

Taking Regression Further: Finding the Best Fit in Multiple Regression

Chapter 7 acted as if there were only one way to conduct a regression analysis. The truth is that statisticians who understand how regression works know that the point is not in simply performing a regression, but in finding the regression model that best fits your data. Crawley says that “All models are wrong” (2007, p. 339). What Crawley means by this is that, whatever model you use to fit your data, there will be error. Thus all models are wrong, but some models are better than others. In general, a simpler model is better than a more complicated one, if they have the same explanatory value. A model with less error will be better than one with more error.

In order to find the best model for your data, Crawley (2007) recommends beginning with the maximal model and then simplifying your model where possible. What is the maximal model? It is the one that includes all of the **main effects** (the effect of the variable by itself) plus all of the **interactions** between the terms. A main effect is the effect due to a variable by itself. Going back to the TOEFL example considered at the beginning of this chapter with the two explanatory variables of MLAT and hours of study, a model with only main effects looks like:

TOEFL score = MLAT score + hours of study

In order to make this example more interesting, let's add a third variable of gender. Gender is of course a categorical variable with only two possible responses. It is perfectly acceptable to add a categorical variable to a regression but we will have to interpret the output for categorical variables a little differently than we did for continuous variables. So adding gender to the

equation, this model has only main effects for the three explanatory variables:

$$\text{TOEFL score} = \text{MLAT score} + \text{hours of study} + \text{gender}$$

However, it may be the case that there is an interaction between gender and hours of study. It may be the case that females study more than males, or vice versa. In other words, the way that gender and hours of study vary is linked. If there is an interaction, then we will want to include that in our equation as well, in order to most effectively model what is happening. An interaction can be shown by putting a colon between the two variables, as in the following model with an interaction between gender and hours of study shown:

$$\text{TOEFL score} = \text{MLAT score} + \text{hours of study} + \text{gender} + \text{gender:hours of study}$$

As stated above, the way a professional statistician would approach a question of regression is to consider the model that has the best fit. The statistician would start with the maximal model, which would include all of the possible interactions. In a model with three explanatory variables, there will be three two-way interactions and one three-way interaction:

Two-way interactions:

MLAT score:hours of study

MLAT score:gender

Gender:hours of study

Three-way interaction:

MLAT score:hours of study:gender

Putting all of the main effects and interactions into one regression model, then, would look like this:

TOEFL score = MLAT score + hours of study + gender + MLAT score:hours of study + MLAT score:gender + gender:hours of study + MLAT score:hours of study:gender

You can see that, if you have more than three explanatory variables, this model can very quickly get very complicated! In addition, you might want to add some quadratic terms, which are variables that are squared or raised to the cubic power. Why would you want to do this? If there is some curvature in your data then squared powers of a variable can help account for that curvature.

Finding the Best Fit with SPSS

The problem with trying to conduct this type of regression analysis in SPSS is that it does not provide an easy way for us to evaluate model fit. In the R program different models can easily be compared using an ANOVA which tests the difference in something called “residual deviance” between the two models. If there is a statistical difference between the residual deviance of two models, you pick the model with the smaller deviance as the simpler model. In R there are also commands that help automate the process of testing each model against simpler ones (I like bootStep AIC). Because SPSS cannot really conduct this type of analysis, in this section I can only recommend that, if you think there might be an interaction, you can try including it in a regression analysis and see whether it helps improve the size of your R^2 and also whether the

parameter coefficients for each part of the regression equation are statistical. You won't be able to formally test whether one model is better than another, but you can still take a good shot at heading toward the minimally adequate model.

The way to include an interaction term in a regression model is to simply create one yourself. I will illustrate this process with a file called `TOEFLexample.sav` that I made up to illustrate the TOEFL example in this chapter. Go to `TRANSFORM > COMPUTE VARIABLE`. In the box called "Target Variable" call the interaction between MLAT score and hours of study "MLAT_Hours" (the colon is an illegal character so I used the underscore). Then move the MLAT variable to the box labeled "Numeric Expression." From the keypad in this screen put a "*" after MLAT to show multiplication. Then add the Hours of Study variable and press OK. In the dataset there is a new variable. I went ahead and made up interaction terms for the rest of the possible interactions with the three variables. Now I'll create a regression with all three main effects (MLAT, Hours of Study, Gender) and all four interactions (MLAT_Hours, MLAT_Gender, Hours_Gender, MLAT_Hours_Gender). I'm interested in finding out the nature and size of the relationship between the TOEFL scores and my explanatory variables, so I'll use standard regression (so the METHOD is "Enter"), and this procedure is shown in Figure 1.

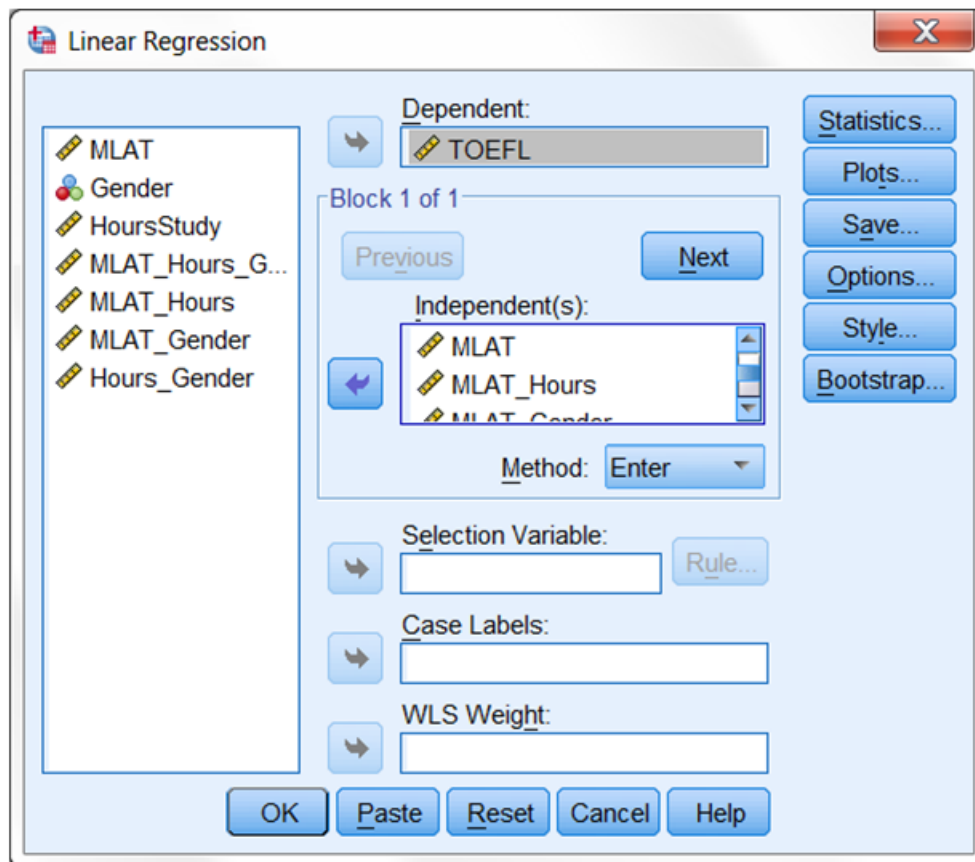


Figure 1 Regression with interaction terms in SPSS.

