Multimodel Inference

2020-02-07

Preparation

In this set of notes, we will learn about multimodel inference, an alternative to selecting a "best" model from a set of candidate models. To do so, we will use the *nels.csv* dataset (see the data codebook) to examine various prediction models to explain variation in students' school achievement.

```
# Load libraries
library(broom)
library(educate) #Need version 0.1.0.1
library(MuMIn)
library(patchwork)
library(tidyverse)

# Import data
nels = read_csv("~/Documents/github/epsy-8252/data/nels.csv")

# View data
head(nels)
```

```
# A tibble: 6 x 6
  social10 self_esteem  locus    ses   gpa parent_ed
     <dbl>       <dbl>  <dbl>  <dbl> <dbl>     <dbl>
1     61.7       -0.1  -0.14  -0.563   3.8         2
2     47.0       -0.45 -0.580  0.123   2.5         3
3     50.5        0.33 -0.59   0.229   2.8         3
4     56.5       -0.02  0.07   0.687   3.5         6
5     55.3       -0.09 -0.85   0.633   3.3         5
6     39.7       -0.28  0.07   0.992   2.5         5
```

Working Hypotheses and Candidate Models

Here, imagine a researcher who has five **scientific working hypotheses** about what underlies students' achievement:

- **H1:** Student achievement is a function of SES, prior achievment, and parent education level.
- **H2:** Student achievement is a function of SES, prior achievment, and parent education level, but also of the student's level of self-esteem.
- **H3:** Student achievement is a function of SES, prior achievment, and parent education level, but also of the student's degree of locus-of-control.
- **H4:** Student achievement is a function of SES, prior achievment, and parent education level, but also of the student's level of self-esteem and degree of locus-of-control.
- **H5:** Student achievement is a function of SES, prior achievment, and parent education level, but also of the interaction between student's level of self-esteem and degree of locus-of-control.

You should be able to tell by the scientific hypotheses, that this scenario represents a very exploratory approach to examining the underlying structure of the models (i.e., which predictors seem important). This is fine, we just need to acknowledge this so that our results re not seen as "final". Results from such an anlaysis are useful in helping plan a confirmatory analysis that can be carried out with new data.

The five working hypotheses are translated into the following five statistical models:

$$\text{Model 1}: \text{Achievement}_i = \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \epsilon_i$$

$$\begin{aligned}\text{Model 2}: \text{Achievement}_i =& \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \\ & \beta_4(\text{Self-Esteem}_i) + \epsilon_i\end{aligned}$$

$$\begin{aligned}\text{Model 3}: \text{Achievement}_i =& \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \\ & \beta_4(\text{Locus-of-Control}_i) + \epsilon_i\end{aligned}$$

$$\begin{aligned}\text{Model 4}: \text{Achievement}_i =& \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \\ & \beta_4(\text{Self-Esteem}_i) + \beta_5(\text{Locus-of-Control}_i) + \epsilon_i\end{aligned}$$

$$\begin{aligned}\text{Model 5}: \text{Achievement}_i =& \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \\ & \beta_4(\text{Self-Esteem}_i) + \beta_5(\text{Locus-of-Control}_i) + \beta_6(\text{Self-Esteem}_i)(\text{Locus-of-Control}_i) + \epsilon_i\end{aligned}$$

These models are then fitted to the data.

```
# Fit Model 1
lm.1 = lm(social10 ~ 1 + ses + gpa + parent_ed, data = nels)

# Fit Model 2
lm.2 = lm(social10 ~ 1 + ses + gpa + parent_ed + self_esteem, data = nels)

# Fit Model 3
lm.3 = lm(social10 ~ 1 + ses + gpa + parent_ed + locus, data = nels)

# Fit Model 4
lm.4 = lm(social10 ~ 1 + ses + gpa + parent_ed + self_esteem + locus, data = nels)

# Fit Model 5
lm.5 = lm(social10 ~ 1 + ses + gpa + parent_ed + self_esteem + locus +
            self_esteem:locus, data = nels)
```

