# Complete Example

# Simultaneous Multiple Regression

**Description:**

* Simultaneous multiple linear regression allows you to use several predictors (Xs) to understand a criterion (Y) at once. You use all variables at once (rather than steps) to see how each variable changes Y *while holding the other variables constant.*
* Definitions to remember:
  + b (little b) = Coefficient, this value is the unstandardized slope for your regression equation. For every one point increase in X, you will get b points increase in Y. This score will be based on the scale of the variable you are using to predict.
  + β (beta) = Coefficient, this value is the standardized slope for your regression equation. With one X/predictor beta is equal to Pearson’s r. Since beta is standardized you can use it to compare across predictors at which IV best explained your DV.
  + *R2* = the amount of variance in the DV scores that your IV/predictors account for. This number is effect size for regression equations.
  + *pr* = partial correlation, the variance from only *that IV* over the variance *not accounted for (error)*. Tells you how much variance your variance accounts for when you only look at variance that you can explain. Proportion of variance in Y not explain by other predictors.
* Two questions to answer:
  + Is my overall model significant?
  + Which of my individual predictors are significant?

**Data Set:** data 1.csv

**IV(s):**

* Books – the number of books a person reads.
* Attend – the attendance of a person in a course.

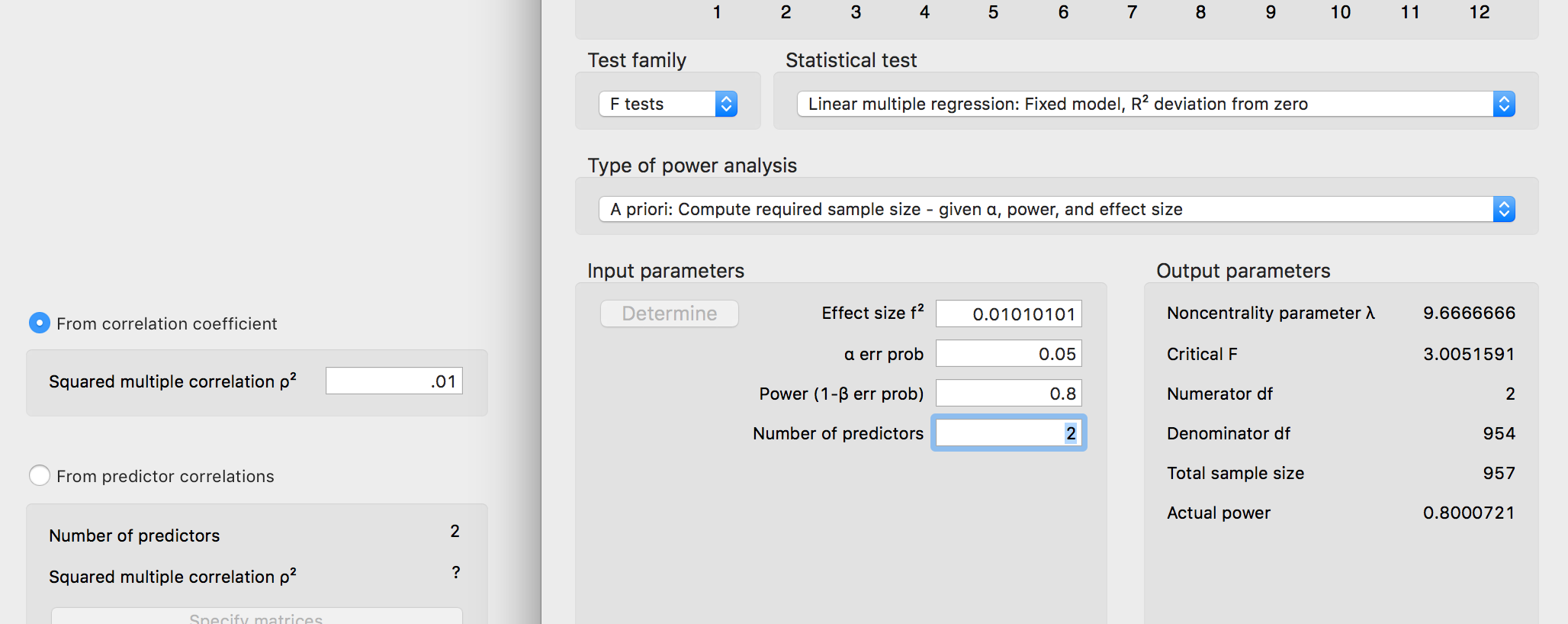
**DV:**

* Grade – final grade in the course.

