

Chapter Answers

△ Chapter 1: Introduction and Overview

< No Exercises or Answers >

△ Chapter 2: Multivariate Statistics: Issues and Assumptions

< No Exercises or Answers >

△ Chapter 3: Hotelling's T^2 : A Two-Group Multivariate Analysis

1. Create and merge two data vectors

```
> x = c(1,7,13,4,9,6,7,8,9,20)
> y = c(16,12,8,19,20,11,12,23,14,25)
> z = matrix(rbind(x,y),10,2)
> z
```

2. Create membership vector with two groups

```
> grp = matrix(c(1,1,1,1,1,2,2,2,2,2),10,)
> grp
```

3. Conduct Hotelling T^2 for data

```
> factor(grp)
> HotellingsT2 (formula = z ~ grp)
```

△ Chapter 4: Multivariate Analysis of Variance

1. One-Way MANOVA

```
# Input Baumann data from car library
> library(car)
> attach(Baumann)

# ? Baumann for description of dependent and independent variables

> ?Baumann

# Run Oneway MANOVA model with summary statistics

> group = factor(group)    # Set group as factor with levels: Basal,
DRTA, Strat
> Y = cbind(post.test.1,post.test.2,post.test.3) # Combine dependent
variables
> fit = manova(Y~group)

# Compute MANOVA summary statistics for Wilks, Pillai, Hotelling-
Lawley, and Roy

> summary(fit,test="Wilks")
> summary(fit,test="Pillai")
> summary(fit,test="Hotelling-Lawley")
> summary(fit,test="Roy")

# Compute means by groups

> describeBy(Baumann,group)
```

Results: The Basal, DRTA, and Strat groups differed on the joint means for the three posttest measures. The joint means were Basal (17.76), DRTA (20.91), and Strat (20.13), so Basal is statistically different from the other two groups.

2. Factorial MANOVA

```
# MANOVA for a randomized block design
# Input Soils data from the car library
> library(car)
```

```

> data()           # List data sets in car package
> attach(Soils)    # Access data set

# See ?Soils for description of variables in the data set and references

> ?Soils           # Describes variables in data set

Dependent variables are: pH,N,Dens,P,Ca,Mg,K,Na,Conduc
Independent variables are: Block, Contour, Depth

# Run MANOVA model

> options(scipen=999)
> soils.mod <- lm(cbind(pH,N,Dens,P,Ca,Mg,K,Na,Conduc) ~ Block +
Contour + Depth +
+ Contour*Depth -1, data=Soils)

# Compute MANOVA summary statistics for Wilks, Pillai, Hotelling-
Lawley, and Roy
> Manova(soils.mod, type=c("III"),test=("Wilks"))
> Manova(soils.mod, type=c("III"),test=("Pillai"))
> Manova(soils.mod, type=c("III"),test=("Hotelling-Lawley"))
> Manova(soils.mod, type=c("III"),test=("Roy"))

```

3. List all data sets in R packages

```

(a) data(package = .packages(all.available = TRUE)) # Lists all data
sets in R Packages

```

△ Chapter 5: Multivariate Analysis of Covariance

< NO EXERCISES OR ANSWERS >

△ Chapter 6: Multivariate Repeated Measures

1. The three assumptions to be met are independent observations, sphericity, multivariate normality.
2. Two advantages of multivariate repeated measures over paired t tests is controlling for Type I error rate, so it has more power; and subjects are their own control, so requires fewer subjects.

3. Sphericity is when correlations among the repeated measures are too high. Sphericity requires that the variance of the differences in pairs of repeated measures be equal.
4. Difference scores provide a control for sphericity. The test of parallelism or groups being similar across time is conducted on the difference scores in a one-way MANOVA.
5. Given the following data set, *ch5ex3.dat*, conduct a multivariate repeated measures analysis using *lmer4* package and **lmer()** function.

```
# specify directory where file has been saved
> setwd("C:/")
```

```
# read in data set and attach data set
> ch5ex3 = read.table(file="ch5ex3.dat",header=TRUE,sep="\t")
> attach(ch5ex3)           # permits using variable names
> head(ch5ex3)             # check to view data read okay
```

```
      ir.1  ir.2  ir.3  ir.4 gender
1  57.160 74.308 68.592 67.163      1
2  20.006 27.151 37.154 25.722      1
3  35.725 64.305 67.163 60.018      0
4  15.719 40.012 48.586 38.583      1
5  11.432 35.725 31.438 35.725      1
6  24.293 42.870 47.157 38.583      0
```

```
# Descriptive statistics
```

```
> install.packages("psych")
> library(psych)
> describeBy(ch5ex3,gender)
```

```
group: 0
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
ir.1	1	50	28.72	15.01	25.72	27.65	11.65	1.43	64.31	62.88	0.62	-0.33	2.12
ir.2	2	50	46.76	19.40	48.59	47.09	22.25	10.00	88.60	78.59	-0.13	-0.71	2.74
ir.3	3	50	48.24	19.91	49.30	48.37	20.13	0.00	90.03	90.03	-0.16	-0.50	2.82
ir.4	4	50	44.81	19.76	44.30	44.41	21.19	5.72	91.46	85.74	0.15	-0.55	2.79
gender	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NaN	NaN	0.00

```
group: 1
      vars   n mean    sd median trimmed   mad   min    max range  skew kurtosis   se
ir.1    1 111 31.77 14.63  31.44   30.99 14.83   7.14 78.59 71.45  0.55    0.04 1.39
ir.2    2 111 50.13 17.49  51.44   50.18 19.07 15.72 87.17 71.45 -0.04   -0.98 1.66
ir.3    3 111 53.19 17.53  54.30   53.29 16.95 15.72 92.89 77.17 -0.08   -0.69 1.66
ir.4    4 111 50.27 17.89  51.44   50.24 21.19 14.29 88.60 74.31 -0.05   -1.06 1.70
gender   5 111  1.00  0.00   1.00    1.00  0.00  1.00  1.00  0.00   NaN     NaN 0.00
```

```
# Create person-period data set
```

```
> install.packages("reshape")
> library(reshape)

> ppch5ex3 = melt(ch5ex3,id="gender",measured=c("ir.1","ir.2","ir.3","ir.4"))
> time = cbind(rep(c(1:4),c(161,161,161,161)))
> id = cbind(rep(c(1:161),c(4)))

> ppch5ex3 = data.frame(cbind(id,ppch5ex3,time))
> attach(ppch5ex3)
> head(ppch5ex3,n=10) # Check data was created correctly
```

```
      id gender variable   value time
1     1      1      ir.1  57.160     1
2     2      1      ir.1  20.006     1
3     3      0      ir.1  35.725     1
4     4      1      ir.1  15.719     1
5     5      1      ir.1  11.432     1
6     6      0      ir.1  24.293     1
7     7      1      ir.1  10.003     1
8     8      1      ir.1  34.296     1
9     9      0      ir.1  28.580     1
10    10      1      ir.1  38.583     1
```