We can compute information-theoretic summaries (e.g., AICc, model probabilities) for these candidate models. To do this, we will use the `model.sel()` function from the **MuMIn** package.

```
#Create table
model.sel(list(lm.1, lm.2, lm.3, lm.4, lm.5))
```

```
Model selection table
  (Int)   gpa prn_ed   ses slf_est   lcs lcs:slf_est df    logLik   AICc delta
3 34.42 5.064 0.4158 3.241         1.369              6 -2603.484 5219.1  0.00
4 34.46 5.050 0.4143 3.237  0.3487 1.165              7 -2603.284 5220.7  1.64
5 34.68 5.028 0.4171 3.223  0.3767 1.263     -0.7216  8 -2602.622 5221.4  2.36
2 34.15 5.169 0.4114 3.321  0.9476                    6 -2605.244 5222.6  3.52
1 33.78 5.304 0.4153 3.393                            5 -2607.343 5224.8  5.68
  weight
3  0.505
4  0.223
5  0.156
2  0.087
1  0.029
Models ranked by AICc(x)
```

The model evidence is less clear about which model is supported. The model having the most empirical evidence (given the data and candidate set of models) is Model 3 (the model that includes the locus-of-control main effect along with the three covariates). This model has a probability (AICc weight) of 0.51. This level of model evidence is far from overwhelming. Models 4 (Main effects of self-esteem and locus-of-control), 5 (the interaction model), and 2 also have a fair amount of empirical support.

Anytime we adopt a model from a set of candidate models (whether through information-theoretic approaches or through statistical inference), we have introduced some amount of uncertainty via the model selection process; referred to as **model selection uncertainty**.

Think about having multiple replicate sets of data and fitting the same candidate models to each set of replication data. We would likely find that the "best" model varies across replicate sets of data. This is model selection uncertainty.

The amount of the model selection uncertainty depends on the level of empirical evidence that supports one model over all others. In our example here, we would introduce quite a lot of model selection uncertainty since the empirical evidence for Model 3 is not overwhelming; the evidence also supports Models 4, 5, and 2.

This is important for a couple reasons. First, adopting one model when there is uncertainty that it is the "best" model may be misleading for the broader scientific enterprise. For example, consider trying to understand the effect of self-esteem on achievement. If we adopt Model 3 there is no effect of self-esteem. However, if Model 4 is actually closest to reality there is a positive effect of self-esteem on achievement. And, if Model 5 is closest to reality, then the effect of self-esteem on achievement depends on locus-of-control. Which are we to believe?

Another reason to think about model selection uncertainty is that any inferential results we report for a model are based around sampling variation being the only source of uncertainty. When there is model selection uncertainty, this is not the case. Increasing uncertainty directly affects the size of the standard errors which in turn affect the magnitude of the *p*-values and confidence intervals.

Multimodel Inference

One solution to dealing with the problem of model selection uncertainty is to use *multimodel inference*. In this approach, information from all (or many) the models included in the candidate set is used to make statistical inferences. This approach to inference is rather new to the statistical sciences and the most commonly used method in multimodel inference is **model averaging**.

Model Averaging

In model averaging, we compute the coefficients and standard errors for the various predictors by incorporating informatin from these estimates from all the models in the candidate set. Rather than simple averaging, the estimates are weighted by the model probabilities. A simple averaging would give us a coefficient of

$$\hat{\beta}_{\text{Self-Esteem}} = \frac{0 + 0.3487 + 0.3767 + 0.9476 + 0}{5} = 0.3346$$

However, this assumes that we believe each of the models equally. This is not true. The model probabilities we computed in Table 1 suggest that we should give the most weight to the coefficients in Model 3, then Model 4, etc. We can incorporate this by computing a weighted average where the weights we use are the model probabilities from the table of model evidence.

$$\hat{\beta}_{\text{Self-Esteem}} = \frac{0(0.505) + 0.3487(0.223) + 0.3767(0.156) + 0.9476(0.087) + 0(0.029)}{1} = 0.218$$

Note that the denominator in the weighted average is simply the sum of the weights, which is 1. In general, we can express the estimate for the kth coefficient, $\hat{\beta}_k$, as

$$\hat{\beta}_k = w_j(\beta_{k_j})$$

where w_j is the AICc weight (or model probability) for candidate model j, and β_{k_j} is the estimated coefficient for predictor k in candidate model j.

