Week 5 Software Lesson: Multiple Linear Regression, Part 4 in R

Goals

The goal of this software lesson is to learn how to

- create graphical summaries (i.e., interaction plot) when there is a potential interaction, and
- include interaction terms in a model and obtain model-based estimates.

Required Packages

You will need to install the following packages in R for this software lesson:

- `jtools`
- `forcats`
- `interactions`
- `emmeans`

DATASET

Let's use the Lung Health Study (LHS) again. Recall this was the dataset used in software lessons in the previous course (PubH 6450: Biostatistics 1).

Recall that the dataset and data dictionary can be found on Canvas, *lhs.csv* and *lhsdoc.docx*, respectively.

```r
lhs <- read.csv(file = file.choose(), header = TRUE)
```

INTERACTIONS

Thus far, you have only dealt with models that are additive (e.g., $y = b_0 + b_1x_1 + b_2x_2$), with continuous or categorical predictors. If there is a binary predictor and a continuous predictor in an additive model, the two groups have similar slopes but different intercepts. However, it may be the case that the two groups have different intercepts AND different slopes. This is a situation of an interaction between the predictors (e.g., $y = b_0 + b_1x_1 + b_2x_2 + b_3x_1x_2$).

In general, interactions can be between any type of predictors: continuous, binary, or multicategorical. This software lesson will present the situation of an interaction term between one binary and one continuous predictor, as well as two binary predictors. The former situation will be presented first followed by the latter.

Interaction between one binary and one continuous predictor

Suppose the Lung Health Study researchers were interested in the following research question:

Does the effect of intervention (smoking intervention vs. usual care) on lung function at 5th year depend on cigarettes smoked per day at baseline?

When dealing with a potential interaction situation, it's helpful to visualize the data prior to obtaining the regression results from the interaction model.

However, before you proceed, notice that the question is about comparing two groups, and not three (as in the original intervention variable). Therefore, to answer this question, you first have to create a new variable for the intervention variable that takes the three interventions (SI-A, SI-P, UC) and collapses the smoking interventions into a single group (SI) so that you are left with two interventions (SI, UC).

Collapse categories in a categorical variable

- To collapse categories of a categorical predictor,
  - create a new variable in the dataset (e.g., `lhs$group2 <-`) and use the `fct_collapse()` function from the `{forcats}` package, specifying the name of the variable you want to collapse and then the new label for the category in quotation marks (e.g., `"SI" =`), followed by the categories that you want to collapse, and
  - (optional) use the base-R pipe operator (`|>`) to string the first task with the `relevel()` function if you want to set the reference group for the new variable:

```r
lhs$groups2 <- fct_collapse(lhs$alphagroup, "SI" = c("SI-A", "SI-P")) |>
               relevel(lhs$alphagroup, ref = "UC")
```

Be sure to check that the code carried out the task as you intended (e.g., `table(lhs$alphagroup)` to check for counts in the groups and then `table(lhs$groups2)` to check that the counts of SI equal the sum of SI-A and SI-P from the original variable).

Create interaction plot

For situations that involve a two-variable interaction, you can visualize the interaction by creating an interaction plot. There are two different ways of plotting this, depending on the types of variables that are interacting.

- To create an interaction plot for a categorical predictor and a continuous predictor,
  - first, save and fit a lm model (e.g., `model1`) that includes all of the variables for the model but putting a star, `*`, between the variables that are interacting (Note: R automatically includes the main effects of the predictors in addition to interaction term),
  - then use the `interact_plot()` function from the `{interactions}` package, specifying
    - the model with the interaction term (`model=`),
    - the continuous predictor for the x-axis (`pred=`) and the categorical predictor for the lines (`modx=`), and
    - any additional arguments for customizing the plot (e.g., `plot.points=TRUE` to plot the points in addition to the estimated regression lines; `main.title=` to add a title to the plot; `x.label=` or `y.label=` to change the axes labels; `legend.main=` to change the legend label):

```r
##Fitting the interaction model
model1 <- lm(FEVFVC5 ~ f10cigs*groups2, data = lhs)

##Creating the interaction plot
interact_plot(model = model1,
              pred = f10cigs,
              modx = groups2,
              plot.points = TRUE,
              x.label = "Number of Cigarettes at Baseline",
              legend.main = "Intervention Groups",
              y.label = "FEV1/FVC % at year 5 annual visit",
              main.title = "Interaction Plot")
```

Estimated regression lines are displayed for each group.

Additional Resources

For additional R resources on interaction plots, see:

- Exploring interactions with continuous predictors in regression models (Jacob Long)

Fit a model with interaction term

You already fit a model with an interaction term in order to create the interaction plot, so all that is left after visualizing the data is to examine the regression model results.