This model explains 41% of the variance ($R^2 = .41$), but none of the terms is statistical! You can tell this by looking at the column in the Coefficients table of output that gives the significance of the t -test (Table 1). None of these numbers is less than .05, meaning that none of the terms is statistical. This model is thus not a very good one. We want to get to the sparsest equation we can where all of the terms have a statistical coefficient.

Table 1 Testing the significance of terms in a maximal regression.

Coefficients ^a						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	427.584	1373.636		.311	.759
	HoursStudy	-.277	5.238	-.289	-.053	.958
	Gender	-308.130	844.838	-2.208	-.365	.719
	MLAT	1.435	30.283	.136	.047	.963
	MLAT_Hours	.015	.112	.869	.130	.898
	MLAT_Gender	6.201	18.474	2.159	.336	.741
	Hours_Gender	1.587	3.105	4.353	.511	.615
	MLAT_Hours_Gender	-.033	.066	-4.378	-.498	.624

Crawley's recommendation is to look at the maximal model first and then simplify by taking out the largest interaction terms first. So I'll run the regression again, taking out the three-way interaction, whose p -value is $p=.624$. For the new model now, R^2 is 40%, and still none of the terms have statistical coefficients. Therefore, the next step would be to take out the two-way interactions, one at a time. Choose the one with the highest p -value for the t -test (the Sig. column), which is Hours_Gender. When I run the regression now without this term, the $R^2 = .40$, and still none of the coefficients is statistical. The next highest p -value for the two-way interactions is for MLAT_Gender. However, I don't see a statistical coefficient appear until I have taken out all of the interactions. With all of the interactions gone and only main effects left (see Table 2), I see that Hours of Study has a statistical coefficient ($p=.004$), and $R^2 = .33$ (not shown in Table 2). I still want to simplify the model as much as I can until only statistical terms remain in the equation, so I'll take out the main effect which has the highest p -value, which is Gender with $p=.91$ (see Table 2).

Table 2 Output for a Regression with Only Main Effects.

Coefficients ^a					
Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
1	(Constant)	411.687	97.046	4.242	.000
	HoursStudy	.525	.164	.549	.3207
	Gender	-2.673	23.479	-.019	.910
	MLAT	.894	1.811	.085	.493

a. Dependent Variable: TOEFL

It turns out that with just MLAT and Hours of Study in the equation, the p -value of MLAT is still .33, not statistical. I therefore continue to reduce the equation to its simplest value, an equation with just the Hours of Study. My final regression accounts for 32% of the variance in TOEFL scores. The Coefficients output for this minimally adequate model is shown in Table 3.

Table 3 Output for a Minimally Adequate Regression.

Coefficients ^a					
Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
1	(Constant)	444.233	43.873	10.125	.000
	HoursStudy	.542	.155	3.505	.002

a. Dependent Variable: TOEFL

You might wonder why you shouldn't stick with the regression that includes MLAT since it accounts for a higher amount (33%) of the variance, but the fact is that the more variables you include, the higher the R^2 will always go. What you are looking for is the minimally adequate model. It might be that if we had been able to check model fit there would be no difference

between the model with two terms and that with one, in which case we would keep both Hours of Study and MLAT, but with SPSS we do not have a good way to determine that. I will also note that the model with only one term doesn't seem to have the troubles with non-constant variances (heterogeneity of variances) that previous models did (as seen in the chart plotting studentized residuals against fitted values), so the model seems to adhere to assumptions better than previous models.

My final regression equation will use the unstandardized coefficients (B) in the final model (you can see these in Table 3):

$$\text{TOEFL score} = 444.2 + .54 * \text{Hours of Study} + \text{error}$$

I have barely scratched the surface here of the many things that would need to be kept in mind when doing this type of model simplification, but hopefully this will give you an idea of how you can take charge of your regression model and look for the minimally adequate model.

Finding the Best Fit with R

When you begin to use R to perform regressions, the syntax for the command is so transparent that you cannot help but have some inkling of what you are doing! Once you understand what you are doing, you also come to understand that there are alternative ways of fitting a regression model to your data, and that you can easily alter the model. The quotation by John Fox at the beginning of chapter 7 summarizes this philosophy—the point is not in simply performing a regression, but in finding the regression model which best fits your data. Crawley says, “Fitting models to data is the central function of R. The process is essentially one of exploration; there

are no fixed rules and no absolutes. The object is to determine a **minimal adequate model** from the large set of potential models that might be used to describe a given set of data” (2002, p. 103).

Let’s get started then! The first step to understanding how to fit models is to understand a little bit more about the syntax of R’s regression commands. You’ve already seen that a plus sign (“+”) simply adds variables together. This syntax asks R to add in the main effects of each variable. Another possibility for fitting a model is to use interaction effects. The colon (“:”) indicates an interaction effect. If you have 3 variables in your regression, a full factorial model that includes all main effects and interaction effects is:

```
model = y~A + B + C + A:B + B:C + A:B:C
```

There is a shortcut for this syntax, however, which is the use of the star (“*”):

```
model = y~A*B*C
```

You can use the carat sign (“^”) to specify to what order you want interactions. If you only want the two-way interaction but not the three-way interaction, you would write:

```
model = y~(A + B + C)^2
```

which equals $A + B + C + A:B + A:C$. You could also achieve the same results this way:

```
model=y~A*B*C - A:B:C
```

which would remove the three-way interaction. Besides interaction terms, you might want your linear regression equation to include quadratic terms, which are squared terms. To raise a number to another number in R you use the carat sign, so A squared would be “A^2”. However, as we saw above, the carat is already used to represent an interaction model expansion, so we will need to use the “as is” function I (upper-case letter i) to suppress the incorrect interpretation of the carat as a formula operator. Thus, to create a model with two main effects plus the quadratic of each main effect, we would write:

```
model=y~A + I(A^2) + B + I(B^2)
```

Table 4 Operators in regression syntax.

Operator	Function	Example
+	adds parts of the	$y \sim A + B$
	regression equation	$y \sim A + B + A:B$
	together	
:	creates an interaction term	$y \sim A + B + C + A:B + A:C + A:B:C$
*	expands to all main effects and interactions	$y \sim A*B = y \sim A + B + A:B$
-	subtracts out terms	$y \sim A*B - A:B = A + B$
^N	expands to Nth-way	$y \sim (A+B+C)^2$

	interaction	$= A + B + C + A:B + A:C$
I	use arithmetic of syntax	$y \sim A + I(A^2)$

I'd like to make a comment here about the type of regression that R is doing. Standard discussions of multiple regression (such as Tabachnick and Fidell, 2001) will mention three types of regression: standard, sequential and stepwise, as noted at the beginning of Chapter 7 in the book. In standard regression only the unique portions of overlap between the explanatory and response variable are counted, while in sequential or hierarchical regression all areas of overlap are counted, so that order of entry into the regression matters. In the next section of this document I will be discussing how to find the best model fit of a regression, and I will be introducing something Crawley (2007) calls stepwise modeling. Now ordinarily, statistics book will tell you that stepwise regression is not a good idea; Tabachnick and Fidell call it a "controversial procedure" (2001, p. 133). However, the type of stepwise modeling that Crawley (and I) are advocating is not one that lets a computer determine the best model. Actually, you can do that using the `step()` command in R, but generally I recommend here conducting your own stepwise regression by hand first, and only later checking the step model. In both sequential and stepwise modeling the order of entry determines how much importance a given variable will have, if explanatory variables are correlated with each other. R is using this type of regression, since Crawley (2007, p. 328) says that "the significance you attach to a given explanatory variable will depend upon whether you delete it from a maximal model or add it to the null model. If you always test by model simplification then you won't fall into this trap."