Conceptually, the weighting across each of the models *shrinks* the value of the coefficent toward 0. This happens because (1) the predictor is not included in some models and, (2) because the weights are less than 1. This shrinkage is how model averaging accounts for model selection uncertainty.

If we carry out the model averaging for each of the coefficients, we find:

$$\begin{aligned}
\hat{\beta}_0 &= 34.42(0.505) + 34.46(0.223) + 34.68(0.156) + 34.15(0.087) + 33.78(0.029) = 34.424 \\
\hat{\beta}_{\text{SES}} &= 3.241(0.505) + 3.237(0.223) + 3.223(0.156) + 3.321(0.087) + 3.393(0.029) = 3.248 \\
\hat{\beta}_{\text{GPA}} &= 5.064(0.505) + 5.050(0.223) + 5.028(0.156) + 5.169(0.087) + 5.304(0.029) = 5.071 \\
\hat{\beta}_{\text{Parent Education}} &= 0.4158(0.505) + 0.4143(0.223) + 0.4171(0.156) + 0.4114(0.087) + 0.4153(0.029) = 0.415 \\
\hat{\beta}_{\text{Locus-of-Control}} &= 1.369(0.505) + 1.165(0.223) + 1.263(0.156) + 0(0.087) + 0(0.029) = 1.147 \\
\hat{\beta}_{\text{Self-Esteem}} &= 0(0.505) + 0.3487(0.223) + 0.3767(0.156) + 0.9476(0.087) + 0(0.029) = 0.219 \\
\hat{\beta}_{\text{Locus-of-Control x Self-Esteem}} &= 0(0.505) + 0(0.223) - 0.7216(0.156) + 0(0.087) + 0(0.029) = -0.112
\end{aligned}$$

We can also obtain these values from the `model.avg()` function from the **MuMIn** package. This utputs two sets of model averaged coefficients. The output in the `full` row, corresponds to the values we just computed. (The weighted average is computed based on the full set of candidate models.)

```
# Model averaged estimates
model.avg(list(lm.1, lm.2, lm.3, lm.4, lm.5))
```

```
Call:
model.avg(object = list(lm.1, lm.2, lm.3, lm.4, lm.5))

Component models:
'1235'   '12345'  '123456' '1345'   '135'

Coefficients:
       (Intercept)      ses      gpa parent_ed    locus self_esteem
full      34.42402 3.248684 5.071313 0.4152767 1.147430   0.2186549
subset    34.42402 3.248684 5.071313 0.4152767 1.298565   0.4699563
       locus:self_esteem
full          -0.1122200
subset        -0.7215835
```

An alternative method of model averaging appears in the `subset` row. This *conditional averaging* only averages over the models where the parameter appears; it computes a weighted average over a subset of the candidate models. For example, the self-esteem predictor only appears in Models 4, 5, and 2, so the conditional weighted average would be computed as

$$\hat{\beta}_{\text{Self-Esteem}} = \frac{0.3487(0.223) + 0.3767(0.156) + 0.9476(0.087)}{0.223 + 0.156 + 0.087} = 0.4699$$

Using the full set of candidate models, the model averaging produces the following fitted model:

$$\begin{aligned}\widehat{\text{Achievement}}_i =& 34.42 + 3.25(\text{SES}_i) + 5.07(\text{Prior GPA}_i) + 0.42(\text{Parent Education}_i) + \\ & 0.22(\text{Self-Esteem}_i) + 1.15(\text{Locus-of-Control}_i) - 0.11(\text{Self-Esteem}_i)(\text{Locus-of-Control}_i)\end{aligned}$$

Estimates of the Unconditional Variance

Ideally, we would also like to obtain variance estimates for each of our coefficients. These typically give us an indication of the amount of sampling uncertainty in the coefficient. The variance estimates (or standard errors) we get from statistical output (e.g., `tidy()`) are *conditional on both the sample size and the model selected*. In other words, the numerical values for these estimates are computed using ML or OLS after the model and sample size are specified.

