Richard Gonzalez
Psych 613
Version 3.1 (Dec 2022)

LECTURE NOTES #8: Advanced Regression Techniques I

Reading Assignment

KNNL chapters 8-11 and skim chapters 16-21; CCWA chapters 3, 5, 6, 8, 9

1. Polynomial Regression.

What can you do when the relation you want to examine is nonlinear? As discussed in lecture notes #7 sometimes it is possible to transform the variables so that the graph is approximately linear (e.g., rule of the bulge). A different technique for dealing with nonlinearity is to add terms of the same variable using a sequence of power transformations. For example, if X is the predictor variable, you could try adding an X^2 term to yield the regression equation

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon \quad (8\text{-}1)$$

Similarly, a third term, X^3, could also be added, etc. With each additional term, the curve of best fit is "allowed" one more bend.

A surprising result is that if N - 1 terms are included (N is the number of subjects) such as X^1, X^2, ..., X^{N-1}, the regression curve will fit the data perfectly (i.e., R^2 will equal 1). That is, the curve goes through every single data point because it is allowed to bend in all the right places. As we saw with polynomial contrasts in the ANOVA section of the course, each additional order adds one more bend to the curve. A straight line has 0 bends, a quadratic has 1 bend, a cubic has 2 bends ("S" shaped), etc.

Here is an example using data from Ott. There are 10 data points that I'll use to show how I can get a perfect fit every time. I'll fit a polynomial regression with nine (N - 1) predictors. The predictor variable is the number of days confined in a hospital bed and the dependent variable is the cost of the hospital stay.

The first plot shows the simple linear regression through the ten data points. Pretty nice fit. But, if we want a perfect fit, we can estimate the model with all terms up to

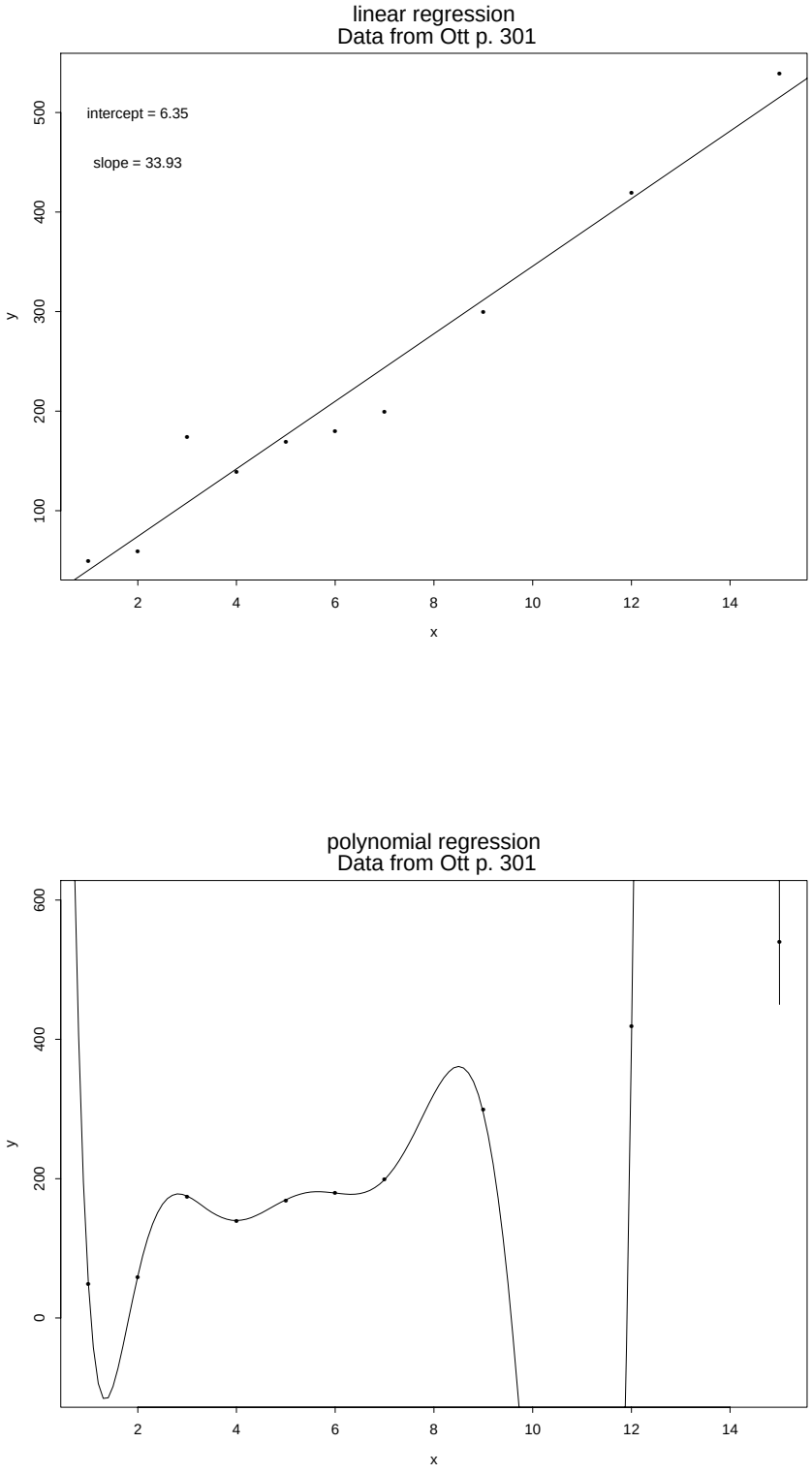


Figure 8-1: Demonstrating a complete polynomial

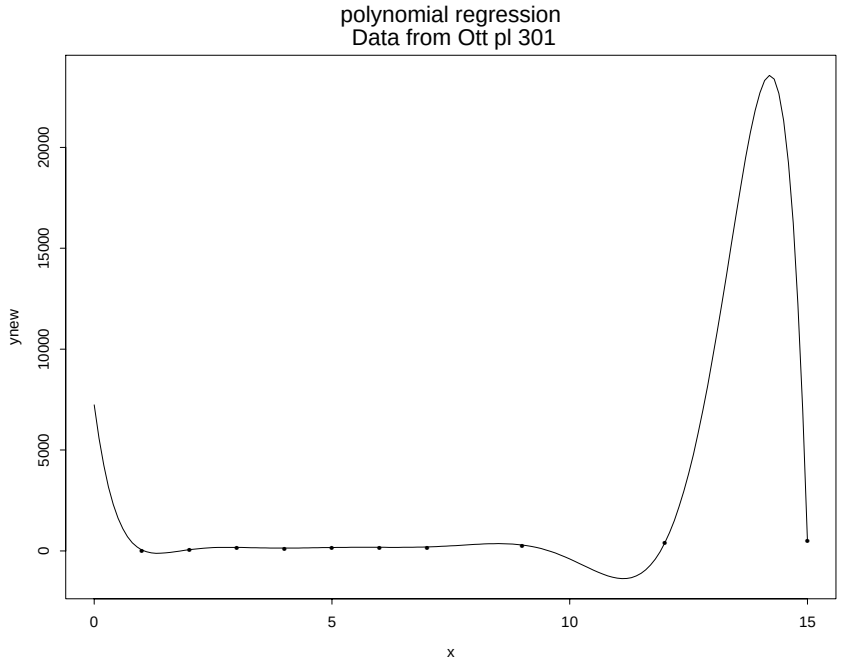


Figure 8-2: Full plot of the complete polynomial

x^9 because there are 10 cases. That curve is displayed in a "blown up" version on the second plot and "in all its glory" in the third plot.

The coefficients for this regression are[1]:

```
Intercept        X1       X2        X3       X4        X5       X6     X7
 7236.888 -18151.79 17803.32 -9145.282 2769.478 -519.2449 60.76967 -4.305

       X8           X9
0.1681843 -0.002769556
```

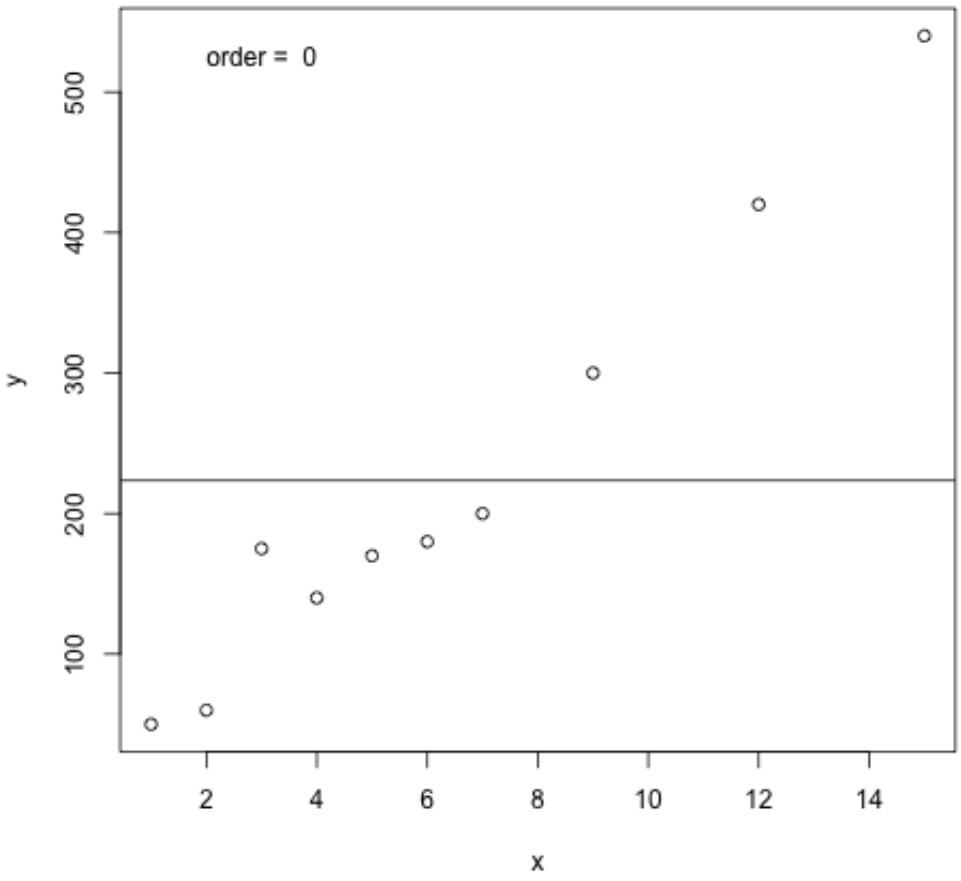
[1]To get these coefficients we need to use linear algebra techniques because most canned statistics packages will barf if there are 0 degrees of freedom for the error term (in addition to complaining about multicollinearity and "ill-conditioned matrices"). The matrix formulation is

$$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \tag{8-2}$$

where the prime indicates the transpose. If you want to attempt these computations yourself, you will also need a good algorithm to compute the matrix inverse.

Animation

Here is an animation of each order polynomial going from 0 (intercept only) to 9 (the maximim possible with N = 10). This pdf needs to be opened in adobe acrobat to view the animation.



I've shown one extreme case—including as many terms in the model as possible. This is clearly **not** something you would do in practice. But, you now know that a perfect fit to a data set can always be found—with enough terms in the polynomial, the curve can bend any way it needs to in order to go through every data point. The goal of most data analysis, however, is to find a parsimonious model that fits well, not necessarily a perfect fitting model to a particular sample.

There is a trade-off between how many parameters you include in the model, how

stable those parameter estimates are going to be and how well the model can predict out of sample. The more parameters you include in the model, the better you'll be able to fit the specific data points in your sample (and get better R^2's). However, the goal of data analysis is not to fit specific data points but to find general patterns that are replicable. The general pattern for the ten subjects was that the points were fairly linear. A parsimonious model with two parameters (intercept and one slope) can nicely capture the data in this example and in a way that will likely replicate across samples from the same population. The specific fit for these 10 data points (Figure 3) will likely not replicate for a new set of 10 observations.

leave one out cross-val (LOOCV)

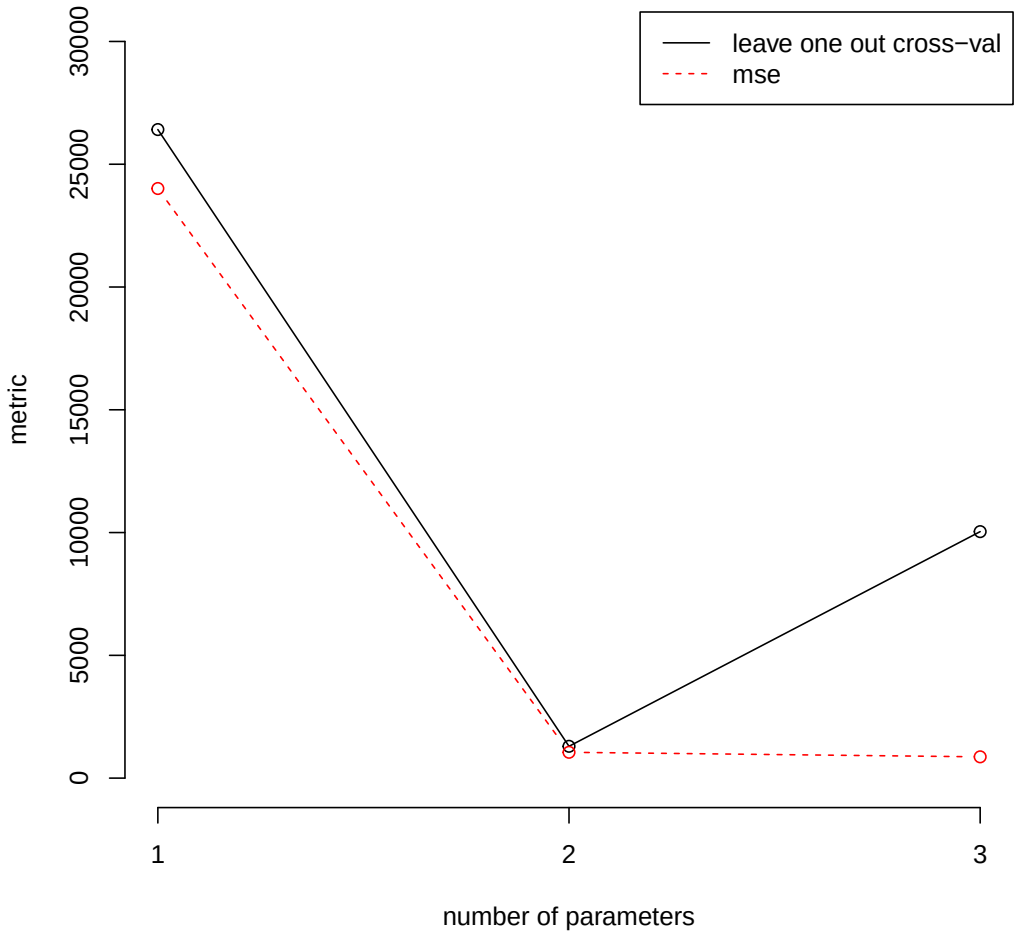
Here is a way to think about how well a model can predict out of sample, how that relates to the number of parameters and how overfitting enters the picture. I'll use the concept of "leave one out cross validation", which means fit the model on all but one data point, use that model to predict the remaining point, compute a discrepancy metric (such as squared deviation between prediction and actual data), do that N times each time leaving out a different data point and aggregate across the discrepancies[2]. I'll use the cv.glm() function in R's boot package[3].

```
library(boot)
data <- as.data.frame(matrix(scan("data.poly"), ncol=2,byrow=T))
colnames(data) <- c("y","x")

#compute leave one out cv
int <- cv.glm(data,glm(y~1,data,
              family=gaussian))$delta[2]
lin <- cv.glm(data,glm(y~x,data,
              family=gaussian))$delta[2]
quad <- cv.glm(data,glm(y~x + I(x^2),data,
              family=gaussian))$delta[2]
cub <- cv.glm(data,glm(y~x + I(x^2) + I(x^3),
                       data,family=gaussian))$delta[2]
quad <- cv.glm(data,glm(y~x + I(x^2) + I(x^3) + I(x^4),
                        data,family=gaussian))$delta[2]
quin <- cv.glm(data,glm(y~x + I(x^2) + I(x^3) + I(x^4)
                        + I(x^5),data,
                        family=gaussian))$delta[2]
```

[2]This logic should seem familiar because it is similar to Cook's distance from Lecture Notes 7 in the sense of rerunning the regression dropping a data point each time. The key difference is that Cook's distance examines the discrepancy between the full regression with all data points and the regression with the data point dropped, whereas cross-validation compares the predicted value $\hat{Y}$ from the regression with a data point omitted to the omitted data point's observed Y. The LOOCV does not make use of the full regression with all the data points.

[3]For those interested in R programming, it is a good exercise to program in R a more efficient way to cycle through several models, compute cross-validation, compute deviance in and out of sample, and produce a plot. The package caret also provides additional features for comparing multiple models. Also, a good exercise to use ggplot for nicer looking plots.

```
#save mse
mseint <- summary(glm(y~1,data,family=gaussian))$deviance/9
mselin <- summary(glm(y~x,data,family=gaussian))$deviance/8
msequad <- summary(glm(y~x+I(x^2),data,
                       family=gaussian))$deviance/7
msecube <- summary(glm(y~x+I(x^2)+I(x^3),data,
                       family=gaussian))$deviance/6

#produce plot
plot(1:3,c(int,lin,quad),ylim=c(0,30000),ylab=
       "metric",xlab="number of parameters",axes=F)
lines(1:3,c(int,lin,quad))
lines(1:3,c(mseint,mselin,msequad), type="b",col="red",lty=2)
legend("topright", legend=c("leave one out cross-val","mse"),
       lty=1:2,col=c("black","red"))
axis(1,at=1:3)
axis(2, at=seq(0,30000,5000))
```



The plot shows the mean square error (MSE) in red for the each model fit: intercept only, linear and quadratic. The intercept only has one parameter, the linear has two parameters, and the quadratic has three parameters. As we add more predictors, the MSE for the model on the entire sample decreases as you'd expect because R^2 can never get worse. However, the curve in black represents the leave one out cross-validation (LOOCV), which shows that adding more predictors does a good job of improving the out-of-sample prediction up to a point but then the prediction gets worse, which is a sign of overfitting. Adding more predictors fits better and better, but when predicting out of sample a model that contains additional predictors can do worse. For this example, we definitely do not want to go to three parameters because while the MSE is better, there is an indication that we are overfitting the model. We sometimes need to use several metrics to evaluate model fit because each metric has its pros and cons. Unfortunately, there are times when the different metrics do not

coincide and each metric makes a different recommendation. In those cases, we need to prioritize our goals such as minimizing mean square error for inference problems on the entire data set or the ability to predict out of sample for prediction problems.

For a satirical take on using polynomials to fit data perfectly, see Sue Doe Nimh's (aka Michael Birnbaum; ☺) paper in *American Psychologist*, 1976, 31, 808-, with comments in 1977, 782-.