```
# Conduct multivariate repeated measures
```

```
> install.packages("lmer4")
> library(lme4)

> lmer.out = lmer(value ~ gender + time + gender * time + (1|id),
data = ppch5ex3)
> anova(lmer.out)
```

```
Analysis of Variance Table
```

	Df	Sum Sq	Mean Sq	F value
gender	1	182.4	182.4	2.2310
time	1	25090.5	25090.5	306.9423
gender:time	1	133.6	133.6	1.6349

```
# significance of F values
```

```
> pgender = 1 - pf(2.2310,1,159)
> pgender
```

```
[1] 0.1372481
```

```
> ptime = 1 - pf(306.9423,1,159)
> ptime
```

```
[1] 0
```

```
> pgender.time = 1 - pf(1.6349,1,159)
> pgender.time
```

```
[1] 0.2028898
```

The multivariate repeated measures summary table provided the F values for the *gender*, *time*, and *gender * time* effects. The *gender * time* effect is a test of parallelism—that is equal profiles between the groups. The F value for the *gender * time* effect is nonsignificant ($F = 1.6349$, $p = .20$); therefore, we conclude that the groups have parallel slopes. The F value for *gender* effect was not statistically significant ($F = 2.2310$, $p = .14$); therefore, we conclude that males and females did not differ in their average induction reasoning. Finally, the F value for *time* was statistically significant ($F = 306.9423$, $p < .00001$); therefore, we conclude that induction reasoning was different across the four testing periods. We would report the means and standard deviations using the basic R commands:

```
# means and standard deviations for time variable
```

```
> mean(ir.1);sd(ir.1)
```

```
[1] 30.82557
```

```
[1] 14.76676
```

```
> mean(ir.2);sd(ir.2)
```

```
[1] 49.08304
[1] 18.11385
```

```
> mean(ir.3);sd(ir.3)
```

```
[1] 51.65702
[1] 18.38516
```

```
> mean(ir.4);sd(ir.4)
```

```
[1] 48.57712
[1] 18.6
```

The induction reasoning means across *time* indicated that for the first three test periods, the means increased, but in the last test period, the mean decreased. This would signify a nonlinear trend in the means. I reran the model using **nlmer()** function for nonlinear mixed models and obtained the same results. I suspect that there was no significant departure from linearity.

Note: The means in the **describeBy()** function above matched those in Raykov and Marcoulides (2008), but the means for the *time* variable are slightly different from theirs (explains the slight difference in *lmer* analysis results from theirs).

△ Chapter 7: Discriminant Analysis

1. (a) Mutually exclusive equal group sizes, (b) normality, (c) equal group variance–covariance, (d) no outliers, and (e) no multicollinearity among independent variables.
2. MANOVA places group membership as independent variable with multiple continuous dependent variables. Discriminant analysis places group membership as the dependent variable with multiple continuous independent variables. The difference is that the dependent variables and independent variables are located on the opposite side of the equation.
3. Conduct a discriminant analysis.

a. Find list of data files, attach file, list first 10 records

```
> data()
> data(amis)
> attach(amis)
```

b. Print first 10 lines of data file

```
> head(amis,10)
```

	speed	period	warning	pair
1	26	1	1	1
2	26	1	1	1
3	26	1	1	1
4	26	1	1	1
5	27	1	1	1
6	28	1	1	1
7	28	1	1	1
8	28	1	1	1
9	28	1	1	1
10	29	1	1	1

c. Run discriminant analysis

```
> myout = lda(period ~ speed + warning)
> myout
```

Call:

```
lda(period ~ speed + warning)
```

Prior probabilities of groups:

	1	2	3
	0.3318715	0.3318715	0.3362570

Group means:

	speed	warning	# Warning was a dichotomous variable
1	37.36429	1.500000	
2	37.46071	1.500000	
3	38.63694	1.519915	

Coefficients of linear discriminants:

	LD1	LD2
speed	0.15330389	-0.03289942
warning	0.06397474	2.03198879

Proportion of trace:

```
LD1    LD2
0.9998 0.0002
```

d. Output group prediction, put in data frame, view first 10 lines

```
> myresult = predict(myout)$class
> myresult = cbind(myresult)
> myprior = cbind (period)
> myout = data.frame(myprior, myresult)
> head(myout,10)
```

```
period myresult
1      1         1
2      1         1
3      1         1
4      1         1
5      1         1
6      1         1
7      1         1
8      1         1
9      1         1
10     1         1
```

e. Assess the accuracy of prediction—total percent correct

```
> count = table(myprior, myresult)
> diag(prop.table(count))
```

```
          1          2          3
0.163209672 0.008415314 0.185018371
```

```
> sum(diag(prop.table(count)))
```

```
[1] 0.3566434
```

f. Show cell counts and proportions

```
> mytable = table(myprior,myresult)
> mytable
> prop.table(mytable)
> mytable
```

```

      myresult
myprior    1    2    3
      1  1377  81 1342
      2  1414  71 1315
      3  1181  95 1561

```

```
> prop.table(mytable)
```

```

      myresult
myprior    1          2          3
      1  0.163209672  0.009600569  0.159061278
      2  0.167595117  0.008415314  0.155861088
      3  0.139978665  0.011259927  0.185018371

```

g. Calculate chi-square for classification accuracy

```
> chisq.test(myprior, myresult)
```

```
Pearson's Chi-squared test
```

```
data: myprior and myresult
X-squared = 52.5642, df = 4, p-value = 0.0000000001051
```

h. Calculate effect size

```

# Effect Size
# Groups must be numeric
# Install and load CCA package
# Canonical r and Bartlett chi-square in CCA package

```

```

> install.packages("CCA")
> library(CCA)

```

```

> DV = cbind(speed,warning)
> outfit = cca(DV,period)
> summary(outfit)

```

Canonical Correlation Analysis - Summary

Canonical Correlations:

```

      CV 1
0.08010192

```

Shared Variance on Each Canonical Variate:

```

      CV 1
0.006416318

```

Bartlett's Chi-Squared Test:

```

          rho^2      Chisq  df          Pr(>X)
CV 1 0.0064163 54.2895791    2 0.000000000001626 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

i. Interpret results

The group membership variable, *period*, indicated three conditions: before warning sign, after warning sign, and sometime later. Speed (*speed*) was measured at 14 different locations (*pair*) with one site having a warning sign and the other no warning sign (*warning* variable). The study investigated whether *speed* and *warning* variables could distinguish between the three conditions (*period*). Group sizes were equal. Group means showed an increase from 37.36 (Period 1), 37.46 (Period 2), to 38.64 (Period 3). Classification accuracy was 36%, which was statistically significant (Pearson chi-square = 52.56, $df = 4$, $p < .0001$). The effect size was $r^2 = .01$, which is a small effect size but statistically significant (Bartlett chi-square = 54.29, $df = 2$, $p < .0001$). Although these findings were statistically significant, a researcher should be cognizant of how large sample sizes inflate the chi-square value.