To learn about more advanced operations, such as nesting, non-parametric models or polynomial regression, Chapter 9 of Crawley (2007) is a good place to start.

First Steps to Finding the Minimal Adequate Model in R

In order to find the best model for your data, Crawley (2007) recommends beginning with the maximal model and then simplifying your model where possible. The way to simplify is to remove one term at a time, beginning with the highest-order interactions first. If you have more than one higher-order interaction or main effect to choose from, start with the one that has the highest p -value (the least statistical one). You will then compare your simplified model with the original model, noting the residual deviance and using an ANOVA to check whether the removal of the term resulted in a statistically different model. If the ANOVA is statistical, meaning there is a statistical difference in the deviance of the two models, you will keep the term in the model and continue to test for other terms to delete. By the way, if you have reason to keep an interaction, you must also keep all of the components of the interaction in your regression model as well. In other words, if your Condition:Gender interaction is statistical, you must also keep the main effect for Condition and the main effect for Gender as well. If there is no difference, however, you will accept the second model as the better and continue forward with the simplified model. This process repeats until the model contains the least amount of terms possible.

This process probably sounds confusing so we will walk through an example of how it works. This whole process can get very complicated with a large number of terms (some statisticians say that you shouldn't test more terms than you can interpret, and draw the line at three explanatory variables), so I will explore the Lafrance and Gottardo (2005) data using only the three variables that were most important in the relative importance metrics: phonological

awareness in the L2 (PAL2), Kindergarten L2 reading performance (KL2WR), and naming speed, measured in the L2 (NS). If you are following along with me, first make sure you have the SPSS file Lafrance5.sav imported into R, and call it `lafrance5`. Section 7.4.4 in the book discussed how this dataset could be imputed, and ended up with a file called `implafrance`. If you do not have this file already created, here is how you would obtain it:

```
library(mice) #remember you can use import.packages("mice") #if you do not
#have this package uploaded yet

imp<-mice(lafrance5)

complete(imp)

implafrance<-complete(imp)

lafrance<-implafrance
```

Because I am going to be writing the variables again and again, I am going to change their names:

```
lafrance$PAL2<-lafrance$phonologicalawarenessinl2
lafrance$KL2WR<-lafrance$kindergl2readingperformance
lafrance$NS<-lafrance$namingspeed
lafrance$G1L1WR<-lafrance$grade1l1readingperformance
```

The research question is how well these three variables explain what is going on in the response variable, which is grade 1 L1 reading performance (G1L1WR).

The first step is to create the maximal model. Crawley (2007, p. 326) says a maximal model is one that “[c]ontains all (p) factors, interactions and covariates that might be of any interest.” This contrasts with the saturated model, which contains one parameter for every data point. The Lafrance and Gottardo dataset contains 40 rows (37 before we imputed data), so a saturated model would contain 40 parameters.

We will start with a full factorial model that includes the main effects of the three explanatory variables as well as the three 2-way interactions and one 3-way interaction. This model thus has 7 parameters. Be aware that we could indeed start with an even more complicated maximal model that would include quadratic terms if we thought those were necessary, but for this demonstration we won’t get that complicated.

```
model1=lm(G1L1WR~PAL2*KL2WR*NS, na.action=na.exclude, data=lafrance)
```

<code>model1=lm (. . .)</code>	This puts the regression model into an object called “model” (you can name it anything you like)
----------------------------------	---

<code>lm (formula, data, . . .)</code>	<code>'lm'</code> fits linear models
---	--------------------------------------

<code>G1L1WR ~ . . .</code>	The tilde (“~”) means the thing before it is modeled as a function of the things after it
-----------------------------	--

<code>PAL2*KL2WR*NS</code>	The explanatory variables; this formula expands to: <code>PAL2 + KL2WR + NS + PAL2:KL2WR + KL2WR:NS + PAL2:NS + PAL2:KL2WR:NS</code>
----------------------------	---

<code>na.action=na.exclude</code>	Excludes missing (NA) values; it is not needed strictly to fit the regression model but it is needed for the diagnostics, specifically involving the residuals, to come out correctly, so we will enter it here (although we don't actually need it because we imputed missing data)
-----------------------------------	--

<code>data=lafrance</code>	If you have not attached the dataframe, specify it here
----------------------------	---

Tip: If all of the variables in the dataframe are to be used, there is an alternative notation that is much shorter than typing in all of the variable names. The “.” to the right of the tilde means to include all of the remaining variables (N.B. items in red should be replaced with your own data name):

```
model=lm(G1L1WR ~ ., data=lafrance, na.action=na.exclude)
```

Having now fit a linear model to our equation, we can do all kinds of things with our fitted object (`model1`). First look at the summary command that we have seen before. Note that if you are following along with me, the values in your summary will be slightly different from mine. This is because the imputation procedure of the data will produce slightly different values every time it is used.

```
summary(model1)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.0294	9.0683	-0.224	0.824
PAL2	2.1802	6.2845	0.347	0.731
KL2WR	35.7481	43.4422	0.823	0.417
NS	0.4351	3.8463	0.113	0.911
PAL2:KL2WR	-21.5109	26.5296	-0.811	0.423
PAL2:NS	-0.2518	2.7183	-0.093	0.927
KL2WR:NS	-15.9204	20.0674	-0.793	0.433
PAL2:KL2WR:NS	9.5948	12.2571	0.783	0.439

Residual standard error: 0.33 on 32 degrees of freedom

Multiple R-squared: 0.5575, Adjusted R-squared: 0.4606

F-statistic: 5.758 on 7 and 32 DF, p-value: 0.0002252

We should look at the residual deviance (“Residual standard error”) of the model. It is .33 on 32 degrees of freedom. Models with a better fit will reduce the residual deviance. We don’t have anything to compare the .32 deviance to yet, but we will later. We also look at the t -tests for each term. At the moment, none are below $p = .05$ so none are statistical (remember that these p -values are unadjusted, meaning we haven’t taken into account the fact that we have a lot of tests and some might be low just by chance; of course, this may make us remember the discussion in Chapter 4 about how p -values are not very stable in the first place and may be high or low by chance anyway, but we won’t worry about that just now).

Since no term is statistical in this model because none of the p -values in the Coefficients area are under $p = .05$, we’ll take out the highest-order interaction, the three-way interaction, first. The easy way to do this is to use the update command. When you use this command, you’ll need to be careful with the syntax. After the name of the original model, you’ll need to use the sequence “comma tilde dot minus.”