An aside.... Inspection of the graph in the previous example suggests that the specific method I mentioned above for getting R^2=1 will work only when there are no ties on the predictor variable having different values of the criterion variable Y. If there are ties on the predictor variable having different criterion values, one can see that the graph of the "function" would have to be perfectly vertical and that is not permitted by the standard definition of a function (i.e., one to many mappings are not functions). So, the trick above for getting R^2=1 doesn't work when there are ties on the predictor variable. But, all is not lost. I can find other ways of getting R^2=1, even when the predictor variable consists entirely of the same number (i.e., all subjects are tied on the predictor). This is not too illuminating so I'll leave that as an exercise for anyone who is interested (hint: just a little linear algebra, the idea of "spanning a space," and the recognition that regression is really a system of equations is all you need to figure it out).

Another concern about using polynomials with high order is that they can produce systematic error in local regions of the curve. Intuitively, in order to bend in just the right way to match part of the curve, it may have to miss other parts of the data. This is known as Runge's phenomenon. Here is a good description: `https://en.wikipedia.org/wiki/Runge's_phenomenon`. A good general heuristic to follow is to keep things simple.

2. An interesting observation about multiple regression

When the predictor variables are not correlated with each other (i.e., correlations between all possible pairs of predictors are exactly 0), then the total R^2 for the full regression equals the sum of all the squared correlations between the criterion and the predictors. In symbols, for predictors 1 to k:

$$R^2 = r_{y1}^2 + r_{y2}^2 + \ldots + r_{yk}^2 \quad (8\text{-}3)$$

Thus, there is a perfect way to decompose the R^2, which is the omnibus summary of all predictors decomposed into separate orthogonal pieces for each predictor. Notice the similarity here with the pie chart and orthogonal predictors that we used in the context of ANOVA. In ANOVA factorial designs are orthogonal when the design is balanced (equal sample sizes across cells, recall Lecture Notes #5). The analogous situation to orthogonality in regression is when the predictors all correlate exactly 0

with each other, then the predictors are "orthogonal" and the overall omnibus R^2 will equal the sum of the squared correlations of each predictor with the outcome variable (Equation 8-3).

However, if there are correlations between the predictors so the predictors are not orthogonal with each other, then Equation 8-3 no longer holds. There is no unique way to decompose the omnibus R^2. The term *multicollinearity* refers to the case where the predictors have nonzero correlations with each other.

part correlation

In the situation of *multicollinearity* (i.e., correlated predictors), one can assess the unique contribution of a particular predictor variable by comparing the R^2 from two different regressions: a full regression that includes the predictor of interest and a reduced regression that omits the predictor of interest. The difference in R^2, i.e.,

$$R^2_{\text{full}} - R^2_{\text{reduced}} \tag{8-4}$$

is the unique contribution of that variable. If you take the square root of this difference in R^2 you have what is known as the "part correlation," also called the "semi-partial correlation."

decomposing R^2

We can use the part correlation to understand the total R^2 in the presence of correlated predictors because Equation 8-3 will not hold in this case. I'll denote the part correlation between variable Y and predictor variable 1 controlling for predictor variable 2 as $r_{y1.2}$, the part correlation between variable Y and variable 1 controlling for predictor variables 2 and 3 as $r_{Y1.23}$, etc. The R^2 for the k predictors is now given as:

$$R^2 = r^2_{y1} + r^2_{y2.1} + \ldots + r^2_{yk.[12..(k-1)]} \tag{8-5}$$

Focusing on just three predictor variables will make this more concrete. The following lines are three different, but equivalent, ways of decomposing R^2.

$$R^2 = r^2_{y1} + r^2_{y2.1} + r^2_{y3.12} \tag{8-6}$$
$$R^2 = r^2_{y2} + r^2_{y3.2} + r^2_{y1.32} \tag{8-7}$$
$$R^2 = r^2_{y3} + r^2_{y1.3} + r^2_{y2.31} \tag{8-8}$$

Thus, there are many ways to decompose an R^2 in the presence of correlated predictors.

For each line the last term on the right hand side is the unique contribution the last variable adds to R^2. That is, we see the unique contribution of predictor variables 3, 1, and 2, respectively in each line. It turns out that each β term in the full structural model (i.e., the model containing all three predictors) tests the significance of the unique contribution of the predictor variable. So, the t-test for each β corresponds to the test of significance for that predictor's unique contribution to R^2. Each β is the

unique contribution, and it is that reason why we interpret each β in a regression as "the unique linear contribution of that variable holding all other predictors fixed." We remove the linear effect of all other predictors and examine what is left over in the relation to the dependent variable Y.

Each of these decompositions reflect one particular order of entering a variable first, second, etc. We saw this analogous idea in the unequal sample size issue in ANOVA in Lecture Notes #5. The hierarchical method provided a particular order for entering main effects and interactions. The regression method examined each variable as though it was entered last; in the present terminology, the very last r^2 in each of the lines above (Equations 8-6 to 8-8).

suppressor effects

When the predictors are correlated we sometimes see strange results in the regression. For example, it is possible for R^2 to be greater than the sum of the individual correlations squared. This is known as a suppressor effect (see Hamilton, 1987, *American Statistician, 41,* 129-32, for a tutorial). This is well-known in the methods literature and is typically taught in many courses, but researchers seem to forget about suppressor effects when they analyze their own data. Often, when researchers add more predictors as control variables, or when testing mediation models (as we will see in Lecture Notes #13), or running complex structural models, they inadvertently may be including highly correlated predictors in their model.

partial correlation

There is another measure of unique contribution that some people like to use. It is called the "partial correlation;" it is given by

$$\sqrt{\frac{R^2_{\text{full}} - R^2_{\text{reduced}}}{1 - R^2_{\text{reduced}}}} \tag{8-9}$$

The numerator is the part correlation (aka semi-partial correlation), so this is just the part correlation normalized by a measure of the amount of error variance in the reduced model.

partial correlation through residuals

Another way to compute the partial correlation, which may shed light on how to interpret it, is to do two regressions. One regression is the reduced regression above using the criterion as the dependent variable and all other variables except the variable of interest as predictors. The second regression does not use the criterion variable. Instead, the variable of interest takes the role of the criterion and all other predictor variables are entered as predictors. Each of these two regressions produces a column of residuals. The residuals of the first regression are interpreted as a measure of the criterion that is purged from the linear combination of all other predictors, and the residuals from the second regression are interpreted as a measure of the predictor of interest purged from the linear combination of all other predictors. Thus, "all other predictors" are purged from both the predictor variable(s) of interest and the criterion variable. The correlation of these two residuals is identical to the partial correlation

(i.e., Equation 8-9). To make this concrete, suppose the criterion variable was salary. You want to know the partial correlation between salary and age holding constant years of education and number of publications. The first regression uses salary as the criterion with years of education and number of publications as predictors. The second regression uses age as the criterion with years of education and number of publications as predictors. The residuals from the first regression represent the portion of salary not linearly related to years of education and number of publications; the residuals from the second regression represent the portion of age not linearly related to the years of education and number of publications. The correlation between these two sets of residuals is equal to the partial correlation between salary and age controlling for both years of education and number of publications.

part correlation through residuals

The part correlation can also be computed from a correlation of residuals. One needs to correlate the raw dependent variable with the residuals from the second regression above that places one of the predictor variables as the criterion variable. To continue with the salary example, take the residuals from the second regression (age on years of education and number of publications) and correlate those residuals with the raw salary data.

Note that for the part correlation the "holding all other predictors" constant is done from the perspective of the predictor in question (in our example, age), not the dependent variable; whereas, the partial correlation the "holding all other predictors" constant is done from the perspective of both the predictor in question and the dependent variable. In other words, in the part correlation, the residuals represent "part" of predictor 1 (rather than the whole) because the linear relation of predictor 2 is removed from predictor 1. In the partial correlation, the linear effect of predictor 2 is removed from BOTH predictor 1 and the dependent variable.

In the special case of one control variable there are formulas expressed completely in terms of the three observed correlations, and they provide additional intuition about the part and partial correlations. These expressions are identical to the square root of the difference in R^2 between the full and reduced models I gave earlier in these lecture notes. The special case of one control variable corresponds to the full and reduced models differing by only one variable. The part correlation between variables 1 and y controlling for variable 2 is

$$\begin{aligned} r^{\text{part}}_{y1.2} &= \frac{r_{y1} - r_{12}r_{y2}}{\sqrt{1 - r^2_{12}}} \\ &= \sqrt{R^2_{\text{full}} - R^2_{\text{reduced}}} \quad \text{(connecting with the earlier definition; this equals the line above)} \end{aligned}$$

and the partial correlation between variables 1 and y controlling for variable 2 is

$$\begin{aligned} r^{\text{partial}}_{y1.2} &= \frac{r_{y1} - r_{12}r_{y2}}{\sqrt{(1-r^2_{12})(1-r^2_{y2})}} \\ &= \sqrt{\frac{R^2_{\text{full}} - R^2_{\text{reduced}}}{1 - R^2_{\text{reduced}}}} \quad \text{(connecting with the earlier definition; this equals the line above)} \end{aligned}$$

Those two equations for the part and partial correlation have the identical numerator; the key difference is the extra term in the denominator of the partial correlation.

relating part correlation to regression slope

The part correlation is the basic driver of the regression slope. For example, the slope β_1 in the model

$$Y = \beta_0 + \beta_1 X1 + \beta_2 X2 + \epsilon$$

can be computed as a function of the part correlation through

$$\beta_1 = \left(\frac{r_{y1} - r_{12}r_{y2}}{1 - r^2_{12}}\right)\left(\frac{s_y}{s_1}\right) \tag{8-10}$$

where s_y is the standard deviation of variable y and s_1 is the standard deviation of variable 1. The reason the square root does not appear in the denominator of Equation 8-10 is that we have the part correlation times a factor of $\frac{1}{\sqrt{1-r^2_{12}}}$, so typically people just write it as I did in Eq 8-10 with $(1 - r^2_{12})$ in the denominator.

These formulae are at the heart of understanding ideas such as suppressor effects, which can also happen for regression slopes. A suppressor effect, mentioned on page 8-11, occurs when the part correlation is more extreme than the simple correlation between the dependent variable y and one of the predictors. That is, controlling for variable 2 increases the part correlation between variable 1 and y relative to the correlation between variable 1 and y. This can happen due to the signs of the correlations. If the product term in the numerator is negative, then the product is added to the regular correlation between variable 1 and y. This effect will propagate through to the slope term in the regression.

I'll verify these equations on part and partial correlations, their relations to R^2, the computations using residuals, and the relation to the regression slope later when we do a numerical example.

SPSS

There is a sub-command in SPSS REGRESSION called "zpp". If you put zpp in the statistics sub-command of the regression command, as in

```
regression list_of_variables
```

```
/statistics anova coef ci r zpp
ETC...
```

you will get the part and partial correlations for each predictor automatically without having to compare all the regressions mentioned above. I recommend you always use the zpp option when running regressions in SPSS so that you automatically have the part and partial correlations in the output.

This is a good place to introduce another nice feature of the SPSS regression command. It is possible to have multiple method=enter sub-commands in the same regression command so that one can automatically test the change in R^2 when moving from a reduced model to a full model. For example, if X1, X2 and X3 are three predictors and you want to examine separately the change in R^2 in adding X2 to just having X1, and also the change in R^2 in adding X3 to the reduced model of both X1 and X2, you can use this syntax:

```
regression
 /statistics anova coef ci r zpp change
 /dependent y
 /method = enter X1
 /method = enter X1 X2
 /method = enter X1 X2 X3.
```

This command will run three regressions all in one output and also compute the change in R^2 in moving from the first method=enter line to the second, and again in moving from the second to the third. You can have as many method=enter lines as you like. I also added the word "change" to the STATISTICS sub-command; this produces the additional output for the F tests of the changes in R^2. The change output is usually next to the information about the R^2 values (at least in more recent versions of SPSS).

R

R users can load the ppcor package and use the pcor and spcor for the partial and semi-partial correlations, respectively[4]. Or I just run two regressions, save the R^2s and use the formulas I gave above to compute the partial and semi-partial correlations directly from R^2s of full and reduced regressions.

The way one computes a series of regression models in R (analogous to the multiple /method lines in SPSS) is by having several lm() commands. The anova() command compares two or more models using the increment in R^2 F test (analogous to the change option in the regression command in SPSS). Example:

[4]I've had some trouble with ppcor; look at SDSregressionR on github and its pCorr function as an alternative if ppcor continues to have issues.

```
model1 <- lm(y ~ X1, data=data)
model2 <- lm(y ~ X1 + X2, data=data)
model3 <- lm(y ~ X1 + X2 + X3, data=data)
anova(model1,model2,model3)
```

Some regression output prints both the raw beta (labeled "B") and the standardized beta (labeled "beta"). The standardized beta corresponds to the slope that you would get if all variables were converted to Z-scores (i.e., variables having mean 0 and variance 1). Note that when all variables are converted to Z-scores, the intercept is automatically 0 because the regression line must go through the vector of means, which is the 0 vector.

Some people prefer to interpret standardized betas because they provide an index for the change produced in the dependent variable corresponding to *one standard deviation change* in the associated predictor. I personally prefer interpreting raw score betas because it forces me to be mindful of scale, but the choice is yours. Some methodologists such as Gary King have criticized the use of standardized coefficients in interpreting multiple regression because the standardization can change the relative importance of each variable in the multiple regression.

Standardized betas are neither correlations nor partial correlations. It is possible for a standardized beta in a multiple regression to be greater than 1, especially in the presence of multicollinearity. Correlations can never be more extreme than -1 or 1. Deegan (1978, *Educational and Psychological Measurement*, p. 873-88) has an instructive discussion of these issues.

SPSS and R both provide standardized betas (e.g., in R they can be computed through the lm.beta package). One can always get standardized betas directly from a regression by first converting all variables (predictors and DV) to Z scores (i.e., for each variable subtract the mean and divide the difference by the standard deviation) and run the regression with all variables as Z scores.

3. Adjusting R^2 for multiple predictors

Wherry[5] (1931) noted that the usual R^2 is biased upward, especially as more predictors are added to the model and when sample size is small. He suggested a correction to R^2, known today as adjusted R^2. Adjusted R^2 is useful if you want an unbiased estimate of R^2 that adjusts for the number of variables in the equation. It is also useful when you want to compare the R^2 for the full model from two regressions that differ both in their predictors and the number of predictors.

[5]I gave the Wherry lecture in 2009 at the Ohio State University. His name appeared in my lecture notes way before 2009.

adjusted R^2

The adjusted R^2 used by SPSS is

$$1 - (1 - R^2)\left(\frac{N-1}{N-p-1}\right) \tag{8-11}$$

where N is the total number of subjects in the regression, p is the number of predictor variables, and R^2 is defined in the usual way as SSregression/SStotal. As N gets large the correction becomes negligible.

Other adjustments have been proposed over the years. Other statistics packages may differ from the formula presented, but they all accomplish the analogous goal: adjust R^2 downward in relation to the sample size and the number of predictors.

4. Multicolinearity

Recall that the interpretation of the slope β_i in a multiple regression is the change in Y produced by a unit change in the predictor X_i holding all other predictors constant. Well, if the predictors are correlated, then the "holding other predictors constant" is a somewhat meaningless phrase. One needs to be careful about a few things in regression when the predictors are correlated.

In general, even in the presence of multicollinearity the regression slopes are still unbiased estimates. Thus, it is possible to use the regression line to make predictions (i.e., the $\hat{Y}$ values are okay). However, the slopes depend on the other predictors in the sense that the slopes can change greatly (even sign) by removing or adding a correlated predictor. In the presence of multicollinearity the interpretation of the individual regression slopes is tricky because it is not possible to hold other variables constant. We'll come back to this point later when covering mediation where we will bring up the concept of total differential. Also, the standard errors of the slopes become large when there are correlated predictors, making the t-tests conservative and the confidence intervals wide.

Note that in polynomial regression the power terms will tend to be highly correlated. It didn't matter at the beginning of these lecture notes because I was just using the regression equation to predict the data points (and perfectly at that). If I wanted to test the significance of the different powers as separate predictors, I would run into a multicollinearity problem because the predictors would be highly correlated. One way of reducing the problem of correlated predictors in the context of polynomial regression is to "center" the variables, i.e., subtract the mean from the variable before squaring as in $(X - \overline{X})^2$. This simple trick helps reduce the effect of multicollinearity and makes the resulting standard errors a little more reasonable. I will illustrate the idea of centering below when I talk about interactions, which extends the idea of polynomial regression. Here I will make use of a toy problem. Consider the simple predictor X of the five numbers 1, 2, 3, 4, and 5. The X^2 of those numbers is, of

course, 1, 4, 9, 16, and 25. The correlation between these two variables, X and X^2, is .98 (note that a high correlation results even though linearity is violated). However, if we first center the X variable (subtract the mean), X becomes -2, -1, 0, 1, and 2, and the corresponding X^2 variable becomes 4, 1, 0, 1, and 4. The correlation between the centered X and its squared values is now 0. We went from a correlation of .98 to a correlation of 0 merely by centering prior to squaring. Thus, in a multiple regression if you enter both X and X^2 as predictors, you'll have multicollinearity. But if you enter both $X - \overline{X}$ and $(X - \overline{X})^2$ (i.e., mean centered and mean centered squared) the problem of multicollinearity will be reduced or even eliminated.

Tackling multicollinearity is not easy and the solution depends on the kind of question the researcher wants to test. Most approaches rely on some form of variable reduction—find a way to reduce the number of predictors so the remaining predictors do not have high correlation. Sometimes one can perform a principal components analysis (discussed later in the semester) to reduce the number of predictors prior to the regression; other times a modified regression known as ridge regression can be used. Ridge regression can be weird in that one gives up having unbiased estimators in order to achieve smaller standard errors. For details see Neter et al. or an excellent writeup by van Wieringen on arXiv. Ridge regression has other uses as well such as when one wants to use regression to make predictions and wants to reduce the number of predictors. Ridge regressions shrinks small betas even smaller, effectively making that predictor negligible; a related technique is the lasso, which is like ridge regression but it sends small betas to zero (rather than close to zero) so completely eliminating those predictors that have their betas set to zero producing a lean regression equation with fewer predictors. Both the lasso and ridge regressions are examples of what is called regularization that we will cover in a later lecture notes under the goal of reducing the number of predictors in a regression equation. These methods have a free parameter, called a tuning parameter, that needs to be set using additional procedures such as cross-validation. The phrase "no free lunch" applies here because while regularization methods may appear attractive, they come at the cost of introducing bias and adding complication in having to set the tuning parameter.

ridge regression

Ridge regression is related to Bayesian regression with a Gaussian prior distribution on the betas. Recall that in Lecture Notes 6 I made the connection between classic regression and Bayesian regression with noninformative prior distributions on the betas. So, ridge regression amounts to a change in the prior distribution, and the related lasso method also has a Bayesian counterpart merely by switching the prior distribution on the betas. This is one popular feature of the Bayesian approach: many well-known tests and procedures fall out automatically simply by an appropriate choice of prior.

