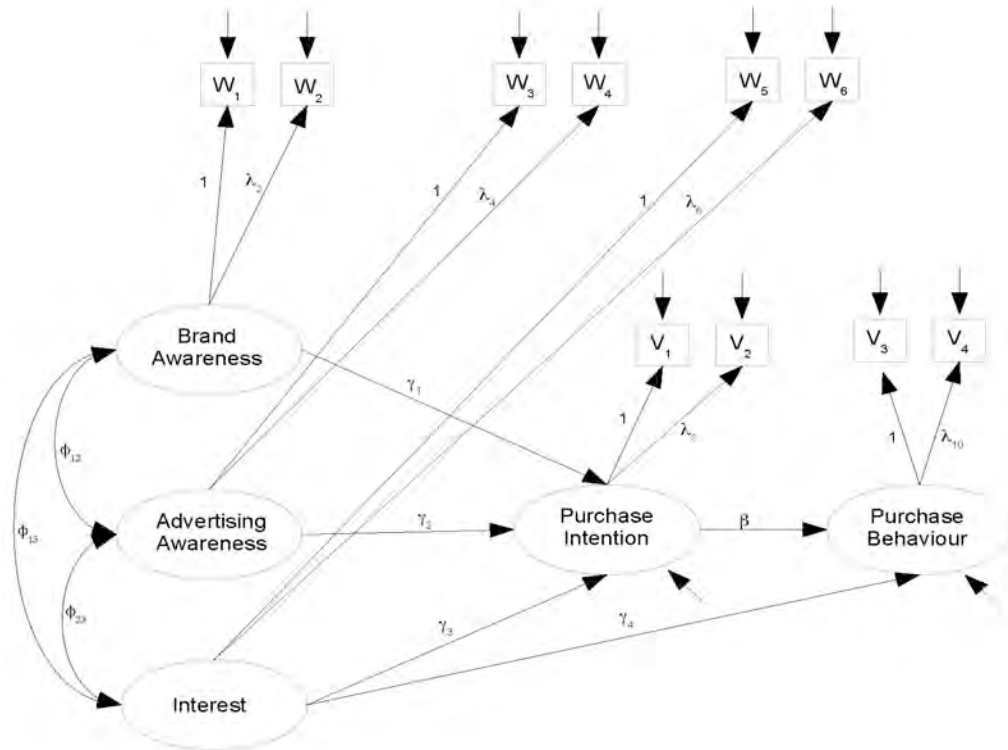




```

> lavInspect(fit5, "cov.lv")
      L_BrAw L_AdAw L_Intr L_PI  L_PBeh
L_BrAw 18.730
L_AdAw 12.498  9.691
L_Inter 13.407 11.621 14.851
L_PI    16.411 14.534 15.565 21.059
L_PBeh  7.261  6.469  6.554  9.572 17.054
> lavInspect(fit4, "cov.lv") # True MLE
      L_BrAw L_AdAw L_Intr L_PI  L_PBeh
L_BrAw 19.135
L_AdAw 12.297 15.914
L_Inter 13.502 11.306 14.980
L_PI    16.248 15.070 15.564 21.128
L_PBeh  7.883  6.144  6.533  9.619 17.155
>
> # Believe the values from fit4. They are the real MLEs