```
model2=update(model1,~.- PAL2:KL2WR:NS, data=lafrance)
```

```
summary(model2)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.13849	8.94365	-0.127	0.899
PAL2	1.59187	6.20270	0.257	0.799
KL2WR	1.82986	3.11001	0.588	0.560
NS	0.07074	3.79558	0.019	0.985
PAL2:KL2WR	-0.75298	0.79531	-0.947	0.351
PAL2:NS	-0.01402	2.68536	-0.005	0.996
KL2WR:NS	-0.24097	1.21956	-0.198	0.845

Residual standard error: 0.3281 on 33 degrees of freedom
Multiple R-squared: 0.549, Adjusted R-squared: 0.467
F-statistic: 6.695 on 6 and 33 DF, p-value: 0.0001055

We notice that the residual error is slightly smaller (.3281) in this model and that we have one more degree of freedom. The unstandardized regression coefficients for several of the terms have become much smaller; for example, the unstandardized regression coefficient for **KL2WR** has shrunk from 35.7 to 1.8, and its standard error from 43.4 to 3.1.

Now we compare the two models using ANOVA:

anova(model1,model2)

Analysis of Variance Table

```
Model 1: G1L1WR ~ PAL2 * KL2WR * NS
Model 2: G1L1WR ~ PAL2 + KL2WR + NS + PAL2:KL2WR + PAL2:NS + KL2WR:NS
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1      32 3.4852
2      33 3.5519 -1 -0.066739 0.6128 0.4395
```

The ANOVA has a p -value = .4395, indicating that there is a non-statistical difference in deviance between the two models, so we retain the simpler model 2. In general, we prefer the simpler model to the more complex one, if they do not differ in explanatory value. When looking at different models, you should keep in mind that a saturated model (which has as many parameters as data points) will have a perfect fit, meaning that the R^2 value would be 1.0.

However, this model would have no explanatory value. So we are always doing a balancing act between trying to reduce the number of parameters (and get more degrees of freedom) and trying to improve the goodness of fit (a higher R^2 value). A value called Akaike's information criterion (AIC) calculates a number that "explicitly penalizes any superfluous parameters in the model" (Crawley, 2007, p. 353) while rewarding a model for better fit. In other words, the AIC tries to give a number to that balancing act between reducing the number of parameters (or conversely, increasing the degrees of freedom) and increasing the fit. Thus, the smaller the AIC, the better the model. We can use the AIC function to evaluate models as well:

```
AIC(model1,model2)
```

	df	AIC
model11	9	33.90093
model12	8	32.65966

Because model 2 has a lower AIC, it is the preferable model. The automatic stepwise deletion function, `boot.stepAIC`, which we will examine soon, will use the AIC to evaluate models.

We will now continue our deletion procedure by choosing the next term to delete. Since the 3-way interaction is gone, we will next select a 2-way interaction. There are three of them, so we'll pick the one with the highest p -value from the summary for model2. That was the `PAL2:NS` interaction with a p -value of .996.

```
model3=update(model2,~.- PAL2:NS, data=lafrance)
```

```
summary(model3)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.09357	2.41838	-0.452	0.65400
PAL2	1.55961	0.55961	2.787	0.00864 **
KL2WR	1.83465	2.92825	0.627	0.53515
NS	0.05135	0.77607	0.066	0.94763
PAL2:KL2WR	-0.75163	0.74139	-1.014	0.31783
KL2WR:NS	-0.24414	1.04180	-0.234	0.81612

We finally see a term in model 3 which is statistical, which is the main effect of phonological awareness. Let's compare model 2 and model 3:

```
anova(model2,model3)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	33	3.5519				
2	34	3.5519	-1	-2.936e-06	0	0.9959

There is no difference in deviance between the two models (the p -value is higher than .05), so we will accept the simpler Model 3 (you could also verify that the AIC of model 3 is lower than model 2) and delete the next 2-way interaction with the highest p -value of the two-way interactions, which is **KL2WR:NS**.

```
model4=update(model3,~.- KL2WR:NS, data=lafrance)
```

```
summary(model4)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.71736	1.78411	-0.402	0.69007
PAL2	1.48807	0.46265	3.216	0.00279 **
KL2WR	1.20490	1.14748	1.050	0.30090
NS	-0.07044	0.56853	-0.124	0.90211
PAL2:KL2WR	-0.68863	0.68154	-1.010	0.31924

Notice that the residual standard error of every term also decreases in each subsequent model.

Use **anova()** to compare models 3 and 4:

```
anova(model3,model4)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	34	3.5519				
2	35	3.5577	-1	-0.0057372	0.0549	0.8161

No difference, so we'll take out the last 2-way interaction term.

```
model5=update(model4,~.- PAL2:KL2WR, data=lafrance)
```

```
summary(model5)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.41715	1.75970	-0.237	0.81396
PAL2	1.40681	0.45574	3.087	0.00388 **
KL2WR	0.05265	0.12742	0.413	0.68192
NS	-0.14591	0.56377	-0.259	0.79725

```
anova(model4,model5)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	35	3.5577				
2	36	3.6615	-1	-0.10377	1.0209	0.3192

Again, there's no difference between the two models, but we still have a model that only has one statistical term so we'll continue now to delete main effects. From the summary for model 5, the main effect with the highest p -value is NS, with $p = .79$. Deleting this term:

```
model6=update(model5,~.-NS, data=lafrance)
```

```
summary(model6)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.85546	0.47199	-1.812	0.078038 .
PAL2	1.48240	0.34539	4.292	0.000122 ***
KL2WR	0.04795	0.12452	0.385	0.702384

```
anova(model5, model6)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	36	3.6615				
2	37	3.6683	-1	-0.0068131	0.067	0.7972

The ANOVA shows there is no difference in the models, so we remove the last main effect that is not statistical, **KL2WR**.

```
model7=update(model6,~.-KL2WR, data=lafrance)
```

```
summary(model7)
```

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.9713      0.3596  -2.701   0.0103 *
PAL2          1.5771      0.2398   6.577 9.24e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3113 on 38 degrees of freedom
Multiple R-squared:  0.5323,    Adjusted R-squared:  0.52
F-statistic: 43.26 on 1 and 38 DF,  p-value: 9.24e-08

```

The minimal adequate model is one with only phonological acquisition in it. This model explains $R^2 = .53$ of the variance in scores, with .31 residual standard error on 38 degrees of freedom. Remember that Model 1 had a residual standard error of .32 on 29 df, so we have improved the fit (by reducing the amount of error) and decreased the number of parameters (thereby increasing the df). We can check to make sure the minimal adequate model is not just the null model (with only one parameter, the overall mean) by comparing the fit of model 7 with the null model in model 8:

```
model8=lm(G1L1WR~1,data=lafrance,na.action=na.exclude)
```

```
anova(model7,model8)
```

Analysis of Variance Table

Model 1: G1L1WR ~ PAL2

Model 2: G1L1WR ~ 1

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	38	3.6830				
2	39	7.8753	-1	-4.1924	43.256	9.24e-08 ***

Thankfully, our one-parameter model is indeed statistically different from the null model. Note that the probability of the ANOVA test is recorded in statistical notation, with 9.24e-08 being shorthand for $p = .0000000924$. We continue by assuming that model7 is our minimal adequate model. We are not done yet, however, because we need to check regression assumptions for this model, which we will see in the next section.

This section has shown how to conduct a backwards stepwise analysis by hand until a model is reached where all terms are statistical. Another way to go about a stepwise analysis is to use the `boot.stepAIC()` function from the `bootStepAIC` package (Rizopoulos, 2009). This function uses a bootstrap procedure to help evaluate different models, and is much more accurate and parsimonious than the step procedure found in the base package of R, according to Crawley (2007).

Let's see what results we get if we use `boot.stepAIC()` with our original model:

```
library(bootStepAIC)#use install.packages("bootStepAIC") if you don't have it yet
```

```
boot.stepAIC(model1,data=lafrance)
```

The beginning of the output gives some numbers that summarize how many times each variable was selected as an independent predictor in each bootstrap sample (100 is the default). Austin

and Tu (2004) found 60% was an optimal cut-off level for including the predictors in the final equation. However, because this process is transparent (you can see the percentages in the column titled “Covariates selected”), this information gives the researcher tools to determine on their own the strength of the predictors.