5. Example of a multiple regression with correlated predictors

Here is a data set where multicollinearity in the predictors produces strange results in the interpretability of the slope estimates. First, we examine the scatter plots of the three variables of interest. I'm also throwing in a 3d plot to give us a different perspective. In the 3d plot, X is SES, Y is MOM, and Z is SCOR. The plots suggest there may also be a problem with the equality of variance assumption.

```
data list free /
SCHOOL     SLRY      WHTC       SES      TCHR       MOM      SCOR
  begin data
     1      3.83     28.87      7.20     26.60      6.19     37.01
     2      2.89     20.10    -11.71     24.40      5.17     26.51
     3      2.86     69.05     12.32     25.70      7.04     36.51
     4      2.92     65.40     14.28     25.70      7.10     40.70
     5      3.06     29.59      6.31     25.40      6.15     37.10
     6      2.07     44.82      6.16     21.60      6.41     33.90
     7      2.52     77.37     12.70     24.90      6.86     41.80
     8      2.45     24.67      -.17     25.01      5.78     33.40
     9      3.13     65.01      9.85     26.60      6.51     41.01
    10      2.44      9.99      -.05     28.01      5.57     37.20
    11      2.09     12.20    -12.86     23.51      5.62     23.30
    12      2.52     22.55       .92     23.60      5.34     35.20
    13      2.22     14.30      4.77     24.51      5.80     34.90
    14      2.67     31.79      -.96     25.80      6.19     33.10
    15      2.71     11.60    -16.04     25.20      5.62     22.70
    16      3.14     68.47     10.62     25.01      6.94     39.70
    17      3.54     42.64      2.66     25.01      6.33     31.80
    18      2.52     16.70    -10.99     24.80      6.01     31.70
    19      2.68     86.27     15.03     25.51      7.51     43.10
    20      2.37     76.73     12.77     24.51      6.96     41.01
end data.
```

```
 set width=80.

 correlation SCOR MOM SES.
```

```
                    - -  Correlation Coefficients  - -

             SCOR       MOM        SES

SCOR        1.0000      .7330**    .9272**
MOM          .7330**   1.0000      .8191**
SES          .9272**    .8191**   1.0000

* - Signif. LE .05      ** - Signif. LE .01      (2-tailed)
```

Something interesting to point out in the individual scatter plots. It seems that SES has less variability around SCOR than does MOM (the first two scatter plots). It turns out that predictors with less variability will be more likely to stand out as being the predictor that is more significant in a multiple regression (all other things being equal). Recall that the estimate of the slope has the variance of the predictor variable in the denominator. So be careful of studies that pit predictors against each other to

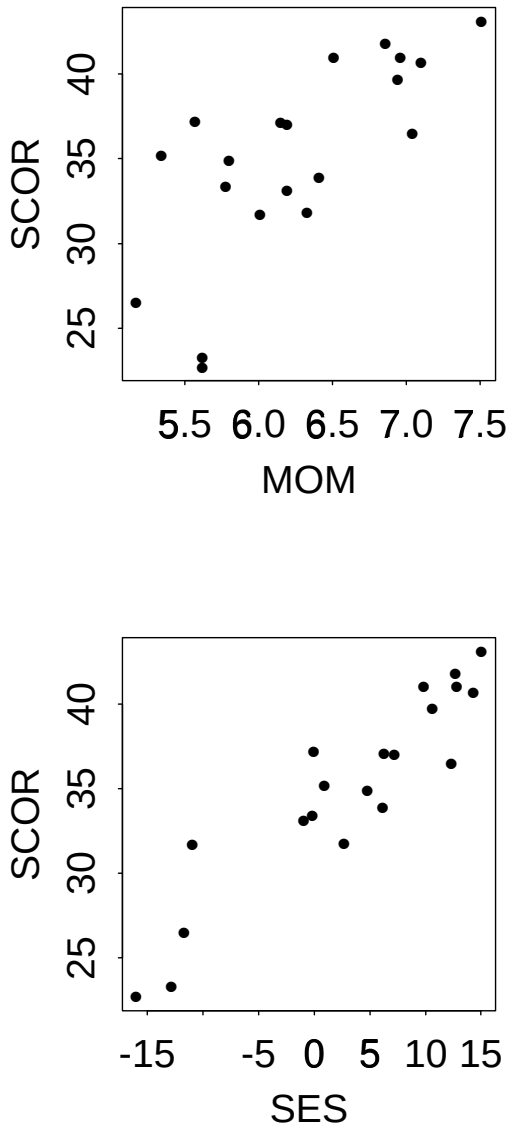


Figure 8-3: Plots of dependent variable against the independent variables.

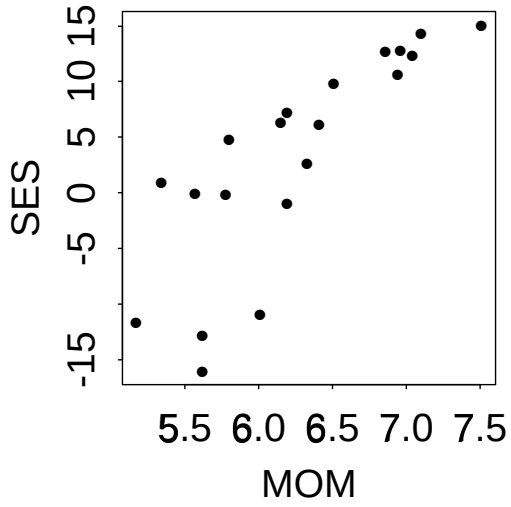


Figure 8-4: Plot of two independent variables

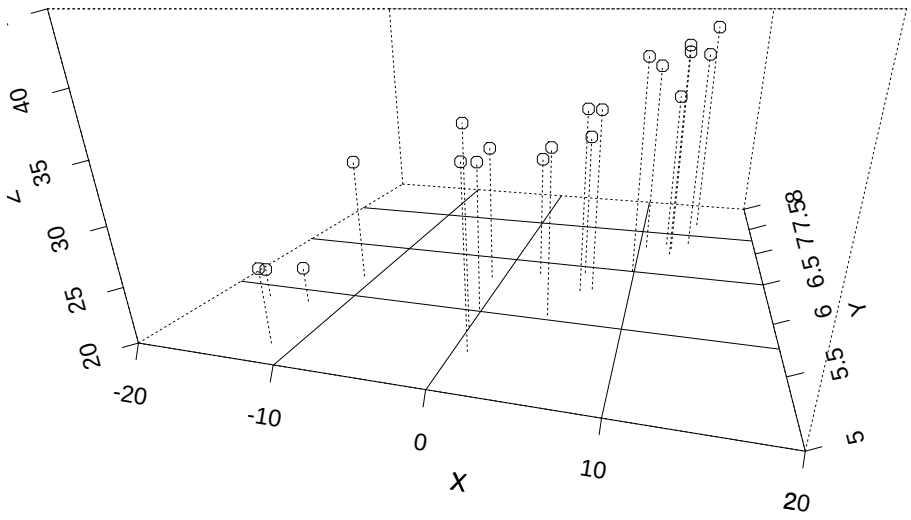


Figure 8-5: Three dimensional scatter plot

find the best single predictor. Usually, such a procedure merely finds the predictor that is most reliable.

Now we move to a series of regression. Suppose the researcher enters MOM as a predictor of SCOR and then wants to see whether SES adds any predictive power (i.e., is SES essential).

```
regression variables = all
  /stats anova r ci coef zpp
  /dependent SCOR
  /method=enter MOM
  /method=enter SES.

 1..    MOM Regression


Multiple R           .73299
R Square             .53727
Adjusted R Square    .51156
Standard Error      4.06545

Analysis of Variance
                   DF      Sum of Squares      Mean Square
Regression          1           345.42292        345.42292
Residual           18           297.50145         16.52786

F =      20.89944       Signif F =  .0002


Variable              B        SE B     95% Confdnce Intrvl B       Beta

MOM            6.516436    1.425420     3.521740     9.511133    .732986
(Constant)    -5.677808    8.962226   -24.506747    13.151130


Variable     Correl Part Cor  Partial         T  Sig T

MOM         .732986  .732986  .732986     4.572  .0002
(Constant)                                -.634  .5344


   2. MOM & SES regression

Multiple R           .92830
R Square             .86175
Adjusted R Square    .84548
Standard Error      2.28660

Analysis of Variance
                   DF      Sum of Squares      Mean Square
Regression          2           554.03887        277.01943
Residual           17            88.88551          5.22856

F =      52.98198       Signif F =  .0000


Variable              B        SE B     95% Confdnce Intrvl B       Beta

MOM            -.713569    1.397457    -3.661945     2.234807   -.080264
```

```
SES             .600056     .094997      .399630      .800482    .992902
(Constant)    37.661396    8.513826    19.698794    55.623998


Variable     Correl Part Cor  Partial         T  Sig T

MOM         .732986 -.046048 -.122905     -.511  .6162
SES         .927161  .569631  .837393     6.317  .0000
(Constant)                                4.424  .0004
```

I used the zpp option in the statistics in sub-command, which printed the part correlation (R users can use the ppcor package as described above). Recall that the squared part correlation is identical to the increment in R^2 of adding that predictor last. Double check this for your own understanding (e.g., adding the predictor SES in addition to the single predictor MOM, leads to a part correlation = .5696, which is equal to the square root of the R^2 change .86175-.53727).

R examples

The same regressions can be run in R and the ppcor package can be used for the part and partial correlations (or just correlated residuals as described above).

```
library(ppcor)
data <- read.table("dat", header = T)
summary(lm(SCOR ~ MOM, data = data))
```

```
##
## Call:
## lm(formula = SCOR ~ MOM, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2446 -1.8876  0.1324  2.7201  6.5813
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept)   -5.678      8.962  -0.634
## MOM            6.516      1.425   4.572
##             Pr(>|t|)
## (Intercept) 0.534358
## MOM         0.000237 ***
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
```

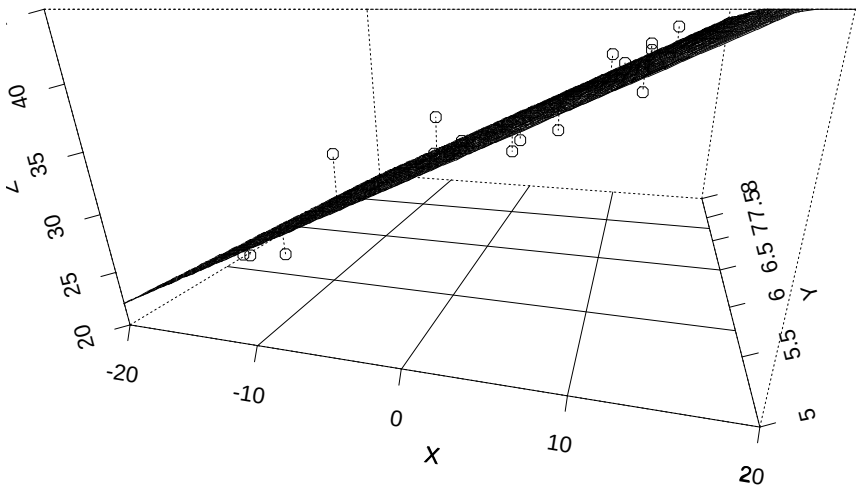


Figure 8-6: The same data with the regression fit using two predictors.

```
##
## Residual standard error: 4.065 on 18 degrees of freedom
## Multiple R-squared:  0.5373,Adjusted R-squared:  0.5116
## F-statistic:  20.9 on 1 and 18 DF,  p-value: 0.0002366
```

```
summary(lm(SCOR ~ MOM + SES, data = data))
```

```
##
## Call:
## lm(formula = SCOR ~ MOM + SES, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5206 -1.3659  0.0029  0.9510  4.9218
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept)  37.6614     8.5138   4.424
## MOM          -0.7136     1.3975  -0.511
## SES           0.6001     0.0950   6.317
##             Pr(>|t|)
## (Intercept) 0.000372 ***
## MOM         0.616184
## SES         7.74e-06 ***
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
##
## Residual standard error: 2.287 on 17 degrees of freedom
## Multiple R-squared:  0.8617,Adjusted R-squared:  0.8455
## F-statistic: 52.98 on 2 and 17 DF,  p-value: 4.963e-08
```

```
# part correlation (printing only relevant two values)
spcor(data[, c("SES", "MOM", "SCOR")])$estimate[3, 1:2]
```

```
##         SES         MOM
##  0.56963124 -0.04604776
```

```
# partial correlation (printing only relevant two values)
```

```
pcor(data[, c("SES", "MOM", "SCOR")])$estimate[1:2, 3]
```

```
##        SES        MOM
##  0.8373928 -0.1229045
```

Note that the sign of the slope for MOM changed when we went from one predictor to two predictors—the effect of multicollinearity. In this example, the new slope was not significantly negative but we can imagine cases where significance might have occurred in the second regression too. The lesson here is that we need to be careful in how we interpret the partial slopes in a multiple regression when there is multicollinearity. When the predictor variables are correlated it is difficult to make the "hold other variables constant" argument for the interpretation of a single slope because if the predictors are correlated it can't be possible to "hold constant" all other predictors without affecting the variable in question. Remember earlier in these lectures notes I mentioned the concept of suppressor effects, how they are subtle and how they aren't always appreciated by researchers?

Another weird problem that can occur with multicollinearity is that each of the predictor variables may not have a statistically significant slope (i.e., none of the t tests are statistically significant), yet the R^2 for the full model can be statistically significant from zero (i.e., the model accounts for a significant portion of the variance even though no single variable has significant unique variance—this test is the same as the F test for the entire regression). In other words, there is a sufficient amount of "shared" variance that all predictors soak up together (yielding a significant R^2 for the total regression) but none of the individual predictors account for a significant portion of unique variance as seen in the nonsignificant slopes for each variable. We wouldn't see such a thing occur in the context of orthogonal contrasts in ANOVA because by the definition of orthogonality the predictors are independent (hence multicollinearity cannot occur).

In class I will demonstrate a three dimensional plot to illustrate multicollinearity. The regression surface is balanced on a narrow "ridge" of points and is unstable; the surface can pivot easily in different directions when there is multicollinearity, the implication being that the standard errors of the slopes will be excessively high.

Next, I'll verify the earlier claims I made with formulas about the relation between part correlations, R^2, correlations of residuals and regression slopes.

We saw the part correlation between MOM and SCOR controlling for SES was -.04604776. That can be computed directly from the three pairwise correlations.

```
ry1 <- cor(data$MOM, data$SCOR)
ry2 <- cor(data$SES, data$SCOR)
r12 <- cor(data$MOM, data$SES)

(ry1 - ry2 * r12)/sqrt(1 - r12^2)
```

```
## [1] -0.04604776
```

The same part correlation can be computed by correlating the residuals from a regression that treats MOM as the dependent variable and SES as the sole predictor (the residuals represents the portion of the MOM data that has the linear relation with SES subtracted out) and the dependent variable as described above

```
cor(data$SCOR, resid(lm(MOM ~ SES, data)))
```

```
## [1] -0.04604776
```

The part correlation can also be computed through the square root of the difference in R^2 between full and reduced regressions, being careful to keep track of the sign. Here I'll compute the part correlation for MOM by using the difference in R^2 for the full regression with both MOM and SES as predictors and the R^2 for the reduced regression with only SES as the predictor (so the difference in those two R^2 values tests the increment of adding MOM over and above SES):

```
full <- summary(lm(SCOR~ MOM + SES,data))$r.squared
full
```

```
## [1] 0.8617481
```

```
reduced <- summary(lm(SCOR~ SES,data))$r.squared
reduced
```

```
## [1] 0.8596277
```

```
sqrt(full - reduced)
```

```
## [1] 0.04604776
```

```
#recall when there is only one predictor in the regression,
#the R2 is equal to the square of the correlation
cor(data$SCOR, data$SES)^2
```

```
## [1] 0.8596277
```

```
#which is equal to the R2 for the reduced term a few lines up
```

Finally, we saw earlier that the slope for the MOM variable in the regression that had MOM and SES both as predictors was -0.7136. That same value emerges from the relation between the slope and the part correlation described in Equation 8-10

```
(ry1 - ry2*r12)/(1-r12^2) *
  sqrt(var(data$SCOR)/var(data$MOM))
```

```
## [1] -0.7135687
```

It all checks out. There are many ways of computing the same quantity, each approach illuminates a slightly different (but equivalent) way of interpreting a slope in a regression equation.

Big picture: Every slope in a regression equation is related to a part correlation that provides a metric of the relation between that predictor and the dependent variable holding constant the other predictors. So, every slope is also a measure of the unique contribution of that predictor variable holding constant the other predictor variables. Each slope reflects the unique additional contribution of that predictor to the overall regression equation. It is as though that variable was entered last in a sequence of regressions. This logic is related to the Type III sum of squares we saw in LN5 in the unequal N problem.

6. Summary of remedial measures for multicollinearity

   (a) aggregate highly correlated predictors (or simply drop redundant predictors from the analysis)

(b) sample the entire space predictor space to avoid the "narrow ridge" problem (more on this when we cover interactions later in these lecture notes)

(c) ridge regression or lasso—not necessarily a great idea because even though you get smaller standard errors for the slope, the slope estimates themselves are biased; it is a tradeoff in that ridge or lasso reduced the increase in standard error due to multicollinearity at the expense of introducing biased estimators of the slopes.

7. Interactions

One can also include interaction terms in multiple regression by including products of the predictor variables of interest. For example, using the three dimensional structure I presented in Lecture Notes #7, the curved surface that results when one includes three predictors: X_1, X_2, and X_1X_2 is shown in Figure 8-7. By including an interaction there is no longer a plane but a curved surface that is fit to the data. This is tantamount to saying that the effect on the dependent variable of each dependent variable depends not only on the marginal effects (main effects) but also on something related to each specific combination of predictors (two-way interactions, three-way interactions, etc).

In the same way that several variables can be added to the regression equation, it is possible to add interaction terms, i.e., new variables that are products of variables already in the equation. More concretely, suppose I am predicting subjects' weight loss (denoted W) from the amount of time they spend exercising (denoted E) and the average daily caloric intake over a 3 week period (denoted C). A simple model with only main effects would be

$$\mathrm{W} = \beta_0 + \beta_1\mathrm{E} + \beta_2\mathrm{C} \tag{8-12}$$

I can include the possibility of an interaction between exercise and caloric intake by adding to the model a third variable that is the product of C and E. In other words, create a new column of data in your data set that is the product of C and E, then add that new variable as another predictor in the regression equation

$$\mathrm{W} = \beta_0 + \beta_1\mathrm{E} + \beta_2\mathrm{C} + \beta_3\mathrm{EC} \tag{8-13}$$

You can do this in SPSS by first creating a new variable with the COMPUTE command that is the product of E and C. This new variable can then be entered into the regression like any other variable. In R, just create a new variable that is the product and include it in the regression equation (or you can use the * or : in the formula notation in R).

Equation 8-13 is the structural model for the case of two main effects and the two-way interaction. It will be more illuminating to re-arrange the terms to produce this

Interaction with residuals plotted.