In a perfect world, we would use one set of data for the model selection process, and then use a second set of data (of identical sample size) to then obtain the coefficient and variance estimates by fitting the "best" model we previously identified. In practice, we tend to perform both model selection and estimation on the same data set. This, means that our variance estimates (which only assume sampling uncertainty), are too small, as they should be augmented to include model selection uncertainty in addition to the sampling uncertainty. Conceptually,

$$\text{Var}(\hat{\beta}_k) = \text{Sampling variance given a model} + \text{Variance due to model selection uncertainty}$$

Specifically, we can compute this updated variance estimate using,

$$\text{Var}(\hat{\beta}_k) = \sum w_i \left(\text{Var}(\hat{\beta}_{k_i}|\text{Model } i) + \left[\hat{\beta}_{k_i} - \hat{\bar{\beta}}_k \right]^2 \right)$$

where

- w_i is the model probability (AICc weight) for Model i,
- $\text{Var}(\hat{\beta}_{k_i}|\text{Model } i)$ is the sampling variance for $\hat{\beta}_k$ in Model i,
- $\hat{\beta}_k$ is the estimated coefficient in Model i, and
- $\hat{\bar{\beta}}_k$ is the model averaged value for $\hat{\beta}_k$ across all the candidate models.

The variance estimate is a weighted sum of two terms. The first term captures the sampling uncertainty for a particualar term ($\hat{\beta}_k$) given that it is in Model i. The second term captures the model selection uncertainty by measuring how much that coefficient changes from one model to another. Note that when the coefficient doesn't change much from one model to another, this second term is small and the overall variance for $\hat{\beta}_k$ is essentially the conditional sampling variance.

If, however, the coefficient changes a great deal from model-to-model, omitting this last term (as we usually do when we only report standard errors from `tidy()`), underestimates the amount of uncertainty for the coefficient. This error propogates through the inference by underestimating the size of the p-values and confidence intervals (both should be larger).

Computing the Unconditional Variance for the Self-Esteem Coefficient

We can obtain the estimated coefficients and the sampling variances from the `tidy()` output for each model. Note that this output gives the conditional standard errors for the coefficients which can be squared to obtain the sampling variances. The AICc weights are part of the `model.sel()` output.

Model	w_i	$\hat{\beta}_k$	$\text{Var}(\hat{\beta}_{k_i}\|\text{Model } i)$
1	0.029	0.0000	0.0000000
2	0.087	0.9476	0.2147396
3	0.505	0.0000	0.0000000
4	0.223	0.3487	0.3059196
5	0.156	0.3767	0.3063622

We can compute the term for the model selection uncertainty by substituting in the model averaged value for $\hat{\bar{\beta}}_k$ that we computed earlier using the `model.avg()` function. For the self-esteem coefficient, the model averaged value is 0.22. For example, computing the model selection uncertainty term for Model 1,

$$\begin{aligned} \left[\hat{\beta}_{k_i} - \hat{\bar{\beta}}_k \right]^2 &= \left[0 - 0.22 \right]^2 \\ &= 0.0484 \end{aligned}$$

Model	w_i	$\hat{\beta}_k$	$\text{Var}(\hat{\beta}_{k_i}\|\text{Model } i)$	$\left[\hat{\beta}_{k_i} - \bar{\hat{\beta}}_k\right]^2$
1	0.029	0.0000	0.0000000	0.0479463
2	0.087	0.9476	0.2147396	0.5309068
3	0.505	0.0000	0.0000000	0.0479463
4	0.223	0.3487	0.3059196	0.0168308
5	0.156	0.3767	0.3063622	0.0248799