```
Summary of Bootstrapping the 'stepAIC()' procedure for

Call:
lm(formula = G1L1WR ~ PAL2 * KL2WR * NS, data = lafrance, na.action = na.exclude)

Bootstrap samples: 100
Direction: backward
Penalty: 2 * df

Covariates selected
              (%)
PAL2           97
KL2WR          79
NS             66
PAL2:KL2WR     69
KL2WR:NS       56
PAL2:NS        54
PAL2:KL2WR:NS  41
```

The next part of the output concerns the stability of the predictors. The bootstrap procedure samples randomly with replacement, and each sample will contain regression coefficients.

Austin and Tu (2004) note that we would ideally want all of the coefficients to be either positive or negative, and that if half the coefficients were positive and half were negative this would be a sign of instability in the model. The `boot.stepAIC()` procedure, unlike the step procedure, is able to assess this measure of stability and provide it to the researcher. The output shows that **NS** and **PAL2:NS** were quite unstable.

```
Coefficients Sign
              + (%) - (%)
KL2WR         91.14  8.86
PAL2          78.35 21.65
NS            56.06 43.94
PAL2:KL2WR:NS 95.12  4.88
PAL2:NS       44.44 55.56
KL2WR:NS      12.50 87.50
PAL2:KL2WR     2.90 97.10
```

The next part of the output shows what percentage of the time the term was a statistical predictor at the $\alpha=.05$ level in the model.

```
Stat Significance
              (%)
PAL2          55.67
KL2WR         45.57
NS            24.24
PAL2:KL2WR    49.28
PAL2:KL2WR:NS 46.34
KL2WR:NS      44.64
PAL2:NS       18.52
```

From this output I might note that although phonological awareness (PAL2) was a statistical predictor over 50% of the time, the Kindergarten reading measure (KL2WR) was a predictor close to 50% of the time, as was the interaction between the two terms (PAL2:KL2WR). If the researcher had a theoretical reason for wanting to create a model with these three terms (even though we will see in the end of the output that this is not the model that the stepwise procedure finds) there would be ample room to argue that this was also a very good model.

The next part of the output lists the initial model (the one with 7 terms) and the final minimal adequate model that `boot.stepAIC()` finds, which has only the term for phonological awareness.

```
Initial Model:
G1L1WR ~ PAL2 * KL2WR * NS

Final Model:
G1L1WR ~ PAL2
```

Last of all is the analysis of deviance table for various permutations of the model (7 in all here). This recreates the steps we took manually as we deleted one term at a time (although there is a difference as the interaction between phonological awareness and the Kindergarten reading measure was deleted after the main term of naming speed (NS)).

	Step	Df	Deviance	Resid.	Df	Resid. Dev	AIC
1					32	3.485201	-81.61415
2	- PAL2:KL2WR:NS	1	6.673893e-02		33	3.551940	-82.85542
3	- PAL2:NS	1	2.935973e-06		34	3.551943	-84.85539
4	- KL2WR:NS	1	5.737209e-03		35	3.557680	-86.79083
5	- NS	1	1.560346e-03		36	3.559240	-88.77330
6	- PAL2:KL2WR	1	1.090265e-01		37	3.668267	-89.56641
7	- KL2WR	1	1.470134e-02		38	3.682968	-91.40642

The analysis of deviance column (“Deviance”) shows the change in deviance each time and AIC column shows the AIC value (which, since it is negative in this case, is smaller the larger the number is!). Because `boot.stepAIC` uses bootstrap resampling methods, it is an excellent performer that very rarely retains spurious terms in the final model.

I will note that the `boot.stepAIC()` command will not work with datasets that have missing values. You can try to clear out the missing values before running it

`(lafrance=na.omit(lafrance))` or impute the data as I did before I started. If we have data that includes missing values, the data are **non-orthogonal**. If your dataset is totally complete and there is no missing data, it is said to be **orthogonal**. Crawley (2007) notes that when datasets are non-orthogonal, as this one was before we imputed values, and explanatory variables correlate with each other, as they do here, then the importance you attach to a variable will depend on whether you subtract it from the maximal model or add it to the null model. Crawley recommends always subtracting from the maximal model to avoid problems with this, and that is what we did manually, and that is also what the bootstep model does as well, it just does it many times and then evaluates what percentage of the time terms were included.

Although `boot.stepAIC` is extraordinarily accurate at finding the model with the lowest AIC, this doesn’t mean you can just use it and forget the method of hand deletion that I have shown in

this section, because this method generalizes to mixed effect models in a later chapter that `boot.stepAIC` can't model. Another case you may be forced to use hand deletion in is one where you have more parameters than cases, which is called overparameterization. We will see an example of this in a later section of this chapter.

Once you have determined what your minimal adequate model is, you will want to check regression assumptions, as explained in Section 7.4.6 of the book. You should also find out the relative importance of the terms in your model. As mentioned in Section 7.4.5 of the book, this is best done using the `calc.relimp` function in the `relaimpo` package. Since our minimal adequate model involves only one term, we obviously cannot determine the relative importance of terms, but this process was illustrated above. Just don't forget to do it if you have more than one term left in your minimal adequate model!

Summary Finding a Minimal Adequate Regression Model

Here are the steps that Crawley (2007, p. 327) suggests for the model simplification process:

- 1 Fit the maximal model (include factors, interactions, possibly quadratic terms).
- 2 Inspect model summaries and take out highest-order non-statistical terms first, one at a time, using `update(model1, ~.-A:B:C)`.
- 3 Using ANOVA, compare model 1 to model 2. If there is no statistical difference between the models, retain newer model, inspect summary of newer model and continue to delete least statistical highest-order terms.
- 4 If the ANOVA shows a statistical difference, keep the older model.
- 5 I recommend checking your model with `boot.stepAIC` (`bootStepAIC` library).
- 6 Don't forget to check regression assumptions (see Section 7.4.6 in the book) and run `calc.relimp` (`relaimpo` library, described in Section 7.4.5 of the book) to determine the relative importance of the terms in your regression equation.

Notes:

- If an interaction is kept in a model, its component main effects must be kept too (in other words, if you have the interaction NS:PAL2, you must keep both NS and PAL2 in the model as well).
- Do not fit more parameters than you have data points (the number of data points equals the number of rows in your dataset that have no missing data).
- If deletion results in no statistical parameters, the null model ($y \sim 1$) is the minimal adequate one (back to the drawing board as far as experimental design!).
- If your design is non-orthogonal (meaning there are some missing values in some variables), the order in which you conduct the regression will make a difference in how important the variable seems to be.

Further Steps in Finding the Best Fit: Overparameterization and Polynomial Regression

In the previous section we explored a model of the data for Lafrance and Gottardo (2005) that included only 3 variables. But let's say we actually want to fit all 5 variables that the authors fit. If we fit a full factorial model with all main effects and interactions, we will have 1 5-way interaction, 5 4-way interactions, 10 3-way interactions, 10 2-way interactions, and 5 single terms, for a total of 31 interactions. Crawley (2007) says a useful rule of thumb is not to estimate more than $n/3$ parameters during a multiple regression. Since the Lafrance and Gottardo (2005) dataset has $n=37$ data points, that means we shouldn't estimate more than about 12 parameters at any one time. In the case of only 3 terms, there were 7 parameters and that was fine. However, for the case of 5 terms, we will want to conduct the regression in separate runs. One rule for doing this is that if you have an interaction, you'll need to also have the component main effects in the regression at the same time (see Crawley, 2007, p. 446).

Actually, just to make things a little more fun, let's start by including quadratic terms along with the main effects, just to check whether there is any curvature in our data. In the scatterplots for this data examined previously, there did seem to be some possibility of curvature in relationships between explanatory variables and the response variable. A regression involving quadratic (or higher) terms is called a polynomial regression by Crawley (2007). Although the model is still considered linear, having quadratic terms would mean we are fitting a line with some curvature.

If you're following along with me, use the imputed dataset called `lafrance` that we created in the section titled "First steps to finding the minimal adequate model in R." We'll need to change a couple more of the names to make it easier to see what we're doing.