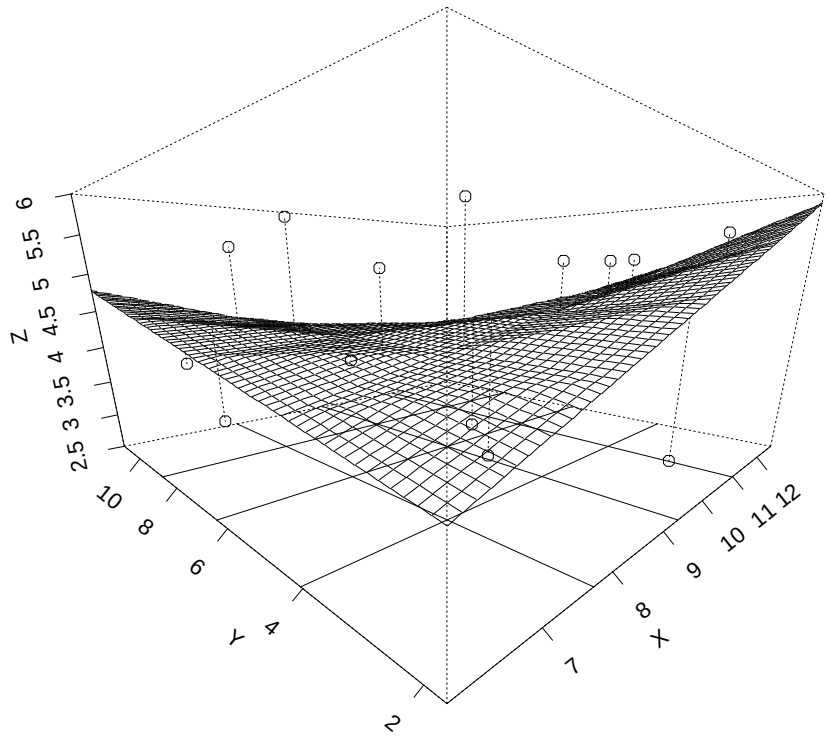


Figure 8-7: Interaction surface.

equivalent form:

$$\mathrm{W} = (\beta_0 + \beta_2\mathrm{C}) + (\beta_1 + \beta_3\mathrm{C})\mathrm{E} \quad (8\text{-}14)$$

The first term in parentheses is an intercept that varies with C and the second term in parentheses is a slope that varies with C. To clarify, I reprint Equation 8-14 with boxes around those terms

$$\mathrm{W} = \boxed{(\beta_0 + \beta_2\mathrm{C})} + \boxed{(\beta_1 + \beta_3\mathrm{C})}\,\mathrm{E}$$

intercept slope

Thus, the inclusion of the interaction term can be conceptualized as a linear relation between W and E where both the intercept and the slope of that linear relation depend on an additional predictor C.

moderator

Sometimes the term *moderator* is used, such as variable C moderates the intercept and moderates the slope between W and E. In other words, there isn't a single intercept but different intercepts for different values of C and there isn't a single slope but different slopes for different values of C. This notion contrasts with the standard linear regression model that does not include an interaction, where every subject has the same slope and intercept. The interaction ends up tailoring the slope and intercept for each subject on the basis of that subject's value on predictor C. You should understand the role that β_2 and β_3 play in this regression. I could have written an analogous equation with the roles of C and E interchanged (the p-values from such a model would be identical but the interpretation of the parameters would be slightly different because then it would be interpreted as predictor E moderating the slope and intercept that link dependent variable W and predictor C). The logic of growth curve analyses common in many areas such as developmental psychology extends this idea of an interaction by allowing each subject to have their own slope and intercept, and each can also be a function of other predictors (e.g., an extension of Equation 8-14 applied separately to each subject where subject is treated as a random effect factor so that each subject has their own set of βs). We'll return to this idea in a later lecture notes.

An important caveat is that if E and C are correlated, the inclusion of the interaction term renders the tests of the main effects difficult to interpret (the slopes for the "main effects" will have high standard errors due to the multicollinearity with the interaction term). We saw this in Lecture Notes 5 in the context of unequal N and the issue being in how to interpret the main effects.

Centering helps with this problem, as we saw for polynomials and also see below. When the interaction is included, the linear effects of E and C should also be present in the regression. It wouldn't be good to include the more complicated product term without including the simpler main effects that makeup the interaction; although in

some settings it would be ok to omit the main effects, such as if there is good reason to only expect a product term as in the physics equation F = m*a (it wouldn't make any sense to have F = m+a+m*a or rather the "slopes" attached to the main effect will go to 0 leaving just the m*a term).

A common solution to testing interactions created from continuous variables is to first perform a regression that includes only the main effects. Interpret the parameters and tests of significance for those main effects. Then run a second regression by adding the two-way interactions along with the main effect predictors, interpret the new interaction parameters (but don't reinterpret the main effects in this second regression), and perform the tests of significance on the interaction terms. Repeat with three-way interactions, etc. This is analogous to the "sequential procedure" we saw as a method for dealing with unequal sample sizes in LN5[6].

The sequential method is preferred if you are primarily checking whether the interaction adds anything over and above that already predicted by the main effects. That is, the method is useful when the investigator is interested in answering the question "Are interaction terms needed to fit the data better or provide a better description of the data?" or "Do slopes and intercept vary as a function of one of the predictors?" In this case, the interaction term is tested as a secondary priority merely as a sensitivity check to see if the slopes and intercepts vary as a function of the other variable. This is usually not the concern in ANOVA. In ANOVA we want to know the unique effect of all the main effect terms and interactions. In ANOVA there is also a clear MSW term that serves as the error term for all three methods for dealing with unequal sizes in ANOVA, whereas in many regression applications we are interested in constructing parsimonious models and want to add parameters when necessary. So the sequential approach uses a different error term when testing the regression with only main effects and a different error term in the full regression with main effects and interactions. This reflects that ANOVA and regression have slightly different goals and so that leads to different analytic strategies.

centering

A trick some have found useful is to first "center" the predictors[7] (e.g., subtract the mean from each predictor) and then create the product term. To make this more concrete, one could test this model, which is identical in form to Equation 8-13 except using centered predictors rather than raw predictors

$$W = \beta_0 + \beta_1(E - \overline{E}) + \beta_2(C - \overline{C}) + \beta_3(E - \overline{E})(C - \overline{C}) \quad (8\text{-}15)$$

It turns out that centering is an important thing to do in multiple regression when you have an interaction term. The reason is that if you don't center, then the regression

[6] When there are unequal sample sizes in an experimental design, orthogonality is compromised so, in a sense, it creates a problem where the predictors are correlated and that is why we see similar ideas in both ANOVA and regression.

[7] There is no need to center the dependent variable; you only need to center the predictors. But there is no harm in centering the dependent variable if you choose to do so.

becomes "scale dependent" in the sense that adding a constant to one of the predictors before creating the interaction can lead to different regression results on the main effects. It is only the main effect terms that "suffer" from such scale dependence. The interaction coefficient and its test are okay regardless of centering (i.e., they remain invariant regardless of any scale change to the main effects). We saw something similar in ANOVA when studying unequal sample sizes—the interaction remained invariant across the different methods. Again, the reason for centering is so that one may interpret the main effects in the presence of an interaction. If you do center, then there is no need to perform a sequential analysis. You can enter each main effect and all the interactions in one regression (as long as each main effect is centered and all interactions were created from the centered variables). This permits tests of the "unique" contribution of each main effect variable and interaction(s). Centering gives both sensible slope coefficients and sensible standard errors because centering removes some of the multicollinearity present in models that include interaction terms. The approach also allows us to think about the regression in terms of moderation on centered variables using the logic developed around Equation 8-14, where we allow slopes and intercepts to vary as a function of another predictor variable.

Typically, we center using the mean of the predictor but it is fine to center on other computed values such as the median. In repeated measures designs, we can center in several ways such as on the grand mean (mean of all data for all subjects) or we can center on the group mean (the mean of all data in a group subjects) or we can center on the subject mean (each subject's scores are centered relative to their own mean, thus creating a variable that is interpreted as deviations from each individual's own mean). Centering on subject means is helpful if we want to ignore that some subjects are high and some are low, and simply focus on fluctuations around their own mean. This is common in various literatures such as pain perception, stress, anxiety, and heart rate where we know, for example, that some people have on average higher heart rates than others, but we may be interested in studying not absolute heart rate but heart rate relative to one's own mean. A heart rate of 70bpm at rest may be a particularly high value for one person, whereas the same heart rate of 70bpm at rest may be relatively low for another person. This type of centering allows us to decouple absolute and relative patterns in data (sometimes these are called between-person effects vs within-person effects).

It is also possible to center on a particular value that may be meaningful. For example, if we are studying the precursors and consequences of the Great Recession, which started in 2007, we could center year around 2007, so -1 refers to 2006, 0 refers to 2007, 1 refers to 2008, etc. This way the intercept is more meaningful, representing the start of the Great Recession, rather than "year zero" in the Gregorian calendar for which we are greatly extrapolating our dependent variable that far back in time. Or, if we are examining impact of a surgery one year out, we may not care that some patients had surgery in 2021 and others in 2022, so we could center around the year the subject had the surgery, treating that as time 0 and then, say, count the number

of months out from surgery. This way everyone in the sample has time starting at 0, even though their surgeries may have happened in different years.

Some methodologists suggest using the sequential method on centered variables, but I like the "regression method" (aka Type III sum of squares per LN5) on centered variables instead because it separates the unique contribution of main effects from the unique contribution of interactions. This approach also has the limitation that because the main effects are not tested in the presence of an interaction (the first step in the sequential approach), then the test for whether the intercept varies as a function of another variable is under the constraint that there is a common slope. The full model with main effects and interactions provides a test of whether the intercept and slope varies as a function of the other predictor, but one is instructed not to interpret the main effects in that regression so one isn't permitted to see if there the intercept varies as a function of the other predictor in the context of different slopes.

As with ANOVA designs the concern due to correlated predictors occurs only on the main effects—the interactions are the same regardless of which method (e.g., sequential, hierarchical, regression) is used.

Recently, I've seen the suggestion that researchers should *always* run regressions with both polynomial and interaction terms. For example, if you want to include two predictors X_1 and X_2, the suggestion is that you automatically should run

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 + \epsilon \quad (8\text{-}16)$$

where X_1 and X_2 have been centered. This is the killer model that puts everything in. One then drops terms that aren't significant. My objection to this approach is that multicollinearity will kill you, unless you have 1000's of subjects to achieve small error terms. My suggestion, instead, is to run a simpler regression first with just main effects:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \quad (8\text{-}17)$$

Then do residual analysis to check whether you need additional terms (e.g., curvature in one of the predictors may suggest an X^2 term, curvature in both predictors simultaneously may suggest an interaction term). To check this you could plot residuals separately against X_1, X_2, and X_1X_2. Following this more principled approach, you will develop a parsimonious model that also accounts for the data at hand, which may not be the same model that Equation 8-16 would give because the latter is more susceptible to multicollinearity effects.

If your research calls for testing interactions between continuous variables you should read a little book by Aiken & West, *Multiple Regression: Testing and Interpreting Interactions*, published by Sage. They go into much more detail than I do here (the book is 212 pages, so the present lecture notes are quite superficial by comparison)

as well as give some interesting plots that can be used to describe 2- and 3-way interactions made up of continuous variables.

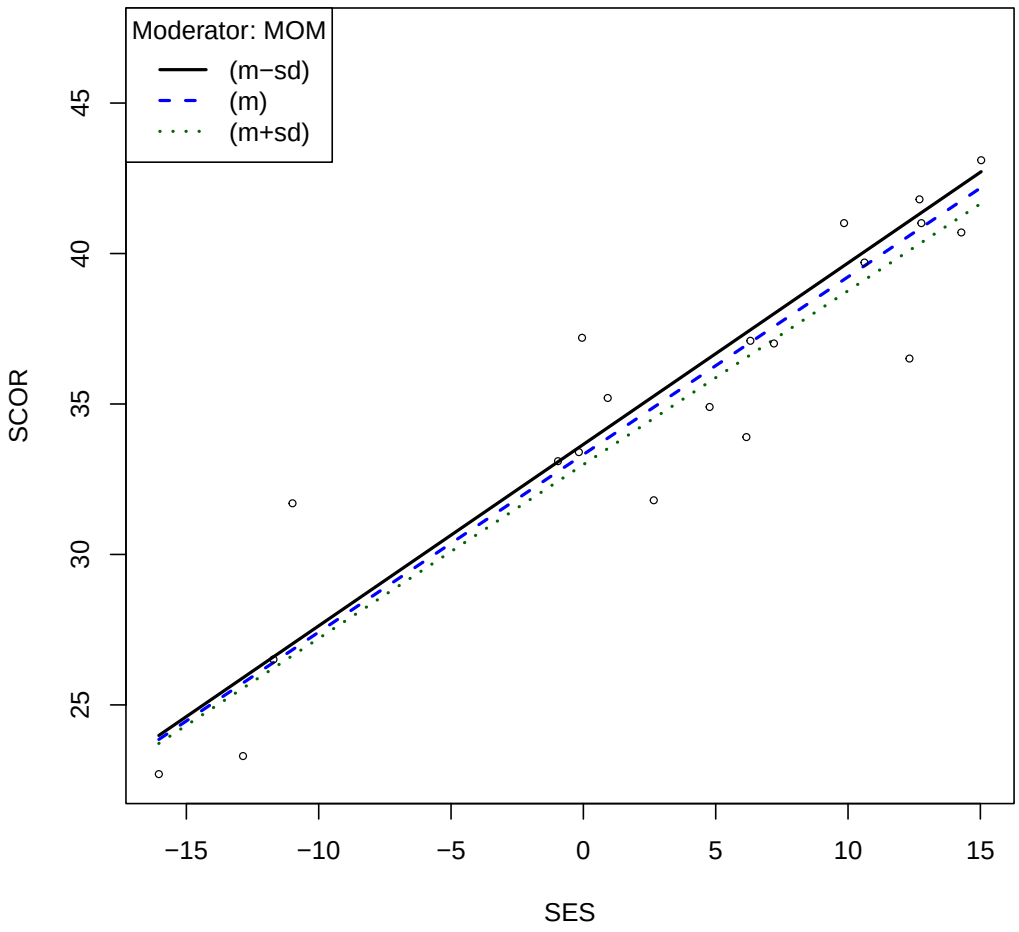
One of the Aiken & West plots that has become popular is a simplified version of the 3d plot presented in Figure 8-7. Figure 8-7 is ideal in that it depicts the raw data (points), the fitted surface (the wire mesh that represents the regression fit), and the residuals (the vertical segments between the points and the surface). But,until recently, it has difficult to draw and on a printed page not easy to rotate. So a shortcut is to plot only the wire mesh for select values of one of the predictors, which simplifies the wire mesh to lines. For example, in Figure 8-7 take the surface for X=7, and create a plot that has Z on the vertical and Y on the horizontal. The line will have a negative slope. Do that again for a couple more values of X, such as X=9 which produces a slope on the Z-Y plot that is almost 0, and say X=12, which produces a line in the Z-Y plot with a positive slope. With three such values one can represent the complicated surface relatively quickly "as the X variable goes from 7 to 12, the slope of Z on Y moves from negative, to flat, to positive." This plot though is merely a poor depiction of the model—the Z-Y plot does not present the raw data and does not present the residuals (as I showed in Figure 8-7). How to choose the values of X on which to draw particular lines in Z-Y? A standard approach is to pick three values for X: the mean of X, one standard deviation below the mean and one standard deviation above the mean. This produces a Z-Y plot with three lines. Of course, the roles of X and Y can be reversed so that one can select three values of Y, and plot three lines representing the 3d surface in a Z-X plot. Obviously, it would be much better to produce the 3d plot with points, model and residuals.

This type of plot can be depicted easily with some SPSS macros written by Hayes (see `http://www.afhayes.com/`. The rockchalk package in R produces these plots too, with many bells and whistles such as the ability to plot confidence or prediction intervals around each line, etc. Below is an example in R using the MOM and SES example from earlier in these lecture notes.

```
data <-
  read.table("dat",
             header=T)
out.lm <- lm(SCOR~ MOM * SES,data=data)
summary(out.lm)
```

```
##
## Call:
## lm(formula = SCOR ~ MOM * SES, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.4941 -1.3413 -0.0418  0.9932  4.7910
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept) 36.55262   11.15762   3.276
## MOM         -0.51620    1.89206  -0.273
## SES          0.71235    0.70554   1.010
## MOM:SES     -0.01949    0.12129  -0.161
##             Pr(>|t|)
## (Intercept)  0.00475 **
## MOM          0.78848
## SES          0.32769
## MOM:SES      0.87434
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
##
## Residual standard error: 2.355 on 16 degrees of freedom
## Multiple R-squared:  0.862,Adjusted R-squared:  0.8361
## F-statistic: 33.31 on 3 and 16 DF,  p-value: 4.12e-07


library(rockchalk)
plotSlopes(out.lm, plotx="SES", modx="MOM",
           modxVals="std.dev.")
```



Just as we discussed in ANOVA, nonparallel lines suggest the presence of an interaction. For these data, the three lines formed by setting MOM's score to take on three different values (mean, mean + sd, mean − sd), plugging each of those values into the regression for MOM, then plotting lines representing the relation between the dependent variable SCOR and the predictor SES where each of those three lines can have a separate slope and intercept governed by the chosen values for the predictor MOM. This follows the same logic I introduced for interpreting interactions in Equation 8-14.