(sample size was 8,437).

Note: The *amis* data set is in the *boot* library. It contains 8,437 rows and 4 columns. The study was on the effect of warning signs on speeding at 14 locations. The group variable, *period*, represents (1) before warning sign, (2) shortly after warning sign, and (3) sometime later. The *speed* variable was in miles per hour; the *warning* variable was (1) sign present and (2) no sign erected; and *pair* variable was a number from 1 to 14 that indicated the location. Detailed information is available at `> help.search("amis")`.

△ Chapter 8: Canonical Correlation

1. A researcher should first screen his/her data to avoid issues related to multicollinearity, outliers, missing data, and small sample sizes, which affect statistical analyses. The important assumptions in canonical correlation analysis are normally distributed variables, linear continuous variables, and equal variances among the variables. Failure to investigate and correct these data issues and assumptions can affect the results.

2. Discriminant analysis has a single categorical dependent variable, while canonical correlation has multiple linear continuous dependent variables. Discriminant analysis is focused on how well a set of independent variables can predict group membership (dependent variable), while canonical correlation is interested in how well two linear sets of variables are correlated. The two linear sets of variables form a dimension and reflect latent variables.
3. Run several R functions to report the matrices, the canonical correlations, unstandardized loadings, plot of the dimensions, F test of canonical variates, and the standardized canonical loadings.

```
# Install R package
> install.packages("CCA")
> library(CCA)

# Report the Rxx, Ryy, Rxy, and Ryx matrices used in canonical correlation

> matcor(Xvar,Yvar)

$Xcor      # Rxx matrix

      TS      TC
TS  1.0000000 -0.1610515
TC -0.1610515  1.0000000

$Ycor      # Ryy matrix

      BS      BC
BS  1.0000000 0.05106855
BC  0.05106855 1.00000000

$XYcor      # Rxy and Ryx matrices are boldfaced

      TS      TC      BS      BC
TS  1.0000000 -0.1610515 0.75804843 -0.34081317
TC -0.1610515  1.0000000 0.10964393  0.85699823
BS  0.7580484  0.1096439 1.00000000  0.05106855
BC -0.3408132  0.8569982 0.05106855  1.00000000

# Run canonical correlation
```

```

> belly = cc(Xvar,Yvar)
> belly

$cor      # Canonical correlations on two dimensions
[1] 0.9141425 0.7624749

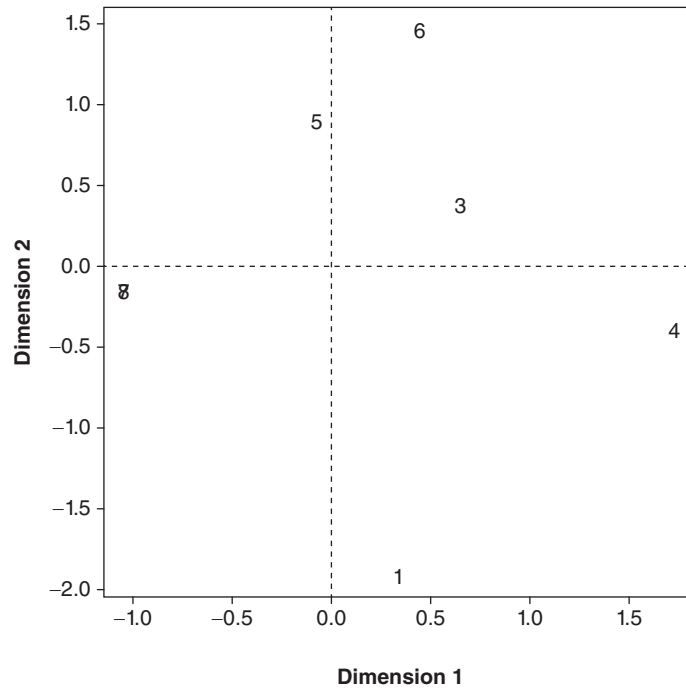
$xcoef    # Unstandardized loadings - X variables
      [,1] [,2]
TS -0.2297895 0.2929561
TC 0.2488132 0.2703732

$ycoef    # Unstandardized loadings - Y variables
      [,1] [,2]
BS -0.1696940 0.3087618
BC 0.3721068 0.1804102

# Plot the dimensions

> plt.cc(belly,type="i")

```



```
# F test of canonical variates
> install.packages("yacca")
> library(yacca)
```

```
> options(scipen=999)
> belly.fit = cca(Xvar,Yvar)
> F.test.cca(belly.fit)
```

F Test for Canonical Correlations (Rao's F Approximation)

	Corr	F	Num df	Den df	Pr(>F)
CV 1	0.91414	5.62496	4.00000	8	0.01870 *
CV 2	0.76247	6.94366	1.00000	5	0.04625 *

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

```
# Compute standardized canonical coefficients
```

```
# Top measured X variables
```

```
> bellyXsd = diag(sqrt(diag(cov(Xvar))))
> outX = matrix(bellyXsd**belly$xcoef,nrow=2,ncol=2,byrow=FALSE,dimnames=
+ list(c("TS","TC"),c("CV1","CV2")))
```

	CV1	CV2
TS	-0.6253363	0.7972344
TC	0.6861160	0.7455690

```
# Bottom measured Y variables
```

```
> bellyYsd = diag(sqrt(diag(cov(Yvar))))
> outY = matrix(bellyYsd**belly$ycoef,nrow=2,ncol=2,byrow=FALSE,dimnames=
+ list(c("BS","BC"),c("CV1","CV2")))
```

	CV1	CV2
BS	-0.4822757	0.8775106
BC	0.9009947	0.4368332

Interpret Results

The canonical correlation analysis indicated that top movement and bottom movement of belly dancers were statistically significantly related on two dimensions. The first canonical variate (dimension) had $r = .91$ ($F = 5.62$,

$df = 4, 8, p = .018$). The first set of canonical loadings indicated that top circle (.68) and bottom circle (.90) were opposite top shimmy (−.62) and bottom shimmy (−.48). The second canonical variate (dimension) had $r = .76$ ($F = 6.94, df = 1, 5, p = .046$). The second set of canonical loadings indicated that top shimmy (.79), top circle (.74), and bottom shimmy (.87) were mostly related, although bottom circle (.43) had a positive weight. The effect sizes for the canonical variates were 83% (eigenvalue = .83), since canonical $r_1 = .91$, and 58% (eigenvalue = .58), since canonical $r_2 = .76$, respectively. The two dimensions overlap, thus not orthogonal. The plot indicates that belly dancers 3 and 6 were high on both dimensions, thus moving and shaking both the top and bottom. Belly dancer 4 was high on the first dimension, so her movements were mostly top and bottom circles.