Finally we sum the conditional variance and the model selection variance terms and multiplying by the AICc weights. For example, for the first model this turns out to be,

$$\begin{aligned} w_i\left(\text{Var}(\hat{\beta}_{k_i}|\text{Model } i) + \left[\hat{\beta}_{k_i} - \bar{\hat{\beta}}_k\right]^2\right) &= 0.029\Big(0.0000000 + 0.0479463\Big) \\ &= 0.001390443 \end{aligned}$$

Model	w_i	$\hat{\beta}_k$	$\text{Var}(\hat{\beta}_{k_i}\|\text{Model } i)$	$\left[\hat{\beta}_{k_i} - \bar{\hat{\beta}}_k\right]^2$	$w_i\left(\text{Var}(\hat{\beta}_{k_i}\|\text{Model } i) + \left[\hat{\beta}_{k_i} - \bar{\hat{\beta}}_k\right]^2\right)$
1	0.029	0.0000	0.0000000	0.0479463	0.0013904
2	0.087	0.9476	0.2147396	0.5309068	0.0648712
3	0.505	0.0000	0.0000000	0.0479463	0.0242129
4	0.223	0.3487	0.3059196	0.0168308	0.0719733
5	0.156	0.3767	0.3063622	0.0248799	0.0516738

These terms are added together to obtain the estimate of the unconditional variance for the self-esteem coefficient.

$$\begin{aligned} \text{Unconditinal Variance}(\hat{\beta}_{\text{Self-Esteem}}) &= 0.0013904 + 0.0648712 + 0.0242129 + 0.0719733 + 0.0516738 \\ &= 0.2141216 \end{aligned}$$

We can convert this to a standard error (a more common metric for reporting sampling variation) by computing the square root of this value.

$$\text{Unconditinal SE}(\hat{\beta}_{\text{Self-Esteem}}) = \sqrt{0.2141216} = 0.4627328$$

Automating the Computation of the Unconditinal SEs

We can use the `coefTable()` function from the **MuMIn** package to automate the computation of both the model averaged coefficients and unconditional standard errors for each of the terms in the candidate models. This function takes the output from `model.avg()` as its main input. We also use the argument `full=TRUE` to compute the model averaged coefficients and unconditional standard errors based on all of the models in the candidate set.

```
# Coefficient table using full set of models
coefTable(
  model.avg(list(lm.1, lm.2, lm.3, lm.4, lm.5)),
  full = TRUE
  )
```

```
                  Estimate Std. Error
(Intercept)       34.42402     1.5719
ses                3.24868     0.5872
gpa                5.07131     0.4477
parent_ed          0.41528     0.2416
locus              1.14743     0.6607
self_esteem        0.21865     0.4630
locus:self_esteem -0.11222     0.3609
```

As a note, this output can be piped into the `kable()` function if you are using RMarkdown.

Confidence/Compatibility Intervals for Inference

The model averaged coefficients and unconditional standard errors can be used to create compatibility intervals (confidence intervals) using the Wald method,

$$\hat{\beta}_k \pm 1.96\Big[\text{SE}(\hat{\beta}_k)\Big]$$

Here we compute these intervals for each coefficient in the model. We first coerce the output from `coefTable()` into a data frame.

```
# Coerce table into a data frame
tab = coefTable(
  model.avg(list(lm.1, lm.2, lm.3, lm.4, lm.5)),
  full = TRUE
  ) %>%
  data.frame()

# View data frame
tab
```

```
                    Estimate Std..Error df
(Intercept)       34.4240160  1.5719387 NA
ses                3.2486841  0.5872096 NA
gpa                5.0713130  0.4476832 NA
parent_ed          0.4152767  0.2416334 NA
locus              1.1474305  0.6606630 NA
self_esteem        0.2186549  0.4629545 NA
locus:self_esteem -0.1122200  0.3608926 NA
```

