



Short Course on Multilevel Modeling
Kellogg School of Management

Paul D. Bliese

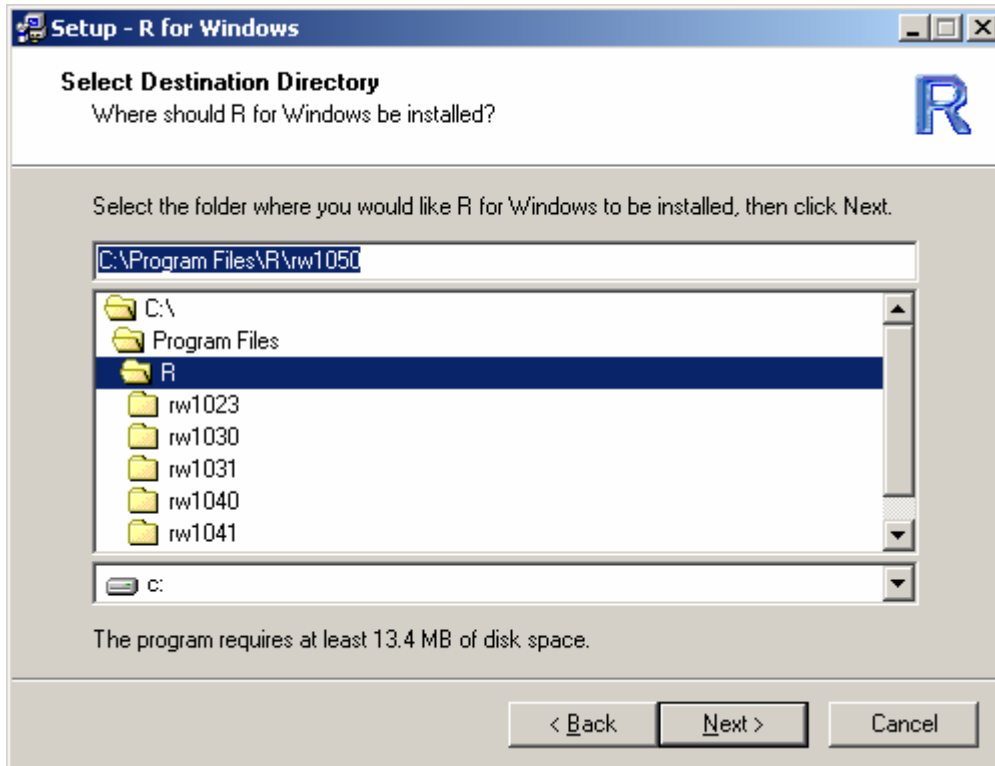
Outline

- I. Introduction and overview to Multilevel Models (PowerPoint)
- II. Introduction to R (PowerPoint)
 - A. Exercise 1: Installing R and the Multilevel package
 - A. Exercise 2: A Sample Session
 - B. Exercise 3: Importing Data from EXCEL and SPSS
- III. Composition Models, Agreement and Reliability (PowerPoint)
 - A. Exercise 4: Estimating Within-Group Agreement and Reliability
- IV. Analytic Methods for multilevel modeling (PowerPoint)
 - A. Exercise 5: Data manipulation Functions for Multilevel Models
 - B. Exercise 6: Contextual Models
 - C. Exercise 7: Covariance Theorem Decomposition (WABA)
 - D. Exercises 8-10: Multilevel Random Coefficient Models
 - Exercise 8: Examining the DV
 - Exercise 9: Adding Level-1 and Level-2 Main Effect Predictors
 - Exercise 10: Cross-level Interactions
- V. Growth Modeling (PowerPoint)
 - A. Exercise 11: Growth Modeling Data Set Up
 - B. Exercise 12: Growth Modeling Example

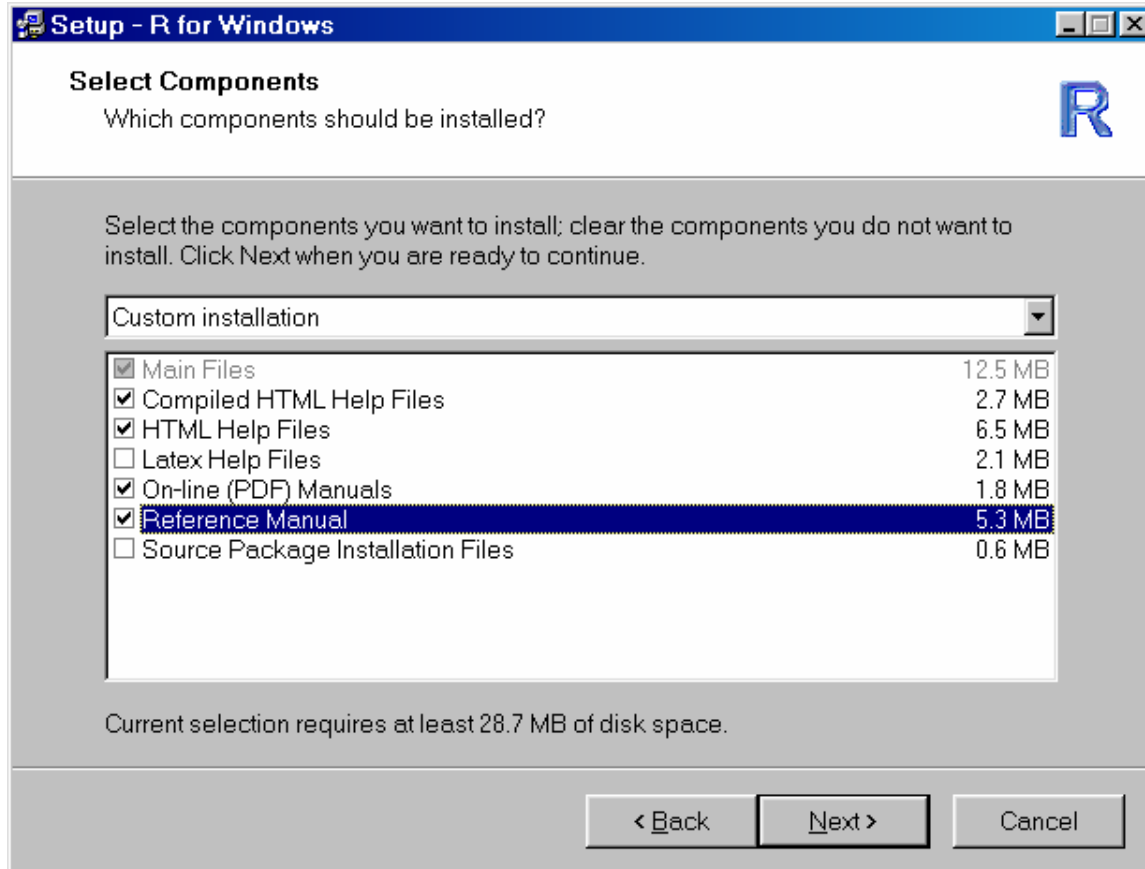
Exercise 1: Installing R and the Multilevel Package

Installing R:

1. Run “SetupR.exe” from the CD.
2. Install R in “Program Files” (the default). Generally, this will be on the C Drive.

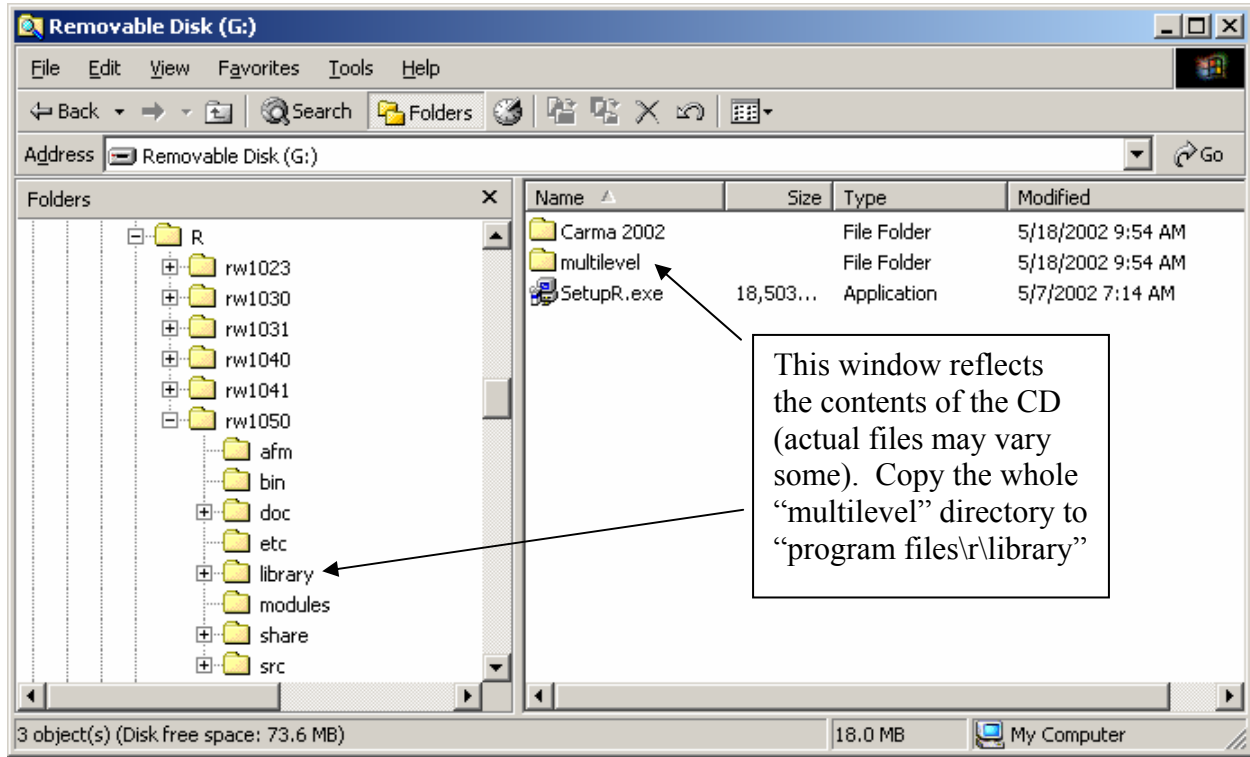


3. Perform a custom installation with the following options:

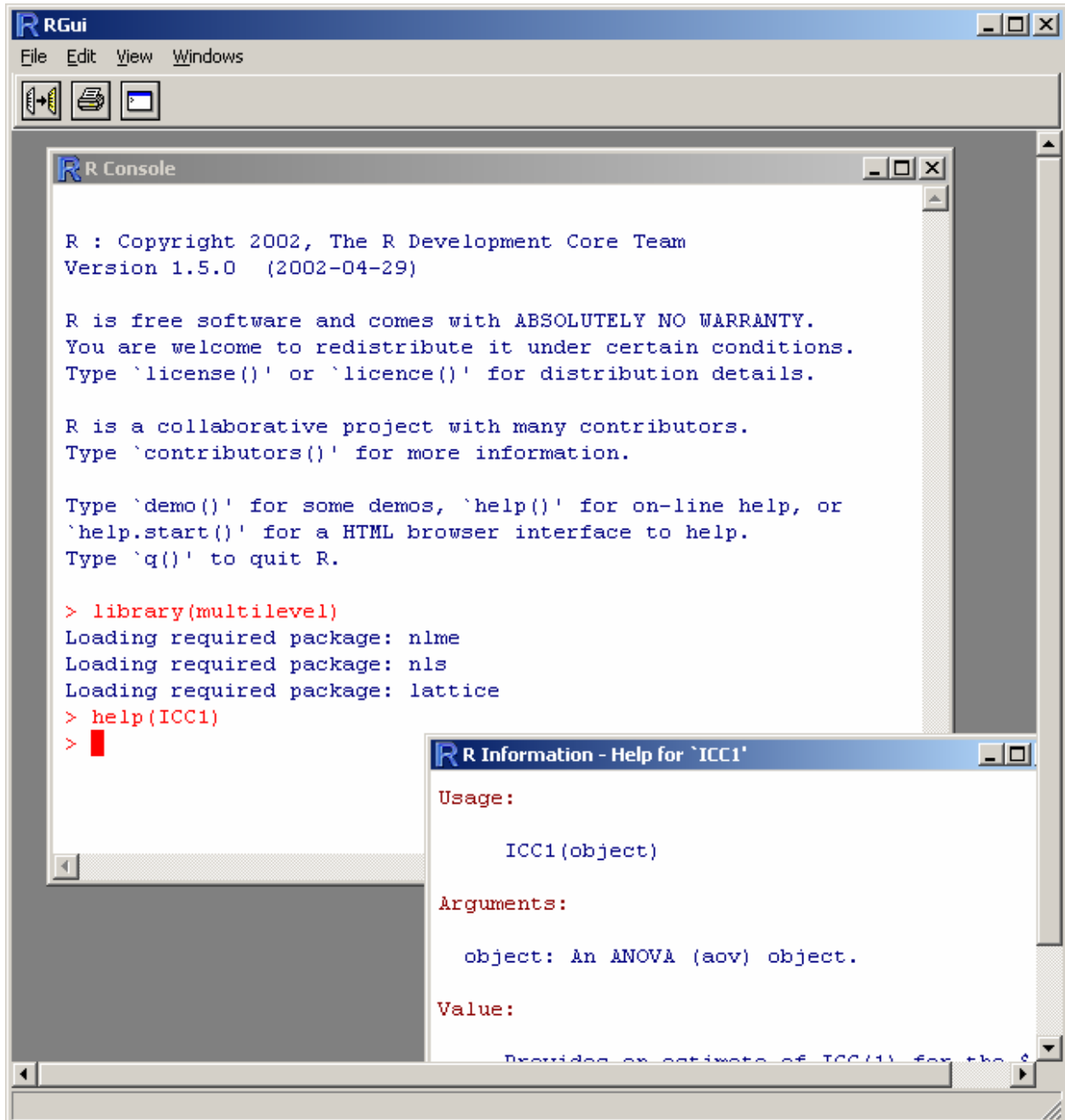


4. Stay with the defaults, and once completed, there should be a R 1.5.0 Shortcut on the Desktop. Hold off on executing it until you complete the next step.

5. To use the multilevel package you need to copy the whole “multilevel” subdirectory from the CD to the library subdirectory in R.



6. Double click on the R GUI on the desktop. Once it provides you with a “>” prompt, type in “library(multilevel)”. If your screen looks like the screen below you are set! Type in “help(ICC1)” to see an example of a help file. NOTE THAT R IS CASE SENSITIVE.



Exercise 2: A Sample Session:

The following session is intended to introduce to you some features of the R environment by using them. Many features of the system will be unfamiliar and puzzling at first, but this puzzlement will soon disappear.

The R program begins, with a banner. (The prompt on the left hand side `>` will not be shown).