The β values in the regresion that includes main effects and interactions (such as Equation 8-14) can be used to find the point where the nonparallel lines intersect. The value on the horizontal axis is

$$\frac{-\beta_2}{\beta_3}$$

and the value on the horizontal axis is

$$\beta_0 - \frac{\beta_1 \beta_2}{\beta_3}$$

where β_1 refers to the variable along the horizontal, β_2 refers refers to the variable that specific values are selected to produce different lines, and β_3 refers to the interaction of those two variables. If you switch the role of the two variables in the plot (i.e., which one is used along the horizontal axis and which one is fixed at different values), then switch the roles of β_2 and β_3 in the definitions below. So, to illustrate I'll build the plot manually rather than use the rockchalk package.

```
plot(data$SES, data$SCOR, xlim=c(-50,20),
     ylim=c(0, 45))
beta <- coef(out.lm)
beta
```

```
## (Intercept)         MOM         SES
## 36.55261945 -0.51619914  0.71234566
##     MOM:SES
## -0.01949183
```

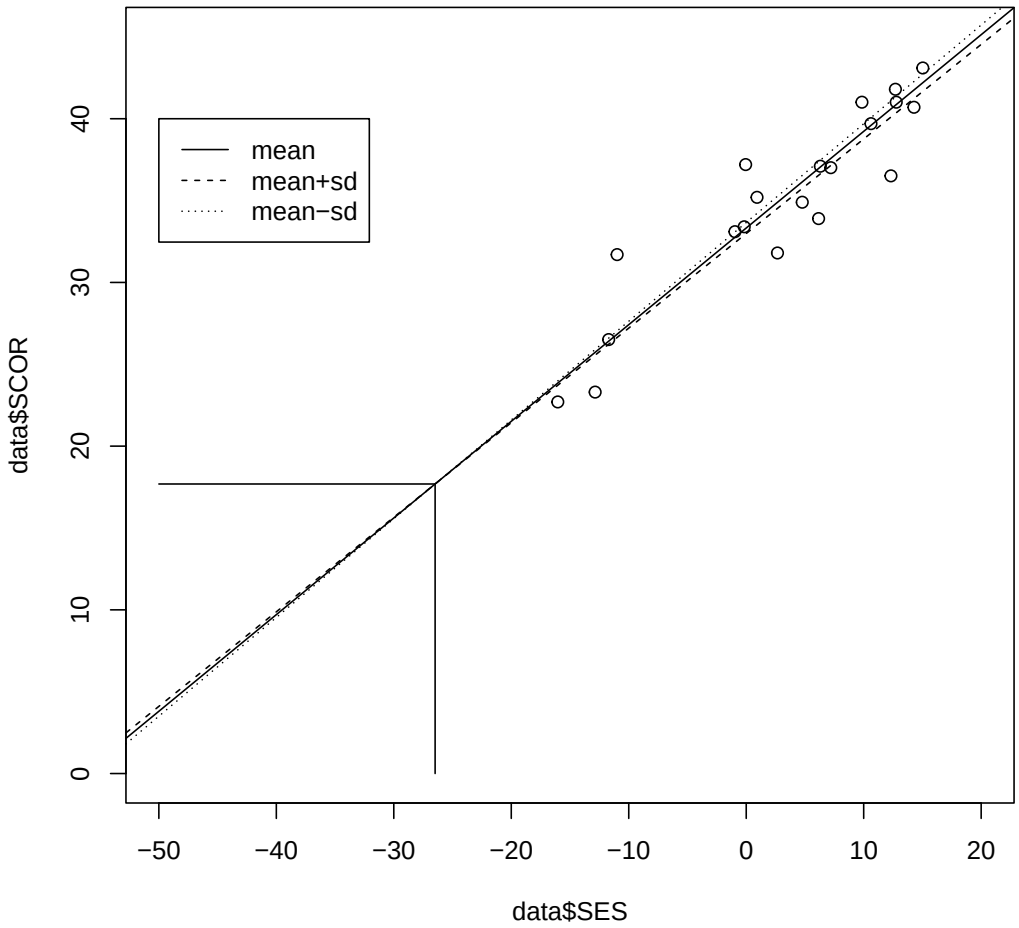
```
mean.mom <- mean(data$MOM)
sd.mom <- sqrt(var(data$MOM))

abline(beta[1] + beta[2]*mean.mom,
       beta[3] + beta[4]*mean.mom)
abline(beta[1] + beta[2]*(mean.mom+sd.mom),
       beta[3] + beta[4]*(mean.mom+sd.mom),lty=2)
abline(beta[1] + beta[2]*(mean.mom-sd.mom),
       beta[3] + beta[4]*(mean.mom-sd.mom),lty=3)
legend(-50,40,legend=c("mean","mean+sd","mean-sd"),lty=1:3)

#draw two line segments showing intersection point of all lines
#fixing mom to three different values,
#note role of beta2 and beta3 are switched from the formulas in the text
#because variables are in different order
ypt <- beta[1] -beta[3]*beta[2]/beta[4]
xpt <- -1*beta[2]/beta[4]
xpt
```

```
##       MOM
## -26.48285
```

```
ypt
```

```
## (Intercept)
##    17.68768
```

```
segments(xpt,ypt,xpt,0)
segments(xpt,ypt,-50,ypt)
```



The point that these lines all intersect is -26.48 on the horizontal axis and 17.69 on the vertical axis. Note that I had to extend the axes of the plot to the left beyond

where there are data in order to display this intersection point. This can lead to misleading conclusions because the nonparallel lines suggest the patterns reverse, but the region of reversal is outside where data were observed (that is why I also included the observed points in this plot, but, unfortunately, this is not a common practice). The actual data region involves a "fanning out" pattern rather than a reversal pattern. Of course, there may exist data in the reversal region but that is outside the scope of the present data set. As you can see, these plots are not difficult to build up yourself in R by using basic plotting commands and overlaying plot features such as lines for each value of the moderator.

8. McClelland & Judd observations on sampling and regression

An article by McClelland & Judd (1993, *Psychological Bulletin, 114,* 376-90) makes an interesting observation about the difficulties of finding interactions created from multiplying continuous predictors.[8] I encourage you to read this article if in your research you examine interactions of continuous predictors. The basic argument is that an experimental design guarantees that there will be observations in all possible cells defined by the design. However, suppose you had set up a 2x2 factorial design and there were no subjects in two of the cells—it would be impossible to extract an interaction term from such an result. This is precisely what can happen in a field study. For example, weight might be related to exercise times caloric intake, but it might be difficult to find subjects at all levels of exercise and caloric intake. McClelland and Judd present their observations in the context of why it is relatively easy to find interactions in the lab but relatively rare to find interactions in field studies. Figure 8-8 shows that statistical power also depends on the sampling the entire space (recall that unequal sample size was one contributor to multicollinearity issues, in particular when computing main effects and the row or cell means are biased by meaningful the unequal sample sizes).

I'll give an example using trees. Suppose a researcher went out to study the volume of trees and used length and width of the trunk as a simple measure. The researcher runs a multiple regression using length, width and length times width as three predictors and the measure of volume as the dependent variable. Nature tends not to have very tall thin trees (they tend to blow over in the wind, though there's bamboo, technically a grass) and very short wide trees (though take a trip to Madagascar and check out the baobab—see Fig 8-9). So the researcher playing with regression will get strange results when examining the relation between volume, length and width, and may find that the interaction term is not significant simply because of very few observations in some cells.

9. ANOVA and multiple regression.

[8]This kind of interaction differs from what we saw in the context of ANOVA where we used orthogonal codes.

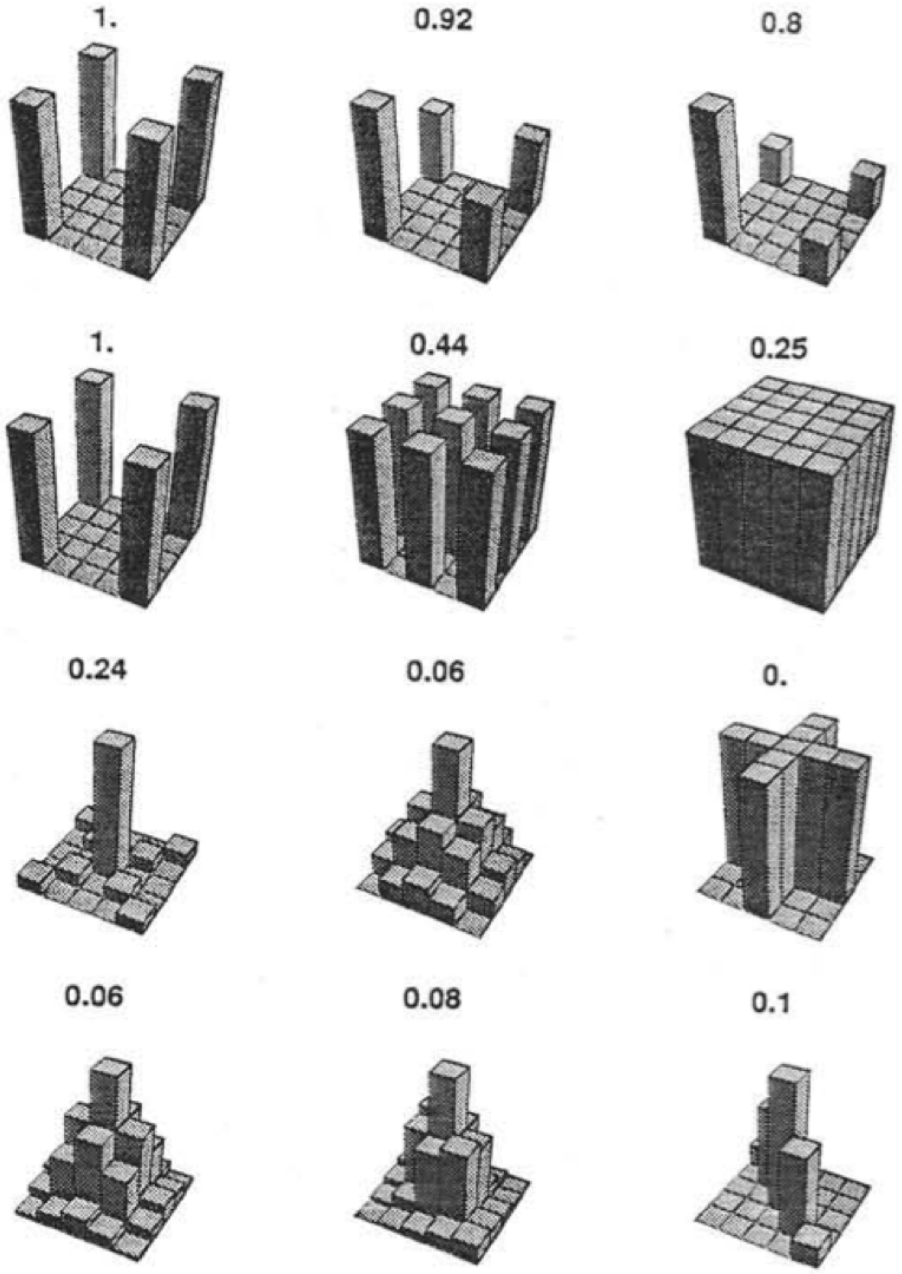


Figure 8-8: McClelland and Judd, 1993, power and interactions; number above figure is the power efficiency relative to the upper left corner, which has 1.

Figure 8-9: Bamboo (tall and narrow trunk) on the left and baobab (relatively short and wide trunk) on the right.

ANOVA is a special case of regression. We saw a glimpse of this relation earlier in Lecture Notes #7 when I showed how a simple regression analysis yields the same results as the two sample t test comparing the difference between two independent means.

The trick to getting this equivalence across a more general class of ANOVA problems is to use predictor variables that code the cells in the ANOVA. I like doing that with contrasts but you can use effect coding or dummy codes as long as you are careful.

Suppose you have a one-way ANOVA with four groups. Could you have one predictor variable with four levels (e.g., groups codes of 1, 2, 3, 4)? Will this give the same result as the one-way ANOVA? Hint: think about the degrees of freedom and how one would do contrasts. How many orthogonal contrasts are needed with four groups? It turns out that there should be as many predictors as there are contrasts. A factor with 4 levels requires three orthogonal contrasts so the regression analogue of ANOVA should also have three predictors to achieve the same degrees of freedom and decomposition of sum of squares.

Recall the two-sample t test example using regression in Lecture Notes #7. There we saw several different codings that all gave the same t and p values for the slope, which was equivalent to the test of the difference of two means. One coding was 0's for one group and 1's for the other group. This is called *dummy coding.* I verified that the variable of 0's and 1's gave the same t value as the two sample t test. Further, the slope of the regression, $\hat{\beta}_1$, was equal to the difference between the two means and the intercept was equal to the mean of the group that was coded with 0's. A different coding I used to make another point had 1's for one group and -1's for the other group. This is called *effect coding.* When the variable of 1's and -1's was used as a predictor variable we saw that the test for the slope was also identical to the t test from the two sample means test. Further, with "effects coding" the slope of the

regression is identical to the treatment effect, $\hat{\alpha}$, in the ANOVA structural model and the intercept is equal to grand mean, $\hat{\mu}$. The dummy code version defines one group as the reference (the group that receives all 0s) and the beta for a particular dummy code is the difference between each cell mean and the reference group mean. Both "dummy coding" and "effects coding" yield the identical omnibus F test and sum of squares for both between and within.

Sometimes "dummy codes" are easier to create, sometimes using "effect coding" is handy because you get the parameter estimates from the ANOVA structural model $\hat{\mu}, \hat{\alpha}, \hat{\beta}, \hat{\alpha\beta},$, etc. automatically. I prefer a third type of coding, contrast coding, because it comes in handy when creating interactions. I'll show this later—the advantage is that contrast coding preserves orthogonality when you multiply predictors to create the interaction and they are already centered. If you center dummy codes they are automatically converted to effect codes (e.g., in two groups with equal sample sizes, the dummy codes 0 and 1 when centered become -.5 and .5, respectively).

All these points generalize to the factorial ANOVA. That is, both dummy coding and effect coding can be used to get regression to give the identical results as ANOVA. The motivation for doing ANOVA-type designs through regression is that you can add all the additional machinery you have available in regression. For example, you can perform residual analysis to examine the fit of the model, the residual analysis can point to assumption violations, you can check if other variables should be added to the structural model, you can check for outliers using Cook's d, etc. You can also do new types of tests that are not possible in ANOVA, as we will see in Lecture Notes #9.

10. Coding the Main Effect in a One-Way ANOVA.

The first thing to remember is that the degrees of freedom for the numerator in an F test is T - 1, where T is the number of groups. In regression analysis each predictor variable has one degree of freedom. Therefore, we will need T - 1 separate predictor variables in a regression to get the same results as the main effect in an ANOVA. For example, if there are 4 groups, there must be three dummy codes or three effect codes as predictors. **The most common mistake people make is to have one predictor variable that takes on values 1, 2, 3, and 4.** One predictor variable with four levels will not yield the same result as the one-way ANOVA. Rather, the single predictor with four levels (1-4) will tell you the linear relation between the dependent variable and the values 1 to 4. That is, the slope is the change in the dependent variable in moving from a value of 1 to a value of 2 (and is the same change for any increment of 1, as in moving from a value of 3 to a value of 4). A single predictor with the codes 1-4 is not testing the difference between the four treatment means. If the goal is to compare four treatment means, then you need three predictors.

11. Example: one-way design analyzed with regression techniques.

These data come from Neter, Wasserman, and Kutner (1985, p. 559). An economist compiled data on productivity improvements for a sample of firms producing electronic computing equipment. The firms were classified according to the level of their average expenditures for research and development in the past three years (low, moderate, and high). The results of the study follow (productivity improvement is measured on a scale from 0 to 100). The first column is the standard "grouping variable" (what we used back in the ANOVA part of the course). The second column is the dependent variable. Columns three and four are the dummy codes we need for the regression analysis. Columns five and six illustrate "effects coding". The only difference between dummy codes and effects codes is in the value the "reference group" receives. That is, in both methods one group is the "reference group" (in this example group 3 was arbitrarily chosen as the reference group)–in dummy codes the reference group receives a value of 0 for all predictors and in effects codes the reference group receives a value of -1 for all predictors. Columns seven and eight we'll talk about later; they are contrast codes. In real analyses you wouldn't need to enter all three types of codes (dummy, effects and contrast); I put all three types of codings in this data set so that I can compare them to each other.

```
1 7.6  1 0    1  0    1  1
1 8.2  1 0    1  0    1  1
1 6.8  1 0    1  0    1  1
1 5.8  1 0    1  0    1  1
1 6.9  1 0    1  0    1  1
1 6.6  1 0    1  0    1  1
1 6.3  1 0    1  0    1  1
1 7.7  1 0    1  0    1  1
1 6.0  1 0    1  0    1  1
2 6.7  0 1    0  1   -1  1
2 8.1  0 1    0  1   -1  1
2 9.4  0 1    0  1   -1  1
2 8.6  0 1    0  1   -1  1
2 7.8  0 1    0  1   -1  1
2 7.7  0 1    0  1   -1  1
2 8.9  0 1    0  1   -1  1
2 7.9  0 1    0  1   -1  1
2 8.3  0 1    0  1   -1  1
2 8.7  0 1    0  1   -1  1
2 7.1  0 1    0  1   -1  1
2 8.4  0 1    0  1   -1  1
3 8.5  0 0   -1 -1    0 -2
3 9.7  0 0   -1 -1    0 -2
3 10.1 0 0   -1 -1    0 -2
3 7.8  0 0   -1 -1    0 -2
3 9.6  0 0   -1 -1    0 -2
3 9.5  0 0   -1 -1    0 -2
```

First, I run a regression analysis on these data using the dummy code predictors.

```
regression variables= dv dummy1 dummy2
```

```
     /statistics = r anov coeff ci
     /dependent=dv
     /method = enter dummy1 dummy2
     /resid=defaults sepred
     /casewise=defaults sepred  cook all
     /save resid(resid)  pred(fits).
```

```
Multiple R           .75307          Analysis of Variance
R Square             .56711                           DF      Sum of Squares      Mean Square
Adjusted R Square    .53103          Regression        2           20.12519         10.06259
Standard Error       .80006          Residual         24           15.36222           .64009

                                     F =      15.72053       Signif F =  .0000


Variable              B        SE B     95% Confdnce Intrvl B       Beta         T  Sig T

DUMMY2          -1.066667     .400029    -1.892286     -.241048   -.462323    -2.666  .0135
DUMMY1          -2.322222     .421668    -3.192501    -1.451943   -.954866    -5.507  .0000
(Constant)       9.200000     .326622     8.525885     9.874115               28.167  .0000
```

Compare the above output with the ONEWAY command. The source table and the omnibus F test are identical.

```
  oneway dv by grpcodes(1,3)
   /statistics  descriptives

                                ANALYSIS OF VARIANCE
                                SUM OF         MEAN            F      F
        SOURCE           D.F.  SQUARES       SQUARES         RATIO  PROB.

BETWEEN GROUPS              2       20.1252       10.0626      15.7205  .0000
WITHIN GROUPS              24       15.3622         .6401
TOTAL                      26       35.4874

                           STANDARD STANDARD
GROUP   COUNT      MEAN  DEVIATION    ERROR   MINIMUM   MAXIMUM  95 PCT CONF INT FOR MEAN

Grp 1       9    6.8778      .8136    .2712    5.8000    8.2000    6.2524 TO   7.5032
Grp 2      12    8.1333      .7572    .2186    6.7000    9.4000    7.6522 TO   8.6144
Grp 3       6    9.2000      .8672    .3540    7.8000   10.1000    8.2900 TO  10.1100

TOTAL      27    7.9519    1.1683    .2248    5.8000   10.1000    7.4897 TO   8.4140
```

The dummy codes made group 3 the reference group. Check how the intercept equals the mean for group 3; the two slopes are (1) the difference between group 1 and group 3 and (2) the difference between group 2 and group 3.

Next look at residual analysis from the regression.

RESIDUALS

Case #	DV	*PRED	*RESID	*COOK D	*SEPRED
1	7.60	6.8778	.7222	.0382	.2667
2	8.20	6.8778	1.3222	.1280	.2667
3	6.80	6.8778	-.0778	.0004	.2667
4	5.80	6.8778	-1.0778	.0851	.2667
5	6.90	6.8778	.0222	.0000	.2667
6	6.60	6.8778	-.2778	.0057	.2667
7	6.30	6.8778	-.5778	.0244	.2667
8	7.70	6.8778	.8222	.0495	.2667
9	6.00	6.8778	-.8778	.0564	.2667
10	6.70	8.1333	-1.4333	.1061	.2310
11	8.10	8.1333	-.0333	.0001	.2310
12	9.40	8.1333	1.2667	.0829	.2310
13	8.60	8.1333	.4667	.0112	.2310
14	7.80	8.1333	-.3333	.0057	.2310
15	7.70	8.1333	-.4333	.0097	.2310
16	8.90	8.1333	.7667	.0304	.2310
17	7.90	8.1333	-.2333	.0028	.2310
18	8.30	8.1333	.1667	.0014	.2310
19	8.70	8.1333	.5667	.0166	.2310
20	7.10	8.1333	-1.0333	.0551	.2310
21	8.40	8.1333	.2667	.0037	.2310
22	8.50	9.2000	-.7000	.0612	.3266
23	9.70	9.2000	.5000	.0312	.3266
24	10.10	9.2000	.9000	.1012	.3266
25	7.80	9.2000	-1.4000	.2450	.3266
26	9.60	9.2000	.4000	.0200	.3266
27	9.50	9.2000	.3000	.0112	.3266
Case #	DV	*PRED	*RESID	*COOK D	*SEPRED

Let me show you how we can throw some of the new concepts we learned in regression to ANOVA problems. Here I'll do some residual analysis. It turns out that cases 1, 6, 7, 9, 14, 15, 17, 18, 22, and 25 are all US firms. Almost all have negative residuals. This suggests that country might be an important blocking variable to include in a subsequent model because there is a pattern in the residuals associated with country.