**Research Question:** Do attendance and books **both** predict overall course grade?

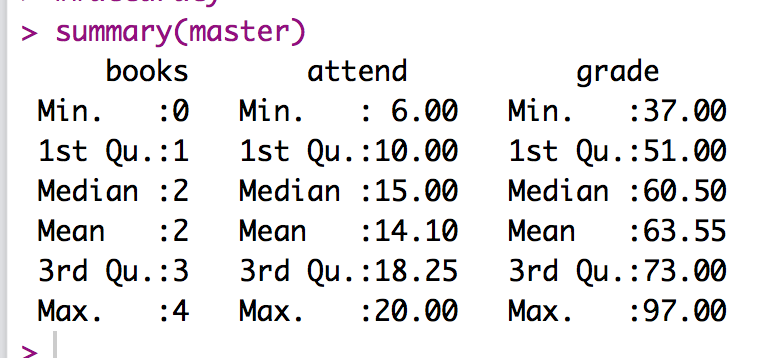
**Power:**

1. Open Gpower!
   1. Test family: F-test
   2. Statistical Test: Linear multiple regression: fixed model, R2 deviation from zero.
      1. We are using multiple regression because we have more than one predictor.
      2. R2 deviation from zero indicates that we are interested in the overall model, rather than asking if the addition of more predictors to previous model are useful.
   3. Estimate an effect size: click determine 🡪 use R square sizes you think might be accurate, remember small, medium, and large estimates from the notes.
   4. Alpha = .05
   5. Power (1-beta .20) = .80
   6. Number of predictors: number of IVs/X variables.
2. Let’s estimate the following:
   1. Small effect size (*R2* = .01)
   2. Number of predictors: 2
3. Says we needed to run 957 people to find a significant effect with a small effect size.



**Assumptions:**

1. Accuracy:
   1. Use the summary(*dataset name*) function to get the basic information for the data.
   2. Let’s check out minimum and maximum:
      1. Attendance and books cannot be negative.

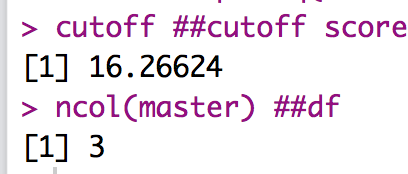


1. Missing:
   1. I can see from my summary function that I do not have missing data. Remember that you will need at least twenty variables to estimate missing data for participants – so mostly you won’t be estimating for regression.
2. Run the lm model for your data.
   1. Seems like a strange place to stop and run the analysis – but we need the regression to calculate values for outliers below. So you will run the FINAL model of your analysis for data screening.
   2. Therefore, if you are running a mediation/moderation/hierarchical be sure to run the model with ALL the variables here.
   3. output = lm(*DV* ~ *IV + IV + IV…,* data = *dataset*)
3. Outliers:
   1. First: Mahalanobis scores:
      1. mahal = mahalanobis(*dataset*,

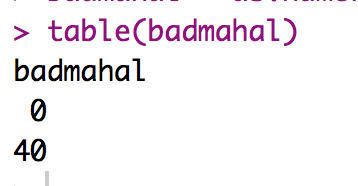
colMeans(*dataset*, na.rm = T),

cov(*dataset*, use = “pairwise.complete.obs”))

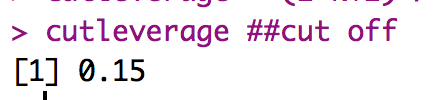
* + 1. Create the cut off score:
       1. cutoff = qchisq(1-.001, ncol(*dataset*))
    2. Remember you can use:
       1. cutoff to get the cutoff score
       2. ncol(*dataset*) to get the *df*



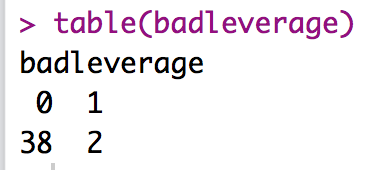
* 1. SAVE the scores:
     1. mahalout = as.numeric(mahal > cutoff) – notice that we have used > …
     2. We are checking if people are greater than the cutoff (that’s bad), and if so, giving them a 1 to mark they are an outlier. The as.numeric changes the TRUE for outlier to 1, while FALSE no outlier is a 0.
     3. This procedure is slightly different than before, because we are not simply going to keep people who are less than the cut off score – we need to keep a total of their bad scores.
     4. Check out the number of outliers (1 is bad!):
        1. table(badmahal)



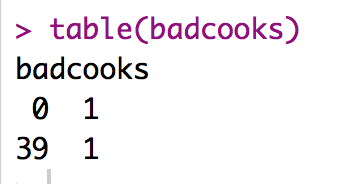
* + - 1. No outliers!
  1. Leverage scores:
     1. Remember that leverage is the influence of a single person over the slope.
     2. k = number of predictors.
     3. To get leverage values:
        1. leverage = hatvalues(output)
     4. To get the cut off score:
        1. (2\*k+2)/N
        2. cutleverage = (2\*k+2) / nrow(*dataset*)
     5. Run cutleverage to see the cut off score:



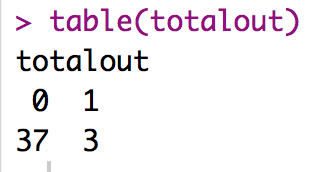
* + 1. Save the scores and see how many outliers:
       1. badleverage = as.numeric(leverage > cutleverage)
       2. table(badleverage)



* + 1. We have two outliers.
  1. Cook’s scores:
     1. Remember that Cook’s is a measure of influence and discrepancy.
     2. To get Cook’s values:
        1. cooks = cooks.distance(output)
     3. Get the cutoff score:
        1. 4 / (N-k-1)
        2. cutcooks = 4 / (nrow(*dataset*) - k - 1)
     4. Save the scores and see how many outliers:
        1. badcooks = as.numeric(cooks > cutcooks)
        2. table(badcooks)

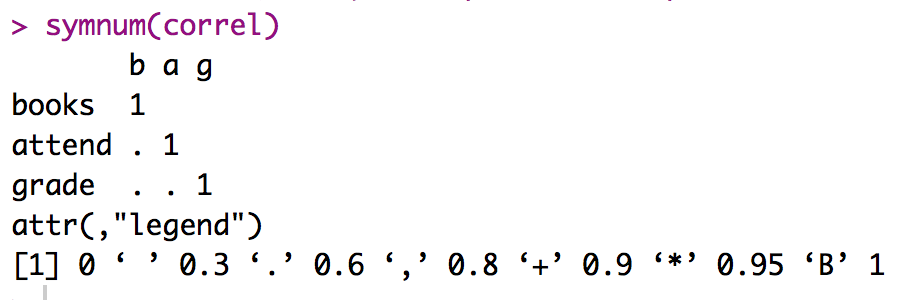


* + 1. We have one outlier!
  1. So, what does that mean overall?
     1. We want to create a total score for each participant of outliers.
     2. So, we add them up for total outlier-ness.
        1. totalout = badmahal + badleverage + badcooks
        2. table(totalout)
        3. Remember that top row = their score: 0, 1, 2, 3
        4. Bottom row is the number of people who have that score.

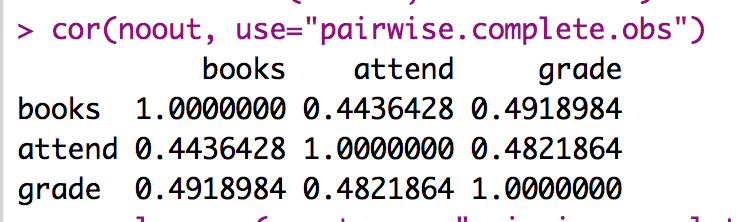


* + 1. Now, any people we have two or more problems need to get excluded:
       1. noout = subset(master, totalout < 2)
       2. Here we didn’t have anyone who was over multiple cut offs, but if we did, that code would exclude them.