Keep in mind that use can use the up arrow key to bring up and edit previous commands. Using this feature will greatly increase your speed!

```
help.start()
    Start the html interface to on-line help (using a web browser available at your
    machine). You should briefly explore the features of this facility with the mouse.

library(multilevel)
    Make the functions and data sets in the multilevel package available

data()
    Examine the data sets that are available in R and its packages

x <- rnorm(50)
y <- x+rnorm(50)
    Generate two correlated pseudo-random numbers

plot(x,y)
    Plot the points in the plane. A graphics window will appear
    automatically.

TDATA<-data.frame(x=x, y=y)
    Put the two variables into a data frame called TDATA.

objects()
    See which R objects are now in the R workspace.

names(TDATA)
    See the names of the elements in TDATA

rm(x,y)
    Remove objects no longer needed. (Clean up).

objects()
    Note that x and y are removed from the R workspace.

search()
    List the elements in the search path

attach(TDATA)
    Put the TDATA data frame in the search path. This lets you access the elements
    of TDATA by name (e.g., "x" and "y")

search()
    List the elements in the search path. TDATA is in position 2.

cor(x,y)
    Calculate the x and y correlation
```

```
detach(TDATA)
```

Take the data frame TDATA out of search position 1.

```
search()
```

Note that TDATA is no longer in the search path.

```
cor(x,y)
```

This no longer works because R can not find x and y.

```
cor(TDATA$x, TDATA$y)
```

Calculate the x and y correlation telling R that x and y are in the TDATA data frame.

```
help(lm)
function.
```

Look at the help files for the linear model (OLS regression)

```
OLS.MOD<-lm(y~x, data=TDATA)
```

Create an ordinary least squares regression object using the lm (linear models) function.

```
summary(OLS.MOD)
```

Summarize the regression model.

```
plot(OLS.MOD)
```

Look at plots of the residuals. Hit return to scroll through the plots.

```
TDATA$z<- .55*TDATA$y+.84*rnorm(50)
```

Create a z variable and put it in the TDATA data frame. The z variable should have a .30 correlation with y.

```
cor(TDATA)
```

Examine the correlation matrix for TDATA.

```
var(TDATA)
```

Examine the variance, covariance matrix for TDATA.

```
OLS.MOD2<-update(OLS.MOD, ~x+z)
```

Update the lm model by adding z to the predictors. This is the same as estimating the model $\text{lm}(y \sim x + z, \text{data} = \text{TDATA})$

```
summary(OLS.MOD2)
```

Summarize the regression model.

```
OLS.MOD3<-update(OLS.MOD2, ~x*z)
```

Update the lm model by adding the x and z interaction term. This is the same as estimating the model $\text{lm}(y \sim x * z, \text{data} = \text{TDATA})$


```
summary(OLS.MOD3)
```

Summarize the regression model. Note that the interaction probably is not significant.

```
rm(TDATA, OLS.MOD, OLS.MOD2, OLS.MOD3)
```

Remove the objects that we just created (clean up workspace).

```
q()
```

Exit from the R session.

Exercise 3: Importing Data from EXCEL and SPSS

Large data objects will usually be read as values from external files rather than entered with the keyboard during an R session. There is a clear presumption by the designers of R that you will be able to modify your input files using other tools, such as file editors, or spreadsheets such as EXCEL to fit in with the requirements of R. Generally this is very simple.

If variables are to be held mainly in data frames, as they should be, an entire data frame can be read directly with the `read.table()` or `read.csv()` function.

To read an entire data frame directly, the external file will normally have this form.

- The first line of the file should have a name for each variable in the data frame.
- Each additional line of the file has values for each variable.

So the first few lines of a file to be read as a data frame might look as follows.

UNIT	PLATOON	COH01	COH02	COH03	COH04	COH05
1044B	1ST	4	5	5	5	5
1044B	1ST	3	NA	5	5	5
1044B	1ST	2	3	3	3	3
1044B	2ND	3	4	3	4	4
1044B	2ND	4	4	3	4	4
1044B	2ND	3	3	2	2	1
1044C	1ST	3	3	3	3	3
1044C	1ST	3	1	4	3	4
1044C	2ND	3	3	3	3	3
1044C	2ND	2	2	2	3	2
1044C	2ND	1	1	1	3	3

This format is, of course, the format one typically observes in SPSS or EXCEL. The simplest way to import data into R is to use EXCEL to process the data file into a comma delimited (*.csv) format. Note that most statistical packages (SAS, SPSS) can save data as an EXCEL file. Users who use SPSS and export data to EXCEL may encounter the error type value marker "#NULL!" for missing values. This must be changed to NA as under the second entry under COH02 in the example above to avoid problems in R. In addition, all blank spaces and any other missing value markers should be replaced with NA to facilitate data frame creation.

Once the comma delimited file is created using the “Save As” feature in EXCEL one can import it into R using either the `read.table()` or the `read.csv()` function.

Part 1:

1. Read in the EXCEL file “cohesion.xls” from the CD
2. Search for blanks and replace with NA
3. Save the file in the “c:\temp” subdirectory as command delimited file “cohesion.csv” using the “save as” option in EXCEL.

4. Start R and read in the cohesion comma delimited file using the following command:

```
>cohesion<-read.table("c:\\temp\\cohesion.csv", header=T, sep=",")
```

Note that subdirectories are designated using double slashes. Also note that R is case sensitive.

Alternatively, one can use read.csv()

```
>cohesion<-read.csv("c:\\temp\\cohesion.csv",header=T)
```

5. Look at all of the elements of the cohesion data frame

```
>cohesion
```

6. conduct a summary of the variables in the data frame using the summary command

```
>summary(cohesion)
```

Reading in SPSS

SPSS files can be read in using the “foreign” library.

```
> library(foreign)
```

```
> search() #see what position foreign is attached
```

```
[1] ".GlobalEnv"          "package:foreign"
```

```
[3] "package:multilevel"  "package:lattice"
```

```
[5] "package:grid"        "package:nlme"
```

```
[7] "package:nls"         "package:ctest"
```

```
[9] "Autoloads"          "package:base"
```

```
> objects(2) #Programs in the foreign package
```

```
[1] "data.restore" "lookup.xport" "read.dta"      "read.epiinfo"
```

```
[5] "read.mtp"     "read.S"       "read.spss"    "read.ssd"
```

```
[9] "read.xport"  "SModeNames"  "write.dta"
```

```
> help(read.spss) #look at the documentation on read.spss
```

```
> cohesion2<-read.spss("g:\\cohesion.sav") #note double \\
```

```
> cohesion2 #Look at the cohesion data set – it is not a typical data.frame because it contains labels, etc.
```

```
> cohesion2<-data.frame(cohesion2) #convert to data.frame
```

```
> cohesion2
```

	UNIT	PLATOON	COH01	COH02	COH03	COH04	COH05
1	1044B	1ST	4	5	5	5	5
2	1044B	1ST	3	NA	5	5	5
3	1044B	1ST	2	3	3	3	3
4	1044B	2ND	3	4	3	4	4
5	1044B	2ND	4	4	3	4	4
6	1044B	2ND	3	3	2	2	1
7	1044C	1ST	3	3	3	3	3
8	1044C	1ST	3	1	4	3	4
9	1044C	2ND	3	3	3	3	3
10	1044C	2ND	2	2	2	3	2
11	1044C	2ND	1	1	1	3	3

On your own:

1. Convert the Klein2000.xls file into an R data.frame named “klein2000” by using the `read.table` command. The file is on the CD. This is the simulated data set created by Bliese and used in Klein, et al., (2000). We will use this data set for a number of examples.
2. Convert the klein2000.sav file into an R data.frame named kien2000 using the `read.spss` command.
3. Regress Job Satisfaction (JOBSAT) on cohesion (COHES), workload (WLOAD) and the COHES*WLOAD interaction. You will use the `lm` (linear model) command that you first encountered in the sample session. Recall that a formula is specified as: $DV \sim IV1 + IV2$ or $DV \sim IV1 * IV2$. The first form regresses the DV on the main effects, and the second form regresses the DV on the main effects and the interaction.

Exercise 4: Estimating Within-group agreement and reliability

Step-by-step instructions

In this exercise we use the multilevel library to estimate within-group agreement and reliability. The data used in this exercise are taken from Bliese, Halverson & Rothberg (2000). The examples are based upon the `bhr2000` data set from the multilevel package. Thus, the first step is to examine the `bhr2000` data set and make it available for analysis.

```
> help(bhr2000)
> data(bhr2000) #puts the data in the working environment
> names(bhr2000)
[1] "GRP"    "AF06"  "AF07"  "AP12"  "AP17"  "AP33"  "AP34"
"AS14"  "AS15"  "AS16"  "AS17"  "AS28"  "HRS"   "RELIG"
> nrow(bhr2000)
[1] 5400
```

The `names` function tells us that there are 14 variables. The first one, `GRP`, is the group identifier. The variables in columns 2 through 12 are individual responses on 11 items that make up a leadership scale. `HRS` represents individuals' reports of work hours, and `RELIG` represents individuals' reports of the degree to which religion is a useful coping mechanism. The `nrow` command indicates that there are 5400 observations. To find out how many groups there are we can use the `length` command in conjunction with the `unique` command

```
> length(unique(bhr2000$GRP))
[1] 99
```

There are several functions in the multilevel library that are useful for estimating agreement. These functions are `rwg`, `rwg.j`, `rwg.j.lindell`, and `rgr.agree`. The `rwg` function estimates the James, Demaree & Wolf (1984) r_{wg} for single item measures; the `rwg.j` function estimates the James et al. (1984) $r_{wg(j)}$; the `rwg.j.lindell` function estimates $r^*_{wg(j)}$ (Lindell, & Brandt, 1997; 1999), and the `rgr.agree` function performs a Random Group Resampling (RGR) agreement test (see Bliese, et al., 2000).

In addition to the agreement measures, there are two multilevel reliability measures, `ICC1` and `ICC2` than can be used on ANOVA models. As Bliese (2000) and others (e.g., Kozlowski & Hattrup, 1992; Tinsley & Weiss, 1975) have noted reliability measures such as the `ICC(1)` and `ICC(2)` are fundamentally different from agreement measures; nonetheless, they often provide complementary information to agreement measures, so in this section, we illustrate the use of each of these functions with the data frame `bhr2000`.

Agreement: r_{wg} and $r_{wg(j)}$

Both the `rwg` and `rwg.j` functions are based upon the formulations described in James et al. (1984). Both functions require the user to specify three pieces of information. The first piece of information is the variable of interest (X), the second is the grouping variable (`GRPID`), and third is the estimate of the expected random variance (`ranvar`). The default estimate of `ranvar` is 2, which is the expected random variance based upon the rectangular distribution for a 5-point item (i.e., σ_{EV}^2). See `help(rwg)`, James et al., (1984), or Bliese et al., (2000) for details on selecting appropriate `ranvar` values.

To use the `rwg` function to estimate agreement for the comfort from religion item (`RELIG` in the `bhr2000` data frame), one would issue the following commands.

```
> rwg(bhr2000$RELIG, bhr2000$GRP)
  GNAME      rwg gsize
1      1 0.11046172   59
2      2 0.26363636   45
3      3 0.21818983   83
...      .      .      .
```

This returns a data frame with three columns. The first column contains the group names (`GNAME`), the second column contains the 99 r_{wg} estimates – one for each group, and the third column contains the group size. To estimate the mean r_{wg} value issue the command:

```
> mean(rwg(bhr2000$RELIG, bhr2000$GRP) [, 2])
[1] 0.1864222
```

Other useful options might include sorting the values

```
> sort(rwg(bhr2000$RELIG, bhr2000$GRP) [, 2])
```

or summarizing the values to get minimum and maximum values

```
> summary(rwg(bhr2000$RELIG, bhr2000$GRP) [, 2])
```

or viewing the values with a histogram

```
> hist(rwg(bhr2000$RELIG, bhr2000$GRP) [, 2])
```

To estimate r_{wg} for work hours, we need to change the expected random variance because work hours was asked using an 11-point item, so EV based on the rectangular distribution (σ_{EV}^2) is 10.00 ($\sigma_{EV}^2 = (11^2 - 1) / 12$) – see the `rwg` help file for details). The command to calculate the average r_{wg} in this case is:

```
> mean(rwg(bhr2000$HRS, bhr2000$GRP, ranvar=10.00) [, 2])
[1] 0.7353417
```

The use of the `rwg.j` function is nearly identical to the use of the `rwg` function except that the first argument to `rwg.j` is a matrix instead of a vector. In the matrix each column represents one item in the multi-item scale, and each row represents an individual response. For

instance, columns 2-12 in `bhr2000` represent a leadership scale, so to estimate average $r_{wg(j)}$ values we would issue the following command:

```
> mean(rwg.j(bhr2000[,2:12],bhr2000$GRP)[,2])
[1] 0.8876115
```

Note that Lindell and colleagues (Lindell & Brandt, 1997, 1999; 2000; Lindell, Brandt & Whitney, 1999) have raised important concerns about the mathematical underpinnings of the $r_{wg(j)}$ formula. Specifically, they note that this formula is based upon the Spearman-Brown reliability estimator. Generalizability theory provides a basis to believe that reliability should increase as the number of measurements increase, so the Spearman-Brown formula is defensible for measures of reliability. There may be no theoretical grounds, however, to believe that generalizability theory applies to measures of agreement. That is, there may be no reason to believe that agreement should increase as the number of measurements increase (but see also Cohen, Doveh & Eick, 2001).

To address this potential concern with the $r_{wg(j)}$, Lindell and colleagues have proposed the $r^*_{wg(j)}$. The $r^*_{wg(j)}$ is calculated by substituting the average variance of the items in the scale into the numerator of r_{wg} formula in lieu of using the $r_{wg(j)}$ formula ($r_{wg} = 1 - \text{Observed Group Variance/Expected Random Variance}$). Note that Lindell and colleagues also recommend against truncating the Observed Group Variance value so that it matches the Expected Random Variance value in cases where the observed variance is larger than the expected variance. This results in a case where $r^*_{wg(j)}$ values can take on negative values. We can use the function `rwg.j.lindell` to estimate the $r^*_{wg(j)}$ values for leadership.

```
> mean(rwg.j.lindell(bhr2000[,2:12],bhr2000$GRP)[,2])
[1] 0.4288838
```

The average $r^*_{wg(j)}$ value of .43 is considerably lower than the average $r_{wg(j)}$ value of .89 listed earlier.

Agreement: Random Group Resampling

The final agreement related function in the multilevel library is `rgr.agree`. This function (a) uses Random Group Resampling to create pseudo groups and calculate pseudo group variances, (b) estimates actual group variances, and (c) performs tests of significance to determine whether actual group and pseudo group variances differ. To use `rgr.agree`, one must provide three variables. The first is a vector representing the variable upon which one wishes to estimate agreement. The second is group membership (`GRPID`). The third parameter is the number of pseudo groups that one wants to create.

The third parameter requires a little explanation because in many cases the number of pseudo groups returned in the output will *not* exactly match the third parameter. For instance, in our example, we will request 1000 pseudo groups, but the output will return only 990. This is because the `rgr.agree` algorithm creates pseudo groups that are identical in size characteristics to the actual groups. In so doing, however, the algorithm creates sets of pseudo groups in “chunks.” The size of each chunk is based upon the size of the number of actual groups. So, for instance, if there are 99 actual groups, then the total number of pseudo groups must be evenly divisible by 99. Nine-hundred-and-ninety is evenly divisible by 99, while 1000

is not. Rather than have the user determine what is evenly divisible by the number of groups, `rgr.agree` will do this automatically. Below is an example of using `rgr.agree` on the work hours variable.

```
> RGR.HRS<-rgr.agree(bhr2000$HRS,bhr2000$GRP,1000)
```

The first step is to create an RGR Agreement object named `RGR.HRS`. The object contains a number of components. In most cases, however, users will be interested in the estimated t-value indicating whether the within-group variances from the actual groups are smaller than the variances from the pseudo groups. A useful way to get this information is to use the `summary` command. This command is widely used to summarize most R objects, and when applied to the RGR agreement object it provides standard deviations, variance estimates, and an estimate of the t-value.

```
> summary(RGR.HRS)
  NRanGrps Av.RGR.Var SD.RGR.Var AV.Grp.Var  t-value
1         990   3.328113  0.7884424   2.646583 -8.60068
```

The first column points out that although we requested 1000 random groups, we got 990 (for reasons described previously). The second column provides the mean within-group variance from the pseudo groups. The third column provides the Standard Deviation of the pseudo group variances. The fourth column provides the mean of the within-group variances for the actual groups. Notice that the average within-group variance from the actual groups (2.64) is smaller than the average within-group variance from the pseudo groups (3.33). The fifth column provides the estimated t-value indicating whether the actual group variances are significantly smaller than the population of pseudo group variances given the estimated variability of the pseudo group variances. Note that in our example, the value of -8.60 suggests that, as a whole, the variances from the actual groups are significantly smaller than the variances from the pseudo groups. This suggests that group members agree about work hours. Recall that a t-value greater than or less than 1.96 signifies significance at $p < .05$, two-tailed.