```
lafrance$NR<-lafrance$nonverbalreasoning
```

```
lafrance$WM<-lafrance$workingmemory
```

Here is the first model we will try, which includes all 5 main effects and their associated quadratic terms:

```
model1=lm(G1L1WR~NR+I(NR^2)+WM+I(WM^2)+NS+I(NS^2)+PAL2+  
I(PAL2^2)+KL2WR+I(KL2WR^2),data=lafrance)
```

Remember that the `(NR^2)` term means it is a squared term. The I (capital letter "i") in front of the parentheses means that the caret (^) is performing its arithmetic job of squaring, instead of its regression function of expanding a regression term to a certain number of interactions. Let's look at a summary of this model:

```
summary(model1)
```

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.660e+01  1.445e+01   1.148   0.2602
NR           8.557e-02  7.906e-02   1.082   0.2880
I(NR^2)      -8.527e-04  8.003e-04  -1.066   0.2954
WM           1.612e-01  1.595e-01   1.011   0.3205
I(WM^2)      -1.831e-02  3.146e-02  -0.582   0.5650
NS           -2.167e+01  1.359e+01  -1.594   0.1217
I(NS^2)       4.842e+00  3.014e+00   1.607   0.1190
PAL2          7.982e+00  4.664e+00   1.711   0.0977 .
I(PAL2^2)     -2.382e+00  1.651e+00  -1.442   0.1599
KL2WR         1.280e-01  3.462e-01   0.370   0.7142
I(KL2WR^2)    7.666e-03  2.513e-01   0.031   0.9759

```

Nothing is statistical. Instead of a manual stepwise deletion, however, I am going to use the automatic **boot.stepAIC** procedure. One thing I like about this (over manual deletion in this situation of overparameterization) is that it will be more stable because it can simulate taking 100 samples of the variables.

```
library(bootStepAIC)
```

```
boot.stepAIC(model1, data=lafrance)
```

The automatic procedure tells me that my best model is the following:

```
model2=lm(G1L1WR~ NS + I(NS^2) + PAL2 + I(PAL2^2), data=lafrance)
```

```
summary(model2)
```

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   16.822      13.557   1.241   0.2229
NS            -19.454      12.806  -1.519   0.1377
I(NS^2)         4.298       2.837   1.515   0.1388
PAL2            7.440       3.632   2.048   0.0481 *
I(PAL2^2)      -2.029       1.249  -1.625   0.1132

```

Not all of the terms are statistical and it seems `boot.stepAIC` may have been generous at leaving terms in, but for now we'll keep these until we do a final culling.

Our next step will be to test the interaction terms. Following Crawley (2007), we will enter the names of the 10 2-way interactions, then randomize them. Note that to get the names of all of the two-way interactions, I just fit the full factorial model, asked for the summary, and looked at the summary not for any statistical information, but just for the full specification of all of the interactions!

```
getfullmodel=lm(G1L1WR~NR*WM*NS*PAL2*KL2WR,data=lafrance)
```

```
summary(getfullmodel)
```

#Output not printed but it tells me the name of all the 2-way interactions:

```
"NR:WM", "NR:NS", "WM:NS", "NR:PAL2", "WM:PAL2", "NS:PAL2", "NR:KL2WR",  
"WM:KL2WR", "NS:KL2WR", "PAL2:KL2WR"
```

We'll conduct 2 separate tests with about half of the 2-way interaction terms per test and all of the 5 main effects as well.

```
model3= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+  
PAL2:KL2WR + NR:NS + NS:PAL2 + NR:WM+NS:KL2WR, data=lafrance)  
model4= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+  
WM:KL2WR + NR:KL2WR+WM:PAL2+WM:NS+NR:PAL2, data=lafrance)
```

Here are coefficient terms for these two models:

`summary(model3):`

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.493901  12.057454   0.787   0.4374
NR            -0.304256   0.222001  -1.371   0.1810
WM            -0.729379   0.479151  -1.522   0.1388
NS            -3.768012   5.025259  -0.750   0.4594
PAL2           3.843637   6.350892   0.605   0.5497
KL2WR          3.654772   3.329725   1.098   0.2814
PAL2:KL2WR    -1.933163   1.024904  -1.886   0.0693 .
NR:NS          0.113845   0.091819   1.240   0.2250
NS:PAL2       -0.965094   2.738708  -0.352   0.7271
NR:WM          0.016525   0.009983   1.655   0.1086
NS:KL2WR      -0.212036   1.307879  -0.162   0.8723

```

`summary(model4):`

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -6.056450   5.668426  -1.068   0.2941
NR             0.096539   0.087349   1.105   0.2782
WM             0.207495   1.532853   0.135   0.8933
NS             0.132473   1.316886   0.101   0.9206
PAL2           5.286975   3.041512   1.738   0.0928 .
KL2WR          -1.684883   1.198879  -1.405   0.1705
WM:KL2WR       -0.096495   0.127417  -0.757   0.4550
NR:KL2WR        0.041144   0.026208   1.570   0.1273
WM:PAL2         -0.091446   0.401312  -0.228   0.8213
WM:NS           0.003571   0.505618   0.007   0.9944
NR:PAL2         -0.077842   0.064039  -1.216   0.2340

```

Performing a `boot.stepAIC (boot.stepAIC(model3,data=lafrance))` on both model 3 and model 4, it recommends leaving in these 2-way interactions only: `PAL2:KL2WR, NR:WM, WM:KL2W, NR:KL2WR, NR:PAL2.`

Crawley (2007) recommends putting all the interaction terms that are statistical (or nearly so) into one model with the 5 main effects and seeing which ones remain statistical. Since we only

found 5 2-way interaction terms to keep, this will not overload the number of parameters in one run:

```
model5= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+  
PAL2:KL2WR+ NR:WM +WM:KL2WR +NR:KL2WR +NR:PAL2, data=lafrance)
```

```
Coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) -6.63565      4.41919  -1.502   0.1440  
NR            0.13152      0.08393   1.567   0.1280  
WM           -1.44997      0.60857  -2.383   0.0240 *  
NS           -0.01864      0.60228  -0.031   0.9755  
PAL2          8.53645      3.15962   2.702   0.0114 *  
KL2WR        -0.03748      1.84660  -0.020   0.9839  
PAL2:KL2WR   -0.22792      1.02196  -0.223   0.8251  
NR:WM         0.03470      0.01366   2.540   0.0167 *  
WM:KL2WR     -0.31947      0.12467  -2.562   0.0158 *  
NR:KL2WR      0.02994      0.02330   1.285   0.2090  
NR:PAL2      -0.16137      0.06889  -2.342   0.0262 *
```

Not all of the 2-way interaction terms included are statistical, so we'll run a `boot.stepAIC` analysis to see what we should keep for this model.