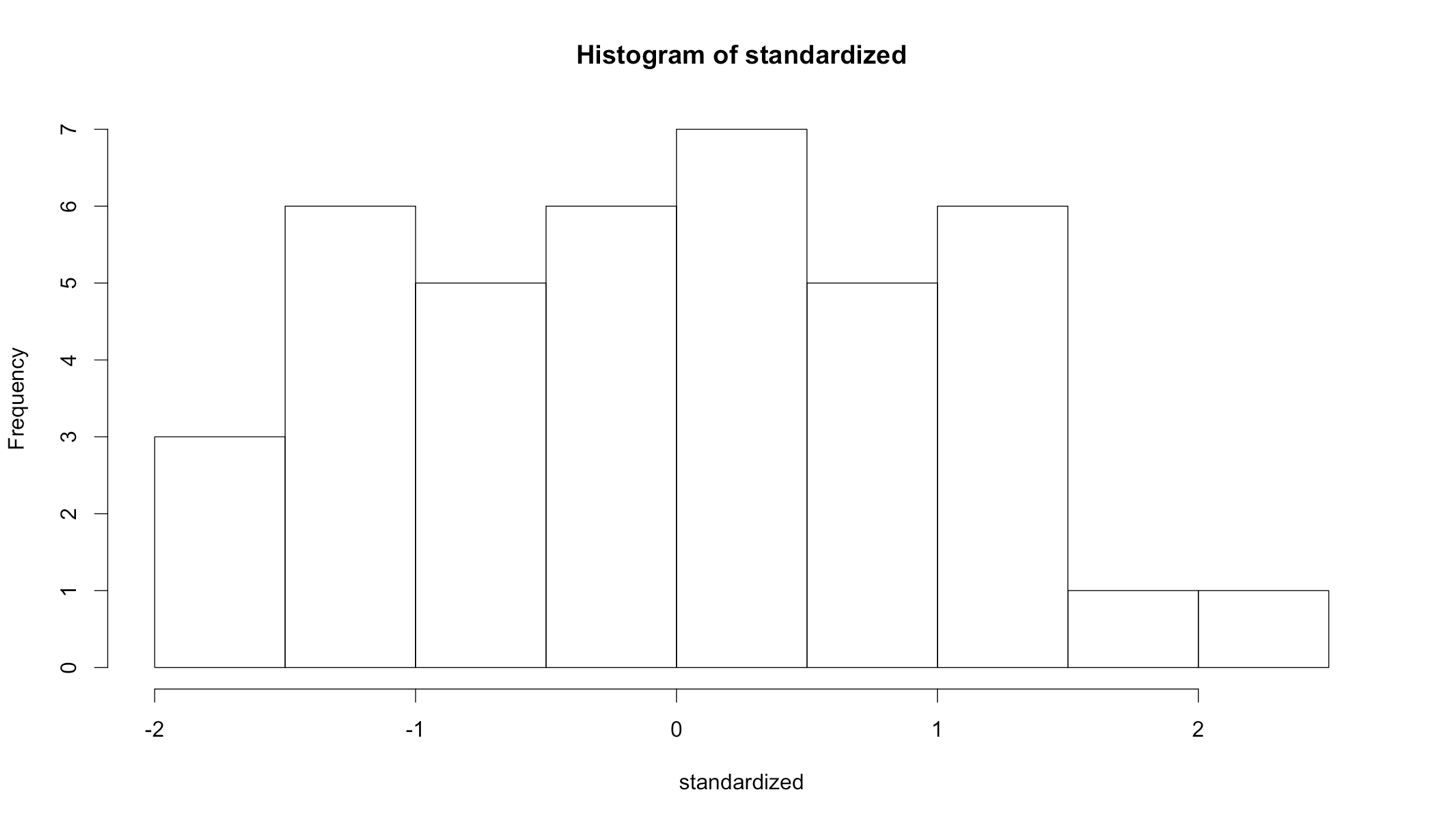
1. Additivity
   1. We do not want correlated IVs! It is a waste of time!
      1. The rule is technically *r* = .90.
      2. However, I would argue you shouldn’t really go over *r* = .70 – at that point the IVs are so correlated, you are risking suppression of one or both variables.
   2. Get the correlations:
      1. correl = cor(*dataset*, use = “pairwise.complete.obs”)
   3. Get the symbols chart:
      1. symnum(correl)
   4. Look for things with a , or higher.
   5. NOTE: I included the DV in this analysis (just to run it easily). You WANT strong correlations with the DV. So don’t exclude an IV if it’s correlated with the DV (doh!) or you’ve excluded your hypothesis!