Keep in mind that if one replicates this example and the examples that follow, one is likely to get slightly different results. This is because the `rgr.agree` function uses a random number generator to create pseudo groups; thus, the results are partially a product of the specific numbers used in the random number generator. While the exact numbers may differ, the conclusions drawn should be very similar.

The object created by `rgr.agree` also contains the pseudo group variances. In our example, for instance, the `RGR.HRS` object contains the 990 within-group variances created from randomly assigning individuals to pseudo groups. The random variances are labeled `RGRVARS`, and can be accessed via the “\$” operator. For instance `RGR.HRS$RGRVARS` will list the 990 variances.

One can use the `quantile` command on the pseudo group variances to estimate whether specific groups do or do not display agreement. For instance, if we want to know how small a variance must be so that it is in the lowest 10% of all pseudo group variances, then we issue the command:

```
> quantile(RGR.HRS$RGRVARS, c(.10))
```


10%

2.432390

This information tells us that we can be 90% sure (one-tailed test) that any group with a variance smaller than 2.43 is different than the pseudo groups. That is, we can be 90% sure that groups with variances lower than 2.43 have significant agreement about work hours. If we want to see which groups meet this criterion, we can use the `tapply` function in conjunction with the `sort` function. The `tapply` function partitions the first variable by the level of the second variable performs the specified function. Thus, `tapply(HRS, GRP, var)` partitions HRS into separate Groups (GRP), and calculates the variance for each group (var). Using `sort` in front of this command simply makes the output easier to read. For example consider the following output:

```
> sort(tapply(bhr2000$HRS,bhr2000$GRP,var))
      33      43      38      19      6      39      69      17
0.8242754 1.0697636 1.1295681 1.2783251 1.3166667 1.3620690 1.4566667 1.4630282
      20      99      98      44      4      53      61      63
1.5009740 1.5087719 1.5256410 1.5848739 1.6384615 1.6503623 1.6623656 1.7341430
      66      14      76      71      21      18      59      50
1.7354302 1.7367089 1.7466200 1.7597586 1.7808500 1.7916027 1.8112599 1.8666667
      48      60      83      8      22      2      75      11
1.8753968 1.9267300 1.9436796 1.9476190 1.9679144 2.0282828 2.1533101 2.1578947
      96      23      54      47      55      26      74      57
2.1835358 2.1864802 2.2091787 2.2165242 2.2518939 2.2579365 2.2747748 2.2808858
      45      97      64      35      32      41      1      24
2.2975687 2.3386525 2.3535762 2.3563495 2.3747899 2.4096154 2.4284044 2.4391678
      82      37      81      68      42      73      34      25
2.4429679 2.4493927 2.5014570 2.5369458 2.5796371 2.6046154 2.6476418 2.6500000
      93      62      92      12      40      88      5      29
2.6602168 2.7341080 2.7746106 2.7906404 2.7916084 2.8505650 2.8672087 2.8748616
      85      70      77      51      3      13      79      87
2.8974843 2.9938483 3.0084034 3.0333333 3.0764032 3.1643892 3.1996997 3.2664569
      7      95      78      84      46      27      36      15
3.2712418 3.2804878 3.3839038 3.3886048 3.4084211 3.4309008 3.4398064 3.4425287
      89      16      58      49      9      31      90      72
3.4444444 3.4461538 3.4949020 3.5323440 3.6258065 3.6798419 3.8352838 3.9285714
      91      80      86      10      94      28      30      56
3.9565960 3.9729730 3.9753195 4.0336134 4.0984900 4.0994152 4.6476190 4.7070707
      65      52      67
4.7537594 5.2252964 5.3168148
```

If we starting counting from group 33 (the group with the lowest variance of 0.82) we find 47 groups that have variances smaller than 2.43. That is, we find 47 groups that have smaller than expected variance.

It may also be interesting to see what a “large” variance is when defined in terms of pseudo group variances. Thus we could issue the command:

```
> quantile(RGR.HRS$RGRVARS, c(.95))
      95%
4.663694
```

and find out that a within-group variance of 4.66 is “large” by pseudo group standards. Given this criterion, we have four groups that meet or exceed this standard. In an applied setting, one might be very interested in examining this apparent lack of agreement in groups 56, 65, 52 and 67. That is, one might be interested in determining what drives certain groups to have very large differences in how individuals perceive work hours.

Note that `rgr.agree` only works on vectors. Consequently, to use `rgr.agree` with the leadership scale we would need to create a leadership scale score. We can do this using the `apply` function. The `apply` function takes three arguments. The first argument is a matrix or data frame with multiple columns. In our case, the first argument will be a data frame where the 11 columns are the individual items that make up the leadership scale. The second argument to `apply` is a "1" or a "2". A "1" indicates that an action will be performed on each row, and a "2" indicates that an action will be performed on each column. Since we want to estimate each individuals' average response to the 11 items, we will perform the action on each row (recall that rows correspond to subjects). The third argument to `apply` is the function to be performed. In this case, we want to estimate the average value of the 11 scale items, so the function is `mean`. We will create a leadership scale (LEAD) and put it in the `bhr2000` data frame, so the specific command we issue is:

```
>bhr2000$LEAD<-apply(bhr2000[,2:12],1,mean)
```

Now that we have created a leadership scale score, we can perform the RGR agreement analysis on the variable.

```
> summary(rgr.agree(bhr2000$LEAD,bhr2000$GRP,1000))
      NRanGrps Av.RGR.Var SD.RGR.Var AV.Grp.Var   t-value
      990    0.5954029  0.1318061  0.5156757 -6.018504
```

The results indicate that the variance in actual groups about leadership ratings is significantly smaller than the variance in pseudo group (i.e., individuals agree about leadership). For interesting cases examining situations where group members do not agree see Bliese & Halverson (1998a) and Bliese and Britt (2001).

Reliability: ICC(1) and ICC(2)

The multilevel package also contains the reliability functions, `ICC1` and `ICC2`. These two functions are applied to ANOVA models and are used to estimate ICC(1) and ICC(2) as

described by Bartko, (1976), James (1982), and Bliese (2000). To use these functions, one first performs a one-way analysis of variance on the variable of interest. For instance, to calculate a one-way analysis of variance of work hours, we issue the following command using the `aov` ANOVA function that is part of the R base package. Note that in using the `aov` function, we use the `as.factor` function on GRP. This is important because it tells `aov` that GRP (which is numeric in this data frame) is to be treated as a categorical variable; consequently, R creates N-1 dummy codes in the model matrix (the exact form of the effects coding can be controlled, but will not be discussed in detail here). In the present example, there are 99 groups, so the `as.factor` function results in the creation of 98 dummy coded categories (98 df). Interested readers who estimate the model without the `as.factor` option will see that GRP erroneously only accounts for 1 df if the `as.factor` command is omitted.

```
> data(bhr2000)
> hrs.mod<-aov(HRS~as.factor(GRP),data=bhr2000)
> summary(hrs.mod)
              Df  Sum Sq Mean Sq F value    Pr(>F)
as.factor(GRP)  98  3371.4    34.4  12.498 < 2.2e-16 ***
Residuals      5301 14591.4     2.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> ICC1(hrs.mod)
[1] 0.1741008
> ICC2(hrs.mod)
[1] 0.9199889
```

Bliese (2000) provides a thorough interpretation of these values, but briefly, the ICC(1) value of .17 indicates that 17% of the variance in individual perceptions of work hours can be “explained” by group membership. The ICC(2) value of .92 indicates that groups can be reliably differentiated in terms of average work hours.

On your own

Use the `klein2000` data set that you created earlier to:

1. Estimate ICC(1) and ICC(2) for
 - Physical Environment
 - Cohesion
 - Pay
2. Perform a RGR agreement analysis with 1000 pseudo groups and estimate the t-value for
 - Physical Environment
 - Cohesion

Pay

3. Plot out the histogram of random variances from the RGR object for Physical Environment (you can copy a graph to the clipboard as a metafile and paste it in Word)

```
>histogram(rgr.mod$RGRVARS, nint=25, col=3)
```

4. Pretend that Cohesion was measured on a 7-point scale and estimate the average r_{wg} value for the 50 groups.

Exercise 5: Data manipulation functions for multilevel models

Step-by-Step Instructions.

One of the key data manipulation tasks that must be accomplished prior to estimating several of the multilevel models (specifically contextual models and random coefficient models) is that group-level variables must be “assigned down” to the individual. Typically one begins the process of assigning down data with two separate data frames. One data frame contains individual-level data, and the other data frame contains group-level data. To create a data frame that can be used in multilevel analyses, one must combine the individual and group data. This is accomplished using the `merge` function.

For instance, let us consider the `cohesion` data that I introduced when I showed how to read data from external files. The `cohesion` data is included as a multilevel data set, so we can use it without having to use `read.csv` or `read.table`.

```
> data(package="multilevel")
Data sets in package `multilevel':
bhr2000          Bliese Halverson and Rothberg (2000)
                  agreement data
bh1996          Bliese and Halverson (1996) data
cohesion        Platoon Cohesion file
klein2000       Klein et al. (2000) simulation data
univbct         Univariate form data for growth modeling
                  examples
```

To use the `cohesion` data frame in the immediate environment, we can simply issue the following command if we have attached the multilevel package:

```
>data(cohesion)
>cohesion
      UNIT PLATOON COH01 COH02 COH03 COH04 COH05
1  1044B     1ST     4     5     5     5     5
2  1044B     1ST     3    NA     5     5     5
3  1044B     1ST     2     3     3     3     3
4  1044B     2ND     3     4     3     4     4
5  1044B     2ND     4     4     3     4     4
6  1044B     2ND     3     3     2     2     1
7  1044C     1ST     3     3     3     3     3
8  1044C     1ST     3     1     4     3     4
```

9	1044C	2ND	3	3	3	3	3
10	1044C	2ND	2	2	2	3	2
11	1044C	2ND	1	1	1	3	3

Now assume that we have another data frame with platoon sizes. We can create this data frame as follows:

```
> group.size<-data.frame(UNIT=c("1044B","1044B","1044C","1044C"),
  PLATOON=c("1ST","2ND","1ST","2ND"),PSIZE=c(3,3,2,3))
> group.size #look at the group.size data frame
  UNIT PLATOON PSIZE
1 1044B     1ST     3
2 1044B     2ND     3
3 1044C     1ST     2
4 1044C     2ND     3
```

To create a single file (`new.cohesion`) that contains both individual and platoon information, we use the `merge` command.

```
> new.cohesion<-merge(cohesion,group.size,
  by=c("UNIT","PLATOON"))
> new.cohesion
  UNIT PLATOON COH01 COH02 COH03 COH04 COH05 PSIZE
1 1044B     1ST     4     5     5     5     5     3
2 1044B     1ST     3    NA     5     5     5     3
3 1044B     1ST     2     3     3     3     3     3
4 1044B     2ND     3     4     3     4     4     3
5 1044B     2ND     4     4     3     4     4     3
6 1044B     2ND     3     3     2     2     1     3
7 1044C     1ST     3     3     3     3     3     2
8 1044C     1ST     3     1     4     3     4     2
9 1044C     2ND     3     3     3     3     3     3
10 1044C     2ND     2     2     2     3     2     3
11 1044C     2ND     1     1     1     3     3     3
```

Notice that every individual now has a value for `PSIZE` – a value that reflects the number of individuals in the platoon.

The aggregate function

In many cases in multilevel analyses, one will be interested in creating a group-level variable from individual responses. For example, one might be interested in calculating the group mean and reassigning it back to the individual. In these cases, the `aggregate` function in combination with the `merge` function is particularly useful. In our cohesion example, for instance, we might want to have platoon means on COH01 and COH02 reassigned back to the individual.

The first step in this process is to create a group-level file. Creating this file is where one uses the `aggregate` function. The `aggregate` function has three key arguments. First is a vector or matrix of variables that one wants aggregated. Second are the grouping variable(s) included as a list, and third is the function (`mean`, `var`, `length`, etc.) executed on the variables from each group. To calculate the means of COH01 and COH02 which are in columns 3 and 4 of the cohesion data frame we can issue the command:

```
>TEMP<-aggregate(cohesion[,3:4],
list(cohesion$UNIT,cohesion$PLATOON),mean)
> TEMP
  Group.1 Group.2   COH01   COH02
1  1044B   1ST 3.000000      NA
2  1044C   1ST 3.000000 2.000000
3  1044B   2ND 3.333333 3.666667
4  1044C   2ND 2.000000 2.000000
```

Notice that COH02 has an “NA” value for the mean. This is because there was a missing value at the individual-level. If we decide to base the group mean on the non-missing group values we can add the parameter `na.rm=T`, to designate that NA values should be removed prior to calculating the group mean.

```
> TEMP<-aggregate(cohesion[,3:4],
list(cohesion$UNIT,cohesion$PLATOON),mean,na.rm=T)
> TEMP
  Group.1 Group.2   COH01   COH02
1  1044B   1ST 3.000000 4.000000
2  1044C   1ST 3.000000 2.000000
3  1044B   2ND 3.333333 3.666667
4  1044C   2ND 2.000000 2.000000
```

To merge the `TEMP` data frame with the `new.cohesion` data frame, we must change the names of the group identifiers in the `TEMP` frame to match the group identifiers in the `new.cohesion` data frame. We also want to change the names of COH01 and COH02 to reflect the fact that they are group means. We will use “G.” to designate group mean.

```
> names(TEMP)<-c("UNIT","PLATOON","G.COHO1","G.COHO2")
```

Finally, we merge TEMP up with new.cohesion to get the complete data set.

```
> final.cohesion<-merge(new.cohesion,TEMP,
by=c("UNIT","PLATOON"))
> final.cohesion
```

	UNIT	PLATOON	COH01	COH02	COH03	COH04	COH05	PSIZE	G.COH01	G.COH02
1	1044B	1ST	4	5	5	5	5	3	3.000000	4.000000
2	1044B	1ST	3	NA	5	5	5	3	3.000000	4.000000
3	1044B	1ST	2	3	3	3	3	3	3.000000	4.000000
4	1044B	2ND	3	4	3	4	4	3	3.333333	3.666667
5	1044B	2ND	4	4	3	4	4	3	3.333333	3.666667
6	1044B	2ND	3	3	2	2	1	3	3.333333	3.666667
7	1044C	1ST	3	3	3	3	3	2	3.000000	2.000000
8	1044C	1ST	3	1	4	3	4	2	3.000000	2.000000
9	1044C	2ND	3	3	3	3	3	3	2.000000	2.000000
10	1044C	2ND	2	2	2	3	2	3	2.000000	2.000000
11	1044C	2ND	1	1	1	3	3	3	2.000000	2.000000

With the aggregate and merge functions, one has the tools necessary to manipulate data and prepare it for subsequent multilevel analyses (excluding growth modeling which I consider later). Note that I have illustrated a more complex situation where there are two levels of nesting (Unit and Platoon). In cases where there is only one grouping variable (for example, UNIT) the commands for aggregate and merge contain the name of a single grouping variable. For instance,

```
>TEMP<-aggregate(cohesion[,3:4],list(cohesion$UNIT),mean,na.rm=T)
```

On your Own

In the klein2000 data frame, we need to have group means assigned back to the individuals for a number of the variables. For simplicity sake, let us just go ahead and create group mean variables for all the predictors (everything except GPRID and JOBSAT).