Next, let's examine how to do contrast coding on the same data. Here is the ONEWAY output with a pair of orthogonal contrasts.

CONTRAST COEFFICIENT MATRIX

	Grp 1	Grp 2	Grp 3
CONTRAST 1	1.0	-1.0	0.0
CONTRAST 2	1.0	1.0	-2.0

		POOLED VARIANCE ESTIMATE				SEPARATE VARIANCE ESTIMATE			
	VALUE	S. ERROR	T VALUE	D.F.	T PROB.	S. ERROR	T VALUE	D.F.	T PROB.
CONT 1	-1.2556	0.3528	-3.559	24.0	0.002	0.3483	-3.605	16.7	0.002
CONT 2	-3.3889	0.7424	-4.565	24.0	0.000	0.7891	-4.295	7.6	0.003

The same tests can be done in regression using contrast coding (columns seven and eight in the data matrix). We get the same overall F test as we saw with the dummy coding and with the oneway command. The t-tests for the slopes are the same as the t-tests for the contrasts in the ONEWAY command.

```
  regression variables= dv cont1 cont2
    /statistics = r anov coeff ci
    /dependent=dv
    /method = enter cont1 cont2.

Multiple R          .75307       Analysis of Variance
R Square            .56711                        DF    Sum of Squares    Mean Square
Adjusted R Square   .53103       Regression        2          20.12519       10.06259
Standard Error      .80006       Residual         24          15.36222         .64009

                                      F =      15.72053       Signif F =  .0000


Variable             B       SE B    95% Confdnce Intrvl B       Beta         T  Sig T

CONT2          -.564815    .123737    -.820195     -.309434   -.614460    -4.565  .0001
CONT1          -.627778    .176396    -.991842     -.263714   -.479076    -3.559  .0016
(Constant)     8.070370    .160258    7.739614     8.401127                50.359  .0000
```

The numerical values for each of the two slopes are not equal to the contrast values (the "I hats") from the ONEWAY ANOVA command presented earlier. The t-tests though are equivalent so the same statistical decisions will occur in both the regression version and the ONEWAY ANOVA version. To understand the values of the slopes just look at the entire regression. Group 1 has a 1 for the first predictor and a 1 for the second predictor, so the predicted score will be 8.07037 - .627778 - .564815 = 6.877777, the mean for group 1 (the sum of the intercept plus 1 times each of the two slopes). Group 2 has a -1 for the first predictor and a 1 for the second predictor, so the predicted score will be 8.07037 + .627778 - .564815 = 8.133333, the mean for group 2. Group 3 has a 0 for the first predictor and a -2 for the second predictor, so the predicted score will be 8.07037 + 2 * .564815 = 9.2, the mean for group 3. So, the overall regression codes all three group means (even though they have unequal sample sizes). The slopes are appropriately scaled so everything adds up as it should, and that is why the slopes are not equal to the I hats from the contrasts but the t-tests are the same.

In sum, this example shows 1) the equivalence between ANOVA and regression with dummy codes and 2) the equivalence between contrasts in ANOVA and regression slopes with contrast codes. Of course, the t-tests for the dummy code version are not equal to the t-tests for the contrast code version because the dummy codes are testing different contrasts (group 1 vs group3 and group 2 vs group 3) than the particular orthogonal contrasts I used in the example (group 1 vs group 2 and the average of groups 1 and 2 vs group 3).

R example

The same example conducted in R. I'll read in data and label columns, then run

the regression, print the output and print Cook's D. The results are the same as the regression and ANOVA results presented earlier. The results for the effect coding and contrast coding versions are identical to what I showed earlier for SPSS.

```
oneway.data <- read.table("oneway-data.txt", header = F)
colnames(oneway.data) <- c("group", "dv", "d1", "d2", "ec1",
    "ec2", "c1", "c2")
# print first few rows to see structure
head(oneway.data)
```

```
##   group  dv d1 d2 ec1 ec2 c1 c2
## 1     1 7.6  1  0   1   0  1  1
## 2     1 8.2  1  0   1   0  1  1
## 3     1 6.8  1  0   1   0  1  1
## 4     1 5.8  1  0   1   0  1  1
## 5     1 6.9  1  0   1   0  1  1
## 6     1 6.6  1  0   1   0  1  1
```

```
out.lm <- lm(dv ~ d1 + d2, oneway.data)
summary(out.lm)
```

```
##
## Call:
## lm(formula = dv ~ d1 + d2, data = oneway.data)
##
## Residuals:
##      Min       1Q   Median       3Q
## -1.43333 -0.50556  0.02222  0.53333
##      Max
##  1.32222
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept)   9.2000     0.3266  28.167
## d1           -2.3222     0.4217  -5.507
## d2           -1.0667     0.4000  -2.666
##             Pr(>|t|)
## (Intercept)  < 2e-16 ***
## d1          1.16e-05 ***
## d2            0.0135 *
## ---
```

```
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
##
## Residual standard error: 0.8001 on 24 degrees of freedom
## Multiple R-squared:  0.5671,Adjusted R-squared:  0.531
## F-statistic: 15.72 on 2 and 24 DF,  p-value: 4.331e-05
```

```
round(cooks.distance(out.lm), 3)
```

```
##     1     2     3     4     5     6
## 0.038 0.128 0.000 0.085 0.000 0.006
##     7     8     9    10    11    12
## 0.024 0.050 0.056 0.106 0.000 0.083
##    13    14    15    16    17    18
## 0.011 0.006 0.010 0.030 0.003 0.001
##    19    20    21    22    23    24
## 0.017 0.055 0.004 0.061 0.031 0.101
##    25    26    27
## 0.245 0.020 0.011
```

12. Coding Main Effects in a Factorial ANOVA.

    Main effects in a factorial ANOVA are tested in a similar way as when there is only one independent variable. Each factor is coded separately. For example, if there are two factors each with three levels, then there are two sets of predictors each with two contrasts or codes for a total of four main effect predictors in the structural model; two from one factor and two from the second factor. One set codes one main effect, the second set codes the other main effect. Any number of main effects are coded the same way. The regression structural model includes all the codes for each of the main effects.

    But, we also have to deal with the interaction(s).

13. Coding Interactions of Discrete Predictors (aka Factors).

    The coding of interactions doesn't require much thought if you do *effects coding* or *contrast coding*. You simply multiply the contrast codes that have already been created for the main effects, and automatically you have the interaction terms. On the

other hand, if you used dummy codes, then multiplication of the main effects predictors will not necessarily produce the decomposition you anticipated. This is a major misunderstanding that his wide-spread. Much of the published literature continues to use dummy codes incorrectly.

Let's look at an example. Contrast coding of a 2 × 2 factorial design might look like this, where the interaction is simply the product of the main effects. In the case of a 2 × 2 design, each subject gets three different codes (one for main effect A, one for main effect B, and one for the interaction). I'll use contrast codes in the example.

group	main effect A	main effect B	interaction
1	1	1	1
2	1	-1	-1
3	-1	1	-1
4	-1	-1	1

For the special case of two levels for each factor, contrast coding is identical to effects coding.

Here is the old biofeedback example we saw in Lecture Notes #4. First, I'll analyze these data with the ONEWAY command and contrasts, then I'll do the regression command using the contrasts as predictors. The results are identical for both analyses. The first column are the data, the second column are the grouping codes 1-4, and the last three columns are the three contrasts we want to test.

```
  data list free / dv grp me1 me2 int.

begin data
158    1  1  1  1
163    1  1  1  1
173    1  1  1  1
178    1  1  1  1
168    1  1  1  1
188    2  1 -1 -1
183    2  1 -1 -1
198    2  1 -1 -1
178    2  1 -1 -1
193    2  1 -1 -1
186    3 -1  1 -1
191    3 -1  1 -1
196    3 -1  1 -1
181    3 -1  1 -1
176    3 -1  1 -1
185    4 -1 -1  1
190    4 -1 -1  1
195    4 -1 -1  1
200    4 -1 -1  1
180    4 -1 -1  1
```

```
end data.

oneway dv by grp(1,4)
 /contrast 1, 1, -1, -1
 /contrast 1 -1 1 -1
 /contrast 1 -1 -1 1.

regression variables= dv me1 me2 int
     /statistics = r anov coeff ci
     /dependent=dv
     /method = enter me1 me2 int.
```

ONEWAY OUTPUT

Source	D.F.	Sum of Squares	Mean Squares	F Ratio	F Prob.
Between Groups	3	1540.0000	513.3333	8.2133	.0016
Within Groups	16	1000.0000	62.5000		
Total	19	2540.0000			

Contrast Coefficient Matrix

	Grp 1	Grp 2	Grp 3	Grp 4
Contrast 1	1.0	1.0	-1.0	-1.0
Contrast 2	1.0	-1.0	1.0	-1.0
Contrast 3	1.0	-1.0	-1.0	1.0

	Pooled Variance Estimate				
	Value	S. Error	T Value	D.F.	T Prob.
Contrast 1	-20.0000	7.0711	-2.828	16.0	.012
Contrast 2	-24.0000	7.0711	-3.394	16.0	.004
Contrast 3	-16.0000	7.0711	-2.263	16.0	.038

REGRESSION OUTPUT

Multiple R	.77865
R Square	.60630
Adjusted R Square	.53248
Standard Error	7.90569

Analysis of Variance

	DF	Sum of Squares	Mean Square
Regression	3	1540.00000	513.33333
Residual	16	1000.00000	62.50000

F = 8.21333 Signif F = .0016

Variable	B	SE B	95% Confdnce Intrvl B		Beta	T	Sig T
ME1	-5.000000	1.767767	-8.747498	-1.252502	-.443678	-2.828	.0121
ME2	-6.000000	1.767767	-9.747498	-2.252502	-.532414	-3.394	.0037
INT	-4.000000	1.767767	-7.747498	-.252502	-.354943	-2.263	.0379
(Constant)	183.000000	1.767767	179.252502	186.747498		103.520	.0000

However, when you use dummy codes you get a strange effect when you multiply the main effects to get the interaction term:

group	main effect A	main effect B	interaction
1	1	1	1
2	1	0	0
3	0	1	0
4	0	0	0

The entry in the fourth row/fourth column is 0 but it should be a 1 to code the correct interaction term. The mathematics of regression are such that the total R^2 for the full model (all three predictors) will be correct if you enter these three predictors, but the test of the main effects will be wrong because the resulting codes are not the ones intended by the researcher. One needs to be very careful when using dummy coding. Because of their simplicity dummy codes are the most frequently used codes in regression and it makes me wonder how many incorrect main effects are reported in the literature.

Let's go back to the biofeedback example to illustrate the problem that occurs when dummy codes are multiplied to create interaction terms. Same data set as above but now coded with dummy codes. Variable dumint1 (column 5) is the "interaction" resulting from multiplying dum1 and dum2 (what many analysts do when they create interaction terms from dummy codes); variable dumint2 (column 6) is the correct interaction code replacing group 4's 0 with a 1. Columns 5 and 6 differ only in the last five rows corresponding to group 4.

```
  data list free / dv grp dum1 dum2 dumint1 dumint2 .

begin data
158    1 1 1 1 1
163    1 1 1 1 1
173    1 1 1 1 1
178    1 1 1 1 1
168    1 1 1 1 1
188    2 1 0 0 0
183    2 1 0 0 0
198    2 1 0 0 0
178    2 1 0 0 0
193    2 1 0 0 0
186    3 0 1 0 0
191    3 0 1 0 0
196    3 0 1 0 0
181    3 0 1 0 0
176    3 0 1 0 0
185    4 0 0 0 1
190    4 0 0 0 1
195    4 0 0 0 1
200    4 0 0 0 1
180    4 0 0 0 1
end data.
```

I'll run the regression with the incorrect dummy code interaction (i.e., the variable resulting from multiplying dum1 and dum2) to illustrate. Even though the slope for the interaction term (β_3) is correct (i.e., equivalent to the interaction contrast reported above in the ONEWAY command), the two t-tests for the slopes corresponding to the main effects are not the same as in the ONEWAY output above; the following two main effects are incorrect.

```
  33  0  regression variables= dv dum1 dum2 dumint1
  34  0       /statistics = r anov coeff ci
  35  0       /dependent=dv
  36  0       /method = enter dum1 dum2 dumint1.
  37  0
```

```
Multiple R           .77865          Analysis of Variance
R Square             .60630                          DF    Sum of Squares    Mean Square
Adjusted R Square    .53248          Regression       3        1540.00000      513.33333
Standard Error      7.90569          Residual        16        1000.00000       62.50000

                                     F =       8.21333       Signif F =  .0016


Variable            B       SE B     95% Confdnce Intrvl B       Beta        T  Sig T

DUM1        -2.000000   5.000000   -12.599526     8.599526   -.088736    -.400  .6944
DUM2        -4.000000   5.000000   -14.599526     6.599526   -.177471    -.800  .4354
DUMINT1    -16.000000   7.071068   -30.989994    -1.010006   -.614779   -2.263  .0379
(Constant) 190.000000   3.535534   182.505003   197.494997              53.740  .0000
```

What happened here? Why are the main effect tests not consistent with the result of the ONEWAY command? The explanation can be seen by plugging the values of the predictors into the structural model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \tag{8-18}$$

For Group 4, all three predictors are 0, so the intercept β_0 is identical to the mean for Group 4, $\overline{Y}_4$. So, for this example, whenever we see the intercept β_0 we can interpret it as the mean for Group 4, i.e., $\overline{Y}_4$.

For Group 3, the three predictors take on the value 0, 1, and 0 and plugging these three values into the structural model leads to the prediction of the Group 3 mean

$$\begin{aligned} \overline{Y}_3 &= \beta_0 + \beta_2 \\ &= \overline{Y}_4 + \beta_2 \end{aligned}$$

So, β_2 is equal to the difference of the Group 3 mean and the Group 4 mean. In this model, the slope β_2 is NOT a main effect predictor as one would think by looking at the codes. The codes assign groups 1 and 3 the value 1, and groups 2 and 4 the value 0, yet the slope associated with that predictor tests the difference between groups 3

and 4. The predictor that on the surface looked like it was testing a main effect is actually testing something else. So, in the case of dummy codes with interactions one cannot directly infer what the predictor is testing without also looking at the other predictors in the model.

An analogous argument shows that the slope β_1 is the difference between the Group 2 mean and the Group 4 mean; it is NOT the other main effect. All is not good in the world of dummy codes; maybe there is another reason they are called dummy codes?

Can be skipped if not interested in linear algebra: A more formal explanation of what is going on emerges with a little bit of linear algebra, which will be covered in Lecture Notes 11. The set of "contrasts" that are actually used in a regression is given by the "hat" matrix, which in R would be solve(t(x)%*%x)%*%t(x) where x is the set of codes (dummy, contrast, effect, whatever) you are using along with an extra column of 1s for the intercept, %*% denotes matrix multiplication, solve computes the inverse, and t() is the transpose function. This is related to the material presented in LN3 where I showed how R handles nonorthogonal contrasts and we had to work backwards to figure out what contrasts R actually puts into the regression in order to test the desired set of contrasts.

So, as we saw above, even though you enter this set of dummy codes:

group	main effect A	main effect B	interaction
1	1	1	1
2	1	0	0
3	0	1	0
4	0	0	0

The predictors, due to multicollinearity, are converted to another matrix in the process of computing the regression. The actual predictors are these, and you can see they correspond exactly to what I wrote above (intercept corresponds to group 4, β_1 corresponds to the comparison of groups 2 and 4, β_2 corresponds to the comparison of groups 3 and 4, and β_3 corresponds to the correct interaction term):

group	intercept	predictor with β_1	predictor with β_2	predictor with β_3
1	0	0	0	1
2	0	1	0	-1
3	0	0	1	-1
4	1	-1	-1	1

However, if the correct dummy code interaction is included in the regression, the

correct main effects are tested. The correct dummy code interaction is (1, 0, 0, 1), unlike the incorrect dummy code interaction variable created by multiplying the main effect dummy codes is (1, 0, 0, 0).

```
  38  0  regression variables= dv dum1 dum2 dumint2
  39  0       /statistics = r anov coeff ci
  40  0       /dependent=dv
  41  0       /method = enter dum1 dum2 dumint2.
  42  0

OUTPUT OF DUMMY CODES USING CORRECT INTERACTION TERM. THE T-TESTS
CORRESPOND TO THE OUTPUT OF THE ONEWAY COMMAND ABOVE.

Multiple R           .77865          Analysis of Variance
R Square             .60630                          DF    Sum of Squares    Mean Square
Adjusted R Square    .53248          Regression       3        1540.00000      513.33333
Standard Error      7.90569          Residual        16        1000.00000       62.50000

                                     F =       8.21333       Signif F =  .0016


Variable              B       SE B     95% Confdnce Intrvl B       Beta         T  Sig T

DUM1          -10.000000   3.535534   -17.494997    -2.505003   -.443678    -2.828  .0121
DUM2          -12.000000   3.535534   -19.494997    -4.505003   -.532414    -3.394  .0037
DUMINT2        -8.000000   3.535534   -15.494997     -.505003   -.354943    -2.263  .0379
(Constant)   198.000000   3.535534   190.505003   205.494997                56.003  .0000
```

If you are interested in understanding why this version works, you can substitute the values of the predictors into the structural model as I did above. In this structural model and with these three predictors, the four group means are modeled as

$$\begin{aligned}
\overline{Y}_1 &= \beta_0 + \beta_1 + \beta_2 + \beta_3 \\
\overline{Y}_2 &= \beta_0 + \beta_1 \\
\overline{Y}_3 &= \beta_0 + \beta_2 \\
\overline{Y}_4 &= \beta_0 + \beta_3
\end{aligned}$$

With a little rearranging and substituting, you will find that $2\beta_0 = \overline{Y}_2 + \overline{Y}_3 + \overline{Y}_4 - \overline{Y}_1$. From this, you can further play with symbols to show, for instance, that β_1 is identical to testing the main effect contrast 1, 1, -1, -1, and β_2 is identical to testing the other main effect 1, -1, 1, -1.