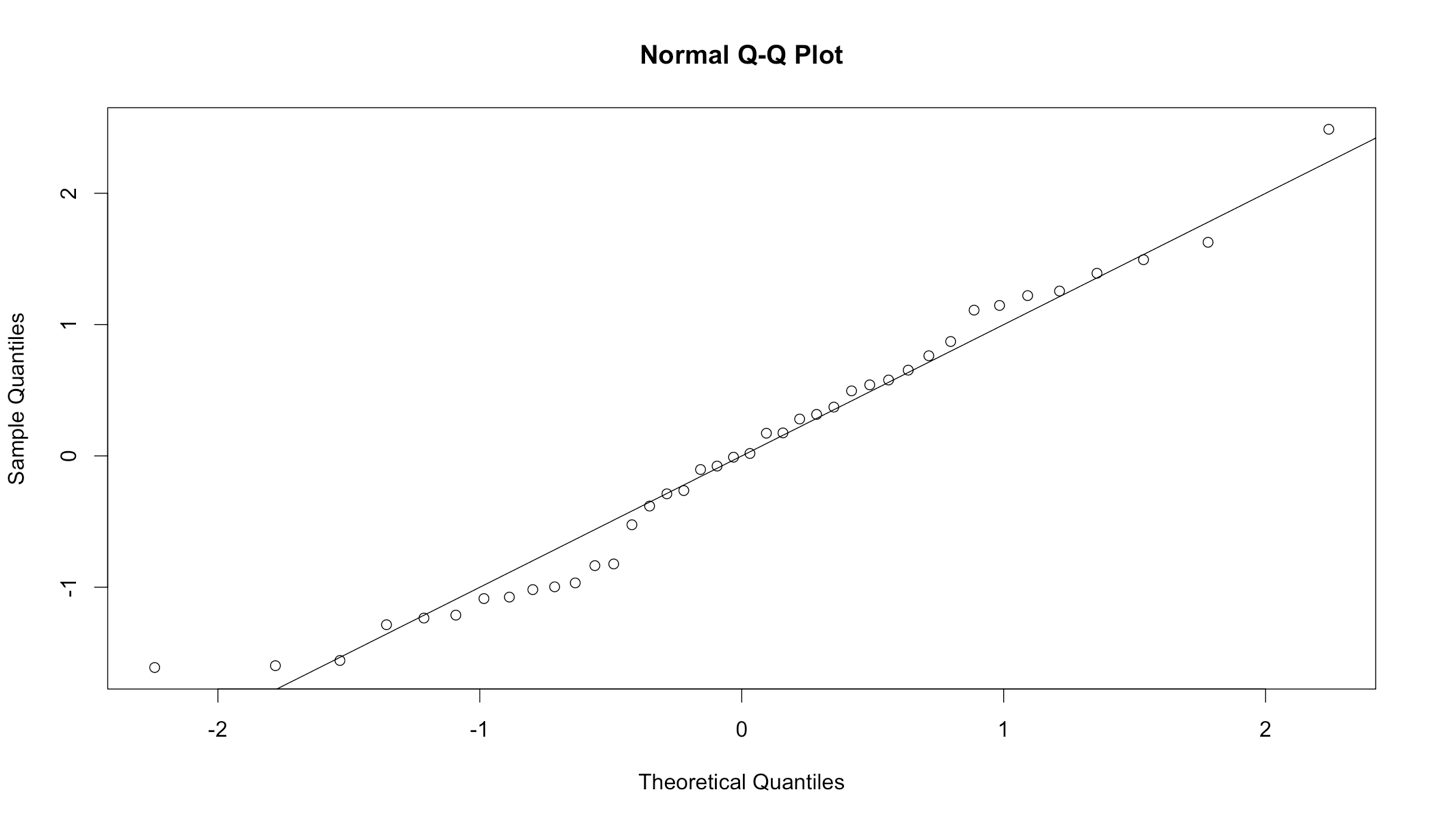
* 1. Also helps to look at the real correlations, since , = > .6 which is a lower cut off than we want, but it’s a good place to start looking for problems.



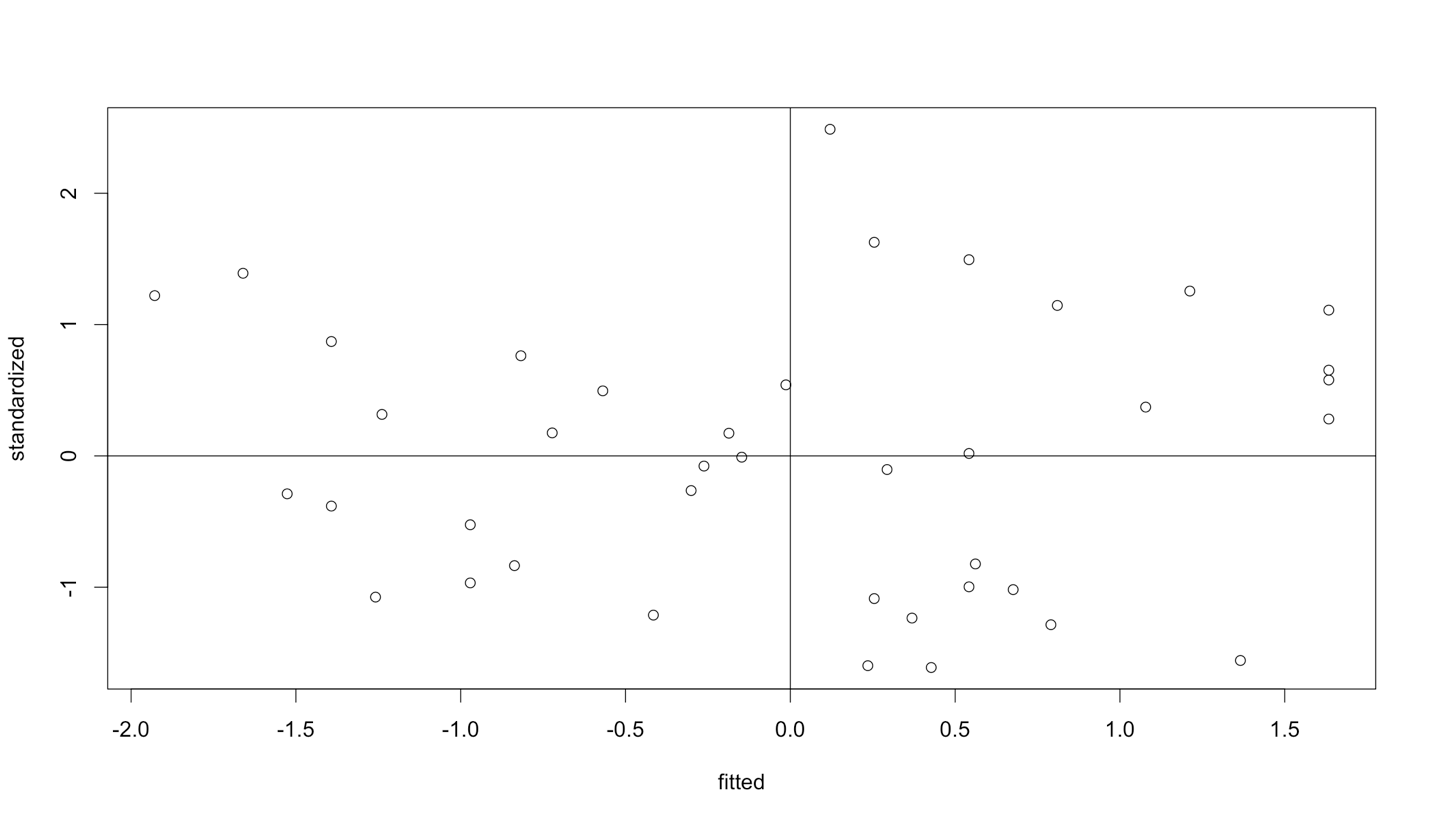
1. Set up the rest of the assumptions:
   1. Run the real analysis again with the no outliers dataset.
   2. No fake or randomness! It’s real regression!
   3. Create the standardized residuals:
      1. standardized = rstudent(output)
   4. Create the fitted values:
      1. fitted = scale(output$fitted.values)
2. Normality:
   1. hist(standardized)
   2. Most of the data is between -2 and 2 and is centered over 0. We are good to go!