Note: Interpretation of the belly dancers is directed toward whether they are high or low on the two dimensions. In some cases, they are high on both dimensions or low on both dimensions. The clearer you can be on what the dimensions represent, the clearer the interpretation.

△ Chapter 9: Exploratory Factor Analysis

1. (a) Correlations are not multicollinear (no singularity/identity matrix), (b) correlation matrix is not a nonpositive definite matrix, (c) positive determinant of correlation matrix, (d) adequate sample size, and (e) interitem correlations are positive (reliability).
2. Factor analysis reduces the number of variables into a smaller set of factors. The factors are identified by the common shared variance among the variables. The contribution of each variable is identified by their communality (b^2). Principal components analysis determines components that provide weighting of the observed variables. A component score is derived from the linear weighting of the observed variables.
3. The regression method has a mean = 0 and variance = b^2 (communality estimate). It results in the highest correlation between factor and factor scores. Bartlett method has a mean = 0 and variance = b^2 (same as regression method), but factor scores only correlate with their factor. Anderson–Rubin produces factor scores with mean = 0 and standard deviation = 1. It results in factor scores that are uncorrelated with each other.
4. EFA using Harman.8 data in *psych* package.

```

> install.packages("psych")
> library(psych)
> data(Harman.8) # Correlations of 8 physical
variables, 305 girls (Harman, 1966, 1976)
> Harman.8      # Print correlation matrix

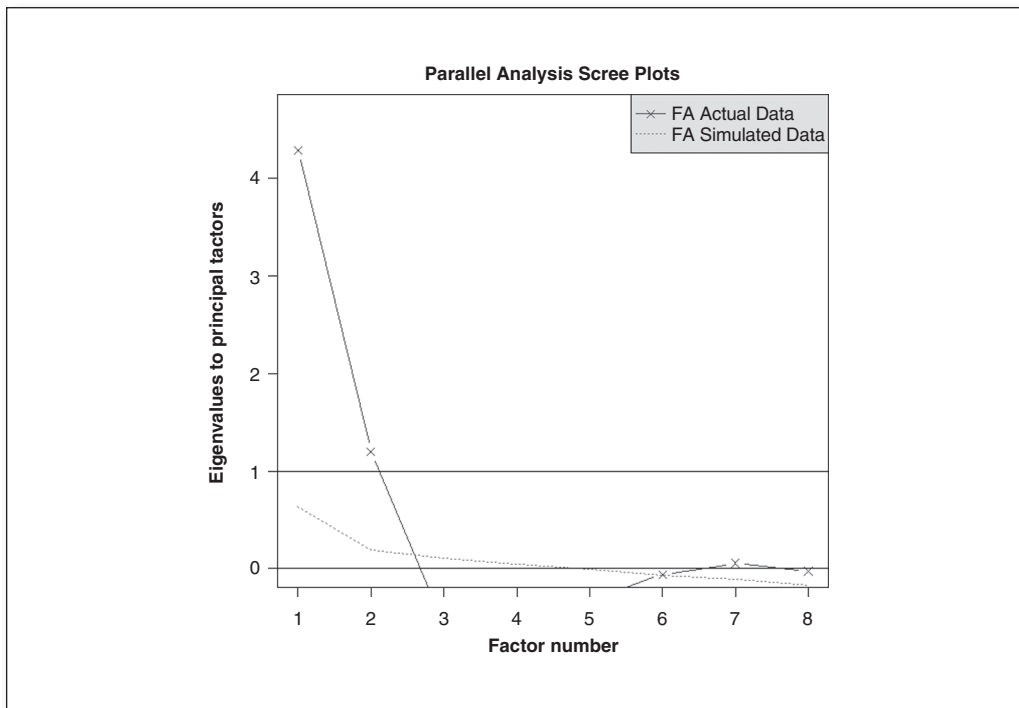
```

1. Run a Scree Plot

```

> fa.parallel(Harman.8, n.obs=305, fm="minres", fa="fa")

```



The scree plot indicates a two factor structure.

2. Run `nfactor = 2` and `nfactor = 3` factor solutions

The `nfactors = 2` shows a common and unique factor as follows:

```

> options(scipen=999) # decimal values rather than
                        scientific notation
> efa2 = fa(Harman.8, nfactors=2, n.obs = 305, rotate="none", fm="minres")

```



```
> print(efa2)
```

```
Factor Analysis using method = minres
```

```
Call: fa(r = Harman.8, nfactors = 2, n.obs = 305, rotate = "none",
  fm = "minres")
```

```
Standardized loadings (pattern matrix) based upon correlation matrix
```

	MR1	MR2	h2	u2	com
Height	0.89	-0.19	0.83	0.17	1.1
Arm span	0.89	-0.31	0.89	0.11	1.2
Forearm	0.86	-0.30	0.83	0.17	1.2
Leg length	0.87	-0.22	0.80	0.20	1.1
Weight	0.67	0.67	0.89	0.11	2.0
Hips	0.56	0.58	0.65	0.35	2.0
Chest girth	0.50	0.59	0.59	0.41	1.9
Chest width	0.56	0.40	0.47	0.53	1.8

	MR1	MR2
Proportion Explained	0.74	0.26

```
Test of the hypothesis that 2 factors are sufficient.
```

```
The total number of observations was 305 with MLE Chi Square = 76.22
with prob < .000001
```

```
Fit based upon off diagonal values = 1
```

The EFA with `nfactors = 3` displays two common factors and a unique factor.

```
> efa3 = fa(Harman.8, nfactors = 3, n.obs = 305, rotate = "none",
  fm = "minres")
> print(efa3)
```

```
Factor Analysis using method = minres
```

```
Call: fa(r = Harman.8, nfactors = 3, n.obs = 305, rotate = "none",
  fm = "minres")
```

```
Standardized loadings (pattern matrix) based upon correlation matrix
```

	MR1	MR2	MR3	h2	u2	com
Height	0.86	0.15	0.33	0.87	0.127	1.4
Arm span	1.00	-0.04	-0.02	1.00	0.005	1.0

```

Forearm      0.89  0.02  0.14  0.81  0.193  1.0
Leg length   0.84  0.12  0.35  0.84  0.157  1.4
Weight       0.41  0.86 -0.02  0.91  0.090  1.4
Hips         0.35  0.71 -0.09  0.64  0.359  1.5
Chest girth  0.30  0.70 -0.12  0.59  0.410  1.4
Chest width  0.43  0.53 -0.21  0.51  0.489  2.3

              MR1  MR2  MR3
Proportion Explained 0.62 0.33 0.05

```

```

Test of the hypothesis that 3 factors are sufficient.
The total number of observations was 305 with MLE Chi Square
= 22.81 with prob < 0.0018
Fit based upon off diagonal values = 1

```

The factor analysis with two common factors and a unique factor more clearly shows a two factor structure indicated by the scree plot.