The bottom line is that you should avoid dummy codes unless you are confident you know what you are doing. The mindless thing to do so that you automatically get the correct results is to center the dummy codes before you create the interaction term, and enter the centered "dummy variables" into the regression equation. Centering will give you the correct results even if you use dummy codes in the case of interactions

being present in the model. This is one reason why almost all methodologists advise to always center whenever one includes interactions and polynomial terms in a regression; if you automatically center, you will typically correct any such issues that arise. Or you could just use contrast codes from the beginning because those automatically work nicely.

This example will be identical if run in R.

14. Coding Contrasts.

There is a third type of coding (in addition to "dummy" and "effect") that is also useful to know. It is called "contrast coding", and, as the name suggests, its the way to perform contrasts in regression. Contrast coding is the natural extension of the contrasts we learned in the first part of Psych613. Suppose you had three groups and wanted to perform a one-way analysis of variance using regression. You know that two predictor variables are needed because the ANOVA has T - 1 degrees of freedom in the denominator (where T is the number of groups) so the regression equation must have T - 1 predictors.

If you want to test contrasts you will need an orthogonal set of contrasts as your predictors. Recall that there can be T - 1 contrasts in an orthogonal set. One such set consists of two contrasts: A) 1, -1, 0 and B) 1, 1, -2. These two contrasts can be converted to predictors. Predictor A assigns 1's to all subjects in the first group, -1's to all subjects in the second group, and 0's to all subjects in the third group. Predictor B assigns 1's to all subjects in the first two groups, and -2's to all subjects in the third group. This yields the regression model

$$\text{Y} = \beta_0 + \beta_1 \text{PredictorA} + \beta_2 \text{PredictorB} \tag{8-19}$$

This coding is nice because it gives you three different tests. Testing the F of the regression equation (i.e., is the R^2 different than zero, or equivalently, do the two predictors jointly account for a significant portion of the variability in Y) is equivalent to the omnibus test of the ANOVA. The test of significance for β_1 tests the contrast corresponding to Predictor A. The test of significance for β_2 tests the contrast corresponding to Predictor B. If there are an equal number of subjects in each group, then these predictors will be orthogonal, as we saw in the ANOVA section of the course. Further, the MSE from the source table can be used for post hoc tests such as Tukey and Scheffe (it is the same MSE as in the ANOVA).

The main advantage of formulating ANOVA in terms of regression is for the ease in which residual analysis may be performed (checking assumptions, using Cook's D for outlier identification, etc.). There are also some benefits in terms of covariates (or blocking factors) that we will explore in Lecture Notes #9.

Contrast codes share the same property with effect coding that in a factorial design the product of the main effect contrasts corresponds to the interaction. All you have to do to find interactions is multiply the relevant contrast codes (or effect codes) for the main effects. Recall that this trick of multiplying main effects does not work if you used dummy codes (codes of 0's and 1's). Of course, you need not force the analyses into the main effect/interaction terminology. When there are T groups you can have T - 1 orthogonal contrasts (hence requiring T - 1 orthogonal predictors in a regression), and this is true regardless of how the groups are arranged in a between-subjects experiment. This is the same as the trick we used in ANOVA designs to convert a between-subjects design into a one-way design.

centering, coding, unequal N

CAVEAT: Recall the discussion in Lecture Notes 4 about unequal N and the different types of approaches for decomposing sum of squares (i.e., regression, hierarchical, and sequential approaches), and how each approach can be interpreted as implying a different set of contrast weights. If there are unequal N across the groups, you need to be careful about centering because the unequal N will create an imbalance in the contrast, effect codes or dummy codes. There are analogs to the hierarchical method, the sequential method and the regression method to use in regression analyses, but one key difference is that in the ANOVA context the norm is to always use the same MSE term (recall how back in LN4 we manually specified the error to be equal to MSE for each of the three methods). However, in the regression world the norm is to lump the sum of squares of terms that aren't included in the model into the error term (e.g., if running the sequential method that enters all the main effects, the interaction term is not entered but its sums of squares and degrees of freedom are typically lumped into the error term). I think there is a good paper to write to make all of this clear, contrasting the norms in the ANOVA and regression literatures, explaining how the error terms can differ in these two approaches, making recommendations for when to center and when not to center, and connecting the point above that when multiplying main effects that are not defined as contrasts but, say, dummy codes, then one may get strange results even in the case of equal n if codes aren't centered prior to multiplying to create the interaction predictors. Things get more complicated in the case of mixed models with random effects where the meaning of centering can vary (see, for example, Hamaker et al, 2019, *Psychological Methods*) and there are more issues to address when selecting the appropriate error term (i.e., the expected mean square term in the denominator should have the same terms as the numerator except for the single term being tested).

15. Example: factorial design through regression

The easiest way to understand what we are doing is to think about a series of regressions where we omit particular variables at each stage. This is not the easiest way to calculate a factorial ANOVA but it does give insight into what a factorial ANOVA is doing. Thinking in terms of regression facilitates understanding because it sheds

new light on how to interpret ANOVA as well as providing the ability to analyze the residuals.

In a two-way factorial design we know that there will be two main effects and one interaction (i.e., three different F tests). So, we need to do regression in such a way to get three different F tests. A conceptually easy method is to do four different regressions. One regression, the full regression, has all contrast codes entered (main effects and interaction). A second regression includes all contrast predictors except those corresponding to the main effect of factor A. A third regression includes all contrast predictors except those corresponding to the main effect of factor B. Finally, a fourth regression includes all contrast predictors except those corresponding to the interaction of A and B. We can then compare the R^2 fits of the full regression to each of the three "reduced" regressions to test the fit of that particular main effect or interaction.

Here is an example we did previously when discussing a 2×3 factorial design. The four regressions follow.

```
DATA WITH CONTRAST CODING
column 1 and 2 are the grouping codes (included for purposes of running
 MANOVA later)
column 3 is the dependent variable
column 4/5 are the orthogonal contrast codes for the main effect for
 factor A
column 6 is the orthogonal contrast code for the main effect for factor B
column 7/8 are the orthogonal contrast codes for the interaction of A
 and B (note that column 7 is the product of cols 4 and 6; column 8 is the
 product of cols 5 and 6)

1 1 53  1  1  1     1  1
1 1 49  1  1  1     1  1
1 1 47  1  1  1     1  1
1 1 42  1  1  1     1  1
1 1 51  1  1  1     1  1
1 1 34  1  1  1     1  1
1 2 44  1  1 -1    -1 -1
1 2 48  1  1 -1    -1 -1
1 2 35  1  1 -1    -1 -1
1 2 18  1  1 -1    -1 -1
1 2 32  1  1 -1    -1 -1
1 2 27  1  1 -1    -1 -1
2 1 47 -1  1  1    -1  1
2 1 42 -1  1  1    -1  1
2 1 39 -1  1  1    -1  1
2 1 37 -1  1  1    -1  1
2 1 42 -1  1  1    -1  1
2 1 33 -1  1  1    -1  1
2 2 13 -1  1 -1     1 -1
2 2 16 -1  1 -1     1 -1
2 2 16 -1  1 -1     1 -1
2 2 10 -1  1 -1     1 -1
2 2 11 -1  1 -1     1 -1
2 2 6  -1  1 -1     1 -1
3 1 45  0 -2  1     0 -2
```

```
3 1 41  0 -2  1     0 -2
3 1 38  0 -2  1     0 -2
3 1 36  0 -2  1     0 -2
3 1 35  0 -2  1     0 -2
3 1 33  0 -2  1     0 -2
3 2 46  0 -2 -1     0  2
3 2 40  0 -2 -1     0  2
3 2 29  0 -2 -1     0  2
3 2 21  0 -2 -1     0  2
3 2 30  0 -2 -1     0  2
3 2 20  0 -2 -1     0  2

THE FULL MODEL
  regression variables= score mainA1 mainA2 mainB int1 int2
   /statistics = r anov coeff ci
   /dependent=score
   /method = enter mainA1 mainA2 mainB int1 int2.

Multiple R           .84384
R Square             .71207
Adjusted R Square    .66408
Standard Error      7.45654

Analysis of Variance
                   DF      Sum of Squares      Mean Square
Regression          5          4125.00000        825.00000
Residual           30          1668.00000         55.60000

F =      14.83813       Signif F =  .0000


Variable              B        SE B     95% Confdnce Intrvl B       Beta

INT2           2.166667     .878762      .371996     3.961338    .241550
INT1          -4.000000    1.522060    -7.108461     -.891539   -.257462
MAINB          7.833333    1.242757     5.295285    10.371381    .617513
MAINA2         -.500000     .878762    -2.294671     1.294671   -.055742
MAINA1         7.000000    1.522060     3.891539    10.108461    .450559
(Constant)    33.500000    1.242757    30.961952    36.038048

Variable           T  Sig T

INT2           2.466  .0196
INT1          -2.628  .0134
MAINB          6.303  .0000
MAINA2         -.569  .5736
MAINA1         4.599  .0001
(Constant)    26.956  .0000

OMIT MAIN EFFECT FOR A
  regression variables= score mainA1 mainA2 mainB int1 int2
    /statistics = r anov coeff ci
    /dependent=score
    /method = enter mainB int1 int2.

Multiple R           .71131
R Square             .50596
Adjusted R Square    .45964
Standard Error      9.45714

Analysis of Variance
                   DF      Sum of Squares      Mean Square
Regression          3          2931.00000        977.00000
Residual           32          2862.00000         89.43750
```

```
F =      10.92383       Signif F =  .0000


Variable              B        SE B     95% Confdnce Intrvl B       Beta

INT2           2.166667    1.114535     -.103566     4.436899    .241550
INT1          -4.000000    1.930431    -7.932158     -.067842   -.257462
MAINB          7.833333    1.576190     4.622740    11.043927    .617513
(Constant)    33.500000    1.576190    30.289406    36.710594


Variable           T  Sig T

INT2           1.944  .0607
INT1          -2.072  .0464
MAINB          4.970  .0000
(Constant)    21.254  .0000

OMIT MAIN EFFECT FOR B
  regression variables= score mainA1 mainA2 mainB int1 int2
    /statistics = r anov coeff ci
    /dependent=score
    /method = enter mainA1 mainA2 int1 int2

Multiple R           .57510
R Square             .33074
Adjusted R Square    .24439
Standard Error     11.18322

Analysis of Variance
                   DF      Sum of Squares      Mean Square
Regression          4          1916.00000        479.00000
Residual           31          3877.00000        125.06452

F =       3.83002       Signif F =  .0121


Variable              B        SE B     95% Confdnce Intrvl B       Beta

INT2           2.166667    1.317956     -.521322     4.854655    .241550
INT1          -4.000000    2.282766    -8.655733      .655733   -.257462
MAINA2         -.500000    1.317956    -3.187989     2.187989   -.055742
MAINA1         7.000000    2.282766     2.344267    11.655733    .450559
(Constant)    33.500000    1.863871    29.698610    37.301390


Variable           T  Sig T

INT2           1.644  .1103
INT1          -1.752  .0896
MAINA2         -.379  .7070
MAINA1         3.066  .0045
(Constant)    17.973  .0000


OMIT INTERACTION TERMS
  regression variables= score mainA1 mainA2 mainB int1 int2
    /statistics = r anov coeff ci
    /dependent=score
    /method = enter mainA1 mainA2 mainB.

Multiple R           .76644
R Square             .58743
```

```
Adjusted R Square    .54875
Standard Error      8.64219

Analysis of Variance
                    DF      Sum of Squares      Mean Square
Regression           3          3403.00000       1134.33333
Residual            32          2390.00000         74.68750

F =      15.18773       Signif F =  .0000


Variable              B        SE B     95% Confdnce Intrvl B       Beta

MAINB          7.833333    1.440366     4.899405    10.767262    .617513
MAINA2         -.500000    1.018492    -2.574601     1.574601   -.055742
MAINA1         7.000000    1.764080     3.406686    10.593314    .450559
(Constant)    33.500000    1.440366    30.566072    36.433928


Variable           T  Sig T

MAINB          5.438  .0000
MAINA2         -.491  .6268
MAINA1         3.968  .0004
(Constant)    23.258  .0000
```

Next, we take differences between the "full" model (all variables included) and each of the "reduced" regressions (where some variables were omitted). In this example this procedure creates three differences between R^2s. Each difference in R^2 will have an F test that corresponds to a main effect or an interaction.

A quick way to calculate the F values from the R^2 above information is (see also end of Lecture Notes 7)

$$F = \frac{\frac{R^2_{\text{full}} - R^2_{\text{reduced}}}{\text{dfred - dfful}}}{\frac{1-R^2_{\text{full}}}{\text{dfful}}} \tag{8-20}$$

This formula can be expressed as a product of the following three terms: the partial correlation squared (Equation 8-9), a factor of degrees of freedom, and a factor of $(1\text{-}R^2_{\text{reduced}})/(1\text{-}R^2_{\text{full}})$ to change the denominator of the partial correlation to that needed by the F test. You compare this observed F to the tabled F (where for the tabled F dfred-dfful is the degrees of freedom for the numerator and dfful is the degrees of freedom for the denominator).

This formula is equivalent (a few algebraic manipulations will verify this) to the formula I presented in an earlier lecture.

$$F = \frac{\frac{\text{SSE(R) - SSE(F)}}{\text{dfred - dfful}}}{\frac{\text{SSE(F)}}{\text{dfful}}} \tag{8-21}$$

where SSE(R) and SSE(F) are the sum of squares for the reduced and full models, respectively, and dfred and dfful are the degrees of freedom (for the denominator) in the reduced regression and the full regression, respectively. Eq. 8-20 and Eq. 8-21 are equivalent.

For example, the difference in R^2 between the full model and the model that omitted the main effect for A is 0.71207 - 0.50596 = 0.20611. The degrees of freedom (error) for the full and reduced model are 30 and 32, respectively. Just plug these numbers into Equation 8-20 and you see that the $F = 10.7375$.

Now the output from MANOVA just to verify the equivalence.

```
manova score by lecture(1,3) method(1,2)
  /design lecture method lecture by method.

Tests of Significance for SCORE using UNIQUE sums of squares
Source of Variation          SS      DF        MS         F  Sig of F

WITHIN CELLS            1668.00      30     55.60
LECTURE                 1194.00       2    597.00     10.74      .000
METHOD                  2209.00       1   2209.00     39.73      .000
LECTURE BY METHOD        722.00       2    361.00      6.49      .005
```

The main effect for A has the same F value in both the "increment in R^2" test and in the ANOVA output. I showed the "increment in R^2 test" for the main effect for A. You should compute the other main effect and the interaction using the change in R^2 test to double check you get the same F tests as reported in the MANOVA output.

I just showed an example of a 2 × 3 factorial ANOVA. It is possible to convert this design into a one-way ANOVA with six cells. Because there are six groups we will need five orthogonal contrasts to define the design. Any set of five orthogonal contrasts will do. The t-test corresponding to each of the slopes of the five predictors will be identical to the t test corresponding to the contrasts. The full regression in previous example gave one particular set of five orthogonal predictors.

twoway ANOVA in regression using R

Here is the beginning of the previous example done in R.

```
twoway.data <- read.table("twoway-data.txt", header=F)
colnames(twoway.data) <- c("g1", "g2", "dv","g1.c1","g1.c2",
                  "g2.c1","g1.c1.by.g2.c1","g1.c2.by.g2.c1")
#print first few rows to see structure
head(twoway.data)


##   g1 g2 dv g1.c1 g1.c2 g2.c1
```

```
## 1  1  1 53     1     1     1
## 2  1  1 49     1     1     1
## 3  1  1 47     1     1     1
## 4  1  1 42     1     1     1
## 5  1  1 51     1     1     1
## 6  1  1 34     1     1     1
##   g1.c1.by.g2.c1 g1.c2.by.g2.c1
## 1              1              1
## 2              1              1
## 3              1              1
## 4              1              1
## 5              1              1
## 6              1              1
```

```
out.twoway.lm <- lm(dv ~ g1.c1 + g1.c2 + g2.c1 +
                      g1.c1.by.g2.c1 +g1.c2.by.g2.c1,
                    twoway.data)
summary(out.twoway.lm)
```

```
##
## Call:
## lm(formula = dv ~ g1.c1 + g1.c2 + g2.c1 + g1.c1.by.g2.c1 + g1.c2.by.g2.c1,
##     data = twoway.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -16.00  -3.25  -0.50   4.00  15.00
##
## Coefficients:
##                Estimate Std. Error
## (Intercept)     33.5000     1.2428
## g1.c1            7.0000     1.5221
## g1.c2           -0.5000     0.8788
## g2.c1            7.8333     1.2428
## g1.c1.by.g2.c1  -4.0000     1.5221
## g1.c2.by.g2.c1   2.1667     0.8788
##                t value Pr(>|t|)
## (Intercept)     26.956  < 2e-16 ***
## g1.c1            4.599 7.21e-05 ***
## g1.c2           -0.569   0.5736
## g2.c1            6.303 5.99e-07 ***
## g1.c1.by.g2.c1  -2.628   0.0134 *
```

```
## g1.c2.by.g2.c1   2.466   0.0196 *
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
##
## Residual standard error: 7.457 on 30 degrees of freedom
## Multiple R-squared:  0.7121,Adjusted R-squared:  0.6641
## F-statistic: 14.84 on 5 and 30 DF,  p-value: 2.382e-07
```

The rest of the regressions where I omit different terms to compute the change in R^2 tests are identical in R.

16. Dichotomizing continuous variables

Many researchers can't resist the urge to dichotomize variables in order to simplify a multiple regression. For example, if you have three continuous predictors why not dichotomize each predictor (say by doing a median split) and run a 2x2x2 factorial ANOVA (or equivalently a regression with seven dichotomous predictors to code the 3 main effects, the 3 two-way interactions and the 3-way interaction)? One argument that people use in favor of dichotomizing continuous variables is that it is a whole lot easier to interpret a 2x2x2 design than the continuous version; it is not easy to interpret interactions directly from the betas in a regression. Some journals, such as ones in Public Health, mandate dichotomization of continuous variables so that one can report means in each "group" rather than slopes on the original continuous versions of the variables.

The traditional argument against dichotomization has been one of loss of power. There is loss of information when one dichotomizes variables, and that loss of information can translate into a drop in power. More importantly, the loss of information could be crucial in some settings where subjects are very expensive so sample sizes are relatively small or where one is very close to the threshold $p = .05$.

In a clever paper by Maxwell and Delaney (1993, *Psychological Bulletin, 113,* 181-190) we see that this power argument isn't the complete story. Sometimes there is bias in the direction that dichotomization can increase the chance of finding a result that isn't there. M&D provide counterexamples showing that when the true interaction on continuous variables is nonexistent, the interaction in the context of dichotomized variables may be statistically significant. In other words, they present an example where loss of information (what people thought would be a loss of power, a move that makes finding statistical significance harder) can actually lead to converting a nonexistent result into a significant one. Further, dichotomization may mask a

nonlinear relation (curvature) in such a way that curvature appears as an interaction between two variables (see Lubinski & Humphreys, 1990, *Psychological Bulletin*). So, dichotomization could lead to an incorrect conclusion of an interaction between two variables when really the underlying data in original continuous form show a curve rather than straight line on one of the variables. Regression (and ANOVA too since it is a special case) on artificially dichotomized variables can be very misleading.