```
boot.stepAIC(model5, data=lafrance)
```

The bootstrapped algorithm suggests taking out the main effect of NS and the first two interaction terms (`PAL2:KL2WR` and `NR:WM`). Updating the model:

```
model6=update(model5,~-NS-PAL2:KL2WR-NR:WM, data=lafrance)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5.44113	3.79132	-1.435	0.1609
NR	0.09704	0.08000	1.213	0.2340
WM	0.09042	0.06073	1.489	0.1463
PAL2	5.09077	2.79666	1.820	0.0781 .
KL2WR	-1.64649	1.12292	-1.466	0.1523
WM:KL2WR	-0.12153	0.07403	-1.642	0.1105
NR:KL2WR	0.04187	0.02386	1.755	0.0889 .
NR:PAL2	-0.07906	0.05832	-1.356	0.1847

Again, the bootstrapped procedure has left in terms that are not statistical, but we have whittled the second-order interactions down so we will keep these in the model for now. We now need to worry about the higher-order interactions. We repeat the process of testing the 3-way interactions in two separate runs with all of the main effects included as well. Another way to proceed would be to include only the 3-way interactions that involve the 2-way interactions that survived, and create a larger model with the 8 parameters in model 5 plus the 8 3-way interactions that involve any of the terms, but this surpasses our limit of 12 parameters by quite a bit, so I will continue with the simplification of the 3-way terms.

All 3-way interactions are:

"NR:WM:NS", "NR:WM: PAL2", "NR:NS: PAL2", "WM:NS: PAL2", "NR:WM: KL2WR", "NR:NS: KL2WR", "WM:NS: KL2WR", "NR: PAL2: KL2WR", "WM: PAL2: KL2WR", "NS: PAL2: KL2WR".

`model7= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+`

`NS:PAL2:KL2WR+NR:WM:PAL2+NR:NS:KL2WR+NR:WM:NS+`

`WM:PAL2:KL2WR, data=lafrance)`

`model8= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+`

`NR:PAL2:KL2WR+WM:NS:KL2WR+NR:WM:KL2WR+NR:NS:PAL2+`

```
WM:NS:PAL2, data=lafrance)
```

Summaries of the models show that none of the 3-way interactions are statistical, but

```
boot.stepAIC
```

 would keep two of the 3-way parameters:

```
NR:WM:NS
```

 and

```
WM:PAL2:KL2WR
```

(both are from model7).

Moving on to the 5 4-way interactions and the one 5-way interaction, I'll add the 5 main effects and test this 11-parameter model.

```
model9= lm(G1L1WR~NR+WM+ NS+ PAL2+ KL2WR+  
NR:WM:NS:PAL2 + NR:WM:NS:KL2WR + NR:WM:PAL2:KL2WR +  
NR:NS:PAL2:KL2WR + WM:NS:PAL2:KL2WR + NR:WM:NS:PAL2:KL2WR,  
data=lafrance)
```

According to the summary, none of the higher-way interactions are statistical. The

```
boot.stepAIC
```

 run, however, keeps

```
NR:WM:NS:KL2WR
```

 and

```
WM:NS:PAL2:KL2WR
```

. You can see that the whole process is quite complicated and lengthy, even using our automated procedure. It also depends somewhat upon the choices you make as to what to include in each regression run. At this point I will put together all of the terms that

```
boot.stepAIC
```

 has retained at each juncture, plus all 5 of the main effects, producing a 14-parameter model, which is higher than the 12 we wanted but we'll go with it.

```

model10=lm(G1L1WR~ NR + WM + NS + I(NS^2) + PAL2 + I(PAL2^2) + KL2WR+
WM:KL2WR +NR:KL2WR +NR:PAL2+ #two-way interactions
NR:WM:NS +WM:PAL2:KL2WR + #three-way interactions
NR:WM:NS:KL2WR + WM:NS:PAL2:KL2WR, #four-way interactions
data=lafrance)

```

The summary of model 10 results in a number of statistical effects, but not all, and here is `boot.stepAIC`'s final model, with 9 terms:

```

model11=lm(G1L1WR~ NR + WM + NS+ I(NS^2)+PAL2 +KL2WR+
WM:KL2WR + NR:PAL2+NR:WM:NS, data=lafrance)

```

```

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  31.939055   13.425588   2.379  0.02392 *
NR            0.061234    0.060069   1.019  0.31616
WM           -1.611992    0.505896  -3.186  0.00335 **
NS           -29.075848   11.934696  -2.436  0.02099 *
I(NS^2)       5.989491    2.594727   2.308  0.02805 *
PAL2          5.896447    2.206202   2.673  0.01205 *
KL2WR         0.855697    0.297588   2.875  0.00735 **
WM:KL2WR     -0.239070    0.078823  -3.033  0.00496 **
NR:PAL2      -0.108002    0.047503  -2.274  0.03031 *
NR:WM:NS      0.016642    0.004931   3.375  0.00205 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2846 on 30 degrees of freedom
Multiple R-squared:  0.6913,    Adjusted R-squared:  0.5987
F-statistic: 7.466 on 9 and 30 DF,  p-value: 1.216e-05

```

Amazingly, this results in a pretty good model with all higher-order terms being statistical (way to go, `boot.stepAIC`!). I repeated this entire process manually without `boot.stepAIC`, and this resulted in the deletion of all but the `PAL2` term. We can compare these two models:

```
model12=lm(G1L1WR~PAL2, data=lafrance)
```

```
summary(model12)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.9713      0.3596  -2.701   0.0103 *
PAL2           1.5771      0.2398   6.577 9.24e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3113 on 38 degrees of freedom
Multiple R-squared:  0.5323,    Adjusted R-squared:  0.52
F-statistic: 43.26 on 1 and 38 DF,  p-value: 9.24e-08
```

Model 11 with more terms has a higher R^2 (.69 vs. .53) and a smaller residual error (.28 vs. .31).