3. Report results

The 8 physical characteristics of the 305 women can be explained by two factors (constructs). Height, arm span, forearm, and leg length measurements go together (share common variance) and are labeled, *lankiness*. Weight, hip, chest girth, and chest width variables go together (share common variance) and are labeled *stockiness*. Therefore *lankiness* and *stockiness* are two distinguishing characteristics of the 305 girls.

Note: We could output the factor scores on these two factors and create scaled scores from 0 to 100 to provide a meaningful interpretation of the *lankiness* and *stockiness* constructs (traits).

△ Chapter 10: Principal Components Analysis

1. Principal components analysis is a data reduction method designed to explain variable variance in one or more components. It computes eigenvalues that represent the distribution of variable variance across the extracted principal components.
2. Determinant of a matrix is a measure of freedom to vary and indicates whether an inverse matrix is possible to compute eigenvalues and eigenvectors.
3. Eigenvalue is a measure of generalized variance. In principal components analysis, it is the SS loading for each extracted component. The sum of the eigenvalues will equal the sum of the variable variances.

Eigenvectors are the principal component weights used to compute the component scores. It is recommended that the component scores be converted to scaled scores from 0 to 100 for meaningful interpretation.

4. The following R commands produce the summary output for answer.

The determinant of the matrix is positive (13273689529754), the Bartlett chi-square is statistically significant (chi-square = 98.75, $p < .001$), and KMO (.76) is close to 1.0. These three assumptions indicated that it is okay to proceed with principal components analysis (PCA).

The PCA was run with 5 components for the 7 variables. It indicated two eigenvalues > 1 , PC1 (3.72) and PC2 (1.14). This was confirmed by the scree plot. The two components extracted 53% (PC1) and 16% (PC2), with the remaining variance spread across the three remaining components. Cronbach's $\alpha = .84$, which indicates a high level of internal consistency of response.

PC1 comprises rating, complaints, privileges, learning, and raises based on component weights.

PC2 comprises critical and advance based on component weights.

PC1 is named job satisfaction.

PC2 is named negativity toward job.

Note: The sum of the eigenvalues (SS loadings) is equal to the sum of the variances in the diagonal of the variance–covariance matrix.

```
# Chapter 10 exercise
# Install and load packages

> install.packages(psych)
> install.packages("rela")
> install.packages(MASS)
> install.packages("parallel")
> library(psych); library(rela); library(MASS); library(parallel)

> options(scipen=999) # Report decimal values

# Use file.choose() to find attitude file downloaded from website
# Read in file called mydata
# mydata = read.table(file= "C:/attitude.txt",header=TRUE,sep=" ")

# Optional use of file.choose() to open dialog window and select file
```

```
# 7 variable names for 30 subjects
# Variables are: rating, complaints, privileges, learning, raises,
critical, advance
```

```
> mydata = read.table(file.choose(), header = TRUE, sep = " ")
> pcddata = mydata[,-1] # delete id
> pcddata=as.matrix(pcddata)
```

```
# Covariance matrix
```

```
> mycov = cov(pcddata)
> mycov
```

	rating	complaints	privileges	learning	raises	critical	advance
rating	148.171	133.779	63.464	89.105	74.689	18.843	19.423
complaints	133.779	177.283	90.952	93.255	92.641	24.731	30.766
privileges	63.464	90.952	149.706	70.846	56.671	17.825	43.216
learning	89.105	93.255	70.846	137.757	78.139	13.468	64.198
raises	74.689	92.641	56.671	78.139	108.102	38.774	61.423
critical	18.843	24.731	17.825	13.468	38.774	97.909	28.846
advance	19.423	30.766	43.216	64.198	61.423	28.846	105.857

```
# Basic Check on Suitable PCA
```

```
# Use raw data file
```

```
> paf.pc = paf(pcddata,eigcrit=1,convcrit=.001)
> summary(paf.pc)
```

```
$KMO
[1] 0.76328
```

```
$Bartlett
[1] 98.753
```

```
> cortest.bartlett(pcddata,n = 30)
```

```
$chisq
[1] 98.753
```

```
$p.value
[1] 0.0000000000047925
```

```
$df
[1] 21
```

```
> det(mycov)
```

```
[1] 13273689529754
```

```
# Principal Component Analysis
```

```
> pcmodel = principal(mycov, nfactors=5, rotate="none") # no
scores
> pcmodel
```

```
Principal Components Analysis
```

```
Call: principal(r = mycov, nfactors = 5, rotate = "none")
```

```
Standardized loadings (pattern matrix) based upon correlation
matrix
```

	PC1	PC2	PC3	PC4	PC5	h2	u2
rating	0.80	-0.42	0.24	-0.18	0.08	0.91	0.08732
complaints	0.85	-0.36	0.21	0.00	-0.16	0.92	0.08360
privileges	0.68	-0.10	-0.17	0.70	0.00	0.99	0.00553
learning	0.83	-0.05	-0.30	-0.19	0.40	0.97	0.03282
raises	0.86	0.19	0.04	-0.19	-0.32	0.92	0.08262
critical	0.36	0.64	0.65	0.12	0.17	1.00	0.00047
advance	0.58	0.61	-0.46	-0.09	-0.08	0.93	0.06659

	PC1	PC2	PC3	PC4	PC5
SS loadings	3.72	1.14	0.85	0.61	0.32
Proportion Var	0.53	0.16	0.12	0.09	0.05
Cumulative Proportion	0.56	0.73	0.86	0.95	1.00

```
# Cronbach Alpha
```

```
> alpha(mycov)
```