The overall conclusion is that on both power and bias grounds you shouldn't dichotomize on a regular basis. What I usually do is try my best to interpret the regression results using continuous predictors, then dichotomize and run an ANOVA (possibly implemented as a regression) so I can compare the means. If the pattern of significant results remains the same across both analyses (dichotomized variables and non-dichotomized variables), then I'm comfortable using the means to help me interpret the original regression results, but I still report the continuous version of the analyses (the dichotomized version is just to get a clear understanding of the data pattern). If there are differences between the analysis with continuous variables and the analysis with dichotomized variables, then I would not trust the analyses with dichotomized variables. If you routinely use dichotomized variables in your research and see it in the literature that you read, I encourage you to look at the M&D paper or the paper by MacCallum et al (2002, *Psychological Methods, 7*, 19-40) that reviews these issues. You don't want to publish false results and make incorrect conclusions just because a co-author, colleague or reviewer suggested to simplify the analysis by dichotomizing those continuous variables.

17. Repeated measures analysis through regression

Recall that the structural model for a repeated measures design includes a random effects term for subject (lecture notes #5). For instance, a paired *t*-test has the structural model

$$Y = \mu + \alpha + \pi + \epsilon \tag{8-22}$$

This design is a randomized block with subject treated as a random effect playing the role of a blocking factor. There is no interaction term because there is only one subject per cell.

Equation 8-22 contains two factors: one for the treatment and one for subject. Our rule for converting ANOVAs into regressions is to create T - 1 codes for every factor (denoting the number of levels by T). So, we will need T - 1 predictors for the Time factor and N - 1 predictors for the subject factor. Interaction codes between the time factor and the subject dummy codes are not needed because there is one observation per cell. For the paired t-test the regression will contain 2N cases (each of the N people are measured twice). The observed scores are the dependent variable, two for each subject. One predictor is a code for time 1 or time 2 (e.g., time one = -1 and

time two = 1). There are also N - 1 predictors that code for subject (e.g., a code that gives 1 to subject 1's two scores and zero to everyone else, a code that gives 1 to subject 2's two scores and zero to everyone else, ..., a code that gives 1 to subject (N - 1)'s two scores and zero to everyone else). There is no interest in looking at the N - 1 predictor that correspond to subject because they are just codes for the blocking factor. The t-test on the slope for the code corresponding to time is exactly the same as the paired t-test. The degrees of freedom are correct—in the sense that you can view codes for subjects as bringing down the 2N scores to N - 1 degrees of freedom. You should check this yourself: start with 2N, subtract all the predictors including the unit vector (i.e., the intercept), and you should end up with a final degrees of freedom equal to N - 1. Since we aren't computing interactions it is fine to use dummy codes for the subject factor, but you could also use effect coding or contrast coding if you prefer.

A numerical example follows. Here is the paired t-test example we computed in Lecture Notes #5, only this time computed with the regression command. There are 10 subjects measured twice (time 1 & time 2). The dependent variable is a long column of all 20 observations. The predictors are the time contrast (1 or -1), which informs the regression which score belongs to time 1 and which belongs to time 2, and the N - 1 dummy codes (which are treated as blocking factors to soak up variance due to subject).

```
data list free /dv time d1 d2 d3 d4 d5 d6 d7 d8 d9.

begin data
12  1 1 0 0 0 0 0 0 0 0
16 -1 1 0 0 0 0 0 0 0 0
11  1 0 1 0 0 0 0 0 0 0
9  -1 0 1 0 0 0 0 0 0 0
12  1 0 0 1 0 0 0 0 0 0
15 -1 0 0 1 0 0 0 0 0 0
10  1 0 0 0 1 0 0 0 0 0
10 -1 0 0 0 1 0 0 0 0 0
10  1 0 0 0 0 1 0 0 0 0
9  -1 0 0 0 0 1 0 0 0 0
9   1 0 0 0 0 0 1 0 0 0
12 -1 0 0 0 0 0 1 0 0 0
7   1 0 0 0 0 0 0 1 0 0
9  -1 0 0 0 0 0 0 1 0 0
6   1 0 0 0 0 0 0 0 1 0
10 -1 0 0 0 0 0 0 0 1 0
4   1 0 0 0 0 0 0 0 0 1
9  -1 0 0 0 0 0 0 0 0 1
4   1 0 0 0 0 0 0 0 0 0
3  -1 0 0 0 0 0 0 0 0 0
end data.

regress variables = all
 /statistics = r anov coeff ci
 /dependent=dv
 /method = enter time d1 d2 d3 d4 d5 d6 d7 d8 d9.
```

The output is long but in this example all we care about is the t-test corresponding to the TIME predictor, which is the same as the 2.15 reported in Lecture Notes #5.

```
Multiple R           .93299
R Square             .87047
Adjusted R Square    .72655
Standard Error      1.76541

Analysis of Variance
                    DF      Sum of Squares      Mean Square
Regression          10           188.50000         18.85000
Residual             9            28.05000          3.11667

F =       6.04813       Signif F =  .0062

Variable              B        SE B     95% Confdnce Intrvl B       Beta

TIME           -.850000     .394757    -1.743003      .043003   -.258318
D1            10.500000    1.765408     6.506371    14.493629    .957296
D2             6.500000    1.765408     2.506371    10.493629    .592612
D3            10.000000    1.765408     6.006371    13.993629    .911711
D4             6.500000    1.765408     2.506371    10.493629    .592612
D5             6.000000    1.765408     2.006371     9.993629    .547027
D6             7.000000    1.765408     3.006371    10.993629    .638198
D7             4.500000    1.765408      .506371     8.493629    .410270
D8             4.500000    1.765408      .506371     8.493629    .410270
D9             3.000000    1.765408     -.993629     6.993629    .273513
(Constant)     3.500000    1.248332      .676078     6.323922


Variable           T  Sig T

TIME          -2.153  .0597
D1             5.948  .0002
D2             3.682  .0051
D3             5.664  .0003
D4             3.682  .0051
D5             3.399  .0079
D6             3.965  .0033
D7             2.549  .0312
D8             2.549  .0312
D9             1.699  .1235
(Constant)     2.804  .0206
```

This idea of coding the N - 1 subjects generalizes to any type of repeated measures design. In some literatures, this N - 1 approach is called the *fixed effect* approach to repeated measures because the N - 1 dummy codes are treated as fixed effects rather than as random effects (e.g., Hamaker et al., 2019, *Psychological Methods*).. A subtle difference though, beyond treating the subject dummy codes as fixed or random, is that the fixed effects approach models potential correlations in a better way than the random effects model (i.e., when there is a time-varying covariate/blocking factor, the fixed effects approach permits correlations across the subject-level dummy codes, which denote subject means, and the covariates, whereas the random effects approach typically assumes that there is no association between the mean of subject and the

covariates; see Hamaker et al, 2019, for a review). The regression approach though has some drawbacks, e.g., look back to the mixed models we covered in Lecture Notes #5 (MIXED in SPSS and lmer in the lme4 R package or lme in the nlme R package), where we used a regression representation to run repeated measures ANOVA. There we had to work hard to get the right output in terms of correct degrees of freedom and the correct error term so that we do not make unnecessary assumptions such as compound symmetry. A limitation of basic regression approaches to repeated measures is that they want to pool the error term given that there is only one ϵ term in the model), so we need to use special syntax to invoke the "multivariate" approach to yield multiple error terms as we saw in Lecture Notes 5. In LN12 and LN13 we will consider more general approaches to test repeated measures designs that use the appropriate error term with the multivariate approach.

paired t test through regression in R

The syntax for the paired t test conducted through regression would be the following with 10 predictors, one for the two time points and 9 dummy codes for the 10 participants.

```
data <- read.table("paired-t-reg.dat")
names(data) <- c("dv", "time", paste("d", 1:9, sep = ""))
out.paired <- lm(dv ~ time + d1 + d2 + d3 + d4 + d5 + d6 + d7 +
    d8 + d9, data)
summary(out.paired)
```

```
##
## Call:
## lm(formula = dv ~ time + d1 + d2 + d3 + d4 + d5 + d6 + d7 + d8 +
##     d9, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1.85  -1.15   0.00   1.15   1.85
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept)   3.5000     1.2483   2.804
## time         -0.8500     0.3948  -2.153
## d1           10.5000     1.7654   5.948
## d2            6.5000     1.7654   3.682
## d3           10.0000     1.7654   5.664
## d4            6.5000     1.7654   3.682
## d5            6.0000     1.7654   3.399
## d6            7.0000     1.7654   3.965
## d7            4.5000     1.7654   2.549
## d8            4.5000     1.7654   2.549
```

```
## d9            3.0000     1.7654   1.699
##             Pr(>|t|)
## (Intercept) 0.020586 *
## time        0.059723 .
## d1          0.000216 ***
## d2          0.005061 **
## d3          0.000308 ***
## d4          0.005061 **
## d5          0.007890 **
## d6          0.003279 **
## d7          0.031247 *
## d8          0.031247 *
## d9          0.123478
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
##
## Residual standard error: 1.765 on 9 degrees of freedom
## Multiple R-squared:  0.8705,Adjusted R-squared:  0.7265
## F-statistic: 6.048 on 10 and 9 DF,  p-value: 0.006164
```

The key test in this output is the slope for the time predictor. The t value is -2.15 with 9 degrees of freedom, paralleling the SPSS result. This is a lot of ink on the page for just one t test....

This approach includes dummy codes for each subject as separate predictors, but R users can equivalently define a subjects factor and let R automatically create the set of N - 1 dummy codes as in

```
data$subject <- factor(rep(1:10, each = 2))
summary(lm(dv ~ time + subject, data))
```

```
##
## Call:
## lm(formula = dv ~ time + subject, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1.85  -1.15   0.00   1.15   1.85
##
## Coefficients:
```

```
##             Estimate Std. Error t value
## (Intercept)  14.0000     1.2483  11.215
## time         -0.8500     0.3948  -2.153
## subject2     -4.0000     1.7654  -2.266
## subject3     -0.5000     1.7654  -0.283
## subject4     -4.0000     1.7654  -2.266
## subject5     -4.5000     1.7654  -2.549
## subject6     -3.5000     1.7654  -1.983
## subject7     -6.0000     1.7654  -3.399
## subject8     -6.0000     1.7654  -3.399
## subject9     -7.5000     1.7654  -4.248
## subject10   -10.5000     1.7654  -5.948
##             Pr(>|t|)
## (Intercept) 1.37e-06 ***
## time        0.059723 .
## subject2    0.049706 *
## subject3    0.783414
## subject4    0.049706 *
## subject5    0.031247 *
## subject6    0.078736 .
## subject7    0.007890 **
## subject8    0.007890 **
## subject9    0.002148 **
## subject10   0.000216 ***
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
##
## Residual standard error: 1.765 on 9 degrees of freedom
## Multiple R-squared:  0.8705,Adjusted R-squared:  0.7265
## F-statistic: 6.048 on 10 and 9 DF,  p-value: 0.006164
```

where subject is a factor with 10 levels.

There were several examples in the R appendix of LN5 using repeated measures using regression approach (e.g., gls, lme, lmer). You can tell they used the regression approach because the data had to be written in long format, meaning a subject's data appears in multiple rows of the data file. LN5 focused on mixed models, where subject effects were treated as random effects rather than the basic regression approach I illustrate here that uses the "fixed" effect appraoch. The mixed effects version of this test (analogous to LN5) leads to the identical t test for the difference in the means across time. Rather than define separate dummy codes, this approach treats subject as a

random effect and assigns each subject a unique intercept. These subject-level random intercepts are printed below. Note that they are less extreme than the corresponding slopes attached to the subject as a factor approach presented above. The mixed model approach has an additional feature where estimates are a weighted average between the grand mean and the individual-level effects where the weight corresponds to reliability. This doesn't have much effect on the test of significance, but if one wanted to predict subject means the mixed approach is more appropriate than the dummy code approach presented above. We'll go into more detail on this in Lecture Notes 12 and 13.

```
# using lmerTest because lme4 doesn't provide pvalues
library(lmerTest)
out.lmer <- lmer(dv ~ time + (1 | subject), data)
summary(out.lmer)
```

```
## Linear mixed model fit by REML.
##   t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: dv ~ time + (1 | subject)
##    Data: data
##
## REML criterion at convergence: 94
##
## Scaled residuals:
##      Min       1Q   Median       3Q
## -1.29873 -0.50435 -0.01381  0.57422
##      Max
##  1.10725
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 8.111    2.848
##  Residual             3.117    1.765
## Number of obs: 20, groups:  subject, 10
##
## Fixed effects:
##             Estimate Std. Error      df
## (Intercept)   9.3500     0.9833  9.0000
## time         -0.8500     0.3948  9.0000
##             t value Pr(>|t|)
## (Intercept)   9.508 5.43e-06 ***
## time         -2.153   0.0597 .
## ---
## Signif. codes:
```

```
##   0 '***' 0.001 '**' 0.01 '*' 0.05
##   '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## time 0.000
```

```
ranef(out.lmer)
```

```
## $subject
##    (Intercept)
## 1    3.9006033
## 2    0.5452456
## 3    3.4811836
## 4    0.5452456
## 5    0.1258259
## 6    0.9646653
## 7   -1.1324332
## 8   -1.1324332
## 9   -2.3906923
## 10  -4.9072106
##
## with conditional variances for "subject"
```

18. Regression vs. ANOVA approaches

    We see that ANOVA is a special case of regression and if you run the regression properly you can reproduce ANOVA models. Why bother with this extra knowledge? You learned many new things in the context of regression such as detecting outliers with Cook's D, concepts of partial correlation, multicollinearity, residual analysis, etc. Now you can apply these new tools to ANOVA.

    But there are other advantages of regression too. It is possible to mix and match features of both regression and ANOVA to create new types of analyses, such as using both categorical factors like in ANOVA and continuous predictors like in regression in the same structural model. This allows models where, for example, you can include continuous blocking variables in a randomized block design, or add a categorical blocking factor to a linear regression, as we'll see in Lecture Notes #9.

    Sometimes I get asked "Should I run a regression or an ANOVA?" In some sense it

doesn't matter because you can get the identical results if you do them properly. The answer to that question depends on factors such as which approach makes it easy to communicate your results. If you are merely comparing means across groups, then ANOVA would be sufficient. If you have a complex design with continuous and categorical predictors, or want to do fancy modeling with repeated measures designs or want to make use of features that are part of most regression programs like Cook's D, then use regression (or write a program that takes ANOVA output and computes something you need such as Cook's D).

The steps in general regression

19. A general perspective on performing regression and testing the general linear model

(a) Before you perform a regression analysis you need to decide which variables should be included in the equation. Ideally, this variable selection phase should be driven by theory. If you are comparing several theories, you should think about the implications the different theories have for which variables need to be included in the regression.

(b) Once the variables have been selected, then you formulate the model. Will you include any interactions? How will you deal with hypothesized nonlinearities (transform or do a polynomial regression or conduct a nonlinear regression)?

(c) Assess the model and check assumptions. This is the step where you run a statistics package such as SPSS. You are comparing the actual data to the model you derived in b above. Use residuals to check normality, the equality of variance assumption, and outliers. The residual plots may help identify whether you missed an important variable (e.g., recall the example with European and US firms) and whether the way you handled the nonlinearities worked.

(d) Make any changes to the model as suggested by residual analysis. Repeat step c.

(e) Replicate in a new sample any modifications made to the original hypothesized model. The reason this step is so important is that the p-values in step b are no longer completely correct as soon as you make changes to the model based on analyses (e.g., if you choose the change the model based on residual analysis). The best way to guard against Type I error in the case of actual data analysis where one is balancing model construction, adequacy of assumptions, etc., is to replicate the findings. If replication is too difficult in the domain, you could consider a cross-validation procedure such as setting a fraction of your data aside (holdout sample), do your model building on the remaining data and once you finalize the model, then evaluate the final model in the holdout sample to see

how well the final model's predicted scores compare to the actual scores in the holdout sample.

You fit the initial model, check the residuals, make adjustments to the model, check the residuals again, etc. This trial-and-error process usually gives you a much better understanding of your data and the relevant theories you are testing. Of course, if the initial model you tested in b does not fit the data and you found a model that does better, then you need to revise your theory. The revised theory must then be tested with a new set of data to guard against chance fit. Ideally, any new theory should make predictions that differ from the original theory and these new predictions should be tested as well on new data.

One goal of regression analysis is to find the best model that describes your data. Once you have found that model, then it can be used for different purposes such as testing hypotheses, building confidence intervals around model parameters, making predictions of new Y scores, building confidence intervals around the new predictions, etc.

Another goal of regression analysis is to test existing theory or test competing theories. For example, if you show that a predictor variable is irrelevant (nonessential) to a dependent variable, you damage any theory that claims that variable is key in modeling the dependent variable, of course, assuming you have enough power to detect effects if they are there and that other alternative explanations such as multicollinearity aren't contributing to the lack of significance.

Appendix 1: Additional R commands

Multicollinearity

The package car has a variance inflation command vif() to help assess degree of multicollinearity. See the Kutner book for discussion of vif.

The package rockchalk has a multicollinearity diagnostic, the command mcDiagnose(). A tutorial on the package rockchalk is available at
http://cran.r-project.org/web/packages/rockchalk/vignettes/rockchalk.pdf.

Interaction Plots

The package QuantPsych has a pair of functions, sim.slopes() to compute slopes at the mean and +/- 1 standard deviation and graph.mod() to plot the slopes.

The package rockchalk has additional tools for probing interactions and doing various regression diagnostics and graphs. See the rockchalk tutorial mentioned above and the rockchalk example earlier in these lecture notes.

The package ggplot2 is great for producing plots but you'll have to build up the plot through multiple overlays. The great thing about ggplot2 is that it can be completely customized to the way you want the plot to look, but that is the feature that can make it difficult to use. The package pequod adds functionality to ggplot2 to plot simple slopes, and also provides tools for testing interactions.

Checking Assumptions in R Revisited

There are some R packages that automatically check a whole bunch of assumptions, such as normality of residuals, violations of independence, and checking for outliers all in one or two commands. A standard one is the gvlma package. I personally don't use these because they focus on tests of significance (e.g., a test of normality with p-value), provide an omnibus test across all the assumption significance tests it performs, and gives plots that are too cluttered and often useless. You know how I feel about using a test of significance to test an assumption needed to perform a test of significance. Those tests also make assumptions, so you should check those too. It just doesn't make sense to me. I can get more insight into assumptions through helpful plots myself.

To facilitate workflow I've written some functions that I typically use in my own work that check assumptions in the way I like to have them tested (e.g., a quantile plot, a plot of residuals, Cook's D). But I find it that it is too much work to maintain these functions across all the different types of ANOVA and regressions that I run (linear models, generalized

linear models, mixed models, regularized regressions, etc.) that for many problems I'm more efficient constructing specific checks of the necessary assumptions suitable to that particular problem.

Checking assumptions involves some work, sometimes more work than coding the key ANOVA/regression. And if assumptions are violated you should try one or two different remedial measures (like a transformation and, if available, a test that doesn't make the assumption), then recheck the assumptions, and finally check for robustness and sensitivity of your final conclusion as you would feel more comfortable with conclusions that were consistent across several approaches to addressing the assumption violation.