Examining the two models with an ANOVA shows a p -value that is over .05 ($p=.09$).

```
anova(model11, model12)
```

```
Model 1: G1L1WR ~ NR + WM + NS + I(NS^2) + PAL2 + KL2WR + WM:KL2WR + NR:PAL2 +
  NR:WM:NS
Model 2: G1L1WR ~ PAL2
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     30 2.4308
2     38 3.6830 -8    -1.2522 1.9318 0.09172 .
```

Remember that an ANOVA test between models tests the hypothesis that there is no difference in deviance between the two models (Crawley, 2007). If the p -value is greater than .05, we assume that there is no difference between models, and previously we have then always picked the simpler model. The probability here is low though and one could argue that there is a possible difference between models, in which case we should pick the more complex model that has lower deviance. If you calculate the deviance of each model, you see that the model with

more terms (model 11) has lower deviance (lower deviance is better than higher deviance as it is a measure of error).

```
deviance(model11)
```

```
[1] 2.476189
```

```
deviance(model12)
```

```
[1] 3.682968
```

Thus the model that one would pick would depend on the questions. If the question were whether phonological awareness was really, by itself, the best predictor of first grade reading scores, then this could clearly be argued for. If there is no real difference between the models, we will pick the simplest model. If the question were which combination of factors could produce the highest level of explanatory value, the more complex model 10 would be the best choice. As you can see, finding the best fit of a model is not a simple endeavor, and it is considerably lengthened by including more parameters! As Crawley (2002, p. 484) says, “If you really care about the data, this can be a very time-consuming business.”

Reporting the Results of a Minimal Adequate model

Crawley (2007) recommends that you report whether any data are missing or not, report on correlations between your explanatory variables and then present your minimal adequate model. You should also let your reader know what steps you took in your search by giving a list of the non-statistical terms that were deleted, and the change in deviance. If you report these things, then Crawley (2007, p. 329) says, “Readers can then judge for themselves the relative magnitude of the non-significant factors, and the importance of correlations between the explanatory

variables.” In the R summary the residual standard error is not the same as the deviance, but the **boot.stepAIC** summary gives the deviance of the model under the column “Resid. Dev” in the last bit of output where the deviance and AIC values of each tested model is listed. You might want to calculate the fit for each model manually with the **deviance()** command. Remember that lower deviance is better but a simpler model will have a higher deviance. The **anova()** function tests whether the difference in deviance is statistically different between models.

Here, then, is a summary of how I would report on my search for a minimal adequate model with three factors using the Lafrance and Gottardo (2005) data from the section called “First steps to finding the minimal adequate model in R”:

Using data from Lafrance and Gottardo (2005) I modeled a regression analysis with scores on grade 1 L2 reading performance with the three variables of phonological awareness in L2 (PAL2), Kindergarten scores on the L2 reading performance (KL2WR) and naming speed (NS). There were high intercorrelations among all 3 explanatory variables (PAL2-KL2WR, $r=.7$; PAL2-NS, $r=-.7$; KL2WR-NS, $r=-.4$). There were missing data points in the naming speed data, so the data were non-orthogonal, but I imputed the data first using R’s **mice** package. To search for a minimal adequate model, I started with a full factorial model of all 3 main effects plus all 2-way interactions and the 3-way interaction between terms. Deleting the terms and then checking for differences between models, my minimal model was one with only the phonological awareness term. This model explained $R^2=53\%$ of the variance in scores on the grade 1 reading test. For PAL2, the estimate for the unstandardized coefficient was 1.58, meaning that for every 1% increase in phonological awareness, there was a 1.58% increase in scores. This term was

statistical, $t=6.6$, $p<.0001$. The table below gives the steps of my model search and the change in deviance:

Model	Terms	deviance	Δ deviance
model1	PAL2*KL2WR*NS	3.48	
model2	-PAL2:KL2WR:NS	3.55	.07
model3	-PAL2:NS	3.55	.00
model4	-KL2WR:NS	3.56	.01
model5	-PAL2:KL2WR	3.66	.10
model6	-NS	3.67	.01
model7	-KL2WR	3.68	.01

In checking model assumptions, this model showed heteroscedasticity and non-normal distribution of errors.

Application Activity for Finding a the Best (Minimally Adequate) Fit

- 1 Howell (2002). Import the HowellChp15Data.sav file as **howell**. Chapter 15 in Howell included a dataset where students rated their courses overall and also aspects of the courses on a five-point scale (where 1 = very bad and 5 = exceptional). Use the **overall** variable as the response variable and the other variables (teaching skills of instructor,

quality of exams, instructor's knowledge of subject matter, the grade the student expected in the course where $F = 1$ and $A = 5$, and the enrollment of the course) as explanatory variables. Exercise #4 from Section 7.4.9 of the book addressed this dataset, and a scatterplot matrix found that all data have a linear relationship with **overall** except for **enroll**, which seemed to be a vertical line with a few outliers. Therefore, exclude the variable **enroll**. Start with the full factorial model and find the minimal adequate model (try doing this by hand for this item) and report the unstandardized coefficients and the R^2 for the model with only statistical predictors. Calculate the relative importance of the remaining terms of the regression equation. Comment on regression assumptions by examining residual plots.

- 2 Dewaele and Pavlenko Bilingual Emotions Questionnaire (2001–2003). Use the BEQ.Swear.sav file (import as **beqSwear**). Let us take as a given that the variables that might help explain how frequently a person swears in their L2 (**swear2**) are the frequency that the person uses their L2 (**l2freq**), the weight they give to swearing in their L2 (**weight2**), and their evaluation of their speaking and comprehension skills in L2 (**l2speak**, **l2_comp**). Exercise #5 from Section 7.4.9 of the book addressed this dataset, and scatterplot matrices looked fairly weird but I found that Loess lines on the plots showed linear trends except for plots that were combined with the variable of L2 comprehension. Therefore, we will exclude the variable **l2_comp**. Start with the full factorial model and conduct an analysis to determine which of these variables effectively predicts frequency in swearing in an L2 until you arrive at the minimal adequate model (you may try this by hand or use the **boot.stepAIC()** command). Calculate the relative importance of the remaining terms of the regression equation. Report the unstandardized

coefficients in a regression equation and the R^2 for the model with only statistical predictors, and comment on regression assumptions.

- 3 Larson-Hall2008.sav (import as `larsonhall2008`). Are amount of hours of input in a foreign language (`totalhrs`), aptitude (`aptscore`), and scores on a phonemic task (`rlwscore`) useful predictors of scores on a grammaticality judgment test (`gjtscore`)? Exercise #6 from Section 7.4.9 of the book addressed this dataset, and scatterplot matrices showed some curvature. Perform a regression using the GJT (`gjtscore`) as the response variable, and the three other variables as explanatory variables. In order to model the curvature, add in quadratic (squared) terms of all of the individual terms. Conduct an analysis to determine which model most effectively predicts frequency in swearing in an L2 until you arrive at the minimal adequate model (you may try this by hand or use the `boot.stepAIC()` command). Calculate the relative importance of the remaining terms of the regression equation. Report the unstandardized coefficients and the R^2 for the model with only statistical predictors, and comment on regression assumptions.

Bibliography

- Austin, P. C., & Tu, J. V. (2004). Bootstrap methods for developing predictive models. *The American Statistician*, 58(2), 131–137.
- Crawley, M. J. (2007). *The R book*. New York: Wiley.
- Crawley, M. J. (2002). *Statistical computing: An introduction to data analysis using S-PLUS*. New York: Wiley.
- Lafrance, A., & Gottardo, A. (2005). A longitudinal study of phonological processing skills and reading in bilingual children. *Applied Psycholinguistics*, 26(4), 559–578.

Rizopoulos, D. (2009). bootStepAIC: Bootstrap stepAIC. R package version 1.2-0 [Software].

Available from <http://CRAN.R-project.org/package=bootStepAIC>.

Tabachnick, B. G., & Fidell, L. S. (2001). *Using multivariate statistics* (4th ed.). Boston, MA:

Allyn & Bacon.