1. Linearity:
   1. qqnorm(standardized)
   2. abline(0,1)
   3. Oh! Pretty!

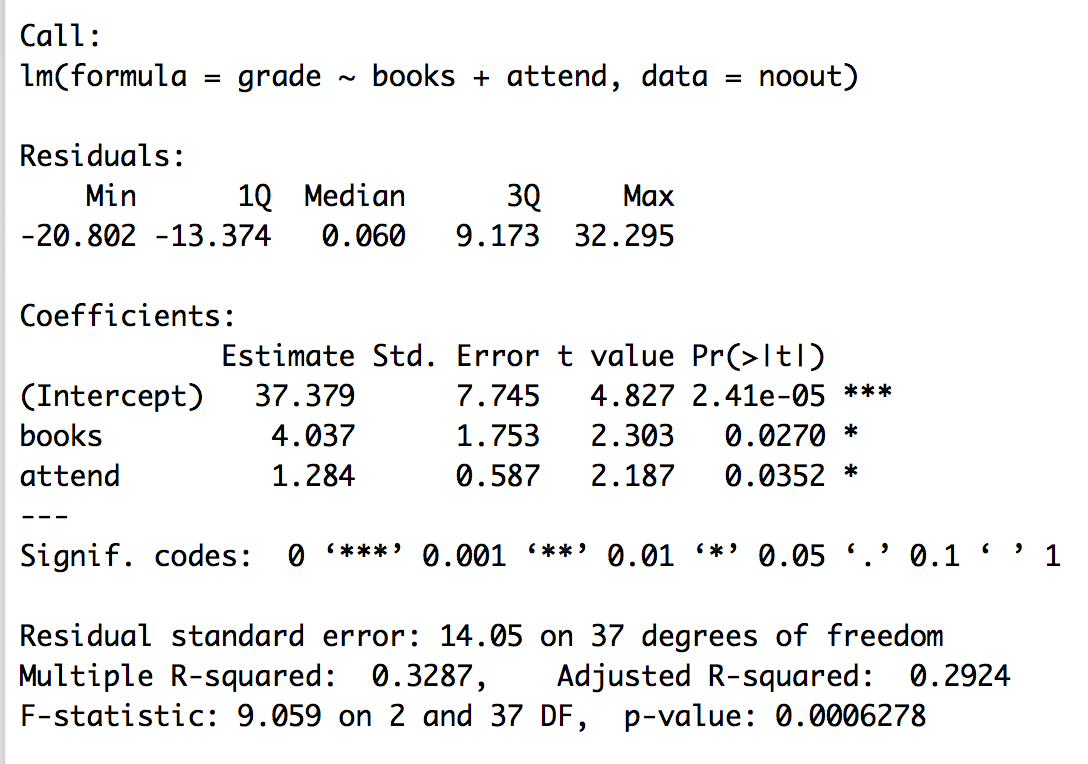


1. Homogeneity:
   1. plot(fitted,standardized)
   2. abline(0,0)
   3. abline(v = 0)
   4. Spread looks ok both vertically and horizontally.
2. Homoscedasticity:
   1. The spread around the line looks fairly blobby / uniform, so I would say this graph is ok.