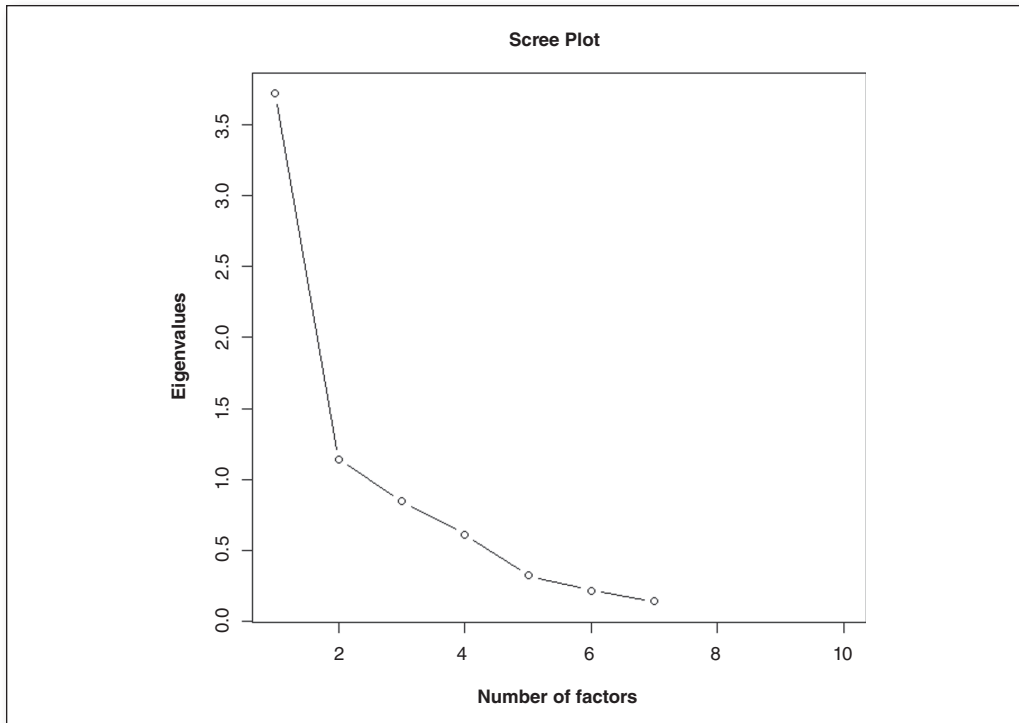
```
Reliability analysis
```

```
Call: alpha(x = mycov)
```

raw_alpha	std.alpha	G6(smc)	average_r	S/N
0.84	0.84	0.88	0.43	5.2

```
# Scree Plot
```

```
> plot(pcmodel$values, type = "b", xlim=c(1,10), main = "Scree
Plot", xlab="Number of
+ Factors", ylab="Eigenvalues")
# Principal Component Eigenvector estimates
```



```
> pcaout = eigen(mycov)    # Place eigenvectors in a file
> V = (pcaout$vectors)     # Put only eigenvector values in a file
> tV = t(pcaout$vectors)   # Transpose of eigenvectors of S
> tV
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	-0.44672	-0.520624	-0.375773	-0.42100	-0.37625	-0.130030	-0.229074
[2,]	0.42184	0.372077	-0.076327	-0.14567	-0.23340	-0.398290	-0.665922
[3,]	-0.24003	-0.143226	0.651322	0.18648	-0.22392	-0.633035	0.109576
[4,]	0.12619	-0.108109	-0.626328	0.48514	0.10410	-0.517067	0.257973
[5,]	0.20102	-0.372379	0.077836	0.62078	-0.44682	0.377987	-0.294907
[6,]	0.47249	0.022005	-0.018141	-0.30156	-0.59329	0.018067	0.576784
[7,]	0.53413	-0.647424	0.173423	-0.23474	0.43742	-0.114743	-0.076591

Note: Each row is the component weights for the linear combination of a variable

```
> Identity = V %*% tV      # Multiply matrix of eigenvector values
                             (vectors) by transpose
> round(Identity,2)
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	1	0	0	0	0	0	0
[2,]	0	1	0	0	0	0	0
[3,]	0	0	1	0	0	0	0
[4,]	0	0	0	1	0	0	0
[5,]	0	0	0	0	1	0	0
[6,]	0	0	0	0	0	1	0
[7,]	0	0	0	0	0	0	1

△ Chapter 11: Multidimensional Scaling

1. The classical or metric MDS analysis enters exact distances in the proximity matrix—for example, distances between cities. The non-metric MDS analysis enters self-reported ordinal distances in the proximity matrix—for example, responses to Likert-type scaled survey questions.
2. The direct method assigns a numerical value to indicate the distance between pairs of objects. The indirect method uses data from subjects who rate pairs of objects to express their perception of similarity or dissimilarity.
3. STRESS is a goodness of fit index with 0 indicating a perfect model fit. It is affected by the number of dimensions expressed in the solution. A value greater than .20 is a poor model fit. It is a subjective measure.
4. The amount of generalized variance explained by the MDS solution can be expressed as P_2 or Mardia criteria. P_2 is the ratio of the sum of the eigenvalues over the total sum of the eigenvalues. Mardia criteria squares the numerator and denominator of the P_2 values. Both P_2 and Mardia criteria are scaled from 0 to 1, with values closer to 1.0 indicating a good fit.
5. The number of dimensions is a critical part of the MDS solution. Too few dimensions and the objects are not distinguished, while too many dimensions would indicate every object as defining its own dimension. The scree plot provides a good indication of the number of eigenvalues greater than 1.0 in the proximity matrix. Dimensions with eigenvalues greater than 1.0 yield significant amounts of explained variance.
6. Classical MDS analysis is conducted as follows:

```

# Install and load packages
> install.packages("MASS")
> install.packages("psy")
> install.packages("psych")

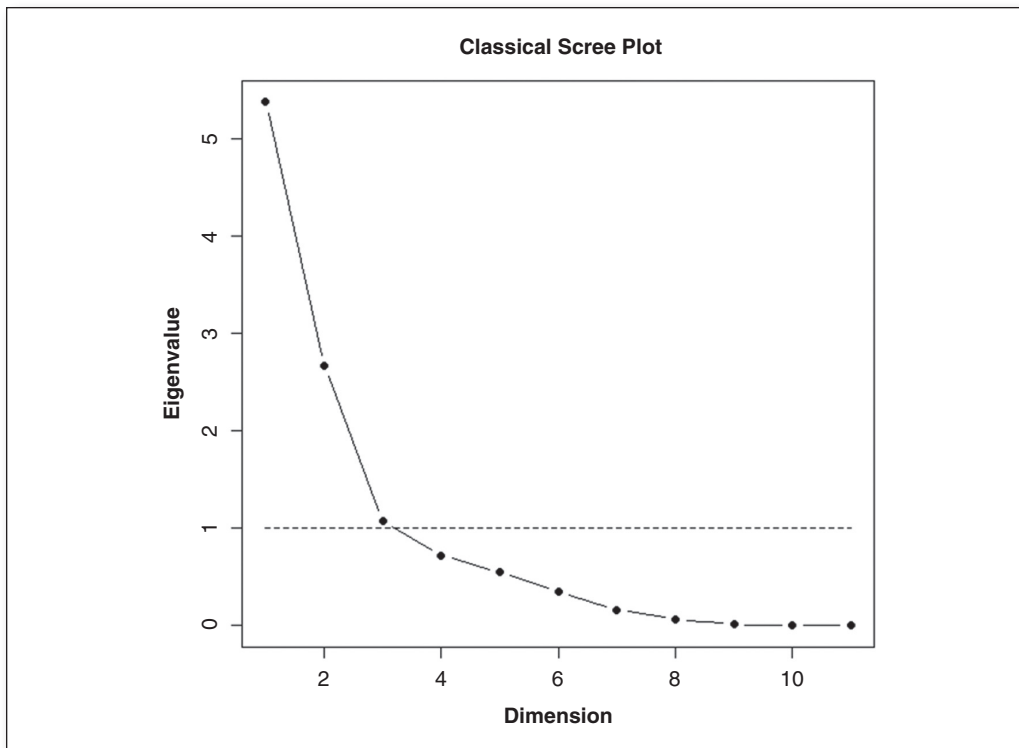
> library(MASS)           # Shepard diagram
> library(psy)            # contains scree.plot() function
> library(psych)         # burt data set
> library(stats)         # dist() and cmdscale() functions
# burt data set - 11 Emotional variables in correlation matrix
# Burt (1915)
# Correlation Matrix

> data(burt)
> burt

# Scree plot from correlation matrix

> scree.plot(burt,title="Classical Scree Plot",type="R")