Overview of steps:

1. Use aggregate to create a temporary file with group means. Hint – there is only one level of nesting (GRPID), but you will still need to present the second argument to aggregate as a list e.g., list(klein2000\$GRPID).
2. Use names to rename the variables in the temporary file. Add a “G.” in front of the variable to designate it as a group mean.
3. Use merge to create a new “klein2000” data frame. Your final product should look like this:

```
> names(klein2000)
```


Multilevel Exercises 25

[1] "GRPID" "JOBSAT" "COHES" "POSAFF" "PAY" "NEGLEAD"
[7] "WLOAD" "TASKSIG" "PHYSEN" "G.COHES" "G.POSAFF" "G.PAY"
[13] "G.NEGLEAD" "G.WLOAD" "G.TASKSIG" "G.PHYSEN"

Exercise 6: Contextual Models

Prior to the introduction of multilevel random coefficient models, OLS regression models were widely used to detect contextual effects. Firebaugh (1978) provides a good methodological discussion of these types of contextual models as does Kreft and DeLeeuw (1998) and James and Williams (2000).

The basic logic behind these models is that a group mean variable (the potential contextual effect) can explain unique variance over and above an individual variable of the same name. So, for instance, Bliese (2002) found that average group work hours explained unique variance in individual well-being, over-and-above individual reports of work hours. This occurs because there is no mathematical reason why the group-level relationship between means must be the same as the individual-level relationship. When the slope of the group-mean relationship differs from the slope of the individual-level relationship, then we say that a contextual effect is present.

To estimate contextual regression models in R, one uses the OLS regression function `lm` introduced earlier to simultaneously test the significance of the individual and group mean variable. If the group-mean variable is significant in this model it indicates that the individual-level and group-level slopes are significantly different, and one has evidence of a contextual effect (Firebaugh, 1978; Snijders & Bosker, 1999).

Step-by-Step Instructions

In this example we use the `bh1996` data frame to illustrate the estimation of a contextual model. The `bh1996` data frame has group mean variables included; however, we will pretend that it does not so we can illustrate the use of the `aggregate` and `merge` functions.

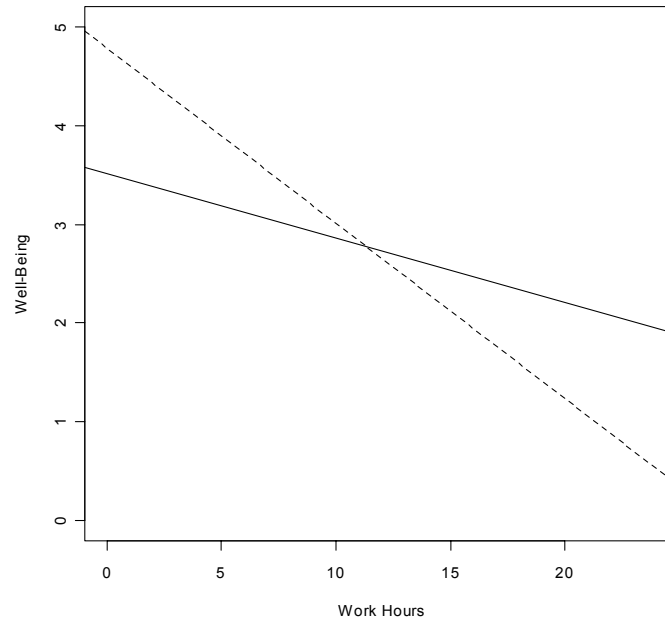
```
> data(bh1996)
> names(bh1996)
[1] "GRP"      "COHES"    "G.COHES"  "W.COHES"  "LEAD"     "G.LEAD"
[7] "W.LEAD"   "HRS"      "G.HRS"    "W.HRS"    "WBEING"   "G.WBEING"
[13] "W.WBEING"
> TDAT<-bh1996[,c(1,8,11)] #create a data frame with GRP, HRS and WBEING
> names(TDAT)
[1] "GRP"      "HRS"      "WBEING"
> TEMP<-aggregate(TDAT$HRS,list(TDAT$GRP),mean,na.rm=T)
> names(TEMP)
[1] "Group.1" "x"
> names(TEMP)<-c("GRP","G.HRS")
> TBH1996<-merge(TDAT,TEMP,by="GRP") #merge the group and individual data
> names(TBH1996)
```

```
[1] "GRP"      "HRS"      "WBEING"  "G.HRS"
> tmod<-lm(WBEING~HRS+G.HRS,data=TBH1996) #estimate the linear model
> summary(tmod,cor=F)
Call:
lm(formula = WBEING ~ HRS + G.HRS, data = TBH1996)
Residuals:
      Min       1Q   Median       3Q      Max
-2.87657 -0.57737  0.03755  0.64453  2.37267
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.783105    0.136395  35.068  <2e-16 ***
HRS          -0.046461    0.004927  -9.431  <2e-16 ***
G.HRS        -0.130836    0.013006 -10.060  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.8902 on 7379 degrees of freedom
Multiple R-Squared:  0.0398,    Adjusted R-squared:  0.03954
F-statistic: 152.9 on 2 and 7379 DF,  p-value:      0
```

Notice that G.HRS is significant. This provides evidence of significant contextual effects. If we want to examine the form of the relationship, we can plot out the regression slopes for the two models using the following commands:

```
> plot(TBH1996$HRS,TBH1996$WBEING,xlab="Work Hours",
ylab="Well-Being",type="n") #type = n omits the points which is important since we
have 7,382 observations
> abline(lm(WBEING~HRS,data=TBH1996)) # plots the individual-level slope
> abline(lm(WBEING~G.HRS,data=TBH1996),lty=2) #group-level slope
```

This produces the plot provided below. Notice that the group-mean slope (the dotted line) is considerably steeper than the individual slope (the solid line).



While contextual models are valuable, a major limitation with them is that they do not account for the fact that individuals are nested within groups. In essence, the models “pretend” that individual observations are independent instead of acknowledging that responses from individuals might be more similar than would be expected by chance. For instance, individual responses on well-being are somewhat influenced by group membership (as we will show later). This has the effect of biasing the standard errors, and making one a little too likely to detect contextual effects. Specifically, it is likely that the standard error of 0.013 associated with G.HRS is too small. This in turn makes the t-value too large. Better models are those that account for this non-independence – specifically random coefficient models. We will illustrate the estimation of these later. For more details on the effects of non-independence see Bliese (2002); Bliese, Hanges & Halverson (2000); Kenny & Judd, 1986; Snijders & Bosker, 1999.

On your Own

Using the klein2000 data frame, conduct a contextual Analysis regressing job satisfaction on:

1. Individual reports of cohesion (COHES) and group cohesion (G.COHES)
2. Individual reports of positive affect (POSAFF) and group positive affect (G.POSAFF)
3. Plot out the COHES and G.COHES relationship with Job Satisfaction.

Exercise 7: Covariance Theorem Decomposition (WABA)

OLS contextual models provide a way of determining whether or not regression slopes based on group means differ from regression slopes of individual-level variables. The covariance theorem provides a way of doing a similar thing for correlations nested in a two-level structure. Essentially, the covariance theorem allows one to break down a raw correlation into two separate components – the portion of the raw correlation attributable to within-group (individual) processes, and the portion of the correlation attributable to between-group (group-level) processes.

Robinson (1950) was one of the first researchers to propose the covariance theorem, but Dansereau and colleagues increased the visibility of the theorem by incorporating it into an analysis system they labeled WABA for Within-And-Between-Analyses (Dansereau, Alutto & Yammarino, 1984). WABA is actually two integrated procedures, WABA I and WABA II. WABA I uses a set of decision tools based on eta values to inform decisions about the individual or group-level nature of the data. However, since eta values are highly influenced by group size, and WABA I makes no group size adjustments, there is little value in WABA I unless one is working with dyads (see Bliese, 2000; Bliese & Halverson, 1998b). Perhaps a more useful way of drawing inferences from eta-values is to contrast eta-values from actual groups to eta-values from pseudo groups. We will illustrate this in a Random Group Resampling extension of the covariance theorem decomposition.

Step-by-Step Instruction

Dansereau et al.'s (1984) WABA II revolves around the estimation of the covariance theorem components, and the `waba` function in the multilevel library provides the covariance theorem components for the relationship between two variables. For example, to decompose the correlation between work hours and well-being into the between-group and within-group component we would issue the following command. Note that for comparative purposes we use the same data as we did in OLS contextual model example (section **Error! Reference source not found.**).

```
> waba (bh1996$HRS, bh1996$WBEING, bh1996$GRP)
$COVTHEOREM
      RawCorr      EtaBX      EtaBY      CorrB      EtaWX      EtaWY
CorrW
1 -0.1632064 0.3787881 0.2359287 -0.7121729 0.9254834 0.9717704
-0.1107031
$NOBS
[1] 7382
$NGROUPS
[1] 99
```

The `waba` function returns a list with three elements. The first element is the covariance theorem with all its components. The second element is the number of observations used in the estimate of the covariance theorem. The third element is the number of groups. The latter two elements should routinely be examined because the `waba` function, by default, performs listwise deletion of missing values.

This formula shows that the raw correlation of $-.163 = (\text{EtaBX} * \text{EtaBY} * \text{CorrB}) + (\text{EtaWX} * \text{EtaWY} * \text{CorrW})$ or $(.379 * .236 * -.712) + (.925 * .972 * -.111)$. Everything in the first set of parentheses represents the between-group component of the correlation, and everything in the second set of parentheses represents the within-group component of the correlation.

The group-mean correlation of $-.71$ definitely looks larger than the within-group correlation of $-.11$. Furthermore, since these two correlations are independent, we can contrast them using the `cordif` function. This function performs an r to z' transformation of the two correlations (see also the `rtoz` function) and then tests for differences between the two z' values using the formula provided in Cohen & Cohen (1983, p. 54). There are four arguments that must be provided to `cordif`. These are (1) the first correlation of interest, (2) the second correlation of interest, (3) the N on which the first correlation is based, and (4) the N on which the second correlation is based. In our example, we already have the two correlations of interest ($-.13$ and $-.66$); to get the N for the between-group correlation, we need to know the number of groups. We can get this N by determining how many unique elements there are in `GRP`.

```
> length(unique(bh1996$GRP))
[1] 99
```

The N for the within-group correlation is slightly more complicated. It is calculated as the total N minus the number of groups (see Dansereau, et al., 1984). In our example, we already know that the total N is 7,382 because we selected the first 7,382 responses. We also know that the number of groups is 99. Thus, the N for the within-group correlation is $7,382 - 99$ or 7,283. For illustrative purposes, however, we will use the `nrow` function to get the number of observations.

```
> nrow(bh1996) - 99
[1] 7283
```

With this information, we have all the necessary components for the `cordif` function.

```
> cordif(-.1107, -.7122, 7283, 99)
$"z value"
[1] 7.597172
```

The z -value is larger than 1.96, so we conclude that the two correlations are significantly different. That is, the between-group correlation is much stronger than the within-group correlation. This finding mirrors what we found in our contextual analysis. Note that the within-group correlation is based on X and Y deviation scores. These deviation scores are estimated by subtracting the group mean of X from X , and the group mean of Y from Y . In random coefficient modeling, these deviation scores are also called group-mean centered scores.

Random Group Resampling of Covariance Theorem (`rgr.waba`)

As noted above, it may be interesting to see how the eta-between, eta-within, between group and within-group correlations vary as a function of the group-level properties of the data. To do this, one can use the `rgr.waba` function. Essentially, the `rgr.waba` function allows one to answer questions such as "is my eta-between value for x larger than would be expected by chance?" The `rgr.waba` routine randomly assigns individuals into pseudo groups having the exact size characteristics as the actual groups, and then calculates the covariance theorem parameters. By repeatedly assigning individuals to pseudo groups and re-estimating the covariance theorem components, one can create sampling distributions of the covariance theorem components to see if actual group results differ from pseudo group results (see Bliese & Halverson, 2002). Below I illustrate the use of `rgr.waba`. Note that this is a very computationally intensive routine, so it may take some time to complete. The example below took five minutes with only 1000 of the 7382 observations. For comparative purposes, I begin by re-estimating the covariance theorem components using the first 1000 observations.

```
> TDAT<-bh1996[1:1000,c(1,8,11)]
> waba(TDAT$HRS,TDAT$WBEING,TDAT$GRP) #Model for first 1000 obs
      RawCorr      EtaBX      EtaBY      CorrB      EtaWX      EtaWY      CorrW
1 -0.1500598  0.4136304  0.192642  -0.6302504  0.9104449  0.9812691  -0.1117537

> RGR.WABA<-rgr.waba(TDAT$HRS,TDAT$WBEING,TDAT$GRP,1000)

> round(summary(RGR.WABA),dig=4)
      RawCorr      EtaBX      EtaBY      CorrB      EtaWX      EtaWY      CorrW
NRep 1000.0000 1000.0000 1000.0000 1000.0000 1000.0000 1.00e+03 1000.0000
Mean  -0.1501   0.1236   0.1241  -0.1409   0.9921  9.92e-01  -0.1501
SD     0.0000   0.0209   0.0217   0.2463   0.0026  2.80e-03   0.0040
```

The summary of the `rgr.waba` object produces a table giving the number of random repetitions, the means and the standard deviations from analysis. Notice that the raw correlation has a standard deviation of zero because it does not change. In contrast, the between-group correlation has the highest standard deviation (.25) indicating that it varied across pseudo group runs. It is apparent that all of covariance theorem components in the actual groups significantly vary from their counterparts in the pseudo group analysis. This is obvious because most actual group components are close to two standard deviations different from the pseudo group means. To test for significant differences in this resampling design, however, one can simply look at the sampling distribution of the random runs, and use the 2.5% and 97.5% sorted values to approximate 95% confidence intervals. Any values outside of this range would be considered significantly different from their pseudo group counterparts. To estimate the 95% confidence intervals we can use the `quantile` function.

```
> quantile(RGR.WABA, c(.025, .975))
      EtaBX      EtaBY      CorrB      EtaWX      EtaWY      CorrW
2.5%  0.08340649 0.08288485 -0.6048007 0.9861588 0.9857920 -0.1585368
97.5% 0.16580367 0.16797054  0.3613034 0.9965156 0.9965591 -0.1417005
```

Notice that all of the covariance theorem values based on the actual groups are outside of the 95% confidence interval estimates. That is, all of the actual group results are significantly different than would be expected by chance ($p < .05$). If we estimate the 99% confidence intervals we find that the between-group correlation is no longer outside of the 99% confidence interval, but the other values are.

```
> quantile(RGR.WABA, c(.005, .995))
      EtaBX      EtaBY      CorrB      EtaWX      EtaWY      CorrW
0.5%  0.07280037 0.07128845 -0.7216473 0.9843644 0.9831655 -0.1608020
99.5% 0.17614418 0.18271719  0.4825655 0.9973465 0.9974557 -0.1386436
```

Keep in mind in estimating the `rgr.waba` models that one's results are likely to differ slightly from those presented here because of the random generation process underlying random group sampling.