Then we can compute the lower and upper limits for the compatibility interval using the Wald method. To make things look good, we also add a column that includes the predictor names. (Note these are the rownames of the `coefTable()` output.) We also rename the `Std..Error` column to `SE`, and drop the `df` column from the table.

```
# Compute lower limit (LL) and upper limit (UL) for CI
tab = tab %>%
  mutate(
    LL = Estimate - 1.96*Std..Error,
    UL = Estimate + 1.96*Std..Error
  ) %>%
  mutate(
    Term = row.names(tab)
  ) %>%
  rename(SE = Std..Error) %>%
  select(Term, Estimate, SE, LL, UL)

# View updated table
tab
```

```
               Term   Estimate        SE          LL         UL
1       (Intercept) 34.4240160 1.5719387 31.34301620 37.5050157
2               ses  3.2486841 0.5872096  2.09775334  4.3996148
3               gpa  5.0713130 0.4476832  4.19385392  5.9487720
4         parent_ed  0.4152767 0.2416334 -0.05832489  0.8888782
5             locus  1.1474305 0.6606630 -0.14746896  2.4423299
6       self_esteem  0.2186549 0.4629545 -0.68873590  1.1260458
7 locus:self_esteem -0.1122200 0.3608926 -0.81956949  0.5951295
```

Coefficient Plot

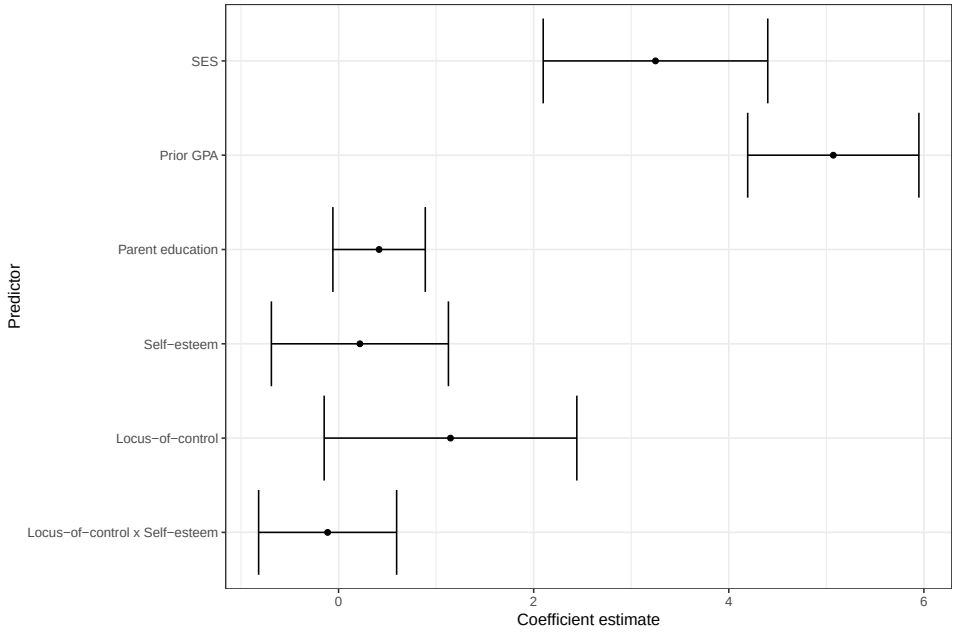
We can create a coefficient plot of these intervals by plotting each CI. To do this we:

- Filter out the intercept row so that it is omitted from the plot (after all we are interested in the effects of the predictors);
- Since the `Term` values will be plotted in alphabetical order, we coerce this column into a factor so that we can specify the order that the terms will be plotted, consistent with how they are presented in the model (e.g., the interactin is presented at the bottom of the plot);
- Add the compatibility interval using the `geom_errorbarh()` function, which draws an error bar horizontally between the values identiied in `xmin=` and `xmax=`, respectively;
- Add the coefficient estimate to the error bar as a point.

```r
tab %>%
  filter(Term != "(Intercept)") %>%
  mutate(
    Term = factor(Term,
      levels = c("locus:self_esteem", "locus", "self_esteem", "parent_ed", "gpa", "ses"),
      labels = c("Locus-of-control x Self-esteem", "Locus-of-control", "Self-esteem",
                 "Parent education", "Prior GPA", "SES"),
    )
  ) %>%
  ggplot(aes(x = Estimate, y = Term)) +
    geom_errorbarh(aes(xmin = LL, xmax = UL)) +
    geom_point() +
    theme_bw() +
    xlab("Coefficient estimate") +
    ylab("Predictor")
```