```




```

# Compute classical MDS
# Uses proximity matrix from dist() function

> distburt = dist(burt) # proximity matrix for cmdscale() function

> burtmds=cmdscale(distburt,eig=TRUE,k=2)
> burtmds

$points
           [,1]      [,2]
Sociality -0.629791610 -0.42784282
Sorrow    -0.187098457 -0.59544628
Tenderness 0.106836888 -0.66117966
Joy        -0.331353569 -0.31114669
Wonder     -0.453914343  0.14491300
Elation    -0.314919790  0.21214452
Disgust     -0.007677537  0.21490414
Anger       -0.385376752  0.70351608
Sex         -0.322669146  0.61657274
Fear        1.009313565 -0.06684368
Subjection  1.516650751  0.17040865

$eig
[1] 4.429698e+00 2.092369e+00 7.858327e-01
[4] 5.533302e-01 3.913281e-01 2.902469e-01
[7] 1.370066e-01 5.646893e-02 1.244370e-02
[10] 4.024411e-04 4.197964e-17

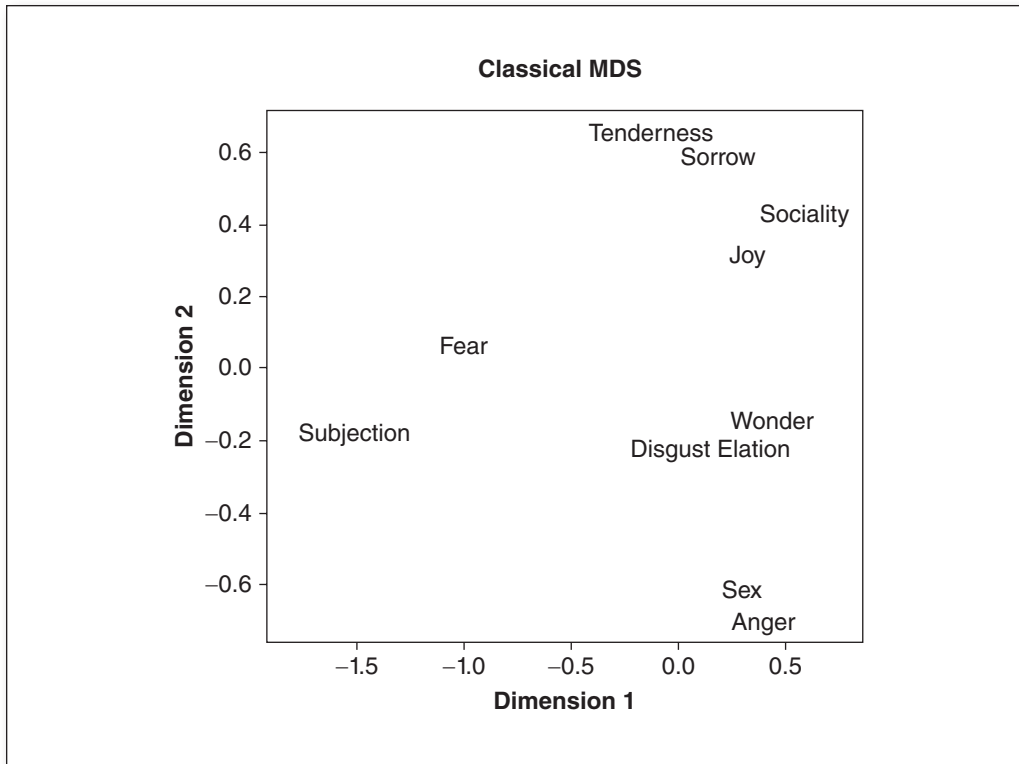
$GOF
[1] 0.7454535 0.7454535

# Plot 11 Emotional variables
# Use negative sign to reverse plot
# Use colnames() function with data set name

> x = cbind (- burtmds$points[,1])
> y = cbind (- burtmds$points[,2])

> plot(x, y, xlab="Dimension 1",ylab="Dimension 2", main = "Classical
+ MDS",type="n",xlim=range(x)*1.2)
> text(x,y,labels = colnames(burt))

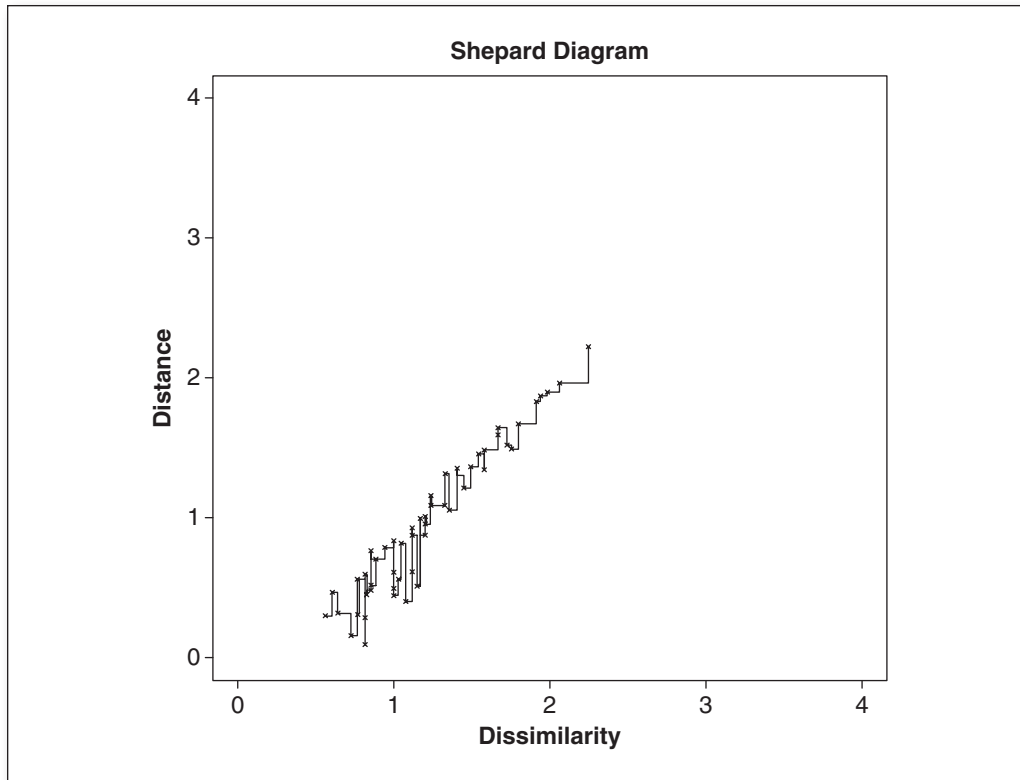
```



```
# Plot Shepard diagram
```

```
> dist_burt = Shepard(distburt,burtmds$points)
> plot(dist_burt, pch="*", xlab= "Dissimilarity", ylab=
"Distance", xlim= c(0,4), ylim= c(0,4)),
+ main="Shepard Diagram")
> lines(dist_burt$x,dist_burt$y,type="s")
```