On your Own

Using `klein2000` conduct a covariance theorem decomposition analysis including tests for significant differences between the within-group and between-group correlations for:

1. The Job Satisfaction – Cohesion relationship
2. The Job Satisfaction – Negative Leadership relationship
3. Run an `rgr.waba` analysis and determine whether:

(a) the between-group correlation for job satisfaction and cohesion for actual groups differs from that found in pseudo groups.

(b) the between-group eta for job satisfaction in actual groups differs from the pseudo group value.

Exercise 8: MRCM – Examining the DV

Random coefficient modeling fundamentally involves estimating and predicting variance at different levels. Therefore, one typically begins a multilevel analysis by determining the levels at which significant variation exists. In the case of the two-level model (the only models that I will consider here), one generally assumes that there is significant variation in σ^2 – that is, one assumes that within-group variation is present. One does not necessarily assume, however, that there will be significant intercept variation (τ_{00}) or between-group slope variation (τ_{11}).

Generally, one begins by examining intercept variability (see Bryk & Raudenbush, 1992; Hofmann, 1997). If τ_{00} does not differ by more than chance levels, there may be little reason to use random coefficient modeling since simpler OLS modeling will suffice. Note that if slopes randomly vary even if intercepts do not, there may still be reason to estimate random coefficient models (see Snijders & Bosker, 1999).

Essentially in Step 1, one wants to explore the group-level properties of the outcome variable. Specifically, one is interested in determining three things about the outcome variable. First, what is the ICC of the outcome variable. That is, how much of the variance in the outcome can be explained by group membership. Second, one examines whether the group means of the outcome variable are reliable. By convention, we would like the group mean reliability to be around .70 because this indicates that groups can be reliably differentiated (see Bliese, 2000). Third, we want to know whether the variance of the intercept (τ_{00}) is significantly larger than zero.

These three aspects of the outcome variable are examined by estimating an unconditional means model. An unconditional means model does not contain any predictors, but includes a random intercept variance term for groups. This model essentially looks at how much variability there is in mean Y values (i.e., how much variability there is in the intercept) relative to the total variability. In the two stage HLM notation, the model is:

$$Y_{ij} = \beta_{0j} + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + u_{0j}$$

In combined form, the model is: $Y_{ij} = \gamma_{00} + u_{0j} + r_{ij}$. This model states that the dependent variable is a function of a common intercept γ_{00} , and two error terms: the between-group error term, u_{0j} , and the within-group error term, r_{ij} . In this model, one gets two estimates of variance; τ_{00} for how much each groups' intercept varies from the overall intercept (γ_{00}), and σ^2 for how much each individuals' score differs from the group mean. Bryk and Raudenbush (1992) note that this model is directly equivalent to a one-way random effects ANOVA – an ANOVA model where one predicts the dependent variable as a function of group membership.

The unconditional means model and all of the other random coefficient models that we will consider are estimated using the `lme` (for linear mixed effects) function in the NLME package. (see Pinheiro & Bates, 2000). There are two formulas that must be specified in any `lme` call: a fixed effects formula and a random effects formula.

In the unconditional means model, the fixed portion of the model is γ_{00} (an intercept term) and the random component is $u_{0j} + r_{ij}$. The random portion of the model states that intercepts will

be allowed to vary among groups. We begin by attaching both the nlme and multilevel packages and making the bh1996 data set in the multilevel package available for analysis.

```
> library(nlme)
Loading required package: nls
> library(multilevel)
> data(bh1996)
> Null.Model<-lme(WBEING~1, random=~1|GRP, data=bh1996)
```

In the model, the fixed formula is `WBEING~1`. This simply states that the only predictor of well-being is an intercept term. The random formula is `random=~1|GRP`. This specifies that the intercept can vary as a function of group membership. This is the simplest random formula that one will encounter, and in many situations a random intercept may be all that is needed.

Estimating ICC. The unconditional means model provides between-group and within-group variance estimates in the form of τ_{00} and σ^2 , respectively. As with the ANOVA model, it is often valuable to determine how much of the total variance is between-group variance. This can be accomplished by calculating the Intraclass Correlation Coefficient (ICC) using the formula: $ICC = \tau_{00}/(\tau_{00} + \sigma^2)$ (see, Bryk & Raudenbush, 1992; Kreft & De Leeuw, 1998). Bliese (2000) notes that the ICC is equivalent to Bartko's ICC(1) formula (Bartko, 1976) and to Shrout and Fleiss's ICC(1,1) formula (Shrout & Fleiss, 1979). To get the estimates of variance for an lme object, one uses the `VarCorr` function.

```
> VarCorr(Null.Model)
GRP = pdSymm(1)
              Variance  StdDev
(Intercept) 0.03580079 0.1892110
Residual     0.78949727 0.8885366
> 0.03580079/(0.03580079+0.78949727)
[1] 0.04337922
```

The estimate of τ_{00} (between-group variance) is 0.036, and the estimate of σ^2 (within-group variance) is 0.789. The ICC estimate ($\tau_{00}/(\tau_{00} + \sigma^2)$) is .04.

To verify that the ICC results from the random coefficient modeling are similar to those from an ANOVA model and the `ICC1` function (see section 0) one can perform an ANOVA analysis on the same data.

```
> tmod<-aov(WBEING~as.factor(GRP), data=bh1996)
> ICC1(tmod)
[1] 0.04336905
```

The ICC value from the random coefficient model and the ICC(1) from the ANOVA model are basically identical.

Estimating Group-Mean Reliability. When exploring the properties of the outcome variable, it can also be of interest to examine the reliability of the group mean. The reliability of group means often affects one's ability to detect emergent phenomena. In other words, a prerequisite for detecting emergent relationships at the aggregate level is to have reliable group means. By convention, one strives to have group mean reliability estimates around .70 (see Bliese, 2000). Group mean reliability estimates are a function of the ICC and group size (see Bliese, 2000; Bryk & Raudenbush, 1992). The `GmeanRel` function from the multilevel package calculates the ICC, the group size, and the group mean reliability for each group.

When we apply the `GmeanRel` function to our `Null.Model` based on the 99 groups in the `bh1996` data set, we are interested in two things. First, we are interested in the average reliability of the 99 groups. Second, we are interested in seeing whether or not there are specific groups that have particularly low reliability.

```
> Null.Model<-lme(WBEING~1, random=~1|GRP, data=bh1996)
> GREL.DAT<-GmeanRel(Null.Model)
> names(GREL.DAT)
[1] "ICC"      "Group"    "GrpSize"  "GmeanRel"
> GREL.DAT$ICC      #ICC estimate
[1] 0.04337922
> GREL.DAT$GmeanRel
 [1] 0.7704119 0.7407189 0.8131975 0.6557120 0.8222325
 [6] 0.5594125 0.5680426 0.6065741 0.6387944 0.7466758
[11] 0.6387944 0.6201282 0.7996183 0.8099782 0.7860071
[16] 0.6759486 0.8116016 0.7860071 0.6557120 0.7437319
[21] 0.8066460 0.6661367 0.7839102 0.8131975 0.5920169
[26] 0.7210397 0.8222325 0.6065741 0.7245244 0.6134699
[31] 0.6557120 0.6852003 0.5843267 0.8178269 0.8066460
[36] 0.7940029 0.6896308 0.7174657 0.6610045 0.8131975
[41] 0.7376341 0.6610045 0.8193195 0.7061723 0.7727775
[46] 0.8207878 0.6557120 0.7407189 0.7795906 0.5680426
[51] 0.6201282 0.6265610 0.5994277 0.7407189 0.7137989
[56] 0.7750949 0.8163095 0.7437319 0.7959093 0.8099782
[61] 0.7022044 0.8207878 0.6939384 0.7022044 0.7704119
[66] 0.7376341 0.8099782 0.6661367 0.5994277 0.8193195
[71] 0.7860071 0.4048309 0.6502517 0.7604355 0.7279232
[76] 0.7959093 0.6852003 0.7523651 0.7210397 0.6939384
[81] 0.8964926 0.7210397 0.9110974 0.8795291 0.8788673
[86] 0.9088937 0.8863580 0.7860071 0.8277854 0.9100090
```

```
[91] 0.8083266 0.8379118 0.8886532 0.8330020 0.8250530
[96] 0.6661367 0.7551150 0.4204716 0.5504306
> mean(GREL.DAT$GmeanRel) #Average group-mean reliability
[1] 0.7335212
```

Notice that the overall group-mean reliability is acceptable at .73, but that several groups have quite low reliability estimates. Specifically, group 71 and group 98 have reliability estimates below .50.

We can show that the group-mean reliability from the random coefficient model is equivalent to the ICC(2) from the ANOVA model by using the `bh1996` data to estimate the ICC(2) in an ANOVA framework (see section 0.).

```
> tmod<-aov(WBEING~as.factor(GRP),data=bh1996)
> ICC2(tmod)
[1] 0.7717129
```

In this case the ICC(2) estimate from the ANOVA model is slightly higher than the group-mean reliability estimate from the random coefficient model. This occurs because group sizes are unequal. If all the groups were the same size, then the two measures would be nearly identical.

With reference to ICC(2) values and group-mean reliability, note that there are alternate ways of estimating group-mean reliability. Snijders and Bosker (1999) show, for example, that one can estimate overall group-mean reliability by determining what percentage of the total group variance is made up by τ_{00} . Interested readers are directed to the help files for the `groupvar.ols` function in the multilevel library for details.

Finally, keep in mind that the estimates of within-group and between-group variance from the random coefficient model will be nearly identical to those from the ANOVA model as long as restricted maximum likelihood estimation (REML) is used in the random coefficient modeling (this is the default in the `lme` routine of the NLME package). If full maximum likelihood is used, the variance estimates may differ somewhat from the ANOVA estimates particularly in small sample situations. In our running example, the use of REML versus full maximum likelihood makes little difference. Interested readers may calculate ICC values from a `lme` model with maximum likelihood to verify this result.

```
> mod.ml<-lme(WBEING~1,random=~1|GRP,data=bh1996,method="ML")
> VarCorr(mod.ml)
GRP = pdLogChol(1)
              Variance StdDev
(Intercept) 0.03531699 0.1879282
Residual     0.78949525 0.8885354
```

Note that the maximum likelihood estimate of the ICC is 0.042 which is very similar to the 0.043 estimate with REML.

Determining whether τ_{00} is significant. Returning to our original analysis involving well-being from the `bh1996` data set, we might be interested in knowing whether the intercept variance (i.e., τ_{00}) estimate of 0.036 is significantly different from zero. To do this we compare -2 log likelihood values between (1) a model with a random intercept, and (2) a model without a random intercept.

A model without a random intercept is estimated using the `gls` function in the NLME library. The -2 Log likelihood values for a `lme` or `gls` object are obtained using the `logLik` function and multiplying this value by -2 . If the -2 log likelihood value for the model with random intercept is significantly larger than the model without the random intercept (based on a Chi-square distribution), then one concludes that the model with the random intercept is significantly “better” than the model without the random intercept. A significant Chi-square indicates that there is significant intercept variation. Note that we can directly contrast models using the `anova` function.

```
> Null.Model.2<-gls(WBEING~1,data=bh1996)
> logLik(Null.Model.2)*-2
`log Lik.` 19536.17 (df=2)
> logLik(Null.Model)*-2
`log Lik.` 19347.34 (df=3)
> 19536.17-19347.34
[1] 188.83
> anova(Null.Model,Null.Model.2)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
	Null.Model	1	3	19353.34	19374.06	-9673.669		
	Null.Model.2	2	2	19540.17	19553.98	-9768.084	1 vs 2	188.8303 <.0001

The -2 log likelihood value for the `gls` model without the random intercept is 19536.17. The -2 log likelihood value for the model with the random intercept is 19347.34. The difference of 188.8 is significant on a Chi-Squared distribution with one degree of freedom (one model estimated a random intercept, the other did not, and this results in the one df difference). These results suggest that there is significant intercept variation.

In summary, we would conclude that there is significant intercept variation in terms of general well-being scores across the 99 Army companies in our sample. We also estimate that 4% of the variation in individuals' well-being score is a function of the group to which he or she belongs.

On your Own

Conduct a null model analysis of the Job Satisfaction variable in the `klein2000` data frame. Include:

1. A random coefficient modeling estimate of the ICC
2. An examination of the average group mean reliability. Contrast this value with the ICC(2) value in an ANOVA model.

3. Determine whether τ_{00} is significantly larger than zero by contrasting models with and without random intercepts.

Exercise 9: MRCM – Adding Level-1 and Level-2 Main Effect Predictors

At this point in our example we have two sources of variation that we can attempt to explain in subsequent modeling – within-group variation (σ^2) and between-group intercept (i.e., mean) variation (τ_{00}). In many cases, these may be the only two sources of variation that we are interested in explaining so let us begin by building a model that predicts these two sources of variation.

To make things interesting, let us assume that individual well-being is related to individual reports of work hours. We expect that individuals who report high work hours will report low well-being. At the same time, let us also assume that average work hours in an Army Company is related to the average general well-being of the Company over-and-above the effect due to the individual-level work-hours and well-being relationship. Using Hofmann and Gavin's (1998) terminology, this means that we are testing an incremental model where the level-2 variable predicts unique variance after controlling for level-1 variables. This is also directly equivalent to the contextual model that we estimated earlier.

The first choice that we have to make in testing our contextual model is how we are going to center the variables. That is, we have to decide whether any transformations of the variables are necessary. Three options are available. The first option is to perform no transformation and use the raw variables. The second option is to grand-mean center the variables – subtract the variable's overall mean from each variable. The third option is to group-mean center the variables – subtract the variables' group mean from each variable. For details on centering see Hofmann and Gavin (1998). For simplicity, we will elect to use raw variables in the test of our incremental work hour model.

The form of the model using Bryk and Raudenbush's (1992) notation is:

$$\begin{aligned} WBEING_{ij} &= \beta_{0j} + \beta_{1j}(HRS_{ij}) + r_{ij} \\ \beta_{0j} &= \gamma_{00} + \gamma_{01}(G.HRS_j) + u_{0j} \\ \beta_{1j} &= \gamma_{10} \end{aligned}$$

Let us consider each row of the notation. The first row states that individual well-being is a function of the groups' intercept plus a component that reflects the linear effect of individual reports of work hours plus some random error. The second line states that each groups' intercept is a function of some common intercept (γ_{00}) plus a component that reflects the linear effect of average group work hours plus some random between-group error. The third line states that the slope between individual work hours and well-being is fixed—it is not allowed to randomly vary across groups.

When we combine the three rows into a single equation we get an equation that looks like a common regression equation with an extra error term (u_{0j}). This error term indicates that WBEING intercepts (i.e., means) can randomly differ across groups. The combined model is:

$$WBEING_{ij} = \gamma_{00} + \gamma_{10}(HRS_{ij}) + \gamma_{01}(G.HRS_j) + u_{0j} + r_{ij}$$

This model is specified in lme as:

```
> Model.1 <- lme(WBEING ~ HRS + G.HRS, random = ~1 | GRP, data = bh1996)
```

```

> summary(Model.1)
Linear mixed-effects model fit by REML
Data: bh1996
      AIC      BIC    logLik
19222.28 19256.81 -9606.14

Random effects:
Formula: ~1 | GRP
      (Intercept)  Residual
StdDev:   0.1163900 0.8832353

Fixed effects: WBEING ~ HRS + G.HRS
              Value Std.Error   DF   t-value p-value
(Intercept)  4.740829 0.21368746 7282 22.185808 <.0001
HRS          -0.046461 0.00488798 7282 -9.505056 <.0001
G.HRS        -0.126926 0.01940357   97 -6.541368 <.0001
Correlation:
      (Intr) HRS
HRS    0.000
G.HRS -0.965 -0.252

Standardized Within-Group Residuals:
      Min          Q1          Med          Q3          Max
-3.35320562 -0.65024982  0.03760797  0.71319835  2.70917777

Number of Observations: 7382
Number of Groups: 99

```

Notice that work hours are significantly negatively related to individual well-being. Furthermore after controlling the individual-level relationship, average work hours (G.HRS) is related to groups' well-being intercept.

At this point one can also estimate how much of the variance was explained by these two predictors. Because individual work hours were significantly related to well-being, we expect that it will have “explained” some of the within-group variance σ^2 . Similarly, since average work hours were related to the group well-being intercept we expect that it will have “explained” some of intercept variance, τ_{00} . Recall that in the null model, the variance estimate for the within-group residuals, σ^2 , was 0.789; and the variance estimate for the intercept, τ_{00} , was 0.036. The `VarCorr` function on the `Model.1` object reveals that each variance component has changed slightly.