```



```
# Based on the sub-model  $y_l = \gamma^t x + \epsilon_l$  (details omitted)
# Get estimates of gamma and psil = Var( $\epsilon_l$ )
```

```
>
> # The names of all these quantities should include "hat."
> Phi = lavInspect(fit4, "cov.lv")
> Phix = Phi[1:3,1:3]; Phix
      L_BrAw  L_AdAw  L_Inter
L_BrAw 19.13510 12.29660 13.50213
L_AdAw 12.29660 15.91372 11.30579
L_Inter 13.50213 11.30579 14.98033
> Phixy = as.matrix(Phi[1:3,4]); Phixy
      [,1]
L_BrAw 16.24761
L_AdAw 15.07005
L_Inter 15.56443
> gamma = t(Phixy) %*% solve(Phix); gamma
      L_BrAw  L_AdAw  L_Inter
[1,] 0.1996458 0.3932861 0.5622287
> psil = Phi[4,4] - as.numeric(gamma %*% Phix %*% t(gamma)); psil
[1] 3.206661
>
> # These numbers are much more reasonable. See if I can get away with specifying just 10
starting values. Drop the inequality constraints too, since lavaan will issue a warning if
any variance estimate is negative.
```

```

>
> torus6 =
+ '
+ # Latent variable model
+   L_PI ~ gamma1*L_BrAw + start(0.1996458)*L_BrAw +
+   gamma2*L_AdAw      + start(0.3932861)*L_AdAw +
+   gamma3*L_Inter     + start(0.5622287)*L_Inter
+   L_PBeh ~ gamma4*L_Inter + beta*L_PI
+ # Measurement model
+   L_BrAw =~ 1*w1 + lambda2*w2
+   L_AdAw =~ 1*w3 + lambda4*w4
+   L_Inter =~ 1*w5 + lambda6*w6
+   L_PI   =~ 1*v1 + lambda8*v2
+   L_PBeh =~ 1*v3 + lambda10*v4
+ # Variances and covariances
+   # Exogenous latent variables
+   L_BrAw ~~ phi11*L_BrAw + start(19.13510)*L_BrAw # Var(L_BrAw) = phi11
+   L_BrAw ~~ phi12*L_AdAw + start(12.29660)*L_AdAw # Cov(L_BrAw,L_AdAw) = phi12
+   L_BrAw ~~ phi13*L_Inter + start(13.50213)*L_Inter # Cov(L_BrAw,L_Inter) = phi13
+   L_AdAw ~~ phi22*L_AdAw + start(15.91372)*L_AdAw # Var(L_AdAw) = phi22
+   L_AdAw ~~ phi23*L_Inter + start(11.30579)*L_Inter # Cov(L_AdAw,L_Inter) = phi23
+   L_Inter ~~ phi33*L_Inter + start(14.98033)*L_Inter # Var(L_Inter) = phi33
+   # Errors in the latent model (epsilons)
+   L_PI ~~ psi1*L_PI + start(3.206661)*L_PI # Var(epsilon1) = psi1
+   L_PBeh ~~ psi2*L_PBeh # Var(epsilon2) = psi2
+   # Measurement errors
+   w1 ~~ omega1*w1 # Var(e1) = omega1
+   w2 ~~ omega2*w2 # Var(e2) = omega2
+   w3 ~~ omega3*w3 # Var(e3) = omega3
+   w4 ~~ omega4*w4 # Var(e4) = omega4
+   w5 ~~ omega5*w5 # Var(e5) = omega5
+   w6 ~~ omega6*w6 # Var(e6) = omega6
+   v1 ~~ omega7*v1 # Var(e7) = omega7
+   v2 ~~ omega8*v2 # Var(e8) = omega8
+   v3 ~~ omega9*v3 # Var(e9) = omega9
+   v4 ~~ omega10*v4 # Var(e10) = omega10
+ ' # End of model torus6
> fit6 = lavaan(torus6, data=coffee)
> fit6

```

lavaan 0.6-11 ended normally after 108 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	28
Number of observations	200

Model Test User Model:

Test statistic	18.962
Degrees of freedom	27
P-value (Chi-square)	0.871


```
> summary(fit6)
lavaan 0.6-11 ended normally after 108 iterations
```

```
Estimator ML
Optimization method NLMINB
Number of model parameters 28

Number of observations 200
```

Model Test User Model:

```
Test statistic 18.962
Degrees of freedom 27
P-value (Chi-square) 0.871
```

Parameter Estimates:

```
Standard errors Standard
Information Expected
Information saturated (h1) model Structured
```

Latent Variables:

	Estimate	Std.Err	z-value	P(> z)
L_BrAw =~				
.w1	1.000			
.w2 (lmb2)	0.528	0.077	6.861	0.000
L_AdAw =~				
.w3	1.000			
.w4 (lmb4)	0.543	0.090	6.013	0.000
L_Inter =~				
.w5	1.000			
.w6 (lmb6)	1.092	0.081	13.528	0.000
L_PI =~				
.v1	1.000			
.v2 (lmb8)	0.707	0.066	10.745	0.000
L_PBeh =~				
.v3	1.000			
.v4 (lm10)	1.040	0.110	9.457	0.000

Regressions:

	Estimate	Std.Err	z-value	P(> z)
L_PI ~				
L_BrAw (gmm1)	0.229	0.145	1.581	0.114
L_AdAw (gmm2)	0.369	0.161	2.285	0.022
L_Inter (gmm3)	0.553	0.170	3.253	0.001
L_PBeh ~				
L_Inter (gmm4)	-0.129	0.257	-0.502	0.615
L_PI (beta)	0.546	0.224	2.438	0.015

Covariances:

	Estimate	Std.Err	z-value	P(> z)
L_BrAw ~~				
L_AdAw (ph12)	12.301	1.864	6.598	0.000
L_Inter (ph13)	13.480	1.831	7.360	0.000
L_AdAw ~~				
L_Inter (ph23)	11.312	1.694	6.679	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z)
L_BrAw (ph11)	19.200	3.110	6.174	0.000
L_AdAw (ph22)	15.910	3.033	5.246	0.000
L_Inter (ph33)	14.961	2.153	6.949	0.000
.L_PI (psi1)	3.301	1.340	2.463	0.014
.L_PBeh (psi2)	12.620	2.097	6.019	0.000
.w1 (omg1)	5.041	2.075	2.430	0.015
.w2 (omg2)	12.974	1.413	9.179	0.000
.w3 (omg3)	7.038	2.218	3.172	0.002

```
.w4      (omg4)    13.400    1.477    9.074    0.000
.w5      (omg5)    6.224    0.960    6.484    0.000
.w6      (omg6)    6.098    1.063    5.735    0.000
.v1      (omg7)    8.280    1.479    5.598    0.000
.v2      (omg8)    10.299   1.215    8.477    0.000
.v3      (omg9)    4.612    1.682    2.742    0.006
.v4      (om10)   3.809    1.789    2.129    0.033
```

```
> parTable(fit6)
```