The burt data set was input as a correlation matrix. The **scree.plot()** function used the burt data set to extract and plot eigenvalues. The scree plot indicated three dimensions—that is, three eigenvalues greater than 1.0. The classical (metric) MDS analysis used the **cmdscale()** function with a proximity matrix and two dimensions. The proximity matrix was created using the **dist()** function. Results indicated that 75% of the variance relation among the 11 emotional variables was explained (P_2 and Mardia criteria = .75). A plot of the two dimensions displayed a separation in the



11 emotional variables. The Shepard diagram indicated a fairly stable monotonic increasing trend along a line of fit.

Note: Would the results be similar if we use a nonMetric MDS with the correlation to distance function in the *psych* package, **cor2dist()** function?

△ Chapter 12: Structural Equation Modeling

1. A nonpositive definite matrix can occur for many reasons, but the basic explanation is that the matrix values do not permit the calculation of parameter estimates. If a matrix has a determinant of zero, then the inverse is zero, and division by zero is inadmissible. Similarly, if the eigenvalues of matrix are zero or negative, then there is no generalized variance and no solution to the set of simultaneous equations.

2. The determinant of a covariance (correlation) matrix yields the generalized variance of the matrix. The generalized variance takes into account the covariance, thus the determinant is the variance minus the covariance. It is calculated by multiplying the row and columns of the covariance matrix by its cofactor values and summing. The trace is the sum of the diagonal values in the matrix, whereas the determinant is the variance–covariance.
3. Eigenvalues are the amount of variance for a specific set of eigenvector weights in a set of simultaneous equations. For example, in factor analysis, more than one factor structure is possible—that is, subset of variables. When a subset is given, each factor has variance (eigenvalue)—that is, sum of the factor loadings squared (communality). The solution, however, is considered indeterminate, because other solutions are possible—that is, other eigenvectors with corresponding eigenvalues. If the rank of a matrix is 3, then there are three nonzero eigenvalues with associated eigenvectors.
4. Observed variables have a scale—that is, mean and standard deviation. A latent variable is created from the observed variables without any scale (reference point). A latent variable by default is assigned a mean = 0 and variance = 1. If an observed variable is assigned to the latent variable, generally by using the value of 1, then the mean and the standard deviation of that observed variable are assigned to the latent variable. The process of assigning the observed variable scale to the latent variable is referred to as reference scaling.

```
# Chapter 12 Growth Model Exercise 5
# 4 time periods - 100 students Math Achievement
```

```
> library(lavaan)

> LGM = data.frame(lower2full
l(c(1,.64,1,.59,.67,1,.45,.57,.62,1),
+ diagonal=TRUE),row.names =c("Math1","Math2","Math3","Math4"))
> names(LGM) = c("Math1","Math2","Math3","Math4")
> LGM = as.matrix(LGM)
> LGM
```

```
      Math1 Math2 Math3 Math4
Math1 1.00 0.64 0.59 0.45
Math2 0.64 1.00 0.67 0.57
Math3 0.59 0.67 1.00 0.62
Math4 0.45 0.57 0.62 1.00
```

```
# sample means

> LGMmn = c(2.3,2.6,2.7,2.9)
> LGMmn

[1] 2.3 2.6 2.7 2.9

# Linear Growth Model
# Model 1 - Intercept and Slope

> LGM.model = `
  i =~ 1*Math1 + 1*Math2 + 1*Math3 + 1*Math4
  s =~ 0*Math1 + 1*Math2 + 2*Math3 + 3*Math4'

> LGM.fit = growth(LGM.model,sample.cov=LGM, sample.nobs=100,
+ sample.mean = LGMmn)

> summary(LGM.fit,standardized=TRUE)
```

The results indicate a good model fit, $\chi^2 = 3.92$, $df = 5$, $p = .56$. The intercept increase and a linear trend in slope values are supported (see column Std.all). The intercept and slope are not correlated significantly.

lavaan (0.5-16) converged normally after 26 iterations

Number of observations	100
------------------------	-----

Estimator	ML
Minimum Function Test Statistic	3.920
Degrees of freedom	5
P-value (Chi-square)	0.561

Parameter estimates:

Information	Expected
Standard Errors	Standard

Estimate	Std.err	Z-value	P(> z)	Std.lv	Std.all
----------	---------	---------	---------	--------	---------

Latent variables:

i =~					
Math1	1.000			0.843	0.829
Math2	1.000			0.843	0.863
Math3	1.000			0.843	0.870
Math4	1.000			0.843	0.827

```

s =~
  Math1      0.000                0.000  0.000
  Math2      1.000                0.206  0.211
  Math3      2.000                0.413  0.426
  Math4      3.000                0.619  0.607

Covariances:
i ~~
s      -0.068  0.043  -1.607  0.108  -0.393  -0.393

Intercepts:
  Math1      0.000                0.000  0.000
  Math2      0.000                0.000  0.000
  Math3      0.000                0.000  0.000
  Math4      0.000                0.000  0.000
  i          2.339  0.097  24.109  0.000  2.774  2.774
  s          0.191  0.033   5.739  0.000  0.924  0.924

Variances:
  Math1      0.324  0.094                0.324  0.313
  Math2      0.337  0.064                0.337  0.354
  Math3      0.332  0.063                0.332  0.353
  Math4      0.356  0.096                0.356  0.342
  i          0.711  0.141                1.000  1.000
  s          0.043  0.020                1.000  1.000

```