```

> VarCorr(Model.1)
GRP = pdSymm(1)
      Variance StdDev
(Intercept) 0.01354663 0.1163900
Residual    0.78010466 0.8832353

```

Specifically, the variance estimates from the model with the two predictors are 0.780 and 0.014. That is, the variance of the within-group residuals decreased from 0.789 to 0.780 and the

variance of the between-group intercepts decreased from 0.036 to 0.014. We can calculate the percent of variance explained by using the following formula:

$$\text{Variance Explained} = 1 - (\text{Var with Predictor} / \text{Var without Predictor})$$

To follow through with our example, work hours explained $1 - (0.780/0.789)$ or 0.011 (1%) of the within-group variance in σ^2 , and group-mean work hours explained $1 - (0.014/0.036)$ or 0.611 (61%) of the between-group intercept variance τ_{00} . While the logic behind variance estimates appears pretty straightforward (at least in models without random slopes), the variance estimates should be treated with some degree of caution because they are partially dependent upon how one specifies the models. Interested readers are directed to Snijders and Bosker (1994; 1999) for an in-depth discussion of variance estimates.

On your own

1. Use the klein2000 data set to conduct a random coefficient model contextual analysis where you regress Job Satisfaction on individual reports of cohesion (COHES) and group cohesion (G.COHES). Compare the lme (random coefficient) and lm (OLS) results.
2. Simultaneously regress job satisfaction the variables below and estimate the amount of variance explained in σ^2 and τ_{00} .

Positive Affect (level-1)

Negative Leader Behaviors (level-1)

Pay (level-1)

Workload (level-1)

Cohesion (level-2)

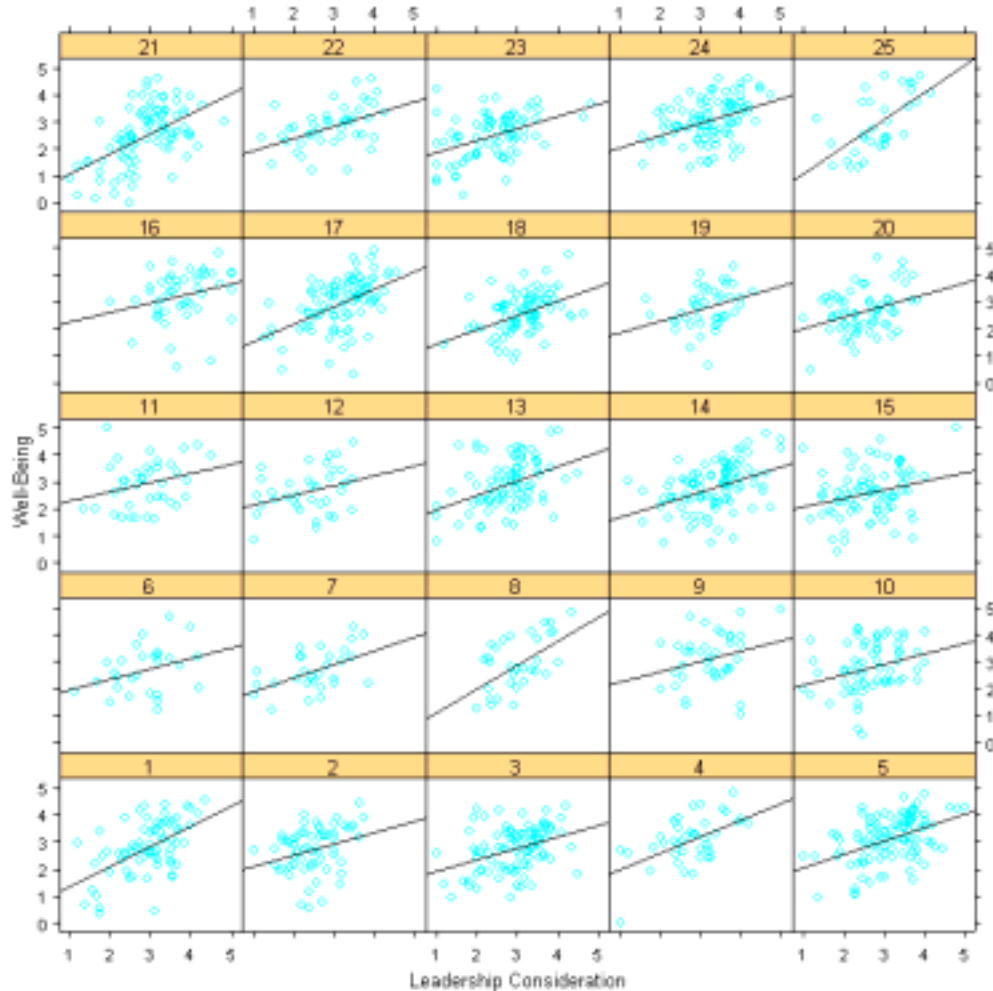
Exercise 10: Cross-level interactions

Step-by-Step

Let us continue our analysis by trying to explain the third source of variation, namely, variation in our slopes (τ_{11} , τ_{12} , etc.). To do this, let us examine another variable from the Bliese and Halverson (1996) data set. This variable represents Army Company members' ratings of leadership consideration (LEAD). Generally individual soldiers' ratings of leadership are related well-being. In this analysis, however, we will consider the possibility that the strength of the relationship between individual ratings of leadership consideration and well-being varies among groups.

We begin by examining slope variation among the first 25 groups. Visually we can do this using `xyplot` from the `lattice` package.

```
> library(lattice)
> trellis.device(device="windows",bg="white")
> xyplot(WBEING~LEAD|as.factor(GRP),
  data=bh1996[1:1582,],panel=function(x,y){
  panel.xyplot(x,y)
  panel.abline(lm(y~x,na.action=na.omit))},
  xlab="Leadership Consideration",
  ylab="Well-Being")
```



From the plot of the first 25 groups in the `bh1996` data set, it seems likely that there is some slope variation. The plot, however, does not tell us whether or not this variation is significant. Thus, the first thing to do is to determine whether the slope variation differs by more than chance levels.

Is slope variation significant? We begin our formal analysis of slope variability by adding leadership consideration to our model and testing whether or not there is significant variation in the leadership consideration and well-being slopes across groups. Once again, we must consider centering issues. When testing a cross-level interaction model as we are doing here, Hofmann & Gavin (1998) contend that the best centering choice for the level-1 variable is group-mean centering. Group-mean centering is the preferred choice because it provides the “purest” estimate of the within-group slope. That is, slope estimates based on raw variables and grand-mean centered variables can be partially influenced by between-group factors. In contrast, group-mean centered variables have any between-group effects removed.

Consequently, in our example we use the group-mean centered version of leadership consideration (`W.LEAD`) for within-group leadership. If this variable was not already included in the data set, we could create it by subtracting the group mean from the raw score

```
> W.LEAD<-bh1996$LEAD-bh1996$G.LEAD
```

Since we will ultimately use average work hours in a group as a level-2 variable in a cross-level interaction, we will grand-mean center group average work hours (we subtract the overall work hour mean from the group mean). To calculate the grand-mean centered variable (GG.HRS) and add it to the bh1996 data set we issue the following command:

```
> bh1996$GG.HRS<-bh1996$G.HRS-mean(bh1996$HRS)
```

Mathematically, grand-mean centering and raw variable models are identical, but grand-mean centering does facilitate model estimation and interpretation in random coefficient models just as it does in multiple regression models (see Hofmann & Gavin, 1997; Bryk & Raudenbush, 1992). The model that we test is:

$$\begin{aligned} WBEING_{ij} &= \beta_{0j} + \beta_{1j}(HRS_{ij}) + \beta_{2j}(W.LEAD_{ij}) + r_{ij} \\ \beta_{0j} &= \gamma_{00} + \gamma_{01}(GG.HRS_j) + u_{0j} \\ \beta_{1j} &= \gamma_{10} \\ \beta_{2j} &= \gamma_{20} + u_{2j} \end{aligned}$$

The last line of the model includes the error term u_{2j} . This term indicates that the leadership consideration and well-being slope is permitted to randomly vary across groups. The variance term associated with u_{2j} is τ_{12} . It is this variance term that interests us in the cross-level interaction hypothesis. Note that we have not permitted the slope between individual work hours and individual well-being to randomly vary across groups.

In combined form the model is: $WBEING_{ij} = \gamma_{00} + \gamma_{10}(HRS_{ij}) + \gamma_{20}(W.LEAD_{ij}) + \gamma_{01}(GG.HRS_j) + u_{0j} + u_{2j} * W.LEAD_{ij} + r_{ij}$. In R this model is designated as:

```
> Model.2<-lme(WBEING~HRS+W.LEAD+GG.HRS, random=~W.LEAD|GRP,
data=bh1996)
```

```
> summary(Model.2)
```

Linear mixed-effects model fit by REML

Data: bh1996

	AIC	BIC	logLik
	17853.76	17909.01	-8918.881

Random effects:

Formula: ~W.LEAD | GRP

Structure: General positive-definite

	StdDev	Corr
(Intercept)	0.1251704	(Intr)
W.LEAD	0.1033210	-0.194
Residual	0.8009007	

Fixed effects: WBEING ~ HRS + W.LEAD + GG.HRS

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.0977925	0.05306054	7281	58.38223	<.0001
HRS	-0.0279420	0.00447276	7281	-6.24716	<.0001
W.LEAD	0.5086165	0.01732498	7281	29.35740	<.0001
GG.HRS	-0.1411046	0.01929291	97	-7.31381	<.0001

```

Correlation:
      (Intr) HRS      W.LEAD
HRS      -0.952
W.LEAD   -0.103   0.078
GG.HRS   0.205  -0.233  -0.019

Standardized Within-Group Residuals:
      Min          Q1          Med          Q3          Max
-3.82055625 -0.66186907  0.04197464  0.70040326  3.99625224

Number of Observations: 7382
Number of Groups: 99

```

In line with our expectations, leadership consideration is significantly related to well-being. What we are interested in from this model, however, is whether τ_{12} , the slope between leadership consideration and well-being significantly varies across groups. To determine whether the slope is significant, we test the -2 log likelihood ratios between a model with and a model without a random slope for leadership consideration and well-being. We have already estimated a model with a random slope. To estimate a model without a random slope we use `update` on `Model.2` and change the random statement so that it only includes a random intercept.

```

> Model.2a<-update(Model.2, random=~1|GRP)
> anova(Model.2a,Model.2)

```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
	Model.2a	1	6 17868.44	17909.87	-8928.218			
	Model.2	2	8 17853.76	17909.01	-8918.881	1 vs 2	18.67456	1e-04

The difference of 18.67 is significant on two degrees of freedom. Note that there are two degrees of freedom because the model with the random slope also estimates a covariance term for the slope-intercept relationship. The log likelihood results indicate that model with the random effect for the leadership consideration and well-being slope is significantly better than the model without this random effect. This indicates significant slope variation.

Now that we know that we have significant variation in the leadership and well-being slope, we can attempt to see what group-level properties are related to this variation. In this example, we hypothesize that when groups are under a lot of strain from work requirements, the relationship between leadership consideration and well-being will be relatively strong. In contrast, when groups are under little strain, we expect a relatively weak relationship between leadership consideration and well-being. We expect these relationships because we believe that leadership is relatively unimportant in terms of individual well-being when groups are under little stress, but that the importance of leadership consideration increases when groups are under high stress. We are, in essence, proposing a contextual effect in an occupational stress model (see Bliese & Jex, in press).

A proposition such as the one that we presented in the previous paragraph represents a cross-level interaction. Specifically, it proposes that the slope between leadership consideration and well-being within groups varies as a function of a level-2 variable, namely group work demands. In random coefficient modeling we test this hypothesis by examining whether a level-2 variable

explains a significant amount of the level-1 slope variation among groups. In our example, we will specifically be testing whether average work hours in the group “explains” group-by-group variation in the relationship between leadership consideration and well-being. In Bryk and Raudenbush’s (1992) notation, the model that we are testing is:

$$\begin{aligned} WBEING_{ij} &= \beta_{0j} + \beta_{1j}(HRS_{ij}) + \beta_{2j}(W.LEAD_{ij}) + r_{ij} \\ \beta_{0j} &= \gamma_{00} + \gamma_{01}(GG.HRS_j) + u_{0j} \\ \beta_{1j} &= \gamma_{10} \\ \beta_{2j} &= \gamma_{20} + \gamma_{21}(GG.HRS_j) + u_{2j} \end{aligned}$$

In combined form the model is:

$$WBEING_{ij} = \gamma_{00} + \gamma_{10}(HRS_{ij}) + \gamma_{20}(W.LEAD_{ij}) + \gamma_{01}(GG.HRS_j) + \gamma_{21}(W.LEAD_{ij} * GG.HRS_j) + u_{0j} + u_{2j} * W.LEAD_{ij} + r_{ij}.$$

In lme we specify this model as:

```
> Final.Model<-lme(WBEING~HRS+W.LEAD+GG.HRS+W.LEAD:GG.HRS,
random=~W.LEAD|GRP, data=bh1996)
> round(summary(Final.Model)$tTable, dig=3)
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.099	0.053	7280	58.413	0.000
HRS	-0.028	0.004	7280	-6.271	0.000
W.LEAD	0.506	0.017	7280	29.947	0.000
GG.HRS	-0.145	0.019	97	-7.469	0.000
W.LEAD.GG.HRS	0.040	0.020	7280	2.038	0.042

The `tTable` results from the final model indicate there is a significant cross-level interaction (the last row). This result indicates that average work hours “explained” a significant portion of the variation in τ_{12} – the vertical cohesion and well-being slope.