id	lhs	op	rhs	user	block	group	free	ustart	exo	label	plabel	start	est	se
1	L_PI	~	L_BrAw	1	1	1	1	0.200	0	gamma1	.p1.	0.200	0.229	0.145
2	L_PI	~	L_AdAw	1	1	1	2	0.393	0	gamma2	.p2.	0.393	0.369	0.161
3	L_PI	~	L_Inter	1	1	1	3	0.562	0	gamma3	.p3.	0.562	0.553	0.170
4	L_PBeh	~	L_Inter	1	1	1	4	NA	0	gamma4	.p4.	0.000	-0.129	0.257
5	L_PBeh	~	L_PI	1	1	1	5	NA	0	beta	.p5.	0.000	0.546	0.224
6	L_BrAw	==	w1	1	1	1	0	1.000	0		.p6.	1.000	1.000	0.000
7	L_BrAw	==	w2	1	1	1	6	NA	0	lambda2	.p7.	0.476	0.528	0.077
8	L_AdAw	==	w3	1	1	1	0	1.000	0		.p8.	1.000	1.000	0.000
9	L_AdAw	==	w4	1	1	1	7	NA	0	lambda4	.p9.	0.421	0.543	0.090
10	L_Inter	==	w5	1	1	1	0	1.000	0		.p10.	1.000	1.000	0.000
11	L_Inter	==	w6	1	1	1	8	NA	0	lambda6	.p11.	0.724	1.092	0.081
12	L_PI	==	v1	1	1	1	0	1.000	0		.p12.	1.000	1.000	0.000
13	L_PI	==	v2	1	1	1	9	NA	0	lambda8	.p13.	0.594	0.707	0.066
14	L_PBeh	==	v3	1	1	1	0	1.000	0		.p14.	1.000	1.000	0.000
15	L_PBeh	==	v4	1	1	1	10	NA	0	lambda10	.p15.	0.807	1.040	0.110
16	L_BrAw	~~	L_BrAw	1	1	1	11	19.135	0	phi11	.p16.	19.135	19.200	3.110
17	L_BrAw	~~	L_AdAw	1	1	1	12	12.297	0	phi12	.p17.	12.297	12.301	1.864
18	L_BrAw	~~	L_Inter	1	1	1	13	13.502	0	phi13	.p18.	13.502	13.480	1.831
19	L_AdAw	~~	L_AdAw	1	1	1	14	15.914	0	phi22	.p19.	15.914	15.910	3.033
20	L_AdAw	~~	L_Inter	1	1	1	15	11.306	0	phi23	.p20.	11.306	11.312	1.694
21	L_Inter	~~	L_Inter	1	1	1	16	14.980	0	phi33	.p21.	14.980	14.961	2.153
22	L_PI	~~	L_PI	1	1	1	17	3.207	0	psi1	.p22.	3.207	3.301	1.340
23	L_PBeh	~~	L_PBeh	1	1	1	18	NA	0	psi2	.p23.	0.050	12.620	2.097
24	w1	~~	w1	1	1	1	19	NA	0	omega1	.p24.	12.120	5.041	2.075
25	w2	~~	w2	1	1	1	20	NA	0	omega2	.p25.	9.162	12.974	1.413
26	w3	~~	w3	1	1	1	21	NA	0	omega3	.p26.	11.474	7.038	2.218
27	w4	~~	w4	1	1	1	22	NA	0	omega4	.p27.	9.046	13.400	1.477
28	w5	~~	w5	1	1	1	23	NA	0	omega5	.p28.	10.593	6.224	0.960
29	w6	~~	w6	1	1	1	24	NA	0	omega6	.p29.	11.965	6.098	1.063
30	v1	~~	v1	1	1	1	25	NA	0	omega7	.p30.	14.725	8.280	1.479
31	v2	~~	v2	1	1	1	26	NA	0	omega8	.p31.	10.439	10.299	1.215
32	v3	~~	v3	1	1	1	27	NA	0	omega9	.p32.	10.797	4.612	1.682
33	v4	~~	v4	1	1	1	28	NA	0	omega10	.p33.	11.085	3.809	1.789

```
> # Likelihood ratio test of
```

```
> # H0: lambda2 = lambda4 = lambda6 = lambda8 = lambda10 = 1
```

```
> anova(fit1,fit6)
```

```
Chi-Squared Difference Test
```

	Df	AIC	BIC	Chisq	Chisq diff	Df diff	Pr(>Chisq)
fit6	27	10947	11039	18.962			
fit1	32	10996	11071	77.752	58.789	5	0.00000000002162 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
> # Which ones are different from one?
```

```
> pt6 = parTable(fit6); dim(pt6)
```

```
[1] 33 15
```

```
> z = as.numeric( (pt6[,14]-1)/pt6[,15] )
```

```
> # Extract only meaningful z statistics (for lambda_j)
```

```
> z = z[c(7,9,11,13,15)]
```

```
> names(z) = c('lambda2', 'lambda4', 'lambda6', 'lambda8', 'lambda10')
```

```
> z
```

```
lambda2 lambda4 lambda6 lambda8 lambda10
-6.1368432 -5.0581710 1.1367154 -4.4540676 0.3614714
```

```
> pt6[c(7,9,11,13,15),14] # Corresponding lambda-hats
```

```
[1] 0.5278696 0.5431214 1.0917385 0.7069418 1.0397428
```