**Running the real analysis:**

1. In this example, we are running a simultaneous regression, so we’ve been running the real analysis this whole time. But at this point for other types of regression, you want to back up and run step 1.
2. Get the regression and model:
   1. model = lm(*DV* ~ *IV* + *IV*, data = *noout*)
   2. summary(model)

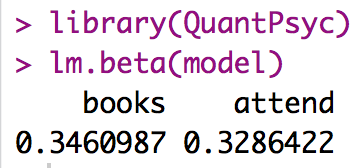


**Interpret the output:**

1. Is the overall model significant?
   1. Yes, *F*(2, 37) = 9.06, *p* < .001, *R2 =* .33
2. Are our individual predictors significant?
   1. Books, yes: *b* = 4.04, *t*(37) = 2.30, *p* = .03
   2. Attend, yes: *b* = 1.28, *t*(37) = 2.19, *p* = .04
3. Interpretation:
   1. As the number of books a person reads for class increases, so does their final grade.
   2. As the number of days attended increases, so does their final course grade.

**Beta:**

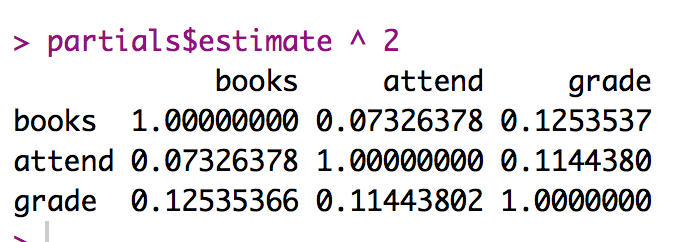
1. At the moment, I can’t compare books and attend because they are on different scales.
   1. We can usually use the *QuantPsy* package and the lm.beta function.
   2. library(QuantPsy)
   3. lm.beta(model)



* 1. Looking at these values, it appears that books has a slightly stronger relationship with grades than attendance (but not by much!).
  2. Remember these are standardized, NOT correlations. They can be negative and be larger than 1. Strength is about distance from zero, not sign (i.e. + or -).

**Effect Size:**

1. Another way to determine which predictor was stronger is to examine their effect sizes. We are going to use *pr2* for predictor effect size in the *ppcor* library with the *pcor* function.
   1. library(ppcor)
   2. partials = pcor(*dataset*, method = "pearson")
   3. partials$estimate ^ 2
      1. The second line calculates and saves the partial correlations – you want to control / exclude the variance for all other variables to see the correlation of the IV and DV.
      2. The third line squares and prints out the correlation table. We square the values because it’s effect size (remember *R* is squared for effect size).
      3. Special NOTE: these effect sizes are partials – they have different denominators and will not add up to *R2*.

****

* 1. Only look at the line with the DV.
  2. Books: *b* = 4.04, t(37) = 2.30, *p* = .03, *pr2* = .13
  3. Attend: *b* = 1.28, t(37) = 2.19, *p* = .04, , *pr2* = .11
  4. Again, we see that books are slightly more predictive.

**Make a picture:**

1. Pictures for regression analyses are not the most common thing, because they show the overall predicted values to the actual Y values (and not the IVs individually).
2. However, we will use them for some types of analyses, so let’s make some!
3. Load ggplot2.
   1. library(ggplot2)
4. Run some cleanup coding.
   1. cleanup = theme(panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.background = element\_blank(),

axis.line = element\_line(colour = "black"),

legend.key = element\_rect(fill = "white"),

text = element\_text(size = 15))

1. Get the finalized fitted values:
   1. fitted = model$fitted.values
2. Make the plot:
   1. scatter = ggplot(*dataset*, aes(*fitted, y values*))
3. Build that graph:
   1. scatter +