We can examine the form of our interaction by predicting four points – high and low group work hours and high and low leadership consideration. These points can then be plotted in presentation software like Powerpoint. For convenience of interpretation, we will re-estimate a the model with Group Work Hours (G.HRS) in its raw form. Recall that this does not change the model. To make a prediction, we first create a small data set (TDAT) with high and low values for the interactive variables, and mean values for the non-interactive variables. We then use the `predict` function to get estimates of the outcome given the values of the variables.

```
> tmod<-lme(WBEING~HRS+W.LEAD+G.HRS+W.LEAD:G.HRS,
random=~W.LEAD|GRP, data=bh1996)
> TDAT<-data.frame(HRS=c(11.2987, 11.2987, 11.2987, 11.2987),
W.LEAD=c(1, 1, -1, -1),
G.HRS=c(12, 7, 12, 7),
```

```

GRP=c(1,1,1,1)
> predict(tmod,TDAT)
      1      1      1      1
3.256956 3.780753 2.033420 2.955928
attr(,"label")
[1] "Predicted values"

```

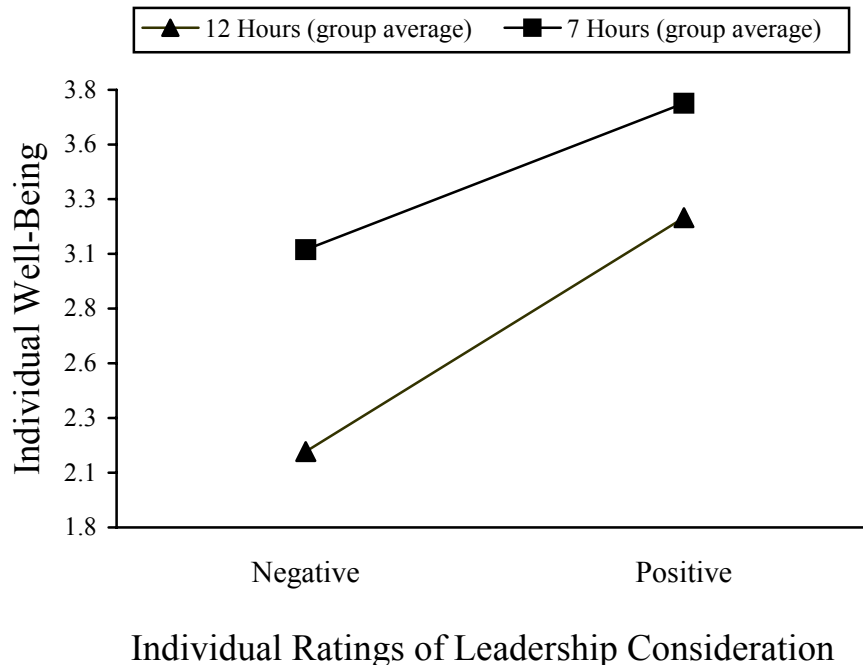
The predicted values in this case are specifically for GRP 1. Each group in the sample will have different predicted values because the slopes and intercepts randomly vary among groups. In many cases, one will not be specifically interested in the predicted values for specific groups, but interested in the patterns for the sample as a whole. If one is interested in estimating overall values, one can change the level of prediction to `level=0`.

```

> predict(tmod,TDAT,level=0)
[1] 3.214878 3.738712 2.146867 3.069459
attr(,"label")
[1] "Predicted values"

```

Notice that the values for the sample as a whole differ from those for GRP 1. When the values are plotted, the form of the interaction supports our proposition (note that this plot was generated in PowerPoint using the predicted values).



Soldiers' perceptions of leadership consideration are positively related to their well-being regardless of the number of hours that the group, on average, works; however, the relationship between leadership consideration and well-being is stronger (steeper slope) in groups with high

work hours than in groups with low work hours. Another way to think about the interaction is to note that well-being really drops (in relative terms) when one perceives that leadership is low in consideration and one is a member of a group with high work hours. This supports our proposition that considerate leadership is relatively more important in a high work demand context.

In this model one can also estimate how much of the variation in the slopes is “explained” by the group work hours. Recall that the estimate of the between group slope variance, τ_{12} , in the model with no slope predictors is 0.011. The estimate of the variance once average work hours has “explained” the slope variance is 0.009. Thus, average group work hours accounts for $1 - (0.009/0.011)$ or 18% of the slope variance. Once again, I emphasize that this is a rough estimate, and I direct readers to Snijders and Bosker (1994; 1999) for additional information on estimating effect sizes.

On-your-own

Use the klein2000 data set to conduct a full cross-level random coefficient model analysis.

1. As in the previous example, regress Job Satisfaction on:
 - Positive Affect (level-1)
 - Negative Leader Behaviors (level-1)
 - Pay (level-1)
 - Workload (level-1)
 - Cohesion (level-2)
2. Determine whether the workload-job satisfaction slope significantly varies across groups using the log likelihood method.
3. See whether there is a significant cross-level interaction between group task significance (level-2) and the workload-job satisfaction slope. This is basically a test of whether group task significance (a level-2 variable) interacts with individual workload (a level-1 variable).

Exercise 11: Growth Modeling Data Set Up

Step-by-Step

The first step in conducting a growth modeling analysis is to create a data set that is amenable to analysis. Often data is stored in a format where each row represents observations from one individual. For instance, an individual might provide three measures of job satisfaction in a longitudinal study, and the data might be arranged such that column 1 is the subject number; column 2 is job satisfaction at time 1; column 3 is job satisfaction at time 2, and column 4 is job satisfaction at time 3, etc.

The `univbct` data frame in the multilevel library allows us to illustrate this arrangement. This data set contains three measures taken six-months apart on three variables – job satisfaction, commitment, and readiness. It also contains some stable individual characteristics such as respondent gender, marital status and age at the initial data collection time. These latter variables are treated as level-2 predictors in subsequent modeling.

For convenience, the `univbct` data frame has already been converted into univariate or stacked form. Thus, it is ready to be analyzed in a growth model; however, for the purposes of illustration, we will select a subset of the entire `univbct` data frame and transform it back into multivariate form. With this subset we will illustrate how to convert a typical multivariate data frame into the format necessary for growth modeling.

```
> library(multilevel)
> data(univbct)
> names(univbct)
 [1] "BTN"      "COMPANY" "MARITAL" "GENDER"  "HOWLONG" "RANK"    "EDUCATE"
 [8] "AGE"      "JOBSAT1" "COMMIT1" "READY1"  "JOBSAT2" "COMMIT2" "READY2"
[15] "JOBSAT3" "COMMIT3" "READY3"  "TIME"    "JSAT"    "COMMIT"  "READY"
[22] "SUBNUM"
> nrow(univbct)
 [1] 1485
> length(unique(univbct$SUBNUM))
 [1] 495
```

These commands tell us that there are 1495 rows in the data set, and that there are data from 495 individuals. To create a multivariate data set out of the `univbct` data frame, we can select every third row of the `univbct` data frame. In this illustration we restrict our analyses to the three job satisfaction scores and to respondent age at the initial data collection period.

```
> GROWDAT<-univbct[3*(1:495),c(22,8,9,12,15)] #selects every third row
> GROWDAT[1:3,]
  SUBNUM AGE  JOBSAT1 JOBSAT2 JOBSAT3
3      1  20  1.666667      1      3
6      2  24  3.666667      4      4
```

```
9      3  24 4.000000      4      4
```

The data frame `GROWDAT` now contains data from 495 individuals. The first individual was 20 years old at the first data collection time. At time 1, the first individual's job satisfaction score was 1.67; at time 2 it was 1.0, and at time 3 it was 3.0.

Because of the nature of the `univbct` data frame in the multilevel package, we have added additional steps by converting a univariate data frame to a multivariate data frame; nonetheless, from a practical standpoint the important issue is that the `GROWDAT` data frame represents a typical multivariate data set containing repeated measures. Specifically, the `GROWDAT` data frame contains one row of information for each subject, and the repeated measures (job satisfaction) are represented by three different variables.

From a growth modeling perspective, the key problem with multivariate data frames like `GROWDAT` is that they do not contain a variable that indexes time. That is, we know time is an attribute of this data because we have three different measures of job satisfaction; however, analytically we have no way of explicitly modeling time. Thus, it is critical to create a new variable that explicitly indexes time. Thus requires transforming the data to univariate or a stacked format.

The `make.univ` function from the multilevel package provides a simple way to perform this transformation. Two arguments are provided in the `make.univ` function. The first argument required is the original data frame, and the second argument required is a data frame with just the columns containing the repeated measures. The latter argument must be time-sorted. That is column 1 must be time 1, column 2 must be time 2, etc. For instance to convert `GROWDAT` into univariate form we issue the following command:

```
> UNIV.GROW<-make.univ(GROWDAT,GROWDAT[,3:5])
> UNIV.GROW[1:9,]
```

	SUBNUM	AGE	JOBSAT1	JOBSAT2	JOBSAT3	TIME	MULTDV
X3	1	20	1.666667	1	3	0	1.666667
X31	1	20	1.666667	1	3	1	1.000000
X32	1	20	1.666667	1	3	2	3.000000
X6	2	24	3.666667	4	4	0	3.666667
X63	2	24	3.666667	4	4	1	4.000000
X64	2	24	3.666667	4	4	2	4.000000
X9	3	24	4.000000	4	4	0	4.000000
X95	3	24	4.000000	4	4	1	4.000000
X96	3	24	4.000000	4	4	2	4.000000

Note that each individual now has three rows of data indexed by the variable "TIME". Time ranges from 0 to 2. To facilitate model interpretation, the first time is coded as 0 instead of as 1. Doing so allows one to interpret the intercept as the level of job satisfaction at the initial data collection time. Second, notice that the `make.univ` function has created a variable called "MULTDV". This variable represents the multivariate dependent variable. The variable

“TIME” and the variable “MULTDV” are two of the primary variables used in growth modeling. Finally, notice that AGE, SUBNUM and the values for the three job satisfaction variables were repeated three times for each individual. By repeating the individual variables, the `make.univ` function has essentially created a data frame with level-2 variables in the proper format. For instance, subject age can now be used as a level-2 predictor in subsequent modeling.

On-Your-Own

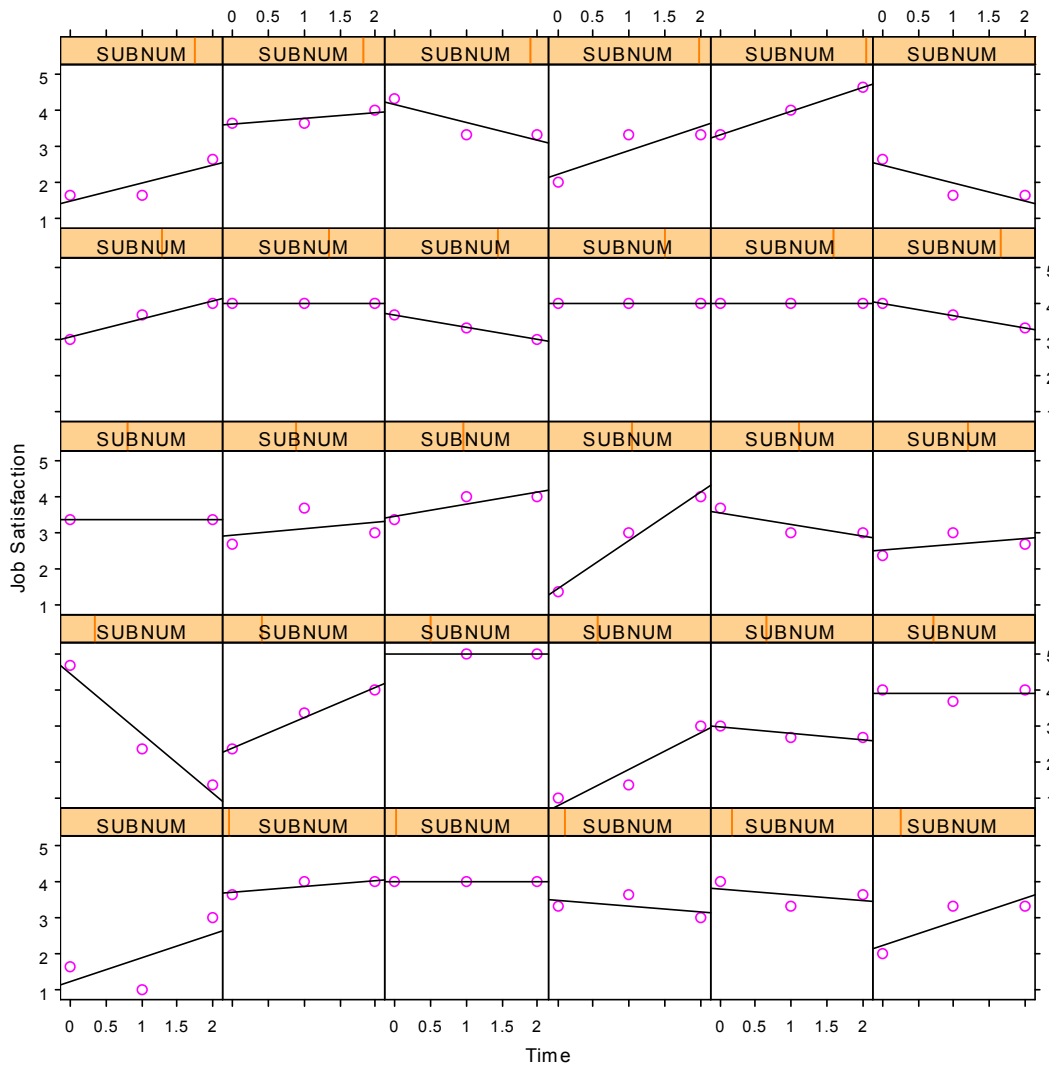
Read in the SPSS growth file and convert it to a univariate data frame ready for growth modeling. The data file contains a Subject number, respondent age, respondent marital status, respondent gender and three measures of commitment over time.

Exercise 12: Growth Modeling Example

Step-by-Step

With the data in univariate form, we can begin to visually examine whether or not we see patterns between time and the outcome. For instance, the commands below use the lattice package to produce a plot of the first 30 individuals:

```
>xyplot (MULTDV~TIME | SUBNUM,
data=UNIV.GROW[1:90, ],
panel=function(x,y){
panel.xyplot(x,y,col=6)
panel.abline(lm(y~x,
na.action=na.omit))}),
xlab="Time",
ylab="Job Satisfaction")
```



From this plot, it appears as though there is considerable variability both in overall levels of job satisfaction and in how job satisfaction changes over time. The goal in growth modeling is to determine whether or not we can find consistent patterns in the relationship between time and job satisfaction. Thus, we are now ready to illustrate growth modeling in a step-by-step approach. In this illustration, we follow the model comparison approach outlined by Bliese and Ployhart (2002) and in also discussed in Ployhart, Holtz and Bliese (in press).

As overview of the steps, the basic procedure is to start by examining the nature of the outcome. Much as we did in multilevel modeling, we are interested in estimating the ICC and determining whether the outcome (job satisfaction) randomly varies among individuals. Second, we are interested in examining the form of the relationship between time and the outcome. In essence, we want to know whether the outcome generally increases, decreases, or shows some other type of relationship with time. The plot of the first 30 individuals shows no clear pattern in how job satisfaction is changing over time, but the analysis might identify an overall trend among the 495 respondents. Third, we attempt to determine whether the relationship between time and the outcome is constant among individuals or whether it varies individual-by-individual. Fourth, we model in more complicated error structures such as autocorrelation, and finally we add level-2 predictors of intercept and slope variances.