- To examine the results from the model, use the `summ()` function from the `{jtools}` package on the saved `lm` object (`model1`) to obtain the regression coefficient values, R^2 and adjusted R^2 values, confidence intervals (`confint=TRUE`), and any additional arguments for customizing the output (e.g., `ci.width=` to specify the confidence level; `digits=` to change number of digits reports):

```
summ(model1, confint = TRUE, ci.width = 0.95, digits = 4)
```

The output is similar to the output shown previously, except now you have one row for the interaction term between `f10cigs` and `groups2`. You should examine the significance of the interaction term to determine what the next step will be.

Interaction Significant? Obtain slopes of the continuous predictor for each category

If an interaction exists, you can obtain the continuous predictor's slope for each category.

Although the interaction between intervention group and number of cigarettes in this scenario isn't significant, the code for obtaining this information will be presented for demonstration purposes.

- To obtain group slopes when an interaction is significant, first, save the information under an object name (e.g., `model1.slopes <-`), and then use the `emtrends()` function from the `{emmeans}` package, specifying the model, the `pairwise ~` argument, specifying the categorical predictor, followed by a vertical line, `|`, and then the continuous predictor, and then the `var =` argument, specifying the continuous predictor in quotation marks. Finally, pull out the slope by group information from the `emtrends` object (`model1.slopes$emtrends`) to obtain slope values (descriptive and inferential):

```
model1.slopes<-emtrends(model1, pairwise ~ groups2|f10cigs, var = "f10cigs")

model1.slopes$emtrends
```

The values presented include the slopes (`...trend`), the model-estimated standard errors (`SE`), degrees of freedom for the confidence interval (`df`), and the lower and upper values for the 95% confidence interval (`lower.CL` and `upper.CL`) for each group.

Interaction NOT Significant? Obtain adjusted slopes for each main effect

Recall that if the interaction term is not statistically significant, you might consider dropping it from the model and fit a main effects model instead. Note: This week is the first time you've seen the phrase "main effects", which essentially means predictors that are not interacting with one another and their regression coefficients are thus adjusted for the other variables in the model. When a model doesn't have an interaction term in it, it's called a "main effects model".

- To specify a main effects model, use the code you have seen thus far, but include the `+` operator between the two predictors instead of the star, `*`, operator:

```
model2 <- lm(FEVFVC5 ~ groups2 + f10cigs, data = lhs)

summ(model2, confint = TRUE, ci.width = 0.95, digits = 4)
```

The output is similar to the output shown previously.

Interaction between two binary predictors

Unlike in the previous interaction scenario, where there was a slope for a continuous variable for each group, when there is an interaction term with two binary predictors, you obtain group means and can examine the pairwise difference between group means.

Now suppose the Lung Health Study researchers were interested in the following research question:

Does the effect of intervention (smoking intervention vs. usual care) on lung function at 5th year depend on biological sex?

Let’s go through a similar approach as show in the previous scenario.

Create interaction plot

- To create an interaction plot for two categorical predictors,
    - first, save and fit a lm model (e.g., `model3`) that includes all of the variables for the model but putting a star, `*`, between the variables that are interacting (Note: R automatically includes the main effects of the predictors in addition to interaction term),
    - then use the `cat_plot()` function from the `{interactions}` package, specifying
        - the model with the interaction term (`model=`),
        - one of the categorical predictors for the x-axis (`pred=`) and the other categorical predictor for the lines (`modx=`), and
        - any additional arguments for customizing the plot (e.g., `interval=FALSE` to not display the confidence intervals for the values; `geom=` to change type of plot to points only, `"point"`, or lines + plots, `"line"`; `main.title=` to add a title to the plot; `x.label=` or `y.label=` to change the axes labels; `legend.main=` to change the legend label):

```
##Fitting the interaction model
model3 <- lm(FEVFVC5 ~ groups2*AGENDER, data = lhs)

##Creating the interaction plot
cat_plot(model = model3,
         pred = groups2,
         modx = AGENDER,
         interval = FALSE,
         geom = "line",
         x.label = "Intervention Groups",
         legend.main = "Sex",
         y.label = "FEV1/FVC % at year 5 annual visit",
         main.title = "Interaction Plot")
```

Lines and points (which are the observed group means) are displayed for each group combination.

Additional Resources

For additional R resources on interaction plots, see:

- Plotting interactions among categorical variables in regression models vignette (Jacob Long)

Fit a model with interaction term

You already fit a model with an interaction term in order to create the interaction plot, so all that is left after visualizing the data is to examine the regression model results.