Figure 1

Coefficient Plot using the Model Averaged Coefficients and Unconditional Standard Errors

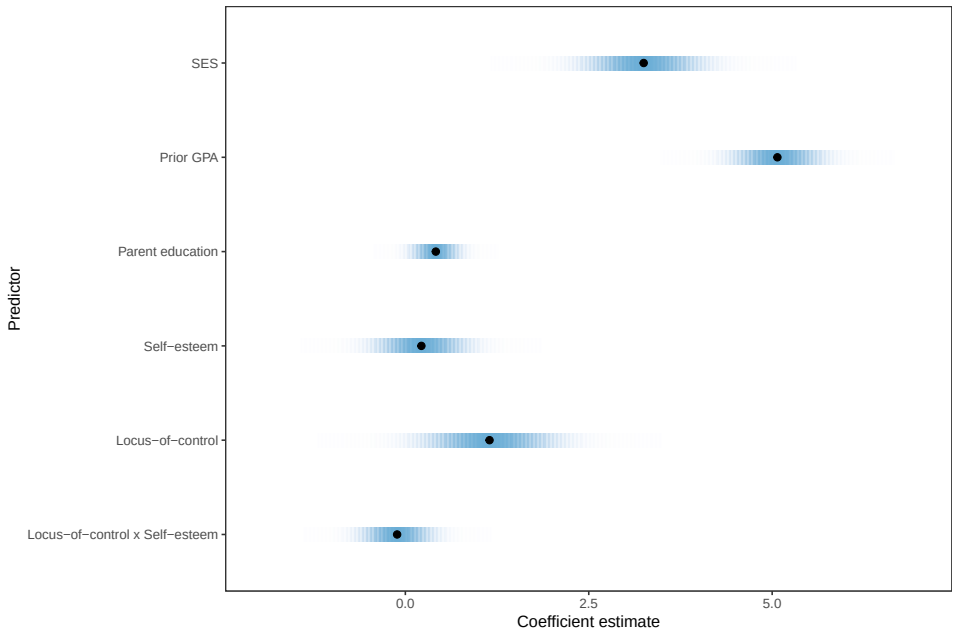


You could also use the functions from the **ungeviz** package to produce this plot as well! See the Introduction to Multiple Regression Notes from EPsy 8251 for a reminder of how to do this.

Remember that the **ungeviz** package cannot be downloaded from CRAN. Follow the installation instructions here to install this package.

Figure 2

Coefficient Plot (Created with ungeviz) using the Model Averaged Coefficients and Unconditional Standard Errors



Based on this plot we make the following inferences:

- There is a great deal of uncertainty about the direction of the interaction effect between locus-of-control and self-esteem (it may be negative, have no effect, or be a positive effect), after controlling for the parent education, prior GPA and SES. Similarly, we have quite a bit of uncertainty about the self-esteem main effect after controlling for the other factors in the model.
- There is some uncertainty about the direction of the locus-of-control main effect and the parent education main-effect, after controlling for the other factors in the model. However, most of the compatibility interval is on the positive side of 0, suggesting that the evidence is beginning to point toward a positive effect. (Perhaps another study could add be used to further elucidate the direction of these two effects.) It is worth noting that the magnitude of the locus-of-control interval is large, indicating that although we are thinking that the effect is positvely associated with achievement, we are incredibly uncertain about the size of this effect relative to the magnitude of the parent education effect.
- The compatibility intervals for the SES and GPA main effects suggest that both are positively related to achievement, after controlling for the other factors in the model. We are somewhat uncertain about the magnitude of these relationships, but it does seem that prior GPA has a larger effect on achievement than SES.

Additional studies (i.e., collecting new data) could be used to try and replicate the direction of the effects. Carrying these studies out with larger sample sizes would reduce the uncertainty and also help illuminate the direction for those effects we were less sure about the direction for. (Although the sample size in this study was already $n = 744$.)