Step 1: Examine the DV

The first step in growth modeling is to examine the properties of the dependent variable. As in multilevel modeling, one begins by estimating a null model and calculating the ICC.

```
> null.model<-lme (MULTDV~1, random=~1 | SUBNUM, data=UNIV.GROW,
na.action=na.omit)
> VarCorr(null.model)
SUBNUM = pdLogChol(1)
              Variance  StdDev
(Intercept)  0.4337729  0.6586144
Residual     0.4319055  0.6571952
> 0.4337729 / (0.4337729+0.4319055)
[1] 0.5010786
```

In our example using the UNIV.GROW data frame, the ICC associated with job satisfaction is .50. This indicates that 50% of the variance in any individual report of job satisfaction can be explained by the properties of the individual who provided the rating. Another way to think about this is to note that individuals tend to remain fairly constant in ratings over time, and that there are differences among individuals. This observation is reflected in the graph of the first 30 respondents.

Step 2: Model Time

Step two involves modeling the fixed relationship between time and the dependent variable. In almost all cases, one will begin by modeling a linear relationship and progressively add more complicated structures such as quadratic, cubic, etc. To test whether there is a linear relationship

between time and job satisfaction, we regress job satisfaction on time in a model with a random intercept.

```
> model.2<-lme(MULTDV~TIME, random=~1|SUBNUM, data=UNIV.GROW,
na.action=na.omit)
> summary(model.2)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.21886617	0.04075699	903	78.977040	0.00000000
TIME	0.05176461	0.02168024	903	2.387640	0.01716169

An examination of the fixed effects indicates that there is a significant linear relationship between time and job satisfaction such that job satisfaction increases by .05 each time period. Note that since the first time period was coded as 0, the intercept value in this model represents the average level of job satisfaction at the first time period. Specifically, at the first time period the average job satisfaction was 3.22.

More complicated time functions can be included in one of two ways – either through raising the time variable to various powers, or by converting time into power polynomials. Below both techniques are illustrated.

```
> model.2b<-lme(MULTDV~TIME+I (TIME^2) , random=~1|SUBNUM,
data=UNIV.GROW, na.action=na.omit)
> summary(model.2b)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.23308157	0.04262697	902	75.8459120	0.00000000
TIME	-0.03373846	0.07816572	902	-0.4316273	0.6661154
I (TIME^2)	0.04276425	0.03756137	902	1.1385167	0.2552071

```
> model.2c<-lme(MULTDV~poly (TIME, 2) , random=~1|SUBNUM,
data=UNIV.GROW, na.action=na.omit)
> summary(model.2c)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.2704416	0.0346156	902	94.478836	0.00000000
poly (TIME, 2) 1	1.5778835	0.6613714	902	2.385775	0.01724863
poly (TIME, 2) 2	0.7530736	0.6614515	902	1.138517	0.25520707

Both models clearly show that there is no significant quadratic trend. Note that a key advantage of the power polynomials is that the linear and quadratic effects are orthogonal. Thus, in the second model the linear effect of time is still significant even with the quadratic effect in the model. In either case, however, we conclude that time only has a linear relationship with job satisfaction.

Step 3: Model Slope Variability

A potential limitation with model 2 is that it assumes that the relationship between time and job satisfaction is constant for all individuals. Specifically, it assumes that each individual increases job satisfaction by .05 points at each time period. An alternative model that needs to be tested is one that allows slopes to randomly vary. Given the degree of variability in the graph of the first 30 respondents, a random slope model seems quite plausible with the current data. A random slope model is tested by adding the linear effect for time as a random effect. In the running example, we can simply update model.2 by adding a different random effect component and contrast model 2 and model 3.

```
> model.3<-update(model.2, random=~TIME | SUBNUM)
> anova(model.2,model.3)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
	model.2	1	4	3461.234	3482.194	-1726.617		
	model.3	2	6	3434.132	3465.571	-1711.066	1 vs 2	31.10262 <.0001

The results clearly indicate that a model that allows the slope between time and job satisfaction to randomly vary fits the data better than a model that fixes the slope to a constant value for all individuals.

In cases where higher-level trends were also significant, one would also be interested in determining whether allowing the slopes of the higher-level variables to randomly vary also improved model fit. For instance, one might find that a quadratic relationship varied in strength among individuals.

Step 4: Modeling Error Structures

The fourth step in developing the level-1 model involves assessing the error structure of the model. It is important to carefully scrutinize the level-1 error structure because significance tests may be dramatically affected if error structures are not properly specified. The goal of step 4 is to determine whether one's model fit improves by incorporating (a) an autoregressive structure with serial correlations and (b) heterogeneity in the error structures.

Tests for autoregressive structure (autocorrelation) are conducted by including the `correlation` option in `lme`. For instance, we can update model.3 and include lag 1 autocorrelation as follows:

```
> model.4a<-update(model.3, correlation=corAR1())
> anova(model.3,model.4a)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
	model.3	1	6	3434.132	3465.571	-1711.066		
	model.4a	2	7	3429.771	3466.451	-1707.886	1 vs 2	6.360465 0.0117

A model that allows for autocorrelation fits the data better than does a model that assumes no autocorrelation. A summary of model 4a reveals that the autocorrelation estimate is .367 (see the Phi coefficient).

```
> summary(model.4a)
Linear mixed-effects model fit by REML
```

```
Data: UNIV.GROW
      AIC      BIC    logLik
3429.771 3466.451 -1707.886
.....
Correlation Structure: AR(1)
Formula: ~1 | SUBNUM
Parameter estimate(s):
      Phi
0.3676831
```

Finally, we can examine the degree to which the variance of the responses changes over time. A simple preliminary test of variance homogeneity can be conducted by examining the variance of job satisfaction at each time point using the `tapply` command.

```
> tapply(UNIV.GROW$MULTDV, UNIV.GROW$TIME, var, na.rm=T)
      0      1      2
0.9681912 0.8831397 0.7313358
```

The analysis suggests the variance of job satisfaction is decreasing over time. To model decreasing variance one can use the `varExp` option. In cases where variance increases can use the `varFixed` option (see Pinheiro & Bates, 2000 for details).

```
> model.4b<-update(model.4a, weights=varExp(form=~TIME))
> anova(model.4a, model.4b)
      Model df      AIC      BIC    logLik    Test  L.Ratio p-value
model.4a    1   7 3429.771 3466.451 -1707.886
model.4b    2   8 3428.390 3470.309 -1706.195 1 vs 2 3.381686 0.0659
```

The model that includes both autocorrelation and allows for decreases in variance fits the data marginally better (using a liberal p-value) than does the model that only includes autocorrelation. In subsequent analyses, however, `model.4b` ran into convergence problems. Consequently, we elect to use `model.4a` as our final level-1 model.

With the completion of step 4, we have exhaustively examined the form of the level-1 relationship between time and job satisfaction. This analysis has revealed that (a) individuals randomly vary in terms of their mean levels of job satisfaction, (b) there is a linear, but not quadratic, relationship between time and job satisfaction, (c) the strength of the linear relationships randomly varies among individuals, and (d) there is a fair amount of autocorrelation in the data. At this point, we are ready to add level-2 variables to try and explain the random variation in intercepts (i.e., mean job satisfaction) and in the time-job satisfaction slope.

Step 5: Predicting Intercept Variation

Step 5 in growth modeling is to examine factors that can potentially explain intercept variation. Specifically, in our case we are interested in examining factors that explain why some

individuals have high job satisfaction while other individuals have low job satisfaction. In this example, we explore the idea that age is related to intercept variation.

To model this relationship, the individual-level characteristic, age, is used as a predictor of the job satisfaction intercept. The model that we will test is represented below using the Bryk and Raudenbush (1992) notation.

$$Y_{ij} = \pi_{0j} + \pi_{1j}(\text{Time}_{ij}) + r_{ij}$$

$$\pi_{0j} = \beta_{00} + \beta_{01}(\text{Age}_j) + u_{0j}$$

$$\pi_{1j} = \beta_{10} + u_{1j}$$

This equation states that respondent j 's initial job satisfaction (π_{0j}) can be modeled as a function of two things. One is the mean level of job satisfaction (β_{00}) for all respondents. The second is a coefficient associated with the individual's age (β_{01}). Note that the error term for the intercept (u_{0j}) now represents the difference between an individual's intercept and the overall intercept after accounting for age. In lme the model is specified as:

```
> model.5<-lme(MULTDV~TIME+AGE, random=~TIME|SUBNUM,
  correlation=corAR1(), na.action=na.omit, data=UNIV.GROW)

> round(summary(model.5)$tTable, dig=3)
              Value Std.Error   DF t-value p-value
(Intercept)  2.347      0.146  897  16.086  0.000
TIME          0.053      0.024  897   2.205  0.028
AGE           0.034      0.005  486   6.241  0.000
```

Model 5 differs only from Model 4a in that Model 5 includes a new fixed effect, AGE. Notice that age is positively related to initial levels of job satisfaction. Also notice that there are fewer degrees of freedom for age than for time since age is an individual (level-2) attribute.

In interpreting the coefficients from model 5, we conclude that in cases where age is 0 and where time is 0, the expected level of job satisfaction is 2.347. In some ways, this interpretation is strange because age will never actually be 0 in this population. Consequently, it may be useful to reparameterize age by grand-mean centering the variable (see Singer, 1998). Grand mean centering involves subtracting the overall mean from each observation. A model using a grand-mean centered version of age (AGE2) is presented below.

```
> UNIV.GROW$AGE2<-UNIV.GROW$AGE-mean(UNIV.GROW$AGE, na.rm=T)
> model.5b<-lme(MULTDV~TIME+AGE2, random=~TIME|SUBNUM,
  correlation=corAR1(), na.action=na.omit, data=UNIV.GROW)
> round(summary(model.5b)$tTable, dig=3)
              Value Std.Error   DF t-value p-value
(Intercept)  3.216      0.043  897  74.564  0.000
TIME          0.053      0.024  897   2.205  0.028
AGE2          0.034      0.005  486   6.241  0.000
```

With age grand-mean centered, the intercept estimate of 3.216 now represents the initial job satisfaction value for a respondent of average age (25.7 years old). Notice that the t-values for time and age did not change between this and the previous model. While we will continue our analyses using the untransformed age variable, readers should keep in mind that grand-mean centering is often valuable in terms of both enhancing the interpretability of models.

Step 6: Predicting Slope Variation

The final aspect of growth modeling involves attempting to determine attributes of individual respondents that are related to slope variability. In this section, we attempt to determine whether respondent age can explain some of the variation in the time-job satisfaction slope. The model that we test is presented below:

$$Y_{ij} = \pi_{0j} + \pi_{1j}(\text{Time}_{ij}) + r_{ij}$$

$$\pi_{0j} = \beta_{00} + \beta_{01}(\text{Age}_j) + u_{0j}$$

$$\pi_{1j} = \beta_{10} + \beta_{11}(\text{Age}_j) + u_{1j}$$

This model is similar to the model specified in step 5 except that we now test the assumption that the slope between time and job satisfaction for an individual (π_{1j}) is a function of an overall slope (β_{10}), individual age (β_{11}), and an error term (u_{1j}). In `lme`, the model is specified as:

```
> model.6<-lme(MULTDV~TIME*AGE, random=~TIME | SUBNUM,
correlation=corAR1(), na.action=na.omit, data=UNIV.GROW)
```

Note that the only difference between model 5 and model 6 is that we include an interaction term for time and age. A summary of model 6 reveals that there is a significant interaction between time and age.

```
> round(summary(model.6)$tTable, dig=3)
              Value Std.Error   DF t-value p-value
(Intercept)  2.098      0.186  896  11.264  0.000
TIME          0.271      0.104  896   2.608  0.009
AGE           0.043      0.007  486   6.180  0.000
TIME:AGE     -0.008      0.004  896  -2.153  0.032
```

In the multilevel modeling section we illustrated how to use the `predict` command to generate points that could be used to plot out interactions. An alternative approach is to use the overall coefficients from the final model in conjunction with high and low values for the predictors to generate points for plots. Notice in the example that follows that the first row in the `TDAT` data frame is a row of 1s for the intercept, while the other rows contain high and low values for time, age and the time*age interaction.

```
> TDAT<-data.frame(COEF5=(summary(model.6)$tTable)[,1],
CASE1=c(1,0,21,0), CASE1=c(1,0,31,0),
CASE3=c(1,2,21,42), CASE4=c(1,2,31,62))
```

```
> TDAT
```

```

                COEFS CASE1 CASE1 CASE3 CASE4
(Intercept)  2.097720117      1      1      1      1
TIME          0.271036716      0      0      2      2
AGE           0.043449071     21     31     21     31
TIME:AGE     -0.008432157      0      0     42     62

```

```
> sum(TDAT[,1]*TDAT[,2])
```

```
[1] 3.010151
```

```
> sum(TDAT[,1]*TDAT[,3])
```

```
[1] 3.444641
```

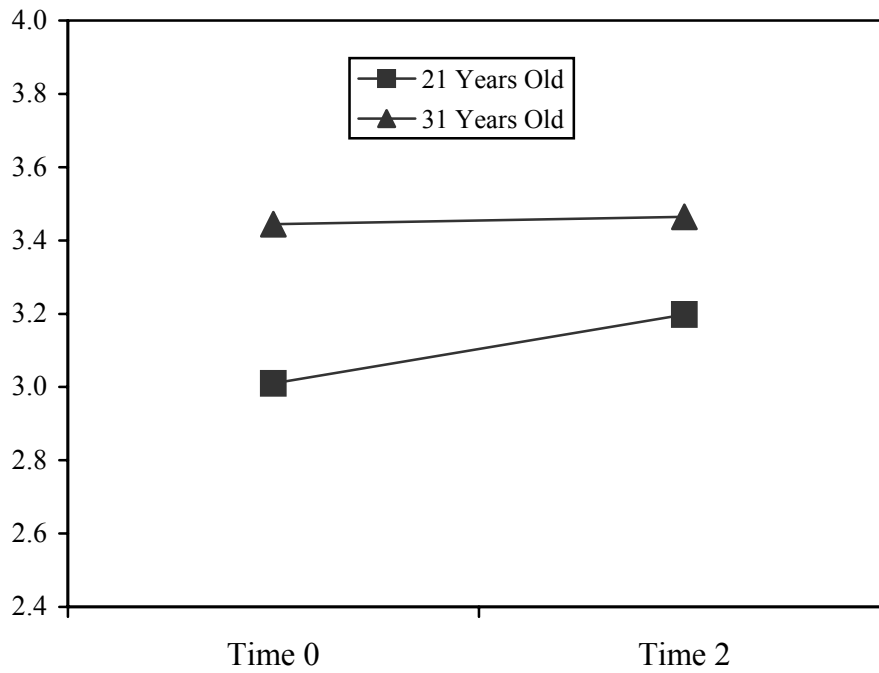
```
> sum(TDAT[,1]*TDAT[,4])
```

```
[1] 3.198073
```

```
> sum(TDAT[,1]*TDAT[,5])
```

```
[1] 3.463921
```

These points are used in the plot of the interaction. Notice that older individuals reported higher job satisfaction initially, and tended to show a very slight increase over time. In contrast, younger respondents tended to report lower initial job satisfaction, but showed a more pronounced increase in job satisfaction over time.



On-Your-Own

Use the univariate form of the growth data you imported in exercise 11 to perform a growth-modeling analysis of commitment.

1. Start with a plot of time and commitment for the first 30 respondents
2. Follow each step of the process as outlined above
3. See if the commitment intercept and the time-commitment slope vary as a function of age.

Hints: both the linear and quadratic effects are significant, so you need to test whether the linear and quadratic effects also randomly vary across individuals. To do this, contrast three models, (a) random intercept, (b) random linear slope, (c) random linear and quadratic slope.

[random= \sim poly(TIME,1)|SUBNUM] vs [random= \sim poly(TIME,2)|SUBNUM]