- To examine the results from the model, use the `summ()` function from the `{jtools}` package on the saved `lm` object (`model3`) to obtain the regression coefficient values, R^2 and adjusted R^2 values, confidence intervals (`confint=TRUE`), and any additional arguments for customizing the output (e.g., `ci.width=` to specify the confidence level; `digits=` to change number of digits reports):

```
summ(model3, confint = TRUE, ci.width = 0.95, digits = 4)
```

The output is similar to the output shown previously, except now you have one row for the interaction term between `groups2` and `AGENDER`. You should examine the significance of the interaction term to determine what the next step will be.

Interaction Significant? Obtain group means and pairwise differences between group means

If an interaction exists, you can obtain statistics for the group means and also differences between the group means.

Although the interaction between intervention group and sex in this scenario aren't significant, the code for obtaining this information will be presented for demonstration purposes.

- To obtain group means when an interaction is significant, first, save the information under an object name (e.g., `model3.stats <-`), and then use the `emmeans()` function from the `{emmeans}` package, specifying the model, and the `pairwise ~` argument to match the model of interest. Finally, pull out the group mean information from the `emmeans` object (`model3.stats$emmeans`) to obtain summary results (descriptive and inferential):

```
model3.stats<-emmeans(model3, pairwise ~ groups2*AGENDER)

model3.stats$emmeans
```

The values presented include the group mean (`emmean`), the model-estimated standard errors (`SE`), degrees of freedom for the confidence interval (`df`), and the lower and upper values for the 95% confidence interval (`lower.CL` and `upper.CL`) for each group.

- To obtain pairwise differences between the group means when an interaction is significant, pull out the pairwise differences information ("contrasts") from the `emmeans` object (`model3.stats$contrasts`), then use the base-R pipe operator (`|>`) to string the first task with the `summary(infer=TRUE)` function to obtain summary results (descriptive and inferential):

```
model3.stats$contrasts |>
     summary(infer = TRUE)
```

The values include differences between the group means (`estimate`), the model-estimated standard errors for the difference between group means (`SE`), degrees of freedom for the test statistic (`df`), the lower and upper values for the 95% confidence interval (`lower.CL` and `upper.CL`) for the differences, the t-statistic (`t.ratio`), and its *p*-value (`p.value`), with the latter two adjusted for multiple comparisons using Tukey method.

Interaction NOT Significant? Obtain adjusted means and average differences between adjusted means

Recall that if the interaction term is not statistically significant, you might consider dropping it from the model and fit a main effects model instead.

- To specify a main effects model, use the code you have seen thus far, but include the `+` operator between the two predictors instead of the star, `*`, operator:

```
model4 <- lm(FEVFVC5 ~ groups2 + AGENDER, data = lhs)
```

Then you can obtain statistics for the model-predicted group means (i.e., adjusted means) for the group combinations as well as for each individual categorical predictor.

- To obtain the model-predicted group means for the group combination, use the `emmeans()` function again, but modify the `pairwise~` part to match the main effects model:

```
model4.stats<-emmeans(model4, pairwise ~ groups2 + AGENDER)

model4.stats$emmeans
```

The values presented in the group means output include the model-estimated group means (`emmean`; a.k.a. adjusted means for each group combination). Note: This is essentially putting 1's and 0's in to the regression model equation to get adjusted means for each of the group combinations. The other values are the model-estimated standard errors (`SE`), degrees of freedom for the confidence interval (`df`), and lower and upper values for the 95% confidence interval (`lower.CL` and `upper.CL`) for each group combination.

- To obtain the model-predicted adjusted means for each individual predictor, use the `emmeans()` function again, but modify the `pairwise~` part to match the variable you want to estimate, and pull out adjusted means (`$emmeans`) or differences between groups (`$contrasts |> summary(infer=TRUE)`):

```
###Main effects for intervention
model4.stats.groups2<-emmeans(model4, pairwise ~ groups2)

model4.stats.groups2$emmeans #adjusted means for each intervention

model4.stats.groups2$contrasts |>
     summary(infer = TRUE) #descriptive and inferential statistics for main effects
                           #(differences between interventions)

###Main effects for sex
model4.stats.sex<-emmeans(model4, pairwise ~ AGENDER)

model4.stats.sex$emmeans

model4.stats.sex$contrasts |>
     summary(infer = TRUE) #descriptive and inferential statistics for main effects
                           #(differences between sexes)
```

The output is similar to before, except that the results are presented for each predictor individually. The information presented in the `$emmeans` output are values for each category of the predictor and the information presented in the `$contrasts` output is value for the average differences between adjusted means (i.e., main effect for the predictor).