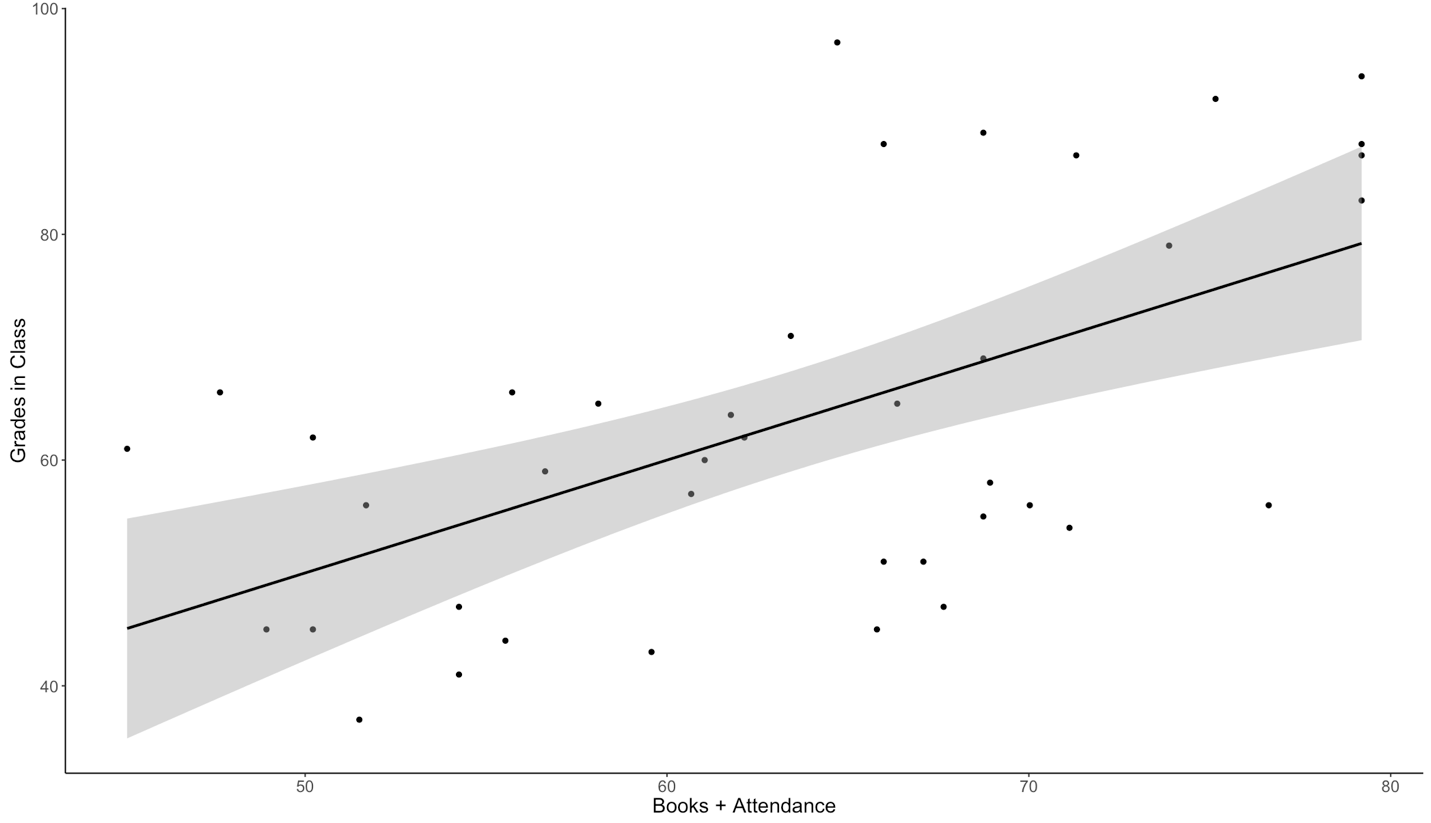
cleanup +

geom\_point() + ##to get the dots

geom\_smooth(method = “lm”, color = “black”) + ##to get the regression line

xlab(“Formula of line”) +

ylab(“DV”)



*Figure 1*. Predicted values of regression equation for books and attendance predicting final course grades. Gray band indicates 95% confidence interval.

**Results**

The number of books a student read per semester and their overall attendance in the semester was used to predict final course grade. The data were screened for missing data, outliers and regression assumptions. Although three participants met the cut off for Cook’s and Leverage values, they were included in the analysis because they did not have multiple outlier indicators (i.e. two of three values used for screening: Mahalanobis distance, Cook’s Leverage). Linearity, normality, additivity, homogeneity, and homoscedasticity were all met.

The overall regression model was significant, indicating the books and attendance combined predicted final course grade, *F*(2, 37) = 9.06, *p* < .001, *R2*= .33. As students read more books throughout the semester, they were more likely to increase their course grade, β = 0.35, *t*(37) = 2.30, *p* = .03, *pr2=*.13. Attendance was also a significant predictor of course grade, so that students who attended class more had higher grades, β = 0.33, *t*(37) = 2.19, *p* = .04, *pr2=*.11. See Figure 1 for a representation of the regression equation for this data.

NOTE: you can include b or beta in the write up, just be consistent.